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 assigns 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 have 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:

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

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:

$$\text{MCT} = \text{ET}_{ij} + r_j.$$  

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
   $$\text{CT}_{ij} = \text{ET}_{ij} + r_j$$
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

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
   \[ CT_{ij} = ET_{ij} + r_j \]
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 \(t_i\) 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

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>
<thead>
<tr>
<th>Name of Entity</th>
<th>Total Number of Entities Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>1</td>
</tr>
<tr>
<td>Virtual Machines (VMs)</td>
<td>2</td>
</tr>
<tr>
<td>Tasks (Cloudlets)</td>
<td>50</td>
</tr>
</tbody>
</table>

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


