Proposing Block Rearrangement for Video-On-Demand Servers

Nabil J. Sarhan *Chita R. Das*

Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802
Phone: (814) 865-0194
E-mail: {sarhan,das}@cse.psu.edu

Abstract

Video-on-demand (*VOD*) has grown dramatically in popularity, especially in the domains of education, business, and entertainment. Therefore, the investigation of various alternatives to improve the performance of *VOD* servers has become a major research focus. The reduction of the disk access time through intelligent data placement strategies is one such avenue and is the theme of this paper.

Movie rental patterns indicate that accesses to movies are highly localized, with only a small number of movies receiving most of the hits. In this paper, we exploit this locality and propose an adaptive rearrangement of the blocks on each disk within the server. With this approach, the blocks of the movies with comparable access frequencies are kept closer to each other. We analyze two rearrangement schemes, called *centered* and *sequential*. In the centered layout, blocks are placed according to their access patterns starting with the most popular movie at the center. The sequential layout places movies in the order of their popularity starting with the most popular movie at the edge of the disk.

We compare and evaluate the effectiveness of these layouts with respect to arbitrary layouts through an intensive simulation study. The simulation results indicate that significant disk improvements can be attained by adopting the proposed schemes, and that the centered layout is the best performer.

1 Introduction

Recent advances in storage and communication technologies have spurred a strong interest in *video-on-demand* (VOD) systems. In contrast with broadcast-based systems, VOD enables customers to choose what they want to watch when they want it. VOD also allows customers to apply VCR-like operations such as pause, resume, fast forward, and fast rewind. Motion picture studios and cable companies, realizing the importance of providing customized services, have started looking for ways to utilize the computer technology for delivering VOD services. In addition to entertaining the needs of their customers, these services can create a market for their huge on-the-shelf media content.

In this paper, we consider the design of *video-on-demand* (*VOD*) servers. The major performance metric of these servers is the number of concurrent streams that they can serve while maintaining a “reasonable” quality of service (QoS). Unfortunately, this number is highly constrained by the real-time playback and the high transfer rates requirements. Hence, improving the performance of *VOD* servers has been the objective of numerous research studies. Common ways to accomplish that include efficient striping [25, 3], replication [6, 3], scheduling [21, 18], block allocation [20, 6], and caching [4, 5, 15, 16, 17]. The objective of this study is to enhance the performance of *VOD* servers by using an *adaptive block rearrangement*.

Disks are identified as a major performance bottleneck in *VOD* servers because of their relatively high access times and the relatively less effective caching in these servers. Indeed, striping data among multiple disks increases the I/O bandwidth, but the service time of each of the disks remains a limiting factor in the overall performance. The disk service time consists of a head-positioning time and a data transfer time. The seek time and the rotational time contribute to the head-positioning time, and both depend primarily on the disk technology. The seek time, however, depends not only on the disk technology, but also on the request pattern, the disk scheduling algorithm, and the block arrangement.

Rental patterns indicate that only a small number of movies receive most of the hits [2]. We propose an *adaptive block rearrangement* policy that reduces disk seek time by exploiting this locality. With this policy, the server monitors the access to movies and rearranges the blocks on each disk accordingly, whenever that is needed, and the server is not too busy. The policy ensures a faster disk access by placing the blocks of the movies with comparable access frequencies closer to each other.

We present two algorithms for the adaptive rearrangement: *centered layout* and *sequential layout*. In the *centered-layout* rearrangement, blocks are placed based on a variation of the organ pipe heuristic [13]. The organ pipe heuristic places the most frequently accessed data in the center of the disk. The next most frequently accessed data is placed to either side of the center, and the process continues until the least frequently accessed data has been placed at or near the edges of the disk. The *sequential-layout* rearrangement places the blocks in the order of their popularity, starting at the edge of the disk, with the most popular movie being stored first.

We compare, through an intensive simulation study, the centered layout, the sequential layout, and the possible layouts in the absence of any rearrangement. We have conducted the simulation study on a disk array of *Quantum Atlas10K*. We have used the *DiskSim* simulator [9] to simulate the disk unit of the disk array. In order to ensure high accuracy in the disk simulation, we have used actual measurements to determine the seek time for a given seek distance. We use random layouts to characterize placements in systems where no rearrangement is maintained, and we study two degrees of randomness. In a random layout, movies are placed randomly.

We analyze the following performance parameters in this paper: the seek distance, the seek time, and the disk access time. We also study the impacts on these parameters of the size of the disk array, the stripe unit, and the movie length. The simulation results show that an adaptive block rearrangement based on either of the presented algorithms can yield significant gains in disk performance. The benefits of the centered layout relative to random layouts can be summarized as follows. The improvements generally increase with the size of the disk array, provided that the movies are striped across all disks. For example, in a disk array of 4 disks, the centered-layout rearrangement reduces the mean seek distance by about 50%. This translates to an improvement of about 20% to 25% in the mean seek time and an improvement of more than 13% in the mean disk access time. In contrast, for a disk array of 20 disks, the mean seek distance is minimized by more than 60%, which shortens the mean seek time by more than 30% and the mean disk access time by about 15%. The results also indicate that the improvements increase as the stripe unit decreases, and as the movie length decreases. Similarly, the sequential layout performs better than random layouts but worse than the centered. In particular, the mean seek distance is about 25% to 45% longer than that of the centered.

The rest of the paper is organized as follows. In Section 2, we give an overview of the related research work. Then, we discuss the block rearrangement policies in Section 3, and the performance evaluation in Section 4. In Section 5, we present and analyze the simulation results. Finally, conclusions are drawn in the last section.

2 Related Work

Block rearrangement was proposed for improving disk performance in general-purpose systems [27, 22, 26, 1]. It was shown that the organ pipe heuristic places data optimally if data references are derived from an independent random process with a known fixed distribution [12, 28]. Variations of this heuristic were shown to be effective in practical systems. The organ pipe heuristic has been used in the cylinder shuffling technique [27, 22]. This technique monitors the access of disk cylinders over some period of time and then reorders the cylinders based on the measured frequencies. A reduction of 45% to 50% in the mean seek time was reported in [27], and a reduction of up to 10% in the disk service time was reported in [22]. Moreover, the organ pipe heuristic has been employed by iPcress (an experimental filesystem) [26]. The iPcress filesystem observes the file access patterns and moves the files with high "temperatures" near the center of the disk. In this context, "temperature" is the frequency of access divided by the file size. More recently, an adaptive block rearrangement technique [1] based on this heuristic was proposed. This technique copies frequently referenced blocks from their original locations to a reserved space near the center of the disk. This scheme was implemented by modifying a UNIX device driver. Simulation results showed that seek times can be substantially reduced by copying only a small number of blocks to a reserved space in the middle of the disk.

To the knowledge of the authors, the effectiveness of block rearrangement has not been investigated for *VOD* servers, where the block access size is much larger than that of traditional filesystems. The constrained block allocation technique [20] is of the most relevance to our work. In this technique, the separation of blocks of a strand (stored stream) in a multimedia server is constrained to guarantee bounds on access times of successive blocks of a strand and thus meet the continuous playback requirement. This technique, however, does not utilize the skewness in access patterns. In another study [6], the allocation of blocks to the different disks in a disk array was investigated without considering the way they should be placed on each of them.

3 Adaptive Block Rearrangement for *VOD* Servers

Movie rental patterns indicate that accesses to movies are highly localized. It was shown in [2] that one example of movie rental history approximately matches *Zipf*'s distribution [31]. Zipf's distribution states that the probability of choosing the n^{th} most popular of M movies is C/n, where $C = 1/(1 + 1/2 + 1/3 + ... + 1/M)$. For example, with this distribution, the probability of selecting the 4^{th} movie is one quarter of that of the first movie. Many research studies assume this distribution in their evaluations of *VOD* servers [3, 17, 15, 4].

In this paper, we propose an *adaptive block rearrangement* that exploits the movie locality. With this policy, the server observes the access to movies and places the data of the movies with comparable access frequencies closer to each other. The server triggers the rearrangement whenever it expects significant performance gains, it is not too busy, and it has sufficient confidence in the measured scores of the popularity of movies. The server can estimate the potential improvement by determining the level of discrepancy between the current placement and the optimal layout placement. We suggest using operator's hints to specify the expected popularity of each movie, which may be expressed as initial scores. This can prevent unnecessary rearrangements and can allow the rearrangement process to start as soon as the server is lightly loaded.

We present two algorithms for the adaptive block rearrangement: *centered-layout* and *sequential-layout*. The *centered-layout* rearrangement places the blocks based on a variation of the organ pipe heuristic. With this layout, the blocks of each movie on each disk are divided into two groups: one group is stored above the center of the disk while the other is stored below it. The *sequential-layout* rearrangement, however, places the blocks on each disk in the order of their popularity with the blocks of the most popular movie being stored at the edge of the disk.

We also consider a third alternative, called *random layout*, where movies are randomly placed on each disk. Figure 1 illustrates the different layouts for one of the disks. The disk contains data blocks of five movies. The access frequency decreases from movie 1 to movie 5. M_i represents all the blocks of movie i on a certain disk. M_{i1} represents the upper half of the blocks of movie i on a disk, while M_{i2} represents the lower half. The blocks are placed in the same way for all other disks in the server.

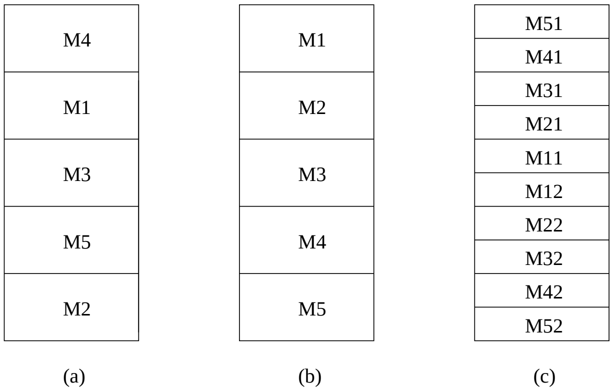

Figure 1: Block Arrangement: (a) random (b) sequential (c) centered.

We study two degrees of randomness for implementing a random layout. In the first case, called *Random+*, the blocks of the same movie are stored contiguously on each disk, whereas the place of the first block is random. *Random+* may be a result of aggressive constrained block allocation. In a system where no rearrangement is done, the constraint that the blocks of the same movie are stored contiguously may not be met. Thus, in the second case, called *Random++*, we study a higher degree of randomness to better characterize such a system. In this arrangement, the blocks of every movie are divided into G groups. The blocks in each group are stored contiguously, but the place of each group on disk is random. We choose $G = 4$ as much larger or smaller values may not be realistic.

We believe that these rearrangements are worthwhile and possible for the following reasons.

- The rearrangement can result in a significant performance improvement, as we will show in Section 5.

- The rearrangement can be done when the server is idle or lightly loaded in order to avoid or minimize the overhead. Fortunately, these periods of time are not uncommon in many *VOD* servers.
- Considering the rapid improvements in storage subsystems, such rearrangement can be done in a reasonable time.
- Incomplete or suboptimal rearrangement can be applied while attaining reasonable performance benefits.
- The releases of new very popular movies are not very frequent. Thus, most new movies can be placed near the edges of the disks without a pressing need for rearrangement.

4 Performance Evaluation Methodology

In this section, we discuss the workload characteristics, the simulation environment, and basic disk drive modeling.

4.1 Workload Characteristics

We assume, as has been done in most prior studies, that the arrival of the requests to the *VOD* server follows a Poisson Process with an average arrival rate λ. Hence, the inter-arrival time is exponentially distributed with a mean $T = 1/\lambda$. We also assume that the accesses to movies are highly localized and follow Zipf's distribution. Furthermore, we assume that the movies are stored using the MPEG-II compression standard with a data rate of 3 Megabits per second. Finally, we assume that the movie length is 90 minutes unless otherwise indicated.

4.2 Simulator

To conduct the simulation, we generated synthetic workloads based on the centered, the sequential, and random layouts. We used the DiskSim simulator [9] to simulate the disk unit in the disk array. DiskSim was used in several research studies [7, 8, 29, 30]. We used the *Quantum Atlas10K*, a relatively recent high-end disk, as the disk unit. This drive was designed to meet the high-performance requirements of data-intensive server, workstation, and storage subsystem applications [19]. These applications include online transaction processing (OLTP), 3-D image rendering and broadcast video subsystems [19]. Table 1 summarizes some important parameters of that disk. We fed DiskSim with the parameters obtained from [10]. These parameters were extracted automatically with the *DIXtrac* disk characterization tool [24].

Table 1: Main Disk Parameters

Model Number	$TM09100W$
Year	1999
Storage Capacity	9.1 GB
Rotation Speed	$10,025$ rpm
Blocks Per Disk	$179,389,986$
Number of Cylinders	1004
Number of Surfaces	6
Full Strobe Seek Time	10.82800 msec
Number of Buffer Segments	10
Number of Write Segments	1
Segment Size	374 blocks

We assume that movies are striped across all disks on 1/2-second [3] or 1/5-second intervals. We use interval length and stripe unit interchangeably in this paper. Previous studies [14] show that striping on a very large fine grain is not advantageous because the seek overhead will dominate the disk access time. Conversely, striping larger blocks increases the buffer space requirements. An interval size should be based on a reasonable tradeoff. A study for determining interval size for multimedia file servers was presented in [25].

The results in [1] show that little interaction exists between disk head scheduling and block rearrangement. Thus, we did not experiment with different head scheduling algorithms in this study. We only used the shortest-seek-time-first algorithm.

4.3 Disk Modeling

Now, we provide a brief introduction to disk drive modeling. The disk access time is composed of a head-positioning time and a data transfer time. The head-positioning time consists of a seek time and a rotational time. Both the seek time and the rotational time depend primarily on the disk technology. The seek time, however, depends on the request pattern, the employed disk scheduling algorithm, and the block arrangement as well as the disk technology. The disk seek time consists of

- a speedup, where the arm is accelerated until it reaches half of the seek distance or a fixed maximum velocity,
- a coast for long seeks, where the arm moves at its maximum velocity,
- a slowdown, where the arm is brought to rest close to the desired track, and
- a settle, where the disk controller adjusts the head to access the desired location [23].

Short seeks, therefore, are dominated by the speedup time, whereas long seeks are dominated by the coast time. Very short seeks are dominated by the settle time. Additional details can be found in [23], which provides an excellent treatment of disk drive modeling.

Because the objective of this paper is to improve the performance of *VOD* servers through minimizing the seek overhead, accurate evaluation of the seek time is required. Thus, instead of relying on analytical models, we used actual measurements to determine the mean seek time for a given seek distance.

5 Simulation Results

Next, we present and analyze the main performance results. We consider the following performance metrics: the seek distance, the seek time, and the disk access time. The reported improvements of the centered and sequential layouts, in this paper, are with respect to the random layout. For the random layout, we consider the average of 120 randomly generated layouts. For simplicity, we refer to the average case of these randomly generated layouts as the random layout. We did not conduct an exhaustive test of all possible layouts because that requires a prohibitive simulation time. For each set of parameters (number of disks, stripe unit, etc), we generated synthetic workloads for the corresponding 120 random layouts, centered layout, and sequential layout. We selected the number of movies for different disk configurations so that all the disks are utilized to almost maximum capacities.

5.1 Effect of the Layouts on Performance

In this subsection, we analyze the effect of the three layouts on performance for 0.5-second and 0.2-second stripe units. The evaluation here is based on a disk array of four disks.

5.1.1 Results for a 0.5-second Stripe Unit

Next, we present the results for a 0.5-second stripe unit ($SU = 0.5$). Figures 2, 3 and 4 show the effects of the layouts on the mean seek distance, the mean seek time, and the mean disk access time, respectively. The three performance metrics are shown as functions of the mean request inter-arrival time. We selected the range of the mean inter-arrival times so that the mean inter-arrival time that is equal to the mean disk access time falls within that range. As expected, the centered layout achieves the shortest seek distance. Namely, it reduces the mean seek distance by about 48% to about 53%. This translates to an improvement of about 23% to about 27% in the mean seek time and an improvement of about 12% in the mean disk access time. On the other hand, the improvements achieved by the sequential layout in the mean seek distance, the mean seek time, and the mean disk access time approximately range from 29% to 37%, from 15% to 19%, and from 10% to 11%, respectively. The results indicate that the seek distance in the case of the centered layout is about 25% to 29% shorter than that of the sequential layout. This is because the head is near the center of the disk most of the time in the centered-layout case, whereas it is near the edge of the disk in the sequential-layout case. Therefore, with the centered-layout rearrangement, the head travels a shorter distance to reach any cylinder in the disk.

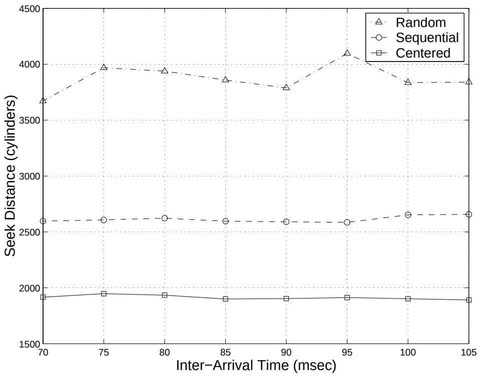

Figure 2: Effect of Layouts on Seek Distance ($SU = 0.5$, 4 $disks$)

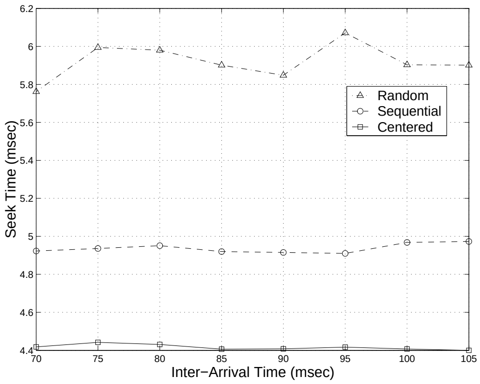

Figure 3: Effect of Layouts on Seek Time ($SU = 0.5$, 4 $disks$)

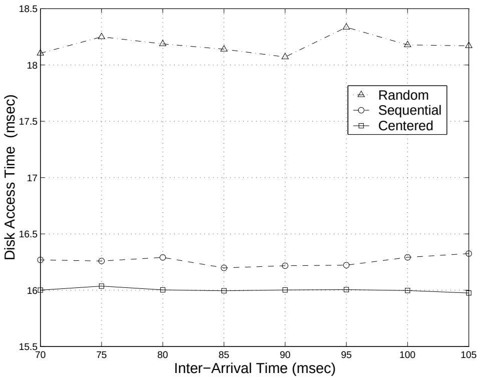

Figure 4: Effect of Layouts on Access Time ($SU = 0.5$, 4 $disks$)

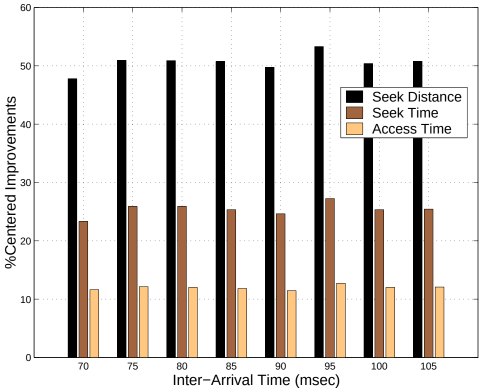

Figure 5: Comparing the Improvements Achieved by the Centered Layout ($SU = 0.5$, 4 $disks$)

Figure 5 compares the improvements achieved by the centered-layout rearrangement in the three performance metrics. The figure demonstrates that the improvement in the seek distance substantially surpasses that in the seek time. This is due to the aggressive advances in the disk technology. Similarly, the improvement in the seek time translates to a much lower improvement in the disk access time because the placement primarily impacts the seek component of the disk access time.

5.1.2 Results for a 0.2-second Stripe Unit

Now, we present the results for a 0.2-second stripe unit ($SU = 0.2$). Figures 6, 7, and 8 demonstrate the impacts of the layouts on the mean seek distance, the mean seek time, and the mean disk access time, respectively. Figure 9 compares the improvements achieved by the centered layout in the three performance metrics. We observe a similar pattern to that of a 0.5-second stripe unit. In addition, the effects of the layouts on the mean seek distance and the mean seek time are almost quantitatively identical for both stripe units. The reason is that the stripe unit mainly impacts the transfer time component of the disk service time, and this is also the reason that explains the difference in the case of the mean disk access time. We will discuss the effect of the stripe unit in more detail in subsection 5.3.

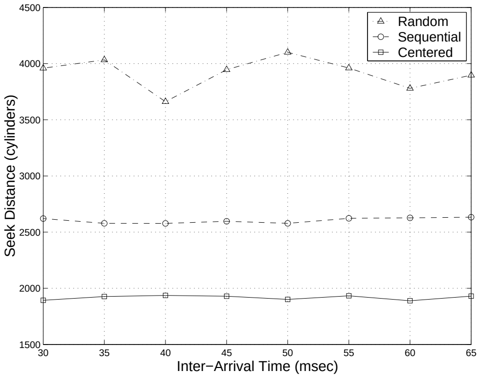

Figure 6: Effect of Layouts on Seek Distance ($SU = 0.2$, $4\ disks$)

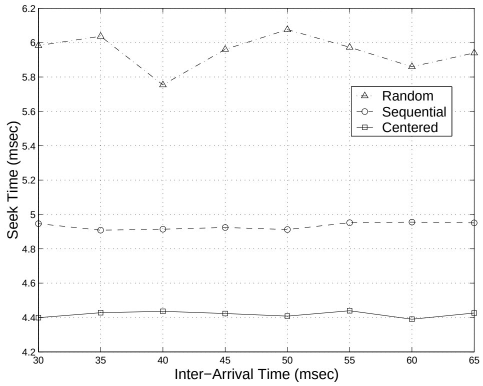

Figure 7: Effect of Layouts on Seek Time ($SU = 0.2$, $4\ disks$)

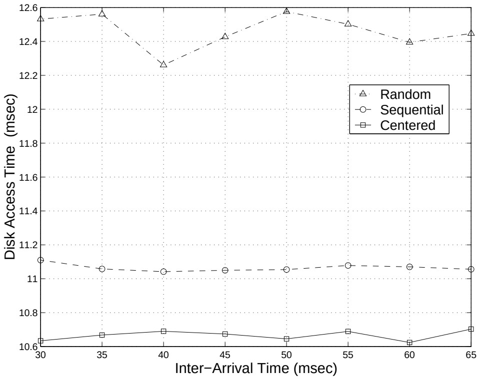

Figure 8: Effect of Layouts on Access Time ($SU = 0.2$, $4\ disks$)

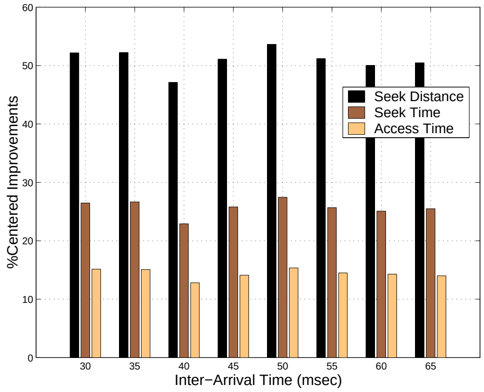

Figure 9: Comparing Seek Distance, Seek Time and Access Time Improvements ($SU = 0.2$, $4\ disks$)

Table 2 summarizes the improvements achieved by the centered and sequential layouts in the three performance metrics for both stripe units.

	Sequential Layout		*Centered Layout*	
Improvement in \ Stripe Unit	0.2 sec	0.5 sec	0.2 sec	0.5 sec
Seek Distance	30%	33%	47%	51%
Seek Time	15%	17%	23%	26%
Disk Access Time	10%	10%	13%	12%

Table 2: Summary of Improvements of Centered and Sequential Layouts ($4disks$)

5.2 Effect of the Number of Disks

Now, let us discuss the impacts of the three layouts for different sizes of the disk array. We varied the size of the disk array from 4 to 20 disks, and we varied the number of movies accordingly to keep all the disks as highly utilized as possible.

We first present the results for a stripe unit of 0.5 second. Figures 10, 11, and 12 plot the mean seek distance, the mean seek time, and the mean disk access time for the three layouts versus the number of disks, respectively. For the random and sequential layouts, the impact of the disk array size is almost negligible except for some occasional dips in the random-layout case. By contrast, with the centered layout, the three performance metrics invariably decrease with the number of disks. This can be explained as follows. As the number of movies increases with the number of disks, the share of each disk of each movie's blocks diminishes. Because of this and the high locality of reference, the centered-layout rearrangement causes most of the disk accesses to fall within a smaller area around the center of the disk.

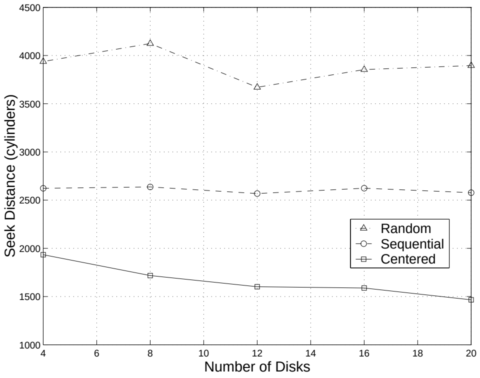

Figure 10: Effect of Number of Disks on Seek Distance ($SU = 0.5$)

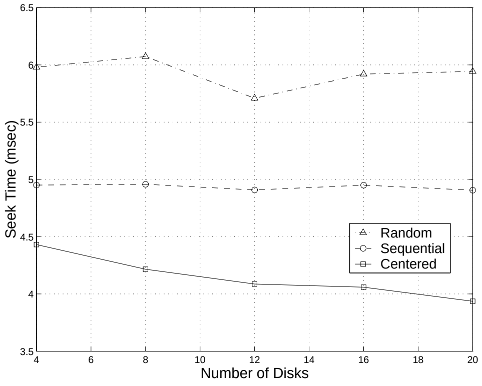

Figure 11: Effect of Number of Disks on Seek Time ($SU = 0.5$)

The performance gap between the best and the worst layouts widens as the number of possible layouts increases factorially with the number of movies. Therefore, the improvements achieved by the

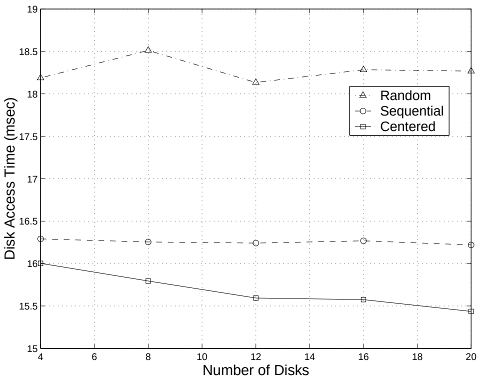

Figure 12: Effect of Number of Disks on Disk Access Time ($SU = 0.5$)

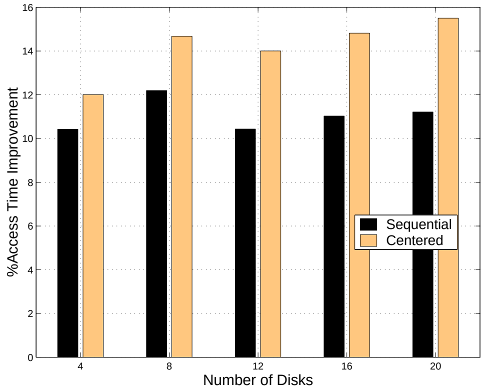

Figure 13: Effect of Number of Disks on Improvements ($SU = 0.5$)

centered and sequential layouts should be higher as the number of disks increases. Figure 13 plots the mean disk access time improvements achieved by these layouts versus the size of the disk array. These results indicate that the improvement achieved by the centered layout increases from about 12% when the number of disks is 4 to about 16% when the number of disks is 20. In contrast, not much of a difference exists in the case of the sequential layout. We believe that the actual improvements achieved by the centered and sequential layouts will grow more significantly with the size of the disk array than those shown in Figure 13. In the simulation, only 120 random layouts are generated (for each parameter set). This fixed and small number provides an increasingly and substantially reduced coverage of the set of all possible layouts as the number of disks increases. Although this number may not provide a fair study, it was necessary to avoid the factorial growth in the simulation time.

We now present the results for a stripe unit of 0.2 second. Figures 14 and 15 show the mean seek distance and the mean seek time for the three layouts versus the number of disks, respectively. Unsurprisingly, these results are very similar to those in Figures 10 and 11 of a stripe unit of 0.5 second. Figure 16 depicts the mean disk access time for the three layouts versus the number of disks. Despite that these results differ quantitatively from those for a stripe unit of 0.5 shown in figure 12, they exhibit a similar behavior. The reason for the quantitative difference is due to the difference in the data transfer times.

5.3 Effect of the Stripe Unit and the Number of Disks

In this subsection, we study the impact of both the stripe unit and the number of disks. Figure 17 depicts the improvement achieved by the centered layout in the mean disk access time versus the size of the disk array. As expected, the improvement is higher for the smaller stripe unit. This is because a smaller stripe unit demands a shorter transfer time. Therefore, the improvement achieved by the

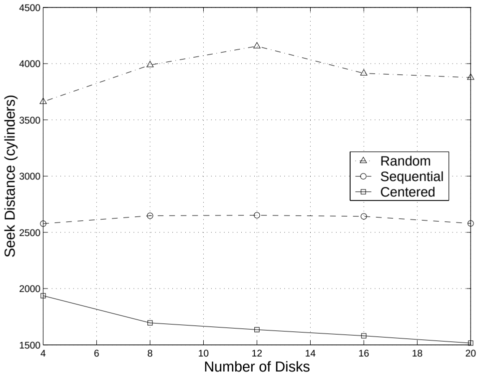

Figure 14: Effect of Number of Disks on Seek Distance ($SU = 0.2$)

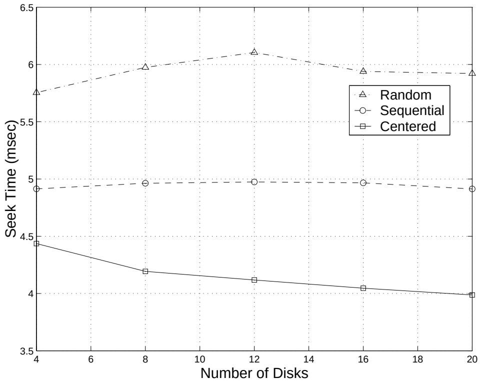

Figure 15: Effect of Number of Disks on Seek Time ($SU = 0.2$)

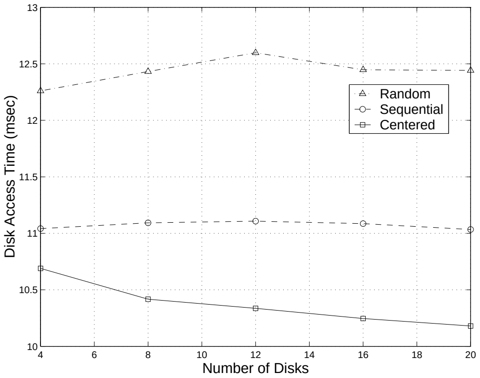

Figure 16: Effect of Number of Disks on Disk Access Time ($SU = 0.2$)

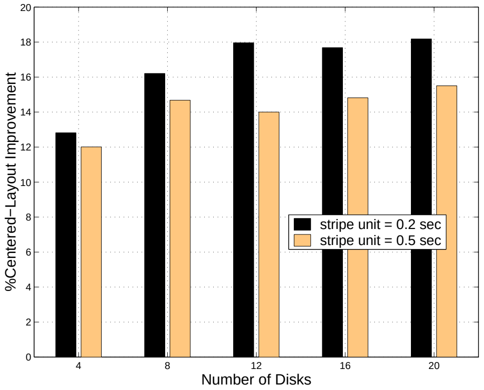

Figure 17: Effect of Stripe Unit on Centered-Layout Improvement

centered layout in the seek time produces a higher improvement in the disk access time.

5.4 Effect of Movie Length

In order to discuss how movie length affects the performance results, let us consider two different movie sets. The first set contains 90-minute movies, while the second set contains a balanced mix of 60-minute and 90-minute movies. Consequently, the number of movies in the second set is about 25% larger than the first.

Figure 18 plots the improvement achieved by the centered layout in the mean disk access time for both movie sets versus the array size. These results generally indicate that the centered layout improves performance slightly better in the mixed-movie case than in the uniform case. In comparison, Figure 19 shows that the sequential layout improves performance more significantly in the mixed-movie case than in the uniform case. Both figures are for a system with 0.2-second stripe units. We expect the improvements (achieved by the centered and sequential layouts) to grow more dramatically with decreasing movie length if the number of randomly generated layouts is adjusted to provide the same coverage of all possible layouts.

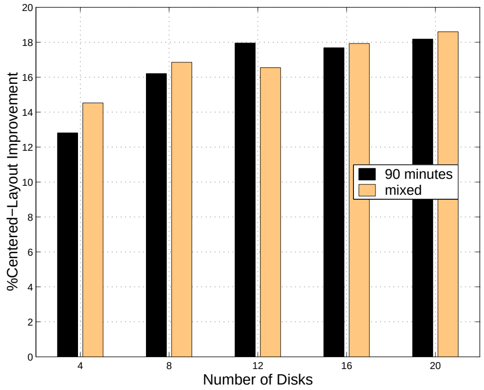

Figure 18: Effect of Movie Length on Centered-layout Improvement ($SU = 0.2$)

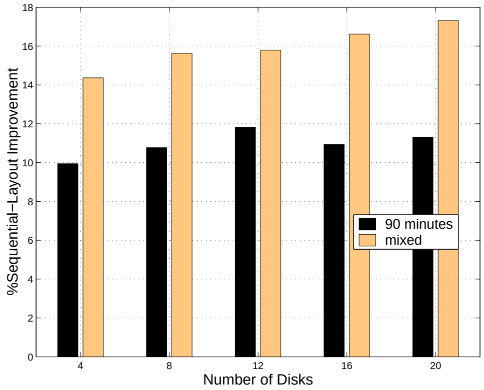

Figure 19: Effect of Movie Lengths on Sequential-layout Improvement ($SU = 0.2$)

5.5 Effect of the Degree of Randomness

Finally, let us discuss the improvement achieved by the centered layout with respect to the degree of randomness available to the random layout. Figure 20 shows the improvement achieved by the centered layout in the average disk access time versus the size of the disk array for the two cases: *Random+* and *Random++*. The results are for a stripe unit of 0.5 second. We observe that the improvement is lower with the higher degree of randomness. This is because, in *Random++*, the blocks of each movies on each disk are divided into four groups, which are randomly placed. These groups share the same access frequency, so the probability of poor placement decreases.

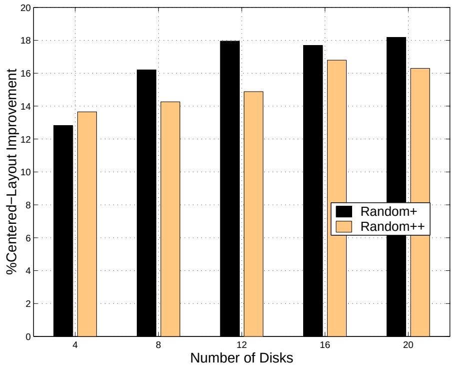

Figure 20: Effect of Degree of Randomness on Centered-layout Improvement ($SU = 0.5$)

6 Conclusions and Future Work

In this paper, we have proposed an adaptive block rearrangement policy based on movie access patterns and have shown that it is a worthwhile technique to enhance the performance of *VOD* servers. We have presented two algorithms for this rearrangement: *sequential layout* and *centered layout.*

We have conducted a simulation study to demonstrate the advantages of such rearrangement on disk performance. The results indicate that significant performance benefits can be attained, and that the centered layout is the best performer. Compared with random layouts, the mean seek distance of the centered layout decreases by approximately 50% when the number of disks is 4, and to about 60% when the number of disks is 20. The corresponding reductions in the mean disk service time are between 13% to 15% as the number of disks increases from 4 to 20. The performance also improves with smaller stripe units and shorter movie lengths. We have studied the effects of two degrees of randomness. The results demonstrate that the improvements are lower when the degree of randomness is higher, but they remain significant.

In this study, we assume that movies are stored using the MPEG-II compression standard with a data rate of 3 Megabits per second. We expect a higher data rate to have a similar impact as increasing the movie length - a slight reduction in improvements.

In future work, we will conduct a higher level study to evaluate how the improvements in disk performance translate to the overall performance. Namely, we will study how the adaptive block rearrangement contributes to the number of concurrent streams that can be served by a *VOD* server, while maintaining a “reasonable” QoS. Moreover, we will study the impacts of caching, admission control, interactive operations (pause, fast forward, etc), and variable QoS on the proposed schemes.

References

[1] S. Akyurek and K. Salem. Adaptive Block Rearrangement. *ACM Transactions on Computer Systems*, 13(2): 89-121, May 1995.

[2] A. L. Chervenak. *Tertiary Storage: An Evaluation of New Applications.* PhD thesis, U.C. Berkeley, December 1994. University of California at Berkeley Technical Report UDB/CSD 94/847, December 1994.

[3] A. L. Chervenak, D. A. Paterson, and R. H. Katz. Choosing the Best Storage Systems for Video Service. *In Proceedings of ACM Conference on Multimedia*, pages 109-119, 1995.

[4] A. Dan, and D. Sitaram. Buffer Management Policy for an On-Demand Video Server. Technical Report RC 19347, IBM Watson Research Center, January 1994.

[5] A. Dan, D. M. Dias, R. Mukherjee, D. Sitaram, R. Tewari. Buffering and Caching in Large-Scale Video servers. *In Digest of Papers.* IEEE International Computer Conference, 1995.

[6] R. Flynn, and W. Tetzlaff. Disk Striping and Block Replication Algorithms for Video File Servers. *In Proceedings of International Conference on Multimedia Computing and Systems*, pages 590-597, 1996.

[7] G. Ganger, and Y. Patt. The Process-Flow Model: Examining I/O Performance from System's Point of View. *In Proceedings of International Conference on Measurement and Modeling of Computer Systems (ACM SIGMETRICS)*, pages 86-97, May 1993.

[8] G. Ganger. Generating Representative Synthetic Workloads An Unsolved Problem. *In Proceedings of Computer Measurement Group (CMG) Conference*, pages 1263-1269, December 1995.

[9] G. Ganger, B. Worthington, and Y. Patt. *The DiskSim Simulation Environment.* Version 2, Reference Manual, CMU, December 1999.

[10] G. Ganger and J. Schindler. *Database of Validated Disk Parameters for DiskSim.* Available at: `http://www.ece.cmu.edu/~ganger/disksim/diskspecs.html`.

[11] S. Ghandeharizadeh and D. Kim. On-line Re-organization of Data in Scalable Continuous Media Servers. Technical report USC-CS-TR96-634, Department of Computer Science, University of Southern California, 1996.

[12] D. D. Grossman, and H. F. Silverman. Placement of Records on a Secondary Storage Device to Minimize Access Time. *In Journal of the ACM* 20(3): 429-438, 1973.

[13] G. H. Hardy, J. E. Littlewood and G. Polya. *Inequalities.* Cambridge University Press, Cambridge, England, 1952.

[14] K. Keeton, A. L. Drapeau, D. A. Patterson, and R. H. Katz. Storage Alternatives for Video Service. *In Digest of Papers.* Thirteen IEEE Symposium on Mass Storage Systems, pages 100-105, June 1994.

[15] S. Kim, A. Sivasubramaniam, C. R. Das. Analyzing Cache Performance for Video Servers. *In Proceedings of International Conference on Parallel Processing Workshops on Architectural and OS Support for Multimedia Applications*, pages, 38-47, 1998.

[16] S. Kim and C. R. Das. A Reliable Statistical Admission Control Strategy for Interactive Video-On-Demand Servers with Interval Caching. *In Proceeding of International Conference on Parallel Processing*, pages 135-142, 2000.

[17] S. Kim, C. R. Das, and A. Sivasubramaniam. Performance Analysis of A Buffer Management Technique for Interactive Video-on-Demand. *In Proceedings of International Conference on Multimedia Modeling*, 2000.

[18] J. Nieh and M. S. Lam. The Design, Implementation and Evaluation of SMART: A Scheduler for Multimedia Applications. *In Proceedings of the Sixteen ACM Symposium on Operating Systems Principles*, pages 184-197, October 1997.

[19] Quantum Corporation. A New Standard of Performance. Article, 2000. Available at: `http://www.quantum.com/products/hdd/article_atlas10k.htm`.

[20] P. V. Rangan, and H. M. Vin. Designing File Systems for Digital Video and Audio. *In Proceedings of the 13th ACM Symposium on Operating Systems Principles*, pages 81-94, October 1991.

[21] A. L. N. Reddy, J. Wyllie. Disk Scheduling in a multimedia I/O system. *In Proceedings of the First ACM Conference on Multimedia*, pages 225-233, August 1993.

[22] C. Reummler, and J. Wilkes. Disk Shuffling. HPL-91-156, Hewlett-Packard Laboratories, Palo Alto., California, October 1991.

[23] C. Ruemmler, and J. Wilkes. An introduction to disk drive modeling. *IEEE Transactions on Computer*, 27(3): 17-29, March 1994.

[24] J. Schindler and G. Ganger. Automated Disk Drive Characterization. *In Proceedings of the international conference on International Conference on Measurements and Modeling of Computer Systems (ACM SIGMETRICS)*, pages 112-113, 2000.

[25] P. Shenoy, and V. Harric. Efficient Striping Techniques for Variable Bit Rate Continuous Media File Servers. *Performance Evaluation Journal, 38(2)*: 175-199, 1999.

[26] C. Staelin, and H. Garcia-Molina. Smart Filesystems. *In Proceedings of the Winter 1991 USENIX Conference,* pages 45-51, 1991.

[27] P. Vongsathorn, and S. D. Carson. A system for adaptive disk rearrangement. *Software – Practice and Experience*, 20(3): 225-242, March 1990.

[28] C. K. Wong. Minimizing Expected Head Movement in One-Dimensional and Two-Dimensional Mass Storage Systems. *ACM Computing Surveys*, pages 167-178, June 1980.

[29] B. Worhington, G. Ganger, and Y. Patt. Scheduling Algorithms for Modern Disk Drives. *In Proceedings of International Conference on Measurement and Modeling of Computer Systems (ACM SIGMETRICS)*, pages 146-156, May 1994.

[30] Worthington, B. *Aggressive Centerized and Distributed Scheduling of Disk Requests.* Ph.D Dissertation, CSE-TR-244-95, University of Michigan, An Arbor, June 1995.

[31] G. K. Zipf. *Human Behavior and Principles of Least Effort: An Introduction to Human Ecology.* Addison Wesley, Cambridge, Massachusetts, 1949.