A Brief Introduction of Resource Management Techniques in Cloud Computing Environment

Miss. Pragya Tripathi.
Computer Engineering Department
Pillai's Institute of Information Technology
New Panvel
e-mail:123pragya.tripathi@gmail.com

Prof. Manjusha Deshmukh.
Computer Engineering Department
Pillai's Institute of Information Technology
New Panvel
e-mail: mdeshmukh@mes.ac.in

Abstract—Cloud computing has become a new era technology that has huge potentials in enterprises and markets. By using this technology, Cloud user can access applications and associated data from anywhere. It has many application for example, Companies are able to rent recourses from cloud for storage and other computational purposes so that infrastructure cost can be reduced significantly. For managing large amount of virtual machine request ,the cloud providers require an efficient resource scheduling algorithm. Here in this paper we summarize different recourse management strategies and its impacts in cloud system we try to analyze the resource allocation strategies based on various matrices and it points out that some of the strategies are efficient than others in some aspects. So the usability of each of the methods can varied according to their application area .

Keywords: *Cloud computing,QoS,Scheduling,Resource management strategies,Virtualization.*

*****

I. INTRODUCTION

Cloud computing is a technology which provides software, platform and infrastructure as a services. It emerges as a new computing paradigm that promises reliable, custom-made and QoS (Quality of Service) warranted computing in dynamic environments for the end user [4].Cloud computing referred to as the on demand technology because of its it dynamic and versatile resource allocation for reliable and warranted services in pay as-you-use manner to user [4],[5].

It is a technology that takes support of web and central remote servers to take care of data and applications and permits end customers to use applications without any installation and access their personal files at any computer with the help of internet access only [1].

A model for enabling ubiquitous, convenient, on demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. This technology allows rather more efficient computing by consolidative processing , data storage, and bandwidth. The main specialty of this technology which makes it different from other upcoming technology is that any variety of cloud services can be simultaneously accessed by any variety of users. So it is very necessary that every user should get minimum or sufficient resources in a well-organized manner.

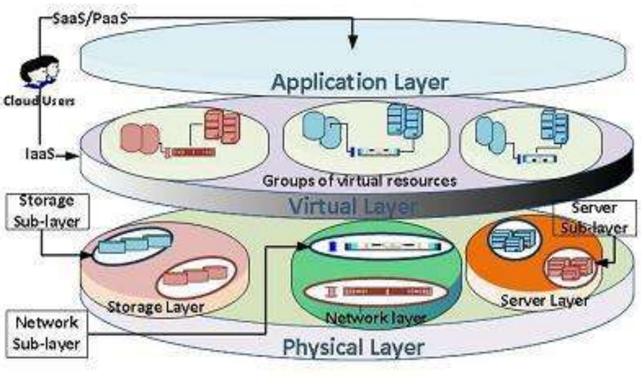

Figure1 Cloud in 3D view

II. MOTIVATION

Due to fast increased use of Cloud Computing, moving of more and more applications day by day on cloud and demand of clients for more services with better results, resource management in Cloud has become a very important research area. Resource management is essential for efficient and effective operations in distributed environments. The cloud also eases much of the technological burden involved with IT systems support and maintenance, helping companies focus on the productive business use of their workloads rather than on underlying systems and software. Regardless of the motivation, business owners and data center managers are increasingly turning to cloud for vital computing services.

III. MAIN TYPES OF RESOURCES IN CLOUD

We now briefly outline the main types of resources that comprise the subjects of a cloud resource management system[1]:

A. *Compute Resources*

Compute resources are typically collection of Physical Machines (PMs), each comprised of one or more memory processors, network interface and local I/O, which all together provide the computational capacity of a cloud environment for end user .

B. *Networking Resources*

Compute resources (on PMs) within a data center are packaged into racks and are typically organized as clusters of thousands of hosts for resource allocation purposes.

Current data center network topologies are based on hierarchical, tree-like topologies similar to those used in early telephony networks, although a number of alternative topologies including proposals based on fat trees , hyper-cubes and randomized small-world topologies have emerged.

C. *Storage Resources*

Public Cloud Providers ,best example Amazon, over persistent storage services of different types, ranging from virtual disks and database services to object stores, each service having different levels of data consistency guarantees and reliability.

D. *Power Resources*

In a data center power is consumed by servers , and power is also required for networking equipment, power distribution equipment, as well as cooling and supporting infrastructure.

IV. RESOURCE MANAGEMENT STRATEGIES

In this section we are aiming to analyze the resource allocation strategy that are already present in the cloud environment and their bedrocks .

A. LINEAR SCHEDULING STRATEGY
B. PRE-COPY APPROACH FOR SCHEDULING
C. MATCH MAKING AND SCHEDULING
D. JUST-IN-TIME RESOURCE ALLOCATION
E. MIYAKODORI: A MECHANISM FOR MEMORY REUSE

A. ***Linear Scheduling Strategy***

The resource allocation is taken into account by the parameters like CPU and memory utilization as well as throughput etc. For every client the cloud environment has to maintain these things so that it could offer maximum service to all of it clients. When we are taking about scheduling of resources and tasks in an individual basis it will create giant waiting time and response time. So as to remove this drawback a new approach called Linear Scheduling for Tasks and Resources (LSTR) is introduced.

Here scheduling algorithms mainly focus on the distribution of the resources among the user or we can say requestors which is able to maximize the chosen QoS parameters. Here QoS parameter which is selected is the cost function. The scheduling algorithm which we are designing is based on the tasks and the total available virtual machines together and named as LSTR scheduling strategy. The best part of this strategy is that it offers maximum resource utilization. The scheduling algorithm is meted out based on the prediction that the initial response to the request is formed merely when assembling the resource for a finite amount of time (say 1 day or 1 hr like that) but not allocating the resource as they arrive. However in case of dynamic allocation , it could be carried out by the scheduler dynamically on request for a few extra resources. This process is achieved by the continuous evaluation of the threshold value in the system. This approach work best when we consider SJF (shortest job first) rather than taking FCFS (first come first serve) approach. The reason behind taking SJF is that the algorithm LSTR sorts the requests on the basis of threshold value rather than arrival time.

Algorithm

```
1.All the requests are collected and processed within every pre determined interval of time .
2.Resources  let say Ri-{$R_1$,$R_2$,$R_3$------$R_n$}
3.Requests $RQ_i$--{$RQ_1$,$RQ_2$,$RQ_3$------$RQ_n$}
4.Threshold should be static at initial point of time.
5.Th=$\sum R_i$
6.For every unsoted array  Aand B
7.Sort A and B
8.For every $RQ_i$
9.Check condition  if $RQ_i$<Th then
10.Add $RQ_i$ in lower array , A[$RQ_i$]
11. Else if $RQ_i$>Th then Add $RQ_i$ in higher array , B[$RQ_i$]
12.For every B[$RQ_i$] allocate resource for $RQ_i$ of B
13.$R_i$=Ri-$RQ_i$; Th=$\sum R_i$
14.Satisfy the  resource of A[$RQ_i$]
15.for every A[$RQ_i$] , allocate resource for $RQ_i$ of A
16. $R_i$=Ri-$RQ_i$; Th=$\sum R_i$
17. Satisfy the  resource of B[$RQ_i$]
```

Best fit strategy is used to satisfy the request alternatively in A[RQi] and B[RQi] which is based on the available VM. The algorithm is explained with a simple input of memory request in GB's such as Ri=(R1,R2,R3…..Rn). The algorithm takes prediction that the initial response to the request is made only after collecting the resource for a finite amount of time let it will be (1 day or 1 hr) but it will not allocate the resource as soon as they arrive. If we have to do the dynamic allocation , then the scheduler have to do this dynamically. This dynamic allocation is made possible by the continuous evaluation of the threshold value[9].

Advantages And Disadvantage Of LSTR

This approach has many advantages but the most important advantage is that it has a better throughput and response time.

The disadvantage of above mentioned algorithm is that it is not appropriate for the interactive real-time applications reason behind it is that algorithm doesn't take into consideration the arrival time during its processing. For interactive real time applications the requests are considered in a “first come first serve” manner. Hence the arrival time is important regarding this type of systems.

B. Precopy Approach For Scheduling

To understand the concept of precopy approach first we have to focus on live migration .

Live Migration

As we know re-placement of VMs means capability to move, or migrate, a VM from one physical host to another. If the migration is performed in such a way that the connected clients do not feel any disturbance during processing ,then this is known as live migration.

Live mgration is useful in many scenarios, for example, in server consolidation, if we are using live migration no need to shut down shut down before they are moved. Live migration is not limited to a single datacenter but it can also be used to transfer VMs between cloud sites over wide area network(WANs). We take a conservative approach to the management of migration with regard to safety and failure handling.

To achieve this, refer figure2, which shows the migration process as a transactional interaction between the two hosts below[10] :

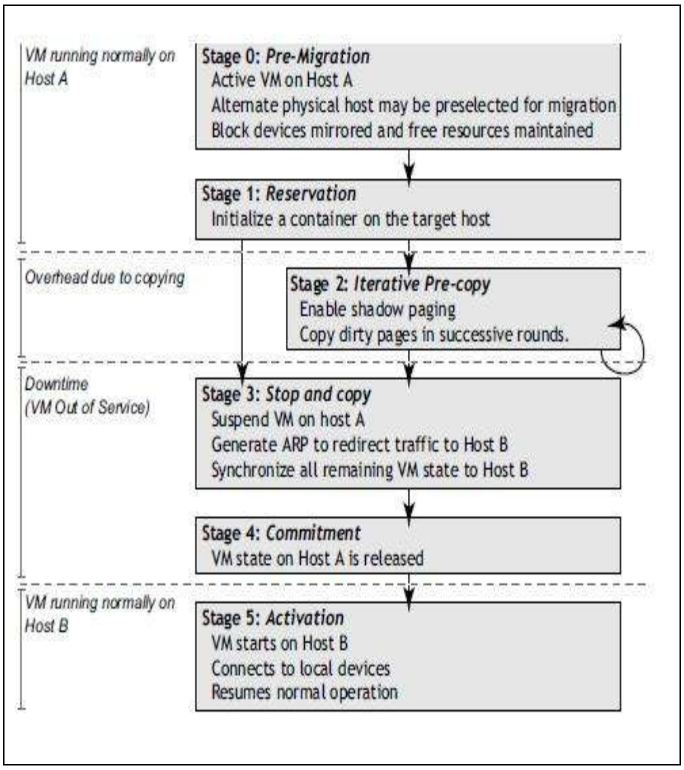
VM running normally on Host A
Stage 0: Pre-Migration
Active VM on Host A
Alternate physical host may be preselected for migration
Block devices mirrored and free resources maintained
Stage 1: Reservation
Initialize a container on the target host
Overhead due to copying
Stage 2: Iterative Pre-copy
Enable shadow paging
Copy dirty pages in successive rounds.
Downtime (VM Out of Service)
Stage 3: Stop and copy
Suspend VM on host A
Generate ARP to redirect traffic to Host B
Synchronize all remaining VM state to Host B
Stage 4: Commitment
VM state on Host A is released
VM running normally on Host B
Stage 5: Activation
VM starts on Host B
Connects to local devices
Resumes normal operation

Figure2 logical steps that we execute when migrating an OS

Precopy Approach

With the help of precopy migration the VM state is transferred in the background in a series of iterations means it takes number of iteration while the source VM is running and responding to requests. When enough of the state has been transferred, the source VM is suspended to stop further memory writes[1]. The decision when to switch is based on two things first one is the amount of memory remaining to transfer, or the switch can be forced after a set amount of time. After the source VM has been suspended, the remaining state is transferred and the VM is finally restart or called as resumed on the destination host.Pre-copy migration technique gives best results ,when memory pages can be copied to the destination host faster than they are dirtied by the migrating VM[3].

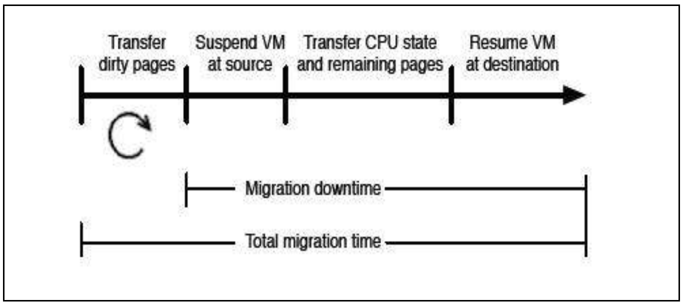

Figure3 Shows precopy migration

Advantages And Disadvantage Of Precopy Migration

The advantage of precopy approach is that it provide Page level protection hardware .

Some of the drawbacks that are faced in the above mentioned approach is that , it will result in long forwarding chains. This will create a delay to the user experiences with the system.

C. Match Making And Scheduling

The main idea behind our new scheduling technique is to give every slave node a chance to hold local tasks before any non-local tasks are assigned to any slave node. Since this algorithm find a match, i.e., a slave node that contains the input data, for every unassigned map task, we named this new technique the matchmaking scheduling algorithm .

But there are some uncertainties that are associated with such type of “match making” and scheduling. Uncertainties can be of type :

1. *Error Associated With Estimation Of Job Execution Time*

This uncertainties is associated with execution time .It is consider that estimating the execution time for a job is a tedious task and errors might happen fairly often. There is many abnormal conditions like formation of “resource idle time” which is happened because of certain unwanted conditions like jobs may run for a smaller time compared to their estimated execution time another condition is estimated time is less than the actual execution time.

The under estimation of job execution times might lead to job terminations because the resource may be booked for executing another job right after the completion of the primary job’s execution. Both of the above conditions i.e. over estimation and under estimation of job execution time are unattractive.

2.Lack Of Knowledge Regarding Local Resource Management Policies

Matchmaking is very tricky and challenging in cloud systems because the scheduling policy used at each resource may not be known to the resource broker. Resource broker have full control for advanced reservations during the request to resource mapping. This is not a good condition the reason behind is that the exact system configuration for a cloud may not be fully known during the time of system design .It can be changed many times during the lifetime of the entire system [4].

D. Just- In-Time Resource Allocation

It focuses on cost based workload prediction and just -in-time resource allocation.

1.Workload Prediction

As the name suggest, here optimization of the system behavior is carried out by taking the minimization of the cost sustain to the application.

2.Just-in-time Resource Allocation

In this just in time resource allocation the three components of the cost function refer individually to the penalty for violation of SLA bounds, cost of leasing a machine, and cost of reconfiguring the application when machines are either leased or released. But for the look-ahead implementation of the time interval for each task need the implementation of recursive data structures. And the prediction of this look-ahead time also results in some prediction[4].

E. MiyakoDori: A Mechanism For Memory Reusing

As mentioned early in this paper , Live migration techniques are used to optimize virtual machine placements. However, these techniques create heavy network traffic, as existing live migration techniques transfer the entire memory image of the target VM.

This process makes migration to take long time and delays for completion of the VM placements optimization. The execution of live migration is considered as a penalty parameter in an optimization problem of VM packing . If we use dynamic VM consolidation system it does not impose such a delay because it uses post-copy live migration. But the drawback behind it is that, a post-copy live migration slows down the performance of a target VM for a certain time after a migration.

To overcome above mentioned problem we propose another efficient technique which we called as memory reusing, a mechanism to reduce the amount of transferred data of a live migration in dynamic VM consolidation. A dynamic VM consolidation system executes live migrations many times. During these migrations, a VM might migrate back to a host on which it has been executed before. When a VM transit from an idle state to a busy state, the VM must be moved from a shared server to an idle server. But when the VM becomes idle again, it is moved to the shared host.

Basic Idea of Memory Reusing

The idea behind memory reusing is that we have to keep memory image of a VM on a host for later use it simply means that , in order to do this we keep the memory image of a VM on a host when the VM migrates from another host at a later stage.

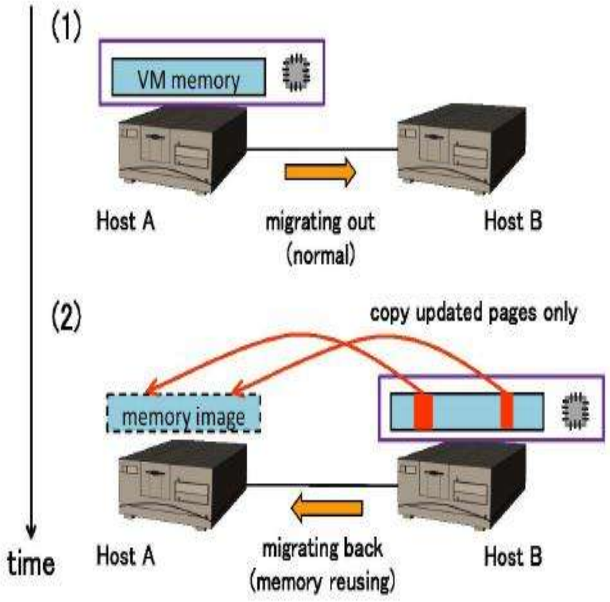

Figure4 shows the basic idea of memory reusing

In the figure4 above upper part of figure indicates the scenario when we are not applying memory reusing so, during migration from host A to host B because it is the first migration of VM . All the memory pages of the VM must be transferred on the network. At this time, we keep the memory image of the VM on host A in order to reuse it later.

The lower part shows the scenario when we are applying memory reuse so in this case , when the VM migrates back from host B to host A , memory pages of the VM have been updated since the memory image has been kept on host A. But many memory pages will not changed and they are not transferred from host B to host A[8].

Live Migration With Memory Reusing: MiyakoDori

If we are using live migration with memory reusing ,here each memory page of a VM has a generation, which tell us how many times the page has been updated since the boot of the VM. At the time of booting of VM, generations of all the memory pages are set to zero. We call the set of all generations of a VM as the generation table of the VM. A generation table is managed by server manages. We use the tuple (V, A) to refer to the generation table of VM V associated with host A.

Figure5 illustrates the behavior of MiyakoDori when a VM V is migrated from host A to host B for the first time so no need of memory reusing.

Here are the steps which explain the figure5 :

1) Stop VM V on host A temporarily.

2) Using dirty page tracking detect memory pages that have been updated since the boot time of VM V .

3) Then updated page numbers are send to the generation server, which do changes to (V, A).

4) Restart VM V. This halt is less than a second because the data transferred in (3) is quite small,

5) Migrates VM V to host B by using a live migration mechanism, but keeps the memory image on host A for later reuses.

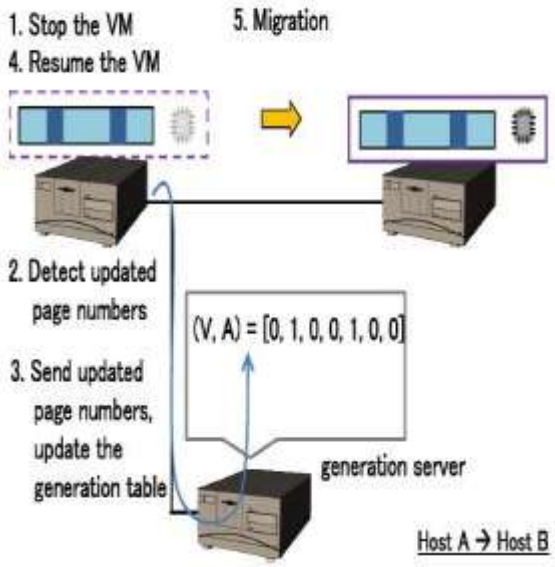
1. Stop the VM
4. Resume the VM
5. Migration
2. Detect updated page numbers
(V, A) = [0, 1, 0, 0, 1, 0, 0]
3. Send updated page numbers, update the generation table
generation server
Host A → Host B

Figure5 First migration of a VM

Here in figure6 all the memory pages that have different generations in the source and the destination are copied. Other pages are reused and do not need to be copied.

Here are the steps which explain the figure6:

1) Again stop VM V on host B temporarily.

2) Using dirty page tracking, detect memory pages that have been updated since the last migration to host B .

3) Then updated page numbers are send to the generation server, which do changes to (V, B).

4) To find reusable pages generation server matches (V, A) with (V, B) to find reusable pages, means those pages that have the same generations in the two generation tables.

5) The generation server then send those reusable page numbers to the hosts.

6) Restart VM V.

7) Migrates VM V to host A, without transferring reusable memory pages.

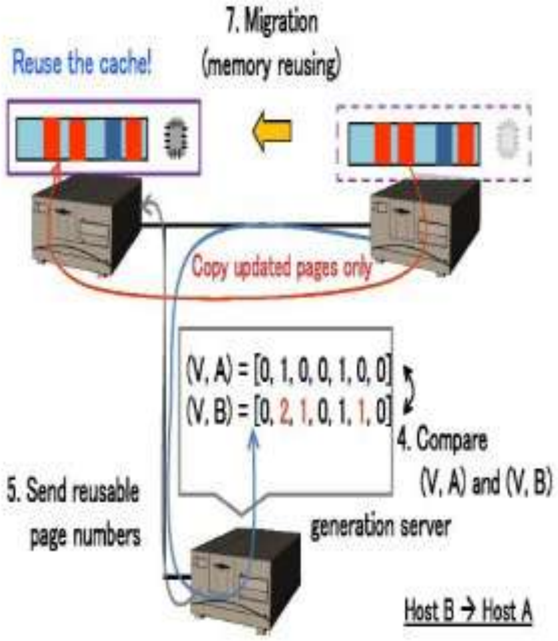
7. Migration (memory reusing)
Reuse the cache!
Copy updated pages only
(V, A) = [0, 1, 0, 0, 1, 0, 0]
(V, B) = [0, 2, 1, 0, 1, 1, 0]
4. Compare (V, A) and (V, B)
5. Send reusable page numbers
generation server
Host B → Host A

Figure6 Migration back to a host on which the VM has once been executed

Advantages And Disadvantage Of Miyak Dori

The advantage of this algorithm is that it reduces the amount of transferred memory and total migration time of a live migration and thus reduces the energy consumption of a dynamic VM consolidation system. It means Memory reuse and Shorter migration time.

The disadvantage is that algorithm is Efficient only in cases where migration back to the same system[8].

Table1. Comparison between the resource management strategies

S.NO	METHOS	MERITS	DEMERITS
1	Linear Scheduling Strategy	1.Improved throughput 2.Response time. 3.Improved resource utilization	1.Not suitable for interactive real time applications
2	Pre-copy Approach	1.Page level protection hardware	1.Long forwarding chains. 2. Delayed user Experiences.
3	Match making and scheduling	Cost effective, less delay	Error Associated with Estimation of Job Execution Times. Lack of Knowledge of Local Resource Management Policies.
4	Just-in-time Resource allocation	Cost effective	Prediction error and use of recursive data structures.
5	MiyakoDori	1.Memory reuse 2. Shorter migration time	1. Efficient only in cases where migration back to the same system.

V. CONCLUSION

Now a days every enterprises and business markets use cloud computing technology in the tremendous rate. cloud environments are being used for many application, an effective resource allocation strategy is required for achieving user satisfaction as well as for maximizing the profit for cloud service providers also.

In this paper, various scheduling algorithms with the perspective of Cost effective, Energy efficient and Security aware are explained. After the detailed study of each of the method we come to the conclusion that is one strategy may be useful for real time interactive application may not be suitable for some other application area.

REFERENCES

[1] Petter Svard, "Dynamic Cloud Resource Management Scheduling, Migration, and Server Disaggregation" ,PHD THESIS April 2014 department of Computer Science UME°A UNIVERSITY SWEDEN.

[2] Dilshad H. Khan, Prof. Deepak Kapgate, Prof. P.S Prasad ,"A Review on Virtual Machine Management Techniques and Scheduling in Cloud Computing", International Journal Volume 3, Issue 12, December 2013 ISSN: 2277 128X .

[3] Sushil Kumar Soni, Assosiate prof. Ravi Kant Kapoor," A Survey on Pre-copy Based live Migration of Virtual Machine in Cloud Environment" ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 10, October 2013.

[4] K. Rasmi and V. Vivek," Resource Management Techniques in Cloud Environment - A Brief Survey" Department of Computer Science and Engineering, Karunya University, Coimbatore, Tamilnadu, India-641 114, 4Apr. 2013.

[5] Vignesh V, Sendhil Kumar KS, Jaisankar N,"resource management and scheduling in cloud environment", International Journal of Scientific and Research Publications, ISSN 2250-3153 Volume 3, Issue 6, June 2013 .

[6] Brendan Jennings and Rolf Stadler R," Resource Management in Clouds: Survey and Research Challenges", March 2013 available at Springer via http://dx.doi.org/10.1007/s10922-014-9307-7.

[7] V.Vinothina, Dr.R.Sridaran, Dr. Padmavathi Ganapathi ," A Survey on Resource Allocation Strategies in Cloud Computing", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No.6, 2012.

[8] Soramichi Akiyama, Takahiro Hirofuchi, Ryousei Takano, Shinichi Honiden , "MiyakoDori: A Memory Reusing Mechanism for Dynamic VM Consolidation",

Fifth International Conference on Cloud Computing, IEEE 2012.

[9] Abirami S.P., Shalini Ramanathan , "Linear Scheduling Strategy for Resource allocation in Cloud Environment", International Journal on Cloud Computing and Architecture, vol.2, No.1, February2012.

[10] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hanseny, Eric July, Christian Limpach, Ian Pratt, Andrew Warfield, "Live Migration of Virtual

Machines", 2nd Symposium on Networked Systems Design and Implementation (NSDI), May 2005.