IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 15, Issue 5 (Nov. - Dec. 2013), PP 69-72
www.iosrjournals.org

Role of Virtual Machine Live Migration in Cloud Load Balancing

Stuti Dave, Prashant Mehta
[1,2]*B H Gardi College of Engineering and Technology-Rajkot, Gujarat, India*

Abstract: *Cloud computing has touched almost every field of the life. Hence number of cloud application consumers is increasing every day and so as the number of application request to the cloud provider. This leads increment of workload in many of the cloud nodes. The motive to use load balancing concepts in cloud environment is to efficiently utilize available resources keeping in mind that no any single system is heavily loaded or not a single system is idle during the active phase of the request completion. Even though cloud computing being a software facility most often, how does it actually performs well in heavily loaded environment at processor level, is discussed in the paper. This paper aims to throw some light on what is cloud load balancing and what is the role of Virtual machine migration in improving it.*

Keywords: *Cloud load balancing, Live Migration, Migration, Virtualization, Virtual machine.*

I. INTRODUCTION

Cloud computing is the Advanced networking concept that lets you access any computer resources available on the internet in form of data, calculations and transparent web services in ubiquitous, convenient, and on-demand network access manner. Thus according to [1] Cloud Computing is a series of service which evolved from web collaboration technologies and models that include distributed processing, parallel processing, and grid computing and so on.

The complete paper is designed in such a way that section II discusses basic cloud environment infrastructure. Section III is based on need of load balancing n cloud environment. Section IV explains virtualization and Virtual Machine (VM). Section V throws light on migration and live migration of VM and section VI concludes the paper.

II. CLOUD ENVIRONMENT

To understand cloud environment, let us consider figure-1 where a customer is asking for some web service response from the internet cloud. Client is the end user who interacts with the cloud to generate requests and get response from it. Basically there are 3 types of client:

1. Mobile client: They are mobile by their nature and basically depend upon the remote server to actually perform the request processing.
2. Thin client: It does not perform complex tasks on the request and hence it often requires support from some other computer systems.
3. Thick client: It performs most of the processing to respond any client request itself.

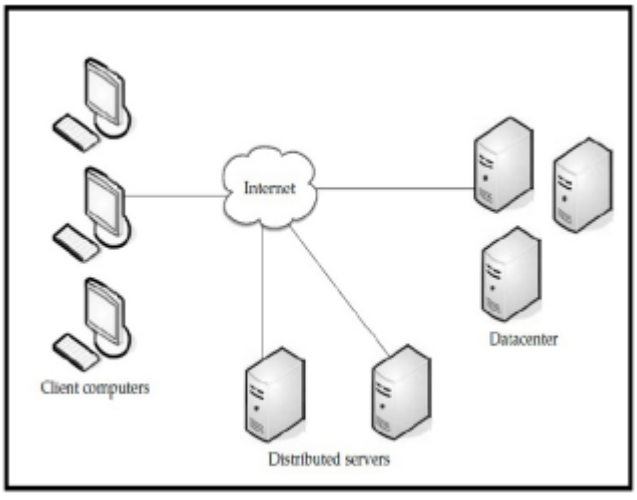

Figure 1 Simple Cloud Environment

A data center is nothing but a collection of server hosting different applications which a client can subscribe to for its required applications. A distributed server is a managing server which continuously checks for its host services provided to the client.
When a client initiates any request it gets the list of available matching services from the data centers from that services client can communicate with the distributed server managing the host running that service or application.

Sometime it happens that a server get lots of requests for a host and the other available host stay idle. This situation affects cloud performance substantially. The Load balancing concept helps to overcome the situation and improve overall performance.

III. LOAD BALANCING

Load balancing is the technique using which incoming client requests are distributed among available hosts efficiently so that no any host is heavily loaded and no any host is idle. As described in [3] Load balancing is often implemented with a view to achieve goals like:

1. To improve system performance drastically
2. To improve stability of the cloud system
3. To introduce scalability to improve cloud performance
4. To improve system working condition under heavy workload or request rate.

According to [4] depending on the application, Load balancing algorithm can be any of the following form:

1. Initiated by Sender: Here algorithm is initiated by request sender itself.
2. Initiated by Receiver: Here algorithm is initiated at application server.
3. Symmetric: It collectively uses the concepts of above two types.

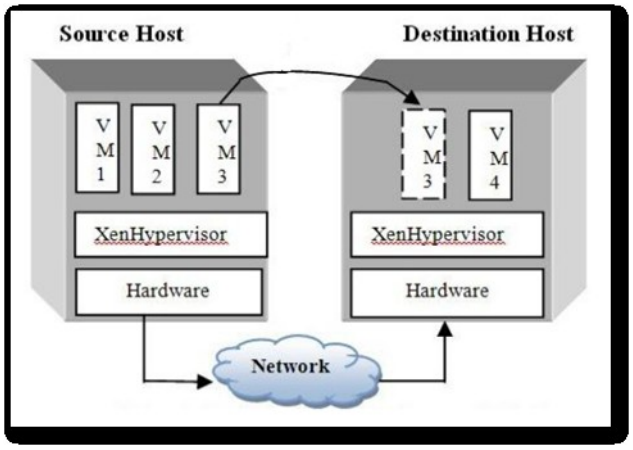

Figure 1 Simple migration process [6]

Dependency of current state, express Load Balancing algorithm in another two types [4]:

1. Static: It does not depend on the current state of the system. Earlier before setting up the request it decides that at which host the request will be executed.
2. Dynamic: Decisions on load balancing depends on current state of the system. The load balancer analyses the current load statistics at each available host and executes request at appropriate host.

Depending on the requirement any of the above type of load balancing logic is used in cloud environment. Although dynamic load balancing technique is better, as compared to static one, as it provides load balancing solutions based on current system scenario rather on the previous data which may lead to wrong load balancing decision. So the key process of such dynamic load balancing algorithm is discussed in following sections.

IV. VIRTUALIZATION AND VIRTUAL MACHINES

The main goal for which cloud computing came into existence was to share the cloud resources among the cloud consumers, cloud vendors and even cloud partners [10]. The few promising characteristics of cloud computing hence involve transparent resource distribution, efficient extendibility and virtualization. Out of all of them virtualization plays lead role in helping cloud computing to achieve its goal. The reason for this is with virtualization a single physical node handles plenty of Virtual machine running to respond lots of cloud client requests [11]. Consequently Virtualization indirectly contributes much in power aware cloud environment set up also.

In most of the cloud system, to serve more requests at a time and to utilize resources in better manner virtualization is used. In virtualization we do not have the resource physically but we can use virtual form of the device for our requirement fulfillment. Here by resources we mean server, storage device, network or an operating system where the proper division of the available resources provides one or more executing environments. [6] Says that Virtualization provides separation among the physical hardware, that is Computer, and the virtual Software, that is operating system and application, by imitating hardware using software. The virtualization in cloud system is introduced using virtual machines.

Logically cloud performance can be improved by decreasing management costs and utilizing the resources efficiently with the help of Virtualization adopted by virtual machines at the distributed hosts.

V. MIGRATION VS. LIVE MIGRATION

In dynamic load balancing strategy when a host starts getting more number of requests than it can handle, it starts searching for migrating its virtual machine to some other capable and comparatively less busy host in the same cloud. This is called “cold” migration or simply VM migration. The migration decisions are based on the calculation of incoming load, Processor utilizations, and resource availability at the physical host and based on that the host switching off strategy has also been developed to add power awareness as a desirable feature in the cloud infrastructure [11].

This complete process occurs under eyes of Hypervisor or Virtual Machine Monitor (VMM). VMM is a computer hardware, software or firmware which is responsible of generating and managing Virtual Machines. In a single host a VMM may contain one or more Virtual Machines. The OS running in the Virtual machine is called Guest OS. The VMM presents the guest operating systems with a virtual operating platform. The VMM also manages the execution of the guest operating systems of each VM available to that specific host.

The Migration of a VM from an overloaded host to another non critical host makes it possible to improve resource utilization and better load sharing [5]. Even though it is a useful concept, it is an expensive operation too in terms of resource utilization at sender host as well as receiver host and network utilization to transfer system and memory states of the sender host. Once resource manager at the server feels some host exhausted with overload it plans for migration. Basic migration follows steps as shown in fig 3.

Such migration requires that each memory state is stored consistently at application level state and kernel internal state on the target machine. This complete process degrades the cloud performance for a specific amount of time and may disappoint an active user. Even though this technique helps much in improving load balancing in cloud environment its downtime is the main negative effect of the cloud experience.

The live migration overcomes this limit by allowing migration of Virtual machine while it is running. Such process is called "Live" or "Hot" migration. As the live migration relies on the VMM it provides few desirable properties to the cloud infrastructure like improved load balancing, transparent mobility, and high availability services, fault tolerance, application concurrency, and consolidation management.

To live migrate a VM, the VMM pre-copies memory pages of the VM to the destination without disturbing the OS or its applications. The page copying process is executed repeatedly in multiple rounds on which dirty pages are continuously transferred. Subsequently, the VM can be resumed in the new server [9].

Network migration simply requires IP redirection and disk storage redirection can be achieved by storage net – share technology but the memory migration is the root problem of the live migration technique.

Memory migration can be achieved by following the steps very similar to migration as shown below

1. Pre-migration and Reservation
2. Iterative Pre-copy
3. Stop-and-Copy
4. Commitment and Activation

During all these phases VM continuously keeps on running on sender host or destination host and it takes very less time to restart at the destination host (called as downtime).

The downtime in live migration is often very small that even the VM cannot identify it.

Based on migration decision mechanism it has two flavours like, centralized and decentralized. According to the memory copying process, the live migration can be implemented in following flavours:

5.1 Pre-Copy Live Migration:

In this type of live migration, memory transfer phase starts with copying memory pages while the VM is running at the sender host. If in the i^{th} round a memory page is dirtied it is sent again to the destination host in $i+1^{th}$ round of page transfer. This iterative process continuous until, either a reasonable amount of working writable set (WWS) is available at destination host or pre-set number of iterations has been completed. Then during service downtime the dirty page transfer occurs along with process state transfer. And finally VM is activated at destination host and copy of migrated VM at sender side is destroyed. This approach particularly helps to minimize the VM downtime and application degradation [12]. Although this must be kept in mind that this approach works well with read intensive workload, whereas write intensive workload decreases its promising features.

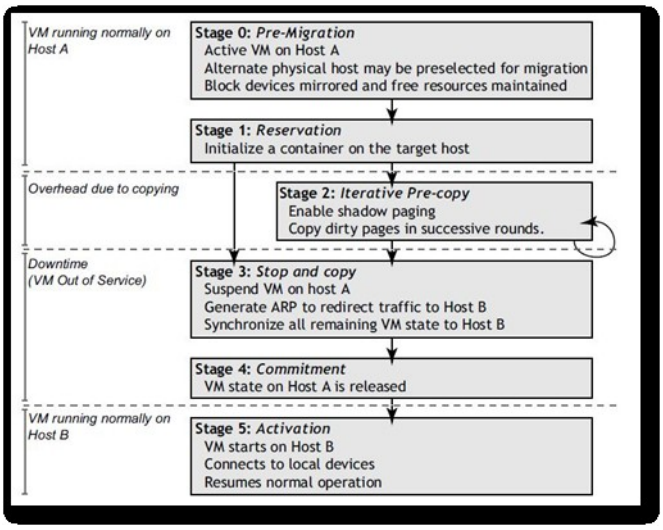

Figure 2 Basic Migration process timeline [7]

5.2 Post-Copy Live Migration:

In contrast with pre-copy live migration, Post-copy live migration first copies sender processor state at destination host and resumes the VM CPU state at the destination host. And after resuming VM at destination host, post-copy live migration technique starts memory page transfer. Concurrently any page which is faulted at destination host

and still has not reached there is demand paged from sender host. Thus in this technique each memory page is transferred at most once to the destination host. This overcomes the limitation of pre-copy method of duplicate transmission overhead [12].

The measuring parameters of any Live VM migration techniques which are used to find out usefulness of any technique can be defined as [13]:

1. Total Migration Time: It indicates the time from when the sender host enters migration process to time when destination host finishes it and starts working normally.
2. Network Traffic Reduction Percentage: The amount of data transfer saved because of the memory page compression and de-duplication of it during live migration
3. Downtime: The time duration up to which the VM is actually suspended to transfer CPU state as well as transfer WWS to the destination host.
4. Application Degradation: it is the extent up to which live migration has slow down the application performance of migrating VM.

VI. CONCLUSION

By using dynamic load balancing algorithms we can surely enhance the cloud performance but the larger downtime introduced due to cold migration process is the disappointing factor of this technique. The live migration is implemented without affecting the request execution process on the application host. Thus VM live migration overcomes the cold migration limitation and utilizes the available resources efficiently. Even in Live migration we have few limitations like dirty page and less applicability in write-intensive applications in pre-copy migration and fatal destination failure after resuming VM at destination in post-copy migration. Thus one must choose the live VM migration technique keeping in mind all the pros and cons of all methods to achieve remarkable performance of the cloud infrastructure in mass incoming load situation.

REFERENCES

[1] Qian Li, Nanqiang Xia, The Utilization of Cloud Computing in Network Collaborative Commerce Chain, Fourth International Conference on Business Intelligence and Financial Engineering, IEEE. 2011.

[2] Soumya Ray, Ajanta De Sarkar, "Execution Analysis Of Load Balancing Algorithms In Cloud Computing Environment*". International Journal on Cloud Computing: Services and Architecture (IJCCSA),Vol.2, No.5*, October 2012

[3] Punit Gupta, Mayank Kumar Goyal, Prakash Kumar. Trust and Reliability based Load Balancing Algorithm for Cloud IaaS, *3rd IEEE International Advance Computing Conference (IACC),* IEEE 2013.

[4] 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.

[5] Mauro Andreolini, Sara Casolari, Michele Colajanni, Michele Messori, Dynamic load management of virtual machines in cloud architectures, *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering Volume 34, Springer. 2010, pp 201-214*.

[6] Varsha P. Patil, G.A. Patil, Migrating Process and Virtual Machine in the Cloud: Load Balancing and Security Perspectives, *International Journal of Advanced Computer Science and Information Technology 2012, Volume 1, Issue 1, pp. 11-19, Article ID Tech-21,* November 2012.

[7] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Live Migration of Virtual Machines, *NSDI '05: 2nd Symposium on Networked Systems Design & Implementation*, 273-286, 2005

[8] Xiaohong Jiang, Fengxi Yan, Kejiang Ye, Performance Influence of Live Migration on Multi-Tier Workloads in Virtualization Environments, *The Third International Conference on Cloud Computing, GRIDs, and Virtualization, IARIA*, 2012.

[9] William Voorsluys, James Broberg, Srikumar Venugopal,Rajkumar Buyya, Cost of Virtual Machine Live Migration inClouds: A Performance Evaluation, *Springer* 2009.

[10] Asma Letaifa, Amel Haji, Maha Jebalia, Sami Tabbane, State of the Art and Research Challenges of new services architecture technologies: Virtualization, SOA and Cloud Computing, *International Journal of Grid and Distributed Computing Vol. 3, No. 4*, December, 2010.

[11] Xiaoying Wang, Xiaojing Liu, Lihua Fan, and Xuhan Jia, A Decentralized Virtual Machine Migration Approach of Data Centers for Cloud Computing, *Mathematical Problems in Engineering Volume 2013, Article ID 878542*, http://dx.doi.org/10.1155/2013/878542, June 2013

[12] Michael R. Hines and Kartik Gopalan, Post-Copy Based Live Virtual Machine Migration Using Adaptive Pre-Paging and Dynamic Self-Ballooning, *ACM 978-1-60558-375-4/09/03 2009.*

[13] Umesh Deshpande, Xiaoshuang Wang, and, Kartik Gopalan, Live Gang Migration of Virtual Machines, *ACM 978-1-4503-0552-5/11/06*, June 2011.