A Security State Transfer Model for Virtual Machine Migration in Cloud Infrastructure

Santosh Kumar Majhi
Department of Computer Science and Engineering
VSS University of Technology, Burla, India

Sunil Kumar Dhal
Faculty of Management Studies
Sri Sri University, Cuttack, Odisha, India

ABSTRACT
Virtual machine migration (VMM) is one of important services is used as a tool to facilitate system maintenance, load balancing, fault tolerance, on-demand service offerings. Live VMM transfers an active Virtual Machine (VM) from one physical host to another across different data centres. It involves a sequence of operations in iteration under a specific protocol/method for migrating execution context (active memory) and control data of a VM to the destination machine. These operations are dependent on data transfer schedule, availability of resources and overall timing constraints. The migration process is implemented by establishing shared network storage and/or a network communication channel. Along with the execution context the security configuration of the VM need to be transferred. In this paper we have proposed a security context migration framework. Both the static as well as dynamic security context is considered for migration.

General Terms
Cloud Computing, Security

Keywords
VM Migration, Security Context, Security Configuration

1. INTRODUCTION
Cloud computing enables easy, on-demand and efficient utilization of resources (storage, CPU, network) to develop and remotely execute large number of user applications across various domains in a virtualized platform. The Virtual machine (VM) migration is the process of transferring run time context and process states of a VM from a source physical machine to a target machine under the control of hypervisor. In IaaS cloud, VM migration is used to facilitate load balancing, server maintenance, improved network scalability and storage backup. Some other factors like unplanned downtime, lack of critical resources, underlying hardware failure or individual VM failure, server failure are the reason behind VM migration. In IaaS Cloud, the VM migration is performed as user demands more scalability on resources. Therefore, the main objective of migration is to provide high accessibility of resources to the customers, ensuring quality of service (QoS) within a allowable time-span.

A virtual machine is an abstract execution system with assigned pool of resources (computing, storage and network) and set of applications running on the resources. An important proporty of a virtual machine is that the running applications in VM must be compliant with limited resources and abstractions facilitated by the VM. Virtual machines are categorized into system VM and process VM, based on their usage and degree of communication with any physical machine. A VM provides a complete system platform for supporting the execution of the operating system.

Cloud operating system allows creating multiple virtual machines with sharing the underlying physical machine resources amongst them where each of the VM may run its own operating system. Hypervisor is a software module that runs over the cloud OS providing the virtualization platform for VMs. It virtualizes all of the resources of a physical machine, thereby defining and supporting the execution of multiple virtual machines in a single physical machine. The widely used cloud operating systems are Openstack, Microsoft HyperV, Oracle Virtual box. Openstack supports different virtualization technologies to build and run VMs. It is evident that a number of virtual machines can be created and executed in a single physical machine and a group of virtual machines across the data centres can communicate with each other for serving a particular task. The communication between VMs and other system level operations are supported by different services in the cloud OS.

Virtual machine migration (VMM) is one of such important services that is used as a tool to facilitate system maintenance, load balancing, fault tolerance, on-demand service offerings. Live VMM transfers an active VM from one physical machine to another across different data centres under the control of hypervisor. It involves a sequence of operations in iterations under a specific protocol/method for migrating execution context (active memory) and control data of a running VM to the destination machine. The methods used for migrating active memory pages are (i) pre-copy; (ii) post-copy and (iii) hybrid. On the other hand, well accepted method for transferring common migration data is EDAMP method [1]. In our framework, we have extended this approach for migrating security context of VM.

2. LITERATURE REVIEW
In this section, we present the literature review with respect to two main directions: Virtual machine migration in cloud computing and analysis of security context in VM migration. In addition, we have studied research on consistency analysis for detecting and resolving anomalies and conflicts within a given security policy configuration. Here, we present some of the most relevant contributions in all research areas.

The live VM migration across subnet based on overlay approaches discussed in [4] [5] [6]. They may require mobile IP support [7]. Even so, Mobile IP has not well suited for the offline VM migration in the large network [8]. The overlay approach considers the large network as homogeneous network. However, practically, data centers deployed with heterogeneous environments. Greenberg et al. [8] discussed the interconnection among data centers to allow unseamed

VM migration. The migration within the data center or across the data center is possible by use of the Open Flow [8] network controller. Mysore et al. [9] discussed heterogeneity of data centers and single CSP. The VM migration changes the target system at the hypervisor or Operating System level is popularly known as host based approaches. Many researchers have also proposed the host based VM migration [9] [10] [11] [12] [13]. The transfer of disks over WAN (Wide Area Network) has been proposed by Bradford et al. [14]. Pu et al. [15] discussed the VM migration by the modification of guest VM, which establish a connection with Open Flow Virtual switch [11]. The source and target systems consist of similar environment (i.e., homogeneous) for VM migration. The VL2 system discussed the creation of overlay network using the location address (LA). A centralized directory (CD) is maintained for mapping the LA with the application address. The OS traps the ARP requests originating from end users and forwards to the CD for un-interrupted VM migration [9]. It does not address the problem of multiple data centers where the CD may not have control for all data centers over the network. Similarly, a hybrid approach has been discussed by [10] [14], where the target host changes its state, and mobile IP is required for VM migration. However, the above discussed approaches only use the homogeneous cloud environment. However, practically the platform is not uniform.

The network-based migration approaches propose the layer2 network and layer3 network for VM migration. The approaches, discuss (i) the replacement of existing data center architecture with layer 3 routs to overcome the disadvantages of layer 2 routes [15] and (ii) the creation of layer 2 networks over various data centers by the extension of layer 2 technologies, e.g., layer2 VPN [16] and OTV [17]. Mobile IP [10] [18] discuss the requirement of new IP addresses after migration. It does not require the existing IP address. However, Mann et al. [8] present a new approach to combine the advantages of layer2, and layer3 to address the triangular routing problem. The Openflow based systems [17] [19] resolve issues related to living VM migration over data centers. In the above discussion, researchers address the problem of VM migration across data centers from diverse aspects.

Network elements such as stateful firewalls contribute in enforcing security in Cloud platform. During VM migration the security enforced to the VM, need to be transferred to the destination machine. Zahra et al [2] presented a framework for security context migration in a firewall secured VM environment. In their work, the security context transfer approach along with implementation has been discussed. The Security Context (SC) module in the hypervisor extracts SC of source VM and transfer the SC to destination host along with VM state. SC migrator establishes a connection (TCP) between source and destination. At the destination side, the SC module enforces the security context. Authors evaluated the work using three test case scenarios. In the first test case, a VM migration was performed without any SC information. In the second test case, a VM was migrated with static SC information. In the third test case, migrated a VM with static and dynamic SC information. The test cases are not enough to proof the correctness and consistency of the security context migration. In addition, this method considers IP address does not change after VM migration to a different host.

3. OVERVIEW OF PRECOPY AND POSTCOPY APPROACH

3.1 Precopy approach

In this method, firstly, the Hypervisor typically copies/transfered the dirty or modified memory pages from source to target host while the VM is executing at the source. Then, the modified memory pages ('dirty pages') are iteratively copied until the rate of page being dirtied reaches to a threshold value [3]. Figure 1 shows the flow of pre-copy method.

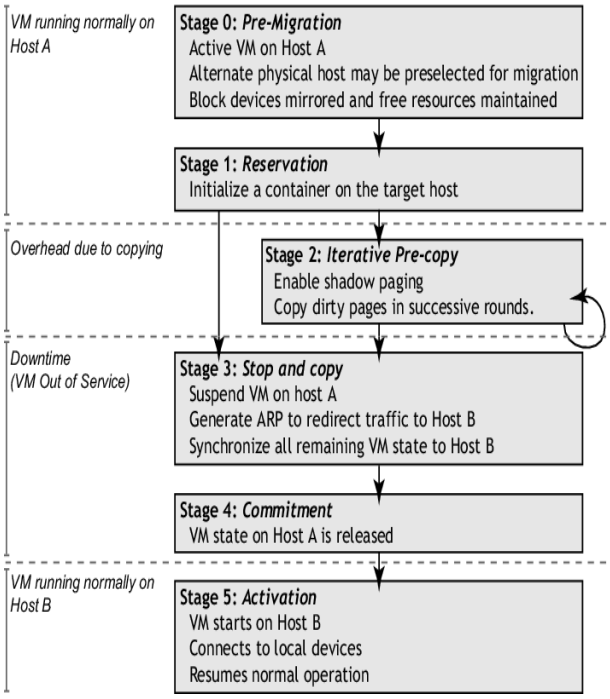

Fig 1: Precopy Memory Page Transfer [3]

3.2 Postcopy Approach

This method is initiated by suspending the migrating VM at the source machine. Then, a minimal subset of the current execution context the VM (CPU state, registers and, non-pageable memory) is copied to the target. The VM is then presumed at the destination. Concurrently, the source host actively pushes the remaining memory pages of the VM to the destination machine. This activity is known as pre-paging. On the other hand, if the running VM at the target want to access a page that has not yet been available, a page-fault occurs. These page-faults are reported at the destination and redirected to the source host, which then sends the requested pages. Figure 2 shows the operational flow of post-copy method. An ideal pre-paging scheme would mask majority of page-faults, although its performance depends upon the memory access pattern of the VM's workload [20].

In post-copy method, each page is exactly transferred once over the network. In contrast, pre-copy can transfer the same page multiple times if the page is dirtied repeatedly at the source during migration. On the other hand, pre-copy retains an up-to-date state of the VM at the source during migration, whereas with post-copy, the VM's state is distributed over both source and destination. If the destination fails during migration, pre-copy method can recover the VM, whereas the same is not possible in post-copy method [20].

4. PROPOSED SECURITY CONTEXT MIGRATION MODEL

The proposed security context migration method exports the enforced security context such as firewall filtering rules, connection tracking information and IPsec state information from the source machine to the destination machine. Our proposed method allows generating a difference set of security context between source and destination machine. This set can be identified by a set difference operation of the entire source VM against the product set, which is an intersection between the source and destination VM. After the set calculation, our proposed method transfers the difference set of the source VM and overwrite it into the destination VM. The detail of the proposed method overview is presented in figure 3.

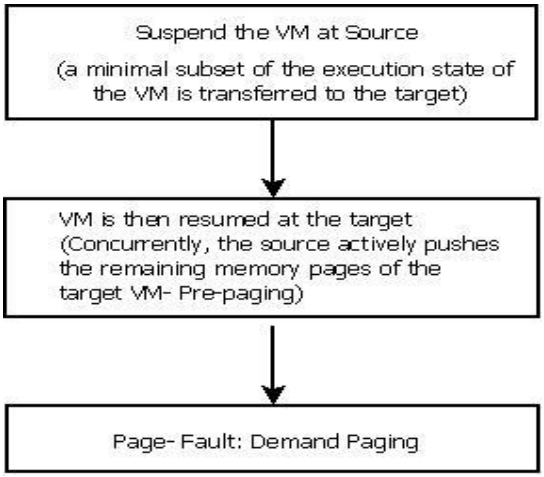

Fig 2: Post Copy Memory page Transfer

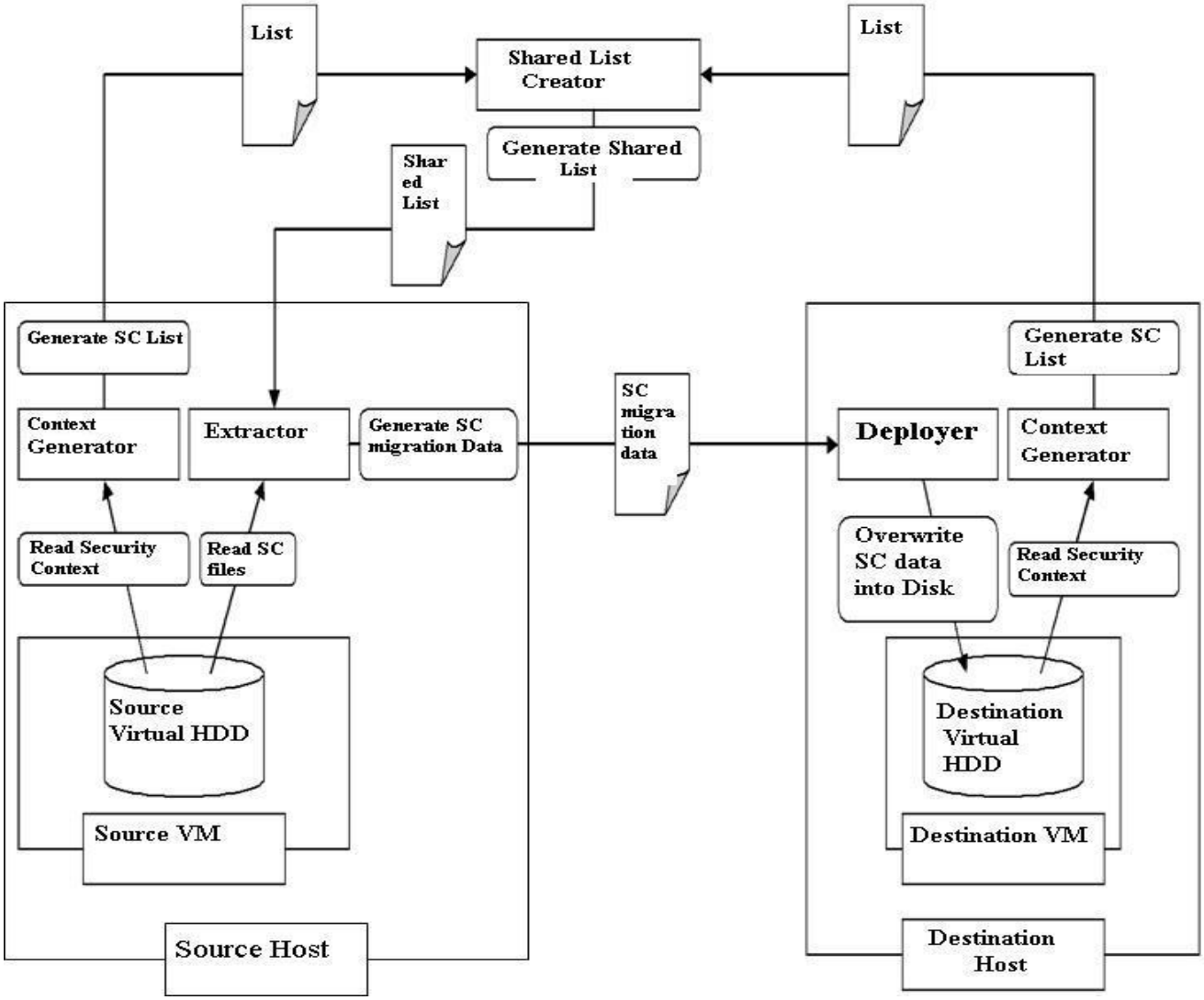

Fig 3: Security State Transfer Model

The proposed method consists of five phases to complete the task of security context migration.

1. Initialization: The framework generates a list of security context files on each VM based on both host and network security data.
2. Shared List Creation: To determine the difference in security data between source VM and destination VM, a shared list is created, which contains the product set of security data of source and destination VM.
3. Setup Phase: The source VM identifies the dependent process on the security data. Along with memory the security context need to be transferred. The security configuration may impact the availability of services. So, before transferring memory, security analysis is required for safe and correct migration.
4. Extraction: The extractor module generates the difference set of security data from the shared list and current context data. It can identify the files that do not exist on the destination VM because the shared list contains only shared files. The identified files are added to the migration data. The migration data contain the difference set of the source VM that includes the files changed (added or modified) before and after the setup phase.
5. Deployment: The extracted security data is deployed into the destination VM and physical host.

The key component of the system along with their functionality is described in Table 1. The key components are the separate implementable module in the cloud platform.

Table 1: Key Components of Migration Module

Sl. No.	Components/Modules	Function
1	Host	Provides platform for creating/managing VM
2	VM	Provides a virtual computing environment
3	Context Generator	Generate the context (security) from the VM by reading security context data. This module is present at every host. In our framework we have shown the presence of this module at source and destination.
4	Extractor	The extractor module read the security context file from source virtual hard disk drive (VHDD) and compare with the shared list then generates the required security context for migration.
5	Shared List Creator	It takes the input files from the source and destination Host and VM then generates a file that contains the common attributes present at both machines.
6	Deployer	This module deploy the migrated security context in the destination VM

This is a prototype for security context migration. This prototype can be deployed in the hypervisor to facilitate the VM migration. Along with the memory page transfer the security context can be combined. In addition to this the security compatibility can be verified in beforehand at the extractor module to avoid the failure of VM migration due to the incompatible security data at destination. Here we present only a prototype of the system.

5. CONCLUSION

We discussed briefly about the VM migration and its techniques. We proposed a security context migration method and described the components of the prototype. The extractor module plays an important role in the framework. This method is under development phase. We are working to develop a module to implement on top of hypervisor.

6. REFERENCES

[1] Yuki Ashino and Masayuki Nakae, "Virtual Machine Migration Method between Different Hypervisor Implementations and its Evaluation", 26th International Conference on Advanced Information Networking and Applications Workshops, 2012.

[2] Zahra Tavakoli et al.,"A Framework for Security Context Migration in a Firewall Secured Virtual Machine Environment", In the IFIP, LNCS Vol. 7479, 2013.

[3] Clark et al.,"Live Migration of Virtual Machines", In the Proc. of USENIX Symposium on Networked Systems Design, 2005.

[4] X. Jiang et al.,"Violin: Virtual internetworking on overlay infrastructures," *in ISPA*, 2005.

[5] "VMware Vnetwork distributed switches", https://www.vmware.com/products/vnetworkdistributedswitch/overview.html.

[6] "Cisco Nexus 1000V Series Switches," http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9902/data sheet c78492971.html.

[7] VMware VMotion for Live Migration of virtual machines, http://www.vmware.com/products/vi/vc/vmotion.html, July 14, 2014.

[8] V. Mann et al., "CrossRoad: Seamless VM Mobility Across Data Centres through Software Defined Networking", *IEEE NOMS*, 2012.

[9] A. Greenberg et al., "VL2:A Scalable and Flexible Data Center Network," *in ACM SIGCOMM*, 2009.

[10] E. Silvera et al., "IP mobility to support live migration of virtual machine across subnets," *in SYSTOR,* May 2009.

[11] "Open vSwitch - Open Virtual Switch," http://www.openvswitch.org

[12] Bradford et al., "Live wide-area migration of virtual machines including local persistent state," *in ACM VEE*, 2007.

[13] Y. Pu et al., "Cloud rack: Enhanced virtual topology migration approach with open vswitch," *in IEEE ICOIN*, 2011.

[14] C. Kim et al., "Floodless in seattle: a scalable ethernet architecture forlarge enterprises," *in ACM SIGCOMM*, 2008.

[15] R. Mysore et al., "PortLand: A Scalable Fault-Tolerant Layer 2 DataCenter Network Fabric," *in ACM SIGCOMM* ,2009.

[16] "Virtual Private LAN Services (VPLS)," http://www.cisco.com/en/US/products/ps6648/ products ios protocol option home.html.

[17] B. Boughzala et al., "Openflow supporting inter-domain virtual machine migration," *in IEEE/IFIP WOCN*, 2011.

[18] E. Harney et al., "The efficacy of live virtual machine migrations over the internet," *in ACM VTDC*, 2007.

[19] F. Hao et al., "Enhancing dynamic cloud-based services using network virtualization," *ACM SIGCOMM CCR*, 2010.

[20] Live migration, Available online: https://en.wikipedia.org/wiki/Live_migration