Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 1, January 2015, pg.539 – 543

REVIEW ARTICLE

A REVIEW ON CLIENT SIDE LOAD BALANCING

Prof. Vikas Nandgaonkar, Prof.Prashant Dongare, Harshal Mahajan, Awadhoot Lele, Akshay Gaikwad

University of Pune, Pune

Abstract: Load balancing is an important issue while managing server resources in a cloud environment. The concept behind load balancing is to manage server load which includes number of resources like available RAM, CPU bandwidth, etc. as well as to manage incoming request on the server. In cloud environment, it is important that even small application requests from clients must be served with an appropriate response, but in conventional approach, it becomes difficult to serve small data resources over large ones. Here load balancing plays an important role by managing and distributing load from one server evenly across multiple servers. Our approach is to perform load balancing at client side which means to shift load management process at client side hence reducing server's load balancing overhead.

Keywords— string matching, experimental algorithms, text processing, automaton, pattern

I. INTRODUCTION

Cloud computing may be a new term within the computing world and it signals the appearance of a brand new computing. Cloud computing is Associate in Nursing on demand service within which shared resources, data, computer code and alternative devices area unit provided in step with the purchasers demand at specific time. It's a term that is mostly employed in case of web. the complete web is viewed as a cloud. Capital and operational prices is cut victimisation cloud computing. Load equalisation in cloud computing systems is absolutely a challenge currently. Continually a distributed resolution is needed. Jobs can't be appointed to acceptable servers and purchasers separately for economical load equalisation as cloud may be a terribly complicated structure and elements area unit gift throughout a good unfold space. Our aim is to produce Associate in Nursing analysis and comparative study of those approaches.

Cloud computing could be a bunk meaning totally different things to different individuals. For some, it's simply in our own way of describing IT (information technology) "outsourcing"; others use it to mean any computing service provided over the Internet or an identical network; and a few outline it as any bought-in laptop service you utilize that sits outside your firewall.

Different 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.

Different services provided by Cloud --

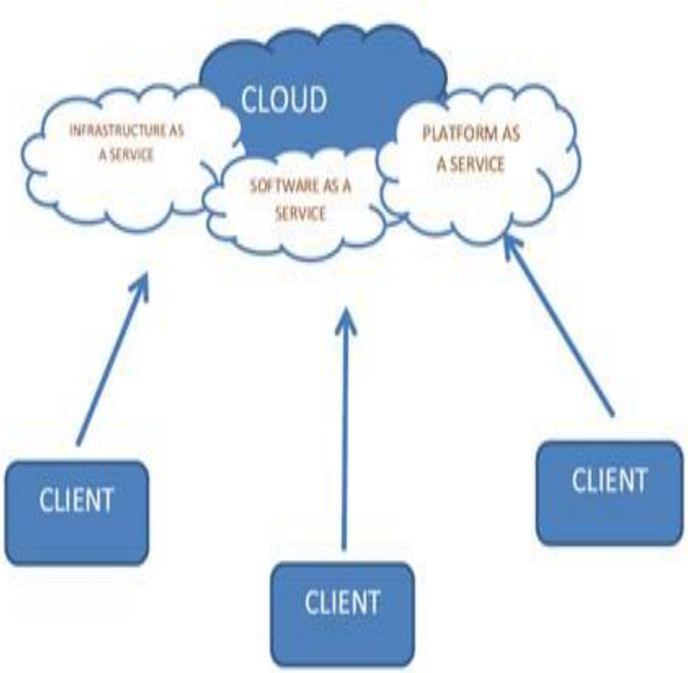

Fig 1: Services of cloud.[1].

A. **Infrastructure as a Service (IaaS) -** Means we have a tendency to area unit buying access to raw computing hardware over world wide web, such as servers or storage. Since we have a tendency to get what you would like and pay-as-you-go, this is {often|this can be} often said as utility computing. normal net hosting may be a straightforward example of IaaS: we have a tendency to pay a monthly subscription or a permegabyte gigabyte fee to own a hosting company serves up files for our web site from their servers.

B. **Software as a Service (SaaS) - M**eans we use a complete application running on someone else's system.
Web-based email and Google Documents are perhaps the best-known examples.

C. **Platform as a Service (PaaS) - M**eans we develop applications 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, and payment mechanism running on a merchant's server. Force.com (from salesforce.com) and the Google App Engine are examples of PaaS.

Existing Load Balancing Algorithm -

A. Dynamic Load Balancing Algorithm:

In a distributed system, dynamic load equalization is worn out 2 totally different ways: distributed and non-distributed. within the distributed one, the dynamic load equalization algorithmic program is dead by all nodes gift within the system and also the task of load equalization is shared among them. The interaction among nodes to attain load equalization will take 2 forms: cooperative and non-cooperative [4].Dynamic load equalization algorithms of distributed nature, typically generate additional messages than the non-distributed ones as a result of, every of the nodes within the system must move with each alternative node. A benefit, of this can be that though one or additional nodes within the system fail, it'll not cause the overall load equalization method to halt, it instead would effects the system performance to some extent. Distributed dynamic load equalization will introduce Brobdingnagian stress on a system within which every node must interchange standing info with each alternative node within the system. In non-distributed kind, either one node or a gaggle of nodes do the task of load equalization. Non-distributed dynamic load equalization algorithms will take 2

forms: centralized and semi-distributed. within the initial kind, the load equalization algorithmic program is dead solely by one node within the whole system: the central node. This node is exclusively chargeable for load equalization of the entire system. The opposite nodes move solely with the central node. In semi-distributed kind, nodes of the system square measure partitioned off into clusters, wherever the load equalization in every cluster is of centralized kind. A central node is nonappointive in every cluster by acceptable election technique that takes care of load equalization at intervals that cluster.
Hence, the load equalization of the entire system is completed via the central nodes of every cluster[4].

Strategies in Dynamic Load Balancing:
1)Transfer Policy: The part of the dynamic load balancing algorithm which selects a job for transferring from a local node to a remote node is referred to as Transfer policy or Transfer strategy.

2) Selection Policy: It specifies the processors involved in the load exchange (processor matching) .

3) Location Policy: The part of the load balancing algorithm which selects a destination node for a transferred task is reffered to as location policy or Location strategy.

4)Information Policy: The part of the dynamic load balancing algorithm responsible for collecting information about the nodes in the system is reffered to as Information policy or Information strategy.

B. Distributed Load Balancing For the Clouds:

(a)Honeybee Foraging Algorithm:
In load-balancing operation,[2] every server takes a specific bee role with possibilities post exchange or pr. These values area unit wont to mimic the bee colony whereby an explicit range of bees area unit maintained as foragers – to explore (px); instead of as harvesters – to take advantage of existing sources. A server with success fulfilling asking can post on the advert board with likelihood pr. A server might at random select a virtual server's queue with likelihood px(exploring), otherwise checking for an ad (watching a waggle dance). In summary, idle servers (waiting bees) follow one in every of 2 behaviour patterns: a server that reads the advert board can follow the chosen advert, then serve the request; therefore mimicking harvest behaviour. A server not reading the advert board reverts to forage behaviour; pairing a random virtual server's queue request. associate degree corporal punishment server can complete the request and calculate the profit of the just-serviced virtual server.

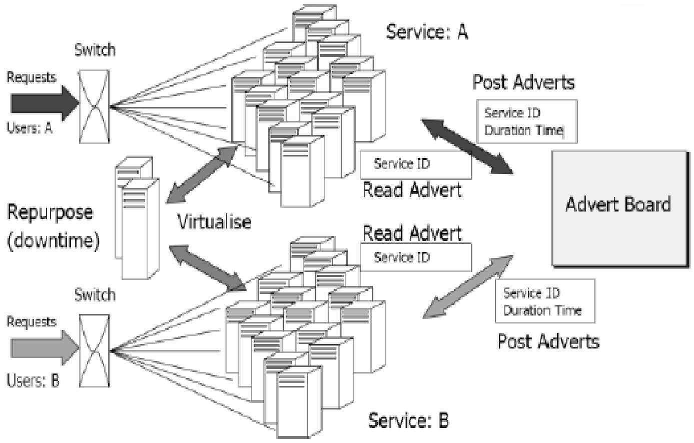

Fig 2 :Virtual Servers and Advert Boards[2]

II. Problem Statement

To develop scalable, secure and fault tolerant client side load balancing application to leverage strength of cloud components[1] by using signature driven load management algorithm along with dynamic time wrapping[3].

Proposed System

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. What is Resource Monitoring?

Cloud computing has become a key manner for businesses to manage resources, that square measure currently provided through remote servers and over the web rather than through the recent hardwired systems that appear therefore out of date nowadays. Cloud computing permits corporations to source some resources and applications to 3rd parties and it means that less problem and fewer hardware in an exceedingly company. rather like any outsourced system, though, cloud computing needs watching. What happens once the services, servers, and web applications on that we tend to have faith in run into hassle, suffer period, or otherwise don't perform to standard? however quickly can we tend to notice and the way we tend toll can we react? Cloud watching permits America to trace the performance of the cloud services we would be victimisation. whether or not we tend to square measure victimisation in style cloud services like Google App Engine, Amazon net Services, or a made-to-order answer, cloud watching ensures that every one systems square measure going. Cloud watching permits America to follow response times, service accessibility and a lot of of cloud services in order that we are able to respond within the event of any issues.

B. Approach to Resource Monitoring

Here during this section we tend to area unit developing Associate in Nursing application in java where we tend to area unit observance the node resources like RAM, CPU, Memory, Bandwidth, Partition data, Running method data and utilization by employing a Third Party merchant application like SIGAR (System data Gatherer and Reporter).

<u>Proposed Algorithm</u> ---

- Client side load balancing system which leverages strength of cloud components and overcomes above mentioned disadvantages

Signature Driven Load Management (SigLM) using Cloud

The above algorithm works by capturing system's signature like available RAM, current CPU bandwidth available and other resources. Once captured, that value is compared with default threshold value and accordingly load like incoming requests is shifted to target node machine using Dynamic Time Wrapping (DTW) technique.

Dynamic time wrapping works by considering source node as given by SigLM algorithm and makes some calculations to predict target node to which the load is to be shifted.

This algorithm has better results than conventional algorithms with following advantages :

- Caption of resource signature.
- Scheduling by comparing signature of each server.
- 30%-80% improved performance than existing approaches
- Scalable, efficient and 0.0% overhead
- Dynamic time wrapping (DTW) for selection of target node at runtime.
- Client side means to perform load balancing <u>before</u> requests hit to server.

Conclusion:

In this paper we tend to created non-public Cloud setup mistreatment Ubuntu, xen and Eucalyptus which we tend to use as a workplace for closing implementation of DTW algorithmic program. we tend to conjointly did literature survey of existing resource observation tools additionally as load leveling tools and are available up with Associate in Nursing algorithmic program for various design with higher performance.

In this paper we tend to discuss the implementation modules of Signature pattern matching DTW algorithmic program with the right flow diagrams that simplifies the work of Load Balancer. The planned metrics may be any refined by taking a lot of elaborate formalism for every 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 & LoadBalancing*, 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 TimeWarpingProblem: From Continuous to Discrete. *In Time Warps,String Edits and Macromolecules: The Theory and Practiceof Sequence Comparison, pp. 125-161, Addison-WesleyPublishing Co.*, 1983.

[6] Matthew Syme , Philip Goldie: *Optimizing Network Performancewith Content Switching: Server, Firewall and Cache Loadbalancing"*, 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 Edition22*International Journal of Advances in Computing and Information ResearchesISSN: 2277-4068, Volume 1– No.2, April 2012*

[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 Computer Systems, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.

[11] http://www03.ibm.com/press/us/en/pressrelease/22613.ws

[12] http://www.amazon.com/gp/browse.html?node=20159001

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

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

[15] Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Journal of Knowledge and Information Systems*,2004.23