Exploiting Redundancy Among VMs to Reduce Network Load and Migration Time in Cloud Environments

Mohammad Khan[1], Ehtesham Zahoor[2]
FAST
Pakistan
mkafridi_19@yahoo.com
ehtesham.zahoor@nu.edu.pk

ABSTRACT: *Continuous computational evolution has led to cloud computing. Technologies like virtualization and distributed computing have allowed virtual machines to migrate from one physical machine to another without compromising the operations being performed. Storage Area Networks have helped reduce migration time and network load by removing storage as a factor affecting the migration process. At the same time increase in hardware size and complexity of operating systems have increased data in the memory that needs to be transferred during migration. Techniques like Pre-copy and Post-copy have been developed for live VM migration but increase migration times could degrade system performance and increase network load. Bulk of the research for virtual machine migration focuses on second phase, i.e., the reduction of data transfer when dirty pages get resent. Very little importance has been given to reduction of data during first phase when complete image of the memory and CPU states get transferred. This is where bulk of data gets transferred but compressing this data becomes infeasible due to the unavailability of resources. To solve this problem we have proposed a technique that reduces data without requiring additional resources.*

Keywords: Virtual machine, Migration, Data reduction.

Received: 11 May 2016, Revised 13 June 2016, Accepted 20 June 2016

1. Introduction

Virtualization has become an integral part of a Cloud environment due to its ability to remove hardware and software interdependencies. Virtual Machine (VM) is hardware independent through the presence of an abstraction layer provided by VM monitor between the hardware and VMs [2]. This ability of virtualization allows multiple VMs to be hosted on a single Physical Machine (PM) or multiple PMs in a single VM. Virtualization has enabled VMs to migrate from one PM to another PM over geographical boundaries. A key advancement in virtualization technology is the migration of VMs while they are still executing, and this type of migration is known as live migration [1]. Live migration plays a key role in enabling fault tolerance, load balancing and power optimization.

Live migration was developed for the migration of a VM while it was still running transparently from the user. But there exist some conditions that might cause downtime perceivable by the user, like writable working set becomes larger than transfer rate and the process fails to converge. On the other hand migration may cause brownout, a condition in which VM continues to operate but there is degradation in performance. Any of these issues may cause Service Level Agreement (SLA) violations resulting in the loss of business for both cloud service provider and the cloud users, results in legal issues.

Pre-copy [4, 5] and Post-copy [6] are some of the earliest and basic techniques developed for the live migration of VM both in Local Area Network (LAN) and Wide Area Network (WAN) environments. A Storage Area Network (SAN) is a centralized location with a complete network of storage devices working together to provide storage to all the PM within the datacenter. When comparing VM migration over LAN requires little effort due to no storage transfer or involvement of IP address management, while VM migration over WAN considerably prolongs the duration of VM migration due to storage migration, network congestion, limited available bandwidth, IP management, and the error-prone nature of WAN links [1].

The Pre-copy live migration of a VM over LAN, MAN or WAN can be divided into four main steps:

1. Transferring complete memory, CPU and device state.
2. Recursive transfer of pages that got dirty during transfer.
3. Stop the source VM and copy remaining pages.
4. Start VM on destination PM.

A number of techniques have been developed to reduce the data transfer during second step of live VM migration. Most prominent of these techniques is delta compression with run length encoding, where it can significantly reduce data during migration. While for the data transferred during the first step very research has been done with requirement of such high resource demand which in most cases renders the technique useless. To solve this issue we have focused our research on the first phase of live VM migration. During our research we have proposed ESCDR (Exploit Static Content for Data Reduction) an approach that describes how and which PMs should be selected as candidates for the migration process so that relevant material is already present on the destination PM resulting in reduced migration data.

The rest of the paper is organized as follows: Section 2 presents a summary of work related to VM migration. Section 3 presents the approach to reduce network traffic generated during VM migration. In section 4 an analysis of the proposed approach has been presented. Finally a conclusion is provided in Section 5 which states the issues currently faced by VM migration and the future direction of this research work.

2. Related Work

A lot of research has already been conducted on VM migration in a cloud environment. Let us start with Pre-copy live migration proposed by Nelson et al [4] and Clark et al [5], in which the virtual machine is sent once after which pages that get dirty during the data transfer are sent recursively until certain terminating conditions are met. The recursive resending increases the data sent over the network and resource utilization. Post-copy is presented by Hines et al [6], a technique in which machine state is sent to destination machine and the destination starts execution, while the required pages get fetched over the network. Then to improve performance of dynamic ballooning Hines et al [7] introduced the concept of multiple-pivot points so that multiple threads could be executed concurrently. The application of Post-copy may introduces brownout with consequences similar to downtime resulting in no advantage over Offline or Pre-copy VM migration.

Delta compression has been the focus of most of the researchers trying to reduce migration traffic. Bradford et al [8] used delta compression on pages that were resent over WAN to reduce storage and memory contents. Hacking et al [16] used delta compression after which run length encoding was applied on dirty pages that were resent. Zhang et al [9] named the technique MDD (Migration with Data Deduplication) which used delta compression after which run length encoding was applied on dirty pages that were resent. Svärd et al [3] have presented an evaluation of delta compression techniques. The evaluation showed a significant improvement in VM migration for large VMs as well as transparent migration on a VM with live streaming.

Huang et al [17] and Isci et al [18] proposed another technique to improve the latency of packets sent over the network called Remote Direct Memory Access, which was utilized to reduce the latency caused by TCP protocol. A major flaw of these

techniques is high level of complexity in the implementation and bypassing the security parameters. Travostino et al [22] proposed peer-to-peer network for the migration of virtual machines over wide area networks. Peer-to-peers networks reduce the overheads caused by routing protocols by forming a tunnel between source and destination machines.

A number of researchers have proposed various hybrid approaches to enhance the live migration process. Sahani et al [11] proposed a technique which detects the working set and then sends only the working set during the initially data transfer. After this Post-copy is initiated for the live migration and the reason for sending working set is that working set is most demanded section of the memory, which will be present on destination resulting in reduction of page faults. Lu et al [12] presented a hybrid of Pre and Post-copy by starting with Pre-copy, then sending the bitmap of dirty pages and finish with Post-copy which will just request pages that were marked as dirty. Hu et al [14] further enhanced the hybrid technique by introducing second bitmap to reduce the cache size for delta compression of dirty pages. Luo et al [26] combined Pre and Post-copy by starting with Pre-copy, then sending the bitmap of dirty pages and finish with Post-copy which will just request pages that were marked as dirty. Hirofuchi et al [24] proposed that the storage migrated over WAN should be demand based so that requested data gets served first and in the absence of any requests remaining data gets transferred.

Various other techniques have also been presented in order to improve the live migration. Shrivastava et al [13] proposed that the selection of the destination machines could be made on the bases of load on the edges and destination machines. Kashyap et al [10] proposed that reliability can be increased for Post-copy if source is updated asynchronously, so in the case of destination failure the source machine can takeover. Ye et al [25] proposed that resources should be pre-engaged on the destination machine so that resources get engaged in other activity and the migration fails. Jin et al [30] proposed that if the CPU is throttled down during migration the pages getting dirty could be reduced and migration will end sooner. Usually this technique is not a viable option as this violates Service Level Agreements.

Ma et al [27] presented an improvement of Pre-copy by flagging pages that get updated often and does not send get sent during iterations. Only in the last iteration the pages that have been flagged are sent so that network load could be reduced by not sending pages that we know will get dirty and have to be resent in the next iteration. Ramakrishnan et al [28] proposed methodology for the migration across datacenter. Scenario based methodologies include synchronous if data is sensitive otherwise asynchronous data transfer for faster data transfer. Tunnel is made between source and destination during migration so that data received at source could be directed to destination machine. Liu et al [19] proposed that instead of sending dirty pages we should send logs of changes occurring so that VM at destination replays logs until both machines are synchronized. Joe et al [29] states that most of the memory contents are also present on the storage which could be simultaneously copied along with data in the memory to speed up the memory transfer process. Narander et al [15] also presented that redundancy should be exploited between memory and storage to enhance the migration process.

Some research has been conducted on improving efficiency of cluster or gang migration of VMs. Deshpande et al [20, 21] performed delta compression among the virtual machines by first selecting a suitable virtual machine and ten calculating the delta for the rest of the virtual machines. Then the selected virtual machine is sent and for the rest of the virtual machines only delta is sent. On destination deltas are combined with the virtual machine and all the virtual machines are recreated. But in [21] the virtual machines are present within a rack instead of a single physical machine. The argument presented by the author is that physical machines in a rack are usually connected through a single switch due to which the rest of the network will not be overloaded and the latency will also be low. Jin et al [23] proposed a compression technique which analysis the type of data in the page and selects corresponding type of compression technique. This technique requires around 50% of the CPU to be available for processing which is usually not the case in cloud environments.

3. ESCDR Architecture

The aim of the ESCDR project is to provide an approach that helps in reducing the data transfer during first phase of VM migration. ESCDR focuses on certain type of content present in the memory for the VM. This content is the static portion of the memory that remains the same no matter when, how or where the VM got deployed. The key targets are the OS kernel code and drivers but will not be limited to these. The process has been divided into multiple steps that need to be performed in a sequence to achieve the desired results.

3.1 Maintain Database of VMs

Usually a Cloud environment has a database that has information regarding each VM that is running or offline. While there

might be scenarios where such database is not maintained or information stored is not sufficient then there is a need to maintain a separate database. The information stored in the database for each VM will have data regarding the version of the OS, hypervisor and the PM it is running on. Later when migration is taking place the information from the database will help identify which PM has exact same version of OS and hypervisor. When PMs having VMs with the same Version of OS and hypervisor are found other techniques to find shortest and least costly path can be utilized.

This database will also be utilized when new VMs are being initiated such that when a request for initializing a new VM is placed the candidate PMs will be that that do not contain a VM with same version of OS and hypervisor. In this way similar VMs will get spread across the datacenter so that when a VM migrates it will have more PMs to select from to migrate to.

3.2 Identification of Static Content

A two dimensional list will be maintained with the first dimension containing all the OS available for use in that cloud environment and the second dimension will contain all the hypervisors being used similar to Fig. 1. Each version of OS and hypervisor will be treated as a distinct entity and used separately in the list. When accessing the list the OS and hypervisor will act as keys to unlocking the element.

Once the list has been created it will be filled by markers identifying portions of the memory that remain static no matter when, how and where the VM was initiated. To identify these parts in the VM a delta of multiple VMs with the same OS and hypervisor will be generated and parts that remained unchanged throughout all the VMs will be marked as static content. This process will be repeated for all the entries in the list and stored in the list against the OS and hypervisors they were generated against.

```
1  list[OS 1][Hyp 1] = ids of static pages;
2  list[OS 1][Hyp 1] = ids of static pages;
3  list[OS 1][Hyp 3] = ids of static pages;
4  list[OS 2][Hyp 1] = ids of static pages;
5  list[OS 2][Hyp 2] = ids of static pages;
6  ----------------------------------------
7  ----------------------------------------
8  ----------------------------------------
9  list[OS n][Hyp n] = ids of static pages;
```

Figure 1. Two dimensional list of static pages in memory

3.3 Live Migration

A step has been added to the migration process, between the migration is initiated and it actual starts. In this step the OS and hypervisor versions will be used as keys to the list and identifiers stored against these keys will be acquired. After acquiring the markers the information gets shared between source and destination PMs. At the source pages in the memory identified as static get marked so that these pages do not get sent to destination. Fig. 2 presents description of the flow of events.

While at the destination PM, the VM due to which this PM got selected due to having similar OS and hypervisor to source VM gets selected. From the selected VM based on the markers used for the identification of static content, memory pages get copied to destination VM. This process occurs inside the same memory of the destination PM.

4. ESCDR Analysed

At first ESCDR might seem too complex to be feasible but it must be noted that after the technique has been implemented the benefits are long lasting. The benefits during a single migration are small but they are constant and in a large datacenter where thousands of VMs are migrating, the combined reduction in network load and migration time will be significantly large.

The benefits of deploying ESCDR are directly proportional to the number of migrations performed. Reason behind the benefits

reduced network load and migration time being directly proportional to the number of migrations performed is because the implementation is a onetime investment at a centralized location with a single database and list.

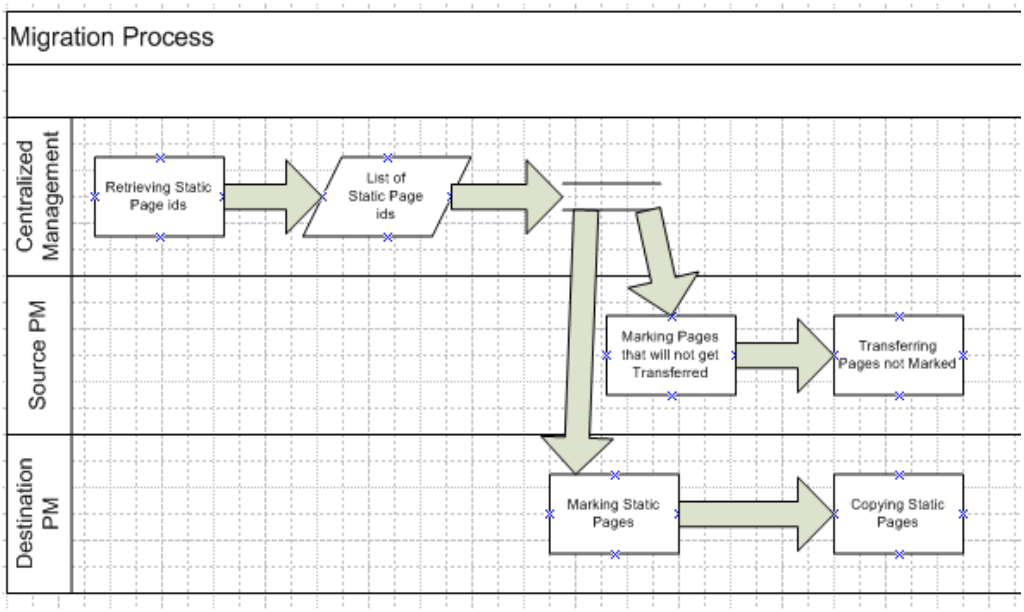

Figure 2. Migration Process

Another factor involved in the implementation of the ESCDR is that it does not need to be implemented into the VMs of the datacenter. This reduces the complexity in implementation as ESCDR becomes a distinct entity and implementation of ESCDR will not result in conflicts with current systems.

Acknowledgement

This work is supported by Higher Education Commission under the funding of Indigenous 5000. Also thanking National University of Computer & Emerging Sciences for their cooperation and support in completing this work.

References

[1] Ahmad, R. W. Gani, A. Hamid, S. H. A.. Shiraz, M . Xia, F Madani, S. A. (2015).Virtual machine migration in cloud data centers: a review, taxonomy, and open research issues, *Journal of Supercomputing*, 71 (7) 2473-2515, July.

[2] Strunk, A. (2012). Costs of Virtual Machine Live Migration: A Survey, 8th IEEE World Congress on SERVICES, p. 323-329, June 2012

[3] Svärd, P. Hudzia, B.Tordsson, J. Elmroth, E. (2011). Evaluation of Delta Compression Techniques for Efficient Live Migration of Large Virtual Machines, ACM Iinternational conference on VEE, 46 (7) 111-120.

[4] Nelson, M. Lim, Beng-Hong Hutchins, G. (2005). Fast Transparent Migration for Virtual Machines, USENIX annual conference on ATEC, p. 391–394.

[5] Clark, C., Fraser, K., Hand, S., Gorm Hansen, J., Jul, E., Limpach, C., Pratt, I., Warfield, A. (2005). Live Migration of Virtual Machines, 2nd USENIX Conference on NSDI, p. 273-286.

[6] Michael, R., Hines, K., Gopalan, (2009). Post-Copy Based Live Virtual Machine Migration Using Adaptive Pre-Paging and Dynamic Self-Ballooning, ACM International conference on VEE, p. 51-60.

[7] Michael R., Hines, U., Deshpande, K., Gopalan, (2009). Post-Copy Live Migration of Virtual Machines, ACM SIGOPS Operating Systems Review, p. 14-26, 2009

[8] Bradford, R., Kotsovinos, E., Feldmann, A., Schi¨oberg, H. (2007). Live Wide-Area Migration of Virtual Machines Including Local Persistent State, 3rd ACM international conference on VEE, p. 169-179.

[9] Zhang , X., Huo, Z., Ma, J., Meng, D. (2010). Exploiting Data Deduplication to Accelerate Live Virtual Machine Migration, IEEE International Conference on CLUSTER, p. 88-96, Sep. 2010.

[10] Kashyap, S., Dhillon, J.S., Purini, S.(2014). RLC - A Reliable approach to Fast and Efficient Live Migration of Virtual Machines in the Clouds, IEEE International Conference on CLOUD, p. 360-367, 2014

[11] Sahni, S., Varma, V.(2012). A Hybrid Approach To Live Migration Of Virtual Machines, IEEE International Conference on CCEM, p. 1-5, Oct. 2012

[12] Lu, P., Barbalace, A., Ravindran, B (2013). HSG-LM: Hybrid-Copy Speculative Guest OS Live Migration without Hypervisor, 6th ACM International conference on SYSTOR, Article No. 2, 2013

[13] Shrivastava, V., Zerfos, P., Lee, Kang-won, Jamjoom, H., Liu , Yew-Huey.,Banerjee, S. (2011). Application-aware Virtual Machine Migration in Data Centers, IEEE Conference on INFOCOM, p. 66-70, April 2011

[14] Hu, L., Zhao, J., Xu, G., Ding, Y., Chu, J.(2013). HMDC: Live Virtual Machine Migration Based on Hybrid Memory Copy and Delta Compression, *International Journal on AMIS*, p. 639-646, 2013.

[15] Narander, K., Swati,S (2014).An Efficient Live VM Migration Technique in Clustered Datacenters, *Research Journal on Recent Sciences*, p. 13-20, Sep. 2014.

[16] Hacking, S., Hudzia, B. (2009). Improving the Live Migration Process of Large Enterprise Applications, 3rd ACM International workshop on VTDC, p. 51-58.

[17] Huang, W., Gao, Q., Liu, J., Panda, D. K (2007). High Performance Virtual Machine Migration with RDMA over Modern Interconnects," IEEE International Conference on Cluster Computing, p. 11-20, Sep. 2007.

[18] Isci, C.. Liu, J., Abali, B., Kephart, J. O., Kouloheris, J. (2011). Improving server utilization using fast virtual machine migration," IBM Journal of Research and Development, 1 (4) 4-12.

[19] Liu, H., Jin, H., Liao, X.,Yu, C., Xu, Cheng-Zhong (2011). Live Virtual Machine Migration via Asynchronous Replication and State Synchronization, IEEE Transactions on Parallel and Distributed Systems, p. 1986-1999, Dec. 2011

[20] Deshpande, U., Wang, X., Gopalan, K (2011). Live Gang Migration of Virtual Machines, 20th ACM international symposium on HDPC, p. 135-146.

[21] Deshpande, U., Wang, X., Gopalan, K (2012). Inter-rack Live Migration of Multiple Virtual Machines, 6th ACM international workshop on VTDC, p. 19-26.

[22] Travostino, F., Daspit, P., Gommans, L., Jog, C., de Laat, C., Mambretti, J., Monga, van Oudenaarde, I.B., Raghunath, S., Wang, P. Y. (2006). Seamless live migration of virtual machines over the MAN/WAN, *Journal of FGCS*, 22 (8) 901–907, Oct. 2006

[23] Jin, H., Deng, L., Wu, S., Shi, X., Pan, X. (2009). Live Virtual Machine Migration with Adaptive Memory Compression, IEEE International Conference on CLUSTER, p. 1 - 10.

[24] Hirofuchi, T., Ogawa, H., Nakada, H., Itoh, S., Sekiguchi, S (2009). A Live Storage Migration Mechanism over WAN for Relocatable Virtual Machine Services on Clouds," IEEE/ACM International Symposium on CCGRID, p. 460-465, May 2009.

[25] Ye, K., Jiang, X., Huang, D., Chen, J., Wang, B. (2011). Live Migration of Multiple Virtual Machines with Resource Reservation in Cloud Computing Environments, 4th IEEE International Conference on Cloud, p. 267–274, July 2011.

[26] Luo, Y., Zhang, B., Wang,X., Wang, Z., Sun, Y., Chen, H (2008). Live and Incremental Whole-System Migration of Virtual Machines Using Block-Bitmap," IEEE International Conference on Cluster Computing, p. 99–106, Oct. 2008.

[27] Ma, F., Liu, F., Liu, Z.Live Virtual Machine Migration based on Improved Pre-copy Approach, IEEE International Conference on ICSESS, p. 230–233, July 2010.

[28] Ramakrishnan, K. K. Shenoy, P.. Van der Merwe, J (2007). Live Data Center Migration across WANs: A Robust Cooperative Context Aware Approach, ACM Workshop on INM, p. 262-267, 2007.

[29] Jo, C. Gustafsson, E. Son, J. Egger, B. (2013). Efficient Live Migration of Virtual Machines Using Shared Storage, 9th ACM international conference on VEE, p. 41-50.