ISSN: 2321-7782 (Online)
Volume 2, Issue 3, March 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Paper / Case Study
Available online at: www.ijarcsms.com

Performance Analysis of Server Consolidation Algorithms in Virtualized Cloud Environment with Idle VMs

Susheel Thakur[1]
Student M.Tech
Department of Computer Science
Himachal Pradesh University
Shimla – India

Arvind Kalia[2]
Professor
Department of Computer Science
Himachal Pradesh University
Shimla – India

Jawahar Thakur[3]
Associate Professor
Department of Computer Science
Himachal Pradesh University
Shimla – India

Abstract: Virtualization is a vital concept in enabling the "Computing-as-a-service" vision of cloud based solutions. Virtualization technology provides the ability to transfer virtual machines (VMs) between the physical systems using the technique of live migration mainly for improving the efficiency. Server Consolidation through live migration provides an efficient way towards energy conservation in cloud centers. Although a lot of research and study has been conducted on server consolidation, a range of issues have mostly been presented in isolation of each other. The focus of this research paper is to present the details of the server consolidation heuristics and their usage toward dynamic resource management in the virtualized cloud computing environment. We try to simulate and investigate the impacts of different server consolidation heuristics on the performance of live migration in both source and target machine. Here, we present performance evaluation of these heuristics and fundamental insights aimed at reducing server sprawl, minimizing power consumption and balancing load across physical machines

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

I. INTRODUCTION

In 1969, Leonard Kleinrock [12], one of the chief scientists of the original Advanced Research Projects Agency Network (ARPANET) which seeded the Internet, said: "As of now, computer networks are still in their infancy, but as they grow up and become sophisticated, we will probably see the spread of computer utilities‟ which, like present electric and telephone utilities, will service individual homes and offices across the country." This vision of computing utilities based on a service provisioning model anticipated the massive transformation of the entire computing industry in the 21st century whereby computing services will be readily available on demand, like other utility services available in today's society. Cloud Computing is defined by NIST[17] 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.

With the rapid development in the processing and 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, n/w, etc.) are provided through the internet to user as general utilities in a pay-as-you-go and on-demand basis. 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[22]. 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 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 [10].

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 [5, 14] 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 [11]. Live migration helps to optimize the efficient utilization of available CPU resources.

Server consolidation is an approach to 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 accommodate more space and consume more resources that can be justified by their workload. Many organizations are turning to 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 centers as well.

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 PMs and free up resources/PMs if possible[3].

The rest of this paper is organized as follows: Section II describes the survey of existing literature of various server consolidation algorithms for cloud computing environment. Section III provides the overview of the chosen server consolidation algorithms for performance analysis in virtualized cloud environment. Section IV gives details about the experimental test bed used in performance analysis. Section V discusses the experimental evaluation and results. Section VI provides the conclusion and future work.

II. LITERATURE SURVEY

Several research groups are working on server consolidation in both academia and industry. This section presents studies and systems related to server consolidation. Bobroff et al. [15] proposed and evaluated a dynamic server consolidation algorithm to reduce the amount of required capacity and rate of SLA violations. The algorithm makes use of historical data to predict future demand and relies on periodic executions to consolidate the number of physical servers to support the virtual machines. Verma et al. [1] designed the pMapper architecture and a set of server consolidation algorithms for heterogeneous virtualized resources. The algorithm takes into consideration power and migration costs and the performance benefits while consolidating applications into physical servers. Mehta and Neogi [19] introduced the ReCon tool, aimed at recommending dynamic server consolidation in multi-cluster data centers. This tool considers static and dynamic costs of physical servers, the cost of VM migrations, and the historical resource consumption data from the existing environment to provide an optimal dynamic plan for mapping VMs to physical server over time. Ameek et al. [2] has developed a HARMONY set-up with ESX server and SAN (storage area network) controller for integrating both server and storage virtualization technologies to design an agile data centers. Keller et al. [7] design Golondrina multi-resource management for operating system-level virtualized environment with client systems, manager server and cluster gate. Starling [9] introduced affinity based VM placement and migration in a decentralized approach with a 8-node cluster of 2x dual-core AMD machines. Speitkamp and Bichler [4, 13] introduced a linear programming for the static and dynamic server consolidation problems. They also formulated extension constraints for limiting the number of virtual machines in a physical server, mapping virtual machines to a specific set of physical servers that have some unique attribute and, limiting the total number of migrations for dynamic consolidation. In addition, they also designed an LP-relaxation based heuristic for minimizing the cost of solving the linear programming formulations.

III. SERVER CONSOLIDATION ALGORITHMS

Server consolidation is an approach to 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 and vsr are computed as:

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

$$vsr = \frac{vol}{size}$$

where cpu, memory and n/w refers to cpu, memory and n/w usages of the PMs and VMs respectively and size refers to the memory footprint of the VM.

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 [20, 21].

TABLE 2.1: Chosen Migration Heuristics

Algorithms	Goal	Metrics Used	Virtualization	Resource Considered	Platform
Sandpiper[14]	Hotspot Mitigation	Volume, Volume/size	Yes	cpu, memory & network	Xen 4.1
Khanna's Algorithm[6]	Server Consolidation	Residual Capacity, Variance	Yes	cpu, memory	Xen 4.1
Entropy[5]	Server Consolidation	No. of Migrations	Yes	cpu, memory	Xen 4.1

B. Khanna's Algorithm

Khanna et al., in [8, 20], 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:

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

$$\text{mean} = \frac{\text{rescpu} + \text{resmem} + \text{resnet} + \cdots}{m}$$

$$r_n = \sqrt{\text{var}_{p1}^2 + \text{var}_{p2}^2 \ldots + \text{var}_{pn}^2}$$

In above equation, 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. r_n is the magnitude of the vector which comprises of the individual variances across 'n' physical machines.

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.

TABLE 3.2: VM Migration Heuristics [20]

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.

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 [6, 20].

IV. EXPERIMENTAL TEST BED

Our implementation and evaluation is based on Xen Hypervisor. Four PMs with Xen 4.1 hypervisor installed were used to serve as the physical hosts and another machine was used as NFS [16] Server to house the VMs images. Physical machines which worked as clients were used simultaneously to generate load on the virtual machines hosted on the PMs. Seven VMs were created with Ubuntu 10.04, lucid host operating system with each 256 MB memory size and Ubuntu 11.10 as PMs Host operating system. They all have Apache, PHP and MySQL configured on them to act as web and Database servers. A separate machine has been configured which acts as the Management node, which runs the controller and the Decision Engine. The VIRT-M cluster management tool was implemented on this management node. Python programming was used to prototype these heuristics. Apart from these, RRD tool [17] was installed on the Management node running the Decision Engine, for storage of resource data. Our Experimentation takes these heuristics into consideration and implements these on idle VMs to investigate their impacts on performance of live migration in both source and target machine.

V. EVALUATION AND RESULTS

An experiment with idle VMs was performed to check how reactively the algorithms behave towards consolidation when no workload is generated. The following topology was created:

PM84 contained 1 VM - 10.1.6.252 lucid08

PM170 contained 1 VM -10.1.6.214 lucid09

PM161 contained 2 VMs -10.1.6.210 lucid12 and (10.1.6.246) lucid10

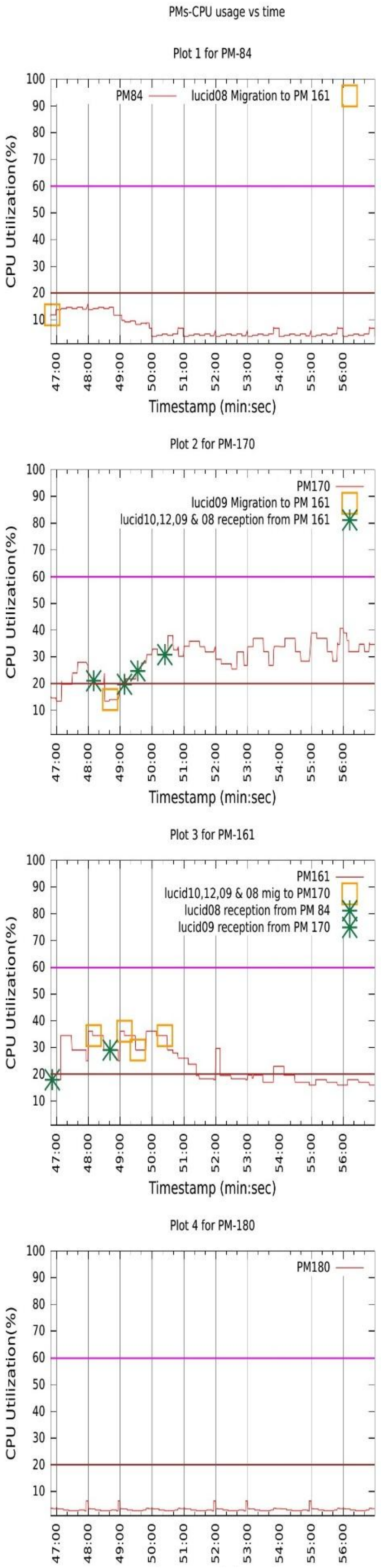

Figure 5.1 Khanna's Algorithm

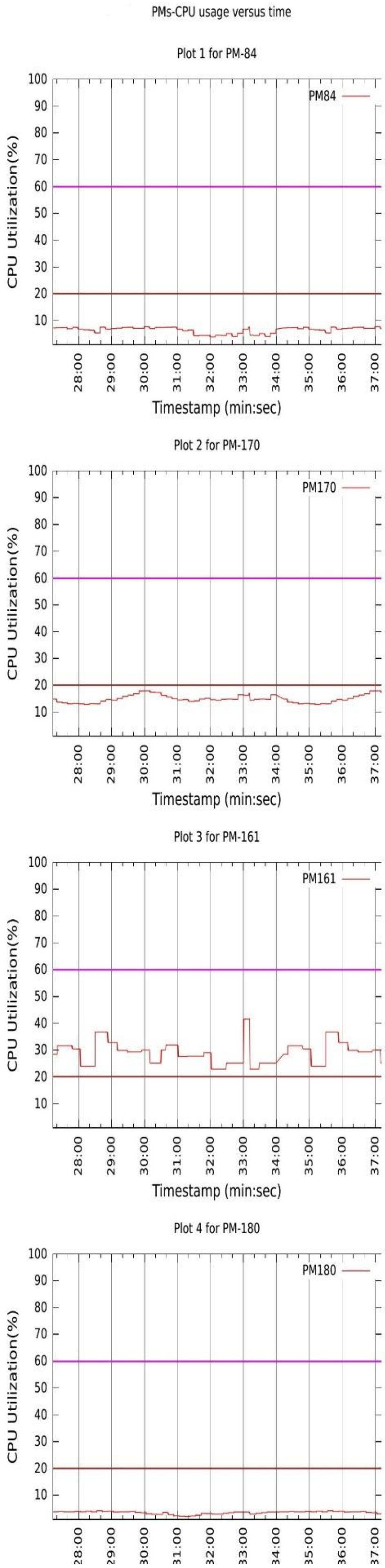

Figure 5.2 Sandpiper

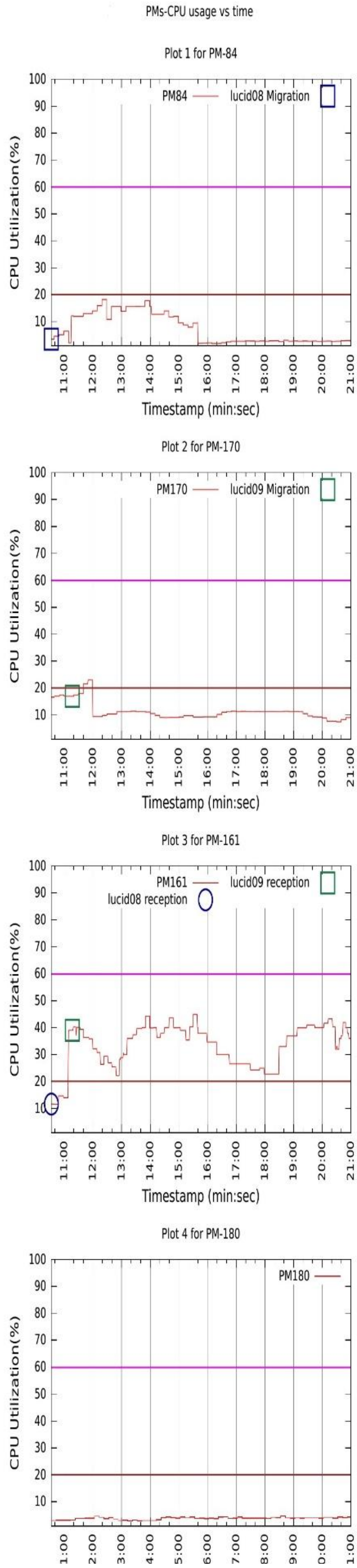

Figure 5.3 Entropy Algorithm

PM180- 10.129.41.180 – empty

VMs have been named as lucid08, lucid09 and so on. This was the topology to start with. The three algorithms were executed for 10 minutes duration. The results are depicted in figures 5.1, 5.2 and 5.3. As seen from figures 5.1, 5.2 & 5.3, the cpu utilizations of PMs in case of sandpiper is more or less stable but has very less fluctuations. Since there were no threshold violations, no possibility of migrations exist in case of Sandpiper. It was observed that the cpu utilization of PM161 is slightly higher, although just two idle VMs are running, remote logging through "sshd" commands from PM161 to other machines for running around 6 scripts per machine might have contributed to some extent for a slightly higher cpu usage. Khanna's Algorithm has lot of variations because of instant detection of coldspots and hence, the utilizations had some variations due to migrations. Entropy also triggers migrations due to which there are fluctuations in cpu usage levels, more for PM161 than the others which will be discussed below. After the experimental run, the new topology generated was:

• Sandpiper - no change

• Khanna's Algorithm - all the VMs in PM170

• Entropy - all the VMs in PM161

From figures 5.1, 5.2 and 5.3, it was observed that Khanna's Algorithm triggers migration of lucid08 from PM84 to PM161 very soon, because it detected a coldspot on PM84. Its coldspot mitigation heuristic sorted all four VMs as per their current utilizations and initiated a series of migrations, trying to increase the overall system variance. It frees up PM84 and PM161 in approximately 4 minutes and 10 seconds from the start of the algorithm. Since PM180 was empty at the onset, it was asked to be turned off by the algorithm. Thus, all the VMs were packed on PM170 reducing 2 machines. It can be very well observed that the gradual increase in CPU usage for PM170 after it received VMs from PM161. Also, the fact that PM161 was not chosen as the PM for packing all the VMs verified the correctness of Khanna's algorithm. It uses the PM with least residual capacity big enough to hold the VM, PM161 had more capacity than PM170, and hence PM161 is not expected for selection. Entropy on the other hand, finds out that only 1 PM is needed to house all the VMs, its configuration plan entailed a set of two migrations, lucid09 from PM170 and lucid08 from PM84. Using 1 PM, at least 3 migrations are needed if two VMs from PM161 were migrated. Since entropy optimizes the migration cost, it chooses PM161 as the final destination for all the VMs by reducing 1 extra migration. This packing was done in little more than 1 minute time. The increasing cpu utilization at PM161 due to VM reception from other PMs and decreasing cpu usage at the other PMs could be observed distinctly.

The following table enlists the measured statistics:

Table 5.1: Experiment with idle VMs-Measured Evaluation Metrics

Algorithm	**No. of PMs**	**No. of Migrations**	**Time Taken(mins)**
Sandpiper	3	0	N/A
Khanna's Algorithm	1	6	4.10
Entropy	1	2	1.35

Clearly Khanna's Algorithm issues more migration increasing the migration overhead over entropy. But it looks up for a better fit of PMs for all the VMs over entropy.

VI. CONCLUSION AND FUTURE SCOPE

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 paper, we presented a performance evaluation of the chosen server consolidation algorithms in virtualized cloud computing environment when no workload is generated on the client machines. Sandpiper 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. 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.

While dealing with idle VMs, Khanna's Algorithm takes more time to consolidate VMs over entropy, because it detects a coldspot immediately and triggers migrations to increase the overall system variance. Entropy globally finds out an optimal solution to pack the VMs, minimizing the memory transfer of VMs. Since Khanna's Algorithm tries to find a best fit solution, it packs all the VMs on a PM which is just sufficient to house all the VMs, but entropy on the other hand, do not enlist any explicit way of checking the best fit solution, as its primary emphasis is on reducing the migration overhead. These algorithms try to efficiently prevent server sprawl and ensure non-disruptive load balancing in data centers. Efficiency of the algorithm depends on the resource parameters and metrics considered. Hence, a comparative performance analysis was carried out to analyse their applicability, goodness and incurred overhead. In near future, Evaluation of these algorithms with constant and variable load 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.

Moreover, more algorithms which does similar jobs like consolidation can be chosen in near future and their relative behavior can be analysed with the already chosen algorithms.

References

1. A. Verma, P. Ahuja, A. Neogi, "pMapper: Power and migration cost aware application placement in virtualized systems", in Proceedings of the ACM/IFIP/USENIX 9th International Middleware Conference, 2008.
2. Aameek Singh, Madhukar Korupolu, Dushmanta Mohapatra. "Server-Storage Virtualization: Integration and Load Balancing in Data Centers", in Proceedings of the SC 2008 ACM/IEEE conference on Supercomputing, 2008.
3. Anju Mohan and Shine S, "Survey on Live VM Migration Techniques", International Journal of Advanced Research in Computer Engineering & Technology, vol 2, Jan. 2013.
4. B. Speitkamp, M. Bichler, "A Mathematical Programming Approach for Server Consolidation Problems in Virtualized Data Centers", IEEE Transactions on Services Computing.
5. 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.
6. 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.
7. G. Keller and H. Lutfiyya, "Replication and migration as resource management mechanisms for virtualized environments", in Proceedings of the Sixth International Conference on Autonomic and Autonomous Systems (ICAS), pages 137-143, 7-13, 2010.
8. 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.
9. James Greensky, Jason Sonnek, Robert Reutiman, and Abhishek Chandra, "Starling: Minimizing communication overhead in virtualized computing platforms using decentralized affinity-aware migration", 39th International Conference on Parallel Processing (ICPP), pages 228-237, September 2010.
10. 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.
11. 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.
12. L. Kleinrock, "A Vision for the Internet", ST Journal of Research, 2(1):4-5, Nov, 2005.
13. M. Bichler, T. Setzer, B. Speitkamp, "Capacity planning for virtualized servers", in Proceedings of the 16th Annual Workshop on Information Technologies and Systems (WITS'06), 2006.

14. 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.
15. N. Bobroff, A. Kochut, K. A. Beaty, "Dynamic placement of virtual machines for managing sla violations", in Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management (IM'07), 2007.
16. NFS setup in Ubuntu, https://help.ubuntu.com/community/NFSv4Howto.
17. NIST Definition of Cloud Computing v15, www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf.
18. RRD Tool, http://oss.octiker.ch/rrdtool/.
19. S. Mehta, A. Neogi, "ReCon: A Tool to Recommend Dynamic Server Consolidation in Multi-Cluster Data Centers", in Proceedings of the IEEE Network Operations and Management Symposium (NOMS'08), Salvador, Bahia, 2008.
20. Susheel Thakur, Arvind Kalia and Jawahar Thakur, "Server Consolidation Algorithms for Cloud Computing Environment: A Review", International Journal of Advanced Research in Computer Science and Software Engineering, vol 3(9), September 2013.
21. 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.
22. 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.

AUTHOR(S) PROFILE

Susheel Thakur received his B.Tech (I.T) Degree from University Institute of Information Technology, Himachal Pradesh University, Shimla, India and pursuing M.Tech (CS) Degree from Department of Computer Science, Himachal Pradesh University, Shimla, India. His field of Interest is Cloud Computing, Database Management System and Software Engineering.

Arvind Kalia is currently working as Professor in Department of Computer Science, Himachal Pradesh University, Shimla (H.P) India. He is having 25 years of teaching and research experience. His area of interest includes Software Engineering, Computer Networks, Data mining and Cloud Computing. He is a member of different science bodies like CSI, Indian Science Congress, etc.

Jawahar Thakur is currently working as Associate Professor in Department of Computer Science, Himachal Pradesh University, Shimla, India. He has received his B.Tech (CSE) Degree from NIT Hamirpur, India and M.Tech (CSE) from NITTTR, Panjab University Chandigarh. He is currently pursuing PhD in Computer Networking.