Game Theoretic Task Allocation to Reduce Energy Consumption in Containerized Cloud

Manoj Kumar Patra, Dimple Patel, Bibhudatta Sahoo, Ashok Kumar Turuk
Department of Computer Science & Engineering
National Institute of Technology
Rourkela, India
{manojpatra.in, dimplepatel982, bibhudatta.sahoo, akturuk}@gmail.com

***Abstract*—Cloud computing provides information technology based solutions to the end-users as a utility. Virtual machine or the virtualization technology is the backbone of implementing cloud computing technologies. However, such implementation encounters the problem of tremendous energy consumption. One of the foremost issues in implementing cloud computing is high energy consumption. This can be reduced to some extent by proper allocation and efficient utilization of resources. At present, containerization is one of the broadly discussed techniques as an alternative to traditional virtualization solutions. In this paper, we propose a game-theoretic approach for resource allocation and a containerized cloud architecture which drastically reduces energy consumption than a virtual machine based cloud. We have used Google cluster traces data set for our experiment in the cloud with virtual machine and containerized cloud. Experimental results show that the energy consumption is minimized in the containerized cloud than a cloud with virtual machines.**

***Index Terms*—Cloud Computing, Container, Virtual Machine, Game Theory, Virtualization Technique, Resource Allocation**

I. Introduction

Cloud computing provides different types of computing resources using virtualization techniques over the internet. It allows the users to access data on demand from a data center or a centralized server. Data centers are one of the most significant components in cloud computing which include connectivity, cabling system, UPS, servers, network, etc. The operations and maintenance of such components consume a significant amount of energy in the cloud data center. Inappropriate use of resources makes servers idle and consumes a huge amount of energy without doing any work. Proper utilization of resources reduces the energy consumption in cloud data centers. There are different ways we can optimize or reduce energy consumption such as proper scheduling of jobs keeping the server busy all the time, distributing the work to all available servers equally called as load balancing, moving a running application or virtual machine(VM) between different servers called VM migration and proper use of resources.

The traditional virtualization technique in cloud computing uses a virtual machine that provides hardware virtualization. Virtual machines run on top of a physical machine using a hypervisor and hypervisor run on a host machine. Even though real-time energy consumption is reduced in the virtualization cloud, the amount of energy consumed by the hypervisor running on top of the host machine is quite high and has not been understood so far [1]. On the other hand a container, unlike a virtual machine that virtualizes the hardware, provides software level virtualization such as OS-level virtualization by abstracting the “user space”.

In this paper, we propose an architecture model for containerized cloud and a game-theoretic approach for resource allocation to reduce the overall energy consumption. The proposed game-theoretic approach is applied in both VM based cloud and Containerized cloud. The experimental results show that the containerized cloud consumes less energy than a VM based cloud. Furthermore, we implemented a few algorithms in containerized cloud and compared the result of our proposed technique with the best one. To select the optimal number of VM, Best Fit Descending algorithm has been used. The performance of the proposed algorithm is better in terms of energy consumption.

The rest of the paper is organized as follows: in Section-II, an extensive review has been done as Motivation and Related Work . Section-III describes the system model, which includes a containerized framework, architecture model, description of data set and game theoretic approach for resource allocation. In Section-IV, the experimental results are presented and finally in Section-V, with some observations of the work, a conclusion is drawn.

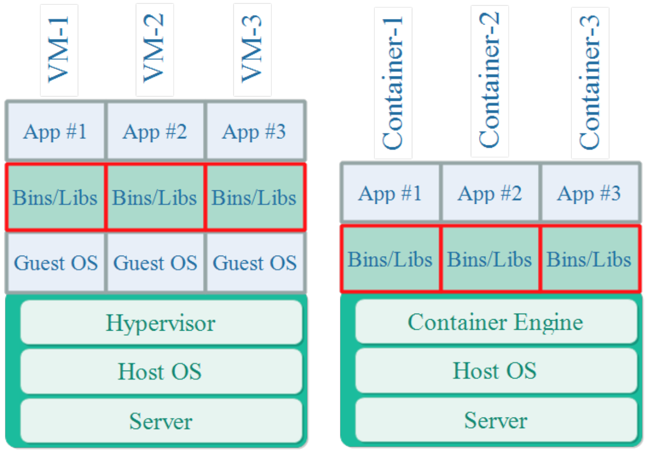
II. Motivation and Related Work

In this section, a comparative study of virtual machines and containers, the difference in their architectural model and a literature review are presented.

A. *Virtual Machine Vs Container*

The virtual machines and containers both do not require physical hardware, resulting in better utilization of computing resources, both in terms of cost-effectiveness and energy consumption. The architectural difference between them is as follows and diagrammatically described in Fig. 1.

The virtual machine architecture has a fairly ubiquitous virtual hardware layer between the virtual machine and host OS called hypervisor. The hypervisor is responsible for interacting with all different types of NIC cards from all types of hardware and storage devices in physical machine [2]. The server provides resources to the virtual machines which include RAM, CPU, etc. The size of the virtual machine need not be the same if a virtual machine is running a heavy



(a) VM Architecture (b) Container Architecture

Fig. 1. Virtual Machine Vs Container

application, the hypervisor will allocate more resources to that VM than others. Every virtual machine has its CPU, storage, virtualized network adapter and full-fledged guest operating system making it heavily weighted.

On the other hand, a container provides OS-level virtualization by abstracting the user space. It is not required to install a separate operating system in every container. A large benefit of a container is the ability to package a lot of library and binary files in a container, an application needs from the host operating system or server [3].

The advantage of using container over virtual machine can also be used in areas like Internet of Things, Fog Computing and smart city applications. A model of architecture proposed in [4] can be improved by using container at the server end and hence performance can be improved.

B. Related Work

Recent research on energy consumption in cloud computing has used several different techniques such as virtual machine, virtual machine consolidation and migration, dynamic voltage and frequency scaling(DVFS), etc.

In [5], authors have proposed a joint optimization technique for delay and energy consumption in the cloud to thing continuum. The technique takes delay and a weighted sum of energy consumption as an objective function. They have proposed an algorithm which is adopted by fog node, to optimize power transmission and frequency of processing. G. Prasad Babu and A.K. Tiwari in [6], proposed a scheduling algorithm using a prediction model for energy efficiency in cloud computing systems. The proposed algorithm is having two main components: an iterative fractal model-based prediction model and an improved heuristic algorithm based scheduler. The scheduling algorithm reduces energy consumption while producing the same QoS. OHSA Ahvar, Anne-C´ecile Orgerie and Adrien Lebre in [7], proposed a model to estimate energy consumption in different infrastructures like cloud, fog, and edge. The model considers all types of energy consumption including energy consumed by network and cooling systems.

In [8], Hua Peng, Wu-Shao Wen, Ming-Lang Tseng, and Ling-Ling Li proposed a scheduling scheme to optimize the energy consumption of mobile devices in the cloud. They have considered a sequence of task execution, position of task execution and operating frequency and voltage of mobile node and used whale optimization algorithm based on dynamic voltage and frequency scaling technique to optimize energy consumption. In [9], Elhadj Benkhelifa, Thomas Welsh, Loai Tawalbeh, Yaser Jararweh, and Anas Basalamah proposed a system model for profiling the usage of energy per application in a mobile cloud, so that the predictive model will predict the future consumption and decide whether it is economically viable or not.

Zhou Zhou et al. [10], presented an adaptive energy-aware algorithm to reduce power consumption in a cloud data center with minimal Service Level Agreement (SLA) violation. Their approach considers the types of the application running on a virtual machine and memory resources during deployment of a virtual machine. Huda Ibrahim, Raafat O, Aburukba and Khaled El-Fakih in [11], proposed an Integer Linear Programming (ILP) model and a dynamic task scheduling algorithm to reduce consumption of energy in cloud data centers. Furthermore, they proposed an Adaptive Genetic Algorithm (GA) to illustrate the dynamic nature of the cloud environment. Awada Uchechukwu, Keqiu Li and Yanming Shen in [12], investigated the patterns of energy consumption and proposed a methodology to minimize the energy consumption in cloud data centers.

Hancong Duan, Chao Chen, Geyong Min and Yu Wu in [13], presented PreAntPolicy energy-aware scheduling technique that uses a prediction model. The model is based on ant colony algorithm and fractal mathematics. The job of the prediction model is to find out whether to trigger the execution of the scheduler by load trend prediction in a heterogeneous cloud computing system. Nidhi Jain Kansal and Inderveer Chana in [14], presented an energy-aware technique for virtual machine migration in the cloud computing environment based on Firefly algorithm. The technique describes how to migrate a virtual machine that is heavily loaded while maintaining the same energy efficiency and performance.

Feifei Chen, John Grundy, Yun Yang, Jean-Guy Schneider and Qiang He in [15], have proposed a model for energy consumption in the cloud computing system and conducted extensive experiments to make the model operational. They have used three different types of tasks: communication intensive, data-intensive and computation-intensive tasks. Weiwei Lin et al. in [16], provides the measurement system for energy consumption in a heterogeneous cloud server environment. A Distributed Energy Meter system has been proposed that can estimate the energy consumption in a heterogeneous cloud environment and also support many CPU power, consumption model. Congfeng Jiang et al. in [17], have used different workloads, explored the power and energy characteristics of several hypervisors to emulate realistic multi-tenant cloud environments.

III. System Model

This section describes the data set in III-A, describes the architecture of the proposed system model in III-B, and a game-theoretic approach for resource allocation in III-C and an energy model in III-D.

A. Description of Dataset

We have used Google Cluster Trace dataset [18] for our experiment, which is publicly available by Google. It contains information about the cell of about 29 days. Each cell consists of a cluster of several machines. Each job in the dataset composed of several tasks where each task may be associated with many processes in a single machine. If we consider the Google Cluster Trace dataset as a relational model, it consists of six relations: Machine Events Relation, Machine attribute's Relation, Jobs Events Relation, Task Events Relation, Task Constraints Relation, and Task Resource Usage Relation. A relation is a set of tuples, or rows in a table, with each tuple sharing a set of attributes or columns. Following is a brief description of each of the relation:

- Machine Events Relation:
  This relation describes each machine in a cell. The attributes of machine events are Timestamp, which stores the time at which the system was started. MachineID, represent the identity of a machine. Three event types: ADD(0) make a machine available to a cluster, REMOVE(1) remove a machine from a cluster, and UPDATE(2) make a machine available to the cluster, had its available resources changed. The chipset version of the machine and microarchitecture is represented by PlatformId. Capacity represents the size of the RAM.
- Machine Attributes Relation:
  It describes the properties of a machine such as a clock speed, version of kernel and presence of any external IP address. It contains several attributes such as Timestamp, MachineID, Attribute name, Attribute value and Attribute deleted which is a boolean value represent whether an attribute was deleted or not.
- Jobs Events Relation:
  This relation consists of total eight attributes, Timestamp, Missing info, JobID represents the unique ID of a job, Event type which represents the state at which the job is in, User name, the sensitivity of latency for a job is represented by Scheduling class, Job name, and Logical job name.
- Task Events Relation:
  This relation consists of total 13 attributes, Timestamp, Missing info, JobID, Task Index represent the index of a task within a job, Event type, User name, Scheduling class, Priority of a task used in scheduling, Resource request for CPU core, Resource request for RAM, Resource request for local disk space, and Different-machine constraint which is a boolean value and if this represents true, task must be scheduled to execute in different machine.

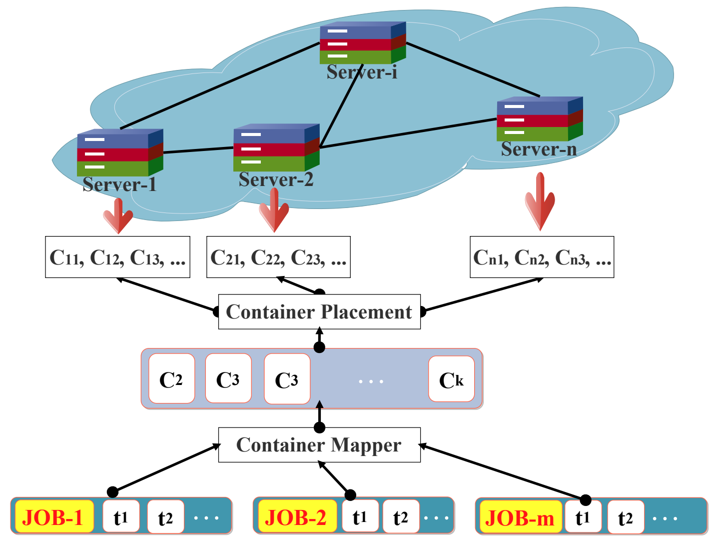


Fig. 2. Architecture Model

- Task Constraints Relation:
  Task constraint relation provides information about the restriction of a machine in which it can run. This may contain zero or more task constraints. Attributes of this relation are Timestamp, JobID, TaskIndex, Attribute name, Attribute value and Comparision operator which can be $<, >, =,$ and $! =$.
- Task Resource Usage Relation:
  This contains the information about the usage of resources in every 5 minutes. It consists of a total of 20 attributes, few of them are CPU usage, memory usage, assigned memory, maximum memory usage, disk I/O time, CPU rate, cycle per instruction, sampling rate and aggregation type, remaining are already described above.

B. Architecture Model

The proposed architecture model has three main components, that are Job created by users, Container Mapper and Cloud Server. Jobs are created by users and each job is having several independent tasks. The attributes of a job are arrival time(AT), size in terms of Million Instruction(MI), and completion time(CT). The details of the data set are described in section III-C. Jobs are submitted to container mapper based on their completion time. The job with earlier completion time gets scheduled first.

Then, the container mapper is responsible for mapping tasks into respective containers. Once tasks acquire the container, the process of placing containers into the virtual machine starts. Container placement is an NP-hard combinatorial problem that provides a certain placement of a non-overlapping container on a server in such a way that it minimizes the number of the server. Heuristic and meta-heuristic algorithms are used for such a problem. This algorithm gives a near-optimal solution to the problems. Different heuristic algorithms are,

- First Fit: This approach tries to place the task in the very first container of any server that can accommodate the task.
- First Fit Descending: This will arrange the tasks in descending order and then works same as First fit.
- Next Fit: This approach first tries to fit the task in the current server, if unsuccessful, it assigns a new container in a different server for the task to be mapped.
- Best Fit: This technique tries to fit the current task in the server which will have the minimum remaining capacity, if the task is placed in that server.
- Best Fit Descending: This will arrange the tasks in descending order and then works same as Best fit.

We have implemented all five techniques for efficiently placing a container on a cloud server to minimize the number of servers and hence reducing the energy consumption. While experimenting, we varied the number of tasks from 400 to 1200, applied the algorithm, and compared them. Then, we took the average of results for deciding the best algorithm for our problem. Our experimental results show that Best Fit Descending is performing better than other algorithms.

C. Game Theoretic Approach for Resource Allocation

This section, a game-theoretic approach to mapping tasks with virtual machines and containers has been presented. A game requires participation of a set of players $P = \{p_1, p_2, p_3, ...\}$ who make strategic decision during play, a set of strategies $S = \{s_1, s_2, s_3, ...\}$ for all players, and a payoff that represent the gain or loss of a player. Different attributes of our two-player game model are described below.

- **Strategy**: In VM based cloud, each VM is considered as a strategy of the game. A set of VM, $V = \{v_1, v_2, v_3, ...\}$ is responsible for executing users task. Each VM has its speed of executing user task $sp_j, j = \{1, 2, 3, ...\}$. Similarly in the containerized cloud, each server act as a strategy. A set of Server, $S = \{S_1, S_2, S_3, ...\}$ is responsible for providing execution environment for user task in a container.
- **Player**: Each task is considered as player of the game. A task has three attributes, arrival time(at_i), task size(ts_i), and deadline(d_i), $i = \{1, 2, 3, ...\}$, where i represent i^{th} task.
- **Payoff**: Payoff quantifies the gain or loss of a player based on outcome of the game.
- **Payoff Matrix**: Payoff matrix, $PM_{\text{m*n}}$ is a tabular collection of data in which each row represents the strategy of player-1, also called as a pure strategy of player-1 and each column represents the strategy of player-2, also called as a pure strategy of player-2. Each entry in the payoff matrix is having two values, the first one represents the payoff of player-1 and the second one represents the payoff of player-2.

To apply game theory on the payoff matrix, first, the payoff matrix is converted into a normalized payoff matrix $P = (p_{\text{mn}})$ by subtracting the payoff of player-2 from player-1. A positive value in the normalized payoff matrix indicates, gain to player-1 and loss to player-2. Similarly, a negative value in the normalized payoff matrix indicates, gain to player-2 and loss to player-1. Player-1 follows the maximin principle to maximize the minimum guaranteed gain. This principle evaluates the worst possible outcome for each option and selects the best among all. Similarly, player-2 follows the minimax principle to minimize the maximum loss. It evaluates the best possible outcome for each option and selects the worst among all.

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1\text{n}} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2\text{n}} \\ p_{31} & p_{32} & p_{33} & \cdots & p_{3\text{n}} \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ p_{\text{m}1} & p_{\text{m}2} & p_{\text{m}3} & \cdots & p_{\text{mn}} \end{bmatrix}$$

Now, consider the game with normalized payoff matrix $P = (p_{\text{ij}})$ from player-1's point of view, player-1 will follow the minimax principle so, it will find the maximum values of each row

$$Maximize\ p_{\text{ij}}\ over\ j = 1, 2, .., n$$

and then select the minimum one as p_1 to minimize the maximum loss.

$$p1 = \min_{i=1,\dots,m} \max_{j=1,\dots,n} p_{\text{ij}}$$

Now, from player-2's point of view, player-2 will follow the maximin principle so, it will find the minimum values of each column

$$Minimize\ p_{\text{ij}}\ over\ i = 1, 2, .., m$$

and then select the maximum one as p_2 to maximize the minimum gain.

$$p2 = \max_{j=1,\dots,n} \min_{i=1,\dots,m} p_{\text{ij}}$$

The minimum value for player-1 $= p_1$ and maximax value for player-2 $= p_2$. The corresponding row and column giving $p1$ and $p2$ are the selected strategy for player-1 and player-2 respectively.

D. Energy Consumption Model

Assumptions: The job with its requirement and dependencies are specified. Each of these jobs is divided into atomic task and this task is mapped to container. So, for each job, we have a set of container $\{c_1, c_2, \dots c_n\}$.

Let $x_i(k)$ denotes the workload of container i on k^{th} host and there are n number of containers on k^{th} node, then

$$Workload(k) = \sum_{i=1}^{n} x_i(k) \tag{1}$$

Total workload on k^{th} host must be less than or equal to the capacity of the k^{th} host i.e.

$$\sum_{i=1}^{n} x_i(k) <= Capacity(k) \tag{2}$$

Let c_{ij} represents the i^{th} container for job j where $i = 1, 2, 3, ..., n$ and r_{ij} is the amount of resources required by container i of job j then,

$$\sum_{i=1}^{n} r_{ij} \leq R_j \tag{3}$$

Where R_j, is the total required resources for job j.

We consider the power utilization of the CPU for estimation of power consumption as this is the main component that presents the largest variance in power consumption in regards to its utilization rate [19]. The correlation between server utilization and the electric power consumption proposed in [20], i.e.

$$P_k(u) = (P_{k,max} - P_{k,idle}) \cdot u + P_{k,idle} \tag{4}$$

where, u is utilization rate represented by,

$$u = \frac{\sum_{i=1}^{n} x_i(k)}{Capacity(k)} \tag{5}$$

The total power consumption is given by,

$$P = \sum_{k=1}^{m} P_k(u) \tag{6}$$

the objective is to

$$minimize \sum_{k=1}^{m} P_k(u) \tag{7}$$

subjected to following three constraints

$$Workload(k) = \sum_{i=1}^{n} x_i(k) \tag{8}$$

$$\sum_{i=1}^{n} x_i(k) <= Capacity(k) \tag{9}$$

$$\sum_{i=1}^{n} r_{ij} \leq R_j \tag{10}$$

IV. Experimental Results

We have performed our experimentation with virtual machine based cloud and containerized cloud using the two-player game described in III-C. In virtual machine based cloud, completion time is considered as a payoff and in containerized cloud load factor as a payoff, the player will always choose the server with a lesser load factor. Both the evaluation process is described below.

- Virtual Machine Based Cloud: The key component of this approach is a virtual machine, speed of the virtual machine, task size, arrival time of the task, and deadline or completion time of the task. Tasks are sorted in increasing order of their completion time. In our two-player game model, to create a payoff matrix, two tasks are taken up at a time and calculate their payoff for different VM based on the speed of VM. Then a normalized payoff matrix is created and the game model was applied.

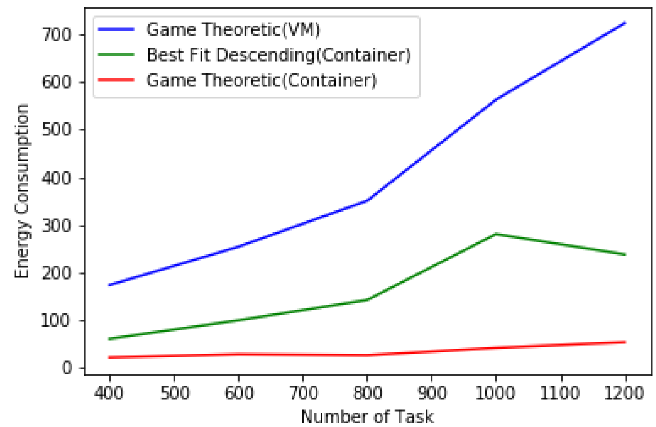


Fig. 3. Total Energy Consumption

- Containerized Cloud: To apply the proposed game model in the containerized cloud, first, the tasks are sorted in descending order of their deadline. First 20% of the tasks are assigned to all the servers sequentially in the FCFS basis and remaining tasks are assigned to different servers by applying the proposed game model.

First, we apply the best fit descending algorithm to fix the number of servers. Then, we apply the proposed game-theoretic model by taking the same number of servers. The game model was also applied by taking the same number of virtual machines in VM based cloud. We compared the result of the proposed game-theoretic model with VM based cloud and containerized cloud with the best fit descending algorithm. The experimental results show that the containerized cloud consumes less energy than VM based cloud and our proposed game-theoretic model performs better than first fit descending.

The experimental results in Fig-3, show that the containerized cloud consumes less energy than a virtual machine based cloud and our proposed game-theoretic model in containerized cloud performs better than best fit descending in the containerized cloud.

V. Conclusion

This paper presents a game-theoretic mechanism for resource allocation in a containerized cloud to reduce energy consumption. We analyze the total energy consumption of VM based cloud and a containerized cloud and found that a containerized cloud always consumes less energy. Several algorithms has been implemented in the containerized cloud such as First Fit, First Fit Descending, Next Fit, Best Fit, and Best Fit Descending. It is found that the Best Fit Descending is performing better than all others. The Best Fit Descending is used to fix the number of servers in the containerized cloud and VM in a virtual machine based cloud. We observed that, in VM based cloud with a fixed number of virtual machine, energy consumption goes on increasing with increasing the number of tasks. The proposed approach is compared with

Best Fit Descending with the same number of servers and the same numbers of virtual machines in VM based cloud. The experimental results show that the proposed resource allocation technique consumes significantly less amount of total energy than other approaches in the containerized cloud and VM based cloud.

REFERENCES

[1] Jiang, Congfeng, Yumei Wang, Dongyang Ou, Youhuizi Li, Jilin Zhang, Jian Wan, Bing Luo, and Weisong Shi. "Energy efficiency comparison of hypervisors." Sustainable Computing: Informatics and Systems (2017).

[2] Shirinbab, Sogand, Lars Lundberg, and Emiliano Casalicchio. "Performance Comparision between Scaling of Virtual Machines and Containers using Cassandra NoSQL Database." CLOUD COMPUTING 2019 (2019): 103.

[3] Pahl, Claus. "Containerization and the paas cloud." IEEE Cloud Computing 2, no. 3 (2015): 24-31.

[4] Patra, Manoj Kumar. "An architecture model for smart city using Cognitive Internet of Things (CIoT)." In 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), pp. 1-6. IEEE, 2017.

[5] Wei, Xianglin, Chaogang Tang, Jianhua Fan, and Suresh Subramaniam. "Joint Optimization of Energy Consumption and Delay in Cloud-to-Thing Continuum." IEEE Internet of Things Journal 6, no. 2 (2019): 2325-2337.

[6] Babu, G. Prasad, and A. K. Tiwari. "Energy Efficient Scheduling Algorithm for Cloud Computing Systems Based on Prediction Model." International Journal of Advanced Networking and Applications 10, no. 5 (2019): 4013-4018.

[7] Ahvar, Ehsan, Anne-Cécile Orgerie, and Adrien Lebre. "Estimating Energy Consumption of Cloud, Fog and Edge Computing Infrastructures." IEEE Transactions on Sustainable Computing (2019).

[8] Peng, Hua, Wu-Shao Wen, Ming-Lang Tseng, and Ling-Ling Li. "Joint optimization method for task scheduling time and energy consumption in mobile cloud computing environment." Applied Soft Computing 80 (2019): 534-545.

[9] Benkhelifa, Elhadj, Thomas Welsh, Loai Tawalbeh, Yaser Jararweh, and Anas Basalamah. "User profiling for energy optimisation in mobile cloud computing." Procedia Computer Science 52 (2015): 1159-1165.

[10] Zhou, Zhou, et al. "Minimizing SLA violation and power consumption in Cloud data centers using adaptive energy-aware algorithms." Future Generation Computer Systems 86 (2018): 836-850.

[11] Ibrahim, Huda, Raafat O. Aburukba, and Khaled El-Fakih. "An integer linear programming model and adaptive genetic algorithm approach to minimize energy consumption of cloud computing data centers." Computers & Electrical Engineering 67 (2018): 551-565.

[12] Uchechukwu, Awada, Keqiu Li, and Yanming Shen. "Energy consumption in cloud computing data centers." International Journal of Cloud Computing and Services Science (IJ-CLOSER) 3, no. 3 (2014): 145-162.

[13] Duan, Hancong, Chao Chen, Geyong Min, and Yu Wu. "Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems." Future Generation Computer Systems 74 (2017): 142-150.

[14] Kansal, Nidhi Jain, and Inderveer Chana. "Energy-aware virtual machine migration for cloud computing-a firefly optimization approach." Journal of Grid Computing 14, no. 2 (2016): 327-345.

[15] Chen, Feifei, John Grundy, Yun Yang, Jean-Guy Schneider, and Qiang He. "Experimental analysis of task-based energy consumption in cloud computing systems." In Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering, pp. 295-306. ACM, 2013.

[16] Lin, Weiwei, Haoyu Wang, Yufeng Zhang, Deyu Qi, James Z. Wang, and Victor Chang. "A cloud server energy consumption measurement system for heterogeneous cloud environments." Information Sciences 468 (2018): 47-62.

[17] Jiang, Congfeng, Yumei Wang, Dongyang Ou, Youhuizi Li, Jilin Zhang, Jian Wan, Bing Luo, and Weisong Shi. "Energy efficiency comparison of hypervisors." Sustainable Computing: Informatics and Systems (2017).

[18] Reiss, Charles, John Wilkes, and Joseph L. Hellerstein. "Google cluster-usage traces: format+ schema." Google Inc., White Paper (2011): 1-14.

[19] Blackburn, Mark, and G. Grid. "Five ways to reduce data center server power consumption." The Green Grid 42 (2008): 12.

[20] Feller, Eugen, Louis Rilling, and Christine Morin. "Energy-aware ant colony based workload placement in clouds." In Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing, pp. 26-33. IEEE Computer Society, 2011.