AN EXTENDED MIN-MIN SCHEDULING ALGORITHM IN CLOUD COMPUTING

Article *in* i-manager's Journal on Civil Engineering · December 2018

AN EXTENDED MIN-MIN SCHEDULING ALGORITHM IN CLOUD COMPUTING

By

J. Y. MAIPAN-UKU * A. MISHRA ** A. ABDULGANIYU *** A. ABDULKADIR ****

*,*** Lecturer, Department of Computer Science, Facuty of Natural Sciences, Ibrahim Badamasi Babangida University Lapai (IBBUL), Niger State, Nigeria.
** Senior Lecturer, Department of Computer Science, Baze University, Abuja, Nigeria.
**** **** Lecturer, Department of Mathematics, Faculty of Natural Sciences, Ibrahim Badamasi Babangida Lapai (IBBUL), Niger State, Nigeria.

Date Received: 11/01/2019 Date Revised: 06/02/2019 Date Accepted: 18/03/2019

ABSTRACT

Cloudlet scheduling seems to be the most fundamental problem of cloud computing as per Infrastructure as a Service (IaaS). Proper scheduling in cloud lead to load balancing, minimization of makespan, and adequate resources utilization. To meet consumers' expectations, the execution of cloudlet simultaneously is required. Many algorithms have been implemented to solve the cloud scheduling problem. This include Min-Min which gave priority to cloudlet with minimum completion time. Min-Min scheduling algorithm has two clear weaknesses; a high value of makespan being generated and low resource utilization. To address these problems, this research proposes an Extended Min-Min Algorithm which assigns cloudlet base on the differences between maximum and minimum execution time of cloudlets. CloudSim was used to implement and compare the performances of the proposed algorithm with the benchmarks. The results of the extensive experiments show that the proposed algorithm is able to perform better in terms of makespan minimization compared to the existing heuristics.

Keywords: Component, Cloud Computing, Min-Min, EMinMin, Makespan.

INTRODUCTION

The advancement of technology and how businesses and organizations run in this cloud era are a clear evidence that cloud computing has come to stay as platform for businesses and individual computing needs. Cloud computing has gained massive attention and popularity (Zhang, Huang, & Wang, 2016) in the IT sector because majority of organizations, companies, and individuals are unable to buy resources (hardware and software) due to the high cost involved in getting them (Kaur & Sharma, 2014). Cloud computing is increasingly becoming a preferable platform for organizations and individuals because; it has been able to lower the barriers of the high costs associated with providing the requisite software and hardware (Pasupuleti, Ramalingam, & Buyya, 2016). Cloud computing has converted storage, for example, into a 'pay as you go service'. This has enabled individuals and organizations not to require the purchase of their own resources (hardware and software) or paying for maintenance cost (Lin, Thai, Wang, & Lai, 2015).

Cloudlet scheduling seems to be the most fundamental problem of cloud computing as per infrastructure as a Service (IaaS). Proper scheduling in cloud lead to load balancing, minimization of makespan, and adequate Virtual Machines (vm) utilization. To meet consumers' expectations, the execution of cloudlet simultaneously is required. Many algorithms have been implemented to solve the cloud scheduling problem (Duan, Prodan, & Li, 2014) Min-Min attempts to tackle these problems, but produces high makespan and poor resource utilization when number of tasks with low completion time are more than tasks with high completion (i.e. it gives priority to tasks with minimum completion time).

This research work addressed the issues of Min-Min by developing a suitable approach called an Extended Min-Min (EMin-Min) that is capable of selecting and assigning cloudlets to appropriate Virtual Machines. The main objective of this research work is to optimize jobs' finishing time.

1. Task Scheduling in Cloud

Scheduling cloudlet in cloud computing is NP-hard problem (Masdari, ValiKardan, Shahi, & Azar, 2016), which has been a platform constituted by potentially large numbers of users from different environments and with different needs and demands on thousands or millions of Virtual Machines waiting for their turn to be served (Keshk, El-Sisi, & Tawfeek, 2014; Huang & Huang, 2010). It is the wish of every cloud service provider to ensure that every available resource is fully utilized to avoid resources being idle (Choudhary & Peddoju, 2012).

The idea of task scheduling is to accomplish a high level of system throughput by dispatching a job to a resource that has the highest capacity to execute it within a shorter period of time and also to match application needs by consumer's/user demands with the available resources under a given cloud standard (Mohammadi, Jamali, & Bekravi, 2014; He, Sun, & Von Laszewski, 2003).

1.1 Min-Min Algorithm

Min-Min scheduling algorithm is used in cloud and grid computing environments in order to minimize makespan, cost, and maximize profit and resource utilization. This is done by selecting a cloudlet in the cloudletlist with the lowest execution time and assigning it to virtual machine that produces its minimum completion time. This algorithm gave priority to tasks with minimum completion time (Elzeki, Rashad, & Elsoud, 2012; Santhosh & Manjaiah, 2014).

However, assigning cloudlets in this manner lead to increase of total response time of the system (Santhosh & Manjaiah, 2014) when cloudlets with minimum completion time are much more in number. Thus, making it inefficient.

Researchers have recommended Min-Min as one of the best methods of scheduling in cloud and grid computing. These researchers have contributed tremendously by making Min-Min an efficient task scheduling algorithm. The efficiency of this algorithm has made cloud computing acceptable in both educational institutions and industries as a preferred platform for data storing and information dissemination.

1.2 Max-Min Algorithm

This algorithm works in the opposite direction of Min-Min algorithm. While Min-Min algorithm first selects and assigns cloudlet to a Virtual Machine (vm) with minimum completion time before considering cloudlets with maximum execution time, Max-Min does the opposite by selecting and assigning cloudlets with maximum completion time first.

Max-Min algorithm gave priority to cloudlets with maximum completion time (Fazel et al., 2014). However, its disadvantage is that it sometimes leaves the short tasks unattended or waited for so long before execution (Liu, Li, & Xu, 2013) when we have much more number of cloudlets with maximum completion time.

1.3 Round Robin

Round Robin is one of the most frequently used algorithms in resource allocation and task scheduling. In Round Robin, a dedicated time slot is allocated to each job waiting in a queue to be scheduled by the scheduler (Kaur & Kaur, 2015). This means that no task will be allocated to a resource for more than the allocated time and if a task is not able to complete within the allocated time, it will be preempted by another task and sent back in the queue to give way for other tasks.

1.4 First Come First Served (FCFS) Algorithm

Analysis made from various researchers on resource allocation and task scheduling algorithms has identified First-Come-First-Served (FCFS) as the simplest scheduling algorithm as far as task scheduling and resource allocation are concerned. FCFS is a non-preemptive scheduling algorithm that assigns tasks to resources based on their arrival time. It is non-preemptive in the sense that it does not release the resource to other tasks until it finishes scheduling (Keshk et al., 2014). One disadvantage of FCFS is that its turnaround and response time is quite low because tasks are not preempted. The smaller tasks with the shortest execution time in the queue may have to wait for the larger task with large execution time ahead to finish (Agarwal & Jain, 2014).

2. Related Works

Cloud computing is becoming the most attractive platform for doing businesses because of its vast benefits, which include multi-tenancy, scalability, security, and on-demand applications that are made available to cloud users via the Internet. Task scheduling is however one of the fundamental concerns in cloud and grid computing. Task scheduling is done by selecting a set of tasks from a list of tasks to be executed on available cloud resources that has the ability of executing them within a shortest period of time for effective and efficient utilization of computational power (Rana, Bilgaiyan, & Kar, (2014).

The main reason behind task scheduling in cloud and grid computing is to increase system throughput by matching tasks with resources that has the capacity to execute them for effective cloud service delivery.

The past years have witnessed an exciting movement for cloud task scheduling and resource allocation. It has maintained its growth in the computer networks, telecommunication and the storage market, and improved its concentration with the introduction of innovative and groundbreaking scheduling algorithms by various researches such as, Min-Min (Keshk et al., 2014; Konjaang, Maipan-uku, & Kubuga, 2016) Max-Min (Elzeki et al., 2012; Santhosh & Manjaiah, 2014; Kaur & Kaur, 2015; Konjaang, Ayob, & Muhammed, 2017), Min-Max (Zhou & Zhigang, 2014), Game Theoretic (Duan et al., 2014; Wei, Vasilakos, Zheng, & Xiong, 2010), FCFS (Marphatia, Muhnot, Sachdeva, & Shukla, 2013), and the other algorithms have made resource allocation and task scheduling in the cloud more efficient include 'An improved league championship algorithm' in (Abdulhamid, Latiff, & Idris, 2015) and 'Symbiotic Organism Search optimization based task scheduling' in (Abdullahi & Ngadi, 2016). Though cloud computing has gone through significant reforms and transformations, however, the rapid growth in information and the increasing demand for cloud services in recent times calls for a proactive for provision of an optimal scheduling techniques to provide better and cost effective cloud services to meet the increasing demands and requirements of cloud consumers/users.

For example, in (Konjaang et al., 2016) an Enhanced Load Balanced Min-Min algorithm was studied. Weighted Round Robin is another approach presented in (Supreeth & Biradar, 2013) with the mandate of allocating all incoming tasks to idle VMs. The idea of Weighted Round Robin algorithm was conceived through the traditional Round Robin. The proposed weighted Round Robin allocates task to resources in Round Robin fashion based on the weight of the income request instead of the present load of the virtual machine as in case of the traditional Round Robin. Though, limited parameters were used in the analysis of result, but notwithstanding, the experimented results saw weighted round robin putting a better performance with regards to time.

Though these researchers have contributed immensely in making Min-Min algorithm more effective in terms of makespan optimization. Thus, the proposed approach aims at addressing it by introducing a new dimension to make Min-Min more effective and proactive in selecting and scheduling both task with minimum and maximum execution time simultaneously without any given priority.

3. Proposed Algorithm (EMIN-MIN)

An Extended Min-Min algorithm works on the basis of multiple parameters that consider both large cloudlets and small cloudlets. Unlike the traditional Min-Min algorithm, for all submitted workflows in the CloudletList, it calculates completion time $CT_{ij} = et_{ij}+rt_j$; for each cloudlet to determine the cloudlet with minimum and maximum completion time. Then, it selects a cloudlet (c_i) in the CloudletList and compares with MaxClt, if the c_i has maximum execution time, then it will assign c_i to the resource that produces its minimum completion time for execution, else it will assign MinClt to the resource that has the capability to execute it within a short time. The CloudletList and VM's ready time will be updated. The pseudo-code of the Extended Min-Min Algorithm is displayed in Figure 1.

Pseudo Code of EMin-Min algorithm

1.While there are cloudlets in CloudletList
2.for all submitted cloudlets in the set; c_i
3.for all VM_s; vm_j

- Calculate completion time (CT_{ij}) = et_{ij} + rt_j; (for each cloudlet in all Vms)
- find the cloudlet with minimum execution time (MinClt)
- find the cloudlet with maximum execution time (MaxClt)
- Select a cloudlet (c_i) in the CloudletList and compare with MaxClt
- If the c_i has maximum execution time, assign the c_i to the resource that produce it minimum completion time for execution

2. Else assign MinClt to the resource that produce it minimum execution time
3.End if
4.Update the CloudletList
5.Update ready time (rt_j) of the selected vmR_j
6.Update ct_{ij} for all c_i
7.End While

Figure 1. Pseudo Code of EMin-Min algorithm

4. Research Methodology

Due to the difficulties in implementing, testing, and evaluating the efficiency of the proposed algorithm with the existing ones using a testbed, CloudSim toolkit, a simulation environment for cloud computing was used to experiment and ascertain the performance of these algorithms. Some scientific workflows such as Inspiral and Montage was used as dataset. The simulation was run on Intel® Pentium (R) CPU, 500GB HDD and 4GB RAM on 64-bit operating system (Windows 10 Home). Table 1 shows the cloudsim configuration adopted.

4.1 Performance Metric

Makespan as one of the performance metrics is used to major the efficacy of the proposed Extended Min-Min algorithm with the standard Min-Min, Max-Min, Round Robin, and FCFS.

4.2 Makespan

Makespan is defined as the time that surfaces the newest task. This parameter shows the quality of assignment of resources from the executional time perspectives. It can be calculated as shown in the equations below;

If $C_i = c_1, c_2, c_3..., c_m$, is the set cloudlet submitted to the scheduler, and $VM_j = vm_1, vm_2, vm_3..., vm_n$, be the set of Virtual Machines available at the time of cloudlet arrival,

$$\text{Makespan} = \max\{\text{completion }[j] \mid j \text{ in Resource}\} \quad (1)$$

$$\text{Completion Time }(CT_{ij}) = (ET_{ij}) + (R_j) \quad (2)$$

where ET_{ij} is the expected execution time of the cloudlet c_i on machine vm_j and r_j is the ready time of vm_j i.e. the time when vm_j becomes ready to execute c_i.

Entities		Values
Workflows	Inspiral	50, 100, 1000
	Montage	50, 100, 1000
Data center		1
Virtual Machines		15
CPU/VM		1

Table 1. CloudSim Configurations

5. Results and Discussion

Tables 2 and 3 show the values of makespan generated by different algorithms considered in this research using Montage and Inspiral datasets and their respective graphs are shown in Figures 2 and 3. From the results, we can see that, the proposed algorithm (EMin-Min) substantially outperforms the standard algorithms in all scenarios. This is because the proposed algorithm was able to adequately assign cloudlets to the considerable VM. This signifies that the proposed algorithm is more efficient in choosing and allocating cloudlets to VM and able to optimize the finishing time (Makespan).

Conclusion and Future Work

Scheduling cloudlet in cloud computing is NP-hard problem has been a platform constituted by potentially large number of users from different environments with different needs and demands on thousands or millions of Virtual Machines waiting for their turn to be served.

Quality of scheduled cloudlets by different resource scheduler implies the level of participant satisfaction in cloud computing. Applying the proposed EMin-Min task

Montage Workflow		EMin-Min	Min-Min	Max-Min	RR	FCFS
Small	50	280.04	600.29	409.19	523.43	551.50
Large	100	463.03	640.94	650.50	701.24	799.17
Heavy	1000	4746.89	5269.62	5128.97	10172.42	5437.23

Table 2. Makespan Comparison of Different Algorithms Using Montage Workflow

Inspiral Workflow		EMin-Min	Min-Min	Max-Min	RR	FCFS
Small	50	3902.79	5308.40	4277.22	5500.15	6429.16
Large	100	6029.18	12869.69	7089.98	7649.11	9321.62
Heavy	1000	78187.49	78703.95	84269.07	48051.09	83895.72

Table 3. Makespan Compa

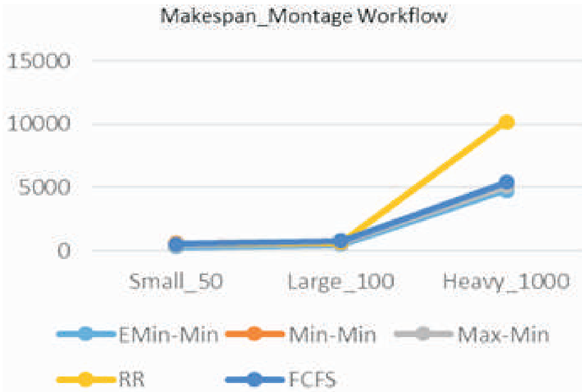

Figure 2. Makespan Comparison of different Algorithms using Montage workflow

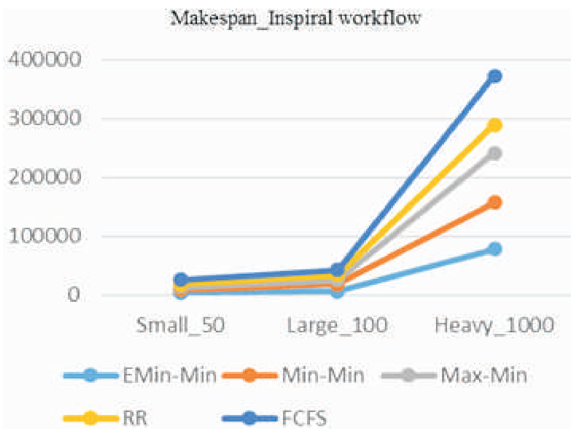

Figure 3. Makespan Comparison of different Algorithms using Inspiral workflow

scheduling algorithm, the experimental result shows a significant improvement of minimizing makespan when compared with the benchmarked algorithms.

The proposed approach has been critically analyzed and few limitations have been observed. Additionally, the simulation environment is semi-dynamic and it cannot reflect on the real cloud environment sufficiently, which promotes further research in the proposed area. In future research, some more factors like load balancing, Resource utilization, and cost will be taken into account.

References

[1]. **Abdulhamid, S. M., Latiff, M. S. A., & Idris, I. (2015).** Tasks scheduling technique using league championship algorithm for makespan minimization in IAAS cloud. *arXiv preprint arXiv:1510.03173.*

[2]. **Abdullahi, M., & Ngadi, M. A. (2016).** Symbiotic Organism Search optimization based task scheduling in cloud computing environment. *Future Generation Computer Systems*, 56, 640-650.

[3]. **Agarwal, D., & Jain, S. (2014).** Efficient optimal algorithm of task scheduling in cloud computing environment. arXiv preprint arXiv:1404.2076.

[4]. **Choudhary, M., & Peddoju, S. K. (2012).** A dynamic optimization algorithm for task scheduling in cloud environment. *International Journal of Engineering Research and Applications (IJERA)*, 2(3), 2564-2568.

[5]. **Duan, R., Prodan, R., & Li, X. (2014).** Multi-objective game theoretic schedulingof bag-of-tasks workflows on hybrid clouds. *IEEE Transactions on Cloud Computing*, 2(1), 29-42.

[6]. **Elzeki, O. M., Rashad, M. Z., & Elsoud, M. A. (2012).** Overview of scheduling tasks in distributed computing systems. *International Journal of Soft Computing and Engineering*, 2(3), 470-475.

[7]. **He, X., Sun, X., & Von Laszewski, G. (2003).** QoS guided min-min heuristic for grid task scheduling. *Journal of Computer Science and Technology*, 18(4), 442-451.

[8]. **Huang, Q. Y., & Huang, T. L. (2010, October).** An optimistic job scheduling strategy based on QoS for Cloud Computing. In *Intelligent Computing and Integrated Systems (ICISS), 2010 International Conference on* (pp. 673-675). IEEE.

[9]. **Kaur, G., & Sharma, E. S. (2014).** Optimized utilization of resources using improved particle swarm optimization based task scheduling algorithms in cloud computing. *International Journal of Emerging Technology and Advanced Engineering*, 4(6), 110-115.

[10]. **Kaur, N., & Kaur, K. (2015).** Improved max-min scheduling algorithm. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 17(3), 42-49.

[11]. **Keshk, A. E., El-Sisi, A. B., & Tawfeek, M. A. (2014).** Cloud task scheduling for load balancing based on intelligent strategy. *International Journal of Intelligent Systems and Applications*, 6(5), 25-36.

[12]. **Konjaang, J., Ayob, F. H., & Muhammed, A. (2017).** An optimized Max-Min scheduling algorithm in cloud

computing. *Journal of Theoretical & Applied Information Technology*, 95(9), 1916-1926.

[13]. Konjaang, J. K., Maipan-uku, J. Y., & Kubuga, K. K. (2016). An efficient Max-Min resource allocator and task scheduling algorithm in cloud computing environment. *arXiv preprint arXiv:1611.08864.*

[14]. Lin, Y. D., Thai, M. T., Wang, C. C., & Lai, Y. C. (2015). Two-tier project and job scheduling for SaaS cloud service providers. *Journal of Network and Computer Applications*, 52, 26-36.

[15]. Liu, G., Li, J., & Xu, J. (2013). An improved min-min algorithm in cloud computing. In *Proceedings of the 2012 International Conference of Modern Computer Science and Applications* (pp. 47-52). Springer, Berlin, Heidelberg.

[16]. Marphatia, A., Muhnot, A., Sachdeva, T., & Shukla, E. (2013). Optimization of FCFS based Resource Provisioning Algorithm for cloud computing. *IOSR Journal of Computer Engineering (IOSR-JCE),* 10(5), 1-5.

[17]. Masdari, M., ValiKardan, S., Shahi, Z., & Azar, S. I. (2016). Towards workflow scheduling in cloud computing: A comprehensive analysis. *Journal of Network and Computer Applications*, 66, 64-82.

[18]. Mohammadi, F., Jamali, S., & Bekravi, M. (2014). Survey on job scheduling algorithms in cloud computing. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, 3(2), 151-154.

[19]. Pasupuleti, S. K., Ramalingam, S., & Buyya, R. (2016). An efficient and secure privacy-preserving approach for outsourced data of resource constrained mobile devices in cloud computing. *Journal of Network and Computer Applications*, 64, 12-22.

[20]. Rana, M., Bilgaiyan, S., & Kar, U. (2014, July). A study on load balancing in cloud computing environment using evolutionary and swarm based algorithms. In *Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on* (pp. 245-250). IEEE.

[21]. Santhosh, B., & Manjaiah, D. H. (2014). An improved task scheduling algorithm based on max-min for cloud computing. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(2), 84-88.

[22]. Supreeth, S., & Biradar, S. (2013). Scheduling virtual machines for load balancing in cloud computing platform. *International Journal of Science and Research (IJSR)*, 2(6), 437-441.

[23]. Wei, G., Vasilakos, A. V., Zheng, Y., & Xiong, N. (2010). A game-theoretic method of fair resource allocation for cloud computing services. *The Journal of Supercomputing*, 54(2), 252-269.

[24]. Zhang, J., Huang, H., & Wang, X. (2016). Resource provision algorithms in cloud computing: A survey. *Journal of Network and Computer Applications*, 64, 23-42.

[25]. Zhou, Z., & Zhigang, H. (2014). Task scheduling algorithm based on greedy strategy in cloud computing. *The Open Cybernetics & Systemics Journal*, 8(1), 111-114.

ABOUT THE AUTHORS

Jamilu Yahaya Maipan-uku is a Lecturer in the Department of Computer Science at Faculty of Natural Sciences, Ibrahim Badamasi Babangida University Lapai (IBBUL), Niger State, Nigeria. He holds M.Sc. Degree in Computer Networks from the Universiti Putra Malaysia. His research interests include; Cloud Computing, Cyber Security, Machine Learning, and Grid Computing. His publications appears in reputable Journals with Global Database (e.g Springer/Scopus).

Dr. Amit Mishra is a Senior Lecturer in the Department of Computer Science at Baze University Abuja, Nigeria. His research interests include; Data Structure & Algorithm, Machine Learning, Software Engineering, Reverse Engineering, and Software Measurement. He has been publishing in the above-mentioned fields in various National and International Journals. He is also Microsoft Certified professional in Win2K and Exchange Server.

Abdulrahman Abdulganiyu is a Lecturer in the Department of Computer Science at Faculty of Natural Sciences, Ibrahim Badamasi Babangida University Lapai (IBBUL), Niger State, Nigeria. He is currently pursuing his PhD Computer Science University of Abuja. He holds a B.Sc. in Computer Science from the University of Abuja, Abuja, Nigeria, a Master's Degree in Computer System from the University of Ibadan, Ibadan. His scholarly publication has appeared and has been quoted in reputable peer-refereed Journals, Conference Proceedings, Newsletters, and Edited Books. His research interests include; Cyber Security & Algorithm, Machine Learning, Internet of Things, Reverse Engineering, and Software Measurement.

Dr. A. Abdulkadir is a Lecturer in the Department of Mathematics at Faculty of Natural Sciences, Ibrahim Badamasi Babangida University Lapai (IBBUL), Niger State, Nigeria. His scholarly publication has appeared and has been quoted in reputable peer-refereed Journals, Conference Proceedings, Newsletters and Edited Books. His research interests include; Algorithm, Machine Learning, and Topology.