Computing, Information Systems, Development Informatics & Allied Research Journal
Vol. 11 No. 1, 2020 - www.cisdijournal.net

Comparative Analysis of Scheduling Algorithms in Cloud Computing Environment

[1]Ayorinde I. T. & [2]Daniel A. A.
[1,2]Department of Computer Science
University of Ibadan
Ibadan, Nigeria.
E-mail: [1]temiayorinde@yahoo.com ; [2]danayo1988@gmail.com
Phone: [1]+2348035289814; [2]+2348168139099

ABSTRACT

The ability of a scheduler to utilize resources maximally, manage computing performance and increase the number of tasks that are completed successfully determines its efficiency. As cloud computing enables the storage and accessing of data and programs over the internet and several millions of people access it, it becomes pertinent to make use of highly efficient scheduling algorithms that will give a fair distribution of the various tasks to the different virtual machines available and get the jobs done on time. Credit based, Min-Min and Priority based scheduling algorithms have all been used in cloud computing. This study improves the credit based algorithm by ordering the length and priority of tasks in a descending order and also used virtual machines of different processing speed and power. The extended credit based, min-min and priority based scheduling algorithms were simulated using the same environment and their results compared based on their makespan, execution time and throughput values. Cloudsim 3.0.3 simulator on eclipse Integrated Development Environment (IDE) was used for the simulation. The makespan, execution time and throughput values for the extended credit based are 94 seconds, 358.5 seconds and 0.0558 bits per second (bps). Min- Min algorithm has 115 seconds, 364.04 seconds and 0.0549 bps while Priority based scheduling algorithm has 110 seconds, 359.1 seconds and 0.0557 bps respectively. The result shows that the extended credit based scheduling algorithm performs better than the other two algorithms. Hence, this new model reduces the execution time of user's tasks and increases the throughput of the cloud computing system. It is hereby recommended for task scheduling in cloud computing.

Keywords: Extended credit based algorithm, Cloud computing, Cloudsim 3.0.3

CISDI Journal Reference Format
Ayorinde I. T. & Daniel A. A. (2020): Comparative Analysis of Scheduling Algorithms in Cloud Computing Environment. Computing, Information Systems & Development Informatics Journal. Vol 11 No 1, Pp 53-66
Available online at www.computing-infosystemsjournal.info

1. INTRODUCTION

Cloud computing is referred to as a parallel and distributed computing system consisting of a pool of interconnected and virtualized computers. They dynamically allocate computing resources to the users based on pre-agreed service level agreements (Odun-Ayo et al., 2017; Buyya et al., 2009).

Cloud computing is also described as a means of storing and accessing data and programs over the internet instead of individual's hard drive (Griffith, 2016). It is easily accessible, cost effective, flexible and secured (OpenCirus, 2018). Effective task scheduling, coupled with quality of service is crucial in cloud computing. The aim is usually to yield minimum execution time and increase throughput. Otherwise, there could be inappropriate allocation of resources and users tasks will not be executed on time thereby leading to frustration on the part of the users and loss of customers on the side of cloud providers. Hence, the need for an efficient task scheduling algorithm.

The scheduling algorithms map users' tasks to suitable resources in the form of virtual machines (VMs) that execute them (Ahmed, 2017). Some algorithms used in the literature are Min-Min, Priority based and Credit based algorithms. While Min-Min algorithm uses task length only for scheduling, priority based uses tasks priority for scheduling and credit based uses both tasks length and tasks priority (Thomas, Krishnalal and Raj, 2015; Kokilavani and Di, 2011).

In cloud computing environment, datacenters receive tasks from the datacenter brokers which arrived from different users. In some cases, these tasks may be associated with priorities such that, the broker needs to consider these priorities for the purpose of scheduling and is also responsible for assigning the tasks (Thomas, Krishnalal and Raj, 2015).

1.1 Credit Based Scheduling

According to Bichkule and Mante (2017) and Thomas, Krishnalal and Raj (2015), the credit of each task is based on both task length and task priority and they are calculated as follows:

Step 1: Find task length.

i. Find the length of each task
ii. Then calculate average of task length. (If there are 5 tasks, then divide the sum of 5 tasks by 5).
iii. Calculate the difference between task length and average of task length.
iv. After finding the difference between task length and average of task length, assign the output value to that particular task.
v. Repeat steps iv and v for the amount of tasks you have.

Step 2: Find task priority credit.

i. Each task may have different priority.
ii. Choose division factor to calculate Priority_factor (if the value of priority is between 0 to 99, then division factor will be 10, if value of priority is between 100 to 999, then division factor will be 100) and so on .
iii. In the same way, calculate Priority_factor for each task.

Step 3: Find total credit.

i. Total credit =Task length * Priority_factor. (Calculate this for each task.)
ii. Sort the tasks based on their total credit.
iii. The task which has highest credit will execute first and so on.

In this way, credit based scheduling is mainly used to schedule various tasks on clouds.

1.2 Min-Min Scheduling Algorithm
The algorithm for Min-Min scheduling is shown below:

1. **For all submitted tasks in the set; T_i**
2. **For all resources; R_j**
3. $Ct_{ij}=Et_{ij}+rt_j$; End **For;End For;**
4. **Do while tasks set is not empty**
5. Find task T_k that cost minimum execution time
6. Assign T_k to the resource R_j while gives minimum Expected complete time
7. Remove T_k from the task set
8. Update ready time rtj for select R_j
9. Update Ct_{ij} for all T_i
10. End **Do**

Figure 1: Min-Min Algorithm (Thomas, Krishnalal and Raj, 2015)

Let the tasks set be T_1,T_2,T_3,...etc and resources set be R_1,R_2,R_3,..etc. Expected completion time of task i on resource j is denoted as Ct_{ij}. It is calculated by using the equation (1).

$Ct_{ij} = Et_{ij} + rt_j$ …………………………Equation (1)

rt_j represents the ready time of resource R_j. Et_{ij} stands for execution time of task i.

The major drawback with this algorithm is that it considers only task length and not user's priority. It is possible to have a situation where high priority tasks exist and hence, priority needs to be taken into consideration. In such a case, this algorithm may not be very effective for scheduling.

1.3 Priority Based Scheduling Algorithm
The algorithm for a priority based scheduling is shown in Figure 2. This kind of scheduling is based on users Priority. The priority is calculated based on the cost users associate with tasks. This method usually gets good results when shortest job is executed first but the major challenge occurs when a task with larger length is associated with the highest priority.

1) Enter $J=\{j_1,\ldots,J_m\}$ a set of jobs
2) Enter $C=\{c_1,\ldots,C_d\}$ a set of resource
3) For all jobs make consistent comparison matrix according to priority of resources accessibilities(d matrix with *m* row and *m* column)
4) Compute priority vector for all d matrixes in step 3 based on Eq. (3).
5) Make a matrix with priority vectors in step 4 based on Eq. (8) and name it Δ .
6) For C compute a consistent comparison matrix according to decision maker(s).
7) Compute priority vector for matrix in step 6 same as step 4 and name it γ .
8) Compute *PVS* which is a vector included value of priority of jobs from Eq. (9).
9) Choose a job with maximum amount of priority value based on *PVS* and allocate it suitable resource.
10) Update the list of jobs.
11) End.

Figure 2: A General Algorithm for Priority Based Scheduling (Ghanbari and Othman, 2012).

2. RELATED WORKS

Scheduling algorithms are expected to achieve full utilization of resources in cloud computing. This is why so many of such algorithms exist, some of which are explained below:

2.1 Dynamic Cloud Task Scheduling

To maximize task scheduling performance and minimize non reasonable task allocation in clouds, this paper proposes a method based on a two-stage strategy where at the first stage, a job classifier motivated by a Bayes classifier's design principle is utilized to classify tasks based on historical scheduling data and in the second stage, tasks are matched with concrete VMs dynamically which improved the cloud's scheduling performance and achieve the load balancing of cloud resources in comparison with existing methods (Zhang and Zhou, 2017).

2.2 Non Dynamic Cloud Task Scheduling

Here, the research work minimizes the execution cost of task using the Amazon EC2 pricing model under deadline constraint (Ahmed, 2017). The algorithm does not support dynamic task scheduling.

2.3 Workload Based Scheduling

The current research can provide a general guidance as to how to schedule multiple tasks with different workload in clouds, thereby achieving optimized system performance under different circumstances. This research did not consider the continuous arrival of tasks at different times (Liu et al., 2017).

2.4 Cost Based Scheduling

The available resources in the cloud environment and its usage have cost implication for the users in execution of tasks. Each cloud resource is associated with a cost. A dynamic cost-effective deadline-constrained heuristic algorithm for scheduling a scientific workflow in a public cloud which exploit the advantages offered by cloud computing while taking into consideration, the virtual machine performance variability and instance acquisition delay to identify a just-in-time schedule of a deadline constrained scientific workflow at lesser costs was carried out (Sahni and Vidyarthi, 2015).

The scheduling algorithm addresses three major issues of cloud environment which includes: virtual machine performance variation, resource acquisition delays and heterogeneous nature of cloud resources. Scheduling based on the resource cost has limitations especially on how to appropriately schedule tasks to resource(s) that will not cost much as a means to minimize cost.

3. METHODOLOGY

The three major algorithms to be compared are given in sections 1.1, 1.2 and 1.3. A little adjustment was made to the credit based scheduling algorithm and that is why it is referred to as extended credit based scheduling algorithm in this study. The major differences are shown in Table 1 below:

Table 1: Differences between Credit Based and Extended Credit Based Scheduling Algorithms

	Extended Credit Based Scheduling	Credit Based Scheduling
1	The virtual machines used are of different processing speed and power	The virtual machines used are of the same processing speed and power
2	Tasks with higher total credits are assigned to the best virtual machines	This is not necessary as all virtual machines are the same.
3.	Sorting was done in descending order	Sorting was done in ascending order

Table 2 describes the configuration of the simulation environment. One datacenter, one host, one broker and twenty cloudlets were used.

Table 2: Configuration of the Simulation Environment

Number of Datacenter	1
Number of Host	1
Number of Cloudlets	20
Number of Broker	1

The datacenter houses the host machine that services the virtual machines which are also located in the datacenter. The twenty cloudlets are also referred to as tasks while the broker is responsible for mapping tasks to virtual machines in the datacenter. Five virtual machines were used.

The twenty cloudlets (tasks) with different lengths used are:
100000,
120000,
140000,
160000,
180000,
200000,
220000,
240000,
260000,
280000,
300000,
320000,
340000,
360000,
380000,
400000,
190000,
310000,
270000 and
280000.

The makespan, execution time and throughput were used to compare the results of the algorithms.

4. IMPLEMENTATION AND DISCUSSION OF RESULTS

The implementation was done using Cloudsim 3.0.3 on an HP Elite book with Intel (R) core (TM) 2 Duo CPU P8600 at 2.4gHz, 2GB RAM and 32-bit Windows 7 Operating System.

4.1 Result of Extended Credit Based Scheduling Algorithm

Figure 3 shows the mapping of the cloudlets with the virtual machines. The mapping is based on the total credit value of each cloudlet, which is generated from the product of the task length credit and the priority factor value. Virtual Machines are sorted in descending order based on million instructions per second (MIPS) values and cloudlets with the highest total credit value(s) are scheduled first with best virtual machines.

```
Starting Extended Credit Based Scheduling Algorithm ...
Initialising...
1. Vm ID: 3, Vm Mips:20000.0
2. Vm ID: 1, Vm Mips:17000.0
3. Vm ID: 0, Vm Mips:15000.0
4. Vm ID: 2, Vm Mips:13000.0
5. Vm ID: 4, Vm Mips:10000.0
1. Cloudlet ID: 7, Cloudlet Length:240000, Cloudlet credit:3, Cloudlet qos:3, Cloudlet totalcredit:9
2. Cloudlet ID: 6, Cloudlet Length:220000, Cloudlet credit:3, Cloudlet qos:3, Cloudlet totalcredit:9
3. Cloudlet ID: 5, Cloudlet Length:200000, Cloudlet credit:3, Cloudlet qos:3, Cloudlet totalcredit:9
4. Cloudlet ID: 16, Cloudlet Length:190000, Cloudlet credit:3, Cloudlet qos:3, Cloudlet totalcredit:9
5. Cloudlet ID: 4, Cloudlet Length:180000, Cloudlet credit:3, Cloudlet qos:3, Cloudlet totalcredit:9
6. Cloudlet ID: 3, Cloudlet Length:160000, Cloudlet credit:3, Cloudlet qos:3, Cloudlet totalcredit:9
7. Cloudlet ID: 12, Cloudlet Length:340000, Cloudlet credit:2, Cloudlet qos:4, Cloudlet totalcredit:8
8. Cloudlet ID: 11, Cloudlet Length:320000, Cloudlet credit:2, Cloudlet qos:4, Cloudlet totalcredit:8
9. Cloudlet ID: 17, Cloudlet Length:310000, Cloudlet credit:2, Cloudlet qos:4, Cloudlet totalcredit:8
10. Cloudlet ID: 10, Cloudlet Length:300000, Cloudlet credit:2, Cloudlet qos:4, Cloudlet totalcredit:8
11. Cloudlet ID: 9, Cloudlet Length:280000, Cloudlet credit:2, Cloudlet qos:4, Cloudlet totalcredit:8
12. Cloudlet ID: 19, Cloudlet Length:280000, Cloudlet credit:2, Cloudlet qos:4, Cloudlet totalcredit:8
13. Cloudlet ID: 18, Cloudlet Length:270000, Cloudlet credit:2, Cloudlet qos:4, Cloudlet totalcredit:8
14. Cloudlet ID: 8, Cloudlet Length:260000, Cloudlet credit:2, Cloudlet qos:4, Cloudlet totalcredit:8
15. Cloudlet ID: 0, Cloudlet Length:100000, Cloudlet credit:5, Cloudlet qos:1, Cloudlet totalcredit:5
16. Cloudlet ID: 2, Cloudlet Length:140000, Cloudlet credit:4, Cloudlet qos:1, Cloudlet totalcredit:4
17. Cloudlet ID: 1, Cloudlet Length:120000, Cloudlet credit:4, Cloudlet qos:1, Cloudlet totalcredit:4
18. Cloudlet ID: 15, Cloudlet Length:400000, Cloudlet credit:1, Cloudlet qos:1, Cloudlet totalcredit:1
19. Cloudlet ID: 14, Cloudlet Length:380000, Cloudlet credit:1, Cloudlet qos:1, Cloudlet totalcredit:1
20. Cloudlet ID: 13, Cloudlet Length:360000, Cloudlet credit:1, Cloudlet qos:1, Cloudlet totalcredit:1
Starting CloudSim version 3.0
Datacenter_1 is starting...
Broker_1 is starting...
Entities started.
```

Figure 3: Mapping of Cloudlets with Virtual Machines for Credit Based Algorithm

Figure 4 shows the output of the extended credit based scheduling algorithm. It gives a detailed execution time of each cloudlet. The values of its makespan, total execution time and throughput were calculated as follows:

Makespan (MS) = Finish Time of batch of tasks – Start time of tasks

From Figure 4:

Start time = 0.1 while Finish time = 94.1. Hence,

Makespan = 94.1 – 0.1 = 94seconds.

Total Execution Time (TET) = Sum of execution time of each cloudlet

Total Execution Time = 12 + 12.94 + 13.33 + 14.62 + 18 + 8 + 20 + 14 + 21.33 + 23.85 + 7 + 30 + 16.47 + 18 + 7.06 +10 + 20 + 26.67 + 29.23 + 36 = 358.5 seconds

Throughput= Number of task/Total Execution Time

Throughput = 20/358.5=0.0558 bps

```
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_1 is shutting down...
Broker_1 is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
    7        SUCCESS        2            3        12        0.1        12.1
    6        SUCCESS        2            1        12.94     0.1        13.04
    5        SUCCESS        2            0        13.33     0.1        13.43
    16       SUCCESS        2            2        14.62     0.1        14.72
    4        SUCCESS        2            4        18        0.1        18.1
    3        SUCCESS        2            3        8         12.1       20.1
    12       SUCCESS        2            1        20        13.04      33.04
    9        SUCCESS        2            3        14        20.1       34.1
    11       SUCCESS        2            0        21.33     13.43      34.77
    17       SUCCESS        2            2        23.85     14.72      38.56
    2        SUCCESS        2            3        7         34.1       41.1
    10       SUCCESS        2            4        30        18.1       48.1
    19       SUCCESS        2            1        16.47     33.04      49.51
    18       SUCCESS        2            0        18        34.77      52.77
    1        SUCCESS        2            1        7.06      49.51      56.57
    0        SUCCESS        2            4        10        48.1       58.1
    8        SUCCESS        2            2        20        38.56      58.56
    15       SUCCESS        2            0        26.67     52.77      79.43
    14       SUCCESS        2            2        29.23     58.56      87.79
    13       SUCCESS        2            4        36        58.1       94.1
Extended Credit Based Scheduling Algorithm Finished
```

Figure 4: Output of Extended Credit Based Scheduling Algorithm

4.2. Result of Min-Min Scheduling Algorithm

Figure 5 shows the mapping of the cloudlets with the virtual machines while implementing the Min-Min scheduling algorithm. The cloudlets were sorted in ascending order based on their lengths. Cloudlets with shorter lengths are scheduled first with the best virtual machines until all tasks were executed.

```
Starting Min Min Algorithm ...
Initialising...
1. Vm ID: 3, Vm Mips:20000.0
2. Vm ID: 1, Vm Mips:17000.0
3. Vm ID: 0, Vm Mips:15000.0
4. Vm ID: 2, Vm Mips:13000.0
5. Vm ID: 4, Vm Mips:10000.0
1. Cloudlet ID: 0, Cloudlet Length:100000
2. Cloudlet ID: 1, Cloudlet Length:120000
3. Cloudlet ID: 2, Cloudlet Length:140000
4. Cloudlet ID: 3, Cloudlet Length:160000
5. Cloudlet ID: 4, Cloudlet Length:180000
6. Cloudlet ID: 16, Cloudlet Length:190000
7. Cloudlet ID: 5, Cloudlet Length:200000
8. Cloudlet ID: 6, Cloudlet Length:220000
9. Cloudlet ID: 7, Cloudlet Length:240000
10. Cloudlet ID: 8, Cloudlet Length:260000
11. Cloudlet ID: 18, Cloudlet Length:270000
12. Cloudlet ID: 9, Cloudlet Length:280000
13. Cloudlet ID: 19, Cloudlet Length:280000
14. Cloudlet ID: 10, Cloudlet Length:300000
15. Cloudlet ID: 17, Cloudlet Length:310000
16. Cloudlet ID: 11, Cloudlet Length:320000
17. Cloudlet ID: 12, Cloudlet Length:340000
18. Cloudlet ID: 13, Cloudlet Length:360000
19. Cloudlet ID: 14, Cloudlet Length:380000
20. Cloudlet ID: 15, Cloudlet Length:400000
Starting CloudSim version 3.0
Datacenter_1 is starting...
Broker_1 is starting...
Entities started.
0.0: Broker_1: Cloud Resource List received with 1 resource(s)
0.0: Broker_1: Trying to Create VM #3 in Datacenter_1
0.0: Broker_1: Trying to Create VM #1 in Datacenter_1
```

Figure 5: Mapping of Cloudlets with Virtual Machines for Min-Min Algorithm

Figure 6 shows the output of the Min-Min scheduling algorithm. It gives a detailed execution time of each cloudlet. The values of its makespan, total execution time and throughput were calculated as follows:

Makespan = Finish Time of batch of tasks – Start time of tasks

Makespan = 115.1 – 0.1 = 115seconds.

Total Execution Time (TET) = Sum of execution time of each cloudlet

Total Execution Time = 5 + 7.06 + 9.33 + 12.31 + 9.5 + 18 + 11.76 + 14.67 + 13.5 + 18.46 + 16.47 + 18.67 + 16 + 26 + 23.08 + 20 + 24 + 31 + 29.23 + 40 = 364.04 seconds

Throughput = Number of tasks/ Total Execution Time

Throughput = 20/364.04 = 0.0549 bps

```
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_1 is shutting down...
Broker_1 is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
```

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
0	SUCCESS	2	3	5	0.1	5.1
1	SUCCESS	2	1	7.06	0.1	7.16
2	SUCCESS	2	0	9.33	0.1	9.43
3	SUCCESS	2	2	12.31	0.1	12.41
16	SUCCESS	2	3	9.5	5.1	14.6
4	SUCCESS	2	4	18	0.1	18.1
5	SUCCESS	2	1	11.76	7.16	18.92
6	SUCCESS	2	0	14.67	9.43	24.1
18	SUCCESS	2	3	13.5	14.6	28.1
7	SUCCESS	2	2	18.46	12.41	30.87
9	SUCCESS	2	1	16.47	18.92	35.39
19	SUCCESS	2	0	18.67	24.1	42.77
11	SUCCESS	2	3	16	28.1	44.1
8	SUCCESS	2	4	26	18.1	44.1
10	SUCCESS	2	2	23.08	30.87	53.95
12	SUCCESS	2	1	20	35.39	55.39
13	SUCCESS	2	0	24	42.77	66.77
17	SUCCESS	2	4	31	44.1	75.1
14	SUCCESS	2	2	29.23	53.95	83.18
15	SUCCESS	2	4	40	75.1	115.1

```
Min Min Algorithm Finished
```

Figure 6: Output of Min-Min Scheduling Algorithm

4.3 Result of Priority Based Scheduling Algorithm

Figure 7 shows the mapping of the cloudlets with the virtual machines while implementing the Priority based scheduling algorithm.

```
Starting Priority  Based Scheduling Algorithm ...
Initialising...
1. Vm ID: 3, Vm Mips:20000.0
2. Vm ID: 1, Vm Mips:17000.0
3. Vm ID: 0, Vm Mips:15000.0
4. Vm ID: 2, Vm Mips:13000.0
5. Vm ID: 4, Vm Mips:10000.0
1. Cloudlet ID: 1, Cloudlet Length: 120000, Cloudlet Qos Value:5
2. Cloudlet ID: 12, Cloudlet Length: 340000, Cloudlet Qos Value:5
3. Cloudlet ID: 15, Cloudlet Length: 400000, Cloudlet Qos Value:5
4. Cloudlet ID: 0, Cloudlet Length: 100000, Cloudlet Qos Value:5
5. Cloudlet ID: 10, Cloudlet Length: 300000, Cloudlet Qos Value:4
6. Cloudlet ID: 19, Cloudlet Length: 280000, Cloudlet Qos Value:4
7. Cloudlet ID: 4, Cloudlet Length: 180000, Cloudlet Qos Value:4
8. Cloudlet ID: 14, Cloudlet Length: 380000, Cloudlet Qos Value:4
9. Cloudlet ID: 13, Cloudlet Length: 360000, Cloudlet Qos Value:4
10. Cloudlet ID: 11, Cloudlet Length: 320000, Cloudlet Qos Value:4
11. Cloudlet ID: 18, Cloudlet Length: 270000, Cloudlet Qos Value:4
12. Cloudlet ID: 7, Cloudlet Length: 240000, Cloudlet Qos Value:4
13. Cloudlet ID: 6, Cloudlet Length: 220000, Cloudlet Qos Value:2
14. Cloudlet ID: 16, Cloudlet Length: 190000, Cloudlet Qos Value:2
15. Cloudlet ID: 5, Cloudlet Length: 200000, Cloudlet Qos Value:2
16. Cloudlet ID: 8, Cloudlet Length: 260000, Cloudlet Qos Value:2
17. Cloudlet ID: 2, Cloudlet Length: 140000, Cloudlet Qos Value:2
18. Cloudlet ID: 17, Cloudlet Length: 310000, Cloudlet Qos Value:2
19. Cloudlet ID: 3, Cloudlet Length: 160000, Cloudlet Qos Value:2
20. Cloudlet ID: 9, Cloudlet Length: 280000, Cloudlet Qos Value:1
Starting CloudSim version 3.0
Datacenter_1 is starting...
Broker_1 is starting...
Entities started.
```

Figure 7: Mapping of Cloudlets with Virtual Machines for Priority Based Algorithm

The scheduling of cloudlets here was based on priority value (also known as quality of service (QoS)), generated from users costs which is associated with each cloudlet. Cloudlets having highest priority value are given priority to execute first. Figure 8 shows the output of the Priority base scheduling algorithm. It gives a detailed execution time of each cloudlet. The values of its makespan, total execution time and throughput were calculated as follows:

Makespan = Finish Time of batch of tasks – Start time of tasks
Makespan = 110 – 0.1 = 110 seconds.
Total Execution Time for the cloudlets = 6 + 7.69 + 14 + 20 + 26.67 + 30 + 10.59 + 13.5 + 27.69 + 14.12 + 13 + 14.62 + 25.33 + 8.24 + 32 + 12.31 + 14.67 + 20 + 20.67 + 28 seconds
Total Execution for the cloudlets = 359.1 seconds.
Throughput= Number of tasks/Total Execution Time
Throughput = 20/359.1 = 0.0557 bps

```
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_1 is shutting down...
Broker_1 is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
```

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
1	SUCCESS	2	3	6	0.1	6.1
0	SUCCESS	2	2	7.69	0.1	7.79
19	SUCCESS	2	3	14	6.1	20.1
12	SUCCESS	2	1	20	0.1	20.1
15	SUCCESS	2	0	26.67	0.1	26.77
10	SUCCESS	2	4	30	0.1	30.1
4	SUCCESS	2	1	10.59	20.1	30.69
18	SUCCESS	2	3	13.5	20.1	33.6
13	SUCCESS	2	2	27.69	7.79	35.48
7	SUCCESS	2	1	14.12	30.69	44.81
8	SUCCESS	2	3	13	33.6	46.6
16	SUCCESS	2	2	14.62	35.48	50.1
14	SUCCESS	2	0	25.33	26.77	52.1
2	SUCCESS	2	1	8.24	44.81	53.04
11	SUCCESS	2	4	32	30.1	62.1
3	SUCCESS	2	2	12.31	50.1	62.41
6	SUCCESS	2	0	14.67	52.1	66.77
5	SUCCESS	2	4	20	62.1	82.1
17	SUCCESS	2	0	20.67	66.77	87.43
9	SUCCESS	2	4	28	82.1	110.1

```
Priority Based Scheduling Algorithm Finished
```

Figure 8: Output of Priority Based Scheduling Algorithm

The summary of the three algorithms is presented in Table 3 below.

Table 3: Summary of Results of the Three Algorithms

S/N	Algorithm	Makespan	Execution Time	Throughput
1	Extended Credit Based Scheduling Algorithm	94 seconds	358.5 seconds	0.0558 bps
2	Min-Min Scheduling Algorithm	115 seconds	364.04 seconds	0.0549 bps
3	Priority Based Scheduling Algorithm	110 seconds	359.1 seconds	0.0557 bps

From Table 3, extended credit based scheduling algorithm performs better than the Min-Min and priority based scheduling algorithms as it has the lowest makespan, execution time and the largest throughput value.

5. CONCLUSION

This research work shows that the extended credit based scheduling algorithm increased the quality of service parameter in terms of execution time and throughput. It is therefore recommended for task scheduling in cloud computing. Further work could be evaluation of the hybridization of these algorithms to see its effectiveness in cloud computing scheduling.

REFERENCES

1. Ahmed, D. (2017). "Glowworm Swarm Optimisation Based Task Scheduling for Cloud Computing". ICC'17 Proceedings of the second International Conference on Internet of things, Data and Cloud Computing Article No. 152.
2. Bichkule, S. S and Mante, R. V. (2017). Credit based Scheduling using Genetic Algorithm for Resource Allocation on Clouds. International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET) Website: www.ijirset.com Vol. 6, Issue 5, May 2017. DOI:10.15680/IJIRSET.2017.06050667814.
3. Buyya, R., Yeo, C., Venugopal, S., Broberg, J and Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision hype and reality for delivering computing as the 5th utility Future Generation Computer Systems, *Journal of Future Generation Computer Science*, Vol. 25, no. 6, pp. 599-616, 2009.
4. Ghanbari, S., & Othman, M. (2012). "A Priority based Job Scheduling Algorithm in Cloud Computing". International Conference on Advances Science and Contemporary Engineering 2012 50(Icasce), 778–785. https://doi.org/10.1016/j.proeng.2012.10.086
5. Griffith, E. (2016). What is Cloud Computing. Retrieved on April 26, 2020 from: https://www.pcmag.com/news/what-is-cloud-computing
6. Kokilavani, T., and DI, G. A. (2011). "Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing". International Journal of Computer Applications. Number 2 - Article 7; 2011.

7. Liu, Y., Xu, X., Zhang, L., Wang, L., & Zhong, R. Y. (2017). "Robotics and Computer – Integrated Manufacturing Workload-based multi-task scheduling in cloud manufacturing". Robotics and Computer Integrated Manufacturing, 45(October 2015), 3–20. https://doi.org/10.1016/j.rcim.2016.09.008
8. Odun-Ayo, I., Ajayi, O., Akanle B. and Ahuja R. (2017). An Overview of Data Storage in Cloud Computing. International Conference on Next Generation Computing and Information Systems (ICNGCIS). Jammu, India. IEEE. DOI: 10.1109/ICNGCIS.2017.9.
9. OpenCirus (2018). Why is cloud computing important?. Retrieved on April 26, 2020 from: https://opencirrus.org/cloud-computing-important/
10. Sahni, J., &Vidyarthi, D. P.(2015)."A Cost-Effective Deadline-Constrained Dynamic Scheduling Algorithm for Scientific Workflows in a Cloud Environment". School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi7161(c). https://doi.org/10.1109/TCC.2015.245164.
11. Thomas, A., Krishnalal G, and Raj, J. V. P. (2015). "Credit Based Scheduling Algorithm in Cloud Computing Environment". Procedia - Procedia Computer Science, 46(ICICT, 2014), 913–920. https://doi.org/10.1016/j.procs.2015.02.162
12. Zhang, P., & Zhou, M. (2017). "Dynamic Cloud Task Scheduling Based on a Two-Stage Strategy".IEEE Transactions on Automation Science And Engineering 1–12.