www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 5 May, 2013 Page No. 1636-1640

A NEW APPROACH FOR LOAD BALANCING IN CLOUD COMPUTING

S. Mohana Priya, B. Subramani
M. Phil Research Scholar
Department of Computer Science Dr. NGP Arts and Science College Coimbatore, India
mohana.priya633@gmail.com

Head, department of Information Technology Dr. NGP Arts and Science College Coimbatore, India
subramaningp@gmail.com

Abstract

Cloud computing is built on the base of distributed computing, grid computing and virtualization. The eminence of the place of cloud computing in future converged networks is incontestable. It is due to the obvious advantages of the cloud as a medium of storage with ubiquity of access platforms and minimal hardware requirements on the user end. The scalability, high availability, load balancing, cost, performance are some major issues. Cloud computing environments provide scalability for applications by providing virtualized resources dynamically. Moving applications to a cloud computing environment triggers the need of scheduling as it enables the utilization of various cloud services to facilitate execution. In the proposed algorithm it uses the active monitoring load balancing algorithm and resource aware scheduling algorithm for improved resource utilization and scheduled load balancing for high performance in cloud systems.

Key words: Computing, Virtualization, Scheduling, Loadbalancing, Cloudsim

I. INTRODUCTION

The flexibility of cloud computing is a function of the allocation of resources on demand. It provides secure, quick, convenient data storage and computing power with the help of internet. Virtualization, distribution and dynamic extendibility are the basic characteristics of cloud computing [1]. Nowadays most software and hardware have provided support to virtualization. Many virtualized factors such as IT resource, hardware, software, operating system and net storage, can be managed in the cloud computing platform; every environment has nothing to do with the physical platform. 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. There are relatively a large number of scheduling algorithms to minimize the total completion time of the tasks in distributed systems [2]. This type of scheduling algorithms try to minimize the overall completion time of the tasks by finding the most suitable resources to be allocated to the tasks. It should be noticed that minimizing the overall completion time of the tasks does not necessarily result in the minimization of execution time of each individual task. The proposed system focus on various scheduling algorithms on load balancing for high performance. This paper uses the active monitoring scheduling algorithm and resource aware scheduling algorithm, which gives the efficient resource allocation for each incoming request and to minimize the overall completion time of tasks. The resource aware scheduling algorithm calculates execution time of each task and it is assigned to resources to improve performance. The remaining part of the paper consists Sections the need for scheduling in cloud, presents various scheduling algorithms and concludes the paper with a summary of our contributions.

II. NEED FOR SCHEDULING IN CLOUD

The primary benefit of moving to Clouds is application scalability. When compared to Grids, scalability of Cloud resources allows real-time provisioning of resources to meet application requirements. The cloud services like storage and bandwidth resources are available at substantially lower costs. The tasks are scheduled usually according to user requirements. New scheduling strategies need to be proposed to overcome the problems posed by network properties between user and resources. New scheduling strategies may use some of the conventional scheduling concepts to merge them together with some network aware strategies to provide solutions for better and more efficient job scheduling [1]. Actually tasks are scheduled by user requirements. Initially, scheduling algorithms were being implemented in grids [2] [3]. Due to more usage of clouds for various services, now there is a need to implement scheduling algorithms in cloud. The major benefit of moving to Clouds is application scalability. When compared to Grids, scalability of Cloud resources allows real-time provisioning of resources to meet application requirements. It enables workflow management systems to readily meet Quality of- Service (QoS) requirements of applications [4], as opposed to the traditional approach that required advance reservation of resources in global multi-user Grid environments. Cloud services like application access, compute, storage and bandwidth resources are available at substantially lower costs. In cloud application services often require very complex execution environments. It is very difficult to create such environments on grid resources [2]. Virtual machines allow the application developer to create a fully customized, convenient execution environment configured specifically for their application. Traditional way for scheduling in cloud computing tended to use the direct tasks of users as the overhead application base. The difficulty is that there may be no relationship between the overhead application base and the way that different tasks cause overhead costs of resources in cloud systems [1]. For large number of simple tasks this increases the cost and the cost is decreased if we have small number of complex tasks.

In 2011, Laiping Zhao, Yizhi Ren & Kouichi Sakurai proposed a DRR (Deadline, Reliability, Resource-aware) scheduling algorithm, which schedules the tasks such that all the jobs can be completed before the deadline, ensuring the Reliability and minimization of resources [8].In 2011, S. Sindhu & Saswati Mukherjee proposed two algorithms for cloud computing environment and compared it with default policy of cloudsim toolkit while considering computational complexity of jobs. This paper provided us a framework for our investigation [9]. The Min-min algorithm begins with the set *U* of all unscheduled tasks. After that, the set of minimum completion times for each of the tasks exiting in *U* is found. Then the task with the overall minimum completion time from unscheduled tasks is selected and assigned to the corresponding resource (hence the name Min-min).Last, the newly scheduled task is removed from *U* and the process repeats until all tasks are scheduled. Both the Min-min and Max-min algorithms consider a hypothetical assignment of tasks to resources, projecting when a resource will become idle based on the hypothetical assignment. Both algorithms have time complexities of *O(mn2)*, where *m* is the number of resources in the system and *n* is the number of tasks which should be scheduled to be executed [5].

III. VARIOUS SCHEDULING ALGORITHMS:

Task Scheduling and Load-balancing method:

A task is an activity that uses a set of inputs to produce a set of outputs. Fixed set processes are statically assigned to processors either at compile-time or at start-up (i.e. partitioning). Load - balancing algorithms are used to avoid the overhead of loads. In grid computing algorithms can be broadly categorized as centralized or decentralized, static or dynamic [7], or the hybrid policies in latest trend. Larger systems are supported by Centralized load balancing approach. Hadoop system takes the centralized scheduler architecture. All information is known in advance in static load balancing algorithm and tasks are allocated according to the prior knowledge and will not be affected by the state of the system. Tasks to the processors are allocated dynamically as they arrive in dynamic load-balancing mechanism. Redistribution of tasks has to take place when some processors become overloaded [6].

In cloud computing, each application of users will run on a virtual operation system the cloud systems distributed resources among these virtual operation systems. Each application is completely different and is independent and has no link between each other whatever, for example, some require more CPU time to compute compound task, and some others may need more memory to store data, etc. Resources are sacrifice on activities performed on each individual unit of service. The direct costs of applications, every individual use of resources (CPU cost, memory cost, I/O cost, etc.) must be measured. The direct data of each individual resources cost has been measured, profit analysis and more accurate cost [8].

Min-min algorithm: Establishes the minimum completion time for every unscheduled job (in the same way as MCT), and then assigns the job with the minimum completion time (hence Min-min) to the processor which offers it this time[10].

Max-min algorithm: Is very similar to Min-min. Again the minimum completion time for each job is established, but the job with the maximum minimum completion time is assigned to the corresponding processor[10].

IV. PROPOSED LOAD BALANCING ALGORITHM

The Proposed Load balancing algorithm is divided into three parts. The first phase is the initialization phase. In the first phase, the expected response time of each Virtual Node is to be found. In second Phase, efficient Virtual Node is found. Last Phase return the ID of efficient Virtual Node.

Efficient algorithms find expected response time of each Virtual machine (expected response time find with the help of resource info program). When a request from the Data Center Controller to allocate a new Virtual node arrives, Algorithm finds the most efficient Virtual Node i.e., node having least loaded, minimum expected response time for allocation. Efficient algorithms return the id of the efficient node to the Datacenter Controller. New allocation is notified by Datacenter Controller. Proposed algorithm updates the allocation table increasing the allocation count for that Virtual node. Once when the Virtual node finishes processing the request and the Datacenter Controller receives the Response. Data center controller notifies the efficient algorithm for the Virtual Node de-allocation. Continue From Step2.

The purposed algorithm finds the expected Response Time of each Virtual Nodes because virtual machines are of heterogeneous platform, the expected response time can be find with the help of the following formulas

Response Time = Fin - arrt + TDelay (1)

Where, arrt is the arrival time of user request and Fin is the finish time of user request and the transmission delay can be determined using the following formulas

TDelay = Tlatency + Ttransfer (2)

Where, TDelay is the transmission delay T latency is the network latency and Ttransfer is the time taken to transfer the size of data of a single request (D) from source location to destination.

Ttransfer = D / Bwperuser (3)

Bwperuser = Bwtotal / Nr (4)

Where, Bwtotal is the total available bandwidth and Nr is the number of user requests currently in transmission. Internet Characteristics also keeps track of the number of end-user requests in-flight between two regions for the value of Nr.

RASA: Our proposed scheduling algorithm, RASA, is presented in fig 1. The algorithm builds a matrix *C* where *Cij* represents the completion time of the task *Ti* on the resource *Rj*. If the number of available resources is odd, the Min-min strategy is applied to assign the first task, otherwise the Max-min strategy is applied. The remaining tasks are assigned to their appropriate resources by one of the two strategies, alternatively. 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.

```
for all tasks Ti in meta-task Mv
    for all resources Rj
        Cij=Ej+rj
        do until all tasks in Mv are mapped
        if the number of resources is even then
            for each task in Mv find the earliest completion
            time and the resource that obtines it
            find the task Tk with the maximum earliest
            completion time
            assigne task Tk to the resource Rl that gives
            the earliest completion time
            delete task Tk from Mv
            update rl
            update Cil for all i
        else
           for each task in Mv find the earliest completion
           time and the resource that obtines it
           find the task Tk with the minimum earliest
           completion time
           assigne task Tk to the resource Rl that gives the
           earliest completion time
           delete task Tk from Mv
           update rl
           update Cil for all i
       end if
    end do
```

Fig 1: Algorithm (RASA)

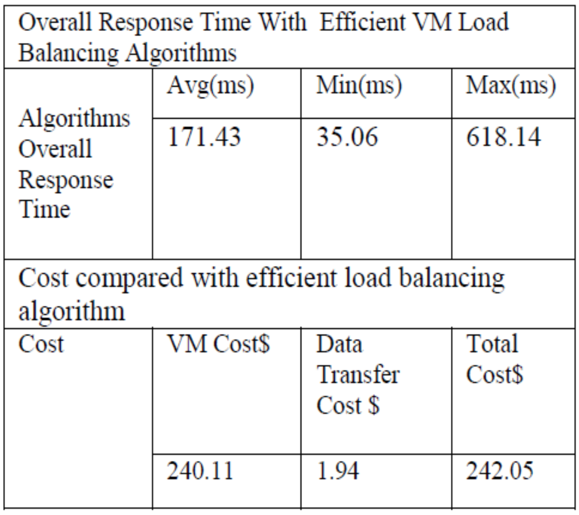
RASA is the combination of the Max-Min and Min-Min algorithms and have no time consuming instruction, the time complexity of RASA is given as *O(mn2)* where *m* is the number of resources and *n* is the number of tasks (similar to Max-min and Min-min algorithms).

The proposed algorithm gives the efficient resource allocation for tasks and minimised execution time of each task which gives the minimised completion time, it is the combination of both active monitoring algorithm and resource aware scheduling algorithm.

Which results in the better performance and resource utilization in cloud computing.

V. RESULT

The proposed algorithm implemented through simulation packages like CloudSim and tool related to CloudSim. Java language is used for implementing virtual machine load balancing algorithm. Assuming the application is deployed in one data center having 50 virtual machines (with 1024Mb of memory in each virtual machine running on physical processors capable of speeds of 100 MIPS) and Parameter values are as in table represents the result details of an efficient experimental load balancing in the virtual nodes.



Overall Response Time With Efficient VM Load Balancing Algorithms			
Algorithms Overall Response Time	Avg(ms)	Min(ms)	Max(ms)
	171.43	35.06	618.14
Cost compared with efficient load balancing algorithm			
Cost	VM Cost$	Data Transfer Cost $	Total Cost$
	240.11	1.94	242.05

In order to illustrate our algorithm, assume we have T1, T2, T3 and T4 as tasks which are in meta-tasks and scheduling manager has two resources R1 and R2as problem set. Figure 2 RASA algorithm achieves total makespan 9 seconds, choose alternatively between large tasks and small tasks respectively because of number of resources is even [2] and uses just only one resource. The result of illustration gives that the algorithm provides the efficient execution of task on resources, which are found by active load balancing algorithm.

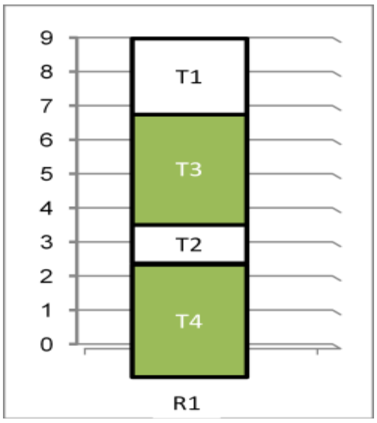


Fig 2: Gantt chart of improved rasa algorithm

VI. CONCLUSION

The main reason for possible success of cloud computing and vast interest from organizations throughout the world is due to the broad category of services provided with cloud. As there are many services provided by cloud there is a decrease in performance due to over load of requests to the server. In this paper a new load balancing algorithm for the Virtual machines and a task scheduling algorithm were proposed. The experiment result is that if an efficient virtual machine is selected for process and minimum execution time of task, it increases the performance and decreases the average response time and cost in cloud network. The future work of this paper is to implement and improve the cost efficiency parallel to the increased performance.

REFERENCES

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

[2] 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.

[3]Nithiapidary Muthuvelu, Junyang Liu, Nay Lin Soe, Srikumar Venugopal, Anthony Sulistio and Rajkumar Buyya. "A Dynamic Job Grouping-Based Scheduling for Deploying Applications with Fine-Grained Tasks on Global Grids",in Australasian Workshop on Grid Computing and e-Research (AusGrid2005), Newcastle, Australia. Conferences in Research and Practice in Information Technology, Vol. 44.

[4]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.

[5]Freund, R.F., M. Gherrity, S. Ambrosius, M. Campbell, M. Halderman, D. Hensgen, E. Keith, T. Kidd, M. Kussow, J.D. Lima, F. Mirabile, L. Moore, B. Rust and H.J. Siegel, 1998. Scheduling Resource in Multi-User, Heterogeneous, Computing Environment with SmartNet. In the Proceeding of the Seventh Heterogeneous Computing Workshop.

[6] M. Zaharia, D. Borthakur, J. S. Sarma, K. Elmeleegy, S. Shenker and I. Stoica, "Job scheduling for multi-user mapreduce clusters," EECS Department, University of California, Berkeley, Tech. Rep. USB/EECS-2009-55, Apr 2009.

[7]H.J. Braun, T. D.and Siegel, N. Beck, L.L. Blni, M. Maheswaran, A.I.Reuther, J.P. Robertson, M.D. Theys, and B. Yao, "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems," Journal of Parallel and Distributed Computing, vol. 61, no. 6, 2001, pp.810–837.

[8]Zhao, L., Ren, Y., Sakurai, K.: ―A Resource Minimizing Scheduling Algorithm with Ensuring the Deadline and Reliability in Heterogeneous Systems‖. In: International Conference on Advance Information Networking and Applications, AINA.(IEEE 2011)

[9]Sindhu, S., Mukherjee S.: ―Efficient Task Scheduling Algorithms for Cloud Computing Environment‖. In: International Conference on High Performance Architecture and Grid Computing (HPAGC-2011), vol 169, pp 79-83 (2011)

[10]D.Doreen Hephzibah Miriam and K.S.Easwarakumar A Double Min Min Algorithm for Task Metascheduler on Hypercubic P2P Grid Systems, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 4, No 5, July 2010.