# Simulator for Priority based Scheduling of Resources in Cloud Computing

P. K. Suri
Former Dean (Sc. /E&T)
and Professor & Chairman
(Department of Computer
Science and Applications),
Kurukshetra University, Kurukshetra

Sunita Rani
Assistant Professor,
Department of CSE & IT,
B.P.S.M.V., Khanpurkalan,
Sonepat, Haryana, India

## ABSTRACT
The resource management system is the central component of cloud computing. A resource management system matches requests to resources, schedules the matched resources, and executes the requests using scheduled resources. Resource allocation and resource scheduling is considered as very important part of resource management system. This paper discusses priority based scheduling technique on heterogeneous resources in cloud computing based upon PERT technology. This scheduling model minimizes the completion time, makespan.

## Keywords
Cloud Computing, resources, scheduling, DAG (Directed Acyclic Graph).

## 1. INTRODUCTION
Cloud computing is a type of parallel and distributed computing which is collection of interconnected and virtualized resources that are dynamically provisioned [1]. Cloud computing is defined by National Institute of Standards and Technology (NIST) [2] as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Cloud computing has various characteristics like virtualization, service orientation, elasticity, dynamicity, distributed environment, sharing, economic, market oriented, pay as you go, autonomic [3] etc. Resource Management is a very important issue in cloud computing and some various important factor such as cost, performance, functionality are affected by resource management. Various cloud computing resource management models has presented like Market Model, Resource Service Model, Application Model [4], Parallel Programming Model [5], Resource Provisioning Model [6], and Hierarchical Resource Management Model [7].

Cloud resource management is mainly concerned with two aspects i.e. resource allocation and task scheduling. Resource allocation is to allocating resources to the needed applications as per the availability of resources. Task scheduling is to schedule jobs on the allocated resources achieving maximum profit, efficient resource utilization and to meet user's QoS requirement. Cloud computing resource management model has interconnected shared resources and interdependent, interrelated tasks that fall into workflow application model [8]. Workflow application model can be represented by Directed Acyclic Graph (DAG), where nodes represent task and edges represent interdependency and relationship between these tasks.

In this paper, authors present a priority based scheduling algorithm based on workflow application and uses PERT (Program Evaluation Review Technique) technique that are used to plan, schedule and control large cloud computing resources with many tasks.

In the PERT, all the input details about cloud and cloud resources are not known with certainty. This approach uses network representation that is represented by Directed Acyclic Graph (DAG) to represent the relationship between various tasks and help resource managers to identify the various queries related to the cloud resources and tasks i.e. total expected completion time of all tasks on cloud, start and completion time for individual tasks, which critical tasks must be completed as scheduled to meet the estimated completion time of cloud, how much delay can be tolerated for non-critical tasks without incurring a delay in the estimated completion time of entire tasks on cloud, to find targeted completion time with minimum cost a for and also gives a probabilistic results for of completing a cloud within a given time frame and variability in the cloud completion time.

This scheduling model uses probabilistically generated task durations by normal distribution. Priority of task is decided by the total completion time of that task and earliest finish time of that task.

This scheduling model is compared with other two algorithms which is based upon PERT techniques, first one is which is PERT model according to this firstly execute all critical tasks that have lowest completion time and then all non-critical tasks with lower completion time. According to second one execute critical task according to their higher priority and then execute non-critical tasks according to their high priority.

In this algorithm execute tasks according to their higher priority from critical and non-critical tasks.

And comparison made among these three considering parameters of completion time and makespan, our proposed priority based algorithm gives better performance in terms of completion time and makespan.

## 2. RELATED WORK
Various resource allocation strategies and resource scheduling and job scheduling algorithms are proposed in cloud computing environment. In [8], a dual objective optimization model for workflow application on grid platforms is presented and its performance is better for completion time and resource

utilization. In [9], authors discussed MapReduce scheduling and its schedulers and also discussed the provisioning of virtual machines to resources in the cloud. In [10], presented a stochastic simulator for optimal cloud resource allocation in a heterogeneous environment that allocating the job to the best machines and minimize makespan and execution cost. In [11],authors presented allocation and scheduling model on workflow application model in cloud contents, considering execution time, cost expenditure and both(execution time and cost). In [12], a cloud computing model without central management node is presented which is exploration of IaaS cloud architecture. In [13], this paper a model is presented for efficient allocation of resource using M-P-S (Memory-Processor- Storage) matrix model. In [14], presented a priority based resource allocation model for cloud computing environment which allocates resources with minimum wastage and provides maximum profit. In [15], a QoS- based workflow scheduling algorithm in grid computing is presented which uses partial critical path method and minimize cost of workflow execution time while meeting a user-defined deadline. In [16], a cost-based resource scheduling algorithm is presented which used market theory to schedule resource to meet user's requirement. In [17][18] a priority based scheduling algorithm is presented for dynamic allocation of preemptable jobs in cloud considering multiple SLA objectives of job.

# 3. SCHEDULING MODEL REPRESENTATION

The scheduling model is modeled by Directed Acyclic Graph (DAG) G (T, E) where T represents the set of n tasks and E represents the set of m directed edges.

A DAG can be called as Task Graph where nodes represent tasks of a workflow application and edges represent precedence constraints and relationship between these tasks. Each edge $t_{ij}=(t_i,t_j) \in E$ between task $t_i$ and $t_j$ represents inter-tasks communication and precedence constraints and relationship which specifies that task $t_i$ must complete its execution before starting task $t_j$.

In the task graph, the immediate predecessor must be completed before starting given task and also immediate successor of a task follows the completion of given task. In a given task graph, a task without any parent (or immediate predecessor) is called an entry task and a task without any child (or immediate successor) is called an exit task.

For this algorithm a single entry task and a single exit task is required, so two dummy tasks $t_{entry}$ and $t_{exit}$ is added to the beginning and end of the task graph, respectively. These dummy tasks have zero execution time and they are connected with zero- weight edges to the actual entry and exit tasks, respectively.

A sequence of connected task nodes in the task graph from entry task to exit task, is called Path. The length of the path is measured by the sum of weights on the task edges on the path.

The path with the longest length is called a critical path and the corresponding task nodes are called critical task nodes in that they must be completed as scheduled to meet the scheduled completion time. A task is called a ready task if its entire predecessor has been completed.

Communication Time CT $(t_i,t_j)$ is the time required for the transmission of data from tasks $t_i$ and $t_j$ and is represented in a matrix form CT $[t_i,t_j]_{n*n}$ , where n is number of tasks. The weight on an edge, $w_{ij}$ represents the communication time CT $(t_i,t_j)$ between two tasks $t_i$ and $t_j$.

A resource r is required for executing a task, set of resources is denoted by R= $\{r_1, r_2, r_3 \ldots r_m\}$ $r(t_j)$ means the resource r that execute task $t_j$.

A task $t_i$ is a logical unit of work that is executed by a resource.

Execution Time gives the estimated execution time to complete task $t_i$ by resource $r_j$ and can be represented in the form of matrix, ET $[t_i,r_j]_{n*m}$ where n is number of tasks and m is number of resources.

Available Time is the earliest time at which resource $r_j$ is ready for task execution means resource $r_j$ can be available if it finishes execution of currently executing tasks and resource $r_j$ cannot be available if it is executing another task. That task has to wait until allocated resource completes the execution of currently executing task. In this case, waiting time, $WT[t_{ij}]$ of task $t_i$ for resource $r_j$ is zero, if resource available instantly means resource does not executing another task. Otherwise $WT[t_{ij}]$ will be the execution time of currently executing task. Availability of resources follows the exponential distribution.

In this paper, scheduling model is represented with probabilistic task durations where task durations are not with certainty and can be known through normal probability distributions.

# 4. SCHEDULING MODEL

## 4.1 Problem Definition:

Priority based scheduling algorithm for cloud computing resources allocates resources in effective, efficient and optimized way. This algorithm assumes that a set of resources is geographically distributed and heterogeneous in nature; priority is assigned to tasks on the basis of completion time and earliest finish time of task.

## 4.2 Probabilistic task durations of scheduling model:

In this approach, duration of tasks is not known with certainty. The duration of these tasks is assumed to have normal distribution, following this distribution task duration is to be, $x = \sigma * s + \mu$;

Where, x is task duration,

σ is sigma(standard deviation),

μ is mue (mean value),

s is a sample from the standardized normal distribution.

In our algorithm (program code) code is written in matlab as

```
P=sigma(kk)*randn(1,10000)+mue(kk);
```

Where, kk is number of runs or number of tasks duration,

randn() is a predefined function in matlab that generates random number using normal distribution,

randn(1,10000) means that this function generate ten thousand random number based on normal distribution,

And we choose one random number based on normal distribution as task duration for our scheduling model.

## 4.3 Notations and variable definition

In this scheduling model, the communication times of tasks are known through probability that follows the normal distribution.

Let EST $(t_i,r_j)$ is the earliest execution start time of task $t_i$ on resource $r_j$.

EFT $(t_i,r_j)$ is the earliest execution finish time of task $t_i$ on resource $r_j$.

LST $(t_i,r_j)$ is the latest execution start time of task $t_i$ on resource $r_j$.

LFT $(t_i,r_j)$ is the latest execution finish time of task $t_i$ on resource $r_j$.

pred$(t_i)$ is the set of immediate predecessor of task $t_i$.

succ$(t_i)$ is the set of immediate successor of task $t_i$.

ST$[t_i]$ Slack Time of task $t_i$.

CP is Critical path

CT is Critical Task

$$\text{EST}(t_i,r_j) = \begin{cases} 0, & \text{if ti = entry task} \\ \text{CT (ti, tj)} + \max\{\text{EFT (tp)}\}, & \text{otherwise} \\ \quad \text{tp} \in \text{pred(ti)} & \end{cases}$$

EFT $(t_i,r_j)$ = EST $(t_i,r_j)$+ CT $(t_i,t_j)$

LST $(t_i,r_j)$ =LFT $(t_i,r_j)$ - CT $(t_i,t_j)$

$$\text{LFT}(t_i,r_j) = \begin{cases} \text{EFT (ti, rj)}, & \text{if ti = exit task} \\ \min\{\text{LFT (tp, rj)}\} - \text{CT (ti, tj)}, & \text{otherwise} \\ \text{tp} \in \text{succ(ti)} & \end{cases}$$

The slack time ST$[t_i]$ is the length of time that can be tolerated without causing an overall delay in the completion time. The slack time per task is calculated by the difference of LST and EST.

Slack Time = LST-EST;

Those tasks have zero slack time are called critical tasks and belongs to critical path and these tasks cannot be delayed without delaying the entire workflow.

The remaining tasks are non-critical task and these tasks provide freedom to managers about when to start or complete task without delaying the workflow. It is also possible for a workflow to have multiple critical path(s). A task with smaller slack time will need tighter control than a task with a large slack time.

The purpose of critical path analysis is to identify critical tasks so that resources may be concentrated on these tasks in attempt to reduce cloud project finish time.

Completion Time= minimum execution time of task $t_i$ on resource $r_{j\,+}$Available time of resource $r_{j+}$ Earliest finish time of task

The completion time of each task $t_i$ in the workflow is the minimum execution time of task $t_i$ on resource $r_{j.}$

**Priority Function**:
scheduling priority function is defined by using the completion time of each task and the earliest finish time of a task.
Priority Function= CompletionTime$_i$/ EFT$_{ij}$, where EFT$_{ij}$>0
According to our priority function, higher priority is assigned to those tasks whose completion time is higher and earliest finish time of task is low. Critical tasks have high priority as compared to non-critical tasks.

## 4.4 Scheduling Algorithm

1. Input n task and m resources
2. Input a list of n tasks in the DAG
3. Generate randomly Communication Time CT $[t_i,t_j]_{n*n}$
4. Generate randomly Execution Time ET $[t_i,r_j]_{n*m.}$
5. Generate randomly Available Time of Resources AT $[r_j]_m$.
6. For all task i = 1 to n

Find resource with minimum execution time for each task

Find Available time of that resource with minimum execution time

if ($t_i$= entry task)

then EST$(t_i,r_j)$ = 0

else

EST $(t_i,r_j)$ = CT (ti, tj)+max(EFT$(t_p)$)

$t_p \in \text{Pred}(ti)$

where, $\text{Pred}(ti)$ is the set of immediate predecessor of task $t_{i.}$

EFT $(t_i,r_j)$ = EST $(t_i,r_j)$+ CT (ti, tj)

if ($t_i$= exit task)

then LFT$(t_i,r_j)$ = EFT$(t_i,r_j)$

else

LFT $(t_i,r_j)$ = min(LFT$(t_p, r_j)$) - CT (ti, tj)

$t_p \in \text{Succ}(ti)$

where, $\text{succ}(ti)$ is the set of immediate successor of task $t_{i.}$

LST $(t_i,r_j)$ = LFT $(t_i,r_j)$ - ET $(t_i,r_j)$

7. Find slack time for all tasks

   ST(i)= LST(i)-EST(i);

8. If(ST(i)==0)

   Add to critical path and critical nodes

   else

   Add to non-critical path and Non- Critical nodes

9. Find completionTime for all task i = 1 to n

   CompletionTime$_i$ = At(rj)+EFT $(t_i)$+ET$(t_i)$

10. For all task $t_i$ i= 1 to n

    Priority value $(t_i) = \frac{\text{CompletionTime} \quad i}{\text{EFTi}}$

11. Sort in decreasing order of Priority value $(t_i)$ for all task

12. For all task who has higher priority value and belongs to critical node,

    Execute that task

    CompletionTimeW=CompletionTime$_i$+ CompletionTimeW

13. Makespan = completionTimeW
14. End for

## 5. RESULTS AND DISCUSSION

In this paper, priority based scheduling model has proposed and its performance is evaluated. Simulation is performed using MATLAB. A comparison is made among three algorithms, first one is based upon our proposed priority algorithm, second is based upon PERT techniques, and third one is based upon priority and PERT based scheduling algorithm. According to priority algorithm execute tasks according to their priority defined, second is based upon PERT model according to this firstly execute all critical tasks that have lowest completion time and then all non-critical tasks with lower completion time and third one is execute critical task according to their higher priority and then execute non-critical tasks according to their high priority considering various objective parameters i.e., completion time, makespan.

Here we have made two analyses for the proposed algorithm; one is based on small network and second is large network.

**ANALYSIS 1:**

Firstly, simulation is performed for small workflow network for DAG as shown in figure 1 that have eight numbers of nodes (or tasks) and eleven numbers of activities among these tasks.

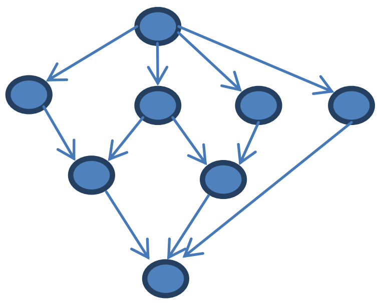

**Figure 1: Example small workflow for DAG**

Mue and Sigma is given here that is used to generate normally distributed communication time among tasks

Mue= [7.3 8.8 2.3 8.0 5.1 7.1 13.6 2.2 16.12 17.11 3.6];

Sigma= [2.0 1.0 2.9 1.0 2.0 1.4 0.8 1.2 1.8 0.3 1.75];

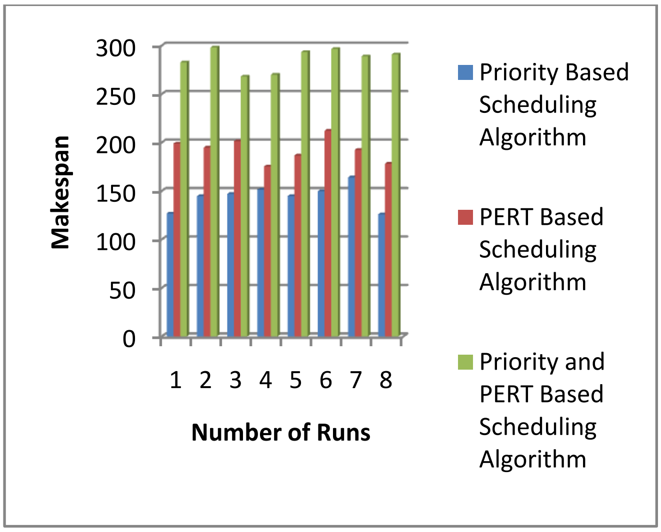


**Figure 2: Makespan**

In figure 2 shows that if tasks are executed according to priority based scheduling model then makespan is less as compared to when tasks are executed according to above discussed algorithm.

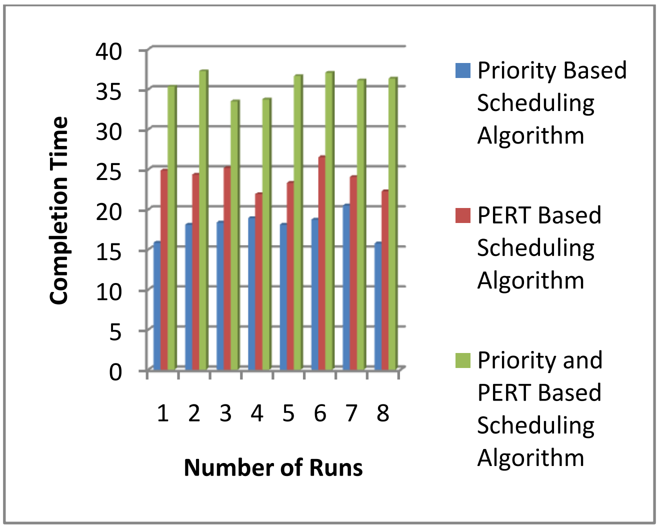


**Figure 3: Completion Time**

In figure 3 shows that if tasks are executed according to priority based scheduling model then completion time is less as compared to when tasks are executed according to above discussed algorithm.

**ANALYSIS 2:**

Secondly, simulation is performed for large workflow network for DAG as shown in figure 4 that have sixteen numbers of nodes (or tasks) and thirty numbers of activities among these tasks.

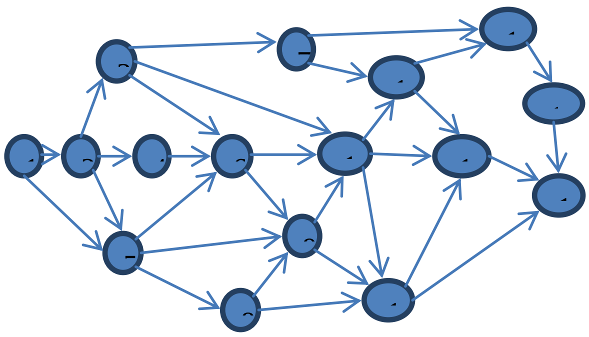

**Figure 4: Example small workflow for DAG**

Mue and Sigma is given here that is used to generate normally distributed communication time among tasks.

mue=[5 7 4 8.5 8 8 5 10 11 7.5 7 20 9.5 7 5 3 11 7.5 5 9 6 6 7 8 8 10 15 4.5 4 4 ];

sigma=[1.22 0.5 0.95 2.05 1.65 0.75 1.0 2.2 3.15 1.15 1.52 4.2 3.0 0.25 0.9 0.2 2.75 2.35 1.75 3.0 1.83 1.15 2.25 2.6 2.25 2.75 3.38 1.0 1.5 1.55];

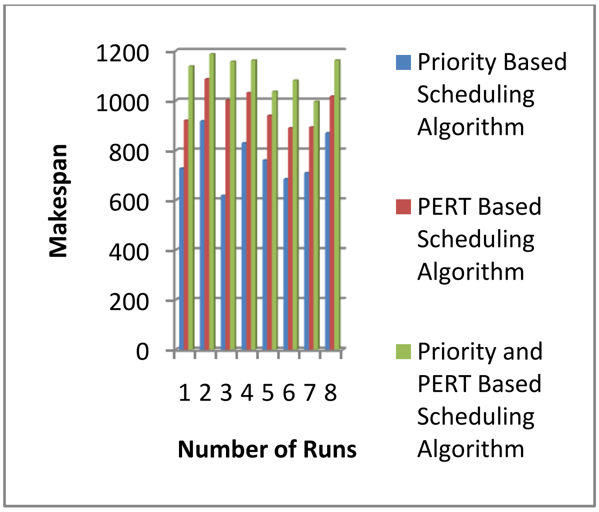


**Figure 5: Makespan**

In figure 5 shows that if tasks are executed according to priority based scheduling model then makespan is less as compared to when tasks are executed according to above discussed algorithm.

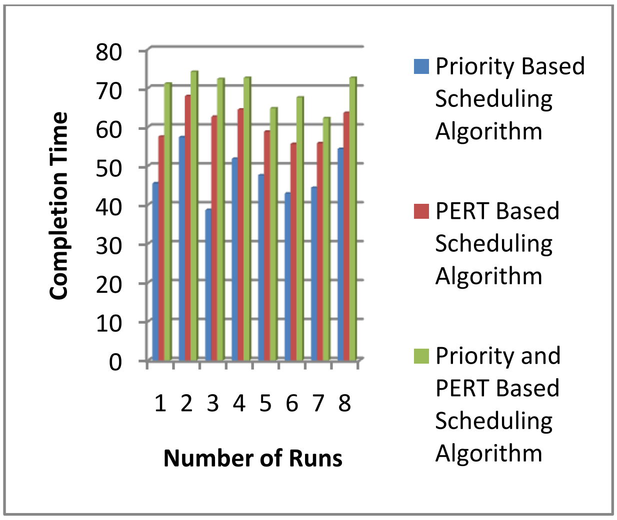


**Figure 6: Completion Time**

In figure 6 shows that if tasks are executed according to priority based scheduling model then completion time is less as compared to when tasks are executed according to above discussed algorithm.

**ANALYSIS 3:**

Third Analysis is based upon number of tasks in the workflow network (as shown in Figure 1 and Figure 4). In figure 7 and figure 8 shows that large workflow network has small makespan and small completion time as compared to small workflow network. As the size of workflow network increases makespan and completion time decreases.

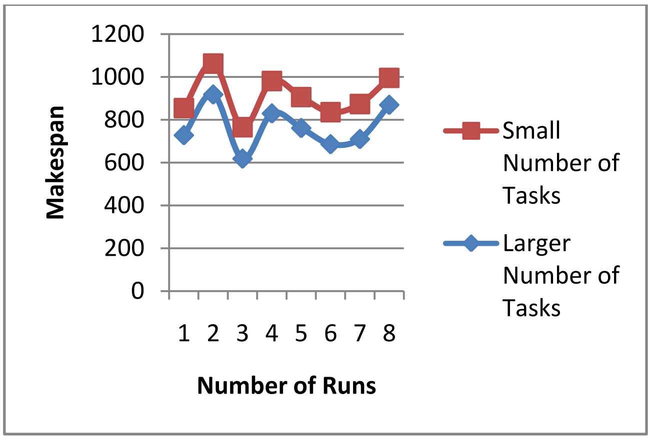


**Figure 7: Makespan**

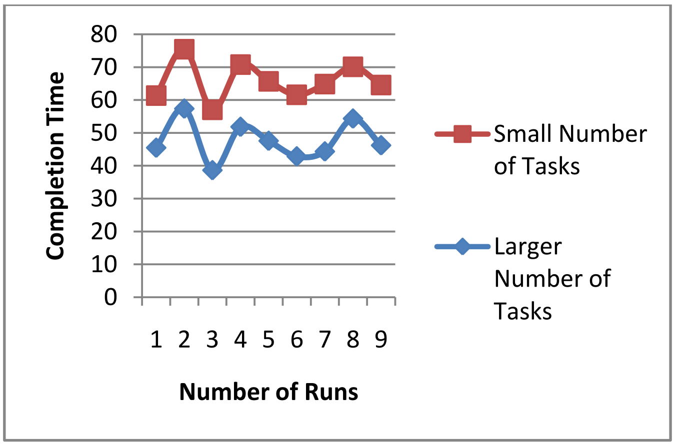


**Figure 8: Completion Time**

## 6. CONCLUSION

Resource management is one of the important issues to be solved in cloud computing environment. In this paper, a priority based scheduling model is proposed and compared with other scheduling model considering some objective parameter i.e. completion time, makespan. Simulation results demonstrate the efficiency and effectiveness of the proposed model. The proposed model reduces the completion time, makespan. In future work criticality and priority will be considered to make more effective and more efficient model in resource management in cloud computing.

## 7. REFERENCES

[1] Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. “Cloud computing and emerging IT platforms: Vision, type, and reality for delivering computing as the 5th utility”, Future Generation Computer Systems, 599–616, 2009.

[2] Mell, P. and Grance, T., “The NIST Definition of Cloud computing, National Institute of Standards and Technology”, 2009.

[3] Saurabh Kumar Garg, Rajkumar Buyya, “Green Cloud computing and Environmental Sustainability”,

[4] Mengkun Li, Ming Chen, lun Xie, “Cloud Computing: A Synthesis Models for Resource Service Management”, Second International Conference on Communication Systems, Networks and Applications, IEEE, pp. 208-211, 2010.

[5] Liu xuning, Song hongwei, He dongbin, Yang hao, "Research of Campus Resource Management Based on Cloud Computing", The 5th International Conference on Computer Science & Education Hefei, China, pp. 1407-1409, August 24–27, 2010.

[6] Seoyoung Kim, Jung-in Koh, Yoonhee Kim, Chongam Kim, "A Science Cloud Resource Provisioning Model using Statistical Analysis of Job History", Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing, IEEE Computer Society, IEEE,pp. 792-793, 2011.

[7] Ping Guo, Ling-ling Bu, "The Hierarchical Resource Management Model Based on Cloud Computing", IEEE Symposium on Electrical & Electronics Engineering (EEESYM), IEEE, pp. 471-474, 2012.

[8] Young Choon Lee, Riky Subrata, and Albert Y. Zomaya, "On the Performance of a Dual-Objective Optimization Model for Workflow Applications on Grid Platforms", IEEE transactions on parallel and distributed systems, VOL. 20, NO. 9, pp. 1273-1283, SEPTEMBER 2009.

[9] Eman Elghoneimy, Othmane Bouhali, Hussein Alnuweiri,"Resource Allocation and Scheduling in Cloud Computing", Workshop on Computing, Networking and Communications", IEEE,pp. 309-314, 2012.

[10] P. K. Suri, Himanshi Goyal, "Stochastic Simulator for Optimal Cloud Resource Allocation in a Heterogeneous Environment", International Journal of Computer Applications (0975 – 8887) Volume 101– No.2,pp. 9-13, September 2014.

[11] Kahina Bessai, Samir Youcef, Ammar Oulamara, Claude Godart and Selmin Nurcan, "Resources allocation and scheduling approaches for business process applications in Cloud contexts", 4th International Conference on Cloud Computing Technology and Science, IEEE Computer Society, pp. 496-503,IEEE, 2012.

[12] Zongheng Liu, Weiqin Tong, Zhixun Gong, Jianhua Liu, Yue Hu, Shaojie Guo, "Cloud Computing Model Without Resource Management Center", 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, IEEE Computer Society, pp.442-446, IEEE, 2011.

[13] T.R. Gopalakrishnan Nair, Vaidehi M, "Efficient Resource Arbitration And Allocation Strategies In Cloud Computing Through Virtualization",Proceedings of IEEE CCIS2011, pp.397-401, IEEE, 2011.

[14] K C Gouda, Radhika T V, Akshatha M, "Priority based resource allocation model for cloud computing", International Journal of Science, Engineering and Technology Research (IJSETR) Volume 2, Issue 1, ISSN: 2278 – 7798, pp. 215-219, January 2013.

[15] Saeid Abrishami, Mahmoud Naghibzadeh, and Dick H.J. Epema, "Cost-Driven Scheduling of Grid Workflows Using Partial Critical Paths",IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. 8, IEEE Computer Society, pp.1400-1414, AUGUST 2012.

[16] Zhi Yang, Changqin Yin, Yan Liu, "A Cost-based Resource Scheduling Paradigm in Cloud Computing", 12th International Conference on Parallel and Distributed Computing, Applications and Technologies, IEEE Computer Society, pp.417- 422, IEEE, 2011.

[17] Chandrashekhar S. Pawar, Rajnikant B. Wagh, "Priority Based Dynamic Resource Allocation in Cloud Computing with Modified Waiting Queue", International Conference on Intelligent Systems and Signal Processing (ISSP), pp. 311-316, IEEE, 2013.

[18] Chandrashekhar S. Pawar, Rajnikant B. Wagh, "Priority Based Dynamic resource allocation in Cloud Computing", International Symposium on Cloud and Services Computing, IEEE Computer Society, pp. 1-6, IEEE, 2012.