Volume 3, Issue 9, September 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Server Consolidation Algorithms for Cloud Computing Environment: A Review

Susheel Thakur[*]
Student M.Tech, Computer Science,
Himachal Pradesh University, Shimla
India

Arvind Kalia
Professor,
Himachal Pradesh University, Shimla
India

Jawahar Thakur
Associate Professor,
Himachal Pradesh University, Shimla
India

Abstract— Virtualization technology provides the ability to transfer virtual machines (VM) between the physical systems using the technique of live migration mainly for improving the efficiency. Server consolidation through live migration is an efficient way towards energy conservation in cloud data centers. The main objective of this study is to prevent server sprawl, which refers to the usage of multiple-utilized servers in data centers incurring high system management cost. To prevent server sprawl, server consolidation aims at reducing the number of server machines used in the data centers by consolidating load and enhancing resource utilization of physical systems. Although a lot of research and study has been conducted on server consolidation, a range of issues involved have mostly been presented in isolation of each other. Therefore, here an attempt has been made to present a comprehensive survey to study the different server consolidation algorithms in a cloud computing environment.

Keywords— Cloud computing, live migration, Load balancing, Server Sprawl, Virtual Machine Monitor (VMM).

I. INTRODUCTION

With the rapid development in the processing, storage technologies and the success of the internet, computing resources have become reasonable, powerful and globally available than ever before. Personnel in businesses are trying to find out methods to cut costs while maintaining the same performance standards. Their aspirations to grow have led them to try new ideas and methods even under the peer pressure of limiting computing resources. This realization has enabled the actualization of a new model for computing called cloud computing, in which the resources (e.g. cpu, network, etc.) are provided through the internet to user as general utilities in a pay-as-you-go and on-demand basis.

Cloud Computing is defined by NIST[10] as a model for enabling 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.

For simplicity, a cloud is a pool of physical computing resources i.e. a set of hardware, processors, memory, storage, networks, etc which can be provisioned on demand into services that can grow or shrink in real-time scenario[16]. Virtualization plays a vital role for managing and coordinating the access from the resource pool. A virtualized environment that enables the configuration of systems (i.e. compute power, bandwidth and storage) as well as the creation of individual virtual machines is the key features of the cloud computing. Virtualization is ideal for delivering cloud services. Virtualization Technology enables the decoupling of the application payload from the underlying physical hardware and provides virtualized resources for higher-level applications. An important feature of a virtual machine is that software running inside it is limited to resources and abstractions provided by the virtual machine (VM). The software layer that provides the virtualization is called virtual machine monitor (VMM). VMM virtualizes all of the resources of physical machine, thereby supporting the execution of multiple virtual machines. Virtualization can provide remarkable benefits in cloud computing by enabling VM migration to balance load across the data centers [7].

In the surge of rapid usage of virtualization, migration procedure has been enhanced due to the advantages of live migration say server consolidation and resource isolation. Live migration of virtual machines [4, 9] is a technique in which the virtual machine seems to be active and give responses to end users all time during migration process. Live migration facilitates energy efficiency, online maintenance and load balancing [8]. Live migration helps to optimize the efficient utilization of available cpu resources.

Server consolidation is an approach for the efficient usage of computer server resources in order to reduce the total number of servers or server locations that an organization requires. This approach was developed in response to the problem of "server sprawl". Server sprawl is a situation in which multiple underutilized servers occupies more space and consume more resources that can be justified by their workload. Many organizations are moving towards server consolidation to reduce infrastructure complexity, improve system availability and save money. With increasingly powerful computing hardware, including multi-core servers; organizations can run large workloads and more applications on few servers. Reducing the numbers of servers has tangible benefits for the data center as well.

Server Consolidation can be categorized into:

1. Location Consolidation-The number of physical locations containing hardware is reduced.

2. Physical Consolidation-The number of physical hardware is reduced.
3. Rationalized Consolidation-implementing multiple applications on fewer, more powerful problems, usually through workload management and partitioning.

Workload management describes a set of techniques which enable different applications to run together on a single instance of an operating system (OS). The aim of this technique was to balance the resource demands that each of the applications places on the system so that all of them co-exist. Partitioning involves the division of a server, which might ordinarily run on a single instance of an OS, into smaller systems each of which can run its own copy of an OS. In short, workload management allows several applications to run on one OS, partitioning allows several OS to run on one machine.

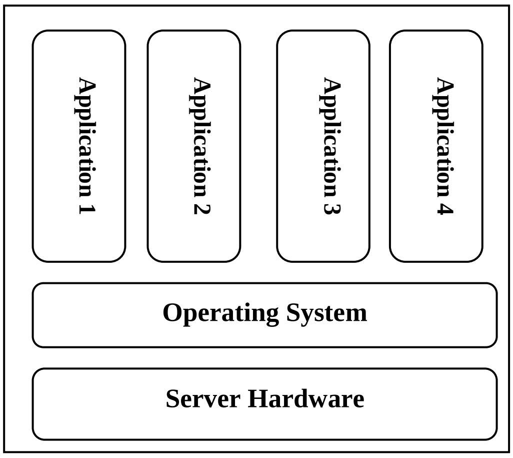

Figure.1 Workload Management

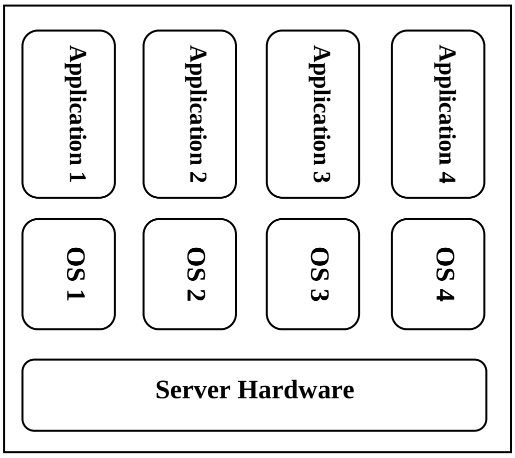

Figure.2 Partitioning

Consolidation will result in reduced power consumption and thus reducing overall operational costs for data center administrators. Live migrations achieve this. Based on the load conditions, under-utilized machines having resource usage above a certain threshold are identified and migrations are triggered to tightly pack VMs to increase overall resource usage on all physical machines (PMs) and free up resources/PMs if possible[2].

The rest of this paper is organized as follows: Section II describes various chosen server consolidation algorithms for cloud computing environment. Section III provides conclusion and future work.

II. SERVER CONSOLIDATION ALGORITHMS

Server consolidation is an approach for the efficient usage of computer server resources in order to reduce the total number of servers or server locations that an organization requires. This approach was developed in response to the problem of "server sprawl". In order to reduce server sprawl in the data centers, server consolidation algorithms are implemented. These algorithms are VM packing heuristics which try to pack as many VMs as possible on the physical machine (PM) so that resource usage is improved and under-utilized machines can be turned off.

A. Sandpiper

Sandpiper is a system that automates the task of monitoring and detecting hotspots, determining a new mapping of physical resources to virtual resources, by resizing or migrating VM's to eliminate the hotspots. Sandpiper makes use of automated black-box and gray box strategies for virtual machine provisioning in cloud data centers. Specifically the black-box strategy can make decisions by simply observing each virtual machine from the outside and without any knowledge of the application resident within each VM. The authors present a gray-box approach that assumes access to OS-level statistics in addition to external observations to better inform the provisioning algorithm. Sandpiper implements a hotspot detection algorithm that determines when to resize or migrate virtual machines, and a hotspot migration algorithm that determines what and where to migrate and how many resources to allocate. The hotspot detection component employs a monitoring and profiling engine that gathers usage statistics on various virtual and physical servers and constructs profiles of resource usage. These profiles are used in conjunction with prediction techniques to detect hotspots in the system. Upon detection, Sandpiper grants additional resources to overloaded servers if available. If necessary, Sandpiper's migration is invoked for further hotspot mitigation. The migration manager employs provisioning techniques to determine the resource needs of overloaded VMs to underloaded servers.

Sandpiper supports both black-box and gray-box monitoring techniques that are combined with profile generation tools to detect hotspots and predict VM Resource requirements. Hotspots are detected when CPU usage values are violated with respect to the CPU thresholds set. Physical machines (PMs) are classified as underloaded or overloaded. The PMs are sorted based on the descending order of their volume metric, and VMs are sorted based on the descending order of their vsr metric, where volume (vol) and vsr are computed as:

$$vol = \left(\frac{1}{1-cpu}\right) * \left(\frac{1}{1-mem}\right) * \left(\frac{1}{1-net}\right) \quad \text{eq. (1)}$$

$$\mathrm{vsr} = \frac{\mathrm{vol}}{\mathrm{size}}$$ eq. (2)

where cpu, mem and net refers to cpu, memory and n/w usages of the PMs and VMs respectively and size refers to the memory footprint of the VM. vsr is the volume-size metric.

To mitigate hotspot on an overloaded PM, the highest vsr VM is migrated to a least loaded PM amongst the underloaded ones. If the least loaded PM can't house the PM, next PM in the sorted order is checked. Similarly, if the VM cannot be housed in any of the underloaded PMs, next VM in the sorted order is checked. This way sandpiper tries to eliminate hotspots by remapping VMs on PMs through migration. The experimental results showed that migration overhead is less than that of swapping overhead; however, swapping increases the chances of mitigating hotspots in cluster with high average utilization [14].

B. Khanna's Algorithm

Gunjan et al., in [6], proposed Dynamic Management Algorithm (DMA) that is based on Polynomial-Time Approximation Scheme (PTAS) heuristic algorithm. The algorithm operates by maintaining two types of ordering lists, which are migration cost list and residual capacity list. The PMs are sorted according to the increasing order of their residual capacities across any resource dimension like CPU. The VMs on each PM are sorted according to the increasing order of their resource utilization like CPU usage. Migration costs of the VMs are determined based on their resource usage i.e. high usage implies high costly migration. Whenever a hotspot is detected on a PM due to violation of upper threshold, VM with least resource usage is chosen for migration to target host which has the least residual capacity to house it. If a PM cannot accommodate the VM, next PM in the sorted order is checked. Similarly, if the VM cannot be accommodated by any of the candidate target PMs, next least usage VM from the sorted order is checked.

Whenever coldspots are detected, the least usage VMs across all the underloaded PMs is chosen and migrated to a targeted PM, only if addition of the new VM increases the variance of residual capacities across all the PMs, else we choose the next VM in order. If there is no residual space left for the chosen VM, then the heuristic for coldspot mitigation stops. Variance is defined as follows:

$$\mathrm{variance}, \mathrm{R(t)} = \frac{(\mathrm{mean} - \mathrm{rescpu})^2 + (\mathrm{mean} - \mathrm{resmem})^2 + (\mathrm{mean} - \mathrm{resnet})^2 \ldots}{(\mathrm{m} - 1)}$$ eq. (3)

$$\mathrm{mean} = \frac{\mathrm{rescpu} + \mathrm{resmem} + \mathrm{resnet} + \cdots}{\mathrm{m}}$$ eq. (4)

$$r_n = \sqrt{\mathrm{var}_{p1}^2 + \mathrm{var}_{p2}^2 \ldots + \mathrm{var}_{pn}^2}$$ eq. (5)

In above equation 4, mean is defined as the average of normalized residual capacities across 'm' different resources like cpu, memory, networks, etc. rescpu , resmem , resnet … stands for residual capacities across different resource dimensions. In equation 5, r_n is the magnitude of the vector which comprises of the individual variances across 'n' physical machines.

TABLE I: Chosen Migration Heuristics

Algorithms	**Goal**	**Metrics Used**	**Hypervisor**	**Resource Considered**
Sandpiper[14]	Hotspot Mitigation	Volume, Volume/size	Xen	cpu, memory & network
Khanna's Algorithm[6]	Server Consolidation	Residual Capacity, Variance	VMware ESX	cpu, memory
Entropy[5]	Server Consolidation	No. of Migrations	Xen 3.0.3	cpu, memory
Sercon[3]	Server Consolidation	Cpu & Memory Load Capacity	Sercon Software	cpu, memory
Memory Buddies[15]	Server Consolidation & Hotspot Mitigation	Memory Size	VMware ESX	memory
MiyakoDori[13]	Server Consolidation	Dirty Page Bit	Qemu/KVM	Memory

Khanna’s Algorithm packs the VMs as tightly as possible trying to minimize the number of PMs by maximizing the variance across all the PMs. Thus, Khanna’s algorithm minimizes power consumption by detecting underutilization in the managed using Max-Min thresholds selection model. When the resource usage of a running PM violates a minimum predefined threshold value, the algorithm tries to pack the running VMs as close as possible thus trying to minimize the number of running physical machines.

C. Entropy

Entropy proposes a consolidation algorithm based on constraint problem solving. The main idea of the constraint programming based resource manager is to formulate the VM resource allocation problem as constraint satisfaction problem, and then applies a constraint solver to solve the optimization problem. The ability of this solver to find the global optimum solution is the main motivation to take this approach. Entropy resource manager utilizes Choco constraint solver to achieve the objectives of minimizing the number of the running nodes and minimizing the migration cost. Entropy iteratively checks optimality constraint i.e. the current placement uses minimum number of the running nodes. If Entropy is successful in constructing a new optimal placement (uses fewer nodes) at VM packing problem (VMPP) phase, it will activate the re-allocation. Entropy employs a migration cost model that relates memory and CPU usage with migration context. High parallelism migration steps increases the cost. Using constraint programming techniques facilitates the task of capturing such context in two phases. In the first phase, Entropy computes a tentative placement (mapping of VMs to PMs) based on the current topology and resource usage of PMs and VMs and reconfiguration plan needed to achieve the placement using minimum number of PMs required. In the second phase, it tries to improve the reconfiguration plan by reducing the number of migrations required. Since obtaining the placement and reconfiguration may take a considerable amount of time, the time given to the CSP solver is defined by the users, exceeding which whatever immediate value the solver has computed is considered for dynamic placement of VMs. VMs are classified as active or inactive based on their usage of CPU with respect to thresholds set. The author define a viable configuration as one in which every active VM present in the cluster has access to sufficient cpu and memory resources on any PM. There can be any number of inactive VM on the PM satisfying the constraint. The CSP solver takes this viable condition into account in addition to the resource constraints, while procuring the final placement plan. However, considering only viable processing nodes and Cpu-Memory Resource model is the limitation of the Entropy model [5].

TABLE II: Virtual Machine (VM) Migration Heuristics

Algorithms	**Goal**	**When to migrate?**	**Which to migrate?**	**Where to Migrate?**
Sandpiper[14]	Hotspot Mitigation	Resource usage exceed thresholds set	Most loaded VM	Least loaded PM
Khanna's Algorithm[6]	Server Consolidation	Resource usage violate the thresholds set	VM with lowest resource usage	Best first PM by residual capacity
Entropy[5]	Server Consolidation	No. of Migrations	Whichever minimizes reconfiguration cost.	Whichever minimizes reconfiguration cost.
Sercon[3]	Server Consolidation	Resource usage is less than minimum thresholds set	VM with lowest resource usage.	Most loaded VM, trying to compact them as possible
Memory Buddies[15]	Server Consolidation & Hotspot Mitigation	Resource usage is less than minimum threshold set.	VM with highest resource usage.	Best feasible PM that can accommodate that VM.
MiyakoDori[13]	Server Consolidation	Resource usage exceeds threshold set.	Most loaded VM	Best fit PM and back to source host once task is completed.

D. Sercon

This server consolidation algorithm aimed at consolidating the virtual machines (VMs) so that minimum nodes (physical machines) are used and reducing the number of VM migrations. Certain constraints are taken into account in Sercon. They include compatible virtualization software, comparable CPU’s types, and similar n/w topology and shared storage usage on both source and destination nodes, choosing the right value of CPU threshold to prevent performance degradation and migration of VM is done if it results in releasing a node. The algorithm goes like this: The nodes based on loads by VMs are sorted in decreasing order. Then, the VMs on the least loaded node in this list are selected as a candidate for migration and are again sorted according to their weights. They are allocated to the most loaded node first and so on, thus trying to compact them and so can release the least loaded node. By using this method, we can avoid numerous migrations which might otherwise be necessary if there are nodes that are still least loaded. These steps are repeated until no more migrations are possible. Cpu and Memory are considered for representing load in VMs [3].

E. MiyakoDori: Memory Reusing Mechanism

MiyakoDori proposes a memory reusing mechanism to reduce the amount of transferred data in a live migration process. In the case of dynamic VM consolidation, a VM may migrate back to the host where it was once executed. The memory image of the VM is left on the host when it migrates out from the host and the image will be reused when the VM migrate back to host later. The few amounts of data contribute to a shorter migration time and greater optimization by VM placement algorithms. Evaluations showed MiyakoDori 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 [13].

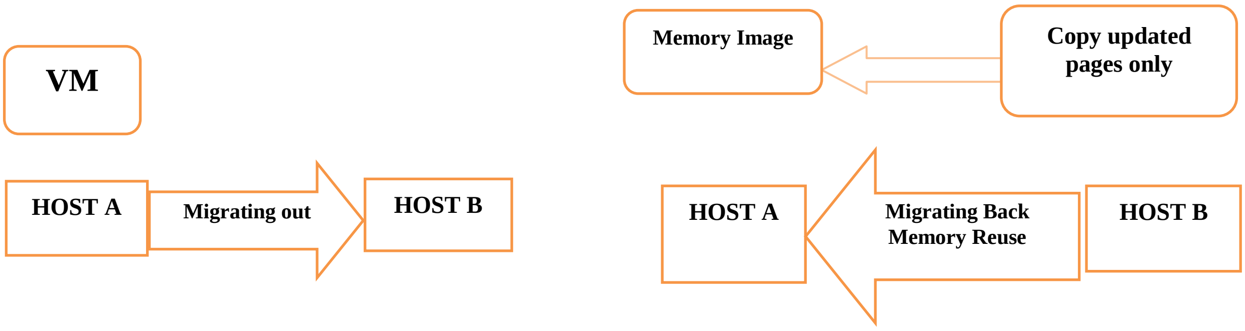

FIGURE.2 Basic idea of Memory reusing

F. Memory Buddies

Memory Buddies is a memory sharing aware placement system for virtual machines. It is a memory fingerprinting system to efficiently determine the page sharing potential among a set of VMs, and compute more efficient placements. It makes use of live migration to optimize VM placement as workload changes. Memory buddies detects sharing potential to realize these benefits. The memory buddies system consists of a nucleus, which runs on each server, and a control pane, which runs on distinguished control server. Each nucleus generates a memory footprint of all memory pages within the VMs resident on that server. This fingerprint represents the page-level memory contents of a VM in a way which allows efficient calculation of the number of pages with identical content across two VMs. The control plane is responsible for virtual migration placement and hotspot mitigation. For placing a virtual machine, it compares the fingerprint of that VM against server fingerprints in order to determine a location for it which will maximize sharing opportunities. It then places the VM on that server, initiating migrations if necessary. The control plane interacts with the VMs through a VM management API such as VMware's Virtual Infrastructure or the libvirt API.

Memory Buddies server consolidation algorithm opportunistically identifies servers that are candidates for shutting down and attempts to migrate virtual machines to hosts with high sharing opportunities. In doing so, it attempts to pack VMs onto servers so as to reduce aggregate memory footprint and maximize the number of VMs that can be housed in the data center. Once the migrations are completed the consolidation candidates can be retired from service or powered down until new server capacity is required, thereby saving on operational costs. The consolidation algorithm runs periodically to check the list of hosts which are candidates for consolidating if its mean usage remains below a low threshold for an extended duration. The system considers only memory usages when identifying consolidation candidates. Once the consolidation candidates are identified, the algorithm determines a new physical server to house each VM. To do so, VMs are arranged according to their decreasing order of the memory sizes and consider them one at a time. Firstly for each VM, the algorithm determines the set of feasible servers in the data centers and then the host which will provide the greatest level of sharing is selected for each VM. Once new targets have been determined for each VM on the consolidation servers, actual migration is performed using live migration. Live migration ensures transparency and near-zero down-times for the application executing inside the migrated VMs. To ensure minimum impact of network copying triggered by each migration of application performance, the algorithm places a limit on the number of the concurrent migrations; once each migration completes, a pending one is triggered until all VMs have migrated to their new hosts. The servers are then powered off and retired or moved to a shutdown so that they can be reinitialized later if memory requirements increase [15].

III. CONCLUSION AND FUTURE WORK

With the popularity of cloud computing systems, live virtual machines migration will be great beneficial tool for dynamic resource management in the modern day data centers. To prevent server sprawl, server consolidation aims at reducing the number of server machines by consolidating load, enhancing resource utilization of physical systems along with provision of isolation & security of the application hosted. In this study, a survey of the various server consolidation algorithms for cloud computing has been presented.

Sandpiper, Sercon, Memory Buddies and Khanna's Algorithm uses a threshold based technique of triggering VM migrations. Entropy relies on CSP solver 4.4.1 to perform consolidation by providing a set of constraints, optimizing the number of PMs needed to house the VMs and the migration cost to determine the selection of configuration plan. In sandpiper, the migration cost is in terms of vsr metric whereas Khanna's algorithm considers the resource utilization as the migration cost metric. Sercon maintains two parameters: total migrations and unsuccessful migration attempts, respectively for computing the migration cost. All of them intend to reduce migration cost in terms of the memory allocated to the VMs. Unlike other algorithms, Entropy tries to obtain a globally optimal solution, which distinguishes itself in its consolidation approach. Unlike other algorithms does, Entropy considers all the hosts in the topology and based on their current resource usages, finds out an optimal solution which tries to decrease the migration overhead in terms of memory. The other algorithms try to achieve consolidation on a per host basis, making sure that resource violations are prevented every time each host is scanned, and then the VMs are packed as closely as possible. Memory Buddies is a memory fingerprinting system to efficiently determine the sharing potential among a set of VMs, and compute more efficient placements by 17%. The system effectively detects and resolves memory hotspots due to changes in sharing patterns. Moreover memory buddies can be easily integrated in existing hypervisors such as VMware ESX server and their management infrastructure to optimize the placement and migration of VMs in Data Centers. Memory Buddies takes into account on only memory resource.

MiyakoDori is a memory reusing mechanism to reduce the amount of data transferred in a live migration. A VM migrates back to the host it has once executed during dynamic VM consolidation. The fewer amount of data contributes to shorter migration time and greater optimization by VM placement algorithms and thus reduces the energy consumption of a dynamic VM consolidation system. Memory resource is considered in MiyakoDori.

Most researchers go through the tedious process of designing their own set-up to test their formulated algorithm. Several algorithms exists in literature, which are trying to efficiently prevent server sprawl or ensuring non-disruptive load balancing in data centers. Efficiency of the algorithm depends on the resource parameters and metrics considered. It is not straightforward to state which algorithm is better than the other and in which case one might be beneficial because an algorithm which tries to minimize system load will have different evaluation metrics over the one which tries to consolidate servers. Cost of migration is also calculated in different ways. Hence, a structured quantitative study is needed to analyse their applicability, goodness and incurred overhead.

Moreover, more algorithms which does similar jobs like consolidation can be chosen in near future and their relative behaviour can be analysed with the already chosen algorithms. Evaluation of more algorithms can facilitate in figuring out the distinct cases where an algorithm will behave well and hence can be used in those cases only to leverage the maximum benefits.

REFERENCES

[1] A. Souvik Pal and B. Prasant Kumar Pattnaik, “Classification of Virtualization Environment for Cloud Computing”, *International Journal of Science and Technology*, vol 6, Jan. 2013.

[2] Anju Mohan and Shine S, “Survey on Live VM Migration Techniques”, *International Journal of Advanced Research in Computer Engineering & Technology*, vol 2, Jan. 2013.

[3] Aziz Murtazaer and Sangyoon Oh, “Sercon: Server Consolidation Algorithm using live migration of virtual machines for Green Computing”, *IETE TECHNICAL REVIEW*, vol 28, May-Jun. 2011.

[4] C. Clark, K. Fraser, S. Hand, J. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfiled, “Live Migration of Virtual Machines”, *In Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation*, vol. 2, pp. 286, 2005.

[5] Fabein Hermenier, Xavier Lorca, Jean-Marc Menuad, Gilles Muller and Julia Lawall, “Entropy: A Consolidation Machine Manager for Clusters”, *In Proceedings of the 2009 ACM SIGPLAN/SIGOPS Internal Conference on Virtual Execution Networks, VEE, 2008*, pp.41-50, 2009.

[6] Gunjan Khanna, Kirk Beaty, Gautam Kar and Andrzej Kochut, “Application Performance Management in Virtualized Server Environments”, *10th IEEE/IFIP Conference on Network Operations and Management in Virtualized Server Environments, NOMS*, 2006.

[7] Jyothi Sekhar, Getzi Jeba and S. Durga, “A survey on Energy Efficient Server Consolidation through VM Live Migration”, *International Journal of Advances in Engineering and Technology*, vol 5, pp.515-525, Nov. 2012.

[8] Kejiang Ye, Xiaohong Jiang, Dawei Huang, Jianhai Chen, and Bei Wang, “Live migration of Multiple Virtual Machines with Resource Reservation in Cloud Computing Environments” , *In Proceedings of 2011 IEEE 4th International Conference On Cloud Computing*, pp. 267-274, 2011.

[9] M. Nelson, B. Lim, and G. Hutchins, “Fast transparent migration for virtual machines”, *In Proceedings of the Annual Conference on USENIX Annual Technical Conference*, pp. 25, 2005.

[10] NIST Definition of Cloud Computing v15, *www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf* Retrieved 14 Oct, 2012.

[11] Sarnpreet Kaur and Ravi Bhushan, “Resource Optimization of Servers using Virtualization”, *International Journal of Advance Research in Computer Science and Software Engineering*, vol 3, pp.323-326, Jun.2013.

[12] Sarnpreet Kaur and Ravi Bhushan, “Review Paper on Resource Optimization of Servers using Virtualization”, *International Journal of Advance Research in Computer Science and Software Engineering*, vol 3, pp.327-332, Jun.2013.

[13] 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.

[14] Timothy Wood, Prashant Shenoy, Arun Venkataramani and Mazin Yousif, “Sandpiper: Black-box and Gray-box Resource Management for Virtual Machines”, *Journal of Computer Networks*, vol.53, pp.2923-2938, Dec.2009.

[15] T.Wood, G. Tarasuk-Levin, Prashant Shenoy, Peter desnoyers, Emmanuel Cecchet and M.D.Corner “Memory Buddies:Exploiting Page sharing for Smart colocation in Virtualized Data Centers” *In Proc. of the ACM SIGPLAN/SIGOPS International conference on Virtual execution environments,VEE*,pages 31–40, New York, NY, USA, 2009.

[16] V. Sarathy, P. Narayan, and Rao Mikkilineni, “Next Generation Cloud Computing Architecture- Enabling Real-time Dynamism for Shared Distributed Physical Infrastructure”, *19th IEEE International Workshops on enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE*, pp.48- 53, June 2010.