JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION
Vol. 9, No. 4, December, pp. 314–335, 1998
ARTICLE NO. VC980400

A Practical Approach to Resource Allocation in Video-on-Demand Servers[1]

Emmanuel L. Abram-Profeta and Kang G. Shin

Real-Time Computing Laboratory, Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Michigan 48109-2122
E-mail: abram@eecs.umich.edu, kgshin@eecs.umich.edu

Received January 20, 1998; accepted September 9, 1998

This paper addresses the problem of organizing resources in large video-on-demand (VoD) front-end servers, so a large collection of movie titles and concurrent channels can be made available at the lowest possible cost, and with the highest admission quality of service (QoS). Coarse-grained striping (CGS) in disk arrays is shown to be a viable candidate due to its cost-effective storage utilization and low admission latency. We study the feasibility of storage organizations in which disks are partitioned into several clusters to lessen service disruptions during reconfigurations. In parallel with this, we present an algorithm for optimal video allocation across clusters, and show that high service availability can be achieved cost-effectively with minor side-effects on customers' admission latency. We also identify specific situations when storing one movie per disk performs acceptably well. Finally, we show that full-fledged VCR functionality can be provided at reasonable cost.

Key Words: **Deterministic video-on-demand (D-VoD); true video-on-demand (T-VoD); disk arrays; coarse-grained striping; cost; nonstationary arrivals.**

1. INTRODUCTION

In the traditional pay-per-view (PPV) service provided by broadcast cable companies, subscribers "tune in" to prescheduled programs, on which they have no control. The most basic improvement over this type of service is to provide users with VCR playback of rental programs. A service is called "deterministic" VoD (D-VoD) if no further customer interaction is allowed after a program starts, because movies are of fixed length, and hence, customers receive deterministic service. A service is known as "true" VoD (T-VoD) if it provides VCR functionality [1] (e.g., stop, pause, fast-forward capability). In between these two extremes, variations on the basic pay-per-view scheme depending on the degree to which VoD services make use of user interactivity have been proposed in the literature [1].

The operation of a VoD server comprises two steps. First, upon receiving a request, the server must be able to reserve resources and decide if the request can be honored. This is the *program scheduling* phase, in which the VoD server uses an *admission control* algorithm to decide when to start program transmission, based on such constraints as the available capacity of the movie archive on disks or disk arrays in compressed format. Once a service has been scheduled, run-time operation of the VoD server will ensure continuous and seamless playout by retrieving in real-time and delivering a stream of compressed video data. Video data is then decompressed by the customer's premise equipment (CPE), modulated to a conventional television carrier, and displayed on a standard TV. This is the *servicing* phase.

This paper focuses on the program scheduling phase. During the scheduling phase, customers' QoS can be expressed in terms of the waiting time before receiving service, or *admission latency,* and the defection rate due to long waits. Also, adding extra resources, changing video material, or in the event of disk failures, taking resources off-line for reconfiguration should incur minimum disruption of service to existing connections. Thus, customer's QoS is also affected by service availability, or minimum support provided by the VoD server in case of reconfiguration. From the VoD server's point of view, the ideal storage architecture must be continuously available and expandable and ensure that load is evenly balanced. Customers' QoS can thus be seen as a combination of (1) *scalability,* defined as the ability to accommodate significant workload fluctuations and overloads without affecting admission latency, and (2) *versatility,* defined as the ability to reconfigure the VoD server with minimal disturbance to service availability during these changes. A high level of versatility is also desirable for expandability, to allow the addition of new devices. When designing a VoD server,

[1] The work reported in this paper was supported in part by the National Science Foundation under Grant MIP-9203895 and the Office of Naval Research under Grant N00014-94-1-0229.

1047-3203/98 $25.00

one must deal with several constraints on resource allocation to provide all of these features. First, a VoD server has to offer a predetermined selection of movie titles and support nonuniform access to these movie titles. Access locality in video services, caused by customers' specific demand patterns, is best described by a commonly used criterion for partitioning movie titles, known as probability of movie access, or ''popularity'' [1]. In addition to movie selection, the maximum number of concurrent channels and storage organization also determine the cost of a VoD server.

Our objective is to specify storage organization and long-term resource allocation in distributed VoD servers so that a large collection of movie titles and channels can be accessed with the highest QoS and at the lowest cost. In case of distributed VoD service, customers typically interact locally with front-end VoD servers. Thus, it is at this level that customers' QoS will be inferred from the storage organization. We will therefore first overview two basic, well-understood types of storage technology, specifically chosen for their distinct ways of allocating resources among different movie titles, to illustrate their respective limitations. The first type ''completely'' partitions the storage among different movie titles. An example of such organizations, called *completely partitioned* (CP) organizations, may be found in small-scale VoD servers which store one movie title per disk (OMPD). The second type completely shares the storage among different movie titles. An example of such partitions, called *completely shared* (CS) organizations, is VoD servers which store movie titles using fine-grained striping (FGS) or coarse-grained striping (CGS) [2] of videos across disks in order to effectively utilize disk bandwidth.

These forms of storage constitute, to some extent, the two extreme cases. CP organizations typically trade availability—disks can fail or be brought off-line for update without affecting the entire service—for increased latency and inefficient use of storage capacity and bandwidth whenever data access is skewed towards particular disks. CS organizations, on the other hand, ensure a very low latency and high storage utilization, but reconfiguration risks the availability of the entire VoD server. Thus, it is reasonable to expect that at the local level, the storage consists of several clusters and combines both storage models into a hybrid organization. Each cluster will then serve as a repository for a subset of all movie titles, for which it will provide a fraction of the total channel capacity. In this paper, we shall therefore address how to partition a CS organization into clusters, how to choose clusters' capacity, and how to assign videos to separate devices so that customers' QoS may be maximized at minimal cost.

The paper is organized as follows. We first focus on D-VoD servers because, while being the easiest system to understand and the first implemented in research labs, deterministic service is *the minimum service one can expect from a video server.* As background, we outline a generic metropolitan VoD architecture and present the two basic storage organizations. We then compare OMPD, FGS, and CGS in terms of admission latency and cost to provide a given channel capacity and movie selection. In the next section, we investigate the increase in cost and customers' latency that enable disk arrrays to be clustered to provide customers with higher service availability. Given that VoD service providers target at the home consumer market, we study resource allocation in various realistic scenarios including customers' partial patience, and time variations in the request arrival pattern. We show that OMPD, although often viewed as a naive approach to storage configuration, may be economically acceptable for highly reliable service of a small number of movies to a large user population. Last, since customers' interactive behavior after admission affects the video retrieval rate and the holding time of VoD server's resources, we adapt our methodology to T-VoD service. The paper concludes with Section 8.

2. HIERARCHICAL VoD SERVICE

Advances in video compression, video servers, and networking have stimulated a frenzy of alliances between cable providers, movie studios, telephone companies, service providers, and equipment manufacturers. Seen today as one of the most lucrative new markets emerging from such corporate activities, large-scale digital video service has the potential to replace neighborhood videotape rental stores and provide convenient and unrestricted access to a large collection of movie titles. Considering the costs in networking, storage, and customers' premise equipment (CPEs), a mass market is needed for the technology to be viable. To support such deployment (e.g., 10,000–100,000 customers viewing 2,000–20,000 movies), it is usually recognized [1, 3–7] that a hierarchical video-on-demand (VoD) server architecture is preferable for scalability and economical viability.

In the generic metropolitan architecture depicted in Fig. 1, service providers manage their own set of video materials, diseminated from one or more archives to a set of front-end local video servers via high-speed regional and backbone networks which are likely to be an ATM variety. Movie archives serve as a repository for a large collection of movie titles, compressed and stored in video databases on secondary and possibly tertiary storages. These archives are used by service providers to periodically update local video servers during off-peak hours [1]. Caching programs in secondary storage in local VoD servers avoids the high cost of network infrastructure and core network bandwidth demands that would otherwise incur when a large number of customers are served from a single point [3]. Service

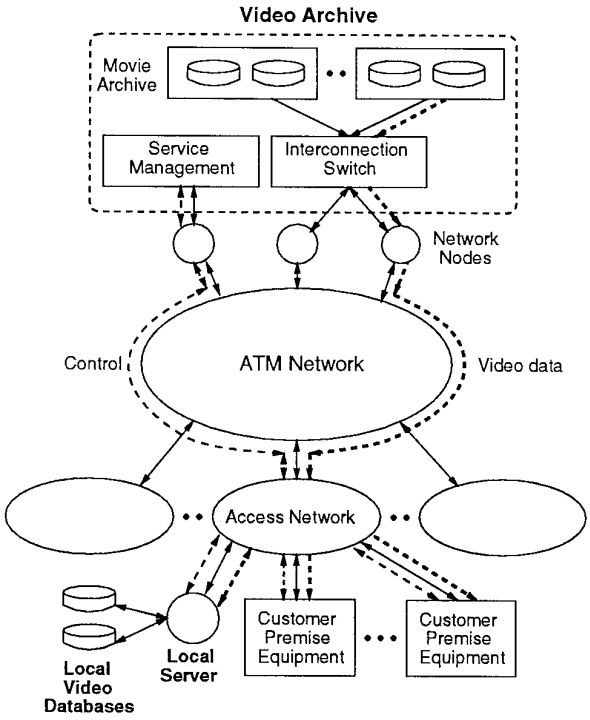

FIG. 1. A metropolitan video-on-demand system architecture.

providers are also responsible for (i) balancing loads among local VoD servers and (ii) video migration between different levels of the hierarchy to maximize the probability of honoring a customer's request locally. One can take advantage of locality and time variations in request arrival rates in conjunction with high-speed networking to improve customers' service by storing frequently accessed videos on faster, more expensive storage while keeping less frequently requested movies on slower, less expensive archives [3, 5–7]. In the rare case of requests for movie titles not available locally, nearby local VoD servers can be queried. Alternatively, data can be transferred at high speeds from the closest archive and cached locally for playout.

Local VoD servers are responsible for billing and connection management during the scheduling phase and real-time retrieval and delivery of digital bit streams containing compressed full-motion video data during the servicing phase. Customers typically interact with local VoD servers via an access network, which requires a technology that will allow several Mbps to be transmitted to individual homes, since typically, VHS-quality MPEG-1 and HDTV-quality MPEG-2 channels require from 1.5 to 16 Mbps. Possible alternatives include asymmetric digital subscriber line (ADSL), fiber-to-the-home (FTTH), fiber-to-the-curb (FTTC), and hybrid-fiber-coax (HFC). ADSL has the advantage that modems can attach to a standard telephone line and thus be used by businesses that do not have cable TV service to offer interactive services. The required modem technology at each end of the connection is, however, currently very expensive. In addition, even though the maximum input data rate has been increased from 1.5 Mbps in 1993 to 50 Mbps in 1995 [8], a maximum output data rate of 640 Kbps makes the system unsuitable for jobs where large amounts of data must flow in both ways. FTTH and FTTC require running high-speed fiber optics all the way into individual homes or to curbside vaults that serve a dozen or so households. These two options are prohibitively expensive, especially considering the billions of dollars already spent on the traditional commercial television cable networks based on coaxial cables and the fact that the network hardware costs would not be shared.

The third alternative, HFC, is widely recognized as the emerging standard for both cable and telephone companies as it allows one to use fiber-optic cables to carry signals to large groups of homes (e.g., 500), yet still use the existing coaxial cable to carry those signals into each home individually. Thanks to emerging coaxial cable systems, cable modem technology, and quadrature amplitude modulation (QAM) of digital information onto analog carriers, HFC will eventually allow the existing cable plant to be upgraded to digital transmission and serve 200- to 1,000-household neighborhoods with up to 500 MPEG-2 digital channels, in addition to the existing capacity of 65 analog channels [8], while leaving some bandwidth available to return, control, and personal communication channels. It may be argued that bidirectional communication will cause congestion periods similar to those commonly observed over the Internet when several interactive users transmit large data files simultaneously. Nevertheless, in interactive VoD, bidirectional communication is needed only for sporadic exchanges of short control messages between subscribers and service providers. In summary, while ADSL can be seen as an intermediate support solution, HFC is rapidly becoming suitable for VoD applications until we have full-optic fiber upgrades.

In the rare case of requests for those movie titles not available locally, nearby local VoD servers can be queried. Alternatively, data can be transferred from the closest archive and cached locally for playout. For the most popular movies, customers' QoS during the scheduling phase is therefore affected by the architectural decisions for the choice of a storage medium and resource allocation in local VoD servers. As mentioned in Section 1, the choice of a VoD storage organization depends on both service versatility and scalability. Multicast VoD systems have been recently proposed [9] to address these issues by "batching" requests made by many different viewers for the same movie within a *batching period*, thereby reducing competition for resources. Our focus in this paper is on local VoD servers which allocate one channel per customer's request, as these systems are increasingly being deployed and exper-

imented with. Also, as we shall see, such systems may be upgraded to T-VoD service more easily than multicast VoD servers [10].

3. TWO BASIC STORAGE ORGANIZATIONS

3.1. Optimal Concurrent Access Profile in CP Organizations

CP organizations partition storage among movie titles by storing one movie per disk (OMPD). While not extremely attractive in terms of cost compared to other more sophisticated storage technologies, OMPD has advantages such as reliability, since when a disk fails, only one copy of the movie becomes unavailable. Currently, a disk has the capacity to hold 1 or 2 movies in compressed format (1–9 GB), while it has the throughput to allow 5 to 20 viewers to simultaneously access its contents (30–60 Mbps). Thus, if a video is not replicated on several disks and is entirely stored on a single disk, the number of concurrent video streams that can be supported is bounded above by the disk bandwidth. Popularities will therefore dictate the number of disks that should be allocated to each individual movie title, hence the partition of the concurrent channel capacity among movie titles. (A more popular movie will be replicated and stored on multiple disks.)

Let d denote the number of disks to be partitioned among N movie titles, R the bandwidth required per stream, B the bandwidth of a single disk in number of video streams, S the storage required per movie, and $C_d > S$ the capacity of a single disk. For a storage organization supporting N different movies, popularity can be defined as a vector of access probabilities $[p_1, p_2, \cdots, p_N]$, $\sum_{m=1}^{N} p_m = 1$, where p_m is the probability that a client will request movie m (e.g., Zipf's popularity model, introduced in Section 4.2). Thus, if the aggregated request arrival process is Poisson with parameter λ (or $\lambda(t)$ in the case of time-dependence)—as has traditionally been used in telephony—then, according to the result of splitting a Poisson process [11], the request arrivals at movie title m are a Poisson process with parameter $\lambda_m = p_m\lambda$. The traffic intensity is given by $\rho = \lambda/s\mu$.

Consider the optimal partitioning of the disk capacity of a VoD server among movie titles requiring an identical playout rate. A reasonable partitioning policy is proportional allocation, which attempts to allocate $d_m = p_m d$ disks to movie title m. However, as $p_m d$ is usually not an integer, the video server may be forced to allocate $d_m = \lceil p_m d \rceil$ for $m > 1$ and $d_1 = d - \sum_{m=1}^{N} d_m$. The major problem with this "blocking effect" is that the *maximum sustainable traffic intensity*—which is the maximum value of $\rho = \lambda/s\mu$ sustainable while keeping the system stable—will be less than 1, as the proportional allocation policy may lead to a traffic intensity $\rho_m \geq 1$ for movie title m. In order to reduce the blocking effect, it is possible to specify an *optimal concurrent access profile* (OCAP), which is the partition $[d_1, \cdots, d_N]$ of d disks among N movies that minimizes the customers' average waiting time. OCAP specification determines, for a given traffic intensity, all the stable partitions, and selects the one corresponding to the highest maximum sustainable traffic intensity.

Our approach in calculating the waiting time is to model the partitioning of VoD storage for a set of movie titles, as a group of N parallel queueing systems of type $M/D/c$,[2] denoted by $M/\mathrm{D}/d_m.B$, $m = 1, \cdots, N$. If the VoD server does not support any interactivity, the service will be provided deterministically at rate $\mu = 1/L$, where L is the average length of movies. For simplicity, we assume that all movie titles are of an identical length, since most movie titles are known to have playback lengths of about 90–100 minutes. While a Gaussian or uniform distribution between, say, one and two hours might provide a better match, it will not affect the general conclusions drawn from our analysis. Denoting by $\mathrm{EW}(M_{\lambda_m}/\mathrm{D}/d_m.B)$ the average waiting time for each virtual queueing system $M_{\lambda_m}/\mathrm{D}/d_m.B$ of type $M/\mathrm{D}/c$ with Poisson arrivals at rate λ_m, deterministic service, and $d_m.B$ servers, our optimization problem can be stated as:

Problem VoD OCAP. Given d concurrent disks, $\bar{d} = [d_1, \cdots, d_N]$ as partition of the disks among $N > d$ different movie titles or groups of movie titles, λ the aggregated request arrival rate, μ the average service rate, $[p_1, \cdots, p_N]$ the movie popularity vector, and $\lambda_m = p_m\lambda$ the arrival rate at each $M/\mathrm{D}/d_m.B$ queue corresponding to movie title m, *determine*

$$\min_{\bar{s}} \sum_{m=1}^{N} p_m \mathrm{EW}(M_{\lambda_m}/\mathrm{D}/d_m.B) \quad \text{such that} \quad \sum_{m=1}^{N} d_m = d.$$

This optimization problem can, in general, be solved using an efficient enumeration technique as shown in [14]. Also, $\mathrm{EW}(M_{\lambda_m}/\mathrm{D}/d_m.B)$ can be accurately inferred from the $M_{\lambda_m}/M/d_m.B$ queueing system with exponentially distributed service times using a technique similar to the one in [15].

3.2. CS Storage Organizations

CS storage organizations completely share the storage among different movie titles by striping videos across *disk farms* or *disk arrays* in order to effectively utilize disk bandwidth. Unlike OMPD, in which the maximum number of concurrent channels for a particular movie title is deter-

[2] This notation reflects standard queueing theory nomenclature [12, 13], whereby the first letter M stands for a Poisson arrival process, the second letter D indicates that the probability distribution of the service times is deterministic, and the last number corresponds to the number of servers.

mined by the number of disks allocated to that particular movie title, in CS organizations each disk can devote its bandwidth to playing out whichever movie has been requested. Thus, unwatched movies require disk capacity but do not waste disk bandwidth, and the concurrent channel capacity is fully available to any movie title. In this paper, we consider striping over disk arrays which, with specialized software and/or hardware controllers, distribute consecutive logical data units on consecutive disks in a round-robin fashion. Two popular schemes are presented in [2]. In *fine-grained striping* (FGS) (similar to RAID-3), the stripe unit (or block size) is relatively small (e.g., a byte) and every retrieval involves all d disks that behave like a single *logical* disk with bandwidth dB. In coarse-grained striping (CGS), each retrieval block consists of a large stripe unit which is read from only a single disk, but different disks can simultaneously handle independent requests. CGS with parity information maintained on one or several dedicated disks corresponds to RAID-5 [16, 17].

We shall henceforth restrict our study to constant bit rate (CBR) media data whose bandwidth requirement remains fixed throughout the entire display of a video. In both striping schemes, due to the periodic nature of video clips, we assume that data for videos in service is retrieved in fixed-length ''rounds,'' using the C-SCAN seek optimizing disk scheduling algorithm [2]. Unlike other disk-scheduling alternatives [18, 19], C-SCAN eliminates random seeks to arbitrary locations by sorting videos and traversing disk surface from the innermost pending request to the outermost pending request. In disk-array-based VoD servers, new customers are served by transferring their requests from a request list to a service list. Unlike FGS, in which only one service list is needed, in CGS one service list is maintained for each disk. If the number of stripe units per video is a multiple of the number of disks[3] the storage server operation at the end of each round is to set the service list for every disk to that of its preceding disk. Otherwise, the *current available bandwidth* scheme presented in [2] is used for starvation-free disk bandwidth allocation. New customers are served by transferring their requests from a request list to the service list of the first disk. For each video, consecutive stripe units of size corresponding to the duration of a round are stored on the d disks in *round-robin fashion*. Retrieval of each video stream then proceeds in lock-step, that is, each client accesses exactly one disk during each round, and consecutive disks are accessed by the same set of clients during successive rounds.[4] Whatever the case may be, the schemes presented in this paper can deal with VBR data by using standard techniques such as pushing data at constant rate R and allocating sufficient buffer space at the client to smooth variations with respect to R.

Differences in degree of concurrency and parallelism affect storage capacity, and therefore, performance modeling and cost evaluation. A high degree of parallelism in FGS eliminates the need for replication, as conjectured in [1], but at the cost of d times higher latency overhead per data access. Effectively, the storage system can be seen as having only one arm, thus resulting in longer rounds, higher latency and higher RAM costs. In CGS, the number of concurrent accesses to a particular movie title is limited by the load on the disk holding the first video segment, but high-degree concurrency is better adapted to retrieval of material such as videos, with relatively low bandwidth in comparison with the bandwidth of a single disk (each disk having enough bandwidth to serve several videos). Consequently, it is proved in [2] that the retrieval cycle is much lower in CGS than in FGS, thus incurring lower disk and RAM requirements for continuous data retrieval and transmission. (As an example, with the numerical assumptions stated in the next section, a server supporting 1000 MPEG-1 channels would require a stripe unit of 1.5 MB and a round duration of 1s in CGS; FGS would require a stripe unit of 22 MB and a round duration of 15s.) In addition to distributing the workload uniformly across disks, disk striping also enables multiple concurrent streams of a video to be supported without replicating the video. With a large number of disks (e.g., 1000), however, CGS has been documented in [4] to cause substantial admission latency if the system is highly loaded, unless movies are replicated in proportion to popularity. Nevertheless, this is an extreme case since disk arrays comprising more than 100 disks, or 2500 MPEG-1 streams in CGS, are not well-adapted to local VoD service.

4. CHOOSING A STORAGE ORGANIZATION

4.1. Cost and Storage Capacity

In our evaluation study, and for the remainder of this paper, disk parameters such as settle time, worst-case seek time, worst-case latency, rotational latency, cost and capacity are drawn from a commercially-available disk [22]. We also assume that all movie titles are compressed with MPEG-1 and have the same length (=100 min), as most movie titles are known to have playback lengths of about 90–100 min. While a Gaussian or a uniform distribution between, say, one and two hours might provide a better

[3] This can be enforced in both FGS and CGS by appending movie previews or advertisements. Such content insertion practice may subsidize VoD service providers in the same way as traditional video and pay-per-view services. As we shall see, however, the stripe unit and number of disks needed for a 1,000- or 2,000-channel CGS disk array are so small that typically less than a minute worth of content insertion can be expected.

[4] The more general issue of accessing a variable bit rate (VBR) encoded media stream using CGS is beyond the scope of this paper.

TABLE 1
Summary of Storage Parameters

Settle time	0.6 ms
Worst-case seek time	17 ms
Worst-case latency	25.5 ms
Worst-case rotational latency	8.34 ms
Storage capacity	2 GB
Inner track transfer rate	45 Mbps
MPEG-1 playback rate	1.5 Mbps
Disk cost	$1500/disk
RAM cost	$40/MB

match, it does not change the general conclusion of our analysis. With an inner track transfer rate of 45 Mbps and a capacity of 2 GB, each disk is able to support several MPEG-1 channels at 1.5 Mbps, whereas it can hold only one 100-minute movie (1.125 GB). This disparity is even greater with higher quality MPEG-2 movies: a 90-min movie would need about 5 GB of storage and an I/O bandwidth of 8 Mbps. In both FGS and CGS, analytical techniques elaborated on in [2] and experimentally validated are used to determine the *optimal retrieval cycle of a given number of concurrent channels at minimal cost*, and therefore optimal stripe unit based on (1) the latency for accessing data on disk, which is the sum of seek, rotational, and settle latencies;[5] (2) round-robin placement of consecutive stripe units; (3) C-SCAN disk scheduling; and (4) number of disks and amount of RAM needed for timely frame retrieval at playback rate R (and therefore, hiccup-free continuous display). Cost evaluation is based on (1) the amount of RAM needed to provide two-stage buffering at each channel in a dual buffer system; and (2) the number of disks needed to provide the required transfer rate. For illustrative purposes, we assumed disks and RAM costs in 1996, that is, $40/MB for RAM, and $1500 per individual disk of capacity 2 GB [2]. Numerical assumptions are summarized in Table 1.

Figures 2 and 3 plot the relationship between maximum movie selection and the channel capacity in number of MPEG-1 channels (henceforth simply referred to as channel capacity) supported by each scheme, for the parameters presented in Table 1, C-SCAN disk scheduling algorithm, round-robin data layout and optimal cycle length calculation (cf. [2]), introduced in Section 3.2. The number of disks was assumed to be the same in OMPD and CGS. As can be deduced from the piecewise constant curves of Fig. 3, the same number of disks can be used to support a certain range of channel capacity, depending on the amount of RAM available. In case of FGS, the penalty caused by accessing all disks in parallel for each block retrieval requires more disks, and hence more RAM, to support the same channel capacity as in CGS and OMPD. Consequently, even though a larger movie selection can be stored, the storage cost, which is, in the rest of this paper, *the sum of both disk and RAM costs*, increases almost exponentially with the channel capacity in FGS, as shown in Figs. 4 and 5. For the same channel capacity as in FGS, CGS and OMPD incur a much lower storage cost, which varies quasi-linearly with the channel capacity. Note that the storage cost is also slightly lower in OMPD than in CGS due to its lower RAM requirement. It is clear, however, that storing 1.78 times more movies in CGS is worth a marginal increase in cost with OMPD. Last, it is also interesting to note that only 44 disks are needed to support a 1,000-channel CGS disk array. For a stripe unit corresponding to one minute worth of video, this means that at most 22 s of content insertion are needed in order to ensure that the number of stripe units per video is a multiple of the number of disks, and therefore trivial, starvation-free request scheduling.

Ideally, the storage organization of a VoD server is cho-

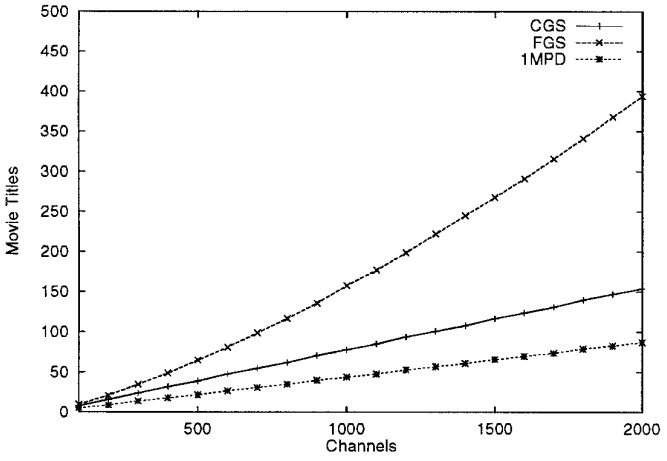

FIG. 2. Number of movie titles supported.

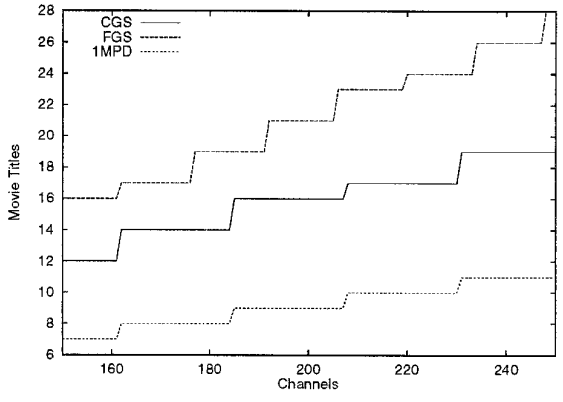

FIG. 3. Number of movie titles supported (close-up).

[5] Parameters usually provided by the manufacturer (cf. [22]).

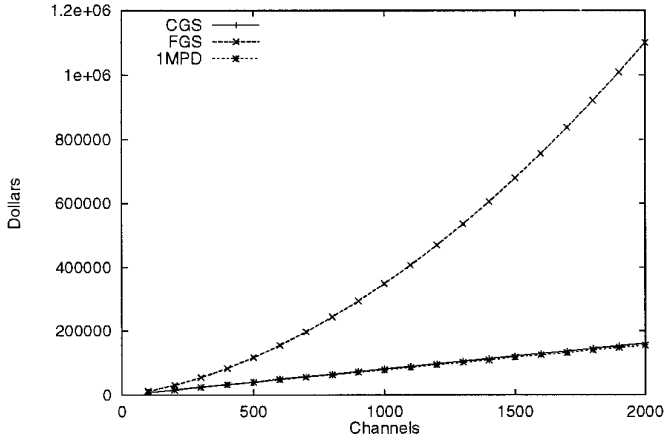

FIG. 4. Storage cost.

sen so a large collection of movie titles and concurrent channels may be made available with the highest QoS at the lowest possible cost. In reality, Figs. 2 and 3 indicate that a feasibility region must be determined to satisfy both constraints as closely as possible. This issue is made clearer in two example constraints, 1 and 2,[6] in the first graph of Fig. 6. In case of constraint 1, CGS yields extra storage space available for new video material or mirroring of existing movies. On the other hand, storing the required number of movies with FGS or OMPD is possible only if more channels than initially required are provided. As can be seen in the second graph of Fig. 6, cost ultimately determines the choice of a system. In fact, regardless of whether the system is disk-arm bound (areas **A** and **C**) or disk-space bound (areas **B** and **D**), CGS can be shown to be the most advantageous scheme since (1) the difference in cost between FGS and CGS increases almost twice faster

[6] For simplicity, the curves corresponding to each scheme are approximated, but this does not affect our conclusion.

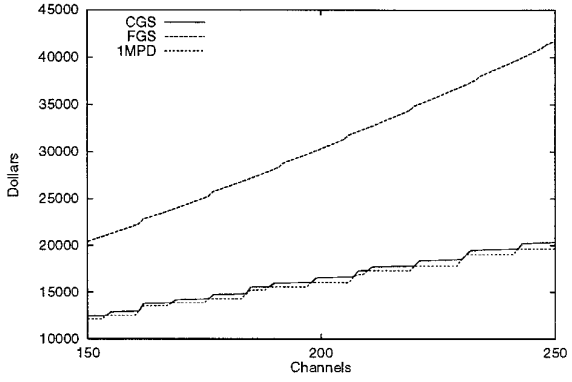

FIG. 5. Storage cost (close-up).

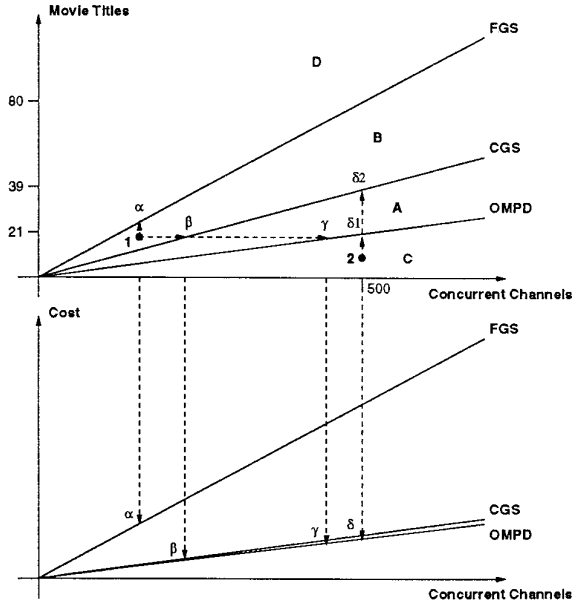

FIG. 6. Choice of a storage organization.

than the difference in number of movie titles supported; (2) at approximately identical costs and channel capacity, CGS can support more movie titles than OMPD. Similar observations can be made by considering storage of 90–100-min MPEG-2 movies on Seagate's newer 9-GB *Ultra* drives (rather than 2 GB), whose transfer rate is three times that of older 2-GB drives.

Note that constraints located in areas **B** and **D** (respectively, constraints of *type* **B** and **D**) are not economically rewarding since they correspond to vastly underutilized storage-limited systems which provide a large selection of movie titles to a small user population. This type of constraint is characteristic of movie archives, for which tertiary storage technology may be better suited. Inversely, it is economically more viable for companies to design front-end servers of type **A** or **C** and serve the most frequently requested movie titles to a subset of the total viewer population. In the extreme case 2 of area **C**, a large number of channels is to be allocated to a small number of movies. Although CGS can support more movies than required, OMPD is still acceptable in terms of cost. The choice of a scheme will then depend on customers' QoS, discussed next.

4.2. Customers' QoS

4.2.1. Admission Latency

The first component of customers' QoS we investigated is the waiting time (in receiving the requested service) averaged over all movie titles. In a similar study in [23, 24], the overall blocking probability was chosen instead

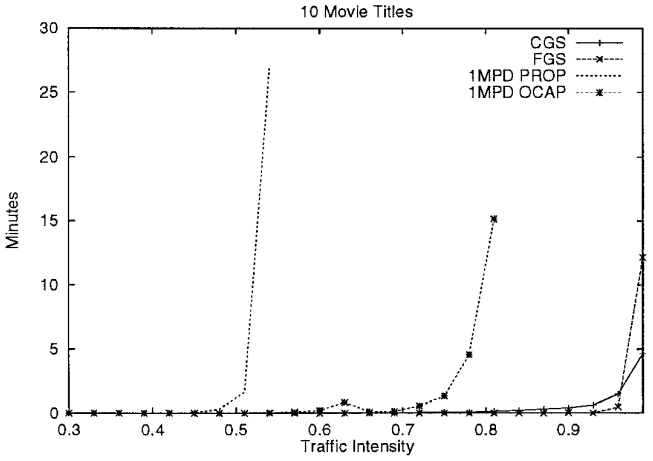

FIG. 7. Average latency before service.

of the average waiting time. However, this choice is not sufficient to evaluate customers' QoS in VoD, because the blocking probability averaged over several queueing systems does not indicate whether customers requesting some arbitrary movie will be blocked with a probability close to 1, hence suffering an unacceptable long wait. As in Section 3.1, we assume that the aggregated request arrival process is Poisson, and that waiting customers are served on a first-come-first-served basis without any form of admission control. We considered a capacity of 500 channels and 10 different movie titles, for each storage organization presented in the previous section. For these values, RAID-5 and OMPD require 21 disks. To evaluate the gain by employing the OCAP in OMPD, we considered two cases of disk partitioning: (1) the OCAP EQ-OPT, determined by the optimal heuristic in [14], (2) the CAP EW-PROP, obtained by allocating the number of channels per movie in proportion to popularities. We adopt Zipf's law [4, 9, 25] as the stationary model of movie popularity, which has proven to work even when the viewing probability of videos is not stationary [26].

Our simulation results, plotted in Fig. 7, show that the admission latency is usually low for the traffic intensities that can be supported. The differences between the various schemes appear to be mostly in the maximum sustainable traffic intensity, beyond which the load exceeds the storage bandwidth and users have to wait for a long time before receiving service. This is because those channels that have been assigned to customers remain allocated for the entire length of a movie. As expected, FGS and CGS perform comparably since CS organizations provide customers with very low latency. For very high traffic intensities, however, higher concurrency in CGS is preferable. Separate simulations confirmed that replication in CGS can only marginally affect customers' latency for a relatively small number of disks (typically, less than 100). In OMPD, the admission latency depends directly on the number of channels allocated to each movie title, that is, on the partitioning decisions and storage allocations made by the D-VoD server based on movie popularities. In this case, partitioning disks among movie titles according to EW-OPT performs better than EW-PROP.

The admission latency in FGS and CGS depends only on the channel capacity regardless of the number of stored movie titles. In OMPD, on the other hand, the admission latency decreases with the selection of movie titles. In the extreme case of only one movie title, the latency in OMPD and CGS is almost the same and hence, OMPD is better for availability. Storing d movie titles with OMPD, on the other hand, results in a very unstable system which cannot honor requests for most popular movies, unless more disks are added. In other cases such as 10 movie titles, the choice of a scheme depends on whether low customers' latency in CGS or availability in OMPD is valued more.

4.2.2. *Availability*

Service availability in a VoD server can be measured by the minimum support provided in case of reconfiguration. In a sense, availability also indicates the VoD server's versatility, i.e., the ability to be reconfigured with minimal effect on service availability. A reconfiguration typically occurs (1) after a disk failure, or (2) when video material is updated, or (3) when the system capacity is upgraded. In the event of disk failures, a disk array can be made as reliable as OMPD by adding redundant data on one or several extra parity disks. In this way, if one disk fails, sophisticated algorithms [17] can be used to reconstruct data from the remaining disks, as a background process, while maintaining the same service level, at the cost of extra buffer. At present, the *mean time to failure* (MTTF) of a single disk is on the order of 600,000 h (approximately 70 years) [17, 27]. Considering the fact that only 78 disks are needed to support 1,000 MPEG-1 channels using CGS, local front-end servers will typically contain less than 100 disks, and it is therefore realistic to assume an aggregate MTTF of 6,000 h (approximately 250 days).

This failure rate is negligible compared to the much higher frequency of the other two types of reconfiguration, which risk the availability of the entire array by requiring all data to be reassigned to disks. Since VoD service will compete with—and eventually replace—the existing video rental facilities, the most popular movies held by front-end VoD servers will be renewed on a weekly basis. Furthermore, insertion of advertisements or movie previews may subsidize VoD service providers in the same way as traditional video and pay-per-view services. Such insertion may occur on a daily basis. When data is striped across all disks, adding disks also requires all data to be reassigned to the disks. Finally, gradually extending a video server

may lead to a heterogeneous system, due, for instance, to disk models being discontinued [27]. In all these cases, as advocated in [28, 29], high availability is achievable only by using disk arrays with *narrow striping* across multiple independent striping groups, or *clusters*.

5. PARTITIONING A CS STORAGE ORGANIZATION

Monolithic servers lack availability during rebuild operations and are not easily scalable to a wide range of requirements. These limitations led researchers to experiment with partitioned (or clustered) disk arrays for their cost-effective combination of concurrency, storage efficiency, and low customer latency. Depending on the total number of striping groups, however, reconfigurations and additional scheduling complexity during reconstruction may degrade performance for an unpredictable period of time, even when sophisticated replication or duplexing algorithms are used [20, 29]. In this section, we determine the tradeoff among cost, customers' latency, and availability, incurred by partitioning a disk array. The quasi-linearity of the cost functions in Fig. 4 suggests that storage fragmentation in CGS may be done at near constant cost and capacity. However, partitioning a storage organization based on FGS, as recently proposed in [30], is not as economically appealing, due to the convexity of its cost function. So we will restrict our study to CGS and assume that storage is partitioned into equal-size clusters. To achieve the appearance of a single resource from multiple clusters, we also assume that a storage switch is inserted between storage and clients. The more general issue of unequal-size clusters is left as future work. However, versatility is difficult to achieve with heterogeneous configurations, because they complicate (1) assignment of videos to clusters, and (2) video migration among clusters, both of which are needed to maintain a balanced, efficient system as movie popularities change.

5.1. Eligible Configurations

Consider a 500-channel disk array with CGS. According to Fig. 6, it comprises 21 2-GB disks of the type considered in Section 4.1 and supports a maximum of 39 movie titles of 100 min each (cf. Fig. 6). Partitioning these 21 disks while keeping the cost as constant as possible leads to only two *eligible configurations*: three clusters of 7 disks each, or seven clusters of 3 disks each. Clearly, partitioning a disk array to keep the total cost constant is not practical as it yields too few eligible configurations to choose from, that is, too few alternative choices for an adequate combination of cost, capacity, and customers' QoS. If, on the other hand, we allow the storage cost to vary within a given range, corresponding, for instance, to the cost of 1 or 2 disks and corresponding RAM, more configurations may be considered and actually offer a better overall trade-off.

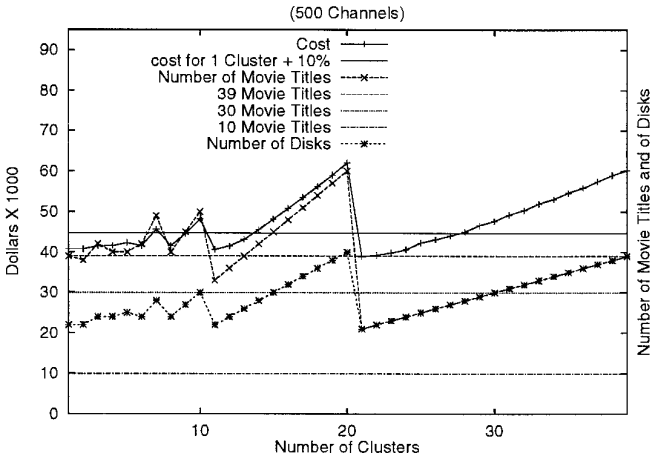

FIG. 8. Storage cost, number of movies titles supported, and number of disks for various number of clusters: constraints of type **A** and **C**.

This is the case, for instance, if a cost increase as low as 5% can be afforded when partitioning a 500-channel disk array.

Depending on the required channel capacity and movie selection, one may distinguish two cases. First, if these constraints define a point located in area **A** or **C** of Fig. 6 (respectively, constraints of *type* **A** or **C**), the storage capacity (in number of titles) is greater than the movie selection to be supported, hence leaving room for replicating the most frequently requested movie titles. Such constraints naturally favor partitioning because, as we shall see, a small degree of replication can significantly reduce loss of viewers due to reconfigurations, even when the system is heavily loaded. Moreover, movie titles can be replicated or added at will, without having to change the storage configuration. In Fig. 8, variations of the total cost (disks and RAM), number of movie titles, and number of disks, found analytically as presented in [2], are represented as a function of the number C of clusters, assuming a minimum channel capacity of 500. To make the graph more readable, various constraints on the number of movie titles are also indicated: (1) 39 movie titles, corresponding to the maximum number of movies stored on one CGS array that supports 500 channels; (2) 30 movie titles, corresponding to type **A** constraint (for the remainder of this paper, we shall refer to (500 channels, 30 movie titles) as constraint **A**); and (3) 10 movie titles, which are a constraint of type **C** (we shall refer to (500 channels, 10 movie titles) as constraint **C**). We also indicated the maximum cost allowed, arbitrarily set to the cost of a 1-cluster system *plus* 10% for illustration purposes. C was varied from 1 to 39.

As can be seen from Fig. 8, increasing C from 11 to 20 clusters results in a linear increase in cost, and correspondingly, in number of movie titles and disks, since for each

of these configurations, the only way to provide 500 channels with equal-size clusters is by allocating two disks per cluster. The same observation can be made in configurations with more than 21 clusters, which correspond to OMPD (each cluster comprising only 1 disk). In summary, whether clusters comprise one or two disks, disk bandwidth is increasingly underutilized as the number of cluster grows, since each cluster is used to support less channels; Beyond a certain number of clusters (or disks), the cost constraint will not be met.

In the case of type **C** constraint, the required movie selection (10 movie titles) is low enough to the supported by most configurations, even by those corresponding to OMPD. The set of eligible configurations is then fully determined by the cost constraint, and corresponds to configurations with $C \in \{1 \rightarrow 6, 8, 11 \rightarrow 13, 21 \rightarrow 27\}$. Constraint **A** (e.g., 30 channels) is more restrictive since configurations that are acceptable (in the sense of cost) do not necessarily provide enough storage space to store 30 titles. So, both cost and selection constraints have to be considered, making OMPD configurations ineligible. In the limited case of 39 movie titles, only very few configurations are eligible ($C = 1, 3, 4, 5, 6, 8, 13$). It is worth noting that, even in this restrictive case, the set of eligible configurations would not change significantly if the cost margin is set to 5%. Clearly, a wide range of configurations are possible even when storage design is restricted by a relatively tight cost constraint.

If constraints on channel capacity and movie selection are of type **B** or **D**, the required movie selection can only be supported by configurations whose bandwidth (in number of MPEG-1 channels) exceeds the channel capacity constraint. As mentioned earlier, such constraints are not typical of front-end VoD servers, and certainly not economically sensible as disk bandwidth may be severely underutilized. In this case, eligible configurations are found by partitioning the number of movie titles required and keeping the channel capacity of each cluster as low as possible. In summary, whenever a service provider decides to increase the capacity of a server, channel capacity should be given priority to viewers' selection so design constraints may be kept in areas **A** and **C** of Fig. 6. This guideline can be followed by adding RAM first, until full utilization of each cluster, then one disk at a time to each cluster, if necessary.

5.2. *The Video Allocation Problem*

Once an eligible configuration has been chosen, movie titles have to be replicated and assigned to different clusters so that a maximum number of sessions may be supported, and customers' admission latency may be minimized. It is shown in [1] that these goals are achieved in a balanced system by assigning movies to clusters so as to distribute requests uniformly to different clusters. The problem of balancing probabilities for homogeneous accesses across clusters is analogous to the ''bin-packing'' problem. Fortunately, in the case of equal-sized clusters, an optimally balanced system can be obtained as follows.

First, the total storage capacity measured in number of movie titles is compared to the constraint on movie selection. If space is available for replication (e.g., if 39 slots are available for 30 movie titles), slots are allocated to videos according to some arbitrary replication policy. Replication can be done in proportion to popularities, or by simple duplication of the most popular videos. (Note that the latter approach may allow more movie titles to be replicated.) Each video is then assigned a *partial popularity*, which is equal to its initial popularity divided by the number of its copies. Assuming C clusters and s_C slots per cluster, our objective is to find an assignment from the list of all $C \cdot s_C$ instances of all movie titles and their corresponding partial popularity to the C clusters so the highest access frequency among all clusters is minimized.

We show in Appendix A that optimum video allocation can be found in $O((C^2 s_C^2 + C^3) \log(C \cdot s_C))$ time, starting from any arbitrary assignment. Our assignment problem is reduced to a binary search, in which each iteration involves solving a *max-flow problem* in a directed network [20], thus resulting in a polynomially bounded complexity. The algorithm presented also works for uneven clusters, provided cluster capacities are known in advance. In practice, the search space can be greatly reduced by listing all instances of all movie titles in decreasing order of partial popularity, then ''folding'' the sorted list as follows. The first C entries in the list are assigned to the C clusters in ascending order. Then, the next C entries are assigned to the C clusters in descending order. The process is repeated until all entries have been assigned. Taking into account the additional time required for sorting the list of videos ($O(C \cdot s_C \log(C \cdot s_C))$ using Quicksort), the time complexity of optimum video allocation remains polynomially bounded.

Depending on the constraint on movie selection, slots may be available for replication and videos duplicated on the same cluster. In such a case, a single disk failure in a cluster can be tolerated by avoiding duplication of video blocks on the same disk. It should be noted, however, that by avoiding replication of the same video on the same cluster, it is possible to (1) provide reliable access in case of failure of an entire cluster, and (2) accommodate request surges with the completely shared channel capacity of two clusters. Moreover, duplicating a video in the same striping group does not significantly improve the latency experienced by requests for this particular video in relatively small disk arrays.

Since better reliability and scalability are achieved by avoiding replicating the same video on the same cluster,

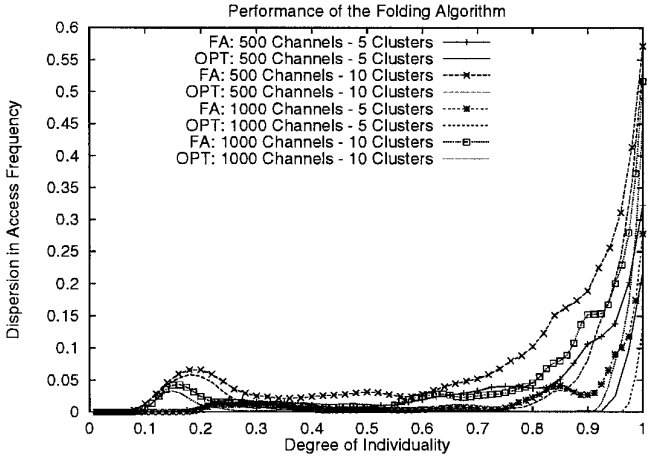

FIG. 9. Performance of various assignments of videos to clusters.

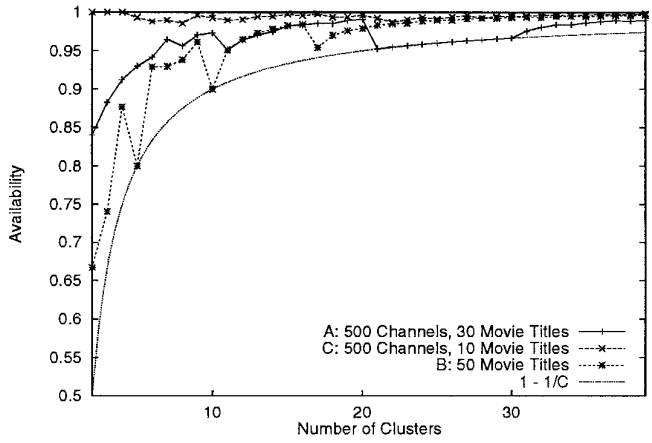

FIG. 11. Availability for constraints **A**, **B**, and **C**.

in our simulations, no video was duplicated on the same cluster. We first evaluated video allocation by considering the coefficient of variation (or *dispersion*) in the access frequency of each cluster, which measures access balancing, as a function of the degree of individually, or *selectivity*, of videos offered to customers. Selectivity 0 corresponds to a unique movie title, whereas selectivity 1 corresponds to a configuration in which no movie title is replicated. Figure 9 depicts the differences in performance between optimum and folding algorithms (FA) in four different configurations, depending on the channel capacity (500 or 1,000) and the number of cluster (5 or 10). Clearly, optimum assignment greatly improves over the commonly-used greedy allocation [28, 31] of FA. To complete this discussion, we plot in Fig. 10 the simulation results for four and eight clusters, in two assignment scenarios: (1) FA vs optimal video assignment, and (2) proportional replication vs duplication (DUP) with respect to the admission latency in the case of constraint **A**. Even a slight difference in the dispersion in access frequency between FA and optimal assignments is shown to result in a noticeable discrepancy in the overall admission latency, especially for systems with a large number of clusters.

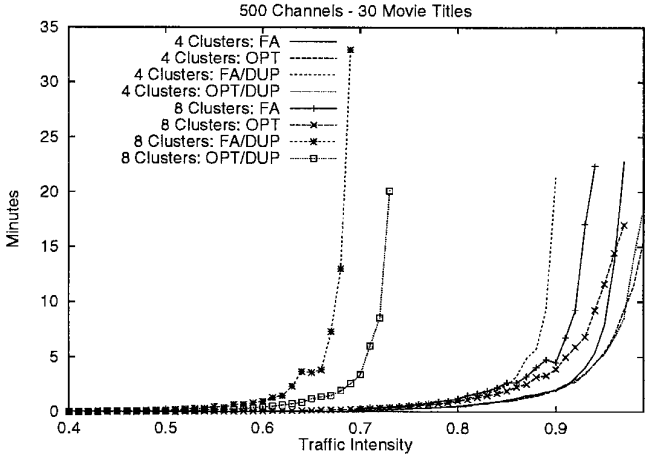

FIG. 10. Average latency before service: Constraint **A**.

5.3. Evaluation of Customers' QoS

5.3.1. *Availability*

Service availability is measured by the proportion of customers which remain unaffected by the failure of a single cluster. Assuming that viewers of a replicated video can be shifted to an alternate cluster, availability A_v depends only on the access frequency to nonreplicated videos:

$$A_v = 1 - \frac{1}{C} \sum_{\text{nonreplicated}} p_i.$$

A_v is plotted in Fig. 11 for constraints **A**, **B**, and **C**. A_v is lower-bounded by $1 - 1/C$, corresponding to the configuration with no replicated movie titles. In constraint **A**, skewed access and moderate replication of the most popular videos in a 2- or 3-cluster configuration is sufficient to ensure $A_v \geq 85\%$. Any additional partitioning beyond six clusters ($A_v = 95\%$) improves availability only marginally. In constraint **B**, variations of A_v are less clearcut since a higher eligible C does not always guarantee greater reliability. In most configurations, however, A_v is strictly greater than the lower bound $1 - 1/C$ as the optimum storage capacity is usually achieved at the cost of excess slots, available for replication. In the case of constraint **C,** a 2-cluster configuration is sufficient to provide near-maximum availability, since most or all of videos can be held within each cluster.

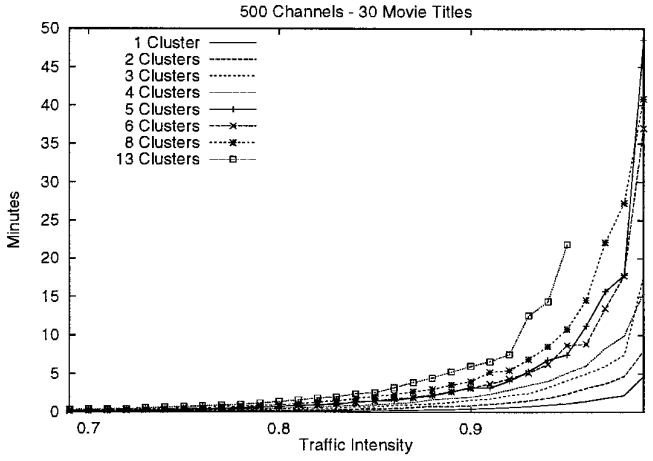

FIG. 12. Average latency before service: Constraint **A**.

5.3.2. *Admission Latency*

Availability is somewhat misleading since whenever a cluster is affected by a disk failure or reconfiguration, the requests received by the remaining clusters will increase by a factor of up to $1 - 1/C$. (This upper bound may be reached under some constraints of type **C**, depending on the number of clusters.) Replicated videos will thus be accessible with a higher latency, called the *latency during single-cluster reconfigurations* (LSCR), and hence, with a very few clusters, the VoD server may only benefit from such high availability for a restricted range of request arrival rate. To investigate this issue further, we plotted in Figs. 12 and 13, for the various number of clusters, the average admission latency and the LSCR for constraint **A**.

For traffic intensities below 0.8, customers' latency is almost independent of the number of clusters. As the traffic intensity increases beyond this point, the choice of a configuration depends on the desired maximum sustainable traffic intensity, and on the maximum latency and LSCR tolerable by customers. Let's assume, for instance, that the admission latency must remain below 5 min. The difference in the maximum traffic intensity guaranteeing such latency between 1-cluster and 13-cluster configurations is approximately 8%. However, a 13-cluster configuration provides near 100% availability and the lowest LSCR among all other configurations. While the 13-cluster configuration represents an extreme case, the choice of a configuration is not as restrictive as it may first appear. In general, as the number of clusters increases, the loss in maximum sustainable traffic intensity is acceptable in view of high availability. The choice of a configuration will therefore depend on which service conditions are expected by the service provider. If reconfigurations are not frequent, choosing a small number of clusters (e.g., 3) provides better scalability since the storage can handle sustained load surges, e.g., during "rush" hour. If, on the other hand, the service provider anticipates frequent reconfigurations (e.g., on a daily basis), several clusters are needed (e.g., 6, 7, or 8).

In practice, the reconfiguration and request arrival rates and customers' tolerance for long waits during peak hours may be difficult to estimate. Thus, the service provider may have to adapt the number of clusters whenever the system is determined to be not operating at the optimal level. The storage can be reconfigured to any eligible configuration if the largest number of disks and amount of RAM required by eligible configurations are allocated. In the case of constraint **A**, the nine-cluster configuration yields the largest number of disks, according to Fig. 8, whereas the configuration with the largest amount of RAM corresponds to one single disk array. Depending on the underlying configuration, excess RAM and disks may then be used to provide a slightly greater channel capacity or movie selection.

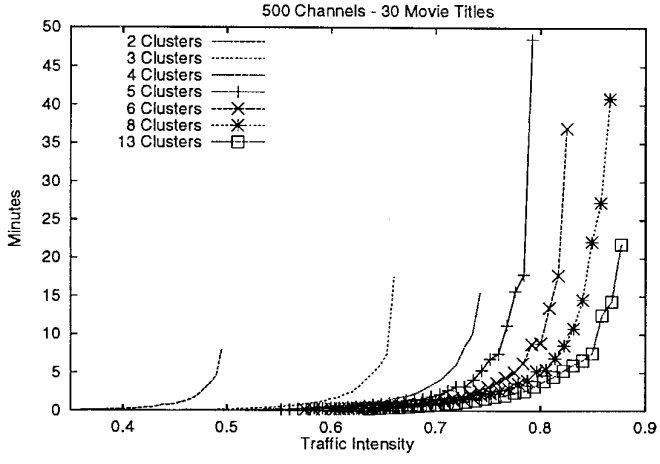

FIG. 13. Average latency during single-cluster reconfiguration: Constraint **A**.

6. REALISTIC TRAFFIC CONDITIONS

Thus far, we have discussed stationary request arrival rates. Consider more realistic time-dependent request arrival rates. For instance, one may expect to observe daily variations in the request rate between "prime time" (e.g., circa 8 p.m.) and "off-hours" (e.g., early morning). On a larger time scale (e.g., one week), changes in movie popularities due to new releases or customers' loss of interest in current titles over time may also cause changes in the request arrival pattern. Among all other possibly nonstationary factors such as popularity, number of movie titles, or customers' patience—which may not vary in a sinusoidal or even simply periodic fashion—it is safe to assume that time variations will be very moderate compared to that of arrival rate, which varies on a relatively

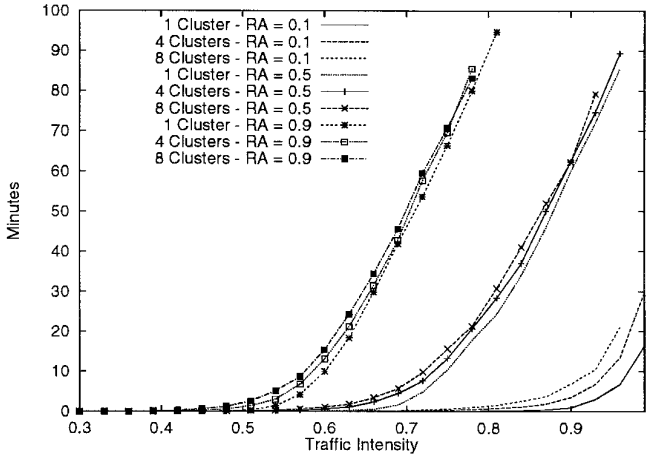

FIG. 14. Average latency before service for nonstationary arrivals.

short time scale. Consequently, our general findings in this section should be applicable to a wide range of practical situations.

6.1. Choice of a Configuration

The impact of time-dependent request rates for VoD service is usually studied through simulations, as there is, unfortunately, little reliable information available on the workload for VoD systems. As mentioned in [32], empirical data about realistic variations in arrival rates is hard to find due to the competitive nature of the industry. Storage system designers are thus forced to approximate the workload. For example, normally distributed requests for a particular movie were assumed in [7], where poor dimensioning is shown to leave a large number of customers unsatisfied during prime time. More general and probably more realistic distributions were considered in [33]. We assume for the remainder of this section that the requests aggregated over all movie titles are a time-dependent Poisson arrival process with rate given by

$$\lambda(t) = \overline{\lambda} + A \cos\left(\frac{2\pi t}{T}\right), \qquad (1)$$

where $\overline{\lambda}$ is the daily average arrival rate, $A(>0)$ the amplitude, and T a 24-h period. So the rate of requests for movie title m is $\lambda_m(t) = p_m \lambda(t)$. We also note that $\lambda_m = p_m \lambda$ and $A_m = p_m A$. The temporal variations of arrival rate are represented by the *relative amplitude* RA $= A/\overline{\lambda}$, which is the same for all movie titles (RA $= 0$ corresponds to the stationary case). As most of the results presented below can be generalized to a wide range of functions, a sinusoidal process is not as limiting as it may first appear. It is sufficient for illustrative and computational purposes and, above all, to capture the essence of cyclic customers' behavior.

In Fig. 14, we plotted the sensitivity of the average admission latency to changes in the request rate for constraint **A,** in the case of 1, 4, and 8 clusters, and for low, moderate, and high levels of nonstationarity (RA = 0.1, 0.5, and 0.9, respectively). For the same value of RA, the maximum sustainable traffic intensity decreases as the number of cluster increases, albeit marginally for RA $\geq$ 0.5. Our results show that an increase in nonstationarity tends to reduce the discrepancy initially observed for stationary request rates in Section 5.3. Important practical implications of this fortuitous result, also observed with other constraints, are that (1) the choice of a configuration is only marginally affected by nonstationarity in the request rate; (2) as nonstationarity increases, partitioning becomes more advantageous since it causes almost no loss in latency.

6.2. Motivation for OMPD

As mentioned in Sections 4.2 and 5.3, OMPD may be economically viable for reliably servicing a large user population with a small number of movies, provided customers' latency is not of paramount importance or the request load remains below the maximum sustainable request rate. Figure 15 represents the average latency for constraint **C1** with 500 channels and 10 movie titles. Constraint **C2**, depicted in Fig. 16, requires 500 channels and 21 movie titles. In both cases, OMPD corresponds to 21 clusters of one disk each. As illustrated in Fig. 16, choosing OMPD in **C2** results in very low sustainable traffic intensity since no replication is possible. In Fig. 15, on the other hand, the difference in maximum sustainable traffic intensity between OMPD and 1-cluster configurations is 0.4 for moderately stationary traffic (RA = 0.1) and 0.1 for RA = 0.9. Thus, although partitioning and duplexing a CGS disk array remains the most attractive configuration, OMPD can actually be strongly advocated with a nonstationary

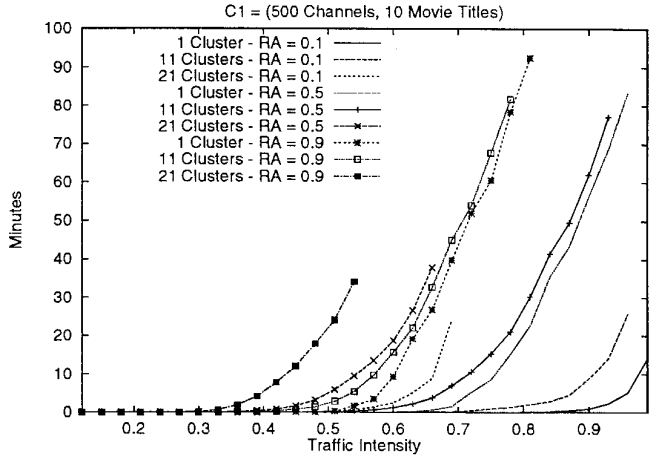

FIG. 15. Average latency before service for nonstationary arrivals.

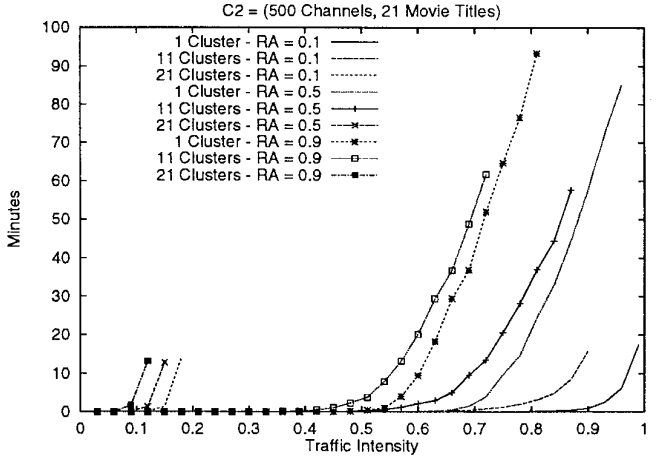

FIG. 16. Average latency before service for nonstationary arrivals.

request rate if, as for constraint **C1**, the number of movie titles is low enough for replication. In case of OMPD, we showed in [14] using approximation techniques for queueing systems with nonstationary request arrival rates that near-optimal performance can be achieved with replication of movies in proportion to movie popularities.

6.3. Related Work

It is impossible to have exact knowledge of customers' request rate. For completeness, we briefly comment on the role of adaptive techniques to alleviate the congestion caused by random short-term or long-term variations in customers' demand.

In the first case, the dynamic policy of segment replication (DPSR), as proposed in [28], may be used to balance the load across the clusters, and protect the VoD server from short-term request fluctuations. DPSR partitions video files into a small number of fixed-size segments, stored on various clusters, and dynamically replicated on specially-reserved storage space. It is shown in [28] that for constraints similar to **A**, DPSR can complement static allocation based on duplication and greedy placement of videos across clusters (a 21% improvement in service rate is documented). If, however, relatively few slots are available for replication, such a scheme will not be able to prevent quick saturation of the storage system due to request surges. If, as in constraint **C**, a large number of slots are available for replication, the improvement yielded by dynamic replication over static allocation of slots in proportion to popularities will be marginal. Also, slots available for replication are completely shared by all movie titles. If the request rate exhibits wide or unpredictable temporal variations, saturation can only be prevented through conservative connection admission control as no guarantee can be made on the latency or blocking probability when video segments need to be replicated. Last, replication decisions are based on deterministic prediction of future load, i.e., in a system without out-of-sequence access to videos.

Short-term adaptation should therefore be combined carefully with capacity planning, especially in the case of longer-term variations (e.g., daily, monthly, or seasonal). Determining whether the system's responsiveness to load surges will be limited by the complexity of long- and short-term adaptation algorithms is still an open issue. It is shown in [31] that predictive video replication, migration, and replication across the storage hierarchy of a distributed video server can efficiently complement adaptive scheduling of requests and dynamic load management. Such techniques are made easier with clustered configurations, since video objects can be reallocated to data sources without noticeable disruption to the existing service.

7. RESOURCE ALLOCATION IN TRUE-VoD

To be considered as a viable alternative to video rentals, D-VoD servers will have to eventually support VCR functionality and thus provide the so-called true-VoD (T-VoD) service. T-VoD servers will offer viewers the ultimate flexibility in (1) selecting a video program at any time, and (2) performing any VCR-like user interactions. Our objective in this section is to determine whether D-VoD systems are well suited to T-VoD. For this purpose, we shall identify specific conditions of customers' *interactive behavior* and storage organization, for which a D-VoD server can be upgraded to T-VoD at low cost and without significant loss in channel capacity.

After admission, customers' interactive behavior consists of the following types of interactions, originally identified in [6]:

Play/Resume. Regular video playout from the beginning or any other location.

Stop/Pause/Abort. Stopping the presentation, without picture or sound (Stop, Abort), or with picture but without sound (Pause). An abort action terminates the connection.

Fast Forward/Rewind. Immediate jump to a particular video location in the forward (backward) direction.

Fast Search/Reverse Search. Quickly moving presentation forward (backward), with picture and possibly sound.

Slow Motion. Moving presentation forward slowly, with picture and, possibly, sound.

A T-VoD server may also provide support for other interactions such as *reverse* and *slow reverse*, which correspond to playing a presentation in the reverse direction, at normal or slow speed [6]. In the rest of this paper,

however, we will not consider these interactions as part of the usual behavior of a customer viewing a video.

Linear access to CBR videos in D-VoD makes it possible to estimate the number of concurrent channels supported by a disk array, then to either accept or reject a storage organization based on cost, the number of movie titles, and channel capacity. In T-VoD provisions for VCR actions will alter data delivery rate, sequential access, and retrieval scheduling. Serving a customer in interaction or ''trick'' mode (e.g., during a slow motion, fast or reverse search) requires the VoD server to adopt a specific policy for video retrieval and delivery. Several schemes, outlined below, have been proposed to exploit the characteristics of video streams and, possibly, human perceptual tolerances, in order to support VCR actions with minimum overhead. Considering the general lack of field data in the area of interactive services, researchers have to make assumptions about customers' behavior and tolerance of QoS degradation. Service providers will therefore experiment with VCR functionality through successive upgrades of an initially specified VoD server configuration. In this section, we present approximation techniques to investigate the expandability of a disk-array-based D-VoD server to T-VoD service, assuming that data layout and disk scheduling remain unchanged; that is, with coarse-grained sequential striping of consecutive stripe units—of optimal size, as determined in [2]—in a round-robin fashion, and C-SCAN disk scheduling.

7.1. Probabilistic Connection Admission Control

As mentioned in [4], random seeks caused by VCR actions are easily schedulable in OMPD and FGS, since a (logical) disk is naturally a random-access medium. In CGS, on the other hand, if the disk array is operating near saturation, providing support for nonsequential access or retrieval spanning across several disks may complicate scheduling operations since available entries must be present in the service list of each target disk during a round at unpredictable times. Here we propose an approximation to the number of concurrent interactive channels supported by a CGS disk array. This approximation can be used during ''probabilistic'' connection admission control, to protect existing connections from hiccups with a certain confidence level. It is reasonable to assume that disk scheduling and data retrieval in a balanced system are marginally affected when a large number of customers perform pause, stop, fast-forward, rewind, and slow-motion actions, provided these actions are equally likely among customers and throughout the duration of a video. We will therefore focus on fast and reverse search actions, as these actions require the VoD server to adopt a specific retrieval and delivery policy.

Several techniques have been proposed to implement fast and reverse search at n times the normal playback rate. In general, depending on the additional resources required (e.g., file format, storage, network support) and on the flexibility in the choice of a browsing speed and in the discontinuity felt by the user, there are three basic types of schemes. First, the simplest scheme retrieves and transmits video frames at n times the normal delivery rate [34] and requires significant additional system resources. The second type skips frames, or a group of frames, if the MPEG standard is used for video storage [35]. In the latter case, the selective segment sampling in [35] allows fast and reverse search in CGS disk arrays transparently, or without interfering with storage throughput, and hence, with the existing connections. However, viewers do not have any flexibility in the speedup factor, and discontinuities are felt during fast scan because the system does not retrieve consecutive blocks. Last, based on the assumption that a lowered picture quality is tolerable during fast or reverse search, the third type stores multiple versions of the same movie material corresponding to different picture qualities and rates. This approach is attractive in view of the fact that typical fast-forward scans of VHS video display approximately every 16th frame. This type of scheme is made possible by scalable multiresolution compression algorithms (e.g., MPEG-2) [36], which allow media files to be stored as a base component and a number of enhancement components to support various spatial and temporal resolutions and varying image quality. Nonscalable video compression algorithms such as MPEG-1 can be modified to emulate scalability [37]. An alternative is to append to the original movie a recompressed copy under forward or backward search playback [38], but at the cost of a significant increase in storage, depending on the choice of a speedup rate.

While it is beyond the scope of this paper to determine which scheme is better suited for disk-array-based video servers, for illustrative purposes, we will simply assume that fast or reverse search actions require a retrieval rate of K_1R, where K_1 is the speedup factor. This assumption holds whenever fast and reverse search operations alter the data retrieval rate, for instance, when video frames retrieved at n times the normal delivery rate, or when separate recompressed copies are retrieved, because, regardless of the encoding scheme, each stripe unit corresponds to a video block of fixed duration.

Our approximation is based on the assumption that conservative connection admission control through resource overreservation, or *overbooking*, is needed to accommodate unpredictable fast scans. If the channel capacity of a disk array is calculated based on a display rate $R_c = R + R_0$ slightly greater that R, only a small percentage of connections will suffer from hiccups when the number of customers is large. Assume further that the probability of a customer requesting a fast scan is constant throughout

the duration of a video and consider an arbitrary disk. Denoting by C_d the number of channels of rate R_c supported by an individual disk, the maximum number of fast-scan channels with rate K_1R is given by $C_t = (R_c/K_1R)C_d$. If viewers served by a disk are homogeneous and independent, the number N of customers performing fast or reverse search reduces to a binomial random variable and the *overloading probability* due to fast or reverse search is given by

$$P(N > C_t) = 1 - \sum_{k=0}^{C_t} \binom{C_d}{k} p_{fs,rs}^k (1 - p_{fs,rs})^{C_d - k}, \quad (2)$$

where $p_{fs,rs}$ is the probability of a viewer being in fast or reverse search model. Clearly, using Eq. (2) for connection admission control requires (1) knowledge of customers' behavior to determine $p_{fs/rs}$ and (2) specification of an average channel rate R_c. We address these two issues in the next section by proposing a model of channel usage, which, albeit speculative, combines analytical tractability and potential realism.

7.2. Modeling VCR Actions

Various models of customers' interactive behavior have been proposed [6, 34]. While these models differ in the nature of the VCR actions under consideration, they usually represent customers' behavior by a succession of normal play and interactive states of exponentially or uniformly distributed durations; transitions between interactive states such as slow motion and reverse search are not part of the model. For the sake of realism, a model should capture two specific parameters for their potentially significant impact on the performance of the VoD server and on the channel rate: (1) the level of interactivity, or frequency of requests for VCR actions; (2) the duration of VCR actions.

For this purpose, we assume that from customers' activities one can specify the channel usage model in Fig. 17, a simplified version of the model introduced by us in [10]. In this model, a set of states corresponding to the different VCR actions are assigned durations and transition probabilities to neighboring states. We assume that fast and reverse search statistics are combined together, as each of these interactions usually involves the same mechanism. The same assumption holds for stop and pause actions, and for fast forward and rewind as well. A channel serves each state for an exponentially distributed period of time. Since we are interested in the rate of a channel during the service (i.e., once a customer has been allocated a channel), it is safe to assume that there is always a customer ready to take the place of a departing user. Users leave the system either after issuing an abort command or when the end of a movie is reached. We assume that both cases are accounted for in p_1, the probability of a transition from play mode to abort.

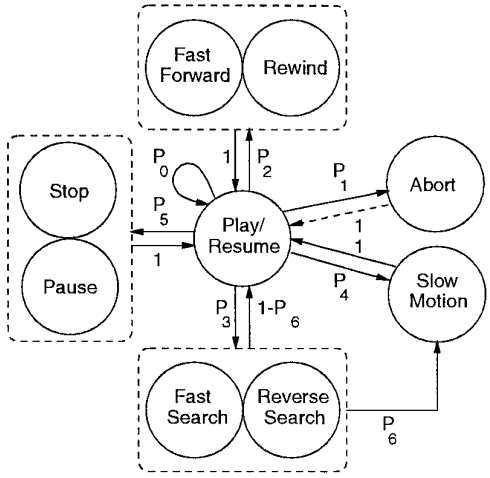

FIG. 17. Simplified transition diagram for channel usage.

The set of transitions depicted in Fig. 17 is chosen arbitrarily to show the key aspects of the problem, so that we can glean heuristics that work well in practice. Other sets of states and transitions are also possible, if they turn out to be closer to real behavior. Meanwhile, finding *representative* values for Fig. 17 is still an open issue since, to our best knowledge, there are no published realistic (or empirically verified as such) models of interactive customers. Until field data collection confirms their pertinence, our assumptions and their proper are therefore inevitable. Also, note that the above-mentioned parameters are captured by this representation of viewers' activity: the level of interactivity can be adjusted by assigning higher or lower transition probabilities from play/resume state to other states and the duration of VCR actions is included in the model. Last, transitions between interactive states such as slow motion and reverse search are modeled.

Our channel usage model is applied with the following durations: $d_{\text{Instant Playback}} = 1/\mu_1$, $d_{\text{Play/Resume}} = 1/\mu_2$,[7] $d_{\text{Fast/Reverse Search}} = 1/\mu_3$, $d_{\text{Slow Motion}} = 1/\mu_4$, and $d_{\text{Stop/Pause}} = 1/\mu_5$. Correspondingly, the transition probabilities are P_j, $j = 0, \cdots, 6$. These usage statistics can be collected in visualization laboratories or libraries with traditional video equipment by monitoring a particular viewing station for a sufficiently long period and keeping track of the time users spend on a particular operation, counting transitions from one state to another, and computing their

[7] Note that we refined the channel usage model by making a distinction between *initial playback*, which corresponds to customers starting a video, and play/resume actions in the middle of an ongoing presentation, as both types of event could be statistically different.

TABLE 2
Transition Probability from Play/Resume

Behavior	P_0	P_1	P_2	P_3	P_5
NVI	0.65	0.12	0.08	0.08	0.07
VI	0.4	0.14	0.16	0.16	0.14

TABLE 3
Increase in Channel Rate in T-VoD

Behavior	NVI − S	VI − S	NVI − L	VI − L
%	1.3	2.5	6	11

relative frequencies. Based on these measurements, Appendix B shows how to calculate the portion of time a particular channel services each interaction, denoted $P(\overline{n}_j)$, $j = 1, \cdots, 5$, given by

$$\frac{\dfrac{\nu_j}{\mu_j}}{\displaystyle\sum_{i=1}^{5} \frac{\nu_i}{\mu_i}},$$

where

$$\nu_2 = \frac{1 - P_1}{P_1}\,\nu_1,$$

$$\nu_3 = \frac{P_3}{P_1}\,\nu_1,$$

$$\nu_4 = \frac{P_4 + P_3P_6}{P_1}\,\nu_1,$$

$$\nu_5 = \frac{P_5}{P_1}\,\nu_1,$$

and last, ν_1 satisfies $\sum_{j=1}^{5} \nu_j = 1$. With the knowledge of $p_{\text{fs/rs}} = P(\overline{n}_3)$, Eq. (2) can now be used for connection admission control.

7.3. Evaluation

In our simulation of T-VoD, VCR actions are assumed to be of equal length and can be either 5 min long (L) or 1 min long (S), with the exception of the duration in play/resume state being fixed at 10 min and in abort state of null duration. As represented in Fig. 17, the transition probability from any state other than play/resume to play/resume is 1, with the exception that $1 - P_6$ is set to 0.5. We also assume that slow motion is requested only after a reverse search (i.e., $P_4 = 0$). For other transitions, customers can be either *very interactive* (VI) or *not very interactive* (NVI). Transition probabilities are summarized in Table 2. We can now specify four different types of interactive behavior depending on the choice of a level of interactivity (NVI or VI) and duration of VCR actions (S or L).

In Eq. (2), specification of an average channel rate R_c can be such that $P(N > C_t)$ remains below an arbitrarily specified level. Alternatively, we arbitrarily set R_c to the *theoretical channel rate*, which, assuming $\mu_1 = \mu_2$, can be expressed from $P(\overline{n}_j)$, $j = 1, \cdots, 5$ as (noting that stop and pause do not require data transmission):

$$\begin{aligned} R_c &= R(P(\overline{n}_1) + P(\overline{n}_2) + K_1P(\overline{n}_3) + K_2P(\overline{n}_4)) \\ &= R\left\{\frac{\dfrac{1}{\mu_2} + K_1\dfrac{P_3}{\mu_3} + K_2\dfrac{P_4 + P_3P_6}{\mu_4}}{\dfrac{1}{\mu_2} + \dfrac{P_3}{\mu_3} + \dfrac{P_4 + P_3P_6}{\mu_4} + \dfrac{P_5}{\mu_5}}\right\}, \end{aligned} \tag{3}$$

where $K_1 \geq 1$ is the speedup factor in fast/reverse search and $K_2 < 1$ the slowdown factor. For $K_1 = 3$ and $K_2 = 2$, the increase in channel rate from D-VoD to T-VoD is summarized in Table 3. Table 4 indicates the corresponding overloading probabilities, which are usually low and therefore acceptable.

7.3.1. *Comparison T-VoD vs D-VoD*

The first issue to address is how much T-VoD service costs compared to D-VoD. We repeated the approach taken in Sections 4 and 5 to determine the cost of providing 500 channels in D-VoD and T-VoD with the four types of interactive behaviors mentioned above. As in Section 5, we assumed that cost must be kept below the cost of the 1-cluster D-VoD configuration *plus* 10%. The results, presented in Fig. 18, show that most configurations that are eligible for D-VoD (i.e., below the cost margin) are also eligible for T-VoD, provided interactive behavior, or equivalently, channel rate, remains within a certain range (VI-L or R + 11% in our case). Correspondingly, dotted curves in Fig. 19 represent the loss in channel capacity when an eligible 500-channel D-VoD server is used for T-VoD. For each type of interactive behavior, curves without dots indicate the maximum number of channels that can be supported by adding RAM without changing the number of disks of the D-VoD server. The cost discrepancy

TABLE 4
Overloading Probability Due to Fast and Reverse Searches

Behavior	NVI − S	VI − S	NVI − L	VI − L
%	0.01	0.05	0.35	1.25

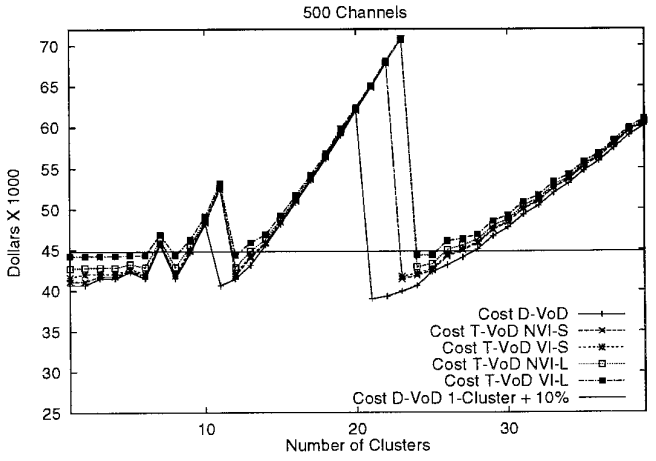

FIG. 18. T-VoD storage cost for various number of clusters.

appears to be mainly in the additional RAM needed to support higher rate T-VoD channels.

The dramatic cost discrepancy between D-VoD and T-VoD in the 11- and 21-cluster configurations in Fig. 18 shows that for small cluster sizes, allocating the minimum number of disks needed to provide 500 D-VoD channels results in a system that can only be upgraded to T-VoD by adding more disks, because each disk bandwidth is fully utilized. If, for instance, we consider the 21-cluster configuration, the only way to support 500 T-VoD channels with 21 clusters is by allocating two disks per cluster instead of one, thus resulting in a severe underutilization of disk transfer rate. By progressively adding more clusters (e.g., switching from 11 to 12 or from 21 to 22, 23, and 24 clusters), it is possible to accommodate the loss of 10% in channel capacity due to higher-rate T-VoD channels at an acceptable cost.

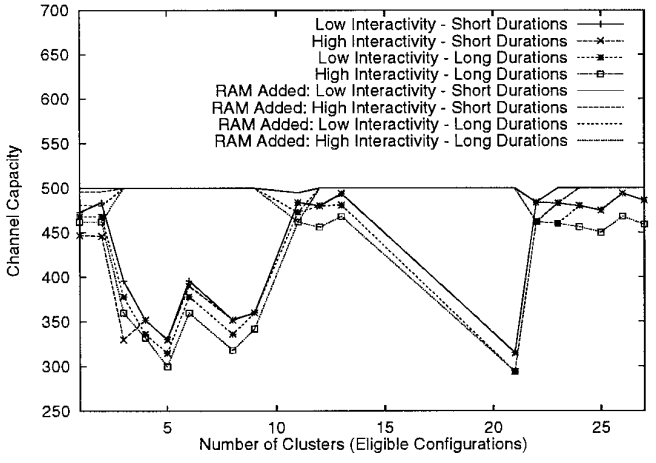

FIG. 19. Channel capacity of eligible D-VoD configurations used for T-VoD.

Smaller-size clusters can accommodate a number of channels almost proportional to the channel rate. For larger-size clusters with three disks or more (i.e., in configurations with less than 10 clusters), the relationship between channel rate and channel capacity is complicated by disk latencies during data retrieval. For this case, Fig. 19 shows that the loss in channel capacity incurred when a D-VoD server is used for T-VoD is much greater and can drop as low as 300–350 channels, even for a 1.3% increase in channel rate in the NVI-S case. Nevertheless, Fig. 18 shows that the cost of adding RAM, and occasionally disks, to support the same number of T-VoD channels is acceptable.

7.3.2. On Choosing a Storage Organization for T-VoD

Irrespective of the storage configuration used, our results indicate that it is advantageous to implement fast and reverse search in order not to increase the retrieval rate ($K_1 \leq 1$). This is achievable if fast access is performed by retrieving a version of the movie intended for that purpose [36, 38], or if fast and reverse search is achieved by selectively retrieving groups of pictures [35]. Both solutions implicitly assume that viewers can either tolerate loss in picture quality or discontinuities during fast and reverse search. In the general case ($K_1 > 1$), when a small movie selection is to be supported, choosing between a configuration based on disk arrays and the OMPD scheme depends on how precisely customers' behavior is known. If accurate knowledge of customers' behavior is available, a set of eligible T-VoD configurations can be specified.

If, on the other hand, service providers are experimenting with VCR functionality without prior knowledge of customers' behavior, the VoD server will undergo successive changes of a configuration initially specified for D-VoD service. In this case, reconfigurations in OMPD simply involve adding more disks, and customers' latency, albeit always high, remains relatively constant by switching storage configuration from D-VoD to T-VoD service.[8] In configurations based on disk arrays, switching from D-VoD to T-VoD service requires additional RAM, and sometimes disks, depending on the cluster utilization. One may then argue that the penalty incurred by restriping videos on larger clusters can be minimized by storing videos on a larger number of clusters. As illustrated in Fig. 20, however, even for a reasonably small number of clusters (e.g., 8), using a D-VoD configuration for T-VoD service results

[8] Temporal jumps or fast search throughout a video will change the duration of a connection, which can no longer be considered as deterministic, unless the number of connection is so large that the service time is approximated well to be constant. Nevertheless, our results indicate that even in local-area front-end servers (e.g., 500–1000 households), admission latency is primarily affected by the channel capacity and remains almost unaffected by changes in program length due to VCR actions.

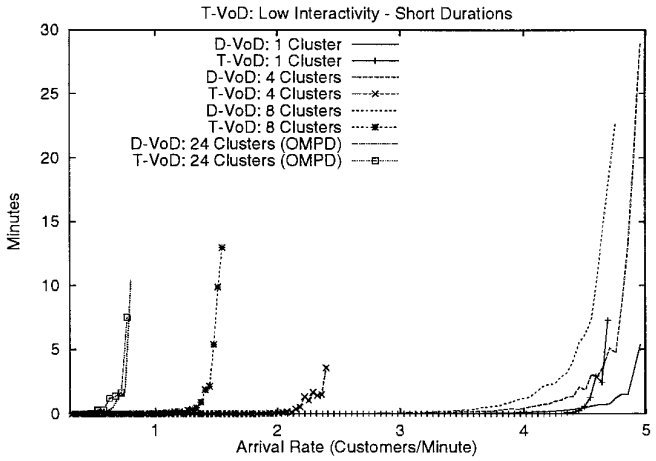

FIG. 20. Average latency in D-VoD and D-VoD used for T-VoD.

in a maximum sustainable request rate comparable to that of OMPD.

8. CONCLUSION

A VoD system must ultimately deliver interactive video services to a large number of customers at a competitive cost relative to broadcast services. This paper proposed how a distributed VoD server architecture can be adapted to become a large-scale service while focusing on long-term resource allocation in front-end servers. Given that VoD service providers target at the home consumer market, we opted for a practical approach and assumed that a given number of movie titles and concurrent channels have to be made available with the highest QoS at the lowest possible cost. We first linked customers' QoS to storage cost and capacity by focusing on two basic and well-understood types of storage technologies: OMPD and disk arrays. Both forms of storage constitute two extreme cases and are used to illustrate the range of choices available to local VoD servers. We showed that coarse-grained striping across disk arrays is a potential candidate for its cost-effective storage utilization and low admission latency. Considering that the entire capacity of a disk array is unavailable during reconfiguration, we next investigated hybrid storage organizations, in which videos are partitioned into several clusters. Assuming CGS, we specified a set of eligible clustered configurations and found that, with adequate video allocation across various clusters, a high level of reliability can be provided in a cost-effective way with minor effects on customers' latency.

We adapted the specification of eligible clustered configurations to realistic service scenarios, by taking into account temporal variations of the request arrival rate. We found that ensuring service scalability is not as critical as in the case of stationary arrival rates. We were also able to show that the OMPD scheme, often regarded as a naive approach to storage configuration in comparison with disk striping, may actually be quite acceptable for highly reliable service of a small number of movies to a large user population. Finally, we considered T-VoD service, which provides customers with full-fledged interactivity. We proposed a probabilistic connection admission control based on a model of channel usage in T-VoD. Our simulation results indicate that, for a reasonable range of customers' behavior, D-VoD can be upgraded to T-VoD for an increase in cost less than 5%. We also identified practical situations in which approximate knowledge of customers' behavior may justify choosing the OMPD scheme instead of a clustered organization based on disk arrays.

From the VoD server's standpoint, the choice of a clustered configuration in D-VoD or T-VoD, depends on which is valued most, service versatility or scalability. As mentioned in Section 2, however, it is possible to provide support for both in a nonconflicting, yet cost-effective way, by delaying the server's response to requests within customers' tolerance and batching customers' requests. If customers are willing to wait for a duration certainly less than that experienced in a saturated system, batching may potentially increase throughput by serving multiple requests for the same popular movie with a single channel. Furthermore, customers' average latency may also be reduced since requests for the same movie that arrived within the same batching period do not have to compete for resources. Choosing a scalable batching policy and providing full-fledged, yet scalable support for VCR actions in multicast VoD systems are two major research issues that warrant further investigation.

APPENDIX A. OPTIMAL ASSIGNMENT OF VIDEOS TO CLUSTERS

We adapt the algorithm initially presented in [20] for dynamic load balancing and optimal retrieval of randomly duplicated VBR video segments in a disk array. Let D be a set of C clusters and M a set of $C \cdot s_C$ instances of videos with partial popularity p_i' to be stored in a clustered disk array. The cluster assinged to video i is denoted by $C(i)$. Clusters' access frequencies are denoted by $F_j = \sum_{i=0}^{C \cdot s_C} p_i' \mathbf{1}_{C(i)=j}$, $j = 1, \cdots, C$. Our problem is to find an assignment $a : M \rightarrow D$ such that the highest access frequency among all clusters, $F_{\max} = \max_{j \in D} F_j$, is minimized. Let $j_{\max}$ denote the index of the corresponding cluster.

THEOREM. *Optimum assignment can be found in* $O((C^2 s_C^2 + C^3) \log(C \cdot s_C))$ *time.*

Consider an arbitrary assignment a. We first list all possible decrements in the access frequency of cluster $j_{\max}$. This list is constructed by replacing each video held by $j_{\max}$ with a different video, held by another cluster, and checking

whether such a change lowers the access frequency of cluster $j_{\max}$. For each possible decrement d_e, the question whether $F_{\max}$ can be lowered by d_e can be reformulated as a *max-flow problem* in a directed network (V, E) constructed as follows. For each cluster j we define a vertex v_j in V, labeled with F_j. For each ordered pair (v_{j1}, v_{j2}) in V, we construct the list of all possible decrements in access frequency. We then define a directed edge (v_{j1}, v_{j2}) in E, labeled with a capacity $c(v_{j1}, v_{j2})$, giving the maximum access frequency assigned to cluster $j1$ but able to be reassigned to cluster $j2$. The capacity of an edge thus gives an upper bound on the flow that may run from one vertex to another. This resulting network is now extended by two additional vertices, a source s and a drain d. For each v_j in V, an edge (s, v_j) is added with capacity $c(s, v_j) = \max(d_e + F_j - F_{\max}, 0)$, and an edge (v_j, d) is added with capacity $c(v_j, d) = \max(F_{\max} - d_e - F_j, 0)$. Now, it can be shown that $F_{\max}$ can be lowered by d_e if and only if a flow of size $\sum_{j=1}^{C} \max(d_e + F_j - F_{\max}, 0)$ can be realized from s to d, that is, if and only if all outgoing edges from source s have a flow equal to their capacity [20]. The maximum load $F_{\max}$ can be lowered by d_e units if and only if for each cluster j with $F_j > F_{\max} - d_e$, F_j can be lowered by at least $F_j - F_{\max} + d_e$ by appropriate reassignment of videos to other clusters, and for each cluster j with $F_j < F_{\max} - d_e$, F_j is increased by at most $F_{\max} - d_e - F_j$.

$F_{\max}$ being lower-bounded by $1/C$, a binary search strategy of complexity $O(\log(C \cdot m))$ can be initiated from any arbitrary assignment (e.g., the greedy heuristic). In each iteration, listing all possible decrements in access frequencies for each cluster takes $O(C^2 s_C^2)$ time, then solving a max-flow problem takes $O(C^3)$ time [39]. Hence, an optimal video allocation is found in $O((C^2 s_C^2 + C^3) \log(C \cdot s_C))$ time.

APPENDIX B. AVERAGE CHANNEL RATE IN T-VOD

In order to determine the probabilities that a customer is in a particular state at any given time, the activity model presented in Fig. 17 needs to be transformed into a Markov chain with five states and transition rate from state i to state j inferred from μ_i, $i = 0, \cdots, 5$ and P_j, $j = 0, \cdots, 6$. Alternatively, we choose to use the method in [11], based on queueing theory.

We model a channel serving one interacting customer as the closed Jackson network represented in Fig. 21. To each interaction i necessitating a different transmission rate or a different retrieval mechanism corresponds to an $M/M/\infty$ queueing system with average service time $1/\mu_i$. The exponential service time corresponds to the exponential duration of each interaction, asssumed in the channel usage model. Fast-forward, rewind, and abort actions do not necessitate any data transmission; fast-forward and rewind are simply performed quasi-instantaneously by retrieving out-of-sequence blocks of video from the storage. From a queueing perspective, these actions are modeled by transitions. If no statistical difference is found between initial playback and random play/resume actions, the Jackson network can be further simplified by merging systems 1 and 2 and assigning to the resulting system the input probability $P_0 + P_1 + P_2$.

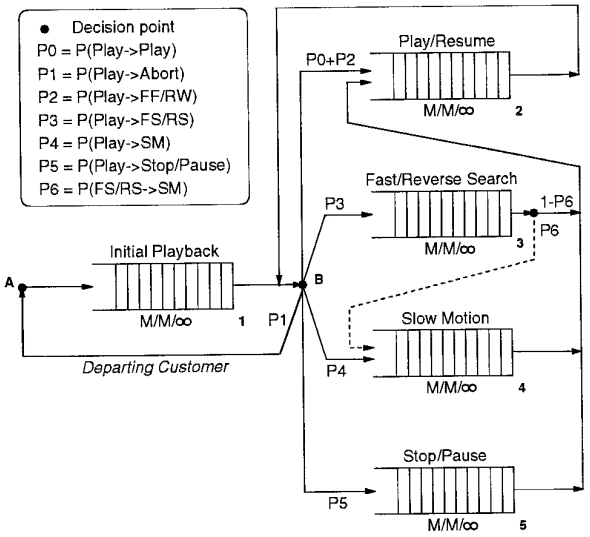

FIG. 21. Closed network model of a channel.

It is important to note that the closed network model is used as a tool to approximate the rate of a connection from the flow between different states. It should be based on a large number of observations of a large number of customers over a representative period of time. In reality, customers do not actually wait in line to get service for a particular interaction, nor are service times exponentially distributed or the number of waiting buffers infinite. Customers are assigned their own channel upon connection and keep it irrespectively of which state they are in.

Now calculate the portion of time spent by the customer in each queueing system of the closed network in Fig. 21. These portions are given by $P(\bar{n}_j)$, $j = 1, \cdots, 5$, probabilities of the customer being at system j, with

$$\bar{n}_j = [n_{j1}, \cdots, n_{j5}] \in \mathcal{N} = \{(1,0,0,0,0), (0,1,0,0,0), (0,0,1,0,0), (0,0,0,1,0), (0,0,0,0,1)\}.$$

We denote the rate at each queueing system j by $\alpha_j = \alpha(1)\nu_j$, where $\alpha(1)$ is the sum of the arrival rates to all five systems when there is one customer in the closed network, and ν_j the relative portion of the arrival rate to system j. We have

$$P(\bar{n}_j) = \chi(1,5) \prod_{i=1}^{5} p_i(n_{ji}), \tag{4}$$

where $\chi(1, 5)$ is a constant which depends on the number of systems and the number of customers, and the $p_i(n_{ji})$ are the standard probability of i customers in the $M/M/\infty$ system j, given in [11] as

$$p_i(n_{ji}) = \rho_j^{n_{ji}} \frac{e^{-\rho_j}}{n_{ji}!}, \tag{5}$$

where $\rho_j = \alpha_j/\mu_j$. The normalization conditions $\sum_{\overline{n}_j \in \mathcal{N}} P(\overline{n}_j) = 1$ and $n_{ji} = 1$ for $i = j$, 0 otherwise, leads to

$$\chi(1, 5) = \frac{\exp\left(\sum_{j=1}^{5} \rho_j\right)}{\sum_{j=1}^{5} \rho_j}. \tag{6}$$

We obtain the portion of time spent by the customer in each interaction system:

$$P(\overline{n}_j) = \frac{\dfrac{\nu_j}{\mu_j}}{\sum\limits_{i=1}^{5} \dfrac{\nu_i}{\mu_i}}, \tag{7}$$

By flow conservation, $\overline{\nu} = [\nu_1, \nu_2, \nu_3, \nu_4, \nu_5]$ is solution of $\overline{\nu}P = \overline{\nu}$, where P is the one-step transition matrix given by

$$P = \begin{bmatrix} P_1 & P_0 + P_2 & P_3 & P_4 & P_5 \\ P_1 & P_0 + P_2 & P_3 & P_4 & P_5 \\ 0 & 1 - P_6 & 0 & P_6 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

After calculations, we obtain

$$\nu_2 = \frac{1 - P_1}{P_1} \nu_1$$

$$\nu_3 = \frac{P_3}{P_1} \nu_1$$

$$\nu_4 = \frac{P_4 + P_3 P_6}{P_1} \nu_1$$

$$\nu_5 = \frac{P_5}{P_1} \nu_1$$

where ν_1 can be found by $\sum_{j=1}^{5} \nu_j = 1$. Finally, replacing $\nu_j, j = 1, \cdots, 5$ by their values given in Eq. (7) results in Eq. (3).

REFERENCES

1. T. D. C. Little and D. Venkatesh, Popularity-based assignment of movies to storage devices in a video-on-demand system, in *Multimedia Systems* **2,** 6, 1995, 280–287.
2. B. Özden, R. Rastogi, and A. Silberschatz, Disk striping in video server environment, in *Proc. IEEE International Conference on Multimedia Computing and Systems, Hiroshima, Japan, 1996*, pp. 580–589.
3. S. A. Barnett and G. J. Anido, A cost comparison of distributed and centralized approaches to video-on-demand, *IEEE J. Selected Areas Commun.* **14,** 6, 1996, 1173–1183.
4. A. L. Chervenak, D. A. Patterson, and R. H. Katz, Choosing the best storage system for video services, in *ACM Multimedia '95*, 1995, pp. 109–119.
5. A. D. Gelman and S. Halfin, Analysis of resource sharing information providing services, *Proc. IEEE GLOBECOM'90, December 1990*, pp. 312–316.
6. V. O. K. Li, W. Liao, X. Qiu, and E. W. M. Wong, Performance model of interactive video-on-demand systems, *IEEE J. Selected Areas Commun.* **14,** 6, 1996, 1099–1109.
7. J.-P. Nussbaumer, B. V. Patel, and F. Schaffa, Multimedia delivery on demand: Capacity analysis and implications, in *Proc. 19th Conference on Local Computer Networks, October 1994*, pp. 380–386.
8. B. Furht, D. Kalra, and A. A. Rodriguez, Interactive television systems, *Multimedia Tools and Applications*, pp. 235–277, Kluwer Academic Publications, Dordrecht/Norwell, MA, 1997.
9. C. C. Aggarwal, J. L. Wolf, and P. S. Yu, On optimal batching policies for video-on-demand storage servers, in *Proc. ACM Multimedia'96*, pp. 253–258, 1996.
10. E. L. Abram-Profeta and K. G. Shin, Providing unrestricted VCR functions in multicast video-on-demand servers, in *Proc. IEEE International Conference on Multimedia Computing and Systems, Austin, TX, June–July 1998*, pp. 66–75.
11. B. D. Bunday, *An Introduction to Queueing Theory*, Arnold, Sevenoaks, 1996.
12. D. Bertsekas and R. Gallagher, *Data Networks*, Prentice–Hall, Englewood Cliffs, New Jersey, 1987.
13. L. Kleinrock, *Queueing Systems, Vol. 1, Theory*, Wiley, New York, 1975.
14. E. L. Abram-Profeta, *Support for Service Scalability in Video-on-Demand End-Systems*, Ph.D. Thesis, University of Michigan, May 1998.
15. T. Kimura, Refining Cosmetatos' approximation for the mean waiting time in the $M/D/s$ queue, *J. Oper. Res. Soc.* **42,** 7, 1991, 595–603.
16. E. W. Biersack and C. Bernhardt, A fault tolerant video server using combined Raid 5 and mirroring, in *Proc. MMNC'97*, SPIE Vol. 3020, pp. 106–117, 1997.
17. B. Özden, R. Rastogi, and A. Silberschatz, Fault-tolerant architectures for continuous media servers, in *Proceedings of SIGMOD Conference, 1996*, pp. 79–90.
18. M.-S. Chen, D. D. Kandlur, and P. S. Yu, Optimization of the grouped sweeping scheduling (GSS) with heterogenous multimedia streams, *ACM Multimedia'93, Anaheim, CA, 1995*, pp. 235–242.
19. A. L. N. Reddy and J. C. Wyllie, I/O issues in a multimedia I/O system, *IEEE Comput.* **27**(3), 1994, 69–74.
20. J. Korst, Random duplicated assignment: An alternative to striping in video servers, in *Proc. ACM Multimedia'97*, pp. 219–226, 1997.
21. E. Chang and A. Zakhor, Cost analyses for VBR video servers, in *Proc. MMNC'96,* SPIE Vol. 2667, pp. 381–397, 1996.
22. *Seagate Product Overview*, October 1993.
23. M. G. Kienzle, A. Dan, D. Sitaram, and W. Tetzlaff, Effect of video server topology on contingency capacity requirements, in *Proc. MMNCC'96,* SPIE Vol. 2667, pp. 320–327, 1996.

24. G. Meempat and M. K. Sundareshan, Optimal channel allocation policies for access control of circuit-switched traffic in ISDN environments, *IEEE Trans. Commun.* **41,** 2, February 1993, 338–350.
25. G. Zipf, *Human Behavior and the Principle of Least Effort*, Addison–Wesley, Reading, MA, 1949.
26. G. Bianchi and R. Melen, Nonstationary request distribution in video on demand networks, in *Proc. IEEE INFOCOM'97*, April 1997.
27. R. Zimmermann and S. Ghandeharizadeh, Continuous display using heterogeneous disk-subsystems, in *Proc. ACM Multimedia'97*, pp. 227–238, 1997.
28. A. Dan, M. Kienzle, and D. Sitaram, A dynamic policy of segment replication for load-balancing in video-on-demand servers, *Multimedia Systems* **3,** 3, 1995, 93–103.
29. R. Flynn and W. Tezlaff, Disk striping and block replication algorithms for video file servers, in *Proc. IEEE International Conference on Multimedia Computing and Systems, Hiroshima, Japan, 1996*, pp. 590–597.
30. Y. Wang, J. C. L. Liu, D. H. C. Du, and J. Hsieh, Video file allocation over disk arrays for video-on-demand, in *Proc. ACM Multimedia'96*, pp. 160–163, 1996.
31. N. Venkatasubramanian and S. Ramanathan, Load management in distributed video servers, in *International Conference on Distributed Computing Systems* (*ICDCS'97*), pp. 528–535, 1997.
32. J. M. McManus and K. W. Ross, Video-on-demand over ATM: Constant-rate transmission and transport, *IEEE J. Selected Areas Commun.* **14,** 6, 1996, 1087–1098.
33. K. C. Almeroth, A. Dan, D. Sitaram, and W. H. Tetzlaff, Long term resource allocation in video delivery systems, in *Proc. IEEE INFOCOM'97*, 1997.
34. J. K. Dey-Sircar, J. D. Salehi, J. F. Kurose, and D. Towsley, Providing VCR capabilities in large-scale video servers, in *Proc. ACM Multimedia'94*, pp. 25–32, 1994.
35. M.-S. Chen, D. D. Kandlur, and P. Yu, Support for fully interactive playout in a disk-array-based video server, in *Proc. ACM Multimedia'94*, pp. 391–398, 1994.
36. K. Keeton and R. H. Katz, Evaluating video layout strategies for a high-performance storage server, *Multimedia Systems* **3,** 2, 1995, 43–52.
37. P. J. Shenoy and H. M. Vin, Efficient support for scan operations in video servers, in *Proc. ACM Multimedia'95*, pp. 131–140, 1995.
38. K. W. Ng and K. H. Yeung, Analysis on disk scheduling for special user functions, in *Proc. ACM Multimedia'96*, pp. 608–611, 1996.
39. J. Van Leeuwen, Graph algorithms, in *Handbook of Theoretical Computer Science*, Vol. A, Algorithms and Complexity, pp. 525–631, Elsevier, Amsterdam/MIT Press, Cambridge, MA, 1990.

EMMANUEL L. ABRAM-PROFETA received the B.S. in electrical engineering from the École Nationale Supérieure de Techniques Avancées, Paris, France in 1991, and both the M.S. and Ph.D. in computer science and engineering from the University of Michigan, Ann Arbor, Michigan in 1992 and 1998, respectively. He is currently a Software Engineer for World Opponent Network, Bellevue, Washington. His current focus is on the design and development of scalable gaming technologies on the Internet. His research interests include video-on-demand systems, multimedia servers, and networked interactive applications with emphasis on scalability and quality-of-service (QoS).

KANG G. SHIN is Professor and Director of the Real-Time Computing Laboratory, Department of Electrical Engineering and Computer Science, the University of Michigan, Ann Arbor, Michigan. He has authored/coauthored about 600 technical papers and numerous book chapters in the areas of distributed real-time computing and control, computer networking, fault-tolerant computing, and intelligent manufacturing. He has co-authored (jointly with C. M. Krishna) the textbook ''Real-Time Systems,'' McGraw–Hill, 1997. In 1987, he received the Outstanding *IEEE Transactions on Automatic Control* Paper Award, and in 1989, the Research Excellence Award from the University of Michigan. In 1985, he founded the Real-Time Computing Laboratory, where he and his colleagues are investigating various issues related to real-time and fault-tolerant computing. His current research focuses on quality of service (QoS) sensitive computing and networking with emphases on timeliness and dependability. He has also been applying the basic research results to telecommunication and multimedia systems, intelligent transportation systems, embedded systems, and manufacturing applications. He received the B.S. in electronics engineering from Seoul National University, Seoul, Korea in 1970, and both the M.S. and Ph.D in electrical engineering from Cornell University, Ithaca, New York in 1976 and 1978, respectively. From 1978 to 1982 he was on the faculty of Rensselaer Polytechnic Institute, Troy, New York. He has held visiting positions at the U.S. Airforce Flight Dynamics Laboratory, AT&T Bell Laboratories, the Computer Science Division within the Department of Electrical Engineering and Computer Science at U.C. Berkeley and the International Computer Science Institute, Berkeley, the IBM T. J. Watson Research Center, and the Software Engineering Institute at Carnegie Mellon University. He also chaired the Computer Science and Engineering Division, EECS Department, the University of Michigan for three years beginning January 1991. He is an IEEE fellow; was the Program Chair of the 1986 IEEE Real-Time Systems Symposium (RTSS), the General Chairman of the 1987 RTSS, the Guest Editor of the 1987 August special issue of *IEEE Transactions on Computers* on Real-Time Systems, and a Program Co-Chair for the 1992 International Conference on Parallel Processing, and served numerous technical program committees. He also chaired the IEEE Technical Committee on Real-Time Systems during 1991–1993, was a Distinguished Visitor of the Computer Society of the IEEE, an Editor of *IEEE Transactions on Parallel and Distributed Computing*, and an Area Editor of the *International Journal of Time-Critical Computing Systems*.