

Migration Jobs in Cloud Computing

Anita Rani and Pankajdeep Kaur

CSE Department, GNDU, RC, Jalandhar
anitasaroay@yahoo.in, pankajdeepkaur@gmail.com

Abstract

Cloud computing is the delivery of computing services over the internet. Cloud services allow individuals and other businesses organization to use data that are managed by third parties or another person at remote locations. In cloud computing, Process and virtual machine migration are two migration techniques. Process migration is a technique in which an active process is moved from one machine to another of possibly different architecture and Virtual Machine (VM) is an emulation of a particular computer system. Virtual machine migration is a useful tool for migrating OS instances across multiple physical machines. Both techniques are used to load balancing, fault management, low-level system maintenance, better communication and reduce energy consumption. This paper presents the various virtual machine and process migration techniques.

Keywords: Process Migration, VM Migration, Live migration, Pre-copy Post-copy

1. Introduction

Process Migration is a technique in which an active process is moved from one machine to another of possibly different architecture. Therefore it is necessary to capture the process's current state of execution and recovering it on the destination machine in a manner understandable to it. Process migration is used to load sharing among the pool of processors and also improve the communication performance, reduced the cost by bringing the processes closer together. This paper describes process capture and recovery of the internal state of process.

Further this paper highlights major migration issues include:

- 1) The mechanism of process state capture and recovery.
- 2) The initiation of the process state capture.

Process state capture in distributed computing systems cannot simplify the process state capture, when the request of capture is received. The process state can be capture and initiated only at certain points –

- 1) When different instances of computation of different architecture's points are equal – so that the process can be again started from the same point where it is halt.
- 2) When the request for capture is received, then the state capture is initiated at the next point. At the same time, the normal execution of process should be ensured.

The growth of the amount of data and the number of user requests is increasing day by day, which require more the computing and processing of the servers. It distributes the computing tasks to the resource pool made from a large number of computers.

VMs refer to one instance of an operating system along with one or more applications running in an isolated partition within the computer. There will be multiple virtual machines running on top of a single physical machine. When one physical host gets overloaded, it may be required to dynamically transfer certain amount of its load to another machine with minimal interruption to the users. This process of moving a virtual machine from one physical host to another is termed as migration. In the past, to move a VM between two physical hosts, it was necessary to shut down the VM, allocate the

needed resources to the new physical host, move the VM files and start the VM in the new host. Live migration makes possible for VMs to be migrated without considerable downtime. The transfer of a VM actually refers to the transfer of its state. This includes its memory, internal state of the devices and that of the virtual CPU. Among these, the most Time-consuming one is the memory transfer. Two parameters are considered while performing the live VM-migration are-

- 1) **Downtime-** Down time refers to the time during which the service of the VM is not available.
- 2) **Migration time-** Migration time refers to the total amount of time required to transfer a virtual machine at source to destination node without affecting its availability

2. Related Work

*a) Process Migration-*For homogeneous computing systems, Process state capture and recovery mechanisms are well developed and can now typically be performed with minimal overhead and latency; less progress has been done in this functionality across heterogeneous architectures.

The Tui System[2] constructed to provide a heterogeneous migration tool that use on four common architectures in the UNIX environment. In this system, the capture and recovery of the state of a process is done in a highly machine dependent manner, that requires complete knowledge of the fixed conventions. Also, when a process state capture is requested, a breakpoint instruction to trap to the capture mechanism is placed at all “pre-emption points” *i.e.* the enabled points of equivalence. To avoid this loss of correctness, it is necessary to fill space at each pre-emption point and it should be enough to adjust the breakpoint instruction.

The process introspection model proposed by Ferrari the portable[4] (or shadow-) check pointing model presented in perform the activation history state’s capture in an machine-independent manner by using the call-return functions provided in the high-level programming language and it can also done by the intermediate instruction set. Here, the architectural differences are occurred it will be handling by the compiler. However, they used a polling approach to process state capture initiation. Poll points are created there, where the process determines if a capture can be initiated or not.

A new poll free solution is given by P. Bungale[3] in which they propose a poll-free solution that involves preserve semantics self-modification of code. However, unlike the other poll-free solutions, when a capture is requested it do not place capture initiation instructions at all enabled points of equivalence. Instead, it places the initiation instructions in a copy of only a small portion of the code. Thus, it removes the introduction of placeholder dummy instructions and thus it increases the performance overhead. One of the major advantages of this design is the modification of program to incorporate state capture and recovery function.

D.S. Miloji [6], proposed a Migration Algorithm to migrate the process from one machine to another without state capture but it uses forwarding references.

1. **A migration request is send to a remote node** after negotiation, migration request has been accepted.
2. **A process is apart from its main node** by suspending its current execution and temporarily redirecting communication as described above.
3. **Communication is temporarily redirected** .This step continues with steps 4, 5, and 6, as long as messages received.
4. **The process state is extracted**, by memory contents processor state, communication state. The communication state is system dependent. Local system’s internal state is not transferable.
5. **A destination process instance is created** in which the new state will be imported.

6. **State is transferred to a new instance**, all the states are not transferred but some of the state can be brought even after migration completion.
7. **Means of forward references** to the migrated process should be maintained. It is also necessary to communicate with the process and to control it.
8. **New instance is resumed only** when sufficient state has been transferred. After this step, process migration completes. When all of process state is transferred from the original node, it can be deleted on the source node.

A advance poll free solution is proposed by D. Ajmire [5], in this by using the various assumptions are made and algorithms are designed by using which process state initiation, capturing and recovery is done. Assumptions that are used in this design-

1. Compiler helps in obtaining the equivalence point table.
2. Operating system supports in generates the current value of program counter for a process's state initiation.
3. Machine dependent optimizations across enabled points of equivalence shall not be allowed because they prevent the different versions of the program's compiled across heterogeneous platforms.

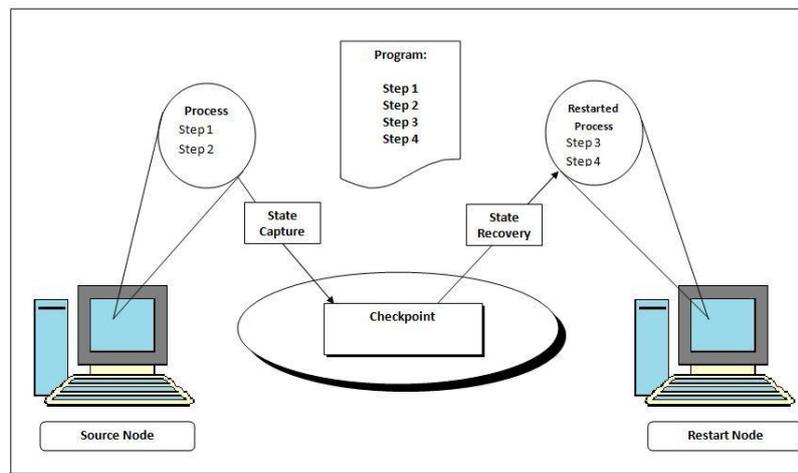


Figure 1. Basic Operation of Process Migration Algorithm [1]

a) State Capture Initiation Algorithm-When a request for a process's state capture is received from an external source, the following steps are taken:

1. The current value of the process's program counter is obtained using operating system helps.
2. The program counter value is used to identify the current executing function and the current segment of code between two points of equivalence containing the current instruction.
3. The following steps are carried out for state capture initiation on next point of equivalence.

b) State Capture Algorithm-Once the control transferred to the state capture function, the following tasks are performed initially:

1. Save the initiated point of equivalence and the interrupted function of capture.
2. The new address of the current activation record is replaced by the saving back copy of the interrupted function.

c) State Recovery Algorithm-Once a new process has been created on the destination machine, the following steps are taken into consideration to perform the process state recovery:

1. A jump instruction with destination as the corresponding restoring back copy is placed at the entry point of each function
2. When first activation record is created, the epilogue will restore the static data from the checkpoint file.
3. The restoring epilogue performs the tasks.
4. Step 3 is repeated until all activations history have not been restored.
5. The process finally resumes normal execution.

By N. Tzirita [7], a new approach is proposed in which resource consumption can be minimizing by using process migration.

Minimizing resource consumption using process migration- pay-per-use model adopted in clouds, according to this, how much resources an application running is used in cloud computing environment, the greater the amount of money will be charged to the owner of application. Therefore, intelligent solutions must be used to minimize the resource consumption. The problem which is common, to identifying an assignment scheme between the interacting components and the computing nodes of a cloud environment, to minimize total amount of resources consumed by the application. Various centralized solutions are found unsuitable for large-scale applications. Therefore, a distributed algorithm is important to assignment resources for minimum resource consumption. For minimizing resource consumption following methods are used to solve:

- (a) Centralized Solutions.
- (b) Distributed Solutions.
- (c) Migration Mechanisms.

a) **Centralized Solutions** - It includes:

- i) **Task allocation-** Older works focused on solving the task assignment problem over a dual processor system such as to minimize the total communication cost.
- ii) **VM placement-** There is also a significant number of works in the literature related to energy consumption and VM consolidation. However, most of them do not take into account the network overhead when deciding about the assignment of VMs onto nodes (servers).

b) **Distributed Solutions**-Over the last few years, large-scale distributed systems (*e.g.*, clouds, grids, sensor networks, *etc.*) have gained a lot of attention. Centralized solutions were rendered useless due to scalability issues, and were finally superseded by distributed ones.

c) **Migration Mechanisms**-In the following text, we mention the most relevant works we found in the literature in regards to the migration mechanisms that they are supported at different levels:

- i) Thread level.
- ii) Process level.
- iii) Process level.

Advantages of Process Migration Are:

1. **Efficiency**-To providing efficient state capture and recovery, the run-time performance overhead introduced by the mechanism should be at acceptable levels. If checkpoints are not performed in execution, then it will not run significantly and slower without this service.
2. **Generality**- The mechanism is used for large number of architectures and a variety of programs, *i.e.* written in a multiple languages.
3. **Minimal Latency**-This mechanism is suitable even for ensuring minimum possible latency that is the time delay between when a capture initiation is requested and

when the capture is actually initiated. This is the time taken to reach at the next possible point of equivalence in the computation.

4. **Ease of Use**- This mechanism is fully automatic, requiring little effort from the application programmer. Such full automation is only possible for programs that expressed in a platform-independent manner. [6]

b) VM Migration-Virtualization allows partitioning one physical machine into multiple virtual machines that runs concurrently and also shares the same physical resources. Virtual Machines (VM) can be provisioned on-demand, replicated and migrated. Virtual machine migration done from one physical machine to another machine for load balancing and physical machine fault tolerant. It can also be used to reduce power consumption of cloud data centers.

Virtual Machine Migration methods are divided into two types:

- 1) Hot (live) migration- Virtual machine keeps running while migrating and does not lose its status.
- 2) Cold (non-live) migration- The status of the VM loses and user can notice the service interruption.

User doesn't feel any interruption in service in hot (live) migration. In live migration process, the state of a virtual machine to migrate is transferred. The state consists of its memory contents and local file system. Local file system need not be transferred. In cold migration, first, VM is suspended, then its state is transferred, at last VM is resumed at destination host.

Live VM Migration- Live migration [8] is a technology used for load balancing and optimization of VM deployment in data centers. With the help of live migration, VMs can be transferred to another node without shutting down. Live migration is classified into two steps:

- (i) Control is switched to the destination.
- (ii) Data Transferring (memory/disk) to the destination.

Pre-copy- In this, first Memory is transferred and after this execution is transferred. The pre-copy method is used to transfer the memory to the destination node over a number of iterations.

Post-copy- In this, First execution is transferred and after this, memory is transferred. Unlike pre-copy, in post copy the Virtual CPU and devices on the destination node is transfer in the first step and starts the execution in second step. Following metrics are used to measure the performance of live migration.

- i) **Preparation**- In this, resources are reserved on the destination which performed various operations.
- ii) **Downtime**- Time during which the VM on the source host is suspended.
- iii) **Resume**- It done the instantiation of VM on the destination but with the same state as suspended source.
- iv) **Total time**- The total time taken in completion of all these phases is called Total Migration time.

Live VM Migration Techniques- The pre copy [7] migration iteratively copies the memory pages from the source machine to the destination host, without ever stopping the execution of the VM. The iterative nature of the algorithm is due to the dirty pages, *i.e.* memory pages that have been modified in the source host since last page transfer must be sent again to the destination host. But the advantage of this approach is that all updating are available at the destination host. It can be activated any time.

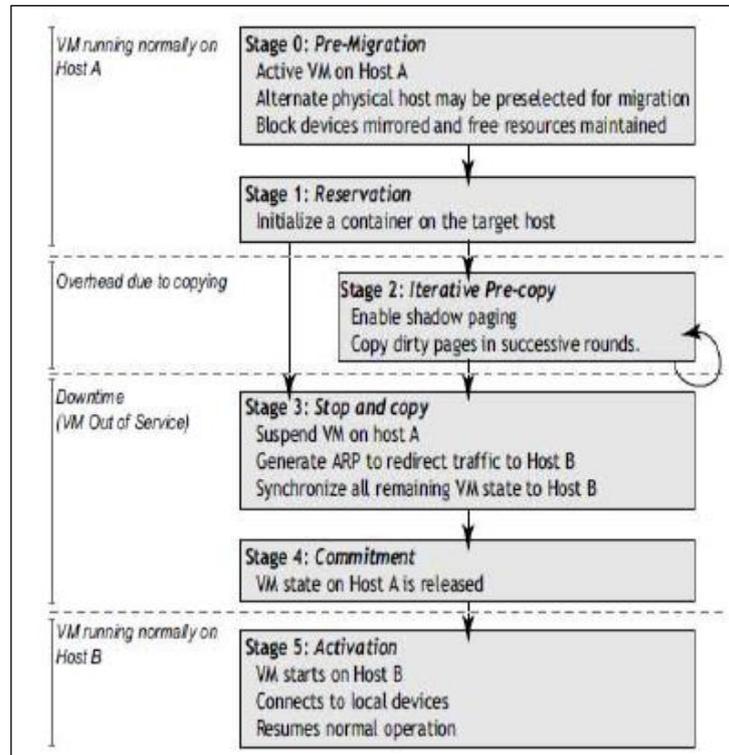


Figure 2. Pre-Copy Algorithm [7]

There are 3 phases in live migration approach:

Warm-up phase-In this, copies all the memory pages from source to destination while the VM is still running on the source. If some memory pages change during memory copy process dirty pages, they will be re-copied until the rate of recopy pages is not less than dirty page rate.

Stop and Copy phase- In this, the VM will be stopped in source and the remaining dirty pages will be copied to the destination and VM will be resumed in destination.

Pre-Copy Phase- At this stage, the VM continuously run, and its memory is copied page wise from the source to the target host. Iteratively means, the algorithm works in several rounds. It starts with transferring all active memory pages.

In modified pre-copy approach [9], a framework is constructed that includes pre-processing phase in traditional pre-copy based live migration so that the amount of transferred data is reduced. For preprocessing, they proposed the working set prediction algorithm. Applying the proposed algorithm which combined LRU (Least Recently Used) cache with splay tree algorithm, the system can reduce the amount transferred memory page. In an approach to optimize live virtual machine migration based on pre-copy algorithm, memory compression is used.

The basic idea of live migration algorithm was first proposed in [10]. First Hypervisor marks all pages as dirty, then algorithm iteratively transfer dirty pages across the network until the number of pages remaining to be transferred is below a certain threshold or a maximum number of iterations is reached. Then Hypervisor mark transferred pages as clean, Since VM operates during live migration, so already transferred memory pages may be dirtied during iteration and must need to be re-transferred. The VM is suspended at some point on the source for stopping further memory writes and transfer remaining pages. After transferring all the memory contents, the VM resumes at destination server.

In [11] describes the design and implementation of a system that uses virtual machine technology to provide fast, transparent application migration; neither the applications nor the operating systems need to be modified. Performance is measured with hundred virtual

machines, migrating concurrently with standard industry benchmarks. It shows that for a variety of workloads, application downtime due to migration is less than a second.

A high performance VM migration design based on Remote Direct Memory Access was proposed in [12]. InfiniBand is an emerging interconnects providing feature such as Operating System bypass and RDMA. Remote Direct Memory Access is a direct memory access from the memory of one computer to that of another computer without involving their operating system. With the help of Remote Direct Memory Access remote memory can be read and write directly.

In [13] introduced a whole-system live migration technique, which transfers the whole system run time state, including CPU state, local disk storage and memory data of the virtual machine. They proposed a three phase migration algorithm as well as an incremental migration algorithm, which migrate the virtual machine back to the source machine in a very short total migration time.

In [14] presented a system for supporting the transparent, live wide-area migration of virtual machines which used local storage for their persistent state. This approach is transparent to the migrated VM, and does not interrupt open network connections to and from the VM during wide area migration, This ensure consistency of the VM's local persistent state at the source and the destination after migration and is able to handle highly write intensive workloads.

[15] Presents a technique to improve the performance of live migration. In this an optimized iterative pre-copy algorithm is used which reduce the dirty rate of VM.

In [16] presents, hybrid approach is used to takes the best of methods of live migration - pre and post-copy. In this, with processor state, they add a lot of useful information of state. Which includes devices and frequently accessed pages of the VM, *i.e.* working set, used to reduce the number of page faults over network while they actively transfer memory.

In [17] Network aware VM Migration strategy is proposed. In this, migration of VM is purely base on network traffic and network latency and various strategies are used for migration.

In [18] present an approach, VM placement and migration for data intensive application, which helps to minimize data transfer time. The proposed approach places the VMs on target physical machines and consider of the network conditions between the physical machines and the data storage.

b) Advantages of VM Migration-

Load balancing: This reduces the inequality of resource usage levels across all the PMs in the cluster. This prevents some machines from getting overloaded in the presence of lightly loaded machines with sufficient spare capacity. Live migration can be used to balance the system. The overall system load can be balanced by migrating VMs from overloaded PMs to under-loaded PMs.

Server Consolidation: In order to reduce server sprawl in data centers, server consolidation algorithms are required. These algorithms are VM packing heuristics which try to pack as many VMs as possible on a PM so that resource usage is improved and unused or under-utilized machines can be turned off. Consolidation will result in reduced power consumption and thus reducing overall operational costs for data centre administrators.

Hotspot & Coldspot Migration: The detection of hotspots and coldspots are always based on thresholds which are set by the data center owner or based on the Service Level Agreements specified by the clients. Usually, a higher resource usage value close to maximum is set as the upper threshold and a very low resource usage value is set as the lower threshold. PMs having resource usage values beyond the upper threshold are said to have formed hotspots, and whose usage values below the lower threshold are said to have

formed coldspots. The former implies over-utilization and the latter implies under-utilization, applicable across any resource dimension.

Process v/s VM Migration: The problem of process migration was thoroughly implemented. The main disadvantage of process migration is, in process migration applications are strongly connected with the system in the form of sockets, *i.e.* descriptors and resource pointers which makes process migration very difficult. But in some cases *e.g.* shared memory between processes, process migration is not possible unless processes are divided by a priority. Migrate an entire node with applications is a much more manageable procedure, especially in the presence of a Virtual Machine Monitor (VMM). VMMs used a interface to the system, because of this entity to be migrated system encapsulates most of the complexity. *i.e.*, the details of events occurring inside the VM can be ignored during migration. Figure 3. shows the migration jobs in cloud computing including process and VM migration.

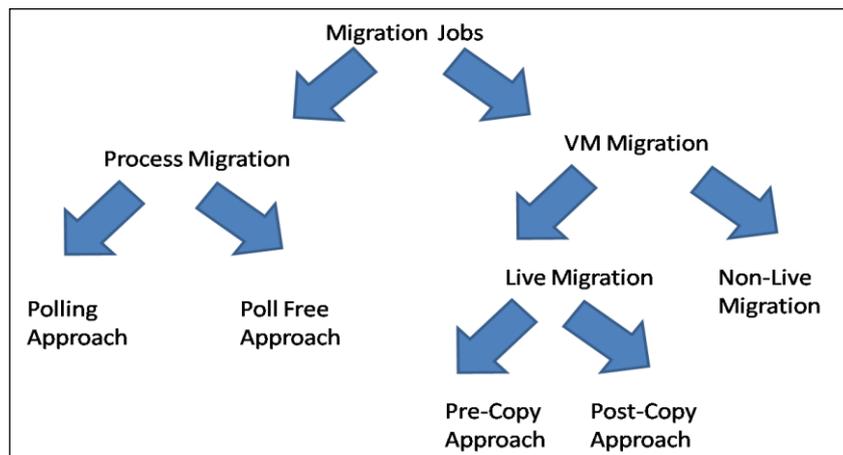


Figure 3. Migration Jobs in Cloud Computing

3. Future Scope

The presented work is done in process migration perform the scheduling and the allocation of the processes to the clouds. Both in case of under load and overload conditions. But, in case of over load condition, the process migration is performed from one cloud to other. The Future scope of the work is possible in the following directions:

- 1) The presented work is defined the overload conditions in terms of deadline as well as the memory limit of the clouds. In future some other parameters can also be taken to decide the migration condition.
- 2) The presented work is defined for the public cloud environment, but in future, the work can be extended to private and the hybrid cloud environment.

Future scope in VM Migration-

VM Migration SLA based- There is a limitation of vm migration technology, when it is used for SLA (Service level Agreements) the optimized data access might still go beyond the time requirement

High dirty rate memory page -The problems like more page fault and higher total migration time occur when high dirty rate of memory pages even memory threshold technique applied.

4. Conclusion

This paper describes a poll-free approach, is suitable for an applications that desiring minimal latency and can afford to effectively enable all points of equivalence. For minimizing the resource consumption the solutions are used to solve the problem. These approaches thus achieve the minimum performance overhead during execution of process. In techniques of live migration of virtual machine includes transfer of a running virtual machine over physical hosts. There are many techniques which are used to minimize the down time & total migration time to provide better performance in low bandwidth. There is less number of network aware migration techniques available which helps more. With the increase in the popularity of cloud computing systems, virtual machine migrations across data centers and resource pools will be greatly beneficial to data center administrators. There is less number of network aware migration techniques available. With the increase in the popularity of cloud computing systems, virtual machine migrations across data centers and resource pools will be greatly beneficial to data center administrators so that downtime and migration time will be reduced.

References

- [1] K. Sammy, R. Shengbing and Ch. Wilson, "Energy Efficient Security Preserving Process Migration In Cloud Computing", International Journal of Computer Science Issues (IJCSI), March (2012).
- [2] P. Bungale, Swaroop Shridhar. "An Approach to Heterogeneous Process State Capture Recovery to Achieve Minimum Performance Overhead during Normal Execution." Proceeding of the International Parallel and Distributed Processing Symposium(IPDPS'03)
- [3] A.J. Ferrari, Stephen J. Chapin, and Andrew S.Grimshaw, "Process Introspection: A Heterogeneous Checkpoint / Restart Mechanism," Department of Computer Science, University of Virginia.
- [4] D.Ajmire, M. Atique"Task migration in Cloud Computing", International Journal of Innovative Research and Development,(2013).
- [5] D.S. Miloji, F.Douglis, Y. Paindaveine, TOG Research Institute, EMC, and "University of Toronto and Platform Computing".
- [6] N. Tzirita, S.U. Khana, Thanasis Loukopoulos "On minimizing resource consumption of cloud applications using process migration" Journal of Parallel and Distributed Computing,(2014).
- [7] L.Gkatzikis," Efficient Task Migration Policies for Cloud Computing Systems", Education and Lifelong Learning, (2014).
- [8] Mohan, Shine, " Survey on Live VM Migration Techniques", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
- [9] M. Chawda, O. Kale," Virtual Machine Migration Techniques in Cloud Environment: A Survey", International Journal for Scientific Research & Development(IJSRD)
- [10] Y.Kejiang, "Live migration of multiple virtual machines with resources." IEEE, (2014).
- [11] Sato, Kento, Hitoshi Sato, and Satoshi Matsuoka. "A model-based algorithm for intensive applications in clouds using vm-based migration, IEEE, (2014).
- [12] Sato, Kento, Hitoshi Sato, and Satoshi Matsuoka. "A model-based algorithm for intensive applications in clouds using vm-based migration, IEEE, (2014).
- [13] Alexander,Thomas Setzer. "Network-aware migration control and scheduling of differentiated virtual machine workloads." IEEE Computer Society, (2013).
- [14] H. Chen, Hui Kang, G. Jiang, Y.Zhang Stage, "Network-aware migration control and scheduling of differentiated virtual machine workloads." IEEE Computer Society, (2012).
- [15] B. Hu, Z. Lei, Y. Lei, D. Xu, and J. Li, "A time-series based precopy approach for live migration of virtual machines", International Conference on Parallel and Distributed Systems (ICPADS).
- [16] C. Xianqin, Gao Xiaopeng, "Application-Transparent Live Migration for Virtual Machine on Network Security Enhanced Hypervisor", China communications, (2012).
- [17] Ei. Zaw and Ni .Thein, "Improved Live VM Migration using LRU", International Journal of Computer Science and Telecommunications, March (2011).
- [18] H.Jin, Li Deng, Song Wu, "Live Virtual Machine Migration with Adaptive Memory Compression", (2010), IEEE.

