Handling Sporadic Tasks in Multimedia File System*

Youjip Won
Div. of Elec. and Comp. Engineering
Hanyang University, Seoul, Korea
yjwon@@ece.hanyang.ac.kr

Y.S. Ryu
Div. of Info. and Comm. Engineering
Hallym University, Chunchon, Korea

ABSTRACT
Handling mixed workload in streaming server becomes important issue as integrated file system gets momentum as the choice for next generation file system. In this article, we present novel approach of handling sporadically arriving non-playback related disk request while minimizing its interference with the timely retrieval of data blocks for ongoing playbacks. The main idea of our approach is to extend the length of *period* in period based disk scheduling in multimedia streaming and subsequently a certain fraction of each period can be set aside for handling unexpected I/O request. We develop the analytical model to establish the relationship between the length of period, request arrival rate, request service time, and P(jitter). This model is used to precisely compute the length of period and the respective buffer size. We present the result of simulation based experiment.

Key Word: multimedia, streaming, mixed workload, disk scheduling

1. INTRODUCTION
Characteristic intensive demand on I/O subsystem, e.g, disk bandwidth, disk space, I/O bus bandwidth, to support multimedia playback requires sophisticated modeling of various system resources, to maintain the continuity of the playback. As on-line interactive multimedia services over the Internet, e.g. distance learning, Movie-On-Demand, etc get popular, supporting the multimedia stream with less amount of system resources become one of the prime issues in designing multimedia server software and also in planning the capacity of such server. In this work, we consider the situation where single server provides the streaming service as well as handles the requests for non-playback related data, e.g. text based web documents, gif images, etc. For this type of server, it is mandatory that the server handles multimedia data retrieval along with occasional text or image based file retrieval/storage simultaneously and seamlessly.

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

Recent study showed that it is more economical to have single integerated file system for various media types rather than to have dedicated file system for each media type[5, 4]. For multimedia playback, a certain amount of memory buffer needs to be dedicated to each stream to synchronize the synchronous playback operation and asynchronous disk retrieval operation. To minimize the system resource consumption for multimedia playback, the server software needs to be stringent upon allocating the memory buffer to individual stream. However, since stringent allocation of the memory buffer does not leave much slack for I/O operations other than playback related disk I/O, unexpected I/O task can easily interfere with the on-going playback and subsequently can either cause jitter on on-going multimedia playbacks or suffer from non trivial amount of I/O latency . In this article, we present novel approach in handling this aperiodic I/O request while prohibiting this I/O request to interfere with the on-going multimedia playbacks. A number of research efforts proposed the model for quantifying the system resource requirement for continous media server[1, 6].

2. APERIODICITY IN STREAMING
To maintain *jitter free* playback, a certain amount of disk bandwidth needs to be guranteed for timely supply of data block to the application. To compromise the discrepency between synchronous playback and asynchronous disk retrieval operations, a certain amount of memory buffer is dedicated to individual streams. Most of the disk scheduling technique for continous media streaming adopts period based disk retrieval approach. In period based disk retrieval, time is divided into fixed time interval called *period* such that the data blocks read from the disk during the period are suffient for the stream to playback during the same amoünt of time. Various period based disk scheduling approach are discussed in [8].

Fig. 1 illustrates the playback of multiple continuous media streams from storage subsystem. In the period based scheduling the continuity requirement can be represented with two condition. The first condition is that the number of data blocks retrieved during T should be greater than the amount of data blocks needed for playback for same period of time: $T \cdot r_i \leq n_i b$. This condition should hold for each stream, $i = 1, \cdots, m$. The second condition is that it should take less than T to retrieve the data blocks for all streams: $T \geq \left\{ \sum_{i=1}^{m} \frac{n_i b}{B_{max}} \right\} + \mathcal{O}(m)$. n_i, m, b and r_i are the number of data blocks read during the period T, the number of streams, size of I/O unit, and playback rate of stream i, respectively. B_{max} stands for maximum transfer rate of the disk. $\mathcal{O}(m)$ is the disk movement overhead such as seek and latency in reading the data blocks for m streams. It is important to note that $\mathcal{O}(m)$ is determined by the disk scheduling policy, e.g. *SCAN*, *FIFO*, and etc. Solving these two equations, the total buffer size increases asymtotically following $\mathcal{O}(\frac{1}{1-\rho})$, where ρ is disk utilization, $\frac{\sum_{i=1}^{m} r_i}{B_{max}}$[2, 8].

In Fig. 1, disk subsystem supports three playbacks. Each playback is fed chunk of data blocks required for T period's playback and disk subsystem can retrieve the data blocks for these playbacks within less than T interval. It also services sporadic I/O requests which are irrelavent to multimedia playback. In Fig. 1, there is sufficient amount of idle time in each period and non playback related I/O request can be serviced without affecting the timely retrieval of multimedia data. When the sporadic tasks arrive in burstier manner, storage subsystem may not afford to service them within idle slack of the period and may interfere with retrieval of the multimedia data. Prospective way of resolving this situation is to make the period length T long enough such that there is sufficient idle slack for each period and that *most* of the sporadic tasks can be handled without affecting the retrieval for on-going playbacks. Extending the period entails the increase in the synchronization buffer as well as the increase in the start-up latency of on-demand request. Hence, it is important to carefully determine the length of period to meet the specific requirement of the system.

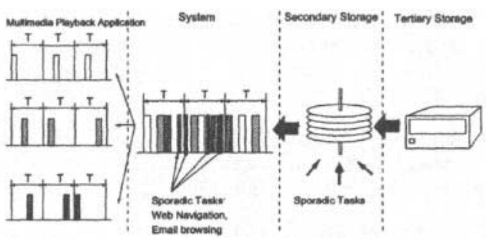

Figure 1: Supporting Multimedia Sessions along with sporadic tasks

3. HANDLING APERIODICITY

In Fig. 2, data blocks for each period needs to be retrieved from the disk at the beginning of each period. At the beginning of the period P_2, three non-playback related tasks were serviced. In the course of servicing these non-plyaback related tasks, retrieving the data blocks for P_2 is delayed by $(T_2 - T_1)$ time and subsequently it entails jitter. While Fig. 2 illustrates data retrieval based on FIFO disk scheduling strategy[3], sporadic tasks can cause jitter independent of disk scheduling strategy. To cope with this problem, we propose to extend the length of period such that occasional non-playback related I/O request does not affect the timely retrieval of the multimedia data blocks. The objective is to determine the length of period such that there is sufficient slack in each period to handle the sporadically arriving

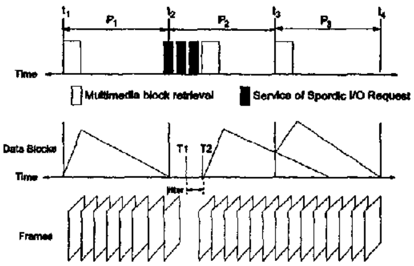

Figure 2: Impact of Handling Sporadic Tasks on Ongoing Playbacks

tasks. In Fig. 1, timeliness of the multimedia block retrieval is not affected by the sporadic tasks since the slack is large enough to handle the sporadic I/O requests.

Our approach provides the statistical guarantee on a maximum probability of jitter, $P(jitter)$ by extending the length of period. $P(jitter)$ is the probability that the sporadic tasks cannot be serviced within the slack of a period, i.e. $N_{sp} \cdot t_{sp} > P_{slack}$ where T_{slack}, N_{sp} and t_{sp} are the length of slack of a period, the number of sporadic tasks in a period and its service time, respectively. We develop a formulation for finding the new buffer size when period T is extend by some fraction to incorporate the sporadic tasks. $\alpha(> 1.0)$ is a coefficient to extend the period. $\alpha = 1.1$ means that $\frac{\alpha-1}{\alpha}$ fraction of a period is set aside to handle randomly arriving disk jobs. Let T be the period which does not contain the slack for handling sporadic tasks. Extending the period by α requires that the data blocks read in a period should be sufficient for αT's playback and time to retrieve that data blocks should be less than T to leave the room for handling sporadic tasks. Thus, finding the new buffer size with extension factor α is equivalent to solving Eq. 1 and Eq. 2.

$$\alpha T r_i \leq n_i b \tag{1}$$

$$T \geq \left\{ \sum_{i=1}^{m} \frac{n_i b}{B_{max}} \right\} + \mathcal{O}(m) \tag{2}$$

Solving these equations, we can obtain the new period length αT and the new buffer size as in Eq. 3. Detailed derivation steps can be found in [7].

$$\mathbf{n} \geq \frac{O(\mathbf{m})\mathbf{r}}{\frac{b}{\alpha B_{max}} \left(B_{max} - \alpha \sum_{i=1}^{n} r_i \right)} \tag{3}$$

Once the length of slack is determined, we can find $P(jitter)$ given the arrival rate, λ and service time, t_{sp} of sporadic task. Our objective is to determine *how much additional buffer is allocated to handle sporadic tasks?*. Thus, actual steps towards solving our problem can be described as follows: *given λ, t_{sp}, and maximum allowable jitter, find α, length of period T and the respective buffer size.*

$$P(jitter) \leq P\left(N_{sp} \geq \left\lfloor \frac{T(\alpha - 1.0)}{t_{sp}} \right\rfloor \right) \tag{4}$$

With Eq. 2, Eq. 3 and Eq. 4, we can obtain the buffer size and the extention factor α. Service time of sporadic tasks t_{sp} can be as large as *full seek* + data transfer time and actually is governed by the disk scheduling strategy adopted in underlying system. [4] proposed an disk scheduling algorithm for mixed workload. Statistical characteristics of L governed by various scheduling policy can be embedded in our framework.

4. SIMULATION

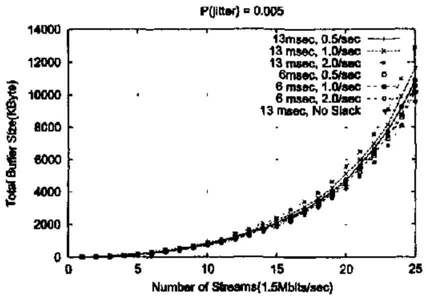

Figure 3: Arrival Rate vs. Buffer Size

There are a number of factors which govern the overhead of handling sporadic tasks: *arrival rate of the sporadic tasks*, *request service time*, and *P(jitter)*. Varying these three parameters, we conduct simulation based experiment and observe the buffer overhead of handling non-playback related sporadic tasks. Disk used in our simulation has 6720 cylinders and the rotational speed is 7200 RPM with 0.6 msec 4KByte block access time. We consider MPEG-1 stram of 1.5Mbits/sec(187KByte/sec) playback rate. In obtaining the buffer size, n, we need to find the extension factor α supporting $P(jitter)$. Finding closed form formulae for the buffer size n in terms of α, $P(jitter)$ from Eq. 2, Eq. 3 and Eq. 4 is non-trivial task. Instead, we take the numerical approach in finding the buffer size, n. Fig. 3 and Fig. 4 illustrate the simulation results. Fig. 3 illustrates the effect of

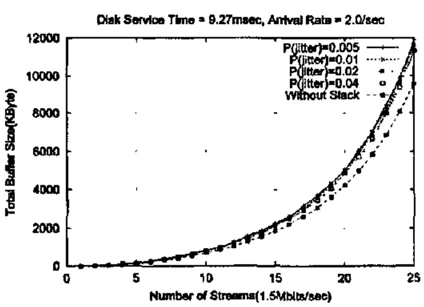

Figure 4: P(jitter) vs. Buffer Size

the request arrival rate of the sporadic tasks and the effect of the request service time on the total buffer size. Maximum allowable jitter probability, $P(jitter)$ is set to 0.005. Request service time is 6 msec and 13 msec. We assume that sporadic tasks arrives in Poisson manner. Simulation is conducted based on three different arrival rate 0.5/sec, 1.0/sec, 2.0/sec. As we can observe, the overhead of handling sporadic tasks increases as there are frequent sporadic request arrivals.

In Fig. 4, we observe the relationship between the buffer size and $P(jitter)$. We vary the $P(jitter)$ as 0.005, 0.01, 0.02, and 0.04. As can be observed, to guarantee smaller jitter probability, larger amount of buffer is required. It is observed that even though we degrade the jitter probability by eight fold(from 0.005 to 0.04), the respective reduction in buffer size is not significant.

5. CONCLUSION

In this article, we present novel approach of handling sporadically arriving non-playback related disk request while guaranteeing a certain level of service quality to ongoing playbacks. Handling mixed workload in the streaming server becomes important issue as integrated file system gets momentum as the choice for next generation file system. This is particulary important when small to medium size streaming server also needs to handle non-trivial amount of http request which is usually for text based web documents, or gif images. The main idea of our approach is to extend the length of *period* in period based I/O scheduling in multimedia streaming and subsequently a certain fraction of each period can be set aside *idle* for the case unexpected I/O request arrival. We develop the framework to obtain the length of period and the respective buffer size to gurantee a certain level of jitter probability given request arrival rate and request service time.

6. REFERENCES

[1] D. Gemmell, H. Vin, D. Kandlur, P. Rangan, and L. Rowe. Multimedia Storage Servers: A Tutorial. *COMPUTER*, 28(5):40–49, May 1995.

[2] D. 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.

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

[4] 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.

[5] P. Shenoy, P. Goyal, and H. Vin. Architectural considerations for next generation file systems. In *Proceedings of ACM Multimedia Conference*, pages pp. 457–467, Orlando, FL, USA, Nov. 1999.

[6] R. Wijayaratne and A. Reddy. Providing qos guarantees for disk i/o. *Multimedia Systems*, 8(1):pp. 57–68, 2000.

[7] Y. Won. Handling mixed workloads in multimedia file system. Technical report, Divison of Electrical and Computer Engineering, Hanyang University, Seoul, Korea, In Preparation.

[8] Y. Won and J. Srivastava. "smdp: Minimizing buffer requirements for continuous media servers". *ACM/Springer Multimedia Systems Journal*, 8(2):pp. 105–117, 2000.