Quality-Assured Cloud Bandwidth Auto-Scaling for Video-on-Demand Applications

Di Niu, Hong Xu, Baochun Li
Department of Electrical and Computer Engineering
University of Toronto

Shuqiao Zhao
Multimedia Development Group
UUSee, Inc.

***Abstract*—There has been a recent trend that video-on-demand (VoD) providers such as Netflix are leveraging resources from cloud services for multimedia streaming. In this paper, we consider the scenario that a VoD provider can make reservations for bandwidth guarantees from cloud service providers to guarantee the streaming performance in each video channel. We propose a predictive resource auto-scaling system that dynamically books the minimum bandwidth resources from multiple data centers for the VoD provider to match its short-term demand projections. We exploit the anti-correlation between the demands of video channels for statistical multiplexing and for hedging the risk of under-provision. The optimal load direction from channels to data centers is derived with provable performance. We further provide suboptimal solutions that balance bandwidth and storage costs. The system is backed up by a demand predictor that forecasts the demand expectation, volatility and correlations based on learning. Extensive simulations are conducted driven by the workload traces from a commercial VoD system.**

I. Introduction

Cloud computing is redefining the way many Internet services are operated and provided, including Video-on-Demand (VoD). Instead of buying racks of servers and building private data centers, it is now feasible for VoD companies to use computing and bandwidth resources of cloud service providers. As an example, Netflix moved its streaming servers, encoding software, data stores and other customer-oriented APIs to Amazon Web Services (AWS) in 2010 [1].

One of the most important economic appeals of cloud computing is its elasticity and auto-scaling in resource provisioning. Traditionally, after careful capacity planning, an enterprise makes long-term investments on its infrastructure to accommodate its peak workload. Over-provisioning is inevitable while utilization remains low during most non-peak times. In contrast, in the cloud, the number of computing instances launched can be changed adaptively at a fine granularity with a lead time of minutes. This converts the up-front infrastructure investment to operating expenses charged by cloud providers. As the cloud's auto-scaling ability enhances resource utilization by matching supply with demand, overall expenses of the enterprise may be reduced.

Unlike web servers or scientific computing, VoD is a network-bound service with stringent bandwidth requirements. As VoD users must download at a rate no smaller than the video playback rate to smoothly watch video streams online, bandwidth, as opposed to storage and computation, constitutes the performance bottleneck. Yet, a major obstacle that prevents numerous VoD providers from embracing cloud computing is that, unlike CPU and memory resources, a *guarantee of bandwidth* is not provided in current cloud services. Instead, each data center has limited outgoing bandwidth shared by multiple tenants with no bandwidth assurance.

We believe that bandwidth reservation will become a near-term value-added feature offered by cloud services to appeal to customers with bandwidth-intensive applications, such as VoD. In fact, there have already been proposals from the perspective of data center engineering to offer bandwidth guarantees for egress traffic from a virtual machine (VM), as well as among VM themselves [2], [3].

Under such a context, we analyze the benefits and address open challenges of cloud bandwidth auto-scaling for VoD applications in this paper. In a nutshell, the benefits of bandwidth auto-scaling are intuitive. As shown in Fig. 1(a), traditionally, a VoD provider acquires a monthly plan from ISPs, in which a fixed bandwidth capacity, e.g., 1 Gbps, is guaranteed to accommodate the anticipated peak demand. As a result, resource utilization is low during non-peak times of demand troughs. Alternatively, a usage-based pay-as-you-go model is adopted by a cloud as shown in Fig. 1(b), where a VoD provider pays for the total amount of bytes transferred. However, the bandwidth capacity available to the VoD provider is subject to variation due to contention from other applications, incurring unpredictable performance issues. Fig. 1(c) illustrates bandwidth auto-scaling and reservation to match resource with the demand, leading to both high resource utilization and quality guarantees. Apparently, the more frequently the rescaling happens, the more closely resource supply will match the demand.

However, a number of important challenges need to be addressed to achieve bandwidth auto-scaling for a VoD provider. *First*, since resource rescaling requires a delay of at least several minutes to update configuration and launch instances, it is best to predict the demand with a lead time greater than the update interval, and scale the capacity to meet anticipated demand. Such a proactive, rather than passive, strategy for resource provisioning needs to take into account demand fluctuations in order to avoid bandwidth insufficiency. *Second*, as statistical multiplexing can smooth traffic, a VoD provider can reserve less bandwidth to guard against fluctuations by *jointly* reserving bandwidth for all its video channels. However, to serve geographically distributed end users, a VoD provider

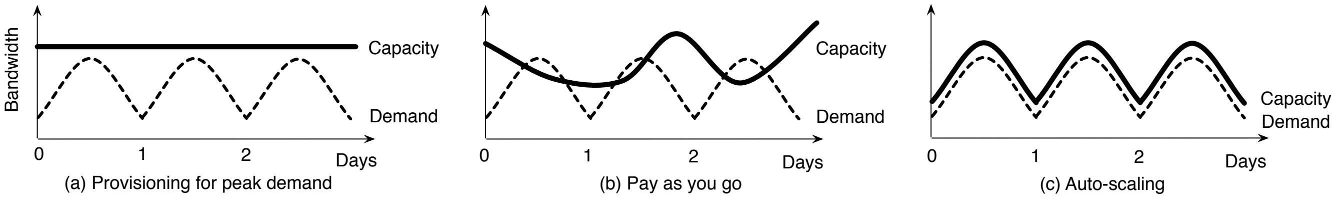


Fig. 1. Bandwidth auto-scaling with quality assurance, as compared to provisioning for the peak demand and pay-as-you-go.

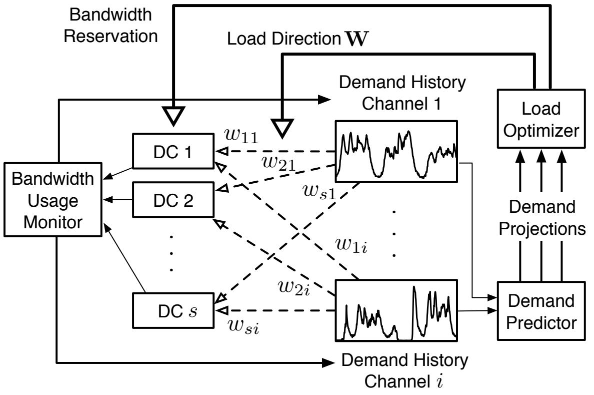


Fig. 2. The system decides the bandwidth reservation from each data center and a matrix $\mathbf{W} = [w_{si}]$ every Δt minutes, where w_{si} is the proportion of video channel i's requests directed to data center s. DC: data center.

usually has its collection of channels served by multiple data centers, which are possibly owned by different cloud providers. The question is — how should the VoD provider optimally split and direct its workload across data centers to save the overall bandwidth reservation cost?

In this paper, we propose a bandwidth auto-scaling facility that dynamically reserves resources from multiple data centers for VoD providers, with several distinct features. *First*, it is predictive. The facility tracks the history of bandwidth demand in each video channel using cloud monitoring services, and periodically estimates the expectation, volatility and correlations of demands in all video channels for the near future using time-series techniques. We propose a novel *channel interleaving scheme* that can even predict demand for new videos that lack historical demand data. *Second*, it provides quality assurance by judiciously deciding the minimum bandwidth reservation needed to satisfy the demand with high probability. *Third*, it optimally mixes demands based on anti-correlation to save the aggregate bandwidth capacity reserved from all data centers, while confining risks of under-provision.

We formulate the bandwidth minimization problem given the predicted demand statistics as input, derive the theoretically optimal load direction across data centers, and propose heuristic solutions that balance bandwidth and storage costs. The proposed facility is evaluated through extensive trace-driven simulations based on a large data set of 1693 video channels collected from UUSee, a production VoD system, over a 21-day period.

II. System Architecture

Consider a VoD provider with N video channels, relying on S data centers for service, which are possibly owned by different cloud service providers. We propose an unobtrusive auto-scaling system that draws beliefs about future demands of all channels and reserves minimal resources from multiple data centers to satisfy the demand. Our system architecture is shown in Fig. 2, which consists of three key components: *bandwidth usage monitor*, *demand predictor* and *load optimizer*. Bandwidth rescaling happens proactively every Δt minutes, with the following three steps:

First, before time t, the system collects bandwidth demand history of all channels up to time t, which can easily be obtained from cloud monitoring services. As an example, Amazon CloudWatch provides a free resource monitoring service to AWS customers at a 5-minute frequency [4].

Second, the bandwidth demand history of all channels is fed into the demand predictor to predict bandwidth requirement of each video channel in the next Δt minutes, i.e., in the period $[t, t + \Delta t)$. Our predictor not only forecasts the expected demand, but also outputs a *volatility* estimate, which represents the degree that demand will be fluctuating around its expectation, as well as the demand *correlations* between different channels in this period. Our volatility and correlation estimation is based on multivariate GARCH models [5], which gained success in stock modeling in the past decade.

Finally, the load optimizer takes predicted statistics as the input, and calculates the bandwidth capacity to be reserved from each data center. It also outputs a load direction matrix $\mathbf{W} = [w_{si}]$, where w_{si} represents the portion of video channel i's workload directed to data center s. Apparently, we should have $\sum_s w_{si} = 1$ if the aggregate data center capacity is sufficient. The matrix $\mathbf{W}$ also indicates the content placement decision: video i is replicated to data center s only if $w_{si} > 0$. In practice, the load direction $\mathbf{W}$ can be readily implemented by routing the requests for video channel i to data center s with probability w_{si}.

The system finishes the above three steps before time t, so that a new bandwidth reservation can be performed at time t for the period $[t, t + \Delta t)$, after which the above process is repeated for the next period $[t + \Delta t, t + 2\Delta t)$.

Bandwidth Reservation vs. Load Balancing. One may be tempted to think that periodic bandwidth reservation is unnecessary, since requests can be flexibly directed to whichever data center that has available capacity by a load balancer. However, the latter will exactly fall in the range of pay-as-you-go model with no quality guarantee to VoD users, whereas bandwidth reservation ensures that the provisioned resource exceeds the projected demand with high probability.

Furthermore, a major difficulty of load balancing is that

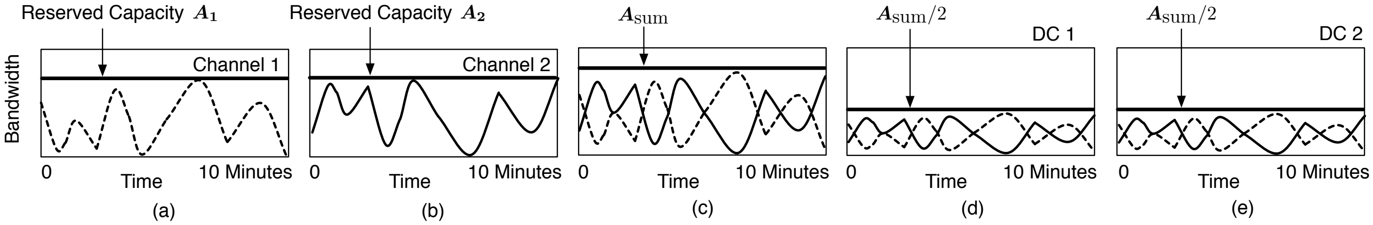


Fig. 3. Using demand correlation between channels, we can save the total bandwidth reservation, even within each 10-minute period, while still providing quality assurance to each channel. DC: data center.

data centers might be owned by different cloud providers. As a result, it is complicated if not infeasible to implement a gateway that can continuously watch resources of each cloud provider (instead of every 5 minutes as offered by free cloud watch services) and redirect requests instantaneously across cloud providers. Even if requests can be redirected instantaneously to lightly loaded data centers when the playback quality degrades, significant engineering efforts are required to monitor the video playback quality at end users.

Quality-Assured Bandwidth Multiplexing. The bandwidth demand of each video channel can fluctuate drastically even at small time scales. To avoid performance risks, the bandwidth reservation made for each channel in each Δt period should accommodate such fluctuations, inevitably leading to low utilization at troughs, as illustrated in Fig. 3(a) and (b). Trough filling within a short period such as 10 minutes is hard with too many random shocks in demand.

However, our load optimizer strives to enhance utilization even when Δt is as small as 10 minutes by multiplexing demands based on their correlations. The usefulness of anti-correlation is illustrated in Fig. 3(c): if we jointly book capacity for two negatively correlated channels, the total reserved capacity is $A_{\mathsf{sum}} < A_1 + A_2$. Besides aggregation, we can also take a part of demand from each channel, mix them and reserve bandwidth for the mixed demands from multiple data centers. As an example, in Fig. 3(d) and (e), the aggregate demand of two channels is split into two data centers, each serving a mixture of demands, which still leads to a total bandwidth reservation of A_{sum}. In each Δt period, we leverage the estimated demand correlations to optimally direct workloads across data centers so that the total bandwidth reservation necessary to guarantee quality is minimized.

III. Load Direction and Bandwidth Reservation

In this section, we focus on the load optimizer. Suppose before time t, we have obtained the estimates about demands in the coming period $[t, t+\Delta t)$. Our objective is to decide load direction $\mathbf{W}$ so as to minimize the total bandwidth reservation while controlling the under-provision risk in each data center. The question of how to make demand predictions will be the subject of Sec.IV.

We first introduce a few useful notations. Since we are considering an individual time period, without loss of generality, we drop subscript t in our notations. Recall that the VoD provider runs N video channels. The bandwidth demand of channel i is a random variable D_i with mean μ_i and variance σ_i^2. For convenience, let $\mathbf{D} = [D_1, \ldots, D_N]^\mathsf{T}$, $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_N]^\mathsf{T}$ and $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_N]^\mathsf{T}$.

Note that the random demands $D_1, \ldots, D_N$ may be highly correlated due to the correlation between video genres, viewer preferences and video release times. Denote ρ_{ij} the correlation coefficient of D_i and D_j, with $\rho_{ii} \equiv 1$. Let $\boldsymbol{\Sigma} = [\sigma_{ij}]$ be the $N \times N$ symmetric demand *covariance matrix*, with $\sigma_{ii} = \sigma_i^2$ and $\sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$ for $i \neq j$.

The VoD provider will book resources from S data centers. Denote C_s the upper bound on the bandwidth capacity that can be reserved from data center s, for $s = 1, \ldots, S$. C_s may be limited by the available instantaneous outgoing bandwidth at data center s, or may be intentionally set by the VoD provider to geographically spread its workload and avoid booking resources from a single data center. Let $C_{\mathsf{sum}} = \sum_s C_s$ be the aggregate utilizable bandwidth capacity of all S data centers. Throughout the paper, we assume that C_{sum} is sufficiently large to satisfy all the demands in the system.[1]

We define a load direction decision as a weight matrix $\mathbf{W} = [w_{si}]$, $s = 1, \ldots, S$, $i = 1, \ldots, N$, where w_{si} represents the portion of video i's demand directed to and served by data center s, with $0 \leq w_{si} \leq 1$ and $\sum_s w_{si} = 1$. We observe that $\mathbf{w}_s = [w_{s1}, \ldots, w_{sN}]^\mathsf{T}$ represents the *workload portfolio* of data center s. Given $\mathbf{w}_s$, the aggregate bandwidth load imposed on data center s is a random variable

$$L_s = \textstyle\sum_i w_{si} D_i = \mathbf{w}_s^\mathsf{T}\mathbf{D}. \tag{1}$$

We use A_s to denote the amount of bandwidth reserved from data center s for this period. Clearly, we must have $A_s \leq C_s$. Let $\mathbf{A} =: [A_1, \ldots, A_S]^\mathsf{T}$. To control the under-provision risk, we require the load imposed on data center s to be no more than the reserved bandwidth A_s with high probability, i.e.,

$$\Pr(L_s > A_s) \leq \epsilon, \quad \forall s, \tag{2}$$

where $\epsilon > 0$ is a small constant, called the *under-provision probability.*

A. The Optimal Load Direction

Given demand expectations $\boldsymbol{\mu}$ and covariances $\boldsymbol{\Sigma}$, and the available capacities $C_1, \ldots, C_S$, the load optimizer can decide the optimal bandwidth reservation $\mathbf{A}^*$ and load direction $\mathbf{W}^*$

[1] A rigorous condition for supply exceeding demand is given in Theorem 1.

by solving the following optimization problem:

$$\min_{\mathbf{W},\mathbf{A}} \textstyle\sum_s A_s \tag{3}$$

$$\text{s.t.} \quad A_s \le C_s, \quad \forall s, \tag{4}$$

$$\Pr(L_s > A_s) \le \epsilon, \quad \forall s, \tag{5}$$

$$\textstyle\sum_s w_{si} = 1, \quad \forall i. \tag{6}$$

Through reasonable aggregation, we believe that L_s follows a Gaussian distribution. We will empirically justify this assumption in Sec. V using real-world traces. When L_s is Gaussian-distributed, constraint (2) is equivalent to

$$A_s \ge \mathbf{E}[L_s] + \theta\sqrt{\mathbf{var}[L_s]}, \quad \text{with } \theta := F^{-1}(1-\epsilon), \tag{7}$$

where $F(\cdot)$ is the CDF of normal distribution $\mathcal{N}(0,1)$. For example, when $\epsilon = 2\%$, we have $\theta = 2.05$. Since

$$\mathbf{E}[L_s] = \mu_1 w_{s1} + \ldots + \mu_N w_{sN} = \boldsymbol{\mu}^\mathsf{T}\mathbf{w}_s,$$
$$\mathbf{var}[L_s] = \textstyle\sum_{i,j} \rho_{ij}\sigma_i\sigma_j w_{si} w_{sj} = \mathbf{w}_s^\mathsf{T}\boldsymbol{\Sigma}\mathbf{w}_s,$$

it follows that (2) is equivalent to

$$A_s \ge \boldsymbol{\mu}^\mathsf{T}\mathbf{w}_s + \theta\sqrt{\mathbf{w}_s^\mathsf{T}\boldsymbol{\Sigma}\mathbf{w}_s}. \tag{8}$$

Therefore, the bandwidth minimization problem (3) is now converted to

$$\min_{\mathbf{W}} \textstyle\sum_s A_s \tag{9}$$

$$A_s = \boldsymbol{\mu}^\mathsf{T}\mathbf{w}_s + \theta\sqrt{\mathbf{w}_s^\mathsf{T}\boldsymbol{\Sigma}\mathbf{w}_s}, \tag{10}$$

$$\text{s.t.} \quad \boldsymbol{\mu}^\mathsf{T}\mathbf{w}_s + \theta\sqrt{\mathbf{w}_s^\mathsf{T}\boldsymbol{\Sigma}\mathbf{w}_s} \le C_s, \quad \forall s, \tag{11}$$

$$\textstyle\sum_s \mathbf{w}_s = \mathbf{1}, \tag{12}$$

$$\mathbf{0} \preceq \mathbf{w}_s \preceq \mathbf{1}, \quad \forall s, \tag{13}$$

where $\mathbf{1} = [1,\ldots,1]^\mathsf{T}$ and $\mathbf{0} = [0,\ldots,0]^\mathsf{T}$ are N-dimensional column vectors. We can derive nearly closed-form solutions to problem (9) in the following theorem:

Theorem 1: When $C_{\mathsf{sum}} \ge \boldsymbol{\mu}^\mathsf{T}\mathbf{1} + \theta\sqrt{\mathbf{1}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{1}}$, an optimal load direction matrix $[w_{si}^*]$ is given by

$$w_{si}^* = \alpha_s, \quad \forall i, \quad s = 1,\ldots,S, \tag{14}$$

where $\alpha_1,\ldots,\alpha_S$ can be *any solution* to

$$\sum_s \alpha_s = 1,\ 0 \le \alpha_s \le \min\left\{1, \frac{C_s}{\boldsymbol{\mu}^\mathsf{T}\mathbf{1} + \theta\sqrt{\mathbf{1}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{1}}}\right\},\ \forall s. \tag{15}$$

When $C_{\mathsf{sum}} < \boldsymbol{\mu}^\mathsf{T}\mathbf{1} + \theta\sqrt{\mathbf{1}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{1}}$, there is no feasible solution that satisfies constraints (11) to (13).

Proof Sketch: First, $f(\mathbf{w}_s) = \sqrt{\mathbf{w}_s^\mathsf{T}\boldsymbol{\Sigma}\mathbf{w}_s}$ is a cone and thus a convex function. Hence, $f[(\mathbf{w}_1 + \mathbf{w}_2)/2] \le [f(\mathbf{w}_1) + f(\mathbf{w}_2)]/2$, or equivalently,

$$\sqrt{(\mathbf{w}_1 + \mathbf{w}_2)^\mathsf{T}\boldsymbol{\Sigma}(\mathbf{w}_1 + \mathbf{w}_2)} \le \sqrt{\mathbf{w}_1^\mathsf{T}\boldsymbol{\Sigma}\mathbf{w}_1} + \sqrt{\mathbf{w}_2^\mathsf{T}\boldsymbol{\Sigma}\mathbf{w}_2}.$$

By induction, we can prove

$$\textstyle\sum_s \sqrt{\mathbf{w}_s^\mathsf{T}\boldsymbol{\Sigma}\mathbf{w}_s} \ge \sqrt{(\sum_s \mathbf{w}_s^\mathsf{T})\boldsymbol{\Sigma}(\sum_s \mathbf{w}_s)}. \tag{16}$$

If $\sum_s \mathbf{w}_s = \mathbf{1}$ is feasible, by (11) and (16) we have

$$\textstyle\sum_s C_s \ge \boldsymbol{\mu}^\mathsf{T}\mathbf{1} + \theta\sqrt{(\sum_s \mathbf{w}_s^\mathsf{T})\boldsymbol{\Sigma}(\sum_s \mathbf{w}_s)} = \boldsymbol{\mu}^\mathsf{T}\mathbf{1} + \theta\sqrt{\mathbf{1}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{1}}.$$

If $\sum_s C_s \ge \boldsymbol{\mu}^\mathsf{T}\mathbf{1} + \theta\sqrt{\mathbf{1}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{1}}$, it is easy to verify (15) is feasible. When $w_{si}^* = \alpha_s$ given by (14), we find (11), (12) and (13) are all satisfied. Hence, (14) is a feasible solution and $\sum_s \mathbf{w}_s = \mathbf{1}$ is feasible. By (16), the objective (9) satisfies

$$\begin{aligned} &\textstyle\sum_s(\boldsymbol{\mu}^\mathsf{T}\mathbf{w}_s + \theta\sqrt{\mathbf{w}_s^\mathsf{T}\boldsymbol{\Sigma}\mathbf{w}_s}) \\ &\ge \boldsymbol{\mu}^\mathsf{T}\textstyle\sum_s \mathbf{w}_s + \theta\sqrt{(\sum_s \mathbf{w}_s^\mathsf{T})\boldsymbol{\Sigma}(\sum_s \mathbf{w}_s)} \\ &= \boldsymbol{\mu}^\mathsf{T}\mathbf{1} + \theta\sqrt{\mathbf{1}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{1}}. \end{aligned}$$

We find that $[w_{si}^*]$ given by (14) achieves the above inequality with equality, and thus is also an optimal solution to (9). □

Theorem 1 implies that in the optimal solution, each video channel should split and direct its workload to S data centers following the same weights $\alpha_1,\ldots,\alpha_S$, which can be found by solving the linear constraints (15). Moreover, the optimal workload portfolio of each data center s has a similar structure of $\mathbf{w}_s = \alpha_s\mathbf{1}$, where α_s depends on its available capacity C_s through the constraints (15).

Under the optimal load direction, the aggregate bandwidth reservation reaches its minimum value:

$$\textstyle\sum_s A_s^* = \sum_s(\boldsymbol{\mu}^\mathsf{T}\mathbf{w}_s^* + \theta\sqrt{\mathbf{w}_s^{*\mathsf{T}}\boldsymbol{\Sigma}\mathbf{w}_s^*}) = \boldsymbol{\mu}^\mathsf{T}\mathbf{1} + \theta\sqrt{\mathbf{1}^\mathsf{T}\boldsymbol{\Sigma}\mathbf{1}},$$

which does not depend on S, the number of data centers. This means that having demand served by multiple data centers instead of one big data center does not increase bandwidth reservation cost as long as $w_{si} = \alpha_s, \forall i$ given by (14). Therefore, the load optimizer can first aggregate all the demands and then split the aggregated demand into different data centers subject to their capacities.

B. Suboptimal Heuristics with Limited Replication

Although solution (14) is optimal and efficient, it encounters two major obstacles in practice. *First,* as long as $\alpha_s > 0$, $w_{si}^* = \alpha_s > 0$ for all i, which means that data center s has to store all N videos. In other words, a video has to be replicated at all the data centers that has $\alpha_s > 0$. This incurs significant additional storage fees at the VoD provider charged by data centers. *Second,* each video channel i splits its workload into S data centers according to the weights $\alpha_1,\ldots,\alpha_S$. When S is large and D_i is small, such fine-grained splitting will not be technically feasible. Therefore, in practice, we need to limit the *replication degree* of each video, or equivalently, limiting the number of videos stored in each data center.

To achieve the above goal, we propose suboptimal solutions to problem (9) that addresses replication concerns. First, we need the following heuristic to bridge the optimal load direction to replication-limited load direction:

Heuristic 1: Per-DC Optimal. The algorithm iteratively outputs $\mathbf{w}_1^{**},\ldots,\mathbf{w}_S^{**}$ for one data center after another. Initially, set $\mathbf{b} = \mathbf{1}$. Repeat the following for $s = 1,\ldots,S$:

1) Solve the following problem to obtain $\mathbf{w}_s^{**}$:

$$\max_{\mathbf{w}_s} \ \boldsymbol{\mu}^\mathsf{T}\mathbf{w}_s \tag{17}$$

$$\text{s.t.} \quad \boldsymbol{\mu}^\mathsf{T}\mathbf{w}_s + \theta\sqrt{\mathbf{w}_s^\mathsf{T}\boldsymbol{\Sigma}\mathbf{w}_s} \le A_s \le C_s, \tag{18}$$

$$0 \le \mathbf{w}_s \le \mathbf{b}. \tag{19}$$

2) Replace $\mathbf{b}$ in (19) by $\mathbf{b} - \mathbf{w}_s^{**}$.

The program terminates if $\mathbf{b} = \mathbf{0}$.

Heuristic 1 packs the random demands into each data center, one after another, by maximizing the expected demand $\boldsymbol{\mu}^{\mathsf{T}}\mathbf{w}_s$ each data center s can accommodate subject to the probabilistic performance guarantee in (18). As a result, the total amount of resources needed to guard against demand variability is reduced. Clearly, under Heuristic 1, the aggregate bandwidth reservation from all data centers is

$$\textstyle\sum_s A_s^{**} = \sum_{s=1}^{S}(\boldsymbol{\mu}^{\mathsf{T}}\mathbf{w}_s^{**} + \theta\sqrt{\mathbf{w}_s^{**\mathsf{T}}\boldsymbol{\Sigma}\mathbf{w}_s^{**}}). \tag{20}$$

Note that Heuristic 1 is also computationally efficient since (17) is a standard second-order cone program.

Now we can introduce the replication-limited load direction, which only requires the VoD provider to upload k videos to each data center. We modify Heuristic 1 to cope with this constraint as follows:

Heuristic 2: Per-DC Limited Channels. The algorithm iteratively outputs $\mathbf{w}'_1, \ldots, \mathbf{w}'_S$ for one data center after another. Initially, set $\mathbf{b} = \mathbf{1}$. Repeat the following for $s = 1, \ldots, S$:

1) Solve problem (17) to obtain $\mathbf{w}_s^{**}$.
2) Choose the top k channels with the largest weights and solve problem (17) again only for these k channels to obtain $\mathbf{w}'_s$.
3) Replace $\mathbf{b}$ in (19) by $\mathbf{b} - \mathbf{w}'_s$.

The program terminates if $\mathbf{b} = \mathbf{0}$.

Under Heuristic 2, the aggregate bandwidth reserved is

$$\textstyle\sum_s A'_s = \sum_{s=1}^{S}(\boldsymbol{\mu}^{\mathsf{T}}\mathbf{w}'_s + \theta\sqrt{\mathbf{w}'^{\mathsf{T}}_s\boldsymbol{\Sigma}\mathbf{w}'_s}). \tag{21}$$

In Sec. V, we will show through trace-driven simulations that Heuristic 2, though suboptimal, effectively limits the content replication degree, thus balancing the savings on storage cost and bandwidth reservation cost for VoD providers.

IV. Demand Forecasting Models

The derivation of load direction decisions critically depends on parameters $\boldsymbol{u}$ and $\boldsymbol{\Sigma}$, which are estimates of the expected demands and demand covariances for the short-term future $[t, t + \Delta t)$. In this section, we present efficient time series forecasting methods to make such predictions based on past observations.

We assume that the bandwidth demand of channel i at any point in the period $[t, t + \Delta t)$ can be represented by the same random variable D_{it}. This is a reasonable assumption when Δt is small. Similarly, let $\boldsymbol{\mu}_t = [\mu_{1t}, \ldots, \mu_{Nt}]$ and $\boldsymbol{\Sigma}_t = [\sigma_{ijt}]$ represent the demand expectation vector and demand covariance matrix for all N channels in $[t, t + \Delta t)$. We assume that before time t, the system has already collected all the demand history from cloud monitoring services with a sampling interval of Δt. The question is how to use the available sampled bandwidth demand history $\{D_{i\tau} : \tau = 0, \ldots, t-1, i = 1, \ldots, N\}$ to estimate $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$?

In this paper, we combine our previously proposed seasonal ARIMA model [6] for conditional mean (expectation conditioned on the history) prediction with the GARCH model [7] for conditional variance prediction to obtain a *multivariate GARCH* model that can forecast the demand covariance matrix. The model extracts the periodic evolution pattern from each channel's demand time series, and characterizes the remaining *innovation* series as autocorrelated GARCH processes. We briefly describe these statistical models here. Interested readers are referred to [6], [7] for details.

The difficulty in modeling the bandwidth demand of a channel i is that it exhibits diurnal periodicity, a downward trend as the video becomes less popular over time, and changing levels of fluctuation as population goes up and down. Such *non-stationarity* in traffic renders unbiased linear predictors useless. We tackle this problem by applying one-day-lagged differences (the lag is 144 if $\Delta t = 10$ minutes) onto $\{D_{i\tau}\}$ to remove daily periodicity to obtain the transformed series $\{D'_{i\tau} := D_{i\tau} - D_{i\tau-144}\}$, which can be modeled as a low-order autoregressive moving-average (ARMA) process:

$$\begin{cases} D'_{i\tau} - \phi_i D'_{i\tau-1} = N_{i\tau} + \gamma_i N_{i\tau-1}, \\ D'_{i\tau} = D_{i\tau} - D_{i\tau-144}, \end{cases} \tag{22}$$

where $\{N_{i\tau}\} \sim \mathrm{WN}(0, \sigma^2)$ denotes the uncorrelated white noise with zero mean. Model (22) falls in the category of seasonal ARIMA models [6], [8].

Model parameters ϕ_i and γ_i in (22) can be trained based on historical data using a maximum likelihood estimator [8]. To predict the expected demand μ_{it} of channel i, we first predict $\mu'_{it} := \mathbf{E}[D'_{it}|D'_{it-1}, D'_{it-2}, \ldots]$ for the transformed series $\{D'_{i\tau}\}$ to obtain the estimate $\hat{\mu}'_{it}$, using an unbiased *minimum mean square error* (MMSE) predictor. We then retransform $\hat{\mu}'_{it}$ into an estimate $\hat{\mu}_{it}$ of the conditional mean μ_{it}, with the inverse of one-day-lagged differencing.

Given the conditional means $\{\hat{\mu}_{i\tau}\}$ of channel i over all time τ, we denote the *innovations* in $\{D_{i\tau}\}$ by $\{Z_{i\tau}\}$, where

$$Z_{i\tau} := D_{i\tau} - \hat{\mu}_{i\tau}. \tag{23}$$

Since the innovation term $Z_{i\tau}$ represents the fluctuation of $D_{i\tau}$ relative to its projected expectation $\hat{\mu}_{i\tau}$, and such fluctuation may be changing over time, we model the innovations $\{Z_{i\tau}\}$ using a GARCH process:

$$\begin{cases} Z_{i\tau} = \sqrt{h_{i\tau}}e_\tau, \quad \{e_\tau\} \sim \mathrm{IID}\ \mathcal{N}(0,1), \\ h_{i\tau} = \alpha_{i0} + \alpha_{i1}Z_{i\tau-1}^2 + \beta_i h_{i\tau-1}, \end{cases} \tag{24}$$

where $\{Z_{i\tau}\}$ is modeled as a zero-mean Gaussian process yet with a time-varying conditional variance $h_{i\tau}$. Instead of assuming a constant variance for $\{Z_{i\tau}\}$, (24) introduces autocorrelation into volatility evolution and forecasts the conditional variance h_{it} of Z_{it} as a regression of past $h_{i\tau}$ and $Z_{i\tau}^2$. The model parameters in (24) can be learned using maximum likelihood estimation (pp. 417, [8]) based on training data.

Finally, to predict covariance matrix $\boldsymbol{\Sigma}_t$, we introduce a *constant conditional correlation* (CCC) model [9], which is a popular multivariate GARCH specification that restricts the correlation coefficients ρ_{ij} to be constant. ρ_{ij} can be estimated as the correlation coefficient between series $\{Z_{i\tau}\}$ and $\{Z_{j\tau}\}$ in recent time periods, and $\rho_{ij} = 1$ if $i = j$. The covariance

σ_{ijt} between video i and j at time t is thus predicted as

$$\hat{\sigma}_{ijt} = h_{ijt} = \rho_{ij}\sqrt{h_{it}h_{jt}}, \tag{25}$$

with h_{it} and h_{jt} predicted using (24) for channels i and j individually.

The full statistical model is a seasonal ARIMA conditional mean model (22) with a CCC multivariate GARCH innovation model given by (24) and (25). The above seemingly complex model is extremely efficient to train, as the five parameters ϕ_i, γ_i, α_{i0}, α_{i1} and β_i are learned for each video i separately following the procedures mentioned above, and ρ_{ij} is calculated straightforwardly from recent history.

V. Experiments Based on Real-World Traces

We conduct a series of simulations to evaluate the performance of our bandwidth auto-scaling system. The simulations are driven by the replay of the workload traces of UUSee video-on-demand system over a 21-day period during 2008 Summer Olympics [10]. As a commercial VoD company, UUSee streams on-demand videos to millions of Internet users across over 40 countries through a downloadable client software. The dataset contains performance snapshots taken at a 10-minute frequency of 1693 video channels, including sports events, movies, TV episodes and other genres. The statistics we use in this paper are the time-averaged total bandwidth demand in each video channel in each 10-minute period. There are 144 time periods in a day. We ask the question—what the performance would have been if UUSee had all its workload in this period served by cloud services through our bandwidth auto-scaling system?

We conduct performance evaluation for 4 typical time spans which are near the beginning, middle and end of the 21-day duration. We implement statistical learning and demand prediction techniques presented in Sec. IV to forecast the expected demands $\boldsymbol{\mu}_t$ and demand covariance matrix $\boldsymbol{\Sigma}_t$ every 10 minutes. The model parameters are retrained daily, with training data being the bandwidth demand series $\{D_{i\tau}\}$ in the recent 1.25 days of each channel i. Once trained, the models will be used for the next 24 hours. Although video users may join or quit a channel unexpectedly, our prediction is still effective, since it deals with the *aggregate demand* in the channel which features diurnal evolution patterns. We assume that there is a pool of data centers from which UUSee can reserve bandwidth. To spread the load across data centers, we set $C_s = 300$ Mbps for each s. The QoS parameter $\theta := F^{-1}(1-\epsilon)$ is set to $\theta = 2.05$ to confine the under-provision probability to $\epsilon = 2\%$.

A. A Novel Channel Interleaving Scheme

There are two practical challenges with regard to demand prediction. *First*, many of the 1693 video channels are released during the 21 days: a new video does not have sufficient demand history for statistical model learning. *Second*, there are many small channels with only a few users online for which prediction is hard. We propose a novel channel interleaving scheme to circumvent these obstacles. It has been shown from the traces [6] that videos released on different dates but around the same time of day exhibit similar initial demand evolution patterns, though with possibly different popularities. The main reason is that most users watch VoD channels around several peak times of a day. Therefore, it is possible to predict demands for new videos based on earlier videos.

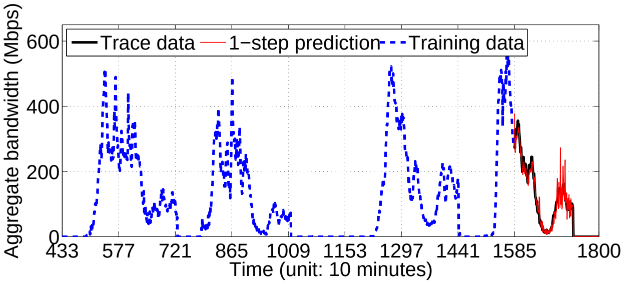


Fig. 4. The conditional mean demand prediction for virtual new channel 11, with a test period of 1.5 days from time 1585 to 1800.

We define **virtual new channel** k as a combination of all video channels with an age less than 1.25 days and released in hour $k \in \{1, \ldots, 24\}$ on any date. Upon release, a new video joins virtual new channel k based on its release hour k. Similarly, we aggregate small video channels and set up 24 **virtual small channels**. When a video reaches the age of 1.25 days, it quits its virtual new channel. If its demand never exceeded a threshold (e.g., 40 Mbps) in the first 1.25 days, it will join one of the virtual small channels in a round robin fashion. Otherwise, it becomes a **mature channel**.

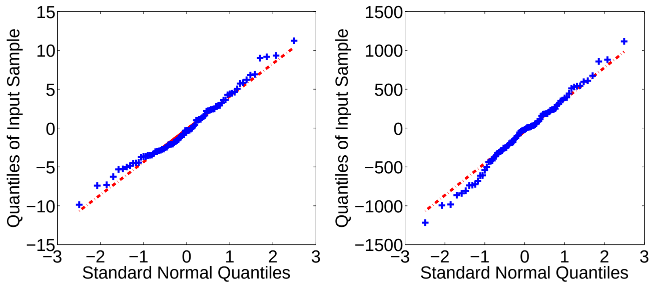
Each mature or virtual channel is deemed as an entity to which predictions and optimizations are applied. For example, we make 10-minutes-ahead conditional mean prediction for virtual new channel 11 and plot results in Fig. 4. The bandwidth demand exhibits repetition of a similar pattern because the videos in this virtual channel are all released in hour 11 (possibly on different dates). Although conditional mean prediction is subject to errors, the GARCH model can forecast the changing error variance, which contributes to the risk constraint (11) in resource minimization (9).

B. Algorithms for Comparison

We compare our optimal load direction (14), Heuristic 1 and Heuristic 2 with the following benchmark algorithms:

Reactive without Prediction. Initially replicate each video to K random data centers. This limits the initial content replication degree to K. Each client requesting channel i is randomly directed to a data center that has video i and idle bandwidth capacity. A request is dropped if there is no such data center. In this case, the algorithm reacts by replicating video i to an additional data center chosen randomly that has idle capacity. Replicating content is not instant: we assume that the replication involves a delay of one period of time.

Random with Prediction. Initially, let $s = 1$ and $\mathbf{b} = 1$. Second, randomly generate $\mathbf{w}_s$ in $(\mathbf{0}, \mathbf{b})$ and rescale it so that the QoS constraint (11) is achieved with equality for s. Update $\mathbf{b}$ to $\mathbf{b} - \mathbf{w}_s$ and update s to $s+1$. Go to the second step unless $\mathbf{b} = 0$ or $s = S + 1$, in which case the program terminates.



(a) Z_{it} of a typical channel (b) $\sum_i Z_{it}$ of all channels combined

Fig. 5. QQ plot of innovations for t =1562—1640 vs. normal distribution.

The reactive scheme represents provisioning for peak demand in Fig. 1 in some way, with limited replication. It does not leverage prediction or bandwidth reservation. We assume in Reactive, the total cloud capacity allocated is always the minimum capacity needed to meet the peak demand in the system. The random scheme leverages prediction and makes bandwidth reservation, but randomly directs workloads instead of using anti-correlation to minimize bandwidth reservation.

C. Assumption Validation

First, we verify that D_{it} approximately follows Gaussian distribution in each 10-minute period. For each channel i, given conditional mean prediction $\hat{\mu}_{it}$ at time t, the innovation is $Z_{it} := D_{it} - \hat{\mu}_{it}$. Fig. 5(a) shows the QQ plot of Z_{it} for a typical channel $i = 121$ from time period 1562 to 1640, which indicates $\{Z_{it}\}$ sampled at 10-minute intervals is a Gaussian process. Thus, it is reasonable to assume D_{it} follows a Gaussian distribution within the 10 minutes following t, with mean $\hat{\mu}_{it}$. Fig. 5(b) shows the QQ plot of $\sum_i Z_{it}$, which indicates that the aggregated demand $\sum_i D_{it}$ tends to Gaussian even if D_{jt} is not for some channel j. Since the load L_s of each data center is aggregated from many videos, it is reasonable to assume L_s is Gaussian.

Furthermore, it has been verified in [7] that the innovations $\{Z_{it}\}$ forms a stationary uncorrelated series whereas $\{Z_{it}^2\}$ is auto-correlated, justifying the validity of GARCH modeling of innovations $\{Z_{it}\}$ in Sec. IV.

D. Predictive Auto-Scaling vs. Reactive Provisioning

We implement all of the five schemes discussed above, and present their performance comparison in Table I for each of the four time spans. Note that the channels in the table include mature channels, virtual new and virtual small channels. The number of videos in each virtual channel can vary over time. As new videos are introduced, more channels are present in later test periods. We evaluate the performance with regard to QoS, bandwidth resource occupied, and replication cost.

Table I shows that Reactive generally has a more salient QoS problem than all four predictive schemes in terms of both the number of unsatisfied channels and request drop rate, demonstrating the benefit of utilizing demand prediction. Fig. 6 presents a more detailed comparison for a typical peak period from time 702 to 780. Without surprise, Reactive has many unfulfilled requests at the beginning. Since the videos are randomly replicated to $K = 2$ data centers (shown in Fig. 6(c) at $t = 702$) and requests are randomly directed, it is likely that a channel does not acquire enough capacity to meet its demand. As Reactive detects the QoS problem, videos are replicated to more data centers to acquire more capacity, with a gradually increasing replication degree over time, as in Fig. 6(c). We can see that after 140 minutes, when the replication degree reaches above 4, the QoS of Reactive becomes relatively stable in Fig. 6(a). However, around time 763, Reactive suffers from salient QoS problems again, due to a sudden ramp-up of demand. In contrast, the predictive schemes foresee and prepare for demand changes, resulting in much better QoS, even in the event of drastic demand increase.

The predictive schemes also achieve higher resource utilization. Utilization of a predictive scheme is the ratio between the actual used bandwidth and the total booked bandwidth in all data centers. For Reactive the utilization is the actual bandwidth demand divided by the peak demand. Although Fig. 6(b) shows that Reactive achieves a high utilization for the peak demand around time 763, its average utilization is merely 77.19% in the test period from 702 to 780. Predictive auto-scaling enhances utilization to 85.67% with Per-DC Limited Channels, to 89.99% with Per-DC Optimal, and to 92.9% with the theoretical optimal solution. In addition, the prediction and optimization in predictive methods are computationally efficient, e.g., prediction and Per-DC Optimal finish in 2 minutes, well before the deadline of 10 minutes.

E. Theoretical Optimal vs. Replication-limited Heuristics

Now we focus on each of the four predictive schemes. Among them, as shown in Table I, Optimal books the minimum necessary bandwidth and achieves the highest bandwidth utilization, yet with the highest replication overhead: a video is replicated to every data center. The VoD provider thus needs to pay a high storage fee to the cloud.

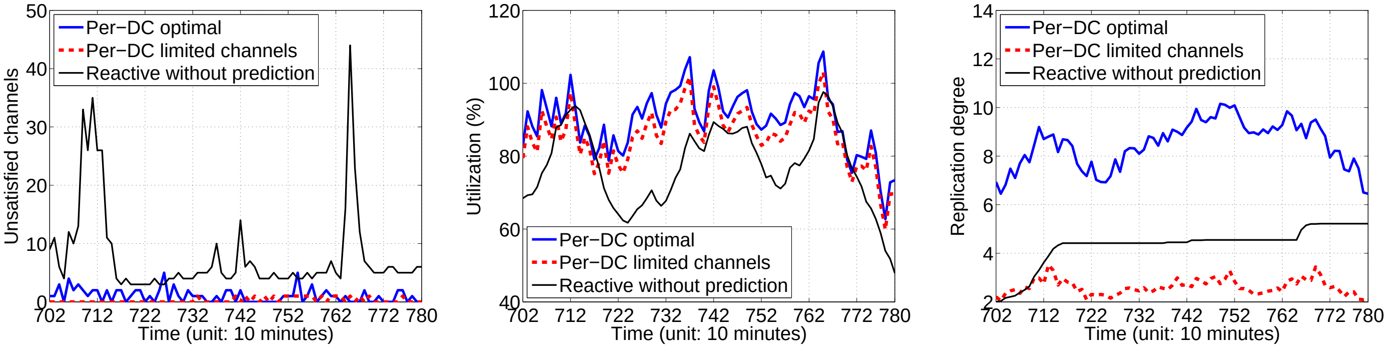
Per-DC Optimal can reduce the replication degree while maintaining other performance metrics. By further imposing a channel number constraint on each data center, Per-DC Limited Channels strikes a balance between replication overhead and bandwidth utilization. It aggressively reduces the replication degree to a very small value of 2.4-2.6 copies/video, which is the smallest among all four schemes, with an extremely low drop rate and an over-provisioning ratio only slightly higher than Optimal and Per-DC Optimal. Random achieves the lowest utilization, since it is blind to the correlation information in workload selection and direction.

We further show a detailed comparison between the three predictive *heuristics* from time 1562 to 1640 in Fig. 7. The efficiency of predictive bandwidth booking can be evaluated by the *cushion bandwidth* needed, which is the gap between the booked bandwidth and actual required bandwidth. Fig. 7(a) plots the cushion bandwidth. While being on the same QoS level, random load direction results into a cushion bandwidth up to 3 Gbps compared to a mean demand of 5.62 Gbps, representing significant over-provisioning. Using Per-DC Optimal, the cushion bandwidth can be saved by 50% on average,

TABLE I
THE PERFORMANCE OF 5 SCHEMES AVERAGED OVER EACH TEST PERIOD, IN TERMS OF QOS, RESOURCE UTILIZATION, AND REPLICATION.

Periods	**Time periods 702—780** (91 mature and virtual channels) Peak demand 6.56 Gbps, mean demand 5.19 Gbps						**Time periods 1422—1480** (161 mature and virtual channels) Peak demand 6.81 Gbps, mean demand 4.91 Gbps					
	Short	Drop	Util	Rep	Booked	Over-prov	Short	Drop	Util	Rep	Booked	Over-prov
Optimal	0.2 Chs	0.66%	92.9%	91.0	6.57 Gbps	108.5%	0.1 Chs	0.25%	91.1%	161.0	6.38 Gbps	110.3%
Per-DC Opt	1.0 Chs	0.37%	90.0%	8.5	6.79 Gbps	112.2%	1.2 Chs	0.13%	88.6%	6.9	6.56 Gbps	113.4%
Per-DC Lim	0.3 Chs	0.06%	85.7%	2.6	7.13 Gbps	117.8%	0.2 Chs	0.03%	84.6%	2.4	6.86 Gbps	118.8%
Random	5.9 Chs	0.02%	83.3%	3.8	7.33 Gbps	121.2%	7.6 Chs	0.00%	82.2%	3.0	7.08 Gbps	122.4%
Reactive	7.9 Chs	0.47%	77.2%	4.3	7.91 Gbps	132.4%	7.2 Chs	0.34%	70.4%	3.6	8.20 Gbps	146.0%
Periods	**Time periods 1562—1640** (176 mature and virtual channels) Peak demand 7.55 Gbps, mean demand 5.62 Gbps						**Time periods 2402—2500** (199 mature and virtual channels) Peak demand 9.19 Gbps, mean demand 7.62 Gbps					
	Short	Drop	Util	Rep	Booked	Over-prov	Short	Drop	Util	Rep	Booked	Over-prov
Optimal	0.1 Chs	0.31%	91.1%	176.0	7.27 Gbps	110.4%	0.0 Chs	0.11%	85.4%	199.0	10.54 Gbps	118.1%
Per-DC Opt	0.7 Chs	0.16%	88.3%	7.3	7.51 Gbps	114.0%	1.0 Chs	0.09%	82.7%	6.3	10.87 Gbps	121.8%
Per-DC Lim	1.4 Chs	0.00%	83.9%	2.4	7.89 Gbps	119.9%	20.7 Chs	0.17%	82.3%	2.5	10.95 Gbps	122.6%
Random	6.2 Chs	0.00%	80.4%	3.3	8.28 Gbps	125.4%	33.4 Chs	0.02%	77.9%	4.5	11.54 Gbps	129.3%
Reactive	5.9 Chs	0.27%	72.7%	3.5	9.08 Gbps	140.4%	15.8 Chs	0.43%	74.6%	3.6	12.01 Gbps	140.3%

Short: # channels with dropped requests; *Drop*: the request drop rate; *Util*: utilization of allocated resources; *Rep*: replication degree; *Booked*: the booked bandwidth; *Over-prov*: over-provisioning ratio.



(a) Channels with unsatisfied requests

(b) Resource Utilization

(c) Replication Degree

Fig. 6. Predictive vs. reactive bandwidth provisioning for a typical peak period 702–780. There are 35 data centers available, each with capacity 300 Mbps, and 91 channels, including 52 popular channels, 24 small channels, 15 non-zero new channels. $K = 2$, $k = 10$.

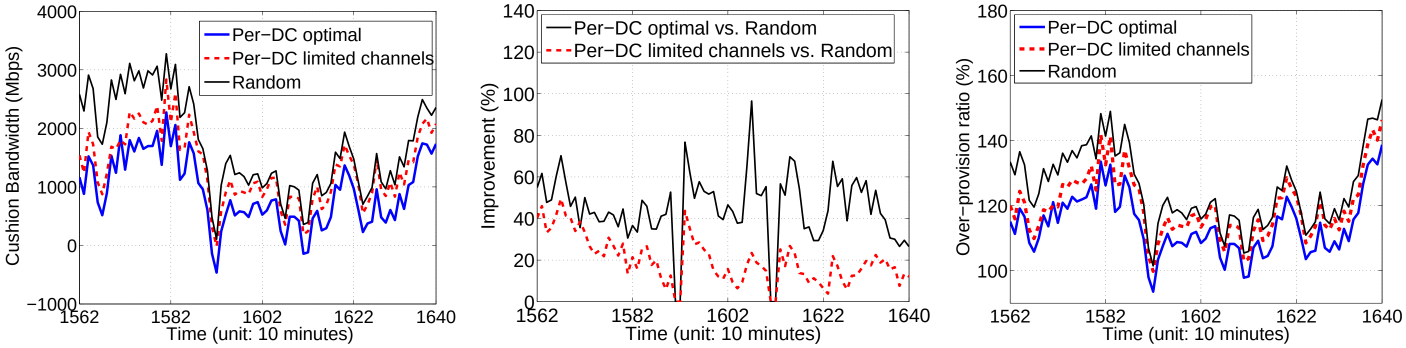
as shown in Fig. 7(b). Even Per-DC Limited Channels, with a replication degree of 2.4 copies/video, can save cushion bandwidth by around 30% as compared to Random, which has a higher replication degree of 3.3 copies/video.

QoS problems occur if bandwidth is under-provisioned, leading to a cushion bandwidth below 0 and an over-provisioning ratio less than 100%. From Fig. 7(a) and Fig. 7(c), we observe that QoS problems occur occasionally for Per-DC Optimal but seldom for Per-DC Limited Channels from time 1562 to 1640, because the latter scheme conservatively books more cushion bandwidth. In addition, we note that request drop rates in Table I are significantly lower than the frequency of under-provisioning in the figures, because when under-provisioning happens, most user requests are still served. Only the demand exceeding the booked capacity is dropped. From the above analysis, we conclude that Per-DC Limited Channels achieves the best tradeoff in the domain of utilization, QoS and replication overhead.

VI. RELATED WORK

Researches on exploiting virtualization techniques for delivering cloud-based IPTV services have been conducted by major VoD providers like AT&T [11]. The importance of VoD bandwidth demand projection on capacity planning has also been recognized. It is shown that demand estimates can help with optimal content placement in AT&T's IPTV network [12]. More advanced video demand forecasting techniques have been proposed, such as the non-stationary time series models introduced in [6], [7], and video access pattern extraction via principal component analysis in [13].

Predictive and dynamic resource provisioning has been proposed mostly for virtual machines (VM) and web applications with respect to CPU utilization [14]–[17] and power consumption [18], [19]. VM consolidation with dynamic bandwidth demand has also been considered in [20]. Our work exploits the unique characteristics of VoD bandwidth demands and distinguishes from the above work in three aspects. *First*, our bandwidth workload consolidation is as simple as solving convex optimization for a load direction matrix. We leverage the fact that unlike VM, demand of a VoD channel can be *fractionally* split into video requests. *Second*, our system forecasts not only the expected demand but also the demand volatility, and thus can control the risk factors more accurately. In contrast, most previous works [15], [17] assume a constant demand variance. *Third*, we exploit the statistical correlation



(a) Cushion Bandwidth

(b) Savings on Cushion Bandwidth

(c) Over-Provisioning Ratio

Fig. 7. Workload portfolio selection vs. random load direction for a typical peak usage period from time 1562 to 1640. $K = 2$, $k = 10$.

between bandwidth demands of different video channels to save resource reservation while previous works such as [20] consider independent workloads.

The idea of statistical multiplexing and resource overbooking has been empirically evaluated for a shared hosting platform in [21]. Our novelty is that we formulate the quality-assured resource minimization problem using Value at Risk (VaR), a useful risk measure in financial asset management [22], with the aid of accurate demand correlation forecasts. We believe our theoretically grounded approach bears stronger robustness against intractable demand volatility in practice.

VII. Concluding Remarks

In this paper, we propose an unobtrusive, predictive and elastic cloud bandwidth auto-scaling system for VoD providers. Operated at a 10-minute frequency, the system automatically predicts the expected future demand as well as demand volatility in each video channel through ARIMA and GARCH time-series forecasting techniques based on history. Leveraging demand prediction, the system jointly makes load direction to and bandwidth reservations from multiple data centers to satisfy the projected demands with high probability. The system can save the resource booking cost for VoD providers with regard to both bandwidth and storage.

We exploit the predictable anti-correlation between demands to enhance resource utilization, and derive the optimal load direction that minimizes the bandwidth resource reservation while confining under-provision risks. Two suboptimal heuristics have also been proposed to limit the storage cost. From extensive simulations driven by the demand traces of a large-scale real-world VoD system, we observe that suboptimal heuristics have practical appeals due to their ability to balance the costs of bandwidth and storage.

References

[1] "Four Reasons We Choose Amazon's Cloud as Our Computing Platform," *The Netflix "Tech" Blog*, December 14 2010.

[2] C. Guo, G. Lu, H. J. Wang, S. Yang, C. Kong, P. Sun, W. Wu, and Y. Zhang, "SecondNet: a Data Center Network Virtualization Architecture with Bandwidth Guarantees," in *Proc. ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, 2010.

[3] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards Predictable Datacenter Networks," in *Proc. of SIGCOMM'11*, Toronto, ON, Canada, 2011.

[4] "Amazon Web Services," http://aws.amazon.com/.

[5] T. Bollerslev, "Generalized Autoregressive Conditional Heteroskedasticity," *Journal of Econometrics*, vol. 31, pp. 307–327, 1986.

[6] D. Niu, Z. Liu, B. Li, and S. Zhao, "Demand Forecast and Performance Prediction in Peer-Assisted On-Demand Streaming Systems," in *Proc. IEEE INFOCOM Mini-Conference*, 2011.

[7] D. Niu, B. Li, and S. Zhao, "Understanding Demand Volatility in Large VoD Systems," in *Proc. the 21st International workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, 2011.

[8] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*. Wiley, 2008.

[9] W. Enders, *Applied Econometric Time Series*, 3rd ed. Hoboken, NJ: Wiley, 2010.

[10] Z. Liu, C. Wu, B. Li, and S. Zhao, "UUSee: Large-Scale Operational On-Demand Streaming with Random Network Coding," in *Proc. IEEE INFOCOM*, 2010.

[11] V. Aggarwal, X. Chen, V. Gopalakrishnan, R. Jana, K. K. Ramakrishnan, and V. A. Vaishampayan, "Exploiting Virtualization for Delivering Cloud-based IPTV Services," in *Proc. IEEE INFOCOM Workshop on Cloud Computing*, 2011.

[12] D. Applegate, A. Archer, V. G. S. Lee, and K. Ramakrishnan, "Optimal Content Placement for a Large-Scale VoD System," in *Proc. ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*, 2010.

[13] G. Gürsun, M. Crovella, and I. Matta, "Describing and Forecasting Video Access Patterns," in *Proc. IEEE INFOCOM Mini-Conference*, 2011.

[14] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic Placement of Virtual Machines for Managing SLA Violations," in *Proc. 10th IFIP/IEEE International Symposium on Integrated Network Management*, 2007.

[15] Z. Gong, X. Gu, and J. Wilkes, "PRESS: PRedictive Elastic ReSource Scaling for Cloud Systems," in *Proc. IEEE International Conference on Network and Services Management (CNSM)*, 2010.

[16] C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici, "A Scalable Application Placement Controller for Enterprise Data Centers," in *Proc. ACM WWW*, 2007.

[17] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper, "Workload Analysis and Demand Prediction of Enterprise Data Center Applications," in *Proc. IEEE Symp. Workload Characterization*, 2007.

[18] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and Performance Management of Virtualized Computing Environments via Lookahead Control," *Cluster computing*, vol. 12, no. 1, pp. 1–15, March 2009.

[19] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic Right-Sizing for Power-Proportional Data Centers," in *Proc. IEEE INFOCOM*, 2011.

[20] M. Wang, X. Meng, and L. Zhang, "Consolidating Virtual Machines with Dynamic Bandwidth Demand in Data Centers," in *Proc. of IEEE INFOCOM 2011 Mini-Conference*, 2011.

[21] B. Urgaonkar, P. Shenoy, and T. Roscoe, "Resource Overbooking and Application Profiling in Shared Hosting Platforms," in *Proc. USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2002.

[22] A. McNeil, R. Frey, and P. Embrechts, *Quantitative Risk Management: Concepts Techniques and Tools*. Princeton University Press, 2005.