Delivering Social Multimedia Content on the Cloud

Irene Kilanioti[1], Elli Rapti[2], Francisco Couto[3], George Suciu[4], Luis Correia[5], Anthony Karageorgos[6], and George A. Papadopoulos[1]

[1] Department of Computer Science, University of Cyprus, 1 University Avenue, 2109 Nicosia, Cyprus
ekoila01@cs.ucy.ac.cy, george@cs.ucy.ac.cy
[2] Institute for Research and Technology Thessaly, CERTH, Volos, Greece
erapti@ireteth.certh.gr
[3] Department of Informatics, Faculty of Sciences of the University of Lisbon, Campo Grande, 1749-016 Lisbon, Portugal
fcouto@di.fc.ul.pt
[4] Beia Consult International, 12-22 Peroni Str. District 4, 041386 Bucharest, Romania
george@beia.ro
[5] Faculty of Sciences of the University of Lisbon, Campo Grande, 1749-016 Lisbon, Portuga
Luis.Correia@ciencias.ulisboa.pt
[6] Applied Informatics Laboratory, Technological Educational Institute of Thessaly, Karditsa, Greece
karageorgos@computer.org

Abstract. The improvement of user experience through scaling bandwidth-demanding content largely depends on the exploitation of usage patterns found in OSNs. Big data analysis tasks (such as cascades characterization) accommodate large volumes of data for the improvement of user experience, e.g. via prefetching in the framework of a CDN infrastructure that streaming providers own. Usage patterns information can be applied in situations where traditional bandwidth-intensive content scaling is infeasible: global replication demanded by traditional CDNs for the voluminous content produced becomes expensive and, moreover, user-generated content is especially difficult due to its long tail nature, with each item probably not popular enough to be replicated globally, but with the long-tail altogether getting sufficient accesses. To achieve scalability we enable proactive content delivery to key CDN nodes, by predicting future demand based on social relationships among the users and multimedia contextual information. This work presents a cloud-based architecture that could support the proposed solution for delivering social multimedia content with scalability.

Key words: Social Multimedia Content, Scalable Content Delivery, Content Delivery Networks

1 Introduction

In the recent years there has been an increased rise in the use of online social network services (OSNs), such as Facebook and Twitter, and online content sharing services, such as YouTube and Flickr. The convergence of those two services has in turn led to the development of Social Multimedia Contents, user generated content (UGC) shared online through social networks, which are becoming increasingly popular [1]. According to ForeSee [2], more than 18% of online users are influenced by the social network when accessing online multimedia contents [3].

In comparison to traditional Internet multimedia services, social multimedia content features highly dynamic contents and demands, and typically more stringent requirements. For example, a response latency in video viewing requests, which are usually short, higher than a few tens of seconds would be intolerable to a viewer. It is therefore challenging to design and scale a social media application that is most cost-effective [1]. First of all, its the intrinsic characteristics of social multimedia content. The huge volume of social multimedia content, such as videos and photographs, requires a large amount of storage and network resources. In addition, the popularity distribution of multimedia contents is long tailed, which means that most of the multimedia contents have close-to-uniform popularity profiles, and the requests to multimedia content are highly volatile. Secondly its the social propagation features, which refer to the exchange of social messages with embedded multimedia content URLs. Such information usually spreads along the social connections, especially among closely connected friends. In addition, users friends are usually located in close proximity, which means that multimedia content will be propagated in a small group with geo-locality. Furthermore, users with similar interests can be grouped into small communities that are more likely to share similar multimedia contents [1, 3, 4].

Traditional content delivery network (CDN) architectures [5] dynamically replicate multimedia content to servers at different geographical locations. Therefore, services deployed over this architecture are characterized by low flexibility on controlling underlying hardware, such as networking, computation, and storage, and pricing. These drawbacks are getting worse as the volume of multimedia contents is larger and continuously growing [1].

To address the aforementioned problems, this paper proposes the use of cloud computing to support CDNs for scalable delivery of Social Multimedia Content. Cloud computing can be defined as a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned. They are presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers [6].

The major focus of this work is to provide a solution approach for delivering social multimedia content in CDNs with scalability. To achieve scalability we enable proactive content delivery to key CDN nodes, by predicting future demand based on social relationships among the users and multimedia contextual information. The main contributions of this paper are: a) a state of the art approach for optimal distribution of social multimedia content over cloud-based CDN servers considering OSN relationships, and b) a cloud-based architecture that could support the proposed approach. The remainder of this paper is structured as follows: Section 2 presents the proposed architecture and Section 3 describes the proposed approach for scalable multimedia content delivery in CDNs. Section 4 provides a brief review on the pertinent literature, while in Section 5 the benefits of the proposed approach are discussed. Finally, Section 6 concludes this work.

2 Socially-Aware Content Delivery Architecture

To achieve scalable delivering of social multimedia content we consider a geo-distributed cloud-based social-aware architecture, inspired by [1], [4] and [7]. The proposed architecture consists of two separate layers, the cloud-based geo-distributed CDN layer and the social network layer, as shown in Fig. 1.

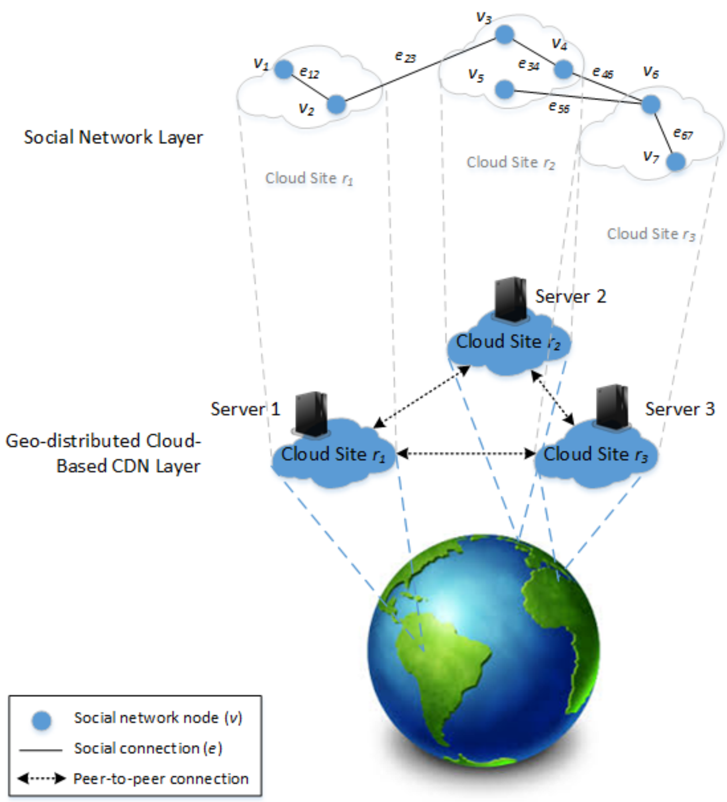

Fig. 1: Cloud-based social-aware CDN architecture

The cloud-based geo-distributed CDN layer consists of multiple cloud sites distributed in different geographical locations, owned by one or multiple cloud service providers. Each different cloud site corresponds to a CDN edge node, denoted as r, that is able to provide multimedia content to the users with the lowest latency. The proposed layer provides storage, bandwidth and computation resources from data centers that is able to offer content caching and media streaming services. Cloud providers, such as Amazon and Windows Azure, operate data centers that offer Internet scale content storage and delivery capabilities and encapsulate these services in the form of virtual machines [3]. Cloud sites are connected through a peer-to-peer (P2P) network, where content delivery is achieved among edge nodes. This P2P network can ensure the speed and quality of content delivery to the users [8].

On top of the proposed cloud-based geo-distributed CDN layer, we adopt a social network layer to provide efficient delivery of multimedia content. We view the social network as an undirected graph $G = (V, E)$, where where V is a set representing the social network users and E is a set representing the social connections between them. The proposed layer will define the optimal delivery pattern of multimedia content among the edges of the cloud-based CDN, as replicating the content in too many edges of the CDN incurs significant storage resource and limit the reduction on bandwidth resource usage [4].

3 Socially-Aware Multimedia Content Delivery

This work is based on [9] and [10]. We aim at the reduction of the response time for the user, increase of the hit ratio of our request, as well as restriction of the cost of copying from the origin server to distributed servers. We consider the network topology, the server location, and restrictions in the cache capacity of the server. Taking as input data from OSNs and actions of users over them, we want to recognize objects that will eventually be popular in the realm of the OSN platform so that to efficiently distribute them among the CDN (Fig. 2).

We search a dynamic policy such that given the social network graph G, a set of R cloud sites, where the nodes of the social network are distributed, and the posts P of the nodes, it will recognize the set of objects O, which represent social multimedia contents, that will be popular only in a subset of the cloud sites, where the content is likely to be copied. The policy is represented by the following function: function $Put(n_i, Predict(G, P, R, O))$, which takes as input a surrogate server $n_i \in N$ and the results of function $Predict$ (set of g objects

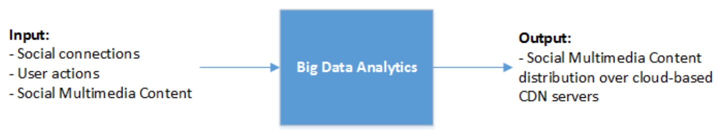

Fig. 2: Distribution of Social Multimedia Content based on OSN information

that will be globally popular and λ objects that will be locally popular), such that:

$$\frac{Q_{hit}}{Q_{total}} \tag{1}$$

is maximum, whereas constraint

$$\sum_{\forall i \in O} S_i f_{ik} \leq C_k \tag{2}$$

is fulfilled, where:

$$f_{ik} = \begin{cases} 1 & \text{if object } i \text{ exists in the cache of surrogate server } k \\ 0 & \text{if object does not exist} \end{cases} \tag{3}$$

Function (1) returns the set of objects $oinO$ that have to be placed in surrogate server $n_i inN$.

For each new request arriving in the cloud-based CDN and for each new object in the surrogate server we consider different algorithms, as described below (more details about the algorithms are provided in [9] and [10]). Internally, the module communicates with the module processing the requests and each addressed server separately, as shown in Fig. 3.

3.1 For Every New Request in the CDN

The central idea in all policies for every new request $(timestamp, Vi(t), o)$ in the CDN is to check whether specific time has passed after the start of the

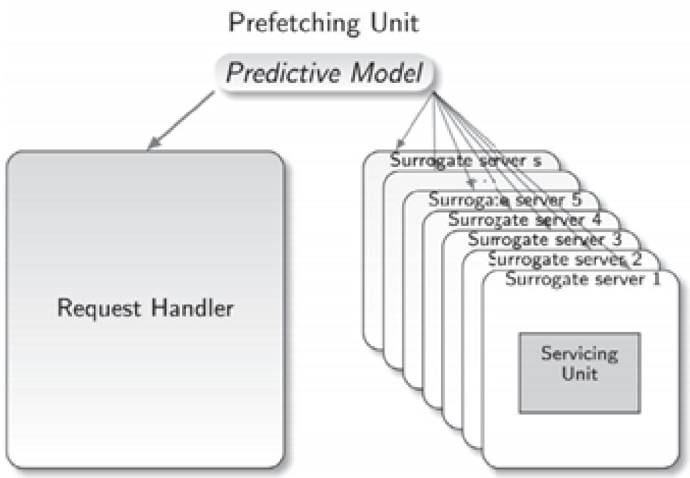

Fig. 3: Social-aware delivery of multimedia contents in a cloud-based CDN

Algorithm 1 Social Prefetcher

```
if o.timestamp == 0 then
    o.timer = 0
    o.timestamp = request_timestamp
else if o.timestamp != 0 then
    o.timer = o.timer + (request_timestamp - o.timestamp)
    o.timestamp = request_timestamp
end if
if o.timer > time_threshold then
    o.timer = 0
    o.timestamp = 0
else if o.timer < time_threshold and user.authority_score > authority_threshold
then
    copy object o to surrogate that serves user's V_i geographic zone
    for user V_y that follows user V_i do
        find surrogate server n_j that serves V_y's geographic zone
        copy object o to n_j
    end for
else if o.timer < time_threshold then
    copy object o to surrogates n_j that Subpolicy Lobby-index or Subpolicy HITS
decides
end if
```

Algorithm 2 Subpolicy Lobby-index

```
find X geographic zones where (user V_i has mutual followers  and they are closer to
user's V_i geographic zone)
find the L ⊆ X that have the highest Lobby − index score
for geographic zones that belong to L do
    find surrogate server n_j that serves geographic zone
    copy object o to n_j
end for
```

cascade, and then define to what extent the object will be copied. Initially, we check whether it is its first appearance. The variable *o.timestamp* depicts the timestamp of the last appearance of the object in a request and helps in calculating the timer related to the duration of the cascade. If it is the first appearance of the object, the timer for the object cascade is initialized and *o.timestamp* takes the value of the timestamp of the request. If the cascade is not yet complete (its timer has not surpassed a threshold), we check the importance of the user applying the Hubs Authorities (HITS) algorithm and checking its authority score (Alg. 1). Additionally we check alternatively the viewership of the object in the media service platform (Alg. 4) or if the time of the transmission is not within the peak-time range of the region of the user (Alg. 5), and devised a simple Predictive Model for efficient calculation of the number of retweets of a video (Alg. 6).

Algorithm 3 Subpolicy HITS

find X geographic zones where (user V_i has mutual followers **and** they are closer to user's V_i geographic zone
find the $H \subseteq X$ that have the highest $HITS$ score
for geographic zones that belong to H **do**
  find surrogate server n_j that serves geographic zone
  copy object o to n_j
end for

Algorithm 4 Popular Items

if $o.timestamp == 0$ **then**
  $o.timer = 0$
  $o.timestamp = request\_timestamp$
else if $o.timestamp$!= 0 **then**
  $o.timer = o.timer + (request\_timestamp - o.timestamp)$
  $o.timestamp = request\_timestamp$
end if
if $o.timer > time\_threshold$ **then**
  $o.timer = 0$
  $o.timestamp = 0$
else if $o.timer < time\_threshold$ **and** $user.authority\_score > authority\_threshold$ **then**
  copy object o to surrogate that serves user's V_i geographic zone
  for user V_y that follows user V_i **do**
    find surrogate server n_j that serves V_y's geographic zone
    copy object o to n_j
  end for
else if $o.timer < time\_threshold$ **and** $o.\Pi_i > \Pi_i\_threshold$ **then**
  copy object o to surrogates n_j that Subpolicy HITS decides
end if

For users with a high authority score, we copy the object to all surrogate servers of the users geographic zone and to the surrogate servers serving the geographic zones of all followers of the user (global prefetching). Otherwise, selective copying includes only the surrogates that the subpolicy decides (local prefetching). Subpolicies check the X closest geographic zones where a user has mutual friends and out of them, the Y with the highest value of the centrality metric as an average, denoting that the object is likely to be asked for more times. Copying is performed to the surrogate servers that serve the above geographic zones. Subpolicies for the plain algorithm, i.e. Subpolicy Lobby-index and Subpolicy HITS (Alg. 2, Alg. 3), include lobby-index and HITS ranking of geographic zones, respectively, whereas Alg.7 describes the subpolicy for the case of Predictive Model algorithm.

The heuristics applied in our approach are based on the observations that users are more influenced by geographically close friends, and moreover by mutual followers, social cascades have a short duration, whereas the majority of

Algorithm 5 Efficient Timing

```
if o.timestamp == 0 then
    o.timer = 0
    o.timestamp = request_timestamp
else if o.timestamp != 0 then
    o.timer = o.timer + (request_timestamp - o.timestamp)
    o.timestamp = request_timestamp
end if
if o.timer > time_threshold then
    o.timer = 0
    o.timestamp = 0
else if o.timer < time_threshold and user.authority_score > authority_threshold
then
    copy object o to surrogate that serves user's V_i geographic zone
    for user V_y that follows user V_i do
        find surrogate server n_j that serves V_y's geographic zone
        copy object o to n_j
    end for
else if o.timer < time_threshold then
    if o.timestamp ∋ (pts_{r_{V_i}}, pte_{r_{V_i}}) and o.timestamp ∋ (pts_{r_{n_j}}, pte_{r_{n_j}}) then
        copy object o to surrogates n_j that Subpolicy HITS decides
    end if
end if
```

cascades end within 24 hours [10]. However, we introduce a varying time threshold for the cascade effect and the time that an object remains in cache. Values given in the time threshold variable also include 48 hours, as well as threshold covering the entire percentage of requests.

3.2 For Every New Object in the Surrogate Server

Surrogate servers keep replicas of the web objects on behalf of content providers. In the case that the new object o in the surrogate server k does not fit in the surrogate servers cache, we define the time threshold as the parameter for the duration that an object remains cached. We check for items that have remained cached for a period longer than the time threshold and we delete those with the largest timestamp in the cascade. In case there exist no such objects or all objects have the same timestamp, we apply various policies for the removal of objects, including Least Recently Used, Least Frequently Used and Size-Adjusted LRU (Alg. 8)

4 Related Work

Brief review of the literature and how the proposed approach is different.

Algorithm 6 **Predictive Model**

if $o.timestamp == 0$ **then**
  $o.timer = 0$
  $o.timestamp = request\_timestamp$
else if $o.timestamp$!= 0 **then**
  $o.timer = o.timer + (request\_timestamp$ - $o.timestamp)$
  $o.timestamp = request\_timestamp$
end if
if $o.timer > time\_threshold$ **then**
  $o.timer = 0$
  $o.timestamp = 0$
else if $o.timer < time\_threshold$ **and** $user.Score > Score\_threshold$ **then**
  copy object o to surrogate that serves user's $V_i(t)$ geographic zone
  for user $V_y(t)$ that follows user $V_i(t)$ **do**
    find surrogate server n_j that serves $V_y(t)$'s geographic zone
    copy object o to n_j
  end for
else if $o.timer < time\_threshold$ **then**
  copy object o to surrogates n_j that Subpolicy Score decides
end if

Algorithm 7 **Subpolicy Score**

find the Y geographic zones that depict the highest average values of Predictive Model($Score$, $dScore$, $content\_dist$) for user $V_i(t)$
for geographic zones that belong to Y **do**
  find surrogate server n_j that serves geographic zone
  copy object o to n_j
end for

5 Discussion

A brief discussion on the benefits arising from the proposed approach.

6 Conclusions

As the use of online social network services, such as Facebook and Twitter, and online content sharing services, such as YouTube and Flickr, has increased over the past years, Social Multimedia Contents are becoming increasingly popular. In addition, more than 18% of online users are influenced by the social network when accessing online multimedia contents. In this report, to address the problem of scalability in delivering social multimedia contents we proposed the use of cloud computing to support traditional CDNs. Advanced data analysis algorithms are used to predict social multimedia content consumption for optimal replication of multimedia content.

Algorithm 8 **Caching scheme LRU / LFU / SIZE**

```
if o.size + current_cache_size ≤ total_cache_size then
    copy object o to cache of surrogate n_k
else if o.size + current_cache_size > total_cache_size then
    while o.size + current_cache_size > total_cache_size do
        for object o′ in current_cache do
            if (current_timestamp - o′.timestamp) + o′.timer > time_threshold then
                copy o′ in CandidateList
            end if
            if CandidateList.size > 0 and CandidateList.size! = total_cache_size then
                find o′ that o′.timestamp is maximum and delete it
            else if CandidateList.size == 0 ∨ CandidateList.size == total_cache_size then
                use LRU/LFU/SIZE to delete any object o ∈ O
            end if
        end for
    end while
    put object o to cache of surrogate n_k
end if
```

References

1. Zhi Wang, Wenwu Zhu, Xiangwen Chen, Lifeng Sun, Jiangchuan Liu, Minghua Chen, Peng Cui, and Shiqiang Yang. Propagation-based social-aware multimedia content distribution. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(1s):52, 2013.
2. Foresee. `http://www.foreseeresults.com`. Accessed: 2012.
3. Han Hu, Yonggang Wen, Tat-Seng Chua, Jian Huang, Wenwu Zhu, and Xuelong Li. Joint content replication and request routing for social video distribution over cloud cdn: A community clustering method. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(7):1320–1333, 2016.
4. Han Hu, Yonggang Wen, Tat-Seng Chua, Zhi Wang, Jian Huang, Wenwu Zhu, and Di Wu. Community based effective social video contents placement in cloud centric cdn network. In *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, pages 1–6. IEEE, 2014.
5. Kianoosh Mokhtarian and Hans-Arno Jacobsen. Caching in video cdns: Building strong lines of defense. In *Proceedings of the Ninth European Conference on Computer Systems*, page 13. ACM, 2014.
6. Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6):599–616, 2009.
7. Yu Wu, Chuan Wu, Bo Li, Linquan Zhang, Zongpeng Li, and Francis Lau. Scaling social media applications into geo-distributed clouds. *IEEE/ACM Transactions on Networking (TON)*, 23(3):689–702, 2015.
8. Yuedui Wang, Xiangming Wen, Yong Sun, Zhenmin Zhao, and Tianpu Yang. The content delivery network system based on cloud storage. In *Network Computing*

and Information Security (NCIS), 2011 International Conference on, volume 1, pages 98–102. IEEE, 2011.
9. Irene Kilanioti and George A Papadopoulos. Socially-aware multimedia content delivery for the cloud. In *Utility and Cloud Computing (UCC), 2015 IEEE/ACM 8th International Conference on*, pages 300–309. IEEE, 2015.
10. Irene Kilanioti and George A Papadopoulos. Content delivery simulations supported by social network-awareness. *Simulation Modelling Practice and Theory*, 71:114–133, 2017.