International Journal of Emerging Trend in Engineering and Basic Sciences (IJEEBS)
ISSN (Online) 2349-6967
Volume 2 , Issue 1(Jan-Feb 2015), PP210-215

Survey on Scheduling Algorithms in Cloud Computing

Miss.K.A.Deshmane[1], Miss.B.T.Pandhare[2]
[1]*SVERI's College of Engg. Padharpur, Solapur University, Maharashtra*
[2] *SVERI's College of Engg. Padharpur, Solapur University, Maharashtra*

Abstract:-Cloud computing is a rapidly growing area. There are so many users who looking forward for cloud computing and everybody wants to know about it. User applications may need large data retrieval very often and the system efficiency may degrade when these applications are scheduled taking into account only the 'execution time'. In addition to optimizing system efficiency, the cost arising from data transfers between resources as well as execution costs must also be taken into account while scheduling. This paper provides the survey on scheduling algorithms. There working with respect to the resource sharing. We systemize the scheduling problem in cloud computing,

Keywords: Cloud computing, Scheduling, Scheduling Algorithm

I. INTRODUCTION

Cloud computing has emerged as a new large-scale distributed computing paradigm that provides a dynamically scalable service delivery and consumption platform facilitated through virtualization of hardware and software with the provision of consuming various services on demand over the internet. For cloud computing based services, users consume the services when they need to, and pay only for what they use. Cloud computing delivers three kinds of services: Infrastructure as aService (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS)[1]. These services are available to user in a Pay-per- use –on demand model.

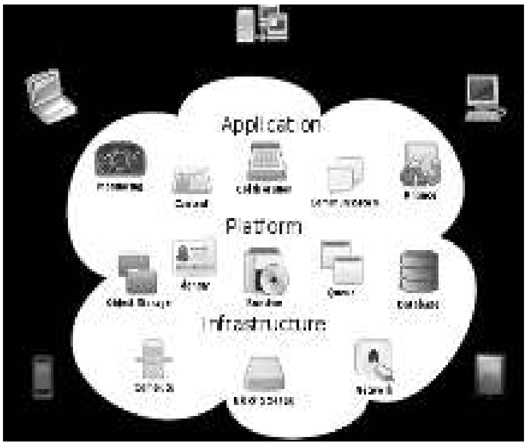

Fig. 1 Overview of Cloud Computing

To make effective use of the tremendous capabilities of the cloud, efficient scheduling algorithms are required. These scheduling algorithms are commonly applied by cloud resource manager to optimally dispatch tasks to the cloud resources. The objective of this paper is to be focus on various scheduling algorithms. The rest of the paper is organized as follows. Section 2 presents the need of scheduling in cloud. Section 3 presents various existing scheduling algorithms and section 4 concludes the paper with a summary of our contributions.

II. NEED OF SCHEDULING IN CLOUD

Scheduling of tasks is a critical issue in Cloud Computing, so a lot of researches have been done in this area. The primary benefit of moving to Clouds is application scalability. Unlike Grids, scalability of Cloud resources allows real-time provisioning of resources to meet application requirements. Cloud services like compute, storage and bandwidth resources are available at substantially lower costs. Usually tasks are scheduled by user requirements. New scheduling strategies need to be proposed to overcome the problems posed by network properties between user and resources [2].

Scheduling theory for cloud computing is gaining consideration with day by day hike in cloud popularity. In general, scheduling is the process of mapping tasks to available resources on the basis of tasks' characteristics and requirements. It is an essential aspect in efficacious working of cloud as many task parameters need to be considered for proper scheduling. The available resources should be utilized efficiently without affecting the service parameters of cloud. Scheduling process in cloud can be generalized into three stages namely–

1. Resource discovering and filtering - Datacenter Broker discovers the resources present in the network system and collects status information related to them.
2. Resource selection - Target resource is selected based on certain parameters of task and resource. This is deciding stage
3. Task submissions - Task is submitted to resource selected

III. TASK SCHEDULING TYPES

3.1 Cloud Service Scheduling

At user level scheduling deals with problems raised by service provision between providers and customers. The system level scheduling handles resource management within datacenter. Datacenter consists of many physical machines. Millions of tasks from users are received; assignment of these tasks to physical machine is done at datacenter. This assignment or scheduling significantly impacts the performance of datacenter.

3.2 Static and Dynamic Scheduling

Static scheduling allows for pre-fetching required data and pipelining different stages of task execution. Static scheduling imposes less runtime overhead. In case of dynamic scheduling information of the job components/task is not known before hand. Thus execution time of the task may not be known and the allocation of tasks is done on fly as the application executes.

3.3 Heuristic Scheduling

Optimization problems are in Class NP-hard. These problems can be solved by enumeration method, heuristic method or approximation method. In enumeration method, an optimal solution can be selected if all the possible solutions are enumerated and compared one by one. When number of instances is large, exhaustive enumeration is not feasible for scheduling problems. In that case heuristic is a suboptimal algorithm to find reasonably good solutions reasonably fast. Approximation algorithms are used to find approximate solutions to optimized solution. These algorithms are used for problems when exact polynomial time algorithms are known. Enhancing task data locality in large scale data processing systems is crucial for the job completion time.

3.4 Real Time Scheduling

The primary objectives of real time scheduling are to increase throughput and minimize average response time instead of meeting deadlines.

IV. EXISTING SCHEDULING ALGORITHMS

4.1 Enhanced Max-min Algorithm

Upendra Bhoi [3] proposed an Enhanced Max-min task scheduling algorithm that offers an improved task scheduling algorithm based on Max-min. The Max-min algorithm selects the task with the maximum completion time and assigns it to the resource on which achieve minimum execution time while Enhanced Max-min assign task with average execution time (average or Nearest greater than average Task) to resource produces Minimum completion time (Slowest Resource). There are some situations when largest task has long execution time compared to other tasks in Meta-task, in such case overall makespan is increased because this large task is executed by slowest resource first while other tasks are executed by faster resource. Therefore, instead of selecting largest task if we select Average or nearest greater than average task then overall makespan is reduced and also balance load across resources.

Algorithm:

1. for all submitted tasks in Meta-task; Ti
    1.1. For all resources; Rj
        1.1.1. Cij = Eij + rj
2. Find task Tk costs maximum execution time (Largest Task).
3. Assign task Tk to resource Rj which gives minimum completion time (Slowest resource).
4. Remove task Tk from Meta-tasks set.
5. Update rj for selected Rj.
6. Update Cij for all j.
7. While Meta-task not Empty
    7.1. Find task Tk costs maximum completion time.
    7.2. Assign task Tk to resource Rj which gives minimum execution time (Faster Resource).
    7.3. Remove Task Tk form Meta-tasks set.
    7.4. Update rj for Selected Rj.
    7.5. Update Cij for all j.

Algorithm take set of tasks need to be executed as input. Than for all submitted tasks in meta-tasks, Execution and Completion time is calculated. After that task Tk having cost average or nearest greater than average execution time is selected and assigned to resource Rj which gives minimum completion time. Then this task is removed from Meat-tasks set and scheduled time of tasks in resource Rj is updated. Simultaneously completion times of tasks in j are also updated. In next step, until the Meta-task set not get empty, Task Tk having maximum completion time is selected and assigned to resource Rj which gives minimum execution time and then removed from the Meta-task and as above explained, the scheduled time and completion time of task in j is updated. So in Enhanced Max-min, task selection scenario is changed, it is stated as "Select task with Average or nearest greater than average execution time (Average or nearest greater than average task) then assign to be executed by resource with minimum completion time (Slowest resource)".

4.2 Improved Genetic Algorithm:

As we can conclude from Genetic Algorithm that the solutions that are fit; give better generations further when we apply genetic operators on them. So, an idea is proposed for generating initial population by using Min-Min and Max-Min, which can provide better initial population than if we choose the initial population randomly. So the proposed idea can give an Improved Genetic Algorithm.

4.3 Scalable Heterogeneous Earliest-Finish-Time Algorithm (SHEFT):

Cui Lin, Shiyong Lu [4] proposed an SHEFT workflow scheduling algorithm to schedule a workflow elastically on a Cloud computing environment. The experimental results show that SHEFT not only outperforms several representative workflow scheduling algorithms in optimizing workflow execution time, but also enables resources to scale elastically at runtime.

4.4 Multiple QoS Constrained Scheduling Strategy of Multi-Workflows (MQMW):

Meng Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi [5] worked on multiple workflows and multiple QoS.They has a strategy implemented for multiple workflow management system with multiple QoS.

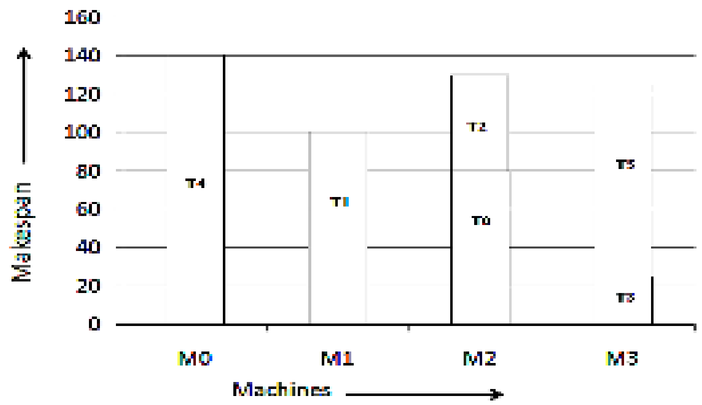

Fig.2

The scheduling access rate is increased by using this strategy. This strategy minimizes the make span and cost of workflows for cloud computing platform.

The new Improved Genetic Algorithm will be like:
1. Begin
2. Find out the solution by Min-Min and Max-Min
3. Initialize population by result of Step 2
4. Evaluate each candidate
5. Repeat until (termination condition occur)
6. Do
a. Select parents
b. Recombine pairs of parents
c. Mutate the resulting offsprings
d. Evaluate new candidate
e. Select individuals for next generation
7. End

If we apply this improved genetic algorithm on the sample data of Table 1, then the tasks are assigned as shown in Figure 4. This technique results in a makespan of 140 seconds.

4.5 A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications:

Suraj Pandey, LinlinWu, Siddeswara Mayura Guru, Rajkumar Buyya [6] presented a particle swarm optimization (PSO) based heuristic to schedule applications to cloud resources that takes into account both computation cost and data transmission cost. It is used for workflow application by varying its computation and communication costs. The experimental result shows that PSO can achieve cost savings and good distribution of workload onto resources.

4.6 Improved Cost-Based Algorithm for Task Scheduling:

Mrs.S.Selvarani, Dr.G.Sudha Sadhasivam [7] proposed an improved cost-based scheduling algorithm for making efficient mapping of tasks to available resources in cloud. The improvisation of traditional activity based costing is proposed by new task scheduling strategy for cloud environment where there may be no relation between the overhead application base and the way that different tasks cause overhead cost of resources in cloud. This scheduling algorithm divides all user tasks depending on priority of each task into three different lists. This scheduling algorithm measures both resource cost and computation performance, it also Improves the computation/communication ratio.

4.7 Resource-Aware-Scheduling algorithm (RASA):

Saeed Parsa and Reza Entezari-Maleki[8] proposed a new task scheduling algorithm RASA. It is composed of two traditional scheduling algorithms; Max-min and Min-min. RASA uses the advantages of Max-min and Min-min algorithms and covers their disadvantages. Though the deadline of each task, arriving rate of the tasks, cost of the task execution on each of the resource, cost of the communication are not considered. The experimental results show that RASA is outperforms the existing scheduling algorithms in large scale distributed systems.

RASA applies Max-min and Min-min algorithms alternatively to assign tasks to the resources. 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.

4.8 Innovative transaction intensive cost-constraint scheduling algorithm:

Yun Yang, Ke Liu, Jinjun Chen [9] proposed a scheduling algorithm which takes cost and time. The simulation has demonstrated that this algorithm can achieve lower cost than others while meeting the user designated deadline.

4.9 Scalable Heterogeneous Earliest-Finish-Time Algorithm (SHEFT):

Cui Lin, Shiyong Lu [10] proposed an SHEFT workflow scheduling algorithm to schedule a workflow elastically on a Cloud computing environment. The experimental results show that SHEFT not only outperforms several representative workflow scheduling algorithms in optimizing workflow execution time, but also enables resources to scale elastically at runtime.

V. 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] Satish Kumar and Nagesh Salimath, A survey on cloud computing and its services, *International journal of scientific Research, Volume 2 ,ISSN No.2277-8179*

[2] Sujit Tilak and Dipti Patil, A Survey of Various Scheduling Algorithms in cloud Environment, *International journal of Engineering Inventions, Vol 1, Issue 2, ISSN: 2278-7461*

[3] Upendra Bhoi and Purvi N. Ramanuj,Enchanced Max-min Task Scheduing Algorithm in cloud Computing, *International Journal of Application or Innovation in Engineering and management, Vol. 2, Issue 4, ISSN 2319-4847*

[4] Cui Lin, Shiyong Lu, " Scheduling ScientificWorkflows Elastically for Cloud Computing" in IEEE 4th International Conference on Cloud Computing, 2011.

[5] Meng Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi, "A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing", in 2009 IEEE International Symposium on Parallel and Distributed Processing.

[6] Suraj Pandey1; LinlinWu1; Siddeswara Mayura Guru; Rajkumar Buyya, A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments

[7] Mrs.S.Selvarani1, Dr.G.Sudha Sadhasivam "Improved Cost-Based Algorithm for Task Scheduling in Cloud Computing" 2010 IEEE.

[8] Saeed Parsa and Reza Entezari-Maleki," RASA: A New Task Scheduling Algorithm in Grid Environment" in World Applied Sciences Journal 7 (Special Issue of Computer & IT): 152-160, 2009.Berry M. W., Dumais S. T., O'Brien G. W. Using linear algebra for intelligent information retrieval, SIAM Review, 1995, 37, pp. 573-595.

[9] Y. Yang, K. Liu, J. Chen, X. Liu, D. Yuan and H. Jin, An Algorithm in SwinDeW-C for Scheduling Transaction-Intensive Cost-Constrained Cloud Workflows, Proc. of 4th IEEE International Conference on e-Science, 374-375, Indianapolis, USA, December 2008.

[10] Cui Lin, Shiyong Lu, "*SCPOR: An Elastic Workflow Scheduling Algorithm for Services Computing*", IEEE2011 International Conference on Service Oriented Computing & Applications, pp. 1-8, Irvine, USA, 2011
.