IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 5, July 2014.
www.ijiset.com
ISSN 2348 – 7968

A Secure Load Balancing Technique based on Cloud Partitioning for Public Cloud Infrastructure

Nidhi Bedi[1] and Shakti Arora[1]

[1] Computer Science & Engineering Department, Kurukshetra University Krurkshetra/Geeta Engineering College, Naultha Panipat, Haryana 132107/NORTH, India

[1] Abstract

In Cloud Computing different terms like distributed computing, virtualization, networking, software and web services encompassing several elements such as clients, data enter and distributed servers are used. Load balancing is used to improve the performance of the entire cloud. In Load balancing process, load distributed among various nodes of a distributed system to improve job response time and resource utilization as well as to avoid a situation where some of the nodes are over-loaded while other nodes are idle or under-loaded. Here, game theory is used to develop a highly robust load balancing or job assignment approach for distributed cloud system. Also Nash Bargaining solution approach is used to develop this system. This work also facilitates a symmetric public key cryptographic based authentication system which provides authenticated data storage for cloud infrastructure.

Keywords: Clod Computing, Load Balancing, Cloud Partion, Game Theory, Public Cloud Infrastructre.

[2] 1. Introduction

Generally, Cloud Computing having a bunch of distributed servers also known as masters, which provide demanded resources and services to the clients with reliability and scalability in a network of datacenter. These masters provide on-demand services. Services may be of any kind like software resources (e.g. Software as a Service, SaaS) or physical resources (e.g. Platform as a Service, PaaS) or hardware/infrastructure (e.g. Hardware as a Service, HaaS or Infrastructure as a Service, IaaS). Amazon EC2 (Amazon Elastic Compute Cloud) is an example of cloud computing services [11].Cloud computing services are different from our traditional web service because there are five basic principles behind cloud computing. These principles are: resource pooling, virtualization, elasticity, automatic/easy resource deployment, metered billing. These principles make cloud computing to bring more automation, cost-savings and flexibility to the users. A Cloud system having 3 major components named as clients, datacenter, and distributed servers. Each component has a definite purpose and plays a specific role. **Clients**: Mobile, Thin, Thick clients are used by end users to interact and manage information related to the cloud. **Datacenter**:This is a collection of servers hosting various applications for end users. **Distributed Servers:** These are the parts of a cloud which are present throughout the Internet hosting different applications.

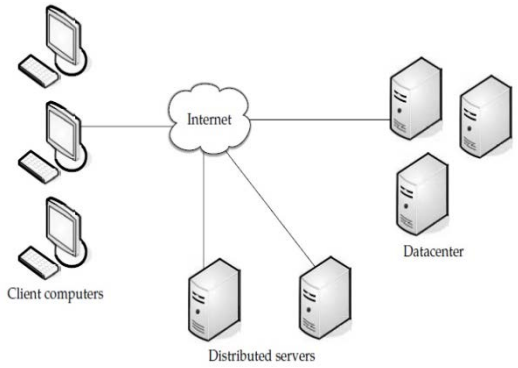

Fig 1: Three components make up a cloud computing solution [11].

Clouds can be of 3 types: Public Clouds, Private Clouds, Hybrid Clouds (combination of both private and public clouds). The word Service means various types of applications provided by different servers across the cloud. This is generally describes as “as a service”. There are three type of services a cloud can have [12]: **Software as a Service (SaaS),** In SaaS, the end users use different software applications like video conferencing, IT service management, web analytics, hosted from different servers through the Internet.

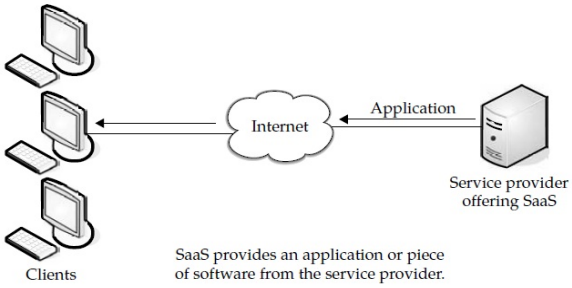

Fig 2: Software as a service (SaaS) [12]

Platform as a Service (PaaS), This provides all the resources which are required for building applications completely with the help of Internet, without installing and downloading software [12].

ISSN 2348 – 7968

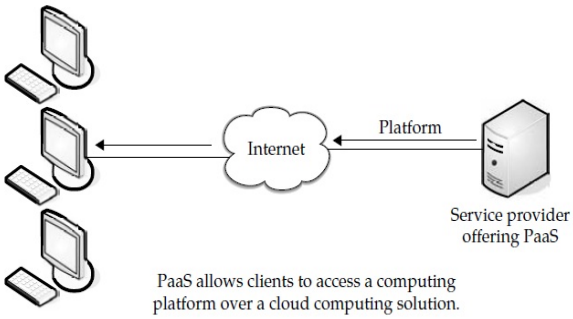

Fig 3: Platform as a service (PaaS) [12]

Hardware as a Service (HaaS), It is also known as Infrastructure as a Service (IaaS). This allows the end user to "rent" resources as Server space, Memory, Network equipment.

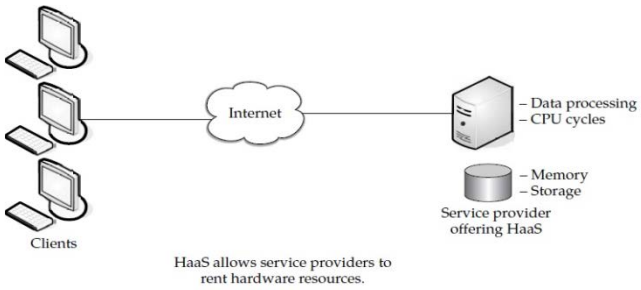

Fig 4: Hardware as a service (HaaS) [12]

Load Balancing: This process is used to improve the response time of the job. This is done simultaneously by removing a condition in which few nodes are over loaded while others are under loaded. It depends on the present behavior of the system. The main things to consider in developing such algorithm are : interaction between the nodes, comparison of load, performance of system, estimation of load, nature of work to be transferred, selecting of nodes [13]. As given in [13], the goals of load balancing are: To improve the performance substantially, there must be a backup plan in case the system fails even partially. Load balancing algorithms can be of three categories as given in [13]: **Sender Initiated:** If the load balancing algorithm is initialized by the sender, **Receiver Initiated:** If the load balancing algorithm is initiated by the receiver and **Symmetric:** It is the combination of both senders initiated and receiver initiated. Depending on the current state of the system, load balancing algorithms can be divided into **Static:** It doesn't depend on the current state of the system. Prior knowledge of the system is needed and **Dynamic:** Decisions on load balancing are based on current state of the system. No prior knowledge is needed. So it is better than static approach.

2. Related Work

Ranjan Kumar et al [1], implemented ant colony optimization method and considered the normal performance of ants and how powerfully the ants do discover their straight path along with the parallel mathematical formulation has been engaged by cloud computing system. **Suriya et al [2],** presented a variety of feature pertaining to province of cloud computing, its development, its general problem, as well as predominantly to problem connected to load balancing. This investigation couldn't lane its inspection for additional improvement and optimization for load assessment for cloud communications. **Yu-lung Lo et al [3],** supported a method for database portability as well as its load evaluation in cloud request. The equations employed for estimating the variation of database portion. This proposed system was oriented towards presented database portability. So, this cannot be considered optimum for public cloud implementation. **Meriem Meddeber et al [4],** proposed system with dual systems modes. The proposed approach is in fact distributed with certain local decision model. **Venubabu Kunamneni et al [5],** proposed a dynamic load balancing scheme for a cloud infrastructure which is used to perform job scheduling. **N. G. Shivaratri et.al [6],** emphasized on the issue of optimum and fair load or task scheduling so that performance of systems can be enhanced. Numerous factors such as issues being faced with load distributing in generic cloud infrastructures, encompassing the various motivating factors and numerous design trade-offs for load-distributing algorithms were analyzed and discussed. **Chronopoulos et. al. [7],** presented a game theoretic scheme for solving the issues associated with static load balancing for both the classes; single-class as well as multi user jobs in certain distributed system where the functional entities such as computers are joined together using certain communication media. The objective of this work was to facilitate fairness to all the tasks in case of a single-class system and the various user entities in jobs for multi-user system. **D. Grosu et.al [8],** considered Nash Bargaining Solution (NBS) that facilitates a Pareto optimal job scheduling scheme that is functional fairly with all tasks. They even developed the static load balancing issue with single class job distributed framework using a cooperative game theoretic approach amongst various configured computer systems. The Author then developed a cooperative load balancing technique of scheduling for computing NBS. Ultimately, the performance of their cooperative load balancing scheme was exhibited with various performance parameters. **K. Nishant et.al [9]** developed a scheme approach for load distribution across the cloud network of a cloud by employing Ant Colony Optimization (ACO) technique.

ISSN 2348 – 7968

3. Proposed Methodology

Since the job arrival pattern is not predictable and the workload control is crucial to improve system performance and maintain stability. A dynamic scheme is used here for its flexibility. In this paper, game theory is used to develop load balancing or job assignment approach for cloud system. The main goal of this paper is too distribute the load among nodes available across the network properly in balanced manner so that response time can be enhanced. The game theory implementation resulted into optimum execution delay. Also cooperative game theory is applied for developing dynamic load balancing strategies. This system also uses Nash Bargaining equilibrium (NBE) solution which facilitates a Pareto optimal solution in case of the cloud system. In this paper the main work is to emphasize the job allocation approach with the reduction in execution time of job scheduling. The developed system also provides a secure data storage facility for public cloud infrastructure.

In this proposed system, divided cloud represents a subarea in public cloud. It has been illustrated in following figure (Fig. 6). Once creating the cloud partitions the load balancing would initiate functioning and whenever the task reaches at the system then the deployed main controller section takes the decision that which particular cloud partition must be getting cloud access or must receive any task. The deployed load balancer then exhibits decision making that how to allocate tasks to certain definite nodes. In case the load status of the node is in normal situation, then the partitioning could be effectively accomplished on local basis.

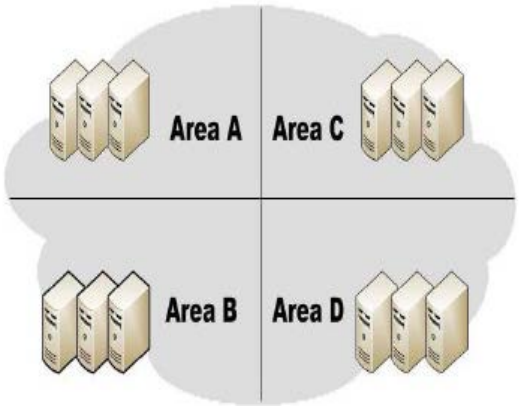

Fig. 6: typical cloud partitioning

Different load balancing strategies are used for efficient load balancing.

Load balance strategy for the idle status

When the cloud partition is idle, it means many computing resources are available and very few jobs are arriving. In this situation, this cloud partition can process jobs as quickly as possible. The Round Robin algorithm is the simplest load balancing algorithms for such type of situations, which passes each new request to the next server in the queue.

Load balancing strategy for the normal status

In normal status the cloud partition get job much faster than idle status. So a different strategy is used for the load balancing in this case. Because every user wants his jobs get completed in the shortest reasonable time. Penmatsa and Chronopoulos [7] proposed a static load balancing strategy based on game theory for distributed systems. The system then reaches the Nash equilibrium, where each decision maker makes the optimized decision.

System Modules

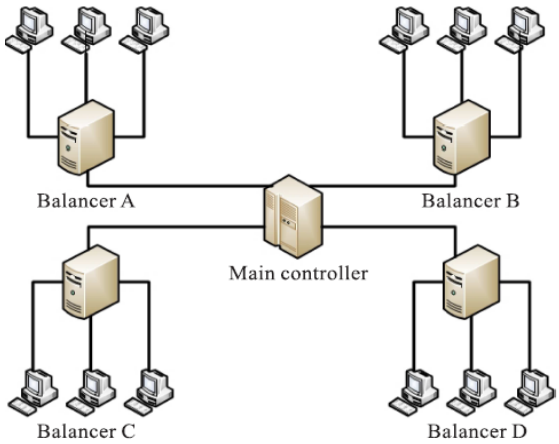

Fig. 7: Relationships between the main controllers, the balancers, and the nodes

Main controller and balancers

The main controller module initially performs assignment of tasks to the proper part of cloud infrastructure which is then followed by communication with the load balancers in every cloud partition for refreshing the information associated with the cloud. Fig. 7 illustrates the relationship developed between different cloud balancers and its associated main controller.Whenever certain job approaches to certain public cloud, the initial phase is to select the optimum cloud partition. The cloud partition status can be further classified into three kinds:**Idle, Normal, Overload.**

The entire encompassing phenomenon is as follows in fig. 8.

ISSN 2348 – 7968

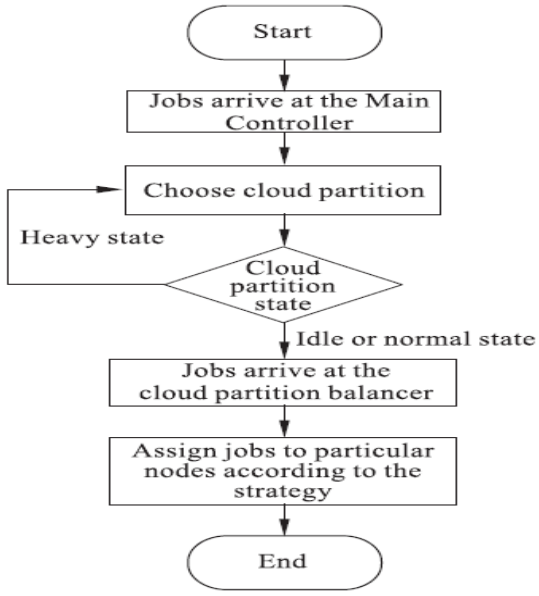

Fig. 8: Job assignment scheme

Assigning jobs to the nodes in the cloud partition

In this developed system, the balancers associated with the cloud partitions collects details from every encompassed node for estimating the cloud partition status. This estimation of individual node's load status plays significant role. The initial function is to define certain load degree for every encompassed node in the cloud.

Algorithm 1 Best Partition Searching

```
Begin
  while job do
    searchBestPartition (job);
    if partitionState == idle || partitionState == normal
then
      Send Job to Partition;
    else
      search for another Partition;
    end if
  end while
end
```

The degree of load is estimated from these below mentioned entities.

Phase-1

In this phase it represents a load variable F={F_1,F_2,------ F_m} possessing each $F_i(1 \leqslant i \leqslant m, F_i \in [0,1])$ parametric variable being in its stage of dynamic and variable m presents the entire counts of parameters.

Phase-2

Estimate the degree of load at cloud node

$$Load_{degree(N)} = \sum_{i=1}^{m} a_i F_i, \qquad (1)$$

$a_i(\sum_{i=1}^{n} a_i = 1)$ refer the weights that could differ for varied types of tasks or jobs and N refers for number of nodes being considered.

Phase-3

Decide the benchmarking parameter for estimation and then estimate the mean degree of cloud partition from the assistance of load degree at particular nodes. It can be given as follows:

$$Load\_degree_{avg} = \frac{\sum_{i=1}^{n} Load\_degree(N_i)}{n} \qquad (2)$$

The benchmark Load $Load\_degree_{high}$ is then decided for varying circumstances on the basis of $Load\_degree_{avg}$.

Phase-4

The encompassing nodes load and their respective status can be evaluated by the following defined conditions:

- Idle When

$$Load_{degree(N=0)}, \qquad (3)$$

It states that there is no task being processed by that specific node in partitioned cloud and therefore it is stated as Idle.

- **Normal** For

$$0 < Load_{degree}.N \leq Load_{degree_{high}} \qquad (4)$$

In this case the nodes are stated to be normal the node is normal and it can process other tasks or jobs.

- **Overloaded** When

$$Load\_degree_{high} \leq Load_{degree(N)} \qquad (5)$$

In such scenario, when the node is not accessible for exhibiting certain job like receiving any task until the cloud comes or returns back to its normal status. The load degree consequences are then feed into the load status tables generated while employing balancers of associated cloud partition. The individual balancer possesses its own table for load status and then it exhibits data update at every defined interval that results into automatic updation of node details for estimating the status of the specific cloud partition. The individual status of the cloud partition possesses varied load balancing solution.

Job Allocation and Load Balancing in Distributed Systems

A distributed system often consists of heterogeneous computing and communication resources. There are two main categories of load balancing policies: static policies

ISSN 2348 – 7968

and dynamic policies. Static policies base their decisions on collected statistical information about the system. They do not take into consideration the current state of the system.

Dynamic policies base their decisions on the current state of the system, where state could refer to, for example, the number of jobs waiting in the queue to be processed and job arrival rate. The nodes (computers) exchange this information periodically and will try to balance the load by transferring some jobs from heavily loaded nodes to lightly loaded nodes. Despite the higher runtime complexity, dynamic policies can lead to better performance than static policies.

4. Implementation and Results

In order to develop the complete system model, Java with Swing has been considered for programming application. For database requirement MySql or MS Access has been considered. For effective and user friendly development Eclipse development tool was employed. The phase of research implementation comprises well defined research planning, survey, system modeling its individual implementation of the existing approach and its system entities on implementation, developing the schemes for accomplishing changeover and estimation of changeover approaches. The overall system has been developed into some parts of segments. Four individual system models have been developed. These system components are: Main Modules, System Model, Main controller or balancers, Cloud partition based load scheduling or balancing.

User Module: A well defined GUI model for cloud owner, cloud server and receiver has been developed individually so as to present a better system handling and effective presentation. The overall system handling facilities for users have been developed.

System Model: Broadly the cloud infrastructures are classified into three categories; Public cloud, private cloud and hybrid cloud infrastructure. Here in this research work, we have taken into consideration of public cloud infrastructure. In system development for a public cloud of large size, it often encompasses several nodes and the associated nodes would be available in diverse geographical locations. Considering the prime contribution of main controller presenting a crisp function that which particular cloud portion is supposed to get the job responsibility. In the developed system the specific load balancer then decides the way of assigning jobs to the particular cloud nodes.

Main controller and balancers:

In this system model the main controlling section initially performs assignment of certain jobs to the proper partitioned node or cloud which in later exhibits communication with the balancer nodes in every encompassing node partition for refreshing the node information.

Data Flow diagram

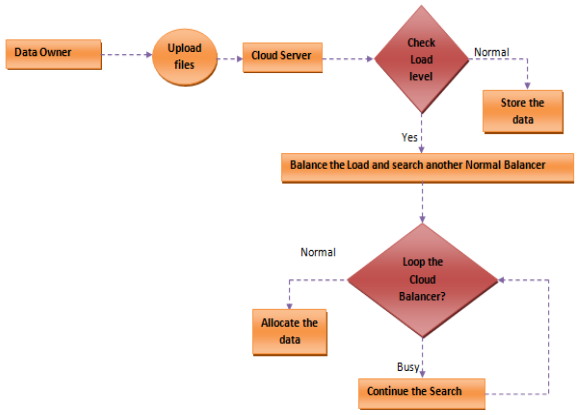

Fig 9: Data flow Diagram

The above mentioned figure represents the data flow diagram for the developed system model. The data originating or incepting from data owner till the load balancing has been presented in this figure.

Sequence Diagram

The load level estimation and respective load balance has been depicted in this figure. From this figure it is clear that after getting the data from cloud server the load level at different nodes is obtained and on the basis of load status (Idle, Normal and Overload) present at every nodes the scheduler starts functioning for further scheduling.

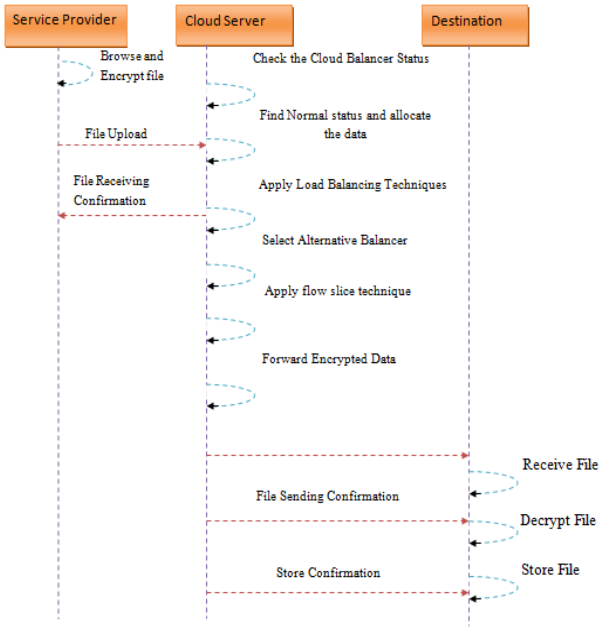

Fig 10: Sequence Diagram

ISSN 2348 – 7968

Use Case Diagram

The overall system function and its sequential procedures have been presented. As presented in the figure it can be found that the process of execution initiates form service provider and extends till destination or overall objectives. In this initial phase the balancer status is estimated and as per the available status the scheduling for data storage is accomplished. If the node is possessing overload status, the game theoretic based scheduling is done, where the overloaded nodes are scheduled and switched to the idle state. Meanwhile, on the basis of game theory the rotation of nodes and associated loads takes place. Once the overloaded node is scheduled for normal, the overload situation is ascended to next and thus the scheduled node becomes in Normal state. Once the load scheduling with game theoretic approach has been done then the scheduling with flow slice technique is implemented which is then followed by forwarding of the data which has already been encrypted with RSA public key cryptography. The forwarded encrypted file is then transmitted to the receiver terminal after authentication with symmetric key cryptographic technique.

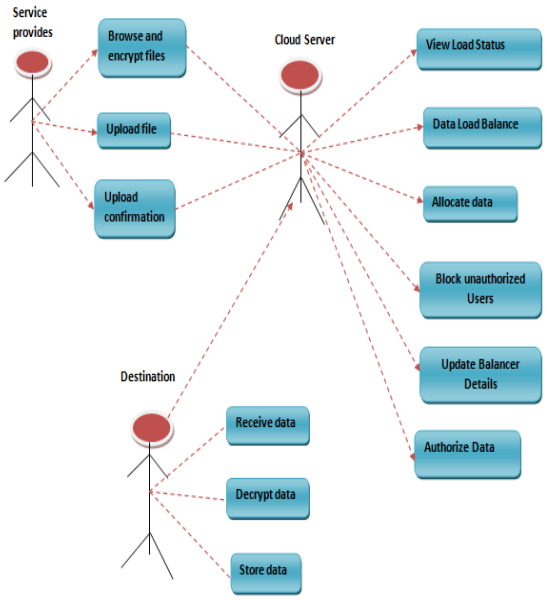

Fig 11: Use case Diagram

At the last stage of the data at receiver or destination terminal, using provided encryption approach the authentication of data would be ensured. It should be noted that in this process the public key cryptographic process has been employed and the generated encryption key can only decrypt the data file to be hosted on cloud infrastructure. In case user fails to present or feed genuine security key, that specific user would be blocked for further steps and it would not be able to store any data on public cloud infrastructure. This is the matter fact that the secured and authenticated data storage at public cloud infrastructure is the predominant issue and therefore a secured and authenticated mechanism of data validation is must. In order to accomplish the task of secured data storage in this work an optimized public key cryptographic approach has been implemented.

Result and Analysis

In this paper, the developed cloud framework not only justified its optimum function in terms of minimization of execution or response time but also providing a secure data storage facility for public cloud infrastructure. In order to accomplish the goal of secure and authenticated data storage on public cloud, a symmetric key cryptographic technique has been implemented. Considering the robustness and effective authenticated communication function of RSA cryptosystem, here in this work, the public key cryptographic scheme; RSA has been implemented. If the user tries to store data with wrong security key, in that case the model automatically register that specific user as faulty or malicious user and thus the user gets blocked by system. In order to retrieve the data and perform certain changes, the provided security key is needed. Once the genuine key has been provided, the user would be authorized to perform certain task, otherwise it would be blocked automatically and data could not be decrypted for further processing. Thus implementing such secure system, the unwanted data storage and malicious data storage can be avoided that would help administrator to utilize resources for genuine users.

Thus, considering the overall system developed and associated functions, it can be realized that the developed system, has illustrated better results not only in terms of efficient switching based load scheduling but incorporating the presented approach, the overall execution time can be minimized. The results obtained have illustrated better results in terms of swift switching operations. The consideration of security issues with public cloud infrastructure, the implementation of RSA cryptosystem based security feature, has illustrated better results for authenticated data retrieval of management on cloud. This can help administrator as well as users for secure cloud utilization.

5. Conclusion and Future Scope

Consideration the overall performance of the developed scheme, it can be stated that the proposed and hence developed mechanism of load balancing has exhibited better results for load balancing for public cloud infrastructure.

Cloud computing system has been considered as one of the revolutionary technologies for information technologies

ISSN 2348 – 7968

based applications. Although this is the predominant technology for industry, it still suffers from a number of limitations. Few of the issues related with cloud computing are proper load distribution and load balancing, migration of virtual machines (VMs), and unification of cloud servers and security for public, private and hybrid cloud infrastructures. Load balancing is the dominant issues in cloud computing. Considering the significance of public cloud infrastructure, the efficient load balancing and secured data portability is most important factor to be enhanced. In order to accomplish higher efficiency and efficient job allocation across cloud infrastructure, a game theoretic based approach has been developed. This work has also been enhanced with divisible load balance scheduling which has resulted into efficient load balancing with minimum execution time. In order to authorize the data hosting by genuine user, here in this work a noble public key cryptography scheme has been employed. RSA algorithm has been employed in this work for ensuring authentication of users to store data on public cloud. In this work the author has defined three different node statuses for every defined node. Four different balancers have been developed where individual balancer functions for three status; idle, normal and overloaded. In case of overload situation at certain node, the balancer can perform switching of load from one node to another, so as to balance the load at its optimum condition.

This is the matter of fact that the developed system represents a noble scheme of load balancing in public cloud infrastructure, but still it needs to be incorporated with real time functional scenario in real time cloud computing platform. Since, the developed scheme represents a conceptual framework; it still possesses opportunities for further enhancement in terms of real time application based system development, implementation of load balancing with certain real time cloud framework.

References

[1] Ranjan Kumar and G Sahoo, “Load Balancing Using Ant Colony in Cloud Computing”, International Journal of Information Technology Convergence and Services (IJITCS), Vol. 3, No.5, October 2013.

[2] Suriya Begum, Dr. Prashanth C.S.R, “Review of Load Balancing in Cloud Computing”, IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No. 2, January 2013 ISSN (Print): 1694-0784 | ISSN (Online): 1694-0814.

[3] Yu-lung Lo and Min-Shan Lai, “The Load Balancing of Database Allocation in the Cloud”, Proceedings of the International Multi Conference of Engineers and Computer Scientists IMECS 2013, Vol I, , March 13 - 15, 2013, Hong Kong.

[4] Belabbas Yagoubi, Meriem Meddeber, “Distributed Load Balancing Model for Grid Computing”, Revue ARIMA, Vol. 12, 2010, pp. 43-60.

[5] Venubabu Kunamneni, “Dynamic Load Balancing for the Cloud”, International Journal of Computer Science and Electrical Engineering (IJCSEE), ISSN No. 2315-4209, Vol. 1, Issue 1, 2012.

[6] N. G. Shivaratri, P. Krueger, and M. Singhal, “Load distributing for locally distributed systems”, Computer, Vol. 25, No. 12, December. 1992, pp. 33-44.

[7] S. Penmatsa and A. T. Chronopoulos, “Game-theoretic static load balancing for distributed systems, Journal of Parallel and Distributed Computing, Vol. 71, No. 4, April. 2011, pp. 537-555.

[8] 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, April. 2002, pp. 52-61.

[9] 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, March. 2012, pp. 28-30.

[10] Adhikari, J., Patil, S,. “Double threshold energy aware load balancing in cloud computing” Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013 pp.1 - 6

[11] Martin Randles, David Lamb, A. Taleb-Bendiab, “A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing”, IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, 2010.

[12] Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, “Cloud Computing A Practical Approach”, TATA McGRAW-HILL, 2010.

[13] Ali M. Alakeel, “A Guide to Dynamic Load Balancing in Distributed Computer Systems”, IJCSNS International Journal of Computer Science and Network Security, Vol.10 No.6, June 2010.