International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 12, December-2014

Security in Live Migration of Virtual Machine with Automated Load Balancing

Anupam Tamrakar
Dept. of Computer Science
NIT Allahabad
Allahabad, India

Abstract **— Virtualization refers to creating virtual machines having separate execution environment for operating systems running under same hardware resource. One important feature of virtualization is live virtual machine migration which serves us several advantages. It enables organisation to migrate running VM from one physical host to another without interrupting the application running onto it. Live migration Technique is also vulnerable to network attack as data can be sniffed while migrating Virtual machines and security becomes the major concern as confidentiality of users can be lost. To solve it, IPSEC tunnel can be deployed between source and destination physical system for migrating virtual machine between them which is more secure.**

With data centre size, workload on system administration also increases. This is because migration is initiated manually. With sudden workload, reshuffling of virtual machines, re-balancing the physical system load, need for automating live migration process increases. Our work will focus on exploiting the live migration process, provide secure solution for migration and automate it which will decrease the burden from system administration and enhances the reliability in customers.

***Keywords*— Live VM Migration, XEN Hypervisor, IPSEC Security, Load-Balancing.**

I. Introduction

Today, many organizations are moving towards the cloud computing technology offered for getting the services like hardware and software. Virtualization technology is used to achieve the scalability property of cloud computing system. The use of virtualization has become increasingly popular in organizational network.

The operators and administrators are turning to live migration of VM (Virtual Machine). It allows separation between hardware and software. Live migration of VM is the process of moving a VM from one physical host to another with little or no downtime for services hosted by virtual machine. The live migration functionality is provided by vendors such as Xen, VMware, OpenVZ and VMotion. Due to increased demand of virtualization technology, it is essential to provide live migration of virtual machine in a more secured manner. Virtual machine is typically stored as regular file on the disk and this file is migrated from one host to another using network system. The network across which VM instance is migrated is not entirely secure. Secured live migration of VM is required because the attacker inside a network employing live migration can facilitate untrusted access to migrating VM image [1]. The attackers can view or modify the data associated with VM instance.

Security in Virtualization is the main concern. Several attacks can be perform on virtual machines both from outside world and from management environment as it holds the highest privilege to access the guest OS. It is considered difficult to avoid these attacks but we can manage our own shield to protect virtual machine from the attackers by creating secure tunnel for live migration virtual machine in XEN virtualization.

II. Related Work

The problems existing with process migration is residual dependencies that migrated process retains in machine from where it was migrated. Residual dependencies include open file descriptors, process control block details unmodified, other resources. This problem degrades the performance of migrated process.

Process migration moves a process from one physical host to another. Zap [2] uses partial OS virtualization for migration of process domains (pods) with the help of a Linux kernel. VMware added VMotion [3] as an advantage for OS migration to their Virtual Centre management software. In contrast to process migration, OS migration handles all the limitations of the process migration and still does the VM migration efficiently. The difference between OS migration and process migration is that operating system migration overcomes the problem of residual dependencies with the help of a narrow interface between a virtualized OS and the hypervisor. The administrator need not be concerned with what is running within the VM; instead they can migrate the OS and its associated processes as one unit.

Konig and Steinmetz show that the round-trip time (RTT) of ICMP packets is a promising metric for remotely detecting VM migration processes. By targeting a VM with ICMP packets, they can determine when that specific VM is migrating to another physical machine [4]. NomadBIOS run several linux instances concurrently with ability to migrate; it runs on top of L4 linux kernel [5].

Jon Oberheide [6] using his tool Xensploit, performs man-in-middle attack in live migration of virtual machine which exploit the vulnerabilities exist in migration process and manipulate the memory contents of VM as it traverses through network. H. Jin et al [7] work on compressing the memory pages sent over the network which eventually decrements total migration time. In [8], authors explain load-sharing mechanism, sender and receiver scheme. An overloaded node looks to share its load and lightly loaded nodes take initiative to look for their work.

In [9], authors develop pull and push strategy for automate the live VM migration scheme. In push strategy, machine try to get rid of running VM and in pull strategy, physical system try to steal running VM from another machine. Strategy deployed can be balance the entire system for work load.

III. BACKGROUND

A. *Live Migration*

Virtualization technique allows us to run multiple operating systems within same hardware resource. When this hardware resource will get powered down for maintenance purpose, then virtual machines running onto it will need to be migrated to another physical server with proper execution without halting. This Virtual machine migration with minimum downtime is termed as live VM migration. Downtime is the period during which the service is unavailable due to there being no currently executing instance of the VM. This time is directly visible to client as no running instance of their VM can be found.

B. *Live Migration Techniques*

Migration Techniques involves migration of CPU, memory states, hardware device states running VM from one host to another. Migration technique differs in order of state transfer.

1) *Stop and Copy:* In stop and copy mechanism, running virtual machine on one host is halted using SAVE command and memory states, hardware states, and CPU content is stored in an image file. This image file is then transferred to another host using any secure mechanism where it is restored to resume normal processing using RESTORE command. Meanwhile state of VM remains halted. Its cost is large migration and downtime.

2) *Pre-copy Migration:* Most hypervisors like xen, kvm, vmware uses pre-copy migration approach. It is a simple approach where memory pages are transferred in iterative fashion; if memory page got dirty then it will again transfer in next iteration. This process continues until small amount of pages are left which will be transferred as stop and copy phase. Benefit is reduced downtime as memory pages resides in another host before the VM relocation.

3) *Post Copy:* VM migration is initiated by suspending the VM at the source. With the VM suspended, a minimal subset of the execution state of the VM (CPU registers and non-pageable memory) is transferred to the target. The VM is then resumed at the target, even though most of the memory state of the VM still resides at the source. At the target, when the VM tries to access pages that have not yet been transferred, it generates page-faults. These faults are trapped at the target and redirected towards the source over the network.

As in warm up phase, each and every page need to be copied while machine is still running. There may be multiple applications running simultaneously at this running VM. Therefore, securities of memory pages transfer become issue. If it is unencrypted, then memory pages can easily be intercepted through any network cryptanalyst attack as all these pages will be travelling through unsecure channel. Man-in-middle attacker can easily intercept and read the memory contents of pages. There exist needs of creating a secure channel through which all these memory pages will travel and are kept away from the reach of network attackers.

C. *Live Migration benefits*

1) Live VM migration finds its importance in load balancing among different physical servers. When cpu load of physical server increases, then VMs running onto it will be migrated to lightly loaded servers for load balancing.
2) Transparent movement of Virtual machine.
3) Live migration removes the problem of residual dependencies of Virtual machines.

D. *Requirements of Live VM Migration*

1) The two hosts should be configured with the same version of Xen. Both hosts should be in the same layer 2 network and IP subnet.
2) The disk image and the configuration file should be stored on the shared storage. Both hosts should have access to it.
3) Xen should be installed on same type of processor. Migration is not supported for different types.
4) The Xen configuration for the Xen daemon is stored in the file xend-configure.sxp. This file needs to be modified on both hosts.

IV. EXPERIMENTAL SETUP

A. *For performing live VM migration iSCSI (internet Small Computer System Interface) shared storage is used.*

ISCSI target is created at a physical system, which is having storage capacity at-least to store a VM image; we also have to tell our target that we are connecting storage to different initiator’s ip-addresses. Different physical systems using open-iscsi as initiator to access that shared storage.

B. *IPsec Tunnel Establishment:*

IPsec implementation in Linux for creating tunnel between two hosts is done through OpenSwan which is IPsec package released under GNU license. We need to configure few basic settings like enable ip-forwarding net.ipv4.ip_forward = 1**,** add some iptable rules that modify the source ip address to be sent out. IPsec authentication also needs to set in IPsec. Secrets file where secret key is shared between source and destination.

C. Test Bed for live VM migration :

1) Hardware:

Two HP Compaq dx7200 with Intel Pentium 4HT processor installed with Ubuntu 12.04 and a Dell Inspiron N4110 with Intel core i3 installed with Ubuntu 13.10 configured with XEN hypervisor is used as servers. One HP Compaq system is configured with ISCSI shared storage which acts as target for other HP system and Dell Inspiron initiators. Both the systems are in same subnet.

2) Software:

To perform live VM migration Xen hypervisor is chosen. Xen hypervisor is installed at all servers through which multiple VMs can run. Ubuntu 10.04 (Lucid), Ubuntu12.04 and Debian Lenny are installed as virtual machine without GUI, it will be run through command-line which is fairly easy to configure. Wireshark packet capturing tool is also installed at central server which captures migrating packets and analyse them for VM security.

D. Demonstration:

To perform live capture using wireshark, migrating VM will run a small script which continuously prints certain message. As live VM migration starts using "xm migrate –live <VM name> <destination ip>" command, this small script keeps running on VM. At this time, any third person in same subnet can capture the migrating packets and read the packet details which is not in encrypted form. As the data of VM running script is vulnerable. When migration completes, VM resides on another server which is uninterrupted having same script running.

Live VM migration is not secure as data travels through unsecure channel, confidentiality of user data in running VM is at risk as it can be sniffed easily and manipulated. Thus, data theft is becoming first priority and need to boost network security increases. For live VM migration, a secure way is to migrate VM through a secure channel, which can be created using IPsec tunnel. IPsec encrypts each and every packet travelling through network and provides secure authentication using protocol stack. But IPsec exerts extra overhead in execution time but it is much more secure in nature and prevents sniffing of data.

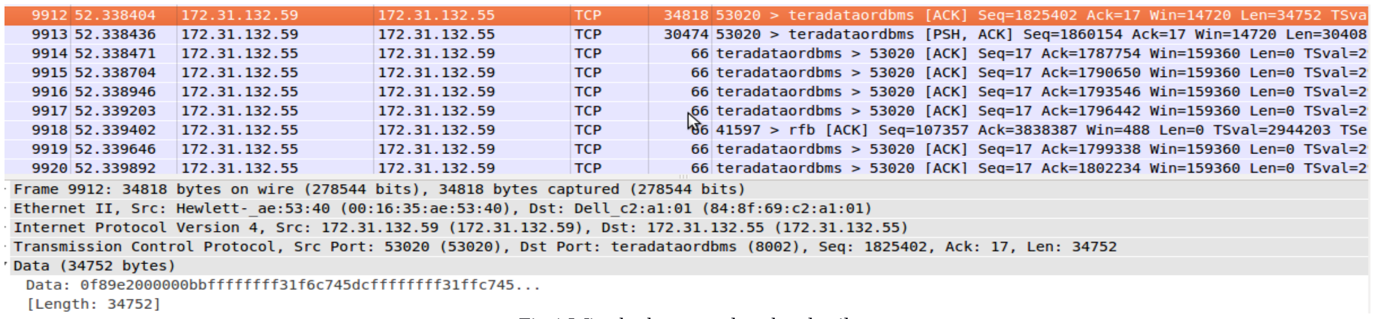
Some other mechanism can also be used to reduce the total migration time and downtime. We can compress the memory pages of VM and later send through network, this will reduce the size of entire VM and migration time will be less. Another way of securing is to encrypt the entire VM before migrating with any of the encryption algorithm then migrate the VM but here total migration time will be high.

Researches are going on about how to reduce the downtime and total migration time which in turn increase the efficiency of live migration process using XEN hypervisor. Also, can we devise any other tool for secure tunnelling of network packets at the time of migration process?

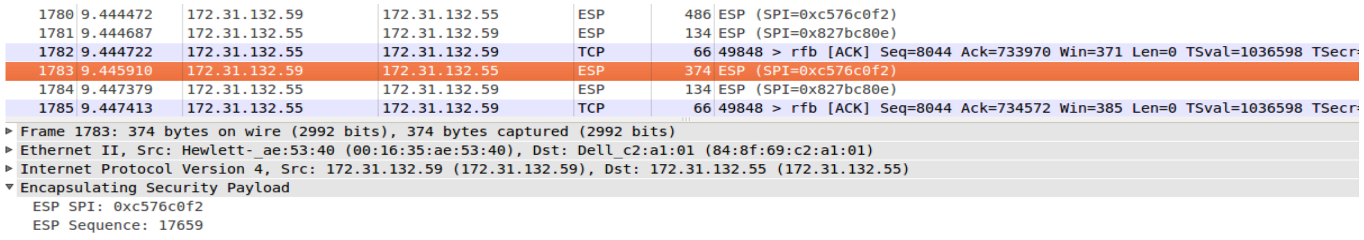
V. RESULT & ANALYSIS:

Demonstration shows that while VM migration, data packets travels through network, sniffer can sniff the packet details using wireshark tool and exploit it.

Fig 1 shows the packet details after capturing it without having IPsec implemented for secure tunnelling, here we can clearly see the details and data exist in the packet, though the data is in different format, we can convert it into human readable format.


```
9912 52.338404  172.31.132.59   172.31.132.55   TCP   34818 53020 > teradataordbms [ACK] Seq=1825402 Ack=17 Win=14720 Len=34752 TSva
9913 52.338436  172.31.132.59   172.31.132.55   TCP   30474 53020 > teradataordbms [PSH, ACK] Seq=1860154 Ack=17 Win=14720 Len=30408
9914 52.338471  172.31.132.55   172.31.132.59   TCP      66 teradataordbms > 53020 [ACK] Seq=17 Ack=1787754 Win=159360 Len=0 TSval=2
9915 52.338704  172.31.132.55   172.31.132.59   TCP      66 teradataordbms > 53020 [ACK] Seq=17 Ack=1790650 Win=159360 Len=0 TSval=2
9916 52.338946  172.31.132.55   172.31.132.59   TCP      66 teradataordbms > 53020 [ACK] Seq=17 Ack=1793546 Win=159360 Len=0 TSval=2
9917 52.339203  172.31.132.55   172.31.132.59   TCP      66 teradataordbms > 53020 [ACK] Seq=17 Ack=1796442 Win=159360 Len=0 TSval=2
9918 52.339402  172.31.132.55   172.31.132.59   TCP      66 41597 > rfb [ACK] Seq=107357 Ack=3838387 Win=488 Len=0 TSval=2944203 TSe
9919 52.339646  172.31.132.55   172.31.132.59   TCP      66 teradataordbms > 53020 [ACK] Seq=17 Ack=1799338 Win=159360 Len=0 TSval=2
9920 52.339892  172.31.132.55   172.31.132.59   TCP      66 teradataordbms > 53020 [ACK] Seq=17 Ack=1802234 Win=159360 Len=0 TSval=2
Frame 9912: 34818 bytes on wire (278544 bits), 34818 bytes captured (278544 bits)
Ethernet II, Src: Hewlett-_ae:53:40 (00:16:35:ae:53:40), Dst: Dell_c2:a1:01 (84:8f:69:c2:a1:01)
Internet Protocol Version 4, Src: 172.31.132.59 (172.31.132.59), Dst: 172.31.132.55 (172.31.132.55)
Transmission Control Protocol, Src Port: 53020 (53020), Dst Port: teradataordbms (8002), Seq: 1825402, Ack: 17, Len: 34752
Data (34752 bytes)
  Data: 0f89e2000000bbffffffff31f6c745dcffffffff31ffc745...
  [Length: 34752]
```
Fig 1 Wireshark captured packet detail

After establishing IPsec tunnel between source and destination host, migrating packets will travel through this secure tunnel and when we capture the packet using same way using wireshark tool, we have seen that these packets are more secure than normal packets and not able to retrieve any data from here. Packets having authentication header and ESP which provide security at network layer instead of application layer. Fig 2 shows packet details when IPsec tunnel is established between two migrated hosts.



```
1780 9.444472   172.31.132.59    172.31.132.55    ESP    486 ESP (SPI=0xc576c0f2)
1781 9.444687   172.31.132.55    172.31.132.59    ESP    134 ESP (SPI=0x827bc80e)
1782 9.444722   172.31.132.55    172.31.132.59    TCP     66 49848 > rfb [ACK] Seq=8044 Ack=733970 Win=371 Len=0 TSval=1036598 TSecr
1783 9.445910   172.31.132.59    172.31.132.55    ESP    374 ESP (SPI=0xc576c0f2)
1784 9.447379   172.31.132.55    172.31.132.59    ESP    134 ESP (SPI=0x827bc80e)
1785 9.447413   172.31.132.55    172.31.132.59    TCP     66 49848 > rfb [ACK] Seq=8044 Ack=734572 Win=385 Len=0 TSval=1036598 TSecr
▶ Frame 1783: 374 bytes on wire (2992 bits), 374 bytes captured (2992 bits)
▶ Ethernet II, Src: Hewlett-_ae:53:40 (00:16:35:ae:53:40), Dst: Dell_c2:a1:01 (84:8f:69:c2:a1:01)
▶ Internet Protocol Version 4, Src: 172.31.132.59 (172.31.132.59), Dst: 172.31.132.55 (172.31.132.55)
▼ Encapsulating Security Payload
    ESP SPI: 0xc576c0f2
    ESP Sequence: 17659
```

Fig 2 IPsec implemented in captured packet

We have also analysed the performance of tunnel for different VM migration and calculate the total migration time taken under light and heavily loaded system. Without IPsec implementation time taken will be less under different scenario as IPsec encapsulate each and every packet and migrate, therefore total migration time will be high. Table 1 and Table 2 shows Migration time calculation for IPsec under heavily loaded system and lightly loaded system for different scenarios.

TABLE I. Migration Time Calculation with IPsec Implementation: With heavy load

Virtual Machine	Downtime Time	Total Migration Time
Ubuntu 10.04 LTS (Lucid)	7.5 sec	65 sec
Ubuntu 12.04 LTS (Precise)	8 sec	69 sec
Debian Lenny	7.3 sec	66 sec

TABLE II. Migration Time Calculation with IPsec Implementation: With light load

Virtual Machine	Downtime Time	Total Migration Time
Ubuntu 10.04 LTS (Lucid)	6.2 sec	52 sec
Ubuntu 12.04 LTS (Precise)	6.6 sec	57 sec
Debian Lenny	6.5 sec	55 sec

After implementing IPsec tunnel between two hosts, Security cannot be breached and data remain confidential as IPsec deploy ESP (Encapsulating Security payload) and AH (Authentication Header) for each and every packet traversing.

VI. AUTOMATING LIVE MIGRATION PROCESS

As the size of data centre increases, managing the entire environment running with numerous virtual machines of autonomous users becomes cumbersome task. Also in case of sudden workload change, manual migration of VM to different host becomes hectic. Automated live migration for workload balancing seeks to improve the performance of distributed system. This ensures uniform workload for each and every host, if VM unable to satisfy user's requirement then performance degrades and Service Level Agreement (SLA) breaks.

First and foremost priority of any organisation is to make sure that their customers are happy with their strategy and trusting them, because customer is their source of income. So organisations always provide best and efficient techniques for their customers.

The idea is to identify overloaded host and underutilised host then pick a virtual machine from overloaded host and migrate it to underutilised one having enough physical resources to accommodate it.

1) For calculating load and free resource of different physical system a script is used at each and every physical system which continuously calculates load and appends it in a log file after a particular interval of time. Load is shown in figure 3.

2) The log file of each system is appended in a main log file stored at ISCSI shared storage. This log file will contain cpu-stat, free cpu-stat, ipaddress of system.

3) Details of main log file is read by central server who sort it according to free available resources to find ip-address of

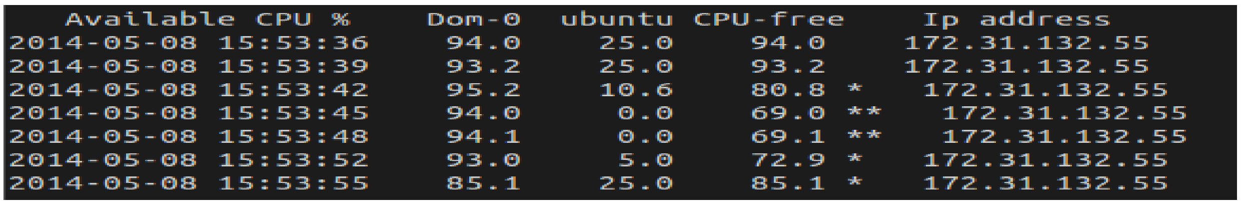
heavily loaded system and lightly loaded system after specified time intervals.

4) Central server will act as monitoring tool which monitors load of each and every system and if load goes beyond acceptable threshold value, it instructs physical system to migrate VM to lightly loaded system.

Central server also examines the shared main log file which continuously gets updated after specific interval of time with every physical machine load details. Shared access of file need to be handled properly as every process apply lock and update its CPU load detail and release lock for other process to update it.

Proposed approach should provide uniform load balancing, otherwise lightly loaded system became heavier and heavier loaded will become lighter which in turn create a loop like environment.

This migration takes place directly based on defined threshold limit of a system load. Also, if there sudden increase in the VM creation and migration then we have to add extra hardware resource to mitigate it.



```
  Available CPU %      Dom-0  ubuntu CPU-free      Ip address
2014-05-08 15:53:36     94.0    25.0     94.0     172.31.132.55
2014-05-08 15:53:39     93.2    25.0     93.2     172.31.132.55
2014-05-08 15:53:42     95.2    10.6     80.8 *    172.31.132.55
2014-05-08 15:53:45     94.0     0.0     69.0 **    172.31.132.55
2014-05-08 15:53:48     94.1     0.0     69.1 **    172.31.132.55
2014-05-08 15:53:52     93.0     5.0     72.9 *    172.31.132.55
2014-05-08 15:53:55     85.1    25.0     85.1 *    172.31.132.55
```

Fig 3 Load calculation of a system

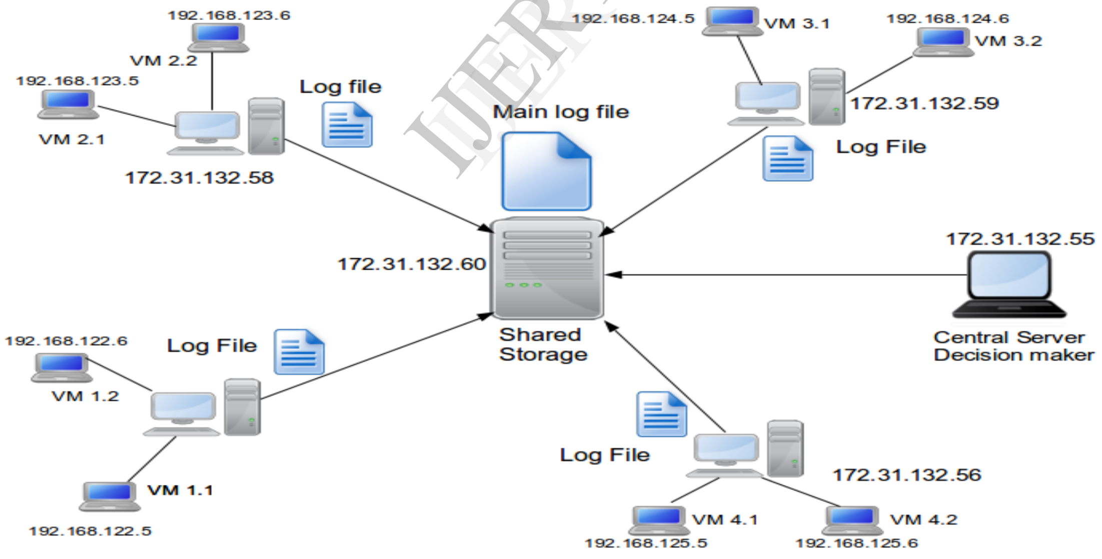


Fig 4.Automated live VM migration scenario

This migration of VM also needs to be secure from theft. For this, site to site IPsec tunnel is implemented from a host to other host which is shown above, to generate secure mesh structure within LAN. Our proposed approach provides uniform load balancing as virtual machine migrates from heavily to lightly loaded system.

VII. CONCLUSION AND FUTURE WORK

Our work shows that live VM migration is not secure as man-in-middle attack can be performed and data of running VM can be sniffed. IPsec tunnel is best suited to make it secure but the cost incurred is increased downtime and total migration time. Different analyses are done for migration time & downtime for heavy and lightly loaded systems. Creating IPsec tunnel for migration and extra overhead due is also analysed. Secure live VM migration is the need for load-balancing among servers. Manual migration becomes cumbersome with ever increasing virtual machines as we don't know which machine to migrate and where to migrate. It became a hectic task to do it manually. Automating the entire system will help data centre administration to manage maintenance task and re-balance the system properly without users noticing that their VM got migrated. For this, a mechanism is developed and implemented, which calculates resource available in all servers, append in a main log file stored in shared storage and takes decision about migrating machines based on some threshold value. It would be interesting to embed strategy in open source XEN hypervisor code and analyse the performance.

VIII. REFERENCES

1. A Trusted Virtual Machine in an Untrusted Management Environment "*IEEE transactions on services computing,"* Vol 5,No 4, oct-dec 2012
2. S. Osman, D. Subhraveti, G. Su, and J. Nieh. "The design and implementation of zap: *A system for migrating computing environments, Proc. 5th USENIX Symposium on Operating Systems Design and Implementation*," 2004
3. Vmware Vmotion, *Live Migration for Virtual Machines Without Service Interruption* Vmware Inc.
4. Konig, A. and Steinmetz, R. (2011). "*Detecting Migration of Virtual Machines. In Proceedings of the 11th Euroview*," 2011.
5. Jacob G. Hansen and Eric Jul. Self-migration of operating systems. "*In Proceeding of the 11th ACM SIGOPS Europe Workshop"* (EW 2004)
6. Jon Oberheide, "*Empirical Exploitation of Live Virtual Machine Migration,"* 2008.
7. Hai Jin, Li Deng, Song Wu, Xuanhua Shi, and Xiaodong Pan. Live virtual machine migration with adaptive, memory compression. In CLUSTER '09. *"IEEE International Conference on Cluster Computing and Workshops,"* pages 1–10, September 2009.
8. Anders Svensson. Dynamic alternation between load sharing algorithms. In System Sciences, 1992. "*Proceedings of the Twenty-Fifth Hawaii International Conference on*," volume 1, pages 193–201. IEEE, 1992.
9. Mattias Forsman & Adnreas Glad, *"Automated Live Migration of Virtual Machines,"* 2013