NEURO-FUZZY SYSTEM BASED DYNAMIC RESOURCE ALLOCATION IN COLLABORATIVE CLOUD COMPUTING USING MULTI ATTRIBUTE QOS

Anitha [1] and Anirban Basu[2]

[1]Research Scholar, VTU, EPCET, Bangalore
[2]Professor, Department of CSE, APS College of Engg, Bangalore

ABSTRACT

Cloud collaboration is an emerging technology which enables sharing of computer files using cloud computing. Here the cloud resources are assembled and cloud services are provided using these resources. Cloud collaboration technologies are allowing users to share documents. Resource allocation in the cloud is challenging because resources offer different Quality of Service (QoS) and services running on these resources are risky for user demands. We propose a solution for resource allocation based on multi attribute QoS Scoring considering parameters such as distance to the resource from user site, reputation of the resource, task completion time, task completion ratio, and load at the resource. The proposed algorithm referred to as Multi Attribute QoS scoring (MAQS) uses Neuro Fuzzy system. We have also included a speculative manager to handle fault tolerance. In this paper it is shown that the proposed algorithm perform better than others including power trust reputation based algorithms and harmony method which use single attribute to compute the reputation score of each resource allocated.

KEYWORDS

Collaborative Cloud, Neural Network, Fuzzy system, Multi attribute, reputation

1. INTRODUCTION

In the current IT world, Cloud computing is gaining momentum, and cloud providers provide customer with the IT infrastructure in a virtual manner which can be accessed using the internet. High amount of data is exchanged between the cloud customers all over variety of computing resources in high bandwidth along with data storage at the secondary level. Due to this; there is a huge requirement for scalable resources among various cloud customers. But the problem with the present single cloud computing will not be able to manage the connecting and detecting activities for an application while it is running. Hence there is a need for virtual lab environment to be built for the researchers so that they can connect multiple cloud servers, thus Collaborative Cloud Computing (CCC) is one such proposal which is led by the advance research.

In the CCC, the resource are collectively distributed in a cooperative manner so that the different organization and also different desktop types are interconnected into a virtual organization using the CCC, so that if the resources provided in a single cloud server is not sufficient then the cloud provider will switch to a different cloud application.

DOI : 10.5121/ijccsa.2016.6102

CCC comprises of a millions of cloud resources which are sourced from different parts in a distributed format. Thus the environment will use resource management (resMgt) efficiently and Quality of Service (QoS) may be provided by different node types.

The Amazon's family of web services in cloud platforms [1], using an Infrastructure-as-a-Service (IaaS) model provides abstractions that are general enough to support a wide range of existing distributed computing platforms tailored to specific application scenarios. Amazon allows purchaser to rent virtual machines (EC2), storage volumes (EBS), and storage objects (S3) on-demand and expend only for resources they use. While pricing models vary for each resource, purchasers typically pay a fixed rate for both their length of use and their aggregate network and disk I/O bandwidth.

Many consider IaaS platforms a natural evolution of ongoing work on high-performance scientific and grid computing [2], which focuses predominantly on supporting large-scale execution of computationally-intensive scientific tasks. Due to the generality of IaaS platforms, applications with other models of computation have also become increasingly popular. In particular, Google's family of services, tailor abstractions for sub-tasks that are useful for efficiently storing and searching unstructured, and largely static, customer and web data.

Reputation management and the resource management are utilized for gaining the performance with the QoS and for selection of resources in trust worthy manner. There are three way of connecting the cloud server in a trustworthy manner with effective and efficiency such as: identifying the trust worthy resource, choosing the right resources and using other system for utilizing the resources fully [3].

To address the trust worthy allocation, reputation based algorithms were used which does not focus on the QoS and resource heterogeneity was neglected by assigning each node one reputation value for providing all of its resources. So we propose an algorithm based on multi attribute QoS selection and allocation of resources based on different reputation values using the combination of neural network and fuzzy logic in CCC.

2. RELATED WORK

In this paper [4] the cloud information extraction technique was supported by the Collaborative Cloud Computing. Neural Network (NN) based system was used for information retrieval which is located in the different system as these data are accessed directly. The mechanism such as the Artificial Neural Network, the output values are used here to activate the input functions; no additional effort is needed to get the information. In the paper, they propose a solution where the neural network is combined with the learning system so that the single point of failure is eliminated and most of the cloud computing issues are eliminated and hence making an effective and efficient information extraction in a cloud computing collaborative environment.

In this paper[5] the security issues are addressed by the CTrust Framework, this is done by connecting different VT (Virtualization technology) and then be able to access the resources such as the network, storage and software's. Cloud running applications root trusts are made using the Secure Hypervisor framework. One of the major issues of the cloud computing is the cloud security, even though the cloud computing is used in online auction or e-commerce commerce; it can be used in many different fields. The NIST (National Institute of Standards and Technology) says the main concern in the cloud computing is its security issues. Here the operating system and hardware are coupled using the software abstraction.

The older methods such as the reputation management and resource management method are not effective in long run and there is a lack of support for the dynamic and large environment which is needed for the CCC. The older method focus is on one of the QoS parameter which is either the efficiency or the security. Here the difference between the reputation management and the resource management is understood by a method called Harmony [6], which has helped in a proposal of the solution with a CCC platform which is a combination of the reputation and resource management.

In order to have large scale CCC, selecting trustworthy resources, choosing these resources and utilizing them to the maximum extent, must be executed in order. Many techniques have been proposed for resource and reputation management but these two issues have been discussed separately. When we combine both resource management and reputation management in CCC, it is creating high overhead [7]

There exist another method which used harmony; uses the cycloid structure [8] in which all the nodes are connected to one QoS parameter It can have a maximum of $n=d*2^{d}$ nodes, where d is dimension. Each cyclic node id consists of two indices: cyclic index and cubic index.

3. PROPOSED SOLUTION

The proposed solution in done by QoS scoring which is multi attributed .The resources are allocated with QoS score based on the attributes such as distance, reputation, task completion time along with completion ratio and load.

The neural network is trained on QoS scoring system and the resource with best score is allocated for user task.

The QoS score is calculated as explained below:

Let m1, m2, m3 ….mN be the number of machines in coordinated cloud.
Let t1, t2… tr be the number of task arriving for execution in cloud. Each of these tasks have a deadline time dt1, dt2… dtr.

Let the actual completion time of jobs be at1, at2… atr
The job of schedulers is to allocate the tasks to machine in such a way that

Task Completion (TC) =$\sum$ ¥ (ati-dti) == 0 with 0< i < r

With objective function of maximizing the TC such a way that: TC-r is close to zero.

4. THE MULTI ATTRIBUTE QOS SCORING (MAQS)

The proposed solution called MAQS algorithm involves three stages:

Stage 1: Training the neural Network
Stage 2: Data collection for availability
Stage 3: Scheduling

Training

The multi-layer feed forward neural network is trained periodically with five attributes which are distance, reputation score, task completion time, task completion ratio and load and provides price as the QoS output.

The five attributes are explained in detail below:

1. Distance is an important factor that affect the response time. If the processing center is located close the request generated site, the response time in carrying the job to site and carrying the result back to requested site is lowered. So we have considered distance as one of the parameter.

2. Reputation of the resource is based on the number of times the jobs are completed without any errors. In our solution, once the machine is allocated job and reply arrives, the user can rate whether reply is accurate. Based on this reputation is calculated. For calculation of Reputation Score (RS) we employ weighted averaging scheme based on equation 1

   $$RS = \alpha * RS + (1-\alpha) RS_{old} \text{ ------------ } (1)$$

   New RS is given as feed back by user in scale of 1 to 5 based on the accuracy of result and the new RS score is calculated using the weighted averaging scheme. We use weighted scheme because, reputation should consider the history and not instant value of the current reputation alone.

3. Task completion time is an indication of how fast the machine can run for job execution. The machine with fast CPU cycles must be preferred more than others, so we have used task completion time as one of the parameter for calculating the QOS.

4. Task completion ratio is indicator of how good the machine executed the job without downtime. Machine with low downtime must be preferred than others with high down time for job execution. So we have considered this parameter.

5. Scheduling policy must also be able to fairly share the load. Although machine with faster CPU, closely located with less down time are more preferred, if all load is scheduled to these machine alone, it will have a negative effect. So a load threshold must be there on the machine. This is ensured by including load as one parameter for calculating the QOS.

To generate training data set we have used Fuzzy Decision system. Fuzzy membership function is devised for all the 5 input parameters and 1 output parameter and rule set is defined to convert input to output.

Once the fuzzy logic system is created, we randomly generate different values for the input parameters and use the fuzzy system to get the QOS value for those input variables and the result is written to training file. Neural Network will use this training file to train the neurons. So at core we have created Neuro Fuzzy System.

Distance (D) is in range of 0 to 1000 km and the membership function for it is defined by splitting the distance to three ranges Near (N), Middle (M), and Far (F) as in figure 1

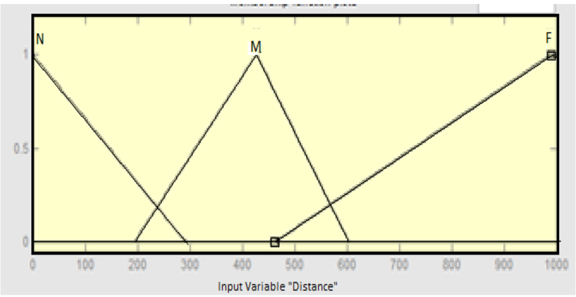

Fig 1: Range of variable Distance

Reputation (R) is in range of 0 to 10 and membership function for it is defined by splitting the reputation to three ranges Bad (B),Medium(M), good(G) as shown in figure 2

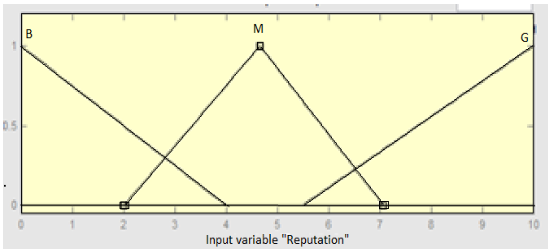

Fig 2: Range of variable Reputaion

Task completion time (T) is converted in terms of ratio to maximum completion time expected in system. It is in range of 0 to 1. The membership function for it is defined by splitting to three ranges Less (L), More (M), High (H) as in figure 3.

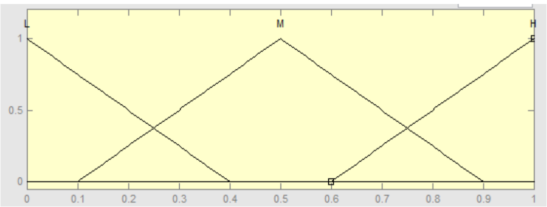

Fig 3: Range of variable Task completion Time

Task completion ratio (TR) is in range of 0 to 1. The membership function for it is defined by splitting to three ranges Less (L), Medium (M), High (H) as shown in figure 4.

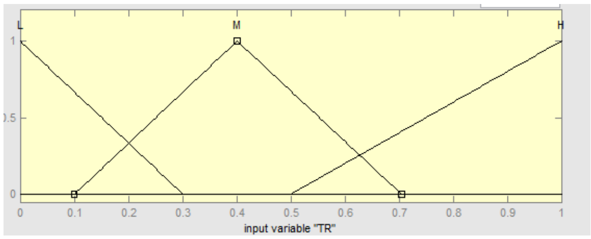

Fig 4: Range of variable Task completion ratio

Load (L) at resource is in range of 0 to 100. The membership function for it is split to three ranges Less (L), Medium (M) and High (H) as in figure 5

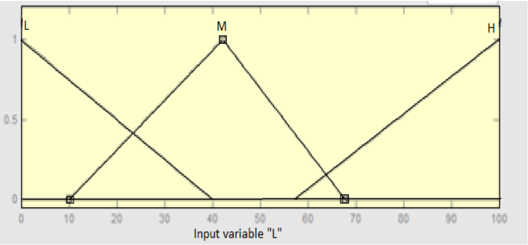

Fig 5: Range of variable Load

The output variable QOS is in range of 0 to 10 and it is split to three membership function Low(L) , Medium(M), High(H) as shown in figure 6

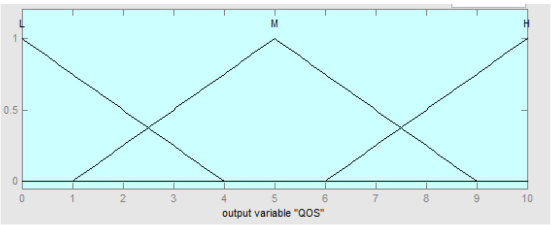

Fig 6: Output variable QoS

The rule set for fuzzy system is given below in table 1

Table 1 Fuzzy Rule set

D	R	T	TR	L	QOS
N	B	L	L	L	M
N	B	L	L	M	M
N	B	L	L	H	L
N	B	L	M	L	M
N	B	L	M	M	M
N	B	L	M	H	L
N	B	L	H	L	H
N	B	L	H	M	L
N	B	L	H	H	L
N	M	L	L	M	M
N	M	L	L	H	L
N	M	L	M	L	M
N	M	L	M	M	M
N	M	L	M	H	L
N	M	L	H	L	H
N	M	L	H	M	L
N	M	L	H	H	L
N	M	M	L	L	M
N	M	M	L	M	M
N	M	M	L	H	L
N	M	M	M	L	M
N	M	M	M	M	M

N	M	M	M	H	L
M	B	M	L	M	M
M	B	M	L	H	L
M	B	M	M	L	M
M	B	M	M	M	M
M	B	M	M	H	L
M	B	M	H	L	H
M	B	M	H	M	L
M	B	M	H	H	L
M	B	H	L	L	M
M	B	H	L	M	M
M	B	H	L	H	L
M	B	H	M	L	M
M	B	H	M	M	M
M	B	H	M	H	L
M	B	H	H	L	H
M	B	H	H	M	L
M	B	H	H	H	L
M	M	L	L	L	M
M	M	L	L	M	M
M	M	L	L	H	L
M	M	L	M	L	M
M	M	L	M	M	M
M	M	L	M	H	L
M	M	L	H	L	H
M	M	L	H	M	L
M	M	L	H	H	L
M	M	M	L	L	M
M	M	M	L	M	M
M	M	M	L	H	L
M	M	M	M	L	M
M	M	M	M	M	M
M	M	M	M	H	L
H	B	L	L	L	M
H	B	L	L	M	M
H	B	L	L	H	L
H	B	L	M	L	M
H	B	L	M	M	M
H	B	L	M	H	L
H	B	L	H	L	H
H	B	L	H	M	L
H	B	L	H	H	L
H	B	M	L	L	M
H	B	M	L	M	M
H	M	M	L	M	M
H	M	M	L	H	L
H	M	M	M	L	M
H	M	M	M	M	M
H	M	M	M	H	L
H	M	M	H	L	H

Data Collection

The data is collected frequently from all the nodes, which are separated over the world in this phase. Nodes which are participated in collaborative cloud (CC) have dissect installed and this dissect collects the census and reports to central manager about this. It maintains the periodic heartbeat with the nodes in collaborative cloud to know the availability.

Scheduling

User tasks are allocated with resources based on the QOS.

Whenever user task arrives we calculate the QOS score for the node by providing the data collection from node to neural network. The best QOS score node is selected and the task is allocated.

The scheduling algorithm flow is given below

```
Input:  the job to schedule
Output:  the machine to allocate

For i=1 to all resources
    Load (i)   //get load at resource
End

For i=1 to all resources
  Taskcompltime (i)   // get average task completion time
End

For i=1 to all resources
   Taskcomplratio (i)    // get task complete ratio
End

For i=1 to all resources
   rep (i)     //   get reputation score
End

For i=1 to all resources
  QOS(i)=get_neural_score(distance to resource,  rep(i), taskcompltime(i),
                                        taskcomplratio(i),load (i))

End

Sort resources on QOS in descending order;

Res = machine (QOS (1));    // the machine allocated

Return Res.
```

5. Mathematical Model Of The System

We design a 3 layer feed forward neural network for QOS scoring as shown in figure 7

Number of input neurons = 5
Number of hidden neurons = 11
Number of output neurons = 1

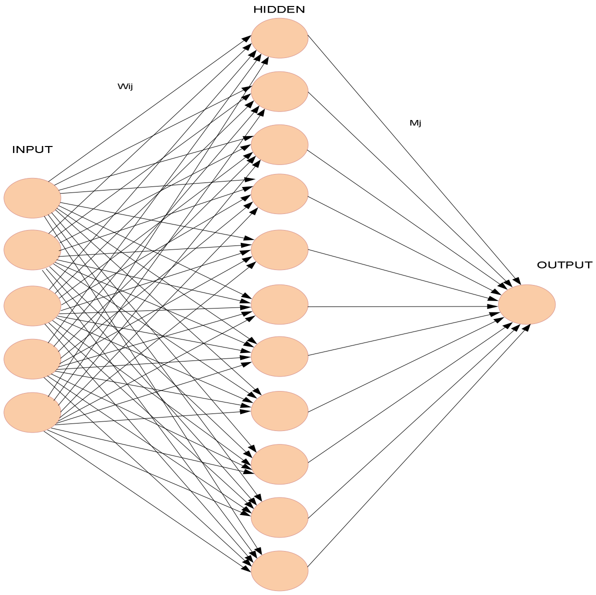

Fig 7: Feed Forward Neural Network

Each hidden layer neuron can be modelled as in figure 8

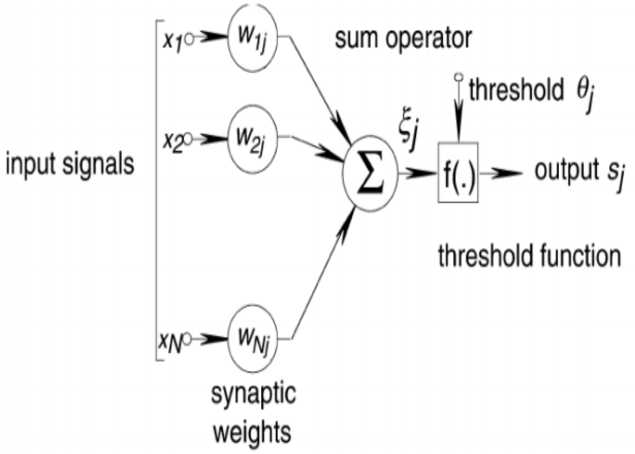

Fig 8: Modelling of hidden layer

With N being 5 in the above case.

For thresholding, we use sigmoid function as shown in Figure 9

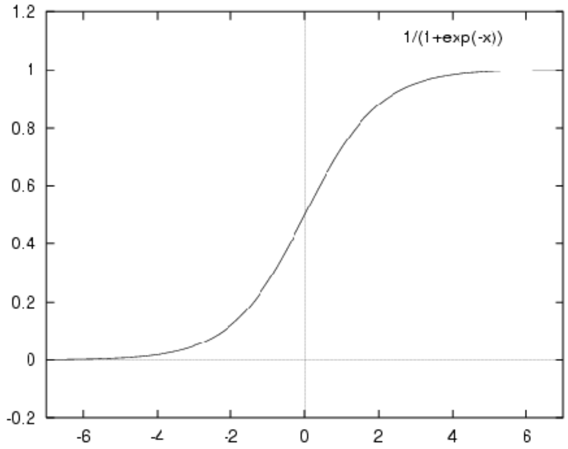

Fig 9: Sigmoid function for threshold

Let X_i be the input from the i the neuron

Let W_{ij} be the weight of link between input neuron i and the hidden neuron j.

The input to each hidden neuron j is given as in equation 2

In_j = $\sum_{i=0}^{N} Xi * Wij$ ------------------ (2)
The output from each hidden neuron is as in equation 3

Out_j = Sigmoid ($\sum_{i=0}^{N} Xi * Wij$) ----------- (3)

Let Mj be the weight of the link from the j th hidden neuron to the single output neuron.

The input to the hidden neuron is modeled as in equation 4

Hj = $\sum_{j=0}^{2*N+1}$ Mj * $sigmoid(\sum_{i=0}^{N} Xi * Wij)$------ (4)

The output from the output neuron is modeled in equation 5

O =sigmoid ($\sum_{j=0}^{2*N+1}$ Mj * $sigmoid(\sum_{i=0}^{N} Xi * Wij)$) ---(5)

Assume that, we have Z number of input request arriving to cloud broker and there are M machines.

Each machine is of varied processing capability and distributed at different distances from the user site.

For each scheduling decision we calculate the QOS for each of M machines as {Qos1, Qos2, Qos3… QosM}

We choose the J machine such that

QosJ > ¥ Qos k   with J<k<M, k≠J

Once the scheduling is complete, we calculate the average completion time(TC) of each task as in equation 6

$$\text{Avg TC} = \frac{\sum_{j-0}^{Z} \text{Tcj}}{Z} \quad \text{--------------} \ (6)$$

We calculate the Job Success Ratio (JSR) as in equation 7

$$\text{JSR} = \frac{\text{No of jobs completed in expected time}}{Z} \quad \text{---------} \ (7)$$

6. IMPLEMENTATION

The proposed modular architecture is as in figure 10 explained below:

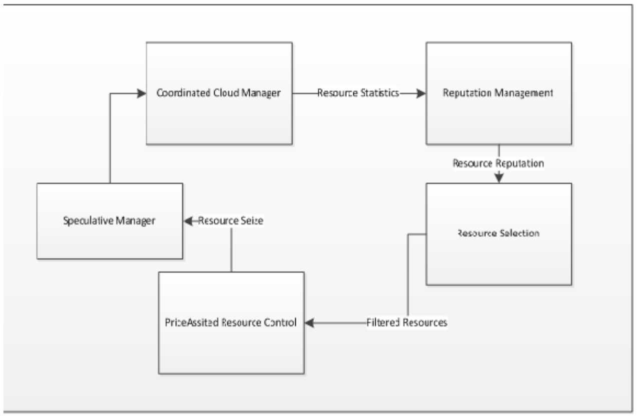

Fig 10: Proposed Architecture

Coordinated Cloud Manager: The Module helps in registering of the Coordinated Cloud which is distributed geographically. This module also acquires the heartbeats of the Coordinated Cloud.

Reputation Manager: This Modules helps in collecting census report from the Coordinated Cloud and it calculates the scoring.

Resource Selection: The resources based on the good reputation score are selected in this module for task and also select QOS for each task.

Price Assisted Resource Control: There may be an overload at some nodes because the node's resource with good score is always selected. This module helps in selecting the next less overloaded resource based on price.

Speculative Manager: This module will take care of speculative replication to provide fault tolerance.

7. PERFORMANCE ANALYSIS

We used Google cluster dataset for modelling the tasks and to test the performance of the proposed solution. A Google cluster is a set of machines, packed into racks, and connected by a high-bandwidth cluster network. A cell is a set of machines, typically all in a single cluster that shares a common cluster-management system that allocates work to machines. Work arrives at a cell in the form of jobs. A job is comprised of one or more tasks, each of which is accompanied by a set of resource requirements used for scheduling (packing) the tasks onto machines. Each task represents a Linux program, possibly consisting of multiple processes, to be run on a single machine. Tasks and jobs are scheduled onto machines according to the lifecycle described below: Resource requirements and usage data for tasks are derived from information provided by the cell's management system and the individual machines in the cell. A single usage trace typically describes several days of the workload on one of these compute cells. A trace is made up of several datasets. A dataset contains a single table, indexed by a primary key that typically includes a timestamp. Each dataset is packaged as a set of one or more files, each provided in a compressed CSV format.

We generated the tasks for cloudsim from the task table of Google cluster data.

The task resource usage table contains the following fields as in table 1

Table 1: The task resource usage table

Field No	Parameter	Field No	Parameter
1	start and end time	6	assigned memory
2	job ID	7	page cache memory usage
3	task index	8	cycles per instruction (CPI)
4	machine ID	9	memory accesses per instruction (MAI)
5	memory usage	10	sampling rate

Using these fields, we create the task list for testing the performance of our solution.

The average success rate and total waiting time is shown in figure 11 and figure 12 respectively. From this we can observe that success rate is high and total waiting time is reduced as compared to existing system.

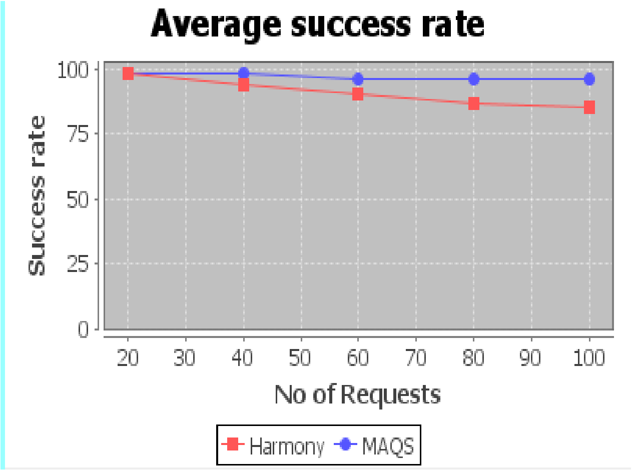

Fig 11: success rate graph

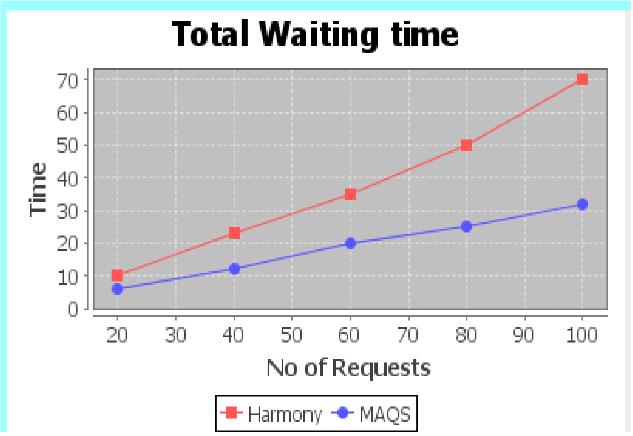

Fig 12: waiting time graph

The node utilization for the varied number of request rate is measured and from this we see that utility is high in case of proposed MAQS as in figure 13

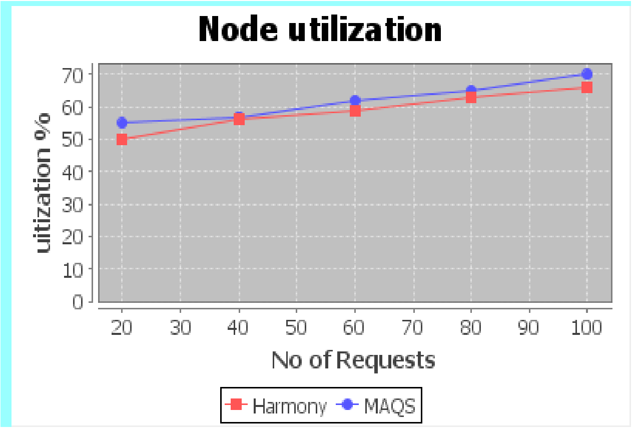

Fig 13: Node utilization Graph

8. Conclusion And Enhancements

The previous implementation focused on resource allocation considering one reputation value ignoring the heterogeneity of resources. Hence the proposed solution for resource allocation is based on multiple attributes to compute the reputation score focusing on QoS parameter. This mechanism has better success ratio and completion ratio of the task. In future, we plan to deduce a mathematical formula to calculate the optimal time period to train the feed forward neural network to improve on QoS parameters considering more than one parameter.

References

[1] Amazon Elastic Compute Cloud (EC2)

[2] Younge et al.," Analysis of Virtualization Technologies for High Performance Computing environments", IEEE International Conference on Cloud Computing, July 2011

[3] Bhaskar and Aravind Gosh, "Literature Survey on Collaborative Cloud Computing for Sharing Resource in Trustworthy Manner", IJIRCCE, volume 3, March 2015.

[4] B. Hema," An Efficient Information Retrieval Approach for Collaborative Cloud Computing", ICMACE-2014.

[5] Satyajeet Nimgaonka, Srujan Kotikela and Mahadevan Gomathisankaran, "CTrust: A Framework for Secure and Trustworthy Application Execution in Cloud Computing", (ISBN 978 – 1 – 62561 – 001 - 0...)

[6] Haying shen and Guoxin Liu, "An Efficient and Trustworthy resource sharing platform for collaborative cloud computing", IEEE Transactions on Parallel and Distributed Systems, Volume 25, No. 4,April 2014

[7] C. Liu, B.T. Loo, and Y. Mao, "Declarative Automated Cloud Resource Orchestration," Proc. Second ACM Symposium on Cloud Computing (SOCC '11), 2011.

[8] H. Shen and G. Liu, "Harmony: Integrated Resource and Reputation Management for Large- Scale Distributed Systems," Proc. 20thInt'l Conf. Computer Comm. and Networks (ICCCN), 2011.