www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 7 July, 2014 Page No. 6866-6870

A Review of Cloud Based Schedulers on Cloud Computing Environment

Renu Bala[1], Gagandeep Singh[2],Sahil Vashist[3]

[1]M Tech Scholar, Dept. Of CSE , Chandigarh Engineering College,
Landran , Mohali , Punajb
renu.versha2007@gmail.com

[2]Assistant Professor, Dept. Of CSE , Chandigarh Engineering College,
Landran , Mohali ,Punjab
cecm.cse.gagandeep@gmail.com

[3]Assistant Professor, Dept. Of CSE , Chandigarh Engineering College,
Landran , Mohali ,Punjab
cecm.cse.sahil@gmail.com

Abstract- ***Cloud computing is Internet-based computing that share resources that are software and hardware resources on-demand from computers and other devices. The consumers request for available services according to their desired Quality of Service and they are charged on a pay-per-use basis. One of the most challenging problems in cloud computing is task scheduling to satisfy the Quality of service as well as minimizing the task execution time. Scheduling strategy is the key technology in cloud computing. This paper surveyed different types of scheduling algorithms and compare their various parameters. Traditional scheduling algorithms are not able to provide scheduling in the cloud environments. Therefore, new scheduling strategies are needed to overcome the problems between users and resources posed by network properties.***

Keywords: Cloud Computing, Scheduling, Algorithm.

1. INTRODUCTION

Cloud computing is an Internet-based computing that share resources that are software and hardware resources on-demand from computers and other devices. Cloud computing manages these resources and provide consumers and businesses to access advanced software applications without installation and access their personal files on any computer with internet access[8].

Cloud computing has tremendous capabilities and to make effective use of limitless capabilities, efficient scheduling algorithms are required. Scheduling strategy is the key technology in cloud computing[9]. The scheduling process in the cloud can be summarized into three stages:

S.No.	Stages of Scheduling Process	Description
1.	Resource discovery and filtering	The resources present in the network system is discovered and information related to them is collected.
2.	Resource selection	The target resource is selected based on certain parameters of task and resource.
3.	Task submission	The task to be executed is submitted to resource selected.

The goal of scheduling algorithms in distributed systems is spreading the load on the processors and maximizing their utilization while minimizing the total task execution time.

There are main two types of scheduling algorithm:
1) Static scheduling algorithm
2) Dynamic scheduling algorithm.

Both have their own advantages and limitations. However, Dynamic scheduling algorithm performs better than static scheduling algorithm but has a lot of overhead compare to it. There has been various types of scheduling algorithm exist in a distributed computing system. Traditional scheduling algorithms are not able to provide scheduling in the cloud environments. However, there is an urgent need to develop new scheduling strategies which overcome the problems between users and resources posed by network properties.

2. EXISTING SCHEDULING ALGORITHMS

There are various existing scheduling algorithms that are prevalent in clouds and out of them some are discussed as follows:

2.1 Heterogeneous Earliest Finish Time Algorithm (HEFT)

The objective of efficient scheduling is to map the tasks onto processors and execution order is set onto the processors and execution order is set so that task precedence requirements are satisfied and minimum schedule length is given. The existing algorithms for heterogeneous environments are not generally efficient to solve the DAG scheduling problem due to their high complexity and the quality of the results. Therefore, low-complexity efficient heuristic, Heterogeneous Earliest Finish Time (HEFT) Algorithm is proposed [2]. The HEFT algorithm is an application scheduling algorithm for a bounded number of heterogeneous processors. It has two major phases:

1) Prioritizing Tasks
2) Processor Selection Phase.

The algorithm first constructs a priority list of tasks and then locally optimal allocation decisions for each task are made on the basis of the task's estimated finish time.

The HEFT algorithm is an effective solution for the DAG scheduling problem in heterogeneous systems because of its robust performance, low running time, and the ability to give stable performance over a wide range of graph structures. The HEFT algorithm notably outperformed the other algorithms in terms of both performance and cost metrics including average schedule length ratio, speed up, the frequency of the best results, and the average running time.

The limitation of HEFT algorithm is that it uses techniques that are all static approaches of the mapping problem that assume static conditions for a given period of time and also in complex situations it can easily fail to find the optimal scheduling.

2.2 Resource-Aware-Scheduling Algorithm (RASA)

The efficient task scheduling algorithms are required to make effective use of the capabilities of the computational grids. Task scheduling algorithms are customarily applied by grid resource managers to optimally dispatch tasks to the grid resources. The grid manager in a computational grid tries to distribute the submitted tasks amongst the grid resources in such a way that the total response time is minimized. There are generally a substantial number of task scheduling algorithms to minimize the total completion time of the tasks in distributed systems. Two well-known examples of such algorithms are Max-min and Min-min[1]. These two algorithms estimate the execution and completion times of each of the tasks on each of the grid resources. These algorithms try to minimize the overall completion time of the tasks of finding the most suitable resources to be allocated to the tasks. By minimizing the overall completion time of the tasks, it does not necessarily result in the minimization of execution time of each individual task. To resolve the problems faced by Max-min and Min-min algorithms, a task scheduling algorithm known as Resource-Aware-Scheduling Algorithm (RASA) is proposed [4].

Resource-Aware-Scheduling Algorithm (RASA) is a scheduling algorithm that is composed of two traditional scheduling algorithms: Max-min and Min-min. RASA applies Max-min and Min-min algorithms alternatively to assign tasks to the resources. It uses the advantages of both the algorithms and overcomes their disadvantages. The Min-min algorithm executes smaller tasks, then large tasks and it is the reverse of Max-min algorithm. The problem with the Min-min algorithm is that it cannot schedule tasks when the number of small tasks is more than number of large tasks and also makespan of the system gets relatively large. The Max-min algorithm selects the task with the maximum completion time and assigns to the resource that executes the task in minimum execution time and Max-min algorithm is preferred when a number of small tasks are more than large ones and in other cases, the early execution of large tasks increases the total completion time of submitted tasks.

The experimental results obtained by applying RASA within the GridSim simulator shows that RASA outperforms the existing scheduling algorithms in large scale distributed systems. The limitation of this algorithm is that it ignores the waiting time of the large task in Min-min scheduling and only considers the resource demand not user preferences.

2.3 Scalable Heterogeneous Earliest-Finish-Time Algorithm (SHEFT)

In a cloud computing environment, even though the number of assigned resources to a workflow can be elastically scaled, there exists some scheduling problems such as the number of resources cannot be automatically determined on demand of the size of a workflow and the resources assigned to a workflow are not released until the workflow completes an execution. To solve these scheduling problems, Scalable Heterogeneous Earliest-Finish-Time Algorithm (SHEFT) is proposed[5].

In Scalable Heterogeneous Earliest-Finish-Time Algorithm (SHEFT), firstly, a model of cloud computing environment and a workflow graph representation is formalized which is followed by a formalization of the workflow scheduling problem. Then, to schedule workflows in a cloud computing environment, a SHEFT workflow scheduling algorithm is proposed. It consists of two mechanisms:

1) Listing Mechanism, it is a modified version of the Heterogeneous Earliest Finish Time (HEFT) Algorithm and
2) Machine Assignment Mechanism, which is assigned to each task so that makespan should be minimized.

The experiments show that SHEFT not only outperforms several representative workflow scheduling algorithms in enhancing workflow execution time, additionally it enables resources to scale elastically at runtime. The SHEFT algorithm does not concentrate on the multiple parameters such as scheduling success rate, cost, time, makespan, availability and reliability.

2.4 Critical-Path-on-a-Processor Algorithm (CPOP)

The scheduling of tasks on processors is the key issue for high-performance computing and the existing algorithms for heterogeneous environments are not generally efficient because of their high complexity and the quality of results. Therefore, low-complexity efficient heuristic Critical-Path-on-a-Processor Algorithm (CPOP) is proposed which schedules tasks on a bounded number of heterogeneous processors [6].

The Critical-Path-on-a-Processor Algorithm (CPOP) is a list-based scheduling algorithm for scheduling tasks on a bounded number of heterogeneous processors. This algorithm has three phases, namely, task prioritizing, task selection and processor selection phase. In the first phase, task prioritizing, the CPOP algorithm uses the upward rank and downward rank to give a priority to each task in the DAG. The upward rank is the length of the longest path from the task to exit task, including the computational cost of the task and the downward rank is the length of the longest path from a state task to the task. The CPOP algorithm uses the sum of upward rank and downward rank to assign priority of each task. In task selection phase, the task with highest priority is selected from ready task. The next phase, the production selection phase have two options: If the selected task is on the critical path, then it is scheduled on the critical-path processor (CPP); otherwise, it is assigned to the processor that minimizes its earliest finish time. The main feature of CPOP is the assignment of all the tasks that belong to the critical path to a single processor.

The observation of performance evaluation study shows that the CPOP algorithm has better scheduling quality performance and running time results than existing algorithms or at least comparable to them.

2.5 Improved Cost-Based Algorithm for Task Scheduling

In cloud computing, the scheduling of task groups where resources have different resource costs and computation performance is challenging. Since the cost of each task in cloud resources is different to one another, scheduling of user tasks in the cloud is not the same as in traditional scheduling methods, therefore an Improved Cost-Based Scheduling Algorithm is proposed for task scheduling[7].

The Improved Cost-Based Scheduling Algorithm in cloud computing is employed for making efficient mapping of tasks to available resources in the cloud. This algorithm selects a set of resources to be used for computing. It, then groups tasks according to the processing capability of the resources available. The priority levels of the tasks are calculated based on the cost of the individual use of resources by the task, the profit earned by the task and the priority level of the task. The tasks are sorted according to their priority and they are placed in three different lists based on three levels of priority: high priority, medium priority and low priority. After this, job grouping algorithm is applied to the above lists in order to allocate the task-groups of different available resources.

The results show that the processing cost spent to complete tasks after grouping the tasks is very less when compared with the processing cost spent to complete the tasks without grouping the tasks. Therefore, it concludes that improved cost-based algorithm performs better than the activity-based costing algorithm. This algorithm only takes the initial research on task scheduling in cloud platform.

Further improvement should be done to handle more complicated scenarios involving dynamic factors such as dynamically challenging cloud environment. The further improvement of this algorithm should concentrate on discussing simultaneous instead of independent task scheduling in a cloud environment.

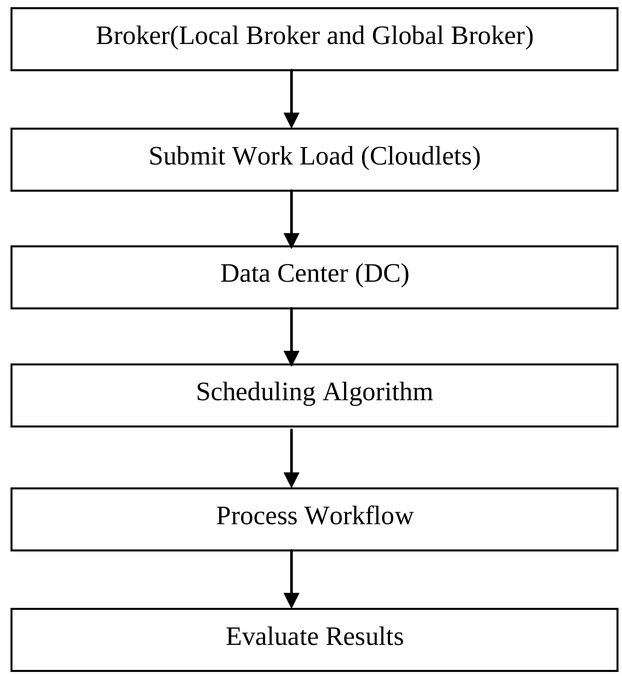

Figure 1. Flow diagram of workflow processing in cloud computing

3. COMPARISON OF VARIOUS SCHEDULING ALGORITHMS

TABLE 1:COMPARING DIFFERENT SCHEDULING ALGORITHMS BASED ON VARIOUS FACTORS

Scheduling Algorithm	Scheduling Method	Scheduling Factors	Performance Metric(s)	New Factors Included	Environment
Heterogeneous Earliest Finish Time (HEFT) Algorithm [2][3]	Dependency mode	Highest upward rank	Makespan	1) It reduces makespan of tasks in a DAG.	Grid Environment
Resource-Aware-Scheduling Algorithm (RASA) [4]	Batch mode	Grouped task	Makespan	1) It is used to reduce the makespan.	Grid Environment
Scalable Heterogeneous Earliest-Finish-Time Algorithm (SHEFT) [5]	Dependency mode	A group of tasks	Execution time, Scalability	1) It is used for optimizing workflow execution time. 2) It also enables resources to scale elastically at runtime.	Cloud Environment
Critical-Path-on-a-Processor Algorithm (CPOP)[3][6]	Dependency mode	Highest upward and downward rank	Performance, Execution time	1) It assigns all the tasks that belong to the critical path to a single processor.	Grid Environment
Improved Cost-Based Scheduling Algorithm [7]	Batch mode	Unscheduled task groups	Cost, Performance	1) It measures both cost and computation performance. 2) It also improves the computation/communication ratio.	Cloud Environment

TABLE 2: METRICS CONSIDERED BY EXISTING SCHEDULING ALGORITHM

Metrics \ Algorithm	[2]	[4]	[5]	[6]	[7]
Time	✗	✗	✔	✔	✗
Cost	✗	✗	✗	✗	✔
Scalability	✗	✗	✔	✗	✗
Scheduling Success Rate	✗	✗	✗	✗	✗
Makespan	✔	✔	✗	✗	✗
Speed	✗	✗	✗	✗	✗
Resource Utilization	✗	✗	✗	✗	✗
Reliability	✗	✗	✗	✗	✗

4. CONCLUSION

Scheduling is one of the most important tasks in a cloud computing environment. This paper surveyed the various existing scheduling algorithms in cloud computing and compare their various parameters. Existing scheduling algorithms give high throughput and cost-effective but they do not consider reliability and availability. Therefore, there is a need to implement a new scheduling algorithm that can minimize the execution time and improve availability and reliability in a cloud computing environment. The improvement can also do with building algorithms which also take user preferences while scheduling and one more aspect can help to improve the design of algorithm, which may include new factors like inter-node bandwidth etc., which have not been considered for resource matching.

References

[1] Y. Mao, X. Chen, X. Li, “Max-Min Task Scheduling Algorithm for Load Balance in Cloud Computing”, In Proc. of International Conference on Computer Science and Information Technology”, Springer India, Vol. 225, Pages(s) 457-465, 2014.

[2]L.F. Bittencourt, R. Sakellariou, E.R.M. Madeira, “DAG Scheduling Using a Lookahead Variant of the Heterogeneous Earlist Time Algorithm”, In Proc. of 18th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), Publication IEEE Conference, 2010.

[3] H. Topcuoglu, S. Hariri, Min-You Wu, “Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing”, In Proc. of IEEE Transactions on Parallel and Distributed Systems, Vol. 13, Issue 3, Page(s) 260-274, 2002.

[4] S. Parsa, Reza Entezari-Maleki, “RASA: A New Task Scheduling Algorithm in Grid Environment”, In Proc. of World Applied Sciences Journal, Vol. 7, Page(s) 152-160, 2009.

[5] Cui Lin, Shiyong Lu, “Scheduling Scientific Workflows Elastically for Cloud Computing”, In Proc. of IEEE International Conference on Cloud Computing, 2011.

[6] H. Topcuoglu, S. Hariri, Min-You Wu, “Task Scheduling Algorithm for Heterogeneous Processors”, In Proc. of Heterogeneous Computing Workshop, Publication IEEE Conference, 1999.

[7]S. Selvarani, G.S. Sadhasivam, “Improved Cost- Based Algorithm for Task-Scheduling in Cloud Computing”, In Proc. of IEEE International Conference on Computational Intelligence and Computing Research(ICCIC), 2010.

[8]C. Haffa, G. Mehta, T. Freeman,et. al. , “On the Use of Cloud Computing for Scientific Works”, 4th IEEE International Conference on eScience, 2008.

[9] Zhifeng Yu, Weisong Shi, “An Adaptive Rescheduling Strategy for Grid Workflow Applications”, IEEE , 2007.

[10]K. Navjot, S.A. Taranjit, C.S. Rajbir, “Comparison of Workflow Scheduling Algorithms in Cloud Computing”, International Journal of advanced Computer Science and Applications, Vol. 2, No. 2, 2011.