Energy Aware Linear Programming Model for Task Scheduling in Cloud Computing

G. Sreenivasulu
Department of Computer Science
Bharathier university,
Coimbatore,
Tamilnadu,
India
sreenumitsg@gmail.com

IlangoParamasivam
Professor,
Department of CSE,
PSG institute of Technology and Applied Research,
Coimbatore, Tamilnadu, India
dr.p.ilango@gmail.com

Abstract:

One of the advantages of cloud computing is its ability to perform the tasks in the data centers. However, the energy consumption of these facilities is a major issue. This paper presents a linear programming model that takes into account the matrix formation to allocate the tasks to the servers. The proposed model achieves the efficiency of reducing the data center's energy consumption by implementing the greedy approach. The paper presents a linear model that takes into account the matrix formation to allocate the tasks to the servers. It achieves the efficiency of reducing the data center's energy consumption by implementing the greedy approach.

Keywords: Task Allocation, Cloud Computing, Energy, Linear approach, Data Center

DOI Number: 10.14704/nq.2022.20.12.NQ77012 **NeuroQuantology 2022; 20(12): 110-119**

1. Introduction

The rise of cloud computing has created a paradigm shift in the way computing is done [1]. It allows organizations to access a variety of resources and services, such as storage, without having to build a huge infrastructure. Most of the services offered by the cloud are used in data centers, which are usually equipped with large number of servers and storage units. The operational cost of the cloud is usually diverted to the energy expenses [2-3].

Various studies are conducted on the energy consumption of data centers. One of the most common techniques used to reduce the energy consumption is by implementing a strategy that involves reducing the number of network traffic and the start-up servers [4-8]. This can be done through the use of a combination of techniques such as power dissipation [10]. In order to identify the optimal cooling zones, a framework has been developed that uses a combination of temperature sensing and loosely coupled models [9].

The introduction of pricing models in real time electricity, which is focused on generating utility function, was recently proposed [11]. This mechanism deals with the differences between prices of electricity for different regions and to regulate the load over the data centers to increase the profit. Server and virtual machine configuration has been studied [12].

Many algorithms are proposed to reduce the consumption of power in data centers [5]. The authors in [6] concentrated on QoS parameters to assign the VMs to the data centers to reduce the consumption of energy. In [7], the authors proposed the VM provisioning for data centers to analyse the consumption of energy in different data centers. In [8], the authors developed the power aware model to guarantee that the VMs are operated at low temperature. In [13], the authors proposed the Lyapunov

eISSN 1303-5150

optimization for dynamic allocation of resources. This mechanism studies the time varying workloads and their impacts in the resource allocation.

The paper begins with a summary of the main ideas and techniques involved in the development of energy aware model. Section 2 explains about the related work regarding scheduling approaches in cloud. Section 3 explains the mathematical model for task assignment in cloud. Section 4 proposed about the energy aware model for data centers. Section 5 explains about the Greedy algorithm for efficient server allocation. Section 6 deals with experimental evaluation of the proposed model and furthermore the conclusion is discussed in Section 7.

2. Literature Survey

Beloglazov et al. [15] made a study on energy efficient techniques for cloud computing. They studied static and dynamic power management. In static power management, the study concentrates on optimizing methods at the time of circuit design, logical and architectural levels. The dynamic power management contains the runtime adaptation strategies of the model to reduce the energy consumption.

The DVFS technique has been proposed for the reduction of power consumption in the data centers. For example, in [16] garg et al. proposed energy efficient scheduling algorithms that influences DVFS to reduce the frequency of the CPU, minimizes the CO_2 emission and maximize the CPU utilization.

In [17], Rizvandi et al. made an analysis on the energy consumption of the clusters and proposed the MVS-DVFS algorithm for reducing energy consumption in the processers.

There are several studies which mainly focus on the greedy approaches [18-22] for solving the optimization problems. Li et al. proposed ant colony optimization algorithm for task scheduling and the results proved that it is efficient in total task completion time [23]. In [24], the fuzzy- based genetic algorithm is used for optimization of scheduling and the results are proved that the model is efficient.

Some of the mechanisms are concentrated on multi objective scheduling algorithms for reducing power consumption in data centers. In [25], shieh et al. proposed an energy aware algorithm for scheduling periodic tasks in processors by considering voltage overheads. They suggested the linear integer programming model for calculating the energy consumption in the multicore processors and also considered the core numbers. In [26], wang et al. proposed energy aware map reduce model for optimization. They considered three steps for multi job scheduling scheme. As a first step, it considered the performance of servers along with energy consumption, then it considered the data locality and as a final step it considered the integer programming model for task scheduling.

Although a significant study is carried out in the area of energy reduction in cloud computing, the efficient model for task assignment and scheduling is needed for task optimization. The selection of efficient server for task allocation is an effective for reducing energy consumption. The data centers with in the cloud contains server and each server is assigned with a specific functionality. Whenever tasks is assigned to the data center with the help of virtualization the server is divided into number of hosts, so there is no problem for assigning the task to the available server. Server energy will be consumed based on the number of resources demanded by the task at thetime of execution.In the following section, we consider a liner programming model of task scheduling and we propose an optimizing algorithm for reducing the energy consumption.

3. Mathematical Model for Task Assignment

eISSN 1303-5150

The objective of the research is to find a way to distribute tasks over the servers and to develop an energy-aware model for scheduling them. There will be a set of tasks that the user will submit to the cloud. The tasks have to be assigned to the appropriate server with the given capacity.The computational resources used by the server and the tasks are calculated by taking into account the processing time, bandwidth, and allocated CPU. However, it is not possible to determine the exact capacity of each task due to its dynamic nature.A simple way to determine the load of a task is by taking into account the server's capacity. For instance, divide the task T into multiple subtasks $\{t_1,t_2..t_n\}$. Each subtask is assigned to the appropriate server.The total load of a task is 25% of the server's available capacity. If the load of the task changes continuously, then the allocation of resources may not be optimal.

The goal of this problem is to set a theory for the distribution of load between various disjoint sets. It states that the set should be divided into many blocks.The set theory problem is commonly used in computer science to distribute load between various blocks [14]. For instance, set T is divided into 15 different ways. For instance, set T= {t1, t2, t3, t4}.

.

Table 1:Task T with different subsets

$\{t_1t_2t_3t_4\}$	$\{t_1t_2t_3\|t_4\}$	$\{t_1t_2t_4\|t_3\}$
$\{t_1t_2\|t_3t_4\}$	$\{t_1t_2\|t_3\|t_4\}$	$\{t_1t_3t_4\|t_2\}$
$\{t_1t_3\|t_2t_4\}$	$\{t_1t_3\|t_2\|t_4\}$	$\{t_1t_4\|t_2t_3\}$
$\{t_1\|t_2t_3t_4\}$	$\{t_1\|t_2t_3\|t_4\}$	$\{t_1t_4\ \|t_2\|t_3\}$
$\{t_1\|t_2t_3\|t_4\}$	$\{t_1\|t_2\|t_3t_4\}$	$\{t_1\|t_2\|t_3\|t_4\}$

Table 1 shows the possible load distribution of each partition. For instance, in the 14th partition, there are three blocks: t1, t2 and t3t4. Each block requires a single processing unit to complete the task. Since there are three units involved, If the load of the blocks is less than or equal to the processing unit's capacity, then the partition is feasible.The goal of the set partition is to simplify the process of determining the optimal size of each partition. The three steps in this process are: 1) representation of the partitions, 2) objective function, and 3) constraints. After collecting the necessary criteria, the next step is to find the most feasible solution.

In this paper, we introduce the integer programming model to solve the set partition problem. The model can be used to develop a two-step procedure that involves the representation of the various partitions.

Step 1: set T is generated with feasible blocks.

$$T = \{t_j : j = 1..K\} \quad (1)$$

$$t_j = \{L_w : w \in 1..n\} \quad (2)$$

Where n denotes the load items and K denotes the number of blocks.

Step 2: Construction of optimal partition using the previous blocks which are generated.

$$\sum_{i=1}^{K} \lambda_i \rightarrow \min \quad (3)$$

$$\sum_{i=1}^{K} M_{ij}.\lambda_i = 1 \quad i = 1..n \quad (4)$$

eISSN 1303-5150

$$\lambda_j \in \{0,1\} \quad j = 1..K \tag{5}$$

In equation 5, if λ_j =1, then j^{th} block is included, λ_j =0 then the j^{th} block is not included in the optimal partition.

$$M_{ij} = \begin{cases} 1 \ if \ \ L_i \in t_j \\ 0 \ if \ \ otherwise \end{cases} \tag{6}$$

M_{ij}represents the element in M of size $n \times K$ and it is identified in Eq. 6.

After completing the above process, the next step is to optimize the set by adding the optimal partition to it. This will allow the load item $L_i$to appear only in one block of the set. For simplification these constraints has been shown below.

$$M.\lambda = 1 \tag{7}$$

Here M denotes the matrix, λ represents the vector with decision variables of K.

In order to understand the concept of set partitioning, we first consider a task T,in this case, the matrix M of size is taken as the size 3×4 then.

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then, the Eq.7 is developed as

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{8}$$

In Eq. 8, the first row is having two type 1 tasks, in the second row it is having one type 2 task, and in third row it is having one type 3 task.

4. Energy Aware Model for Data Centers

The data center is a collection of servers δ_ϕ, where each server handles different types of tasks. Therefore, each server handles the tasks with different processing units and different energy consumption. Suppose, we consider that there is η type of tasks to be computed by the data center and each server in the data center is denoted as δ_j, where $1 \le j \le \varphi$. The number of tasks of type i is λ_i, where $1 \le i \le \eta$. M_{ij} denotes the number of tasks of type i allocated to the server δ_j.ρ_{ij} denotes the power consumption of a server to compute the task of type i.ϑ_{ij} denotes the time taken to compute the task t_i at server δ_j. β_{ij}denotes the capacity of the server δ_j can queue and process the number of tasks of type i. ω_{ij} denotes the number of waiting tasks of type i at the server δ_j. $\alpha_\omega$is the average task waiting time.

The total energy consumption of the data center is given as $\mathcal{E}_{total}$which is shown in equation 9.

$$\varepsilon_{total} = \sum_{i=1}^{\eta}\sum_{j=1}^{\varphi} \vartheta_{ij}\chi_{ij}M_{ij} \quad (9)$$

The ε_{total} is defined as the total energy consumed for processing the tasks at particular event of time. Here χ_{ij}denotes the power consumed by the task of type i at the server δ_j, M_{ij} denotes the matrix obtained from the linear programming model. The 0 and 1 in the matrix represents the scheduling sequence of tasks which is explained in equation 8.

The goal of implementing a linear programming model is to reduce the total time it takes to complete a task, which is very important for reducing the data center's energy consumption. There are two main cases that are used in this process: 1) No tasks under execution and 2) Waiting at the queue.The first case is that the data center does not have any task under execution. The incoming tasks are then assigned to the appropriate server. In the second case, the incoming tasks are queued and wait for the execution of the task to complete.

No tasks under execution: If the data center does not contain the tasks in the queue and there are no tasks under execution with in the server then the incoming tasks are immediately assigned to the available server. The optimization problem for case 1 is formulated as follows.

$$\min_{M} \varepsilon_{total}(M) = \sum_{i=1}^{\eta}\sum_{j=1}^{\varphi} \vartheta_{ij}\chi_{ij}M_{ij} \;\; such that \sum_{j=1}^{\varphi} M_{ij} = \lambda_i, M_{ij} \le \beta_{ij} \quad (10)$$

The α_ω will simply bind to the $\alpha_j M_j \omega$,

Here α_j ranges from (α_{1j}, α_{2j}, … , $\alpha_{\eta j}$),

M_j is the $\eta \times \sum_{i=1}^{\eta} M_{ij}$ matrix,

ω is the column vector $(\sum_{i=1}^{\eta} M_{ij} - 1, \sum_{i=1}^{\eta} M_{ij} - 2, ..., 1, 0)^{\alpha}$.

*Tasks waiting at the Queue:*The tasks which are under execution at the data center. The new incoming tasks are routed to the queue; the average task waiting time at the queue is given in equation 11.

$$\alpha_w = (\sum_{i=1}^{\eta}\sum_{j=1}^{\varphi} \vartheta_{ij} M_{ij} + \alpha_j M_j \omega)(\sum_{i=1}^{\eta} \lambda_i)^{-1} \quad (11)$$

The optimization problem for the case 2 is formulated as follows.

$$\min_{M} \varepsilon_{total}(M) = \sum_{i=1}^{\eta}\sum_{j=1}^{\varphi} \vartheta_{ij}\chi_{ij}M_{ij} \;\; such that \sum_{j=1}^{\varphi} M_{ij} = \lambda_i, M_{ij} \le \beta_{ij} - \omega_{ij} \quad (12)$$

5. Greedy Assignment Algorithm for Finding Efficient Server

The higher the computation capacity of the server, the more energy-efficient it is. This is done through the selection of the appropriate servers for the scheduling of tasks. The greedy assignment algorithm then takes into account the energy consumption of the system. The energy efficiency of the servers is considered when it comes to the scheduling of tasks. The tasks are then allocated to the most energy-efficient server. This process continues until all tasks are completed and the queue is full. The algorithm used for this computation is known as Algorithm 1.

eISSN 1303-5150

Algorithm 1: Greedy algorithm for efficient server allocation

Input: η, δ_φ
Output: Scheduling of η to δ_φ
Initialize x=1;
y=1;
y=1;
For task M of type i do
For every δ_j do
Compute energy consumption of server from eq. 9
If ($\mathcal{E}(i, j) \leq \mathcal{E}(x, y)$) then
x = i and y = j
end
end
end
Schedule the task x to δ_y
For each waiting tasks do
For every δ_j do
Compute energy consumption of $\mathcal{E}'(i, j)$
If ($\mathcal{E}'(k_i, j) \leq \mathcal{E}(k_i, z)$) then
z = j
end
end
end
Schedule the task k_i to δ_j
end

6. Experimental Setup

The simulation environment is designed by using Cloudsim 3.0.3. We are considered 25 PMs, 20 VMs, 200 arrival tasks and the proposed greedy algorithm is simulated using Cloudsim. The homogeneous tasks with an exponentially distributed task arrival rate of 200 tasks are submitted to the data center. The experimental evaluation is based on measuring the total energy consumption, total number of tasks and average waiting time of the tasks at the queue.

eISSN 1303-5150

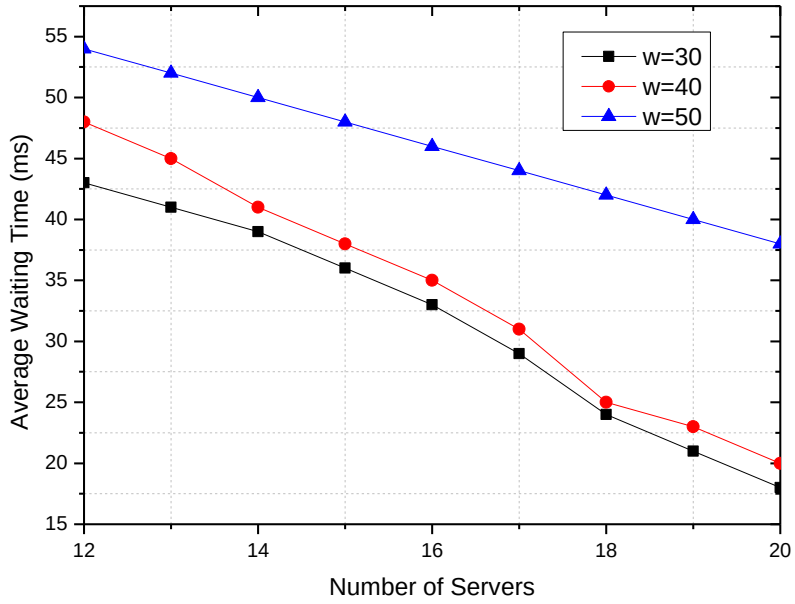

Figure 1: Average waiting time of the tasks at the varied queue length

Figure 1 shows the average waiting time of the tasks under diverse scenarios of queuing capacity the number of servers assigned for task handling increases means the waiting time of the tasks at the queue is automatically decreases. Moreover, there is minimum waiting time for the task for server assignment. The queue length at the servers is taken as w = {30, 40, and 50}.The average waiting time of the task at different workloads i.e. the task arrival rate {50, 100, and 200} is tested with queue length of 50 is shown in figure 2. The number of tasks increases means the waiting time of the task at the queue also increases at the server.

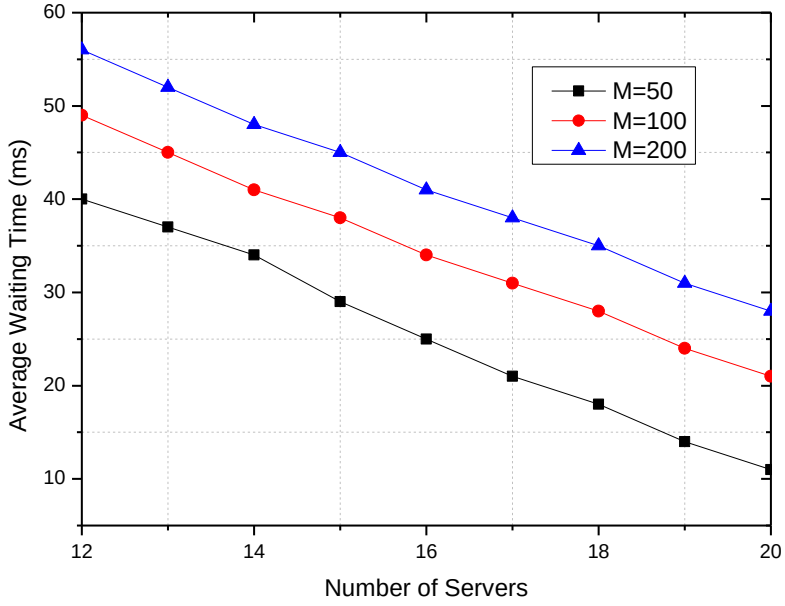

Figure 2: Average Waiting Time of the tasks with different Task arrival rates (Queue Length is 50)

The total energy consumption of the servers at different task arrival rates with queue length is 50 shown in figure 3.

eISSN 1303-5150

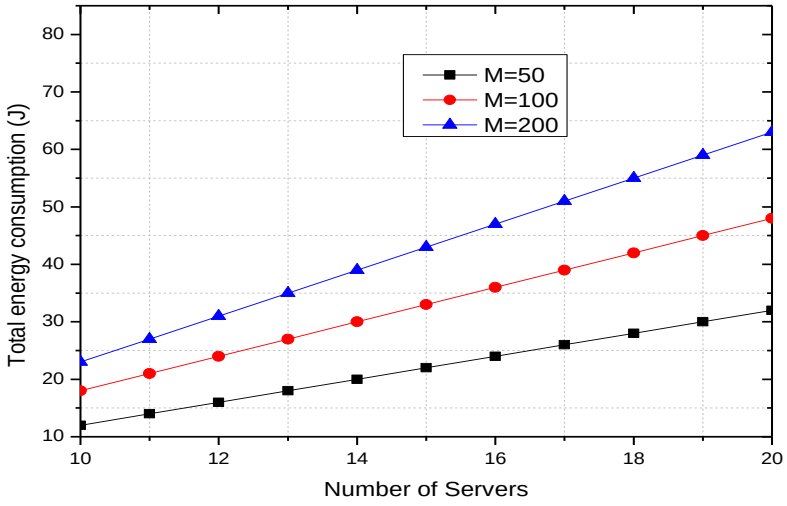

Figure 3: Total Energy Consumption of the Servers with Various Task rates

The energy consumption of the servers is increased with increase in the number of tasks. The number of server increases means it directly impacts on the total energy consumption, so the minimum number of servers is utilized to decrease the overall energy consumption.

The greedy algorithm for scheduling process is evaluated with other existing algorithms. The performance of the proposed algorithm with respect to waiting time of the tasks at queue with 50, 100 and 200 tasks is given in figure 4.

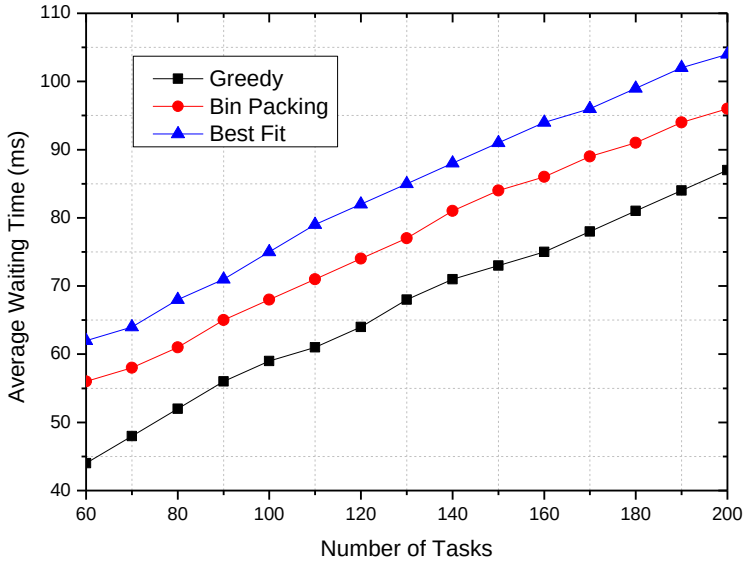

Figure 4: Comparison of Existing algorithms with proposed Greedy Method

In Figure 4, it is shown that the average waiting time of the proposed greedy algorithm is less when compared to other traditional algorithms like Bin Packing and Best fit algorithms.

7. Conclusion

This paper presents a linear programming model that aims to allocate the tasks to the available servers efficiently. It takes into account the energy consumption of the tasks and the average wait time of the queue. This paper proposes a greedy algorithm that takes into account the energy consumption of the

eISSN 1303-5150

tasks. It is modelled using the cloud simulator cloudsim. The results of the study are analysed using different parameters. The proposed model proved to be very efficient.

References

[1] "Google docs," http://docs.google.com.

[2] C. D. Patel and A. J. Shah, "Cost model forplanning, development and operation of a data center,"http://www.hpl.hp.com/techreports/2005/HPL-2005-107R1.pdf.

[3] J. Baliga, R. Ayre, K. Hinton, and R. Tucker, "Green cloudcomputing: Balancing energy in processing, storage, andtransport," in Proceedings of the IEEE, vol. 99, no. 1, January2011, pp. 149 –167.

[4] A. Bohra and V. Chaudhary, "Vmeter: Power modelling forvirtualized clouds," in Parallel Distributed Processing, WorkshopsandPhd Forum (IPDPSW), 2010 IEEE InternationalSymposium on, April 2010, pp. 1 –8.

[5] I. Goiri, F. Juli and, R. Nou, J. Berral, J. Guitart, and J. Torres,"Energy-aware scheduling in virtualized datacenters," inCluster Computing (CLUSTER), 2010 IEEE InternationalConference on, sept. 2010, pp. 58 –67.

[6] A. Beloglazov and R. Buyya, "Energy efficient resourcemanagement in virtualized cloud data centers," in Proceedingsof the 2010 10th IEEE/ACM International Conference onCluster, Cloud and Grid Computing. Washington, DC,USA: IEEE Computer Society, 2010, pp. 826–831. [Online].Available: http://dx.doi.org/10.1109/CCGRID.2010.46

[7] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan,"Energy aware network operations," in INFOCOM Workshops2009, IEEE, April 2009, pp. 1 –6.

[8] Q. Tang, S. Gupta, and G. Varsamopoulos, "Energyefficientthermal-aware task scheduling for homogeneoushigh-performance computing data centers: A cyber-physicalapproach," in Parallel and Distributed Systems, IEEE Transactionson, vol. 19, no. 11, Nov. 2008, pp. 1458 –1472.

[9] V. Raj and R. Shriram, "Power aware provisioning in cloudcomputing environment," in Computer, Communication andElectrical Technology (ICCCET), 2011 International Conferenceon, march 2011, pp. 6 –11.

[10]R.SenthamilSelvan"Intersection Collision Avoidance in DSRC using VANET" on Concurrency and Computation-Practice and Experience, ISSN: 1532-0626/1532-0634, Volume 34, Issue 13/e5856, 4 June 2020.

[11]A.-H. Mohsenian-Rad and A. Leon-Garcia, "Optimal residentialload control with price prediction in real-time electricitypricing environments," Smart Grid, IEEE Transactions on,vol. 1, no. 2, pp. 120 –133, sept. 2010.

[12]G. Wen, J. Hong, C. Xu, P. Balaji, S. Feng, and P. Jiang,"Energy-aware hierarchical scheduling of applications in largescale data centers," in Cloud and Service Computing (CSC), 2011 International Conference on, dec. 2011, pp. 158 –165.

[13]R. Rojas-Cessa, S. Pessima, and T. Tian, "Experimentalevaluation of energy savings of virtual machines in theimplementation of cloud computing," in Wireless and OpticalCommunications Conference (WOCC), 2012 21st Annual,april 2012, pp. 65 –70.

[14]D. Knuth, The Art of Computer Programming, Volume 4, Fascicle 3. Addison-Wesley, 2005.

[15]A. Beloglazov, R. Buyya, Y.C. Lee, A. Zomaya, A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems, 2010.

[16]Garg, S.K., Yeo, C.S.,Anandasivam,A., Buyya, R., 2011. Environment-conscious scheduling of HPC applications on distributed cloud-oriented data centers. J. Parallel Distrib. Comput. 71 (6), 732–749

[17]Rizvandi, N.B., Taheri, J., Zomaya, A.Y., 2011. Some observations on optimal frequency selection in DVFS-based energy consumption minimization. J. Parallel Distrib. Comput. 71 (8), 1154–1164.

[18]Radha, K., et al. "Allocation of Resources and Scheduling in Cloud Computing with Cloud Migration." *International Journal of Applied Engineering Research, ISSN* (2014): 0973-4562.

[19]Raju, R., et al. "A heuristic fault tolerant MapReduce framework for minimizing makespan in Hybrid Cloud Environment." *Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on*. IEEE, 2014.

[20]Kapur, Ritu. "A workload balanced approach for resource scheduling in cloud computing." *Contemporary Computing (IC3), 2015 Eighth International Conference on*. IEEE, 2015.

[21]Kaur, Tarandeep, and InderveerChana. "Energy Efficiency Techniques in Cloud Computing: A Survey and Taxonomy." *ACM Computing Surveys (CSUR)* 48.2 (2015): 22.

[22]Bhalerao, Bhushan, and Mr Shailendra W. Shende. "A Review on Different Scheduling Algorithms for Workflows in Cloud environment." *International Journal on Recent and Innovation Trends in Computing and Communication*3.2 (2015).

[23]Li, J., Peng, J., Cao, X., Li, H.-y.: A task scheduling algorithm based on improved ant colony optimization incloud computing environment. Energy Procedia 13, 6833–6840 (2011)

[24]Tayal, S.: Tasks scheduling optimization for the cloud computing systems. Int. J. Adv. Eng. Sci. Technol.5(2), 111–115 (2011)

[25]Shieh, W.-Y., Pong, C.-C.: Energy and transition-aware runtime task scheduling for multicore processors. J.ParallelDistrib. Comput. 73(9), 1225–1238 (2013)

[26]Wang, X., Wang, Y., Cui, Y.: A new multi-objective bi-level programming model for energy and localityaware multi-job scheduling in cloud computing. Futur. Gener. Comput. Syst. 36, 91–101 (2014)

eISSN 1303-5150