cHiPSet

cHiPSet - Research Work Results

Grant Period 1

COST
EUROPEAN COOPERATION IN SCIENCE & TECHNOLOGY

Funded by the Horizon 2020 Framework Programme of the European Union

Preface

This document is the report of the research work of the cHiPSet members. It surveys the recent developments on infrastructures and middleware for Big Data modelling and simulation. The detailed taxonomy of Big Data systems for life, physical, and socio-economical applications. is also presented.

Contents

List of Contributors

Ewa Niewiadomska-Szynkiewicz
Institute of Control and Computation Engineering, Warsaw University of Technology, Poland,
e-mail: ens@ia.pw.edu.pl

Rajendra Akerkar
Western Norway Research Institute, Norway,
e-mail: rak@vestforsk.no

Luca Agnello
Department of Biopathology and Medical Biotechnologies, University of Palermo, Italy,
e-mail: luca.agnello@gmail.com

Marcel Antal
Technical University of Cluj-Napoca, Romania,
e-mail: Marcel.Antal@cs.utcluj.ro

Piotr Arabas
Institute of Control and Computation Engineering, Warsaw University of Technology, Poland,
e-mail: parabas@ia.pw.edu.pl

Albert Comelli
Department of Biopathology and Medical Biotechnologies, University of Palermo, Italy,
e-mail: albert.comelli@gmail.com

Ciprian Dobre
University Politehnica of Bucharest, Romania,
e-mail: ciprian.dobre@cs.pub.ro

Silvia Franchini
University of Palermo, Italy,
e-mail: silvia.franchini@unipa.it

Daniel Grzonka

Institute of Computer Science, Cracow University of Technology, Poland,
e-mail: grzonka.daniel@gmail.com

Mauro Iacono
Seconda Università degli Studi di Napoli, Italy,
e-mail: mauro.iacono@unina2.it

Agnieszka Jakóbik
Institute of Computer Science, Cracow University of Technology, Poland,
e-mail: agneskrok@gmail.com

Michał Karpowicz
Research and Academic Computer Network, Warsaw, Poland,
e-mail: michal.karpowicz@nask.pl

Irene Kilanioti
Department of Computer Science, University of Cyprus, Nicosia, Cyprus,
e-mail: irenekilanioti@gmail.com

Joanna Kołodziej
Institute of Computer Science, Cracow University of Technology, Poland,
e-mail: jokolodziej@pk.edu.pl

Olga Kurasova
Vilnius University, Institute of Mathematics and Informatics, Lithuania,
e-mail: Olga.Kurasova@mii.vu.lt

Michał Marks
Research and Academic Computer Network, Warsaw, Poland,
e-mail: michal.marks@nask.pl

George Mastorakis
Technological Educational Institute of Crete, Greece,
e-mail: gmastorakis@staff.teicrete.gr

Constandinos Mavromoustakis
Department of Computer Science, University of Nicosia, Cyprus,
e-mail: mavromoustakis.c@unic.ac.cy

Viktor Medvedev
Vilnius University, Institute of Mathematics and Informatics, Lithuania,
e-mail: Viktor.Medvedev@mii.vu.lt

Valentina Nejkovic
University of Nis, Faculty of Electronic Engineering, Department of Computer Science, Laboratory of Intelligent Information Systems, Serbia,
e-mail: valentina@elfak.ni.ac.rs

George Angelos Papadopoulos

Department of Computer Science, University of Cyprus, Nicosia, Cyprus,
e-mail: george@cs.ucy.ac.cy

Svetozar Rancic
University of Nis, Faculty of Science and Mathematics, Department for Computer Science, Serbia,
e-mail: rancicsv@yahoo.com

Ioan Salomie
Technical University of Cluj-Napoca, Romania,
e-mail: Ioan.Salomie@cs.utcluj.ro

Andrzej Sikora
Research and Academic Computer Network, Warsaw, Poland,
e-mail: andrzej.sikora@nask.pl

Magdalena Szmajduch
Institute of Computer Science, Cracow University of Technology, Poland,
e-mail: magdalena.szmajduch@gmail.com

Milorad Tosic
University of Nis, Faculty of Electronic Engineering, Department of Computer Science, Laboratory of Intelligent Information Systems, Serbia,
e-mail: milorad.tosic@elfak.ni.ac.rs

Salvatore Vitabile
University of Palermo, Italy,
e-mail: salvatore.vitabile@gmail.com

Andrzej Wilczyński
Institute of Computer Science, Cracow University of Technology, Poland,
e-mail: awilczynski@pk.edu.pl

Chapter 1
Introduction

Joanna Kołodziej, Michał Marks and Ewa Niewiadomska-Szynkiewicz

Distributed computing paradigm endeavors to tie together the power of large number of resources which are distributed across a network. Each user has its own requirements and needs which is being shared among the network through proper communication channel [2].

There are three major reasons for using distributed computing paradigm. Such networks are necessary for generating the data that are required for the execution of tasks on remote resources. Second, most of the parallel applications have multiple processes that run concurrently on many nodes communicating over a high-speed interconnect. The use of high performance distributed systems for parallel applications is beneficial as compared to a single Central Processing Unit (CPU) machine for practical reasons. The ability of services distributed in a wide network is low-cost and makes the whole system scalable and adapted to achieve the desired level of the performance efficiency. Third, the reliability of the distributed system is higher than a monolithic single processor machine. A single failure of one network node in a distributed environment does not stop the whole process as compared to a single CPU resource.

Some techniques for achieving reliability on a distributed environment are check pointing and replication. Scalability, reliability, information sharing, and information exchange from remote sources are the main motivations for the users of distributed systems [1].

Distributed High Performance Computing (HPC) systems will drive progress in high energy physics, chemistry and material science, shape aeronautics, automotive and energy industries, they will also support world-wide efforts to solve emerging problems of climate change, economy regulation and medicines development. Indeed, HPC systems are viewed as strategic resources for Europe's future. Nonetheless, implementing such systems faces different new technological challenges, the major ones including reduction of energy and power consumption.

Modelling and simulation offer suitable abstractions to manage the complexity of analysing Big Data in various scientific and engineering domains. This document describes survey of the state-of-the-art

Joanna Kołodziej
Institute of Computer Science, Cracow University of Technology, Poland, e-mail: jokolodziej@pk.edu.pl

Michał Marks
Research and Academic Computer Network, Warsaw, Poland, e-mail: michal.marks@nask.pl

Ewa Niewiadomska-Szynkiewicz
Institute of Control and Computation Engineering, Warsaw University of Technology, Polande-mail: ens@ia.pw.edu.pl

of infrastructures and middleware for Big Data modelling and simulation. Furthermore, it provides the comprehensive full taxonomy of Big Data systems for life, physical, and socio-economical applications.

The attention is focused on the following topics:

- Big Data infrastructure (clouds & grids, GPU, FPGA, IoT, WSN, mobile computing),
- Big Data architectures and software platforms (for HPC computing, clouds, distributed storage),
- data analytic and middleware for e-Health, multi-agent systems, simulation, etc.,
- data representation and management (fusion, aggregation, management and visualization),
- resource management and scheduling,
- energy aware infrastructure for Big Data,
- Big Data security.

References

[1] Amir Y, Awerbuch B, Barak A, Borgstrom RS, Keren A (2000) An opportunity cost approach for job assignment in a scalable computing cluster. IEEE Transactions on Parallel and Distributed Systems 11(7):760–768, DOI 10.1109/71.877834

[2] Andrews G (2000) Foundations of Multithreaded, Parallel, and Distributed Programming. Addison-Wesley, URL https://books.google.pl/books?id=npRQAAAAMAAJ

Chapter 2
Infrastructure for Big Data processing

George Mastorakis, Rajendra Akerkar, Ciprian Dobre, Silvia Franchini, Irene Kilanioti, Joanna Kołodziej, Michał Marks, Constandinos Mavromoustakis, Ewa Niewiadomska-Szynkiewicz, George Angelos Papadopoulos, Magdalena Szmajduch and Salvatore Vitabile

2.1 Introduction to Big Data systems architecture

R. Akerkar

George Mastorakis
Technological Educational Institute of Crete, Greece, e-mail: gmastorakis@staff.teicrete.gr

Rajendra Akerkar
Western Norway Research Institute, Norway, e-mail: rak@vestforsk.no

Ciprian Dobre
University Politehnica of Bucharest, Romania, e-mail: ciprian.dobre@cs.pub.ro

Silvia Franchini
University of Palermo, Italy, e-mail: silvia.franchini@unipa.it

Irene Kilanioti
Department of Computer Science, University of Cyprus, Nicosia, Cyprus, e-mail: irenekilanioti@gmail.com

Joanna Kołodziej
Institute of Computer Science, Cracow University of Technology, Poland, e-mail: jokolodziej@pk.edu.pl

Michał Marks
Research and Academic Computer Network, Warsaw, Poland, e-mail: michal.marks@nask.pl

Constandinos Mavromoustakis
Department of Computer Science, University of Nicosia, Cyprus, e-mail: mavromoustakis.c@unic.ac.cy

Ewa Niewiadomska-Szynkiewicz
Institute of Control and Computation Engineering, Warsaw University of Technology, Poland, e-mail: ens@ia.pw.edu.pl

George Angelos Papadopoulos
Department of Computer Science, University of Cyprus, Nicosia, Cyprus, e-mail: george@cs.ucy.ac.cy

Magdalena Szmajduch
Institute of Computer Science, Cracow University of Technology, Poland, e-mail: magdalena.szmajduch@gmail.com

Salvatore Vitabile
University of Palermo, Italy, e-mail: salvatore.vitabile@gmail.com

Big data technology is a new scientific trend. Driven by data analysis in high-dimension, big data technology works out data correlations to gain insight to the inherent mechanisms. Data-driven results only rely on an unrestrained selection of system raw data and a general statistical procedure (for data processing). On the other side, procedures for conventional model-based analysis, particularly decoupling a practical interconnected system, are always based on assumptions and simplifications. Model-based results rely on identified causalities, specific parameters, sample selections, and training processes; imprecise or incomplete formulas/expressions, biased sample selections, and improper training processes will all lead to bad results. The results are often barely satisfied or even unsatisfied as the system size grows and complexity increases. Generally speaking, data-driven analysis tools, rather than model-based ones, are more suitable to complex large-scale interconnected systems with readily accessible data.

Big data (data intensive) technologies are targeting to process high-volume, high-velocity, high-variety data assets to extract intended data value and ensure high-veracity of original data and obtained information that demand cost-effective, innovative forms of data and information processing (analytics) for enhanced insight, decision making, and processes control; all of those demand (should be supported by) new data models (supporting all data states and stages during the whole data life-cycle) and new infrastructure services and tools that allows also obtaining (and processing data) from a variety of sources (including sensor networks) and delivering data in a variety of forms to different data and information consumers and devices.

The software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both. The term 'structure' points to the fact, that an architecture is an abstraction of the system described in a set of models. It typically describes the externally visible behaviour and properties of a system and its components, that is the general function of the components, the functional interaction between them by the mean of interfaces and between the system and its environment as well as the non-functional properties of the elements and the resulting system. However, an architecture typically has not only one, but several 'structures'. Different structures represent different views onto the system. These describe the system along different levels of abstraction and component aggregation, describe different aspects of the system or decompose the system and focus on subsystems.

So, an architecture is abstract in terms of the system it describes, but it is concrete in the sense of it describing a concrete system. It is designed for a specific problem context and describes system components, their interaction, functionality and properties with concrete business goals and stakeholder requirements in mind. A reference architecture abstracts away from a concrete system, describes a class of systems and can be used to design concrete architectures within this class. Put differently a reference architecture is an 'abstraction of concrete software architectures in a certain domain' and shows the essence of system architectures within the domain. A reference architecture shows which functionality is generally needed in a certain domain or the solve a certain class of problems, how this functionality is divided and how information flows between the pieces (called the reference model). It then maps this functionality onto software elements and the data flows between them. Moreover, reference architectures incorporate knowledge about a certain domain, requirements, necessary functionalities and their interaction for that domain together with architectural knowledge how to design software systems, their structures, components and internal as well as external interactions for this domain which fulfil the requirements and provide the functionalities. The goal of bundling this kind of knowledge into a reference architecture is to facilitate and guide future design of concrete system architectures in the respective domain. As a reference architecture is abstract and designed with generality in mind, it is applicable in different contexts, where the concrete requirements of each context guide the adoption into a concrete architecture. So far there is no extensive and overarching reference architecture for analytical big data systems available or proposed in literature. One can find however several concrete, smaller-scale architectures. Some of them are industrial architecture and product-oriented, that is they reduce the scope to the products from a certain company or from a group of companies. Some of them are merely technology-oriented or on a lower lever. These typically omit a functional view and mappings of technology to functions. None of them really fits into the space of an extensive, functional reference architecture. To a large extent that is by definition, as these are typically concrete architectures.

One of those product-oriented architectures is the HP Reference Architecture for MapR M5. MapR is a company selling services around their Hadoop distribution. The reference architecture described in this white paper is then more an overview of the modules incorporated in MapR's Hadoop distribution and of the deployment of this distribution on HP hardware. One could consider it a product-oriented deployment view, but it is definitely far from a functional reference architecture.

Oracle described a very high-level big data reference architecture along a processing pipeline with the steps 'acquire', 'organize', 'analyze' and 'decide' [24]. They keep it very close to traditional information architecture based on a data warehouse supplemented with unstructured data sources, distributed file systems or key value stores for data staging, MapReduce for the organization and integration of the data and additional sandboxes for experimentation. Their reference architecture is however, just a mapping of technology categories to the high-level processing steps. They provide little information about interdependencies and interaction between modules. However, they provide three architectural patterns. These can be useful, even if they are kind of trivial and mainly linked to Oracle products. These patterns are (1) mounting data from the Hadoop Distributed File System via virtual table definitions and mappings into a database management system so they can be directly queried with SQL tools, (2) using a key value stores to stage low-latency data and provide it to a streaming engine, while using Hadoop to calculate rule models and provide them to the streaming engine for processing the real-time data and raising alerts if necessary, (3) using the Hadoop file system and key value stores as staging areas from which data is processed via MapReduce either for advanced analytics applications (e.g. text analytics, data mining) or for loading the results into a data warehouse, where they can be further analyzed using in-database analytics or be accessed by further business intelligence applications (e.g. dashboards). The key principles and best practices that are focussed on in the paper are first, to integrate structured and unstructured data, traditional data warehouse systems and big data solutions, that is to use e.g. MapReduce as a pre- and post-processor for traditional, relational sources and link the results back. The second key principle they mentions, is to plan for and facilitate experimentation in a sandbox environment.

Sunil Soares of IBM describes as part of company's big data governance framework [104]. Here, the proposed reference architecture is kind of high level and while it provides a good overview of software modules or products applicable for big data settings, it no little information about interdependencies and interaction between these modules. Furthermore, the semantics are not clear. There is e.g. no explanation, what the three arrows mean. It is also not clear, what the layers mean, e.g. if there is an chronological interdependency or if the layer got ordered depending on usage relations. They are also on different levels and there are some overlaps between layers. Data warehouses and data marts e.g. are implemented using databases. That means, they are technically on different levels. A usage relation would be applicable, but this does not work for data warehouses and big data sources, as those are different systems and both an the same, functional level. An overlap exists e.g. between Hadoop Distributions and the Open Source Foundational Componens (HDFS, MapReduce, Hadoop Common, HBase). Therefore, the reference architecture can give some ideas, what functionality and software to take into account, but it is far from a functional reference architecture.

Nowadays, several organizations manage complex, multi-layered data environments that involve multiple data sources and a variety of data storage, warehouses and processes. With these current envi-

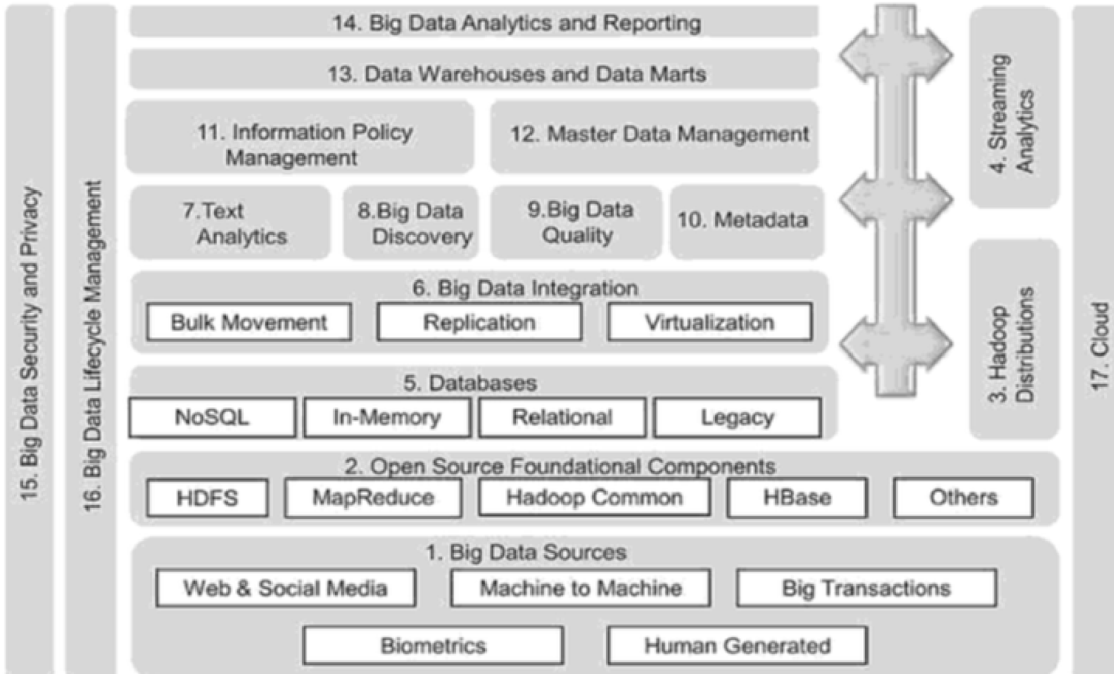

Fig. 2.1 A reference architecture for big data taken from Soares

ronments it's challenging to gradually introduce Hadoop into data architectures without interrupting business processes, data access, user data flows and other things end users have grown used to. Doing so requires architectures that can incorporate big data, while still leveraging existing investments in environment, processes and people. An analytic platform that can leverage an organization's data warehouses as well as big data sources will help that organization connect and leverage existing hierarchies, data dimensions, measures and calculations, and also connect users to data pipelines developed in Hadoop.

As customers use multiple channels and devices to engage with a business, the ability to piece together every interaction to build a granular as well as holistic picture of the customer becomes harder. Currently most analysts do not have the means to piece together structured and semi-structured information as it lies in different systems and in different formats. Even businesses that use Big Data technologies need to transform semi-structured data and load into structured databases for holistic insights. This increases the cost of storage and the time to insights. It is also practical to extend big data capabilities with a hybrid approach to data management. The hybrid data architecture that drives its analytics applications will make it easy and economical for organisations to use all of their structured and semi-structured customer data to gain accurate insights.

2.2 Clusters, grids, clouds

J. Kolodziej, M. Szmajduch

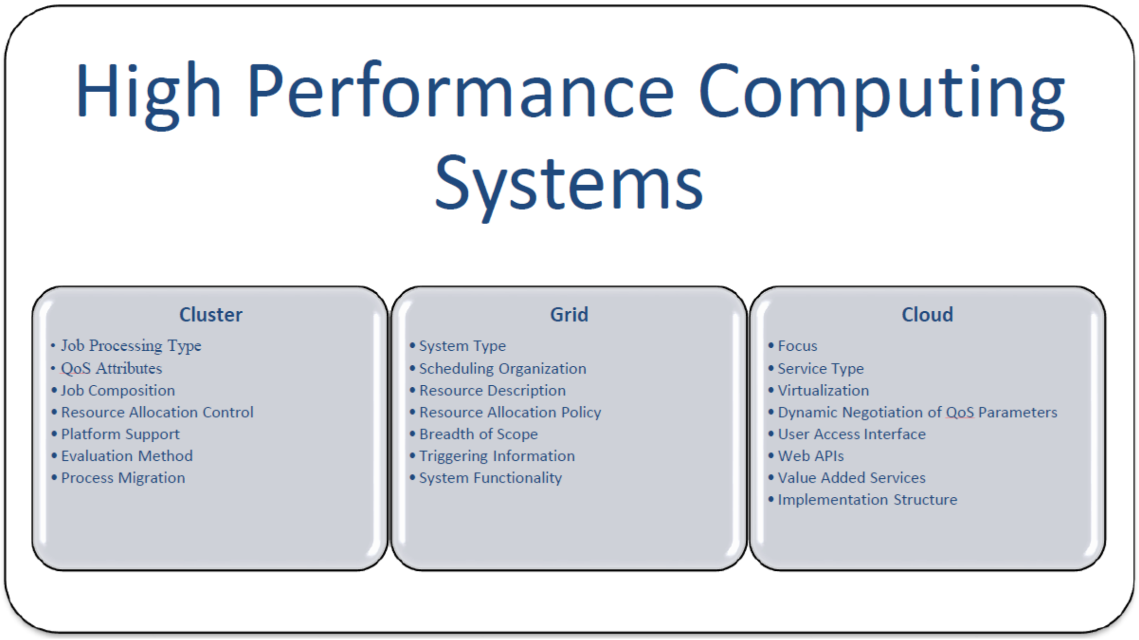

Fig. 2.2 Overview of High Performance Computing System categories and their main features

General classification of HPC systems into three main categories, namely (i) clusters, (ii) grids and (iii) clouds, is presented in Fig. 2.2.

2.2.1 Cluster Computing Systems

Cluster computing is best characterized as the integration of a number of off-the-shelf commodity computers and resources integrated through hardware, networks, and software to behave as a single computer. Initially, the terms cluster computing and high performance computing were viewed as one and the same. However, the technologies available today have redefined the term cluster computing to extend beyond parallel computing to incorporate load-balancing clusters (for example, web clusters) and high availability clusters. Clusters may also be deployed to address load balancing, parallel processing, systems management, and scalability. Today, clusters are made up of commodity computers usually restricted to a single switch or group of interconnected switches operating at Layer 2 and within a single virtual local-area network (VLAN). Each compute node (computer) may have different characteristics such as single processor or symmetric multiprocessor design, and access to various types of storage devices. The underlying network is a dedicated network made up of high-speed, low-latency switches that may be of a single switch or a hierarchy of multiple switches.

2.2.2 Grid Computing Systems

Grid computing is a term referring to the combination of computer resources from multiple administrative domains to reach a common goal. The grid can be thought of as a distributed system with non-interactive workloads that involve a large number of files. What distinguishes grid computing from conventional high performance computing systems such as cluster computing is that grids tend to be more loosely coupled, heterogeneous, and geographically dispersed. Although a grid can be dedicated to a specialized application, it is more common that a single grid will be used for a variety of different purposes. Grids are often constructed with the aid of general-purpose grid software libraries known as middleware. Grid size can vary by a considerable amount. Grids are a form of distributed computing whereby a âĂIJsuper virtual computerâĂİ is composed of many networked loosely coupled computers acting together to perform very large tasks. Furthermore, âĂIJdistributedâĂİ or âĂIJgridâĂİ computing, in general, is a special type of parallel computing that relies on complete computers (with onboard CPUs, storage, power supplies, network interfaces, etc.) connected to a network (private, public or the Internet) by a conventional network interface, such as Ethernet. This is in contrast to the traditional notion of a supercomputer, which has many parts

2.2.3 Cloud Computing Systems

Cloud computing describes a new supplement, consumption, and delivery model for IT services based on the Internet, and it typically involves over-the-Internet provision of dynamically scalable and often virtualized resources [34, 51]. It is a byproduct and consequence of the ease-of-access to remote computing sites provided by the Internet [76]. This frequently takes the form of web-based tools or applications that users can access and use through a web browser as if it were a program installed locally on their own computer [114]. The National Institute of Standards and Technology (NIST) provide a somewhat more objective and specific definition here [103]. The term "cloud" is used as a metaphor for the Internet, based on the cloud drawing used in the past to represent the telephone network [43] and later to depict the Internet in computer network diagrams as an abstraction of the underlying infrastructure it represents [100]. Typical cloud computing providers deliver common business applications online that are accessed from another Web service or software like a Web browser, while the software and data are stored on servers. Most cloud computing infrastructures consist of services delivered through common centers and built on servers. Clouds often appear as single points of access for consumers' computing needs. Commercial offerings are generally expected to meet quality of service (QoS) requirements of customers, and typically include service level agreements (SLAs) [110].

2.2.4 Computer Clusters: features and requirements

Several features are dug out for the analysis and evaluation of cluster systems. These features are selected because they are generic and have high impact on the performance and working of the systems as well; since these are the basic features which every cluster system may possess. We selected these feature to focus our study and analyze the existing cluster systems. The detail of the shortlisted features is given below.

Job Processing Type

The job processing type describes the nature of the job which is to be processed by the system. Generally, the job is classified into two categories i.e. Parallel and Sequential. In sequential processing, the job executes on one processor independently. In parallel processing, the parallel job has to be distributed to multiple processors before executing these multiple processes simultaneously. Thus, parallel job processing type speeds up processing and is often used for solving complex problems. Most of the market-based cluster RMSs supports sequential processing since it is the basic requirement. Most compute-intensive jobs submitted to clusters require parallel processing to speed up processing of complex applications, so parallel support is also essential to support in Cluster systems. For example, Enhanced MOSIX citeAmir:Y:2000 and REXEC [51] support parallel job processing type.

QoS Attributes

QoS attributes describe service requirements which consumers require the producer to provide in a service market. General attributes involved in QoS are Time, Cost, Efficiency, Reliability and Security. Customers pay for the QoS which are provided by the system. So it is very important to provide all the services to the customer. If these services are not met it could create bad image in the market and sometime user ask the producer for compensation. For example, in Cluster-On-Demand [76] and LibraSLA [114], there are several penalties if the job requirements are not fulfilled. An example of time QoS attribute is that Libra [103] guarantees, jobs accepted into the cluster finish within the users' specified deadline (time QoS attribute) and budget (cost QoS attribute). There is no market-based cluster RMS that currently supports either reliability or trust/security QoS attribute.

Job Composition

Job composition depicts the number of tasks involved in a single job prescribed by the user. A job is said to be a single-task job if it is composed of only one task whereas if the job is composed of multiple tasks then it is said to be a multiple-task job. In multiple tasks job composition the task can be dependent or independent. Independent tasks are simple to execute and can be performed in parallel to minimize the processing time. Dependent multiple tasks are cumbersome and they must be processed in a pre-defined manner to ensure that all its required dependencies are satisfied. It is essential for market-based cluster RMSs to support all three job compositions i.e. singlet task, independent multiple-task, and dependent multiple-task.

Resource Allocation Control

The Resource Allocation control represents how the resources are being managed and controlled in the clusters. All the jobs in centralized scheduling control are being administered centrally by the single resource manager. Whereas in decentralized control there are several resource manager to manage subset of resources in parallel. Decentralized manager has to communicate to allocate resources efficiently where as a centralized manager has all the knowledge so it does not require any communication. Because there is only one resource manager so there are huge chances of occurring bottlenecks in centralized resource allocation.

Evaluation Method

Evaluation methods are metrics defined to determine the effectiveness of different market-based cluster RMSs. There are two categories of these metrics System-centric and User-Centric. System-Centric methods measure performance from the system perspective and User-centric evaluation methods assess performance from the participant perspective. System centric depict the overall operational performance of the cluster whereas user centric portray the utility achieved by the participants. In order to assess the effectiveness of RMS system-centric and user-centric evaluation factors are required. System-centric factors make sure that system performance is not compromised. Whereas user-centric factors assured that desired utility of various RMS are achieved from participant perspective.

Process Migration

Transformation of a given job from one computer to another without restarting the machine, is usually defined as process migration. Process migration in homogeneous systems is usually provided in RMS. In heterogeneous systems, process migration is much more complex and requires huge amount of information about the available servers and resources, and additional conversion from source server to the destination machine.

2.2.5 Computational Grid Systems: features and requirements

Scheduling Organization

Scheduling is defined as the process of making scheduling decisions involving allocating jobs to resources over multiple administrative domains [43]. In order to meet the requirements of the user this process searches multi administrative domains to use available resources from the grid infrastructure. Scheduling organization means, the way or the mechanism with which resources are being allocated. In this survey we have considered three main organization of scheduling and those are Centralized, Decentralized and Hierarchal (Fig. 2.3).

In centralized scheduling only a single controller is responsible for performing scheduling operations and decisions. It could have many advantages like ease of managing resources, deployment can be done easily and can easily co-allocate resources. In addition to centralized scheduling organization there are Hierarchical [100] and Decentralized scheduling organizations. In hierarchical scheduling there are levels of scheduling. There are higher levels and lower level. The higher level scheduler manages large sets of resources while the lower level of controller manages small set of resources. The advantage of using hierarchical scheduling is that it incorporates scalability and fault-tolerance issues and also retains some of the advantages of the centralized scheme such as co-allocation.

There is another way of scheduling known as decentralized scheduling which naturally addresses several important issues such as fault-tolerance, scalability, site-autonomy, and multi-policy scheduling.

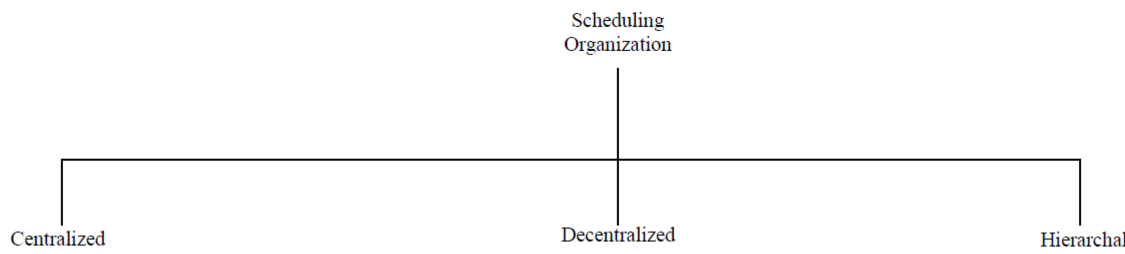

Fig. 2.3 Scheduling organization

This scheme is used for large scale network sizes but scheduling controllers needs to coordinate with each other every time for smooth scheduling. They can coordinate through resource discovery or resource trading protocols.

Resource Descriptions

Resource description is required by the under reviewed system. Resources systems For eample, Cactus Worm [32] needs the independent service responsible for resource discovery and selection based on application-supplied criteria, Using GRid Registration Protocol (GRRP) and GRid Information Protocol (GRIP).

Resource Allocation Policies

Ordering of jobs and requests when any rescheduling is required; some policy has to be defined. Those policies are termed as Scheduling policies. There are different resource utilization policies for different systems due to different administrative domains (Fig. 2.4). In a fixed approach predefined policy is implemented by the resource manager. It is further classified into two categories i.e. System oriented and Application oriented. System Oriented means to maximize the throughput of the system. While Application oriented means to optimize the specific attribute like time and cost. There are many examples of systems which uses application oriented resource allocation policies like PUNCH, WREN, CONDOR etc.

Policies which allow external agents or entities to change the scheduling policies are termed as Extensible scheduling policy. Fixed Scheduling policy can be implemented by using Ad-Hoc extensible schemes and it also provides an interface through which an agent can change the resulting scheduling policy. This process is only performed on particular resources which require more attention. Scheduling process and associated are modeled in structured extensible scheme. Existing or default scheduling procedures can be override by the agents in this scheme for example a well structured Extensible scheduling schemes are available in legion system.

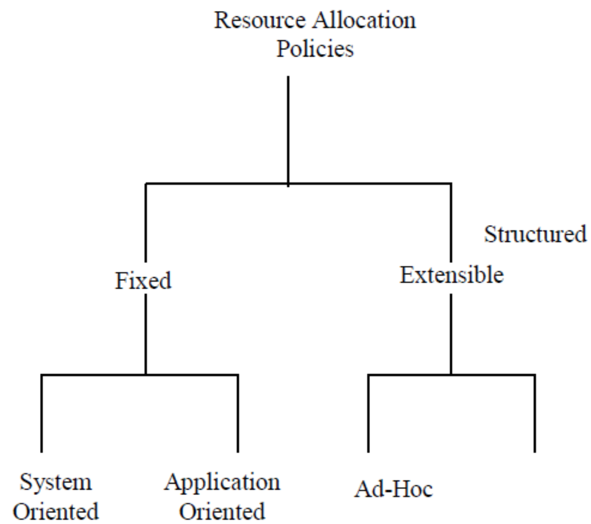

Fig. 2.4 Resource Allocation Policies

Breadth of Scope

Breadth of Scope is a self-adaptive and scalable mechanism in grids. If this is designed for specific types of applications then it could lead to poor performance of other applications not covered by the mechanism. For example, certain schedulers that treat applications as if there were no data dependencies between their tasks or ignore link bandwidth as a resource. If the system or application is designed only for specific platform or application then its Breadth of scope is low. Systems that are more scalable and self-adaptive have medium or high Breadth of scope.

2.2.6 Computational Cloud systems: features and requirements

Cloud computing has emerged along with the need to efficiently and rapidly process the huge volume of daily produced data, as well as the need of providing better quality of web services. Naturally and in order to optimally address the demands, several architectural models were introduced, each one trying to satisfy the needs of the client.

The layered architecture depicted in the table below, distinguishes four kinds of cloud-provided services: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS) and Hardware-as-a-Service (HaaS). The services can be exploited from anyone, anytime and anywhere in the world and (although this is not the case) the cloud gives the impression that has a single point of access. Furthermore, a cloud system can be separated according to the targeted application: a) The Private cloud which operates inside an organization, does not allow internet access and avoids network communication limitations b) The Public cloud that operates over the internet and provides scalable web-service solutions c) The Hybrid cloud that combines both private cloud and public cloud good qualities.

Software-as-a-Service (SaaS)
Platform-as-a-Service (PaaS)
Infrastructure-as-a-Service (IaaS)
Hardware-as-a-Service (HaaS)

A critical obstacle that âĂŞ not only âĂŞ a cloud system must overcome is the unavoidable hardware or software failures. In case of a breakdown, a production-ready data consistent machine must be hot swapped to replace the faulted one, in a way that will allow a seamless and uninterrupted continuation of the provided services. There are several examples of colossal organizations like Google (Jan 31, 2009) and Microsoft (March 13, 2008) that were unable of confronting such issues, resulting to hours of downtime, which can be translated to mind-blowing income losses. Additionally, another point that must be taken care off after a failure is the load balancing. This essentially means that the distribution of data nodes in the system must be configured in such a way that minimizes network overhead and maximizes performance.

Services

Cloud computing systems guarantee consistent services delivered through advanced data centers that are built on compute and storage virtualization technologies. It is very important while studying a system to consider the type of services a system provide to the user; since it is one of the key feature in the evaluation of the system.

Virtualization

Cloud Computing is defined as a system of virtualized computer resources. Virtualization the Cloud Computing paradigm allows workloads to be deployed and scaled-out quickly through the rapid provisioning of virtual machines or physical machines. Each cloud system can be also evaluated based upon the entities or the processes responsible for performing virtualization.

Dynamic QoS Negotiation

Real-time middleware services must guarantee predictable performance under specified load and failure conditions, and ensure graceful degradation when these conditions are violated. Quality of Service (QoS) allows for the selection and execution of Grid workflow to better fulfill customer expectations. Dynamic QoS negotiation in many systems are probably performed by either any process or by some entity dedicated for this purpose i.e. in Eucalyptus Group Managers perform the dynamic QoS operations through resource services.

Access Interface

User access interfaces provides the way to deal with the system. Access interfaces must be equipped with the relevant tools, necessary for better performance of the system. The interface could be a command line interface, query based, console based etc.

Web APIs

Web API is a development in web services (in a movement called Web 2.0) where emphasis has been moving away from SOAP based services towards Representational State Transfer (REST) based communications [38].

Value Added Services

A range of additional services beyond the standard services provided by the system that is available for a modest additional fee or sometime it's free. Usually these are offered as an attractive and cost effective alternative to promote the most popular cloud services and to draw the customers.

Implementation Structure

Implementation Structure can be defined as an informal logical âĂIJlanguageâĂİ or the package in which the system has been implemented. There could be a single language in which the system has been implemented or it could be a combination of multiple languages or packages like Google App Engine is implemented in Python and Sun Network.com (Sun Grid) can be implemented in Solaris OS, Java, C, C++, and FORTRAN.

Scalable Storage

Another important feature that cloud systems offer is the ability to store data, allowing the client to stop worrying about how it is stored or backing it up. The points of interest regarding cloud storage, is the security (which brings reliability) and the scalability. Applications that are not able to scale in a vertical way, are led to use more resources, hence increase their running costs.

Cloud offerings

There are considerable cloud offerings over the last decade mainly due to the combination with external technologies and their multi usage. These technologies existed before, however their colocation into an existing virtual hosting environment is what makes it interesting. The benefits unfortunately are coupled with challenges and a wide range of deficiencies that must be addressed in the future.

One of the disadvantages of the existing cloud platforms is the utilisation of expensive resources. We have yet to find a clever way to overcome the latency that is inherited by the virtualisation nature of the systems. In fact, the problem is even more complicated since all services must exhibit certain quality assurance characteristics, one of them is the Quality of Service. To guarantee the continuity of services, processes tend to be replicated and they cause slower performances. Dave Durkee [57] lists some factors that threaten the long-term viability of the cloud vendors. Some of them include, deploying older hardware, extra charges, long term commitments, paid range on service level agreement levels and customer service efficiency.

As the fight for cloud domination between providers continues even more aggressively over the years and they try to gain as much as possible bigger pie of the market share, users of the technology are bound to choose and to get hooked up on a specific vendor. Therefore there is no great deal of flexibility in managing their space and they depend on the vendor for handling their space, performance and security.

Interoperability

The topic of software interoperability is one of the oldest issues that can be identified in the research literature, which is needed to be accounted for in Cloud Computing so as to permit applications to be ported between clouds. Moreover, recent advances in technologies create the need for hybrid cloud and cross-cloud deployments that increases interoperability issues. The multiple cloud offerings (e.g., Windows Azure, Amazon AWS, OpenStack) are another key factor that contributes to interoperability issues. In specific, most cloud offerings are highly transparent solutions for end-users, which is completely acceptable by users up to the point there is an access issue, and/or a problem with a particular cloud offering. This creates an entirely different issue that requires for a solution, enabling a choice and transition between or across cloud providers [4]. In fact, with the wide range of different platforms, interfaces and APIs exposed by the various cloud offerings, there is a growing demand for addressing interoperability and portability. In recent work, the focus for resolving the interoperability is on a more general approach that aims in providing environments that can host specific capabilities in multiple environments [4].

Load Balancing

The continuation of a service in response to failure of one or more components is of critical importance for business actors. Load balancing is a technique the is often used to implement a failover and thus enable and ensure the continuation of the service. Monitoring the system's components allows detecting any problems and/or failures in any of the system components and load balancing (e.g., commonly applied through a load balancer component) enables reacting and re-directing load to other components. This is an inherited feature in grid-based and cloud-based computing systems, which when applied correctly offers energy conservation and resource consumption, as well as other important features such as scalability.

Expert Systems

Human experts usually use more knowledge to reason than expert systems do and often use experience in quantitative reasoning whereas expert systems cannot [109]. An expert system should facilitate the process of making decisions for human beings based on some knowledge or rules. Its special use is in the areas where the problem is inherently complex, information is updated frequently or there is need to collect knowledge over time and analyse this knowledge as basis for decisions [4]. For instance, data mining uses techniques from statistics and artificial intelligence to extract interesting patterns from large data sets. Cloud Computing provides the infrastructure that serves large data centres that offer tremendous data sets and are an excellent candidate for the application of decision making expert systems. In specific, in the domain of Cloud computing, there are efforts at W3C through an Incubator group to make a base for decision making and decisions in the context of Web [23]. Decision making expert systems are a highly suitable candidate for providing a solution and offering the means for avoiding vendor-lock in and enabling a choice amongst different cloud providers by exploiting the Web APIs offered by these cloud offerings.

Reasoning

The reasoner thereby builds up on the data gathered in the expert system and thus can create logical relationships between profile information, analysis, annotation and expert knowledge. A semantic reasoner component is a piece of software able to infer logical consequences from a set of asserted axioms. The notion of a semantic reasoner generalizes that of an inference engine, by providing a richer set of mechanisms. The inference rules are commonly specified by means of an ontology language, and often a description language. Case-based reasoning (CBR), a popular problem solving methodology in data mining, solves new problems by analyzing solutions for similar past problems. It works by matching new problems to "cases"; from a historical database and then adapting successful solutions from the past to current situations. The many advantages of CBR include rapid learning, the ability to use numerous unrestricted domains, minimal knowledge requirements, and effective presentation of knowledge [47]. Case Based Reasoning has been used successfully in several scientific and industrial application domains [99], which makes it a highly suitable candidate for reasoning amongst many possible cloud deployment offering, even hybrid cloud deployments and or adaptive load balancing, having to take into consideration multiple constraints in all these cases.

Distributed programming

In elastic environments, applications should behave in a clear and consisting manner to be supported by the resource platforms. However, this is usually not the case as applications often consist of different properties that should be identified and exploited in order to improve performance and manageability. In general, one way to achieve the above is by composing applications as a flow of individual services that can be hosted on dedicated providers. This flow defines the sequence of tasks (services) to be executed. This approach requires that the developer logically segments the application, a process that takes a considerable amount of effort. Depending on the level these services are, the process can become quite complicated âĂŞ the lower the level the more the complexity. Also, cloud systems having generally unpredictable latency and performance, it is very difficult to employ this approach for applications that have to meet hard quality (real-time) criteria.

The alternative approach defines a bottom up approach instead, meaning that it requires development of applications in a manner that they will already contain all the distribution information.

Security

The latter is of high concern. Large organisations, such as banks, financial institutions and others are very reluctant in sharing their customer's details or other sensitive information in exchange of the cloud advantages. In the wrong hands it could create a civil liability and/or criminal charges.

2.3 CPU/GPU and FPGA computing

S. Vitabile, S. Franchini

The management and analysis of large-scale, high-dimensional datasets in Big-Data applications increasingly require High Performance Computing (HPC) resources. The traditional computing model based on conventional Central Processing Units (CPUs) is unable to meet the requirements of these compute-intensive applications. New approaches to speedup High Performance Computing systems include [84], [107]:

- Multi-Core CPUs;
- Graphics Processing Units (GPUs);
- Hybrid CPUs/GPUs;
- Field Programmable Gate Arrays (FPGAs).

2.3.1 Multi-Core CPUs

In CPUs based on a single processing unit, performance increase has reached a limit since power consumption and heat dissipation hinder the increase of clock frequencies. For this reason, to further enhance microprocessor performance, CPUs based on multiple processing units or multi-core CPUs have been introduced. Multi-core CPUs integrate multiple processing cores on the same microprocessor chip allowing for parallel execution of multiple instructions and achieving higher computing throughput. This requires additional complexity of software applications since parallel computing algorithms capable of exploiting hardware parallelism in the most efficient way have to be developed in software [40], [55]. Leading commercial multi-core processors include Intel Xeon [7], AMD Opteron [2], and Tilera TILE-GX [22] processor families. The Tilera TILE-GX72 processor integrates 72 processing cores on the same chip. These multi-core CPUs are commonly used in mainstream desktop and server processors that require high-performance CPUs with intensive processing capabilities [89].

2.3.2 Graphics Processing Units (GPUs)

GPUs are highly parallel processing units capable of providing huge computing power as well as very high memory bandwidth. GPUs integrate many processing cores that can run large numbers of parallel threads and can therefore process huge data blocks in parallel. They are originally designed for real-time graphics rendering that requires very high processing capabilities to render complex, high-resolution three-dimensional scenes at interactive frame rates. Graphics applications have huge inherent parallelism. Multiple parallel threads are needed to draw multiple pixels in parallel. For this reason, modern GPUs have evolved to have an increasing number of parallel processing cores needed to run an increasingly high number of parallel threads. In the past, GPUs have been used exclusively for graphics processing. In the past decade, they have entered in the general-purpose computing traditionally supported by CPUs (General-Purpose GPU or GPGPU). Nowadays, GPUs are used not only for graphics rendering but also for parallel computing in computationally demanding general-purpose applications, such as HPC applications [94], [44]. Most advanced GPUs, such as the NVidia Tesla [19] and AMD FirePro [25] families, have a massively parallel architecture consisting of thousand of cores designed to run multiple tasks simultaneously. To exploit the computational power of GPUs for general-purpose applications, novel dedicated development platforms, such as CUDA [27] and OpenCL [72], have been designed. The main features of the leading commercial GPU accelerators, namely NVidia Tesla K80 and AMD FirePro S9170, are reported in Table 2.1.

Table 2.1 NVidia Tesla K80 and AMD FirePro S9170 GPUs Specifications

	Nvidia Tesla K80	**AMD FirePro S9170**
Processor Cores	4,992	2,816
GPU Clock	Up to 875 MHz	930 MHz
On-board Memory	24 GB	32 GB
Memory Bandwidth	480 GB/s	320 GB/s
Memory Clock	5 GHz	5 GHz
Power consumption	300 W	275 W
Single-precision Floating-Point Operations	Up to 8.74 TFLOPS	Up to 5.24 TFLOPS
Double-precision Floating-Point Operations	Up to 2.91 TFLOPS	Up to 2.62 TFLOPS

2.3.3 Hybrid CPUs/GPUs

Another common approach to achieve increased performance in computationally intensive systems is based on the hybrid CPUs/GPUs computing. This approach is widely used in supercomputers, but also in desktop computers. These hybrid systems are equipped with both multi-core CPUs and high-performing GPUs. CPU and GPU cores can be integrated on a single chip or GPUs can be used as accelerators of the CPU integrated in the same system main board. The first approach is used in the AMD Accelerated Processing Units (APUs) [1] and in the NVidia Tegra processors [14], which are Systems-on-Chip combining a multi-core CPU and more GPU cores on a single silicon die, while the second approach is used in the hybrid supercomputers, such as TianHe-1A [111] and Titan [10]. The most advanced AMD A-Series APUs are equipped with 12 compute cores (4 CPU cores and 8 AMD Radeon R7 GPU cores) working together on a single chip, while the NVidia Tegra X1 processor is based on a 8-core ARM CPU and an NVIDIA Maxwell GPU with 256 cores. The TianHe-1A (TH-1A) supercomputer, built by the Chinese National University of Defense Technology, is based on a hybrid architecture that integrates CPU and GPU cores by a proprietary high-speed interconnect network. TH-1A includes 7,168 compute nodes, each configured with two Intel Xeon X5670 CPUs and one NVIDIA Tesla M2050 GPU. It has a theoretical peak performance of 2.5 PetaFlops ($2.5 \cdot 10^{15}$ floating-point operations per second) and has been used in many applications, such as weather forecast and bio-medical research. The Titan supercomputer, developed by the Oak Ridge National Laboratory in United States, contains 18,688 nodes, with each holding a 16-core AMD Opteron 6274 processor and an NVIDIA Tesla K20 GPU accelerator, and can process more than 20 PetaFlops. One of the most important challenges in HPC systems is power consumption. Hybrid CPU/GPU systems allow for a power reduction since combining GPUs and CPUs in a single system requires less power than CPUs alone.

2.3.4 Field Programmable Gate Arrays (FPGAs)

FPGAs are programmable integrated circuits that can be configured by the costumer after manufacture. These digital devices can implement any complex digital circuit as long as the available resources on the chip are adequate [70], [98]. The most popular and widely used commercial FPGAs from Xilinx and Altera [28], [18] are high-density devices with multimillion programmable logic gates per chip. Using FPGA devices allows researchers and developers to combine hardware benefits with software flexibility. FPGAs can be programmed using electronic design automation (EDA) tools. These automated tools allow the designer to entry a high-level algorithmic description of the system written using a hardware description language, such as VHDL or Verilog, and obtain a hardware circuit implementing the desired functionality. Last-generation FPGAs enable higher performance and lower power consumption than the previous generations. They provide an increasing number of embedded Digital Signal Processing (DSP) units, RAM memory blocks, and Multi-Gigabit Transceivers capable of operating at serial bit rate over 1 Gb/s to achieve higher I/O bandwidth. The most recent FPGA families include complete Systems-on-Chip (SoC) that integrate both a general-purpose processor and programmable logic fabric on the same FPGA chip. Such SoCs include the Xilinx Zynq-UltraScale FPGA devices [28] that combine the ARM v8-based Cortex-A53 64-bit application processor with the ARM Cortex-R5 real-time processor and the configurable logic blocks of the Xilinx UltraScale architecture to create a Multi-Processor SoC (MPSoCs). Altera's Stratix 10 SoCs include an embedded hard processor system based on a quad-core 64-bit ARM Cortex-A53 [18]. FPGAs are increasingly used for the optimized hardware implementation of compute-intensive algorithms since they enable a high degree of parallelism and can achieve orders of magnitude speedups over general-purpose processors. FPGAs represent an attractive alternative to cost-prohibitive Application Specific Integrated Circuits (ASICs). Several compute-intensive algorithms in different application domains have been migrated to FPGAs. Highly parallel algorithms based on appropriate computing paradigms are needed to exploit the increasing parallel processing capabilities of FPGAs in the most efficient way. Recent research works have proposed to exploit the parallel processing capabilities of FPGAs coupled with the powerful computing paradigm based on Clifford geometric algebra to accelerate complex calculations in different application domains, such as computer graphics, robotics, and medical image processing [62], [61], [60]. The main features of the most advanced FPGA families from Xilinx and Altera, namely Xilinx Virtex Ultrascale+ and Altera Stratix 10, are listed in Table 2.2.

Table 2.2 Xilinx Virtex Ultrascale+ and Altera Stratix-10 FPGAs Specifications

	Xilinx Virtex Ultrascale+	Altera Stratix-10
Logic Cells (k)	862 – 3,763	484 – 5,510
Block RAM (Mb)	25.3 – 94.5	43 - 229
DSP Units	2,280 – 11,904	1,152 – 5, 760
Transceivers	40 - 128	24 - 144
Max Transceiver Speed (Gb/s)	32.75	17.4 - 30
I/0 Pins	416 - 832	488 – 1,640

2.3.5 GPU/FPGA Comparison

The most important advantage of FPGAs is their flexibility that allows the designer to implement a fully optimized hardware circuit for each specific application. For this reason, FPGAs can achieve high performance in many applications in spite of their low operational frequencies [35]. Conversely, GPUs run software and processing an algorithm in software takes more time. Operational frequencies in GPUs are higher than FPGAs, but a bit slower than CPUs. However, GPUs uses a large number of parallel processing cores (hundreds or thousands) so that their peak performance outperforms conventional CPUs. In GPUs, power consumption has been historically a problem, even though the latest GPU cores have reduced this disadvantage. FPGAs have higher power efficiencies compared to both CPUs and GPUs. On the other hand, GPUs provide greater bandwidth and higher data transfer rates. GPUs can execute efficiently applications that require intensive floating-point operations since they are native hardware floating-point processors. However, the latest FPGA devices show increased floating-point performance. GPUs allow for better backward compatibility. A new algorithm can be executed on older GPUs. Conversely, upgrading an algorithm on an FPGA or porting an old algorithm to a new FPGA platform is more problematic. The trend seems to be the integration of different technologies with the aim of exploiting the strengths of each type of processor and maximizing system efficiency. High Performance Computing systems are moving toward heterogeneous platforms consisting of a mix of multi-core CPUs, GPUs and FPGAs that work together. These hybrid systems can potentially lower the power consumption enabling at the same time relevant performance enhancements.

2.4 Mobile computing

C. Dobre, G. Mastorakis, C. Mavromoustakis, I. Kilanioti, G. A. Papadopoulos

Mobile communication systems have changed the way people communicate in the past two decades. The rapid growth of such systems enables devices, like smart phones and tablets to create continuously increasing demands for mobile data services. Customers of mobile communication systems electronically store, maintain and move data across the world in the cloud in a matter of seconds. Information needs to be accessible and available continuously. All these have been achieved though wireless access technologies. Wireless access technologies are about to reach its fifth generation (5G) mobile networks. Nevertheless there are lots issues that all these emerging technologies have to deal with. A key issue is how we achieve access in Big Data combining high throughput and mobility and how we manage data in cloud. In this part of the report we present the proposed architectures for 5G mobile networks, as well as data storage systems and data management in the cloud.

During the last few years there has been a phenomenal growth in wireless industry. Widespread wireless technologies, increasing variety of user friendly and multimedia enabled terminals encourages the growth of user centric networks. This leads to need for efficient network design and trigger demands for higher data rates. Mobile data traffic has been forecasted to grow more than 24 fold between 2010 and 2015, and more than 500 fold between 2010 and 2020 [48]. These trends enforce the deploy of fourth generation (4G) of mobile computing which have the ability is to support high data rate 1Gbit/s. Fifth generation (5G) focus on building an important role with higher data rate, energy efficiency and more services benefits to the upcoming generation over 4G. 5G will be smarter technology that 4G and will connect whole world with no limits.

In order to achieve all these goals industry is starting to study new technologies. For example, in the new IEEE 802.11ax task group, there is a pronounced increase in the presence of cellular operators, something not previously seen. Different technologies will grow up, consisting of combination of existing technologies and combining multiple interconnected communication standards. In 5G mobile networks users will have access on mobile cloud in seconds. By saying mobile cloud we mean that mobile applications store its data and take advantage of data processing of powerful and centralized computing platform in the cloud [56]. Plethora of mobile applications use mobile cloud services like [59] image processing (optical character recognitions programs), natural language processing, crowd computing, sharing GPS/internet data, sensor data applications, multimedia and social networking applications.

2.4.1 Network architectures for 5G

A big issue that 5G mobile networks have to deal with, is low data throughput. Nowadays mobile users, in the same area, share limited bandwidth. Future mobile networks should provide high throughput for mobile users. They will have access in the cloud resources without delays. In this section we provide network architectures that provide connectivity, mobility and high data throughput.

Software Define Networks (SDN)

The key issues Software Define Networks try to deal with are the concurrent operation of different network types (Heterogeneous Networks) and low data bandwidth, while significant role has the utilization and development, of existing network technologies. Term Heterogeneous Network (HetNet) means that network may consist of computers networks, mobile networks, mobile devices with different capabilities, deferent operating systems, hardware, protocols, etc. To determine this, different classes of base stations, macro-, pico-, and femto- base stations are cooperate.

1) New Carrier Type:

Nowadays mobile networks consist of cells. This architecture should change to enable small cells to be deployed anywhere with Internet connectivity. By reducing the size of the cell, area spectral efficiency is increased through higher frequency reuse.

Small cells enable the separation of the control plane and the user plane. The control plane provides the connectivity and mobility while the user plane provides the data transport. In such a way, the user equipment (UE) will maintain connection with two different base stations, a macro and a small cell, simultaneously. The macro cell will maintain connectivity and mobility (control plane) using lower frequency bands, while the small cell provides high throughput data transport using higher frequency bands.

According to 3rd Generation Partnership Project(3GPP), collaboration between groups of telecommunications associations, known as the Organizational Partners, cell specific reference signals are always transmitted regardless of whether there are data to transmit or not. Transmitters cannot be switched off even when it does not transmit data. Using new carrier type, where cell specific control signals, such as reference and synchronization signals, are removed, we solve this problem. In new carrier type, the role of macro cell is to provide the reference signals and information blocks, supplying connectivity and mobility, while the small cells role is to deliver data at higher spectrum efficiency, throughput. Additionally, when there is no data to transit, they can now be switched off. At this way we obtain energy saving.

2) Challenges:

In 5G mobile networks, heterogeneous networks, will be a mixture of different radio access technologies as well. Moreover Wireless Local Area Network (WLAN) technologies can offer seamless handovers to and from the cellular infrastructure, and device to device communications. To analyze this we should say that in this way we lighten the burden on cellular networks and shift load from licensed bands providing higher throughput to users. On the other hand we will have poor throughput if a high concentration of user terminals offload data simultaneously. A challenging issue in 5G are to solve problems such us high density of access points and high density of users terminals.

Another problem is Device to Device communication (D2D). Nowadays there is no licensed provision for devices to communicate directly with nearby devices. Communication between devices achieved through the base station and gateway or unlicensed outside of the cellular standard using technologies such as Bluetooth or Wireless networks in ad hoc mode.

One of the biggest problems for HetNets is inter-cell interference especially if we have unplanned deployment of small cells. Moreover, the concurrent operation of small and traditional macro cells will produce irregular shaped cell sizes and hence inter-tier interference.

Finally, efficient medium access control should be designed to solve user low throughput that create dense deployment of access points.

3) Software Defined Cellular Networks:

Software Defined Networking (SDN) were developed in parallel with software defined radio in wireless communications, industry. The concept of SDN originates from Stanford University's Open Flow system which enables abstraction of low level networking functionality into virtual services. In this way, they achieve to decouple network control plane from network data plane, by simplifying network management and facilitating easer introduction of new services or configuration changes into the network.

The SDN architecture features were established according to a standardization body of SDN, Open Networking Foundation (ONF). The network control plane decoupled from the data plane and software-based SDN controllers maintain the global view of the network. Network design and operation were simplified via open standards-based and vendor-neutral APIs. Additionally, network administrators can dynamically, through automated SDN programs to change the configuration, in order to achieve network optimization and traffic adjustment flows according to needs. Moreover, open APIs and virtualization could facilitate fast innovation by separating the network service from physical infrastructure. Although, a clear definition of SDN is still lacking and an open issue is scalability in order to support a huge number of devices and a large number of small cells.

Cloud Radio Access Networks (C-RAN)

Cloud-RAN architecture is designed in such a way that allow us to study the network core (Evolved Packet Core or EPC) with an abstraction of the physical network. The core network is presented with a simplified overview by grouping macro cells with collocated small cells. The abstraction is a flexible way to isolate both systems and allows us to study and make inventions to either of them without the risk of negative impacting the other, in order to achieve improvements.

The role of C-RAN technology is to improve radio access networks (RAN) by providing denser deployment and increasing the number of base station for user equipment (UE) to connect to. It focus on [54]:

- allowing all multi-cell operation to be handled within a single C-RAN BS,
- improving resource use in the C-RAN BS,
- allowing multiple operators to share the same infrastructure.

1) Cloud-RAN components:

In the following text we describe the elements C-RAN consists of. A centralized pool of computing resources called Base Station Pool. The Base Station Pool (BSP) provides signal processing and coordination functionality required by each area's cells. BSP is responsible for inter cell coordination too. Optical Fibers used to transmit baseband data in the RAN. Finally, user equipment connect to RAN via light weight radio units and antennae called Remote Radio Heads (RRHs). RRHs can be located in any place is needed, from macro down to femto and pico cell, in order to provide connectivity to users.

2) Evolved Packet Core (EPC):

The EPC consists of the following elements. A Home Subscriber Server (HSS) has store all related information about users and provide mobility management functions. Mobility Management Entity (MME) is responsible for mobility related signaling in the control plane handling. A Serving Gateway (SGW) connects the RAN and EPC in data plane entity. Finally, the Packet Data Network Gateway (PDNGW) connects the EPC with external networks, like internet and corporate intranets. It is connected with SGW too.

3) C-RAN Architecture:

The C-RAN architecture make use of modern internet features like Virtual Local Area Networks (VLANs) and Network Address Translator (NAT), moreover, have to deal with additional challenges like right resources allocation in order to provide connectivity to user equipment (UE) in all area locations.

A VLAN controller groups together physical cells and make virtual cells; a NAT is used to represent the virtual cells to EPC as a single macro cell. Signaling and data traffic on to the EPC will controlled by the VLAN controller. Mobility anchoring is required to provide endpoint communications from end to the UE while it transfers from cell to cell. Mobility manager manages the handover between cells, not depending if the cells are virtual. Moreover, MME is responsible for signals modifications from end to the UE, while UE moves between virtual cells in order to provide connectivity to user.

In future we expect to see new user centric designed protocols that redefine mobility with EPC function in VLAN, decoupled from RAN changes.

SDN-Based Data Offloading for 5G Mobile Networks

The proposed LTE/Wi-Fi with Software Define Network (SDN) abstraction architecture enables programmable data offloading policies according to real time network conditions, user equipment status and applications needs. Main challenge of the proposed architecture is to make real time decisions about offloading flows, services and application provision, according to user needs in the specifed time, if we take in consideration, user device information, networks availability and networks condition. For this purpose, it is important to collect and utilize information from user' devices, such as battery usage, radio conditions, relative motion and previous user history.

SDN-Based data offloading focus on redirecting dynamically the traffic towards the lower cost RAN. New standards and architectures have developed by 3GPP in order to support the simultaneously use of different cellular network such as LTE, femtocells and Wi-Fi [33].

1) Network Discovery and Selection:

An Access Network Discovery and Selection Function (ANDSF) [6] entity helps user equipment (UE) to discover 3GPP and nonâĄČ3GPP access networks, such as Wi-Fi, in addition to, provides the policies to access these networks in order to communicate data.

The ANDSF server provides the framework for network selection while an ANDSF client runs in UE. In this way the operator steers traffic between Wi-Fi and cellular networks.

On the network side criteria for the network selection are, current network condition of all available networks, user location and user device type. On the user side the criteria are user subscription level, user profile according to usage history. Moreover other factors are taken in consideration like, how information is collected and captured and other policies about network selection.

2) Programmable Data Offloading:

ANDSF client sends information to the server about user profile und user current conditions like effective radio condition, throughput over existing connection, active applications, pending traffic, and battery levels. Now, the system has perfect knowledge about users profile, users type and information about current conditions and users needs. Then, the policy controller takes service information from the deep packet inspection function and changes QoS parameters such as traffic handling priority and maximum bit rate.

In the proposed model, the control plane functionality is located in the SDN controller. Functions and rules policies derived from SDN controller and a local agent measures the condition of the radio network.

SDN Controller runs application modes such as policy charging and rules functions (PCRF) and radio resource management (RRM) and authorize local control agent with some control and measurements capabilities like to change the weight or priority of a queue, when the traffic counter exceeds the threshold.

3) Policy Derivation and Offloading Mechanism:

SDN Controller in order to derive policies takes in consideration network load information and signal threshold and other operator policies.

To analyze load information state, the operator controls, cellular and Wi-Fi, networks in a given area, when cellular network is congested the operator serves customers via Wi-Fi and the opposite. Moreover, operator takes into consideration and other factors like the condition of the radio on the cellular network. For example, if we have a cell-edge user who experiences poor radio on the cellular network and Wi-Fi network has acceptable quality and load, the operator will prefer to serve the customer from Wi-Fi network even when the cellular network is not congested. As far as the signal strength threshold concerned, it is the minimum signal strength that the operator take in consideration to steer traffic towards Wi-Fi or cellular Network according to user's services needs.

To avoid ping-pong phenomenon and steer large number of UEs once to Wi-Fi network and the other to cellular network and back again there is an algorithm that controls and regulates network balance.

Finalizing, each mobile device will be equipped with a Connection Manager (CM) that take network selection decisions following the steps below. The local agent send information to SDN Controller, SDN Controller combines this information with operator's policies and subscriber's profile according to these it derives policies and sends these policies to CM. CM combine these policies with UE information and ANDFS policies and make the network selection.

2.4.2 Big data in cloud

In this section we explain what is "Big Data" and we categorize data according to type it belongs to. We propose ways to manage big data in order to extract useful information. Finally, we present storage systems for big data in 5G mobile cloud networks and ways to have quick and secure access to data through mobile equipment.

Big data in self organized networks

A huge amount of data expected to be generated within 5G Mobile Cloud evolution. The capacity required is supposed to increase about 1000X [37] in order to cover applications' requirements and 5G mobile networks expected to be more complex. The approach of Big Data to Self Organized Networks seems to be particularly promising and appealing.

Data is distinct pieces of information, usually formatted in a special way; this information can be users' files or any other recorded information for a system's needs. In this section we study information about Self Organized Networks requirements.

1) Sources of Information Within A Mobile Network:

Information comes from various network resources and can be classified in the following categories:

- Control information related to regular short-term network operation, recorded information about network current condition as far as concern customer services like idle and connected mode mobility, radio resource control QoS etc.
- Control information related to SON functions like cell load signaling, inter cell interference etc.
- Management Information covering long-term network operation functionalities like fault, configuration, accounting, performance and security management.
- Authentication, Authorization and Accounting
- All information about customers like all problems they had etc.

It is critical for next generation networks to save as much information as possible can about network and services conditions such as customer most usual issues, problem types, severity, how it solved, how easy was to solved it and the required steps for solving it.

2) Opportunities for Applying a Big Data Approach to Mobile Networks:

Proposed Big Data approach for 5G networks provides high level classification in order to extract information for different purposes. By analyzing and leveraging all available information we will achieve

Excellence Operation to mobile networks. Based on this information the SON should be able to perform the following functions:

- Identify a problem; find out the cause and the solution of the problem without engineer invention.
- Coordinate functions and provides the best performance in network by analyzing the correlation among conflicting performance goals.
- Dynamically share network resources according to network condition.

In this way, customer service will find problems before users do and which performance of the network is possible to fault. Moreover, mobile network operators could extract data for public benefit taking in consideration security and privacy laws.

3) Network Sharing Optimization:

Network infrastructure (e.g. servers, networks and storage) sharing is essential measure to reduce the cost. In 5G mobile network Infrastructure Providers will share network resources to Participating Operators dynamically. A key technology for network sharing is Network Function Virtualization. According to this technology, infrastructure provider will configure all the network entities for each virtual network operator. One system will have lots of virtual machines; each virtual machine will support a network entity. In this way, infrastructure provider will easily move resources from one entity to other according to specifications.

4) Virtualization Service Models:

Cloud computing leverage virtualization in order to maximize computing power, reduce the cost and provide all type services.

The available services models are [119]:

Infrastructure as a Service (IaaS) is a model in which the provider provides to the customer the necessary equipment to support operations, including storage, hardware, servers and networking components.

Platform as a service (PaaS) is a model that provides hardware, operating systems, storage and network capacity over the Internet.

Software as a Service (SaaS) is a model that provides software. Applications are hosted in service provider hardware and are available to customers over a network.

Location-based Data Storage for Mobile Devices in the Cloud

Lots of mobile devices' applications are client/server applications. These applications make use of cloud resources for computation performance, data storage and data sharing. As multimedia applications requirements are increasing and storage space on mobile phones is limited the requirements for storage in clouds is increasing [85]. Another reason that this type of data should be stored in clouds is because it is close to computational resources. Data stored in clouds should replicate locally for quicker access by user. So, it is critical user location to be known. Additionally, if the system knows users future location and what part of data is possible to be needed, could replicate this data before the user try to have access on it. Furthermore, there are lots of mobile applications that store data in cloud and require to replicate part of them locally for better availability. Such applications are web applications, live traffic and sensing applications and media player applications.

1) Preliminary Results:

If we study dense populated regions we can easily find out users habits. For sparsely populated areas it is not so easy, but if we compose the results of several queries we could find this information. It is critical for future networks to know users habits because in this way an application might guess the places a user is most likely to visit and which part of data is possible to access.

As [105] refers, "Ideally, the system should make automatic decisions about what data to replicate at a given time /location".

In practice, the application collects information from users' mobile phones and builds its databases. Nowadays, most mobile devices have GPS chipsets, as well as high quality beacon databases for WiFi/cellular- based localization. Replicas are local copies of data. Applications use filters to determine which subset of data is stored in a given replica. The maximum number of bytes each replica is allowed to store depends on storage capacity. Data with common characteristics associated with geographical location, can be grouped. Replication systems have efficient data structure in order to synchronize replicas, check replicas to guarantee that replica's data match with the filter and finally to check the replica's data version.

2) Client/ Server System:

In a client server system, the server runs in the cloud and in mobile phone device runs the client. Client has two basic building blocks, a location service and a replication system. The location service provides information about user possible future locations while the replication system is responsible for data items synchronization with the cloud.

Additionally, user's mobile device runs a client called StartTrack client. This client periodically captures user current location and sends this information to StartTrack server that runs in the cloud. According this information StartTrack server makes a graph called Place Transition graph.

3) Big Data Management:

In recent years, business, communities, science and government produce an increasing amount of data. We have lots of different types of granular data. Key sources of big data are public data, private data, data exhaust (metadata), community data and self-quantifications data [65] "data revealed by individual through quantify personal actions and behaviours".

In cloud the volume of significant data is immense. By analyzing this data we could find very important information for business innovations, social trends, economic crises or political upheavals. Users can have access in all this information through their mobile devices.

Future Data Center Networking

Future mobile networks will be able to deliver high-value services to all customers with highly scalable, secure virtualized data center solutions [91]. A huge resource pool in cloud will provide virtual processing, bandwidth, storage and so on. Future data center can contain as many as 100.000 servers while the peak communication bandwidth can reach up to 100Tb/s [45].

The issues that future datacenters have to address are:

- Scalable bandwidth capacity: Huge bandwidth and an efficient interconnection architecture is needed to provide connection among servers and clients connection.
- Energy efficiency: Well designed cooling systems will save energy.
- User-oriented QoS provisioning: Quality of services will differ from user to user, even in the same user can change dynamically over time.
- Survivability/reliability: The system should be stable and able to guaranteed almost uninterrupted services.

1) Architecture for Data Center Networking:

Servers will be stacked in racks. All servers in the same rack will be connected to the top of the rack (ToR) switch and ToR switches will be interconnected through higher-layer switches.

There are four networking categories inside a data center:

- Electronic Switching Technologies: Uses electronic switches for connection inside data centers. The problem is that switches have limited number of switching ports.
- Wireless Data Center: Supplemental routing paths are provided to top of the rack (ToR) switches through 60 GHz transceivers connection. The use of this flyway system provides directly and indirectly connection to routers reducing, in this way, congestion on hot links.
- All-Optical Switching: Optical switching is divided into two categories, optical circuit switching and optical packet switching. Optical circuit switching can be used for the core witching layers. This offers super high switching capacity but cannot handle burstly DC traffic. Optical packet switching provides grained and adaptive switching capacity but it is not technologically mature.
- Hybrid Technologies: Hybrid technologies combine electronic switching based networks with optical switching based networks. Large amount of data will be transmitted through optical switching based networks.

2) Inter-DCN Communications:

Communication between DCNs, located in various locations referred as inter-DCN communication. Optical circuit switching (OCS) can used to interconnect multiple DCNs, providing large bandwidth. The only problem is that it is not easy to reconfigure OCSs quickly.

Two OCS technologies are considered for this purpose, wavelength-division multiplexing (WDM) and coherent optical-orthogonal frequency- division multiplexing (CO-OFDM) technology. WDM uses a single carrier laser optical source. A number of laser sources needs to transmit mass data, increasing the cost and energy consumption. More flexible seems to be CO-OFDM that makes conventions from optical to electronics and from electronics to optical according to requirements, reducing in this way the cost and energy consumption. Software-Defined Network technology is offered to boost DCNs bandwidth over 100 Gb/s. SDN switches separate the data plane and control plane, so network part can be abstracted from data part. In this way DCNs can be optimized according to data part demands. For intra DCS and Inter DCN connections are proposed three connection levels according to the characteristics of flow between intra DCN and inter DCN. For rack level communications electronic switching or hybrid electronic/optical switching technologies can be used. For intra DCN level communication can used optical switches. Although, the use of optical switch increases the cost, it is compensated with the device's long life span. Finally for multi DCN level, a level that downstream and upstream flows are asymmetric, WDM with a multicarrier optical source technology can be used.

3) Large-Scale Client-Supporting Technologies:

In order to support lots of users with different requirements and different priority and importance levels, virtualization technologies are proposed. These technologies focus on server's virtualization obtaining server better utilization and energy consumption efficiency. Through server consolidation the administrator can manage applications and application's data.

Another important topic that future networks have to deal with, is how to design a highly efficient algorithm to serve the required dynamically and randomly changing data flow demands. A Demand Oblivious Routing (DOR) algorithm can be scheduled to make efficient routing selections in case of sudden flow change, but it will suffer from low efficiency for specific flow matrixes. On the other hand a Demand Specific Routing (DSR) algorithm can be scheduled to have higher performance when flow demand is known, but it will also suffer when flow changes dynamically. A Hybrid Scheme that includes DOR and DSR will achieve better overall performance.

Mobile Cloud Network Security Issues

It is critical for users, mobile cloud to be trustworthy. In other words to provide reliability, integrity and confidentiality. The notion of trust means that the system will eliminate all possible risks and ensure data safety and availability. The system should also have authorization mechanisms to determine the level of access of each particular user. It is important a user to have access on its data all the time and to be sure that unauthorized users can not have access to his/her data.

A proposed framework for secure mobile cloud is MobiCloud [73]. According to this framework each mobile device is treated as a Service Node (SN). Service Nodes are mirrored to one or more Extended SemiSadow Images (ESSIs) in the cloud. Each ESSI is used to address communication and computation deficiencies of a mobile device. ESSI also provides security and privacy protection. In this way cloud advantages are provided to mobile nodes. Actually ESSI is a virtual machine wherein mobile users store their data and have full access. ESSI utilizes service provider resources.

Cloud trusted domain is detached from cloud public service and storage and can belong to two different service providers. A Trusted Authority (TA) provides certificates and security keys to mobile users. The TA will deploy an Attribute-Based Identity Management that will be linked with a unique native ID. Every user will have a unique ID that will be connected with his/her unique identification like social security number. Furthermore ID will encompass private key issuer and validation period, it will also be connected with owners device MAC address.

Data access policies are constructed using known attributes as the Public Key. The cloud public service and storage domain provide services for ESSIs and mobile devices. According to this framework users can define if their data will be stored in ESSI and the security level for specific data protection. Data will be classified in two categories, critical data and normal data. Data encryption key for critical data will be generated by the user and encryption key for normal data will be generated by the cloud storage service provider. In this way users can reassured that even their cloud providers can't have access to their critical data.

2.4.3 I. Kilanioti, G. A. Papadopoulos

Mobile Computing (MC) entails the processing and transmission of data over a medium, that does not constraint the human-medium interaction to a specific location or a fixed physical link. Fig. 2.5 depicts a general overview of the MC paradigm in its current form. It is the present decade that signifies the proliferation of MC around the world, although handheld devices have been widely used for around two decades in the form of Personal Digital Assistants (PDAs) and early smartphones. Almost ubiquitous Wi-Fi coverage and rapid extension of mobile-broadband (around 78 active subscriptions per 100 inhabitants in Europe and America) provide undisrupted connectivity for mobile devices, whereas 97% of the world's population is reported to own a cellular subscription in 2015 [8]. Mobile-specific optimizations for applications along with drastically simplified and more intuitive use of devices (e.g. with multi-touch interactions instead of physical keyboards) contribute to mobile applications becoming the premium mode of accessing the Internet, at least in the US [9]. Moreover, the MC paradigm is nowadays further combined with other predominant technology schemes leading to the paradigms of Mobile Cloud Computing [29], Mobile Edge Computing [12], Anticipatory Mobile Computing [95], etc.

Fig. 2.6 depicts a classification of MC infrastructure into three main categories: (i) Hardware, (ii) Software and (iii) Communication.

Hardware

Devices

Today's mobile devices include smartphones, wearables, carputers, tablet PCs, and e-readers. They are not considered as mere communication devices, as they are in their majority equipped with sensors that can monitor a user's location, activity and social context. Thus, they foster the collection of Big Data by allowing the recording and extension of the human senses [81]. The processor speed, memory size, and disk capacity of mobile devices are largely affected by considerations on weight, ergonomics, and heat dissipation of the latter. Moreover, despite the rapid improvements in processing power, storage capacities, graphics, and high-speed connectivity, mobile devices are restricted by the fact that they rely solely on battery power in the absence of a portable generator or other power outlet. Battery capacity is not experiencing the same exponential growth curve as other technologies such as processing power and storage. Hence, considerable advancement in mobile power technology is necessary, as power requirement has to compromise with the trend for smaller and faster devices and energy-consuming during data transfer WLAN interfaces (Bluetooth and 802.11) have become ubiquitous [106].

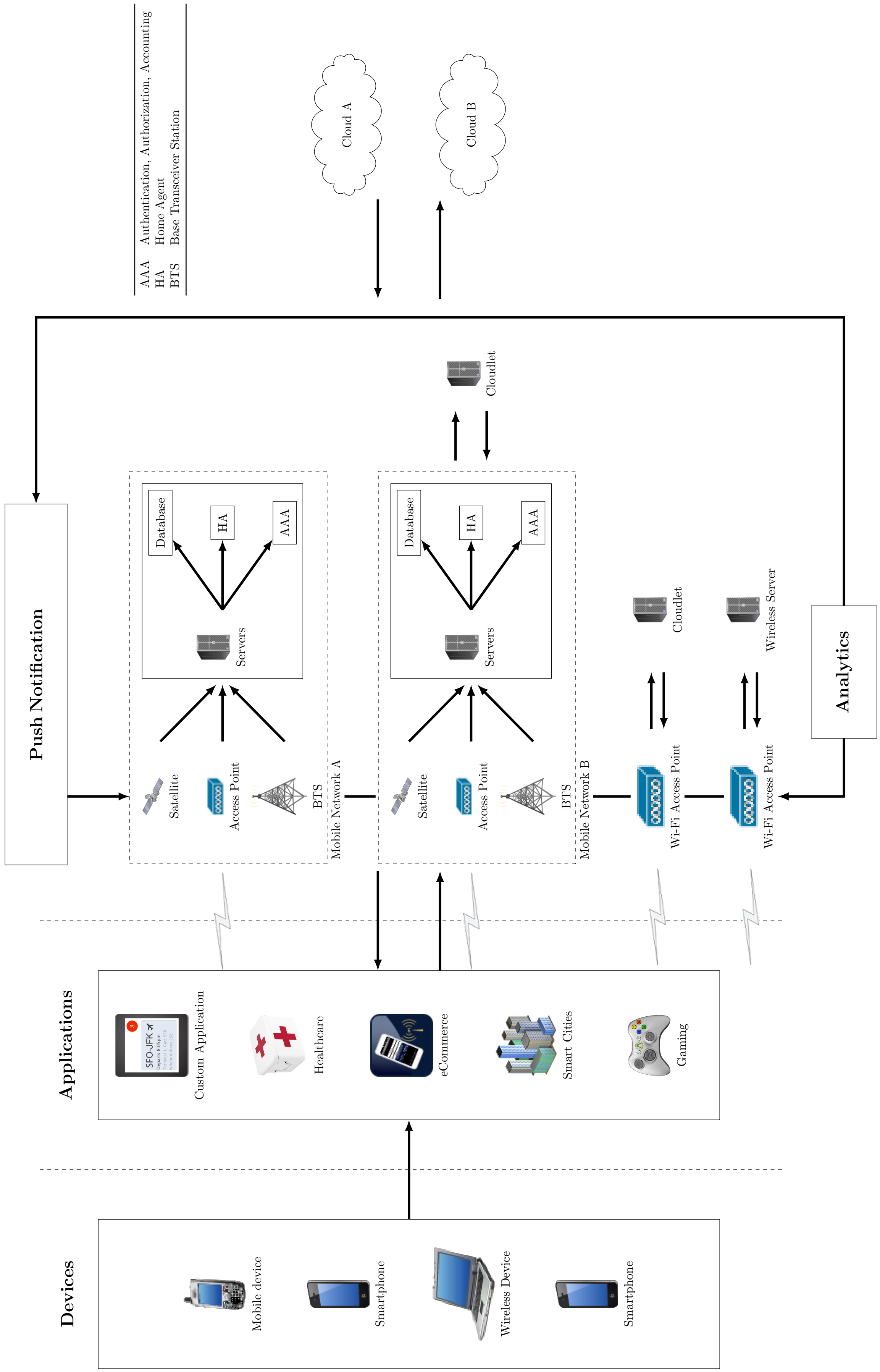

Fig. 2.5 A Mobile Computing Overview.

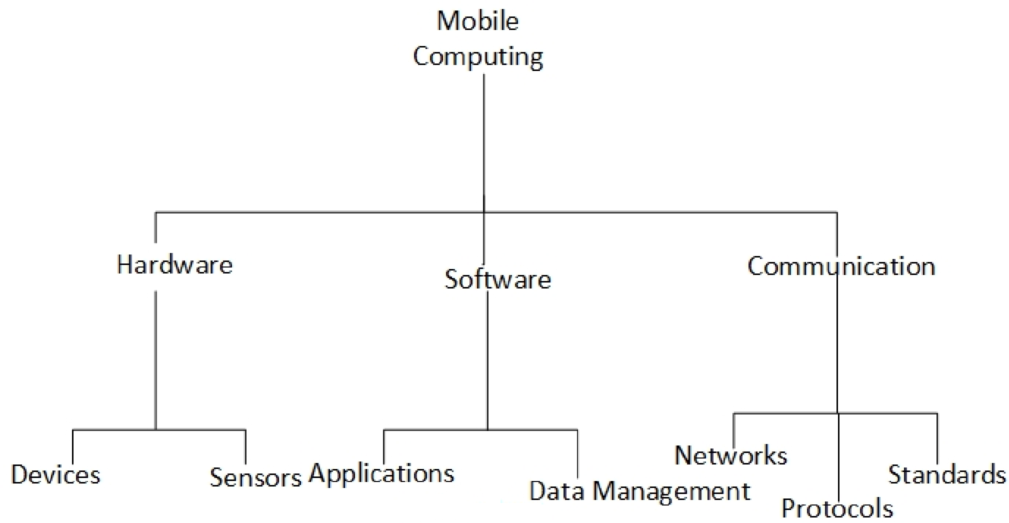

Fig. 2.6 A Taxonomy of Mobile Computing.

Sensors

Sensors embedded in mobile devices include accelerometers (measuring the acceleration of the handset, whereas the orientation is portrait or landscape), gyroscopes (providing more precise orientation information), magnetometers (used by compass applications), proximity sensors (comprised of infrared light detectors and aiming to detect nearby objects), light sensors to automatically adjust the display's brightness, GPS trackers, barometers (for the measuring of atmospheric pressure), internal or external thermometers, air humidity sensors, dedicated pedometers for counting the number of steps of the user, heart-rate sensors that detect the pulsations inside the user's finger, fingerprint sensors as an advanced security layer to unlock screen, microphones and cameras for the recording of audio and video, respectively, etc.

Software

Applications

Fig. 2.7 depicts a classification of MC Applications according to their (i) Content and (ii) Type.
Content:

- Social Applications: Mobile social networking involves the interactions between users with similar interests or objectives through their mobile devices within virtual social networks [41]. Recommendation of interesting groups based on common geo-social patterns, display of geo-tagged multimedia content associated to nearby places, as well as automatic exchange of data among mobile devices by inferring trust from social relationships are among the possible mobile social applications benefiting from real-time location and place information.

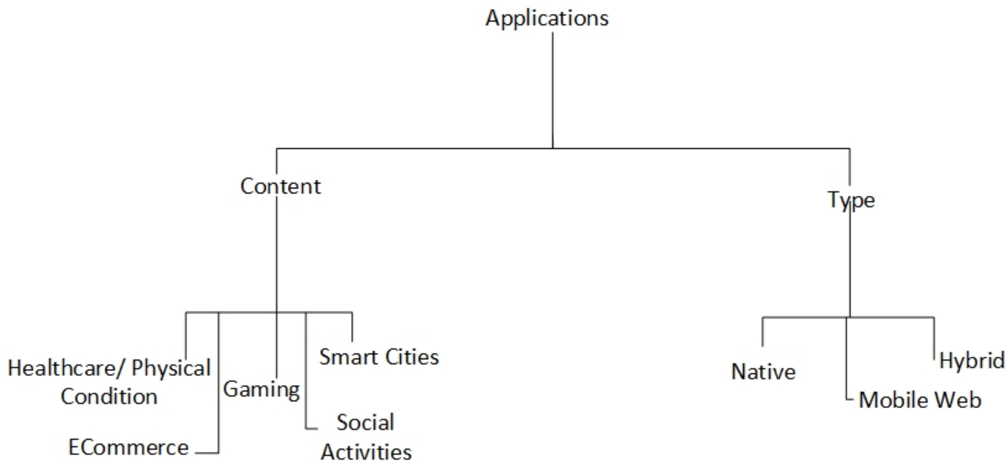

Fig. 2.7 A Taxonomy of MC Applications.

- Smart Cities/ Environment / Emergency Management: Users may opportunistically accept to host an autonomous application on their device or may be actively involved in a data collection project concerning environmental monitoring [58], traffic congestion mitigation, road surface monitoring and hazard detection [75], collection of parking space occupancy information through distributed sensing from passing-by vehicles [90], or even notification for an emergent situation in their neighborhoods [83].
- Healthcare: Mobile applications have gradually been used for medical diagnosis [67] and delivery of personalised therapies [92], [113]. The Radiation Emergency Medical Management (REMM) application, for instance, gives healthcare providers guidance on diagnosing and treating radiation injuries. Some mobile medical apps can diagnose cancer and heart rhythm abnormalities, or determine the quantity of glucose given to an insulin-dependent diabetic patient. Therapies are pre-loaded on users phones and devices react according to the sensed context of the patient. Some of applications in this category aim to help healthcare professionals improve and facilitate patient care, whereas others aim at assisting individuals in maintaining a healthy lifestyle by keeping track of their everyday behaviors (e.g., by continuously monitoring a userÃćâĆňâĎćs physical activity, social interaction and sleeping patterns, like [52] and [82]).

Type:

Native applications integrate directly with the mobile device's operating system, can manipulate its hardware and use local APIs. Mobile Web applications run directly from an online interface, and their web-based nature ensures that they are not necessarily platform-specific. Hybrids combine the interface and coding components of a web-based interface with the functionality derived from a native application, exploiting access to the device capabilities.

Data Management

Data management schemes in mobile environments include the push/pull data dissemination (servers push data and validation reports through a broadcast channel to a community of clients or clients request and validate data by sending uplink messages to server, respectively), broadcast disks (periodic broadcast of one or more disks using a broadcast channel, with disks broadcasted at different speeds and disk speed changed based on client access patterns), indexing on air (the server dynamically adjusts a broadcast hotspot, and clients perform selective tuning), as well as various client caching strategies and invalidation algorithms due to traditional cache invalidation techniques inefficiency (client disconnection, limited scalability for application servers with a large number of clients, limited caching capacity due to client memory and power consumption limitations, etc.).

Communication

Communication technologies used by MC infrastructure include GSM, CDMA, GPRS, EDGE, 3G, and 4G (LTE, LTE-Advanced) networks and a plethora of standards [5]. Lower latency in download and upload speeds is enabled within the range of 3G, 4G and upcoming 5G mobile networks, designed to provide enhanced capacity for a large number of mobile devices and expected to be main enabler of the evolution of the Internet of Things (IoT).

Wi-Fi access points offer a range that extends typically from one to a few hundred feet and depends on the specific 802.11 protocol they run, the strength of the device transmitter, as well as the existence of physical obstacles or radio interference in the surrounding area.

Satellite Internet is used to deliver one or two-way communications services (voice / data) to mobile users and is often interconnected with ground-based mobile networks. Compared to ground-based communication, all geostationary satellite communications experience high latency due to the signal having to travel to the satellite in geostationary orbit and return to Earth.

To address the increasing demand for high-speed wireless communication and novel wireless communication applications technologies including terrestrial Wi-Max and White Spaces have arisen. Wi-Max provides speeds up to 1Gbit/s to compatible devices, whereas White Spaces exploit the under-utilized licensed spectrum in bands allocated to broadcasting services. Thus, spectrum chunks functioning as buffers between digital TV channels to avoid interference and geographic areas where the spectrum is not being used for broadcasting are being exploited [71]. Innovative extension of the mobile network to rural communities and remote outposts introduces high-altitude balloons, solar-powered drones and low-orbiting satellite systems [6].

2.4.4 Mobile Cloud Computing

Ciprian Dobre

The main idea behind mobile cloud computing is to offload the execution of power-hungry tasks and to move the data pertaining to said computation from mobile devices and onto clouds with respect to the intrinsic mobility of mobile users and human behavioral patterns. Hoang et al. [56] point out the main advantages of using MCC: extending the battery lifetime by moving computation away from mobile devices, better usage of data storage capacity and of available processing power, and improving the reliability and availability of mobile applications. Furthermore, MCC inherits the benefits of cloud computing as well: dynamic provisioning, scalability, multitenancy and ease of integration. These features of MCC enable mobile application developers to create a consistent user experience regardless of the multifarious devices on the market, thus offering fairness and equality to all users [78]. The remainder of this section tackles the background work, as well as other solutions related to or similar to our own. Not only do we describe solutions that offload execution from smartphones onto traditional computing clouds, but also solutions that imply mobile devices as being explicit resources in the cloud.

In the mobile-to-cloud offloading model introduced by MCC, most challenges arise from partitioning the mobile application code into tasks that can be executed remotely and tasks that are bound to the device, based on the dependencies of each task. As such, MCC solutions can be categorized by partitioning technique into static and dynamic, and by offloading semantics into explicit and implicit. The following proposed solutions directly impact the burden of the application developers when integrating with the cloud, as each solution must be weighed in terms of the problem that needs to be solved. For example, implicit solutions might be preferred as they imply less understanding of cloud computing. However, they are more likely to lead to programming errors as developers do not fully understand how device dependencies should be handled when offloading a task (e.g. failing to send the device context along with the task). There is no perfect answer for application partitioning, although, as will be further seen, composing the above techniques usually leads to better solutions.

Zhang et al. [117] highlight the need for creating a secure elastic application model which should support partitioning into multiple autonomous components, called weblets. They propose a secure elastic runtime which presents authentication between remote and local components, authorization based on weblet access privileges in order to access sensitive data, and the existence and identification of a trusting computing base for deployment of the elastic application. Furthermore, Zhang et al. [118] extend the previous work by adding a contextual component responsible for the decision of offloading onto the cloud based on device status (CPU load, battery level), performance measures of the application for quality of experience and, of course, user preferences. By doing so, the application model supports multiple running modes: power-saving mode, high speed mode, low cost mode or offline mode.

As opposed to using explicit language constructs for remote code [117, 118], Cuervo et al. [53] use meta-programming and reflection to identify remoteable methods in applications running on their system, MAUI. The basic idea behind MAUI is to enable a programming environment in which developers partition application methods statically by annotating which methods can be offloaded for remote execution. Furthermore, each method of an application is instrumented as to determine the cost of offloading it. The MAUI runtime then decides which tasks should be remotely executed, driven by an optimization engine which is aimed at achieving the best possible energy efficiency by analysing three factors: the device's energy consumption characteristics, the application's runtime characteristics and the network characteristics. Moreover, they rule in favour of bringing cloud resources closer to mobile devices, as they prove that minimizing latency not only fixes the overwhelming energy consumption of networking, but also addresses the mobility of users. Chun et al. [49, 50] also use static partitioning and dynamic optimization for offloading, but take such systems one step further by migrating the entire thread of execution into a virtual machine clone running on the cloud. This approach proposes maintaining a clone of the entire virtual machine from the mobile device onto the cloud, by constantly synchronizing the state of the device with it's cloud counterpart, thus guaranteeing that an offload will execute correctly in terms of device context. Such an approach eases the mobile application developer's effort of assuring that all of an offloaded task's state and dependencies are correctly sent to the cloud. Moreover, given that the cloud will execute the same implementation (as if it were running locally), cloud computing knowledge is no longer a requirement thus reducing the development life cycle of cloud-enhanced mobile applications. However, maintaining such a strict synchronization between the mobile device and the cloud incurs considerable mobile network traffic which leads to high operational costs. As opposed to the previous research, Chun and Maniatis [39] rule in favor of dynamically partitioning mobile applications as to provide better user experience. The decision to execute remote computation is taken at runtime based on predictions obtained after offline analysis in the sense of optimizing what is actually needed: battery usage, CPU load, operational cost.

However, the benefits of using such solutions must be weighed with the security threats of cloud computing. The main concern of MCC is the lack of privacy in storing data on clouds as it faces multiple impediments: judicial, legislative and societal [101]. Bisong and Rahman [39] advise developers in MCC not to store any data on public clouds, but to turn their attention towards internal or hybrid clouds. In this sense, Marinelli [87] proposes the use of smartphones as the resources in the cloud and not as its mere clients. He introduces Hyrax, a platform derived from Hadoop which offers cloud computing for Android devices, allowing client applications to access data and offload execution on heterogeneous networks composed of smartphones. Interestingly enough, Hyrax offers such features in a distributed and transparent manner towards the developer. Although Hyrax is not primarily oriented at reducing energy consumption, but at fully tapping into distributed multimedia and sensor data without large network transfers, it represents a valuable endeavor due to its general architecture in which smartphones are the main resources offered by the cloud and by treating mobility as a problem of fault tolerance.

Moreover, Huerta-Canepa and Lee [74] introduce a framework that allows creating of ad-hoc cloud computing providers based on devices in the nearby vicinity which are focused on reaching a common goal in order to treat disconnections from the cloud in a more elegant fashion. As such, they rule in favor of enforcing existing cloud APIs over communities of mobile devices as to allow seamless integration with cloud infrastructures while preserving the same interface for the collocated virtual computing providers. On the other hand, Kemp et al. [78] take advantage of the existing Android partitioning of the user interface (through Activities) from the background computational code (through Services) and inject remote execution of methods by bypassing the compilation process and rewriting the interface between the aforementioned components transparently to both the developer and the user. By doing so, they ease the design and implementation of MCC applications as mobile application developers do not require any cloud computing knowledge, such as integrating with offloading APIs.

Murray et al. [93] take one further step towards opportunistic computing by introducing crowd computing which combines mobile devices and social interactions as to that advantage of the substantial aggregate bandwidth and processing power from users in opportunistic networks in order to achieve large-scale distributed computing. By deploying a task farming computing model similar to that of Marinelli [87] onto real-world traces, they place an upper-bound on the performance of opportunistically executing computational-intensive tasks and obtain a 40% level of useful parallelism.

2.5 Wireless sensor networks

M. Marks, E. Szynkiewicz

When we analyze the key properties of big data usually the first set of considered features is 3V: volume, velocity and variety. The volume indicates that a huge amount of data is gathered for processing and analysis. The velocity refers to the high speed of providing and analysing data (in many cases real time), while the variety refers to the fact the data is of highly varied structures (e.g. social media posts, images, docs, log files or web sites). A big part of the variety is the consequence of collection data from a wide range of sources such as mobiles, Machine-to-Machine (M2M), Internet of Things (IoT) or Wireless Sensor Network (WSN) devices. It is anticipated that more and more data will be generated by sensors/RFID devices such as thermometric sensors, atmospheric sensors, motion sensors or accelerometers. In the next few years, the volume of data generated by sensors and RFID devices is expected to reach the order of petabytes. Therefore WSN are responsible for generation of big data in big volume and also in a wide variety.

A WSN can generally be defined as a network of nodes that cooperatively sense, monitor and control the environment, enabling interaction between people or their computers and the surrounding environment [42]. Nowadays WSNs usually are not restricted only to sensor nodes but they include also actuator nodes, gateways and application server providing applications for clients. A large number of sensor nodes deployed randomly inside of or near the monitoring area (sensor field), form networks through self-organization. The typical structure of modern Wireless Sensor Network is presented in Figure 2.8.

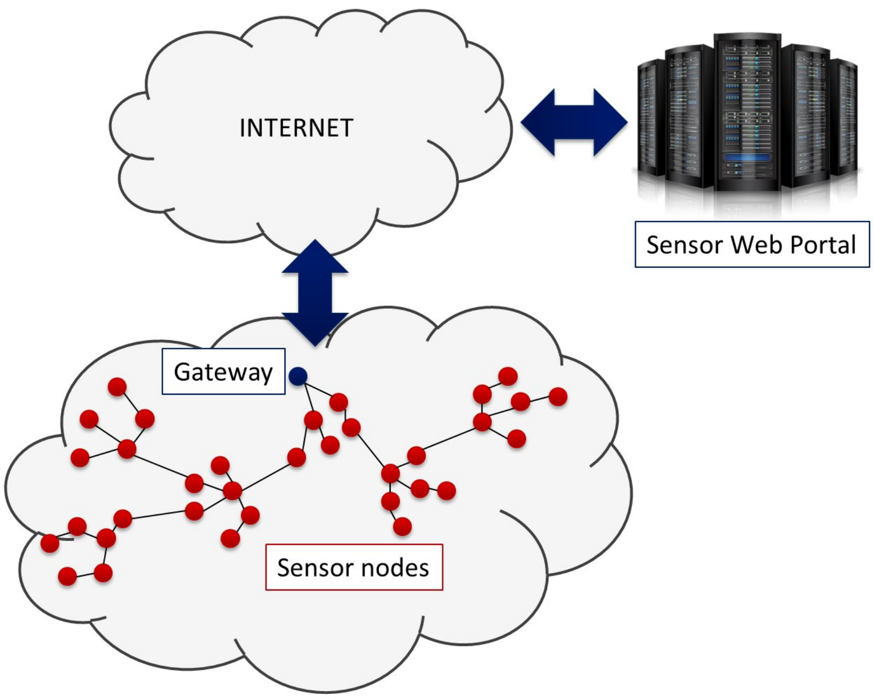

Fig. 2.8 Modern Wireless Sensor Network with WebApp

The crucial properties in context of Big Data analysis are hardware capabilities and data collection/aggregation/storing. In the subsequent sections sensor node architecture, access network technologies and data aggregation are presented.

2.5.1 Sensor node architecture

The hardware of a sensor node generally includes six parts: a microcontroller, a wireless transceiver, RAM/FLASH memory, direct communication interfaces, the power with power management module and a set of sensors, see Figure 2.9.

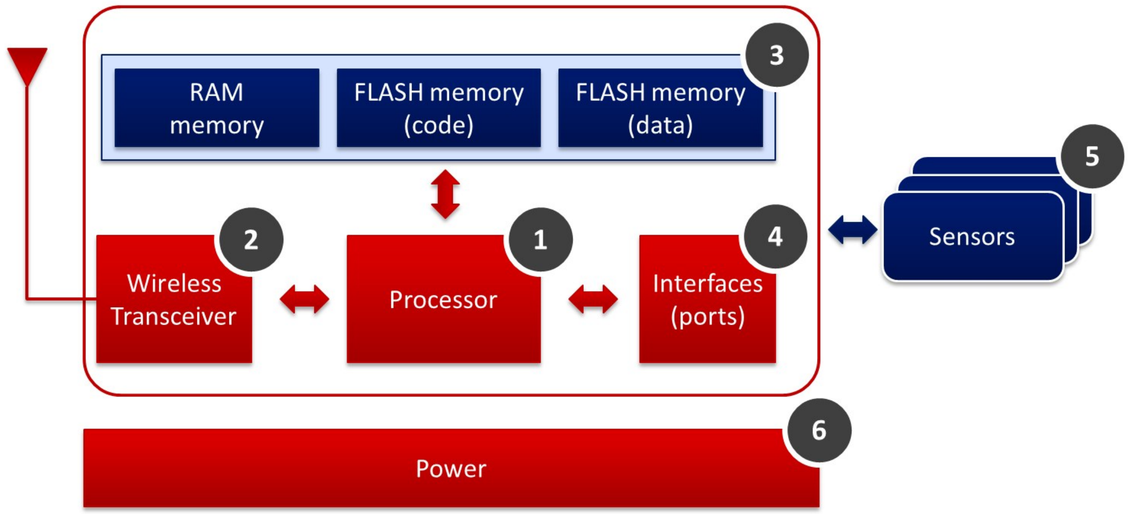

Fig. 2.9 Typical hardware used in WSN

Microprocessor

The microcontroller is the heart of WSN mote. It performs tasks, processes data and controls the functionality of other components in the sensor node. Since the beginning of WSN development mostly 8/16-bits processors were used. Nowadays there is more and more WSN platform utilizing 32-bit processors like ARM Cortex (ex. Texas Instruments CC2538 SoC [26]). What is the most important regardless of number of bits – the microprocessor must be energy-efficient to allow battery-powered motes on reasonable lifetime.

Radio Transceiver

The radio transceiver allows on data exchange between network's nodes. The most popular solution is to make use of ISM band, which gives free radio, spectrum allocation and global availability. The possible choices of wireless transmission media are radio frequency (RF), optical communication (laser) and infrared, where the last two options are used rarely as they are expensive and need a line-of-sight for communication in case of laser and have very limited capacity in case od infrared. Radio frequency-based communication is the most relevant that fits most of the WSN applications.

The most popular standard is IEEE 802.15.4 which is an energy-efficient equivalent for widely used IEEE 802.11. Usually license-free communication frequencies are used: 433, 868 MHz and 2.4 GHz. Since 2011, there is an interesting trend to apply Ultra-wideband (UWB) radio chips in WSN platforms, according to IEEE 802.15.4a specification [116]. The operational states of transceivers are transmit, receive, idle, and sleep. Current generation transceivers have built-in state machines that perform some operations automatically. What is worth noting the radio chips' energy consumption is usually higher than energy consumption caused by the rest of MCU operations.

Memory

From an energy perspective, the most relevant kinds of memory are the on-chip memory of a microcontroller and Flash memory. Flash memories are used due to their cost and storage capacity. Memory requirements are very much application dependent. Two categories of memory based on the purpose of storage are: user memory used for storing application related or personal data, and program memory used for programming the device. Even now in the year 2016, the amount of RAM and Flash memory doesn't exceed 1MB what makes programming WSN motes a demanding task not to exceed available memory.

Interfaces

Different type of interfaces allow to read data from sensors and control actuators. They are also necessary to program motes. The expendability and possibility to connect very different sensors is one of fundamental properties for WSN. Typically for direct communication USB, SPI, I2C and UART are used.

Power source

A wireless sensor node is a popular solution when it is difficult or impossible to run a mains supply to the sensor node. However, since the wireless sensor node is often placed in a hard-to-reach location, changing the battery regularly can be costly and inconvenient. An important aspect in the development of a wireless sensor node is ensuring that there is always adequate energy available to power the system. The sensor node consumes power for sensing, communicating and data processing. More energy is required for data communication than any other process. The energy cost of transmitting 1 Kb a distance of 100 metres is approximately the same as that used for the execution of 3 million instructions by a 100 million instructions per second/W processor.

Power is stored either in batteries or capacitors. Batteries, both rechargeable and non-rechargeable, are the main source of power supply for sensor nodes. They are also classified according to electrochemical material used for the electrodes such as NiCd (nickel-cadmium), NiZn (nickel-zinc), NiMH (nickel-metal hydride), and lithium-ion. Current sensors are able to renew their energy from solar sources, temperature differences, or vibration.

Sensors

Sensors are used by wireless sensor nodes to capture data from their environment are the crucial elements differentiating wireless sensor networks from other computer networks. Due to their presence sensor nodes can interact with environment. Sensors are hardware devices that produce a measurable response to a change in a physical condition like temperature or pressure or chemical like CO concentration. They measure physical data of the parameter to be monitored and have specific characteristics such as accuracy, sensitivity etc. The continual analog signal produced by the sensors is digitized by an analog-to-digital converter and sent to controllers for further processing.

Sensors are classified into three categories: passive, omnidirectional sensors; passive, narrow-beam sensors; and active sensors. Passive sensors sense the data without actually manipulating the environment by active probing. They are self powered; that is, energy is needed only to amplify their analog signal. Active sensors actively probe the environment, for example, a sonar or radar sensor, and they require continuous energy from a power source. Narrow-beam sensors have a well-defined notion of direction of measurement, similar to a camera. Omnidirectional sensors have no notion of direction involved in their measurements.

2.5.2 Access network technologies

TBD

2.5.3 Data aggregation

In the energy-constrained sensor network environments, it is unsuitable in numerous aspects of battery power, processing ability, storage capacity and communication bandwidth, for each node to transmit data to the sink node. This is because in sensor networks with high coverage, the information reported by the neighbouring nodes has some degree of redundancy, thus transmitting data separately in each node while consuming bandwidth and energy of the whole sensor network, which shortens lifetime of the network.

To avoid the above mentioned problems, data aggregation techniques have been introduced. Data aggregation is the process of integrating multiple copies of information into one copy, which is effective and able to meet user needs in middle sensor nodes.

The introduction of data aggregation benefits both from saving energy and obtaining accurate information. The energy consumed in transmitting data is much greater than that in processing data in sensor networks. Therefore, with the node's local computing and storage capacity, data aggregating operations are made to remove large quantities of redundant information, so as to minimize the amount of transmission and save energy. In the complex network environment, it is difficult to ensure the accuracy of the information obtained only by collecting few samples of data from the distributed sensor nodes. As a result, monitoring the data of the same object requires the collaborative work of multiple sensors which effectively improves the accuracy and the reliability of the information obtained.

The performance of data aggregation protocol is closely related to the network topology. It is then possible to analyze some data aggregation protocols according to star, tree, and chain network topologies.

Data aggregation technology could save energy and improve information accuracy, while sacrificing performance in other areas. On one hand, in the data transfer process, looking for aggregating nodes, data aggregation operations and waiting for the arrival of other data are likely to increase in the average latency of the network. On the other hand, compared to conventional networks, sensor networks have higher data loss rates. Data aggregation could significantly reduce data redundancy but lose more information inadvertently, which reduces the robustness of the sensor network.

2.6 Internet of things

G. Mastorakis, C. Mavromoustakis, C. Dobre, I. Kilanioti, G. A. Papadopoulos

2.6.1 Introduction

This sub-section of the report aims to present shortly the correlation between Internet of Things and the expansion of Big Data. The need for data storage and processing is now more critical than ever, as data is produced by users and appliances with tremendous rates all over the planet. In this context, a state-of-the-art content is presented on handling Big Data in the era of Internet of Things.

Internet of things (IoT) is an emerging paradigm in the science of computers and technology in general. In the last years it has invaded into our lives and is gaining ground as one of the most promising technologies. According to the European Commission, IoT involves âĂIJThings having identities and virtual personalities operating in smart spaces using intelligent interfaces to connect and communicate within social, environmental, and user contextsâĂİ [36]. IoT has been in the center of focus for quite some years now, but in the last years it has actually become reality. Indeed, attractive appliances appeared in the market, promising to make our homes smarter and our lives easier: e.g. a smart fork helping its owner monitor and track his eating habits [80] or a smart lighting system adjusting automatically to the outside light intensity, the presence of people in the house and their preferences. But not only are homes made smart, but also our accessories have changed: small wearable devices monitor our everyday movements and are capable of calculating the steps we made and the calories we consumed, or can even store details about our sleeping habits [102]. Moreover, smart watches with e-SIM technology can substitute our mobile phones and jewellery has turned into powerful gadgets (aka smart jewelry) [97]. This rapid evolution of non-mobile gadgets and mobile wearables has become the beginning of a new era!

Internet of Things has also a big impact in communities. Smart cities and industries making use of sensors will be able to control the leakage of water, or sensor equipped cars will transmit information about the traffic on the streets and probably of precarious driving behaviors. In the case of health, sensors on patients will monitor their health condition and initiate an alarm in critical cases. It is without doubt, that Internet will affect our lives at a large scale for the years to come. Imagine small, wireless devices on cars, in homes, even on clothes or food items! The tracking of merchandise will be automatically monitored, the conditions at which food supply is stored will also be written down and our clothes will be separated automatically âĂŞbased on color and textile- for the smart washing machines. But then, the production of IoT data will be enormous! So, the prevalence of Internet in our lives and especially of Internet of the Things will cause an explosion of data. According to a report from International Data Corporation (IDC) the overall created and copied data volume of the world was 1.8ZB (âĽĹ1021 B) in 2011 which increased by 9 times within five years [46]. On top of that, Internet companies handle each day huge volumes of data, e.g. Google processes hundreds of PB and generates log data of over 10 PB per month. Those areas mentioned are only some examples of Big Data and the advantages their analysis may offer. The impact on finance can be worldwide. In public sector making data widely available can reduce search and processing time. In industry, the results of R & D can improve quality. When it comes to business, Big Data allows organizations to customize products and services precisely to meet the needs of the clients. Handling data and being able to extract conclusions out of it can also be important in decision making, e.g. discover trends in pricing or sales. Moreover, it can be the root of innovation; new services and products can be created; for example casualty insurance can be based on the driving habits of the driver [86].

Identifying Big Data

Defining Big Data is not easy. According to IDC ([63]), âĂIJbig data technologies describe a new generation of technologies and architectures, designed to economically extract value from very large volumes of a wide variety of data, by enabling the high-velocity capture, discovery, and/or analysis.âĂİ From this definition, four characteristics âĂŞalso known as 4 Vs- can be extracted [69]:

(a) Volume refers to the constantly growing volume of data generated from different sources.
(b) Variety represents various data collected via sensors, smartphones and social networks. The data collected may be text, image, audio, video, in structured, unstructured or semi-structured format.
(c) Velocity indicates the acquisition speed but also and more importantly the speed at which the data should be analyzed.
(d) Value of big data means the extraction of information out of the raw data. It refers to the discovery of hidden value behind the raw data.

Why is Big Data different?

As mentioned before Big data is characterized by its big Volume and its Variety of types of data. Therefore, it is very hard for Relational Data Base Management Systems (RDBMS) to be utilized in storing and processing of big data, since Big Data have some special features:

- Data representation is critical in selecting models to store big data, because it affects the data analysis.
- Redundancy reduction and data compression: Especially when it comes to data generated from sensors, the data is highly redundant and should be filtered or compressed.
- Data cycle life management: As data is increased at unbelievable rates, it may be hard for traditional storage systems to support such an explosion of data. Moreover, the age of data is very important, in order for our conclusions to be up-to-date.
- Analytical mechanism: Masses of heterogeneous data should be processed within limited time. Therefore, hybrid architectures combining relational and non-relational databases have been proposed.
- Data confidentiality: Data transmitted and analyzed may be sensitive, e.g. credit card numbers and should be treated with care.

- Energy management: With such a massive generation of data, the storage, processing and transmission of big data, request in electric energy will rise.
- Expendability and scalability. The model handling of the data must be flexible and able to adjust to future datasets.
- Cooperation: As big data can be used in various sectors, cooperation on the analysis and exploiting of big data is significant.

It is now obvious to the reader, that such data cannot be handled using SQL- queried RDBMS and new tools (e.g. Hadoop, which is an open source distributed data processing system developed by Apache or NoSQL databases) need to be used.

Extracting the Value out of Big Data

As mentioned above, the Value of Big Data is enormous. In [86] it is estimated that if retailers make use of big data they would gain an extra profit of 60%, or that Europe's potential value to Europe's public sector administration would reach 250 billion of Euros. Indeed, retail and sales is an area where Big Data analysis produces great value and revenues. Analyzing data on in-store behavior in a supermarket can influence the layout of the store, product mix, and shelf positioning of the product. It can even affect the price of products; for example in rural areas, where rice and butter are of higher buying priority, the prices are not elastic. Urban consumers, on the contrary, tend to rate cereals and candy higher among their priorities. Another highly important area where value of big data can be extracted is Health care. A patient's electronic medical file will not only be used for his own health monitoring. It will be compared and analyzed with thousands of others, and may predict specific threats or discover patterns that emerge during the comparison. A doctor will be able to assess the possible result of a treatment, backed up by data concerning other patients with the same condition, genetic factors and lifestyle. Data analysis can prove to be not only precious but life-saving [88].

2.6.2 Big data and IoT

How much is Big Data?

With the prevalence of IoT into our lives, a huge number of 'things' can join the IoT: low-cost, low-power sensors connected wirelessly produce huge amounts of information and make use of a plethora of internet addresses, thanks to the IPv6 protocol (2^128 addresses). Seagate predicts that, by 2025, more than 40 billion devices will be web-connected and the bulk of IP traffic would be driven by non-PC devices. IP traffic due to Machine-to-machine devices is estimated to account for 64 percent, with smartphones contributing 26 percent, and tablets only for 5 percent. What is amazing is that laptop PCs will contribute only 1 percent [115]. Moreover, wearable devices seem to become part of our lives. Seagate claims that 11 million smart watches and 32 activity trackers were sold in 2015. According to predictions, sales of fitness wearables will triple to 210 million by 2020, from 70 million in 2013. As big data is already on its way, the two technologies depend on each other and should be developed intimately: The expansion of IoT can lead to a faster development of big data; the application of big data to IoT can accelerate the research advances in IoT.

Data Features in the IoT Architecture

When it comes to data acquisition and transmission, three layers comprise the architecture in IoT: (i) the sensing layer: that is the technologies responsible for data acquisition. It mainly consists of sensors (RFIDs) or wireless sensor networks (WSNs) [68] that generate small quantities of data (ii) the network layer: its main role is sending the data in the sensor network or through the internet and finally (iii) the application network which implements specific applications in IoT. Data generated by IoT technology can be in structured, unstructured or semi-structured format. Structured data are often managed by SQL, are easy to input, query, store, and analyze and can be stored in RDBMS. Examples of structured data include numbers, words, and dates. Unstructured data include text messages, location information, videos, and social media data. They do not follow a specified format, and considering that the size of this type of data continues to increase through the use of smartphones, the need to analyze and understand such data has become a challenge. Semi-structured data are data that do not follow a conventional database system. Semi-structured data may be in the form of structured data that are not organized in relational database models, such as tables. In addition, Data produced by IoT have the following features:

- Large scale data: In IoT not only is the amount of data vast, but often a need for historical data should be met (e.g. temperature in a room, or a home surveillance video). Therefore, data generated is of large scale.
- Heterogeneity: As the sensors vary in type and input, the data provided by IoT also varies in type: for example it might be numeric (temperature), text (message by a person) or image (a photo of a room).
- Time and space correlation: In applications in IoT, time and place are important and therefore, data should be also tagged with place and time information. For example the heart-rate of a person should also be combined with the time and place where it was measured (e.g. gym).
- Effective data is only a small part of the big data: often among data collected, only a small portion is usable.

Data Collection

The most common methods used for data acquisition in IoT are:

- Through RFID Sensing: Each sensor carries an RFID (Radio Frequency IDentification) tag. It allows another sensor (reader) to read, from a distance a unique product identification code (EPC standing for Electronic Product Code) associated with that tag. The reader then sends the data to a data collection point (server). The RFID tag need not have power supply, as it is powered by the sensor reader; therefore its life time is big. (Actually, Active RFID tags have been developed, which are energy-sufficient).
- Wireless Sensor Nodes (WSN) is another popular solution for the interconnection of sensors, as it meets the need for energy efficiency, scalability and reliability. A WSN consists of various nodes, each one containing sensors, processors, transceivers and power supplies. The nodes of each WSN should implement light-weight protocols in order to communicate and interact with the internet. They mainly implement the IEEE802.15.4 protocol and support peer- to- peer communication [30].
- Through Mobile equipment: Mobile equipment is a powerful tool in our hands and also a mean for sending various types of data: positioning systems acquire information about the location, images

or videos can be captured through the camera, audio can be recorded by the microphone and text can be input through the keyboard. Moreover, many applications have access to data stored in a smartphone, making it a source of data as well.

Processing the Collected Data

As Big Data derives from various sources and is often redundant, preprocessing is necessary, in order to reduce storage needs and maximize the analysis efficiency. Preprocessing can be achieved with various methods:

- Integration: Data Integration is the method of combining data from different sources and integrating them to appear as homogeneous. For example, it might consist of a virtual database showing data from different data sources, but actually containing no data, but rather âĂIJindexesâĂİ to the actual data. Integration is vital in traditional databases.
- Cleaning: Cleaning the data involves identifying inaccurate and incomplete data and modifying them in order to enhance data quality. In the case of IoT, data generated from RFIDs can be abnormal and need to be corrected. This is due to the physical design of the RFIDs and the environmental noise that may corrupt the data.
- Redundancy elimination: As it was mentioned earlier in this reportr, IoT data can be quite redundant. That is, the same data can be repeated without change and without surplus value; for example, when an RFID stays at the same place. In practice, it is useful to transmit only interesting movements and activities on the item. As data redundancy wastes storage space, leads to data inconsistency and can slow down the data analysis, it is significant to reduce the redundancy. Redundancy detection, data filtering and data compression are proposed for data redundancy elimination. The method selected depends on the data collected and the needs of the application.

Storing Big Data in File Systems

As mentioned before, the large volume of Big Data as well as the importance of extracting value from them is two basic features of big data. It is therefore important to store big data assuring that the storage is reliable and that query and analysis of the data can be made quickly and with accuracy. Various proposals have been made, such as DAS, NAS and SAN, but are beyond the scope of this review. Moreover, distributed systems have been implemented for the best storage of big data. Below, we will shortly present Storage Mechanisms which have been introduced for Big Data. Google, a giant of Internet and Big Data, has developed a File System to provide for the low level storage of Big Data. The Google File System (GFS) was designed âĂIJas a scalable distributed file system for large distributed data-intensive applicationsâĂİ, providing âĂIJfault tolerance while running on inexpensive commodity hardware,âĂİ and delivering âĂIJhigh aggregate performance to a large number of clients. Certain deficiencies in GFS drove to the design of Colossus, the successor of GFS, which is specifically built for real-time services.

Data Models with NoSQL Databases

As RDBMS fail to meet the needs for Big Data Storage, NoSQL (Not only SQL) databases gain popularity for big Data. They can handle large volumes of data, support simple and flexible non-relational data models and offer high availability. The most basic models of NoSQL databases and a brief presentation of each are presented below [66]:

- Key-value databases: These NoSQL databases are based on a simple data model based on key-value pairs, which reminds of a dictionary. The key is the unique identifier of the value and allows us to store and retrieve the value. This method provides a schema- free data model, which is suitable for distributed data, but cannot represent relations. Dynamo- developed by Amazon- and Voldemort- used by LinkedIn- are the most important representatives of this category.
- Column-family databases: This category of databases is inspired by Google Bigtable, in which the data is stored in a column oriented way. The basic concept in column-family databases is that the dataset consists of several rows, each of which is addressed by a unique row key, also known as a primary key. Each row consists of a set of column families, but different rows can have different column families. Cassandra, HBase and HyperTable also belong to Column-family databases.
- Document stores use keys to locate documents inside the data store. Most document stores represent documents using JSON (JavaScript Object Notation) or some format derived from it. Document stores are suitable for applications in which the input data can be represented in a document format. MongoDB, SimpleDB and CouchDB are popular examples of Document stores.
- Graph databases: Graph databases use graphs as their data model and are therefore based on graph theory. Graph databases are specialized in handling highly interconnected data and therefore are very efficient in traversing relationships between different entities. They are suitable in scenarios such as social networking applications or pattern recognition.

Querying

One important issue that should be taken into account about big data is the technology used to query the data stored in the databases. One of the most popular tools, MapReduce, was first developed by Google and offers distributed data processing on a cluster of computers. However, due to the huge impact of SQL on querying, it has now also been adopted in the NoSQL world. Some of the prominent NoSQL data stores like MongoDB offer a SQL-like query language or similar variants such as CQL offered by Cassandra and SparQL. In addition, many products offer API support for multiple languages. A REST-based API has been very popular in the world of Web-based applications because of its simplicity. Consequently, in the NoSQL world, a REST-based interface is provided by most solutions, either directly or indirectly through third-party APIs.

Analyzing Big Data Methods

The analysis of Big Data is the last stage in the chain of big value. It aims to extract value out of the data in various fields and is quite complex and demanding. Some of the methods applied for analyzing Big Data are discussed in the following:

- Cluster Analysis is a statistical method for classifying objects. It divides objects them into categories (clusters) according to common features; objects in the same category have high homogeneity, while different clusters will be very different from each other. Cluster analysis is an unsupervised study method without training data.
- Regression Analysis is a mathematical tool for revealing correlations between one variable and other variables. It is used for forecasting or prediction and is valuable in data mining.
- A/B Testing is a technology for determining how to improve target variables by comparing the tested group. Big data will require a large number of tests to be executed and analyzed. Big data enables large amount of tests to be executed, ensuring, this way, that groups are statistically significant to detect differences between the control and treatment groups.
- Statistical Analysis can use probability to make conclusions about data relations between variables (For example, using the âĂIJnull hypothesisâĂİ). Statistics can also reduce the likelihood of Type I errors (âĂIJfalse positivesâĂİ) and Type II errors (âĂIJfalse negativesâĂİ).
- Data mining algorithms are algorithms which extract patterns from large datasets by combining methods from statistics and machine learning with database management. These algorithms include association rule learning, cluster analysis, classification, and regression; the most popular algorithms are C4.5, k-means, SVM, Apriori, EM, Naive Bayes, and Cart.

Analyzing Big Data Tools

Furthermore, specific tools have been developed for the data mining and analysis of Big Data, the most popular of which are summarized below:

- R is an open-source programming language and software environment for statistical computing and visualization. It is very popular for statistical software development and data analysis and several data giants, like Oracle have developed products to support R.
- Rapid-I Rapidminer is open source software suitable for data mining, machine learning, and predictive analysis. The data mining flow is represented in XML and displayed through a graphic user interface (GUI). The source code is written in Java and it can support the R language.
- KNIME is another open-source platform used for rich data integration, data processing, data analysis, and data mining. KNIME was written in Java and, based on Eclipse, can extend its functionality with additional plugins. Moreover it is very user- friendly.
- WEKA / Pentaho is yet another open-source machine learning and datamining software written in Java. Weka is very popular for data processing, feature selection, classification, regression, clustering, association rule, and visualization. Pentaho is one of the most popular open-source BI (Business Intelligence) software. Weka and Pentaho can cooperate very intimately.

2.6.3 IoT Architecture: layers overview

Internet of Things (IoT) is a global infrastructure that interconnects things based on interoperable information and communication technologies, and through identification, data capture, processing and communication capabilities enables advanced services [15]. Things are objects of the physical world (physical things, such as devices, vehicles, buildings, living or inanimate objects augmented with sensors) or the information world (virtual things), capable of being identified and integrated into communication networks. Fig 2.10 depicts an overview of IoT. It is estimated that the number of Internet-connected devices has surpassed the human population in 2010 and that there will be about 50 million devices by 2020 [20], thus, the still undergoing significant innovation IoT is expected to generate massive amounts of data from diverse locations, that will need to be collected, indexed, stored and analysed.

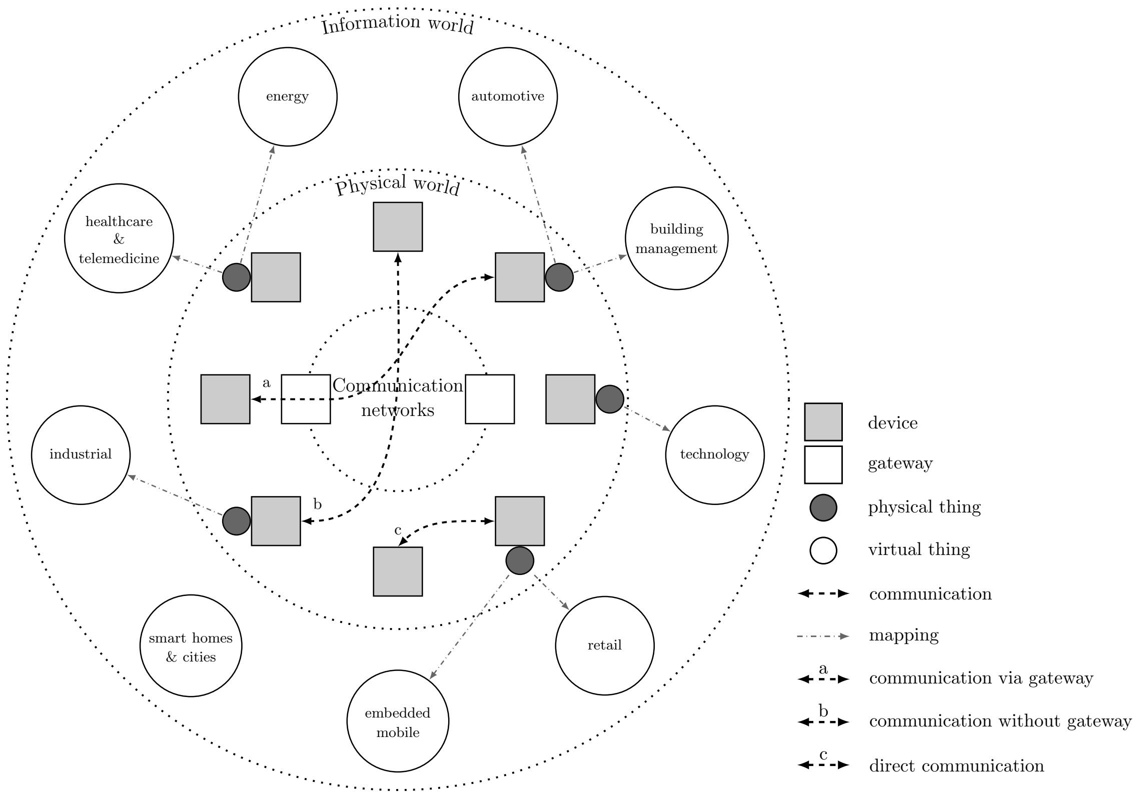

Fig. 2.10 An Overview of IoT.

Fig. 2.11 depicts a classification of IoT infrastructure into the following main categories: (i) Applications, (ii) Service Support, (iii) Network and (iv) Devices, based on the IoT reference model [15].

Application Layer

- *Industrial Applications*: Maintenenance, service, optimization of distributed plant operations is achieved through several distributed control points, so that risk is reduced and the reliability of massive industrial systems is improved [96].
- *Automotive Applications*: Automotive applications capture data from sensors embedded in the road that cooperate with car-based sensors. They aim at weather adaptive lighting in street lights, monitoring of parking spaces availability, promotion of hands-free driving, as well as accident avoidance through warning messages and diversions according to climate conditions and traffic congestion.

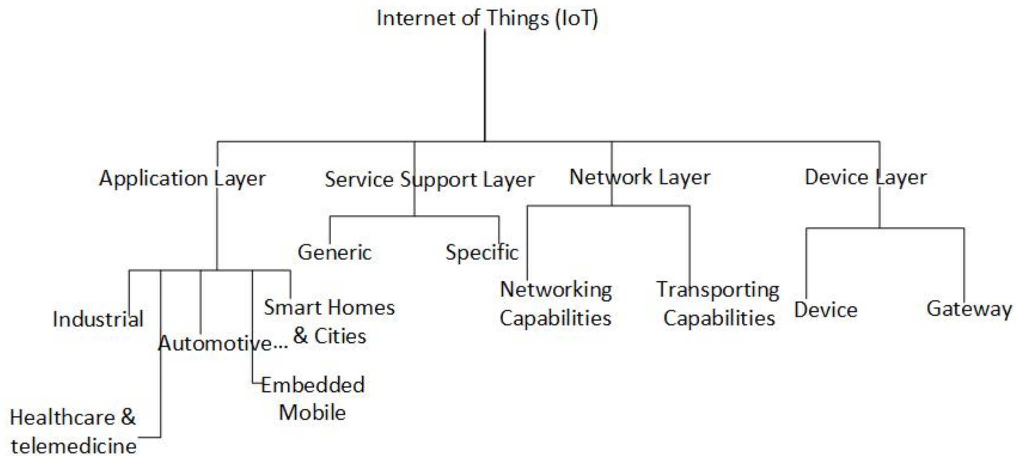

Fig. 2.11 A Taxonomy of IoT.

Applications can promote massive vehicle data recording (stolen vehicle recovery, automatic crash notification, etc.) [11].

- *Retail Applications*: Retail applications include, among many others, the monitoring of storage conditions along the supply chain, the automation of restocking process, as well as advising according to customer habits and preferences.
- *Healthcare & Telemedicine Applications*: Physical condition monitoring for patients and the elderly, control of conditions inside freezers storing vaccines, medicines and organic elements, as well as more convenient access for people in remote locations with usage of telemedicine stations [77].
- *Building Management Applications*: Video surveillance, monitoring of energy usage and building security, optimization of space in conference rooms and workdesks [11].
- *Energy Applications*: Applications that utilize assets, optimize processes and reduce risks in the energy supply chain. Energy consumption monitoring and management ([21], [112]), monitoring and optimization of performance in solar energy plants [3].
- *Smart homes & Cities Applications*: Monitoring of vibrations and material conditions in buildings, bridges and historical monuments, urban noise monitoring, measuring of eletromagnetic fields, monitoring of vehicles and pedestrian numbers to optimize driving and walking routes, waste management [64].
- *Embedded Mobile Applications*: Applications for recommendation of interesting groups based on common geo-social patterns, infotainment, and automatic exchange of data among mobile devices by inferring trust from social relationships. Applications that continuously monitor the user's physical activity, social interactions and sleeping pattern, and suggest a healthier lifestyle.
- *Technology Applications*: Hardware manufacture, among many others, is improved by applications measuring peformance and predicting maintenance needs of the hardware production chain.

Service support Layer

- *Generic Support Capabilities*: They refer to uniform capabilities such as data capture and storage, and they provide uniform handling for underlying resources, for instance remote device management (remote software upgrades, diagnostics and recovery). They are capabilities that offer access to IoT resources, tools for modeling contextual information and information related to physical things, etc.
- *Specific Support Capabilities*: They are tailored to various IoT applications. They may include Location Based Service (LBS) or Geographic Information Systems (GIS) capabilities, support for services related to sensor originating data or actuation services, and services related to different tags such as Radio Frequency Identification (RFID).

Network Layer

- *Networking Capabilities*: They aim at controlling the network connectivity, with access and transport resource control functions, mobility management, as well as authentication, authorization and accounting (AAA). The volume, variety and velocity of data generated within IoT furthermore require content-aware networks that can recognize different services and applications, and can consequently adjust their configuration to optimize performance (for instance, data replication traffic needs low latency, while video and images need high bandwidth and better quality of service), network destinations close to the source (as hierarchical data center network architectures add latency with traffic traversing each network tier and prevent real-time decision-making based on current network conditions), unification of access for diverse networks (so that management overhead is alleviated), and other features.
- *Transport Capabilities*: Capabilities that foster the transmission of IoT service- and application-specific data, as well as the transmission of IoT-associated control information. Multiple potential protocols for communication between the devices and the core of the IoT include HTTP/HTTPS (and RESTful approaches on those), Message Queuing Telemetry Transport protocol (MQTT) [13], a publish-subscribe architecture on top of TCP/IP which allows bi-directional communication between a thing and a MQTT broker in lossy networks, and Constrained Application Protocol (CoAP) [16], designed to provide a RESTful application protocol modeled on HTTP semantics over UDP.

Device Layer

- *Devices*: Devices can be
  - (i) internet-enabled personal items, such as smartphones, tablet PCs and PCs,
  - (ii) ubiquitous internet-enabled smart objects that sense and communicate without human interaction, such as wearable sensors, embedded sensors, and actuators, as well as
  - (iii) items without IP address, using proprietary or standardized RFID, active RFID, Mesh Sensor Networks, or Real Time Location Systems for communication.

  Devices communicate through the communication network via a gateway (Fig 2.10, case a), without a gateway (case b) or directly, that is without using the communication network (case c). Devices can also communicate with other devices in combined ways, for instance using direct communication through a local network (such as an ad-hoc network, case c) and then communicating through the communication network via a local network gateway (case a).

- *Gateways*: Gateways mitigate communication problems for supporting inter-device-layer connections of devices using different kinds of wired or wireless technologies [31]. They also mediate when communications involving both the device layer and network layer use different protocols. Diverse technologies at the network layer include the public switched telephone network (PSTN), 2G/ 3G/ 4G networks, General Packet Radio Service (GPRS), advanced Long-Term Evolution Networks (LTE-A), Ethernet or Digital Subscriber Lines (DSL). Communication protocols include RFID, Near-Field Communication (NFC) [108], Ultra-wideband (UWB) [79], Wi-Fi, Wi-Fi Direct, Bluetooth, Bluetooth Low-Energy (BLE), Zigbee and IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) [17].

References

[1] (????) AMD A-Series Desktop APUs. AMD Corporation, http://www.amd.com/en-us/products/processors/desktop/a-series-apu
[2] (????) AMD Opteron 6000 Series. AMD Corporation, http://www.amd.com/en-us/products/server/opteron/6000
[3] (????) Energy Management Framework. IETF Internet Draft. http://goo.gl/TJXXbn Accessed: 2016-03-15
[4] (????) EU FP7 Project, PaaSage: Model-based Cloud Platform Upperware. http://www.paasage.eu/ Accessed: 2016-03-18
[5] (????) European telecommunications standards institute. mobile communication standards. http://goo.gl/2CyeeA Accessed: 2016-03-15
[6] (????) Ieee spectrum, september 2013. five ways to bring broadband to the backwoods. http://goo.gl/k1aAco Accessed: 2016-03-15
[7] (????) Intel Xeon Processor E7 Product Families. Intel Corporation, http://www.intel.com/content/www/us/en/processors/xeon/xeon-e7-v3-family-brief.html
[8] (????) International Telecommunication Union. 2015. ict facts and figures. the world in 2015. https://goo.gl/cqOSXA Accessed: 2016-03-15
[9] (????) Internet Society. Global Internet Report 2015. Mobile Evolution and Development of the Internet. http://goo.gl/wUMJ8y Accessed: 2016-03-15
[10] (????) Introducing Titan-Advancing the Era of Accelerated Computing. Oak Ridge National Laboratory, https://www.olcf.ornl.gov/titan/
[11] (????) Management of Networks with Constrained Devices: Use Cases. IETF Internet Draft. https://goo.gl/cT5pXr Accessed: 2016-03-15
[12] (????) Mobile-edge computing âĂŞ introductory technical white paper. https://goo.gl/ybrCnq Accessed: 2016-03-15
[13] (????) MQTT 3.1.1 specification. http://goo.gl/V3SqFw Accessed: 2016-03-15
[14] (????) NVidia Tegra X1 Processor. NVidia Corporation, http://www.nvidia.com/object/tegra-x1-processor.html
[15] (????) Recommendation ITU-T Y.2060. Overview of the Internet of Things. ITU. http://goo.gl/Yl0sii Accessed: 2016-03-15

[16] (????) RFC describing Constrained Application Protocol (CoAP). https://goo.gl/pJ2Ngg Accessed: 2016-03-15
[17] (????) RFC describing IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs). https://goo.gl/gBElJe Accessed: 2016-03-15
[18] (????) Stratix 10 FPGA and SoC. Altera Corporation, https://www.altera.com/products/fpga/stratix-series/stratix-10/support.html
[19] (????) Tesla Product Family. NVidia Corporation, http://www.nvidia.com/object/tesla_product_literature.html
[20] (????) The Internet of Things: How the next evolution of the Internet is changing everything. CISCO, San Jose, CA, USA, White Paper, 2011. http://goo.gl/ugKAoN Accessed: 2016-03-15
[21] (????) The Smart Grid: An Introduction. US Department of Energy. http://goo.gl/jTNgf Accessed: 2016-03-15
[22] (????) TILE-GX72 Processor Product Brief. Tilera Corporation, http://www.tilera.com/files/drim__TILE-Gx8072_PB041-04_WEB_7666.pdf
[23] (????) W3C Decisions and Decision-Making Incubator Group. http://www.w3.org/2005/Incubator/decision/#Deliverables Accessed: 2016-03-18
[24] (2012) Oracle Information Architecture: An ArchitectâĂŹs Guide to Big Data. Oracle, rev. d edn
[25] (2015) AMD FirePro S9170 Server GPU. AMD Corporation, http://www.amd.com/Documents/AMD-FirePro-S9170-Datasheet.pdf
[26] (2015) CC2538 Powerful Wireless Microcontroller System-On-Chip for 2.4-GHz IEEE 802.15.4,6LoWPAN, and ZigBee Applications. Texas Instruments, rev. d edn
[27] (2015) CUDA C Programming Guide. nVidia Corporation, version 7.5 edn, http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf
[28] (2016) UltraScale Architecture and Product Overview. Xilinx Corporation, http://www.xilinx.com/support/documentation/data_sheets/ds890-ultrascale-overview.pdf
[29] Abolfazli S, Sanaei Z, Ahmed E, Gani A, Buyya R (2014) Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges. IEEE Communications Surveys and Tutorials 16(1):337–368, DOI 10.1109/SURV.2013.070813.00285, URL http://dx.doi.org/10.1109/SURV.2013.070813.00285
[30] Aggarwal CC, Ashish N, Sheth A (2013) Managing and Mining Sensor Data, Springer US, Boston, MA, chap The Internet of Things: A Survey from the Data-Centric Perspective, pp 383–428. DOI 10.1007/978-1-4614-6309-2_12, URL http://dx.doi.org/10.1007/978-1-4614-6309-2_12
[31] Al-Fuqaha Al, Guizani M, Mohammadi M, Aledhari M, Ayyash M (2015) Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys and Tutorials 17(4):2347–2376, DOI 10.1109/COMST.2015.2444095, URL http://dx.doi.org/10.1109/COMST.2015.2444095
[32] Allen G, Angulo D, Foster I, Lanfermann G, Liu C, Radke T, Seidel E, Shalf J (2001) The cactus worm: Experiments with dynamic resource discovery and allocation in a grid environment. International Journal of High Performance Computing Applications 15(4):345–358
[33] Amani M, Mahmoodi T, Tatipamula M, Aghvami H (2014) SdnâĄČ based data offloading for sdnâĄČ based data offloading for 5g mobile networks g mobile networks. ZTECOMMUNICATIONS p 34
[34] Amir Y, Awerbuch B, Barak A, Borgstrom RS, Keren A (2000) An opportunity cost approach for job assignment in a scalable computing cluster. IEEE Transactions on Parallel and Distributed Systems 11(7):760–768, DOI 10.1109/71.877834

[35] Asano S, Maruyama T, Yamaguchi Y (2009) Performance comparison of fpga, gpu and cpu in image processing. In: Field Programmable Logic and Applications (FPL 2009), 2009. Proceedings. International Conference on, pp 126–131

[36] Atzori L, Iera A, Morabito G (2010) The internet of things: A survey. Computer Networks 54(15):2787 – 2805, DOI http://dx.doi.org/10.1016/j.comnet.2010.05.010, URL http://www.sciencedirect.com/science/article/pii/S1389128610001568

[37] Baldo N, Giupponi L, Mangues-Bafalluy J (2014) Big data empowered self organized networks. In: European Wireless 2014; 20th European Wireless Conference; Proceedings of, pp 1–8

[38] Benslimane D, Dustdar S, Sheth A (2008) Services mashups: The new generation of web applications. IEEE Internet Computing 12(5):13–15, DOI 10.1109/MIC.2008.110

[39] Bisong A, Rahman M, et al (2011) An overview of the security concerns in enterprise cloud computing. arXiv preprint arXiv:11015613

[40] Blake G, Dreslinski RG, Mudge T (2009) A survey of multi-core processors. IEEE Signal Processing Magazine 26(6):26–37

[41] Boyd D, Ellison NB (2007) Social network sites: Definition, history, and scholarship. J Computer-Mediated Communication 13(1):210–230, DOI 10.1111/j.1083-6101.2007.00393.x, URL http://dx.doi.org/10.1111/j.1083-6101.2007.00393.x

[42] BrÃűring A, Echterhoff J, Jirka S, Simonis I, Everding T, Stasch C, Liang S, Lemmens R (2011) New generation sensor web enablement. Sensors 11(3):2652, DOI 10.3390/s110302652, URL http://www.mdpi.com/1424-8220/11/3/2652

[43] Casavant TL, Kuhl JG (1988) A taxonomy of scheduling in general-purpose distributed computing systems. IEEE Transactions on Software Engineering 14(2):141–154, DOI 10.1109/32.4634

[44] Chen G, Li G, Pei S, Wu B (2009) High performance computing via a gpu. In: Information Science and Engineering (ICISE), 2009. Proceedings. 1st International Conference on, pp 238–241

[45] Chen M, Jin H, Wen Y, Leung VCM (2013) Enabling technologies for future data center networking: a primer. IEEE Network 27(4):8–15, DOI 10.1109/MNET.2013.6574659

[46] Chen M, Mao S, Liu Y (2014) Big data: A survey. Mobile Networks and Applications 19(2):171–209, DOI 10.1007/s11036-013-0489-0, URL http://dx.doi.org/10.1007/s11036-013-0489-0

[47] Chen SC (2011) Parameter tuning, feature selection and weight assignment of features for case-based reasoning by artificial immune system. Journal Applied Software Computing 11(8)

[48] Chin WH, Fan Z, Haines R (2014) Emerging technologies and research challenges for 5g wireless networks. Wireless Communications, IEEE 21(2):106–112

[49] Chun BG, Ihm S, Maniatis P, Naik M (2010) Clonecloud: boosting mobile device applications through cloud clone execution. arXiv preprint arXiv:10093088

[50] Chun BG, Ihm S, Maniatis P, Naik M, Patti A (2011) Clonecloud: elastic execution between mobile device and cloud. In: Proceedings of the sixth conference on Computer systems, ACM, pp 301–314

[51] Chun BN, Culler DE (2000) Market-based proportional resource sharing for clusters. Tech. rep., Berkeley, CA, USA

[52] Consolvo S, McDonald DW, Toscos T, Chen MY, Froehlich J, Harrison BL, Klasnja PV, LaMarca A, LeGrand L, Libby R, Smith IE, Landay JA (2008) Activity sensing in the wild: a field trial of ubifit garden. In: Proceedings of the 2008 Conference on Human Factors in Computing Systems, CHI 2008, 2008, Florence, Italy, April 5-10, 2008, pp 1797–1806, DOI 10.1145/1357054.1357335, URL http://doi.acm.org/10.1145/1357054.1357335

[53] Cuervo E, Balasubramanian A, Cho Dk, Wolman A, Saroiu S, Chandra R, Bahl P (2010) Maui: making smartphones last longer with code offload. In: Proceedings of the 8th international conference on Mobile systems, applications, and services, ACM, pp 49–62
[54] Dawson AW, Marina MK, Garcia FJ (2014) On the benefits of ran virtualisation in c-ran based mobile networks. In: Software Defined Networks (EWSDN), 2014 Third European Workshop on, IEEE, pp 103–108
[55] Diaz J, Munoz-Caro C, Nino A (2012) A survey of parallel programming models and tools in the multi and many-core era. IEEE Transactions on Parallel and Distributed Systems 23(8):1369–1386
[56] Dinh HT, Lee C, Niyato D, Wang P (2013) A survey of mobile cloud computing: architecture, applications, and approaches. Wireless Communications and Mobile Computing 13(18):1587–1611
[57] Durkee D (2010) Why cloud computing will never be free. In: Communications of the ACM, ACM, pp 62–69
[58] Dutta P, Aoki PM, Kumar N, Mainwaring AM, Myers C, Willett W, Woodruff A (2009) Common sense: participatory urban sensing using a network of handheld air quality monitors. In: Proceedings of the 7th International Conference on Embedded Networked Sensor Systems, SenSys 2009, Berkeley, California, USA, November 4-6, 2009, pp 349–350, DOI 10.1145/1644038.1644095, URL http://doi.acm.org/10.1145/1644038.1644095
[59] Fernando N, Loke SW, Rahayu W (2013) Mobile cloud computing: A survey. Future Generation Computer Systems 29(1):84–106
[60] Franchini S, Gentile A, Sorbello F, Vassallo G, Vitabile S (2012) Design space exploration of parallel embedded architectures for native clifford algebra operations. IEEE Design & Test of Computers 29(3):60–69, DOI 10.1109/MDT.2012.2206150
[61] Franchini S, Gentile A, Sorbello F, Vassallo G, Vitabile S (2013) Design and implementation of an embedded coprocessor with native support for 5d, quadruple-based clifford algebra. IEEE Transactions on Computers 62(12):2366–2381, DOI 10.1109/TC.2012.225
[62] Franchini S, Gentile A, Sorbello F, Vassallo G, Vitabile S (2015) Conformalalu: a conformal geometric algebra coprocessor for medical image processing. IEEE Transactions on Computers 64(4):955–970, DOI 10.1109/TC.2014.2315652
[63] Gantz J, Reinsel D (2011) Extracting value from chaos. IDC iview 1142:1–12
[64] Gea T, Paradells J, Lamarca M, Roldan D (2013) Smart cities as an application of Internet of Things: Experiences and lessons learnt in barcelona. In: Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference On, IEEE, pp 552–557
[65] George G, Haas MR, Pentland A (2014) Big data and management. Academy of Management Journal 57(2):321–326
[66] Grolinger K, Higashino WA, Tiwari A, Capretz MA (2013) Data management in cloud environments: Nosql and newsql data stores. Journal of Cloud Computing: advances, systems and applications 2(1):22
[67] Grünerbl A, Osmani V, Bahle G, Carrasco JC, Oehler S, Mayora O, Haring C, Lukowicz P (2014) Using smart phone mobility traces for the diagnosis of depressive and manic episodes in bipolar patients. In: 5th Augmented Human International Conference, AH '14, Kobe, Japan, March 7-9, 2014, pp 38:1–38:8, DOI 10.1145/2582051.2582089, URL http://doi.acm.org/10.1145/2582051.2582089
[68] Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of things (iot): A vision, architectural elements, and future directions. Future Generation Computer Systems 29(7):1645 – 1660, DOI http://dx.doi.org/10.1016/j.future.2013.01.010, URL http://www.sciencedirect.com/science/article/pii/S0167739X13000241

[69] Hashem IAT, Yaqoob I, Anuar NB, Mokhtar S, Gani A, Khan SU (2015) The rise of âĂIJbig dataâĂİ on cloud computing: Review and open research issues. Information Systems 47:98 – 115, DOI http://dx.doi.org/10.1016/j.is.2014.07.006, URL http://www.sciencedirect.com/science/article/pii/S0306437914001288

[70] Herbordt MC, VanCourt T, Gu Y, Sukhwani B, Conti A, Model J, DiSabello D (2007) Achieving high performance with fpga-based computing. Computer 40(3):50–57

[71] Holland O, Bogucka H, Medeisis A (2015) Opportunistic Spectrum Sharing and White Space Access: The Practical Reality. John Wiley & Sons

[72] Howes L (2015) The OpenCL Specification Version: 2.1. Khronos OpenCL Working Group, doc. rev.: 23 edn, https://www.khronos.org/registry/cl/specs/opencl-2.1.pdf

[73] Huang D, Zhou Z, Xu L, Xing T, Zhong Y (2011) Secure data processing framework for mobile cloud computing. In: Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on, IEEE, pp 614–618

[74] Huerta-Canepa G, Lee D (2010) A virtual cloud computing provider for mobile devices. In: Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, ACM, New York, NY, USA, MCS '10, pp 6:1–6:5, DOI 10.1145/1810931.1810937, URL http://doi.acm.org/10.1145/1810931.1810937

[75] Hull B, Bychkovsky V, Zhang Y, Chen K, Goraczko M, Miu A, Shih E, Balakrishnan H, Madden S (2006) Cartel: a distributed mobile sensor computing system. In: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, SenSys 2006, Boulder, Colorado, USA, October 31 - November 3, 2006, pp 125–138, DOI 10.1145/1182807.1182821, URL http://doi.acm.org/10.1145/1182807.1182821

[76] Irwin DE, Grit LE, Chase JS (2004) Balancing risk and reward in a market-based task service. In: High performance Distributed Computing, 2004. Proceedings. 13th IEEE International Symposium on, pp 160–169, DOI 10.1109/HPDC.2004.1323519

[77] Istepanian R, Hu S, Philip N, Sungoor A (2011) The potential of Internet of m-health Things âĂIJm-IoTâĂİ for non-invasive glucose level sensing. In: Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE, IEEE, pp 5264–5266

[78] Kemp R, Palmer N, Kielmann T, Bal H (2012) Mobile Computing, Applications, and Services: Second International ICST Conference, MobiCASE 2010, Santa Clara, CA, USA, October 25-28, 2010, Revised Selected Papers, Springer Berlin Heidelberg, Berlin, Heidelberg, chap Cuckoo: A Computation Offloading Framework for Smartphones, pp 59–79. DOI 10.1007/978-3-642-29336-8_4, URL http://dx.doi.org/10.1007/978-3-642-29336-8_4

[79] Kshetrimayum RS (2009) An introduction to UWB communication systems. Potentials, IEEE 28(2):9–13

[80] Lamkin P (2016) Connected cooking: The best smart kitchen devices and appliances. Online, URL http://www.wareable.com/smart-home/best-smart-kitchen-devices

[81] Lane ND, Miluzzo E, Lu H, Peebles D, Choudhury T, Campbell AT (2010) A survey of mobile phone sensing. IEEE Communications Magazine 48(9):140–150, DOI 10.1109/MCOM.2010.5560598, URL http://dx.doi.org/10.1109/MCOM.2010.5560598

[82] Lane ND, Mohammod M, Lin M, Yang X, Lu H, Ali S, Doryab A, Berke E, Choudhury T, Campbell A (2011) Bewell: A smartphone application to monitor, model and promote wellbeing. In: 5th international ICST conference on pervasive computing technologies for healthcare, pp 23–26

[83] Li F, Wang Y (2007) Routing in vehicular ad hoc networks: A survey. Vehicular Technology Magazine, IEEE 2(2):12–22

[84] Liu B, Zydek D, Selvaraj H, Gewali L (2012) Accelerating high performance computing applications using cpus, gpus, hybrid cpu/gpu, and fpgas. In: Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2012. Proceedings. 13th International Conference on, pp 337–342

[85] Malligai V, Kumar VV (2014) Cloud based mobile data storage application system. International Journal of Advanced Research in Computer Science and Technology 2:126–128

[86] Manyika J, Chui M, Brown B, Bughin J, Dobbs R, Roxburgh C, Byers AH (2011) Big data: The next frontier for innovation, competition, and productivity. Online, URL http://www.mckinsey.com/business-functions/business-technology/our-insights/big-data-the-next-frontier-for-innovation

[87] Marinelli E (2009) Hyrax: Cloud computing on mobile devices using mapreduce. Masters dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA

[88] Marr B (2015) How big data is changing healthcare. Online, URL http://www.forbes.com/sites/bernardmarr/2015/04/21/how-big-data-is-changing-healthcare/2/#7eb9ec6b5b7a

[89] Martin C (2014) Multicore processors: Challenges, opportunities, emerging trends. In: Embedded World Conference

[90] Mathur S, Jin T, Kasturirangan N, Chandrasekaran J, Xue W, Gruteser M, Trappe W (2010) Parknet: drive-by sensing of road-side parking statistics. In: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys 2010), San Francisco, California, USA, June 15-18, 2010, pp 123–136, DOI 10.1145/1814433.1814448, URL http://doi.acm.org/10.1145/1814433.1814448

[91] Miluzzo E (2014) I'm cloud 2.0, and i'm not just a data center. IEEE Internet Computing 18(3):73–77, DOI 10.1109/MIC.2014.53

[92] Morris ME, Kathawala Q, Leen TK, Gorenstein EE, Guilak F, DeLeeuw W, Labhard M (2010) Mobile therapy: case study evaluations of a cell phone application for emotional self-awareness. Journal of medical Internet research 12(2):e10

[93] Murray DG, Yoneki E, Crowcroft J, Hand S (2010) The case for crowd computing. In: Proceedings of the Second ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds, ACM, New York, NY, USA, MobiHeld '10, pp 39–44, DOI 10.1145/1851322.1851334, URL http://doi.acm.org/10.1145/1851322.1851334

[94] Nickolls J, Dally WJ (2010) The gpu computing era. IEEE Micro 30(2):56–69

[95] Pejovic V, Musolesi M (2015) Anticipatory mobile computing: A survey of the state of the art and research challenges. ACM Comput Surv 47(3):47, DOI 10.1145/2693843, URL http://doi.acm.org/10.1145/2693843

[96] Perera C, Liu CH, Jayawardena S (2015) The emerging Internet of Things marketplace from an industrial perspective: a survey. Emerging Topics in Computing, IEEE Transactions on 3(4):585–598

[97] Plummer L (2015) Smart rings: The good, the bad and the ugly in smart jewellery. Online, URL http://www.wareable.com/smart-jewellery/best-smart-rings-1340

[98] Qasim SM, Abbasi SA, Almashary B (2009) An overview of advanced fpga architectures for optimized hardware realization of computation intensive algorithms. In: Multimedia, Signal Processing and Communication Technologies (IMPACT ÅŘ09), 2009. Proceedings. International Conference on, pp 300–303

[99] Ram A, Wiratunga N (2011) Case-based reasoning research and development. In: 19th International Conference on Case-Based Reasoning, Springer

[100] Regehr J, Stankovic J, Humphrey M (2000) The case for hierarchical schedulers with performance guarantees. University of Virginia, Technical Report CS-2000-07
[101] Robison WJ (2010) Free at what cost? cloud computing privacy under the stored communications act. Georgetown Law Journal 98(4)
[102] Sawh M (2015) 2016 preview: How wearables are going to take over your life. Online, URL http://www.wareable.com/wearable-tech/2016-how-wearables-will-take-over-your-life-2096
[103] Sherwani J, Ali N, Lotia N, Hayat Z, Buyya R (2004) Libra: a computational economy-based job scheduling system for clusters. Software: Practice and Experience 34(6):573–590
[104] Soares S (2012) Big data governance - an emerging imperative. MC Press Online, LLC
[105] Studei P, Mohomed I, Terry D (2010) Wherestore: Location-based data storage for mobile devices interacting with the cloud. In: Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, ACM, p 1
[106] Vergara EJ, Nadjm-Tehrani S (2013) Energybox: A trace-driven tool for data transmission energy consumption studies. In: Energy Efficiency in Large Scale Distributed Systems - COST IC0804 European Conference, EE-LSDS 2013, Vienna, Austria, April 22-24, 2013, Revised Selected Papers, pp 19–34, DOI 10.1007/978-3-642-40517-4_2, URL http://dx.doi.org/10.1007/978-3-642-40517-4_2
[107] Vestias M, Neto H (2014) Trends of cpu, gpu and fpga for high-performance computing. In: Field Programmable Logic and Applications (FPL), 2014. Proceedings. 24th International Conference on, pp 1–6
[108] Want R (2011) Near field communication. IEEE Pervasive Computing (3):4–7
[109] Wiriyacoonkasem S, Esterline AC (2000) Adaptive learning expert systems. In: Proceedings of the IEEE, IEEE, pp 445–448
[110] Wolski R, Spring NT, Hayes J (1999) The network weather service: A distributed resource performance forecasting service for metacomputing. Future Gener Comput Syst 15(5-6):757–768, DOI 10.1016/S0167-739X(99)00025-4, URL http://dx.doi.org/10.1016/S0167-739X(99)00025-4
[111] Xie M, Lu Y (2012) Tianhe-1a interconnect and message-passing services. IEEE Micro 32(2):8–10
[112] Yan Y, Qian Y, Sharif H, Tipper D (2013) A survey on smart grid communication infrastructures: Motivations, requirements and challenges. Communications Surveys & Tutorials, IEEE 15(1):5–20
[113] Yardley L (2013) Uptake and usage of digital self-management interventions: Triangulating mixed methods studies of a weight management intervention. In: Medicine 2.0 Conference, JMIR Publications Inc., Toronto, Canada
[114] Yeo CS, Buyya R (2005) Service level agreement based allocation of cluster resources: Handling penalty to enhance utility. In: Cluster Computing, 2005. IEEE International, pp 1–10, DOI 10.1109/CLUSTR.2005.347075
[115] Yu E (2015) Iot to generate massive data, help grow biz revenue. Online, URL http://www.zdnet.com/article/iot-to-generate-massive-data-help-grow-biz-revenue/
[116] Zhang J, Orlik PV, Sahinoglu Z, Molisch AF, Kinney P (2009) Uwb systems for wireless sensor networks. Proceedings of the IEEE 97(2):313–331
[117] Zhang X, Schiffman J, Gibbs S, Kunjithapatham A, Jeong S (2009) Securing elastic applications on mobile devices for cloud computing. In: Proceedings of the 2009 ACM Workshop on Cloud Computing Security, ACM, New York, NY, USA, CCSW '09, pp 127–134, DOI 10.1145/1655008.1655026, URL http://doi.acm.org/10.1145/1655008.1655026

[118] Zhang X, Kunjithapatham A, Jeong S, Gibbs S (2011) Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing. Mob Netw Appl 16(3):270–284, DOI 10.1007/s11036-011-0305-7, URL http://dx.doi.org/10.1007/s11036-011-0305-7

[119] Zissis D, Lekkas D (2012) Addressing cloud computing security issues. Future Generation Computer Systems 28(3):583 – 592, DOI http://dx.doi.org/10.1016/j.future.2010.12.006, URL http://www.sciencedirect.com/science/article/pii/S0167739X10002554

Chapter 3
Big Data software platforms

Milorad Tosic, Valentina Nejkovic and Svetozar Rancic

3.1 Cloud computing software platforms

Apostolos N. Papadopoulos

One of the big successes of distributed and parallel computing is that nowadays there exist flexible software tools that help the programmer significantly. The development of a distributed application *from scratch* given a specific programming language is difficult, time consuming and error prone. The programmer must attack several diverse problems such as: 1) to provide the required functionality, 2) to provide the means for the coordination of multiple resources, 3) to guarantee fault tolerance and 4) to guarantee the efficiency of the application.

In this section, we describe two of the most important software platforms that enjoy a wide-spread use in the development of distributed applications: Apache Hadoop and Apache Spark.

3.2 Semantic dimensions in Big Data management

M. Tosic, V. Nejkovic, S. Rancic

Milorad Tosic
University of Nis, Faculty of Electronic Engineering, Department of Computer Science, Laboratory of Intelligent Information Systems, Serbia,
e-mail: milorad.tosic@elfak.ni.ac.rs

Valentina Nejkovic
University of Nis, Faculty of Electronic Engineering, Department of Computer Science, Laboratory of Intelligent Information Systems, Serbia, e-mail: valentina@elfak.ni.ac.rs

Svetozar Rancic
University of Nis, Faculty of Science and Mathematics, Department for Computer Science, Serbia, e-mail: rancicsv@yahoo.com

The Big Data solutions can provide infrastructure and tools for handling, processing and analyzing different datasets. However, efficient approaches that can structure, aggregate, annotate, share and make sense of huge different application domains datasets as well as transform it to valuable knowledge and intelligence are still missing. We envisage that semantic technologies can be the tool to successfully address that challenges. In order to achieve seamless integration of the Big Data in business applications and services, semantic description of different resources in different domains is considered as a key task.

In the context of Big Data, each of the traditionally well-understood data processing tasks, such as computer-human interfaces, application development, system administration, etc., has suffered a fundamental change. The main drivers of the change are non-functional characteristics of the Big Data, often referred to as 4V's [13]. Let us consider application dealing with video streaming of a city traffic as a running example. End user's perception of and expectation from such an application in a traditional case would include providing end users with a mobile/web access to videos of the traffic at selected places in the city with a goal to get a general information about traffic intensity. However, in a case when a large number of cameras are located on every couple of meters at every single street in the city interleaved with sensors, such as temperature and/or CO2, the amount of data grows significantly together with opportunities for new applications. So, in addition to storing, searching and display of large volume of data in a traditional case, the new applications deal with computationally demanding complex feature processing such as trends, anomalies, complex interdependencies, data patterns, etc. An illustrative case for such applications can be for example an imaginary âĂIJpersonal pollution metarâĂİ application that would integrate data from temperature and CO2 sensors with information about air pollution level approximated out of the video streams. While high-level data integration is considered as granted in traditional applications, in the Big Data driven application it is the very challenging task of primary importance. The system infrastructure needed to support such application presents new challenges as well, such as the need to manage a large number of heterogeneous devices within the same system, highly distributed nature of the system including tightly coupled parts in data centers as well as loosely coupled sensor networks, and heterogeneous interconnection networks.

Fig. 3.1 shows the four dimensions of conceptualization identified in Big Data management: D1) Applications, D2) Users, D3) Infrastructure, and D4) Data sources. Most important aspects of each of the dimensions are listed as well. Application domain determines most of the non-functional system requirements. The most important application domains are: Internet of Things (IoT), sensor networks, application development frameworks, medical and dental data, medicine cross linking, bioinformatics, environment data, smart cities, etc. At the application dimension, explicit representation of semantics and knowledge enables system development at higher abstraction levels. In this way, development of new application frameworks and middleware is possible without constraints imposed by functional requirements specific for Big Data. For example, if geo-location knowledge is explicitly encapsulated with the system, usually by using ontologies and knowledge bases, then same software frameworks and architectures are reusable across different applications such as IoT, smart cities, and sensor networks. Functional requirements mostly focus on users concerning algorithmic capabilities and human-computer interaction. Typically, graph and matrix algorithms are implemented for approximations, mining and machine learning applications. End-user interaction is becoming increasingly important where advanced interfaces are de facto commonly expected, such as natural language interface, simplified english, specific language interface, multimedia interface, virtual reality, augmented reality, etc. The pressure that Big Data has put on the global computing infrastructure resulted on diverse emerging architectural principles such as cloud computing, virtualization, compartmentalization, and microservices, that have been applied on large data centres as well as on networking infrastructure. Dimension D3 in Figure 1 relates to the infrastructural aspect of the system. The fourth dimension shown in Figure 1 concerns low-level physical nature of heterogeneity of data sources feeding the Big Data systems. Data is collected from social, environmental and transactional sources including streaming sensors data, data from social networks, financial and business transactional data, multimedia streaming, etc.

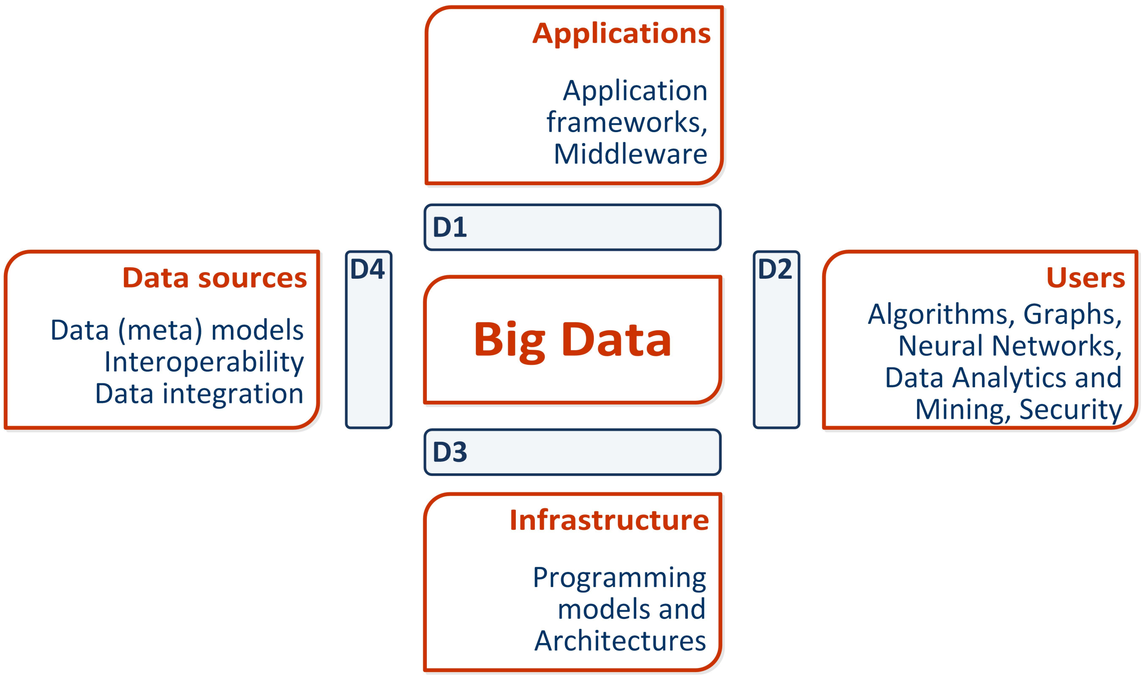

Fig. 3.1 Four dimensions of conceptualization in Big Data

3.2.1 Applications (D1)

The opportunities from Big Data have the potential for different application domains. Based on the reviewed literature, we identified following areas that can benefit from Big Data research:

Internet of Things (IoT)

Today world becomes more and more inter-connected which evolved to the basic idea of IoT that rely on pervasive presence of thing and object or in general entities [3]. Entities interact and cooperate with each others in order to reach some goal. IoT is multidisciplinary and include different domains such as computer science, engineering, electronics, telecommunication and social sciences. IoT is global network of real and virtual world of distributed entities interconnected [10]. IoT includes huge amount of distributed and heterogeneous entities. Distributed and heterogeneous IoT entities should be consistent, formal presented and managed.

We identified following IoT application domains where semantic technologies can be used for solving domain specific problems: a) Smart home with energy waste, interactive walls with useful information, etc; b) Smart cities with productive retail, residential and green spaces with coexistence; c) Retail environments with support reflected to traceability of products and convenient shopping experience to consumers; d) Intelligent transportation systems with easy multimodal transport and interactions of public and private transportations that will enable the best path choosing; e) Smart health environment with monitoring systems that provide a services to prevent illness by selecting drugs or diets; f) Logistics environment with safety and environmental concerns or green logistics with a goal to minimize environmental damage during operations.

Wireless sensor networks

Very important contributor of the big data in the future networks is the wireless sensor networks (WSN) [21]. WSNs are used for many purposes, for example for environmental parameters measuring (temperature, humidity, pressure, wind direction and speed, etc.), industrial applications (such as industrial process management, etc.) etc.. Data from WSNs can be divided to static and dynamic data. Static data for example describe characteristics data of sensors, while dynamic are perceptible data from sensors. In the past, WSNs collected dynamic data for later analysis performed by the expert of environmental sensor network. Today's WSNs use real-time dynamic data processing, and some of them enable control of sensor activity. Data generated across numerous sensors in the densely distributed WSNs networks can produce a significant exponentially increasing amounts of data. Big data approaches and tools can be used for the collection, storage, manipulation, analysis and exploitation of data generated by a WSNs [19]. Additional relevant areas include data aggregation and filtering, security, querying, data transfer protocols, remote and real-time monitoring, and sensor-cloud systems.

Healthcare (medical and dental domain)

Due to many players involved in this domain, very challenging is research on data sharing because of privacy concerns. We met mostly not structured and not linked data, which should be interconnected in order to be used for further data analysis and reasoning [9]. This domain include the pharmaceutical and medical products industries, providers, payers and patients. The complexity of this domain datasets is a challenge in analyzing and using in clinical environments. For example, for clinical practice the evidence based medicine concept is widely used, where therapy and treatment decisions for patients are made based on the best scientific evidence available. Aggregation of individual patients datasets represents the best source for evidence.

The volume of global data in this area grows exponentially, which comes from the digitalization of existing data as well as from day to day new data generation. This data includes personal medical records, radiology images, clinical trial data, FDA submissions, human genetics and population data, genomic sequences, 3D imaging, genomics and biometric sensor readings, etc. [7]. The other data sources that should be also considers are following: a) claims and cost data that describe what services were provided and how they were reimbursed, b) pharmaceutical data that gives medicine details such as: medicine side effects and toxicity, therapeutic mechanism, body behavior etc.; c) patient behavior and data that describe patient activities and preferences, both inside and outside the healthcare context [9].

Social media and web data

The rise of social network sites generate a huge amount of dataset in the form of postings, tweets, blogs, bookmarking, tagging, massages etc [4]. These data and e-commerce transactions are challenging for further analysis. Today we can analyse connections and trajectories that are formed by huge number of experiences, cultural expression, texts and links, which forms two data types surface and deep data [15]. Surface data are data about lots of individuals, while deep data about small groups of individuals. Surface data comes from the domains of where are usually used quantitative methods for data analyzing, such as statistical and computational methods. These domains usually belong to social sciences such as quantitative sociology, economics, political science, communication studies, and marketing research. From the other side, deep data can be obtained from qualitative humanity disciplines such as non-quantitative psychology, sociology, anthropology and ethnography.

Personal location data

Personal location data are not confined to a single domain but rather through many different domains. Sources that produce personal location data are: credit card payments and transactions via automated teller machine; mobile phones that includes cell-tower signals, mobile WiFi and GPS sensors; inside building different personal tracks. Personal location data usage are reflected on location based applications and services (smart routing and mobile phone based location services), global aggregation of personal location data (business intelligence and urban planning) and personal location data usage based on specific needs of organisations (electronic toll collection, insurance pricing, emergency response, advertising based on geographical location etc.) [25]. These data can be applied on domains that includes travel planning, terrorists tracking, child safety, law enforcement etc.

Global manufacturing

Blending data coming from different systems across organizational boundaries in an end-to-end supply chain, with computer-aided design, engineering and manufacturing, collaborative product development management and digital manufacturing are today reality in manufacturing community [16]. Organizations need to be intelligent and agile by using big data, to collaborate with the partners along the value chain, to manage complexity and enhance performances. In order to achieve these goals, manufacturers need to develop new operational capabilities and approaches. Novel big data analytical tools can help identify opportunities to serve new markets and better manage supply chain. Further, big data can creates new opportunities and categories of companies. For example, companies for aggregating and analyzing industry data, which must be in the middle of large information flows where are data on products, services, buyers and suppliers, and which should be captured and analyzed [17].

Bioinformatics

Bioinformatics use bioinformatics technologies for storing, retrieving, organizing and analyzing biological data. The volume, variety and velocity of data today is rapidly increasing thanks to efficacy of such technologies [12]. The biological and biomedical data sets are large, diverse, distributed, and heterogeneous. Usage of big data tools on such data sets can lead to critical data-intensive decision making and support, and provide biological predictions [12], [18]. One of reasons of such data increase within bioinformatics is because biologists no longer use traditional laboratories to discover a novel biomarker for a disease. Technologies for capturing bio data are becoming cheaper and more effective, such as automated genome sequencers. Thus, biologists now can use huge and continuously growing genomic data available through biology data repositories which are made by bioinformatics research community. For example, one of the largest biology data repositories of the European Bioinformatics Institute in 2014 had approximately 40 petabytes of data about genes, proteins, and small molecules.

Smart cities

Smart cities links many aspects related to human life such as transportation, buildings and power in smart manner in order to improve the quality of human life in the city [1]. In order to become smart a city should provide intelligent management of infrastructures and natural resources, which require investing in more technology and effective use of big data. Policies for data accuracy, high quality and security, privacy and control need to be set, as well as documents standards for content and datasets usage guidance should be developed, too [5]. City environments could benefit from having consolidated services across different areas, which can be enabled by the implementation of data warehouses and business intelligence. Smart cities should support a data fusion of a variety of data sources, such as video from different cameras, voice, social media, streaming data, sensor logs, supervisory control systems and data acquisition, traditional structured and unstructured data [6].

Big data applications have huge potential to be used in smart city, for example in order to improve healthcare sector such as preventive care services, diagnosis and treatments, healthcare data management etc; transportation sector such as route and schedules; waste management sector such as waste collection, disposal, recycling and recovery; water management; smart education which can engage people in active learning environments that allow them to adapt to the rapid changes of society and the environment; smart traffic lights such as a good control of the traffic flow within the city; smart grid which improves the efficiency, reliability, economics, and sustainability of the production and distribution of electric power [1].

3.2.2 Users – Functionalities focused on end users (D2)

Functionalities focused on end users should enable easy way to organize, storage, search and process distributed big data to end user applications, as well as enable end users easy access to information using simplified interfaces like natural language interfaces, simplified english or multimedia interfaces.

Regardless the type of the database used for store Big Data, different reasons may lead to use NoSQL (Not Only SQL) database. There are reported successful applications of mixed SQL and NoSQL database solutions, for example global distribution systems Amadeus which distributes flight inventory where read part of the database is a key-value store, which serves online 3 to 4 million access per second. Also, application of NoSQL as a basic database store, like i2O system for businesses with significant water need and management, which highlight scalability and effectiveness on wide columns data, which contain large volumes of time series data. Further, Temetra system, which use cloud-based storage system, for its key-value qualities, which allow to store large volumes of unstructured data.

Despite on possible ACID (Atomic, Consistent, Isolated and Durable) disadvantage of the NoSQL databases, they have essential benefits: Scalability (effective distribution of data and load of simple operation among servers), Availability and Fault tolerance (ensured by replicated and distributed data among servers) and Greater performance (key-value stores effectively uses distributed resources: indexes and main memory for data storage). Those store features are basis for interface engine and further for interface to the user.

Database, as a kind of storage, used for moderate large graphs is mentioned in [2]. Furthermore, accessing very large graphs - in some cases billions of nodes and trillions of edges - poses challenges to efficient processing. The rise of MapReduce concept and Hadoop, its open source implementation, provide a powerful tool for doing analyzes of large data collection, also graphs. It uses Hadoop Distributed File System (HDFS) for store and it facilitate processing of huge graphs. An alternative is Bulk Synchronous Parallel technique, based on [22]. Google started Pregel, a proprietary system, and in [14] is presented a suitable computational model based on iterations and messages. Algorithms can be implemented as sequence of iterations, in each of which a vertex can receive message sent in the previous iteration, send message to other vertices, and modify its own state. This vertex-centric approach is flexible enough to express a broad set of algorithms, through efficient, scalable and fault-tolerant implementation on clusters of commodity computers. Distribution, communication, fault tolerant and other details are hidden behind an abstract API. There are open source BSP model implementation Apache Hama or Apache Giraph. In graph algorithms neighborhood vertices of a vertex should be processed and both approaches do it, but MapReduce uses distributed file system (HDFS), and BSP uses local memory. It determine its features and performances, for comparison see [11] or [20]. Based on this storage and computational approaches we are able to perform offline or online graph analysis, for different purposes: finding shortest paths, clustering, bipartite matching etc.

Sensor systems acquire large collection of data, which deep analytics is challenging. But it can give users very useful information, which can be used proactively, furthermore can improve system efficiency and give competitive advantages. A lot of refining of input data can be needed as first step. Anomaly detection on time series data obtained by sensors, especially online approach [24] is very demanding, but it is important.

Neural network also can be used for processing, let us mention a picture recognizing and classifying. Problem of automated describing the content of an image appear as fundamental and connects computer vision and natural language processing. A generative model based on deep recurrent architecture that combines Vision deep convolutional neural network (CNN) followed by recurrent neural network (RNN) is presented in [23]. It can generate complete sentences in natural language from an input image. Obtained result can be used further to classify pictures, as well as, to give user ability to search similar pictures using their own picture, or describing content by simplified english.

Additional data can be added on existing ones, on database repository, to improve search on them. It ranges from simple idea like indexing structure, a kind of indexer, to more complex approach aimed for knowledge based system (KBS). They can improve search and give user improved interface, as ability to query using natural language interface or simplified english or specific language interface. It can contains an underlying set of concepts, assumptions and rules, a kind of ontology, which a computer system has available to solve problem. It can be a part of an expert system that contains the facts and rules, which provides provides intelligent decisions with justifications and uses artificial intelligence for problem solving and to support human learning and action.

3.2.3 Infrastructure (D3)

The characteristics of Big Data pose significant challenges with respect to processing and analysis. Traditional architectures and systems face difficulties in scaling well. For example, scaled-up database management systems are not flexible enough to handle different data mining tasks. The limitations of existing architectures have led to the era of cluster computing, where many (usually commodity) machines are interconnected using a shared-nothing architecture. Modern data centers are composed of thousands of CPUs and disks that must be coordinated appropriately in order to deliver the required results.

3.2.4 Data sources (D4)

There is variety of domains of big data sources. Those domains should be categorized with a goal of understanding infrastructural requirements for particular data types [8]. Architectures needed for storing, processing and analyzing big data are different and dependent on particular data domains.

Sources of big data can be characterized based on time span, structure and specific domain.

Time span is related to latency. Latency is time available to compute the results or the time span between when an input event occurs and the response on the event. The overall latency of a computation at a high level can be divided on communication network latency and computation latency. Thus, according time span in which big data needs to be analyzed it can be characterized as real-time, near real-time and not real-time. Real-time data arriving âĂIJin real-timeâĂİ. For example, some of applications that use real-time data are: event processing, financial streams, forecast real-time data streams, web analytics, etc. Near real-time is for example ad placement to the web site, while not real-time data applications are bioinformatics, retail, geo-data, etc. Big data based on structure can be structured, semi-structured and unstructured. Examples of structured data are financial, bioinformatics, geo-data, etc. Semi-structured big data are different documents, web logs, emails, while unstructured are web contents, video, images, sensor data etc. Big data are derived from domains such as: large-scale science (Bioinformatics, medical and dental data, high energy physics), sensor-networks (Weather, anomaly detection), financial services, global manufacturing (retail, behavioral data, financial services), smart cities (finance, government, transportation, buildings, power, voice, social media, streaming data, sensor logs, waste; water management; smart education), social media and web data (social networking, sentiment analysis, social graphs, network security).

Regardless the data source or chosen data storage or the needed data analytics a user demands the value from the data. It gives the request for data processing despite the immanent differences. A complex big data application, also a relatively simple one, usually has growing rate of expected features, analytic methods, possible data sources etc. All mentioned facilitate the needs for interoperability and data integration on such way which, as result, gives an open system. This system should easily adopt new data source, as well as, new data storage and new data analytics.

Semantic data modelling facilitates integration of management of new types of data together with their place of storage, thus improving interoperability over heterogeneous resources. It also makes possible to apply data analytics methods once written then applied many times. The semantics data modelling approach also serves well as cohesion mechanism for stages in big data applications, providing integration of data and data processing activities.

References

[1] Al Nuaimi E, Al Neyadi H, Mohamed N, Al-Jaroodi J (2015) Applications of big data to smart cities. Journal of Internet Services and Applications 6(1):1–15, DOI 10.1186/s13174-015-0041-5, URL http://dx.doi.org/10.1186/s13174-015-0041-5

[2] Aljazzar H, Leue S (2011) K*: A heuristic search algorithm for finding the k shortest paths. Artificial Intelligence 175(18):2129 – 2154, DOI http://dx.doi.org/10.1016/j.artint.2011.07.003, URL http://www.sciencedirect.com/science/article/pii/S0004370211000865

[3] Atzori L, Iera A, Morabito G (2010) The internet of things: A survey. Computer Networks 54(15):2787 – 2805, DOI http://dx.doi.org/10.1016/j.comnet.2010.05.010, URL http://www.sciencedirect.com/science/article/pii/S1389128610001568

[4] Barnaghi P, Wang W, Henson C, Taylor K (2012) Semantics for the internet of things: Early progress and back to the future. Int J Semant Web Inf Syst 8(1):1–21, DOI 10.4018/jswis.2012010101, URL http://dx.doi.org/10.4018/jswis.2012010101

[5] Bertot JC, Choi H (2013) Big data and e-government: Issues, policies, and recommendations. In: Proceedings of the 14th Annual International Conference on Digital Government Research, ACM, New York, NY, USA, dg.o '13, pp 1–10, DOI 10.1145/2479724.2479730, URL http://doi.acm.org/10.1145/2479724.2479730

[6] Delloite (2015) Smart cities, big data. online, URL https://www2.deloitte.com/content/dam/Deloitte/fpc/Documents/services/systemes-dinformation-et-technologie/deloitte_smart-cities-big-data_en_0115.pdf

[7] Feldman B, Martin EM, Skotnes T (2012) Big data in healthcare hype and hope. October 2012 Dr Bonnie 360

[8] data working group CSAB (2014) Big data taxonomy. online

[9] Groves P, Kayyali B, Knott D, Kuiken SV (2013) The âĂŸbig data'revolution in healthcare. online

[10] Hitzler P, Janowicz K (2013) Linked data, big data, and the 4th paradigm. Semantic Web 4(3):233–235

[11] Kajdanowicz T, Kazienko P, Indyk W (2014) Parallel processing of large graphs. Future Generation Computer Systems 32:324 – 337, DOI http://dx.doi.org/10.1016/j.future.2013.08.007, URL http://www.sciencedirect.com/science/article/pii/S0167739X13001726, special Section: The Management of Cloud Systems, Special Section: Cyber-Physical Society and Special Section: Special Issue on Exploiting Semantic Technologies with Particularization on Linked Data over Grid and Cloud Architectures

[12] Kashyap H, Ahmed HA, Hoque N, Roy S, Bhattacharyya DK (2015) Big data analytics in bioinformatics: A machine learning perspective. arXiv preprint arXiv:150605101

[13] Kune R, Konugurthi PK, Agarwal A, Chillarige RR, Buyya R (2016) The anatomy of big data computing. Softw Pract Exper 46(1):79–105, DOI 10.1002/spe.2374, URL http://dx.doi.org/10.1002/spe.2374

[14] Malewicz G, Austern MH, Bik AJ, Dehnert JC, Horn I, Leiser N, Czajkowski G (2010) Pregel: a system for large-scale graph processing. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, ACM, pp 135–146

[15] Manovich L (2011) Trending: The promises and the challenges of big social data. Debates in the digital humanities 2:460–475

[16] Manyika J, Sinclair J, Dobbs R, Strube G, Rassey L, Mischke J, Remes J, Roxburgh C, George K, O'Halloran D, Ramaswamy S (2012) Manufacturing the future: The next era of global growth and innovation. online

[17] Nedelcu B, et al (2013) About big data and its challenges and benefits in manufacturing. Database Systems Journal 4(3):10–19

[18] Nemade P, Kharche H (2013) Big data in bioinformatics & the era of cloud computin. IOSR Journal of Computer Engineering 14(2):53–56

[19] Rios LG, Diguez JAI (2014) Big data infrastructure for analyzing data generated by wireless sensor networks. In: 2014 IEEE International Congress on Big Data, pp 816–823, DOI 10.1109/BigData.Congress.2014.142

[20] Shao B, Wang H, Xiao Y (2012) Managing and mining large graphs: Systems and implementations. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, ACM, New York, NY, USA, SIGMOD '12, pp 589–592, DOI 10.1145/2213836.2213907, URL http://doi.acm.org/10.1145/2213836.2213907

[21] Takaishi D, Nishiyama H, Kato N, Miura R (2014) Toward energy efficient big data gathering in densely distributed sensor networks. IEEE Transactions on Emerging Topics in Computing 2(3):388–397, DOI 10.1109/TETC.2014.2318177

[22] Valiant LG (1990) A bridging model for parallel computation. Commun ACM 33(8):103–111, DOI 10.1145/79173.79181, URL http://doi.acm.org/10.1145/79173.79181

[23] Vinyals O, Toshev A, Bengio S, Erhan D (2015) Show and tell: A neural image caption generator. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3156–3164

[24] Yao Y, Sharma A, Golubchik L, Govindan R (2010) Online anomaly detection for sensor systems: A simple and efficient approach. Performance Evaluation 67(11):1059–1075

[25] Zaslavsky A, Perera C, Georgakopoulos D (2013) Sensing as a service and big data. arXiv preprint arXiv:13010159 URL http://arxiv.org/ftp/arxiv/papers/1301/1301.0159.pdf

Chapter 4
Data analytic and middleware

Mauro Iacono, Luca Agnello, Albert Comelli, Daniel Grzonka, Joanna Kołodziej, Salvatore Vitabile and Andrzej Wilczyński

4.1 Multi-agent systems

D. Grzonka, A. Wilczynski, J. Kolodziej

The integration of the Multi-Agent Systems (MAS) with computational clouds is in itself a new approach. Recently, however, many examples of successful applications of the MAS systems for supporting the cloud systems have been developed. Such integration can be successfully used also in modelling the crowd and decision support in emergency situations (see e.g. [44], [82]). The simple example of the integration of MAS with the cloud application layer is presented in Fig. 4.1. In general, there are three main types of agent-based models supporting various types of the cloud systems, namely:

- agent-based support for the management of cloud computing systems (top-down approach),
- agent-based ad hoc simulation of the mobility and activities of the system users and system dynamics (bottom-up approach), and

Mauro Iacono
Seconda Università degli Studi di Napoli, Italy, e-mail: mauro.iacono@unina2.it

Luca Agnello
Department of Biopathology and Medical Biotechnologies, University of Palermo, Italy, e-mail: luca.agnello@gmail.com

Albert Comelli
Department of Biopathology and Medical Biotechnologies, University of Palermo, Italy, e-mail: albert.comelli@gmail.com

Daniel Grzonka
Institute of Computer Science, Cracow University of Technology, Poland, e-mail: grzonka.daniel@gmail.com

Joanna Kołodziej
Institute of Computer Science, Cracow University of Technology, Poland, e-mail: jokolodziej@pk.edu.pl

Salvatore Vitabile
University of Palermo, Italy, e-mail: salvatore.vitabile@gmail.com

Andrzej Wilczyński
Institute of Computer Science, Cracow University of Technology, Poland, e-mail: awilczynski@pk.edu.pl

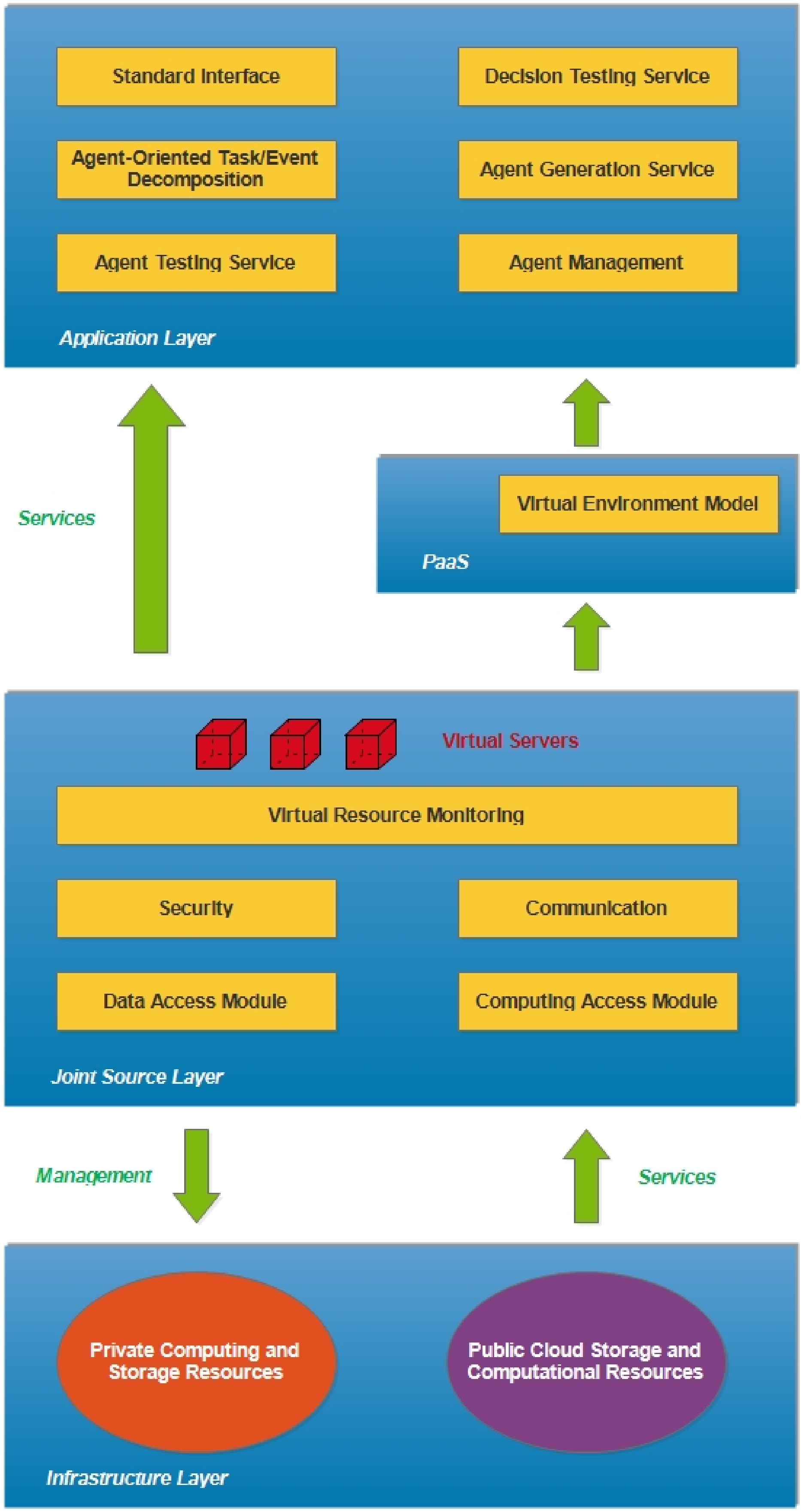

Fig. 4.1 Integration of Multi-Agent System with the Cloud Environment.

- software (mobile) agent integrated with the SaaS (Software as a Service) cloud layer for bolstering cloud service discovery, service negotiation, and service composition [91] [92] [93].

The multi-agent system (MAS) model of the multilayer cloud infrastructure is usually referred to as a top-down MAS approach, where the system organizational model is defined a priori and modified during system runtime. To ensure proper adaptation of the MAS, communication and reorganization are crucial properties in the modelling of the cloud system dynamics. Although numerous reorganization models have been proposed in the literature, they do not permit the representation of the dynamics of large-scale and highly distributed systems like in the cloud. As an example, AgentCoRe [52] is a framework designed for coordination and reorganization in MAS. The decision-making modules are capable of selecting coordination mechanisms, creating and updating tasks structures, assigning subtasks, and reorganizing them. GPGP/TAEMS [72] also proposes to decompose the organization function into goals and subgoals structured in trees in which leaves are interpreted as tasks. The limitations of such an approach reside in the limited agent adaptation capabilities because of the dependency on a given schedule; the domain-independent scheduler creates a single point of failure and the lack of scalability of the coordination mechanism. Wang and Liang [102] propose a three-layer organizational model divided into three graphs: social structure, role enactment, and agent coordination. In this approach the reorganization is based on a set of predefined graph reorganization rules. It is limited to changing the structural dimension of the organization, and thus not its functionalities. Kota et al. [70] present an adaptive self-organizing MAS that can reorganize on task assignment (i.e. functional) and structural levels. Agents in an organization provide services and possess a set of computational resources, which correspond to a CC system. Important concepts of organization cost, benefit and efficiency are also introduced. The reorganization process here is decentralized, but it only includes four possible actions: create/remove peer and create/ remove authority. Moise+ [58] is a top-down organization model that includes reorganization modeling capabilities and an agent programming methodology [57], and it was recently transformed into the ORA4MAS [59] Agent & Artifact framework. These recent developments are used in the JaCaMo multi-agent programming framework, which includes similar reorganization modeling mechanisms. One advantage of Moise+ is its ability to specify three organizational dimensions - structural, functional and deontic [60]. However, the functional specification is based on goal decomposition trees that limit the possibilities of strongly parallel executions. The basic Moise+ reorganization is designed for small systems, as only one agent is responsible for reorganization.

Cloud agents are usually defined as Mobile Software Agents (MSAs), which are implemented as software modules that autonomously migrate from machine to machine in a heterogeneous network, interacting with services at each machine to perform some desired task on behalf of the user. Cloud software agents are usually integrated with the SaaS cloud layer (see Fig. 4.1) for offloading the application tasks. Scavenger [71] is an example of such a mobile agent framework that uses a mobile code approach to partition and distribute tasks among the cloud servers and Wi-Fi as the connection technology. In the WMWave [95] project, software agents are utilized in order to manage and assign resources and to distribute applications and data. Other projects focus on the design of the mobile agent systems for a transformation of cloud services through different virtual machines and in different cloud contexts. Aversa et al. [12] developed the Cloud Agency architecture, which aims at offering Virtual Clusters on top of an existing grid security architecture with the support of mobile agent-based services. An interesting agent-based solution for the cloud system is the Mobile Agent Based Open Cloud Computing Federation (MABOCCF) [107]. In this system the mobile agents transport the data and code between the system devices. Each mobile agent is executed in a Mobile Agent Place (MAP) virtual machine. MABOCCF is an open source system.

Under WG1 works, it has been designed and implemented a cloud-based, multi-agent evacuation simulation platform [103]. The platform is based on the MapReduce programming model and the Hadoop framework. The architecture of the multi-agent simulation module is shown in Fig. 4.2) The proposed model consist of the head management node, two agent queues, the number of map and reduce workers, cell selection module and evacuation simulation display-module. It can be used with cloud physical clusters, which usually have head node and several slave nodes. In proposed model, the head node (Global Synchronization Manager) is responsible for coordination and management of whole simulation processing. Simulation is loop-based, it indicates that is performed continuously. After each loop, state and position of each agent is updated and at the end simulation scene is generated and displayed. The

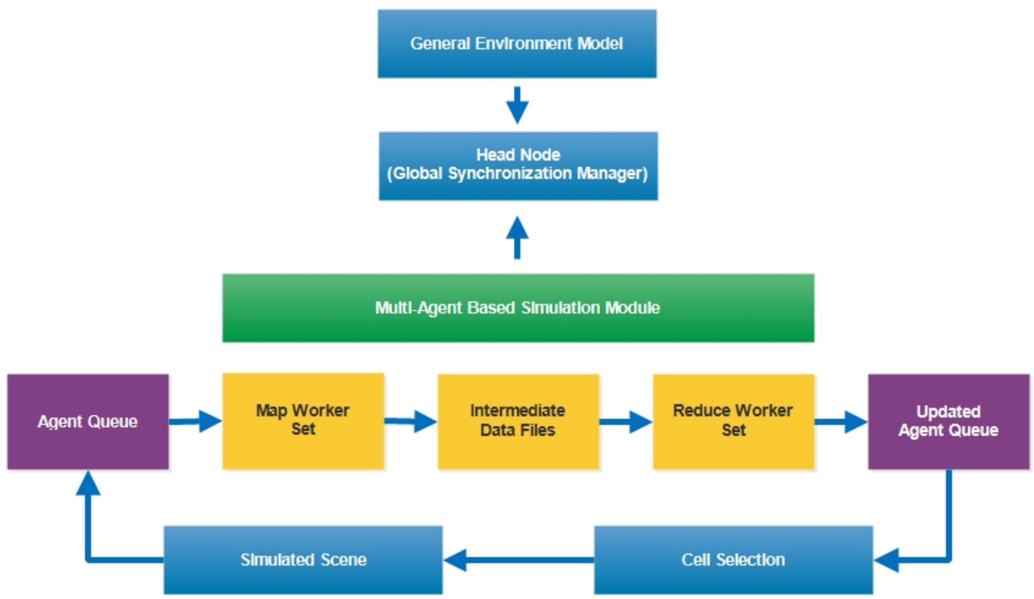

Fig. 4.2 MapReduce-based Simulation Module Architecture.

agent stores several information about its state, including: position, speed, health, recent moves, a set of weights and information about a candidate cell queue (surround the current cell agent). Based on the cell queue, the agent decides whether to perform motion in a given time step or not. It can only move to one of the cells in the candidate list. After that, the agents are sorted by Hadoop framework depending on their state of health. Then, agent items are transferred to the mapper.

The mapper creates a set of intermediate key/value pairs and calculates the candidate cell attraction value for all surrounding cells. Based on achieved results, an intermediate table is constructed. Next, table is sorted by the Hadoop framework according to cell attraction value of each pair. Based on the keys, the reducer collect information from sorted table and generate agent information. Again, the results are transmitted to the Hadoop, where the list of agents is composed. At the end, for each agent is selected one cell as a target. Then the agent is moved and the information about its location is updated. To prevent potential errors caused by parallel processing, the parallel map and reduce operations do not move agents.

The tests conducted in the cloud showed that the memory working set size for each individual in case of Hadoop is only 34% of the value achieved for the single Server, this is cause in the cloud environment, data processing operations are processes successfully executed for different data sets. This shows potential benefits of using cloud system in case of evacuation crown and proves to us that cloud-based systems can reduce significantly the complexity of the management of individuals in the crowd [103].

4.2 Middleware for e-Health

L. Agnello, A. Comelli, S. Vitabile

4.2.1 Brief introduction and early classification

Middleware layer is the link between a distributed application the underlying operating system that sends messages to remote component. Essentially, middleware is supplementary software that connects two or more software together [23],[6]. A more precise definition stated that the middleware is the software that assists an application to interact or communicate with other applications, networks, hardware, and/or operating systems [24].

Since the great variety of middleware and the different kind of services covered by them, there is the need to organize such component in a taxonomy based on the way each middleware aids applications or permits integration of multiple applications into a system. In [24], the authors proposed a taxonomy divided into two categories: integration type and application type. In what follows, middleware subcategories with relevant examples are briefly described.

Integration Type

These types of middleware have a specific way of being integrated into its heterogeneous system environment across different communication protocols or ways of operating between the other software. They are divided into five subcategories:

- Procedure Oriented Middleware: The client stub converts the parameters of a procedure into a message (marshalling) and sends this message to the server that converts the message back into the parameters and sends the procedure call to the server application where it is processed. The client stub also checks for errors, sends the results to the calling software, and then suspends the thread [23] [48].
- Object Oriented Middleware: It supports distributed object requests. The communication between objects (client object and server object) can be synchronous, deferred synchronous, and asynchronous using threading policies. Object middleware supports multiple transactions by grouping similar requests from several client objects into a transaction [48]. It also operates on many different data types, forms and states and is a self-manager of distributed objects (supports multiple simultaneous operations using multi-threading) [47].

- Message Oriented Middleware: MOM sends and receives messages between distributed systems. MOM allows application modules to be distributed over heterogeneous platforms.it can be subdivided into two subcategories:
  - message passing and/or message queuing supporting both synchronous and asynchronous interactions between distributed computing processes. The MOM system ensures message delivery by using reliable queues (either FIFO, based on a priority scheme, or based on a load-balancing method) and by providing the directory, security, and administrative services required to support messaging [65];
  - Message Publish/Subscribe - publishâĂŞsubscribe is a messaging pattern where senders of messages, called publishers, do not program the messages to be sent directly to specific receivers, called subscribers, but instead characterize published messages into classes without knowledge of which subscribers, if any, there may be. Similarly, subscribers express interest in one or more classes and only receive messages that are of interest, without knowledge of which publishers, if any, there are [100].
- Component Based or Reflective Middleware: This middleware designed to co-operate with other components and applications, it performs particular tasks [6]. It is a configuration of components selected either at build-time or at run-time. this middleware is that it has an extensive component library and component factories, which support construction on multiple platforms and it is configurable and re-configurable. Re-configurability can be accomplished at run-time, which offers a lot of flexibility to meet the needs of a large number of applications [25].
- Agents: this type of middleware consists of several components: entities (objects, threads), media (communication between one agent and another), and laws (rules on agent's communication coordination). Media like a monitors, channels, or more complex types (e.g. pipelines). Laws identify the interactive nature of the agents such as its synchronization or type of naming schemes [40]. An agent is capable of autonomous actions to meet its design objectives [43].

Application Type

The application classification includes middleware that work specifically with an application. In what follows, middleware subcategories with relevant examples are briefly described.

- Data Access Middleware: in DAM there is interaction of the application with local and/or remote databases (legacy, relational and non-relational), data warehouses, or other data source files. This middleware can:
  - include transaction processing (TP) monitor, database gateways, and distributed transaction-processing middleware.
  - assist with special database requirements such as security (authentication, confidentiality, and access control), protection, and the ACID properties [29].
  - modify the appearance of the returned data in a format that makes the data easier to use by the application or the user [51].
  - query the databases directly or communicate with the DBMS [74].
- Desktop Middleware: It is limited to the presentation of the data as required by a user for the monitoring and assistance applications, manage transport services (such as terminal emulation, file transfer, printing services), and to provide backup protection [4]. Additional services include

graphical desktop middleware management, sorting, character and string manipulation, records and file management, directory services, database information management, thread management, process planning, services event notification, management of software installation, cryptographic services and access control [23].

- Web-based Middleware: It assists the user during browsing [5]. It provides authentication service for a large number of applications [4] and interprocess communication that is independent from underlying OS, network protocols, and hardware platforms [3]. Some core services provided by web-based middleware include directory services, e-mail, billing, large-scale supply management, remote data access (and remote applications)
  - e-Commerce: One the subcategories of the web-based is e-Commerce, which pertains to the communication between two or more businesses and patrons performed over the web. This middleware controls access to customer profile information, allows the operation of business functions such as purchasing and selling items, and assists in the trade of financial information between applications. This business middleware can provide a modular platform to build the web applications [99].
  - Mobile or Wireless: Mobile or Wireless middleware is the other main subdivision of this web middleware. It integrates distributed applications and servers without permanent connections to the web. It provides mobile users secure, wireless access to e-mail, calendars, contact information, task lists, etc. [1].
- Real Time Middleware: Real-time is based on the reception of correct data on time [88]. The real-time middleware supports time sensitive request and scheduling policies. It is used for improve user applications. It can be divided into the different applications using them (real-time database application, sensor processing, and information passing). Strengths of time-dependent middleware are that they provide a decision process that determines the best approach for solving time-sensitive procedures and they can assist the operating system in the allocation of resources to aid time-sensitive applications to meet their deadlines [46].
  - Multimedia: Multimedia middleware is a subcategory of Real-time and can reliably handles a variety of data types, like speech, images, natural language processors , music, and video. The data need to be collected, integrated, and then delivered to the user in a time sensitive manner [79].
- Specialty Middleware: The middleware that require specific needs fall into this category, such as the Multi-Campus system and medical middleware [2].

4.2.2 E-Health Domain

In the current ICT scenario, clinical data produced by health care systems, diagnostic devices, sensors, IoT devices, user-generated contents, and mobile computing are of considerable amount, and comply with all the characteristics of Big Data: volume, variety, veracity, and velocity. The most of sources and techniques for Big Data in Healthcare are structured EHR data, unstructured clinical notes, medical imaging data, genetic data, epidemiology and behavioral data. The main big data challenges in e-health are the knowledge inference and extraction from complex heterogeneous patient sources, the unstructured clinical notes understanding and reasoning, the efficient management of large volumes of medical imaging data, the useful information and biomarkers extraction for available data, the genomic data analysis, the smart layering and combination of genomic data and standard clinical data, the patient's behavioral data gathering through several sensors and devices. The huge availability of data requires new tools and techniques to collect, process and distribute this enormous amount of information and knowledge. However, a very important role is covered by the middleware in the processing stack. Big Data middleware should present some innovative features for Big Data management, processing, and analysis. Big data creates a huge quantity of data and an unpredictable data flow that can quickly overwhelm existing infrastructures. Infrastructure scalability must cover storage, processing power, and middleware. However, infrastructure needs to scale without sacrificing reliability to avoid to spiral out of control in terms of robustness, rack space, power consumption and complexity. The flood of information in Big Data applications can be hard to handle, especially by consumers that are linked by slower networks, can't keep up with bursts of data, or only connect periodically to retrieve updates. Middleware must be able to handle the delayed delivery of messages, services and resources to slow users and processes without affecting real-time users and processes. Big data are generated and consumed by thousands of applications and users around the world. The middleware must be able to efficiently distribute real-time data globally via wide area networks, the Internet and mobile applications. Every infrastructure to keep up with growing data volumes and changing requirements leads to a complicated and fragile system. The middleware should be based on a simple, stable architecture that's easy to adapt as user need evolve. Finally, the infrastructure needs to automatically recover from a wide range of system and datacenter-level fault, since Big data applications demand for 24x7x365 availability. In what follows, some relevant works on middlewarefor e-health are briefly described. In [81], the SOA architecture is applied on e-health model using six main components responsible for defining interactions among the consumers and health services, where monitoring system collects a range of patient healthcare measurement data, and the health monitoring system (one of several) focuses on a particular aspect of patient care, such as pregnancy, cancer, heart disease, and others. In [78], authors address smart objects (such as physical or virtual objects that can be identified, addressable, controlled, and monitored via Internet) in the e-health scenario, allowing the continuous monitoring of vital signs of patients. The EcoHealth middleware platform integrates heterogeneous body sensors enabling the remote monitoring of patients and improving medical diagnosis. In [61], authors deal with the broad variety of medical sensors that need to be connected and integrated in order to get a holistic view of a patients general condition: proprietary solutions and custom-build software tools for data analysis are issues that prevent standardization. They propose a data transformation middleware that provides consistent access to streams of medical data by exploiting a common nomenclature and data format based on the x73 family of standards in order to aggregate devices to fit the requirements found in mobile e-health environments. The SOA based framework of [22] allows a seamless integration of different technologies, applications, and services. It also integrates mobile technologies to smoothly collect and communicate vital data from a patient's wearable biosensors for monitoring of Chronic Diseases. In order to exchange healthcare information reliability, the authors of [87] exploits a cloud based SOA suite for electronic health records (EHR) empowering the transparent, extensible, protected methods for distributing real and secured information only for intended recipients. In [109] authors describe current developments in the fields of sensing, networking, and machine learning, with an aim of underpinning the vision of the sensor platform for healthcare in a residential environment (SPHERE) project, where a generic platform fuses complementary sensor data in order to generate rich datasets in support to detection and management of various health conditions. What is really missing, especially in the big-data era, it is the ability to manage huge amounts of worldwide health data. Storage, management, data access protocols, data mining and machine-learning methodologies, that allow the correlation of information enabling decision support, discovery of new interactions, or just to manage the health information as a whole, are still missing. The framework in [83] is based on a service-oriented message-oriented middleware (âĂIJSAI middlewareâĂİ) easing the development of context-aware and information integration applications. describe a case study in a application for patient conditions monitoring, alarm detection and policy-based handling. In [21] authors deal with ageing related diseases using a middleware that provides a valid alternative to IoT solutions. A instant messaging protocol (XMPP) to guarantee a (near) real-time, secure and reliable communication channel among heterogeneous devices, realizing a remote body movement monitoring system, aimed at classify daily patients' activities, designed as a modular architecture and deployed in a large scale scenario. In [27], the author presented a prototype of the lightweight middleware for an e-Health WSN based system. The framework would govern the interaction between the applications and the network and it must be capable of sensing, processing, and communicating vital health signs. The paper proposed in [63] presents an agent-based content adaptation middleware for use in a mobile e-Health (mHealth) system. The middleware was developed by utilizing an object-oriented database version of the Wireless Universal Resource FiLe (WUR-FL), an Application Programming Interface (API). It was integrated into a cloud system. The paper proposed in [64] presented a personalized middleware for mobile eHealth (mHealth) services for using it in Ubiquitous Consumer Wireless World (UCWW). The developed middleware was based on the ISO/IEEE 11073 personal health data (PHD) standards. It works as a multi-agent system (MAS) to provide intelligent collection of physiological data from medical sensors attached to human body. Successively, it sends gathered data to a log data node by utilizing the Always Best Connected and best Served (ABC&S) communication paradigm. In[36], authors presented some feasible e-health application scenarios based on a Wireless Sensor Networks WSN. The first application was for firemen/women monitoring, while the other one was for sports performance. The network is organized in a hierarchical way according to our semantic middleware. In [68], authors presented an approach for e-health systems that enables PnP integration of medical devices and can be reconfigured dynamically during runtime. An aggregation middleware that dynamically integrates medical devices by reloading device architecture allows for devices sharing among different networks and operators, which were defined such as the medical device cloud. The paper proposed in [30] presented a middleware solution to secure data and network in the e-healthcare system. Authors show an efficient and cost effective approach and the proposed middleware solution is discussed for the delivery of secure services. The paper proposed in [86] presented a middleware for separating the information from affability management, client discovery and transit of database. In [69], authors presented a middleware solution for the integration of medical devices and the aggregation of resulting data streams, that is able to adapt itself to the requirements of patients and Care Delivery Operators, using a modular approach and external knowledge repositories. In [38], authors showed a single step health data integration solution where users can select the data sources and the data elements from multiple sources, and our platform performs the data standardization and data integration to prepare an integrated dataset. A key aspect of the architecture is data standardization, where we have used SNOMED-CT as a terminology standard to standardize health data from multiple sources. The paper presented in [78] introduced EcoHealth (Ecosystem of Health Care Devices), a middleware platform that integrates heterogeneous body sensors for enabling the remote monitoring of patients and improving medical diagnosis. The authors proposed in [73] a medical information platform based on Hadoop, named after Medoop. It uses HDFS to store the merged CDA documents efficiently, organize the feature information in CDA documents, according to frequent business queries, and compute distributed statistic in the MapReduce paradigm. In [75], authors presented the design of the BDeHS architecture that supplies data operation management capabilities, regulatory compliance, and e-Health meaningful usages. In this paper [62] the authors present building an ambient e-health system using Hydra middleware. Hydra provides a middleware framework that facilitates developers to build efficiently scalable, embedded systems while offering web service interfaces for controlling any type of physical device. It comprises a set of sophisticated components taking care of security, device and service discovery using an overlay P2P communication network. The obtained results show that Hydra provides a good foundation for integrating heterogeneous devices. The tested tools and software components significantly ease the tasks of a device developer for pervasive environments. Finally, the table 4.1 summarizes the main features of the most available middleware for e-health applications.

4.3 Evaluation of Big Data architectures

M. Iacono

A key factor for the success in Big Data is the management of resources: these platforms use a significant and flexible amount of virtualized hardware resources to try and optimize the trade off between costs and results. The management of such a quantity of resources is definitely a challenge.

Modeling Big Data oriented platforms presents new challenges, due to a number of factors: complexity, scale, heterogeneity, hard predictability. Complexity is inner in their architecture: computing nodes, storage subsystem, networking infrastructure, data management layer, scheduling, power issues, dependability issues, virtualization all concur in interactions and mutual influences. Scale is a need posed by the nature of the target problems: data dimensions largely exceed conventional storage units, the level of parallelism needed to perform computation within useful deadlines is high, obtaining final results requires the aggregation of large numbers of partial results. Heterogeneity is a technological need: evolvability, extensibility and maintainability of the hardware layer imply that the system will be partially integrated, replaced or extended by means of new parts, according to the availability on the market and the evolution of technology. Hard predictability results from the previous three factors, the nature of computation and the overall behavior and resilience of the system when running the target application and all the rest of the workload, and from the fact that both simulation, if accurate, and analytical models are pushed to the limits by the combined effect of complexity, scale and heterogeneity.

The value of performance modeling is in its power to enable informed decisions by means of developers and administrators. The possibility of predicting the performances of the system helps in better managing it, and allows to reach and keep a significant level of efficiency. This is viable if proper models are available, that benefit of information about the system and its behaviors and reduce the time and effort required for an empirical approach to management and administration of a complex, dynamic set of resources that are behind Big Data architectures.

The inherent complexity of such architectures and of their dynamics translates into the non triviality of choices and decisions in the modeling process: the same complexity characterizes models as well, and this impacts on the number of suitable formalisms, techniques, and even tools, if the goal is to obtain a sound, comprehensive modeling approach, encompassing all the (coupled) aspects of the system. Specialized approaches are needed to face the challenge, with respect to common computer systems, in particular because of the scale. Even if Big Data computing is characterized by regular, quite structured workloads, the interactions of the substanding hardware-software layers and the concurrency of different workloads have to be taken into account. In fact, applications potentially spawn hundreds (or even more) cooperating processes across a set of virtual machines, hosted on hundreds of shared physical computing nodes providing locally and less locally [84] [49] distributed resources, with different functional and non functional requirements: the same abstractions that simplify and enable the execution of Big Data applications complicate and modeling problem. The traditional system logging practices are potentially

Table 4.1 Main features of most common middleware for e-health domain

Reference	MOM	Reconfigurability (P&P)	BAN	WSN & Mobile Devices	Data Aggregation & Standandization	Security Policies	HDFS & Hadoop	Application Context
Bazzani et al. [21]	x		x	x				classify the movements of patients who received rehabilitation
Omar et al. [81]	x		x		x	x		e-health monitoring system that uses an SOA as a model for deploying, discovering, integrating, implementing, managing and invoking e-health services
Maia et al. [78]	x		x					integrates heterogeneous body sensors for enabling the remote monitoring of patients and improving medical diagnosis
Ivanov et al. [61]		x		x	x			data transformation middleware that provides consistent access to streams of medical data by exploiting a common nomenclature and data format based on the x73 family of standards
Benharref et al. [22]	x		x	x	x			framework to collect patients' data in real time, perform appropriate non-intrusive monitoring, and propose medical and/or life style engagements, whenever needed and appropriate
Rapalli et al. [87]					x	x		cloud based SOA suite for EHR integration by empowering the transparent, extensible, protected methods for distributing real and secured information only for intended recipients
Zhu et al. [109]			x		x			multimodality sensing system as an infrastructure platform fully integrated, at design stage, with intelligent data processing algorithms driving the data collection
Paganelli et al. [83]	x	x						patient conditions monitoring, alarm detection and policy-based handling
Boulmalf et al. [27]			x	x				patient's drug administration and remote supervision
Ji et al. [63] [64]				x			x	content adaptation middleware
Castillejo et al. [36]			x	x				firemen/women monitoring - sports performance
Kliem et al. [68]		x		x	x	x		PnP integration of medical devices dynamically reconfigured during runtime
Bruce et al. [30]						x		data and network security over e-Healthcare system sing medical sensor networks
Potdar et al. [86]	x			x		x		provide the healthy environment to the patient
Kliem et al. [69]		x		x	x			integration of medical devices and the aggregation of resulting data streams, using a modular approach and external knowledge repositories
Cha et al. [38]					x		x	data standardization, data integration, data selection, data analysis and data visualization
Lijun et al. [73]					x		x	construct a comprehensive platform for health information storage and exchange, utilizing the components in Hadoop ecosys-

themselves, on such a scale, Big Data problems, that in turn require significant effort for an analysis. The system as a whole has to be considered, as in a massively parallel environment many interactions may affect the dynamics, and some computations may lose value if not completed in a timely manner.

Performance data and models may also affect the costs of the infrastructure. A precise knowledge of the dynamics of the system may enable the management and planning of maintenance and power distribution, as the wear and the required power of the components is affected by their use profile.

Some introductory discussions to the issues related to performance and dependability modeling of big computing infrastructures can be found in [35][18][34][37] and [105][106][19]. More specifically, several approaches are documented in literature for performance evaluation, with contributions by studies on large-scale cloud- or grid-based Big Data processing systems. They can roughly classified into monitoring focused and modeling focused, and may be used in combination for the definition of a comprehensive modeling strategy to support planning, management, decisions and administration. There is a wide spectrum of different methodological points of view to the problem, that include classical simulations, diagnostic campaigns, use and demand profiling or characterization for different kinds of resources, predictive methods for system behavioral patterns.

4.3.1 Monitoring focused approaches

In this category some works are reported that are mainly based on an extension, or redesign, or evolution of classical monitoring or benchmarking techniques, that are used on existing systems to investigate their current behavior and the actual workloads and management problems. This can be viewed as an empirical approach, that builds predictions onto similarity and regularity assumptions, and basically postulates models by means of perturbative methods over historical data, or by assuming that knowledge about real or synthetic applications can be used, by means of a generalization process, to predict the behaviors of higher scale applications or of composed applications, and of the architecture that supports them. In general, the regularity of workloads may support in principle the likelihood of such hypotheses, specially in application fields in which the algorithms and the characterization of data are well known and runs tend to be regular and similar to each other. The main limits of this approach, that is widely and successfully adopted in conventional systems and architectures, is in the fact that for more variable applications and concurrent heterogeneous workloads the needed scale for experiments and the test scenarios are very difficult to manage, and the cost itself of running experiments or tests can be very high, as it requires an expensive system to be diverted from real use, practically resulting in a non negligible downtime from the point of view of productivity. Moreover, additional costs are caused by the need for an accurate design and planning of the tests, that are not easily repeatable for cost matters: the scale is of the order of thousands of computing nodes and petabytes of data exchanged between the nodes by means of high speed networks with articulated access patterns.

Two significant examples of system performances prediction approaches that represent this category are presented in [45] and [108]. In both cases, the prediction technique is based on the definition of test campaigns that aim at obtaining some well chosen performance measurements. As I/O is a very critical issue, specialized approaches have been developed to predict the effects of I/O over general application performances: an example is provided in [80], that assumes the realistic case of an implementation of Big Data applications in a Cloud. In this case, the benchmarking strategy is implemented in the form of a training phase that collects information about applications and system scale to tune a prediction system. Another approach that presents interesting results and privileges storage performance analysis is given in [90], that offers a benchmarking solution for cloud-based data management in the most popular environments.

Log mining is also an important resource, that extracts value from an already existing asset. The value obviously depends on the goals of the mining process and on the skills available to enact a proper management and abstraction of an extended, possibly heterogeneous harvest of fine grain measures or events tracking. Some examples of log mining based approaches are given in Chukwa [28], Kahuna [96] and Artemis [41]. Being this category of solutions founded onto technical details, these approaches are bound to specific technological solutions (or different layers od a same technological stack), such as Hadoop or Dryad: for instance, [66] presents an analysis of real logs from a Hadoop based system that is composed of 400 computing nodes, while [54][104] offers data from Google cloud backend infrastructures.

4.3.2 Simulation focused approaches

Simulation based approaches and analytical approaches are based on a previous knowledge or on reasonable hypotheses about the nature and the inner behaviors of a system, instead of inductive reasoning or generalization from measurements. Targeted measurements (on the system, if existing, or on similar systems, if not existing yet) are anyway used to tune the parameters and to verify the goodness of the models.

While simulation (e.g. event based simulation) offers in general the advantage of allowing great flexibility, with a sufficient number of simulation runs to include stochastic effects and reach a sufficient confidence level, and eventually by means of parallel simulation or simplifications, the scale of Big Data architectures is still a main challenge. The number of components to be modeled and simulated is huge, consequently the design and the setup of a comprehensive simulation in a Big Data scenario are very complex and expensive, and become a software engineering problem. Moreover, being the number of interactions and possible variations huge as well, the simulation time that is needed to get satisfactory results can be unacceptable and not fit to support timely decision making. This is generally bypassed by a tradeoff between the degree of realism, or the generality, or the coverage of the model and simulation time. Simulation is anyway considered a more viable alternative to very complex experiments, because it has more economic experimental setup costs and a faster implementation.

Literature is rich of simulation proposals, specially borrowed from the Cloud Computing field. In the following, only Big Data specific literature is sampled.

Some simulators focus on specific infrastructures or paradigms: Map-Reduce performances simulators are presented in [31], focusing on scheduling algorithms on given MapReduce workloads, or provided by non workload-aware simulators such as SimMapReduce [98] [97], MRSim [53], HSim [77][76], or Starfish [55][56] what-if engine. These simulators do not consider the effects of concurrent applications on the system. MRPerf [101] is a simulator specialized in scenarios with Map-Reduce on Hadoop; X-Trace [50] is also tailored on Hadoop and improves its fitness by instrumenting it to gather specific information. Another interesting proposal is Chukwa [28]. An example of simulation experience specific for Microsoft based Big Data applications is in [7], in which a real case study based on real logs collected on large scale Microsoft platforms.

To understand the importance of the workload interference effects, specially in cloud architectures, for a proper performance evaluation, the reader can refer to [39], that proposes a synthetic workload generator for map-reduce applications.

4.3.3 Analytical models focused approaches

The definition of proper comprehensive analytical models for Big Data systems suffers as well the scale. Classical state space based techniques (such as Petri nets based approaches) generate huge state spaces, that are non treatable in the solution phase, if not exploiting (or forcing) symmetries, reductions, strong assumption or narrow aspects of the problem. In general, a faithful modeling requires an enormous number of variables (and equations), that is hardly manageable if not with analogous reductions or with the support of tools, or by having a hierarchical modeling method, based on overall simplified models that use the results of small, partial models to compensate approximations.

Literature proposes different analytical techniques, sometimes focused on part of the architecture.

As the network is a limiting factor in modern massively distributed systems, data transfers have been targeted in order to get traffic profiles over interconnection networks. Some realistic Big Data applications have been studied in [94], that points out communication modeling as foundation on which more complete performance models can be developed. Similarly [85] founds the analysis on communication patterns, that are shaped by means of hardware support to obtain sound parameters over time.

A classical mathematical analytical description is chosen in [89] and in [10] and [11], in which "Resource Usage Equations" are developed to take into account the influence on performances of large datasets in different scenarios. Similarly, [32] presents a rich analytical framework suitable for performance prediction in scientific applications. Other sound examples of predictive analytical model dedicated to large scale applications is in [67], that presents the SAGE case study, and [42], that focus on load performance prediction.

An interesting approximate approach, suitable for the generation of analytical stochastic models for systems with a very high number of components, is presented, in various applications related to Big Data, in [33][19][17][14][34][18][35][37][20]. The authors deal with different aspects of Big Data architectures by applying Mean Field Analysis and Markovian agents, exploiting the property of these methods to exploit symmetry to obtain a better approximation as much as the number of components grows. This can be also seen as a compositional approach, i.e. an approach in which complex analytical models can be obtained by proper compositions of simpler model according to certain given rules. An example is in [9] that deals with performance scaling analysis of distributed data-intensive web applications. Multiformalism approaches, such as [13], [15] and [16], can also fall in this category.

Within the category of analytical techniques we finally include two diverse approaches, that are not based on classical dynamical equations or variations. In [26] workload performances is derived by means of a black box approach, that observes a system to obtain, by means of regression trees, suitable model parameters from samples of its actual dynamics, updating them at major changes. In [8] resource bottlenecks are used to understand and optimize data movements and execution time with a shortest needed time logic, with the aim of obtaining optimistic performance models for MapReduce applications that have been proven effective in assessing the Google and Hadoop MapReduce implementations.

References

[1] (????) Fenestrae mobile data server. URL http://yahoo.bitpipe.com, (Date last accessed 26-April-2016)
[2] (????) Internet2. URL http://middleware.internet2.edu, (Date last accessed 26-April-2016)
[3] (????) It pro. URL http://www.itprodownloads.com, (Date last accessed 26-April-2016)

[4] (????) Knowledge storm. URL `http://www.knowledgestorm.com/`, (Date last accessed 26-April-2016)
[5] (????) Mit media lab: Software agents. URL `http://agents.media.mit.edu/projects`, (Date last accessed 26-April-2016)
[6] (????) Webopedia. URL `http://www.webopedia.com`, (Date last accessed 26-April-2016)
[7] Ananthanarayanan G, Kandula S, Greenberg A, Stoica I, Lu Y, Saha B, Harris E (2010) Reining in the outliers in map-reduce clusters using mantri. In: Proceedings of the 9th USENIX conference on Operating systems design and implementation, USENIX Association, pp 1–16
[8] Anderson E, Ganger GR, Wylie JJ, Krevat E, Shiran T, Tucek J (2010) Applying Performance Models to Understand Data-Intensive Computing Efficiency
[9] Andresen D, Yang T, Ibarra OH, Eğecioğlu Ö (1998) Adaptive partitioning and scheduling for enhancing www application performance. Journal of Parallel and Distributed Computing 49(1):57–85
[10] Armstrong B, Eigenmann R (1997) Performance forecasting: Characterization of applications on current and future architectures. Purdue Univ School of ECE, High-Performance Computing Lab Technical report ECE-HPCLab-97202
[11] Armstrong B, Eigenmann R (1998) Performance forecasting: Towards a methodology for characterizing large computational applications. In: Parallel Processing, 1998. Proceedings. 1998 International Conference on, IEEE, pp 518–525
[12] Aversa R, Di Martino B, Rak M, Venticinque S (2010) Cloud agency: A mobile agent based cloud system. In: Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on, IEEE, pp 132–137
[13] Barbierato E, Gribaudo M, Iacono M (2013) A Performance Modeling Language For Big Data Architecture. In: 27th European conference on modelling and simulation (ECMS 2013). IEEE Soc. German Chapter, DOI 10.7148/2013-0511
[14] Barbierato E, Gribaudo M, Iacono M (2013) Modeling apache hive based applications in big data architectures. In: Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, ValueTools '13, pp 30–38, DOI 10.4108/icst.valuetools.2013.254398
[15] Barbierato E, Gribaudo M, Iacono M (2013) Modeling Apache Hive based applications in Big Data architectures. In: 7th International Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS 2013, DOI 10.7148/2013-0511
[16] Barbierato E, Gribaudo M, Iacono M (2013) Performance Evaluation of NoSQL Big-Data applications using multi-formalism models. Future Generation Computer Systems to appear, DOI http://dx.doi.org/10.1016/j.future.2013.12.036
[17] Barbierato E, Gribaudo M, Iacono M (2013) A performance modeling language for big data architectures. In: Rekdalsbakken W, Bye RT, Zhang H (eds) ECMS, European Council for Modeling and Simulation, pp 511–517
[18] Barbierato E, Gribaudo M, Iacono M (2014) Performance evaluation of NoSQL big-data applications using multi-formalism models. Future Generation Computer Systems 37(0):345–353, DOI http://dx.doi.org/10.1016/j.future.2013.12.036
[19] Barbierato E, Gribaudo M, Iacono M (2015 (to appear)) Modeling and evaluating the effects of big data storage resource allocation in global scale cloud architectures. International Journal of Data Warehousing and Mining

[20] Barbierato E, Gribaudo M, Iacono M (2016) Modeling and evaluating the effects of Big Data storage resource allocation in global scale cloud architectures. International Journal of Data Warehousing and Mining 12(2):1–20, DOI 10.4018/IJDWM.2016040101
[21] Bazzani M, Conzon D, Scalera A, Spirito MA, Trainito CI (2012) Enabling the iot paradigm in e-health solutions through the virtus middleware. In: Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on, IEEE, pp 1954–1959
[22] Benharref A, Serhani MA (2014) Novel cloud and soa-based framework for e-health monitoring using wireless biosensors. Biomedical and Health Informatics, IEEE Journal of 18(1):46–55
[23] Bernstein PA (1996) Middleware: a model for distributed system services. Communications of the ACM 39(2):86–98
[24] Bishop TA, Karne RK (2003) A survey of middleware. In: Computers and Their Applications, pp 254–258
[25] Blair GS, Blair L, Issarny V, Tuma P, Zarras A (2000) The role of software architecture in constraining adaptation in component-based middleware platforms. In: Middleware 2000, Springer, pp 164–184
[26] Bodík P, Griffith R, Sutton C, Fox A, Jordan M, Patterson D (2009) Statistical Machine Learning Makes Automatic Control Practical for Internet Datacenters. In: Proceedings of the 2009 conference on Hot topics in cloud computing, USENIX Association, Berkeley, CA, USA, HotCloud'09
[27] Boulmalf M, Belgana A, Sadiki T, Hussein S, Aouam T, Harroud H (2012) A lightweight middleware for an e-health wsn based system using android technology. In: Multimedia Computing and Systems (ICMCS), 2012 International Conference on, IEEE, pp 551–556
[28] Boulon J, Konwinski A, Qi R, Rabkin A, Yang E, Yang M (2008) Chukwa, a large-scale monitoring system. In: Proceedings of CCA, vol 8
[29] Britton C, Bye P (2004) IT architectures and middleware: strategies for building large, integrated systems. Pearson Education
[30] Bruce N, Sain M, Lee HJ (2014) A support middleware solution for e-healthcare system security. In: Advanced Communication Technology (ICACT), 2014 16th International Conference on, IEEE, pp 44–47
[31] Cardona K, Secretan J, Georgiopoulos M, Anagnostopoulos G (2007) A grid based system for data mining using MapReduce. Tech. rep., Technical Report TR-2007-02, AMALTHEA
[32] Carrington L, Snavely A, Gao X, Wolter N (2003) A performance prediction framework for scientific applications. Computational Scienceï£ · ICCS 2003 pp 701–701
[33] Castiglione A, Gribaudo M, Iacono M, Palmieri F (2013) Exploiting mean field analysis to model performances of big data architectures. Future Generation Computer Systems In press(0):–, DOI 10.1016/j.future.2013.07.016
[34] Castiglione A, Gribaudo M, Iacono M, Palmieri F (2014) Exploiting mean field analysis to model performances of big data architectures. Future Generation Computer Systems 37(0):203–211, DOI http://dx.doi.org/10.1016/j.future.2013.07.016
[35] Castiglione A, Gribaudo M, Iacono M, Palmieri F (2014) Modeling performances of concurrent big data applications. Software: Practice and Experience pp n/a–n/a, DOI 10.1002/spe.2269
[36] Castillejo P, Martinez JF, Rodriguez-Molina J, Cuerva A (2013) Integration of wearable devices in a wireless sensor network for an e-health application. Wireless Communications, IEEE 20(4):38–49
[37] Cerotti D, Gribaudo M, Iacono M, Piazzolla P (2015) Modeling and analysis of performances for concurrent multithread applications on multicore and graphics processing unit systems. Concurrency and Computation: Practice and Experience pp n/a–n/a, DOI 10.1002/cpe.3504

[38] Cha S, Abusharekh A, Abidi Sr S (2015) Towards a'big'health data analytics platform. In: Big Data Computing Service and Applications (BigDataService), 2015 IEEE First International Conference on, IEEE, pp 233–241
[39] Chen Y, Ganapathi AS, Griffith R, Katz RH (2010) Towards Understanding Cloud Performance Tradeoffs Using Statistical Workload Analysis and Replay. Tech. Rep. UCB/EECS-2010-81, EECS Department, University of California, Berkeley, URL http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-81.html
[40] Ciancarini P (1996) Coordination models and languages as software integrators. ACM Computing Surveys (CSUR) 28(2):300–302
[41] Crețu-Ciocârlie GF, Budiu M, Goldszmidt M (2008) Hunting for problems with artemis. In: Proceedings of the First USENIX conference on Analysis of system logs, USENIX Association, pp 2–2
[42] Dinda PA, O'Hallaron DR (1999) An evaluation of linear models for host load prediction. In: High Performance Distributed Computing, 1999. Proceedings. The Eighth International Symposium on, IEEE, pp 87–96
[43] Ding Y, Malaka R, Kray C, Schillo M (2001) Raja: a resource-adaptive java agent infrastructure. In: Proceedings of the fifth international conference on Autonomous agents, ACM, pp 332–339
[44] Domnori E, Cabri G, Leonardi L (2011) Multi-agent approach for disaster management. In: P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2011 International Conference on, IEEE, pp 311–316
[45] Duan S, Thummala V, Babu S (2009) Tuning Database Configuration Parameters with iTuned. Proc VLDB Endow 2(1):1246–1257
[46] Duran HA, Blair GS (2000) A resource management framework for adaptive middleware. In: Object-Oriented Real-Time Distributed Computing, 2000.(ISORC 2000) Proceedings. Third IEEE International Symposium on, IEEE, pp 206–209
[47] Emmerich W (1999) Distributed objects. In: Proceedings of the Conference on The future of Software engineering
[48] Emmerich W (2000) Software engineering and middleware: a roadmap. In: Proceedings of the Conference on The future of Software engineering, ACM, pp 117–129
[49] Esposito C, Ficco M, Palmieri F, Castiglione A (2013) Interconnecting Federated Clouds by Using Publish-Subscribe Service. Cluster Computing 16(4):887–903, DOI 10.1007/s10586-013-0261-z
[50] Fonseca R, Porter G, Katz RH, Shenker S, Stoica I (2007) X-trace: A Pervasive Network Tracing Framework. In: Proceedings of the 4th USENIX Conference on Networked Systems Design & Implementation, USENIX Association, Berkeley, CA, USA, NSDI'07, pp 20–20
[51] Fraternali P (1999) Tools and approaches for developing data-intensive web applications: a survey. ACM Computing Surveys (CSUR) 31(3):227–263
[52] Ghijsen M, Jansweijer W, Wielinga B (2008) Towards a framework for agent coordination and reorganization, agentcore. In: Coordination, Organizations, Institutions, and Norms in Agent Systems III, Springer, pp 1–14
[53] Hammoud S, Li M, Liu Y, Alham NK, Liu Z (2010) Mrsim: A discrete event based mapreduce simulator. In: Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on, IEEE, vol 6, pp 2993–2997
[54] Hellerstein J (2010) Google cluster data. URL http://googleresearch.blogspot.com/2010/01/google-cluster-data.html
[55] Herodotou H, Babu S (2011) Profiling, what-if analysis, and cost-based optimization of MapReduce programs. Proc of the VLDB Endowment 4(11):1111–1122

[56] Herodotou H, Lim H, Luo G, Borisov N, Dong L, Cetin FB, Babu S (2011) Starfish: A self-tuning system for big data analytics. In: Proc. of the Fifth CIDR Conf

[57] Hübner JF, Sichman J, Boissier O (????) Using the moise+ model for a cooperative framework of mas reorganization. Advances in Artificial Intelligence, SBIA 2004 pp 481–517

[58] Hübner JF, Sichman JS, Boissier O (2002) A model for the structural, functional, and deontic specification of organizations in multiagent systems. In: Advances in artificial intelligence, Springer, pp 118–128

[59] Hubner JF, Sichman JS, Boissier O (2007) Developing organised multiagent systems using the moise+ model: programming issues at the system and agent levels. International Journal of Agent-Oriented Software Engineering 1(3-4):370–395

[60] Hübner JF, Boissier O, Kitio R, Ricci A (2010) Instrumenting multi-agent organisations with organisational artifacts and agents. Autonomous Agents and Multi-Agent Systems 20(3):369–400

[61] Ivanov D, Kliem A, Kao O (2013) Transformation middleware for heterogeneous healthcare data in mobile e-health environments. In: Proceedings of the 2013 IEEE Second International Conference on Mobile Services, IEEE Computer Society, pp 39–46

[62] Jahn M, Pramudianto F, Al-Akkad A, Brinkmann A (2009) Hydra middleware for developing pervasive systems: A case study in ehealth domain. In: We are pleased to present the Workshop Proceedings of the 32nd Annual Con-ference on Artificial Intelligence (KI 2009), which is held on September 15-18 in Paderborn. This year the volume includes papers or abstracts of ten workshops: 3rd Workshop on Behavior Monitoring and Interpretation-Well Being, Complex, Citeseer, p 80

[63] Ji Z, Zhang X, Ganchev I, O'Droma M (2012) A content adaptation middleware for use in a mhealth system. In: e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on, IEEE, pp 455–457

[64] Ji Z, Zhang X, Ganchev I, O'Droma M (2012) A personalized middleware for ubiquitous mhealth services. In: e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on, IEEE, pp 474–476

[65] Jung DG, Paek KJ, Kim TY (1999) Design of mobile mom: Message oriented middleware service for mobile computing. In: Parallel Processing, 1999. Proceedings. 1999 International Workshops on, IEEE, pp 434–439

[66] Kavulya S, Tan J, Gandhi R, Narasimhan P (2010) An analysis of traces from a production mapreduce cluster. In: Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on, IEEE, pp 94–103

[67] Kerbyson DJ, Alme HJ, Hoisie A, Petrini F, Wasserman HJ, Gittings M (2001) Predictive performance and scalability modeling of a large-scale application. In: Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM), ACM, pp 37–37

[68] Kliem A, Kao O (2013) Cosemed-cooperative and secure medical device cloud. In: e-Health Networking, Applications & Services (Healthcom), 2013 IEEE 15th International Conference on, IEEE, pp 260–264

[69] Kliem A, Boelke A, Grohnert A, Traeder N (2014) Self-adaptive middleware for ubiquitous medical device integration. In: e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on, IEEE, pp 298–304

[70] Kota R, Gibbins N, Jennings NR (2008) Decentralised structural adaptation in agent organisations. In: AAMAS-OAMAS, Springer, pp 54–71

[71] Kristensen MD (2010) Scavenger: Transparent development of efficient cyber foraging applications. In: Pervasive Computing and Communications (PerCom), 2010 IEEE International Conference on, IEEE, pp 217–226

[72] Lesser V, Decker K, Wagner T, Carver N, Garvey A, Horling B, Neiman D, Podorozhny R, Prasad MN, Raja A, et al (2004) Evolution of the gpgp/taems domain-independent coordination framework. Autonomous agents and multi-agent systems 9(1-2):87–143
[73] Lijun W, Yongfeng H, Ji C, Ke Z, Chunhua L (2013) Medoop: A medical information platform based on hadoop. In: e-Health Networking, Applications & Services (Healthcom), 2013 IEEE 15th International Conference on, IEEE, pp 1–6
[74] Linthicum DS (1999) Database-oriented middleware. DM
[75] Liu W, Park E (2014) Big data as an e-health service. In: Computing, Networking and Communications (ICNC), 2014 International Conference on, IEEE, pp 982–988
[76] Liu Y, Li M, Alham NK, Hammoud S (2011) Hsim: A mapreduce simulator in enabling cloud computing. Future Generation Computer Systems
[77] Liu Y, Li M, Alham NK, Hammoud S (2013) HSim: A MapReduce Simulator in Enabling Cloud Computing. Future Gener Comput Syst 29(1):300–308
[78] Maia P, Baffa A, Cavalcante E, Delicato FC, Batista T, Pires PF (2015) A middleware platform for integrating devices and developing applications in e-health. In: Computer Networks and Distributed Systems (SBRC), 2015 XXXIII Brazilian Symposium on, IEEE, pp 10–18
[79] Mills KL (1999) Introduction to the electronic symposium on computer-supported cooperative work. ACM Computing Surveys (CSUR) 31(2es):1
[80] Mytilinis I, Tsoumakos D, Kantere V, Nanos A, Koziris N (2015) I/O performance modeling for big data applications over cloud infrastructures. In: 2015 IEEE International Conference on Cloud Engineering, IC2E 2015, Tempe, AZ, USA, March 9-13, 2015, IEEE, pp 201–206, DOI 10.1109/IC2E.2015.29, URL `http://dx.doi.org/10.1109/IC2E.2015.29`
[81] Omar WM, Taleb-Bendiab A (2006) E-health support services based on service-oriented architecture. IT professional 8(2):35–41
[82] Othmane B, Hebri RSA (2012) Cloud computing & multi-agent systems: a new promising approach for distributed data mining. In: Information Technology Interfaces (ITI), Proceedings of the ITI 2012 34th International Conference on, IEEE, pp 111–116
[83] Paganelli F, Parlanti D, Giuli D (2010) A service-oriented framework for distributed heterogeneous data and system integration for continuous care networks. In: the Proc. of 7th Annual IEEE Consumer Communications and Networking Conference, Las Vegas
[84] Palmieri F, Pardi S (2010) Towards a federated Metropolitan Area Grid environment: The SCoPE network-aware infrastructure. Future Generation Computer Systems 26(8):1241–1256
[85] Papaefstathiou E, Kerbyson DJ, Nudd GR (1994) A layered approach to parallel software performance prediction: A case study Technical Report CS-RR-262
[86] Potdar MS, Manekar AS, Kadu RD (2014) Android" health-dr." application for synchronous information sharing. In: Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on, IEEE, pp 265–269
[87] Rallapalli S, Gondkar R (2016) A study on cloud based soa suite for electronic healthcare records integration. In: Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics, Springer, pp 143–150
[88] Schmidt DC (2002) Middleware for real-time and embedded systems. Communications of the ACM 45(6):43–48
[89] Schopf JM, Berman F (1998) Performance prediction in production environments. In: Parallel Processing Symposium, 1998. IPPS/SPDP 1998. Proceedings of the First Merged International... and Symposium on Parallel and Distributed Processing 1998, IEEE, pp 647–653

[90] Shi Y, Meng X, Zhao J, Hu X, Liu B, Wang H (2010) Benchmarking cloud-based data management systems. In: Proceedings of the second international workshop on Cloud data management, ACM, New York, NY, USA, CloudDB '10, pp 47–54, DOI 10.1145/1871929.1871938, URL http://doi.acm.org/10.1145/1871929.1871938

[91] Sim KM (2009) Agent-based cloud commerce. In: Industrial Engineering and Engineering Management, 2009. IEEM 2009. IEEE International Conference on, IEEE, pp 717–721

[92] Sim KM (2010) Towards complex negotiation for cloud economy. In: Advances in Grid and Pervasive Computing, Springer, pp 395–406

[93] Sim KM (2013) Complex and concurrent negotiations for multiple interrelated e-markets. Cybernetics, IEEE Transactions on 43(1):230–245

[94] Sivasubramaniam A, Singla A, Ramachandran U, Venkateswaran H (1995) On characterizing bandwidth requirements of parallel applications. In: ACM SIGMETRICS Performance Evaluation Review, ACM, vol 23, pp 198–207

[95] Staff V (2012) Virtualization overview. White Paper, http://wwwvmwarecom/pdf/virtualizationpdf

[96] Tan J, Pan X, Marinelli E, Kavulya S, Gandhi R, Narasimhan P (2010) Kahuna: Problem diagnosis for mapreduce-based cloud computing environments. In: Network Operations and Management Symposium (NOMS), 2010 IEEE, IEEE, pp 112–119

[97] Teng F, Yu L, Magoulès F (2011) Simmapreduce: a simulator for modeling mapreduce framework. In: Multimedia and Ubiquitous Engineering (MUE), 2011 5th FTRA International Conference on, IEEE, pp 277–282

[98] Teng F, Yu L, Magoulï£ · s F (2011) SimMapReduce: A Simulator for Modeling MapReduce Framework. In: Multimedia and Ubiquitous Engineering (MUE), 2011 5th FTRA International Conference on, pp 277–282

[99] Umar A, Bianchi M, Caruso F, Missier P (2000) A knowledge-based decision support workbench for advanced ecommerce. In: Research Challenges, 2000. Proceedings. Academia/Industry Working Conference on, IEEE, pp 93–100

[100] Uramoto N, Maruyama H (1999) Infobus repeater: a secure and distributed publish/subscribe middleware. In: Parallel Processing, 1999. Proceedings. 1999 International Workshops on, IEEE, pp 260–265

[101] Wang G, Butt AR, Pandey P, Gupta K (2009) A simulation approach to evaluating design decisions in MapReduce setups. In: Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009. MASCOTS'09. IEEE International Symposium on, IEEE, pp 1–11

[102] Wang Zg, Liang Xh (2006) A graph based simulation of reorganization in multi-agent systems. In: Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology, IEEE Computer Society, pp 129–132

[103] Wilczynski A, Kolodziej J (2016 (in press)) Cloud implementation of agent-based model for evacuation simulation. In: the 30th European Conference on Modelling and Simulation, Proceedings of

[104] Wilkes J (2011) More google cluster data. URL http://googleresearch.blogspot.com/2011/11/more-google-cluster-data.html

[105] Xu L, Cipar J, Krevat E, Tumanov A, Gupta N, Kozuch MA, Ganger GR (2014) Agility and performance in elastic distributed storage. Trans Storage 10(4):16:1–16:27, DOI 10.1145/2668129

[106] Yan F, Riska A, Smirni E (2012) Fast eventual consistency with performance guarantees for distributed storage. In: Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on, pp 23–28, DOI 10.1109/ICDCSW.2012.21

[107] Zhang Z, Zhang X (2009) Realization of open cloud computing federation based on mobile agent. In: Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on, IEEE, vol 3, pp 642–646

[108] Zheng W, Bianchini R, Janakiraman GJ, Santos JR, Turner Y (2009) JustRunIt: Experiment-based Management of Virtualized Data Centers. In: Proceedings of the 2009 Conference on USENIX Annual Technical Conference, USENIX Association, Berkeley, CA, USA, USENIX'09, pp 18–18

[109] Zhu N, Diethe T, Camplani M, Tao L, Burrows A, Twomey N, Kaleshi D, Mirmehdi M, Flach P, Craddock I (2015) Bridging e-health and the internet of things: The sphere project. Intelligent Systems, IEEE 30(4):39–46

Chapter 5
Big Data representation and management

Viktor Medvedev, Olga Kurasova and Ciprian Dobre

Traditional technologies and techniques for data storage and analysis are not efficient anymore, in the Big Data and Data Science era, as the data is produced in high-volumes, come with high-velocity, has high-variety and there is an imperative need for discovering valuable knowledge in order to help in decision making process. Big data bring new challenges to data mining and data visualization because large volumes and different varieties must be taken into account. The common methods and tools for data processing and analysis are unable to manage such amounts of data, even if powerful computer clusters are used. To analyse big data, many new data mining and machine learning algorithms as well as technologies have been developed. So, big data do not only yield new data types and storage mechanisms, but also new methods of analysis and visualization.

There are many challenges when dealing with big data handling, starting from data capture to data visualization. Regarding the process of data capture, sources of data are heterogeneous, geographically distributed, and unreliable, being susceptible to errors. Current real-world storage solutions such as databases are populated in consequence with inconsistent, incomplete and noisy data. Data quality is very important, especially in enterprise systems. Data mining techniques are directly affected by data quality. Poor data quality means no relevant results. The cost of devices with sensing capabilities (devices that have at least one sensor, a processing unit and a communication module) has reached the point where deploying a vast number of them is an acceptable option for monitoring larger areas. This is what drives ideas such as Wireless Sensor Networks and Internet of Things and turns them into reality. Smart Phones, Smart watches, Smart wrist bands and new wearable computing platforms are hastily becoming ubiquitous. And, all these sensors generate raw data that was previously unavailable or scarce. This data once processed can provide insights into human behavior and can help us understand the environment in which we live. In this section, we present a set of filters that can be used to minimize the amount of data needed for processing and without negatively impacting the result or the information that can be extracted from this data.

Viktor Medvedev
Vilnius University, Institute of Mathematics and Informatics, Lithuania, e-mail: `Viktor.Medvedev@mii.vu.lt`

Olga Kurasova
Vilnius University, Institute of Mathematics and Informatics, Lithuania, e-mail: `Olga.Kurasova@mii.vu.lt`

Ciprian Dobre
University Politehnica of Bucharest, Romania, e-mail: `ciprian.dobre@cs.pub.ro`

5.1 Solutions for Data Mining and Scientific Visualization

Viktor Medvedev, Olga Kurasova

In the modern world, we often deal with big data, because nowadays technologies are able to store and process data ever a larger and larger amount of data. Big data is the term for a collection of data sets so large and complex that it becomes difficult to process and analysis using traditional data processing and mining tools. Big data can be characterized by three V's: volume (large amounts of data), variety (includes different types of data), and velocity (constantly accumulating new data) [1]. Data become big data when their volume, velocity, or variety exceed the abilities of IT systems to store, analyse, and process them. Big data are not just about lots of data, they are actually a new concept providing an opportunity to find a new insight into the existing data. There are many applications of big data: business, technology, telecommunication, medicine, health care, and services, bioinformatics (genetics), science, e-commerce, finance, the Internet (information search, social networks), etc. Some sources of big data are actually new. Big data can be collected not only from computers, but also from billions of mobile phones, social media posts, different sensors installed in cars, utility meters, shipping and many other sources. In many cases, data are just being generated faster than they can be precessed and analysed.

Big data bring new challenges to data mining and data visualization because large volumes and different varieties must be taken into account [2]. The common methods and tools for data processing and analysis are unable to manage such amounts of data, even if powerful computer clusters are used. To analyse big data, many new data mining and machine learning algorithms as well as technologies have been developed. So, big data do not only yield new data types and storage mechanisms, but also new methods of analysis.

Clustering, classification problems, association rule learning and other data mining problems often arise in the modern world. There is a lot of applications in which extremely large or big data sets need to be explored, but which are much too large to be processed by traditional data mining methods.

Data and information have been collected and analyzed right through history. New technologies and Internet have boosted the ability to store, process and analyze the data. Recently, the amount of data being collected across the world has been increasing at exponential rates. Various data come from devices, sensors, networks, transactional applications, web and social media âĂŞ much of them are generated in real time and very large scale. The existing technologies and methods that are available to store and analyze data cannot work efficiently with such amount of data. That is why big data should speed-up technological progress. Big data become a major source for innovation and development. However, the huge challenges, related to the storage of big data, their analysis, visualization and interpretation, arise. Size is only one of several features of big data. Other characteristics, such as the frequency, velocity or variety, are equally important in defining big data.

Big data analytics become mainstream in the era of new technologies. It is the process of examining big data to uncover hidden and useful information for better decisions. Big data analytics assists data researchers to analyze such data which cannot be processed by conventional analysis tools. Big data analytics involves multiple analysis including visual presentation of data. The usage of visual analytics for solving big data problems brings new challenges as well as new research issues and prospects.

Data mining is an important issue in the data analysis process and knowledge discovery in medicine, economics, finance, telecommunication and other scientific fields. For this reason, for several decades, the attention has been focused on development of data mining techniques and their applications. The widely used data mining systems are desktop applications: Weka [3], Orange [4], KNIME [5], RapidMiner [6].

The systems include data pre-processing, classification, clustering, regression, association rule techniques, etc.

Recently, researches of web services have been rapidly developed. A web service is a collection of functions for use by other applications through a standard protocol. It offers a possibility of transparent integration between heterogeneous platforms and application interface [7]. Therefore data mining algorithms have been implemented as services. Only a few data mining systems are implemented as cloud-based web applications, for example, ClowdFlows [8], [9] and Dame [10].

Big data can include both unstructured and structured data. Unstructured data are the data that either do not have a pre-defined data model or are not organized in a pre-defined manner. Structured data are relatively simple and easy to analyse, because usually the data reside in databases in the form of columns and rows. The challenge for scientists is to develop tools to transform unstructured data to structured ones.

A structured data set X consists of data items $X_1, X_2, \dots, X_m$ described by the features $x_1, x_2, \dots, x_n$, where m is the number of items, n is the number of features. So, $X = \{X_1, X_2, \dots, X_m\} = \{x_{ij},\ i = 1, \dots, m,\ j = 1, \dots, n\}$, where x_{ij} is the jth feature value of the ith object. If m is large enough, then the data set is called a multidimensional big data; if n is large, then the data set is called high dimensional data. Often, it is necessary to solve variuos big data mining problems, for example, dimensionality reduction as well as visualization [11], [12].

The aim of the research is to make a comparative analysis of data mining systems suitable for data visualizationon (or dimensionality reduction), to highlight their advantages and disadvantages and to derive properties which are the most significant and useful for data mining experts in order to suggest the system based on these properties.

5.1.1 Direct Data Visualization

The direct data visualization is a graphical presentation of the data set that provides a qualitative understanding of the information contents in a natural and direct way. The commonly used methods are scatter plot matrices, parallel coordinates, Andrews curves, Chernoff faces, stars, dimensional stacking, etc. [13], [14].

The direct visualization methods do not have any defined formal mathematical criterion for estimating the visualization quality. Each of the features $x_1, x_2, \dots, x_n$ characterizing the object $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$, $i \in \{1, \dots, m\}$, is represented in a visual form acceptable for a human being.

Geometric Methods

Geometric visualization methods are the methods where multidimensional points are displayed using the axes of the selected geometric shape.

Scatter plots are one of the most commonly used techniques for data representation on a plane R^2 or space R^3. Points are displayed in the classic (x, y) or (x, y, z) format [15],[13]. Usually, the two-dimensional $(n = 2)$ or three-dimensional $(n = 3)$ points are represented by this technique. A two-dimensional example is shown in Fig. 5.1.

Using a *matrix of scatter plots*, the scatter plots can be applied to visualize more higher dimensionality data. The matrix of scatter plots is an array of scatter plots displaying all possible pairwise combinations

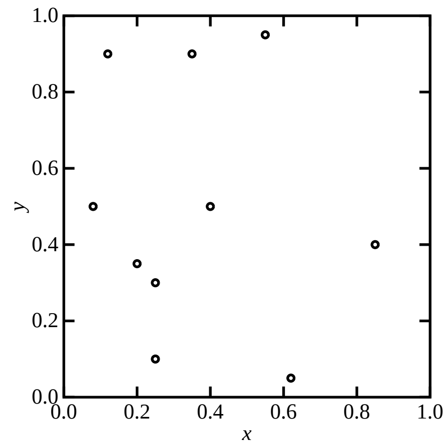

Fig. 5.1 Scatter plot of two-dimensional points

of features. If n-dimensional data are analyzed, the number of scatter plots is equal to $\frac{n(n-1)}{2}$. In the diagonal of the matrix of scatter plots, a graphical statistical characteristic of each feature can be presented, for example, a histogram of values. The matrix of scatter plots is useful for observing all possible pairwise interactions between features. The matrix of scatter plots of the Iris data is presented in Fig. 5.2. A real data set with 150 random samples of flowers from iris species setosa, versicolor, and virginica. From each species there are 50 observations of sepal length, sepal width, petal length, and petal width in cm. The iris flowers are described by 4 attributes. We can see that Setosa flowers (blue) are significantly different from Versicolor (red) and Virginica (green). The scatter plots can also be positioned in a non-array format (circular, hexagonal, etc.). Some variations of scatter plot matrices are also developed [13].

Andrews curves of the Iris data set are presented in Fig. 5.3, where different species of irises are painted in different colors [14]. An advantage of this method is that it can be used for the analysis of data of a high dimensionality n. A shortcoming is that when visualizing a large data set, i.e. with large enough m, it is difficult to comprehend and interpret the results.

Parallel coordinates as a way of visualizing multidimensional data are proposed by Inselberg [16]. In this method, coordinate axes are shown as parallel lines that represent features. An n-dimensional point is represented as $n-1$ line segments, connected to each of the parallel lines at the appropriate feature value.

The Iris data set is displayed by using parallel coordinates in Fig. 5.4 [14]. The image is obtained using the system *Orange* (`http://orange.biolab.si/`). Different colors correspond to the different species of irises. We see that the species are distinguished best by the petal length and width. It is difficult to separate the species by the sepal length and width.

The parallel coordinate method can be used for visualizing high-dimensional data. When the coordinates are dense, it is difficult to perceive the data structure. When displaying a large data set, i.e. when the number m of objects is large, the interpretation of the results is very complicated, often it is almost impossible.

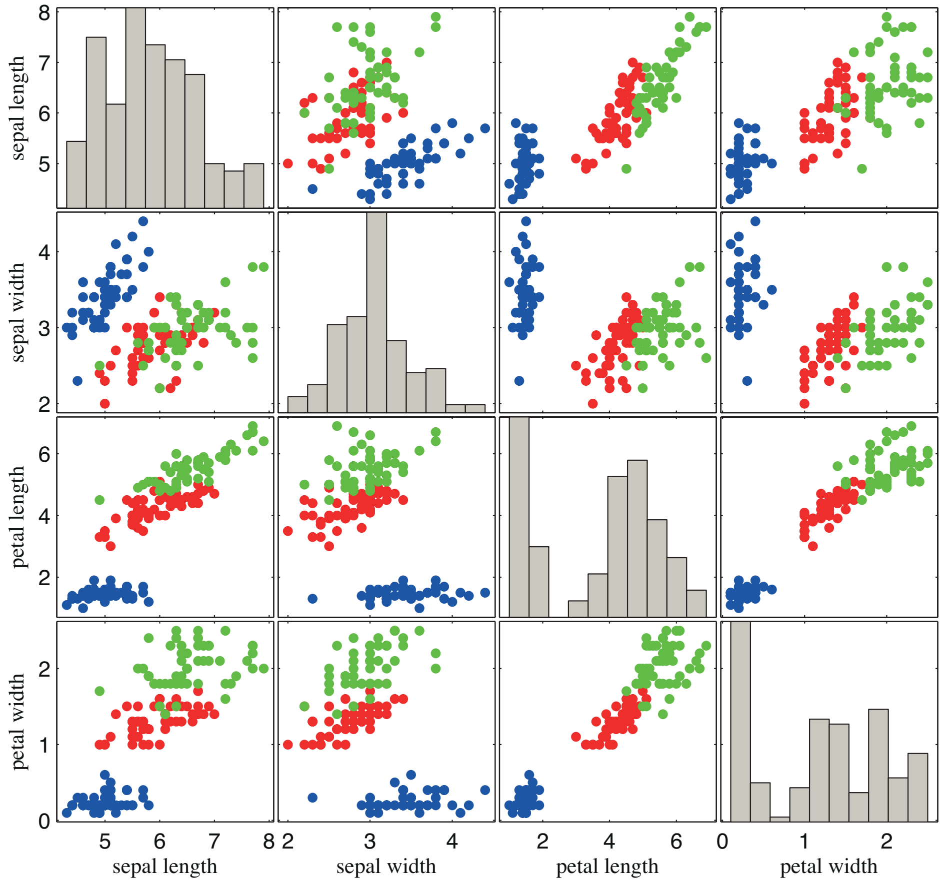

Fig. 5.2 Scatter plot matrix of the Iris data: *blue* – Setosa, *red* – Versicolor, *green* – Virginica

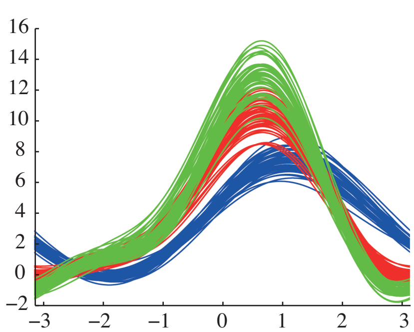

Fig. 5.3 Andrews curves of Iris data: *blue* – Setosa, *red* – Versicolor, *green* – Virginica

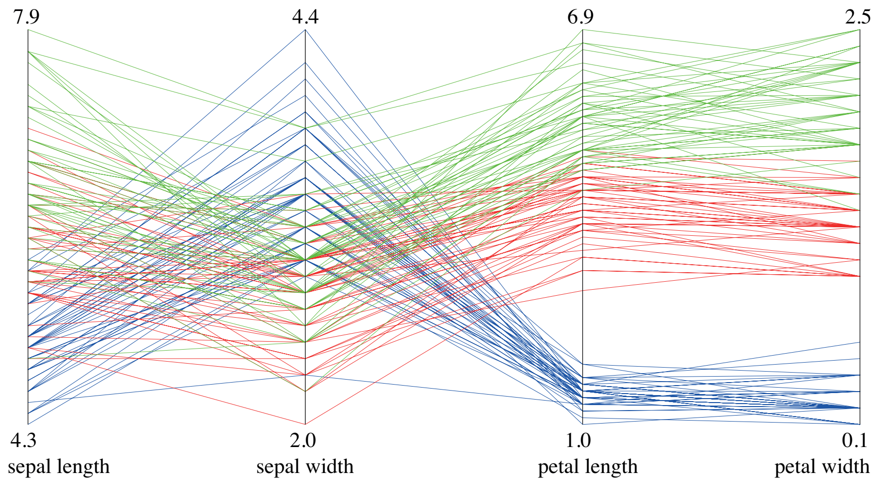

Fig. 5.4 Iris data represented on the parallel coordinates: *blue* – Setosa, *red* – Versicolor, *green* – Virginica

Iconographic Displays

The aim of visualization of multidimensional data is not only to map the data onto a two- or three-dimensional space, but also to help perceiving them. The second aim may be achieved visualizing multidimensional data by *iconographic display* methods. Each object that is defined by the n features is displayed by a glyph. Color, shape, and location of the glyph depend on the values of features. The most famous methods are *Chernoff faces* [17] and the *star* method [18].

In *Chernoff faces* [17], data features are mapped to facial features, such as the angle of eyes, the width of a nose, etc. The Iris data set visualized by Chernoff faces, are presented in Fig. 5.5, where sepal length corresponds to the size of face, sepal width corresponds to the shape of forehead, petal length corresponds to the shape of jaw, and petal width corresponds to the length of nose.

Other glyphs commonly used for data visualization are *stars* [18]. Each object is displayed by a stylized star. In the star plot, the features are represented as spokes of a wheel circle, but their lengths correspond to the values of features. The angles between the neighboring spokes are equal. The outer ends of the neighboring spokes are connected by line segments. The Iris data plotted by star glyphs are presented in Fig. 5.6, where the stars, corresponding to Setosa irises, are smaller than the two other species [14].

5.1.2 Dimensionality Reduction

It is difficult to perceive the data structure using the direct visualization methods, particularly when we deal with large data sets or data of high dimensionality.

Another group of visualization methods (so-called projection methods) is based on reduction of the dimensionality of data. Their advantage is that each n-dimensional object is represented as a point in the space of low-dimensionality d, $d < n$, usually $d = 2$. There exists a lot of methods that can be used for reducing the dimensionality. The aim of these methods is to represent the multidimensional data

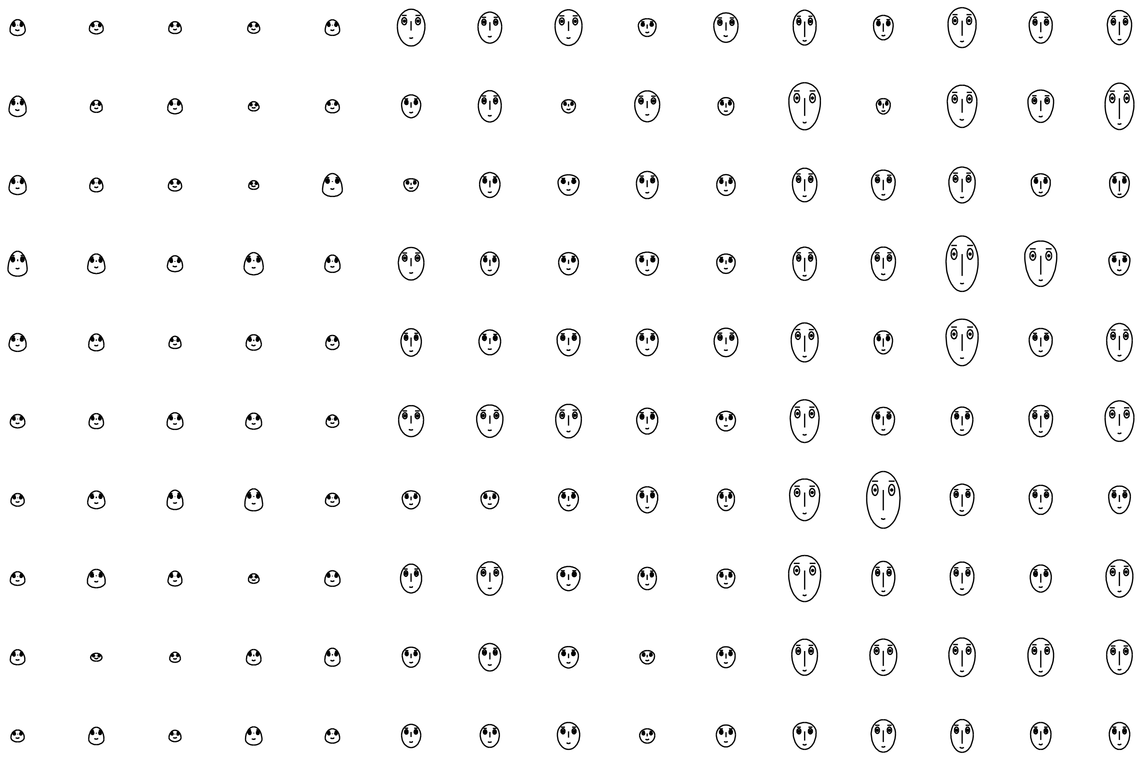

Fig. 5.5 Iris data visualized by Chernoff faces

in a low-dimensional space so that certain properties (such as distances, topology or other proximities) of the data set were preserved as faithfully as possible. These methods can be used to visualize the multidimensional data, if a small enough resulting dimensionality is chosen.

One of these methods is a principal component analysis (PCA). The well-known principal component analysis [19] can be used to display the data as a linear projection on such a subspace of the original data space that best preserves the variance in the data. Effective algorithms exist for computing the projection, even neural algorithms. The PCA cannot embrace nonlinear structures, consisting of arbitrarily shaped clusters or curved manifolds, since it describes the data in terms of a linear subspace. Projection pursuit [21] tries to express some nonlinearities, but if the data set is high-dimensional and highly nonlinear, it may be difficult to visualize it with linear projections onto a low-dimensional display even if the âĂIJprojection angleâĂİ is chosen carefully.

Several approaches have been proposed for reproducing nonlinear higher-dimensional structures on a lower-dimensional display. The most common methods allocate a representation for each data point in a lower-dimensional space and try to optimize these representations so that the distances between them are as similar as possible to the original distances of the corresponding data items. The methods differ in that how the different distances are weighted and how the representations are optimized.

Multidimensional scaling (MDS) refers to a group of methods that is widely used [22], [20]. The starting point of MDS is a matrix consisting of pairwise dissimilarities of the entities. In general, the

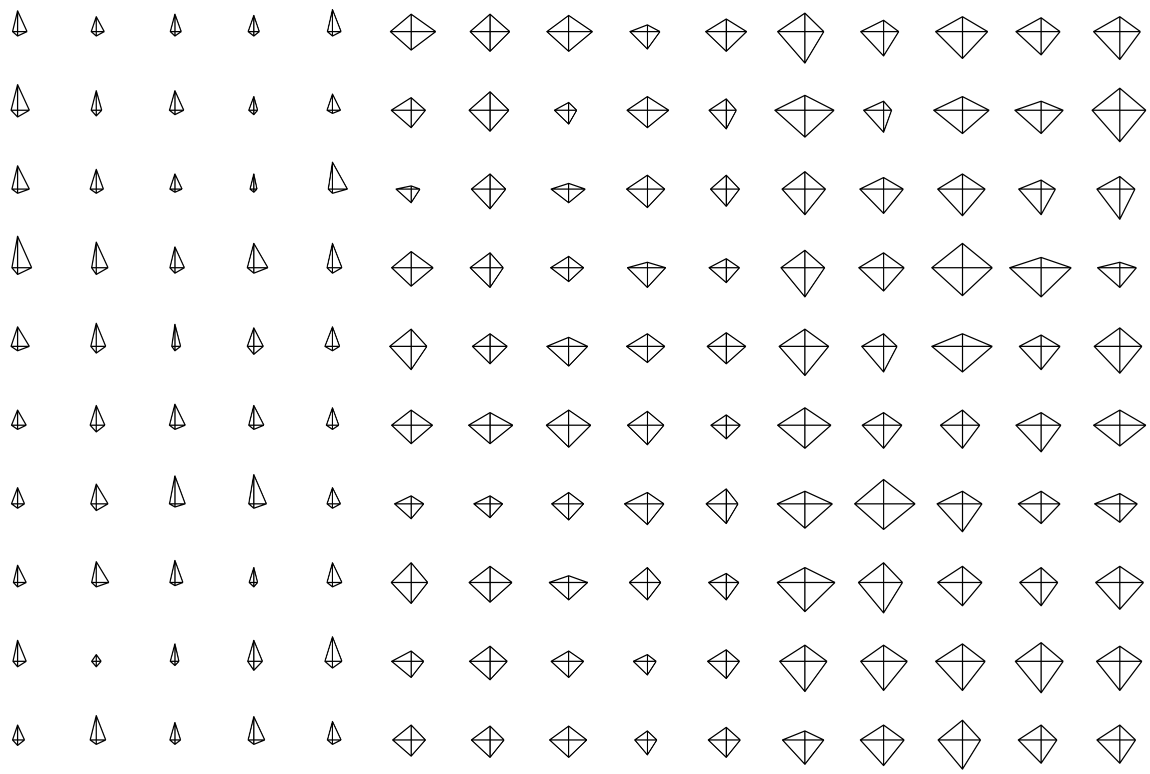

Fig. 5.6 Iris data set visualized by star glyphs

dissimilarities need not be distances in the mathematically strict sense. There exists a multitude of variants of MDS with slightly different cost functions and optimization algorithms. The MDS algorithms can be roughly divided into two basic types: metric and nonmetric MDS. The goal of projection in the metric MDS is to optimize the representations so that the distances between the items in the lower-dimensional space would be as close to the original distances as possible.

An example of visual presentation of the data table ($n = 6$, $m = 20$) using MDS is presented in Fig. 5.7. In this example, the dimensionality of data is reduced from 6 to 2.

There are some formal mathematical criteria of the projection quality. These criteria are optimized in order to get the optimal projection of multidimensional data onto a low-dimensional space. The main goal is to preserve the proportions of distances or estimations of other proximities between the multidimensional points in the image space as well as to preserve, or even to highlight other characteristics of the data multidimensional data (for example, clusters).

	x_1	x_2	x_3	x_4	x_5	x_6
1	0,57	0,11	0,47	0,54	0,84	0,53
2	0,65	0,14	0,53	0,55	0,89	0,54
3	0,61	0,13	0,49	0,52	0,92	0,57
4	0,44	0,16	0,45	0,42	0,99	0,45
5	0,55	0,11	0,53	0,52	0,92	0,55
6	0,44	0,16	0,43	0,41	0,98	0,43
7	0,49	0,12	0,56	0,46	0,82	0,48
8	0,41	0,17	0,42	0,43	0,98	0,42
9	0,54	0,12	0,59	0,46	0,85	0,48
10	0,69	0,16	0,63	0,56	0,91	0,49
11	0,66	0,16	0,51	0,48	0,87	0,55
12	0,70	0,13	0,52	0,48	0,81	0,52
13	0,65	0,14	0,63	0,52	0,89	0,49
14	0,59	0,11	0,51	0,57	0,89	0,52
15	0,53	0,11	0,52	0,57	0,89	0,50
16	0,52	0,13	0,51	0,61	0,92	0,51
17	0,58	0,12	0,55	0,61	0,95	0,52
18	0,45	0,10	0,52	0,53	0,92	0,51
19	0,42	0,17	0,41	0,42	0,98	0,40
20	0,53	0,11	0,42	0,57	0,91	0,57

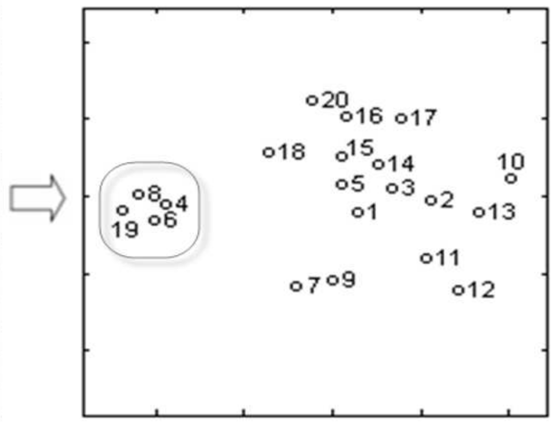

Fig. 5.7 Example of data visualization (dimansionality reduction)

5.1.3 New Possibilities for Big Data Mining and Representation

This section presents new trends for data representation and analysis in big data era. The main notation related to web services is defined. The main principles of Grid and Cloud computing for data mining are also described. Hadoop software is introduced. Advantages of scientific workflows are presented.

Cloud Computing Solutions

Common software has been based on the so-called *Service Oriented Architecture* (SOA). It is a set of principles used to build flexible, modular, and interoperable software applications. The implementation of SOA is represented by web services. A *Web Service* is a collection of functions that are packaged as a single entity and published in the network for use by other applications through a standard protocol [37]. The web service allows us to integrate heterogeneous platforms and applications. Services are running independently in the system, external components do not know how the services perform their functions. The components only care that the services would return the expected results. So, web services are widely used for on-demand computing [7].

Some important technologies related to web services are as follows: WSDL (Web Service Definition Language), SOAP (Simple Object Access Protocol) and REST (REpresentational State Transfer). WSDL is an XML format to describe web services as a set of endpoints operating on messages that consist of either document-oriented or procedure-oriented information. SOAP is a protocol for exchanging structured information in web services implementation in computer networks. REST is a style of software architecture that abstracts the architectural elements within distributed systems.

There is a large number of distributed and data-intensive applications and data mining algorithms that simply cannot be fully exploited without Grid or Cloud technologies [34, 39]. The main goal of cloud computing is to give a possibility to access distributed computing environments that can utilize computing resources on demand. Cloud-based data mining allows us to distribute a compute-intensive data analysis among a large number of remote computing resources.

Special tools for building the Grid and Cloud infrastructure have been developed. The Globus toolkit (`http://www.globus.org`), provided by the Globus Alliance, includes various software for different purposes: security, information infrastructure, resource management, data management, fault detection [30]. Nimbus (`http://www.nimbusproject.org/`) is an open-source toolkit focused on providing Infrastructure-as-a-Service (IaaS) capabilities to the scientific community. Amazon Elastic MapReduce (Amazon EMR) is a web service that enables researchers and data analysts to easily process large amounts of data. Using Amazon EMR, it is possible to use different capacities to perform data intensive tasks for applications such as data mining, machine learning, scientific simulation, web indexing, and bioinformatics research (`http://aws.amazon.com/whitepapers/`). Google introduced an on-line service to process large volumes of data. The service supports ad hoc queries, reports, data mining, or even web-based applications (`https://cloud.google.com/bigquery/`).

Hadoop software (`http://hadoop.apache.org/`) is widely used for distributed computations. The software includes some open source software projects: MapReduce, HDFS (Hadoop Distributed File System), Hbase, Hive, Pig, etc. [42]. MapReduce is a programming model for processing large data sets as well as a running environment in large computer clusters. Due to HDFS Hadoop allows us to save the computing time, needed for sending data from one computer to another. The Hadoop Mahout (`http://mahout.apache.org/`) library is developed for data mining, where some classification, clustering, regression, dimensionality reduction algorithms are implemented. However, there are not many data mining algorithms and, due to the MapReduce particularity, not always the existing data mining algorithm can be used easily and effectively.

Scientific Workflows

Many data mining systems based on web services are implemented using scientific workflow principles. A scientific workflow is a specialized form of workflows, developed specifically to compose and execute a series of data analyses and computation procedures in scientific application. The development of scientific workflow-based systems is under the influence of e-science technologies and applications [25]. The aim of e-science is to enable researchers to collaborate when carrying out a large scale of scientific experiments and knowledge discovery applications, using distributed systems of computing resources, devices, and data sets [23]. Scientific workflows play an important role in order to reach this aim. First of all, usage of scientific workflows allows researchers to compose convenient platforms for experiments by retrieving data from databases and data warehouses and running data mining algorithms in Cloud infrastructure. Secondly, web services can be easily imported as a new component of workflows.

Some merits of usage of scientific workflows are as follows: providing an easy-to-use environment for researchers to create their own workflows for individual applications; providing interactive tools for the researchers that enable them to execute the workflows and view the results in real-time; simplifying the process of sharing and reusing workflows among the researchers.

Some systems for scientific workflow management are developed, for example, Triana (`http://www.trianacode.org`), Pegasus workflow management system (`http://pegasus.isi.edu`), and Apache Airavata (`http://airavata.apache.org`).

5.1.4 Web Service-based Data Mining Systems

Data mining is the centre of attention among various intelligent systems, because it focuses on extracting useful information from big data. Data mining methods were rapidly developed by extending mathematical statistics methods and creating new ones. Later on, the data mining systems, in which several methods were usually implemented, were developed in order to facilitate solving the data mining problems. Many of them are open source and available for free, therefore they have become very popular among researchers. Recently, software applications have been developed under a SOA paradigm. Thus, some new data mining systems are based on web services. Attempts are made to develop scalable, extensible, interoperable, modular and easy-to-use data mining systems.

An Overview of Data Mining Systems

Some popular data mining systems have been reoriented to web services. Here, at first, we discuss extensions of Weka [3], Orange [4], KNIME [5] and MATLAB (http://www.mathworks.com).

Weka4WS is an extension of the widely used Weka [3] to support distributed data mining on Grids [40]. The system should be installed on a computer, but there is a possibility to select computing resources in the Grid environment. Weka4WS adopts the web services resource framework (WSRF) technology, provided by Globus Toolkit, for running remote data mining algorithms and managing distributed computing. In Weka, the overall data mining process takes place on a single machine and the algorithms can be executed only locally. In Weka4WS, the distributed data mining tasks can be executed on various Grid nodes by exploiting the data distribution and improving the application performance. Weka4WS permits us to analyse a single data set, using different data mining algorithms or using the same algorithm with different control parameters in parallel on multiple Grid nodes.

Orange4WS [36] is an extension of another well-known data mining system – Orange. Comparing with Orange, Orange4WS includes some new features. The ability to use web services as workflow components is implemented. The knowledge discovery ontology describes workflow components (data, knowledge and data mining services) in an abstract and machine-interpretable way. Usage of ontologies enables an automated composition of data mining workflows. In Orange4WS, there is a possibility to import external web services, only the WSDL file location should be specified. An example is presented in Fig. 5.8a). Here some workflow components (File to string, Data viewer, Get attribute, String) are Orange4WS components. FullCountryInfo is an external web service component (http://webservices.oorsprong.org/websamples.countryinfo/CountryInfoService.wso).

The KNIME system [5] is also extended by a possibility to use web services. In KNIME labs, a web service client node is developed that allows us to import web services to KNIME workflows. In Fig. 5.8b), an example of importing a web service to the KNIME system is presented. Using the component 'Generic Webservice Client', an external web service is imported here.

Usage of web services is implemented in a widely used programming and computing environment MATLAB (http://www.mathworks.com), too. There are implemented two types of web services RESTful and SOAP.

According to new technologies for data analysis, web service- and Cloud computing-based data mining systems have been designed and still underdeveloped. Some tools, related to web services, have been developed through the project myGrid (http://www.mygrid.org.uk). The tools support the creation of e-laboratories and have been used in domains of various systems: biology, social science, music, astronomy, multimedia, and chemistry. The tools have been adopted by a large number of projects: my-

Experiment [28], BioCatalogue [24], Taverna [35], [44], etc. MyExperiment is a collaborative environment where scientists can safely publish their workflows, share them with other scientists [28]. BioCatalogue is a registry of web services for live science [24]. Taverna is an open source and domain-independent workflow management system – a suite of tools used to create and execute scientific workflows [35], [44]. The system is oriented to web services, but the existing services are not suitable for data mining. However, there is implemented a possibility to import external data mining web service.

All the data mining systems previously reviewed (Weka4WS, Orange4WS, KNIME and Taverna) still remain desktop applications. Nowadays web applications become more popular due to the ubiquity of web browsers. ClowdFlows is a web application based on a service oriented data mining tool [8], [9]. ClowdFlows is an open sourced cloud-based platform for composition, execution, and sharing of interactive machine learning and data mining workflows. The data mining algorithms from Orange and Weka are implemented as local services. Other SOAP services can be imported and used in ClowdFlows, too. An example is presented in Fig. 5.8c).

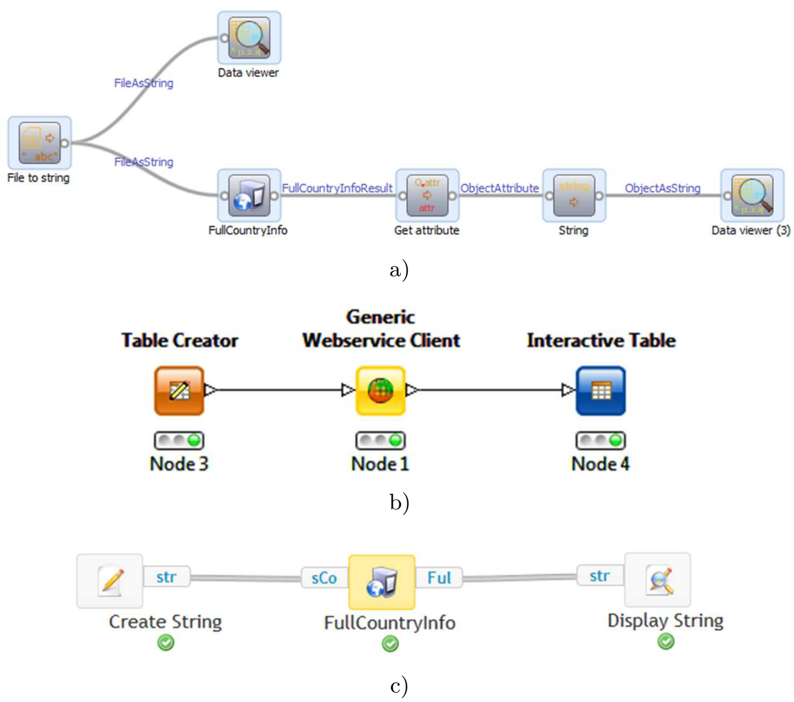

Fig. 5.8 Usage of an external web service in Orange4WS

DAME (DAta Mining & Exploration) is another innovative web-based, distributed data mining infrastructure (`http://dame.dsf.unina.it/`), specialized in large data sets exploration by data mining methods [10]. The DAME is organized as a cloud of web services and applications. The idea of DAME is to provide a user friendly and standardized scientific gateway to ease the access, exploration, processing and understanding of large data sets. The DAME system includes not only web applications, but also several web services, dedicated to provide a wide range of facilities for different e-science communities.

The systems that provide platforms for the data analysis, using different data mining methods and technical frameworks for computing performances of learning algorithms, as well as standardize the testing procedure, recently have gained popularity. The key aim of such systems is to provide a service for comparing and testing a number of data mining methods on a large number of real data sets.

Using MLcomp (`http://mlcomp.org`) users can upload programs or data sets (or use the existing data sets) and run any available algorithms on any available data set through the web interface. TunedIT offers an on-line algorithm comparison, too [43]. TunedIT specializes in the fields of data mining, machine learning, computational intelligence and statistical modelling. TunedIT has a research platform where a user can run automated tests on data mining algorithms and get reproducible experimental results through a web interface. Galaxy is an open, web-based platform for accessible, reproducible, and transparent computational biomedical research [31]. Users without programming experience can easily specify parameters and run tools and workflows. Galaxy captures information so that any user can repeat and understand a complete computational analysis.

Grid-based Data Mining

Distributed computing plays an important role in data mining. Data mining often requires huge amounts of resources in the storage space. Data are often distributed into several databases. Tasks of data mining are time consuming as well. It is important to develop mechanisms that distribute the workload of the tasks among several places in a flexible way. Distributed data mining techniques allow us to apply data mining in a non-centralized way [29]. Data mining algorithms and knowledge discovery processes are compute-intensive and data-intensive, therefore Grids offer a computing and data management infrastructure for supporting a decentralized and parallel data analysis [27]. So, a Grid is an inseparable part of distributed data mining.

As mentioned before, the usage of Weka4WS allows us to distribute computations to some resources. Weka4WS is implemented using the WSRF libraries and services provided by Globus Toolkit [40]. KNIME Cluster Execution allows us to run the data mining task in computer clusters.

A review of some systems and projects for Grid-based data mining is presented in [41]: Knowledge Grid, GridMiner, FAEHIM (Federated Analysis Environment for Heterogeneous Intelligent Mining), Discovery Net, DataMiningGrid, Anteater, etc. Most of them were created by some projects. Unfortunately, when the projects are completed, the systems are not supported.

Classification of Data Mining Systems

In this subsection, a classification of data mining systems, based on web services and reviewed above is proposed. Five classes are identified:

- The extensions of the existing popular data mining systems by web services are assigned to the class 'Extended by WS'. Their former properties remain and new ones are added.
- The systems developed using web services are assigned to the class 'Developed using WS'.
- The web application systems are assigned to the class 'Web apps'.
- The systems developed to running experiments are assigned to the class 'Services for experiments'.
- The systems based on Grid technologies are assigned to the class 'Grid-based'.

The reviewed data mining systems can be assigned to some classes (Fig. 5.9). For example, Weka4WS is an extension of the Weka system and the tasks can be run in Grids. ClowdFlows is a web application based on web services.

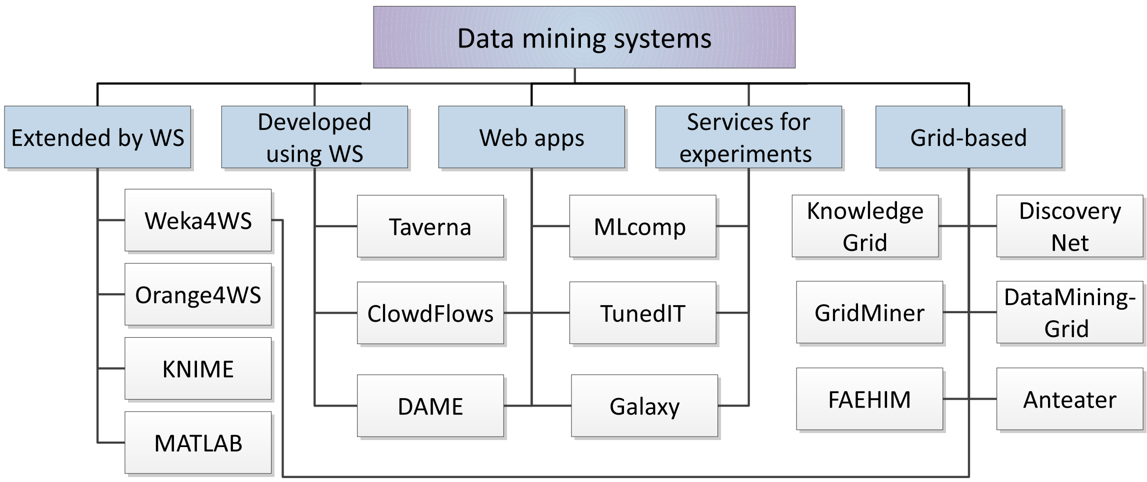

Fig. 5.9 Classification of data mining systems

5.1.5 Comparative Analysis of Data Mining Systems

Some comparative analyses of data mining systems are made in [38], [32], [26], [33]. Usually, attention is focused on data mining problems and tasks, while a comparative analysis according to web services and Cloud computing is missing. The aim of the analysis presented in the research is to compare web service-based underdeveloped data mining systems in order to highlight their merits and demerits and to determine the properties which should have a scalable, extensible, interoperable, modular and easy-to-use data mining systems.

First of all, a set of the criteria needs to be selected, according to which the systems will be evaluated and compared. The selected criteria, their possible values and grounding are presented in Table 5.1.

The data mining systems, assigned to two classes 'Extended by WS' and 'Developed using WS' (Fig. 5.9), are selected for evaluation. Taverna is not included into the evaluation, because it has no web services for data mining. The systems assigned to other classes should be evaluated by other criteria and it is out of scope of this research. The evaluation results are presented in Table 5.2. Here the sign '+'/'–' means that a system satisfies/unsatisfies the criterion, respectfully. In the last column, the total numbers of the pluses for each system are presented. The numbers of the systems, satisfying the corresponding criterion, are presented in the last row. From the results, presented in Table 5.2, the conclusions can be drawn:

- Almost all systems use only SOAP web services. The DAME system uses only the RESTful protocol. MATLAB is the only system in which two types of web services are implemented.

Table 5.1 Grounding of evaluation criteria for data mining systems

No.	Criteria	Possible values	Grounding
C1	Web service type	SOAP, RESTful	SOAP and RESTful are the most popular types of web services, thus it is important that both types would be implemented in the systems.
C2	Operating system	Yes, No	It is important that the systems would run on all popular operating systems Windows, Linux, Mac.
C3	Web application	Yes, No	An advantage of web applications is that they are used and controlled by a web browser without additional installing.
C4	External web services	Yes, No	It is important that it would be possible to import external web services without additional programming.
C5	Cloud computing	Yes, No	Cloud and Grid computing allows us to solve complicated and time consuming data mining problems.
C6	Repository of user's data	Yes, No	Users can save the data file into the web repository and it allows performing the different experiments with the same data without uploading them.
C7	Open source	Yes, No	Open source provides to extend and improve the systems.
C8	Scientific workflow	Yes, No	Scientific workflows allow us to create an environment for experiments, to save that for further investigations.
C9	Data mining methods	Classification, Clustering, Dimensionality reduction	The system universality is an important property in solving data mining problems, often it is necessary to apply some various data mining methods to the same problem.

- Al systems run on Windows, Linux, and Mac operating systems, only Weka4WS does not run on Mac.
- Only two systems ClowdFlows and DAME are implemented as web applications.
- Four systems Orange4WS, Knime, MATLAB, and ClowdFlows have a capability to import external web services. This criterion is very important, as data mining experts can use the web services created by others.
- Four systems are open source, so the system can be extended by adding new functionality.
- Cloud computing is supported in the most reviewed systems for big data mining.
- Scientific workflows are implemented in four systems Weka4WS, Orange4WS, KNIME, and ClowdFlows. The main advantage of this option is that data mining experts can choose different tools and create their workflows to solve a specific data analysis task.
- All the three groups of data mining methods are implemented in four systems Weka4WS, Orange4WS, KNIME and MATLAB).

The total results of the analysis have showed that the system satisfying most of the criteria is ClowdFlows (9 out of 12 points). The main advantages of the ClowdFlows system are as follows: it supports multi-OS and Cloud computing, it is the open source web application, there is possibility to

Table 5.2 Evaluation of data mining systems based on web services

	C1		C2	C3	C4	C5	C6	C7	C8	C9			
	SOAP	RESTful	Multi-OS support	Web app	External WS	Cloud comp.	Web repository	Open source	Workflows	Classification	Clustering	Dim. reduction	**Total**
Weka4WS	+	−	−	−	−	+	−	+	+	+	+	+	**7**
Orange4WS	+	−	+	−	+	−	−	+	+	+	+	+	**8**
KNIME	+	−	+	−	+	−	−	+	+	+	+	+	**8**
MATLAB	+	+	+	−	+	+	−	−	−	+	+	+	**8**
ClowdFlows	+	−	+	+	+	+	−	+	+	+	+	−	**9**
DAME	−	+	+	+	−	−	+	−	−	+	+	−	**6**
Total	**5**	**2**	**5**	**2**	**4**	**3**	**1**	**4**	**4**	**6**	**6**	**4**	

import external web services and to create scientific workflows. Orange4WS, KNIME and MATLAB have got 8 points. DAME is rated poorly (only 6 out of 12 points). We can clearly see from the comparative analysis that there is no data mining system that would satisfy all the criteria proposed. So, it is necessary to design and implement a new data mining system that should absorb all the merits of the existing systems and correct the shortcomings. The main properties obligatory to the new data mining system are highlighted and presented in Fig. 5.10.

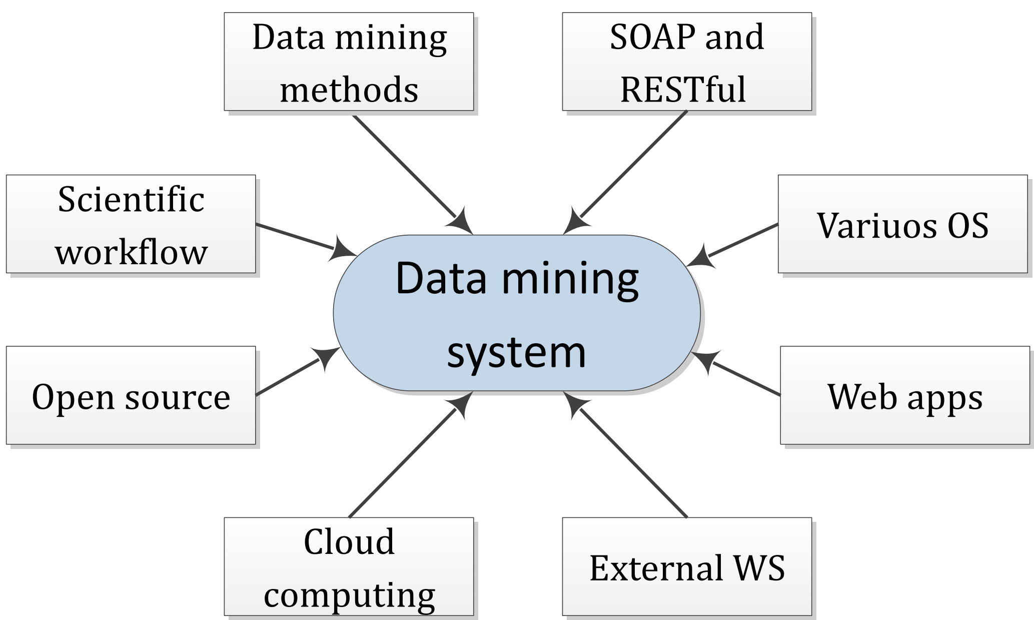

Fig. 5.10 Properties of a new data mining system

5.1.6 Generalization

In this research, solutions for data mining and scientific visualization have been discussed. Capabilities of scientific visualization have been illustrated using direct visualization methods and dimensionality reduction-based approaches. The review of web service-based data mining systems, suitable for scientific visualization, has been presented. A way of classification of data mining systems has been proposed. Some criteria for evaluating the systems have also been proposed. The analysed systems are compared by the proposed criteria and merits and demerits are highlighted. The comparative analysis of the data mining systems, based on web services, shows that the KNIME system is ranked best, it satisfies most criteria. Orange4WS and ClowdFlows systems are slightly behind. The DAME system is rated poorly.

The results of the comparison show that there is no data mining system that would satisfy all the criteria. Some properties required for a new system are determined: it should be possibility to import both SOAP and RESTful web services; at least three most popular operating systems (Windows, Linux, and Mac) must be supported; the system should be implemented by the popular programming language; it must be a web-based application; the possibilities to import external web services and to create the new ones should be realized; the distributed computing should be supported; the system should be open source; scientific workflows as well as various data mining methods should be implemented.

Additionally, the new data mining system should run on mobile devices and should have the ability to analyse big data. A possibility of choosing computing resources should be implemented. The system should be multi-user with a possibility to set user priorities, when a task queue is formed, the tasks of users of a higher priority must be run first of all. Users should be able to supplement the data repository of the system by new data sets. There should be a possibility to make the results of data mining experiments and appropriate workflows accessible for other users.

5.2 Data fusion, aggregation and management

Ciprian Dobre

Traditional technologies and techniques for data storage and analysis are not efficient anymore, in the Big Data and Data Science era, as the data is produced in high-volumes, come with high-velocity, has high-variety and there is an imperative need for discovering valuable knowledge in order to help in decision making process. There are many challenges when dealing with big data handling, starting from data capture to data visualization. Regarding the process of data capture, sources of data are heterogeneous, geographically distributed, and unreliable, being susceptible to errors. Current real-world storage solutions such as databases are populated in consequence with inconsistent, incomplete and noisy data. Therefore, several data preprocessing techniques, such as data reduction, data cleaning, data integration, and data transformation, must be applied to remove noise and correct inconsistencies and to help in decision making process.

Data quality is very important, especially in enterprise systems. Data mining techniques are directly affected by data quality. Poor data quality means no relevant results. Data cleaning in a DBMS means record matching, data deduplication, and column segmentation. Three operators are defined for data cleaning tasks: fuzzy lookup that it is used to perform record matching, fuzzy grouping that is used for deduplication and column segmentation that uses regular expressions to segment input strings. Fuzzy lookup and fuzzy grouping, available in several DBMS technologies (e.g., Microsoft SQL Server), are based on computing based on âĂIJdegrees of truthâĂİ rather than the usual "true or false" (1 or 0) Boolean logic. The Fuzzy Lookup and Fuzzy Grouping transformations help make the ETL (extract, transform, and load) process more resilient to common errors observed in real data (e.g., misspellings, truncations, missing or inserted tokens, null fields, unexpected abbreviations). They address generic matching and grouping problems without requiring an expert collection of domain-specific rules and scripts.

5.2.1 Case Study for Crowd Sensing

The cost of devices with sensing capabilities (devices that have at least one sensor, a processing unit and a communication module) has reached the point where deploying a vast number of them is an acceptable option for monitoring larger areas. This is what drives ideas such as Wireless Sensor Networks and Internet of Things and turns them into reality. Smart Phones, Smart watches, Smart wrist bands and new wearable computing platforms are hastily becoming ubiquitous. These devices incorporate a constantly increasing variety of sensors (eg. GPS, microphone, accelerometer, pedometer, gyroscope). And, all these sensors generate raw data that was previously unavailable or scarce. This data once processed can provide insights into human behavior and can help us understand the environment in which we live. Based on this data we can extract information and create better models of individuals, groups of individuals or their surroundings. Unlike previous methods, the large, constant data output may permit us to achieve these goals in real time and with high precision.

We note here two distinct types of sensors and approaches to large scale sensing: using static sensors and using mobile sensors. Static sensors are the ones that are part of a fixed infrastructure that has high deployment costs and requires some sort of administration. Mobile sensors are carried around by individuals that volunteer to offer the sensed data, because of their nature they require direct cooperation from the individuals that are part of this sensing system. Both approaches have been extensively researched in the literature, however we have yet to identify any previous work that tries to implement and deploy both systems simultaneously and directly measure or compare their results to asses each systems benefits.

Crowd sensing is a young yet extremely popular research topic. Smart phones have made mobile device data gathering a reality and because so many people own Smart phones having data from these devices at an extremely large scale is a reality. What makes Smart phones special is the combination of a constantly increasing large number of sensors attached to a powerful processor, and a communication module, be it 3G, Wi-Fi or Bluetooth. But Smart phones are just the start, more and more wearable devices are entering consumer use, such as smart watches or smart bracelets. Wearable devices have a lot of challenges and these are explored in [45]. Another more directed research is done in [46] where they explore the use of wearable devices specifically for low-cost health care. Moving from health care, smart devices can also be used as human assistive technology [52]. But all these devices serve a larger role when we consider them at scale. Having health data, environment data from different individuals can open new way to understanding the effects of complex systems over our activities and our wellbeing.

Wearables are currently extended with smart fabrics [49], materials which have sensing and electronic capabilities. These opens the door to wearable devices that are actually made of the clothes we wear, clothes that would otherwise serve only an esthetic purpose. Having data from a large number of sensors leads to ideas such as Smart Cities, Cities that use a large number of data source to enabled better conditions for its inhabitants. A conceptualization of Smart Cities can be found in [47] and a different focus is presented in [51]. The use of smart sensors has been identified as early as 2004 in works such as [53]. But these are considered only as static sensors.

The same ideas that put together Smart Cities can be used in a more general topic such as that of Ambient Intelligence [48]. The authors describe it as an emerging discipline which brings intelligence to every day environments.

We have to consider that crowd sensing is part of a larger field known as crowd work [50]. Crowd work represents all types of processes that require the cooperation of large crowds to be accomplished. These can be voluntarily or rewarded with compensations.

5.2.2 Data Aggregation for Crowd Sensing

Smart cities form an important focus of current research. People are envisioning more and better ways in which technology and information can improve our everyday life. To do so, we need to understand that life. In a city, traffic constitutes one of its crucial elements. Understanding traffic flows may help us to improve city life. In this paper, we concentrate on measurements on flows of pedestrians. In particular, we are interested in automatic and unobtrusive measurements on how pedestrians move through a specific designated area.

Tracking groups of people, understanding traffic flows, making sense of population densities, and so on, can be done in different ways. One solution takes advantage of the fact that many individuals carry smartphones or similar devices that are not only Wi-Fi enabled but also transmit Wi-Fi packets at regular intervals (for instance to search for new Wi-Fi hotspots). By deploying sensors that gather these packets we can gain more insight in pedestrian behavior.

Let us consider each Wi-Fi packet received at a sensor as a data point that we can later analyze. The number of data points depends on the number of people that carry Wi-Fi devices (we believe it to be correlated with the total number of people), the amount of Wi-Fi traffic these devices produce (dependent on the owner's usage of the device) and finally with the number of sensors that are deployed. Cities are large population centers with a high number of individuals, occupying large areas, requiring potentially tens to hundreds of sensors to cover the area of interest. As a consequence, the amount of data that we need to process can easily grow dramatically. Moreover, when having to deal with very large data sets, real-time analysis can become impossible, excluding many interesting applications, such as, for example, real-time crowd control by guiding people through less crowded areas.

In these detection systems there are a number of sensors placed around an area of interest. The sensors gather all Wi-Fi packets, construct detections and forward this data to a centralized server. The server then processes the data and creates visualization, analysis and/or long term storage. This architecture is common over all similar systems we encountered in this type of projects.

Understanding crowds and their movement has been an interesting research topic for some time. The benefits it can provide in transportation, simulations, and improving day-to-day activities has kept it as a main focus for research. Popular methods for understanding crowds are using visual analysis of camera feeds. An example of such a system can be seen in [55]. A similar system that uses cameras to measure the number of people in an area and to model small crowds through merging and splitting events is [56]. These systems can work at different scales. Previously we gave examples of systems that work at the size of crowds, but there are also systems that treat individuals movements such as described in [57]. An overview of visual systems can be found in [58]. We mention that visual systems also require filters for their data like in the case of [59] where the crowd is filtered out to reveal object left behind such as bags or packets.

The results of camera vision systems that track individuals can be used to detect human behavior. One solution for this uses models [61]. Extracting models of human behavior is one important output of crowd monitoring. These models have many uses in games and entertainment or medical and architectural aplications as stated in [61]. Many try to create better models like in the case of [62] or [63]. But without real-life measurements these models still lack realism. In [64] models are created that manage to mimic real-life measurements and offer a more realistic results. However these models use the scale of the entire city, here the model makes the correct assumption that humans have favorite locations where they spend most time, like home or work. After the models are created and refined enough they can be used in all kind of simulations such as to better identify opportunistic network algorithms [65].

When scale is required, video streams are not a valid option. Because of this many projects are researching other methods of extracting movement data without the use of camera systems. One example [66] uses data from a device carried around by individuals that gathers Wi-Fi and GPS signals. This data gives insight in human mobility and features of a city such as Access Point popularity in different areas. However, when there is a need to gather data from even more people using a device or an application installed on individual's phones is not acceptable. Because of this Wi-Fi systems have been built that manage to track humans without the need for them to be part of the system. Some works use patterns in signals on the Wi-Fi frequencies to identify individuals walking [67], groups [68] or even to count how many people are part of a group [69].

Different works focus on using Wi-Fi packet detectors, these are hotspots that act as sensors listening to packets in accordance to the 802.11(a/b/g/n) standard. Most Wi-Fi packets contain a MAC address permitting tracking of a device over multiple sensors. The advantage here is that most people already own a smartphone capable of communicating via Wi-Fi and they carry these devices with them most of the time. The disadvantage is that only people that have Wi-Fi enabled on their mobile device can be tracked. These systems have high popularity for indoor environments as can be seen in [70], [71] and [72] where not only localization is achieved but it is done with a high degree of accuracy, with errors less than 1 m. These systems can even be used to measure queues of people and their dynamics [73].

The systems that we are most interested in use Wi-Fi packet detection to measure movements over large areas (more than a few buildings). A good example is given by taking such measurements a botanical garden or a beach [75].

In our research we noticed that the authors of such works, to improve quality of sensed data, often use data filters as at least one step in their processing of the data. In [76] devices that do not appear at multiple hotspots are filtered out because they cannot show movement information if added to the final data set. This is one of the filters we will present in more detail in this paper. Duplicate detections (detections at multiple hotspots at the same time) and static devices (Wi-Fi enabled printers) are filtered out in the work presented in [77] where the data is used to analyze movement inside a campus. Finally [78] filter out devices that are known to be part of the buildings infrastructure and staff; they also filter indoor detections from outdoor ones.

The location of filters in the processing stream is discussed in detail in [74], where they show how distributing the computation of the filters can dramatically improve the processing time of applications. This is similar to our solution of filtering data as early as possible, on the sensors that detect the Wi-Fi packets that we describe next.

Filtering at the Sensor

In our studies, we deployed a small set of Wi-Fi packets sensors in the cities of Arnhem and Amstelveen (The Netherlands). These sensors are hotspots that monitor Wi-Fi and log all detected packets (meaning mostly Wi-Fi protocol headers; for security implications, we do not look inside the communication taking place). In our architecture, all logged packets are further sent to a centralized server, where further filtering discussed in the next section, and data processing algorithms are used (i.e., for tracking crowd mobility and crowd control applications).

When trying to achieve scalability in our system our main concern was the bandwidth utilization between the hotspots and the central server that gathers the data. To minimize bandwidth usage and control to some degree the quality of sensed data, we implemented three filters that would minimize the amount of packets we consider to be detections. These filters also have a correctness role: for instance, we do not want to consider devices that are not mobile, like a laptop that is permanently in use and in range of one of our hotspots, or a different hotspot.

- filter 1 – this filter accepts packets that have a *transmitter MAC address* and that MAC address is of a wireless device. Not all packets have a transmitter address and without one we cannot know who we assign the detection to; without it, we cannot track the device along multiple detections or multiple hotspots. We also mention that we are only interested in wireless devices, this is especially true in the case of data packets, and data packets have two bit fields named "from DS and to DS". These fields indicate if the packet is coming or going towards the wired distribution system. If the field "from DS" is set to 1 then the packet is coming from a device that is in a wired network with the access point used in the wireless communication. We are not interested in these wired devices and we filter them off. We believe that in other works [79] this filter might be missing and is causing the appearance of "Mystery OUIs". This filter is a correctness one, as it does however also make the entire system use less bandwidth and resources. The filter is also extremely fast: it only needs to check the type of the packet and in case it is a Data packet it needs to check the "from DS" field and this is all that is needed to make the decision if a packet passes the filter or not.
- filter 2 and 0 – Filter 2, also a correctness filter, eliminates all packets that come from access points. It is important to have the filter because in our case studies we encountered packets that have "from DS" set to 0 and the packets themselves have an access point as a transmitter. We know it was an access point because we encountered "Beacon" packets with the same MAC Address. Filter 2 does need to have a list with all encountered access points and this list is generated by filter 0. Complementary, filter 0 makes a list with all the MAC addresses of the "Beacons" it received, these packets are only sent by the access point and have their address in them, filter 0 also eliminates all "Beacon" packets. We mention here that all packets that are eliminated by filter 0 would have also been eliminated by filter 1 because we know "beacon" packets have no transmitter address different than the BSSID. The list of "Beacon" transmitter addresses saved by Filter 0 has a maximum size of 50. We chose 50 because we wanted it to be small and we do not expect that many sensors in range of one of our hotspots.
- filter 3 – this is not a correctness filter but it does offer high efficiency gains. This filter is temporal as it filters all the packets that have the same transmitter address as a detection made in the last 3 seconds. The 3 second interval was chosen empirically. For instance in the work of [79] a 1-second interval is used as an aggregation point that has the same effect as our filter. The larger the time frame is, less packets will be detected and less data will be sent to the central server; however this comes with a loss in accuracy. Another part of this filter is focused on correctness and it filters all the devices that we have seen for more than a few hours, devices that we consider non-mobile. We

chose the number of hours to be 5, but any reasonable amount can be used here. To be able to account for all detected devices this filter keeps a log with all the MAC addresses of the transmitter of all the packets that are considered detections. This log is kept in the form of a hash table with a statically allocated size.

We found empirically that filters work best in 0, 1, 2, and 3 ordering, this is also forced by some of the dependencies they have on each other. All the packets that pass all three filters are considered to be a detection. For these packets the transmitter's MAC are sent to the centralized server.

To evaluate our filters we used two distinct data sets. One data set is obtained on one of the hotspots inside a room of a student complex of VU University Amsterdam. The other data set is obtained in the city of Arnhem from the barXO hotspot. These two data sets are rather different, but they do have a similar number of total packets and size. We obtained the data sets by making a tcpdump on the hotspots that run for a few days. Because the two traces were run in different environments, particularities of these environments can be seen in the data. For instance the student-complex trace has a lots of data packets that are sent by only two devices. We also mention the channels: the student-complex data was gathered on a channel where only one active Wi-Fi network existed, while the Arnhem trace was run on a channel where no devices were actively communicating.

To test our filters we used both data sets but we did not put any time constraint limitations. In both cases we left the software process the data as fast as possible. This means that we analyzed a few days of data in under 3 minutes, and this has some dramatic effects on the effectiveness of the 3rd filter. The 3rd filter is however the only one affected by time, the others are independent.

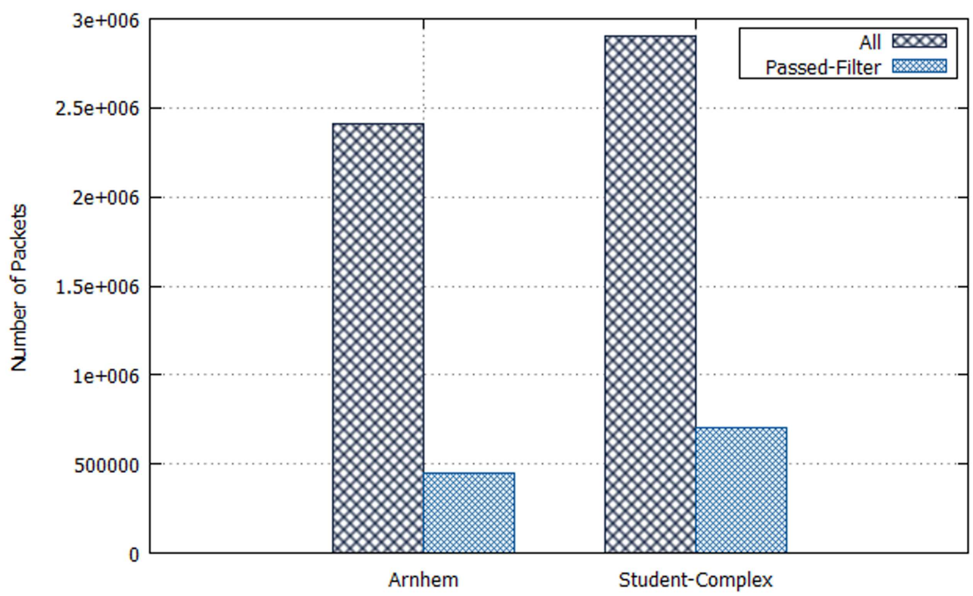

Fig. 5.11 Results for Filter 1

To have an accurate representation of how the filters functioned we stopped all the other filters and tested them independently. At the end all filters were operational and we tested them as a whole. The first filter eliminates all the packets that do not have a transmitter. As one can see in Figure 5.11 in both scenarios the filters managed to reduce the number of packets from over 2.5 Million to about 0.5 Million. To validate the functionality, we also sampled the data and manually inspected the filtered packets, as well as the packets that passed; no false negatives were found in the process. The more effective the filter, the smaller the red bar would be in comparison to the blue one. This filter serves in first place a correctness function, as all the packets that do not have a transmitter cannot be used as a detection. However this is an extremely fast and powerful filter as it eliminates about 80% of the packets.

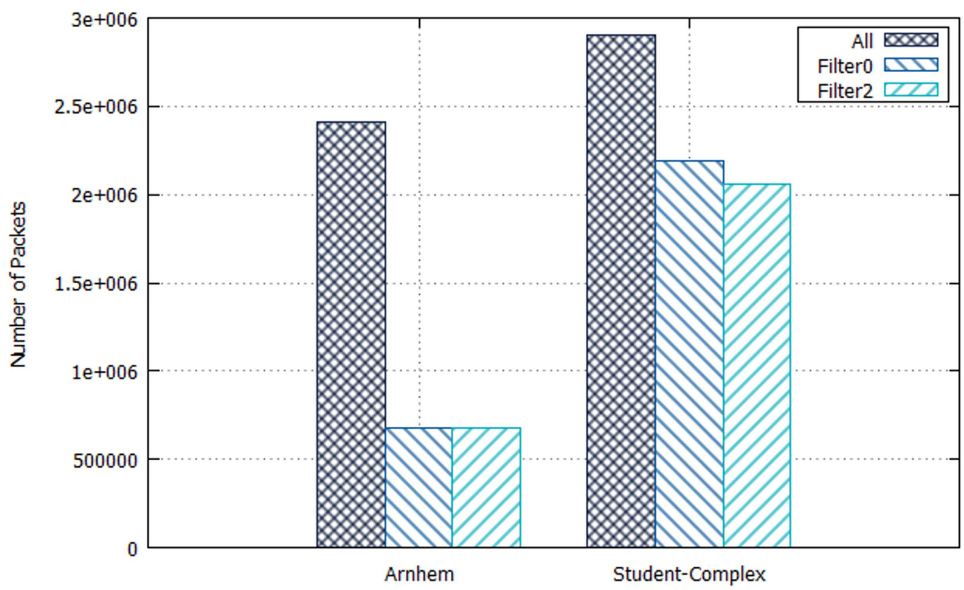

Fig. 5.12 Results for Filter 0 and 2

The results for filters 2 and 0 can be seen in Figure 2. The results vary so dramatically between the 2 scenarios because of the extremely large number of Beacon packets that dominate the Arnhem trace. Filter 2 is a correctness filter. This means that even though the number of packets filtered out by filter 2 in comparison to filter 0 and filter 1 is just minimal, the filter itself should still exist to eliminate all possible detections of non-mobile devices, of access points. Figure 5.12 shows how filter 2 works best in an environment where a lot of data traffic is expected. For instance, it might prove to be extremely useful in a residential area where people use Wi-Fi to stream movies or other high bandwidth usage content. However, in an area with coffee shops where most people just check their e-mail or do small amount of browsing, the filter might not be so efficient.

The final filter eliminates all packets that have transmitter MAC addresses that have been detected in the last 3 seconds. To be able to filter these packets, the filter needs to keep a log of all packets that have been registered as a detection. This filter is kept as a statically allocated hash table, with a maximum length. The size of the hash table directly affects the key calculation and the number of collisions it would have. To properly test the effectiveness of the third filter we compared the results on both data sets with varying maximum size of the hash table. We do this because we want to use the least amount of memory as possible while still having a maximally effective filter. The results are displayed in Figure 5.13. Here we can see a similar trend for both traces. Because we processed a few

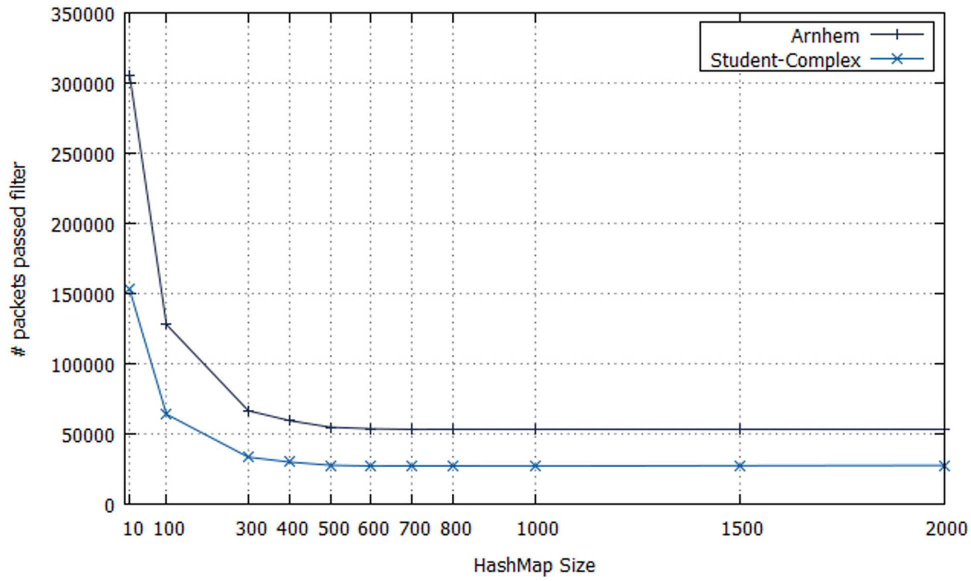

Fig. 5.13 Results for Filter 3

days of data in under two minutes and there were not an extremely high number of unique devices in the trace, this filter was very effective, there were a lot of packets with the same transmitter address that were processed in under three seconds, forcing most of them to be filtered.

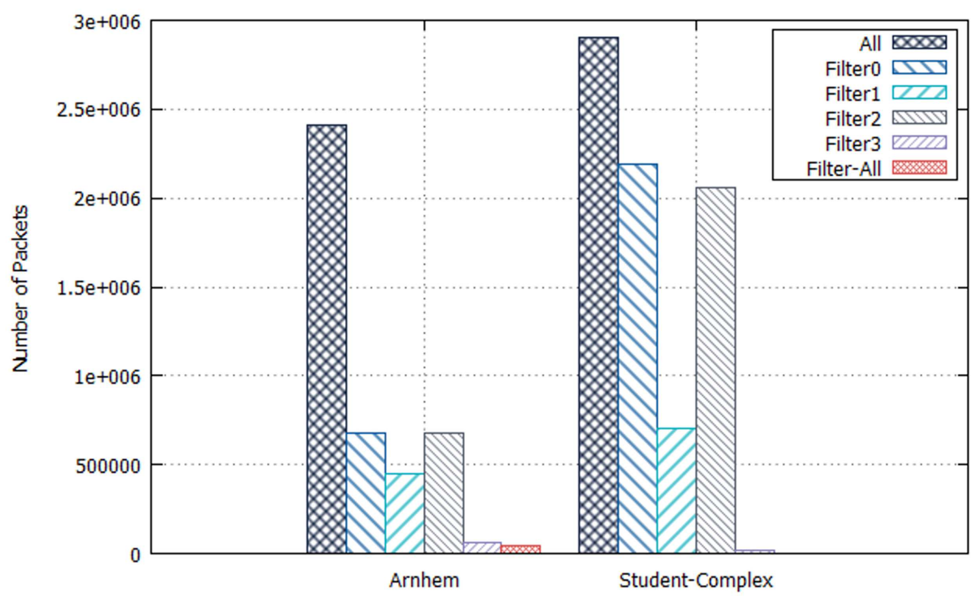

Fig. 5.14 Results for all Filters

In Figure 5.14 we compare all filters. Filter 0 to 3 are started one at a time (note here filter 2 cannot exist without filter 0). Then the last one is with all the filters started at the same time. Here we can easily see that the filters that make most of the difference are filter 1 and 3. Having all filters active minimizes the number of accepted packets even further, this happens mostly because packets that are accepted by filter 1 are not accepted by filter 3, the same is also true in reverse. This proves that all filters are important and that working together a very large number of packets are removed and only the most relevant ones are kept and sent to the central server.

Filtering Data After Ccentralization

The form of the data set that we produce is: *sensorid* âĂŞ *deviceid* âĂŞ *timestamp* Here *sensorid* and *deviceid* identify the sensor (as a way to extract its physical location) and, correspondingly, the detected mobile device (usually in the form of an MD5 hash). We have found that this format is similar across most projects that gather this type of data.

Filtering duplicates. This filter eliminates all data duplicates. We consider duplicates any two data points that have all three values (sensorid, deviceid and timestamp) equal. Without making a comparison with our approach, we note that other works consider duplicates any two data points with the same deviceid and timestamp (allowing different sensorids).

Filtering by time. Usually only a part of the data set is of interest to data analysis or the data set needs to be processed in chunks that span over one day or one week. This filter eliminates all data points that have a timestamp that does not fit between two given values.

Filtering Apple products. Apple products randomize their MAC address when sending probe requests to identify new hotspots (These probe requests are the packets that we capture when the device is not connected to a network, and we expect this to be the general case). Because this address is randomized a device using this feature cannot be tracked over multiple sensors. Even worse, two different devices can send out the same MAC address making the data set noisier.

Filter devices detected at only one sensor. As we are interested in movements of crowds devices that do not move or that have only been detected once and never seen again bring no information about the behavior of crowds, they just create noise in the data.

Table 5.3 Data set characteristics

	Total number of detections	Number of Sensors	Days of Interest
Arnhem	2472380	5	1
Assen	11860349	27	3

To test these filters we used two different data sets, one from the city of Arnhem that we produced using our own sensors and one from the city of Assen that was provided for us.

In Table 5.3 we notice the characteristics of the two data sets, we can observe how the two traces are very different in the number of detections and the number of hotspots that gathered these detections, the interest data also spans over different time frames. In Arnhem's case we were interested in the Living Statues Festival and for the case of Assen the interest was a TT Festival, these 2 festivals each attracted about a hundred thousand visitors to these 2 very similar cities.

By using the filters onto these two distinct data sets we obtained the results in Figure 5.15. We can observe how the initial data set has been reduced and how applying all four filters reduces the data set

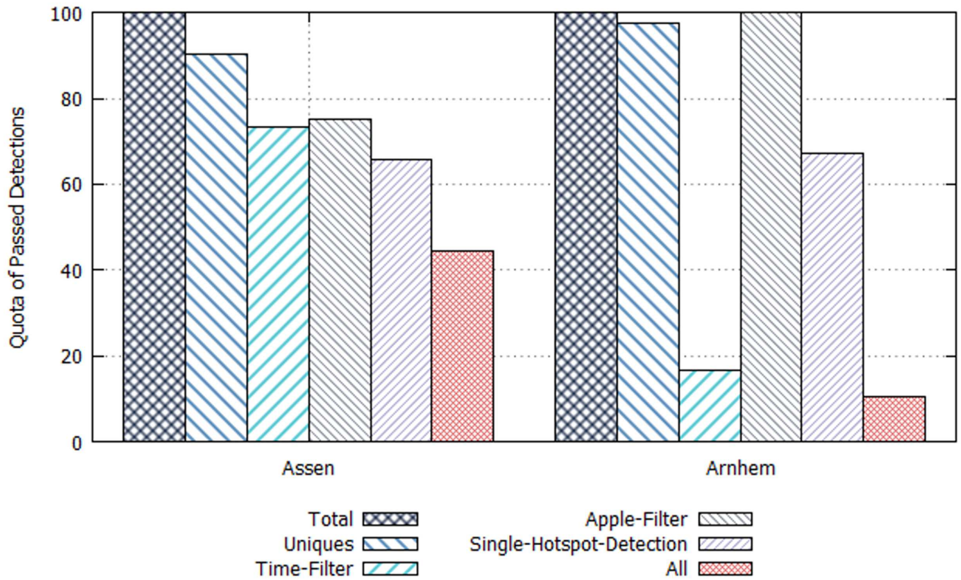

Fig. 5.15 Centralized Filter Results

even more. The Arnhem data set has been reduced to 10% of its original size and the Assen one to 44% of the original size. The large difference between the two is given by the number of data points outside the interest time, in the Assen data set there were a few extra hours of detection data while Arnhem data set had a few days of data.

References

[1] S. Schmidt, *Data is exploding: the 3 v's of big data*. Business Computing World (2012)

[2] D. Che,Safran, M., Peng, Z., *From big data to big data mining: Challenges, issues, and opportunities*. In: Database Systems for Advanced Applications. Volume 7827 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 1âĂŞ15

[3] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and Ian H. Witten. *The weka data mining software: An update*. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009. ISSN 1931-0145.

[4] J. Demšar, T. Curk, A. Erjavec, Črt Gorup, T. Hočevar, M. Milutinovič, M. Možina, M.a Polajnar, M.Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik, and B. Zupan. *Orange: Data mining toolbox in python*. *Journal of Machine Learning Research*, 14:2349–2353, 2013. URL `http://jmlr.org/papers/v14/demsar13a.html`.

[5] T. Meinl, N. Cebron, T R. Gabriel, F. Dill, and T Kötter, *The konstanz information miner 2.0*. *informatik.uni-konstanz.de*, 2009.

[6] M. Hofmann and R. Klinkenberg. *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. Chapman & Hall/CRC, 2013. ISBN 1482205491, 9781482205497.

[7] D. Birant. *Service-Oriented Data Mining*. New Fundamental Technologies in Data Mining, Prof. Kimito Funatsu (Ed.), 2011.

[8] J. Kranjc, V. Podpecan, and N. Lavrac. Clowdflows: A cloud based scientific workflow platform. In *Machine Learning and Knowledge Discovery in Databases*, Volume 7524 of *Lecture Notes in Computer Science*, pages 816–819. Springer Berlin Heidelberg, 2012.

[9] J. Kranjc, J. Smailovič, V. Podpečan, M. Grčar, M. Žnidaršič, and N. Lavrač. Active learning for sentiment analysis on data streams: Methodology and workflow implementation in the clowdflows platform. *Information Processing & Management*, (0), 2014. ISSN 0306-4573. URL `http://www.sciencedirect.com/science/article/pii/S0306457314000296`.

[10] B. Massimo, L. Giuseppe, Castellani M, Cavuoti S, D'Abrusco R, and Laurino O., *DAME: A Distributed Web Based Framework for Knowledge Discovery in Databases*. *Metnorie della Societa Astronomica Italiana Supplement*, 19:324–329, 2012.

[11] V. Medvedev, G. Dzemyda, O. Kurasova, V. Marcinkevicius, *Efficient data projection for visual analysis of large data sets using neural networks*. Informatica 22(4) (2011) 507âĂŞ520

[12] G. Dzemyda, O. Kurasova, V. Medvedev, *Dimension reduction and data visualization using neural networks*. In Maglogiannis, I., Karpouzis, K., Wallace, M., Soldatos, J., eds.: Emerging Artificial Intelligence Applications in Computer Engineering. Volume 160 of Frontiers in Artificial Intelligence and Applications. IOS Press (2007) 25âĂŞ49

[13] Hoffman, P.E., Grinstein, G.G., *A survey of visualizations for high-dimensional data mining*. In: U. Fayyad, G.G. Grinstein, A. Wierse (eds.) Information Visualization in Data Mining and Knowledge Discovery, pp. 47–82. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2002)

[14] G. Dzemyda,O. Kurasova, J. Zilinskas. *Multidimensional data visualization: methods and applications*. New York : Springer, 2013. XII, 250 p. ISBN 9781441902351.

[15] Grinstein, G., Trutschl, M., Cvek, U., *High-dimensional visualizations. In: Proceedings of Workshop on Visual Data Mining*, ACM Conference on Knowledge Discovery and Data Mining, pp. 1–14. ACM Press, New York (2001)

[16] A.Inselberg, *The plane with parallel coordinates*. The Visual Computer **1**(2), 69–91 (1985)

[17] H. Chernoff, *The use of faces to represent points in k-dimensional space graphically*. Journal of the American Statistical Association **68**(342), 361–368 (1973)

[18] J.M. Chambers, *Graphical Methods for Data Analysis (Statistics)*. Chapman & Hall/CRC (1983)

[19] D.N. Lawley, A.E. Maxwell, *Factor Analysis as a Statistical Method*. London: Butterworths, 1963.

[20] E. Oja, *Principal components, minor components, and linear neural networks*, Neural Networks 5 (1992), 927-935.

[21] C. Brunsdon, A.S. Fotheringham, M.E. Charlton, *An Investigation of Methods for Visualising Highly Multivariate Datasets in Case Studies of Visualization in the Social Sciences*, In: D. Unwin and P. Fisher (eds.) Joint Information Systems Committee, ESRC, Technical Report Series 43 (1998), 55-80.

[22] I. Borg, P. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, Springer, 1997.

[23] A. Barker and Jano I Van Hemert. Scientific Workflow: A Survey and Research Directions. *PPAM*, 4967:746–753, 2007.

[24] Jiten Bhagat, Franck Tanoh, Eric Nzuobontane, Thomas Laurent, Jerzy Orlowski, Marco Roos, Katy Wolstencroft, Sergejs Aleksejevs, Robert Stevens, Steve Pettifer, Rodrigo Lopez, and Carole A. Goble. BioCatalogue: A universal catalogue of web services for the life sciences. *Nucleic Acids Research*, 38, 2010.

[25] N. Cerezo, J. Montagnat, and M. Blay-Fornarino. Computer-assisted scientific workflow design. *Journal of Grid Computing*, 11(3):585–612, 2013.

[26] X. Chen, Y. Ye, G. Williams, and X. Xu. A survey of open source data mining systems. 4819:3–14, 2007.

[27] A. Congiusta, D. Talia, and P. Trunfio. Service-oriented middleware for distributed data mining on the grid. *Journal of Parallel and Distributed Computing*, 68:3–15, 2008.

[28] D. De Roure, C. Goble, and R. Stevens. *The design and realisation of the Virtual Research Environment for social sharing of workflows*, 2009.

[29] Talia Domenico and Trunfio Paolo. *Service-oriented distributed knowledge discovery*. Chapman and Hall/CRC, 2012.

[30] I. Foster. *Globus toolkit version 4: Software for service-oriented systems*. 3779:2–13, 2005.

[31] J. Goecks, A. Nekrutenko, and J. Taylor. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome biology*, 11: R86, 2010.

[32] Moez Ben Haj Hmida and Yahya Slimani. Meta-learning in grid-based data mining systems. *Int. J. Commun. Netw. Distrib. Syst.*, 5(3):214–228, 2010. ISSN 1754-3916.

[33] A. Jovic, K. Brkic, and N. Bogunovic. An overview of free software tools for general data mining. *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*, 11(3):1112–1117, 2014.

[34] V. Kravtsov, T. Niessen, V. Stankovski, and A. Schuster. Service-based resource brokering for grid-based data mining. In *in: Proc of Int Conference on Grid Computing and Applications*, page 163169, 2006.

[35] P. Missier, S. Soiland-Reyes, S. Owen, Wei Tan, A. Nenadic, I. Dunlop, A. Williams, T. Oinn, and C. Goble. Taverna, reloaded. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6187 LNCS, pages 471–481, 2010.

[36] V. Podpečan, M. Zemenova, and N. Lavrač. Orange4WS environment for service-oriented data mining. *Computer Journal*, 55:82–98, 2012.

[37] Web Services and Conceptual Architecture. Web Services Conceptual Architecture (WSCA 1.0). *Architecture*, 5:6–7, 2001.

[38] Vlado Stankovski, Martin Swain, Valentin Kravtsov, Thomas Niessen, Dennis Wegener, Jörg Kindermann, and Werner Dubitzky. Grid-enabling data mining applications with DataMiningGrid: An architectural perspective. *Future Generation Computer Systems*, 24:259–279, 2008.

[39] D. Talia and P. Trunfio. *How distributed data mining tasks can thrive as knowledge services*, 2010.

[40] D., Paolo Trunfio, and O. Verta. The Weka4WS framework for distributed data mining in service-oriented Grids. *Concurrency Computation Practice and Experience*, 20:1933–1951, 2008.

[41] D. Werner. *Data Mining Meets Grid Computing: Time to Dance?* John Wiley and Sons, Ltd, 2009.

[42] T. White. *Hadoop: The definitive guide*, volume 54. 2012.

[43] Marcin Wojnarski, Sebastian Stawicki, and Piotr Wojnarowski. *Tunedit.org: System for automated evaluation of algorithms in repeatable experiments*. 6086:20–29, 2010.

[44] Katherine Wolstencroft, Robert Haines, Donal Fellows, Alan Williams, David Withers, Stuart Owen, Stian Soiland-Reyes, Ian Dunlop, Aleksandra Nenadic, Paul Fisher, Jiten Bhagat, Khalid Belhajjame, Finn Bacall, Alex Hardisty, Abraham Nieva de la Hidalga, Maria P. Balcazar Vargas, Shoaib Sufi, and Carole Goble. The taverna workflow suite: designing and executing workflows of web services on the desktop, web or in the cloud. *Nucleic Acids Research*, 41(W1):W557–W561, 2013.

[45] Jiang, He, Xin Chen, Shuwei Zhang, Xin Zhang, Weiqiang Kong, and Tao Zhang, *Software for Wearable Devices: Challenges and Opportunities*. arXiv preprint arXiv:1504.00747 (2015).

[46] Fletcher, Richard R., Ming-Zher Poh, and Hoda Eydgahi. *"Wearable sensors: opportunities and challenges for low-cost health care."* In Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, pp. 1763-1766. IEEE, 2010.

[47] Nam, Taewoo, and Theresa A. Pardo. *"Conceptualizing smart city with dimensions of technology, people, and institutions."* In Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times, pp. 282-291. ACM, 2011.

[48] Cook, Diane J., Juan C. Augusto, and Vikramaditya R. Jakkula. *"Ambient intelligence: Technologies, applications, and opportunities."* Pervasive and Mobile Computing 5, no. 4 (2009): 277-298.

[49] Luprano, Jean. *"European projects on smart fabrics, interactive textiles: Sharing opportunities and challenges."* In Workshop Wearable Technol. Intel. Textiles, Helsinki, Finland. 2006.

[50] Kittur, Aniket, Jeffrey V. Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. *"The future of crowd work."* In Proceedings of the 2013 conference on Computer supported cooperative work, pp. 1301-1318. ACM, 2013.

[51] Nam, Taewoo, and Theresa A. Pardo. *"Smart city as urban innovation: Focusing on management, policy, and context."* In Proceedings of the 5th International Conference on Theory and Practice of Electronic Governance, pp. 185-194. ACM, 2011.

[52] Alam, Muhammad Mahtab, and Elyes Ben Hamida. *"Surveying wearable human assistive technology for life and safety critical applications: Standards, challenges and opportunities."* Sensors 14, no. 5 (2014): 9153-9209.

[53] Spencer, Billie F., Manuel E. Ruiz-Sandoval, and Narito Kurata. *"Smart sensing technology: opportunities and challenges."* Structural Control and Health Monitoring 11, no. 4 (2004): 349-368.

[54] Ma, Huadong, Dong Zhao, and Peiyan Yuan. *"Opportunities in mobile crowd sensing."* Communications Magazine, IEEE 52, no. 8 (2014): 29-35.

[55] Chan, A. B., Z.-S. J. Liang and N. Vasconcelos, *"Privacy preserving crowd monitoring: Counting people without people models or tracking.","* in Computer Vision and Pattern Recognition, 2008.

[56] R. D., D. S., F. C. and Sridharan, *"Crowd counting using group tracking and local features,"* in Advanced Video and Signal Based Surveillance , 2010.

[57] Aggarwal, J. K. and Q. Cai, *"Human motion analysis: A review,"* Computer vision and image understanding , vol. 73, no. 3, pp. 428-440, 1999.

[58] Z. Beibei, D. N. Monekosso, P. Remagnino, S. A. Velastin and L.-Q. Xu, *"Crowd analysis: a survey,"* Machine Vision and Applications, vol. 19, no. 5-6, pp. 345-357, 2008.

[59] C. Chuan-Yu, W.-H. Tung and J.-S. Wang, *"A crowd-filter for detection of abandoned objects in crowded area,"* in International Conference on Sensing Technology, 2008.

[60] Andrade, E. L., S. Blunsden and R. B. Fisher, *"Modelling crowd scenes for event detection."* in International Conference Pattern Recognition, 2006.

[61] Loscos, Celine, D. Marchal and A. Meyer, *"Intuitive crowd behavior in environments using local laws,"* in Theory and Practice of Computer Graphics, 2003.

[62] B. A., N. S., C. S. and D. Manocha, *"Densesense: Interactive crowd simulation using density-dependent filters,"* in Symposium on Computer Animation, 2014.

[63] K. Huber, M. Schreckenberg and T. Meyer-KÃűnig, *"Models for crowd movement and egress simulation,"* Traffic and Granular Flow, pp. 357-372, 2005.

[64] E. Frans, A. KerÃďnen, J. Karvo and J. Ott, *"Working day movement model,"* in ACM SIGMOBILE workshop on Mobility models, 2008.

[65] K. Dmytro, C. Boldrini, M. Conti and A. Passarella, *"Human mobility models for opportunistic networks,"* Communications Magazine, vol. 49, no. 12, pp. 157-165, 2011.

[66] S. P., S. A., G. R. and L. S, *"Tracking Human Mobility using WiFi signals,"* in arXiv preprint arXiv:1505.06311., 2015.

[67] W. Yan, J. Liu, Y. Chen, M. Gruteser, J. Yang and H. Liu, *"E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures,"* in international conference on Mobile computing and networking, 2014.

[68] S. Depatla, A. Muralidharan and Y. Mostofi, *"Occupancy Estimation Using Only WiFi Power Measurements,"* JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, vol. 33, no. 7, p. 1381, 2015.

[69] X. Wei, J. Zhao, X.-Y. Li, K. Zhao, S. Tang, X. Liu and Z. Jiang, *"Electronic frog eye: Counting crowd using wifi,"* in INFOCOM, 2014.

[70] K. Yungeun, H. Shin and H. Cha, *"Smartphone-based Wi-Fi pedestrian-tracking system tolerating the RSS variance problem,"* in PerCom, 2012 .

[71] R. Valentin and M. K. Marina, *"Himloc: Indoor smartphone localization via activity aware pedestrian dead reckoning with selective crowdsourced wifi fingerprinting,"* in International Conference on Indoor Positioning and Indoor Navigation , 2013.

[72] H. F., Z. Y., Z. Z., W. M., F. Y. and Z. Guo, *"WaP: Indoor localization and tracking using WiFi-Assisted Particle filter,"* in Local Computer Networks , 2014.

[73] W. Yan, J. Yang, H. Liu, Y. Chen, M. Gruteser and R. P. Martin, *"Measuring human queues using WiFi signals,"* in international conference on Mobile computing and networking, 2013.

[74] O. Chris, J. Jiang and J. Widom, *"Adaptive filters for continuous queries over distributed data streams,"* in ACM SIGMOD international conference on Management of data, 2003.

[75] A. Schmidt, *"Low-cost Crowd Counting in Public Spaces"*.

[76] B. Bram, A. Barzan, P. Quax and W. Lamotte, *"WiFiPi: Involuntary tracking of visitors at mass events,"* in World of Wireless, Mobile and Multimedia Networks , 2013.

[77] K. Eftychia, R. Sileryte, M. Lam, K. Zhou, M. v. d. Ham, E. Verbree and S. v. d. Spek, *"Passive WiFi Monitoring of the Rhythm of the Campus,"* in AGILE, Lisbon, 2015.

[78] Ruiz-Ruiz, A. J., H. Blunck, T. S. Prentow, A. Stisen and M. B. Kjaergaard, *"Analysis methods for extracting knowledge from large-scale WiFi monitoring to inform building facility planning,"* in Pervasive Computing and Communications , 2014.

[79] M. A. B. M. and J. Eriksson, *"Tracking unmodified smartphones using wi-fi monitors,"* in 10th ACM conference on embedded network sensor systems, 2012.

Chapter 6
Resource management and scheduling

Florin Pop, Magdalena Szmajduch, George Mastorakis, and Constandinos Mavromoustakis

6.1 Resource management in data centres

F. Pop

6.2 Scheduling for Big Data systems

C. Mavromoustakis, A. Mastorakis, M. Szmajduch

Florin Pop
University Politehnica of Bucharest, Romania, e-mail: `florin.pop@cs.pub.ro`

Magdalena Szmajduch
Institute of Computer Science, Cracow University of Technology, Poland, e-mail: `magdalena.szmajduch@gmail.com`

George Mastorakis
Technological Educational Institute of Crete, Grece, e-mail: `gmastorakis@staff.teicrete.gr`

Constandinos Mavromoustakis
Department of Computer Science, University of Nicosia, Cyprus, e-mail: `mavromoustakis.c@unic.ac.cy`

Chapter 7
Energy aware infrastructure for Big Data

Ioan Salomie, Marcel Antal, Joanna Kolodziej, Magdalena Szmajduch, Michał Karpowicz, Andrzej Sikora, Piotr Arabas, Ewa Niewiadomska-Szynkiewicz, George Mastorakis, and Constandinos Mavromoustakis

7.1 Introduction

J. Kolodziej

Over the last decade we have witnessed a modification and upgrading the infrastructures of computing services to high-performance computing systems that can meet the increasing demands of powerful newer applications. Simultaneously, almost in concert, computing manufacturers have consolidated and moved

Ioan Salomie
Technical University of Cluj-Napoca, Romania, e-mail: Ioan.Salomie@cs.utcluj.ro

Marcel Antal
Technical University of Cluj-Napoca, Romania, e-mail: Marcel.Antal@cs.utcluj.ro

Piotr Arabas
Institute of Control and Computation Engineering, Warsaw University of Technology, Poland, e-mail: parabas@ia.pw.edu.pl

Michał Karpowicz
Research and Academic Computer Network, Warsaw, Poland, e-mail: michal.karpowicz@nask.pl

Joanna Kołodziej
Institute of Computer Science, Cracow University of Technology, Poland, e-mail: jokolodziej@pk.edu.pl

George Mastorakis
Technological Educational Institute of Crete, Greece, e-mail: gmastorakis@staff.teicrete.gr

Constandinos Mavromoustakis
Department of Computer Science, University of Nicosia, Cyprus, e-mail: mavromoustakis.c@unic.ac.cy

Ewa Niewiadomska-Szynkiewicz
Institute of Control and Computation Engineering, Warsaw University of Technology, Poland, e-mail: ens@ia.pw.edu.pl

Andrzej Sikora
Research and Academic Computer Network, Warsaw, Poland, e-mail: andrzej.sikora@nask.pl

Magdalena Szmajduch
Institute of Computer Science, Cracow University of Technology, Poland, e-mail: magdalena.szmajduch@gmail.com

from the stand-alone servers to rack mounted blades. The aforementioned trends alone are increasing electricity usage in large-scale high-performance computing systems, such as data centers, computational grids, and cloud computing. An optimal energy utilization has reached to a point that many information technology (IT) managers and corporate executives are all up in arms to identify a holistic solution that can reduce electricity consumption (so that the total cost of operation is minimized) of their respective large-scale computing systems and simultaneously improve upon or maintain the current throughput of the system.

There are two fundamental problems that must addressed when considering a holistic solution that will deliver an energy-efficient resource allocation in high-performance computing, namely, the (a) total cost of operation (TCO) and (b) heat dissipation. However, both of these problems can be resolved by energy-efficient (re) allocation and management of resources. If absolutely nothing is done, then the power consumption per system is bound to increase which in turn would increase electricity and maintenance costs. Most of the electricity consumed by the high-performance processors is converted to heat, which can have a severe impact on the overall performance of a tightly packed system in racks (or blade chassis). Therefore, it is imperative to gather the consent of researchers to muster their efforts in proposing unifying solutions that are practical and applicable in the domain of high-performance computing systems.

Because of its sheer size, the modern large-scaled distributed systems, such as Computational Grids (CG) and Cloud Systems (CS) require advanced methodologies and strategies to efficiently schedule users tasks and applications to resources. Scheduling becomes even more challenging when energy efficiency, classical makespan criterion and user perceived Quality of Service (QoS) are treated as first-class objectives in the resource allocation methodologies. Therefore, the current efforts in the high-performance computing research is to design new effective schedulers, which can simultaneously minimize all such criteria.

7.2 Taxonomy of energy management in modern distributed computing systems

J. Kolodziej

A significant volume of research has been done in the domain of energy aware resource management in today's large-scale computing system. Following a taxonomy for cloud computing proposed in [17] the management methods in modern distributed computing systems can be classified into two main categories: static energy management (SEM) and dynamic energy management (DEM), as shown in Fig. 7.1.

Static methods contain all technologies that are applied for the system components, architecture and software optimization. At the hardware level the system devices can be replaced by the low-power battery machines or nano-processors and the system workload can be effectively distributed. It allows to optimize the energy utilized for computing the applications, storage and data transfer by reducing the fully consider the implementation of programs that are executed in the system in order to achieve a high and fast reduction in the energy usage. Even with perfectly designed hardware, poor software design can lead to significant power and energy losses. Therefore the process of compilation or code generation and the order of instructions in application source code may have an impact on energy utilization. All static methods should be easily adapted to a given system and application specific parameters and conditions.

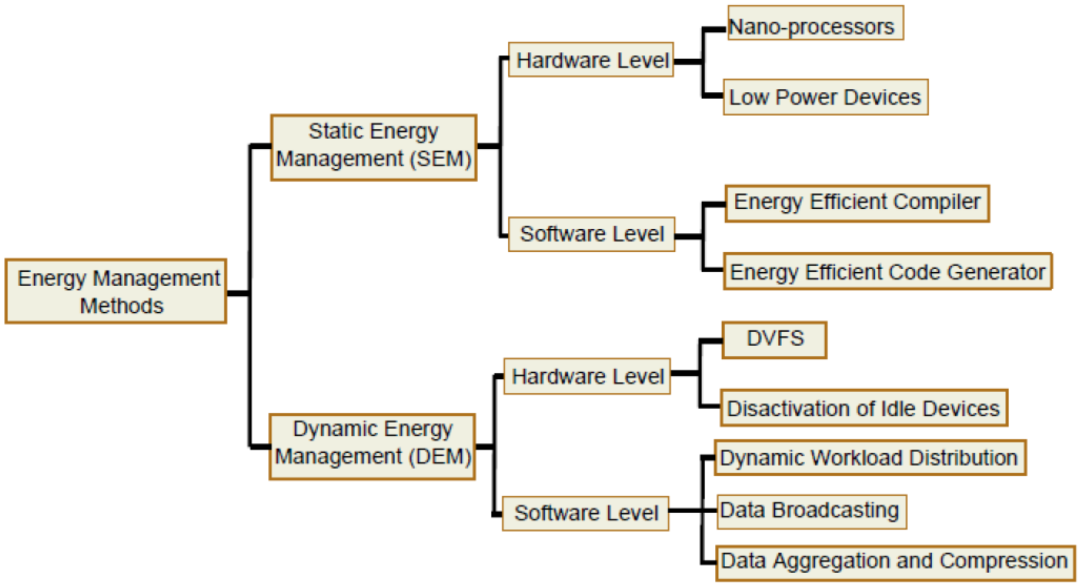

Fig. 7.1 Taxonomy of energy management in large-scale distributed computing systems

Dynamic energy management techniques include the strategies for dynamic adaptation of the system performance to current resource requirements and other parameters of the system's state. The major assumption enabling dynamic techniques is that systems experience variable workloads during their operation allowing the dynamic adjustment of power states according to current performance requirements. Similarly to static solutions, dynamic management methodologies can be distinguished by their application levels into hardware and software local categories. Hardware tools can be classified as dynamic performance scaling (DPS), such as DVFS, and partial or complete dynamic deactivation of inactive processors. The software techniques class includes all optimization techniques connected with dynamic workload distribution, efficient data broadcasting, data aggregation and dynamic data (and memory) compression.

7.3 Energy aware CPU control

M. Karpowicz, P. Arabas

According to statistical data [26, 27], utilization of servers in data centers rarely approaches 100%. Most of the time servers operate at 10-50% of their full capacity, which results from the requirements of providing sufficient quality of service provisioning margins. Over-subscription of computing resources is applied as a sound strategy to eliminate potential breakdowns caused by traffic fluctuations or internal disruptions, e.g. hardware or software faults. A fair amount of slack capacity is also required for the purpose of maintenance tasks. Since the strategy of resource over-provisioning is, clearly, a source of energy waste, the provisioned power supply is less than the sum of the possible peak power demands of all the servers combined. This, however, rises the problem of power distribution in data center. To keep the actual total power use within the available power range, servers are equipped with power budgeting mechanisms (e.g. ACPI-based) capable of limiting their power usage. The challenge of energy-efficient data center control is, therefore, to design control structure improving the utilization of servers and reducing energy consumption subject to quality of service constraints in highly stochastic environment (random stream of submitted jobs, OS kernel task scheduling), capable of providing fast responses to fluctuations in application workloads.

Server power control in data centers is a coordinated process that is carefully designed to reach multiple data center management objectives. As argued above, the main objectives are related with maintaining reliability and quality of data center services. These include avoiding power capacity overloads and system overheating, as well as fulfilling service-level agreements (SLAs). In addition to the primary goals, server control process aims to maximize various energy efficiency metrics subject to reliability constraints. Mechanisms designed for these purposes exploit closed-loop controllers dynamically adjusting operating modes of server components to the variable system-wide workload. Structures of currently developed control systems usually consist of two layers. The system-wide control layer adjusts power consumption of computing nodes in order to keep data center power consumption within the required bounds. Quality of services provided by the application layer is controlled by server-level control layer, see e.g. [136, 137, 101, 67].

Power consumption controllers are usually designed according to well-established methods of feedback control theory in order to reach specified objectives and to provide sufficient guarantees for performance, accuracy and stability of server operations [66, 23, 24, 32] . Apart from these goals additional requirements are introduced by hardware and software related constraints.

Since power consumption of CMOS circuit is proportional to its switching frequency and to the square of its operating voltage, performance and power consumption of a processor can be efficiently controlled with with dynamic voltage and frequency scaling (DVFS) mechanisms. These standard mechanisms, simultaneously adapting frequency and voltage of voltage/frequency islands (VFIs) present on the processor die [73], are commonly used as a basic method for server power control. First, contribution of CPU in server power consumption spans from 40% to 60% of total, which shows the dominant role CPU plays in the server power consumption profile [130, 27, 86]. Furthermore, since the difference between the maximal and minimal minimal power usage of CPU is high, DVFS allows to compensate for the power variation of other server components. Second, most servers support DVFS of processors, power throttling of memory or other computing components is not commonly available [136].

In the Linux systems DVFS is implemented by ACPI-based controllers. Translation of commands responsible for the CPU frequency control is provided by the `cpufreq` kernel module [117, 90]. The module allows to adjust performance of CPU by activating a desired software-level control policy implemented as a CPU *governor*. Typically, the calculated control inputs are mapped to admissible performance ACPI P-states and passed to a processor driver (e.g. `acpi_cpufreq`). Each governor of the `cpufreq` module implements a frequency switching policy. There are several standard build-in governors available in the recent versions of the Linux kernel. The governors named `performance` and `powersave` keep the CPU at the highest and the lowest processing frequency, respectively. The `userspace` governor permits user-space control of the CPU frequency. Finally, the default energy-aware governor, named `ondemand`, dynamically adjusts the frequency to the observed variations of the CPU workload. The sleeping state, or ACPI C-state, which CPU enters during idle periods, is independently and simultaneously determined by the `cpuidle` kernel module [118].

A straightforward DVFS-based policy of energy-efficient server control can be derived immediately from the definition of energy efficiency metric. In order to increase the number of computing operations performed per Watt it is necessary to reduce the amount of time the processor spends running idle loops or stall cycles [96]. Therefore, energy-efficiency maximizing controller should implement a workload following policy dynamically adjusting CPU performance state to the observed short-term CPU utilization or application-related latency metrics.

The above control concept is implemented in the CPU frequency governors of the Linux kernel, namely `intel_pstate` and `cpufreq_ondemand`. The output signal used in the feedback loops is an estimated CPU workload [11]. In the case of Intel IA32 architectures it can be calculated as the ratio of the MPERF counter, `IA32_MPERF` (0xe7), running at a constant frequency during active periods (ACPI C0 state), and the time stamp counter, `TSC` (0F 31), incremented every clock cycle at the same frequency during active and inactive periods (ACPI C-states). Another commonly applied CPU workload estimate, though somewhat less accurate, is given by the ratio of the amount of time the CPU was executing instructions since the last measurement was taken, $\mathrm{wall\_time} - \mathrm{idle\_time}$, to the length of the sampling period, $\mathrm{wall\_time}$. Given the CPU workload estimate the `intel_pstate` governor applied PID control rule to keep the workload at the default reference level of 97%. The `ondemand` governor calculates CPU frequency according to the following policy. If the observed CPU workload is higher than the upper-threshold value then the operating frequency is increased to the maximal one. If the observed workload is below the lower-threshold value, then the frequency is set to the lowest level at which the observed workload can be supported.

Many application-specific designs of the energy-efficient policy have been investigated as well. Design of decoding rate control based on DVFS, applied in multimedia-playback systems, is presented in [104]. The proposed DVFS feedback-control algorithm is designed for portable multimedia system to save power while maintaining a desired playback rate. In [92] a technique is proposed to reduce memory bus and memory bank contention by DVFS-based control of thread execution on each core. The proposed control mechanism deals with the problem of performance gap between processors and main memory, and the scenario in which memory requests simultaneously generated by multiple cores result in memory accesses delay for computing threads. A process identification technique applied for the purpose of CPU controller design is presented in [139]. Based on stochastic workload characterization, a feedback control design methodology is developed that leads to stochastic minimization of performance loss. The optimal design of the controller is formulated as a problem of stochastic minimization of runtime performance error for selected classes of applications. Adaptive DVFS governors proposed in [128] take the number of stalls introduced in the machine due to non-overlapping last-level cache misses as their input, calculate performance/energy predictions for all possible voltage-frequency pairs and select the one that yields the minimum energy-delay performance. In [85] a supervised learning technique is used in DVFS to predict the performance state of the processor for each incoming task and to reduce the overhead of the state observation. Finally, in [90] an attempt is made to characterize a possibly general structure of energy-efficient DVFS policy as a solution to stochastic control problem. The obtained structure is characterized in terms of short-term best-response function minimizing weighted sum of average costs of server power-consumption and performance. It is also compared to the `ondemand` governor, a default DVFS mechanism for the Linux system. The following interpretation can be given to the derived policy. Whenever it is possible for the server to process workload with the CPU frequency minimizing short-term operational cost, then frequency minimizing short-term costs should be selected. Otherwise, the controller should set the frequency optimizing long-term operational cost, this however being bounded from below by the frequency minimizing short-term costs.

Finally, power consumption controllers are designed and implemented directly in the server firmware [111, 97, 65, 73, 113]. The proposed solutions allow to keep the reference power consumption set point, control peak power usage, reduce of power over-allocation and maximize energy-efficiency by following the workload demand. It should be noted that in this case hardware producers often provide mechanisms that optimize operations of all interacting server components directly during runtime.

Currently observed trends in server power control exploit increasing possibilities provided by high-resolution sensors of modern computing hardware and software. The related research efforts seem to be drifting beyond CPU control towards increasing power proportionality of other subsystems, including memory and network interfaces. More advanced control algorithms and structures, outperforming standard PID controllers, are also proposed for device drivers and kernels of modern operating systems.

7.4 Energy aware schedulers

I. Salomie, M. Szmajduch, M. Antal

7.4.1 Introduction

Energy-aware schedulers are intelligent components, part of the Data Center (DC) Management Software, aiming at planning a series of actions to be executed by the DC subsystems for optimizing DC operation in terms of energy-efficiency by using energy flexibility mechanisms. In the following, the concepts of energy-aware schedulers and energy-aware optimizers will be used interchangeably.

An energy-aware or green DC Scheduler/Optimizer should determine a plan, defined below as a set of *(action, t)* tuples which should be applied sequentially upon DC subsystems for each timeslot t_s in W in order to minimize DC energy defined as an *objective function* and without causing any violations of the policies and constraints defined on the DC domain model:

$$Plan = \{(action, t) | action \in \{Actions\} \textbf{ and } t \in W = \{1..T\}\} \tag{7.1}$$

where W is the scheduling/optimizing time window, consisting of T time slots, each timeslot t_s in W being of t_p units of time. *Actions* represent the set of all possible actions that can be applied on the DC subsystems. The *objective function*, single or multi-criteria, is defined by the DC Top Management in line with the energy-aware operation strategy defined for the considered time window of DC operation.

When defining the *energy-aware operation strategy*, the DC Management may take into account the available flexibilities of the DC subsystems and the current context of DC operation. For example, considering low electrical energy price on the local marketplace and a long period of sunshine as the current operational context, a possible DC operation strategy defined by the DC management could be *âĂđmatch as much as possible the contracted energy for the current operation day and sell on the marketplace the energy surplus produced by the local RENâĂİ*.

For the scheduling / optimization problem, the unknown is the set of actions that should be determined in line with the operation strategy defined by the DC management that applied on the DC subsystems for the considered time window period will minimize the energy-related objective function. The challenge is to solve a combinatorial problem defined on the domains of these variables and determine the subset of values that minimize the optimization function while fulfilling all the defined constraints.

It can be shown that the resulting optimization problem is NP by reducing it to one of the well-known NP problems, such as bin-packing, job-shop scheduling, generalized assignment problem, or by modeling it using mathematical programming.

In the following sections, the topic of energy-aware schedulers/optimizers will be developed in the context of cloud-computing data centers consisting of IT subsystem, Cooling subsystem, Batteries, TES (Thermal Energy Storage) and Local renewable energy (REN) source such as solar panels or wild mills. We will also consider that the DC is interconnected with a smart grid through an energy marketplace and also with a set of partner DCs (Federated DCs).

7.4.2 Basic actions of DC subsystems

The goal of the energy-aware schedulers is to determine a plan as a set of actions that should be applied upon the DC subsystem controllers to satisfy a predefined energy consumption related objective function.

The most common optimization actions that can be algorithmically controlled are listed below for each of the DC subsystems:

- IT subsystem actions
  - Server level actions: turn power on, turn power off, idle - put node in idle state, DVFS (Dynamic Voltage Frequency Scaling) - adapt dynamically the frequency of the server processor.
  - VM actions: deploy the VM on a server, migrate the VM from a server to another server, resize the allocated resources for a VM.
- Cooling subsystem actions: turn on power on the cooling system, turn off power on the cooling system, adjust intensity and compensate using Thermal Energy Storage (TES).
- Battery and Thermal Energy Storage (TES) actions: charge unit, discharge unit.
- Local REN actions: use the amount of green energy produced locally.
- Network actions: schedule traffic flows between nodes, route traffic flows between nodes.
- Grid-Marketplace actions: buy energy; sell energy.
- Federated DC actions: inter-cloud workload migrate.

7.4.3 Energy-aware flexibility mechanisms

Modern energy-efficient Data Centers are scheduling/optimizing their operation by considering (i) the basic actions that can be executed by the controllers of its subsystems and (ii) specific flexibilities that can be used in association with its subsystems.

This section discusses the energy-efficient related flexibilities of the DC subsystems.

IT subsystem

The IT subsystem resources consist of processing nodes grouped together in racks. Depending on the computation types, the processing nodes of modern DCs can be either *blade servers*, with onboard CPUs, RAM, HDD, Graphical Processing Units (GPU), network interface cards and shared storage systems, or *compute nodes* grouped in *clusters* for high performance computing.

An approximation of the power consumed by main server components CPU, disk, and network interface under different configurations is presented in [21]. Results show that the CPU consumption is influenced by load, CPU frequency and the active numbers of cores. Similarly, the CPU frequency together with the I/O blocks size influence the hard disk consumption while Network Interface Controller (NIC) depends on CPU frequency and on the packet size and transmission rate. To estimate the total energy consumed by an application, the sum of the following terms is considered: Eb (baseline power times the duration of the execution), Ec (CPU consumption), Ed (disk consumption) and En (network consumption). The proposed estimation was validated using a Hadoop application and a second process that generated network traffic. The results show that the estimations are accurate, the error being below 7%.

The main flexibility means at the IT subsystem level that an energy-aware scheduler/optimizer can play with are virtualization, workload features, SLA, energy-aware resource allocation through resource provisioning and consolidation and dynamic power management. All these flexibilities will be addressed below.

Virtualization

It has been shown [88], [8] that in large classical DCs with thousands of servers the average utilization of computing resources less than 30% thus providing huge opportunities for energy savings. Nowadays, modern DCs organized in clouds have massively adopted the virtualization technique thus allowing (i) application virtualization into Virtual Machines (VM), (ii) multiple virtualized applications to be hosted on the same physical server and (ii) the on-demand provisioning of the capacity of the required computing resources. Virtualization allows for (i) running multiple operating system environments or applications on the same physical server; (ii) high level of isolation between the applications running in different VMs and (iii) dynamically reallocating the applications hosted on VMs on different physical server hardware thus achieving a higher computational density and higher workload consolidation during periods with low requests.

Workload

Workload can be classified as real-time, which has strong constraints on real-time execution, and delay-tolerant workload, which can be executed any-time until a given deadline. The delay tolerant workload offers a range of flexibilities in energy-aware scheduling. For this type of workload, DC power demand can be reduced at timeslot **t** with the amount of energy needed to execute the delay tolerant load that is shifted at timeslot **t+u**, while the DC power demand at timeslot **t+u** is increased with the amount of power needed to execute the delay-tolerant load shifted from timeslot **t**.

SLA

Workload associated SLA levels have a direct impact on (i) the amount of computing resources allocated for executing that workload, (ii) the execution deadline (for the delay-tolerant workload), and (iii) the amount of energy needed to execute the workload. As a consequence, a decrease of the SLA level will have as result a decrease of the DC energy consumption and a better pricing mode for the end-customer. As a flexibility mean, to facilitate energy-efficient scheduling, the workload associated SLAs contracted by the customers may be decreased, within agreed limits, and for limited periods of time. This way, DCs Green Performance Indicators (GPIs) could be improved while keeping the Key Performance Indicators (KPIs) within agreed limits.

Energy-aware resource allocation

The objective of the energy-aware resources allocation is to distribute the VMs hosting clients' applications on the DC's hardware resources in such a way that the energy consumption is minimized while the SLA agreements are kept within agreed ranges. Energy-aware resource allocation is achieved in two stages: (1) provisioning of computing resources for the new VMs and (2) consolidation of the DC by migrating the existing, active VMs between servers.

Resource provisioning aims at placement of the applications on a set of physical servers in such a way that a multi criteria objective function is minimized or maximized. This is a well studied NP problem where the energy consumption, Green and Key Performance Indicators (GPIs and KPIs) and other metrics are the usually considered as the optimization criteria.

Resource consolidation techniques take advantage of virtualization by proposing models to migrate the VMs in a DC from one server (or cluster) to another so that they better fit to the DC's computing resources. By migrating VMs, the workload can be consolidated on a smaller number of physical machines allowing for servers, or even for the entire operation nodes, to be completely shut down [129]. The advantage of migration is that the DC hardware computing resources are used more efficiently. In [108] the authors state that the usual candidate servers for resource consolidation are using less than 20% of their CPUs before consolidation and experience a dramatic improvement in terms of CPU utilization afterwards.

An important aspect related to virtualization and consolidation is the overhead introduced by workload migration. This depends on various factors such as VM size and network parameters. In [18] VM migration within the same DC accounted for 10W power consumption as overhead when using cold migration. When dealing with migration between federated DCs, network overhead is an important factor, too. A recent study [9] shows that the contribution of the network to the total cost, in CO2 emissions, can be significant especially when data travels over the internet due to the relatively high power demand of routers.

Virtualization and consolidation strategies are broadly accepted by many DCs to cut down the associated operational costs. In 2009 EMC reported 7.5 Million USD in energy saving over five years by virtualizing and consolidating its servers and data storage [4]. Unfortunately, this approach has also a drawback, since the virtual hardware abstraction within the VM engine induces latency [42]. For example, VMware's benchmark on ESX latency found an average I/O latency of 13% [134]. Because of performance and I/O concerns, VMs are not broadly utilized in some specific application domains which have extremely strict requirements for system performance such as High Performance Computing (HPC) centers [138].

Dynamic Power Management of IT resources

Dynamic Power Management (DPM) techniques allow for powering-down the unused DCs components or even entire servers, to reduce the power consumption. The power management techniques can be classified into software techniques and hardware techniques. Hardware techniques refer mainly to the design of power-efficient devices (CPUs, memory, various controllers) and high conversion-rate power supplies [120]. Software techniques on the other hand, deal with power management of hardware components (for example processor, memory, hard disk, or the entire server) by transitioning them into one of the several different low-power states when they are idle for a long period of time [129]. According to the strategy used for deciding to trigger a power state transition for a hardware component, three types of dynamic power management techniques are identified [91]: predictive techniques, heuristic techniques and QoS and Energy trade-offs. Predictive techniques employ predictive mechanisms to determine how long into the future the hardware component is expected to stay idle and use that knowledge to determine when to reactivate the component into a high power state [56]. Heuristics techniques use quickly accessible information for decision making. They have little overhead but in complex systems they are loosely applicable and sometimes their application can result in erratic performance degradation [75]. The QoS and Energy trade-offs techniques exploit the energy saving opportunities presented by narrowing the performance request levels of the running tasks [98].

Cooling subsystem

The cooling subsystem must remove all the heat generated by the servers in order to maintain the ambient temperature less than a threshold imposed by the producer. Most of nowadays datacenters have a cooling system based on recirculating air blown by a set of fans through chillers in the server rooms while supercomputing cooling is based on cooling liquid acting directly upon the CPU [99]. DC cooling subsystem is considered as the most important energy consumer and contributor to high DC operational costs. The percentage of total power consumption used by cooling subsystem alone in today's average size DCs can be as high as 50% [2].

The energy flexibility potential of the cooling system can be exploited in two ways depending on the duration period of enacting the flexibility. For short periods of time (up to 5 minutes or less according to [143]) pre-cooling or post-cooling the DC is used to shift flexible energy over a short time interval. Pre-cooling is a technique that over-cools the DC server rooms and then decreases or turns off the cooling system to save energy until the room reaches its initial temperature. Post-cooling is a similar technique that turns off cooling until the server room reaches its critical temperature to save energy and then overcools the room to restore the normal operating temperatures. For longer periods, Thermal Energy Storage (TES) devices are used to dynamically substitute the electrical cooling system thus using less electrical energy.

Thermal Energy Storage (TES)

TES is an energy storage system which relies on storing energy as ice or chilled water and is used to overcool the DC without using electricity. The flexibility mechanism works as follows: (i) the amount of energy discharged from the TES at moment t is compensated with an equal amount of energy saved by lowering the intensity of the electrical cooling system and (ii) the amount of energy charged in the TES at moment **t** is compensated with an equal amount of energy consumed by the DC by increasing the intensity of the electrical cooling system to over-cool the TES tanks.

Electrical Storage Devices (ESD)

The flexibility mechanism defined for the electrical storage devices (ESD), or batteries, works as follows: (i) the amount of energy discharged from the device at timeslot **t** is compensated with an equal reduction of the DC energy demand and (ii) the amount of energy charged in the device at moment **t** is compensated with an equal increase of the DC energy demand.

Network

The servers in a DC are interconnected through a network of switches and routers, organized in a topology of three layers, where the equipment from a layer is connected only to the adjacent layers [102]. At the highest level, we have the core switches that achieve the connection with the outer world. The core switches are connected to the mid-layer aggregation switches which at their turn are further connected to the lowest level from the hierarchy, the edge switches, that route information to physical servers. This tree topology has different implementation variants depending on the number of redundant links between equipment on adjacent layers. The Fat-tree topology [102] has the highest redundancy by connecting all the elements from a layer with the elements from the adjacent layer. The downside of this approach is that no energy saving can be performed by turning off a device. The Elastic-tree topology (opposite to Fat-tree topology), has only one link between each element. Energy saving can be obtained by using this topology, with the risk of providing a low reliability in case of failure because of the lack of redundant links.

Renewable Energy, Smart Grid and Marketplace

Renewable energy (REN) production systems, such as solar and wind power, have started penetrating the market and, due to their green nature, once available, they should have the highest priority for consumption in the electricity grid. In the recent years, large DCs began to be equipped with Local REN Sources, such as photovoltaic panels or wind turbines, thus being able to either generate electricity for their own purposes, or sell the energy on the local energy marketplace, becoming key players in their local energy ecosystem. Being of intermittent nature, REN requires the utility consumers, including DCs, to use the existing flexibilities such as delay-tolerant workload or workload migration to federated DCs in order to shape their energy consumption as to match the renewable energy generation and also to sell the energy surplus on the open energy market.

The energy marketplace is responsible for the energy trades aiming at scheduling a balanced energy flow between the energy generation and energy consumption units. The marketplace is organized in three models based on different time granularities: Day-ahead, Intra-day and Near-real time. In the Day-ahead model, the producers and consumers may sell/buy energy for the next day while in the Intra-Day model producers and consumers may adjust the contracted energy (consumed and produced) in the Day-ahead model), to accommodate the current day needs. Day-ahead and Intra-day are using an auctioning model based on bids and offers while Near-real time uses a model of real time balancing of the energy market by generating/consuming real time extra energy when needed.

Clients, including DCs that trade energy on these marketplaces have several trading sessions, most of them in advance of the actual energy consumption or production. Thus, they need prediction tools to forecast the amount of traded energy and operational planning/scheduling to assure that they will meet the energy demand. By using energy-shifting based flexibility mechanisms, DCs may become key players in their local energy ecosystem. Consequently, special green DC planners and schedulers based on predictions were developed to generate DC operational plans for specific periods of time.

Federated DCs

The Federation of DCs is usually modelled as Weighted Undirected Graphs [107], where the nodes represent the DCs and the edges represent the network interconnections between partner DCs. Several parameters can be represented as weights on the graph edges such as bandwidth, length or utilization. Other characteristics such as SLA, local REN generation and energy prices should be also represented into the model. An energy-aware DC scheduler, part of a network of geographically distributed partner DCs, should consider load migration scheduling strategies such as *follow the sun, follow the moon or follow the marketplaces of low energy price*.

7.4.4 Reactive and proactive scheduling/optimization techniques

An energy-efficient DC scheduler/optimizer can be seen as a self-adaptive system. As shown in the introductory section, according to relation (1), the energy-aware scheduler/optimizer determines a set of actions to be applied sequentially on the DC subsystems for each timeslot t in the scheduling/optimizing time window $W = \{1..T\}$. If $(T = 1)$, the scheduler/optimizer will determine the set of actions only for the transition into the next state. This scheduling/optimizing technique is of reactive type. If $(T > 1)$ the scheduler/optimizer will determine the actions for a set of future states. This scheduling/optimizing technique is of *proactive type* [74].

Reactive scheduling/optimizing techniques

Reactive scheduling/optimizing techniques know all their inputs before starting the computation of the appropriate actions, so they does not make any guesses about the future states. As a result, the adaptation process of the DC resources takes place after the monitored events have occurred. As major drawback, the reorganization of the resources as a result of executing the planned actions can occur only after some of the policies have been violated, thus energy being consumed in excess. Another disadvantage is that the resource reconfiguration can further affect the execution of the running application, especially if the adaptation time is long, thus leading to more violated policies. The most important reactive optimization techniques are *resource consolidation* and *time scheduling*. Basically, these techniques try to place the running tasks or VMs on the minimum number of DC servers while keeping SLAs, thus minimizing the overall energy consumption.

A virtual machine consolidation (VMC) algorithm consists of four main tasks [16]: (1) apply policies for detecting overloaded hosts, (2) apply policies for detecting under-loaded hosts, (3) apply VMs selection policies to select VMs to migrate from overloaded hosts, (4) apply VMs placement policies to allocate selected VMs to new hosts. Finally, the servers which remain idle after the consolidation are turned off. In order to select a destination server for a VM, several parameters are taken into account such as CPU, RAM, SLA, cooling, and network.

Proactive scheduling/optimizing techniques

Proactive scheduling / optimizing techniques use predictions to determine future inputs thus being able to prepare and trigger the adaptation in advance, before reaching the execution point that might violate the policies [22]. The accuracy of these techniques rely on the prediction module. An underestimate or overestimate of the predicted values may lead to wrong schedules. There are several prediction techniques reported in the literature, but the most used are *time-series* [74], [6] and *linear-regression* [61],[60],[59],[72]. Other prediction methods are based on *machine learning*. In the supervised machine learning category we notice the use of *artificial neural networks* to predict PUE in [68], *dynamic backfilling* to avoid policy violations in [31] or *k-nearest neighbor regression* to predict server resource usage [58]. Prediction techniques are used at various levels in the DC, from hardware level such as predicting the CPU utilization [61], to application level of VM utilization [60],[59] and overall DC energy consumption [69],[122], [68].

7.4.5 State of the art scheduling and optimizing techniques in energy-aware DCs

This section presents the most representative energy-aware scheduling/optimizing techniques reported in the literature.

State of the art scheduling and optimizing of IT resources

Resource provisioning

Resource provisioning of VMs using a variation of *bin packing problem* where the bin is usually represented by a VM, while the package is represented by a server is addressed in [28]. The incoming VMs is ordered using a modified *best fit decreasing algorithm* based on their request for CPU and allocating them to the fitting server with minimum power consumption. *Machine learning* based techniques to optimally distribute tasks in a DC by using a set of scheduling policies to reduce the number of unused servers in line with the workload needs in each moment are presented in [31] and [40]. *Biologically-inspired* heuristics is a large class of techniques for reducing the search space of NP scheduling problems. A *Swarm Intelligence* based solution for task allocation and scheduling is presented in [84]. It involves an *Ant Colony Optimization (ACO)* technique for discovery the appropriate computing resources in the grid using a set of distributed agents (mapped on ACO heuristics) working in parallel and independently of each other along with a constraint-based solver where both cost, in terms of energy, and execution time are used for the task allocation criteria. A novel meta-scheduler that allocates the VMs to servers to minimize the energy consumption based on a *Self-adaptive Particle Swarm Optimization technique* for VM provisioning is presented in [83]. An improved *genetic algorithm* with fuzzy multi-objective evaluation for VM placement based on resource utilization, power consumption and thermal dissipation is presented in [140]. Identifying patterns and making predictions about future workload trends can be made by *mining historical data*. A VM placement algorithm that forecasts the future workload trends from already collected historical workload levels is described in [35]. The forecasting technique is based on decomposing the workload in periodic patterns and residual components. The periodic patterns are determined using the average of multiple similar historical periods, while the residual workload is determined using a time-series analysis method. A VM allocation controller based on a workload predictor component is proposed in [30]. The forecaster component applies statistical prediction techniques on workload historical intensity levels and constructs a workload predictive model which is then used to optimally allocate the DC current workload based on its intensity levels and SLA requests. Nephele [121] is a *cloud specific* dynamic resource allocation framework which tries to assign discrete jobs to the right VMs. The main component is the Job Manager that is connected to the Cloud operator, which allows it to allocate or de-allocate VMs. Each job's task is executed by a Task Manager, which runs inside an instance (VM). Another *cloud specific* scheduling technique which sets priorities to user requests based on the type and cost of the resources and the time required for access is presented in [70]. This approach allows the cloud administrator to easily modify the parameters for prioritizing and have an overview of the cloud profit by allocating the available resources to certain requests. In [12], the resource allocation problem and resource exchange methodology are approached in the context of horizontal expansion of *heterogeneous cloud platforms*. The approach uses a multi-layered management framework aiming at relaxing the resource allocation complexity problem by decoupling it into three layers: (i) Application Management Layer (AML), with optimization done at running application granularity, (ii) Cloud Management Layer (CML), employing user virtualization for further optimizations, and (iii) Inter-Cloud Management Layer (ICML), where the optimization is achieved by advice managers.

Resource consolidation

Consolidation techniques can be classified in three types by considering the time granularity of the consolidation process [134]: static, semi-static, and dynamic. In the static consolidation, VMs are dispatched in a consolidated manner without any further migration [144]. This approach is recommended to be used in case of static workload that does not change over time. In semi-static consolidation, the server workload is consolidated once every day or every week [132]. The dynamic consolidation deals with workloads that frequently change in time. It requires a run-time manager that may deploy and migrate workload applications, turn on and off servers, or a combination of the above, as a response to workload variation [41]. In the following, we present the most relevant state-of-the-art approaches for dynamic consolidation.

To enable energy-efficient dynamic consolidation, the inter-relationships between energy consumption, resource utilization, and performance of consolidated workloads must be considered [131] as well as the energy-performance trade-offs for consolidation and optimal operating points [129]. As shown in [134], the greatest challenge for any consolidation method is to decide which workloads should be combined on a common physical server since resource usage, performance, and energy consumption are not additive. In case of resources consolidation, many small physical servers are replaced by one larger physical server, to increase the utilization of costly hardware resources such as CPU [82].

The problem of power and performance management in virtualized DCs, approached by dynamically provisioning VMs, consolidating the workload, and turning on and off servers is discussed in [95]. The appropriate consolidation decisions are selected by using a predictive, look-ahead control technique. An energy-aware scheduling and consolidation algorithm for VMs in DCs based on a fitness metric to evaluate the consolidation decision effectiveness is discussed in [127]. Biologically-inspired heuristics are often used as a mean to reduce the search space in dynamic VM consolidation. A swarm-inspired algorithm based on the Ant Colony Optimization meta-heuristic for solving a multi-dimensional bin-packing problem which represents the model of an energy-aware resource consolidations in DCs is proposed in [62]. ORM (Online Resource Management) [105] is a method for controlling the number of active servers in order to minimize the DC operational costs. The algorithm uses a discrete time model to divide the budgeting time interval into k time slots such that for each time period it can accurately predict future information (for example workload, renewable energy, and electricity price). The algorithm also uses a load distributor, which places the delay-tolerant jobs in a queue and serves the sensitive ones.

A novel consolidation solution was implemented as part of the EU FP7 GAMES (Green Active Management of Energy in IT Service centers) project [5]. The solution is based on *reinforcement learning technique* to create a context aware adaptive load balancing system capable of dynamically consolidating the datacenter by scaling up and down the allocated hardware resources in order to save energy [51]. The load balancer uses reinforcement learning to take its decisions starting from the current cloud snapshot and building a tree for determining the best adaptation action, each time the current entropy of the datacenter exceeds a predefined threshold. The cloud snapshots are constructed by collecting data from the physical cloud. In order to prove the effectiveness of the proposed technique, a Green Cloud Scheduler (GCS) [7] was implemented as an alternative load balancer for the default OpenNebula [10] scheduler. GCS decreases the amount of energy consumed in cloud DCs by intelligently scheduling and consolidating VMs aiming at reducing the number of used servers. For run-time optimization, GCS combines context-aware and autonomic computing techniques with control theory specific methods. The GCS operates as a MAPE loop, consisting of *Monitoring* (collecting and storing the energy consumption and workload related data), *Analysis* (analyses the current context, evaluates the cloud greenness level and decides if new actions are necessary for improving the DC energy efficiency), *Planning* (uses a reinforcement learning algorithm based on what-if analysis for determining the sequence of adaptation actions to improve the DC energy efficiency) and *Execution* (dynamically enforces at run-time the sequence of adaptation actions) phases. However, GCS does not take into account other optimization criteria and the energy related flexibilities of other DC subsystems such as IT facility, renewable energy production, energy storage facility or energy price.

State of the art scheduling and optimizing of cooling resources

Due to the large amount of energy consumption, the problem of workload scheduling for minimizing cooling requirements is frequently addressed in the literature. As shown in [14] and [45] there is a trade-off between consolidation requirements and energy-efficient cooling of the servers. Workload consolidation is advised in order to increase resource utilization ratio and turn off the unutilized ones, thus reducing the energy consumptions. As a result, the high density of workload after consolidation creates hot spots thus conflicting with the thermal management policies advocating the placement of workload across the servers in such a way as to minimize the generating heat. Seeking a balance between power management and thermal management conflicts becomes an important research issue. To solve the problem, a joint optimization technique, PowerTrade and Surge-Guard [14], aims to calculate the optimal trade-off between server idle power and cooling power in a subset of servers as to eventually reduce the sum of the idle and cooling power. Two other temperature-aware placement techniques, Zone-Based Discretization (ZBD) and Minimizing Heat Recirculation (MinHR) are proposed in [110]. A range of tools to enable DC designers, operators, suppliers and researchers to plan and operate DCs facility in a more efficient way is developed by the EU FP7 project CoolEmAll [3]. The main project outcome is to provide a simulation, visualization and decision support toolkit (Simulation Visualization and Decision Toolkit) that enables the analysis and optimization of IT infrastructures thus enforcing an efficient airflow, thermal distribution and optimal arrangement in a DC. The project combines efficient cooling methods and waste heat re-use with thermal-aware workload management at a single DC level, without considering RES usage, Smart Grid interaction or workload relocation in federated DCs.

State of the art scheduling and optimizing DC ecosystem

This section presents the state of the art research reported in the literature involving energy-aware scheduling/optimizing workload, resources and DC operation in an ecosystem perspective by considering own renewable energy sources (REN), interaction with Smart Energy Grids and Federation of DCs.

A novel approach which models the energy flow in a DC ecosystem is presented in [103]. The authors claim that the energy cost and the environmental impact is reduced by predicting energy production and workload followed by scheduling the workload and resources to minimize the energy needs of IT equipment and cooling systems. This is achieved through (i) demand shifting by allocating delay-tolerant workload such that the IT resources and cooling systems work at optimum capacity and (ii) using a DC cost model and a constrained optimization problem solved by the workload planner in order to determine the best energy demand shifting. This work defines a complex pricing model and takes into account both interactive and batch workloads. A DC management system that takes into account workload, cooling systems and renewable energy power supply in a network of DCs is presented in [123] and aims at maximizing the renewable energy consumption and minimizing the cooling power by relocating services (workload) between various DCs. The best DC location of each service is computed using Genetic Algorithms, workload placement being determined based on the level of renewable energy and the cooling conditions at each DC site.

GreenWare [141] is a system that tries to answer two questions often posed by DC operators: how to maximize the renewable energy by distributing services at various locations and how to achieve this relocation without exceeding the budget. GreenWare is a novel middleware system contributing to (i) dynamically dispatching service request among various DCs in different geographical locations based on energy levels and monthly budgets; (ii) defining an explicit model of renewable energy generation, such as wind turbines and solar panel, being able to predict energy levels and (iii) defining an optimization problem based on constraints to compute the optimal deployment solution of the services.

The All4Green EU FP7 [1] project investigates and proposes solutions, in the context of REN exploitation, for matching energy demand patterns of DCs on one hand and the energy production and supply patterns of the energy producers and providers on the other hand. As main ecosystem actors, All4Green projects considers the ICT users deploying services in the DC, electrical power providers and DCs cooperating in a federated way.

The most complex approaches aim at planning the operation of a DC in advance [69],[52]. A scheduler for planning workloads execution and for selecting the energy source to use at a given time among solar panels, batteries or grid is presented in [69]. The decisions are based on workload and energy predictions, battery level, workload analytic models, DC characteristics and grid energy prices. Another complex approach is present in the FP7 EU GEYSER [6] project that uses a proactive scheduler to optimize the DC operation. A set of proactive optimization loops with different time granularity compute the optimal actions for DC operations based on predictions made on monitored data. Each control loop aims to optimize the DC operation with one of the marketplace components: Day-ahead , Intra-day and Near-real time. The project implements several optimization strategies such as maximizing the usage of locally generated renewable energy [52], response to ancillary service demands [19] and trade energy on marketplace to obtain profit. In order to plan the operation of the DC, several flexibility mechanisms are used such as workload scheduling, leveraging electrical and non-electrical cooling, using batteries to shave power peaks and planning the diesel generator maintenance in periods with power shortage.

7.5 Energy aware interconnections

E. Niewiadomska-Szynkiewicz, A. Sikora, P. Arabas

The reduction of power consumption by a network infrastructure including both datacenter interconnect networks and wide area network is another key aspect in the development of modern computing clouds. Recently new solutions both in hardware and software have been developed to achieve the desired trade-off between power consumption and the network performance according to the network capacity, current traffic and requirements of the users. In general, new network equipment is more energy effective because of modern hardware technology including moving more tasks to optical devices. However, even greater savings are obtained by employing energy-aware traffic management and modulating the energy consumption of routers, line cards and communication interfaces. In the following subsections the techniques developed for keeping the connectivity and saving the energy that can be used for energy-efficient dynamic management in backbone and datacenter interconnect networks and LANs are surveyed and discussed.

7.5.1 Energy consumption of network devices modeling

the architecture of modern network devices in many aspects resembles that of general computers. Each network node is a multi-chassis device which is composed of many entities, i.e. chassis, cpu, memory, switching fabric, line cards with communication ports, power supply, fans, etc. The only specific element – switching fabric may be considered as a kind of specialized processor connected to common busses. The main difference is however in number and power consumption of line cards. While, in a general purpose PC there are from one to several network interfaces consuming only a fraction of power in an average switch or router there are usually from tens to hundreds network ports occupying several line cards. Therefore, a hierarchical view of the internal organization of network devices is employed in commonly used energy models. It is represented through several layers, namely a device itself (the highest level), cards (the middle level) and communication interfaces (the lowest level). Each component is energy powered. The hierarchical composition is important when adjusting power states is considered – obviously it is not possible to lower energy state of a line card without lowering energy states of its ports [116].

Let us consider a computer network formed by the following components: R routers ($r = 1, \ldots, R$), C line cards ($c = 1, \ldots, C$) and I communication interfaces $i = 1, \ldots, I$). As mentioned above, the hierarchical representation of a router is assumed, i.e., each router is equipped with a number of line cards, and each card contains a number of communication interfaces. All pairs of interfaces from different cards are connected by E links ($e = 1, \ldots, E$). In case of modern networks equipped with mechanisms for dynamic power management all network components can operate in K energy states (EASs) defined as power settings, and labeled with $k = 1, \ldots, K$. Two ports connected by the e-th link are in the same state k. In general, the corresponding power $P_{net}(q)$ consumed by a backbone network transforming a total traffic q can be calculated as a sum of power consumed by all network devices. In the network built of standard devices, the energy usage is relatively constant regardless of network traffic. The legacy equipment can operate only in two energy states: deep sleep ($k = 1$) and active with full power ($k = 2$). Thus, the corresponding power consumption $P_d(q)$ for total traffic q served by the network device d, i.e. a router ($d = r$), line card ($d = c$) or link connecting two interfaces ($d = e$)) can be described as follows [124], (see Fig. 7.2a):

$$P_d(q) = \begin{cases} P_{d1} & \text{if } q = 0, \\ P_{d2} & \text{if } q > 0, \end{cases} \tag{7.2}$$

where P_{d1} and P_{d2} denote fixed power levels associated to the device d in deep sleep and active states, respectively.

Novel devices equipped with mechanisms for dynamic power management – e.g. InfiniBand cards switching between 1x and 4x mode or bundled WAN links [39, 77] – can operate in a number of dynamic modes ($k = 1, \ldots, K$), which differ in power usage [116] (see Fig. 7.2b):

$$P_d(q, k) = \begin{cases} P_{d1} & \text{if } q = 0, \\ P_{dk} & \text{if } q > 0, \end{cases} \tag{7.3}$$

where P_{d1} denotes fixed power level associated to the device d in deep sleep state and P_{dk} a fixed power level in the k-th active energy state, $k = 2, \ldots, K$.

The 802.3az standard [78] defines the implementation of low power idle for Ethernet interfaces. Numerous formal models that describe the correlation between an amount of transmitted data and energy consumption are provided in the literature. The simplest approximation is a piece-linear modification of (7.2), see Fig. 7.2c:

$$P_d(q) = \begin{cases} P_{d1} & \text{if } q = 0, \\ \alpha_d + \gamma_d q & \text{if } q > 0, \end{cases} \tag{7.4}$$

where α_d denotes a constant offset, γ_d a coefficient defining the slope of approximated characteristic of the power consumed by the device d. The results of application of piece-linear energy models for traffic engineering are presented in [80]. More precise models require usage of nonlinear functions to model power consumption (e.g. logarithmic, see [71]) or combine (7.4) with (7.3) to describe the effects of switching energy states [38].

Finally, the energy model in which a given device can operate in a number of energy states and the correlation between consumed energy and provided throughput is modelled by a piece-linear function can be formulated (see Fig. 7.2d):

$$P_d(q, k) = \begin{cases} P_{d1} & \text{if } q = 0, \\ \alpha_{dk} + \gamma_{dk} q & \text{if } q > 0, \end{cases} \tag{7.5}$$

where α_{dk} and γ_{dk} denote coefficients for k-th active energy state, $k = 2, \ldots, K$.

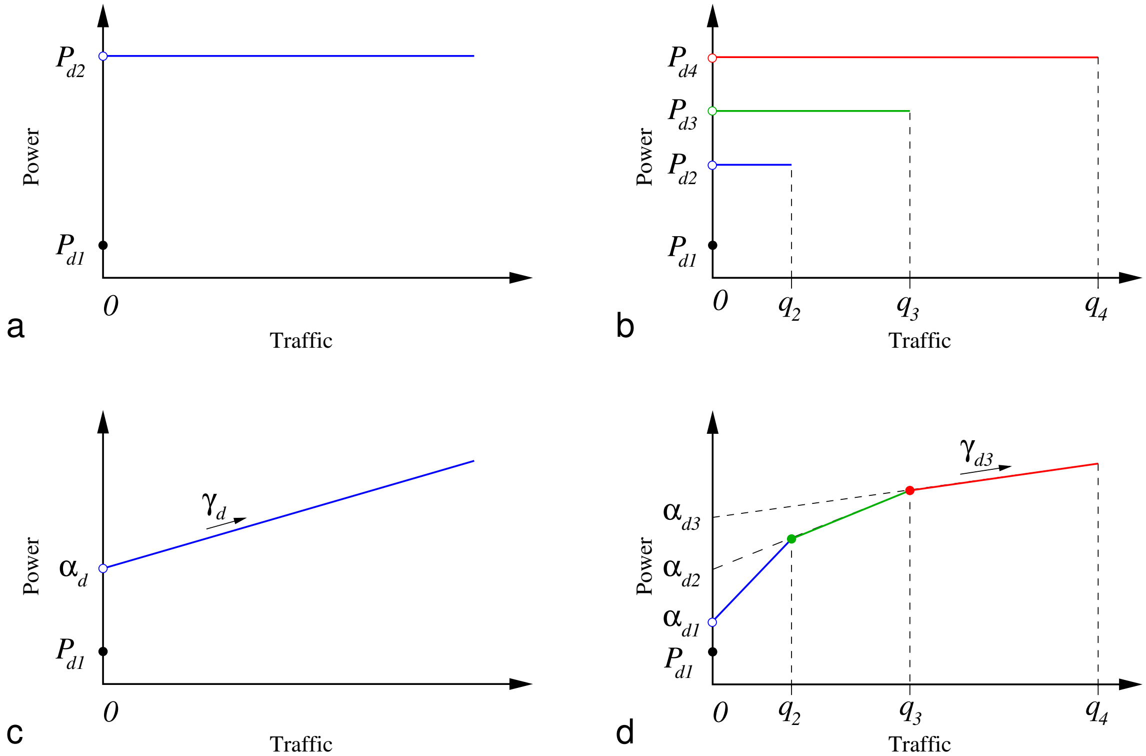

Fig. 7.2 Power consumption models a) two energy states, b) multiple energy states, c) piece-linear approximation, d) multiple energy states and piece-linear approximation.

7.5.2 Low energy consumption backbone networks

Network energy saving problem formulation

Energy consumption trends in the next generation networks have been widely discussed and the optimization of total power consumption in today's computer networks has been a considerable research issue. Apart from improving the effectiveness of network equipment itself, it is possible to adopt energy-aware control strategies and algorithms to manage dynamically the whole network and reduce its power consumption by appropriate traffic engineering and provisioning. The aim is to reduce the gap between the capacity provided by a network for data transfer and the requirements, especially during off-peak periods. Typically backbone network infrastructure is to some degree redundant to provide the required level of reliability. Thus, to mitigate power consumption some parts of the network may be switched off or the speed of processors and links may be reduced. According to resent studies concerning internet service providers networks [33, 125], the total energy consumption may be substantially reduced by employing such approaches. The common approach to power management is to formulate an optimization problem that resembles traditional network design problem [119] or QoS provisioning task [79, 106], but with a cost function defined as a sum of energy consumed by all components of the network. In contrary to the traditional network design problem, data transfers should be aggregated along as few devices as possible instead of balancing traffic in a whole computer network. The major drawback is complexity of the optimisation problem that is much more difficult to solve than typical shortest path calculation task. The complexity has roots in NP-completeness of flow problems formulated as mixed integer programming, dependencies among calculated paths and requirement for flow aggregation. Furthermore, energy consumption models are often non-convex making even continuous relaxation difficult to solve and introducing instability suboptimal solutions [133].

Various formulations of a network energy saving problem are provided and discussed in the literature; starting from a mixed integer programming (MIP) formulation to its relaxation in order to obtain a continuous problem formulation and employ simple heuristics. The aim is to calculate optimal energy states of all network devices and corresponding flow transformation through the network. In general, due to mentioned above high dimensionality and complexity of the optimisation problem linear energy models are preferred. Some authors limit the number of energy states of network equipment (routers and cards) into active and switched off and use energy model (7.2), [47]. Furthermore, they propose to use multi-path routing, which is typically avoided. However, the recent trend in green networking is to develop devices with the ability to independently adapt performance of their components by setting them to one of a number of energy-aware states (energy model (7.3) or (eq:multilinear)) [37, 36]. It is obvious that modeling such situation implies larger dimensionality of the optimisation problem and more complicated dependencies among network components. Another approach is to exploit properties of optical transport layer to scale link rates by selectively switching off fibres composing them [64] or even build two level model with IP layer set upon optical devices layer [77]. L. Chiaraviglio et al. describe in [47] an integer linear programming formulation to determine network nodes and links that can be switched off. J. Chabarek et al. in [43] solve a large mixed-integer linear problem for a specific traffic matrix to detect the idle links and line cards. The common solution is to aggregate nodes and flows to decrease the dimension of the optimisation problem [47]. P. Arabas et al. describe in [20] four formulations of the network energy saving problem, starting from an exact MIP formulation, including complete routing and energy-state decisions, and presenting subsequent aggregations and simplifications in order to obtain a continuous problem formulation:

LNPb: Link-Node Problem: a complete network management problem stated in terms of binary variables assuming full routing calculation and energy state assignment to all devices and links in a network.
LPPb: Link-Path Problem: a formulation stated in terms of binary variables assuming predefined paths (simplification of LNPb).
LNPc: Link-Node Problem: a complete network management problem stated in terms of continuous variables assuming full routing calculation.
LPPc: Link-Path Problem: a formulation stated in terms of continuous variables assuming predefined paths (simplification of LNPc).

Given the notation from section 7.5.1, a complete network management problem LNPb stated in terms of binary variables k assuming energy state assignment to routers $(k = 1, \ldots, K_r)$, line cards $(k = 1, \ldots, K_c)$ and communication interfaces $(k = 1, \ldots, K_e)$, and full routing calculation for recommended network configuration can be formulated as follows

$$\min_{x_{rk}, x_{ck} x_{ek}, u_{ed}} \left[P_{net}^{LNPb} = \sum_{r=1}^{R} \sum_{k=1}^{K_r} P_{rk} x_{rk} + \sum_{c=1}^{C} \sum_{k=1}^{K_c} P_{ck} x_{ck} + \sum_{e=1}^{E} \sum_{k=1}^{K_e} P_{ek} x_{ek} \right], \tag{7.6}$$

subject to the set of constraints presented and discussed in [115, 116]. The power consumption and corresponding throughput are presented in Fig. 7.3:

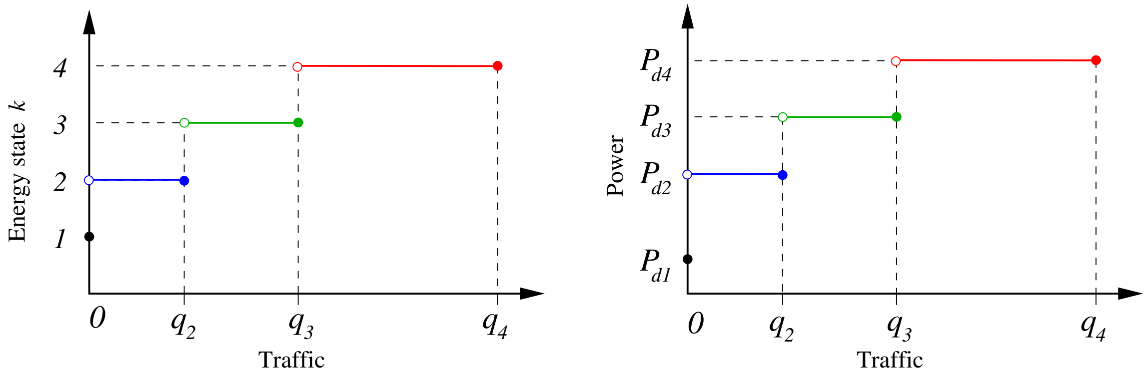

Fig. 7.3 Power consumption and throughput (various energy states); stepwise energy model.

The objective function P_{net}^{LNPb} is the total power consumed by all network devices calculated for energy models of all network devices expressed by (7.3). Variables and constants used in above formulas denote: $x_{rk} = 1$, $x_{ck} = 1$, $x_{ek} = 1$ if the router r, card c, link e, respectively is in the state k (0 otherwise), $l_{ci} = 1$ if the interface (port) i belongs to the card c (0 otherwise), $u_{ed} = 1$ if the path d belongs to the link e (0 otherwise). P_{rk}, P_{ck}, P_{ek} denote the fixed power consumed by router, card and link in the state k. To calculate the optimal energy states of network equipment the optimisation problem (7.6) has to be solved for number of assumed demands ($d = 1, \ldots, D$) imposed on the network and transmitted by means of flows allocated to given paths. The presented above formulation requires predictions of the assumed rate V_d of each flow d that is associated with a link connecting any two ports: ports of the source and the destination nodes for the demand d.

The problem formulation obtained after flows aggregation (LPPb) is provided in [20]. Although LPPb is easier to solve due to smaller number of constraints, but is still too complex for medium-size networks. As stated previously the widely used direction to reduce complexity of a network optimization problem is to transform the LNPb problem stated in terms of binary variables to the LNPc with continuous variables x_{rk}, x_{ck}, x_{ek}, u_{ed}

$$\min_{x_{rk}, x_{ck}, x_{ek}, u_{ed}} \left[P_{net}^{LNPc} = \sum_{r=1}^{R} \sum_{k=1}^{K_r} P_{rk} x_{rk} + \sum_{c=1}^{C} \sum_{k=1}^{K_c} P_{ck} x_{ck} + \sum_{e=1}^{E} \sum_{k=1}^{K_e} P_{ek} x_{ek} \right] \tag{7.7}$$

and solve it subject to the set of constraints presented and discussed in [116]. In the above formulation the energy consumption and throughput utilization of the link e in the state k are described in the form of incremental model. The current values of fixed power consumption (P_{ek}) and the throughput of the link e (q_{ek}), both in the state k are calculated as follows: $P_{ek} = P_e(k) - P_e(k-1)$ and $q_{ek} = q_e(k) - q_e(k-1)$; where respectively $P_e(k)$ denotes power used by the link e in the state k and $q_e(k)$ denotes load of the link e in the state k. Furthermore, it is assumed that a given link can operate in more than one energy-aware state. The additional constraints are provided to force binary values of variables x_{rk}, x_{ck} in case when x_{ek} takes a binary value. The efficient heuristic to solve relaxed optimization task (7.7) was developed and presented in [116].

The energy-aware network management problem defined above has an important disadvantage. The vector of assumed flow rates V_d of flows $d = 1, \ldots, D$, being crucial for the problem in practice is difficult to predict. When operating close to capacity limits of the network, which is often the case, a poor estimation of V_d can lead to the infeasible solution. The two criteria optimization problem defined in [81, 89] use utility functions instead of a demand vector. The modified formulation of the optimisation problem is presented below. The modification consists in relaxing flow rates denoted by v_d. The original objective function P_{net}^{LNPb} (7.6) is augmented with a QoS related criterion Q_d, which represents a penalty for not achieving the assumed flow rate V_d by the flow d. $Q_d(v_d)$ is a convex and continuous function, decreasing on interval $[0, V_d]$. It is reaching minimum (zero) at V_d, the point in which user expectations are fully satisfied. Finally, a two criteria - i.e., reflecting energy costs and QoS - mixed integer network problem of simultaneous optimal bandwidth allocation and routing is formulated as follows

$$\min_{x_{rk}, x_{ck}, x_{ek}, u_{dl}, v_d} \Bigg\{ P_{net}^{2C} = \alpha P_{net}^{LNPb} + (1-\alpha) \sum_{d=1}^{D} Q_d(v_d) =$$

$$= \alpha \left[\sum_{e=1,3,5,\ldots}^{E-1} \sum_{k=1}^{K_e} P_{ek} x_{ek} + \sum_{c=1}^{C} \sum_{k=1}^{K_c} P_{ck} x_{ck} + \sum_{r=1}^{R} \sum_{k=1}^{K_r} P_{rk} x_{rk} \right] +$$

$$+ (1-\alpha) \sum_{d=1}^{D} Q_d(v_d) \Bigg\}, \tag{7.8}$$

subject to a set of constraints defined and discussed in [89]. In the above formulation $\alpha \in [0, 1]$ is a scalarizing weight coefficient, which can be altered to emphasize any of the objectives.

Control frameworks for dynamic power management

Various control frameworks for resource consolidation and dynamic power management of the backbone network through energy-aware routing, traffic engineering and network equipment activity control have been designed and investigated [39, 116, 114, 87, 34]. They utilize smart standby and dynamic power scaling, i.e., the energy consumed by the network is minimized by deactivation of idle devices (routers, line cards, communication ports) and by reduction of the speed of link transfers. The implementation of such framework providing two variants of network-wide control, i.e., centralized and hierarchical (with two coordination schemes) both with central decision unit is described and discussed in [116]. In presented approach it is assumed that network devices can operate in different energy-aware states (EAS), which differ in the power usage (energy model (7.3)). The implementation of both centralized and hierarchical frameworks provide two control levels:

- Local control level – control mechanisms that are implemented in the network devices level.
- Central control level – network-wide control strategies implemented in a single node controlling the whole infrastructure.

The objective of the central unit is to optimize the network performance to reduce power consumption. The decisions about activity and power status of network equipment are determined by solving LNPb or LNPc problems of minimizing the power consumption utilizing a holistic view of the system. Each local control mechanism implements adaptive rate and low power idle techniques on a given networking device.

The outcomes of the levels of the control frame depend on the implementation. In the centralized scenario the suggested power status of network devices are calculated by the optimization algorithm executed by the central unit, and then sent to adequate network devices. Furthermore, the routing tables for the MPLS protocol for recommended network configuration are provided. Hence, in this scenario, the activity of each local controller is reduced to comply with the recommendations calculated by the central unit, taking into account constraints related to current local load and incoming traffic. In the hierarchical scenario the central unit does not directly force the energy configuration of the devices. The outcome of the central controller is reduced to routing tables for the MPLS protocol that are used for routing current traffic within a given network. The objective of the local algorithm implemented in the devices level is to optimize the configuration of each component of a given device in order to achieve the desired trade off between energy consumption and performance according to the incoming traffic and measured load of a given device.

Utility and efficiency of control frameworks employing LNPb and LNPc schemes for calculating optimal energy states of devices for small-, medium- and large-size network topologies were verified by simulation and by laboratory tests. The test cases were carried on a number of synthetic as well as on real network topologies, giving encouraging results. The results are presented and discussed in [116].

In the presented control frameworks the total power utilized in a network for finalizing all required operations is minimized and end-to-end QoS is ensured. However, both in centralised and hierarchical variants the holistic view of a network is utilized to calculate the optimal performance of a system. Furthermore, most calculations are conducted on a selected node. Due to networks scalability and reliability distributed control is recommended. Distributing energy-aware control mechanisms extend existing mechanisms – typically routing protocols e.g. OSPF and MPLS [50, 34, 55], BGP and MPLS. Important profit of close cooperation with signaling protocols is that the observed state of the network may be used to estimate flows and to reconstruct the traffic matrix [46].

An agent-based heuristic approach to energy-efficient traffic routing has been presented in [87]. Identical agents are running on top of OSPF processes, that activate or deactivate local links, and the decisions are based on common information about whole network state. Individual decisions affect in turn OSPF routing. Simulations show that perfect knowledge about the origin-destination matrix improves energy savings not much more than when simple local heuristics are applied by agents. On the other hand, imperfect information about origin-destination matrix can make the result worse than in case when there is no energy-saving algorithm running at all. The proposed approach is viable to implement on any routing device, through command line and basic OSPF protocol.

Bianzino et al. describe in [34] the distributed mechanism GRiDA (Green Distributed Algorithm) adopting the sleep mode approach for dynamic power management. GRiDA is an on line algorithm designed to put into sleep mode links in an IP-based network. Contrary to the centralised and hierarchical control schemes utilizing LNPb and LNPc for energy saving in computer networks, GRiDA does neither require a centralized controller node, nor the knowledge of the expected traffic matrix. This solution is based on a reinforcement learning technique that requires only the exchange of periodic Link State Advertisements (LSA) in the network. Thus, the switch off decision is taken considering the current load of incident links and the learning based on past decisions. GRiDA is fully distributed among the nodes to: (i) limit the amount of shared information, (ii) limit coordination among nodes, and (iii) reduce the problem complexity. It is a robust and efficient solution. The simulation results of its application to various network scenarios are presented and discussed in [34].

7.5.3 Datacenter interconnect network

The networks in datacenters and cluster applications must fill specific requirements of speed, latency and reliability. Another important goal is to provide clear cabling structure allowing to pack equipment in cabinets effectively. Typical topologies are highly regular, usually hierarchical. The most dominant technologies are Ethernet – 1Gb/s, 10Gb/s or higher speeds and InfiniBand [29]. Using consistent technology across datacenter allows to build switched network limiting delays and complexity. Usually networks are constructed with three layers of switches. Lowest level switches (ToR – *Top of the Rack*) are installed in the same cabinet as the servers connected directly to them. To attain high reliability and multiply bandwidth computers use more than one network interface connected to different switches. Similarly ToR switches connect with two upper level – aggregation devices. The pairs of aggregation switches are interconnected using multiplied links to allow each of them replace another in case of malfunction. Then, thanks to limited number of devices the aggregation layer can be interconnected in full mesh via the highest layer – core switches [53].

Resulting multiple tree topologies are substantially redundant and allow to separate traffic using paths leading through disjoint physical links to attain high throughput, reliability and, if necessary security. On the other hand such architecture implies relatively high energy consumption. It seems natural that it can be reduced in periods of lower load by switching off some links and possibly switches. Furthermore as a whole network is managed by the same institution and collocated with the rest of equipment usage of centralized controller is possible and in many aspects favorable. Unfortunately large number of nodes and links makes mathematical programming task (e.g. of the form known from [119]) formidable and impossible to solve in acceptable time.

The described above topology offers high reliability and bandwidth on the cost of using costly and energy exacting provider grade equipment in upper layers. Recent works propose a number of simpler topologies allowing use of commodity switches at least at two lower layers. Moreover, limited number of redundant links and regular topology makes it possible to apply effective heuristic algorithms to manage power consumption. An example of such topology may be combined Clos and fat-tree topology proposed in [15]. The authors use large number of commodity switches in all layers to multiply their capacity (in terms of number of ports and bandwidth) as well as bisection bandwidth of the whole network. Another approach is to use highly specialized equipment and flatten network structure to reduce diameter like in Flattened Butterfly topology proposed in [13]. The idea is to use switches with relatively high number of ports linked along several dimensions. In a simplest case a group of n switches forms a dimension, each of them connects to $n-1$ switches in all dimensions it belongs to, it also serves n computing nodes. The topology is flat in the sense that longest shortest path (diameter) may be shorter than in fat-tree network (3 vs. 4 when basic examples are considered). Despite complicated cabling the network requires less equipment allowing to save energy. Further savings are possible using link rate adaptation assisted by adaptive routing.

7.6 Offloading techniques for mobile devices

G. Mastorakis, C. Mavromoustakis

Towards the great expansion of the applications, which are used in mobile devices (e.g., smartphones and tablets) and the rising approach of next generation computing infrastructures, Mobile Cloud Computing has been put forward to be an emerging technology for mobile services, towards providing full advantages, as well as offering new types of services and capabilities. Whereas mobile users have gained maturity in using a variety of services from mobile applications (e.g., iPhone apps and Google apps), which run on the devices and/or on remote servers via wireless networks, mobile devices are still confronting concerns, including their limited resources that obstruct the progress of service qualities. Cloud is acknowledged to have brought forward a new generation of computing, whereas cloud computing upgrades mobile systems with computing capability, and mobile users are provided with application services through the Internet. In this context, this part of the report surveys the current literature on the above-mentioned emerging research paradigms.

Mobile terminals and wireless communications seem to come up against significant advances. Applications, which were designed for desktop PCs are now demanded in mobile devices. Apple's iPhone commercial âĂIJThere's an app for everythingâĂİ is evidenced when using smartphones and other mobile devices to watch videos, to use online banking, to browse the Internet, use GPS and online maps, send emails, etc. As more apps are coming out, the more eager we become to install and experience them all. The size and weight of these applications in combination with the mobile devices' network capacity and battery life, though, makes that difficult to be accomplished. So we it is not possible to use the device to the full when we keep worrying about charging it, or for how many phone calls and videos it will last, as all the above are affecting the battery lifetime [25]. The performance of compute-intensive applications on limited-resource devices, except for the great amount of time might also take a large amount of energy. Mobile Cloud Computing (MCC) is able to reduce this running cost with the support of Cloud Computing. MCC achieves to combine Cloud Computing with the mobile environment and to predominate in advantages regarding performance (e.g., battery life, storage), environment (e.g., scalability, and availability), and security [57] issues, which are examined in mobile computing. While it provides data processing and storage services in clouds and the processing of all the complicated computing modules is done in the clouds, mobile devices can be discharged of powerful configuration (e.g., CPU speed and memory capacity). Mobile systems coupled with the cloud, not only provide users with a data backup on-the-fly, but also manage to alleviate smartphones' battery consumption. When battery is one of the main concerns for mobile devices, this survey is examining how MCC can give a fine solution to extending battery lifetime.

There is a need to intelligently manage the disk and screen, and enhance the CPU performance to reduce power consumption without the necessity of any changes in the hardware or the structures of mobile devices which can be proved unattainable and costly. Various studies, including a 2009 Nokia poll, classify longer battery lifetime to be on top of all other features (e.g., storage, camera) in mobile systems like smartphones. There are many applications which, although can run on mobile systems, they may overload them. In exchange for a good quality performance, computations of such applications should be performed in the cloud for avoiding consuming significant amounts of energy. In this context, the rest of this sub-chapter provides the advantages of Cloud Computing and the different techniques which are used. Then, it presents how cloud-based applications can affect the battery of a mobile device and under which circumstances. Next, the connection between Cloud Computing and Virtualization is outlined and finally, section is summarized and concluded.

7.6.1 Computation Offloading

An important issue of the integration of CC and mobile networks, in computing side of MCC, is computing offloading. Amongst many features of MCC, offloading is one of the main that can assist mobile devices to improve battery lifetime, and the application's performance. The execution time of an application on mobile devices may take long resulting in large amount of power consumption. This problem can be alleviated with the beneficial technique of Computation offloading where the complex processing and large computations can be migrated from resource-limited devices (i.e., mobile devices) to resourceful machines (i.e., servers in clouds). This specific technique is proved, in several experiments, to be able to save energy significantly. Specifically, up to 45% of energy computation can be reduced for large matrix calculation [126]. In this framework, it is possible to reduce up to 41% of energy consumption, by offloading a compiler optimization for image processing, and up to 27% of energy consumption in computer games, by migrating mobile game components to servers in the cloud. With cloud computing the amount of data, which will be offloaded is already stored on the server and not required to be sent over the wireless network. It is possible to undergo computation in the cloud just by sending a pointer to the data. This assumption applies in services such as Google's Picasa and Amazon S3, which can store data and afterwards perform computation on them.

Efficient and dynamic offloading, on the other hand, are included in many related issues that exist in different environments. Experiments have proven that energy saving is not always efficiently achieved by offloading [126]. For example, for the sake of compiling a small-sized code, more energy might be consumed in offloading than in local processing. So, in order to improve the energy efficiency, mobile devices have to resolve if offloading is the way to achieve it and, thereby, take into account which portions of the application's codes to offload. Offloading in a dynamic network environment (e.g., changing connection status and bandwidth) seem to have effective approaches. There are cases in which the transmitted data fail to reach the destination, or the data executed on the server are lost when it has to be returned to the sender. Where environment changes are responsible for such issues, an approach in [94], appears to study the case of eliminating computation all together, in which case, the mobile system is no longer responsible of performing the computation. The authors consider chess game and image retrieval for providing an example of the benefits from offloading in applications. By representing the chess game state in bytes, and by examining this value with a server's speed and the amount of computation needed, makes it amply clear that most wireless networks benefit from offloading. Looking at the same values and by examining them in the image retrieval application in mobile devices, where a lot of computation is done in advance, shows that offloading saves energy in cases of high bandwidth. Although further analysis shows that the energy saved seems to depend on the wireless bandwidth, the amount of computation to be performed, and the amount of data to be transmitted, existing studies are more interested in how these factors relate to each other and in choosing offloading computation by predicting them.

To offload a mobile application to a remote cloud presupposes a high connectivity scenario, otherwise the system would fail. Experiments in [109] show that the appropriate use of nearby resources relieves both memory and processing constraints in mobile devices. Thereat, many related researches have been exploring computing frameworks in order to test their feasibility on using local resources. A 'mobile cloud computing framework' has to detect the resources and exploit them accordingly, and operate dynamically with runtime resources along with their diverse connectivity. At same time, there should be qualifications of supporting both high and low end devices. Such an implementation is a mobile cloud computing framework [63], which has a purpose of testing sharing workload at runtime. They try to face challenges like detecting a potential cloud device, mobility, job partitioning and distribution in the cloud, and connectivity options. According to this approach, a mobile cloud is described as a 'cloud' of local resources, conducive to a common goal. Different individuals would be able to use such local cloud resources, which would be typically mobile, such as laptops, mobile phones, tablets etc. When a mobile phone tries to run an application with much computational power, it would result in draining the battery lifetime. A mobile cloud computing framework will enable the user to set up a 'virtual cloud' of local computational resources that exist nearby. As offloading computational jobs to the local 'mobile cloud' could be a solution for battery and size issues, they propose other mobile devices to be included to the local cloud, for avoiding additional infrastructure in order to enable mobility.

Dynamic partitioning can address some issues which are formed in MCC by workloads and the heterogeneity of environments. The act of dynamical instantiation of what processing to do at the cloud and at the device should be according to the different input content, and size. On weak devices, applications are structured in a way of being statically partitioned between a server running in the cloud, and the weak device. In one exemplary model of an application partition (e.g., on Facebook and Tweeter), the server(s) in the cloud does most of the application's processing, and the end-user device(s) acts like a thin client running simple tasks such as UIs. Another way which differs in execution time and energy consumption, is to perform most of the application's (e.g., in interactive graphic-intensive games) processing at the client. As the inputs and the environments in which applications run vary, there are many partitioning choices. There is no single partitioning that fits all networks, devices, clouds and workloads. For example, using different partitioning between device and cloud for an application may differently affect the performance on an Android netbook, and the performance on an Android phone. An adaptive system proposed in [48], is capable of dynamically choosing different partitioning between weak devices and clouds, in different environments. Dynamic partitioning of applications between weak devices and cloud has been examined, and proved to be beneficial for better supporting applications running in different devices, and in different environments as well. Thus, dynamic partitioning is a proposed solution, believed to play an important role in future mobile cloud computing.

Although many experiments have focused on energy efficiency of mobile devices, the main concern was reducing data center energy costs. The energy consumption when using cloud computing on mobile devices remains a matter of further analysis. In contrast with most prior work, a research [112], which focused on such constrained devices, explored how cloud-based applications have an impact on battery life, and under which circumstances. The authors in this work attempted to compare three different cloud and non-cloud applications running both locally and on a remote server, with different requirements in amounts of communication and processing. As reported by the results, multimedia applications of intensive computation, with significant local resources requirements, characterize mobile computing to be most energy-inefficient, as their cloud version consumes more energy wherever they are run. For applications which are computation-intensive only when run locally, such as a chess game, mobile computing proved to be most energy-efficient. They noticed that when chess was played in a cloud-based scenario, the mobile device required more communication but less computation, as it was done remotely with a powerful server. Moreover, results indicate that when cloud-based word processing applications use a considerable amount of resources for local execution, but not a lot of communication, are at least as energy-efficient as non cloud-based ones. In the same study, device form-factor is marked as another important matter when considering the energy efficiency of cloud-based applications. Energy consumption of cloud computing must be examined for a variety of form-factors in mobile devices. It is demonstrated by the results that cloud-based applications are more beneficial, in terms of communication's energy consumption, for larger form-factor devices. They believe that the reason for that is that as all cloud-based applications use the WiFi interface, the energy consumed in smartphones is greater than in laptops.

7.6.2 Virtualization and cloud computing

Microsoft, Google, Yahoo!, IBM, Amazon and Sun, are a paradigm of how large companies engage with the advantages of what cloud computing has to offer to academic and industrial community amongst others. In this regard, different cloud platforms are developed to allow consumers and enterprises use services to access cloud resources. Recently, virtualization technology is developing rapidly. Live migration, resource isolation, such as server consolidation, are some of the reasons why virtualization is preferred by more computing environments in order to support cloud computing. Except from the traditional suspend/resume migration there is the widely used strategy of live migration of virtual machines, in which the virtual machine can respond during all the time of the clients' migration process. As opposed to the former, the implementation of this strategy can provide load balancing, energy saving, and online maintenance. Single virtual machine migration competes with the increasing usage of live migration of multiple virtual machines in maximizing the live migration efficiency. In [142], the authors present a series of experiments on a live migration framework of multiple virtual machines which investigates some resource reservation methods in the live migration process between a source and a target machine, along with an evaluation of other complex migration strategies. According to their discoveries, they propose three optimization methods to improve the migration efficiency.

- Optimization in the source machine, by dynamically adjust memory and CPU resources in the source machine,
- Parallel migration of multiple virtual machines, when system resources in the source machine are sufficient, and
- Workload-aware migration of multiple virtual machines, by examining the target machine's virtual machine workload.

Content-based image retrieval (CBIR) is recommended from several studies, for battery powered devices which need energy conservation. It allows users to access specific sets of images and not manually browsing through all of them. By being offloaded with a high wireless bandwidth, is rendered more energy efficient. In [112] the authors include a design of an algorithm, called GreenSpot, coupled with an application comparison metric, called AppScore, which was implemented as an 'app' on Android OS for smartphones in order to assist a device to choose between a cloud and a non-cloud version of an application, based on the features and the energy-performance tradeoffs. A program partitioning for offloading, a fine suggestion of Kumar and Lu [94], is capable of calculating the trade-off between the computation costs affected by the computation time and communication costs which depend on the size of transmitted data and the network bandwidth, before its execution. As information including the communication requirements and/or the computation workload may differ, a program partitioning should make optimal decisions at a runtime dynamically. An approach in [100] for properly deciding about the tasks of procedure calls, presents a partition scheme surrounded by information also about computation time and data sharing in parallel to procedure calls which uses a cost graph and an algorithm in order to offload computational tasks on mobile devices. A similar approach in [44] refers to decisions about offloading components of Java programs. However, such approaches don't address to diverse applications. A different computation offloading scheme on mobile devices is introduced by Wang and Li [135], which approves tasks to be partitioned at any level (e.g., a basic block, a loop, or even a function), by having the privilege of mapping all physical memory references into the references of abstract memory locations. In this case, there are distributed subprograms which run on a device and a server, and an abstraction of the original program divided in clusters which, in combination with an algorithm, manage to find the optimal partition with a view to minimize the execution cost. An example of an automatic distributed partitioning system, called Coign, is introduced by Hunt and Scott [76]. An assertion of Li et al. [100], presents that energy of mobile devices can be saved by offloading multimedia code and, by that, game playing time can be increased. Projects like MAUI [54] and the CloneCloud [49] have focused on offloading computing tasks to cloud. These projects, however, are evidenced to be lacking in scalability and in applications/inputs/ environmental conditions. ThinkAir [93] is a new mobile cloud computing framework which addresses these voids, and makes the migration of smartphone applications to the cloud simple to developers. It is claimed to be innovative as it performs on-demand resource allocation efficiently, and dynamically creates virtual machines in the cloud, resumes and destroys them, when necessary. The importance of on-demand resource allocation is based on the fact that there are different workloads and deadlines for tasks, hence, different computational power. It also focuses on the important issue of parallelism, achieving in enhancing mobile cloud's power by providing a new infrastructure of execution offloading.

By combining CC and mobile web, mobile users have an extremely useful tool to access applications and services on the Internet. It is rather difficult to take full advantage of the benefits that mobile computing has to offer, due to resource deficiency. Problems around finite energy source, with poor resources, don't allow the user to be fully supported with the devices' mobility. In Mobile Cloud Computing, offloading is a feature that can effectively support mobile devices with energy saving and better applications performance. The heterogeneity of devices platform, network types, cloud conditions, as well as workloads are issues confronted by dynamic partitioning. It is also trusted as a promising solution, and an important topic in future mobile cloud computing. Îd'he numerous benefits from transposing computing to the cloud may be different in mobile systems, where computing for mobile users is limited due to wireless bandwidth and limited energy. Energy savings can be provided to mobile users as a service, but it also raises some unique challenges. When cloud-based applications are more beneficial for larger form-factor devices, multimedia applications of intensive computation find mobile computing to be most energy-inefficient, as their cloud version consumes more energy. When virtualization is acknowledged to support cloud computing, live migration of multiple virtual machines advances it even more by maximizing the live migration efficiency.

References

[1] (????) All4green. http://www.all4green-project.eu/
[2] (????) Calculating total power requirements for data centers. http://www.apcmedia.com/salestools/vavr-5tdtef/vavr-5tdtef_r1_en.pdf
[3] (????) Coolemall. http://coolemall.eu/
[4] (????) Emc 2009 sustainability report - advancing the journey. http://www.emc.com/collateral/magazine/h7215-sustainability-long-bro-hr.pdf
[5] (????) Games. http://www.green-datacenters.eu/
[6] (????) Geyser. http://www.geyser-project.eu/
[7] (????) Green cloud scheduler. http://community.opennebula.org/ecosystem:green_cloud_scheduler
[8] (????) The green grid. http://www.thegreengrid.org/~/media/TechForumPresentations2010/UnusedServerStudy.ashx?lang=en

[9] (????) Greenitamsterdam. http://www.greenitamsterdam.nl/wp-content/uploads/2013/06/Transporting-Bits-or-Transporting-Energy-does-it-matter-May2013.pdf

[10] (????) Opennebula. http://opennebula.org/

[11] (????) www.kernel.org/doc/Documentation/

[12] Abdul-Rahman O, Aida K (2013) Multi-layered architecture for the management of virtualized application environments within inter-cloud platforms. In: Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on, vol 2, pp 238–243, DOI 10.1109/CloudCom.2013.138

[13] Abts D, Marty MR, Wells PM, Klausler P, Liu H (2010) Energy proportional datacenter networks. SIGARCH Comput Archit News 38(3):338–347, DOI 10.1145/1816038.1816004

[14] Ahmad F, Vijaykumar TN (2010) Joint optimization of idle and cooling power in data centers while maintaining response time. In: Proceedings of the Fifteenth Edition of ASPLOS on Architectural Support for Programming Languages and Operating Systems, ACM, New York, NY, USA, ASPLOS XV, pp 243–256, DOI 10.1145/1736020.1736048, URL http://doi.acm.org/10.1145/1736020.1736048

[15] Al-Fares M, Loukissas A, Vahdat A (2008) A scalable, commodity data center network architecture. In: Proc. SIGCOMM 2008 Conference on Data Communications, Seattle, WA, pp 63–74

[16] Alboaneen DA, Pranggono B, Tianfield H (2014) Energy-aware virtual machine consolidation for cloud data centers. In: Utility and Cloud Computing (UCC), 2014 IEEE/ACM 7th International Conference on, pp 1010–1015, DOI 10.1109/UCC.2014.166

[17] Ali S, Siegel HJ, Maheswaran M, Hensgen D, Ali S (2000) Task execution time modeling for heterogeneous computing systems. In: Heterogeneous Computing Workshop, 2000. (HCW 2000) Proceedings. 9th, pp 185–199, DOI 10.1109/HCW.2000.843743

[18] Anghel I, Pop CB, Cioara T, Salomie I, Vartic I (2013) A swarm-inspired technique for self-organizing and consolidating data centre servers. Scalable Computing: Practice and Experience 14

[19] Antal M, Pop C, Valea D, Cioara T, Anghel I, Salomie I (2015) Optimizing data centers operation to provide ancillary services on-demand. In: Interneational Conference on Economics of Grids, Clouds, Systems and Services (GECON) 2015

[20] Arabas P, Malinowski K, Sikora A (2012) On formulation of a network energy saving optimization problem. In: Proc. of 4th Inter. Conference on Communications and Electronics (ICCE 2012), pp 122–129

[21] Arjona Aroca J, Chatzipapas A, Fernández Anta A, Mancuso V (2014) A measurement-based analysis of the energy consumption of data center servers. In: Proceedings of the 5th International Conference on Future Energy Systems, ACM, New York, NY, USA, e-Energy '14, pp 63–74, DOI 10.1145/2602044.2602061, URL http://doi.acm.org/10.1145/2602044.2602061

[22] Aschoff R, Zisman A (2011) Qos-driven proactive adaptation of service composition. In: Proceedings of the 9th International Conference on Service-Oriented Computing, Springer-Verlag, Berlin, Heidelberg, ICSOC'11, pp 421–435, DOI 10.1007/978-3-642-25535-9_28, URL http://dx.doi.org/10.1007/978-3-642-25535-9_28

[23] Åström KJ, Hägglund T (2006) Advanced PID control. ISA-The Instrumentation, Systems, and Automation Society; Research Triangle Park, NC 27709

[24] Åström KJ, Wittenmark B (2011) Computer-controlled systems: theory and design. Dover Publications, Mineola, NY

[25] Barbera MV, Kosta S, Mei A, Stefa J (2013) To offload or not to offload? the bandwidth and energy costs of mobile cloud computing. In: INFOCOM, 2013 Proceedings IEEE, pp 1285–1293, DOI 10.1109/INFCOM.2013.6566921
[26] Barroso LA, HÃűlzle U (2007) The case for energy-proportional computing. IEEE computer 40(12):33âĂŞ37
[27] Barroso LA, Clidaras J, Hölzle U (2013) The datacenter as a computer: An introduction to the design of warehouse-scale machines. Morgan & Claypool Publishers
[28] Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gener Comput Syst 28(5):755–768, DOI 10.1016/j.future.2011.04.017, URL http://dx.doi.org/10.1016/j.future.2011.04.017
[29] Benito M, Vallejo E, Beivide R (2015) On the use of commodity ethernet technology in exascale hpc systems. In: Proc. IEEE 22nd International Conference on High Performance Computing (HiPC), pp 254–263, DOI 10.1109/HiPC.2015.32
[30] Bennani MN, Menasce DA (2005) Resource allocation for autonomic data centers using analytic performance models. In: Second International Conference on Autonomic Computing (ICAC'05), pp 229–240, DOI 10.1109/ICAC.2005.50
[31] Berral JL, Goiri In, Nou R, Julià F, Guitart J, Gavaldà R, Torres J (2010) Towards energy-aware scheduling in data centers using machine learning. In: Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, ACM, New York, NY, USA, e-Energy '10, pp 215–224, DOI 10.1145/1791314.1791349, URL http://doi.acm.org/10.1145/1791314.1791349
[32] Bertsekas DP (2005) Dynamic Programming and Optimal Control, 3rd edn. Athena Scientific, Belmont, MA
[33] Bianco F, Cucchietti G, Griffa G (2007) Energy consumption trends in the next generation access network – a telco perspective. In: Proc. 29th Inter. Telecommunication Energy Conf. (INTELEC 2007), pp 737–742
[34] Bianzino AP, Chiaraviglio L, Mellia M (2011) GRiDA: a green distributed algorithm for backbone networks. In: Online Conference on Green Communications (GreenCom 2011), IEEE, pp 113–119
[35] Bobroff N, Kochut A, Beaty K (2007) Dynamic placement of virtual machines for managing sla violations. In: 2007 10th IFIP/IEEE International Symposium on Integrated Network Management, pp 119–128, DOI 10.1109/INM.2007.374776
[36] Bolla R, *et al* (2011) Econet deliverable d2.1 end-user requirements, technology, specifications and benchmarking methodology. https://www.econet-project.eu/Repository/DownloadFile/291
[37] Bolla R, *et al* (2011) Econet deliverable d4.1 definition of energy-aware states. https://www.econet-project.eu/Repository/Document/331
[38] Bolla R, Bruschi R, Ranieri A (2009) Green support for pc-based software router: performance evaluation and modeling. In: ICC'09 Communications International Conference, IEEE, pp 1–6
[39] Bolla R, Bruschi R, Davoli F, Lago P, Bakay A, Grosso R, Kamola M, Karpowicz M, Koch L, Levi D, Parladori P, Suino D (2016) Large-scale validation and benchmarking of a network of power-conservative systems using etsi's green abstraction layer. Trans Emerging Tel Tech 2016 27:451–468, DOI 10.1002/ett.3006
[40] Borgetto D, Costa GD, Pierson JM, Sayah A (2009) Energy-aware resource allocation. In: 2009 10th IEEE/ACM International Conference on Grid Computing, pp 183–188, DOI 10.1109/GRID.2009.5353063
[41] Borgetto D, Stolf P, Costa GD, Pierson JM (2010) Energy aware autonomic manager. https://www.irit.fr/publis/ASTRE/eEnergy-poster-borgetto.pdf

[42] Brasol SM (2009) Analysis of advantages and disadvantages to server virtualization. Master's thesis, âĂć

[43] Chabarek J, Sommers J, Barford P, Estan C, Tsiang D, Wright S (2008) Power awerness in network design and routing. In: Proc. 27th Conference on Computer Communications (INFOCOM 2008), pp 457–465

[44] Chen G, Kang BT, Kandemir M, Vijaykrishnan N, Irwin MJ, Chandramouli R (2004) Studying energy trade offs in offloading computation/compilation in java-enabled mobile devices. IEEE Transactions on Parallel and Distributed Systems 15(9):795–809, DOI 10.1109/TPDS.2004.47

[45] Chen Y, Gmach D, Hyser C, Wang Z, Bash C, Hoover C, Singhal S (2010) Integrated management of application performance, power and cooling in data centers. In: 2010 IEEE Network Operations and Management Symposium - NOMS 2010, pp 615–622, DOI 10.1109/NOMS.2010.5488433

[46] Chiaraviglio L, Mellia M, Neri F (2009) Energy-aware backbone networks: a case study. In: Proc. 1st International Workshop on Green Communications, IEEE International Conference on Communications (ICC'09), IEEE, pp 1–5

[47] Chiaraviglio L, Mellia M, Neri F (2011) Minimizing ISP network energy cost: formulation and solutions. IEEE/ACM Transactions on Networking 20:463 – 476

[48] Chun BG, Maniatis P (2010) Dynamically partitioning applications between weak devices and clouds. In: Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, ACM, New York, NY, USA, MCS '10, pp 7:1–7:5, DOI 10.1145/1810931.1810938, URL http://doi.acm.org/10.1145/1810931.1810938

[49] Chun BG, Ihm S, Maniatis P, Naik M, Patti A (2011) Clonecloud: Elastic execution between mobile device and cloud. In: Proceedings of the Sixth Conference on Computer Systems, ACM, New York, NY, USA, EuroSys '11, pp 301–314, DOI 10.1145/1966445.1966473, URL http://doi.acm.org/10.1145/1966445.1966473

[50] Cianfrani A, Eramo V, Listani M, Marazza M, Vittorini E (2010) An Energy Saving Routing Algorithm for a Green OSPF Protocol. In: Proc. IEEE INFOCOM Conference on Computer Communications, IEEE, pp 1 – 5

[51] Cioara T, Anghel I, Salomie I (2011) Context aware and reinforcement learning based load balancing system for green clouds

[52] Cioara T, Anghel I, Antal M, Crisan S, Salomie I (2015) Data center optimization methodology to maximize the usage of locally produced renewable energy. In: Sustainable Internet and ICT for Sustainability (SustainIT), 2015, pp 1–8, DOI 10.1109/SustainIT.2015.7101363

[53] Cisco Systems, Inc (2011) Cisco Data Center Infrastructure 2.5 Design Guide

[54] Cuervo E, Balasubramanian A, Cho Dk, Wolman A, Saroiu S, Chandra R, Bahl P (2010) Maui: Making smartphones last longer with code offload. In: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, ACM, New York, NY, USA, MobiSys '10, pp 49–62, DOI 10.1145/1814433.1814441, URL http://doi.acm.org/10.1145/1814433.1814441

[55] Cuomo F, Abbagnale A, Cianfrani A, Polverini M (2011) Keeping the connectivity and saving the energy in the Internet. In: Proc. IEEE INFOCOM 2011 Workshop on Green Communications and Networking, IEEE, pp 319–324

[56] Delaluz V, Kandemir M, Vijaykrishnan N, Sivasubramaniam A, Irwin MJ (2001) Hardware and software techniques for controlling dram power modes. IEEE Trans Comput 50(11):1154–1173, DOI 10.1109/12.966492, URL http://dx.doi.org/10.1109/12.966492

[57] Dinh HT, Lee C, Niyato D, Wang P (2013) A survey of mobile cloud computing: architecture, applications, and approaches. Wireless Communications and Mobile Computing 13(18):1587–1611, DOI 10.1002/wcm.1203, URL http://dx.doi.org/10.1002/wcm.1203

[58] Farahnakian F, Pahikkala T, Liljeberg P, Plosila J (2013) Energy aware consolidation algorithm based on k-nearest neighbor regression for cloud data centers. In: Utility and Cloud Computing (UCC), 2013 IEEE/ACM 6th International Conference on, pp 256–259, DOI 10.1109/UCC.2013.51

[59] Farahnakian F, Ashraf A, Liljeberg P, Pahikkala T, Plosila J, Porres I, Tenhunen H (2014) Energy-aware dynamic vm consolidation in cloud data centers using ant colony system. In: 2014 IEEE 7th International Conference on Cloud Computing, pp 104–111, DOI 10.1109/CLOUD.2014.24

[60] Farahnakian F, Liljeberg P, Pahikkala T, Plosila J, Tenhunen H (2014) Hierarchical vm management architecture for cloud data centers. In: Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on, pp 306–311, DOI 10.1109/CloudCom.2014.136

[61] Farahnakian F, Liljeberg P, Plosila J (2014) Energy-efficient virtual machines consolidation in cloud data centers using reinforcement learning. In: 2014 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, pp 500–507, DOI 10.1109/PDP.2014.109

[62] Feller E, Rilling L, Morin C (2011) Energy-aware ant colony based workload placement in clouds. In: 2011 IEEE/ACM 12th International Conference on Grid Computing, pp 26–33, DOI 10.1109/Grid.2011.13

[63] Fernando N, Loke SW, Rahayu W (2011) Dynamic mobile cloud computing: Ad hoc and opportunistic job sharing. In: Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on, pp 281–286, DOI 10.1109/UCC.2011.45

[64] Fisher W, Suchara M, Rexford J (2010) Greening backbone networks: reducing energy consumption by shutting off cables in bundled links. In: Proc. 1st ACM SIGCOMM workshop on Green networking (Green Networking'10), ACM, pp 29–34

[65] Floyd M, Allen-Ware M, Buyuktosunoglu A, Rajamani K, Brock B, Lefurgy C, Drake AJ, Pesantez L, Gloekler T, Tierno JA, et al (2011) Introducing the adaptive energy management features of the Power7 chip. IEEE Micro (2):60–75

[66] Franklin GF, Powell JD, Workman ML (1998) Digital control of dynamic systems, vol 3. Addison-wesley Menlo Park

[67] Gandhi A, Harchol-Balter M, Das R, Lefurgy C (2009) Optimal power allocation in server farms. In: ACM SIGMETRICS Performance Evaluation Review, ACM, vol 37, pp 157–168

[68] Gao J (2014) Machine learning applications for data center optimization

[69] Goiri In, Katsak W, Le K, Nguyen TD, Bianchini R (2013) Parasol and greenswitch: Managing datacenters powered by renewable energy. SIGARCH Comput Archit News 41(1):51–64, DOI 10.1145/2490301.2451123, URL http://doi.acm.org/10.1145/2490301.2451123

[70] Gouda KC, Radhika TV (2013) Priority based resource allocation model for cloud computing. International Journal of Science, Engineering and Technology Research

[71] Hays R (2008) Active/Idle Toggling with Low-Power Idle. presentation at IEEE802.3az Task Force Group Meeting, http://www.ieee802.org/3/az/public/jan08/hays_01_0108.pdf

[72] Hieu NT, Francesco MD, YlÃď-JÃďÃďski A (2015) Virtual machine consolidation with usage prediction for energy-efficient cloud data centers. In: 2015 IEEE 8th International Conference on Cloud Computing, pp 750–757, DOI 10.1109/CLOUD.2015.104

[73] Howard J, Dighe S, Vangal SR, Ruhl G, Borkar N, Jain S, Erraguntla V, Konow M, Riepen M, Gries M, et al (2011) A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling. IEEE Journal of Solid-State Circuits 46(1):173–183

[74] Hoyer M, Schröder K, Schlitt D, Nebel W (2011) Proactive dynamic resource management in virtualized data centers. In: Proceedings of the 2Nd International Conference on Energy-Efficient Computing and Networking, ACM, New York, NY, USA, e-Energy '11, pp 11–20, DOI 10.1145/2318716.2318719, URL http://doi.acm.org/10.1145/2318716.2318719

[75] Hsu CH, Kremer U (2003) The design, implementation, and evaluation of a compiler algorithm for cpu energy reduction. In: Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation, ACM, New York, NY, USA, PLDI '03, pp 38–48, DOI 10.1145/781131.781137, URL http://doi.acm.org/10.1145/781131.781137

[76] Hunt GC, Scott ML (1999) The coign automatic distributed partitioning system. In: Proceedings of the Third Symposium on Operating Systems Design and Implementation, USENIX Association, Berkeley, CA, USA, OSDI '99, pp 187–200, URL http://dl.acm.org/citation.cfm?id=296806.296826

[77] Idzikowski F, Orlowski S, Raack C, Rasner H, Wolisz A (2010) Saving energy in IP-over-WDM networks by switching off line cards in low-demand scenarios. In: Proc. 14th Conference on optical network design and modeling (ONDM'10), IEEE

[78] IEEE (2012) Institute of Electrical and Electronics Engineers, IEEE 802.3az Energy Efficient Ethernet Task Force. http://grouper.ieee.org/groups/802/3/az/public/index.html

[79] Jaskóła P, Malinowski K (2004) Two methods of optimal bandwidth allocation in TCP/IP networks with QoS differentiation. In: Proc. Summer Simulation Multiconference (SPECTS'04), pp 373–378

[80] Jaskóła P, Arabas P, Karbowski A (2014) Combined calculation of optimal routing and bandwidth allocation in energy aware networks. In: Proc. 26th International Teletraffic Congress, Karlskrona, IEEE, pp 1–6

[81] Jaskóła P, Arabas P, Karbowski A (2016) Simultaneous routing and flow rate optimization in energy-aware computer networks. International Journal of Applied Mathematics and Computer Science 26(1):231–243, DOI {10.1515/amcs-2016-0016}

[82] Jerger NE, Vantrease D, Lipasti M (2007) An evaluation of server consolidation workloads for multi-core designs. In: 2007 IEEE 10th International Symposium on Workload Characterization, pp 47–56, DOI 10.1109/IISWC.2007.4362180

[83] Jeyarani R, Nagaveni N, Ram RV (2012) Design and implementation of adaptive power-aware virtual machine provisioner (apa-vmp) using swarm intelligence. Future Generation Computer Systems 28(5):811 – 821, DOI http://dx.doi.org/10.1016/j.future.2011.06.002, URL http://www.sciencedirect.com/science/article/pii/S0167739X11001130, special Section: Energy efficiency in large-scale distributed systems

[84] Jonathan S, Seshadri J, Chandrasekhar A (2005) A comprehensive architectural framework for task management in scalable computational grids. In: Annual National level Technical Symposium, pp 14–15

[85] Jung H, Pedram M (2010) Supervised learning based power management for multicore processors. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 29(9):1395–1408

[86] Kambadur M, Kim MA (2014) An experimental survey of energy management across the stack. In: Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications, ACM, pp 329–344

[87] Kamola M, Arabas P (2013) Shortest path green routing and the importance of traffic matrix knowledge. In: Digital Communications - Green ICT (TIWDC), 2013 24th Tyrrhenian International Workshop on, pp 1–6, DOI 10.1109/TIWDC.2013.6664215

[88] Kaplan JM, Forrest W, Kindler N (2008) Revolutionizing data center energy efficiency. Tech. rep., McKinsey and Company, http://www.ecobaun.com/images/Revolutionizing_Data_Center_Efficiency.pdf

[89] Karbowski A, Jaskóła P (2015) Two approaches to dynamic power management in energy-aware computer networks - methodological considerations. In: Proc. of Federated Conference on Computer Science and Information Systems (FedCSIS), pp 1177–1182, DOI 10.15439/2015F228

[90] Karpowicz MP (2016) Energy-efficient CPU frequency control for the Linux system. Concurrency and Computation: Practice and Experience 28(2):420–437, DOI 10.1002/cpe.3476, URL http://dx.doi.org/10.1002/cpe.3476, cpe.3476

[91] Khargharia B (2008) Adaptive power and performance management of computing systems. PhD thesis, The University of Arizona, http://arizona.openrepository.com/arizona/bitstream/10150/193653/1/azu_etd_2708_sip1_m.pdf

[92] Kondo M, Sasaki H, Nakamura H (2007) Improving fairness, throughput and energy-efficiency on a chip multiprocessor through DVFS. ACM SIGARCH Computer Architecture News 35(1):31–38

[93] Kosta S, Aucinas A, Hui P, Mortier R, Zhang X (2012) Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In: INFOCOM, 2012 Proceedings IEEE, pp 945–953, DOI 10.1109/INFCOM.2012.6195845

[94] Kumar K, Lu YH (2010) Cloud computing for mobile users: Can offloading computation save energy? Computer 43(4):51–56, DOI 10.1109/MC.2010.98

[95] Kusic D, Kephart JO, Hanson JE, Kandasamy N, Jiang G (2008) Power and performance management of virtualized computing environments via lookahead control. In: Autonomic Computing, 2008. ICAC '08. International Conference on, pp 3–12, DOI 10.1109/ICAC.2008.31

[96] Lefurgy C, Rajamani K, Rawson F, Felter W, Kistler M, Keller TW (2003) Energy management for commercial servers. Computer 36(12):39–48, DOI 10.1109/MC.2003.1250880

[97] Lefurgy C, Wang X, Ware M (2007) Server-level power control. In: The 4th IEEE International Conference on Autonomic Computing, IEEE, DOI 10.1109/ICAC.2007.35

[98] Lefurgy C, Wang X, Ware M (2007) Server-level power control. In: Fourth International Conference on Autonomic Computing (ICAC'07), pp 4–4, DOI 10.1109/ICAC.2007.35

[99] Li L, Zheng W, Wang X, Wang X (2014) Coordinating liquid and free air cooling with workload allocation for data center power minimization. In: 11th International Conference on Autonomic Computing (ICAC 14), USENIX Association, Philadelphia, PA, pp 249–259, URL https://www.usenix.org/conference/icac14/technical-sessions/presentation/li_li

[100] Li Z, Wang C, Xu R (2001) Computation offloading to save energy on handheld devices: A partition scheme. In: Proceedings of the 2001 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, ACM, New York, NY, USA, CASES '01, pp 238–246, DOI 10.1145/502217.502257, URL http://doi.acm.org/10.1145/502217.502257

[101] Lim H, Kansal A, Liu J (2011) Power budgeting for virtualized data centers. In: 2011 USENIX Annual Technical Conference (USENIX ATC'11), p 59

[102] Linh TD, Chung NT (2015) Protected elastic-tree topology for data center. In: Proceedings of the Sixth International Symposium on Information and Communication Technology, ACM, New York, NY, USA, SoICT 2015, pp 129–134, DOI 10.1145/2833258.2833275, URL http://doi.acm.org/10.1145/2833258.2833275

[103] Liu Z, Chen Y, Bash C, Wierman A, Gmach D, Wang Z, Marwah M, Hyser C (2012) Renewable and cooling aware workload management for sustainable data centers. In: Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems, ACM, New York, NY, USA, SIGMETRICS '12, pp 175–186, DOI 10.1145/2254756.2254779, URL http://doi.acm.org/10.1145/2254756.2254779

[104] Lu Z, Hein J, Humphrey M, Stan M, Lach J, Skadron K (2002) Control-theoretic dynamic frequency and voltage scaling for multimedia workloads. In: Proceedings of the 2002 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, ACM, pp 156–163

[105] Mahmud ASMH, Ren S (2013) Online resource management for data center with energy capping. In: Presented as part of the 8th International Workshop on Feedback Computing, USENIX, San Jose, CA, URL https://www.usenix.org/conference/feedbackcomputing13/workshop-program/presentation/Mahmud

[106] Malinowski K, Niewiadomska-Szynkiewicz E, Jaskóła P (2010) Price Method and Network Congestion Control. Journal of Telecommunications and Information Technology 2:73–77

[107] Mandal U, Habib MF, Zhang S, Mukherjee B, Tornatore M (2013) Greening the cloud using renewable-energy-aware service migration. IEEE Network 27(6):36–43, DOI 10.1109/MNET.2013.6678925

[108] Mangan T (2011) Perceived performance. http://www.thincomputing.net/2011/05/23/perceived-performance-whitepaper-updated/

[109] Messer A, Greenberg I, Bernadat P, Milojicic D, Chen D, Giuli TJ, Gu X (2002) Towards a distributed platform for resource-constrained devices. In: Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on, pp 43–51, DOI 10.1109/ICDCS.2002.1022241

[110] Moore J, Chase J, Ranganathan P, Sharma R (2005) Making scheduling "cool": Temperature-aware workload placement in data centers. In: Proceedings of the Annual Conference on USENIX Annual Technical Conference, USENIX Association, Berkeley, CA, USA, ATEC '05, pp 5–5, URL http://dl.acm.org/citation.cfm?id=1247360.1247365

[111] Nakai M, Akui S, Seno K, Meguro T, Seki T, Kondo T, Hashiguchi A, Kawahara H, Kumano K, Shimura M (2005) Dynamic voltage and frequency management for a low-power embedded microprocessor. IEEE Journal of Solid-State Circuits 40(1):28–35

[112] Namboodiri V, Ghose T (2012) To cloud or not to cloud: A mobile device perspective on energy consumption of applications. In: World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a, pp 1–9, DOI 10.1109/WoWMoM.2012.6263712

[113] Naveh A, Rajwan D, Ananthakrishnan A, Weissmann E (2011) Power management architecture of the 2nd generation Intel Core microarchitecture, formerly codenamed Sandy Bridge. In: Hot Chips, vol 23, p 0

[114] Niewiadomska-Szynkiewicz E, Sikora A, Arabas P, Kołodziej J (2012) Control Framework for High Performance Energy Aware Backbone Network. In: Proc. of European Conference on Modelling and Simulation (ECMS 2012), pp 490–496

[115] Niewiadomska-Szynkiewicz E, Sikora A, Arabas P, Kołodziej J (2013) Control system for reducing energy consumption in backbone computer network. Concurrency and Computation: Practice and Experience 25:1738–1754

[116] Niewiadomska-Szynkiewicz E, Sikora A, Arabas P, Kamola M, Mincer M, KoÅĆodziej J (2014) Dynamic power management in energy-aware computer networks and data intensive systems. Future Generation Computer Systems 37:284–296

[117] Pallipadi V, Starikovskiy A (2006) The ondemand governor. In: Proceedings of the Linux Symposium, vol 2, pp 215–230

[118] Pallipadi V, Li S, Belay A (2007) cpuidle: Do nothing, efficiently. In: Proceedings of the Linux Symposium, vol 2, pp 119–125

[119] Pióro M, Mysłek M, Juttner A, Harmatos J, Szentesi A (2001) Topological Design of MPLS Networks. In: Proc. GLOBECOM'2001

[120] Prabha VL, Monie EC (2007) Hardware architecture of reinforcement learning scheme for dynamic power management in embedded systems. EURASIP J Embedded Syst 2007(1):1–1, DOI 10.1155/2007/65478, URL http://dx.doi.org/10.1155/2007/65478

[121] Prabu A, Sairamprabhu S (2014) Dynamic resource allocation using nephele framework in cloud. In: International Journal of Innovative Research in Computer and Communication Engineering

[122] Ravindran K, L MC (1978) Optimization and Operations Research: Proceedings of a Workshop Held at the University of Bonn, October 2–8, 1977, Springer Berlin Heidelberg, Berlin, Heidelberg, chap On the Computational Complexity of Integer Programming Problems, pp 161–172. DOI 10.1007/978-3-642-95322-4_17, URL http://dx.doi.org/10.1007/978-3-642-95322-4_17

[123] Raymond C, Sasitharan B, Dmitri B, William D (2012) Bio-Inspired Models of Network, Information, and Computing Systems: 5th International ICST Conference, BIONETICS 2010, Boston, USA, December 1-3, 2010, Revised Selected Papers, Springer Berlin Heidelberg, Berlin, Heidelberg, chap Application of Genetic Algorithm to Maximise Clean Energy Usage for Data Centres, pp 565–580. DOI 10.1007/978-3-642-32615-8_56, URL http://dx.doi.org/10.1007/978-3-642-32615-8_56

[124] Restrepo J, Gruber C, Machuca C (2009) Energy Profile Aware Routing. In: Proc. 1st International Workshop on Green Communications, IEEE International Conference on Communications (ICC'09), pp 1–5

[125] Roy SN (2008) Energy logic: a road map to reducing energy consumption in telecom munications networks. In: Proc. 30th International Telecommunication Energy Conference (INTELEC 2008)

[126] Rudenko A, Reiher P, Popek GJ, Kuenning GH (1998) Saving portable computer battery power through remote process execution. SIGMOBILE Mob Comput Commun Rev 2(1):19–26, DOI 10.1145/584007.584008, URL http://doi.acm.org/10.1145/584007.584008

[127] Sharifi M, Salimi H, Najafzadeh M (2012) Power-efficient distributed scheduling of virtual machines using workload-aware consolidation techniques. J Supercomput 61(1):46–66, DOI 10.1007/s11227-011-0658-5, URL http://dx.doi.org/10.1007/s11227-011-0658-5

[128] Spiliopoulos V, Kaxiras S, Keramidas G (2011) Green governors: A framework for continuously adaptive DVFS. In: 2011 International Green Computing Conference and Workshops (IGCC), IEEE, pp 1–8

[129] Srikantaiah S, Kansal A, Zhao F (2008) Energy aware consolidation for cloud computing. In: Proceedings of the 2008 Conference on Power Aware Computing and Systems, USENIX Association, Berkeley, CA, USA, HotPower'08, pp 10–10, URL http://dl.acm.org/citation.cfm?id=1855610.1855620

[130] Subramaniam B, Feng Wc (2013) Towards energy-proportional computing for enterprise-class server workloads. In: Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering, ACM, pp 15–26

[131] Torres J, Carrera D, Beltran V, Poggi N, Hogan K, Berral JL, GavaldÃă R, AyguadÃľ E, Moreno T, Guitart J (2008) Tailoring resources: The energy efficient consolidation strategy goes beyond virtualization. In: Autonomic Computing, 2008. ICAC '08. International Conference on, pp 197–198, DOI 10.1109/ICAC.2008.11

[132] Uddin M, Rahman AA (2010) Server consolidation: An approach to make data centers energy efficient and green. CoRR abs/1010.5037, URL http://arxiv.org/abs/1010.5037

[133] Vasić N, Kostić D (2010) Energy-aware traffic engineering. In: Proc. 1st Inter. Conference on Energy-Efficient Computing and Networking (E-ENERGY 2010)

[134] Verma A, Dasgupta G, Nayak TK, De P, Kothari R (2009) Server workload analysis for power minimization using consolidation. In: Proceedings of the 2009 Conference on USENIX Annual Technical Conference, USENIX Association, Berkeley, CA, USA, USENIX'09, pp 28–28, URL http://dl.acm.org/citation.cfm?id=1855807.1855835

[135] Wang C, Li Z (2004) A computation offloading scheme on handheld devices. J Parallel Distrib Comput 64(6):740–746, DOI 10.1016/j.jpdc.2003.10.005, URL http://dx.doi.org/10.1016/j.jpdc.2003.10.005

[136] Wang X, Wang Y (2011) Coordinating power control and performance management for virtualized server clusters. IEEE Transactions on Parallel and Distributed Systems 22(2):245–259

[137] Wang Y, Wang X, Chen M, Zhu X (2011) Partic: Power-aware response time control for virtualized web servers. IEEE Transactions on Parallel and Distributed Systems 22(2):323–336

[138] Wolf C (????) The myths of virtual machine consolidation. http://searchservervirtualization.techtarget.com/news/1226492/The-myths-of-virtual-machine-consolidation

[139] Wu B, Li P (2012) Load-aware stochastic feedback control for DVFS with tight performance guarantee. In: 2012 IEEE/IFIP 20th International Conference on VLSI and System-on-Chip (VLSI-SoC), pp 231–236

[140] Xu J, Fortes JAB (2010) Multi-objective virtual machine placement in virtualized data center environments. In: Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on Int'l Conference on Cyber, Physical and Social Computing (CPSCom), pp 179–188, DOI 10.1109/GreenCom-CPSCom.2010.137

[141] Yanwei Z, Yefu W, Xiaorui W (2011) Middleware 2011: ACM/IFIP/USENIX 12th International Middleware Conference, Lisbon, Portugal, December 12-16, 2011. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, chap GreenWare: Greening Cloud-Scale Data Centers to Maximize the Use of Renewable Energy, pp 143–164. DOI 10.1007/978-3-642-25821-3_8, URL http://dx.doi.org/10.1007/978-3-642-25821-3_8

[142] Ye K, Jiang X, Huang D, Chen J, Wang B (2011) Live migration of multiple virtual machines with resource reservation in cloud computing environments. In: Cloud Computing (CLOUD), 2011 IEEE International Conference on, pp 267–274, DOI 10.1109/CLOUD.2011.69

[143] Zhang Y, Wang Y, Wang X (2012) Testore: Exploiting thermal and energy storage to cut the electricity bill for datacenter cooling. In: 2012 8th international conference on network and service management (cnsm) and 2012 workshop on systems virtualiztion management (svm), pp 19–27

[144] Zhu Q, Zhu J, Agrawal G (2010) Power-aware consolidation of scientific workflows in virtualized environments. In: 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, pp 1–12, DOI 10.1109/SC.2010.43

Chapter 8
Big Data security

Agnieszka Jakóbik

8.1 Main concepts and definitions

The Big Data Systems (BD) are defined by three features: huge volume of data gathered and preceded, promptness velocity of data flow, and large variety of data itself. Petabyte-scale data gathered by such systems are processed using dedicated techniques, [1]. Sentiment Analysis, Web Server Log Analysis, data warehouse service, real-time big data stream, Enterprise Data Hubs are offered as a service. The data in BD systems are treated differently in comparison to the traditional systems. The splitting and conjunction of the data is important for data updating and processing because data are dispersed widely. It was stated also in [2] that that 80% of the effort involved in processing data is cleaning and changing the incoming stream of data into applicative form, .

Big data projects rarely involve single organization and a lot of communication between sides is incorporated in the process. Moreover, a lot of analysis in the fields of engineering, computer science, physics, economics and life sciences is made. This analysis is performed using machine learning and automatic methods,These approaches are beyond the direct control of human, [3]. During all this stages: reliance, integrity, confidentiality, genuineness and availability, have to be monitored and provided nearly in the real time and on he massive scale.

The systems and socialmediabasedon Big Data paradigm such as like LinkedIn, Netflix, Facebook, Twitter, Expedia, national and local politics, big sales companies and a lot of other organizations are generating enormous economic, social, and political impact into modern society. The decision made based on Big Data analytic has large consequences in the real world, [4].

Therefore is very important to assure the save acquisition, storage, and usage of such data, especially when the facts about peopleâĂŹs attributes, behavior, preferences, relationships, and locations are being gathers, processes and used in favor of many subjects.

The problem of assuring security in BD systems is very complicated and not very well recognized. The rapid growth of popularity of such systems imposes the necessity to formulation and formal description of this problem.

Agnieszka Jakóbik
Institute of Computer Science, Cracow University of Technology, Poland, e-mail: agneskrok@gmail.com

8.2 Big Data security

ISO/IEC 27002 norm introduces definition of *information security* as the the protection of information from a wide range of threats in order to ensure business continuity, minimize business risk, and maximize return on investments and business opportunities. Information security is implemented in the form of policies, processes, procedures, organizational structures and software and hardware functions. These components need to be established, implemented, monitored, reviewed and improved constantly. *Security of computer systems* are specified as the following C-I-A model.

The preservation of **confidentiality** - information should be accessible only to those authorized to have access, **integrity** - information should be accurate and complete and **availability** - ensuring that authorized users have access to information when required), citenorm *'Security' of the system user* may be considered deferentially as far as the role of user is concerned :

- security from data uploaders point of view, ex. the right to privacy,
- security from data users point of view, ex. the right for not being deceived by having wrongly uploaded or manipulated data,
- security from society point of view, ex. not being abused but those who obtained information from BD sources.

All fields specified in the ISO/IEC 27002:2013 standard for information security published by the International Organization for Standardization (ISO) have to be fulfilled in BD systems.

1. Organization of Information Security - responsibilities for information security should be assigned to individual persons and duties should be allocated across roles and to avoid conflicts of interest and prevent inappropriate information leaks.
2. Human Resource Security - security responsibilities have to be monitored during recruiting employees, workers and temporary staff.
3. Asset Management - data should be segregated according to the security protection that is necessary, and security policies should be diversified appropriately.
4. Access Control - data access should be restricted and control over roles and privileges.
5. Cryptography - cryptography protocols should be constantly monitored due to their validity.
6. Physical and environmental security - physical barriers, with physical entry controls and working procedures, should protect staff offices, data centers storage places, delivery/loading areas against unauthorized access.
7. Operation Security - protection against harmful software, backup done regularly, logging and monitoring as far as users and administrators activities, incidents, faults and security violation, synchronizing clokcs, operational software life cycle monitoring, operational systems updating.
8. Communication security - all networks and network services should be properly secured.
9. System acquisition, development and maintenance - all software installed should be monitored, the development environment should be protected, and external resources should be controlled.
10. Information security incident management - incidents should be reported, responded as soon as possible to and learn from
11. Compliance with legal and contractual requirements should be enforced.

Aforementioned security fields have to be considered together with treats for security resulting from with three feathers of Big Data systems itself:

- Variety: changing traditional relational database into non-relational databases enforces different kind of methods as far as the need to disable reading the data by unauthorized persons, storing data in the ciphered form, and disable the identity detection from anonymized data sets by correlating with public databases.

- Volume: The volume of Big Data enforces the storage in multitiered storage media. That cases additional problem of securing data integrity and inviolability when data are segmented, moving between tiers and merged.
- Velocity: The velocity of data collection enforces usage of such a security methods, algorithms and hardware that can proceed fast enough not to disrupt the flow of data.
- Variability: Security and privacy requirements have to be ensured also when the data are moved to another vendor or are no longer valid and shroud be erased. [6]

In the Big Data systems some problems with security and privacy may occur also in contrary to the traditional systems. Big Data may be collected from variety of end points. The roles that are incorporated into authentication and authorization include more types than traditional system providers, consumers, data owners. In particular, mobile users and social network users have to be taken into consideration. Data aggregation and dissemination must be secured inside the storage system, because lean clients do not have enough computing power to perform necessary numerical operations involved in data ciphering or hasing. The secure availability of data on demand by a broad group of stakeholders have to be ensure by using readily understood security frameworks, dedicated to users who do not have any knowledge in the subject. Data search and selection can also lead to privacy or security policy concerns. Legacy security solutions need to be retargeted to Big Data to be used in High Performance Computing (HPC) resources. Attention must be given particularly to systems that are collecting data from fully public domains. Methods to prevent adversarial manipulation and preserve integrity of data have to be incorporated. Its extreme scalability causes that Big Data systems are also sensitive to recovery methods and practices. Traditional backup tools are impractical. Big Data systems enforces that monitoring of prevention and protection against hacker attacks have be scaled enormously. Security and privacy may be weakened by unintentional operations made by uneducated users,[8]. Therefore educational policies for end users have to be considered and the methods for detecting accidentally made security threads.

1. Data Confidentiality
   Confidentiality of data in Big Data systems have to be assured during three stages of data processing: in transit, at rest, during processing. Each stage requires and enables different kind protocols to compromise between effectiveness and computational cost. Moreover methods of computing on encrypted data have to be used especially searching and reporting methods. Aggregating data without compromising privacy, and data anonymization methods have to be incorporated.
2. Provenance. A mechanisms to validate if input data were uploaded from authenticated trustful sources are necessary. For example digital signatures may be incorporated. Moreover validation at a syntactic level and semantic validation is obligatory.
3. Identity and Access Management Key management methods have to take into consideration greater variety of users types and have to be suitable for volume, velocity, variety, and variability of data. In Big Data systems virtualization layer identity, application layer Identity, end-user layer identity and identity of provider is necessary. Key Management life-cyle involves: generating Keys, assuring nonlinear Keyspaces, transferring Keys, verifying Keys, using Keys, updating Keys, storing Keys, backup Keys, compromised Keys and destroying Keys for all participants involved in the process and on the massive scale, [7].

8.3 Methods and solutions for security assurance BD systems

Various cryptographic methods and protocols used in socalled '**traditional cryptography**' may be transformed and successfully used alsoin BD approaches. We can include to the wide class of such methods the : symmetric Cryptography methods, One-Way Hash Functions, Public-Key Cryptography, Digital Signatures with Encryption, Authentication and Key-Exchange Cartographic Protocols, Multiple-Key Public-Key Cryptography, Secret Splitting and Secret Sharing protocols, Undeniable Digital Signatures, Designated Confirmer Signatures, Blind Signatures, Identity-Based Public-Key Cryptography methods, [9], [10].

8.3.1 Authorization and authentication methods

Authentication (determining whether someone is who they claim to be), authorization (specifying access rights to resources), dedicated to the Big Data systems requires scalability and instant high performance, moreover machines on which such a systems are processing are equipped with massively parallel software running on many commodity computers in distributed computing frameworks. For this reason cartographic protocols and schemes have to be optimized to such environment. Big Data (BD) systems are build from few components: Master System (MS) receives data from data sources, then task Distribution function is applied to distributing tasks to the workers/slaves of the system (Cooperated System - CS), after Data distribution function distributing data to the cooperated systems, after tasks are finished Result Collection function gathers information and send it to users. These functions may be installed in different hosts and all stages needs authorization and authentication methods to obtain the proper flow of the data. Security techniques for Acces Controll for each component is necessary. The authorization, is much more complicated than non-Big Data systems, because of the necessity of synchronizing access privileges between the MS and CSs. **Security Agreement** is made between data uploaders (BD source providers) and the MS. It helps to caterorize security classes of data sources. The aim SA is to make decisions about levels of security (or trust) that is required by the data uploaders (ex. different security levels for emails,and free stock photographs). **Trust Workers List** (TCSL) lists the trusted Workers and categorizes them by the security classes. **MS access Policy** is characterizing a set of access rules that are imposed by Master System into Workers. **Worker access Policy** is the list of rules of access to the distributed resources managed by the particular worker (ex.disk space, Virtual Machines).

To apply proper data flow, coordination of data up loaders with Master System by Security Agreement have to be completed, then Master System is gathering information about available Workers. After matching security needs of Master Systems with security offers by Workers the data are uploaded, task are formulated and sending the the Workers. The proposed scheme can be formalized by the following rules: âĂć A set of security classes from c_1 to c_n combined as

$$C = \{c_1, ..., c_n\} \tag{8.1}$$

âĂć A set of BD source providers from bd_1 to bd_n combined as

$$BD = \{bd_1, ..., bd_n\} \tag{8.2}$$

âĂć A set of Cooperated System from cs_1 to cs_n in the form of the set

$$CS = \{cs_1, ..., cs_n\} \tag{8.3}$$

âĂć A set of security attributes from at_1 to at_n in the set

$$AT = at_1, ..., at_n$$

âĂć A set of (bd_x, c_x) pairs in the set SA in BD xi C. âĂć A set of (cs_x, c_x) pairs in the set TCSL in CS xi C. A set of MS policy rules from mp_1 to mp_n in the set

$$MSP = \{mp_1, ..., mp_n\} \tag{8.4}$$

where

$$mpi = (at_x, a_x, bd_x)$$

is a tuple where, at_x from AT is an attributes, a_x is an action, and bd_x from BD is a source provider that means subject with attribute at_x is permitted to perform action a_x on object from bd_x âĂć A set of AC policy rules for CS CSi collected in the set

$$CSPi = \{cspi_1, ..., cspi_n\} \tag{8.5}$$

where each

$$cspii = (bd_x, a_x, rs_x) \tag{8.6}$$

is a tuple where bd_x is the element from BD, a_x is an action, and rs_x is a local resource from CSi that means subject from bd_x is permitted to perform action a_x on object rsx

Let the $BDU = (u, a_u, bd_u)$ is a user request; where u is an authenticated BD user by the MS, a_u is a requested action, and bd_u is a BD source provider of the data or service that u is request to perform a_u from. The algorithm to accept (u, a_u, bd_u) on the cs_l is, [25]:

for $C_u = \{c_1, ...c_k\}$ **such that** (bd_u, c_i) is from SA;
if $C_u =$ empty set then
$\{$ request = deny (for this cs_l)
else
$CS_u = cs_1, ...cs_k$ such that (cs_i, c_i) is from TCSL and c_i is from C_u;
if $CS_u =$ empty set or cs_l is not in the set CS_u $\{$
request = deny (for this cs_l)
else
if (there exist $(mp_x = (at_x, a_x, bd_x)$ in MSP set, such that
$a_u == a_x$ and $bd_u == bd_x$) and (there exist $(csp_x = (bd_x, a_x, rs_x)$ in set CSP_l
such that $a_u == a_x$ and $bd_u == bd_x$ and $rs_x = =$ resource required))
$\{$
request = granted for this cs_l

if perform a_u on $rs_x == $ success
return result to RC
else
return âĂIJRC resource from cs_l unavailableâĂİ
}
else
request = deny for this cs_l
}
}
}

8.3.2 Data privacy

Big Data analytics invades the personal privacy of individuals, [21]. The negative impact of the improper usage of the analysis that is made based on the Data may be:

- Discrimination. The usage of predictive analytics to make determinations about users intelligence, habits, education, health, ability obtain a job, finance. The use of users associations in predictive analytics to make decisions that have a negative impact on individuals. Such opinions are âĂIJautomated,âĂİ and therefore more difficult to detect or prove, and may influence for example employment, promotions to fair housing and many more.
- An embarrassment. Online shops and restaurants, government agencies, universities, online media corporations may be the reaseon for the personal information lickage resulting in revealing the personal information users, employees, especially very private information that people would like to keep separated from their business life (health problems, sexual orientation or an illnesses).
- The lost of anonymity. If data masking is not done effectively, analysis could easily match the individual whose data has been masked to this person.
- Government exemptions. Personally Identifiable Information (PII): name, any aliases, race, sex, date and place of birth, Social Security number, passport and driverâĂŹs license numbers, address, telephone numbers, photographs, fingerprints, financial information like bank accounts, employment and business information and more are collected by governments. The discredit of such a data bases is the great threat and might lead to the destabilization of the county, [22].

8.3.3 Technologies and strategies for Privacy Protection

Encryption algorithms, anonymization or deâĂŘidentification, deletionand non âĂŘ retention methods helps to protect privacy.

There are three basic types of cryptographic algorithms, [24],that were successfully adopted to BD systems:

1. Cryptographic hash functions. Hash function produces a short (the length is always known) representation of a any longer message. Hash function is a one-way function: it is easy to compute the

hash value from a particular input; but calculating the input from hash value is extremely computationally difficult or impossible. Hash function are also collision resistant: is also extremely difficult to find two particular inputs that produce the same hash value. Because of these feathers, hash functions used to determine whether or not data has changed and are the part of the digital signature scheme. In Big Data systems data integrity errors could appear because of hardware errors, software errors, intrusions, or user errors therefore needs to be checked after each stage of data processing. Approved safe hash functions are: from SHA-2 family: SHA-224, SHA-256, SHA-384 and SHA-512,and SHA-3.
2. Symmetric algorithms (secret key algorithms) incorporating a single key to both ciphering and to remove or check the protection (deciphering). Additional methods of safe key exchange between sender and receiver have to be used. The Approved algorithms for symmetric encryption and decryption are: the Advanced Encryption Standard (AES) and the Triple Data Encryption Algorithm (TDEA), based on the Data Encryption Standard (DES). Using symmetric key block cipher algorithm, in case of the multiple blocks in a message (data stream) are encrypted they must not be processed separately. The Recommendation for Block Cipher Modes of Operation defines modes of operation that have to be used, citemodes.
3. Asymmetric algorithms (public key algorithms) incorporated two keys (i.e., a key pair): a public key (may be put in public database) and a private key (must be kept secret) that cannot be calculated one from another. Approved safe asymmetric algorithm is RSA and The length of the key should be set properly, [23].
4. Message Authentication Codes (MACs) provide authenticity and integrity. A MAC is a cryptographic checksum on the data that is used to provide assurance that the data has not changed or been altered and that the MAC was computed by the expected sender. NIST SP 800-38B, Recommendation for Block Cipher Modes of Operation: the CMAC Authentication Mode, defines ways to compute a MAC using approved block cipher algorithms. FIPS 198, The Keyed Hash Message Authentication Code (HMAC), defines a MAC that incorporates a cryptographic hash function in together with a secret key. HMAC shall be used with an Approved cryptographic hash function.
5. Digital Signatures and the Digital Signature Standard (DSS). A digital signature is an electronic analogue of a hand written signature and it is used proving to the recipient or a third party that the message was signed by the originator. Signature generation process incorporates a private key to generate a signature. Signature verification process uses the public key that matches to public key, to verify the signature. Hash functions are used to exclude manipulations with the send data. Digital Signature Standard (DSS) includes three digital signature algorithms: the Digital Signature Algorithm (DSA), the Elliptic Curve Digital Signature Algorithm (ECDSA) and RSA. The DSS is used with Secure Hash Standard, [25].
6. Public Key Infrastructure (PKI) regulates generating and distributing methods for public key certificates and ways of maintaining and distributing certificate status information for unexpired certificates. PKI defines components:

   - certification authorities to create certificates and certificate status information,
   - registration authorities to verify the information in the public key certificates and determine certificate status,
   - authorized repositories to distribute certificates and certificate revocation lists
   - online Certificate Status Protocol servers to distribute certificate status information,
   - key recovery services to backup private keys,
   - credential servers to distribute private key material and the corresponding certificates.

The example of certificates used in PKI may be the X.509 Certificate , [27].

Also many cryptography solutions were proposed that are dedicated to the BD systems: quantum cryptography and privacy with authentication for mobile data center, [16], group key transfer based on secret sharing over big data, [17], an ID-based generalized signcryption method to obtain confidentiality or/and authenticity, [19], capability based authorization,[20].

8.3.4 Quantum cryptography and privacy with authentication for mobile data center

Quantum cryptography was proposed with GroverâĂŹs algorithm (GA) ,[34], and PairHand authentication protocol, [35], to asset secure communications between the mobile users and authentication servers.

Proposed model includes several layers, and supports secure big data sending by mobile user to the nearest mobile data center.

- Data center front end Layer: verifications and identifications of the mobile user and big data using Quantum cryptography and authentication protocols
- Data reading interface Layer: during each operation of the interface, provides the best performance to minimize the complexity
- Quantum key processing Layer: quantum key distribution (QKD) based on QC is taken into considerations, and the size of the big data and level of the security
- key management Layer: the size of the big data and traffic load, the security key generations is performed, protocols based on QC are applied
- Application Layer: depending on the applications that are used by data upleader, division should be made according to organization policy with different level of the security and privacy.

It was stated that designing mobile data center with the PairHand protocol reduces the computational cost and increases the efficiency of the handover authentication,

8.3.5 Group key transfer based on secret sharing over big data

A key transfer protocol for secure group communications over big data was proposed and is designed particularly for group-oriented applications over big data. Linear secret sharing schemes, are used, [37] . A secret is divided into shares and is shared among a set of shareholders by a trusted dealer in such a way that authorized subsets of shareholders can reconstruct the secret but unauthorized subsets of can not.The Vandermonde Matrix is used as the share generation algorithm, [36]. Key transfer protocol consists of two phases: the secret establishment phase and the session key transfer phase.

8.3.6 ID-based generalized signcryption method to obtain confidentiality or/and authenticity

Generalized signcryption (GSC) methods ware used to provide multi-receiver identity-based generalized signcryption (MID-GSC) method. Bilinear Diffie-Hellman (BDH) assumption and Computational Diffie-Hellman (CDH) assumption was used to ensure safety of the system. Either a single message or multiple messages can be signcrypted for one or multiple receivers and by one or multiple senders.

8.3.7 Capability based authorization methods

The capability based authorization models have many additional feathers in comparing to the traditional models, that is:

- delegation support: a user can obtain access rights to another user, and the permit ion to further delegate the rights. Moreover, the delegation depth is controllable at each level,
- capability revocation: the right to the resources may be canceled by authorized subjects,
- information granularity: dynamic adaptation of permissions and access privileges is assumed by the provider to react on changes in users needs,
- capability token is used and Role Based Access Control (RBAC) systems or The Attribute Based Access Control (ABAC) are incorporated.

These models were proposed for the systems where, at the same time, usersâĂŹ needs could be highly dynamic and limited in time therefore no permanent link between a user and an service is beneficial.

A user that has specified a service for which he needs access submits a request to the Digital Ecosystem Clients Portal. This request is passed to the Policy Decision Point (PDP). The PDP decides if the userâĂŹs request may be accepted or heave to be denied. In the first case it provides an access token (capability token) that enables access to the service. This token is given to the user. Each capability token has the following characteristics: the resource(s) the it gives the permission to use, the subject (user) to which the rights have been allowed, the granted privileges, and the authorization chain, that user is required to fulfill in order to prove his identity. The model is using Zero Knowledge Proofs, to prove, without disclosing any personal information, the user has the right to receive an access capability for a resource.

The proposed CapBAC architecture consists several components:

- the resource in the form of information service or an application service that has to be identifiable and may serve user,
- authorization capability that is the characterization of rights to be given together with the specification which ones can be delegated further, and their delegation depths, in contests of the resources on which those permissions are executed,
- capability revocation rules together with the specification of users who may revoke a single capability, a complex capability and all its descendants,
- service/operation request center that is processing requests from users and sending them to one single vendor,
- Policy Decision Point in charge of managing resource access requests,

- the Monitoring Service that checks capability tokens and digitally signs, the request to prove it is the owner of the capability. It also is doing formal validity of all capabilities in the authorization chain and logical validity of the operation request,
- the particular resource manager that proceeds users requests for the particular resource.

The access rights capability token consists of the AccessRightsCapabilityID, IssueInstant attributes, the Issuer, the Subject, the ResourceID:, the AccessRightsCapabilityRevocationService, the ValidityCondition, the IssuerAccessRightsCapability, and the AccessRights element.

8.3.8 No QSL data basis security

NoSQL databases such as CouchDB, MongoDB and Cassandra were initially not designed with security issues as an main feature. Therefore, third party tools and services have to be used. Sharding that is taking place also generates security risks, caused by to geographic distribution of data, unencrypted data storage, unauthorized exposure of backup and replicated data, insecure communication over the network. Security of NoSQL databases involves securing data-at-rest on a single node, data security during transmission between various nodes in a sharded environment that are made often between countries and inside international structures, [11]. Such methods as Consistent Hashing (Distributed Hash Tables), Flexible Partitioning and High Availability monitoring, Inter-cluster communication, Denial of Service problem governing, continuous Auditing, Potential for injection attacks monitoring methods, Intrusion Detection Systems, Kerberos Mechanism, Bull Eye Algorithm Approach supports ensuring the safety of data in such systems, [12], [13]. Action aware access control roadmap was proposed in,[18]. As far as platform selection and analysis is concerned selection of existing MapReduce systems and NoSQL datastores shloud be considered at the first place. Then identification of policy components have to be made. The next step is then definition of enforcement mechanisms for chosen BD data base:

- MongoDB with Role-based access control (RBAC) at database and collection level access control
- Cassandra RBAC at key-space and table level access control
- Redis for which access control can only be achieved at application level
- HBase incorporating control lists at column family and/or table level
- CouchDB with no native access control
- Hive equipped with fine grained access control and relational model
- Hadoop having Access control lists at resource level
- Spark incorporating Access control lists at resource level

8.3.9 Trust management

Trust according to the international standards and the international law management/monitoring have to assured with special concern about the fact that cryptography methods and protocols may become outdated. The listed below institution are publishing the updated guidance and regulation according to the Big Data security:

- National Security Agency (NSA),

- National Computer Security Center (NCSC)
- National Institute of Standards and Technology (NIST)
- RSA Data Security, Inc.
- International Association for Cryptologic Research (IACR)
- International Organization for Standardization (ISO) www.iso.ch.
- Federal Information Processing Standards (FIPS)
- Cloud Security Alliance
- Electronic Privacy Information Center (EPIC)
- Academic researchers centers (MIT Computer Science and Artificial Intelligence Laboratory, Lawrence Berkeley National Laboratory, Industry University cooperative research center for Intelligent Maintenance Systems (IMS) at university of Cincinnati).

The massiveness of Big Data systems characteristics has a great impact of privacy, security and consumer welfare, moreover it was stated that social and economic costs and potential negative externalities is very high,[14].

Therefore Big Data systems providers have to respect data protection laws.The policies may different for particular regions in with the data are collected,for example:

- Canada: The Personal Information Protection and Electronic Documents Act (PIPEDA), [29], specifies the rules to govern collection, use or disclosure of personal information, and gives users the rights to understand the reasons why organizations collect, use, or disclose personal information, to expect organizations to protect the personal information in a reasonable and secure way.
- European-Union: the 8th article of the European Convention on Human Rights (ECHR) gives the right to respect oneâĂŹs âĂIJprivate and family life, his home and his correspondenceâĂİ is provided. EU membersâĂŹ states adopted legislation pursuant Data Protection Directive, [26], adapted their existing laws.

Resolving conflicts of different security rule sets according to different laws in each country, and the territorial dissipation between data uploaders, data owners and data users caused the need for international certification of systems gathering and processing big amount of data. The examples of such ceryficates are:

- TRUSTeâĂŹs APEC Privacy Certification program, checking among the others way in which collected information is used, types of Third Parties, if any, with whom collected data is shared and for what purpose, method for updating privacy settings, types of passive collection technologies used (e.g., cookies, web beacons, device recognition technologies). A statement that collected information is subject to disclosure pursuant to judicial or other governmental subpoenas, warrants, orders, or goes bankrupt, or to protect the rights of the Participant, or protect the safety of the Individual or the safety of others is also required, [28] .
- The ISO 27000 family of standards: ISO 27001 that covers security in the cloud, ISO 27002, ISO 27018 that is the description of policy for personally identifiable information to the scope of 27001. ISO 27018 compatibility means that owner of the data will not use customer data for their own independent purposes, such as advertising and marketing, without the customer's express consent and establishes clear and transparent parameters for the return, transfer and secure disposal of personal information, [33].
- The SSAE16 Auditing Standard: If the system is protected against unauthorized access, if system is available for operation and use as committed or agreed, processing data is complete, accurate, timely, and authorized and if data designated as confidential is protected, [30].

8.3.10 Secure communication

Big Data gathering, recessing and storage require cooperation of may members of the system. Data uploaders may use different means and methods to connect to the storage servers: computers, tablets, mobile phones, lean stations equipped with web browser only.

Network layer security protocols ensures secure network communications at the layer where packets are routing across networks. Transport layer security methods describe security at the layer responsible for end-to-end communications. Transmission Control Protocol/Internet Protocol (TCP/IP) model, Secure Sockets Layer (SSL) and Transport Layer Security (TLS), [31] are recommended.

During transferring data across networks, the data is transported from the highest layer through intermediate layers to the lowest layer of the system:

1. Application Layer sends data from particular applications, such as Domain Name System (DNS), HyperText Transfer Protocol (HTTP), and Simple Mail Transfer Protocol (SMTP).
2. Transport Layer provides services for transporting data from application layer to destination networks. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are used as protocols.
3. Network Layer. This layer routes packets across networks. Internet Protocol (IP), Internet Control Message Protocol (ICMP) and Internet Group Management Protocol (IGMP) are used, [32].

8.4 Summary

Main problems concerning security in Big Data systems were presented. Security from user,data owner and data uploader point of view was considered. Chosen methods for the preservation confidentiality, integrity and availability were presented. Some of them were adopted to from traditional systems and still have to be adjusted to be fully applicative. The methods particularly designed for BD systems and may be not still tested enough to provide security. The data lifecycle in BD systems is complicated and involves a lot users and operation on data. Therefore there are many week points as fa as security is concerned. Data being sent, data at rest, data being processed and deleted from the system requires different kind of techniques to assure authenticity and provenance. The need for third parity trust centers was emphasized. The necessity for external control as far as international low is concerned was stated.

Further studies on the impact of variety, volume and velocity on data security is very important. The preservation of confidentiality, integrity and availability should be the main concern assuring the trust for BD services.

References

[1] Marz, N., Warren, J., Big Data: Principles and best practices of scalable realtime data systems, Manning Publications, (2015)

[2] Big Data Now: 2012 Edition, OâĂŹReilly Media, Inc., (2012)

[3] Liu, H., Gegov, A., Cocea, A.: Rule Based Systems for Big Data A Machine Learning Approach, Springer, ISBN 978-3-319-23696-4, (2016)

[4] Davis, K., Patterson D.: Ethics of Big Data, OâĂŹReilly Media, Inc., (2012)

[5] INTERNATIONAL STANDARD ISO/IEC 27002, Information technology âĂŤ Security techniques âĂŤ Code of practice for information security management, ISO/IEC FDIS 17799:2005(E),(2005)

[6] NIST Special Publication 1500-1, NIST Big Data Interoperability Framework: Vol. 1, Definitions, NIST Big Data Public Working Group (NBD-PWG) , doi: 10.6028/NIST.SP.1500-1

[7] NIST Special Publication 1500-4, NIST Big Data Interoperability, Security and Privacy, NIST Big Data Public Working Group, doi: 10.6028/NIST.SP.1500-4

[8] Top Ten Big Data Security and Privacy Challenges, Cloud Security Alliance, $http : //www.isaca.org/groups/professional - english/big - data/groupdocuments/big_data_top_ten_v1.pdf$, cited 22 Mar 2016

[9] van Tilborg, H. C.A., Jajodia, S. (Eds.), Encyclopedia of Cryptography and Security, Springer, ISBN 978-1-4419-5905-8

[10] Schneier,B.: Applied Cryptography Protocols, Algorithms, and Source Code in C, John Wiley and Sons, (1996)

[11] Zahid, A., Masood, R., Awais Shibli M.: Security of Sharded NoSQL Databases: A Comparative Analysis , Conference on Information Assurance and Cyber Security (CIACS), Doi: 978-1-4799-5852-8/14/, (2014)

[12] Okman,L., Gal-Oz,N., Gonen, Y., Gudes, Y. :Security Issues in NoSQL Databases, DOI: 10.1109/TrustCom.2011.70, IEEE,(2011)

[13] Pazhanirajaa,N., Victer Paula,P., Saleem Bashab M.S., Dhavachelvanc P.:Big Data and Hadoop-A Study in Security Perspective, doi:10.1016/j.procs.2015.04.091, Procedia Computer Science, Vol. 50, Big Data, Cloud and Computing Challenges, (2015)

[14] Kshetri,N.B.: Big data's impact on privacy, security and consumer welfare, Telecommunications Policy, 38(11), 1134-1145, www.elsevier.com/locate/telpol, cited 22 mar 2016

[15] Barker,E.B., Barker, W. C., Lee A.: NIST Special Publication 800-21 , Guideline for Implementing Cryptography In the Federal Government,U.S. Department of Commerce,(2005), $http : //csrc.nist.gov/publications/nistpubs/800 - 21 - 1/sp800 - 21 - 1_Dec2005.pdf$, cited 22 mar 2016

[16] Thayananthan, V., Albeshri,A. :Big data security issues based on quantum cryptography and privacy, with authentication for mobile data center, Procedia Computer Science 50 (2015) 149 âĂŞ 156, 2nd International Symposium on Big Data and Cloud Computing (ISBCCâĂŹ15),(2015), doi:10.1016/j.procs.2015.04.077

[17] Zeng,B., Zhang, M.: A novel group key transfer for big data security, Applied Mathematics and Computation 249 (2014) 436âĂŞ443, doi: 10.1016/j.amc.2014.10.051

[18] Colombo,P., Ferrari,E.: Privacy Aware Access Control for Big Data: A Research Roadmap, Big Data Research 2 (2015) 145âĂŞ154, doi: 10.1016/j.bdr.2015.08.001

[19] Wang,H., Jiang, X., Kambourakis, G.: Special issue on Security, Privacy and Trust in network-based Big Data,Information Sciences 318 (2015) 48âĂŞ50, doi:10.1016/j.ins.2015.05.040

[20] Piccione,S., Rotondi, D., A capability-based security approach to manage access control in the Internet of Things, Mathematical and Computer Modelling 58 (2013) 1189âĂŞ1205,

[21] Rotenberg M., COMMENTS OF THE ELECTRONIC PRIVACY INFORMATION CENTER, THE OFFICE OF SCIENCE AND TECHNOLOGY POLICY Request for Information: Big Data and the Future of Privacy , Electronic Privacy Information Center (EPIC), 2014, https://epic.org/privacy/big-data/EPIC-OSTP-Big-Data.pdf, cited 22 Mar 2016

[22] Armerding, T., The 5 worst Big Data privacy risks (and how to guard against them), http://www.csoonline.com/article/2855641/big-data-security/the-5-worst-big-data-privacy-risks-and-how-to-guard-against-them.html, cited 22 Mar 2016

[23] NIST Special Publication 800-57 (SP 800-57), Recommendation for Key Management, provides guidance on the management of cryptographic keys, $http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf$, cited 22 Mar 2016
[24] Stallings W.: Cryptography and Network Security: Principles and Practice, Pearson; (2013)
[25] Hu,V.C., Grance, T., Ferraio D. F., Kuhn,D., : An Access Control Scheme for Big Data Processing, National Institute of Standards and Technology, USA, $http://csrc.nist.gov/projects/ac-policy-igs/big_data_control_access_7-10-2014.pdf$, cited 22 Mar 2016
[26] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, Official Journal L 281 , 23/11/1995 P. 0031 - 0050
[27] X.509 : Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks, http://www.itu.int/rec/T-REC-X.509/en,cited 22 Mar 2016
[28] APEC Certification Standards, https://www.truste.com/privacy-certification-standards/apec/, cited 22 Mar 2016,
[29] Personal Information Protection and Electronic Documents Act, http://laws-lois.justice.gc.ca/PDF/P-8.6.pdf, Published by the Minister of Justice, Canada, (2016)
[30] The SSAE16 Auditing Standard, (2015), http://www.ssae-16.com/, cited 22 Mar 2016
[31] Guide to SSL VPNs, Special Publication 800-113, Recommendations of the National Institute of Standards and Technology,(2008), http://csrc.nist.gov/publications/nistpubs/800-113/SP800-113.pdf, cited 22 Mar 2016
[32] Kizza,J.: Computer Network Security, Springer, (2005), ISBN-10: 0387204733
[33] The International Standard for Data Protection in the Cloud, ISO/IEC 27018, https://www.iso.org/obp/ui/iso:std:iso-iec:27018:ed-1:v1:en, (2014) cited 22 Mar 2016
[34] Goorden, S. A., Horstmann M., Mosk, A. P., Åăkori, B., and Pinkse, P. W. H., Quantum-Secure Authentication Of A Physical Unclonable Key, Optica, Vol. 1, No. 6, (2014)
[35] He, D., Jiajun B., Chan,s. and Handauth,Ch.: Efficient Handover Authentication with Conditional Privacy for Wireless Networks, IEEE Transactions on Computers, VOL. 62, NO. 3, (2013)
[36] C.-F. Hsu, Q. Cheng, X. Tang, B. Zeng, An ideal multi-secret sharing scheme based on msp, Inf. Sci. 181 (7) (2011) 1403âĂŞ1409
[37] O. Farras, C. PadrÃş, Ideal hierarchical secret sharing schemes, in: Theory of Cryptography, Springer, 2010, pp. 219âĂŞ236.

Index