International Journal of Emerging Trend in Engineering and Basic Sciences (IJEEBS)
ISSN (Online) 2349-6967
Volume 2 , Issue 1(Jan-Feb 2015), PP388-391

Load Balancing Architecture for Public Cloud

Apurva Kamble[1], Priyanka Jadhav[2], Ankit Soni[3], Vaishali Barkade[4]
[1 2 3 4]*(Computer Engineering Department, Rajarshi Shahu College of Engg., India)*

Abstract :- Cloud computing is model for enabling convenient, on demand network access to a shared pool of configurable computing resources. Load Partitioning is an optimal approach for public cloud. The Load balancing is a process of distribution of workload among different nodes or processors. Load balancing is to improve performance of cloud environment through an appropriate distribution strategy. A dynamic load balancing scheme is used for its flexibility. The random arrival of load in public cloud can cause some server to be heavily loaded while other server is idle or lightly loaded. The model strategy is to divide the public cloud into several cloud partitions which helps to overcome from the problem of the load on cloud by improved response time and processing time.

Keywords :- About five key words in alphabetical order, separated by comma

I. INTRODUCTION

A cloud has elements such as clients, data center and distributed servers. It provides benefits like fault tolerance, high availability, scalability, flexibility, reduced overhead for users, reduced cost of ownership, on demand services. Central to these issues lays the establishment of an effective load balancing algorithm. The load on the server can be a CPU load, memory capacity, delay or network load.

We construct Model, as Load balancing model which gives the process of distributing of the server load among various nodes of a distributed system to improve resource utilization as well as job response time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work.[1]

This technique can be sender initiated, receiver initiated or symmetric type (combination of Sender initiated and the Receiver initiated types). Our aim is to perform effective load balancing using cloud partitioning algorithm maximize or minimize different performance parameters for the public clouds.

Load balancing basically performs two major tasks, one is the resource providing or resource allocation and other is task scheduling in distributed environment or systems.[2]

Task	Sub Category	Issue Resolved
Resource Providing	At Host and VM level	Efficient Utilization of resources
Task Scheduling	Space Sharing Time sharing	Minimizing overall response time of tasks

TABLE1. COMPARISON BETWEEN RESOURCE PROVIDING AND TASK SCHEDULING

II. RELATED WORK

In Existing model, all tasks can work in parallel to speed up execution of the job. The job cannot be finished until all tasks are done. The time span between job's start time and job's finish time is called *job completion time*. In addition, if a task reads its block from server's local disk, it is called *data local*; otherwise, the task is called *data-remote* if it retrieves a copy of its block from a remote server. Since most cloud computing systems are implemented on commodity or virtual hardware, the data transfer cost gives a great impact on the system performance.

To improve data locality, some approaches have attempted to delay schedule the job until an appropriate server is arrived. A limitation of these policies is that servers are not always become idle quickly enough as assumed. If the cluster is *overloaded*, preserving high data locality wastes a large amount of time waiting. [1]

III. SYSTEM ARCHITECTURE

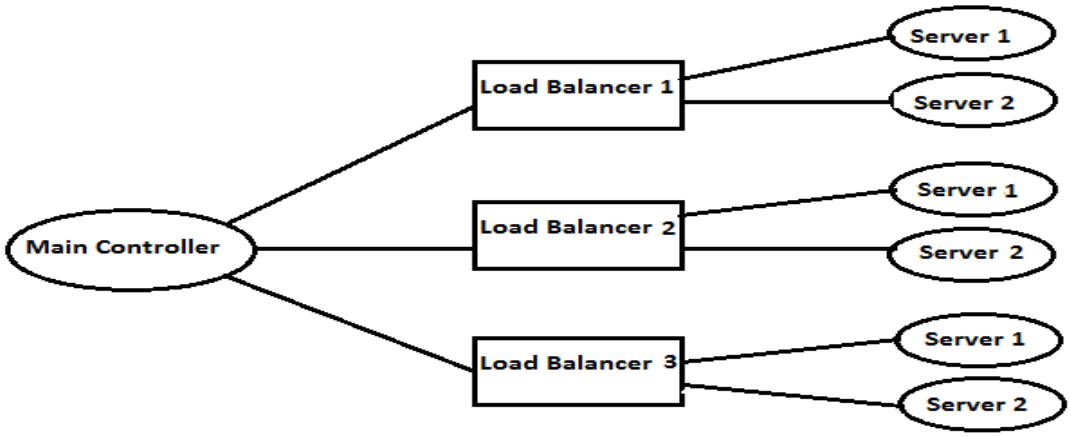

Fig 1. System Architecture

Public Cloud has a large network where the resources are shared via internet and hence it contains the Main controller, Load balancers and servers that performs task.[3]

Main Controller receives task from different locations in public cloud. This tasks then allocated to the load balancers by considering the specific partitions with the help of partitioning algorithms like Round Robin, ESCE.Servers are then performs the task and give result to the main controller in effective way.

Load balancing model follows the below procedure:

Receives task from Different Clients.

3.1 Assign task to load balancer and then to servers by observing the status of server:

A. Under loaded
B. Normal
C. Over loaded

3.2 If the server is over loaded, task will not assign to that server and it will temporarily save into buffer of main controller.

3.3 As soon as the server gets free it will inform to main controller and new task is then assign to specific server.

MODULES:

1. **Task Preparation :**

   In this Module, we will give input as Text File. This Text File then divided into half of the partitioning such that the divided parts are assigning to all the nodes. This Divided file is output for the next Module. This modification is done in a semantics preserving manner so that the normal execution of the program is not affected.

2. **Task Scheduling:**

   In this, Task is scheduled and performed. It will take input from the last module and then give to performer for process. Half of the task is done by one Performer and remaining half will be done by other performer. This is The Task Scheduling.

3. **File Allocation:**

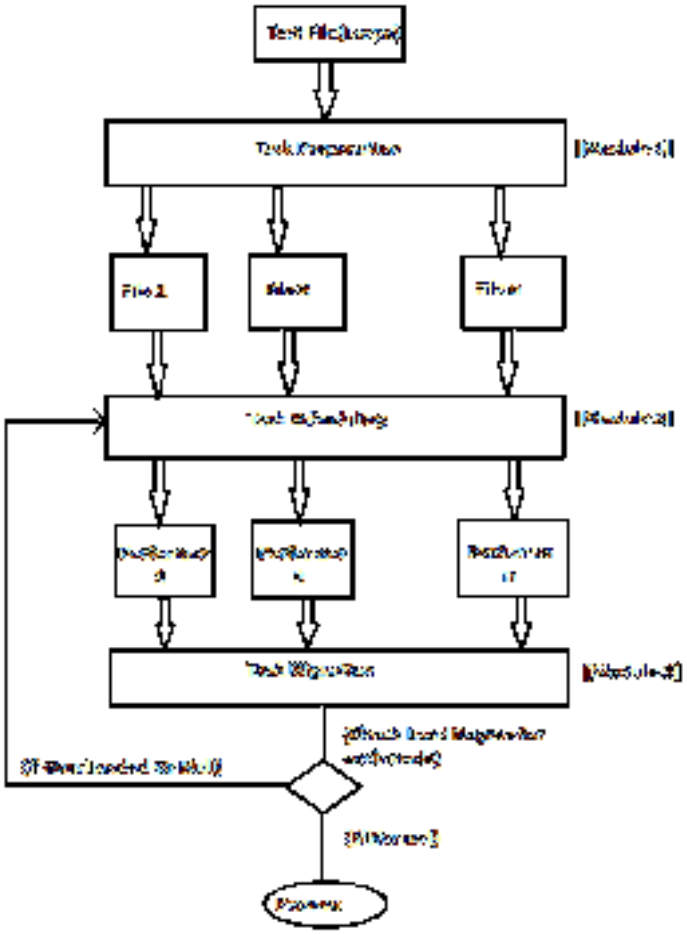

Fig 2. File allocation

IV.CONCUSION

Overall Purpose of this Model is to balance the load on the Public Cloud. It improves the Response time of the tasks that could be performed by server by reducing the Completion time of tasks. This will also help to dynamically allocate jobs to the least loaded servers. It has a backup plan in case of system failure. Hence we will construct a Load Balancing model for better usage of resources and better performance of cloud services.

IV. ACKNOWLEDGEMENT

It is our privilege to acknowledge with deep sense of gratitude to our project guide, Ms. V. M. Barkade for her valuable suggestions and guidance throughout our course of study of our project titled Load balancing Architecture for Public Cloud.

REFERENCES

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

[2] Tiwari et al., International Journal of Advanced Research in Computer Science and Software Engineering 4(2), February - 2014, pp. 807-812.

[3] Divya et al., International Journal of Advanced Research in Computer Science and Software Engineering 4(3), March - 2014, pp. 626-630.

[4] 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.

[5] B.Adler, Load balancing in the cloud: Tools, tips and techniques, Load-Balancing-in the Cloud.pdf, 2012.