Workload Aware Partitioning and Load Balancing in Cloud Computing

Varsha Salunkhe, Prashant Yawalkar

***Abstract*— Load Balancing plays a significant role in improving system performance. As cloud is composed of huge servers, load balancing techniques are used to balance load over multiple servers. Workload aware partitioning helps to balance the load in cloud efficiently as it distributes the incoming load as per the type of work. This reduces the response time and thus helps to improve the performance of the system. In this paper, we proposed an algorithm for logical partitioning of resources in cloud and distributing the workload accordingly to avoid overloading of servers. The goal of this approach is to design and develop the concept of load balancing using cloud partitioning by classification of jobs. Thus proposed partitioning algorithm should help in placement of workloads in correct region and balance the overall load effectively.**

***Index Terms*— load balancing, partition, workload**

I. Introduction

Cloud Computing is an emerging technology in today's world. Cloud offers various services and resources to the consumers on demand and pay as per usage basis over internet. Cloud Services are categorized in three types: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). In the world of internet, cloud is formed from logical grouping of systems depending upon their locations. Hence, cloud can be viewed as distributed system over internet [2]. Cloud computing mainly deals with computation, data access and storage services that may not require end-user knowledge of the physical location and configuration of the system that is delivering the services.

Load balancing is the technique of distributing the entire load equally to all individual nodes in a group of systems. Load balancing schemes avoids overloading of servers and improves response time. Job arrival pattern is not predictable in cloud as well as capacity of each node is different [1]. This makes workload control difficult. Load balancing in computer networks is a technique used to distribute workload across several network links of computers. Load balancing simplifies networks and resources by providing a maximum throughput with minimum time, thus it helps to improve performance by optimized use of available resources and reduces latency and response time. Multiple resources are distributed to multiple servers or nodes so that load is distributed evenly to carry out client's request. Load balancing helps to accomplish a high user satisfaction and better resource utilization. When one or more components of any service fail, load balancing facilitates continuation of the service by implementing fair-over, that is, it helps in provisioning and deprovisioning of instances of applications without any failure. It distributes each computing resource equally and efficiently.

II. Literature Survey

Many studies and analysis have been performed on load balancing for the cloud environment. There are many load balancing algorithms, such as Round Robin, Particle Swarm Optimization algorithm, Equally Spread Current Execution Algorithm, Self-organized Load Balancing algorithm, Ant Colony algorithm etc. Chaczko et al [3] demonstrated the critical role that the load balancing plays in improving and maintaining the availability in cloud systems. Availability of clouds is mainly evaluated by ubiquity of information comparing with resource scaling. The network structure or topology should be taken into account when creating the logical rules for the load balancer.

In Cooperative Load Balancing algorithm [4], system acts as to play cooperative game among themselves. It works on static load balancing problem. Grosu et al. proposed a Nash Bargaining Solution (NBS) which provides a Pareto optimal allocation. It considers a game in which each computer is a player and it must minimize the expected execution time of jobs that it processes. To balance the queues of service requests, Self-organized Load Balancing algorithm [5] was developed in which highest capacity node act as super peers. These nodes offer different types of services. At first level, algorithm find out the capacity of every peer, i.e., the amount of service requests that peer is able to fulfill in a client time unit. Round Robin algorithm [6] is the simplest algorithm used for load balancing in cloud computing to avoid overloading and underloading of servers. Equally spread current execution algorithm [7] which assigns a job to each node with priority. It makes use of spread spectrum technique in which it distributes the load over various nodes by checking its load size. Once the

Manuscript received May 23, 2015.

Varsha Salunkhe, Computer Department, MET BKC IOE, Savitribai Phule Pune University, Nasik, India,

Prashant Yawalkar, Computer Department, MET BKC IOE, Savitribai Phule Pune University, Nasik, India,

available resource (virtual machine) size and the size of the job matches, then immediately the job scheduler allocates appropriate virtual machine or resource to the job in a queue.

Junjie Pang et al [1] introduced a concept of better load balancing model to handle the workload on huge public cloud. He introduced a strategy based upon cloud partitioning concept which divides a huge cloud into partitions which are geographically distributed. When the environment is very large and complex, these divisions simplify the load balancing.

According to the algorithm specified in [1], jobs are assigned to the partition having minimal load.

III. System Architecture

In a cloud data center environment there are multiple nodes zoned in various areas depending on the processor type or networking technologies. It is a problem to the cloud service provider to identify node for a particular type of service on demand. The user may provision multiple workloads to do similar work or may run completely different workloads randomly. While doing a collective work, a user expects maximum performance by going through minimal overheads.

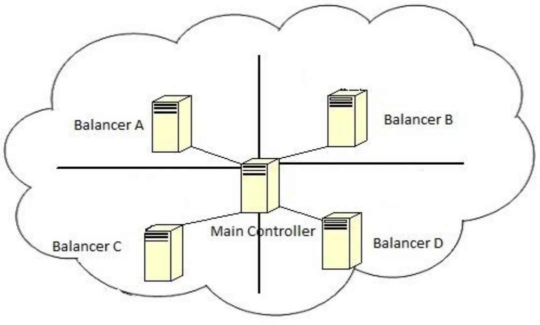

Fig 1. Main Controller and cloud partitions

A cloud which has various nodes with distributed computing resources. Thus, proposed model divides the cloud into several cloud partitions by identifying the similarity between nodes. The load balancing strategy is based on the cloud partitioning concept. In this model, cloud is logically partitioned into small segments. Each partition has a Load balancer associated with it as shown in fig1. There is a main controller system which is responsible to manage the load balancer. After partitioning the cloud into different partitions, load balancing then starts [2]. Load is assigned to the node with highest weight. Different types of services are requested by client as a job to be executed. This job first enters the load balancer that is, main controller where resources required for the job are identified. Then, load balancer searches for the best suitable partition for the job and checks whether it is available or not. If the status of the partition is normal, then the job is assigned to the particular partition and then the partition balancers choose the appropriate node to process the job. Load status information of each node is updated at partition balancers through which it can select the appropriate node for job execution. Similarly, load status information of partition balancers are updated and maintained at main controller. Each partition has categorized in any of three classes [1]:

- *Idle*: When the percentage of idle nodes exceeds α, 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 γ , change to overloaded status.

The parameters α, β , and γ are set by the cloud Partition balancers.

A. Job Assignment (JA):

In this model, Main controller is responsible for receiving the jobs from different end users. Then, choose a specific partition for the received jobs and check the status of the cloud partition. If the partition Status=Overload, there will be no allocation.

If partition Status=Idle or Normal then forward the Jobs to the respective Load balancer (LB). Now the load balancer activated and starts its work.

B. Load Point (Lp):

The Load point is the amount of load a node carries (Lp). Load point of a node N in any cloud partition is calculated periodically from following equation:

$$Lp(N) = \sum_{i}^{m} W_i \, Ri$$

where $R=\{R_{1,} R_{2,} R_{3, \ldots,} R_m\}$

and N=Current Node, Ri are the load parameter set such that (1<=i<=m) , m represents the total number of parameter . Wi are weights that may differ for different kinds of job for all (1<=i<=n). Parameters can be CPU speed, RAM, disk space, SSID, Processor type etc. Average Load Point (Lp) of the cloud partition will be calculated as:

$$Lp_{avg} = \sum (N_i)/n$$

IV. Implementation Details

This section gives idea about the strategy of implementation. The system can be divided into four stages.

A. Job Classification

This stage classifies the type of job requested by client. Job may include compression, conversion of files, decompression, encryption, decryption, data storage. Client uploads the files and selects the type of job to be processed further by load balancer.

B. Resource Classification

Resource Classification divides the servers in cloud based on the available resources. Cloud is partitioned such that servers provide specialized services. Resources are the parameters used to divide the cloud in partition. They include parameters such as CPU type, processing speed, RAM, Disk Space, CPU Usage.

C. Job Assignment

A Job request from client initially arrives at load balancer. Load balancer then chooses appropriate partition according to the resources required for the job and assigns job to the selected partition.

D. Resource Monitoring

This system keeps a track of resources utilized and updates load balancer about the availability of resources. When the workload is assigned to particular server, main controller updates the resources tables. It maintains log tables to keep information about resources utilized and available.

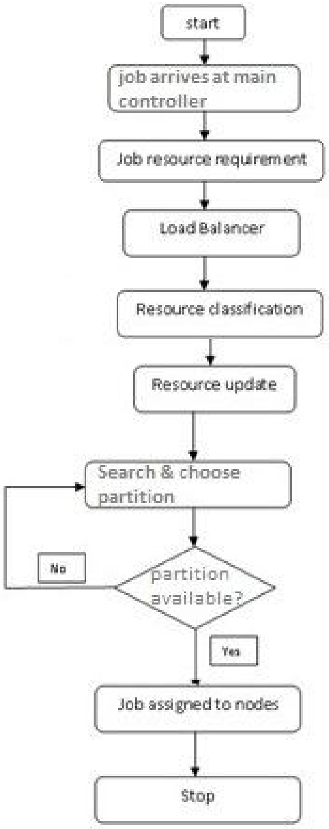

Fig 2. System Flow

As shown in Fig 2., job arrives at the main controller where job resource requirements are identified. Load balancer then searches for the partition and selects the appropriate partition. It then updates the resources utilized for the particular job. Job is assigned to the partition and gets executed and result is returned to the client and stored on the server. Load status information of each node is updated at partition balancers through which it can select the appropriate node for job execution.

Following are the data structures used in the algorithm:

TABLE 1: NOTATIONS AND DATA STRUCTURES USED IN ALGORITHM

Notation	Description
Jtype	Type of job
P	Maximum value at which server is said to be overloaded
req[i][j]	Resource requirement of job
avail[p][j]	Available resources
Load	Total allocated load
Wsst	Weight assigned to partitions according to jtype

Algorithm used for placing the job in appropriate partitions is given below:

```
1. For job j=1 to n
2. Identify type of job jtype
3. Find partition p from normal class such that
      req[i][j] == avail[p][j] && wt == jtype && load<= ρ
4. If found,
      Place the job for execution
5. Else Find partition p from normal or idle class such
      that req[i][j] == avail[p][j] && load <= ρ
6. Place the job for execution
7. Return result
8. End if
9. If load >= ρ
10. Shift nodes from normal class to overload class
      and Place job in queue
11. End if
12. End for
```

State Diagram for the proposed model can be viewed as following:

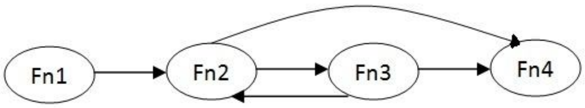

Fig 3. State diagram

In fig 3 ,Fn1 is Set of Jobs, Fn2 denotes function for Load Balancer, Fn3 is function responsible for Resource monitoring and Fn4 is function for partition.

V. RESULT ANALYSIS

A. Experimental setup

Experimentation was performed on four nodes in cloud with different RAM size and CPU Capacities. Specification and specialization of these nodes are listed in table 2. The experimental results of execution of job on cloud server are shown in table 3 where five type of jobs are been considered. They are compression, conversion(text to pdf and vice versa), encryption, decryption, load/save. The parameters for partitioning are CPU type, CPU speed, RAM, disk space, processing features.

TABLE 2: SPECIFICATION OF NODES ON CLOUD SERVER

Processor		Intel(R) Xeon(R) CPU E5-2640V2	
Nodes	**RAM(MiB)**	**CPU Speed(MHz)**	**Specialization**
Node 1	768.00	74	Encrypt, Conversion
Node 2	512.00	49	Decompress
Node 3	896.00	87	Decrypt
Node 4	640.00	62	Upload, Compress

TABLE 3: JOB EXECUTION ON SPECIALIZED SERVER AND THEIR TIMINGS

Job Type	**File size(Kb)**	**Node**	**Time(ms)**
Compression	11.5	Node 4	0.534
Compression	928	Node 4	20.859
Encryption	73	Node 1	1.711
Decryption	88	Node 3	1.408
Text to pdf	77	Node 1	2.14

B. Response Time

Response Time of each Job execution is different because each job is heterogeneous. The expected response time of each job can be found with the help of the following formula:

Response Time = Fin_time – Arr_time + T_delay (1)

Where Arr_time is the arrival time of user request and Fin_time is the finish time of user request and T_delay is the transmission delay [8].

VI. Conclusion

As load balancing plays an important role in increasing performance of the system, an algorithm that can schedule the job by assigning it to appropriate partition should help in avoiding overloading of servers and improving the systems performance. Our algorithm works on the same principle and finds the expected response time for each job which helps in monitoring the system by keeping track of the resources utilized.

Acknowledgment

The authors wish to thank MET Institute of Engineering Bhujbal knowledge city Nashik, India for providing lab facilities. Authors are also thankful to Gaochao, Xu Junjie Pang, Xiaodong Fu, “A load balancing model based on cloud partitioning for the public clouds,” for their work in this area. The mentioned paper is major guideline in this work.

References

[1] Gaochao, Xu Junjie Pang, Xiaodong Fu, “A load balancing model based on cloud partitioning for the public clouds,” IEEE. Trans On cloud computing year 2013

[2] V S Salunkhe, “Review on Load Balancing Model in Cloud Computing”,Proc. International Journal of Computer Applications, Conf. Innovations and Trends in Computer and Communication Engineering, Dec 2014.

[3] Z. Chaczko, V. Mahadevan, S. Aslanzadeh, and C. Mcdermid, “Availability and load balancing in cloud computing”, International Conference on Computer and Software Modeling, Singapore, 2011.

[4] 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.

[5] Giuseppe Valetto, Paul Snyder, Daniel J. Dubois, Elisabetta Di Nitto and Nicolo M. Calcavecchia, “A self-organized load balancing algorithm for overlay based decentralized service networks”,2013.

[6] Pooja Samal, Pranati Mishra, “Analysis of variants in Round Robin Algorithms for load ba;ancing in Cloud Computing”, IJCSIT, vol.4(3),2013.

[7] Ram Prasad Padhy ,P Goutam Prasad Rao, “Load Balancing in Cloud Computing Systems”, National Institute of Technology, Rourkela, India, 2011.

[8] Mangal Nath Tiwari ,“Analysis of Public Cloud Load Balancing using Partitioning Method and Game Theory”, IJARCSSE, Volume 4, Issue 2, February 2014.