

Power Constraints: Another Dimension of Complexity in Continuous Media Playback^{*}

Jungwan Choi and Youjip Won

Division of Electrical and Computer Engineering, Hanyang University,
17 Hangdangdong Seongdongku, Seoul, Korea
{chrys, yjwon}@ece.hanyang.ac.kr

Abstract. 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.

Keywords: Disk Scheduling, Power Management, Multimedia, Playback, Mobile Device

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%[13]. 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.

Typical approach to reduce the power consumption for hard disk drive is to shutdown the disk when there is no outstanding I/O requests[7]. Disk shutdown algorithms can be classified into two categories: *predictive*[3] and *stochastic*[2]

^{*} This work is supported by KOSEF through Statistical Research Center for Complex System at Seoul National University.

schemes. State transition from *standby* state to operational state entails extra power consumption. Thus, it is required to elaborately model the energy consumption profile for a given workload and to establish the disk operation schedule.

Different from general purpose computer, mobile handheld device usually has dedicated purpose. This leaves us great chance of optimization in various aspect of system design since the workload exhibits rather unique and predictable behavior. Continuous media retrieval from low power disk is not an exception. We believe that it is possible to achieve greater reduction in power consumption by effectively exploiting the workload characteristics and by incorporating it into I/O scheduling.

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[5, 10]. Preceding works assume that the disk drive operates always in the *steady* state. However, disk drive controlled by dynamic power management can adaptively change the state of operation, e.g. *active* or *standby*, with respect to the state of workload [11,13]. While this feature can significantly extend the battery life, it adds another dimension of complexity in scheduling of the multimedia data retrieval. 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. We presently assume that the multimedia file is constant bit rate(CBR) encoded.

1.2 Related Works

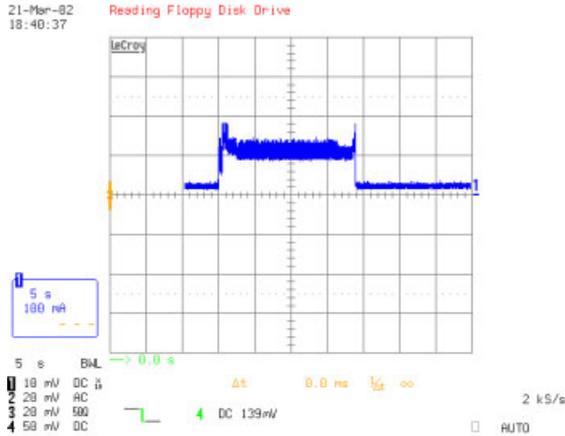
There have been a number of works regarding the hard disk power management. A set of hard disk requests are grouped into a number of distinct sessions, and disk is shutdown between the sessions. The prime issue here is how to detect the session termination. Simunic et al.[11] used semi-Markov decision process model. Lu et al.[7] 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[13]. Greenawalt[4] et al. formulates the relationship between the number of disk state transitions, power consumption, and system performance.

There have been a number of power management approach from the operating system's point of view[1,9,6,12]. Helmbold et al.[8] introduced an algorithm that decides when to spin down the disk of a mobile computer in order to reduce the power consumption. The algorithm called the *share algorithm* dynamically chooses a time-out value as a function of recent disk activity. The above mentioned power management strategy assumes that the device operates in the *generic* workload. When the device(or embedded system) is designed to perform specific task, e.g. MP-3 player, digital camera, personal video recorder, etc.,

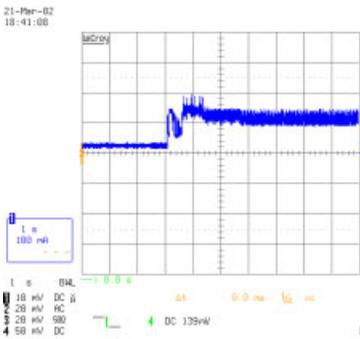
it is possible to further effectively exploit the characteristics of the underlying workload to minimize the power consumption of the storage drive.

2 Power Consumption Profile of Low Power Drive

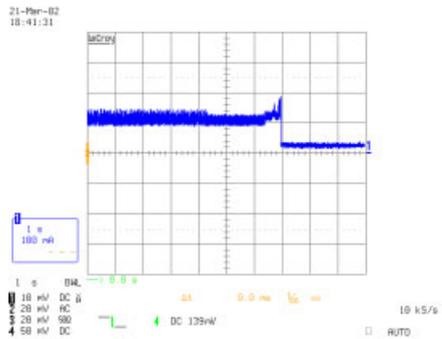
We first examine the power consumption profile of the low power disk drive¹. Figures in Fig. 1 illustrate the power consumption profile of the lower power disk



(a) Power Consumption Profile for entire play-back duration



(b) Power Consumption profile during Startup phase



(c) Power Consumption profile during Finish phase

Fig. 1. Power Consumption Profile of IBM MicroDrive(DMDM-10340)

¹ It is measured using WavePro 950 from LeCroy. Sampling rate is 16 Gigasamples/sec

drive. We measure the power consumption of real-time playback of MP3 file(1.5 Mbits/sec, 20 sec) in the IBM MicroDrive(DMDM-10340). It was measured on Compaq Notebook(ARMADA M700) loaded with Windows XP. Windows media player is used for music playback. Fig. 1(a) shows the power consumption during entire playback. Fig. 1(b) magnifies the power consumption behavior during the start of the playback. We find that spinning up the disk requires extra overhead in power consumption and it takes approximately 1.7 sec before the disk reaches the steady state. Fig. 1(c) illustrates the power consumption profile during the disk shutdown. Disk platter stops rotating when there is no outstanding I/O requests for a certain period of time. According to our experiment(Fig. 1(c)), the length of *finish* phase is approximately 0.8 sec²

We model the power consumption behavior of the disk drive as in Fig. 2. 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.

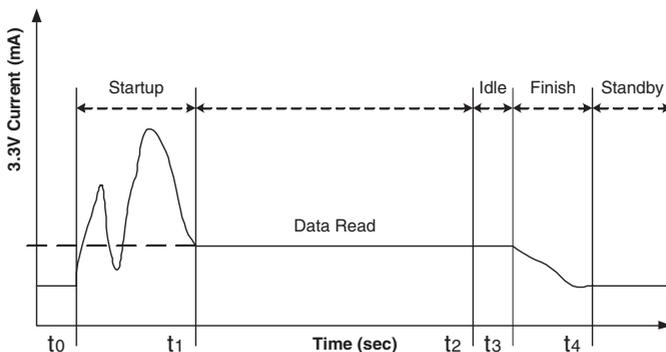


Fig. 2. Schematic View of Power Consumption Profile of Data Read Operation

² We will use the term *phase* and *state* interchangeably.

For real-time multimedia playback, an application issues burst of reads periodically and the disk supplies the data block in periodic fashion conformant to the playback rate. The size of read burst and the interval between the adjacent read bursts need to be carefully determined based upon the available buffer size, tolerable startup latency, disk overhead such as seek and rotational latency[15]. The power saving feature of low power disk drive introduces another dimension of complexity in determining these value due to the existence of *standby* period. *Standby* period should be effectively incorporated into retrieval schedule in order to minimize the power consumption for multimedia playback.

3 Adaptive Round Merge(ARM) Scheduling

3.1 Problem Formulation

Fig. 3 illustrates the multimedia data retrieval operation in desktop environment and mobile environment. The term "desktop environment" is used to denote the disk drive which does not have dynamic power management feature. In desktop environment, the disk stays active during entire playback period. In the low power disk drive equipped with dynamic power management feature, it is possible that the disk drive retrieves the data block in burstier manner and goes into the standby state(right-hand side of Fig. 3). This is primarily to reduce the fraction of *active period* in entire playback.

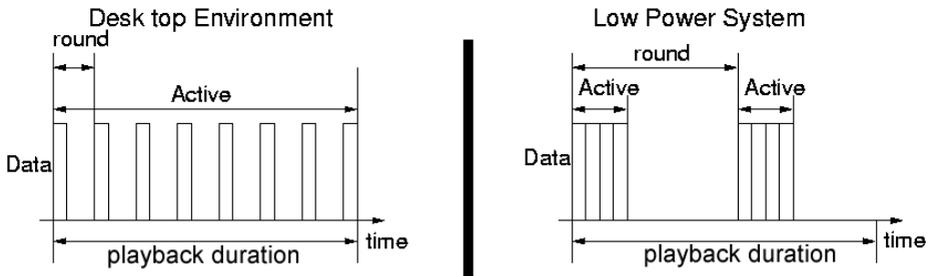
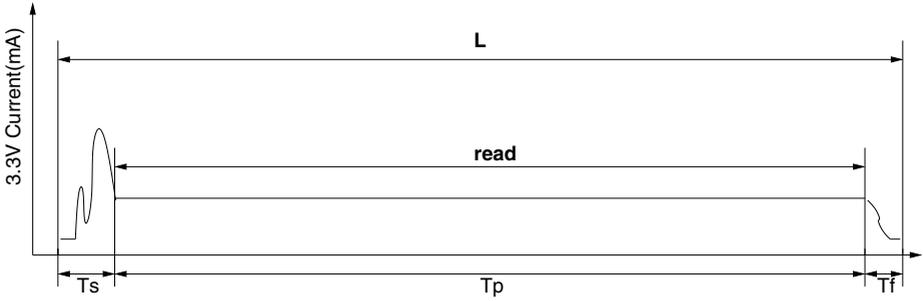


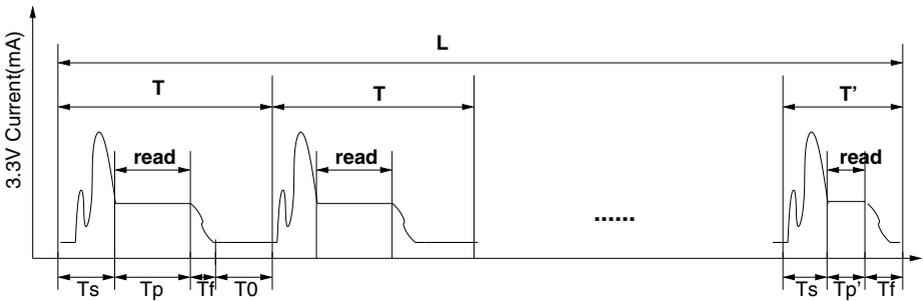
Fig. 3. Multimedia Data Retrieval: Desktop environment vs. Mobile Environment

The Adaptive Round Merge algorithm proposed in this work is designed to minimize power consumption to retrieve multimedia data from the disk. We first formulate the power consumption behavior in multimedia playback. Without power management algorithm(left-hand side of Fig. 3), disk drive is in *read* state during the entire playback period. We define it as *Normal Playback*.

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 $\frac{B^*}{R-r} + \frac{Lr-B}{r}$. This is because the transfer rate from the disk is bounded by the playback rate once the buffer is full. P_s



(a) Power Consumption Profile in Normal Playback Strategy



(b) Power Consumption profile in Full Buffering Strategy

Fig. 4. Multimedia Data Retrieval and Power Consumption Profile

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. 4(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, \mathcal{P}_N , as Eq. 1

$$\begin{aligned} \mathcal{P}_N &= P_s + P_f + \alpha T_p \\ &= P_s + P_f + \alpha \left(\frac{B^*}{R-r} + \frac{Lr - B}{r} \right) \end{aligned} \quad (1)$$

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. 4(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

$\frac{B^*}{R-r}$. T_s, T_p, T_f and T_0 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 $\frac{T_p^*R}{r}$. 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, $\mathcal{P}_{\mathcal{F}}$, as in Eq. 2.

$$\mathcal{P}_{\mathcal{F}} = N(P_s + P_f + \alpha T_p) + (P_s + P_f + \alpha T_p')I \tag{2}$$

$$= \left\lfloor \frac{Lr}{\frac{B^*R}{R-r}} \right\rfloor (P_s + P_f + \alpha \frac{B^*}{R-r}) + (P_s + P_f + \alpha T_p')I \tag{3}$$

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.

3.2 Adaptive Round Merge Algorithm

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 which 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_0 , can be computed as $\frac{T_p^*R}{r} - (T_s + T_f + T_p)$. It takes $\frac{B^*}{R-r}$ 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. 4.

$$B = \begin{cases} T_p R & , \text{ if } T_p \leq \frac{B^*}{R-r} \\ \frac{B^*}{R-r} R + (T_p - \frac{B^*}{R-r})r, & \text{ otherwise.} \end{cases} \tag{4}$$

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 $L \cdot r - N \cdot B$. Power consumption in the last round corresponds to $P_s + P_f + \alpha \frac{B_l}{R}$. 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

Table 1. Adaptive Round Merge Algorithm

```

Algorithm: Adaptive Round Merge Algorithm() {
  i=0;
  compute N;      /* N denotes the number of equal length rounds. */
  compute PN;    /* PN is total power consumption in Normal playback */
  compute PF;    /* PF is total power consumption in Full buffering */
  compute P*;
  IF( PN ≤ PF ) {
    startup();    /* spin up, focus and tracking */
    continue reading; /* Disk is in read state during the playback */
    finish();
  }
  ELSE {
    WHILE(N≠ 0) {
      Dr = L · r - i · B;
      startup();
      read();
      IF(Dr < B AND P* < 0) {
        continue reading;
        finish();
        N--;
      }
      finish();
      N--; i++;
    }
  }
}

```

last round. However, *read* phase in the preceding round is extended by $\frac{B_l}{r}$ and additional power consumption, $\frac{\alpha B_l}{r}$ is ensued. We can finally establish function \mathcal{P}^* to determine whether to merge the last round to its immediately preceding round or not as in Eq. 5.

$$\mathcal{P}^* = \underbrace{P_s + P_f + \alpha \frac{B_l}{R}}_{\text{without Merge}} - \underbrace{\alpha \frac{B_l}{r}}_{\text{with Merge}} \quad (5)$$

If $\mathcal{P}^* > 0$, 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. Table 1 illustrates the details of the algorithm.

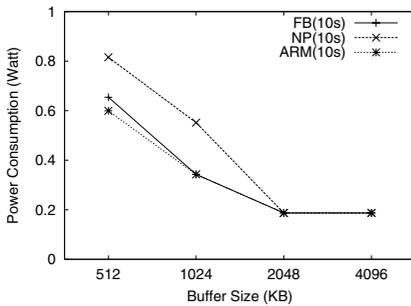
4 Simulation

We examine the effectiveness of our algorithm via simulation based experiment. The disk parameters are modeled after IBM microdrive DMDM-10340. *Startup*

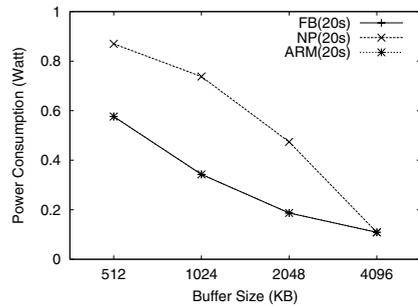
phase is 1 sec long and consumes 0.858 Joule. *Finish* phase is 0.4 sec long and consumes 0.7 Joule. Power consumption in steady state read operation is 0.924 Watt. Disk transfer rate is 41.6 Mbits/sec. The playback rate is 1.4 Mbits/sec[14].

4.1 Buffer Size vs. Power Consumption

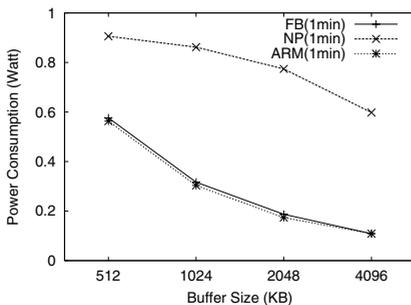
We first compare the power consumption under different buffer sizes, B^* . The X-axis and the Y-axis in Fig. 5 denotes playback length and the power consumption, respectively. As shown in Figures in Fig. 5, power consumption is inversely proportional to the buffer size. With the larger buffer size, the disk can fetch larger amount of data in each round and subsequently power efficiency improves. Let us consider 10 sec playback. With 512 KByte buffer, merging the round with the preceding one in *Adaptive Round Merge* brings approximately 9% reduction in power consumption against the Full Buffering strategy. The effect of round merge manifests itself typically when the playback length is relatively short, typically less than 60 sec.



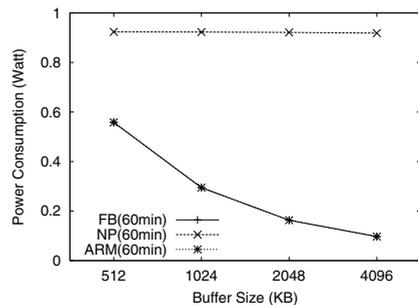
(a) Playback Length = 10 sec



(b) Playback Length = 20 sec



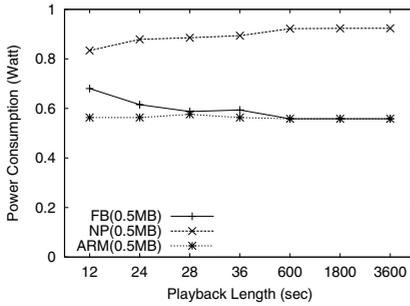
(c) Playback Length = 1 min



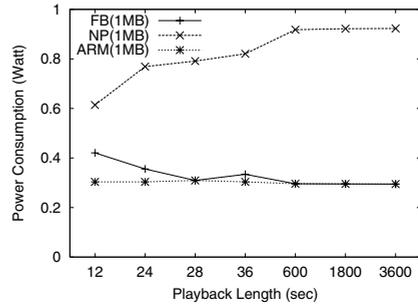
(d) Playback Length = 60 min

Fig. 5. Buffer Size vs. Power Consumption

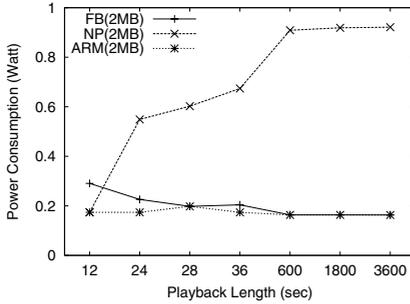
In *Normal Playback* strategy, effectiveness of using larger buffer greatly depends on the length of playback. With short playback (Fig. 5(a), Fig. 5(b)), using the larger size buffer improves power consumption behavior. This is because significant portion of the file can be loaded on to memory when playback length is short. However, when playback is long, e.g. 1 min or beyond, using the large size buffer (upto 4 MByte) does not bring any improvement in power consumption.



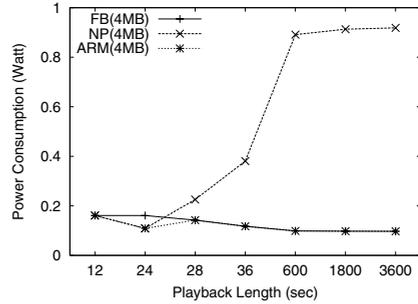
(a) Buffer Size = 0.5MBytes



(b) Buffer Size = 1MBytes



(c) Buffer Size = 2MBytes



(d) Buffer Size = 4MBytes

Fig. 6. Playback Length vs. Power Consumption

4.2 Effect of Playback Length

Fig. 6 illustrates the relationship between the playback length and the power consumption. The X and Y axis denotes the playback length and the power consumption, respectively. We consider four different buffer sizes, 0.5, 1, 2 and 4 MByte. In *ARM* algorithm, it makes the decision of merging the last round

with preceding one based upon Eq. 5. We can observe the advantage of using power saving feature in retrieving the data blocks. In Full Buffering and Adaptive Round Merge scheme, disk drive goes into *standby* mode when the buffer becomes full. The disk drive remains in *standby* mode until buffer becomes empty. In Normal Playback mode, there is no notion of *standby* mode and therefore the power consumption rate converges to the power consumption rate of *read* phase. The advantage of merging the last round with the previous one becomes more visible when playback length is relatively short. With the playback length of 12, 24 and 36 sec, ARM reduces the power consumption by 18%, 28% and 40% against Full Buffering strategy, respectively, with 0.5 MByte Buffer. These values correspond to 9%, 15% and 23% with 1 MByte buffer.

5 Conclusion

Realtime 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 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 60% with 1 MByte buffer. With larger size buffer, the reduction becomes even more dominant. While our simulation parameter is based upon IBM microdrive, this algorithm can be applied to any disk based storage device with power management feature.

References

1. L. Benini, A. Bogliolo, S. Cavallucci, and B. Ricco. Monitoring system activity for os-directed dynamic power management. In *Proceedings of IEEE International Symposium on Low Power Electronics and Design*, pages 185–190, 1998.
2. E.-Y. Chung, L. Benini, A. Bogliolo, and G. D. Micheli. Dynamic power management for non-stationary service requests. In *Proceedings of Design Automation and Test in Europe*, 1999.
3. F. Douglass, P. Krishnan, and B. Bershad. Adaptive disk spin-down policies for mobile computers. *Computing Systems*, 8:381–413, 1995.

4. P.M. Greenawalt. Modeling power management for hard disks. In *Proceedings of the Second International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems(MASCOTS '94)*, pages 62–66, 1994.
5. 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.
6. Yung-Hsiang Lu, Luca Benini, and Giovanni De Micheli. Operating-system directed power reduction. In *Proceedings of the International Symposium on Low Power Design, 2000*. ACM, New York, NY, USA, 2000.
7. Yung-Hsiang Lu and Giovanni De Micheli. Adaptive hard disk power management on personal computers. In *Proceedings of the IEEE Great Lakes Symposium on VLSI, 1999*. IEEE, Los Alamitos, CA, USA, 1999.
8. David P.Helmbold, Darrell D.E.Long, Tracey L.Sconyers, and Bruce Sherrod. Adaptive disk spin-down for mobile computers. *Mobile Networks and Applications*, 5(4):285–297, Dec 2000.
9. 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 Cirtuits and Systems*, 21(3):291–305, March 2002.
10. 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.
11. T. Simunic, L. Benini, P. Glynn, and G. De Micheli. Dynamic power management of laptop hard disk. In *Proceedings of Design, Automation and Test in Europe Conference and Exhibition 2000*, page 736, 2000.
12. 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.
13. IBM Storage Systems Division. Adaptive power management for mobile hard drives. Technical report, IBM Co. Storage Systems Division, San Jose, CA, USA, 1999.
14. IBM Storage Systems Division. Oem hard disk drive specifications for dmdm - 10340/10170 ibm microdrive revision 1.0. Technical report, IBM Co. Storage Systems Division, USA, 1999.
15. Youjip Won and Jaideep Srivastava. "smdp: Minimizing buffer requirements for continuous media servers". *ACM/Springer Multimedia Systems Journal*, 8(2):pp. 105–117, 2000.