Research Article

Adaptive Content Management for UGC Video Delivery in Mobile Internet Era

Qilin Fan,¹ Hao Yin,¹ Zexun Jiang,¹ Haojun Huang,² Yan Luo,³ and Xu Zhang¹

¹National Laboratory for Information Science and Technology (TNList), Department of Computer Science and Technology, Tsinghua University, China

²Department of Communication Engineering, Wuhan University, China

³Department of Electrical and Computer Engineering, University of Massachusetts Lowell, USA

Correspondence should be addressed to Qilin Fan; fql11@mails.tsinghua.edu.cn

Received 8 January 2016; Revised 18 April 2016; Accepted 21 April 2016

Academic Editor: Dik Lun Lee

Copyright © 2016 Qilin Fan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The demand of storing and transferring user generated content (UGC) has been rapidly growing with the popularization of mobile devices equipped with video recording and playback capabilities. As a typical application of software-defined networks/network functions virtualization-based pervasive communications infrastructure, content delivery networks (CDNs) have been widely leveraged to distribute contents across different geographical locations. Nevertheless, the content delivery for UGC is inefficient with the existing “pull-based” caching mechanism in traditional CDNs, because there exists a huge volume of lukewarm or cold UGC which results in a low cache hit ratio. In this paper, we propose a “push-based” caching mechanism to efficiently and economically deliver UGC videos. Different from traditional CDNs which separate the original content storage and caching, we directly store UGC videos into selective servers which serve as both reliable storages and user-facing uploading servers. By carefully and dynamically selecting the storage locations of each UGC object based on its popularity and locality, we not only guarantee the data availability but also remarkably improve the content distribution performance and reduce the distribution cost.

1. Introduction

Videos in video on demand (VOD) systems have historically been created and supplied by a limited number of media producers. The emergence of mobile devices such as high quality smart phones and tablets equipped with video recording capabilities has enabled the general public to record events, generate videos, and upload them to video-sharing sites such as YouTube. Nowadays, Internet users are not only content consumers, but also content publishers as well. Besides, users could access contents via mobile devices at any time and anywhere. Such advent of user generated content (UGC) in mobile Internet era has remarkably reshaped the online video industry.

As a typical application of software-defined networks/network functions virtualization-based pervasive communications infrastructure, content delivery networks (CDNs) have been playing a critical role in offering fast and reliable

communication services by distributing content to cache or edge servers located close to users. Today, video providers rely on overlay CDNs like Akamai, Limelight to leverage their presence across different geographical locations to serve video contents. However, the explosive video consumption paradigm shift in the mobile Internet environment has introduced new challenges in distributing UGC videos for CDNs.

(1) The conventional caching schemes utilized in traditional CDNs are ineffective with UGC. We crawl the request logs from Youku (<http://www.youku.com/>), the largest video-sharing website in China, to simulate the impact of UGC on traditional CDN and compare it with provider generated content (PGC). We sample 20,000 UGC videos and 2000 PGC videos in Youku. Figure 1 shows how the hit rate of UGC and PGC evolves with cache size using most common caching technique, least recently used (LRU). After cache size achieving 10% of the total video volume, the hit rate gain becomes

FIGURE 1: Hit rate as function of cache size.

FIGURE 2: Cost as function of cache proportion.

slower while expanding cache size in UGC. It suggests that the long tail of lukewarm videos in UGC exacerbates the efficacy of cache deployment.

(2) The passive content management does not apply to cost control in UGC. Supposing that the UGC and PGC video number is 2×10^8 and 2×10^5 , we calculate the total cost including storage and bandwidth as cache proportion increases according to current pricing norms [1] (<http://www.bizety.com/2014/08/24/cdn-storage-selling-feature/>). Figure 2 illustrates that caching 50% content in edge servers passively is sufficient for PGC to achieve low cost, while partial passive caching in traditional CDNs does not contribute to cost reduction in UGC. Meanwhile, the content volume is still increasing at a speed of 53% (<http://blog.performics.com/performics-weekly-digital-digest-5-23-13/>) per year, faster than the decreasing speed 28% (<http://www.dostor.com/article/2013-08-29/3649023.shtml>) of storage cost, thus leading to ever-increasing cost.

(3) The usefulness of the most widely used DNS name resolution approach based on URL in traditional CDNs is declining for massive content. Authoritative DNS maps the URL to an IP address in TTL period. However, the explosive growth of the UGC namespace has decreased the effectiveness of DNS caching. Further, the timeout-based invalidation of stale mapping cannot guarantee cache coherency [2]. Nowadays, many UGC video portals build their own naming system. However, as far as we know, the study has stayed in measurement and analysis [3].

In this paper, we firstly propose a general framework for UGC video delivery, namely, adaptive content management- (ACM-) based CDN. By introducing proactive content management, joint content replication, and request routing design principles into system design, it could achieve high scalability, flexibility, and performance goals and reduce cost as well.

Second, after conducting extensive measurement on Youku, we analyze temporal popularity evolution and geographic location distribution for UGC videos. Based on

the popularity predictability and geographic locality characteristics, we present data-driven content replication and request routing algorithms so that videos are replicated at “cost-effective” locations and server selection is content-aware.

In order to evaluate our system, we build an experiment platform with realistic UGC traces. Our trace-driven simulation clearly demonstrates the quantitative benefits of our ACM-based CDN. In particular, our ACM-based CDN reduces latency by 14.7%, network load by 63%, and server load by 50% at 95th percentile.

2. Framework Design

In this section, we begin with an overview of our goals and principles for guiding the design of framework for UGC video delivery. Then we describe the key components to satisfy the design goals and philosophies.

2.1. Design Goals. To design UGC video delivery framework, we first illustrate four significant design goals.

(1) *High Scalability.* At the fundamental level, scalability for UGC video means handling more clients, content, and traffic (e.g., over 1 billion unique users have visited YouTube each month, over 100 hours of video has been uploaded every minute, and over 6 billion hours of video has been watched each month on YouTube (<https://www.youtube.com/yt/press/statistics.html>)). This also means that a name resolution system should support ever-growing number of content and distributed servers.

(2) *Flexibility.* The video popularity and geographical access change dynamically as time goes by under UGC environment [4]. This requires the system to support changes in name-address mapping to rapidly propagate new mappings to users.

(3) *High Performance*. The multimedia streaming service requires higher QoS, such as lower startup delay, lower transmission delay, and higher continuity. Any degradation in any of these factors may impact users' experience. Balachandran et al. [5] observed that an increase of the buffering ratio of only 1% can lead to more than three minutes of reduction in the user engagement.

(4) *Controllable Cost*. The massive number of user generated videos and visits consume a huge volume of storage and network resource (e.g., Tudou (<http://www.tudou.com/>) consumed 1 PB bandwidth each day for transferring videos in 2012). The system should be carefully designed to reduce unnecessary resource consumption.

2.2. Design Principles. Then, we introduce two design principles into framework design to satisfy the above requirements. The first principle is to *bring proactive self-adapting content management into content distribution*. Content management is strategically vital to a CDN for efficient content delivery and for overall performance. The UGC video distribution and propagation in mobile Internet environment have brought new features with respect to highly dynamic access, flattening popularity distribution [4], and marginalized content delivery. Therefore, it is necessary to develop a proactive self-adapting content management mechanism into UGC content delivery and guarantee favourable users' experience.

The second principle is to *merge content replication and request routing together*. A CDN must decide on how and where to replicate the content in an intelligent way, referred to as content replication problem. Also, it is challenging for a CDN to select the best server to respond to the user, known as request routing problem. Both problems are interdependent and thus should be considered together to operate in an efficient manner.

2.3. Architecture. According to the above design goals and philosophies, we propose a general architecture for UGC video delivery. As shown in Figure 3, it has three components.

(1) *Video ID Space*. Each video is uniquely identified by a "flat" name instead of the Internet's current host-centric naming for the reason that DNS overloads the names and rigidly associates them with specific network locations, making it inconvenient to migrate data. Flat namespace's scalability issue could be solved by distributed hashing table (DHT).

(2) *Name Resolution System*. It comprises the log collection, machine learning, mapping, and performance monitoring modules. Log collection module keeps track of users' access behaviour. Machine learning module analyzes collected data and predicts video access characteristics with respect to video's popularity evolution and geographic distribution to guide the content replication. Performance monitoring module gathers information about performance of network and servers and maintains an up-to-date view of network resources. The mapping module establishes video ID-to-IP address index and provides content-aware request routing service.

FIGURE 3: Architecture for UGC video delivery.

(3) *Distributed-Tiered Hybrid Storage System*. Traditional CDN consists of a centralized storage data center which stores all the content and multiple delivery servers responsible for handling users requests. However, it is challenging for a centralized data center to store massive and rapidly growing content. Different from traditional CDN which separates the original content storage and caching, we directly store UGC into selective servers which serve as both reliable storages and user-facing uploading servers. We adopt distributed storage to manage video collection and guarantee dataset integrity. While videos with different popularity have varying access characteristics, they should not be considered collectively. We sort videos into three logical layers according to their popularity: hot, medium, and cold. Each layer determines how many replicas are kept on servers, where to place these replicas, and how often to update them.

The name resolution space combined with flat video ID space achieves qualitative goals. (1) *Scalability*. In our design, mapping module could provide IP addresses in content-granularity to users directly. This solves the inefficient caching and slow update problems [2] existing in traditional DNS resolution mechanism. Further, due to loose coupling between mapping module and physical servers, it is straightforward to expand service capability of mapping servers or storage servers individually, without mutual interference. (2) *Flexibility*. Data becomes the first-class entity; it can be freely migrated or replicated across hosts and administrative boundaries. We can easily decouple content replication and request routing from sticky dependency via name resolution

space and flexibly customize policies separately. Further, mobility and multihoming can be elegantly supported.

The name resolution space combined with hybrid storage system achieves quantitative goals. (1) *High Performance*. Using data-driven technology to guide the videos replicating at most appropriate locations, users could quickly find a nearby replica. The periodical push-based approach reduces frequent video fetch and replacement, thus avoiding network congestion. Further, each server only deals with a small set of videos, refraining from overloading a server. (2) *Controllable Cost*. Our design could save storage cost as videos are replicated on demand and change as time goes by. We could even find a tradeoff between the storage cost and performance in our future work. Meanwhile, the content-aware request routing can eliminate the bandwidth waste due to frequent content migration in a conventional network-aware request routing.

3. Key Algorithms

In this section, we first investigate the popularity and geographical distribution of UGC and then explore how these characteristics can be used for guiding content replication and request routing, two important algorithms in the system design.

3.1. UGC Measurement and Analysis. In our measurement, we have crawled our dataset from Youku during the first two weeks of August 2015, using snowball sampling with initial set consisting of 10 random videos. We processed our collected datasets to remove (1) videos with missing or inconsistent information and (2) non-UGC videos according to category. The total number of samples was 200,000. For each video, we collected the following attributes: (1) its total number of views; (2) its views per day over time since it is uploaded; (3) its geographic distribution which represents how many views it received from each province; and (4) its top ten list of cities with the most traffic.

3.1.1. Temporal Popularity. Szabo and Huberman [6] first observed that the log-transformed popularity exhibits strong correlations between early and later periods. In this paper, we use Multivariate Linear (ML) model [7] to predict the popularity $\hat{x}_t(v)$ of a video v on target day t . Given the number of views $x_i(v)$ on each day before the target day t , we can define the feature vector $X_{t,n}(v)$ of video v as

$$X_{t,n}(v) = (x_{t-1}(v), x_{t-2}(v), \dots, x_{t-n}(v))^T. \quad (1)$$

Then we can estimate the number of views video v can get on target day t as

$$\hat{x}_t(v) = \Theta_n \cdot X_{t,n}(v), \quad (2)$$

where $\Theta_n = (\theta_{t-1}, \theta_{t-2}, \dots, \theta_{t-n})$ is the vector of model parameters. Intuitively, each parameter value represents the importance of each day for estimating views on target day. To train model parameters, we use mean Relative Square Error

FIGURE 4: mRSE as various previous n days.

(mRSE) as cost function on training videos set V . We define cost function $J(\Theta_n)$ as follows:

$$J(\Theta_n) = \frac{1}{|V|} \sum_{v \in V} \left(\frac{\hat{x}_t(v)}{x_t(v)} - 1 \right)^2, \quad (3)$$

where $x_t(v)$ is the actual number of views of v on target day t .

The global optimal solution is to find the best parameter vector which minimizes the cost function:

$$\begin{aligned} \Theta_n^* &= \arg \min_{\Theta_n} \frac{1}{|V|} \sum_{v \in V} \left(\frac{\hat{x}_{t,v}}{x_{t,v}} - 1 \right)^2, \\ &= \arg \min_{\Theta_n} \frac{1}{|V|} \sum_{v \in V} \left(\Theta_n \cdot \frac{X_{t,n}(v)}{x_t(v)} - 1 \right)^2. \end{aligned} \quad (4)$$

The optimization problem can be solved by gradient descent algorithm which starts with some initial Θ_n and repeatedly performs the update:

$$\Theta_{n,j} = \Theta_{n,j} - \alpha \frac{\partial}{\partial \Theta_{n,j}} J(\Theta_n), \quad (5)$$

where α is the learning rate and $\Theta_{n,j}$ is the j th weight of Θ_n .

To validate our model training process, we randomly extract three categories: music, game, and entertainment. For each category, along with all videos, we use 10-fold cross validation to calculate predicted popularity. During training process, given different number n , we can get different model parameter vector Θ_n , and then we use Θ_n^* to estimate $\hat{x}_t(v)$ on validation data based on (2). Figure 4 shows the correlation between mRSE and latest n days for different categories. We can draw two conclusions that are helpful for guiding content replication from temporal perspective.

- (1) The prediction of popularity on target day based on historical data is highly accurate. In fact, we only need views data on latest three days, yielding 18% to 25% prediction error to predict video's popularity.

TABLE 1: Popularity category.

Category	# of views	% of videos	Average views
C1	$[0, 10^4)$	45.0	2792
C2	$[10^4, 10^5)$	35.1	38710
C3	$[10^5, 10^6)$	16.8	281105
C4	$[10^6, \infty)$	3.1	2427914

FIGURE 5: Geographic cumulative distribution of views.

- (2) Prediction on subsamples of the dataset extracted by video category reduces 5% mRSE on average compared to the whole dataset. Prediction based on category is more accurate.

3.1.2. Geographic Location. We examine the geographic distribution of views for UGC videos on province granularity and city granularity. We divide our dataset into four categories according to view numbers (Table 1).

Province Granularity. For each video, geographic views from each province are sorted in decreasing order. We then compute the cumulative distribution of views of each video and plot the average over each popularity category. Figure 5 shows that 30% of provinces could cover 70% of video views for all categories.

City Granularity. Due to the limitation of dataset, we use linear fitting approach for ten most traffic cities in log-log coordinate to establish geographic distribution on city granularity (Figure 6). The distribution function can be expressed as follows:

$$\log y_n = k \log n + b, \quad (6)$$

where n is the city traffic rank and y_n is the corresponding views. From Figure 6, we observe that cities distribution satisfies power law profile. China has 287 cities. Top 10 cities (3.5% of cities) could cover 30% of all traffic. Given (6), we can

FIGURE 6: Views distribution of top 10 cities.

calculate that 60 cities (20% of all cities) could hold 70% of the overall traffic.

The above measurement illustrates that locality of geographic access is universal, whether from coarse or fine granularity. It is feasible to direct content replicating in small-scale locations to achieve most of the traffic.

3.2. Content Replication

Initial Replication. When video is first uploaded by a user, it will be stored by a server which is closest to uploader in cold level.

Replication Update. Assume time advances in time slots. First, we estimate the number of views $\hat{x}_t(v)$ video v may achieve at time t using (2). The thresholds of medium level and hot level are denoted as T_M and T_H . Only if $\hat{x}_t(v)$ exceeds the corresponding threshold, can the video migrate to higher level. The cold level is responsible for permanent storage. The replication location sets for cold, medium, and hot level are denoted as L_C , L_M , and L_H . We define $\lambda_{l,t}(v)$ as the views of video v generated from location l at time slot t . α_M and α_H represent geographic access locality in medium level and hot level. For example, if China's provinces ($|L_H| = 34$) reach the hot level, then α_H can be set 70%/30%, denoting 70% of the views divided by corresponding location numbers ($34 \times 30\%$) as the geographic propagation threshold. This threshold guides video replicating at desirable locations and these parameters can be tuned for balancing between user performance and storage cost. Our content replication algorithm is illustrated in Algorithm 1.

3.3. Request Routing. We use an abstract function $\text{QoS}(u, r)$ to quantify the quality of service between user u and replica r . The QoS metric can be related to many factors such as latency, network congestion, and server load. This provides content provider the flexibility to define its own QoS metric.

```

(1) if  $v$  is newly updated then
(2)   replicate  $v$  at closest location  $l \in L_C$ 
(3) else
(4)   estimate  $\hat{x}_t(v)$ 
(5)   if  $\hat{x}_t(v) < T_M$  then
(6)     if  $v$  is in medium or hot level at time  $t - 1$  then
(7)       delete  $v$  from medium or hot level
(8)     end if
(9)   else if  $T_M \leq \hat{x}_t(v) < T_H$  then
(10)    for  $l \in L_M$  do
(11)      if  $\lambda_{l,t-1}(v) \geq x_{t-1}(v) \times \alpha_M / |L_M|$  then
(12)        replicate  $v$  at location  $l$ 
(13)      end if
(14)    end for
(15)    delete extra replicas of  $v$  in medium or hot level
(16)  else
(17)    for  $l \in L_H$  do
(18)      if  $\lambda_{l,t-1}(v) \geq x_{t-1}(v) \times \alpha_H / |L_H|$  then
(19)        replicate  $v$  at location  $l$ 
(20)      end if
(21)    end for
(22)    delete extra replicas of  $v$  in medium or hot level
(23)  end if
(24) end if

```

ALGORITHM 1: Content replication algorithm.

Further, multihoming is naturally supported in our content-aware request routing mechanism. Many specific server selection or scheduling algorithms [1, 8] based on multihoming can also be applied in our video delivery framework. For intuitive comparison with traditional CDNs, we adopt a simple server selection mechanism: choose one replica r^* with best QoS serving the user u by using criterion $r^* \leftarrow \operatorname{argmax}_{r \in R_v} \text{QoS}(u, r)$, where R_v is the replica list for video v .

4. Simulation and Evaluation

In this section, we mark our proposed framework as ACM-based CDN. We collected 5,000 pieces of real request data for a week from September 1 to September 7 in 2015 from Youku portal. We conduct the experiment based on the data to analyze ACM-based CDN's performance from the following perspectives: (1) latency; (2) network congestion; (3) server load.

4.1. Experiment Setup. Our experiment is conducted using an event-based simulator implemented in Java. Figure 7 shows our experiment map with locators indicating the replication locations. The locations in medium level or cold level are chosen by k -means clustering. To keep the experiment simple and generic, we select server according to the geographical distance. More QoS definitions of server selection will be planned in our future work. We set the time slot to 1 day due to the granularity of access data we could achieve. According to the result illustrated in Figure 4, we set $n = 4$ which is accurate enough for videos which are uploaded more than 4 days. That is, we only need to persist latest 4 day's historical access

FIGURE 7: Experiment map of ACM-based CDN for UGC video delivery.

data for prediction. In order to reduce the perturbations on system performance in new time slot, videos will be replicated incrementally during idle time. Vectors of model parameters Θ_1 to Θ_4 for each category and replication strategy are calculated offline. The simulator uses extra previous 4 days of the logs to build request histories and does not report the performance of the caching algorithm during those days. This is because our algorithm needs the data of previous days to replicate the initial videos in the cache. Table 2 summarizes more detailed experiment parameters.

TABLE 2: Experiment parameters.

Parameter	Definition	Value
T	Duration of time	7
N_{old}	Initial number of old views	5000
N_{new}	Number of videos updated per time slot	10
T_M	Threshold for medium level	50
T_H	Threshold for hot level	500
L_C	Number of replication locations in cold level	4
L_M	Number of replication locations in medium level	12
L_H	Number of replication locations in hot level	34
α_M	Geographic locality indicator in medium level	2
α_H	Geographic locality indicator in hot level	2

For comparison, we take traditional content delivery network as the baseline. We assume that it is composed of origin server containing all the videos which is located in the map's center, L_M parent servers, and L_H edge servers which are in the same locations illustrated in Figure 7. The origin server, parent servers, and edge servers form a tree-shaped network. A request arrives at the closest edge server and is routed along the closest parent server towards the origin server until it finds the server with the requested video. A traditional CDN adopts LRU cache replacement policy when request misses.

In order to compare fairly, we try to allocate the same storage capacity for different CDNs. We first compute the average cache percentage consumed in hot and medium level in our framework. Depending on the result, 7%, the cache percentage for parent servers and edge servers in traditional CDN is set 10%, considering that our framework requires extra name resolution system.

4.2. Results. For comparison, we conduct experiments of two CDN systems based on the same real request data. The experiment shows that our ACM-based CDN outperforms traditional CDN for each performance metric.

Latency. We first present the latency which indicates the transmission time between the request and the location from which it was served. In our experiment, we assume the latency is proportional to geographical distance which is expressed as tenfold geographic distance between the user and selected server in longitude and latitude coordinates. Figure 8(a) shows the latency gap between our ACM-based CDN and traditional CDN is 19 ms at 90th percentile and 27 ms at 95th percentile, that is, 13.2% and 14.7% improvement, respectively. The reason is that, by inferring videos access temporal pattern and geographical locality, better prediction of videos access times can be utilized to select closer replication positions to client, especially for a majority of lukewarm videos compared to traditional “pull-based” caching approach.

Network Load. Next, we investigate the network congestion under two different CDNs. The network load is calculated as the number of videos transferring over the links. Figure 8(b)

shows that the network load in our ACM-based CDN is only 37% of that in traditional CDN at 95th percentile.

Server Load. Finally, we study the load on the servers in Figure 8(c). The metrics is the requests served by the servers. We normalize the total server load in our ACM-based CDN. Servers for each level in the traditional CDN bear more load than servers in our ACM-based CDN: 90.4%, 37.1%, and 253.8%, respectively. On the condition that the storage capacity is almost equal, the total server load in the traditional CDN doubles that in the ACM-based CDN. The comparison experiment of network and server load illustrates that our content-aware request routing improves the efficiency to find an appropriate server for serving client, instead of wasting bandwidth on connecting to the upper server to find and transfer the content.

From the result of experiment, the “push-based” adaptive content replication algorithm, together with content-aware request routing mechanism, supports faster video delivery and imposes less traffic burden in network level. Furthermore, CDNs pay for bandwidth based on how many bits exit their servers which can be reflected by our server load experiment. Therefore, our framework together with our algorithm could distribute UGC videos efficiently and economically as well. But it is necessary to point out that compared to traditional CDN which utilizes DNS for request routing, our framework builds its own name resolution system to implement refined content management. As video scale grows, we will partition the mappings into different index servers using consistent hashing. To look for the accurate index server for a particular video ID, each index server needs to be cooperative with the previous and next index servers to establish route. Our ACM-based CDN will take a longer time to find the video ID-to-IP address mapping but achieving higher performance and lower cost in video transferring phase is worth the sacrifice.

5. Related Work

Caching mechanisms exploit storage capacity to absorb traffic by replicating content closer to the network edge rather than storing it in a central location which requires high processing power. Most caching schemes utilized in wide-area, distributed systems are initiated by clients (pull-based). The problem of pull-based caching and eviction has received many research efforts. For example, [9, 10] focused on online eviction algorithms (LRU, FIFO, and LFU) and their variants such as greedy and randomized versions. The drawback of these pull-based approaches is that an optimal server is not always chosen to serve content request.

To further improve the Web performance, several works [11, 12] proposed push-based caching as a complementary technique. It is formulated as an optimization problem under a given traffic pattern and a set of resource constraints.

Many studies have examined the characteristics of user generated videos. Cha et al. [4] observed the skewed distribution with long tails. Huguenin et al. [13] showed the correlation between the content locality and geographical

FIGURE 8: Cost and performance comparison.

locality. Cha et al. [14] observed the correlation between a video's history and its future demand.

In this paper, we utilize these characteristics to derive models for prediction and guide intelligent “push-based” caching, rather than assuming idealistic traffic pattern [11, 12]. In addition, compared to the unique solution given by previous works [11, 12], server capacity in our framework can be scaled up and down by adjusting the thresholds in hot and medium level. It provides much more flexibility in cost management.

6. Conclusions

We address in this paper the challenges in distributing UGC videos, resulting from the gap between the new features of

UGC in mobile Internet environment and rigid architecture of traditional CDN. We propose a new framework for UGC video delivery that takes proactive content management, joint content replication, and request routing into consideration. Based on the UGC trace analysis for temporal predictability and geographic locality, we present a data-driven content replication algorithm, which distributes content into “cost-effective” locations, and corresponding content-aware request routing algorithm. Extensive experiments driven by the real-world traces demonstrate the high performance and low cost of our design.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported in part by the National Basic Research Program of China (973 Program) under Grant no. 2012CB315801, in part by the independent research project of Tsinghua University under Grant no. 20131089304, in part by the projects of Tsinghua National Laboratory for Information Science and Technology (TNList), in part by the European Seventh Framework Programme (FP7) under Grant no. PIRSES-GA-2012-318939, in part by the National Natural Science Foundation of China under Grant no. 61402343, and in part by Jiangsu International Cooperation Program of Science and Technology under Grant no. BZ2013018.

References

- [1] H. H. Liu, Y. Wang, Y. R. Yang, H. Wang, and C. Tian, "Optimizing cost and performance for content multihoming," in *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '12)*, pp. 371–382, Helsinki, Finland, August 2012.
- [2] V. Ramasubramanian and E. G. Sirer, "The design and implementation of a next generation name service for the internet," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 331–342, 2004.
- [3] V. K. Adhikari, S. Jain, Y. Chen, and Z. L. Zhang, "Vivisectioning youtube: an active measurement study," in *Proceedings of the IEEE Computer and Communications Societies, IEEE Annual Joint Conference (INFOCOM '12)*, pp. 2521–2525, Orlando, Fla, USA, March 2012.
- [4] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Transactions on Networking*, vol. 17, no. 5, pp. 1357–1370, 2009.
- [5] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "Developing a predictive model of quality of experience for internet video," in *Proceedings of the ACM Conference on SIGCOMM*, pp. 339–350, ACM, 2013.
- [6] G. Szabo and B. A. Huberman, "Predicting the popularity of online content," *Communications of the ACM*, vol. 53, no. 8, pp. 80–88, 2010.
- [7] H. Pinto, J. M. Almeida, and M. A. Gonçalves, "Using early view patterns to predict the popularity of YouTube videos," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM '13)*, pp. 365–374, Rome, Italy, February 2013.
- [8] P. Wendell, J. W. Jiang, M. J. Freedman, and J. Rexford, "Donar: decentralized server selection for cloud services," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 231–242, 2010.
- [9] P. Cao and S. Irani, "Cost-aware www proxy caching algorithms," in *Proceedings of the Usenix Symposium on Internet Technologies and Systems*, vol. 12, pp. 193–206, 1997.
- [10] K. Psounis and B. Prabhakar, "Efficient randomized web-cache replacement schemes using samples from past eviction times," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 441–454, 2002.
- [11] J. Kangasharju, J. Roberts, and K. W. Ross, "Object replication strategies in content distribution networks," *Computer Communications*, vol. 25, no. 4, pp. 376–383, 2002.
- [12] T. Bektaş, J.-F. Cordeau, E. Erkut, and G. Laporte, "Exact algorithms for the joint object placement and request routing problem in content distribution networks," *Computers & Operations Research*, vol. 35, no. 12, pp. 3860–3884, 2008.
- [13] K. Huguenin, A. Kermarrec, K. Kloudas, and F. Taïani, "Content and geographical locality in user-generated content sharing systems," in *Proceedings of the 22nd International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '12)*, pp. 77–82, Toronto, Canada, June 2012.
- [14] M. Cha, H. Kwak, P. Rodriguez, Y. Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC '12)*, pp. 1–14, San Diego, Calif, USA, October 2007.