A priority based on min-min algorithm for reducing make span task scheduling in cloud computing

Abdulrahman Mohammed Hussein Obaid 1 *, Santosh Kumar Pani 1, Prasant Kumar Pattnaik 1

1 School of Computer Science and Engineering, KIIT University, Bhubaneswar, Odisha, India
*Corresponding author E-mail: obaid.eng@gmail.com

Abstract

This paper proposes a two-phase technique for task scheduling which works on third-party broker. The priority algorithm is executed by selecting the task that has the highest priority. However, if more than one task has the same priority; it goes to second phase to execute the traditional Min-Min algorithm. Experiments are conducted by considering random tasks in order to compare the performance of the proposed algorithm with the Min-Min algorithm. The recorded experimental outcomes indicate that the proposed technique is given 10% better results as compared to the traditional Min-Min algorithm.

Keywords: Cloud Computing; Min-Min Algorithm; User-Priority; Cloud Task Scheduling; Cloudsim.

1. Introduction

Cloud computing is a virtual shared pool of computing resources such as CPU, networks, memory, and storage. It is accessible to the users through internet. The services provided by service provider makes the virtual shared pool useful. The service provider offers three levels of services, Software (SaaS), Platform (PaaS), and Infrastructure (IaaS). These services work under (pay per use) [1, 5]. Availability, Performance and Bandwidth Cost are some of the important issues in cloud computing environment. To deal with the high cost and to make all resources available for all users, the way of sending and receiving data between users and cloud should be carefully selected. The scheduler is an operation that selects the next jobs to be submitted to the system and the next process to run. In cloud computing, task scheduling is performed on the user side and is done by datacenter broker (DC). Datacenter broker may be in between the customer and cloud service provider [10, 12].

Many researchers proposed models for load balancing and scheduling algorithm in cloud computing. Some common scheduling algorithms are: First Come First Service algorithm (FCFS), Max-Min algorithm, Min-Min algorithm, Round Robin algorithm etc [2, 11]. In order to handle some shortcomings of these algorithms, many improved algorithms have been proposed in cloud computing. The Min-Min algorithm is still the basis of present cloud scheduling algorithm [13]. The Min-Min algorithm has shortcomings that make it unsuitable for the users who pay for good services. Some of the shortcomings discussed in this algorithm are taken and analyzed to give a more efficient algorithm for tasks scheduling.

The paper is organized as follows: In the next section, we discuss the related works. Next section presents the problem statement that motivated us to propose the priority based Min-Min algorithm. The proposed Architecture and PBMM algorithm (Priority Based on Min-Min Algorithm) is presented in the next section. Next section includes Performance Evaluation by using cloudsim platform to demonstrate the degree of efficiency for the proposed algorithm in comparison to Min-Min algorithm. Next section concludes the paper.

2. Related works

Many algorithms have been investigated by several researchers for scheduling tasks in cloud computing. Some of those algorithms are effective and simple to give a better performance with less Makespan. The most used algorithm in cloud computing is named Min-Min algorithm. “The Min-Min algorithm is simple and still basis of present cloud scheduling algorithm” [13] Braun, T. D., et al, [3] proposed a comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. In this paper, the authors compared the eleven heuristics for scheduling independent tasks. One of them describes the Min-Min as heuristic which starts with U set of all unscheduled tasks and by having M that symbolizes a set of minimum execution times. The task that has the minimum execution time is selected and executed by the corresponding machine (cloud computing). The newly scheduled task is removed from U. same steps are repeated until all tasks are scheduled. Blythe, J., et al [2] presented Task Scheduling Strategies for Workflow-based Applications in Grids with the Min-Min algorithm being one of the strategies. They assume that there is a set of tasks and each task T (j, r) has two variables: resource j and job r. For every T (j, r) defined in terms of Execution Time ET (j, r), Completion Time CT (j, r) and Ready time RJ as shown below.

\[ CT (j, r) = ET (j, r) + RJ \] (1)

A task is scheduled based on the minimum CT. Figure 1 illustrates the Min-Min Scheduling Algorithm.
Algorithm 1 The Min-Min Scheduling Algorithm

1: for all submitted tasks in the set; T_i do
2: for all resources; R_j do
3: \( C(t) = E(t) + r(t); \)
4: end for
5: end for
6: Do while tasks set is not empty
7: Find task \( T_k \) that cost minimum execution time
8: Assign \( T_k \) to the resource \( R_j \) which give minimum expected complete time
9: Remove \( T_k \) from the tasks set
10: Update ready time \( r(t) \) for select \( R_j \)
11: Update \( C(t) \) for all \( T_i \)
12: End Do

Fig. 1: The Min-Min Scheduling Algorithm.

Kokilavani, T. and D. G. Amalarethinam, [6] proposed load balanced min-min algorithm for static meta-task scheduling in grid computing. The proposed algorithm named LBMM is performed in two phases. The Min-Min algorithm phase is executed first. The second phase reassigns the weighty-load resources into light-load resources. This ensures that the tasks which were waiting for resources with heavy load for getting execution will go to the resources that have light load. The experiments’ results obtained by LBMM algorithm show that it performs better than the Min-Min scheduling algorithm.

Parsa, S. and R. Entezari-Maleki [7] introduced a new algorithm named RASA, by using the Min-Min algorithm and Max- Min algorithm. Min-Min algorithm first identifies the task with the minimum execution time among all and then assign the task on that resource which provides minimum completion time for the task. The Max-Min algorithm works opposite to Min-Min algorithm. First, it identifies the task with the largest execution time among all the tasks. In RASA algorithm, first the scheduler allocates the resources according to the number of the available resources. Then it chooses Min-Min algorithm to allocate the resources by taking the smaller tasks. This operation is repeated till all the tasks are scheduled. The resulted simulation shows that the RASA algorithm outperforms the Min-Min algorithm and Max-Min algorithm.

Patel, S. J. and U. R. Bhoi [8] introduced an Improved Priority Based Job Scheduling Algorithm in Cloud Computing Using Iterative Method. The algorithm uses three levels of priorities. The first level is scheduling, the second level is resources, and the third level is jobs. Scheduling level is the goal that done by the scheduler, resources level is the features that are available to achieve the required goal, and job level decides the best job that should be scheduled first. The proposed algorithm makes comparison matrices for resources and determines which resource has the highest priority. It is used to find resources and priority of jobs to achieve better performance and minimizing makespan.

3. Problem statement

In cloud computing, the service provider gives the customer different levels of services, ‘Pay-as-you-use’. It means that the customer can select different packages such as Storage Space, CPU and RAM, etc. They don’t get different quality of tasks execution, especially data processing and moving from user to cloud storage. For example, by having two customers, one using a free account and the second is using a paid package. There are different hardware equipments between the two customers but the data that are sent from both of them run and process on the same level of priority.

The main drawback in Min-Min algorithm is that it selects the smallest task first, that makes all resources available for all tasks equally, without giving any priority to tasks that have high priority [4-9]. As a result, the schedule produced by Min-Min is not optimal when tasks have a different priority and it doesn’t work for Multi-level priority. The main shortcomings of Min-Min algorithm are presented below.

1) The makespan is too long for high priority user.
2) Sometimes resource utilization is not available for high priority user.
3) High priority tasks may wait for a long time to run.
4) It does not support the priority levels.

4. Proposed PBMM Model

The proposed model tries to minimize the waiting time for customers who pay money to get a good service. For scheduling data, a third-party broker applies the following techniques:

In third-party broker, the model operates in two phases:

1) First it uses priority algorithm to determine which task has high priority.
2) Second it uses the traditional Min-Min algorithm to decide which task to execute first in case there is more than one task of the same priority.

The proposed PBMM (Priority Based on Min-Min) scheduling algorithm is presented in Fig 3.

Algorithm 2 The PBMM Scheduling Algorithm

1: for all submitted tasks in the set; T_i do
2: for all resources; R_j do
3: \( C(t) = E(t) + r(t); \)
4: end for
5: end for
6: Do while tasks set is not empty
7: Find task \( T_k \) that give minimum execution time
8: Assign task to resource \( R_j \) that gives maximum priority and minimum expected completion time
9: Remove task \( T_k \) from the set
10: Update ready time \( r(t) \) for selected resources \( j \)
11: else
12: Assign task to resource \( R_j \)
13: end if
14: Delete task \( T_k \) from the set
15: Update \( C(t) \) for all \( T_i \)
16: End Do

Fig. 3: The PBMM Scheduling Algorithm.

This algorithm starts by scanning all tasks to find the one with highest priority. In case, there is more than one task having same priority, the Min-Min algorithm calculates the completion time for all tasks. Finally, the algorithm selects the task having minimum execution time. The process is repeated until all tasks assigned to the resource. The Pseudocode for the algorithm is shown in figure 4.
5. Performance evaluations

The proposed PBMM scheduling algorithms uses both priority algorithm and Min-Min algorithm. In this section, we compare the traditional Min-Min algorithm and PBMM algorithm using cloudsim platform.

5.1. Simulation setup

Based on the properties and attributes of cloud computing and by changing some parameters and adding new ones in the cloudsim platform, the following simulation environment was set.

1) Virtual Machines (VM), for each VM requires one CPU with 1000 MIPS, 512 MB of RAM, 100 MB of storage, 1000 Kbits/s Bandwidth and Xen for VMM. In this experiment simulation used only one VM to achieve our proposal algorithm.

2) Each host has one CPU, and the CPU performance is 1000 MIPS, 2048 MB of RAM, 100000 MB of storage, and 1000 Kbits/s Bandwidth.

3) Cloudlets (Tasks): Tasks generated dynamically and randomly. In this experiment by choosing 7 tasks for simulation and each task is modeled to have 40000 lengths of instructions, 300 kb input filesize, 400kb output filesize.

4) In addition, 0.11 s for running time for each task.

5) Each task belongs to pack and every pack has a priority value is given by random number between 0 and 3. The first Pack is free pack, it has the lowest priority 0. The second pack is silver pack, it has priority 1. The third pack is gold pack, it has priority 2. And finally, the fourth pack is platinum pack, it has the highest priority 3, as shown in table 1.

Table 1: Priority Given to Tasks

<table>
<thead>
<tr>
<th>Packs</th>
<th>Free</th>
<th>Silver</th>
<th>Gold</th>
<th>Platinum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority given</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

5.2. Performance impact of Min-Min algorithm

The scheduler schedules the tasks that are begotten by using Min-Min algorithm. As mentioned in the previous section, the Min-Min algorithm works by first finding the task that has minimum execution time over all tasks, the task that has the minimum time will run first, then the task with second minimum time and so on. This process will be repeated till all the tasks are comp. To illustrate the working of Min-Min algorithm, we simulated seven tasks (Task0, Task1, Task2, Task3, Task4, Task5, and Task6) which are submitted by different users using Min-Min algorithm as shown in Figure 5.

Fig. 5: Tasks and Execution Time (Example).

By using the Min-Min algorithm, the task that has the minimum execution time is found first. Task 5 will be executed first, followed by the second task that has the second minimum execution time (task3). The procedure goes on for the remaining tasks until all the tasks are scheduled. The performance of the seven tasks in terms of Min-Min algorithm is shown in Figure 6 and figure 7.

Fig. 6: Execution Tasks by Using Min-Min Algorithm.

5.3. Performance impact of PBMM algorithm

This section demonstrates the effectiveness of the proposed PBMM algorithm by using same tasks. It considers the priority
assigned to the tasks submitted by different users (Task0, Task1, ..., Task6). The priority of every task is shown in Table 1. The tasks are scheduled by using PBMM algorithm. Figure 8 depicts tasks execution.

<table>
<thead>
<tr>
<th>Cloudlet ID</th>
<th>Task ID</th>
<th>Task Priority</th>
<th>Task Size</th>
<th>Running Time</th>
<th>Start Time</th>
<th>Final Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>0.11</td>
<td>0.1</td>
<td>0.22</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>0.11</td>
<td>0.21</td>
<td>0.31</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>0.11</td>
<td>0.32</td>
<td>0.43</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>0.11</td>
<td>0.48</td>
<td>0.54</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0.11</td>
<td>0.54</td>
<td>0.65</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>0.11</td>
<td>0.65</td>
<td>0.76</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0.11</td>
<td>0.76</td>
<td>0.87</td>
</tr>
</tbody>
</table>

Figure 8: Depicts Tasks Allocation to Resource According to PBMM Algorithm.

The PBMM algorithm first schedules the tasks based on the priority. Since task0 and task3 have the highest priority, the traditional Min-Min algorithm is used to decide the task having minimum execution time. Hence, task3 will run first followed by task0. The same procedure is repeated until all the tasks are scheduled. The graphical representation showing the improvement of PBMM Algorithm is shown in Figure 9.

Figure 9: Graphical Representation to Show Improvement of PBMM Algorithm.

5.4. Difference performance between min-min algorithm and PBMM algorithm

The difference in execution between Min-Min algorithm and PBMM algorithm is shown in Figure 10.

Figure 11 presents the difference between using traditional Min-Min algorithm and using PBMM algorithm. When Min-Min algorithm executes task3, it takes 0.32s to finish execution, but when PBMM executes task3, it takes 0.21s. Because task3 has the highest priority value (platinum priority = 3) as shown in Table 1.

Figure 11: difference Performance of PBMM and Min-Min for Task3.

Similarly, task1 having gold priority (priority= 2), the traditional Min-Min algorithm takes 0.65s whereas the proposed PBMM algorithm reduces it to 0.43s. Execution of task1 is shown below in Figure 12.

Figure 12: Performance of PBMM and Min-Min for Task1.

Task5 has free priority (priority = 0), the traditional min-min algorithm takes 0.21s for execution. However, using PBMM algorithm, it will take 0.87s. To run task5 faster, the user needs to pay for good service to be given high priority. Figure 13 below illustrates task5 execution.

Figure 13: difference Performance of Min-Min and PBMM for Task5.

6. Conclusion

In this paper, we propose an algorithm named PBMM which operates in two phases. All tasks are rescheduled by PBMM algorithm to use the unutilized resources effectively and provide a good...
service for users who have high priority. The analysis and evaluation of the performance of our new algorithms were done through a simulation tool named cloudsim platform. Detailed scheduling and experiment results shows that the PBMM algorithm can reduce the makespan, enhance efficiency, and make it faster than the traditional Min-Min algorithm for users who have high priority.

References