

Track Globally, Deliver Locally: Improving Content Delivery Networks by Tracking Geographic Social Cascades

Salvatore Scellato
Computer Laboratory
University of Cambridge
ss824@cam.ac.uk

Mirco Musolesi
School of Computer Science
University of St. Andrews
mirco@cs.st-andrews.ac.uk

Cecilia Mascolo
Computer Laboratory
University of Cambridge
cm542@cam.ac.uk

Jon Crowcroft
Computer Laboratory
University of Cambridge
jac22@cam.ac.uk

ABSTRACT

Providers such as YouTube offer easy access to multimedia content to millions, generating high bandwidth and storage demand on the Content Delivery Networks they rely upon. More and more, the diffusion of this content happens on online social networks such as Facebook and Twitter, where social cascades can be observed when users increasingly repost links they have received from others.

In this paper we describe how geographic information extracted from social cascades can be exploited to improve caching of multimedia files in a Content Delivery Network. We take advantage of the fact that social cascades can propagate in a geographically limited area to discern whether an item is spreading locally or globally. This informs cache replacement policies, which utilize this information to ensure that content relevant to a cascade is kept close to the users who may be interested in it.

We validate our approach by using a novel dataset which combines social interaction data with geographic information: we track social cascades of YouTube links over Twitter and build a proof-of-concept geographic model of a realistic distributed Content Delivery Network. Our performance evaluation shows that we are able to improve cache hits with respect to cache policies without geographic and social information.

Categories and Subject Descriptors

H.3.5 [Information Storage and Retrieval]: Online Information Services; C.2.4 [Distributed Systems]: Distributed applications

General Terms

Measurement, Design

Keywords

Online Social Networks, Content Delivery Networks, Information Propagation

1. INTRODUCTION

The amount of Internet traffic generated every day by online multimedia streaming providers such as YouTube has reached unprecedented numbers [3, 7]. These providers often rely on Content Delivery Networks (CDNs) to distribute their content from storage servers to multiple locations over the planet. CDN servers exchange content in a cooperative way to maximize the overall efficiency.

Nowadays content diffusion is fostered by weblinks shared on Online Social Network (OSN) posts, which may often generate floods of requests to the provider through the cascading across a user's social links. For instance, more than 1 billion pieces of content are shared each day on Facebook. This type of "word-of-mouth spreading" occurring in OSNs may already be driving a large part of the daily requests to content providers. Even if the proportion of traffic generated by social spreading is difficult to estimate, on Twitter there are more than 400 messages per minute with a YouTube link [17]. Given the increasing size of Twitter and other OSNs, they may generate millions of accesses to YouTube, accounting for a consistent fraction of the total number of daily requests. The resulting load on a CDN is exacerbated by all those users who see the links and who might request the content, even if not reposting it.

However, OSNs are increasingly becoming location-aware, allowing users to share information about their geographic locations. There are OSN services based on the idea of advertising the exact user location, such as Foursquare and Gowalla. Similarly, Twitter has always provided its users with the option of sharing some information about their location, with a considerable proportion of them already doing so. The availability of geographic information for popular online social networks opens unprecedented opportunities to enhance engineering of world-wide systems based on human communication and interaction [12, 14, 2, 15].

In fact, in this paper we show how geographic information extracted from social cascades over OSNs can be exploited to improve the design of large-scale systems, such as CDNs. We rely on this novel finding: *social cascades are likely to spread on geographically local distances*. Users tend to share content over short-distance social connections, despite the presence of several long-range links: although many users exhibit these long-distance connections, we have found that still about 40% of steps in social cascades involve users that are, on average, less than 1,000 km away from each others.

Our key idea is that *content should be kept within servers which are close to users that are interested in it, minimizing the impact on network traffic*. In other words, since content servers act as caches of items, we aim to introduce geo-social characteristics of the users that are sharing the content in the design of large-scale systems such as CDNs, so that we can serve more requests immediately from the closest server rather than waiting for the content to be transferred to the server from somewhere else.

In order to validate our approach, we use a *novel and unique geo-social dataset from Twitter containing geographic location, follower lists and tweets for 409,093 users*. Then, we have tracked the spreading of more than one million of YouTube videos over this network, analyzing a corpus with more than 334 millions messages and extracting about 3 millions single messages with a video link. Finally, we have designed a proof-of-concept CDN model using the geographic properties of the commercial CDN once used by YouTube, Limelight. We show how cache replacement policies driven by geo-social metrics improve the system performance.

Our contributions can be summarized as follows:

- We present an extensive study of information dissemination on a novel geo-social dataset gathered from a largely popular OSN (Twitter): by taking into account user geographic information we are able to investigate the extent of social cascades and to characterize them over space and time.
- We propose to enhance large-scale system design with geo-social properties of social cascades, and we build a proof-of-concept CDN model where content cache replacement strategies exploit these properties.
- We validate our approach through simulation: we generate realistic workloads extracted from messages in our dataset and we show how geo-social information can improve cache performance with respect to standard replacement policies.

The paper is structured as follows: Section 2 motivates the work describing the problems of current CDNs, while in Section 3 we describe the geo-social properties of our Twitter dataset. Section 4 illustrates how geographic social cascades of YouTube links extracted from our Twitter dataset spread over space. Strategies to improve content provision based on social cascades are presented in Section 5 along with a CDN geographic model, while Section 6 reports the results of our evaluation of the model. In Section 7 we discuss the implications of our results and in Section 8 we offer a review of related work, concluding the paper with Section 9.

2. IMPROVING CDN PERFORMANCE

A Content Delivery Network (CDN) is a system of networked servers holding copies of data items, placed at different geographic locations. The aim of a CDN is to efficiently deliver content to clients: each request is served by a geographically close server, while content is moved across servers to optimize the quality of service perceived by users. Modern commercial CDNs deploy numerous servers all over the Internet, often over multiple backbones and ISPs, and offer their services to other companies which want to deliver content to users on a planetary scale, such as dynamic Web pages, software updates, multimedia content, live streams and so on [11].

2.1 Factors Impacting CDN Performance

CDNs have become progressively more important: the number of users with broadband Internet connections is constantly increasing and, along with faster connectivity, come greater expectations for better content delivery. As an example, requests of video content over the Web are more than 1 billion per day. Even if exploiting additional resources provided by CDNs, this demand puts a considerable pressure over the entire Internet. This issue becomes even more important if we consider future trends: as the size of distributed content keeps growing, the distance between server and client becomes more critical to the overall performance, since longer distances increase the likelihood of network congestion and packet loss which result in longer transfer times [11].

In addition, the geography of the requests influences the performance of CDNs: it would be extremely useful to understand whether an item becomes popular on a planetary scale or just in a particular geographic area. A globally popular content item should be replicated at every location, since it experiences many requests from all around the world. On the other hand, when content is only locally popular, it should be cached only in the locations that will experience most requests. The key to such a strategy is being able to predict quickly whether a piece of content is becoming locally popular in order to optimize its placement over the CDN before it undergoes the popularity surge.

2.2 Improving CDN Performance through Social Cascades

The popularity of content over the Web can be driven by public media coverage or through word-of-mouth spreading [4]. The former takes place when content is advertised by large information sources, such as search engines, news and social aggregator websites (i.e., SlashDot, Reddit, etc.). This type of phenomena often results in globally popular items, which should be widely replicated throughout a CDN, since they are likely to experience requests from all over the world. On the other hand, content may become popular because people share it and talk about it, leading to some sort of viral spreading along social connections. These connections may be real-life contacts or interactions on online social networks, with the latter becoming increasingly common.

As a result, content may easily spread from a small set of users to a vast audience through *social cascades* [4]. The amount of content requests generated by these social cascades is hard to estimate. In our analysis of a real Twitter dataset, we have found that about 1% of Twitter messages contain a link to a YouTube video. This suggests that a potentially large number of content requests might be generated by a cascade. The combined effect of the popularity of several online services may cause millions of such requests.

Social cascades can be tracked and analyzed: OSNs can provide all the information, including user location, to track items shared by users and to understand the properties of a social cascade while this evolves over time. To exploit these aspects for improving CDN performance, we need to understand the key geographic properties of social networks. For instance, we need to study and characterize how social cascades are unrolling over space and analyze if geography affects the spreading process. In particular, *is it possible to estimate whether cascades will spread globally or locally just*

from the geo-social properties of the users participating in the cascade?

In the next sections we will answer this question: we will describe the geo-social properties of a geographic social network and, then, we will characterize how social cascades evolve over space. Finally, we will illustrate how we can exploit our findings to improve CDN performance.

3. GEO-SOCIAL NETWORK METRICS

OSNs encourage users to share personal and activity information with friends. Users are also prompted to indicate their location: for example, they can provide details about their hometown, home neighborhood or, more recently, their exact location, by uploading their mobile device’s GPS readings. In this section we analyze a large Twitter dataset and we illustrate the geo-social properties of its users.

3.1 Geographic Social Networks

We adopt the representation of a *Geographic Social Network* presented in [14], where connections between nodes are weighted with the geographic distance between them. Moreover, we exploit *node locality* to quantify the geographic closeness of the friends of a certain node.

Let us consider a node i and the set Γ_i of its neighbors. The node degree k_i is the number of these neighbors, that is $k_i = |\Gamma_i|$. Then, we can define its *node locality* as a measure of how much geographically close its neighbors are. The locality of node i is computed as

$$L_i = \frac{1}{k_i} \sum_{j \in \Gamma_i} e^{-l_{ij}/\beta} \quad (1)$$

where l_{ij} is the weight of the link between nodes i and j and β is a scaling factor. This definition can be generalized to directed graphs to compute *node in-locality* and *node out-locality*.

3.2 Twitter Geo-social Properties

Twitter is a social networking service where users can send 140-character short text messages called *tweets*. Tweets are shown on the author’s personal page and also sent to the author’s *followers*. It does not enforce reciprocity in social connections: a user can follow another one even if the latter is not following back. As a result, the social graph is directed.

In this work we use a Twitter dataset whose collection and preprocessing is described in [14]. We extract a directed graph from our dataset where each node represents a user with a geographic location and a link from user A to user B denotes that user A follows user B . This graph has $N=409,093$ nodes and $K=182,986,353$ directed links, with an average degree of 477. The average geographic distance of a social link is 5,117 km, which indicates how Twitter users tend to engage in long-range connections.

We compute node in-locality for every user in the dataset, as given in Eq. (1) but taking into account only the set of incoming links. This geo-social metric is equal to 1 when all the followers of a user are in the same location as the user and decreases to 0 when they are further away. In this work we have chosen a network scaling factor of $\beta = 2,000$ km: this value allows us to assign two users which are about 1,000 km apart with a locality value of $e^{-0.5} \approx 0.6$. We report

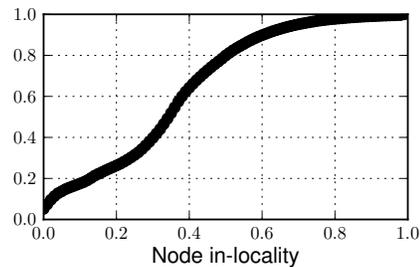


Figure 1: Cumulative Distribution Function of node in-locality for the Twitter graph. Node in-locality is computed using a scaling factor of $\beta = 2000$ km.

the distribution of node in-locality in Fig. 1: in the Twitter dataset the node locality distribution exhibits a large mass around 0.3, with a few nodes that have a locality value larger than 0.6. Nonetheless, there are around 10% of users with node locality greater than 0.6 and 20% with node locality greater than 0.5. This indicates that only a minority of Twitter users have a geographically limited audience, whereas many other users have a set of followers which encompasses a much larger region. On the other hand, we will show in the next section how users are instead spreading content mainly over the less frequent shorter connections.

4. IDENTIFYING GEOGRAPHIC SOCIAL CASCADES

We now describe how we extract and evaluate social cascades over geographic social networks. Indeed, OSNs represent a popular way of sharing information. A piece of information can quickly disseminate from a user to another as a virus in an epidemics: somebody shares some new content with its friends who might share it again and so on. We usually refer to this phenomenon as *social cascade* [4]. We define two measures that quantify how a social cascade is spreading over space and the extent of its propagation. Finally, we analyze how information spreads over the geo-social structure of Twitter, focusing on traceable bits of information: weblinks to YouTube videos contained in tweets.

4.1 Geographic Social Cascades

A cascade over a social network begins when the first user shares some content and becomes the initiator of the cascade. After this event, some of his contacts will share the same content again, with the result that the cascade will recursively spread over the social links.

In order to estimate the influence of the social network on the information dissemination process, we combine the information about social connections (i.e., list of friendships in OSNs) with the time instants of the posts of each user.

More formally, we say that user B was reached by a social cascade about content c if and only if:

- there is another user A which posted content c and
- user A posted content c before B posted it and
- there was a social connection from user A to user B when A posted c .

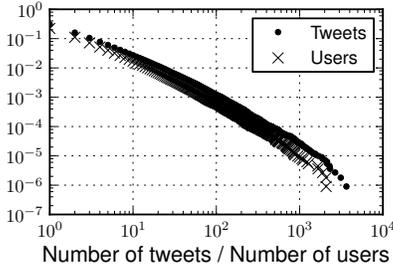


Figure 2: Complementary Cumulative Distribution Function of the number of tweets containing a given video link and of the number of users tweeting a given video link.

While this does not guarantee the dependency of the posts, in most cases we conjecture that there is a correlation between the two events. If more than one user among the social connections of B posted c , we say that B was reached by the cascade only through the user which posted it last. Therefore, we always have only one previous user in the cascade process. This arbitrary choice will only affect the shape of the cascade, not its size nor its overall geographic properties.

In order to describe these social cascades we exploit the geographic social network model defined in the previous section. This cascade can be represented as a tree over the geographic social network, with the initiator node as the root of the tree. For the same item there might be more than a single cascade. Moreover, the same user may publish the same content at different times. In order to take into account these details, we need to annotate the cascade links with temporal information. Each social cascade is represented as a tree, where a link from user A to user B indicates that user B has received some content as a result of a social cascade from user A . A link between A and B is annotated with the time instants t_1 and t_2 : t_2 is the time instant when B posted content c for the first time and t_1 is the time instant of the last time user A posted content c before B did, so that $t_2 \geq t_1$. We define such a cascade step by using a time threshold: consecutive steps in a cascade must be within 48 hours from each other.

In order to investigate the geographic properties of a social cascade we define two measures:

- the **geodiversity** of a cascade is the geometric mean of the geographic distances between all the pairs of users in the cascade tree;
- the **georange** of a cascade is the geometric mean of the geographic distances between the users and the root user of the cascade tree.

We adopt the geometric mean since geographic distances span several orders of magnitude in our dataset. For a given social cascade these two quantities are correlated, however they can be used to emphasize different properties of the cascade. The geodiversity is computed among all the pairs of users in a cascade, regardless of whether they are connected or not, while the georange is only related to the cascade initiator. On the other hand, the georange allows us to understand how close the initiator of a cascade is to the other people involved in it.

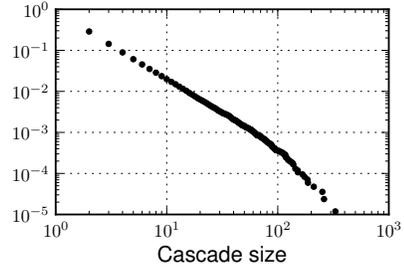


Figure 3: Complementary Cumulative Distribution Function of social cascade size.

4.2 Cascades of YouTube Links on Twitter

YouTube is the largest user-generated video content website, which allows users to upload their own videos and then share them on a variety of platforms and devices. Twitter fosters the popularity of YouTube, since its users tend to tweet about videos they like, triggering a spreading of the video links that can be classified as a viral phenomenon [3].

Through the Twitter API we have access to the 3,200 most recent tweets for each user in our geographic Twitter graph. We downloaded these tweets for all the users in the graph, obtaining 334,407,185 tweets. The duration of the data crawling was 12 days, from February 1 to February 11, 2010. For each tweet we have crawled the author, the time when it was sent and the actual content of the message. Among these tweets we have isolated 570,617 messages containing a direct link to a YouTube video. Furthermore, we have extracted all the messages containing a URL shortened with URL shortener services, obtaining additional 2,332,390 messages with a YouTube link.

Thus, after removing invalid YouTube links, we extract a total of 2,903,007 tweets containing a valid direct link to a YouTube video. These links point to 1,111,586 different YouTube videos, hence some videos are contained in more than a single tweet. The average number of tweets per YouTube video is 2.61. In Fig. 2 we show the popularity distribution for the video links we have extracted: both the distribution of the number of tweets containing a given video link and the number of different users tweeting a given video link have heavy tails. Thus, there is a very small amount of videos that are tweeted more than 4,000 times or by thousands of different users, while the vast majority is tweeted only 1 or 2 times by a few users. Such popularity distribution can greatly affect content delivery, since it shows how popular items can easily dominate in terms of number of requests. Furthermore, every tweet is potentially spawning many more actual video requests, since all followers of the author can view the link and follow it. While difficult to estimate, this portion of additional traffic might even constitute a large fraction of social-driven web traffic.

Then, we use the cascade definition presented in Section 4.1 to analyze the tweets and, as a result, we extract 84,337 social cascades for 63,798 different videos. Each cascade involves the initiator and at least another user. Unfortunately, we have no information about when a user started to follow another one, so we assume that all the social relationships that we have in our social graph were in place when the tweet was sent.

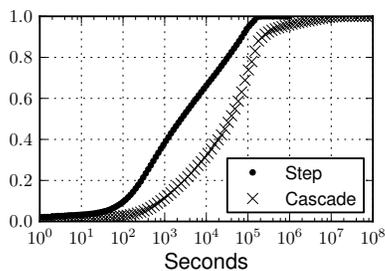


Figure 4: Cumulative Distribution Function of time delay between two consecutive tweets and total cascade duration (from the first to the last tweet). Cascade duration is shown only for cascades with at least two users except the initiator.

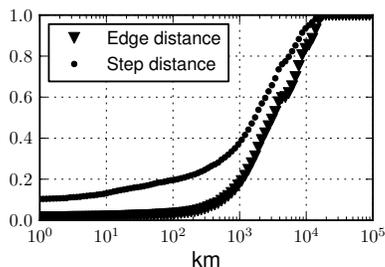


Figure 5: Cumulative Distribution Function of cascade step distance and of social connection distance: social cascades take place on short-range social connection.

4.3 Analysis of Social Cascades

We define the *size* of the cascade as the number of users involved in it, including the initiator. In Fig. 3 we report the distribution of the cascade size: we notice again a long-tail, with more than 60,000 cascades involving only two nodes and a few cascades reaching up to hundreds of users. This measure of popularity demonstrates how it is rare to have large cascades, but when they do take place they can become extremely large. Again, it is worth noting that social cascades include only users that have tweeted a certain video link: however, each tweet can be viewed by all the followers of the author, thus the potential audience that a YouTube video may reach by means of a social cascade is much larger, even if only few users are involved.

In Fig. 4 we illustrate the distribution of the *time delay* between two consecutive tweets in a cascade. About 40% of the tweets in cascades have a delay of about 15 minutes from the previous message, with around 10% having a delay of around 2 minutes. This result shows how YouTube links can spread on Twitter on a time scale of some minutes, even though further spreading does happen even after some hours. This indicates that links to videos can quickly spread over the social network, potentially leading to many views in a short period of time. In Fig. 4 we also show the distribution of cascade duration from the first tweet to the last tweet for each cascade with at least 2 users except the initiator: about 80% of the cascades end within 24 hours, with 40% ending in less than 3 hours.

In Fig. 5 we show the distribution of the *geographic dis-*

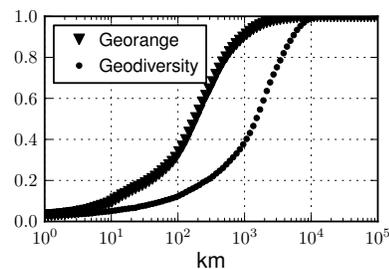


Figure 6: Cumulative Distribution Function of geodiversity and georange for social cascades with at least 2 users after the initiator.

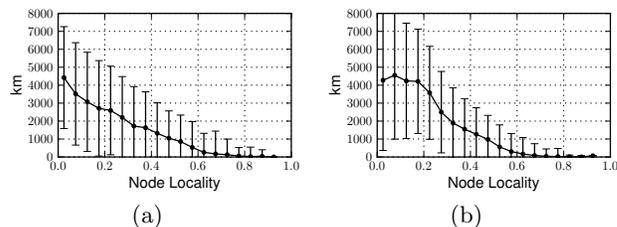


Figure 7: Average geodiversity of a social cascade as a function of the average locality of the first nodes in the cascade: locality of the first node (a) and of the first two nodes (b). Error bars show standard deviation around the average.

tance between authors of two consecutive tweets in a social cascade. Around 10% of cascade steps are less than 1 km, with 20% of them shorter than 100 km and more than 30% shorter than 1,000 km. This result is in slight contrast with the distribution of link lengths of the Twitter network, also presented in Fig. 5: even if less than 5% of the social connections are shorter than 100 km, within cascade steps this fraction increases up to 20%. Content spreading through social cascades is more likely than expected to travel over geographically short-range social connections rather than over the more numerous long-distance links.

Moreover, in Fig. 6 we show the distribution of *geodiversity* and *georange* for all the social cascades which involve at least two users except the initiator. About 40% of these cascades exhibit geodiversity lower than 1,000 km, with around 20% of geodiversity values lower than 300 km. Thus, even though many cascades reach a broad audience, some of them remain geographically limited. On the other hand, about 90% of georange values are smaller than 1,000 km, with about 30% of values smaller than 100 km. This is an indication that a cascade may take place in a broad region but with each user still close to the initiator.

Finally, we are interested in properties of a social cascade that may help us predict its geographic spreading from the very first messages that are tracked. Thus, we investigate how the node locality of the first users that participate in a social cascade is related to the final geodiversity and georange values. We report in Fig. 7 the average cascade geodiversity as a function of the average locality of the first users involved in the cascade. We observe that even the initial locality of the first user is already correlated with the geographic spreading of the cascade. Moreover, by including

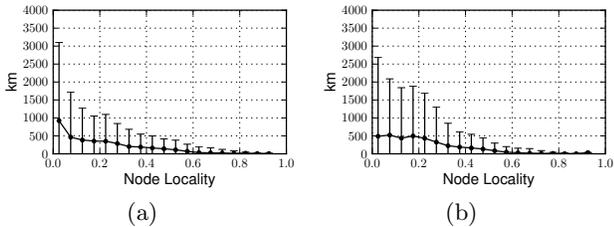


Figure 8: Average georange of a social cascade as a function of the average locality of the first nodes in the cascade: locality of the first node (a) and of the first two nodes (b). Error bars show standard deviation around the average.

the locality of the second user we get a stronger relationship. A similar result can be seen in Fig. 8 for the georange: in this case the correlation is clearer, with less variance especially for high locality values. Thus, the final properties of a cascade can be estimated even from the users involved in the initial stages. Also, even the geographic and social properties of the initiator are sufficient to understand whether a cascade will spread locally or globally, and by taking into account a few more steps we are able to give a more accurate estimate of the final outcome.

Given the importance of social cascades and their geographic properties, in the next section we will show how we can exploit these findings to improve the design of cache replacement strategies for CDNs.

5. GEO-SOCIAL CDN MODEL

We have described how geography can be included in the analysis of social networks and we have presented the geographic properties of social cascades. We now show how these findings can be exploited in the design of a proof-of-concept CDN with improved performance.

5.1 Assumptions and System Model

We envisage a single entity able to access information about content shared by users on social networks and control the CDN which delivers the content that users are sharing. This can be mapped to reality in various ways: i) assuming that CDNs will have access to information from OSNs about the dynamics of cascades, which is reasonable as they are providing the content sharing service or ii) assuming that, plausibly, in future OSNs and content providers might merge into single entities or cooperate (e.g., it is not unlikely that companies like Facebook might expand their business and also become content providers).

We model our system as a collection of server clusters placed around the planet. Each cluster contains a certain number of servers: we assume that all servers within the CDN have identical properties. The only difference among clusters is the amount of deployed servers. We assume that there is a central catalogue of content items: clients from all over the world request content items to the CDN and they are redirected to the geographically closest server. If the server already contains the requested item, it is immediately served. Otherwise, the item is retrieved from another portion of the CDN and served. We assume that, as observed in real systems [11], different clusters are interconnected by a dedicated network. Then, we assume that it is faster to

Location	Country	Servers	Location	Country	Servers
Washington	USA	552	Frankfurt	Germany	314
Los Angeles	USA	523	London	UK	300
New York	USA	438	Amsterdam	Netherlands	199
Chicago	USA	374	Tokyo	Japan	126
San Jose	USA	372	Toronto	Canada	121
Dallas	USA	195	Paris	France	120
Seattle	USA	151	Hong Kong	Hong Kong	83
Atlanta	USA	111	Changi	Singapore	53
Miami	USA	111	Sydney	Australia	1
Phoenix	USA	3			

Table 1: Geographic distribution of the server clusters in the Limelight network.

move content among servers to bring an item as close as possible to the client, rather than redirecting the request to another server further away which already holds a copy. This seems plausible, even if geographic distance may not always be the only factor influencing performance.

Server clusters act as caches: they keep copies of already requested items for future requests but they have finite storage. A cache replacement strategy is used to remove an item from the cache when this is full. We also assume that the servers within a cluster coordinate to act as a single large cache. Therefore, every server can host up to k items and if there are N servers in a cluster, that cluster is equivalent to a single cache able to host kN different items. This simplifies the definition of the model but still captures the heterogeneity of cluster sizes around the planet. We do not model file size: we assume that the size of a file does not vary much across the items, as we have observed in our specific dataset of YouTube videos.

5.2 Model Parameters

In order to ground our model in reality we have parametrized our CDN model with the real properties of Limelight [9]¹, the commercial CDN once used by YouTube to deliver content to users worldwide. Limelight has clusters of servers deployed at 19 different locations around the world and each cluster has a different number of servers. Limelight deploys 2,830 out of its 4,147 servers in the United States, where there are 10 clusters. On the other hand, Europe and Asia are served only by seven clusters in total and Australia only by one, while the rest of the world does not contain any cluster.

In our model, cache size should be interpreted with respect to the total number of items present in the system and not as an absolute number, since we do not have access to the whole YouTube item catalogue. Hence, we will also express cache size as a percentage of the total data catalogue. As an example, since we have about 1 million videos in our dataset, a cache size of 100 items is comparable to a cache that can host about 0.01% of a real catalogue: in the case of YouTube, with hundreds of thousands of videos added every day, there are more than 100 millions videos, hence this would represent a cache size with more than 10,000 different videos.

¹We are aware that the paper has been withdrawn by Microsoft for some criticisms about the system performance results presented in it. We only use information about server locations from this work.

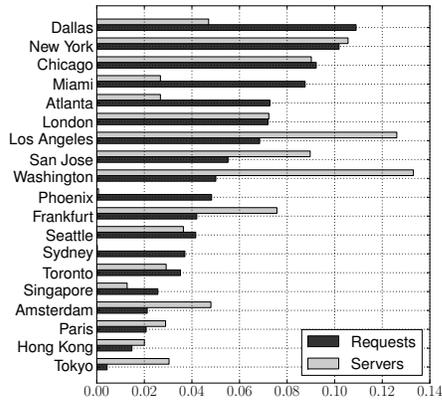


Figure 9: Fraction of video requests handled by each cluster and fraction of servers contained in each cluster. Different workloads do not significantly change the distributions of requests.

5.3 Content Caching Policies

We now define the caching policies adopted by our model to store and replace content within the servers. A server cluster adopts a *cache replacement strategy* to remove an item when the cache is full and a new request arrives. Each strategy assigns priorities to the items in memory and, when a deletion is needed because the cache is full, the item with the lowest priority is removed. The priority of an item might be updated whenever a request for that item is issued.

Our approach is to use standard caching policies and then augment them with geo-social information. Each policy assigns a priority $P(v)$ to a video v and, when a video has to be removed, that with the lowest priority is chosen for deletion. A random choice is made when more videos have the lowest priority. We adopt three different caching policies: *Least-Recently-Used (LRU)*, *Least-Frequently-Used (LFU)* and *Mixed*. In LRU the priority of a video v is given by $P(v) = clock$, where *clock* is an internal counter which is incremented by one whenever a new item is requested. This policy provides a simple aging effect: when an item is not requested for a long time, it is eventually removed. However, it does not take into account item popularity. In LFU the priority of a video v is given by $P(v) = Freq(v)$, where $Freq(v)$ is the number of times video v has been requested since it was stored in the cache for the last time. LFU favors popular content: if an item receives a large number of requests it will stay in the cache for a long time. However, LFU is less flexible: an item which was largely popular in the past tends to stick in the cache even if it is not requested anymore. The Mixed policy combines both LRU and LFU features and the priority of video v is given by $P(v) = clock + Freq(v)$, in order to balance both temporal and popularity effects [5]. In this case *clock* starts at 0 and it is updated for each replacement with the priority value of the removed file. Thus, a video increases its priority when it is requested many times, but, if there are no more requests, it will eventually be removed from the cache.

Then, we define two *priority weights* for each video v , based on the characteristics of the social cascades involving this video:

- **Geosocial:** the weight of video v is given by the sum of the node locality values of all the users that have posted a message about it, even if they are not involved in a social cascade;
- **Geocascade:** the weight of video v is given by the sum of the node locality values of all the users participating in the item’s social cascade (or cascades, if an item happens to be posted on more than one cascade).

These weights are used to capture the idea that if a video is tweeted many times by users with high node locality values, then it is likely that it is spreading in a local region, thus future requests will hit the same content server. While the first weight takes into account all the messages regarding a particular content item, the second one only uses the messages caused by a social cascade. By using two different weights based on geo-social information we want to investigate what contribution social cascades provide with respect to using only geographic information of social ties.

The weight of every tweet with a link to a video is updated according to whether that tweet is or is not in a cascade. For every request, content servers get also the video weight and multiply it to the priority of the underlying cache replacement policy. Hence, for every cache replacement strategy we have three different versions: with no weight, with a Geosocial weight and with a Geocascade weight.

6. EVALUATION

In order to test whether information extracted from geographic social cascades can effectively be exploited to improve the performance of CDNs we have investigated through simulation how different cache replacement policies impact the performance of the system. Our results show that global system performance can be improved with respect to standard policies, which means potentially avoiding tens of millions of video file transfers per day.

6.1 Simulation Strategy

In our simulation we create a sequence of content requests to the CDN directly from the Twitter messages within our dataset. We assume that every video contained in a Twitter message is requested by each follower of the author with a certain probability p and with a random temporal delay modeled with the same distribution of delay between cascade steps. This assumption is simple and can be far from reality, but we do not have information about what amount of traffic arises from Twitter messages. However, our results show improvements for any value of p we adopted. We generate 5 different workloads, corresponding to the values of $p = 0.001, 0.002, \dots, 0.005$, and we run every workload for 20 different times, averaging the results.

As said, we always route a request to the server cluster closest to the user. However, the geographic distribution of the requests does not change for p , since it is only influenced by the geographic distribution of Twitter users, which does not change for different workloads. As shown in Fig. 9, some servers receive much more traffic than others: as an example, the cluster in Dallas accounts for more than 11% of global requests. Additionally, some locations hold a large fraction of traffic even though they contain only a small number of servers. These properties may impact the performance of the cache replacement strategies for different locations.

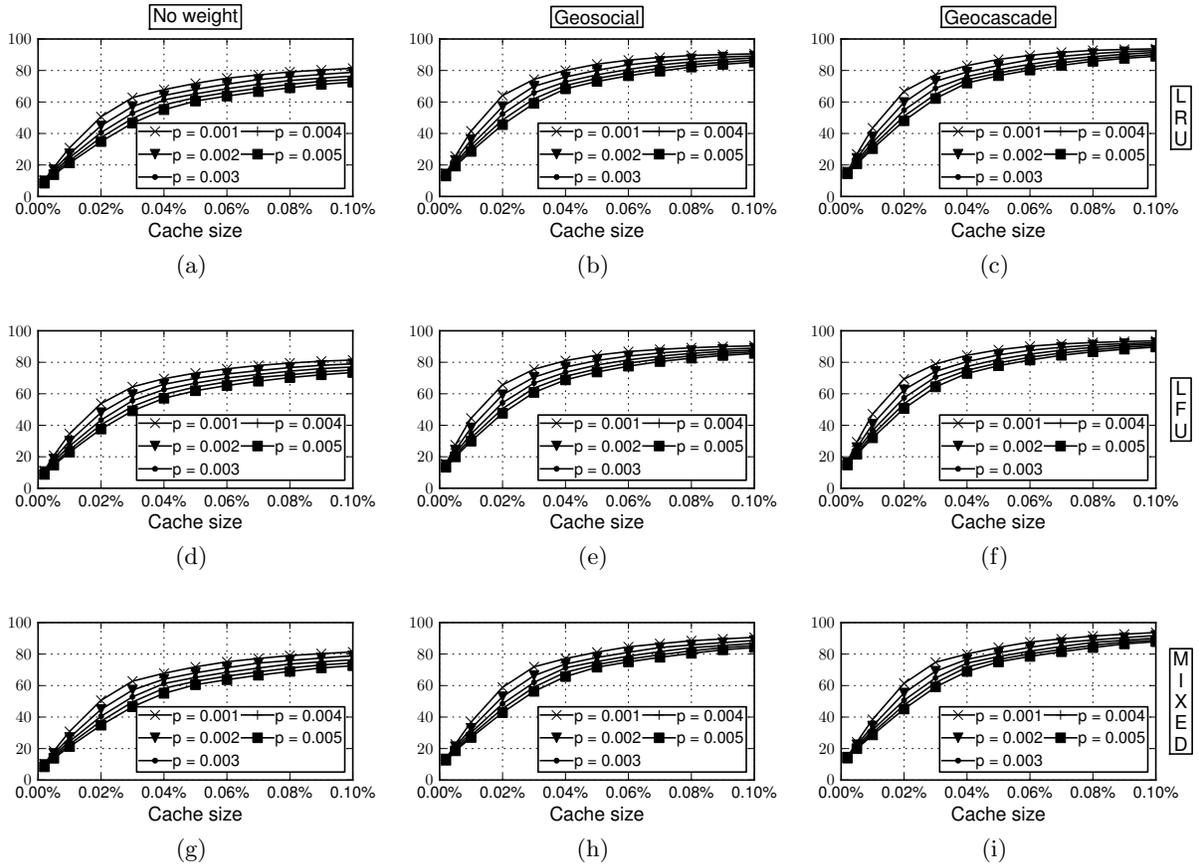


Figure 10: Percentage of total hits with respect to the infinite cache case as a function of cache size for different data combinations. Cache size is expressed as a fraction of the entire data catalogue. In the first row LRU cache policy is shown, in the second row LFU and in the third row Mixed. In the first column no weight is used, in the second column Geosocial weight and in the third column Geocascade weight. Every simulation is run 20 times with randomly generated workloads; standard deviation is negligible.

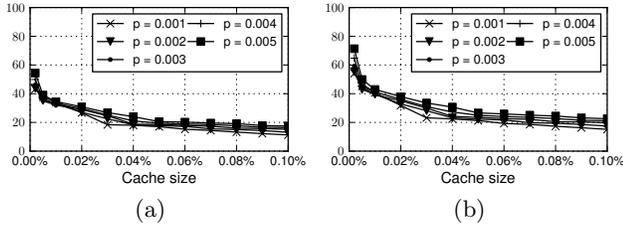


Figure 11: Performance increment (%) with the respect to the case without weight as a function of the cache size and for different workloads when the LRU strategy is used with Geosocial weight (a) and with Geocascade weight (b).

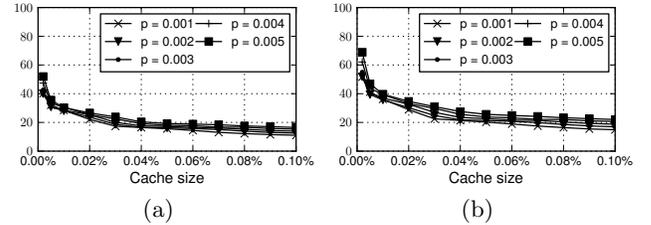


Figure 12: Performance increment (%) with the respect to the case without weight as a function of the cache size and for different workloads when the LFU strategy is used with Geosocial weight (a) and with Geocascade weight (b).

6.2 Global Performance

First we investigate how different policies perform with respect to the case of infinite cache size, i.e., in conditions where no item is ever removed from the cache: the number of hits in this case is the maximum achievable, both on each cluster and globally.

As a global performance metric for our system we consider all the hits on all the clusters: every request is directed to

the closest server and there it may result in a hit or a miss. For each cache replacement strategy and for each different workload, we compute the total number of hits obtained and we take the ratio between this value and the performance with infinite cache. This metric shows how different policies react when some parameters of the system are changed, but it does not emphasize differences in their performance.

In Fig. 10 we observe the change in system performance

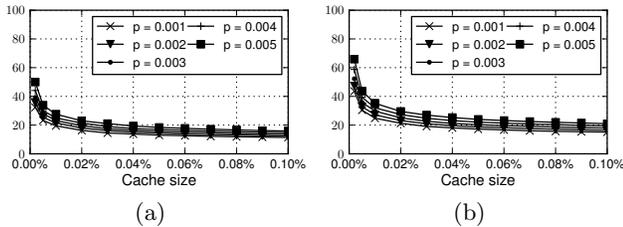


Figure 13: Performance increment (%) with the respect to the case without weight as a function of the cache size and for different workloads when the Mixed strategy is used with Geosocial weight (a) and with Geocascade weight (b).

as a function of cache size and for different workloads: when the size increases, every policy steadily improves its performance. Larger workloads have worse performance, but differences among them disappear at larger cache sizes. Moreover, as the cache size grows larger, all workloads reach a plateau, since increasing the cache size beyond a certain limit provides only a diminishing performance increment. This is due to the fact that there is a portion of content which is requested only a few times and for which caching policies can hardly offer advantages. In addition, we observe that while using no weights results in the lowest hit ratio, by adopting instead the Geosocial and Geocascade weights we achieve noticeable improvements, because the servers are now able to discern geographically popular items and keep them in memory for future local requests. However, we need a direct comparison to appreciate the different performance achieved by using these weights.

6.3 Policy Comparison

In order to understand which policy provides better results, we evaluate the relative performance improvements between the weighted policies and the other strategies.

We illustrate in Fig. 11 the performance increment when we augment the LRU strategy with geo-social information. Geosocial-LRU reaches a maximum 55% performance increment, while increasing the cache size results in a smaller increment. Instead, Geocascade-LRU achieves more than 70% increment on LRU for smaller cache sizes, while the benefit decreases as cache size increases. In Fig. 12 we investigate how the use of priority weights improves LFU. Geosocial-LFU achieves a top increment of about 50% against LRU with small cache sizes, with the increment going down as the size increases. However, the improvement is larger in the case of the Geocascade weight, with a maximum increment of 70% and a smaller decrement with cache size. Finally, in Fig. 13 we investigate the difference between the Geocascade and the Geosocial weight for the Mixed cache policy. Again, the Geosocial weight gives a maximum improvement of 50%, while the Geocascade one improves up to 65% the baseline performance.

Both weights improve cache performance, since they recognize content that is bound to become popular only locally and to result in many requests to the same local servers. Indeed, items that are popular on a global scale may see their requests across different servers around the planet and may not trigger cache prioritization in single CDN clusters. Furthermore, including information about the spreading of

social cascades appears as a better predictor of local popularity, since the Geocascade weight exhibits higher performance than the Geosocial one. It is also important to note that the performance improvement is smaller when the cache becomes larger. Indeed, with a cache so large that can host 0.1% of the entire data corpus, it becomes easier to accommodate more items and performance easily reaches a saturation point, as seen in Fig. 10. Nonetheless, for a given cache size larger workloads have a larger relative improvement, since their absolute performance are smaller.

7. IMPLICATIONS

The main result of this work is that locality information from social cascades can be extracted and used to improve large-scale system design. We see a great potential in exploiting geographic properties of human communication over online services. Geographic locality of online interactions can be exploited to do pre-fetching of Web content, caching of normal HTTP traffic, datacenter design and placement and even to devise security mechanisms [15, 2].

In addition, our approach can be generalized to be used on a number of different OSNs. The information needed can be efficiently exposed by an anonymized API, which could provide only the aggregated geo-social metrics corresponding to a given cascade of a certain shared item. Moreover, information coming through public Twitter feeds, private Facebook posts and emails can be anonymized and exposed in order to classify items according to their geographic popularity and feed this information into CDNs.

In the specific example we have discussed, improvement largely depends on cache size: when it is possible to cache a considerable portion of the whole item catalogue, cache policies matter less and the improvement obtained by social information is smaller. However, if cache size is not sufficient to store that portion, because it is too small with respect to item size or because the catalogue contains too many items, geo-social properties can make a difference. Moreover, if in the future social cascades can be tracked on a larger scale on OSNs, the advantage given by geo-social metrics may impact not only CDN caching policies but, more generally, other large-scale systems.

As already mentioned, our results are obtained using a sample from a single OSN. Although it is generally unknown which portion of the traffic directed to CDNs is coming from OSNs, it is not far from reality that this traffic may become considerable: the fraction of content links among messages in our dataset is already appreciable and the number of users on OSNs is still increasing.

An improvement in the number of cache hits for requests coming from these OSNs, as observed in our simulation, would imply that millions of video daily requests could be served locally instead of being transferred over the network. In addition, videos are getting larger, with higher quality demanded by users, meaning bulkier files. Caches need to grow larger and larger to cope with this trend. This is impacting (and will increasingly greatly impact) on the running costs of modern CDNs. For instance, Limelight runs a global private fiber-optic network that avoids sending files over the busy public Internet connections. As a result, any reduction on the number of files sent across the network would reduce the investments on network infrastructure, which account for a considerable part of a CDN total expenditure [13].

8. RELATED WORK

Two research areas are related to our work: the analysis of online social cascades and the design of large-scale CDNs.

Social Cascades. Social cascades have been studied in sociology, economics and marketing for more than 60 years: an eminent example is the threshold model proposed by Granovetter [8]. Recently, thanks to the availability of large datasets, many other studies have been presented. In [1] the authors analyze the diffusion of information in blogs by applying epidemic models of information spreading. Similarly, a characterization of cascades using data from Flickr, a photo-sharing website, is illustrated in [4].

Finding ways of harnessing the potential of information constantly generated by users of OSNs is a key and promising research area for the networking community and it is still largely unexplored. Our work is one of the first examples of how this information can be effectively used to improve the performance of large-scale networked systems, and, more specifically, of CDNs.

Content Distribution Networks. Given the success and economic importance of CDNs, many solutions to improve the performance of this class of systems have been proposed with respect to the location-aware selection of servers. Key examples of experimental systems in this area are Meridian [16], a node selection mechanism based on network locality, and OASIS [6], an overlay anycast service infrastructure. WhyHigh is a system to redirect queries based on the measurements of the latency of the Google's CDN [10]. This system is not only based on geographic proximity but also on measurements of client latencies across all CDN nodes, in order to identify the prefixes with inflated latencies.

While these systems have used some intelligence on geography or request load to improve the system, we have also taken advantage of information from OSNs interaction to enhance the content placement decision process.

9. CONCLUSIONS AND FUTURE WORK

We have presented how geo-social properties of users participating in OSN cascades can be exploited to improve the efficiency of caching in CDNs. We have studied cascades of a real OSN and have used our findings in a simulation-based validation of a model of a CDN. While our study is limited in scope by the choice of network and data set, our results are more generally applicable and the impact of the approach can be potentially high for large-scale systems whose traffic is driven by online social services.

Our research agenda includes the generalization of the proposed technique to deal with information from multiple OSNs and the investigation of mechanisms for improving the accuracy of the prediction of the size and temporal evolution of social cascades. Finally, we plan to investigate how these techniques can be effectively implemented within real CDNs.

Acknowledgments

The authors would like to thank Paolo Costa for many insightful discussions on this topic and the members of the NetOS research group of the Computer Laboratory at the University of Cambridge for their suggestions. The authors also thank the anonymous reviewers for their thoughtful comments.

10. REFERENCES

- [1] Eytan Adar and Lada A. Adamic. Tracking Information Epidemics in Blogspace. In *Proceedings of WI '05*, Washington, DC, USA, 2005. IEEE Computer Society.
- [2] Lars Backstrom, Eric Sun, and Cameron Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *Proceedings of WWW '10*, New York, NY, USA, 2010. ACM.
- [3] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's largest User Generated Content Video System. In *Proceedings of IMC '07*, pages 1–14, New York, NY, USA, 2007. ACM.
- [4] Meeyoung Cha, Alan Mislove, and Krishna P. Gummadi. A Measurement-driven Analysis of Information Propagation in the Flickr Social Network. In *Proceedings of WWW '09*, New York, NY, USA, 2009. ACM.
- [5] Ludmila Cherkasova. Improving WWW Proxies Performance with Greedy-Dual-Size-Frequency Caching Policy. Technical report, HP, 1998.
- [6] Michael J. Freedman, Karthik Lakshminarayanan, and David Mazières. OASIS: Anycast for any Service. In *Proceedings of NSDI '06*, Berkeley, CA, USA, 2006. USENIX.
- [7] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. Youtube Traffic Characterization: a View from the Edge. In *Proceedings of IMC '07*, pages 15–28, New York, NY, USA, 2007. ACM.
- [8] Mark Granovetter. Threshold Models of Collective Behavior. *American Journal of Sociology*, 83(6):1420–1443, 1987.
- [9] Cheng Huang, Angela Wang, Jin Li, and Keith W. Ross. Measuring and Evaluating Large-scale CDNs . In *Proceedings of IMC '08*, New York, NY, USA, 2008. ACM.
- [10] Rupa Krishnan, Harsha V. Madhyastha, Sridhar Srinivasan, Sushant Jain, Arvind Krishnamurthy, Thomas Anderson, and Jie Gao. Moving beyond End-to-end Path Information to Optimize CDN Performance. In *Proceedings of IMC '09*, New York, NY, USA, 2009. ACM.
- [11] Tom Leighton. Improving Performance on the Internet. *Communications of the ACM*, 52:44–51, February 2009.
- [12] David Liben-Nowell, Jasmine Novak, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Geographic routing in social networks. *PNAS*, 102(33):11623–11628, August 2005.
- [13] Asfandiyar Qureshi, Rick Weber, Hari Balakrishnan, John Guttag, and Bruce Maggs. Cutting the electric bill for internet-scale systems. In *Proceedings of SIGCOMM '09*, New York, NY, USA, 2009. ACM.
- [14] Salvatore Scellato, Cecilia Mascolo, Mirco Musolesi, and Vito Latora. Distance Matters: Geo-social Metrics for Online Social Networks. In *Proceedings of WOSN '10*, Boston, MA, USA, 2010. USENIX.
- [15] Mike P. Wittie, Veljko Pejovic, Lara Deek, Kevin C. Almeroth, and Ben Y. Zhao. Exploiting locality of interest in online social networks. In *Proceedings CoNEXT '10*, New York, NY, USA, 2010. ACM.
- [16] Bernard Wong, Aleksandrs Slivkins, and Emin Gün Sirer. Meridian: a Lightweight Network Location Service Without Virtual Coordinates. In *Proceedings of SIGCOMM '05*, Philadelphia, Pennsylvania, USA, 2005.
- [17] YouTube Blog. Share and share alike. <http://youtube-global.blogspot.com/2011/01/share-and-share-like-weve-acquired.html>.