IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-11, Nov 2018 E-ISSN: 2347-2693

Energy-Efficient Heuristics Job Scheduling Algorithm using DVFS Technique for Green Cloud Data Centers

K. Sutha[1*], G. M. Kadhar Nawaz[2]

[1]Computer Science, Research and Development Centre, Bharathiar University, Coimbatore - 641046, Tamilnadu, India
Dr.MGR Janaki College of Arts and Science for Women, Chennai
[2]Dept. of Master of Computer Applications, Sona College of Technology, Salem - 636005, Tamilnadu, India

**Corresponding Author: ksutha1986@gmail.com*

Available online at: www.ijcseonline.org

Accepted: 20/Nov/2018, Published: 30/Nov/2018

Abstract— Cloud computing provides unlimited on-demand resources and services through remote servers based on pay-per-use model. It includes Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS). Cloud computing facilitates efficient utilization of computing resources in large-scale cloud data centers. Day-by-day, increasing usage of cloud computing services leads to increasing energy consumption and operational cost. Moreover, it produces high amount of Co_2, causing huge environmental damage. Heavy usage of cloud data centers has also become a problem to sacrifice system performance and Quality of Services (QoS). In order to overcome these problems, an efficient job-scheduling algorithm is required to reduce energy consumption and execution time without diminishing performance of the system. Apart from this, a green cloud data center plays a significant role in cloud computing to reduce Co_2 emissions. Energy-efficient heuristics model is used to find an optimal solution for executing jobs of varying sizes and timings. In this paper, using Dynamic Voltage Frequency Scaling (DVFS), we introduce Energy-Efficient Job Scheduling (EEJS) algorithm to green cloud data centers. Our proposed algorithm is compared to Energy-Conscious Scheduling algorithm (ECS) and Green Energy-Efficient Scheduling algorithm (Green-EES). Experimental results are evaluated using CloudSim 3.0.3 toolkit and simulation results are validated in low-, medium-, and high-workload conditions. Compared to other two algorithms, EEJS demonstrates reduced energy consumption and execution time without violating Service Level Agreements (SLA).

Keywords— Cloud Computing, Job Scheduling, Heuristics Model, DVFS, Energy Consumption, SLA Violation

I. INTRODUCTION

Cloud Computing provides enormous resources such as servers, storage, networks and databases. It also offers shared pool of computing resources with minimum management efforts. The goal of energy-efficient computing is to provide efficient use of computing resources that generate less heat and cooling system to achieve moderate temperature [1-2]. To maintain energy efficiency in cloud data center measures such as efficient job/task scheduling, personalized virtual machine creation and migration, appropriate resource utilization, balancing data center load, switching-off of idle servers are needed. Recently, International Energy Agency (IEA) highlighted numerous benefits provided by cloud data centers such as energy saving, environmental sustainability, increased asset values, macro-economic development, industrial productivity, energy security, energy access, energy prices, public budgets, disposable income and reduced local air pollution [3]. Even then, provided benefits are not sufficient to cater the needs of million users increasing every day. Thus, increasing access of internet/cloud data centers leads to high operational cost and reduced return on investment (ROI). Moreover, cloud providers have to spend more money in order to get maximum utilization of cloud resources. Sometimes, providers are third-party agents and their expenses are directly borne by the cloud users either knowingly or unknowingly. To avoid such problems, we have focused on reducing energy consumption and execution time for increasing resource utilization of cloud data centers. Traditional energy-saving algorithms are typically focused on current CPU workload and allocate jobs accordingly. In this paper, we have proposed energy-efficient heuristics job scheduling algorithm using DVFS technique to allocate jobs in dynamic cloud environment. Heuristics model is used to find an optimal solution when job size varies for different users. Besides, it is designed to identify feasible solution with less computational complexity. The proposed algorithm works based on parallel heuristics model and runs several jobs in parallel. This model supports VM manager to locate all available virtual machines and allocate jobs accordingly.

The rest of this paper is organized as follows: Section 2 discusses related work and Section 3 demonstrates green cloud architecture. Section 4 discusses problem model and Section 5 discusses proposed work. Section 6 discusses experimental analysis and results. Conclusion and future scope of work are described in Section 7.

II. RELATED WORK

Energy-efficient job scheduling mainly focuses on maximizing CPU utilization, minimizing resource utilization and carbon emission. DVFS technique plays an important role in reducing energy consumption in all electronic devices such as desktop, laptop, handheld devices like mobile phones, PDAs (Personal Digital Assistant) and smart watches. Thandar Thein et al. [4] have proposed a framework in which they show effective performance for achieving the high data center energy efficiency and preventing Service Level Agreement (SLA) violation. This framework works based on Reinforcement Learning (RL) mechanism and Fuzzy Logic for green solutions. Ali Naghash Asadi et al. [5] have evaluated the power consumption and performance measures using Stochastic Activity Networks (SANs). In addition, they have discussed a problem whether servers execute different or same number of VMs.

Ning Liu et al. [6] have proposed job scheduling algorithm for minimizing energy consumption in cloud data centers, where they stress on less response time and minimum number of active servers needed to execute a task using greedy task scheduling algorithm. To minimize energy expenditure, they have applied Most-Efficient-Server-First task scheduling algorithm. However, they have discussed energy consumption without focusing on operational cost and SLA violations. Xingjian Lu et al. [7] have studied geographical job scheduling in heterogeneous cloud data centers, where they have proposed two new models, namely Joint Job Scheduling and Alternating Direction method. Joint job scheduling model is used to find optimal distribution of jobs over each data center and it works based on alternating direction method.

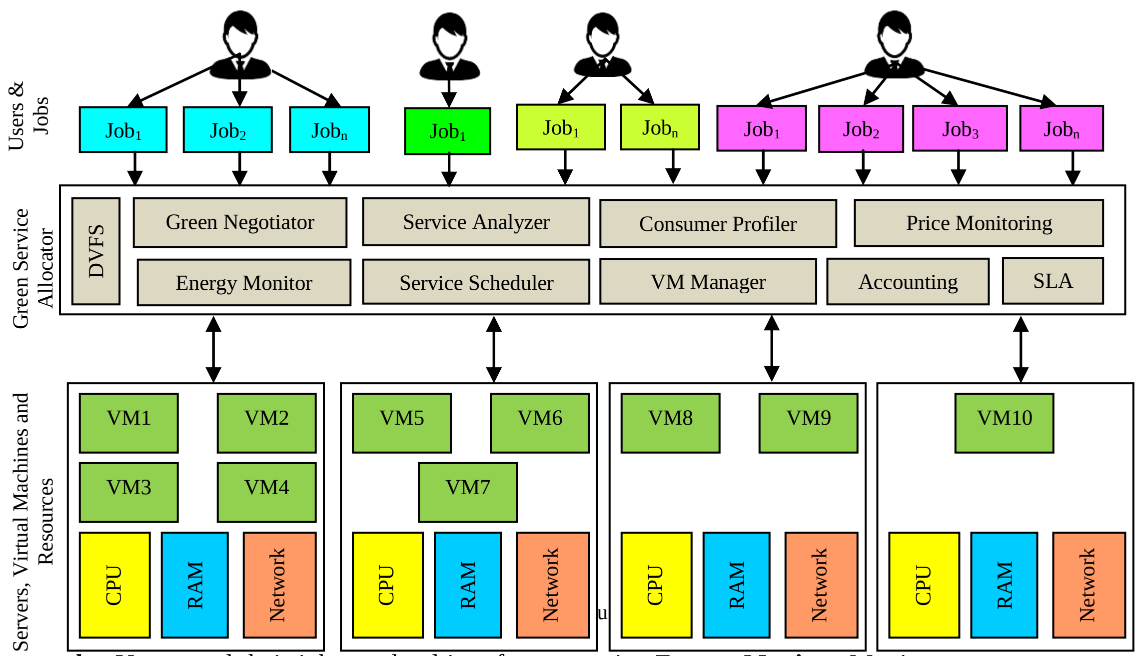
In our earlier research work [8], we have discussed various job-scheduling algorithms and its performance metrics. Energy-efficient job scheduling algorithms focus on performance metrics such as execution time, makespan, waiting time, response time, scalability, reliability, resource utilization and so on. Finally, we have concluded that future research should focus more on energy consumption to reduce operational cost of cloud data centers. A V Karthick et al. [9] have proposed Multi-Queue Job Scheduling algorithm using burst time and experimental results in dynamic cloud environment to avoid starvation problem. Dynamic job selection helps to utilize unused free space for increasing resource utilization. Nowadays, many algorithms are available for job scheduling instead of conventional algorithms. However, proposed work is compared to traditional job scheduling algorithms, namely First Come First Serve (FCFS) and Shortest Job First (SJF) scheduling algorithm. Chien-Hung Chen et al. [10] have introduced deadline-constrained job scheduling for heterogeneous cloud and their experimental results were tested using MapReduce software. They have proposed Bipartite Graph Modelling called, BGMRS, to reduce execution time and increase node performance. If active jobs violate predefined deadline, then BGMRS can minimize the number of jobs placed on data locality. However, they have not focused on energy consumption of jobs in deadline-constrain model.

Sindhu S et al. [11] have proposed efficient task scheduling algorithm to reduce turnaround time and improve resource utilization. In this paper, they have discussed two new algorithms, namely Longest Cloudlet Fastest Processing Element (LCFP) and Shortest Cloudlet Fastest Processing Element (SCFP). Each task is assigned based on computational complexity and computing capacity. However, they have validated experimental results without concentrating on performance of the system. Anton Beloglazov et al. [12] have discussed detailed survey on energy-efficient cloud data centers. They have explained causes and problems of high power consumption of data centers. In each section, they have discussed energy-efficient computing system focusing on hardware, software, operating systems, virtualization and data center design. Bin Hu et al. [13] have investigated dynamic task scheduling via Policy Iteration Scheduling (PIS) approach. For dynamic changes of workload in cloud data centers, they have focused on hardware updation and task queuing. Meanwhile, PIS facilitates to optimize each task independently and reduce total execution time. However, they have discussed execution time without focusing on energy consumption and SLA violation of task.

Auday Al-Dulaimy et al. [14] have investigated design and implementation of dynamic virtual machine placement for energy-efficient data centers. It includes VM placement and VM consolidation to improve energy efficiency. Multiple Choice Knapsack Problem achieves VM selection, placement and migrations. However, they have discussed efficient VM strategies for energy consumption without covering SLA violations. Ziqian Dong et al. [15] have proposed energy-efficient task scheduling using greedy approach. Moreover, “Most Efficient Server First” scheduling algorithm is used to minimize energy consumption and response time. Greedy task scheduler monitors the number of active servers running on a physical machine and controls high-energy consumption. Mateusz Zotkiewicz et al. [16] have discussed two approaches such as workflow scheduling and energy-aware task scheduling. Virtual deadline is assigned to each task, which it executes without disturbing other tasks. Each task is dynamically assigned based on active server network link and available computing resources. Moreover, implementation of energy-efficient task scheduling is done by minimum-dependencies energy-efficient directed acyclic graph (MinD+ED) model. Marco Polverini et al. [17] have discussed problem of scheduling batch jobs distributed in geographical data centers where they have introduced provably-efficient online algorithm called, GreFar, for optimizing energy cost and temperature control.

III. Green Cloud Architecture

Green cloud architecture with multiple resources is shown in Fig.1. In this figure, each layer illustrates the role of individual component and usage model. In addition, VM manager acts as an administrator/controller and allocates VMs according to current job requirement.



a) ***Users & Jobs:*** Users send their jobs to cloud interfaces. Once request has been received from cloud users, service provider searches and allocates appropriate virtual machines matching the job size.

b) ***Green Service Allocator:*** Once job gets appropriate VMs, allocator acts as a service broker between users and cloud infrastructure. In addition, energy-efficient heuristics job scheduling algorithm involves the following:

i. ***Green Negotiator:*** Improving QoS in cloud data centers, green negotiator calculates usage prices for SLA violation and resource utilization. Energy-saving scheme is also included in this layer and penalties are added to prices when users make SLA violations.

ii. ***Service Analyzer:*** Analyzes service requirements before executing a job. It will reduce huge server load and energy consumption.

iii. ***Consumer Profiler:*** Collects necessary information about the users for giving priorities to privileged customers.

iv. ***Pricing:*** Calculates usage charges based on service request and resource requirements.

v. ***DVFS:*** Dynamic energy consumption is calculated using DVFS technique. It will reduce energy consumption via dynamic voltage and frequency scaling factors.

vi. ***Energy Monitor:*** Monitors energy consumption of each virtual machine and physical server. Periodical reports are sent to VM manager for reducing energy consumption and increasing resource utilization.

vii. ***Service Scheduler:*** Service scheduler calculates weight of each job and allocates resources for appropriate VMs.

viii. ***VM Manager:*** Monitors overall physical and virtual machine execution status and finds availability of free VMs for new arrival jobs.

ix. ***Accounting:*** Monitors and maintains complete usage of resource and service cost. It will help the service providers to increase number of cloud users in future.

x. ***SLA:*** Service Level Agreement involves overall agreement between user and provider. It includes types of resources are requested by the user and duration of usage.

c) **Servers, VMs and Resources:** Each server has multiple VMs and resources like CPU, memory/RAM and network interfaces. Servers and resources are allocated to particular user based on current job requirement.

IV. Problem Model

Cloud computing distributes on-demand resources such as disk storage, CPU, memory and network interfaces in heterogeneous cloud data centers. Heavy usage of cloud data centers increases many computational issues in dynamic cloud environment. Efficient utilization of cloud resources reduces energy consumption and minimizes operational cost. In this paper, we have proposed energy-efficient job scheduling algorithm for efficient utilization of cloud resources to find available resources in cloud data centers and allocate those resources to VMs. The aim of this paper is utilizing energy-efficient job scheduling algorithm, to utilize minimum resource and provide maximum befit to both cloud users and providers. Energy consumption is closely related to DVFS technique and includes three parameters: frequency (f), capacitance (c) and voltage (v). Adjusting such parameters either low/high according to the workload enables to reduce energy consumption. The following definitions are involved to formalize such scenario:

Definition 1:

S = {S_1,S_2,S_3,S_4……,S_n}, where S denotes server and S_1,S_2,S_3,S_4……S_n indicate number of servers residing in a single data center. Each server has limited capacity and resources. Based on server capacity, VM manager can allocate jobs to the server or else decide to create new VMs. Each server has maximum (R_{max}) and minimum (R_{min}) resources.

$$\text{Data Center DC} = \sum_{i=1}^{n} \{S_1, S_2, S_3, \dots, S_n\} \;, \tag{1}$$

$$\text{Server S} = \sum_{j=1}^{n} \{VM_1, VM_2, VM_3, \dots, VM_n\} , \tag{2}$$

$$\text{Each VM} \le \sum_{k=1}^{n} \{J_1, J_2, J_3, \dots, J_n\} \;, \tag{3}$$

$$S_i \le VM_i \le J_i \;, \tag{4}$$

$$S_i = \{R_{min}, R_{max}\} \;. \tag{5}$$

Definition 2:

VM = {VM_1,VM_2,VM_3,..,VM_n}, where VM denotes the virtual machine. Then VM_1,VM_2,VM_3,...,VM_n are considered as number of virtual machines residing in a server. Each VM has minimum (F_{min}) and maximum (F_{max}) working frequency:

$$VM_i = \{F_{min}, F_{max}\} \;. \tag{6}$$

Definition 3:

J = {J_1,J_2,J_3,J_4…….J_n}, where J denotes the job and J_1,J_2,J_3,J_4…….,J_n are number of jobs from single user. Working frequency of each job is considered as F_{min} and F_{max}:

$$J_i = \{F_{min}, F_{max}\} \;. \tag{7}$$

Definition 4:

SLA defines a commitment between service provider and a user. It ensures that quality, availability and responsibility as agreed between both parties [18]. Besides, it is designed as either lower or higher level SLA based on resource requirement of each job. A job is assigned by various SLA parameters such as type of service, service duration, performance level, monitoring process, reporting etc., Service providers may levy heavy penalties on users for violating SLA conditions. To avoid such instants, the following constraints are defined:

- Let N be = the total number of SLA level
- Let L be = each level of SLA
- Lower and higher level SLA ranging from $0 \le L \le N-1$
- Lower SLA Level = minimum resource requirement
- Higher SLA Level = maximum resource requirement

SLA levels and number of VMs are shown in Table 1. Lower level SLA is assigned to minimum resource usage and higher-level SLA is assigned to maximum resource requirement. Let N be the total number of SLA levels and L be the individual level. SLA allocation constraints are addressed ranging from 0 to N-1, $0 \le L \le N-1$. Lower level SLA initiates from 1–to–100 and higher-level SLA is assigned based on the dynamic load of current job.

Table 1. SLA Levels and Number of VMs

SLA Level	Number of VMs
0	1–100
1	101–200
2	201–300
3	301–400
4	401–500
5	501 and above

Definition 5:

Power and energy model [19] can be defined as:

$$P = \frac{W}{T} \;, \tag{8}$$

$$E = P \cdot T \;, \tag{9}$$

where P is power, W is total work done by period of time interval, T is period of time and E is energy. Dynamic power consumption is considered as:

$$P_{dynamic} = a \cdot C \cdot V^2 \cdot f \;, \tag{10}$$

where a is switching activity, c is physical capacity of server, v is supply voltage and f is working clock frequency. Therefore, power consumption of server is defined as:

$$P(u) = P_{idle} + (P_{busy} - P_{idle}) \times u \;, \tag{11}$$

where P is estimated power consumption, P_{idle} is power consumption of an idle server, P_{busy} is power consumption of active server when fully utilized and u is active CPU utilization of server.

V. PROPOSED WORK

The primary performance goal of computing systems is to reduce execution time while increasing throughput. To achieve these systems, developers focus on creating high performance computing. Y C Lee et al. [20] addressed the problem of scheduling precedence-constrained parallel applications on multiprocessor computer systems. They have proposed an algorithm called, Energy Conscious Scheduling heuristics (ECS and ECS+idle), for reducing energy consumption using dynamic voltage scaling (DVS). In ECS algorithm, they have applied Relative superiority (RS) and Makespan-Conservative Energy Reduction (MCER) techniques for optimizing system performance. Chia-Ming Wu et al. [21] have introduced Green Energy-Efficient Scheduling (Green-EES) algorithm for increasing resource utilization and reducing energy consumption. In that algorithm, the weight of each virtual machine is arranged in an increasing order wherein, heavy-weight job have to wait for long time to get appropriate VM. To overcome this problem, we have proposed Energy-Efficient Job Scheduling heuristics (EEJS) using DVFS technique for green cloud data centers. In this algorithm, VM manager has to monitor current CPU load and allocate jobs according to VM requirement. Decreasing order of VM weight helps VM manager to allocate jobs with less energy consumption and reduced execution time. Algorithm 1 demonstrates less execution time and energy consumption of VMs residing and executing in a server.

Algorithm 1. Energy-Efficient Job Scheduling (EEJS) Algorithm

Input	: job and SLA level
Output	: job allocation
Abbreviations	VMM is Virtual Machine Manager
	W_i is weight of virtual machine
	P_i is unit power cost
	R_i is resource usage cost

```
1   Receives job and SLA level from user
2   Search and select available VM
3   Calculate VMWeight Wi = Pi * Ri
4   VMWeight.sortDecreasingorder()
5   if (VMWeight = = jobWeight) || (VMWeight > jobWeight)
6   |        allocate VM ← job
7   |        else if (VMWeight < jobWeight) goto step 2
8   |        |   else (jobWeight != VMWeight) && (VMWeight = = 0) then
9   |        |   |     VMM ← noAvailableVM
10  |        |   |     VMM←suspendedServerlist, goto step 18
11  |        |    end if
12  |         end if
13  end if
14  if (suspendedServer = = NULL)
15  |    |   jobAssignment.failure() goto step 20
16  |    |   jobMoveTo.waitState(), goto step 2
17  |   else
18  |    |  ServerList.add(suspendedServer), goto step 3
19  end if
20  VMM ← jobAssignmentResult()
```

VI. PERFORMANCE ANALYSIS AND RESULTS DISCUSSION

A. Experimental Setup

The implementation of cloud computing research work in real world is more expensive and difficult to conduct repeatable experiments. Therefore, to ensure the repeatability of experiments, we have chosen Cloudsim 3.0.3 toolkit as a simulation platform. Cloudsim is a library for the simulation of cloud scenarios, which comprises of power management, network management, memory management etc., It also provides essential classes for describing data centers, computational resources and virtual machines for scheduling and provisioning resources in cloud environment [22]. In our experiment, each job is composed of 1000, 2000 or 3000MIPS instructions with one CPU core for each physical machine. In addition, 16GB RAM, one TB of storage space is allocated for performance evaluation. The power consumption for each host is ranging from 175W to 250W. In which, 175W is consumed when a host is in 0% utilization and 250W for 100% CPU utilization. Besides, each VM needs 1 CPU core and 250, 500, 750 or 1000MIPS, 128MB RAM and 1 GB storage space used.

B. Simulation Results

Energy-efficient job scheduling enhances system performance and increases higher potential growth of cloud usage. In this section, we have elaborated on execution time and energy consumption of cloud data centers. We have evaluated our simulation results in Cloudsim toolkit and three workload categories are considered for comparison: low, medium and high. These three variations can help identify performance analysis of CPU workload easily. Moreover, we have validated our test results compared with three scheduling algorithms, namely Energy-Conscious Scheduling, Green Energy-Efficient scheduling and Energy-Efficient Job Scheduling. The reason for choosing these three algorithms is that they work with reduced energy consumption. While comparing these algorithms, our proposed EEJS algorithm consumes less energy and execution time. In addition, SLA violation rules are also defined before allocating any jobs to the VM. Table 2 shows performance analysis of execution time and energy consumption in low-, medium-, and high-workload conditions.

Table 2. Performance analysis of Execution Time and Energy Consumption

Algorithm	Low-workload Execution Time (Sec.)	Low-workload Energy Consumption (kWh)	Medium-workload Execution Time (Sec.)	Medium-workload Energy Consumption (kWh)	High-workload Execution Time (Sec.)	High-workload Energy Consumption (kWh)
Energy-Conscious Scheduling (ECS)	63	3.67	342	74.49	770	118.50
	70	4.33	379	80.12	802	124.41
	79	4.70	388	85.47	846	130.39
	93	5.33	419	96.32	880	136.76
	112	5.74	439	101.55	924	141.17
	123	6.43	458	115.10	975	146.33
	139	7.16	472	120.00	1001	153.10
	155	7.78	483	127.52	1047	160.67
	167	8.05	500	134.95	1052	169.36
	178	8.60	510	141.33	1097	178.16
Green Energy-Efficient Scheduling (Green-EES)	59	3.34	376	85.86	735	110.72
	65	3.69	389	90.02	774	119.86
	74	4.22	400	98.76	800	125.84
	89	4.69	432	106.32	832	130.16
	100	5.03	451	115.95	870	137.69
	111	5.41	465	126.12	904	141.07
	130	6.00	486	134.67	935	147.62
	147	6.72	502	150.29	976	153.36
	153	7.33	512	166.55	991	160.10
	161	8.19	523	179.95	1062	165.44
Energy-Efficient Job Scheduling (EEJS)	54	2.97	320	70.04	683	103.77
	59	3.26	352	76.43	712	111.33
	66	3.59	376	80.17	756	119.67
	75	3.85	389	87.12	793	122.62
	89	4.10	410	93.47	826	127.13
	96	4.55	423	99.70	870	130.00
	110	5.20	444	106.59	900	135.36
	125	5.93	455	110.32	929	141.10
	138	6.55	467	118.76	946	146.27
	147	7.16	495	124.99	980	154.32

Fig. 2 shows execution time of jobs with low-workload simulation. In this simulation, we have chosen job size ranging from 10000MIPS to 19000MIPS (Million Instruction per Second) which increases by 1000MIPS every step. The execution time of low-workload job with 10000MIPS executed by ECS is 63sec, by Green-EES is 59sec and by EEJS is 54sec. Meanwhile, when increasing job size, execution time also gradually increases. The execution time taken by ECS, Green-EES and EEJS for executing a job of 19000MIPS is 178sec, 161sec, and 147sec, respectively. Therefore, simulation results have shown that compared to ECS and Green-EES, EEJS has reduced execution time in low-workload condition.

Fig. 3 shows comparison between ECS, Green-EES and EEJS energy consumption of jobs on low-workload condition. Energy consumption of each job is measured by kilowatt-hour (KWh). The mean energy consumption of job of 10000MIPS executed by ECS is 3.67kWh, by Green-EES is 3.34kWh and by EEJS is 2.97kWh. The energy consumption of job with 19000MIPS is 8.60kWh, 8.19kWh and 7.16kWh. Moreover, EEJS consumes 19% less energy consumption compared to ECS and 11% less energy consumption compared to Green-EES while executing a job size of 10000MIPS. Thus, EEJS consumes 17% less energy consumption compared to ECS and 13% less energy consumption compared to Green-EES while executing a job size of 19000MIPS. Therefore, experimental results prove that EEJS consumes less energy consumption than ECS and Green-EES algorithms in low-workload condition.

Fig. 4 shows execution time of jobs with medium-workload simulation. In this simulation, we have chosen job size ranging from 30000MIPS to 66000MIPS successively increasing by 4000MIPS gradient. The execution time of medium-workload job with 30000MIPS executed by ECS is 342sec, by Green-EES is 376sec and by EEJS is 320sec. In this result, ECS and EEJS execution timings are reduced compared to Green-EES. The execution time taken by ECS, Green-EES and EEJS for executing a job of 66000MIPS is 510sec, 523sec, and 495sec, respectively. Therefore, simulation results have shown that EEJS execution time of medium-workload is reduced compared to ECS and Green-EES.

Fig. 5 shows comparison between ECS, Green-EES and EEJS energy consumption of jobs on medium-workload condition. The mean energy consumption of job of 30000MIPS executed by ECS is 74.49kWh, by Green-EES is 85.86kWh and by EEJS is 70.04kWh. The energy consumption of job with 66000MIPS is 141.33kWh, 179.95kWh and 124.99kWh. Consequently, EEJS consumes 6% less energy consumption compared to ECS and 21% less energy consumption compared to Green-EES while executing a job size of 30000MIPS. In addition, EEJS consumes 12% less energy consumption compared to ECS and 31% less energy consumption compared to Green-EES while the job size is 66000MIPS. Therefore, experimental results have shown that EEJS consumes less energy consumption than ECS and Green-EES algorithms in medium-workload condition.

Fig. 6 shows execution time of jobs with high-workload simulation. In this simulation, we have chosen job size ranging from 100000MIPS to 220000MIPS, which increases by 12000MIPS every step. The execution time of high-workload job with 100000MIPS executed by ECS is 770sec, by Green-EES is 735sec and by EEJS is 683sec. In this result, EEJS execution time is reduced compared to Green-EES and ECS. The execution time taken by ECS, Green-EES and EEJS for executing a job of 220000MIPS is 1097sec, 1062sec, and 980sec, respectively. Therefore, simulation results have shown that EEJS execution time of high-workload is reduced compared to ECS and Green-EES.

Fig. 7 shows comparison between ECS, Green-EES and EEJS energy consumption of jobs on high-workload condition. The mean energy consumption of job of 100000MIPS executed by ECS is 118.50kWh, by Green-EES is 110.72kWh and by EEJS is 103.77kWh. The energy consumption of job with 220000MIPS is 178.16kWh, 165.44kWh and 154.32kWh. Meanwhile, EEJS consumes 12% less energy consumption compared to ECS and 6% less energy consumption compared to Green-EES while executing a job size of 100000MIPS. Thus, EEJS consumes 13% less energy consumption compared to ECS and 7% less energy consumption compared to Green-EES while the job size is 220000MIPS. Therefore, experimental results have shown that EEJS consumes less energy consumption than ECS and Green-EES algorithms in high-workload condition.

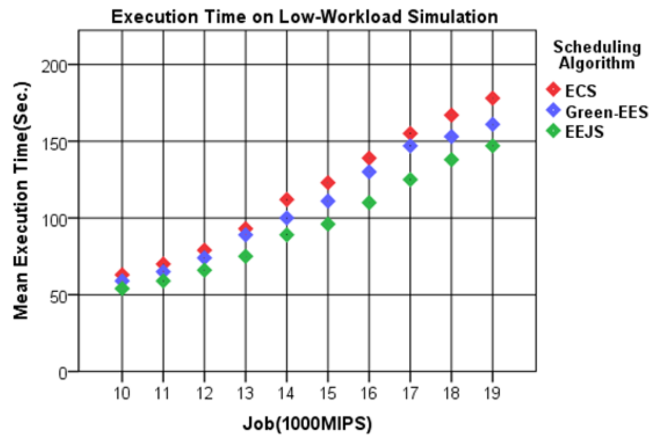


Fig.2 Execution time of jobs in low-workload condition

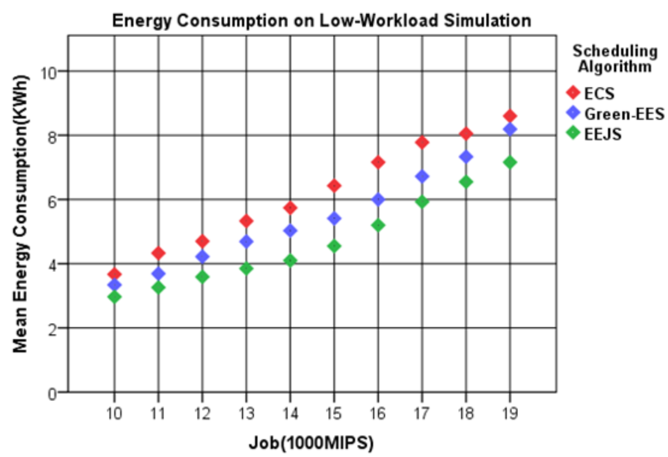


Fig.3 Energy consumption of jobs in low-workload condition

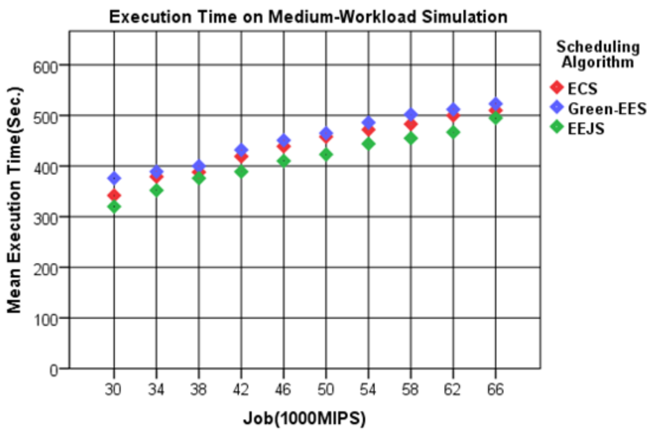


Fig.4 Execution time of jobs in medium-workload condition

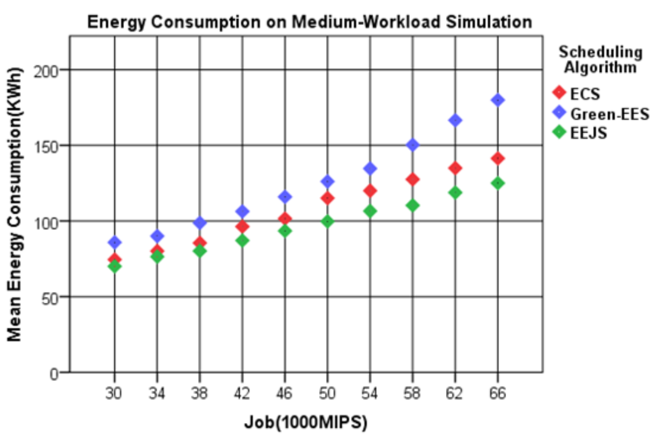


Fig.5 Energy consumption of jobs in medium-workload condition

Fig.6 Execution time of jobs in high-workload condition

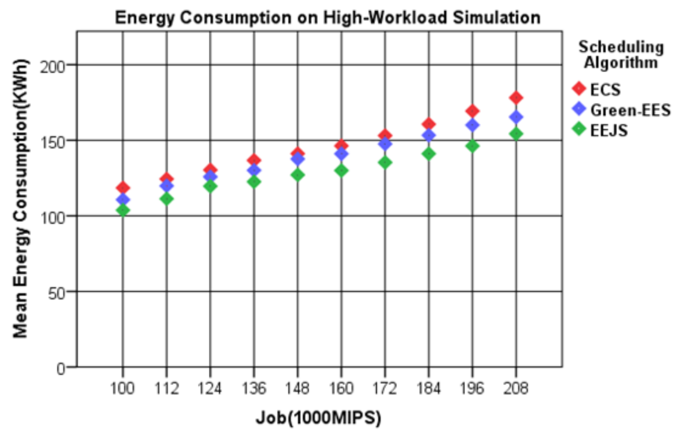


Fig.7 Energy consumption of jobs in high-workload condition

VII. Conclusion

Cloud computing is an emerging technology that provides wide range of on-demand virtual resources over the internet. Energy-efficient heuristics cloud computing facilitates reducing energy consumption and increases resource utilization. Meanwhile, uninterrupted services lead to more energy consumption and increasing operational cost. In this paper, we have focused on efficient job scheduling algorithm for reducing energy consumption and execution time of cloud data centers. Evaluations of test results were compared to three scheduling algorithms, namely Energy-Efficient Job Scheduling (EEJS), Energy-Conscious Scheduling (ECS) and Green Energy-Efficient Scheduling (Green-EES). Our proposed EEJS algorithm works efficiently based on parallel heuristics model in low-, medium-, and high-workload conditions. We have validated our simulation results using Cloudsim toolkit for substituting different workload inputs. Therefore, compared to ECS and Green-EES, our proposed EEJS experimental results demonstrate less energy consumption and execution time without compromising performance of the system. In future, we have planned to work with more parameters to increase scalability, reliability and reduce operational cost.

Acknowledgment

My sincere thanks to Dr.G.M.Kadhar Nawaz MCA., Ph.D, Director & Professor, Department of Master of Computer Applications, Sona College of Technology, Salem, India, for his valuable guidance and constant support to this research work.

References

[1] Mohammad Masdari, Sima ValiKardan, Zahra Shahi, Sonay Imani Azar, *Towards workflow scheduling in cloud computing: A comprehensive analysis*, Journal of Network and Computer Applications, Vol. 66, pp. 64-82, May 2016.

[2] Ali Vafamehr, Mohammad E. Khodayar, *Energy-aware cloud computing*, The Electricity Journal, Vol. 31, No. 2, pp. 40-49, March 2018.

[3] *Capturing the Multiple Benefits of Energy Efficiency*, International Energy Agency (IEA), OECD/IEA, 2014.

[4] Thandar Thein, Myint Myat Myo, Sazia Parvin, Amjad Gawanmeh, *Reinforcement Learning based Methodology for Energy-efficient Resource Allocation in Cloud Data Centers*, Journal of King Saud University - Computer and Information Sciences, November 2018. doi: https://doi.org/10.1016/j.jksuci.2018.11.005

[5] Ali Naghash Asadi, Mohammad Abdollahi Azgomi, Reza Entezari-Maleki, *Evaluation of the impacts of failures and resource heterogeneity on the power consumption and performance of IaaS clouds*, The Journal of Supercomputing, Springer Science+Business Media, LLC, part of Springer Nature, November 2018. https://doi.org/10.1007/s11227-018-2699-5

[6] Ning Liu, Ziqian Dong, Roberto Rojas-Cessa, *Task Scheduling and Server Provisioning for Energy-Efficient Cloud-Computing Data Centers*, IEEE 33rd International Conference on Distributed Computing Systems Workshops, Philadelphia, PA, USA, pp. 226-231, July 2013.

[7] Xingjian Lu, Fanxin Kong, Jianwei Yin, Xue Liu, Huiqun Yu, Guisheng Fan, *Geographical Job Scheduling in Data Centers with Heterogeneous Demands and Servers*, IEEE 8th International Conference on Cloud Computing, New York, NY, USA, pp. 413-420, July 2015.

[8] K Sutha and G M Kadhar Nawaz, *Research Perspective of Job Scheduling in Cloud Computing*, IEEE Eighth International Conference on Advanced Computing (ICoAC), Chennai, India, pp. 61-66, January 2017.

[9] A V.Karthick, Dr.E.Ramaraj, R.Ganapathy Subramanian, *An Efficient Multi Queue Job Scheduling for Cloud Computing*, World Congress on Computing and Communication Technologies, Trichirappalli, India, pp. 164-166, March 2014.

[10] Chien-Hung Chen, Jenn-Wei Lin, and Sy-Yen Kuo, *MapReduce Scheduling for Deadline-Constrained Jobs in Heterogeneous Cloud Computing Systems*, IEEE Transactions on Cloud Computing, Vol. 6, No. 1, pp. 1-14, August 2015.

[11] Sindhu S, Mukherjee S, *Efficient Task Scheduling Algorithms for Cloud Computing Environment*, International Conference on High Performance Architecture and Grid Computing, Chandigarh, India, pp. 79-83, July 2011

[12] Anton Beloglazov, Rajkumar Buyya, Young Choon Lee, and Albert Zomaya, *A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems*, Advances in Computers, Vol. 82, No. 2, pp. 47-111, 2011.

[13] Bin Hu, Ning Xie, Tingting Zhao and Xiaotong Zhang, Dynamic *Task Scheduling Via Policy Iteration Scheduling Approach for Cloud Computing*, KSII Transactions on Internet and Information Systems, Vol. 11, No. 3, pp. 1265-1278, March 2017.

[14] Auday Al-Dulaimy, Wassim Itani, Rached Zantout, Ahmed Zekri, *Type-Aware Virtual Machine Management for Energy Efficient Cloud Data Centers*, Sustainable Computing: Informatics and Systems, May 2018.

[15] Ziqian Dong, Ning Liu and Roberto Rojas-Cessa, *Greedy scheduling of tasks with time constraints for energy-efficient cloud-computing data centers*, Journal of Cloud Computing: Advances, Systems and Applications, pp. 2-14, March 2015.

[16] Mateusz Zotkiewicz, Mateusz Guzek, Dzmitry Kliazovich, *Minimum Dependencies Energy-Efficient Scheduling in Data Centers*, IEEE Transactions on Parallel and Distributed Systems, Vol. 27, No. 12, pp. 1-14, December 2016.

[17] Marco Polverini, Antonio Cianfrani, Shaolei Ren, Athanasios V. Vasilakos, *Thermal-Aware Scheduling of Batch Jobs in Geographically Distributed Data Centers*, IEEE Transactions on Cloud Computing, Vol. 2, No. 1, pp. 71-84, January-March 2014.

[18] P. Wieder, J.M Butler, W. Theilmann, R. Yahyapour, *Service Level Agreements for Cloud Computing,* Springer Science+Business Media, LLC. pp. 43–68, 2011. ISBN 978-1-4614-1614-2.

[19] Anton Beloglazov, Jemal Abawajy, Rajkumar Buyya., *Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing*, Future Generation Computer Systems, Vol. 28, No. 5, pp. 755–768, May 2012.

[20] Y C Lee, A Y Zomaya., *Energy Conscious Scheduling for Distributed Computing Systems under Different Operating Conditions,* IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 8, pp. 1374-1381, 2011.

[21] Chia-Ming Wu, Ruay-Shiung Chang, Hsin-Yu Chan, *A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters*, Future Generation Computer Systems, Vol. 37, pp. 141-147, July 2014.

[22] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose and Rajkumar Buyya, *CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms*, Software – Practice and Experience, Volume 41, Issue 1, pp. 23-50, January 2011.

Authors Profile

Mrs.K.Sutha is currently pursuing a doctoral degree in Computer Science from Bharathiar University, Coimbatore, India. She is working as an Assistant Professor in Dr.MGR Janaki College of Arts and Science for Women, Chennai, India. She received her Master's degree in Computer Applications from Anna University, Chennai, India in 2009. She is a Lifetime Member of Indian Society for Technical Education (ISTE), New Delhi since 2011. She has published research papers in reputed international journals including Thomson Reuters (SCIE & Web of Science) and conferences including IEEE. Her main research work focuses on energy-efficient resource scheduling and load balancing in cloud computing.

Dr.G.M.Kadhar Nawaz is currently working as a Director in the department of Master of Computer Applications, Sona College of Technology, Salem, India. He has more than 20 years of teaching and research experience in reputed engineering colleges. He has published around 50 research papers in highly indexed SCI, SCIE and SCOPUS journals. He received his Ph.D in Computer Science from Periyar University, Salem, India in 2007 and his area of research includes digital image processing and steganography. He is a Professional Member of Indian Society for Technical Education (ISTE), New Delhi.