

A Comparative Analysis of Task Scheduling Algorithms of Virtual Machines in Cloud Environment

Saleh Atiewi, Salman Yussof and Mohd Ezanee

College of Information Technology, Tenaga National University, Kajang, Malaysia

Article history

Received: 25-07-2015

Revised: 11-08-2015

Accepted: 15-09-2015

Corresponding Author:

Saleh Atiewi

College of Information
Technology, Tenaga National
University, Kajang, Malaysia
Email: s.atiewi@gmail.com

Abstract: Cloud computing is an interesting and beneficial area in modern distributed computing. It enables millions of users to use the offered services through their own devices or terminals. Cloud computing offers an environment with low cost, ease of use and low power consumption by utilizing server virtualization in its offered services (e.g., Infrastructure as a Service). The pool of Virtual Machines (VMs) in a cloud computing Data Center (DC) needs to be managed through an efficient task scheduling algorithm to maintain quality of service and resource utilization and thus ensure the positive impact of energy consumption in the cloud computing environment. In this study, an experimental comparative study is carried out among three task scheduling algorithms in cloud computing, namely, random resource selection, round robin and green scheduler. Based on the analysis of the simulation result, we can conclude which algorithm is the best for scheduling in terms of energy and performance of VMs. The evaluation of these algorithms is based on three metrics: Total power consumption, DC load and VM load. A number of experiments with various aims are completed in this empirical comparative study. The results showed that there is no algorithm that is superior to the others. Each has its own pros and cons. Based on the simulation performed, the green scheduler gives the best performance with respect to energy consumption. On the other hand, the random scheduler showed the best performance with respect to both VM and DC load. The round robin scheduler gives better VM and DC load than the green scheduler but have more energy consumption than both random and green schedulers. However, since the RR scheduler distributes the tasks fairly, the network traffic is balanced and neither the server nor the network node will get overloaded or congested.

Keywords: Cloud Computing, Scheduling, Energy Efficient, GreenCloud, Virtual Machine

Introduction

In cloud computing, the term “cloud” is a metaphor for the Internet (Maggiani, 2009). A cloud shape is used in network diagrams to conceal the Internet’s flexible topology and abstract its underlying infrastructure (Jin *et al.*, 2010). Cloud computing utilizes the Internet to deliver different computing services, including software, hardware and programming environments, while keeping users unaware of the underlying infrastructure and security. Various experts have defined cloud computing from different perspectives. The most relevant cloud computing definition in this study is from (Vaquero *et al.*, 2008), who defined clouds as “a

large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically re-configured to adjust to a variable load (scale), allowing for optimum resource utilization. This pool of resources is typically exploited by a pay-per-use mode which guarantees are offered by the infrastructure provider by means of customized Service Level of Agreement (SLA).”

From the above definition, cloud computing can be depicted as a set of Data Center (DCs) that connect to the Internet to offer their services. These DCs are based on the virtualization of their infrastructure, with the Virtual Machine (VM) as the basic unit of computation. In general, DCs offer hardware services (i.e., VM for

computations) or software services, which are provided by mutual agreements through an SLA contract and are charged based on a per-use pricing method (Vaquero *et al.*, 2008). The above definitions indicate the need for a scheduling algorithm in the DC that finds the VM with the ability to meet client requirements. The VM must have sufficient resources, such as CPU, RAM and storage, to handle user tasks.

There are two purposes held by the majority of scheduling algorithms: To enhance the service quality when carrying out the tasks and supplying the expected output on time, other than to keep the efficiency and fairness for all tasks assigned intact (Mohialdeen, 2013).“Fig. 1” demonstrates the recommended cloud framework, with its established four tiers: Cloud users, cloud DC, network infrastructure and connected hosts.

As shown in “Fig. 1”, cloud users send their tasks to a DC where the tasks are queued in the main DC queue. The DC controller maps the submitted tasks to the host that suits the requirements (VM load or host queue size). All the tasks must be passed before they can proceed to the second layer (DC network layer), which consists of a set of routers and links. All the layers have their own energy monitors for monitoring energy consumption on each layer (Mehdi *et al.*, 2011).

The objectives of this work are to analyze and investigate three task scheduling algorithms, where they are Round Robin (RR), random resource selection and green scheduler. The evaluation will be based on how able they are in delivering quality service for the tasks and their total consumed energy in the cloud computing environment. Furthermore, the study aims to observe the behavior of these scheduling algorithms and determine the most appropriate scheduling algorithm in cloud environment.

Task Scheduling in a Cloud Computing DC

Task scheduling is one of the most important roles in cloud computing environment (Foster *et al.*, 2008). Scheduling primarily aims to maximize the resource use and minimize the process time of the tasks. All tasks should be balanced by a task scheduler to maintain quality of service, efficiency and fairness (Mohialdeen, 2013).

The efficient use of task scheduling aims to produce less response time so the submitted tasks will be done within the stipulated time. It will also result in additional tasks being submitted from cloud users, which ultimately accelerates the business performance and efficient resource utilization of the cloud system (Vaquero *et al.*, 2008) (Bilgaiyan *et al.*, 2015).

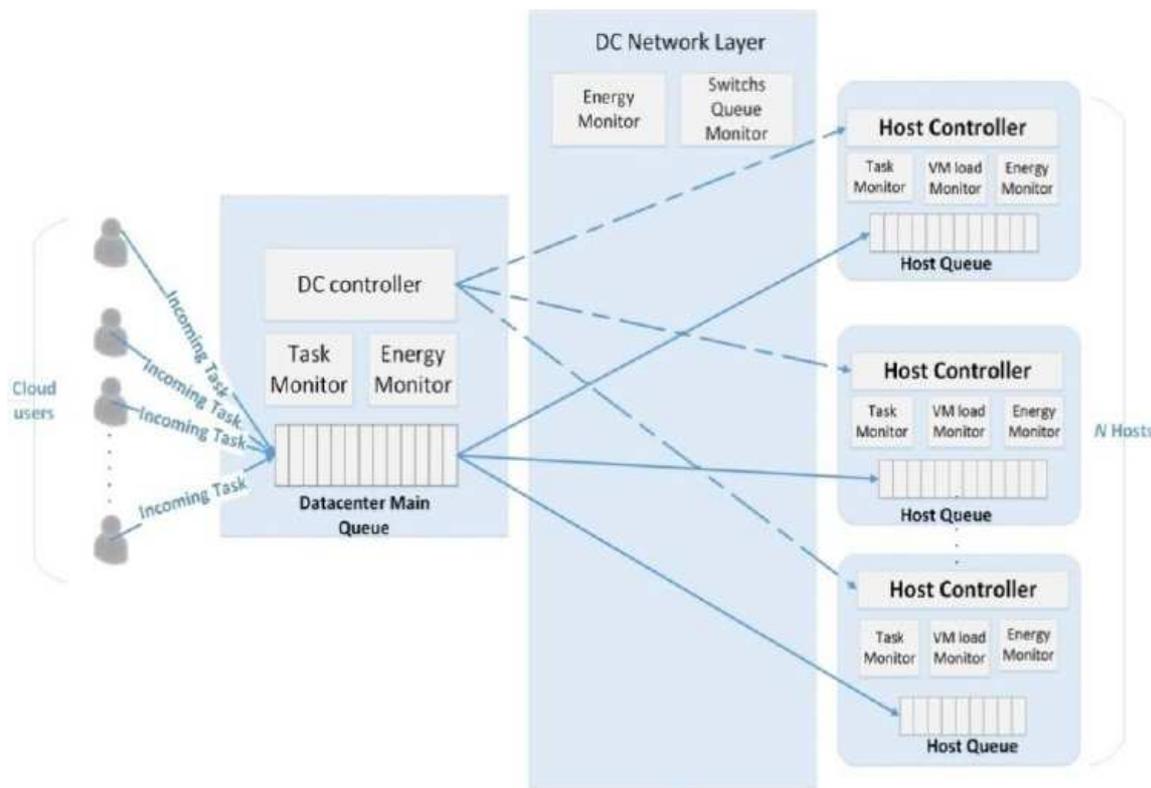


Fig. 1. Proposed DC framework (inspired by Mehdi *et al.*, 2011)

VM Scheduling

The complimentary setting in scheduling refers to the set of processes or tasks planned as indicated in the particular requirements and used algorithm. According to the requirement fulfilled with the requested resources (i.e., RAM, memory, bandwidth, etc.), VM scheduling algorithms function to schedule the VM requests to the Physical Machines (PMs) of a specific DC (Prajapati, 2013).

A scheduling algorithm generally works in three levels: In the first level, the appropriate PM is identified for the set of VMs; in the second level, the proper provisioning scheme is determined for the VMs; and in the third level, the tasks are scheduled on the VMs (Frincu *et al.*, 2013).

VM Scheduling Algorithms

This section will dwell further into the VM scheduling algorithms; or in particular, those that serve

to optimize different aspects, for example time, cost, energy and security. Algorithms that provide VMs with the perspective of the neighboring VMs or nodes security are scarce.

Random VM Scheduling Algorithm

The random resource selection algorithm has the notion of assigning the preferred task in a random manner to the available VM. The status of the VM is dismissed, even if the VM carries a load that is either heavy or light. This action can lead to heavy-load VM; thus, the task will propel a long waiting time prior to it being served. As shown in some cases, the task may fail in effect because the deadline is overdue. This algorithm is not really complex as it does not necessitate any overhead or pre-processing. “Fig. 2” illustrates the process of giving tasks to any VMs available (Liu *et al.*, 2013).

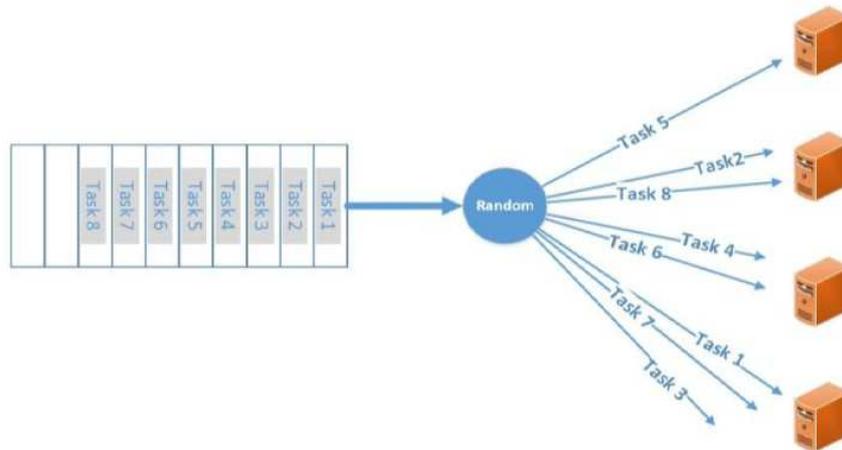


Fig. 2. Random resource selection algorithm for selecting VM

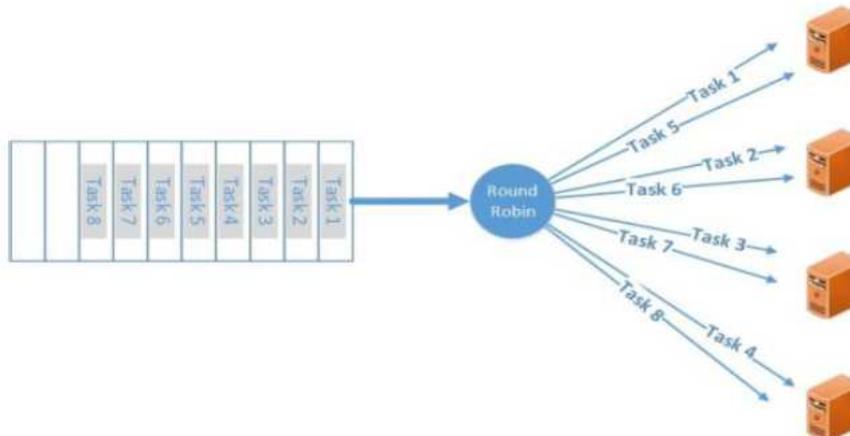


Fig. 3. RR Algorithm for selecting VM

RR VM Scheduling Algorithm

The RR task scheduling algorithm that has been contemplated in this study assigns the selected tasks over the available VMs in a round-robin order, where each task is equally administered. It is the idea of this algorithm to send the tasks chosen to the available VMs in round-robin. Figure 3 represents the mechanism of the RR task scheduling algorithm. Whichever pre-processing, overhead, or scanning of the VMs to name the task executor is not needed by the algorithm (Agarwal and Jain, 2014). Since the RR scheduling algorithm can distribute tasks fairly among all servers, the load balancing is achieved, while congestion and delay can well be averted. Furthermore, there is also the possibility that failed task is minimized (Mathew *et al.*, 2014).

Green VM Scheduling Algorithm

The workloads arriving at the DC are scheduled to be carried out by the energy-aware “green” scheduler. This “green” scheduler collects the workloads in the minimal computing servers. To explain the high-performance computing workloads, the scheduler tracks the buffer occupancy of network switches on the path in a continuous manner. Whenever congestion takes place, the scheduler stays away from the congested routes even if they are led to the servers that can meet the computational requirement of the workloads. The servers idled are set into sleep mode (dynamic shutdown DNS scheme),

whereas the supply voltage is minimized Dynamic Voltage Frequency Scaling (DVFS scheme) on the under-loaded servers (Kliazovich *et al.*, 2012) (Lin *et al.*, 2015).

Empirical Study

In this part of the work, we present a case study that simulates an energy-aware DC in three-tier architecture. Simulation is the process of emulating the actual system. If we are presented with the difficulties in testing the recommended system in a real system, a simulation was run for performance evaluation using the GreenCloud simulator (Kliazovich *et al.*, 2012). The GreenCloud Simulator was built upon as an extended work of the network simulator Ns2 (Issariyakul and Hossain, 2011) for the study of cloud computing environments. It supplies a comprehensive fine-grained modeling of the energy used up by the elements found in the DC, such as servers, switches and links. Moreover, GreenCloud performs a detailed investigation on the workload distribution (Atiewi and Yussuf, 2014).

The farm of servers in current DCs contains more than 100,000 hosts, where about 70% of the communications are performed within (Audzevich *et al.*, 2012). The most frequently applied DC architecture is the three-tier architecture. “Figure 4” represents the three layers of the DC architecture, which are the core network, aggregation network and access network (Baliga *et al.*, 2011).

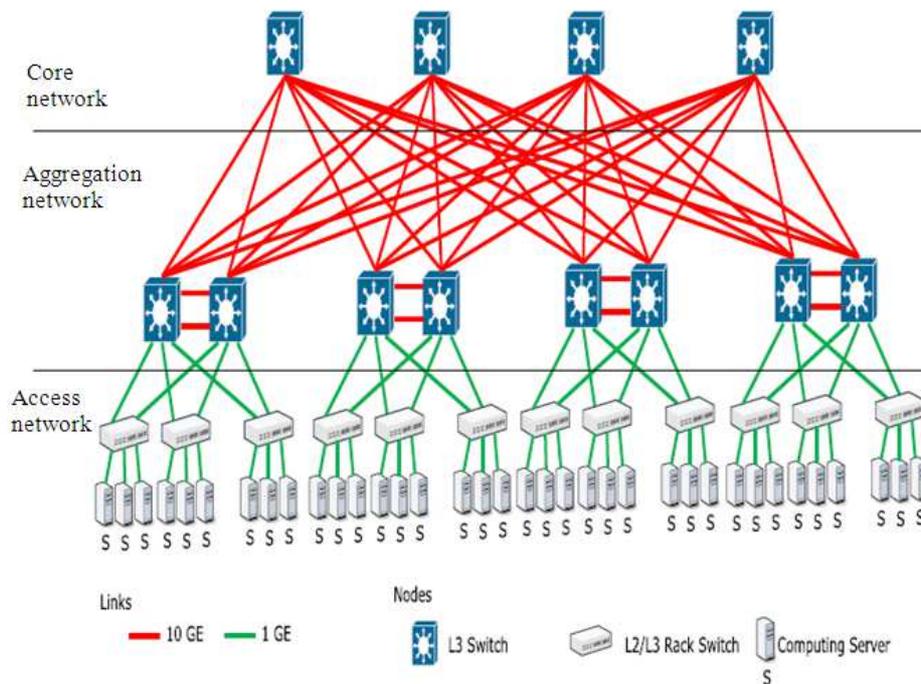


Fig. 4. Three-tier DC architecture

Table 1. Simulation setup parameters

Parameters	Value
DC type	Three-tier topology
# of core switches	1
# of aggregation switches	2
# of access switches	3
# of servers	144
Access links	1 Gb
Aggregation links	1 Gb
Core links	10 Gb
DC load	0.1 0.2 0.3 0.9 1.0
Simulation time	60 min
Power management in server	DVFS and DNS
Task size	100000 bit
DC computing capacity	144014400 MIPS
Task generation rate	2456/60=40.93

The three-tier DC topology opted for the simulations contains 144 servers which are set into three racks (48 servers per rack) linked together using one core switch, two aggregation switches and three access switches. The network links that connect the core and aggregation switches have a bit rate of 10 Gb/s. The network links which connect the aggregation and access switches, also the access links connecting computing servers to the top-of-rack switches have a bit rate of 1 Gb/s. The propagation delay of all links is fixed to 3.3 ms. Table 1 summarizes the simulation setup parameters.

The experiment was conducted to compare the amount of power consumption in hard deadline tasks of the three scheduling algorithms and find which scheduler can execute a set of tasks with minimum power consumption while maintaining the SAL.

Simulation Results

Three experiments were carried out in this study. The GreenCloud simulator was used in all the experiments to analyze several performance metrics. The first performance metric is the DC load which represents the percentage of computing resources allocated for incoming tasks with the respect to the data center capacity. The load should be between 0 and 100%. The load close to 0 represents an idle data center, while the load equal to 100% would saturate data center (Kliazovich *et al.*, 2012). The second one is the VM load which is equal to the ratio of current VM load to the maximum computing capability (Kliazovich *et al.*, 2013). The third one is the total energy consumption in DC which represents sum of

energy consumed by both servers and switches (Kliazovich *et al.*, 2012).

Figure 5 shows the distribution of 2456 tasks over 144 servers in the DC. In this figure, the green scheduling algorithm sends more tasks to a lesser number of servers. The behavior of the RR scheduling algorithm is also observed to scan numerous servers and send the task to all of them. Meanwhile, the random resource selection algorithm constantly varies the number of tasks among all the servers.

Figure 6 describes the amount of power required to execute the set of tasks over three different algorithms. The worst algorithm for power consumption is RR because of its ability to distribute the load to all servers, which leads to a request for more servers and consumes more power. The random resource selection algorithm consumes less energy than the RR algorithm, whereas the green scheduling algorithm consumes less power than both algorithms because of task distribution over the servers, as shown in “Fig. 5”.

Figure 7 depicts the DC load under different simulation load scenarios, starting from 10% load and ending with 100% load. In the figure, all the algorithms maintain the same load from 10 to 30%. At 40% load, both RR and green algorithm start to have more load than the random resource selection algorithm, which is attributed to the complexity of both algorithms compared to the random algorithm.

Figure 8 demonstrates the VM average load at a variety of input loads (10-100%). Owing to the nature of the green scheduler algorithm which tends to classify the workloads in the smallest possible amount of computing servers, we can see from “Fig. 8” that almost half of VMs obtained load from 90% and further lessened to 50%, where the second half obtained less than 50% down to 0% for the ideal server (servers in the sleep mode). This is due to the less number of the total tasks. Contrastingly, the RR scheduler retains the load for all VM of approximately 50%. This is due to the fact that the tasks are equally distributed among all the VMs. At the same time, the random algorithm produces a vacillating load between 30 and 55% as the algorithm varies the number of tasks among all the servers randomly and yet, constantly. The RR scheduler distributes computing and communication loads equally among servers and switches; thus, the network traffic is balanced and no server is loaded more than it should. Nevertheless, one flaw is that no server or network switch is left idle for powering down, simultaneously making the round-robin scheduler the least energy-efficient.

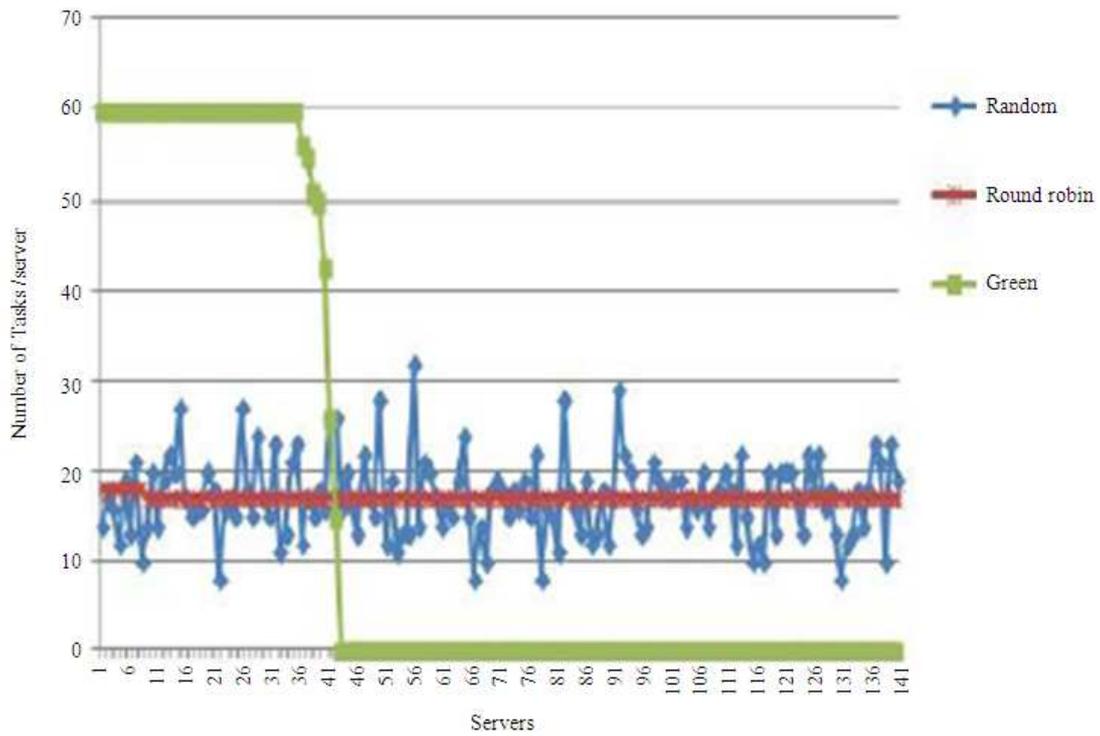


Fig. 5. Tasks distribution of over 144 servers

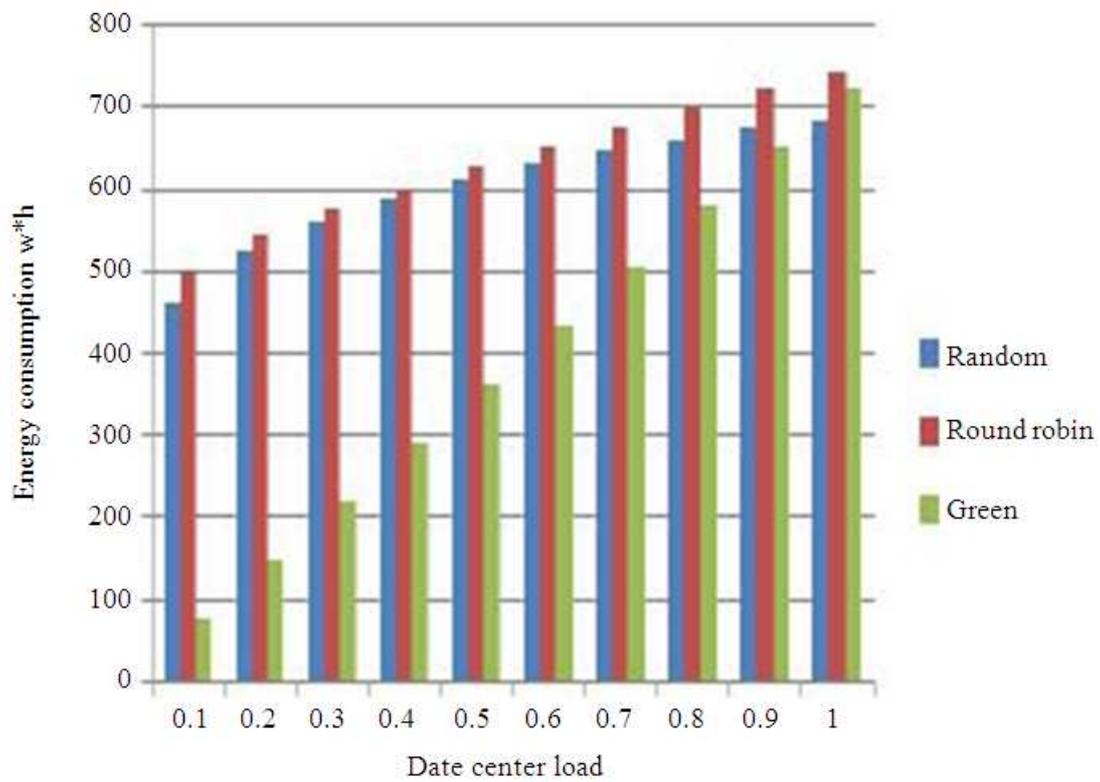


Fig. 6. Amount of consumed energy at different DC loads

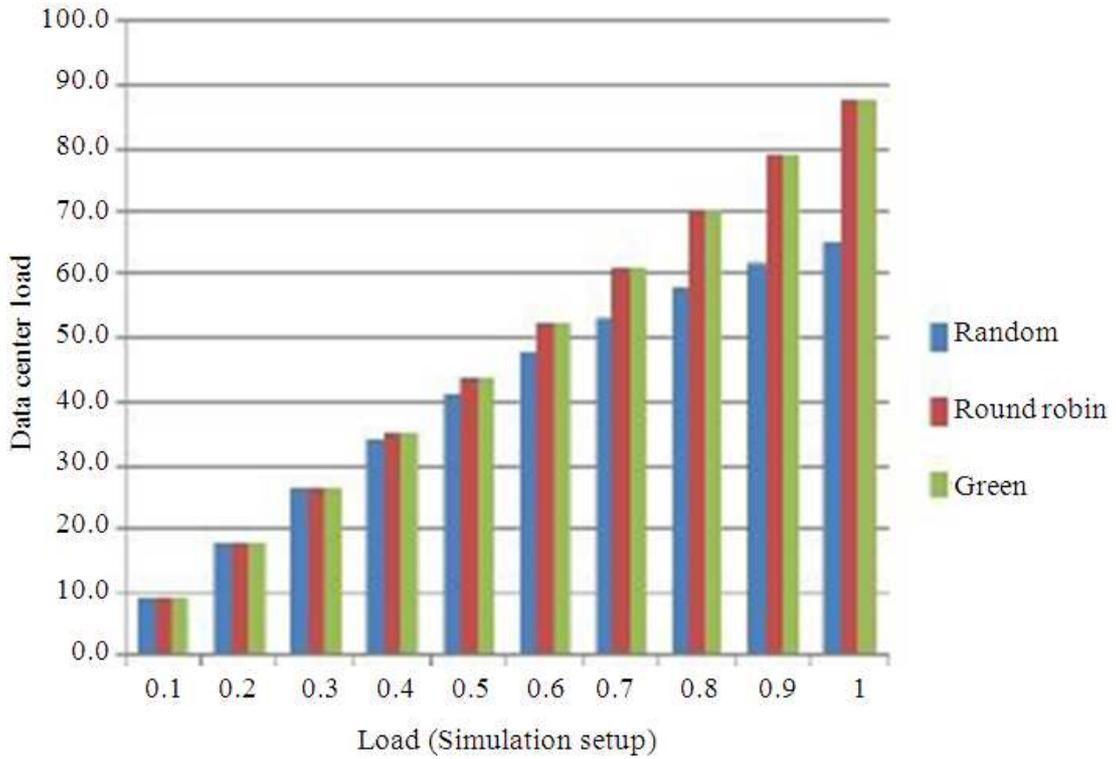


Fig. 7. DC load under different simulation load scenarios

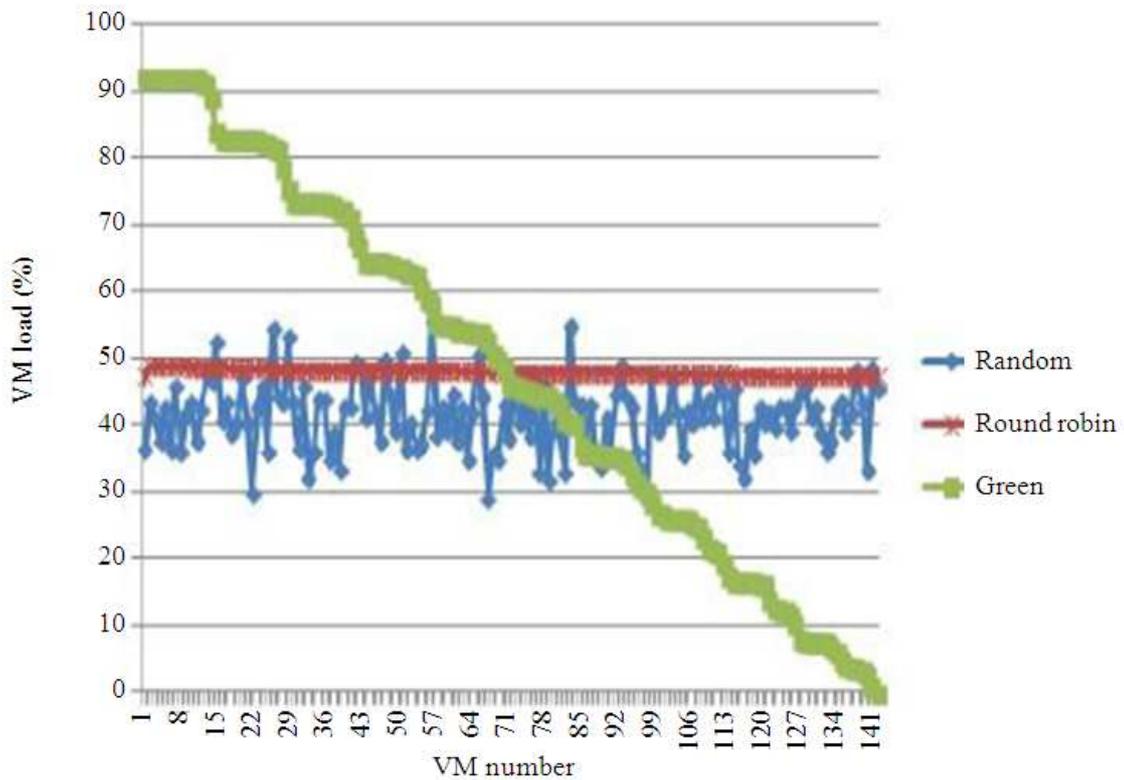


Fig. 8. VM average load at various input load (10-100%)

Conclusion

In this study, the behavior of three task scheduling algorithms, namely, RR, random resource selection and green scheduler, were investigated and examined under the cloud computing environment using the GreenCloud simulator. An extensive evaluation of these task scheduling algorithms was conducted by focusing on the energy consumption, DC load and VM load. The simulation results revealed that each scheduling algorithm has its own pros and cons. Green scheduling algorithm consumed less energy than RR and random scheduling algorithm. The experiments showed that spreading the load over multiple servers can increase power consumption more than expected. Therefore RR scheduling algorithm consumed more energy than Green and Random scheduling algorithm. The results also showed that the complexity of the algorithm can increase the DC load. Therefore green scheduling algorithm and RR have more DC load than random scheduling algorithm. The experiments also showed that random algorithm has lower VM load than both green scheduler and RR. However, with respect to load balancing, the RR scheduler performed the best compared to the other algorithms because it distributes the tasks fairly to all VMs. Finally, the Random algorithm performed the worst compared to the other scheduling algorithms with regard to load balancing. This is because the random algorithm randomly assigns the selected tasks to the available (VM). The algorithm does not take into considerations the VM status whether it was under high or low load. On the basis of these results, no single scheduling algorithm can provide superior performance with respect to various types of quality services.

Acknowledgement

We thank Tenaga National University and faculty of information and communication technology for supporting us.

Funding Information

The authors have no support or funding to report.

Author's Contributions

Saleh Atiewi: Design the model, participated in all experiments coordinated the data-analysis and contributed to the writing of the manuscript.

Salman Yussof: Design the research plan and organized the study.

Mohd Ezanee: Contribute in drafting the article or reviewing it critically for significant intellectual content.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

- Agarwal, D. and S. Jain, 2014. Efficient optimal algorithm of task scheduling in cloud computing environment. *Int. J. Comput. Trends Technol.*, 9: 344-349.
- Atiewi, S. and S. Yussof, 2014. Comparison between cloud SIM and green cloud in measuring energy consumption in a cloud environment. *Proceedings of the 3rd International Conference on Advanced Computer Science Applications and Technologies*, Dec. 29-30, IEEE Xplore Press, Amman, pp: 9-14. DOI: 10.1109/ACSAT.2014.9
- Audzevich, Y., A. Moore, A. Rice, R. Sohan and S. Timotheou *et al.*, 2012. Intelligent Energy Aware Networks. In: *Handbook of Energy-Aware and Green Computing*, Ahmad, I. and S. Ranka (Eds.), Chapman and Hall/CRC, Boca Raton, ISBN-10: 1439850402, pp: 239-282.
- Baliga, J., R.W.A. Ayre, K. Hinton and R.S. Tucker, 2011. Green cloud computing: Balancing energy in processing, storage and transport. *Proc. IEEE*, 99: 149-167. DOI: 10.1109/JPROC.2010.2060451
- Bilgaiyan, S., S. Sagnika, S. Mishra and M. Das, 2015. Study of task scheduling in cloud computing environment using soft computing algorithms. *Int. J. Modern Educ. Comput. Sci.*, 7: 32-38. DOI: 10.5815/ijmecs.2015.03.05
- Foster, I., Y. Zhao, I. Raicu and S. Lu, 2008. Cloud computing and grid computing 360-degree compared. *Proceedings of the Grid Computing Environments Workshop*, Nov. 12-16, IEEE Xplore Press, Austin, pp: 1-10. DOI: 10.1109/GCE.2008.4738445
- Frincu, M.E., S. Genaud and J. Gossa, 2013. Comparing provisioning and scheduling strategies for workflows on clouds. *Proceedings of the IEEE 27th International Parallel and Distributed Processing Symposium Workshops and PhD Forum*, May 20-24, IEEE Xplore Press, Cambridge, MA., pp: 2101-2110. DOI: 10.1109/IPDPSW.2013.55
- Issariyakul, T. and E. Hossain, 2011. *Introduction to Network Simulator NS2*. 2nd Edn., Springer, U.S., ISBN-10: 978-1-4614-1406-3, pp: 512.
- Jin, H., S. Ibrahim, T. Bell, W. Gao and D. Huang *et al.*, 2010. *Cloud Types and Services*. In: *Handbook of Cloud Computing*, Borko, F. and A. Escalante (Eds.), Springer, US., ISBN-10: 978-1-4419-6524-0, pp: 335-355.

- Kliazovich, D., P. Bouvry and S.U. Khan, 2012. GreenCloud: A packet-level simulator of energy-aware cloud computing data centers. *J. Supercomput.*, 62: 1263-1283.
DOI: 10.1007/s11227-010-0504-1
- Kliazovich, D., P. Bouvry and S.U. Khan, 2013. Simulation and Performance Analysis of Data Intensive and Workload Intensive Cloud Computing Data Centers. In: *Optical Interconnects for Future Data Center Networks*, Kachris, C., K. Bergman and I. Tomkos (Eds.), Springer-Verlag New York, ISBN-10: 978-1-4614-4630-9, pp: 47-63.
- Lin, X., Y. Wang, Q. Xie and M. Pedram, 2015. Task scheduling with dynamic voltage and frequency scaling for energy minimization in the mobile cloud computing environment. *IEEE Trans. Services Comput.*, 8: 175-186.
DOI: 10.1109/TSC.2014.2381227
- Liu, N., Z. Dong and R. Rojas-Cessa, 2013. Task scheduling and server provisioning for energy-efficient cloud-computing data centers. *Proceedings of the IEEE 33rd International Conference on Distributed Computing Systems Workshops*, Jul. 8-11, IEEE Xplore Press, Philadelphia, PA., pp: 226-231.
DOI: 10.1109/ICDCSW.2013.68
- Maggiani, R., 2009. Cloud computing is changing how we communicate. *Proceedings of the IEEE International Professional Communication Conference*, Jul. 19-22, IEEE Xplore Press, Waikiki, HI., pp: 1-4. DOI: 10.1109/IPCC.2009.5208703
- Mathew, T., K.C. Sekaran and J. Jose, 2014. Study and analysis of various task scheduling algorithms in the cloud computing environment. *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, Sept. 24-27, IEEE Xplore Press, New Delhi, pp: 658-664.
DOI: 10.1109/ICACCI.2014.6968517
- Mehdi, N.A., A. Mamat, A. Amer and Z.T. Abdul-Mehdi, 2011. Minimum completion time for power-aware scheduling in cloud computing. *Proceedings of the Developments in E-systems Engineering*, Dec. 6-8, IEEE Xplore Press, Dubai, pp: 484-489.
DOI: 10.1109/DeSE.2011.30
- Mohialdeen, I.A., 2013. Comparative study of scheduling algorithms in cloud computing. *J. Comput. Sci.*, 9: 252-263.
DOI: 10.3844/jcssp.2013.252.263
- Prajapati, K.D., 2013. Comparison of virtual machine scheduling algorithms in cloud computing. *Int. J. Comput. Appl.*, 38: 12-14.
DOI: 10.5120/14523-2914
- Vaquero, L.M., L. Rodero-Merino, J. Caceres and M. Lindner, 2008. A break in the clouds: Towards a cloud definition. *ACM SIGCOMM Comput. Commun. Rev.*, 39: 50-55.
DOI: 10.1145/1496091.1496100