JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

STUDY OF SOME ALGORITHMS FOR TASK SCHEDULING IN IaaS CLOUD COMPUTING ENVIRONMENT – PERFORMANCE AND EFFICIENCY

Rajendra T. Kaple
Programmer,
B N College of Engineering, Pusad, India

Abstract: Important outcomes of the cloud computing environment are cost saving, energy efficiency, flexibility, high accessibility, rapid implementation and scalability. In the computing domain, there are various kinds of regular practices being adopted based on technological innovations. The computing has various paradigms including the High-Performance Computing (HPC), Parallel Computing, Distributed Computing, Cluster Computing, Grid Computing, Cloud Computing, Mobile Computing, Quantum Computing, Fog Computing, Bio Computing, Optical Computing, Nano computing. As computing systems become more capable and faster, it demands the quality of modern computing, optimum scheduling and high security.

Index Terms - Scheduling, Algorithms, Cloud Computing, Heuristics, QoS, DoS, Security.

I. INTRODUCTION

Important outcomes of the cloud computing environment are cost saving, energy efficiency, flexibility, high accessibility, rapid implementation and scalability [1][2][3].

In the computing domain, there are various kinds of regular practices being adopted based on technological innovations. The computing has various paradigms including the High Performance Computing (HPC), Parallel Computing, Distributed Computing, Cluster Computing, Grid Computing, Cloud Computing, Mobile Computing, Quantum Computing, Fog Computing, Bio Computing, Optical Computing, Nano computing. As computing systems become more capable and faster, it demands the quality of modern computing, optimum scheduling and highly security [4][5][6].

Task scheduling algorithms have a direct impact on the proficiency of user tasks and also in professional utilization of resources in IaaS cloud computing environment. Hence, how to realize the optimum distribution of user tasks is still an unsolved question for task scheduling in this environment, as shown in Figure 1. The algorithm of task scheduling as included in the field of cloud computing is as follows:

Initially, resources and tasks are mapped regarding the existing task and information of resources in accordance with the basic methods.

At this point, tasks are mapped among the quality of service (QoS) requirements of cloud users and the resources are distributed to the application of the task to validate the competence of the task. In conclusion, the summary of the consequences are implemented by submitting the users' demand [7-8].

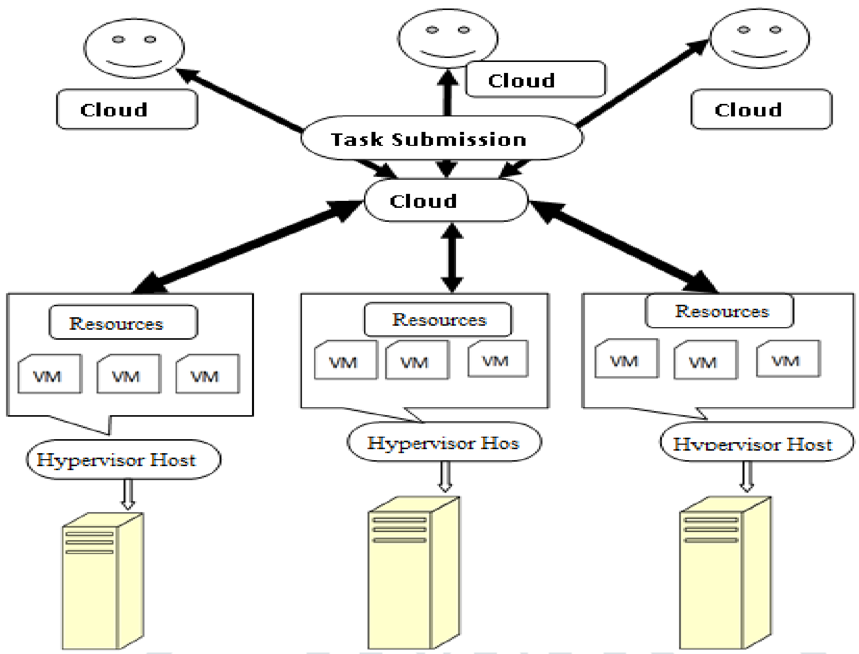


Figure 1: Task scheduling in IaaS cloud computing.

Optimal task scheduling in cloud computing environment is known to be an NP-complete problem [8, 9]. Existing heuristic algorithms for tasks scheduling are not easy to compare due to the divergent underlying assumption by each heuristic algorithm.
The word heuristic is used for algorithms which discover solution among all probable ones, but they do not guarantee that the best will be found, therefore they may be considered as approximate and not perfect algorithms. These algorithms generally get a solution close to the best one and they find it quick and readily. The procedure used from a heuristic algorithm is one of the identified methods, such as greediness, but these algorithms ignore some of the parameters.
Heuristic is a method intended for solving a problem rapidly when traditional methods are too time-consuming, or for finding an fairly accurate solution when conventional technique fail to yield an exact answer.
Cloud computing infrastructure is suitable for computational needs of large task sizes. Best possible scheduling of tasks in cloud computing environment is an NP-complete problem; hence the appliance of heuristic method is required. Many heuristic algorithms are developed and used in solving this problem but selecting the suitable algorithm for solving task assignment problem of a particular nature is difficult since the methods are developed under diverse assumptions. To deal with the autonomous tasks in homogeneous and heterogeneous environments, six heuristic algorithms are considered, for comparing their performance in terms of cost, degree of imbalance, makespan and throughput. They are, First Come First Serve (FCFS), Minimum Completion Time (MCT), Minimum Execution Time (MET), Max-min, Min-min and Sufferage algorithms.
Independent tasks are used for scheduling, which is done off-line, that is the execution time of the tasks are known in advance (knowledge that comes from the power of reasoning based on obvious fact). The metrics of performance comparison considered are cost, degree of imbalance, makespan and throughput. Tasks are executed on Virtual Machines (VM) in order of their arrival time and only one task is applied on a VM at a time and pre- emption is not allowed. The number of tasks and VMs are known in advance.
Intention of this work is to study a basis for evaluation and insight into situation, where one scheduling heuristic will put into action better than the other. The main scheme here is, to explore heuristic algorithms for task scheduling and outline a deviation among them so as to finish at a conclusion about the most excellent available heuristic algorithm for cloud environment. Rests of the sections are ordered as follows:
In the subsequent part, we assess study of task scheduling in the area of IaaS cloud computing. We authenticate the description of rule-based scheduling heuristic algorithms in the methodology section. Results and discussion point up study of performance evaluation of heuristic algorithms with the help of investigational simulation. The last section consists of the conclusion, recommendation and future works.

II. WHY SCHEDULING IS NEEDED IN CLOUD COMPUTING

One of the key rewards of switching to the clouds is the scalability ability of the applications. Contrary to the grids, the ability to scale the cloud resources allows their real-time provisioning so as to meet the application constraints. Cloud services would be accessible at a much lower cost. Typically, tasks are scheduled according to the user requirements. New scheduling policies are required to overcome several complications put forward by the network properties between the consumer and the resources. Currently, scheduling policies work at combining conventional scheduling policies as well as network aware policies to propose an improved and much more effective job scheduling framework. Formerly, scheduling algorithms were executed in the grids but due to decreased performance, there was a need to realistic scheduling algorithms in the cloud itself. It would ease the workflow management system and resolve the Quality of Service (QoS) issue whereas the conventional approach still requires the advance resource reservation. Cloud services are available at a lower cost, resources like bandwidth, and storage. It has to be noted that cloud applications usually demand heterogeneous and complex execution environments. In the case of cloud computing, a user need to handle a number of virtualized resources; hence it is not possible to assign jobs manually. Such environments are hard to fabricate on the grid resources. Scheduling is a vital feature in the field of cloud computing and need to bring about competent assigning of the resources.

As a user of latest technology and a person with mathematical back ground, the technological need and further scope of research (Mathematical Modeling) in the area of task scheduling in IaaS cloud computing environment attracted my attention. Therefore, for my research work, I decided to focus on the work carried out by, Professor, Side Humid Husain Mani, Faculty of Computing, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia, on Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment.

A lot of features like user needs, type of system etc. have to be considered while designing a scheduling algorithm. Avoid indefinite starvation i.e. a process must not wait for an indefinite period during the process service. Minimizing overhead as overhead causes wastage of the resources and if overhead is reduced, the overall performance of the system gets better. Imposing the priorities i.e. if a system allot any priorities to the job, the algorithm must process the job with mentioned priority first. Accomplishing sense of balance between response and utilization, so that, all the resources of the system are eventful. Depending on the type of system, a user can expect the following things from the scheduler.

(a) **Quality of Service:** The main aim of the cloud is to offer computing and storage services to the user, and meet their resource demand. The performance of the resources supplied by the provider is judged through quality of service.

(b) **Load Balance:** Job scheduling and load balancing are directly associated in the cloud computing environment. Task scheduling is responsible for proper matching of the tasks and the resources. Load balancing states the level two loads in the task scheduling - virtual machine load stage and resource layer load stage.

(c) **Best running time:** Tasks are separated into various categories according to the user need and then, a finest running time is set according to different goal of each task. It ultimately improves the QoS.

(d) **Economic Principles:** Cloud computing resources are broadly distributed among diverse organizations all around the world. Each organization has its individual management policies which are very much affected by the location of the resources as well. As a business mode, cloud computing offer related services using these resources according to unlike user requirements, making demand charges very realistic. The market economy is the key driver for task scheduling as well as resource management and we must make sure that both - consumer and provider must be benefited from the cloud services so that the technology as a whole can move ahead.

The performance of scheduling is associated with several parameters:
1. CPU Usage: CPU should be kept active at 100% of time.
2. Throughput: Number of process that typically ends executing in the given moment of time.
3. Turnaround time: Time needed for the execution of a process.
4. Waiting time: It is the time that a process must wait in queue ready to be executed.
5. Response time: This is the time between the receptions of the request made to the first response.
Let us consider three processes P1, P2 and P3 with the duration, order and arrival time as follows:

Process	Duration	Order	Arrival Time
P1	20	1	0
P2	7	2	0
P3	4	3	0

Gantt chart for this is as follows.

P1(20)	P2(7)	P3(4)

0 20 27 31

As we see from the Gantt chart:

For P1, waiting time is 0, for P2 waiting time is 20 and for P3 waiting time is 27.

Therefore, the average waiting time = (0+20+7)/3 = 9.

IIi. RELATED WORKS

In this part, we review current studies which use a variety of heuristic, meta-heuristic and hybrid algorithms for task scheduling in IaaS cloud computing system.

Abdullahi, et al. [11] put forth a Discrete Symbiotic Organism Search (DSOS) algorithm for an ideal schedule of tasks on resources in cloud computing system. Investigational result shows that the DSOS perform better than Particle Swarm Optimization (PSO) in term of convergence rate. Further, Abdullahi and Ngadi et.al. [12], also present a hybrid Simulated

Annealing (SA) and Symbiotic Organisms Search (SOS) algorithm called SASOS to attain optimum scheduling of tasks in cloud computing. The outcome confirm that the suggested algorithm make better than DSOS to achieve better convergence ratio and quality of results.

Bansal, et al. [13] review the parameters for cost and load balancing by Virtual Machine Tree (VMT) enhanced task scheduling algorithm and recognized that the parameter for cost is not so effective with considered algorithm.

Thomas, et.al.[14], recommend a Min-Min algorithm that takes into consideration both cloud users requirement and resource accessibility. Proposed algorithm decreases makespan of the tasks by analyzing task size.

To diminish the makespan in IaaS cloud, Abdul Humid, et.al. [15], proposed a League Championship Algorithm (LCA) for the efficient task scheduling in IaaS cloud computing system.

Xue S. et.al. [16], ACO-LB algorithm resourcefully gathers the appropriate resources at job concluding point and assistances in resource allocation in a peer group.

Abdul Humid, et.al. [17] propose Global League Championship Algorithm (GBLCA) algorithm to solve the non-deterministic problem of safe scheduling of tasks by reducing the makespan and response time.

In cloud computing, vital features of task scheduling are discussed by Liu C-Y et.al. [18], which spotlight on an algorithm for task scheduling that is intended, based on genetic-Ant Colony Algorithm (GACO). The advantage is having a flexible enthusiastic reaction of ACO algorithm, and compelling into interpretation the convergence ratio of the algorithm.

Hung, et al. [19], recommends a method for task scheduling, while keeping in view the clashes connected with operating cost and network of cloud in order to reduce the recovery time for the upgrading of steadiness.

A better form of task scheduling in cloud computing is proposed by Zhao, et. al. [20]; which takes the intelligence firefly algorithm into account. With the behavior of firefly algorithm, the cloud computing research demonstrates the extreme resolution for task scheduling.

With the help of fuzzy clustering, Li, et al. [21] suggested an algorithm and model to distribute suitable resources to tasks mapping. It achieves the needs of tasks and reserve for the dominant resources.

Wu, et al. [22], put forward a task scheduling QoS driven algorithm based on MCT algorithm in cloud computing. Task Scheduling QoS (TS_QoS) algorithm compute the priority of task discussing to the appearances and at that point organizes the tasks with respect to their priority order.

For optimizing task scheduling Gabi, et al. [23], propose Orthogonal Taguchi-based Cat Swarm Optimization (OTB-CSO) hybrid algorithm to reduce the delay in total task execution. The idea is to reduce the makespan and degree of imbalance for all schedule tasks on VMs.

For reducing the flaw of the cloud computing data center in resource management, to confirm that cloud computing provides superior QoS service. Ant colony optimization (ACO) is applied in the model of cloud computing to manage the resource and schedule regarding to the actual QoS parameters necessary for the cloud computing [24].

An effective VMs allocation algorithm and job scheduling policies directly effect on the transaction between cloud providers and users. For this purpose, Cao, et al. [25] compare the various job scheduling policies including FCFS, SJNF, SJEF, LJNF and LJEF for resource utilization and cost optimization by using Python-based simulation package SimPy. Further, He, et al [26] introduces and compares the five heuristic algorithms to evaluate the performance of CloudSim tool. Sequence Scheduling (SS), FCFS, Shortest Task First (STF), Balance Scheduling (BS) and Greedy Scheduling (GS) algorithms are used to solve the issue of task scheduling in cloud computing.

Patel G, et al. [27] reviewed heuristic algorithms for the static task scheduling in cloud computing, consist of Opportunistic Load Balancing (OLB), MCT, MCT, Max-min, Min-min and Load Balancing Min-min (LBMM) and proposed Enhanced (LBMM) algorithm for static task scheduling in cloud computing. Moreover, detailed studies of several task scheduling algorithms are presented for the cloud computing by [28].

These algorithms are FCFS, RR, OLB, Min-min, Max-min, GA, SA, Switching Algorithm, Sufferage, etc. Also, a brief study of many scheduling parameters is discussed including the makespan, deadline, execution time, completion time, energy, performance, QoS and load balancing for task scheduling in cloud computing.

Akilandeswari and Srimathi et.al. [29], present the comparative analysis of static and dynamic task scheduling algorithms used by cloud providers in cloud computing. For static task scheduling FCFS, RR, Min-min and Max-min algorithms, while for the dynamic task scheduling ACO, GA, PSO and SA are proposed for implementation.

Likewise, Thaman, et al. [30] proposed categorization for task and job scheduling meta-heuristic and heuristic algorithms. These, categorization are based on the objective and constraint oriented task scheduling algorithms.

Tabak, et al. [31]; present an algorithmic improvement that asymptotically decreases the execution time of Min-min algorithm without disturbing the quality of service. Also, the recently projected Min-min algorithm is combined with Max-min and Sufferage algorithm, to obtain two hybrid algorithms. The outcome of hybrid algorithms is to reduce the disadvantage of Max-min in resolving problematic instances with highly skewed cost circulations and also enhance the execution time results of Max-min algorithm.

IV. SCHEDULING HEURISTICS (RULE BASED)

In this section, the rule based scheduling heuristics algorithms are considered to lay down the base for task scheduling and talk about their functioning in IaaS cloud computing system.

In cloud computing, heuristic algorithms are proposed to resolve the problematic issues quicker than meta-heuristic algorithms, when their performance is too lethargic. Also, heuristic algorithms are used to acquire an optimum solution, when meta-heuristic algorithms are failed to discover the accurate or optimal solution. These are accomplished by accuracy, completeness, optimal transaction or speed [32-33].

(1) First Come First Serve (FCFS)

FCFS algorithm is recognized to schedule and manage processes that automatically executes tasks or resource and precedes them by the order of their arrival demand of users. With FCFS algorithm, first arrival demand of task or resource is satisfied first and then next demand in a queue will be executed once the prior is complete. It is also based on the FIFO algorithm.

It offers efficient, error-free and simple procedure for scheduling by reducing the VMs or resources in cloud computing. CloudSim, iFogSim CEPSim and GridSim [34-35] simulators used FCFS algorithm by default for the scheduling purpose of the tasks and resources in cloud and grid environment.

Abdul Humid, et.al. [15], compare the LCA (League Championship Algorithm) with three other existing algorithms including the FCFS, Best Effort First (BEF) and Last Job First (LJF) to estimate the performance of suggested LCA task scheduling algorithm by reducing the makespan time.

Further, Jamali, et. al. [36], compare the Particle Swarm Optimization, Genetic Algorithm and FCFS algorithms for minimizing the makespan, waiting time and enhancing the performance of given tasks sets.

Lakra and Yadav et.al. [37], evaluate the multi objective task scheduling algorithm with FCFS and priority scheduling algorithm to cut the throughput for task scheduling.

Moreover, Zoo, et al. [38], detailed a multi objective ACO algorithm for enhancing the cost, makespan, resource utilization and time deadline for the task scheduling and contrasts the results with FCFS and Min-min algorithm.

Raju, et al. [39], suggest Deadline Aware Two Stage Scheduling and evaluate the metrics of average turnaround time, average waiting time and violation in deadlines to schedule VMs by comparing with FCFS and Shortest Job First (SJF) algorithm.

Moreover, Mondal, et al. [40], evaluate the Stochastic Hill Climbing technique with FCFS and round robin (RR) for load balancing.

Dasgupta, et al. [41], compare the Greedy Algorithm with FCFS, RR and Stochastic Hill Climbing algorithms for load balancing in IaaS cloud computing environment for task scheduling.

Mukherjee and Sindhu, et.al.[42], propose, Longest Cloudlet Fastest Processing Element (LCFP) and Shortest Cloudlet Fastest Processing Element (SCFP) for scheduling jobs in a private cloud in order maintain the lowest makespan time. Moreover, FCFS is used in the study for the comparison of the performance of the algorithms in the simulation.

Further, they present a bi-objective GA based on scheduler for resource scheduling that improves the makespan and resource utilization as evaluation with FCFS and RR algorithms.

Similarly, Tewfik, et.al. [43], advised a task scheduling policy based on ACO algorithm for optimum task scheduling and show the improved makespan as comparison with FCFS and RR algorithms. Minimum Completion Time (MCT) algorithm assign tasks to VMs or resources based on the finest expected completion time for that task in arbitrary order. Each task is allocated to the VM or resource that has most basic completion time. With MCT algorithm, some tasks are assigned to the VMs or resources having no minimum execution time. It tries to merge the advantages of OLB and MET algorithms while avoiding their disadvantage [44]. Fig 2 shows the pseudo-code for MCT algorithm.

Du Kim and Kim et.al.[44], advise an innovative scheduling algorithm MECT consist of MET algorithm and MCT algorithm for on-line scheduling in heterogeneous computing systems. MECT demonstrate better performance than the basic MET algorithm and MCT algorithm for dropping makespan. Minimum Execution Time (MET) Minimum Execution Time (MET) algorithm allot tasks to VMs or resources based on the best expected completion time for that task without regard to resource availability. The core evaluation of rule-based algorithms for scheduling in cloud computing environment idea of MET is to assign a task to VM or resource based on minimum execution time, which sometimes result to high load imbalance, since the assignment is not reliant on the availability.

(2) Max-min

Similar to the Min-min algorithm, after determining the completion times for each task on all machines, the task with maximum completion time is scheduled on the reliable machine in the case of max-min and the process is repetitive until all the tasks are scheduled. In Min-min algorithm, the expectation is that if more tasks are scheduled on machines that execute them earliest and fastest, lesser makespan will be attained. Max-min algorithm is usually used in the circumstances where there are less long and shorter tasks. As well as, it can reduce starvation for the longer tasks, since it will allow the longer tasks to be scheduled along with shorter ones. In this situation, max min ensures better makespan and lower degree of imbalance among machines.

Mao, et al. [45] and Li, et al. [46], suggest Max-min algorithm for task scheduling to balance the load of elastic cloud. The suggested algorithm conserves a task position table to evaluate the real time workload of VMs and expected execution time of tasks. The simulation result says that, Max-min algorithm boost the utilization of resource and shrink the response time for task scheduling.

The principal aim of improved Max-min algorithm is assigned task with maximum execution time to the resource, which gives minimum completion time than basic Max-min algorithm. Improved Max-min algorithm is established on predictable execution time as an alternate of complete time, which gives lower makespan [47]. For task scheduling in cloud computing, performance of Max-min algorithm is not achieved the better results. To solve this issue, Ming and Li [48] present an enhanced algorithm MMST based on Max-min. It decreases the waiting time and get the better resource utilization of tasks. Also, MMST algorithm shrinks the cost of cloud providers.

(3) Min-min

Min-min algorithm starts with a set of un-scheduled tasks and then establishes the minimum completion time for each task on all machines. Then the task with generally minimum completion time is selected and scheduled on the resultant machine. The scheduled task is then removed from task list and the procedure is repeated until the all un-scheduled tasks are finished.
Wang and Yu, et.al. [49], put forward an improved Min-min algorithm for task scheduling for enhancing the proficiency of cloud computing system. However, Min-min algorithms continuously complete the minimum and entire execution time for task firstly, and then simply complete in the shortest period is characterized for scheduling. The consequences show that the algorithm is effective for the task scheduling in cloud computing.

Further, Zhang and Xu et.al. [50], advise a Min-min task scheduling algorithm based on QoS limitation in cloud computing.

The recommended algorithms measure the similarity of resources or tasks, and then send to the users to fulfill their demands. Simulation results reveal that, Mul-QoS-Min-Min performs superior in enhancing the execution time and QOS fulfillment as compared with basic Min-min algorithm in cloud computing.

Patel G, et.al. [27], improved the Load Balancing Min-min (LBMM) algorithm for static task scheduling and maximize the utilization of resource in cloud computing.

Further, Chen, et.al. [51], bring in two new algorithms for scheduling to enhance the makespan, resource deployment and user priority in cloud computing. LBIMM algorithm and PA-LBIMM algorithm are based on Min-min algorithm. The simulation results show that both the LBIMM and PA-LBIMM algorithms are outperformed than the basic Min-min algorithm to improve the completion time, load balancing and user priority.

(4) Sufferage

Sufferage algorithm begins by calculating values of tasks for the minimum and second minimum completion times. The difference of the values is determined in the second stage and task with a minimum difference (sufferage) is allocated to the consistent VM or resource. Then the task is detached from un-assigned task list and resource availability list is updated. The procedure is repeated until all the tasks are scheduled.

Han, et.al. [52], propose a new scheduling algorithm composed of Sufferage algorithm and Min-min algorithm to improve the QoS for task **3**scheduling. In the comparison of simulation results, proposed algorithm show better performance in decreasing the makespan for task scheduling for cloud computing.

(5) Performance metrics

This comparative investigation of performance metrics for task scheduling is stand on cost, makespan, throughput and degree of imbalance. The performance metrics are discussed below:

a) Cost

Cost means the total expense generated against the utilization or usage of resources, which is compensated to the cloud providers by the cloud users. The major purpose is, the increase of revenue and profit for cloud providers while reducing the operating expense for cloud user with efficient utilization. Suppose the cost of a VM fluctuate from one another based on time substantial and VM's specifications as specified by the cloud providers, then Eq. 1 holds for the cost of executing task of a VM.

$$\text{Cost} = \sum_{i=1}^{n} task^{i}(Ci * Ti) \quad \text{------------------Eq.5.1}$$

Where *Ci* represents the cost of i^{th} VM and represents the execution time of i^{th} task.

b) Degree of imbalance

Degree of imbalance (DI) describes the amount of load distribution amongst the VMs regarding to their execution competencies. Here, *T*max, *T*min and *Tavg* signify the maximum, minimum and average overall execution time of task among total VMs, correspondingly [11].

$$DI = \frac{T\max - T\min}{Tavg} \quad \text{------------------Eq.5.2}$$

c) Makespan

Makespan is used to estimate the maximum completion time, by evaluating the finishing time of the latest task, when all tasks are scheduled. If the makespan of specific cloudlet or task is not minimized then the demand will not be completed on time [27, 82].

$$Makespan = max_{taski}\,(Fnh_{Time}) \quad \text{------------------ Eq.5.3}$$

Where, Fnh_{Time} shows the finishing time of i^{th} task.

d) Throughput

Throughput makes use of the consideration of total number of tasks, which are implemented successfully. In cloud computing, throughput means some tasks completed in a certain time period. Minimum throughput is required for task scheduling [81[[82][83].

$$Throughput = \sum_{task^{i}} Exe_{Time} \quad \text{------------------Eq.5.4}$$

Where, Exe_{Time} shows the execution time of i^{th} task.

These algorithms are implemented in CloudSim simulator in homogeneous and heterogeneous environment with and without using workload traces.

V. HOMOGENEOUS ENVIRONMENT

In the homogeneous environment, we have fixed specification of the VMs to check the performances of the heuristic algorithms for task scheduling for IaaS cloud computing, while changing the number of cloudlet with and without the workload traces using HPC2N [55] in our simulation. Table 1 shows the setting of experimental parameters for CloudSim in homogeneous environment.

Table 1: Simulation parameters for homogeneous environment.

Sr.No.	Entities	Parameter	Values
1	User	Number of users	5
2	Cloudlet	Number of Cloudlets	100-1000
		Length	2000
3	Host	Number of hosts	2
		RAM	2 MB
		Storage	10 GB
		Bandwidth	1 GB
4	Virtual Machine	Number of VM’s	15
		Type of policy	Time share
		RAM	512 MB
		Bandwidth	1000
		Speed in MIPS	1000
		Size	10 MB
		VMM	Xen
		Operating system	Linux
		Number of CPUs	2
5	Data Centre	No. of Data centers	2

Figure 5.1(a) shows the evaluation of cost between FCFS, MCT, MET, Max-min, Min-min and Sufferage algorithms without using workload traces in homogeneous environment.

The x-axis signifies number of cloudlets and y-axis signifies the cost per hour of the execution of tasks. The comparison outcomes show that the FCFS algorithm gives minimum cost than other heuristic algorithms without using the workload traces in homogeneous environment.

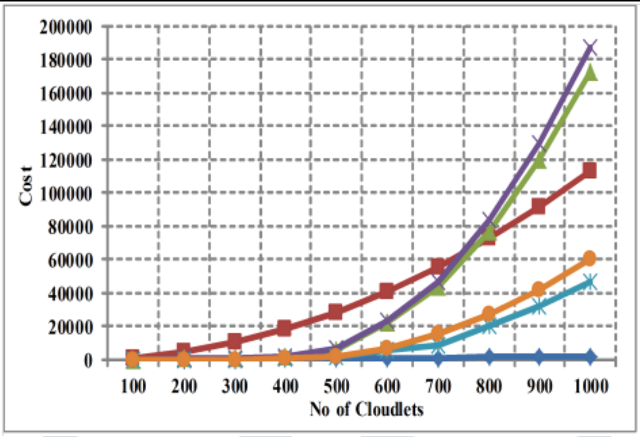


Figure 5.1(a): Cost in Homogeneous Environment without Workload traces

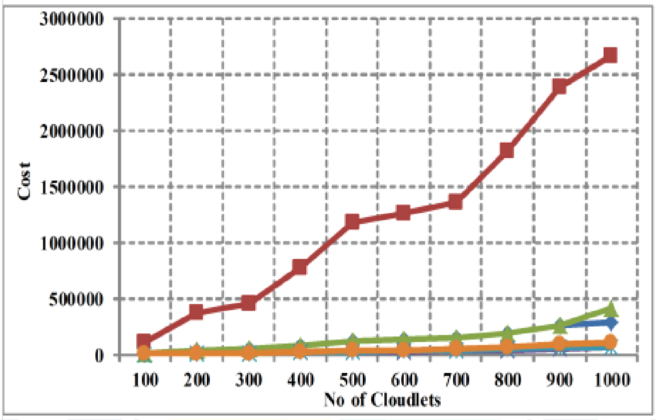


Figure 5.1 (b): Cost in homogeneous environment with workload traces.

Figure 5.1(b) shows the evaluation of cost between FCFS, MCT, MET, Max-min, Min-min and Sufferage algorithms by using workload traces in homogeneous environment.
The comparison outcomes demonstrate that the Max-min and Min-min algorithms give minimum cost (with minor difference) than other heuristic algorithms by using the workload traces in homogeneous environment.

Figure 5.2(a) shows the evaluation of degree of imbalance between FCFS, MCT, MET, Max-min, Min-min and Sufferage algorithms without using workload traces and Figure 5.2(b) with workload traces in homogeneous environment. Horizontal line signifies number of cloudlets and vertical line signifies the Degree of imbalance. The comparison results show that the MCT algorithm gives better Degree of imbalance than other heuristic algorithms in both cases of homogeneous environment.

In Figure 5.3(a), the evaluation of makespan produced is shown between FCFS, MCT, MET, Max-min, Min-min and Sufferage algorithms without using workload traces in homogeneous environment. The x-axis indicates the number of cloudlets and the y-axis indicates the makespan time. When the numbers of cloudlets are less, then FCFS, Min-min and sufferage algorithms give enhanced makespan. When the number of cloudlets is increased, FCFS algorithm produces better makespan time in homogeneous environment without workload traces. Figure 5.3(b) illustrates the difference in makespan produced between FCFS, MCT, MET, Max-min, Min-min and Sufferage algorithms by using workload traces in homogeneous environment.

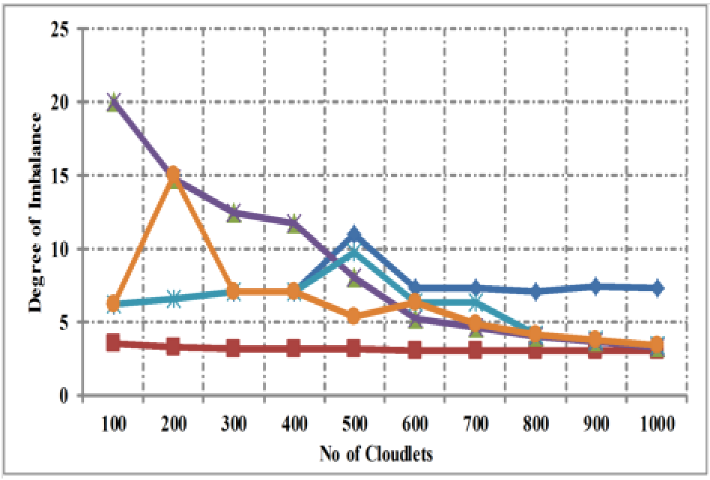


Figure 5.2(a): Degree of Imbalance in homogeneous environment without workload traces

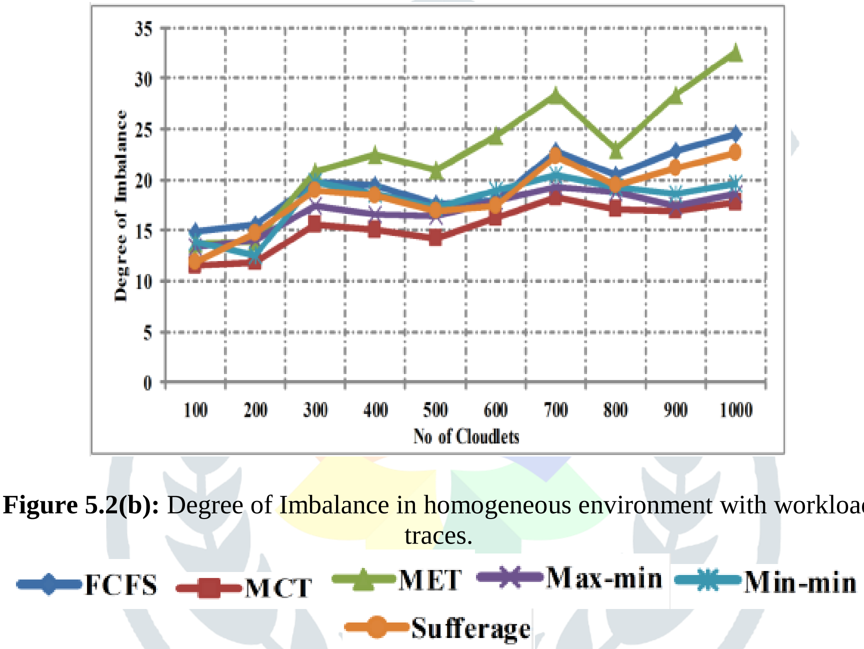


Figure 5.2(b): Degree of Imbalance in homogeneous environment with workload traces.

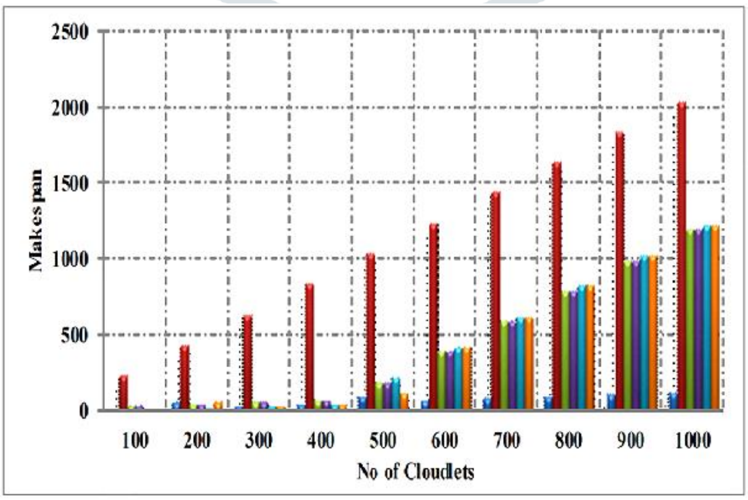


Figure 5.3(a): Makespan in homogeneous environment without workload traces

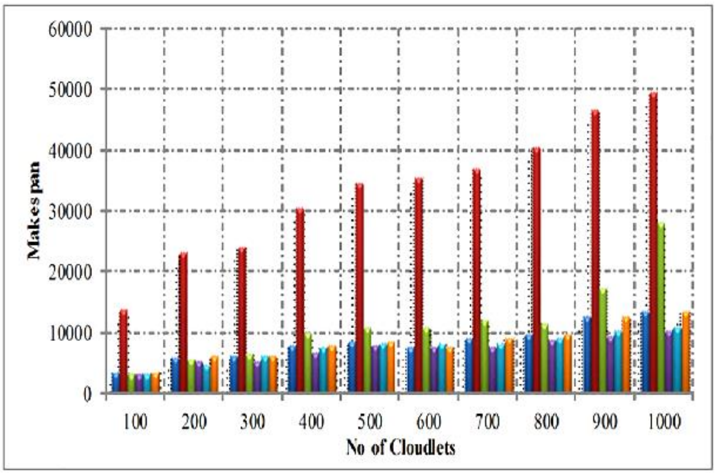


Figure 5.3(b): Makespan in homogeneous environment with workload traces.

FCFS MCT MET Max-min Min-min Sufferage

The evaluations of results clearly prove that the Max-min algorithm generates enhanced makespan than other heuristic algorithms by using the workload traces in homogeneous environment.

Figure 5.4(a) explains the evaluation of throughput between FCFS, MCT, MET, Max-min, Min-min and Sufferage algorithms without using workload traces in homogeneous environment. Horizontal axis denotes number of cloudlets and vertical axis denotes the throughput time. The simulation outcomes clearly prove that the Min-min algorithm provides better throughput than other heuristic algorithms, but the difference is not too much in homogeneous environment. Therefore, Sufferage and Max-min algorithms also show the better performance for throughput time.

Figure 5.4(b) indicates the assessment of throughput between FCFS, MCT, MET, Max-min, Min-min and Sufferage algorithms by using workload traces in homogeneous environment. The evaluation of results clearly displays that the Max-min algorithm provides better throughput than other heuristic algorithms with using the workload traces, but difference is not too much than Min-min algorithm in homogeneous environment.

In homogenous environment, the specifications of all VMs are same as static. In this case, on the behavior of algorithms performances are depended. FCFS algorithm shows more efficient performance for the cost and Makespan without workload traces. Similarly, MCT algorithm gives the better performance for

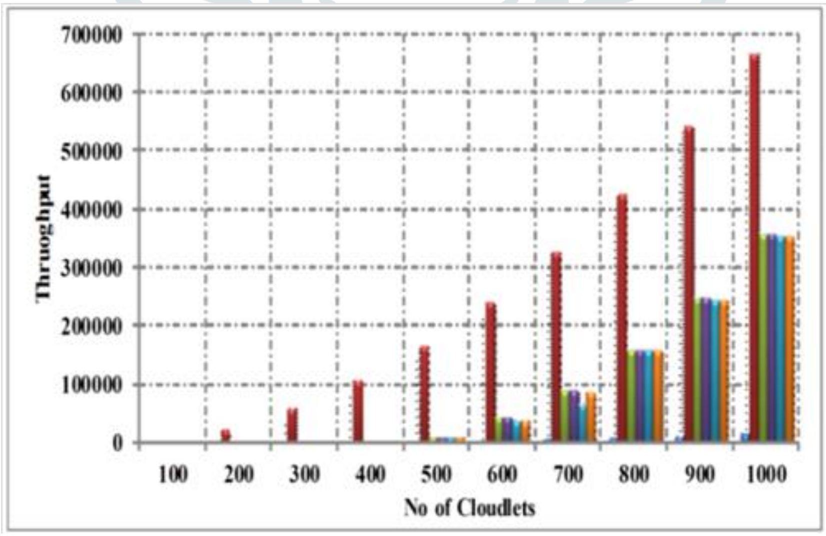


Figure 5.4(a): Throughput in homogeneous environment without workload traces

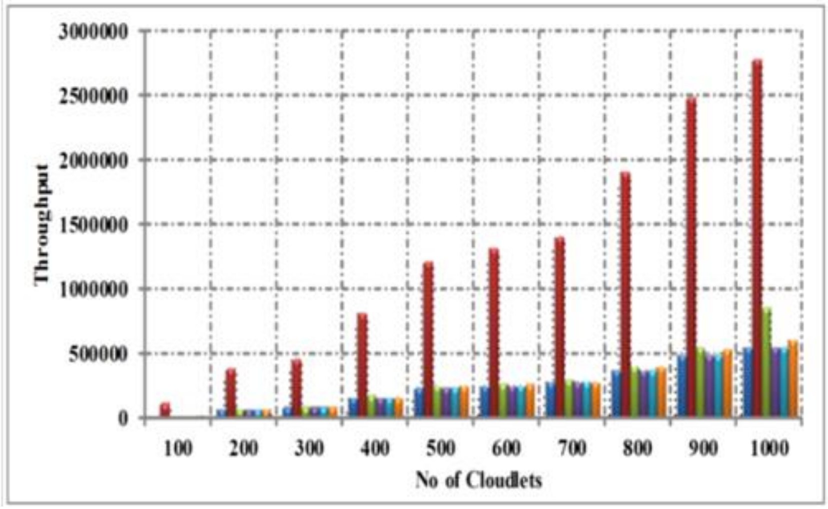
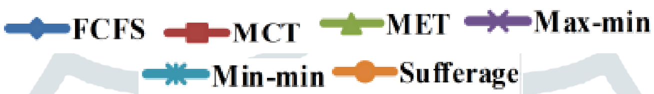


Figure 5.4(b): Throughput in homogeneous environment with workload traces



measuring the degree of imbalance in both cases. However, Max-min and Min-min show good performance with workload traces for achieving the minimum cost, makespan and throughput.

VI. HETEROGENEOUS ENVIRONMENT

In the heterogeneous environment, the VMs are selected randomly with different RAM, Bandwidth and MIPS, to check the performance of the heuristic algorithms for task scheduling for IaaS cloud computing.

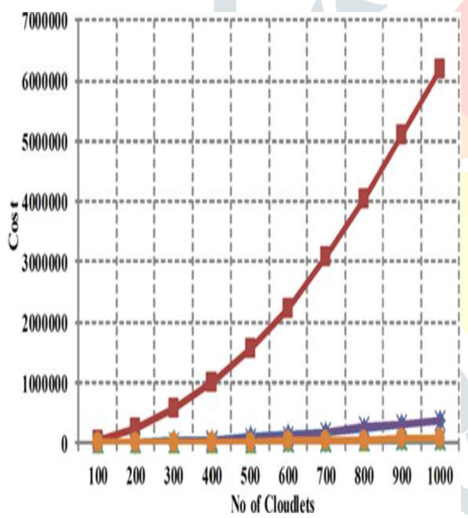
Six special workload traces are used to evaluate the performance by cost, degree of imbalance, makespan and throughput in heterogeneous environment. Four of them are generated using the uniform, normal, left-skewed and right-skewed distribution presented as S01, S02, S03 and S04 respectively. Uniform distribution shows the equal amount of small, large and medium size tasks. Normal distribution represents on the more medium, while less small and large size tasks. Skewness is the amount of asymmetric of probability distribution of tasks in the datasets. It can be left (negative) or right (positive). Left-skewed illustrates that the tail of the distribution is to the left of its mean, which includes the smaller and less large size tasks the dataset. Hence the right-skewed denotes that the tail of the distribution is to the right of its mean which includes the less small and large size task in the data sets. These datasets show the behavior of heuristics algorithm with different workloads. S05 and S06 are generated from Parallel Workload Archives consist of HPC2N (High-Performance Computing Center North) [55] and NASA Ames iPSC / 860 [56]. These workload archives are provided by Ake Sandgren and Bill Nitzberg, in the standard workload format (swf) recognized by the CloudSim tool. HPC2N contains the information of 527,371 tasks and NASA contains the information of 14,794 tasks. These workloads are mostly used to evaluate the performance of algorithms in cloud computing environment [11][12][91]. Table 7.2 shows the setting of experimental parameters for CloudSim in heterogeneous environment.

In Figure 6.1, the comparison of cost is shown between FCFS, MCT, MET, Max-min, Min-min and Sufferage algorithms with using workload traces including the Uniform, Normal, Left-Skewed, Right-Skewed, HPC2N and NASA in heterogeneous environment. The x-axis signifies the number of cloudlets and the y-axis signifies the cost per hour for the task execution.

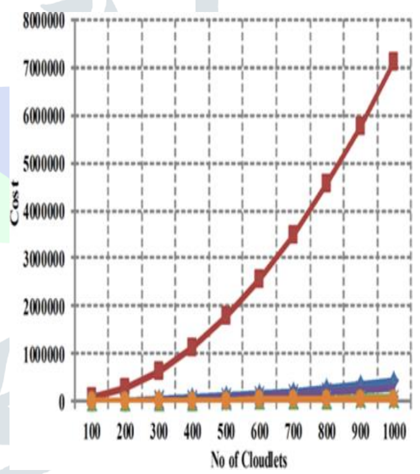
Table 2: Simulation parameters for heterogeneous environment.

Sr.No.	Entities	Parameter	Values
1	User	Number of users	10
2	Cloudlet	Number of Cloudlets	100-1000
		Length	2000
3	Host	Number of hosts	2
		RAM	20 GB
		Storage	1 TB
		Bandwidth	10 GB
4	Virtual Machine	Number of VM’s	25
		Type of policy	Time share
		RAM	128 to 15360 MB
		Bandwidth	128 to 15360 MB
		Speed in MIPS	256 to 30720
		Size	10 GB
		VMM	Xen
		Operating system	Linux
		Number of CPUs	2
5	Data Centre	No. of Data centers	2

The comparison of results clearly expresses that the Min-min algorithm provides improved makespan than other heuristic algorithms in all six S01 to S06 for heterogeneous environment.



(1) Uniform



(2) Normal

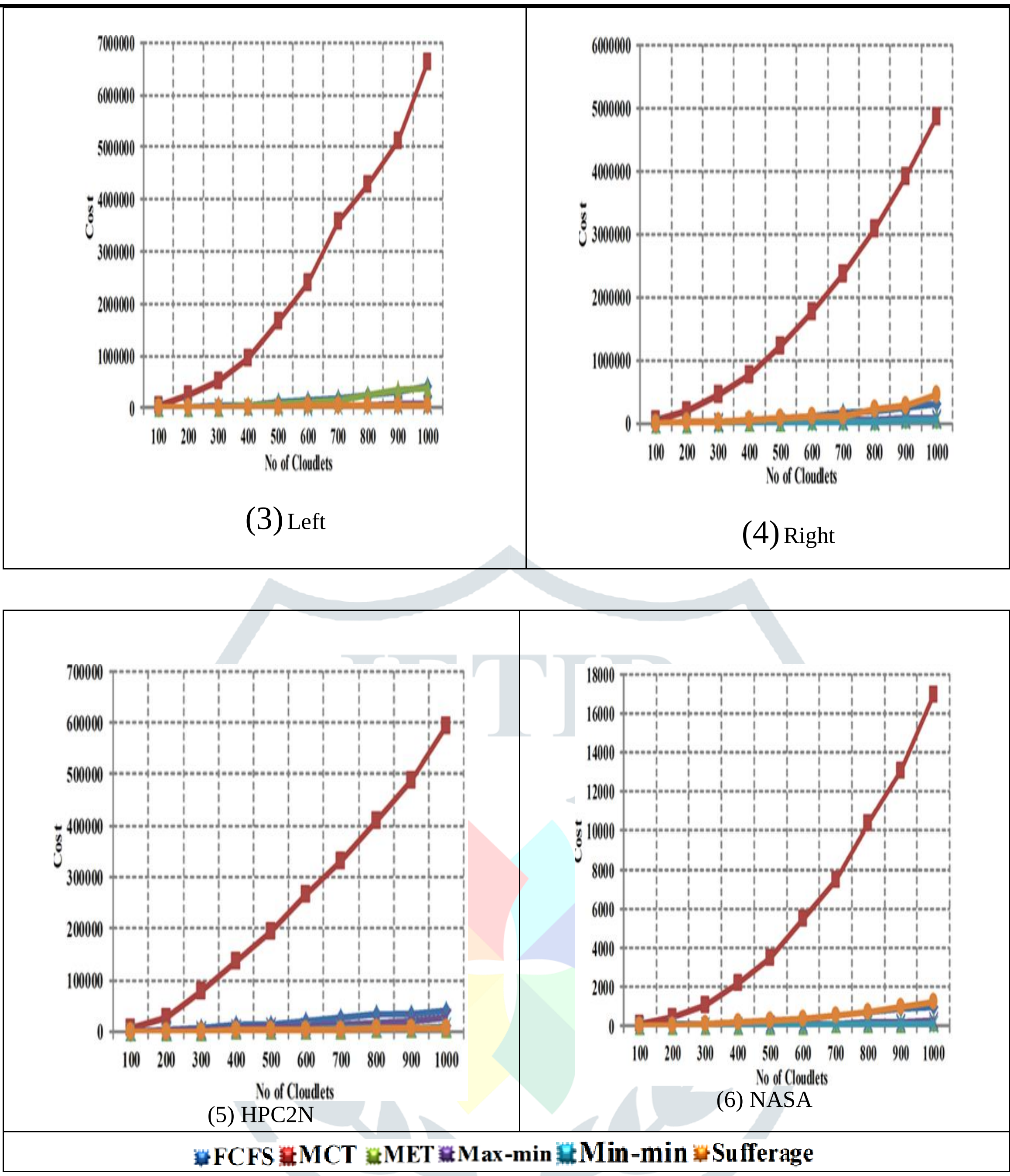
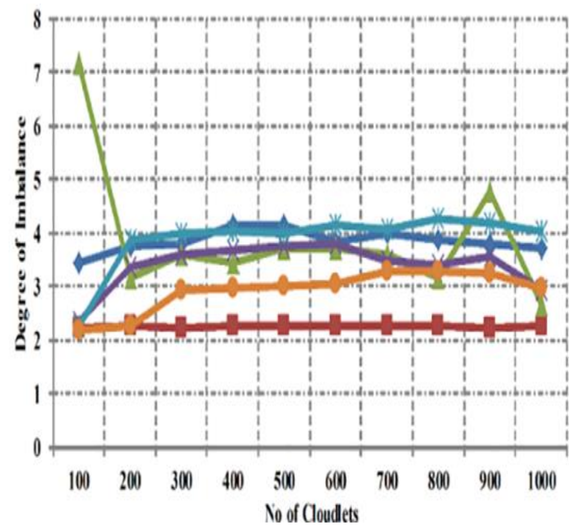
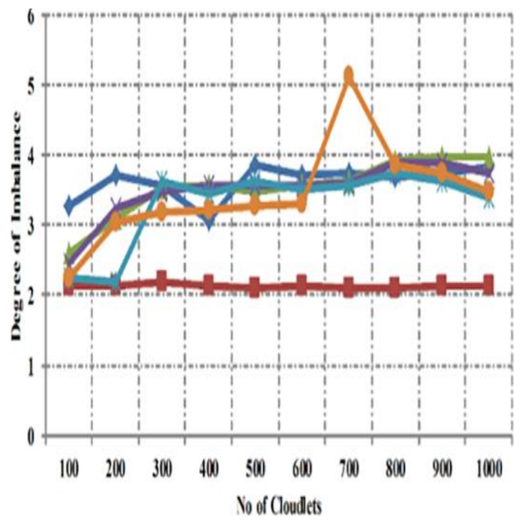


Figure 6.1: Cost in heterogeneous environment using (1) Uniform distribution, (2) Normal distribution, (3) Left skewed, (4) Right-skewed (5) HPC2N and (6) NASA.

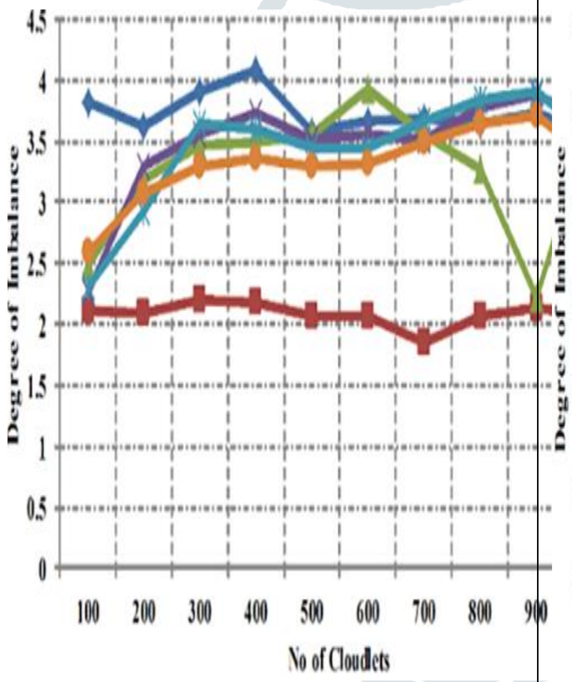
In Figure 6.2, the comparison of degree of imbalance is shown between FCFS, MCT, MET, Max-min, Min-min and Sufferage algorithms with using workload traces including the Uniform, Normal, Left-Skewed, Right-Skewed, HPC2N and NASA in heterogeneous environment. The horizontal axis signifies the number of cloudlets and the vertical axis signifies the throughput time. The simulation results show that the MCT algorithm provides better performance in uniform distribution, normal distribution and left-skewed, FCFS in right-skewed and sufferage algorithm in HPC2N and NASA for homogeneous environment.



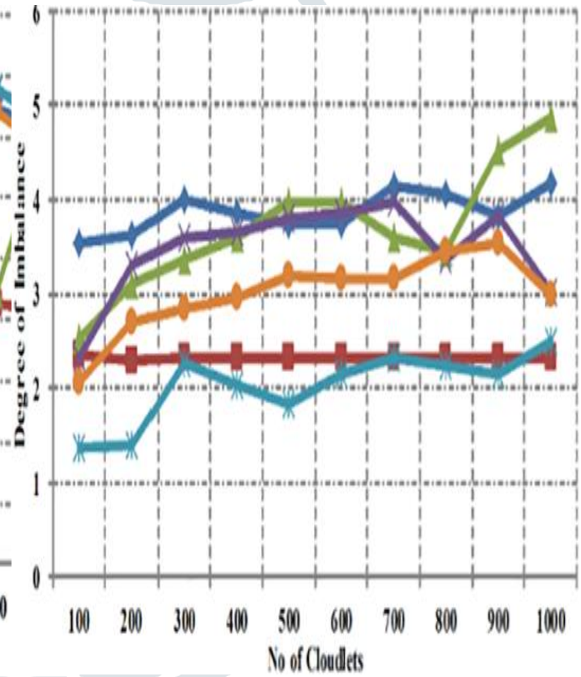
(1) Uniform



(2) Normal



(3) Left



(4) Right

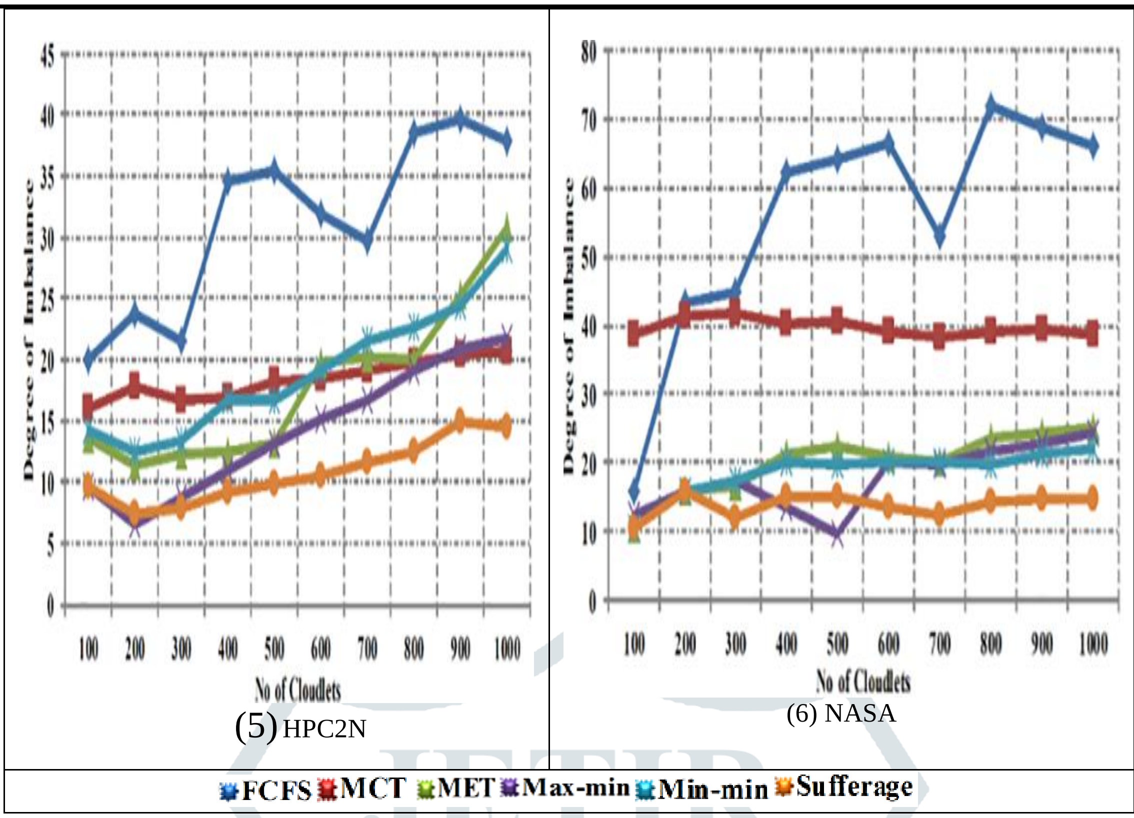
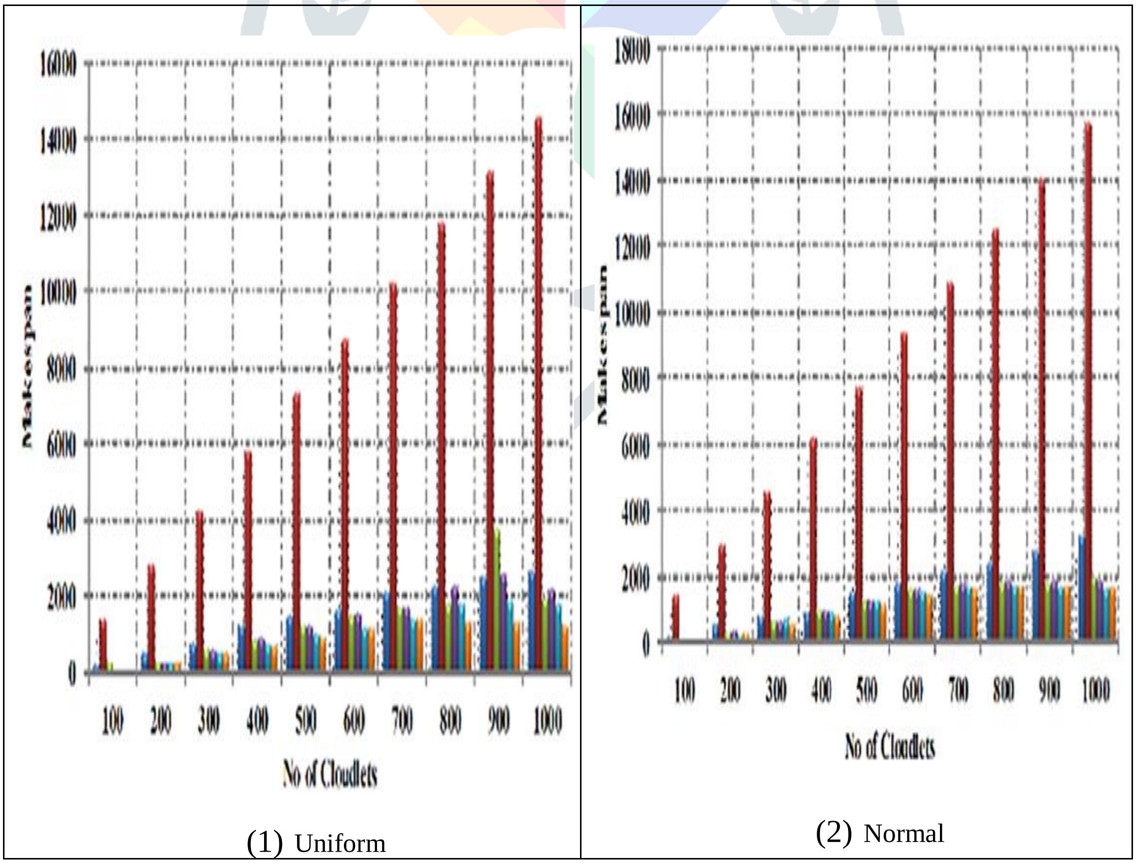


Figure 6.2: Degree of imbalance in heterogeneous environment using (1) Uniform distribution, (2) Normal distribution, (3) Left-skewed, (4) Right-skewed (5) HPC2N and (6) NASA.

In Figure 6.3, the evaluation of makespan time is shown between FCFS, MCT, MET, Max-min, Min-min and Sufferage algorithms with using workload traces including the Uniform, Normal, Left-Skewed, Right-Skewed, HPC2N and NASA in heterogeneous environment.



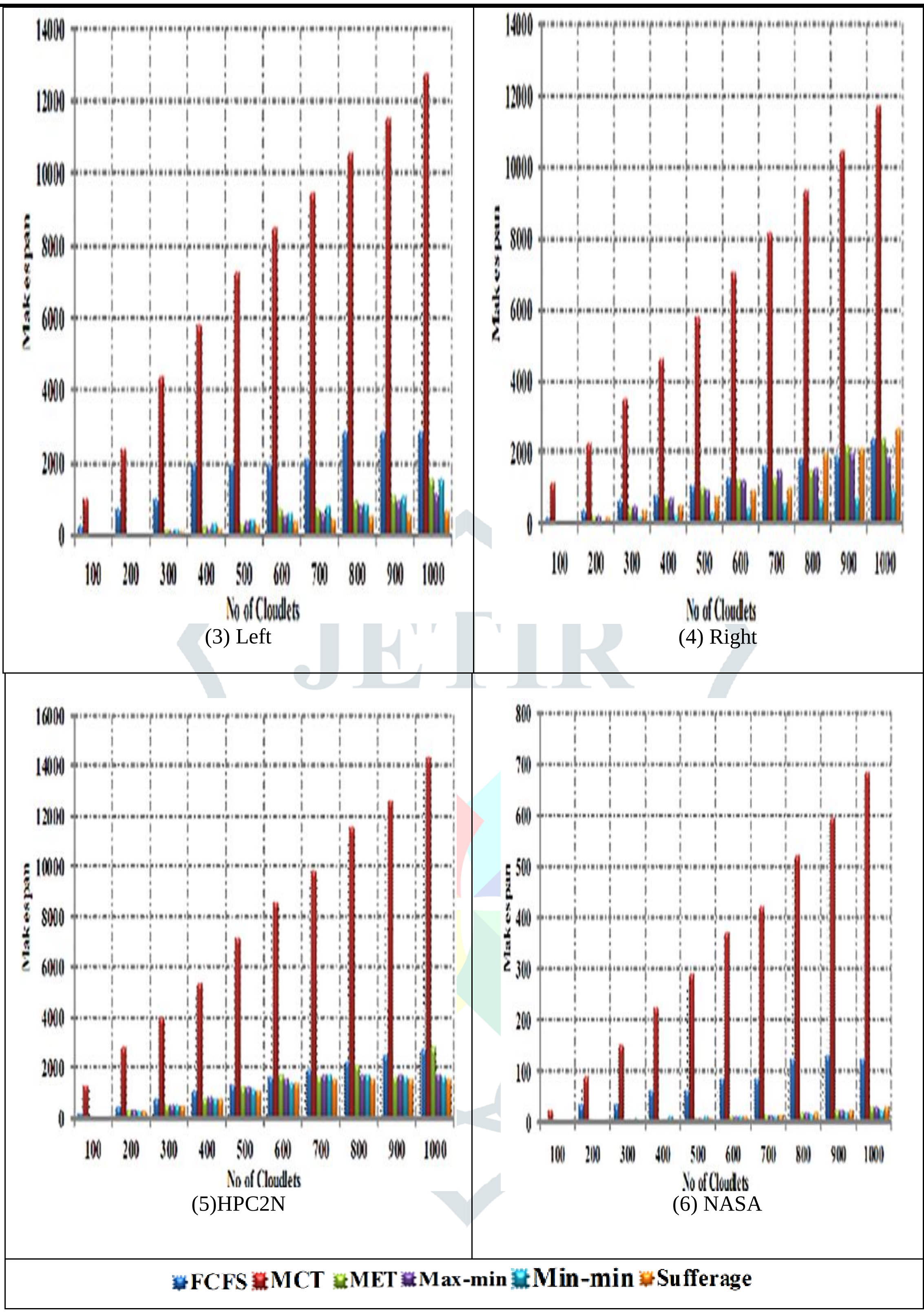
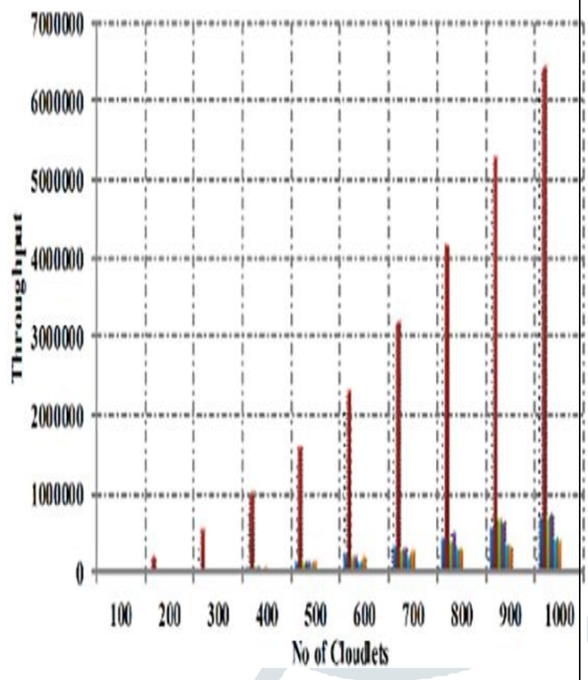


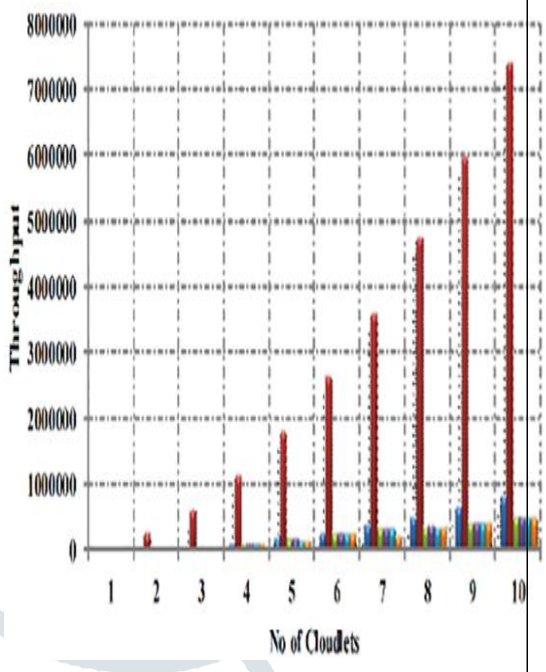
Figure 6.3: Makespan in heterogeneous environment using (1) Uniform distribution, (2) Normal distribution, (3) Left-skewed, (4) Right-skewed (5) HPC2N and (6) NASA.

The x-axis signifies the number of cloudlets and the y-axis signifies the makespan time. The comparison of results clearly shows that the Min-min algorithm provides improved makespan in uniform distribution, right-skewed and NASA; sufferage algorithm delivers in Left-skewed and HPC2N. While in normal distribution, both Min-min and sufferage algorithms give minimum makespan for heterogeneous environment.

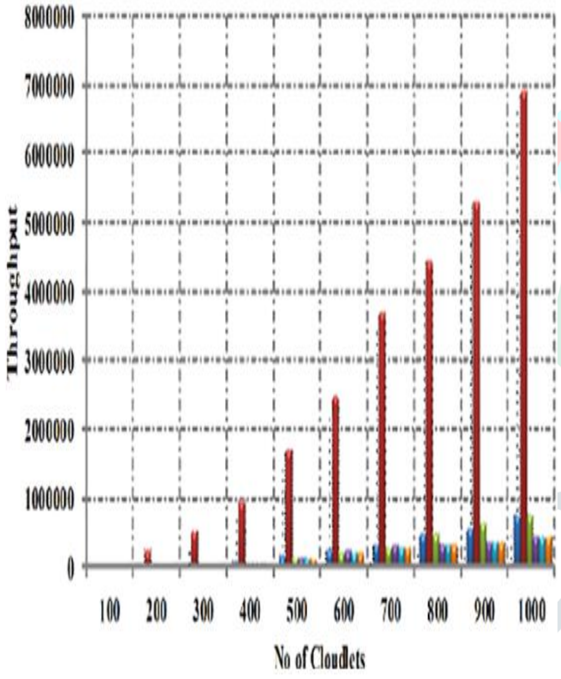
In Figure 6.4, the comparison of throughput time is shown between FCFS, MCT, MET, Max-min, Min-min and Sufferage algorithms with using workload traces including the Uniform, Normal, Left-Skewed, Right-Skewed, HPC2N and NASA in heterogeneous environment.



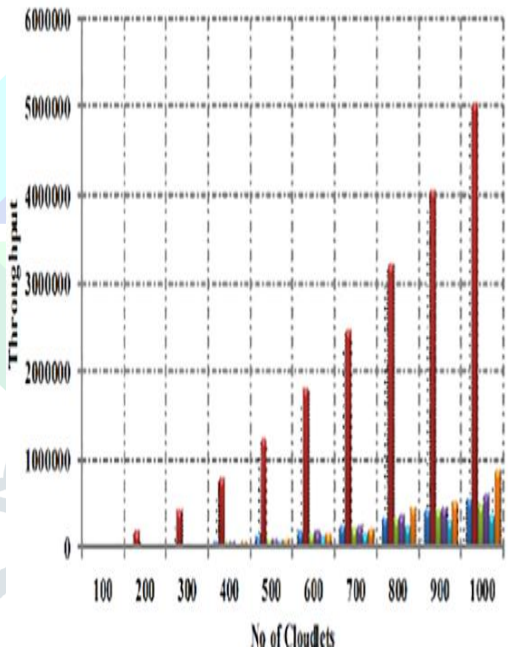
(1) Uniform



(2) Normal



(3) Left



(4) Right

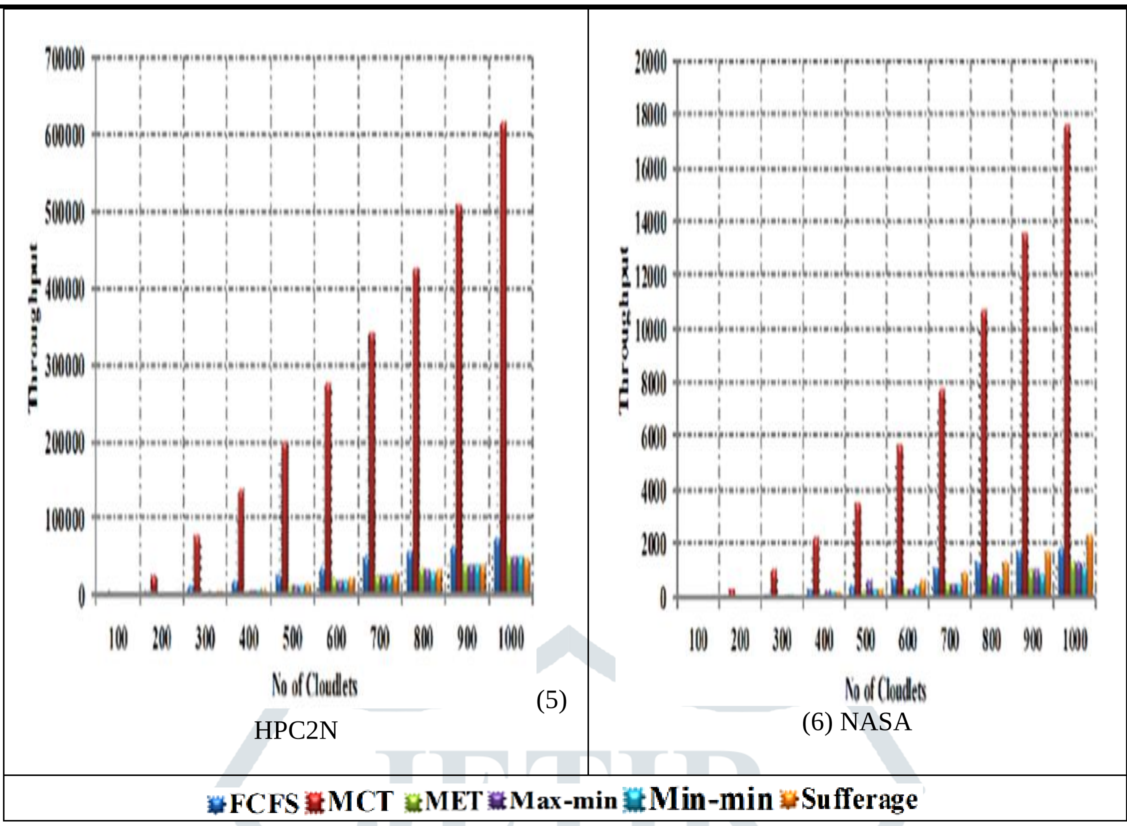


Figure 6.4: Throughput time in heterogeneous environment using (1) Uniform distribution, (2) Normal distribution, and (3) Left-skewed, (4) Right-skewed (5) HPC2N and (6) NASA

Horizontal axis signifies the number of cloudlets and vertical axis signifies the throughput time. The comparison of simulation results clearly show that the Min-min algorithm offers enhanced throughput in S01, S02, S03, S04 and S06, while sufferage algorithm offer improved throughput in S05 for heterogeneous environment.

In the heterogeneous environment, Min-min algorithm outperformed in S01 to S06 for optimizing the cost, makespan and throughput. Although the literature of task scheduling show that Min-min is not used for optimizing the parameter of cost for task scheduling in cloud computing. Mostly it is applied for minimizing the makespan, throughput and degree of imbalance. So that researchers can use Min-min algorithm for the optimizing cost for task scheduling in cloud with improved version and hybrid with other heuristics and meta-heuristics algorithms.

Further, sufferage algorithm is also better performed for makespan and throughput in heterogeneous environment. FCFS and MCT show better results for achieving the degree of imbalance for task scheduling.

In both homogeneous and heterogeneous environment with and without workload traces, the performance of FCFS algorithm is not good in finding the optimal cost, makespan and throughput. Hence, FCFS algorithm has performed poorly in terms of the degree of imbalance for task scheduling in both situations in IaaS cloud computing. MET algorithm performs averagely in as compared with all selected heuristics algorithms in relations to cost, makespan and throughput, while in case of degree of imbalance its performance is not considered to be good in all setups of homogeneous and heterogeneous environment for task scheduling. MCT algorithm also performs poorly in finding the cost makespan and throughput time in both homogeneous and heterogeneous environment with and without workload traces, whereas it performs averagely as compared with other algorithms in finding the degree of imbalance in both scenarios for the task scheduling in IaaS cloud computing.

In case of Min-min algorithm, it shows average performance in finding makespan and throughput in homogeneous environment, while it gives an ideal result when finding optimum cost, makespan and throughput in heterogeneous environment with and without workload traces. Also, Min-min algorithm gives average results in searching the degree of imbalance in all four scenarios for the task scheduling in IaaS cloud computing. Max-min only archives the optimal makespan in homogeneous environment with workload traces, otherwise it shows average results as compared with all selected heuristic algorithms for task scheduling in finding the cost, makespan, throughput and degree of imbalance in IaaS cloud computing system. Sufferage algorithm always accomplishes the median results in searching for optimal the cost and degree of imbalance in both scenarios, but it shows the best results when trying to achieve an enhanced makespan and throughput in heterogeneous environment for task scheduling in IaaS cloud computing.

After evaluating the performances of heuristic algorithms, we conclude that Min-min is most suitable for optimizing the cost, makespan and throughput, while Max-min algorithm also shows good performance for achieving the optimal task scheduling in IaaS cloud computing.

For the degree of imbalance, MET algorithm always shows better optimal results in attaining the optimal task scheduling in IaaS cloud computing. It is one the reasons that most researchers are using these heuristic algorithms as a standard for the comparison of their proposed techniques in cloud computing environment.

Babur Hayat Malik, Mehwashma Amir et.al.[59], carried out a study related to different existing task scheduling algorithms in a cloud environment has been presented. A short description of each algorithm methodology has been presented and most algorithms consider on one or two parameters. More satisfactory results can be achieved by adding more metrics to existing algorithms; based on different scheduling parameters such as execution time, load balance, Quality of service, performance, response time and makespan. The major problem in task scheduling is load balancing, response time, resource utilization and memory storage. Efficient scheduling algorithm can be achieved by combining different parameters to existing algorithms which will improve their overall performance of cloud environment.

Dr. Sudhir D. Sawarkar, Pratiksha D. Warule, et. al. [60], found that in the dynamic cloud environment where various jobs are submitted at anytime from anywhere, the management and scheduling of such tasks becomes complex, so there is a requirement of redefined mechanism to investigate algorithm before applying to the actual environment. In this paper priority-based task scheduling algorithms are reviewed in comparative form stating their advantages and disadvantages. The performance of the cloud computing depends on the algorithm used for job scheduling. From the above all priority-based algorithms, PA_LBIMM is superior as it considers load balancing with user priority which maximizes the performance of the algorithm.

Ruba Abu Khurma, Ahmad Abdel-Aziz Sharieh, et. al. [61], analyzed seven common task scheduling algorithms under cloud computing environment, namely: RR, MaxMin, MinMin, FCFS , MCT, PSO, and GA. They tested RR and MRR in the same environment using CloudSim simulator. Used the same parameters for both schedules. The number of cloudlets from 10 up to 300 tasks on a set of one VM up to six VMs. They noticed that, the values show a remarkable growth in the average waiting time when the number of cloudlets increases; this is because the total requested power of cloudlets increases while the available resource power is limited. The results show that when using MRR to schedule number of Cloudlets over number of VMs, the average waiting time becomes less than when using RR, using the same numbers of cloudlets and the CC environments.

Rahul Bhatt, Mayank Agarwal, et. al. [62], used the CloudSim toolkit which provides the good components for simulation of cloud environment. The experiments are performed with Sequential assignment which is default in CloudSim and with very common CPU scheduling algorithms i.e. first Come First Serve (FCFS). The simulations are conducted for different combinations of RAM and CPU. It is observed that, by varying the VM characteristics affect the time taken for cloudlet execution, debt incurred. The debt is increased with the increase in capacity of RAM Further analyzes of scheduling policies are required to study the impact of VMs in cloud computing. Presented, a Scalable Simulation of FCFS using CloudSim. In cloud computing environment task scheduling is one main problem, so there may be many challenges for the researchers. There are many researchers are working in the area improvement of task scheduling algorithms, but each algorithms have some limitations and drawbacks.

VII. DISCUSSION

Here, the performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing system is presented. These algorithms are executed with the help of Cloud-Sim simulator in homogeneous and heterogeneous environments with and without using workload traces. These algorithms are compared with each other based on some parameters like cost, degree of imbalance, makespan and throughput. For the heuristics considered here, overall, Min-min algorithm performs superior to other heuristics, while Max-min and sufferage algorithm give good results and MET algorithm always shows better performance in attaining the degree of imbalance for optimal task scheduling in IaaS cloud computing.

Further, heuristic algorithm is accepted as a standard to compare new future algorithms to enhance and resolve task scheduling and other vital research issues in IaaS cloud computing system. Due to the simplicity and easiness in implementation, heuristic algorithm shows quicker and optimal outcomes. Hybridization of heuristic and meta-heuristic algorithms give optimal results and cover the ambiguity of each other to achieve the optimization of task scheduling in IaaS cloud computing. In future work, we want to compare the performance of meta-heuristic algorithms for task scheduling in IaaS cloud computing. Furthermore, the Min-min algorithm needs to be improved for optimizing the cost for task scheduling in cloud computing.

REFERENCES

1. Foster I, Zhao Y, Raicu I, Lu S. “Cloud computing and grid computing 360-degree compared,” Grid Computing Environments Workshop, 2008 GCE'08; 2008: IEEE.

2. Mell P, Grance T. “The NIST definition of cloud computing,”, National Institute of Standards and Technology, 2009; 53(6):50.

3. Rimal BP, Choi E, Lumb I. “A Taxonomy, Survey, and Issues of Cloud Computing Ecosystems,” Cloud Computing: Springer; 2010 p. 21±46.

4. Gani A, Nayeem GM, Shiraz M, Sookhak M, Whaiduzzaman M, Khan S. “A review on interworking and mobility techniques for seamless connectivity in mobile cloud computing,” Journal of Network and Computer Applications. 2014; 43:84±102.

5. Ab Rahman NH, Choo K-KR “A survey of information security incident handling in the cloud,” Computers & Security. 2015; 49:45±69.

6. Khan S, Ahmad E, Shiraz M, Gani A, Wahab AWA, Bagiwa MA. “Forensic challenges in mobile cloud computing,” Computer, Communications, and Control Technology (I4CT), 2014 International Conference on; 2014: IEEE.

7. Guo L, Zhao S, Shen S, Jiang C. "Task scheduling optimization in cloud computing based on heuristic algorithm," Journal of Networks. 2012; 7(3):547±53.

8. Abdulhamid SM, Latiff MSA, Bashir MB "Scheduling Techniques in on-demand Grid as a Service Cloud: A Review," Journal of Theoretical & Applied Information Technology. 2014; 63(1).

9. Gorbenko A, Popov V. "Task-resource scheduling problem," International Journal of Automation and Computing. 2012; 9(4):429±41.

10. Mani SHH, Latiff MSA, Coulibaly Y. "An Appraisal of Meta-Heuristic Resource Allocation Techniques for IaaS Cloud," Indian Journal of Science and Technology. 2016;9(4).

11. Abdullahi M, Ngadi MA, Abdulhamid SM "Symbiotic Organism Search optimization based task scheduling in cloud computing environment," Future Generation Computer Systems. 2016; 56:640±50.

12. Abdullahi M, Ngadi MA "Hybrid Symbiotic Organisms Search Optimization Algorithm for Scheduling of Tasks on Cloud Computing Environment", PloS one. 2016; 11(6):e0158229. https://doi.org/10.1371/journal.pone.0158229 PMID: 27348127

13. Bansal N, Awasthi A, Bansal S. "Task Scheduling Algorithms with Multiple Factor in Cloud Computing Environment", Information Systems Design and Intelligent Applications: Springer; 2016. p. 619±27.

14. Thomas A, Krishnalal G, Raj VJ "Credit Based Scheduling Algorithm in Cloud Computing Environment," Procedia Computer Science. 2015; 46:913±20

15. Abdul Humid SiM, Latiff MSA, Idris I "Tasks Scheduling Technique using League Championship Algorithm for Makespan Minimization in IaaS Cloud," ARPN Journal of Engineering and Applied Sciences. 2014; 9(12).

16. Xue S, Li M, Xu X, Chen J, Xue S "An ACO-LB Algorithm for Task Scheduling in the Cloud Environment," Journal of Software. 2014; 9(2):466±73.

17. Abdulhamid SiM, Latiff MSA, Abdul-Salaam G, Mani SHH "Secure Scientific Applications Scheduling Technique for Cloud Computing Environment Using Global League Championship Algorithm", PloS one. 2016 ; 11(7) : e0158102. https://doi.org/10.1371/journal.pone.0158102 PMID: 27384239

18. Liu C-Y, Zou C-M, Wu P "A Task Scheduling Algorithm Based on Genetic Algorithm and Ant Colony Optimization in Cloud Computing," Distributed Computing and Applications to Business, Engineering and Science (DCABES), 2014 13th International Symposium on; 2014: IEEE.

19. Hung PP, Nguyen MV, Aazam M, Huh E-N "Task scheduling for optimizing recovery time in cloud computing," Computing, Management and Telecommunications (ComManTel), 2014 International Conference on; 2014: IEEE.

20. Zhao LF, Zhou SH, Chang WB., "Task Scheduling in Cloud Computing with Improved Firefly Algorithm," Applied Mechanics and Materials; 2014: Trans Tech Publ.

21. Li FF, Xie DQ, Gao Y, Xie GW, Guo QY "Research on Multi-QoS and Trusted Task Scheduling in Cloud Computing Environment," Applied Mechanics and Materials. 2013; 263:1892±6.

22. Wu X, Deng M, Zhang R, Zeng B, Zhou S. "A task scheduling algorithm based on QoS-driven in Cloud Computing," Procedia Computer Science. 2013; 17:1162±9.

23. Gabi D, Ismail AS, Zainal A, Zakaria Z, Abraham A. "Orthogonal Taguchi-based cat algorithm for solving task scheduling problem in cloud computing,". Neural Computing and Applications. 2016:1±19.

24. Zhu Y, Liang H. "Research for the virtual machine-oriented cloud resource scheduling algorithm," Information Management, Innovation Management and Industrial Engineering (ICIII), 2013 6th International Conference on; 2013: IEEE.

25. Cao Y, Ro C, Yin J. "Comparison of Job Scheduling Policies in Cloud Computing," Future Information Communication Technology and Applications: Springer; 2013. p. 81±7.

26. He ZT, Zhang XQ, Zhang HX, Xu ZW. "Study on New Task Scheduling Strategy in Cloud Computing Environment Based on the Simulator CloudSim," Advanced Materials Research; 2013: Trans Tech Publ.

27. Patel G, Mehta R, Bhoi U. "Enhanced Load Balanced Min-min Algorithm for Static Meta Task Scheduling in Cloud Computing," Procedia Computer Science. 2015; 57:545±53.

28. Mathew T, Sekaran KC, Jose J. "Study and analysis of various task scheduling algorithms in the cloud computing environment," Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on; 2014: IEEE.

29. Akilandeswari P, Srimathi H. "Survey and analysis on Task scheduling in Cloud environment," Indian Journal of Science and Technology. 2016; 9(37):1±6.

30. Thaman J, Singh M. "Current perspective in task scheduling techniques in cloud computing: A review," International Journal in Foundations of Computer Science & Technology. 2016; 6:65±85.

31. Tabak EK, Cambazoglu BB, Aykanat C. "Improving the performance of independent task assignment heuristics min-min, max-min and sufferage," IEEE Transactions on Parallel and Distributed Systems. 2014; 25(5):1244±56.

32. Gandomi AH, Yang X-S, Talatahari S, Alavi AH. "Meta heuristic applications in structures and infrastructures," : Newnes; 2013.

33. Beheshti Z, Shamsuddin SMH. "A review of population-based meta-heuristic algorithms," International Journal of Advanced Soft Computing Applications. 2013; 5(1):1±35.

34. Gupta H, Dastjerdi AV, Ghosh SK, Buyya R. "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments," arXiv preprint arXiv:160602007. 2016.

35. Higashino WA, Capretz MA, Bittencourt LF. "CEPSim: Modeling and simulation of Complex Event Processing systems in cloud environments," Future Generation Computer Systems. 2015.

36. Jamali S, Alizadeh F, Sadeqi S. "Task Scheduling in Cloud Computing Using Particle Swarm Optimization," The Book of Extended Abstracts. 2016: 192.

37. Lakra AV, Yadav DK. "Multi-objective tasks scheduling algorithm for cloud computing throughput optimization," Procedia Computer Science. 2015; 48:107±13.

38. Zuo L, Shu L, Dong S, Zhu C, Hara T. "A Multi-Objective Optimization Scheduling Method Based on the Ant Colony Algorithm in Cloud Computing," Access, IEEE. 2015; 3:2687±99.

39. Raju IRK, Varma PS, Sundari MR, Moses GJ, "Deadline Aware Two Stage Scheduling Algorithm in Cloud Computing," Indian Journal of Science and Technology. 2016; 9(4).

40. Mondal B, Dasgupta K, Dutta P. "Load balancing in cloud computing using stochastic hill climbing-a soft computing approach," Procedia Technology. 2012; 4:783±9.

41. Dasgupta K, Mandal B, Dutta P, Mandal JK, Dam S. "A genetic algorithm (ga) based load balancing strategy for cloud computing," Procedia Technology. 2013; 10:340±7.

42. Sindhu S, Mukherjee S. "Efficient task scheduling algorithms for cloud computing environment," High Performance Architecture and Grid Computing: Springer; 2011. p. 79±83.

43. Tewfik M, El-Sis A, Keck AE, Torkey F. "Cloud task scheduling based on ant colony optimization," Computer Engineering & Systems (ICCES), 2013 8th International Conference on; 2013: IEEE.

44. Du Kim H, Kim JS, "An online scheduling algorithm for grid computing systems," International Conference on Grid and Cooperative Computing; 2003: Springer.

45. Mao Y, Chen X, Li X. "Max±min task scheduling algorithm for load balance in cloud computing," Proceedings of International Conference on Computer Science and Information Technology; 2014: Springer.

46. Li X, Mao Y, Xiao X, Zhuang Y. "An improved max-min task-scheduling algorithm for elastic cloud. Computer, Consumer and Control," (IS3C), 2014 International Symposium on; 2014: IEEE.

47. Devipriya S, Ramesh C. "Improved max-min heuristic model for task scheduling in cloud. Green Computing, Communication and Conservation of Energy," (ICGCE), 2013 International Conference on; 2013: IEEE.

48. Ming G, Li H. "An improved algorithm based on max-min for cloud task scheduling", Recent Advances in Computer Science and Information Engineering: Springer; 2012. p. 217±23.

49. Wang G, Yu HC. “Task Scheduling Algorithm Based on Improved Min-Min Algorithm in Cloud Computing Environment,” Applied Mechanics and Materials. 2013; 303:2429±32.

50. Zhang Y, Xu B. “Task Scheduling Algorithm based-on QoS Constrains in Cloud Computing,” International Journal of Grid and Distributed Computing. 2015; 8(6):269±80.

51. Chen H, Wang F, Helian N, Akanmu G. “User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing,” Parallel Computing Technologies (PARCOMPTECH), 2013 National Conference on; 2013: IEEE.

52. Han H, Deyui Q, Zheng W, Bin F. “A Qos Guided task Scheduling Model in cloud computing environment,” Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on; 2013: IEEE.

53. Mani SHH, Latiff MSA, Coulibaly Y. “Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities,” Journal of Network and Computer Applications. 2016; 68:173±200.

54. Tsai J-T, Fang J-C, Chou J-H. “Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm,” Computers & Operations Research. 2013; 40(12):3045±55.

55. Abdulhamid, SM, Mani, SHH, Abdul Latiff MS, Abdullahi M, Usman MJ (2017): “Cloud Workloads (HPC2N Dataset),” Figshare. Retrieved: 11 25, Apr 18, 2017 (GMT)

56. Abdulhamid, S.M.Madni, SHH, Abdul Latiff MS, Abdullahi M.Usman MJ (2017): “Cloud Workloads (NASA Dataset),” Figshare. Retrieved: 10 20, Apr 17, 2017 (GMT)

57. Tchernykh A, Lozano L, Schwiegelshohn U, Bouvry P, Pecero JE, Nesmachnow S, et al. “Online bi-objective scheduling for IaaS clouds ensuring quality of service. Journal of Grid Computing,” 2016; 14 (1):5±22.

58. PLOS ONE | https://doi.org/10.1371/journal.pone.0176321 May 3, 2017.

59. Babur Hayat Malik, Mehwashma Amir, et.al. “Comparison of Task Scheduling Algorithms in Cloud Environment,” (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 5, 2018.

60. Dr. Sudhir D. Sawarkar, Pratiksha D. Warule, et. al. Mar 2018, “Review on Priority Based Task Scheduling In Cloud Computing,” International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 5, Issue 3, March 2018, ISSN (Online) 2394-2320.

61. Ruba Abu Khurma, Ahmad Abdel-Aziz Sharieh, et. al. “Task scheduling algorithm in cloud computing based on modified round robin Algorithm,” Journal of Theoretical and Applied Information Technology September 2018.

62. Rahul Bhatt, Mayanak Agarwal, et. al. [96], “A Technological Review on Scheduling Algorithm to Improve Performance of Cloud Computing Environment,” International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-8 Issue-6C, April 2019.