Resource Monitoring and Workload Balancing Model for Public Cloud

Pragathi M , Swapna Addamani, Venkata Ravana Nayak

* Dept of Computer Science and Engineering, GSS Institute of Technology, Bangalore, India

Abstract- Workload balancing in the cloud computing environment has an important impact on the performance. Good workload balancing makes cloud computing more efficient and improves user satisfaction. This paper introduces a better workload balance model for the public cloud based on the cloud partitioning concept with a switch mechanism to choose different strategies for different situations. When the nodes are at idle status dynamic round robin algorithm is used.when the nodes are in normal status the algorithm which applies the game theory to the workload balancing strategy is used to improve the efficiency in the public cloud environment. A game theoretic frame work for obtaining a user optimal workload balancing scheme as been presented for non-cooperative game the structure of Nash equilibrium is used to select appropriate nodes to balance the workload The goal of cloud based architecture is to provide elasticity and ability to expand capacity on-demand. The workload balancing feature efficiently reduces the waiting time at each process steps.

Index Terms- loadbalancing model , cloud partition, public cloud, game theory

I. INTRODUCTION

Cloud computing is an attracting technology in the field of computer science. In Gartner's report, it says that the cloud will bring changes to the IT industry. 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. NIST gave a definition of cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool 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.

Cloud computing is efficient and scalable but maintaining the stability of processing so many jobs in the cloud computing environment is a very complex problem . Since the job arrival pattern is not predictable and the capacities of each node in the cloud differ, for load balancing problem, workload control is crucial to improve system performance and maintain stability. The workload balancing model is aimed at the public cloud which has numerous nodes with distributed computing resources in many different geographic locations. Thus, this model divides the public cloud into several cloud partitions. When the environment is very large and complex, these divisions simplify the load balancing. The cloud has a main controller that chooses the suitable partitions for arriving jobs while the balancer for each cloud partition chooses the best load balancing strategy main controller and balancer helps to balance the load and to improve the efficiency

II. PROPOSED APPROACH

The proposed model has a main controller and balancers to gather and analyze the information. Thus, the dynamic control has little influence on the other working nodes. The system status then provides a basis for choosing the right load balancing strategy. The load balancing model given aims at the public cloud which has numerous nodes with distributed computing resources in many different geographic locations. Thus, this model divides the public cloud into several cloud partitions. When the environment is very large and complex, these divisions simplify the load balancing. The cloud has a main controller that chooses the suitable partitions for arriving jobs while the balancer for each cloud partition chooses the best load balancing strategy.

ADVANTAGES

1.The proposed system is dynamic and there is equally the cloud partition is made to balance the load between 'N' Number of Partitions.

2.Dynamic round robin algorithm is used in the proposed system in which the system will take less time and less cost to balance the load and allocate the cloud file.

3.Here the cloud admin can change the status of cloud from one state to another state

4.As soon as the job arrives the cloud partition will start the load balancing to schedule the job in the cloud and over comes FCFS(First Come and First Serve).

III. WORKLOAD BALANCING

Workload balancing is the technology to distribute workload across multiple computers or a computer cluster, central processing units, disk drives, RAM, or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, avoid overload, and minimize application down time. The workload balancing services is usually provided by dedicated software or hardware.

The purpose of Workload balancing include:

- To spread the load amongst a number of machines/locations
- To provide redundancy in case one machine/server fails

• To provide zero down time during patch installations on servers or updates to applications on server

Workload balancing in public cloud

Workload balancing is a major key for a successful implementation of cloud environments. The main goal of a cloud-based architecture is to provide elasticity, the ability to expand and contract capacity on- demand. Sometimes additional instances of an application will be required in order for the architecture to scale and meet demand. That means there is a need for a mechanism to balance requests between two or more instances of that application. The mechanism most likely to be successful in performing such a task is a load balancer.

There's no other way to assume increased load other than adding new instances and distributing that load with software or hardware Similarly, when the additional instances of that application are de- provisioned, the changes to the network configuration need to be reversed, but software and hardware load balance is easy to scale up or scale down. Obviously a manual process would be time consuming and inefficient, effectively erasing the benefits gained by introducing a cloud-based architecture in the first place.

Advantages of Workload balancing

• Esnures that connections are not directed to a server that is down.
• Effective distribution of traffic among multiple servers
• Works as a driver rather than as a service
• Manages resources efficiently
• Improves the application response time
• If we have two members in load balance pool, with priority function we can send all the traffic to one node and keep other node as a backup thus helps with disaster recovery.

IV. RESOURCE MONITORING

Resource Monitoring refers to monitoring critical resources like RAM, CPU, memory, bandwidth, partition information, running process information and utilization and swap usages etc. Cloud computing has become a key way for businesses to manage resources, which are now provided through remote servers and over the Internet instead of through the old hard-wired systems which seem so out of date today. Cloud computing allows companies to outsource some resources and applications to third parties and it means less hassle and less hardware in a company. Just like any outsourced system, though, cloud computing requires monitoring.

V. IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system.. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

5.1 System Model :

A public cloud is based on the standard cloud computing model, with service provided by a service provider. A large public cloud will include many nodes and the nodes in different geographical locations. Cloud partitioning is used to manage this large cloud. A cloud partition is a subarea of the public cloud with divisions based on the geographic locations. with the main controller deciding which cloud partition should receive the job. The partition load balancer then decides how to assign the jobs to the nodes. When the load status of a cloud partition is normal, this partitioning can be accomplished locally. If the cloud partition load status is not normal, this job should be transferred to another partition.

5.2 Main controller and balancers:

The load balance solution is done by the main controller and the balancers. The main controller first assigns jobs to the suitable cloud partition and then communicates with the balancers in each partition to refresh this status information. Since the main controller deals with information for each partition, smaller data sets will lead to the higher processing rates. The balancers in each partition gather the status information from every node and then choose the right strategy to distribute the jobs as shown in the conceptual diagram below

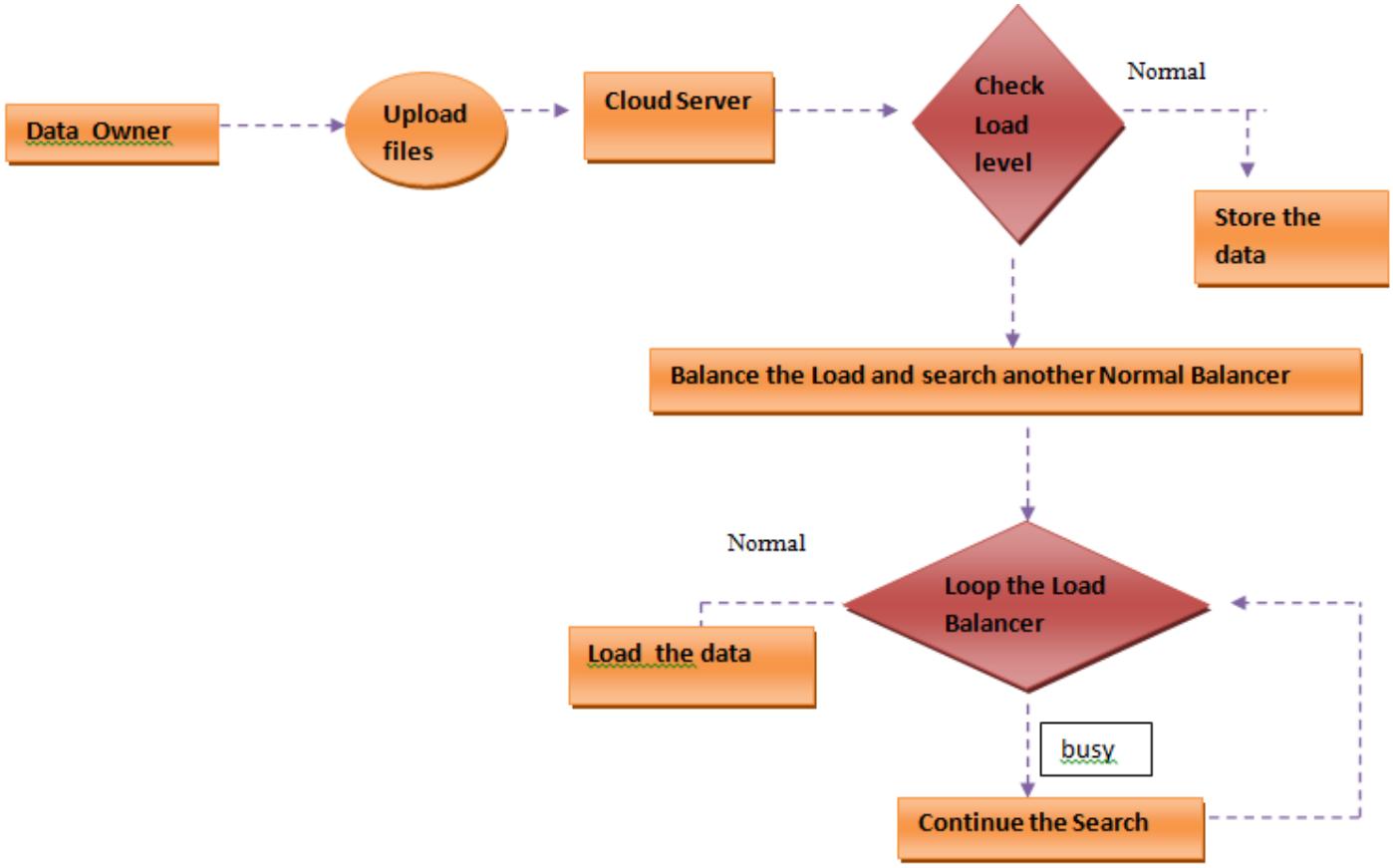

Fig 1 : Job assignment strategy

5.3 Assigning jobs to cloud partition

When a job arrives at the public cloud, 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 α_c, change to idle status.
- Normal: When the percentage of the normal nodes exceeds β, change to normal load status.
- Overload: When the percentage of the overloaded nodes exceeds ϒ, change to overloaded status.

The parameters α , β, ϒ are set by the cloud partition balancers.

The main controller has to communicate with the balancers frequently to refresh the status information. The main controller then dispatches the jobs using the following strategy: When job i arrives at the system, the main controller queries the cloud partition where job is located. If this location's status is idle or normal, the job is handled locally. If not, another cloud partition is found that is not overloaded. The algorithm is shown in Algorithm 1.

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

5.4 Assigning jobs to nodes in cloud partition

The first task is to define the load degree of each nodes. The load degree is computed from these parameters as below:

Step 1: Define a load parameter set:
F = { F1,F2,….Fm} with each parameter being either static or dynamic. m represents the total number of the parameters.

Step 2: Compute the load degree as:
Load_ degree (N)= $\sum_{i=1}^{m} \alpha i \, Fi$
αi are weights that may differ for different kinds of jobs.N represents the current node.

Step 3 Define evaluation benchmarks. Calculate the average cloud partition degree from the node load degree statistics as

Load degree(avg) = $\frac{\sum_{i=1}^{m} Load\_degree(Ni)}{n}$

The bench mark Load degreehigh is then set for different situations based on the Load degreeavg.

Step 4 Three nodes load status levels are then defined as:

- Idle When Load degree.(N)== 0;

There is no job being processed by this node so status is idle

- Normal For 0 < Load degree.(N) <= Load degreehigh;

The node is normal and it can process other jobs

- Overloaded When Load degreehigh <= Load degree(N);

The node is not available and cannot receive jobs until it return to normal

VI. CLOUD PARTITION WORKLOAD BALANCING STRATEGY

Good load balance will improve the performance of the entire cloud. Therefore, the current model integrates several methods and switches between the load balance method based on the system status. Here, the idle status uses an improved Round Robin algorithm while the normal status uses a game theory based load balancing strategy.

6.1 Workload balance strategy for the idle node status

When the cloud partition is idle, many computing resources are available and relatively few jobs are arriving. In this situation, this cloud partition has the ability to process jobs as quickly as possible so a simple load balancing method can be used.

First, the nodes in the load balancing table are ordered based on the load degree from the lowest to the highest and two Load Status Tables should be created as: Load Status Table 1 and Load Status Table 2. A flag is also assigned to each table to indicate Read or Write. When the flag = "Read", then the Round Robin based on the load degree evaluation algorithm is using this table. When the flag = "Write", the table is being refreshed, new information is written into this table. Thus, at each moment, one table gives the correct node locations in the queue for the improved Round Robin algorithm, while the other is being prepared with the updated information. Once the data is refreshed, the table flag is changed to "Read" and the other table's flag is changed to "Write". The two tables then alternate to solve the inconsistency.

6.2Workload balance for the normal node status

When the cloud partition is normal, jobs are arriving much faster than in the idle state and the situation is far more complex, so a different strategy is used for the load balancing. Each user wants his jobs completed in the shortest time, so the public cloud needs a method that can complete the jobs of all users with reasonable response time .In order to achive this game theory is used.

Game theory has non-cooperative games and cooperative games.

Cooperative game: In this case there are several decision makers that cooperate in making decisions such that each of them will operate at its optimum Decision makers have complete freedom of preplay communication to make joint agreements about there operating points.

Non-cooperative approach:In this case there are several decision makers that are not allowed to cooperate in making decisions.Each decision maker optimizes its own response time independently of the others and they all eventually reach an equilibrium. This situation can be viewed as a non cooperative game among decision makers.The equilibrium is called Nash equilibrium and it can be obtained by non distributed non cooperative policy. At the Nash equilibrium decision maker cannot recive any further benefit by changing its own decision.

In non-cooperative games, each decision maker makes decisions only for his own benefit. The system then reachs the Nash equilibrium, where each decision maker makes the optimized decision. The Nash equilibrium is when each player in the game has chosen a strategy and no player can benefit by changing his or her strategy while the other players strategies remain unchanged.

The players in the game are the nodes and the jobs. Suppose there are n nodes in the current cloud partition with N jobs arriving, then define the following parameters:

μi : Processing ability of each node, i=1……..n

Øj :Time spending of each job

$Ø = \sum_{j=1}^{N} Øj$:Time spent by entire cloud partition

sji: Fraction of job j that assigned to node i

sji is calculated using below algorithm

Input : Available processing ability of each node μ1, μ2,μ3……μn

Time spent for each job.

Output : sj1,sj2,sj3,……sji

1. Sort the nodes in decreasing rates of their processing ability (μ1 >= μ2>=…….μn);

2 . $t \leftarrow \frac{\sum_{i=1}^{n} \mu i - Øj}{\sum_{j=1}^{n} \mu i}$

3. while (t >μi) do

sji ← 0

n ← n -1

$t \leftarrow \frac{\sum_{i=1}^{n} \mu i - Øj}{\sum_{i=1}^{n} \mu i}$

4. for i=1……n do

sji ← (μi – t (μi) 1/Øj

In this model the most important step is finding the value of sji of all node. This procedure gives the nash equilibrium to minimize response time of each job.

VII. RESULTS

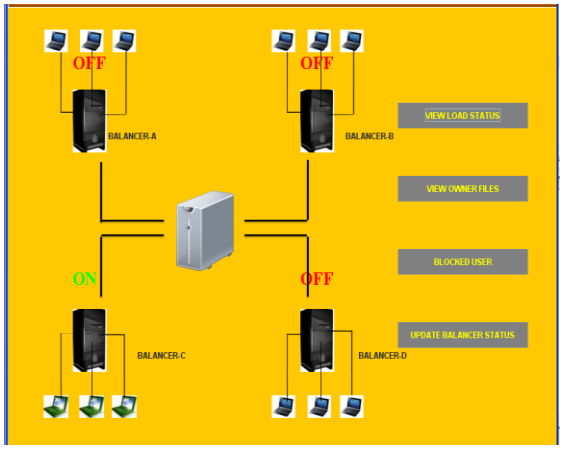

Fig 2. Main controller, balancer, and nodes

Here Main controller will assign job to suitable cloud partition and then communicate with balancer In the figure maincontroller has assigned the job to balancer C .

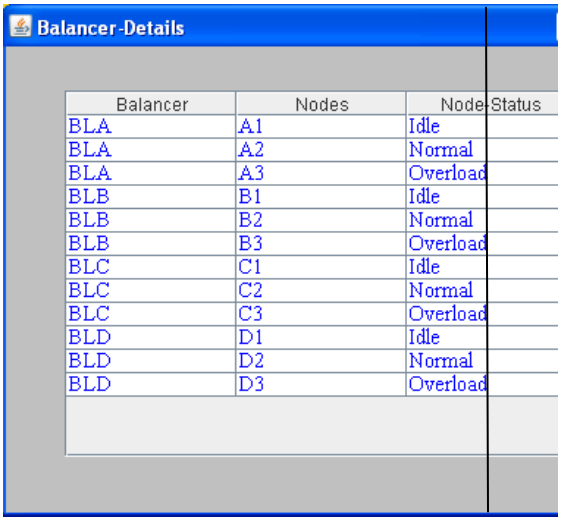

Fig 3 Balancer Details

Node status can be viewed as shown in the above figure as the load arrives if load status is idle or normal then job is handeled by that node or else another cloud partition is found.

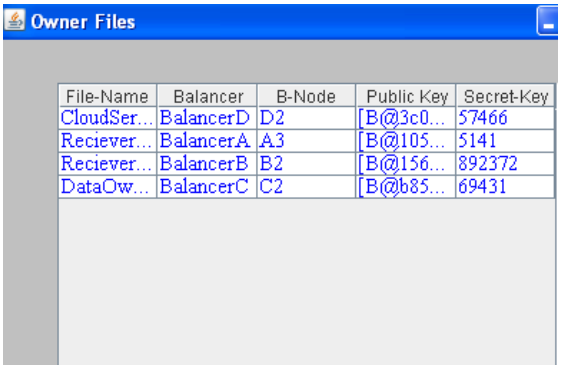

Fig 4 Veiw Files

The files that are uploaded at different nodes can be viewed

VIII. CONCLUSIN AND FUTURE WORK

Main aim was to develop a resource monitoring and workload balancing model in order to improve performance and maintain stability of processing so many jobs in public cloud. This objective is achived by constructing a workload balancing model for public cloud based on cloud partitioning concept with switch mechanism to choose different strategy to improve the efficiency in public cloud environment

8.1 FUTURE WORK

Cloud division rules: Cloud division is not a simple problem. Thus, the framework will need a detailed cloud division methodology.

How to set the refresh period: In the data statistics analysis, the main controller and the cloud partition balancers need to refresh the information at a fixed period. If the period is too short, the high frequency will influence the system performance. If the period is too long, the information will be too old to make good decision.

A better load status evaluation: A good algorithm is needed to set Load degreehigh and Load degreelow, and the evaluation mechanism needs to be more comprehensive.

Find other load balance strategy: Other load balance strategies may provide better results, so tests are needed to compare different strategies. Many tests are needed to guarantee system availability and efficiency.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] D. MacVittie, Intro to load balancing for developersThealgorithms, https://devcentral.f5.com/blogs/us/introduction-to-load-balancing-for-developers-ndash-th algorithms, 2012.

[5] B. Adler, Load balancing in the cloud: Tools, tips and techniques, http://www.rightscale. com/infocenter/white- papers/Load-Balancing-in-the-Cloud.pdf, 2012

[6] M. Randles, D. Lamb, and A. Taleb-Bendiab, A comparative study into distributed load balancing algorithms for cloud computing, in Proc. IEEE 24th International Conference on Advanced Information Networking and Applications, Perth, Australia, 2010, pp. 551-556.

[7] 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.

AUTHORS

First Author – Pragathi M, Dept of Computer Science and Engineering, GSS Institute of Technology, Bangalore, India
Second Author – Swapna Addamani, Dept of Computer Science and Engineering, GSS Institute of Technology, Bangalore, India

Third Author – Venkata Ravana Nayak, Dept of Computer Science and Engineering, GSS Institute of Technology, Bangalore, India