A LOAD BALANCING MODEL IN PUBLIC CLOUD USING ANFIS AND GSO

Suresh M.
PG Scholar
SNS College of Technology,
Tamil Nadu, India
kksureshm@gmail.com

Santhosh Kumar B.
Assistant Professor
SNS College of Technology,
Tamil Nadu, India
b.santhoshkumar@gmail.com

Dr.S.Karthik
Professor and Dean
SNS College of Technology,
Tamil Nadu, India
profskarthik@gmail.com

Abstract

Recent days, cloud computing is an improving area in research and industry, which includes virtualization, distributed computing, internet, and web services. A cloud contains elements such as data centers, clients, distributed servers, internet which includes fault tolerance, high accessibility, efficiency, scalability, flexibility, reduced overhead for users, less cost of ownership, on demand services and etc. Cloud computing services are becoming omnipresent and serve as the primary source of calculating power for different applications like activity and personal computing applications. It has many benefits all along with some fundamental problems to be resolved in order to improve reliability of cloud environment. Also that, these problems is associated with load management, tolerance and different security issues in cloud environment. In this paper introduces a better load balancing model for the public cloud based on the cloud partitioning concept with a switch mechanism to select different strategies for different situations. Adaptive neuro-fuzzy inference system (ANFIS) based load balancing algorithm and Glowworm swarm optimization (GSO) based load balancing algorithm are proposed to the load balancing strategy to improve the efficiency in the public cloud environment. Experimental result shows that the proposed method gives better results when compared with other traditional methods.

Keywords—Load balancing; Neuro-fuzzy logic; Cloud partition; Glowworm Swarm optimization algorithm.

I. INTRODUCTION

Cloud computing is a new pattern of large-scale distributed computing. It has stimulated computing and data away from desktop and manageable PCs, into large data centers [1]. It has the ability to connect the power of Internet and wide area network (WAN) to make use of resources that are available remotely, thereby providing cost-effective solution to most of the real life requirements [2]. It gives the scalable IT resources such as applications and services, in addition to the infrastructure on which they control, over the Internet, on pay-per-use basis to regulate the capacity rapidly and easily. It helps to occupy changes in demand and helps any organization in stay away from the capital costs of software and hardware [3] [4]. Therefore, cloud computing is a structure for enabling an appropriate, on-demand network access to a common pool of computing resources.Cloud service is divided into three models. They are,as shown in Fig. 1

Cloud Software as a service (Saas): The competence provided to the consumer is to make use of the provider's applications consecutively running on a cloud communications. The applications are easy to get from several client devices throughout a thin client interface such as a web browser. The consumer does not deal with the fundamental cloud infrastructure.

Cloud Platform as a Service (Paas): The capability provided to the consumer is to arrange on the cloud communications consumer formed or obtained applications created by means of programming languages and tools sustained by the provider. The consumer does not supervise or control the fundamental cloud structure, but has control over the applications and perhaps application hosting environment configurations.

Cloud Infrastructure as a Service (Iaas): The capability provided to the consumer is to stipulation processing, storage space, networks, and other basic computing resources where the consumer is capable to deploy and run random software, which contains operating systems and applications. The consumer does not control the underlying cloud communications but has control over operating systems, deployed applications, storage and perhaps limited control of select networking components.

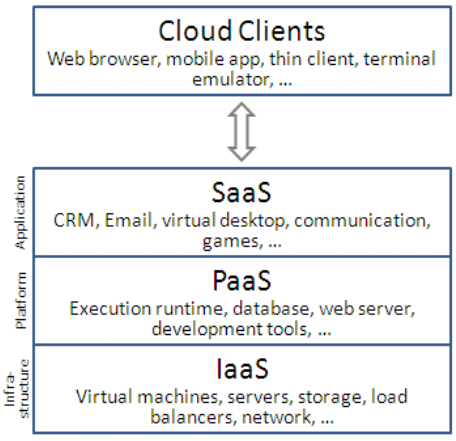

Fig. 1: Cloud Service Model

978-1-4799-3966-4/14 $31.00 © 2014 IEEE
DOI 10.1109/ICICA.2014.27

II. RELATED WORKS

Y. Zhao et al. [5] explained the problem that happens in intra-cloud load balancing with substantial hosts by adaptive survive migration of virtual machines. A load balancing model is introduced by the author to reduce virtual machines migration time by shared storage, to balance load with servers based on their processor or IO usage, etc. and to remain virtual machines' zero-downtime in the process. This algorithm guarantees that the migration of VMs is at all times from high-cost physical hosts to low-cost host but considering that each physical host has sufficient memory which is a weak assumption.

A. Bhadani et al. [6] presented a Central Load Balancing Policy for Virtual Machines (CLBVM) that balances the load equally in a distributed virtual machine/cloud computing environment. This strategy enhances the performance of the system but not considering fault-tolerant of the system.

Z. Zhang et al. [7] discussed about a load balancing system based on ant colony and complex network theory (ACCLB) in an open cloud computing group. It uses small-world and scale-free features of a composite network to attain improved load balancing.

X. Liu et al. [8] introduced a lock-free multiprocessing load balancing solution that keep away the use of shared memory in difference to further multiprocessing load balancing solutions which is used to share memory and lock to maintain a user session. It is achieved by modifying Linux kernel.

J. Hu et al. [9] discussed about a scheduling approach on load balancing of VM resources that make use of historical data and present state of the system. This approach attains the better load balancing and condensed dynamic migration by means of a genetic algorithm. It helps in determine the problem of load-imbalance and high cost of migration therefore achieves better resource utilization.

V. Nae et al. [10] presented an eventdriven load balancing algorithm for real-time Massively Multiplayer Online Games (MMOG). This algorithm once receiving capacity events as input, analyzes its components in context of the resources and the global state of the game session, thus producing the game session load balancing actions. It is accomplished of scaling up and down a game session on multiple resources according to the variable user load but has occasional QoS breaches.

III. CLOUD PARTITIONING FOR THE PUBLIC CLOUD

There are numerous cloud computing approaches with this work may focus on a public cloud. A public cloud is based on the typical cloud computing model, through the service provided by a service provider [11]. A large public

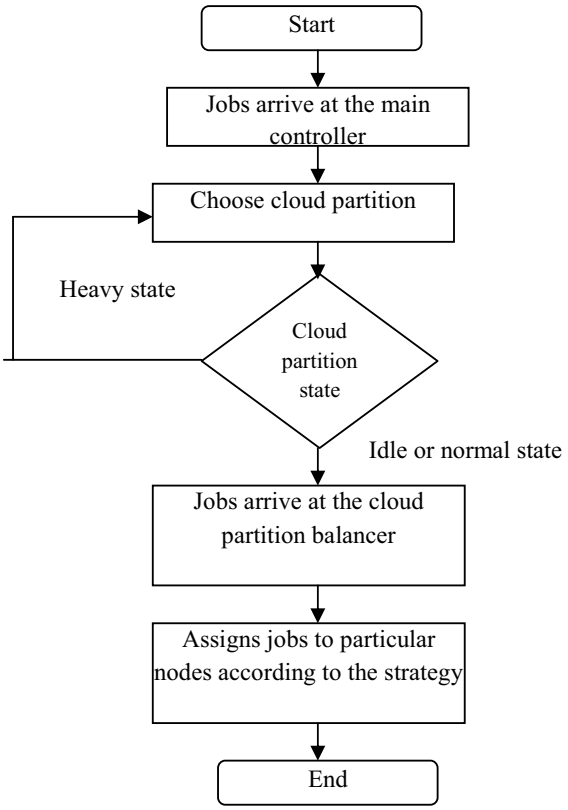

Fig. 2 Job assignment strategy

cloud will comprises of several nodes and the nodes are in different physical locations. It is used to manage this large cloud.

3.1 Job assignment strategy

Algorithm for assigning the jobs to cloud partition as shown in Fig. 2

Step 1: jobs arrive at the main controller

Step 2: choosing the cloud partition

Step 3: if cloud partition state is idle or normal state then

Step 4: jobs arrive at the cloud partition balancer.

Step 5: assigning the jobs to particular nodes based on the strategy.

Step 6: process ends.

IV. LOAD BALANCING IN CLOUD PARTITION BASED ON DIFFERENT STATES

In cloud, Load Balancing is a technique to allocate workload over one or more servers, network boundary, hard drives, or other total resources. Representative datacenter implementations depends on massive, significant computing hardware and network communications, which are subject to the common risks linked with any physical device, including

hardware failure, power interruptions and resource limits in times of high demand.

High-quality of load balance will increase the performance of the entire cloud. Though, there is no general method that can adapt to all possible different circumstances. Various methods have been developed in improving existing solutions to resolve new problems. Each particular method has advantage in a particular area but not in all situations. Therefore, the current model integrates several methods and switches between the load balance methods based on the system status.

4.1 Load balance approach for the idle status

When the cloud partition is idle, several computing resources are presented and comparatively few jobs are receiving. In these circumstances, this cloud partition has the capability to process jobs as fast as possible so an effortless load balancing method can be used.

4.1.1 Adaptive neuro-fuzzy inference system (ANFIS)

Zadeh [12] proposed a fuzzy set theory in which the set boundaries were not precisely defined, but in fact boundaries were gradational. Such a set is characterized by continuum of grades of membership (characteristic) function which assigns to each object a grade of membership ranging between zero and one [12]. These days, techniques in artificial neural networks, fuzzy set and fuzzy system have been combined together which is termed as soft computing or intelligence technique.

The ANFIS is a fuzzy Sugeno model lay in the structure of adaptive systems to make simple learning and adaptation [13]. This structure provides the ANFIS modeling more resourceful and less dependent on proficient knowledge. The ANFIS structural design is presented by two fuzzy if–then rules based on a first order Sugeno model are calculated as

Rule 1: If (x is A_1) and (y is B_1) then (f1 = $p_1x + q_1y + r_1$).

Rule 2: If (x is A_2) and (y is B_2) after that (f2 = $p_2x + q_2y + r_2$),

where x and y are the inputs of the ANFIS model. A_i and B_i are denoted as fuzzy sets, f_i are the outputs through the fuzzy part defined by the fuzzy rule base, pi; qi and ri are the design parameters predicted during the training process. Fig. 3 shows the ANFIS architecture with in the form of two rules in which a circle point out a fixed node, while a square indicates an adaptive node.

The nodes in the first layer are the adaptive nodes and the outputs produced are the fuzzy membership grade which are given by following equation (1) and (2)

$$O_i^1 = \mu_{Ai}(x), \qquad i = 1,2, \tag{1}$$

$$O_i^1 = \mu_{Bi-2}(y), \qquad i = 3,4, \tag{2}$$

where $\mu_{Ai}(x)$, $\mu_{Bi-2}(y)$ can adopt any fuzzy membership function. For instance, if the bell shaped membership function is employed, $\mu_{Ai}(x)$ is given by equation (3)

$$\mu_{Ai}(x) = \frac{1}{1 + \{\left(\frac{x - c_i}{a_i}\right)^2\}^{b_i}} \tag{3}$$

where a_i, b_i and c_i are the parameters of the membership function, governing the bell-shaped functions, as a result. The nodes are fixed a node which is to be presented in a second layer. They are labeled with M, representing that they carry out as an easy multiplier. The outputs of this layer can be correspond to as equation (4)

$$O_i^2 = w_i = \mu_{Ai}(x)\mu_{Bi}(y), \qquad i = 1,2, \tag{4}$$

The nodes in the third layer are also fixed nodes and are denoted with N, representing their normalization position to the firing strengths from the preceding layer.The outputs of this layer can be correspond to as equation (5)

$$O_i^3 = \overline{w}_i = \frac{w_i}{w_1 + w_2}, \qquad i = 1,2, \tag{5}$$

which are the so-called normalized ring strengths. The nodes in the fourth layer are considered as the adaptive nodes. Each node forms an output based on the product of the normalized firing strength and a first-order polynomial. Thus, the outputs of this layer are given by equation (6)

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i(p_i x + q_i y + r_i), \qquad i = 1,2 \tag{6}$$

There is a single fixed node in the fifth layer indicated with S. This node carries out summation of all incoming signals. Therefore, the overall output of the framework is given by equation (7)

$$O_i^5 = \sum_{i=1}^{2} \overline{w}_i f_i = \frac{(\sum_{i=1}^{2} w_i f_i)}{w_1 + w_2}, \quad i = 1,2 \tag{7}$$

It can be experiential that there are two adaptive layers in this ANFIS structural design, that is the first layer and the fourth layer. In the first layer, there are three changeable parameters $\{a_i, b_i, c_i\}$ which are connected to the input membership functions are called as premise parameters. . In the fourth layer, there are also three modifiable parameters$\{p_i, q_i, r_i\}$, pertaining to the first order polynomial. And these parameters are known as consequent parameters [14,15].Then, the fuzzifier performs the fuzzification process that converts tw types of input data

The task of the learning algorithm for this architecture is to tune all the modifiable parameters, namely $\{a_i, b_i, c_i\}$ and $\{p_i, q_i, r_i\}$, to make the ANFIS output match the like balanced load which are needed in the inference system. When the premise parameters a_i, b_i and c_i of the membership function are fixed, the output of the ANFIS model can be written as:

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \tag{8}$$

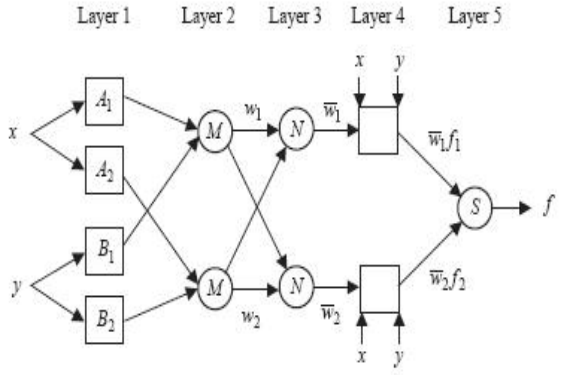

Fig. 3 ANFIS Architecture

Then now the equation (5) is applied to equation (8) changed as

$$f = \overline{w}_1 f_1 + \overline{w}_2 f_2 \quad (9)$$

Substitution of fuzzy if then rules above equation (9) it changed as

$$f = \overline{w}_1(p_1 x + q_1 y + r_1) + \overline{w}_2(p_2 x + q_2 y + r_2 \quad (10)$$

which is a linear combination of the modifiable consequent parameters p_1, q_1, r_1, p_2, q_2 and r_2.

The algorithm is illustrated to maintain the load in VM of cloud

```
Begin
    Connect_to_resource ()
    L1
    If (resource found)
   Begin
      Select neuro – fuzzy_connection ()
    Return resource to requester
   End
  Else
   Begin
    If (Anymore resource available)
     Choose_next_resource ()
    Go to L1
   Else
     Exit
   End
End
```

Computing as follows

4.2 Load balancing approach for the normal status

When the cloud partition is normal, tasks are arriving much faster than the idle state and the situation is far more difficult, thus a different strategy is used for the load balancing. Every user needs his jobs done in the shortest time, as a result the public cloud needs a method that can complete the jobs of all users with reasonable response time.

4.2.1 Glowworm Swarm Optimization Algorithm (GSO)

In GSO, each glowworm distributes in the objective function definition space [16]. These glowworms carry own luciferin values and have the respective field of idea scope called local-decision range. As the glow seeks for the neighbor set in the local-decision range, in the set, a brighter glow has a higher attraction to attract this glow toward this traverse, and the flight direction each time different will change along with the choice neighbor. Moreover, the local-decision range size will be influenced by the neighbor quantity, when the neighbor density will be low, glow's policy-making radius will enlarge favors seeks for more neighbors.

Each glowworm i encodes the object function value $J(x_i(t))$ at its current location $x_i(t)$ into a luciferin value l_i and broadcasts the same within its neighbourhood. The set of neighbours ($N_i(t)$) of glowworm i consists of those glowworms that have a relatively higher luciferin value and that are located within a dynamic decision domain and updating by formula (1) at each iteration.
Local-decision range update is given by equation (11):

$$r_d^i(t+1) = \min\left\{r_s, max\{0, r_d^i(t) + \beta(n_t - |N_i(t)|\}\right\} (11)$$

and $r_d^i\ (t+1)$ is the glowworm i *'s* local-decision range at the $t+1$ iteration, r_s is the sensor range, n_t is the neighbourhood threshold, β which affects the rate of change of the nerghbourhood range. The number of glow in local-decision range is given by equation (12):

$$N_i(t) = \{j: \|x_j(t) - x_i(t)\| < r_d^i; l_i(t) < l_j(t)\}; \quad (12)$$

and, $x_j(t)$is the glowworm i *'s* position at the t iteration, (t) is the glowworm i *'s* luciferin at the t iteration.; the set of neighbours of glowworm i consists of those glowworms that have a relatively higher luciferin value and that are located within a dynamic decision domain whose range r_d^i is bounded above by a circular sensor range $r_s\left(0 < r_d^i < r_s\right)$. Each glowworm as given in equation (13) i selects a neighbour j with a probability $p_{ij}(t)$ and process toward it as .

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)} \quad (13)$$

Movement update is given in equation (14):

$$x_i(t+1) = x_i(t) + s\left(\frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|}\right); \quad (14)$$

Luciferin-update is given in equation (15):

$$l_i(t) = (1-\rho) l_i(t-1) + \gamma J\big(x_i(t)\big); \quad (15)$$

and $l_i(t)$ is a luciferin value of glowworm i at the t iteration, $\rho \in (0,1)$ lead to the reflection of the cumulative kindness of the path followed by the glowworms in their current luciferin values, the parameter γ only scales the function fitness values, $J(x_i(t))$ is the value of test function.

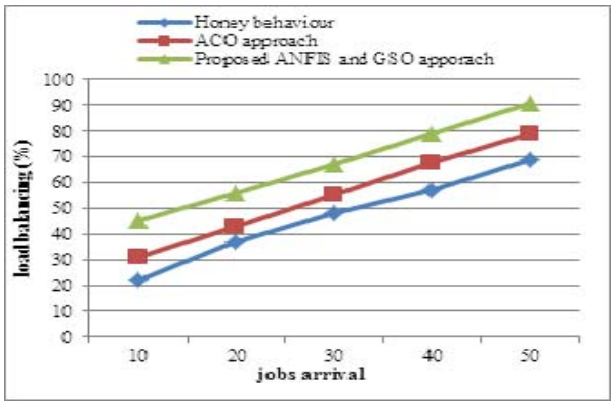

Fig. 4 Load balancing in the cloud partition

V. EXPERIMENTAL SETUP AND PARAMETER

In cloud simulator, the cloudsim, that facilitates modeling and simulation of Cloud computing systems and application surroundings. It supports both system and behavior modeling of Cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies. Using the load balancing parameters of virtual machine and data center, we have examined on cloudsim-3.0 a cloud computing simulator and evaluate the performance of proposed work using job assigning to the cloud partition. Fig. 4 represents the graphical representation of the load balancing in each cloud partition and the proposed approaches such as ANFIS based load balancing algorithm and GSO based load balancing algorithm are shows the better results than the existing techniques.

VI. CONCLUSION

Load balancing is one of the main challenges in cloud computing. It is required to distribute the dynamic local workload evenly across all the nodes to achieve a high user satisfaction and re-source utilization ratio by making sure that every computing re-source is distributed efficiently and fairly. Existing load balancing techniques have many disadvantages, therefore an efficient load balancing technique that can improve the performance of cloud computing by balancing the workload across all the nodes in the cloud along with maximum resource utilization is required. In this paper, proposed the new efficient load balancing technique using ANFIS based load balancing technique and GSO based load balancing algorithm are used to obtain measurable improvements in resource utilization and availability of cloud-computing environment.

REFERENCE

[1] Nidhi Jain Kansal, Inderveer Chana, published ," Cloud Load balancing techniques: A step towards green computing", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012.

[2] A.Bhadani, and S. Chaudhary, "Performance evaluation of web servers using central load balancing policy over virtual machines on cloud", Proceedings of the Third Annual ACM Bangalore Conference (COMPUTE), January 2010.

[3] R. W. Lucky, "Cloud computing", IEEE Journal of Spectrum, Vol. 46, No. 5, May 2009, pages 27-45.

[4] M. D. Dikaiakos, G. Pallis, D. Katsa, P. Mehra, and A. Vakali, "Cloud Computing: Distributed Internet Computing for IT and Scientific Research", IEEE Journal of Internet Computing, Vol. 13, No. 5, September/October 2009, pages 10-13.

[5] Zhao Y. and Huang W. (2009) 5th International Joint Conference on INC, IMS and IDC, 170-175.

[6] Bhadani A. and Chaudhary S. (2010) 3rd Annual ACM Bangalore Conference.

[7] H. Mehta, P. Kanungo, and M. Chandwani, "Decentralized content aware load balancing algorithm for distributed computing environments", Proceedings of the International Conference Workshop on Emerging Trends in Technology (ICWET), February 2011, pages 370-375.

[8] Xi. Liu, Lei. Pan, Chong-Jun. Wang, and Jun-Yuan. Xie, "A Lock-Free Solution for Load Balancing in Multi-Core Environment", 3rd IEEE International Workshop on Intelligent Systems and Applications (ISA), 2011, pages 1-4.

[9] J. Hu, J. Gu, G. Sun, and T. Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment", Third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), 2010, pages 89-96.

[10] V. Nae, R. Prodan, and T. Fahringer, "Cost-Efficient Hosting and Load Balancing of Massively Multiplayer Online Games", Proceedings of the 11th IEEE/ACM International Conference on Grid Computing (Grid), IEEE Computer Society, October 2010, pages 9-17.

[11] A.Rouse, Public cloud, http://searchcloudcomputing.techtarget.com/definition/public-cloud, 2012.

[12] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (1965), 338–353.

[13] J. Finol, Y.K. Guo, X.D. Jing, A rule based fuzzy model for the prediction of petrophysical rock parameters, J. Pet. Sci. Eng. 29 (2001) 97–113.

[14] W. Pedrycz, An identification algorithm in fuzzy relational systems, Fuzzy SetsSyst. 13 (1984) 153–167.

[15] W. Pedrycz and F. Gomide, "An introduction to fuzzy sets: analysis and design", complex adaptive systems. MIT Press, 1998.

[16] Jiakun LIU, Yongquan ZHOU†, Kai HUANG, Zhe OUYANG, Yingjiu WANG "A Glowworm Swarm Optimization Algorithm Based on Definite Updating Search Domains" Journal of Computational Information Systems 7: 10 (2011) 3698-3705.