Simulations on the Energy Consumption of WRF on Meteorological Cloud

Junwen Lu[1], Yongsheng Hao[2,3](✉), and Xianmei Hua[4]

[1] Engineering Research Center for Software Testing and Evaluation of Fujian Province,
Xiamen University of Technology, Xiamen 361024, China
[2] School of Mathematics and Statistics, Nanjing University of Information Science
& Technology, Nanjing 210044, China
yongshenghao@yahoo.com
[3] Network Centre, Nanjing University of Information Science
& Technology, Nanjing 210044, China
[4] Xiamen Lihan Information Technology, Xiamen 361024, China

Abstract. In the paper, we try to evaluate the energy consumption of meteorological applications on meteorological cloud of on different kinds of processors. We take WRF (Weather Research and Forecasting model) model as the typical model. Three major factors are including in the evaluation: the energy consumption, the execution time, and the parallelism. The moldable parallel tasks have a scope of parallelisms. But after the job has an execution state, and the parallelism cannot be changed during the execution. Different to most of past research, our system support slots time and every job needs a few slot times to execute it. We give a detailed analysis of DVFS (Dynamic Voltage and Frequency Scaling) model for WRF and evaluate the different performance of three kinds of CPUs in different aspects, and at last, based the analysis of the attributes of the three CPUs and the nonlinear speedup of WRF under different numbers of resources, simulations result are given to address the energy consumption of WRF under different environments. We hope our research can help us to enhance the scheduling method of parallel tasks.

Keywords: Moldable parallel tasks · Energy consumption · WRF · Simulations · Energy efficiency

1 Introduction

Many methods have been used to the schedule parallel tasks. Most of methods pay attention to the execution time of the parallel tasks [1]. Those methods try to meet the deadline of job and maximize the output of the system. Different to those methods, this paper pays attention to the energy consumption of parallel tasks (ECPT) when the system supports slot time. A detailed example of WRF is given to discuss the energy consumption under different voltages and computing frequency. We do not want to give and compare the scheduling method. Our target is to evaluate some factors when every

X. Zhang et al. (Eds.): CloudComp 2019/SmartGift 2019, LNICST 322, pp. 594–602, 2020.
https://doi.org/10.1007/978-3-030-48513-9_47

jobs needs some slots to execute it. Those factors include: the execution time, energy consumption and CPUs. We hope our study can help us deeply understand the scheduling influence to the job and the system targets. Besides that, because of the importance of the weather forecast, we also hope we can give methods for the scheduling of weather models.

ECPT has been widely evaluated in many platforms. DAG (Directed Acyclic Graph) are widely used to simulate parallel tasks. According to the DAG of paralle tasks, some methods are addressed to save energy consumption, and keep the performance of others aspects, such as execution time, fairness and so on [2, 3]. Different to those methods, this paper tries to consider the two aspects at the same time: the speedup of the parallel tasks and the DVFS of the resources.

In this paper, we will give a example of parallel tasks-WRF, which is widely used in the weather forecast. Based on the speedup of WRF [4] and the DVFS of three CPUs, we will model and evaluate the energy consumption of WRF under different conditions. Last, simulations are given for the energy consumption of WRF.

The contributions of the paper include:

(1) We give the DVFS model for WRF. The model supports wall time (or slot time). In fact, this model can be used for all moldable parallel tasks that which support wall time.
(2) We model consumed resources and the related energy consumption for moldable parallel tasks, which supports wall time, and can be used for all moldable parallel tasks.
(3) We give a detail comparisons of the performance of three kinds of CPUs in the energy efficiency.
(4) We give the energy consumption of WRF on three kinds of CPUs. We hope this can help we to better understand the energy consumption of MPA (moldable parallel tasks) under different QoSs (Quality of services) requirements (such as deadline, execution time).

We organize the paper as follows: The related work is addressed in Sect. 2. We describes the consumed resource model of WRF in Sect. 3. Section 4 gives the DVFS model of the resource. Section 5 illustrates energy consumption model of WRF. Section 6 addresses the simulation results. Section 7 give the sumary of the paper.

2 Related Work

Energy consumption is a hot problem is the scheduling. For the parallel tasks, many methods have been proposed from different aspects, either in the Grid, Cluster, or in the Cloud. Those methods mainly focus on the attribute of DAG, and use DVFS technology to reduce energy consumption. Energy-Aware method [2] are used to save energy for scheduling parallel tasks in Clusters. They take a threshold to make a tradeoff between energy consumption and execution time. I. Manousakis et al. gives a OpenMP-like tool to support energy-aware scheduling task-parallel applications. The tool traces the parallel execution and supports estimating the per-task energy consumption [8]. In order to reduce energy consumption and satisfy the QoS requirement, T. Zhou et al. [9] address a Workflow task scheduling algorithm based on DVFS under the Cloud environment. Through merging the relatively inefficient processors by reclaimings the slack time, which use the slack time after severs are merged. Y. Xia et al. [10] present a novel stochastic framework to enhance energy efficiency. There are four important facts in the scheduling, including tasks (virtual machine requests) arrival rate, and other metrics of datacenter servers. And then using queuing network-based analysis, they address analytic solutions for three targets. Since scientific workflows are widely executed across multiple clouds, and thus enhances energy consumption, X. Xu et al. [11] uses EnReal to hand those changes. They also address a energy-aware scheduling algorithm for virtual machine management to support scientific workflow executions.

Some methods also engage to save the energy consumption by considering the speedup of the parallel tasks [12, 13]. Past work always based upon the Amdahl's law and assumed an ideal system without overheads [13].

Those methods all focus on how to schedule resources to save energy consumption. And in this paper, we address the relation between the performance, the energy consumption, and the execution time. We hope the result can help us further to improve the performance and save energy. We use a WRF model to discuss those relation.

3 Consumed Resources Model of WRF

The speedup in Fig. 1 has a nonlinear speedup. WRF can work under different speedups under different parallelisms, WRF is modeled as:

$$WRF_i = \left\{ <para_{i,j}, et_{i,j}> \mid max_i \geq j \geq 1 \right\} \tag{1}$$

For the job WRF_i, the parallelism has a scope between 1 and max_i. If $para_{i,j}$ is smaller than max_i, $et_{i,j}$ will be shorter; with the increase of the number of resources (more than max_i) for the job, it lengthens the execution time, because the system consumes more time to get (or send) data between various nodes. $para_{i,j}$ and $et_{i,j}$ are the parallelism and the related execution time. The speedup of the job is [6]:

$$sp_i = \frac{et_{i,1}}{et_{i,j}}$$

In Fig. 1, WRF (for some certain weather parameters) can work under the parallelism in [1, 48]. The X axes are the numbers of the allocated resources and the Y axes are the execution times. When the value of X axes is more than 48, enhancing number of resources cannot save the execution time. So in the scheduling, it is not always right more resource can decrease the execution time. When we consider the energy consumption, the problem becomes more difficult.

There are some reasons to the explain the relation in Fig. 1. Amdahl's Law and Gustafson's Law [14, 15] are always used to explain the relation in Fig. 1.

Amdahl's Law: the maximum parallel execution time percentage is *par*, the serial execution time percentage is 1–*par*, and the parallelism is *p*. Amdahl's Law gives the

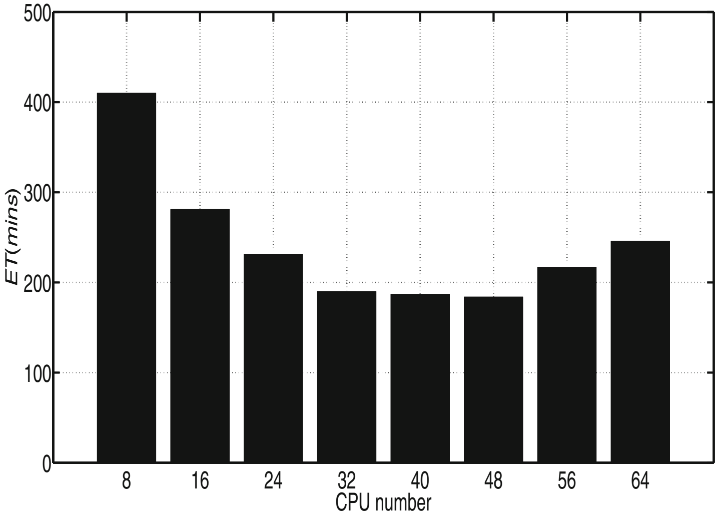

Fig. 1. Execution time vs. number of CPUs

speedup sp of a parallel task when the parallelism is p:

$$sp \leq \frac{1}{(1 - par) + \frac{par}{p}} \tag{2}$$

Some people criticized this that it dose not consider overheads, including communication, synchronization, and so on. Amdahl's Law has drawback especially with the increasing number of resources, the execution time would be lengthen.

Gustafson's Law is the law for scaled speedup. Scaled speedup is as follows:

$$sp \leq p + (1 - p)s$$

where, p is the parallelism, s is the serial execution time percentage of a parallel task on a given data set size.

Most of time, the speedup of a parallel task has a maximum value and can not be enhanced when it gets the maximum parallelism. Only some of time, the speedup is a constant, called the phenomenon as super-linear speedup. Most of time, the speedup has a range under a certain condition. We can get the maximum speedup by Amdahl's Law and Gustafson's Law. But for the two laws, there is difficulty in estimating the value of par and s.

We also can get some support for Fig. 1 for the two laws. With the enhancement of parallelisms, WRF reduces in the execution time (while the number of resources is less than 48), then, the value gradually increase with the further increasing of the number of resources (which the number of resources is more than 48). The reason is with the increasing of the resources, the system consumes more cost in exchanging information between resources.

At the same time, WRF supports wall time (wat), which is the maximum of continuous execution time. After a wat, the system stops the parallel tasks and then saves the data to the next wat. This wall time helps the user to find the error in the algorithm and improves the efficiency of the system if the job has an error in it.

In formula (1), we suppose the WRF_i works under the standard CPU, in other words, the resource has a working frequency of 1.

4 DVFS Model of WRF

A DVFS-enable resource has a mapping-set between the working voltage v_n and executing frequencies f_n [5]:

$$\mathrm{RS} = \cup_{1 \le ntemp \le N}\{v_n, f_n, e_n\} \tag{3}$$

Where

- v_n is the nth resource supply voltage;
- f_n is the nth resource computing frequency;
- e_n is the power consumption when the resource is working under v_n and f_n.

If $n_1 < n_2$, then $f_1 \le f_2$. Though the resource supplies the dynamic voltage, and here, we suppose that the voltage is a constant during in a time slot (wall time). Because if the voltage of every resources can be changed during the time slot, there is difficulty to keep steps of every parallel tasks.

We model the energy consumption according to CMOS logic circuits model. The dynamic power consumption can be defined as:

$$P_{dy} = \alpha C_l V_s^2 f_c \tag{4}$$

Where,

- α: switch rate;
- C_l: capacitance;
- V_s: working voltage;
- f_c: computing frequency.

Suppose the highest working voltage is V_{hest}, and the maximum dynamic consumed energy is P_{hest}. Then we can get the value of the dynamic consumed energy power consumption P_{sp} under the supply voltage is V_{sp} and the clock frequency is f_{sp}:

$$P_{sp} = P_{hest}\frac{f_{sp}V_{sp}^2}{f_{hest}V_{hest}^2} \tag{5}$$

Even there are no jobs on the resource, the resource also needs consuming energy, called as the idle power consumption P_{idle}. The total power consumption P is:

$$P = P_{sp} + P_{idle} \tag{6}$$

Formula (7) is used to denote energy efficiency:

$$E - F_n = \frac{e_n}{f_n} \tag{7}$$

5 Energy Consumption for WRF

For the parallel task WRF_i, let us suppose it has been assigned $para_{i,temp}$ resources, in other words, the parallelism of the job is $para_{i,temp}$. When the resources are working under $<v_{ntemp}, f_{ntemp}, e_{ntemp}>$ and the wall time of resources is wat. The execution time of WRF_i is:

$$ET(WRF_i) = \left\lfloor \frac{et_{i,temp}}{f_n * wat} \right\rfloor * wat \tag{8}$$

The total number of consumed resources:

$$CR(WRF_i) = \left\lfloor \frac{et_{i,temp}}{f_n * wat} \right\rfloor * wat * para_{i,temp} \tag{9}$$

The total consumed idle energy consumption is:

$$IEC(WRF_i) = \left\lfloor \frac{et_{i,temp}}{f_n * wat} \right\rfloor * wat * para_{i,temp} * P_{idle} \tag{10}$$

The total dynamic energy consumption:

$$DEC(WRF_i) = \frac{et_{i,temp}}{f_n} * para_{i,temp} * P_{dy} \tag{11}$$

According to formula (10) and (11), the total energy is:

$$TCE(WRF_i) = IEC(WRF_i) + DEC(WRF_i) \tag{12}$$

6 Simulations and Comparisons

In this section, first, we will evaluate three kinds of CPUs [5] on the energy efficiency, then we will give a comparison of energy consumption of WRF on the three kinds of CPUs.

In this section, We give a comparison for the energy consumption of WRF on three kinds of CPUs. Evaluate the energy consumption of WRF.

The WRF is modeled as [5], which has a nonlinear speedup. We evaluate the energy consumption of a same task under three kinds of CPUs.

Figures 2, 3 and 4 are the energy consumption of WRF (under a certain kinds of weather parameters) on different kinds of CPUs. In Figs. 2, 3 and 4, the axis has the same meaning. The X axis are the work state of Different kinds of CPUs. Those include the working voltage and the working frequency. The Y axis are the parallelisms of resources, in other words, they are the parallelisms of WRF. The Z axis are the energy consumptions of WRF under different kinds of working states. From Figs. 2, 3 and 4, we find that no matter under which kinds of CPUs, the energy consumption increases with the increasing of parallelisms. The reason is with the increasing of parallelism, the job need consumed more resources. We also find that INTEL540 always has the largest

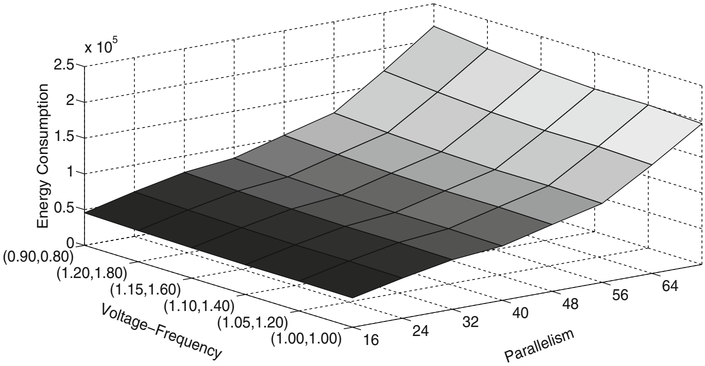

Fig. 2. Energy consumption of a WRF job on AMD34

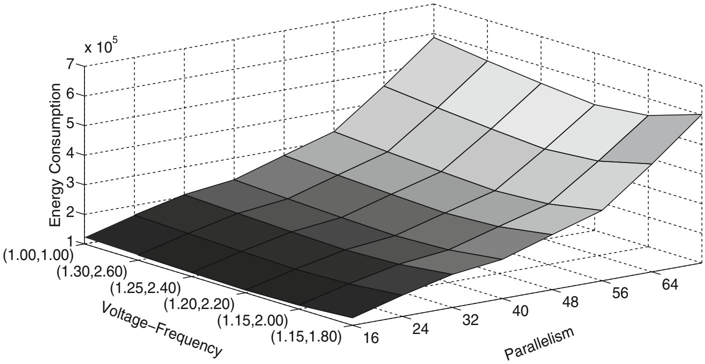

Fig. 3. Energy consumption of a WRF job on AMD2218

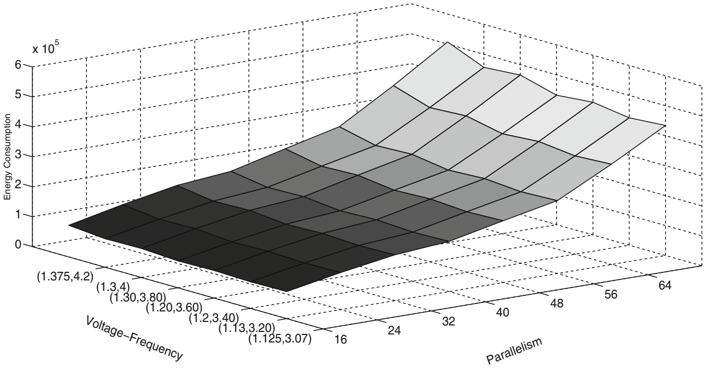

Fig. 4. Energy consumption of a WRF job on INTEL540

value in energy consumption, on the contrary, AMD34 always has the lowest value. For the special CPUs, even under the same parallelism, we also need to assign the suitable voltage and computing frequency to reduce energy consumption. For example, under INTEL540, when the parallelism is 64, (1.125, 3.07) has the lowest energy consumption.

In general, the parallelism has the largest influence to the energy consumption, and then we should select the right kinds of CPUs and decide the right working voltage and frequency to ensure the job satisfy the deadline requirement. Most of time, we can choose the resources with the highest energy efficiency while the job satisfies its deadline. But sometime, if the job can not be completed before its deadline, we should give a priority to the resources that have a higher speed.

7 Conclusions

Here, we analyze the energy consumption of WRF. We give the models for DVFS, resources model and energy consumption model. The performance of three kinds of CPUs is given in the simulation when the system support slot time. And based on the three kinds of processors, we model the energy consumption of WRF. Simulation results show that we should not only select the right processor which has the lowest power consumption, but also reduce energy consumption by selecting the working voltage and frequency. Some methods also have been used to reduce overall *energy according to multiple* aspects [16–18]. Though those methods target to reduce the energy consumption, but none of them consider the attributes of the scheduling target (the attributes of the program) and the resources (the attributes of those resources) at the same time. If we can consider those aspects at the same time, we can find some new methods to minimize the energy consumption of the scheduling. In the future, we hope we can not only evaluate the energy consumption of more meteorological models, but also we will consider different kinds of deadlines (soft and hard deadline) to the job. At the same time, we also hope we can find a scheduling method for those weather models which not only meet the deadline, at the same time, has a relative smaller value in the energy consumption.

Acknowledgment. The work was partly supported by the National Social Science Foundation of China (No. 16ZDA054).

References

1. Meixner, A., Redgrave, J.R., Shacham, O., et al.: Energy efficient processor core architecture for image processor. U.S. Patent Application 10/275,253, 30 April 2019
2. Ziliang, Z., Adam, M., Xiaojun, R., Xiao, Q.: EAD and PEBD: two energy-aware duplication scheduling algorithms for parallel tasks on homogeneous clusters. IEEE Trans. Comput. **60**(3), 360–374 (2011)
3. Singh, V., Gupta, I., Jana, Prasanta K.: An energy efficient algorithm for workflow scheduling in iaas cloud. J. Grid Comput. 1–20 (2019). https://doi.org/10.1007/s10723-019-09490-2
4. Hao, Y., Cao, J., Ma, T., et al.: Adaptive energy-aware scheduling method in a meteorological cloud. Future Gener. Comput. Syst. **101**, 1142–1157 (2019)

5. Liu, W., Wei, D., Chen, J., Wang, W., Zeng, G.: Adaptive energy-efficient scheduling algorithm for parallel tasks on homogeneous clusters. J. Netw. Comput. Appl. **41**, 101–113 (2014). https://doi.org/10.1016/j.jnca.2013.10.009. ISSN 1084-8045
6. Gillespie, M.: Amdahl's law, gustafson's trend, and the performance limits of parallel applications (2008). WWW-sivu: http://software.intel.com/sites/default/files/m/d/4/1/d/8/Gillespie-0053-AAD_Gustafson-Amdahl_v1__2_.rh.final.pdf
7. Li, X., Garraghan, P., Jiang, X., et al.: Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy. IEEE Trans. Parallel Distrib. Syst. **29**(6), 1317–1331 (2018)
8. Manousakis, I., Zakkak, F.S., Pratikakis, P., Nikolopoulos, D.S.: TProf, an energy profiler for task-parallel programs. Sustain. Comput. Inform. Syst. **5**, 1–13 (2015). https://doi.org/10.1016/j.suscom.2014.07.004. ISSN 2210-5379
9. Tang, Z., Qi, L., Cheng, Z., Li, K., Khan, S.U., Li, K.: An energy-efficient task scheduling algorithm in DVFS-enabled cloud environment. J. Grid Comput. **14**(1), 55–74 (2015). https://doi.org/10.1007/s10723-015-9334-y
10. Xia, Y., Zhou, M., Luo, X., Pang, S., Zhu, Q.: A stochastic approach to analysis of energy-aware DVS-enabled cloud datacenters. IEEE Trans. Syst. Man Cybern. Syst. **45**(1), 73–83 (2015)
11. Xu, X., Dou, W., Zhang, X., et al.: EnReal, an energy-aware resource allocation method for scientific workflow executions in cloud environment. IEEE Trans. Cloud Comput. **4**, 166–179 (2015)
12. Lazarescu, M.T., Cohen, A., Lavagno, L., et al.: Energy-aware parallelization toolset and flow for C code (2019)
13. Kavanagh, R., Djemame, K., Ejarque, J., et al.: Energy-aware self-adaptation for application execution on heterogeneous parallel architectures. IEEE Trans. Sustain. Comput. **5**(1), 81–94 (2019)
14. Banerjee, K.B.P.: Approximate algorithms for the partitionable independent task scheduling problem. In: Proceedings of the 1990 International Conference on Parallel Processing, vol. I, pp. 72–75 (1990)
15. Tao, M., Dong, S.: Two-tier policy-based consolidation control for workload with soft deadline constrain in virtualized data center. In: 2013 25th Chinese Control and Decision Conference (CCDC), pp. 2357–2362, 25–27 May 2013
16. Barros, C.A., Silveira, L.F.Q., Valderrama, C.A., Xavier-de-Souza, S.: Optimal processor dynamic-energy reduction for parallel workloads on heterogeneous multi-core architectures. Microprocess. Microsyst. **39**(6), 418–425 (2015). https://doi.org/10.1016/j.micpro.2015.05.009. ISSN 0141-9331
17. Jia, Z., Zhang, Y., Leung, J.Y.T., Li, K.: Bi-criteria ant colony optimization algorithm for minimizing makespan and energy consumption on parallel batch machines. Appl. Softw. Comput. **55**, 226–237 (2017). https://doi.org/10.1016/j.asoc.2017.01.044
18. Jin, X., Zhang, F., Fan, L., Song, Y., Liu, Z.: Scheduling for energy minimization on restricted parallel processors. J. Parallel Distrib. Comput. **81–82**, 36–46 (2015). https://doi.org/10.1016/j.jpdc.2015.04.001. ISSN 0743-7315