International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

An Integrated Approach of Token based Heuristic Algorithm and Genetic Algorithm for Cloud Computing Load Balancing

Ritika Trivedi†* and Ubeeka Jain†

†Department of Computer Science & Engg. RIEIT Rail Majra, India

Accepted 28 Nov 2015, Available online 03 Dec 2015, **Vol.5, No.6 (Dec 2015)**

Abstract

Cloud Computing revolves around internet based acquisition and release of resources from a data center. Being internet based dynamic computing; cloud computing also may suffer from overloading of requests. With Tremendous increase in the users and their demand of different services on the cloud computing platform, fruitful or efficient usage of resources in the cloud environment became a critical concern. Load balancing is playing a vital role in maintaining the rhythm of Cloud computing. However this aspect of cloud computing has not been paid much attention yet. Although load balancing is being considered as an important aspect for other allied internet based computing environments such as distributed computing, parallel computing etc. Many algorithms had been proposed for finding the solution of load balancing problem in these fields. But very few algorithms are proposed for cloud computing environment. Since cloud computing is significantly different from these other types of environments, separate load balancing algorithm need to be proposed to cater its requirements. In this paper, we have proposed an integrated approach of Token Based Heuristic Algorithm (THA) and Genetic Algorithm. The proposed mechanism has been implemented and found to provide satisfactory results. This algorithm gives comparison with Round Robin Algorithm, Random Algorithm and Genetic Algorithm.

Keywords: *Token Based Heuristic Algorithm, Genetic Algorithm, Load Balancing, Round Robin Algorithm, Random Algorithm*

Introduction

Load balancing (Cybenko, G., 1989) is a method that distributes the workload among diverse nodes in the given environment such that it ensures no node in the system is over loaded or sits idle for any instant of time. An efficient load balancing algorithm will make sure that every node in the system does more or less same volume of work. The responsibility of load balancing algorithm is that to map the jobs which are set forth to the cloud domain to the unoccupied resources so that the overall available response time is improved as well as it provides efficient resource utilization. Balancing the load became one of the crucial concerns in cloud computing since we cannot predict the number of requests that are issued at each second in cloud environment. The unpredictability is due to the ever changing behaviour of the cloud. The main focus of load balancing in the cloud domain is in allocating the load dynamically among the nodes in order to satisfy the user requirements and to provide maximum resource utilization by assorting the overall available load to distinct nodes (Mondal, B. *et al*., 2012; Lombardi, F., *et al*. 2011).

*Corresponding author: **Ritika Trivedi**

An appropriate or an ideal load balancing algorithm help in making use of the available resources most favorably, thereby ensuring no node is over loaded or under loaded. Load balancing enables scalability, avoids bottlenecks and also reduces time taken to give the respond. Many load balancing algorithm (Xu, X., 2012) have been designed in order to schedule the load among various machines. But so far there is no such ideal load balancing algorithm has been developed which will allocate the load evenly across the system. It has been proved that allocating the tasks evenly across the system is considered to be an NP complete problem (Papadimitriou, C. H., 1977).

In this paper, we have proposed an integrated concept of Token Based Heuristic Algorithm (Xu, Y. *et al*., 2011) and Genetic Algorithm (Horn, J. *et al*., 1994) for the cloud computing load balancing technique. The proposed concept is evaluated by comparison with Round Robin algorithm, Random algorithm and Genetic algorithm by considering the evaluation parameters of processing time, response time, mean error etc.

The rest of the paper is organized in the following manner: Section II describe the basic concept involved in the hybridization concept. Section III explains the proposed concept. Section IV gives the result values

and comparison with other concepts. Section V concludes the paper.

Basic Concepts

This section covers the concept of basics of Token based Heuristic algorithm and Genetic Algorithm. These two concepts are explained as below:

Token based Heuristic Algorithm

The main objective of Token Routing is to minimize the system cost by moving the tokens around the system. But in a scalable cloud system agents cannot have the enough information of distributing the work load due to communication bottleneck (Sureshbabu, D., *et al*., 2014). So the workload distribution among the agents is not fixed. The drawback of the token routing algorithm can be removed with the help of heuristic approach of token based load balancing. This algorithm provides the fast and efficient routing decision. In this algorithm agent does not need to have an idea of the complete knowledge of their global state and neighbours working load. To make their decision where to pass the token they actually build their own knowledge base. This knowledge base is actually derived from the previously received tokens. So in this approach no communication overhead is generated. In this algorithm, each site maintains information about the state of other sites in the system and uses it to select a set of sites that are likely to have the token. The site requests the token only from these sites, reducing the number of messages required to execute the Critical Section (PARK, J. S. *et al*. 2014).

Genetic Algorithm

Genetic algorithms (GAs) can be described as a heuristic search and optimisation technique that is inspired by natural evolution (Jones, G. *et al*., 1997), which transposes the notions of natural evolution to the world of computers and imitates natural evolution. GAs were initially introduced by John Holland for explaining the adaptive processes of natural systems and for creating new artificial systems that work on similar bases. In Nature, new organisms adapt to their environment through evolution. Genetic algorithms evolve solutions to the given problem in a similar way (Mühlenbein, H. *et al*., 1991). The implementations of genetic algorithms can significantly differ in the way of constructing a new population. Some implementations create a separate population of new individuals in every generation by applying genetic operators. Other implementations extend the current population by adding new individuals and then create the new population by omitting the least fit individuals. There are GAs that do not use generations at all and instead have continuous replacement. According to the method of creating a new population, the GA adapts other operators, especially the selection operator, as well the position of the fitness calculation of the individuals. On the one hand, a simple core of the genetic algorithm is available; on the other hand, we must consider the specific problem that we want to solve. Presentation, control mechanisms, fitness function, the method of initialisation and genetic operators should also be appropriately adapted to the problem (Houck, C. *et al*. 1995). A GA has a specific strength because standard components can be re-used, with adaptation to many different situations, which eases the implementation.

Proposed Concept

In this section, integrated concept of Token based heuristic algorithm and Genetic Algorithm is applied for the cloud computing load balancing. The main goal of this hybridization concept is to improve Token based heuristic algorithm by selecting the continuous replacement of population solutions with dominating features. This process helps the THA algorithm for the improving the solution to assign the task to VM having lesser load with the continuous change in the task assignment. Genetic algorithm always reconstructs the population for the assignment of tasks with dominating results. When token is transferred from one channel to another, then the assignment of load depends upon the condition provided by the genetic algorithm. This concept is structured as below:

Step 1: Consider we have V as a Cloud Data Center and in this data centre we have 'n' number of the Virtual machines VMs, as V = {vm_1, vm_2, …,vm_n}. We have to maintain the load on these virtual machines by assigning the different tasks to these VMs having optimizing load.
Step 2: Consider a Threshold value 'Th' for each VM. Start the process of Token based heuristic algorithm to assign a predefined task to the VM. For this, token is sent to the each VM for the detection of destination and load value.
Step 3: Token is transferred to each site to assign the task having minimum load. To complete the condition of assignment, check for the threshold value:
If (Assigned task >Threshold Value)
Send the token to next site
If (Assigned task < Threshold Value)
Assign the task
Step 4: If an agent gets task from its neighbor, the agent may infer that its neighbor or the part of network close to the neighbor may not be able to provide the working load beyond the threshold requested. Therefore, the agent is less likely to pass the other tokens with higher thresholds to that neighbour.
Step 5: If agent passed the task to another next location due to threshold value, then check for the next updated site loads each and again.
Step 6: To optimize the VM with continuous updation, apply the concept of genetic algorithm by assuming the population of the genetic algorithm as the sites of VMs and generations as the task. As the population continuous shows the updation values with next task assignment, so it infer accordingly.

6.1: Initialize the population.
6.2: Find the fitness value of each chromosome in the population.
6.3: Reproduce a new population by repeating the following steps.
6.4: Select two individual chromosomes from a population according to selection method, roulette wheel selection.
6.5: Cross over the selected parents if crossover probability met, to produce a new child. Otherwise, the children are an exact copy of parents.
6.6: Mutate each new offspring (child) if a mutation probability met, at each locus (position in the chromosome).
6.7: Place new child in a reproduced population.
6.8: Store Best individual solution and assign the task to best find solution.
Step 7: Repeat the process till the entire load is distributed.
Step 8: Obtain the output as a balanced load.

Results & Discussion

The simulation tool used for the cloud computing load balancing is MATLAB. To do the implementation of cloud computing load balancing using integrated approach of THA and Genetic Algorithm, the system is developed as shown in figure 1. Here, we are showing the load balancing strategy for the Random algorithm, Genetic Algorithm and Hybrid Token based and Genetic algorithm. The figure 1(a) shows the load balancing task by Random Algorithm. In random algorithm task assigned by little triangular shapes as shown in figure, is not properly assigned. This spread of triangular shapes shows the lesser accuracy of the task assignment. Now, proceed towards the figure 1(b), this shows the task assignment in better manner as compare to Random algorithm. The dark blue color on the higher peaks shows the proper task assignment that manages the heavy load.

Figure 1(d) Shows the final solution for the load balancing with hybrid Token and genetic algorithm. Figure 1(c) shows the comparison of Random algorithm, Hybrid THA & Genetic algorithm and Round Robin algorithm for the comparison on the basis of storage capacity and number of generations.

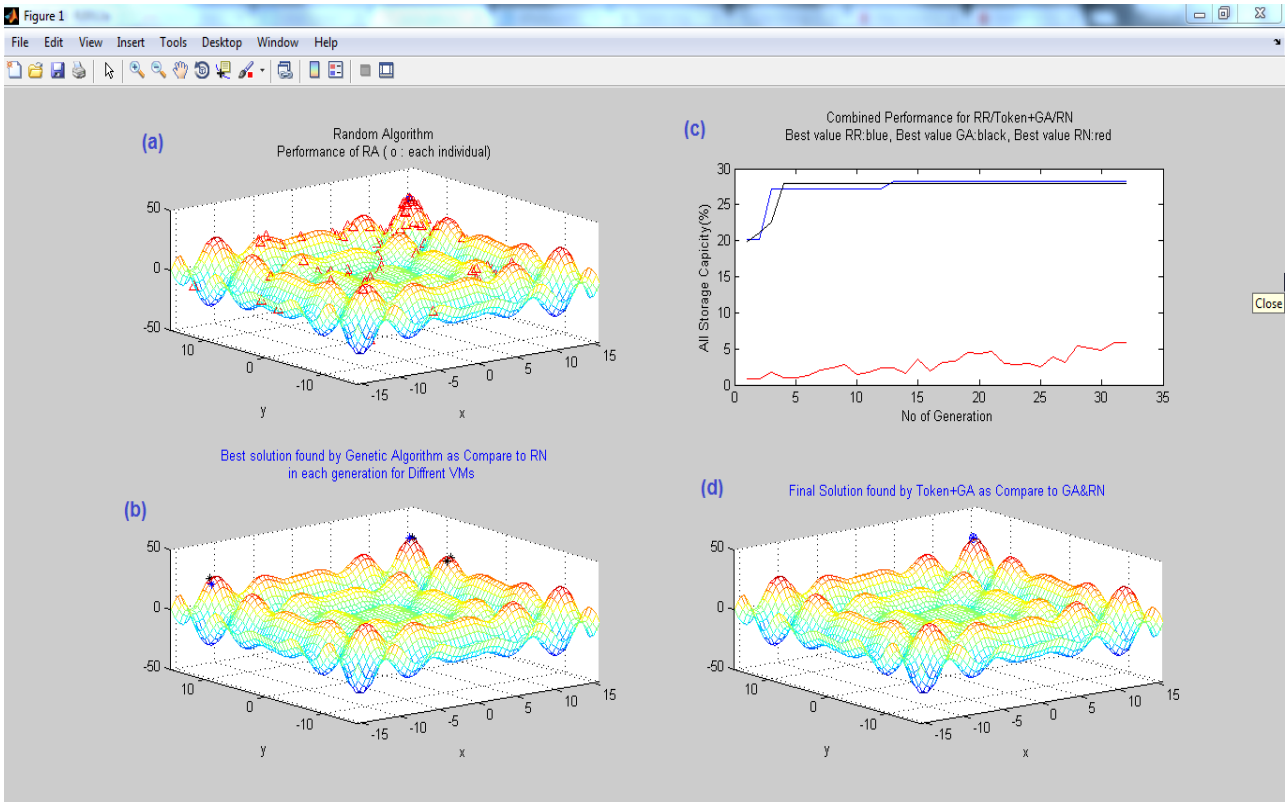

Figure 1: Simulation of Cloud computing load balancing using Hybrid THA and Genetic algorithm

A. Evaluation Parameters

There are various metrics used to evaluate different techniques. In our work we used the parameters of Standard Deviation, Mean Error, MeanVMs Time, Standard Processing Time, Mean Processing Time and Response Time.

These are calculated for the Proposed Concept, Random Algorithm, Round Robin Algorithm and Genetic Algorithm. These are evaluated as below in table I:

Table 1: Evaluation Parameters for the Comparative concepts

Algorithm/Parameters	SD	Mean Error	Mean VM Time	Standard Processing Time	Mean processing Time	Response Time
Token+ Genetic	44.1475	0.7696	1626.5500	20.9271	0.2936	0.0250
Random Algorithm	37.2975	0.5961	3318.2500	20.5782	0.7808	0.1124
Genetic Algorithm	38.6775	0.6463	3372.5000	30.5795	0.6006	0.0489
Round Robin	37.3800	0.5921	4944.8000	37.1251	1.1782	0.1218

From the above comparison table, it shows that Token and Genetic integrated concepts takes lesser time to response and processing time is also less as compare to other algorithm. So, the proposed algorithm is suitable for cloud computing load balancing. The comparison is also shows by the graphical representation of the above algorithms by the graphs below:

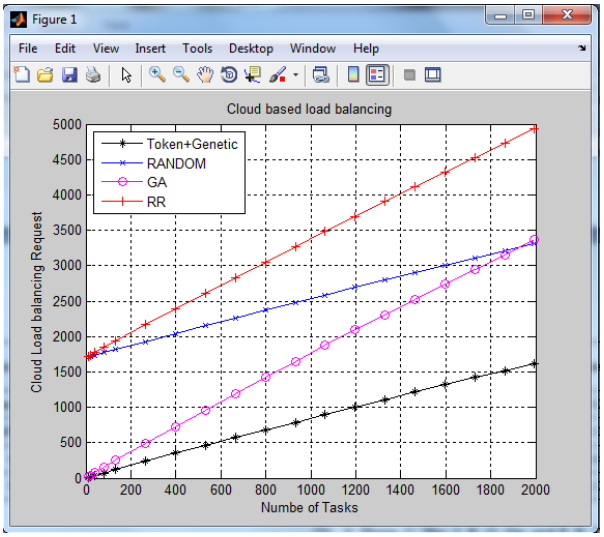

Figure 2: Cloud based load balancing with no. of tasks

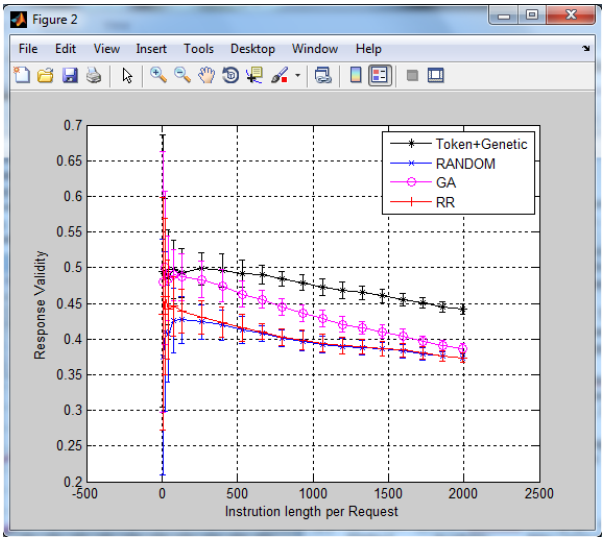

Figure 3: Task assignment and respective response validity

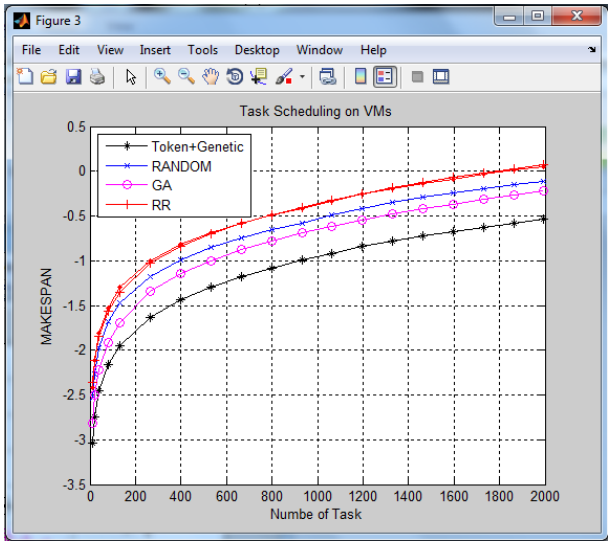

Figure 4: Task Scheduling on VMS

Conclusion

Load balancing is one of the important issues in cloud computing. The current load balancing scheduling algorithms in cloud computing environment have some deficiency and this would affect the performance. Therefore we proposed a hybrid algorithm to enhance the cloud computing performance. The hybrid algorithm based on Token Base Heuristic algorithm and Genetic Algorithm, they take the advantages of both genetic & token based algorithm and consider the evaluation parameters as shown in table I to achieve the objectives. The experiments were implemented in the MATLAB Simulator. From the simulation results, we have found that proposed concept takes less processing time and respond in less time span as compare to other Round Robin, Genetic and Random Algorithm. This was due to the equivalent distributing of loads between all the VMs. This was shown by figure 1 to figure 4.

References

Cybenko, G. (1989). Dynamic load balancing for distributed memory multiprocessors. *Journal of parallel and distributed computing*, *7*(2), 279-301.

Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994, June). A niched Pareto genetic algorithm for multiobjective optimization. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on* (pp. 82-87). Ieee.

Houck, C. R., Joines, J., & Kay, M. G. (1995). A genetic algorithm for function optimization: a Matlab implementation. *NCSU-IE TR*, *95*(09).

Jones, G., Willett, P., Glen, R. C., Leach, A. R., & Taylor, R. (1997). Development and validation of a genetic algorithm for flexible docking. *Journal of molecular biology*, *267*(3), 727-748.

Lombardi, F., & Di Pietro, R. (2011). Secure virtualization for cloud computing.*Journal of Network and Computer Applications*, *34*(4), 1113-1122.

Mondal, B., Dasgupta, K., & Dutta, P. (2012). Load balancing in cloud computing using stochastic hill climbing-a soft computing approach. *Procedia Technology*, *4*, 783-789.

Mühlenbein, H., Schomisch, M., & Born, J. (1991). The parallel genetic algorithm as function optimizer. *Parallel computing*, *17*(6), 619-632.

Papadimitriou, C. H. (1977). The Euclidean travelling salesman problem is NP-complete. *Theoretical Computer Science*, *4*(3), 237-244

Park, J. S., & Gavish, B. (2013). LP-Based Heuristic Algorithms for Interconnecting Token Rings via Source Routing Bridges. Heuristic Approaches for Telecommunications Network Management, Planning and Expansion: A Special Issue of the Journal of Heuristics, 6, 149-166.

Sureshbabu, D. G., & Srivatsa, D. S. (2014), A Review of Load Balancing Algorithms for Cloud Computing. *International Journal of Engineering And Computer Science ISSN*, 2319-7242.

Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and computer-integrated manufacturing*, *28*(1), 75-86.

Xu, Y., Wu, L., Guo, L., Chen, Z., Yang, L., & Shi, Z. (2011, August). An Intelligent Load Balancing Algorithm Towards Efficient Cloud Computing. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*.