AREN: a popularity aware replication scheme for cloud storage

Guthemberg Silvestre, Sébastien Monnet, Ruby Krishnaswamy, Pierre Sens

AREN: a popularity aware replication scheme for cloud storage

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

***Abstract*—Delivering on-demand web content to end-users in order to carry out strict QoS metrics is not a trivial task for globally distributed network providers. This task becomes still harder when content popularity varies over the time and the SLA definitions have to include both transfer rate and latency metrics. Current worldwide content delivery approaches and datacenter infrastructures rely on cumbersome replication schemes that are agnostic to edge-network resources, and damage content provision.**

In this work we present AREN, an novel replication scheme for cloud storage on edge networks. AREN relies on a collaborative cache strategy and bandwidth reservation to adapt the replication degree according to strict SLA contracts and content popularity growth. We have evaluated the performances of replication schemes on edge networks using Caju, a content distribution system for edge networks. Compared to a non-collaborative caching, evaluations show that AREN prevents nearly 99.8% of all SLA violations when the storage system is heavily loaded. We also show that AREN provides a sevenfold decrease in the amount of storage usage for replicas, and it increases by roughly 20% the aggregate bandwidth, hence accelerating content delivery.

***Keywords*-Datacenter, replication, online services, SLA, popularity growth.**

I. Introduction

Multimedia content delivery has changed dramatically in the recent years. Content distributed networks (CDNs) have allowed operators to provide content to the masses. Nowadays, ordinary users are able to reach worldwide audiences thanks to web platforms deployed on top of CDNs.

In order to deliver popular content efficiently, CDNs have to provide mechanisms, and schemes, such as data replication, that are able to track content popularity growth properly.

Today's CDN architectures are deployed on big, remote and centralized sites, close to the core networks. Despite being definitively scalable architectures for content delivery, datacenters remain huge distributed systems that are very expensive to build and operate. Its resource allocation efficiency relies mainly on over-provisioning. Since most of CDN's mechanisms are agnostic to edge networks load, their infrastructures are not able to enforce strict Service Level Agreement (SLA) contracts that include transfer rate.

Resource allocation at the edge of the networks presents several advantages over traditional CDN deployments, such as the lowest ever latency, and fined grained bandwidth allocation. It might also be seen as eco-friendly, because it allows us to reduce the energy cost of data transmission, since it might dramatically decrease the path length between the content source and destination.

Bandwidth and storage capacities available on edge networks have increased dramatically in the recent years. In the beginning of 2011, Free, a French internet provider, offered to their subscribers internet connection speed up to 100Mbps , and storage capacities at home in the order of 250GB. These available resources have contributed to create and popularize internet service offers, such as video on demand, high quality, streaming, backup and high speed storage synchronization.

One of the most important new opportunities for network providers is to provide cloud storage at the edge of the network. Cloud storage in edge networks will allow system architects to design services with outstanding content delivery guarantees that take advantage of very low latencies, high data transfer, and huge amounts of storage capacities. However, edge resources management to deliver cloud services remains a big challenge for edge operators, and must be wisely allocated.

We consider that data replication plays an important role on scenario. However, it is hard to define replication schemes for edge networks that fairly adapts the placement and the number of replicas for popular content, especially if strict SLA contracts have to be enforced.

This work presents AREN, an Adaptive Replication scheme for Edge Networks that enforces strict SLA metrics with efficient resource allocation. AREN minimizes the number of SLA violations by (i) tracking bandwidth reservation mechanism on edge nodes, and (ii) operating collaborative caching mechanism properly. By simulation, we evaluate the number of strict SLA violations, storage, and bandwidth usage, for AREN and compare our results to common replication schemes. We show that AREN prevents the vast majority of SLA violation under heavily load situations. It also reduces by nearly seven-fold the required storage usage for replication through caching, and it increases by roughly 20% the aggregate bandwidth.

This work makes two main contributions:

- By simulations on top of PeerSim[11], we evaluate extensively the performance of different data replication schemes for providing popular content delivery with strict SLA at the edge of the networks.
- We present the design and evaluation of AREN, a novel replication scheme, that provides high-quality content delivery for popular content. AREN prevents most of SLA violations, and improves the cloud storage performance, by increasing the aggregate bandwidth and reducing storage usage for replication.

The rest of this work is organized as follows. Section II

covers some background of the today's content distribution systems. Section III presents our approach to tackle replication of popular content, and provides an in-depth description of Caju, our evaluation scheme for CDNs at edge networks. In Section IV, we analyse and explain our evaluation scenario and performance results. In Section V, we present related work. Finally, Section VI shows future work and concludes.

II. Background

In this section, we briefly discuss the role of edge networks in CDNs, and we present challenges faced by network providers in order to deal with popular content delivery properly.

Content distribution networks and edge networks: Content distributions networks (CDN) are distributed system that maintain content servers in many different locations in order to cope with load management in scalable way, and also to enhance latency and bandwidth available for clients. There are two types of servers in CDN compositions: origin and replica servers (so-called surrogate servers) [14]. We can therefore differentiate CDNs on the basis of their surrogate servers placement, and classify them into core and edge architectures. Core CDN architectures rely on private datacenters deployment close to ISP points of presence (PoP). This has been a successful approach used by pioneers as Akamai, as well as by major content and service providers. Akamai platform [12] has been built on top of large number of small server clusters highly distributed in many different countries. Hence, such architectures require complex algorithms for locating and delivering content properly, e.g. very precise infrastructure mapping and monitoring. Some content providers, including Amazon and Google [10], and service providers, such as Limelight, have opted to deploy very expensive and large datacenters in very few strategic locations. As core architectures are connected to PoPs, they do not have control of traffic throughout ISP until the end-customer, that undermines QoS guarantees enforcement. Interoperable CDNs in edge networks have emerged to tackle directly these issues. Network service providers look forward to (i) taking advantage of their infrastructure, (ii) deploying their own datacenters, and (iii) delivering content as close as possible to end-customer. The aim is to be able to offer differentiated QoS guarantees to regular customers[1]. Another highly distributed approach of edge CDN architectures is P2P network distribution. This consists of content servers deployed on consumer-edge devices, where peers cooperate to share and distribute the content. P2P network distribution comprises video stream handlers, such as PPLive and Zattoo, and content swarming, e.g. BitTorrent, eMule, and NaDa, a distributed content distribution platform based on nanodatacenter in home gateways. NaDa relies on BitTorrent protocol to manage unused edge resources. In this work, we are mostly interested in challenges risen by edge CDN architectures.

[1]Enabling digital media content delivery: Emerging opportunities for network service providers. www.velocix.com, 2010.

Popular content: Multi-media content distribution over the internet has increased dramatically in the recent years. A recent study published by Cisco System, Inc[2] revealed that the global internet video traffic has surpassed peer-to-peer traffic since 2010, becoming the largest internet traffic type. Cisco also forecasts that internet video traffic will reach 62% of the consumer internet traffic by 2015. Many studies [8], [17] have drawn attention to reach a better understanding of internet video properties, such as popularity growth. In general, these studies point out that well-known popularity characteristics are applicable to multimedia content. For instance, they show that internet multimedia popularity distribution follows power law, time scale might vary from hours to weeks according to the media type, and that popularity bursts have a short duration and are quite likely to happen just after the content publication, especially for internet videos. But, these studies fails to define a trustful and definitive multimedia growth pattern due to the inherent unpredictability of publication, search, and promotion engines used by content providers. A way to overcome resource allocation problems caused by unpredictable multimedia growth pattern is providing adaptive replication schemes that are fit for purpose.

III. Approach

This section describes the approach used in this work. First, we present a short description of the target problem. Then, we briefly describe our evaluation scheme based on Caju. Finally, we present AREN, our adaptive replication scheme.

A. Problem statement

Consider a set of storage elements $\acute{\mathcal{J}}_o$ that stores o. Assume that is necessary to allocate at least the bandwidth b_o^j on each $j \in \acute{\mathcal{J}}_o$ in order to enforce SLA definitions properly. We focus on the dynamics of adaptive resource allocation and request scheduling for nodes of $\acute{\mathcal{J}}_o$. We particularly aim to achieve two main goals: (i) minimize the network and storage usage on edge networks, and (ii) minimize the overall number of SLA violations. We focus on the SLA violations concerning *GETs*. A *GET* is a request done by a client to retrieve a stored content.

B. Caju's architecture

To evaluate the performances of our replication scheme, AREN, we propose Caju, a tool which models a content distribution system for edge networks on top of PeerSim.

In Caju, service provider infrastructure is organized in federated storage domains, as depicted in Figure 1. A storage domain is a logical entity that aggregates a set of storage elements that are located close to each order. For instance, a storage domain might be formed of ISP storage elements that are all connected to the same digital subscriber line access multiplexer (DSLAM). These storage elements are partitioned in two different classes: (i) operator-edge, furnished by storage operators, e.g. small-sized datacenters, and (ii)

[2]Cisco visual networking and methodology, 2010-2015. www.cisco.com, 2011.

consumer-edge, those by consumers, such as set-top boxes. Edge devices contribute with their resources to cloud storage. A node per storage domain, normally from operator-edge class, plays a role of coordinator. On top of each coordinator runs a couple of services that handle request scheduling and replication for its storage domain. Coordinators interact to each other to share information about availability and location of content and resources. A detailed description of Caju design is available in [16].

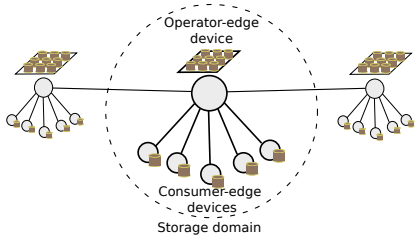

Figure 1. Storage elements and storage domains

C. Replication scheme

AREN stands for Adaptive Replication for Edge Networks scheme. AREN replication scheme relies on bandwidth reservation and collaborative caching to provide an adaptive number of replicas for popular content. AREN provides request scheduling and content replication mechanisms to minimize strict SLA violations and edge resources usage. Our replication scheme is simple and easy to implement.

Request scheduling Within a storage domain, AREN relies on the coordinator to track bandwidth reservation and selects nodes accordingly. Sources are selected to respond a request only if there is enough unreserved bandwidth. Scheduled sources contribute with the same amount of bandwidth, and cooperate to enforce SLAs by reserving bandwidth.

To enhance resource allocation in edge networks, AREN implements two simple scheduling policies.

- **Divide-and-conquer.** GET requests are served by either consumer-edge nodes or mini-datacenters. The *divide-and-conquer* scheduling policy gives priority to consumer-edge nodes and uses mini-datacenters only if there is no more spare bandwidth for reservation in the set of consumer-edge nodes of the requested object. It permits to save mini-datacenter bandwidth for creating replicas to popular content faster.
- **Nearest source selection.** We assume that intra-domain transfers are preferable. For that reason, this scheduling policy prioritizes the selection of GET sources that comes from the same Storage Domain of the request destination. That allows AREN to reduce the inter-domain traffic load.

Content replication After scheduling a request to an object, the coordinator updates its current aggregate bandwidth demand, and decides if it is worth creating a new replica in caching of the destination node. The coordinator computes the utility of a new replica based on thresholds. Replica utility measures the benefit of creating replicas with regard to popularity and current bandwidth consumption of an object. We consider two thresholds for aggregate reserved bandwidth: P_{min} and P_{max}. Our replication strategy is based on two main mechanisms:

1) Popular content classification: An object becomes popular whenever its aggregate active bandwidth is greater than a factor of the maximum threshold. For instance, consider a popularity factor Q, the threshold percentage P_{max}, and object o that has a single replica into a consumer-edge storage element of network capacity of b. Let $U(o)$ be the current bandwidth reservation for object o, o is popular if $U(o) > Q * P_{max} * b$.
2) Replica maintenance for popular content: This mechanism adapts the number of copies of popular objects regarding the thresholds and the current aggregate reserved bandwidth. It is performed whenever a GET is scheduled or periodically for maintenance purposes. New replicas are created in the GET destination when aggregate bandwidth is greater than the maximum threshold, and randomly removed when smaller than the minimum threshold.

IV. Evaluation

Our evaluation has two main goals. (i). To verify if it is reasonable to use edge devices, including operator-edge devices, to offer distributed storage service with strict QoS metrics. (ii). To evaluate the performance of our bandwidth threshold-based approach as an adaptive replication scheme. Towards these goals, we designed and implemented an evaluation scheme based on Caju that simulates our network topology and data flows among storage elements in a very precise way. It was built on top of PeerSim, a stable and extremely scalable event-driven simulation engine.

The evaluation scenario (Figure 2) includes 4002 numbered nodes, arranged across two storage domains. There are one operator-edge device (nodes 1 and 2002) and 2000 consumer-edge devices per storage domain. Storage and network capacities differ accordingly to the class of device. 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 the total amount of overall edge resources, namely additional 10% of storage capacity and only 2% of the overall network capacity. This draws our attention to the performance of replication schemes and resources allocations towards non-expensive small-sized datacenters in edge networks. We also assume that storage elements of a storage domain are connected to the same edge network, where a maximum limit of 80% is enforced to aggregate traffic. Edge networks are connected to the operator network that ensures inter-storage domain connectivity.

Workload was carefully set-up to match multimedia popular content distribution, as described in Section II. Table I lists default values for workload parameters respectively. Objects

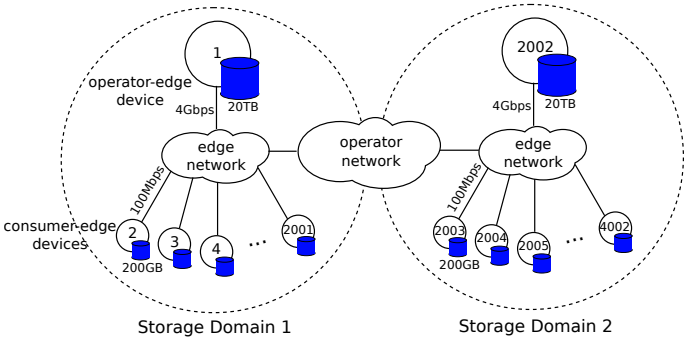

Figure 2. Evaluation scenario

Table I
DEFAULT VALUES FOR WORKLOAD PARAMETERS

Workload	
Requests per client	uniform
Experiment duration	1h 40min
Mean requests per second	400
Requests division	5% of PUTs, 95% of GETs
Object size (follows Pareto)	shape=3, smallest=26MB, biggest=1.6GB
Content popularity (Zipf-Mandelbrot)	shape=0.8, cutoff=number of objects
PUTs (Poisson)	λ=PUTs per second
Popularity growth (Weibull)	shape=2, scale $\propto$ duration

are always divided in chunks of fixed size, 2MB. SLA contracts differ to each other by transfer rate. Thus, we consider three SLA classes, in chunks per second: (a) 41, (b) 21, and (c) 14 chunks/s. Each customer has a SLA according to the following distribution: 40% class (a), 40% (b), and the remaining 20% (c). We assume that a SLA violation occurs when any transfer of a consumer does not observe her minimum contracted transfer rate.

We use emphhappiness, or number of customers without SLA violations, as a key performance metric. Along with happiness,we are interested in evaluating the amount of resources allocated to deliver content in edge networks properly. We focus on number of flows, storage, and network usage, exploring in details the resource allocation performance for the most popular content.

The rest of this section is structured as follows. Subsection IV-A describes the 4 replication schemes evaluated in this work. Subsection IV-B shows how efficient common replication schemes are for disseminating popular content in edge networks. We initially evaluate the performance of our evaluation scheme comparing two easy-to-deploy approaches. Then, Subsection IV-C shows how our replication scheme brings off challenges related to content deliver in edge networks.

A. Evaluated replication schemes

Besides AREN, we have evaluated four other replication schemes.

Uniform replication scheme with fixed number of replicas This is the simplest approach to replicate objects into a system, that is broadly used in current datacenter deployments. Given a fixed number of replicas n as a parameter, we simulate a chain of object-replication of n stages just after the initial insertion (PUT). GETs are randomly scheduled to provide load balancing. Each request is served by at most R nodes with equal load. The actual number of sources is $r = min(n, R)$.

Non-collaborative LRU caching Simple adaptive replication schemes based on non-collaborative caching, such as those that implements Least Recent Used algorithm, are easy to implement and deploy. In our evaluation, a new replica is created whenever a client, connected to a operator-edge device, performs a GET to any object. LRU replacement is enforced regarding a static percentage of the local storage capacity for caching. Request scheduling is quite similar to that of uniform approach. Initial placement requires two replicas in different devices classes of the same storage domain.

DAR The main goal of Distributed and Adaptive Replication scheme is to balance the expected bandwidth load per node. DAR algorithm intuition is replacing object replicas in the local caches based on their current transfer rate. Fresh objects replaces local cached objects with higher transfer rate, removing the highest first. We assume that there is a logically centralized coordinator that tracks and computes the latest transfer rate of any object. Since this approach was initially proposed to a P2P architecture and did not handle directly strict SLA targets, we had to sightly enhance our implementation as follows. If no object with higher transfer rate was found, but there exists stale objects, apply LRU as replacement policy. For DAR, we use exactly the same request scheduling and initial placement of LRU algorithm.

Unlimited We have made an assumption of unlimited network and storage capacities at both consumer and operator edge nodes. Source nodes reserve the strict bandwidth necessary to a transfer according to the SLA contracts. It differs from our AREN approach in two points. First, it ignores spare bandwidth, keeping bandwidth reservation value as a hard limit. Second, it avoids creating additional replicas since nodes always have enough resources.

B. Performing efficient content deliver in edge networks

We initially measure the feasibility of delivering popular content with strict SLA contracts using Caju. We compare two approaches: uniform replication with a fixed number of replicas, and non-collaborative LRU caching.

We have evaluated the required number of replicas of uniform replication for different request rates in order to prevent SLA violations. We have varied the number of replicas from one to 10. We have compared it to a non-collaborative LRU caching. We have simulated different caching sizes percentages: 1%, 5%, and 10% of the storage capacity. Figure 3 plots an initial evaluation of storage usage and *happiness* for these

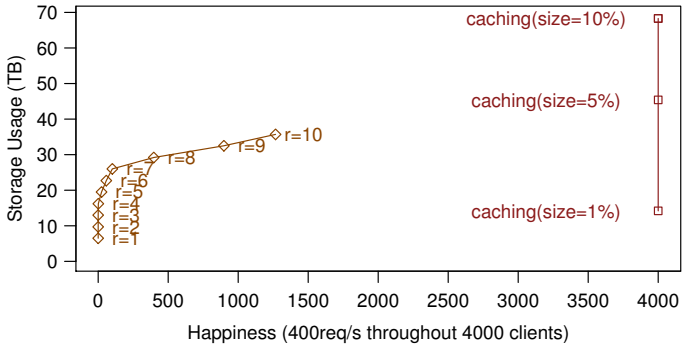

Figure 3. Happiness and storage usage with fixed number of replicas and caching.

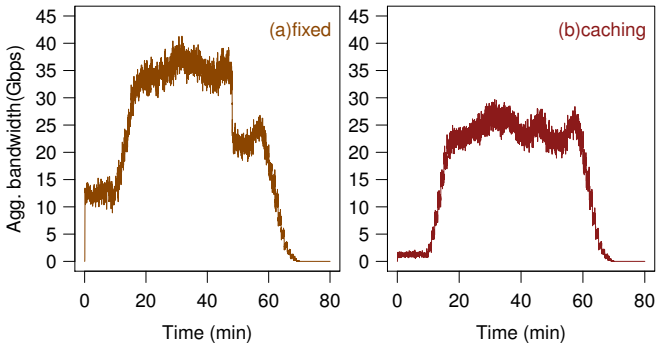

Figure 4. Aggregate bandwidth (a,b) using fixed number of replicas (r=10) and LRU caching (size=1%).

two replication schemes. Even with the smallest cache percentage of 1%, caching performs much better than uniform replication. Caching consistently improves *happiness* metric by preventing violations. It allowed us to slash violations occurrences from 2953, with uniform scheme, to 1. It required only 14.20TB, that is similar to a uniform scheme with 3 replicas, 12.98TB.

We have plotted in Figure 4 the aggregate bandwidth for caching and uniform approaches. We selected results from caching with local cache size of 1% of the node storage capacity, and uniform replication with 10 replicas. By using caching, we have been able to reduce the aggregate bandwidth by a third. We have omitted further detailed evaluations of uniform replication and caching on edge networks due to space constraints. But they are available on our technical report [16].

These results show that (i) simple caching is much more efficient in replicating popular content on edge networks than uniform approach in terms of number of SLA violations, (ii) caching allows us to reduce network resources consumption, and (iii) it permits edge node to contribute with tiny amounts of storage capacity contribution (2GB) in order to maintain enough replicas for popular content.

C. Exploring popular content delivery with AREN

Our targets in delivering popular content in edge networks are to minimize the number of SLA violations, and resources utilization. We have shown in Subsection IV-B that a non-collaborative LRU caching copes with these issues quite fairly compared to a uniform replication scheme. But an increasing demand for multimedia content, especially as VoD, might overload cloud storage systems, damaging its performance. We assume that replication schemes in edge networks should be able to adapt accordingly. Here, we present challenges raised by heavily loaded cloud storage systems, and we show how AREN uses collaborative caching and bandwidth reservation to overcomes these issues.

We compare AREN to non-collaborative LRU caching, and a collaborative caching based on DAR replication algorithm. All schemes were configured to use a cache size equal to 1% of the local storage capacity and consumers are able to GET objects by setting up transfers from up to five different sources, according to number of available replicas. Chunks size remains unchanged. We have set up AREN to enforce bandwidth reservation, and we chose the minimum threshold percentage to 5% and the maximum one to 30%.To simulate higher workload loads, we slightly modified the default values of object size distribution from Table I, by changing the shape and lower bound (smallest) parameter of Pareto distribution.

We initially show in Figure 5 the *happiness* measurements when the mean object size increases. DAR and LRU caching approaches perform poorly in higher loads, while AREN is resilient to load increasing. Overall, *happiness* falls sharply when the workload's mean object size increases, except for AREN replication scheme. Under the heaviest load, mean object size equal to 140MB, we observed *happiness* metric equal to 3949 for AREN, versus 2 for caching and 1 for DAR. While AREN suffered only 51 violations, caching and DAR suffered 27539 and 30071 respectively. For the remaining evaluations, we assume 140MB as the default mean object size.

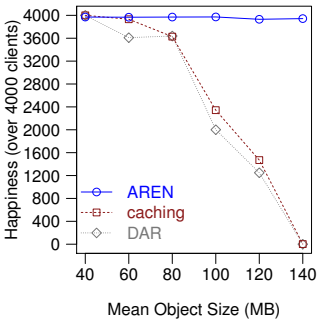

Figure 5. *happiness* for big loads.

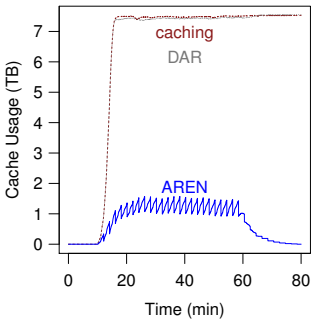

Figure 6. Replication's footprint.

We consider that edge nodes' primary goal is not to provide cloud storage. Therefore storage usage has to be minimized as much as possible. Since all schemes of this Subsection perform the same initial placement, storage usage differs exclusively in cache usage for replication. Figure 6 shows the storage usage by replicas . AREN scheme provides roughly a sevenfold decrease in the amount of cache usage compared to DAR and LRU caching approaches. AREN performs better than both DAR and LRU caching approaches because it

creates new replicas for popular content only, and it is able to remove unnecessary replicas thanks to the AREN's coordinator role in monitoring and tracking aggregate bandwidth reservation. DAR and LRU caching have similar results due to a unwanted behaviour of our DAR implementation for delivering popular content, detailed in Subsection IV-A. Our enhanced DAR implementation reduces SLA violations, but increases the storage usage.

Since network capacity is a scarce resource in edge networks for cloud storage, an efficient replication scheme must minimize the bandwidth usage on operator-edge storage elements, so-called mini-datacenters. We present the aggregate bandwidth usage for non-collaborative LRU caching, and AREN schemes in Figures 7, and 8, respectively. AREN scheme reduces roughly 50% of aggregate upload bandwidth. AREN performs better thanks to *divide-and-conquer* policy, described in Subsection III-C, that prioritizes consumer-edge nodes as GET requests' sources. Instead, non-collaborative LRU caching and DAR schemes rely on pure random scheduling that overloads mini-datacenters upload link. Aggregate download bandwidth has the same level for all two schemes because a common initial placement policy requires the primary copy to be stored in a mini-datacenter. With our DAR implementation, we have found results quite similar to caching.

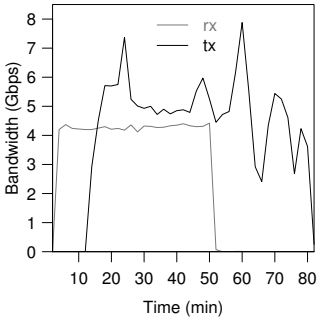

Figure 7. Bandwidth utilization of LRU caching scheme on operator-edge storage elements.

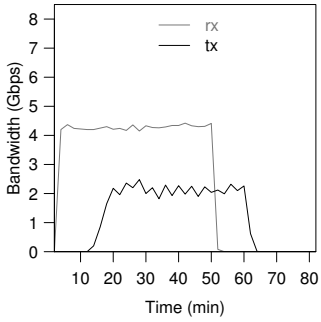

Figure 8. Bandwidth utilization of AREN scheme on operator-edge storage elements.

Bandwidth in edge networks have to be allocated wisely. The request scheduling must handle requests properly in order to reduce the traffic burden, particularly for the most popular content. Figure 9 shows a box plot, including minimum value, fist quartile, median, third quartile, whisker and outliers values, from upper quantile of GET durations of the 10 most popular objects. These objects account for 1.5% of the all GET requests. AREN presents the smallest degree of dispersion amongst the evaluated replication schemes. That happens because AREN scheme is able to better schedule GET requests through bandwidth reservation, avoiding that GET request for popular content either last too long, causing violations, or too short, wasting network resources. Overall, we have verified that 99% of all violations with caching last at least 4.64 seconds, with outliers up to 48 minutes. Since a straightforward implementation of non-collaborative caching relies on random scheduling, it lacks essential information for preventing edge nodes' overloading, and therefore provides poor resource allocation for popular content.

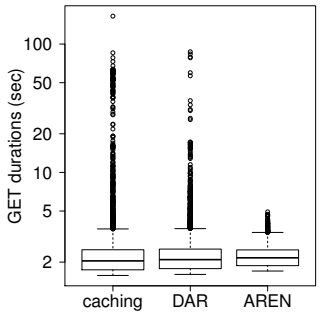

Figure 9. Upper quartile of GET durations of the 10 most popular objects.

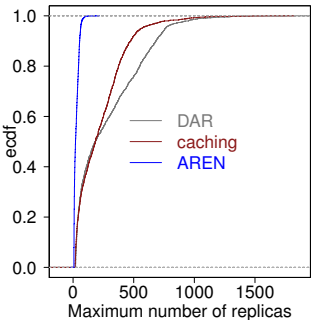

Figure 10. Maximum number of replicas for the 2% most replicated objects.

We analyzed the number of replicas of the most popular content. Figure 10 plots the maximum number of replicas for the 2% most replicated objects. It shows that the vast majority of the content had a small number of copies. For instance, 98% of objects with DAR scheme had less than 21 copies. Although, we are able to drop this number by two-thirds with AREN scheme. Our replication scheme performs still better for the most replicated content. While the maximum number of replicas using DAR and LRU reached respectively 1574 and 1740, AREN's most replicated object had only 188 replicas. This means that AREN adapts replication efficiently for the most popular content. It reduces the amount of required storage space for additional replicas, as well as the number of allocated edge nodes per stored object.

We have evaluated the bandwidth allocation efficiency of AREN, caching, and DAR, in terms of aggregate bandwidth and bandwidth allocation variance during the peak of utilization, and we have compared their results to an assumption of unlimited network and storage capacities (described in Subsection IV-A). Figure 11 shows that AREN's bandwidth allocation through reservation is similar to the unlimited assumption. That allows us to increase the aggregate bandwidth by almost 20%, hence achieving faster content delivery. DAR and LRU weaken the system's performance due to their fair sharing bandwidth allocation policy. Measurements of bandwidth allocation variance, Figure 12, show highest variance values for unlimited scheme. Alongside AREN, second highest values, it shows that imbalance in bandwidth allocation per node matters to deliver popular content with strict SLA. We have seen the higher the imbalance in bandwidth is, the better is the network resource allocation.

In order to control the impact of storage traffic in edge networks, transfers between nodes from different storage domains must be avoided as much as possible. AREN scheme enforces a *nearest source selection* scheduling policy, described in Subsection III-C, which prioritizes the selection of intra-domain sources for requests. That allows us to reduce significantly the inter-domain traffic burden compared

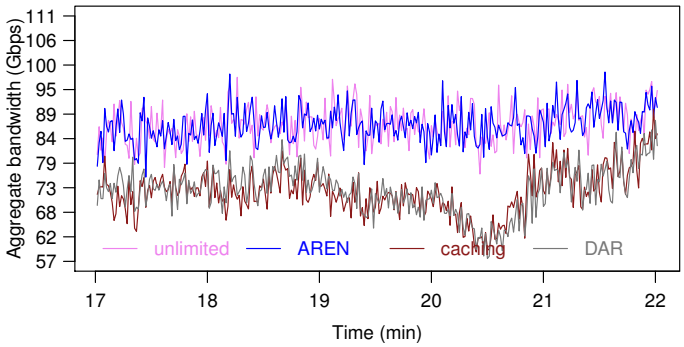

Figure 11. Aggregate bandwidth for three replication schemes and an assumption of unlimited resources, during utilization peak.

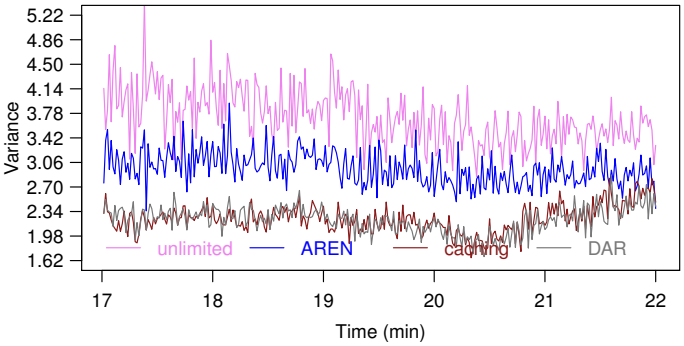

Figure 12. Variance for three replication schemes and an assumption of unlimited resources during utilization peak.

to a pure random scheduling. Figure 13 and 14 plot the aggregate bandwidth exchanged between the two storage domains. The enforcement of our straightforward policy in AREN scheme reduces nearly 60% of the overall traffic inter-storage domains compared to non-collaborative LRU caching. Our DAR scheme implementation performed very similar to caching.

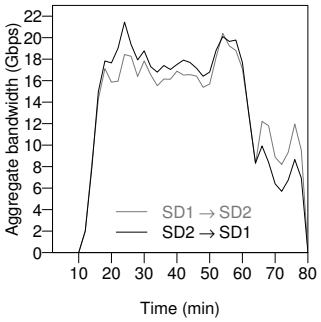

Figure 13. Aggregate bandwidth between storage domain 1 and 2 for caching.

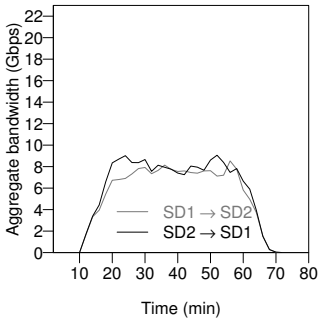

Figure 14. Aggregate bandwidth between storage domain 1 and 2 from AREN replication scheme.

V. Related work

Our related work is organized in two parts: content replication and QoS guarantees for content delivery.

Content replication: A large number of replication schemes have been proposed in the recent years, particularly for P2P networks. Broadly speaking, these schemes fall into three categories according to their resource allocation strategies: uniform, proportional or adaptive replication schemes. The Google File System (GFS) [9] and Ceph [19] adopt a pragmatic approach where the number of replicas per object is uniform and fixed. This approach has had a considerable success in the industry, particularly for datacenters deployment, because it is easy to adopt. However, it relies on overprovision to provide enough resources for popular content, and despite of using commodity servers, it is inefficient and quite expensive. We have verified in this work that a simple non-collaborative LRU cache outperforms uniform replication schemes. Cohen *et al.* [5] initially suggested that storage capacity and bandwidth must be taken into account to enhance proportional replication algorithms, but their aim was limited to minimize the expected search size in unstructured P2P networks. Adya *et al.* [1] and On *et al.* [13] propose a interesting proportional replication schemes based on availability of untrusted storage nodes. Although they fail to state replication in terms of response time, storage and bandwidth capacity, that are primary issues for consumer-edge devices. Carbonite [4] extends these concepts and introduces an adaptive replication scheme that takes into account both availability (for GETs) and durability (for PUTs) of stored objects. On the one hand it shows how much bandwidth usage is important for replication schemes, but on the other hand their assumptions are based on a very idealized mathematical model, that ignores object popularity and node overhead. We have shown with AREN that tracking popularity growth and load over nodes are essential for enforcing strict SLA contracts. EAD [15] and Skute [3] tackle these issues 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. Skute provides a replication management scheme that evaluates replicas price and revenue across different geographic locations. Skute's evaluation technique relies on equilibrium analysis of data placement. Despite being highly scalable and providing an efficient framework for replication in distributed systems, these approaches result in inaccurate transfer rate allocation, hence inappropriate for high-quality content delivery. AREN overcomes these issues by combining bandwidth reservation and collaborative caching successfully. More recently, Zhou *et al.* proposed DAR [21], an adaptive replication algorithm for P2P-assisted Video-on-Demand (VoD). DAR permits distributing content in scalable way by balancing the expect bandwidth load per node. But, we have seen that DAR approach performs worse than caching to enforce strict QoS metrics. AREN provides proper bandwidth imbalance to prevent SLA violations and improve resource allocation.

QoS guarantees for content delivery: The increasing competition between network service providers, along with ever-growing demand for multimedia content, push through tighter delivery guarantees. Consumers and providers engage in Service Level Agreement (SLA), that formally establishes which system performance is expected for a particular service.

While response time and mean request rate are commonly included in current negotiated contracts [6], high-quality metrics, such as end-to-end latency and strict transfer rate, are avoided by providers because they are very tough to enforce. Evans *et al.* [7] show that appropriate engineering of edge networks is key for tight SLA. Integrated Services architecture (InServ) with Resource reSerVation Protocol (RSVP) guarantees QoS metrics by reserving end to end resources before sending any data, but suffers from poor scalability. Differentiated Service (DiffServ) groups distinct traffic flows in classes and configures routers to correctly follow a Per Hop Behaviour (PHB) without resource reservation. However, it lacks proper end to end QoS enforcement due to PHB mismatches across different ASes. Recently, researchers have extensively studied this problem in datacenters networks [20], [2], [18]. D3 [20], provides a deadline-aware protocol that uses explicit rate control to apportion bandwidth according to flow deadline. That allows to increase the aggregate throughput in datacenter environments compared to TCP. But D3 and previous proposals are particularly designed for transporting tiny objects from homogeneous nodes across datacenter Ethernets with very low delay and high throughput, and furthermore they are agnostic to popularity peaks, and they are not customized for wide area network environments with unpredictable transfer rate demands and high variances in network latency. The bandwidth reservation used by AREN is strongly based on D3 findings. Performance evaluations of AREN show that combining bandwidth reservation and collaborative caching mechanisms is essential to enforce transfer rates as QoS metrics in edge networks.

VI. Conclusions

This work presented AREN, an novel adaptive replication scheme for cloud storage in edge networks that enforces strict SLA metrics with efficient resource allocation. AREN minimizes the number of SLA violations by tracking bandwidth reservation mechanism on edge nodes for operating collaborative caching mechanism properly. Our evaluations show that AREN consistently outperforms common replication schemes. For future work, we will investigate adaptive replication schemes' mechanisms to cope with unpredictable popularity growth patterns of web content, by validating our simulations with real traces.

References

[1] A. Adya, W. Bolosky, M. Castro, R. Chaiken, G. Cermak, J. Douceur, J. Howell, J. Lorch, M. Theimer, and R. Wattenhofer. Farsite: Federated, available, and reliable storage for an incompletely trusted environment. In *OSDI*, 2002.

[2] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan. Data center tcp (dctcp). In *SIGCOMM*, 2010.

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

[4] B.-G. Chun, F. Dabek, A. Haeberlen, E. Sit, H. Weatherspoon, F. Kaashoek, J. Kubiatowicz, and R. Morris. Efficient replica maintenance for distributed storage systems. In *NSDI*, 2006.

[5] Edith Cohen and Scott Shenker. Replication strategies in unstructured peer-to-peer networks. In *SIGCOMM*, 2002.

[6] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: amazon's highly available key-value store. In *SIGOPS*, 2007.

[7] J. Evans and C. Filsfils. Deploying diffserv at the network edge for tight slas, part 1. *Internet Computing, IEEE*, 2004.

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

[9] S. Ghemawat, H. Gobioff, and S.-T. Leung. The google file system. In *SOSP*, 2003.

[10] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. The flattening internet topology: Natural evolution, unsightly barnacles or contrived collapse? In *PAM*. 2008.

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

[12] E. Nygren, R. K. Sitaraman, and J. Sun. The akamai network: a platform for high-performance internet applications. *SIGOPS*, 2010.

[13] G. On, J. Schmitt, and R. Steinmetz. Quality of availability: Replica placement for widely distributed systems. In *IWQoS*, 2003.

[14] M. Pathan and R. Buyya. A taxonomy of cdns. In *Content Delivery Networks*. Springer Berlin Heidelberg, 2008.

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

[16] G. Silvestre, S. Monnet, R. Krishnaswamy, and P. Sens. Caju: a content distribution system for edge networks. Technical Report 8006, 2012.

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

[18] V. Vasudevan, A. Phanishayee, H. Shah, E. Krevat, D. G. Andersen, G. R. Ganger, G. A. Gibson, and B. Mueller. Safe and effective fine-grained tcp retransmissions for datacenter communication. In *SIGCOMM*, 2009.

[19] S. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn. Ceph: A scalable, high-performance distributed file system. In *OSDI*, 2006.

[20] C. Wilson, H. Ballani, T. Karagiannis, and A. Rowstron. Better never than late: Meeting deadlines in datacenter networks. In *SIGCOMM*, 2011.

[21] Y. Zhou, T. Z. J. Fu, and D. M. Chiu. A unifying model and analysis of p2p vod replication and scheduling. In *INFOCOM*, 2012.