Predicting Popularity and Adapting Replication of Internet Videos for High-Quality Delivery

Guthemberg Silvestre*†, Sébastien Monnet*, David Buffoni*, and Pierre Sens*
*LIP6/UPMC/CNRS/INRIA
4 place Jussieu - 75005 Paris - France.
†Orange Labs
38-40 rue du Général Leclerc - 92130 Issy - France.
Email: {firstname.lastname}@lip6.fr

***Abstract*—Content availability has become increasingly important for the Internet video delivery chain. To deliver videos with an outstanding availability and meet the increasing user expectations, content delivery networks (CDNs) must enforce strict QoS metrics, like bitrate and latency, through SLA contracts. Adaptive content replication has been seen as a promising way to achieve this goal. However, it remains unclear how to avoid waste of resources when strict SLA contracts must be enforced. In this work, we introduce Hermes, an adaptive replication scheme based on accurate predictions about the popularity of Internet videos. Simulations using popularity growth curves from YouTube traces suggest that our approach meets user expectations efficiently. Compared to a non-collaborative caching, Hermes reduces storage usage for replication by two orders of magnitude, and under heavy load conditions, it increases the average bitrate provision by roughly 90%. Moreover, it prevents SLA violations through an application-level deadline-aware mechanism.**

Keywords*—*Video Quality, Popularity Growth, Peer-to-Peer, Hybrid CDN, Replication, SLA, Prediction.

I. Introduction

Multimedia content distribution over the Internet has increased dramatically in the recent years. A study published by Cisco System, Inc [3] revealed that the global Internet video traffic has surpassed peer-to-peer traffic since 2010, becoming the largest type of Internet traffic. Cisco also forecasts that video traffic will reach 86% of the global consumer traffic by 2016, including TV, video-on-demand (VoD), live streaming, and peer-to-peer (P2P) file sharing.

In parallel, Internet access has become ubiquitous, continuously faster, and cheaper. These advances have contributed to increase the expectations of consumers on Internet services. Today, content availability is critical, particularly for streaming traffic, that includes VoD and live streaming. On the one hand, for many workloads, such as social network messaging or search engines, QoS metrics can be defined in term of latency of short transactions. On the other hand, streaming traffic is more sensitive to buffering, where a stable and high bitrate is essential. For example, Dobrain *et al.*[5] found that a 1% increase in buffering ratio can reduce the consumer's expected viewing time by more than three minutes. This suggests that SLA contracts must include the bitrate as a key QoS metric.

Yet current CDN architectures are not ready to fulfil the requirements of the increasing demand for streaming and meet consumers' expectations. Through fine-grained client-side measurements from over 200 million client viewing sessions, Liu *et al.*[9] showed that 20% of these sessions experience a re-buffering ratio of at least 10%, 14% of users have to wait more than 10 seconds for video to start up, more than 28% of sessions have an average bitrate less than 500Kbps, and 10% of users fail to see any video at all.

To cope with these issues, CDN providers have started to adopt for hybrid designs, that combine datacenters and edge network resources [1]. The aim is to combine the advantage of infrastructure-based and P2P systems. But, the resource allocation on hybrid CDNs to meet user expectations still imposes big challenges, particularly if a minimal average bitrate has to be enforced. This paper identifies *adaptive content replication* as one of such challenges. Adaptive replication plays an important role on the content availability. As the popularity of a video increases, the number of replicas must be adapted accordingly. Generally speaking, the faster and more precise the replication scheme reacts to changes on videos popularity, the better is the resource allocation towards high content availability. However, to identify

1521-9097/13 $31.00 © 2013 IEEE
DOI 10.1109/ICPADS.2013.64

popular videos precisely and to define the replication degree properly are far from being trivial tasks.

In this work, we present Hermes, an adaptive replication scheme for offering highly available Internet videos on hybrid CDNs. Hermes is based on predictions of videos' popularity. For that, we designed a learning model using non-linear *support vector machine* (SVM) methods. Inputs of our model come from lightweight measurements of the request arrival process. Evaluations with growth curves from YouTube traces show that our predictions of popularity are accurate. That allows us to prevent violations of strict SLAs by enforcing simple replication policies. Our approach is flexible and can be easily extended to different CDN scenarios.

This work makes two main contributions:

- We design and evaluate a predictor of Internet video popularity with YouTube traces. Our predictor tracks the dynamics of popularity growth curves accurately based on measurements of request arrivals; thus, the prediction model is flexible enough for being used in different deployments.
- Based on our accurate predictions, we designed and evaluated Hermes, an easy-to-deploy, adaptive replication scheme that provides highly available Internet videos. Simulations on top of PeerSim[10] show that Hermes outperforms a non-collaborative caching by reducing storage and network usage. Unlike most of the recent deadline-aware approaches, Hermes does not require any modification of network stack to enforce strict QoS metrics.

The rest of this work is organized as follows. Section II presents our datasets, and measurements for predictions. Section III describes the learning model. In Section IV, we describe our evaluation scenario for edge resources in a hybrid CDN, then we show the performance analysis of our replication scheme. Section V discusses related work, and Section VI concludes.

II. Motivation and Measurements for Predictions

In this section, we discuss the role of adaptive replication schemes in content distribution. We present our workload with popularity growth curves from real YouTube traces, measurements, and datasets.

A. On the Track of YouTube Popularity Growth Curves

A fair reproduction of user interactions to Internet videos is essential to evaluate an adaptive replication scheme properly. Hence, we carefully set-up our workload to combine YouTube traces [6] to well-known videos' access patterns [14].

Figueiredo *et al.* [6] collected and characterized the growth patterns of YouTube videos, whose datasets are currently available online [16]. They analysed three types of YouTube videos sets: videos that appear on YouTube top list, videos that were banned from YouTube due to copyrights violations, and videos that were randomly selected through API calls. They crawled once a number of videos' daily features. For each video, there are up to 100 daily measurements, or daily available samples, per feature. In this work, we are mostly interested in the measurements of *view data* feature, that depicts the popularity growth curve of a video through a array of cumulative number of daily views ranging from 0 to the total number of views.

Before integrating to our workload, we first processed the YouTube datasets to remove inconsistent measurements, such as videos with no views. Basically, we got rid of videos with small number of total views (those smaller than the first quartile) and videos with few daily measurements (those smaller than the third quartile). That allows us to pick off 20% most representative YouTube growth patterns, accounting for 21827 distinct curves. Then, for a matter of simplicity, we randomly selected, with a uniform distribution, curves from this preprocessed data to be assigned to videos of our workload. To summarize, Table I lists default values for workload parameters. In our simulations, videos are always divided in chunks of fixed size, 2MB. Assuming that all consumers expect the same minimal QoS metric for buffering their videos, we define a SLA contract whose the minimal average bitrate is 14 chunks/s. We consider that a SLA violation occurs whenever a viewer does not observe her minimum average bitrate.

TABLE I. Default values for workload parameters

Workload	
Requests per user	uniform
Experiment duration	4 hours
Mean requests per second	100
Requests fractions	5% of creations, 95% of views
Object size (follows Pareto)	shape=3, from 13MB to 1.6GB
Video popularity (Zipf-Mandelbrot)	shape=0.8, cutoff=# of videos
Videos' creation (Poisson)	λ=creations per second
Popularity growth from YouTube traces	21827 distinct patterns

B. Adaptive Replication Schemes for Highly Available Content

Replication schemes have become an important building block for Internet video providers to improve content availability and meet consumers expectations. A *good* popularity-aware replication scheme should offer content replica maintenance to handle popularity growth properly.

Non-collaborative caching remains the simplest approach to provide popularity-aware replication of web content [8]. They adapt the replication degree to the content popularity using cache replacement policies, and assuming fair-sharing as a key scheduling strategy. But, Internet videos' workloads on hybrid CDNs present new challenges for non-collaborative caching, e.g. smaller and highly heterogeneous storage for replicas, and a growing need for high bitrate provision for meeting consumers' expectations. Therefore, relying just on cache replacement policies and fair-sharing scheduling can undermine the performance of the whole system.

Our previous work, AREN [13], presents a novel adaptive replication scheme which was designed with these issues in mind. AREN relies on collaborative caching and bandwidth reservation mechanism to adapt the replication degree of contents and to enforce SLA contracts for costumers. It applies a simple mechanism of popularity classification and content replication based on the current sum of bandwidth reservation and low/high bandwidth thresholds. Simulations with synthetic workload demonstrated that this approach provides near-optimal results, providing an outstanding content availability. It outperformed non-collaborative caching by preventing almost 99.8% of SLA violations. By reducing the total number of replicas, AREN reduces storage usage for replication and increases the aggregate bandwidth. Unlike non-collaborative caching, AREN reduces the dependency on cache replacement policies by decreasing consistently the number of replicas.

Although AREN's results showed that it is highly efficient in replicating Internet video workloads, its deployment raises considerable issues for Internet providers. One of the main disadvantage of this approach, that can make Internet providers reluctant to its use, is that it requires changes of the functioning of the network stack. Efficient bandwidth reservation for meeting deadlines, like D^2TCP[15], requires major adjustments to the transport network layer to provide end-to-end bandwidth reservation properly.

To overcome this important issue, and encouraged by findings with AREN's threshold-based approach, we introduce a flexible learning model for predicting popularity and replication degree. It tracks popularity growth of Internet videos based on lightweight measurements of the request arrival process. The aim is to instrument a collaborative caching, creating and deleting replicas, according to video access patterns. We argue that, through accurate predictions, we are able to react to popularity growth changes promptly, and prevent SLA violations.

C. Measurements and Dataset for Predictions

One of our first efforts towards accurate predictions was to gather as much information about users' interactions as possible in an easy manner. We run simulations with the workload described in Subsection II-A for collecting those measurements.

Our data comes from 10 lightweight measurements of the request arrival process: video size, network availability, network usage (load), current number of viewers and replicas, inter-arrival time between requests (delta), aggregate number of views, mean of time between requests (mtbr), life time, and average bitrate. We chose this approach because it provides a simple procedure to collect information of consumers' interactions. In hybrid CDNs, this data can be collected from logically centralised coordinator servers that are already in charge of accountability or admission control tasks. In addition, we added labels to each line of our measurements. Labels track the behaviour of AREN functioning, and allow us to classify requests. For instance, labels permit distinguishing *popular* from *non-popular* videos. We described these labels as follows:

Non-popular videos: Videos with non-popular labels are those whose access pattern of its request arrival process has not trigged any increasing on the initial replication degree. According to recent findings [14], the popularity of Internet videos follows a Zipf-like distribution, consequently most of them likely belong with this group. In AREN, they do not require any extra replica.

Popular videos: If during the simulations, a video has its replication degree modified by AREN, we attribute a popular label to it. In addition, we introduced further information to this group in order to capture the behaviour of the replication maintenance. Depending on the decision taken by AREN, there will be three types, or subclasses, of popular videos: *increasing*, *keeping*, or

decreasing. This allows us to interpret the measurement as a trigger for changing the resource allocation of that video, in our specific case, modifying the number of replicas.

III. Learning Model

We describe our statistical learning model in this section. First, we present a brief overview of statistical learning modelling. Then, we explain the design of our model, detailing our two-step approach. Finally, we describe our implementation and framework for learning.

A. Statistical Learning Overview

Statistical learning is about learning from seen data in order to predict unseen data with minimal error. Data comprises measurements represented by a feature vector $\mathbf{x}$ with a fixed number of dimensions p ($\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p$) from the input space $\mathcal{X}$. Broadly speaking, there are two ways of learning from data: supervised and unsupervised learning.

In supervised learning, each measurement or input is coupled with a y, a label, from the output space $\mathcal{Y}$. To learn, we have N pairs $(\mathbf{x}, y)$ drawn *independent and identically distributed* (i.i.d.) from a fixed but unknown join probability density $Pr(X, Y)$. This is true for both training and testing datasets. For instance, we consider the training dataset $S = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$ of N pairs $(\mathbf{x}, y)$. From this dataset, the supervised learning algorithm searches for a function $f : \mathcal{X} \to \mathbb{R}$ in a fixed function class $\mathcal{F}$. State-of-the-art algorithms, such as *support vector machines* (SVM) [4] or *Adaboost*[7], aim to find $f^\star$ in $\mathcal{F}$ with the lowest empirical risk defined as:

$$f^\star \in \arg\min_{f \in \mathcal{F}} \mathtt{r}_{emp}(f)$$

where $\mathtt{r}_{emp}(f) = \frac{1}{N} \sum_{i=1}^{N} I_{\{f(\mathbf{x}) \neq y_i\}}$ is computed over the training set, and $I_{\{.\}}$ is the indicator function which returns 1 if the predicate $\{.\}$ is true and 0 otherwise.

In unsupervised learning, we have N samples $(x_1, x_2, \ldots x_N)$ of a random p-vector X having probability density Pr(X). Unlike supervised learning, we do not have outputs to learn. Instead, we are interested in inferring the properties of the probability density Pr(X). This allows us to have insights into how the data is organized or clustered.

B. Learning Model for Internet Videos

In this work, data comes from users' interactions with Internet videos, so-called request arrival process. We assume that there exists data with *near-optimal* results from where we can learn. As described in Section II, the data for learning comes from simulations using AREN replication scheme. Each dataset line contains 10 lightweight measurements of request arrival process and a label, as described in Subsection II-C. We denote as *inputs* the measurements of the request arrival process, and as *outputs* the popularity labels.

In Subsection II-C, we present two classes of outputs: non-popular and popular. Then, we describe that there are three subclasses for popular videos. Therefore, we model our problem in a two-step approach as follows:

Popularity classifier: This learner allows us to classify videos into non-popular and popular. Since the popularity of Internet videos follows a Zipf-like distribution, popular videos can be seen as rare events. Hence, we identify popular videos as anomalies through an unsupervised learning method with binary outputs.

Replication classifier: Here we consider popular videos only. There are three subclasses of replication for popular videos: increasing, keeping, and decreasing. In this case, we use a multi-class supervised learning method.

C. Framework for Learning and Predicting, and Implementation

Our two-step classifier is based on *support vector machine* (SVM) methods [4]. According to Friedman *et al.*, SVMs are a set of robust supervised learning methods, that produce accurate, non-linear boundaries for classifiers by constructing a linear boundary in a large, transformed version of the input space. We implemented our learning model as a as part of Hermes replication scheme using Scikit-learn, a general-purpose machine learning library [11]. From Scikit-learn, we selected two main procedures: `sklearn.svm.OneClassSVM` for popularity classifier, and `sklearn.svm.SVC` for replication classifier.

We designed a simple framework to use our Hermes's learning module, depicted in Figure 1. Our framework has two phases: (i) learning and (ii) predicting. Each phase has its own YouTube-like workload. In the learning phase, we first generate the training dataset with AREN. Then we feed this training dataset to Hermes in order to identify YouTube popularity patterns. Once the learning

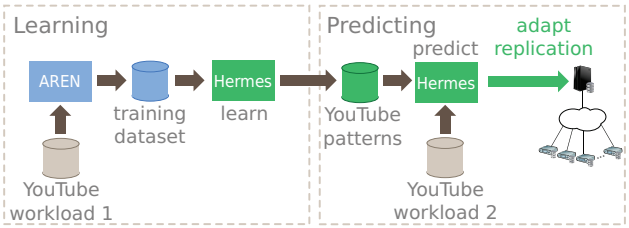

Fig. 1. Framework for learning and predicting Internet videos popularity.

phase has been accomplished, Hermes can use its learning module in a predicting phase, as indicated in the right-hand side of Figure 1. In this phase, inputs come from measurements of the the request arrival process of workload 2, that permit classifying requests to popular videos and instrumenting replication accordingly.

IV. Evaluation

Our utmost performance goal is to prevent *all* SLA violations. As detailed in Subsection II-A, a SLA violation occurs whenever a viewer does not observe her minimum average bitrate. We are also interested in reducing storage and network usage as much as possible. We focus on the storage usage for replication. In terms of network usage, we are particularly interested in evaluating the bitrate provision under heavy load. First we introduce the scenario and the replication schemes evaluated in this work. Then we present our most relevant results.

A. Evaluation Scenario and Replication Schemes

We evaluated this work with Caju, a tool which models a content distribution system for edge networks on top of PeerSim simulation engine. In Caju, the service provider infrastructure is organized in federated storage domains, as depicted in Figure 2. A storage domain is a logical entity that aggregates a set of storage elements that are located close to each order. There are two different classes of devices: (i) operator-edge, furnished by storage operators, e.g. small-sized datacenters, represented by big nodes up on Figure 2, and (ii) consumer-edge, the small ones, whose consumers are connected to, such as home gateways.

System interactions are straightforward. Users can either share or view videos. For sharing, given a fixed number of initial replicas n, it simulates the initial video creation and a chain of object-replication of $n-1$ stages. A view request is served by at most R nodes with uniform load. Available sources come from $r = min(n, R)$. We set R to five for all experiments. A detailed description of Caju is available in [13].

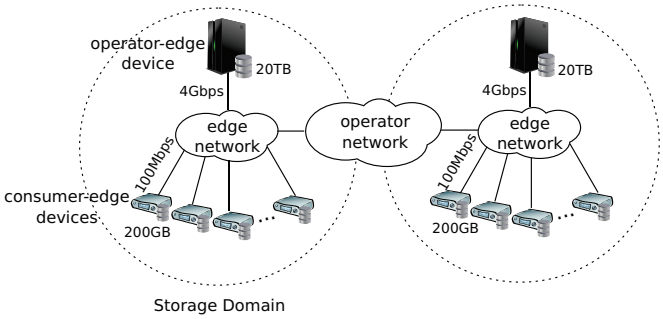

Fig. 2. Evaluation scenario

Our evaluation scenario (Figure 2) includes 4002 nodes, arranged across two storage domains. There are one operator-edge device and 2000 consumer-edge devices per storage domain. Storage and network capacities differ according to the device class. Operator-edge devices have 20TB of storage capacity and full-duplex access link of 4Gbps. Consumer-edge devices contribute 200GB each, equipped with 100Mbps full-duplex links. Note that the two operator-edge devices contribute with a small fraction of aggregate edge resources, i.e. 5% of the storage capacity and only 2% of the total network capacity. This draws our attention to the performance of replication schemes towards consumer-edge resource usage. We assume that edge networks are connected to the operator network that ensures inter-storage domain connectivity. We assume only 1% consumer-edge devices storage is available for caching additional replicas, namely 2GB.

We evaluate three replication schemes.

Non-collaborative caching Adaptive replication schemes based on non-collaborative caching, such as those that use Least Recent Used (LRU) algorithm, are easy to implement and deploy. A new replica is created whenever a user requests to view a video. LRU replacement is enforced regarding the static percentage of the local storage capacity for caching of 1%.

AREN That stands for Adaptive Replication for Edge Networks. It relies on bandwidth reservation and collaborative caching to adapt the replication degree of popular content. Considering a logically centralized coordinator, AREN tracks the active aggregate bandwidth per content, and decides if it is worth creating a new replica in the viewer side. The coordinator computes the utility of new

replicas based on thresholds. Replica utility measures the benefit of creating replicas with regard to popularity and current bandwidth consumption of a video. It also checks if replicas are redundant and must be deleted. For scheduling on edge networks, AREN enforces two simple policies: *divide-and-conquer* and *nearest source selection*. Further details about AREN are available in our previous work [13].

Hermes This is our main contribution. It provides a proper, adaptive replication scheme for Internet videos that enforces strict SLA contracts through accurate predictions. More interestingly, it does not requires any modification of the network stack, as most of deadline-aware approaches do. Hermes implements our learning model for Internet videos, described in Section III. This module permits identifying popular Internet videos based on lightweight measurements of request arrival process. Since Hermes predicts accurately requests to popular videos, we argue that enforces simple replication policies is enough to prevent both violations and waste of resources. To evaluate this idea, we define d as the number of additional replicas to cope with the popularity growth curve. Therefore, whenever a video is classified as popular, new d-replicas are created. Similarly, Hermes reduces replication degree according to the video popularity. Hermes enforces the same AREN's policies for requests scheduling on edge resources.

B. Predictions and Replication Performance

Hermes relies on predictions of content demand to identify popular videos and to enforce QoS metrics through replication. Hermes' performance depends mainly on (i) prediction accuracy and (ii) the efficiency of the replication policy. In Section III, we explain that our two-step classifier relies on SVM methods. To measure the prediction accuracy of each step, we vary the kernel, the main SVM parameter. We consider four kernels: *Radial Basis Function* (RBF), Linear, Polinomial (Poly), and Sigmoid. For evaluating our classifier, we use the framework described in Subsection III-C.

Popularity prediction accuracy: The first step of our learning model predicts Internet videos popularity through a binary classification. We used a dataset with 286823 samples of view requests, whose 1.31% of them belong to popular videos. Figure 3 depicts the *receiver operating characteristic* (ROC) curve. ROC curve is one of the most common ways of evaluating the efficiency of a binary classifier. This plot allows us to select the best classifier by measuring the true positive rate versus the false positive rate, and by computing the area under the ROC curve (AUC), where the value 1 represents the optimal classifier. Using RBF kernel, our classifier reaches an AUC of 0.97, quite close to the optimal value. Therefore, RBF kernel is the best choice for predicting popularity.

Replication prediction accuracy: For the second step of our learning model, the goal is to predict the replication action for popular videos in three classes. The dataset for this step contained 612754 view requests. Figures 4 shows total precision rates using different SVM kernels. RBF outperforms the three other kernels with the highest precision rate of 0.98, becoming our best choice. Unlike popularity predictions results, Linear and Poly kernels performed quite well, both scoring 0.97.

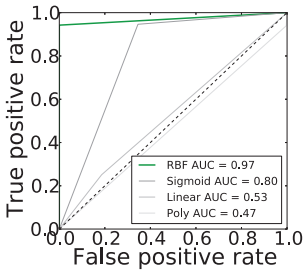

Fig. 3. ROC curve for popularity classifier.

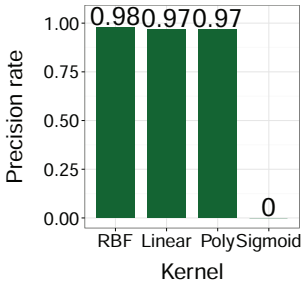

Fig. 4. Precision rate per kernel.

Evaluating the replication policy: Whenever the learning module of Hermes predicts that a video needs more replicas, we assume that d new replicas must be created once for preventing violations. Figure 5 measures the number of violations for different values of d, whose values vary from one to 13. When d ranges from seven to 10, there is no violations. This suggests that since popularity predictions are accurate, a simple replication policy should suffice. However, if d is bigger than 10, replication adds enough load to cause violations. Hence, we select d equal to seven as the most appropriate value for preventing violations.

C. Resource Allocation Results and Analysis

We compare Hermes with a non-collaborative caching and AREN, all described in Subsection IV-A. We evaluate the network and storage usage, as well as the number of violations.

We aim to adapt the number of replicas to the number of views of a video, especially for the most popular

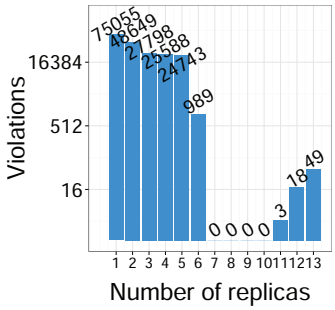

Fig. 5. Parameter d for adjusting replication degree.

ones. Figure 6 plots the maximum number of replicas for the 1% most popular videos. Using caching, the maximum number of replicas is high, raging from 817 to 1377. AREN permits decreasing significantly the lower and upper limits, to 7 and 39. Hermes also reduces the maximum replica range, which is from 9 to 58. More interestingly, the shape of the replication curves of Hermes and AREN are quite similar indeed. It confirms that our predictions are accurate, and that a simple replication policy works properly.

Reducing the number of replicas implies that the systems requires less storage for replication. Figure 7 shows storage usage for replication by replication scheme. Although Hermes utilizes more storage for replication than AREN, its usage remains two orders of magnitude below a non-collaborative caching. The maximum storage usage for AREN, Hermes, and a non-collaborative caching were 3, 49, and 7956 GB respectively. Hermes creates more replicas than AREN because it does not rely on bandwidth reservation to prevent violations. Despite that, Hermes maintains replicas efficiently, keeping storage usage very low, and making cache replacement policies redundant.

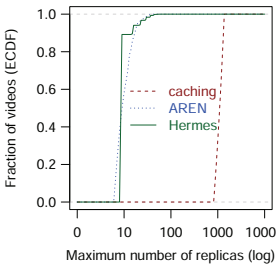

Fig. 6. The maximum number of replicas for the 1% most popular videos.

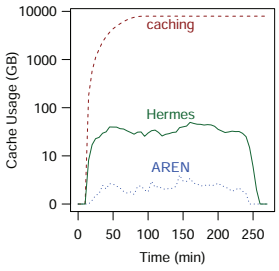

Fig. 7. Storage usage for replication.

In terms of violations, Hermes performance is also quite similar to AREN. Hermes prevents all violations. Each point of the Figure 8 represents the number of SLA violations for intervals of five minutes. Overall, caching caused 1569 violations affecting almost one third of all viewers, AREN had one violation, and Hermes none. As AREN, Hermes prevents violations by (i) creating new copies for popular videos only, and (ii) adapting the number of replicas properly. Vertical lines in Figure 8 represent the first access to the three popular videos with the worst content provision through caching. They account for 96.81% of all caching violations. The appearance of these videos puts the system under heavy load, which makes caching fails to prevent violations.

Figure 9 depicts the average bitrate for viewers of the three videos with the worst content provision using caching. When caching was under heavy load, half of viewers experienced a very low bitrate, raging between 460Kbps and 4860Kbps. The mean bitrate with caching was 45Mbps. On average, Hermes improved this bitrate by roughly 90% under heavy load. AREN comes just behind, improving bitrate provision by 87%. This find suggests that Hermes largely outperforms caching, and provides still better than AREN under heavy load conditions.

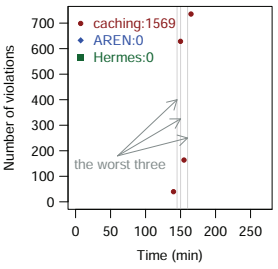

Fig. 8. SLA violations. Vertical lines show when the first access to three videos with the worst content provision using caching happen.

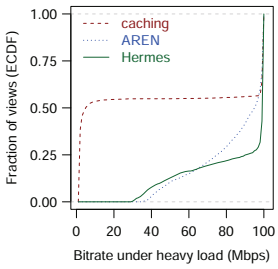

Fig. 9. Bitrate for viewers of the three most popular videos under heavy load.

V. Related Work

Our related work is two-fold: Internet videos and adaptive replication schemes.

Internet videos: Recent studies [6], [14] have drawn attention to reach a better understanding of Internet videos properties, such as popularity growth. They point out that well-known online content popularity characteristics are applicable to multimedia content. For instance, Internet videos popularity distribution follows power law,

and popularity bursts have a short duration and are quite likely to happen just after the content publication. Dobrian *et al.* [5] shed some light on the performance of Internet videos provision on CDNs. They show that average bitrate plays an important role in videos availability. Liu *et al.* [9] make a case for a video control plane that can use a global view of client interactions and network conditions to dynamically optimize the video delivery in order to provide a high quality viewing experience despite an unreliable delivery infrastructure. However, the granularity of their server selection mechanism is at a CDN, ignoring edge network resources. Hermes addresses this issue by adapting replication close to the viewers. Thus, Hermes can be play a important role in collaborating with an Internet control plane.

Adaptive replication schemes: Non-collaborative caching remains the simplest approach to provide popularity-aware replication of web content through cache replacement policies[8]. However, we showed when we adapt the number of replicas according to the Internet video popularity properly, cache replacement policy becomes redundant. EAD [12] and Skute [2] adapt the number of replicas by using a cost-benefit approach over decentralized and structured P2P systems. EAD creates and deletes replicas throughout the query path with regard to object hit rate using an exponential moving average technique. Similarly, Skute provides a replication management scheme that evaluates replicas price and revenue across different geographic locations. Despite presenting an efficient framework for replication, they provide an inaccurate bitrate provision, hence inappropriate for high-quality video delivery. AREN [13] overcomes these issues by combining bandwidth reservation and collaborative caching successfully. Yet, its functioning depends on modification of the network stack. Hermes solves this issue through analysing the request arrival process, performing accurate predictions of Internet videos popularity, and maintaining replication degree accordingly.

VI. Conclusions

In this work, we presented Hermes, an adaptive replication scheme for offering highly available Internet videos on hybrid CDNs. To adapt replication, we proposed a learning model that tracks popularity growth curves based on lightweight measurements of the request arrival process. Simulations with YouTube traces showed that our predictions are accurate. That allowed Hermes to maintain the replication degree of Internet videos properly. Our evaluation results highlight that Hermes increases the average birate provision by roughly 90%, contributing decisively to enhance viewing experience of users. Our future work will mainly cover a proof-of-concept prototype for evaluating Hermes using a real testbed.

References

[1] Akamai acquires red swoosh. http://www.akamai.com/html/about/press/releases/2007/press_041207.html, 2007.

[2] N. Bonvin, T. G. Papaioannou, and K. Aberer. A self-organized, fault-tolerant and scalable replication scheme for cloud storage. In *SOCC*, 2010.

[3] Cisco visual networking index: Forecast and methodology, 2011-2016. www.cisco.com, 2012.

[4] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 1995.

[5] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang. Understanding the impact of video quality on user engagement. In *SIGCOMM*, 2011.

[6] F. Figueiredo, F. Benevenuto, and J. M. Almeida. The tube over time: characterizing popularity growth of youtube videos. In *WSDM*, 2011.

[7] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*. Springer-Verlag, 1995.

[8] S. Jin and A. Bestavros. Popularity-aware greedydual-size web proxy caching algorithms. In *ICDCS*, 1999.

[9] X. Liu, F. Dobrian, H. Milner, J. Jiang, V. Sekar, I. Stoica, and H. Zhang. A case for a coordinated internet video control plane. In *SIGCOMM*, 2012.

[10] A. Montresor and M. Jelasity. PeerSim: A scalable P2P simulator. In *P2P*, 2009.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2011.

[12] H. Shen. An efficient and adaptive decentralized file replication algorithm in p2p file sharing systems. *IEEE Transactions on Parallel and Distributed Systems*, 2010.

[13] G. Silvestre, S. Monnet, R. Krishnaswamy, and P. Sens. Aren: a popularity aware replication scheme for cloud storage. In *ICPADS*, 2012.

[14] G. Szabo and B. A. Huberman. Predicting the popularity of online content. *Communications of the ACM*, 2010.

[15] B. Vamanan, J. Hasan, and T.N. Vijaykumar. Deadline-aware datacenter tcp (d2tcp). *SIGCOMM*, 2012.

[16] The tube over time: Characterizing popularity growth of youtube videos. http://www.vod.dcc.ufmg.br/traces/youtime/data/, 2013.