IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

SOM ALGORITHM FOR CLOUD DATA CENTRE LOAD BALANCING

Ms. Sonpreet Juneja*, Mr.Abhay Kothari

* M.Tech Scholar, CSE Department, Acropolis Institute of Technology & Research, Indore, India
Prof, CSE Department, Acropolis Institute of Technology & Research, Indore, India

ABSTRACT

Cloud computing is an emerging technology in parallel and distributed computing which requires large amount of infrastructure and resources. To optimally serve the needs of the clients all over the world, their providers have leveraged data centres at different geographical locations. Load Balancing is one of the key issue in cloud computing. A load can be a CPU load, memory capacity or network load. It is the process of distributing workload equally on all servers in order to improve resource utilization and response time. Such algorithms mainly aim at ignoring a state where some nodes are heavily loaded while others are lightly loaded or idle. In my presented work, comparative study of Round Robin, Honey Bee Foraging and Genetic Algorithm is conducted and is evaluated using CPU utilization time. The proposed Algorithm SOM (Self Organizing Map) provides optimum and efficient scheduling with low resource consumption and it is simulated using CloudSim.

KEYWORDS: Cloud Computing, Load Balancing, CPU Utilization time, Kohonen's SOM and CloudSim Tool.

INTRODUCTION

The cloud infrastructure is invented to deal with the huge number of request and provide efficient computational environment for different aspects of user applications. Thus a significant amount of request is made over the cloud servers and most of the time that busy in resolving the request of users of different time zones. The word "cloud" often refers to the Internet and more precisely to some data centre connected to the Internet. Clouds have evolved as the next generation platform that facilitates creation of wide-area on-demand renting of computing or storage services for hosting application services. Additionally the cloud offers maintenance fee, installation free and pay per usage basis service to the end client. Therefore the popularity of cloud computational engines as well as data centres are increases rapidly. But due to this arising demand of the computational advantage need optimum techniques to satisfy the user requests. Because due to huge request that experience highly variable workloads and requires high availability and performance.

In order to find the efficient and effective availability and performance the load balancing algorithm helps to resolve the issues of frequently arises workloads. Thus in this presented work the various techniques and methods are investigated for load distribution among the available processing units. The investigation of these methods is provided on the basis of the experimentations with the cloud computing environment. Additionally need to find the performance of the promising techniques which are providing the provision of the load balancing in computational cloud for efficient scheduling of jobs during highly loaded servers.

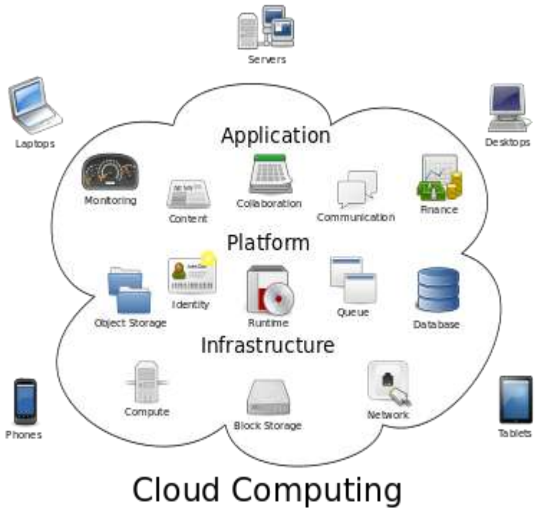


Fig.1 Cloud Computing Components

PROPOSED WORK

Load balancing is a computer networking method for distributing workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units or disk drives. Due to the unpredictable nature of work load in the cloud servers load balancing techniques helps to manage the work load and handle the request

Load balancing aims to optimize resource use, maximize throughput, and minimize response time estimation of load, comparison of load, stability of different system, performance of system, interaction between the nodes, nature of work to be transferred and selecting of nodes.A real life example of load balancing is ***surfing of websites***. Without efficient load balancing algorithm, user experiences delays, timeouts and long responses.

A. Problem Domain

The cloud computing is adoptable for large scale service distribution, data storage, applications and data owner management for different business and educational domains. But to rapidly increasing demand of the cloud applications there are required to handle the cloud server load. Most commonly in multitasking and client/server environments these issues are much common. There is more than one strategy for handling the cloud server load is previously proposed and implemented. Among them required to investigate the best method. And a new solution is desired for providing efficient scheduling with low resource consumption.

B. Solution Domain

SOM is a novel algorithm for improving the elasticity in cloud which can provide a uniform distribution of load in all data centres. The difference between our proposed approaches with previous approach is that it is able to adaptively select next data centre not only based on their topological distances but also based on their load and their density in each setup phase by using SOM neural network. It's a contractive allocation, mapping many tasks to the particular processors in order to minimize communication while using as much processors as available.

The simulation of the cloud platform is prepared to provide the load over the cloud host for execution of a sequence of real time domain workload. Here a log is provided as input, which contains the real time work of actual cloud host for 24 hours.

C. Simulation Architecture

In order to perform the simulation basically some assumptions are defined first and then after using the architectural dependencies the simulation model is presented. There are n number of processing units are connected through the dedicated line or network. Additionally that is directly connected through a load balancer which is connected through the internet connection. In this conditions the through the internet number of requests for job execution is arises. These submitted jobs are queued on load balancing server. Now there is a job allocation technique is required which allocate jobs to the processing units.

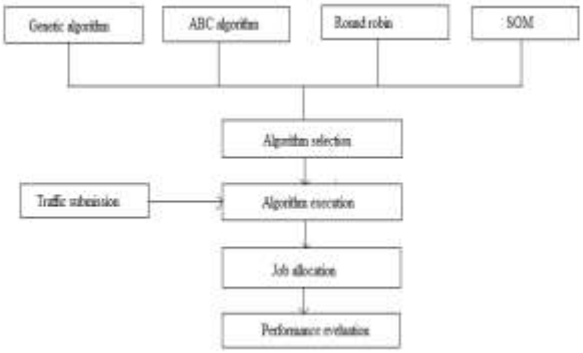


Fig2. Load balancing architecture

In this given diagram first these algorithms are implemented using the JAVA technology and then a new graphical user interface is created for selecting appropriate algorithm. The selected algorithm processes the input traffic as job queue and the second parameter is provided as the number of processing units. The algorithm finds the optimum combinations of workload to satisfying with the selected processing element over the cloud infrastructure. After processor allocation the system is performance is measured and their comparative study is performed. The difference between our proposed approaches with previous approach is that it is able to adaptively select next data centre not only based on their topological distances but also based on their load and their density in each setup phase by using SOM neural network. It's a contractive allocation, mapping many tasks to the particular processors in order to minimize communication while using as much processors as available.

D. Research Methodology

i. Genetic Algorithm

It has been used as a soft computing approach, which uses the mechanism of natural selection strategy. In GAs, a population of candidate solutions (called individuals) is evolved toward better solutions with the application of genetic operators: selection, crossover, and mutation. Each solution has a set of properties (called chromosomes) which can be recombined, randomly mutated and altered to form a new set of solutions named generation. In each iteration the population of solutions is evaluated considering the fitness of every individual (usually a value related to the objective function). The more fit individuals are selected and survive for consequent iterations in every new generation, a set of strings is created using information from the previous ones.

ii. Honeybee Foraging

This whole algorithm is based on the process of honeybees finding the food and alarming others to go and eat the food. First forager bees go and find their food. After coming back to their respective beehive, they dance. After seeing the strength of their dance, the scout bees follow the forager bees and get the food. The more energetic the dance is, the more food available is. So this whole process is mapped to overloaded or under loaded virtual servers. The server processes the requests of the clients which is similar to the food of the bees. As the server gets heavy or is overloaded, the bees search for another location i.e. client is moved to any other virtual server. In this way, this whole technique works. Its merits are Maximize throughput & Low overheads. Demerits are Low Priority Load.

iii. Round Robin

This algorithm uses the round robin scheduling of processes. It mainly focuses on distributing load equally to all nodes. In this approach, the scheduler allocates one virtual machine (VM) to each node in a cyclic manner. This process is repeated until all the nodes have been allocated at least on VM and then it returns to first node again. The idea behind round robin is that, multiple IP addresses are assigned to a single domain name and clients are virtualized and expected to choose the server.

The disadvantage with this algorithm is it considers only current load on each VM and there is an additional load vector which decides load distribution.

E. Proposed Algorithm (Modified SOM)

Kohonen's is clustering based self-organizing neural networks. It has a two-layer topology. The first layer is the input layer; the second layer is itself a network in a plane. Every unit in the input layer is connected to all the nodes in the grid in the second layer. Furthermore the units in the grid function as the output nodes.

It works on the principal of “winner takes all”. A winning neuron is the one which corresponds with input vector in the best way.

Based on unsupervised learning, which means that no human intervention is needed during the learning and little needs to be known about the characteristics of the input data.

```
1. for all P ∈ p do
        $lb_f(p) \leftarrow \frac{8n}{(m\ (degree(p)+1))}$
2. If s ≤ 0.75 and m iterations of the Kohonen process performed since last load-balancing step then
For all neurons p do:
If $(load_{new}(p) - load_{avg}) X (load_{old}(p) - load_{avg}) <$ 0 then
3.$lb_f \leftarrow \frac{lb_f}{2}$
4. End for
5. for all neighbouring neurons p, q with $load(p) < load(q)$ do
        $\Delta^{total}(p,q) \leftarrow \left(\frac{lb_f(p) + lb_f(q)}{2}\right) X\Delta(p,q)$
6. $\|w_p\| \leftarrow \|w_p\| \pm \Delta^{total}(p,q)$
7. End for
8. End if
```

IMPLEMENTATION

The simulation of the cloud environment using CloudSim discrete event simulator requires configuring first the cloud infrastructure, then after the simulation scenarios are necessary to be write down using codes.

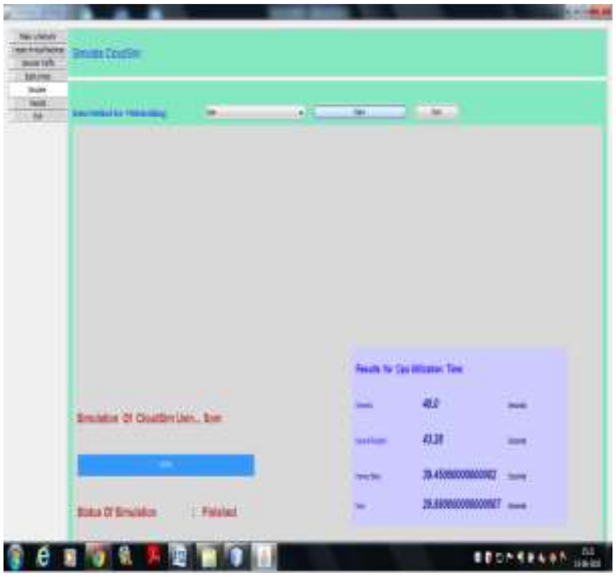
Table 1. Cloud Infrastructure Parameters

Simulation Parameters	Values
Number of Virtual	20
Number of Cloudlets	40
VM image size	10000 MB
RAM	512 MB
Number of instruction per	1000 MIPS
Processing Units	1

After finalizing the cloud infrastructure required to design simulation scenario, for that purpose some network parameters are also required to utilize, the network setup is given using the below given table.

Table 2. Network Parameters

Network parameters	Values
Resource length	1000
File Size	300 MB
Host memory	2048 MB
Storage	1000000MB
Band width	10000
Output size	300 MB
Instruction per second for	1000



Simulation Of CloudSim Usin... Som

Status Of Simulation : Finished

Results for Cpu Utilization Time

40.0
43.28
39.45000000000002
28.86000000000007

Fig 3. Simulation showing SOM giving best CPU utilization time

RESULT ANALYSIS

The performance of scheduling of all four algorithms is given using figure 5.3 where the X axis shows the different process in queue and Y axis shows the average CPU consumption in terms of seconds.

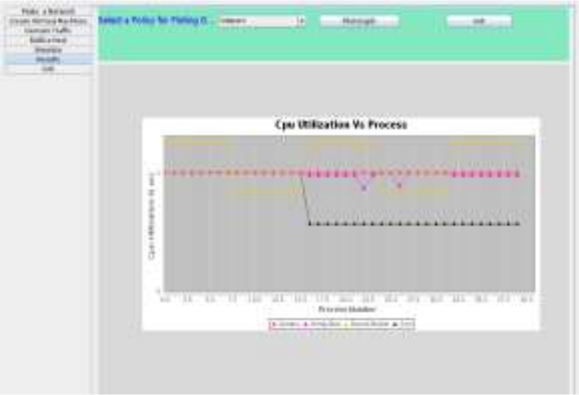


Figure 3 CPU time for job allocation

In the given figure 3 the performance evaluation and execution of jobs during the different load balancing algorithms are provided. For demonstrating the performance of individual load balancing algorithm pink line shows the performance of genetic algorithm, yellow line shows the performance of the artificial honey bee colony algorithm, black line shows the performance of round robin algorithm and the red line shows the performance of SOM algorithm. In order to represent the performance of the algorithms the X axis shows the number of processes in job queue and the Y axis shows the corresponding CUP time cycle required to submit a job into processing unit. According to the obtained performance of the system the genetic algorithm, artificial honey bee colony algorithm and the round robin algorithm needs more time to execute the entire job in job queue and the SOM algorithm provides the too few amount of time to execute the entire job in queue.

CONCLUSION

Cloud computing is a new generation technology which provide ease in efficient computing, managing the resources and easy pay per usage basis interface of different software and hardware resources. Therefore the demand of this infrastructure is rapidly increases among the various organizations. Due to this traffic or workload on servers are arises frequently and the servers are most of the time are overloaded. These loads on server results the software and hardware label conflicts and issues. Thus in order to overcome these issues scheduling or load balancing techniques are utilized which are help to find the less loaded processing units among the available computational elements and by allocating the processes into these elements they frequently able to resolve the load on server machines.

Therefore in this presented work the different available techniques of load balancing is investigated and for finding the most optimum technique for load balancing the comparative study is performed. In order to perform the comparative performance analysis the genetic algorithm, round robin, artificial honey bee colony and self-organizing map algorithms are implemented with the help of JAVA technology and the CloudSim discrete event simulator.

REFERENCES

1. Akshay Daryapurkar, Mrs. V.M. Deshmukh, “Efficient Load Balancing Algorithm in Cloud Environment”, International Journal Of Computer Science And Applications Vol. 6, No.2, Apr 2013
2. Argha Roy, Diptam Dutta, “Dynamic Load Balancing: Improve Efficiency in Cloud Computing”, International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-2, Issue-4), April 2013
3. Alexa Huth and James Cebula, “The Basics of Cloud Computing”, © 2011 Carnegie Mellon University, Produced for US-CERT Introduction to Cloud Computing, White Paper, Dialogic, 2013
4. Nariman Mirzaei, Cloud Computing, Fall 2008,Community Grids Lab, Indiana University Pervasive Technology Institute
5. Mike Ricciuti, “Stallman: Cloud computing is 'stupidity'”, http://news.cnet.com/8301-1001_3-10054253-92.html
6. Understanding The Cloud Computing Stack SaaS, Paas, IaaS, © Diversity Limited, 2011 Non-commercial reuse with attribution permitted,http://broadcast.rackspace.com/hosting_knowledge/whitepapers/Understanding-the-Cloud-Computing-Stack.pdf
7. Maimit A. Patel, Asst. Prof. Rutvik Mehta, “A Comparative Study of Heuristic Load Balancing in Cloud Envioурnment”, International Journal of Advance Engineering and Research Development Volume 2, Issue 1, January -2015.
8. Pradeep Naik, Surbhi Agrawal, Srikanta Murthy, “A survey on various task scheduling algorithms toward load balancing in public cloud”, American Journal of Applied Mathematics 2015; 3(1-2): 14-17 Published online December 30, 2014.
9. Siva Theja Maguluri, R. Srikant, Lei Ying, “Stochastic Models of Load Balancing and Scheduling in Cloud Computing Clusters”, 2012 Proceedings IEEE Infocom.