We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Green-Aware Virtual Machine Migration Strategy in Sustainable Cloud Computing Environments

Xiaoying Wang, Guojing Zhang, Mengqin Yang and Lei Zhang

Additional information is available at the end of the chapter

Abstract

As cloud computing develops rapidly, the energy consumption of large-scale datacenters becomes unneglectable, and thus renewable energy is considered as the extra supply for building sustainable cloud infrastructures. In this chapter, we present a green-aware virtual machine (VM) migration strategy in such datacenters powered by sustainable energy sources, considering the power consumption of both IT functional devices and cooling devices. We define an overall optimization problem from an energy-aware point of view and try to solve it using statistical searching approaches. The purpose is to utilize green energy sufficiently while guaranteeing the performance of applications hosted by the datacenter. Evaluation experiments are conducted under realistic workload traces and solar energy generation data in order to validate the feasibility. Results show that the green energy utilization increases remarkably, and more overall revenues could be achieved.

Keywords: virtual machine migration, resource management, power management, renewable energy aware

1. Introduction

Large-scale datacenters, as the key infrastructure of cloud environments, usually own massive computing and storage resources in order to provide online services for thousands of millions of customers simultaneously. This leads to significant energy consumption, and thus high carbon footprint will be produced. Recent reports estimate that the emissions brought by information and computing technologies grow from 2% in 2010 [1] to 8% in 2016 and will grow to 13% by 2027 [2]. Hence, considering the heavy emissions and increasing impact on

climate change, governments, organizations, and also IT enterprises are trying to find cleaner ways to manage the datacenters, for example, exploiting renewable energy such as wind, solar, and tidal.

However, the intermittency and the instability of the renewable energy sources make it difficult to efficiently utilize them. Fortunately, we know that the datacenter workloads are usually variable, which give us opportunities to find ways to manage the resources and power together inside the datacenters to utilize renewable energy sources more efficiently. On the other hand, to provide guaranteed services for third-party applications, the datacenter is responsible of keeping the quality of service (QoS) at a certain level, subject to the service level agreements (SLAs) [3].

In modern datacenters, applications are often deployed in virtual machines (VMs). By virtualization mechanisms, VMs are flexible and easy to migrate across different servers in the datacenter. In this chapter, we attempt to conduct research on energy-aware virtual machine migration methods for power and resource management in hybrid energy-powered datacenters. Especially, we also employ thermal-aware ideas when designing VM migration approaches. The holistic framework is described, then the model is established, and heuristic and stochastic strategies are presented in detail. Experimental results show the effectivity and feasibility of the proposed strategies. We hope that this chapter would be helpful for researchers to study the features of VM workloads in the datacenter and find ways to utilize more green energy than traditional brown energy.

The remainder of this chapter is organized as follows. Section 2 introduces some relevant prior work in the field of energy-aware and thermal-aware resource and power management. Section 3 presents the entire system architecture we discuss in this chapter. Section 4 formulates the optimization problem corresponding to the issue we need to address. Section 5 describes the methods and strategies we designed to solve the problem. Section 6 illustrates the experimental results by comparing three different strategies, and finally conclusion is given out in Section 7, in which we also discuss about some of the possible future work.

2. Literature review

This section reviews the literature in the area of energy-aware resource management, thermal-aware power management, and green energy utilization in datacenters.

In the recent decade, many researchers started to focus on power-aware management methods to manage workload fluctuation and search trade-off between performance and power consumption. Sharma et al. [4] have developed adaptive algorithms using a feedback loop that regulates CPU frequency and voltage levels in order to minimize the power consumption. Tanelli et al. [5] controlled CPUs by dynamic voltage scaling techniques in Web servers, aiming at decreasing their power consumption. Berl et al. [6] reviewed the current best practice and progress of the energy efficient technology and summarized the remaining key challenges in the future. Urgaonkar et al. [7] employed queuing theory to make decision aiming at optimizing the application throughput and minimizing the overall energy costs. The above work attempts

to reduce the power consumption while guaranteeing the system performance. On the basis of such ideas, we incorporate the usage of renewable energy into the optimization model, which might support performance improvement when the green energy is sufficient enough.

Besides, thermal-aware resource management approaches also attracted some interest of researchers recently. For example, Mukherjee et al. [8] developed two kinds of temperature-aware algorithms to minimize the maximum temperature in order to avoid hot spots. Tang et al. [9] proposed *XInt* which can schedule tasks to minimize the inlet temperatures and also to reduce the cooling energy costs. Pakbaznia et al. [10] combined chassis consolidation and efficient cooling together to save the power consumption while keeping the maximum temperature under a controlled level. Wang et al. [11] designed two kinds of thermal-aware algorithms aiming at lowering the temperatures and minimizing the cooling system power consumption. Islam et al. [12] proposed DREAM which can manage the resources to control allocate capacity to servers and distribute load considering temperature situations. Similarly, we consider the impact of temperature on two kinds of cooling devices in this chapter, which directly decide the cooling power consumption.

As renewable energy becomes more widely used in datacenters, corresponding research starts to put insights into green energy–oriented approaches for managing the resources and power. Deng et al. [13] treated carbon-heavy energy as a primary cost and designed some mechanisms to allocate resources on demand. Goiri et al. designed *GreenSlot* [14] aiming at scheduling batch workloads and *GreenHadoop* [15] which could deal with MapReduce-based tasks. Both of them tried to efficiently utilize green energy to improve the application performance. Li et al. [16] proposed *iSwitch*, which can switch the power supply between wind power and utility grid according to the renewable power variation. Arlitt et al. [17] defined the "Net-Zero energy" datacenter, which needs on-site renewable generators to offset the usage of power coming from the electricity grid. Deng et al. also conducted research on Datacenter Power Supply System (DPSS) and proposed an efficient, online control algorithm SmartDPSS [18] helping to make online decisions in order to fully leverage the available renewable energy and varying electricity prices from the grid markets, for minimum operational cost. Zhenhua et al. [19] presented a holistic approach that integrates renewable energy supply, dynamic pricing, cooling supply, and workload planning to improve the overall attainability of the datacenter.

Upon the basic concepts of these work, we exploit the possibility and of efficient VM migration management toward sufficiently utilizing renewable energy supply, incorporating the flexibility of transactional workloads, cooling power consumption, and the amount of available green energy.

3. Datacenter architecture

This section describes the datacenter architecture, including the hybrid power supply and virtualization infrastructure.

Figure 1 shows the system architecture of the sustainable datacenter powered by both renewable energy and traditional energy supplies. The grid utility and renewable energy are

combined together by the automatic transfer switch (ATS) in order to provide power supply for the datacenter. Both functional devices and cooling devices have to consume power, as shown in the bottom part of the figure.
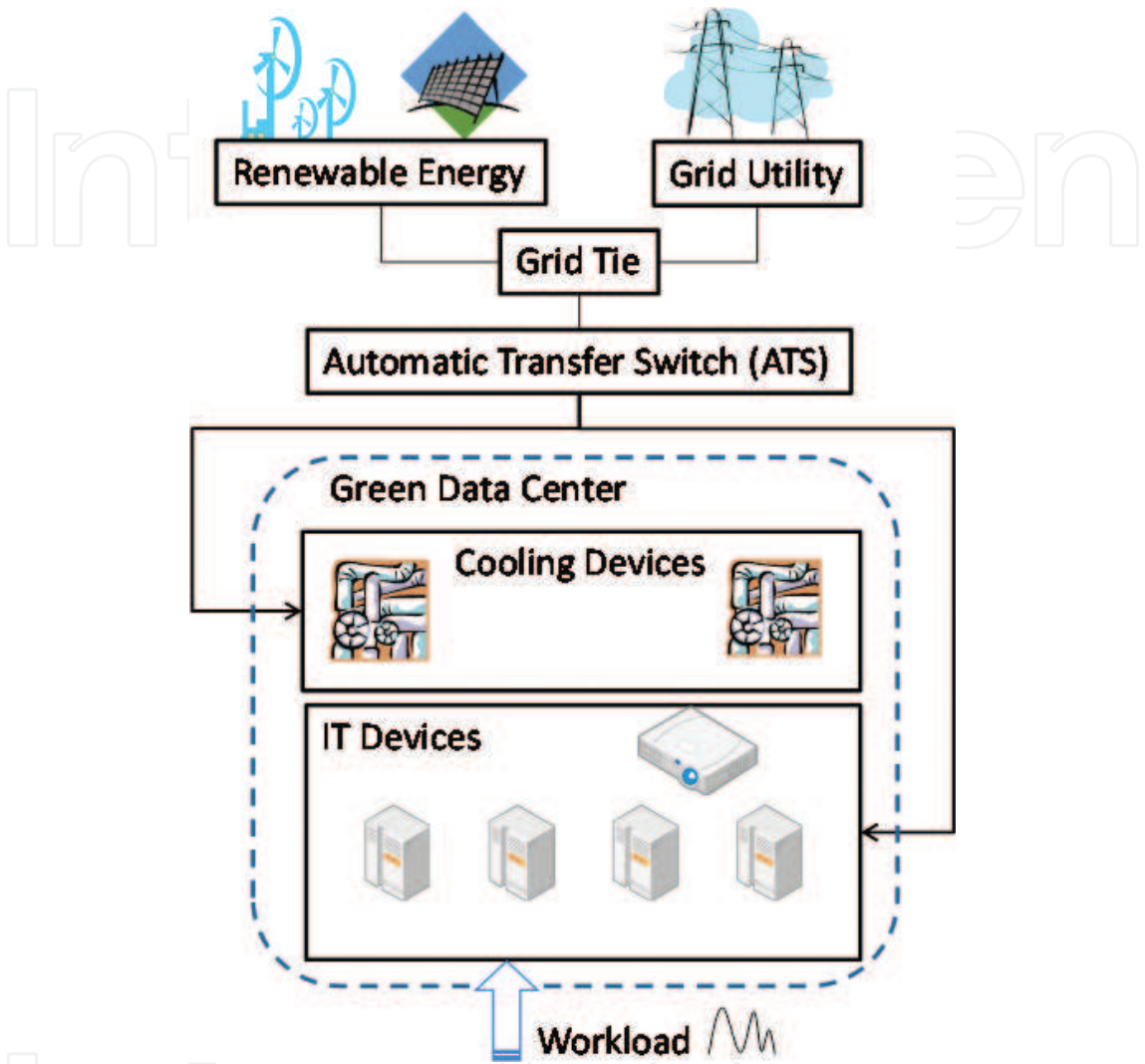

Figure 1. Architecture of the sustainable datacenter powered by both green energy and brown energy.

Figure 2 illustrates the infrastructure of virtualized cloud datacenter. As shown, the underlying infrastructure of the datacenter is comprised of many physical machines (PMs), which are placed onto groups of racks. The utility grid bus and the renewable energy bus are connected together to supply power for the datacenter devices. Renewable sources will be used first, and the grid power will be leveraged as the supplementary energy supply.

As mentioned before, virtual machines (VMs) are running on the underlying infrastructure as used to host multiple applications, as shown in the virtualization layer in **Figure 2**. Different VMs on the same PM might serve for different applications. In this chapter, we mainly discuss about transactional applications which needs CPU resources mostly, other than other types of resources.
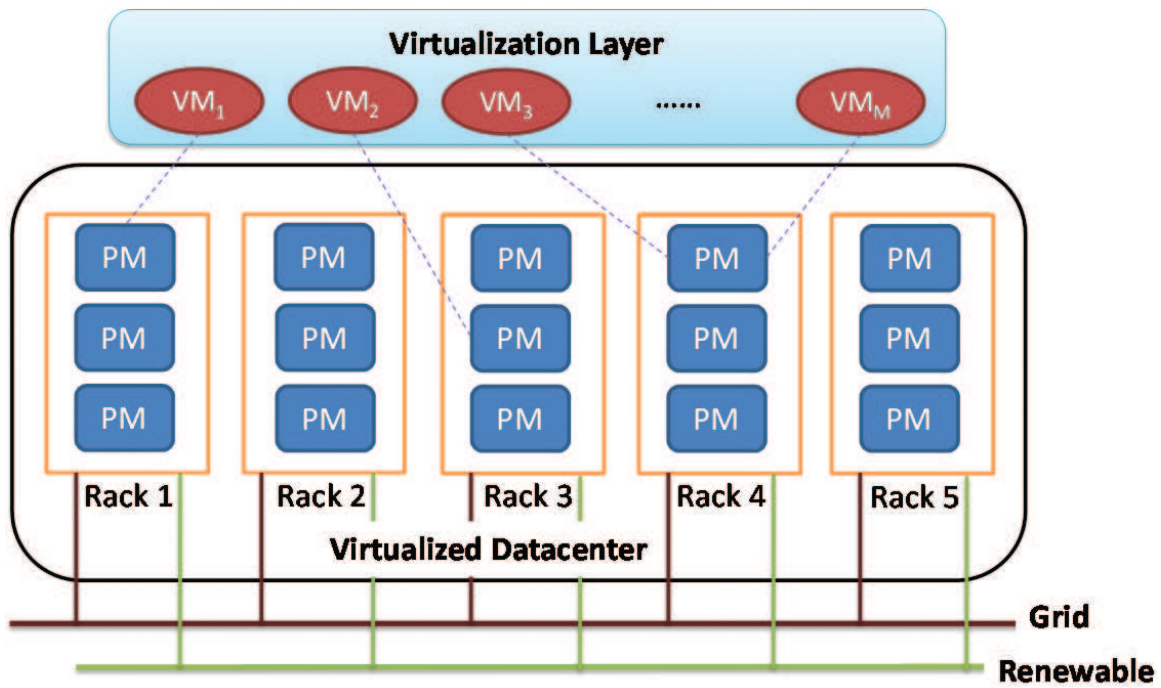
Figure 2. Virtualized datacenter.

4. Problem definition

This section defines necessary variables and also the problem we need to solve throughout this chapter.

4.1. Model of computing and service units

In the target problem, there are N heterogeneous physical machines in the virtualized cloud environment, and the available CPU resource capacity of PM i is denoted as Φ_i. The entire environment is hosting M kinds of different applications, deployed on M different VMs. Denote the jth VM as VM_j. Then, denote x_j as the index of the PM which is hosting VM_j. Denote ϕ_j as the allocated CPU capacity to VM_j and d_j as the demanded CPU capacity of application j at the current time slot.

4.2. Power consumption model

According to the mechanisms of dynamic voltage and frequency scaling (DVFS) techniques, here we use a simple power model which assumes that the power consumption of other components in the PM correlate well with CPU [20]. Denote p_i as the power consumption of PM i in each time slot and p_i^{MAX} as the maximum power consumption of PM i (100% occupied by workloads). Then, the following equation can be used to compute the PM power consumption:

$$p_i = p_i^{MAX} \cdot \left(c + (1 - c) \cdot \theta_i \right) \tag{1}$$

where c is a constant number representing the ratio of the idle-state power consumption of a PM compared to the full-utilized-state power consumption [21] and θ_i is the current CPU utilization of PM i.

Besides, we also consider the power cost spent on cooling devices when establishing the power model, which is usually much related to temperature. The cooling system we discuss here consists of both the traditional computer room air conditioning (CRAC) unit and the air economizer. According to relevant studies [10], the coefficient of performance (CoP) is often used to indicate the efficiency of a cooling system, which can be computed by

$$
\text{CoP} = \begin{cases} 1/k\left(T_{\text{sup}} - T_{\text{out}}\right), & \text{when } T_{\text{out}} \le T_{\text{sup}} \\ 0.0068\, T_{\text{sup}}^2 + 0.0008\, T_{\text{sup}} + 0.458, & \text{otherwise} \end{cases} \tag{2}
$$

where k is a factor reflecting the difference between outside air and target temperature, T_{sup} is the target supply temperature, and T_{out} is the outside temperature. As it can be observed, Eq. (2) contains two parts, corresponding to the situation whether the CRAC or the air economizer will be used for cooling, respectively.

Hence, the total power consumed by both functional devices and cooling devices can be calculated by

$$
p_{DC} = \left(1 + \frac{1}{\text{CoP}}\right) \cdot \sum_{i=1}^{N} p_i \tag{3}
$$

Furthermore, considering the impact of environmental temperature inside the datacenter, we also tried to exploit thermal-aware VM migration strategies. The power consumption of the servers will make the surrounding environmental temperature increase, due to the dissipated heat. Prior studies [11] provided ways to model the vector of inlet temperatures $\boldsymbol{T}_{\text{in}}$ as

$$
\boldsymbol{T}_{\text{in}} = \boldsymbol{T}_s + \boldsymbol{D} \bullet \boldsymbol{p} \tag{4}
$$

where $\boldsymbol{D}$ is the heat transferring matrix, $\boldsymbol{p}$ is the power consumption vector, and $\boldsymbol{T}_s$ is the supplied air temperature vector.

The thermal-aware strategy tries to reduce the cooling power by balancing the temperature over the servers. Accordingly, the workload on different PMs should also be maintained balanced. Denote T_{safe} as the safe outlet temperature and T_{server} as the outlet temperature of the hottest server. In order to lower the server temperature to the safe level, the output temperature of cooling devices should be adjusted by $T_{adj} = T_{safe} - T_{server}$. Then, the output temperature after adjusted will be $T_{new} = T_{sup} + T_{adj}$. Hereafter, the CoP value can be determined by T_{new} and T_{out} [22].

4.3. Modeling overhead and delay

To reduce the power consumption of the PM, it can be switched to sleeping state which can help save energy as much as possible. In addition, the operational costs also include the VM migration costs, since migrating VMs dynamically will definitely lead to some overhead. Denote a_i as the flag recording whether PM i is active or sleeping. Denote c^A as the cost for activating a PM from sleeping state and c^{MIG} as the cost for migrating a VM from one PM to another. Besides, the time delay is also considered and integrated into the experiments in Section 6 for waking up a PM and migrating a VM.

4.4. Optimization problem formulation

From the resource providers' point of view, the objective should be maximizing the total revenues by meeting the requirements of the hosted applications while minimizing the consumed power and other costs. Usually, the revenues from hosting the applications are related to service quality and the predefined level in the SLA. Assume here that the service quality is reflected by the CPU capacity scheduled to the target application. Denote d_j as the demanded CPU capacity of APP j and ϕ_j as the CPU capacity amount scheduled to APP j. Denote $\Omega^j(\bullet)$ as the profit model for APP j, which gives the actual revenue by serving APP j at a certain quality level.

Since the dynamic action decisions are made during constant time periods, denote τ as the length of one time slot. Denote t as the current time slot, and then in time slot $t+1$, the goal is to maximize the net revenue subject to various constraints. Denote x_j as the index of PM currently hosting VM j, and then the VM placement vector X can be denoted as

$$X = (x_1, x_2, ..., x_j, ..., x_M) \tag{5}$$

Hence, the optimizing objective of the defined problem can be expressed as

$$\max \begin{array}{c} \sum_{j=1}^{M} \Omega^j(d_j, \varphi_j) - c^P \cdot p_{DC} \\ -c^A \cdot \sum_{i=1}^{N} \max(0, a_i(t+1) - a_i(t)) \\ -c^{MIG} \cdot \sum_{j=1}^{M} (x_j(t+1) - x_j(t)) \end{array} \tag{6}$$

where the first term is the total revenue summarized over all of the hosted applications, the second term represents the power consumption costs of the entire datacenter, the third term is the PM wake-up cost, and the last term represents the VM migration cost.

With respect to the objective defined above, the constraints could be expressed as

$$\sum_{x_j=i} \varphi_j \leq \Phi_i \cdot a_i \tag{7}$$

$$0 \leq \varphi_j \leq d_j, j = 1, 2, ..., M \tag{8}$$

$$a_i \in \{0, 1\}, x_j \in [1, N] \tag{9}$$

where Eq. (7) means that the allocated capacity cannot exceed the PM CPU capacity, Eq. (8) means that the CPU scheduled to a VM should be less than its demanded value, and Eq. (9) gives the validated ranges of the defined variables.

5. Methods and strategies

In this section, we design some heuristic methods and also the joint hybrid strategy, and describe the ideas in detail.

5.1. Dynamic load balancing *(DLB)*

The idea of the *DLB* strategy is to make the workload on different PMs balanced by dynamically placing VMs. To achieve the balancing effect, if one PM is detected to be more utilized

than the specified upper threshold, some VMs on this PM will be chosen to migrate otherwhere. As a result, the PM utilization ratio will be controlled in a certain range, and there will be as few overloaded PMs as possible.

5.2. Dynamic VM consolidation (*DVMC*)

According to the features of virtualization techniques, VMs could be consolidated together onto a few PMs to make other PMs zero loaded. Hence, the main idea of the *DVMC* strategy is to consolidate VMs as much as possible aiming at saving more power. Both the upper threshold and the lower threshold of the PM utilization level are defined. If one PM is light loaded enough that its utilization is less than the lower threshold, the VM consolidation process will be triggered. After this process, VMs upon underutilized PM will be migrated onto other PMs. Finally, zero-loaded PMs could be turned into inactivate state in order to save more power.

5.3. Joint optimal planning (*JOP)*

The *JOP* strategy aims to optimize the VM placement scheme with the objective of sufficiently utilizing the renewable energy and reducing the total costs.

5.3.1. Renewable energy forecasting

Since renewable energy is used as one source of power supply, we have to forecast the input power value in the next time slot. Here the *k*-nearest neighbor (*k*-NN) algorithm is adopted. A distance weight function is designed to calculate the distance each solar radiation values, as follows:

$$w_i = (1/d^i)/(1/d^1 + 1/d^2 + \ldots + 1/d^k) \tag{10}$$

where d^i is the distance between the ith neighbor and the current point.

Figure 3 shows the forecasting effect on one day in October 2013. The data were measured and collected in Qinghai University, Xining, Qinghai Province of China. By analyzing the data points, the allowed absolute percentage errors (AAPE) of 97.01% data are less than 30%. The accuracy of the prediction method depends on the similar weather conditions in the recent past and may be affected by weather forecast data.

5.3.2. Stochastic search

In order to look for the best scheme of VM placement, we use stochastic search to do the optimization. Specifically, the genetic algorithm (GA) is modified and employed as follows:

For a typical genetic algorithm, there are two basic items as follows:

1. A genetic representation of solution space

Here, for this problem, the decision variable is the vector of VM placement, which can be denoted as $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_M)$.
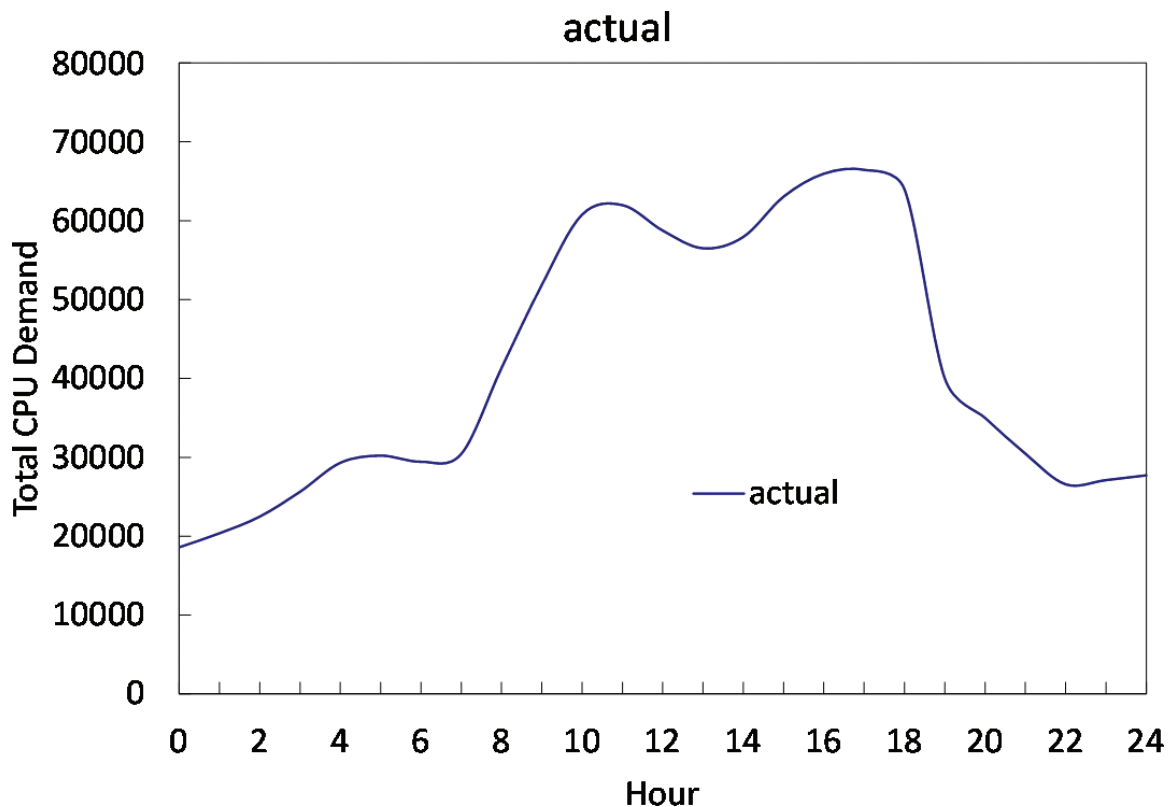
Figure 3. An example of renewable energy forecasting effect.

2. A fitness function to compute the value of each solution

As described, the objective function defined by Eq. (6) could be used as the fitness function. It is functional in measuring the quality of a certain solution. Hereafter, the fitness function will be denoted as *F(X)*.

The procedure of genetic algorithm can be divided into following steps:

  i. Initialization

First, we add the current configuration vector in the last time epoch into the initial generation. Besides, a fixed number (denoted as n_g) of individual solutions will be randomly generated. Specifically, a part of the elements of each solution will be generated randomly, in the range of 0~N−1.

 ii. Selection

After initialization, the generations will be produced successively. For each generation, n_b best-ranking individuals from the current and past population will be selected to breed a new generation. Then, in order to keep the population constant, the remained individuals will either be removed or replicated based on its quality level. The selection procedure is conducted based on fitness, which means that solutions with higher fitness values are more prone to be selected.

According to such concepts, the probability to select an individual X_i can be calculated as

$$P(X_i) = \frac{F(X_i)}{\sum_{k=1}^{n_g} F(X_i)} \tag{11}$$

In this way, less fit solutions will be less likely to be selected, and this helps to keep the diversity of the population and to keep away from premature convergences on poor solutions.

1. Reproduction

After selection, a second generation of population should be generated from those selected solutions through two kinds of genetic operators: crossover and mutation.

The crossover operator first selects two different individuals, denoted as $X^1 = \left(x_1^1, x_2^1, ..., x_M^1\right)$ and $X^2 = \left(x_1^2, x_2^2, ..., x_M^2\right)$. Then, a cutoff point k is set from the range 1~M. Both X^1 and X^2 are divided into two halves, and the second half of them will be swapped and then $X^{1'} = \left(x_1^1, x_2^1, ...x_k^1, x_{k+1}^2, ..., x_M^2\right)$ and $X^{2'} = \left(x_1^2, x_2^2, ..., x_k^2, x_{k+1}^1, ..., x_M^1\right)$. As a result, two new individuals will come out, which is perhaps already in the current population or not.

After crossover, the mutation operator will mutate each individual with a certain probability. The mutation process starts by randomly choosing an element in the vector and then changing its value, and then converts an individual into another.

2. Termination

This production process will repeat again and again until the number of generations reaches to a predefined level.

6. Evaluation results

This section shows our experiments comparing different strategies, and then the results and some details will be discussed.

6.1. Parameter settings

For the following experiments, we used C#.NET to develop the simulation environment and set up the prototype test bed. Specifically, a virtualized datacenter is established, comprised of 40 PMs with CPU capacity of 1500 MIPS each. For the power model, p_i^{MAX} is set to 259W according to Ref. [23] and, c is set to 66% according to Ref. [21]. Then, 100 VMs hosting different applications were simulated and put on the PMs. The workload on each VM fluctuates with time, with the value randomly generated under the uniform distribution.

Table 1 shows all of the parameter settings in detail, and **Figure 4** shows variation of the total CPU demand summarized over all of the workloads, from which it can be seen that there are two peaks in the 24-h period.

We defined a nonlinear revenue function for each application, as mentioned in Section 4. **Figure 5** shows some three typical examples. It can be seen that the revenue of every application changes elastically in a certain range.

	APP 1	APP 2	APP 3
Lower bound	50	40	30
Upper bound	90	60	70
U_j^{max}	100	60	80

Table 1. Parameter settings for example applications.
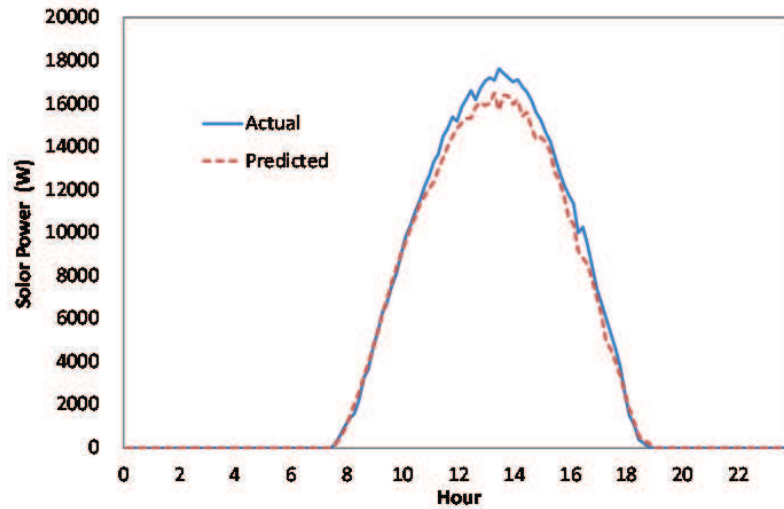

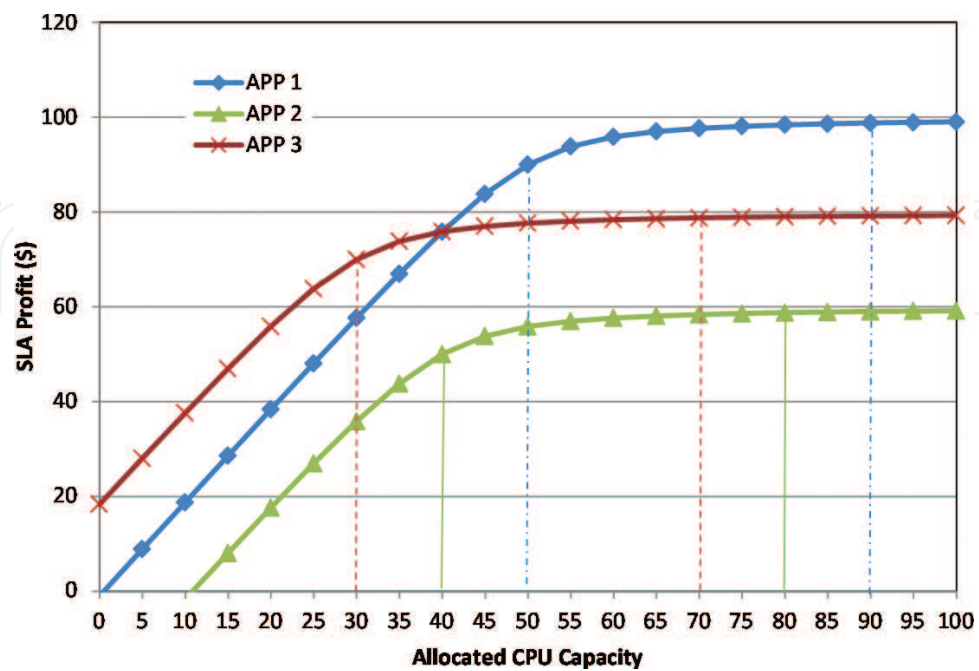**Figure 4.** The workload variation.

Figure 5. The revenue functions of three example applications.

The control interval for reconfiguration actions in the experiment is set to 60 minutes. According to Refs. [24–26], we set c^P to $0.08, set c^A to $0.00024, and set c^{MIG} to $0.00012. The VM migration delay is set to 5s, and the PM wakeup delay is set to 15s. The total experiment time is set to 1440 minutes. The temperature data used in the experiments come from the realistic data measured on 4 October 2013, recorded in the campus of the Qinghai University, Xining, Qinghai Province, China, as shown in **Figure 6**.
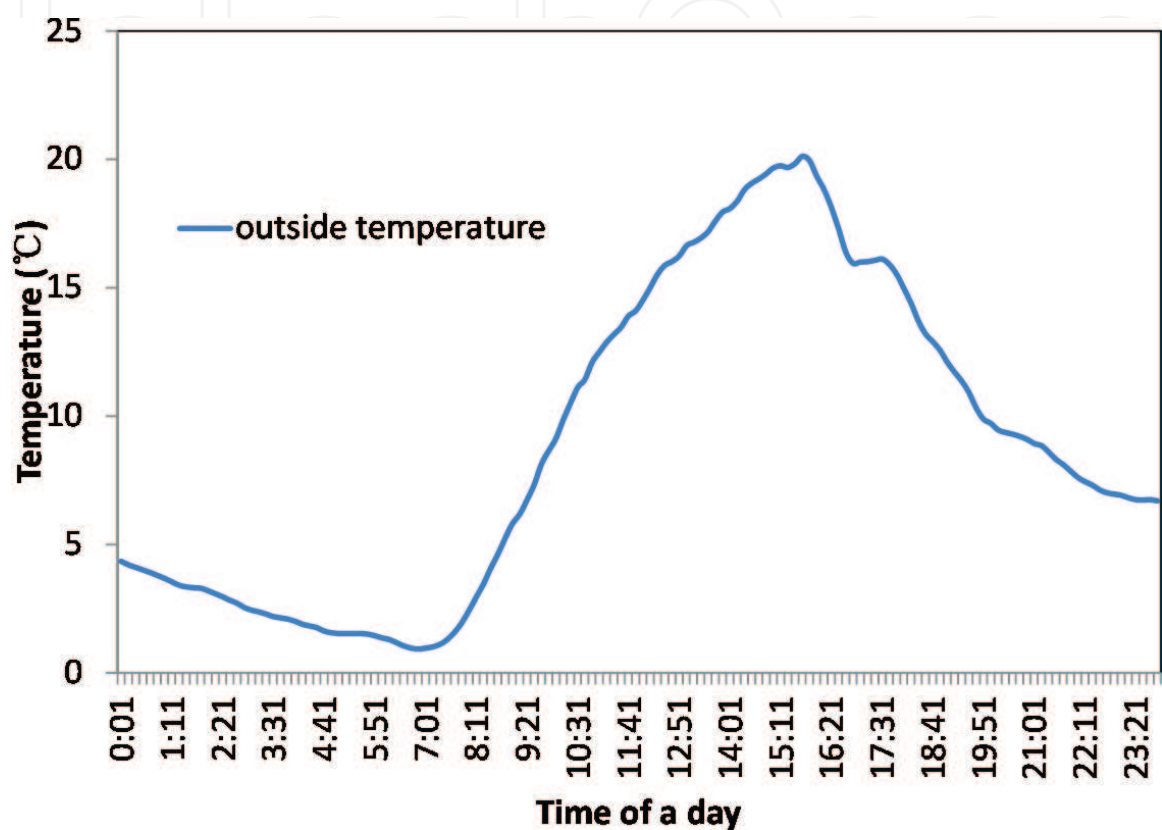
Figure 6. Variation of the outside temperature in 1440 min.

6.2. Results

In order to investigate the effectiveness of the proposed strategy, we will compare the performance among three different strategies – *DLB*, *DVMC*, and *JOP*, as stated in Section 5.

6.2.1. Revenues

As described in Section 4, the net revenue is a main optimizing objective in our problem. **Figure 7** shows the total accumulated net revenues throughout the 1440-min experiment time. It can be observed that the *JOP* strategy can keep the net revenue relatively higher than other ones. Moreover, the *DVMC* approach behaves relatively better than *DLB* since it can save more power by VM consolidation. By examining the detailed data, we found that *JOP* could make the gains 38.2 and 24.2% higher than *DLB* and *DVMC*, respectively, with respect to the accumulated revenue.
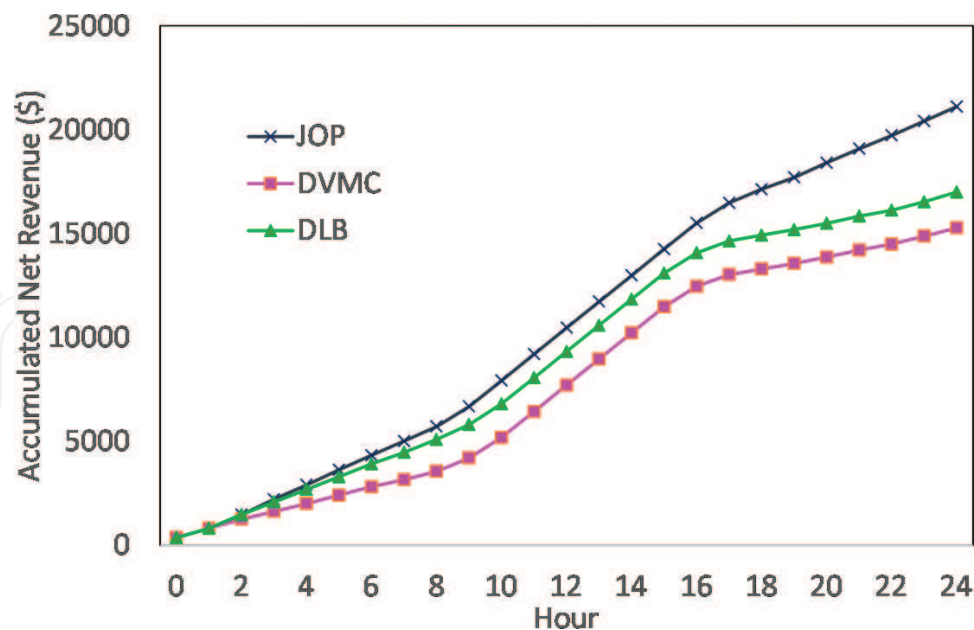
Figure 7. Accumulated net revenue variation.

6.2.2. Power consumption

Now we intend to investigate the power consumption in detail when using *JOP*, as **Figure 8** illustrates. It can be observed from the figure that *JOP* is able to follow the solar energy variation quite well. When the solar power drops to insufficient level, *JOP* is prone to degrade the application performance to save more power. On the contrary, when the solar power arises, *JOP* allows both functional and cooling devices to consume more power, under the constraints of the input power. Interestingly, we can see that the temperature varies more or less in coincidence with solar energy generation, which implies that thermal-aware cosched-uling of energy supply and consumption might be promising, since the temperature also affects energy consumption to some extent.

6.2.3. PM Management

Figure 9 shows the number of active servers when using the three different strategies. We can see that *JOP* can increase or decrease the number of active servers according to the variation of the solar power generation amount. Under the *DLB* strategy, all PMs are kept active so that the system-wide workload could be balanced. Comparatively, *DVMC* uses much fewer active PMs than *DLB* due to VM consolidation. However, it still uses more PMs at night time because it cannot effectively deal with the relationship of revenues and costs. Overall, *JOP* tries to manage PMs dynamically toward the optimization objective and thus can keep the number of active PMs as needed.

6.2.4. Energy for cooling

The cooling energy consumption is also investigated when using the three different strategies, as shown in **Figure 10**. As illustrated, *JOP* allows cooling devices to consume more power

until after 18:00, showing its capability of catching the solar energy variation. By forecasting the solar power generation amount, *JOP* is able to make better decisions for migrating VMs according to the optimized scheme.
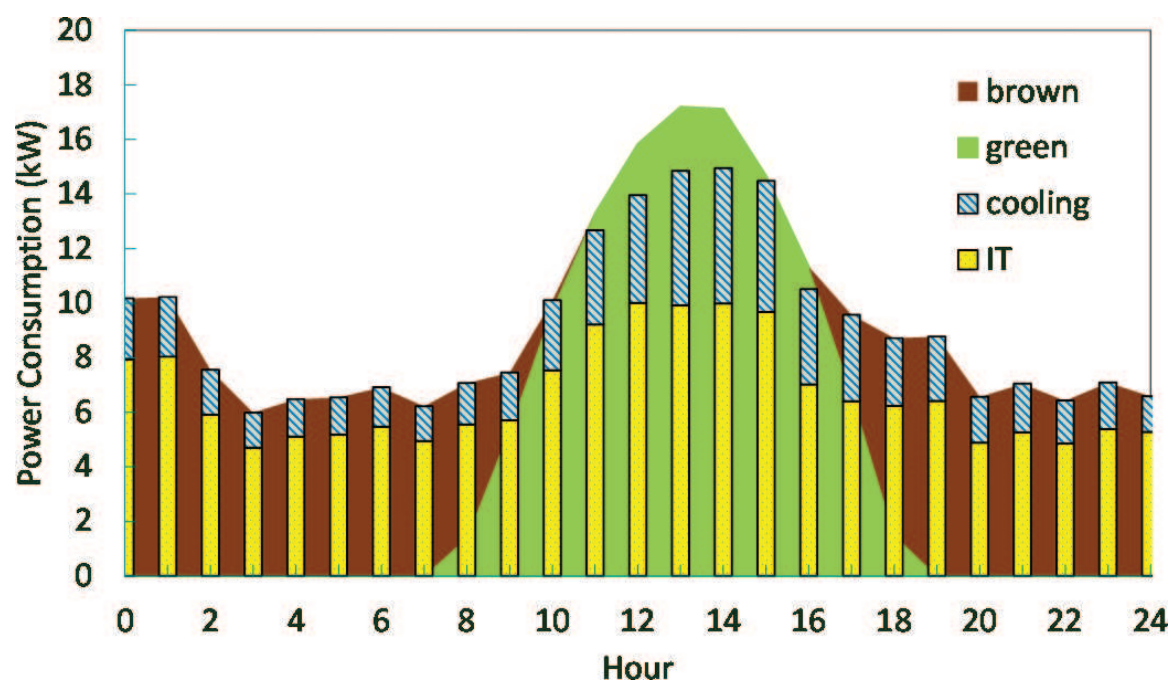

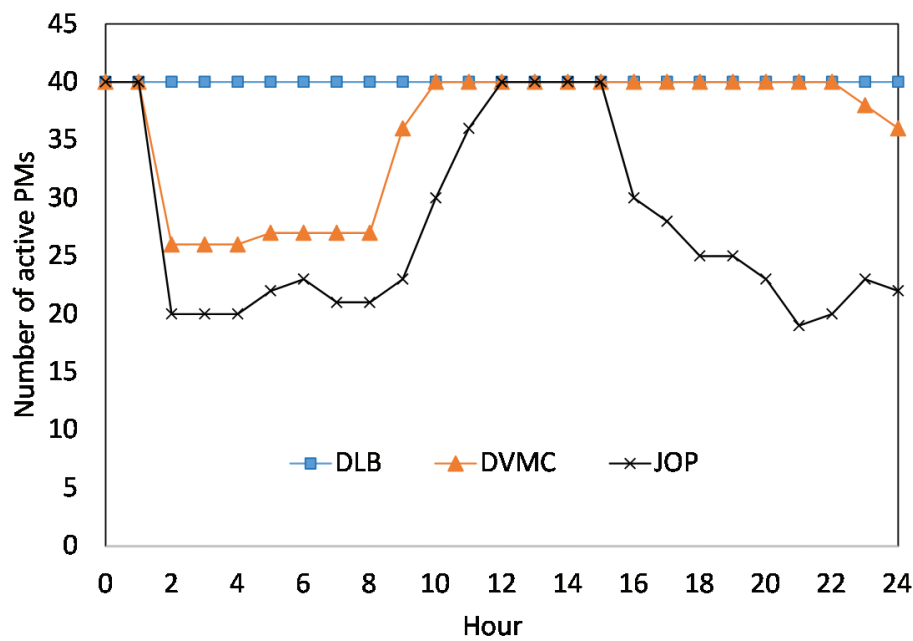**Figure 8.** Power consumption details under *JOP* strategy.

Figure 9. Variation of the number of activated PMs.
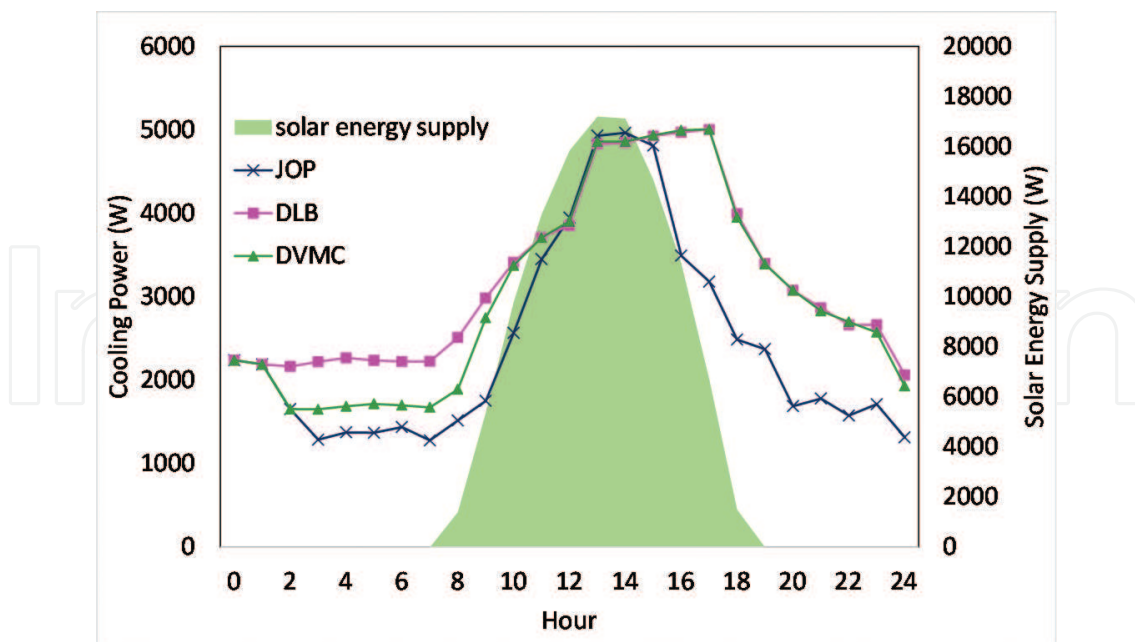
Figure 10. Cooling energy consumption under three different strategies.

7. Conclusion and future work

As the energy consumption of large-scale datacenters becomes significant and attracts more attentions, renewable energy is being exploited by more enterprises and cloud providers to be used as a supplement of traditional brown energy. In this chapter, we introduced the target system environment using hybrid energy supply mixed with both grid energy and renewables. From the datacenter's own point of view, the optimization problem was defined aiming at maximizing net revenues. Accordingly, three different strategies were designed to migrate VMs across different PMs dynamically, among which the *JOP* strategy could leverage stochastic search to help the optimization process. Results illustrate the feasibility and effectiveness of the proposed strategy and further investigation about the accumulated revenues, PM states, and cooling power consumption helps us to see more details of the working mechanisms of the proposed strategy.

As datacenters become larger and larger and thus enormous amount of energy is still needed to power these datacenters, it can be expected that green sources of energy will attract more insights to provide power supplies instead of traditional brown energy. Our work tries to explore some strategies to migrate VMs inside a datacenter in a green-aware way. Nevertheless, there are still a lot of challenges in the field of leveraging sustainable energy to power the datacenters. On one hand, more kinds of clean energy sources besides wind and solar could be exploited, such as hydrogen and fuel cell, and their features should be studied and developed. On the other hand, how to synthetically utilize the battery, utility grid, and datacenter loads to solve the intermittency and fluctuation problems of the energy sources remains a difficult problem for system designers. In addition, it is also necessary and interesting to conduct some research on the air flow characteristics among racks and server nodes inside the datacenter room and develop some thermal-aware scheduling approaches correspondingly.

Acknowledgements

Author details

Xiaoying Wang[1]*, Guojing Zhang[1], Mengqin Yang[1] and Lei Zhang[1,2]

*Address all correspondence to: xy.wang@foxmail.com

1 Department of Computer Technology and Application, State Key Laboratory of Plateau Ecology and Agriculture, Qinghai University, Xining, China

2 College of Computer Science, Sichuan University, Chengdu, China

References

[1] H.S. Dunn. The carbon footprint of ICTs. [Internet]. 2010. Available from: https://www.giswatch.org/thematic-report/sustainability-climatechange/

[2] Greenpeace. Clicking clean: how companies are creating a green internet: 2015 update. [Internet]. 2015. Available from: http://www.greenpeace.org/usa/wp-content/uploads/legacy/Global/

[3] P. Patel, A. Ranabahu, A. Sheth. Service level agreement in cloud computing. In: Cloud Workshops at OOPSLA09; Orlando, Florida; ACM; 2009; pp. 1–10.

[4] V. Sharma, A. Thomas, T. Abdelzaher, et al. Power-aware QoS management in web servers. In: 24th IEEE International Real-Time Systems Symposium; Cancun, Mexico; IEEE; 2003. p. 63.

[5] M. Tanelli, D. Ardagna, M. Lovera, et al. Model identification for energy-aware management of web service systems. In: the 6th International Conference on Service-Oriented Computing; Sydney, Australia; Springer; 2008. pp. 599–606.

[6] A. Berl, E. Gelenbe, M. Di Girolamo, et al.. Energy-efficient cloud computing. Comput. J. 2010;**53**(7):1045.

[7] R. Urgaonkar, U.C. Kozat, K. Igarashi, et al. Dynamic resource allocation and power management in virtualized datacenters. In: 2010 IEEE Network Operations and Management Symposium (NOMS); Osaka, Japan; IEEE; 2010. pp. 479–486.

[8] R. Mukherjee, S.O. Memik, G. Memik. Temperature-aware resource allocation and binding in high-level synthesis. In: Proceedings of the 42nd annual Design Automation Conference; Anaheim, CA, USA; ACM; 2005. pp. 196–201.

[9]   Q. Tang, S. Gupta, G. Varsamopoulos. Thermal-aware task scheduling for datacenters through minimizing heat recirculation. In: 2007 IEEE International Conference on Cluster Computing; Austin, TX, USA; IEEE Computer Society; 2007. pp. 129–138.

[10] E. Pakbaznia, M. Ghasemazar, M. Pedram. Temperature-aware dynamic resource provisioning in a power-optimized datacenter. In: Design, Automation & Test in Europe Conference & Exhibition (DATE); Dresden Germany; IEEE Computer Society; 2010. pp. 124–129.

[11] L. Wang, S.U. Khan, J. Dayal. Thermal aware workload placement with task-temperature profiles in a datacenter. J. Supercomput. 2012;**61**(3):780-803.

[12] M.A. Islam, S. Ren, N. Pissinou, A.H. Mahmud, A.V. Vasilakos. Distributed temperature-aware resource management in virtualized data center. Sustainable Computing: Informatics and Systems 2015;**6**:3-16. DOI: 10.1016/j.suscom.2014.03.002

[13] N. Deng, C. Stewart, D. Gmach, et al. Policy and mechanism for carbon-aware cloud applications. In: Network Operations and Management Symposium; Maui, Hawaii, USA; Institute of Electrical and Electronics Engineers (IEEE); 2012. pp. 590–594.

[14] I. Goiri, R. Beauchea, K. Le, et al. GreenSlot: scheduling energy consumption in green datacenters. In: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis; Seattle, WA, USA; ACM; 2011. p. 20.

[15] I. Goiri, K. Le, T.D. Nguyen, et al. GreenHadoop: leveraging green energy in data-processing frameworks. In: Proceedings of the 7th ACM European conference on Computer Systems; Bern, Switzerland; ACM; 2012. pp. 57–70.

[16] C. Li, A. Qouneh, T. Li. iSwitch: coordinating and optimizing renewable energy powered server clusters. In: 2012 39th Annual International Symposium on Computer Architecture (ISCA); Portland, OR, USA; IEEE Computer Society; 2012. pp. 512–523.

[17] M. Arlitt, C. Bash, S. Blagodurov, et al. Towards the design and operation of net-zero energy datacenters. In: The 13th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm); San Diego, USA; IEEE; 2012. pp. 552–561.

[18] W. Deng, F. Liu, H. Jin C. Wu. SmartDPSS: cost-minimizing multi-source power supply for datacenters with arbitrary demand. In: 2013 IEEE 33rd International Conference on Distributed Computing Systems (ICDCS); Philadelphia, PA; 2013. pp. 420–429.

[19] Z. Liu, I. Liu, S. Low, A. Wierman. Pricing data center demand response. In: The 2014 ACM international conference on Measurement and modeling of computer systems – SIGMETRICS '14; Austin, Texas, USA; ACM; 2014. pp. 111–123.

[20] X. Fan, W.-D. Weber, L.A. Barroso. Power provisioning for a warehouse-sized computer. ACM SIGARCH Comput. Archit. News. 2007; **35**(2): 13–23.

[21] A. Beloglazov, J. Abawajy, R. Buyya. Energy-aware resource allocation heuristics for efficient management of datacenters for cloud computing. Future Gener. Comput. Syst. 2012;**28**(5):755–768.

[22] F. Ahmad, T. Vijaykumar. Joint optimization of idle and cooling power in datacenters while maintaining response time. In: Proceedings of the fifteenth edition of ASPLOS; Pittsburgh, PA, USA; ACM; 2010. pp. 243–256.

[23] K.-D. Lange. Identifying shades of green: the SPECpower benchmarks. Computer. 2009;**42**(3):95–97.

[24] D. Ardagna, B. Panicucci, M. Trubian, et al. Energy-aware autonomic resource allocation in multi-tier virtualized environments. IEEE Trans. Serv. Comput. 2010; 5(1): 2–19.

[25] D. Kusic, J.O. Kephart, J.E. Hanson, et al. Power and performance management of virtualized computing environments via lookahead control. Cluster Comput. 2009;**12**(1):1-15.

[26] M. Steinder, I. Whalley, D. Carrera, et al. Server virtualization in autonomic management of heterogeneous workloads. In: 10th IFIP/IEEE International Symposium on Integrated Network Management; Munich, Germany; IEEE; 2007. pp. 139–148.