Cloud Partitioning Based Secured Load balancing Approach for Public Cloud Infrastructure

[1]Amritpal Singh and [2]Nisha Phogat
[1]Student, Master of Technology, [2]Head of Department, CSE
KIIT College Gurgaon, India

ABSTRACT

There has been a great significance of dynamic load balancing strategies for public cloud infrastructures (PCIs). An optimized load balancing strategy can enhance the performance of PCIs many folds. The predominant requirements for optimal public cloud are the efficient resource utilization and data security. On the other hand the dynamic load scheduling can make cloud system operational efficiently with varying load conditions. In this paper, a cloud partitioning based dynamic load balancing strategy has been proposed for public cloud infrastructure. A Nash-equilibrium based non-cooperative game theoretic approach has been developed for heterogeneous distributed cloud system. To ensure data security on cloud a RSA cryptosystem based user authentication scheme has been implemented that ensures genuine resource utilization and secured data access on PCIs. The developed system has exhibited better for task scheduling and load balancing in PCI systems.

Keywords: *Dynamic Load-balancing, public- cloud, Nash-equilibrium, RSA.*

I. INTRODUCTION

Cloud computing is one of the predominant technique in modern science that facilitates numerous internet based services for various IT enabled services. In this paradigm the users do get services without exploring a lot for details. As per NIST definition, cloud computing is an approach of facilitating ubiquitous, opportune, on-demand network access to numerous computing resources such as networks, servers, storage, applications and various services. These all facilities can be swiftly provided with nominal running cost and management exertion. The user count has been increasing with a very fast pace and in such scenario, the optimum resource management is required to be optimized. The effective load distribution is one of the predominant factors that can make whole system optimized. In fact in real time applications load at the server is unpredictable for public cloud infrastructure (PCI). Therefore, there is a need to have a scalable and efficient load balancing scheme that can effectively optimize load distribution across cloud infrastructure. The present algorithms encompass certain approach of either static load scheduling or scheduling with certain defined constraints, that confines the scalability of system for varied situations. On the other hand, very few works have been done for cloud authenticity. In this paper a cloud partitioning based dynamic load scheduling approach has been developed for distributed heterogeneous cloud systems. Additionally in this work, an RSA cryptography based authentication model has been integrated with proposed load balancing algorithm that authorizes data owner for uploading its data. The remaining manuscript are arranged in a way that Section II discusses related works, Section III presents the research contribution and system modelling which is followed by Section IV that presents system implementation and results. In Section V conclusion has been presented while in the last the references have been given.

II. RELATD WORK

In the effort to optimize the load balancing in cloud computing Grosu et al [1] formulated a non-cooperative and Nash equilibrium based game theoretic framework for optimal load balancing in heterogeneous distributed systems. For cloud partitioning based load balancing, similar works [2][3][4][5] employed game theory approach with Round Robin (RR) scheduling for cloud partitioning. In [4][5] certain switching scheme was employed for load balancing across cloud which function as per dynamic load at various nodes. First Come First Served (FCFS) rules was used in [4] with RR scheduling algorithm for load balancing. Dubey, AK et al [6] advocated a novel cloud computing paradigm for a trusted cloud infrastructure that can be monitored with client as well as cloud administrator and facilitated security to monitor data with RSA and MD 5 algorithm to preserve security of data in public cloud. Yuqi Zhang et al [7] employed Eucalyptus and advocated a high performance multicast algorithms called `steal – and – p2p′ on the basis of `non-steal' and `steal' paradigm. Prabavathy, B. et al [8] developed a dynamic load balancing paradigm for maintaining load balance across numerous nodes while exhibiting load migration in cloud storage. Wuhib, F et al [9] developed an OpenStack based resource management approach for private as well as public clouds and emphasized its application for IaaS. The switching facility in run time was achieved using certain sub-systems for controlling resource management. Kunamneni et al [10] emphasized on reducing waiting time in job scheduling using dynamic load balancing with comprehensive failover abilities in case of failure in server as well as distribution of traffic throughout multiple servers. In this paper a dynamic load scheduling paradigm has been developed while considering key security aspects in public cloud infrastructure.

III. OUR CONTRIBUTION

In this section the proposed system model for cloud partitioning based dynamic load scheduling has been presented. The key components of the proposed system like, load balancing approach, cloud partitioning, Nash equilibrium and game theoretic approaches have been discussed.

A. LOAD BALANCING

The load balancing approach in public cloud infrastructure (PCI) is implemented to ensure the optimum resource utilization and avoid the unfair job assignments to varied nodes. The significant objective is to avoid the situation where one system is being idle while another is overloaded. The optimum load distribution can be achieved by distributing load to those nodes which are having fewer loads. Broadly load balancing paradigms are classified into two categories; first, static Load balancing while another refers dynamic load balancing. In case of static load balancing strategy all the functional attributes and node information are known beforehand and the load scheduling strategies are implemented at compile time. In this scheme the load balancing paradigm remains unaltered while exhibiting functions. On the contrary, in dynamic kind the load balancing is processed in runtime and the associated strategies do vary as per time and present load situations. The dynamic load balancing scheme exhibits better performance in terms of load adaptability and sensitivity to the preciseness of load information across PCIs. Considering the requirement of highly robust resource utilization and load balancing need, dynamic algorithms are advocated. In case of any request rising from a system in dynamic paradigm, the extra loads are distributed across other nodes in the system. To ensure load distribution across all comprising slaves, the responses (from slaves) must be reaching simultaneously. It is must because there mustn't be waiting for any scrupulous computational device to reply before additional procedure could take place. In real time computing scenario, there exists certain heterogeneous computational components and the associated execution time is required to be estimated for every slaves. In heterogeneous kind of cloud infrastructure the load balancers must be characterized with the asymmetric load distribution.

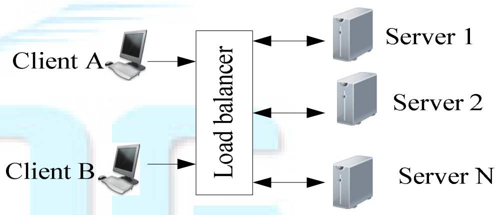

Figure 1: Load Balancer configuration

In fact for optimum resource utility a major portion of load is needed to be assigned to those nodes that are capable of higher computation, but in numerous cases, only having higher computation efficiency can't make a system efficient to decide what workload ratio should be assigned to it. In such scenario, an efficient load distribution can be accomplished by certain load balancers in heterogeneous cloud infrastructure. These potential components; *Load Balancer Units(LBUs)* are accountable for exhibiting priority activation that states that when a computational slave devices drops down to certain defined threshold these LBUs must make them awake to ensure optimum resource utilization and performance.

B. CLOUD PARTITIONING STRATEGIES

In this paper, the load balancing scheme has been developed for standard cloud framework based public cloud infrastructure where the services are facilitated over the Internet. In general PCI encompasses numerous nodes located at varied geographic locations. The partitioning of huge public cloud into certain parts is called cloud partitioning.

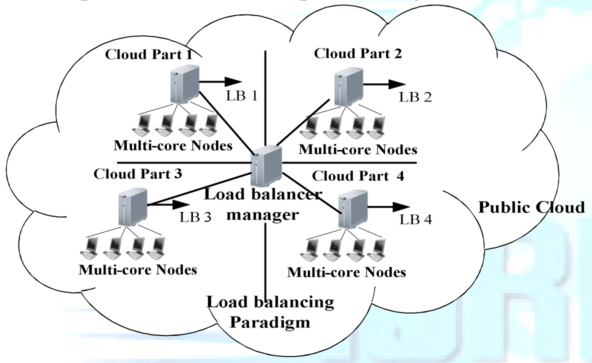

Figure 2: Cloud partitioning model

A partitioned public cloud does have numerous nodes associated with certain specific areas. Such subareas of PCI are stated as a collection of nodes possessing certain load balancing unit. The considered partitioning model has been illustrated in Figure 1. In this cloud partitioning based load balancing system, two consecutive phases have been taken into consideration. In the inceptional phase, considering the feasibility for system implementation the PCI is sliced into four partitions where the individual slice possesses its own *Load Balancer Unit (LBU)* allied with numerous multicore nodes. Additionally to monitor the LBUs a unit called *Load Balancer Monitor (LBM)* is required which is also stated as main controller. In this paper, the developed load scheduling algorithms functions for load balancing only when the cloud is partitioned into four sub-clouds. The nodes with each cloud partition can be in three status; NORMAL, IDLE and OVERLOADED. In IDLE case, majority of nodes are in idle state while in NORMAL state, few nodes are idle while remaining is in overload state. On the other hand all nodes are in overloaded state for OVERLOADED scenario. It must be noted that any of the node in cloud can accept any task request only when it is in its NORMAL state. While exhibiting load balancing the LBM functions for assigning tasks to the cloud part which is being IDLE.

a. LOAD BALANCER MONITOR (LBM)

The functions of the LBMs in PCI systems are as follows:

- To accept varied tasks from cloud users.
- To assign received user tasks.
- Status verification for considered cloud partition whether it's IDLE, NORMAL or OVERLOADED.
- Scheduling paradigm consideration:
  - Status_NODE=OVERLOADED; NODENo more task allocation, as node is already overloaded.
  - Status_NODE=NORMAL; NODEAssign user task to the LBU associated with this NODE.

b. DEGREE OF WORKLOAD (DW)

The degree of workload (DW) for any comprised node in cloud partition can be estimated by

$$DW(N) = \sum_{i=1}^{m} X_i * F_i$$

Where, N states for Current Node, $F_i$represents dynamic attributes with $F_i(1 \leq i \leq m)$ and m states the number of attributes under consideration. The parameter $X_i$represents the weight factor which depends on the user tasks with$F_i(1 \leq i \leq n)$.
The mean degree of workload (Mn_{DW}) on certain node in cloud partition can be evaluated by

$$Mn_{DW} = \sum_{i=1}^{n} DW\left({}^{N_i}/_{n}\right)$$

On the basis of estimated workload on certain partitioned node, in this paper, three node status (Status_NODE) have been defined which have already been discussed. The node status estimation can be presented as follows:

```
$If\ DW(N) = 0;\ then$
$Status\_NODEIDLE;$
$If\ 0 < DW\ (N)\ <= High\_DW;\ then$
$Status\_NODENORMAL;$
$Else\ (High\_DW \leq DW\ (N)$
$Status\_NODEOVERLOADED$
```

In our proposed task scheduling approach, LBM doesn't consider the node possessing OVERLOADED status as it is not capable of executing further user tasks. The LBM unit assigns user task or perform load balancing only to those nodes which possess either IDLE or NORMAL node status.

C. LOAD DISTRIBUTION STRATEGIES

As discussed in previous section, in the proposed system the task scheduling can be done only to those nodes which are either IDLE or NORMAL. This section elaborates the developed task scheduling paradigm for load balancing for nodes having IDLE or NORMAL status.

a. LOAD BALANCING FOR IDLE STATE

In case of IDLE situation the assocaited node or cloud partition possesses the capability of swift processing and for it certain load balancing approach [11] can be taken into consideration. A number of efforts have been made for load balancing with certain random approach, weighted Round Robin scheme, and enhanced Dynamic Round Robin (DRR) approach [12]. Being a simple paradigm generic RR approach is most in use and so, in this paper RR scheduling has been taken into consideration for load balancing. The RR scheme needn't to collect details about all the comprising nodes. In real PCI the connectivity and performance of every comprising node is not supposed to be identical, therefore this paradigm might cause OVERLOAD at certain nodes. Therefore in this paper, an optimized RR scheme has been developed. The optimized RR algorithm has been developed on the basis of *DW* in cloud partition. The node details are stored in a load balancing table which is arranged in certain ordered degree, ranging from Minimum to Maximum level of load degree. The developed system forms a clockwise queuing scheme and the queue is processed continuously. The node possessing minimum load degree would receive user task. Once the LBU has been updated for it status, the order of node gets changed in node status table. Then while, there might be certain contradictions for READ and WRITE with delay T for updating status details. The inconsistency is created if a user task arrives at the same time when status table is being updated. In such case irrespective of the change in node status, the details might be reflecting previous data. A parallel algorithm for node status and update can remove such limitations. The algorithmic approach for status management and task scheduling can be presented as follows:

```
Step 1: Initialize the process
Step 2: Estimate Degree of Workload (DW)
Step 3: Store Nodes in the load balancing table on the basis of Degree of Workload
Step 4: if
        Load Degree =Low, then
        Update Load degree
        Load DegreeJobs
        Else
        Refresh Load Status at the interval of T
Step 5: Update Flag
Step 6: Read Flag for ($T + 1 \geq Flag \geq T$
        If
        Flag=read, then
        Generate Load Degree for RR Scheduling
        Else
        Receive input as Load Degree for RR Scheduling
Step 7: Update Cloud Status and information
Step 8: End Process
```

b. LOAD BALANCING FOR NORMLE STATE

The problem of load balancing becomes more intricate in case of normal status as the dispatch rate of tasks is relatively faster as in idle state. The LBM induces more user tasks and every user is eager to get its tasks done as soon as possible. To resolve this kind of issue in [13] a game theoretic approach for static load balancing was proposed. In this paper we have implemented out dynamic load distribution system for distributed PCI system with game theoretic approach. We have implemented the non-cooperative game theoretic approach where a number of players make such initiatives that potentially affect the choices of other players. In fact the game theoretic approach can be stated as a study of divergence and collaboration. This approach is taken into consideration in the scenario when a number of agents (it may be certain individual, groups or even a mixture of it) and associated functions are interdependent. The consideration of game theory provides a measure to formulate architecture, analysis and strategic planning for load balancing in PCI system. The considered scheme refers the consideration of strategic planning which encompasses the attributes and functional constraints that can facilitate players certain approach to fulfill its interests. The implementation of game theory advocates rational measures for classes of games and it evaluate for respective property characterization. In general there exist three predominant categories of game theocratic load balancing scheme for PCIs. These are *global approach, cooperative approach and non-cooperative approach*. The proposed system has been developed based on non-cooperative game theory. In Non-cooperative game theoretic approach there could be multiple decision makers but they are not allowed to cooperate each other for certain decision making process. In this paper the individual decision maker is supposed to be enhancing its own response period and all decision makers attain equilibrium concurrently. The proposed game theoretic model has been discussed in next section.

D. NON-COOPERATIVE GAME THEORETIC APPROACH FOR LOAD BALANCING

In this section the developed non-cooperative game theoretic approach for dynamic load balancing in PCI system has been discussed. The mathematical formulations are mentioned in following section.

a. MATHEMATICAL MODELING FOR DISTRIBUTED LOAD SCHEDULING

The considered game theoretic scheme for load balancing in PCI encompasses nodes as players in every cloud partition and LBM induces tasks. Here we consider that in every cloud partition there are n numbers of nodes and LBM unit induces p jobs. The LBU has to make a decision for distributing tasks. The notations employed for proposed non-cooperative load balancing system are as follows:

$\mu_i - Average\ execution\ period, i = 1,2,3, \dots n$

$\emptyset_j - Users\ task's\ outputs, j = 1,2,3, \dots n$

$\emptyset - \sum_{j=1}^{m} \emptyset_j\ states\ for\ the\ task\ arrival\ rate.$

Consequently user is supposed to achieve the fraction S_{ji} (fraction of task j assigned to node i). The load balancing strategy for j is given by $S_j = S_{j1}, S_{j2}, \dots, S_{jn}$ and $S = S_{1,}, S_2, \dots, S_n$presents the strategy profile. The Nash equilibrium has been considered for achieving optimum dynamic load balancing. Some coordination between some users might be required to accomplish Nash equilibrium and therefore coordination between users is established for getting information. Considering the real time implementation the decentralization is needed which can be achieved by algorithms like greedy best reply [14]. In our approach M/M/1 queuing scheme has been considered for non-cooperative game model with n heterogeneous servers being accessed by m users. The individual computer has been queued in M/M/1 queuing approach.

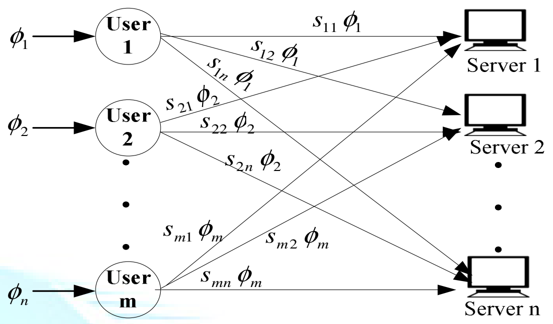

Figure 3: The distributed system model

In M/M/1 scheme the i^{th} server is characterized byμ_i. The task induced by certain user$j$isϕ_j, $\Phi = \sum_{j=1}^{m} \phi_j$ refers the final task arrival rate and $\emptyset$is required to be fewer as compared to cumulative dispensation rate$\Phi < \sum_{i=1}^{n} \mu_i$. Figure 3 illustrates the system modelling architecture. In such manner the users are supposed to decide where to schedule their task to accomplish optimum performance. Hence, the user $j(j = 1, \cdots, m)$ is required to estimate s_{ji} for their task which is further allocated to estimate $\Phi < i \sum_{i=1}^{n} s_{ji} = 1$for minimizing execution time. Consider s_{ji} represents a part of task that user$j$ transmits to server$i$. In this case an attribute $s_j = (s_{j1}, s_{j2}, \cdots, s_{jn})$ is stated as load balancing approach of the user where$s = s_1, s_2, \cdots, s_m$presents its strategic profile for game theoretic load balancing model. Implementing the M/M/1 queuing for individual computer the expected execution time for i^{th}computer is estimated by$F_i(s) = {}^{1}/_{\mu_i - \sum_{j=1}^{m} s_{ji} \phi_j}$. The complete execution time of user$j$ can be derived by the following expression

$$D_j(S) = \sum_{i=1}^{n} s_{ji}\, F_i(s) = \sum_{i=1}^{n} \frac{s_{ji}}{\mu_i - \sum_{k=1}^{m} s_{ki} \phi_k}$$

In our proposed load scheduling scheme a greedy best reply approach [15] has been considered for user's coordination requirement, which is must for information retrieval. This scheme exhibits update load strategies of every user at a defined interval by estimating optimum reply in opposition to the on hand load balancing measures of other users. We have implemented round-robin based update strategies for it. The aforementioned scheme for non-cooperative game model was implemented with a greedy best reply paradigm. The algorithm employed is given in Figure 4. The notations being used in the algorithms are, l – iterations count; S_j^l presents the strategies j estimated at iteration$l$, D_j^l states for expected execution period at iteration$l$; $\epsilon - a$ tolerability for tasks acceptance, Trans $(j,(p,q))$-transmit message (p,q) to$j$; $Rcv(j,(p,q))$ represents the acceptance of $(p,q)$from$j$.

```
User $j, (j = 1, \cdots, m)$ executes:
    Initialization:
        $s_j^{(0)} \leftarrow 0: D_j^{(0)} \leftarrow 0: l \leftarrow 1;$
    while (1) do
        if $(j \neq 1)$
            Rcv$(j-1, (norm, l))$:
            if$(norm = -1)$
                if $j \neq n)$ Trans$(j+1, (-1,-1))$:
                    exit:
      else
      if $(l \neq 1)$
        Rcv$(n, (norm, l))$:
            if $(norm < e)$
                Trans$(j+1, (-1,-1))$:
                    exit:
      if $(j = 1\ norm \leftarrow 0:)$
      for $i = 1, \dots, n$ do
      obtain $\mu_i^j$    Server: $(\mu_i^j \leftarrow \mu_i - \sum_{k=1,k\neq j}^{m} s_{ki}\emptyset_k)$:
      $s_j^{(i)} \leftarrow OPTIMAL: \mu_1^j, \cdots, \mu_n^j \phi_j)$:
      compute $D_j^{(i)}$:
      $nor\ mj \leftarrow |D_j^{(i-1)} - D_j^{(i)}|$;
      $nor\ mj \leftarrow norm + normj$;
      if $(j = 1) l \leftarrow l + 1$;
      if $(j = n)$ Trans $(1, (norm, l))$;
      else
          Trans$(j+1, (norm, l))$;
```

Figure 4: Nash-Equilibrium based game theoretic load balancing

As soon as the Nash-equilibrium is achieved the associated users would keep on employing similar strategy so as to keep system in equilibrium.

E. RSA BASED PUBLIC CLOUD AUTHENTICATION

In our system, a RSA cryptosystem based authentication module has been developed and incorporated with the dynamic load balancing approach. This developed RSA algorithm generates certain key which is required to upload or modify any data on public cloud infrastructure (PCI). This authentication model facilitates data security and even it prevents unwanted attacks on PCI. This also strengthens quality resource utilization for genuine users on PCIs.

IV. IMPLEMENTATION AND RESULT ANALYSIS

The overall proposed system has been developed with Java/Swing programming language with MySql database. The developed model has been analyzed for its scalability and efficiency in terms of execution time and varied task scheduling. For complete system realization, two modules have been developed where the first module facilitates user authentication and security features while second module presents the load balancing. The system has been evaluated for its scalability with varied load and task situations.

V. CONCLUSION

In public cloud infrastructure there is a great significance of dynamic load balancing strategies. It enhances the quality of services and resource utilization to ensure optimum performance. This work introduced a noble dynamic load scheduling approach based on Nash-equilibrium and round robin scheduling schemes for distributed cloud systems. In spite of the effective dynamic load balancing approach in this work, a public key cryptography based user authentication system has been developed that not only authorizes the users for optimum resource utilization but also provides data security from unauthorized data manipulation on public cloud. Overall the presented work can be a potential solution for public cloud performance optimization but it can be further enhanced by means of algorithmic optimization for real time implementation in cloud systems.

REFERENCE

[1] Daniel Grosu; Anthony T. Chronopoulos; “A Game-Theoretic Model and Algorithm for Load Balancing in Distributed Systems”;the 16th IEEE International Parallel and Distributed Processing Symposium on April 15, 2002, pp. 146-153.

[2] Mohammad Manzoor Hussain; Anandkumar Biyani; Bhavana Bidarkar; “Cloud Partitioning for Public Clouds using Load Balancing Model”; in

International Journal of Engineering Development and Research in reference IJEDR1303056.

[3] S.Velmurugan, P.Kumaran; International Journal of Research in Engineering & Advanced Technology; Volume 2; Issue 2; Apr-May, 2014.

[4] R.Logashree; S.Brintha Rajakumari; "Secured Load Balancing Model based on Cloud Partitioning using Round Robin Algorithm for the Public Cloud in Cloud Computing"; International Journal of Science, Engineering and Technology Research (IJSETR), Volume 3, Issue 4, April 2014.

[5] Xu, Gaochao; Pang, Junjie; Fu, Xiaodong; "A load balancing model based on cloud partitioning for the public cloud," Tsinghua Science and Technology on Feb. 2013, vol.18, no.1, pp.34-39.

[6] Dubey, A.K.; Dubey, A.K.; Namdev, M.; Shrivastava, S.S., "Cloud-user security based on RSA and MD5 algorithm for resource attestation and sharing in java environment," Software Engineering (CONSEG), 2012 CSI Sixth International Conference on 5-7 Sept. 2012, pp.1-8.

[7] Yuqi Zhang; Jun Wu; Yan Ma; Xiaohong Huang; Mingkun Xu, "Dynamic load-balanced multicast based on the Eucalyptus open-source cloud-computing system," Broadband Network and Multimedia Technology (IC-BNMT), 2011 4th IEEE International Conference on 28-30 Oct. 2011, pp.456-460.

[8] Prabavathy, B.; Priya, K.; Babu, C., "A load balancing algorithm for private cloud storage," Computing, Communications and Networking Technologies, Fourth International Conference on 4-6 July 2013, pp.1-6.

[9] Wuhib, F.; Stadler, R.; Lindgren, H., "Dynamic resource allocation with management objectives Implementation for an OpenStack cloud," Network and service management (cnsm), 2012 8th international conference and 2012 workshop on systems virtualization management (svm) on 22-26 Oct. 2012, pp.309-315.

[10] Venubabu Kunamneni "Dynamic Load Balancing for the Cloud" in International Journal of Computer Science and Electrical Engineering (IJCSEE) on 2012, Vol-1; Iss-1

[11] Gaochao Xu; Junjie Pang; Xiaodong Fu; "A Load Balancing Model Based on Cloud Partitioning for the Public Cloud"; IEEE Transactions on Cloud Computing Year-2013.

[12] Theodore L. Turocy; Bernhard von Stengel; "Game Theory"; CDAM Research Report LSE-CDAM-2001-09 October 8, 2001.

[13] S. Penmatsa; 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.

[14] T. Basar; G. J. Olsder; "Dynamic Noncooperative Game Theory"; SIAM; 1998.