ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

SLA Based Extended GA Algorithm for Load Balancing

[I]Amrita Verma, [II]Ashish Jalota
[I]M.Tech. Student, [II]Assistant Professor, CSE
[I,II]Desh Bhagat University, Mandi Gobindgarh, Punjab, India

Abstract
Load balancing is a main challenge in cloud environment. For better management of available good load balancing techniques are required. So that loads balancing in cloud becoming more interested area of research. And through better load balancing in cloud, performance is increased and user gets better services. Load balancing is helped to distribute the dynamic workload across multiple nodes to ensure that no single node is overloaded. It helps in proper utilization of resources. It also improves the performance of the system. Many existing algorithms provide load balancing and better resource utilization. There are various types load are possible in cloud computing like memory, CPU and network load. Load balancing is the process of finding overloaded nodes and then transferring the extra load to other nodes. Genetic algorithm gives better load balancing techniques by Kousik Dasgupta, 2013. SLA based extended genetic algorithm has been introduced to improve the performance of the load balancing in cloud computing.

Keywords
Cloud Computing, Cloud Service Model, Load Balancing, SLA

I. Introduction

Cloud computing is a new technology. It providing online resources and online storage to the user's .It provide all the data at a lower cost. In cloud computing users can access resources all the time through internet. They need to pay only for those resources as much they use .In Cloud computing cloud provider outsourced all the resources to their client. There are many existing issues in cloud computing [2]. It allows the users to use resources according to the arrival of their needs in real time. Thus, we can say that cloud computing enables the user to have convenient and on-demand access of shared pool of computing resource such as storage, network, application and services, etc On pay per use basis. The main problem is load balancing in cloud computing. Load balancing helps to Distribute all loads between all the nodes. It also ensures that every computing resource is distributed efficiently and fairly. Load balancing is a relatively new technique that provides high resource utilization and better response time [1].

1. Cloud computing Architecture

Cloud computing is growing in the real time environment. Figure 1 illustrating the three basic service layers that constitute the cloud computing. It provides three basic services that are Software as a Service, Platform as a Service and Infrastructure as a Service [1]. Services mean different types of applications provided by different servers across the cloud [5].

a. Software as services

SaaS provides the vendor with the software. Vendor pays for the time of using the software and can use it anywhere. There is no need to buy the software. It is referred to as "on- demand" software. It works on application layer. Eg. Google App. [4].

b. Platform as services

It provides a platform where resources are available and consumers can themselves create the required applications for e.g. web application data and database data. Providers provide network, servers, and storage services. It works on platform layer. Eg: Microsoft Azure [4].

c. Infrastructure as services

This model provides users with the hardware on rent. It provides virtual machine as service to users. For e.g. server space, network equipment, memory, storage space. In short consumer buys virtual space and works on it. It works on hardware layer. Eg: Amazon EC2 [4].

2. Load Balancing

Load balancing is one of the main issues related to cloud computing. The load can be a memory, CPU capacity, network or delay load. Load balancing is a technique to distribute the load among the various nodes of the distributed system to improve the resource utilization and for better performance of the system [1]. This can help to avoid the situation where nodes are either heavily loaded or under loaded in the network. Load balancing is the process of ensuring the evenly distribution of work load on the pool of system node or processor so that without disturbing, the running task is completed [4].

II. Litereture Survey

Dasgupta, K et.al (2013) states that A genetic algorithm based load strategy for cloud computing has been developed to provide an efficient utilization of resources in cloud environment .Load balancing which is the main challenges in cloud computing so it distributes the dynamic workload across all resources. This can be considered as an optimization problem and a good load balancer should adapt its strategy to the changing environment and the types of tasks. This paper proposes a novel load balancing strategy using Genetic Algorithm (GA). The algorithm thrives to balance the load of the cloud infrastructure while trying minimizing the make span of a given tasks set. The proposed load balancing strategy has been simulated using the Cloud Analyst simulator. Simulation results for a typical sample application shows that the proposed algorithm outperformed the existing approaches like First Come First Serve (FCFS), Round Robbing (RR) and a local search algorithm Stochastic Hill Climbing (SHC).

Desai, T, et.al (2013) proves that (A Survey Of Various Load Balancing Techniques And Challenges In Cloud Computing) Cloud computing is emerging technology .It provides shared resources, information, software packages and other resources as per client requirements at specific time. For better management of available good load balancing techniques are required. So those load balancing in cloud becoming more interested area of

research. And through better load balancing in cloud, performance is increased and user gets better services. They have discussed many different load balancing techniques used to solve the issue in cloud computing environment**.**

Xu, G,et.al (2013) States that(A load balancing model based on cloud partitioning for the public cloud) Load balancing in the cloud computing environment has an important impact on the performance. Good load balancing makes cloud computing more efficient and improves user satisfaction. This article introduces a better load balance model for the public cloud based on the cloud partitioning concept with a switch mechanism to choose different strategies for different situations. The algorithm applies the game theory to the load balancing strategy to improve the efficiency in the public cloud environment.

Roy, A, et.al (2013) states that (Dynamic Load Balancing: Improve Efficiency in Cloud Computing) When many clients request the server simultaneously, server is overloaded which causes fault. In this approach the load balancing technique is used to avoid fault. There are various fault tolerance techniques in existing cloud computing. They are self healing, job migration, static load balancing and replication. There are some drawbacks in this technique. In this proposed method, the dynamic load balancing technique is used to avoid this fault tolerance in cloud computing. The Dynamic Load Balancing algorithm checks the utilization of the CPU. If CPU has less utilization as given in the algorithm, it responses the client request otherwise the request is shifted to another server with the help of load balancer. This technique gives a better result**.**

Ray, S, et.al (2012) mentioned that (Execution analysis of load balancing algorithms in cloud computing environment) Load balancing is a methodology to distribute workload across multiple computers, or other resources over the network links to achieve optimal resource utilization, maximize throughput, minimum response time, and avoid overload. A few existing scheduling algorithms can maintain load balancing and provide better strategies through efficient job scheduling and resource allocation techniques as well. This paper presents a review of a few load balancing algorithms or technique in cloud computing. The objective of this paper is to identify qualitative components for simulation in cloud environment and then based on these components, execution analysis of load balancing algorithms are also presented.

Lu, Yi, et.al (2011) (Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services) propose a novel class of algorithms called Join-Idle-Queue (JIQ) for distributed load balancing in large systems. Unlike algorithms such as Power-of-Two, the JIQ algorithm incurs no communication overhead between the dispatchers and processors at job arrivals. They analyze the JIQ algorithm in the large system limit and find that it effectively results in a reduced system load, which produces 30-fold reduction in queuing overhead compared to Power-of-Two at medium to high load. An extension of the basic JIQ algorithm deals with very high loads using only local information of server load.

Buyya, R, et.al (2010) states that **(**Intercloud: Utility-oriented federation of cloud computing for scaling of application services) Cloud computing providers have setup several data centers at different locations over the Internet to provide services to their customers around the world. However, existing systems do not support mechanisms and policies for load distribution among different Cloud-based data centers in order to determine optimal location for hosting application services to achieve QoS levels. Further, the Cloud computing providers are unable to predict geographic distribution of users consuming their services, hence the load coordination must happen automatically, and distribution of services must change in response to changes in the load. To counter this problem, they advocate creation of federated Cloud computing environment (InterCloud) that facilitates just-in-time, opportunistic, and scalable provisioning of application services, consistently achieving QoS targets under variable workload, resource and network conditions.

Hu, J, et.al (2010, December) states that (A scheduling strategy on load balancing of virtual machine resources in cloud computing environment) The current virtual machine(VM) resources scheduling in cloud computing environment mainly considers the current state of the system but seldom considers system variation and historical data, which always leads to load imbalance of the system. This paper presents a scheduling strategy on load balancing of VM resources based on genetic algorithm. This strategy solves the problem of load imbalance and high migration cost by traditional algorithms after scheduling. Experimental results prove that this method is able to realize load balancing and reasonable resources utilization both when system load is stable and variant.

III. Problem Formulation

As clearly mentioned in the previous algorithms that still scope of improvement is there because they not used any method prior for the selection of population. Though Cloud computing is dynamic but at any particular instance the said problem of load balancing can be formulated as allocating N number of jobs submitted by cloud users to M number of processing units in the Cloud. Each of the processing unit will have a processing unit vector (PUV) indicating current status of processing unit utilization. This vector consists of MIPS , indicating how many million instructions can be executed by that machine per second, α, cost of execution of instruction and delay cost L. The delay cost is an estimate of penalty, which Cloud service provider needs to pay to customer in the event of job finishing actual time being more than the deadline advertised by the service provider. Dasgupta, K., Mandal, B., Dutta, P., Mandal, J. K., & Dam, S. (2013). A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing. Procedia Technology, 10, 340-347. The problem is that it chooses the population randomly as discussed in the proposed algorithm.

IV. Proposed Algorithm

Algorithm: Extended SLA based Genetic Algorithm for Load Balancing

Step 1: Collect the information from all the data centers about the VM's according to the following parameters: computing power of host/physical server in terms of its core processor, processing speed, memory, storage etc.

Step 2: Allocate strength count according to the computing power of the VM's in Datacenter. If one VM is capable of having twice as much load as the other, the powerful server gets a weight of '2' or if it can take four times load then server gets a weight of

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

'4' and so on.

For example:

- Host server with guard core processor, 4GB of memory 4TB of Storage space and 1000000 bandwidth will have weighted count=4 and so on.
- Host server with 4 core processor, 8GB of memory, 2TB of Storage space and 1000000 bandwidth will have weighted count=4
- Host server with single core processor, 1GB of memory, 1TB of Storage space, 1000000 bandwidth will have weighted count=1

Step3: Genetic VM Load Balancer maintains an index table of VMs, associated weighted count and the number of requests currently allocated to the VM. At start all VM's have 0 allocations.

Step 4: Load the SLA agreement of the user from where job request is coming.

Step 5: Initialize a population of processing from index table unit after encoding them into binary strings [Start].

Step 6: Evaluate the fitness value of each population using equation according to SLA agreement with user. Priority should be given to the premium users and less functionality to the general users.

Step 7: While either maximum number of iteration are exceeded or optimum solution is found Do:

Step 7(a): Consider chromosome with lowest fitness twice and eliminate the chromosome with highest fitness value to construct the mating pool [Selection].

Step 7(b): Perform single point crossover by selecting the crossover point to form new offspring matched with the index of VM. [Crossover]

Step 7(c): Mutate new offspring with a mutation probability of (0.05) [Mutation].

Step 7(d): When a request to allocate a new VM from the Data Center Controller arrives, it parses the table and identifies the least loaded VM.

Step 7(e): Place new offspring as new population and use this population for next round of iteration [Accepting].

Step 7(f): Test for the end condition [Test].

Step 8: End.

V. Conclusion

In this paper, An SLA based extended genetic algorithm for load balancing strategy for Cloud Computing has been developed to provide an efficient utilization of resource in cloud environment. Analysis of the results indicates that the proposed strategy for load balancing not only provides new techniques but also guarantees the Quality of services requirement of customer job. Though it has been assumed in last paper all the jobs are of the same priority which is improved in this paper by Priority levels using SLA based extended genetic algorithm, this can be accommodated in the JUV and subsequently taken care in fitness function. Also a very simple approach of Extended Genetic Algorithm is used variation of the crossover and selection strategies for more efficient results.

1. Flow Chart

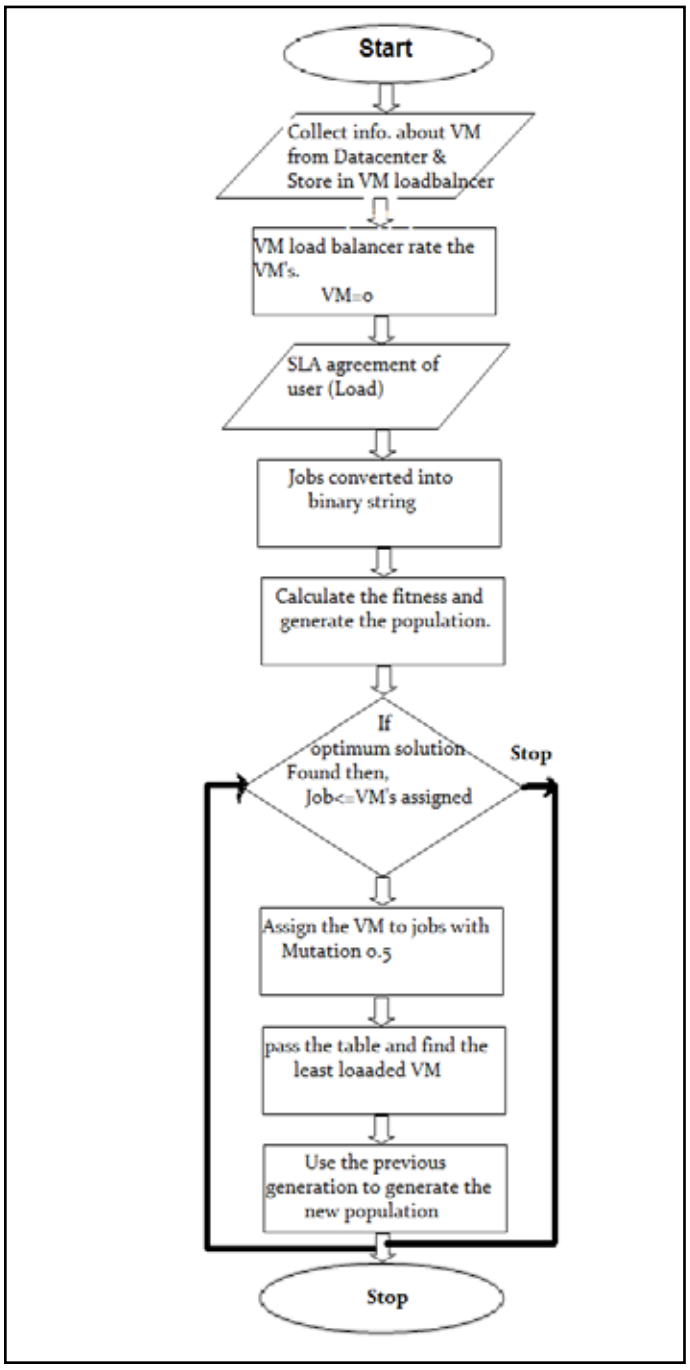

Fig. 1: Flow chart of SLA Based Genetic Algorithm

References

[1]. *Desai, T., & Prajapati, J.(2013) A Survey Of Various Load Balancing Techniques And Challenges In Cloud Computing. (IJSTR)*

[2]. *Kaur, R., Luthra, l. (2013) Load Balancing in cloud computing. (ACEEE)*

[3]. *Dasgupta, K., Mandal, B., Dutta, P., Mandal, J. K., & Dam, S. (2013). A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing. (Procedia Technology)*

[4]. *Gupta, R. (2014) Review on existing load balancing techniques of cloud computing. (IJARCS)*

[5]. *Buyya, R., Ranjan, R., & Calheiros, R. N. (2010). Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. (IEEE)*

[6]. *Hu, J., Gu, J., Sun, G., & Zhao, T. (2010, December). A*

scheduling strategy on load balancing of virtual machine resources in cloud computing environment. In Parallel Architectures, Algorithms and Programming (PAAP), 2010 Third International Symposium on IEEE.

[7]. Lu, Y., Xie, Q., Kliot, G., Geller, A., Larus, J. R., & Greenberg, A. (2011). Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services.

[8]. Ray, S., & De Sarkar, A. (2012). Execution analysis of load balancing algorithms in cloud computing environment. International Journal on Cloud Computing: Services and Architecture (IJCCSA).

[9]. Xu, G., Pang, J., & Fu, X. (2013). A load balancing model based on cloud partitioning for the public cloud IEEE.

[10]. Paulin, F., V.shanti (2014). A Load Balancing Model Using Firefly Algorithm in Cloud Computing IJCS.

[11]. Roy, A., & Dutta, D. (2013). Dynamic Load Balancing: Improve Efficiency in Cloud Computing IJERMT.