Event-Driven Power Management
of Portable Systems

Tajana Simunic[†]   Luca Benini[*]   Giovanni De Micheli[†]

[†] Stanford University
Computer Systems Laboratory
Stanford, CA 94305

[*] DEIS
Universitá di Bologna
Bologna, ITALY 40136

Abstract

The policy optimization problem for dynamic power management has received considerable attention in the recent past. We formulate policy optimization as a constrained optimization problem on continuous-time Semi-Markov decision processes (SMDP). SMDPs generalize the stochastic optimization approach based on discrete-time Markov decision processes (DTMDP) presented in the earlier work by relaxing two limiting assumptions. In SMDPs, decisions are made at each event occurrence instead of at each discrete time interval as in DTMDP, thus saving power and giving higher performance. In addition, SMDPs can have general inter-state transition time distributions, allowing for greater generality and accuracy in modeling real-life systems where transition times between power states are not geometrically distributed.

1 Introduction

Dynamic power management is a widely-employed low-power design technique. Dynamic power management efficiently exploits system resources by controlling their mode of operation and selectively shutting down system components when they are idle. Dynamic power management is profitably exploited at many abstraction levels (refer to [2] for a review). In this paper we focus on system-level power management. The abstract model of the system resources adopted in system-level power management is state-based [9] where each state represents modes of operation that trade off performance for power. State transitions are controlled by commands issued by a *power manager* (PM) that observes the workload of the system and decides the when and how to force power state transitions. We call *power management policy* (*policy* for brevity) the control law adopted by the power manager for deciding upon the state of operation of the system and the system components.

In this paper, we focus on the *policy optimization* problem, which can be summarized as designing the control law that minimizes power under performance constraints (or maximizes performance under power constraint). Several heuristic power management policies based on workload prediction have been investigated in the past [5, 15, 7] but no strong optimality result has been proven. Policy optimization has recently been formulated as a stochastic optimization problem on discrete-time Markov decision processes (DTMDP) [12].

The contribution of this work is to relax two major assumptions made in the DTMDP formulation. First, we move from a discrete-time to a continuous-time model. As a result, we can implement an event-driven power manager, as opposed to a clock-driven manager which is required in the DTMDP model. Event-driven PMs are inherently more power efficient than clock-driven PMs because they make decisions only in response to the changes in the workload and in the state of operation of the system without creating additional activity on each clock cycle when the system is idle. Second, we propose a stochastic model based on Semi-Markov decision processes [13] for the formulation of policy optimization and we describe a procedure for its *exact* solution. The semi-Markov model is considerably more general than the Markov model as it allows general inter-state time distributions instead of requiring geometric or exponential distributions. As a result, it can accurately represent a larger class of systems. The solution of policy optimization for SMDPs is computed in polynomial time by solving a linear optimization problem.

2 Related work

Systems are usually designed for peak performance under a worst-case workload although they typically operate under non-uniform workloads. Power management focuses on minimizing power waste when the system is underutilized. When power state transitions are almost instanta-

neous, the optimum power management policy is trivial: resources should be shut down as soon as they become idle. In many practical cases, however, transitioning to and from a low power state is expensive in terms of time and power. For example, transitioning from a sleep state to an active state can be a slow and power-consuming process, as in the case of hard-disk drives, where the disk must be accelerated to a very high speed [8]. When transitions to and from sleep states are non-instantaneous, the policy optimization problem becomes non-trivial, and effective policies that minimize power without compromising performance are required.

Shut-down policies for hard-disk drives have been extensively studied in the past (refer to [5] for a complete set of references). The *timeout policy* is by far the most common in practice. Timeout policies assume that there is a high probability of having a long idle period if the system has been idle for a longer period. Their main limitation is that they waste power while waiting for the timeout to expire. More aggressive power management policies for hard disks are *predictive*, i.e., they attempt to predict future idle times based on past history [5, 4, 6, 10]. Shutdown is performed as soon as the system is idle, if the predictor estimates that the idle period is going to be long. Several predictive or timeout-based power management policies have been studied for interactive terminals, such as personal digital assistants and electronic clipboards, etc. [15, 7, 1].

A common feature of all the above mentioned approaches is that policies are formulated heuristically, then tested with simulations or measurements to assess their effectiveness. The policy optimization technique proposed by Paleologo et al. [12] formulates a probabilistic system model based on discrete-time Markov decision processes (DTMDP). In contrast with previous work, policy optimization can be solved exactly and efficiently in the DTMDP model.

The main limitations of the stochastic optimization approach proposed in [12] are the assumption of a discrete time setting, and the assumption of geometrically-distributed transition times between states. PM decisions are synchronized to the system clock. This assumption implies that the PM must wake up at every clock tick to issue a command, even when the system is idle, thus wasting power. The assumption of geometrically-distributed state transition times may not hold in many practical cases. For instance, transition times may be deterministic or uniformly distributed.

We generalize DTMDP with the *Semi-Markov decision process* (SMDP). SMDP allows the power manager to choose actions whenever the system state changes in continuous time, instead of at each discrete time increment. Thus the model becomes event-driven which is more appropriate for the event-driven environment found on most computers and embedded systems. In contrast with the *Continuous-time Markov decision process* (CTMDP) [14] where the times spent in a state are exponentially distributed,e Tbe Semi-Markov decision process allows the time between transitions to follow arbitrary probability distribution. The next section introduces the theoretical background of the SMDP.

3 Theoretical Background

In this section we review the Semi-Markov decision process optimization problem [13]. We use upper-case bold letters (*e.g.*, $\mathbf{M}$) to denote matrices, lower-case bold letters (*e.g.*, $\mathbf{v}$) to denote vectors, calligraphic letters (*e.g.*, $\mathcal{S}$) to denote sets, upper-case letters (*e.g.*, S) to denote scalar constants and lower-case letters (*e.g.*, s) to denote scalar variables.
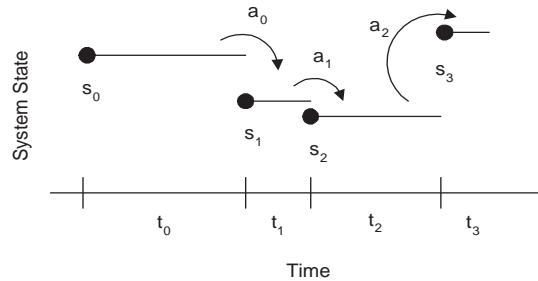

Figure 1. SMDP Progression

In SMDP decisions are made at times of event occurrences, called decision epochs. A sample path consists of a series of decisions taken at each decision epoch $i = 0, 1, 2, \ldots$ Figure 1 shows a sample progression of the SMDP through decision epochs. *Inter-event time set* is defined as $\mathcal{T} = \{t_i, \text{ s.t. } i = 0, 1, 2, \ldots, T\}$ where each t_i is the time between the two successive event arrivals Let $\mathcal{S}$ be a finite system state space, with cardinality S. We denote by $s_i \in \mathcal{S}$ the system state at decision epoch i. Commands are issued whenever the system state changes. We define an *action set* $\mathcal{A}$ with cardinality A. We denote by $a_i \in \mathcal{A}$ an action (or command) that is issued at decision epoch i. When action a_i is chosen in system state s_i, the probability that the next event will occur by time t_i is defined by the cumulative probability distribution $E(t_i | s_i, a_i)$. Also, the probability that the system transition to state s_{i+1} at or before the next decision epoch t_i is given by $p(s_{i+1} | t_i, s_i, a_i)$. The total elapsed time including the time spent in the current state s_i is $\sigma_i = \sum_{j=0}^{i} t_j$.

The SMDP model also defines cost metrics. The expected total cost incurred between two successive decision epochs (events) is defined in Equation 1 as a sum of the lump sum cost $k(s_i, a_i)$ incured when action a_i is chosen in

2

state s_i, followed by the cost incured at rate $c(s_{i+1}, s_i, a_i)$ as long as the system is in state s_{i+1} after choosing action a_i in state s_i.

$$cost(s_i, a_i) = \quad k(s_i, a_i) + \int_0^\infty E(du|s_i, a_i) \sum_{s_{i+1} \in S}$$
$$\int_0^u e^{-\alpha t} c(s_{i+1}, s_i, a_i) p(s_{i+1}|t_i, s_i, a_i) dt \quad (1)$$

Notice that we chose discounted total cost formulation, where future costs are exponentially scaled down. The discount factor α takes into account the uncertainty of future costs [13].

A *policy* π in SMDP is a discrete, semi-infinite sequence of decisions, one for each time epoch. In this work we consider *Markovian randomized stationary policies*. Such policies can be compactly represented by associating a value $x(s, a) \leq 1$ to each state and action pair in the SMDP. The value $x(s, a)$ is the probability of issuing command a when the system is in state s. It has been shown that the exact and optimal solution to the SMDP policy optimization problem always belong to the set of Markovian randomized stationary policies [13].

To compute the optimal policy we need to minimize the expected total discounted cost. The total expected cost for policy π is defined as the sum of expected costs paid at each decision epoch, discounted to their present value as shown below.

$$v_\pi(s_0) = E_\pi \{ \sum_{i=0}^\infty e^{-\alpha \sigma_i} cost_{i+1}(s_i) \} \quad (2)$$

In the case of the stationary policy, the value of total expected cost for each initial system state is given by:

$$\mathbf{v}_\pi = \mathbf{cost}_\pi + \mathbf{M}_\pi \mathbf{v}_\pi \quad (3)$$

where $\mathbf{M}_\pi$ is $S \times S$ matrix with elements $m(s_{i+1}|s_i, a_i)$ defined by:

$$m(s_{i+1}|s_i, a_i) = \int_0^\infty e^{-\alpha t} p_\pi(s_{i+1}|t_i, s_i, a_i) E_\pi(dt|s_i, a_i) \quad (4)$$

Notice that E_π and p_π for policy π can be easily derived from E and p [13]. The optimal solution the policy optimization problem is the one that minimizes expected total cost:

$$\mathbf{v} = \min_\pi \{ \mathbf{cost}_\pi + \mathbf{M}_\pi \mathbf{v} \} \quad (5)$$

The policy optimization problem can be solved in polynomial time (in $S \cdot A$) by several well-known algorithms, such as value iteration, policy iteration, action elimination or linear programming [13]. The unconstrained version of the problem can be augmented with constraints on different cost metrics without changing its complexity.

4   System Model

We use Semi-Markov decision process to perform optimization of energy consumption under performance constraint (or vice-versa) in two different applications: a hard disk in a laptop computer and a personal communication interactive device that is an extension of the SmartBadge [11], XBadge.
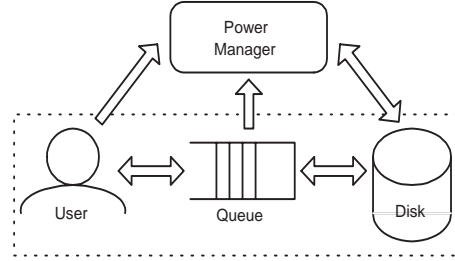

Figure 2. System Model

The system we are modeling consists of the service requestor – SR (user) and service provider – SP (hard disk or XBadge) with queue – Q (the buffer associated with the disk or with the XBadge) as shown in Figure 2. The power manager observes the states of the service provider, the service requestor and the queue at each event occurrence and makes decisions on what state the hard disk should transition to next, in order to minimize energy consumption for a given performance constraint. Each system component is described in detail in the next sections.

4.1   Service Requestor

The service requestor is the user of the device. We analyzed user's behavior in accessing the hard disk using a PC running Windows operating system with standard software. As can be seen from the Figure 3, the request inter-arrival times follow most closely exponential distribution. In the case of the XBadge, we monitored the accesses to the Xserver. Again, we found that we can model the inter-arrival times using exponential distribution with rate of 3.5 requests/s. Thus we can model the service requestor with rate λ_{SR} and the request interarrival time $\frac{1}{\lambda_{SR}}$ where the probability of the hard disk or the XBadge receiving a user request within time interval t follows the cumulative probability distribution shown below.

$$E_{SR}(t) = 1 - e^{-\lambda_{SR} t} \quad (6)$$

Notice that in this case we are assuming that the user request rate is known and stationary. Non-stationary user request rates can be treated using an adaptive policy interpolation procedure similar to the approach presented in [3].
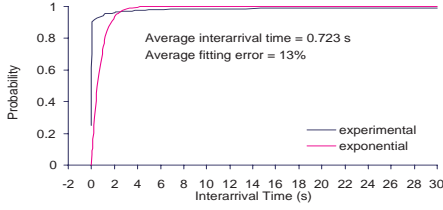
3

Figure 3. Hard Disk Interarrival Time Distribution

4.2 Service Provider

The hard disk or the XBadge are service providers with multiple power states. The XBadge has three power states: active, idle and standby. The hard disk interface supports the ACPI standard [9]. The hard disk we used has three states: active, idle and low-power idle.
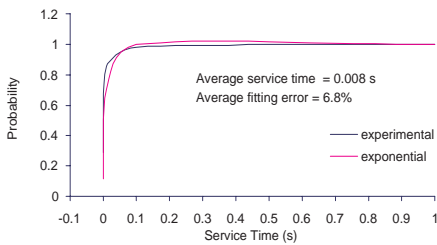

Figure 4. Hard Disk Service Time Distribution

In the active state, the service provider services requests coming from the user. Service times on the PC most closely follow an exponential distribution as shown in the Figure 4. The average service time is defined by $\frac{1}{\lambda_{SP}}$ where λ_{SP} is the average service rate. Service rate of the XBadge accessing the server can also be modeled with the exponential distribution. Equation 7 defines the cumulative probability of the service provider servicing a user request within time interval t.

$$E_{SP}(t) = 1 - e^{-\lambda_{SP}t} \qquad (7)$$

The Travelstar 3GN [8] hard disk we used in our experiments supports two low-power states – idle with a power consumption of $1.8W$ and low-power idle that consumes only $0.65W$. In contrast, the power consumption in the active state is $2.3W$. The hard disk automatically transitions into idle state as soon as it is done reading or writing to the disk. In this state the disk is kept spinning and all the electronics are powered up so that the time required to transition into the active state is small.

Power manager can control the transitions between the active and the low-power idle (LPidle) state. Once in LPidle state, the hard disk waits for the service request arrival before returning back to the active state. The transition between active and LPidle states is best described using uniform distribution, where t_0 and t_1 can be defined as $t_{ave} - \Delta t$ and $t_{ave} + \Delta t$ respectively. The transition from the active state into the LPidle state takes on average $2ms$ with variance of $1ms$. The transition back into the active state is much longer, requiring $40ms$ on average with $5ms$ variance. The cumulative probability function for the uniform distribution is shown below.

$$E_{SP}(t) = \begin{cases} \frac{t-t_0}{t_1-t_0} & t_0 \le t \le t_1 \\ 0 & otherwise \end{cases} \qquad (8)$$

The XBadge also supports two lower power states: idle and standby. The idle state is entered by each component in the system as soon as that particular component is not accessed. That transition is very fast, so the transition time can be neglected as compared to other rates in the system. The entry into the standby state requires a special command sequence for all the components on the XBadge and is thus under the full control of the power manager. Similar to the hard disk, the transition from standby into the active state can be best described using the uniform probability distribution. Components in the XBadge, the power states and the transition times of each component from standby into active state are shown in the Table 1. Note that the XBadge consists of two types of data memory – slower SRAM (1MB, 80ns) from Toshiba and faster DRAM (4MB, 20ns) from Micron that is used only during MPEG decode.

Table 1. XBadge component characteristics

Component	Active Pwr (mW)	Idle Pwr (mW)	Standby Pwr (mW)	t_{ave} (ms)
Display	1000	1000	100	100
RF Link	1500	1000	100	80
SA-1100	400	170	0.1	10
FLASH	75	5	0.023	0.6
SRAM	115	17	0.13	5.0
DRAM	400	10	0.4	4.0
Total	3.5 W	2.2 W	200 mW	150 ms

4.3 Queue

Queue models buffering of the service requests that occurs in accessing the hard disk or the portable communication device such as the XBadge. Since we did not have access to the detailed information about the real-time size of each queue, we measured the queue size with an experiment on a real hard disk using a typical user trace. The

maximum queue size measured is 9 jobs, with average of 0.8.

Measurement of queue size was not possible for the XBadge. But, since its service rate is much higher than the service rate of the hard disk, and the request arrival rate is comparable, we can safely assume that the maximum queue size of 10 jobs is applicable in this case as well.

The number of requests pending in the queue is defined as the queue state. We are assuming that there is no priority associated with the requests and that the queue has final capacity, hence once the limit is reached, a large delay penalty is paid. The queue is filled by the service requestor and emptied by the service provider.

4.4 System States

Each system state can be characterized by the power consumption in the state, the number of jobs in the queue (i.e., performance penalty) and the probability distribution defining the time spent in the state. Table 2 shows system state transitions. Commands are represented with S (go to sleep) and W (wakeup). The exponential transition time probability density function (exp) is a function of the transition rate shown in parenthesis. The uniform transition time pdf (unf) is a function of the average transit time. For example, when a job arrives to the system that is in the active state $D0$ with $q > 0$ jobs waiting in the queue, the next state will be $(D0, q+1)$ and the transition time between the two states depends on the exponential time distribution $\exp(\lambda_{SR})$.

Table 2. System State Transitions

Current State (SP,Q)	Event (Command)	Next State (SP,Q)	Pdf Transit Time
$(D0, q > 0)$	arrival (W)	$(D0, q+1)$	$\exp(\lambda_{SR})$
$(D1, 0)$	arrival (W)	$(D0, 1)$	$\exp(\lambda_{SR})$
$(D0, q > 1)$	done (W)	$(D0, q-1)$	$\exp(\lambda_{SP})$
$(D0, 1)$	done (W)	$(D1, 0)$	$\exp(\lambda_{SP})$
$(D1, 0)$	sleep (S)	$(D3, q_a)$	$\text{unf}(t_{aveS})$
$(D3, 0)$	arrival (W)	$(D3, 1)$	$\exp(\lambda_{SR})$
$(D3, q)$	wakeup (W)	$(D0, q+q_a)$	$\text{unf}(t_{aveW})$

5 Results

The Semi-Markov decision process formulation allows us to derive the optimal policy that minimizes energy consumption while maximizing performance. In order to compare the results of optimization, we developed a simulator for event-driven systems such as the hard disk in a laptop or the XBadge.

Inputs to the simulator are the system description, the expected time horizon, the number of simulations to be performed and the policy. The system description is characterized by the power consumption in each state, the performance penalty, and the function that defines the transition time pdf and the probability of transition to other states given a command from the power manager. The policy is directly obtained from the results of the optimization. For each simulation to be performed, the simulation length is computed using exponential distribution and the expected duration $1/\alpha$. In each state power manager decides what command to give to the service provider based on the policy implemented. In addition, once a command is chosen, the probability of transition into the next state and the expected time until the next event are computed using the system description.
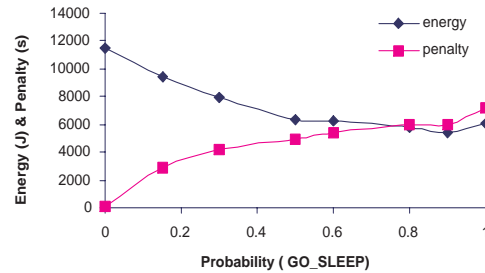

Figure 5. Hard Disk Simulation Results

The results of the simulation are within a few percent of the optimization. The trade-off of performance versus energy for the hard disk is shown in Figure 5 and for the XBadge in Figure 6. Energy (Joules) and performance penalty (seconds) are the totals spent during a 7200 second simulation with the probability of the power manager issuing a sleep command shown on the x-axes. Performance penalty is measured in total seconds that all requests waited in the queue before being serviced by the service provider.

In the case of the XBadge, increasing the probability of issuing a sleep command from 0.15 to 0.30 decreases the energy consumption by 36% while increasing the performance penalty by 27%. The energy consumption is two times larger with an "always-on" policy as compared to the 0.3 probability of issuing sleep command. Similar results can be seen for the hard disk. The energy consumption of the disk is 1.5 times lower with the 0.3 probability of going to sleep as compared to an "always-on" policy. The performance penalty increases 32% from a 0.15 to a 0.3 probability of going to sleep.

We compare the results obtained with our accurate Semi-Markov decision process model to the Continuous-time Markov decision process results [14]. CTMDP assumes
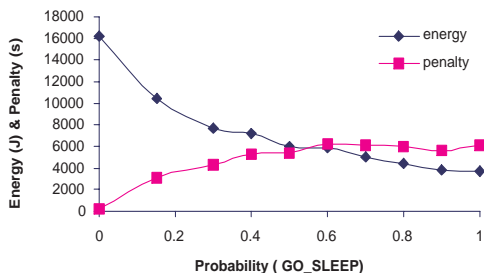
Figure 6. XBadge Simulation Results

that all state transitions are exponentially distributed, including the transitions between the active and LPidle or standby states that are better described by uniform distribution. As a result, CTMDP tends to underestimate the total energy consumption and performance penalty by as much as 25% in the hard disk example and 24% for the XBadge. Table 3 summarizes the results for the different policies implemented on the hard disk and the XBadge. Each policy is represented by the probability of issuing a sleep command while in the idle state, p(S). We show the percent difference in both the total energy and the total performance penalty between SMDP and CTMDP.

Table 3. SMDP and CTMDP Results

Policy p(S)	Hard Disk ΔEnergy	Hard Disk ΔDelay	XBadge ΔEnergy	XBadge ΔDelay
0.0	0%	0%	0%	0%
0.3	1%	1%	4%	4%
0.5	3%	5%	17%	18%
0.6	18%	18%	24%	22%
0.8	14%	14%	7%	5%
1.0	25%	25%	5%	5%

6 Conclusions

The main contribution of this paper is the development of the Semi-Markov decision process (SMDP) model for the formulation of policy optimization in event-driven systems. This model generalizes the previous approaches by allowing event-based policy implementation instead of discrete time. In addition, it allows for arbitrary transition time distributions between the system states instead of limiting the transition time to either a geometric distribution as required by the DTMDP or to an exponential distribution required by the CTMDP.

We analyzed two case studies - the Travelstar hard disk and the XBadge that clearly reflect the trade-off between the performance and the energy consumption. In addition, we developed a simulator for event-driven power management and found that its results were within a few percent of the optimization output. With our approach a system designer can quickly select the optimal policy given constraints in terms of energy and performance.

7 Acknowledgments

References

[1] L. Benini, R. Hodgson, and P. Siegel. System-level power estimation and optimization. In *International Symposium on Low Power Electronics and Design*, pages 173–178, 1998.

[2] L. Benini and G. D. Micheli. *Dynamic Power Management: design techniques and CAD tools*. Kluwer, 1997.

[3] E. Chung, L. Benini, and G. D. Micheli. Dynamic power management for non-stationary service requests. In *Proceedings of DATE*, pages 77–81, 1999.

[4] F. Douglis, P. Krishnan, and B. Bershad. Adaptive disk spin-down policies for mobile computers. In *Second USENIX Symposium on Mobile and Location-Independent Computing*, pages 121–137, 1995.

[5] R. Golding, P. Bosch, and J. Wilkes. Idleness is not sloth. *HP Labs Technical Report*, HPL-96-140, 1996.

[6] D. Helmbold, D. Long, and E. Sherrod. Dynamic disk spin-down technique for mobile computing. In *IEEE Conference on Mobile Computing*, pages 130–142, 1996.

[7] C. H. Hwang and A. Wu. A predictive system shutdown method for energy saving of event-driven computation. In *International Conference on Computer Aided Design*, pages 28–32, 1997.

[8] IBM. *2.5-Inch Travelstar Hard Disk Drive*, 1999.

[9] Intel, Microsoft, and Toshiba. Advanced configuration and power interface specification. *http://www.intel.com/ial/powermgm/specs.html*, 1996.

[10] Y. Lu and G. D. Micheli. Adaptive hard disk power management on personal computers. In *IEEE Great Lakes Symposium on VLSI*, pages 50–53, 1999.

[11] G. Q. Maguire, M. Smith, and H. W. P. Beadle. Smart-badges: a wearable computer and communication system. In *6th International Workshop on Hardware/Software Codesign*, 1998.

[12] G. Paleologo, L. Benini, A. Bogliolo, and G. D. Micheli. Policy optimization for dynamic power management. In *Design Automation Conference*, pages 182–187, 1998.

[13] M. Puterman. *Finite Markov decision processes*. John Wiley and Sons, 1994.

[14] Q. Qiu and M. Pedram. Dynamic power management based on continuous-time markov decision processes. In *Design Automation Conference*, pages 555–561, 1999.

[15] M. Srivastava, A. Chandrakasan, and R. Brodersen. Predictive system shutdown and other architectural techniques for energy efficient programmable computation. *IEEE Transactions on VLSI Systems*, 4(1):42–55, March 1996.