✚IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

IMPLEMENTATION OF EFFICIENT ALGORITHMS FOR LOAD BALANCING MODELING WEB-BASED CLOUD APPLICATIONS

Shambhu Prasad Sah*, Sanjeev Kumar Panjiyar, Purushottam Das, Ankur Singh Bist
Assistant Professor, Deptt. of CSE Graphics Era Hill University, Bhimtal (Nainital) Uttarkhand, INDIA
M.Tech. (Computer Science) JNTU, Hyderabad, INDIA
Assistant Professor, Deptt. of CSE Graphics Era Hill University, Bhimtal (Nainital) Uttarkhand, INDIA
Assistant Professor, Deptt. of CSE KIET, Ghaziabad UP, INDIA

ABSTRACT
Load balancing is one of the central issues in cloud computing. Since millions of users are accessing the cloud every moment, the concept of load balancing has an important impact on the performance of cloud computing. This paper implements two efficient algorithms for performing 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 an improved round robin technique and game theory to the load balancing strategy to improve the efficiency in the public cloud environment.

KEYWORDS: Cloud, Load Balancing, Main Controller, Balancer, Game theory.

INTRODUCTION
Cloud computing is an attracting technology in the field of computer science. Cloud computing is an on demand service in which shared resources, information, software and other devices are provided according to the clients requirement at specific time. It's a term which is generally used in case of Internet. The whole Internet can be viewed as a cloud. Capital and operational costs can be cut using cloud computing.

The cloud is changing our life by providing users with new types of services. Users get service from a cloud without paying attention to the details. Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a sharedpool of configurable computing resources(e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. More and more people pay attention to cloud computing.

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 paper implements two efficient algorithms to achieve a better load balance model for the public cloud. It is based on the cloud partitioning concept with a switch mechanism to choose different strategies for different situations. The algorithm applies an improved round robin technique and game theory to the load balancing strategy to improve the efficiency in the public cloud environment.

RELATED WORK
There have been several load balancing methods are developed for the cloud environment. Each particular method has some advantage in some stage but not in all the situations. The different load balancing algorithms are Round Robin, Equally Spread Current Execution Algorithm, and Ant Colony algorithm.

Nishant et al.[10] used the ant colony optimization method in nodes load balancing. It is used for load distribution of workloads among nodes of a cloud environment.Randles [11] considered an algorithm called Honeybee Foraging for the load balancing in the cloud. This algorithm is derived from the behavior of honey bees for finding and reaping

food.D.Grosu [15]considered load balancing as a cooperative game in the cloud computing. S. Aote and M. U. Kharat [16] develops a game-theoretic model fordynamic load balancing in distributed systems.
An improved Round Robin technique is used here for its simplicity. It also uses a game theory approach for efficient load balancing in the cloud environment.

PROPOSED SYSTEM
The proposed system implements two efficient algorithms for balancing the loads in the cloud. It helps to improve the resource utilization and response time.

Modules
The system has three main modules which describes the functionality of the system.
- Area based cloud partitioning
A big public cloud has many nodes and the nodes are at differentphysical locations. Thus, this model divides the public cloud into several cloud partitions.
When the environment is very large and complex, the load balancing is simplified by partitioning the clouds. Thereis a main controller in cloud which chooses the suitable partitions for arriving jobs. The best load balancingstrategy helps to select the appropriate partition.All the status information is gathered and analyzed by main controller and balancers. They also perform the loadbalancing operations. The system status will be stored in a load status table and it provides a basis for choosing the right load balancing strategy.
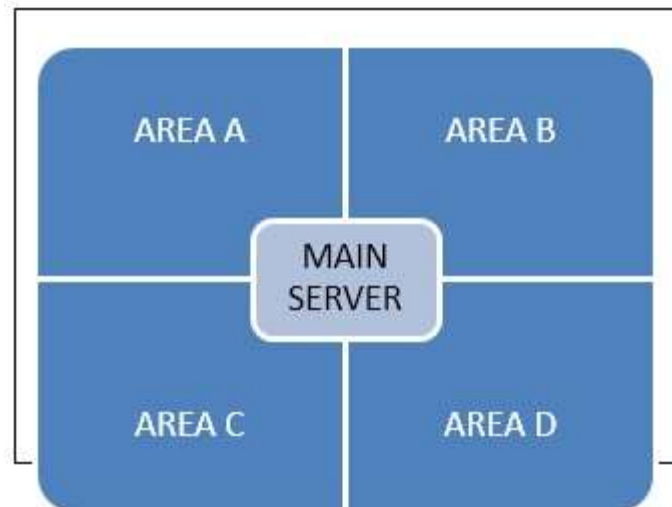

Fig. 1. Cloud Partitions

The figure shows the typical cloud partitions. In this paper the main server is divided into approximately 4 different servers, which are partitioned into small clouds called balancers (each balancer will have some servers).The cloud partitioning is performed based on the geographic locations.

When a job arrives at the main server, the first step is to choose the right partition. The cloud partition status can be divided into three types:
- Idle: When the percentage of idle nodes exceeds alpha, change to idle status.
- Normal: When the percentage of the normal nodes exceeds beta, change to normal load status.
- Overload: When the percentage of the overloaded nodes exceeds gamma, change to overloaded status.
The values of alpha, beta and gamma can be set accordingly.

Round Robin Technique
When the cloud partition is Idle, the algorithm uses round robin technique for performing load balancing in the cloud computing. In this paper it uses an improved round robin technique called as Round Robin based on Load

degree evaluation. It evaluates the system status every time which is recorded in the load status table. Based on the system status, using round robin technique load is balancing between the servers.

Game Theory Technique
When the cloud partition is Normal, the algorithm uses the game theory technique for performing load balancing in the cloud computing. It calculates the shortest distance between the servers using the distance formula. When the selected server is overloaded, and when there are no idle servers available, by use of game theory it calculates the shortest distant server and load is assigned to the server.

Functional description
The cloud computing is a dynamic environment. The availability of cloud systems is analyzed in this paper. The loadbalancing model is applied across different data centers to ensure the network availability. This paper highlights two load balancing algorithms that help to improve performance, resource utilization and availability of cloud computingenvironment. It tries to reduce the cost of cloud systems.

Two algorithms are implemented in the model described by GaochaoXu[1].The system performs an effective load balancing in the cloud environment. The total load is equally divided between the servers so there will not be any servers which are overloaded or under loaded.

Whenever a new user is accessing the application, it checks the IP address of the user in the IP lookup table to ensure that it as an authenticated IP address. Then it checks the server status, based on which request is assigned to particular server.

Algorithms
- Round Robin based on load degree evaluation algorithm:

Step 1: Let Server A is an Overloaded
Step 2: Let s[n] is an array consists of server which
are in Idle state.
Step 3: Let c=1.
Step 4: If new connection Came for Server A
Then
                Send the connection to s[c]
                After that make c=c+1;
                If c==n then c=1
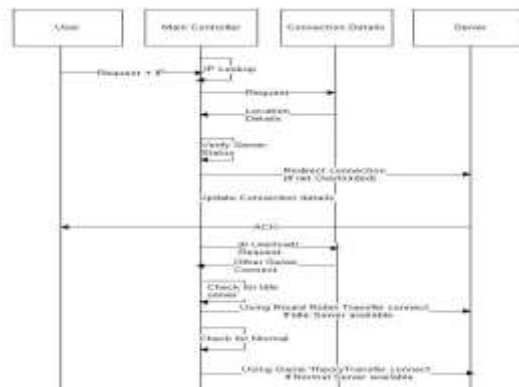   Else
                Wait;
Step 5: go to step 4.

Fig. 1. Sequence diagram of the system

The figure 2 shows the sequence diagram of the system. It describes the different operations performed in system under different conditions.

- Game Theory algorithm:

Step 1: Let Server A is an Overloaded

Step 2: Let w[n] is an array consists of server which are in Normal state. n is the total number of server.

Step 3: if new connection Came for Server A

      Then

            Calculate distance of server A and w[n]'s.

            Select minimum distance server

            Send the connection to w[n]

      Else

            Wait;

Step 4: go to step 3.

i.   Distance Formula:

$$d=\sqrt{(x_2-x_1)^2 +(y_2-y_1)^2} \qquad (1)$$

x_1 & x_2 = Latitude and longitude of Server A.

y_1 & y_2 = Latitude and longitude of w[n]

When the system status is Normal, game theory is used for assigning the load. It uses the distance formula given above to calculate the distance between the servers. In order to calculate the distance, the latitude and longitude of each of the servers has to be calculated.

System Flow

The flowchart shows below the overall operations performed in the system. It specifies the step by step procedure of the application and also the different conditions happening in the system.

Whenever a new request arrives, main controller will assign it to the appropriate partition, and then the balancer will check the load degree of each node and assign the request to the low degree node.

The figure 3 shows the system flowchart. The complete process of the system is shown in above the figure.

CONCLUSION

The overall goal of this project is to balance the load on clouds. Balancing load on the cloud will improve the performance of cloud services substantially. It will prevent overloading of servers, which would otherwise degrade the performance. The response time will also improve.

This software can be used for efficient data storage on clouds and load balancing. This software will help to dynamically allocate jobs (data) to the least loaded server. Thus overall performance of cloud services will not beaffected.

There are also provisions to accommodate future modifications in the system. Thus, this system can be efficiently used for performing load balancing in the cloud computing.
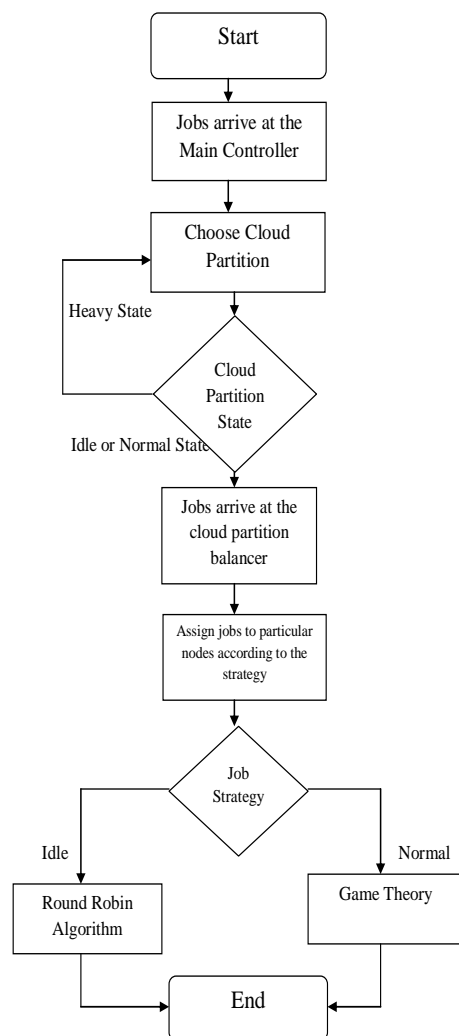
*Fig. 2. **System Flowchart***

FUTURE WORK

More work is needed to resolve new problems and to make the system more efficient.

Cloud divisioning: Partitioning the cloud is a complex problem. A detailed methodology for partitioning the cloud must be known.

Setting up a refresh period: For load balancing the main controller and balancers should have the most recentstatus information. So, the statistical information should be refreshed at fixed time intervals. The intervalshould not be too short or too long. Tests and statistical tools are needed to set the refresh interval.

A better method for load status calculation:An efficient method must be developed for calculating the system load status.A good algorithm is required to set $Load\_degree_{high}$and $Load\_degree_{low}$.

REFERENCES

[1] GaochaoXu, Junjie Pang, XiaodongFuJaya, BharathiChintalapati,Srinivasa Rao T.Y.S. " A Load Balancing Model Based on Cloud".

[2]   R. Hunter, The why of cloud, htttp://www.gartner.com/DisplayDocument?doc cd=226469&ref= g noreg, 2012.

[3]   M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis,and A. Vakali, Cloud computing: Distributed internet computing for IT and scientific research, Internet Computing, vol.13, no.5, pp.10-13, Sept.-Oct. 2009.

[4]   P. Mell and T. Grance, The NIST definition of cloud computing, http://csrc.nist.gov/ publications/nistpubs/800-145/SP800-145.pdf, 2012

[5]   Microsoft Academic Research, Cloud computing, http://libra.msra.cn/Keyword/6051/cloud-computing?query=cloud%20computing, 2012.

[6]   Google Trends, Cloud computing, http://www.google.com/trends/explore#q=cloud%20computing, 2012.

[7]   N. G. Shivaratri, P. Krueger, and M. Singhal, Load distributing for locally distributed systems, Computer,vol. 25, no. 12, pp. 33-44, Dec. 1992.

[8]   B. Adler, Load balancing in the cloud: Tools, tips and techniques, http://www.rightscale. com/info center/whitepapers/Load-Balancing-in-the-Cloud.pdf, 2012

[9]   Z. Chaczko, V. Mahadevan, S. Aslanzadeh, andC. Mcdermid, Availability and load balancing in cloud computing, presented at the 2011 International Conference on Computer and Software Modeling, Singapore, 2011.

[10] K. Nishant, P. Sharma, V. Krishna, C. Gupta, K. P. Singh,N. Nitin, and R. Rastogi, Load balancing of nodes in cloud using ant colony optimization, in Proc. 14th International Conference on Computer Modelling and Simulation (UKSim), Cambridgeshire, United Kingdom,Mar. 2012, pp. 28-30.

[11] M. Randles, D. Lamb, and A. Taleb-Bendiab, Acomparative study into distributed load balancingAlgorithms for cloud computing, in Proc. IEEE 24th International Conference on Advanced InformationNetworking and Applications, Perth, Australia, 2010,pp. 551-556.

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

[13] D. MacVittie, Intro to load balancing for developers —The algorithms, https://devcentral.f5.com/blogs/us/introto- load-balancing-for-developers-ndash-the-algorithms,2012.

[14] S. Penmatsa and A. T. Chronopoulos, Game-theoretic static load balancing for distributed systems, Journal of Parallel and Distributed Computing, vol. 71, no. 4,pp. 537-555, Apr. 2011

[15] D. Grosu, A. T. Chronopoulos, and M. Y. Leung, Load balancing in distributed systems: An approach using cooperative games, in Proc. 16th IEEE Intl. Parallel and Distributed Processing Symp., Florida, USA, Apr. 2002,pp. 52-61

[16] S. Aote and M. U. Kharat, A game-theoretic model for dynamic load balancing in distributed systems, in Proc. The International Conference on Advances in Computing,Communication and Control (ICAC3 '09), New York,USA, 2009, pp. 235-238.

[17] Partitioning for the Public Cloud" IEEE transactions on cloud computing year 2013.