Journal of Theoretical and Applied Information Technology
20th May 2016. Vol.87. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DYNAMIC LOAD BALANCING FOR CLOUD PARTITION IN PUBLIC CLOUD MODEL USING VISTA SCHEDULER ALGORITHM

[1]MANISHANKAR S, [2]SANDHYA R, [3]BHAGYASHREE S

[1]Assistant Professor, Department of Computer Science, Amrita Vishwa Vidyapeetham, Amrita University, Mysuru, India

[2, 3] P G Scholars, Department of Computer Science, Amrita Vishwa Vidyapeetham, Amrita University, Mysuru, India

E-mail:[1]manishankar1988@gmail.com, [2]sandyaramesh93@gmail.com, [3]bhagyashreeshekhar@gmail.com

ABSTRACT

Larger the information technology grows, larger will be the data generated, balancing the huge data is always a big question for the information management industry. Cloud computing uses virtual data storage and infrastructure which manages stores and processes huge volume of data. As the cloud offers major services to the clients, it is very important to balance the incoming requests. The aim of the system is to propose an efficient scheduling algorithm and to achieve optimal load balancing. The algorithm address the challenges faced with cloud load management by selecting best cloud partition for workload management, incorporating a novel selection and scheduling algorithm with an assignment problem principle approach for scheduling known as VISTA scheduling algorithm and achieving optimized solution with minimum utilization of processing metrics.

Keywords *:- Load Balancing, Public Cloud, Cloud Partition, Assignment Problem, Vista Scheduling Algorithm*

1. INTRODUCTION

The need for a reliable, scalable and flexible environment led to a new era of computing that is cloud environment. NIST [1] defines cloud computing as a model for enabling convenient on demand network utilization to a shared pool of configurable dedicated resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. It is an agile innovative and performance based system which utilizes of virtualization technology in mainstream. Due to supporting aspects such as easy management, cost reduction, uninterrupted services, disaster management and green computing [2], every organization wants to deploy a cloud environment for its booming and hassle free operation through the internet.

Progressively as the organizations migrate to cloud environment, there is an exponential increase in data and resource monitoring and framework for management of the data generated. It is the responsibility of cloud service provider to overcome and solve issue that arises in public cloud model dynamically.

A cloud computing environment can be classified based on deployment type as private cloud, public cloud, community cloud and hybrid cloud. One of the most dominant forms of cloud based model is Public cloud. [1] Defines public cloud as an infrastructure which is provisioned to be used by the general public, which can be managed or owned by any organization or institution. It is managed and owned by a cloud service provider. Public cloud model is popular for the easy access and simplicity features for very large clients.

The need for efficient management of the resources deployed for dedicated purposes and data management is done by load balancing. Load balancing can be defined as the capability of processors to schedule them to ensure that all the processors and resources are kept engaged while the instruction stream is active or until it is available. Through load balancing the client requests are scheduled to the dedicated resources

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and the requests are executed. Figure 1 shows the performance of load balancer in public cloud model.

Based on the strategy of load balancing, it can be of two types' static load balancing and dynamic load balancing.

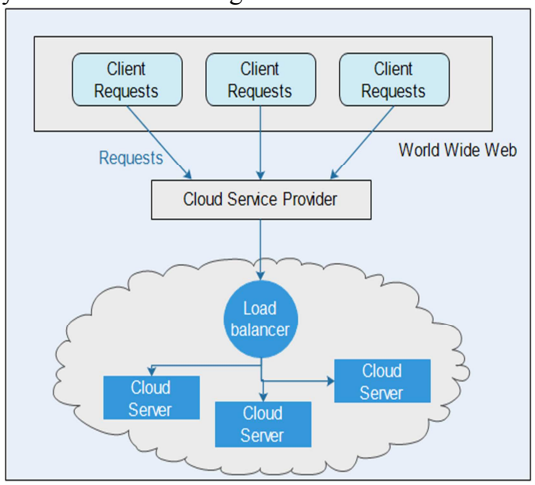

Figure 1: Load Balancer in Cloud Environment

In static balancing method the available resources and parameters such as processing metrics, network bandwidth and CPU and server capacity etc, is considered initially and later the scheduling of workload is accomplished. Some of the popularly used static scheduling algorithms include round-robin algorithm, central manager algorithm, threshold algorithm [3] etc., Whereas in dynamic load balancing, based on the client requests and on the size of jobs the load balancer chooses the suitable and optimal resources required for performing the job. If the workload fluctuation happens in mid way of a process execution, the allocated resources can be still refreshed to perform the new state of the job, it includes algorithms like central queue, local queue and least connection algorithm [3]. Though static load balancing is simpler and faster for moderate workload, it fails to perform with a constantly changing workload.

2. RELATED WORK

In this sub division, we discuss various contributions for load balancing in cloud computing. Various existing algorithm contribution are also discussed.

[4] Discuss various load balancing techniques which focus on priority based scheduling, virtualization and optimization of cloud service model. It gives a brief description of available load balancing techniques. Available algorithms include round robin which is improvised in Central Load Balancing Decision Model (CLBDM), Load Balancing Min Min technique (LBMM), and Ant colony optimization algorithm. It mainly proposes load balancing algorithms and techniques in cloud for efficient resource utilization.

[5] Describes various algorithms proposed to resolve the issues of load balancing and task scheduling in cloud computing. The comparison shows that static load balancing algorithms are steadier than dynamic algorithms but due to capability of performing accurate in distributed systems, dynamic load balancing is chosen over static load balancing algorithms. This analysis further can also help in designing new load balancing algorithms.

[6] Discuss load balancing in computing environment, distributed system and message oriented model. An overview of transaction metrics and activity based costing is given with a scenario of hospital management system the features of availability of cloud services and data retrieval system is examined and also resource utilization is managed.

[7] Provide the issues and the limitations of conventional load balancing technique implemented in cloud environment and also propose a scalable architecture for cloud by providing a generalized framework for building the cloud environment. An overview of best practices of Amazon web service is given in an intention to support the scalable load balancing architectures.

[8] Reviews issues related to the development of dynamic load balancing algorithm for distributed system. It provides a guide to critical issues that need to be addressed for the development and study of dynamic load balancing algorithm and it also tries to perform evaluation based on certain metrics.

3. SYSTEM MODEL

Public Cloud infrastructure consists of a number of computing resources such servers, data centre etc, which may be located in various parts of the world. In our work we consider dynamic load balancing strategy to perform efficient load balancing with respect to issues such as cost effectiveness, scalability, flexibility and priority of jobs by considering the best available cloud partition to perform the client request. By using assignment problem methodology, best cloud partition to perform the allocated job is selected.

Cloud partitioning approach helps to select which is the most optimal resource to complete client request. A cloud partition is a conceptual region of the public cloud with divisions based on

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the geographic locations. Every cloud partition consists of servers and other resources (Child nodes), which are monitored and managed by a sub node. The entire sub node is supervised by a main node, which in turn maintains a status table in order to know the state of each cloud partition. The setup of the cloud model is shown in the Fig.2.

Main node maintains a status table which is dynamically refreshed and the state of the sub node is uploaded. Based on the client request arrival, the main node selects the best partition to do the job and schedules the job to the sub node. The status of every partition can be any of the 3 conditions; they are 1.Lightly loaded 2.Moderately loaded and 3. Heavily loaded.

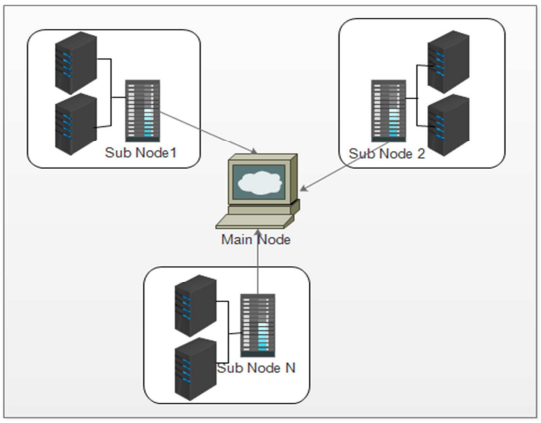

Figure 2: Model For Partition Based Cloud Model Public Cloud Model

If the status of the partition is loaded lightly or moderately then the partition can handle the workload assigned by the main node but if the status of partition is heavily loaded, then the scheduler has to queue the jobs to assign the job to the best partition or has to choose other partition which is not heavily loaded. This explains the basic model of the public cloud model with cloud partitioning approach. Next concern is load balancing and scheduling of jobs in public cloud.

4. LOAD BALANCING AND JOB SCHEDULING METHODOLOGY

The public cloud model must be mainly based on the quality metrics such as resource utilization, scalability, availability, fault tolerance, throughput and overhead management, all these are the challenges of load balancing that need to be achieved [4].

The public cloud model considers various cloud partitions while balancing the load and scheduling the tasks.

The main strategy for efficiently balancing the load is done by the selection of the best cloud partition. The selection process is done by partition selection algorithm

Algorithm 1:

Partition Selection Algorithm (job j, N, P_i, S [P], MN)

begin

jobs ($j_1, j_2, j_3 \ldots j_n$) → MN

MN computes $Load\_Degree_N$ and $Load\_Degree_{AVG}$

while ($j_i \in P_i$)

if S [P_i] == lightly loaded/moderate

Assign job j_i to P_i

else

choose other available partitions

end if

end while

end

Once the jobs or client request $j_1, j_2, j_3 \ldots j_n$ arrives at MN and MN computes $Load\_Degree_N$ and $Load\_Degree_{AVG}$ using the parameters from equations (1) and (2) respectively. If the job j_i belongs to partition P_i and the status of the partition S [P_i] is lightly loaded or moderate then the job is assigned to the respective partition from where it arrived otherwise other partition is selected for completion of the request.

$$\mathbf{Load\_degree(N_i)} = \sum_{i=1}^{m} \alpha_i F_i \qquad (1)$$

Equation (1) Provides load degree of a particular selected node where, N is any selected node form the partition,F_I= (F_1, F_2, F_3 …F_N) is a set of differing load parameters. And average load degree of all child nodes present in a partition is given in (2). Based on average of load degree from (2) the $Load\_degree_{HIGH}$ is calculated.

$$\mathbf{Load\_degree_{AVG}} = \sum_{i=1}^{n} \frac{\mathbf{Load\ degree(N_i)}}{\mathbf{n}} \qquad (2)$$

This acts as parameter to fix on status of the partition whether it is lightly, moderately or heavily loaded. After the selection of partition, the next task is by sub node to balance the node within the cloud partition. Load degree of each node in the partition must be defined. Load degree is calculated based on static and dynamic parameters. Number of CPU's, processing speeds, the RAM size forms the static parameters whereas the ratio of memory and resource utilization and network strength which cannot be rigid are considered as dynamic parameters[9].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\text{Load\_degree} \leq \text{Load\_degree}_{AVG} \quad (3)$$

$$\text{Load\_degree}_{HIGH} \geq \text{Load\_degree} \quad (4)$$

If (3) follows then the status of the partition is considered as either lightly or moderately loaded but, if it follows (4) then considered as heavily loaded. The load degree of the cloud partition is finally updated to the sub node of the cloud partition.

Load Balancing for lightly loaded nodes: During low peak hours the load on the cloud will be very less; in this scenario the status of the cloud partition will be idle or lightly loaded then computing resources may be easily available so at this situation we use Balancer scheduler Algorithm
Main node maintains a status table which comprises of status of all cloud partitions. Based on the status, it assigns job to sub nodes, progressively which schedules the jobs to processors for accomplishment. Status table is refreshed and updated periodically. This algorithm is particularly to select the cloud partition and not for the sub nodes to schedule the job for child nodes.

Algorithm 2:
Balancer Scheduler Algorithm (job j, N, P_i, S [P], MN, SN, AT)
begin
For j_1 to N upto N.length
Assign Jobs ($j_1, j_2, j_3 \ldots j_n$) → Sub Node ($j_1, j_2, j_3 \ldots j_n$) →que_{SN} [ ]
Based on the arrival time AT, jobs are scheduled to the processors
Update Status table ()
According to status table, the processor whose assigned job is completed will be freed and made available for considering a new job from the queue que_{SN} [ ]
End

As the jobs assigned by the main node arrive at the sub node, the scheduling of jobs to the processors is carried. The entire job requests are kept in the queue and the processor are assigned and the processor whose assigned job is completed will be freed and made available for considering a new job from the queue thus the processor will not be idle for any instant of time. It need not wait for the previously assigned job to be completed to consider the new job from the queue. It is illustrated in Figure 3, where j1, j2 … jn were assigned to the processors P1, P2 … Pn. If Processor Pi completes the job assigned it can consider a new job from job queue without waiting for other processors completing the previous jobs from the queue.

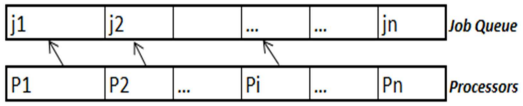

Figure 3: Illustration of balancer Scheduler Algorithm

Load Balancing for moderately loaded nodes using VISTA Scheduler Algorithm: If the partition status is moderately loaded, then we can assume that few processors are engaged in other jobs and few processors may be free. In this situation there is a requirement of an efficient assignment algorithm.

The VISTA scheduler algorithm provides an optimal solution with minimum cost consumption. If there are 'n' available processors, the processing capacities may take variable time to complete the same assigned job. VISTA scheduler algorithm solves these circumstances of best assignment. It can be rooted from [10] which provide an assignment problem approach of Hungarian Algorithm to provide the available optimal solution for any problem. But in current scenario it is altered to consider best processor to accomplish the client request. Figure 4 shows the flow of control for VISTA scheduler algorithm.

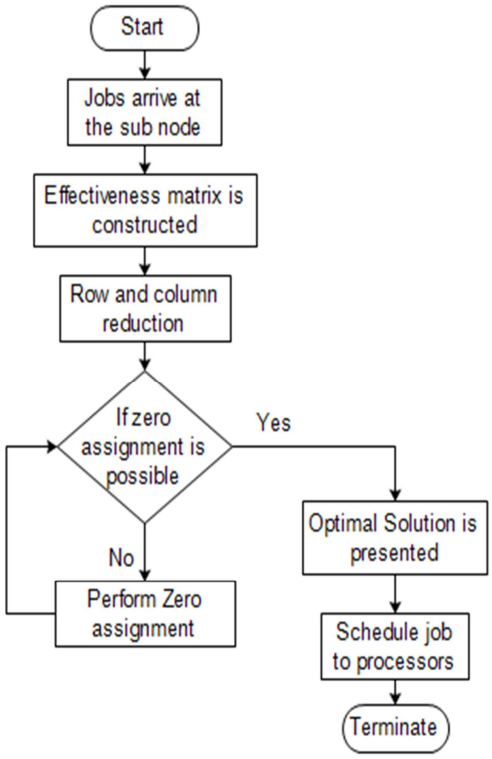

Figure 4: Flow of Control for Vista Scheduler Algorithm

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The jobs that need to be scheduled within the partition arrive at the sub node of the cloud partition. The sub node schedule the jobs to the available resources, an effectiveness matrix is constructed, which consists of processors and jobs as rows and columns. Row and column reduction of the matrix is performed based on least value reduction. Then zero assignment is carried on the reduced matrix. After zero assignment the matrix element which consists of zero is scheduled for the processor and it is carried until all the jobs are assigned to processors. Thus VISTA algorithm helps to reduce cost of processing and increases efficiency.

5. IMPLEMENTATION ANALYSIS

In this section we discuss the implementation of the above algorithm in order to full fill the client request for a file. A framework of the public cloud model with the cloud partition is considered. The setup is developed according to the previously described methods and proposed algorithms for Selection of the best partition in the cloud, Scheduling of the jobs at the sub nodes and assignment of the jobs to processors using VISTA algorithm. Assuming a scenario of a client requesting for file from file server located in public cloud and execution of the request by successfully with the requested file is depicted below.

Main node computes the Load Degree at File Server. Load degree is calculated on parameter including CPU utilization, RAM utilization and Network bandwidth. Once the load degree of the partition estimated it will be sent to Balancer, through a TCP connection.

The partition status at balancer can be ranged as 0% - 30% for lightly loaded, 31% - 70% for moderately loaded and greater than 70% as heavily loaded. Based on the load degree generated the status table maintained at the sub node is updated and status is further transferred to the main node.

As the status table is updated job assignment at the main node can be completed. At this phase client request for file can be received through TCP. Selection algorithm selects the best available partition for fulfilling the request. The request is then forwarded to the sub node.

Job assignment at the sub node must be made through two cases: *Case1:* If status of the partition is lightly loaded or idle then Algorithm2 i.e. scheduling algorithm does the job scheduling and *Case2:* If status of the partition is moderately loaded then Assignment Problem algorithm should be implemented. Once the successful scheduling of jobs is done the job assignment at the sub node is completed and the respective file server is requested for transferring the file.

Thus the jobs were assigned to the best possible optimally available processors based on the load degree of the cloud partitions; the request for the file from the client is served from the file server

This approach provides the optimal solution for performing load balancing of jobs at the server's side through efficient resource management and utilization without excluding any resource left unutilized during peak hours.

Figure 5 shows the load balancing performance with the usage report of CPU, RAM and Network Usage. The status table with varying load and partition status with data rate of transmission in Bytes/Sec with a refresh rate of 'n' seconds is given in figure 6. It also shows based on varying workload how the cloud adapts to changes in workload.

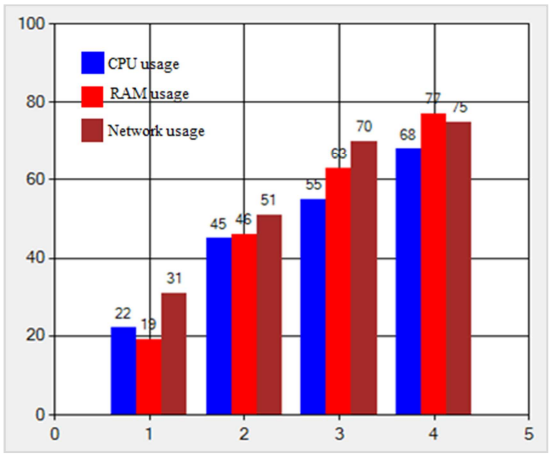

Figure 5: Cloud Status Report of Resource Utilization

6. CONCLUSION

Load balancing is an important aspect in concern to performance of any cloud environment. Cloud partitioning approach helps in efficient management of cloud resources. The work is mainly focused to present an efficient dynamic load balancing algorithm for a public cloud model. The proposed algorithm helps to distribute load based on various aspects of the cloud environment to provide best optimal cloud partition to complete the job. This in turn does the effective resource utilization and management. It reduces the cost of processing. Performance issues are enhanced but availability and security issues can be improved in future.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Available Servers with System Status (Status table)			
ServerIpAddress:Port	LoadDegree	MaxDatarate(Bytes/Sec)	CurrentDatarate(Bytes/Sec)
192.168.0.104:6060	Idle - 24%	10485760 Bytes/Sec	7969177 Bytes/Sec
192.168.0.104:6060	Normal - 47%	10485760 Bytes/Sec	5557452 Bytes/Sec
192.168.0.104:6060	Normal - 62%	10485760 Bytes/Sec	3984588 Bytes/Sec
192.168.0.104:6060	Overloaded - 78%	10485760 Bytes/Sec	2831155 Bytes/Sec

Figure 6: Status Table At The Balancer

REFERENCES

[1] Mell P, Grance T. The NIST definition of cloud computing.

[2] Jadeja Y, Modi K. Cloud computing-concepts, architecture and challenges. InComputing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on 2012 Mar 21 (pp. 877-880). IEEE.

[3] Nuaimi KA, Mohamed N, Nuaimi MA, Al-Jaroodi J. A survey of load balancing in cloud computing: Challenges and algorithms. InNetwork Cloud Computing and Applications (NCCA), 2012 Second Symposium on 2012 Dec 3 (pp. 137-142). IEEE.

[4] Desai T, Prajapati J. A Survey Of Various Load Balancing Techniques And Challenges In Cloud Computing. International Journal of Scientific & Technology Research. 2013 Nov 25;2(11):158-61.

[5] Beniwal P, Garg A. A comparative study of static and dynamic load balancing algorithms. Int J Adv Res Comput Sci Manag Stud. 2014;2:12.

[6] Chaczko ZC, Mahadevan V, Aslanzadeh S, Mcdermid C. Availability and load balancing in cloud computing. InInternational Conference on Computer and Software Modeling IPCSIT 2011, 2011. IACSIT Press, Singapore, http://www. ipcsit. com/vol14. htm.

[7] Kumar IP, Kodukula S. A Generalized Framework for Building Scalable Load Balancing Architectures in the Cloud. IJCSIT) International Journal of Computer Science and Information Technologies. 2012;3(1):3015-21.

[8] Alakeel AM. A guide to dynamic load balancing in distributed computer systems. International Journal of Computer Science and Information Security. 2010 Jun;10(6):153-60.

[9] Xu G, Pang J, Fu X. A load balancing model based on cloud partitioning for the public cloud. Tsinghua Science and Technology. 2013 Feb;18(1):34-9.

[10] Kuhn HW. The Hungarian method for the assignment problem. Naval Research Logistics (NRL). 2005 Feb 1;52(1):7-21.

[11] Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M. A view of cloud computing. Communications of the ACM. 2010 Apr 1;53(4):50-8.

[12] Fox A, Griffith R, Joseph A, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I. Above the clouds: A Berkeley view of cloud computing. Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS. 2009 Feb 10;28(13):2009.

[13] Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems. 2009 Jun 30;25(6):599-616.

[14] Vecchiola C, Pandey S, Buyya R. High-performance cloud computing: A view of scientific applications. InPervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on 2009 Dec 14 (pp. 4-16).IEEE.