Miss.Rudra Koteswaramma / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 6, November- December 2012, pp.167-171

Client-Side Load Balancing and Resource Monitoring in Cloud

Miss.Rudra Koteswaramma M.Tech
#Assistant .Professor, DepartmentofComputer scienceAnd Engineering

ABSTRACT

Load Balancing is a method to distribute workload across one or more servers, network interfaces, hard drives, or other computing resources. Typical datacenter 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 demand. Load balancing in the cloud differs from classical thinking on load-balancing architecture and implementation by using commodity servers to perform the load balancing. This provides for new opportunities and economies-of-scale, as well as presenting its own unique set of challenges.

Keywords- ***IaaS, PaaS , SaaS, Cloud Computing, Load balancing***

I. INTRODUCTION

Cloud computing means that nstead of allthe computer hardware and software we are using sitting on our desktop, or somewhere inside our company's network, it's provided for us as a service by another company and accessed over the Internet, usually in a completely seamless way. Exactly where the hardware and software is located and how it all works doesn't matter to us, the user—it's just somewhere up in the nebulous "cloud" that the Internet represents[11].

Cloud computing is a buzzword that means different things to different people. For some, it's just another way of describing IT (information technology) "outsourcing"; others use it to mean any computing service provided over the Internet or a similar network; and some define it as any bought-in computer service you use that sits outside your firewall.

There is no standard definition of Cloud computing. Generally it consists of a bunch of distributed servers known as masters, providing demanded services and resources to different clients known as clients in a network with scalability and reliability of datacenter. The distributed computers provided On-demand services. Services may be of software resources (e.g. Software as a Service, SaaS) or physical resources (e.g. Platform as a Service, PaaS) or hardware/infrastructure (e.g. Hardware as a Service, HaaS or Infrastructure as a Service, IaaS).

II. TYPES OF CLOUD

Based on the domain or environment in which clouds are used, clouds can be divided into 3 categories: **Public Clouds-**It is type of cloud which can be access from anywhere in the world and can be accessed by anyone. Examples of this cloud are Amazon's or Google's cloud which are open to all after specific SLA between user and provider.

Private Clouds -In this type of cloud the specific organization's or company's employee can only get access and it will be accessible only within organization's premises and by authenticating each and every user, it is not open to all.

Hybrid Clouds (combination of both private and public clouds)-This types of cloud are combination of both public as well as private cloud. Most of the commercial use is influenced by this type of cloud. There are three different kinds of services provide by cloud computing, where different services are being provided for the user

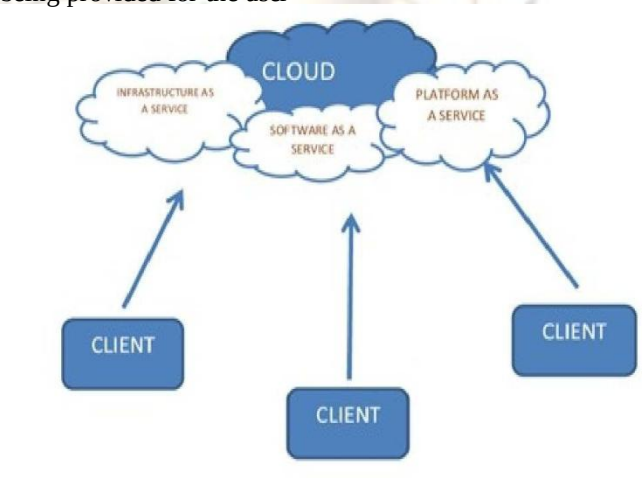

Fig. 1 Different services provided by cloud

means weare buying access to raw computing hardware over the Net, such as servers or storage. Since we buy what you need and pay-as - you-go, this is often referred to as utility computing. Ordinary web hosting is a simple example of IaaS:

wepayamonthlysubscriptionorapermegabyte/giga bytefeetohaveahost ingcompanyserves up files for our website fromtheirservers.

B. **Software as a Service (SaaS)** :

means we use acomplete application running on someone else's system. Web-based email

and Google Documents are perhaps the best-known examples. Zoho is another well-known SaaS provider offering a variety of office applicationsonline.

C. **Platform as a Service (PaaS)** :

means we developapplications using Web-based tools so they run on systems software and hardware provided by another company. So, for example, we might develop your own ecommerce website but have the whole thing, including the shopping cart, checkout, andpaymentmechanismrunning on a merchant'sserver.Force.com(fromsalesforce.comant he
Google App Engine are examples of PaaS.

When the technology increasesthe complexities also increases with that. So in cloud computing when nodes (computer resources) increases in the cluster it becomes a challenging job to monitor the resources as they all integrated and they all made to perform different task at run time. And also it becomes a challenge to us, to shift the load on one node to another depend on the scenario. In our proposed model we first establish a cloud using two nodes.Then we monitor the resources like CPU,RAM, Harddisc etc and alsobyUsingtheloadbalancingalgorithm we design a system thatautomaticallybalances the load and shifts the control to another node in the cluster.

III. OUR APPROACH TO THE PROBLEM

In our proposed model we establish cloud setup between two computers using Ubuntu, xen and Eucalyptus on peer to peer network. This can be discussed as follows-

1. Cloud Setup - Creating cloud (test bed) by using (Ubuntu, Xen and Eucalyptus

2. Resource Monitoring - monitoring critical resources like RAM, CPU, memory, bandwidth, partition information, running process information and utilization and swap usages etc.

3. Load Balancing - load balancing algorithm for homogeneous and heterogeneous architectures.

4. Testing - In order to evaluate the performance of complete setup, need to deploy resource monitoring and load balancing tools on test bed and evaluate performance of our algorithm.

A. Why Resource Monitoring?

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.

What happens when the services,servers, and Internet applications on which we rely on run into trouble, suffer downtime, or otherwise don't perform to standard? How quickly will we notice and how well will we react? Cloud monitoring allows us to track the performance of the cloud services we might be using. Whether we are using popular cloud services such as Google App Engine, Amazon Web Services, or a customized solution, cloud monitoring ensures that all systems are going. Cloud monitoring allows us to follow response times, service availability and more of cloud services so that we can respond in the event of any problems.

B. Approach to Resource Monitoring

Here in this section we are developing an application in java where we aremonitoring the node resources like RAM, CPU, Memory, Bandwidth, Partition information, Running process information and utilization.

C. What is Load Balancing?

Load Balancing is a technique in which the workload on the resources of a node is shifts to respective resources on the other node in a network without disturbing the running task. A standard way to scale web applications is by using a hardware-based load balancer [5]. The load balancer assumes the IP address of the web application, so all communication with the web application hits the load balancer first. The load balancer is connected to one or more identical web servers in the back-end. Depending on the user session and the load on eachweb server, the load balancer forwards packets to different web servers for processing. The hardware-based load balancer is designed to handle high-level of load, so it can easily scale.

However, a hardware-based load balancer uses application specific hardware-based components, thus it is typically expensive. Because of cloud's commodity business model, a hardware-based load balancer is rarely occurred by cloud providers as a service. Instead, one has to use a software based load balancer running on a generic server.

A software-based load balancer [8, 12, 1] is not a scalable solution, though. The scalability is usually limited by the CPU and network bandwidth capacity of the generic server that the load balancer runs on and a generic server's capacity are much smaller than that of a hardware-based load balancer. For example, in our test [11], we found that an Amazon EC2 instance can handle at most 400 Mbps

combined ingress and egress traffic.

Even though some cloud platforms, such as Google App. Engine [7], implicitly over a hardware-based load balancer, we cannot easily get around their limitations because of the limited control we have. In our test, Google App Engine is only able to handle 10 Mbps in/out or less traffic because of its quota mechanism.

HTTP protocol [6] has a built-in HTTP redirect capability, which can instruct the client to send the request to another location instead of returning the requested page. Using HTTP redirect, a front -end server can load balance traffic to a number of back- end servers. However, just like a software load balancer, a single point of failure and scalability bottleneck still exists.

D. Approach to Load Balancing

Here in this load balancing task we will set a threshold value for each resource. And we run a thread to monitor the resource load. Once this load crosses its threshold value then we first gather all information about process task and then shift that task to another node's resource without disturbing running task. The Algorithm we are going to use is Signature Driven Load management for Cloud Computing Infrastructure.

IV. IMPLEMENTATION

a) Module number 1

The SigLM algorithm first selects the different resources like RAM, hard disk space, etc to monitor and set some threshold value to each and every resources through which we can divert the load to another node present in the cloud. Fig.2 shows the flow diagram of it.

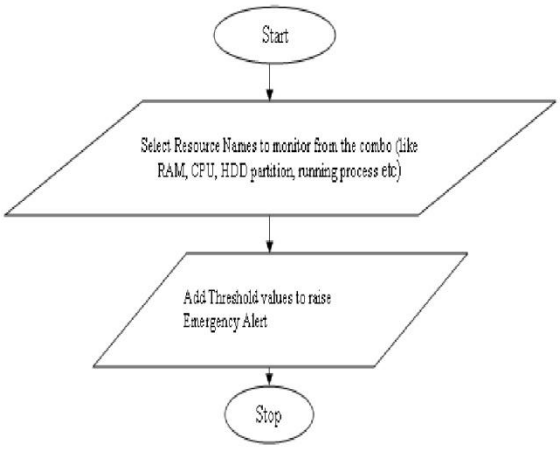

Fig. 1.2 Module number 1

b) Module number 2

In this module we are submitting jobs to the cloud, that job is intended to be submitted to the different nodes present in cloud, by checking the threshold value of each and every node decision will be taken and next module get called. Fig.3 shows the flow diagram of it.

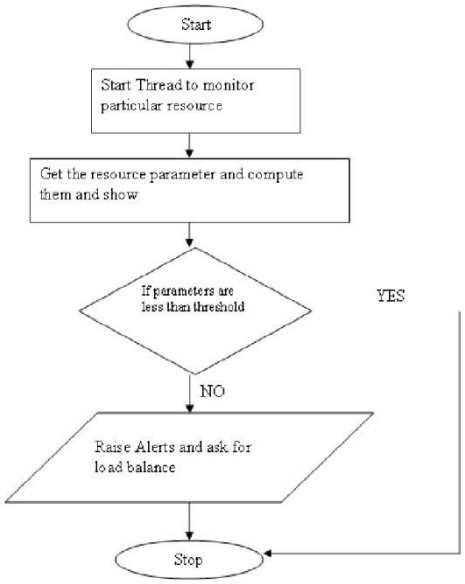

Fig. 3 Module number 2

c) Module number 3

In module number 3 threads which are ready to be submitted are checked by or load balancing algorithm along with that it also verifies the threshold value of the node as well as threshold value of the upcoming load if it satisfied the request will be forwarded to module 4 otherwise that request will get declined. Fig.4 shows the flowchart of it.

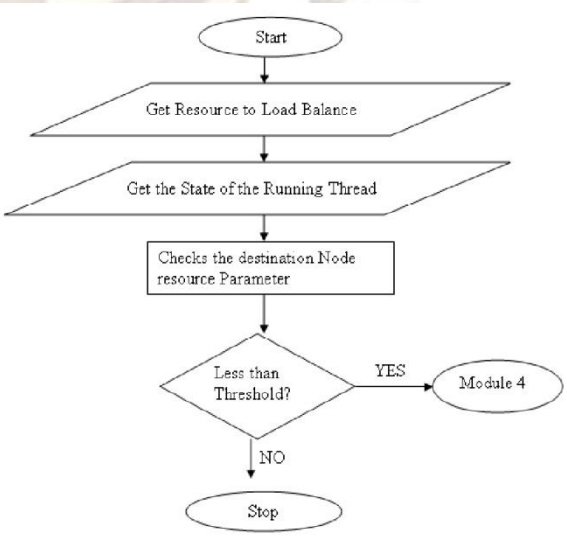

Fig. 4 Module number 3

d) Module number 4

This module checks the signature of the task and node if it matches it calculate the Dynamic time Wrapping and according to it, it is forwarded to module 5, if DTW [5] does not match the task will get declined. Fig.5 shows the flow diagram of it

Fig. 5 Module number 4

e) Module number 5

Balancing using Signature DTW Algorithm [14], it creates thread on node, and creates DSN thread and put it on wait, after that it connects source node thread to destination thread by calling RMI and in this way complete execution of task. After completion of task it is submitted to the Module 6. Fig.6 shows the flow diagram of it.

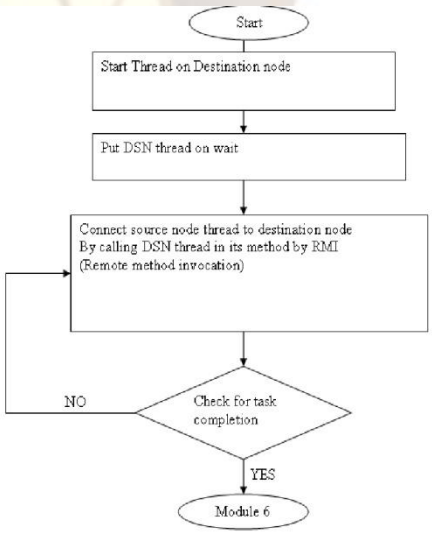

Fig. 6 Module number 5

f) Module number 6

It frees the threads, by pulling source node thread back to the source and put it on to sleep. It also resumes DSN thread and stop the execution.Fig.7 shows the flow diagram of it.

V. CONCLUSION

In this paper I created private Cloud setup using Ubuntu, xen and Eucalyptus and that we use as a test bed for carrying out implementation of DTW algorithm. We also did literature survey of existing resource monitoring tools as well as load balancing tools and come up with an algorithm for different architecture with better performance.

In this paper we discuss the implementation modules of Signature pattern matching DTW algorithm with the proper flow diagrams which simplifies the work of Load Balancer. The proposed metrics could be further refined by taking more detailed formalism for each module.

REFERENCES

[1] Tony Bourke: ***Server Load Balancing***, O'Reilly, ISBN 0-596-00050-2

[2] Chandra Kopparapu: ***Load Balancing Servers, Firewalls & Caches***, Wiley, ISBN 0-471-41550-***2***

[3] Robert J. Shimonski: ***Windows Server 2003 Clustering & Load Balancing***, Osborne McGraw-Hill, ISBN 0-07-222622-***6***

[4] Jeremy Zawodny, Derek J. Balling: ***High Performance MySQL***, O'Reilly, ISBN 0-596-00306-4

[5] J. Kruskall and M. Liberman. The Symmetric TimeWarping Problem: From Continuous to Discrete.
In Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison, pp.

125-161, Addison-Wesley Publishing Co., 1983.

[6] Matthew Syme, Philip Goldie: ***Optimizing Network Performance with Content Switching: Server, Firewall and Cache Load balancing''***, Prentice Hall PTR, ISBN 0-13-101468-***5***

[7] Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing A Practical Approach, TATA McGRAW-HILL Edition

[8] 2010.Martin Randles, David Lamb, A. Taleb-Bendiab, A Comparative Study into Distributed

[9] Load Balancing Algorithms for Cloud Computing, 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops. Mladen A. Vouk, Cloud Computing Issues, Research and Implementations, Proceedings of the ITI 2008 30th Int. Conf. on Information Technology Interfaces, 2008, June 23-26.

[10] Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed

[11] http://www.amazon.com/gp/browse.html?node=201590011

[12] Amazon Elastic Compute Cloud http://aws.amazon.com/ec2/.

[13] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh. Indexing Multi-Dimensional Time-Series with Support for Multiple Distance Measures. ***Proc. of SIGKDD***, 2003.

[14] Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. ***Journal of Knowledge and Information Systems***, 2004.