COMMUNITY BASED EFFECTIVE SOCIAL VIDEO CONTENTS PLACEMENT IN CLOUD CENTRIC CDN NETWORK

Han Hu[1], *Yonggang Wen*[2], *Tat-Seng Chua*[1], *Zhi Wang*[3], *Jian Huang*[4], *Wenwu Zhu*[3], *and Di Wu*[4]

[1] National University of Singapore, Singapore, Singapore 117417
[2] Nanyang Technological University, Singapore 639798
[3] Tsinghua University, Beijing, China
[4] Sun Yat-Sen University, Guangzhou, China
{huh, chuats}@comp.nus.edu.sg, ygwen@ntu.edu.sg, {wangzhi@sz., wwzhu@}tsinghua.edu.cn
{huangj77@mail2.,wudi27@}sysu.edu.cn

ABSTRACT

The increasing popularity of online social networks (OSNs) has been transforming the dissemination pattern of social video contents. Considering the unique features of social videos, e.g., huge volume, long-tailed, and short length, how to utilize the information propagation pattern to improve the efficiency of content distribution for social videos attracts more and more attention. In this paper, we first conduct a large scale measurement to explore the social video viewing behavior under the community classification. Based on the measurement, we investigate the community driven sharing video distribution problem under the cloud-centric content delivery network (CDN) architecture. In particular, we formulate it as a constrained optimization problem with the objective to minimize the operational cost. The constraint is the averaged transmission delay. Following that, we propose a dynamic algorithm to seek the optimal solution. Our trace-driven experiments further demonstrate our algorithm can make a better tradeoff between monetary cost and QoS, and outperforms the traditional method with less operational cost while satisfying the QoS requirement.

***Index Terms*—** Community Detection, Social Video, Cloud CDN Network

1. INTRODUCTION

With the growing popularity of OSNs and online video service, more and more users are accustomed to watching video (e.g., user generated content (UGC)), and then posting tweets to deliver their comments. An inevitable trend is that OSN and online video service are influencing each other mutually and slowly fusing. As reported by ForeSee [1], 18% users are influenced by the social network when accessing video contents. This fusion brings new video consuming patterns, compared with the traditional mode in VoD service where recommendation plays an important role to affect the choice of video watching. It is fascinating to utilize the unique features of video related information in OSNs to improve the performance of multimedia communication systems.

To design an effective UGCs distribution system, we should consider the inherent characteristics of UGCs, including huge volume, long-tail, and close-to-uniform and highly volatile popularity profile. Moreover, these UGCs are propagated in OSNs, bringing along new dissemination behaviors: 1) Video information spreads along the social connection, especially among closely connected friends; 2) Most of a user's friends are in the same region, hence, videos are propagated in a small clique with geo-locality; 3) Users with similar interest can be grouped into communities, and similar videos are more likely to be shared among these communities.

These characteristics posit significant challenges to the traditional CDN architecture and content placement strategies. Mislove et al. [2] observed that a large deduction of cache hit ratio when traditional caching schemes are used to replicate social contents. In the last decades, client assisted architecture, e.g., peer-to-peer (p2p), has been advocated for live video streaming [3]. Wang et al. [4] investigated the video prorogation behavior, and proposed a hybrid edge-cloud and peer-assisted video replication framework. In spite of some successful implementation of p2p systems, users are still reluctant to embrace the client assisted mechanism. The reasons include privacy, copyright protection, the need for clients to install a specific software, and the difficulty of version control. Recently, the emergence of cloud CDN [5] sheds new lights into this field. Based on a cloud computing infrastructure that provides scalable and on-demand resource allocation, Cloud CDNs are able to provide cost efficient content distribution. However, these strategies still face similar problems to deliver UGCs with QoE satisfaction.

In this paper, we propose a community based sharing

This work is supported by Singapore National Research Foundation under its International Research Center at Singapore Funding Initiative and administered by the IDM Programme Office.

video placement framework using the cloud CDN infrastructure. First, we incorporate geo-location and sharing video watching history into the traditional social network representation, and classify users into different video communities. Second, we conduct a large scale measurement to explore the sharing video watching behavior under the community classification. Based on the measurement, we investigate the community driven sharing video distribution problem. In particular, we formulate it as a constrained optimization problem with the objective to minimize the operational cost. The constraint is the average transmission delay. Leveraging the stochastic optimization framework, we derive a dynamic algorithm to find the optimal solution.

The rest of the paper is structured as follows. Section 2 presents a measurement study on social video viewing behavior. Section 3 introduces the community based video distribution architecture and models the system to a constrained optimization problem. Section 4 solves the problem based on the Lynapunov framework. The evaluation results are detailed in Section 5 and Section 6 concludes the paper.

2. COMMUNITY BASED MEASUREMENT

In this section, we first introduce our dataset and community classification method, and then we investigate the community characteristics over the dataset.

2.1. Data Set and Community Division

Using Sina Weibo API, we retrieve and crawl 10K users' video tweets posted during Jun. 1, 2012 to Jun. 15, 2012. By removing duplicated video tweets with the same video links and unavailable video tweets, we obtain 57, 445 tweets, which correspond to 2,302 unique video links. Then we collect video-related information, including video size, view count, from their corresponding video-sharing web sites.

We study a crowd of tweet users from multiple geographical regions $\mathcal{Z}$ that share a collection of videos $\mathcal{V}$. For each user u, his location is $\mathbb{Z}(u)$, and the watching history is a subset of $\mathcal{V}$, represented by $\mathbb{V}(u)$. We can employ a weighted graph $\mathcal{G}(U, E)$ to describe tweet users and their connections. U is the vertex set representing tweet users, and $E = \{e_{uu'}\}$ is the weighted edge. Each link $e_{uu'}$ represents a connection between two users u and u', and can be expressed as:

$$\begin{aligned} e_{uu'} = & \omega_1 \times fri(u, u') + \omega_2 \times dis(u, u') \\ & + \omega_3 \times sim(\mathbb{V}(u), \mathbb{V}(u')), \end{aligned} \tag{1}$$

The weighted link is the sum of three items, corresponding to social relationship, geo-distance, and preference similarity successively. $fri(u, u')$ is an indicator function with binary value (i.e., 0 or 1) to judge whether u and u' are friends. $dis(u, u')$ denotes their geo-distance. $sim(\mathbb{V}(u), \mathbb{V}(u'))$ represents the preference similarity (cosine similarity measurement is employed in this work). ω_1, ω_2, and ω_3 are weight factors to adjust the ratio of three items in weighted link. Higher weight indicates the corresponding item has higher effect on the community classification. In this paper, geo-close tweet users with similar preference are more likely to be scheduled to the same content node to save storage cost. Thereby, we prefer to endow higher priority to the geo-location (i.e., the second item) and preference similarity (i.e., the third item). Based on the weighted graph model, we utilize the affinity propagation algorithm [6] to classify users into a collection of communities, denoted by $\mathcal{C}$.

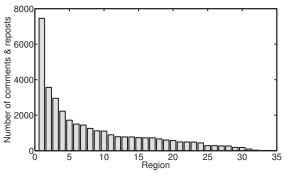


Fig. 1. Number of video tweets over different regions

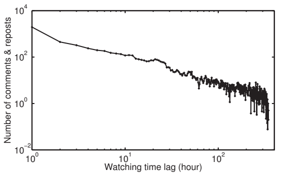


Fig. 2. Number of comments and reposts versus the time lag

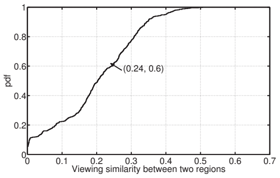


Fig. 3. Viewing similarity between different regions

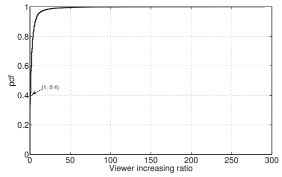
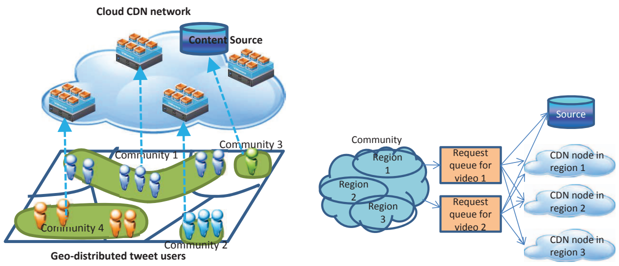


Fig. 4. Community based viewer expansion ratio

2.2. Community based Video Popularity Measurement

1) *Geographical & Temporal Diversity:* Fig. 1 illustrates the geographical distribution of all the video tweets. Guangdong, Beijing, and Shanghai are the top three regions with the largest viewer population. The geographical distribution of the viewer population is skewed, which is similar to the video popularity. Hence, we should consider the geographical imbalance in the content distribution system. In this dataset, we also observe that users are more likely to repost new video contents, as depicted in Fig. 2. Most of the reposts happen in the recent hours. It indicates that the traditional popularity based algorithms may suffer lower cache hit ratio.
2) *Community Based Viewer Expansion:* Fig. 3 demonstrates the viewing similarity between two regions. We can see that the similarity is less than 24% in the 60% cases. In contrast, when we group users into different communities as described in the previous section, viewers in the same community may access the same CDN node. Under this mechanism, the viewer number will be doubled in most cases, as shown in Fig. 4. This observation motivates us to design the subsequent com-



(a) Community based system architecture

(b) Request distribution

Fig. 5. Community based social video distribution

munity based cloud CDN architecture.

3. SYSTEM ARCHITECTURE & PROBLEM FORMULATION

This section first presents the architecture of the community based social video distribution scheme. Then we give a mathematical formulation to the devised architecture.

3.1. System Architecture

We employ a community based video sharing architecture on top of the cloud CDN infrastructure, as shown in Fig. 5(a), to provide efficient UGCs delivery. The cloud CDN infrastructure carves out storage, bandwidth and computation resources from data centers to provide content caching and media streaming services. Due to the storage limit, each CDN node can only store parts of video set from content sources. Users are grouped into geo-distribution communities, corresponding to a set of CDN nodes. When a user requests a video, the system will retrieval whether this content is stored in CDN nodes related to his community. When there is a copy, this user will download the video from one of these CDN nodes; otherwise, he should access the content source with higher delay.

The system cost highly depends on the content placement and replacement policy, which decides the number of replicas for contents and their locations. Specifically, on one hand, replicating content to different places can reduce the distance between users and contents, which leads to less bandwidth resources usages; on the other hand, too many copies incur significant storage resource and limit the reduction on bandwidth resource usage.

3.2. Problem Formulation

3.2.1. Cloud Centric CDN Model

Without loss of generality, for each region $z \in \mathcal{Z}$, we assume there is a corresponding CDN node DC_z, building upon the underlying data center topology. Each CDN node provides storage and computation resources. All those resources are limited and monetary sensitive. Since users can access social video either from CDN nodes or the content source, the dispatching policy depends on the access delay. Let $d_{zz'}$ denote the round-trip delay between region z and $z' \in \mathcal{Z}$. reflecting the geographic distance between two regions. For simplicity, d_{z0} represents the round-trip delay between region z and the content source.

We consider the following service charging model. The charge of storage in CDN nodes DC_z is $pr_{1,z}$ per byte. The cost to stream a byte from data center DC_z is $pr_{2,z}$. The unit cost to copy video contents to CDN nodes and stream videos to end users from the content source are $pr_{1,0}$ and $pr_{2,0}$ respectively. Storage in the content source and removal of videos from a CDN node are cost free. These charges follow the typical charging models of leading commercial cloud providers, such as Amazon EC2 [7] and S3 [8].

3.2.2. Community Based Request Model

We utilize the method in the previous section to group users into different communities. For a community c, users may come from several regions, denoted by $\mathbb{Z}(c)$. In this research, we adopt a discrete time slot model, in which the time horizon is discredited into time slots $t = \{0, 1, 2, ..., T\}$, where T is the window size of interest. Each time slot is long enough for replicating video contents to CDN nodes. In time slot t, let $a_c^{(v,z)}(t)$ represent the number of requests for video v with size b_v from users in region z, $z \in \mathbb{Z}(c)$. We assume that the request generation is an arbitrary process over time, with A_{max} being the maximum number of requests arising from each community for a video in each time slot.

3.2.3. Content Placement & Request Distribution

Within our architecture, user requests can be dispatched to the corresponding CDN nodes or the content source, as depicted in Fig. 5(b). A *control center* is responsible for collecting user requests, buffering them in request queues, and then dispatching them. It also decides whether a video is to be replicated or removed from a CDN node. Let $Q_c^{(v)}$ denote the request queue caching requests for video v from community $c, \forall c \in \mathcal{C}, \forall v \in \mathcal{V}$, whose length (i.e., the queue backlog) at time slot t is denoted by $Q_c^{(v)}(t)$.

The decisions that the control center needs to make in each time slot t of the dynamic system include: (1) Whether video v should be stored in the CDN node DC_z in time slot t or not, as indicated by control decision variable $s_{DC_z}^{(v)}(t)$ (1 for 'yes' and 0 for 'no'), $\forall z \in \mathcal{Z}, v \in \mathcal{V}$. (2) How many requests for video v from the region z in community c should be dispatched to the content source and how many to each CDN node, denoted by $\delta_c^{(v,z)}(t)$ and $\gamma_{c,z'}^{(v,z)}(t)$, respectively. It should be noted that the request can be dispatched to the CDN node where the corresponding video is stored, i.e. $\gamma^{(v,z)}_{c,z'}(t) > 0$ only if $s^{(v)}_{DC_{z'}}(t) = 1$.

The backlog of request queues are updated as follows:

$$\begin{aligned} Q^{(v)}_c(t+1) = \max\big[Q^{(v)}_c(t) - &\sum_{z\in\mathbb{Z}(c)} \delta^{(v,z)}_c(t) \\ - \sum_{z\in\mathbb{Z}(c)} \sum_{z'\in\mathbb{Z}(c)} &\gamma^{(v,z)}_{c,z'}(t), 0\big] + \sum_{z\in\mathbb{Z}(c)} a^{(v,z)}_c(t) \end{aligned}$$

3.2.4. Optimization Objective

Our objective is to design a dynamic algorithm for the control center to optimize the video replication and request dispatching over time, such that the overall operational cost is minimized while the service quality is guaranteed. The operational cost in time slot t consists of three items, (1) the bandwidth charge $Bandwidth(t)$ for streaming video contents to users from the content source and CDN nodes; (2) the storage cost $Storage(t)$ for replicated videos at CDN nodes; (3) the content replicating cost $Copy(t)$ for copying videos from the content source to the CDN nodes. The bandwidth charge can be expressed as:

$$\begin{aligned} Bandwidth(t) = \sum_{v\in\mathcal{V}} b_v \sum_{c\in\mathcal{C}} \big[pr_{2,0} &\sum_{z\in\mathbb{Z}(c)} \delta^{(v,z)}_c(t) \\ + \sum_{z\in\mathbb{Z}(c)} \sum_{z'\in\mathbb{Z}(c)} &\gamma^{(v,z)}_{c,z'}(t) pr_{2,z'}\big], \end{aligned}$$

The storage cost can be expressed as:

$$Storage(t) = \sum_{v\in\mathcal{V}} b_v \sum_{z\in\mathcal{Z}} s^{(v)}_{DC_z}(t) pr(1,z),$$

The replication cost can be derived as:

$$Copy(t) = \sum_{v\in\mathcal{V}} b_v \sum_{z\in\mathcal{Z}} [s^{(v)}_{DC_z}(t) - s^{(v)}_{DC_z}(t-1)]^+ pr(1,0),$$

where $[x]^+ = x$ if $x \geq 0$ and $[x]^+ = 0$ if $x < 0$. Hence, the operational cost, $C(t)$, is modeled as follows:

$$C(t) = Bandwidth(t) + Storage(t) + Copy(t). \tag{2}$$

Based on the content replication policy, users can be served from CDN nodes or the content source with different level of delay. The sum of request delay in time slot t is given by:

$$\begin{aligned} Delay(t) = \sum_{v\in\mathcal{V}} \sum_{c\in\mathcal{C}} \big[&\sum_{z\in\mathbb{Z}(c)} \delta^{(v,z)}_c(t) d_{z0} \\ + \sum_{z\in\mathbb{Z}(c)} \sum_{z'\in\mathbb{Z}(c)} &\gamma^{(v,z)}_{c,z'}(t) d_{zz'}\big], \end{aligned}$$

The total request number can be expressed as:

$$\begin{aligned} Request(t) = \sum_{v\in\mathcal{V}} \sum_{c\in\mathcal{C}} \big[&\sum_{z\in\mathbb{Z}(c)} \delta^{(v,z)}_c(t) \\ + \sum_{z\in\mathbb{Z}(c)} \sum_{z'\in\mathbb{Z}(c)} &\gamma^{(v,z)}_{c,z'}(t)\big]. \end{aligned}$$

In this paper, we define the QoS metric $q(t)$ as the average round-trip delay as follows:

$$q(t) = \frac{Delay(t)}{Request(t)}. \tag{3}$$

The optimization pursued by our dynamic algorithm is formulated as follows:

$$\min \quad \bar{\mathbf{C}} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} C(t) \tag{4}$$

$$\text{s.t.} \quad \sum_{z\in\mathbb{Z}(c)} \delta^{(v,z)}_c(t) + \sum_{z\in\mathbb{Z}(c)} \sum_{z'\in\mathbb{Z}(c)} \gamma^{(v,z)}_{c,z'}(t) = a^{(v,z)}_c(t), \tag{5}$$

$$0 \leq \gamma^{(v,z)}_{c,z'}(t) \leq \mu_{max} s^{(v)}_{DC_{z'}}(t), \tag{6}$$

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} q(t) \leq \alpha, \tag{7}$$

where (5) indicates that all the requests should be dispatched in the current slot. (6) states that requests for video v are only dispatched to CDN nodes that storing the corresponding video at the time, and the number of requests dispatched from a request queue stays within an upper bound μ_{max}. (7) guarantees that the averaged time delay is smaller than α.

4. DYNAMIC VIDEO REPLICATION AND REQUEST DISPATCHING POLICY

Based on the Lyapunov optimization theory [9], we design a dynamic algorithm to solve the optimization problem in (4). Constraints (5)(6) can be addressed in each time slot. Since constraint (7) is on time-averaged variable values, we construct a virtual queue to bound the averaged delay.

4.1. Bounding Time Averaged Delay

A virtual queue R is introduced to satisfy constraint (7), which is updated in each time slot as follows:

$$R(t+1) = \max[R(t) + Delay(t) - \alpha Request(t), 0], \tag{8}$$

In this virtual queue, the 'arrival' rate is the total delay, while the 'departure' rate is the product of the total number of requests and the pre-set upper bound of round-trip delay per request. Our algorithm should adjust $\delta^{(v,z)}_c(t)$ and $\gamma^{(v,z)}_{c,z'}(t)$ to make sure $R(t)$ is stable, which indicates that the time-averaged arrival rate would not exceed the time-averaged departure rate, and hence the constraint (7) is satisfied.

4.2. Dynamic Algorithm

We adopt the Lyapunov optimization framework to stabilize all kinds of queues modeled above, and solve the problem.

Let $\mathbf{\Theta}(t) = [\mathbf{Q}(t), \mathbf{R}(t)]$ be the vector of all queues in the system. We define the quadratic Lyapunov function:

$$L(\mathbf{\Theta}(t)) = \frac{1}{2}\sum_{v\in\mathcal{V}}\sum_{c\in\mathcal{C}}(Q_c^{(v)}(t))^2 + \frac{1}{2}(R(t))^2. \qquad (9)$$

Then we define the one-slot Lyapunov drift as $\triangle(\mathbf{\Theta}(t)) = E\{L(\mathbf{\Theta}(t+1)) - L(\mathbf{\Theta}(t))|\mathbf{\Theta}(t)\}$.

According to the *drift-plus-penalty* framework in Lyapunov theory, simultaneously minimizing the upper bound of the "penalty" and stabilizing queues can be achieved by minimizing the upper bound of $\triangle(\mathbf{\Theta}(t)) + VC(t)$ in each time slot, where V is a non-negative parameter. V represents the tradeoff between the operational cost and the averaged time delay. In this way, we can derive the following inequality:

$$\begin{aligned}
&\triangle(\mathbf{\Theta}(t)) + VC(t)\\
\le& B - \sum_{v\in\mathcal{V}}\sum_{c\in\mathcal{C}}\sum_{z\in\mathbb{Z}(c)} \delta_c^{(v,z)}(t)[Q_c^{(v)}(t) + (\alpha - d_{z0})R(t)\\
&- b_v V pr_{2,0}] - \sum_{v\in\mathcal{V}}\sum_{c\in\mathcal{C}}\sum_{z\in\mathbb{Z}(c)}\sum_{z'\in\mathbb{Z}(c)} \gamma_{c,z'}^{(v,z)}(t)[Q_c^{(v)}(t)\\
&+ (\alpha - d_{zz'})R(t) - b_v V pr_{2,z'}] + V\sum_{v\in\mathcal{V}}\sum_{z\in\mathcal{Z}} b_v\\
&\big[s_{DC_z}^{(v)}(t)pr(1,z) + [s_{DC_z}^{(v)}(t) - s_{DC_z}^{(v)}(t-1)]^+ pr(1,0)\big]\\
&+ \sum_{v\in\mathcal{V}}\sum_{c\in\mathcal{C}} Q_c^{(v)}(t)\sum_{z\in\mathbb{Z}(c)} a_c^{(v,z)}(t),
\end{aligned}$$

where

$$\begin{aligned}
B =& \frac{1}{2}|\mathcal{V}||\mathcal{C}|\big[A_{max}^2 + 2(f + c_{max}\mu_{max})^2\big]\\
&+ \frac{1}{2}[|\mathcal{V}||\mathcal{C}|(fd_{0,max} + c_{max}\mu_{max}d_{max})]^2 \qquad (10)\\
&+ \frac{1}{2}\alpha^2[|\mathcal{V}||\mathcal{C}|(f + c_{max}\mu_{max})]^2,
\end{aligned}$$

where f is the maximum number of requests the content source can serve in a time slot, $c_{max} = \max_{c\in\mathcal{C}}|\mathbb{Z}(c)|$, μ_{max} is the maximum number of requests dispatched from each request queue to a CDN node; $d_{0,max} = \max_{z\in\mathcal{Z}} d_{0z}$; $d_{max} = \max_{z,z'\in\mathcal{Z}} d_{zz'}$.

By minimizing the right-hand-side of inequality, we can minimize the upper bound of $\triangle(\mathbf{\Theta}(t)) + VC(t)$, and thus the upper bound of the averaged operational cost. To simplify the description, we define the following notation:

$$\begin{aligned}
\rho_c^{(v,z)} &= Q_c^{(v)}(t) + (\alpha - d_{z0})R(t) - b_v V pr_{2,0},\\
\varrho_{c,z'}^{(v,z)} &= Q_c^{(v)}(t) + (\alpha - d_{zz'})R(t) - b_v V pr_{2,z'}, \qquad (11)\\
\sigma_z^{(v)} &= Vb_v(pr(1,z) + 1_{\{s_{DC_z}^{(v)}(t-1)=0\}}pr(1,0))
\end{aligned}$$

all of them are constants in time slot t. In addition, the last term in equality (4.2) is also constant in each time slot. Hence, the optimization problem is equivalent to:

$$\begin{aligned}
\min \quad & \sum_{v\in\mathcal{V}}\sum_{c\in\mathcal{C}}\sum_{z\in\mathbb{Z}(c)} \delta_c^{(v,z)}(t)\rho_c^{(v,z)}\\
&+ \sum_{v\in\mathcal{V}}\sum_{c\in\mathcal{C}}\sum_{z\in\mathbb{Z}(c)}\sum_{z'\in\mathbb{Z}(c)} \gamma_{c,z'}^{(v,z)}(t)\varrho_{c,z'}^{(v,z)}\\
&- \sum_{v\in\mathcal{V}}\sum_{z\in\mathcal{Z}} s_{DC_z}^{(v)}(t)\sigma_z^{(v)} \qquad (12)\\
\text{s.t.} \quad & constraints \quad (5)(6)
\end{aligned}$$

This is an integer linear program, which can be solved by traditional methods. To summarize, our dynamic algorithm works as follows: the system maintains a table of video replication information with entries $s_{DC_z}^{(v)}$. In each time slot, the system receives $a_c^{(v,z)}$ requests for video v originated from region z and community c, and enqueues them to request $Q_c^{(v)}$. Virtual queues R is updated accordingly. By constructing the queue vector $\mathbf{\Theta}$, we then solve the problem (12) to calculate the optimal video replication strategies and request dispatching policies.

We can prove that for any control variable $V > 0$, the online algorithm can stabilize the system. In particular, the resulted operational cost $\bar{C}$ satisfies $\bar{C} < C^* + \frac{B}{V}$, where C^* is the theoretical lower bound. Due to the page limit, we will demonstrate this feature in our experiments.

5. PERFORMANCE EVALUATION

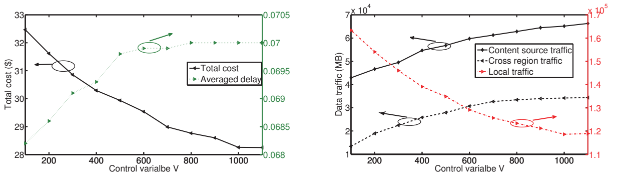
In this section, we leverage the aforementioned Sina Weibo messages to evaluate the characteristics of the proposed algorithm, and conduct a performance comparison with the traditional Least Frequently Used (LFU) algorithm.

5.1. Experiment Parameter Setting

We use the Weibo messages as the trace of video requests. The length of a time slot is one hour. The CDN pricing model is from Amazon EC2 and S3. In particular, the storage cost is $\$2 \times 10^{-13}$ per byte per hour for each CDN node. The bandwidth price for different regions follows a uniform distribution within range $[\$0.96 \times 10^{-10}, \$1.44 \times 10^{-10}]$. The bandwidth price for the content source is $\$1.2 \times 10^{-10}$. The round-trip delay between users in regions and the content source follows a uniform distribution within range $[0.05, 0.25]$, while the round-trip delay between two regions is proportional to their geo-distance.

5.2. Performance Evaluation

1) *Tradeoff Among Metrics Under Different V:* We characterize the tradeoff among the total cost vs. the time averaged time delay and different network traffics, under distinctive V. Fig. 6(a) indicates that with the increase of V, the total cost



(a) Total cost and time averaged delay under different V

(b) Data traffic from different sources under different V

Fig. 6. Tradeoff among different metrics, $\alpha = 0.07$

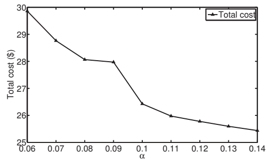


Fig. 7. Total cost under different α

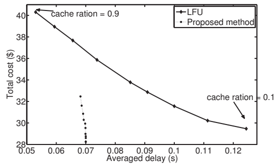


Fig. 8. Performance comparison with the traditional LFU algorithm

decreases and converges to the optimal value, which corroborates our previous discussions. However, the time averaged delay grows with the increase of V, and approaches to the pre-set α. This suggests that we can dynamically choose V to manage the tradeoff between the total cost and the time averaged delay. In Fig. 6(b), we can see that the local traffic decreases with the growth of V, while the content source traffic and the cross-region traffic increase. Hence, we can conclude that the saving cost is mainly due to more requests are served by the content source, leading to less storage cost.

2) *Tradeoff Between Time Delay and Total Cost:* To characterize the tradeoff between the averaged time delay and the total cost, we set $V = 800$ and adjust the given constraint parameter α from 0.06 to 0.14. When α increases, which means longer delay can be tolerated by users, more requests are tend to be dispatched to the content source and the total cost will be reduced. However, our system should make a tradeoff between the service quality and the operational cost.

3) *Performance Comparison:* We compare our algorithm with the LRU algorithm, in which only the most popular videos will be cached to the local CDN node and users can only access videos either from the content source or the local CDN node. In particular, we set $V = 100, 200, ..., 1100$ for the proposed algorithm with the delay constrain $\alpha = 0.07$, and the cache ration with the range of $[0.1, 0.9]$ for the LFU algorithm. From Fig. 8, we can see that as the decrease of the cache ratio, the operation cost will be lessened prominently with the sacrifice of longer round-trip delay. In contrast, our algorithm can keep the round-trip delay around the pre-set threshold. Moreover, our algorithm can achieve less cost than the LFU algorithm, while having small averaged time delay.

6. CONCLUSION

This paper investigated the social video distribution in the context of cloud-centric CDN network, with the objective of minimizing the operational cost, while satisfying the QoS requirement. We first took a measurement on the Sina Weibo dataset to reveal the community characteristics of the social video viewing behavior. Based on the observation, we formulated the content distribution into a constrained optimization problem. Applying the Lynapunov optimization framework, we developed a dynamic algorithm to seek the optimal solution without the knowledge of the future status. Finally, we verified the derived algorithm based on the Weibo messages and the Amazon cloud pricing model. The results suggested that our algorithm can make a better balance than the traditional method, with lower operational cost and better QoS.

7. REFERENCES

[1] “ForeSee,” 2012, http://www.foreseeresults.com/.

[2] Alan Mislove, Bimal Viswanath, Krishna P Gummadi, and Peter Druschel, “You are who you know: inferring user profiles in online social networks,” in *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 251–260.

[3] Andrea Passarella, “A survey on content-centric technologies for the current internet: Cdn and p2p solutions,” *Computer Communications*, vol. 35, no. 1, pp. 1–32, 2012.

[4] Zhi Wang, Lifeng Sun, Xiangwen Chen, Wenwu Zhu, Jiangchuan Liu, Minghua Chen, and Shiqiang Yang, “Propagation-based social-aware replication for social video contents,” in *Proceedings of the 20th ACM international conference on Multimedia*. ACM, 2012, pp. 29–38.

[5] Feng Wang, Jiangchuan Liu, and Minghua Chen, “Calms: Cloud-assisted live media streaming for globalized demands with time/region diversities,” in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 199–207.

[6] Brendan J. Frey and Delbert Dueck, “Clustering by passing messages between data points,” *Science*, vol. 315, pp. 972–976, 2007.

[7] “Amazon Elastic Compute Cloud,” http://aws.amazon.com/ec2.

[8] “Amazon Simple Storage Service,” http://aws.amazon.com/s3.

[9] Michael J Neely, “Stochastic network optimization with application to communication and queueing systems,” *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.