저전력 멀티미디어 재생 기법

최정완[0] 원유집
한양대학교 전자통신전파공학부
(chrys[0], yjwon)@ece.hanyang.ac.kr

PoMP : Power conscious Multimedia Player

Jung-Wan Choi[0] You-Jip Won
Division of Electrical and Computer Engineering Hanyang University

Abstract

Electricity is the prime commodity in mobile device, e.g. smart phone, PDA, MP3 player and etc. This strict restriction on power consumption requirement of the mobile device puts unique demand in designing hardware and software components of the device. In this paper, we address the issue of minimizing the power consumption in retrieving the continuous media data from the disk drive for real-time playback purpose. Different from the legacy text based data, real-time multimedia playback requires that the storage supplies the data block continuous fashion. This may put immense burden on the power scarce environment since the disk is required to be active for the entire playback duration. We develop elaborate algorithm which carefully analyzes the power consumption profile of the disk drive and which establishes the data retrieval schedule for the given playback. It computes the amount of data blocks to read, the length of active and standby period. According to our simulation result, the ARM algorithm exhibits superior performance in continuous media retrieval from the aspect of power consumption to legacy playback scheme.

1. Introduction

1.1 Motivation

Due to the rapid deployment of the mobile devices and the concern of environmental impact of electronic systems, reducing the power consumption of the system becomes one of the most important issues. The disk portion of the power consumption has decreased in past few years from 25% to 10%[10]. However, it is still one of the major components which take up significant fraction of power in entire system. While the disk based storage device, e.g. hard disk and optical disk becomes small enough to be used in mobile devices, the practical usage of which leaves much to be desired due to the stringent power consumption restriction of the mobile device.

In this work, we like to address the problem of minimizing the power consumption of the disk drive for real-time multimedia data playback. The major issue in supporting the real-time playback of multimedia data in the local storage is how to guarantee the continuous flow of data. Round based disk scheduling algorithm has been widely used to support the continuous flow of the data[2, 7]. Preceding works assume that the disk drive operates always in the steady state. We elaborately model the power consumption behavior of the low power disk drive and develop an algorithm which guarantees real-time multimedia playback while minimizing the power consumption involved in operating the storage device.

1.2 Related Works

There have been a number of works regarding the hard disk power management. A set of hard disk requests are grouped into a number of distinct sessions, and disk is shutdown between the sessions. The prime issue here is how to detect the session termination. Simunic et al.[8] used semi-Markov decision process model. Lu et al.[4] adaptively changes the threshold value for detecting session termination. By increasing the number of disk states, it is possible to reduce the power consumption of the disk[10]. Makoto et al.[5] invented a disk storage method which reduces power consumption by allowing a disk controller itself to detect a buffer memory. Furthermore, there have been a number of power management approaches from the operating system's point of view[1,3,6,9].

2. Operational Characteristics of Mobile Disk Drive

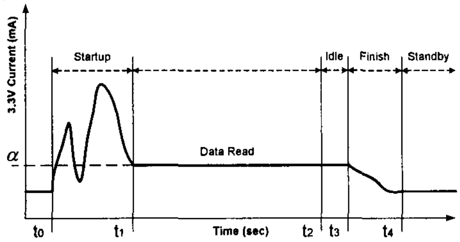
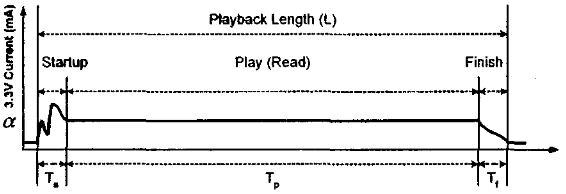


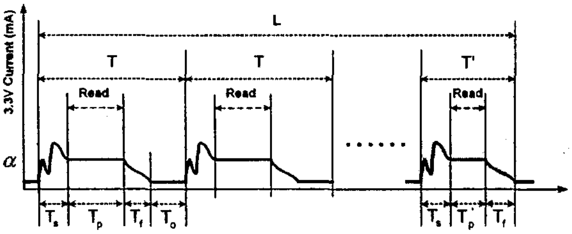
Fig. 1 Schematic View of Power Consumption of Data Read Operation

We model the power consumption behavior of the disk drive as in Fig.1. Retrieving data blocks from the disk drive consists of five phases: Startup, Read, Idle, Finish and Standby phases. The device is initially in Standby state. In the standby mode, it goes into startup mode when the device interface accepts the commands, e.g. I/O requests. Startup phase includes the operation of spin up, focus and tracking. In Read phase, the disk drive transfers data from the disk. When all outstanding I/O requests are serviced, disk drive state changes from Read state to Idle state. In idle phase, the disk head is in the parking position and the spindle is still rotating at the full speed. Disk drive state changes to read when new command arrives. Power saving algorithm has time-out value for idle phase. If no request arrives for time-out period of time, disk drive goes into finish phase where the platter spins down and which eventually goes into standby state. In standby phase, disk head is in the parking position. The spindle stops rotating and all circuitry except host interface are in power saving mode. It is important to note that commodity small size disk based storage device retrieve the data based on startup, read, idle, finish, and standby phases.

3. Formulation



(a) Power Consumption Profile in Normal Playback



(b) Power Consumption Profile in Full Buffering

Fig. 2 Power Consumption Profiles of Two Strategy

Let B^* be the size of buffer. R and r denotes the maximum transfer rate of the disk and the playback rate. The term L denotes the playback length. The length of read phase, T_p, corresponds to $\{B^*/(R-r)\}+\{(Lr-B)/r\}$. This is because the transfer rate from the disk is bounded by the playback rate once the buffer is full. P_s and P_f denote the power consumption in startup and finish phases, respectively. The term α denotes the power consumption rate in read phase. Fig.2(a) shows the power consumption profile in retrieving the multimedia data using normal playback strategy. We can formulate the total power consumption in normal playback, P_N, as Eq. (1)

$$P_N = P_s + P_f + \alpha\left(\frac{B^*}{R-r} + \frac{Lr-B}{r}\right) \quad (1)$$

Another way of retrieving the multimedia data is Full Buffering strategy. In full buffering strategy, the disk reads the data blocks until the buffer is full and immediately goes into standby state. In this strategy, entire playback is made up of sequence of rounds. In each round, active period consists of startup, read and finish phase. The read phase ends when the buffer is full. Fig.2(b) illustrates the power consumption profile in retrieving the multimedia data using Full Buffering strategy. Length of read phase in each round, T_p, corresponds to $B^*/(R-r)$. T_s, T_p, T_f and T_o denotes the length of startup, read, finish and standby phase, respectively. Let T be the length of a round. If the length of read phase is T_p^*, we can compute the length of a round, T, as large as $T_p^* R/r$. Let us formulate the total power consumption in Full Buffering strategy. All the rounds except the last one have the same length and the last round in the playback can be shorter than the preceding ones. Let N be the number of rounds of the same length. We can compute the total power consumption in Full buffering, P_F, as in Eq.(2).

$$P_F = \left\lfloor \frac{L}{T} \right\rfloor \left(P_s + P_f + \alpha\frac{B^*}{R-r}\right) + (P_s + P_f + \alpha T_p')I \quad (2)$$

T' and T_p' denotes time to read the remaining data in the last round and the length of read phase in the last round. I is an index function, which is 0 if $(L \bmod T) = 0$ and 1, otherwise. If playback length, L, is integer multiples of T, i.e. $I = 0$, Full Buffering yields minimum power consumption.

4. Adaptive Round Merge (ARM) Scheduling

In Full Buffering, the disk drive stops reading when buffer is full and then goes into standby phase. This is because efficiency of the power consumption is significantly degraded when the disk continues reading after the buffer is full. However, restarting another rounds accompanies start and finish phase which is just an overhead. If the amount of remaining data is small, it may not be worth reading the remaining blocks in separate round that accompanies startup and finish phase from the perspective of the power consumption. When the buffer is full, we need to decide whether to keep reading the remaining data block in the current round or to enter finish phase. We develop a framework, which determines whether to read the remaining data blocks in the current round or in the separate round. We assume that the power consumption during the standby phase is negligible. The length of the standby period, T_o, can be computed as $(T_p R/r)-(T_s+T_f+T_p)$. It takes $B^*/(R-r)$ to fill the empty buffer. Once the buffer is full, disk can retrieve the data block only at the rate of consumption. Thus, the amount of data blocks read during T_p can be represented as in Eq.(3).

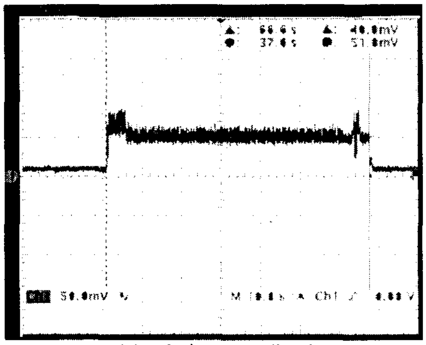
$$B = \begin{cases} T_p R & , if\ T_p \le \frac{B^*}{R-r} \\ \frac{B^*}{R-r}R + (T_p - \frac{B^*}{R-r})r, & otherwise \end{cases} \quad (3)$$

Let B the amount of data blocks retrieved during a single round. Then, the amount of remaining data blocks read in the last round, B_l is corresponded to $Lr - NB$. Power consumption in the last round corresponds to $P_s + P_f + \alpha B_l / R$. If we merge the last round with its immediately preceding one(round merge), we can save the power consumption of startup and finish phases of the last round. However, read phase in the preceding round is extended by B_l / r and additional power consumption, $\alpha B_l / r$ is ensued. We can finally establish function P^* to determine whether to merge the last round to its immediately preceding round or not as in Eq.(4).

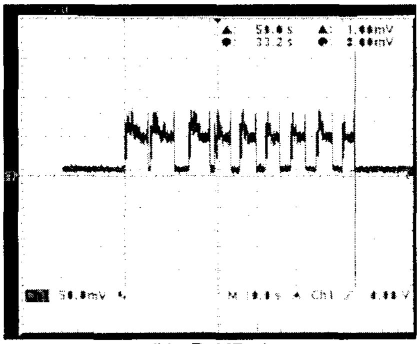
$$P^* = P_s + P_f + \alpha\frac{B_l}{R} - \alpha\frac{B_l}{r} \quad (4)$$

If $P' > 0$, the last round is merged with the preceding one. Otherwise, last B_l data is retrieved in the separate round. This algorithm achieves the minimum power consumption in retrieving given multimedia data.

5. Experiment



(a) windows media player



(b) PoMP player

Fig. 3. Power Consumption Profiles of the playback in two player

We present the results of experiments the effectiveness of the proposed algorithm and compare its performance with that of previous methods. The disk in this experiment is modeled after IBM micro-drive DMDM-10340. The specific character of used file in this experiment are as follows; it is called santafe.mpeg, and then a size, an average bit rate, frame number, and playback length of the file denotes 9.8(MB), 1.246(Mbps), 1535(frame), and 63(sec), respectively.

We conducted experiments with two different multimedia players: PoMP player v0.1 based on the proposed algorithm in this paper and windows media player v7.10. Fig.3 shows the power consumption for the playback in windows media player and PoMP player. As shown in Fig.3(a), a disk drive keeps an active mode during an entire playback. Then, it goes into a standby mode from the active mode after the service for all read commands is completed. Therefore, if windows media player plays a multimedia file, we cannot save the consuming power for the entire playback. As Fig.3(b) shows, PoMP player makes up an entire playback of rounds to 9. This player is sufficiently stableness and regularity against window media player. With windows media player and PoMP player, we find that the length of active states takes approximately 66.6 and 41 sec. Therefore, PoMP player brings approximately 38.5% reduction in power consumption against windows media player.

6. Conclusion

Real-time playback of multimedia data puts unique demand on the disk subsystem. Retrieval of data blocks needs to be scheduled properly so that it can meet the playback deadline of individual data blocks and also minimizes the various overheads, e.g. service startup latency, buffer requires, etc. Power management feature of a low power disk drive opens up a new chance to develop further power efficient multimedia player. However, it adds another dimension of complexity in determining the block retrieval schedule. In this paper, we analyze the power consumption behavior of the low power disk drive and propose scheduling algorithm, Adaptive Round Merge, which guarantees continuous retrieval of data blocks and which minimizes the power consumption for the given data retrieval. Given the power consumption profile of disk drive, data transfer rate of the disk, and playback rate, ARM algorithm computes the size of read burst in a round and the length of standby period. Our algorithm generates the disk operation schedule which determines when to start and when to stop the spindle. In the simulation based experiment, we observe that the ARM algorithm makes significant improvement on power consumption for the given playback. Compared with Normal Playback which does not have power management feature, ARM algorithm can decrease the power consumption by 38.5% with 1 MBytes buffer. With larger size buffer, the reduction becomes even more dominant.

7. Reference

[1] L. Benini, A.Bogliolo, S.Cavallucci, and B.Ricco. Monitoring System Activity for os-directed Dynamic Power Management. In Proc, of IEEE International Symposium on Low Power Electronics and Design, pages 185-190, 1998.

[2] 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 IEEE., Boston, MA, May 1994.

[3] Yung-Hsiang Lu, Luca Benini and Giovanni De Micheli. Operating system directed power reduction. In Proceeding of the International Symposium on Low Power Design, 2000. ACM, New York, NY, USA, 2000.

[4] Yung-Hsiang Lu and Giovanni De Micheli. Adaptive hard disk power management on personal computers. In Proceeding of the IEEE Great Lakes Symposium on VLSI, 1999. IEEE, Los Alamitos, CA, USA, 1999.

[5] Kawasaki Makoto, Ichikawa Yasuhiko, and Ishii Shuichi. Disk Storage Device and Power Supply Control Method for Same Device, JP patent, Nov, 1999.

[6] Dinesh Ramanathan, Sandra Irani, and Rajesh K.Gupta. An Analysis of System Level Power Management Algorithms and their Effects on Latency. IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems, 21(3):291-305, March 2002.

[7] A.L.N.Reddy and J.Wyllie. Disk Scheduling in a Multimedia I/O System. In Proc. ACM Multimedia Conference. Pages 225-233. ACM Press, New York, 1992.

[8] T.Simunic, L.Benini, P.Glynn, and G.De Micheli. Dynamic Power Management of Laptop Hard Disk. In Proc. of Design, Automation and Test in Europe Conference and Exhibition 2000, page 736, 2000.

[9] T.Simunic, L.Benini, P.Glynn, and G.De Micheli. Event Driven Power Management, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 20(7):840-857, July 2001.

[10] IBM Storage Systems Division. Adaptive power management for mobile hard drives. Technical report, IBM Co. Storage Systems Division, San Jose, CA, USA, 1999.