Designing Hierarchical Storage Systems for Interactive On-Demand Video Services

S.-H. Gary Chan Fouad A. Tobagi
Department of Electrical Engineering
Stanford University
Stanford, CA 94305

Abstract— **A hierarchical storage system achieves scalable storage in providing on-demand video services. In such a system, video files are stored in a tertiary level (such as a library or jukebox), and transferred to a secondary level (such as magnetic disks) from which it is streamed to the users. In this paper, we study the design of such a system for interactive applications in which a movie has to be completely transferred (or "staged") to the secondary level before it is displayed to its user; thereby incurring some user delay. We specify the required storage and bandwidth to meet a delay goal, given application characteristics. We have developed an accurate and yet simple model for the system, which greatly facilitates the design process and allows us to extract important performance characteristics of the system. We provide some design examples, and study the influence of video popularity.**

Keywords— **Hierarchical storage systems, interactive video services, tertiary storage, secondary storage, storage system model**

I. Introduction

On-demand video services, or video-on-demand (VOD), refers to services in which a user is able to request from a server any video content at any time. VOD encompasses many important applications pertaining to entertainment, information, education and advertising, such as movie-on-demand, news-on-demand, distance learning, home shopping, etc.[1], [2], [3], [4], [5] In VOD, once a file is displayed to a user, the user may to a certain extend interact with the videos using fast-forward, rewind, pause, etc. However, VOD applications are usually read-only, i.e., users are not allowed to modify the contents of the files.

A video system has to offer scalable storage in order to provide video services accommodating large number of video titles. Our goal in this paper is to address this scalability issue by designing a video system offering high service quality (in terms of, for example, user delay experienced) with low cost and complexity.

Most video servers reported on in the literature or existing today use magnetic disks to store and stream videos [6], [7], [8], [9], [10], [11]. Magnetic disk technology is used due to its high throughput, low access latency, random data access, and adequate unit storage capacity. However, magnetic disks are still not ideal to store large volume of video files (e.g., $>$ 1000s video files for some applications). Though disk arrays promise high storage capacity, there are still reliability and scalability issues to be resolved before they can scale easily up to terabytes of data. Furthermore, magnetic disks nowadays still suffer comparatively high cost — currently, storing a single 90-minutes movie of MPEG-I quality ($\simeq$ 1 GB) with such a medium can cost more than a hundred dollars.

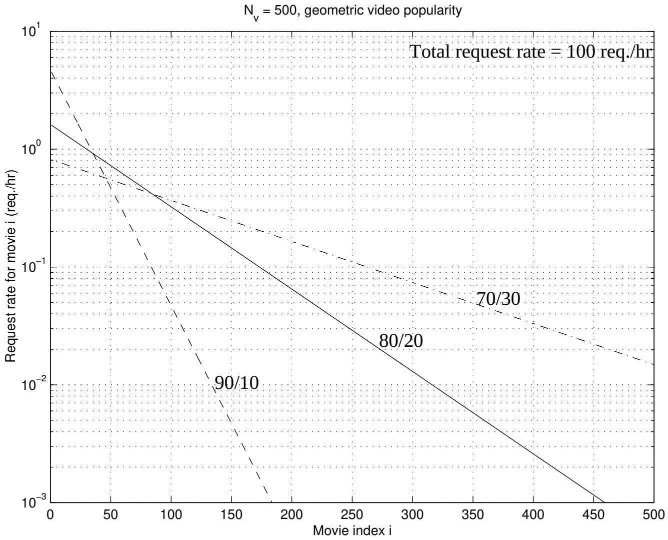

Fig. 1. Request rate for movie i in a video system.

In a video system, the popularity of the files can also differ markedly. As some videos may be rarely accessed, storing all files on-line regardless of their popularities is not very cost-effective [12], [13]. For example, consider a geometric video popularity model (i.e., the probability of accessing a movie in the system follows a geometric distribution) [13], [14]. We show in Fig. 1 the request rate of movie i in a system of 500 movies with a total request rate of 100 req./hr, given a certain skewness in the popularity indicated by r/m (the notation r/m represents that $r\%$ of the requests ask for $m\%$ of the movies). We see that the request rate of the movies can differ by many orders of magnitudes (such request rate may also change with time). The more skewed the video popularity is, the wider the discrepancy in the movie request rate.

A server based on hierarchical storage system provides cost-effective video storage. It consists of a tertiary and a secondary level. Tertiary storage, commonly referred to as library or "jukebox," is used to store all video files, which are dynamically transferred or "staged" into the secondary level to be displayed. The secondary level in the hierarchi-

This work was supported in part by a grant from the Center for Telecommunications at Stanford University.

S.-H. Gary Chan was with Stanford University when this work was done. He is currently with the Department of Computer Science, University of California, Davis, CA 95616.

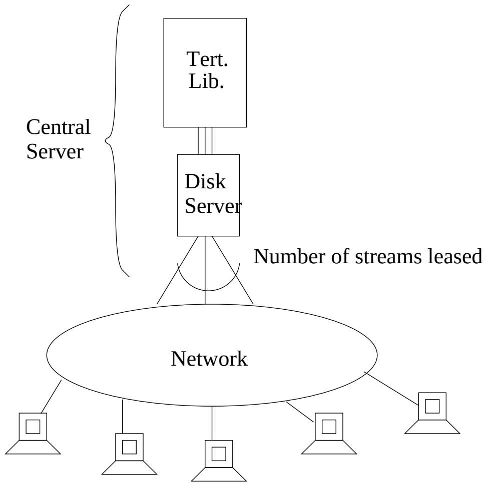

Fig. 2. A hierarchical storage system.

cal storage system therefore acts as a "caching" or "staging" platform for video display. As the media cost of tertiary storage can be orders of magnitude lower than that of secondary storage, the storage cost of videos is much lower. Furthermore, as a tertiary library can shelve hundreds or even thousands of removable tapes or disks, with each tape or disk offering high storage capacity ($\sim$10 – 100 GB), such a system achieves scalable storage capacity in excess of terabytes.

In a video system, the users may be served by many service providers, each of whom sets up its own server and provides services independently from the others. We show in Fig. 2 a hierarchical storage system serving a pool of users who make their requests over a network. The disk server streams the cached videos over a local or long-distance network (e.g., by using cable or satellite channels). Network channels may be acquired on a demand basis, or may be leased by the service provider. In the latter case, it is of interest to know the number of network streams required to achieve a certain performance objective.

In this paper, we consider the design of a hierarchical storage system for interactive video services (i.e., once a video is displayed, the user can interact with the video from its beginning to its end). This means that movies have to be completely transferred to the secondary level before they can be displayed. Since users are delayed (due to file transfer), the storage system has to meet a certain performance goal in user delay, given application characteristics (such as file sizes and popularity). We address the design using simulation and analysis. With simulation, we incorporate different levels of system details and complexity to study their impacts on performance. Analysis complements simulation by capturing explicitly parameter dependency, and identifying system bottlenecks and performance trends.

This paper is organized as follows. We first briefly describe previous work in Sect. II. In Sect. III, we pose our design problem and in Sect. IV, we specify the bandwidth and storage required in order to meet a certain performance goal, and the performance characteristics of the system. In Sect. V we provide design examples while in Sect. VI we address the effect of non-uniform video popularity. We conclude in Sect. VII.

II. Previous Work

Optimization of automated tertiary systems in data storage and delivery has been extensively studied previously. Nemoto et al. in [15] consider a system of multiple libraries and propose using a migration wagon to move tapes from one library to another in order to achieve load-balancing. Golubchik et al., Drapeau and Katz, and Ford et al. apply the idea of RAID to library systems to improve its performance [16], [17], [18]. Lau and Lui in [19] investigate the influence of the tape block size in a library on user delay for continuous media.

Tertiary level was discussed in the context of a hierarchical system as early as 1978 [20]. Misra in [21] analyzes IBM 3850 hierarchical storage system using a closed queuing network (a machine-repairman model). Pentakalos et al. in [22] models the Unitree mass storage system. All this work deals with data applications, while our work deals with continuous media with much larger file sizes.

Kienzle et al. in [23] study the cost aspect of using hierarchical storage system for video applications, and given their model they show that files of low popularity should be directly streamed from the tertiary level. Chervenak et al. in [24] use average cost per video stream to evaluate a number of storage systems. Barnett and Anido in [25] compare the cost of a centralized video system with a distributed system. Giovanni et al. consider minimizing the total cost of a hierarchical storage system in order to meet a certain user blocking probability [14]. The above work focuses on the cost advantage of using hierarchical storage system, while our work addresses the sizing of architectural parameters of the system in order to meet a certain delay/loss requirement.

We briefly discuss some other related work as follows. Suzuki et al. in [26] study the delay issue of a hierarchical storage system given a certain secondary storage capacity, in which some files are permanently stored in the secondary level, while some others have the initial parts stored in the disk. We differ from them in the way of operating the system, and instead of *given* a secondary level storage capacity, we *size* it to meet user delay/loss requirement. Similar work is reported by Merchant et al. in [27]. Our work is mostly parallel to and independent of theirs, and with different approaches and focuses (see [28], [29]). While analysis is the main focus in [27], we have used substantial simulation here (to study system performance and to validate the analytic results), and derive many performance characteristics of a hierarchical storage system. While Won and Srivastava, and Bianchi et al. study the replacement algorithms for a hierarchical storage system [30], [31], we focus more on parameter dimensioning than the operational procedures. The use of pipelining in video staging to reduce user's start-up delay files has been discussed in [32], [33], [34]. Their schemes, however, do not offer interactive capability to the users.

III. Architectures and Operations of a Hierarchical Storage System

A. Description of system components

We show in Fig. 3 an architecture of a hierarchical storage system. The request queue holds the users whose requested videos have not been displayed yet, and the system control dispatches the accepted requests to the respective storage level for service. The secondary and tertiary levels are described as follows:

• *Secondary level:* The secondary storage level consists of disks (magnetic or re-writable optical disks), which may be configured as RAID (Redundant array of inexpensive disks) [35], [36], [37]. The secondary level has a certain aggregate "effective" bandwidth, B_2 (which depends on the access overhead in the disks and the block size). Given the streaming rate of a movie b_0, the maximum number of concurrent streams that can be opened in the secondary level is,

$$s_2 = \lfloor B_2/b_0 \rfloor, \tag{1}$$

where $\lfloor x \rfloor$ denotes the largest integer less than or equal to x. At any particular time t, a portion of B_2 is used for video streaming ($B_2^{strm}(t)$), and another portion is used for video staging ($B_2^{stg}(t)$). Obviously, we must have,

$$B_2^{stg}(t) + B_2^{strm}(t) \quad \leq \quad B_2. \tag{2}$$

B_2 can be either "partitioned" or "shared" between staging and streaming, depending on whether or not a certain amount of bandwidth is permanently set aside for staging or streaming purpose.

The secondary level has a limited storage capacity C_2; therefore the maximum number of video files that can be stored in the secondary level, N_2, is

$$N_2 = \frac{C_2}{C_f}, \tag{3}$$

and $N_3 \triangleq N_v - N_2$ is the number of non-resident video files.

• *Tertiary level:* The tertiary level is characterized by three components: i) A cabinet which shelves many ($\geq\sim 100$) removable storage media, such as tapes (magnetic tapes or optical tapes) or optical disks (e.g., CD, WORM, MO disks, or Phase-change disks, etc.); ii) a number $N_{dr}^{(3)}$ of drives, which is shared by all the tapes or disks in the library ($N_{dr}^{(3)}$ typically ranges from 2 – 16). Each drive has a certain "effective" bandwidth (taking into account file access overhead and block size), b_3; hence the maximum bandwidth the tertiary storage level can deliver at any time is given by $B_3 = b_3 N_{dr}^{(3)}$; and iii) an automated mechanism consisting of one or more robotic arms, which swap the tapes/disks in and out of the drives. The time it takes for an exchange is called exchange latency, T_{ex}.

All video files are stored in the tertiary level. Video files are staged by using a certain amount of B_3 ($B_3^{stg}(t)$) and also a portion of the secondary bandwidth B_2 ($B_2^{stg}(t)$). We obviously must have

$$B_3^{stg}(t) \quad \leq \quad B_3, \tag{4}$$
$$B_3^{stg}(t) \quad = \quad B_2^{stg}(t). \tag{5}$$

Obviously, we only need to consider $B_3 \leq B_2$.

B. Design problem

We design the hierarchical storage system with the following application characteristics:

- Poisson arrivals with rate λ (requests/minute);[1]
- Constant request holding time, T_h (minutes);
- Constant streaming bandwidth, b_0, for all video files at all times;[2] and
- All video files are of similar size C_f (GB).

For interactive video services, a file has to be completely staged in the secondary level before it is displayed. Users therefore may experience some initial delay due to staging. We address the following design problem:

Given:
- Application characteristics
  (user holding time, file characteristics, and interactive characteristics), and
- target arrival rate λ_0 (We consider large video base, i.e., $\lambda_0 T_h \ll N_v$);

Select appropriate:
- Architectural parameters:
  C_2 (secondary level storage),
  B_2 (secondary level bandwidth),
  B_3 (tertiary level bandwidth), and
  N_{req} (network streams leased); and
- Operational procedures

In order to:
Satisfy a delay requirement, e.g.,
(Start-up delay requirement) $\overline{D}_{st} \leq \hat{D}$,
(Miss delay requirement) $\overline{D}^{Miss} \leq \hat{D}^{Miss}$, or
(Statistical delay guarantee) $P(D_{st} > \hat{D}) \leq \varepsilon$.

Note that there may be a range of parameter values to meet the delay requirement. Specific value can be chosen if i) there is a cost associated with each value and the total cost of the system is to be minimized; ii) given today's technology, the feasible parameters may not take on continuous values and hence, there is only a few sets of values that are possible (e.g., a disk comes with a certain storage and bandwidth [28]); or iii) there is a "knee" in the trade-off curve, making the point likely to be the appropriate design choice in reality.

[1] λ is expressed as requests/hr in our graphs.

[2] Using simulation, we find that a system with deterministic holding time and streaming bandwidth has similar average delay performance as a system with arbitrary distribution so long as they have the same mean. Therefore, the deterministic consideration here is not so restrictive as it appears. Deterministic holding time and streaming bandwidth, however, lead to simpler and tractable analysis.

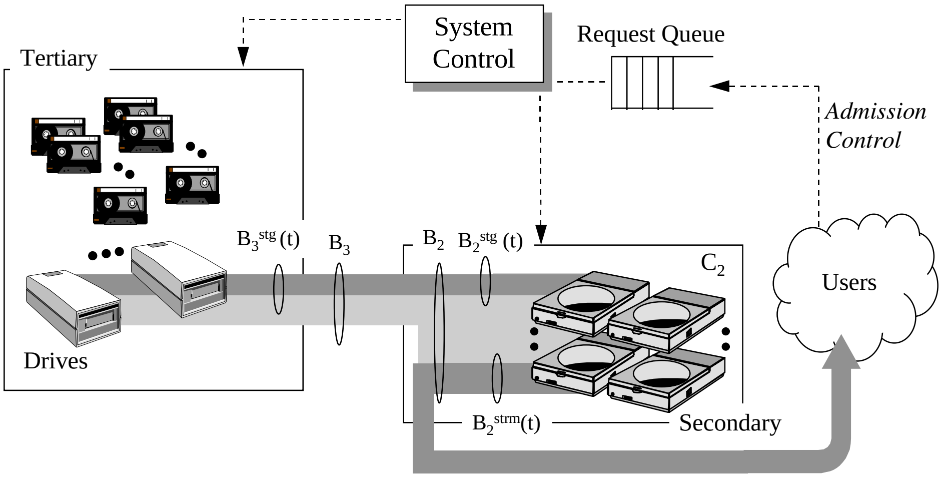

Fig. 3. A hierarchical storage system for an on-demand video server.

In any case, we see from the design problem that there are many parameters to be specified. Simulation may be used to search for the appropriate parameter values, but the process would be slow and inefficient. To facilitate the design process, we have used analytic modeling. The model developed is then validated by simulation.

IV. Designing a Hierarchical Storage System

A. System model

We show in Fig. 4 the model for the hierarchical storage system with external arrival rate λ (The design point is given by $\lambda = \lambda_0$), with each arrival independently requesting movie i with probability p_i ($1 \leq i \leq N_v$). For simplicity, we will first discuss the case with uniform video popularity (i.e., $p_i = 1/N_v$); we address the case of non-uniform popularity in Sect. VI (Note that the performance under the assumption of uniform popularity is pessimistic if video popularity were in fact skewed). With uniform popularity, it does not matter which replacement algorithm should be used; for concreteness, let's choose the Least-Recently-Used (LRU) algorithm (in which the file which is closed for the longest time is deleted).

A hit request is put into the hit queue and waits for an available stream. If a request is a miss, it joins the staging-movie queue, which keeps the name of the movies to be staged. The movies are staged according to the First-Come-First-Serve (FCFS) algorithm. Once a file is completely staged, all the misses corresponding to the movie become hits and join the hit queue.

Note that since the hit and miss rates depend on the system state, the analysis of the hierarchical storage system is difficult.[3] Furthermore, since the staging time and waiting time for a deletable file in the secondary level W_d are not exponential, the transition between system states is not a Markovian process. Due to these difficulties, we assume that the probability of an arrival finding its file *being* staged is negligible (this is justified when staging time is much shorter than the movie holding time) so that the hit probability α_2 and the miss probability α_3 are constant in time, i.e., $\alpha_2 = N_2/N_v$ and $\alpha_3 = 1 - N_2/N_v$, respectively. This is in fact verified in simulation. Therefore, the arrival process of the tertiary level is Poisson with rate $\lambda_3 \triangleq \alpha_3\lambda$.

The tertiary level and the secondary level are coupled through the waiting time for a deletable file. Furthermore, if B_2 is shared between staging and streaming, the staging bandwidth in the tertiary level would depend on the bandwidth available in the secondary level (recall the expression $B_3^{stg}(t) = \min\{B_2 - B_2^{strm}(t), B_3\}$).[4] Note that both couplings lead to staging inefficiency and hence low performance (later demonstrated in Figs. 10 and 14). When designing a system with the target arrival rate λ_0, such inefficiency due to coupling can be mostly eliminated by having large enough C_2 and B_2; the resultant system is hence a "decoupled" system with the two levels operating independently. We show in Figs. 5 and 6 the resultant subsystem models for secondary level and tertiary level, respectively. We point out here that, when $\lambda \leq \lambda_0$, the system can be analyzed using the decoupled models. However, when $\lambda > \lambda_0$, system may become inefficient and the decoupling may no longer be true and we should use simulation to obtain its performance.

Let $\overline{D}_{st}^{Hit}$ be the average delay in the hit queue, and $\overline{D}^{Miss}$ be the average delay in the tertiary subsystem. The average delay for all users is then

$$\overline{D}_{st} = \overline{D}_{st}^{Hit} + \alpha_3\overline{D}^{Miss}. \tag{6}$$

[3]This is illustrated as follows: let $\mathbf{N} = \{N_{res}, N_{stg}, N_v^{stg}\}$, $\mathbf{N} \geq 0$, be the state of the system at any point in time, where N_{res}, N_{stg} and N_v^{stg} are the number of files resident in the secondary level, files being staged and the total number of files to be staged (including those being staged), respectively. Obviously, we have $N_{stg} \leq \min\{N_{dr}^{(3)}, N_v^{stg}\}$, $N_{res} + N_v^{stg} \leq N_v$, and $N_{res} \leq N_2$. With large N_v, the state space is large (of order $O(N_v^2)$) and the transitions can be quite complex.

[4]Obviously, there would be no such coupling if B_2 is partitioned between staging and streaming.

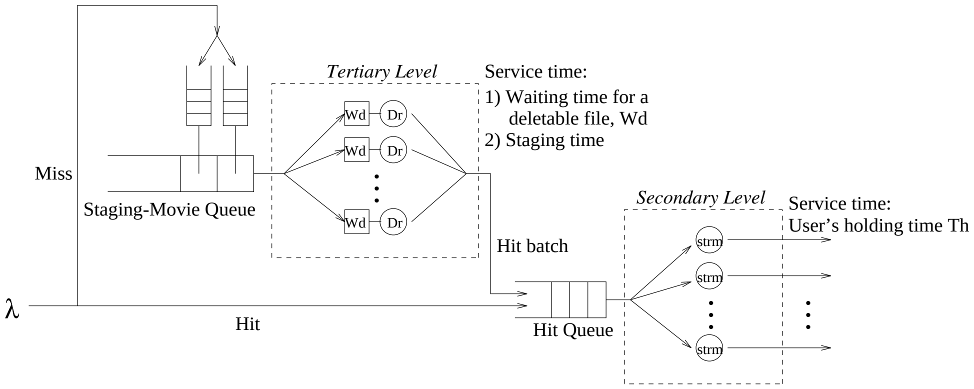

Fig. 4. A model for a hierarchical storage system.

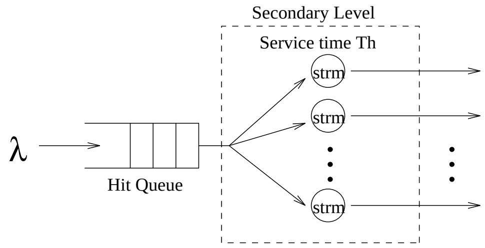

Fig. 5. Secondary sub-system model.

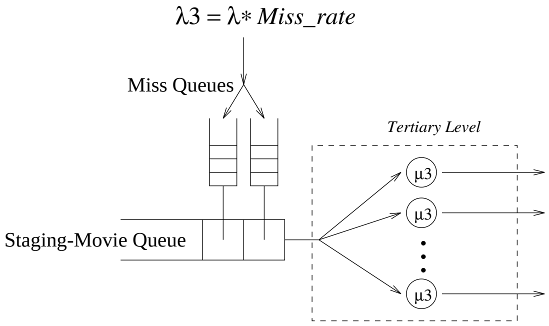

Fig. 6. Tertiary sub-system model.

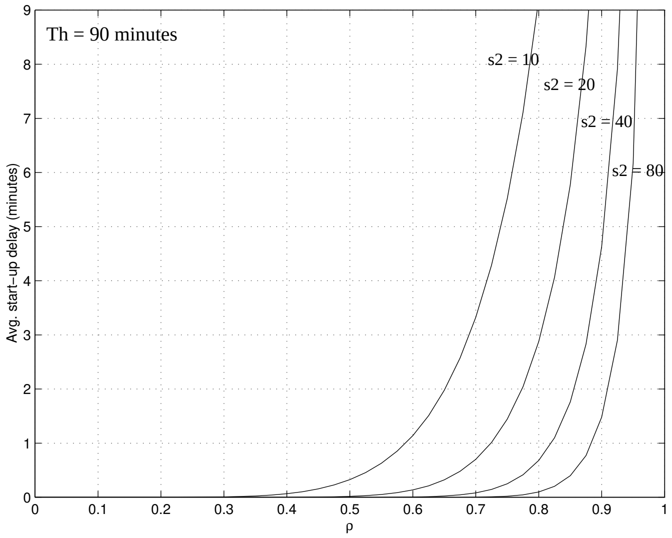

Fig. 7. $\overline{D}_{st}^{Hit}$ for the secondary subsystem versus ρ ($\triangleq \lambda T_h/s_2$).

B. Secondary subsystem design

The secondary level has a hit queue and s_2 streams to be shared between streaming and staging. With large number of video titles (i.e., $\lambda_0 T_h \ll N_v$), each request requires a storage space C_f (and a stream) for a duration of T_h minutes.

The arrival rate to the secondary system is λ. Using simulation, we found that the arrival process in the secondary level can be approximated very well by a Poisson process [38]. Hence, the secondary level can be modeled as an $M/D/s_2$ system with holding time T_h. Therefore, the average hit delay is given by [39] (recall that hits are served first in our system),

$$\overline{D}_{st}^{Hit} \approx T_h \frac{(\lambda T_h)^{s_2}}{2(s_2-1)!(s_2-\lambda T_h)^2\left[\sum_{j=0}^{s_2-1}\frac{(\lambda T_h)^j}{j!}+\frac{(\lambda T_h)^{s_2}}{(s_2-1)!(s_2-\lambda T_h)}\right]}. \tag{7}$$

We show in Fig. 7 the average hit delay $\overline{D}_{st}^{Hit}$ as the utilization $\rho \triangleq \lambda T_h/s_2$ increases, given s_2 ($T_h = 90$ minutes). We see that as ρ increases, the delay is quite low initially and then increases rapidly. If the streaming capacity of the system is not very low and the system is not over-loaded (i.e., consider a system with $s_2 = 40$ or 80, and $\rho \le 0.8$), the hit delay is very low (≤ 1 minute).

We next obtain the C_2 to achieve negligible waiting time for a deletable file, given the arrival rate λ_0. Note that since on average $\lambda_0 T_h$ files would be open (since the staging time is negligible compared with T_h), for stability we must have $C_2 \ge \lambda_0 T_h C_f$. Letting $C_2 = (\lambda_0 T_h + n_2) C_f$, we plot in Fig. 8 the average waiting time for a deletable file $\overline{W}_d$ against n_2 (based on simulation). We see that n_2 can be very small, in the range of only 8 – 18, to achieve very low waiting time for a deletable file. The network channels needed for streaming can also be read from the same graph, i.e., $N_{req} = \lambda_0 T_h + n_2$.

If B_2 is partitioned between staging and streaming, the total B_2 required is then $B_2 = (\lambda_0 T_h + n_2) b_0 + B_3$. On the other hand, if it is shared, the required B_2 may be lower, i.e., $B_2 = (\lambda_0 T_h + n_2) b_0 + B_3 + \Delta$, with $\Delta \le 0$. For a well-designed system in which the B_3 utilization is not

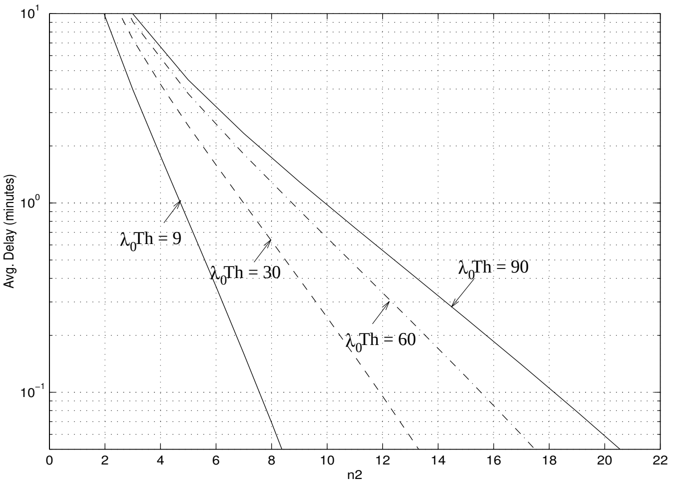

Fig. 8. $\overline{W}_d$ vs. n_2, given $\lambda_0 T_h$.

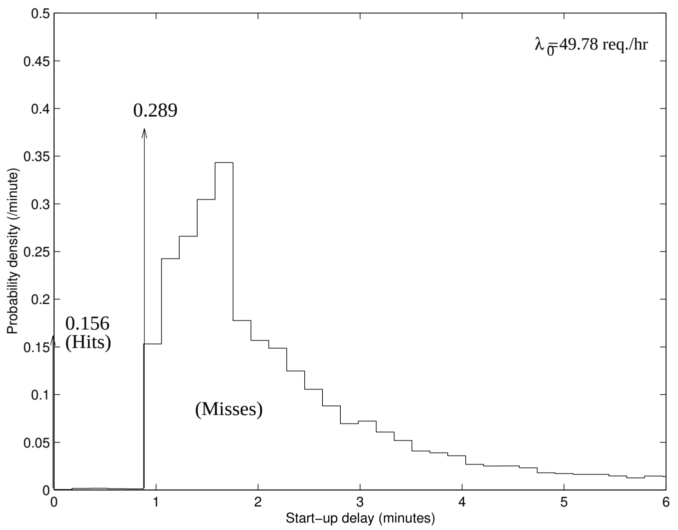

Fig. 9. Distribution for the start-up delay ($\lambda_0 \simeq 50$ req./hr, $C_2 =$ 78 GB, $B_2 = 35$ MB/s, $B_3 = 19$ MB/s, $N_{dr}^{(3)} = 1$, $b_0 = 1.5$ Mb/s, $C_f = 1$ GB, $N_v = 500$, and $T_h = 90$ minutes).

low, simulations have shown that $\Delta \approx 0$ (indicating that there is no negative correlation between staging bandwidth and streaming bandwidth), giving $B_2 \approx (\lambda_0 T_h + n_2)b_0 + B_3$ even for the shared case.

Now the only parameter to be determined is B_3, which is obtained by examining the tertiary subsystem.

C. Tertiary subsystem design

The tertiary subsystem consists of $N_{dr}^{(3)}$ drives, each with service time $C_f/b_3 + T_{ex}$. In general, $T_{ex} \ll C_f/b_3$ and hence, for simplicity, we will consider in the following that T_{ex} can be ignored (T_{ex} can be factored into our expressions if it were not negligible). The arrival rate in the tertiary level is Poisson with rate $\lambda_3 = \alpha_3\lambda$ (at $\lambda = \lambda_0$, λ_3 is hence given by $\alpha_3\lambda_0$). Since N_v is large, each miss would likely choose a movie different from the outstanding ones to be staged. Therefore the subsystem can be modeled as an $M/D/N_{dr}^{(3)}$ system. We will use an M/M model due to its simplicity and similarity to the M/D queue [38]. The performance obtained using an M/M model is also pessimistic.

We show in Fig. 9 the typical start-up delay distribution for the hierarchical storage system. The impulse at the origin corresponds to the hits, indicating that hits enjoy low delay. There is another impulse corresponding to those misses with no queuing delay (and hence its start-up delay is given by C_f/B_3) — the spread of the distribution is due to the waiting time for an available drive in the tertiary level. Given the low hit delay, we may approximate the average delay for all users $\overline{D}_{st}$ in Eq. (6) as,

$$\overline{D}_{st} \simeq \alpha_3 \overline{D}^{Miss}. \tag{8}$$

Since the user delay is determined by the misses, B_3 is chosen to satisfy user delay requirement.

Given an overall average delay requirement $\hat{D}$, the miss delay requirement, $\hat{D}^{Miss}$, is obtained as,

$$\begin{aligned} \hat{D}^{Miss} &\simeq \hat{D}/\alpha_3 \\ &= \hat{D}\frac{N_v}{N_v - N_2}. \end{aligned} \tag{9}$$

Given $\hat{D}^{Miss}$, we solve for μ_3 by using the $M/M/N_{dr}^{(3)}$ formula [40], [41], i.e,

$$\hat{D}^{Miss} = \frac{1}{\mu_3} + \frac{P_Q}{N_{dr}^{(3)}\mu_3 - \lambda_3}, \tag{10}$$

where P_Q is given by ($\rho^{(3)}$ is the tertiary level utilization given by $\lambda_3/(N_{dr}^{(3)}\mu_3)$),

$$\begin{aligned} P_Q &= \left[\sum_{n=0}^{N_{dr}^{(3)}-1} \frac{(N_{dr}^{(3)}\rho^{(3)})^n}{n!} + \frac{(N_{dr}^{(3)}\rho^{(3)})^{N_{dr}^{(3)}}}{N_{dr}^{(3)}!(1-\rho^{(3)})}\right]^{-1} \times \\ & \frac{(N_{dr}^{(3)}\rho^{(3)})^{N_{dr}^{(3)}}}{N_{dr}^{(3)}!(1-\rho^{(3)})}. \end{aligned} \tag{11}$$

The above equation can be solved numerically when $N_{dr}^{(3)} \geq 2$ (closed-form expression for μ_3 exists when $N_{dr}^{(3)} = 1$). Given μ_3, the required B_3 is then given by $B_3 = N_{dr}^{(3)} C_f \mu_3$.

D. Performance characteristics

Given our model, we obtain the following performance characteristics of the hierarchical storage system.

• *Marginal and elasticity of substitution: The marginal substitution and the elasticity of substitution between B_3 and C_2 are given by,*

$$\frac{\partial B_3}{\partial C_2} \approx -\frac{N_{dr}^{(3)} + \hat{D}\lambda_0}{\hat{D}N_v}, \tag{12}$$

$$\frac{\partial B_3/B_3}{\partial C_2/C_2} \approx -\frac{N_2}{N_v - N_2}, \tag{13}$$

where $\hat{D}$ is the average delay requirement in the system.

Marginal substitution reflects how flexible two variables can be traded off with each other. In a hierarchical storage system, a higher tertiary bandwidth B_3 can generally be traded with a lower secondary storage C_2, and vice versa. In a system designed for low waiting time (our case of interest), user delay is comparable with the tertiary staging time. Let $\overline{T}_{M/M/n}$ be the average system time for an $M/M/n$ queue. Since the tertiary level may be approximated by an M/M system, the average delay of the system $\hat{D}$ is given by,

$$\begin{aligned}\hat{D} &\approx \alpha_3 \overline{T}_{M/M/N_{dr}^{(3)}} \\ &\approx \frac{N_3}{N_v} N_{dr}^{(3)} \overline{T}_{M/M/1} \\ &= \frac{N_3}{N_v} \frac{N_{dr}^{(3)}}{N_{dr}^{(3)}\mu_3 - \lambda_3},\end{aligned} \quad (14)$$

which gives,

$$\mu_3 = \frac{1 + \hat{D}\lambda_0 / N_{dr}^{(3)}}{\hat{D}(N_v/N_3)}. \quad (15)$$

With $B_3 = N_{dr}^{(3)} b_3 = N_{dr}^{(3)} C_f \mu_3$, we have

$$\begin{aligned}B_3 &\simeq \frac{C_f(N_{dr}^{(3)} + \hat{D}\lambda_0)}{\hat{D}N_v}(N_v - N_2) \\ &= \frac{N_{dr}^{(3)} + \hat{D}\lambda_0}{\hat{D}N_v}(C_f N_v - C_2).\end{aligned} \quad (16)$$

Therefore, $\partial B_3/\partial C_2$ is given according to Eq. (12). We remark finally that, since $B_2 \approx B_3 + (\lambda_0 T_h + n_2)b_0$, $\partial B_2/\partial B_3 \approx 1$, $\partial B_2/\partial C_2 = \partial B_3/\partial C_2$.

The "elasticity" of substitution measures how well two variables can trade off with each other in terms of their percentage change, and for bandwidths and storage, such elasticity is given by $(\partial B_2/B_2)/(\partial C_2/C_2) = (\partial B_3/B_3)/(\partial C_2/C_2) = \partial \ln B_3/\partial \ln C_2$. Using Eq. (16), we get the elasticity of substitution according to Eq. (13). Note that usually the elasticity is low (since $N_2 \ll N_v$), meaning that a large percentage increase in C_2 only leads to a relatively small decrease in B_3.

For illustrative purpose, we show in Fig. 10 the trade-off between C_2 and B_3 given $\overline{D}_{st}$ (based on simulation) ($\lambda_0 = 30$ req./hr, $B_2 = 20$ MB/s, $N_{dr}^{(3)} = 1$, and $N_v = 500$). The linear region indicates the trade-off between C_2 and B_3 (little coupling between the two levels), with a slope given by $\partial C_2/\partial B_3$. There is a limit to such a trade-off indicated by the horizontal asymptote due to conflict of resources (i.e., coupling between the two levels) — at low $\overline{D}_{st}$ (i.e., $overline D_{st}$ in the range of 1–2 minutes), such limit is due to insufficient B_2; while at higher $\overline{D}_{st}$ (i.e., with $\overline{D}_{st}$ more than 6 minutes), such limit is due to long waiting time for a deletable file (coupling between the storage levels) and, as discussed earlier, at $C_2 \approx (\lambda_0 T_h + n_2)C_f$. We therefore see that when the two levels are coupled, inefficiency and low performance result. We finally remark that, since B_2 and B_3 cannot be traded off with each other, the trade-off curve between C_2 and B_2 would have similar trends.

• *Performance invariance: Two hierarchical storage systems with the same user holding time have the same delay performance if the ratio of their parameters ($b_0(i)$, $C_f(i)$, $C_2(i)$,$B_2(i)$,$B_3(i)$) (where $i = 1, 2$ is the system index) scales according to the following:*

$$\frac{b_0(2)}{b_0(1)} = \frac{C_f(2)}{C_f(1)} = \frac{C_2(2)}{C_2(1)} = \frac{B_2(2)}{B_2(1)} = \frac{B_3(2)}{B_3(1)}. \quad (17)$$

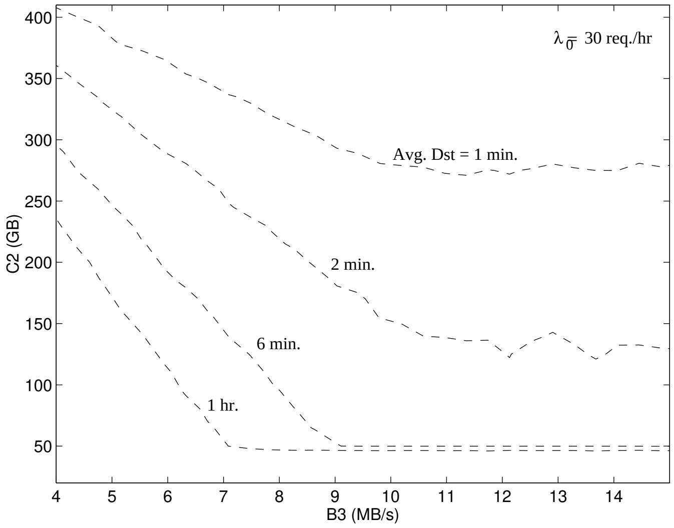

Fig. 10. Trade-off between C_2 and B_3 to achieve a given $\overline{D}_{st}$ ($\lambda_0 = 30$ req./hr, $B_2 = 20$ MB/s, $N_v = 500$, $N_{dr}^{(3)} = 1$ and uniform video popularity).

First note that if b_0 is scaled by a certain factor, the required secondary streaming bandwidth would then be similarly scaled. If both C_f and C_2 are scaled by the same factor, the miss probabilities will be the same in both systems. If C_f and B_3 are scaled by the same factor, μ_3 will remain the same and hence is the miss delay. Now if B_2 is also similarly scaled, both systems will have the same performance. Analogously, following the same line of argument, if T_h changes while b_0 remains unchanged, system performance remains the same if $T_h(2)/T_h(1) = C_f(2)/C_f(1) = C_2(2)/C_2(1) = B_2(2)/B_2(1) = B_3(2)/B_3(1)$.

As an example, we show in Fig. 11 $\overline{D}_{st}$ versus λ based on simulation, for two systems with b_0 changed from 1.5 Mb/s to 3 Mb/s and the other parameters C_f, C_2, B_2, and B_3 scaled accordingly (while maintaining the same holding time $T_h = 90$ minutes). Obviously the two systems have the same performance.

• *Number of robotic arms in the tertiary level: In order to achieve low waiting time for a free robotic arm, the number of arms N_{rbt} should satisfy,*

$$N_{rbt} \gg (1 - \alpha_2)\lambda_0 T_{ex}. \quad (18)$$

Note that the utilization of a robotic arm in the tertiary level is given by $\lambda_3 \overline{T}_{ex}/N_{rbt}$. Therefore, to achieve low waiting time for a free arm, we should have $\lambda_3 \overline{T}_{ex}/N_{rbt} \ll 1$, giving the condition above.

• *Drive parallelism in the tertiary level: For interactive applications, in order to decrease start-up delay, the drive*

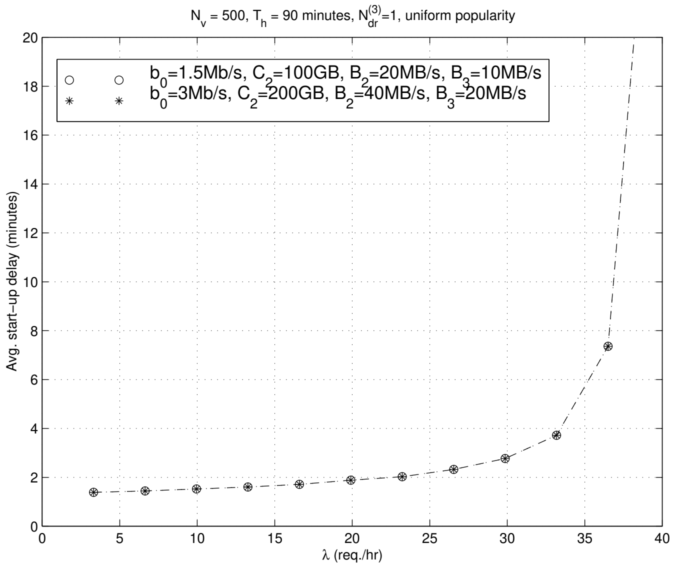

Fig. 11. Performance invariance for a change in b_0 (from 1.5 Mb/s to 3 Mb/s) ($N_{dr} = 1$, $N_v = 500$ and uniform video popularity).

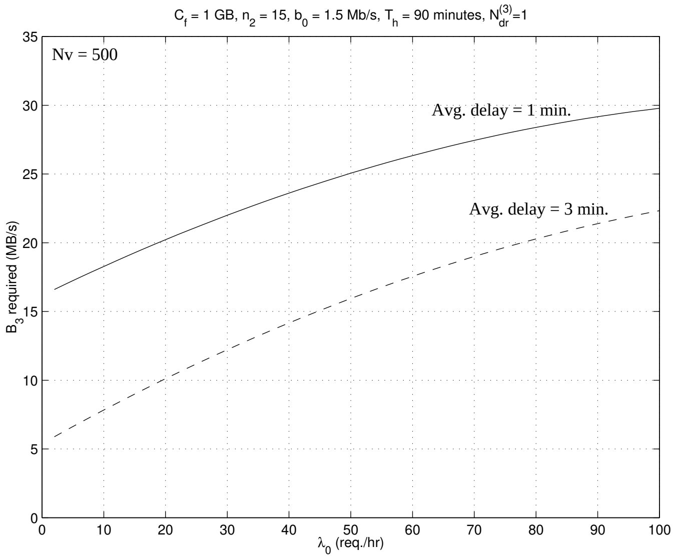

Fig. 12. B_3 required vs. λ_0 given $\hat{D}$ ($N_v = 500$, $C_f = 1$ GB, $b_0 = 1.5$ Mb/s, $T_h = 90$ minutes, $N_{dr}^{(3)} = 1$, $n_2 = 15$).

bandwidth in the tertiary level should be used in parallel. Since a video file cannot be displayed until it is completely staged, to reduce the staging time, the drive bandwidth should therefore be used in parallel (making $N_{dr}^{(3)} = 1$).

V. Design examples

In this section, we illustrate some design examples of specifying the required storage and bandwidth to meet a certain user delay requirement given by $\overline{D}_{st} \leq \hat{D}$. We design the system at the trade-off "knee."

Figure 12 shows the required B_3 versus the target arrival rate λ_0, given $\hat{D}$ ($N_v = 500$ ($C_f = 1$ GB, $b_0 = 1.5$ Mb/s, $T_h = 1.5$ hr., $N_{dr}^{(3)} = 1$, $n_2 = 15$). The required B_3 increases rather linearly in λ_0. Though the required B_3 for

TABLE I

Design specifications to achieve $\hat{D} = 2$ minutes ($C_f = 1$ GB, $b_0 = 1.5$ MB/s, $T_h = 90$ minutes, $N_{dr}^{(3)} = 1$, $n_2 = 15$).

	$\lambda_0 = 20$ req./hr.	$\lambda_0 = 50$ req./hr
C_2 (GB)	45	90
α_3	0.91	0.82
B_3 (MB/s)	12.63	18.21
B_2 (MB/s)	21.1	35.1

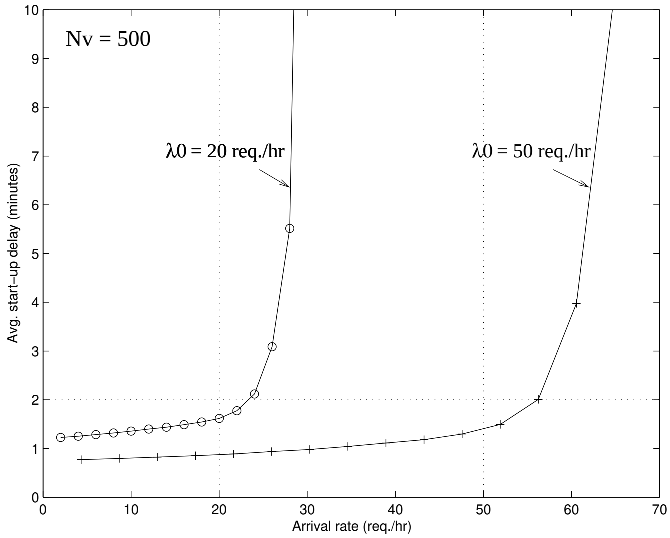

Fig. 13. Server design satisfying $\hat{D} = 2$ minutes at $\lambda_0 = 20$ req./hr or $\lambda_0 = 50$ req./hr ($N_v = 500$, $C_f = 1$ GB, $b_0 = 1.5$ Mb/s, $T_h = 90$ minutes, $N_{dr}^{(3)} = 1$, and $n_2 = 15$).

$\hat{D} = 1$ minute is higher than that with $\hat{D} = 3$ minutes, it is not three times as high.

We show in Table I the design of a hierarchical server satisfying $\hat{D} = 2$ minutes, given a target arrival rate ($N_v = 500$, $C_f = 1$ GB, $T_h = 1.5$ hr., $b_0 = 1.5$ Mb/s, $N_{dr}^{(3)} = 1$, $n_2 = 15$). We see that the required C_2 is much less than $N_v C_f$, showing that the hierarchical storage system is able to greatly reduce the system storage cost while satisfying the user delay requirement. There is also an "economy of scale" in terms of the bandwidth and storage requirements — higher λ_0 does not lead to a proportional increase in the requirements. We show in Fig. 13 the system performance (by simulation) using the parameters specified in Tab. I. Clearly, the delay requirement at λ_0 is met, showing that our model is useful. Note that the delay at λ_0 is slightly lower than what is required, because we have used the simpler, and yet slightly pessimistic, M/M model.

Deployed servers designed for a particular target arrival rate λ_0 may be used at a different rate. Figure 13 shows how an "optimized" design would perform as the arrival rate changes. Though the server satisfies the delay requirement at λ_0, the delay increases rapidly as the arrival rate

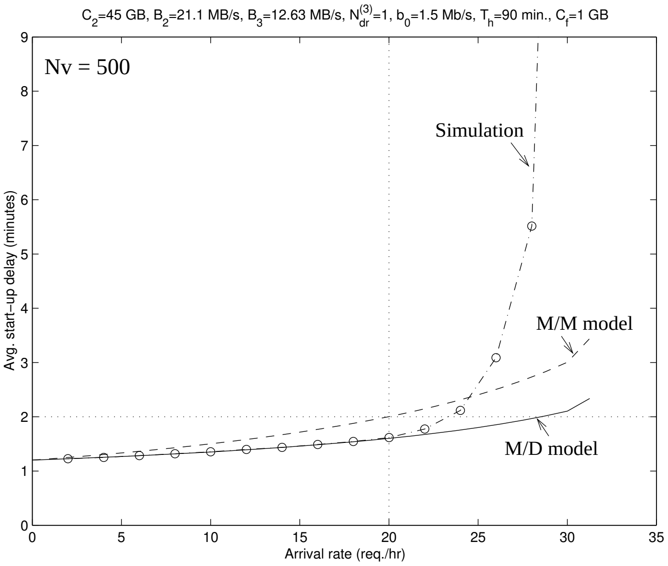

Fig. 14. Comparison of simulation and analysis ($N_v = 500$, $C_2 = 45$ GB, $B_2 = 21.1$ MB/s, $B_3 = 12.63$ MB/s, $C_f = 1$ GB, $b_0 = 1.5$ Mb/s, $T_h = 90$ minutes, $N_{dr}^{(3)} = 1$, and $n_2 = 15$).

increases beyond λ_0. Therefore, in designing a hierarchical video server, it is important to consider the maximum arrival rate in order to prevent unacceptable delay should the arrival rate increase.

In Fig. 14 we compare our analytic model with simulation, using the parameters satisfying $\hat{D} = 2$ minutes and $\lambda_0 = 20$ req./hr. We show two analytic curves, one based on the M/M model (exponentially-distributed staging time) and another one based on the M/D model (deterministic staging time), with the total staging bandwidth available at any time given by $\min\{B_2 - \lambda T_h b_0, B_3\}$ (assuming negligible waiting time for a deletable file). We clearly see that the M/D model is accurate up to the knee, beyond which the two storage levels are coupled which the analytic model can no longer capture. We see that the decoupling assumption is indeed good up to λ_0. The performance of the M/M model is apparently a bit pessimistic up to the knee.

VI. Non-uniform video popularity

Our model can be extended to design a hierarchical storage system with non-uniform video popularity. Consider a system with multiple video popularity classes, within each of which video popularity is uniform. Let A be the number of popularity classes, each of which contains $N_{v,a}$ videos, where $1 \leq a \leq A$. Obviously, the total number of videos is,

$$N_v = \sum_{a=1}^{A} N_{v,a}. \tag{19}$$

Let P_a be the probability of a random arrival choosing popularity class a (the request rate for each class is hence given by $\lambda_a = P_a \lambda_0$). Obviously, $\sum_a P_a = 1$, and the popularity of the ith movie in class a is given by $p_{i,a} = P_a / N_{v,a}$, for $1 \leq i \leq N_{v,a}$.

TABLE II

Design specifications to achieve $\hat{D} = 2$ minutes, with $A = 3$ popularity classes ($C_f = 1$ GB, $b_0 = 1.5$ MB/s, $T_h = 90$ minutes, $N_{dr}^{(3)} = 1$ and $n_2 = 15$).

	$N_v = 500$	
	$\lambda_0 = 20$ req./hr.	$\lambda_0 = 50$ req./hr
C_2 (GB)	45	90
α_3	0.8382	0.6764
B_3 (MB/s)	11.64	15.03
B_2 (MB/s)	20.07	31.9

The design of such a hierarchical storage system follows the same steps as given in Sect. IV. The C_2 required is still $(\lambda_0 T_h + n_2) C_f$. Since the extra n_2 space in the secondary level is shared among the popularity classes, the hit probability for class a can be estimated as,

$$\alpha_{2,a} = \frac{\lambda_a T_h + P_a n_2}{N_{v,a}}. \tag{20}$$

Therefore the overall miss probability is,

$$\alpha_3 = \sum_{a=1}^{A} P_a (1 - \alpha_{2,a}), \tag{21}$$

and the arrival rate to the tertiary level is given by $\lambda_3 = \alpha_3 \lambda_0$. Finally, the required B_2 is given by $B_3 + (\lambda_0 T_h + n_2) b_0$.

As an example, let's consider designing a server with $\hat{D} = 2$ minutes, with $A = 3$ popularity classes. The class popularities are $P_1 = 0.05$ (unpopular movies), $P_2 = 0.25$, and $P_3 = 0.7$ (popular movies), and the number of movies in each class is $N_{v,1} = 150$, $N_{v,2} = 200$, and $N_{v,3} = 150$ (and hence $N_v = 500$), respectively. Table II shows the bandwidth and storage requirements for the system. The requirements in Tab. II is slightly lower than that of the uniform popularity case (Ref. Tab. I). This is because the miss rate α_3 in this example is only slightly lower than that of the uniform case (If the video popularity is more skewed, we expect a more significant decrease in bandwidth and storage requirements). We present in Fig. 15 the performance of the system versus λ given the above specifications (based on simulation), showing that the performance goal is met at λ_0.

VII. Conclusions

Video-on-demand encompasses important services such as movie-on-demand, distance-learning, news-on-demand, home-shopping, various distributed/interactive training programs, etc. In order to accommodate large number of video titles, the storage of a video system has to be scalable. We have studied how such scalability can be achieved with a hierarchical storage system consisting of a secondary level (characterized by fast file access and high throughput) and a tertiary level (characterized by low-cost and large storage). In such a system, video files are stored in the tertiary level and staged onto the secondary level to be displayed.

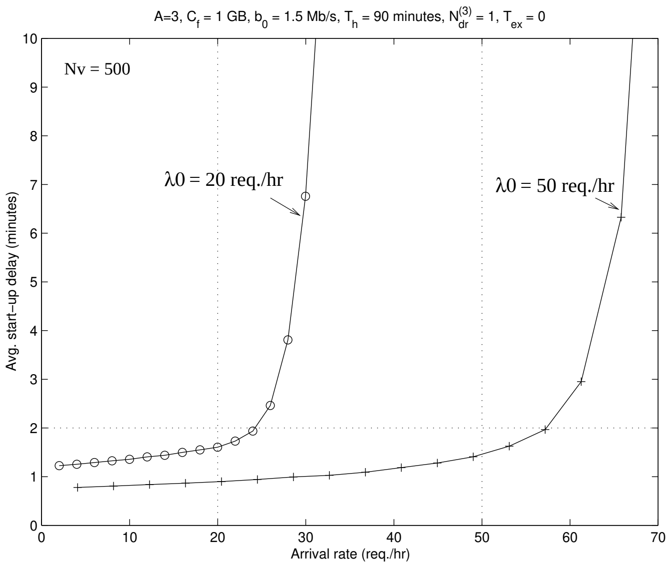

Fig. 15. Designing a hierarchical storage system with non-uniform video popularity ($\hat{D} = 2$ minutes, $\lambda_0 = 20$ req./hr and 50 req./hr, $A = 3$, $N_v = 500$, $C_f = 1$ GB, $b_0 = 1.5$ Mb/s, $T_h = 90$ minutes, and $N_{dr}^{(3)} = 1$).

In order to provide interactive services, video files have to be completely staged onto the secondary level before they can be displayed; and hence users experience some start-up delay in the system. Through simulation and analysis, we have developed a very simple and yet accurate model for a hierarchical storage system, which greatly facilitates the design of such a system and allows us to extract important performance characteristics of the system. Our model is used to specify the storage and bandwidth (bandwidths in each level, network channels required, and secondary level storage) as needed in order to meet a specific delay requirement, given application characteristics and a target arrival rate.

There are two types of users in the system (hits and misses). Hits enjoy negligible delay, and therefore the tertiary bandwidth should be specified to meet user delay requirement. The tertiary level should be efficient in such a system, and, for interactive applications, we should parallelize the tertiary drives. We found that the user delay deteriorates quickly as the arrival rate increases beyond a certain point, indicating the importance of admission control in the system. We have shown that the number of files stored in the secondary level can be much lower than the total number of files in the system while satisfying user delay requirement, and hence a hierarchical storage system is able to achieve much lower system cost compared to a system with secondary storage only.

References

[1] T. Little and D. Venkatesh, "Prospects for interactive video-on-demand," *IEEE Multimedia Magazine*, pp. 14–24, Fall 1994.

[2] V. O. Li and W. Liao, "Distributed multimedia systems," *Proceedings of the IEEE*, vol. 85, pp. 1063–1108, July 1997.

[3] F. A. Tobagi, "Distance learning with digital video," *IEEE Multimedia Magazine*, pp. 90–94, Spring 1995.

[4] K. Cleary, "Video on demand — competing technologies and services," in *Proceedings of International Broadcasting Convention*, (Amsterdam, Netherlands), pp. 432 – 437, 14–18 Sept. 1995.

[5] B. Furht, D. Kalra, F. L. Kitson, A. A. Rodriguez, and W. E. Wall, "Design issues for interactive television systems," *IEEE Computer Magazine*, pp. 25–38, May 1995.

[6] R. L. Haskin and F. L. Stein, "A system for the delivery of interactive television programming," in *Compcon: Digest of papers*, pp. 209–215, 1995.

[7] F. A. Tobagi, J. Pang, R. Baird, and M. Gang, "Streaming RAID^{TM} – a disk array management system for video files," in *Proceedings of First ACM International Conference on Multimedia*, (Anaheim, CA), August 1-6 1993.

[8] R. Buck, "The Oracle media server for nCube massively parallel systems," in *Proceedings of the 8th International Parallel Processing Symposium*, pp. 670–673, IEEE, 1994.

[9] A. Laursen, J. Olkin, and M. Porter, "Oracle media server framework," in *Compcon: Digest of Papers*, pp. 203–208, IEEE, 1995.

[10] A. N. Mourad, "Issues in the design of a storage server for video-on-demand," *ACM Multimedia Systems Journal*, vol. 4, pp. 70–86, April 1996.

[11] H. M. Vin and V. Rangan, "Designing a multiuser HDTV storage server," *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 153–164, January 1993.

[12] W. D. Sincoskie, "System architecture for a large scale video on demand service," *Computer Networks and ISDN Systems*, vol. 22, pp. 155–162, 1991.

[13] Y. N. Doğanata and A. N. Tantawi, "Making a cost-effective video server," *IEEE Multimedia Magazine*, pp. 22–30, Winter 1994.

[14] L. D. Giovanni, A. M. Langellotti, L. M. Patitucci, and L. Petrini, "Dimensioning of hierarchical storage for video on demand services," in *Proceedings of the IEEE ICC'94*, pp. 1739–1743, 1994.

[15] T. Nemoto, Y. Sato, K. Mogi, K. ichiro Ayukawa, M. Kitsuregawa, and M. Takagi, "Performance evaluation of cassatte migration mechanism for scalable tape archiver," in *Proceedings of SPIE High-Density Data Recording and Retrieval Technologies*, (Philadelphia, PA), Oct 1995.

[16] L. Golubchik, R. Muntz, and R. Watson, "Analysis of striping techniques in robotic storage libraries," in *Proceedings of IEEE 14th Symposium on Mass Storage Systems*, (Monterey, CA), pp. 225–38, 11-14 Sept. 1995.

[17] A. Drapeau and R. Katz, "Striped tape arrays," in *Proceedings of IEEE Symposium on Mass Storage Systems*, pp. 257–265, 1993.

[18] D. A. Ford, R. J. T. Morris, and A. E. Bell, "Redundant arrays of independent libraries (rail): A tertiary storage system," in *Proceedings of COMPCON'96*, pp. 280–285, 1996.

[19] S.-W. Lau and J. C. S. Lui, "Scheduling and data layout policies for a near-line multimedia storage architecture," *Multimedia Systems*, vol. 5, pp. 310–323, September 1997.

[20] C. Schünemann and W. G. Spruth, "Storage hierarchy technology and organization," in *Digital Memory and Storage* (W. E. Proebster, ed.), pp. 377–389, Braunschweig: Vieweg, 1978.

[21] P. N. Misra, "Capacity analysis of the mass storage system," *IBM System Journal*, vol. 20, no. 3, pp. 346–364, 1981.

[22] O. I. Pentakalos, D. A. Menascé, M. Halem, and Y. Yesha, "Analytic performance modelling of hierarchical mass storage systems," *IEEE Transactions on Computers*, vol. 46, pp. 1103–1118, October 1997.

[23] M. G. Kienzle, A. Dan, D. Sitaram, and W. Tetzlaff, "Using tertiary storage in video-on-demand servers," in *Compcon: Digest of Papers*, pp. 225–232, IEEE, 1995.

[24] A. L. Chervenak, D. A. Patterson, and R. H. Katz, "Choosing the best storage system for video service," in *Proceedings of ACM Multimedia*, pp. 109–119, 1995.

[25] S. A. Barnett and G. J. Anido, "A cost comparison of distributed and centralized approaches to video-on-demand," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 1173–1183, August 1996.

[26] H. Suzuki and K. Nishimura, "Performance analysis of a storage hierarchy for video servers," *Systems and Computers in Japan*, vol. 28, pp. 300–308, March 1997.

[27] A. Merchant, Q. Ren, and B. Sengupta, "Hierarchical storage servers for video on demand: Feasibility, design and sizing," in *Proceedings of IEEE Globecom'96*, (London, UK), pp. 272–278, 18-22 November 1996.

[28] S.-H. G. Chan and F. A. Tobagi, "Hierarchical storage systems for on-demand video servers," in *Proceedings of the SPIE (the International Society for Optical Engineering) High-Density Data Recording and Retrival Technologies*, (Philadelphia, PA), pp. 103–120, SPIE, October 1995.

[29] S.-H. G. Chan and F. A. Tobagi, "Hierarchical storage systems for interactive video-on-demand." Technical report, CSL-TR-97-723, Computer Systems Laboratory, Stanford University, April, 1997.

[30] Y. Won and J. Srivastava, "Minimizing blocking probability in a hierarchical storage based vod server," in *Proceedings of International Workshop on Multimedia Database Management Systems*, (Blue Mountain Lake, NY, USA), pp. 12–19, 14-16 Auguest 1996.

[31] G. Bianchi, R. Melen, and A. Rainoni, "Performance of video storage hierarchy in interactive video services networks," in *Proceedings of IEEE Globecom'95*, (Singapore), pp. 815–810, 13-17 November 1995.

[32] S. Ghandeharizadeh and C. Shahabi, "On multimedia repositories, personal computers, and hierarchical storage systems," in *Proceedings of ACM Multimedia*, pp. 407–416, 1994.

[33] S. Ghandeharizadeh, A. Dashti, and C. Shahabi, "A pipelining mechanism to minimize the latency time in hierarchical multimedia storage managers," *Computer Communication*, vol. 18, pp. 170–184, March 1995.

[34] J. Z. Wang and K. A. Hua, "A bandwidth management technique for hierarchical storage in large-scale multimedia servers," in *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, (Otta, Ont., Canada), pp. 261–268, IEEE, 3–6 June 1997.

[35] D. Patterson, G. Gibson, and R. Katz, "A case for redundant arrays of inexpensive disks (RAID)," in *Proceedings of ACM SIGMOD Conference on the Management of Data*, pp. 109–116, 1988.

[36] G. Ganger, B. Worthington, R. Hou, and Y. Patt, "Disk arrays – high-performance, high reliability storage subsystems," *Computer Magazine*, pp. 30–36, March 1994.

[37] P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz, and D. A. Patterson, "RAID: High-performance, reliable secondary storage," *ACM Computing Surveys*, vol. 26, pp. 145–185, June 1994.

[38] S.-H. G. Chan and F. Tobagi, "Scalable services for video-on-demand." Technical report, CSL-98-773, Computer Systems Laboratory, Stanford University, December 1998.

[39] S. A. Nozaki and S. M. Ross, "Approximations in finite-capacity multi-server queues with poisson arrivals," *J. of Appl. Prob.*, no. 15, pp. 826–834, 1978.

[40] L. Kleinrock, *Queueing Systems: Theory*, vol. 1. Wiley Interscience, 1975.

[41] H. Kobayashi, *Modeling and Analysis: An Introduction to System Performance Evaluation Methodology.* Addison Wesley, 1981.