AMSE JOURNALS-AMSE IIETA publication-2017-Series: Advances B; Vol. 60; N°2; pp 442-454
Submitted Mar. 27, 2017; Revised May 28, 2017; Accepted Jun. 12, 2017

Virtual Machine Scheduling Model Based on Energy and Interference Awareness in Cloud Environment

*Xin Sui, **Li Li*, ***Dan Liu, ****Huan Wang, *****Jilong Gong, ******Zetian Zhang

*College of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China;
Jilin Provincial Institute of Education, Jilin 130022, China
**College of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China (ll@cust.edu.cn)
***College of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
****College of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
*****College of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
******College of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China

Abstract

Thanks to the rapid development of cloud datacenter, virtual machine (VM) scheduling has become the key to optimizing energy consumption, service-level agreement, network traffic, etc. Focusing on the optimization of server CPU utilization, energy consumption, network traffic, service performance and so on, the current VM scheduling model often fails to recognize the performance interference between VMs as an optimization parameter. In light of the above, this paper proposes a VM scheduling model, considering both server power consumption and VM performance interference, seeking to lower the energy consumption of the datacenter and the interference between VMs. The experimental results demonstrate that the proposed model outshines the other two models in server CPU utilization, energy consumption, and VM process

time.

Key words

Cloud computing, Energy consumption, VM scheduling, Interference awareness.

1. Introduction

Cloud computing has laid the infrastructure, built the platform and prepared the software (application) for provision of demand-based service to consumers. Depending on their forms, such services are called Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Many computing providers, including Google, Microsoft, Yahoo and the IBM, are competing to set up datacenters around the world to provide cloud computing services.

The cloud datacenters manage to save energy through virtualization. Aside from increasing the CPU utilization of the physical host, the technology makes it possible to deploy multiple virtual machines (VMs) on one physical host, each of which computes various applications as an independent unit. Furthermore, the virtualization helps to merge the application and its execution environment into one entity, and configure the entity based on VM images. The VM images are usually deployed on the VM manager or VM monitor in the hardware-independent manner. Being a software component responsible for hosting a VM, the VM manager serves the upper application services by abstracting the virtualized versions of processors and other system devices (I/O devices, storage, memory, etc.) from physical resources.

Despite the adequate isolation provided by VM managers (e.g. security and failure), performance interference poses a major challenge to the system running at resource-intensive loads. If the same hardware is shared by multiple VMs, the many user mode schedulers will work independently without understanding each other, due to the isolation between the VM manager and the user. In this case, the performance will be less optimal than that of the single operating system running at comparable workloads. The interference between VMs will become increasingly serious in the future, as more and more VMs share the same hardware to improve utilization. Such a problem can be partially alleviated by bettering VM scheduling and reducing virtualization.

The remainder of this paper is organized as follows. Chapter 2 reviews the previous research related to this paper; Chapter 3 defines energy consumption and performance interference; Chapter 4 establishes an energy and interference-aware VM scheduling model; Chapter 5 experimentally compares the proposed model with other models; Chapter 6 discusses the experimental results; Chapter 7 wraps up this research with some valuable conclusions.

2. Literature Review

Much research has been done on VM scheduling, server power consumption and VM performance interference. Some of the most representative studies are listed below.

Predicted the degradation of any possible deployment through only a linear number of measurements [1], and selected the most efficient consolidation pattern to pursue the required performance under resource constraints. An average prediction error of less than 4% was achieved across various benchmark workloads, using Xen VM Manger on Intel Core 2 Duo and Nehalem quad-core processors. It is demonstrated that the prediction technique can achieve better workload deployment for given performance and resource cost objective.

Based on the classification and regression functions of support vector machines, Sam [2] developed a novel approach to cluster and identify several types of applications with distinct performance profiles, and introduced several new scheduling techniques to evaluate the proposed performance models. The evaluation results show that the overall performance of multiplexed workloads is significantly enhanced by integrating such models in the scheduling logic.

Examined the effect of performance interference from the angle of system-level workload features [3]. Two VMs were deployed in a physical host, each of which ran a sample application selected from various benchmark and real-world workloads. For each combination, the performance metrics and runtime features were collected by an instrumented Xen hypervisor. The collected data were then analysed to pinpoint the applications acting as certain types of interference sources, and the mathematical models were created to predict the performance of a new application based on its workload features.

Featuring the scalability to model multi-resource systems and flexibility to represent different policies and cloud-specific strategies, stochastic reward nets (SRNs) were adopted by Dario Bruneo to build an analysis model. The established model was relied on to analyse the behaviour of a cloud datacenter against such metrics as utilization, availability, waiting time, and responsiveness. In the meantime, a resiliency analysis was conducted in consideration of load bursts. Then, a general approach was developed based on system capacity, aiming to help system mangers opportunely set the datacenter parameters under different working conditions.

From robust analytical methods, C. Delimitrou derived the Paragon, an online scalable datacenter scheduler aware of heterogeneity and interference. Instead of profiling each application in details, the Paragon leverages the existing information in the system on the previously scheduled applications. With the aid of collaborative filtering techniques, the scheduler can quickly and accurately classify an unknown incoming workload based on the heterogeneity and interference in multiple shared resources, and the resemblances to the previous applications. The classification allows Paragon to schedule applications in a manner that minimizes interference and maximizes server utilization. After the initial application deployment, Paragon will monitor application behaviour and adjust the scheduling decisions at runtime to avoid performance degradations.

In order to predict application performance, [6] established the mathematical relationship between high-level performance and low-level CPU multiplexing, and designed a synthetic workload with controllable CPU demands to emulate interference workloads in the cloud. The measurements started in a controlled environment to reveal the impact of CPU allocation on application performance. The measured results verified the interdependency between CPU steal time and application performance, and confirmed the effect of the percentage of CPU steal time on application performance, even when the workloads of equal parameters were submitted for processing on the same system platform.

Proposed by Ron C. Chiang, the novel task and resource allocation control framework TRACON mitigates the interference from concurrent data-intensive applications and elevates the overall system performance. Based on the modelling and control techniques from statistical machine learning, the framework consists of three major components: 1) the interference prediction model that infers application performance from the observed resource consumption of different VMs, 2) the interference-aware scheduler designed to apply the model to effective resource management, and 3) the task and resource monitor that collects application features at the runtime for model adaption.

Dug deep into the performance isolation effect of virtualization technology on various microarchitectural resources [8]. The method is to map the Cloud Suite benchmarks to different sockets, different cores of the same chip, and different threads of the same core. Besides, the scholars investigated the correlation between performance variation and resource contention by changing VM mapping policies according to different application features.

Put forward the architecture and principles of energy-efficient cloud computing [9], and, on this basis, presented open research challenges, resource provisioning and allocation algorithms for energy-efficient management of cloud computing environments. In addition, the scholars rolled out the energy-aware resource allocation heuristics for supplying datacenter resources to client applications, and succeeded in delivering the negotiated quality of service (QoS) at improved energy efficiency of the datacenter.

Proposed the MADLVF algorithm to overcome such problems as resource underutilization [10], high energy consumption, and large CO_2 emissions. Comparing the algorithm with the traditional MADRS algorithm, they concluded that the MALVF algorithm outperforms its traditional counterpart in energy consumption and the number of VM migrations.

Probing into the power-efficient and resource-guaranteed VM placement and routing problem (PER-TTA) for dynamically arriving TTA requests, [11] suggested using the multi-component utilization-based power model to minimize the total power consumption, and adopted the Least-Active Most-Utilized policy, which avoids powering on new devices by maximizing the resource utilization of the powered-on devices. In this way, the PER-TTA was transformed into a mixed-integer linear programming (MILP) optimization problem.

3. Methodology

3.1 Energy Consumption

The energy consumption of the physical server can be calculated by CPU utilization, as shown in Equation (1).

$$P_i(u) = r_i * P_i^{\max} + (1 - r_i) * P_i^{\max} * u_i \tag{1}$$

where $P_i(u)$ is the power of the physical server at a certain time; P_i^{max} is the maximum power of server i; r_i was the ratio of minimum power to maximum power of server i; u_i is the CPU utilization of server i.

The energy consumption of server i is expressed in equation (2).

$$E_i = \sum_{t_1}^{t_n} P(u_i(t_j)) \tag{2}$$

where E_i is the energy consumption of server i at time $[t_1, t_n]$; $P\left(u_i(t_j)\right)$ is the power of server i at time t_j.

3.2 Performance Interference

Whereas the single VMs on the same server compete for physical resources, the performance degradation parameter should be introduced to calculate the resource competition interference between the VMs, and the resulting additional VM management.

Suppose VM_i and VM_j are deployed on the same server, $PD_{i,j}$ is used as the performance degradation parameter between the two VMs, and the process time for running only one VM VM_i is denoted by T. Hence, the process time of VM_i is $T * (1 + PD_{i,j})$ when VM_i and VM_j share the same server. Assuming that VM_i sharing the same server with the set VM_{set}, then the performance degradation parameter of VM_i can be expressed as follows:

$$PD_{i,set} = (1 + PD_{i,1}) * (1 + PD_{i,2}) * \ldots * (1 + PD_{i,n}) - 1 \tag{3}$$

where $PD_{i,n}$ is the performance degradation parameter of VM_i and VM_n in the VM set. If the VMs on the same server have the same performance degradation parameters, equation (3) can be rewritten as $PD_{i,set} = (1 + PD)^n - 1$, and the process time of VM_i can be expressed as $T_i^{PD} = Ti * (1 + PD_{i,set})$. In this research, the VM performance degradation is calculated based on the competition time of the VM:

$$C_i^{PD} = a^{\frac{T_i^{PD} - T_i}{T_i}} - 1 \tag{4}$$

where T_i is the process time of VM i running in the server alone; T_i^{PD} is the process time of VM i running in the server with other VMs; C_i^{PD} is the interference cost of VM i.

According to the performance interferences between the VMs in Figure 1, the starting times of VM_1, VM_2 and VM_3 are 0, 0 and 2, respectively. Figure 1 shows that the time for running VM_1, VM_2 and VM_3 alone on the same server is 4, 2 and 2, respectively. Figure 2 assumes the performance degradation parameters between the three VMs as $PD_{1,2} = PD_{2,1} = PD_{1,3} = PD_{3,1} = PD_{2,3} = PD_{3,2} = 1$. Therefore, when the running time varies from 0 to 2, VM_1 and VM_2 run simultaneously on the same server, with VM_1 completing 1/4 of entire task and VM_2 completing 1/2 of entire task. When the running time falls between 2 and 6, VM_1, VM_2 and VM_3 run simultaneously on the same server, with VM_1 completing 1/2 of entire task, VM_2 completing the entire task, and VM_3 completing 1/2 of entire task. When the running time ranges from 6 to 8, VM_1 and VM_3 run simultaneously on the same server, with VM_1 completing 3/4 of entire task, and VM_3

completing the whole task. When the running time fluctuates in the interval [8, 9], only VM_1 runs on the server and completes the entire task.

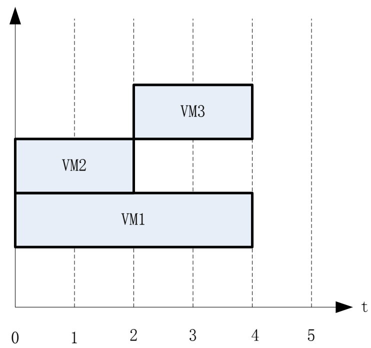

Fig.1. The VM Running alone on the Server

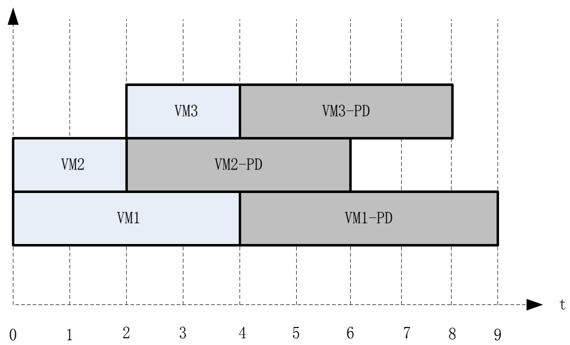

Fig.2. VM Performance Interference

4. VM Scheduling Model

This paper proposes an energy and interference-aware scheduling model for the VMs (EIAVMS), considering both energy consumption and performance interference. In addition to the energy consumption of server, the EIAVMS takes account of the performance interference cost of each VM under the server. The objective function is expressed as:

$$F_1 = \min \sum_{i=1}^{n} \sum_{j=1}^{m} (E_i + b * C_j^{PD}) \tag{5}$$

where n is the number of servers in the cloud datacenter; m is the number of the VMs; E_i is the energy consumption of server i; C_j^{PD} is the performance interference cost of VM_j on server i; b is the weight of VM performance interference cost.

Therefore, the purpose of VM scheduling is to minimize energy consumption and performance interference cost:

$$\left\{ \begin{array}{l} \sum_{j=1}^{m} V_j^{cpu} * P_{i,j} < C_i^{cpu} \\ \sum_{j=1}^{m} V_j^{mem} * P_{i,j} < C_i^{mem} \\ \sum_{j=1}^{m} V_j^{store} * P_{i,j} < C_i^{store} \end{array} \right\} \tag{6}$$

where V_j^{cpu}, V_j^{mem} and V_j^{store} are the CPU capacity, the memory capacity, and the hard drive capacity requested by the VM_j, respectively; C_i^{cpu}, C_i^{mem} and C_i^{store} are the CPU capacity, the memory capacity, and the hard drive capacity of server I; $P_{i,j} = (0,1)$ is the deployment of VM_j on the server i.

In essence, equation (6) sets out the constraint on VMs allocation: the sum of CPU capacity, memory capacity, and hard drive capacity requested by VMs located on the server should not exceed the sum of such capacities of the server.

5. Simulation and Comparison

In this research, the CloudSim platform was adopted to simulate the cloud computing environment, and the proposed EIAVMS model was analysed and compared with the IQR and MAD models in terms of CPU utilization, energy consumption and VM process time.

5.1 Parameter Setting

The experiments were performed in the cloud datacenter consisting of 500 servers and 500 VMs. The configuration parameters of the servers and the VMs are presented in Tables 1 and 2, respectively.

Tab.1. Server Configuration Parameter

Parameter	Host1	Host2
MIPS	1860	2660
PES(Number)	2	2
RAM(MB)	4096	4096
BW(Gbit/s)	1	1
STORAGE(TB)	1	1

Tab.2. Vm Configuration Parameter

Parameter	Vm1	Vm2	Vm3	Vm4
MIPS	2500	2000	1000	500
PES(Number)	1	1	1	1
RAM(MB)	870	1740	1740	613
BW(Mbit/s)	100	100	100	100
STORAGE(TB)	2.5	2.5	2.5	2.5

5.2 Results Analysis

1. The average CPU utilization at different numbers of servers and VMs: to obtain the average CPU utilization of the three models (IEAVMS, IQR, MAD), the first experiment was conducted with different numbers of servers and VMs on the cloud platform. The result shows that the average

CPU utilization of the EIAVMS fell between 40% and 45%, while that of the IQR and MAD ranged from 35% to 38%. The proposed model is obviously more effective than the other two models in average CPU utilization (Figure 3).

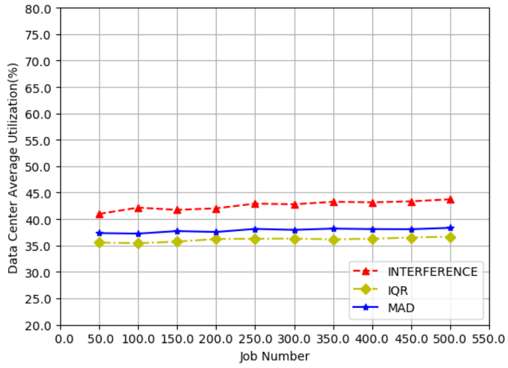

Fig.3. Average CPU Utilization of Datacenter at Different Numbers of Servers and VMs

2. The average CPU utilization at different times with a fixed number of servers and VMs: to test the average CPU utilization of the three models at different times, the second experiment was carried out with a certain number of servers and VMs on the cloud platform. According to the experimental results, the EIAVMS still maintained a higher average CPU utilization (35%~70%) than the IQR and MAD models (28%~60%) (Figure 4). Moreover, the average CPU utilization of the EIAVMS model always stayed in a relatively middle position, indicating that the CPU utilization is neither too high nor too low. This is because the EIAVMS model can effectively optimize the performance interference problem between the VMs.

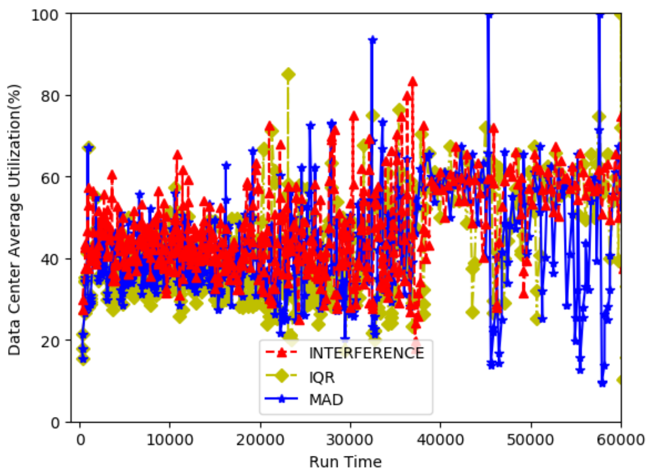

Fig.4. Average CPU Utilization of Datacenter at Different Times

3. The energy consumption of cloud datacenter with different numbers of servers and VMs: to check the datacenter energy consumption of the three models, the third experiment was performed with different numbers of servers and VMs on the cloud platform. It is observed that the proposed model consumed 7.23%~10.21% less energy than the IQR model, and 7.23%~10.21% less energy than the MAD model (Figure 5).

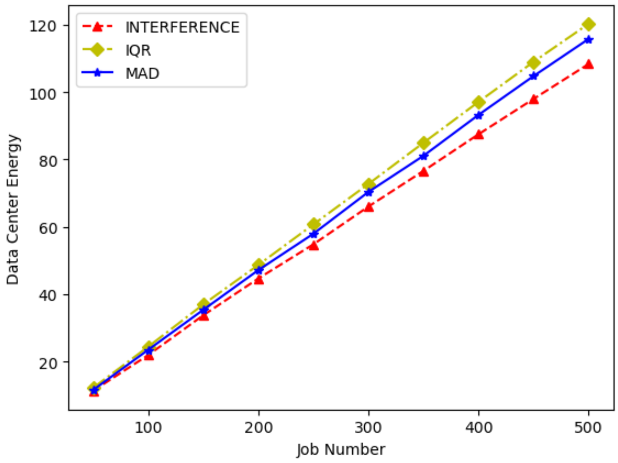

Fig.5. Energy Consumption of Datacenter at Different Numbers of Servers and VMs

4. The average VM process time with different numbers of servers and VMs: to ascertain the average VM process time of the three models, the fourth experiment was implemented with different numbers of servers and VMs on the cloud platform. It can be seen that the average VM process time of the EIAVMS model is shorter than the IQR and MAD models in most cases. The time-saving phenomenon is attributable to the consideration of both energy consumption and performance interference in the proposed model. Hence, a low energy consumption of the cloud datacenter and a high CPU utilization can shorten the VM process time (Figure 6).

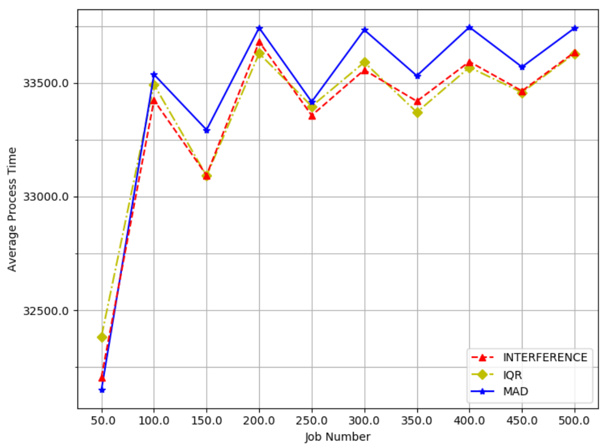

Fig.6. Average process time of VMs at different numbers of servers and VMs

5. The process time of different VMs with a fixed number of servers and VMs: to identify the VM process time of the three models, the fifth experiment was executed with a fixed number of servers and VMs on the cloud platform. It is concluded that the proposed model boasts a shorter VM process time and better effect than the other two models in most cases (Figure 7).

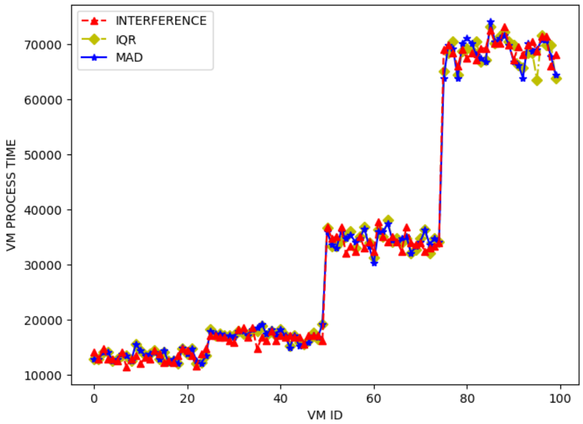

Fig.7. VM Process Time at a Fixed Number of Servers and VMs

6. Discussion

Probing into the problem of server energy consumption and VM performance interference, the author developed the VM scheduling model EIAVMS to optimize the server energy consumption and VM performance interference. Considering both the energy consumption of server and the interference between the VMs on a single server, the proposed model reduced the energy consumption of the server, improved the average CPU utilization of the server, shortened the average process time of the VM, and prevented CPU utilization of individual servers from highlighting.

Conclusion

This paper is an indepth exploration of server energy consumption and interference between VMs. The VM scheduling model EIAVMS was proposed to optimize server energy consumption and VM performance interference. Then, the cloud computing software Cloud Sim was introduced to compare the EIAVMS, IQR and MAD models in terms of average server CPU utilization, energy consumption and VM process time. The experiment results prove that the proposed model is more efficient than the IQR and MAD models in all three aspects.

Acknowledgment

This work was supported in part by the science and technology project of "13th Five-Year" planning of the Education Department of Jilin Province (JJKH20170631KJ); the key project of "13th Five-Year" planning of the Education Science of Jilin Province (ZD16024); and the Key Science and Technology Project of Jilin Province (20160204019GX).

References

1. S. Govindan, J. Liu, A. Kansal, A. Sivasubramaniam, Cuanta: Quantifying effects of shared on-chip resource interference for consolidated virtual machines, 2011, Acm Symposium on Cloud Computing, New York, USA.
2. S. Verboven, K. Vanmechelen, J. Broeckhove, Black box scheduling for resource intensive virtual machine workloads with interference models. 2013, Future Generation Computer Systems, vol. 29, pp. 1871-1884.
3. Y. Koh, R. Knauerhase, P. Brett, M. Bowman, Z.H. Wen, C. Pu, An analysis of performance interference effects in virtual environments, 2007, IEEE International Symposium on Performance Analysis of Systems & Software, CA, USA.
4. D. Bruneo, A stochastic model to investigate data center performance and QoS in IaaS cloud computing systems, 2014, IEEE Transactions on Parallel and Distributed Systems, vol. 25, pp. 560-569.
5. C. Delimitrou, C. Kozyrakis, QoS-aware scheduling in heterogeneous datacenters with paragon, 2013, ACM Transactions on Computer Systems, vol. 31, New York, USA.
6. A.O. Ayodele, J. Rao, T.E. Boult, Performance measurement and interference profiling in multi-tenant clouds, 2015, IEEE 8th International Conference on Cloud Computing, New York, USA.
7. R.C. Chiang, H.H Huang, TRACON: Interference-aware scheduling for data-intensive applications in virtualized environments, 2014, IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 5, pp. 1-12.
8. T.N. Xu, X.F. Sui, Z.C. Yao, J.Y. Ma, Y.G. Bao, L.X. Zhang, Rethinking virtual machine interference in the era of cloud applications, 2013, IEEE International Conference on High Performance Computing and Communications, Zhangjiajie, China.
9. A. Beloglazov, J. Abawajy, R. Buyya, Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing, 2012, Future Generation Computer Systems, vol. 28, pp. 755-768.

10. J.K. Verma, C.P. Katti, P.C. Saxena, MADLVF: An energy efficient resource utilization approach for cloud computing, 2014, International Journal of Information Technology and Computer Science, vol. 6, no. 7-8, pp. 56-64.
11. A. Dalvandi, M. Gurusamy, K.C. Chua, Power-efficient resource-guaranteed VM placement and routing for time-aware data center applications, 2015, International Journal of Computer and Telecommunications Networking, New York, USA, vol. 88, pp. 249-268.