© 2018 IJSRSET | Volume 4 | Issue 2 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
National Conference on Advanced Research Trends in Information and Technologies (NCARTICT-2018), Department of IT, L. D. College of Engineering, Ahmedabad, Gujarat, India. In association with International Journal of Scientific Research in Science, Engineering and Technology

An Improved Rasa Algorithm in Task Scheduling

Neema Mehulkumar Desai[1], Hardik J. Joshi[2], Dr. Chirag Suryakant Thaker[3]
[1]Department of Computer Science, Gujarat University, Ahmedabad, Gujarat, India
[2]Department of Computer Science, Gujarat University, Ahmedabad, Gujarat, India
[3]Information Technology Department, L.D. Engineering Collage, Ahmedabad, Gujarat, India

ABSTRACT

Cloud computing delivers a computing environment where different resources are delivered as a service to the customer or multiple tenants over the internet. Task scheduling is an essential and most important part in a cloud computing environment. The multi-dimensional task scheduling algorithm is based on the availability of CPU, memory, and VMs. This algorithm is built based on RASA algorithm and the concept of Max-min strategy. This algorithm is developed to outperform scheduling process of RASA in case of total complete time for all submitted jobs. Proposed algorithm is based on expected execution time instead of complete time. So the scheduling tasks within cloud environment using this algorithm can achieve lower make span rather than original Max-min.

Keywords: Cloud Computing, Scheduling Algorithms, max-min algorithm, min-max algorithm, Resource Awareness Scheduling Algorithm (RASA).

I. INTRODUCTION

A Cloud is the collection of interconnected computer that are provided by one or more unified computing resources. Barrie Sosinsky defined cloud computing as "Cloud computing refers to applications and services that run on a distributed network using virtualized resources and accessed by common Internet protocols and networking standards." In this growing market of business and organization, cloud computing is the alternative for their day-by-day increasing needs. A Cloud provider first constructs a computing system called cloud in this we have several virtual machines interconnected through this the provider processes the task of the users.Cloud computing environment where multiple virtual machines (VMs) can share physical resources (CPU, memory, and bandwidth) on a single physical host and multiple VMs can share the bandwidth of a data center by using network virtualization. Because many users and applications essentially share system resources, a proper task-scheduling scheme is difficult to resource utilization and system performance. Many system parameters, such as processor power, memory space, and network bandwidth, affect the efficiency of task scheduling.

In addition, difference in computing sources in different nodes adds to the complexity of task scheduling. Furthermore, frequent data exchange among nodes, hosts, and clusters in data-intensive cloud applications makes the task-scheduling procedure extremely complicated. Most of these methods focus on allocating CPU and memory resources to various cloud-computing tasks, assuming that all physical nodes and VMs have unlimited network bandwidth. This algorithm considers the limitation of resources, and provides resources according to task needs and resource loads.

However, this algorithm did not consider the bandwidth requirements of tasks, nor did it consider the dynamic change of their resource requirements.

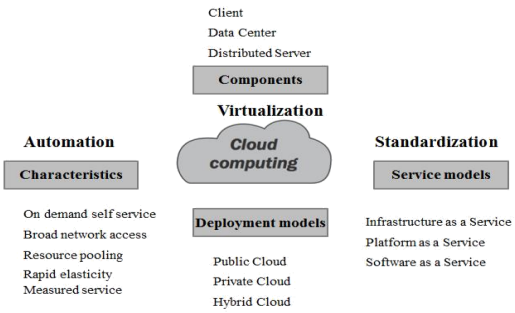

Figure 1. Fundamentals of Cloud Computing

System resources like CPU, memory and bandwidth are used by many users, so it is little difficult to construct an efficient task scheduling algorithm. The efficiency of the algorithm is affected by many things like the processor power, speed, space and memory. Generally, task scheduling is the main process in infrastructure as a service model. While scheduling the task we consider virtual machines as scheduling machines. The main aim of task scheduling algorithms in cloud environment is to maintain the correct load on processors by considering the network bandwidth and increase their usage, efficiency and to reduce their task execution time.

II. TASK SCHEDULING IN CLOUD COMPUTING

The advantage of job scheduling algorithm is to achieve a high performance computing and the best system throughput. The available resources should be utilized efficiently without affecting the service parameters of cloud. Scheduling process in cloud can be categorized into three stages they are Resource discovering and filtering, Resource selection, and Task submission [10]. In resource discovery data centre broker discovers the resources present in the network system and collects status information related to them. During resource selection process resource is selected based on certain parameters of task and resource. Then during task submission task is submitted to the selected resource.

This are the various parameters used to understand how different scheduling algorithms works,

- ✓ **Execution time:** The exact time taken to execute the given task is known as execution time. The ultimate goal of any scheduling algorithm is minimizing the execution time.
- ✓ **Completion time:** The time taken to complete the whole execution of a job. It also includes the execution time and the delay caused by the cloud system.
- ✓ **Makespan:** It is defined as the total completion time of all the tasks in a job queue. The makespan should be reduced to increase the performance of particular algorithm.
- ✓ **Response time:** The elapsed time between the end of an inquiry or demand on a computer system and the beginning of a response.
- ✓ **Resource utilization:** Resource utilization is the use of the resource in such a way that increases the throughput of the system.
- ✓ **Deadline:** It is the period of time from submitting a task to the time by which it must be completed.
- ✓ **Energy consumption:** Many different scheduling algorithms has designed to reduce the power consumption and improving the performance.
- ✓ **Performance:** Performance indicates the overall efficiency given by the scheduling algorithm in order to provide good services to the user as per their requirements.
- ✓ **Scalability:** It is the ability of the system to function well when it is changed in size to satisfy the user need.

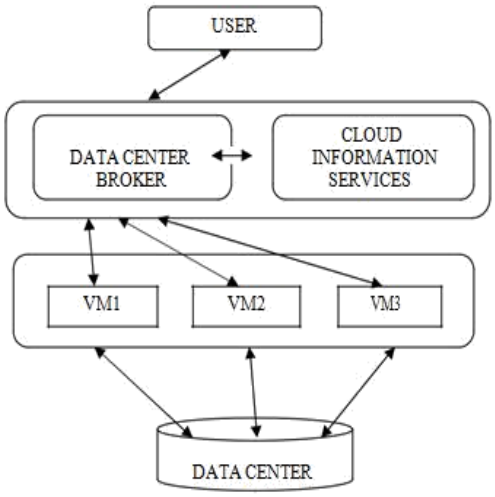

Figure 2. Scheduling in Cloud

III. EXISTING SCHEDULING ALGORITHMS

In this paper we describing various task scheduling algorithms in a nutshell.

A. FIRST COME FIRST SERVE ALGORITHM (FCFS): The First Come First Serve Algorithm (FCFS) collects the tasks in a queue until the resources are available and when the resources are available the tasks are assigned to them based on the arrival time of the task. It is the simplest algorithm and fewer complexes in nature but it does not consider any other criteria for scheduling the tasks to the machine. Here always the first task will be executed first so the tasks arriving later will have to wait for a longer period of time.

B. ROUND ROBIN ALGORITHM (RR): RR scheduling is simple, easy to implement, and starvation-free. This algorithm uses the ring as its queue to store jobs. Each job has the same execution time and it will be executed in turn. If a job can't complete its work during its turn, it will be stored back to the queue waiting for the next turn. Main feature of RR algorithm is execution of each job in turn and it doesn't have to wait for the previous one to get completed. But if the load is found to be heavy, RR will take a long time to complete all the jobs.

C. EARLIEST DEADLINE FIRST ALGORITHM: In this scheduling algorithm, the scheduler points the task having the shortest deadline. Whenever a scheduling event occurs then the queue will be searched for the process that is closest to its deadline, the found process will be scheduled for execution.

D. MIN-MIN ALGORITHM: The Min-Min Algorithm is based on the Minimum Completion Time (MCT) that is used to assign tasks to the resources having minimum expected completion time. It works in two phases, in the first phase, the expected completion time will be calculated for each task in a list and during the second phase, the task with the overall minimum expected completion time from the list is chosen and assigned to the corresponding machine. Then the task is removed from the list and the process is repeated until all the tasks in the list are mapped to the corresponding resources.

E. RESOURCE AWARENESS SCHEDULING ALGORITHM (RASA): The Resource Awareness Scheduling Algorithm (RASA) is the hybrid of Min-Min and Max-Min algorithm. In this algorithm the Min-Min and Max-Min algorithms are applied alternatively to take advantage of both the algorithm and overcome their drawbacks.

F. IMPROVED MAX-MIN ALGORITHM: This algorithm assigns task with maximum execution time to the resource which produce minimum completion time rather than Max-Min algorithm which assigns the task with maximum completion time to the resource which provides minimum execution time.

G. ENHANCED MAX-MIN ALGORITHM: This algorithm modifies the Max-min algorithm. It is based on the expected execution time in which it assigns a task with average execution time on the machine which gives minimum completion time.

IV. PROPOSED SCHEDULING ALGORITHM

Our proposed scheduling algorithm, Improved RASA algorithm in Tsk Scheduling, is presented in Fig 3. The algorithm represents the completion time of the task *Ti* on the resource *Rj*. If the number of available resources is even, the Min-min strategy is applied to assign the first task, otherwise the Max-min strategy is applied. For instance, if the first task is assigned to a resource by the Min-min strategy, the next task will be assigned by the Max-min strategy. In the next round the task assignment begins with a strategy different from the last round. For instance if the first round begins with the Max-min strategy, the second round will begin with the Min-min strategy. Experimental results show that if the number of available resources is odd it is preferred to apply the Min-min strategy the first in the first round otherwise is better to apply the max-min strategy the first.

Alternative exchange of the Min-min and Max-min strategies results in consecutive execution of a small and a large task on different resources and hereby, the waiting time of the small tasks in Max-min algorithm and the waiting time of the large tasks in Min-min algorithm are ignored. As RASA is consist of the Max-Min and Min-Min algorithms and have no time consuming instruction, the time complexity of RASA is

$O(mn^2)$ where m is the number of resources and n is the number of tasks (similar to Max-min and Min-min algorithms).

```
1. for all tasks T_i in meta task M_v
2.         for all resources R_j
3.                   C_ij = E_ij + r_j
4. do until all tasks in M_v are mapped
5.         if the number of resources is odd then
6.                   for each tasks in M_v find the earliest
                        complete time and the resources that
                        obtains it
7.                   find the task T_k with the maximum earliest
                        completion time
8.                   assign task T_k to the resources R_x that gives
                        the earliest completion time
9.                   delete tasl T_k from M_v
10                   update R_x
11.                  update C_ij for all I
12.        else
13.                  for each task in M_v find the earliest
                     completion time and the resources that
                     obtains it
14.                  find the task T_k with the minimum earliest
                     completion time
15.                  assign task T_k to the resources R_x that
                     gives the earliest completion time
16.                  delete task T_k from M_v
17.                  update r_x
18.                  update c_ij for all i
19.        end if
20.end do
```

Figure 3. Proposed Scheduling Algorithm

Suppose that m resources Rj(j = 1,...,m) have to process n tasks Ti(i = 1,..., n). A schedule for each task is an allocation of one or more time intervals to one or more resources [16]. The expected execution time Eij of task Ti on resource Rj is defined as the amount of time taken by Rj to execute Ti given Rj has no load when Ti is assigned. The expected completion time Cij of task Ti on resource Rj is defined as the wall-clock time at which Rj completes Ti (after having finished any previously assigned tasks). Let bi denote to the beginning of the execution of task Ti. From the above definitions, Cij=bi+Eij. Let Ci be the completion time for task Ti and it is equal to Cij where resource Rj is assigned to execute task Ti. The makespan for the complete schedule is then defined as $Max_{Ti \in K}$ (C_i). Makespan is a measure of the throughput of the heterogeneous computing system (like computational grid)[9,11].

V. PROPOSED ARCHITECTURE

There are two options to run proposed algorithm, one is VMware and another is cloudSim. By using this both services we can easily access the all features and its advantage in our system and also environment too run our algorithm. CloudSim is framework of modeling and simulation of cloud computing infrastructure and services. It's been written in JAVA.

VMware is a platform for virtualization software and services. It can be run in Windows, Linux and MacOS. VMware, a global leader in cloud infrastructure and digital workspace technology, accelerates digital transformation by enabling unprecedented freedom and flexibility in how our customers build and evolve IT environments. With VMware solutions, organizations are improving business agility by modernizing data centers and integrating public clouds, driving innovation with modern apps, creating exceptional experiences by empowering the digital workspace, and safeguarding customer trust by transforming security [13].

CloudSim, A suitable alternative is the utilization of simulations tools, which open the possibility of evaluating the hypothesis prior to software development in an environment where one can reproduce tests. Specifically in the case of Cloud computing, where access to the infrastructure incurs payments in real currency, simulation-based approaches offer significant benefits, as it allows Cloud customers to test their services in repeatable and controllable environment free of cost, and to tune the performance bottlenecks before deploying on real Clouds. At the provider side, simulation environments allow evaluation of different kinds of resource leasing scenarios under varying load and pricing distributions [14].

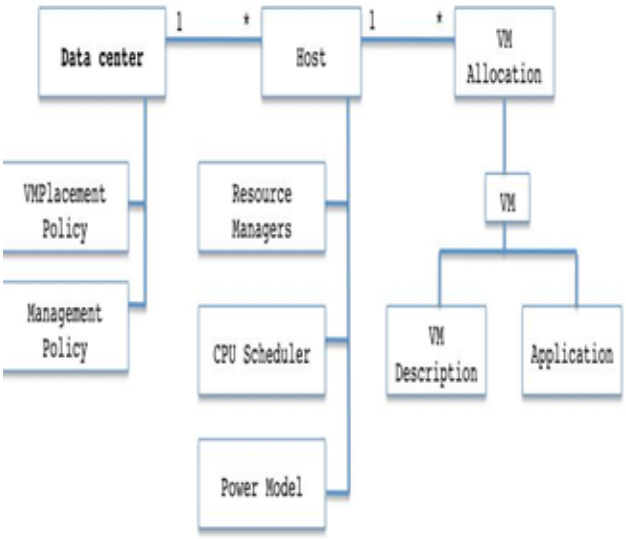

Figure 4. CloudSim Architecture

VI. OPEN ISSUES

Based on the survey on various task scheduling algorithms we came to know that there is still lots of improvements that can be carried out. The major issues in task scheduling algorithms are response time, cost, resource allocation, deadline, energy consumption and many other. Some techniques can be adopted to

improvise the different issues and increase the performance of the system.

VII. CONCLUSION AND FUTURE SCOPE

Min-min and Max-min algorithms are applicable in small distributed systems. To achieve this, in this paper, a new task scheduling algorithm, Improved RASA algorithm in Tsk Scheduling, is proposed. Improved RASA algorithm in Tsk Scheduling is composed of two traditional scheduling algorithms; Max-min and Min-min. Improved RASA algorithm in Tsk Scheduling uses the advantages of Max-min and Min-min algorithms and covers their disadvantages. In this paper, the deadline of individual task, arriving rate of the different tasks, cost of the task execution on individual of the resource, cost of the communication and many other cases that can be a topic of research are not considered.

VIII. REFERENCES

[1]. Peter Mell, Timothy Grance, "The NIST definition of Cloud Computing (September, 2011)", Accessed on May, 2014.

[2]. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A.Rabkin, I. Stoica, and M. Zaharia. "A view of cloud computing", Communications of the ACM,53(4):50-58, 2010.

[3]. Etminani. K, and Naghibzadeh. M, "A Min-min Max-min Selective Algorithm for Grid Task Scheduling". The Third IEEE/IFIP International Conference on Internet, Uzbekistan, 2007.

[4]. Mrs.S.Selvarani1; Dr.G.Sudha Sadhasivam, improved cost-based algorithm for task scheduling in Cloud computing, IEEE 2010.

[5]. 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.

[6]. S. Devipriya, C. Ramesh, "Improved Max-Min Heuristic Model for Task Scheduling in Cloud". International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), 2013

[7]. El-Sayed T. El-kenawy, Ali Ibraheem El-Desoky, Mohamed F. Al-rahamawy "Extended Max-Min Scheduling Using Petri Net and Load Balancing" International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-4, September 2012

[8]. Maheswaran, M., Sh. Ali, H. Jay Siegel, D. Hensgen and R.F. Freund, 1999. Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems. Journal of Parallel and Distributed Computing, 59: 107-131.

[9]. Shu-Ching Wang, Kuo-Qin Yan, Shun-Sheng Wang, Ching-Wei Chen, "A Three-Phases Scheduling in a Hierarchical Cloud Computing Network", IEEE, 2011-888

[10]. Braun, T.D., H. Jay Siegel, N. Beck, L.L. Boloni, M. Maheswaran, A.I. Reuther, J.P. Robertson, M.D. Theys and B. Yao, 2001. A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems. Journal of Parallel and Distributed Computing, 61: 810-837.

[11]. Wikipedia. Cloud computing. Retrieved from http://en. wikipedia.org/wiki/Cloud-computing,2011

[12]. https://www.vmware.com/in/company.html VMware Toolkit

[13]. http://www.cloudbus.org/cloudsim CloudSim

[14]. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future generation computer systems, vol. 25, no. 6, pp. 599-616, June 2009.