Certain Investigations on Opulence Split Algorithm and Particle Swarm Optimization Algorithm for Task Scheduling in a Virtualized Cloud Environment

Jothi KR,Kalyanaraman P, Balakrishnan P,Faraz Ahmad

Abstract: With the outstanding development in high-speed net technologies, the conception of virtualization and cloud computing has become an additional standard. The cloud framework provides advantageous and on demand access to computing resources over the web. Individual and enterprises can get access to the product resources and equipment, for example, arrange, capacity, server and applications which are found remotely effectively with the assistance of Cloud Service. The tasks/jobs that are submitted to this cloud environment need to be serviced on time exploiting the resources available in order to achieve proper resource utilization, efficiency and lesser makespan. The proposed work surveys a series of currently existing task scheduling algorithms with respect to their characteristics and come up with a new method of real time task scheduling algorithm based on equal resource split.

Keywords: Cloud platform; job scheduling; efficiency; makespan, resource.

I. INTRODUCTION

Cloud computing is the current buzzword in the Information Technology industry. With its characteristic behavior, several features and growing popularity the world seems to be currently shifting towards the cloud platform. The demand for cloud services continues to grow with the passage of time. Cloud Computing alludes to application and services that are executed on distributed networks utilizing virtual assets, regular web protocols and system administration models. In cloud computing every entity from power to infrastructure is provided to the user as a service. The services are classified under three models as follows **[1]**:-

1. **Infrastructure as a service(IaaS) :** Involves delivery of technology as an on demand scalable service. Provides access to fundamental resources like
2. physical and virtual machines as well as storage.
3. **Platform as a service (PaaS**): Provides runtime environment for application development and deployment tools. It has a highly scalable multitier architecture. It provides all applications and life cycles from the internet.

4. **Software as a service (Saas):** Allows the end user to use software applications as a service. Software delivery methodology to provide multi-tenant access to software and its function is remotely a web service.

The other model in cloud computing is the deployment model which can be stated as follows **[2]**:-

1. Public: This is the most used model of cloud computing. This is used by general public cloud consumers. The provider has its own policies on which various users agree upon and use the service. Examples from day to day life are Google Drive, AWS, Azure, etc.

2. Private: This is the kind of cloud infrastructure which is operated within a single organization for its sole purpose. This kind of cloud model is beneficial for an organization in many ways. More number of in house resources, cutting the cost for management of it by some third party and full control and access of data are some of them

3. Hybrid: This type of cloud infrastructure is combination of two or more types of cloud deployment models. This increases data and application portability. **[3]**

4. Community: Many small organizations come together to and create their own cloud infrastructure on settling down to policies applied after each of them agrees to all of them.

II. LITERATURE SURVEY

A. Background

High speed internet is giving the world many facilities which were almost impossible earlier, today they have proven to be of utmost importance in the daily life. One of them is cloud computing.

Revised Manuscript Received on April 07, 2019.
Jothi KR, School of Computer Science Engineering, Vellore Institute of Technology, Vellore, India.
KalyanaramanP, School of Computer Science Engineering, Vellore Institute of Technology, Vellore, India.
Balakrishnan P, School of Computer Science Engineering, Vellore Institute of Technology, Vellore, India.
Faraz Ahmad, School of Computer Science Engineering, Vellore Institute of Technology, Vellore, India.

It gives access to computing resources available over internet. One of its major benefits is that individuals and enterprises can access the software and hardware such as network, storage, server and applications which are located remotely easily with the help of Cloud Service. The tasks/jobs submitted to this cloud environment needs to be executed on time using the resources available so as to achieve proper resource utilization, efficiency and lesser makespan which in turn requires efficient task scheduling algorithm for proper task allocation. Cloud task scheduling is a NP complete problem. In the process of task scheduling, the users submit their jobs to the cloud scheduler. Cloud Information Service is a registry which contains the resources on the cloud like the data centre and hosts or virtual machines. Each data centre has a host and every host has a hardware configuration like number of processing elements and RAM. The cloud scheduler is responsible for assigning the user tasks to multiple virtual machines on the basis of availability. In our proposed work, we plan to survey the existing Task Scheduling Algorithm for various performance metrics and aim to adapt the features of various existing algorithms in order to propose a model for efficient distribution and scalability characteristics of cloud resources. Currently, there is a lack of uniform standard for job scheduling in cloud computing. Resource management and job scheduling are the key technologies of cloud computing that plays a vital role in an efficient cloud resource management. Cloud computing makes collaboration simpler and can reduce platform-incompatibility problems. Since this review aims at the methods of task scheduling in cloud computing, special emphasis will be paid to the scheduling of tasks in the cloud based environment. Cloud Computing is an integral component of advanced computing systems. Computational innovations and designs have advanced in the most recent decades. Numerous angles are liable to innovative advancements. Cloud computing is a registering innovation that is quickly combining itself as the subsequent stage in the advancement of computing. In order to to gain the maximum benefit from cloud computing, attempts must be made to design mechanisms that optimize the use of architectural and deployment paradigms. The role of Virtual Machine's (VMs) has emerged as an important issue. With the advent of virtualization, cloud infrastructure is made more scalable. Therefore developing an optimal scheme for scheduling of virtual machines is an important issue. The architecture of cloud computing can be broken into three layers of models as stated above. The performance and efficiency of cloud computing services rely on the performance of the tasks submitted to the cloud environment. Scheduling of the Stasks plays significant role in improving performance of the cloud services. **[4]**

B. Existing Task Scheduling Algorithms and Related Work

Scheduling of Tasks alongside the proper allocation of resources are the most vital characteristics of the cloud computing environment which straightforwardly influence the execution of a framework. So as to accomplish high throughput and minimized time span, different task scheduling algorithms have been actualized. The adaption of the appropriate algorithm decides the performance of the system. Another concept that has to be kept in mind while designing a scheduling algorithm is the makespan. Makespan refers to the complete time taken by tasks to execute. Some existing task scheduling algorithms can be classified as follows:-

1. **Min-Min:**It involves the process of mapping a set of tasks by calculating the minimum expected time. If a particular resource is to be allocated then the task with the least time of completion is selected. It is a heuristic approach.

2. **Max-Min:** This algorithm involves the analysis of the maximum completion time of each task with respect to the available resource. The task with the maximum completion time is allotted the resource for execution. This process is rehashed until meta-assignment isn't unfilled. Max-Min Algorithm is useful for processes having larger completion time which tend to end up in the waiting queue.

3. **RASA:** RASA exploits the features of Max-min and Min-min algorithm to allocate tasks, leading to increase synchronization of execution.

A cloud environment involves scheduling of tasks using the various algorithmic techniques available. While scheduling tasks, the algorithms pursue a similar vitality in every circumstance once in a while prompting increment in makespan of the procedures. The proposed work undertakes the implementation of the various algorithms available in order to combine their features and thus leading to a more efficient alternative. Throughput, execution time, makespan, turnaround time, waiting time are some metrics which form the basis of calculations. Load balancing is similarly critical, which aims at enhancing the distribution of workload among the various resources of computation. The process involves analysis of each algorithm and the total time taken for execution of the same. This helps in calculating the various metrics for each resource. The following process also gives an idea about the feasibility of execution and thus a task which fails to meet the constraints is migrated to another resource in order to maximize utilization of resource. **[5]**

With tremendous research in the field of cloud computing, researchers tend to move towards scheduling alternatives which combine features of different algorithms in order to provide an optimized solution. Optimization algorithms increase efficiency and thus preferred over conventional algorithms. The following models tend to optimize task scheduling solutions:-

1. **Ant Colony optimization:** This is an approach for balancing the load between the different nodes of a cloud system. It involves the solution through probabilistic estimation and path analysis. There have been several with time in this algorithm by different computer scientists to gain efficiency in the cloud computing environment. **[6]**

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

2. Berger model: Baomin et al. **[7]** proposed an algorithm based on considerations of commercialization and virtualization characteristics of cloud computing. It involves the classification of tasks by preferential order of Quality of Service. It derives a function to establish a constraint on the process of selection of resources.

3. Round Robin algorithm: The Round Robin algorithm is focussed towards the equal distribution of resources. The semantics of round robin in a cloud environment is derived from process scheduling in operating systems. The tasks are allotted in a cyclic manner and are executed for a fixed time quantum, In a cloud environment the cloud broker provdes the resources to the virtual machines. The process continues until the successful completion of all the tasks have taken place.**[8]**

A Survey on Different Scheduling Algorithms in Cloud Computing drove us to the foundation of fundamental ground definitions identified within the field of cloud. An essential presentation of Cloud Computing characterizes it as 'Processing administrations conveyed to the client over the web'. It is fundamentally sharing of assets between individuals who need to retrieve data, software and even hardware resources, anywhere and at any time. Cloud computing is pay per use administration and many significant players which give such administrations include Google, Amazon and Microsoft. These are referred to as Cloud Service Providers. Cloud engineering has two segments, Frontend and Backend. These two segments are associated in a system utilizing web. Front end segments are those components which give the application and interface to get to the cloud stage. The backend segment involves the processing and computations.

The two main classification of scheduling techniques which currently exist are as follows:-

1. **Static Scheduling:** The information about the process, tasks and resources is available to scheduler before execution. It has less runtime overhead.

2. **Dynamic Scheduling:** The Information about the task components is not available before execution. In dynamic scheduling, cloudlets are created dynamically while the process is executing. The details about process and execution is not known until a task is encountered. Compared to static scheduling, dynamic scheduling has more runtime overhead.

These algorithms, take into account two major parameters, namely Task Length and Deadline. The parameters that contribute towards the scheduling techniques are as follows:-

1. **Earliest Feasible Deadline:** The task with shortest deadline gets scheduled. It is a dynamic scheduling technique. After completion of one process, the queue is searched for the task which is closest to the deadline and is scheduled next.

2. **Priority Based Job Scheduling Algorithm**: Mathematical calculations are involved. For scheduling, the tasks are allotted a weighted priority. The job requests for a resource with the priority.

3. **Greedy Based Job scheduling:** In order to reduce the completion time, leading to faster solutions the greedy algorithmic technique is used. The process involves selecting the next best state out of all the available states This algorithm has proven to be very efficient for cloud in business enterprises.

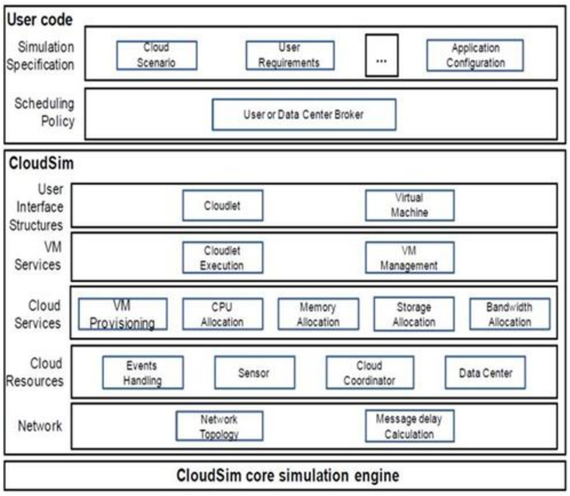

Figure 1: The CloudSim Architecture [11]

As the distributed computing innovation is changing step by step a ton of new challenges arise. One of the major challenges is the scheduling of tasks for a cloud computing environment. The main purpose of scheduling algorithms is to maximise resource utilization and minimize makespan**[9].**

Cloud computing is an internet based improvement and utilization of innovation and its one among the progressed and the most recent processing models involving applications and knowledge administrations accessible over the net **[10]**. Cloud computing is frequently sketched out as a strategy of computing which is progressively versatile, highly scalable, dynamic and involve virtualized resources provided as administrations over the web. Task allocation and management is a standout among the most troublesome hypothetical issues in the distributed computing arena. Job management is an essential constituent of a cloud computing framework, task scheduling issues are principle that identifies with the proficiency of the total computing framework. Task scheduling is often viewed as a mapping instrument from clients' undertakings to the right selection of computing resources and its execution. Task scheduling is an adaptable and convenient process. Scheduling can be powerfully self-versatile. Increasing and contracting applications inside the cloud additionally rely upon the need of proper resource allocation. The virtual registering assets in cloud framework may also grow or contract at the indistinguishable time.

Task scheduling can be split into two categories, one would be the pool scheduling of applications and the other would be the scheduling of resources among the cloud. Therefore, the main challenge and idea behind scheduling is to utilize resources to the maximum extent possible and increase efficiency.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

III. PROPOSED WORK

The proposed work includes analysis and survey of various currently existing task scheduling algorithms. The various algorithms have been analysed for various scheduling criteria which are later used to find out the most efficient algorithm. This involves successful implementation of the task scheduling algorithms in order to come up with a new workforce dynamic opulence split algorithm. Other algorithms analysed include First Come First Serve, Shortest Job First and Max Min Scheduling, Round Robin, Particle Swarm Optimization etc. The simulation of cloud platform has taken place on cloudsim integrated by eclipse, a java based integrated development environment.

A. Flowchart of Execution

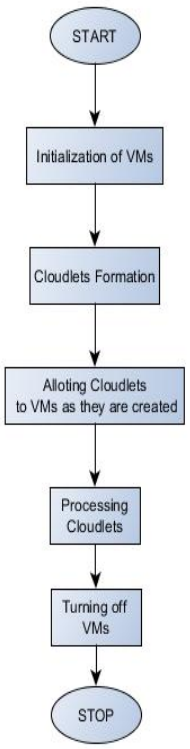

Figure 2: FCFS

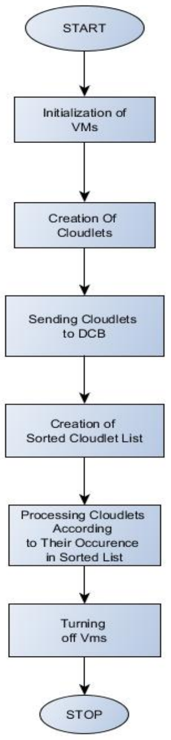

Figure 3: SJF

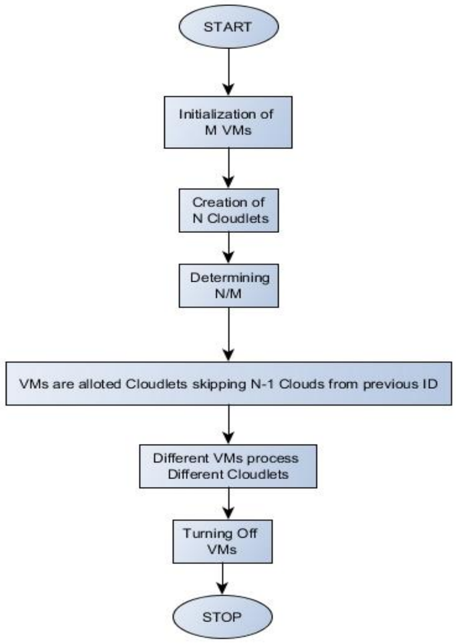

Figure4: OPULENCE SPLIT

IV. ALGORITHM FOR EXECUTION

The opulence split algorithm exploits dynamism and fault tolerance in order to generate an algorithm which leads to formation of cloudlets and allocation of resources in real time. After getting the execution time and the utilisation percentage, the scheduler can either complete the task or allot it to a different virtual machine, therefore taking in the next task which is waiting in the queue. The main aim of the algorithm is to minimize waiting time and maximize throughput.

Opulence Split Algorithm

Step 1: Start

Step 2: Initialization of virtual machines

Step 3: Cloudlet formation

Step 4: Determine N/M

Step 5: Every nth cloudlet will be allocated to the virtual machine skipping N-1 Cloud from previous ID

Step 6: Get Execution time and Utilisation Rate

Step 7: Process cloudlets in a circular fashion with n-1 different Virtual Machine processing different cloudlets.

Step 8: Close down virtual machines

`

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

V. RESULT AND PERFORMANCE EVALUATION

The execution of the algorithms was conducted in cloud sim. The different algorithms like FCFS, SJF, Opulence Split and Particle Swarm Optimization were implemented and their executions were observed. The different algorithms were implemented in cloudsim with a task size of 10. The task sizes are directly proportional to the total completion time of the algorithms. The particle swarm optimization algorithm was implemented with a particle size=30 in order to track the performance of the algorithm. Figure 5,6,7 and 8 show the completion times of different algorithms.

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
02	SUCCESS	05	05	2405.12	00.1	2405.22
00	SUCCESS	02	02	2598.4	00.1	2598.5
03	SUCCESS	04	04	2643.06	00.1	2643.16
01	SUCCESS	06	06	2894	00.1	2894.1
06	SUCCESS	03	03	3894.74	00.1	3894.84
04	SUCCESS	06	06	1088	2894.1	3982.1
09	SUCCESS	02	02	1594.43	2598.5	4192.92
10	SUCCESS	05	05	1967.3	2405.22	4372.52
05	SUCCESS	04	04	3289.26	2643.16	5932.41
12	SUCCESS	03	03	2324.88	3894.84	6219.73

Figure 4: FCFS EXECUTION

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
08	SUCCESS	02	02	1379.77	00.1	1379.87
02	SUCCESS	03	03	1863.62	00.1	1863.72
13	SUCCESS	04	04	1939.54	00.1	1939.64
00	SUCCESS	06	06	3445.49	00.1	3445.59
03	SUCCESS	06	06	365.38	3445.59	3810.97
01	SUCCESS	05	05	4066.28	00.1	4066.38
24	SUCCESS	04	04	2411.76	1939.64	4351.4
11	SUCCESS	02	02	3340.98	1379.87	4720.86
04	SUCCESS	06	06	1088	3810.97	4898.96
05	SUCCESS	03	03	4230.05	1863.72	6093.78

Figure 5: SJF EXECUTION

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
03	SUCCESS	03	03	365.38	00.1	365.48
20	SUCCESS	05	05	906.79	00.1	906.89
07	SUCCESS	04	04	1145.88	00.1	1145.98
22	SUCCESS	06	06	1230.53	00.1	1230.63
04	SUCCESS	03	03	1088	365.48	1453.48
14	SUCCESS	04	04	907.35	1145.98	2053.32
27	SUCCESS	06	06	1311.59	1230.63	2542.22
00	SUCCESS	02	02	2598.4	00.1	2598.5
15	SUCCESS	04	04	580.75	2053.32	2634.08
21	SUCCESS	05	05	2045.46	906.89	2952.36

Figure 6: OPULENCE SPLIT EXECUTION

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
00	SUCCESS	03	06	3445.49	00.2	3445.69
01	SUCCESS	02	03	3483.54	00.2	3483.74
02	SUCCESS	02	05	2405.12	00.2	2405.32
03	SUCCESS	02	05	2909.7	2405.32	5315.02
04	SUCCESS	02	05	2410.35	5315.02	7725.37
05	SUCCESS	02	03	4230.05	3483.74	7713.8
06	SUCCESS	02	03	3894.74	7713.8	11608.54
07	SUCCESS	02	03	1145.88	11608.54	12754.42
08	SUCCESS	02	04	2193.71	00.2	2193.91
09	SUCCESS	02	05	2529.81	7725.37	10255.18
10	SUCCESS	02	03	1128.66	12754.42	13883.08

Figure 7:PARTICLE SWARM OPTIMIZATION

Figure 9 compares the make span of the different algorithms. Particle Swam Optimization is highly efficient in in dealing with a number of tasks. Makespan refers to the total time elapsed between the arrival and completion of tasks. On the other hand, a parallel study on the performance of particle swarm optimization involved variation of the particle size to observe the effect on the makespan and overall performance of the algorithm. Figure 10 explains the variance of makespan with the particle size.

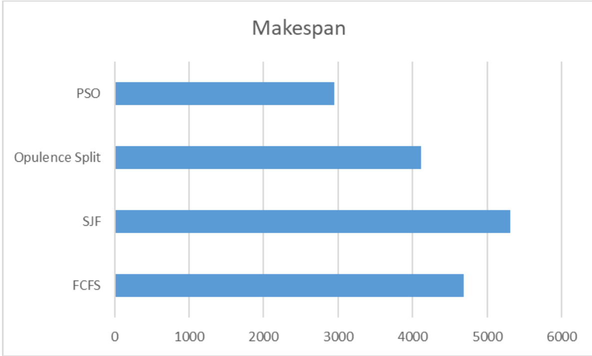

Figure 9:MAKESPAN COMPARISON

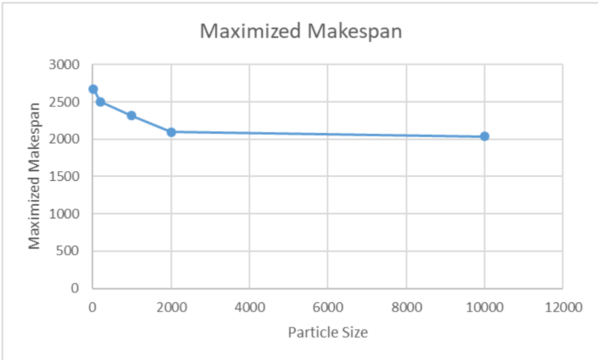

Figure 8. MAKESPAN VS PARTICLE SIZE

Figure 11 makes a comparison with respect to the task completion time in the different algorithms. Table 1 is a concise explanation of the researched algorithms.

Figure 9: TIME COMPARISON

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Scheduling Technique	Criteria of Scheduling	Advantage	Disadvantage
FCFS	Arrival on basis of cloudlet id	Easy to implement	No considerations of optimisations
MAX-MIN	Completion Time	Better makespan	Poor Load Balancing
SJF	Completion time, arrival	Lesser Starvation Minimised TAT	Starvation of larger processes.
OPULENCE SPLIT	Division of Cloudlets is made possible.	Load Balancing and concurrency control implementation	Sometimes, number of cloudlets are uneven, hence energy is wasted.
PSO	Particle Size	Highly Efficient and scalable	Difficult to Integrate with Real Time Workflows

Table 1: COMPARISON OF VARIOUS TASK SCHEDULING ALGORITHMS

VI. CONCLUSION

In this paper we successfully analysed the different cloudlet scheduling algorithms with respect to their properties. We also proposed an improved algorithm named Opulence Split implemented to balance the load between different machines. Simulation of all the algorithms was done on CloudSim. It was observed that Particle Swarm Optimization is a highly efficient and scalable algorithm. With respect to the increasing number of tasks, it can be used to obtain better throughput and reduced makespan. However it fails to be effective as it requires memory latency and is harder to integrate with real time scenarios. FCFS, SJF, Max-Min, Min-Min etc are traditional scheduling algorithms pretty useful in their respective scenarios. However they have their own limitations and drawbacks. The Opulence Split Algorithm invokes concurrency control and increased load balancing and fault tolerance to maximise resource utilization by using dynamic cloudlet creation. However, the algorithm did take time to execute, it provided better performance compared to FCFS and SJF. It is easier to implement Opulence Split Algorithm when dealing with real time scenarios and there is a resource constraint. The techniques currently present are effective although they are not working on finiteness and may fail to implement principles of concurrency. Future work can be used to make the algorithms more energy efficient and increase dynamism.

REFERENCES

1. Mell, Peter M., and Timothy Grance. "Sp 800-145. thenist definition of cloud computing." (2011).
2. M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "Above the clouds: A Berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, 2009.
3. Dillon, Tharam, Chen Wu, and Elizabeth Chang. "Cloud computing: issues and challenges." Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on. Ieee, 2010.
4. Fazel, M. D., S. Jamali, and M. Bekravi. "Survey on job scheduling algorithms in cloud computing." InternationalJournal of Emerging Trends and Technology in Computer Science (2014): 151-4.
5. Mittal, Shubham, andAvitaKatal. "An Optimized Task Scheduling Algorithm in Cloud Computing."Advanced Computing (IACC), 2016 IEEE 6th International Conference on IEEE, 2016.
6. Nishant, Kumar, et al. "Load balancing of nodes in cloud using ant colony optimization." Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on. IEEE, 2012.
7. Xu, Baomin, et al. "Job scheduling algorithm based on Berger model in cloud environment." Advances in Engineering Software 42.7 (2011): 419-425.
8. Mathew, Teena, K. ChandraSekaran, and John Jose. "Study andanalysis of varioustask scheduling algorithms in the cloud computing environment." Advances in Computing, CommunicationsandInformatics (ICACCI, 2014 International Conference on . IEEE, 2014.
9. Wadhonkar, A., &Theng, D. (2016, February). A survey on different scheduling algorithms in cloud computing. In Advances in Electrical, Electronics, Information, Communicationand Bio-Informatics (AEEICB), 2016 2nd International Conference on (pp. 665-669). IEEE
10. Guo, L., Zhao, S., Shen, S., &Jiang, C. (2012). Task scheduling optimization in cloud computing based on heuristic algorithm. JNW,7(3),547-553.
11. Cloud Environment Simulator, C. (2017). CloudSim Architecture. Available at: http://www.thedailyprogrammer.com/2016/09/cloudsimsimulation-software-for-cloud.html [Accessed 1 JAN. 2019].