International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 10 • Issue 2 pp. 901-912 Dec 2018 www.csjournals.com

Performance Analysis of Tasks Scheduling Algorithms in Homogeneous Cloud Environment

Jai Bhagwan[1], Sanjeev Kumar[2]
[1,2]Department of Computer Science & Engineering
Guru Jambheshwar University of Science & Technology, Hisar, India

Abstract: Cloud computing is becoming most popular nowadays in various fields. It is because it provides all types of computing facilities viz. application uses for data processing, applications development without having personal software and noteworthy physical infrastructure, virtualized resources usages etc. The information is stored and hardware is installed on a 'cloud' in a remote area and can be accessed anywhere with a single and simple computing device. This allows pay-per-usage and reduces computing cost. That's why many industries preferably use cloud computing and this resultsincrements in the number of usersfrequently. This situation leads to increase the load on a cloud. In order to manage the load, many techniques can be used such as optimized hardware, optimized network, optimized databases etc. one of them is scheduling of task submitted by users on virtual machines of a datacenter. The main exertion of the scheduling is to assign a virtual machine to a specific task. For the purpose of scheduling, many algorithms have been developed. In this paper, FCFS, Round-Robin, Min-Min, and Max-Min have been used for simulation in order to measure their performance while working with scientific tasks. Four scenarios have been designed for simulation purpose. After simulation, it is found that Max-Min performs better than other algorithms in accounts of makespan, resource utilization under the desired cost in maximum situations of all scenarios.

Keywords: Cloud Computing Environment, Workflow Scheduling, Computing Resources, Makespan, Scheduling Algorithms.

1. Introduction

Cloud computing is becoming the necessity of every IT organization in order to process its data under minimum cost. Cloud computing is a process where information is stored in datacenters and can be accessed in every corner of the world with the help of a common computing device. Every day many industries transfer their applications and data to the cloud due to its dynamic and flexible infrastructure. The information can be managed over a cloud with minimal management [1]. The cloud computing technology follows a parallel and distributed system where integrated usage of the computing resources is done based on their availability. In order to execute the scientific applications, a large number of communication resources and computation power are needed. Workflows are one of the typical models used in science and engineering which requires powerful computing resources.As a result, scheduling of workflows in cloud computing is a big challenge. In workflow scheduling, it is to assign the tasks to available computing resources without losing the dependency of tasks [2]. So, task scheduling is a hot topic of research in cloud computing.

The academic organizations and industries are moving towards the cloud in a regular interval. So, the numbers of users are getting increased. The main objective of task scheduling is to map a task with the best available resource. Task scheduling can be used for proper utilization of computing resources which are available in a cloud system. The major goals of task scheduling include reducing makespan of tasks, an increment in computing resources utilization, reducing resources cost and managing system loads [21]. The algorithms used in this paper for performance analysis are followings.

a) First Come First Serve –This is the simplest and non-preemptive algorithm which maps the tasks to available computing resources in a first in first out manner. In FCFSalgorithms, the smaller tasks have to wait for a long time while larger tasks are being executedor in waiting. So, it leads to bad turnaround and response times andit causes poor performance than other algorithms for task scheduling[16][18].
b) Round-Robin – This algorithm apreemptive type algorithm which works on the principle of FCFS algorithm except it uses a time slicei.e.every job has to spend a specific time slice on available computing resources for its completion.If a task is taking a lot of time for its execution then the CPU is transferred to another task for execution and the previous task has to wait to its turn [19].
c) Min-Min – Min-Min algorithm begins its working with the set of Meta-Task listwhere all submitted tasks are present. First of all, it finds the tasks available in Meta-Task list which ishaving minimum

completion time on all computing resources and then faster available resources are found. After this,Min-Minselects the task from Meta-Task list havingan overall minimum expected completion time and sends it to the faster available resource. Now the executed task is removed from Meta-Task list and this process remainscontinues till all tasks are assigned to available computing resources.This is how;Min-Minschedules small tasks first.The disadvantage of Min-Min is that it increases the makespan and reduces resource utilization when the numbers of smaller tasks are less in amount than larger tasks [17][20].

d) Max-Min – This algorithmfollows a principle of maximum expected completion time. The Max-Min algorithm is alike to Min-Min, except it bargains for the maximum expected completion time of each task in Meta-Task list and maps it to the faster resource. This algorithm executes larger tasks first and then smaller tasks are executed concurrently. In reverse to Min-Min, it gives better performance when a few larger tasks are available along with a lot of smaller tasksin Meta-Task list [20].

The remaining paper is organized as follows. It has been described the related work in Section 2. Section 3 is having experimental setup describing simulation parameters, cost plan, scientific tasks, and performance metrics. Section 4 includes the results discussions. Finally section 5 briefs the conclusion.

2. Related Work

In cloud computing,research is in progress in the field of task scheduling and load balancing. Many algorithms have been proposed and tested with different parameters and scenarios.Authors in [1] designed an Extended Intelligent Water drops Algorithm for scheduling of workflowsin cloud computing. This algorithm is an extension of the Intelligent Water Drops algorithm. The experiments were carried out taking into accountofmakespan and cost. The results showed that the proposed IWDC algorithm performed well in most cases than Max-Min, Min-Min, FCFS, Round-Robin, and MCT algorithms.Authors of [2] introduced an adaptive workflow scheduling algorithm forthe achievement of reducedmakespan and load balancing. The proposed technique out-performedas compared toMin-Min, Max-Min, HEFT, Re-LSDS, Re-DCP-G and AHEFT methodsin term of makespan and load balancing.Authors of [3] proposed a scheduling scheme using Ant Colony Optimization (ACO) by taking into accountthe user-specified QoS constraints. For calculation of the pheromone values,the proposed methodusesworkflow and heuristics. In [4], the authors designed PSO based technique for workflows scheduling problem; the parameters likedata transmission cost and computation cost are taken into consideration by the proposed scheduler for experiments. Authors usedan Immune Particle Swarm Optimization in [5]. The authors designed a grid task scheduling model whichwas used for the solution ofmulti-objective optimization problems of heterogeneous grid.Authors implemented Max-Min [6] algorithm to execute simultaneously a number of dependent and independent tasks having large execution time. The proposed technique is better as it is able to schedule a number of tasks on various virtual machines by reducing execution time. The authors of [7] introduced and enhanced Max-Min scheduling algorithm by selecting average or nearest greater than average tasks instead of scheduling larger tasks first. The results show that the execution time is reduced by proposed technique and the load balancing is also achieved efficiently.In [8], scientists designed a modified Ant Colony Optimization technique which uses probability for scheduling the tasks on target machine. After experiments, the authors find that the proposed algorithm reduces average makespan in comparison to the basic Ant Colony Optimization technique.Scientists in [9] proposed an algorithm which is based on Max-Min algorithm. In this algorithm, the tasks are collectedbased onthe Max-Min in the first phase.For resources selection, mean of completion time was compared with available resources completion time. If resources completion time were found less than or equal to mean completion time, the taskshaving maximum completion time were scheduled first otherwise the tasks having the best maximum execution time were scheduled. Many authors also introduced hybrid algorithms like in [10] an improved Max-Min algorithm was proposed for the solution of task scheduling in the cloud environment. In this algorithm, the taskshaving maximum execution time were assigned to the resources having minimum completion time rather than assignment of taskshaving maximum completion timeto the resources which had minimum execution time and results were found better.In [11], the authors introduceda filter Min-Min algorithmwhichfollowed the criteria of standard deviation and average completion time of all resources and selected either Max-Min or Mix-Min for tasks and resources mapping. Similarly, scientists in [12] designed a Resource Aware Scheduling Algorithm which is called RASA. In this algorithm, Max-Min was applied when the available virtual machines were even in number; otherwise, Min-Min was applied.Researchers of [13] introduceda switcher algorithm whichselects between the two algorithms either Min-Min or Max-Minunder specified conditions. Similarly, authors of [14] improved makespanusing aproposed technique where the techniquewas designed to select either Min-Min or Max-Min algorithm for tasks mapping to available virtual machines after comparing thetasks summation time. The authors of [15] introduced a SOS (Symbiotic Organism Search) population-based meta-heuristic novel algorithm which was used to solvenumerical optimization problems.The proposed algorithm SOS mimics the symbiotic associations among various species in an ecosystem.

3. Simulation Setup

For evaluation of the scheduling algorithms, an open source tool WorkflowSim has been used. This tool is an enhancementof CloudSim which has been written in Java. WorkflowSim is having the functionalities to simulate scientific workflows. The experiments are performed over a machine with Intel(R) Core(TM) i3-5005U CPU @ 2.00 GHz having 4.00 GB RAM and running Windows 10 64-bit operating system.

Further, one PowerDatacenter has been createdwith one host, 2048 MB RAM, 2000 MIPSprocessing speed and bandwidthof 10000 bps. The datacenter's is havinga Linux operating system. The system architecture of this datacenter is x86 and VMM is Xen. Table 1 is showing the parameters configured for simulation with homogeneousVMs. The number of VMs varies according to experiments. VMs contain 1 PE, 512 MB RAM, 1000 bps Bandwidth, 1000 MIPS of Processing Element and 10000 Image size. The scheduling policy isset as Time Shared.

Table 1. Simulation Parameters

Parameter	Values
Number of Data-Centers	1
System Architecture	x86
Number of Hosts	1
VMM	Xen
OS	Linux
Transfer Rate	15 MB/s
Numbers of VMs	3, 6, 12, 24 and 48
CPU (PEs Number)	1
RAM per VM	512 MB
Bandwidth	1000
PE's MIPS per VM	1000
Image Size	10000
Policy Type	Time Shared

3.1 Cost Plan

The cost plan of a system includes processing cost, bandwidth cost, memory usages cost and storage cost which is illustrated in table 2. The cost evaluation parameters' values are set as 3.0 per processor, 0.1 for storage, 0.05 per MB of RAM and 0.1 per MB of Bandwidth.

Table 2. Cost Plan

Resource	Processor	RAM	Storage	Bandwidth
Size	1000 MIPS	512 MB	Unlimited	1000 bps
Cost	3.0 per processor	0.05 per MB	0.1	0.1 per MB

3. 2Scientific Tasks

For simulation, four scientific workflows named CyberShake_1000, Montage_1000, Inspiral_1000 and Sipht_1000 have been used.

CyberShakeis used to define the earthquake hazards by generating synthetic seismograms. It requires a large memory and CPU for execution purpose. **Montage** is used to build the bulky image mosaics of the sky. It requireslow CPU processing capacity for execution. **Inspiral**application is used to sense gravitational waves signatures in data collected by large scale interferometers. **Sipht**is used to systematize the process of searching for sRNA encoding-genes for bacterial replicons in bioinformatics centres. Maximumnumber of the tasks of this workflow requiresa powerful CPU and less usage of I/O devices [1].

3.3 Performance Metrics

The performance of a technique can be measured by setting a few constraints. In this paper, three performance metrics are used which are described below.

3.3.1 Makespan

Makespan [24] is defined as the finishing time of the last task which has been executed. It is an important metric to measure the performance of an algorithm in cloud computing.A lowermakespanspecifies that the method isan efficient one. The makespan is calculated by the given the equation (i).

$$\text{Makespan} = \max(CT_i)_{ti \in tasks} \quad \text{……. (i)}$$

Where, CT_i is completion time of task i

3.3.2 Resource Utilization

Another important metric is resource utilization. The more utilization rate specifies thatthe method is efficient [24][27]. The average resource utilization is recorded by a formula given in the equation (ii).

$$\text{Average Resource Utilization} = \frac{\sum_{i=1}^{n} \text{Time taken by resource } i \text{ to finish all jobs}}{\text{Makespan} \times n} \text{……. (ii)}$$

3.3.3 Cost

Cost is alsoa vital metric of simulation because end-users desire servicesata minimum cost. The cost can be measured by the equation (iii).

$$\text{Cost} = \text{Cost per sec. of Resources} \times (\text{Finished Time} - \text{Execution Start Time}) \text{……. (iii)}$$

4.Results Discussion

In this paper, asetof four scenarios and a set of 3, 6, 12, 24 and 48 virtual machines have been used for simulation purpose.The performance of the four most widely used algorithms viz. FCFS, Round-Robin, Min-Min, and Max-Min has been measured. The results which are recorded in term ofmakespan arespecified in table 3.

Table 3. Scheduling AlgorithmsMakespan using Scientific Workflows

Scenarios	FCFS	Round-Robin	Round-Robin	Min-Min	Max-Min
Scenario – 1 (CyberShake_1000)	3	7650.22	7667.89	7664.51	**7649.03**
	6	3882.95	3902.59	3912.21	**3880.04**
	12	2021.24	2045.15	2046.4	**2017.42**
	24	1104.66	1114.68	1116.38	**1086.16**
	48	650.59	656.22	656.4	**627.5**
Scenario - 2 (Montage_1000)	3	**4037.26**	4039.14	4042.59	4038.71
	6	2195.46	2197.78	2195.84	**2195.25**
	12	1272.55	1273.07	1276.12	**1271.11**
	24	813.65	813.58	817.56	**812.76**
	48	587.91	587.83	594.07	**587.69**
Scenario - 3 (Inspiral_1000)	3	76119.42	76069.73	76117.01	**76065.49**
	6	38174.89	38193.58	38300.49	**38058.03**
	12	19322.66	19294.63	19340.81	**19141.6**
	24	9800.6	9862.31	9971.94	**9707.77**
	48	5073.03	5137.73	5403.99	**5011.64**
Scenario – 4 (Sipht_1000)	3	57896.74	59226	60297.4	**57893.83**
	6	29655.61	30353.58	32097.6	**29350.74**
	12	15279.88	16433.81	17480.8	**15207.76**
	24	**8374.62**	9860.95	10185.01	8530.05
	48	6104.33	6104.37	6759.95	**5315.73**

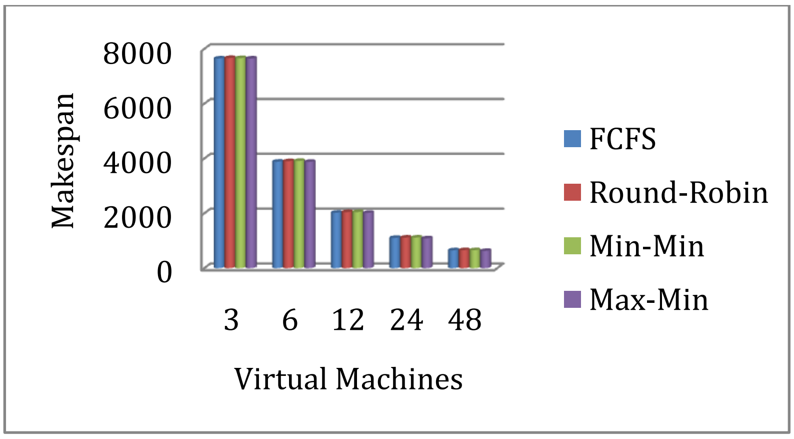


Figure 1. Makespan Comparison for CyberShake_1000

Figure 1 is representing the results of makespanwhich have been found after simulation with the dataset CyberShake_1000. In this figure,the x-axis represents to VMs and y-axis to makespan. One can see that the makespan is reducing with the increment of the number of virtual machines. This is because a large number of VMs work in a faster way for a certain amount of tasks. The CyberShake_1000 dataset comprises 1000 tasks; among them, maximum numbers of tasks are of large and medium lengths. With all sets of VMs i.e. 3, 6, 12, 24 and 48, Max-Min out-performs than all algorithms because the Max-Min executes larger tasks on faster resources with concurrent execution of small tasks as the numbers of small and medium-sized tasks in this scenario are higher than larger tasks.

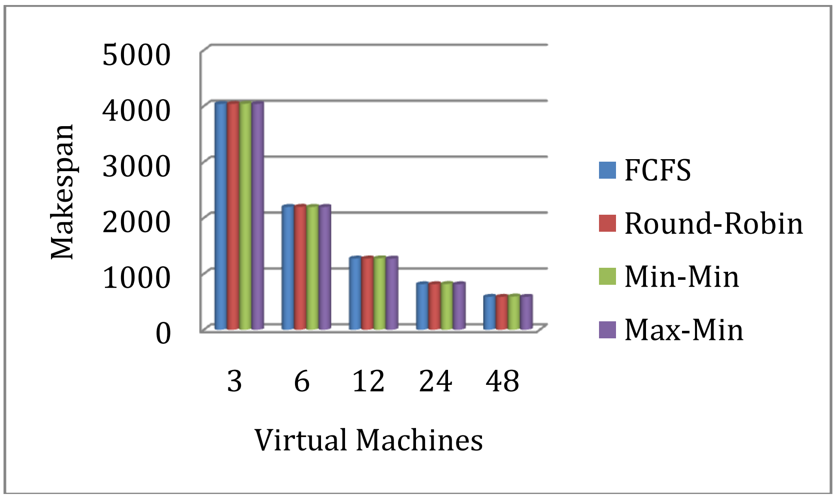


Figure 2. Makespan Comparison for Montage_1000

Figure 2 is demonstrating the results of makespan found after experiments with Montage_1000 dataset. This figure is displaying Virtual Machines at x-axis and Makespan atthe y-axis. The Montage_1000 dataset encompasses 1000 medium and small-sized tasks. With 6, 12, 24 and 48 VMs, Max-Min performs slightly better than all algorithms. But with 3 VMs, Max-Min is comparable with FCFS because the amount of medium and small tasks is higher and VMs are homogeneous in this experiment.

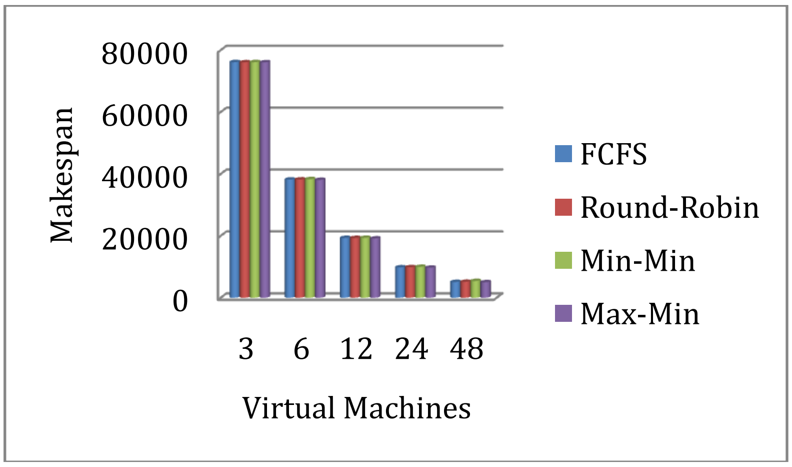


Figure 3. Makespan Comparison for Inspiral_1000

Figure 3 is indicating the results of Makespan with respect to the Inspiral_1000 dataset. Thispicture is representing VMs at x-axis and Makespan atthe y-axis. The dataset Inspiral_1000 contains 1000 large, medium and small length tasks. These tasks are almost in the same amount and nosevere difference is present among these tasks length. Max-Min performs better thanother algorithms in overall circumstances because of the homogeneous environment and faster machines executed large-sized tasks with medium tasks concurrently.

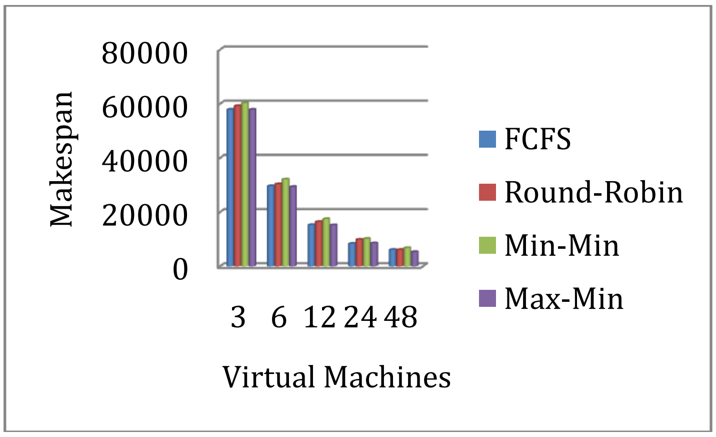


Figure 4. Makespan Comparison for Sipht_1000

Figure 4 is presentation the makespan comparison for Sipht_1000 dataset.The graph is demonstrating Virtual Machines at x-axis and Makespan atthe y-axis. The Sipht_1000 dataset contains 1000 tasks. Among them, maximum numbers of tasks are small in size with few extra-large sized tasks and medium-sized tasks as well. Here, Max-Min out-performs than other algorithms on all VMs except a set of 24 VMs due to the scheduling of faster resources to large tasks first and then, the smaller tasks are executed synchronously with medium length tasks. In the case of 24 VMs, the results of Max-Min and FCFS are comparable due to few extra-large tasks are performed on homogeneous virtual machines.

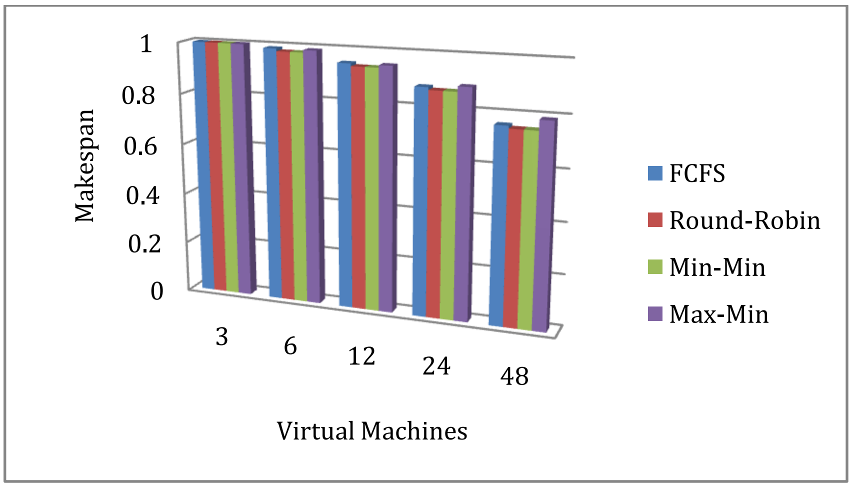
With these results, it is found that Max-Min is having lower makespanin case ofall scenarios.

Table 4. Average Resource Utilization Rate

Scenarios	VMs	FCFS	Round-Robin	Min-Min	Max-Min
Scenario - 1 CyberShake_1000	3	**1.00**	**1.00**	**1.00**	**1.00**
	6	**0.99**	0.98	0.98	**0.99**
	12	**0.95**	0.94	0.94	**0.95**
	24	0.88	0.87	0.87	**0.89**
	48	0.76	0.75	0.75	**0.79**
Scenario - 2 Montage_1000	3	**0.94**	**0.94**	**0.94**	**0.94**
	6	**0.87**	**0.87**	**0.87**	**0.87**
	12	**0.75**	**0.75**	**0.75**	**0.75**
	24	**0.59**	**0.59**	0.58	**0.59**
	48	**0.41**	**0.41**	0.40	**0.41**
Scenario - 3 Inspiral_1000	3	**1.00**	**1.00**	**1.00**	**1.00**
	6	0.99	0.99	0.99	**1.00**
	12	0.98	0.98	0.98	**0.99**
	24	0.97	0.96	0.95	**0.98**
	48	0.94	0.92	0.88	**0.95**
Scenario - 4 Sipht_1000	3	1.00	0.98	0.96	**1.00**
	6	0.98	0.95	0.90	**0.99**
	12	0.95	0.88	0.83	**0.95**
	24	0.86	0.73	0.71	**0.85**
	48	0.59	0.59	0.54	**0.68**

For each scenario, the comparison of resource utilization is illustrated in table 4 with respect to a set of3, 6, 12, 24 and 48 VMs. Max-Min out-performs at maximum levels for all datasets because it distributes large tasks to faster virtual machines at first priority and small tasks are scheduled concurrently.

Figure 5. Average Resource Utilization for CyberShake_1000



For scenario 1, figure 5 is indicating the comparison of average resource utilizationusing CyberShake_1000 dataset. Virtual Machines are represented at x-axis and Makespan is atthe y-axis in this figure. This graph is indicating that the rate of resource utilization is decreased with the number of VMs increments because the same work is divided amongvariousvirtual machines. The graph is also showing that Max-Min, FCFS, and Round-Robin algorithms utilized resources equally when VMs are 3whereas Max-Min and FCFS are performing equallyon 6 and 12 VMs. This is because the makespan of FCFS is slightly higher than Max-Min when numbers of resources are less as described earlier in table 3. Max-Min keeps busy maximum resources at a higher rate while working with 24 and 48 VMsbecause the makespan of Max-Min is lower while working with these machines.

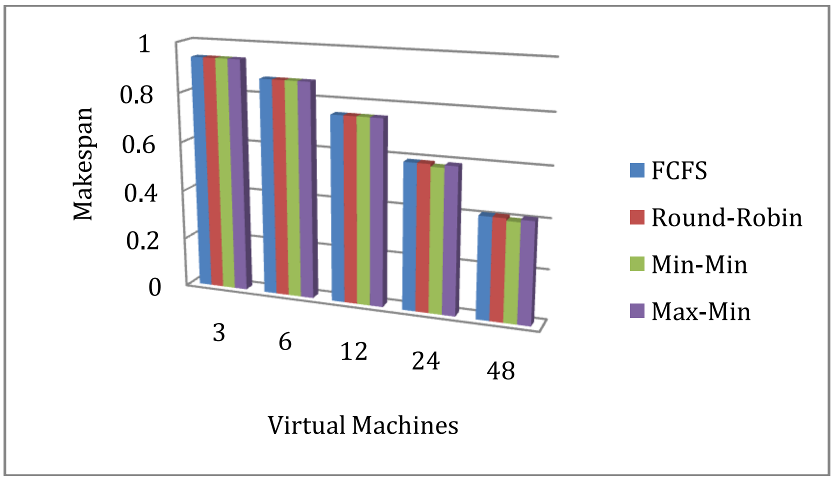


Figure 6. Average Resource Utilization for Montage_1000

For scenario 2, figure 6 is demonstrating the comparison of average resource utilization using Montage_1000 dataset. The graph is showing Virtual Machines at x-axis and Makespan atthe y-axis. The resource utilization is decreasing with the number of VMs increment because the same work is divided between multiple resourcesas described earlier too. This figure is representing that Max-Min FCFS and Round-Robin algorithms are using resources in a comparable rate in the entire scenario becausethe makespansof these three algorithmsare comparable for task completion. Min-Min also performs comparably with Max-Min while working with 3, 6 and 12 VMs.

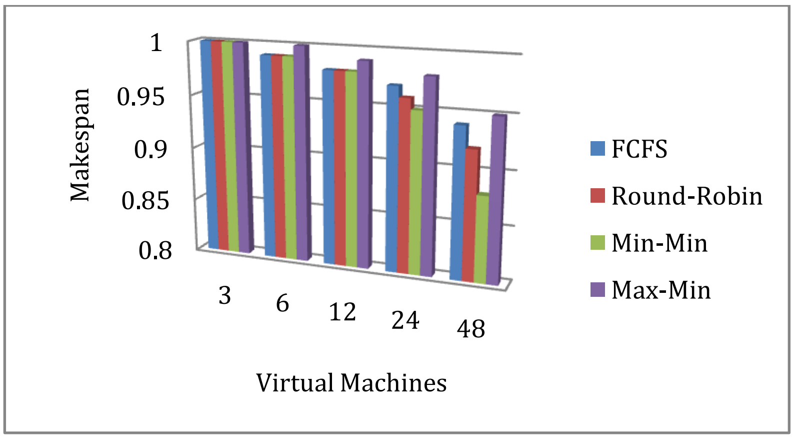


Figure 7. Average Resource Utilization for Inspiral_1000

For scenario 3, figure 7 is representing the comparison of average resource utilization using Inspiral_1000 dataset. The graph is showing Virtual Machinesat x-axis and Makespanatthe y-axis. The resource utilization is getting decreased with the growth of VMs because the same work is divided amongvariousmachines. This figure is alsodemonstrating that Max-Min, FCFS, Round-Robin,and Min-Min algorithms are using resources ata comparable rate while working with3VMs. This is because the comparable makespans have beenused by these all algorithms in order to execute the all tasks as shown in table 3 too.In rest 6, 12, 24 and 48 VMs, Max-Min algorithm is consuming resources at a higher rate because of its lower makespans taken for completion of all tasks.

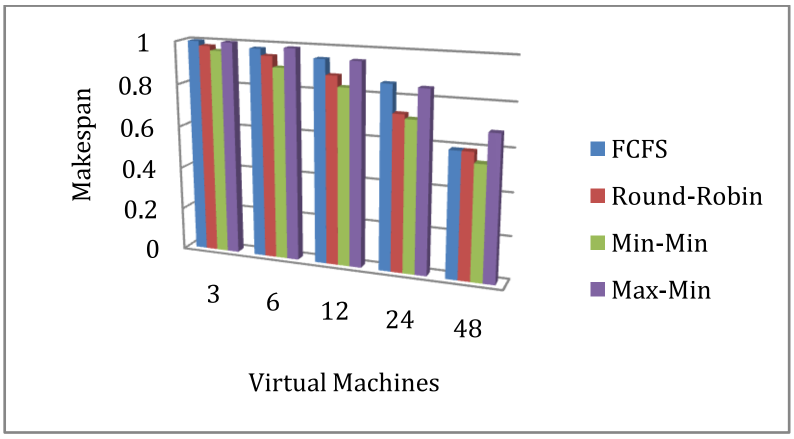


Figure 8. Average Resource Utilization for Sipht_1000

For scenarios 4, figure 8 is showing the comparison of average resource utilization using Sipht_1000 dataset. In this picture, Virtual Machines and Makespan are represented at x-axis and the y-axis respectively. One can see that the resource utilization is decreasing with the number of VMs increments. This is because the same work is divided among various virtual machines, described earlier too. Figure 8 is also indicating that Max-Min is utilizing maximum resources at a higher rate than other algorithms due to its lower makespan while completing all requested tasks.

Inthe end, it is concluded that Max-Min is utilizing the resources with higher rates for all scenarios at maximum situations.

Table 4. Evaluation of Resources Cost

Scenarios	VMs	FCFS	Round-Robin	Min-Min	Max-Min
Scenario - 1 CyberShake_1000	3	99321.26	99318.61	**99318.35**	99324.77
	6	99478.47	99473.58	**99472.96**	99484.54
	12	99785.88	99781.62	**99780.57**	99795.51
	24	100402.78	**100398.21**	100399.08	100407.8
	48	101637.51	101631.18	101592.78	**101586.93**
Scenario - 2 Montage_1000	3	51854.53	51847.61	50343.82	**50342.56**
	6	51977.25	51970.68	**50429.65**	50620.74
	12	52065.4	52090.06	**50499.06**	50620.74
	24	52176.62	52179.92	50689.36	**50682.85**

	48	52132.95	52158.02	**50148.35**	50643.45
Scenario - 3 Inspiral_1000	3	980531.69	981404.36	**980279.27**	980298.67
	6	**979771.89**	980017.38	982103.49	981343.34
	12	981725.81	981032.45	983037.24	**980545.06**
	24	986446.35	987027.86	**982953.54**	984591.05
	48	988760.49	986678.59	**982668.36**	985910.18
Scenario - 4 Sipht_1000	3	744953.62	**744650.7**	744678.26	744935.93
	6	752502.77	748690.2	745902.72	**744942.14**
	12	754374.89	752796.64	**749400.09**	752904.43
	24	763259.16	758866.94	**737984.44**	746799.13
	48	771979.94	783608.57	**711300.16**	718291.63

Table 4 is representing the comparison of the cost of resources used in the simulation for all scenarios as per the cost plan which is given in table 2.

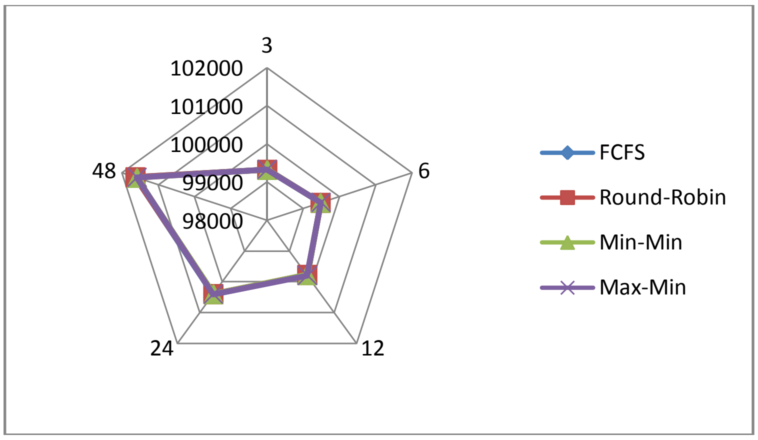


Figure 9. Cost Utilization for CyberShake_1000

Figure 9 is depicting the cost comparison of resources used for CyberShake_1000 dataset in scenario 1. The radar graph's edges are representing 3, 6, 12, 24 and 48 VMs and radar values are showingthe cost of resources. Smaller the radar web lesser the cost. The graph is indicating that Max-Min and Min-Min algorithms consuming the comparable costs in cases of 3, 6 and 12 VMs because of their comparableresource utilization rate. While working with 24 virtual machines Max-Min and Round-Robin algorithms are consuming almost equal cost because of their comparable resource utilization rate. On 48 VMs, Max-Min is having lesser cost because of its efficient resource usages.

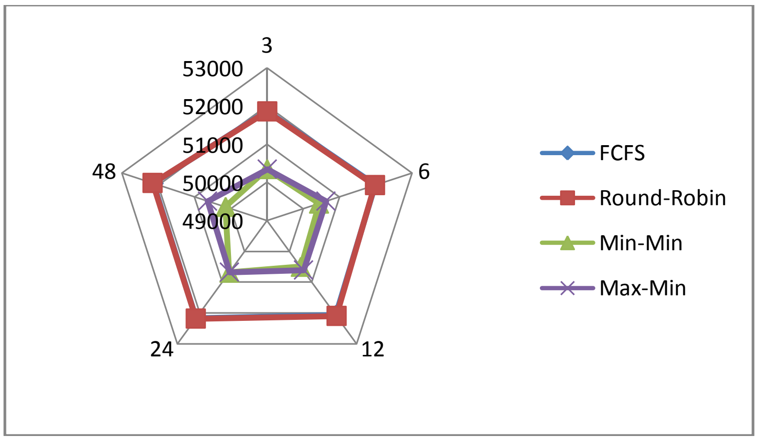


Figure 10. Cost Utilized for Montage_1000

Figure 10 is representing the cost comparison of resources used for Montage_1000 dataset in scenario 2. The radar graph's edges are showing 3, 6, 12, 24 and 48 VMs and radar values are indicatingthe cost of resources. Smaller the radar web lesser the cost. The figure indicates that Max-Min is consuming lesser cost than all algorithms while working on 3 and 24 VMs because of its efficient resources usage. In the case of 6, 12 and 48 VMs; Max-Min and Min-Min algorithms are consuming comparable costs because of their comparable resource utilization rates.

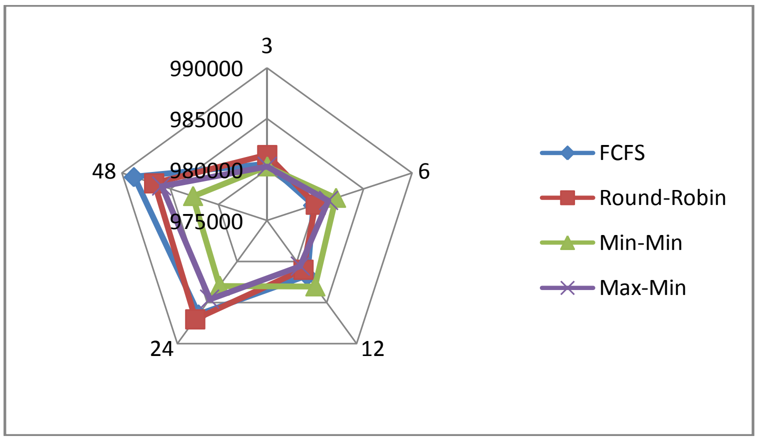


Figure 11. Cost Utilized for Inspiral_1000

Figure 11 is presenting the cost comparison of resources used for Inspiral_1000 dataset in scenario 3. The radar graph's edges are representing 3, 6, 12, 24 and 48 VMs and radar values are expressingthecost of resources. Smaller the radar web lesser the cost. When working on 3, 24 and 48 VMs, the cost is comparable for Max-Min and Min-Min algorithms because of their comparable resource utilization rates. The cost of FCFS is slightly lower than Max-Min because of comparable resource utilization of both algorithmwhile working on 6 VMs. When working with 12 VMs, the cost of Max-Min is lower because Max-Min is having an efficient rate of resource utilization at this stage.

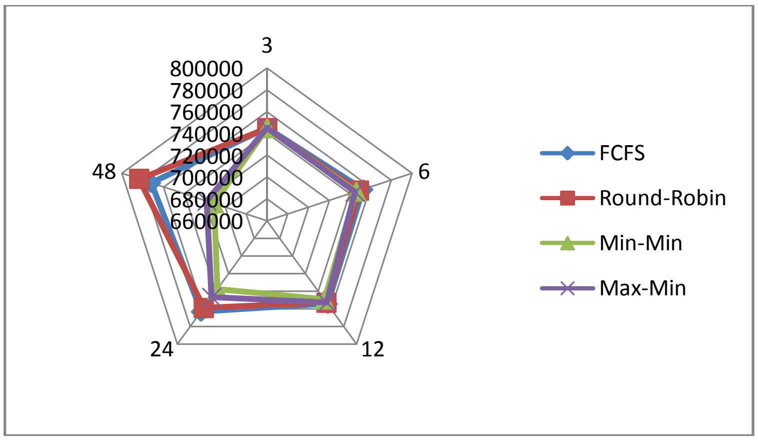


Figure 12. Cost Utilization for Sipht_1000

Figure 12 is demonstrating the cost comparison of resources used for Sipht_1000 dataset in scenario 4. The radar graph's edges are indicating 3, 6, 12, 24 and 48 VMs and radar values are depicting the cost of resources. This figure also tells that smaller the radar web lesser the cost. While working on 3 VMs, the cost of the Round-Robin algorithm is comparable with Max-Min and Min-Min because of the comparable resource utilization rate of these algorithms. For 12, 24 and 48 VMs, the cost of Max-Min and Min-Min algorithms are comparable due to the comparable resource utilization rate of both algorithms at these stages. For 6 VMs, the cost of Max-Min is lesser than all algorithms because of its better resource utilization rate.

With these outcomes, it found that Max-Min algorithm out-performs than FCFS, Round-Robin and Min-Min algorithms with consuming either less or almost comparable costs and having better makespan and resource utilization ratefor maximum cases. So, Max-Min algorithm beats all other algorithms in an overall ratio in all scenarios.

5. Conclusion and Future Prospects

Cloud computing is becoming more and more important computing facility among various organizations due to its flexibility i.e. giving all computing services without demanding a specific area or place. So the industries and other organizations areswitching for their computing work on the cloud; as a result,cloud load is increasingrapidly. In order to manage this load, a virtual machine or task scheduling technique is required. In the field of task scheduling, various algorithms have been developed so far. This paper is giving focus on four major algorithms' performance evaluation for task scheduling in a homogeneous environment. The four major algorithms which are used for simulation includes Max-Min, Min-Min, FCFS, and Round-Robin. Four scenarios have been designed with the help of CybeShake_1000, Montage_1000, Inspiral_1000 and Sipht_1000 datasets in WorkflowSim tool which is written in Java. Total 5 sets of VMs were created. The first set is having 3 VMs, the second set is having 6, the third set is containing 12, and fourth and fifth sets are having 24 and 48 VMs respectively. All machines are having homogeneous characteristics.

For CyberShake_1000, Max-Min algorithm out-performed for makespan and resource utilization with consuming comparable costs than other algorithms specially Min-Min. In the case of Montage_1000, Max-Min performed well than other algorithms for makespan and consumed comparable resourceswith FCFS and Round-Robin algorithms. For cost consideration, Max-Min consumed either less cost than all other algorithms or comparable with Min-Min. For Inspiral_1000 dataset, Max-Min performed well than other algorithms in terms of makespan and resource utilization. For cost consideration, it is comparable with Min-Min. In the case of Sipht_1000 dataset, Max-Min algorithm performed well for makespan and resource utilization.With cost consuming factor, it is comparable with Min-Min in few cases and better than all other algorithms in rest situations.

It isconcluded that Max-Min is out-performing than other algorithms in terms of makespan, resource utilization with minimum or comparable consuming costs. In the future, a new algorithm can be introduced by merging two or more scheduling algorithms. Soft computing techniques may alsobe a good choice for performance optimization.

References

[1] S. Elsherbiny, E. Eldaydamony, M. Alrahmawy and A. E. Reyad, "An Extended Intelligent Water Drops Algorithm for Workflow Scheduling in Cloud Computing Environment," Egyptian Informatics Journal, Article in Press, 2017.
[2] R. Garg and A. K. Singh, "Adaptive Workflow Scheduling in Grid Computing Based on Dynamic Resource Availability," Engineering Science and Technology an International Journal, Vol. 18, pp. 256-269, 2015.
[3] Chen W-N, J. Zhang,"An ant colony optimization approach to a grid workflowscheduling problem with various QoS requirements," IEEE Transactions Vol. 39, Issue 1, pp. 29–43, 2009.
[4] S. Pandey, L. Wu, S. M. Guru and R. Buyya, "A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments,"International Conference on Advanced Information Networking and Applications (AINA), IEEE, 2010.
[5] Hu X-H et al. "An IPSO Algorithm for Grid Task Scheduling Based on Satisfaction Rate,"International Conference on Intelligent Human-Machine Systems and Cybernetics, IEEE, 2009.
[6] S. Brar and S. Rao,"Optimizing Workflow Scheduling using Max-Min Algorithm in Cloud Environment,"International Journal of Computer Applications, Vol. 124, Issue 4, pp. 44–49, 2015.
[7] S. Bilgaiyan, S. Sagnika and M. Das,"Enhanced Max-Min Task Scheduling Algorithm in Cloud Computing,"InternationalJournal of Application or Innovation in Engineering and Management, Vol. 2, Issue 4, pp. 259-264, 2013..
[8] K. N. Baxodirjonovich and T. Choe, "Dynamic Task Scheduling Algorithm based on Ant Colony Scheme,"Internation Journal of Engineering and Technology, Vol. 7, Issue 4, pp. 1163–1172, 2015.
[9] D.I. George Amalarethinam and V.S. Kfatheen, "Max-min Average Algorithm for Scheduling Tasks in Grid Computing Systems,"International Journal of Computer Science and Information Technologies, Vol. 3, Issue 2, pp. 3659-62, 2014.
[10] S. Devipriya S and C. Ramesh,"Improved Max-Min Heuristic Model for Task Scheduling in Cloud,"IEEE, pp. 883-888, 2013.
[11] W. Li, and W. Zhang, "An improved Scheduling Algorithm for Grid Tasks,"International Symposium on Intelligent Ubiquitous Computing and Education, Vol. 35, pp. 9-12, 2009.
[12] S. Parsa, and E.M. Reza, "RASA: A New Task Scheduling Algorithm in Grid Environment,"World Applied Sciences Journal,Vol. 7, pp. 152-155, 2009.
[13] K. Gupta, and M. Singh, "Heuristic Based Task Scheduling In Grid,"International Journal of Engineering and Technology, Vol. 4, pp. 254 – 258, 2012.
[14] S. Anousha, A. Shoeib, and M. Ahmadi,"A New Heuristic Algorithm for Improving Total Completion Time in Grid Computing,"Springer-Verlag Berlin Heidelberg, pp. 17-26, 2014.
[15] M-Y Cheng and D. Prayogo, "Symbiotic organisms search: A New Metaheuristic Optimization Algorithm," Computer and Structures, Vol. 139, pp. 98–112, 2014.
[16] J. K. Konaang, F. H. Ayob and A. Muhammed, "An Optimized Max-Min Scheduling Algorithm in Cloud Computing," Journal of Theoretical and Applied Information Technology, Vol. 95, Issue 9, pp. 1916-1926, 2017.
[17] J. Y. Maipan-uku, A. Muhammed, A. Abdullah and M. Hussin, "Max-Average: An Extended Max-Min Scheduling Algorithm for Grid Computing Environment," Journal of Telecommunication, Electric and Computer Engineering, Vol. 8, Issue 6, pp. 43-47, 2016.

[18] S. Rekha and R. Santosh Kumar, "Priority Based Job Scheduling for Heterogeneous Cloud Environment," International Journal of Computer Science Issues, Vol. 11, Issue 3, pp. 114-119, 2014.

[19] J. Bhagwan and S. Kumar, "An Intense Review of Task Scheduling Algorithms in Cloud Computing," International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, Issue 11, 2016.

[20] K. Etminani, M. Naghibzadeh and N. R. Yanehsari, "A Hybrid Min-Min Max-Min Algorithm with Improved Performance," Department of Computer Engineering, Ferdowsi University of Mashad, Iran.

[21] N. Moganarangan, R.G. Babukarthik, S. Bhuvaneswari, M.S. SaleenBasha and P. Dhavachelvan, "A novel algorithm for reducing energy-consumption in cloud computing environment: Web service computing approach", Journal of King Saud University – Computer Science and Information Sciences, Vol. 28, pp. 55-67, 2016.