Implementation Paper Of Balancing A Load In Cloud Environment By Using Partitioning Techniques

A.A.Deshmukh[1], Prof. R.L.Paikrao[2]
[1, 2] Department of Computer Engineering,
[1, 2] AVCOE, Sangamner, Maharashtra

Abstract *- Balancing the load in the cloud computing environment has an important effect on the performance of the server. Better load balancing makes cloud computing more systematic and improves user gratification. Paper introduces a better load balance model for the public cloud based on the cloud partitioning concept with a shift or switch mechanism to choose different criteria for different situations. The proposed algorithm applies the game theory to the load balancing criteria to improve the efficiency in the public cloud environment.*

Keywords- load balancing model; public cloud; cloud partition; game theory

I. INTRODUCTION

Balancing the load model defined at the public cloud which has many more nodes with distributed computing methods 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 select the suitable partitions for incoming jobs while the balancer for each cloud partition select the best load balancing criteria. Cloud computing is novel and fast growing technology in the field of computer science and engineering. Researchers says that the cloud will bring changes to the software industry. The cloud is changing our life by providing users with new types of services like SaaS, PaaS, etc. Users get service from a cloud without paying attention to the details. Cloud computing is 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 little management effort or service provider interaction. Cloud computing is efficient and scalable but maintaining the stability of processing so many jobs in the cloud computing environment. So it is a very complex and crucial problem with balancing the load that receiving much attention for researchers. Since the job arrival pattern is not inevitable and the capacities of each node in the cloud differ for load balancing problem, workload control is complex to improve system performance and maintain stability. Load balancing schemes depending on whether the system can be either static or dynamic. Static schemes do not use the system information and are less complex than dynamic scheme is used here for its flexibility. he model has a main controller and balancers to collect 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 criteria as well as strategy.

An important strategy for energy diminution is concentrating the load on a subset of servers and, whenever possible, switching the rest of them to a state with low energy consumption. This observation suggested that the traditional concept of load balancing in a large-scale system could be again formulated as follows: distribute evenly the workload to the smallest set of servers operating at optimal or near-optimal energy levels, while observing the Service Level Agreement (SLA) between the CSP and a cloud user. An optimal energy level is one when the performance per Watt of power is maximized.

In order to integrate business requirements and application level needs, in terms of Quality of Service (QoS), cloud service provisioning is regulated by Service Level Agreements (SLAs): contracts between clients and providers that express the price for a service, the QoS levels required during the service provisioning, and the penalties associated with the SLA violations. In such a context, performance evaluation plays a key role allowing system managers to evaluate the effects of different resource management strategies on the data center functioning and to predict the corresponding costs/benefits.

II. RELATED WORK

R.Hunter[1], in this paper they introduce an energy-aware operation model used for load balancing and application scaling on a cloud. The basic idea is defining an energy optimal operation control and attempting to maximize the number of servers operating in this control. Idle and lightly loaded servers are switched to one of the sleep states to save the energy. The load balancing and scaling algorithms also used for some of the most affecting features of server consolidation mechanisms discussed in this literature.

M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis,and A. Vakali[2], One vision of 21st century processing is that clients will get to Internet benefits over lightweight - convenient gadgets instead of through some relative of the conventional desktop PC. Since clients won't have (or be keen on) capable machines, who will supply the registering power? The response to this question lies with distributed computing. Distributed computing is a late pattern in IT that moves registering and information far from desktop and convenient PCs into extensive server farms. It alludes to applications conveyed as administrations over the Internet and additionally to the genuine cloud framework to be specific, the equipment and frameworks programming in server farms that give these administrations. The key main impetuses behind distributed computing are the pervasiveness of broadband and remote systems administration, falling capacity expenses, and dynamic upgrades in Internet figuring programming. Cloud benefit customers will have the capacity to include more limit at pinnacle request, diminish costs, explore different avenues regarding new administrations, and expel unneeded limit, though specialist co-ops will build usage by means of multiplexing, and take into consideration bigger interests in programming and equipment.

P. Mell and T. Grance[3], Cloud figuring is a model for empowering advantageous, on-request arrange access to a common pool of configurable processing assets (e.g., systems, servers, stockpiling, applications, and administrations) that can be quickly provisioned and discharged with negligible administration exertion or specialist organization communication.This cloud display advances accessibility and is made out of five fundamental attributes, three administration models, and four arrangement models. Deepak Gupta and Pradip Bepari, In this paper they shows a distributed system consists of possibly heterogeneous, computing nodes connected by a communication network. Such a system can be used effectively only if the software presents a single system. Thus all resources of any node should be easily and transparently accessible from any other node. One of the most important of such resources is the CPU. The CPUs of all the nodes in the system can be made transparently available to all nodes if the nodes share their computing load. In this manner the framework ought to choose the best hub to execute any employment paying little mind to where the occupation began and may even relocate a few employments amid their execution. This calls for straightforward process movement among the hubs. In this paper, they examine the issues in such load sharing. There are two orthogonal sorts of issues to be tended to here. The main kind identify with the arrangements for movement. For instance, where another employment ought to be executed, when and where an executing procedure ought to be moved and so forth. The second sort of issues identify with the systems for movement. For instance, how to checkpoint and exchange the condition of a running procedure, can a procedure be moved to a machine with an alternate design, can old projects be moved without adjustments and so forth.

B. Adler[7], Load balancing is a method to distribute workload across one or more servers, network interfaces, hard drives, or other computing resources. Typical data centre implementations rely on large, powerful (and expensive) computing hardware and network infrastructure, which are subject to the usual risks associated with any physical device, including hardware failure, power and/or network interruptions, and resource limitations in times of high. APKS based on HPE over an encrypted data.

Z. Chaczko, V. Mahadevan, S. Aslanzadeh, and C. Mcdermid,[9] "Cloud computing" is a term, which involves virtualization, distributed computing, networking, software and web services. A cloud consists of several elements such as clients, datacenter and distributed servers. It includes fault tolerance, high availability, scalability, flexibility, reduced overhead for users, reduced cost of ownership, on demand services etc. Central to these issues lies the establishment of an effective load balancing algorithm. The load can be CPU load, memory capacity, delay or network load. Load balancing is the process of distributing the load among various nodes of a distributed system to improve both resource utilization and 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. Load balancing ensures that all the processor in the system or every node in the network does approximately the equal amount of work at any instant of time. This technique can be sender initiated, receiver initiated or symmetric type (combination of sender initiated and receiver initiated types). Our objective is to develop an effective load balancing algorithm using Divisible load scheduling theorem to maximize or minimize different performance parameters (throughput, latency for example) for the clouds of different sizes.

III. PROBLEM DEFINITION

"The Problem Statement is Server gets overloaded due to excess of request from different host then it goes into the rest mode. So by using Server consolidation properties transfer the request from sleep mode to running mode on other server, Use Auto scaling and peak energy level"

IV. SYSTEM ARCHITECTURE

There are three primary and main contributions of this proposed paper:

1) A novel of cloud servers that is based on different operating controls with various degrees of energy efficiency.
2) novel algorithm that balancing the load and application scaling to maximize the number of servers operating in the energy-optimal control and analysis and comparison of techniques for load balancing and application scaling using three differently-sized clusters and two different average load profiles.

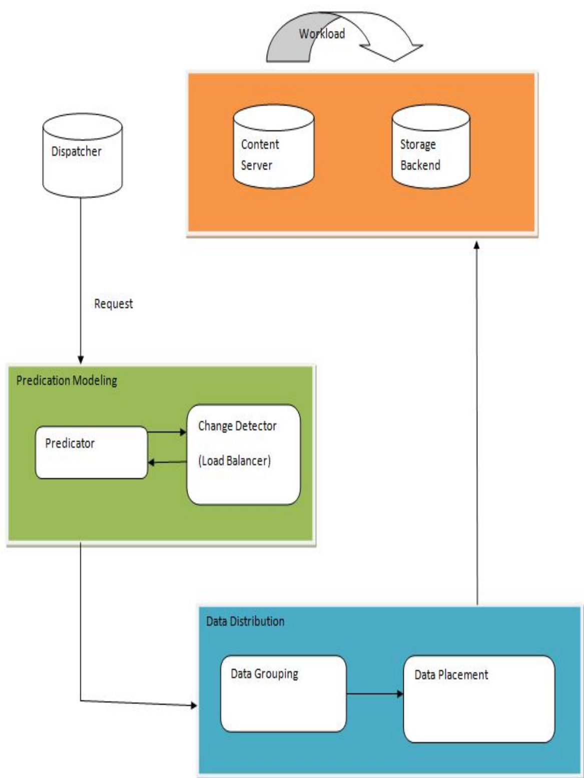

Figure 1. System Architecture

3) The main Moto of the algorithm is to guard that the largest possible number of active servers operate within the boundaries of their respective optimal operating control. The actions implementing this policy are: (a) Relocate the VMs from a server operating in the undesirable-low control and then switch the server to a rest or sleep state. (b) Switch an idle server to a rest or sleep state and reactivate servers in a sleep state when the cluster load increases. (c) Relocate the VMs from an overloaded server, a server operating in the undesirable-high control with applications divine to increase their demands for computing in the next reallocation cycles.

V. MATHEMATICAL MODEL

Input

U(Z) = u1,u2,u3.......un
F(Z) = f1,f2,f3.....fn
S(Z) = s1,s2,s3
D(Z) = d1,d2,d3....dn
M(Z) = m1,m2,m3....mn
Where
U(Z)= User
F(Z)= File
S(Z)= Server
D(Z)= Data
M(Z)= Migration
U(Z) U F(Z)

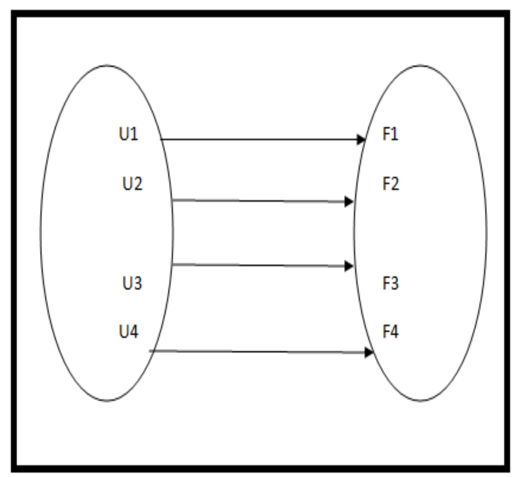

Figure 2. output

U(Z) U F(Z)D(Z)US(Z)

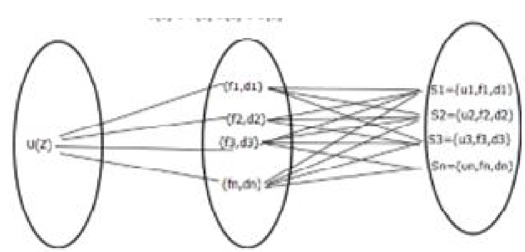

Figure 3. output

VI. ALGORITHMS

Input:

Workload (W) →{w1,w2,w3.....}
Resource(R) →{R1,R2,R3...}

Output:
Migration List (M) →{m1,m2,m3...}

ENERGY EFFICIENT ALGORITHM:

1) Start
2) Extract Total workload list
3) Access total Resource list
4) User upload file
5) Check first cloud server work load
6) 6) Limitation of server depends on energy level
7) If server is go to energy threshold
8) File migrate to another server HOT SPOT Process.
9) Check remaining server workload.
10) Find Min workload Resources optimization
11) Manage workload of every server Green Computing.
12) Check energy level
13) End

Resource Allocation Algorithm :

Input:

v, type of vm request on the basis of size.

Output:

a resource allocation scheme.

Step 1 : While V do
Step 2 : Satisfied Flag false;
Step 3 : Other Flag false
Step 4 : For each started server
Step 5 : if(util[t] < minThUtil || newutil < upThUtil)
Step 6 : Place the request on the server
Step 7 : Satisfied flag true
Step 8 : Update the utilization and remaining capacity of server t
Step 9 : If (Satisfied flag false)
Step 10 : Other flag true
Step 11 : Start a new server tnew
Step 12 : Started status[tnew] truel
Step 13 : Place the vm request on tnew
Step 14: Satisfied flag true
Step 15: Update the utilization and remaining capacity of tnew
Step 16: Add the tnew to the list of started servers.

VII. EXPERIMENTAL SETUP

In this the Structure consist of technologies like JAVA , HTML , CSS , Java script. For back end MySQL is used. Hence before investigational set up Software like Eclipse, Tomcat is predictable to be installed on server. User must have basic windows Family, good browser to view the results. Supervised Dataset or Un-Supervised dataset is used for testing in MySQL is tested.

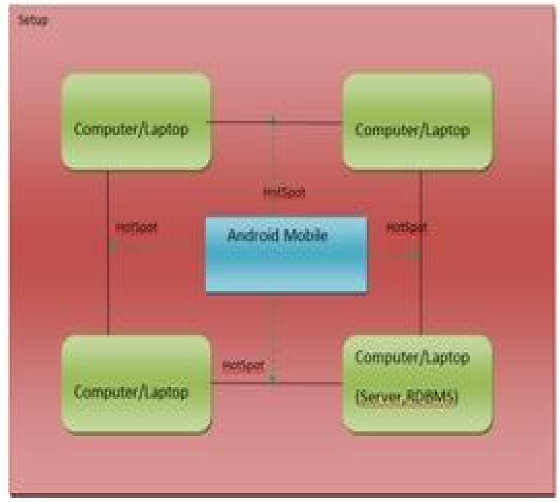

Figure 4. Experimental Setup

VIII. RESULTS AND DISCUSSIONS

Table 1. Result Analysis

Load (%)	0	10	20	30	40
P-Power(W)	92	584	1296	1712	1895
T-Transaction	492	1039	1204	1620	1803
Efficiency(T/P)	0.18	0.56	1.07	1.05	1.05

P = Power how much consume time to execute process in nanosecond

T = Transaction Time Average efficency maintain (E)

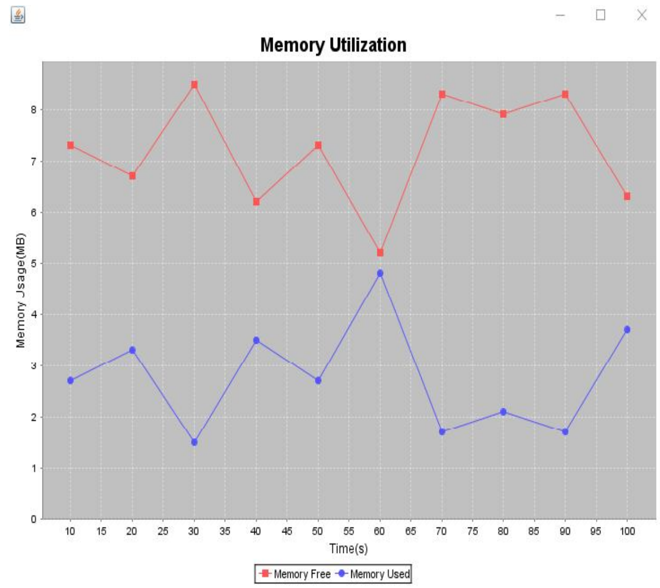

Figure 5. Memory Utilization

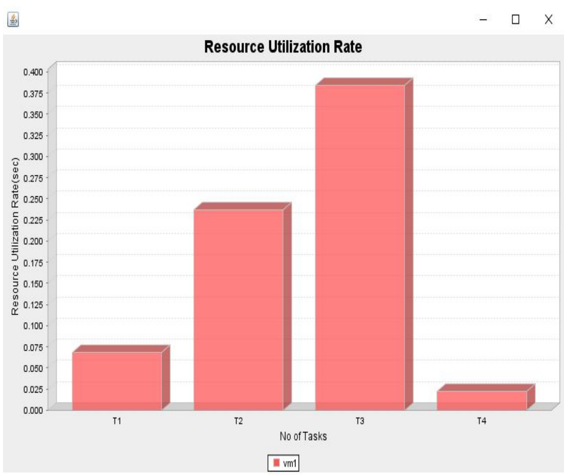

Figure 6. Resource Utilization Rate

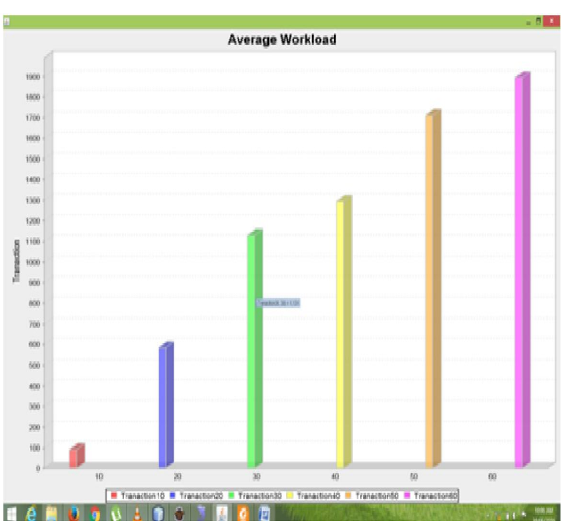

Figure 7. Average workload

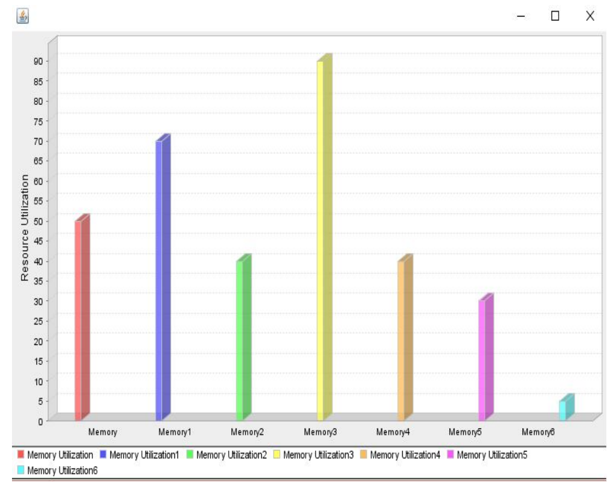

Figure 8.

IX. CONCLUSION

Finally Conclude that By considering the resources like requests on network, memory of requested data and CPU utilization of server and by checking and analyzing such a resources, we can balance the load by migrating or switching the workload to the different server and get the optimal efficiency for improving the throughput and minimize the cost of the system using the Green computing that uses the auto scaling by the server.

ACKNOWLEDGMENT

A very firstly I gladly thanks to my project guide and HOD of computer department Mr.R.L.Paikrao for her valuable guidance for implementation of proposed system. I will forever remain a thankful for their excellent as well as polite guidance for preparation of this report. Also I would sincerely like to thank to other staff for their helpful coordination and support in project work. I also thankful to our principal Dr. M. A. Venkatehsh.

REFERENCES

[1] R.Hunter,"Thewhyofcloud",http://www.gartner.com/DisplayDocument?doc cd=226469ref= g noreg, 2012.

[2] 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.

[3] P. Mell and T. Grance, "The NIST definition of cloud computing", http://csrc.nist.gov/ publications/nistpubs/800- 145/SP800-145.pdf,2012

[4] Microsoft Academic Research, ”Cloud computing”,http://libra.msra.cn/Keyword/6051/cloud-computing?query= cloud 20computing,2012.

[5] Google Trends, Cloud computing, http://www.google.com/trends/ exploreq=cloud 20computing,2012.

[6] N.G.Shivaratri,P.Krueger,andM.Singhal,”Loaddistributin gforlocally distributed systems”, Computer, vol. 25, no. 12, pp. 33-44, Dec. 1992.

[7] P. Gaochao Xu, X. Fu ;“A Load Balancing Model Based on Cloud Partitioning for the Public Cloud; SSN 1007-0214 04/ 12 pp34-39 Volume 18, Number 1, February 2013

[8] B. Adler, ”Load balancing in the cloud: Tools, tips and techniques”, http://www.rightscale.com/infocenter/whitepapers/Load-Balancing-inthe-Cloud.pdf, 2012..

[9] Z.Chaczko,V.Mahadevan,S.Aslanzadeh,andC.Mcdermid, ”Availability and load balancing in cloud computing”, presented at the 2011 International Conference on Computer and Software Modeling, Singapore,2011

[10] 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.