HARMONIC DATA PLACEMENT: FILE SYSTEM SUPPORT FOR SCALABLE STREAMING

Youjip Won, Seungheon Yang

Dept. of Electronics and Computer Engineering
Hanyang University, Seoul, 133-791, Korea
yjwon@ece.hanyang.ac.kr, shyang@lge.com

Sooyong Kang *

Department of Computer Science Education
Hanyang University, Seoul, 133-791, Korea
sykang@hanyang.ac.kr

Abstract

Scalable encoding scheme enables the player or streaming server to adaptively change the playback rate of multimedia content. However, in scalable streaming of layer encoded content, sequential playback of content does not necessarily coincide with the sequential scan of a file. This property introduces another dimension of complexity in the scheduling of data block retrieval. In this work, we develop a novel file organization strategy, Harmonic Interleaving, which can effectively handle the dynamically changing playback rate of multimedia data retrieval. The proposed scheme not only eliminates the retrieval of unnecessary blocks but also minimizes disk head movement. Via experiment, Harmonic Interleaving exhibits superior disk utilization on a moderately loaded network.

1. INTRODUCTION

Due to the sequential access nature of real-time multimedia playback, file system puts great emphasis on placing data in *seek* optimized fashion. Hence, file systems support for real-time multimedia playback has been the subject of intense research recently. Some of these studies strive to find the relationship between real-time requirements of individual playbacks and file system level data retrieval scheduling[1, 3, 4]. Others exploit the variability in linear bit density of the hard disk track in placing the multimedia blocks[6, 7].

However, those works do not consider the disk access pattern under scalable streaming service. When the file is organized as a sequence of logical units, e.g. frames, playback of a single layer multimedia file yields a simple sequential scan. However, when content is created with a layered encoding scheme, playback no longer yields a sequential scan. This is because only a subset of individual frame information may be selected for transmission. The subset selected for transmission dynamically changes as a function of network bandwidth availability. Therefore, a layered encoding scheme introduces another dimension of complexity from the file system's point of view. A legacy file system model and a disk scheduling strategy do not incorporate the characteristic of layered encoding and leaves much to be desired.

In this work, we propose a novel file organization technique, *Harmonic Interleaving* which effectively exploits the access characteristics of layer encoded multimedia content.

*Corresponding author

Table 1: Notations

Parameters	Description
n	number of layers
m	number of segments in a media file ($m = T/T_{seg}$)
B_i	data rate of ith layer
B_{movie}	total data rate($B_1 + B_2 + \ldots + B_n$)
α_i	transmit ratio of layer i
L_{ij}	data unit for j-th segment of i-th layer ($1 \leq i \leq n, 1 \leq j \leq m$)
$\mathcal{L}_i$	set of data units belonging to i-th layer
T	playback length of movie
T_{seg}	playback time of a segment(eg. 1sec)
l	number of low layers
C_{movie}	number of cylinders that movie occupy
T_{cyl}	time to read a single cylinder
T_{total}	time of reading data from disk
T_{data}	time of reading data purely of T_{total}
$T_{overhead}$	disk overhead

Harmonic Interleaving combines advantages of the above mentioned two file organization types.

The rest of the paper is organized as follows. In section 2, we present three different file organization techniques. In section 3, we develop the performance model for each file organization scheme. In section 4, we examine the efficient of individual file organizations via simulation as well as physical experiments. In section 5, we conclude the paper.

2. LAYER ALLOCATION STRATEGY

In this section, we present three file organization strategies for scalable streaming. We view a media file as a collection of logical storage units, called *segments*. A segment can be a frame or group of pictures. Table 1 illustrates the notations used in this paper.

The first file organization strategy is *Progressive* placement. In *Progressive* placement, as in legacy file organization, the file is physically organized as a sequence of logical units (frames) and each logical unit is organized as a collection of a layers. Fig. 1 illustrates the *Progressive Placement* strategy.

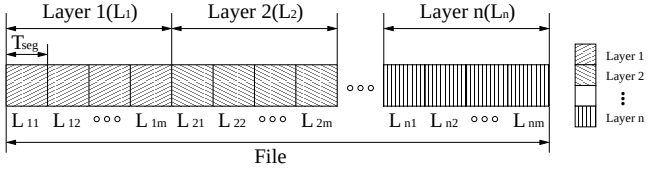


Figure 1: Progressive Placement Scheme

The second file organization strategy is *Interleaved* placement. In *Interleaved* placement, a file is physically organized as a sequence of layers and each layer consists of the data blocks of the respective layer for each logical unit. Fig. 2 illustrates the *Interleaved Placement* scheme for scalable coded data.

The *Progressive* placement and *Interleaved* placement strategies are two extremes in a wide spectrum of file organization techniques. Neither of these organizations yields satisfactory efficiency under dynamically changing network conditions. Hence, we propose a novel file organization strategy, *Harmonic Interleaving*. Fig. 3 illustrates the file organization under the *Harmonic Interleaving* placement strategy. In Harmonic Interleaving, data blocks are partitioned into two groups based upon the layers which they belong to. The layers are categorized into two groups: a set of lower layers, $\mathcal{L}_{lower}$ and a set of upper layers, $\mathcal{L}_{upper}$. For example, with five layers, the layers can be partitioned as follows: $\mathcal{L}_{lower} = \{L_1, L_2, L_3\}$ and $\mathcal{L}_{upper} = \{L_4, L_5\}$. In this case, we assume that up to layers 3 are frequently requested and layer 4 and layer 5 are hardly requested. *Harmonic Placement* adopts *Progressive Placement* for *inter-group* placement and Interleaving for *intra-group* placement. Using this scheme, we can reduce disk seek time by increasing the physical continuity of data blocks belonging to frequently serviced layers. Hence, when only lower layers of information is streamed in most of the time and information in the upper layers is rarely used, this scheme outperforms other schemes. However, if the upper layers are accessed frequently, the required disk seek overhead can result in lower performance. Therefore, the boundary of the layer groups needs to be carefully determined by considering both network bandwidth and the client device. The effectiveness of Harmonic Interleaving is subject to the layer partitioning policy and the variability in network bandwidth availability. We will explore this issue more formally in section 3.

3. MODELING OF DISK RETRIEVAL OPERATION

We use disk utilization, ρ, to quantify the effectiveness of the placement strategy. Disk utilization, ρ, is a ratio between total elapsed time to read data and the time spent on reading the actual information from the disk excluding the disk overhead. Total elapsed time, T_{total}, consists of the time to read the data, T_{data}, and overhead, $T_{overhead}$. $T_{overhead}$ consists of seek, rotational latency, head switch, command processing, etc. T_{data} depends upon the amount of data to read. $T_{overhead}$ is governed by the data placement strategy. Let $\alpha_i (0 \leq \alpha_i \leq 1)$ be the fraction of playback duration during which $L_1, \ldots, L_i$ are presented. Let D_i and B be the amount of data blocks in L_i (Byte) and maximum disk transfer rate(Bytes/sec), respectively. Then, the amount of time spent on reading data blocks, excluding disk overhead, during the entire service time can be calculated as $T_{data} = \sum_{i=1}^{n} \left(\sum_{j=i}^{n} \alpha_j \right) \frac{D_i}{B}$.

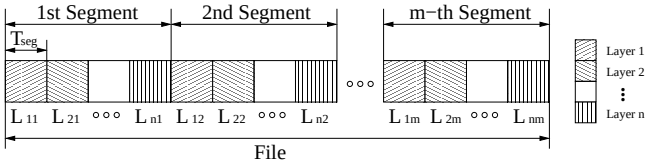


Figure 2: Interleaved Placement

3.1. Modeling Progressive Placement Scheme

Let C_i be the number of cylinders occupied by $\mathcal{L}_i$. Then, $C_i = \lceil \frac{B_i}{B_{movie}} \times C_{movie} \rceil$. Let $T_{seek}(i)$ be the seek time for i cylindrical distance. Let us assume that the disk retrieves data for the lower i layers. The overhead incurred in retrieving a single data unit is the sum of seek times between adjacent layers and the seek time returning to the lowest layer from layer i, i.e. $\sum_{j=1}^{i-1} T_{seek}(C_j) + T_{seek}(\sum_{j=1}^{i-1} C_j)$. The total overhead in playback can be formulated as $T_{overhead} = m \sum_{i=2}^{n} \alpha_i \left(\sum_{j=1}^{i-1} T_{seek}(C_j) + T_{seek}(\sum_{j=1}^{i-1} C_j) \right)$.

As we can see from the equation, *Progressive Placement* scheme results in large disk overhead when α_1 is small and $\alpha_i, (2 \leq i \leq n)$ is large. On the other hand, if α_1 is very large, e.g. due to poor network bandwidth, then this file organization scheme yields efficient disk utilization.

3.2. Modeling Interleaving placement Scheme

In *Interleaving Placement* scheme, segments are placed with respect to their temporal order. For a segment, the data blocks are placed with respect to their layers. If the application needs to retrieve all layers for a segment, it will simply yield a sequential scan. A problem may occur when the application retrieves data blocks in a proper subset of all layers. When we do not need data blocks in all layers, there can be two retrieval strategies: (i) we can either read all layers and discard unnecessary data blocks (*blind scan*), or (ii) we can read only the data blocks in the selected layers (*selected retrieval*). In this paper, we assume *blind scan* strategy. Using *blind scan*, the total time to read data blocks can be modeled as $T_{total} = C_{movie} \times T_{cyl} + (C_{movie} - 1) \times T_t$, where T_t represents the seek time between adjacent cylinders.

3.3. Modeling Harmonic Interleaving Placement Scheme

Both *Progressive* and *Interleaved* file organization have their own advantages and disadvantages. In *Progressive Placement, inter* layer data block accesses can cause excessive disk seek. On the other hand, Interleaved placement can cause retrieval of unnecessary data blocks. Let C_{lower} and C_{upper} be the number of cylinders occupied by the layers in $\mathcal{L}_{lower}$ and $\mathcal{L}_{upper}$, respectively. Fig. 4 shows the general data retrieval sequence in Harmonic Interleaving scheme. In this figure, $\mathcal{L}_{lower} = \{L_1, L_2\}$ and $\mathcal{L}_{upper} = \{L_3, L_4\}$. We are to read segment 3. Based upon the current network bandwidth availability, data blocks in L_1, L_2, L_3 and L_4 are selected for transmission. The total time for data read here consists of (i) time to read lower layer blocks (T_1), (ii) seek time from the end of segment 3 in a lower layer to the start of segment 3 in an upper layer (T_i), (iii) time to read upper layer

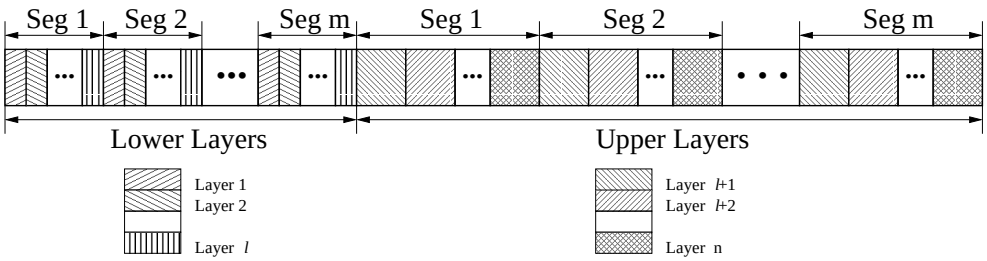


Figure 3: Harmonic Interleaving Placement

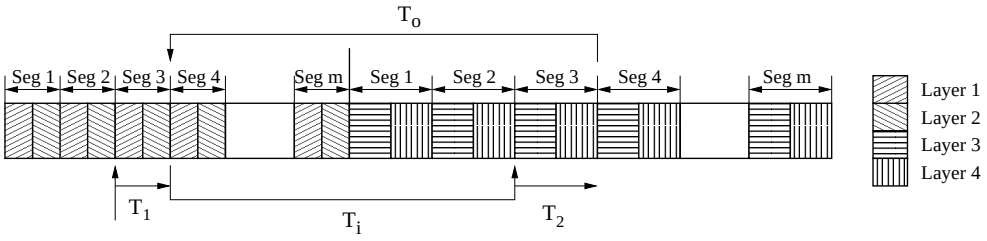


Figure 4: Retrieval Sequence in Harmonic Interleaving

blocks (T_2) and (iv) seek time from the end of segment 3 in the upper layer to the start of segment 4 in the lower layer (T_o). $_i$ and $_o$ is used to represent *in-sweep* and *out-sweep* of the disk head. If only lower layers are requested, T_i, T_2 and T_o are not necessary. Since the disk head skips $m-1$ segments during T_i and m segments during T_o, T_i and T_o are approximately the same. Let C_o and C_i be the cylindrical distance in T_o and T_i. Then, the expected value of C_o and C_i is $(C_{lower} + C_{upper})/2$. Subsequently, T_i and T_o can be formulated as $T_i \approx T_o = T_{seek}(\frac{C_{lower}+C_{upper}}{2})$. Assuming that up to the i^{th}-layer belonging to $\mathcal{L}_{upper}$ are selected for transmission, we can calculate T_2 as $T_2 = \frac{(\sum_{j=l+1}^{i} D_j)/m}{B} = \frac{\sum_{j=l+1}^{i} D_j}{mB}$. Throughout the playback, data blocks belonging to layers in $\mathcal{L}_{lower}$ are retrieved once and only once. Hence, the aggregated time of T_1 can be calculated as $C_{lower} \times T_{cyl}$. Therefore, the total time to read a file in the *Harmonic Interleaving* placement scheme is computed as follows: $T_{total} = m\left(T_1 + \sum_{j=l+1}^{n} \alpha_j (T_i + T_2 + T_o)\right) = C_{lower} \times T_{cyl} + m\sum_{j=l+1}^{n} \alpha_j \left(\frac{\sum_{k=l+1}^{i} D_k}{mB} + 2 \cdot T_{seek}(\frac{C_{lower}+C_{upper}}{2})\right)$

As we can see from the above formula, when layers in $\mathcal{L}_{upper}$ are frequently selected for transmission the Harmonic Interleaving placement scheme is expected to show large overhead, while only layers in $\mathcal{L}_{lower}$ are selected mostly it is expected to show little overhead.

4. PERFORMANCE EXPERIMENT

We examine the efficiency of each file organization technique. We generate synthetic trace for network bandwidth variability and transform the trace into a sequence of layers. This sequence of *selected* layers are fed to the disk. We examine the utilization of the disk under different file organization techniques. It is very important that the analytical model for each file organization technique properly reflect the behavior of the physical disk. We examine the disk utilization obtained from the analytical models as well as the physical experiment. The disk model used in this experiment is the IBM ultrastar 36LP DPSS-309170. To obtain the bandwidth variability, we used network simulator, *ns*. Fig. 5 illustrates the network topology in our network simulation. There are four sources and four sinks. Two sources generate TCP flow. The two network pairs $\{6, 8\}$ and $\{7, 9\}$ are ftp server and ftp client pairs. The other two sources generate real-time video traffic. The two sources which generate the multimedia streaming traffic adaptively change the number of layers to cope with the congestion status of the subnet. We assume that these nodes control the transmission rate using RAP protocol[5]. To examine the network bandwidth trace under various different network conditions, we use three different bottleneck bandwidths: 6,000 Kbits/sec, 2400 Kbits/sec and 1440 Kbits/sec. TCP Sack1 protocol is used. Drop tail management is used in the bottleneck queue. The file used in our experiment is 40 min long and partitioned into 5 layers. The bandwidth requirement for each layer should be carefully chosen so that the file can support a wide variety of connection bandwidth. Table 2 shows the bandwidth requirement for each layer used in our experiment. This partition strategy is based upon Helix Producer user's guide[2]. We compare the disk utilization of indi-

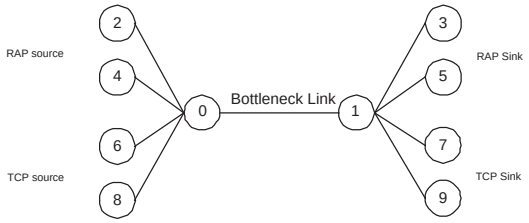
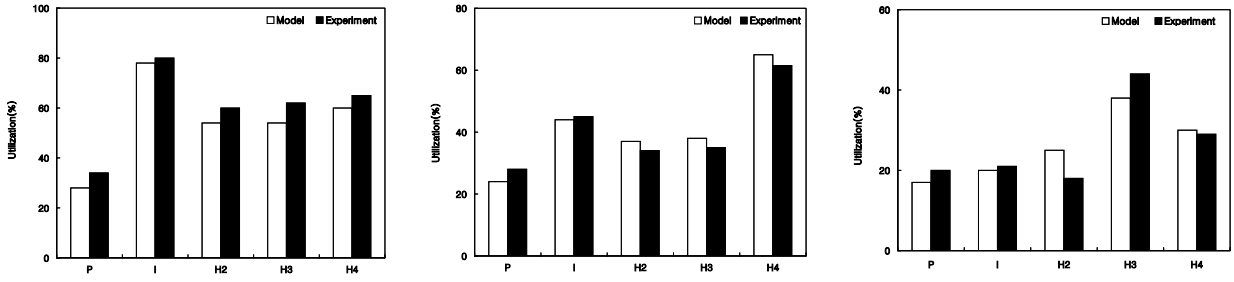


Figure 5: Network topology



(a) 6,000 Kbps (b) 2,400 Kbps (c) 1,440 Kbps

Figure 6: Disk utilization under different bottleneck links

Table 2: Partition of Layers

Layer	B/W	Cumulative B/W	Network Medium
1	34 Kbps	34 Kbps	56 Kbps Dial-up
2	46 Kbps	80 Kbps	128 Kbps Dual ISDN
3	270 Kbps	350 Kbps	384 Kbps DSL or Cable Modem
4	350 Kbps	700 Kbps	768 Kbps DSL or Cable Modem
5	800 Kbps	1500 Kbps	1.5 Mbps or over DSL

vidual placement strategies under different network settings. We use three different bottleneck bandwidths between node 0 and node 1 in Fig. 5: 1440 Kbits/sec, 2400 Kbits/sec and 6,000 Kbits/sec. In *Harmonic Interleaving*, layers are partitioned into two groups. To examine the performance of *Harmonic Interleaving*, we examine the performance under three different layer grouping policies: H_2, H_3 and H_4. H_2 partitions the layers into two groups $\{L_1, L_2\}$ and $\{L_3, L_4, L_5\}$. H_2 is intended to be used for 120k Dual ISDN connections. H_3 partitions the layers into two groups $\{L_1, L_2, L_3\}$ and $\{L_4, L_5\}$. H_3 is used for 384 Kbits/sec DSL connections. H_4 partitions the layers into two groups $\{L_1, L_2, L_3, L_4\}$ and $\{L_5\}$. H_4 is used for 768 Kbits/sec DSL connections.

Fig. 6 illustrates the relationship between the placement strategy and disk utilization under different bottleneck bandwidths. When network capacity is 6000 Kbits/sec, Interleaving exhibits the best disk utilization. This is because there is sufficient network bandwidth availability and thus most layers are transported. When the bandwidth of a bottleneck link is 2,400 Kbps, each of the four sessions is allocated on average 800 Kbits/sec data rate and hence up to layer 4 is transferred most of the time. In this case, placement strategy H_4 yields the best performance. When the bandwidth of a bottleneck link is 1,440 Kbps, placement strategy H_3 yields the best performance.

5. CONCLUSION

Legacy file organization of multimedia contents places the data blocks in a temporal order. This placement strategy may not yield optimal performance in a scalable streaming environment. We find that organizing the file data blocks solely in a temporal order or via layer major order does not properly exploit the disk performance. In this work, we propose a novel file organization technique, *Harmonic Interleaving* and developed an elaborate model to capture the behavior of the disk. In our experiment, we examine the disk utilization via a physical experiment and verify that our disk model closely captures the physical behavior. Via simulation and physical experiment, we show that harmonic placement scheme yields superior disk utilization on a moderately loaded network.

6. REFERENCES

[1] Mon-Song Chen, Dilip D. Kandlur, and Philip S. Yu. Optimization of the grouped sweeping scheduling (gss) with heterogeneous multimedia streams. In *Proceedings of the first ACM international conference on Multimedia*, pages 235–242. ACM Press, 1993.

[2] Real Networks. Inc. Helix producer user's guide, June 2002.

[3] D.R. Kenchammana-Hosekote and J. Srivastava. Scheduling Continuous Media on a Video-On-Demand Server. In *Proceedings of International Conference on Multimedia Computing and Systems*, pages 19–28, Boston, MA, May 1994. IEEE.

[4] P. Rangan, H. Vin, and S. Ramanathan. Designing an on-demand multimedia service. *IEEE Communication Magazine*, 30(7):56–65, July 1992.

[5] R. Rejaie, M. Handely, and D. Estrin. Rap: An end-to-end rate-based congestion control mechanism for real-time streams in the internet. In *Proceedings of IEEE Infocom*, pages 1337–1345, New York, NY, USA, March 1999.

[6] Renu Tewari and Richard King and Dilip Kandlur and Daniel M. Dias. Placement of Multimedia Blocks on Zoned Disks. In *Proceedings of SPIE West '96*, 1996.

[7] P.K.C. Tse and C.H.C. Leung. Improving multimedia systems performance using constant-density recording disks. *Multimedia Systems*, 8(1):47–56, January 2000.