A Research on an Optimized Adaptive Dynamic Power Management

Jie CHEN, Deyuan GAO, Qiaoshi ZHENG
Department of Computer Science and Engineering
Northwestern Polytechnical University
Xi'an, 710072, China
{jerrychen, gaody, qiaoshi2008}@mail.nwpu.edu.cn

Abstract—**Low-power design for various applications has always been a challenge for system designers. Dynamic power management, by selectively shutting down idle components, is widely studied and considered to be effective in reducing power consumption. Management strategies based on online algorithm exhibit a feature of easy implementation and fast processing speed. However, merely based on the historical distribution of idle periods, these strategies will make inaccurate prediction if the real distribution of idle periods changes sharply. This paper presents an optimized adaptive dynamic power management for further power saving. We introduce a differential adjusting factor to optimize the exponential-average algorithm to rapidly and accurately adjust the predicted idle period to the real distribution. Experimenting results demonstrate that our policy of power management can reduce the power dissipation of processors in a larger scale and be utilized in diverse applications.**

Keyword-dynamic power management; adaptive; exponential-average algorithm; adjusting factor

I. INTRODUCTION

In the design of VLSI circuits, especially for microprocessor, power dissipation becomes an indispensable issue. Since certain hardware devices, within a VLSI system, do not always reside in the working status, it's applicable to minimize the power consumption by detecting idle devices and cutting off their power supply. Common methods for that can be described as: if the current workload doesn't need a specific hardware device, the system shuts down the device and enters it into a low-power stand-by status; otherwise, when the hardware device is needed by the upcoming workload, it is woke up and then recovered into the working status.

The effectiveness of dynamic power management (DPM) has been verified by many past works. Most DPM techniques, at the runtime of the system, utilize the previous workload records to dynamically predict the possible idleness and then selectively shut down hardware components to regulate the overall system power consumption and performance. Currently, hardware-based DPM techniques are generally applied in I/O intensive devices. Alternatively, from an OS perspective, using the software-oriented techniques for power management is mainly applied in CPU intensive devices. By exploiting the OS records on task-level information, power management software is used to adjust CPU clock frequency or to turn part of the hardware into low-power status. For communication intensive devices focusing on energy conservation with low system degradation, new DPM techniques were also researched in the past to perform the trade-off between energy conservation and data delivery quality [1].

Commonly, power management policies are divided into three classes: time-out [2, 3, 12], predictive [4, 5, 6], and stochastic policies [7, 8]. This paper is mainly focusing on a hardware perspective to study the prediction strategy of DPM. The accuracy of a prediction strategy, based on the old or previous workloads to estimate the next idle period, decides the effective of a DPM. Prediction strategy can either adopt an offline or an online algorithm. An offline algorithm is, based on offline analysis of usage traces, not suitable for nonstationary request streams whose statistical properties are not known a priori [8]. Online algorithms are easy to be implemented and can quickly process the input data. However, merely based on the historical distribution of idle periods, online algorithms will make inaccurate prediction if actual idle periods are distributed in a sharply changing way. This paper presents an optimized adaptive dynamic power management for further power saving. We introduce a differential adjusting factor to optimize the exponential-average algorithm to rapidly and accurately adjust the predicted idle period to the real distribution.

II. RELATED WORK

Past researches put a magnitude of emphasis on the foregoing mentioned three power management policies. Time-out policy shuts down an idle device after a timeout value, which divides into two mechanisms: fixed time-out policy shutting down the device after a fixed length of idle time, and the adaptive time-out policy adjusting the time-out referring the previous history. Compared with time-out policies, predictive techniques predict the amount of idle periods and shut down the system if the predicted length of idle period surpasses the overhead of entering into and recovering from a low-power mode [6]. Stochastic policy model the arrival of requests and device power-state changes as stochastic processes, such as Markov processes [10]. This method can be applied to general systems and user requests [8].

Srivastava et al. proposed a predictive power management strategy. They tried to predict the length of idle time based on the computation history, and then shut the

978-1-4244-4520-2/09/$25.00 ©2009 IEEE

processor down if the predicted length of idle time justified the cost-in terms of both power and responsiveness-of shutting down [5]. Chi-Hong Hwang et al. presented a new predictive system shut down method for event-driven applications. They adapted the exponential-average approach used in the CPU scheduling problem for the prediction of the idle period and this strategy has a better performance [4]. Eui-Yong Chung et al. presented a novel adaptive predictive method applicable to systems with an arbitrary number of sleep states. This policy is based on a new dynamic data structure called an adaptive learning tree. Using the tree, the most appropriate low-power sleep state can be predicted accurately at the start of an idle period [6]. G. A. Paleologo et al. provided a stochastic model which used in power-managed system. They proposed that the fundamental problem of finding an optimal policy which maximizes the average performance level of a system, subject to a constraint on the power consumption, can be formulated as a stochastic optimization problem. And a polynomial-time can solve this problem. But this approach based on an assumption that the Markov model of the workload is known and stationary [7]. Then Eui-Yong Chung et al. proposed an online adaptive DPM scheme which overcomes the drawback in [7]. This online adaption is required to deal with initially unknown or nonstationary workloads, which are very common in real-life systems [8]. Qinru Qiu et al. introduced a continuous-time, controllable Markov process model of a power-managed system. The system model is composed of the corresponding stochastic models of the service queue and the service provider [9].

III. ADAPTIVE DPM STRATEGY

A. Basic Theory

In this paper, we present an adaptive DPM strategy based on the exponential-average algorithm [4]. The algorithm applies the last time predicted idle period and the actual one as weighting factors. The weighting for each older data point decreases exponentially, giving much more importance to recent observations while still not discarding older observations entirely. The formula of the algorithm is demonstrated in (1).

$$T_{p(n+1)} = b \cdot T_n + (1-b) \cdot T_{p(n)} \tag{1}$$

T_n is the last actual idle period; $T_{p(n)}$ is the last predicted idle value; b 和($1-b$) is the weighting parameter for T_n and $T_{p(n)}$, respectively; and $T_{p(n+1)}$ is the current predicted idle value. Then, (1) can be expanded as (2).

$$T_{p(n+1)} = b \cdot T_n + b \cdot (1-b) \cdot T_{n-1} + \cdots + b \cdot (1-b)^n \cdot T_0 + b \cdot (1-b)^{n+1} \cdot T_{p(0)} \tag{2}$$

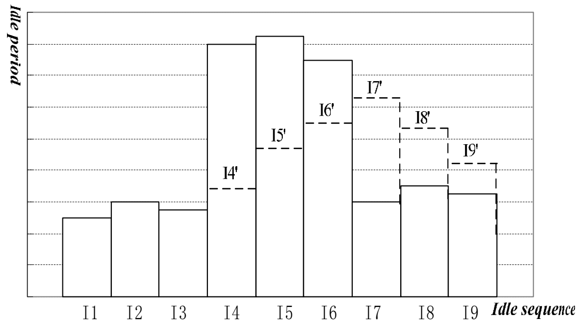

Figure 1. Exponential-average algorithm makes undesirable prediction of idle period if distribution of idle periods change sharply.

Thus, the degree of weighing decrease is expressed as a constant smoothing factor b , a number between 0 and 1. The current predicted idle period is an average of all the old idle periods with exponentially decreasing weighting. The more recent idle periods exert more influence on the current prediction value. However, merely based on the historical distribution of idle periods, the algorithm will make undesirable prediction if actual idle periods are distributed in a sharply changing way as shown in Fig.1 [11].

From the perspective of cybernetics, we introduce a differential adjusting factor a in (3) to optimize the exponential-average algorithm to rapidly accommodate the predicted values to the regularly sharply changed distribution of idle periods shown in (4).

$$T_{p(n+1)} = a \cdot (b \cdot T_n + (1-b) \cdot T_{p(n)}) \tag{3}$$

$$a = \begin{cases} 1, if \dfrac{T_n}{T_{n-1}} \in [U_{\min}, U_{\max}] \\ \dfrac{T_n}{T_{n-1}}, if \dfrac{T_n}{T_{n-1}} \notin [U_{\min}, U_{\max}] \end{cases}, \tag{4}$$

where $U_{\min} < 1 < U_{\max}$.

If the last two actual idle periods T_n and T_{n-1} have a ratio within the threshold interval between low bound $U_{\min}$ and upper bound $U_{\max}$, we consider they have smooth and mild change then we can keep using exponential-average. In this case, $a=1$ and (1) equals (3). Otherwise, if the ratio fails to locate within threshold interval between low bound $U_{\min}$ and upper bound $U_{\max}$ there is a sudden and large change between the past two idle periods. If we still use the exponential-average to predict next idle period, the prediction error would beyond our expectation. However, in such a case, we set the adjusting factor $a = \dfrac{\mathrm{T}_n}{T_{n-1}}$, which reflects the spanning ratio of idleness change, and add the ratio as an influential factor to the prediction formula. Notably, the parameter b , $U_{\min}$ and $U_{\max}$ are decided by specific application environment, which will be demonstrated in our experiments.

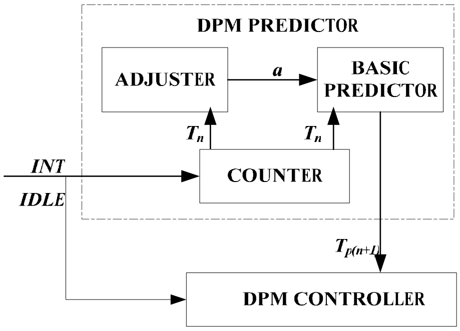

Figure 2. Adaptive DPM model.

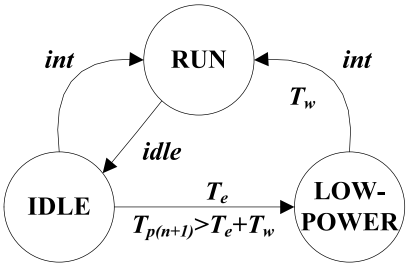

Figure 3. Finite state machine of DPM controller.

B. Design Models

Our adaptive DPM model consists of two parts: DPM Predictor and DPM Controller. We introduce a DPM Predictor to implement the proposed adaptive prediction strategy which can be divided into three modules: Counter, Basic Predictor and Adjuster. Counter is deployed to record T_n, which stands for the length of last actual idle period. The Counter starts counting immediately after the processor enters into the idle status and stops counting when detecting a processor interruption. Then, the value of T_n is sent simultaneously to the Basic Predictor and the Adjuster. When Adjuster receives the new value of T_n, it calculates the adjusting factor *a* according to (4). Since the calculation of *a* need two previous idle time value, the Adjuster conserves data using a simple FIFO with depth of 2. When the new value comes in, the old one is discarded. After the calculation of adjusting factor *a* is ready, Basic Predictor uses *a* to calculate $T_{p(n+1)}$ in accordance with the (3).

Fig.3 shows the finite state machine of DPM Controller which embraces three states: RUN, IDLE and LOW-POWER. Initially, when the system is in the RUN state, the processor executes tasks at the normal clock rate, leaving DPM suspended. Once the processor executes the idle instruction, it stops running tasks, though keeping the clock rate as normal. At this time, the system enters into the IDLE state. Here we take T_e as the representation of the timing cost of entering Low-power state and T_w of withdrawing from the LOW-POWER state. During the IDLE state, DPM Controller reads the predicted idle period from Predictor and determines whether it compensates the overall timing cost of entering and withdrawing from LOW-POWER state as in (5).

$$T_{p(n+1)} > T_e + T_w \tag{5}$$

If $T_{p(n+1)}$ compensates the cost, the system enters LOW-POWER state. Otherwise, the system stays in IDLE state until an interruption is detected by the processor and then turns into RUN state. In the LOW-POWER state, the clock rate of processor is lowered or even stopped for reducing power dissipation; when an interruption is detected, the system goes back to RUN state.

IV. Performance Evaluation

Our experiment was carried out on the control-oriented enhanced “Longtium” DSP core, which is designed by Aviation Microelectronic Center of Northwestern Polytechnical University. The software core of the DSP processor has been verified on FPGA and has already been applied in a third-party MP3 audio processing SoC. We use the TSMC 0.25μm CMOS process to develop the DSP core which achieves clock rate of 150 MHz and integrated level of 0.32 million.

We implemented the adaptive DPM strategy on the “Longtium” DSP core and conducted the simulation on FPGA. The implementation was based on Xilinx Virtex-II FPGA, while we utilized the Xilinx ISE 4.1i accessory XPower to conduct the power analysis. Power consumption consists of dynamic power, State power, output power, wire power and etc. Table.1 shows the relationship between power consumption and clock rate which indicates that during a DPM process if we lower the clock rate of a processor only dynamic power is affected. Therefore we only make statistics of dynamic power.

We adopted as our application programmes the ADPCM, Signal Energy Spectrum Analysis (SEPA) and Two-Dimensional Convolution on Image (TDCI) for the simulation. As for the exponential-average algorithm, each time we adopt all the values in the set of（0.1, 0.3, 0.5, 0.7, 0.9）for the weighting parameter *b*. As for the optimized adaptive exponential-average algorithm, we set the threshold interval ($U_{\min}, U_{\max}$) as (1/2, 2) and (1/3, 3), respectively. Fig.4 and Fig.5 show the simulation results.

TABLE I. The Relationship Between Power Consumption and Clock Rate in “Longtium” DSP Core

Clock rate (MHz)	100	50	25	12.5	0
Dynamic Power (mW)	569	291	152	82	13
Other Power (mW)	231	231	231	232	231
Power Sum (mW)	800	522	383	314	244

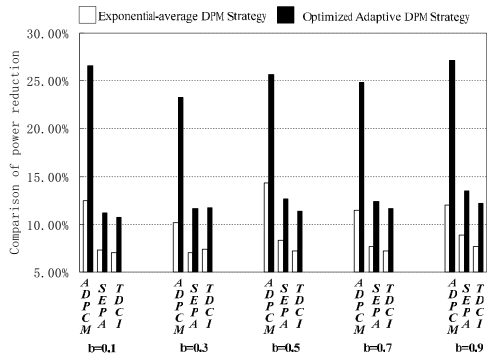

Figure 4. Comparison of power reduction.（U_{min} =1/2, U_{max} = 2）

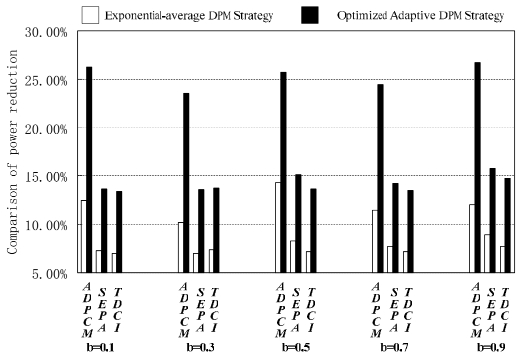

Figure 5. Comparison of power reduction.（U_{min} =1/3, U_{max} = 3）

Fig.4 demonstrates that, when the threshold interval (U_{min}, U_{max}) is set as (1/2, 2), an exponential-average DPM strategy can reduce the dynamic power by 7.0%~14.3%, while an optimized adaptive DPM strategy can reduce the dynamic power by 10.7%～27.1%.

At the same time, Fig.5 illustrates that, when the threshold interval (U_{min} , U_{max}) is set as (1/3, 3), an exponential-average DPM strategy can reduce the dynamic power by 7.0%~12.5%, while an optimized adaptive DPM strategy can reduce the dynamic power by 13.4%～26.7%. Therefore, our proposed adaptive DPM strategy can reduce power dissipation in a larger scale.

Furthermore, under the adaptive DPM strategy, when (U_{min}, U_{max}) changes from （1/2,2）to （1/3,3）, dynamic power consumption of ADPCM programme changes little, while programmes of signal energy spectrum analysis and two-dimensional convolution on images have much lower power consumption. Conspicuously, by regulating various threshold intervals for different types of applications, our proposed adaptive DMP strategy can work well for bringing down the power consumption of their running.

V. CONCLUSION

In this paper, we've presented an optimized adaptive DPM strategy based on online exponential-average algorithm. Utilizing the differential adjusting factor we've optimized the exponential-average algorithm and made the online algorithm rapidly accommodate to regularly sharply changed distribution of idle periods. Our experimenting results have shown that, compared to the basic exponential-average algorithm, our strategy is more flexible and capable to reduce power dissipation of the DSP processor with various applications.

ACKNOWLEDGMENT

We would like to thank the referees of this paper for their extremely useful suggestions. This paper is supported by National Natural Science Foundation of China under Grant (No. 60736012).

REFERENCES

[1] Rodrigo M.Passos, Jose Augusto Nacif, Raquel A.F.MiNi et al. "System-level Dynamic Power Management Techniques for Communication Intensive Devices,"Very Large Scale Integration,2006 IFIP Internation Conference,16-18 Oct.2006,pp:373-378.

[2] Richard A. Golding, Peter Bosch II, Carl Staelin, Tim Sullivan, and John Wilkes, "Idleness is not sloth," in *USENIX Winter*, 1995, pp. 201–212.

[3] F. Douglis et al., "Adaptive Disk Spin-Down Policies for Mobile Computers," *Computing System*s,vol. 8, no. 4, 1995, pp. 381-413.

[4] Chi-Hong Hwang and Allen C.H. Wu, "A Predictive System Shutdown Method for Energy Saving of Event-Driven Computation", *International Conference on Computer-Aided Design*, pp. 28-32, November 1997.

[5] M.Srivastava, A. Chandrakasan, R. Brodersen, "Predictive system shutdown and otherarchitectural techniques for energy efficient programmable computation", IEEE Transactions on VLSI Systems, Vol.4,No.1(1996),pages 42-55.

[6] Eui-Young, Chung Luca Benini, Giovanni De Micheli, "Dynamic Power Management Using Adaptive Learning Tree", *International Conference on Computer-Aided Design*, November 1999.

[7] G. A. Paleologo et al., "Policy Optimization for Dynamic Power Management," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,vol. 16, no. 6, June 1999, pp. 813-833.

[8] E.Y. Chung, L. Benini, A. Bogliolo and G. De Micheli, "Dynamic Power Management for non-stationary service requests" DATE - Proceedings of the Design Automation and Test in Europe Conference and Exibition, 1999, pp. 77-81.

[9] Q. Qiu and M. Pedram, "Dynamic power management based on continuous-time Markov decision processes", Design Automation Conference, pp. 555-561, June 1999.

[10] Yung-Hsiang Lu, "Comparing System-Level Power Management Policies", IEEE Design&Test of Computers, March-April 2001, pp. 10-19.

[11] L. Benini, A. Boglioloz, G. De Micheliy, "A Survey of Design Techniques for System-level Dynamic Power Management", IEEE Transactions on VLSI Systems, March 2000.

[12] D. Ramanathan and R. Gupta, "System Level On-Line Power Management Algorithms," Proc.Design Automation and Test in Europe, IEEE CS Press, Los Alamitos, Calif., 2000, pp. 606-611.