

Quality of Service based Task Scheduling Algorithms in Cloud Computing

Sirisha Potluri¹, Katta Subba Rao²

¹Department of CSE, KL University, Green Fields, Vaddeswaram, Guntur, Andhra Pradesh 522502, India,

²Department of CSE, KL University, Green Fields, Vaddeswaram, Guntur, Andhra Pradesh 522502, India

Article Info

Article history:

Received Oct 26, 2016

Revised Jan 3, 2017

Accepted Jan 27, 2017

Keyword:

Cloud computing

Quality of service

Task scheduling

ABSTRACT

In cloud computing resources are considered as services hence utilization of the resources in an efficient way is done by using task scheduling and load balancing. Quality of service is an important factor to measure the trustiness of the cloud. Using quality of service in task scheduling will address the problems of security in cloud computing. This paper studied quality of service based task scheduling algorithms and the parameters used for scheduling. By comparing the results the efficiency of the algorithm is measured and limitations are given. We can improve the efficiency of the quality of service based task scheduling algorithms by considering these factors arriving time of the task, time taken by the task to execute on the resource and the cost in use for the communication.

*Copyright © 2017 Institute of Advanced Engineering and Science.
All rights reserved.*

Corresponding Author:

Sirisha Potluri,

Department of CSE,

KL University,

Green Fields, Vaddeswaram, Guntur, Andhra Pradesh 522502, India,

Email: sirisha.vegunta@gmail.com

1. INTRODUCTION

Interface and infrastructure together forms a cloud environment [1]. The resources are provided by the cloud service provider and user can access them as a service by using the service models (PaaS, IaaS, SaaS). The organization can deploy the cloud based on their requirement by using deployment models (Private, Public, Community, and Hybrid) [2]. Even though cloud is a good servant to the user in turn many security issues involved in it. It involves risk to put all the data in cloud and application runs at different place [3]. The cloud architecture contains Users, SLA resource locator, VMs and Physical machines [4]. In cloud computing resources are considered as services hence utilization of the resources is done in an efficient way by using task scheduling and load balancing [5]. In cloud computing multiple users sends the requests in same time hence using scheduling the resources are allocated to the tasks [6].

Quality of service is an important factor to measure the user's satisfaction in cloud environment. Using quality of service in task scheduling will address the problems of security in cloud computing. Parameters considered for the algorithms studied in this paper are QoS, make span, accepted rate, cost, fairness completion time, minimum completion time. The existing algorithms can be improve the efficiency by considering these factors arriving time of the task, time taken by the task to execute on the resource and the cost in use for the communication. In the following section quality of service based task scheduling algorithms are studied and compared their advantages and disadvantages.

2. QoS BASED TASK SCHEDULING ALGORITHMS

2.1. QoS guided Min-Min heuristic for Grid task scheduling

In grid computing the task scheduling algorithm should address the issues like security, QoS and central control over data to get high throughput from the system. As shown in Figure 1, this algorithm is a QoS guided based on general adaptive scheduling characteristics. The task may or may not have QoS; task with no QoS can be executed on any resource. The task having high QoS can be executed on a resource with high QoS. So the task with low QoS can be executed on a resource with high QoS resulting resources with low QoS remains idle [7].

QoS is affects the effectiveness of the computing environment, for ex: Assume the network is having high bandwidth and if the scheduler assigns the task which does not require high bandwidth, meanwhile the tasks requiring high bandwidth have to wait in the queue. By considering QoS factors in scheduling gives a better scheduling algorithm [8]. In this algorithm instead of mapping the tasks to the hosts, here mapping is done first for the tasks with high QoS.

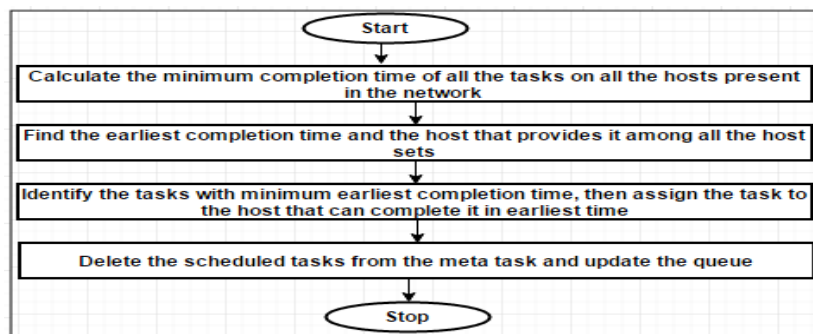


Figure 1. QoS guided Min-Min heuristic for Grid task scheduling

Results: Parameters considered: Make span and Expected time of completion. Advantage: Reduced the make span and used the bandwidth parameter. Disadvantage: Poor load balancing and QoS factors are not considered

2.2. Job scheduling algorithm based on Berger model in cloud computing

Cloud computing is a combination of parallel and grid computing. Virtualization hides all the differences between different physical devices present on the cloud. The main entities in cloud computing are users, resource providers and scheduling system [9]. The scheduling factor for this algorithm is fairness constraint [10].

1. QoS based task classification: The user's tasks can be classified based on QoS parameters like Completion time and Bandwidth.
2. Fairness constraint: The fairness of resource makes the cloud to provide reasonable resources which are available to execute the user's tasks.
3. General expectations constraint: The tasks require the resources to complete it. Because of different characteristics of the user's applications the tasks are having QoS preferences. In local structure, based on the general expectations we can optimize the selection process of the resources.
4. Description of tasks and resources: Cloud computing uses the technology called virtualization to use the resources. The scheduling in cloud computing is implemented in application layer and virtual machine layer. Scheduling is to map a task to the resource.
5. Tasks and resource mapping: Keeping QoS parameters in mind the user's tasks are mapped to the resources. The ratio of expectations of resources and the actual allocations gives the justice.
6. Completion time: The completion time of a task is the combination of waiting time, execution time and sending time.
7. Bandwidth: It is very useful in applications where frequent communication is happening.
8. Integrated general expectation: If the task needs many QoS requirements then integrated general expectation is suitable.

As shown in Figure 2, The algorithm is stated as follows:

1. According to Quality of Service classification, the tasks having general expectation constraint acts as the constraints of fairness to select and allocate a resource are established.

2. Virtual machine selects the better resources to run the task by taking parameterized task characteristics and general expectation constraint.
3. Calculate user satisfaction, accordingly adjust the model.

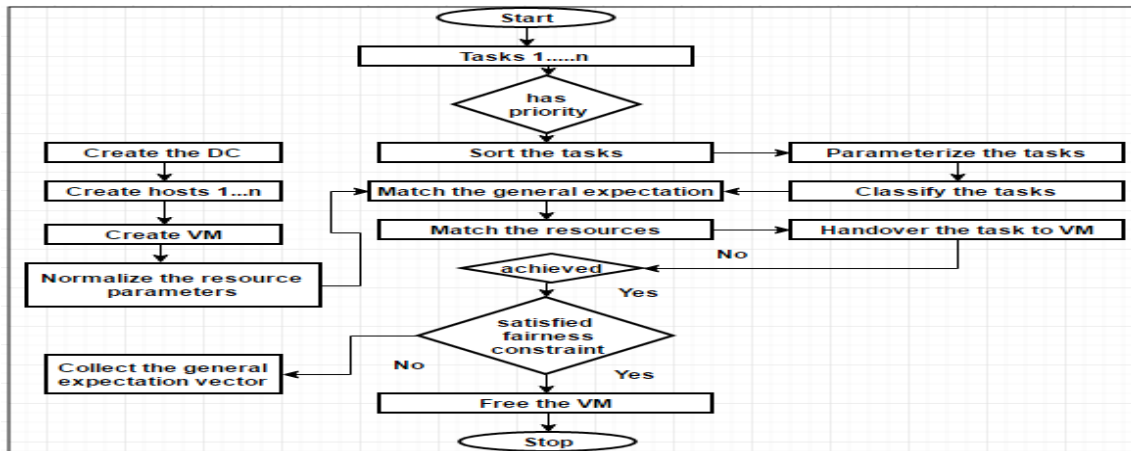


Figure 2. Job scheduling based on Berger model

Results: Parameters considered: QoS, fairness completion time. Advantage: Improved the performance and task execution. Disadvantage: Rescheduling of tasks at each level will increase the complexity of the algorithm and takes more time

2.3. Improved cost based algorithm for task scheduling in cloud computing

This algorithm is useful to measure the computation performance and cost of resources. Virtual machines are used to run the applications because the resources are distributed virtually in cloud computing. Some application may require more CPU time and some applications may require more memory to store the data. Resources are used to perform each task. To measure the cost of the application measuring CPU cost, Input cost, Output cost, memory cost are required [11]. Using Customer Facilitated Cost-based Scheduling (CFCSC) algorithm we can balance the load and cost. This algorithm uses cost function to reduce input, output and monetary costs [12]. As shown in Figure 3 the algorithm can be stated as follows:

Take T_i is set of n independent tasks and R_j is set of m computing resources where $i=\{1,2,3,\dots,n\}$, $j=\{1,2,3,\dots,m\}$. The processing capacity of a resource can be measured as MIPS and the size of the task can be measured as MI. The total computing time can be calculated as $TT_{totime}(R)=TT_{exe}(R)+TT_{commtime}(R)$. Where $TT_{totime}(R)$ represents total time, $TT_{exe}(R)$ represents the total computation time of all resources and $TT_{commtime}(R)$ represents total communication time.

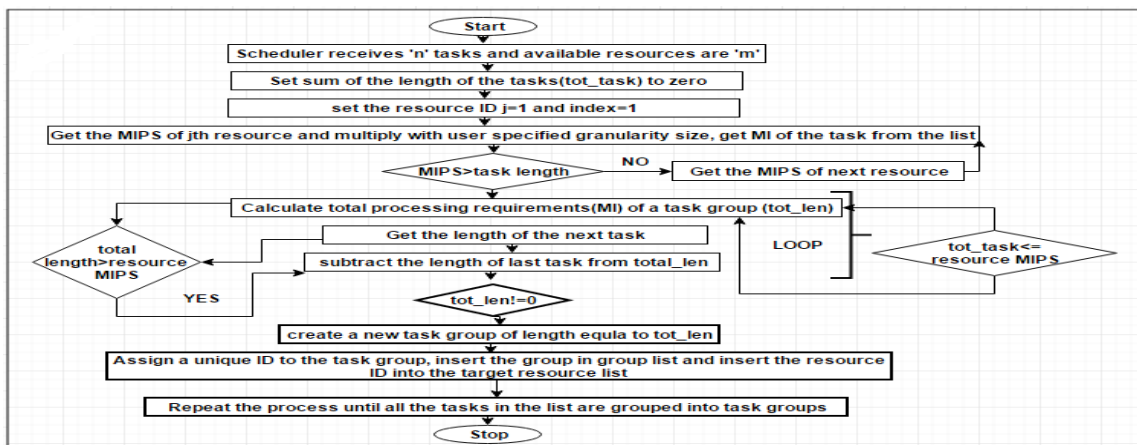


Figure 3. Improved cost based algorithm for task scheduling in cloud computing

Results: Parameters considered: Cost, make span. Advantage: Improved the computation and reduced resource cost. Disadvantage: This algorithm not considered the dynamic cloud environment and mainly focused on cost.

2.4. RASA

RASA algorithm considers distribution and characteristics of scalability of grid resources [13]. This algorithm is taking the advantages of Min-min and Max-min task scheduling algorithms and covering their disadvantages. As shown in Figure 4, this algorithm calculates the completion time of the tasks by taking all available resources. Then it applies Max-min and Min-min algorithms alternatively, for small tasks it uses Min-min and for large ones it applies Max-min to avoid the delays in large tasks execution [14]. The algorithm builds a matrix called M . M_{ij} represents the completion time of the task. T_i task and R_j resources.

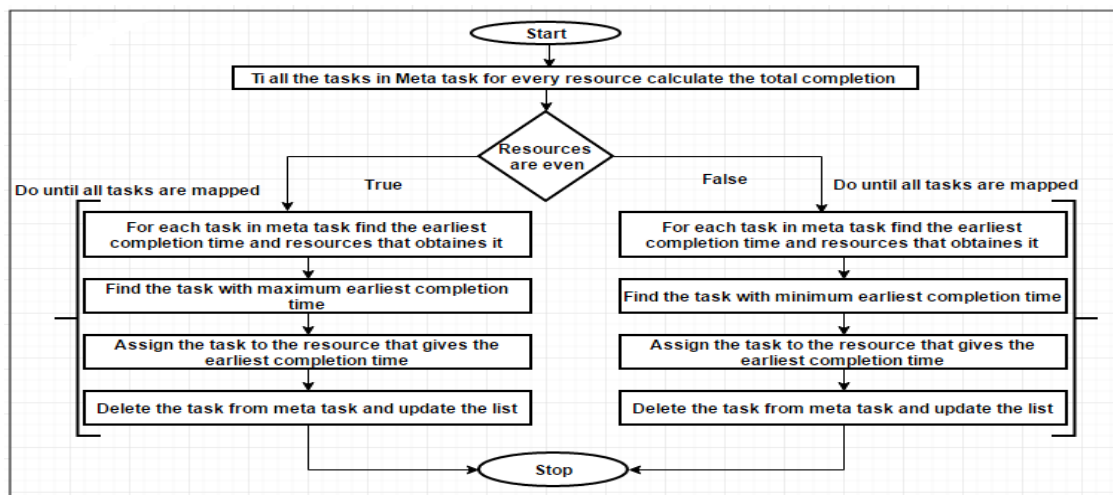


Figure 4. RASA

Results: Parameters considered: Make span. Advantage: Reduced the make span. Disadvantage: More emphasis can be given to QoS attributes; algorithm should consider the heterogeneous environments.

2.5. A QoS based Predictive Max-Min, Min-Min switcher algorithm for job scheduling in a grid

The history information about the execution of tasks is taken into consideration to predict the performance of the resources available. As shown in Figure 5, this algorithm selects the best algorithm between QoS Min-Min and QoS Max-min by taking length of the job into consideration. Based on this decision, the tasks having high QoS are mapped first then jobs with low QoS are mapped. The algorithm calculates the standard deviation of all unassigned jobs. Then it will identify the position where the difference between the completions times of two consecutive jobs. If the difference is more than standard deviation and if it is present in the first half contains the number of jobs more than the number of short jobs. Then min-min algorithm outperforms max-min, so min-min is selected to map the jobs. If the position is present in the second half means max-min outperforms the min-min, so max-min is selected to map the jobs. If the position is not existed then is compared with threshold value, if it is less means min-min is used otherwise max-min is used [15]. Improved met-tasking algorithm in grid computing schedules the tasks based suffrage value. Based on suffrage value min-min or max-min algorithm is implemented. This met-tasking algorithm uses these parameters: flow time, make span, resource utilization [16].

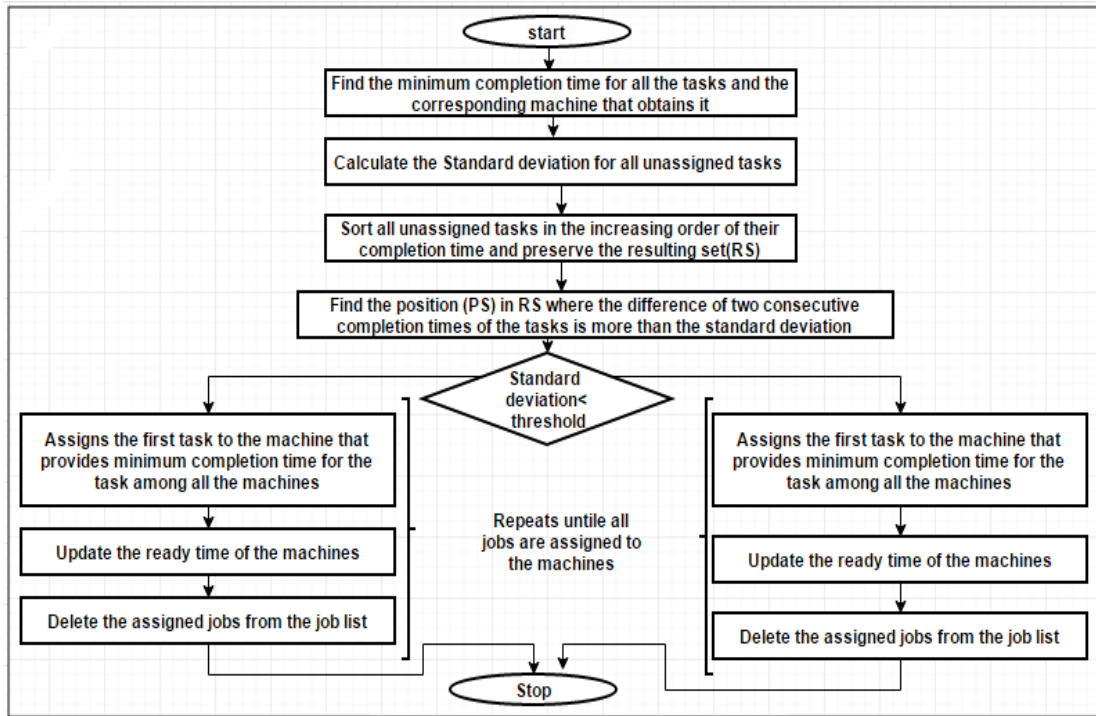


Figure 5. A QoS based Predictive Max-Min, Min-Min scheduler algorithm for job scheduling in a grid

Results: Parameters considered: Make span. Advantage: Improved performance with QoS. Disadvantage: Switching results in taking more time and involving more cost

2.6. Towards improving QoS-guided scheduling in grids

As shown in Figure 6, to resolve the problem of scheduling the tasks in heterogeneous systems this algorithm is giving two optimization schemes based on QoS Min-Min scheduling technique. Those are named as Make span optimization Rescheduling and Resource optimization rescheduling [17].

Make span optimization rescheduling (MOR): The aim of this technique to improve the make span to achieve the better performance. Resource optimization rescheduling (ROR): The aim of this technique to improve the resource optimization to achieve the better performance.

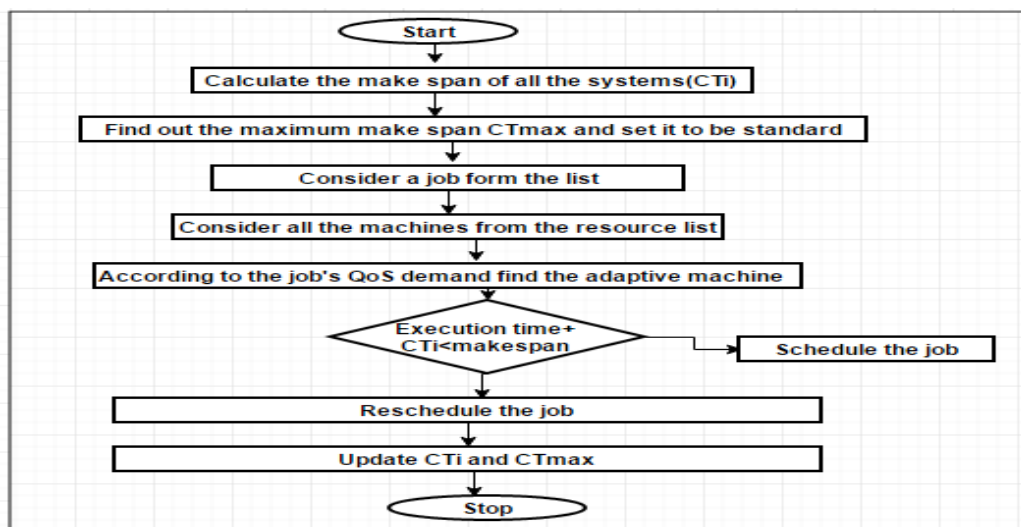


Figure 6. Towards improving QoS-guided scheduling in grids

Results: Parameters considered: Make span. Advantage: Improved the make span and by using rescheduling reduced the resource need. Disadvantage: This algorithm results poor load balancing for dynamic cloud environment and can be improved by considering QoS attributes

2.7. A grid task scheduling algorithm based on QoS priority grouping

As shown in Figure 7, this algorithm is used to group the grid tasks based on the QoS. This algorithm uses deadline property of the task, tasks acceptance rate and make span of the computing systems. n is the number of tasks in the grid environment, so the tasks can be grouped into n groups. Using Sufferage algorithm the tasks are grouped in descending order of their QoS [18]. The algorithm is as follows:

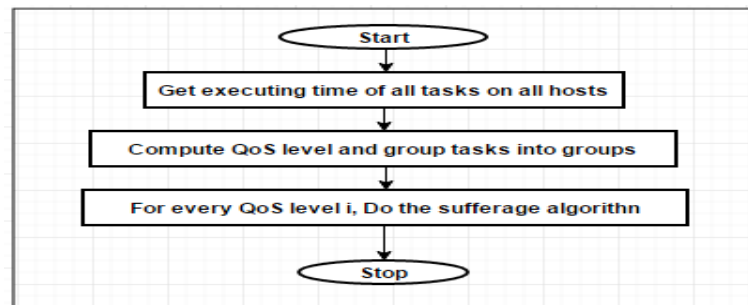


Figure 7. A grid task scheduling algorithm based on QoS priority grouping

Results: Parameters used: Acceptance rate, completion time. Advantage: Reduced the make span. Disadvantage: Can be improved by considering Make span, QoS attributes like consistency to improve the efficiency and hence complexity can be reduced.

2.8. A Task Scheduling Algorithm based on QoS-Driven in Cloud Computing

As shown in Figure 8, this algorithm computes the priority of the tasks on different services and executes the task on the machine which can complete it as soon as possible. This algorithm uses dual fairness constraint. RRank means a reliability priority rank is used in this algorithm to estimate the priority of the task. In this model tasks are not directly mapped instead they were collected and stored in a queue. The tasks having higher priority will be executed first and task should be completed in less time as soon as possible [19].

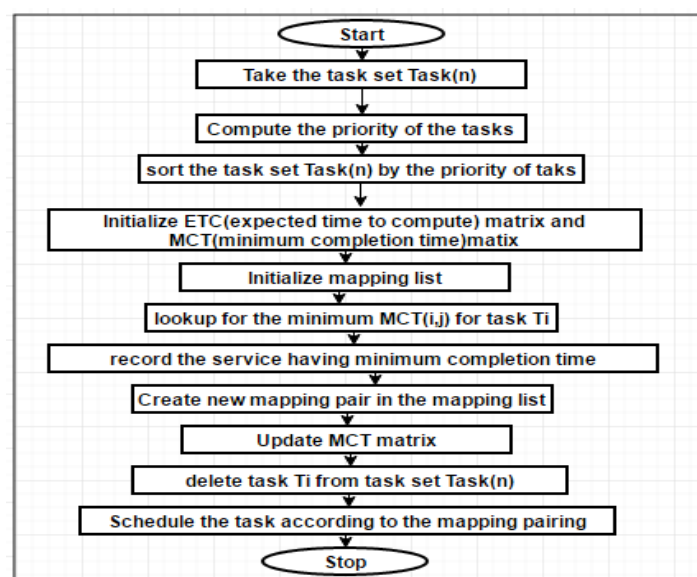


Figure 8. A Task Scheduling Algorithm based on QoS-Driven in Cloud Computing

Parameters used: Priority of task and minimum completion time. Results: Improved performance and load balancing is implemented. Disadvantage: This algorithm should consider the failure occurs in a machine and dynamic cloud environment

3. RESULTS AND ANALYSIS

Task Scheduling is the primary and key issue in cloud computing environment. The existing algorithms are based on Quality of service and the limitations are mentioned I Table 1:

Table 1. The existing algorithms are based on Quality of service and the limitations

S.No	Algorithm	Limitations
1	QoS guided Min-Min heuristic for Grid task scheduling	Load balancing is poor and QoS attributes are not used in algorithm
2	Job scheduling algorithm based on Berger model in cloud computing	Rescheduling of tasks at each level will increase the complexity of the algorithm and takes more time
3	Improved cost based algorithm for task scheduling in cloud computing	This algorithm not considered the dynamic cloud environment and mainly focused on cost.
4	RASA	More emphasis can be given to QoS attributes; algorithm should consider the heterogeneous environments.
5	A QoS based Predictive Max-Min, Min-Min switcher algorithm for job scheduling in a grid	Switching results in taking more time and involving more cost
6	Towards improving QoS-guided scheduling in grids	This algorithm results poor load balancing and can be improved by considering QoS attributes
7	A grid task scheduling algorithm based on QoS priority grouping	Can be improved by considering Make span, QoS attributes like consistency to improve the efficiency and hence complexity can be reduced
8	A Task Scheduling Algorithm based on QoS-Driven in Cloud Computing	This algorithm should consider the failure occurs in a machine and dynamic cloud environment

4. CONCLUSION

As cloud services are increasing day by day. To meet the on demand service and to maintain efficiency of load balancing and task scheduling of resources many algorithms are proposed. The existing task scheduling algorithms are studied in this paper. . We can improve the efficiency of the quality of service based task scheduling algorithms by considering these factors arriving time of the task, time taken by the task to execute on the resource and the cost in use for the communication.

REFERENCES

- [1] A. Kumar, "World of Cloud Computing & Security," *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*, vol/issue: 1(2), pp. 53-58, 2012.
- [2] R. Buyya, "Introduction to the IEEE Transactions on cloud computing," *IEEE Transactions on Cloud Computing*, vol/issue: 1(1), 2013.
- [3] S. Pal, *et al.*, "Efficient Architectural Framework for Cloud Computing," *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*, vol/issue: 1(2), pp. 66-73, 2012.
- [4] R. Buyya, *et al.*, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, pp. 599-616, 2009.
- [5] Vijindra, *et al.*, "Survey on Scheduling Issues in Cloud Computing," International conference on modeling optimization and computing (ICMOG-2012), Procedia Engineering, vol. 38, pp. 2881-2888, 2012.
- [6] V. S. Rathor, *et al.*, "Survey on Load Balancing Through Virtual Machine Scheduling in Cloud Computing Environment," *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*, vol/issue: 3(1), pp. 37-43, 2014.
- [7] H. E. Xiaoshan, *et al.*, "QoS Guided Min-Min Heuristic for Grid Task Scheduling," *Journal of Computer Science and Technology*, vol/issue: 18(4), pp. 442-451, 2003.
- [8] S. S. Chauhan, *et al.*, "QoS Guided Heuristic Algorithms for Grid Task Scheduling," *International Journal of Computer Applications*, vol/issue: 2(9), 2010.
- [9] B. Xu, *et al.*, "Job scheduling algorithm based on Berger model in cloud environment," *Advances in Engineering Software*, vol. 42, pp. 419-425, 2011.
- [10] D. S. Kalra, *et al.*, "Differentiating Algorithms of Cloud Task Scheduling Based on various Parameters," *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol/issue: 17(6), pp. 35-38, 2015.
- [11] S. Selvarani, *et al.*, "Improved Cost-Based Algorithm For Task Scheduling In Cloud Computing," *IEEE Xplore*, 2010.

- [12] D. I. G. Amalarethinam, *et al.*, "Customer Facilitated Cost-based Scheduling (CFCSC) in Cloud," *International Conference on Information and Communication Technologies (ICICT 2014)*, *Procedia Computer Science*, vol. 46, pp. 660 – 667, 2015.
- [13] S. Parsa, *et al.*, "RASA: A New Task Scheduling Algorithm in Grid Environment," *World Applied Sciences Journal*, vol. 7, pp. 152-160, 2009.
- [14] S. Parsa, *et al.*, "RASA: A New Grid Task Scheduling Algorithm," *International Journal of Digital Content Technology and its Applications*, vol/issue: 3(4), 2009.
- [15] M. Singh, *et al.*, "A QoS based predictive Max-Min, Min-Min switcher algorithm for job scheduling in a grid," *Information Technology Journal*, vol/issue: 7(8), pp. 1176-1181, 2008.
- [16] N. M. Reda, "An Improved Sufferage Meta-Task Scheduling Algorithm in Grid Computing Systems," *International Journal of Advanced Research*, vol/issue: 3(10), pp. 123 -129, 2015.
- [17] C. H. Hsu, *et al.*, "Towards Improving QoS-Guided Scheduling in Grids," *The Third ChinaGrid Annual Conference*, 2008.
- [18] F. Dong, *et al.*, "A Grid Task Scheduling Algorithm Based on QoS Priority Grouping," *Grid and Cooperative Computing, (GCC), Fifth International Conference*, 2006.
- [19] X. Wu, *et al.*, "A Task Scheduling Algorithm based on QoS-Driven in Cloud Computing," *First International Conference on Information Technology and Quantitative Management, Procedia Computer Science*, vol. 17, pp. 1162-1169, 2013.

BIOGRAPHIES OF AUTHORS



Sirisha Potluri

Research Scholar,
Department of CSE,
KL University,
Green Fields, Vaddeswaram, Guntur, Andhra Pradesh 522502, India,
Email: sirisha.vegunta@gmail.com



Dr. Katta Subba Rao

Professor,
Department of CSE,
KL University,
Green Fields, Vaddeswaram, Guntur, Andhra Pradesh 522502, India,
Email: subbarao_cse@kluniversity.in