# Adaptive Block Rearrangement Algorithms 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
{sarhan,das}@cse.psu.edu

## Abstract

*Video-on-demand (*VOD*) is increasingly becoming one of the most important and successful services due to the recent advances in storage subsystems, compression technology, and networking. Therefore, the investigation of various alternatives to improve the performance of* VOD *servers has become a major research focus. The reduction of 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 accesses. In this paper, we exploit the access patterns 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 at the edge of the disk.*

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

## 1. Introduction

There is a strong interest in the online delivery of movies and other multimedia information for entertainment, business, and educational purposes. Taking leverage of the recent advances in storage subsystems, the high compression rates, and the dramatic increase in the bandwidth of networks, multimedia-on-demand in general and video-on-demand (*VOD*) in particular is increasingly becoming one of the most important and successful services.

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). This number is is highly constrained due to the real-time playback and 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 [20, 3], replication [6, 3], scheduling [18, 15], block allocation [17, 6], and caching [4, 5, 13, 14]. 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 of *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.

Rental patterns indicate that accesses to movies are highly localized, with only a small number of movies receiving most of the accesses [2]. We propose an *adaptive block rearrangement* policy that reduces disk seek time by exploiting this locality. A *VOD* server applying the rearrangement monitors the access to movies and rearranges the blocks on each disk accordingly, whenever there is a need and a good opportunity to do that. The policy ensures 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 [11]. The organ pipe heuristic places the most frequently accessed data in the

---

[*] This research was supported in part by NSF grants MIPS-9634197, CCR-9900701, and equipment grants from NSF and IBM.

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 use random layouts to characterize placements in systems where no rearrangement is maintained. In a random layout, movies are placed randomly. We study two degrees of randomness. The simulation study is conducted on a disk array of *Quantum Atlas10K*. The *DiskSim* simulator [7] is used to simulate the disk unit of the disk array. In order to ensure high accuracy in the disk simulation, the seek time for a given seek distance is based on actual measurements.

The performance parameters analyzed in this paper are seek distance, seek time, and disk access time. The effects of the size of the disk array, the stripe unit, and the movie length on these parameters are also studied. The simulation results show that an adaptive block rearrangement based on either of the presented algorithms could yield significant gains in disk performance. The benefits of the centered layout relative to random layouts can be summarized as follows. The improvements 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. Particularly, 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 has been proposed to improve disk performance for general-purpose systems [22, 19, 21, 1]. It has been shown that the organ pipe heuristic places data optimally if data references are derived from an independent random process with a known fixed distribution [10, 23]. Variations of this heuristic have been shown to be effective in practical systems. The organ pipe heuristic has been used in the cylinder shuffling technique [22, 19] and been employed by iPcress (an experimental filesystem) [21]. More recently, an adaptive block rearrangement technique [1] based on this heuristic has been proposed.

To the knowledge of the authors, the effectiveness of block rearrangement has not been investigated for *VOD* servers, in which the block access size is much larger than that of traditional filesystems. The constrained block allocation technique [17] 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. However, the skewness in access patterns has not been utilized. In another study [6], the allocation of blocks to the different disks in a disk array has been investigated without considering the way they should be placed on each of them.

## 3. Adaptive Block Rearrangement for *VOD* Servers

It has been shown in [2] that one example of movie rental history approximately matches the *Zipf*'s law distribution [24]. Zipf's law 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, using this distribution, the probability of requesting the *4th* movie is one quarter of that of the most popular. Many research studies have assumed this distribution in their evaluations of *VOD* servers [3, 14, 13, 4].

In this paper, we propose an *adaptive block rearrangement* that exploits movie locality. A *VOD* server applying this rearrangement monitors the access to movies and places the blocks of movies with comparable access frequencies closer to each other. The rearrangement is triggered whenever significant performance gains are expected, the server is not too busy, and there is reasonable confidence in the measured scores of the popularity of movies. The improvement expectations could be estimated by the discrepancy between the current placement and the optimal layout placement. We suggest using operator's hints to specify the expected popularity of each movies which might be expressed as initial scores. This could avoid unnecessary rearrangements and start with the rearrangement process 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 [11]. 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, starting at the edge of the disk, with the most popular movie being stored first.

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. $M_{i1}$ represents the upper half of the blocks of movie $i$ on a disk, while $M_{i2}$ represents the lower half.

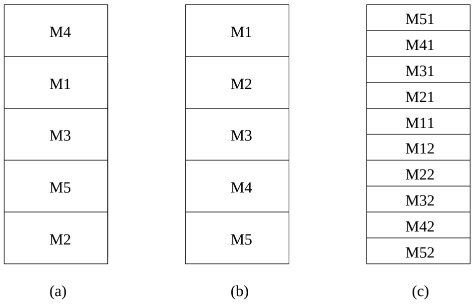

**Figure 1. Block Arrangement: (a) random (b) sequential (c) centered.**

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+* could be viewed as a result of aggressive constrained block allocation. It is important to note that, in a system in which 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 due to the following reasons. First, the rearrangement could result in a significant performance improvement, as will be shown later. Second, the rearrangement could be done when the server is idle or lightly loaded in order to avoid or minimize the overhead as these periods of time are not uncommon in many *VOD* servers. Third, considering the rapid improvements in storage subsystems and compression technology, such rearrangement could be done in a reasonable time. Fourth, incomplete or suboptimal rearrangement could be applied while attaining reasonable performance benefits. Finally, the releases of new very popular movies are not very frequent. Thus, most new movies could 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 and the simulation environment.

### 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 $\lambda$. Hence, the interarrival time is exponentially distributed with a mean $T = 1/\lambda$. We also assume that the accesses to movies are highly localized and follow the 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, synthetic workloads are generated based on the centered, the sequential, and random layouts. The DiskSim simulator [7] is used to simulate the disk unit in the disk array. The *Quantum Atlas10K*, a relatively recent high-end disk, is used as the disk unit. This drive was designed to meet the high-performance requirements of data-intensive server, workstation, and storage subsystem applications [16]. Table 1 summarizes some important parameters of that disk. The disk simulator is fed with the parameters obtained from [8].

**Table 1. Main Disk Parameters**

| | |
|---|---:|
| *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 [12] show that striping on a very large fine grain is not advantageous since the seek overhead will be more dominant. Conversely, striping larger blocks increases the buffer space requirements. An interval size should be chosen based on a reasonable tradeoff.

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

## 5. Simulation Results

Next, we present and analyze the main performance results. The number of movies for different disk configurations is chosen so that all the disks are utilized to almost maximum capacities. In the random layout case, we consider the average of 120 randomly generated layouts and compare it with the corresponding sequential and centered layouts. Conducting an exhaustive test for all possible layouts requires a prohibitive simulation time. For simplicity, we will refer to the average case of 120 randomly generated layouts as the random layout. For each set of parameters (number of disks, stripe unit, etc), synthetic workloads are generated for the corresponding 120 random layouts, centered layout, and sequential layout. The reported improvements of the centered and sequential layouts, in this paper, are relative to the random layout.

### 5.1. Effect of the Layouts on Performance

In this subsection, we analyze the effect of the three layouts on performance. The evaluation here is based on a disk array of four disks.

Figures 2, 3 and 4 show the effect of the layouts on the average seek distance, seek time, and disk access time for a stripe unit of 0.5 second ($SU = 0.5$), respectively. In these figures, the mean seek distance, seek time or disk access time is shown as a function of the mean request inter-arrival time. We observe that the centered layout achieves the shortest seek distance which minimizes the seek time and the disk access time. We also observe that the sequential layout offers better performance than that of the random layout. The improvement of the centered layout in the mean seek distance over the observed average request inter-arrival times ranges from about $48\%$ to $53\%$. This translates to an improvement of about $23\%$ to $27\%$ in the mean seek time and an improvement of about $12\%$ in the mean disk access time. We selected the range of the mean inter-arrival times so that the mean inter-arrival time which is equal to the mean service time (disk access time) falls within that range.

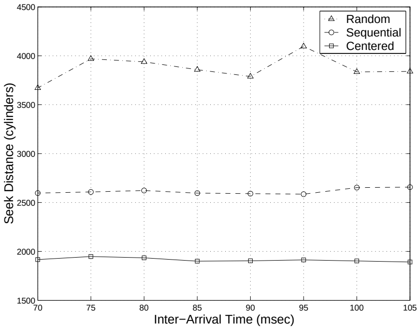

**Figure 2. Effect of Layouts on Seek Distance** ($SU = 0.5$, $4\ disks$)

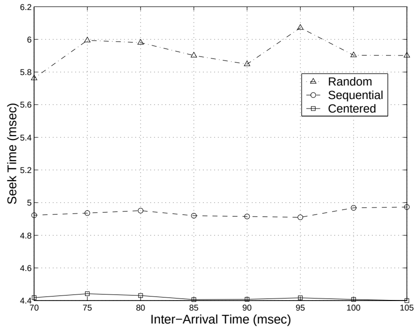

**Figure 3. Effect of Layouts on Seek Time** ($SU = 0.5$, $4\ disks$)

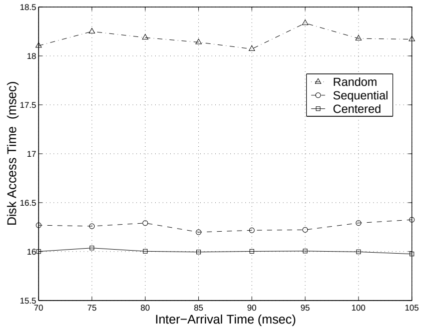

**Figure 4. Effect of Layouts on Access Time** ($SU = 0.5$, $4\ disks$)

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, which can be explained as follows. In the centered-layout case, the head is near the center of disk most of the time, whereas it is almost near the edge of the disk in the sequential layout. Therefore, with the centered-layout arrangement, the head travels a shorter distance to get to any cylinder in the disk.

Figure 5 depicts a comparison between the improvements of the centered-layout rearrangement in the average seek distance, seek time, and disk access time. These results show that the improvement in seek distance is much higher than that in seek time as a result of the aggressive advances in disk technology. Similarly, the improvement in seek time is much higher than that in the disk access time since the placement primarily impacts only the seek component of the disk access.

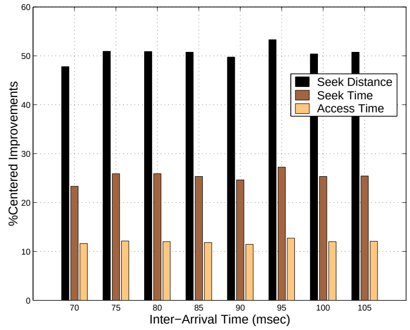

**Figure 5. Comparing the Improvements of the Centered Layout** $(SU = 0.5, 4\ disks)$

## 5.2. Effect of the Number of Disks

Now, we study the impact of the three layouts for different sizes of the disk array. The size of the disk array is varied from 4 to 20 disks, and the number of movies is varied accordingly to keep all the disks as highly utilized as possible.

First, we present the results for a stripe unit of $0.5$ second. Figures 6, 7, and 8 plot the mean seek distance, seek time, and disk access time for the three layouts versus the number of disks, respectively. For the random and sequential layouts, the effect of the disk array size is almost negligible except for some occasional dips in the random layout case. However, in the case of 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 reduces. 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.

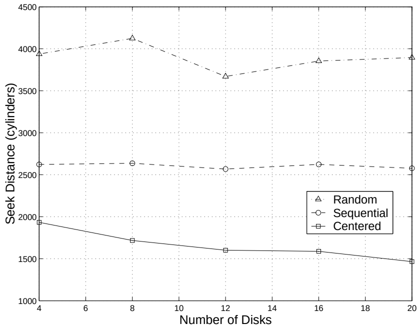

**Figure 6. Effect of Number of Disks on Seek Distance** $(SU = 0.5)$

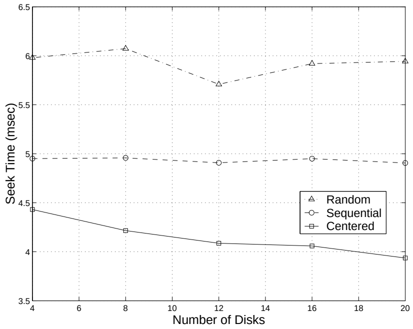

**Figure 7. Effect of Number of Disks on Seek Time** $(SU = 0.5)$

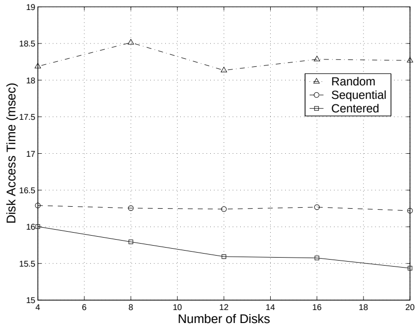

**Figure 8. Effect of Number of Disks on Disk Access Time** $(SU = 0.5)$

The performance gap between the best and the worst layouts is widened as the number of possible layouts increases factorially with the number of movies. Therefore, the improvements of the centered and sequential layouts should be higher as the number of disks increases. Figure 9 shows the average disk access time improvements of the centered and sequential layouts as a function of array size. These results show that the improvement of the centered layout in the mean disk access time generally increases with the number of disks. It increases from about $12\%$ when the number of disks is 4 to about $16\%$ when the number of disks is $20$. In contrast, there is not much of a difference in the case of the sequential layout.

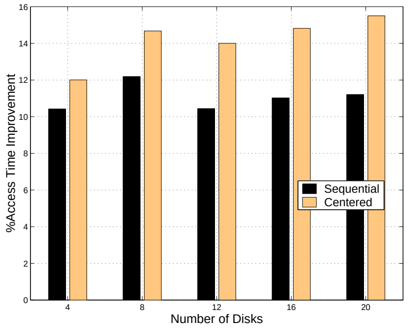

**Figure 9. Effect of Number of Disks on Improvements** $(SU = 0.5)$

We believe that the actual improvements of the centered and sequential layouts will grow more significantly with the size of the disk array than those in figure 9. 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 simulation time.

Now, we present the results for a stripe unit of $0.2$ second. Figures 10 and 11 show the mean seek distance and seek time for the three layouts versus the number of disks, respectively. These results are very similar to those in figures 6 and 7 of a stripe unit of $0.5$ second. Figure 12 depicts the mean disk access time for the three layouts versus the number of disks. The results here are quantitatively different from those for a stripe unit of $0.5$ (refer to figure 8) but exhibit a similar behavior.

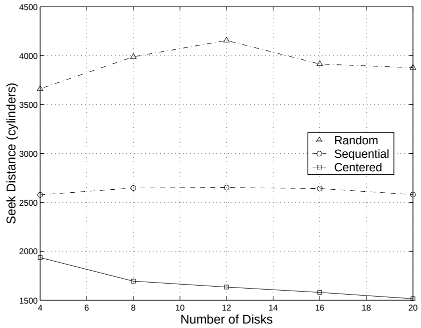

**Figure 10. Effect of Number of Disks on Seek Distance** $(SU = 0.2)$

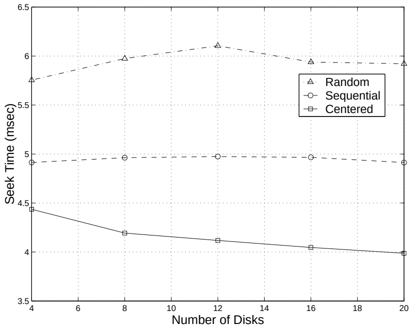

**Figure 11. Effect of Number of Disks on Seek Time** $(SU = 0.2)$

## 5.3. Effect of the Stripe Unit and the Number of Disks

Now, we study the impact of both the stripe unit and the number of disks. Figure 13 depicts the improvement of the centered layout in the average access time as a function of array size. We observe that the improvement of the centered layout is higher for a smaller stripe unit,. This is because a smaller stripe unit requires a shorter transfer time. Therefore, the improvement of the centered layout in seek time translates to a higher improvement in disk access time.

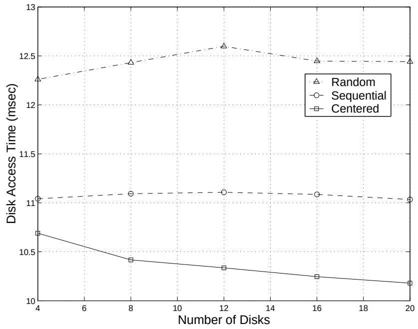

**Figure 12. Effect of Number of Disks on Disk Access Time** $(SU = 0.2)$

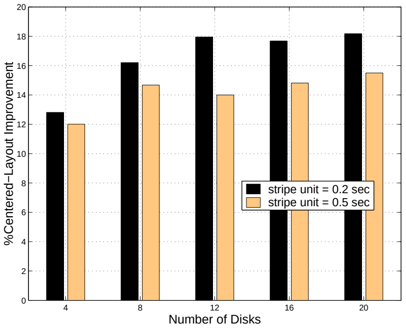

**Figure 13. Effect of Stripe Unit on Centered-Layout Improvement**

## 5.4. Effect of Movie Length

In this subsection, we analyze the effect of the layouts on performance for 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. Thus, the number of movies in the second set is about $25\%$ larger than that of the first. The evaluation here is based on a stripe unit of $0.2$ second.

Figure 14 depicts the improvement of the centered layout in the average disk access time versus array size. The figure generally indicates that the improvement of the centered layout is slightly higher in the mixed-movie case than that in the uniform case since a larger number of movies could be stored. In comparison, the improvement of the sequential layout is much higher in the mixed-movie case compared with that in the uniform case as shown in figure 15.

We expect the improvements to increase more significantly with decreasing movie length if the number of randomly generated layouts is adjusted to provide the same coverage of all possible layouts.

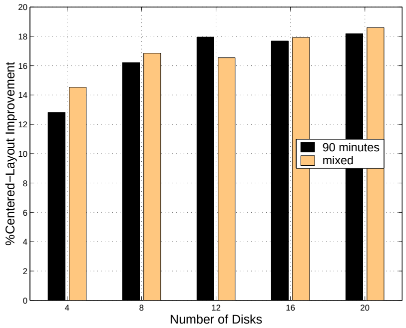

**Figure 14. Effect of Movie Length on Centered-layout Improvement** $(SU = 0.2)$

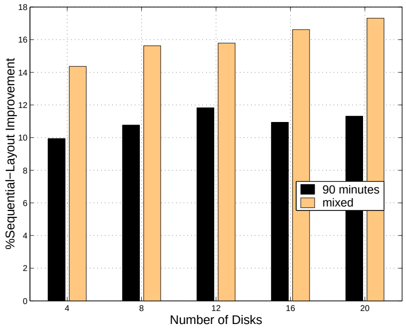

**Figure 15. Effect of Movie Lengths on Sequential-layout Improvement** $(SU = 0.2)$

### 5.5. Effect of the Degree of Randomness

Finally, we study the improvement of the centered layout in the disk performance with respect to the degree of randomness available to the random layout.

Figure 16 shows the improvement of the centered layout in the average disk access time versus array size for the two cases: *Random*+ and *Random*++. The results are based on a stripe unit of $0.5$ second. We observe that the improvement is lower when there is a higher degree of randomness. This is because, in *Random*++, the blocks of each movies on each disk are divided into four groups that are randomly placed. It is important to note that those four groups share the same access frequency as in the Zipf's distribution. Consequently, the probability of getting closer to the worst layout is much lower in the case of *Random*++, although it is a general case of *Random*+.

## 6. Conclusions and Future Work

There are strong indications that *VOD* is increasingly becoming one of the most successful services offered on the emerging technologies. Therefore, the design of *VOD* servers has received numerous research attention.

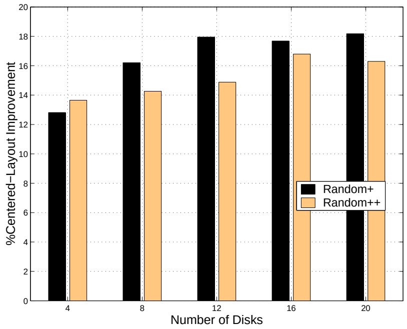

**Figure 16. Effect of Degree of Randomness on Centered-layout Improvement** $(SU = 0.5)$

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 could be attained, and that the centered layout is the best performer. Compared with random layouts, the mean seek distance of the centered layout reduces 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 show that the improvements are lower when the degree of randomness is higher, but the benefits 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 system performance. In particular, we will study how the adaptive block rearrangement contributes to the number of streams that can be served by a *VOD* server, while maintaining a "reasonable" QoS. In addition, we will study the effect 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*, pp. 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*, pp. 590-597, 1996.

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

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

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

[10] 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.

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

[12] 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, pp. 100-105, June 1994.

[13] 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*, pp. 38-47, 1998.

[14] 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.

[15] 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*, pp. 184-197, October 1997.

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

[17] 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*, pp. 81-94, October 1991.

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

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

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

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

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

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

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