

Scheduling algorithm for real-time tasks of embedded systems based on dynamic voltage regulation

CHANG WEIGONG¹

Abstract. In order to test the effectiveness of task energy saving scheduling algorithm, an embedded task energy saving scheduling algorithm based on dynamic voltage regulation is designed. Dynamic voltage regulation technology can effectively reduce energy consumption, so it has been widely used. Now many processors have dynamic voltage regulation capability. With the increasing demand for high performance and energy saving, multi-core processors have become the mainstream of the embedded system market. For the multi-core system for energy-efficient scheduling algorithms, task allocation, sequence arrangement and voltage arrangement have a significant impact on energy efficiency. The purpose is to study how to adjust the order of tasks that the voltage conversion of energy minimum, and study how to design the corresponding dynamic voltage regulation algorithm to solve the problem of multi-core system with fixed voltage conversion time or negligible voltage conversion time. The experimental results show that in the initial scheduling length, the method 2 uses the retimed directed acyclic graph as the object of energy-saving scheduling. Based on the above finding, it is concluded that the proposed method achieves good energy saving effect under the condition of meeting the task setting time limit.

Key words. Dynamic voltage regulation, embedded, energy-saving scheduling, algorithm.

1. Introduction

With the rapid growth in the number of mobile devices and embedded device platforms, energy consumption is gaining more and more attention. High performance is currently a growing requirement, and high performance often comes at the expense of huge energy consumption. Chip area continues to shrink, also makes the energy consumption problem becomes more prominent. Many embedded systems are energy-constrained. Therefore, energy minimization is an important issue in embedded system design [1].

Dynamic voltage regulation technology can effectively reduce energy consumption, so it has been widely used. In order to improve energy efficiency, many current

¹Tianjin University of Science & Technology, 300072, China

processors support dynamic voltage regulation and provide a variety of supply voltage files [2]. The emergence of a variety of supply voltage processors in the market, making it a reality in the processor layer for energy management. In order to meet the needs of high-performance and reduce energy consumption, multi-core processors have become the mainstream of the embedded system market. In a multi-core environment, since each processing core must share the same clock, blindly using the existing dynamic voltage methods applied to single-core processors will result in more unnecessary power consumption. It can be seen that the problem of multi-core processor is more complex than multiprocessor. Therefore, it is necessary to take multi-core processor as the research object, study how to reduce energy consumption under the conditions of meet the time constraints.

2. Literature review

Weiser first proposed a variety of dynamic voltage scheduling algorithm [3], but the shortcomings of the study are non-real-time tasks. However, his research has enlightened the research of dynamic voltage scheduling algorithm, and then various dynamic voltage scheduling algorithms have been proposed, which has solved the energy optimization problem of different types of tasks running on different types of systems.

For multi-processor/multi-core processor on the dynamic voltage regulation of energy-efficient scheduling, reference [4] proposed an improved greedy algorithm based on dynamic voltage regulation, this algorithm takes into account the concern that unrelated tasks reduce energy consumption on multiprocessors with time constraints, obtains the widespread attention. However, these references do not address the issue of using dynamic voltage techniques to save energy in multi-core systems.

Reference [5] proposes a two-stage approach to energy optimization for heterogeneous distributed systems under time constraints. First of all, the start time of each task as late as possible with a random number, the value will be sorted in ascending order will be given priority to the task, and then use the table scheduling algorithm to achieve the task of distribution, apply the method of relaxation budget to distribute the relaxation. The system is heterogeneous distributed system.

3. Methods

Definition: On a multi-core processor, for a scheduling generated by a retimed directed acyclic graph, When the last computation task in a processing kernel has no descendant node in the original corresponding acyclic graph, or a computing node having a descendant node but representing a descendant node is assigned to the same processing core as the last executed computing task, the scheduling length of the processing core is the completion time of the last computing task executed on it [6]. When the last computation task in a processing kernel has the descendant node in the node corresponding to the original acyclic graph, And the computational tasks in the computational tasks on behalf of their descendant nodes are assigned

to the different processing cores, and the last executed computational task on the processing core is allocated to the different processing cores, the scheduling length of the processing core is the maximum completion time of the communication task generated by the last computing task executed on the processing core [7].

Theorem: In the case where the voltage conversion time is a fixed constant or can be ignored, when each task is executed in a frequency/voltage mode, the tasks on each processor is executed in descending order of voltage or in ascending order, resulting in minimal voltage transfer energy.

Assuming that the tasks on a certain processor core have $n+1$ frequency/voltage mode, the voltage of the ascending order is $V_1, V_{1+a_1}, V_{1+a_2}, \dots, V_{1+a_{k-1}}, V_{1+a_k}, \dots, V_{1+a_{m-1}}, V_{1+a_m}, \dots, V_{1+a_n}$, where, $0 \leq a_1 \leq a_2 \leq \dots \leq a_{k-1} \leq a_k \dots \leq a_{m-1} \leq a_m \dots \leq a_n$. The voltage conversion energy of the $n+1$ frequency/voltage mode is

$$A + C_r \cdot [(a_{k-1} - a_k)^2 + (a_k - a_{k+1})^2 + \dots + (a_{m-1} - a_m)^2 + (a_m - a_{m+1})^2 + \dots + (a_{n-1} - a_n)^2], \quad (1)$$

where A is the voltage conversion energy of the first k frequency/voltage modes and C_r is the current ratio. Exchange of the position of V_{1+a_k} and V_{1+a_m} will provide

$$A + C_r \cdot [(a_{k-1} - a_m)^2 + (a_m - a_{k+1})^2 + \dots + (a_{m-1} - a_k)^2 + (a_k - a_{m+1})^2 + \dots + (a_{n-1} - a_n)^2]. \quad (2)$$

Subtraction of formula (3) from formula (2) provides

$$2C_r \cdot (a_k - a_m) \cdot [(a_{k-1} + a_{k+1}) - (a_{m-1} + a_{m+1})] \geq 0. \quad (3)$$

The same result can be obtained by swapping the positions of any two frequency/voltage modes, so the tasks on the processing core are performed in the order of ascending order of the voltage levels to minimize the conversion energy [8]. Similarly, the tasks of the processor cores are performed in the descending order of the voltage profiles produced the minimum switching energy. It is considered that the transition time from the operating state to the idle state of the applied processing core may be a fixed constant and cannot be ignored, in this case, when the difference between the given time limit and the completion time of the last computation task on a certain processing core is smaller than the transition time from the operating state to the idle state, the remaining time is run in the frequency/voltage mode of the last computing task running on the processing core, therefore, in order to reduce the energy consumption, the tasks on the respective processing cores are executed in descending order of the voltage level for a processor whose conversion time from the operating state to the idle state is a fixed constant and cannot be ignored. While the transition time from the running state to the idle state is a negligible processor, the tasks on each processing core can be performed in descending or ascending order

of the voltage file [9].

After the above theoretical analysis, an algorithm is proposed to solve the energy-saving scheduling problem of multi-core processors with task-dependent applications. The energy-saving scheduling algorithm takes the isomorphic multi-core system with dynamic voltage regulation capability as the research object, and realizes the compaction energy and the conversion energy. The work flow diagram of the algorithm is shown in Fig. 1, and the implementation steps are shown in Table 1.

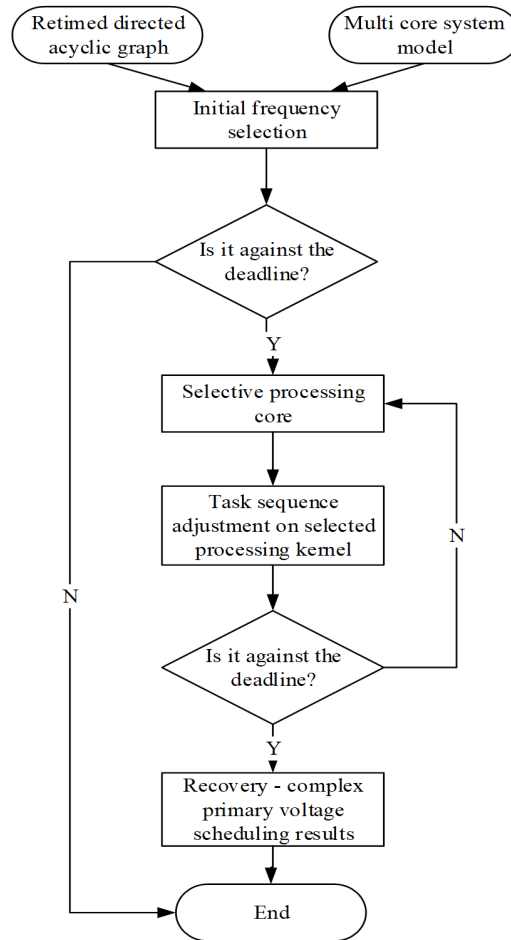


Fig. 1. Design flow diagram of energy - saving scheduling algorithms for dynamic voltage regulation multicore systems

In order to verify the energy-saving effect of the proposed energy-saving scheduling method, the proposed method (method 2) is compared with the method (method 1) proposed in the literature [10]. In the simulation experiment, five randomly generated directed acyclic graphs are used, named task set 1–task set 5, respectively. The number of calculation tasks varies between 27 and 68, while the number of

communication edges varies between 45 and 94. The number of calculation task cycles satisfies a uniform distribution between 10 and 40, specific task characteristics shown in Table 2.

Table 1. Implementation steps of energy-saving scheduling for multi-core system with dynamic voltage regulation

Stage	Step
Initial scheduling	Step 1: The calculation tasks are arranged in descending order of cycles, the mapping of the tasks to the processing kernel is carried out according to the principle of priority assignment of the shortest processing kernel (for the calculation task).
	Step 2: For each processing core, the calculation task with the minimum communication time is set as the computation task last executed by the processing core while the order of the other calculation tasks is kept unchanged.
	Step 3: The computational tasks on each processing core are arranged at the lowest frequency (except for the frequency corresponding to the idle state) of the processing core. If the generated scheduling length is less than or equal to the given time limit, the scheduling is accepted and returned, otherwise go to step 4.
	Step 4: The computational tasks on each processing core are arranged at the highest frequency of the processing core.
Re-scheduling	Step 5: In the processing core with the shortest scheduling length, select a computing task T, When its frequency is reduced by a frequency file and in accordance with its changes in the voltage file in descending order, the computational task on the processing core minimizes the total energy consumption, the computation task T is extended in execution time, that is, the frequency profile of the calculation task is reduced one rating level.
	Step 6: If the new schedule length of the processing core is less than or equal to a given time limit, then accept the change, the calculation task T is executed at the changed frequency profile, the calculation task on the processing core is executed in the order of decreasing voltage range, updating the schedule, go to step 5; otherwise go to step 7.
	Step 7: The calculation task T is executed at the frequency before the conversion, the calculation tasks on the processing core are executed in the order before the change, that is, accept the last change, and return.

For the two computational tasks that have dependencies in the original acyclic graphs, if they are assigned to different processing cores, the number of communication cycles between them is from 1 to 4 randomly selected values. For the two computational tasks that have dependencies in the original acyclic graphs. If they are assigned to the same processing core, the number of communication cycles between them is zero. In the studied system, the frequency on the bus is fixed, set

to 208 MHz. The generated task set was simulated on a multi-core system with 2 and 4 cores respectively. The processor using the Intel PXA270 power model, it can operate in 7 frequency/voltage modes. For Intel PXA270, its mode transition time is negligible. Calculate the voltage conversion energy, the initial time limit is the initial scheduling length of the two nuclear processing system in the method 1, then take the 10 % of the schedule length as the step size, set the time limit, and take the average value from obtained energy consumption.

Table 2. Task set feature table

Task Set Name	Number of nodes	Number of edges	Number of cycles
Task Set 1	27	45	681
Task Set 2	35	54	880
Task Set 3	46	67	1233
Task Set 4	57	74	1520
Task Set 5	68	94	1808

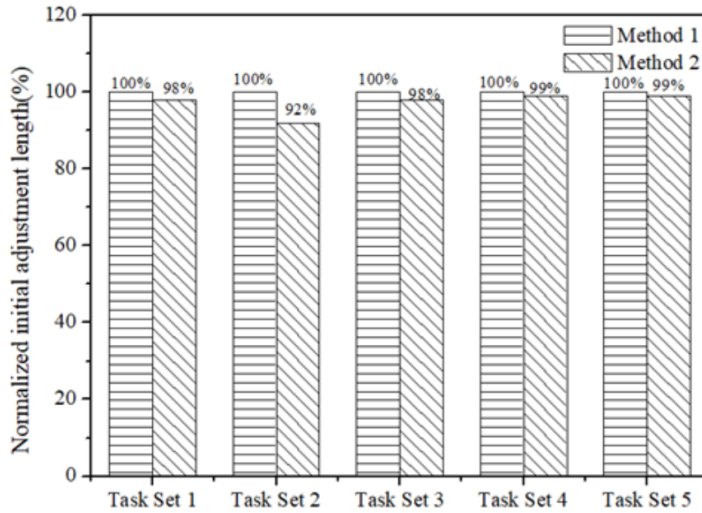
4. Results

The simulation results of method 1 and method 2 are compared respectively from the initial scheduling length and the average energy consumption. Figure 2 shows the initial scheduling length comparison results of method 1 and method 2. Figure 3 shows the results of the average energy consumption comparison of method 1 and method 2.

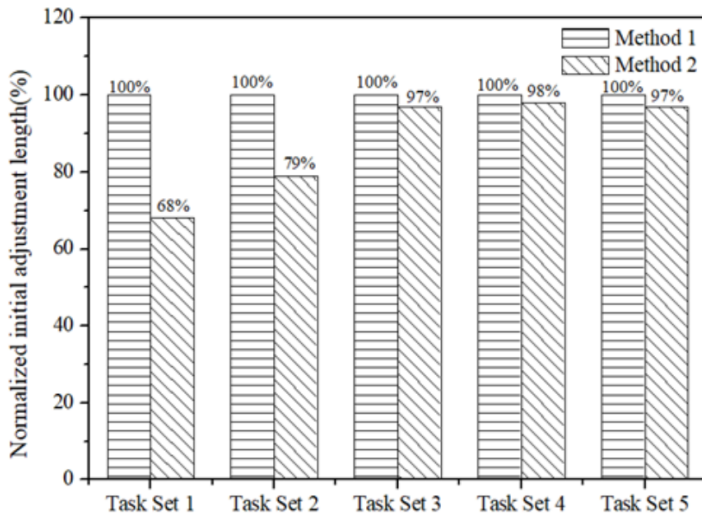
5. Discussion

The results show that, in the initial scheduling length, the method 2 uses the retimed directed acyclic graph as the object of energy-saving scheduling, the tasks in the retimed directed acyclic graphs do not have data dependency in iteration, and improve the parallelism of tasks. Thereby reducing the initial scheduling length. While the method 1 uses a directed acyclic graph, in the directed acyclic graph, the data dependency of the task weakens the parallel ability of the task, resulting in the initial scheduling length of the task is often larger than the initial scheduling length generated by the retimed directed acyclic graph, at the beginning of the average energy consumption, using the retimed directed acyclic graph as the energy-saving scheduling object, so that make the dependencies task in the original directed acyclic graph no longer exist data dependency within iteration, while there is only data dependency between iterations, improve the parallel ability of the task and can produce more slack. When the retimed directed acyclic graph is used as an energy-saving scheduling object, the slack generated on one processing core can be assigned to each task on the processing core, thus providing more frequency/voltage mode adjustment opportunities for the task. Also, since the task execution order is determined in the frequency/voltage mode selection process according to the voltage

level of the task on each processing core, it is possible to simultaneously reduce the dynamic energy consumption and the voltage conversion energy consumption.



(a) Comparison of Initial Scheduling Lengths on Two Processing Cores



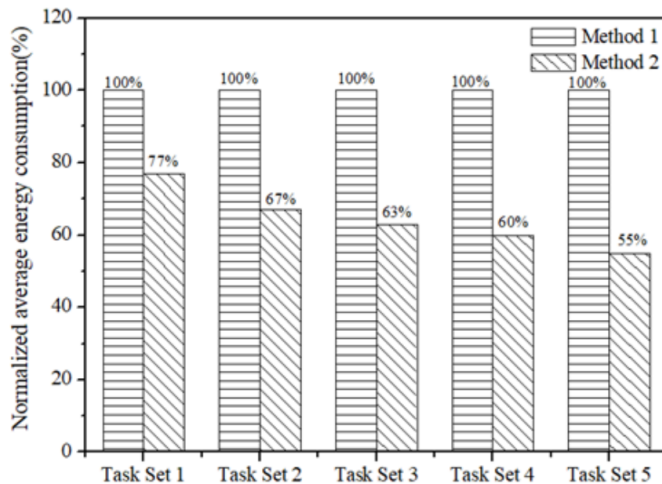
(b) Comparison of Initial Scheduling Lengths over Four Processing Cores

Fig. 2. Comparison of initial scheduling lengths of two methods

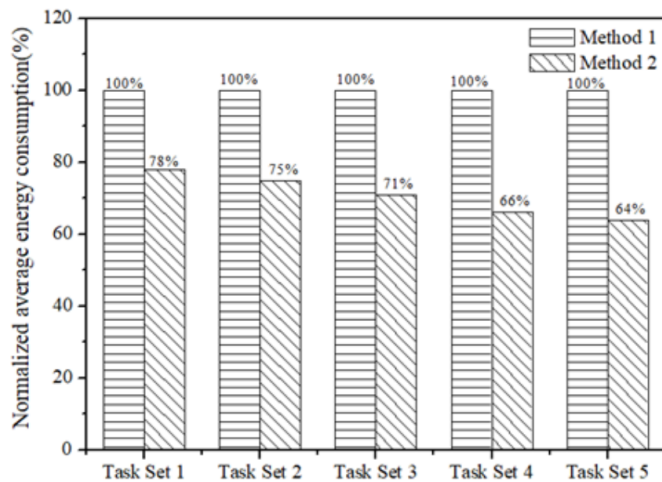
Method 1 due to the influence of data dependency in the iteration, the slack between tasks can only be used to reduce the frequency of the adjacent task and re-

duce the frequency/voltage mode adjustment. Data dependency within the iteration limits the ability to take advantage of multi-core, and therefore leads to a reduction in energy-saving opportunities.

From the simulation results in Fig. 2 upper and bottom parts (a and b), it can be seen that the initial scheduling length of method 2 is smaller than the initial scheduling length of method 1.



(a) Comparison of average energy consumption on two processing cores



(b) Comparison of average energy consumption over four processing cores

Fig. 3. Comparison of average energy consumption of two methods

From the simulation results in Fig. 3 upper and bottom parts (a and b), it can be seen that the energy saving effect of the method 2 is better than that of the method 1.

6. Conclusion

The related research work of energy saving algorithm using dynamic voltage regulation technology is introduced. Taking the multi-core processor as the research object, in order to solve the problem that the voltage conversion time of some processors is fixed or negligible, an energy-saving scheduling method is proposed to reduce energy consumption and voltage conversion energy consumption. The method can constrain the frequency/voltage regulation mode and can use the timed task graph to schedule the object. It can also perform frequency/voltage mode selection and task sequence adjustment. Dynamic voltage regulation technology can effectively reduce energy consumption, so it has been widely used. In order to improve energy efficiency, many current processors support dynamic voltage regulation and provide a variety of supply voltage files. In order to meet the needs of high-performance and reduce energy consumption, multi-core processors have become the mainstream of the embedded system market. In a multi-core environment, since each processing core must share the same clock, blindly using the existing dynamic voltage methods applied to single-core processors will result in more unnecessary power consumption. It can be seen that the problem of multi-core processor is more complex than multiprocessor. Therefore, it is necessary to take multi-core processor as the research object, study how to reduce energy consumption under the conditions of meet the time constraints. The simulation results show that the proposed method achieves good energy saving effect under the condition of meeting the task setting time limit.

References

- [1] J. WANG, W. BAO, X. ZHU, L. T. YANG, Y. XIANG: *Festal: Fault-tolerant elastic scheduling algorithm for real-time tasks in virtualized clouds*. IEEE Transactions on Computers 64 (2015), No. 9, 2545–2558.
- [2] Y. W. ZHANG, R. F. GUO: *Low-power scheduling algorithms for sporadic task with shared resources in hard real-time systems*. Computer Journal 58 (2015), No. 7, 1585–1597.
- [3] Y. W. ZHANG, R. F. GUO: *Low-power scheduling algorithm for mixed task in real-time system*. Journal of Jilin University (Engineering and Technology Edition) 45 (2015), No. 1, 261–266.
- [4] S. RAJPRADHAN, S. SHARMA, D. KONAR, K. SHARMA: *A comparative study on dynamic scheduling of real-time tasks in multiprocessor system using genetic algorithms*. International Journal of Computer Applications 120 (2015), No. 20, 975–8887.
- [5] J. H. ANDERSON, J. P. ERICKSON, C. UMAMAHESWARI, B. N. CASSES: *Optimal semi-partitioned scheduling in soft real-time systems*. Journal of Signal Processing Systems 84 (2016), No. 1, 3–23.
- [6] Z. PENG, G. WANG: *An optimal energy-saving real-time task-scheduling algorithm for mobile terminals*. International Journal of Distributed Sensor Networks 13 (2017), No. 5, 1–12.

- [7] M. L. C. PRABHAKAR, K. MANIVANNAN: *Survey of hard real time task scheduling algorithm on multicore processor*. Asian Journal of Research in Social Sciences and Humanities 6 (2016), No. 12, 135–153.
- [8] H. ALHUSSIAN, N. ZAKARIA, F. A. HUSSIN, H. T. BAHBOUH: *USG, an un-fair, semi greedy real-time multiprocessor scheduling algorithm*. REACTION (Conferences and Workshops), Vancouver, Canada (2013).
- [9] S. SENOBARY, M. NAGHIBZADEH: *Semi-partitioned scheduling for fixed-priority real-time tasks based on intelligent rate monotonic algorithm*. International Journal of Grid and Utility Computing 6, (2015), Nos. 3–4, 184–191.
- [10] H. ALHUSSIAN, N. ZAKARIA, A. PATEL: *An unfair semi-greedy real-time multiprocessor scheduling algorithm*. Computers and Electrical Engineering 50 (2016), 143–165.

Received May 7, 2017