Research Journal of Applied Sciences, Engineering and Technology 5(2): 558-562, 2013
ISSN: 2040-7459; E-ISSN: 2040-7467
© Maxwell Scientific Organization, 2013
Submitted: May 18, 2012 Accepted: June 12, 2012 Published: January 11, 2013

Adaptive Dynamic Power Management Policy: Two Value Alternate Basis

[1,2,3]Yan Wang and [1]Xiangheng Shen
[1]Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130000, China
[2]Graduate University of Chinese Academy of Sciences, Beijing 100049, China
[3]Computing Center of Liaoning University, Shenyang 110036, China

Abstract: In dynamic power management, fixed time-out policy is not optimal because of waiting before shutting down a component and adaptive time-out policy is instability for re-choosing parameters for different workloads. To handle these problems, a new adaptive time-out policy is proposed in this study. Combining the advantage of fixed and adaptive time-out policy zero and spin-down cost are chosen as time-out values in the policy. They alternate as time-out values according history workloads. Zero as one of time-out makes for saving more power. And spin-down cost which only depends on disk model causes policy has more steady performance. Policy based on tow values sufficiently uses the character of cluster and bursty of workloads. Simulation results show that policy proposed in this study is approach to optimal policy and performs more steadily.

Keywords: Dynamic power management, low power, spin-down cost, time out

INTRODUCTION

Serious energy problem of the earth have caused a green revolution in every walk of life. Green computers and low power computer design are paid more attention in computer field. Low power design is essential for computer system, especially for portable electronic device and embedded systems which have limitation on battery life. Dynamic Power Management (DPM) is an important system level low-power design technique. In computer systems, some components have several low power states, such as hard disk, memory and so on. We can shut down them when they do not work in order to reduce power cost. We call them Power Management Component (PMC). DMP is aiming to reduce power consuming by controlling when and how to shut down the PMC.

Traditional policies in DPM are classified into three categories: time-out (Douglis *et al*., 1995; Krishnan *et al*., 1995; Hembold *et al*., 1996; Lu and Micheli, 1999; Irani *et al*., 2003; Wu and Xiong, 2005a), predictive and stochastic policies. Time-out policies select a value as time-out according the history workloads and shut down the PMC when their idle time exceeds the time-out value. The policy is adaptive time-out or fixed time-out according whether change the time-out or not. Time-out policies are simple but not optimal, because they will lose energy when they wait before shutting down the PMC. In contrast with time-out policies, predictive techniques do not wait for a time-out, but shut down the PMC as soon as it becomes idle if the predicted idle time will be long enough to amortize the cost of shutting down. More cost will be saved if the predicted idle time is precise, but the knowledge of user behavior in many cases is initial unknown and non-stationary. Stochastic policies view DMP as stochastic problem and resolve them using stochastic decision model. The policy will gain better effects under the hypothesis of fixed distribution workloads. Furthermore, the cost of these policies is more.

Traditional policies pay attention to idle time; if idle time is long enough we shut down the PMC to save energy. Based on the idea, some new policies which cast about to prolong idle time, are proposed in DMP, such as method based on task scheduling and methods based on data buffer.

Time-out policies are simple and easy to realize, so most computers and portable systems use this kind of policy to control PMC in practice. Wu and Xiong (2005b) have proved that time-out algorithm is the optimal policy for DPM based on stochastic mode in paper 7. Performances of power reduce for fixed time-out policies have been analyzed by experiments and theoretic in 8, 9. Fixed time-out policy cannot gain better effects in frequently fluctuant workloads. To resolve this problem, some adaptive time-out policies are proposed. All of these algorithms cannot reduce energy cost in whole hog because of waiting. To handle this problem, we proposed a new adaptive time-out

Corresponding Author: Cao Yukun, School of Economics and Managemenet, Northeast Forestry University, Harbin 150040, China

policy based on two time-out value in this study. The main contributions of this study can be summarized as follows:

- Handle the vital problem which results in energy losing in time-out policy. Time-out policy will never be optimal, because it will wait for a period time before shutting down some component. Using zero as one time-out value can solve this problem.
- Cluster and bursty character of workloads are sufficiently using in this study. Lots of researches have proved most workloads have the two characters. Based on these, our policy is feasible and obtains better effects.
- Use two fixed value as time-out to fit for many kinds of trace. For this reason, policy in this study has more steadily performance than fixed time-out policy and other adaptive time-out policies which need to re-choose parameters to fit for different workloads. Simulated results in study also show that.

METHODOLOGY

Time-out policies: The complexity of time-out policy is smaller than predictive policy and stochastic policy, so this kind of policy is extensively applied in practice. Fixed time-out policy has a fixed value as time-out and PMC will be shut down while the idle time exceeds the time-out. It is not fit for frequently fluctuant workloads because of the fixed time-out. Furthermore choosing appropriate time-out for different workloads is a difficult problem.

To amend fixed time-out policy, adaptive time-out policy is proposed. Time-out value is adjusted according history workloads. Many methods are used to adjust time-out value, such as adjusting time-out in scale in Douglis' work, adjusting time-out based on machine learning in Helmbold's work, handle DPM as rent-to-buy model in Krishnan's work and dividing requests into sessions in Lu's work. For the works, Lu's method has better effects. Lu compared his method with other adaptive time-out policies, a predictive method and fixed time-out policy. Results show that his work save similar power with other methods except fixed time-out policy, but its number of state changing is less. It means that the method of Lu will prolong disk lifetime. The concept of session will save more power than other adaptive time-out policies without considering lifetime. We will compare our algorithm with Lu's work in later section for its excellent performance.

Algorithm description: The shortage of time-out is the energy loss of waiting before shutting down the PMC. The optimal algorithm is that the PMC will be shut down immediately if idle time is long enough. To handle this problem, we propose an adaptive time-out policy based on two time-out value. Our adaptive time-out algorithm changes the time-out between the spin-down cost and zero.

Now we discuss why we use two time-out values instead of one. Just as we describe in section II, using one value as time-out in DPM is fixed-time out policy. Fixed time-out policy does fit for real workloads. So kinds of adaptive time-out policies are proposed. But you should choose some parameters to fit for different workloads in adaptive time-out policies. This process needs a mass of work. Furthermore it will also induce unstable performance. Choosing parameters is a difficult problem for adaptive time-out policy. So we consider combining the advantage of fixed time-out policy and adaptive time-out policy to produce a new policy which can adjust time-out value according history workloads and does not need choose parameter. Adaptive time-out policy based on two time-out value in this study is proposed based on the thought.

How to choose the two values is vital. We choose zero and spin-down cost as time-out value. Spin-down cost is a time value that the energy cost of keeping the disk spinning equals the energy needed to spin the disk down and then spin it back up. Namely, if the idle time equals to the spin-down cost, shut down the PMC or remain spinning state will bring the same effects. The ideal circumstance is that we shut down the PMC immediately if the idle time is more than spin-down cost and contrarily remain spinning state. Spin-cost is a critical value for DPM, so we use spin-down cost as our first time-out value. Not shutting down PMC immediately causes the main insufficiency of time-out policy. If time-out value is zero, PMC will be shut down immediately. So we choose zero as our second time-out value.

Lots of researches and experiments show that workloads of PMC have the character of cluster, such as workloads shown in Fig. 1 and in study. From these workloads, we see that they have the character of bursty and cluster. So we can conclude idle times of two adjacent requests are small for a period and long for another period. Based on this character, we can use spin-down cost as time-out during the period idle time is small and set time-out zero when the period of longer idle time. For former, the PMC will remain spinning state and for later the PMC will be immediately shut down. Energy cost is optimal in the two circumstances.

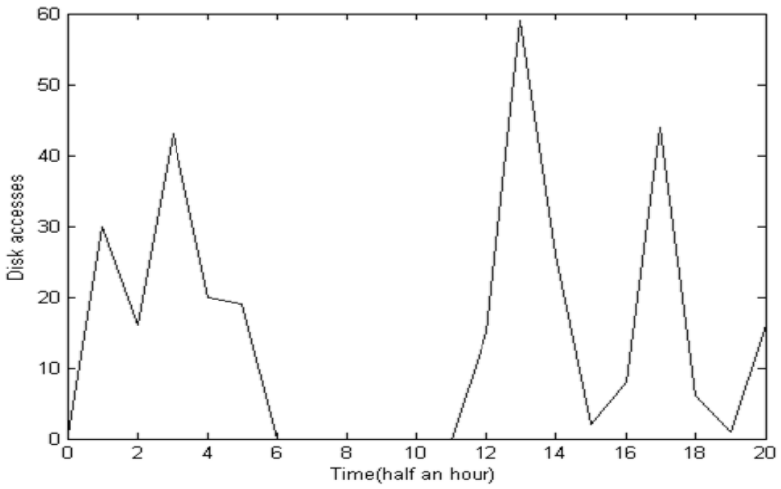

Fig. 1: Disk accesses during 8:30 to 18:30

But our algorithm is not optimal, because the transition between the two circumstances will lose energy. We compute the difference between actual and optimal energy cost after each request access. If energy is lost, we alternate the time-out. The algorithm in this study is sub-optimal and workloads are the more clustered, the effects are better.

First, we use spin-down cost as the time-out and compute the cost of actual and the optimal circumstance. If the former is bigger, energy is lost, alternate the time-out; else remain the time-out. And then repeat the process of energy computing and deciding whether change time-out or not. In the algorithm, we can deduce the trend of next idle time from this idle time based on the cluster character of workloads. Our algorithm is as follows:

Algorithm: (Espin-down, Espin-up, Espinning, Esleeping, Tspin-down, Tspin-up)

```
spin_down =
(Espin-down+Espin-up) /Espinning;
time-out = spin_down;
repeat
if (request is coming)
then
compute T, T is idle time;s
if (T<= time-out)
then
Eactual = Espinning*T;
else
if (T<= Tspin-down+Tspin-up)s
then
Eactual = Espinning*time-out+Espin-down+ Espin-up;
else
Eactual = Espinning*time-out+Espin-down+ Espin-up+Esleeping* (TTspin-down-Tspin-up);
end
end
if (T<= spin_down)
then
Eoptimal = Espinning*T;
else
if (T<= Tspin-down+Tspin-up)
then
Eoptimal = Espin-down+ Espin-up;
else
Eoptimal = Espin-down+Espin-up+sleeping *(T-Tspin-down-Tspin up);
end
end
if (Eactual>Eoptimal)
then
if (time-out == spin_down)
then
time-out = 0;
else
time-out = spin-down;
end
end
end
until no request
```

SIMULATION RESULTS

Hard disk is typical power management component. We apply the proposed policy to hard disk in this study to validate our algorithm. The detailed simulation process is as follows:

Table 1: Disk model

State	Power (W)	Time (S)
Spinning	1.5	-
Sleeping	0.3	-
Spin up	2.5	1
Spin down	1.0	1

Table 2: Simulated policies

Policy	Parameters
Adaptive policy 1	Spin down = 2.33
Adaptive policy 2	th = 40, inc = 1.25
Fixed time-out policy	Time-out = 2.00
Optimal policy	-
No policy	-

We mainly compared the performance of policy in this study and policy of Lu's work. Meanwhile, fixed time-out policy is also simulated. To contrast power saving, we also give the performance of optimal policy which is ideal and circumstance with no policy. Five policies are applied on disk trace in Fig. 1 which is stochastic and disk trace under some distribution. Table 1 shows the disk model in our simulation. In this study we ignore the problem of delay and failure when spin down or spin up disk.

As shown in Table 2, adaptive policy 1 is algorithm in this study and adaptive policy 2 is method of Lu's. According parameters are shown. Spin-down cost only depends on disk model; it is 2.33 sec in our model. Lu's work transforms disk accesses into sessions and its effects are better than other adaptive algorithms. We choose 40 sec as threshold in our simulation. It is not impossible to find out the best fixed time-out policy, so the third policy in Table 2 is relative better for the simulation workloads. We use 2 sec as time-out value. The optimal policy cost minimum amount energy and it is ideal. We use it to evaluate other policies, but cannot put it into practice. No policy is used to contrast how much power is saved. Simulate the five policies in MATLAB to contrast the results.

We use Competitive Ratio (CR), Total Power (TP) and Save Power Radio (SPR) to evaluate the policies. TP is the total power consumption for simulated disk trace. To compute SPR we should give the total power with no policy. We compute CR and SPR according (1) and (2). CR is smaller and SPR is bigger, the effects of policy are better:

$$CR = TPsimulate\_policy/TPoptimal\_policy \quad (1)$$

$$SPR = (TPno\_policy-TPsimulate\_policy)/TPno\text{-}Polic \quad (2)$$

Firstly, policies are applied in disk trace in Fig. 1. Simulated results are in Table 3. No policy is used to compute the value of SPR for simulated policies and the optimal policy major aiding to evaluate other policies.

Table 3: Simulation results for stochastic disk trace

Policy	TP (J)	CR	SPR (%)
Adaptive policy 1	11634	1.0095	78.46
Adaptive policy 2	11737	1.0185	78.26
Fixed time-out policy	12179	1.0568	77.45
Optimal policy	11524	-	78.66
No policy	54000	-	-

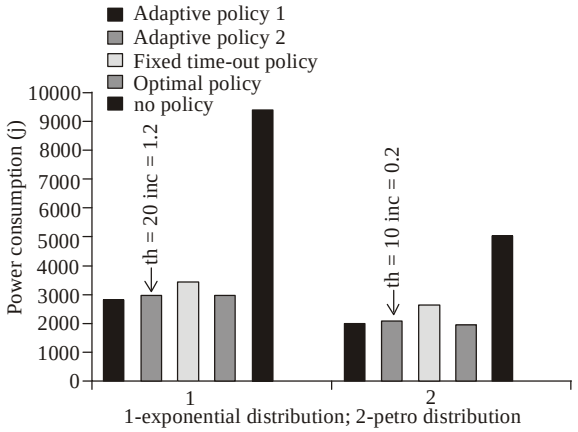

Fig. 2: Total power consumption in two different distribution disk trace

For adaptive policy 2, we choose initial prediction is 40 sec and adjustment parameter is 1.25 sec. If initial prediction is 60 sec, simulated results consume power 13051, but just spin down-up disk 200 times. Results in Table 3 spin down-up disk 305 times. The advantage of adaptive policy 2 is less performance impact and reasonable lifetimes. To contrast the results better, we use the results in Table 3. From Table 3 we can see policy proposed in this study consumes less power. The reason is that our policy avoids energy losing of waiting before shutting down the disk. The policy is simple as traditional time-out policies and further more handle their vital issue.

Results above are about stochastic disk trace on personal computer. The disk trace is from 8:30 to 18:30, just ten hours. Disk traces for long time reflect character of cluster typically. The idle time will be under some distribution if time is long enough. Now we take exponential distribution and Petro distribution as examples to prove our policy can save more power in any circumstance.

The simulated resulted is as Fig. 2. To obtain better effects, we choose different parameters for adaptive policy 2 in different disk traces. The detailed paramet

Table 4: Simulation results for disk trace under distribution

	Exponential distribution		Petro distribution	
Policy	CR	SPR (%)	CR	SPR (%)
Adaptive policy 1	1.0196	70.08	1.0015	62.20
Adaptive policy 2	1.0694	68.62	1.0973	58.59
Fixed time-out policy	1.2250	63.17	1.1389	47.58
Optimal policy	-	70.66	-	62.26
No policy	-	-	-	-

is shown in Fig. 2. Our policy consumed less power and approach to optimal policy in both traces. Table 4 shows the CR and SPR of each policy in different circumstances. For different traces, all policies' CR and SPR are not remain fixedness, especially for adaptive policy 2 and fixed time-out policy. The fluctuation ascribes to choosing parameters in different traces. There are no universal parameters to fit any kind of trace. It will impact performance of DMP policy. Policy in this study does not confront this problem, because spin-down cost only lies on disk model.

From results above, we show that:

- Policy based on two values in this study has better performance on power consumption.
- It also has steady performance whenever for stochastic disk trace or for disk trace under some distributions.

CONCLUSION

Aiming at the shortage of traditional time-out policy, including fixed time-out policy and adaptive time-out policy, an adaptive time-out policy is proposed in this study. The policy is based on spin-down cost and zero which are alternative time-out value. The characters of cluster and bursty assure policy in this study feasible and effective. Zero as time-out can avoid energy losing when waiting before shutting down the PMC in time-out policies. Spin-down cost just depends on disk model and it makes policy with steady performance. We give detail algorithm of the policy and simulate ten hours disk trace and disk traces under distributions with MATLAB on five policies. Simulation results show that policy in this study consumes less power and performs more steadily. It approaches to optimal policy. The policy based on two value alternate hold the advantage of traditional time-out policies: simple and easy to realize and combine the advantage of fixed time-out policy and adaptive time-out policy to settle their vital issue. We do not consider the problem of delay and failure problem when spin down or spin up PMC in this study; we will handle these problems in future work.

REFERENCES

Douglis, F., P. Krishnan and B. Bershad, 1995. Adaptive disk spin-down policies for mobile computers. Comput. Syst., 8: 381-413.

Hembold, D., D. Long and B. Sherrod, 1996. A dynamic disk spin-down technique for mobile computing. Proceeding of the 2nd International Conference of Mobile Computing and Networking, pp: 130-142.

Irani, S., S. Shukla and R. Gupta, 2003. Online strategies for dynamic power management in systems with multiple power-saving sates. ACM Trans. Embedded Comput. Syst., 2: 325-346.

Krishnan, P., P. Long and J. Vitter, 1995. Adaptive disk spindown via optimal rent-to-buy in probabilistic environments. Proceeding of 12th International Conference on Machine Learning, pp: 322-330.

Lu, Y. and G. Micheli, 1999. Adaptive hard disk power management on personal computers. Proceeding of 9th Great Lakes Sym. Very Large Scale Integration, pp: 50-53.

Wu, Q. and G.Z. Xiong, 2005a. Adaptive dynamic power management for non-stationary self-similar requests. J. Software, 16: 1499-1505.

Wu, Q. and G.Z. Xiong, 2005b. Why simple timeout strategies work perfectly in practice. Lect. Notes Comput. Sci., 3605: 468-473.