PERFORMANCE ANALYSIS OF TRANSPORT PROTOCOL DURING LIVE MIGRATION OF VIRTUAL MACHINES

Prakash H R
Dept of CSE, R.V.C.E, Bangalore
prakashrvcecn@gmail.com

Anala M.R
Dept of CSE, R.V.C.E Bangalore
anala_m_r@yahoo.co.in

Dr.Shobha G
Dept of CSE, R.V.C.E Bangalore
shobhatilak@rediffmail.com

Abstract—The Physical servers used in IT are under-utilized. The better utilization of these servers can be achieved using virtualization technology. Virtualization techniques create multiple partitions which are isolated with each other called virtual machines. Each virtual machine (guest) runs their own operating system. The resource allocated for these VMs may fail to execute an application because of resource conflict or un availability of resources. This motivates towards live migration of virtual machines. The live migration copies the running VM from source host to destination host seamlessly using TCP as transport protocol.This paper evaluates performance of TCP in live migration of KVM based virtual machines. The flexibility in UDP which drives the concentration can also be used for this migration.

***Keywords*— Virtualization, Virtual Machines, Live Migration, Transport Protocol, Performance Analysis.**

I INTRODUCTION

Virtualization is a technique for hiding the physical characteristics of computing resources from the way in which other systems, applications or end users interact with those resources. A virtual machine (VM) is a completely isolated operating system installation within the normal operating system. VMs are implemented by either software emulation or hardware virtualization [1]. A virtual machine (VM) is a software implementation of a machine (i.e. a computer) that executes programs like a physical machine. Virtual machines are separated into two major categories, based on their use and degree of correspondence to any real machine. A system virtual machine provides a complete system platform which supports the execution of a complete operating system (OS). In contrast, a process virtual machine is designed to run a single program, which means that it supports a single process. An essential characteristic of a virtual machine is that the software running inside is limited to the resources and abstractions provided by the virtual machine it cannot break out of its virtual world.

A virtual machine was originally defined by Popek and Goldberg as "an efficient, isolated duplicate of a real machine". The main advantages of VMs are

- Multiple OS environments can co-exist on the same computer, in strong isolation from each other.
- The virtual machine can provide an instruction set architecture (ISA) that is somewhat different from that of the real machine.
- Application provisioning, maintenance, high availability and disaster recovery.

The main disadvantages of VMs are:

- A virtual machine is less efficient than a real machine when it accesses the hardware indirectly.

- When multiple VMs are concurrently running on the same physical host, each VM may exhibit a varying and unstable performance (Speed of Execution, and not results), which highly depends on the workload imposed on the system by other VMs, unless proper techniques are used for temporal isolation among virtual machines.

Increase in number of virtual machine in a cluster makes it difficult to manage the resources allocated to VMs. This lead to underperforming of virtual machine or the VM may collapse and fail to continue to serve. The process of moving a virtualized guest from one host to another is called as live migration. The objective is to move a VM [1] running in one physical machine (host) to another physical machine for continued services using TCP as transport protocol and to evaluate the performance of TCP during live migration. In live migration the VM monitor moves a running virtual machine instance nearly instantaneously from one server to another. Traditionally TCP has been used as transport protocol for the live migration since it is reliable but suffers because of overhead due to connection setup and connection release. The paper discusses on the performance of transport protocol during live migration of virtual machines using KVM hypervisor.

The paper is organized as follows section II describes the literature survey on the live migration techniques and live migration in different hypervisors. Need for live migration and advantages of live migration are discussed in section III. Section IV gives the design and implementation of the system and results are discussed in section V and finally the conclusion in section VI.

II LITERATURE SURVEY

The purpose of this literature survey is to provide background information on the issues to be considered in this paper and to emphasize the relevance of the present study [2].

Virtualization is a technique of disassociating the tight bond between software and hardware. It hides the physical characteristics of a computing platform from users, instead showing another abstract computing platform. There are many instances where a virtualization is used like running applications not supported by host OS, server virtualization, evaluating alternate OS and others. There are many types of virtualization viz. Para virtualization, Operating System virtualization and Application level virtualization. Para virtualization does not stimulate hardware and uses special API the guest must use, mostly used in XEN, Virtual box etc. Operating System level virtualization [1] allows multiple virtual servers to be run and application sees isolated OS. Application level virtualization is where an own copy of the components that are not shared are given.

The advent of innovative technologies, such as multi-core, Para-virtualization, hardware assisted virtualization and live migration have contributed to an increasing adoption of virtualization on server systems. At the same time, being able to quantify the pros and cons of adopting virtualization in face of such advancements is a challenging task. The impact of virtualization in a variety of scenarios has been the focus of considerable attention. A number of studies have presented individual and side by side measurements of VM runtime overhead imposed by hypervisors on a variety of workloads by live migration.

The studies presented by Clark et al [2] specifically deals with VM migration. This paper analyzes performance degradation when migrating CPU and memory intensive workloads as well as migrating multiple VMs at the same time, however such study employs a pure stop-and-copy migration approach rather than live migration. The paper introduces XEN live migration and quantifies its effects on a set of four applications common to hosting environments, primarily focusing on quantifying downtime and total migration time and demonstrating the viability of live migration. However, these works have not evaluated the effect of migration in the performance of modern internet workloads, such as multi-tier and social network oriented applications.

Seamless Live Migration of Virtual Machines over MAN/WAN [3] is a paper on live migration over MAN and WAN. The results provides a new stage virtualization one for which computation is no longer localized within a data centre but rather can be migrated across geographical distances with just 1 to 2 seconds of application down time.

A survey of live migration in virtual network environment [5] presents challenges and techniques of live migration in virtual network environment. When overload occurs in one of the virtual machine live migration is needed. This paper has taken into account the resource availability in the host VM and tends to load balance nodes and connections.Zap uses partial OS virtualization [6] to allow the migration of process domains (pods), essentially process groups, using a modified Linux kernel. Their approach is to isolate all process-to-kernel interfaces, such as file handles and sockets, into a contained namespace that can be migrated. Their approach is considerably, largely due to the smaller units of migration. However, migration in their system is still on the order of seconds at best, and does not allow live migration, pods are entirely suspended, copied, and then resumed. Furthermore, they do not address the problem of maintaining open connections for existing services.

The studies presented by Sherif Akoush and group [7] deals with predicting the performance of virtual machine migration. In this paper, they characterize the parameters affecting live migration with particularly emphasizing on the Xen virtualization platform. They discussed the relationships between the important parameters that affect migration and highlight how migration performance can vary considerably depending on workload. Elis Kullberg analyses the live migration of virtual machines with sustaining TCP connections [8], used TUN/TAP drivers for the migration and the throughput is analyzed and documented using iperf tool. This paper analyses the migration using TCP, the throughput of the guests after the migration is analyzed.

The Efficacy of Live Virtual Machine Migrations over the Internet [9] describes progress on developing a system that utilizes Mobile IPv6 to enable constant network connectivity through the migration. Also the sources of delay associated with the live migration are identified and concludes that as long as migrations occur relatively infrequently, live migration over the Internet is practical.

Live and Incremental Whole-System Migration of Virtual Machines Using Block-Bitmap [10], their approach is to minimize the downtime caused by migrating large disk storage data and keep data integrity and consistency, a three-phase migration (TPM) algorithm is being proposed by the authors. To facilitate the migration back to initial source machine, an incremental migration (IM) algorithm is used to reduce the amount of the data to be migrated. Block-bitmap is used to track all the write accesses to the local disk storage during the migration. Synchronization of the local disk storage in the migration is performed according to the block-bitmap**.**

A Study on Performance of Processes in Migrating Virtual Machines [11], evaluates the performance of migrating virtual machines. The experimental results revealed that I/O performance of a process on a virtual machine severely decreases during migration.

III Live Migration

The live migration of virtual machines(fig 1) with sustained TCP-connectivity is a problem. It is difficult to maintain the sustained TCP connectivity during live migration. However, for live migration to get a higher level of acceptance and utilization, more research and development needs to be conducted. The performance is analyzed by using network packet analyzer for capturing the packets during live migration of VMs.

Advantages of Live Migration

- Load balancing - guests can be moved to hosts with lower usage when a host becomes overloaded. A guest becomes so utilized that it is either migrated to a new machine, or other guests on that machine are migrated off to give the busy guest more resources.
- Hardware failover - when hardware devices on the host start to fail, guests can be safely relocated so the host can be powered down and repaired. A system begins warning of soft errors in memory, or over temperature alerts, or other indications of an imminent failure. Guests may be migrated off before they are shut down, and the server can be freed up for maintenance.

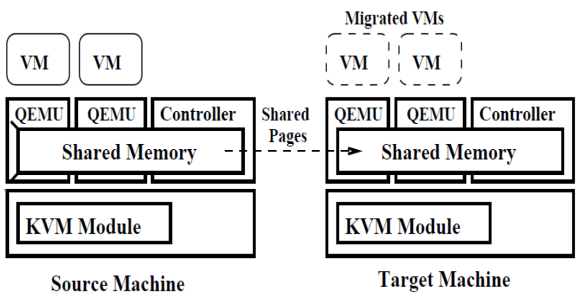

Fig 1 Design of live migration

- Energy saving - guests can be redistributed to other hosts and host systems powered off to save energy and cut costs in low usage periods.
- Geographic migration [3] - guests can be moved to another location for lower latency or in serious circumstances.

Migrating an entire OS and all its applications as one unit allows us to avoid many of the difficulties faced by process-level migration approaches.

Current State Challenges

This addresses the challenges faced by the engineers during the process of Live Migration. The drawbacks of current scenario of Live Migration process are:

- The transport protocol used for live migration of guest OS memory or the image file is TCP. TCP is reliable but it adds too much complexity and overhead this gives the idea for implementing the above using other protocols.
- TCP as a transport protocol has the problem of keeping active TCP-sessions connected for an indefinite time after migration. Most solutions today offer no integrated way to ensure that network connectivity is unaffected by a server migration, thus all existing connections are lost for the end user every time a server is migrated.
- Other transport protocols such as UDP, SCTP etc may attract user in reduced downtime and reduced cost of connections during the initial set up of the migration.

The Solution

The Live Migration using TCP can be improved and further development can be done if it analyzed properly

- The implementation with other connection less protocol may enhance the decrease in the downtime and decrease in the total migration time.
- The analysis work also helps in tuning the live migration according to the requirement.

IV DESIGN

The design issues of live migration using KVM is presented in this section. This experiment set up uses ubuntu10.10 as host and guest OS. The migration of virtual machines is executed successfully. The steps involved in live migration are briefly classified as preparation and migration.

Preparation

In this phase a request is issued to migrate an OS from source physical machine [2] A to target physical machine B. It is required to confirm that the necessary resources are available on B and reserve a VM container of that size. Failure to secure resources here means that the VM simply continues to run on host A unaffected. During the first iteration, all pages are transferred from A to B as shown in figure1. Subsequent iterations copy only those pages dirtied during the previous transfer phase. We suspend the running OS instance at A and redirect its network traffic to B. At the end of this stage there is a consistent suspended copy of the VM at both A and B. The copy at A is still considered to be primary and is resumed in case of failure.

Migration

In this phase host B indicates to A that it has successfully received a consistent OS image [2]. Host A acknowledges this message as commitment of the migration transaction, host A may now discard the original VM, and host B becomes the primary host.

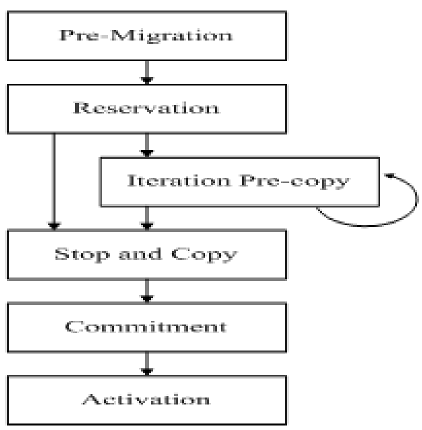

Fig 2 Live Migration Time line

The logical steps that are followed during the preparation and migration [2] are summarized in figure 2. Preparation includes pre-migration process where the destination machine is examined to confirm that the necessary resources are available to run the VM. This is followed by the reservation stage where resources for the new incoming machine are reserved for running the migrated machine in the future. Pre-copy of the VM is done after finishing all the iterations. Then the source VM is stopped and dirty pages are copied to the destination physical machine. Migration includes commitment and activation of the VM in the destination physical machine only if the

VM is successfully running and it is synchronized with the VM running in the source physical machine.

V RESULTS AND DISCUSSION

This section demonstrates the performance analysis of TCP as a transport protocol during live migration of VMs. Wireshark tool is a network packet analyzer and it is used to capture the packets during live migration. The captured packets are stored in the Wireshark tool and can be used for the analysis. For transport protocol performance analysis, the benchmarking tool Netperf can be used to measure various aspects of networking performance. The first parameter considered to evaluate the performance is the total migration time [7]. The total migration time is the time taken to migrate the VM from one physical machine to other physical machine. It is the time between the first packet captured and the last packet captured by the Wireshark tool. A network packet analyzer will try to capture network packets and displays the packet data as detailed as possible. For analyzing the performance of TCP as transport protocol following parameters are considered

- Measuring the Stream performance
- Measuring the request response time
- Total migration time
- Overhead

Measuring Stream Performance of TCP

TCP stream performance during the live migration is conducted for both shared disk storage and non shared storage. For this Netperf tool is used to measure the TCP performance. TCP stream performance is the first area that can be investigated with Netperf. The obtained performance using Xen hypervisor [11] show that the communication by a process in a host OS has higher priority than that by a process in virtual machine. They measured throughput obtained by a Netperf process in a live migrating virtual machine and the measured throughput values show that the highest of 91.3MB/s and lowest of 54.8MB/s. It was observed that the performance of shared disk is 1.125 times more than non shared disk. The results obtained in our experiment are as shown in the table 1 and table 2.

	Receiver socket size bytes on the destination machine	Sender socket size bytes on the source machine	Throughput MB/s (Performance)
1	87380	16384	107.07
2	87380	16384	107.51
3	87380	16384	108.19

Table 1 TCP Stream Performance for shared disk

	Receiver socket size bytes on the destination machine	Sender socket size bytes on the source machine	Throughput MB/s (Performance)
1	87380	16384	95.12
2	87380	16384	95.18
3	87380	16384	95.14

Table 2 TCP Stream Performance for non shared disk

Measuring TCP Request Response Performance

	Send socket size		Receive socket size		Transactions per second
	Local	Remote	Local	Remote	
1	16384/16384		87380/87380		6866.85
2	16384/16384		87380/87380		6550.70
3	16384/16384		87380/87380		6551.06

Table 3 TCP R-R Performance for shared disk

	Send socket size		Receive socket size		Transactions per second
	Local	Remote	Local	Remote	
1	16384/16384		87380/87380		5917.17
2	16384/16384		87380/87380		59245.16
3	16384/16384		87380/87380		5918.42

Table 4 TCP R-R Performance for non shared disk

A TCP request-response test measures the number of request-response transactions per second. The initiating system sends a request packet of a specified size and waits for the other system to return a response packet of a specified size. As soon as the response is received, another request is sent.

TCP request-response performance is measured by building a single TCP connection and sending requests and responses over that connection for the lifetime of the test. Request response performance is the second area that can be investigated with Netperf. Generally Netperf request/response performance is quoted as "transactions/s" for a given request and response size. A transaction is defined as the exchange of a single request and a single response. From a transaction rate, one can infer one way and round-trip average latency. Processing speed means that the transactions per second decreases during migration. In addition, the impact increases during the migration. The results obtained our experiment are as shown in the table 3 and table 4.

Total Migration Time

Total migration [2] [7] time may be defined as the sum of the time spent on all migration stages from initialization at the source host through to activation at the destination. Total migration time [7] is given by the equation (1)- the time taken to migrate virtual machine from one physical machine to another physical machine. Here the total migration time is the time between the first packet and the last packet captured by the Wireshark tool during the live migration.

$$Migration\ time = Initialize + Reserve + \sum pre\ copy + stop\ and\ copy + commit + Activate$$
(Equation 1)

For Xen hypervisor [7] the total migration time falls between the 13.3 seconds to 49.9 seconds for the VM size of 1024 MB on 1Gbps link. Total downtime starts to increase in proportion to the increase in the number of modified pages that need to be transferred in the stop-and copy stage. This is attributable to the fact that more modified pages have to be sent in each pre-copy round. Moreover, the migration sub-system has to go through more iteration with the hope to have a short final stop-and-copy round. In this experiment the total migration time varies for each run of live migration module, for shared disk the maximum time was 135 seconds and the minimum time was 78 seconds keeping the VM memory constant. For non shared disk the maximum time was 198 seconds and the minimum time was 141 seconds keeping the VM memory constant. Total migration time starts with the TCP and HTTP message between the source and destination physical machine.

Total migration time in shared storage and non shared storage is measured when virtual machines have no running process on them. The experimental result is shown in Figure 3. Memory size of a virtual machine and migration time are proportional. On average, total migration time increases linearly with VM size.

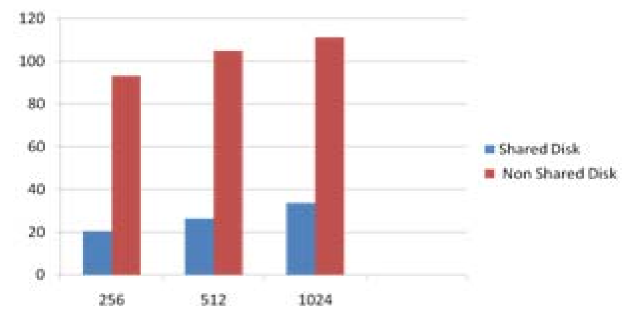

Fig 3 Total Migration Time

Figure 4 explains IO graph for shared storage migration. The packets/tick is on Y axis and time on X axis. It shows that the area marked in red line is the actual copy process of the migration. The pre-copy starts after the 20-23 seconds later. A copy of the VM is available at the destination machine, and starts running at the destination machine. When both source VM and destination VM are synchronized then destination VM sends a commitment, the VM at the source stops and network traffic routed to the destination VM. A small continuity in the throughput shows that process is still communicating with the source machine. Since the migration is done with shared disk memory and that communication is indicated in the graph.

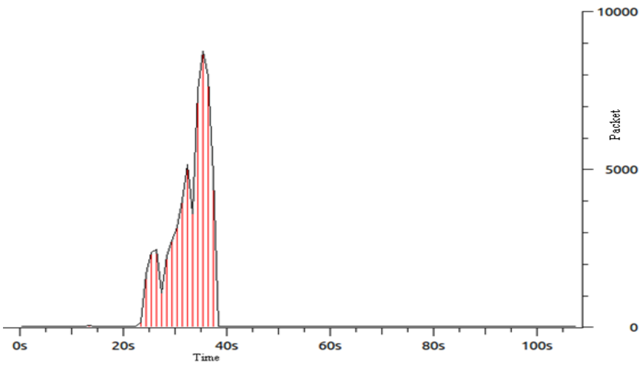

Fig 4 Throughput for shared storage

Figure5 shows IO graph for non shared disk migration. It shows that the area marked in red line is the actual copy process of the migration and it exists for extended time till copying the complete disk memory. In this pre-copy starts after the 20-23 seconds and the copies the VM memory as well as the physical storage of the VM. A copy of the VM is available at the destination machine and starts running at the destination machine. When both source VM and destination VM are synchronized the destination VM sends a commitment message to the source physical machine and VM at source stops and network traffic routed to the destination VM. All the communication is over after the VM successfully runs on the destination VM.

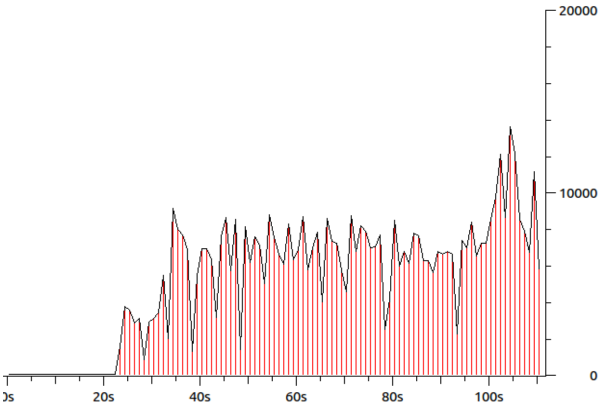

Fig 5 Throughput for non shared storage

Overhead

During the TCP conversations between the source and destination machine during the live migration process, totally 21.38 GB are processed. Huge amount of data is exchanged because of complexity introduced in TCP protocol during the live migration. The overheads are

- Connection set up , acknowledgements and connection termination
- Malformed packets during migration
- Out of order segments
- Retransmission of lost data and acknowledgements

V CONCLUSION

Virtual machines are building blocks of modern data centers and clusters. These virtual machines may fail due to resource conflicts and hardware failure. For effective utilization of resources and higher availability of hardware VMs are migrated between the physical machines on failover. Hence live migration is acceptable solution for this problem. Analysis of performance of transport protocol used for migration is the key issue to improve the effect of live migration process. The outcome of the work carried out presents a cost effective way of analyzing the performance of transport protocol in the process of live migration of virtual machines using KVM as hypervisor. In this VMs are created and virtual machines are live migrated to other physical machine using TCP as the transport protocol. The performance of TCP is analyzed and statistics says total migration time is directly proportional to the VM size. TCP stream performance of shared disk is 1.125 times more than non shared disk. TCP transactions per second are more in case of shared disk when compared to non shared disk. It also shows that during migration, huge amount of data has been exchanged between source and destination physical machine due to complexity and overhead in the TCP protocol.

VI REFERENCES

[1] Smith, J.E.; Ravi Nair; Dept. of Electr. & Comput. Eng., Wisconsin Univ., Madison, WI, USA Smith, James E. "The Architecture of Virtual Machines". Computer (IEEE Computer Society), ISSN: 0018-9162, Pages 32-38, Digital Object Identifier:10.1109/MC.2005.173.

[2] Christopher Clark, Keir Fraser, Steven Hand, Jakob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, Andrew Warfield, JJ Thomson, “Live Migration of Virtual Machines”, NSDI'05 Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2 ,USENIX Association Berkeley, CA, USA 2005.

[3] Travostino Franco, Daspit Paul, Gammans Leon, Joj Chetan, de Laat Cees, Mambretti Joe,Monga Inder, van Oudenaarde Bas, Raghunath Satish, Wang Phil, “Seamless Live Migration of Virtual machines over the MAN/WAN”, journal Future Generation Computer Systems - IGrid 2005: The global lambda integrated facility USA. Digital Object Identifier: 10.1016/j.future.2006.03.007 Volume 22 Issue 8, October 2006.

[4] Kivity Avi, Kamay Yaniv, Laor Dor, Lublin Uri, Ligouri Anthony, “KVM: the Linux Virtual Machine Monitor”, proceedings of the Linux Symposium Volume one, Ottawa Canada,2007, pages 225-230.

[5] Jebalia, Maha; Ben Letaifa, Asma; Tabbane, Sami; Higher School of Communication at Tunis (SUP'COM), TUNISIA. “A Survey of Live Migration in Virtual Network Environment”, 10th Annual International Conference on: New Technologies of Distributed Systems (NOTERE), 2010, Location: Tozeur, Tunisia Print ISBN: 978-1-4244-7067-9, Pages 351-354, Digital Object Identifier: 10.1109/NOTERE.2010.5536543.

[6] 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 (OSDI-02), New York, USA December 2002 ISBN: 978-1-4503-0111-4, pages 361-376, Digital Object Identifier 10.1145/1060289.1060323.

[7] Akoush, S.; Sohan, R.; Rice, A.; Moore, A.W.; Hopper, A.Computer. Lab., University of Cambridge, Cambridge, UK “Predicting the performance of virtual machine migration”, IEEE International Symposium on: Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2010, pages: 37 – 46, Digital Object Identifier: 10.1109/MASCOTS.2010.13.
[8] Elis Kullberg “Live migration of Virtual Machines: Sustaining active TCP-sessions” KTH - Royal Institute of Technology Stockholm, September 3 2009 [Available under Creative Commons ShareAlike 3.0 License -http://creativecommons.org/licenses/by-sa/3.0/]
[9] Harney, E.; Goasguen, S.; Martin, J.; Murphy, M.; Westall, M.; School. Of Computer, Clemson University., Clemson, SC, USA. “The Efficacy of Live Virtual Machine Migrations over the Internet”, Second International Workshop on Virtualization Technology in Distributed Computing (VTDC) 2007, Print ISBN: 978-0-7695-2915-8, pages 1-7, Digital Object Identifier: 10.1145/1408654.1408662.
[10] Yingwei Luo Binbin Zhang Xiaolin Wang Zhenlin Wang Yifeng Sun Haogang Chen Dept. of Computer Science & Technology, Peking Univ., Beijing. ‘Live and Incremental Whole-System Migration of Virtual Machines Using Block-Bitmap”, IEEE International Conference on Cluster Computing, 2008, ISBN: 978-1-4244-2639-3, pages 99-106.