

# Preprocessing of DAG Tasks based on Optimal Frequency Determination Method

Yuxiang Ge <sup>a</sup>, Youlin Ruan

School of Wuhan University of Technology, Wuhan 430010, China;

<sup>a</sup>13675869755@163.com

**Keywords:** DAG, Global Optimization, Multi-core, Optimal frequency, Energy conservation, Global DVFS.

**Abstract.** With the application of multi-core processor, the data processing ability has been improved. More and more scholars pay more attention to the processing optimization of the task with the constraint of each other. Marco E.T. Gerards et al. have given the way to choose the optimal clock frequencies that minimize the energy consumption for global DVFS when processing real-time tasks. This article presents a kind of DAG task preprocessing, then the energy consumption reduces 7%.

## 1. Introduction

With the increasing awareness of environmental protection, people pay more and more attention to energy conservation. As one of the fastest growing industries in the world, the information technology industry accounts for 2% of global carbon emissions and this proportion will double by 2020. For mobile multi core platforms (such as laptops, smartphones) to reduce energy consumption is not only for environmental protection, but also related to the performance of the product. Therefore, the research on task scheduling strategy for multi core platform for energy saving has become a new hot spot. At present, most scholars focus on the research of scheduling strategy. Through the optimization of scheduling strategy to achieve the purpose of energy saving. In addition, application of DVFS technology to reduce energy consumption has a good effect. Marco E.T. Gerards et al.[1] show how to choose the optimal clock frequencies that minimize the energy for global DVFS, and they discuss the relationship between scheduling and optimal global DVFS. With the techniques an energy consumption reduction of 30% can be achieved. And this article presents a kind of preprocessing for DAG tasks that does not affect the constraints between subtasks. With the processing the energy consumption reduction of 7% can be achieved.

## 2. Related Work

With the advent of the era of big data, data begin to grow explosively, the tasks of application become more and more complex. Meanwhile CPU generated more and more energy consumption. Scholars have adopted various methods and techniques to reduce the energy consumption of CPU. In addition, ensure the real-time task completed before the deadline.

The application of DVFS has made great contribution in reducing CPU energy consumption. It is a popular energy management technique, lowers the voltage and clock frequency to reduce the energy consumption [2]. There are tow flavors of DVFS in multi-core/single-chip system: local DVFS and global DVFS. The difference between the two is that local DVFS sets the clock frequency and voltage for each core, and the voltage, frequency can be different between with each other. But global DVFS set the clock frequency and voltage for entire chip. In this paper we use global DVFS.

Marco E.T. Gerards et al. reduce the energy consumption by choosing the optimal frequency. And proved that the optimal clock frequency depend on the parallelism of tasks.

Through research, we found a way of preprocessing of DAG tasks that can reduce the energy consumption. And we discussed in witch case we can reduce the energy consumption with preprocessing of DAG tasks. Through experimental analysis, the energy consumption reduced by 7%.

All the work was completed in the system which with  $M > 1$  homogeneous cores, in this paper.

### 3. Model

#### 3.1 System Model

We adopted homogeneous multi-core system. We assume that the cores are coupled with highly efficient communication mechanisms. And in this case we consider all the tasks arrive at time 0. All the cores have the same clock frequency, if they are running. And the clock frequency can be changed at any time. As global DVFS is used, the clock frequency is used for all the cores which are running, at the same time. We consider that the speed of the cores scale linearly with the clock frequency, and we neglect the influence of caching and shared resources.

#### 3.2 Power Model

The system is homogeneous multi-core system employs global DVFS, which means that at any time  $t$  all the  $M$  cores running at the same clock frequency. The power consumption consists of static power and dynamic power. Because the static power is only small part of the total power consumption.

Static power—the part of the power that is independent of the clock frequency—is typically modeled using a linear function of the voltage [3]. As we use the common assumption that the optimal voltage and clock frequency are linearly related, we get the static power as a linear function of the clock frequency  $f$  with non-negative coefficients:

$$P_s(f) = c_1 f + c_2 \quad (3-1)$$

Dynamic power per core for clock frequency  $f$  is given by [4]:

$$P_d(f) = c_3 f^\alpha \quad (3-2)$$

Where  $c_3 > 0$  is a constant that depends on the average switched capacitance and the average activity factor. The value  $\alpha$  ( $\alpha \geq 2$ ) is a constant (usually close to 3)

The power consumption depends on how many cores are active, i.e., how many cores have tasks scheduled on them. By using clock gating, the clock frequency of an inactive core can be set to zero with little overhead. The power function  $p_m$  now gives for  $m$  cores the total power as a function of the clock frequency, which is the dynamic power times the number of active cores  $m$  plus the static power:

$$P_m(f) = m c_3 f^\alpha + c_1 f + c_2 \quad (3-3)$$

### 4. Preprocessing Algorithm

In this paper we use LPT [5] algorithm as scheduling algorithm. In the DAG task, there is a class of subtasks that do not activate the subsequent node after the completion of the task. It is necessary to wait for the completion of other subtasks. So we can change the order of execution of such subtasks. These subtasks are called free-subtask, in this paper.

Example, the task T1 like figure 1. In the task the subtask  $t_4$  can be can be delayed execution. In this case the task can change to the task T2 like figure 2. The workload for  $t_0$  to  $t_7$  is 5, 4, 5, 6, 5, 6, 7, 7. And the deadline for the task is 40.

According to LPT [5] and the way of choose optimal frequency [1] the energy consumption of T1  $E1 = 18.196$ . After the preprocessing of the task T1 change to T2 the energy consumption  $E2 = 16.017$ , reduced by 12%.

Preprocessing Algorithm:

Find the free-subtask among the subtasks which can execute right now.

Change the execution of the free-subtask, calculate the energy consumption and compare with the former. If the energy consumption is smaller change the order. Then jump to d

If no reduction in power consumption, judge if the free-subtask can be delay any more. If it can be delay, it will be the new free-subtasks.

Execute the subtasks, then jump to a until the Task is end.

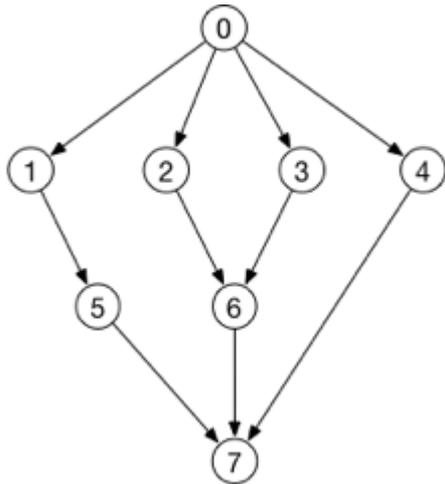


Fig.1 model of task T1

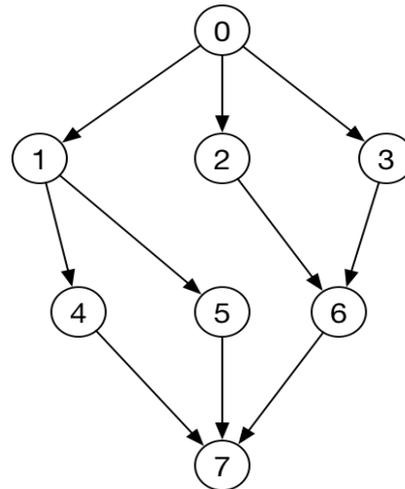


Fig.2 task T2 change from T1

### 5. Analytic evaluation

We use the “Standard Task Graph” (STG) set by Tobita and Kasahara [6], to compare our approach.

From the STG set we use the set of 180 graphs, with 50,100 subtasks with precedence constraints. Figure 3, 4 show the optimization efficiency for the 2 sets.

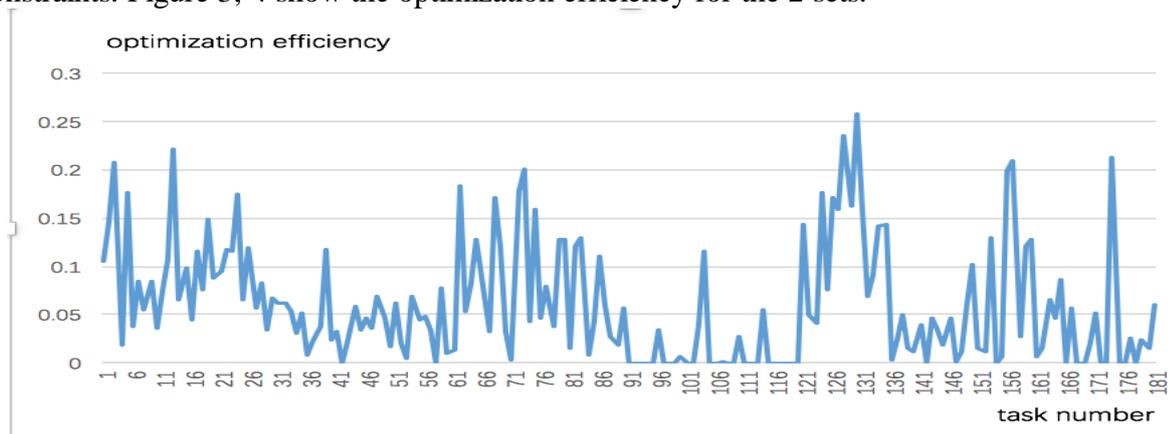


Fig. 3 The optimization of the set with 50 subtasks

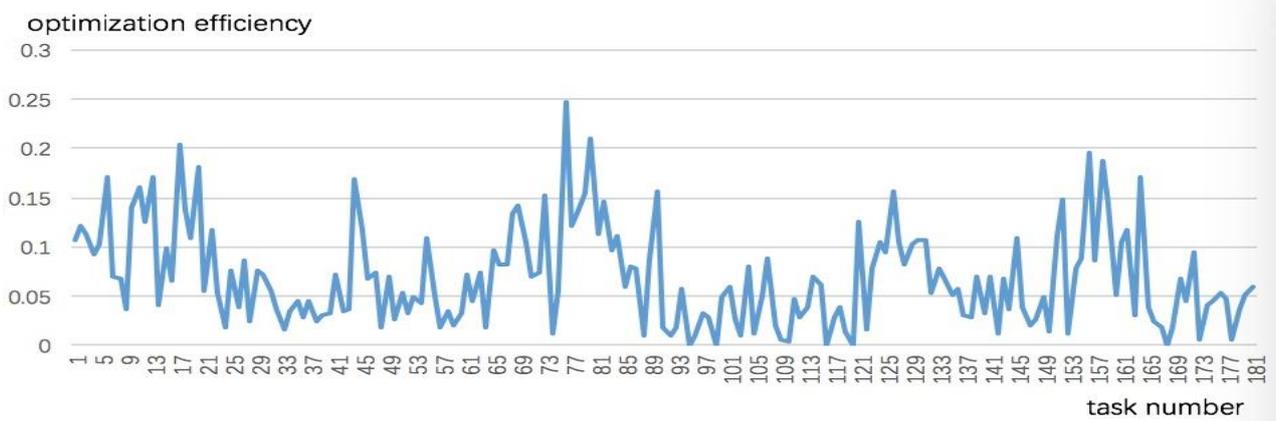


Fig. 4 The optimization of the set with 100 subtasks

From the 2 fig we can find the best optimization efficiency is about 25%. With our techniques an energy reduction of 7% can be achieved. In the Fig 3 there are 35 task could not be optimized. But in Fig 4 shows that there are only 3 tasks could not be optimized. So we can find the more complex the task is, the more appropriate the algorithm is.

## 6. Simulations

In the paper we propose a method for task preprocessing, and the energy consumption reduced by 7%. And when the tasks have 100 subtasks, almost all the tasks are optimized. It is proved that this preprocessing algorithm is very suitable for complex tasks. Next we will focus on proposing a scheduling algorithm, which can reduce power consumption further.

## References

- [1] Marco E.T. Gerards, Johann L. Hurink , jan Kuper, . IEEE Transactions on Computers.2014, [2] 10.1109/TC.2014.2345410.
- [3] M. Weiser, B. Welch, A. Demers, and S. Shenker, “Scheduling for reduced CPU energy,” in *Mobile Computing*, T. Imielinski and H. F. Korth, Eds. Springer US, 1996, vol. 353, ch. The Kluwer International Series in Engineering and Computer Science, pp. 449–471, 10.1007/978-0-585-29603-6 17.
- [4] S.Park,J.Park, D.Shin,Y.Wang,Q. Xie,M.Pedram,andN.Chang, “Accurate modeling of the delay and energy overhead of dynamic voltage and frequency scaling in modern microprocessors,” *Computer-Aided Design of Integrated Circuits and Systems*, IEEE Transactions on, vol. 32, no. 5, pp. 695–708, 2013. [Online]. Available: <http://dx.doi.org/10.1109/TCAD.2012.2235126>
- [5] F. Yao, A. Demers, and S. Shenker, “A scheduling model for re- duced CPU energy,” in *Proceedings of IEEE 36th Annual Foundations of Computer Science*, 1995, pp. 374–382.
- [6] R. L. Graham, “Bounds on multiprocessing timing anomalies,”*SIAM Journal on Applied Mathematics*, vol. 17, no. 2, pp. 416–429,1969.
- [7] T. Tobita and H. Kasahara, “A standard task graph set for fair evaluation of multiprocessor scheduling algorithms,” *Journal of Scheduling*, vol. 5, no. 5, pp. 379–394, 2002. [Online]. Available: <http://dx.doi.org/10.1002/jos.116>