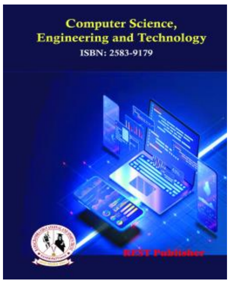
Computer Science, Engineering and Technology

Vol: 1(1), March 2023

REST Publisher; ISSN: 2583-9179 (Online)

Website: https://restpublisher.com/journals/cset/

DOI: https://doi.org/10.46632/cset/1/1/7



Live Data Replication to A Disaster Recovery Site Using XEN Hypervisor

***Sugin lal G, Blessy R**

Bharathiar University, Coimbatore, India.

*Corresponding Author Email: *g.suginlal@gmail.com*

Abstract: *Live Migration allows server administrator to transfer a Virtual Machine in running state or application between different physical machines without disconnect the client. In businesses using a virtualized infrastructure have multiple reasons to move the running virtual machines from one physical server to another server. The live migrations are used for balancing performance needs, routine maintenance, and work distribution. The good virtual infrastructure executes the move as quickly with minimal impact to end users. The ability to transfer the active Virtual Machines as quickly and as possible from one physical machine to another machine without any service disturbance is a key element of any virtualized environment. High availability of online transaction servers are very hard to handle and safeguard, for that reason a very secure environment to safe guard application in the live servers is required. Hence to provide a significant benefit for the admin's in the server maintenance, they are live virtual machine migration from physical host to the destination host with the synchronized Environment for maintain the destination server as a disaster recovery site.*

Index Terms: *Virtualization, Virtual Machine, Disaster Recovery, Live Migration, Hypervisor*

1. INTRODUCTION

Virtualization is a software technology that is quickly transforming the IT landscape and basically changing the way that people compute. At present powerful computers is designed for running a single operating system and application which leaves most machines not utilized. Virtualization allows running multiple virtual machines on single machine and giving the resources across multiple infrastructures. The different virtual machines can able to run different operating systems and multiple applications on the same physical computer. By virtualization can able to run simultaneously more than one operating system on the physical machine. The key idea behind this is to better utilize the available resources in an organization.

2. RELATED WORK

Live migration of virtual machines (VMs) is essential for data center management, but traditional pre-copy methods suffer from high overhead and performance degradation. This paper introduces MECOM, a novel memory-compression-based VM migration approach. By leveraging an adaptive zero-aware compression algorithm, MECOM achieves significant reductions in downtime, total migration time, and transferred data compared to Xen, while minimizing service disruption [1]. Live VM migration is crucial for efficient cloud data center management, but it poses significant performance challenges. This paper provides a comprehensive survey of live VM migration techniques, categorizing them based on duplication mechanisms and context awareness. We analyze performance metrics, identify critical data for transfer, and discuss security threats. By highlighting research gaps and challenges, this study offers valuable insights for cloud professionals and researchers to develop optimized live VM migration solutions [2]. Live VM migration is crucial for optimizing resource utilization and energy efficiency in cloud environments. This paper proposes an adaptive de-duplication mechanism to accelerate live migration by significantly reducing VM disk image size. Our approach combines fixed and variable-length block-level de-duplication with Rabin-Karp hashing and Akka stream processing for efficient data handling. Experimental results demonstrate substantial reductions in image storage space and migration time, leading to improved cloud resource management and cost savings [3]. Live VM migration offers significant advantages for cloud and data center environments. This paper focuses on the pre-copy approach, comparing the performance of

Xen and KVM hypervisors. By simulating VM migration using Cloud Report, we evaluate the impact of hypervisor choice on migration efficiency. Our findings indicate that KVM outperforms Xen in terms of overall performance [4]. In a Virtual Machine cluster, various multiple virtual machines share a Physical resource pool [5]. Due to dynamically varying workloads, various nodes are often underutilized, whereas others may become heavily loaded. Several virtual machine [VM] migration strategies are used to balance Virtual Machine loads among physical nodes. Based on periodically collected resource usage status, some virtual machines are migrated from overloaded machines to light-loaded ones. Live migration is flexible to implement on power budget, power saving and power reservation. By consolidating VMs on several light-loaded physical machines, some idle nodes can be powered off. In different application scenarios Virtual Machine migration is to be done fast and Virtual Machine service degradation is to be low during migration. Science pre-copy is a frequently used approach in the state of the art which is difficult to provide quick migration with low network, because of great amount of transferred data during migration that leads to performance degradation for services of virtual machine.

The Virtual machine live migration relocates the memory and virtual device of a Virtual Machine from one physical machine to several machines with no noticeable downtime of the Virtual Machine [6]. This offers interesting advantages form data center management, including load balancing, power efficiency and transparent infrastructure maintenance. Here the Virtual Machine [VM] migration in a cluster environment where a network accessible storage system such as Storage Area Network or Network-attached Storage is employed. The memory and CPU status should be transferred from the source to the destination. Live migration techniques in the state of the pre-copy approach which creates dirty pages since first it transfers all memory pages and then copies pages just modified during the last round iteratively. As applications like writable working set becomes small or if the maximum iterations is reached, virtual machine [VM] is suspended and only CPU state and dirty pages are transferred in the last round and sent out to the destination. [7] have conducted experiments to ensure the service available after the VM migration, The enough recourses and state of a running computer across a network, including its disks, memory, CPU registers, and I/O devices etc are quickly moved. The first advantage of live migration is the fact that it facilitates proactive maintenance. This is important when the migration is needed for the recovery of original host. Secondly, live migration can also be used in load balancing and here the works is distributed among two or more computers to optimize the use of available CPU resources. Live migration of virtual machines [VMs] allow transparently access the multi-core or many-core, that is the operator or administrator don’t want to concerned with what is occurring within the VM. Two kinds of methods are used in a cloud computing environment, where virtual machines are migrated [8]. One method is non-live migration, and the other way is live migration. In case of non-live migration process the virtual machine stops their processes in migrations. While in live migration process the virtual machine (VM) and its processes keep running during migration, although their performance may severely reduce. Virtualization is being used for a variety of purposes. The ability of virtual machines at system level is to decouple the operating system from the hardware has spurred their use in the area of server consolidation to improve system utilization [9]. This can help to reduce the physical machines and to reduce the associated operation costs, e.g., power, cooling, etc. The encapsulation of VMs can be leveraged to assist in system management and provide user customizable environments. Virtualization also provides interesting opportunities for fault tolerance, e.g., VM migration for proactive fault tolerance. These capabilities are being leveraged for research and development in a range of domains, to include High-Performance Computing (HPC).

Trusted computing technology into virtual machine instances, by means of partly porting the VTPM components in Xen into an administrative read-only domain, where they are protected from illegal operations in Domain 0, consequently enhancing the confidentiality of guest VMs [10]. This strengthens the integrity of virtual TPM itself and optimizes trusted computing base of Xen as well. Our main contributions lay on the construction of an administrative Domain and work of porting VTPM manager and TPM device driver (including TPM driver and TPM backend driver) into Domain A.

As the number of application running in a facilitating virtualized environment is increased, a virtual network circumstance becomes more complex, the security issues thereby have been a concern for academic fields and industrial [6]. However, the current solutions are mostly confined to several patchy works on system which still requires capable hacking skills for administrators and cannot make sure continuous protection for Virtual Machine that result in potential security risks. Here a framework VNSS that provides both guarantee of full lifecycle protection for Virtual Machine and distinct security level requirement. It has implemented a prototype system based on Xen hypervisor to evaluate the framework. This experiment results shows that this framework be able to provide continuous protection for virtual network environment. The Virtual Machine Monitor maintains a copy of the virtual machines memory management data structure [12]. In this data structure, the shadow page table lets the Virtual Machine Monitor precisely control which pages of machines memory are available to a virtual machine. The operating system running in a virtual machine creates a mapping in its page table, the Virtual

Machine Monitor [VMM] identifies the changes and establishes denoting in the related shadow page table entry that denotes to the actual page location in the hardware memory. While the virtual machine is executing the hardware uses shadow page table for memory translation and hence the Virtual Machine Monitor can always control what memory each virtual machine is using. Operating systems create frequent changes to the page tables keeping copies in software can leads to undesirable overhead. The shadow page tables managed by hardware for accelerating CPU virtualization in mainframe virtualization architectures would prove a successful direction. Several work leftovers in for Virtual Machine Monitors and guest operating systems are the future area of research.

The Microwiper is a practical approach to efficiently propagate memory in migrating virtual machines [13]. Two main strategies are used in Microwiper. First, the ordered propagation to transfer dirty pages in the order of their rewriting rates. Second, a transfer throttle is designed to avoid memory hot spot in data transferring. It calculates the available network bandwidth for sending pages to reduce traffic control. After the accumulated rewriting rate grows beyond the estimated bandwidth the next iteration is started frequently, to build a prototype Microwiper system by retrofitting the live migration in Xen hypervisor. Experimental results denotes that compared with pre-copy based migration in Xen, Microwiper can reduce dirtied pages and also shorten service downtime and total migration time. Specifically, Microwiper reduces downtime and transferred memory pages by 50%. Post-copy migration defers the memory transfer till the Virtual Machines CPU state has already been transferred to the destination and resumed there. [14]. Post-copy first sends all processor state to destination host, then starts the Virtual Machine at the target, and then actively pushes the Virtual Machines memory pages from source to target host. At the destination any memory pages that are faulted on by the Virtual Machine and not pushed are demand paged over the network from source host. The Post-copy thus ensures that each memory page is transferred once, hence avoiding the duplicate transmission overhead of pre-copy.

3. DESIGN

The Live Virtual Machine migration denotes relocating a running Virtual Machine from one physical machine to several machine. This process should be clear to the guest Operating System, remote clients and its resident applications of the Virtual Machine (VM). The services provided by the migrated Virtual Machine should not be migration alert. Previous memory to memory approaches meet this requirement in a LAN, and which cause a long duration migration downtime in WANs. The goal is to make the migration feasible in both LAN and WAN with reduced downtime. Furthermore, good tradeoff is done among the following performance metrics:
1) Migration Downtime: The time in which no other CPU cycle is committed to any of the Virtual Machine resident application not at the source or at the destination system. It also provides the time necessary to suspend the virtual Machine on the source, transfer the Virtual Machine and device state that is loaded and next activate the migrated virtual Machine on the remote machine.
2) Total Migration Time: The interval from the time Virtual Machine migration is started to the time the migrated Virtual Machine gets a consistent state i.e. the total time during which the state of two Virtual Machines is synchronized.
3) Total Data Transmitted: The total amount of data transferred while synchronizing the both Virtual Machines states.

Live Migration allows an administrator to transfer a running VM or application between different physical machines without disconnecting the client. At the present the uses of virtual machine migration have significant attention on data centers management. The capability of migrating live virtual machine among different physical hosts provides a significant advantage for multiple Virtual Machine based environments in many key scenarios:

1) Load balancing: Virtual Machines can be arranged across different physical machines in virtual cluster to reload on heavily congested hosts.

2) Online maintenance and fault tolerance: A physical machine may need servicing for future system fault. An administrator should migrate the running Virtual Machines to alternative machine freeing the original machine for recovery. Hence live Virtual Machine migration improves system availability and serviceability.

3) Power management: The load and throughput are not even but statistically normal at dissimilar periods. When a few application running on a distributed physical machines which works at light load can rearrange those applications with their basic Virtual Machines onto a single host and then the original hosts may be decommissioned when migration over. This method of management of power helps to reduce IT operation costs and benefit the natural environment.

Process of live migration:

1) Pre-Migration**:** The target host with needed resources is selected to receive the Virtual Machine migrated. The target host may also be preselected to speed any future migration.

2) Reservation: Following the initial choice, source host S sends a request of migrating a Virtual Machine to target host T. The Virtual Machine container of the source Virtual Machines size is reserved. If the resources are not satisfied, the Virtual Machine simply continues to run on host S with no distress.

3) Make and transfer checkpoint: Create a checkpoint on host S. Once the checkpointing is over, transfer the checkpoint file to host T. For the duration of the transfer log files will be generated.

4) Overlapping iterative log files transfer: To achieve fast convergent live migration the proposed method in the iterative process adopts fine-grained log files division and a replay time unit's feedback mechanism is used. The iterative process is conducted after transferring the log files which get generated in the path of checkpoint at the first round of transferring file which are copied from source host S to target host T, while virtual Machine on host S is constantly running and recording non deterministic events n log files according to time unit. When at the same time log files are replayed on host T after once it had recovered from the checkpoint and received these log files. When the received log files have generated the time units for replaying are counted.

5) Wait and Ack & Stop and Copy: As the time units the target host consume are reduced to a specified value host T inquires S whether the stop and copy stage can be executed. If the log files remaining size is smaller than another specified size the source Virtual Machine is suspended and the remaining log file is transferred to host T. After these log files are replayed, there is a reliable suspended copy of the Virtual Machine at both Source S and target+ T.

6) Commitment: After the iterative process comes to an end, host S indicates to target T that it has successfully synchronized their running states. Host S acknowledges the message for commitment of the migration transaction. Then the original Virtual Machine may be discarded.

7) Activation: The migrated Virtual Machine on host T is activated now. The new Virtual Machine advertises its IP addresses that is moved to the new machine that reattaches device drivers this guarantee the service for entire times running on source Virtual Machine until the migration commits with any risk of failure.

4. ARCHITECTURAL DESIGN

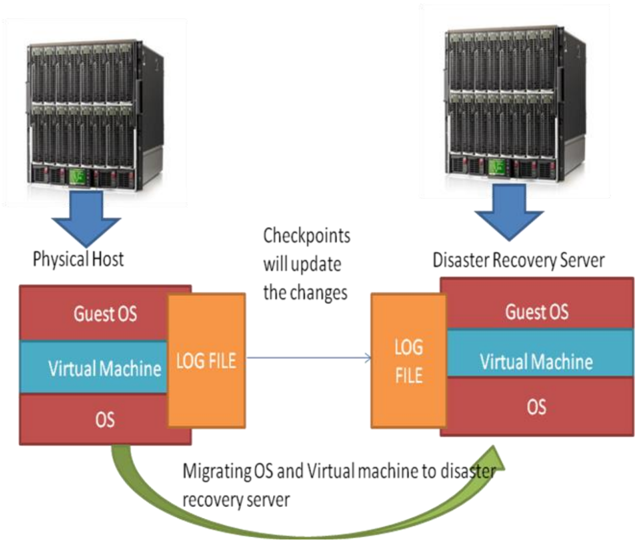


FIGURE 1. Architecture Diagram

Check pointing:

The check pointing size affects the fault tolerance for a complete VC. Additionally, in a VC, multiple Virtual Machines are distributed in different servers as a computing node. The failure at one Virtual Machine can be able to affect the states of other related Virtual Machines, and may sometimes cause them to also fail. Assume that there are two Virtual Machines (VMs), Virtual Machine A and Virtual Machine B, running in a VC. If Virtual Machine B sends some messages to Virtual Machine A and then fails. These messages may be may change the state of Virtual Machine A and correctly received by Virtual Machine A. When Virtual Machine B is rolled-back to its latest, check pointed state before the message is received by virtual machine B. The Virtual Machine A should be rolled back to a check-pointed state. i.e., all the Virtual Machines must be check-pointed at consistent states. Figure; represent the classical primary backup modal for a basic fault-tolerant protocol design for VC. All the Virtual Machines in the VC are represented by primary. State of every primary Virtual Machine is check-pointed at a globally consistent state, and all the checkpoints are saved on the backup, which can be a Virtual Machine or a physical machine. When anyone Virtual Machine fails that leads to VC failure and the backup Virtual Machine server will roll-back and take over each Virtual Machine to its previous check-pointed state, and resume the whole VC from a globally consistent checkpoint. Here there are two different types of downtimes: First type is VC downtime method, i.e. this is the time from at what time the failure is detected in the VC to at what time the VC resumes from last check-pointed state on the backup Virtual Machine server also starts to reply for client requests. Second is Virtual Machine downtime, which is the time when the Virtual Machine pauses to save the checkpoint to the Virtual Machine resumes the checkpoint. Saving a small checkpoint needs less time than saving a large checkpoint. Therefore, a lightweight checkpoint method reduces the Virtual Machine downtime.

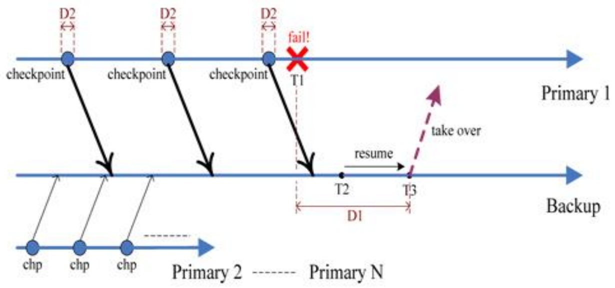


FIGURE 2. Backup modal and downtime problem

Disaster recovery site:

Disaster Recovery Site services provides a powerful alternative to the cost and complexity of dedicated hot-site, flexibly addressing the recovery time, standby facility and recovery point requirements of applications to keep running cost effectively during the site failure.

5. EXPERIMENTAL RESULTS

The experiment is performed on three machines of which two machine consist of same configuration, Intel i3 processor with 3 GB RAM, connected via a fast Ethernet. We have used Fedora 16 as the guest OS. The host kernel is the modified version of Xen. Storage is accessed via iSCSI protocol from a SAN network attached storage server. A VM is migrated five times from the source machine to the target machine in each experiment. The average of the five tests is taken as the results. The virtual machine is migrated within reduced downtime and the total data transferred is increased within the reduced amount of time.

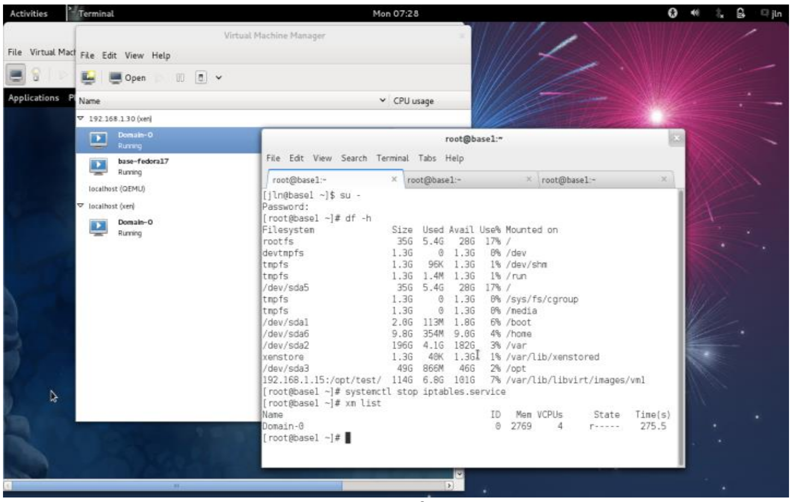


FIGURE 3. Virtual Machine before migration in Machine 1

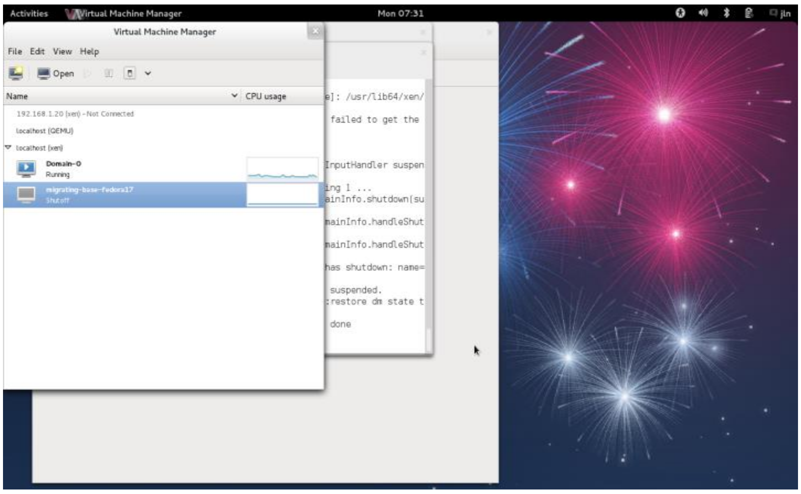


FIGURE 4. Migrating Virtual Machine to Machine 2

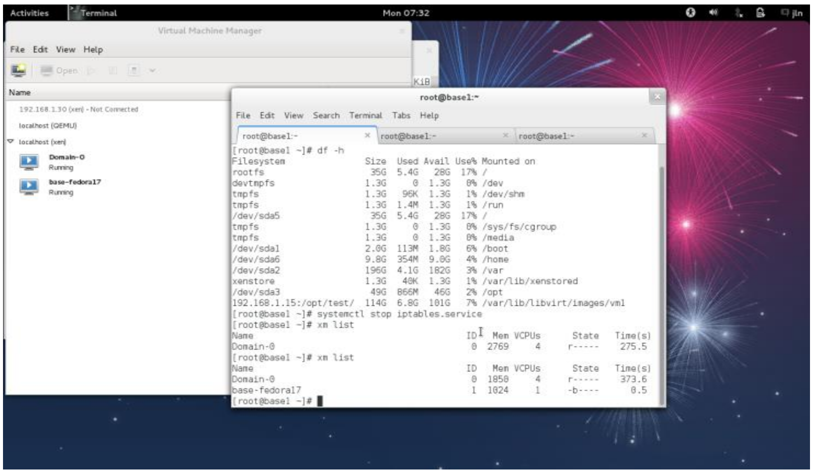


FIGURE 5. Migrated Virtual Machine in Machine 2

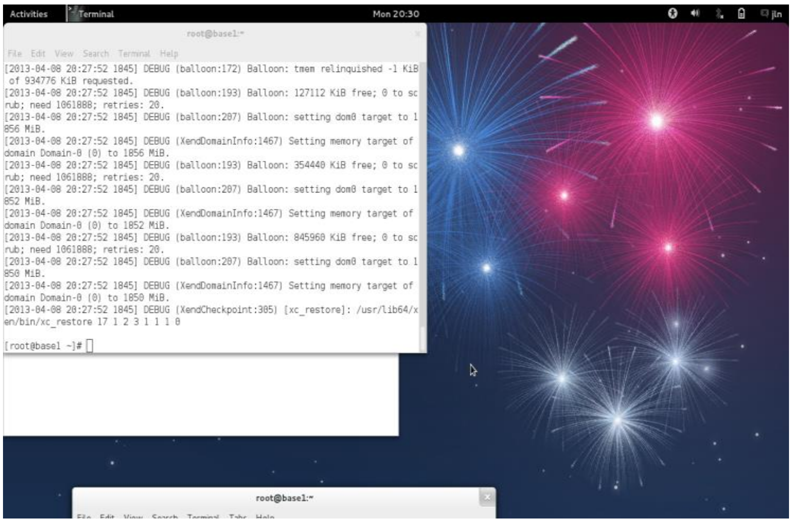


FIGURE 6. Recovering with Checkpoint

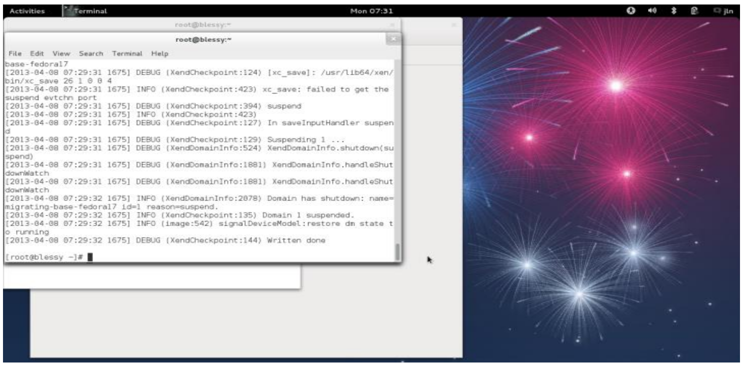


FIGURE 7. Reply with log

6. PERFORMNCE EVALUATION

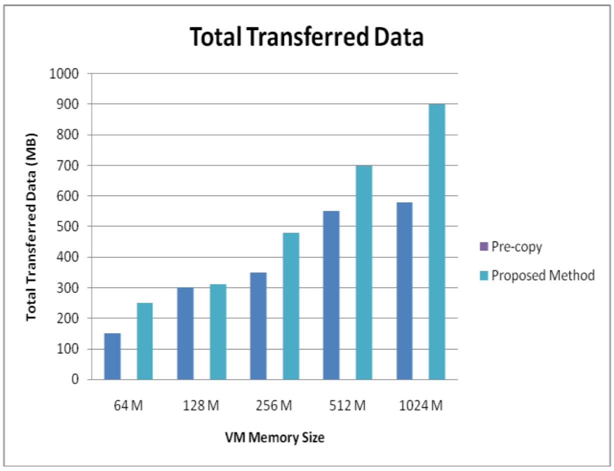


FIGURE 8. Total transferred data of Pre-copy and the proposed method

7. CONCLUSION

Thus the objective of this paper live Operating system migration using Xen virtual machine which enables quick movement of workloads within data centers and clusters. Hence the dynamic network bandwidth allows migration to proceed within a minimal impact on running services, reducing total downtime below the thresholds. Finally, to provide a significant benefit for the administrators in the server maintenance, they are live virtual machine migration from physical host to the destination host with the synchronized Environment for maintain the destination server as a disaster recovery site.

REFERENCE

[1]. Divya Kapil, Tabish Rao Live Virtual Machine migration: A Performance Test Xen and KVM. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-5, January 2020.

[2]. TYJ, N.M., G, V. Adaptive deduplication of virtual machine images using AKKA stream to accelerate live migration process in cloud environment. J Cloud Comp 8, 3 (2019). https://doi.org/10.1186/s13677-019-0125-z.

[3]. Choudhary, A., Govil, M., Singh, G. et al. A critical survey of live virtual machine migration techniques. J Cloud Comp 6, 23 (2017). https://doi.org/10.1186/s13677-017-0092-1.

[4]. Hai Jin, Li Deng, Song Wu, Xuanhua Shi, Hanhua Chen, Xiaodong Pan, MECOM: Live migration of virtual machines by adaptively compressing memory pages, Future Generation Computer Systems, Volume 38, 2014, Pages 23-35, ISSN 0167-739X, https://doi.org/10.1016/j.future.2013.09.031.

[5]. Hai Jin, Li Deng, , Xuanhua Shi, Song Wu, Xiaodong Pan "Live Virtual Machine Migration with Adaptive Memory Compression" IEEE International conference on Cluster Computing and Workshops, 2009.

[6]. Gao Xiaopeng, Wang Sumei,Chen Xianqin " VNSS: A Network Security Sandbox For Virtual Computing Environment" International conference on networking and security, 2010.

[7]. Tsuyoshi Ota, Shusuke Okamoto "Using I/O schedulers to reduce I/O load in virtualization environments" Published by workshops of "International Conference on Advanced Information Networking and Applications", 2011.

[8]. Thomas Naughton, Geoffroy Vall´ee, Ferrol Aderholt "Loadable Hypervisor Modules" Proceedings of 43rd International Conference on System Sciences, Hawaii, 2010.

[9]. Nalini Vasudevan, Prasanna Venkatesh "Design and Implementation of a Process Migration System for the Linux Environment"

[10]. Xuxian Jiang, Zhi Wang, "HyperSafe: A Lightweight Approach to Provide Lifetime Hypervisor Control Flow Integrity" Published in IEEE Symposium On Security And Privacy, 2010.

[11]. Heeseung Jo, Hwanju Kim, Joonwon Lee XHive "Efficient Cooperative Caching for Virtual Machines", IEEE Transactions on Computers, Vol. 60, no. 1, January, 2011.

[12]. Umesh Deshpande, Michael R. Hines, Kartik Gopalan "Post-Copy Live Migration of Virtual Machines", Published by ACM SIGOPS Operating Systems, Vol. 43, 2009.

[13]. Jian Chen, Yuyang Du Hongliang Yu, Guangyu Shi, Weimin Zheng "Microwiper: Efficient Memory Propagation in Live Migration of Virtual Machines" 39th International Conference on Parallel Processing, 2010.

[14]. Mendel Rosenblum, Tal Garfinkel "When Virtual Is Harder Than Real;Security Challenges In Virtual Machine Based Computing Environments" Stanford University Department of Computer Science, Published by the Association for Computing Machinery, 2005.

[15]. Ben Pfaff Jim Chow, Sapuntzakis, Constantine P, Monica S Lam, Ramesh Chandra "Optimizing Live Migration Of Virtual Machines", Computer Science Department Stanford University, 2000.

[16]. Sayantan Chakravorty, Terry Jones, Andrew Tauferner Celso L. Mendes Todd Inglett Laxmikant V. Kal´e Jos´e Moreira " HPCColony: Services and Interfaces for Very Large Systems", Published by Association for Computing Machinery, 2006.

[17]. Y. Luo, B. Zhang Y. Sun, X. Wang, Z. Wang, H. Chen, "Live and Incremental whole System Migration of Virtual Machines using block bitmap," Published by IEEE International Conference on Cluster Computing, PP. 99–106, 2008.

[18]. K. Fraser , C. Clark, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, A. Warfield, "Live migration of virtual machines" Proceedings of second Symposium on Networked Systems Design And Implementation, PP. 273–286, 2005.

[19]. W. Huang, Q. Gao, J. Liu, D. K. Panda, "High Performance Virtual Machine Migration with RDMA over Modern Interconnects" IEEE International Conf. Cluster Computing, September, 2007.

[20]. R. Nathuji, K. Schwan, "Virtual Power: Coordinated Power Management in Virtualized Enterprise Systems", Proceedings of ACM Symp. Operating Systems Principles, October 2007.

[21]. K. K. Ramakrishnan, P. Shenoy, J. van der Merwe, "Live Data Center Migration across WANs: A Robust Cooperative Context Aware Approach", ACM Workshop on Internet Network Management, August, 2007.

[22]. M. Shaw, A. Whitaker, R. S. Cox, S. D. Gribble, "Constructing Services with Interposable Virtual Hardware", Proceedings of First Symp. Networked Systems Design, March, 2004.