MANAGING AND SCHEDULING APPROXIMATE APPLICATIONS TO UTILIZE RENEWABLE ENERGY IN CLOUD COMPUTING DATACENTERS

ZHANG, G. J.[1] – WANG, X. Y.[1,2*] – YANG, M. Q.[1] – ZHANG, L.[3]

[1]*State Key Laboratory of Plateau Ecology and Agriculture, Department of Computer Technology and Applications, Qinghai University, Xining, China, 810016*

[2] *State key Laboratory of Hydroscience and Engineering, Tsinghua University, Beijing, China, 100084*

[3]*College of Computer Science, Sichuan University, Chengdu, China, 610064*

**Corresponding author*
e-mail: xy.wang@foxmail.com

(Received 24th Oct 2016; accepted 20th Dec 2016)

Abstract. With the rapid development of cloud computing, the power consumption of datacenters is getting much higher in recent years. Renewable energy becomes promising as the power supply for datacenters since it helps to greatly save the consumption of traditional power resources. In order to improve the utilization of renewable energy in data centers, we proposed several adaptive scheduling algorithms to manage the approximate applications in the datacenter, based on trading off their performance and accuracy. The algorithms are based on different priorities, precisions and running times of the target applications respectively. As the energy generation amount varies with time, the proposed algorithms can follow the trend of energy fluctuation to utilize the renewable energy more efficiently. Evaluation experiment results show that by employing three different kinds of scheduling algorithms, the energy utilization rate can reach 91.9318%, 91.9266% and 91.9225% respectively. It was proved that the adaptive scheduling algorithm could not only help to sufficiently utilize the renewable energy, but also guaranteed a reasonable quality of service for user.

Keywords: *energy-efficient scheduling; cloud applications; resource management; power management; application performance modeling; green datacenter; sustainable datacenter*

Introduction

Cloud computing (Zhang, 2014) is an emerging computing model, which could provide dynamic, available and convenient virtualized resources to users. Datacenters become the core infrastructure of cloud, and the computing environment develops and progresses rapidly. Cloud computing datacenters of different sizes are usually comprised of hundreds to hundreds of thousands of servers. These datacenters expose many typical characteristics, including large amount of resources, strong heterogeneity, and high energy consumption and so on. Since the cloud datacenters can provide high-quality, safe and reliable services, cloud computing becomes ever more popular in both academic and industry fields. Many papers have been list many methods that made cloud computing energy efficient and more power aware (Nathuji et al., 2007). However, as the size of datacenters gradually expand, more and more resources will be incorporated into datacenters, and thus their energy consumption becomes remarkably significant.

http://www.aloki.hu ● ISSN 1589 1623 (Print) ● ISSN 1785 0037 (Online)
DOI: http://dx.doi.org/10.15666/aeer/1503_307321

From the trends of the user demand, we can know that datacenters become a particularly promising industry, and the consumption of datacenters energy is large and rapidly growth. In 2013, all of the datacenters in the United States consumed 91 billion kWh, and it is predicted that by 2020 the number will increase to 139 billion kWh. As the energy consumption of datacenters become huge and serious, the maintenance costs will rise correspondingly. McKinsey research data show that by 2020, the data amount all over the world will reach today's 44 times. If the data amount increases by such speed, more and more datacenters have to be built, and therefore incur more energy consumption. Although datacenters can bring a lot of benefits and convenience for us, the high energy consumption of datacenters might make the system unstable and incur significant carbon footprint and environmental pollution problems. Hence, how to solve the problem of data centers energy consumption becomes a critical issue. (Wang et al, 2015).

Since datacenters contain various types of resources with characteristics of multiple attributes, in this case, if the resources are not reasonably controlled and scheduled, it will incur remarkable energy waste. Although dynamic resource scheduling strategies can help to save energy consumption, the saving amount is limited after all. Consequently, it is a new trend to use renewable energy as power supply for datacenters, which can help to greatly save the consumption of traditional power resources. In this paper, our main purpose is to appropriately utilize the renewable energy, aiming at typical approximate applications based on the trade-off of performance and accuracy to schedule and manage these tasks. By running such applications, the power consumption of computing nodes could follow the trend of energy input change as much as possible, trying to sufficiently utilize the renewable energy.

In this paper, we first established abstract modeling methods for approximate applications to describe the performance, accuracy, and the trade-off relationship between them. Then, we designed adaptive scheduling algorithms and deployed renewable energy and energy storage devices in datacenters, aiming at scheduling approximate applications, in order to meet the user demands as well as maximizing the utilization of renewable energy. Finally, we verified this algorithm via a series of experiments. Our main contributions include: 1) the potential relaxation of approximate applications which focus on using renewable energy in data centers, so that the energy consumption incurred by running these applications can better match the supply of energy; 2) the exploration of the controllability of two typical algorithms aiming at different energy consumption targets by setting different parameters; 3) the design and evaluation of three adaptive scheduling strategies, which can help provide a reasonable service for the users, by considering both the quality of service and the available energy supply amount. Experimental results demonstrate that by employing the three strategies we proposed, the renewable energy utilization rate can reach 91.9318%, 91.9266% and 91.9225% respectively. It could not only help to sufficiently utilize the renewable energy, but also guaranteed a reasonable quality of service for user at the same time.

The remainder of this paper is organized as follows. Section 2 reviews some relevant work about renewable energy utilization for datacenters. Section 3 we introduce approximate applications and two kinds of algorithm for datacenters, including Genetic Algorithm and PageRank Algorithm. Then, the algorithm models are established and energy consumption is calculated in Section 4. In Section 5, we design and implement adaptive scheduling algorithms, including the priority weighting method, high precision energy consumption weighted method and running time weighted method. Comparison results of performance evaluation experiments are given in Section 6. Finally, Section 7 gives the conclusion and discusses about possible future work.

http://www.aloki.hu ● ISSN 1589 1623 (Print) ● ISSN 1785 0037 (Online)
DOI: http://dx.doi.org/10.15666/aeer/1503_307321

Review of Literature

Building green datacenters to appropriately utilize the renewable energy has attracted a lot of attention all over the world in recent years. Researchers began to conduct studies on datacenter power consumption control from different aspects including energy saving, data centers resource management, task scheduling, load balancing and so on.

Petrucci et al. (2009) did research on virtualization cluster's dynamic resource allocation, which aimed at energy saving. They deployed energy management strategy in the platform to meet the needs of a wide variety of applications, through the process mainly achieved by startup, shutdown and migration of virtual machine. Chen and Zhu (2015) propose a real-time task scheduler structure based on rolling optimization with detailed analysis and build the task energy consumption model. They proposed a real time saving aperiodic task scheduling algorithm *EARH* and test its performance through a large number of simulation experiments. Wang and Li (2013) found that the ant colony algorithm for solving cloud computing task scheduling problem has drawbacks such as slow convergence rate and local-optimal-solution-prone, and then proposed a new cloud computing task scheduling strategy based on dynamic adaptive ant colony algorithm. Xiang and Ding (2013) proposed a virtual machine scheduling based on energy saving algorithm *PVMAP*, and tried to make use of high energy efficiency of servers by dynamic consolidation of server virtual machines, so as to minimize the number of virtual machine migration and the number of servers running at the same time. Xing (2012) started from the overall energy consumption in datacenters, using virtualization technology to integrate datacenters hardware resources.

Nevertheless, the total energy consumption of datacenters is so enormous that reducing the energy consumption with finally reach a limit after all. Carbon emission from the entire data centers will still have a greater impact on the environment, as datacenters become more and more widely used. Thus, researches began to consider expanding the new kinds of energy sources, and tried to explore renewable energy utilization methods of green data centers.

Goiri et al. (2011) proposed a GreenSlot framework to scheduling for batch processing tasks, and proposed and GreenHadoop framework (2012) for MapReduce-type tasks. They are both based on availability of renewable energy to make prediction, using different scheduling strategy to maximize the use of green energy. Bianchini (2012) focused on datacenters powered by green renewable energy, such as solar and wind power, and studied on the way how renewable energy supplied to datacenters in order to achieve the purpose of energy conservation. Goiri et al. (2014) overviewed two of their main efforts: Parasol and GreenSwitch. They have built Parasol and developed GreenSwitch for datacenters partially powered by renewable energy. Deng et al. (2011) introduced a new metric of the renewable-powered instance that can measure the concentration of renewable energy in data centers. They also conducted a simulation based on study of renewable-energy data centers, focusing on the grid tie–the device most commonly used to integrate renewable energy. They found that grid-tie placement has first-order effects on renewable-energy concentration. Aksanli et al. (2012) designed an adaptive data center job scheduler which utilizes short term prediction of solar and wind energy production. Sharma et al. (2011) focused on managing server clusters running on intermittent power, and proposed *blinking* as the primary abstraction for handling intermittent power constraints, and defined multiple types of blinking policies. Goiri et al. (2011) also proposed a series of green-energy-aware scheduling algorithms, mixing grid and green energy to supply power to datacenters.

http://www.aloki.hu ● ISSN 1589 1623 (Print) ● ISSN 1785 0037 (Online)
DOI: http://dx.doi.org/10.15666/aeer/1503_307321

As can be seen, although the energy consumption of datacenters has been concerned and explored in the cloud computing environment comprehensively, the research on the renewable energy utilization is not very mature. Preliminary research results imply that there are still many key issues to be studied and resolved. In recent years, the impact of large datacenters on the atmosphere and the environment becomes more and more prominent, which brings plenty of challenges for future studies in this field.

Cases and Tools for Analyzing Approximate Applications

Approximate applications refer to kinds of application frameworks which could trade accuracy for performance, energy, power or other benefits. By adjusting some controllable parameters, we can make a trade-off between the performance (or quality of service) and energy consumption. In this paper, we take two typical algorithms for example, including Genetic Algorithm (2003) and PageRank algorithm (2010), which both contain controllable parameters and it is supported to adjusting the energy consumption by changing these parameters.

Application cases

Genetic algorithm (GA)

Genetic Algorithm is one kind of evolutionary algorithms, whose basic principle is to emulate the biosphere in the "natural selection, survival of the fittest," the evolution of the law. It is a heuristic search algorithm to solve the optimization problem in the field of artificial intelligence. This heuristic is often used to generate useful solution to optimize and search problem. The algorithm was originally developed by learning from some phenomena in evolutionary biology, including genetics, mutation, natural selection, hybridization and so on.

Genetic Algorithm is one of the typical algorithms, and how to select controllable parameter is the key point of this paper. To determine the controllable parameters of the genetic algorithm, we generate the matching degree between the gene sequence and the desired gene sequence, in the range between 0-64. The higher the matching degree of gene sequences is, the longer operation time of the genetic algorithm will be, and vice versa. Through testing we find that the matching degree between the generated gene sequence and the expected gene sequence could be employed as the controllable parameter to adjust the accuracy of the program. Also, adjusting the controllable parameters also has direct impact on the value of the energy consumption.

PagePank algorithm (PR)

PageRank algorithm was proposed by and named after Larry Page, one of the Google founders. It is a link analysis algorithm used by Google Search to rank websites in their search engine results. PageRank can be used to determine the level of a page or the importance of a web page, which provides a way to measure the importance of website pages. PageRank measure the value of site according to the number and quality of the external links and internal links of the site.

Here, we use the maximum error as the control parameter of the PageRank algorithm, which can help to distinguish the parameters of the PR value gap between web pages. The precision of the maximum error is controlled between the range of

http://www.aloki.hu ● ISSN 1589 1623 (Print) ● ISSN 1785 0037 (Online)
DOI: http://dx.doi.org/10.15666/aeer/1503_307321

0.00000000001~0.001. It can be inferred that the greater the error value, the faster the program will run and vice versa.

In our experiments, we used 784 pages as test data, which are linked to each other by a link matrix.

Performance analysis

To analyze the performance, we use *JConsole* to measure CPU utilization ratio， *JConsole* is a built-in Java performance analyzer, which can be used to monitor the performance of Java applications and trace codes in Java (Swirydczuk, 2015; Brambilla et al., 2016). *JConsole* is a GUI based Java Management Extensions (JMX) tool, which uses the extensive instrumentation of the Java Virtual Machine (Java VM) to provide information about the performance and resource consumption of applications running on the Java platform.

Pre-processing and Performance Modeling

Calculation of energy consumption

By monitoring during operating procedures, we can get the average CPU utilization ratio ω. Put ω into the energy efficiency model (Song et al., 2012), the CPU power consumption P can be computed as:

$$P(f^3, \omega) = A f^3 + B \omega f^3 + C \omega + D \quad \text{(Eq.1)}$$

Where f is the CPU frequency (e.g. 2.8GHz), A, *B*, *C* and *D* are coefficients. The CPU utilization ratio ω and the power *P* were measured in a single machine environment. It can be seen that, when f is fixed, we can verify the formula (1), that ω and p have a linear relationship. Then we can also verify the linear relationship between P and f^3 when ω is fixed. By conducting a fitting procedure using the measured data, we can determine the values of *A*, *B*, *C*, *D*, and get: A=0.15, B=1.4, C=15.2, D=50.8. Hence, the energy consumption during a time period t can be computed as:

$$W = P \cdot t \quad \text{(Eq.2)}$$

Running time estimation of different algorithms

As mentioned above, both of the two algorithms could be controlled to run faster or slower by changing the value of the controllable parameter. Furthermore, since the energy consumption is proportional to the running time of the program, we can also try to control the application execution mode under different energy constraints. In this paper we set 8 predefined modes for each kind of the algorithm by running through a series of experiments repeatedly. Different modes correspond to particular sets of controllable parameters. The setting values of each mode are averaged out of multiple tests by removing the maximum and minimum values. Running the program under different modes will lead to different execution time and thus the energy consumption results will be different.

http://www.aloki.hu ● ISSN 1589 1623 (Print) ● ISSN 1785 0037 (Online)
DOI: http://dx.doi.org/10.15666/aeer/1503_307321

Genetic algorithm (ga) and pagerank algorithm (pr) model

Table 1 gives the running time of Genetic Algorithm under different modes averaged through multiple repeated tests, from which it can be seen that the running time increases from Mode 0 to Mode 7. As mentioned, the controllable parameter here is the matching degree of the gene sequences. Intuitively, as the matching degree becomes lower, the running time of the program becomes faster.

Table 2 gives the running time of the PageRank algorithm under different parameters averaged through multiple repeated tests. The running time is getting longer and longer from Mode 0 to Mode 7. Here, the controllable parameter is the maximum allowed error.

Consequently, by adjusting particular parameters of the approximate application, the running time and thus the energy consumption could be controlled among certain ranges. This provide the basis to establish adaptive scheduling strategies towards energy utilization objectives.

Table 1. *Running time of Genetic Algorithm under different parameters*

Mode	Running time (s)	Mode	Running time (s)
Mode 0	205.11	Mode 4	97.94
Mode 1	167.27	Mode 5	68.62
Mode 2	144.27	Mode 6	54.86
Mode 3	121.00	Mode 7	39.19

Table 2. *Running time of PageRank algorithm under different parameters*

Mode	Running time (s)	Mode	Running time (s)
Mode 0	163.91	Mode 4	111.36
Mode 1	154.66	Mode 5	100.17
Mode 2	143.80	Mode 6	88.43
Mode 3	122.26	Mode 7	77.39

Energy consumption estimation of individual applications

We know that through the *JConsole* we can monitor the CPU utilization ratio, and then the power consumption of the program could be calculated. According to the running time value of *Table 1*, ultimately the energy consumption under different mode could be obtained. *Tables 3-4* give respectively the energy consumption of the Genetic Algorithm and the PageRank algorithm under different modes.

By collecting the CPU utilization ratio, we can calculate the power consumption of the running programs. Then, we can calculate the energy consumption of the program by Equation (2). From *Tables 3-4* it can be observed that the running time of the program is one of the main factors affecting the energy consumption proportionally. Thus, the energy consumption values of individual applications could be obtained from the tables. This provide the basis of further estimation for multiple simultaneously running applications.

http://www.aloki.hu ● ISSN 1589 1623 (Print) ● ISSN 1785 0037 (Online)
DOI: http://dx.doi.org/10.15666/aeer/1503_307321

Table 3. *Energy consumption of Genetic Algorithm under different parameters*

Mode	Energy Consumption (J)	Mode	Energy Consumption (J)
Mode 0	13448.57	Mode 4	6421.93
Mode 1	10968.93	Mode 5	4500.42
Mode 2	9459.02	Mode 6	3597.52
Mode 3	7934.31	Mode 7	2570.17

Table 4. *Energy consumption of the PageRank algorithm under different parameters*

Mode	Energy Consumption (J)	Mode	Energy Consumption (J)
Mode 0	10744.50	Mode 4	7301.87
Mode 1	10140.30	Mode 5	6568.06
Mode 2	9428.87	Mode 6	5797.79
Mode 3	8016.67	Mode 7	5073.99

Estimation of concurrently running applications

Through experiments on realistic testbeds, we found that the total energy consumption of multiple running applications was not simply the summary of the consumption of each individual application. Hence, we have to find the relationship between the overall energy consumption and the number of concurrently running applications. Here, we can get the actual measured value of concurrent programs, and the sum of single application energy consumption values as well. Then, we calculate the difference between the two values, referred as "energy offset" hereafter. In order to estimate the energy consumption of concurrently running programs, we establish a regression model between the number of concurrent programs and the energy consumption difference mentioned above. Specifically, we conducted random tests by running 2~5 applications simultaneously, repeated these tests to get the average value, and obtained the fitting relation between the number of concurrent applications and the energy offset value, as shown in *Fig. 1*.

As shown, Equation (3) is a linear regression model. As long as we know the number of concurrent programs, and the energy offset value can be obtained by this model, and then we can get the actual energy consumption of concurrent programs.

$$y=3992x^{2}-18250x+24825 \quad \text{(Eq.3)}$$

Adaptive Green-energy-aware Scheduling Algorithms

Algorithm framework

The main principle of designing adaptive green-energy aware scheduling algorithms is to reasonably distribute the previously stored energy in order to make full use of

http://www.aloki.hu ● ISSN 1589 1623 (Print) ● ISSN 1785 0037 (Online)
DOI: http://dx.doi.org/10.15666/aeer/1503_307321

renewable energy while meeting the user demands. First, according to the list of user requirements the number of concurrent running applications in time interval T could be determined. Note that the number of applications might change in different time interval as the amount of energy storage varies. Then, it should be determined how much energy will be allocated to each running application. Here we design three different strategies from different point of view, called priority weighting (*PW*), energy weighting (*EW*), and running time weighting (*RTW*) respectively. At last, the mode of each application will be selected according to the amount of energy allocated, and the mode closest to the distribution model will be chosen. The flow diagram of the scheduling algorithms is shown in *Fig. 2.*

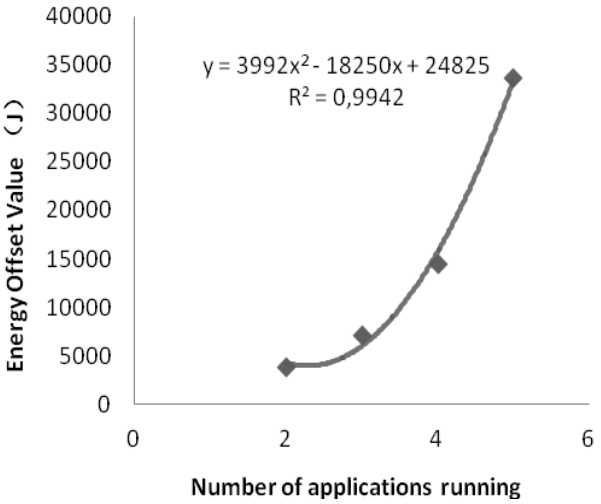

Figure 1. *Relationship between the number of applications and the energy offset value*

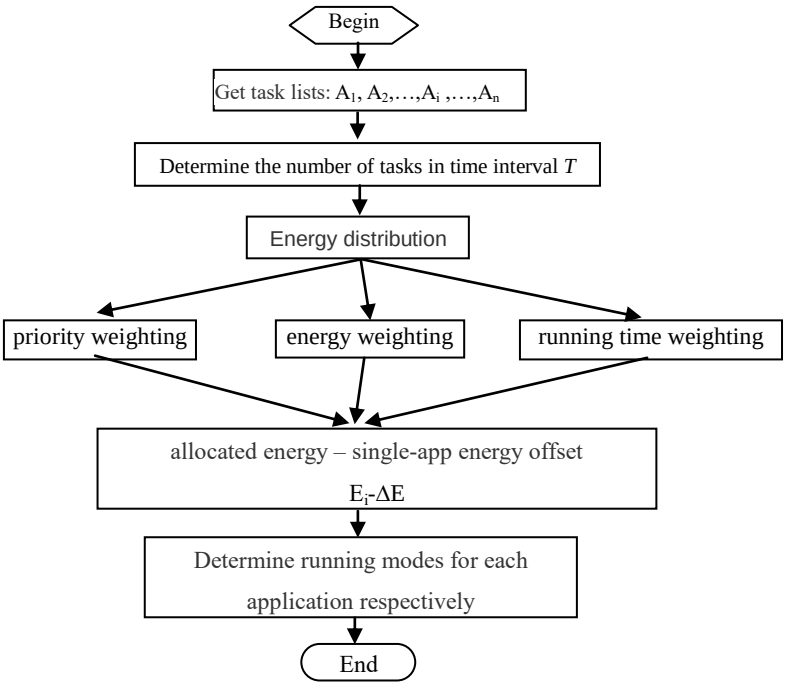

Figure 2. *Flow Diagram of the scheduling algorithm*

http://www.aloki.hu ● ISSN 1589 1623 (Print) ● ISSN 1785 0037 (Online)
DOI: http://dx.doi.org/10.15666/aeer/1503_307321

Determine the number of concurrent applications

Here, the submitted task list is comprised of n applications, denoted as A_1, A_2, A_3, ... A_{i-1}, A_i, ..., A_n. These applications could be derived from the same program or may be all different. Each application could have different running mode, denoted as B_1^1, B_2^1,... , , B_1^2, B_2^2,..., ,..., B_1^n,... , , where j_i represents the number of modes of the ith application. The following procedure is conducted to determine the number of concurrent applications:

- Determine the medium-accuracy mode for each application. First, different modes will be sorted in accordance with the energy consumption values from large to small. If the number of j_i is odd, then take the one with medium energy consumption value as the medium-accuracy mode; if the number of modes is even, then take the one with larger value as the medium-accuracy mode, out of the two medium candidates.
- Determine the number of concurrent applications. Denote the current energy storage as E, and the energy consumption values of the medium-accuracy modes for the concurrent applications are m_1, m_2, m_3,..., m_k, respectively. If $k<=n$, then a part of the tasks from the requirements (i.e. the first k tasks by order) will be running concurrently; if $k>n$, then the tasks will be put into running in a looped way according to the list order. Here, k is the maximum integer which satisfies the following equation:

$$m_1+m_2+ m_3+... +m_k=E \quad \text{(Eq.4)}$$

Determine the accuracy mode for each application

In order to make full use of energy, it is needed to allocate the energy as much as possible to currently running applications. Denote the available energy as E, and the number of concurrent applications as k. We used three different kinds of strategies to determine the weight of each application. Then, the weight will be adopted to calculate the amount of energy allocated to the corresponding application. Finally, according to this energy value, we can find the closest model by searching the energy consumption table of different running modes for the specific application.

Here, we describe in detail the three different strategies which can be used to determine the weight for each application, as follows.

Priority weighting (PW)

We assume that different applications have different priority indicated by their order in the task list. The earlier it appears, the priority of the application is higher, and thus the weight should also be higher. If the number of concurrent applications is k, and according to the order in the task list, the corresponding weight of the ith application can be calculated as

$$\eta_i = \frac{k-i+1}{1+2+3+\cdots+k}, 1 \le i \le k \quad \text{(Eq.5)}$$

http://www.aloki.hu ● ISSN 1589 1623 (Print) ● ISSN 1785 0037 (Online)
DOI: http://dx.doi.org/10.15666/aeer/1503_307321

Energy Weighting (EW)

The basic idea of the *EW* strategy is attempting the increase the accuracy of the applications to higher levels. It uses the high-accuracy modes to compute the weight values for the applications. Corresponding to the order of the task list, denote the energy consumption values of high-accuracy modes as w_1, w_2, w_3,... ,w_k, and then the weight of the ith application can be calculated as:

$$\eta_i = \frac{w_i}{w_1 + w_2 + w_3 + \cdots + w_k}, 1 \le i \le k \quad \text{(Eq.6)}$$

Running Time Weighting (RTW)

The motivation of *RTW* strategy is to perform more tasks during a certain time period. The running time of the applications is used to calculate their weight, which makes the shortest task get higher weight value, aiming at speeding up the execution process and increasing the throughput. Corresponding to the order of the task list, denote running time of high-accuracy modes for the candidate applications as t_1, t_2, t_3,... ,t_k, and the weight the first ith program can be calculated as:

$$\eta_i = \frac{\frac{1}{t}}{\frac{1}{t_1} + \frac{1}{t_2} + \frac{1}{t_3} + \cdots \frac{1}{t_k}}, 1 \le i \le k \quad \text{(Eq.7)}$$

Performance Evaluation

Datacenter energy supply

On the basis of our previous work (Yang, 2014), we use the solar energy as the power supply of the datacenter, which has very strong nonlinear characteristics. In order to forecast the solar power generation, we explore the influence factors of solar radiation intensity and the relationship among solar radiation intensity and ambient temperature, time, humidity, wind speed based on a realistic testbed. We have established four prediction models to estimate the solar power generation values in the near future, including the nearest neighbor model, the highest frequency model, order-based weight model and distance-based weight model. The data were measured and collected on the top of the Qinghai University Engineering Laboratory Building, with the longitude of 101°45'1.17"E and the latitude of 36°43'35.62"N. While the absolute percentage error is limited to less than 20%, the average accuracy of our model can reach 88%~98%.

Here, in the following experiments, we will take the solar energy during 24 hours in a whole day as the power supply. We only consider the time period from 7:31 to 18:31, since in other time interval there was no solar energy generated. The length of the time interval is set to 10 minutes, and excessive energy which is not sufficiently utilized will be wasted as time elapsed. Denote the solar radiation value at time t as e_t (w/m^2), then the generated energy during time interval with length T can be calculated as:

$$W_c = e_t \cdot s \cdot T \quad \text{(Eq.8)}$$

where s is the area of the solar panel. T is set to 10 minutes in the following experiments.

http://www.aloki.hu ● ISSN 1589 1623 (Print) ● ISSN 1785 0037 (Online)
DOI: http://dx.doi.org/10.15666/aeer/1503_307321

Experiment results

In our simulation experiments, the incoming demand sequence is {p g p p g g g p g p p g}, in which p represents the PageRank algorithm and g refers to the Genetic Algorithm.

This paper designs three different strategies from different points of view, called priority weighting (*PW*), energy weighting (*EW*), and running time weighting (*RTW*) respectively. The three algorithms consider from three aspects respectively, such as the priority of the programs, the energy distribution and the running time, aiming at making full use of renewable energy while meeting the user demands. In order to verify the efficiency of three algorithms, we compared these three adaptive scheduling algorithms with two basic strategies, including the high precision strategy (HP) and the low precision strategy (LP). The HP strategy only considers the highest service quality, with the principle of all tasks executing at the highest precision mode. On the contrary, the LP strategy only considers the fastest service, with the principle of all tasks executing at the lowest precision mode. *Fig. 3* shows the energy consumption of the above five strategies together with the energy supply curve in daytime.

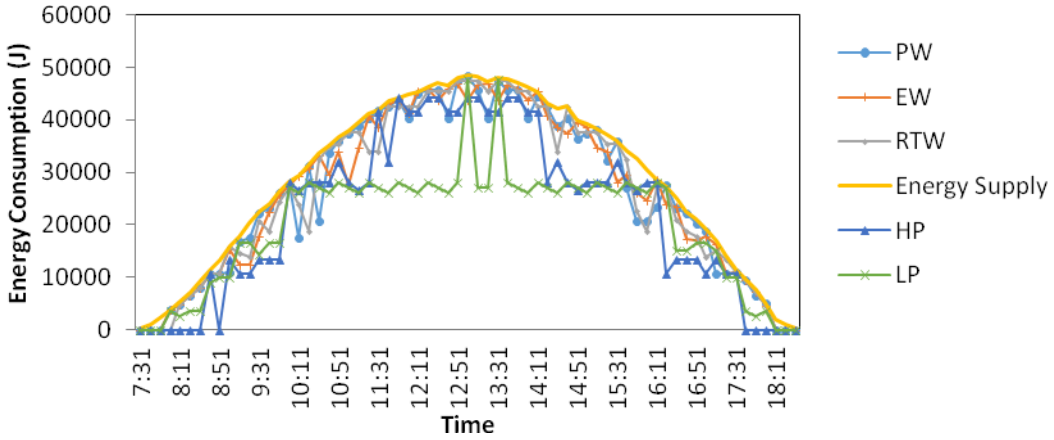

Figure 3. *Energy consumption comparison of five strategies*

From *Fig. 3* it can be observed that the three kinds of adaptive strategies can help to increase the utilization of the supplied renewable energy. *Fig. 3* only consider the time period from 7:31 to 18:31, during which the length of the time interval is set to 10 minutes. From that data statistics, we can find that using the adaptive strategies, the energy utilization ratio could approach more than 90%, while the two basic strategies can only lead to utilization less than 80%. Note that the *LP* strategy behaves the worst, since it only considers the execution speed without considering the quality of service. The *HP* strategy utilizes the renewable energy better than the *LP* strategy.

To make it clearer, *Fig. 4* shows the overall utilization of the energy supply when using the five different strategies. It can be seen from *Fig. 4* that the energy utilization ratio of the three adaptive strategies are obviously higher than the other two strategies.

To further investigate the behavior of different algorithms, we recorded the number of finished tasks by using different strategies, as shown in *Fig. 5*. It can be seen that *HP* finished least tasks, since it guaranteed high accuracy first. *LP* finished more than 200 tasks in total and the number is more than all of the other strategies, but it wastes some of energy supply even when there is extra energy able to process the tasks in higher

http://www.aloki.hu ● ISSN 1589 1623 (Print) ● ISSN 1785 0037 (Online)
DOI: http://dx.doi.org/10.15666/aeer/1503_307321

quality and provide better services. In comparison, the *RTW* strategy can finished 189 tasks which is 69.6776% of the tasks finished by *LP* while keeping the renewable energy utilized much more sufficiently.

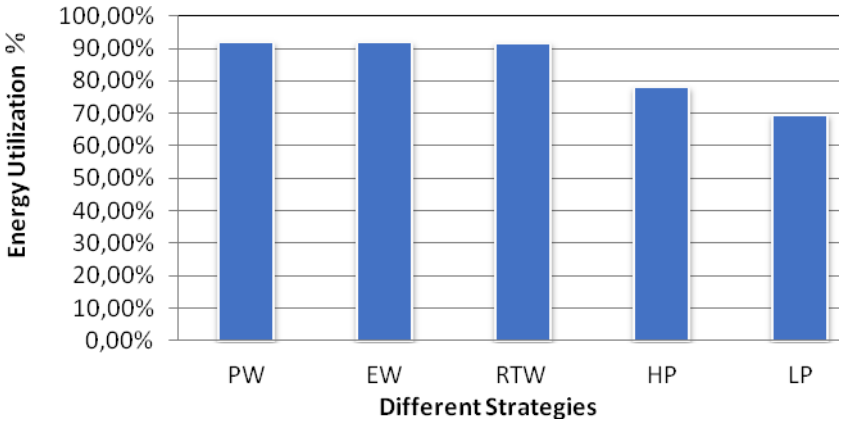

Figure 4. *Energy utilization comparison*

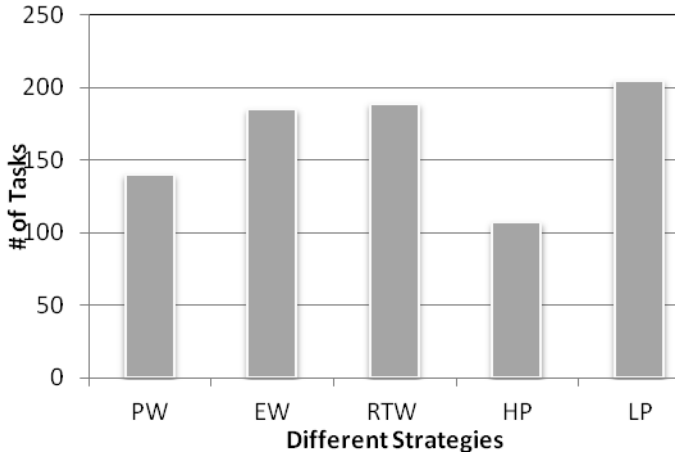

Figure 5. *Number of finished tasks*

To gain deeper insight, *Fig. 6* shows the number of finished tasks in each time interval during the experiment time. It can be observed that during the period of 10:11-15:31, most tasks were completed, due to the sufficient solar energy power generated around noon. We can also see that *RTW* processed more tasks than *EW* and *PW* while following the energy supply curve quite well.

In summary, the three adaptive strategies can effectively follow the variation of solar energy supply, and the utilization of renewable energy could be much higher than the other two basic strategies. In comparison, *HP* only considers the service quality but lead to lowest throughput. On the contrary, although *LP* won the throughput contest for finishing 189 tasks, the adaptive strategy can also complete 189 tasks while increasing the solar energy utilization ratio more than 20%. This leads to more balanced results when considering both the performance and accuracy together, which processes as many tasks as possible while utilizing the supplied energy efficiently.

http://www.aloki.hu ● ISSN 1589 1623 (Print) ● ISSN 1785 0037 (Online)
DOI: http://dx.doi.org/10.15666/aeer/1503_307321

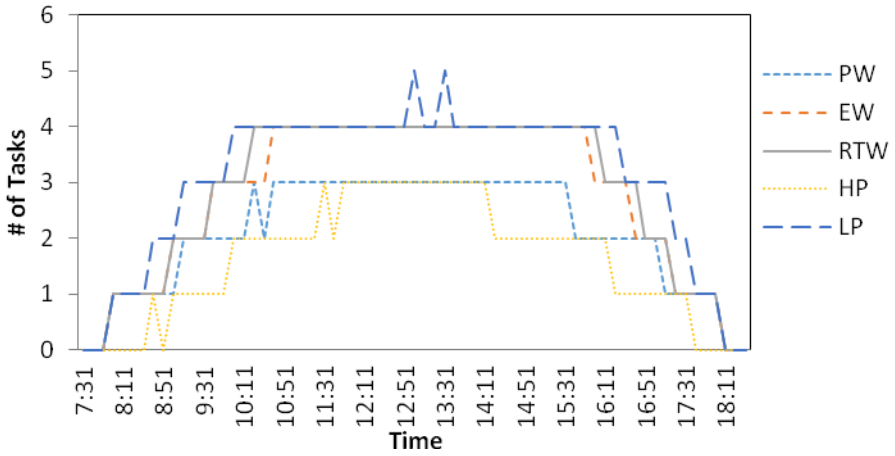

Figure 6. *The number of finished tasks in each time interval*

Conclusion and Future Work

With the wide application of renewable energy as the power supply in datacenters, it becomes an important issue that how to efficiently utilize the supplied renewable energy. In this paper, we focus on approximate applications, a typical kink of workload deployed in the datacenter, and choose *GA* and *PageRank* as two examples to establish the characterization models. We proposed several adaptive scheduling strategies that can manage the corresponding applications, in order and let supply of application able to better meet the supply of energy while keeping the quality of the application service. Three adaptive scheduling algorithms were presented, including priority weighting (*PW*), energy weighting (*EW*), and running time weighting (*RTW*) (Zhou et al., 2015). By simulating the performance modeling and the designed adaptive scheduling algorithm. Evaluation experiment results illustrate that by employing three different kinds of scheduling algorithms, the energy utilization ratio can reach 91.9318%, 91.9266% and 91.9225% respectively. This shows that the adaptive scheduling algorithm could help to sufficiently utilize the renewable energy.

This paper only demonstrates the possibility of controlling approximate applications towards energy consumption targets by using two examples. As future work, we intend to make more effort on the generality of the energy management approaches. On the other hand, more types of applications in reality could be further incorporated into our framework.

Acknowledgements. This paper is partially supported by The National Natural Science Foundation of China (No.61363019, No.61563044, and No.61640206); Open Research Fund Program of State Key Laboratory of Hydroscience and Engineering (sklhse-2017-A-05); National Natural Science Foundation of Qinghai Province (No. 2014-ZJ-718, No. 2015-ZJ-725).

REFERENCES

[1] Aksanli, B., Venkatesh, J., Zhang, L., Rosing, T. (2012): Utilizing green energy prediction to schedule mixed batch and service jobs in data centers. - ACM SIGOPS Operating Systems Review 45(3): 53-57.

http://www.aloki.hu ● ISSN 1589 1623 (Print) ● ISSN 1785 0037 (Online)
DOI: http://dx.doi.org/10.15666/aeer/1503_307321

[2] Bianchini, R. (2012): Leveraging renewable energy in data centers: present and future. - Keynote at Green Metrics: 135-136.
[3] Brambilla, W., Van Rooijen, A., Simeone, S., Ibba, A., DeMuro, S. (2016): Field observations, video monitoring and numerical modeling at poetto beach, Italy. - Journal of Coastal Research SI 75: 825-1368.
[4] Chen, C., Zhu, X. M. (2015): Energy-efficient scheduling for real-time tasks by rolling-horizon optimization in virtualized clouds. - Journal of Software 26(8): 2111-2123. (In Chinese)
[5] Deng, N., Stewart, C., Li, J. (2011): Concentrating renewable energy in grid-tied datacenters. - IEEE International Symposium on Sustainable Systems and Technology (ISSST) 19(5): 1-6.
[6] Goiri, I., Le, K., Haque, M. E., Beauchea, R., Nguyen, T. D., & Guitart, J., (2011): GreenSlot: Scheduling energy consumption in green datacenters. - In Proc. of SC' (11): 20:1-20:11.
[7] Goiri I., et al. (2012): GreenHadoop: Leveraging green energy in data-processing frameworks. - In Proc. Of EuroSys' (12): 57-70.
[8] Goiri, Í., Katsak, W., Le, K., Nguyen, T. D., Bianchini, R. (2013): Parasol and GreenSwitch: managing datacenters powered by renewable energy. - In ACM SIGARCH Computer Architecture News 41(1): 51-64.
[9] Goiri, I., Katsak, W. A., Le, K., Nguyen, T. D., Bianchini, R. (2014): Designing and managing data centers powered by renewable energy. - IEEE Micro 34(3): 8-16.
[10] Kusic, D., Kephart. J. O., et al. (2009): Power and performance management of virtualized computing environments via look ahead control. - Cluster Computing 12(1): 1-15.
[11] Mitchell, T. M. (2003): Machine Learning. - McGraw-Hill Education.
[12] Nathuji, R., Schwan, K. et al. (2007): Coordinated power management in virtualized enterprise systems. - ACM SIGOPS Operating Systems Review 41(6): 265-278.
[13] Petrucci, V., Loques, O., Niteroi, B., Mossé, D. (2009): Dynamic configuration support for power-aware virtualized server clusters. - In WiP Session of the 21th Euromicro Conference on Real-Time Systems. Dublin, Ireland.
[14] Raghavendra, R., Ranganathan, P. et al. (2008): Coordinated multi-level power management for the data center. - SIGARCH Computer Architecture News 36(1): 48-59.
[15] Sharma, N., Barker, S., Irwin, D., Shenoy, P. (2011): Blink: managing server clusters on intermittent power. - In ACM SIGPLAN Notices 46(3): 185-198.
[16] Song, J., Li, T. T., Yan, Z. X., Na, J., Zhu, Z. L. (2012): Energy-efficiency model and measuring approach for cloud computing. - Journal of Software 23(2): 200-214. (In Chinese)
[17] Swirydczuk, J. (2015): Clocking in turbines: remarks on physical nature and geometric requirements. - Polish Maritime Research 22(2): 62-70.
[18] Wang, F., Li, M. A. (2013): Cloud computing task scheduling based on dynamically adaptive ant colony algorithm. - Journal of Computer Applications 33(11): 3161-3169. (In Chinese)
[19] Wang, Y., Liu, Z., Liao, H., Li, C. (2015): Improving the performance of GIS polygon overlay computation with MapReduce for spatial big data processing. - Cluster Computing 18(2): 507-516.
[20] Xiang, J., Ding, E. J. (2013): Energy-saving optimization in datacenter based on virtual machine scheduling. - Journal of Computer Applications 33(12): 3331-3334. (In Chinese)
[21] Xing, W. C. (2012): Green Data Center Management Framework and Energy Consumption Monitoring System. - Fudan University, Shanghai. (In Chinese)
[22] Yang, M. Q., Wang, X. Y. (2014): Hourly solar radiation forecast based on k-nn nonparametric regression model. Applied Mechanics and Materials 571: 217-222.

http://www.aloki.hu ● ISSN 1589 1623 (Print) ● ISSN 1785 0037 (Online)
DOI: http://dx.doi.org/10.15666/aeer/1503_307321

[23] Yen, C. C. (2010): Pagerank algorithm improvement by page relevance measurement. - Journal of Convergence Information Technology 5(8): 502-506.
[24] Zhang, H. (2014): Researches on virtual machine migrating algorithm to save energy in Cloud Datacenter. - Henan University of Science and Technology (In Chinese).
[25] Zhou, W., Zhang, L., Fan, C., Zhao, J., Gao, Y. (2015): Development of a real-time force-controlled compliant polishing tool system with online tuning neural proportional–integral–derivative controller. - Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 229(5): 440-454.

http://www.aloki.hu ● ISSN 1589 1623 (Print) ● ISSN 1785 0037 (Online)
DOI: http://dx.doi.org/10.15666/aeer/1503_307321