International Journal of Engineering Research and Development
e-ISSN: 2278-067X, **p**-ISSN: 2278-800X, www.ijerd.com
Volume 6, Issue 4 (March 2013), PP. 40-46

A Survey on some of the novel techniques used for VM Scheduling and VM Management

Ms. Pallavi R
Department of Computer Science and Engineering, Sir M Visvesvarya Institute of Technology, Yelahanka, Bangalore, Karnataka, India

Abstract:- Cloud computing is a field where we access the big data, over the internet. The big data is placed over multiple servers in the cluster, though the credentials of the network connections are entirely hidden and users cannot access anyone else's data. As the big data created by the applications expand, there exists a growing demand for high performance computing infrastructures. This has led to the establishment of large-scale computing data centers causing a notable increase of electrical power consumed by hardware facilities and cooling systems. High power consumption by the infrastructure affects to substantial carbon dioxide emissions on the global scale contributing to the green house effect. Cloud shares a lot of resources with its users. If these resources can be managed efficiently, then we can view a better data center which is economical and energy efficient. It is seen that VM scheduling and VM management are the two key fields where a drastic change in energy consumption can bring in with various algorithms. Our paper gives an overview of various researches wrt the VM scheduling and VM management.

Keywords:- Cloud computing, Energy efficiency, Resource Management, VM scheduling, VM migrations, VM management

I. INTRODUCTION

Virtualization finds its key importance in the field of computing. Virtualization means creating a virtual version of something, such as a hardware platform, an operating system, a storage device or network resources. This illusion is created through the *virtual machines* (VM). VM don't exist in reality unlike the physical computer, which is clearly a complete and actual machine, both subjectively (from the user's point of view) and objectively (from the admin point of view). Where as a VM is subjectively a complete machine, but objectively merely a set of files and running programs on an actual, physical machine. For example, a computer that is running Microsoft Windows may host a virtual machine that looks like a computer with the Ubuntu Linux operating system. Even something as simple as partitioning a hard drive is considered virtualization because, you take one drive and partition it to create two separate hard drives.

Cloud computing is the virtualization of computer programs through an internet connection rather than installing applications on every office computer. Cloud computing is an updated version of utility computing: basically virtual servers available over the Internet (cloud). The big data [2] is placed over multiple servers in the cluster, though the credentials of the network connections are entirely hidden and users cannot access anyone else's data. As the big data created by the applications expand, there exists a growing demand for high performance computing infrastructures. This has led to the establishment of large-scale computing data centers causing a notable increase of electrical power consumed by hardware facilities and cooling systems. High power consumption by the infrastructure affects to substantial carbon dioxide emissions on the global scale contributing to the problem of global warming.

It has been estimated by Gartner that in 2006 the energy consumed by IT infrastructures in USA was about 61 billion kWh, corresponding to 1.5% of all the produced electricity and 2% of the global carbon emissions, which is equal to the aviation industry, and that these figures are expected to quadruple by 2020. A major reason for this huge amount of consumed power is the inefficiency of data centers, which are often underutilized: only 20-30% of the total server capacity is used on average. Despite the adoption of techniques that try to scale the energy consumption with respect to the actual utilization of a computer (for example, Dynamic Voltage and Frequency Scaling or DVFS [6] [7] which was first implemented on portable and laptop systems), an idle server still consumes approximately 65-70% of the power consumed when it is fully utilized.
The problem of excessive energy consumption can be tackled by *consolidating* the Virtual Machines on as few servers as possible. Unfortunately, the problem of optimally mapping Virtual Machines to servers, which is analogous to the classical *bin packing problem*, is known to be NP-hard [12], that is, it requires an exponential

time to be solved. Therefore current approaches can only lead to sub-optimal solutions, and often require a large number of live migrations of Virtual Machines, which can cause severe performance degradation.

According to a recent report from Pike Research [13], Cloud computing is set to reduce data center energy consumption of 31% from 2010 to 2020. The report indicates that companies have recognized the significant energy-efficiency benefits of cloud computing and its growth in the market will have important implications for both energy consumption and greenhouse gas (GHG) emissions.

This paper is organized as follows: Section II compares the various researches carried out in the field of energy management of resources. It includes mainly two features: VM scheduling and VM management. Section III concludes with the various research angles in the given field with references included.

II. ENERGY AWARE MANAGEMENT OF RESOURCES

Cloud shares a lot of resources with its users. If these resources can be managed efficiently, then we can view a better data center which is economical and energy efficient. Early work in energy aware resource management is devoted to mobile devices with the objective of improving battery lifetime [8] [9]. Later on, the context has been shifted to data centers [10] [11] and virtual computing environments such as Clouds. The two wide areas of research in the field of resource management with respect to the energy efficiency are: the VM scheduling and the VM management.

A. VM SCHEDULING

Many VMs can be assigned to a particular PM (Physical Machine) depending on their infrastructure. Cloud manages several such PMs. Scheduling addresses the placement of VMs within the cloud infrastructure while minimizing the operating costs of the cloud itself. Different system resources are taken into consideration for VM scheduling, like CPU, memory, storage, network bandwidth etc. The objective is to create an efficient scheduling system that minimizes power consumption and maximizes the performance (matching the SLA). VM scheduling can be broadly classified into two: *Thermal aware* and *Power aware*. Thermal aware: Insufficient or malfunctioning cooling system can lead to overheating of the resources reducing system reliability & device lifetime. Power aware: High power consumption by the infrastructure leads to substantial CO_2 emissions contributing to the greenhouse effect. We shall focus on Power aware VM scheduling for further discussions. Several novel techniques have been found in this field. Our paper brings a survey on these techniques. An algorithm is said to be a complete energy efficient if it passes the **C**ost **R**eliability continuous **A**vailability **P**erformance (**CRAP**) test. If we can schedule the VMs such that they reduce the cost, minimize the power, are reliable with continuous availability and give us the maximum performance, then we call such an algorithm as the energy efficient technique. Table 1 classifies the different techniques based on their CRAP test results.

Chaisiri et al [14] proposed an algorithm called Optimal Virtual Machine Placement (OVMP). Here, Cloud providers offer two payment plans: the Reservation plans which works out cheaper and the On-demand plans. This algorithm can minimize the cost spending in each plan for hosting VM in a multiple cloud provider environment under future demand and price uncertainty. It makes a decision based on the optimal solution of stochastic inter programming (SIP) to rent resources from cloud providers. With a slight deviation Ching Chuen Teck Mark et al [15] proposed Evolutionary OVMP (EOVMP) with a demand forecaster. In this, first a demand forecaster predicts the computing demand. Then, EOVMP uses this prediction demand to allocate the VM using reservation & on-demand plans for job processing. The performance or solution is close to optimal solution of SIP and the prediction of the demand forecaster is of reasonable accuracy. Hallett S et al [16] proposed where an optimized solution for data access is identified which places the VM on a physical node with the smallest data transfer time (using round trip time) to the required data. This paper explores the issues surrounding the optimal placement of data and associated processing algorithms in large scale on demand distributed infrastructures. In addition to critical network considerations such as bandwidth, parallelization, co-location, etc, considerations about node performance, cost, storage, operating systems, control middleware, processors, and task interdependencies also need to be taken into account. Where data transfers involve very large files network performance considerations will ultimately determine the resource allocation. Umesh Bellur et al [17] proposed two algorithms viz., Linear programming and Quadratic programming techniques for optimal placement of virtual machines. In this paper, an optimal technique to map virtual machines to physical machines (nodes) such that the number of required nodes is minimized. The two techniques significantly improve over the existing theoretical bounds and efficiently solve the problem of virtual machine (VM) placement in data centers. Wubin Li et al [18] in his paper uses combinatorial optimization techniques for VM placement in cloud environment. He also stated that VMs may be placed/ replicated/ migrated across multiple (at least two) geographical zones. Tang et al 19] proposed an algorithm that can produce within 30sec high-quality solution for hard placement problems with thousands of machines & thousands of VMs. Hermenier et al [20] presented the entropy resource manager for homogenous clusters, which performs dynamic consolidation based on constraint programming & takes [migration overhead into account. Nicola Maria et al

Researchers	Algorithms	CRAP test
Chaisiri et al	OVMP across multiple CP	Cost, availability, performance
Ching Chuen Teck Mark et al	EOVMP with a demand forecaster	Cost, reliability, availability, performance and accuracy
Hallett et al	n/w aware using data transfer time	Cost and locality dependant
Umesh et al	Linear and Quadratic programming	Cost, performance
Wubin et al	Combinatorial optimization techniques	Cost, reliability, availability, performance
Tang et al	High quality scheduling	Cost, reliability, availability, performance
Hermenier et al	Entropy resource manager	Cost, availability
Nicola Maria et al	Backward Speculative Placement	Cost, availability
Deepal Jayasinghe et al	SCAVP	Cost, availability, performance
Cardosa et al	Power efficient allocation of VMs	Does not support strict SLA
Verma et al	Continuos optimization	Cost, reliability, availability, performance
Anton Beloglazov et al	Online deterministic algorithm	Cost, reliability, availability, performance

Table 1

[21] stated that the majority of the existing works ignore the dynamic nature of the incoming stream of VM deployment requests that continuously arrive to the cloud provider infrastructure. They presented a novel technique called Backward Speculative Placement (BSP) that projects the past demand behavior of a VM to a candidate target host. Deepal Jayasinghe et al [22] proposed the Structural Constraint-Aware VM placement (SCAVP) based on three types of constraints: Demand, Communication, and Availability. The algorithm uses hierarchical placement approach. Cardosa et al [23] explored the problem of power efficient allocation of VMs in virtualized heterogeneous computing environment. They have leveraged “min”, ”max”, ”shares” parameters of Virtual Machine Monitor (VMM) relating to CPU allocated to VMs sharing the same resources. This only suits the enterprise environment or private clouds as it does not support strict SLA & requires knowledge of applications priorities to define share parameters. Verma et al [24] formulated the continuous optimization at each time frame. The placements of VMs are optimized to min power and max performance. A recent research conducted by Anton Beloglazov et al [25] proposed the development of the following policies for continuous optimization of VM placement: Optimization over multiple system resources utilization like CPU, RAM & network bandwidth, Network optimization – optimization of virtual n/w topologies created by intercommunicating VMs. n/w communication between VMs should be observed and considered in reallocation decisions in order to reduce data transfer overhead & network device load, Thermal optimization – current temperatures of physical nodes is considered in reallocation decisions. The aim is to avoid the “hot spots” by reducing workload of the overheated nodes & thus decrease error-proneness & cooling system load.

B. VM MANAGEMENT

Once the VMs are allocated & scheduled, frequent management of these VMs are required to control and manipulate the size and placement of VM images in various ways to conserve power and remove unnecessary bloats. Idle physical machines can be dynamically shutdown and restarted to conserve energy during low load situations. VMs are managed through *live migrations* features. Live migrations are presently used for proactive fault tolerance by seamlessly moving VMs away from failing H/W to stable H/W without the user noticing a change in a virtualized environment. Live migrations can be divided into three categories: *Pure stop & copy*, *Precopy* and *Postcopy* migrations. Pure stop & copy based migrations suspend the VM (to be migrated), transferring its entire memory contents & architectural state to another physical host, and then re-instantiating it there. Precopy based migrations transfers the memory contents to the destination host while the VM continues to execute on the source host. This is commonly an iterative process. Due to iterative scanning, tracking and transfer of VM pages consumes additional CPU and n/w resources. Therefore live migration can potentially degrade the performance of the VM that is being migrated, as well as the hosts and n/w involved in the migration. This condition which is called as “brownout” makes it desirable to minimize the time a VM spends in live migration. Postcopy based migrations defer memory transfer after the VM is resumed at the destination, and the memory pages can be retrieved on demand based on the post resume behavior of the VM. Live migrations may increase the number of VMs in transit at any given time, thereby increasing the burden on the system infrastructure. Therefore the gains from agile resource management may diminish as the resource management overheads begin to dominate. Thus it is beneficial to improve the VM migration efficiency either by improving the mechanics of VM migration or by improving the physical server or n/w configurations. The different factors affecting the migration are: VMs themselves, the migration implementation, Hypervisor options and the virtualized infrastructure characteristics such as the servers and the n/w configuration. Few research observations are found below:

Petter et al [26] described the short comings of certain VM migrations – service interruption when migrating VMs with high workloads and or over low – bandwidth network. In this paper we study the application of delta compression during the transfer of memory pages in order to increase migration throughput and thus reduce the downtime. The delta compression live migration algorithm is implemented as a modification to the KVM hypervisor. Van and Tran et al [27] proposed two levels of mappings which must be managed during the VM migrations. The provisioning stage is responsible for allocating resource capacity in the form of VM to application. This stage is driven by performance goals associated with the business level SLA's of the hosted application (e.g. average response time, number of jobs completed/ unit time). VM must then be mapped to physical machine. The author separates the VM provisioning stage from VM placement stage within the global decision layer automatic loop and formulates both the problems as Constraint Satisfaction Problem (CSP), which is the instances of an NP – hard knapsack problem for which a constraint programming seems to be a good fit. VM packing CSP produces the VM placement vectors which are used to place VMs on PMs. Wood et al [28] proposed two main approaches for dynamically mapping VMs on PMs. Black box approach which takes into the system metrics only and the Grey box approach which takes into account application level metrics along with a queuing model. VM packing is performed through a heuristic which iteratively places the higher loaded VM on the least-loaded PM. Ishakian et al [29] suggests more efficient utilization of an instance by appropriating co-locating applications from multiple cloud customers on the same instance. VM co-location could be done in the following ways: It could be offered as a (distinguishing) feature by the CSP, It could be used in a peer-to-peer fashion to allow cloud customers to form coalitions that benefit from co-location. The constraints in the above works are always considered as CPU & memory resources or jobs operations/ sec. Zhao et al [30] seeks to provide a model that can characterize the VM migration process and predicts its performance, based on a Comprehensive experimental analysis. The results show that, given a certain VM's migration time, it is feasible to predict the time for a VM with other configurations, as well as the time for migrating a number of VMs. The paper also shows that migration of VMs in parallel results in shorter aggregate migration time, but with higher per-VM migration latencies. Huang, Gao, Liu et al [31] have worked on the impact of I/O on VM migration time. Currently most VM environments use the socket interface and the TCP/ IP protocol to transfer VM migration traffic. In this paper, the author proposes a high performance VM migration design by using RDMA (Remote Direct Memory Access). RDMA allows direct data placement of data from one nodes memory space into another. This is attained without memory copies on the local side and with no involvement of the remote CPU. This application of RMDA allows for very low migration latencies with minimal consumption of compute resources devoted to the migration task itself. Another technique is SRIOV (Single Root IO Virtualization) technologies with high performance. IO technologies can enable higher application level performance (jobs/ sec, response times) when they are running in VMs while reducing hypervisor overheads and performance interface. Monica Gahlawat, Priyanka Sharma [32] proposed an algorithm to reduce the cost of virtual machine migration in federated cloud environment using component based VM. VM is not considered as a monolithic image but as a collection of various components like kernel, OS, programs and user data. Timothy Wood et al [33] propose a smart stop and copy mechanism to optimize WAN VM migration. The reality of resources being distributed across significant geographic distances and interconnected via static WAN conspire to make the realization of this vision difficult. The challenges need to be addressed are: minimize downtime; minimize n/w reconfigurations, handling WAN links. Jeongseob Ahn et al [34] propose and evaluate 2 cluster level VM scheduling techniques for: Cache sharing and NUMA affinity. Along with CPU and memory, micro-architectural resources such as shared caches, memory controllers and NUMA affinity have only

AUTHOR	**ALGORITHM**	**LIVE MIGRATIONS**		
		stop & copy	**precopy**	**postcopy**
Petter et al	Delta compression transfer of memory pages	√	√	√
Van & Tran et al	Constraint Satisfaction Problem	X	X	X
Wood et al	Black box & Grey box approach	X	X	X
Ishakian et al	Co-locating applications from multiple cloud customers on the same instance	X	X	X
Zhao et al	Comprehensive experimental analysis	√	√	√
Huang, Liu et al	Remote Direct Memory Access	√	√	√
Monica & Priyanka	Component based VM	√	√	√

Timothy wood et al	WAN based VM migration	√	X	X
Jeongseob Ahn et al	2 cluster level VM scheduling	√	√	√
Hong Xu et al	Egalitarian stable matching	√	√	√
Clark et al	Migrating live VMs between LAN nodes	X	X	√
Lagar-Cavilla et al	Cloning live VMs in LAN connected platforms	X	X	X
Sapuntzalis et al	VM live migrations that includes disk images	X	X	√
Hirofuchi et al	Transferring the VM disk image over long-haul n/w	X	√	√
Brandford et al	Transferring the VM disk image	X	√	√

Table 2

relied on intra-system scheduling to reduce contentions on them. The algorithm proposes carefully scheduling of threads (contention-aware scheduling). The technique groups applications to share a cache to minimize the overall cache misses for a system. This paper uses live VM migration to dynamically schedule VMs for minimizing the contention on shared caches and memory controllers. This study also considers the effect of non-uniform memory access (NUMA) in multi socket systems commonly used in cloud server. Hong Xu et al [35] proposed the Egalitarian stable matching for VM migration in cloud computing. Egalitarian approach finds a stable matching fair to both VMs and servers, instead of favoring either side as a result of the deferred acceptance procedure. Servers differ in their traffic loads, affecting the performance of migration. VMs also differ in the volume of disk images to be migrated, incurring different amounts of transmission overhead to the n/w – a critical factor that server needs to consider. Egalitarian stable matching – tries to find the matching that minimizes the total rank sum of the outcome among all stable matching and it applies a polynomial time algorithm to find such egalitarian stable matching. Clark et al [36] build a live VM migration tool capable of migrating live VMs between LAN connected nodes. The mechanism assumes that the source and destination nodes share an n/w-accessed storage system that maintains the VMs persistent image and only proposes a solution for migrating the in-memory state of the live VM. Lagar-Cavilla et al [37] propose a mechanism for cloning live VMs in LAN connected platforms. The proposed mechanism also assumes a shared copy-on-write storage system storing the VM images. Sapuntzalis et al [38] present a system for VM live migration that includes disk images. The approach proposes using copy-on-write images to deliver users custom images from a root image. The approach reduces the amount of data transferred using two optimizations: Exploiting similarities between the transferred image and the root image, the possible similarities with images already present at the destination. Hirofuchi et al [39] present a mechanism that complements the live migration of VM memory with transferring the VM disk image over long-haul n/w. After completing the migration of a live VM state stored in memory and starting the VM at the destination site, the approach transfers the VM disk image to the destination, giving priority to the blocks accessed by the VM after migration. Brandford et al [40] propose transferring the VM disk image at the same time as transferring the live VMs in-memory state. The VM keeps running at the source machine while the transfer of the VM memory and disk image takes place. Table 2 classifies the different algorithms based on: stop & copy, precopy and postcopy based live VM migrations.

III. CONCLUSION

In this paper we have attempted to discuss the various research angles in the field of VM scheduling and VM management, which are the two major fields in energy efficient resource management. There have been bodies of researches in data intensive computing systems that take either heterogeneity or fairness into account, but not both. As an initial step towards research in "Energy Efficient Resource Management in Cloud Computing", this paper brings a small survey on the various work done.

REFERENCES

[1] http://blog.eukhost.com/webhosting/cloud-computing-vs-grid-computing/

[2] http://en.wikipedia.org/wiki/Big_data

[3] Buyya, R., Yeo, C., Venugopal, S., Broberg, J., Brandic, I., "Cloud Computing and Emerging IT Platforms: Vision, Hype and Reality for Delivering Computing as the 5th Utility", in *Future Generation Computer Systems* 25 (2009), 2009, pp. 5999-616.

[4] Gens, Frank, "IT Model in the Cloud Computing Era", in *IDC Enterprise Panel*, August 2008.

[5] I. Foster, C. Kesselman, S. Tuecke, "The Anatomy of the GRID: Enabling Scalable Virtual Organizations", in Int'l Journal of Supercomputer Applications, 15(3), 2001

[6] C.-H. Hsu and W. Chun Feng, “A Feasibility Analysis of Power Awareness in Commodity-Based High-Performance Clusters”, *in Cluster*, 2005, pp. 1-10
[7] C. Hsu and W. Feng, “A power-aware run-time system for high performance computing”, in Proceedings of the 2005 ACM/ IEEE conference on Supercomputing. IEEE Computer Society Washington, DC, USA, 2005.
[8] R. Neugebauer and D. McAuley, “Energy is just another resource: Energy accounting and energy pricing in the nemesis OS”, in Proceedings of the 8th IEEE Workshop on Hot Topics in Operating Systems, 2001, pp. 59-64
[9] H. Zeng, C. S. Ellis, A. R. Lebeck, and A. Vahdat, “ECOSystem: managing energy as a first class operating system resource”, ACM SIGPLAN Notices, vol. 37, no. 10, p. 132, 2002
[10] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath, “Load balancing and unbalancing for power and performance in cluster-based systems”, in Workshop on Compilers and Operating Systems for Low Power, 2001, pp. 182-195
[11] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle, “Managing energy and server resources in hosting centers”, in Proceedings of the 18th ACM symposium on Operating systems principles. ACM New York, NY, USA, 2001, pp. 103-116
[12] http://en.wikipedia.org/wiki/Bin_packing_problem
[13] http://www.pikeresearch.com/newsroom/cloud-computing-could-cut-data-center-energy-consumption-by-nearly-one-third-by-2020
[14] Chaisiri S, “Optimal Virtual Machine Placement across multiple cloud providers”, in Services Computing Conference, 2009, APSCC 2009, IEEE Asia-Pacific, pp 103-110
[15] Ching Cheun Teck Mark, Dusit Niyato, Tham Chen-Khong, “Evolutionary Optimal Virtual Machine Placement and demand forecaster for cloud computing”, in AINA’ 11 proceedings of the 2011 IEEE International Conference on Advanced Information Networking & Applications, pp 348-355
[16] Hallett, Shane, Parr, Gerard and McClean, Sally (2011), “Network Aware Cloud Computing for Data and Virtual Machine Placement” in 12th Annual PostGraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, Liverpool, United Kingdom. PGNet. 6 pp
[17] Umesh Bellur, Chetan S Rao, Madhu Kumar SD (2010), “Optimal Placement Algorithms for Virtual Machines” in Distributed, Parallel and Cluster Computing, ACM classes: C.2.4; G.1.6
[18] Wubin Li (2012), “Virtual Machine Placement in Cloud Environments”, www8.cs.umu.se/research/uminf/reports/2012/013/part1.pdf
[19] Chunqiang Tang, Malgorzata Steinder, Michael Spreitzer, and Giovanni Pacifici, “Scalable Application Placement Controller for Enterprise Data Centers” in International World Wide Web Conference Committee (IW3C2), ACM 9781595936547/07/0005.
[20] Fabien Hermenier, Xavier Lorca, Jean-Marc Menaud, Gilles Muller, Julia Lawall, “Entropy: a consolidation manager for clusters”, in VEE '09 Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments Pages 41-50
[21] Calcavecchia, Nicolo Maria Biran, Ofer; Hadad, Erez; Moatti, Yosef, “VM Placement Strategies for Cloud Scenarios”, in Cloud Computing (CLOUD), 2012 IEEE 5th International Conference, pp 852-859
[22] Deepal Jayasinge, Calton Pu, Tamar Eilam, “Improving Performance & Availability of Services Hosted on IaaS Clouds with Structural Constraint-Aware VM placement”, in 2011 IEEE International Conference on Services Computing, ISBN: 978-0-7695-4462-5
[23] M. Cardosa, M. Korupolu, and A. Singh, “Shares and utilities based power consolidation in virtualized server environments,” in Proceedings of IFIP/IEEE Integrated Network Management (IM), 2009.
[24] A. Verma, P. Ahuja, and A. Neogi, “pMapper: power and migration cost aware application placement in virtualized systems,” in Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware. Springer-Verlag New York, Inc., 2008, pp. 243–264.
[25] Anton Beloglazov, Rajkumar Buyya, “Energy Efficient Resource Management in Virtualized Cloud Data Centers” (2010), in Proceedings of the 10th IEEE/ ACM International Conference on Cluster, Cloud and Grid Computing, pp. 826-831
[26] Petter Svard, Benoit Hudzia, Johan Tordsson, Erik Elmroth, “Evaluation of Delta Compression Techniques for Efficient Live Migrations of Large Virtual Machines”, in VEE’ 11 Proceedings of the 7th ACM SIGPLAN/ SIGOPS international conference on Virtual Execution Environments, pp 111-120
[27] SLA-aware virtual resource management for cloud infrastructures, Hien Nguyen Van, Fr´ed´eric Dang Tran, Orange Labs

[28] T.Wood, P. Shenoy, A. Venkataramani, and M. Yousif. Black-box and Gray-box Strategies for Virtual Machine Migration. 4th USENIX Symposium on Networked Systems Design and Implementation, 2007.
[29] Co-location as a Service: Strategic and Operational Services for Cloud Co-location, Vatche Ishakian, Raymond Sweha, Jorge Londo˜no, Azer Bestavros, Computer Science Dept, Boston University, USA
[30] Experimental Study of Virtual Machine Migration in Support of Reservation of Cluster Resources, Ming Zhao Renato J. Figueiredo, Advanced Computing and Information Systems Laboratory (ACIS) Electrical and Computer Engineering, University of Florida
[31] High Performance Virtual Machine Migration with RDMA over Modern Interconnects, Wei Huang, Qi Gao, Jiuxing Liu, Dhabaleswar K. Panda, Computer Science and Engineering, The Ohio State University and IBM T. J. Watson Research Center
[32] Monica Gahlawat, Priyanka Sharma, "Reducing the cost of Virtual Machine Migration in Federated Cloud Environment using Component based VM ", in Journal of Information Systems and Communication, ISSN: 0976-8742 & E-ISSN: 0976-8750, vol 3, issue 1, 2012, pp. 288-290
[33] Timothy Wood, Prashant Shenoy, "CloudNet: Dynamic Pooling of Cloud Resources by Live WAN Migration of Virtual Machines", in the Proceedings of VEE'11, Newport Beach, California, USA.
[34] Jeongseob Ahn, Changdae Kim, and Jaeung Han, "Dynamic Virtual Machine Scheduling in Clouds for Architectural Shared Resources", in Proceedings of HotCloud '12
[35] Hong Xu, "Egalitarian stable matching for VM migration in Cloud Computing", in Computer Communications Workshops, 2011 IEEE conference, pp. 631-636
[36] C. Clark, K. Fraser, Steven Hand, et al. Live Migration of Virtual Machines. NSDI 2005.
[37] H. A. Lagar-Cavilla, J. Whitney, A. Scannell, P. Patchin, et al. "SnowFlock: Rapid Virtual Machine Cloning for Cloud Computing" in European Conference on Computer Systems (Eurosys). 2009.
[38] C. P. Sapuntzakis, R. Chandra, B. Pfaff, J. Chow, et al. "Optimizing the migration of virtual computers". OSDI. 2002.
[39] T. Hirofuchi, H. Ogawa, H. Nakada, et al. "A Live Storage Migration Mechanism over WAN for Relocatable Virtual Machine Services on Clouds", in International Symposium on Cluster Computing and the Grid (CCGrid). 2009.
[40] R. Bradford, E. Kotsovinos, A. Feldmann, and H. Schioberg, "Live wide-area migration of virtual machines including local persistent state", in International conference on Virtual Execution Environments (VEE). 2007.