Seamless Retrieval of Digital Continuous Media in Hierarchical Storage System*

Youjip Won
Division of Electrical and Computer Engineering
Hanyang University, Seoul, Korea, zip: 133-791
`yjwon@email.hanyang.ac.kr`

Abstract

In massive scale video streaming server, it is necessary to update the contents of the disk storage in regular fashion, e.g. daily, or weekly. This is particularly because the prohibitively huge amount of data maintained in the massive scale server cannot be stored solely on the secondary storage. Operation of retrieving the data blocks from the disk for streaming purpose bears tight timing constraints. Each video stream is assigned a certain amount of main memory buffer to synchronize the asynchronous disk retrieval operation and synchronous playback operation. The disk write operation for content update may harm the QoS of the on-ging stream. Given that the temporal unavailability of the streaming service is not acceptable, we investigate the impact of writing the data blocks from the tape drive to the disk over on-going video playbacks. We propose to dedicate additional amount of memory buffer to individual stream to mitigate the interference of non-real-time disk write operation(content update) with the on-going video streaming. Based on the characteristics of the buffer requirement of the continuous media streaming, we characterize the buffer requirements for individual streams coexisting with content update operation. The main contribution of this work is to precisely indentify the amount of memory buffer for individual streams to incorporate the content update operation. We visualize the impact of content update operation over buffer space requirement via simulation based experiments.

Key Words: Multimedia, Streaming, Retrieval, Hierarchical Storage

1 Introduction

In this work, we focus on the issue of digital multimedia stream retrieval with the inflow of data blocks from the tertiary storage to secondary storage. Hierarchical storage system has disk subsystem as secondary storage and the tertiary storage which is either tape library or optical juke box. Our current effort is focused on the situation where the contents of the secondary storage are updated regularly(everyday, once a week, etc). This regular update activity is natural when user favors the fresh and new title. We can observe the similar phenomenon in video rental store[vid92].

---

*This work was supported by grant No. 1999-1-303-001-3 from the Interdisciplinary Research Program of the KOSEF.

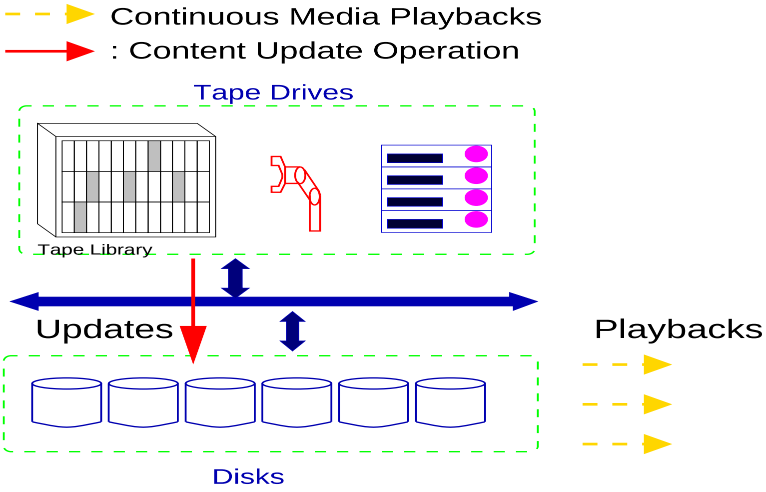

Figure 1: Continuous Media Retrieval and Contents Update in Hierarchical Storage System

To provide *jitter-free* video streaming from the server, a certain amount of memory buffer needs to be allocated to synchronize the asymptotic disk retrieval operation and synchronous playback operations. Any disk I/O operations which are not related to ongoing playbacks can harm the timeliness of the playback related data blocks. The call admission policy for streaming service during the update operation needs to consider the impact of content update operation such that the quality of the streams are not affected by the non-playback related operation. In this article, we describe our approach to maintaining the buffer space for streaming applications under the coexistence of the video stream and the content update operation in hierarchical storage system. Fig. 1 illustrates the hierarchical storage system which has disk subsystem as secondary storage and tape library as tertiary storage. Update operation is performed by reading the data blocks from the tape drive onto the disk.

Recent years have seen a number of studies discussing the issues in multimedia server design[CKY93, KHS94, RVR92, WSar, OBRS94, PD93, Ant96]. Major thesis of these works are how to provide *continuity* in playback operation when data blocks are retrieved from the disk subsystem. There were a number of studies examining the usage of tertiary storage in multimedia server[KDST95, Gha94, Che94, WSZ, Ath97]. A few articles dealt with the scheduling issues of mixed workloads[RW92, RNM$^+$98] in multimedia systems.

2 Buffer Requirement of Multimedia Object Retrieval

In retrieving multimedia data blocks from a disk, a certain amount of buffer space needs to be dedicated to individual streams to synchronize the asynchronous disk retrievals and synchronous playback operations(streaming). Won et al.[WSar] characterizes the invariant in the buffer requirement of the video streaming, which enables us to provide a general framework in determining the buffer size for individual stream. To support multiple streams simultaneously, a set of disk retrieval(or read) operations need to be executed periodically and individual disk read operations

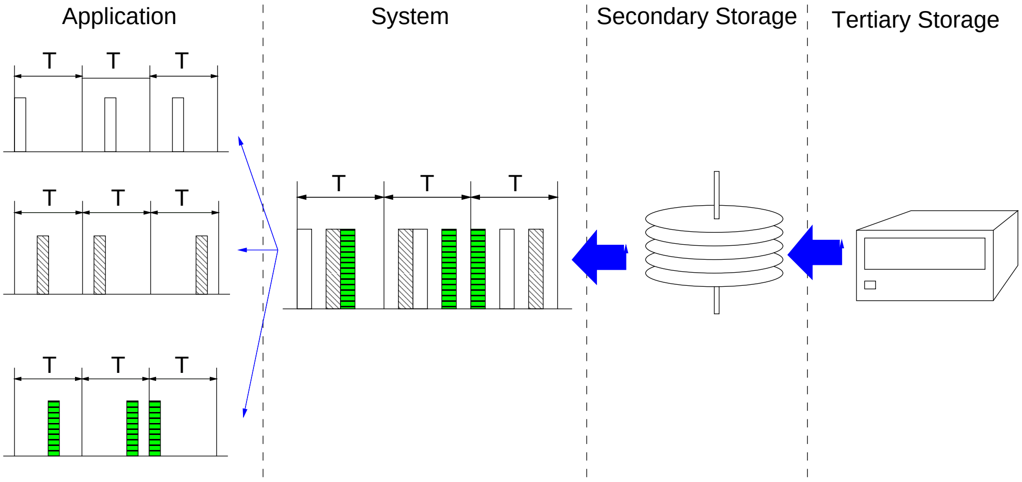

Figure 2: Retrieval and Playbacks and Content Update Operations

have to be completed by their respective firm deadlines. To guarantee the completion of retrieval operation by its respective deadline, a certain amount of system resources such as *disk bandwidth* and *main memory* have to be reserved. We provide Fig. 2 to help the understanding. Fig. 2 illustrates that three streams are fed the data blocks from the disk. The minimum requirement to provide continuity to individual streams is that *the amount of blocks read from the disk during time interval T(sec) should be larger than the amount of blocks that is required by individual streams for T interval.* The total buffer space requirement in supporting a set of streams can be formulated as in Eq. 1[WSar]. T, $\mathbf{s}$, n, and r_i denotes length of the period, a set of streams, the number of streams in $\mathbf{s}$, and the playback rate(Bytes/sec) for stream i, respectively. We use $\mathcal{M}(\mathbf{s})$ to denote the total amount of buffer to support the streams $\mathbf{s}$ in Eq. 1.

α_* and κ_* carry special meanings in this formula. Depending on the disk scheduling strategy of the underlying disk subsystem, the interval between the successive read bursts for a stream in consecutive cycle may vary. For example, this interval can be as large as 2T in worst case, or can be 0(they are consecutive) in SCAN disk scheduling algorithm. Thus, the buffer should maintain the data blocks for 2T(sec)'s playback(or streaming). However, in case of FIFO scheduling strategy, the interval is always T and thus the buffer size is equal to the amount of data for T(sec)'s playback. α_* is 2 and 1 for SCAN and FIFO scheduling strategy, respectively.

Fig. 3 illustrates the possible effect of interval variance. For each interval, disk reads the data blocks for next cycle, T (sec). If the read bursts are more than T sec apart for a stream, the application will suffer from jitter from when the data blocks are depleted T1 in Fig. 3 until the arrival of the data blocks from the disk, T2 in Fig. 3. We use α_* as the multiplication constant to precisely compute the main memory buffer capacity camouflaging the interval variance between the two successive bursts of disk retrievals.

κ_* denotes the maximum disk overhead in retrieving the data blocks for a set of given streams, $\mathbf{s}$. The overhead in this context includes *seek time*, *rotational delay*, etc. The total fraction of time spent on this overhead is also governed by disk scheduling strategy. Let us review the disk head

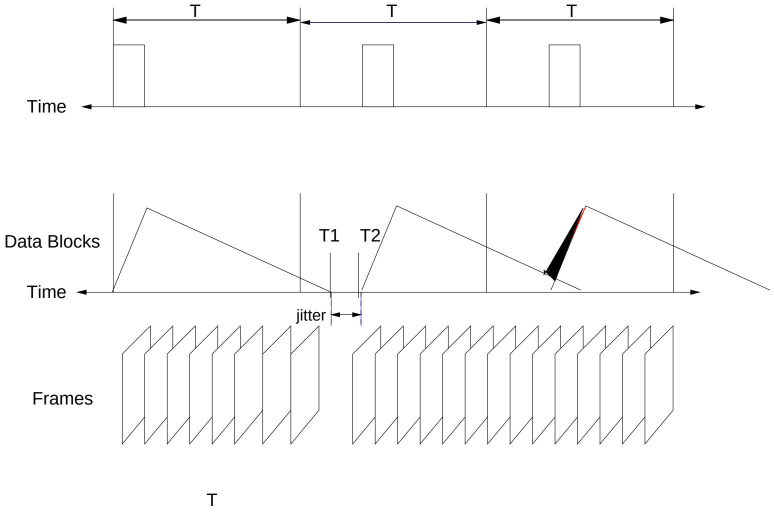

Figure 3: Interval Variance and jitter caused by the respective interval variance between the read bursts

movement charateristics in a single cycle, T. In case of FIFO scheduling strategy, the head can sweep the disk platter $n-1$ times in worst case. In SCAN scheduling strategy, the head sweeps the disk platter just one time with several intermittent stops. Interested readers are refered to Won et al[WSar].

With these two coefficients, α_* and κ_*, we can establish the following equation(Eq. 1) to express the total buffer size requirement for multimedia playbacks.

$$\mathcal{M}(\mathbf{s}) \geq \alpha_* \cdot \kappa_* \frac{\sum_{i=1}^{n} r_i}{B_{max} - \sum_{i=1}^{n} r_i} \tag{1}$$

Eq. 1 defines the lower bound of memory buffer in supporting a set of playbacks. Assuming that main memory is more valuable resource compared to other resources in the system, scheduler(or operating system) should be stringent upon allocating the buffer space to individual streams as long as it can satisfy Eq. 1.

3 Incorporating Content Update Operation in Streaming Environment

Our prime objective in this work is to characterize the buffer requirement for individual streams when disk is also required to incoporate the data blocks from the tertiary storage system(Fig. 2). Important underlying assumption is that the I/O subsystem of the server is configured such that the I/O interconnect is not congested and thus each I/O device can transfer the data at its full capacity. Henceforth, we focus our effort on characterizing the data transfer mechanism between the tape drive and the disk and do not consider the effect of possible traffic shaping caused by I/O interconnect, e.g. delay in I/O queue, etc. Ultrawide SCSI supports upto 20 MBytes/sec and 32 bit 33 MHz PCI can transfer the data at 132 MBytes/sec. Several vendors in this field have formed a group to devise the new I/O interconnect architecture standard, e.g. Storage Area Network, Next Generation I/O(NGIO), Virtual Interface Architecture. These interconnect types provide further abundant communication bandwidth between the host and I/O subsystem by point-to-point connection paradigm, which enables the server and storage vendors to build hierarchical storage system in more cost effective manner.

Let's consider the situation where the video server supports video streams and tape drive loads the new contents to the disk subsystem, concurrently. For each cycle in Fig. 2, the disk not only retrieves the data blocks for streaming but also writes the data blocks supplied from the tape drives. A certain amount of memory buffer needs to be allocated in transfering data from the tape drive to disk. The purpose of this memory section is to provide a temporary buffer space between these two I/O devices. If peer-to-peer communcation is enabled, this memory section may not be needed. In this work, we do not include this buffer space in computing the buffer space used between tape to disk data transfer since the size of this buffer is not affected by the disk subsystem utilization. Given that the overhead of *stop* and *resumption* in tape winding is non-trivial, we presently assume that tape drive reads the data blocks at the sustained rate and disk is required to write the data blocks from the tape drive without blocking the tape drive operation.

For a given set of streams, $\mathbf{s}$, T is computed such that the amount of data blocks read during the interval T for each stream is just as much as the data blocks displayed during T. When the server operates at the *streaming only* situation, making T longer than necessary implies the wastage of valuable buffer memory. However, when there is incoming data blocks to the disk from the tape drive, situation becomes rather different. Under stringent buffer memory allocation, and subsequent slim slack in cycle length T, it will take more than T(sec) to read the data blocks for T sec' playbacks, when incorporating disk write operations. Consequently, the streaming application(s) will suffer from depletion of data blocks. To mitigate the interference caused by the content update operation, the disk needs to read the larger amount of data blocks from the disk, and the respective cycle length T needs to be properly lengthened.

Let T_{cu} be the new cycle in Fig. 2 during which disk also handles the data blocks loaded from the tape drive in a certain sustained rate. Our objective is to characterize the constraints for T_{cu} and subsequently to find the appropriate buffer size to support seamless streaming with content update operation. Let us introduce a number of variables which is to be used in our formula. n_i and n_{cu} denotes the number of data blocks read for stream i and the number of data blocks to be written to the disk during interval T_{cu}. B_{max} and b denotes the maximum transfer rate of the disk and the disk block size. r_{cu} is the sustained data rate(Byte/sec) of the tape drive. The first constraint is that the amount of data read during T_{cu} for stream i should be larger than the amount of data consumed during the same length of time; Eq. 2.

$$T_{cu} r_i \leq n_i b \tag{2}$$

The second constraint is that it should take less than or equal to T_{cu} to retreive the blocks for T_{cu}'s playback for all streams. To formulate this constraint, we introduce new notation δ_i and

δ_{cu} to denote inter-stream head repositioning overhead and head movement overhead to content update operation . Eq. 3 illusrate this contraint.

$$T_{cu} \quad \geq \quad \sum_{i=i}^{n} \left(\frac{bn_i}{B_{max}} + \delta_i \right) + \frac{bn_{cu}}{B_{max}} \tag{3}$$

$n_{cu}b$ is the amount of data (Byte) loaded from the tape drive during T_{cu} (sec) and thus is equivalent to $T_{cu}r_{cu}$. Note that δ_i's and δ_{cu} is disk scheduling dependent factor and we use $\mathcal{O}(\mathbf{s}_{cu})$ to denote generic total disk head movement overhead in reading the data blocks for n streams and content update. Solving the Eq. 2 and Eq. 3, we can finally obtain the required buffer space for individual streams in the content updating situation as in Eq. 4. We can obtain the new cycle T_{cu} by plugging in the n_i's of n_{cu} into Eq. 3.

$$\mathbf{n}_{cu} \geq \frac{\alpha_* \mathcal{O}(\mathbf{s}_{cu})\mathbf{r}}{\frac{b}{B_{max}}(B_{max} - (\sum_{i=1}^{n} r_i + r_{cu}))} \tag{4}$$

In Eq. 4, $\mathbf{n}_{cu}$ and $\mathbf{r}$ denote the array of the number of blocks to be read per cycle and playback rate for each stream. The total buffer size $\mathcal{M}_{cu}(\mathbf{s})$ corresponds to $(\sum_{i=1}^{n} n_i)b$.

4 Simulation Study

We perform simulation based evaluation to visualize the impact of the content update operation over buffer size in streaming operation. We assume that streams are all MPEG-1 compressed with 1.5 MBit/sec(or, 187 KByte/sec) data rate. EXA Byte Mammoth drive has 40 GByte of tape cartridge capacity and sustained uncompressed data rate is 3MByte/sec. DLT 7000 drive from ATL has 5MByte/sec sustained data rate. In our simulation, we assume that *SCAN* scheduling policy is used in reading(for streaming) and writing(for content update) of the data blocks at the disk.

Fig. 4 and Fig. 5 illustrates the total buffer size requirement under different number of streams. X axis denotes the number of MPEG-1 streams from the disk. Y axis denotes the respective *total* buffer space requirement. There are three different sustained tape drive transfer rates: 2 MByte/sec, 3 MByte/sec, and 4 MByte/sec. Disk drive is modeled after HP 97560 disk[WSar]. Given these three different tape drive rates, we provide the total buffer space requirement for playbacks with content update operation. We also provide the buffer space requirement *without* content update operation. Since the total buffer space requirement for individual streams has the form of $O(\frac{1}{B_{max} - \sum r_i})$, it increases very rapidly as the disk bandwidth utilization reaches its full capacity, i.e. $\sum r_i \rightarrow B_{max}$.

In Fig. 4, to support 70 streams with 4 MByte/sec tape drive, the memory buffer requirement increases about 150%. On the other hand, if the main memory buffer space is configured to support upto maximum of 70 streams in *streaming only* environment, the number of supportable streams decreases down to 54 when system performs content update operation with 4 MByte/sec tape drive. This is about 20% reduction of the server streaming capability.

In summary, it is found that the content update operation interferes with the ongoing streaming and significantly larger buffer space is required to nullify the impact of non-realtime content update operation. Given the traffic intensity variation observed in other fields, e.g. telephone company, web traffic[KWar], we carefully conjecture that it is not unreasonale to perform on-line content update operation during off-peak hours since traffic intensity decreases more than 20% during low traffic hours from the peak hours.

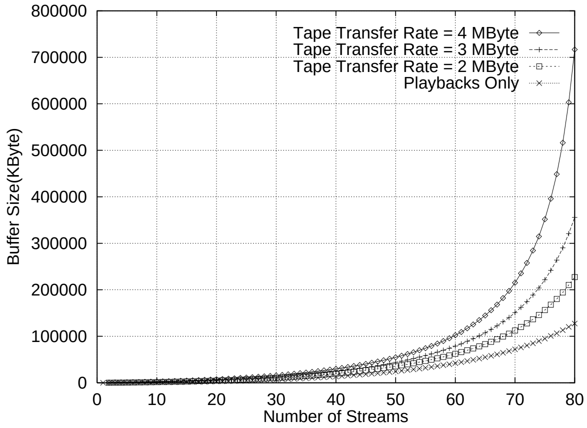

Figure 4: Buffer Requirement of Playbacks with On-going contents Replacement

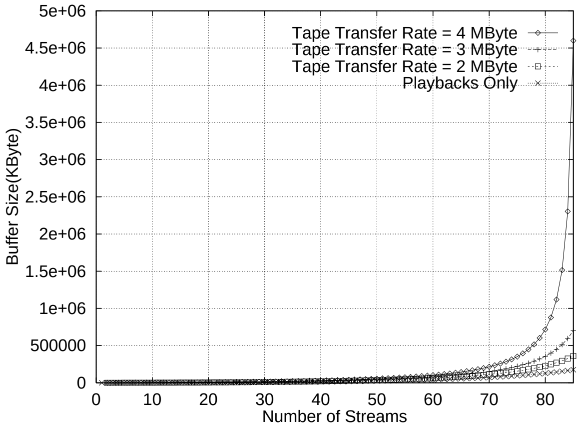

Figure 5: Buffer Requirement of Playbacks with On-going contents Replacement

5 Conclusion

In massive scale video streaming server, regular update of the contents in the secondary storage is indispensable tasks. Given that the temporal unavailability of the streaming service is not acceptable, we investigate the impact of content update operation over the continuity of on-going streams. Each stream is assigned a certain amount of buffer to synchronize the asynchronous disk retrieval operation and synchronous playback operation. To mitigate the interference of non-real-time disk write operation(content update) with on-going playbacks, additional amount of buffer space is required. Based on the characteristics of the buffer requirement of the continuous media streaming, we formulate the buffer requirements for streaming operation coexisting with content update operation. It is found that significanly large amount of buffer is required to support same number of streams along with content update operation. However, given the *peak* to *low* traffic intensity ratio observed in other fields, i.e. more than 5:1, we carefully conjecture that it is not unreasonable to update the content of the disk on-line. By precisely identifying the enhanced buffer space requirement for each stream with content update operation, our work makes a significant contribution in designing a hierarchical storage based continuous media server.

References

[Ant96] Antine Mourad. Issues in the design of a storage server for video-on-demand. *Multimedia*

Systems, 1996(4):70–86, 1996.

[Ath97] Athanassios K. Tsiolis and Mary K. Vernon. Group-Guaranteed Channel Capacity in Multimedia Storage Servers. In *SIGMETRICS 97*, 1997.

[Che94] Ann Louise Chervenak. *Tertiary Storage: An Evaluation of New Applications.* PhD thesis, University of California at Berkeley, December 1994.

[CKY93] Mon-Song Chen, Dilip D. Kandlur, and Philip S. Yu. Optimization of the grouped sweeping scheduling(gss) with heterogeneous multimedia streams. In *ACM Multimedia '93*, pages 235 – 242, 1993.

[Gha94] Ghandeharizadeh, S. and Shahabi, Cyrus. On Multimedia Repositories, Personal Computers, and Hierarchical Storage. In *Conf. Proc. ACM Multimedia*, pages 407–416, 1994.

[KDST95] Martin G. Kienzle, Asit Dan, Dinkar Sitaram, and William Tetzlaff. Using tertiary storage in video-on-demand servers. In *Proc. of IEEE COMPCON '95*, 1995.

[KHS94] D.R. Kenchammana-Hosekote and J. Srivastava. Scheduling Continuous Media on a Video-On-Demand Server. In *Proc. of International Conference on Multi-media Computing and Systems*, Boston, MA, May 1994. IEEE.

[KWar] K. Kand and Youjip Won. "web server capacity planning". *IEEE Transactions on Knowledge and Data Engineering*, To Appear.

[OBRS94] B Ozden, A. Biliris, R. Rastogi, and Avi Silberschatz. A Low-Cost Storage Server for Movie on Demand Databases. In *Proc. of VLDB '94*, 1994.

[PD93] Lougher P. and Shepherd D. The design of a storage server for continuous media. *The Computer Journal*, 36(1):32–42, 1993.

[RNM+98] Y. Rompogiannakis, G. Nerjes, P. Muth, M. Paterakis, P. Triantafillou, and G. Weikum. Disk scheduling for mixed-media workloads in a multimedia server. In *Proceedings of ACM Multimedia '98*, pages 297–302, Bristol, UK, 1998.

[RVR92] P. Rangan, H. Vin, and S. Ramanathan. Designing an on-demand multimedia service. *IEEE Communication Magazine*, 30(7):56–65, July 1992.

[RW92] A. L. N. Reddy and J. Wyllie. Disk Scheduling in a Multimedia I/O system. In *Proc. ACM Multimedia Conf.*, pages 225–233. ACM Press, New York, 1992.

[vid92] Video Store Magazine, December 13 1992.

[WSar] Youjip Won and Jaideep Srivastava. "smdp: Minimizing buffer requirements for continuous media servers". *ACM Multimedia Systems Journal*, To Appear.

[WSZ] Youjip Won, Jaideep Srivastava, and Z.L. Zhang. Optimization configuration of hierarchical storage based continuous media server. In *Proceedings of 16th IEEE Mass Storage Symposium, College Park, Maryland, 1998.*