Relating job scheduling algorithms on job lengths and number of cloudlets in cloud computing

Dr. K. K. Goyal[*1], Vivek Jain[#2], Dr. Shivani Chauhan[#3]
*RBS Mgmt Tech Campus Agra India[*1], Ph.D. Scholar, Bhagwant Univ., Ajmer, India[#2], Bhagwant Institute of Technology, Muzaffarnagar[#3]*

***Abstract*:** *As the world of computing is growing exponentially in global scenario and more applications of computing helping the world in multiple ways. With the increase in the applications, users, and data the need of more robust scheduling is an obvious requirement of the system. Researchers all over the world are providing their valuable ideas to handle this job base in an efficient manner. Job scheduling depends on the results requirement and on the nature of the job environment. In this paper, we have presented the experimental results with four job scheduling algorithms namely FCFS, SJF, Min-Min, Min-Max on a cloud environment by varying the no of cloudlets and job length and analyses its impacts on total execution time and waiting time. On the basis of the results obtained it is found that Min-Max outperforms in both the situations either process load is minimum or maximum.*

Keywords: **Cloud computing, Job Scheduling, FCFS, SJF, Min-Min, Min-Max, cloudlet.**

1. INTRODUCTION

From the invention of computer to till today, the application of computer is always appreciated for minimizing the processing time, cost, waiting time, execution time etc. and improving the overall performance of the system. In the present time, cloud computing becomes very popular for providing all type of computing services over the internet on pay per use basis. As the number of applications and job lengths increasing exponentially in the cloud computing. So, the job scheduling plays important role to handle them in such a way that it will process these jobs with minimum execution time and waiting time.

There are various scheduling algorithms proposed namely FCFS, SJF, min-min, min-max, etc. and used for managing the submitted jobs. Here in this paper, we compare the performance of these scheduling algorithms in the cloud computing perspective. We analyse these algorithms with the given similar process load in a predecided environmental setup and obtained the total execution time and waiting time for the given process load. The basic structure is given the figure 1.

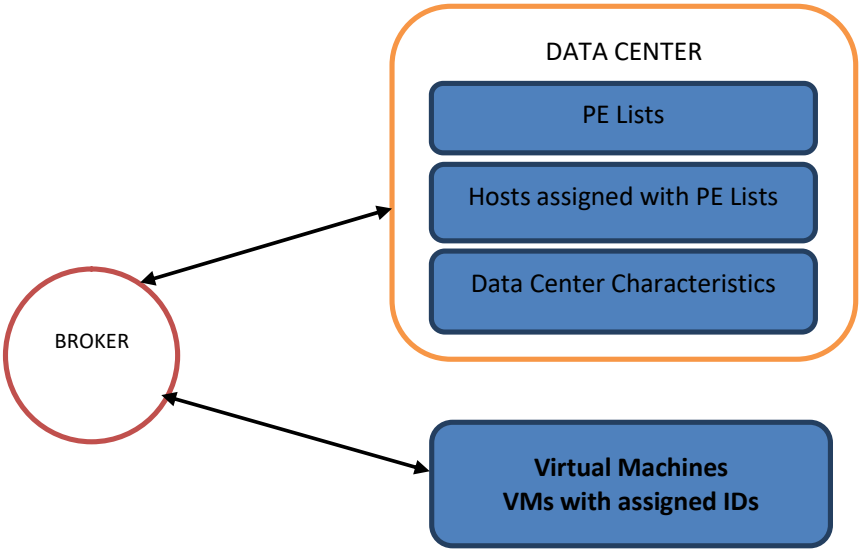

Figure 1 : Basic structure of cloud computing

2. PROPOSED MODEL AND METHODOLOGY

There are the steps to be followed to execute the test for examining the impacts of scheduling strategies on the total execution time and total waiting time with varied job lengths by incrementing the number of cloudlets. Before execution first let us understand the structure of the setup, we are going to use. There are three major building blocks that we using in this set up: (1) Data Center, (2) Broker, and (3) VM. These three are defined below:

1. Data Center:

Following the steps to create a Data Center

1. We need to create a list to store one or more Machines
2. A Machine contains one or more PEs or CPUs/Cores. Therefore, should create a list to store these PEs before creating a Machine.
3. Create PEs and add these into the list.
4. Create Hosts with its id and list of PEs and add them to the list of machines.
5. Create a DatacenterCharacteristics object that stores the properties of a data center: architecture, OS, list of Machines, allocation policy: time- or space-shared, time zone and its price (G$/Pe time unit).
6. Finally, we need to create a PowerDatacenter object.

As listed above, first of all we need to decide the number of machines to be created in a data center, and the configuration of these machines to be decided including the number of CPU's/Cores or PEs in each machine for example dual core, quad core or octa core with the pre-decided MIPS of each processor or PEs. After defining these host machines and its PEs, add all the PEs from all the host machines to prepare multiple PE_Lists. So, now we have multiple PE_Lists with clearly defines the PEs with its capabilities. Assign IDs to hosts and then add list of PEs to these IDs, the important parameters like RAM, storage size and bandwidth has to initialize first, here we have fixed these as following:

Ram=2048 MB
Host Storage=1000000
Bandwidth=10000

In addition to this, we will also add provisioners for the handling the resources like RamProvisionerSimple for handling the allocation of RAM between the created machines or PEs, BwProvisionerSimple for handling the bandwidth allocation between related machines. Virtual Machine Scheduler is very important for creating a virtualized cloud computing environment , here we have VmSchedulerTimeShared(peList1) for Time-shared, VmSchedulerSpaceShared(peList1) for space–shared, and VmSchedulerOportunisticSpaceShared(peList1) for opportunistic space shared allocation policy for the PEs to VMs. Here, in this example we have used the VMSchedulerTimeShared().

Now, create a DatacenterCharacterics object that stores theproperties of a data center: architecture, OS, list of Machines, allocation policy: time- or space-shared, time zoneand its price (G$/Pe time unit). Here, in our example we have used the initial values as follows:

```
String arch = "x86";      // system architecture
String os = "Linux";          // operating system
String vmm = "Xen";
doubletime_zone = 10.0;         // time zone this resource located
double cost = 3.0;              // the cost of using processing in this resource
doublecostPerMem = 0.05;                // the cost of using memory in this resource
doublecostPerStorage = 0.1;   // the cost of using storage in this resource
doublecostPerBw = 0.1;                          // the cost of using bw in this resource
LinkedList<Storage>storageList = new LinkedList<Storage>();
```

Now, all the parameters are initialized and then create the object of DatacenterCharacteristics. After initializing and creating he characteristic object create the Data Center. Now, Data Center is created and ready for execution.

(2) Virtual Machine: In this experimental set up, second building block is the virtual machine. The virtual machine configuration that we are using here is as follows:

```
long size = 10000; //image size (MB)
int ram = 512; //vm memory (MB)
longbw = 1000;
intpesNumber = 1; //number of cpus
String vmm = "Xen"; //VMM name
intnum_user = 1;   // number of grid user
```

(3) Broker: The third building block is the broker. The Broker is the central repository that keep track of all the available resources at the data center and allocation of these resources to the host by data center.

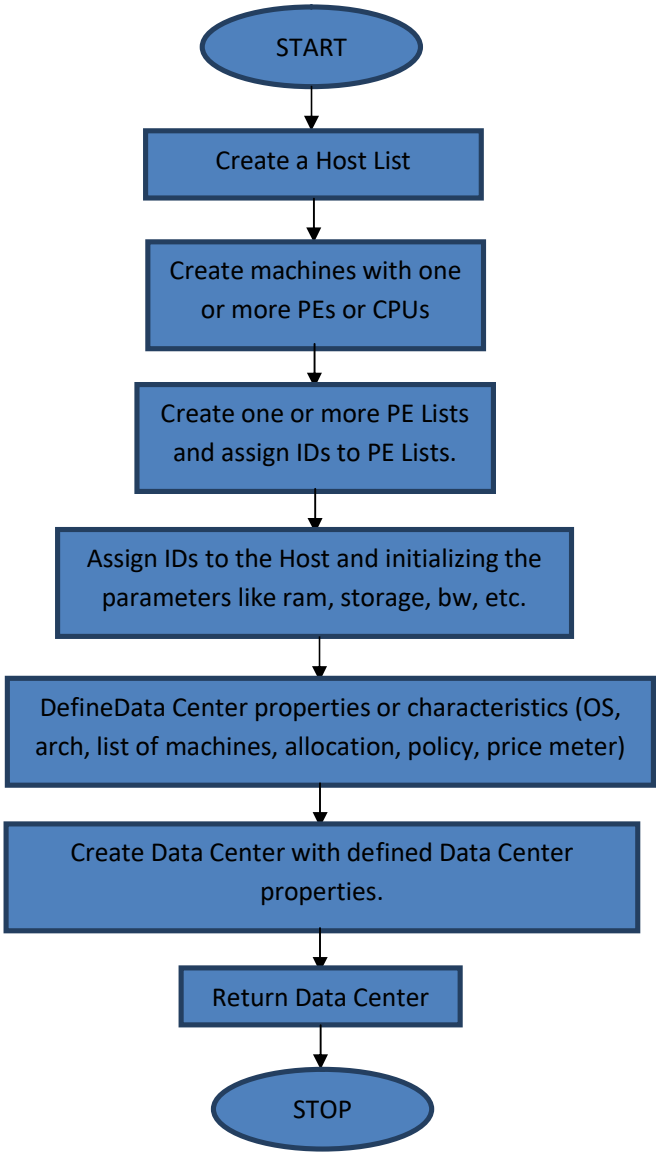

Figure 2 : Flow Chart of creating a Data Center

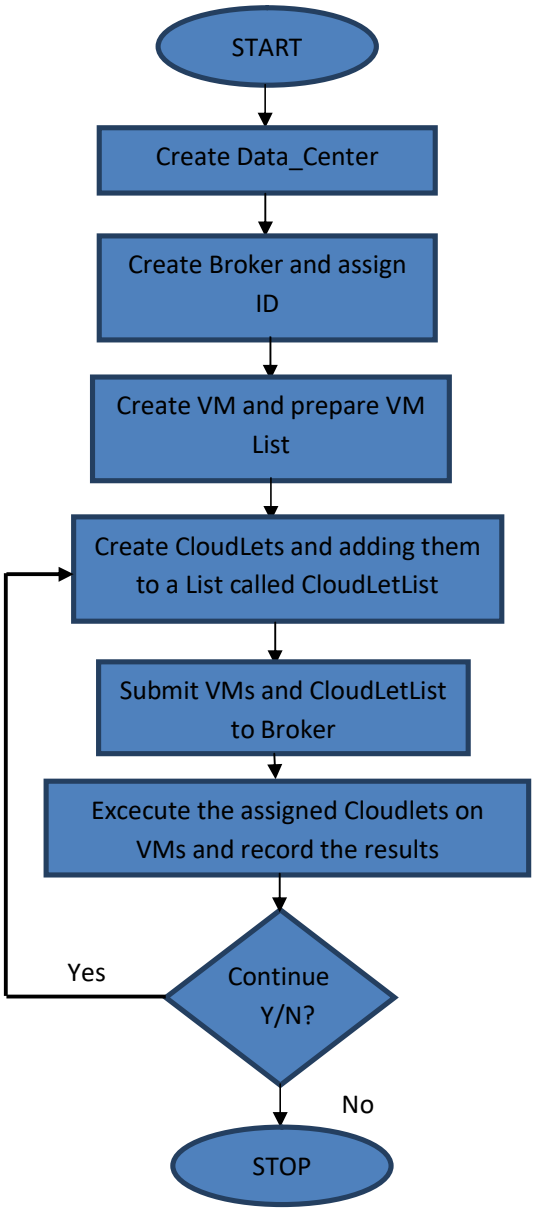

Figure 3: Flow Chart of executing our experimental setup

Now, after creating the datacentre, virtual machines, and the broker, the next step is to Create VMs and preparing the VM list, and creating the Cloudlets and preparing a cloudlist. Now, send these VMs and Cloudlets to broker. This will makes the data center, broker, vm list and cloudlets ready to execute. This whole process is divided into two flowcharts and shown in the figure 2 and figure 3. In the figure 1 flow chart shows the steps to create a data center and in figure 2 flow chart show the steps to execute the experimental setup.

3. RESULT ANALYSIS

In the experimental setup, the following are the system requirements to be set before execution:

No of DC=1(2 Host with Quadcore processor of 2000 MIPS)
No of VM=5 with (500, 750, 1000, 1250, 1500) MIPS processing capabilities
Cloudlet Allocation Policy= Space shared;
No. of grid user =1

Now following are results recorded with different algorithms like FCFS (First Come First Serve), SJF (Shortest Job First), MIN-MIN (on the basis of minimum execution time on available VM with multiple cloudlets and then minimum weighted process from the selected processes), and MIN-MAX (on the basis of minimum execution time on available VM with multiple cloudlets and then maximum weight process from the selected processes) used for scheduling the submitted processes i.e. cloudlets.

(1) **Results Recorded with FCFS (First Cum First Serve) Algorithm :** In this test case, number of Cloudlets are varied from 10-1000 and the total execution time, total waiting time, and total job length is computed with scheduling algorithm FCFS to process the submitted Cloudlets i.e. processes and recorded and is listed above in Table-1.

No of Cloudlets	Execution Time	Waiting Time	Total length
10	156.50253333333333	78.01173333333332	134640
20	302.6143333333333	469.6136	258120
30	425.6845333333332	1136.2644666666665	364370
40	548.7718	2049.506533333333	470190
50	671.3977333333333	3208.424000000001	575810
60	793.9469999999997	4612.734533333334	681550
70	916.5800666666669	6261.6100000000015	787280
80	1031.6890666666668	8156.535200000003	882440
90	1156.4310000000003	10282.509800000003	990000
100	1266.0229333333334	12657.567000000001	1079340
200	3033.478599999996	61052.561733333314	2613800
300	4262.581266666665	133412.4615999999	3673220
400	5489.611266666657	230330.2994666666	4730680
500	6720.481866666658	351823.2729999998	5790730
600	7949.955799999992	497909.649866667	6850610
700	9176.912999999984	668568.5018000006	7907970
800	10405.503533333316	863788.6451999998	8967130
900	11731.030333333334	1084831.1313333341	10155520
1000	13146.658199999998	1331358.7606000009	11378690

Table 1: Total Execution and Waiting Time using FCFS Algorithm

(2) **Results Recorded with SJF (Shortest Job First) Algorithm :** In this test case, number of Cloudlets are varied from 10-1000 and the total execution time, total waiting time, and total job length is computed with scheduling algorithm SJF to process the submitted Cloudlets i.e. processes and recorded and is listed above in Table-2.

No of Cloudlets	Execution Time	Waiting Time	Total length
10	155.75466666666665	76.60053333333333	134640
20	297.6293333333333	421.0601333333333	258120
30	420.8473333333334	974.4285333333335	364370
40	543.4242666666667	1769.3921333333335	470190
50	666.0653333333333	2808.1927333333333	575810
60	788.7277333333333	4092.1933333333336	681550
70	911.2991999999999	5618.700399999999	787280
80	1016.4463333333333	7124.758666666666	882440
90	1141.2013999999997	9118.561133333335	990000
100	1244.2291333333333	10963.356800000001	1079340
200	3030.000733333334	56681.66786666668	2613800
300	4259.17233333333	117408.5037999999	3673220
400	5485.763266666669	202446.23453333354	4730680
500	6715.509933333327	312087.2656666666	5790730
600	7945.23033333332	446227.53386666666	6850610
700	9171.916999999994	604569.7946666667	7907970
800	10400.110066666657	787418.5020666672	8967130
900	11727.240866666676	995569.6707333346	10155520
1000	13127.398000000008	1226076.3371333345	11378690

Table 2: Total Execution and Waiting Time using SJF Algorithm

(3) **Results Recorded with MIN-MIN Algorithm :** In this test case, number of Cloudlets are varied from 10-1000 and the total execution time, total waiting time, and total job length is computed with scheduling algorithm MIN-MIN to process the submitted Cloudlets i.e. processes and recorded and is listed above in Table-3.

No of Cloudlets	Execution Time	Waiting Time	Total length
10	131.3476	72.28439999999999	134640
20	260.8693333333333	401.7418666666666	258120
30	362.56739999999985	979.0679333333334	364370
40	472.9957999999999	1753.3234	470190
50	573.9426666666666	2754.1098666666667	575810
60	684.4226666666668	3952.1691999999994	681550
70	785.5078666666669	5374.1254666666655	787280
80	888.5661333333334	6994.946466666667	882440
90	992.634	8820.473866666667	990000
100	1071.6562666666666	10870.825466666667	1079340
200	2616.4293333333353	52671.022000000026	2613800
300	3675.9809999999998	115022.51153333331	3673220
400	4734.026999999999	198577.77540000007	4730680
500	5793.606333333333	303239.59166666644	5790730

600	6853.564666666666	429136.2341333338	6850610
700	7912.431333333305	576366.7365333327	7907970
800	8970.756666666659	744511.0108	8967130
900	10108.99473333333	935875.0673333331	10155520
1000	11300.012799999995	1149613.835333334	11378690

Table 3: Total Execution and Waiting Time using MIN-MIN Algorithm

(4) **Results Recorded with MIN-MAX Algorithm :** In this test case, number of Cloudlets are varied from 10-1000 and the total execution time, total waiting time, and total job length is computed with scheduling algorithm MIN-MAX to process the submitted Cloudlets i.e. processes and recorded and is listed above in Table-4.

No of Cloudlets	Execution Time	Waiting Time	Total length
10	130.90800000000002	72.20066666666666	134640
20	258.57460000000003	405.2908666666667	258120
30	361.9747333333334	979.9566	364370
40	470.5113333333333	1757.0548000000001	470190
50	573.3219333333334	2755.8938000000003	575810
60	681.8245333333335	3956.3477333333335	681550
70	784.7659333333336	5376.048199999999	787280
80	879.4564000000004	7000.570666666667	882440
90	986.094933333334	8811.664799999999	990000
100	1079.6325333333339	10862.09226666667	1079340
200	2612.597533333333	52663.40666666664	2613800
300	3672.243733333332	115038.32660000003	3673220
400	4730.151999999994	198594.18326666666	4730680
500	5791.056733333331	303312.88120000024	5790730
600	6849.539133333332	429177.59579999937	6850610
700	7907.530266666655	576298.3601999999	7907970
800	8966.17719999999	744432.2065999985	8967130
900	10156.633400000008	935851.0450000004	10155520
1000	11382.453000000005	1149230.4510666665	11378690

Table 4: Total Execution and Waiting Time using MIN-MAX Algorithm

After the close comparison of the results obtained by the applying these scheduling algorithms in cloud computing for executing the multiple processes on multiple virtual machines in the pre-decided fixed environmental parameter values. It is observed that when we varied the number of cloudlets, the total execution time, and total waiting time is minimum for the MIN-MAX algorithm, and maximum for the FCFS algorithm. Hence, MIN-MAX algorithm outperforms in the experimental test. The results are displayed through the charts shown below:

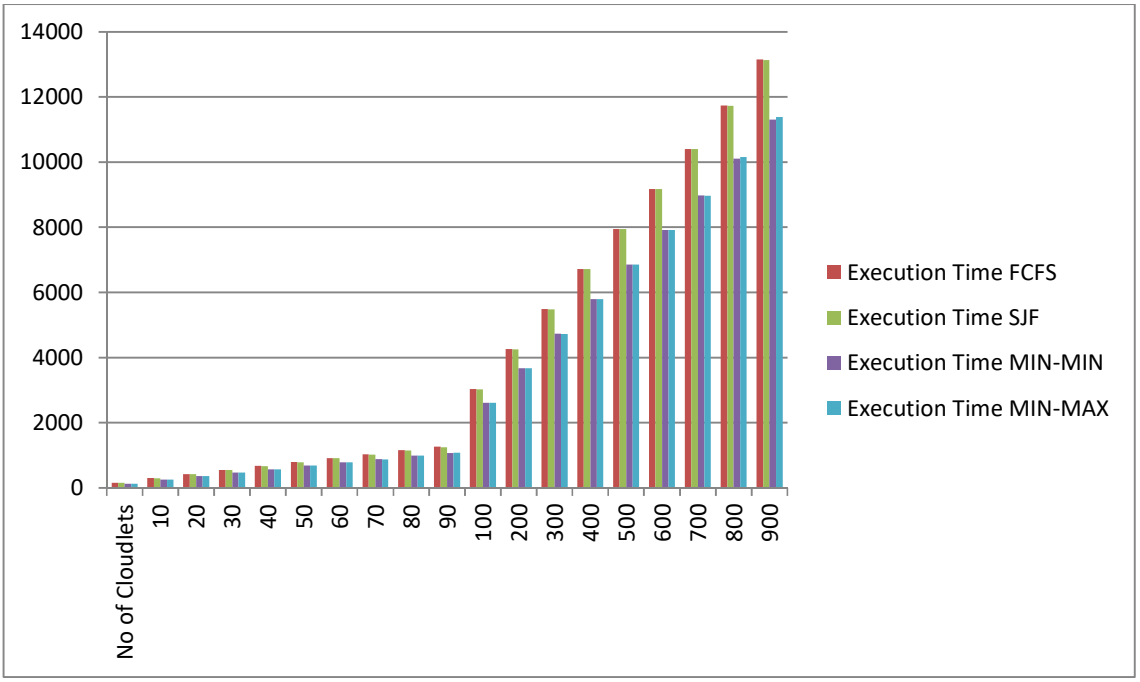

Figure 4: Chart showing total execution time for these four scheduling algorithms

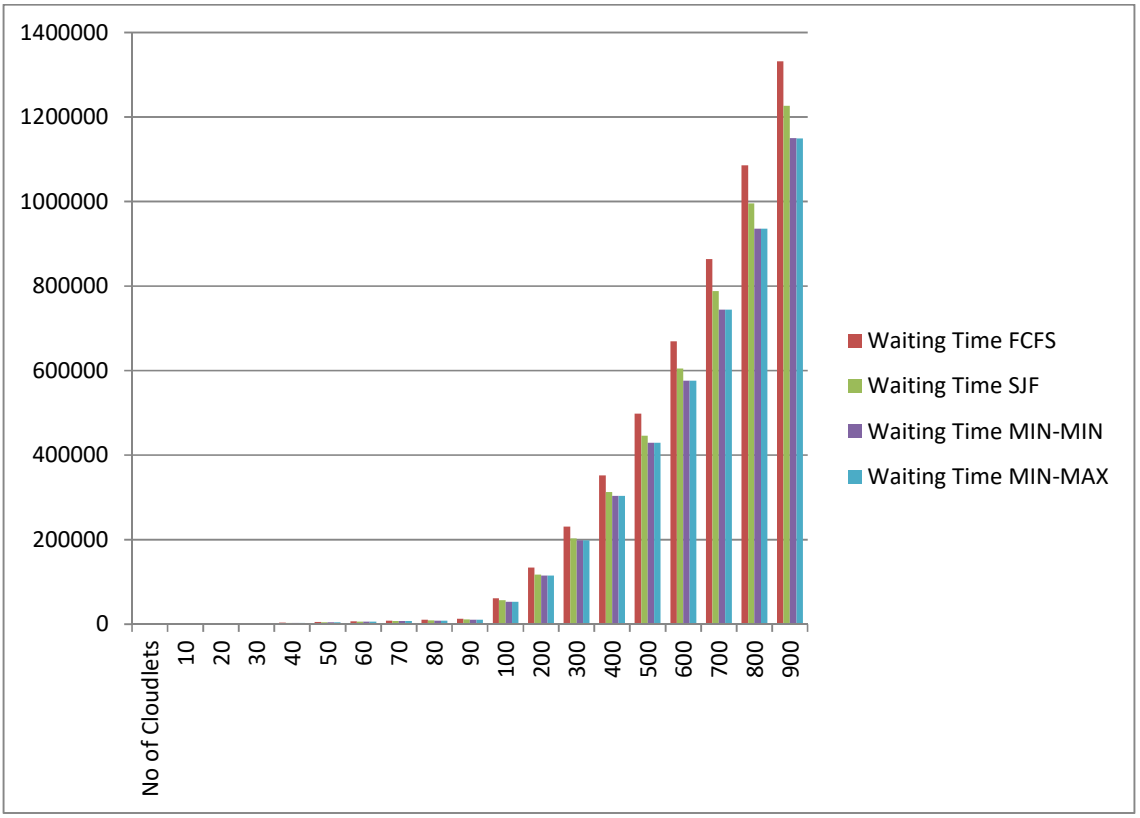

Figure 5: Chart showing total waiting time for these four scheduling algorithms

4. CONCLUSION

As we know that cloud computing provides an execution environment that is cost effective and provides its services on pay per use model. The nature of the traffic is bursty, i.e. unpredictable. There may be sometimes very high traffic and sometimes there may very less traffic. Scheduling algorithm plays a key role in serving the process. Here, we have tested these scheduling algorithms with varying number of cloudlets and total job length and recorded the total execution time and total waiting time during the execution of these cloudlets or processes. It is observed that among these four scheduling algorithms that we used here (i.e. FCFS, SJF, MIN-MIN, MIN-MAX), MIN-MAX out performs even with the less (i.e. only 10) number of cloudlets as well as with 1000 no of cloudlets. Hence it is best scheduling algorithm among these four for scheduling the cloudlets.

REFERENCES

[1]. Essaies M. Nabil T., "A survey on Cloud Computing Scheduling Algorithms", 2016 Global Summit on Computer & Information Technology, pp. 43-47.

[2]. R. Durga Lakshmi, N. Srinivasu, "A Review and Analysis of Task Scheduling Algorithms in Different Cloud Computing Environments", International Journal of Computer Science and Mobile Computing [IJCSMC], Vol. 4, Issue. 12, December 2015, pg.235 – 241, ISSN 2320–088X

[3]. A. Ahluwalia, V. Sharma, "A Survey on Task Scheduling Techniques in Cloud Computing", International Journal of Engineering Trends and Technology (IJETT) – Volume 33 Number 9- March 2016, pp. 448-452

[4]. N. Anwar, H. Deng, "A Hybrid Metaheuristic for Multi-Objective Scientific Workflow Scheduling in a Cloud Environment", Applied Sciences, MDPI, 2018, 1-21

[5]. S.H.H. Madni, M.S.A. Latiff, Y. Coulibaly and S. M. Abdulhamid, "An Appraisal of Meta-Heuristic Resource Allocation Techniques for IaaS Cloud", Indian Journal of Science and Technology, Vol 9(4), DOI: 10.17485/ijst/2016/v9i4/80561, January 2016, pp. 1-14

[6]. Z.G Chen, Z.H Zhan, Y.J. Gong, T.L.Gu, F.Zhao, H.Q.Yuan, X, Chen, Q. Li, J. Zhang, "Multiobjective Cloud Workflow Scheduling: A Multiple Populations Ant Colony System Approach", IEEE TRANSACTIONS ON CYBERNETICS, VOL. 49, NO. 8, AUGUST 2019, pp. 2912-2926

[7]. S. Bilgaiyan, S. Sagnika, M. Das, "An Analysis of Task Scheduling in Cloud Computing using Evolutionary and Swarm-based Algorithms", International Journal of Computer Applications (0975 – 8887), Volume 89 – No.2, March 2014, pp. 11-18

[8]. A. Jain, A. Upadhyay, "Cloud Scheduling using Meta Heuristic Algorithms", IJCSE, Vol. 5, Issue – 10, E-ISSN: 2347-2693, pp. 132-139

[9]. M. Habibi, N.J. Navimipour, "Multi-Objective Task Scheduling in Cloud Computing Using an Imperialist Competitive Algorithm", IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 5, 2016, pp. 289-293

[10]. L. Singh, S. Singh, "A Survey of Workflow Scheduling Algorithms and Research Issues", International Journal of Computer Applications (0975 – 8887), Volume 74– No.15, July 2013, pp.21-28.

[11]. F. MA, F. LIU, Z. LIU, "Multi-objective Optimization for Initial Virtual Machine Placement in Cloud Data Center", Journal of Information & Computational Science, Vol. 9, Issue 16, (2012), pp. 5029–5038