

Priority Intensed Meta Task Scheduling Algorithm for Multi Cloud Environment (PIMTSA)

B.J. Hubert Shanthan¹, L. Arockiam², A. Cecil Donald³, A. Dalvin Vinoth Kumar⁴, R. Stephen⁵

^{1,3,4,5}Assistant Professor, Department of Computer Science, Kristu Jayanti College, Bengaluru.

²Associate Professor, Department of Computer Science, St. Joseph's College, Tiruchirappalli.

Abstract. Cloud Computing plays a momentous role for the speedy computation and provides resources to the user on-demand. The conventional single cloud has assorted number of confrontations and they are named as vendor lock-in, resource un-availability, latency, etc. To resolve these conflicts, multi cloud computing is utilized to provide heterogeneous services to the cloud users on demand. Task scheduling is a mechanism used to assign user requests to the cloud server. The proposed PIMTSA algorithm works well for the independent batch of tasks in the static environment. The algorithm is constituted to miniaturize the makespan of the multi cloud systems. It is designed for the batch mode offline multi cloud systems. The priority based approach works to cater the needs of the cloud user. High priority and low priority tasks are the two different priority levels defined by the user. High priority tasks are given more importance and are executed in the high speed VM with minimum completion time. But, the High priority tasks with maximum execution time are given more priority. After the execution of the high priority tasks, the low priority tasks with minimum execution time are allocated to VMs with minimum completion time in the clouds. This algorithm provides a high reliable and scalable computing service to the user with minimum time. In future this work can be extended for dynamic multi cloud environment.

Keywords— Multi cloud, PIMTSA, Makespan, Task Scheduling.

1. Introduction

Cloud Computing plays a vital role in the modern era of computing Technology [1]. This leads for the transformation of the computation technologies from the mainframe to the client-server architecture. The computing devices of the cloud such as hardware, software and other information are shared to the customers in the form of services. Based on the Quality of Service (QoS) constraints, the computational resources are provided to the client on-demand. The cloud is a pay per usage model and the customers need to pay for the services. The cloud services are managed and owned by the Cloud Service Providers (CSPs) [2]. The cloud services used by the clients who are called Cloud Service Consumers (CSCs) [3]. The users with high speed internet connection can access the cloud service. The Service Level Agreement (SLA) [4] is legal negotiation which is established between the service providers and the consumers. The costs are charged based on SLA. Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) are three major service models in the cloud [5]. Software as a Service (SaaS) comprised of software and applications offered by the service provider that cater the needs of the user which do not require any installation on the local machine. Platform as a Service (PaaS) is an environment in which the cloud customer creates applications which are deployed as cloud infrastructure. Infrastructure as a Service (IaaS) is a service which provides the required network, computing and storage resources so that a consumer can deploy and run arbitrary softwares which include applications and operating systems. There are four types of cloud computing namely public Cloud, private Cloud, multi cloud and community cloud [6]. Public Cloud is an infrastructure which is made available to the general public or to a large industry. The private cloud services offer security, stability, privacy and data persistency.

Multi Cloud is a combination of two or more clouds like private, public or community cloud. Community Cloud is a cloud model which is supported by a specific community which is shared and managed by several organizations. Multi Cloud is an evolutionary development of the cloud computing [7]. It is a present day access in which an organization or IT company uses multiple services from more than one cloud service provider.



Gartner defines Multi Cloud as [8] “The pool of service providers of cloud to reduce dependency single vendor lock in and to mitigate against disasters”. Scheduling has a momentous role in improving the scalability and viability of the multi cloud systems. The efficient task based scheduling algorithm must increase the throughput and reduce the makespan. The priority must be considered as important metric in the scheduling algorithm [9]. The existing priority scheduling algorithms mainly focus on the smaller tasks having minimum execution time, whereas the waiting time of the larger tasks are relatively high. Therefore, it is challenging to design an efficient priority based independent task based scheduling algorithms for larger tasks in the multi cloud system.

The paper focuses more on the priority Intense task Scheduling algorithms for independent multi cloud computing systems. The proposed PIMTSA algorithm works well for the independent batch of tasks in the static environment. The PIMTSA algorithm is designed to minimize the maximum completion time of the tasks in the multi cloud systems. The objective of this paper is to scale down the overall maximum completion time of the virtual machines and to propound a Priority Intense Meta Task Scheduling Algorithm for multi cloud systems. The paper is comprised of sub sections follows. Section 2 analyses the earlier works. Section 3 describes the proposed PIMTSA. Section 4 discusses the results and interpretations of the proposed algorithm. Finally, section 5 includes the conclusion with future research directions.

2. Related Works

Cao et al., [10] proposed a task based scheduling algorithm based on the cost and the priority. The algorithm divides the priority into three levels namely high, mid and low. The system level priorities were given for the user defined tasks. The algorithm comprised of two phases. In the first phase, the priority levels were calculated and the tasks were queued into the list. In the second phase, the high priority tasks were scheduled first, mid priority list of tasks were scheduled and finally the low priority tasks were executed in the virtual machines. The user priorities for larger tasks were ignored in this mechanism. Selvarani et al., [11] developed a cost based independent task priority scheduling algorithm for cloud. The algorithm was based on the three phases. The first phase was used to calculate the priority of each task. The second phase was used to arrange the tasks based on three levels such as high, medium and low. The final phase was used to group the tasks based on their priority levels. The tasks with smaller MI (Millions Instructions) were given high priority and they were allocated to the VM. The next maximum MI was given medium priority. The execution cost is low but the time consumption of this algorithm is high.

Sandana et al., [12] proposed a priority based task scheduling algorithm based on fuzzy logic. The algorithm was designed to improve the performance of the existing standard min-min and max-min algorithms. The priority rule was designed by the fuzzy inference system. The priority was given based on the user and the maximum execution time of the virtual machines. The Mandomi inference rule was framed in the fuzzification process. Fuzzy based methods are used for the uncertainty tasks.

Naoufal Er-raji et al., [13] proposed a priority based task scheduling strategy for the multiple data centres that were available in the multi cloud computing. The priority scheduling strategy was designed based on these three parameters namely task age, task size and task deadline. The task age was computed based on the waiting time of the tasks. Task sizes depend on the task processing capability and it is termed as MI (Million Instructions). Task deadlines were given by the user. The approach was not simulated and tested in the real world multi cloud environment.

Pankajdeep Kauret al., [14] proffered a priority based task scheduling algorithm to increase the faster completion rate of the cloud server. The priority rules were framed under constraints such as user level of the priority, urgency of the task and queuing time of the tasks.

Xiaofang et al., [15] enhanced the max-min algorithm for cloud systems. The algorithm emphasized for the tasks having larger length. Makespan and resource utilization were the two major parameters considered in this algorithm. Task pending time and task response time from the cloud servers to the

users were also considered in this approach. Huankai Chen et al., [16] proposed a user based priority min-min algorithm for Meta tasks in the cloud environment. The user priority task was grouped as VIP users with high priority and the normal user with low priority. The VIP users were executed in the high speed virtual machine with minimum completion time. The low prioritized tasks were selected and assigned to the idle virtual machines in the cloud. Load balancing constraints are not considered prior to scheduling the tasks.

George et al., [17] developed a priority based Meta task scheduling algorithm for cloud environment. The algorithm comprised of two phases. In the first phase, the users set priority were given as the inputs along with the ETC matrix. The ETC matrix was generated based on the EMGEN tool. There were two levels of tasks priority. High priority tasks and Low priority tasks. The high priority tasks from the ETC matrix were separately arranged in the ascending order. The sorted tasks were allocated to the VMs with minimum completion time. The high priority tasks follow the min-min scheduling policy for the allocation and execution of the tasks. Low priority tasks were sorted in the descending order. The sorted tasks were allocated to the VM with minimum completion time.

Pariksh et al., [18] developed a double level priority based scheduling algorithm for independent and dependent tasks. There were two levels of priority used in this algorithm. The first level was used to collect the tasks and the collected tasks were grouped for the execution process. The grouped tasks were placed in the full available category if all the resources were available in the single data center.

The grouped tasks were placed in the partial available category if the required resources were available in more than single data center. The fully available category resources were stated as dependent tasks list. The partial available resources were stated as independent tasks list.

Shamsollah Ghanbari et al., [19] designed a priority based task scheduling algorithm for the heterogeneous cloud systems. It worked under the mechanism of multi criteria decision strategy and comprised of three levels: objective level, attributes level and alternative level. The analytical hierarchy processes (AHP) were used as an attribute for the multi criteria decision. The Consistent comparative matrixes (CCM) were formed as the basis for the AHP. The CCM was computed based on the criteria and attributes specified by the user. Every task requests for VM was determined by the priority. The priority vector was computed based on the accessibility of the VM resources. The maximum computed priority vector was treated as high priority tasks which were allocated to the high speed VM with minimum completion time. Suresh Kumar et al., [20] articulated priority based load balanced scheduling algorithm for cloud systems. This algorithm reduced the maximum completion time of tasks and improved the resource utilization rate of the server. The priority tasks were allocated to the available VMs in the cloud. Opportunistic Load Balancing (OLB) technique was used for the priority tasks. This approach made the VM busy and idle time of the VM was minimized in the approach.

3. Priority Intensed Meta Task Scheduling Algorithm (PIMTSA)

The algorithm is designed to provide the cloud resources to the consumer with minimum time in the multi cloud environment. The ETC is a matrix representation where the number of tasks to be scheduled in the available VMs. The ETC matrix value consists of the earliest completion time of the tasks in the virtual machines (VMs) of the clouds. The ETC matrix comprises of execution time of the tasks given by the user to the virtual machines. The formula for ETC calculation is given by the equation (3.1)

$$ETC_{ij} = IS_i / PS_j \quad \dots (3.1)$$

where, IS_i denotes Millions of Instructions (MI) which is also known as task length and PS_j denotes Millions of Instructions per second (MIPS) also known as virtual machine processing speed. The IS_i denotes the set of i th independent tasks and PS_j denotes the set of processing speed of the j th virtual machines in the clouds. The completion time of each task is computed by the sum of expected

execution time and the virtual machine ready time. The formula for completion time is given below in the equation (3.2).

$$CT_{ij} = ETC_{ij} + RT_i \quad \dots (3.2)$$

where, ETC_{ij} denotes the ETC matrix, RT_i denotes the ready time of each virtual machine available in the multi cloud. The algorithm works under batch mode scheduling mechanism. The Figure 3.1 depicts the methodology of PIMTSA Algorithm.

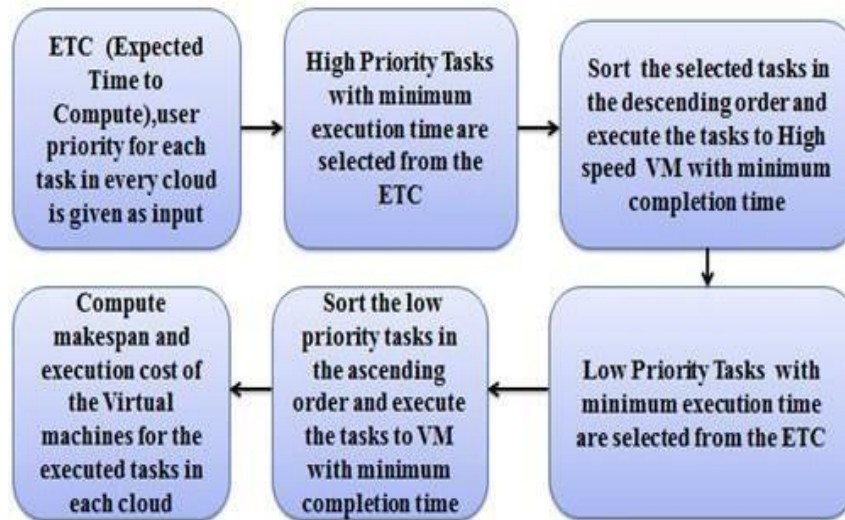


Fig 1: Methodology of PIMTSA Algorithm

The ETC matrix is generated in the cloud manager and it acts as scheduler to allocate the tasks to the available virtual machines (VMs) in the clouds. The ETC (Expected Time to Compute) matrix and the priorities set by the user for each task are given as input. High priority and low priority are the two different priority levels used in this algorithm. There are two different types of virtual machines used in this algorithm, namely high speed VMs and low speed VMs. The High speed VMs complete the tasks with minimum time which require more cost. The Low speed VMs complete the tasks with maximum time with low cost. The algorithm emphasizes the usage of high speed VMs for execution of the tasks.

The high priority tasks with minimum earliest completion time are selected from the ETC matrix. The selected tasks are sorted in the descending order. After sorting, the tasks with maximum execution time are selected and allocated to the virtual machines with minimum completion time. The ready time and completion time for all the high priority tasks are updated frequently. After the execution of the high priority tasks, the low priority tasks with minimum earliest completion time are selected from the ETC matrix. The selected tasks are arranged in non-decreasing order. The tasks with minimum execution time are selected and allocated to the virtual machines (VMs) with minimum completion time.

The Algorithm pseudo code of the PIMTSA is elaborated as follows

Algorithm: PIMTSA

Input: ETC matrix: **Output:** Makespan

Step 1: Start.

Step 2: Read the values of ETC matrix.

Step 3: Read the tasks with High Priority from the ETC.

Step 4: Check whether the total number of tasks and Virtual machines are not empty.

Step 5: Arrange the High priority tasks in the descending order.

Step 6: Select the tasks and match the tasks to the VMs with minimum completion time for each cloud from High Priority tasks.

Step 7: Arrange the Low Priority Tasks in the ascending order.

Step 8: Select the tasks and match the selected tasks with minimum completion time for each cloud from Low Priority tasks

Step 9: Update the completion time of the virtual machines and the clouds

4. RESULTS AND DISCUSSIONS

The proposed PIMTSA algorithm is tested by using CloudSim [21]. The ETC (Expected time to Compute) matrix is generated for the user tasks from the available virtual machines in the clouds. The Table 1 represents the sample ETC matrix for 10 tasks, 4 virtual machines and 2 clouds in which T1, T3, T5, T7 & T10 belongs to high priority users. The cloud C1 represents Amazon EC2 [22] cloud and the cloud C2 represents Google cloud console [22]. The offline batch mode scheduling strategy is followed in this algorithm. In this strategy, all the tasks are executed parallel in the available virtual machines.

Table 1: ETC Matrix

Cloud	Virtual Machines	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
C1	VM 1	7	6	4	4	4	4	3	5	3	6
	VM 2	11	14	9	8	6	8	14	7	10	8
C2	VM 3	9	9	6	5	5	6	8	6	5	7
	VM 4	13	15	10	13	8	14	15	10	11	9

The Table 2 represents the results of task allocation of high priority and low priority tasks executed in the VM with minimum completion time.

Table 2: Allocated Tasks

Cloud	Virtual Machines	High priority Tasks					Low priority Tasks				
		T1	T10	T3	T5	T7	T9	T4	T6	T8	T2
C1	VM 1	7				3		4			6
	VM 2			9						7	
C2	VM 3		7				5		6		
	VM 4				8						

The Table 3 represents the results of the PIMTSA algorithm for 5 high priority tasks and remaining 5 tasks are low priority tasks.

Table 3: Results of PIMTSA task allocation

Cloud	Virtual Machines	T1	T10	T3	T5	T7	T9	T4	T6	T8	T2
C1	VM 1	7				3		4			6
	VM 2			9						7	
C2	VM 3		7				5		6		
	VM 4				8						

Figure 2 depicts the comparative analysis of the makespan values with the existing min-min algorithm. The results are tested for 4 virtual machines and 2 clouds with the tasks ranging from 10, 20, 50, 100 and 400.

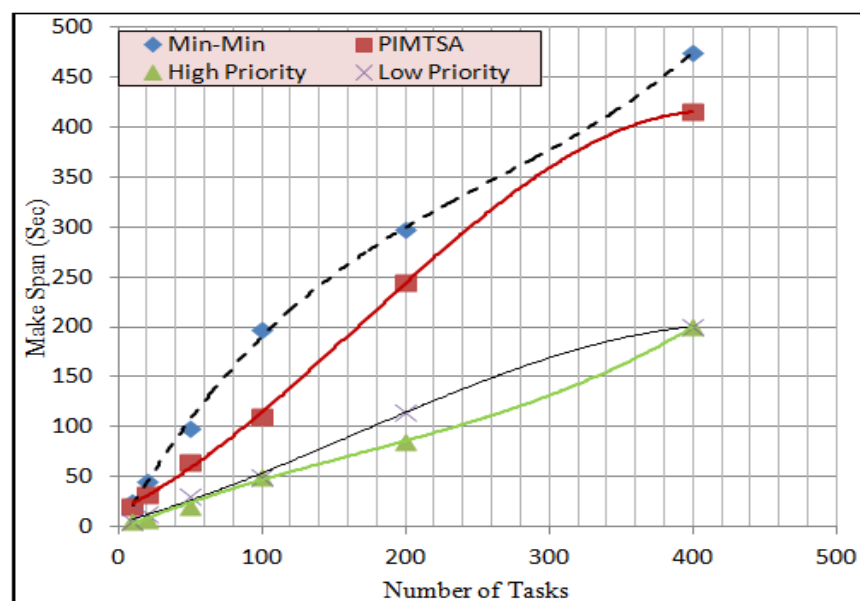
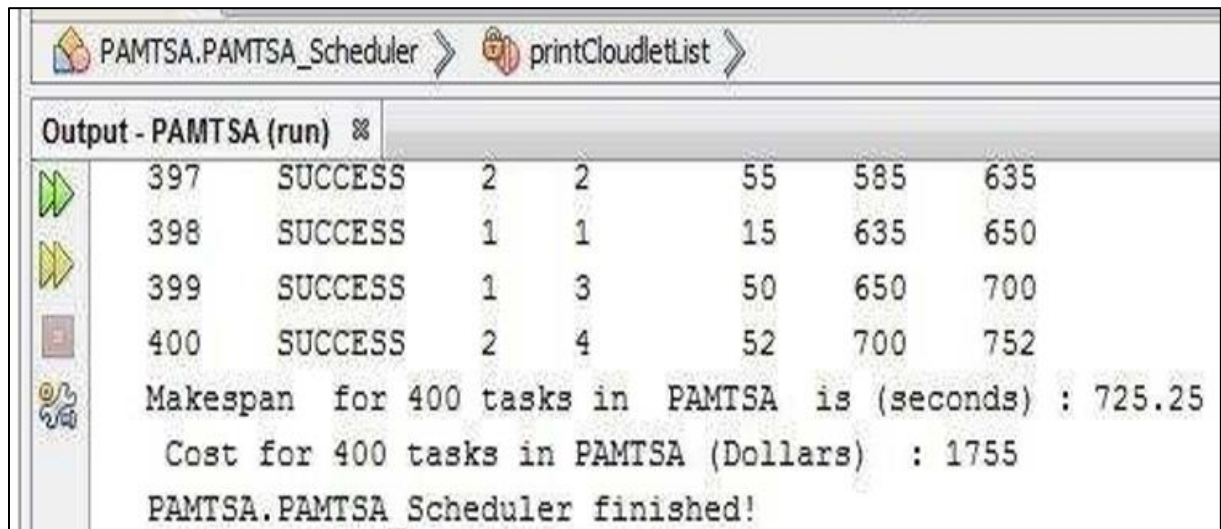


Fig. 2: Comparative Analysis of Makespan Values

The figure 3 clearly illustrates the screenshots for the output results of the PIMTSA Algorithm. The algorithm is tested for 400 tasks in the CloudSim. The simulation result clearly proves that the proposed PIMTSA algorithm outperforms the existing min-min algorithm. The makespan values are reduced in the proposed algorithms and it emphasizes the larger tasks.



The screenshot shows a window titled 'PAMTSA.PAMTSA_Scheduler' with a 'printCloudletList' button. Below the title bar is a table of simulation results. The table has columns for task ID, status, and various performance metrics. The results for tasks 397 to 400 are shown, all with a 'SUCCESS' status. Below the table, the makespan and cost for 400 tasks are displayed, followed by a message that the scheduler has finished.

Output - PAMTSA (run) %							
397	SUCCESS	2	2	55	585	635	
398	SUCCESS	1	1	15	635	650	
399	SUCCESS	1	3	50	650	700	
400	SUCCESS	2	4	52	700	752	
Makespan for 400 tasks in PAMTSA is (seconds) : 725.25							
Cost for 400 tasks in PAMTSA (Dollars) : 1755							
PAMTSA.PAMTSA Scheduler finished!							

Fig. 3: Simulation Results of PIMTSA

5. Conclusion

A Priority Intense Meta Task Scheduling Algorithm (PIMTSA) has been proposed for the independent tasks in the multi cloud environment. The proposed algorithm is designed for the batch mode offline multi cloud systems. The priority based approach works to cater the needs of the cloud user. In the existing min-min Meta task scheduling algorithm, the user priority is not considered for multi cloud environment. The waiting time for the larger tasks is extremely high in the min-min algorithm. To overcome this conflict, the PIMTSA algorithm is proposed to reduce the overall maximum completion time of the tasks. This algorithm provides a high reliable and scalable computing service to the user with minimum time. In future this algorithm might be applied for dependent tasks for increasing the cloud server resource utilization ratio.

References

- [1] Buyya, James Broberg and Andrzej Goscinski, "Cloud Computing Principles and Paradigms", Wiley and Sons Pub Pvt, 2013, pp. 12-55.
- [2] Mell and Grance, "NIST Working Definition on Cloud Computing", National Institute of Standard and Technology, 2011, pp. 1-7
- [3] Rani and Rajan, "A Comparative Study of SaaS PaaS and IaaS in Cloud Computing", International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), Vol. 4, No. 6, 2014, pp. 458-461.
- [4] Vaquero, Roderio Merino, J. Caceres and M. Lindner, "A Break in the Clouds: Towards a Cloud Definition", ACM, SIGCOMM Computer Communications Review, Vol. 39, No. 50, 2000, pp. 15-25.
- [5] BJ.Hubert Shanthan, Dr. L. Arockiam, Resource Based Load Balanced Min Min Algorithm(RBLMM) for static Meta task Scheduling in Cloud, International Conference on

- Advances in computer Science and Technology, International Journal of Engineering and Technology, Special Issue, 2018, pp.1-8.
- [6] BJ.Hubert Shanthan, Dr.L.Arockiam, "Rate aware Meta task Scheduling Algorithm for multi cloud computing (RAMTSA)", Journal of Physics: Conference Series, No.1142, doi:10.1088/1742- 6596/1142/1/012001, 2018, pp.1-10.
 - [7] BJ.Hubert Shanthan, Dr.L.Arockiam, "Spell Aware Meta Task Scheduling Algorithm (SAMTSA) for Multicloud ", International Journal of Scientific Research in Computer Science Applications and Management Studies, Vol.7, No.4, 2018, pp.1-7.
 - [8] Gartner, "Market Insight: Multicloud Becomes Essential for Cloud IaaS Offerings", <https://www.gartner.com/doc/3892385/market-insight-multicloud-essential-cloud>, accessed online 03.04.2019
 - [9] Jemina Priyadharshni, "Optimized Meta Task Scheduling Algorithms for Cloud Environment", PhD. Dis. , Bharathidasan University, Tamilnadu, 2015.
 - [10] Cao, Wei, and Gong, "An Optimized Algorithm for Task Scheduling Based on Activity Based Costing in Cloud Computing", International Conference on Bioinformatics and Biomedical Engineering, IEEE, 2009, pp. 1-3.
 - [11] Selvarani and Sadhasivam, "Improved Cost-Based Algorithm for Task Scheduling in Cloud Computing" IEEE International Conference on Computational Intelligence and Computing Research, 2010, pp. 1-5.
 - [12] Sandana Karuppan, Kumari and Sruthi, "A Priority-Based Max-Min Scheduling Algorithm for Cloud Environment Using Fuzzy Approach", International Conference on Computer Networks and Communication Technologies, Springer, 2019, pp. 819-828.
 - [13] NaoErRaji and Benabbou, "Priority Task Scheduling Strategy for Heterogeneous Multi-Datcenters in Cloud Computing", International Journal of Advanced Computer Science and Applications, Vol. 8, No. 2, 2017, pp. 272-277
 - [14] Pandee Kaur and Singh, "Priority Based Scheduling Algorithm with Fast Task Completion Rate in Cloud", Journal of Advances in Computer Science and Information Technology (ACSIT), Vol. 2, No. 10, 2015, pp. 17-20.
 - [15] Xian Li, Mao, Xiao and Zhuang, "An Improved Max-Min Task- Scheduling Algorithm for Elastic Cloud", International Symposium on Computer, Consumer and Control, IEEE, 2015, pp. 340-343.
 - [16] Huang Chen, Wang, Helian and Akanmu, "User-priority guided Min- Min Scheduling Algorithm for Load Balancing in Cloud Computing", National Conference on Parallel computing technologies (PARCOMPTECH), IEEE, 2013, pp. 1-8.
 - [17] George Amalarethnam and Kavitha, "Priority Based Performance Improved Algorithm for Meta-Task Scheduling In Cloud Environment", International Conference on Computing and Communications Technologies (ICCCT), 2017, pp. 69-73.
 - [18] Parikh and Sinha, "Double Level Priority Based Optimization Algorithm for Task Scheduling in Cloud Computing", International Journal of Computer Applications, Vol. 62, No. 20, 2013, pp. 1-13.
 - [19] Shambori Ghanbari and Othman, "Priority Based Job Scheduling Algorithm in Cloud Computing", Elsevier Procedia Engineering, Vol. 50, 2012, pp. 778-785.
 - [20] Suresh Kumar and Dharmaraj, "PBVMLBA: Priority Based Virtual Machine Load Balancing Algorithm for Cloud Computing", International Journal of Computer Science and Software Engineering, Vol. 6, No. 11, 2017, pp. 233-238.
 - [21] Sanjaya K. Panda, Prasanta K. Jana, SLA-based task scheduling algorithms for heterogeneous multi-cloud environment, Journal of Super Computing, Springer, 2017, pp.1-33.
 - [22] Elhossiny Ibrahim, Nirmeen A. El-Bahnasawy, Fatma A. Omara, Task Scheduling Algorithm in Cloud Computing Environment Based on Cloud Pricing Models, IEEE 2016 World Symposium on Computer Applications & Research, 2016, pp .65-71.