RC 20143 (08/01/95)
Computer Science

IBM Research Report

Consumption-Based Buffer Management for Maximizing System Throughputs of News-on-Demand Multimedia Systems

Kun-Lung Wu, Philip S. Yu

IBM Research Division
T.J. Watson Research Center
Yorktown Heights, New York

LIMITED DISTRIBUTION NOTICE

This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. It has been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies of the article (e.g., payment of royalties).

Research Division
Almaden · T.J. Watson · Tokyo · Zurich

Consumption-Based Buffer Management for Maximizing System Throughputs of News-on-Demand Multimedia Systems

Kun-Lung Wu and *Philip S. Yu*

IBM T.J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Abstract

In a news-on-demand multimedia server, multiple media streams are generally serviced in a cyclic fashion. Due to non-uniform playback rates and asynchronous arrivals of queries, there tends to be spare disk bandwidth in each service cycle. In this paper, we study the problem of dynamically using spare disk bandwidth and buffer to maximize the system throughput of a multimedia server. We introduce the concept of *minimizing buffer consumption* as the criterion to select an appropriate media stream to use the spare system resources. Buffer consumption measures not only the amount of buffer but also the amount of time such buffer space is occupied (i.e., the space-time product). Different alternatives to utilizing spare disk bandwidth are examined, including different rate-adjustable retrievals of an already activated stream and prefetching the next waiting stream. For rate-adjustable retrievals, we study *buffer consumption*-based and *remaining time*-based criteria for selecting an active stream to increase retrievals. Simulations are conducted to evaluate and compare different cases. The results show that (1) minimizing buffer consumption is the right criterion for maximizing the system throughput with spare disk bandwidth; (2) with sufficient buffer space or proper implementation, rate-adjustable retrievals of active streams perform better with minimal buffer consumption as a selection criterion than with minimal remaining time; (3) in general, prefetching a waiting stream incurs more buffer consumption, and thus is less effective than rate-adjustable retrievals of active streams in maximizing the system throughput.

Index Term: multimedia retrieval, buffer management, I/O, news on demand, rate-adjustable retrieval.

1 Introduction

With the fast advances in networking and computer technologies, all kinds of multimedia information, including video, audio, image and text, are being made available in digital forms. For example, one can use a Mosaic-like browser to easily access a vast amount of multimedia information around the globe through the ubiquitous World Wide Web on the Internet. Many projects are currently being established to explore the potential of electronic commerce, such as news on demand, electronic catalogs, electronic shopping, and video on demand. For an information/database provider in this information age, it is vitally important to provide effective multimedia supports.

To support multimedia data, especially video and audio, an information/database server needs to deal with several difficult issues. Firstly, video and audio data are delay-sensitive. Namely, once the playback of a stream begins, the server must guarantee that enough resources are allocated so that it is played back continuously and it meets the real-time requirement. Secondly, video and audio data (even in a compressed form) requires large amounts of system resources, primarily storage and bandwidth both in disk retrieval and in network transmission. Thirdly, a multimedia object may contain multiple components with very different playback rates, such as video, audio and text. The server must ensure that these multiple components be synchronized during playback.

For performance reasons, multiple multimedia streams are usually serviced simultaneously in a cyclic fashion [5, 3, 8, 2, 6]. During each service cycle, data from each stream are retrieved from disk into the buffer for continuous playback. An admission control algorithm is used to ensure that, once a stream is admitted, there be enough data in the buffer for the continuous playbacks of *all* active streams. Due to non-uniform playback rates and stream lengths and asynchronous arrivals of queries, there tends to be some spare disk bandwidth and buffer space left unused in each service cycle. Such spare disk bandwidth can be used to increase media retrievals, making it possible to finish the retrievals of an active stream sooner. This early completion of retrievals for a stream makes disk bandwidth available to admit a new request, reducing the average wait time and improving the system throughput. This is especially useful for news-on-demand multimedia applications, where the lengths of queries are usually not very long (e.g., less than 5 minutes) and the arrival pattern tends to be bursty.

In this paper, we study the problem of dynamically using spare disk bandwidth and buffer to maximize the system throughput of a news-on-demand multimedia server, given a fixed disk bandwidth and buffer. The critical issue here is what criterion should be used to select a candidate stream for which the additional disk bandwidth should be used. We examine different alternatives to the use of such spare system resources, including rate-adjustable retrievals of already active streams and prefetching a waiting stream. We observed that the best use of the spare disk bandwidth is to increase the retrievals of a media stream so that the buffer consumption caused by the additionally retrieved data is minimized. Here, buffer consumption is defined as the product of buffer space and the length of time this space is occupied. For example, if a media stream needs $500K$ bytes of buffer for 100 seconds, then the buffer consumption is $50M$ byte·seconds. Namely, buffer consumption measures both space and time of buffer usage.

For rate-adjustable retrieval schemes, spare disk bandwidth is used to increase the retrievals of an active stream during a service cycle. Therefore, the retrieval rate of the chosen stream becomes non-uniform from cycle to cycle. It behaves as if the retrieval rate were adjustable, and thus we call it *rate-adjustable retrieval.* We compare different criteria for selecting such a candidate stream to increase retrievals: *minimum remaining time* and *minimum buffer consumption.* Simulations are conducted to compare these criteria under two different implementations. The first one is a simple rate-adjustable (SRA) implementation, where once a candidate stream is selected additional retrievals are performed until there is no more buffer available or no more media data left on disk for this stream. The second one is called checking-based rate-adjustable (CRA) implementation, where extra retrievals will be terminated if the additionally retrieved data will occupy the buffer space needed for the first waiting request to be admitted. The results show that (1) indeed minimizing buffer consumption is the right strategy to maximize the system throughput using spare disk bandwidth; (2) with sufficient buffer space or proper implementation, such as CRA, the system throughput is better improved with buffer consumption than with remaining time as the selection criterion; and (3) increased retrievals of an active stream is more effective than prefetching in minimizing buffer consumption and thus maximizing the system throughput.

There are many studies focused on the storage and retrieval aspects of video and audio data, such as those presented in [1, 8, 2, 3, 5, 6, 7]. These studies generally were concerned with intelligent disk scheduling, or data allocation and placement of multimedia data. The issues of utilizing spare disk bandwidth to improve system throughput was first considered by Ng and Yang [4]. In [4], sharing buffer space among active streams and prefetching data for a waiting request were proposed to maximize both buffer and disk utilizations of a multimedia server. Prefetching was shown to improve system throughput.

The rest of the paper is as follows. Section 2 provides the preliminary. Section 3 introduces the concept of minimizing buffer consumption and describes rate-adjustable retrieval schemes of multimedia streams. The different selection criteria and retrieval implementations, including SRA, CRA and prefetching, are presented. Section 4 describes our simulations and analyzes the results.

2 Preliminary

In this section, we provide an overview of the multimedia server environment discussed in this paper. We also describe a simple admission control algorithm that derives the service cycle time T from the resource requirements in disk bandwidth and buffer space for continuous playback. We assume that multiple multimedia streams are retrieved in a cyclic fashion. We first consider fixed order disk scheduling and then generalize to the elevator-type sweeping scheme.

Assume that there are n multimedia streams that are currently active and being displayed. Let the playback rate of stream S_i be p_i and the amount of time retrieving S_i in each service cycle be t_i. Assuming the seek time and rotational latency from S_i to S_j is s_{ij}, then $t_1 + t_2 + \cdots + t_n + s_{12} + s_{23} + \cdots + s_{n1} \leq T$, where T is the length of a service cycle. If we assume the total switching time is s (i.e., $s = s_{12} + \cdots + s_{n1}$), then the disk utilization ρ is given by:

$$\rho = \frac{t_1 + t_2 + \cdots + t_n + s}{T}. \tag{1}$$

Within each service cycle, the total amount of data played back by S_i is $T * p_i$, and the amount of data retrieved for S_i is $t_i * R$, where R is the maximum disk reading rate. Thus, in order to meet the continuity requirement of S_i,

$$t_i * R \geq T * p_i. \tag{2}$$

From Equations 1 and 2, we can obtain a lower bound for T as follows:

$$T \geq \frac{s * R}{R * \rho - P}, \tag{3}$$

where $P = p_1 + \cdots + p_n$ is the total playback rate in a service cycle. Obviously, $R * \rho - P > 0$ must be true in Equation 3. In other words, the effective total retrieval rate must be greater than the total playback rate.

Now let us consider the buffer requirement B_i for stream S_i in a service cycle. Note that for continuous playback of stream S_i, enough data must be retrieved into the buffer in t_i time units and played back over a period of T time units. As a result, buffer space is needed for S_i. In order to reduce the total disk seek time, an elevator-type of sweeping scheme is used for disk scheduling. Hence, the reading order in two different service cycles can be different [8, 2]. In this paper, we assume that double buffering is used to accommodate this kind of disk scheduling[1]. Thus, for

[1]For a more optimized use of buffer space, readers are referred to [4].

Admission Control:

$t_{n+1} = p_{n+1} * T/R;$
$B_{n+1} = 2 * t_{n+1} * R;$
if $(\sum_{i=1}^{n+1} t_i + t_{sweep} + (n+1) * t_{latency} \leq T)$ and $(\sum_{i=1}^{n+1} B_i \leq B_{max})$
    add stream S_{n+1} into the current cycle;

Figure 1: A simple admission control algorithm

Notation	Definition (Default values)
T	service cycle time
S_i	media stream i
B_i	buffer needed in a service cycle for stream S_i
B_{max}	maximum buffer space available
p_i	playback rate of stream S_i
P	total playback rate in a service cycle
R	maximum disk retrieval rate
t_i	retrieval time for stream S_i in a service cycle
t_{sweep}	disk arm one-way sweeping time (32 msec)
$t_{latency}$	mean rotational latency (8.5 msec)
ρ	disk utilization
s	total switching time in a service cycle

Table 1: Notation

stream S_i, the buffer requirement is as follows:

$$B_i = 2 * t_i * R. \tag{4}$$

If the maximum buffer is B_{max}, it is necessary that the following equation must hold true.

$$\sum B_i \leq B_{max}. \tag{5}$$

With Eq. 3 and Eq. 5, we have an admission control algorithm. Assume that $S_1, \cdots, S_n$ are currently active and S_{n+1} is to be determined whether or not it can be admitted. Figure 1 shows a simple admission control algorithm based on Equations 3 and 5. In it, t_{sweep} is the disk arm sweeping time from the outmost cylinder to the innermost cylinder, or visa versa, and $t_{latency}$ is the mean rotational latency for a stream. Table 1 summarizes the notation used in this section.

Example 1: As an example, let us assume that, with enough buffer space, a multimedia server is designed to service 4 simultaneous video streams at 80% disk utilization level. Each video stream has a playback rate of $240K$ bytes/sec. Assume that R is $2000K$ bytes/sec, t_{sweep} is 32 milli seconds and $t_{latency}$ is 8.5 milli seconds. Then from Equation 3, we can compute the service cycle time T as follows:

$$T \geq \frac{(0.032 + 0.0085 * 4) * 2000K}{2000K * 0.8 - 240K * 4} = 0.206. \tag{6}$$

Let us assume that we choose $T = 0.206$ sec. If $B_{max} = 500K$ bytes, and streams $S_1, \cdots, S_4$ with $p_1, \cdots, p_4 = 240K$ bytes/sec are currently active, then we can determine if another video stream S_5 with $p_5 = 240K$ bytes/sec can be admitted into the service cycle. From Fig. 1, $t_5 = 240K * 0.206/2000K = 0.0247$ sec, and $B_5 = 2 * 0.0247 * 2000K = 98.8K$ bytes. Since $(0.0247 + 0.0247 * 4 + 0.032 + 0.0085 * 5) = 0.198$ $(< T)$ and $98.8K * 5 = 494K$ ($< B_{max}$), stream S_5 can be admitted into the service cycle. After S_5 is admitted, however, it cannot admit another stream S_6 with $p_6 = 240K$ bytes/sec since neither disk bandwidth nor buffer space is sufficient to continuously display S_6. Thus, S_6 will have to stay in the waiting queue. □

3 Rate-adjustable retrieval of multimedia streams

For a given disk bandwidth and buffer space, there tends to be spare disk bandwidth in each service cycle that cannot be used to add a new stream and thus remains wasted. If there are available buffer space, such unused disk bandwidth can be used to increase the retrievals of active streams, resulting in earlier completion of retrievals and making more disk bandwidth available sooner for the waiting streams. As a result, the system throughput can be improved. Note that the viewer completion time of a stream S_i does not change even if it is chosen to increase retrievals because p_i is fixed. However, the first waiting stream can be started earlier, resulting in less wait time.

Example 2: Following Example 1, when 5 streams are currently being serviced, the disk is not fully utilized. In each service cycle there is still $(0.206 - 0.198) = 0.008$ sec when the disk is idling. We can use this 0.008 sec to retrieve an additional $0.008 * 2000K = 16K$ bytes of data in a service cycle, provided that there is $16K$ bytes spare buffer space available. In Example 1, $B_{max} = 500K$. So, after admitting 5 streams, the available free buffer is only $6K$ bytes and is not sufficient for an additional retrieval of 0.008 secs. However, if B_{max} is bigger, say $1000K$ bytes, then we can select an active stream to increase the retrieval of $16K$ bytes and thus complete its retrievals sooner. Once completed, a new stream can be admitted because of the disk bandwidth made available sooner, thus improving the system throughput. □

3.1 Minimizing buffer consumption

There are different alternatives to using the spare disk bandwidth and buffer space, including increasing the retrieval of an active stream or prefetching a waiting stream. In this paper, we propose the concept of minimizing buffer consumption as the criterion for maximizing system throughput. Buffer consumption measures not only the amount of buffer space but also the amount of time such space is occupied. To maximize the system throughput with the spare disk bandwidth, the buffer consumption due to increased retrievals must be minimized. Namely, the space time product incurred due to such extra retrievals must be minimized. In general, the buffer space for the prefetched data may be occupied for longer than the entire media playing length. On the other hand, the space for the increased retrievals is occupied only for a fraction of its playing length since the candidate stream tends to be close to its completion. Thus, prefetching a waiting stream in general incurs larger buffer consumption. We first discuss some of the design issues of increasing retrievals of an active stream. More details about prefetching will be discussed in Section 3.3.

Ideally, we want to select the stream whose accelerated retrievals can be completed most quickly so that we can free up the disk bandwidth as soon as possible for a new request. Therefore, it is intuitive to select the one with the least remaining playing time to increase retrievals. However, the minimal remaining time criterion may not result in the fastest completion of retrieval with the additional bandwidth because the playback rate of different streams may differ significantly. Example 3 shows such a scenario.

Example 3: This example illustrates the importance of buffer consumption as a measure of buffer usage for rate-adjustable retrievals. Fig. 2 shows the remaining buffer usages of regular and rate-adjustable retrievals and the resulting extra buffer consumptions of two streams S_1 and S_2. S_1 has 5 remaining cycles to complete while S_2 has 6 remaining cycles. Assume that the buffer needed for normal playback in a cycle are 3 units for S_1 (Fig. 2(a)) and 1 unit for S_2 (Fig. 2(b)). Thus, there are 15 remaining data units to be retrieved and displayed for S_1 and 6 remaining units for S_2. If there are spare disk bandwidth and buffer to allow an extra retrieval of 2 buffer units per cycle, then the retrieval of stream S_1 can be finished in 3 cycles (see Fig. 2(c)) but that of S_2 can be finished in only 2 cycles (see Fig. 2(d)). Obviously, S_2 should be chosen since it completes retrievals first. However, if the minimal remaining time were used as the criterion, S_1 would have been chosen instead. If we had looked at the additional buffer consumption of each stream, S_2 would have been chosen (see Fig. 2 (e) and (f)), since its additional buffer consumption is smaller (12 vs 15). Note that Fig. 2 (e) and (f) only shows the extra buffer consumption. For example, only 3 extra buffer units are needed at the 4th service cycle and 0 extra is needed at the 5th cycle for S_1 in Fig. 2 (e). □

Note that at any given moment, there is at most one candidate stream that can be chosen to increase retrievals with spare resources. Alternating between different streams for rate-adjustable retrievals will result in larger buffer consumption. Thus, once a candidate stream is chosen, no other stream will be chosen again until the candidate stream completes retrievals.

The examples in Fig. 2 are presented in discrete time scenarios. In practice, the increase in

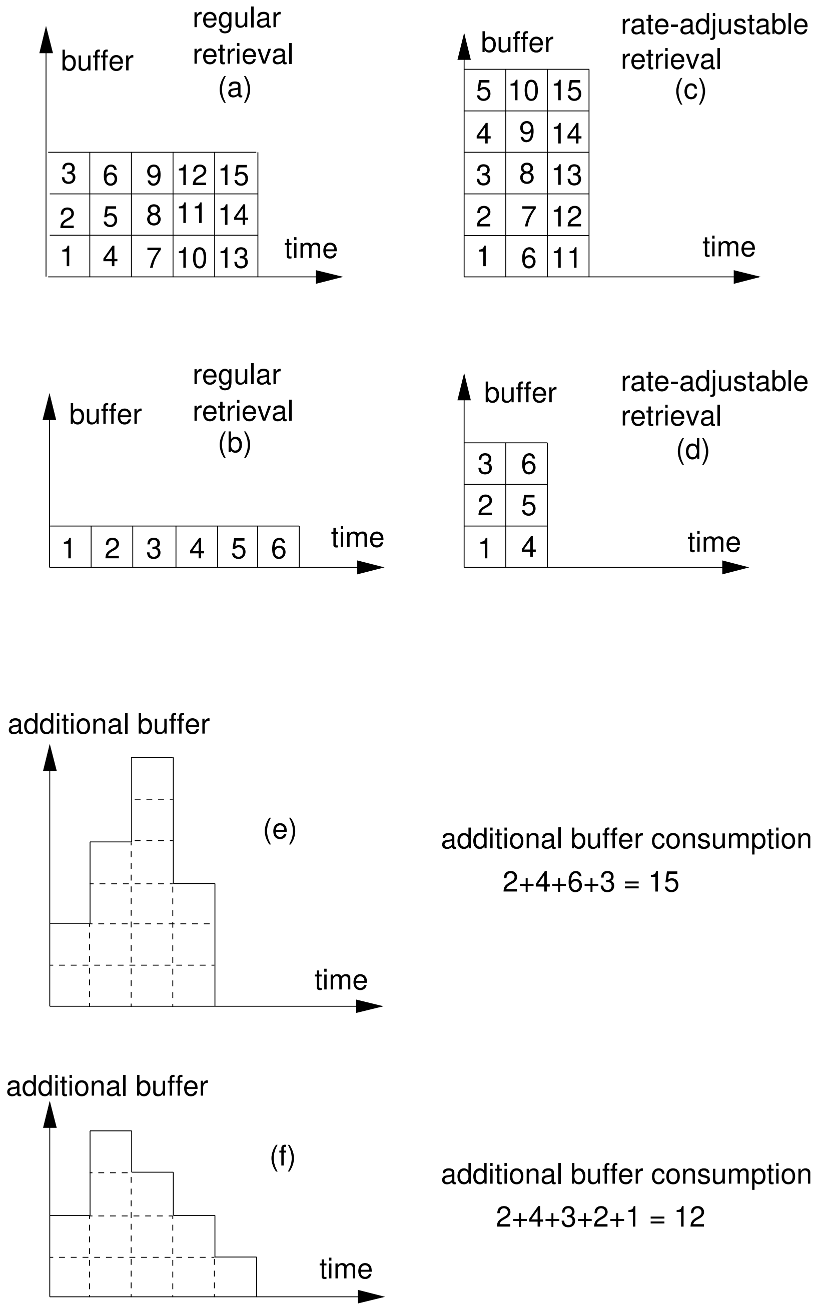

Figure 2: Buffer consumption as a selection criterion for rate-adjustable retrieval.

buffer consumption is a continuous time function. Moreover, the available disk bandwidth and buffer space may change over time. (The available disk bandwidth may change due to the arrival of a new stream that gets started. The server may run out of buffer space after a few additional retrievals.) If a VCR-type function is supported, such as pause/resume, then the next stream completion time may be uncertain. This can also affect the available disk bandwidth. As a result, it is generally not feasible to calculate the exact additional buffer consumption at the moment that a candidate stream is to be determined. Thus, we use two approximations in our simulations.

In order to verify the concept of minimizing buffer consumption as a criterion, we examine three selection criteria: *minimal remaining time*, *minimal remaining area* and *minimal additional area*. Here, *area* represents the space-time product of buffer usage and is used to compute buffer consumption. For the case of minimal remaining time, we simply select the stream S_i with the least remaining playing time r_i. For the case of minimal remaining area, the following formula is used:

$$0.5 * p_i * r_i * r_i. \tag{7}$$

This criterion simply computes the buffer consumption of the remaining data, assuming that all the remaining data are instantaneously fetched into the buffer and gradually played back in time r_i. For the case of minimal additional area, we assume that sufficient buffer space is available to support increased retrievals until completion. Assuming that the available spare disk bandwidth is a, and the early completion time due to increased retrievals is r_i', then

$$r_i' = \frac{p_i * r_i}{a + p_i}, \tag{8}$$

where $p_i * r_i$ is the remaining data to be fetched, and $a + p_i$ is the adjusted retrieval rate. The additional area is thus computed as follows:

$$0.5 * p_i * (r_i - r_i') * r_i. \tag{9}$$

Note that the additional data retrieved during the accelerated retrieving period r_i' is $p_i * (r_i - r_i')$. Therefore, the increased buffer consumption can be derived by $0.5 * r_i * p_i * (r_i - r_i')$.

For example, in Fig. 2(a) the remaining area and additional area are 37.5 and 15, respectively. In Fig. 2(b), the remaining area and additional area are 18 and 12, respectively. Thus, stream S_2 would be chosen with either remaining area or additional area as a criterion. However, if the increased retrieval stops after the first cycle (due to lack of buffer space), the actual buffer consumption for the additional retrieval is 8 and 9 for Fig. 2(a) and 2(b), respectively. In this case, minimal remaining time turns out to be a better criterion.

3.2 Rate-adjustable retrieval of an active stream

To compare the relative performance of using remaining time, remaining area, and additional area as the selection criterion, we run simulations on two different implementations of rate-adjustable retrieval of an active stream. The first one, called simple rate-adjustable (SRA) retrieval, simply continues additional retrievals of the candidate stream with all the available buffer space. It stops the increased retrievals only when there is no more data left to be retrieved or no more spare buffer is available.

Because of its aggressive nature in using available buffer space in SRA, the additional retrievals may even use up the required buffer for the next waiting stream. This is particularly the case when the buffer requirement of the first waiting request is greater than that of the most recently completed stream, and the just completed stream is not the candidate stream whose rate of retrieval has been increased. Because the buffer space occupied by the increased retrievals is not released until the completion of retrieval of the entire stream (see 2(e) and 2(f)), this can result in blocking of the next request and wasting disk bandwidth. If no additional retrieval had been performed, the waiting stream would have been admitted by the admission control algorithm. In order to avoid this problem, we examined another implementation, called checking-based rate-adjustable (CRA) retrieval. In CRA, a checking is performed to see whether or not the required buffer for the first waiting request will be used by the additional retrieval. If yes, then the additional retrieval is not performed.

3.3 Prefetching a waiting stream

Prefetching data into the buffer before a stream is admitted can result in lower demand in disk bandwidth later on when it is finally admitted [4]. This is because during the playback period, there are less data left for retrieving. Thus, once admitted, the retrieval rate of the prefetched stream can be smaller and more waiting streams may be admitted as a result. However, from the standpoint of buffer consumption, the prefetched data typically occupy buffer for longer than the total media playing time since they are not played back until the stream can be admitted. On the other hand, the additional data from rate-adjustable retrievals in general would only occupy buffer for a small fraction of the total playing time. For comparison purpose, we also implemented a simple prefetching scheme that uses the spare resource to prefetch as many data as possible for the first waiting request until it can be admitted with a smaller retrieval rate. (In [4], there is a more sophisticated prefetching scheme.)

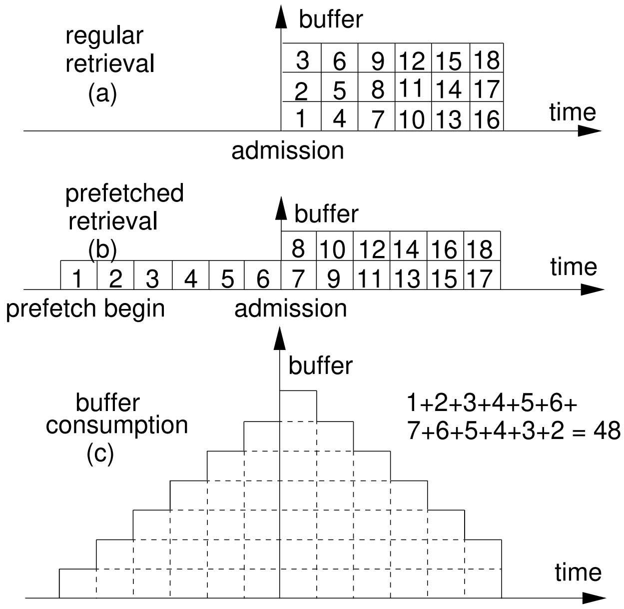

Figure 3: Buffer consumption for prefetching a waiting request.

As an example, Fig. 3 shows the buffer usage of prefetching. Unlike Fig. 2 that shows the remaining data, Fig. 3 shows the entire media stream. Assume that the stream in Fig. 3 requires 3 buffer units for 6 cycles. Also assume that prefetching is used to prefetch 1 buffer unit per cycle for 6 cycles before it can be admitted. Because of prefetching, the retrieval rate per cycle after admission is reduced from 3 to 2 (see Fig. 3(b)). The total buffer consumption for the prefetched case is 48 as shown in Fig. 3(c). So, the extra buffer consumption due to prefetching is $48 - 18 = 30$. Fig. 3(c) clearly illustrates that the buffer space for the prefetched data is occupied for longer than the entire media playing length. Hence, it tends to cause a large buffer consumption.

4 Performance evaluation via simulations

4.1 Simulation model

For evaluating the effectiveness of minimizing buffer consumption as a selection criterion to maximizing the system throughput with spare system resources, we implemented a discrete event driven simulator. We implemented the admission control algorithm presented in Fig. 1, and kept records of buffer usage by each stream. Double buffering was used to make possible of the implementation of an elevator-type of sweeping disk scheduling.

For our experiments, there are two classes of multimedia streams, one with a high playback rate and one with low playback rate. The high playback rate P_{high} is $240K$ bytes/sec and the low playback rate P_{low} is P_{high}/α. We varied α in our simulations. Examples of the high playback-rate class can be video streams, while those of the low playback-rate class can be audio streams. The service cycle time was fixed and was determined by Eq. 3, assuming $R = 2000K$ bytes/sec, $t_{sweep} = 32$ milli seconds, $t_{latency} = 8.5$ milli seconds, and the disk can service 4 high playback-rate streams simultaneously at 80% utilization similar to Example 1. The total playing time of a stream is uniformly distributed between $L/4$ and $7 * L/4$, where L is the mean playing length. In our simulations, $L = 60$ seconds. For each simulation run, 2000 streams are initially in the waiting queue. Of these 2000 streams, β fraction of them belongs to the class of high playback-rate streams and the rest are low playback-rate streams. The admission control algorithm, presented in Fig. 1, tries to admit as many streams as possible. After no more waiting stream can be admitted, additional retrieval or prefetching may be performed if there are spare disk bandwidth and free buffer space. The simulations measure the total completion time of these 2000 streams to demonstrate the improvement in system throughput.

4.2 Simulation results

We first present the performance of SRA using various selection criteria. Fig. 4 shows the total completion times and the corresponding disk utilizations of SRA using remaining area, remaining time, and additional area as a selection criterion. For comparison purposes, we also plotted the cases of simple admission control and prefetching the first waiting request. As indicated in the disk utilizations, there is spare disk bandwidth if we just use the simple admission control. Such spare disk bandwidth can be used to improve the system throughput, manifested by the reduced total completion times in Fig. 4. In general, with sufficient buffer space, using the minimal remaining area can get the best performance. Prefetching the waiting request generally is less effective in maximizing the system throughput as it requires more buffer consumption, compared with rate-adjustable retrieval of an active stream. Nevertheless, prefetching does improve the system performance. SRA with additional area as a selection criterion performs slightly better than with remaining time, but slightly worse than with remaining area. Note that both remaining area and additional area used in our simulations are approximations of the actual buffer consumption by the increased retrievals. Apparently, remaining area approximates the actual extra buffer consumption better.

Notice in Fig. 4 the performance of SRA may worsen when the buffer size is small, as compared with the simple admission control case. This is because the first waiting request may be blocked due to the lack of buffer space as a result of the increased retrievals. This phenomenon can be clearly observed through the disk utilization charts, where utilizations are smaller for the SRA

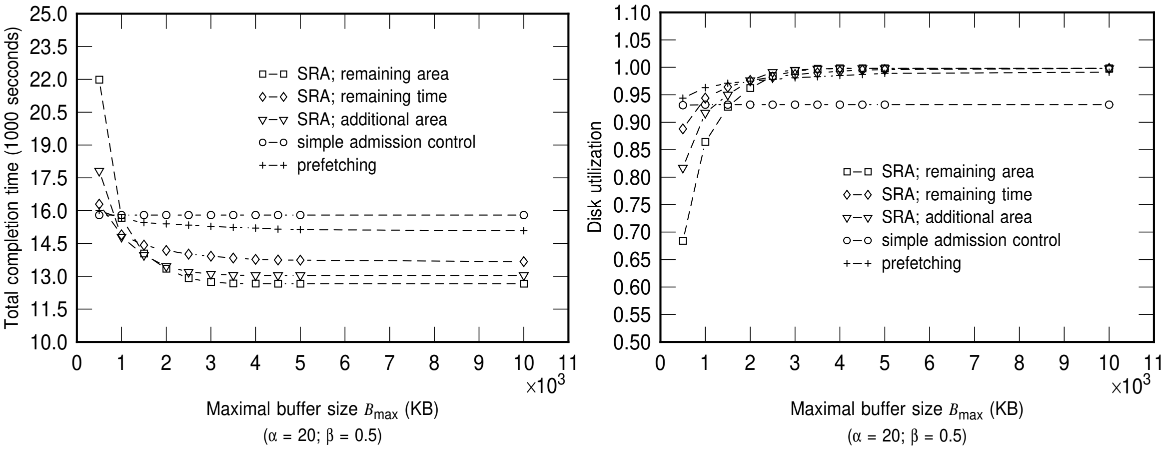

Figure 4: Impact of B_{max} on the performance of SRA.

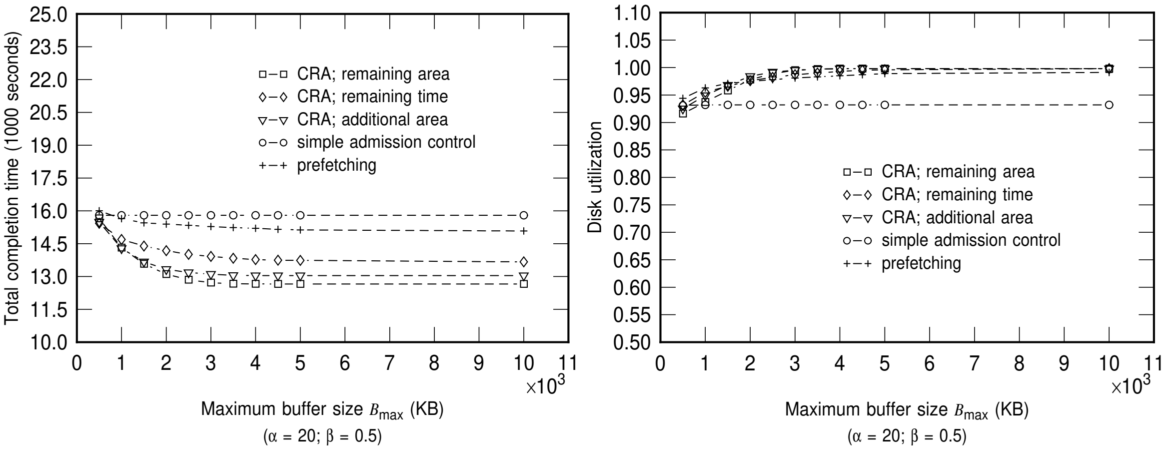

Figure 5: Impact of B_{max} on the performance of CRA.

cases when buffer size is small. With a small buffer size, the performance of SRA with remaining time as a criterion is better than with remaining area or additional area in Fig. 4. This is also due to the blocking effect. As mentioned before, the buffer space occupied by the additional retrieval is not released until the completion of retrieval of the entire stream. If buffer is small, it is likely that buffer space is not sufficient to support the completion of retrieval. In this case, the remaining time is a better criterion because it causes less buffer consumption for the additional retrievals.

The blocking problem can be avoided by the CRA implementation. Fig. 5 shows the total completion times and disk utilizations of the CRA implementation. In CRA, checking is performed before any of the increased retrievals can be allowed. As a result, all the CRA cases perform better than the simple admission control case, and disk utilizations are higher throughout the ranges of different buffer sizes.

In both Fig. 4 and 5, $\alpha = 20$ and $\beta = 0.5$. Namely, a high playback-rate stream is 20 times

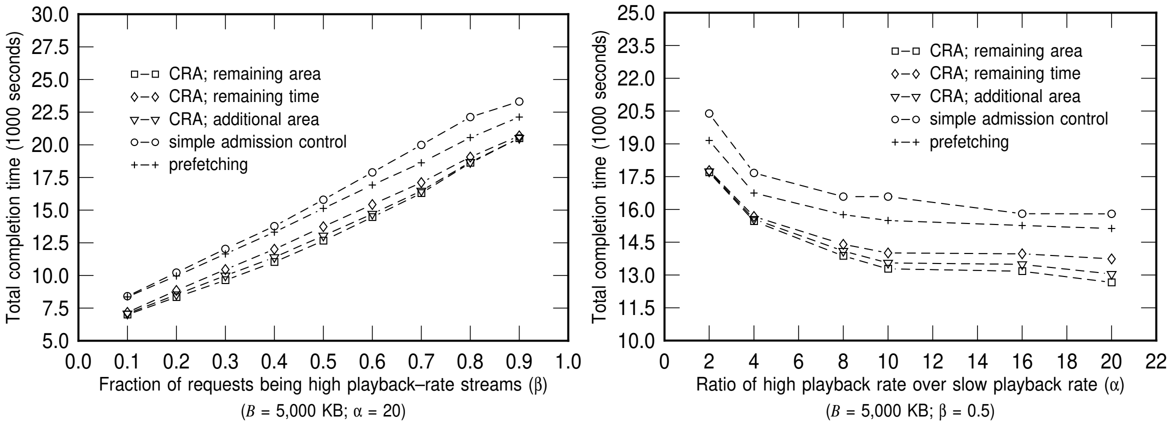

Figure 6: Impact of β and α on the performance of CRA.

faster in playback rate than a low playback-rate stream, and about 50% of the streams are high playback-rate ones and 50% of them are low playback-rate ones. To show the sensitivities of β and α, Fig. 6 shows the impacts of β and α on the total completion times. In general, if the playback rates of the media streams are less uniform (i.e., 50% of them are high playback-rate streams while 50% of them are low playback-rate ones), then there are bigger differences between the different CRA selection criteria. Moreover, if the playback rate ratio of a high playback-rate stream over a low playback-rate stream is larger, the differences are also bigger for different CRA selection criteria. Fig. 6 also confirms that minimizing buffer consumption is a right strategy to maximizing the system throughput using spare disk bandwidth.

5 Summary

Due to non-uniform playback rates and asynchronous arrivals of queries, there tends to be spare disk bandwidth in each service cycle in a news-on-demand multimedia server. In this paper, we presented rate-adjustable retrieval schemes to maximize the system throughput with such spare disk bandwidth. Namely, the spare disk bandwidth is used to increase retrievals of an active stream. We proposed the use of minimizing buffer consumption as a criterion to maximize the system throughput. Buffer consumption measures both buffer space and the time such space is occupied. Simulations were conducted to evaluate different criteria and implementations. The results confirm that minimizing buffer consumption indeed is the right criterion to use for maximizing the system throughput with spare disk bandwidth. Since actual buffer consumption by the additionally retrieved data is difficult or even infeasible to compute due to the dynamic changes in the environment, we examined two approximations. The results show that remaining area is a good approximation in computing buffer consumption in rate-adjustable retrieval of an active stream. Finally, we found that prefetching a waiting stream with the spare disk bandwidth is less effective than rate-adjustable retrieval of an active stream in maximizing system throughput because it incurs much more buffer consumption.

References

[1] D. Anderson, Y. Osawa, and R. Govindan. A file system for continuous media. *ACM Trans. on Computer Systems*, pages 311–337, 1992.

[2] J. Gemmell. Multimedia network file servers: Multi-channel delay sensitive data retrieval. In *Proc. of ACM Multimedia*, pages 243–250, 1993.

[3] J. Gemmell and S. Christodoulakis. Principles of delay-sensitive multimedia data storage and retrieval. *ACM Trans. on Information Systems*, 10(1):51–90, 1992.

[4] R. T. Ng and J. Yang. Maximizing buffer and disk utilizations for news on-demand. In *Proc. of the 20th VLDB Conference*, pages 451–462, 1994.

[5] P. V. Rangan and H. M. Vin. Designing file systems for digital video and audio. In *Proc. of ACM Symp. on Operating Systems Principles*, pages 69–79, 1991.

[6] A. L. N. Reddy and J. Wyllie. Disk scheduling in a multimedia I/O system. In *Proc. of ACM Multimedia*, pages 225–233, 1993.

[7] C. Yu et al. Efficient placement of audio data on optical disks for real-time applications. *Communications of the ACM*, 32(7):862–871, 1989.

[8] P. S. Yu, M.-S. Chen, and D. Kandlur. Grouped sweeping scheduling for DASD-based multimedia storage management. *Multimedia Systems*, pages 99–109, 1993.