A Time Reservation Dynamic Algorithm Based on the Variable Frequency Interval DVFS Technology

Jun Xiao
School of Information Science and Engineering, Hunan University
Lushan South Rood, Changsha
410082, P.R. China
Email: darmac@163.com

Cheng Xu
School of Information Science and Engineering, Hunan University
Lushan South Rood, Changsha
410082, P.R. China
Email: cheng_xu@yeah.net

Lining Zeng
School of Information Science and Engineering, Hunan University
Lushan South Rood, Changsha
410082, P.R. China
Email: will120120@126.com

Abstract—**embedded systems have been used in various areas of people's lives. However, the contradictions between performance and power consumption of embedded products have become increasingly prominent. DVFS technologies provide the possibility for solving the contradiction between performance and power consumption of equipment. In this paper, a DVFS algorithm based on variable interval DVFS technology has been improved. The improved algorithm reduces response delay by reserving part of execution time of the task.**

***Index Terms* — DVFS， variable interval， energy optimization**

I. INTRODUCTION

Nowadays, embedded systems have been integrated into all areas of people's lives. Smart phones can provide good communications services for people; Automatic navigation system can provide intelligent navigation and traffic alerts service for drivers; Medical monitoring system serving for the patient can provide long-term 24-hour monitoring service. Once the patient is abnormal, it will immediately notify doctors and their families to handle unexpected situations quickly; smart home system can not only intelligently control the electronic device in the user's room, but also can remotely receive the user's operating instructions, so that users can feel at ease to outdoor travel. Embedded system's goal is to provide users with a permanent, stable and reliable service. However, in practice, the data processing capacity and power consumption of electronic devices is limited each other, high performance usually means to pay a correspondingly higher energy costs. Therefore, how to ease the contradiction between power consumption and performance has become one of the most popular research areas.

DVFS is one of the most popular technologies of CPU power management, which is able to adjust the voltage and frequency of the CPU. Since power consumption and voltage of the device is directly related, this technique provides an opportunity to adjust system energy consumption. DVFS algorithm's goal is to meet the requirements of system performance, minimize the CPU frequency and voltage, so as to achieve energy savings.

The energy consumption has become a bottleneck for system design in modern embedded systems. CPU power consumption is more than 50% of the energy consumption of the whole system, Good energy optimization can reduce the requirements of the whole system for heat dissipation, so as to reduce energy consumption for cooling CPU, simplify complexity of temperature management systems, and reduce hardware cost, it can also greatly enhance the endurance for portable devices. Thus, science optimization of CPU power will have a significant meaning to the system.

II. RELATED WORKS

DVFS technology has experienced a rapid growth in these years. An online learning method has been used to evaluate the frequency bound in the tasks [1]. Reference [2] and Reference [3] proposed some energy optimization algorithm based on memory-aware technology, and they have considered about the memory access overhead in the system. There is also a mechanism that has combined DVFS technology and DPM technology in [4], so the algorithm in this work can make a better energy optimization. Reference [5] and Reference [6] proposed some methods which can both manage the CPU voltage and memory voltage. these methods adjust system voltage and frequency by the mixed status of CPU and memory, and make a more appropriate frequency adjustment strategy. Reference [7] and Reference [8] used dynamic slack reclamation strategy for different systems. Reference [9] has pointed that about 83% researches in DVFS technology have not considered about the DVFS overhead in recent years.

Based on the research work [7] to [11], our previous work [12] improved the algorithm based on variable interval DVFS technology. We have improved the theoretical models in DVFS research and the corresponding algorithm. In this paper, the work is still based on these improvements. In this section, we briefly reviewed the work we have done before.

A. A Multi-phase Task Model

Reference [12] uses a multi-phase task model to show frequency bound characteristics of the task. The researchers can determine the grain size of each task phase according to their own requirements, for a finer grain of task phase, it can reflect the more detail features of the task. According to this task model, we have designed a static optimization algorithm. A best frequency adjustment strategy can be obtained by using this algorithm. So , we can calculate the minimum energy a task costs.

The task model is as follows:

$$Task = \{n, d, m_i, t_i\} \quad (1)$$

Where n is the phase number of the task, d is the deadline of the task, m_i is the memory bound of phase i, t_i is the completion time of phase i if we run the task just at the frequency of its memory bound.

B. DVFS overhead Module

In addition to the above model, we also made some improvements on the energy consumption model which based on the traditional energy consumption model. We have added DVFS overhead of the energy consumption to enhance the accuracy of the estimation of energy consumption. This improvement of energy consumption model also introduces a new factor for evaluating the performance of the DVFS algorithm.

The simple DVFS energy overhead model we proposed is as follows:

$$E_{overhead}\begin{cases}\int_{V_{start}}^{V_{end}} 17dv,\ if(V_{end} \geq V_{start}) \\ \int_{V_{start}}^{V_{end}} -14.6dv,\ if(V_{end} < V_{start})\end{cases} \tag{2}$$

Where $E_{overhead}$ is energy overhead, V_{start} is the starting voltage, V_{end} is the ending voltage. The units of $E_{overhead}$, V_{start} and V_{end} are uJ, V, V.

C. DVI Algorithm

Based on the improvements of theoretical model, we improved the granularity of the variable interval in DVFS system, by using a coarse granularity interval adjustment granularity instead of original fine-grained interval adjustment granularity. Then, Reference [12] designed a DVI algorithm by switching the interval of frequency adjustment to change the system frequency adjusting speed. Fast frequency interval can enhance the adaptability of CPU on task frequency bound mutation, but it will generate higher energy costs. A normal interval has higher energy efficiency, but it is sometimes difficult to track the changes in the frequency of system requirements. DVI algorithms can obtain better power consumption and performance by adjusting the frequency and the frequency adjustment interval of the system. However, DVI algorithms cannot handle the timeout problem properly. So, in sometimes, it is difficult to satisfy the requirements of real-time of tasks.

III. FREQUENCY CONTROL STRATEGY BASED ON VARIABLE FREQUENCY INTERVAL

In this paper, a TRD (Time Reservation DVI) algorithm is used to reduce or avoid task timeouts problem by reserving part of the execution time of the task.

A. Strategy Goals

The main objection of TRD algorithm is to minimize the energy consumption required for the CPU while there is not obvious delay in the individual task processing. When the task set arrives, TRD algorithm can obtain key information which includes assignments and task deadlines. During the execution of the task, our algorithm can periodically get current stage frequency limit value of the task through the interface provided by system. Then, in order to optimize the energy consumption as much as possible, TRD algorithm will submit the voltage and frequency adjustment strategy to the system according to the estimated frequency bound value and current frequency value.

B. Proposed Frequency Control Strategy

In order to relieve the timeout problem, the TRD algorithm reserved a period of execution time for the task. The proportion value of reserved time is the most important parameters of the algorithm, a too long reserved time will reduce energy consumption optimization performance of the algorithm, and on the contrary, it is difficult to solve the timeout problem of task.

TRD algorithm processing procedure is as follows:

1. When a new task arrives, the algorithm will recalculate timeout and task deadline. If the task has timed out, it will compensate timeout by using the slack time of current task to reduce the cumulative timeout. Then, the algorithm uses a strategy of deception, which deducts 5% of the available time of the task as the reserved time. When the task execution time reaches 95% of the original time, the algorithm considers the task has timed out, and will use a time-out policy for task processing to speed up the completion rate of the task.

2. If a task is completed, the algorithm will switch the frequency update interval mode to the normal mode, and reduce the system frequency. If the task is still in the running state, the algorithm first obtain system voltage, frequency, progress of current task stage and the frequency bound, and determine whether the task timeout:

- If the task has timed out, frequency update interval mode can be set to the quick mode, and our algorithm will maximize the system frequency within the allowed range of frequency bound as soon as possible.
- If the task has not timed out, our algorithm will calculate the mean frequency according to the remaining tasks and time, and use the smaller value between mean frequency and frequency bound to adjust the frequency.

3. TRD will switch system to the fast frequency update mode in these circumstances:

- System frequency is higher than the frequency bound. This state means some unnecessary energy consumption exists in the system.
- The difference between the current system frequency and the target frequency is too large. In this situation, if the algorithm adjusts the frequency with a normal interval, the system may be difficult to achieve the frequency which current task needs in time.
- The difference between current task progress and expected task progress is too large. When a task arrived, TRD algorithm will calculate an expected frequency value that the task need, and it can calculate the expected task progress whenever it needs. If the difference between current task

progress and expected task progress is too large. The task may run out of the time. So, we need adjust the frequency interval to the fast mode and track the frequency bound in the task to minimize the possibility of task timeout.

The description of the improved DVI algorithm is as follows:

```
Algorithm 1: TRD algorithm
Input:
Output: FreqAdj_Dir, FreqAdj_Int
1. If task set is not finished
       Update task_quota , aval_time , task_proc, freq_bound;
       aval_time = aval_time * 95%;
   else
       go to step 7;
2. If new task arrives
       Update task_quota;
       Reset the variable which indicates the degree of task
completion: task_proc <- 0;
       Caculate the available time in this task: aval_time <-
deadline - timeout;
       Estimate the expected average frequency: exp_freq <-
task_quota/aval_time;
3. If a task is finished and the system is waiting a new task
       stg_dir <- dir_down;
4. If current task is not finished
       If task timeouts
           aim_freq<- freq_bound;
       else
           Caculate average frequency: avg_freq <-
task_quota/aval_time;
           aim_freq<- minimum{avg_freq,freq_bound};
           If now_freq > aim_freq
               stg_dir <- dir_down;
           else if now_freq > aim_freq
               stg_dir <- dir_up;
           else
               stg_dir <- dir_none;
               stg_int <- interval_normal;
           If now_freq – aim_freq> threshold_freq
               stg_int <- interval_fast;
           If now_freq > freq_bound
               stg_int <- interval_fast;
           If task_proc > exp_proc
               stg_int <- interval_fast;
           If task timeouts
           stg_int <- interval_fast;
5. Submit the strategy:
           FreqAdj_Dir <- stg_dir;
           FreqAdj_Int <- stg_int;
6. Goto 1;
7. End
```

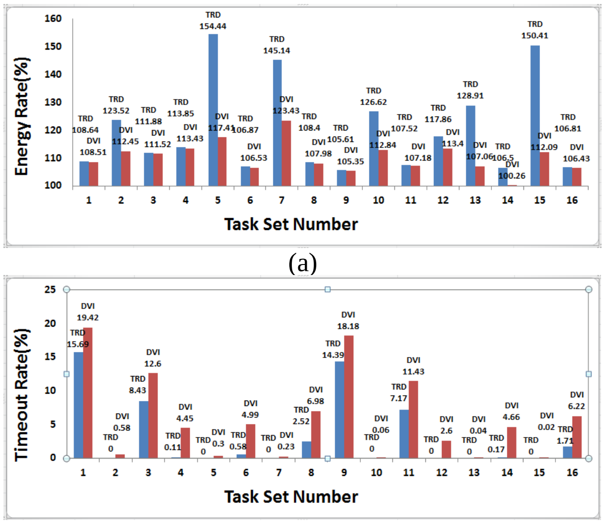
IV. Experiment and Results

In order to test the actual performance of TPD algorithm, we use the same experimental platform and set of tasks as the one in the DVI algorithm in [12], so it's easily to compare the ratio of energy consumption and task timeout with both algorithms. The task sets are as follows:

TABLE I. TASK TYPES

Task Types			
Task set Number	***Phases mark***	***Quota (per phase)***	***Slack time (%)***
1	HL	150,000	16.67
2	HLH	150,000	16.67
3	HLHL	150,000	16.67
4	HLHLH	150,000	16.67
5	LH	150,000	16.67
6	LHL	150,000	16.67
7	LHLH	150,000	16.67
8	LHLHL	150,000	16.67
9	HL	300,000	16.67
10	HLH	300,000	16.67
11	HLHL	300,000	16.67
12	HLHLH	300,000	16.67
13	LH	300,000	16.67
14	LHL	300,000	16.67
15	LHLH	300,000	16.67
16	LHLHL	300,000	16.67

By executing these tasks set in table 1 in our experimental platform, the result is as follows:



(a)

(b)

Figure 1. Comparison result between TRD algorithms and DVI algorithm: (a)energy rate (b)timeout rate

The experiment shows that TRD algorithm can achieve a better response time than DVI algorithm in task processing. With the tasks which DVI algorithm cannot complete on time, TRD algorithms can reduce the response delay of tasks significantly while maintaining close to DVI algorithms energy consumption; however, with the tasks which DVI algorithm can handle correctly, TRD algorithm usually generates much more energy consumption, but only get a slightly lower response time.

V. CONCLUSION

In this paper, a TRD algorithm which improved from the DVI algorithm has been proposed to alleviate the timeout problems. The experiment shows that the timeout ratio has been reduced by using TRD algorithm. This algorithm can effectively improve the response time of the task, but it also may generate too much energy consumption in some tasks. Therefore, we will continue to improve TRD algorithm in the after work.

REFERENCES

[1] G. Dhiman, T. S. Rosing; "Dynamic voltage frequency scaling for multi-tasking systems using online learning," Low Power Electronics and Design (ISLPED). Portland, pp. 207-212, 2007.

[2] Xi Chen, Chi Xu, R. P. Dick; "Memory access aware on-line voltage control for performance and energy optimization," Computer-Aided Design (ICCAD). San Jose, pp. 365-372, 2010.

[3] Wen-Yew Liang, Shih-Chang Chen, Yang-Lang Chang, Jyh-Perng Fang; "Memory-Aware Dynamic Voltage and Frequency Prediction for Portable Devices," Embedded and Real-Time Computing Systems and Applications. Kaohsiung, pp. 229-236, 2008.

[4] Jinuk Choi, Hojung Cha; "A Processor Power Management Scheme for Handheld Systems Considering Off-Chip Contributions," IEEE Transl. J. Industrial Informatics. vol. 6, pp. 255-264, 2010.

[5] Ercan Furkan, Gazala Neven Abou, David Howard; "An integrated approach to system-level CPU and memory energy efficiency on computing systems," Energy Aware Computing. Guzelyurt, pp. 1-6, 2012.

[6] Wen-Yew Liang, Yen-Lin Chen, Ming-Feng Chang; "A memory-aware energy saving algorithm with performance consideration for battery-enabled embedded systems," Consumer Electronics (ISCE). Singapore, pp. 547-551, 2011.

[7] J. Khan, S. Bilavarn, C. Belleudy; "Energy analysis of a DVFS based power strategy on ARM platforms," Faible Tension Faible Consommation (FTFC). Paris, pp. 1-4, 2012.

[8] I. Mhedhbi, R. Ben Atitallah, A. Jemai; "Dynamic slack reclamation strategy for multiprocessor systems," Electrotechnical Conference (MELECON). Yasmine Hammamet, pp. 979-984, 2012.

[9] Jaehyun Park, Donghwa Shin, Naehyuck Chang, M. Pedram; "Accurate modeling and calculation of delay and energy overheads of dynamic voltage scaling in modern high-performance microprocessors," Low-Power Electronics and Design (ISLPED). Austin, pp. 419-424, 2010.

[10] M. Najibi, M. Salehi, A. A. Kusha, M. Pedram, S. M. Fakhraie, H. Pedram; "Dynamic Voltage and Frequency Management Based on Variable Update Intervals for Frequency Setting," IEEE/ACM International Conference on Computer-Aided Design (ICCAD '06). San Jose, pp.755-760, 2006.

[11] M. E. Salehi, M. Samadi, M. Najibi, A. Afzali-Kusha, M. Pedram, S. M. Fakhraie; "Dynamic Voltage and Frequency Scheduling for Embedded Processors Considering Power/Performance Tradeoffs," IEEE Transal. J. Very Large Scale Integration (VLSI) Systems. vol. 19, pp.1931-1935, 2011.

[12] Cheng Xu, Jun Xiao, Lining Zeng, Yan Liu and Manman Pen; "An energy-aware Dynamic Algorithm Based on Variable Interval DVFS Technology," unpbulished.