Equipment Allocation in Video-on-Demand Network Deployments

FREDERIC THOUIN and MARK COATES

McGill University

Video-on-Demand (VoD) services are very user-friendly, but also complex and resource demanding. Deployments involve careful design of many mechanisms where content attributes and usage models should be taken into account. We define, and propose a methodology to solve, the *VoD Equipment Allocation Problem* of determining the number and type of streaming servers with directly attached storage (VoD servers) to install at each potential location in a metropolitan area network topology such that deployment costs are minimized. We develop a cost model for VoD deployments based on streaming, storage and transport costs and train a parametric function that maps the amount of available storage to a worst-case hit ratio. We observe the impact of having to determine the amount of storage and streaming co-jointly, and determine the minimum demand required to deploy replicas as well as the average hit ratio at each location. We observe that common video-on-demand server configurations lead to the installation of excessive storage, because a relatively high hit-ratio can be achieved with small amounts of storage so streaming requirements dominate.

1. INTRODUCTION

Deployments for services such as Video-on-Demand (VoD) that use Internet resources are today still very limited in scope, but there is a strong belief among telecommunication companies that this market will expand exponentially in the next few years. Many service providers wish to take advantage of this opportunity and are planning the deployment of large-scale Video-on-Demand systems [Markman 2006]. VoD dedicates a single channel to each user and allows the video to be started at any time with VCR-like controls (pause, rewind, fast-forward, etc.). While being very user-friendly, this type of service is complex and resource demanding. Deployments of efficient IP network-based VoD services involve careful design of many mechanisms to route requests, and deliver, distribute and allocate content in the network [Masa and Parravicini 2003; Wang et al. 2002; Krishnan et al. 2000; Laoutaris et al. 2005; Wauters et al. 2005] . Furthermore, content attributes and usage models should also be taken into account [Couch 2005].

   The high bandwidth requirements motivate distributed architectures with replication of content, but such architectures imply a substantial increase in storage requirements, a non-negligible factor given the large size of video files. An important and complicated task part of the network planning phase of these distributed architectures is resource allocation. Two emerging trends have made this non-trivial problem even more complex: peer-to-peer and mobile delivery. By accounting for

the resources of each user, peer-to-peer architectures offer better scaling capabilities and represent a promising alternative to reduce high bandwidth and storage requirements. The peer-to-peer architectures do not offer any serious benefits in the star topologies of many metropolitan area networks where VoD network deployments will occur, and there are many other practical issues that must be addressed before a large-scale implementation becomes feasible [Thouin and Coates 2007a]. The major trend is the emergence of video delivery to mobile devices, such as portable phones. Mobile users are interested in accessing a library (and/or their personal library) of videos wherever they go; having a video delivery service that follows them around the world. There is still on-going research trying to determine the best way to approach this problem of meeting delivery requirements for mobile users.

From these tendencies, it is clear that there is an urgent need for design tools that can determine how best to expand upon existing network infrastructure and where to deploy storage and streaming devices. To minimize deployment costs of proxy-based systems, one must determine the equipment to install at each of these proxy sites to reduce as much as possible costs related to transport. The streaming servers, called VoD servers, commonly have directly-attached storage, a structure easier to deploy and manage. These benefits come at a cost, however, because the planning phase becomes more challenging: optimizing resources is more difficult when streaming and storage requirements need to be determined co-jointly. In this paper, we define, and propose a methodology to solve, the *VoD Equipment Allocation Problem* (VoD EAP) of determining the number (and type) of VoD servers to install at each potential location in a metropolitan area network (MAN) topology such that deployment costs are minimized. Our design tool is the first to not only calculate how much of a given library of video objects should be replicated at each video server, but also to explicitly describe the specifications (cost, streaming and storage capacity, etc.) of the equipment that should be deployed to minimize cost. This information is very valuable for equipment manufacturers. By using the tool to simulate a variety of different scenarios, fabricators can obtain precious insight into how to optimize their products to best suit the needs of providers.

The rest of this paper is organized as follows. In Sect. 2, we discuss related previous work and outline our main contributions. In Sect. 3, we define the *VoD Equipment Allocation Problem* as an optimization problem. The first step involved in solving this problem is developing a cost model for VoD deployments; we propose such a model based on streaming, storage and transport costs in Sect. 4. We also take into account file popularity by training a parametric function that maps the amount of available storage to a worst-case hit ratio based on simulations we ran using a file access model suggested by Gummadi et al. [2003], and Sect. 5 describes this procedure. Sect. 6 characterizes our novel heuristics to solve the VoD EAP and two variations (one where many different models can be installed at a single location and another where streaming and storage are optimized independently). Sect. 7 presents results from simulations and the interactive tool that we developed that integrates our cost model and heuristic. Finally, we summarize our observations and propose future research avenues in Sect. 8.

2.   RELATED WORK

Many known optimization techniques used in the context of content delivery networks (CDNs), such as the replica placement and facility location problem [Cornuejols et al. 1990], have been adapted to VoD networks or media delivery design problems: the capacitated facility location problem [Wu et al. 2006], the placement of replica servers [Karlsson et al. 2002; Krishnan et al. 2000; Almeida et al. 2004] or the placement of video objects [Tang et al. 2004]. However, Laoutaris et al. [2005] provide strong evidence that solving the replica placement or video placement problem independently of the resource allocation problem usually leads to significantly suboptimal solutions. They define the *storage capacity allocation problem* as determining the location of each object from a set to achieve minimal cost. Although this approach establishes storage requirements at each location, the authors do not explicitly consider available equipment and streaming requirements. Wauters et al. [2005] address the problem of determining the equipment required for transport (the number of ports, multiplexers and switch ports) at each candidate network nodes, but do not consider storage requirements.

Although many of the cost functions proposed in previous work take into account the cost of servers [Kim and Choi 2003; Nguyen et al. 2003], they typically only include start-up and storage costs and neglect the streaming requirements at each location. In [Thouin et al. 2006], we introduced the *VoD Equipment Allocation Problem* as the task of determining the number of VoD servers to allocate to each potential replica site. We use a network cost function based on the hit ratio [Yang and Fei 2003] that includes streaming, storage, server start-up and transport costs. This problem assumed that the type (model) of equipment installed at each location had been pre-specified. This pre-specification is a challenging task in and of itself and has a dramatic impact on the eventual cost of the system. In [Thouin and Coates 2007b], we re-formulated the problem to determine both the number and model of the servers to install at each location. Instead of fixing the streaming and storage capacity per VoD server at each site (the approach used in [Thouin et al. 2006]), we require only the pre-specification of a set of available VoD servers. We developed heuristics to select the model at each location that minimizes total network cost.

In this paper, we extend the work presented in [Thouin et al. 2006] and [Thouin and Coates 2007b]. We present heuristics to solve three variations of the VoD EAP, conduct a more detailed analysis of our observations and address practical deployment considerations. We propose two variations to the single-model EAP (SMEAP): the relaxed EAP (REAP) and the multi-model EAP (MMEAP). In the multi-model equipment allocation problem, we modify the assumption that a single model must be chosen for each location by permitting multiple VoD server models from the provided set to be installed at each replica site. The flexibility of combining models with different streaming and storage capacity at each location enables us to perform fine-tuning that should lead to lower-cost solutions. In the relaxed EAP, we allow the streaming and storage to be optimized independently; the resulting solutions establish lower bounds on the cost of solutions for the single-model and multi-model equipment allocation problems.

2.1 Contributions

Most previous work has focused on the placement of video servers and the allocation of video objects in the network, neglecting the capabilities and limitations of the server equipment. We formulate the design task as a constrained optimization problem that focuses on the number and configuration of video servers at each location in the network. Our contributions to solving this problem are:

(1) We explicitly specify the equipment (quantity and model of video server) that should be deployed to obtain a cost-efficient design. Due to the complexity involved in solving this problem, we devise a novel heuristic to perform the optimization procedure.

(2) We develop accurate models for all of the factors involved: the available infrastructures, the network topology, usage of the system, the popularity of videos, behaviour of customers, and bandwidth and storage requirements. Specifically, to ensure the functionality of the proposed systems, we establish a relationship between storage and the utilization of a video server that relies heavily on users' behavior.

(3) We integrate our method in a prototype software package that allows the user, through a graphical user interface, to determine the number and model of streaming and storage devices to install at strategic locations in the topology such that deployment costs are minimized. The tool provides the ability to: (i) create topologies and models of network components and VoD infrastructures and (ii) visualize the cost-efficient design generated.

3. VOD EQUIPMENT ALLOCATION PROBLEM

We consider metropolitan area networks (MANs) divided into N clusters of clients connected in a star topology through an origin server that stores the entire library of available objects to users, as depicted in Figure 1. Our focus on such (logical) topologies is motivated by their prevalence in metropolitan areas. Each cluster has worst-case demand M_i (peak usage demand) and is co-located with a potential location to install a proxy server (replica). Replicas serve, using unicast delivery, a fraction of the incoming demand (from their associated cluster) equal to x_i by storing a portion z_i of the library of size L (in TB). We assume that all requests that cannot be served by the replicas are handled directly at the origin as opposed to alternate replicas, a hierarchical architecture also used in Mundur et al. [2004]. This architecture obviates the implementation of sophisticated routing mechanisms that complicate the planning of resources and prediction of demand at each node.

3.1 Single-model support

The *VoD equipment allocation problem* (EAP) consists of determining the number and model of VoD servers to install at each location such that the deployment cost is minimized under worst-case demand. Any solution that satisfies streaming, storage and bandwidth requirements at times of peak usage also meets the requirements at any other time. Figure 1 depicts a potential solution to the problem where three different models are available. We denote the set of available VoD server models as $\mathcal{W} = \{w_j : j = 1, \ldots, W\}$, where w_j is a VoD server with streaming capacity s_j
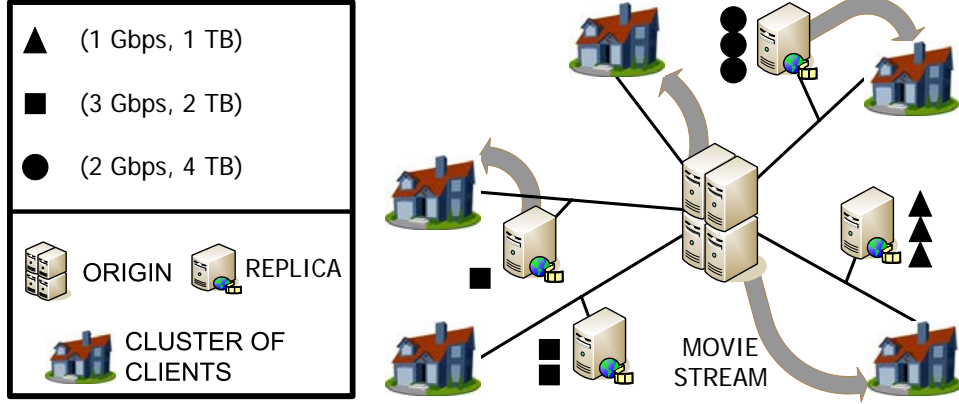
Fig. 1. Pictorial depiction of the VoD equipment allocation problem. The diagram shows the logical connectivity between the origin, the $N = 5$ clusters of clients and the potential replica server locations. Clients' requests (shown as movie stream arrows) are served by replicas (if content is available) or by the origin. The key shows the specifications of $W = 3$ different VoD server models. We show the number and type of VoD servers installed at each potential location. The optimal solution can include clusters without any servers at a potential replica site.

Gbps, storage capacity t_j TB. We define $\underline{n} = [n_0, n_1, \ldots, n_N]$, $\underline{v} = [v_0, v_1, \ldots, v_N]$ and, $\underline{x} = [x_0, x_1, \ldots, x_N]$ where x_i is the fraction of the demand M_i served by the replica, n_i is the number and $v_i \in \mathcal{W}$ is the model of VoD servers installed at cluster i (where $i = 0$ is used to denote the origin server). The optimization problem is expressed as follows:

$$(\underline{n}^*, \underline{v}^*, \underline{x}^*) = \arg \min_{\underline{n}, \underline{v}, \underline{x}} C_{\underline{v}}(\underline{n}, \underline{x}) \tag{1}$$

where $C_{\underline{v}}(\underline{n}, \underline{x})$ is the total cost of the network for a fixed $\underline{v}$. We also define the following constraints:

$$h\left(\frac{n_i \cdot t_i}{L}\right) \geq x_i, \qquad i = 1, \ldots, N \tag{2}$$

$$n_i \cdot s_i \geq x_i \cdot M_i, \quad i = 1, \ldots, N \tag{3}$$

$$n_0 \cdot t_0 \geq L \tag{4}$$

$$n_0 \cdot s_0 \geq \sum_{i=1}^{N} (1 - x_i) \cdot M_i \tag{5}$$

Constraints (2) and (3) state, respectively, that there should be sufficient storage and streaming capacity at each location i to serve a fraction of the demand equal to x_i. Note that x_i refers to the fraction of the worst-case demand that can be served; for demands smaller than M_i, the fraction of the demand could be greater than x_i. In (2), we introduce $h : [0, 1] \rightarrow [0, 1]$, a function that maps the fraction of the library stored (a function of the amount of storage) to an upper bound on the achievable hit ratio. The hit ratio is the fraction of video requests that can

be served by a replica. We describe an example h in Sect. 5. Constraints (4) and (5) ensure that enough storage and streaming is installed at the origin to store the entire library and satisfy un-served requests from every cluster of clients.

3.2 Multi-model support

In the formulation of the equipment allocation problem, only one server model can be installed at each replica (single-model EAP). A possible extension is to allow multiple models at each location (multi-model EAP). For the MMEAP, we redefine n_i as a vector $\underline{n}_i = [n_i(0), n_i(1), \ldots, n_i(W)]$ where $n_i(j)$ represents the number of servers of model w_j installed at cluster i. The multi-model equipment allocation problem (MMEAP) is expressed as follows:

$$(\underline{n}^*, \underline{x}^*) = \arg\min_{\underline{n}, \underline{x}} C_{\mathcal{W}}(\underline{n}, \underline{x}) \tag{6}$$

where $\underline{n} = [\underline{n}_0, \underline{n}_1, \ldots, \underline{n}_N]$ and $C_{\mathcal{W}}(\underline{n}, \underline{x})$ represents the deployment cost for a set of available VoD servers $\mathcal{W}$. Constraints (2)-(5) are replaced by (7)-(10):

$$h\left(\frac{\sum_{j=0}^{W} n_i(j) \cdot t_j}{L}\right) \geq x_i, \qquad i = 1, \ldots, N \tag{7}$$

$$\sum_{j=0}^{W} n_i(j) \cdot s_j \geq x_i \cdot M_i, \quad i = 1, \ldots, N \tag{8}$$

$$\sum_{j=0}^{W} n_0(j) \cdot t_j \geq L \tag{9}$$

$$\sum_{j=0}^{W} n_0(j) \cdot s_j \geq \sum_{i=1}^{N} (1 - x_i) \cdot M_i \tag{10}$$

3.3 No VoD server restrictions

We define a relaxed version of the EAP (REAP) where we do not consider VoD servers and hence can adjust the streaming capacity and storage at each location independently. The motivation for solving this problem is to observe the impact of video servers, which have fixed ratios between storage and streaming capacity, on the deployment cost. The solution to the REAP problem provides a lower bound on SMEAP and MMEAP solutions. Because streaming and storage are independent, we can solve this problem by minimizing solely in terms of x_i and z_i:

$$(\underline{z}^*, \underline{x}^*) = \arg\min_{\underline{z}, \underline{x}} C(\underline{z}, \underline{x}) \tag{11}$$

where $C(\underline{z}, \underline{x})$ is the deployment cost of the network when a fraction of the demand equal to x_i is served at replica i and a fraction of the library z_i is cached at the replica. In that case, the only constraint is the following:

$$h(z_i) \geq x_i, \qquad i = 1, \ldots, N \tag{12}$$

4. NETWORK COST MODEL

To make the optimization problem defined in Sect. 3 concrete, we first derive an expression for the total deployment cost C under the relaxed version of the EAP. In (13), we express the total cost $C^{'}$ as the sum of the costs of transport and servers (which include startup, storage and streaming costs). We introduce four constants, $\alpha^{'}, \beta^{'}, \gamma^{'}$ and $\kappa^{'}$, that represent, respectively, the startup cost of one server, the cost of storing 1 TB, the cost of streaming 1 Gbps, and the cost of transporting 1 Gbps from the origin to a cluster of clients.

$$C^{'}(\underline{z}, \underline{x}) = \sum_{i=0}^{N} \underbrace{\underbrace{\alpha^{'} \cdot (y_i)}_{\text{startup}} + \underbrace{\beta^{'} \cdot \lceil z_i \cdot L \rceil}_{\text{storage}} + \underbrace{\gamma^{'} \cdot \lceil x_i \cdot M_i \rceil}_{\text{streaming}}}_{\text{server}} + \underbrace{\kappa^{'} \cdot [(1 - x_i) \cdot M_i]}_{\text{transport}} \qquad (13)$$

with $y_i = \mathbf{1}(x_i > 0)$ where $\mathbf{1}(x)$ is the indicator function. Note that we round up the streaming and storage requirements to the nearest integer and keep fractional values for the amount of bandwidth to transport because we assume that streaming and storage devices are used solely for the purpose of the VoD network whereas transport infrastructures can be used for other applications.

As one of the objectives of installing replicas is to reduce transport costs, it makes sense to express all other costs relative to $\kappa^{'}$. We normalize $C^{'}$ by dividing the expression inside the summation in (13) by $\kappa^{'}$ and then setting $\kappa^{'} = 1$, and obtain the following formulation:

$$C(\underline{z}, \underline{x}) = \sum_{i=0}^{N} \alpha \cdot (y_i) + \beta \cdot \lceil z_i \cdot L \rceil + \gamma \cdot \lceil x_i \cdot M_i \rceil + (1 - x_i) \cdot M_i \qquad (14)$$

where α, β and γ are the ratios between the cost of startup, storage and streaming, and the cost of transport. Because the fraction of the demand served at the origin, x_0, is always equal to 1, we can extract the origin from the summation:

$$C(\underline{z}, \underline{x}) = \underbrace{\alpha + \beta \cdot \lceil L \rceil + \gamma \left[\sum_{i=1}^{N} (1 - x_i) \cdot M_i \right]}_{\text{origin}}$$

$$+ \sum_{i=1}^{N} \underbrace{\alpha \cdot (y_i) + \beta \cdot \lceil z_i \cdot L \rceil + \gamma \cdot \lceil x_i \cdot M_i \rceil}_{\text{replica}} + \underbrace{(1 - x_i) \cdot M_i}_{\text{transport}} \qquad (15)$$

As previously mentioned, in practice, streaming and storage are often tied together because of the physical structures of VoD servers. To solve the single model EAP, we modify C such that it is expressed in terms of the model of the servers $\underline{v}$ instead of β and γ. We define $B_i \triangleq \beta \cdot t_i + \gamma \cdot s_i$; this is the ratio between the cost of the VoD server installed at i (with streaming capacity s_i and storage capacity t_i) and the transport of 1 Gbps from the origin to a cluster. By setting $\lceil z_i \cdot L = t_i \cdot n_i \rceil$ and $\lceil x_i \cdot M_i \rceil = s_i \cdot n_i$ in (14), we derive an expression for $C_{\underline{v}}$:

$$C_{\underline{v}}(\underline{n}, \underline{x}) = \sum_{i=1}^{N} \underbrace{\alpha \cdot (y_i) + B_i \cdot n_i}_{\text{replica}} + \underbrace{(1 - x_i) \cdot M_i}_{\text{transport}} + \underbrace{\alpha + B_0 \cdot n_0}_{\text{origin}} \qquad (16)$$

where

$$n_0 = \max \left[\frac{L}{t_0}, \frac{\sum_{i=1}^{N}(1 - x_i) \cdot M_i}{s_0} \right] \qquad (17)$$

In the multiple-model EAP, the cost of servers, and the streaming and storage capacity must be calculated as a sum over all models in $\mathcal{W}$ instead of a single model v_i. Thus, we modify the expression for $C_{\underline{v}}$ and define $C_{\mathcal{W}}$, the deployment cost for an available set of VoD server models $\mathcal{W}$:

$$C_{\mathcal{W}}(\underline{n}, \underline{x}) = \underbrace{\sum_{i=1}^{N} \alpha \cdot (y_i) + \left[\sum_{j=1}^{W} B(j) \cdot n_i(j) \right]}_{\text{replica}} + \underbrace{(1 - x_i) \cdot M_i}_{\text{transport}}$$

$$+ \underbrace{\alpha + \left[\sum_{j=1}^{W} B(j) \cdot n_0(j) \right]}_{\text{origin}} \qquad (18)$$

where $B(j)$ is the cost of model w_j relative to transport and $n_i(j)$ is the number of servers of model w_j installed at cluster i.

5. HIT RATIO FUNCTION

The fraction of the demand x_i a replica can serve is limited by both the streaming and storage capacity available. It is straightforward to determine an upper bound on the maximum value of x_i based on the amount of streaming capacity installed at a replica i:

$$x_i \leq \frac{n_i \cdot s_i}{M_i} \qquad (19)$$

However, satisfying this inequality is not sufficient to guarantee that a replica can serve a fraction of the demand equal to x_i. We must also ensure that sufficient requests are for files stored in the cache. We assume that the cache is regularly updated and stores the most popular objects. We wish to determine a lower bound on the fraction of all requests from clients in a cluster for objects stored in the cache of its associated replica, a value we will refer to as the hit ratio h_i, based on the amount of storage available. A hit ratio smaller than the fraction of worst-case demand served at any location ($h_i < x_i$) leads to a suboptimal and defective design: available resources at the replica are under-utilized whereas the link to the origin and the origin itself are over-utilized. On the other hand, the scenario where $h_i > x_i$ does not create any issues as the demand is simply re-routed to the origin server using available resources while the replica is utilized at full capacity. Therefore,

this lower bound assures that by installing a certain amount of storage, the replica is capable of serving a minimum fraction of requests $x_i = h_i$ at peak usage time, and the link between the origin and cluster i will not be utilized over its capacity $((1 - x_i)M_i)$ by requests for objects not stored in the cache.

5.1   File access model

The relationship between storage and hit ratio relies heavily on users' behavior. With access to user data (rental statistics from previously deployed VoD systems or other similar services), one can design an h to fit the data. For example, based on a measurement study for a VoD system that offers free content, Yu et al. [2006] claim that video popularity match the Zipf distribution. Here, however, we consider a pay-per-view system where we expect the behavior of the users to be different. We derive a function h based on a file access model that is driven by Zipf's Law, but that takes into account *fetch-at-most-once* and *new arrivals* factors [Gummadi et al. 2003]. Albeit this is just a candidate model for user behavior in the VoD context, it is still a reasonable one: VoD system users rarely access the same file twice and popularity of titles diminishes in time because new files, periodically added to the system, become the most popular titles. Although it is possible to estimate the worst-case hit ratio through simulations, it is impractical to run a full simulation for every evaluation of the cost function $C_{\underline{v}}$, which must be performed thousands of times during the optimization procedure. Our objective is to train a parametric function that provides an estimate of the worst-case hit ratio based on specified system parameters in a few milliseconds compared to the tens of minutes simulations would take.

---

foreach *week* **do**
 Clients are added to the population at a specified rate;
 if *file arrival rate* > 0 **then**
  foreach *new object o* **do**
   Determine insert position of o in library R;
   Recalculate popularity distribution $p(i)$;
  end
 end
 Fill cache with most popular files;
 foreach *client* **do**
  Remove already requested objects from library R_j;
  Recalculate popularity distribution $p_j(i)$;
  Make a request for an object in R_j according to popularity distribution;
 end
 Sort objects in R according to the total number of requests;
 Calculate weekly hit ratio;
end

Algorithm 1: Sequence of events for our discrete-time simulations used to determine our hit ratio function h.

Table I.   Set of values used for parameters of the hit ratio simulation environment.

Parameter		Values
N_W	Number of weeks	$10, 20, 35, 50, 100$
N_P	Number of clients	$100, 1000, 2500, 5000$
N_C	Number of new weekly clients	$0, 25, 50, 100$
X	Cache ratio size	$0.1, 0.2, 0.3, 0.5, 0.7, 0.9$
Y	Initial library size (in thousands)	$1, 2.5, 5, 6.5, 8, 10$
Z	Number of new weekly files	$0, 10, 25, 50, 75, 100$

We design a simulation environment (described in Alg.1) with a library containing Y files and a cache with $X \cdot Y$ files where files are accessed according to the model described by Gummadi et al. [2003]. The users' requests are generated using a Zipf distribution with coefficient $\alpha = 1$. The probability of selecting the file at rank i in library R_j is given by $p_j(i)$:

$$p_j(i) = \frac{i^{-\alpha}}{\sum_{i \in R_j} i^{-\alpha}}$$

Files that have already been fetched by the user cannot be selected again (fetch-at-most-once-model). After every request a user makes, the selected file is removed from his library R_j and file selection probabilities are recalculated. New files are introduced in the library R and each library R_j at a specified rate. The insert position of a file is determined using a Zipf distribution (with $\alpha = 1$); the ranks of existing files which are less popular are decreased and selection probabilities are recalculated. We calculate the hit ratio by dividing the number of requests for objects in the cache by the total number of requests.

5.2 Parametric function training

We ran simulations using a wide range of values for the parameters described in Table I. From our simulation results, we determined that the only parameters that have a significant impact on the hit ratio are the number of files in the library Y, the number of files added every week Z and the cache size ratio X. For the rest of our simulations, we used $N_W = 30, N_P = 1000$ and $N_C = 50$.

We generated 864 points for the hit ratio H by running the simulation four times for 216 possible combinations of X, Y and Z. The worst-case value of H is obtained by taking the minimum value observed during all iterations of the four simulations ($N_W \times 4 = 120$). In Figure 2, we observe the linear behavior of H as a function of $\log(X)$ for different values of Y and Z and propose the form in (20) for our function h, where $0 \le X \le 1$, $1000 \le Y \le 10000$ and $0 \le Z \le 100$. We construct the bilinear functional form for A and B represented respectively by (21) and (22). In Figure 3 and Figure 4, we show the fitting curves generated with those functional forms with the dashed lines (- -) and the actual values of A and B with the markers. The curves fit the markers for most of the sets depicted; the lines for the file arrival rate $Z = 0$ and library size $Y = 1000$ in both figures do not represent the actual value of A and B as accurately as the other curves. We consider that $Y = 1000$ represents a library size smaller than those of interest for large-scale deployment.
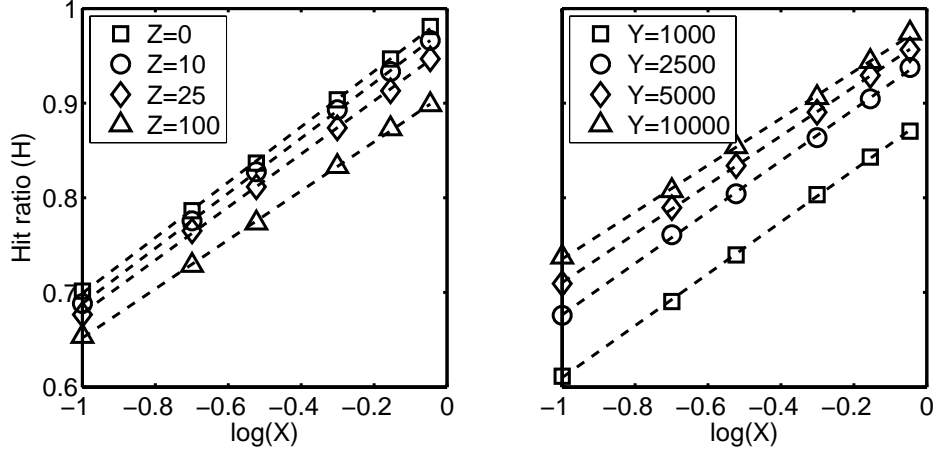
Fig. 2. Data fitting curves to construct the form of the hit ratio function h. The linear curves indicate that $h = A + B\log(X)$ achieves an adequate fit. Markers show values of H and the dashed lines (- -) show the linear fits. We plot the hit ratio H as a function of $\log(X)$ where X is the cache size ratio (Number of files in cache / Total number of files in the library). LEFT: We vary the file arrival rate Z for a fixed library size $Y = 2500$. RIGHT: We vary the library size Y for a fixed file arrival rate $Z = 50$.
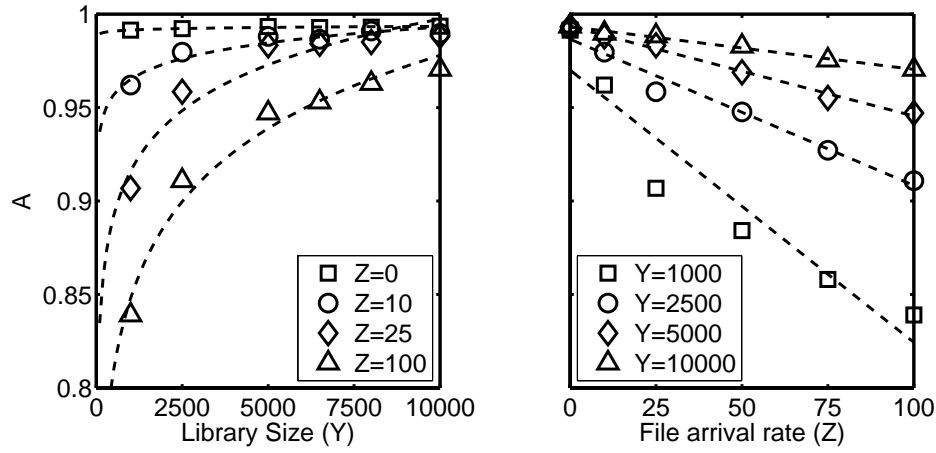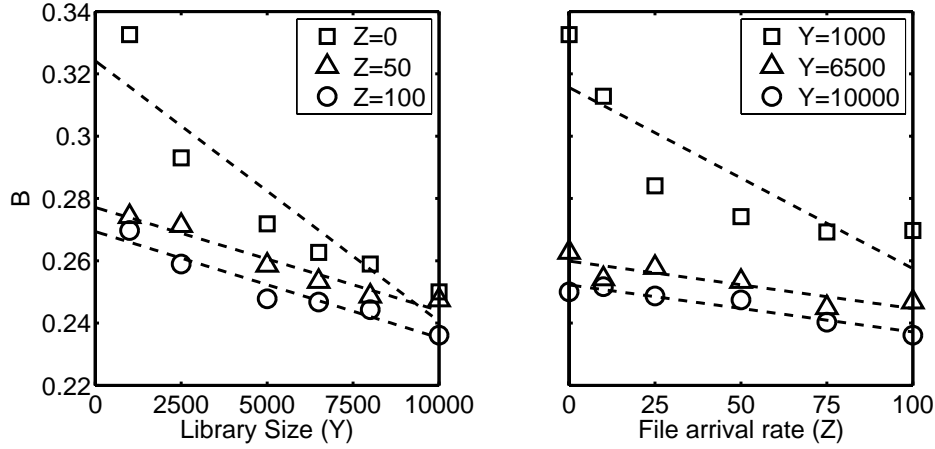

Fig. 3. Data fitting curves to construct the form of A in $h = A + B\log(X)$. Markers show values of A and the dashed lines (- -) show our fit $A = K_1 + K_2Z + K_3\log(Y) + K_4Z\log(Y)$ where Y is the total number of files in the library and Z is the file arrival rate. LEFT: A as a function of the library size Y for different values of file arrival rate Z . RIGHT: A as a function of Z for different values of Y.

$$h = A(Y,Z) + B(Y,Z) \cdot \log(X) \qquad (20)$$

$$A = K_1 + K_2Z + K_3\log(Y) + K_4Z\log(Y) \qquad (21)$$

$$B = K_5 + K_6Z + K_7Y + K_8ZY \qquad (22)$$

Fig. 4. Data fitting curves to construct the form of B in $h = A + B\log(X)$. Markers show values of B and the dashed lines (- -) show our fit $B = K_5 + K_6 Z + K_7 Y + K_8 ZY$ where Y is the library size and Z is the file arrival rate. LEFT: B as a function of the library size Y for different values of file arrival rate Z . RIGHT: B as a function of Z for different values of Y.
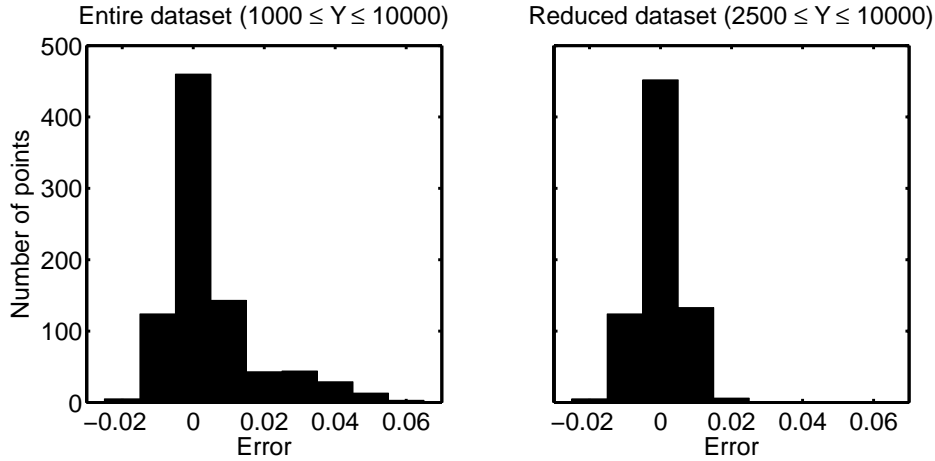
Fig. 5. Histograms of the error $h - H$ between our function h and the value observed during simulations H. LEFT: The error for entire dataset generated by our simulations. RIGHT: The error of a reduced dataset where the values for library size below 2500 are discarded.

We determine the values of the coefficients K_1 to K_8 by solving in the least squares sense the system $KV(X, Y, Z) \cong H$ obtained by substituting (21) and (22) into (20). Our resulting function is accurate, showing less than a 0.02 error eighty-five percent of the time and less than a 0.05 error ninety-nine percent of time. Figure 5 shows the histogram of the error distribution for the entire dataset $(1000 \leq Y \leq 10000)$ for our simulations on the left. On the right, the histogram of a reduced dataset that focuses on the error for library sizes larger than 2500 files. The accuracy of the function estimate for this set is much higher: the error is less

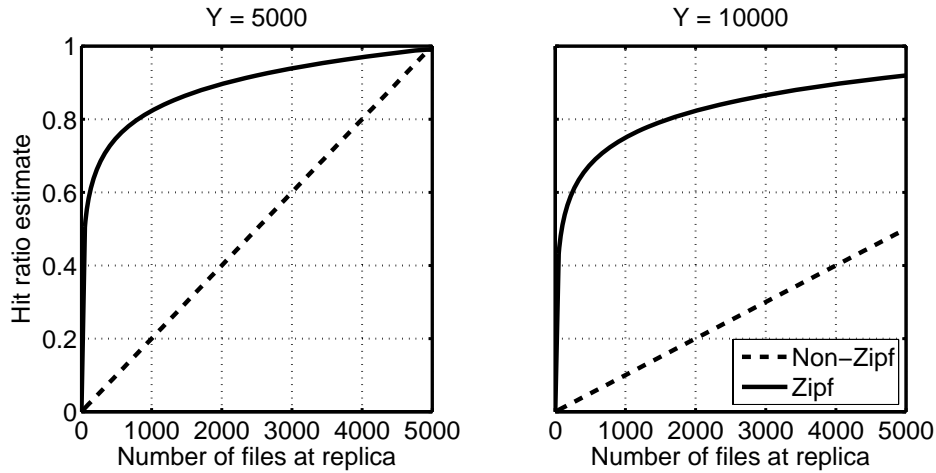than 0.015 ninety-eight percent of the time.

Fig. 6. Estimate of the hit ratio for library sizes of $Y = 5000$ (LEFT) and $Y = 10000$ (RIGHT) as a function of the number of files stored at the replica. We compare our lower bound h (Zipf) to a worst-case scenario where each object has equal popularity $1/Y$ (Non-Zipf). File arrival rate $Z = 0$.

In Figure 6, we compare our popularity model based on a Zipf distribution to a worst-case scenario where each object has equal popularity $1/Y$. This worst-case scenario is a lower bound for hit ratio estimates in the case of file distribution with heavier tail than Zipf. The impact of a library size increase is much more significant in the Non-Zipf case, where a random fraction of the library is stored (whereas the most popular files are stored at the replica in the Zipf case).

6.   SOLVING THE VOD EQUIPMENT ALLOCATION PROBLEM

6.1   Complexity

In this section, we quantify the complexity of the problem we are trying to solve. In Table II, we present the size of the solution space for different network topologies which consist of all possible number of servers and server models at each location. From other experiments, we measured that the machines we used for simulations explore 4000 solutions per second on average, which allows us to estimate to time it would take to explore the entire solution space in order to determine the global optimal solution. From our estimates, it is clear that performing a full search is infeasible as even the smallest problems ($N = 10, W = 1$) can take up to thousands of days to solve depending on the demand and the specifications of the equipement. This shows that heuristics are essential to solve the VoD Equipement Allocation Problem. In the remainder of this section, we describe the heuristics that we developed to solve the single-model EAP (SMEAP) and multi-model EAP (SMEAP) introduced in Sect. 3. As a comparative base for our results, we propose a method to solve a relaxed version of the EAP (REAP) where streaming and storage requirements can be determined independently (no VoD server restrictions).

Table II.   Number of possible solutions for topologies of N locations with W possible VoD server models and estimate of time taken to find the global optimal solution based on an observed average rate of 4000 solutions per second.   Values obtained from 50 different topologies for each pair of (N,W).

N	W	Number of solutions			Estimated time (days)		
		min	median	max	min	median	max
10	1	7.8×10^5	4.5×10^8	9.5×10^{13}	2.2×10^{-3}	1.3×10^0	2.8×10^5
25	1	5.6×10^{14}	7.1×10^{20}	2.4×10^{35}	1.6×10^6	2.1×10^{12}	6.9×10^{26}
50	1	2.0×10^{28}	1.4×10^{40}	1.2×10^{68}	5.9×10^{19}	4.1×10^{31}	3.5×10^{59}
100	1	1.9×10^{60}	5.0×10^{79}	1.3×10^{130}	5.6×10^{51}	1.4×10^{71}	3.7×10^{121}
15	2	9.1×10^{13}	2.0×10^{18}	6.7×10^{23}	2.6×10^5	5.9×10^9	1.9×10^{15}
10	3	5.4×10^{10}	1.0×10^{14}	4.3×10^{17}	1.6×10^2	2.9×10^5	1.3×10^9

6.2   Single-model equipment allocation problem (SMEAP)

To find a near-optimal solution to the single-model EAP, we first determine an expression for x_i in terms of n_i. For any fixed $(\underline{n}, \underline{v})$, we show that by maximizing x_i for all clusters i, we minimize the cost $C_{\underline{v}}(\underline{n}, \underline{x})$. Let $x_i^* \geq x_i$ for $i = 1, \ldots, N$ and $K(\underline{n}, \underline{v}) = \alpha + \sum_{i=1}^{N} \alpha \cdot (y_i) + B_i \cdot n_i$. Assuming that $n_0 = \frac{\sum_{i=1}^{N} (1 - x_i) \cdot M_i}{s_0}$, then for a given pair $(\underline{n}, \underline{v})$ we have:

$$K(\underline{n}, \underline{v}) + \left(\frac{B_0}{s_0} + 1\right) \sum_{i=1}^{N} (1 - x_i^*) \cdot M_i \leq K(\underline{n}, \underline{v}) + \left(\frac{B_0}{s_0} + 1\right) \sum_{i=1}^{N} (1 - x_i) \cdot M_i$$

where $\underline{x}^* = [x_1^*, \ldots, x_N^*]$. So, $C_{\underline{v}}(\underline{n}, \underline{x}^*) \leq C_{\underline{v}}(\underline{n}, \underline{x})$. Because x_i is bounded by streaming and storage limitations, its maximal value is obtained by combining (19) and (20):

$$x_i^* = \min \left[h\left(\frac{n_i \cdot t_i}{L}\right), \frac{n_i \cdot s_i}{M_i} \right] \tag{23}$$

We find a near-optimal solution $(\widehat{\underline{n}}^*, \widehat{\underline{v}}^*)$ through a search procedure consisting of two steps. In the first step, we allow fractional values for n_i to find an initial solution. For each model w_j in $\mathcal{W}$, we find a local minimum to $C_{\underline{v}=w_j \cdot \underline{1}}$ by performing a constrained nonlinear optimization algorithm based on sequential quadratic programming [Fletcher 1987]. Then, we calculate the cost associated with each cluser i (replica + transport) and choose the pair (n_i, v_i), obtained from the optimization procedure, that minimizes this cost for each cluster. We complete the initial solution by determining v_0, the best model of VoD server to install at the origin. In the second step, as $\underline{v}^*$ has been established, we explore the surroundings of the initial solution to find a near-optimal integer solution for $\underline{n}$: we iteratively try to remove or add up to two servers at each location until we find a local minimum $\widehat{n}^*$.

6.3   Multi-model equipment allocation problem (MMEAP)

To solve the MMEAP, we modify x_i and substitute it in (18), the expression for $C_{\mathcal{W}}$:

$$x_i = \min\left[h\left(\frac{\sum_{j=1}^{W} n_i(j) \cdot t_j}{L} \right), \frac{\sum_{j=1}^{W} n_i(j) \cdot s_j}{M_i} \right] \tag{24}$$

where s_j and t_j represent the streaming and storage capacity of model w_j.

We solve the mixed-integer program that we formulated in Sect. 3.2 using the optimization software package CPLEX [cpl 2007]. Since CPLEX does not support nonlinear functions, we linearize h by replacing the log function with a series of tangent hyperplanes.

6.4 Relaxed equipment allocation problem (REAP)

In the REAP, because streaming and storage are independent, we attempt to solve the relaxed version of the problem by first minimizing (15) solely in terms of the fraction of the library stored z_i and then determining the necessary amount of streaming to serve a fraction of the demand equal to $x_i = h(z_i)$.

We developed a routine that performs two searches to obtain a solution to the REAP. The first part consists of determining the uniform amount of storage (equal storage capacity at every location, $\forall i : z_i = z, z \in [0, 1]$) to install such that the cost C is minimized. This is accomplished using a grid search. In the second loop, starting from this initial solution, we iteratively adjust each location i, testing each grid point to determine the z_i that minimizes the cost.

7. RESULTS

7.1 Performance

In this section, we evaluate the performance of our heuristic SMEAP. In our first set of tests, we generated networks with the number of locations $N \in \{1, \ldots, 5\}$ and the number server model $W = 1$ and another series with $N = 3$ and $W \in \{1, 2, 3\}$. We choose small networks to allow comparison with the full search, which cannot produce a solution for larger networks within a reasonable time frame.

In Figure 7, we show the performance of our heuristic by dividing the cost of the solution by the optimal solution provided by the full search. For these small networks, SMEAP performs within 3% of the optimal solution. As we have shown in Sect. 6.1, obtaining the optimal solution through an exhaustive search is only computationally feasible for very small networks, and, for that reason, we use the relaxed version of the EAP (REAP) as a comparative base instead of the full search in the remainder of this paper.

7.2 Simulations

For our simulations, we used a file arrival rate $Z = 50$ and library of $Y = 10000$ files where each file has a size of 2.53 GB (the equivalent of a movie file of 90 minutes with DVD quality compressed with MPEG-4). We used the set of $W = 10$ VoD server models described in Table III where the value of B is calculated with the function presented in Sect. 5. This set of server models allows the possibility of closely matching most values of demand while keeping a reasonable control on the amount of storage. In our simulations, we observe the average hit ratio, the fraction of clusters occupied by replicas and cost relative to the relaxed EAP (REAP) solution
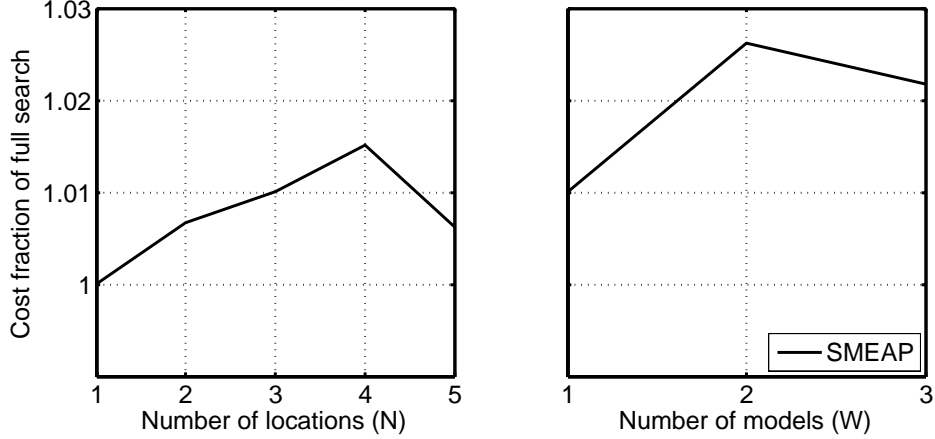
Fig. 7.   Cost ratio between heuristics and the full search averaged over 30 runs.

for different values of the average worst-case demand M, and relative costs α, β and γ described in Table IV.

Table III.   Set of VoD server models used for simulations. For each model, the streaming capacity in Gbps (s) and the storage capacityt in TB (t) is shown.

	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
s	1	2	3	4	5	6	7	8	9	10
t	1	1	1	1	1	1	1	1	1	1

Table IV.   Sets of values used for simulation parameters M, α, β and γ.

Parameter	Simulated values
M	$5, 15, 25, 40, 60, 80, 100$
α	$1, 10, 20, 60$
β	$1, 10, 20, 45$
γ	$1, 10, 20$

We performed tests for $N = 25, 50, 75, 100$ and observed that the number of clusters does not have an impact on our results. Therefore, all results presented in this section were obtained for topologies of $N = 25$ clusters, but these can be generalized to larger networks. Also, we observed that the costs associated with solutions obtained for MMEAP using CPLEX are higher than solutions for SMEAP, which is unexpected as the solution space for the single-model EAP is a subspace of the solution space for the MMEAP. Due to the large size of the solution space, CPLEX's heuristics fail to explore the area where the SMEAP solutions lie. For that reason, we do not present solutions obtained by solving the mixed-integer program that describes the multi-model equipment allocation problem.

In our simulations, we compare scenarios with the same average demand per cluster $\forall i : M_i = M$ (Equal) and cases where the demand is not equal at each
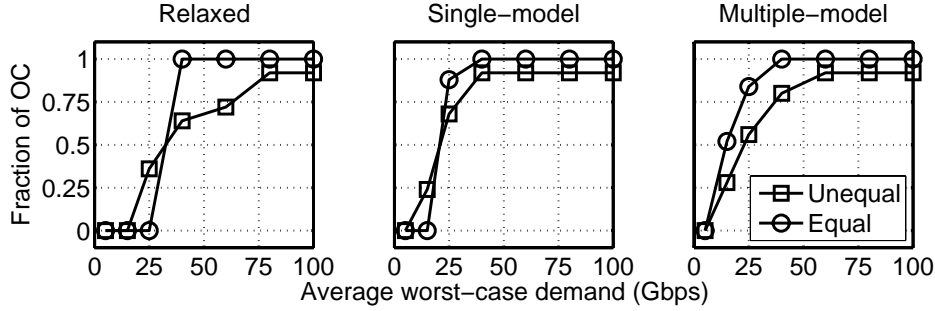
Fig. 8. Fraction of occupied clusters (OC), i.e. where a replica is installed, as a function of the average demand in the network for a topology of $N = 25$ clusters with $\alpha = 10$, $\beta = 10$ and $\gamma = 10$. We observe that for equally distributed demand, the fraction of clusters with a replica is higher.

cluster (Unequal)[1]. In Figure 8, we notice that for cases where the demand is evenly distributed (Equal), the fraction of clusters occupied by a replica is higher and often reaches 1. However, whether the demand is equal for each cluster does not have any impact on the cost of deployment. For the rest of our simulations, we used topologies where the demand is unequal.
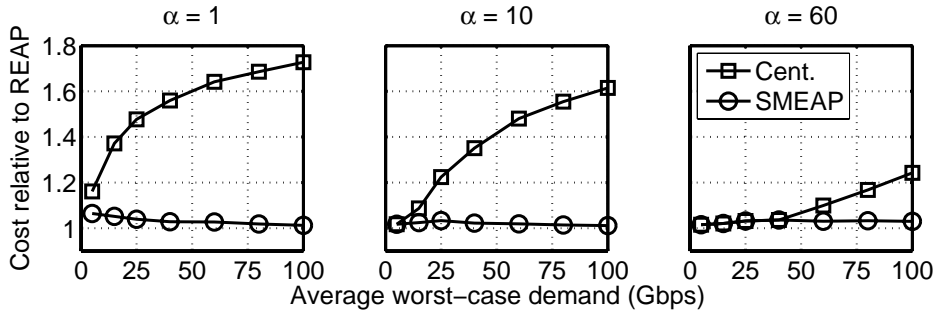
Fig. 9. Ratio between the total cost obtained with Centralized (Cent.) and Single-model Equipment Allocation Problem (SMEAP) heuristics, and the cost for relaxed EAP (REAP) as a function of the average worst-case demand M for $\alpha = 1, 10, 60$ with $\beta = 1$ and $\gamma = 1$.

We now explore the impact of the relative cost of startup α on the performance of our heuristic and the resulting solutions. First, we compare our heuristic to a centralized approach (no replicas installed) using as a reference the solution obtained by our heuristic for solving the relaxed EAP (Figure 9). The SMEAP solution is very close to this lower bound and presents a significant improvement in comparison to a centralized approach. As the relative cost of startup α increases, the gap between the cost of a centralized deployment and our solution decreases; for given values of β and γ, there exists a value of α where a distributed approach is no

---

[1]To determine an unequal demand distribution with average demand M, we draw a vector x uniformly from the $n - 1$-dimensional unit simplex, $\{x \in \Re^n : \sum_i x_i = 1, x_i \geq 0\}$, and then set $M_i = N \cdot M \cdot x_i$.
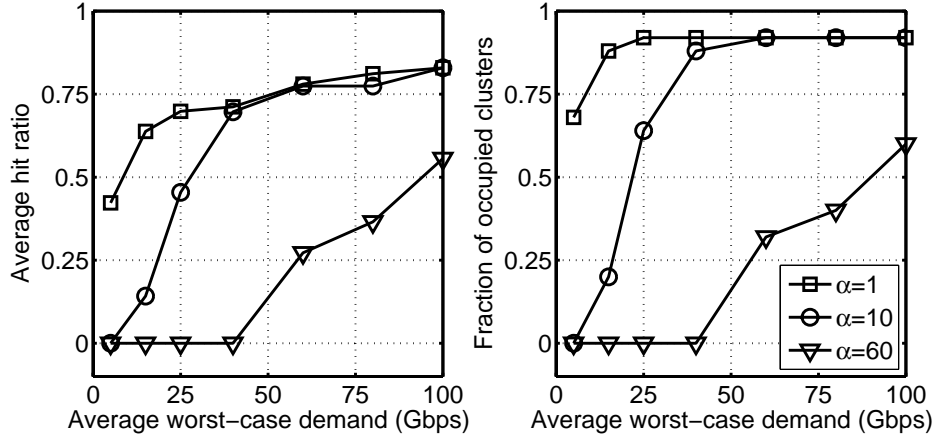
Fig. 10. Average hit ratio (left) and fraction of occupied clusters (right) of solutions (obtained with the single-model heuristic) as a function of the average worst-case demand M for $\alpha = 1, 10, 60$ with $\beta = 1$ and $\gamma = 1$.

longer beneficial from the monetary standpoint. We also note that, for all values of α, the cost of our solutions relative to the simplified scenario is independent of the average demand.

Figure 10 shows the average hit ratio and fraction of occupied clusters for the solutions obtained in the single-model case. These two measures can be used as practical constraints of implementation to, for example, determine the average demand required to achieve a desired hit ratio. Both values increase with the average demand and as the startup cost relative to transport grows, the curve moves down; higher average demand is required is achieve the same hit ratio and fraction of occupied clusters.
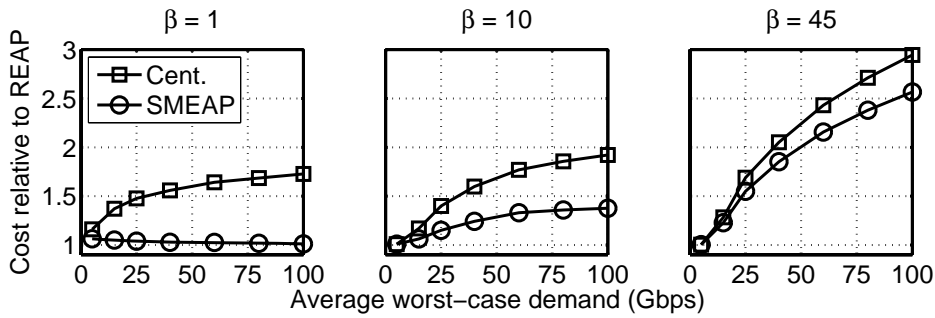

Fig. 11. Ratio between the total cost obtained with Centralized (Cent.) and Single-model Equipment Allocation Problem (SMEAP) heuristics, and the cost for relaxed EAP (REAP) as a function of the average worst-case demand M for $\beta = 1, 10, 45$ with $\alpha = 1$ and $\gamma = 1$.

Figures. 11 and 12 display how the cost of storage relative to transport (β) influences our solutions. Figure 11 illustrates the advantage of using our heuristics over a centralized approach, but this becomes less significant as the value of β increases.
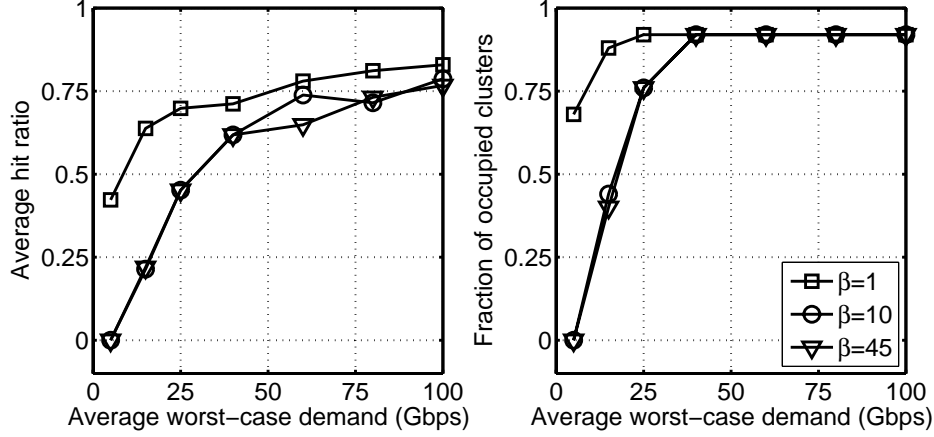
Fig. 12. Average hit ratio (left) and fraction of occupied clusters (right) of solutions (obtained with the single-model heuristic) as a function of the average worst-case demand M for $\beta = 1, 10, 45$ with $\alpha = 1$ and $\gamma = 1$.

The costs of solutions obtained with the centralized approach and of our heuristics both increase with average demand for larger values of β as a result of having storage and streaming determined co-jointly. To maintain the hit ratio for larger demand, additional servers must be installed to increase the streaming capacity. However, this also has the effect of adding unnecessary extra storage capacity (h is independent of the demand). This explains the increasing discrepancy with the relaxed EAP scenario where the streaming can be increased independently. In Figure 12, we notice that for all values of β, the number of replicas installed converges as the average demand increases, whereas for smaller demands, less replicas are deployed when β is greater than one. Although increasing β shifts both curves down, the impact is less dramatic than that caused by increasing α.
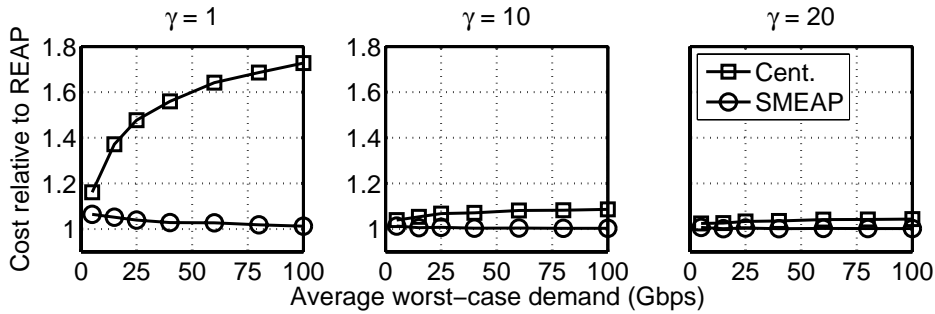

Fig. 13. Ratio between the total cost obtained with Centralized (Cent.) and Single-model Equipment Allocation Problem (SMEAP) heuristics, and the cost for relaxed EAP (REAP) as a function of the average worst-case demand M for $\gamma = 1, 10, 20$ with $\alpha = 1$ and $\beta = 1$.

We observe the impact of increasing γ, the cost of streaming relative to transport, in Figs. 13 and 14. As the average demand and γ increase, the difference between the
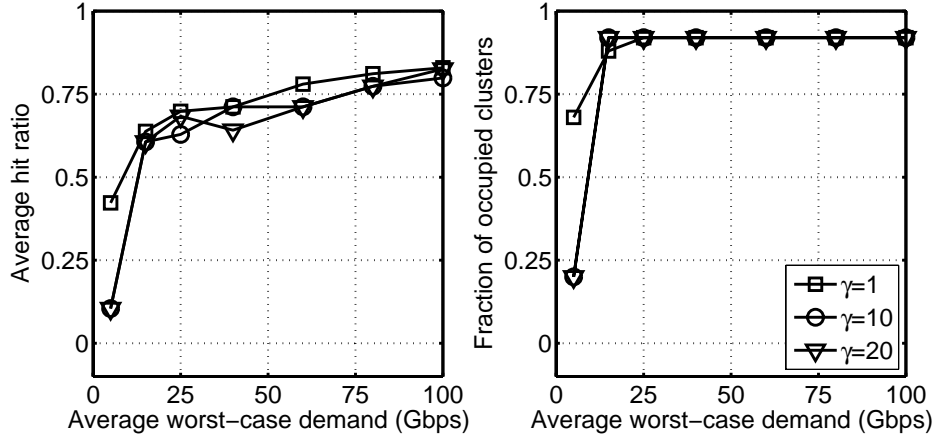
Fig. 14. Average hit ratio (left) and fraction of occupied clusters (right) of solutions (obtained with the single-model heuristic) as a function of the average worst-case demand M for $\gamma = 1, 10, 20$ with $\alpha = 1$ and $\beta = 1$.

costs associated with all solutions (Cent. and SMEAP) and the relaxed EAP cost becomes very small. The total amount of streaming capacity required is identical for all heuristics; no matter how many replicas are deployed, the same amount of streaming is required: $N \cdot M$. For the same reason, the effect of γ on the average hit ratio and number of replicas installed is very minor (Figure 14).
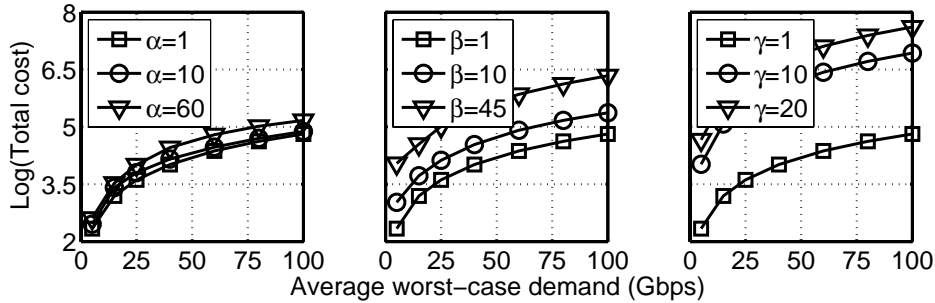

Fig. 15. Total cost obtained with the single-model heuristic on a logarithmic scale as a function of the average demand for $\alpha = 1, 10, 60$ with $\beta = \gamma = 1$ (left), $\beta = 1, 10, 45$ with $\alpha = \gamma = 1$ (middle) and $\gamma = 1, 10, 20$ with $\alpha = \beta = 1$ (right).

In Figure 15, we compare the impact of α, β and γ on the total cost associated with solutions obtained with the single-model heuristic (we show the total cost on a logarithmic scale to compare more easily all three factors). As the average demand increases, the total cost increases with approximately the same rate on the logarithmic scale for all values of α, β and γ. However, we note that increasing the value of γ shifts the cost curve up significantly, whereas increasing α has almost no noticeable effect on the cost. This corroborates the evidence we presented earlier in this section; increasing the cost of storage or streaming relative to transport has

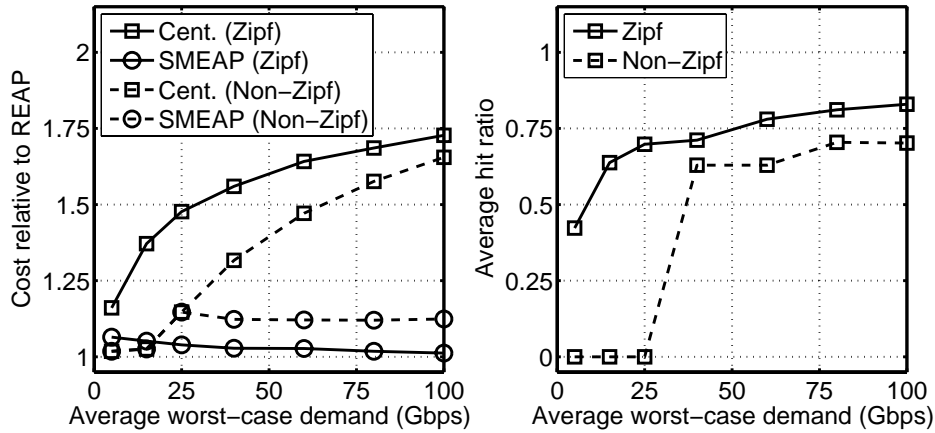a significant impact on the cost associated with the solutions obtained from the single-model heuristic.

Fig. 16. Impact of using a Non-Zipf distribution to calculate the hit ratio on the cost relative to relaxed equipment allocation problem (left) and the average hit ratio (right) for $\alpha = \beta = \gamma = 1$. We compare the values with the results obtained with our model (Zipf).

Figure 16 depicts the impact of using a heavy tail distribution where each file has equal popularity $1/Y$. As expected, the hit ratio that minimizes the cost is higher when using a file access model that takes into account the popularity of each object. Because the cost obtained for the relaxed and single-model EAP is higher in the Non-Zipf case, the relative cost of the centralized approach, not affected by the popularity model, is smaller. Therefore, we can anticipate deployment costs to increase and hit ratios to decrease as libraries grow in size and the fraction of requests for objects with low popularity is higher.

7.3 Design Tool

We developed an interactive design tool, VoD Equipment Allocation Tool, that integrates our network cost model and heuristic to produce near-optimal solutions to the VoD equipment allocation problem. The tool includes two components: the Topology Design Tool (TDT) (developed by Vinokurov [2005]) and our solution. The TDT allows the user to (i) create topologies and models of network components and VoD infrastructures and (ii) visualize the design we generate with our heuristic.

We describe the typical workflow to follow to design a VoD network with the tool. The first step is to create the network topology with all the locations using the TDT Wizard or manually. The second step is building models for network components, VoD equipment (VoD server models) and the VoD network itself. At least one model (cost and specifications) needs to be created, using the model editor, for each of the following components before adding infrastructures to the topology: network interface, DWDM switch, fiber, stored file, VoD server, library server (or origin) and replica server. Once the topology and the models have been generated, the user can create replicas and origin objects using a built-in wizard. A
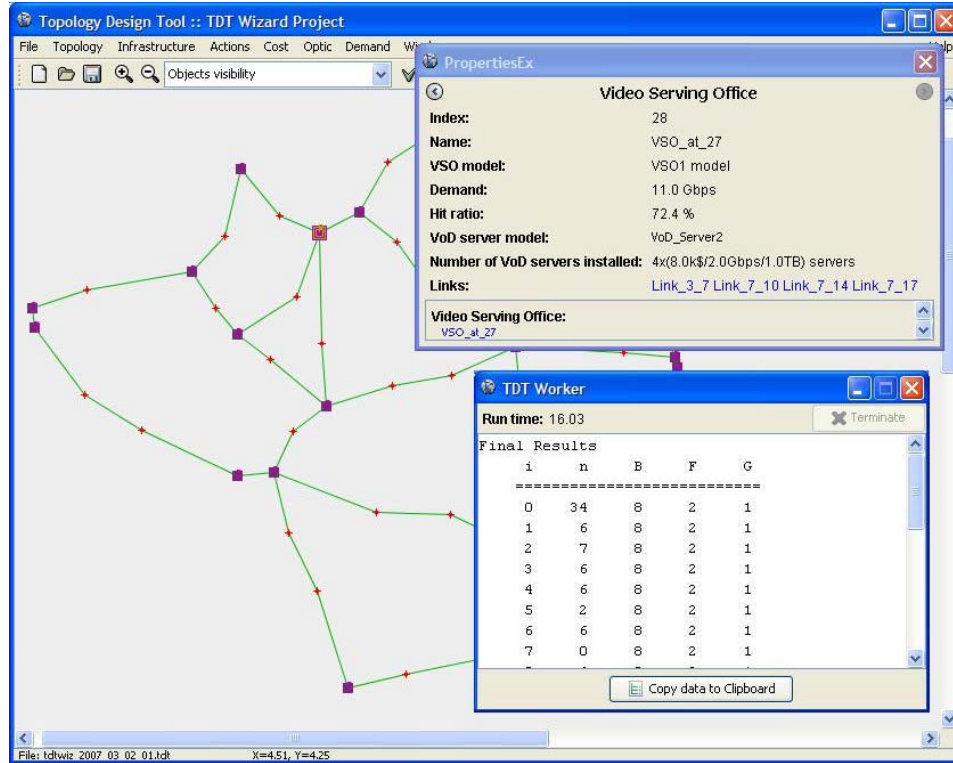
Fig. 17. Interactive design tool that integrates our network cost model and SMEAP heuristic to solve the Video-on-Demand equipment allocation problem. The tool allows the creation of custom topologies, equipment models and VoD networks. Shown in the figure, a randomly generated topology, the properties of one replica server after the optimization procedure (demand, hit ratio, equipment installed), and the results from our heuristic displaying the number, cost, streaming and storage capacity of the servers deployed at each location.

valid VoD network includes only one origin and any number of replicas (up to one per location). With a valid network setup, it is possible to run the optimization program (a MATLAB-compiled executable) to determine near-optimal equipment.

8. CONCLUDING REMARKS

In this paper, we addressed a resource allocation problem associated with Video-on-Demand networks. More precisely, we defined the *VoD equipment allocation problem* (EAP) as determining the number and model of VoD servers to install at each cluster of clients in the topology to minimize the deployment cost while serving the entire demand. We formulated the problem as an optimization problem and proposed a model for the network cost that includes costs related to startup, streaming, storage and transport. We trained a parametric function to map the amount of storage at a server to an estimate of the hit ratio based on a file access pattern proper to VoD networks. Because of the complexity of the problem and the size of the solution space, we developed heuristics to find a near-optimal solution

to the VoD EAP and two variations, the relaxed EAP and the multi-model EAP.

We showed that the performance of our heuristic for the single-model EAP does not depend on the number of cluster of clients in the topology or whether the demand is evenly distributed among all locations or not, but rather on the average demand per cluster. Given the cost of startup, streaming and storage relative to transport, it is possible to determine the average hit ratio and fraction of clusters that require a replica server as a function of the average demand in the network. In practice, this is useful for providers who wish to establish the minimum average demand a cluster must generate in order to achieve a certain hit ratio. This can be considered as measure of performance when it comes to deploying not only cost-efficient but also easily-manageable networks.

Through our simulations, we observed the impact of having to determine the amount of storage and streaming co-jointly through the VoD servers by comparing our results with the simplified EAP. If the file access in VoD is Zipf-like, then relatively small amounts of storage are required to ensure a high hit ratio. Most VoD servers provide a disproportionate amount of storage capacity relative to their streaming capabilities. This leads to the installation of unnecessary storage capacity and higher deployment cost (which we have shown by increasing the cost of storage relative to transport). The results obtained with the multi-model were disappointing; we anticipated lower cost solutions due to the greater flexibility in matching the storage and streaming requirements of each cluster. Even if it is possible to obtain better solutions by modifying our heuristic, we now expect the improvement relative to the single-model case to be marginal and not worth the complexity involved in deploying such a network.

Although proxy-based VoD networks offer important cost savings, their future is uncertain because hybrid peer-to-peer architectures offer better scaling capabilities. In the near future, however, the proxy-based architectures are more mature and the most frequently deployed. It is important to investigate multi-stage deployments, where equipment is added incrementally over time. Optimizing such deployments requires prediction of the variations in demand and costs through time. Intuitively, because the costs of streaming and storage decrease, it is beneficial to wait as long as possible before deploying any equipment. Another challenge is to adapt the mapping between the amount of storage and the hit ratio to larger libraries. As libraries grow in size, there will be a larger portion of requests for less popular objects (the *long tail* of content), making it difficult to maintain the hit ratio without substantially increasing the amount of storage.

REFERENCES

2007. ILOG CPLEX. `http://www.ilog.com/products/cplex/`.

ALMEIDA, J. M., EAGER, D. L., VERNON, M. K., AND WRIGHT, S. 2004. Minimizing delivery cost in scalable streaming content distribution systems. *IEEE Trans. Multimedia 6*, 356–365.

CORNUEJOLS, G., NEMHAUSER, G., AND WOLSEY, L. 1990. The uncapacitated facility location problem. In *Discrete Location Theory*, P. Mirchandani and R. Francis, Eds. Wiley, 119–171.

COUCH, K. 2005. Raising the bar for triple play with VoD. *Converge! Network Digest.* `http://www.convergedigest.com/blueprints/ttp03/2005nortel1.asp?ID=189&ctgy=Headend`.

FLETCHER, R. 1987. *Practical Methods of Optimization.* John Wiley and Sons, New York, NY.

GUMMADI, K. P., DUNN, R. J., SAROIU, S., GRIBBLE, S. D., LEVY, H. M., AND ZAHORJAN, J.

2003. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *Proc. ACM Symp. Operating Systems Principles (SOSP)*. Bolton Landing, NY.

KARLSSON, M., KARAMANOLIS, C., AND MAHALINGAM, M. 2002. A unified framework for evaluating replica placement algorithms. Technical report, Hewlett-Packard Laboratories.

KIM, S.-J. AND CHOI, M. 2003. A genetic algorithm for server location and storage allocation in multimedia-on-demand network. In *Proc. Symp. Trends in Communications*. Bratislava, Slovakia.

KRISHNAN, P., RAZ, D., AND SHAVITT, Y. 2000. The cache location problem. *IEEE/ACM Trans. Networking 8*, 568–582.

LAOUTARIS, N., ZISSIMOPOULOS, V., AND STAVRAKAKIS, I. 2005. On the optimization of storage capacity allocation for content distribution. *Computer Networks Journal 47*, 409–428.

MARKMAN, J. 2006. 2007 is showtime for video on demand. `http://articles.moneycentral.msn.com/Investing/SuperModels/2007IsShowtimeForVideoOnDemand.aspx`.

MASA, M. AND PARRAVICINI, E. 2003. Impact of request routing algorithms on the delivery performance of content delivery networks. In *Proc. Int. Performance Computing Communications Conf. (IPCCC)*. Phoenix, AZ.

MUNDUR, P., SIMON, R., AND SOOD, A. 2004. End-to-end analysis of distributed Video-on-Demand systems. *IEEE Trans. Multimedia 6*, 129–141.

NGUYEN, T., CHOU, C., AND BOUSTEAD, P. 2003. Resource optimization for content distribution networks in shared infrastructure environment. In *Proc. Australian Telecommunications Networks and Applications Conf.* Melbourne, Australia.

TANG, W., WONG, E., CHAN, S., AND KO, K. 2004. Optimal video placement scheme for batching vod services. *IEEE Trans. on Broad. 50*, 16–25.

THOUIN, F. AND COATES, M. 2007a. Video-on-Demand networks: design approaches and future challenges. *IEEE Network - Special Issue on Convergence of Internet and Broadcasting Systems 21*, 42–48.

THOUIN, F. AND COATES, M. 2007b. Video-on-Demand server selection and placement. In *Proc. of International Teletraffic Congress (ITC)*. Ottawa, Canada.

THOUIN, F., COATES, M., AND GOODWILL, D. 2006. Video-on-Demand equipment allocation. In *Proc. IEEE Int. Conf. Network Computing Applications (NCA)*. Cambridge, MA.

VINOKUROV, A. 2005. Tools for optical networks design. In *Proc. European Next Generation Internet Design and Engineering (EURO-NGI)*. Rome, Italy.

WANG, B., SEN, S., ADLER, M., AND TOWSLEY, D. 2002. Optimal proxy cache allocation for efficient streaming media distribution. In *Proc. IEEE Infocom*. New York, NY.

WAUTERS, T., COLLE, D., PICKAVET, M., DHOEDT, B., AND DEMEESTER, P. 2005. Optical network design for video on demand services. In *Proc. Conf. Optical Network Design and Modelling*. Milan, Italy.

WU, L.-Y., ZHANG, X.-S., AND ZHANG, J.-L. 2006. Capacitated facility location problem with general setup cost. *Computers Operations Research 33*, 1226–1241.

YANG, M. AND FEI, Z. 2003. A model for replica placement in content distribution networks for multimedia applications. In *Proc. IEEE Int. Conf. Communications*. Anchorage, AK.

YU, H., ZHENG, D., ZHAO, B., AND ZHENG, W. 2006. Understanding user behavior in large-scale video-on-demand systems. In *Proc. ACM Eurosys*. Leuven, Belgium.