Analyzing the Performance of Asynchronous Disk Arrays for Multimedia Retrieval

Harrick M. Vin, **Prashant Shenoy**, and **Sriram Rao**

Distributed Multimedia Computing Laboratory
Department of Computer Sciences
University of Texas at Austin
Taylor Hall 2.124
Austin, Texas 78712-1188

E-mail: {vin,shenoy,sriram}@cs.utexas.edu, Telephone: (512) 471-9732, Fax: (512) 471-8885

Citation: *In Proceedings of the ISMM International Conference on Distributed Multimedia Systems and Applications, Honolulu, Hawaii, Pages 14-17, August 1994*

This research was supported in part by the National Science Foundation (Research Initiation Award CCR-9409666), NASA, Mitsubishi Electric Research Laboratories (MERL), Sun Microsystems Inc., and the University of Texas at Austin.

Analyzing the Performance of Asynchronous Disk Arrays for Multimedia Retrieval

Harrick M. Vin, **Prashant Shenoy**, and **Sriram Rao**

Department of Computer Sciences, University of Texas at Austin
Taylor Hall 2.124, Austin, Texas 78712-1188
E-mail: {vin,shenoy,sriram}@cs.utexas.edu, Telephone: (512) 471-9732, Fax: (512) 471-8885

Abstract

Due to the periodic nature of media playback, a multimedia server can service multiple clients simultaneously by proceeding in rounds, retrieving a fixed number of media units (e.g., video frames) for each client during each round. In this paper, we analyze the performance of a conservative and an aggressive policy for retrieving media units from an asynchronous disk array. In the conservative servicing policy, a server completely retrieves all the required media units during a round prior to initiating the next round. A server employing the aggressive servicing policy, on the other hand, requires the disks to synchronize only over a finite sequence of rounds. We have carried out extensive simulations to measure the effectiveness of the conservative and the aggressive servicing policies on the utilization of asynchronous disk arrays. We present and analyze our simulation results.

Keywords: multimedia storage servers, asynchronous disk arrays, scheduling algorithms

1 Introduction

Recent advances in computing and communication technologies have made it feasible to design high performance, scalable *multimedia servers* which can provide a wide range of services to a large number of clients over high speed networks [1]. Due to the immensity of the sizes and the data transfer requirements of multimedia objects, such multimedia servers will undeniably be founded on large *disk arrays*. Disk arrays achieve high performance by servicing multiple I/O requests concurrently, as well as by utilizing several disks to service a single request in parallel. The performance of a disk array, however, is critically dependent on the distribution of the workload (i.e., the number of blocks to be retrieved from the array) among the disks. The higher the imbalance in the workload distribution, the lower is the throughput of the disk array. Consequently, designing scalable multimedia servers will require the development of techniques that achieve an equitable distribution of workload among all the disks in the array.

Most of the existing analytical as well as simulation models for analyzing the performance of synchronous and asynchronous arrays have presumed conventional workload (i.e., successive retrieval requests generated by the same process are assumed to be independent of each other) [2, 3, 4, 5, 6, 7]. On the contrary, since digital audio and video playback is inherently sequential, the set of disks accessed by a retrieval request is related to the one accessed by the previous request from the same client. Additionally, since media streams may be encoded using variable rate compression techniques, the amount of information retrieved from the server during each round may vary from one client to another. Although several media compression algorithms have been proposed in the recent past [8], very little work has been done in developing techniques for efficient retrieval of compressed multimedia objects from disk arrays. Techniques for improving the throughput of asynchronous disk arrays for multimedia retrieval is the subject matter of this paper.

2 Multi-Disk Multimedia Servers

Digitization of audio yields a sequence of samples, and that of video yields a sequence of frames. We refer to a continuously recorded sequence of audio samples or video frames as a *strand*. A multimedia server can organize the storage of such media strands on a disk array in terms of fixed size *media blocks* [9]. A media block is the unit of data interleaving among disks, and denotes the maximum amount of logically contiguous data that is stored on a single disk (this has also been referred to as the *striping unit* in the literature [2]). Successive media blocks are stored on consecutive disks in the array.

Due to the periodic nature of media playback, a multimedia server can service multiple clients simultaneously by proceeding in *rounds*. During each round, the server may retrieve a fixed number of media units (e.g., video frames) for each client by accessing a sequence media blocks from a set of consecutive disks in the array. A round is said to be complete when all the required media blocks have been retrieved from the array.

If media strands being requested by the clients are encoded using variable bit rate (VBR) compression techniques, the number of media blocks that may contain the required number of media units may vary from one round to another. Furthermore, since the arrival of client requests at the server are likely to be independent of each other, the set of disks accessed by different clients during a round may be completely uncorrelated. Hence, the number of media blocks to be accessed during a round may vary from one disk to another. Furthermore, since the time spent in retrieving a sequence of media blocks from a disk (referred to as the *service time*) is dependent on the number of media blocks being accessed as well as their relative placement on the disk, service times may also vary from one disk to another.

In such a scenario, a multimedia server may employ various policies for initiating the retrieval of media blocks of successive rounds. In the simplest case, a server may require all the disks to synchronize at the end of each round. Specifically, a multimedia server that employs such a *conservative* servicing policy requires the retrieval of all media blocks to be accessed during a round prior to initiating the next round. This will require a lightly loaded disk to remain idle until a heavily loaded disk completes the retrieval during each round, thereby limiting the performance of the array to that of the most heavily loaded disk. At the other extreme, the server may exploit the sequential nature of media playback and may initiate the retrieval of media information scheduled to be accessed in future rounds from a disk as soon as it becomes idle. The improvement in the utilization of the disk array yielded by such an *aggressive* policy may, however, occur at the expense of increased buffer space requirement at the server.

To precisely quantify the effects of the conservative and aggressive servicing policies, consider a multimedia server that is servicing n clients, each retrieving a video strand (say $S_1, S_2, ..., S_n$, respectively). Let $f_1, f_2, ..., f_n$ denote the number of frames of strands $S_1, S_2, ..., S_n$ retrieved during each round. Assume that each media strand is stored on a disk array of size N.

2.1 Conservative Servicing Policy

A multimedia server employing the conservative servicing policy requires media blocks from all the disks to be completely retrieved prior to initiating the next round. Specifically, if τ_i^r denotes the time spent in retrieving media blocks from disk i during round r, then the server may initiate the retrieval of the set of blocks to be accessed during round $(r+1)$ only after:

$$\tau_{max}^r = \max_{i \in [1,N]} \tau_i^r$$

units of time have elapsed. Consequently, all the disks with $\tau_i^r < \tau_{max}^r$ will be required to remain idle until the disk with the maximum workload completes the retrieval. Formally, if τ_{min}^r denotes the minimum service time for a disk in round r, then the minimum utilization of a disk (denoted by $\mathcal{U}_{min}^C$) during round r can be defined as:

$$\mathcal{U}_{min}^C = \frac{\tau_{min}^r}{\tau_{max}^r} \tag{1}$$

Similarly, the average utilization of the disk array bandwidth during round r can be defined as:

$$\mathcal{U}_{avg}^C = \frac{1}{\tau_{max}^r} * \left(\frac{1}{N} * \sum_{i=1}^{N} \tau_i^r \right) \tag{2}$$

In practice, the difference between τ_{max}^r and τ_{min}^r may depend on the correlation between the set of disks accessed by each client during round r. In fact, the higher the imbalance in the number of media blocks being accessed from the disks, the greater is the difference between τ_{max}^r and τ_i^r, and hence, lower is the utilization of disk array bandwidth. Hence, maximizing the utilization of the disk array requires that the number of media blocks being accessed during rounds to be equitably distributed among all the disks in the array.

2.2 Aggressive Servicing Policy

In order to improve the utilization of the disk array, a server may employ an aggressive servicing policy in which the retrieval of blocks required to be accessed from disk i during round $(r+1)$ can be initiated as soon as disk i completes the retrieval of media blocks required for round r. In addition to improving the utilization of the disk array, such an aggressive policy enables the server to balance the cumulative service times of disks in the array over a sequence of rounds.

To illustrate, consider a multimedia server that is retrieving n media strands, each of which is encoded using a constant bit rate (CBR) compression technique. Let k denote the number of media blocks accessed for each strand during each round. In such a scenario, due to the sequential nature of media playback, disks which are k-apart may retrieve exactly the same number of media blocks during successive rounds. Specifically, if $b_1^r, b_2^r, ..., b_N^r$ denote the number of media blocks required to be accessed from the disks during round r, then

$$\forall j \in [1, N]: \quad b_j^r = b_{j'}^{r+1}$$

where $j' = (j + k) \bmod N$. Consequently, although the workload may not be equitably distributed among all the disks during each round, the total number of blocks accessed from each disk during a sequence of consecutive rounds will be identical.

If, however, media strands are encoded using variable bit rate encoding schemes, the number of media blocks that may contain f_i frames of strand S_i may vary from one round to another. Consequently, contrary to the previous scenario, the cumulative workload imposed by all the strands may not be equitably distributed across the disk array within a finite number of rounds. The imbalance in the aggregate number of media blocks retrieved from various disks in the array over a finite number of rounds, however, is likely to be significantly smaller than the accumulation of the imbalances in each of the rounds. Thus, if $\mathcal{T}_i(r) = \sum_{j=1}^{r} \tau_i^j$ denotes the cumulative service time incurred while accessing media blocks from disk i over r rounds, and if $\mathcal{T}_{max}(r) = \max_{i \in [1,N]} \mathcal{T}_i(r)$ and $\mathcal{T}_{min}(r) = \min_{i \in [1,N]} \mathcal{T}_i(r)$, then we get:

$$\forall r > 1: \quad \mathcal{T}_{max}(r) - \mathcal{T}_{min}(r) \ll \sum_{j=1}^{r} \left(\tau_{max}^j - \tau_{min}^j \right) \tag{3}$$

Furthermore, we can define the minimum and the average utilization yielded by the aggressive servicing policy as:

$$\mathcal{U}_{min}^A(r) = \frac{\mathcal{T}_{min}(r)}{\mathcal{T}_{max}(r)} \tag{4}$$

Disk capacity	2 GBytes
Number of disks in the array	16
Bytes per sector	512 KB
Sector per track	99
Tracks per cylinder	21
Cylinders per disk	2627
Minimum seek time	1.7 ms
Maximum seek time	22.5 ms
Maximum rotational latency	11.1 ms

Table 1 : Disk Parameters of the Seagate-Elite3 disk assumed in the simulations

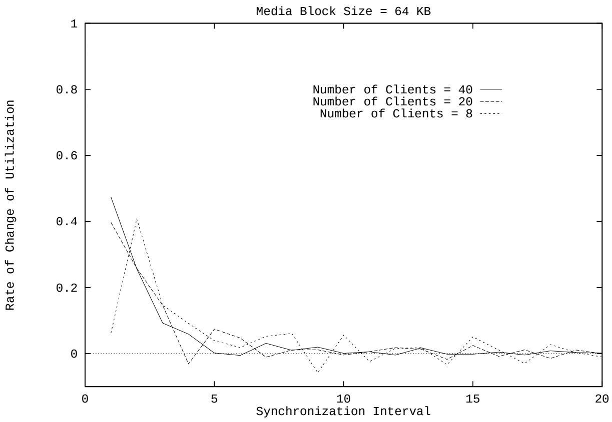

Figure 1 : Variation in the rate of change of utilization with increase in the synchronization interval

and

$$\mathcal{U}_{avg}^{A}(r) = \frac{1}{\mathcal{T}_{max}(r)} * \left(\frac{1}{N} * \sum_{i=1}^{N} \mathcal{T}_i(r)\right) \qquad (5)$$

Hence, due to Equation (3), the utilization yielded by the aggressive policy is guaranteed to be significantly higher than its conservative counterpart.

Observe that the improvement in the utilization of the disk array, however, is at the expense of increased buffer space requirement at the server. Specifically, if B_i denotes the maximum number of media blocks that may be required to be accessed for strand S_i during any round, then, whereas the conservative servicing policy requires buffer space to hold only $2 * B_i$ blocks of strand S_i at the server, the aggressive servicing policy which synchronizes all the disks at the end of r rounds will require buffer space to accommodate at least $2 * r * B_i$ blocks of strand S_i.

3 Empirical Evaluation

To measure the effectiveness of the conservative and the aggressive servicing policies, we have carried out extensive simulations in an environment consisting of an asynchronous disk array with 16 disks. For the purposes of simulations, each media strand is assumed to be striped across the entire array. The characteristics of each disk are shown in Table 1. A poisson arrival of client requests is assumed (i.e., the arrivals of clients are independent of each other). Furthermore, each video strand requested by clients is assumed to be encoded using a Variable Bit Rate (VBR) compression technique. The conventional SCAN disk scheduling algorithm is employed for retrieving media blocks from a disk during each round. The playback rate of each video strand is assumed to be 30 frames/sec, and 15 frames of each video strand are retrieved during each round. Hence, rounds are repeated every 0.5 seconds. The trace data for frame size variation yielded by VBR encoding techniques was obtained from Bellcore, University of California at Berkeley, and Columbia University.

Recall that the conservative servicing policy requires all the disks in the array to synchronize at the end of each round. The aggressive servicing policy, on the other hand, attempts to improve the utilization of the disk array by synchronizing the retrieval process only after every r ($r \geq 1$) rounds. Clearly, the higher the value of r, the greater is the likelihood of balancing workload across all the disks in the array, and hence, the greater is the utilization. Figure 1 depicts the rate of change of in the utilization with increasing values of r. It illustrates that the improvement in the utilization of the disk array saturates once the synchronization interval exceeds about 10 rounds.

In figure 2, we analyze the effects of increasing the number of clients as well as the media block sizes on the disk array utilization yielded by the conservative and aggressive policies. It illustrates that the average utilization yielded by the conservative servicing policy improves initially with increase in number of clients, but saturates at about 75%. On the other hand, for all values of $r \geq 5$, the aggressive policy yields a utilization of about 95%, and is independent of the number of clients being serviced. Figure 2 also illustrates that the utilization yielded by the conservative servicing policy decreases with increase in the media block size. This is because, for higher media block sizes, smaller number of disks may be accessed by each client during a round, thereby increasing the potential for imbalance in workload distribution. The utilization yielded by the aggressive servicing policy, on the contrary, is found to be independent of the choice of media block size.

As a final caveat, observe that if a server contains enough buffer space to implement the aggressive policy which synchronizes the disks in the array at the end of r rounds of duration $\mathcal{D}$ units each, it can also implement the conservative policy with the round duration of $r * \mathcal{D}$. In such a scenario, since the conservative policy with round duration of $r * \mathcal{D}$ units enables the server to retrieve all the media blocks from disk during a single scan (as compared to the r scans required by the aggressive policy), its throughput (measured in terms of bytes/sec) is likely to be slightly higher (see Figure 3). However, since new clients can be admitted for service by a

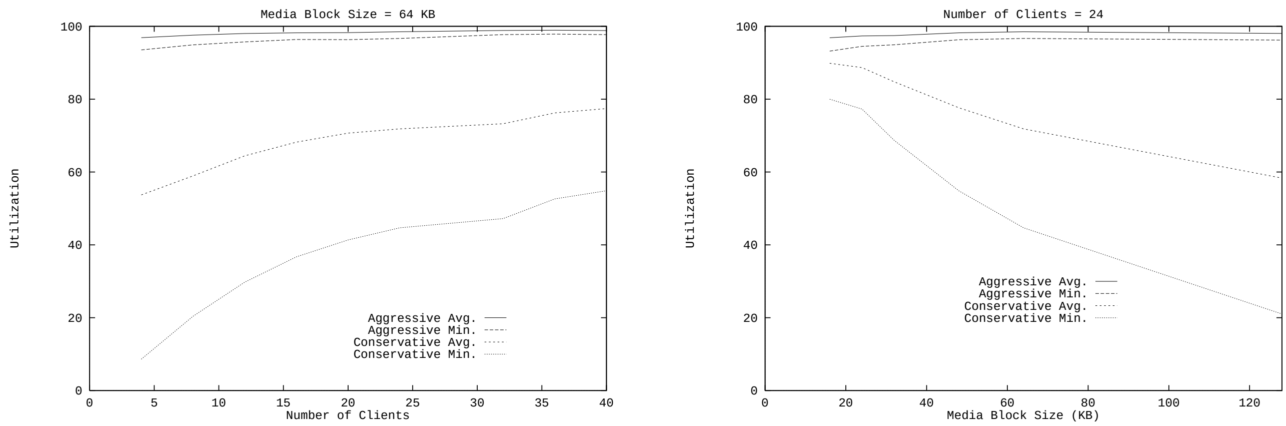

Figure 2 : Effects of increasing the number of clients and the media block size on the utilization of the disk array

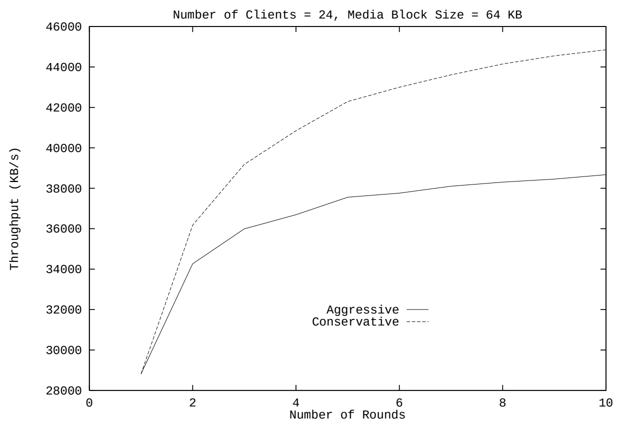

Figure 3 : Variation in Throughput with increase in Synchronization Interval

multimedia server only at round boundaries [9], increasing the duration of a round significantly increases the response time. Whereas such an increase in response time may be tolerable for multimedia servers providing video-on-demand services, it may be intolerable for a wide range of applications in a distributed multimedia computing environment (e.g., initiating playback of a video clip in a hypermedia document). Consequently, in practice, duration of a round may be determined by the response time considerations of target applications, and the length of the synchronization interval may be determined by the buffer space availability at the server.

4 Concluding Remarks

In this paper, we have analyzed the performance of a conservative and an aggressive policy for retrieving media blocks from an asynchronous disk array. We have demonstrated that the aggressive policy, for various combinations of number of clients as well as media block sizes, consistently achieves a higher utilization of disk array as compared to its conservative counterpart. The improvement in the utilization of the disk array yielded by such a policy, however, is at the expense of increased buffer space requirement at the server. A prototype multimedia server, based on the servicing policies presented in this paper, is being implemented at the UT Austin Distributed Multimedia Computing Laboratory.

REFERENCES

[1] G. Miller, G. Baber, and M. Gilliland. News on-demand for multimedia networks. In *Proceedings of ACM Multimedia'93, Anaheim, CA*, pages 383–392, August 1993.

[2] P. Chen and D. Patterson. Maximizing performance in a striped disk array. *Proceedings of ACM SIGARCH Conference on Computer Architecture, Seattle, WA*, pages 322–331, May 1990.

[3] H. Garcia-Molina and K. Salem. Disk stripping. *International Conference on Data Engineering*, pages 336–342, February 1986.

[4] R.H. Katz, G. Gibson, and D. Patterson. Disk system architectures for high performance computing. *Proceedings of the IEEE*, 77(12):1842–1858, December 1989.

[5] M. Kim. Synchronized disk interleaving. *IEEE Transactions on Computers*, C-35(11):978–988, November 1986.

[6] E.K. Lee and R.H. Katz. An analytic performance model for disk arrays. In *Proceedings of the 1993 ACM SIGMETRICS*, pages 98–109, May 1993.

[7] M. Linvy, S. Khoshafian, and H. Boral. Multi-disk management algorithms. *Proceedings of Eleventh Symposium on Operating Systems Principles*, pages 69–76, November 1987.

[8] M. Rabbani and P. Jones. Digital image compression techniques. *SPIE Optical Engineering Press*, 1991.

[9] Harrick M. Vin and P. Venkat Rangan. Designing a multiuser hdtv storage server. *IEEE Journal on Selected Areas in Communications*, 11(1):153–164, January 1993.