ISSN 2348 – 8034
Impact Factor- 3.155

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

VM MIGRATION: A SURVEY

Usha Keshavamurthy*1 and Dr. H S Guruprasad2
*1PG Scholar, Deptartment of CSE, BMSCE
2Professor and Head, Deptartment of CSE, BMSCE

ABSTRACT

Virtual Machine (VM) refers to an operating system or application environment running on a software that imitates dedicated hardware without change in end-user experience. VM migration refers to the movement of VMs from one physical machine to the other while trying to minimize its impact on the performance of VM and its applications. VM migration has several benefits like server consolidation, load balancing and so on. In this survey paper, we look at VM migration benefits, techniques, performance metrics and also, recent research conducted around VM migration techniques and optimizations to improve the process to suit specific workloads.
An abstract of no more than 200 words (10pt Times New Roman, Justified).

Keywords- *Virtual Machine, VM Migration.*

I. INTRODUCTION

A Virtual Machine (VM) is an instance of OS along with certain applications, running in an isolated partition within a physical machine. A physical machine can support several such VMs on it with the help of a specialized software that forms the virtualization layer. The specialized software is commonly known as hypervisor or virtual machine monitor (VMM). This concept of virtualization is key to cloud computing for distribution of resources among millions of users and for optimal resource utilization. Physical machines that support multiple VMs are usually referred to as servers in the domain of cloud computing. Migration of VMs among servers becomes often necessary to adapt to the changing environment like, a server running out of storage and/or compute capacity, need to balance load among a set of servers to achieve application performance SLA and so on. In the following sections, we will see the various benefits, techniques and research relating to VM migration.

Benefits of VM migration

- ✓ Load Balancing - Here, the goal is to balance the workload among a set of servers in order to increase application performance.
- ✓ Server Consolidation - In server consolidation, VM migration is done so as to accommodate the running VMs in minimal number of servers. This is done in order to maximize resource utilization and for power management.
- ✓ Ease of maintenance - VM migration eases maintenance tasks thereby helping to reduce the downtime owing to server maintenance. VM migration can also be used for VMM rejuvenation to increase availability and avoid / postpone future failure due to aging software. When VMM needs to be rejuvenated VMs are migrated to another host and moved back after rejuvenation.[31]
- ✓ Fault Tolerance - Here the VMs are migrated owing to server failure to increase VM availability.

VM Migration Techniques and Metrics

There are several approaches that have been followed for VM migration such as:

- **Stop-and-copy**: Also known as cold migration, in this technique, VMs running on a server are halted, moved (copied) to destination server, and then restarted on the destination server. This was the simple traditional method followed but can cause longer downtime.
- **Post-copy**: In this technique, there is a small stop-and-copy phase that transfers essential kernel data and the VM is started on the destination server. The remaining pages (data) of VM are transferred on demand. Though this results in smaller downtimes, total migration time would be increased.
- **Pre-copy (live migration)**: comprises of two phases - warm-up and stop-and-copy phase. In warm-up phase, the memory pages of VM are copied from source to destination server without halting the VM. Any pages that are modified during this phase, would have to be copied again. Once the no. of pages to be copied reaches below specified threshold, stop-and-copy phase is carried out i.e, VM is halted, remaining pages copied, after which VM is restarted on the destination server. This reduces both downtime and total migration time substantially.

- **Modified versions of pre-copy**: Several modifications have been proposed to use memory compression before copy, and including preprocessing phase using working set prediction algorithm.

Peter Svard et.al [32] provide comparison of VM migration techniques and present principles and performance characteristics of algorithms for live migration.

The key performance metrics used to evaluate VM migration are downtime, total migration time, preparation time and no. of pages transferred during VM migration.

II. LITERATURE SURVEY

In this section, we will briefly look into various research work carried out on both VM migration approaches as well as on how to optimize the migration.

Sapuntzakis et al [1] suggest various optimizations on original stop-and-copy to form post-copy technique, such as using copy-on-write disks to capture only the updates, ballooning zeros unused memory, using demand paging and hashing techniques to reduce data sent over the network. These optimizations target to reduce the start-up time of VM. Clark et al [2] introduce the concept of writable working set and present the design and implementation of 6-stage VM migration using pre-copy technique. The stages include pre-migration (Stage 0), reservation on destination server (Stage 1), Iterative pre-copy (Stage 2) where dirty pages are copied iteratively in successive rounds, stop and copy (Stage 3) where remaining data is transferred, commitment (Stage 4) where VM state on source is released followed by activation (Stage 5) where the VM is restarted on destination server. Luo [3] provides a three phase algorithm for live migration of VMs running state including their local disk storage. Block bitmap is used to track writes to local disk storage and to direct the synchronization. Experimental results show that the algorithm performs well even with write intensive workloads and the overhead due to block bitmap is very low. Jin et al [4] present the design and implementation of migration approach that uses memory compression and is zero aware to provide fast and stable migration. Results show that the proposed approach reduces 27% of downtime, 32% of total migration time and 68.8% of data transferred. Zaw et al [5] present a framework that uses preprocessing phase in traditional pre-copy approach. A working set prediction algorithm is proposed for preprocessing that uses LRU cache with splay tree algorithm. Results show that the proposed framework can reduce around 23% of total data transferred and reduce around 11% of total migration time. Aldhalaan et al [6] derive a model for live VM migration to compute the downtime and network utilization due to VM migration under both uniform and non-uniform dirtying rates. The paper also presents a non-linear model for finding an optimal threshold value, which represents fraction of dirty pages at which simultaneous page copy from running VM is halted and stop-and-copy step of live VM migration takes place. Ferreto et al [7] propose ways to control VM migration which prioritizes VMs with steady capacity. Experimental results show that for server consolidation, it is better to avoid migrating VMs with steady capacity with minimal penalty in no. of physical servers- in comparison to eager-migration-based solutions to minimize the no. of unnecessary migrations. Koto et al [8] propose that pruning memory of VM before memory transfer reduces the amount of data transferred, thereby reducing total migration time and downtime and performance degradation due to migration noise. During pruning, memory pages that need not be transferred are marked as soft pages and are not considered during VM migration, such as file cache and free pages. Results show that migration with the proposed prototype has migration time upto 68% shorter and reduces network traffic upto 83% when compared to Xen based live migration. Hu et al [9] present an optimization to live migration in order to reduce migration overhead. The paper proposes a fast convergent live migration approach that synchronizes VM state between source and destination servers by log tracing and replaying instead of dirty memory pages transferring. Experimental results show that proposed method can significantly reduce migration overhead compared to traditional pre-copy method in a fast network bandwidth. Rai et al [10] observe that there is mismatch between source memory and destination memory constructed with replay. The paper presents a suite of semantic techniques called MiG to show that compression can be used for memory transfer without relying on replay for desktop VM migration. Evaluation results show that using MiG, VM state can be compressed effectively leading to a reduction of about 50-60% bytes to be transferred on average. Bila et al [11] talk about migration of idle desktop VMs. Since migration of idle desktop VMs could have large migration latencies and consume a lot of network resources due to bulk data transfers, a prototype called Jettison is proposed, that is based on partial VM migration. Partial VM migration of idle desktops involves migration of only the working set of idle VM resulting in savings in the range of 85-104% in terms of energy using 10% lesser network resources and 2-3 times reduction in migration latency. Jain et al [12] state that there are two general problems w.r.t VM migration policy optimization- one being migration over bandwidth-oversubscribed network and other being the challenge to meet load constraints at the destination servers. This paper presents approximation algorithms to handle these challenges simultaneously with the objective to reduce the no. of hot servers. Deshpande et al [13] introduce a new performance metric for VM migration called eviction time that represents how quickly the VM can be taken off the source server. This is equal to total migration time for post-copy and pre-copy techniques. The paper presents a new approach termed as "Scatter-Gather", where the source can offload VM data into various intermediaries which can then be collected by destination server based on network bandwidth availability and memory sufficiency. This approach helps decouple source and destination server and the process to be carried out independently of source server. Jo et al [14] present a technique to reduce downtime and total migration time, which is based on the observation that a significant part of physical memory is used to cache data from secondary storage. VM's I/O operations are tracked on NAS device and during iterative pre-copy phase, only the memory to disk mappings is sent to destination server which then fetches the data directly from the NAS device. Experimental results show a reduction of over 30% of total transfer time on average for various benchmarks.

There are two types of optimization that can be achieved w.r.t VM migration - one is host based that includes memory compression, when to stop pre-copy phase and so on, and the other is network based optimization like selecting best path for data copy or optimize network update process to reduce downtime. Salimi et al [15] observe that performance is not isolated for a VM running on physical machine, due to, workload interference and configuration of set of VMs running on that physical machine. In this paper, a batch scheduler is proposed as a host based optimization that would schedule based on minimal interference and a particular VM's impact on proliferation of interference. Pausing of VMs shall not effect as workload considered is of batch type comprising scientific application workload. Liu et al [16] propose Software defined network (SDN) based live migration as a network based optimization. They suggest that it is possible to control traffic forwarding and network update by running programs on the controller, that brings the opportunity to implement network based optimization. Cello et al [17] propose that to reduce congestion events raising due to migration in distributed datacenter, traffic aware clustering needs to be done to identify tight clusters of VMs to be migrated. Wen et al [18] present a solution called VirtualKnotter, to reduce congestion with controllable VM migration traffic and low migration time. VirtualKnotter specifies a VM placement algorithm and VM migration scheduling algorithm to achieve its goals. Experimental results show that proposed solution achieves link utilization close to that of baseline algorithm, while decreasing link congestion time by 53% for production datacenter traffic.

There are several optimizations proposed, based on specific goal, workload or network used. Hwang et al [19] present a heuristic solution to VM consolidation by formulating the problem as multi-capacity stochastic bin packing problem. The resource demands are considered as random variables with known mean and standard deviation values, and assuming co-relation among these variables. Simulation results show that the proposed method produces high quality solutions. Chen et al [20] propose an alternative approach for VM migration to achieve server consolidation by means of effective VM sizing. Here, VM's dynamic load is associated with fixed demand. The resource demand is predicted using statistical multiplexing principles. It has been shown through a real data center load trace that using effective VM sizing, the solution achieves 10-23% more energy savings than traditional approaches. Corradi et al [21] propose a cloud management platform for optimizing VM consolidation and migration along three key dimensions, viz., power, host resources like CPU and memory, and networking. Experimental results show that interferences between co-located VMs have to be carefully considered prior to making VM migrations so that application SLA's are not affected. Ibrahim et al [22] show that pre-copy method may provide sub-optimal performance for memory intensive application workload. The paper also suggests an algorithm for minimizing downtime that controls migration based on rate of change of application memory. Jin et al [23] propose that, in case of memory/write intensive workload or when pre-copy speed is slow, VCPU working frequency can be reduced so that remaining dirty memory to be copied falls to desired small amount. Experimental results show that the optimized scheme can reduce upto 88% of application downtime with minimal overhead. Chanchio et al [24] present the design and implementation of novel Time-bound, thread-based Live Migration (TLM) mechanism for migration of VMs running memory intensive workloads. In this mechanism, additional threads are added to pre-copy algorithm to migrate the VMs within a bounded time period. The mechanism proposes CPU over-committing technique in order to minimize downtime and avoid performance impacts on other virtual machines during migration. Nicolae et al [25] propose a memory-migration independent approach that handles storage transfer for migration of VMs that run large scale, data intensive workloads. The approach uses hybrid active push/ prioritized prefetch strategy that is resilient to rapid disk state changes and is minimally intrusive to support portability across wide range of hypervisors. Experimental results show that proposed approach exhibit improvements of upto 10x faster migration time, 10x lesser bandwidth consumption and 8x lesser performance degradation over state-of-art approach.

Live migration of VMs require underlying infrastructure to have high performance, scalable virtualized I/O architecture. This is a challenge with high speed networks such as InfiniBand and RDMA enhanced Ethernet. Guay et al [26] propose that Single root I/O virtualization (SR-IOV) specification can address the challenges of performance and scalability. This paper discusses one of the first prototype implementations of live migration over SR-IOV enabled InfiniBand devices.

There are two other important aspects to be considered during VM migration- energy efficiency and security. Strunk et al [27] look at power consumption in relation to duration of VM migration. Experimental results show that live migration incurs energy overhead and the overhead is proportional to the size of VM and the available network bandwidth. Gaikwad et al [28] perform a survey on power efficiency of live migration for cloud data centers. They conclude that turning off idle machines, the power consumption by live migration can be reduced. To achieve this, they propose dynamic threshold based approach for host CPU utilization. Anala et al [29] discuss the attack model on virtualization system. The paper also presents the design and implementation of a security framework for secure live migration that addresses role based access, network intrusion, firewall protection and encryption policies. Aiash et al [30] use X.805 security standard to evaluate and analyze the threats and their sources. The paper also suggest ways to tackle these threats and provides a comparison of various approaches to secure live migration.

III. CONCLUSION

Live VM migration is key to cloud computing domain. In this paper, we present various popular techniques that have been adopted for VM migration along with recent research work that propose optimizations to the traditional approaches on multiple facets. We also look at energy efficiency and security aspects of VM migration.

IV. ACKNOWLEDGEMENT

The authors would like to acknowledge and thank Technical Education Quality Improvement Program [TEQIP] Phase 2, BMS College of Engineering and SPFU [State Project Facilitation Unit], Karnataka for supporting the research work.

REFERENCES

1. *C. P. Sapuntzakis, R. Chandra, B. Pfaff, J. Chow, M. S. Lam, M. Rosenblum, "Optimizing the migration of virtual computers", 5th Symposium on Operating Systems Design and Implementation (OSDI-02), Volume 36, Issue SI, pp 377-390, December 2002, DOI: 10.1145/844128.844163.*
2. *Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, Andrew Warfield, "Live migration of virtual machines", NSDI'05: Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation, Volume 2, pp 273-286, 2005.*
3. *Yingwei Luo, "A Three-phase Algorithm for Whole-System Live Migration of Virtual Machines", ATIP's 3rd workshop on High performance computing in China, pp 172-173, Nov 2007, ISBN: 978-1-59593-903-6, DOI: 10.1145/1375783.1375817.*
4. *Hai Jin, Li Deng, Song Wu, Xuanhua Shi, Xiaodong Pan, "Live Virtual Machine Migration with Adaptive Memory Compression", IEEE International Conference on Cluster Computing and Workshops, pp 1-10, Sep 2009, DOI: 10.1109/CLUSTR.2009.5289170*
5. *Ei Phyu Zaw, Ni Lar Thein, "Improved Live VM Migration using LRU and Splay Tree Algorithm", International Journal of Computer Science and Telecommunications, Volume 3, Issue 3, March 2012*
6. *Arwa Aldhalaan, Daniel A. Menasce, "Analytic Performance Modeling and Optimization of Live VM Migration", Proceedings EPEW, Volume 8168, pp 28-42, Oct 2013, DOI: 10.1007/978-3-642-40725-3-4*
7. *Tiago C. Ferreto, Marco A.S. Netto, Rodrigo N. Calheiros, Cesar A.F. De Rose, "Server consolidation with migration control for virtualized data centers", Future Generation Computer Systems, Volume 27, Issue 8, pp 1027-1034, Oct 2011, DOI: 10.1016/j.future.2011.04.016*
8. *Akane Koto, Hiroshi Yamada, Kei Ohmura, Kenji Kono, "Towards Unobtrusive VM Live Migration for Cloud Computing Platforms", APSYS '12 Proceedings of the Asia-Pacific Workshop on Systems, Article no. 7, July 2012, ISBN: 978-1-4503-1669-9, DOI: 10.1145/2349896.2349903*

9. *Liang Hu, Jia Zhao, Gaochao Xu, Decheng Chang, Jianfeng Chu, "A Fast Convergent Live Migration of Virtual Machine", Journal of Information & Computational Science, Volume 9, Issue 15, 2012, pp 4405-4412, Nov 2012*
10. *Anshul Rai, Ramachandran Ramjee, Ashok Anand, Venkata N Padmanabhan, George Varghese, "MiG: Efficient Migration of Desktop VMs using Semantic Compression", Proceedings of 2013 USENIX conference on Annual Technical Conference, June 2013*
11. *Nilton Bila, Eyal de Lara, Kaustubh Joshi, H. Andres Lagar-Cavilla, Matti Hiltunen, Mahadev Satyanarayanan, Jettison, "Efficient Idle Desktop Consolidation with Partial VM Migration", Proceedings of the 7th ACM European conference on Computer Systems, pp 211-224, April 2012, ISBN: 978-1-4503-1223-3, DOI: 10.1145/2168836.2168858*
12. *Navendu Jain, Ishai Menache, Joseph (Seffi) Naor, F. Bruce Shepherd, "Topology-Aware VM Migration in Bandwidth Oversubscribed Datacenter Networks", 39th International Colloquium conference on Automata, Languages, and Programming, pp 586-597, July 2012, DOI: 10.1007/978-3-642-31585-5_52*
13. *Deshpande U, Yang You, Chan D, Bila N, Gopalan K, "Fast Server Deprovisioning through Scatter-Gather Live Migration of Virtual Machines", IEEE 7th International Conference on Cloud Computing, pp 376-383, July 2014, DOI: 10.1109/CLOUD.2014.58*
14. *Changyeon Jo, Erik Gustafsson, Jeongseok Son, Bernhard Egger, "Efficient Live Migration of Virtual Machines Using Shared Storage", ACM SIGPLAN Notices - VEE '13, Volume 48, Issue 7, pp 41-50, July 2013, DOI:10.1145/2517326.2451524*
15. *Hadi Salimi, Mohsen Sharifi, Batch scheduling of consolidated virtual machines based on their workload interference model, Future Generation Computer Systems, Volume 29, Issue 8, Pages 2057–2066, Oct 2013, doi:10.1016/j.future.2013.02.005*
16. *Jiaqiang Liu, Yong Li, Depeng Jin, "SDN-based Live VM Migration Across Datacenters", Proceedings of SIGCOMM'14, Volume 44, Issue 4, pp 583-584, Oct 2014, DOI: 10.1145/2740070.2631431*
17. *Marco Cello, Kang Xi, Jonathan H. Chao, Mario Marchese, "Traffic-Aware Clustering and VM Migration in Distributed Data Center", Proceedings of DCC'14, pp 41-42, Aug 2014, DOI: 10.1145/2627566.2627582*
18. *Xitao Wen, Kai Chen, Yan Chen, Yongqiang Liu, Yong Xia, Chengchen Hu, "VirtualKnotter: Online Virtual Machine Shuffling for Congestion Resolving in Virtualized Datacenter", IEEE 32nd International Conference, pp12-21, June 2012, DOI: 10.1109/ICDCS.2012.25*
19. *Inkwon Hwang, Massoud Pedram, "Hierarchical Virtual Machine Consolidation in a Cloud Computing System", IEEE Sixth International Conference on Cloud Computing, pp 196-203, July 2013, DOI: 10.1109/CLOUD.2013.79*
20. *Ming Chen, Hui Zhangy, Ya-Yunn Suz, Xiaorui Wang, Guofei Jiangy, Kenji Yoshihira, "Effective VM Sizing in Virtualized Data Centers", IFIP/IEEE International Symposium, pp 594-601, May 2011, DOI: 10.1109/INM.2011.5990564*
21. *Antonio Corradi, Mario Fanelli, Luca Foschini, "VM consolidation: A real case based on OpenStack Cloud", Future Generation Computer Systems, Volume 32, March 2014, pp 118-127, June 2012, DOI: 10.1016/j.future.2012.05.012*
22. *Ibrahim K Z, Hofmeyr S, Iancu C, Roman E, "Optimized pre-copy live migration for memory intensive applications", International Conference on High Performance Computing, Networking, Storage and Analysis, pp 1-11, Nov 2011, ISBN: 978-1-4503-0771-0*
23. *Hai Jin, WeiGao, Song Wu, Xuanhua Shi, Xiaoxin Wu, Fan Zhou, "Optimizing the live migration of virtual machine by CPU scheduling", Journal of Network and Computer Applications, Volume 34, Issue 4, pp 1088-1096, July 2011, DOI:10.1016/j.jnca.2010.06.013*
24. *Chanchio K, Thaenkaew P, "Time-Bound, Thread-Based Live Migration of Virtual Machines", 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp 364-373, May 2014, DOI: 10.1109/CCGrid.2014.107*
25. *Bogdan Nicolae, Franck Cappello, "A Hybrid Local Storage Transfer Scheme for Live Migration of I/O Intensive Workloads", pp 85-96, June 2012, ISBN: 978-1-4503-0805-2, DOI: 10.1145/2287076.2287088*
26. *Wei Lin Guaya, Sven-Arne Reinemo, Bjørn Dag Johnsen, Chien-Hua Yen, Tor Skeie, Olav Lysne, Ola Torudbakken, "Early experiences with live migration of SR-IOV enabled InfiniBand", Journal of Parallel and Distributed Computing, Volume 78, pp 39-52, Jan 2015, DOI: 10.1016/j.jpdc.2015.01.004*
27. *Strunk A, Dargie W, "Does Live Migration of Virtual Machines Cost Energy?", 2013 IEEE 27th International Conference on Advanced Information Networking and Applications, pp 514-521, March 2013, DOI: 10.1109/AINA.2013.137*

28. Yogesh Eknathrao Gaikwad, Seema Singh Solanki, "A Survey: Power efficiency Live Migration for Data Centers in Cloud", International Journal of Computer Science and Information Technologies, Vol. 5 (6), June 2014, ISSN: 0975-9646

29. Anala M R, Shetty J, Shobha G, "A framework for secure live migration of virtual machines", Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference, pp 243-248, Aug 2013, DOI: 10.1109/ICACCI.2013.6637178

30. Aiash M, Mapp G, Gemikonakli O, "Secure Live Virtual Machines Migration: Issues and Solutions", Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference, pp 160-165, May 2014, DOI: 10.1109/WAINA.2014.35

31. Fumio Machida, Dong Seong Kim, Kishor S. Trivedi, "Modeling and analysis of software rejuvenation in a server virtualized system with live VM migration", Performance Evaluation, Volume 70, Issue 3, March 2013, pp 212-230, Sep 2012, DOI: 10.1016/j.peva.2012.09.003

32. Petter Svard, Benoit Hudzia, Steve Walsh, Johan Tordsson, Erik Elmroth, "Principles and Performance Characteristics of Algorithms for Live VM Migration", ACM SIGOPS Operating Systems Review, Volume 49, Issue 1, pp 142-155, Jan 2015, DOI: 10.1145/2723872.2723894