Energy-Efficient Independent Task Scheduling in Cloud Computing

Xia Zhu(✉), Mehboob Hussain, and Xiaoping Li(✉)

Southeast University, Nanjing, China
{zhuxia,Mehboob,xpli}@seu.edu.cn

Abstract. The high scientific applications which contain thousands of tasks are usually executed in virtulized cloud for many benefits. With the increment of the processing capability of the cloud system, the computation energy is significantly consumed along. Thus efficient energy consumption methods are quite necessary to save the energy cost. In this paper, the independent task scheduling problem in a cloud data center is considered. It is a big challenge to achieve the tradeoff between the minimization of computation energy and user-defined deadlines. A heuristic is proposed which consist of an energy efficient task sequencing method and a virtual machine searching strategy. Experimental results show that the proposed heuristic clearly outperforms the other algorithms.

Keywords: Virtualized cloud · Energy consumption · Task scheduling

1 Introduction

Cloud computing has started to dominate the computing environment in current days. It provides high and scientific applications being executed in the cloud for various benefits. With increase of processing capabilities, the energy consumption has also increased significantly [1]. Energy efficient execution manner for these tasks in cloud system become very essential. During the past decades, the power consumption of computing resources accounted 45% of the total power consumption in a data center [2]. Many operations have been conducted to reduce the energy consumption of the computing resources [3]. It is usually adopted to assign tasks to the slower virtual Machines (VMs) while meeting the deadline and ensuring quality of services. However, the requirements of cloud users and cloud service provider (CSP) are conflicted. The goal of CSP is to schedule tasks submitted by the cloud users in an optimal way such that it should meet the deadline and quality of service (QoS) with minimum computation energy.

Independent task scheduling problem focuses on the scheduling of a set of independent tasks to be run on heterogeneous VMs. By using the modern virtualization techniques, a large scale of user tasks can be simultaneously executed in cloud. The main goal is to properly schedule these tasks in a way that minimizes the computing resource energy and determines suitable resources for tasks

Y. Tang et al. (Eds.): HCC 2018, LNCS 11354, pp. 428–439, 2019.
https://doi.org/10.1007/978-3-030-15127-0_43

under deadlines. This problem extensively exists in the practical environment such as relational database queries, parametric studies and image processing, where parallel tasks are submitted to cloud provider for execution.

In order to resolve the energy problem in cloud system, many researches tried to improve energy efficiency of computing resources. A common data center usually composes of different types of VMs, which have different process capacity and power consumption characteristics [4]. An effective way is to use energy-aware scheduling approach which leverages computing resources heterogeneity. It was proved in [5] that a hybrid data center with low power VMs and high performance VMs can achieve the efficient energy purpose.

In this paper, the independent tasks scheduling problem with the purpose of minimizing resource energy consumption under user deadlines is considered. The contributions of this paper include:

– Firstly, all tasks are arranged in energy efficient order by three rules.
– Secondly, a novel scheduling algorithm is proposed for searching heterogeneous VMs to effectively reduce energy consumption and finish all tasks before deadlines.
– Lastly, experiments are performed in cloud system simulation environment to validate the proposed algorithm.

The rest of this paper is organized as follows: In Sect. 2, the related works in literature are summarized. Problem definition and mathematical model are given in Sect. 3. In Sect. 4, energy-efficient scheduling approaches for tasks in cloud are proposed. Section 5 contains the simulation experiments and performance analysis. Finally, the paper is concluded in Sect. 6.

2 Related Work

With the expanded demand of cloud computing infrastructures and the explosion in data center sizes, energy efficiency becomes a crucial issue and most of them are proved to be NP-hard. Many researchers have begun to study about energy-efficient policies for reducing energy from computing resources. There are several ways to reduce computer resource energy such as maximize resource utilization, minimize VM migration, dynamic voltage and frequency scaling (DVFS) technique and efficient task scheduling.

From the perspective of energy-efficient task scheduling, most existing works focus on task scheduling and suitable resource allocation in cloud to minimize energy. Garg et al. [6] proposed a technique for task scheduling problem in a heterogeneous data center to get the minimum energy consumption for all type of tasks. But there was an assumption that an upper bound of total jobs arrive rate is known ahead, which may not be available in a real cluster. Yigitbasi et al. [7] proposed some heuristics to get a worst energy saving of workload in Hadoop clusters. But they only focused on the MapReduce workload and neglected other types of workload in current clusters. Liu et al. [8] analyzed a heterogeneous cluster with parallel tasks and proposed an energy consumption model. They also presented an energy-aware task scheduling strategy based on clustering to shorten scheduling lengths while keeping energy consumption minimal. Li et al. [9] proposed an energy-aware task scheduling algorithm for heterogeneous clusters based on Min-Min heuristic. Its goal was to get a best time energy tradeoff. Mukherjee et al. [10] depicted three thermal-aware energy-saving job scheduling techniques to reduce the energy consumption of the data center under some performance constraints.

Some researches considered reducing power consumption in virtual computing environments. Liu et al. [11] proposed the GreenCloud architecture to reduce data center power consumption, while guaranteing the performance and leveraging live VM migration technology. Beloglazov et al. [12] proposed and evaluated heuristics for dynamic reallocation of VMs to minimize energy consumption while providing reliable QoS. Verma et al. [13] presented several approaches to capture the cost-aware application placement problem. Li et al. [14] implemented and validated a dynamic resource provision framework for virtualizing server environments. Kusic et al. [15] examined and evaluated three local resource allocation policies based on shortest queue in a heterogeneous cluster.

In contrast to previous work, the energy efficient task scheduling problem with the constraint of user deadlines is considered in this paper. The proposed scheduling algorithm focuses on reducing the energy consumption by assinging the independent tasks to the lower energy consumption VMs under the consideration of deadlines.

3 Problem Description

In this paper, the problem that a set of independent tasks being scheduled in a cloud data center which comprised of heterogeneous VMs is considered. Each task should be effectively assigned to an appropriate VM with certain quantity of computing capacity to execute. For simplicity, several assumptions for practical environments are used:

- Each task is only executed on one VM, neither task migration nor interruption is allowed.
- VMs reside in single data center, break-down is not concerned.
- Data transmission time of each task is neglected. Only the power consumption of VMs is considered.

Notations to be used in this paper are listed in Table 1. The system model, application and resource models of the considered problem are given in the following sections.

Table 1. Notations used in the paper

Notation	Explanation
N	Number of task v
M	Number of virtual machine $\mathcal{V}$
$\mathcal{V}_j$	The jth virtual machine in a data center
v_i	The ith independent task
W_i	The workload of the task v_i
d_i	The deadline of the task v_i
ζ_j	The speed of virtual machine $\mathcal{V}_j$
p_j	The power of virtual machine $\mathcal{V}_j$
T_i^e	The execution time of the task v_i
$x_{i,j}$	The determined placement of task v_i and virtual machine $\mathcal{V}_j$
B_i	The begin time of task v_i
F_i	The finish time of task v_i
T_i^{slack}	The slack time of task v_i

3.1 System Model

Figure 1 presents the system architecture of the considered problem, which includes two components: the Master Node and the Data Center Component (DCC). Tasks are first submitted to the Master Node by the cloud user. The Master Node is responsible for scheduling tasks to the appropriate VMs to minimize the energy consumption. The Master Node also plays the role as a connector between the cloud user and the DCC. Three types of VMs are configured in advance in the DCC: Small, Medium and Large.

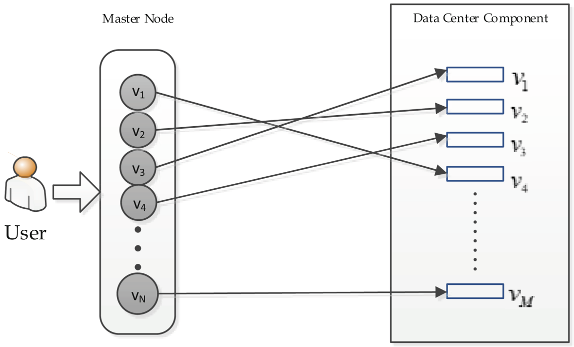

Fig. 1. The system model architecture

3.2 Application and Resource Model

Tasks are represented by $\{v_1, v_2, v_3, \ldots, v_N\}$, and VMs in a data center are defined by $\{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_M\}$. Each task has a workload W_i $(i = 1, \ldots, N)$ and a deadline d_i, which is defined by the cloud user when tasks are submitted. The computation speed and power of VM $\mathcal{V}_j$ $(j = 1, \ldots, M)$ are denoted as ζ_j and p_j, respectively. To minimize the energy consumption, each task is assigned to the slower VM under deadline d_i, which needs less energy. $x_{i,j} \in \{0, 1\}$ is a decision variable, where $x_{i,j} = 1$ only if task v_i is assigned to $\mathcal{V}_j$. The energy consumption of task v_i is determined by the power p_j and the execution time T_i^e, i.e., $T_i^e = \sum_{j=1}^{M} x_{i,j} \times \frac{W_i}{\zeta_j}$.

The considered problem can be mathematically modeled as below:

$$\min Z = \sum_{i=1}^{N} \sum_{j=1}^{M} x_{i,j} \times p_j \times T_i^e \tag{1}$$

s.t.

$$T_{j,0} = 0 \tag{2}$$

$$T_{j,k} = T_{j,k-1} + \sum_{k=1}^{N} x_{k,j} T_k^e \tag{3}$$

$$T_i^e = \sum_{j=1}^{M} x_{i,j} \times \frac{W_i}{\zeta_j} \tag{4}$$

$$F_i = \sum_{j=1}^{M} x_{i,j} \times T_{j,k} \tag{5}$$

$$F_i \leq d_i \tag{6}$$

$$\sum_{i=1}^{N} x_{i,j} = 1 \tag{7}$$

$$\sum_{j=1}^{M} x_{i,j} = 1 \tag{8}$$

$$x_{i,j} \in \{0, 1\} \tag{9}$$

Equation 1 calculates the consumed energy of all tasks. For each $\mathcal{V}_j$, the finish time $T_{j,0}$ is initialized to 0 in Eq. 2. The finish time $T_{j,k}$ of $\mathcal{V}_j$ when executing task v_k is determined by the finish time of previous task v_{k-1} and the execution time $\sum_{k=1}^{N} x_{k,j} T_k^e$ of the current task v_k, $T_{j,k}$ is formulated in Eq. 3, and T_i^e is defined in Eq. 4. The finish time F_i of v_i is presented in Eq. 5. Equation 6 reveals that the finish time F_i of v_i should be less than its deadline d_i. Equations 7 and 8 illustrates that each VM is assigned to one task and each task is assigned to one VM. Equation 9 implies that v_i is allocated to $\mathcal{V}_j$ or not.

4 Proposed Algorithm

In this section, an energy efficient heuristic scheduling algorithm is proposed to find an optimal or near optimal solution to complete all N tasks on M VMs with minimum or near minimum execution energy Z while meet the deadline d_i. As illustrated in Algorithm 1, the proposed heuristic contains two phases: (i) a task sequence is generated according to three rules inspired by HEFT algorithm [16], (ii) tasks are iteratively selected and assigned to the most energy efficient VMs.

Algorithm 1. The Algorithm Framework

```
Input: $Q_v$; $Q_d$; $\{\mathcal{V}_{j,1}, \ldots, \mathcal{V}_{j,m_j}\}$
begin
    $Z \leftarrow 0$;
    Task sequencing;
    foreach $v_i \in Q_v$ do
        $Z_i \leftarrow$VM Searching;
        $Z \leftarrow Z + Z_i$
    return $Z$;
```

4.1 Task Sequencing

After being submitted by users, a set of independent tasks are sequenced according to their deadlines, workloads and slack time. The slack time T_i^{slack} is determined by the actual finish time F_i and d_i of v_i, i.e., $T_i^{slack} = d_i - F_i$. F_i is decided by the execution time T_i^e and the available time of the assigned VM. However, since the heterogeneity of the VMs, the T_i^e is undetermined before scheduling and the average execution time $\overline{T_i^e} = \frac{\sum_{j=1}^{M} W_i}{\sum_{j=1}^{M} \zeta_j}$ is employed to estimate the task execution time. Assumed that the available time of each VM is 0, the T_i^{slack} can be calculated as:

$$T_i^{slack} = d_i - \overline{F_i} \tag{10}$$

$$\overline{F_i} = \sum \overline{T_i^e} \frac{\sum_{j=1}^{M} W_i}{\sum_{j=1}^{M} \zeta_j} \tag{11}$$

Since the deadline, size and slack time are critical to task sequencing, three different rules are developed as follows.

(1) Earliest Deadline First (EDF): Tasks are sequenced based on the ascending order of their deadlines. If the deadlines of two tasks are the same, the one with the smaller size will be ranked with a higher priority.
(2) Smallest Slack Time First (SSF): Tasks are sorted based on the ascending order of their slack time. If the slack time is same of any tasks, the one with the smallest total workload will be arranged first.
(3) Smallest Workload First (SWF): Tasks are sequenced based on the ascending order of their sizes.

4.2 VM Searching

The allocation of each task v_i to a VM is to make a decision on the $x_{i,j}$, i.e., $x_{i,j} = 1$ if the v_i is assigned to $\mathcal{V}_j$. As depicted in Eq. 1, the energy consumption is decided by the power p_j of the assigned VM and the task execution time T_i^e. Performance per Watt PpW_j is used in this paper to characterize the energy efficiency of $\mathcal{V}_j$, which is defined as

$$PpW_j = \frac{\zeta_j}{p_j}. \tag{12}$$

VMs are sorted according to the ascending order of PpW_j, and the available time $T_{j,0}$ of each $\mathcal{V}_j$ is initialized to 0. To energy efficiently assign each task to the optimal VM, the sorted VMs are traversed from the head to tail. If $T_{j,i-1} + T_i^e < d_i$, then $\mathcal{V}_j$ for v_i is identified, and the available time $T_{j,i}$ is dynamically updated. The details of the VM searching algorithm is described in Algorithm 2.

Algorithm 2. VM Searching (VMS)

```
Input: $v_i$: task to schedule
begin
    $Q_{vm} \leftarrow$ Sort the VMs by $PpW_j$ with the ascending order;
    $\mathcal{V} \leftarrow NULL$;
    foreach $\mathcal{V}_j \in Q_{vm}$ do
        $T_{j,0} \leftarrow 0$;
    foreach $\mathcal{V}_j \in Q_{vm}$ do
        Calculate the $T_i^e$ of $\mathcal{V}_j$ by Eq.4;
        if $T_{j,i-1} + T_i^e < d_i$ then
            Calculate the $T_{j,i}$ of $\mathcal{V}_j$ by Eq.3;
            $\mathcal{V} \leftarrow \mathcal{V}_j$;
            break;
    Calculate the consumed energy $Z_i$ by Eq.1;
    return $Z_i$, $\mathcal{V}$;
```

In line 2, VMs are sorted by PpW_j with the ascending order, and put into sequence Q_{vm} in which VMs are iteratively traversed. In line 3, the result sequence of VMs $\mathcal{V}$ are initialized as null. The available time $T_{j,0}$ of each VM in Q_{vm} is initialized to 0. From line 7 to 11, if the sum of the available time of $\mathcal{V}_j$ and the execution time of v_i is less than the deadline d_i, then v_i is assigned to $\mathcal{V}_j$, and the new available time $T_{j,i}$ of $\mathcal{V}_j$ is dynamically updated.

In Algorithm 2, VMs are swapped at least $M \times log(M)$ times in the sorting process. Besides, the traverse of the sorted VMs consumes M times. Totally, the time complexity of Algorithm 2 is $O(M \times log(M))$.

5 Performance Evaluation

In this section, the parameters and performance of the proposed algorithm are investigated. Different components of the proposed method are analyzed to find the best combination. All algorithms are implemented in Java and configured in the same (Intel (R) Core (TM) i5-3475 CPU @ 3.30GHz, 10G Memory). The operation software of the machine is Windows 10 for carried out our experiment.

5.1 Simulation Setup

In the experiment, five different numbers of task nodes $Q_t \in \{50, 100, 200, 400, 500\}$ are generated. The deadline of each task is defined on basis of the following equation:

$$d_i = F_i + \gamma \times F_i \tag{13}$$

The deadline d_i of a task is the sum of the earliest completion time and a certain percentage of the earliest completion time. γ is used as a parameter to control the tightness of the task deadline with value range $\gamma \in \{.2, .4, .6, .8, 1\}$. So each task can get five different size deadlines noted as $D1$, $D2$, $D3$, $D4$, $D5$.

In the algorithm comparison phase, three existing algorithms Energy Aware Rolling-Horizon (EARH) [17], Earliest Deadline First (EDF) [18] and First Come First Serve (FCFS) [19] are selected to verify the effectiveness of the proposed heuristic. For fairness, all the compared algorithms are executed with the same tasks collection and the same setting of the number of tasks and parameter verification.

In order to measure the performance of the algorithms, the Relative Percentage Deviation (RPD) in Eq. 14 is adopted:

$$RPD(\%) = \frac{Z - Z^*}{Z^*} \times 100\% \tag{14}$$

Z represents the value of the objective function obtained when executing the tasks according to the proposed algorithm. Z^* represents the minimum consumption of energy that all algorithms consume when executing the tasks. All the experimental results will be performed by the Analysis of Variance (ANOVA) technique.

The VM configuration of the performance evaluation is presented in the Table 2. There are five types of VMs, different VMs have different processing speed and the different power. The VMs of each VM type are randomly generated, during the parameter calibration and the algorithm comparison, the VM configuration remains the same.

Table 2. VM Specifications.

VM	VM1	VM2	VM3	VM4	VM5
Core	1	1	1	1	1
MIPS/Core	200	400	600	800	1000
Power/Core	50 W	100 W	150 W	200 W	250 W

5.2 Parameter Calibration

Figure 2 shows the multi factor 95.0% Turkey HSD confidence interval of RPD impact on parameter γ. As we can see, γ has a significant influence on the results of the algorithm. When γ from .2 to .6 a value RPD decreased obviously, and when $\gamma > .6$ RPD values has stabilized. Therefore, .6 is selected in this algorithm.

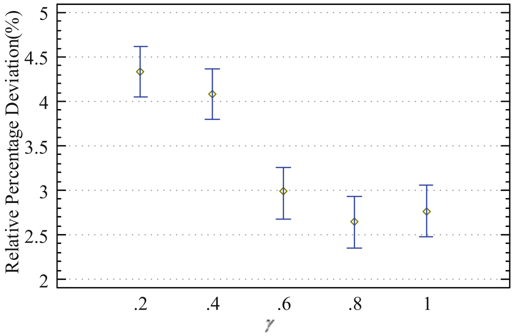

Fig. 2. The parameter γ have 95.0% confidence interval Tukey HSD Mean interval chart

5.3 Task Sequencing Methods

When the tasks are submitted into the scheduling system, three task sequencing rules (EDF, SSF and SWF) are proposed to generate the task scheduling sequence. Three sequencing rules are calibrated to select the most appropriate one. Figure 3 presents the mean plot of three task sequence rules with 95.0% Tukey HSD intervals, the RPD value of SSF is obviously lower than EDF and SWF. It is concluded that the task is scheduled by the task sequence which is generated by the SSF rule leads to smaller energy consumption. Therefore, SSF is selected for the task sequencing component.

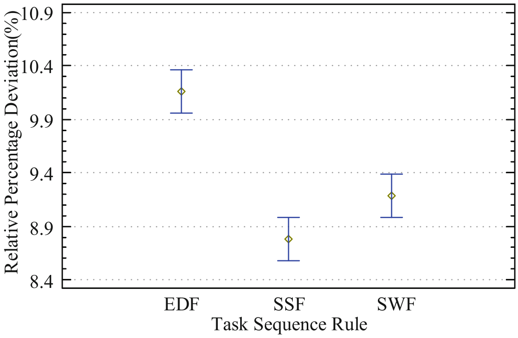

Fig. 3. The mean plot of task sequence rules with 95.0% Tukey HSD intervals

5.4 Algorithm Comparison

To evaluate the performance of the proposed algorithm, three existing task scheduling algorithm: EARH [17], EDF [18] and FCFS [19] are selected as the benchmark algorithms.

Figures 4 and 5 depict the RPD values between the proposed algorithm and the compared algorithms under different deadlines and task numbers. Figure 4 illustrates that the proposed algorithm is not as good as EDF algorithm under the tight deadline, however, with the deadline becoming loose, the RPD values of the proposed EEITS is gradually lower than the compared algorithms. The performance of each algorithm evaluated under various task number is presented in the Fig. 5. The proposed algorithm is obviously better than the compared algorithms.

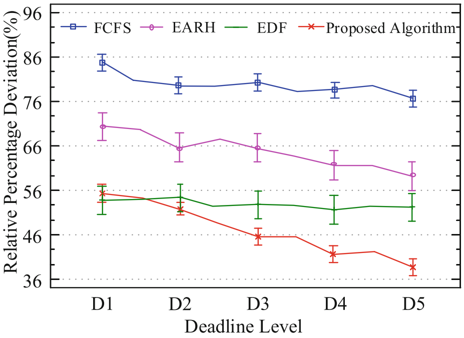

Fig. 4. Comparison of algorithms under different deadlines

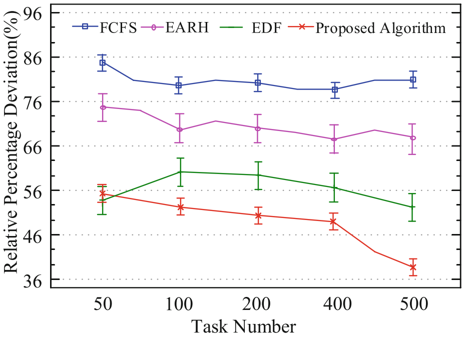

Fig. 5. Comparison of algorithms under different task numbers

6 Conclusion

In this paper, energy-efficient scheduling problem for independent task with user defined deadline in virtulized cloud is investigated. The goal of the paper is to schedule the tasks in an energy efficient way. An energy efficient independent task scheduling heuristic is proposed, which consists of a task sequencing process and an energy efficient VM selection strategy. Experimental results shows that the proposed algorithm outperforms the others in most cases.

References

1. Kizza, J.M.: Guide to Computer Network Security. Springer, London (2017). https://doi.org/10.1007/978-1-4471-4543-1
2. Filiposka, S., Mishev, A., Juiz, C.: Balancing performances in online VM placement. In: Loshkovska, S., Koceski, S. (eds.) ICT Innovations 2015. AISC, vol. 399, pp. 153–162. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-25733-4_16
3. Ebrahimi, K., Jones, G.F., Fleischer, A.S.: A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities. Renew. Sustain. Energy Rev. **31**, 622–638 (2014)
4. Nathuji, R., Isci, C., Gorbatov, E.: Exploiting platform heterogeneity for power efficient data centers. In: Fourth International Conference on Autonomic Computing, 2007, ICAC 2007, p. 5. IEEE (2007)
5. Chun, B.-G., Iannaccone, G., Iannaccone, G., Katz, R., Lee, G., Niccolini, L.: An energy case for hybrid datacenters. ACM SIGOPS Oper. Syst. Rev. **44**(1), 76–80 (2010)
6. Garg, S., Sundaram, S., Patel, H.D.: Robust heterogeneous data center design: a principled approach. ACM SIGMETRICS Perform. Eval. Rev. **39**(3), 28–30 (2011)
7. Yigitbasi, N., Datta, K., Jain, N., Willke, T.: Energy efficient scheduling of mapreduce workloads on heterogeneous clusters. In: Green Computing Middleware on Proceedings of the 2nd International Workshop, p. 1. ACM (2011)
8. Liu, W., Li, H., Du, W., Shi, F.: Energy-aware task clustering scheduling algorithm for heterogeneous clusters. In: 2011 IEEE/ACM International Conference on Green Computing and Communications (GreenCom), pp. 34–37. IEEE (2011)

9. Li, Y., Liu, Y., Qian, D.: An energy-aware heuristic scheduling algorithm for heterogeneous clusters. In: Proceedings of the 15th International Conference on Parallel and Distributed Systems (ICPADS) (2009)
10. Mukherjee, T., Banerjee, A., Varsamopoulos, G., Gupta, S.K.S., Rungta, S.: Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers. Comput. Netw. **53**(17), 2888–2904 (2009)
11. Liu, L., et al.: Greencloud: a new architecture for green data center. In: Proceedings of the 6th International Conference Industry Session on Autonomic Computing and Communications Industry Session, pp. 29–38. ACM (2009)
12. Beloglazov, A., Buyya, R.: Energy efficient allocation of virtual machines in cloud data centers. In: 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), pp. 577–578. IEEE (2010)
13. Verma, A., Ahuja, P., Neogi, A.: pMapper: power and migration cost aware application placement in virtualized systems. In: Issarny, V., Schantz, R. (eds.) Middleware 2008. LNCS, vol. 5346, pp. 243–264. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89856-6_13
14. Li, Z., Ge, J., Hu, H., Song, W., Hu, H., Luo, B.: Cost and energy aware scheduling algorithm for scientific workflows with deadline constraint in clouds. IEEE Trans. Serv. Comput. **11**, 713–726 (2015)
15. Kusic, D., Kephart, J.O., Hanson, J.E., Kandasamy, N., Jiang, G.: Power and performance management of virtualized computing environments via lookahead control. Cluster Comput. **12**(1), 1–15 (2009)
16. Topcuoglu, H., Hariri, S., Min-you, W.: Performance-effective and low-complexity task scheduling for heterogeneous computing. IEEE Trans. Parallel Distrib. Syst. **13**(3), 260–274 (2002)
17. Zhu, X., Yang, L.T., Chen, H., Wang, J., Yin, S., Liu, X.: Real-time tasks oriented energy-aware scheduling in virtualized clouds. IEEE Trans. Cloud Comput. **2**(2), 168–180 (2014)
18. Chetto, H., Chetto, M.: Some results of the earliest deadline scheduling algorithm. IEEE Trans. Softw. Eng. **10**, 1261–1269 (1989)
19. Schwiegelshohn, U., Yahyapour, R.: Analysis of first-come-first-serve parallel job scheduling. In: SODA, vol. 98, pp. 629–638. Citeseer (1998)