GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

LOAD BALANCING TECHNIQUES IN CLOUD COMPUTING

Ms. Heer Dholakia[*1] and Prof. Vaseem Ghada[2]
[*1]P.G Student, Computer Engineering, B.H.Gardi Collage of Engg. & Tech., India
[2]Assistant Professor, Computer Engineering, B.H.Gardi Collage of Engg. & Tech., India

ABSTRACT

Cloud computing provides dynamic virtual computing resources such as storage, processing power etc, base on pay per use. To satisfy increasing request day by day from users of cloud, efficient load balancing is required. It is responsible to balance load Between nodes in a cloud system for better resource utilization and improve system performance. This paper will discuss some of the existing load balancing technique and benefits of it.

Keywords: *Cloud computing ,Load Balancing, Max-Min*

I. INTRODUCTION

In Distributed computing environment, types of computing are Clusters, Grid and cloud computing which is recently blooming in organization for more proficient and efficient way to access data. Clusters are distributed and parallel in which single administrative domain supervise the system. Grid is aggregation of resources such as server, storage, network, etc. and provisioned as needed.

Cloud computing is of two terms, cloud and computing. Cloud contains huge amount of heterogeneous resources which are utilize by users and Computing is performed base on SLA (Service Level Agreement) where user expect maximum resource utilization and minimum cost. Cloud computing contains interconnection between Servers, Data centers, Hosts, Virtual Machines etc. Cloud computing provides self-administration and facility to transmit and storage of files or multimedia content for users.

Types of Cloud are Public, Private, Community and Hybrid. In public clouds resources are offered as service publically, Users need to pay for time duration they use that services (pay per use). Private clouds are secured than public where organization operates it within internal enterprise datacenter. Cloud uses where many organizations jointly construct and shared cloud infrastructure. Hybrid clouds are combination of private and public, It enables the organization to serve its need in private cloud and its occasional needs can be public. [1]

There are some basic characteristics of cloud such as, on demand self-service, broad network access, rapid elasticity, resource pooling, measured service, virtualization, service orientation, geographic distribution, resilient computing. Basic Service provided by Cloud are IaaS – Infrastructure as a Service, PaaS – Platform as a Service and SaaS – Software as a Service. Iaas provide conceptual infrastructure over internet, example is Amazon EC2.In Paas, client can create software using tools and libraries provided by service provider, example is Google app engine. In Saas, developers can use business specific capabilities developed by other, example is google docs.

Load balancing distributes workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units or disk drives. Load balancing is essential for efficient operations in distributed environments. It means distributing the amount of work to do between different servers in order to get more work done in the same amount of time and serve clients faster. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource.[2]

Static approach: - Static load balancing algorithm equally divides the traffic between all the users. The only information that it uses is the average behavior of the system. This approach ignores the current state or the load of the node in the system.

Dynamic approach:-This approach takes into consideration only the current state of the system during load balancing decision. It is more suitable for widely distributed system such a cloud computing. Also this approach overcomes the difficulties of static approach.

Metrics for load balancing[2]
1. Throughput: It is used to the all task whose execution has been completed. The performance of any system is improved if throughput is high

2. Fault tolerance: It means recovery from failure. The load balancing should be a good fault tolerance technique.

3. Migration Time: It is the time to migrate the jobs or resources from one node to other nodes. It should be minimized in order to enhance the performance of the system.

4. Response Time: It is the amount of time that is taken by a particular load balancing algorithm to response a task in the system. This parameter should be minimized for better performance of a system.

5. Scalability: It is the ability of an algorithm to perform load balancing for any finite number of nodes of a system. This metric should be improved for a good system.

II. LITERATURE REVIEW

1) Round Robin Algorithm (RR)[3]
Round Robin algorithm works base on time quantum, where service provider schedules resources to customer for that specific time interval. Here, If time quantum is too large for each user than Round Robin algorithm is similar to FCFS algorithm and If time quantum is too small then Round Robin can be consider as Processor Sharing Algorithm, where number of context switches are High. In static Quantum, Size of quantum need to be decide and it may lead to longer average waiting time, Higher turnaround time and low throughput. Dynamic time quantum may overcome few issues of static quantum.

2) Equally Spread Current Execution(ESCE)[4]
In this algorithm, Load balancer try to balance equal load to each virtual machine in data centre. For that it maintains index table of virtual machine and number of requests assigned to each virtual machine. If any new request arrives, Load balancer scans index table and assign least loaded virtual machine. In case more than one least loaded nodes are available than first identified will assigned to requested client.

3) Shortest Job Scheduling
In this algorithm job which have shortest execution time will select first for execution. Shortest job first had less waiting time for processes which make this approach more powerful.

4) First Come First Serve
It is the simplest parallel task ordering dynamic load balancing algorithm. The implementation of FCFS policy is easily managed with FIFO queue. The data centre controller searches for virtual machine and assign user's request to VM.

5) Throttled Load Balancing Algorithm[4]
Here Load Balancer maintains an index table of virtual machine with it's states, Available or Busy. When user request for computation, load balancer finds first available VM that is suitable to perform requested job and returns VM id to data centre for further communication between VM and data centre. If VM is not found until full scan of index table than load balancer returns -1 to data centre.

6) Max-Min[6]
In this algorithm calculation of completion of all jobs will be done initially than Jobs that have maximum completion time to complete task will be assign to nodes that have minimum completion time to compute jobs.

7) Min-Min[6]
Min-Min algorithm begins with, set of all pending jobs are calculated with time taken to complete task. Then nodes with Minimum completion time for jobs are selected and that nodes will mapped with set of jobs. This process will repeat till set of task are completed.

8) Join Idle Queue[5]
It balances load in large scale by assigning idle processors to the dispatcher. Dispatcher allocate job to processors to reduce the average length of jobs at each processor. This algorithm reduce system load and does not affect response time.

9) Honey Bee [6]
Honey bee foraging algorithm is inspired from the behavior of honey bees. Finder honey bees goes out for the search of the food source and when they find the food they come and perform a special dance known as waggle dance to tell the quality and quantity of the food to the reapers. The dance also tells the distance from the beehive to the food. So, inspired from honey bee foraging algorithm, In load balancing the dance is advert board. This board is also used to advertise the overall colony profit. Servers are grouped under the virtual server with their own virtual queue. Each /server processing the demand from its queue first calculates the profit which is analogous to the quality that bees show in waggle dance. In load balancing this profit or the waggle dance is equal to the amount of time required to fulfil the request.

10) Ant Colony Optimization [6]
This algorithm is developed on the inspiration of behaviour of ant selecting its path in search of its food. Ant algorithm is a multi agent approach to combinatorial optimization problems like travelling sales man problem and quadratic assignment problem. Ant's behaviour is directed more towards the survival of the colonies but for individuals.Ants select its path based on the pheromones trails laid by its predecessors. This is similar to the shortest path selection mechanism used in networks. In this algorithm, pheromone table is used select the next node. Each row in the pheromone table represents the routing preference for each destination, and each column represents the probability of choosing a neighbour as the next hop. Ants are launched from a node with a random destination. This algorithm overcomes heterogeneity, provides high level of scalability, highly adaptable to dynamic environments and is highly fault tolerant.

Table 1. Comparison table of existing load balancing technique

Sr.No	ALGORITHM	MERIT	DEMERIT
1	Round Robin[3]	Fixed time slice.; It is easy to understand; Fairness Performs better for short CPU burst. Also used priority (running time and arrival time).	Larger tasks take long time. Can occur more context switches due to short quantum time. Job should be same to achieve high performance.
2	Equally Spread Current Execution(ESCE)[4]	Response time and processing time of a job is improved.	Not fault tolerant because of single point of failure.
3	Shortest Job Scheduling[5]	Good Resource Utilization.	Less Performance and throughput.
4	First Come First Serve[5]	Easy to implement	No priorities will entertain.
5	Throttled load Balancing[4]	Good performance; List is used to manage the tasks.	Tasks need to be waited.
6	Max- Min[6]	Requirements are prior known. So works better.	It takes long time to complete the task.

7	Min- Min[6]	Smallest completion time value. In presence of more small tasks, it shows best result.	Starvation Machine and tasks variation can not be predicted.
8	Join Idle Queue[5]	Achieves good performance and low makespan.	Response time is too low.
9	Honey Bee[7]	Increases throughput; Minimize response time.	High priority tasks can’t work without VM machine.
10	Ant – Colony[7]	Faster information can be collected by the ants.; Minimizes make span.; Independent tasks; Computationally intensive	Network is over headed so search takes long time. No clarity about the number of ants.

III. SIMULATION TOOL STUDY

CloudSim is basically used for industry and research development. Which provides a framework for experiments, according to specific design issues that can be seamlessly set to offered generalized framework.Conceptually, CloudSim on one side offers classes representing data centers, physical hosts, virtual machines, services to be executed in the data centers, users of cloud services, internal data center networks, and energy consumption of physical hosts and data centers elements. On the other side, CloudSim supports dynamic insertion of simulation elements and provides message-passing application and data centre network topology. In addition, CloudSim supports the modeling and simulating of different cloud computing environments, as well as simulation of many data centres.[8]

CloudAnalyst provides new powerful features such as an easy-to-use GUI, the ability to separate a simulation from the programming code, quick simulation setup, and an enhanced graphical results display, featuring useful formats such as tables and charts. Installation of CloudAnalyst is very easy and time efficient. When the simulation completes, the output panel displays the response time for each user base. The simulator provides a detailed screen with results that include response time for each user, request servicing time by each data center, and a number of all requests serviced by the data center.[8]

In CloudExp, the users need to configure the workplace properties and then decide the number of users, number of cloudlet in that region, the cloudlet type, and other properties of cloudlet; these are shown in When the simulation completes, CloudExp results are saved as Excel sheets to simplify the analysis of results. CloudExp used to simulate different types of configurations with different objectives. CloudExp is extended to support the multi-agent approach for resource management. An evaluation study for map-reduce task scheduling algorithms over cloud computing infrastructure is presented in .The CloudExp tool is extend to implement a number of map-reduce tasks over a cloud-based infrastructure. Evaluating a large-scale health system is presented at which aims at supporting global health awareness. CloudExp is used to evaluate large-scale cloudlet deployment in. An integrated framework for large-scale mobile cloud computing environment is presented in. [8]

IV. PROPOSED APPROACH

In Max-Min algorithm as new task arrive to data center it will calculate execution time of task and among multiple request, priority has been given as, task with maximum execution time will be assigned to node which gives minimum completion time.

ISSN 2348 – 8034
Impact Factor- 4.022

V. RESULT ANALYSIS

Cloud Analyst Simulation Results

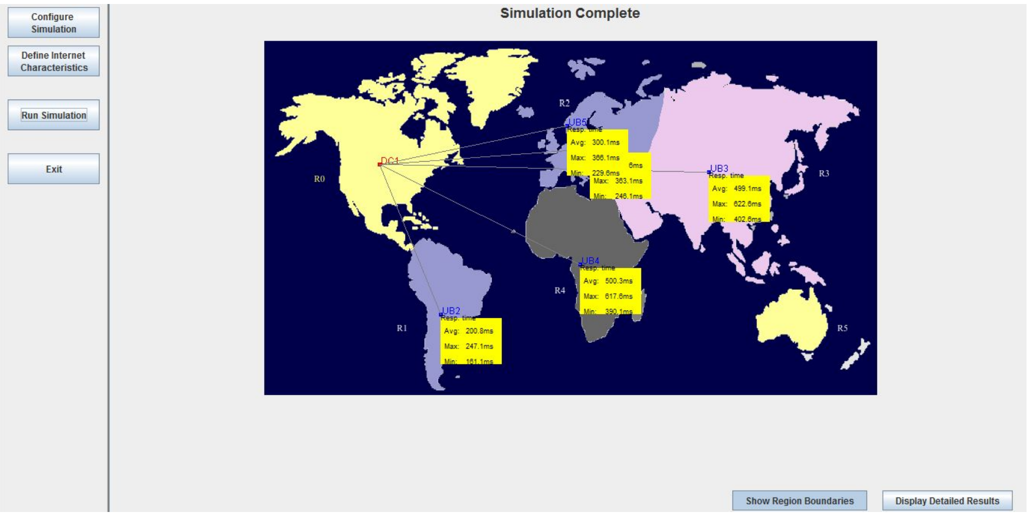
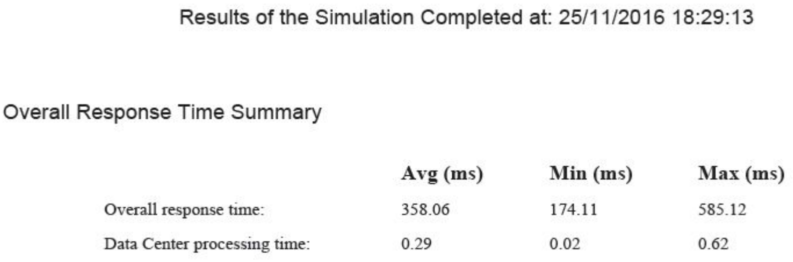


Fig -1 Cloud Analyst Tool

Table -2 Round Robin Response Time Result

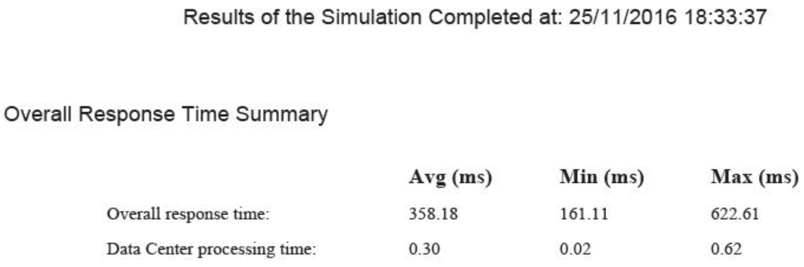


Results of the Simulation Completed at: 25/11/2016 18:29:13

Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	358.06	174.11	585.12
Data Center processing time:	0.29	0.02	0.62

Table -3 ESCE Response Time Result

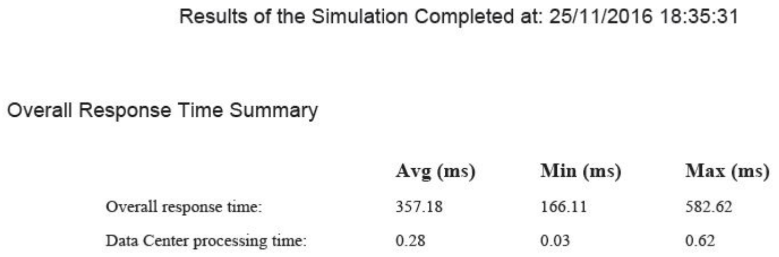


Results of the Simulation Completed at: 25/11/2016 18:33:37

Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	358.18	161.11	622.61
Data Center processing time:	0.30	0.02	0.62

Table -4 Throttled Response Time Result



Results of the Simulation Completed at: 25/11/2016 18:35:31

Overall Response Time Summary

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	357.18	166.11	582.62
Data Center processing time:	0.28	0.03	0.62

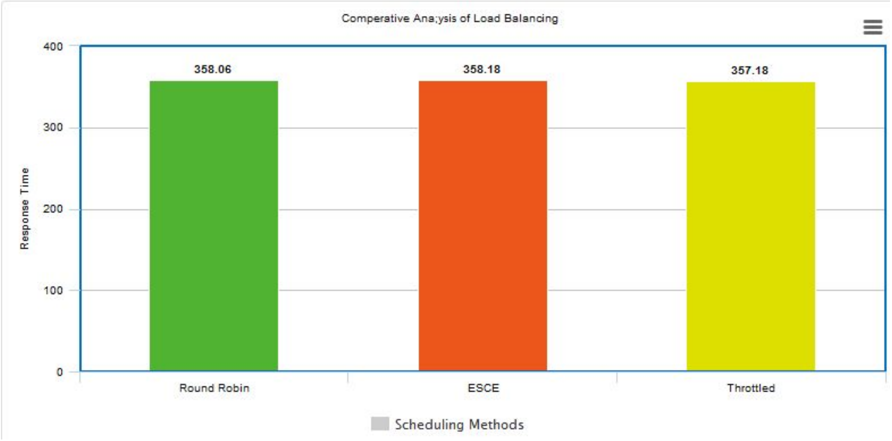


Fig -2 Comparison of existing Load Balancing Algorithm

VI. CONCLUSION

We have discussed current existing static and dynamic load balancing algorithms and compared it. Proposed improved max-min algorithm which will reduce system's makespan and in cloud Analyst simulation tool, we compared response time of three existing algorithms and analyzed it.

REFERENCES

1. *Aswathy B Namboothiri,,Dr. R. Joshua Samuel Raj,A Comparative Study on Job Scheduling Algorithm Augmenting Load Balancing in Cloud, 2016 Second International Conference on Science Technology Engineering And Management (ICONSTEM)*

2. *Shridhar G.Domanal and G. Ram Mohana Reddy, Load Balancing in Cloud Environment using a NovelHybrid Scheduling Algorithm, 2015 IEEE International Conference on Cloud Computing in Emerging Markets*

3. *Salman Arif, Saad Rehman, Farhan Riaz, Design of A Modulus Based Round Robin Scheduling Algorithm,* 2015 9th Malaysian Software Engineering Conference, Dec. 2015

4. *Vishwas Bagwaiya, Sandeep k. Raghuwanshi,HYBRID APPROACH USING THROTTLED AND ESCE LOAD BALANCING ALGORITHMS IN CLOUD COMPUTING,*

5. *G.Punetha,Sarmila,Dr.N.Gnanambigai, Dr.P.Dinadayalan. SurveyonFault Tolerant –Load Balancing Algorithmsin Cloud Computing,,IEEE, ICECS 2015*

6. *Yingchi Mao, Xi Chen and Xiaofang Li,Max–Min Task Scheduling Algorithm for Load Balance in Cloud Computing, Springer India 2014*

7. *Ratan Mishra1 and Anant Jaiswal,Ant colony Optimization: A Solution of Load balancing in Cloud, International Journal of Web & Semantic Technology (IJWesT) Vol.3, No.2, April 2012*

8. *Khadijah Bahwaireth1, Lo'ai Tawalbeh1,2, Elhadj Benkhelifa3, Yaser Jararweh2* and Mohammad A. Tawalbeh3Experimental comparison of simulation tools for efficient cloud and mobile cloud computing applications, EURASIP Journal on Information Security (2016) .*