Priority Task Scheduling Strategy for Heterogeneous Multi-Datacenters in Cloud Computing

Naoufal Er-raji
Department Of Mathematics and Informatics
Laboratory of Modeling and Information Technology
Faculty of sciences Ben M'SIK University Hassan II
Casablanca, Morocco

Faouzia Benabbou
Department Of Mathematics and Informatics
Laboratory of Modeling and Information Technology
Faculty of sciences Ben M'SIK University Hassan II
Casablanca, Morocco

***Abstract*—With the rapid development in science and technology, cloud computing has emerged to be widely adopted in several IT (Information Technology) areas. It allows for the companies as well as researchers to use the computing resources as a service over a network as internet without owning the infrastructure. However, Due to increasing demand of cloud computing, the growing number of tasks affects the system load and performance. Scheduling of multitasks with respect SLA (Service Level Agreement) can face serious challenges. In order to overcome this problem as well as provide better quality of service, the tasks have to be scheduled in optimal way. In this paper, we address the problem of the priority task scheduling through proposing a global strategy over distributed data-center in cloud computing basing on three parameters: tasks deadline, task age and the task length.**

***Keywords*—age; cloud computing; cluster; data-center; deadline; length; node; SLA; priority task scheduling**

I. Introduction

In order to satisfy the rising demand of computing resources, most of the IT industry's and companies start using the cloud computing. This technology is a large-scale distributed computing paradigm that consists of a mix of technologies as virtualization, SOA (Service Oriented Architecture) ... for providing shared pool of computing resources dynamically to the CSC (Cloud Service Consumers) through connecting tens of thousands of servers over a network as internet. Moreover, it offers an easy access to those resources anywhere and at any time [1] [2] [3]. Due to the increasing demand of cloud resources and the limited resources of CSP (Cloud Service Providers), this raises several challenges, and task scheduling is one of them. The task scheduling, is a process of choosing the best suitable available resources for execution the tasks or to allocate computer machines to tasks in such a manner that the completion time is minimized as possible [4] [5].

In this paper, we have focused on the priority tasks because they have a big impact on the services providing and a longer delay can make a violating in the SLA contract. Therefore, we propose a global priority task scheduling strategy that's based on three parameters: task deadline, task length and task age. The main purpose of this strategy is to improve the priority task scheduling, therefore, respecting the deadlines that are documented in the SLA contract. The rest of paper is organized as follow, Section 2 describes the Proposed Model Architecture Cloud Computing, section 3 presents related works, and section 4 explains the priority task scheduling strategy. At last we make a discussion and conclusion of the paper.

II. Proposed Model Architecture Cloud Computing

In the cloud computing, the scheduling can affect the performance in a right or a wrong way. So according to the scheduling process, tasks could be processed quickly or could remain too long time at queue.

The scheduling process can be done at different levels and at different components depending on the nature of the strategy followed by the CSP. Thus, the cloud computing need to be evolved by a good task scheduling strategy and a global view of all processes is needed. This section describes the suggested model. We consider that each CSP has a multi-cloud data-center, each data-center has several clusters and each cluster consists of numerous servers and each server runs numerous VMs (Virtual Machines).

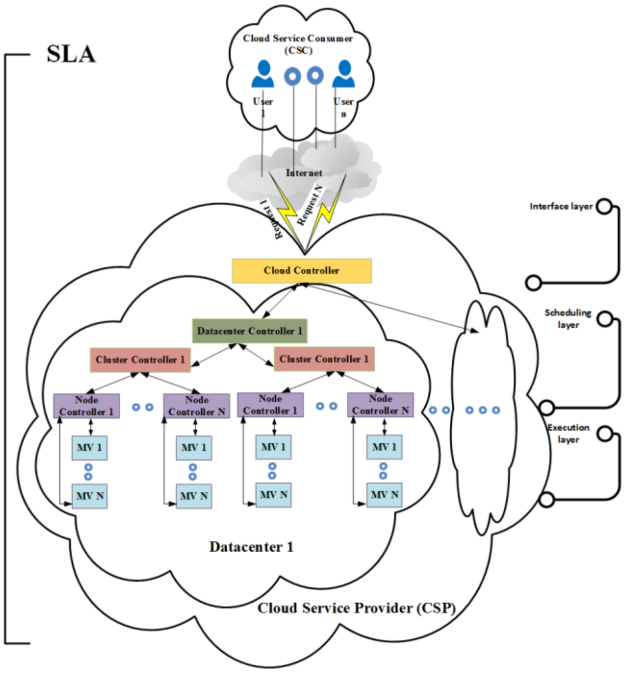

Fig. 1. Proposed Model Architecture Cloud Computing

As shown in the figure one: the proposed Model Architecture Cloud Computing is composed of three layers and six components [6], the characteristics of each one are summarized in the following section. We have the layers:

- **Interface layer**: the responsible for the communication between the CSC and CSP.
- **Scheduling layer**: here where all algorithms and mechanisms worked to schedule the tasks to the suitable resources.
- **Execution layer**: here where the entire task will be executed.

The components are:

- **Users**: they place theirs tasks with the help of cloud controller through their browsers or applications.
- **Cloud Controller**: receive the tasks from the users; keep track of the data-centers and their performance and assign tasks to the suitable data-center.
- **Datacenter Controller**: receive the tasks from the cloud controller, keep track of the clusters and their performance and assign tasks to the suitable cluster according to the cloud controller direction.
- **Cluster Controller**: receive the tasks from the Datacenter controller, keep track of the Nodes and their performance and assign tasks to the suitable Node according to the cloud controller direction.
- **Node Controller:** receive the tasks from the cluster controller, keep track of the VMs and their performance and assign tasks to the suitable VM according to the cloud controller direction.
- **VM**: execute the tasks according to the cloud controller direction.

The CSC submits their tasks having different length, and different deadline (QoS (Quality of Service) requirement), from anywhere at any time, so as to be executed as soon as possible under the SLA constrain. In the other side, the CSP have several data-centers geographically distributed, which consist of numerous servers and each server runs numerous VMs and finally each VM has different capability to execute different tasks. However, with the growing amount of the CSC tasks, with the presence of different priorities tasks, arise several difficulties either for making the best choice of resources or in which order the tasks have to be assigned.

III. Related Works

In the literature, there is a vast number of propositions for improving task scheduling in cloud computing. One of those methods is task scheduling based on priority. Here, some contributions that are based on priority for their propositions:

Atul Vikas Lakraa and Dharmendra Kumar Yadav [7] proposed an algorithm, which assigns priority to different tasks according to the QoS of request task. High QoS task assigned with low QoS value and the low QoS task assigned with high QoS value. Hence, the task with lower QoS value is a high priority and the task with high QoS value is a low priority. And for the resource allocation, the VMs are selecting according to their MIPS (Million Instructions Per Second) Values such that the one having highest MIPS has the highest ability to be assigned.

Samia Ijaz, Ehsan Ullah Munir, Waqas Anwar, and Wasif Nasir [8] proposed a strategy, which assigns priority to different tasks according to the ALST (Absolute Latest Start Time) such that the one having minimal ALST among all the tasks has higher priority. For resource allocation, the VMs are selecting according to EST (Earliest Start Time) and ECT (Earliest Completion Time) which on each VM is reckoned using three mathematical equations and the VM that gives the least ECT for a task is selected.

In Papers [9] [10] the proposed algorithms assign priority to different tasks according to task size such that one having highest size has highest priority. For the resource allocation, the VMs are selecting according to their MIPS values such that the one having highest MIPS has the highest priority to be selected.

Deepika Saxena, R.K. Chauhan and Ramesh Kait [11] proposed an algorithm, which Classify and group all tasks according to their deadline and cost constraints, and assign them to different priority queue (high, mid, low). And for resources allocation the approach is based on greedy resource (VM) allocation for selecting the resources which means that scheduler select VM with minimum turnaround time for each individual task.

Aditi Sharma and Shivi Sharma [12] proposed a technique, which is based on credit system. Each task is assigned a unique credit based upon three parameters namely: Task Length Credit, Task priority credit, Task deadline credit. Based on these credits the tasks are assigned to VM.

Shamsollah Ghanbari and Mohamed Othman [13] propose an algorithm, which is based on the theory of a mathematical model named AHP (Analytical Hierarchy Process) for calculating the priority. It is a MCDM (Multi Criteria Decision Making) and MCDM (Multi Attribute Decision Making) model. The proposed architecture is consisted of three levels namely: objective level, attributes level and alternatives level.

Pankajdeep Kaur and Parampreet Singh [14] proposed an algorithm, which assigns priority to different tasks according to specific attribute: User Level, Task urgency, Task Load and Time queuing up. Then, tasks are arranged in a sorted order by considering the calculated priority. Thus, the task with higher priority scheduled first.

In the literature, there are many contributions. Which make a comparison of approaches either for priority based or other parameters based [15][16][17][18][19][20] [21].

In this section, we propose a comparison of the above propositions. The criteria of comparison are:

- Factor considered for task priority.
- Factor considered for VM priority.
- Centralized Scheduling or distributed scheduling.
- Resources Load.

TABLE I. ALGORITHMS COMPARISON TABLE

Propositions	Factor considered for task priority	Factor considered for VM priority	Centralized Scheduling	Distributed Scheduling	Resources Load
[7]	QoS	MIPS	Yes	No	No
[8]	ALST	EST, ECT	Yes	No	No
[9]	Task SIZE	MIPS	Yes	No	Yes
[10]	Task SIZE	MIPS	Yes	No	No
[11]	DEADLINE, COST	GREEDY (minimum turnaround time)	Yes	No	No
[12]	CREDITS (Task Length Credit, Task Priority Credit, Task Deadline Credit)	No	Yes	No	No
[13]	AHP	AHP	Yes	No	No
[14]	User Level, Task urgency, Task Load and Time Queuing up	No	Yes	No	No

IV. PROPOSED PRIORITY TASK SCHEDULING STRATEGY

A. Priority Task Scheduling

In the literature and according to [22], a task is a single process or multiple processes which will be executed on a compute node presented by a VM in the cloud computing. However, Scheduling is a group of mechanisms that manage the order of execution of multiple tasks on the resources [5].

Task scheduling is divided into two categories: static scheduling and dynamic scheduling. In the static scheduling, the scheduling decisions are taken before tasks are submitted. The dynamic one is divided into two types: batch mode or online mode. In batch mode tasks are queued, and scheduled after fixed period. In online mode task is scheduled dynamically when they arrive in the system [23].

Our priority task scheduling strategy can be considered as dynamic as well as centralized scheduling.

The following figure presents the components of the proposed cloud controller, which is the most important component in this strategy.

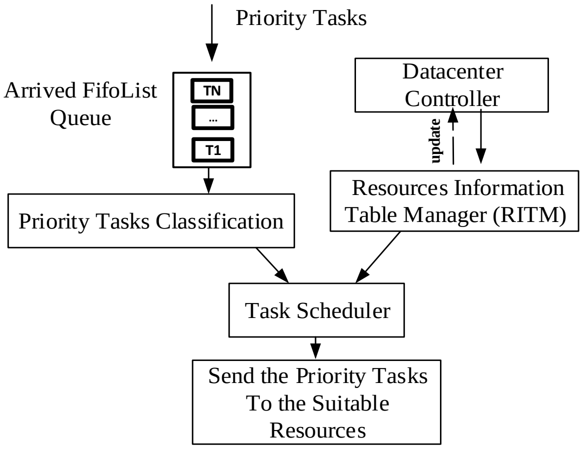
Priority Tasks
Arrived FifoList Queue
TN
...
T1
Datacenter Controller
update
Resources Information Table Manager (RITM)
Priority Tasks Classification
Task Scheduler
Send the Priority Tasks To the Suitable Resources

Fig. 2. Cloud Controller Process

As shown in the figure two, the priority task scheduling in the cloud controller is based on the four following steps.

1) Classification of VMs
2) Priority Tasks Classification
3) Assignment of the tasks to the VMs (Task scheduler)
4) Updating the resources information table

Based on those steps, the cloud controller assign tasks to the appropriate resources.

Here some abbreviations used in this paper:

TABLE II. ABBREVIATIONS TABLE

Notation	Definitions
Ti[Tl]	Length of a task i or the number of instruction that need the task to be executed in a resource.
Ti[Td]	The deadline of the tasks (given from the user SLA).
Ti[Ta]	The age of the task (the task waiting in the system).
QP	Priority Queue
SDC	The data-center speed.
SCL	The cluster speed.
SN	The node speed
P	The processing speed or ability in MIPS For The VM.
Load	The resource load

B. Priority Tasks Classification

In the cloud computing, different CSC generates different tasks having different length, different deadline, different arrived time and the sequence is dependent on the arriving time. In order to have an efficient Priority task scheduling strategy the priority tasks classification is very important. Various parameters can be used like:

- CPU Task utilization.
- RAM Task utilization.
- Bandwidth Task utilization.
- Task length or size.
- File Task size.
- The task deadline.
- The task waiting time in the system (Age).
- …..etc.

In our strategy, we are oriented to use dynamic priority based on three parameters: the task deadline, the task length, and the task age.

1) Task deadline: We have chosen the deadline as priority parameter because it is one of the important parameter that the CSP have to respect in order to respect the QoS that are documented in the SLA.

2) Task age: We have chosen the age as priority parameter because when the task has low priority, it has to wait for a long time and this leads to increase in execution time.

3) Task length: We have chosen the length as priority parameter because when the task has small lengths it can execute rapidly, thus, liberate the resources as soon as possible for the other tasks.

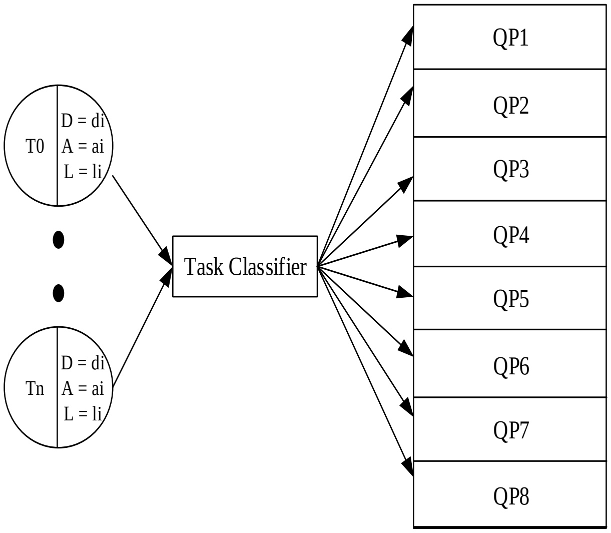

Fig. 3. Tasks Classification

As shown in the figure three, the task classifiers receive the priority tasks in order to classify them in the priority queues (from QP1 to QP8) and will work as follow:

1) Tasks having shorter deadline, shorter length and highest age will reside at the first priority queue (QP1).

2) Tasks having shorter deadline, shorter length and shorter age will reside at the second priority queue (QP2).

3) Tasks having shorter deadline, highest length and highest age will reside at the third priority queue (QP3).

4) Tasks having shorter deadline, highest length and shorter age will reside at the fourth priority queue (QP4).

5) Tasks having highest deadline, shorter length and highest age will reside at the fifth priority queue (QP5).

6) Tasks having highest deadline, shorter length and shorter age will reside at the sixth priority queue (QP6).

7) Tasks having highest deadline, highest length and highest age will reside at the seventh priority queue (QP7).

8) Tasks having highest deadline, highest length and shorter age will be present at the last priority queue (QP8).

The algorithm pseudo-code used in this step is explained in detail below.

ALGORITHM 1: Priority Tasks Classification

```
Repeat
For i from 1 to Arrive FifoList Queue Size do
Switch (Ti [Td] && Ti [Tl] && Ti [Ta]) {
Case Ti [Td] = Min_Deadline && Ti [Tl] = Min_Lenght && Ti [Ta] = Max_Age: Then put Ti in QP1
Case Ti [Td] = Min_Deadline && Ti [Tl] = Min_Lenght && Ti [Ta] = Min_Age: Then put Ti in QP2
Case Ti [Td] = Min_Deadline && Ti [Tl] = Max_Lenght && Ti [Ta] = Max_Age: Then put Ti in QP3
Case Ti [Td] = Min_Deadline && Ti [Tl] = Max_Lenght && Ti [Ta] = Min_Age: Then put Ti in QP4
Case Ti [Td] = Max_Deadline && Ti [Tl] = Min_Lenght && Ti [Ta] = Max_Age: Then put Ti in QP5
Case Ti [Td] = Max_Deadline && Ti [Tl] = Min_Lenght && Ti [Ta] = Min_Age: Then put Ti in QP6
Case Ti [Td] = Max_Deadline && Ti [Tl] = Max_Lenght && Ti [Ta] = Max_Age: Then put Ti in QP7
Case Ti [Td] = Max_Deadline && Ti [Tl] = Max_Lenght && Ti [Ta] = Min_Age: Then put Ti in QP8}
Remove Ti from FIFOList
End for
Until FifoList Queue is empty
```

C. VMs classification

As shown in the figure two, RITM obtains information about all resources through calling the datacenters controller, and each datacenter controller has a table that contains all information about their clusters (through calling their cluster controller) and so on until having information about all VMs. Hence, when the RITM receives the tables from the datacenters controller, it makes an update in its general table. This table contains for each resource its load and speed (performance). At last, it provides to the task scheduler a SVMT (Sorted VM Table) containing all VMs sorted according to their load.

TABLE III. SORTED VM TABLE (SVMT)

VMs (Sorted According To Their Load)
VMxxxx
…
VMxxxx

Note that the first x mean in which data-center the VM is created, the second mean in which cluster, the third in which node and finally the last x mean the number of this VM.

Having this information the task scheduler can choose the VM with the best features to execute a priority task. Moreover, it can choose the location of the data-center if required in the SLA contract.

D. Tasks Assignment to VMs

The pseudo-code of the priority task-scheduling algorithm is as a follow:

ALGORITHM2: Update Task Priority

```
Repeat
For i from 2 to 8 do
    For j from 1 to QPi.Size do
        If (Tj[Td] = Min_ DeadlineQPj) do
            Put Tj in the end of PQi-1
    End for
End for
```

```
Until all priority queues are empty
```

The main priority task scheduling algorithm is as follows:

ALGORITHM 3: Priority Task Scheduling

```
Repeat
    For j from 1 to 8 do
        If (QPj is not empty) do
            i=j
            Repeat
                Assign the first task in the QPi to the first VM in SVMT
            Until QPi is empty
        End if
    End for
    Update Task Priority
    Update SVMT
Until all priority queues are empty
```

Note that our algorithm offers several tasks priorities and this allows scheduling the highest priority tasks first. In the other side for the tasks that have least priority haven’t waited a lot in their queues because they can ascend to the above queue thus have more priority.

E. Updating the resources information table

As shown in the figure two, the RITM is the only component that provides the resources information to the task scheduler. So, as to assign tasks to the resources (VMs). The RITM update dynamically its resources information table after a periodical time which can be defined either by the CSP or if there is a request from the task scheduler. Here, an example of this table.

TABLE IV. RESOURCES INFORMATION TABLE

Datacenters	**Clusters**	**Nodes**	**VMs**
Datacenter (SDCi,loadi)	Cluster$_{ij}$SCL$_{ij}$, Load$_{ij}$)	Node$_{ijk}$(SN$_{ijk}$, load$_{ijk}$)	VM$_{ijkl}$(P$_{ijkl}$, Load$_{ijkl}$)

Note that the speed of each component is calculated through the speed addition of the under components and it is updated dynamically after a periodical time.

V. DISCUSSION AND CONCLUSION

Task scheduling is a big challenging issue in the cloud computing. In order to satisfy both the CSC as well as the CSP, efficient task scheduling strategy is required. The rest of this section is devoted to make a conclusion and a discussion about the proposed priority task scheduling strategy. Having examined existing scheduling algorithms that centered on priority, allow us to make a conclusion that mostly authors do not give an overview of task scheduling from the first step, when the task arrive, to the last step when the tasks are executed in the resources. They didn’t give the importance to the priority tasks, in spite of, having a big importance in the SLA contract. In this paper, we have considered that each CSP has a multi-cloud data-center, each data-center has several clusters. As centralized and dynamic scheduling, the cloud controller is the only component who decides in which resources the priority tasks will be executed. It allows having a global view of all cloud resources components, which can even make a solution for the load balancing problems. Priority queues have also been considered for the tasks classification (from QP1 to QP8), according to three parameters: task deadline, task length and task age. Moreover, the task ascending in the priority queue also has taking into consideration, such as a task can ascend to the above queue if the task deadline equal to the min-deadline compared to the existing tasks deadline in the same queue at a specific time. Those features are the strength of our strategy that accelerated the execution of the priority task compared with the other one. Finally, as future work, our research continues by making the simulation for proposed priority task scheduling strategy, make in consideration the load balancing, and migration of running tasks from one machine to other machines with better performance.

REFERENCES

[1] Michael Armbrust, Armando Fox, Rean Griffith,Anthony D. Joseph, Randy Katz, Andy Konwinski,Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia "A View Of Cloud Computing" Communications Of The ACM – Vol. 53 (4), April 2010.

[2] Amardeep Singh And Inderjit Kaur "A Survey On Cloud Computing And Various Scheduling Algorithms" International Journal Of Advance Research In Computer Science And Management Studies (IJARCSMS) – Vol. 4 (2), February 2016.

[3] Definition of Cloud Computing, https://www.nist.gov/programs-projects/cloud-computing, viewed 07/12/2016.

[4] Raja Manish Singh, Sanchita Paul and Abhishek Kumar “Task Scheduling In Cloud Computing : Review",International Journal Of Computer Science And Information Technologies (IJCSIT) – Vol. 5 (6) , 2014.

[5] Amanpreet Kaur And Usvir Kaur "A Survey For Task Scheduling In Cloud Computing" International Journal Of Advanced Research In Computer Science And Software Engineering (IJARCSSE) – Vol. 6 (5), May 2016.

[6] Tingting Wang, Zhaobinliu , Yi Chen, Yujie Xu and Xiaoming Dai "Load Balancing Task Scheduling Based On Genetic Algorithm In Cloud Computing" The 12th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC), 2014.

[7] Atul Vikas Lakraa and Dharmendra Kumar Yadavb "Multi-Objective Tasks Scheduling Algorithm For Cloud Computing Throughput Optimization" International Conference on Intelligent Computing, Communication & Convergence (ICCC), 2015.

[8] Samia Ijaz, Ehsan Ullah Munir, Waqas Anwar And Wasif Nasir "Efficient Scheduling Strategy For Task Graphs In Heterogeneous Computing Environment " The International Arab Journal of Information Technology (IAJIT), Vol. 10 (5), September 2013.

[9] Amit Agarwal And Saloni Jain "Efficient Optimal Algorithm Of Task Scheduling In Cloud Computing Environment" International Journal of Computer Trends and Technology (IJCTT) – Vol. 9 (7), March 2014.

[10] M. Lawanya Shri, M.B.Benjula Anbumalar, K. Santhi And Deepa.M "Task Scheduling Based On Efficient Optimal Algorithm In Cloud Computing Environment" International Conference on “Recent Research Development in Science, Engineering and Management (ICRRDSEM), May 2016.

[11] Deepika Saxena, R.K. Chauhan And Ramesh Kait "Dynamic Fair Priority Optimization Task Scheduling Algorithm In Cloud Computing: Concepts And Implementations", I. J. Computer Network and Information Security (IJCNIS), February 2016.

[12] Aditi Sharma And Shivi Sharma "Credit Based Scheduling Using Deadline In Cloud Computing Environment", International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE) – Vol. 4 (2), February 2016.

[13] Shamsollah Ghanbari And Mohamed Othman "A Priority Based Job Scheduling Algorithm In Cloud Computing" International Conference on Advances Science and Contemporary Engineering (ICASCE), January 2012.

[14] Pankajdeep Kaur And Parampreet Singh "Priority Based Scheduling Algorithm With Fast Task Completion Rate In Cloud" Advances In Computer Science And Information Technology (ACSIT) – Vol. 2 (10), April-June 2015.

[15] Er-raji Naoufal, Faouzia Benabbou And Ahmed Eddaoui "Task Scheduling Algorithms In The Cloud Computing Environment: Survey And Solutions" International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) – Vol. 6 (1), January 2016.

[16] S. Jeyalakshmi And N. Sankarram "Scheduling Algorithms For Cloud Computing Environments And Research Issues" Journal Of Applied Sciences Research (JASR) – Vol. 12 (3), March 2016.

[17] Jyoti Thaman And Manpreet Singh "Current Perspective In Task Scheduling Techniques In Cloud Computing: A Review" International Journal in Foundations of Computer Science & Technology (IJFCST) – Vol. 6 (1), January 2016.

[18] Rupinderjit Singh And Er.Manoj Agnihotri "A Review Of Cloud Computing Scheduling Strategies" International Journal Of Engineering Trends And Applications (IJETA) – Vol. 3 (4), July-August 2016.

[19] Amardeep Singh And Inderjit Kaur "A Survey On Cloud Computing And Various Scheduling Algorithms" International Journal Of Advance Research In Computer Science And Management Studies (IJARCSMS) – Vol. 4 (2), February 2016.

[20] Sujit Tilak And Prof. Dipti Patil "A Survey Of Various Scheduling Algorithms In Cloud Environment" International Journal Of Engineering Inventions (IJEI) – Vol. 1 (2), September 2012.

[21] Swachil Patel And Upendra Bhoi "Priority Based Job Scheduling Techniques In Cloud Computing: A Systematic Review" International Journal Of Scientific & Technology Research (IJSTR) – Vol 2 (11), November 2013.

[22] Definition Of Tasks, https://msdn.microsoft.com/en-us/library/bb525214(v=vs.85).aspx, viewed 07/12/2016.

[23] P. Akilandeswari and H. Srimathi "Survey And Analysis On Task Scheduling In Cloud Environment" Indian Journal of Science and Technology(INDJST) – Vol 9 (37), October 2016.