SDN BASED NOVEL VM MIGRATION APPROACH FOR DISTRIBUTED CLOUD ACROSS MULTIPLE DATA CENTERS

[1]Kushali B shah, [2]Prof. Nirav Shah
[1]Student, [2]Assistant Professor
[1]Department of Computer Engineering, [2]Information Technology
[1]Silver Oak College of Engineering & Technology, Ahmedabad, Gujarat, India

Abstract— Distributed computing has turned into a popular aspect in cloud computing that impacts many services and clients. It has turned into a foundation for rising advances and an empowering influence of things to come Internet administrations. Software-based network infrastructure is one of the important type of infrastructure which has enabled the on time modification within network. Ideally, Software defined networks (SDNs) have increased more consideration for substantial scale organizes because of the flexibility and snappiness they offer to the system. Energy consumption is also a considerable criterion that raises much higher cost as it's a continuous requirement for a network to process. Parallel to this rise, the power utilization has quickly expanded. In distributed cloud, which is resided across multiple data centers, creating green and feasible arrangements has turned into a prime concern for the cloud providers. A novel algorithm for a SDN-empowered cloud condition is introduced. The novel algorithm is aimed to manage the complexity of multiple data centers. The proposed method is noticeably minimizing migration time and downtime.

Keywords—Cloud Computing, SDN , VM Migration, Distributed computing, Hybrid Algorithm.

I. INTRODUCTION

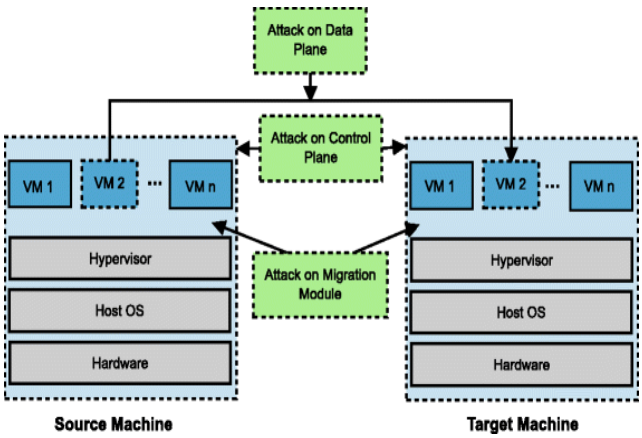
Cloud computing entails you being able to access a virtual machine for you to be able to do what you need to do anywhere. A hypervisor manages these virtual machines. [1]

A hypervisor is a program that would enable you to host several different virtual machines on a single hardware. Each one of these virtual machines or operating systems you have will be able to run its own programs, as it will appear that the system has the host hardware's processor, memory and resources. In reality, however, it is actually the hypervisor that is allocating those resources to the virtual machines. [1]

In effect, a hypervisor allows you to have several virtual machines all working optimally on a single piece of computer hardware.

You may think that the hypervisor is a fairly recent phenomenon. The first hypervisors were introduced in the 1960s to allow for different operating systems on a single mainframe computer. However, its current popularity is largely due to Linux and Unix. Around 2005, Linux and Unix systems started using virtualization technology to expand hardware capabilities, control costs, and improved reliability and security that hypervisors provided to these systems. [2]

Now, hypervisors are fundamental components of any virtualization effort. You can think of it as the operating system for virtualized systems. It can access all physical devices residing on a server. It can also access the memory and disk. It can control all aspects and parts of a virtual machine.



Fig(1). Attacks During Live Migration

Hypervisors may be used in data services for easy cloning and replication. Hypervisor-based replication is also more cost effective and less complex than current replication methods, especially those involving virtual machines.

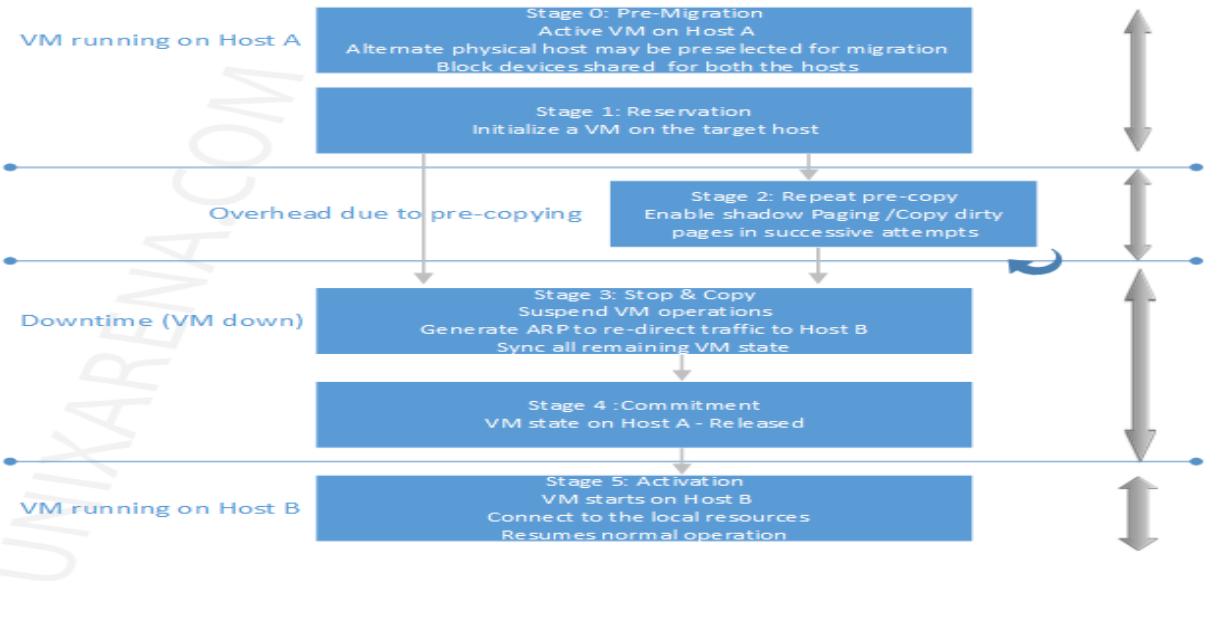
Traditionally, it is very difficult to replicate virtual machines. For one, you would need to know how to manage and do data replication. If you use a storage system-based replication method, you will need to replicate the entire volume of the virtual machine. If you have a lot of virtual machines running on a particular hardware, you will have to get a significant amount of storage in order to be able to store the entire volume. With hypervisor-based replication, you can choose which VMs and what parts are to be replicated, so that you could save up on storage space.

Hypervisor-based replication is also hardware neutral, meaning you could store any data duplicates to any storage device.

If you are currently using VMWare, you can get hypervisor-based replication in vSphere. Microsoft has also included this feature in any Windows package that has Hyper-V.

II. Relevant Work

Migration is the process of moving a virtual machine from one host or storage location to another. Copying a virtual machine creates a new virtual machine. It is not a form of migration. Live migration works with a pre migration stage. And than processes further by following the given stages.[3] The repetitive pre copy and post and copy processes works over downtime. Host waves while it gets the copy of os image successfully as vm. The process stages are as follows



Fig(2). VM Live Migration Timeline

The process of migration starts with selection of destination, with the host having resources available. The available resources are reserved at the destination site. Here it captures whole vm memory and assumes that it is dirty. Until all the memory pages gets copied , it repeats the process of copying all dirty memory pages to destination. Next is to check the condition for stop and copy once it finishes checking all dirty memory pages. Once the process finishes, it will suspend vm and transfer vm state with its cpu state and memory state at destination server. The process resumes when its completed successfully otherwise shows appropriate message.

There are many ways to perform the migration process. Here we can resume the transferring of pages with page fault condition. Furthermore, devising efficient VM migration schemes is also a challenging problem, as it not only requires weighing the benefits of VM migration, but also considering migration costs, including communication cost, service disruption, and management overhead. This proposed approach better techniques that overcomes challenges in data center environments to make the process efficient.

III. Motivation

I have read and analysed many VM migration and scheduling technique by reading papers of several exceptional research papers from the database of many standard journals. In paper[1] the focus is on various traditional migration algorithms which have not performed well in different cases and have not considered other parameters while selecting virtual machine for migration. Some another issues are also involved in migration process like number of migrations, consumption of cost, time and memory. So there is need to develop new approach for load balancing in data centers using VM algorithm that overcome the problems in traditional approaches and improve their performance.[2][3] To solve the problem of too long pre-copy migration time, pre-copy migration methods, including post-copy, time-series based pre-copy method, etc., are therefore proposed. Pre-copy methods can decrease the total migration time but may result in the longer downtime. There are three phases of memory migration: (1) Push Phase, (2) Stop-and-copy Phase, and (3) Pull Phase, There are number of techniques searched and developed to minimize the energy consumption while processing VM migration and load balancing. The results displayed here are performed in C++ simulator that simulates SDN and in cloud controlled environment.[1]

In paper[2], To implement serial migration to improve multiple VM migration techniques. Improved serial migration techniques with post copy migration are proposed to improve timing efficiency for both migration time and down time. The process is based on assumption that the actual number of each VM iteration is n and the number of VMs is m with each VDC request. Paper[3][4] has discussed and highlighted The carried out result trials point out that: (i) the energy savings attained by the proposed bandwidth manager over the state-of-the-art ones currently utilized by Xen, KVM and VMware hypervisors are over 40% and approach 66% under strict QoS constraints; (ii) the proposed bandwidth manager is capable to quickly adapt to the abrupt changes possibly experienced by the dirty rates of the running applications and/or the round trip times of the utilized (possibly, congested) TCP/IP connections; and, (iii) its actual implementation may be carried out in a distributed and scalable way, and it consumes less than 1.5% of the CPU computing power per migrated VM.

VI. Proposed Work

The proposed approach has followed two algorithms to make the hybrid version.

1. Migration Triggered
2. Create target task
3. Start Migration
4. Calculate the load
5. Calculate upper and lower threshold
6. Resume
7. Notify message about completion
8. Update network

V. Proposed Algorithm

1. At the time of migration trigger,
   a. check for input criteria
   b. calculate VM capacity
2. For each task
   a. Calculate load of the task
3. Perform Data center creation, host creation and VM creation process
4. For each VM
   a. Calculate load performed at each VM
   b. Check under load or overload condition based on upper and lower threshold
   c. Select VM based on condition
   d. Maintain matrix of VM for idle and busy VM
   e. Calculate distance between vm and host
   f. Check shortest distance VM from lower threshold matrix in sorted order

5. Select the host to migrate VM based on request processed on FCFS basis
6. Continue until stopping criteria met or task completed

TABLE 1: Results

DC	No.of Vms	Memory	Bandwidth
DC1	40	1024	1000
DC2	20	512	100
DC3	35	512	100
DC4	50	1024	10000
DC5	25	1024	1000

The table I is demonstrating the results of precision, recall and accuracy for different document sets selected to test multi label classification. Each documents consists of textual information about a medical case diagnosis. The formula used for accuracy is f = 2*(precision * recall) / (precision + recall), and precision is calculated based on standard deviation of highest and lowest value found from the results.

VI. CONCLUSION

The classical technique of redundancy can bring substantial benefits to SDNs in terms of flexibility to immediate recovery. The existing failure recovery solutions provide restoration of routes for the data plane, but proactive protection of switches against failures by installing backup switches can mitigate the problems associated with recovery solutions like flow rerouting, flood of control. The process flow defined in this research is found to be promising for expected results. The results are verified based on selected model. The result parameters decided are Migration time and Down Time. Where migration time is the time from start of the live migration process until the virtualization framework notifies that the source host can be deactivate and down time or blackout time is the phase during migration when there is a user perceptible service unavailability. The future work is to test the implementation algorithm with various load of planetlab and show the results of comparison with existing solutions.

VII. REFERENCES

[1] Green Distributed Cloud Services Provisioning in SDN-enabled Cloud Environment, Khaled Alhazmi∗†, Abdallah Moubayed∗, and Abdallah Shami - Elsevier-2017

[2] Live migration improvements by related dirty memory prediction in cloud computing Tin-Yu Wua, Nadra Guizanib,∗, Jhih-Siang Huangc

[3] A new technique for efficient live migration of multiple virtual machines– Elsevier-2015

[4] Minimum-energy bandwidth management for QoS live migration of virtual machines - Enzo Baccarelli∗ , Danilo Amendola, Nicola Cordeschi

[5] Virtualizing vehicular node resources : feasibility study of virtual machine migration–Elsevier-2016

[6] Fine-grained Nested Virtual Machine Performance Analysis Through First Level Hypervisor Tracing -Hani Nemati∗ , Suchakrapani Datt Sharma†, and Michel R. Dagenais.

[7] An Empirical Study of Different Cloud Migration Techniques – IEEE-2017

[8] Mariano Graziano, Andrea Lanzi, and Davide Balzarotti. “Hypervisor Memory Forensics”. In: Research in Attacks, Intrusions, and Defenses: 16th International Symposium, 2013, pp. 21–40.

[9] A. Montplaisir-Goncalves et al. “State History Tree: An Incremental Disk-Based Data Structure for Very Large Interval Dat”. In: Social Computing (SocialCom), 2013 International Conference on. 2013. A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, “kvm: the linux virtual machine monitor,” Proceedings of the Linux Symposium, no. 1, pp. 255–230, 2007.

[10] L. Cherkasova, D. Gupta, and A. Vahdat, "When virtual is harder than real: Resource allocation challenges in virtual machine based it environments," Technical Report HPL-2007-25, February 2007.

[11] G.-B. WANG, Z.-T. MA, and L. SUN, "On load balancing-oriented distributed virtual machine migration in cloud environment," Computer Applications and Software, vol. 30, no. 30, pp. 1–2, 2013.

[12] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience, vol. 41, no. 1, pp. 23 – 50, 2011.

[13] Y.-Y. LIU, Q.-Y. GAO, and Y. CHEN, "A method on load balancing of virtual machine resources in virtual computing environment," Computer Architecture, vol. 36, no. 16, 2010.

[14] J.-J. LIU, G.-L. CHEN, and C.-X. HU, "Virtual machine migration scheduling strategy based on load characteristic," Computer Engineering, vol. 37, no. 17, pp. 276–278, 2011.

[15] X. YANG, Z.-T. MA, and L. SUN, "Performance vector-based algorithm for virtual machine deployment in infrastructure clouds," Journal of Computer Applications, vol. 32, no. 11, p. 1, 2012. Oualha N, Leneutre J, Roudier Y. Verifying remote data integrity in peer-to-peer data storage: A comprehensive survey of protocols [J]. Peer-to-Peer Networking and Applications, 2012, 5(3):231-243.

[16] Y. PEI, J. WU, and X. WANG, "Virtual machine placement strategy based on particle swarm optimization algorithm," Computer Engineering, vol. 38, no. 16, p. 16, 2012.