Combination Load Balancing for Video-on-demand Systems

Jun Guo, Eric W. M. Wong, Sammy Chan, Peter Taylor, Moshe Zukerman, and Kit-Sang Tang

***Abstract*— We observe that an effect of "disk resource sharing" of multi-copy movie traffic has great impact on the blocking performance of a video-on-demand system. This observation leads us to establish a conjecture on how to balance the movie traffic load among "combination" groups of disks to maximize the level of disk resource sharing. For a given file replication instance, the conjecture predicts in general an effective lower bound on the blocking performance of the system. It motivates the design of a numerical index that measures quantitatively the goodness of disk resource sharing on allocation of multi-copy movie files. It also motivates the design of a greedy file allocation method that decides a good quality heuristic solution for each feasible file replication instance. We further develop analytical formulas to obtain approximate results for the bound fast and accurately. These techniques can be utilized by an optimization program to find near-optimal file assignment solutions for the system computationally efficiently.**

***Index Terms*— Combination load balancing, disk resource sharing, blocking probability, fixed-point approximation, video-on-demand.**

I. Introduction

To compete with the prevalent video rental business, a large scale video-on-demand (VOD) system [1] is envisaged to provide a large population of end users with pleasurable on-demand access to a large variety of movie contents coupled with full VCR-like interactive capabilities [2]. For this purpose, a stringent requirement is usually imposed such that a single video stream (logical channel) is allotted to each user request, so that the same movie can be accessed simultaneously by many users with random time offsets and independent temporal control activities. Due to the inherent limitations of the disk striping approach [3], [4], in most cases, user requests for a movie title in an interactive VOD system can only be connected to a disk in the system where a file-copy of that movie is placed. This entails the replication of those popular movie titles over multiple disks so as to increase the number of concurrent video streams that can be supported by the system to satisfy user requests for these hot movies.

A preliminary version of this paper was presented at IEEE ICC'04, Paris, June 2004. The work described in this paper was partially supported by grants from City University of Hong Kong (Project No. 7001224 and Project No. 7001458) and partially supported by the Australian Research Council (ARC).

J. Guo and M. Zukerman are with the ARC Special Research Centre for Ultra-Broadband Information Networks (CUBIN), an affiliated program of National ICT Australia (NICTA), Department of Electrical and Electronic Engineering, The University of Melbourne, Australia (email: j.guo@ee.mu.oz.au; m.zukerman@ee.mu.oz.au).

P. Taylor is with CUBIN and the Department of Mathematics and Statistics, The University of Melbourne, Australia (email: P.Taylor@ms.unimelb.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 (email: eeewong@cityu.edu.hk; eeschan@cityu.edu.hk; eekstang@cityu.edu.hk).

Part of the work was done while J. Guo and M. Zukerman were visiting the Department of Electronic Engineering, City University of Hong Kong.

Since movie connections between users and disks are long-lived by nature, VOD systems are often modelled as loss systems [5]–[7]. That is, a user request wishing to access a movie title is blocked if the system cannot serve it right away. Similar to many other service systems, an important issue in the design and operation of the VOD system is to distribute the movie traffic load evenly across the disks in the system. A load balanced VOD system leads to efficient operation and minimum *request blocking probability* (RBP) [5].

If each movie title could have a file-copy stored on each disk in the VOD system, the movie traffic load would be easily balanced across the disks. In such an ideal *full replication* scenario, a user request for any movie title is blocked only if the video stream capacity of the entire system is used up upon the arrival of the request. However, due to the disk storage space constraint, this is unlikely to be practicable in a large scale VOD system that supports a large library of movie contents. It follows that real VOD systems typically support a selective movie file replication. For a given file replication instance, the establishment of the condition on file allocation to achieve load balancing and hence the minimum RBP in this type of system is not a trivial task, except in certain simplified situation where user requests for multi-copy movies are handled in accordance with what we call a *single random trial* (SRT) resource selection scheme.

Following SRT, when a user request for a multi-copy movie arrives, one of the disks storing a file-copy of the requested movie title is randomly selected. If the disk is fully busy, the request is simply blocked, without further attempting any other disk that keeps a file-copy of the requested movie title. In an earlier study [5], Little and Venkatesh showed that the SRT system is load balanced if movie files are optimally allocated such that each homogeneous disk has an equal probability of being accessed. They proposed the conjecture that at this load balanced state, which we call *disk load balancing* (DLB), the RBP of the VOD system is minimized. If DLB is not achievable in practice, the goodness of a suboptimal file allocation solution, defined as its distance to DLB in terms of the RBP of the system, is related to how evenly the movie traffic load is distributed compared with the uniform distribution. Methods of file allocation to achieve optimal or near-optimal load balancing in the SRT system were proposed in [6], [8], [9].

In comparison with more efficient exhaustive resource selection schemes, SRT is inherently inefficient in utilizing system resources given the existence of multi-copy movies [7]. Moreover, we shall see in this paper that the goodness of a file allocation solution established in the SRT system does not hold true in situations where those exhaustive resource selection schemes are used. Two such schemes, namely, *repeated random trials* (RRT) and *least busy fit* (LBF), have been studied

in [7]. In both schemes, a user request is blocked only if all disks (in an exhaustive sense) storing the requested movie file are found to be fully busy. RRT is a natural extension of SRT where we continue with repeated random trials until all the disks are attempted. Making use of the available system state information, an LBF system always directs a user request for a multi-copy movie to the least busy disk (with the maximal number of available logical channels) where a file-copy of the requested movie title is placed.

It was demonstrated in [7] that, in comparison with RRT, LBF provides superior efficiency not only for multi-copy movies but for single-copy movies as well. This is consistent with the findings in circuit-switched networks [10]–[14] that similar least loaded routing schemes provide better performance than random alternate routing schemes. It needs to be noted that, though for brevity we consider LBF only and do not report the results for RRT in this paper, algorithms and analytical methodologies that we propose for the LBF system apply to the RRT system as well [15].

Our focus in this paper is thus to investigate how the movie traffic load can be balanced in the LBF system. In addition, we show how the findings motivate the design of various computationally efficient techniques that can be utilized to support the nontrivial and challenging task of file assignment optimization for the LBF system [15].

The remainder of this paper is organized as follows. Based on the model of the VOD system to be described in Section II, we shall first demonstrate in Section III through a simple example that, the condition on file allocation to achieve load balancing in the SRT system is inappropriate to interpret the true meaning of load balancing in the LBF system. An important point to be realized from this example is that the performance of the LBF system can mostly benefit from a good file allocation instance of multi-copy movie files.

In Section IV, we link the concept of *disk resource sharing* of multi-copy movie traffic with what is considered a good file allocation instance of multi-copy movie files, and relate them both to the RBP of the LBF system. This intuitive observation leads us to propose a conjecture on how the movie traffic load should be ideally distributed in the LBF system to achieve *combination load balancing* (CLB) [16] and to minimize the RBP of the LBF system. To verify that a movie file allocation instance that achieves CLB yields the minimum RBP in the LBF system, we design an efficient discrete event simulation study in Section IV-A to evaluate the RBP of CLB-LBF (the LBF system that attains CLB). We then justify in Section IV-B that CLB in general predicts an effective lower bound on the RBP of the LBF system. Observing that the factor of disk resource sharing of multi-copy movie traffic has great impact on the RBP of the LBF system, we propose in Section IV-C a measure that estimates for a given file allocation instance how well the multi-copy movie traffic load is shared between the disks, as compared with the ideal file allocation instance that attains CLB. These results are then justified again in Section IV-D by the simulation study of a realistic example comprising a large system.

In Section V, we show how the concept of disk resource sharing further motivates us to devise a greedy file allocation method that obtains a heuristic file allocation instance of sufficiently good quality for a given feasible file replication instance. We discuss the important application of this heuristic algorithm to handle the nontrivial and challenging task of file assignment optimization for the LBF system.

Considering the excessive CPU time generally required by simulation, it is useful to develop a fast analytical method for evaluating the RBP of CLB-LBF. Due to the complicated interactions between user requests for multi-copy movies and selections of disks to serve these requests, an exact solution for CLB-LBF is not tractable, but we are able to apply the fixed-point method [17] to derive approximate results analytically in Section VI. The accuracy of the approximation is validated against simulation. We demonstrate how the analytical model of CLB-LBF can be utilized to improve the runtime efficiency of the file assignment optimization program.

Finally, we give our concluding remarks in Section VII.

II. System Model

Let the VOD system be composed of a set $\mathcal{D}$ of J homogeneous disks, labelled $1,\ 2, \ldots,\ J$. Each disk has a limited storage space of C units. (For example, one unit of storage space could be one Gbyte.) We consider that the independent video streams emanating from a disk are approximately statistically equivalent [18]. Each disk may support up to N concurrent video streams (logical channels). In cases where the system consists of heterogeneous disks, we assume the use of disk merging techniques [19], so that a logical collection of J homogeneous disks can be constructed from the array of heterogeneous disks.

The system offers a large library of movie contents which contains M distinct movie titles, marked $1,\ 2, \ldots,\ M$. The set of these M movie titles is denoted $\mathcal{F}$. The file-size of movie m is L_m units. Therefore, it requires $L = \sum_{m \in \mathcal{F}} L_m$ units of disk storage space to allocate one file-copy for each movie in $\mathcal{F}$. We assume $\max_{m \in \mathcal{F}} L_m \ll C$, so that each disk can store a number of movie files. We also assume $L < JC$, so that the system has spare disk storage space to place multiple file-copies for certain movies in $\mathcal{F}$. The set of movie files placed on disk j is denoted Φ_j.

To extract from a movie file assignment the information of how each distinct movie title is replicated and where the movie file and its replicas (if it is replicated) are allocated, we define the following two concepts. Let a *file replication instance* define a realization of the vector $\mathbf{n} = (n_1, n_2, \ldots, n_M)$, where n_m, $m \in \mathcal{F}$, indicates the integer number of file-copies of movie m, and $1 \leq n_m \leq J$. We call a movie title that has c file-copies a *Type c movie*. Let a *file allocation instance* define a disk location arrangement $\mathbf{\Omega} = (\Omega_1, \Omega_2, \ldots, \Omega_M)$ for the set of movie files specified in a file replication instance $\mathbf{n}$, subject to the storage space constraint on each disk. Each of the elements Ω_m, $m \in \mathcal{F}$, describes the set of n_m different disks where the n_m file-copies of movie m are stored. By the definition of a feasible file replication instance, we require that at least one valid file allocation instance can be realized for the associated file replication instance subject to the disk storage space constraint.

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 [20], 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 movie titles follow a Poisson process with rate λ requests per time unit. (For example, one time unit could be one hour.) The request arrival processes of different movie titles are mutually independent Poisson processes.

The connection time of movie m, taking into consideration the user interactive behaviours [21], follows a lognormal distribution with mean $1/\mu_m$ time units. Without loss of generality, we assume that the value of the mean connection time $1/\mu_m$ for movie m is identical to the value of its file-size L_m, and the standard deviation of the connection time of movie m is equivalent to its mean.

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 $\sum_{m\in\mathcal{F}} p_m = 1$. For a given file replication instance $\mathbf{n}$, the popularity profile $\hat{p}_c$ of its Type c movies is obtained by $\hat{p}_c = \sum_{m\in\mathcal{F}, n_m=c} p_m$. The mean connection time $1/\hat{\mu}_c$ for a Type c movie is thus given by

$$\frac{1}{\hat{\mu}_c} = \frac{1}{\hat{p}_c} \sum_{m\in\mathcal{F}, n_m=c} \frac{p_m}{\mu_m}. \tag{1}$$

In practice, movie popularity profiles are updated periodically to capture the variability of user demand. During the time interval between such updates, the request arrival rate of movie m is given by λp_m. Therefore, the traffic load A_m of movie m is given by $\lambda p_m/\mu_m$. The aggregate traffic load $\hat{A}_c$ of all Type c movies is obtained by $\sum_{m\in\mathcal{F},\ n_m=c} A_m$. The aggregate traffic load A of all movies in $\mathcal{F}$ is computed by $\sum_{m\in\mathcal{F}} A_m$.

We assume in this paper that the popularity profiles of movie titles in a VOD system are distributed following a mathematical function given by

$$p_m = \frac{m^{-\zeta}}{\sum_{k=1}^{M} k^{-\zeta}} \tag{2}$$

for $m \in \mathcal{F}$. The parameter ζ in (2) determines the skewness of the distribution. This distribution function is commonly known as a Zipf-like distribution, since when $\zeta = 1$ it becomes a Zipf distribution [22]. It was found in [23] that such a distribution with $\zeta = 0.271$ statistically matches client access frequencies to various movie titles observed from the video rental business.

III. Disk Load Balancing

If a user request for a multi-copy movie m is handled according to SRT, it is randomly forwarded to only one of the disks in the set Ω_m. No effort is made to handle the request more efficiently among the n_m disks in Ω_m. Given that the request arrival process of each of the M movie titles in the system is Poisson, the request arrival process of movie m is simply decomposed into n_m independent Poisson processes, each of which has rate $\lambda p_m/n_m$, assuming an equal probability. Each file-copy of movie m is therefore equivalent to a single-copy movie with traffic load A_m/n_m on the disk where it is stored.

TABLE I

Movie popularity distribution in the four-disk example

Movies	1	2	3	4	5
Popularity	0.08655	0.07173	0.06427	0.05945	0.05596
Movies	6	7	8	9	10
Popularity	0.05326	0.05108	0.04927	0.04772	0.04638
Movies	11	12	13	14	15
Popularity	0.04519	0.04414	0.04319	0.04233	0.04155
Movies	16	17	18	19	20
Popularity	0.04083	0.04016	0.03954	0.03897	0.03843

Provided that all disks are homogeneous in storage space and stream capacity, Little and Venkatesh conjectured in [5] that the RBP of the SRT system is minimal if and only if movie files can be optimally allocated such that the traffic load on each of the homogeneous disks in the SRT system is identical. At such a load balanced state, the traffic load on each disk is exactly A/J, so that the RBP of the SRT system that achieves DLB can be exactly given by the Erlang B Formula [24]

$$\text{RBP} \stackrel{\text{def}}{=} E\left(\frac{A}{J}, N\right) = \frac{(\frac{A}{J})^N/N!}{\sum_{i=0}^{N}(\frac{A}{J})^i/i!}\ . \tag{3}$$

The following numerical example, however, will demonstrate that the condition on file allocation to achieve load balancing in the SRT system is inappropriate to interpret the true meaning of load balancing in the LBF system. To support our argument in an intuitive and comprehensible manner, we specifically consider a simple example of a small system with four disks and 20 distinct movie titles. Each disk in this example has a storage space of eight units, and supports up to ten concurrent video streams. Each movie title has a file-size of one unit. As a result, the mean connection time of any movie title is one time unit. The popularity profiles of these 20 movie titles, in a descending order, are given in Table I. For this particular example, we assume that the aggregate rate $\lambda = 24$ requests per time unit. By (3), the minimum RBP of the SRT system in this example is calculated to be 4.314%.

We consider a specific file replication instance where movies 1 to 12 have two file-copies, and movies 13 to 20 are all single-copy movies. From many possible disk location arrangements for these 32 movie files, we select three valid file allocation instances as shown in Fig. 1.

We remember in an SRT system that a file allocation instance is optimal if the aggregate movie traffic load A can be uniformly distributed among J disks in the system. To differentiate and compare the level of load balancing among these three file allocation instances in the SRT system, we define a *load balancing index* (LBI), given by

$$\text{LBI} = \sqrt{\frac{1}{J}\sum_{j\in\mathcal{D}}\left(\sum_{m\in\Phi_j}\frac{A_m}{n_m} - \frac{A}{J}\right)^2}\ . \tag{4}$$

LBI measures for a file allocation instance how evenly the movie traffic load is spread among J disks in the SRT system. This is equivalent to the definition of *standard deviation* that

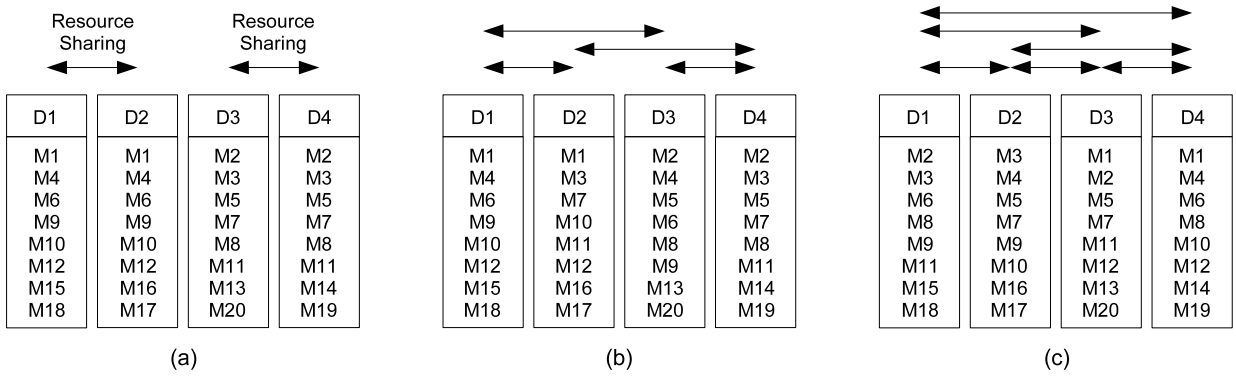

Fig. 1. Movie file allocation instances in the four-disk example: (a) Only two pairs of disks share multi-copy movie traffic; (b) Only four pairs of disks share multi-copy movie traffic; (c) Each pair of disks shares multi-copy movie traffic.

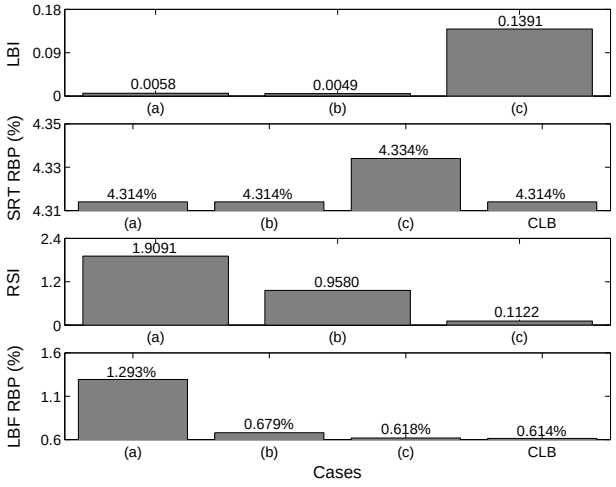

Fig. 2. LBI, SRT RBP, RSI and LBF RBP results in the four-disk example.

was used in [5], so that a smaller value of LBI indicates a more balanced load distribution in the SRT system. After a routine computation of (4) for each of the three file allocation instances considered in Fig. 1, the LBI results reported in Fig. 2 indicate that both (a) and (b) are close to a uniform distribution, but (c) is not as load balanced as (a) and (b).

We next verify the quality of each file allocation instance in the SRT system against what is indicated by LBI. We also check if the goodness of file allocation established in the SRT system applies to the LBF system as well. For this purpose, we conduct a discrete event simulation study [25] for SRT and LBF, respectively. In a typical run of the simulation test, each of the one hundred million random events represents either the arrival of a user request or the termination of a movie connection. We obtain the RBP by counting the total number of request arrivals and the total number of request losses. To guarantee the confidence in our simulation estimates, we repeat the simulation test with multiple independent runs, and we keep the radii of the 95% confidence intervals ([26], page 273) within 1% of the average of the results measured. More details of the simulation study can be found in [15].

We see from the SRT RBP results in Fig. 2 that, the RBP of the SRT system agrees with what Little and Venkatesh conjectured in [5]. Both (a) and (b) achieve DLB in the SRT system, and yield the minimum RBP as computed by (3). However, they produce significantly different RBP results in the LBF system. As we observe from the LBF RBP results in Fig. 2, a user request experiences much smaller RBP in (b) than in (a) when LBF is operated. Moreover, there is an even smaller LBF RBP in (c) despite its poor LBI result in the SRT system. Apparently, LBI is inappropriate in explaining the real goodness of file allocation in the LBF system.

IV. Combination Load Balancing

A closer examination of Fig. 1 reveals that in all the three file allocation instances, the disk locations of the single-copy movies are the same, and the single-copy movie traffic load is almost uniformly distributed. However, the disk locations of Type 2 movies are significantly different. In (a), Type 2 movies are allocated such that there are just two pairs of disks that have common Type 2 movies. Consequently, disk 1 shares Type 2 movie traffic only with disk 2, and disk 3 shares Type 2 movie traffic only with disk 4. In (b), Type 2 movies are interlaced in such a way that there are four different pairs of disks that have the sharing of Type 2 movie traffic within each pair. In (c), the degree of interlace is even higher. Each of the six pairs of disks has common Type 2 movies, so that each disk in the system shares Type 2 movie traffic with each of the other three disks.

Since in this example any pair of disks must be in one of the $\binom{4}{2}$ combination groups of two disks enumerated in the set $\mathcal{D}$, we then carefully work out the proportion of Type 2 movie traffic accessing each of the six combination groups of two disks. We see in Fig. 3 that the Type 2 movie traffic is more balanced among the six combination groups in (b) than that in (a), and the one in (c) is even more balanced.

We observe from these three file allocation instances that, in the LBF system, the factor of disk resource sharing of multi-copy movie traffic shows great impact on the RBP of the system. The more pairs of disks that have common Type 2 movies and the more balanced disk resource sharing of Type 2 movie traffic, the smaller RBP of the system. Similarly, for a system that contains Type c movies, $c \geq 2$, we would expect to maximize the level of disk resource sharing in serving Type c movie traffic, if we could have the uniform resource sharing

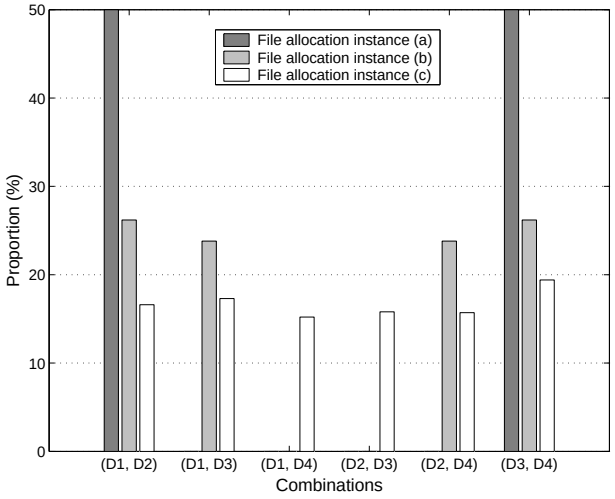

Fig. 3. Proportion of Type 2 movie traffic on each combination group of two disks in the four-disk example.

of Type c movie traffic load within each possible combination group of c disks. Inspired by this intuitive observation, we thus arrive at our following conjecture on how the movie traffic load (of both multi-copy movies and single-copy movies) should be ideally distributed to achieve load balancing and to minimize the RBP of the LBF system.

Conjecture 1: The VOD system is defined as being combination load balanced if, for each c, $c \geq 1$, the traffic wishing to access movies of Type c is uniformly distributed among all $\binom{J}{c}$ combination groups of c disks enumerated in the set $\mathcal{D}$ of all J disks in the system. At such a state of combination load balancing, the RBP of the system is minimized.

Our conjecture suggests that for the LBF system with a specified file replication instance, a file allocation instance that ideally attains CLB always yields a lower bound on the RBP. As a special case, our conjecture also aligns with the conjecture of [5] for the SRT system. Under CLB, for each c, $c \geq 1$, the traffic load of Type c movies is evenly distributed among $\binom{J}{c}$ combination groups of c disks enumerated in the set $\mathcal{D}$. Since any disk in $\mathcal{D}$ is in $\binom{J-1}{c-1}$ combination groups, the traffic of Type c movies wishing to access any disk, assuming SRT, is exactly given by

$$\frac{\binom{J-1}{c-1}\hat{A}_c}{c\binom{J}{c}} = \frac{\hat{A}_c}{J}$$

and the traffic load on each disk due to all types of movies is exactly given by

$$\sum_c \frac{\hat{A}_c}{J} = \frac{A}{J} \ .$$

This demonstrates that CLB-SRT is one realization of DLB.

A. Simulation study of CLB

To evaluate the exact RBP result of CLB-LBF, and also to verify that CLB-SRT is one realization of DLB, we conduct a simulation study for CLB-SRT and CLB-LBF, respectively. To this end, we modify the discrete event simulation described in Section III as follows. During each simulation run, if the random event is a user request, we merely look at the type of the movie requested by the user. If it is a Type c movie, $c \geq 1$, we then find out on which combination group of c disks the c file-copies of the movie are stored. Since under CLB, the traffic of Type c movies is load balanced among $\binom{J}{c}$ combination groups of c disks, each combination group therefore has equal likelihood of being accessed. For the purpose of simulation, instead of maintaining a cumbersome list of $\binom{J}{c}$ combination groups and then randomly choosing one of them upon the request for a Type c movie, we use an equivalent (but more efficient) way of randomly selecting c disks out of the set $\mathcal{D}$. Once such a combination group of c disks is generated, we proceed with the SRT scheme or the LBF scheme to handle the user request. The remaining procedures in processing each user request and obtaining the RBP result of the system readily follow what has been described in Section III.

B. Justification

For the small system example considered in Section III, we conduct the simulation study of CLB and present in Fig. 2 the RBP results for CLB-SRT and CLB-LBF. It confirms that CLB-SRT obtains exactly the same RBP result as what is computed by (3). Comparing with the suboptimal solutions (a), (b) and (c) considered in Fig. 1, CLB-LBF clearly yields the minimum RBP of the LBF system. Moreover, in this particular example, the LBF RBP result due to the file allocation instance (c) is almost indistinguishable from that of CLB.

It must be noted that CLB would be less likely achievable in situations where either the traffic load of a type of multi-copy movies can not be evenly split into each of the associated combination groups of disks, or the distribution of single-copy movie traffic deviates significantly from the uniform distribution. In such situations, however, CLB would be expected to predict a lower bound on the RBP of the LBF system.

Although we are not able to provide a rigorous proof for our conjecture in this paper, we have verified the conjecture through a large number of simulation experiments for many different scales of a VOD system, and have not yet found any counterexamples. For the purpose of justification, eight small system examples are illustrated in Table II. For each of these examples and for the given test file replication instance, we have exhaustively enumerated all valid file allocation instances that satisfy the disk storage space constraint. The LBF RBP results of the optimal solution and of CLB are shown in Fig. 4. These results clearly demonstrate the role of CLB as providing the effective lower bound on the RBP of the LBF system.

C. Resource sharing index

As we have demonstrated, the factor of disk resource sharing of multi-copy movie traffic has great impact on the RBP of the LBF system. In this section, we shall design an efficient measure that allows us to estimate numerically for a given file allocation instance how evenly the multi-copy movie traffic load is shared among the disks, as compared with the ideal file allocation instance that attains CLB.

TABLE II
EXPERIMENTAL SETTINGS FOR CLB JUSTIFICATION

Cases	M	J	$\mathbf{n}$	λ	ζ
(a)	3	2	(2, 1, 1)	5	0.271
(b)	3	2	(2, 1, 1)	5	0.500
(c)	4	3	(2, 1, 1, 1)	7	0.271
(d)	4	3	(2, 1, 1, 1)	7	0.500
(e)	5	3	(2, 2, 1, 1, 1)	7	0.271
(f)	5	3	(2, 2, 1, 1, 1)	7	0.500
(g)	6	4	(3, 3, 2, 2, 1, 1)	10	0.271
(h)	6	4	(3, 3, 2, 2, 1, 1)	10	0.500

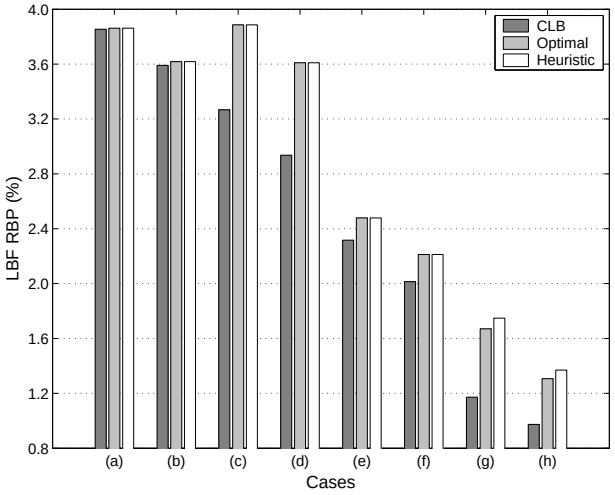

Fig. 4. Experimental results for CLB justification. Heuristic results are obtained from the greedy file allocation method to be presented in Section V.

For a file allocation instance with the existence of multi-copy movies, the traffic wishing to access a multi-copy movie m is distributed among the group of n_m disks in the set Ω_m. Thus, by the concept of disk resource sharing, we say that disk i shares a proportion $S_{ij}^{(m)} = A_m/n_m$ of movie m traffic with disk j, if $i \neq j \in \mathcal{D}$ and both disks are in Ω_m. Using the same reasoning, we can enumerate all multi-copy movies that have a file-copy on both disk i and disk j, and calculate the proportion of multi-copy movie traffic shared between disk i and disk j by $\sum_{m\in\Phi_i,\ m\in\Phi_j} S_{ij}^{(m)}$.

On the other hand, if a file allocation instance ideally achieves CLB, the level of disk resource sharing of multi-copy movie traffic is maximized. For each c, $c \geq 2$, the aggregated traffic load $\hat{A}_c$ of Type c movies is uniformly distributed among all $\binom{J}{c}$ combination groups of c disks enumerated in the set $\mathcal{D}$. Since any two disks in $\mathcal{D}$ coexist in $\binom{J-2}{c-2}$ combination groups of c disks, by the concept of disk resource sharing under CLB, each disk shares with each of the other disks in $\mathcal{D}$ a proportion of Type c movie traffic given by

$$\frac{\binom{J-2}{c-2}\hat{A}_c}{c\binom{J}{c}} = \frac{(c-1)\hat{A}_c}{J(J-1)}$$

and hence a proportion

$$\sum_{c>1} \frac{(c-1)\hat{A}_c}{J(J-1)}$$

of multi-copy movie traffic.

In an equivalent manner to the way we have defined LBI in Section III, we now define a *resource sharing index* (RSI) given by

$$\text{RSI} = \sqrt{\frac{1}{J(J-1)}\sum_{i\neq j\in\mathcal{D}}\left(\sum_{m\in\Phi_i,\ m\in\Phi_j} S_{ij}^{(m)} - \sum_{c>1}\frac{(c-1)\hat{A}_c}{J(J-1)}\right)^2} \tag{5}$$

as a measure of how evenly a file allocation instance distributes the multi-copy movie traffic load. Similarly, a smaller value of RSI indicates a better allocation of multi-copy movie files in the LBF system.

Applying (5) to the three file allocation instances considered in Fig. 1, we obtain the RSI values as presented in Fig. 2. These RSI results reaffirm the goodness on the allocation of multi-copy movie files in the three file allocation instances as we have observed before in Section III.

D. Numerical results for a large system

The size of a large scale VOD system that provides on-demand access to hundreds of distinct movie titles is usually of the order of dozens of disks. Moreover, it typically contains various types of multi-copy movies due to grade of service and reliability requirements and to utilize the spare disk storage space efficiently. In this section, we shall justify our CLB conjecture as well as the RSI measure by considering a large system example of 20 disks and 200 distinct movie titles. Each disk in this example has a storage space of 14 units, and supports up to 30 concurrent video streams. Each movie title has a file-size of one unit. The mean connection time of any movie title is thus one time unit. The popularity profiles of the 200 movie titles follow (2) with $\zeta = 0.271$. By (3), the minimum RBP of the SRT system in this example is calculated to be 2.054%.

For the purpose of this example, we specifically consider a file replication instance where four file-copies are allocated for each of the first three movie titles, three file-copies for movies 4 to 25, two file-copies for movies 26 to 50, and one single file-copy for the remaining 150 movie titles. Among many valid allocation instances of this file replication instance, we choose three of them and compute their respective LBI and RSI values in Fig. 5. The RBP results of the SRT system and of the LBF system are obtained from the simulation study and are presented in Fig. 5 for comparison with LBI and RSI.

We again see in all cases that, CLB clearly justifies its role of being the condition on load balancing in the LBF system so that the minimum RBP of the system can be reached. Moreover, RSI correctly establishes the goodness on the allocation of multi-copy movie files and thus demonstrate the impact of disk resource sharing of multi-copy movie traffic on the RBP of the LBF system.

V. HEURISTIC FILE ALLOCATION

For a given file replication instance, the problem of finding a file allocation instance that achieves CLB (if it is achievable) is NP-hard. This can be established by the theorem in Appendix

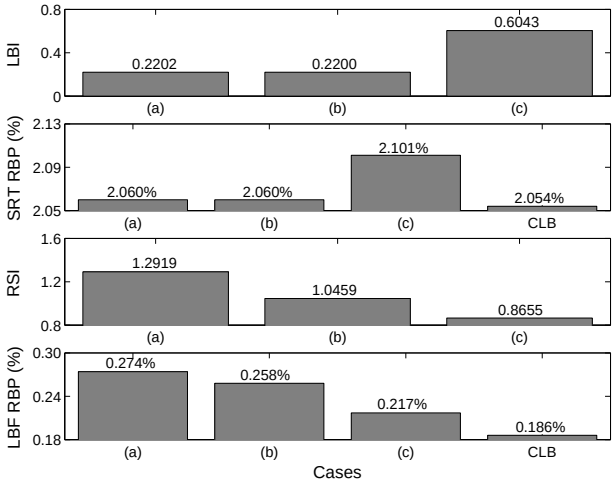

Fig. 5. LBI, SRT RBP, RSI and LBF RBP results in the 20-disk example.

I. Motivated by the concept of disk resource sharing, we present in this section a greedy file allocation method that aims for uniform resource sharing of multi-copy movie traffic as well as uniform distribution of single-copy movie traffic.

Let O_j count the cumulative units of storage space occupied on disk j, $j \in \mathcal{D}$. Let T_j record the cumulative traffic load on disk j, $j \in \mathcal{D}$. Let S_{ij} record the cumulative traffic load shared between disk i and disk j, $i \neq j \in \mathcal{D}$. Once a file-copy of movie m is placed on disk j, we increase O_j by L_m units and T_j by A_m/n_m. If movie m has multiple file-copies allocated in the system, for each i, $i \neq j \in \Omega_m$, we further increase both S_{ji} and S_{ij} by A_m/n_m.

We set out the file allocation procedure by sorting multi-copy movie files in a non-increasing order with respect to A_m/n_m, for all $m \in \mathcal{F}$ and $n_m > 1$. Similarly, we arrange single-copy movie files in a non-increasing order according to A_m, for all $m \in \mathcal{F}$ and $n_m = 1$. We choose to place multi-copy movie files first, due to the stringent requirement that we must always find n_m different disks of sufficient storage space to place the n_m file-copies of a multi-copy movie m. This would be otherwise less likely realizable if we allocate single-copy movie files in advance.

The general steps of our greedy file allocation method proceed as follows: (1) To allocate the first file-copy of a multi-copy movie m, we select disk j with the smallest possible cumulative traffic load, provided $O_j + L_m \leq C$. Subsequently for each of the remaining file-copies of movie m, we select disk i, $i \neq j$, with the smallest possible cumulative traffic load shared with disk j, provided there is not yet a file-copy of movie m stored on disk i and $O_i + L_m \leq C$; (2) To allocate a single-copy movie m, we follow the conventional least loaded first method [8]. Again, we select disk j with the smallest possible cumulative traffic load, provided $O_j + L_m \leq C$.

These steps are repeated until all movie files specified in the file replication instance are successfully allocated, or unless at any stage no disk in $\mathcal{D}$ has sufficient storage space to place a movie file. In the latter case, the file replication instance is treated as *infeasible* due to the inability of the greedy method in finding a valid heuristic file allocation instance. A procedure that implements this greedy method is given in Appendix II.

Clearly, to allocate the first file-copy of each movie title in $\mathcal{F}$, we need to perform a search among the J disks in $\mathcal{D}$ for the disk with sufficient storage space and with the smallest possible cumulative traffic load. To allocate each of the remaining file-copies of a multi-copy movie m, we again need to perform a search among at most $J-1$ disks in $\mathcal{D}$ for the disk with sufficient storage space and with the smallest possible cumulative traffic load shared with the disk where the first file-copy of movie m is placed. Considering that the number of file-copies of each movie title in such a system is at most M, the complexity of the greedy method is $O(M^2J)$.

The good quality of heuristic file allocation due to the proposed greedy file allocation method is evidenced by the file allocation instance (c) in both the four-disk example and the 20-disk example considered in Section III. Both file allocation instances are indeed obtained from the greedy method. In the four-disk example, we have seen in Fig. 2 that the RBP result of the heuristic solution is almost indistinguishable from the CLB bound in the LBF system. Although in the 20-disk example the RBP result of the heuristic solution is nearly 17% deviated from the CLB bound, the actual quality of the heuristic solution is likely better. This is because in situations where CLB is less likely achievable, the percentage deviation in LBF RBP between the real optimal solution and CLB may also be large. This fact can be demonstrated by case (g) of the exhaustive search experiment presented in Fig. 4. In this particular case, the RBP result of the heuristic solution is nearly 50% deviated from the CLB bound, but the percentage deviation between the optimal solution and CLB is also more than 40%. The actual quality of the heuristic solution is only 5% below the optimal solution.

We have conducted extensive experiments to further verify the quality of this greedy file allocation method. Here we report the results obtained from five experiments for (a) a 10-disk-100-movie system, (b) a 20-disk-200-movie system, (c) a 30-disk-300-movie system, (d) a 40-disk-400-movie system and (e) a 50-disk-500-movie system, respectively. For each experiment, we randomly generate 300,000 file replication instances. We report in Fig. 6 the best, average, and worst value of the percentage deviation found between the RBP result of the heuristic solution and the CLB bound for the various feasible file replication instances. We observe in Fig. 6 that, for the best scenario in (a), the heuristic solution is indistinguishable from CLB (0.02% deviated from CLB). In all the experiments, the average quality of heuristic solutions is below 15% deviated from CLB. Although for the worst scenario in (c) the deviation is as high as 102%, such a case is indeed very rare as can be confirmed from the density histogram of the percentage deviation plotted in Fig. 7. Moreover, it may not represent the true quality of the heuristic solution when benchmarked by the real optimal solution of the corresponding file replication instance as we have discussed.

The proposed greedy file allocation method has enabled the design of an evolutionary optimization approach in [15] to find near-optimal file assignment solutions computationally efficiently for the LBF system. An essential part of that approach is a divide-and-conquer strategy, where the entire solution

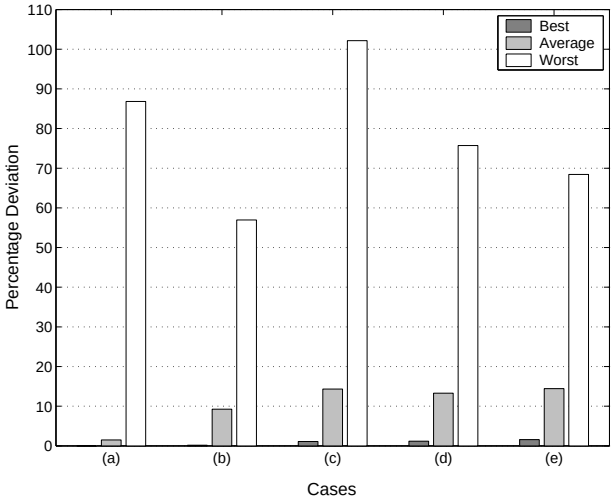

Fig. 6. Quality of the heuristic file allocation method benchmarked by CLB: (a) 10-disk-100-movie; (b) 20-disk-200-movie; (c) 30-disk-300-movie; (d) 40-disk-400-movie; (e) 50-disk-500-movie.

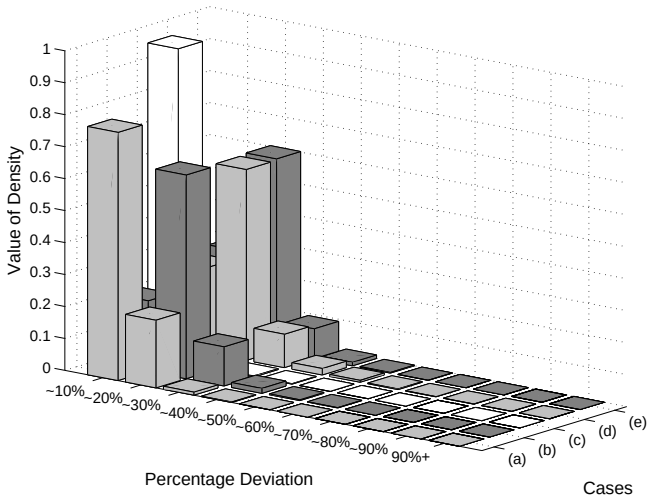

Fig. 7. Density histogram of the percentage deviation in LBF RBP between heuristic solutions and CLB bounds: (a) 10-disk-100-movie; (b) 20-disk-200-movie; (c) 30-disk-300-movie; (d) 40-disk-400-movie; (e) 50-disk-500-movie. $\sim x\%$ denotes the interval $[x\% - 10\%, x\%)$. $x\%+$ represents the interval $[x\%, \infty)$.

space of file assignments is divided into subspaces. Each subspace is an exclusive set of solutions sharing a common file replication instance. For each feasible file replication instance, the greedy file allocation method developed here is used there to decide a good quality heuristic solution within each subspace. In this way, the search space of the file assignment problem is significantly reduced. Numerical results in [15] showed that the near-optimal solution so obtained for the LBF system can improve the RBP performance by a factor of three in comparison with what is obtained from the SRT system.

VI. Approximate Analysis of CLB-LBF

Since performance evaluation by means of simulation generally requires excessive CPU time, we shall see later in this section that it is useful to develop fast analytical solutions to carry out the task. However, due to the complicated interactions between user requests for multi-copy movies and selections of disk resources to serve these requests, an exact solution for CLB-LBF is intractable. It was shown in [7] that the fixed-point approximation method [17] can be used to analyze the LBF system fast and sufficiently accurately. In this section, we shall see if the same methodology can be applied to derive an approximate analytical formula of reasonable accuracy for CLB-LBF.

Recall that when the VOD system attains CLB, for each c, $c \geq 1$, the traffic load of Type c movies is evenly distributed among $\binom{J}{c}$ combination groups of c disks enumerated in the set $\mathcal{D}$. Due to this homogeneity and the assumption that the request arrival process of any Type c movie is a Poisson random process, we can postulate that, at steady state, all disks in $\mathcal{D}$ will yield the same blocking probability. This allows us to choose an arbitrary disk, from which the overall RBP of the system can be derived.

Let $\xi^{(i)}$ be the stationary probability that the chosen disk is in state i, or in other words, it has i logical channels occupied, for $i = 0,\ 1,\ 2, \ldots,\ N$. Define $\vec{\xi} = \left(\xi^{(0)},\ \xi^{(1)}, \ldots,\ \xi^{(N)}\right)$.

For each of the $\binom{J-1}{c-1}$ combination groups of c disks of which the chosen disk is a member, the probability that, provided that the chosen disk is in state i upon the arrival of a request for a Type c movie, among the other $c-1$ disks in the combination group, $h-1$ out of the $c-1$ disks also have i channels occupied, and the remaining $c-h$ disks have more than i channels occupied is

$$P(h,i) = \binom{c-1}{h-1} \left(\xi^{(i)}\right)^{h-1} \Big(\sum_{k=i+1}^{N} \xi^{(k)} \Big)^{c-h} \tag{6}$$

for $i = 0,\ 1, \ldots,\ N-1$ and $h = 1,\ 2, \ldots,\ c$. Note that if $c = 1$, we simply set $P(h,i) = 1$.

Thus, when the chosen disk is in state i, its Type c movie request arrival rate is

$$y^{(i)}(c) = \binom{J-1}{c-1} \frac{\lambda \hat{p}_c}{\binom{J}{c}} \sum_{h=1}^{c} \frac{P(h,i)}{h} = \frac{c\lambda \hat{p}_c}{J} \sum_{h=1}^{c} \frac{P(h,i)}{h} \tag{7}$$

and its total request arrival rate due to all types of movies is

$$y^{(i)} = \sum_{c} y^{(i)}(c). \tag{8}$$

Let $\vec{y} = \left(y^{(0)},\ y^{(1)}, \ldots,\ y^{(N-1)}\right)$. Thus, (6), (7) and (8) define a function $f(\cdot)$ that can be used to obtain $\vec{y}$ from $\vec{\xi}$:

$$\vec{y} = f(\vec{\xi}). \tag{9}$$

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

$$\frac{1}{\bar{\mu}^{(i)}} = \frac{1}{y^{(i-1)}} \sum_{c} \frac{y^{(i-1)}(c)}{\hat{\mu}_c} \tag{10}$$

and $1/\hat{\mu}_c$ is given by (1).

From the steady-state equations of a birth-death process ([24], page 31), we have

$$\xi^{(i)} = \frac{N!}{i! \prod_{k=i}^{N-1} \frac{y^{(k)}}{\bar{\mu}^{(k+1)}}} \xi^{(N)}. \quad (11)$$

By normalization, we obtain

$$\sum_{i=0}^{N-1} \frac{N!}{i! \prod_{k=i}^{N-1} \frac{y^{(k)}}{\bar{\mu}^{(k+1)}}} \xi^{(N)} + \xi^{(N)} = 1. \quad (12)$$

Therefore, for the chosen disk, (1), (10), (11) and (12) define a function $g(\cdot)$ that can be used to obtain $\vec{\xi}$ from $\vec{y}$:

$$\vec{\xi} = g(\vec{y}). \quad (13)$$

The system of equations (9) and (13) composes the following fixed-point equations:

$$\vec{\xi} = g(f(\vec{\xi})). \quad (14)$$

Now assume that a request for a Type c movie that has been denied at the chosen disk is independent of other requests for this Type c movie that have been denied at other disks in its associated combination group of c disks. Solving (14) for $\vec{\xi}$, and using the fact that for a request of the Type c movie to be blocked it would need to be denied at all c disks in the combination group, we therefore deduce that the blocking probability $\hat{B}_c$ of requests for Type c movies is obtained by

$$\hat{B}_c = \left(\xi^{(N)}\right)^c. \quad (15)$$

The RBP of the LBF system at the state of CLB for a given file replication instance is then computed by

$$\text{RBP} = \sum_c \hat{p}_c \hat{B}_c \ . \quad (16)$$

A. Approximation validation

The fixed-point equations (14) can often be solved efficiently by the successive substitution method [27]. For the purpose of validation, we use the 20-disk example considered in Section III. While the simulation study typically takes more than 5,200 seconds on a 2.4 GHz Pentium 4 machine to estimate the RBP of CLB-LBF for this example, the analytical solution carries out the task within only 20 milliseconds.

The simulation estimates of RBP for CLB-LBF are presented in Fig. 8, with λ ranging from 440 requests per time unit to 500 requests per time unit at subsequent increments of 10. The corresponding analytical results are obtained from (16) that allow comparisons with the simulation estimates. We also present in Fig. 8 the RBP results of the heuristic file allocation instance for this particular example. The analytical results of the heuristic file allocation instance are obtained from the fixed-point approximation model of the LBF system provided in [7].

We see that the approximation results match the simulation estimates quite well in both cases. Although the approximation results slightly disagree with the simulation estimates, the distance from the approximation results to the simulation estimates is comparable between the heuristic solution and CLB. This makes the RBP result of CLB yet an effective lower bound in the LBF system even if the RBP is evaluated by the analytical means.

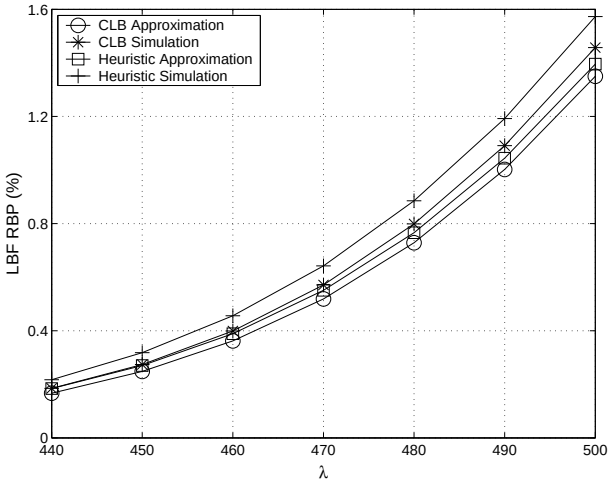

Fig. 8. Validation of the approximate analysis of CLB-LBF.

B. Application of the approximation model

Given the fact that CLB provides an effective lower bound on the RBP of the LBF system for any given file replication instance, we can utilize the approximation model of CLB-LBF to improve the runtime efficiency of the file assignment optimization program presented in [15]. Note that for brevity we will not repeat the details of the evolutionary optimization program in this paper, but provide a brief description as follows in order to demonstrate the usefulness of CLB-LBF in the context of file assignment optimization.

The evolutionary optimization program starts by creating an initial population of randomly generated file replication instances. We make sure that each member in the initial population has a heuristic solution and the RBP result of each heuristic solution is computed from the approximation model of LBF provided in [7]. During each subsequent generation of the evolutionary optimization process, genetic algorithms are adopted to perform a multi-directional stochastic search by means of selection, crossover, mutation and replacement, based on the parent solutions established from the population of the previous generation. We need to decide for each offspring solution so obtained if it can replace any of the parent solutions due to its smaller RBP result.

To this end, a first approach is to directly compute the RBP result for each offspring solution using the approximation model of LBF provided in [7]. On the other hand, a second approach is to compute the CLB bound of the corresponding file replication instance for each offspring solution using (16). Given that the RBP result of each parent solution is known, if the CLB bound of a particular offspring solution is larger than the RBP result of any parent solution, the actual RBP result of that offspring solution must also be larger than that of any parent solution. As a result, such an offspring solution can be safely discarded without the need of further computing its actual RBP result. Only if the CLB bound of the offspring solution is smaller than the RBP result of some parent solution, we then proceed with computing its actual RBP result to confirm if it has indeed a smaller RBP result than that parent and thus can replace that parent in the new population.

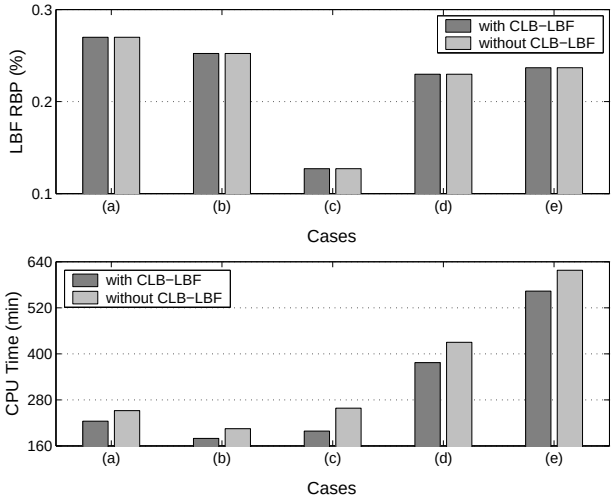

Fig. 9. Runtime efficiency comparison: (a) 10-disk-100-movie, 11% CPU time reduction; (b) 20-disk-200-movie, 12% CPU time reduction; (c) 30-disk-300-movie, 23% CPU time reduction; (d) 40-disk-400-movie, 12% CPU time reduction; (e) 50-disk-500-movie, 9% CPU time reduction.

For the purpose of comparing the runtime efficiency between these two approaches, we conduct experiments again for the five different test systems considered in Section V. For each experiment, we run the evolutionary optimization program for 3,000 generations with the incorporation of CLB-LBF (second approach) or without the incorporation of CLB-LBF (first approach). Results in Fig. 9 confirm that in all the experiments both approaches obtain exactly the same optimization results of LBF RBP. However, the second approach with the incorporation of CLB-LBF can improve the runtime efficiency for up to 23%. This is because the second approach requires much less computations of the LBF model though at the expense of an additional computation of CLB-LBF for each offspring solution. In fact, one computation of the LBF model requires much larger CPU time than that of CLB-LBF. For the 20-disk example considered in Section III, it took a CPU time of over 500 milliseconds on the 2.4 GHz Pentium 4 machine to compute the LBF RBP of the heuristic file allocation instance, but within only 20 milliseconds for the computation of CLB-LBF for the corresponding file replication instance. This is true since the size of the set of the fixed-point equations for CLB-LBF is merely $N + 1$, while the size of the set of the fixed-point equations for LBF is as large as $J(N + 1)$ [7].

VII. Conclusion

In a VOD system, a limited number of movies (usually with high popularity) are replicated over multiple disks to reduce the RBP of the system, while the spare disk storage space can be efficiently utilized. It is an interesting and unresolved issue to examine for a given file replication instance how to balance the movie traffic load and thus to minimize the RBP of the system if we allow an exhaustive resource selection scheme like LBF in serving user requests for multi-copy movies. To this end, we have proposed in this paper a conjecture by suggesting that such a system may only be load balanced when the traffic wishing to access movies of the same type is uniformly distributed among all combination groups of disks enumerated in the system for the associated movie type. At this state of CLB, the RBP of the LBF system is minimized. While a rigorous proof of this conjecture remains open, we have justified it in this paper through extensive experiments.

Our conjecture was inspired from the observation that the disk resource sharing of multi-copy movie traffic has great impact on the RBP of the LBF system, so that we intuitively expect that the maximal level of disk resource sharing of multi-copy movie traffic indicates the best allocation of multi-copy movie files. Although in practice CLB may not be always achievable, two important results have been motivated by this intuitive observation. Firstly, we have designed an efficient numerical index that measures quantitatively the quality of a file allocation instance on distribution of multi-copy movie traffic, as compared with the ideal file allocation instance that attains CLB. Secondly, we have devised a greedy file allocation method that aims for uniform resource sharing of multi-copy movie traffic and uniform distribution of single-copy movie traffic, and therefore results in a sufficiently good quality heuristic file allocation instance. Moreover, we have derived an analytical formula for the evaluation of CLB-LBF using the fixed-point approximation method. The precision of the approximation results is sufficient and enables a fast and effective way in estimating the lower bound on the RBP of the LBF system.

The results of this work can be applied to the design of a large scale VOD system. Specifically, they can be directly utilized by an evolutionary optimization program to find near-optimal file assignment solutions for the LBF system computationally efficiently [15].

Appendix I

Theorem 1: For a given file replication instance, the problem of finding a file allocation instance that achieves CLB (if it is achievable) is NP-hard.

Proof: A special case of the problem is where each of the M movie titles in the set $\mathcal{F}$ has exactly one file-copy. For this special case, the problem reduces to finding a file allocation instance that achieves DLB. Let $J = 2$. The problem further reduces to finding a partition of $\mathcal{F}$ into Φ_1 and Φ_2, such that $\sum_{m\in\Phi_1} A_m = \sum_{m\in\Phi_2} A_m$, where $\Phi_1 \cup \Phi_2 = \mathcal{F}$ and $\Phi_1 \cap \Phi_2 = \varnothing$. The decision version of the latter problem is equivalent to a weighted set partition problem ([28], page 223), which is NP-complete. This completes the proof. ■

Appendix II

See Fig. 10.

References

[1] W. D. Sincoskie, "System architecture for a large scale video on demand service," *Computer Networks and ISDN Systems*, vol. 22, no. 2, pp. 155–162, 1991.

[2] D. Deloddere, W. Verbiest, and H. Verhille, "Interactive video on demand," *IEEE Commun. Mag.*, vol. 32, no. 5, pp. 82–88, May 1994.

[3] M. Reisslein, K. W. Ross, and S. Shrestha, "Striping for interactive video: is it worth it?" in *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, vol. 2, Florence, Italy, Jun. 1999, pp. 635–640.

Procedure: Greedy File Allocation
Set $O_j = 0$ for all $j \in \mathcal{D}$;
Set $T_j = 0$ for all $j \in \mathcal{D}$;
Set $S_{ij} = 0$ for all $i \neq j \in \mathcal{D}$;
Set $\Omega_m = \varnothing$ for all $m \in \mathcal{F}$;
Do for each $m \in \mathcal{F}$, $n_m > 1$, in the non-increasing order according to $\frac{A_m}{n_m}$
  Set $\mathcal{D}^* = \mathcal{D}$;
  If $O_j + L_m > C$ for all $j \in \mathcal{D}^*$
    Return INFEASIBLE;
  Find $j \in \mathcal{D}^*$ with the smallest T_j;
  Increase O_j by L_m;
  Increase T_j by $\frac{A_m}{n_m}$;
  Set $\mathcal{D}^* = \mathcal{D}^* - j$;
  Set $\Omega_m = \Omega_m + j$;
  Set $k = 2$;
  Do while $k \leq n_m$
    If $O_i + L_m > C$ for all $i \in \mathcal{D}^*$
      Return INFEASIBLE;
    Find $i \in \mathcal{D}^*$ with the smallest S_{ij};
    Increase O_i by L_m;
    Increase T_i by $\frac{A_m}{n_m}$;
    Increase both $S_{ij'}$ and $S_{j'i}$ by $\frac{A_m}{n_m}$ for all $j' \in \Omega_m$;
    Set $\mathcal{D}^* = \mathcal{D}^* - i$;
    Set $\Omega_m = \Omega_m + i$;
    Increment k;
  End
End
Do for each $m \in \mathcal{F}$, $n_m = 1$, in the non-increasing according to A_m
  If $O_j + L_m > C$ for all $j \in \mathcal{D}$
    Return INFEASIBLE;
  Find $j \in \mathcal{D}$ with the smallest T_j;
  Increase O_j by L_m;
  Increase T_j by A_m;
  Set $\Omega_m = \Omega_m + j$;
End
Return FEASIBLE;

Fig. 10. Implementation of the greedy file allocation method for a given file replication instance $\mathbf{n}$.

[4] D. Sitaram and A. Dan, *Multimedia Servers : Applications, Environments and Design*. Morgan Kaufmann, 2000.

[5] 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, pp. 280–287, Jan. 1995.

[6] 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.

[7] J. Guo, S. Chan, E. W. M. Wong, M. Zukerman, P. G. Taylor, and K. S. Tang, "On blocking probability evaluation for video-on-demand systems," in *Proc. 18th Int. Teletraffic Congress*, vol. 5a, Berlin, Germany, Sep. 2003, pp. 211–220.

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

[9] D. N. Serpanos, L. Georgiadis, and T. Bouloutas, "MMPacking: a load and storage balancing algorithm for distributed multimedia servers," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 1, pp. 13–17, Feb. 1998.

[10] E. W. M. Wong and T. S. Yum, "Maximum free circuit routing in circuit-switched networks," in *Proc. IEEE INFOCOM 90*, vol. 3, San Francisco, CA, USA, Jun. 1990, pp. 934–937.

[11] 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.

[12] A. Birman, "Computing approximate blocking probabilities for a class of all-optical networks," *IEEE J. Select. Areas Commun.*, vol. 14, no. 5, pp. 852–857, Jun. 1996.

[13] E. W. M. Wong and S. C. H. Chan, "Performance modeling of video-on-demand systems in broadband networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 7, pp. 848–859, Jul. 2001.

[14] A. Sridharan and K. N. Sivarajan, "Blocking in all-optical networks," *IEEE/ACM Trans. Networking*, vol. 12, no. 2, pp. 384–397, Apr. 2004.

[15] J. Guo, "Two problems in stochastic service systems," Ph.D. dissertation, The University of Melbourne, Melbourne, Australia, 2006, available online: http://www.cubinlab.ee.mu.oz.au/∼guo/PhDthesis-JunGuo.pdf.

[16] J. Guo, P. G. Taylor, E. W. M. Wong, S. Chan, M. Zukerman, and K. S. Tang, "Performance benchmarks for an interactive video-on-demand system," in *Proc. IEEE ICC 04*, vol. 4, Paris, France, Jun. 2004, pp. 2189–2193.

[17] F. P. Kelly, "Loss networks," *The Annals of Applied Probability*, vol. 1, no. 3, pp. 319–378, Aug. 1991.

[18] M. Krunz, R. Sass, and H. Hughes, "Statistical characteristics and multiplexing of MPEG streams," in *Proc. IEEE INFOCOM 95*, vol. 2, Boston, MA, USA, Apr. 1995, pp. 455–462.

[19] R. Zimmermann and S. Ghandeharizadeh, "Highly available and heterogeneous continuous media storage systems," *IEEE Trans. Multimedia*, vol. 6, no. 6, pp. 886–896, Dec. 2004.

[20] 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, NY, USA, 2004, pp. 534–543.

[21] P. Branch, G. Egan, and B. Tonkin, "Modeling interactive behaviour of a video based multimedia system," in *Proc. IEEE ICC 99*, vol. 2, Vancouver, BC, Canada, Jun. 1999, pp. 978–982.

[22] G. K. Zipf, *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Cambridge, MA: Addison-Wesley, 1949.

[23] 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, USA, 1994, pp. 15–23.

[24] H. Akimaru and K. Kawashima, *Teletraffic: Theory and Applications*, 2nd ed. London: Springer-Verlag, 1999.

[25] G. S. Fishman, *Discrete-Event Simulation: Modeling, Programming, and Analysis*. New York: Springer-Verlag, 2001.

[26] S. K. Bose, *An Introduction to Queueing Systems*. New York: Kluwer Academic/Plenum Publishers, 2002.

[27] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains*. Princeton, N.J.: Princeton University Press, 1994.

[28] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: W. H. Freeman, 1979.