ISSN No. 0976-5697
Volume 8, No. 3, March – April 2017
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

A Comparative Analysis of Min-Min and Max-Min Algorithms based on the Makespan Parameter

Neha Sharma
M.Tech. Scholar
Department of Computer Science & Applications
Kurukshetra University, Kurukshetra, India

Dr. Sanjay Tyagi
Assistant Professor
Department of Computer Science & Applications
Kurukshetra University, Kurukshetra, India

Swati Atri
Ph.D. Scholar
Department of Computer Science & Applications
Kurukshetra University, Kurukshetra, India

Abstract: In today's computational environment, cloud computing has grown up as a utility. With increase in number of resource requests, there is a need for efficient resource allocation mechanism which aims at proper utilization of resources. Therefore in order to process these requests by making use of the available resources, an efficient task scheduling algorithm is required. An efficient task scheduling mechanism should be able to minimize completion time, maximize resource utilization and minimize makespan etc. Based on the efficiency metrics or parameters there are numerous type of existing scheduling algorithms but there is no such algorithm so far which can optimize all these parameters simultaneously, as each scheduling algorithm works on some particular parameter and tries to optimize it only. In this paper, two heuristic based algorithms: Min-Min and Max-Min have been described, simulated on CloudSim and then comparison based analysis has been done by taking makespan parameter into consideration.

Keywords: Cloud Computing, Heuristic Algorithm, Max-Min Algorithm, Metatask, Min-Min Algorithm, Task Scheduling.

I. INTRODUCTION

Cloud computing is a kind of distributed and parallel system in which various type of resources are dynamically provisioned to the user on the basis of the Service Level Agreement (SLA), established between the service provider and consumers [1]. In cloud computing, software, platform and infrastructure are provided to the customer as a service. As the number of customers increases, the task of allocating resources to each user as per their need becomes difficult. Here comes the concept of task scheduling. Task scheduling in cloud computing is the most effective and efficient concept as it enables the system to achieve the best resource utilization and maximize throughput [2]. Many task scheduling algorithms helped in better scheduling of tasks in cloud and they provide optimization in efficiency of various parameters, but there is always a chance of enhancement [3]. Task scheduling is performed on the user side. Here the datacenter broker is the entity where the scheduling is implemented. Datacenter broker can be defined as the middleman between the customer and cloud service provider. It collects all the information about the available resources, load on resources and cost of communication etc. Then as per the schedule it allocates the cloudlets (tasks) to available Virtual Machines (VMs) [4]. A cloud computing environment consists of following main entities:

1) Task / Cloudlet
2) Cloud Information Service
3) Datacenter Broker
4) Datacenter
5) Host
6) Virtual machine(VM)

Fig. 1 shows the pictorial representation of these entities and their working. A datacenter consists of many hosts. Each host can have multiple virtual Machines abbreviated as VMs. Every host in datacenter takes the virtual machine on the basis of their hardware specifications (processing power, memory and bandwidth etc.) [5]. Processing power is calculated in MIPS (Million Instruction Per Second).

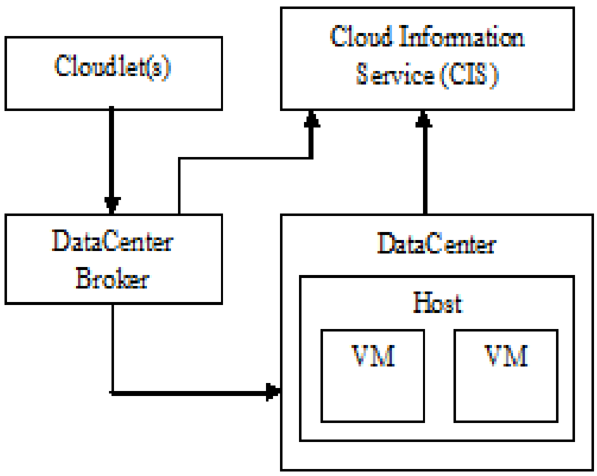

Fig. 1: Task allocation in Cloud computing.

Cloud Information Service (CIS) can be defined as a repository to store the cloud entities. Once a datacenter is created, it has to be registered on a CIS.

When a user requests a service, cloudlet is (task) send to the datacenter broker which collects the information of available resources from the CIS, and then as per the scheduling policy defined in the datacenter broker, it allocates the tasks to each Virtual Machine [6].

In task scheduling, tasks are assigned on the basis of two policies, either Time shared policy or Space shared policy [7]. In time shared policy each virtual machine has an equal amount of time i.e. tasks are performed in parallel. In space shared policy virtual machines has an equal share on the processing element of the system [4].

Rest of the paper has been organized as follows: Section II describes the general terms used in task scheduling algorithms. Section III presents two popular heuristic algorithms- Min-Min and Max-Min along with their pseudo code. In section IV, simulation of these two task scheduling algorithms have been performed on Cloudsim, then a comparative analysis based on makespan has been done and results have been presented graphically. Section V concludes the paper with future scope.

II. RELATED TERMS IN TASK SCHEDULING

This section provides the description of some basic terms used in task scheduling algorithms:

A) MetaTask (MT): It is a set of tasks, which will be mapped on available resources in cloud computing [8]. It can be shown as:

$$MT= \{t_1, t_2, t_3, ---------, t_n\}.$$

B) Makespan: It can be defined as the total time for executing a schedule (i.e. number of tasks in metatask) [9].

C) Minimum Execution Time (MET): Allocating a given task (T_i) to the best available resource (R_j) in order to minimize execution time for that particular task, without considering the machine's availability is known as MET [10].

D) Minimum Completion Time (MCT): Assigning every given task to the available machine in order to fulfill the request as fast as possible, which can give the fastest result, means it consider the availability of the resource before assigning the task. Minimum completion time can be calculated by the addition of the execution time of task t_i, known as ET_{ij} and ready time or start time of resource R_j, known as r_j [11]. It can be represented as:

$$MCT = ET_{ij} + r_j.$$

III. TASK SCHEDULING ALGORITHMS BASED ON HEURISTIC APPROACH

For finding an optimal task scheduling algorithm, various types of strategies have been adopted such as heuristic, meta-heuristic and energy based etc. Here two popularly used heuristic algorithms known as Min-Min and Max-Min have been presented:

A) Min-Min Algorithm: Min-Min scheduling is based on the concept of assigning a task having minimum completion time (MCT) first for execution on the resource, which has the minimum completion time (fastest resource). This algorithm is divided into two steps. In first step, expected completion time of each task in the metatask is calculated on each resource. In second step, the task with minimum expected completion time is selected and assigned to the corresponding resource, and then the selected task is removed from the metatask. This process repeats until all the tasks in metatask get mapped [12] [13].

Pseudo code for Min-Min algorithm:

//Phase 1: Calculation of minimum completion time of each task.

1. For all the submitted tasks (t_i) in metatask (MT)
2. For all the resources R_j
3. Compute completion time
CTij= ETij + r_j
4. End of step 2 loop.
5. End of step 1 loop.

//Phase 2: Assigning task t_i with **minimum** completion time to the resource having **minimum** completion time.

6. For each task in MT, find the task t_i with minimum completion time and that resource on which it is calculated
7. Assign t_i to resource R_j that has minimum completion time
8. Remove task t_i from MT
9. Update resource R_j ready time of r_j
10. Update completion time of all unmapped tasks in MT
11. Repeat step 6-10, until all the tasks in metatask (MT) have been mapped
12. End of step 6 loop.

Min-min algorithm is a simple algorithm but it gives the fast result when the size of task in metatask is small as compared to large size task. On the other hand, if large size tasks overlay the number of smaller tasks, it gives a poor resource utilization and large makespan because large size tasks have to wait for the completion of smaller tasks.

B) Max-Min Algorithm: This algorithm overcomes the drawback of the min-min algorithm (larger makespan when numbers of large size tasks are greater than the small size task) [14]. Max-Min algorithm performs the same steps as the Min-Min algorithm but the main difference comes in the second phase, where a task t_i is selected which has the maximum completion time instead of minimum completion time as in min-min and assigned to resource R_j, which gives the minimum completion time. Hence it named as the Max-Min algorithm. This process is repeated until the metatask get empty or all the tasks are mapped [15] [16].

Pseudo code for Max-Min algorithm:

//Phase 1: Calculation of minimum completion time of each task.

1. For all the submitted tasks (t_i) in metatask (MT)
2. For all the resources R_j
3. Compute completion time
   CTij= ETij + r_j
4. End of step 2 loop.
5. End of step 1 loop.

//Phase 2: Assigning task t_i with **maximum** completion time to the resource which gives **minimum** completion time.

6. For each task in MT, find the task t_i with maximum completion time and that resource on which it is calculated
7. Assign t_i to resource R_j that has minimum completion time
8. Remove task ti from MT.
9. Update resource R_j ready time (r_j)
10. Update completion time of all unmapped tasks in MT
11. Repeat step 6-10, until all the tasks in metatask (MT) have been mapped
12. End of step 6 loop.

The main aim of max-min scheduling algorithm is to reduce the waiting time of large size jobs [17]. In this algorithm, small size tasks are concurrently executed with large size tasks, hence reducing the makespan and providing better resource utilization.

IV. SIMULATION BASED COMPARATIVE ANALYSIS ON CLOUDSIM

Simulation of two heuristic algorithms- Min-Min and Max-Min has been done by using Cloudsim as a simulation tool. Cloudsim has some effective characteristics over other simulation tools such as:

a) Isolation of the multi-layer service abstractions [1]. Multi-layer includes IaaS, PaaS, SaaS etc.
b) Virtualized environment is easy to create.
c) Perform simulation such as working on an actual cloud and there is no limit on number of entities [1].

Data used for simulation purpose is given in Table 1.

Table-1: Numbers of entities

Name of Entity	Total Number of Entities Used
Host	1
Virtual Machines (VMs)	2
Tasks (Cloudlets)	50

Table-2 shows configuration of host, which includes the storage and bandwidth etc.

Table-2: Host Configuration

RAM	2048MB
Processing power	220000 (in MIPS)
Bandwidth	10000 (in Mbps)

Table-3 gives the configuration details of virtual machines in a datacenter.

Table-3: VMs Specifications

Resources	Processing speed (in MIPS)	No. of CPUs or cores
VM1	250	1
VM2	500	1

After implementing the Min-Min and Max-Min algorithms with 50 cloudlets and 2 virtual machines their makespan has been calculated. Value of makespan for both algorithms has been calculated after running the code 20 times and then their mean is calculated, which is shown below, as a final value. Final results have been shown with the help of bar graph in Fig 2.

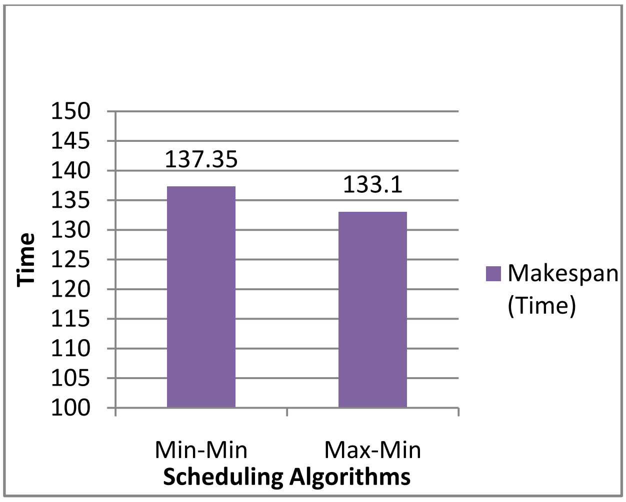

Fig. 2: Comparison of Min-Min and Max-Min on the basis of Makespan in (ms).

The above bar graph shows that the Max-Min gives the better makespan as compared to the Min-Min algorithm.

V. CONCLUSION AND FUTURE SCOPE

Task scheduling is among one of the main concern in cloud computing. Finding an efficient task scheduling algorithm has always been the prominent field for research. In this paper, simulation based comparative analysis on two most popularly used heuristic task scheduling algorithms named as Min-Min and Max-Min has been performed. It seems that the Max-Min outperforms the Min-Min when number of large sized tasks is more than the short length task. But when short length tasks outnumber the long length task, Min-Min can be better choice. In future, there is a wide scope of improvement where an algorithm can be developed which can perform better and give efficient way of scheduling task by utilizing the advantages of above mentioned algorithms and simultaneously overcoming the disadvantages of them.

VI. REFERENCES

[1] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, C´esar 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, John Wiley & Sons, pp. 23-50, June 2011.

[2] S.Devipriya and C.Ramesh, "Improved Max-Min Heuristic Model For Task Scheduling In Cloud," in International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), IEEE, 2013, pp. 883-885.

[3] Xiaonian Wu, Mengqing Deng, Runlian Zhang, Bing Zeng, and Shengyuan Zhou, "Task Scheduling Algorithm Based on QoS-driven in Cloud Computing," Elsevier publications, pp. 1162-1169, 2013.

[4] R.Jemina Priyadarsini and L.Arockiam, "Performance Evaluation of Min-Min and Max-Min Algorithms for Job Scheduling in Federated Cloud," International Journal of Computer Applications, vol. 99, no. 18, pp. 47-54, August 2014.

[5] Gaurang Patel, Rutvik Mehta, and Upendra Bhoi, "Enhanced Load Balanced Min-Min algorithm for Static Meta Task Scheduling in Cloud Computing," Elsevier, pp. 545-553, 2015.

[6] Navdeep Kaur and Khushdeep Kaur, "Improved Max-Min Scheduling Algorithm," IOSR Journal of Computer Engineering (IOSR-JCE), vol. 17, no. 3, pp. 42-49, May-June 2015.

[7] Sanjay Kumar and Atul Mishra, "Application of Min-Min and Max-Min Algorithm for Task Scheduling in Cloud Environment Under Time Shared and Space Shared VM Models," International Journal of Computing Academic Research (IJCAR), vol. 4, no. 6, pp. 182-190, December 2015.

[8] Christoph Buchheim and Jannis Kurtz, "Min-max-min robustness: a new approach to combinatorial optimization under uncertainty based on multiple solutions," Elsevier, pp. 45-52, 2016.

[9] Abdul Razaque, Nikhileshwara Reddy Vennapusa, Nisargkumar Soni, Guna Sree Janapati, and Khilesh Reddy Vangala, "Task Scheduling in Cloud Computing," in Long Island Systems, Applications and Technology Conference (LISAT), IEEE, 2016.

[10] D.I. George Amalarethinam and S. Kavitha, "Enhanced Min-Min Algorithm for Meta-task Scheduling in Cloud Computing," IJCTA , pp. 85-91, 2016.

[11] Huankai Chen, Frank Wang, Na Helian, and Gbola Akanmu, "User-Priority Guided Min-Min Scheduling Algorithm For Load Balancing in Cloud Computing," IEEE, 2013.

[12] Kobra Etminani and M. Naghibzadeh, "A Min-Min Max-Min Selective Algorihtm for Grid Task Scheduling," in ICI 2007. 3rd IEEE/IFIP International Conference in Central Asia , 2007.

[13] Soheil Anousha and Mahmoud Ahmadi, "An Improved Min-Min Task Scheduling Algorithm in Grid Computing," Springer-Verlag Berlin Heidelberg, pp. 103-113, 2013.

[14] Upendra Bhoi and Purvi N. Ramanuj, "Enhanced Max-min Task Scheduling Algorithm in Cloud Computing," International Journal Of Application or Innovation in Engineering and Management(IJAIEM), vol. 2, no. 4, pp. 259-264, April 2013.

[15] Gao Ming and Hao Li, "An Improved Algorithm Based on Max-Min for Cloud Task Scheduling," Recent Advances in Computer Science and Information Engineering in Springer Berlin Heidelberg, vol. Volume 125 of the series Lecture Notes in Electrical Engineering, pp. 217-223, January 2012.

[16] Gaurav Sharma and Puneet Banga, "Task Aware Switcher Scheduling for Batch Mode Mapping in Computational Grid Environment," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 6, pp. 1292-1299, june 2013.

[17] Bhavisha Kanani and Bhumi Maniyar, "Review on Max-Min Task scheduling Algorithm for Cloud Computing," JETIR (ISSN-2349-5162), vol. 2, no. 3, pp. 781-784, March 2015.