Performance Analysis of Resource Selection Schemes for a Large Scale Video-on-Demand System

Jun Guo, *Member, IEEE*, Eric W. M. Wong, *Senior Member, IEEE*, Sammy Chan, *Member, IEEE*, Peter Taylor, Moshe Zukerman, *Fellow, IEEE*, and Kit-Sang Tang, *Member, IEEE*

***Abstract*—The designers of a large scale video-on-demand system face an optimization problem of deciding how to assign movies to multiple disks (servers) such that the request blocking probability is minimized subject to capacity constraints. To solve this problem, it is essential to develop scalable and accurate analytical means to evaluate the blocking performance of the system for a given file assignment. The performance analysis is made more complicated by the fact that the request blocking probability depends also on how disks are selected to serve user requests for multicopy movies. In this paper, we analyze several efficient resource selection schemes. Numerical results demonstrate that our analysis is scalable and sufficiently accurate to support the task of file assignment optimization in such a system.**

***Index Terms*—Blocking probability, fixed-point approximation, resource selection, video-on-demand.**

I. Introduction

Research test-beds of video-on-demand (VOD) services [1] have been prevalent for many years [2]. There have been several service providers who have attempted to capitalize on this commercial opportunity. To compete with the traditional video rental business, it is important for a VOD system to provide a large population of end users with pleasurable on-demand access to a large variety of movie content coupled with full VCR-like interactive capabilities [3]. Due to the large storage space and I/O bandwidth required in storing and delivering movie contents, a large-scale VOD system needs to manage a large cluster of on-line disks to store the large number of movies.

Given the significant asymmetry in access demand and file-size of different movies, and due to the limitation of striping techniques [4], a certain number of movies with high access demand or of small file-size are usually replicated over multiple disks, or logical disks [4] if striping is used in the VOD system. In this way, stream resources from multiple disks (servers) can be managed to serve user requests for popular movies, while the disk storage space is more efficiently utilized. Since a user request for a movie can only be connected to a disk where a file-copy of the movie is stored, the request blocking probability essentially depends on how disks are selected by the VOD scheduler to serve user requests for multicopy movies [5] and how movies are assigned to disks [6].

The designers of a large scale VOD system thus face an optimization problem of deciding how to assign movies to multiple disks so that the request blocking probability is minimized subject to capacity constraints. To solve this problem, it is essential to develop scalable and accurate analytical means to evaluate the request blocking probability of a given file assignment.

A. Related Work

For the sake of a manageable performance analysis, Little and Venkatesh [6] considered what we call a *single random trial* (SRT) resource selection scheme. Following SRT, when a user request for a multicopy movie arrives, the VOD scheduler randomly selects one of the disks storing a file-copy of the requested movie. If the disk is fully busy (i.e., the stream capacity of the disk is used up), the request is simply blocked without further attempting any other disk that keeps a file-copy of the requested movie. SRT is a very simple scheme for which exact blocking probability results are easily obtained. Specifically, with the assumption of Poisson request arrivals (indicated by M), general distribution of channel holding times (identified by G), and letting k denote the disk stream capacity, each disk is conveniently modelled as an $M/G/k/k$ queueing system using Kendall's notation [7].

SRT is inherently inefficient in utilizing system resources given the existence of multicopy movies. Two efficient resource selection schemes of reasonable operating cost were studied by simulation in [5]. In both schemes, a user request is blocked if and only if all disks (in an exhaustive sense) storing the requested movie file are found to be fully busy. The *repeated random trials* (RRT) scheme is a natural extension of SRT where the VOD scheduler continues with repeated random trials until all the disks are attempted. If it is feasible to monitor channel occupancies, a more efficient *least busy fit* (LBF) scheme can be implemented so that a user request for a multicopy movie is always directed to the least busy disk where a file-copy of the requested movie is placed.

Without much concern for cost and complexity of real-time stream scheduling in a large scale VOD system, Tsao *et al.* [8] considered a stream repacking scheme which reschedules the currently playing streams among disks along a migration path, if this allows an extra user request (potentially blocked by LBF) to be accepted. Variations of the stream repacking scheme were also proposed to be initiated at earlier stages to achieve dynamic load balancing [9], [10]. Although in theory the disk resources can be more efficiently utilized by means of stream repacking, it is not clear how realistically these complicated schemes can be implemented in a large scale VOD system.

B. Our Contribution

In this paper, we shall demonstrate by numerical results that a file assignment that achieves a smaller blocking probability in the SRT system does not necessarily lead to a smaller blocking probability in systems where exhaustive resource selection schemes are used. Accordingly, an optimal file assignment solution established in the SRT system is likely to under-utilize system resources if we shall test it in systems where exhaustive resource selection schemes are used. This justifies the importance of our analysis in this paper to support file assignment optimization in a more realistic VOD system using these more efficient resource selection schemes.

Manuscript received March 6, 2006; revised July 18, 2007. This paper was presented in part at the 18th International Teletraffic Congress, Berlin, Germany, September 2003. This work was supported in part by the City University of Hong Kong under Projects 7001224 and 7001458 and in part by the Australian Research Council (ARC). Part of the work presented in this paper was done while J. Guo and M. Zukerman were visiting the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, China. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Simon Lucey.

J. Guo is with the School of Computer Science and Engineering, The University of New South Wales, Kensington, Sydney, NSW 2052, Australia (e-mail: jguo@cse.unsw.edu.au).

E. W. M. Wong, S. Chan, and K. S. Tang are with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong SAR, China (e-mail: eeewong@cityu.edu.hk; eeschan@cityu.edu.hk; eekstang@cityu.edu.hk).

P. Taylor is with CUBIN, the ARC Centre of Excellence for the Mathematics and Statistics of Complex Systems (MASCOS), and the Department of Mathematics and Statistics, The University of Melbourne, Melbourne, VIC, Australia (e-mail: p.taylor@ms.unimelb.edu.au).

M. Zukerman is with the ARC Special Research Centre for Ultra-Broadband Information Networks (CUBIN), Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, VIC, Australia (e-mail: m.zukerman@ee.unimelb.edu.au).

Digital Object Identifier 10.1109/TMM.2007.911281

The major contribution of this paper is to provide an analytical framework for analysis of a large scale VOD system using exhaustive resource selection schemes. We derive scalable and accurate analytical formulas to evaluate the request blocking probability of a given file assignment. Considering the significant performance gain obtainable due to file assignment optimization [6], our analysis in this paper is key to the design and dimensioning of a large scale VOD system. Specifically, the analytical results must be accurate enough to differentiate the quality of various file assignments in terms of request blocking probability. In addition, they must be fast enough to be embedded in the file assignment optimization module to find out the optimal or near-optimal blocking probability of the system for a given configuration of disks [11], [12]. The optimization module can then be embedded in the dimensioning process to find out the minimum number of disks needed for the system to meet the grade of service requirement.

With the complicated interactions between user requests for multicopy movies and selections of disk resources to serve these requests, the entire system is modelled as a multidimensional Markov chain with the dimension equal to the number of disks. Although this chain can be solved theoretically, brute force solutions to the state equations are computationally infeasible for a large scale VOD system due to the "curse of dimensionality." We investigate in this paper if the fixed-point approximation (FPA) method [13], [14] can be applied to a large scale VOD system and if the approximation has reasonable accuracy and computational efficiency.

The use of FPA has been widespread in the literature of telecommunications modeling [15], [16]. Specifically, it has been applied to the analysis of large circuit-switched networks with random alternate routing and with state dependent routing (see, e.g., [17]–[20]). We shall see that in the context of a VOD system with the existence of multicopy movies, FPA works by decoupling the whole system of J disks into J independent subsystems and treating each such subsystem as an $M/G/k/k$ queueing system. This is done by making the assumptions that 1) the occupancy processes at the different disks are statistically independent and 2) the request arrival process seen by each disk is Poisson. Since the blocking probability of an $M/G/k/k$ system is insensitive to the distribution of channel holding times [21] and depends only on its mean, each disk can then be conveniently modelled as an $M/M/k/k$ queueing system with exponentially distributed service times.

An additional contribution of this paper is to propose an efficient variation, namely *global random trial* (GRT), of RRT. The implementation of GRT requires that the VOD scheduler collect the information of "fully busy" or "not fully busy" from a disk at each time epoch when the disk changes between the two states. Taking advantage of such state information, GRT handles the request for a multicopy movie by randomly selecting one disk from those that store a file-copy of the requested movie and that are not fully busy. Clearly, GRT and RRT are equivalent in terms of blocking probabilities since they both block the request if and only if all the servers are busy. Nevertheless, GRT is more efficient in a sense that it avoids multiple redundant attempts that may be made by RRT upon the arrival of the request, which would otherwise introduce some additional setup delay for processing the request. In this paper, we prove that the FPA solutions for RRT and GRT are equivalent, so that we can use either of the two analytical solutions in practice, as none of them is more accurate in approximating the exact blocking performance of the VOD system.

C. Organization

In Section II, we describe the system model. In Section III, we present FPA solutions for RRT, GRT and LBF. We prove that the FPA solutions of RRT and GRT are equivalent. In Section IV, we give numerical results to demonstrate the sufficient accuracy of the approximate solutions in supporting the task of file assignment optimization for the system. Finally, we provide concluding remarks in Section V.

TABLE I
SUMMARY OF MAJOR SYMBOLS

Symbol	Definition
$\mathcal{D}$	Set of disks in the system
J	Number of disks in the system
$\mathcal{F}$	Set of distinct movies in the system
M	Number of distinct movies in the system
C_j	Storage space of disk j
N_j	Stream capacity of disk j
L_m	File-size of movie m
n_m	Number of file-copies of movie m
Ω_m	Set of disks storing a file-copy of movie m
Φ_j	Set of distinct movies placed on disk j
λ	Total request arrival rate
p_m	Popularity of movie m
$1/\mu_m$	Mean channel holding time for movie m

II. SYSTEM MODEL

For the reader's convenience, we provide in Table I a list of major symbols that we shall define and use in this paper.

We consider a large scale VOD system with a set $\mathcal{D}$ of J disks, labelled $1, 2, \ldots, J$, and a set $\mathcal{F}$ of M distinct movies, marked $1, 2, \ldots, M$. The storage space of disk j is C_j units. The file-size of movie m is L_m units. (For example, one unit of storage space or file-size could be 1 GB.) For all $j \in \mathcal{D}$, we assume that $\max_{m \in \mathcal{F}} L_m \ll C_j$, so that each disk can store a number of movie files. We also assume $\sum_{m \in \mathcal{F}} L_m < \sum_{j \in \mathcal{D}} C_j$, so that the system has spare disk storage space to place multiple file-copies for certain movies in $\mathcal{F}$. For a feasible file assignment, movie m has n_m copies, $1 \leq n_m \leq J$. Those copies are allocated on n_m separate disks, which constitute the set Ω_m. The set of movie files placed on disk j is denoted Φ_j, and satisfies the storage space constraint, i.e., $\sum_{m \in \Phi_j} L_m \leq C_j$. If the independent video streams emanating from disk j are considered to be approximately statistically equivalent, disk j may support up to N_j concurrent streams (logical channels). We shall assume that the access link to the J disks has enough bandwidth capacity such that it will not impose further constraint on the number of concurrent video streams that can be supported by the J disks.

In a statistical sense, making a request for a movie in a VOD system is similar to making a call in telephony, where the Poisson assumption is widely accepted. This Poisson assumption was recently justified in [22], where Costa *et al.* observed that inter-arrival times of user requests in streaming multimedia systems are exponentially distributed. We therefore assume that the aggregate arrivals of requests for all movies follow a Poisson process with rate λ requests per time unit. The request arrival processes of different movies are mutually independent Poisson processes. The user holding time of a video stream for movie m is arbitrarily distributed with mean $1/\mu_m$ time units. The demand rate for movie m creates its popularity profile p_m, defined as the relative probability of movie m being requested by a user, and $\sum_{m=1}^{M} p_m = 1$. The request arrival rate of movie m is given by λp_m. We ignore the setup delay in this paper because it has negligible effect on the blocking probability. Notice that relative to the large user holding time of a video stream (e.g., for a 2-h feature-length movie), the setup delay for processing a user request in a VOD system is of the order of milliseconds [23].

III. Performance Analysis

A. Exact Solution of SRT

The analysis of SRT in a system of heterogeneous disks was discussed in [11]. For the purpose of performance comparison with other schemes, we derive it again in this section.

Since the request arrival process of each movie in $\mathcal{F}$ is Poisson, if a request for a multicopy movie m is randomly directed to one of the disks in Ω_m, with no subsequent possible retrials attempted, the request arrival process of movie m is simply decomposed into n_m independent Poisson processes, each of which has rate $y_j(m) = \lambda p_m / n_m$.

For disk $j, j \in \mathcal{D}$, the total request arrival rate due to the superposition of the request arrival processes of all movie files in Φ_j is given by

$$y_j = \sum_{m \in \Phi_j} y_j(m). \tag{1}$$

The mean channel holding time $1/\hat{\mu}_j$ for all movie files in Φ_j is obtained by

$$\frac{1}{\hat{\mu}_j} = \frac{1}{y_j} \sum_{m \in \Phi_j} \frac{y_j(m)}{\mu_m}. \tag{2}$$

Let

$$\vec{\xi'} = \left(\xi_1^{(N_1)}, \xi_2^{(N_2)}, \ldots, \xi_J^{(N_J)}\right) \tag{3}$$

denote the vector of stationary probabilities that disk $j, j \in \mathcal{D}$, is in state N_j, or in other words, it has all N_j logical channels occupied. In the case of SRT, $\xi_j^{(N_j)}$ is simply given by the Erlang B Formula [7]

$$\xi_j^{(N_j)} \overset{\text{def}}{=} E\left(\frac{y_j}{\hat{\mu}_j}, N_j\right) = \frac{\left(\frac{y_j}{\hat{\mu}_j}\right)^{N_j} / N_j!}{\sum_{i=0}^{N_j} \left(\frac{y_j}{\hat{\mu}_j}\right)^i / i!}. \tag{4}$$

For $m \in \mathcal{F}$, the request blocking probability of movie m is calculated by

$$B_m = \frac{1}{n_m} \sum_{j \in \Omega_m} \xi_j^{(N_j)}.$$

We compute the mean request blocking probability of multicopy movies by

$$\tilde{B} = \frac{\sum_{m \in \mathcal{F}, n_m > 1} p_m B_m}{\sum_{m \in \mathcal{F}, n_m > 1} p_m} \tag{5}$$

and the mean request blocking probability of single-copy movies by

$$\hat{B} = \frac{\sum_{m \in \mathcal{F}, n_m = 1} p_m B_m}{\sum_{m \in \mathcal{F}, n_m = 1} p_m} \tag{6}$$

and the mean request blocking probability of all movies from

$$\bar{B} = \sum_{m \in \mathcal{F}} p_m B_m. \tag{7}$$

B. FPA Solution of RRT

If subsequent random trials are repeated until all disks in the set Ω_m are attempted, the request arrival process of a multicopy movie m directed to disk $j, j \in \Omega_m$, is composed of two processes: 1) a Poisson process made of first choice requests to disk j with rate $\lambda p_m / n_m$ and 2) a process made of requests overflowed from disks in Ω_m other than disk j, which is normally non-Poisson.

To ease the bookkeeping of these overflowed requests, we define $\Psi(\Omega_m - \{j\}, x)$ as the set of all possible permutations of arranging x disks out of $\Omega_m - \{j\}$ (all disks in Ω_m except disk j), for $x = 1, 2, \ldots, n_m - 1$. By $\mathcal{S} \in \Psi(\Omega_m - \{j\}, x)$ and $\mathcal{S} = \{s_1, s_2, \ldots, s_x\}$, we say that $\mathcal{S}$ is one such possible permutation, and $s_1, s_2, \ldots, s_x$ are the ordered disks enumerated in $\mathcal{S}$. Assume that a request for any movie m that has been denied at disk j is independent of other requests for movie m that have been denied at other disks in the set Ω_m, the rate of overflowed requests for movie m originally blocked by disk s_1, and subsequently blocked by $s_2, s_3, \ldots, s_x$, and finally offered to disk j is calculated by

$$\frac{\lambda p_m}{n_m} \prod_{i=1}^{x} \frac{\xi_{s_i}^{(N_{s_i})}}{n_m - i}.$$

Taking into account all the possible permutations from $\Psi(\Omega_m - \{j\}, x)$ and all x, the aggregate rate of overflowed requests for movie m from disks in Ω_m except disk j is given by

$$\frac{\lambda p_m}{n_m} \sum_{x=1}^{n_m - 1} \sum_{\mathcal{S} \in \Psi(\Omega_m - \{j\}, x)} \prod_{i=1}^{x} \frac{\xi_{s_i}^{(N_{s_i})}}{n_m - i}.$$

The request arrival rate of disk j due to movie m is therefore given by

$$y_j(m) = \frac{\lambda p_m}{n_m} \left[1 + \sum_{x=1}^{n_m - 1} \sum_{\mathcal{S} \in \Psi(\Omega_m - \{j\}, x)} \prod_{i=1}^{x} \frac{\xi_{s_i}^{(N_{s_i})}}{n_m - i}\right] \tag{8}$$

and the total request arrival rate of disk j due to all movie files in the set Φ_j is given by (1). The mean channel holding time $1/\hat{\mu}_j$ for all movie files in Φ_j is obtained by (2).

FPA works by treating the request arrival process seen by disk j as if it were Poisson, even if it includes a de facto non-Poisson overflow process. Correspondingly, we allow calculating the blocking probability of disk j in the RRT system using (4). For compatibility, we require that the blocking probabilities so calculated be the same as those used to calculate the reduced load in (8). Thus, (1)–(4) and (8) constitute a set of fixed-point equations of the form

$$\vec{\xi'} = e(\vec{\xi'}). \tag{9}$$

These fixed-point equations can often be solved efficiently by the successive substitution method to be presented in Section III-E.

Solving (9) for $\xi_j^{(N_j)}$ of disk j, and using the fact that for a request of movie m to be blocked, it would need to be denied at all n_m disks in Ω_m, we therefore have

$$B_m = \prod_{j \in \Omega_m} \xi_j^{(N_j)}. \tag{10}$$

$\tilde{B}$, $\hat{B}$, and $\bar{B}$ are then computed by (5), (6), and (7), respectively.

C. FPA Solution of GRT

According to this scheme, when a request for a multicopy movie m arrives, the VOD scheduler filters out all disks in the set Ω_m that are fully busy, and randomly selects one of the remaining not fully busy disks to serve the request. The implementation of GRT requires that the VOD scheduler collect the information of "fully busy" or "not fully busy" from a disk at each time epoch when the disk changes between the two states.

Provided that disk j is in Ω_m and has one or more free channels available upon the arrival of the request, it is useful to define $\Upsilon(\Omega_m - \{j\}, x)$ as the set of all possible combinations of choosing x disks out of $\Omega_m - \{j\}$. Note that for RRT, $\Psi(\cdot)$ is a set of permutations, while here $\Upsilon(\cdot)$ is a set of combinations. Then, upon the arrival of the request for movie m, the probability that, among the other $n_m - 1$ disks in Ω_m except disk j, $h-1$ disks also have one or more free channels available, and the remaining $n_m - h$ disks are all fully busy is given by

$$\sum_{\mathcal{S} \in \Upsilon(\Omega_m - \{j\}, h-1)} \prod_{u \in \mathcal{S}} \left(1 - \xi_u^{(N_u)}\right) \prod_{v \in \Omega_m - \{j\} - \mathcal{S}} \xi_v^{(N_v)}.$$

Therefore, the request arrival rate of disk j due to movie m is given by

$$y_j(m) = \lambda p_m \sum_{h=1}^{n_m} \frac{1}{h} \sum_{\mathcal{S}\in\Upsilon(\Omega_m-\{j\},h-1)} \prod_{u\in\mathcal{S}} \left(1-\xi_u^{(N_u)}\right) \cdot \prod_{v\in\Omega_m-\{j\}-\mathcal{S}} \xi_v^{(N_v)} \quad (11)$$

and the total request arrival rate y_j of disk j due to all movie files in the set Φ_j is given by (1). The mean channel holding time $1/\hat{\mu}_j$ for all movie files in Φ_j is obtained by (2). The system of (1)–(4) and (11) comprises a set of fixed-point equations, again of the form (9).

Once $\xi_j^{(N_j)}$ of disk j is obtained by solving (9), by the disk independence assumption and using the fact that for a request of movie m to be blocked, all channels on the n_m disks in Ω_m must be currently occupied, B_m is given by (10). $\tilde{B}$, $\hat{B}$, and $\bar{B}$ are then computed by (5), (6), and (7), respectively.

It is clear that GRT and RRT are equivalent in terms of blocking probabilities since they both block the requests if and only if all the servers are busy. Theorem 1 below establishes the equivalency of GRT and RRT in terms of the reduced load to each disk based on FPA. We first need a combinatorial lemma.

Lemma 1: For integers n and k with $n > k$

$$\sum_{i=0}^{k} \binom{k}{i} \frac{(-1)^i}{n-k+i} = \frac{k!(n-k-1)!}{n!}.$$

The Lemma follows from (1.41) in [24, p. 6].

A closer examination of the right-hand side (RHS) of (8) reveals that the pain of enumerating all possible permutations of arranging x disks out of $\Omega_m - \{j\}$ can be much alleviated by taking advantage of the fact that the product of the probabilities is always the same, regardless of the particular disk order in each of the permutations. The RHS of (8) can thus be expressed as

$$\lambda p_m \left[\frac{1}{n_m} + \sum_{x=1}^{n_m-1} \sum_{\mathcal{S}\in\Upsilon(\Omega_m-\{j\},x)} \frac{x!(n_m-x-1)!}{n_m!} \prod_{u\in\mathcal{S}} \xi_u^{(N_u)}\right]. \quad (12)$$

Now the problem reduces to showing that (12) and the RHS of (11) are the same.

Theorem 1: Let

$$P_A(j) = \frac{1}{n_m} + \sum_{x=1}^{n_m-1} \sum_{\mathcal{S}\in\Upsilon(\Omega_m-\{j\},x)} \frac{x!(n_m-x-1)!}{n_m!} \cdot \prod_{u\in\mathcal{S}} \xi_u^{(N_u)}, \quad (13)$$

and

$$P_B(j) = \sum_{h=1}^{n_m} \frac{1}{h} \sum_{\mathcal{S}\in\Upsilon(\Omega_m-\{j\},h-1)} \prod_{u\in\mathcal{S}} \left(1-\xi_u^{(N_u)}\right) \cdot \prod_{v\in\Omega_m-\{j\}-\mathcal{S}} \xi_v^{(N_v)}. \quad (14)$$

Then $P_A(j) = P_B(j)$.

Proof: To facilitate the proof, we shall rewrite (13) as

$$P_A(j) = \sum_{\mathcal{R}\subseteq\Omega_m-\{j\}} \frac{n(\mathcal{R})!(n_m-n(\mathcal{R})-1)!}{n_m!} \prod_{u\in\mathcal{R}} \xi_u^{(N_u)}$$

and (14) as

$$P_B(j) = \sum_{\mathcal{S}\subseteq\Omega_m-\{j\}} \frac{1}{n_m-n(\mathcal{S})} \cdot \prod_{u\in\mathcal{S}} \xi_u^{(N_u)} \prod_{v\in\Omega_m-\{j\}-\mathcal{S}} \left(1-\xi_v^{(N_v)}\right) \quad (15)$$

where we use $n(\mathcal{R})$ to denote the number of elements in the set $\mathcal{R}$.

Expanding the RHS of (15), we get

$$\begin{aligned} P_B(j) &= \sum_{\mathcal{S}\subseteq\Omega_m-\{j\}} \frac{1}{n_m-n(\mathcal{S})} \cdot \Bigg[\prod_{u\in\mathcal{S}} \xi_u^{(N_u)} - \sum_{v_1\in\Omega_m-\{j\}-\mathcal{S}} \xi_{v_1}^{(N_{v_1})} \prod_{u\in\mathcal{S}} \xi_u^{(N_u)} \\ &\quad + \sum_{v_1\neq v_2\in\Omega_m-\{j\}-\mathcal{S}} \xi_{v_1}^{(N_{v_1})} \xi_{v_2}^{(N_{v_2})} \prod_{u\in\mathcal{S}} \xi_u^{(N_u)} - \dots\Bigg] \\ &= \sum_{\mathcal{S}\subseteq\Omega_m-\{j\}} \frac{1}{n_m-n(\mathcal{S})} \cdot \sum_{\mathcal{S}\subseteq\mathcal{R}\subseteq\Omega_m-\{j\}} (-1)^{n(\mathcal{R})-n(\mathcal{S})} \prod_{u\in\mathcal{R}} \xi_u^{(N_u)} \\ &= \sum_{\mathcal{R}\subseteq\Omega_m-\{j\}} \prod_{u\in\mathcal{R}} \xi_u^{(N_u)} \sum_{\mathcal{S}\subseteq\mathcal{R}} \frac{(-1)^{n(\mathcal{R})-n(\mathcal{S})}}{n_m-n(\mathcal{S})} \\ &= \sum_{\mathcal{R}\subseteq\Omega_m-\{j\}} \prod_{u\in\mathcal{R}} \xi_u^{(N_u)} \sum_{i=0}^{n(\mathcal{R})} \binom{n(\mathcal{R})}{i} \frac{(-1)^i}{n_m-n(\mathcal{R})+i} \\ &= P_A(j) \end{aligned}$$

where we enumerate the subsets $\mathcal{S}$ and make the substitution $i = n(\mathcal{R}) - n(\mathcal{S})$ to get from the third line to the fourth line and the final equivalence follows from the Lemma. ■

Showing that $P_A(j) = P_B(j)$ is not only an interesting mathematical challenge, but it also provides confidence in the correctness of our solutions. As both RRT and GRT will eventually find a free disk if one is available, and also since we have shown they are equivalent, we shall treat them as the same and henceforth consider only GRT in our analysis.

D. FPA Solution of LBF

As discussed, under LBF, when a request for movie m arrives, it will be directed to the least busy disk in the set Ω_m. In the case where there is more than one least busy disk, the request will be randomly dispatched to one of them. The implementation of LBF thus requires that the VOD scheduler update its channel occupancy information of a disk at each time epoch when a video session is terminated from the disk.

Let $\vec{\xi_j} = (\xi_j^{(0)}, \xi_j^{(1)}, \dots, \xi_j^{(N_j)})$ denote the vector of stationary probabilities that disk j is in state $i, i = 0, 1, 2, \dots, N_j$, and $\vec{\xi} = (\vec{\xi_1}, \vec{\xi_2}, \dots, \vec{\xi_J})$.

Given that disk j is in Ω_m, let's again define $\Upsilon(\Omega_m - \{j\}, x)$ as the set of all possible combinations of choosing x disks out of $\Omega_m - \{j\}$. For movie m, when disk j is in state i, the probability that, among the other $n_m - 1$ disks containing movie m, $h - 1$ disks also have i channels occupied, and the remaining $n_m - h$ disks have more than i channels occupied is

$$P_j(h,i) = \sum_{\mathcal{S}\in\Upsilon(\Omega_m-\{j\},h-1)} \prod_{u\in\mathcal{S}} \xi_u^{(i)} \cdot \prod_{v\in\Omega_m-\{j\}-\mathcal{S}} \sum_{k=i+1}^{N_v} \xi_v^{(k)} \quad (16)$$

for $i = 0, 1, \ldots, N_j - 1$ and $h = 1, 2, \ldots, n_m$. Note that if $n_m = 1$, we simply set $P_j(h, i) = 1$.

Hence, when disk j is in state i, its movie m request arrival rate is

$$y_j^{(i)}(m) = \sum_{h=1}^{n_m} \left(\frac{\lambda p_m}{h} \right) P_j(h, i) \quad (17)$$

and its total request arrival rate due to all movie files is

$$y_j^{(i)} = \sum_{m \in \Phi_j} y_j^{(i)}(m). \quad (18)$$

Let $\vec{y}_j = (y_j^{(0)}, y_j^{(1)}, \ldots, y_j^{(N_j-1)})$ and $\vec{y} = (\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_J)$. Thus, (16)–(18) define a function $f(\cdot)$ that can be used to obtain $\vec{y}$ from $\vec{\xi}$

$$\vec{y} = f(\vec{\xi}). \quad (19)$$

On the other hand, let us model the state transition process of disk j as a birth-death process with the birth rate $y_j^{(i)}, i = 0, 1, \ldots, N_j - 1$, and the death rate $i\hat{\mu}_j^{(i)}, i = 1, 2, \ldots, N_j$, where

$$\frac{1}{\hat{\mu}_j^{(i)}} = \frac{1}{y_j^{(i-1)}} \sum_{m \in \Phi_j} \frac{y_j^{(i-1)}(m)}{\mu_m}. \quad (20)$$

From the steady-state equations of a birth-death process (see, e.g., [7, p. 31]), we have

$$\xi_j^{(i)} = \frac{N_j!}{i! \prod_{k=i}^{N_j-1} \frac{y_j^{(k)}}{\hat{\mu}_j^{(k+1)}}} \xi_j^{(N_j)}. \quad (21)$$

By normalization, we obtain

$$\sum_{i=0}^{N_j-1} \frac{N_j!}{i! \prod_{k=i}^{N_j-1} \frac{y_j^{(k)}}{\hat{\mu}_j^{(k+1)}}} \xi_j^{(N_j)} + \xi_j^{(N_j)} = 1. \quad (22)$$

Therefore, for disk $j, j = 1, 2, \ldots, J$, (20)–(22) define a function $g(\cdot)$ that can be used to obtain $\vec{\xi}$ from $\vec{y}$

$$\vec{\xi} = g(\vec{y}). \quad (23)$$

The system of (19) and (23) composes the following fixed-point equations:

$$\vec{\xi} = g(f(\vec{\xi})). \quad (24)$$

By the disk independence assumption and having obtained $\vec{\xi}$ by solving (24), B_m is given by (10). $\tilde{B}$, $\hat{B}$, and $\bar{B}$ are then computed by (5), (6), and (7), respectively.

E. Comments on Computational Effort

The fixed-point (9) and (24) can often be solved efficiently by the following successive substitution method. Let $T(\mathbf{z})$ denote the transformation given by the RHS of (9) or (24), where $\mathbf{z}$ is the vector $\vec{\xi}'$ in (9) or the vector $\vec{\xi}$ in (24). We start the successive substitution procedure with a certain initial vector $\mathbf{z}_0$, and then iteratively compute $\mathbf{z}_{k+1} = T(\mathbf{z}_k)$, for $k = 0, 1, \ldots$, until $\mathbf{z}_{k+1}$ is sufficiently close to $\mathbf{z}_k$.

Since the transformation $T(\mathbf{z})$ is a continuous mapping from the compact set $[0,1]^J$ in (9) or $[0,1]^{\sum_{j=1}^{J} N_j + 1}$ in (24) to itself, by Brouwer's fixed-point theorem [25], a fixed point exists such that $\mathbf{z} = T(\mathbf{z})$, ensuring that the fixed-point solutions for (9) and (24) exist. Although we can not establish the uniqueness of the solutions, for various numerical experiments we have conducted, the iterations have always stabilized at the same fixed point regardless of $\mathbf{z}_0$.

It may appear that the task of enumerating the combination set $\Upsilon(\cdot)$ involved in the computation of (11) and (16) is cumbersome. However, the size of $\Upsilon(\cdot)$ merely depends on the number of copies a movie has in the system. It is important to observe in (10) the dramatic performance gain obtainable by a movie m if its number of replicas increases from n_m to $n_m + 1$. Consequently, it is not necessary in practice to place a large number of copies for any movie to achieve significant reduction in its request blocking probability. Not only does this fact dramatically save storage space, but it guarantees that the size of the combination set is small, as we can set the maximal number of replicas of any movie to be not too large.

IV. Numerical Examples

We consider a VOD system that manages 20 disks and provides users with on-demand access to a library of 200 distinct movies. Each disk has a storage space of 11 units, and supports up to 30 concurrent video streams. The movie file-size is randomly generated, ranging from as small as 0.76 units to as large as 1.29 units. It requires an overall storage space of 201.32 units to allocate a file-copy for each movie. Consequently, the system has an extra storage space of 18.68 units to accommodate multiple file-copies for certain movies.

Let the popularity profiles of the 200 movies be distributed so that $p_m = m^{-\zeta} / \sum_{k=1}^{200} k^{-\zeta}$, for $m = 1, 2, \ldots, 200$, where the parameter ζ determines the skewness of the distribution. This distribution is known as a Zipf-like distribution, since when $\zeta = 1$ it is the Zipf distribution [26]. It was found that such a distribution with $\zeta = 0.271$ statistically matches client access frequencies to various movie titles observed from the video rental business [27]. We assume that the channel holding time for movie m, taking into consideration user interactive behavior [28], follows a lognormal distribution. Without loss of generality, we assume that the value of the mean channel holding time for movie m is equal to the value of its file-size. For the purpose of our simulation, we set the standard deviation of channel holding time for movie m to be equivalent to its mean, and set one time unit to be one hundred minutes.

A. Approximation Validation

In the first example, we investigate the accuracy of our analysis for GRT and LBF by means of a discrete event simulation study. For this purpose, we use one feasible file assignment that allocates two file-copies for 14 movies and one file-copy for all other movies. We validate the analytical results with respect to its mean request blocking probability of multicopy movies, single-copy movies, and all movies in $\mathcal{F}$. The analytical formulas and the simulation tests are implemented in C programming language.

In a typical run of the simulation process, each of the one hundred million random events represents either the arrival of a request or the termination of a video session for a movie in $\mathcal{F}$. We obtain B_m by counting the number of its request arrivals and the number of its request losses. $\tilde{B}$, $\hat{B}$, and $\bar{B}$ are then computed by (5), (6), and (7), respectively. The simulation estimates are presented in Fig. 1 with λ ranging from 420 to 520 at increments of 10. By repeating the simulation test with six independent runs, we have kept the radii of the 95% confidence intervals ([29, p. 273]) between 0.09% and 0.37% of the observed mean at each data point for $\bar{B}$, between 0.09% and 0.37% for $\hat{B}$, and between 0.26% and 3.47% for $\tilde{B}$.

We see that the approximate results for the mean request blocking probability of all movies are very close to those from the simulation. The approximation is also successful in predicting the results of single-copy movies. Moreover, the results of multicopy movies are very good under GRT. Unfortunately, the results of multicopy movies are less accurate under LBF. This is mainly due to the fact that the FPA solution

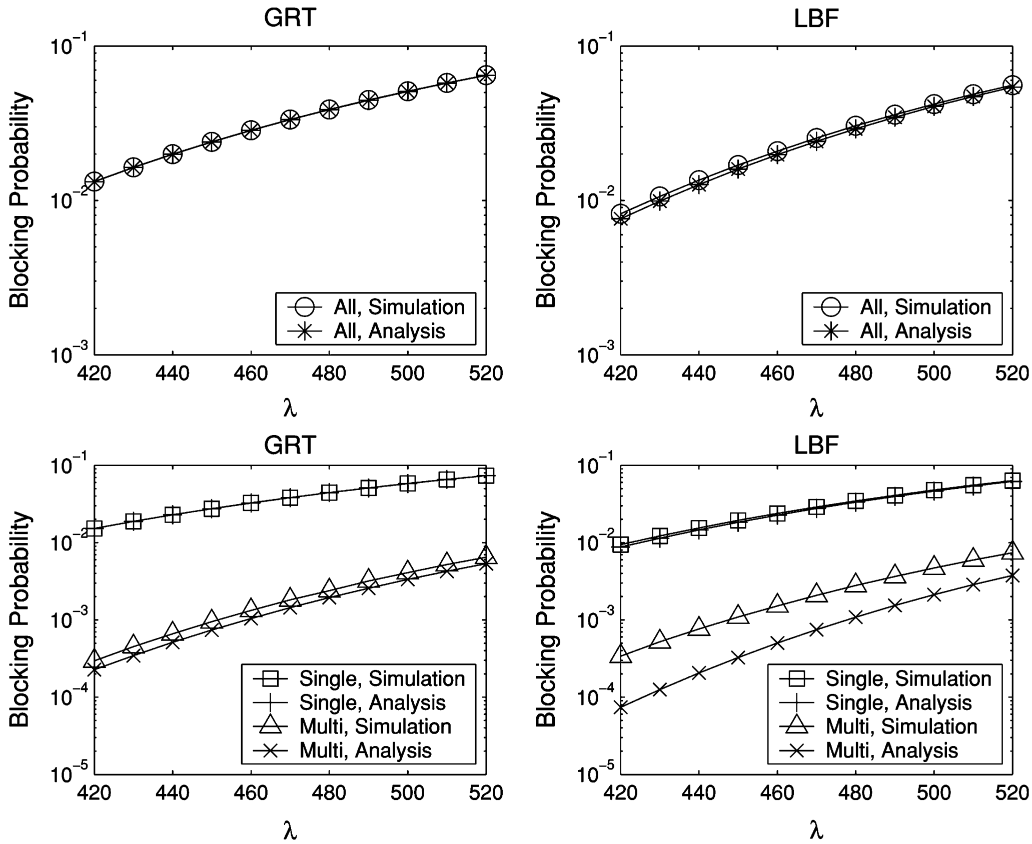

Fig. 1. Approximation validation.

TABLE II
INTENSITY OF REDUNDANT ATTEMPTS MADE BY RRT

λ	420	430	440	450	460	470	480	490	500	510	520
$(q_a - q_r)/q_r$ (%)	1.5	1.8	2.2	2.7	3.2	3.8	4.4	5.0	5.7	6.5	7.3

TABLE III
SIGNALLING OVERHEAD OF GRT IN COMPARISON WITH LBF

λ	420	430	440	450	460	470	480	490	500	510	520
Ratio (%)	2.2	2.6	3.1	3.7	4.3	5.0	5.7	6.4	7.3	8.1	9.0

of LBF involves state-dependent elements. Consequently, the two assumptions made by FPA in Section I-B tend to introduce more errors to the approximate solution of LBF than that of GRT. Nevertheless, this is less of a practical problem since in practice it is more realistic that we measure the performance of a VOD system using either the mean request blocking probability of all movies or the worst blocking performance of the system (usually experienced by requests for single-copy movies). We will use $\bar{B}$ as the performance metric in all following examples.

As a byproduct of the simulation study, we demonstrate the inefficiency of RRT in handling requests for multicopy movies. Let q_a count the number of random attempts for multicopy movies in RRT. Let q_r count the number of requests for multicopy movies. The performance metric used in Table II indicates the intensity of redundant attempts made by RRT. Even though there are only 14 multicopy movies in this example and all of them have merely two file-copies, the overhead due to the round trip signalling involved in the redundant attempts is quite significant. This is especially true under high traffic loads. We also compare the signalling overhead between the implementation of GRT and that of LBF. As we have discussed, both schemes require that the VOD scheduler collect certain state information from the disks. Results in Table III demonstrate that LBF entails considerably larger signalling overhead than GRT. For this particular example, the signalling overhead of GRT is less than 10% of that of LBF even at high traffic loads.

TABLE IV
FILE ASSIGNMENT QUALITY COMPARISON

Cases	SRT $\bar{B}$		GRT $\bar{B}$			LBF $\bar{B}$		
	Analysis	Order	Simulation	Analysis	Order	Simulation	Analysis	Order
1	0.01874	8	0.01650	0.01644	3	0.01017	0.00944	1
2	0.01860	7	0.01651	0.01650	4	0.01051	0.00979	2
3	0.01836	3	0.01636	0.01632	2	0.01062	0.00990	3
4	0.01816	2	0.01626	0.01621	1	0.01076	0.01020	4
5	0.01853	6	0.01674	0.01674	5	0.01159	0.01091	5
6	0.01877	9	0.01709	0.01707	8	0.01196	0.01122	6
7	0.01848	5	0.01697	0.01695	6	0.01242	0.01187	7
8	0.01846	4	0.01703	0.01703	7	0.01254	0.01207	8
9	0.01911	10	0.01760	0.01761	10	0.01282	0.01230	9
10	0.01780	1	0.01755	0.01756	9	0.01661	0.01661	10

B. Applications of the Models

In the second example, we investigate if the proposed analytical solutions are applicable to the task of file assignment optimization in a VOD system. From the large space of feasible file assignments for the test system, we further present the results of nine file assignments for comparison in Table IV with the one studied in the first example. We consider these different cases because their blocking probabilities are

TABLE V
CPU TIME COMPARISON

	Simulation	Computation		Simulation	Computation
GRT	1308.99 sec	0.00082 sec	LBF	1312.54 sec	0.02244 sec

close to each other. In this way, we can see if the analytical models are sufficiently accurate to establish the quality in terms of request blocking probability for a file assignment.

In addition to both simulation and analytical results for GRT and LBF, we also report the analytical results for SRT in Table IV for comparison. Table V gives the average CPU time required to obtain the corresponding simulation and analytical results on a 2.4 GHz Pentium 4 machine. We observe that 1) for both GRT and LBF, the quality of each file assignment established by analysis is consistent with that confirmed by simulation, but the CPU time required by analysis is drastically reduced and 2) a file assignment that achieves a smaller blocking probability in the SRT system does not necessarily lead to a smaller blocking probability in systems where GRT and LBF are used. In fact, for this particular example, the best file assignment established in the SRT system turns out to be the worst in the LBF system. This justifies the importance of our analysis in this paper to support the task of file assignment optimization in the VOD system using the more efficient resource selection schemes.

V. CONCLUSION

In this paper, we have studied three exhaustive resource selection schemes, namely RRT, GRT and LBF, for a large scale VOD system. These schemes utilize system resources efficiently given the existence of multicopy movies. We have provided an analytical framework for analysis of the VOD system using these efficient resource selection schemes. The proposed analytical framework relies on the well-known FPA method, so that fast and sufficiently accurate solutions can be derived to facilitate the design and dimensioning of the VOD system. We have rigorously proved the equivalence between the FPA solutions of RRT and GRT. This allows us to use either of the two analytical solutions in practice, as none of them is more accurate in approximating the exact blocking performance of the VOD system. The results from this paper are directly used in [12], where they support an optimization program aiming to obtain the optimal or near-optimal assignment of movie files that minimizes the request blocking probability of the VOD system given a configuration of multiple disks.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their valuable comments.

REFERENCES

[1] W. D. Sincoskie, “System architecture for a large scale video on demand service,” *Comput. Networks ISDN Syst.*, vol. 22, no. 2, pp. 155–162, Sep. 1991.
[2] K. Cleary, “Video on demand-competing technologies and services,” in *Proc. Int. Broadcast. Conf. IBC ’95*, Amsterdam, The Netherlands, Sep. 1995, pp. 432–437.
[3] D. Deloddere, W. Verbiest, and H. Verhille, “Interactive video on demand,” *IEEE Commun. Mag.*, vol. 32, no. 5, pp. 82–88, May. 1994.
[4] D. Sitaram and A. Dan, *Multimedia Servers: Applications, Environments and Design*. New York: Morgan Kaufmann, 2000.
[5] J. Guo, P. G. Taylor, M. Zukerman, S. Chan, K. S. Tang, and E. W. M. Wong, “On the efficient use of video-on-demand storage facility,” in *Proc. IEEE ICME 03*, Baltimore, MD, Jul. 2003, vol. 2, pp. 329–332.
[6] T. D. C. Little and D. Venkatesh, “Popularity-based assignment of movies to storage devices in a video-on-demand system,” *Multimedia Syst.*, vol. 2, no. 6, pp. 280–287, Jan. 1995.
[7] H. Akimaru and K. Kawashima, *Teletraffic: Theory and Applications*, 2nd ed. London, U.K.: Springer-Verlag, 1999.
[8] S. L. Tsao, M. C. Chen, M. T. Ko, J. M. Ho, and Y. M. Huang, “Data allocation and dynamic load balancing for distributed video storage server,” *J. Vis. Commun. Image Represent.*, vol. 10, no. 2, pp. 197–218, 1999.
[9] J. L. Wolf, P. S. Yu, and H. Shachnai, “Disk load balancing for video-on-demand systems,” *Multimedia Syst.*, vol. 5, no. 6, pp. 358–370, Nov. 1997.
[10] Y. Zhao and C. C. J. Kuo, “Video-on-demand server system design with random early migration,” in *Proc. IEEE ISCAS 03*, Bangkok, Thailand, May 2003, vol. 2, pp. 640–643.
[11] K. S. Tang, K. T. Ko, S. Chan, and E. Wong, “Optimal file placement in VOD system using genetic algorithm,” *IEEE Trans. Ind. Electron.*, vol. 48, no. 5, pp. 891–897, Oct. 2001.
[12] J. Guo, “Two problems in stochastic service systems,” Ph.D. dissertation, Univ. Melbourne, Melbourne, Australia, 2006.
[13] F. P. Kelly, “Blocking probabilities in large circuit-switched networks,” *Adv. Appl. Probability*, vol. 18, no. 2, pp. 473–505, Jun. 1986.
[14] F. P. Kelly, “Loss networks,” *Ann. Appl. Prob.*, vol. 1, no. 3, pp. 319–378, Aug. 1991.
[15] K. W. Ross, *Multiservice Loss Models For Broadband Telecommunication Networks*. London, U.K.: Springer-Verlag, 1995.
[16] G. R. Ash, *Dynamic Routing in Telecommunications Networks*. New York: McGraw Hill, 1998.
[17] J. M. Akinpelu, “The overload performance of engineered networks with nonhierarchical and hierarchical routing,” *Bell Syst. Tech. J.*, vol. 63, no. 7, pp. 1261–1281, 1984.
[18] E. W. M. Wong and T. S. Yum, “Maximum free circuit routing in circuit-switched networks,” in *Proc. IEEE INFOCOM ’90: 9th Annu. Joint Conf.*, San Francisco, CA, Jun. 1990, vol. 3, pp. 934–937.
[19] S. P. Chung, A. Kashper, and K. W. Ross, “Computing approximate blocking probabilities for large loss networks with state-dependent routing,” *IEEE/ACM Trans. Netw.*, vol. 1, no. 1, pp. 105–115, Feb. 1993.
[20] G. R. Ash and B. D. Huang, “An analytical model for adaptive routing networks,” *IEEE Trans. Commun.*, vol. 41, no. 11, pp. 1748–1759, Nov. 1993.
[21] B. A. Sevast’yanov, “An ergodic theorem for Markov processes and its application to telephone systems with refusals,” *Theor. Probab. Appl.*, vol. 2, no. 1, pp. 104–112, 1957.
[22] C. P. Costa, I. S. Cunha, A. Borges, C. V. Ramos, M. M. Rocha, J. M. Almeida, and B. Ribeiro-Neto, “Analyzing client interactivity in streaming media,” in *Proc. 13th Int. Conf. World Wide Web*, New York, 2004, pp. 534–543.
[23] G. T. Kolyvas, S. E. Polykalas, and I. S. Venieris, “Performance evaluation of intelligent signaling servers for broadband multimedia networks,” in *Proc. 2nd IEEE Symp. Comput. Commun.*, Alexandria, Egypt, Jul. 1997, pp. 96–103.
[24] H. W. Gould, *Combinatorial Identities: A Standardized Set of Tables Listing 500 Binomial Coefficient Summations*. Morgantown, WV: West Virginia Univ. Press, 1972.
[25] J. R. Munkres, *Elements of Algebraic Topology*. Cambridge, MA: Addison-Wesley, 1984.
[26] G. K. Zipf, *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Cambridge, MA: Addison-Wesley, 1949.
[27] A. Dan, D. Sitaram, and P. Shahabuddin, “Scheduling policies for an on-demand video server with batching,” in *Proc. 2nd ACM Int. Conf. Multimedia*, San Francisco, CA, 1994, pp. 15–23.
[28] P. Branch, G. Egan, and B. Tonkin, “Modeling interactive behaviour of a video based multimedia system,” in *Proc IEEE ICC’99.*, Vancouver, BC, Canada, Jun. 1999, vol. 2, pp. 978–982.
[29] S. K. Bose, *An Introduction to Queueing Systems*. New York: Kluwer/Plenum, 2002.