Minimizing Blocking Probability in a Hierarchical Storage Based VOD Server

Youjip Won Jaideep Srivastava

Distributed Multimedia Research Center
Department of Computer Science, University of Minnesota
{yjwon|srivasta}@cs.umn.edu

Abstract

In this paper we investigate the relationship between the blocking probability and the configuration of storage hierarchy in a VOD server. Due to limited system resources available in a VOD server, some user requests can not be serviced immediately and are thus blocked. In general, one would expect the increasing the I/O rate will reduce the request blockage. However, we report the interesting observation that increasing the data rate from the secondary storage may increase the blocking probability, and hence degrade the server performance. We provide an analytical model of the blocking probability, which is to be used to find the optimal data rate to minimize the blocking probability for a given system configuration. By enforcing the files to reside at the staging disk for a certain amount of time, it is possible to further improve the blocking probability. We examine the problem of space management for video files on the staging disks, and devise a policy called Enforced Duration *to arbitrates the duration of the file on the staging disk. Results of a simulation based performance evaluation are presented.*

1. Introduction

1.1. Motivation

The advent of continuous media data types such as video and audio opens a new era of information. Although handling continuous media may be achievable with the advances in technology, the cost of the contents and its management will not be easily affordable. Hence, it is desirable that the cost of expensive data is shared by as many people as possible and that the data is stored in a hierarchically organized storage system to achieve the appropriate tradeoff[5].

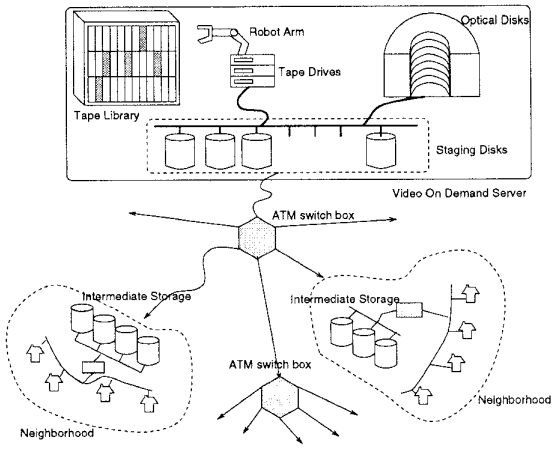

Figure 1. VOD server and global model of the metropolitan area

A *VOD* server is expected to store several thousand continuous media files such as video or audio clips, and its service must cover a metropolitan area, using high speed communication networks. Fig. 1 illustrates the storage organization of a *VOD* server and a high level view of generic network structure over a metropolitan area. The *VOD* server consists of several levels of storage hierarchies - a tape library, optical disks and a staging disk subsystem consisting of magnetic disks. We refer to the magnetic disk subsystem as secondary storage, and the tape library and optical disks as tertiary storage. We assume that all the requests are serviced via the staging disk. Considering the biased nature of the user access pattern over the available video files, it is possible that a large number of viewers access a small number of very popular movies, while a large portion of

0-8186-7469-5/96 $05.00 © 1996 IEEE

the videos in the library are viewed much less often[12].

1.2. Relation to previous work

Recent years have seen a number of literatures of the file system related issues in VOD server design[1, 3, 11]. There have been a number of proposals for cost models of the hierarchical storage configuration[8, 6] and data transfer mechanism between the tertiary and secondary storage[7]. The performance of the server is affected by the video file placement mechanism on the disk array[2, 10].Won et.al.[13] examined the use of a large scale *VOD* server with hierarchical storage in a distributed movie delivery environment.

1.3. Contribution

We analyze the relationship between secondary and tertiary storage subsystems, and develop techniques to minimize *Blocking Probability*, which is the percentage of service requests blocked at the VOD server, i.e. cannot be serviced immediately. We provide an analytical formulation for the *blocking probability* - P(Block). *Blocking probability* is dependent upon the request arrival rate as well as other factors that determine the characteristics of the VOD server. Our analysis reveals an interesting fact, namely that increasing the data transfer rate from the staging disks beyond the threshold may degrade the overall server performance. To achieve a shorter minimum response time, it may be required to increase the data transfer rate from the staging disks faster, which may increase the average response time and worsen the server performance. To resolve the conflict between minimum and average repose time, we propose an *Adaptive Enforced Duration* policy and a *Fixed Enforced Duration* policy. We show that under these two policies, the VOD server can achieve better performance.

In section 2 we present the VOD environment being analyzed and describe its characteristics. In section 3 and section 4, we present a detailed model and its analysis of the system. In section 5, we discuss policies for managing the video files on the staging disks. In section 6, we discuss the results of a simulation-based evaluation of the policies proposed in this work.

2. Environment

2.1. Video Files Characteristics

There are two attributes that determines the characteristics of a video file for our purposes, namely access frequency and size. Access frequency represents the relative popularity of a video file. The access frequency of video file m_i is P_i, where $\sum P_i = 1$. Without loss of generality, we assume that video files are indexed by decreasing order of access frequency, i.e. $i \geq j \Leftrightarrow P_i \leq P_j$. The size of a video file determines its service time on each subsystem of the server, i.e. tape drive and staging disks. Let the size of the video file m_i be s_i bytes, and service rates of a tape drive and staging disks be μ_T Bytes/sec and μ_D Bytes/sec, respectively. Then, it takes $\frac{s_i}{\mu_T}$ sec to load m_i from tape drive to staging disks, and $\frac{s_i}{\mu_D}$ sec to transfer m_i from staging disks onto the network assuming the source of the transfer operation is the slower than its destination.

In determining the configuration of the storage hierarchy, we have to consider the access pattern of the available video files. The access pattern can be skewed towards a small set of hot data , or evenly distributed over the set of available video files There are several functions that can be used to approximate the access pattern. Among them are *zipf* distribution[4] and normalized geometric distribution[6].As described in [4], *zipf* distribution with $\alpha = 0.271$ closely approximates the commercial video rental patterns.

2.2. Storage Organization

As shown in Fig. 1, the proposed *VOD* server consists of a tape library, a robot arm, tape drives, and staging disks. Tape subsystem, which consists of a tape library, robot arm, and tape drives, forms the tertiary storage. The robot arm selects the cartridge for the requested video file from the tape library and puts it in an available tape drive. The tape drive transfers the file to the staging disks. Staging disks consist of an array of disks that makes up the secondary storage. Video files are streamed out from the staging disks to the network to fulfill user requests. We assume that staging disks are space bounded rather than bandwidth bounded [1]. By using *pipelining*, it is possible to overlap the tape drive service time and staging disks service time[7]. The tape drive service time is *initial startup time + pipelining transfer time.* For the staging disks to start transfer, a certain amount of the file i - $Init_i$, has to be available on the staging disks to avoid blocking on the tape drive. This is done at *initial start up* time. *Initial Start Up Time* is *time for robot arm movement* + $\frac{Init_i}{\mu_T}$. The entire service consists of *Initial Start Up Time* on the tape subsystem and *staging disks service* on the staging disks, which is disjoint

[1] Whether a disk subsystem is space bounded or bandwidth bounded are dual conditions, i.e. one can be transformed into the other problem by striping or replication[5].

with each other on the time line in Fig. 2. Tape drive service time L_T in later part of this paper implies the *Initial Start Up Time.*

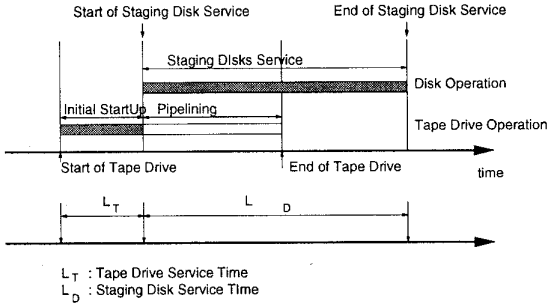

Figure 2. Data Transfer Procedure

3. System Model

Table 1 lists the parameters that we use to characterize the VOD server. We assume that all the video files are of the same length and size. This implies that all video files have the same tape drive service time and same disk service time. Fig. 3 illustrates the organization of hierarchical storage at the VOD server in terms of a queuing model. There are n tape drives in the server. The staging disks subsystem can hold upto m video files. There are four possibilities for an incoming request.

- **Case 1**: The requested video file is already present on the staging disks. The server starts servicing the request immediately from the staging disks. This happens with probability $P(D)$.
- **Case 2**: The requested video file is currently in transition from the tape library to the staging disks. The request is delayed until the video file is available on the staging disks. This happens with probability $P(T)$.
- **Case 3**: Request triggers the robot arm to load the required tape to a tape drive; assumes an empty tape drive exists and sufficient disk space exists. This happens with probability $P(E)$.
- **Case 4**: Request is put into queue when the server cannot service the incoming request immediately due to the system being *full - Blocking.* The system is *full* either when there is no free tape drive or there isn't sufficient space on the disk drives. This happens with probability $1-P(E)-P(D)-P(T)$.

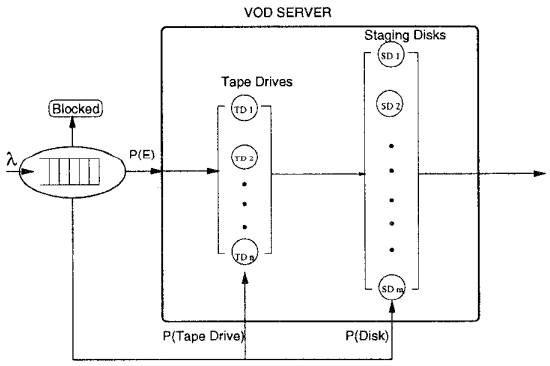

Figure 3. Tape Drives and Staging Disks

λ	arrival process of video file request : Poisson(λ) : movies/min
P_i	Popularity of the video file i.
λ_i	arrival rate of the request for video file i. $\lambda_i = \lambda P_i$.
$Init_i$	The portion of the file that has to be loaded before to starting service from the staging disks
M	total number of video files in the tape library, i.e. number of video files available to user
N_T	Number of Tape Drives in the system
L_T	Latency time at the tape drives, i.e. *Initial Start Up Time*
μ_T	Tape Drive service rate : number of video files per min
L_D	$\frac{\text{file size}}{\mu_D}$, i.e.time to transfer a video file from the staging disks
μ_D	Rate at which the staging disks transfers the video file. : number of files per min
C_D	Capacity of the staging disks subsystem in number of movies.
$E(W_i)$	Expected Duration of video file i on the staging disks
r_i	Request for video file i

Table 1. Variables

Expected Service Time: Using $P(D), P(T)$ and $P(E)$, we can calculate the expected service time for a request. Let S be the total service time for a request, and $E(S)$ be the expected value of S. Then $E(S)$ can be expressed as follows :

$$E(S) = L_D P(D) + (\frac{L_T}{2} + L_D)P(T) + (L_D + L_T)P(E) \quad (1)$$

As can be seen in Eq. (1), the speed of the staging disks is one of the variables that affects the mean service time and *P(Block)* on the VOD server.

4. Blocking in the VOD server

The VOD server is full if either of the two conditions holds - *Tape_Drives_Full* or *Staging_Disks_Full*. Furthermore, these two conditions are mutually exclusive from definition. The requests arriving when the VOD server is full are blocked.

$$P(Block) = P(Tape\_Drives\_Full) + P(Staging\_Disks\_Full) \quad (2)$$

The condition *Tape_Drives_Full* is true when all the tape drives have video cartridges in them and are transferring video files. The condition *Staging_Disks_Full* is true when there is no space left on the staging disks for a new video file. A tape drive is allowed to start loading only when there is sufficient available space to store an entire file on the staging disks. Tape drive reserves space for the video file on the staging disks before it starts transfer. Hence, the number of video files on the staging disks plus the number of video files being loaded from tape drives should be less than C_D. It is possible that a tape drive starts loading when there is a space for $Init_i$ on the staging disks, which is less than the size of a video file. It can lead the staging disks into a *deadlock* if two or more tape drives start transfer simultaneously, hoping for more space subsequently. The server can avoid *deadlock* if it continuously overwrites the data being transfered to the allocated disk space. Then, the subsequent requests for the same video file cannot share the *active* stream on the staging disks since the beginning part of the file is lost already, which can invalidate much of the advantage of using staging disks.

4.1. Effect of L_D on E(S)

In this section, we provide a detailed analytical model of the simplest possible configuration, i.e. $M = 1, N_T = 1, C_D = 1$ to visualize the effect of the data rate on the staging disks. It can be said that tape drive is not needed since there is one disk and only one video file. This becomes invalid if M becomes greater than C_D. The mean service time of the request is proportional to the probability of the server being busy[9]. Hence, we show the effect of $L_D(or \mu_D)$ on $E(S)$ to materialize the effect of $L_D(or \mu_D)$ on *P(Block)*. Consider the service time for the arriving request at each of the three states, i.e. *Empty, Tape_Drives_Busy* and *Staging_Disks_Busy*. It is important to note that there is no blocking in this system since there is only one video file available. To visualize the relationship between $E(S)$ and L_D, we formulate $E(S)$ in terms of L_T, L_D and λ. In Eq. (1), we have provided the formula for $E(S)$, which contains $P(E), P(D)$ and $P(T)$. We interpret $P(E), P(D)$ and $P(T)$ as a function of L_T, L_D and λ. The server is in one of the states *Empty*, *Tape_Drive_Busy*, or *Staging_Disk_Busy*, and repeats cyclic transition among these states. The expected length of each state determines the probability that the VOD server is in its respective state. Length of *Tape_Drive_Busy* is L_T and the length of *Empty* is $\frac{1}{\lambda}$. Let L_{D_0} be the expected duration of the video file on the staging disk. In the given configuration the expected duration of the video file is equivalent to the Length of *Staging Disks Busy* state, which is a function of λ and L_D. Then $P(E), P(T)$ and $P(D)$ can be formulated as follows:

$$P(E) = \frac{\frac{1}{\lambda}}{L_T + L_{D_0} + \frac{1}{\lambda}} \quad (3)$$

$$P(T) = \frac{L_T}{L_T + L_{D_0} + \frac{1}{\lambda}} \quad (4)$$

$$P(D) = \frac{L_{D_0}}{L_T + L_{D_0} + \frac{1}{\lambda}} \quad (5)$$

From Eq. (1), (3), (4), and (5), we can obtain the formula for $E(S)$ which is a function of L_T, L_D and λ. Fig. 4 illustrates $E(S)$ as a function of μ_D and L_D and shows how E(S) changes as the disk subsystem becomes faster. Firstly, E(S) decreases fast. Then it reaches the minimal point. E(S) increases slowly and limits to L_T. Hence, it is not necessary to make the staging disks faster beyond a certain threshold value. It is important to note that *by properly adjusting μ_D (or L_D) $E(S)$ can be less than L_T*. L_{D_0} is equivalnet to $E(W)$ in Eq. (8). The detailed steps of the analysis are given in Won et.al.[14].

4.2. Analytical Modeling

All the possible states and the respective lengths of the time interval have to be taken into account to precisely model *P(Block)* of the server, and to compute the optimal value of L_D. In Eq. (1), there are only three states to consider - *Empty, Tape Drive Busy* and *Staging Disks Busy*. In the general case, the different combinations of video files on the tape drives and staging disks corresponds to the different states of the VOD server. Consequently, the number of states of the VOD server grows exponentially with M and C_D. It is computationally very expensive to obtain the optimal

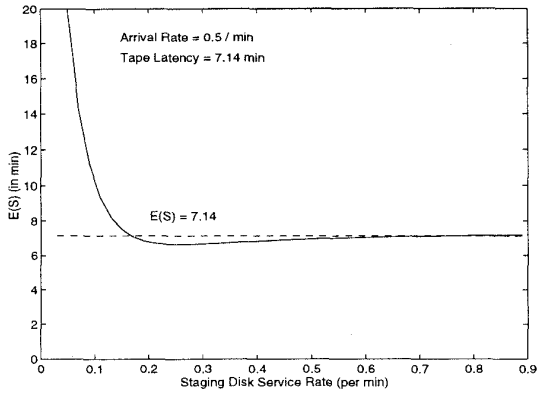

Figure 4. E(S) vs. μ_D

L_D. However, it is essential to investigate the analytical model of *P(Block)* to capture the characteristic behavior of the VOD server based on various factors given in Table 1.

P(Tape_Drives_Full)

Tape_Drives_Full happens when both of the following conditions are met : (1) there are N_T or more different requests during L_T time period, and (2) The requested video file is not in the tape drive nor on the staging disks. It means that for each request during L_T, the interval between its previous request for the same video file and itself is greater than L_D+L_T. Once a request initiates a tape drive's activity, all the subsequent requests for the same video arriving during next L_T are thought of as one from the view point of the tape drive. Let S be the set of video files requested during L_T and $|S| = s$. Let M_s be the collection of the subsets of size s from the available video file. The size of M_s is $\binom{M}{s}$. *P(s different requests arrive during L_T)* can be formulated as follows:

$$P_s = \sum_{S \in M_s} \prod_{i \in S} \left(1 - e^{\lambda_i L_T}\right) e^{-\lambda_i (L_T + L_D)} \qquad (6)$$

From Eq. (6), we can obtain $P_1, P_2, \ldots$. Hence, *P(Tape_Drives_Full)* is

$$P(\text{Tape\_Drives\_Full}) = 1 - e^{-\lambda L_T} - \sum_{s=1}^{N_T - 1} P_s \qquad (7)$$

where $e^{-\frac{\lambda}{\mu}}$ is the probability that there is no arrivals at all.

P(Staging_Disks_Full)

The video file repeats the transition between two states from the aspect of the staging disks - *on the disks* and *off the disks*. The expected length of these two states are important parameter to model the behavior of the staging disks. Once a video file is loaded on the staging disks - *on the disks*, any subsequent requests for it can share the file on the staging disks. The video file remains *active* as long as there is at least one ongoing service from it. The length of the period during which the video file is active on the staging disks is called *duration*. The *duration* of file i is a function of L_D and λp_i. Since the staging disks can hold upto C_D files, staging disks can be interpreted as C_D servers and the duration of the video file is equivalent to the job service time on the server. With this approach, we can approximate the system using a queuing model. Let $E(W)$ be the expected duration of the video file on the staging disks. $E(W)$ of the *active* video stream for the video file with arrival rate λ is as follows:

$$E(W) = \frac{1 - e^{-\lambda L_D}}{\lambda e^{-\lambda L_D}} \qquad (8)$$

. Substituting λ with λp_i in Eq. (8), we obtain the expected duration of a video file i, i.e. $E(W_i)$. Let $\rho_i = r_i^1 r_i^2 r_i^3 \cdots$ be the stream of requests for file i arriving at the staging disks. We can generate the substream of ρ_i,i.e. ρ_i' such that each r_i^k in ρ_i' is the request that initiates the loading of file i on the staging disks. The expected interval between the requests in ρ_i' - $E(W_i')$, can be formulated as follows :

$$E(W_i') = E(W_i) + \frac{1}{\lambda p_i} + L_T \qquad (9)$$

Derivation of Eqs. (8) and (9) is described in [14]. Considering the requests in ρ_i', the arrival rate of request for file i is $\frac{1}{E(W_i')}$, which is equivalent to cartridge loading rate to the tape drives. There are M video files in total. The rate $\overline{\lambda}$ at which the video files are loaded to the staging disks can be formulated as followed.

$$\overline{\lambda} = \sum_{i=1}^{M} \frac{1}{E(W_i')} \qquad (10)$$

Hence, the staging disks can be approximated by a queuing system where the arrival rate is $\overline{\lambda}$, mean service time is $E(W)$, and the number of servers is C_D.

In our model for *P(Staging_Disks_Full)*, we assume that the file loading process from the tape drives to the staging disks is *Poisson*, i.e. the inter-arrival time of the output process from the tape drives follows the exponential distribution. Theoretically this process is not Poisson. However, according to our investigation, the process is sufficiently close to the Poisson process[14]. Let $E(W)$ be the mean service time, i.e. $E(W) = \sum_{i=1}^{M} p_i E(W_i)$. By λ', C_D, and $E(W)$ we can formulate P(Staging_Disks_Full). In general, each video files may have a different access frequency.

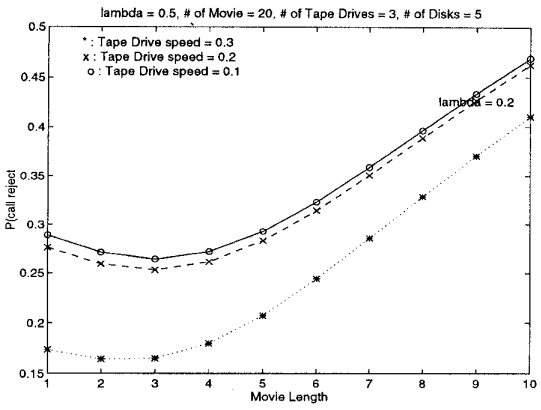

Figure 5. P(Block) vs. L_D

Hence, there are $\sum_{k=1}^{C_D} \binom{M}{k}$ possible different states for the staging disks. It is not trivial to obtain a analytical solution for staging disks blocking probability since the number of states grows exponentially with M and C_D. We omit the formulation for P(Staging_Disks_Full) for the general access pattern in this work. When the access pattern on the video file is not biased, i.e. *uniform*, the state of the VOD server depends only on the number of files rather than on the combinations of the files. Hence, the analytical formulation becomes simpler and is computationally inexpensive. Fig. 5 is the plotted graph of our formulation, which confirms the result in Fig. 4.

5. Space Management on Staging Disks

5.1. *Use and Free* Policy

The analytical examination assumes that the disk space used by the video file is returned to the free space pool when it completes its service, which we call *Use and Free*. This policy is useful when the staging disks are shared by several different applications at the server. Under *Use and Free* policy, determination of L_D is crucial to maximize the server performance, and to avoid waste of system resources, as shown in Fig. 5. Unfortunately, it is not trivial to obtain the optimal value of L_D analytically. Let us describe the relationship between *P(Block)* and $\mu_D (or L_D)$. Let r_i be the request for video file i and t_i be the inter-arrival between consecutive r_i's.

$$\begin{aligned} & P(\text{When } r_i \text{ arrives, file } i \text{ is on the disk}) \\ = \; & P(t_i < L_D) \\ = \; & 1 - e^{-\lambda_i L_D} \end{aligned} \tag{11}$$

If L_D becomes shorter, Eq. (11) gets smaller and hence tape drives becomes busier. This causes an increase in *P(Tape_Drive_Full)*, and consequently increases *P(Block)*. If L_D becomes longer, staging disks holds video file i for a longer period of time, and thus an incoming request for a file not on the staging disk may be blocked due to space limitation. This results in the increase of $P(Staging\_Disks\_Full)$ and consequently increase in $P(Block)$.

5.2. Enforced Duration Policy

To further reduce $P(Block)$ at the optimal data rate configuration under *Use and Return* policy, we propose that the server maintain a video file on the staging disks even though it is not *active* video - *Enforced Duration Policy*. In *ED* policy, server keeps a video file on the staging disks at least for a predefined time, i.e. *life time*, once it is loaded to the staging disks. The server keeps the file on the staging disks although it is *inactive*. By adopting *ED* policy, the server can overcome the anomaly existing between *P(Block)* and L_D In determining the *life time* of the video file we propose two policy.

Fix Enforced Duration (FED) Policy : All the files have the same *life time* independent of their access frequencies.

Adaptive Enforced Duration (AED) Policy : Since video files have different access frequencies, it is more reasonable to determine the *life time* based on the popularity of the video files. We use the *expected duration* in Eq. (8) to determine the life time for the video file. In *AED* policy, the *life time* of the video file is proportional to the expected duration. The access frequencies of the video files are based on the zipf distribution.

6. Performance Evaluation

We performed extensive simulations for each of the policies to examine the validity of our model for *P(Block)*. There are 100 video files in total. Unless noted, the access frequency follows the *zipf* distribution with $\alpha = 0.271$. N_T is 3 and C_D is 8.

6.1. *Use and Free* policy

Fig. 6 illustrates the effect of μ_D on $P(Block)$. X-axis is the transfer rate of the staging disks, μ_D, and the y-axis is the blocking probability of the VOD server, *P(Block)*. The simulation result of Fig. 6 confirms our modeling of $P(Block)$ on VOD server in section 4: *If the data transfer rate grows beyond a certain threshold*

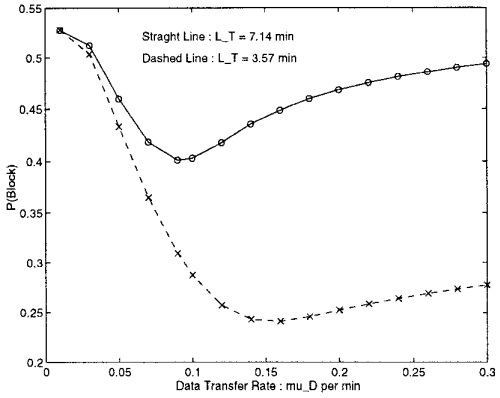

Figure 6. μ_D and P(Block)

value, P(Block) first drops and then gradually increase towards a certain limiting value.

In Fig. 6, λ is 1.0 per minute. As length of *active* duration becomes longer, the incoming requests for the same file are more likely to be serviced from the staging disks. To increase the *active duration of a video file*, L_D has to be increased and consequently μ_D has to be decreased. Hence, decreasing μ_D can contribute to reduce *P(Block)*. As the duration becomes shorter, larger portion of the incoming requests require the video file to be loaded from the tape drives.

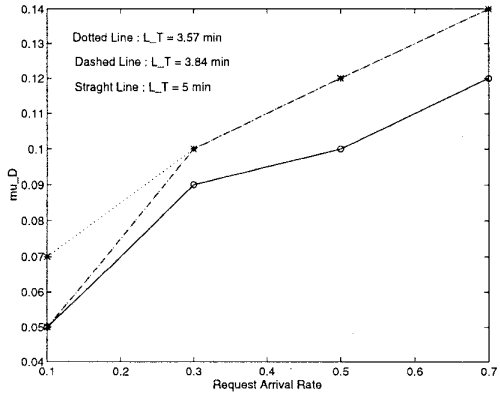

Figure 7. Optimal Staging Disks Transfer rate

The optimal value of μ_D depends on various system parameter such as λ, μ_T or N_T. Fig. 7 illustrates the relationship between the optimal value of μ_D and λ. The optimal μ_D is linearly proportional to λ,i.e. as when there are more requests to the VOD server, μ_D has to be increased to obtain the minimum *P(Block)*. Fig. 8 illustrates the relationship between λ and the minimum *P(Block)* at the optimal μ_D. As in Fig. 8,

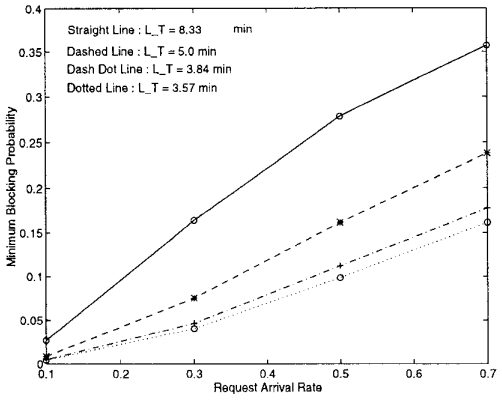

Figure 8. Request Arrival Rate vs. Minimum Blocking Probability

the minimum *P(Block)* is linearly proportional to the arrival rate λ. We thus conclude that the server performance in terms of *P(Block)* is linear function of λ.

6.2. *Enforced Duration*(ED) policy

In *ED* policy, a video file is forced to stay on the staging disks until its *life time* expires. There are two approaches to determine the *life time* for the video file on the staging disks: *Fixed Enforced Duration* Policy and *Adaptive Enforced Duration* Policy. In *FED* policy, all video files are assigned the same *life time* which is a multiple of L_D,i.e. $knob * L_D$ and $knob = 1, 2, ...$ In *AED* each video file is assigned a *life time* which is multiples of it's expected duration on the staging disks - $knob * E(W_i)$.

Our simulation results show that *AED* policy outperforms the *FED* policy. In Fig. 9, λ is 1.0, L_T is 7.14 and μ_D is 0.09, which is the same as in Fig. 6. In Fig. 6, minimum *P(Block)* is about 0.4 with *Use and Return* policy. We select μ_D as 0.9 since it is the optimal value under given VOD server configuration in Fig. 6 i.e., λ is 1.0 and L_T is 7.14 min. In Fig. 10, we visualize the effect of the *access skewness* on the *P(Block)* under *AED* policy. It illustrates *P(Block)* under various values of α in *zipf* distribution. In both the experiments, λ is 1.0, L_T is 7.14 min and μ_D is 0.09. As α gets smaller, the access pattern is more skewed for the small set of the available video files. The graph of *P(Block)* shifts down along the y-axis as α becomes smaller. It means that as the incoming request pattern is more skewed, it is more efficient to do caching in staging disks, using a hierarchical storage based VOD server.

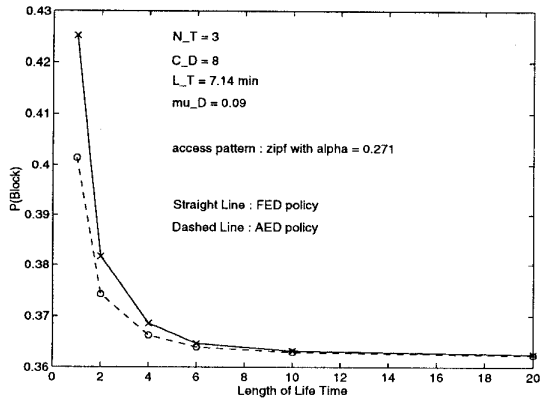

Figure 9. length of *life time* and P(Block)

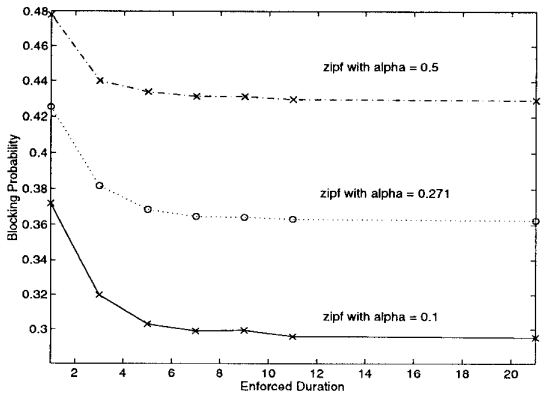

Figure 10. P(Block) vs. access pattern under *AED* policy

7. Conclusion

In this work, we modeled a VOD server analytically, and investigated the impact of data rate of the staging disks on the overall server performance. We found that a data rate beyond certain threshold value does not help to reduce the $P(Block)$. The optimal transfer rate of the staging disks is a function of request arrival rate λ, tape drive speed μ_T, and the access pattern. Unfortunately, it is very difficult to obtain the optimal transfer rate analytically.We provide a formulation for it and visualize the various relationship between the parameters and *P(Block)*. By caching a video file on the staging disks for a given period of time, *life time*, we can further reduce $P(Block)$ under optimal configuration of the server. The analytical modeling of the server is very challenging task due to the complexity of the server states. The fact that the stream can be shared by several requests on the staging disks enhances the complexity of the task. Our current analytical model for *P(Block)* requires further refinement to evaluate the other various disk space management polices.

References

[1] D. P. Anderson, Y. Osawa, and R. Govindan. A File System for Continuous Media. *ACM Trans. Comput. Syst.*, 10(4):311–337, Nov 1992.

[2] M. Chen, D. Kandlur, and P. S. Yu. Storage and retrieval methods to support fully interactive playout in a disk-array-based video server. *Multimedia Systems*, 3(3):126–135, July 1995.

[3] M.-S. Chen, D. D. Kandlur, and P. S. Yu. Optimization of the grouped sweeping scheduling(gss) with heterogeneous multimedia streams. In *ACM Multimedia '93*, pages 235 – 242, 1993.

[4] A. Dan and D. Sitaram. Buffer Management Policy for an On-Demand Video Server. Technical Report IBM Research Report RC 19347, IBM Research Division, T.J. Watson Research Center, Yorktown Heights, NY 10598, 1993.

[5] A. Dan and D. Sitaram. An online video placement policy based on bandwidth to space ratio(bsr). In *Proceedings of ACM SIGMOD*, pages 376 – 385, 1995.

[6] Y. Doganata and A. Tantawi. Making a cost-effective video server. *IEEE Multimedia*, pages 22–30, Winter 1994.

[7] Ghandeharizadeh, S. and Shahabi, Cyrus. On Multimedia Repositories, Personal Computers, and Hierarchical Storage. In *Conf. Proc. ACM Multimedia*, pages 407–416, 1994.

[8] M. G. Kienzle, A. Dan, D. Sitaram, and W. Tetzlaff. Using tertiary storage in video-on-demand servers. In *Proc. of IEEE COMPCON '95*, 1995.

[9] L. Kleinrock. *Queueing Systems.* John Wiley & Sons, Inc., 1975.

[10] T. Little and D. Vankatesh. Popularity-Based Assignment of Movies to Storage Devices in a Video-on-Demand System. *Multimedia Systems*, 2(6):280–287, January 1995.

[11] P. Rangan, H. Vin, and S. Ramanathan. Designing an on-demand multimedia service. *IEEE Communication Magazine*, 30(7):56–65, July 1992.

[12] Video Store Magazine, December 13 1992.

[13] Y. Won and J. Srivastava. Scheduling video delivery in a distributed environment. In I. Miloucheva, editor, *Proc. of Second Workshop on Protocols for Multimedia Systems*, pages 119–135, Salzberg, Austria, Oct. 1995.

[14] Y. Won and J. Srivastava. Design and analysis of hierarchical based storage vod server. Technical report, Dept. of Computer Science, University of Minnesota, Minneapolis, 1996.