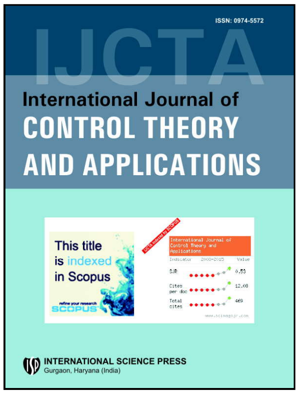


International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 14 • 2017

Task Scheduling Algorithm Using Multi-Objective Functions for Cloud Computing Environment

Abhikriti Narwal[1] and Sunita Dhingra[2]

[1] Research Scholar, Department of CSE, U.I.E.T,M.D.U, Rohtak, Haryana, India,
Email: abhikritiin@gmail.com
[2] Assistant Profesor, Department of CSE, U.I.E.T, M.D.U, Rohtak, Haryana, India,
Email: sunitadhingramdu@rediff.com

Abstract: Cloud computing is a platform for computing resources that are delivered as a service over an internet network to the customers. Since the number of users are increasing day-by-day and volume of data is also expanding extensively. Hence, to meet the needs of customers and to achieve the proper assignment of resources to available VM, task scheduling comes into play. There are many jobs that are required to be executed by the available resources to achieve best performance in order to achieve the various objectives like minimal total time for completion, shortest response time, utilization of resources and etc. To achieve these different objectives and high performance of cloud computing environment, it is needed to design, develop, and propose a scheduling algorithm that outperforms the appropriate allocation of jobs with different factors. Scheduling of tasks with single criteria is not preferable due to increasing complexity. In addition, providing good quality of services to the users is a decisive task for the providers as at the same time there are a large number of tasks running at the provider's side. This paper proposes a multi-objective task scheduling algorithm that considers wide variety of attributes in cloud environment and uses non-dominate sorting for prioritizing the tasks. The proposed algorithm considers three parameters i.e. Total processing cost, total processing time and average waiting time. The paper aims to improve the performance and compare the performance with FCFS, SJF and previously implemented multiobjective task scheduling algorithm.

Keywords: Cloud Computing, Task Scheduling, Priority, Non-Dominating sorting, Quality of Service (QoS), Virtual Machine (VM).

1. INTRODUCTION

Cloud computing has evolved as a new dimension for the dynamic provisioning of computing services supported by data centers that often employ Virtual Machines (VM) for combination and environment segregation purposes. It is outlined as a specialized distributed-computing paradigm that has gained wide known acceptance as a resource platform for on-demand, high-accessibility, and high-adaptability access to resources alongside giving better utilization of distributed resources, while offering dynamic, flexible infrastructures and quality of service (QoS). Cloud computing has realized the dream of computing as a utility by delivering an infrastructure, platform and applications as services on pay per use basis which are known as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) respectively in the industrial terms. Rather than establishing the basic hardware and software setup, the business enterprises now just have to invest finance and time on innovative ideas to create the business value for their services. With the ability to provide the services over the internet anytime and anywhere, cloud computing has successfully become an integral part of business enterprises, academics and research. In this computing model, clients utilize the required administrations without being bothered to know how these administrations is conveyed or from where are they getting facilitated. With the expanding number of clients in distributed computing, the volume of information is likewise hiking. The resource requests for various jobs are fluctuating over time. One of the significant contributions of distributed computing is to avail all of the resources at one spot to structure a cluster and to perform the resource allocation based on request performed by different users. To integrate and make good use of resources at various scales, cloud computing needs efficient methods to manage them. With the continuous rise of Cloud Computing, virtualization gets to be one of the promising advancements to actualize energy efficiency in data centers, which permits the flexible provisioning and arrangement of servers and workloads in the data centers. There are a large number of virtual servers in a single datacenter running at any instance of time, supporting number of tasks and in the meantime the cloud system continues to take up the batches of task requests. During this, few target servers are needed to be picked out of many available servers, which can fulfill a batch of incoming tasks. It demonstrates task scheduling is a valuable issue which is used to determine the performance of cloud service provider. It is crucial and extremely important to develop an efficient task scheduling algorithm to completely utilize the power of cloud computing and to enhance its performance and throughput. A proficient task scheduling calculation enhances the general framework performance and helps service supplier to furnish fine Quality of Service (QoS).

The rest of the paper is organized as follows. Related Work is analyzed in Section II. Proposed task scheduling algorithm is explained in section III. Experimental results are presented in section IV. Concluding remarks are given in section IV.

2. RELATED WORK

With the vast popularity, cloud computing is attracting an increasing number of applications to run in the remote data centers as most of which involve complex structures and need parallel processing capabilities to execute the jobs efficiently. The performance of the whole system completely relies on the efficiency of the task scheduling algorithm. Several algorithms and protocols have been proposed with time to address the task scheduling mechanism in the cloud computing.

Dr. Amit Agarwal et.al [1] explained the working of conventional algorithms FCFS and round robin in context of task scheduling in cloud. FCFS was considered more as a default process where the VMs are allocated to the host in a straightforward manner. FCFS being non preemptive, came up with drawback where the shortest tasks had to wait for the long task at the front to finish and it suffers from the Belady's Anamoly. To deal with it, an alternative Round robin(RR) algorithm was used where each job in a queue had the same execution time and are executed in turn and do not have to be waited for the previous one to get completed. But in case of heavy load, RR takes a long time to complete all the jobs which was the performance hurdle. To address the drawbacks of both the conventional algorithms, a priority based scheduling mechanism was also proposed where the tasks are prioritized according to their size such that the task with highest size is ranked highest and the Virtual Machines are also prioritized according to their MIPS value such that the VM with highest MIPS has the highest rank. Nimisha Singla et.al [2] further came up with priority based scheduling scheme with fault tolerance aspect. The algorithm aims at single fault tolerance among the servers and reallocates the faulty servers task to the new server which has minimum load at the instant of the fault. The number of faulty servers that cloud can hold can be determined as:

$$\text{No. of faulty servers be handled} = \frac{\text{total cloud load - current cloud load}}{\text{max no. of tasks running on a single server}}$$

The algorithm picks up number of tasks per server as a parameter for load calculation and determine on server value of load and determines the server with minimum load for reallocation. In this algorithm the fault of the server is being detected by virtual machine on selection policy of Minimum Migration Time (MMT).

Anton Beloglazov et.al [3] came up with the idea of Green Cloud computing that not only minimizes the operational costs but also deal with the environmental aspect. The paper discusses the open research challenges in energy-aware resource management and aims to develop efficient policies and algorithms for virtualized data centers to achieve a more sustainable and eco-friendly technology to drive commercial, scientific, and technological advancements. It has been observed that data centers represent a huge energy consumption sector of the economy and a significant source of carbon emissions. This paper proposes energy – aware resource allocation algorithm utilizing the dynamic consolidation of VMs to reduce the greenhouse gas emissions. Gregory Katsaros et.al [4] presented a service framework for achieving multi-source monitoring in Cloud environment. The proposed solution allows the collection of metrics from the physical and the virtual infrastructure as well as the services executed within the virtual environment and energy-related parameters. Chia-Ming Wu et.al [5] introduced an environment friendly energy-efficient scheduling algorithm which aims to achieve two goals- to provide the appropriate scheduling for a job and further to provide the appropriate voltage and frequency supply for the servers using Dynamic Voltage and Frequency Scaling (DVFS) technique. To select VMs for executing jobs, the priority job scheduling is used. The VMs are chosen by weight registered and the SLA level required by clients. The DVFS method is utilized to control the supply voltage and recurrence for servers in Cloud computing. This strategy can diminish the energy utilization of a server when it is in the idle mode or the light workload. Altino M. Sampaio et.al [6] proposed two algorithms that involve proactive fault tolerance to address node disappointments. The goal is to augment the helpful work performed by the expended energy in situations where the framework nodes are liable to disappointment. This goal suggests an expansion in the measure of helpful Mflops prepared per energy unit, and in addition the quantity of jobs finished by the expended power. To achieve this objective, two dynamic VM allocation algorithms, POFAME and POFARE have been developed which utilize two distinct strategies to give energy effective virtual clusters to execute assignments under their deadlines. Dian Shen et.al [7] emphasizes upon the issue of right- sizing of the data center in order to achieve energy-efficiency by taking into consideration of server failures and the overhead introduced by virtualization. The paper proposed a stochastic model of data centers using Queueing theory to analyze the dynamic nature of data centers. Ching-Hsien Hsu et.al [8] presents an energy-aware task consolidation (ETC) technique that minimizes energy consumption. ETC minimizes the energy consumption by limiting CPU use under a specified peak threshold. ETC achieves this by consolidating tasks amongst virtual clusters. Moreover, the energy cost model considers network latency when a task migrates to another virtual cluster. Despite of numerous advantages offered by virtualization, it has many drawbacks too which are presented by Peng Xiao et.al [9]. Virtualization is an effective approach to enhance the energy efficiency but in case of I/O virtualization many energy efficiency losses occur. The paper deals with this challenge and proposes an energy-efficiency enhanced virtual machine (VM) scheduling policy known as Share-Reclaiming with Collective I/O (SRC-I/O) which allows VMs to reclaim extra CPU shares in certain conditions so as to increase CPU utilization. Simultaneously, it separates I/O-intensive VMs from CPU-intensive ones and schedules them in a collective manner, so as to reduce the context-switching cost when scheduling mixed workloads. Atul Vikas Lakra et.al [10] proposed a multi-objective task scheduling algorithm for mapping tasks to a Vms improving the throughput of the datacenter and reducing the cost without compromising the SLA (Service Level Agreement) in cloud SaaS environment.

3. THE PROPOSED WORK

3.1. Architectural Framework

The system architecture consists of the cloud broker, the VM manager, the scheduling algorithm, servers and VMs. Cloud involves several data centers which contain a network of virtual services facilitating the user to access and deploy applications from anywhere in the world on demand at competitive costs depending on their QoS requirements. Each VMs have distinctive ability to execute diverse QoS's tasks seek by the client.

Consumers/Brokers: A job is given in the form of minimum and maximum amount of resources in frequency (MHz) that it requires. Cloud consumers or their brokers submit service requests from anywhere in the world to the Cloud. It is important to notice that there can be a difference between Cloud consumers and users of deployed services. For instance, a consumer can be a company deploying a web-application, which presents varying workload according to the number of "users" accessing it.

VM Manager: Keeps track of the availability of VMs and their resource usage. It is in charge of provisioning new VMs as well as reallocating VMs across physical machines to adapt the placement. When the VM manager receives a job and the requirements of the job, it sends the job requirements and states of resources to the scheduling algorithm.

VMs: Multiple VMs can be dynamically started and stopped on a single physical machine according to incoming requests, hence providing the flexibility of configuring various partitions of resources on the same physical machine to different requirements of service requests. Multiple VMs can concurrently run applications based on different operating system environments on a single physical machine. By dynamically migrating VMs across physical machines, workloads can be consolidated and unused resources can be switched to a low-power mode, turned off or configured to operate at low-performance in order to save energy.

The scheduling algorithm searches available servers to create VMs for allocating a job under the requirements of the job. In the algorithm, the SLA also has to be taken into account. The scheduling algorithm selects the solution that consumes the least energy. If all available servers cannot satisfy the requirements of the job, the scheduling algorithm has to notice the VM manager to power on proper servers. After making a decision, the scheduling algorithm sends the solution to the VM manager.

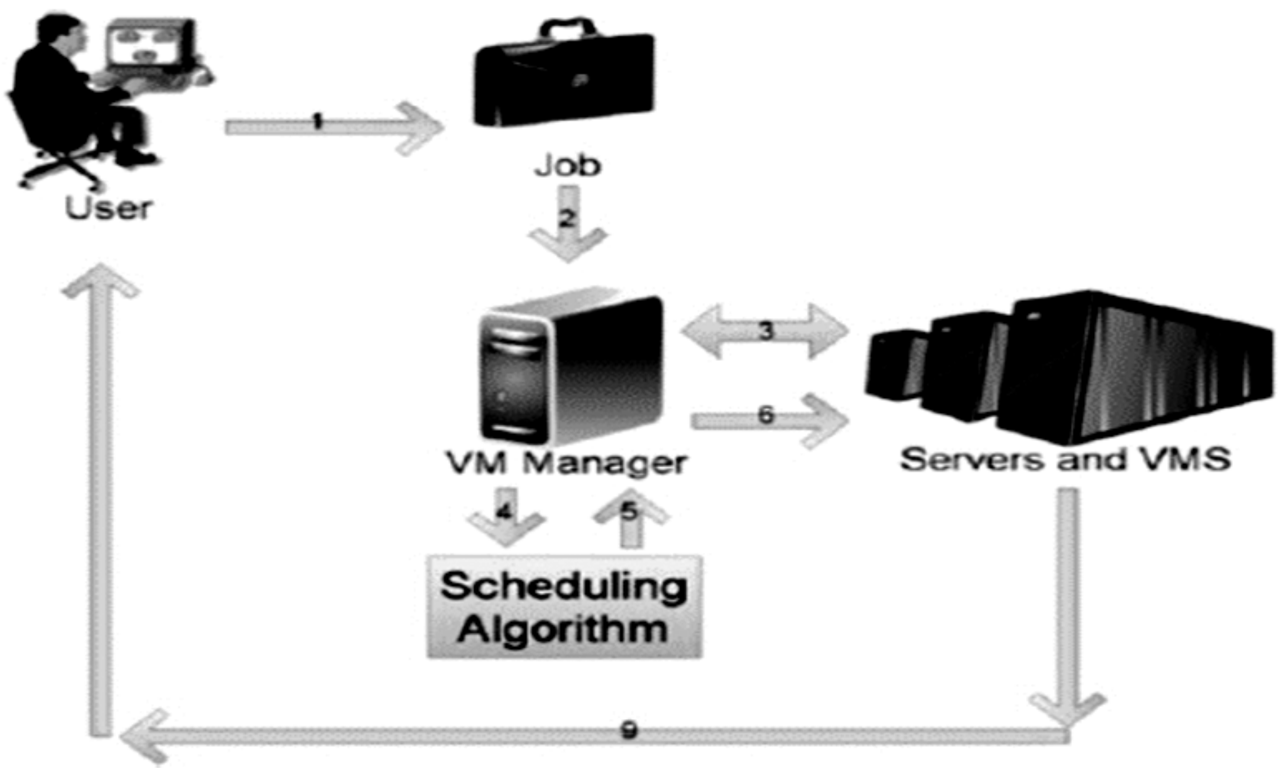


Figure 1: Architectural Framework

1) Initialize the virtual machines list
2) Initialize the task list.
3) Sort the virtual machines using QOS parameters (MIPS and Granularity size).
4) Sort the task list using non-dominating Sorting (considering cloudlet length and cloudlet file outsize as function parameters)
5) Assigning the task to the virtual machines in groups. Like if there are 30 tasks and we have 10 virtual machines the assigning each virtual machine 3 tasks
6) This process of allocation will be repeated for all tasks.

3.2. Problem Formulation

The vast research carried on task scheduling problem describes it as a NP-hard problem. Since scheduling is a challenging subject in cloud computing, it can be handled by numerous heuristic methods which provide the solution for the problem within the restrictions. After the submission of the collection of assignments to the cloud, the most important step is scheduling and to screen the execution of the assignment. Scheduling involves the mapping of the tasks to the resources. The basic idea is to bind the set of assignments received by the broker to the available VMs; with an aim to diminish the total execution time of the tasks and the operational cost.

Various algorithms have been proposed and executed till date, for example, First Come First Serve, Min-Min calculation, Round-Robin calculation; priority based scheduling and so on. These conventional schemes posed drawbacks of resulting into more execution time and reduced throughput. The optimized scheduling of the individual tasks in cloud is still an issue to understand. Since cloud comprises of a number of distinct resources and cost of performing tasks in cloud also varies so the scheduling of tasks in cloud is very different from the traditional methods of scheduling. Cloud task scheduling is performed by choosing the best possible resource available for execution of tasks in order to minimize the completion time. Usually, in scheduling algorithms a list of tasks is structured by assigning a priority to the tasks following any parameter. Tasks are further chosen according to their priorities and are assigned to the available resources to meet the pre-defined objectives.

A large portion of the task scheduling algorithms in cloud computing are single-objective which does not characterize productive resource usage. Many real-world design problems nowadays involve the synchronous optimization of multiple objectives. These objectives such as reducing the time, expense, increasing performance etc. often clash with each other which implies that improvement on one objective value leads to the degradation of another. Instead of seeking for a single optimal solution with respect to one objective, multi-objective optimization aims at seeking for the set of so called Pareto-optimal solutions. Thus, it is critical to deliver multi criteria to improve the framework execution and expand asset use. There are many criteria like execution time, cost, bandwidth for communication, deadline, makespan etc. The paper proposes an efficient multi-objective task scheduling algorithm taking processing time, average waiting time and processing cost as its three criteria. The proposed algorithm also integrated with non-dominated sorting for ordering of tasks.

4. PROPOSED APPROACH

Since millions of users get the advantage of the cloud services by submitting their millions of computing tasks to cloud computing, hence, the scheduling of these millions of tasks is a very essential and challenging job for cloud. The main motive of task scheduling is to attain better cloud performance in terms of better throughput, load balancing, quality of service (QoS), economic feasibility and the optimal operation time. Task scheduling can be viewed from two directions- from the cloud resources users' view, users have to identify which cloud computing resource can meet their job QoS requirements for computing and how much amount to be paid for the cloud computing resources. While from the cloud computing service providers view, its concern is to gain the maximum profits by offering cloud computing resources, apart from meeting the user's job QoS requirements. Scheduling process in cloud completes in three steps known as Resource discovering or filtering, Resource selection, and Task allocation. In Resource discovering or filtering, the datacenter broker discovers the available resources in the network system and collects status information about them. In Resource selection the target resource is selected according to the requirements of task and resource. In task allocation, the task is allocated to selected resource.

The most important entity in cloud computing is the broker who act as an interface between users and cloud service provider for mediating their interactions. The broker's responsibility includes maintaining a list of virtual machines (VMs) and the corresponding QoS. User sends the request to the cloud broker and cloud broker further forwards the request to the VM server. After selecting the proper VM that meet the user's requirement and Service level agreement (SLA), broker binds the task to that particular VM.

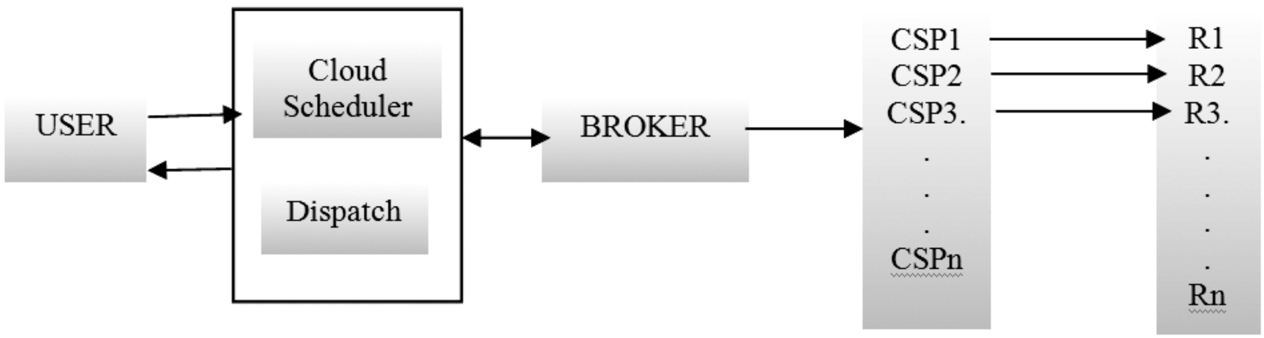


Figure 2: Task scheduling in cloud

The proposed scheduling algorithm considers the QoS parameter for assigning the priorities to the tasks. Low QoS worth is assigned to a High QoS task and vice versa. Hence, the task with lower QoS worth is a high priority and the task with high QoS worth is a low priority. The QoS parameters for tasks are mentioned in SLA. Here, the cloud broker sends request to the cloud service provider for list of VMs created in the datacenters. After getting the list in response from Cloud Service Provider, cloud broker assigns QoS to the VMs using Millions of instructions per second (MIPS) of a VM and the granularity size.

The proposed scheduling algorithm is involving Non-dominated sorting which targets the multi-objective issues considering multiple criteria. In the proposed work, the foremost objective is to minimize the execution time of task. Fundamental objective is accomplished by selecting a task with least task size and least (low) QoS worth. The two target capacities are as per the following.

$$\text{Minimize } f(S_k) = S_k | \ \forall\ j\ \exists\ i,\ f(S_i) \leq f(S_j) \tag{1}$$

$$\text{Minimize } f(Q_k) = Q_k | \ \forall\ j\ \exists\ i,\ f(Q_i) \leq f(Q_j) \tag{2}$$

$S, Q \in T\ (ID, Q, S)$

$i = \{1, 2, 3....n\},\ j = \{1, 2, 3....n\},\ k = \{1, 2, 3....n\}$

Where, S is size of the task and Q is the assignment's QoS worth, T is the arrangement of task and n is the quantity of task. In non-dominated sorting, various goals are applied at time. The proposed algorithm is using length of task (cloudlet length) and output file size (cloudlet fileoutputsize) as function parameters in non-dominated sorting.

The list of VMs is sorted according to the MIPS value and the granularity size in descending order with first position occupied by VM with high QoS and the rear position occupied by low QoS VM. After non-dominated sorting, the resultant non-dominated task's set is bound with the VMs. The process of binding of final tasks to VM is done following a sequential manner considering both the list of tasks and VMs such that the first VM from the VM's list is bound to the first task in the task's list and so on. After reaching the last VM, the following task is again submitted to the first VM and the process is repeated.

Algorithm: Multi-Objective Scheduling algorithm

---

1) Initialize the virtual machines list

2) Initialize the task list.

3) Sort the virtual machines using QOS parameters (MIPS and Granularity size).

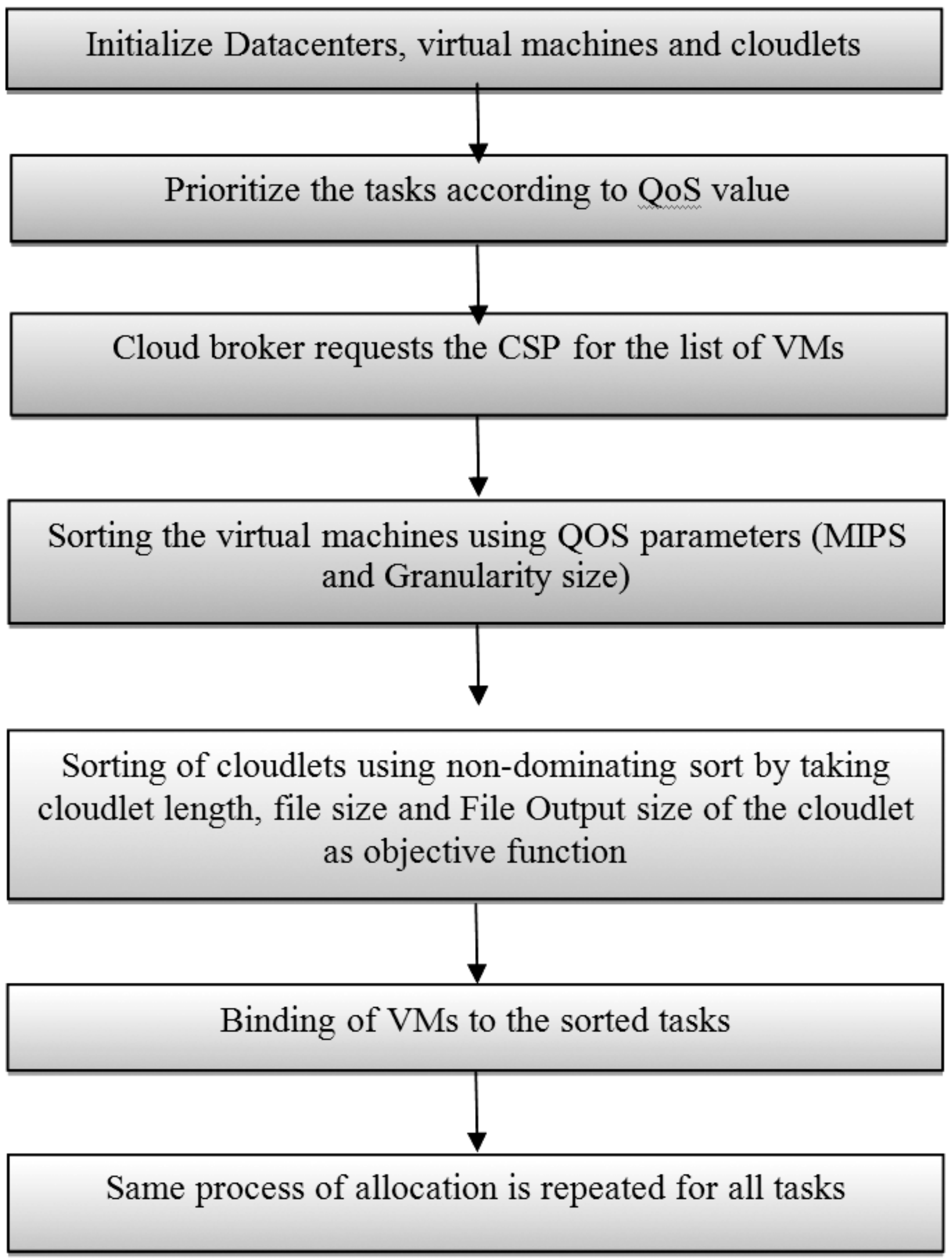
4) Sort the task list using non-dominating Sorting (considering cloudlet length and cloudlet file outsize as function parameters)

5) Assigning the task to the virtual machines in groups. Like if there are 30 tasks and we have 10 virtual machines the assigning each virtual machine 3 tasks

6) This process of allocation will be repeated for all tasks.

```
Non-dominated sorting (task list)
i=0
Create empty non-dominated list
Initially put task (i) into non-dominated list
    for all i=1 to size of task list
        for all j=0 to size of non-dominated list
            if task (j) dominates task (i) i.e. checking the task length
                and task Output file size.
                    Put task (j) into non-dominated set
            else
            if task (i) dominates task (j)then
                Put task (i) into non-dominated set
            else
            Put task (i) and task (j) into non-dominated set
            end if
        end for
    end for.
```

The step by step working of the proposed algorithm is described as:



5. EXPERIMENTS AND RESULTS

This section presents the simulation results of the proposed methodology implemented with the help CloudSim 3.0.2 simulator on windows 7 OS with Core i3 2.10GHz processor. NetBeans IDE 8.0 is used to run CloudSim 3.0.2. The proposed task scheduling algorithm is implemented by taking different datasets of machines and tasks. The performance of the proposed algorithm is evaluated against the conventional FCFS and SJF algorithms and previously existing multi-objective task scheduling algorithm. Three parameters include- the Processing time, Average waiting time of jobs and the Processing Cost.

Table 1
Various Parameters for analysis of the results.

Parameter	*Formulation*
Processing time	CloudletLength / vmMips*vmNumberOfPes
Cost	characteristics.getCostPerMem * vm.getRam
Waitingtime	cloudlet.WaitingTime
ResponseTime	cloudletSubmissionTime-cloudletFinishTime
ExecutionTime	cloudletExecStartTime-cloudletFinishTime

Table 2
Processing time observed in the simulation result

Workload	*Virtual Machines*	*cloudlets*	*Proposed Technique*	*Existing Algorithm*	*SJF*	*FCFS*
WorkLoad1	20	200	1730290.67	1761747.72	1853562.156	1953562.156
WorkLoad2	30	300	3659123.39	3953071.77	4659123.39	5503923.105
WorkLoad3	50	500	9106072.75	9226071.75	9327016.035	9552701.035

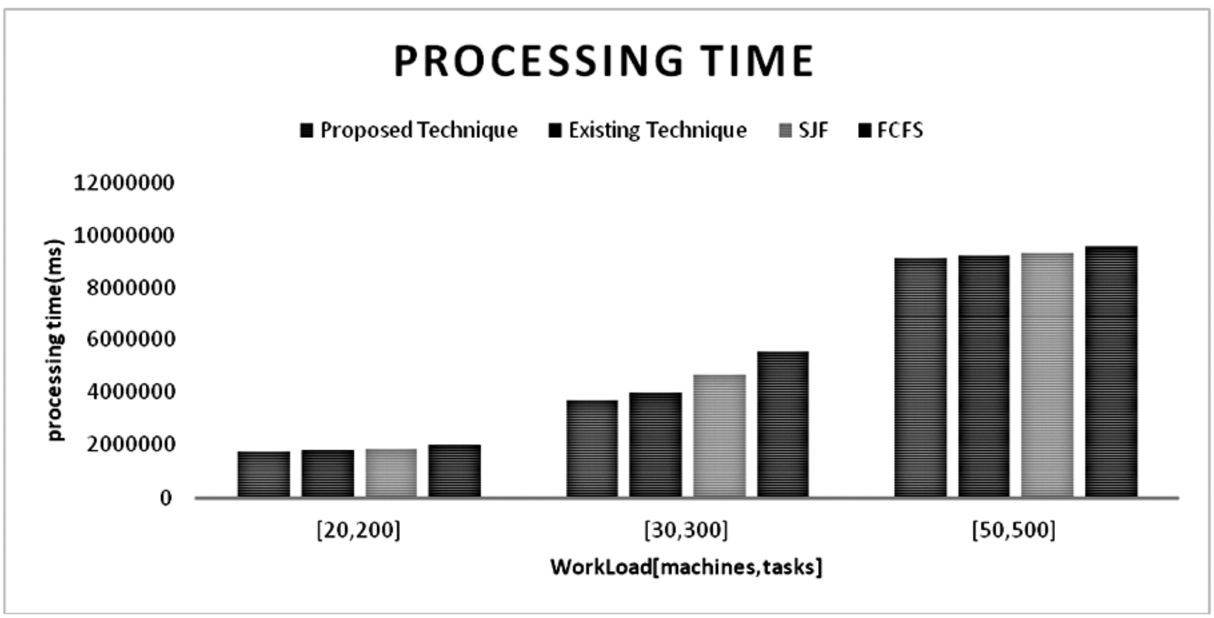


Figure 3: Comparison of Total Processing Time of proposed Scheduling algorithm with SJF, FCFS and previously Existing Algorithm of Lakra & Yadav[10].

Table 3
Average waiting time observed in the result.

Workload	*Virtual Machines*	*cloudlets*	*Proposed Technique*	*Existing Technique*	*SJF*	*FCFS*
WorkLoad1	20	200	0.4927	0.514	0.812	1.582959766
WorkLoad2	30	300	0.495	0.522	0.835	1.488041272
WorkLoad3	50	500	0.49709	0.4976	0.864	1.418611312

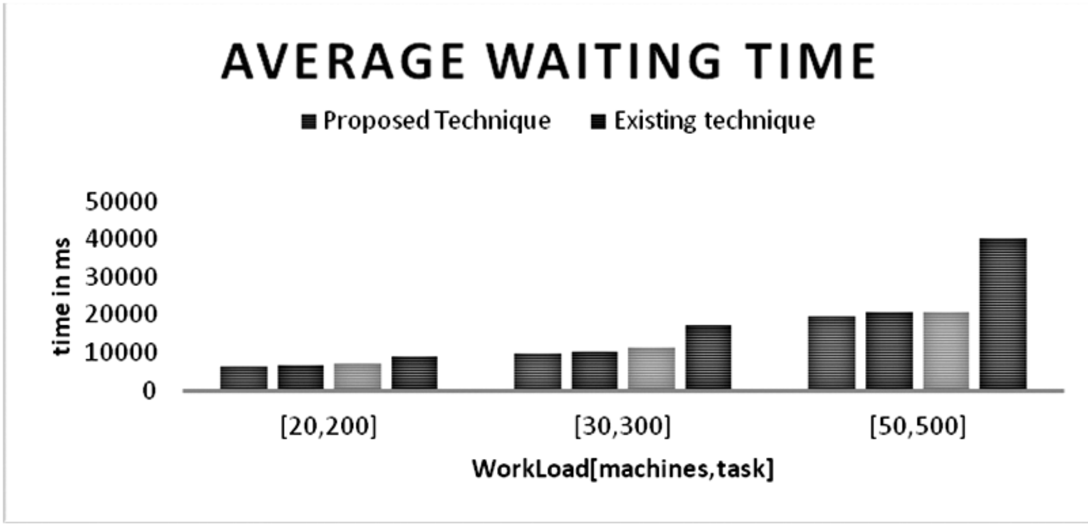


Figure 4: Comparison of Average Waiting Time of Proposed Scheduling algorithm with SJF, FCFS and previously Existing Algorithm of Lakra & Yadav[10].

Table 4
Processing Cost observed in the result.

Workload	*Virtual Machines*	*Cloudlets*	*Proposed Technique*	*Existing Technique*	*SJF*	*FCFS*
WorkLoad1	20	200	6443.69	6668.699	6994	9011.69
WorkLoad2	30	300	9840	10003.05	11203.05	17032.05
WorkLoad3	50	500	19584.55	20584.55	20600.56	40101.75

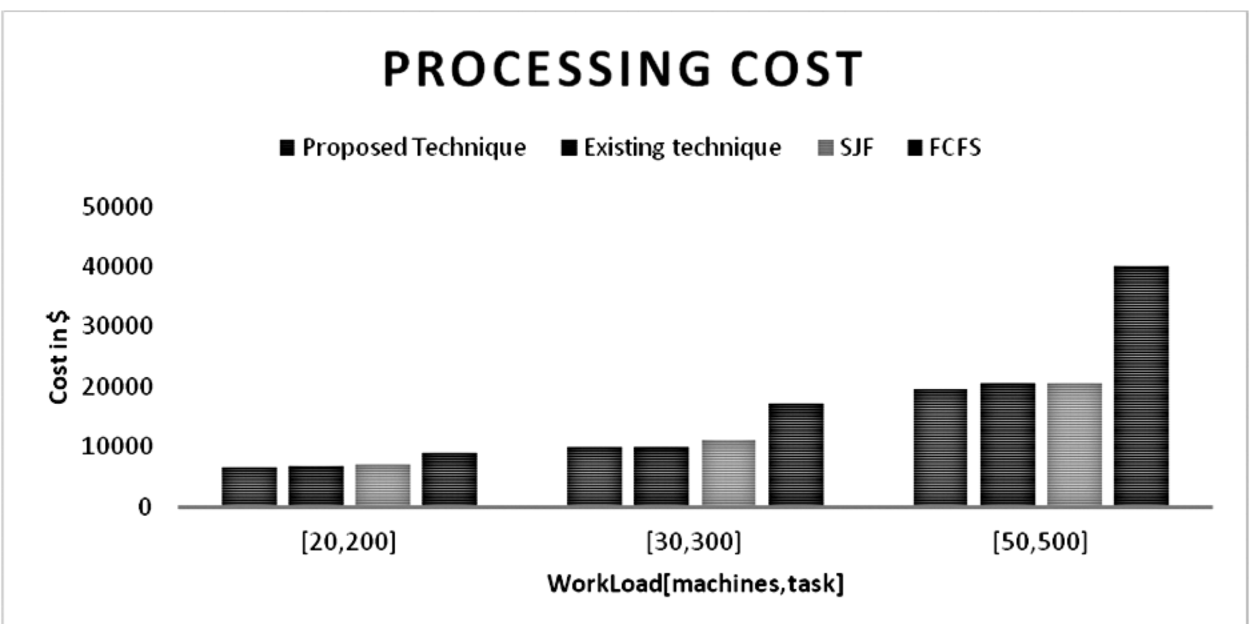


Figure 5: Comparison of Total Processing cost of proposed Scheduling algorithm with SJF, FCFS and previously Existing Algorithm of Lakra & Yadav[10].

From the comparison of these four task scheduling algorithm it is observed that the proposed algorithm performs better other than the three algorithms with less processing time, average waiting time and processing cost of cloud computing system.

6. CONCLUSION AND FUTURE SCOPE

Scheduling of task is one of the most challenging issue in cloud computing environment. The proposed task scheduling algorithm for cloud computing environment is based on multi objective optimization. This algorithm use non-dominated sorting for task ordering. The proposed algorithm has been simulated and the results are compared with FCFS and SJF algorithms and previously implemented multi objective task scheduling algorithm [10].The results reveal that the proposed algorithm results into performance gain according to Average waiting time, processing time and processing cost. The proposed algorithm can be improved by taking consideration of some other QOS parameters. As cloud computing works in real time and single criteria based algorithm may not be the one for task scheduling. So this algorithm can perform even better if the tasks can be scheduled using various QOS parameters and hence further the results can be improved and better scheduling with improved parameters can be achieved on Cloud. It can further be merged with energy aware task scheduling technique.

REFERENCES

[1] Agarwal.A, Jain.S, "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment", International Journal of Computer Trends and Technology (IJCTT)– ISSN: 2231-2803, volume 9 number 7– Mar 2014

[2] Singla.N, Bawa.S, "Priority Scheduling Algorithm with Fault Tolerance in Cloud Computing", Volume 3, Issue 12, December 2013 ISSN: 2277 128X, International Journal of Advanced Research in Computer Science and Software Engineering

[3] Beloglazov.A , Abawajy.J, Buyya.R ," Energy-aware resource allocation heuristics for efficient management of datacenters for cloud computing ," Future Generation Computer Systems , pp.755-768,2012.

[4] Katsaros.G, Subirats.J, Oriol Fitó.J, Guitart.J, Gilet.P, Espling.D, "A service framework for energy-aware monitoring and VM management in Clouds", 0167-739X, 2012 Elsevier

[5] Chia-Ming Wu, Ruay-Shiung Chang, Hsin-Yu Chan, "A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters", 0167-739X, 2013 Elsevier

[6] Sampaioa.A.M, Barbosa.J.G, "Towards high-available and energy-efficient virtual computing environments in the cloud", 0167-739X, 2014 Elsevier

[7] Shena.D, Luo.J, Donga.F, Fei.X , Wanga.W, Jin.G, Li.W, "Stochastic modeling of dynamic right-sizing for energy-efficiency in cloud data centers", 0167-739X 2014 Elsevier

[8] Hsu.C.H, Slagter.K.D, Chen.S.C, Chung.Y.C, "Optimizing energy consumption with task consolidation in clouds", 0020-0255, 2012 Elsevier

[9] Xiao.P, Hub.Z, Liu.D, Zhang.X, Qua.X, "Energy-efficiency enhanced virtual machine scheduling policy for mixed workloads in cloud environments", 0045-7906/ 2014 Elsevier

[10] Lakra.A.V , Yadav.D.K ," Multi-Objective Tasks Scheduling Algorithm for Cloud Computing Throughput Optimization ," International Conference on Intelligent Computing , Communication & Convergence , 2015.

[11] Zhang.J, Hu.X,Tan.X , Zhong.J.H and ,Huang.Q ," Implementation of an Ant Colony Optimization technique for job shop scheduling problem", Transc. Of the Institute of Measurement and Control, pp.93-108,2006.

[12] Foster.I , Zhao.Y , Raicu.I , and Lu.S ," Cloud Computing and Grid Computing 360 –degree compared ," Proceedings of the Grid Computing Environments Workshop , pp. 1-10,2008.

[13] Ghanbari.S , Othman.M ," A priority based Job Scheduling Algorithm in Cloud Computing ," International Conference on Advances Science and Contemporary Engineering ,2012 .

[14] Zhan.S.B , Huo.H.Y ," Improved PSO-based Task Scheduling Algorithm in cloud computing ," In: Journal of Information and Computational Science , vol. 9 , pp.3821-3829,2012.

[15] Patel.S.J , Bhoi.U.R ," Improved Priority job Scheduling Algorithm in Cloud Computing using Iterative Method ," Fourth International Conference on Advances in Computing and Communication , pp.199-202,2014.

[16] Guo.L, " Task Scheduling Optimization in Cloud computing Based on Heuristic Algorithm ," Journal of Networks , vol.7, pp.547-553,2012.

[17] Singh.S ,Kalra.M ," Task Scheduling Optimization for independent Tasks in Cloud Computing using Enhanced Genetic Algorithm ," International Journal of Apllication or Innovation in Engineering & Management , vol.3, pp. 286-291,2014.

[18] Mathukiya.E.S, Gohel.P.V ," Efficient QoS Based Tasks Scheduling using Multi- Objective Optimization for Cloud Computing," International Journal of Innoative Research in Computer and Communication Engineering ," vol.3,pp.7169-7173,2015.

[19] Singh.R.M, Paul.S , Kumar.A ," Task Scheduling in Cloud Computing : Review ," International Journal of Computer Science and Information Technologies ,vol.5,pp.7940-7944,2014.

[20] Lawrance.H, Silar.S ," Efficient QoS Based Resource Scheduling using PAPRIKA Method for Cloud Computing ," International Journal of Engineering Science and Technology , vol.5,pp.638-643,2013.

[21] Xiao.P,Hu.Z,Liu.D,Zhang.X and Qu.X ," Energy-efficiency enhanced virtual machine scheduling poling for mixed workloads in cloud environments ," Computers and Electrical Engineering ,pp.1650-1665,2014.

[22] Abdullah.M , Othman.M ," Cost-Based Multi-QoS Job Scheduling using Divisible Load Theory in Cloud Computing ," International Conference on Computational Science , pp.928-935,2013.

[23] Selvarani.S , Sadhasivam.G ," Improved Cost-Based Algorithm for Task Scheduling in Cloud Computing ," IEEE ,2010

[24] Kanani.B and Maniyar.B," Review on Max-min Task Scheduling Algorithm for Cloud Computing ," Journal of Emerging Technologies and Innovative Research ,vol. 2 , pp. 781-784 ,2015.

[25] Tripathy. L , Patra.R.R ," Scheduling in Cloud Computing ," International Journal on Cloud Computing : Services and Architecture, vol.4, pp.21-27,2014.

[26] Kowsik , K. RajaKumari ," A Comparitive Study on Various Scheduling Algorithms in Cloud Environment , " International Journal of Innovative Research in Computer and Communication Engineering , vol.2 , 2014.

[27] Wu. X, Deng.M,Zhang.R,Zeng.band Zhou.S," A Task Scheduling Algorithm based on QoS– driven in Cloud Computing ," Information Technology and Quantitative Management ,pp.1162-1169,2013.

[28] Bhatia.A and Sharma.R, " An Analysis Report of Workflow Scheduling Algorithm for Cloud Environment ," International Journal of Computer Applications, vol.119, pp. 21-25 ,2015.

[29] Bini.B.T and Sindhu.S ," Scheduling In Cloud Based On Hyper-Heuristics ," International Journal for Research in Applied Science & Engineering Technology , vol.3 , pp. 380-383 ,2015.

[30] Cho.K.M, Tsai.P.W, Tsai.C.W ," A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing," Neural Computing and Applications , 2014.