Live VM Migration Across Cloud Data Centers

Suhib Bani Melhem*, Anjali Agarwal*, Mustafa Daraghmeh*, Nishith Goel† and Marzia Zaman†
*Department of Electrical and Computer Engineering
Concordia University, Montreal, Canada
Emails: {s_banim, aagarwal} @encs.concordia.ca, Mustafa.daraghmeh@gmail.com
†Cistech Limited, Ottawa, Canada
Emails: nishith@cistech.ca, Marzia@cistel.com

Abstract— **Live VM migration is a technique that consists of a selection process and a migration process to migrate a VM from one host to another in the same data center without changing the IP address, or in a different data center with necessity to the VM to get a new IP address. The changing of IP address results into a mobility problem, which may render the service unreachable. In this paper, we propose a system model that selects data center randomly for VM placement while reducing this IP address reconfiguration. A new metric is proposed to indicate number of users that need IP reconfiguration. We extended CloudSim to simulate our work to identify the number of IP reconfigurations required for VM migration across the data centers on random workload.**

Keywords— Live Migration; WAN migration; IP Reconfiguration; Data center; CloudSim.

I. Introduction

The propagation of cloud computing has resulted in the establishment of large-scale data centers across the world, consisting of hundreds of thousands, even millions of servers. The emerging cloud computing paradigm provides administrators and IT organizations with considerable freedom to dynamically migrate virtualized computing services among physical servers in cloud data centers.

Normally, these data centers incur very high investment and operating costs for the computing and network devices as well as for the energy consumption. Virtualization and live VM migration offers significant benefits such as load balancing, server consolidation, online maintenance and proactive fault tolerance along data centers [1]. Live VM migration can be divided into two parts: 1) selection process that targets deciding to migrate a VM from a source host to a destination host to avoid SLA violation or to reduce power consumption, 2) migration process that targets moving the VM in minimum time to avoid any interruption of services.

As a result, resource management is becoming a crucial issue to emphasize on optimal resource utilization, maximum throughput, maximum response time, enhancing scalability, avoiding over-provisioning of resources and prevention of overload to make cloud computing successful.

In cloud data center management, many techniques have been proposed over the last years to solve selection process

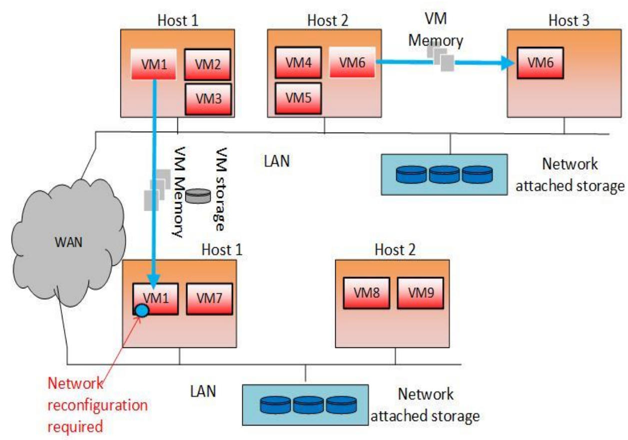

Fig. 1. VMs migration over LAN/WAN

research problem as presented in our previous work [2], but these techniques are more restricted for LAN live VM migration. In a sense, they are assuming that there is no need for IP reconfiguration during the live VM migration, and they do not take any consideration that a VM may be connected to multiple users. Thus, it does not consider the number of users during the live VM migration that may result in increasing the network overload, and an increase in service downtime of the entire system.

Figure 1 shows the migration process that migrates the entire operating system and its associated applications from one host to another that may take place either locally or widespread. VM migration over WAN differs from LAN. Firstly, LAN migration leads to transfer of memory state only, whereas WAN transfers the state of local disks as well. Secondly, network reconfiguration is an issue in the WAN migration, migrating into another subnet obliges the server to get a new IP address and, subsequently, breaks existing network connections. Therefore, WAN live VM migration results into a mobility problem, which may render the service unreachable and increase the downtime of VM during the migration process [3].

In this paper, a modified system model has been proposed to provide proactive selection process techniques that reduce network reconfiguration problem in WAN live VM migration.

2155-6814/17 $31.00 © 2017 IEEE
DOI 10.1109/MASS.2017.83

IEEE computer society

This model has been proposed to consider neglected parameters and metrics that have an effect on live migration cost. In this paper, a data center is selected randomly to receive VM selected for migration to indicate number of users that need IP reconfiguration.

This paper starts by introducing related works in Section II. Section III explains our proposed system model. The IV. Selection process approaches is discussed in detail in Section IV. Section V presents our experimental setup and performance metrics. In section VI experimental results are analyzed. Section VII shows the concluding remarks and future directions.

II. Related Work

Over the last two decades, there has been major significant research to migrate the VMs to different data centers that are located at different geographic locations (i.e. different subnet configurations) to obtain high QoS. Thus, WAN live VM migration techniques have been proposed [4-15], [23]. There are a few techniques proposed to solve IP network reconfiguration [4-6], [9], [12-15].

Bradford et al. [4] proposed a solution that depends on DNS-resolutions to do the job. When VMs migrate, they maintain their canonical names, and the new IP address is registered with the named host. Lookups for the VM based on the canonical name, following migration, will resolve to the new (correct) IP address. This seamless change in original IP address and resolution of new IP address while the VM migrates across different networks is done through IP tunneling. Tunneling is a mechanism for providing a path to networks/LANs of different IP configurations by taking help from the gateways encountered on the way to the destination network (where the designated host resides). Gateways provide tunnel endpoints, preventing any average loss of connectivity. Note that this solution places the burden of managing endpoints on the applications (i.e., they need to be aware of the IP address change).

Silvera et al. [5] proposed not to change IP address of the virtual machine while being migrated between different subnets. So, agents on the source and destination subnets are responsible for ensuring the continued connectivity of the virtual machine via the use of Proxy-ARP. IP-in-IP tunnels are used between the subnet agents to forward between subnets the traffic destined to/originating from the VM.

Wood et al. [6] proposed a combination of layer 3 virtual private networks (VPNs) and layer 2 virtual private LAN service (VPLS) to provide end-to-end routing across multiple networks and bridge LANs at different locations. The unified virtual network provides the view of a LAN to migrating VMs, with VMs maintaining single IP address.

The above methods [4-6] do not support the establishment of a new TCP connection in conjunction with VM migration, which causes increased network delay time and traffic congestion and increased performance degradation.

Kuribayashi et al. [15] proposed mSCTP, which supports multihoming and multiple IP addresses simultaneously. In mSCTP-based migration, VMs will transfer data using different TCP connections before and after migration, which causes this feature to improve response time and enhance throughput.

The existing techniques focused on applying a mobility solution or scheme to maintain the network connectivity and to preserve the open connections during and after the migration during the migration process. But no proactive criteria exist for live WAN migration that minimizes the number of the IP reconfigurations. It is known that if the time needed for IP reconfiguration for all migrated VM users increases, then there will be an increase in the interruption of service, network overhead and performance degradation.

III. Proposed System

The target system is an IaaS environment, represented by large-scale data centers. Each data center consists of ≤J heterogeneous hosts where each host contains multiple public VMs. Each VM can be connected to a number of users. Besides, each host and VM are characterized by the CPU performance metrics defined in term of Millions Instructions Per Second (MIPS), the amount of RAM and network bandwidth. The target system model is depicted in Figure 2 which is a modified version of the model described in [16]

Figure 2 shows the components of the proposed system model that provide proactive selection techniques to address IP reconfiguration issue in WAN live VM migration.

- Host Manager: a component that resides on every host for keeping continuous observation on CPU utilization of the node. It makes local decisions, such as deciding that the host is underloaded, or the host is overloaded and selecting VMs to migrate to other hosts.
- Data Center Manager: a component that resides on every data center and gathers information from hosts managers about the CPU utilization check of its host to manage the allocation of VMs locally or widespread and initiating LAN VM live migrations.
- Global Manager: a component that resides on one of the data centers and gathers information from data centers managers and makes global management decisions, such as mapping VM instances to a data center manager, and initiating WAN VM live migrations.

Host manager will check the CPU utilization status consistently for each host in its data center. If each host CPU utilization is less than a previously defined resource utilization threshold, then the system will be stable, and there is no need for LAN migration. When any host CPU overutilization is detected a VM LAN migration is triggered by data center manager between different hosts in the same data center to maintain the fairness and load balancing between the hosts. In contrast, when the data center CPU utilization is larger than a predefined CPU utilization threshold a VM WAN migration is triggered. Then the CPU utilization status of the data center

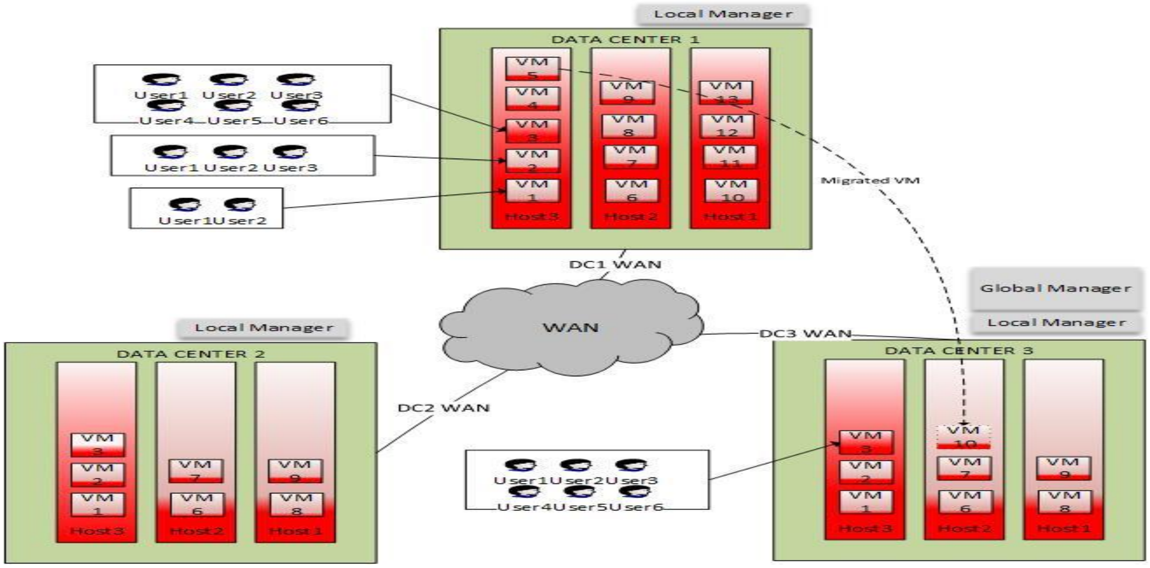

Fig. 2. System Model

should be sent to the global manager to select where to migrate the overloaded VMs based on the other data which is should be sent to the global manager to select where to migrate the overloaded VMs based on the other data which is received periodically by each data center to achieve the load balancing between the data centers as well.

IV. Selection Process algorithms

Based on the proposed system model, the selection process algorithms can be divided into four parts: (1) Host underload/overload detection, (2) Datacenter overload detection, (3) VM selection, (4) VM placement to data center, (5) VM placement to host.

The overview of the selection process algorithms is given for reference only. It is important to note that the presented algorithms are not the main focus of the current paper. The focus of the paper is the design of the system model, which is capable of identifying the number of IP reconfigurations required for VM migration across the data centers.

A. Host underload/overload detection

If a host is underutilized, then all the VMs from this host can be migrated and the host will go to sleep/shutdown mode, or the host will be considered as a good candidate to receive the migrated VMs from the overloaded hosts in the future. On the other hand, the host overload is the process of determining when a given host is overloaded so that some VMs must be selected to migrate from this host to other hosts. In our experiments, the VM host detection algorithms used are:

- Averaging threshold-based algorithm (thr) [17]: computes the mean of the n last CPU utilization values and compares it to the previously defined threshold. The algorithm detects underload state if the average of the n last CPU utilization measurements is lower than the specified threshold.
- Median Absolute Deviation (mad) [18]: specifies a lower threshold empirically, while the upper threshold is calculated using the median of the absolute deviation from the medians of the CPU usage data sets.
- InterQuartile Range (iqr) [18]: is another approach to determine the upper threshold, while the lower threshold is determined empirically as before.

B. Datacenter Overload Detection

When a data center is considered to be overloaded, one or more VM migration is required from data center under consideration. We assume the selected VMs always migrate to another data center.

C. VM Selection

Once a host overload has been detected, it is necessary to determine what VMs are the best to be migrated from the host. In our experiments, the VM selection algorithms used are:

- Maximum Correlation (MC) [19]: is inspired that high correlation between tasks and resource usage might lead to server overloading. MC uses the multiple correlation coefficient which corresponds to the squared correlation between the predicted and the actual values of the dependent variable.
- Minimum Migration Time (MMT) [19]: selects VMs based on the value of the migration time, the less the better. And the migration time can be easily computed as the amount of RAM utilized by the VM divided by the additional network bandwidth available for the current allocated host.
- Maximum Utilization (MU) [17]: Choosing the VMs to migrate from the hotspot based on the largest possible

CPU usage can be expected to minimize the number of migrations.

- Random Choice (RC) [20]: select the necessary number of VMs by picking them according to a uniformly distributed random variable.

D. VM placement to Data center

Finding a new placement of the VM selected for migration from the overloaded hosts and finding the most suitable hosts in the same data center or other data centers. In our experiment, we select a data center to receive the migrated VM randomly.

E. VM Placement to Host

Finding the most suitable hosts in the same data center or other data centers to receive the migrated VM. We used the same VM placement method as in [19]. The VM allocation algorithm selects the destination host to receive the migrated VM, which causes the least increase in the power consumption. The algorithm relies on the traditional greedy algorithm to optimize the allocation of VMs.

V. Experimental Setup

We have simulated five datacenters, each containing 50 heterogeneous physical nodes with two types: Half of the physical nodes are HP ProLiant ML110 G4 server (Xeon3040 1860 MHz, 2 cores, 4 GB) and the other half consists of HP ProLiant ML110 G5 server (Xeon 3075 2660 MHz, 2 cores, 4 GB). Each node is modeled to have two CPU cores with performance equivalent to 1860 MIPS for each core of the HP ProLiant ML110 G4 server, and 2660 MIPS for each core of the HP ProLiant ML110 G5 server. In addition, each node is modeled to have 1GB/s network bandwidth, 4GB of RAM and 50 GB of storage. The users submit requests for provisioning of 250 heterogeneous VMs, which are randomly distributed over four types similar to Amazon EC2 instance types: High-CPU Medium Instance (2500 MIPS, 0.85 GB), Extra Large Instance (2000 MIPS, 3.75 GB), Small Instance (1000 MIPS, 1.7 GB), and Micro Instance (500 MIPS, 0.633 GB). In addition, each VM requires one CPU core with 2500, 2000, 1000 or 500 MIPS, 100 Mbit/s network bandwidth and 2.5 GB of storage. Each VM is randomly connected to a maximum of 10 users.

To evaluate our system model, we have considered four metrics. Three of them are previously defined in the literature, which are SLA violation, total energy consumption, and total number of VM migrations that occur either for hotspot mitigation or for VM consolidation. In this paper, we propose a new metric, which is the Number of IP reconfiguration. All of the four metrics are precisely defined below:

- Number of IP reconfiguration: higher number of users that needs IP reconfiguration increases the network overload, and results in increased service downtime. Following equation can be used to calculate the number IP reconfiguration during a given time interval.

$$Migrations(P, t1, t2) = \sum_{j=1}^{J} \int_{t1}^{t2} WMig_j(P) \qquad (1)$$

where P represents the current placements of VM, $Mig_{VMn}(P)$ shows the number of migration of VM n between time intervals $t_1 and\ t_2$ for the placement P.

- SLA Violation: SLA contains various details of service level that will be provided to a user, such as, minimum capacities of CPU, RAM, storage, and bandwidth. In a case of SLA violation, a party that is responsible for its breach has to pay a fine to the other party. The CPU usage by a VM arbitrarily varies over time. The host is oversubscribed, i.e. if all the VMs request their maximum allowed CPU performance, the total CPU demand will exceed the capacity of the CPU. It is defined that when the request for the CPU performance exceeds the available capacity, a violation of the SLA established between the resource provider and the customer occurs. For our studies, SLA violation is calculated as shown in (2) [19]:

$$SLA\ Violations\ (SLAV) = \frac{1}{J}\sum_{x=1}^{J} \frac{T_{sx}}{T_{ax}} * \frac{1}{N}\sum_{i=1}^{N} \frac{Cd_i}{Cr_i} \qquad (2)$$

where J is number of hosts, T_{sx} the total time that utilization of host x reach to 100%, and T_{ax} is lifetime (total time that host is active) of host x. When host utilization reaches 100%, the applications performance is bounded by the host. N shows number of VMs, Cd_i estimated as 10% CPU utilization of VM_i in all migrations. Cr_i is total CPU requested by VM_i .

- Number of VM migrations: a higher number of VM migrations increases the network load, and results in performance degradation. Following equation can be used to calculate the number of migrations during a given time interval [21].

$$Migrations(P, t1, t2) = \sum_{j=1}^{J} \int_{t1}^{t2} Mig_j(P) \qquad (3)$$

where P represents the current placements of VMs, J is the number of hosts, $Mig_j(P)$ shows the number of migration of Host j between time intervals $t_1 and\ t_2$ for the placement P.

- Energy Consumption: In order to measure the power consumption of a given server at a time t with placement F, we can use following equation [16,22].

$$P_x(F, t) = k * P_{max} + (1 - k) * P_{max} * U_x(F, t) \qquad (4)$$

where P_{max} is the power consumption of the server at 100% utilization, k is the static power coefficient that is equal to the amount of power consumption by an idle processor. According to [16], an idle processor consumes 70% of the power consumed when its utilization is 100%. Therefore, in our experiments, k is set to 70%. In this model, $U_x(\mathrm{F,t})$ is the current CPU utilization of a server at time t, which has a linear relationship with the power consumption.

Total energy consumption of all the servers between time t1 and t2, can be calculated using in (5).

$$Energy(F, t_1, t_2) = \sum_{x=1}^{J} \int_{t1}^{t2} P_x(F,t) \qquad (5)$$

VI. Experimental Results

We selected four performance metrics to compare the proposed selection process scenarios, which are number of SLA violation, total energy consumption, the total number of VM migrations, and the newly proposed IP reconfiguration. We compare with VM selection algorithms presented in [17], 19], [20] including *mc*, *mmt*, *mu* and *rs* among three well-known host detection algorithms in [17], [18] including *iqr*, *mad*, and *thr*. The main goal of these experiments is identifying the number of IP reconfigurations required for VM migration across the data centers.

Figure 3 shows that *rs* VM selection algorithm outperforms the other VM selection policies *mc*, *mmt*, and *mu* in term of number of IP reconfiguration when the host detection algorithm is *iqr*. There is almost the same reduction with the *thr*. VM selection algorithms policies outperform the best when the host detection algorithm is *thr*.

Figure 4 shows that *mmt* VM selection policy outperforms the other VM selection policies *mc*, *mu*, and *rs* in term of SLA violation with all the host detection algorithms.

Figure 5 shows that *rs* VM selection algorithm outperforms the other VM selection policies *mc*, *mmt*, and *mu* in term of Number of VM migrations when the host detection algorithm is *iqr*. There is almost the same reduction with the *thr and iqr*. VM selection algorithms policies perform the best when the host detection algorithm is *thr*.

It can be seen from Figure 6 that all the proposed VM selection algorithms are similar in terms of the energy consumption.

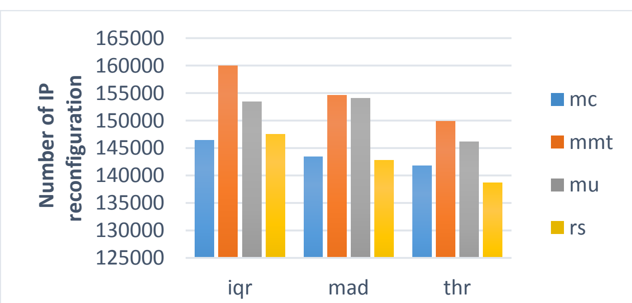

Fig. 3. Number of IP reconfiguration.

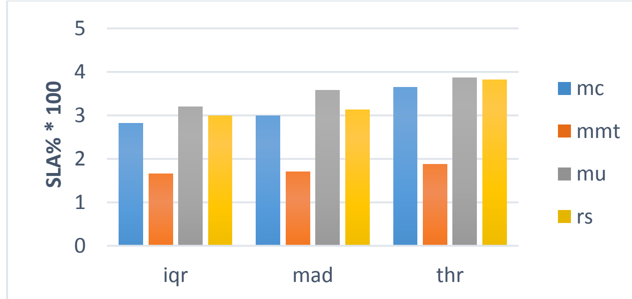

Fig. 4. SLA Violation

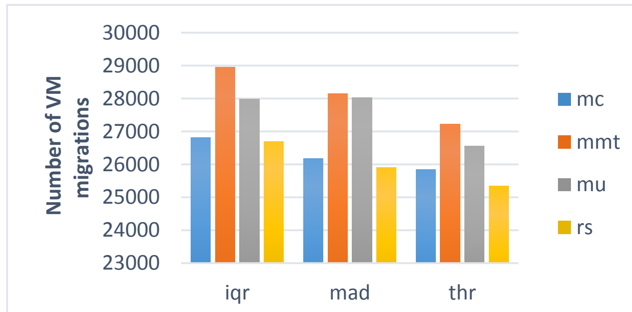

Fig. 5 Number of VM migrations

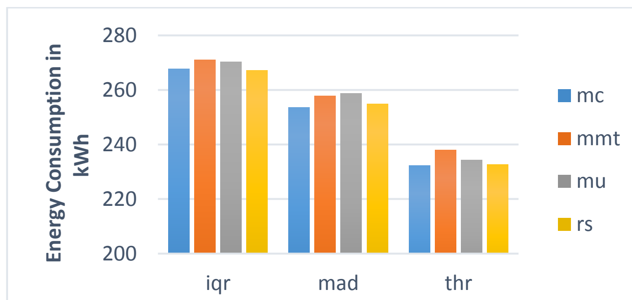

Fig. 6. Energy Consumption

VII. Conclusion

We present a modified system model to indicate the number of users that need IP reconfiguration in case of WAN migration. For the future work, we propose to find a multi-criteria algorithm to find the suitable data center for the placement of the VM selected for migration from the overloaded hosts. This criterion aims to minimize the service downtime. It is known that if the time needed for IP reconfiguration for all migrated VM users is increased, then there will be an increase in the service interruption time, network overhead and performance degradation. Furthermore, finding a VM selection algorithm aims to be a proactive solution for decreasing performance degradation by minimizing the number of IP reconfigurations that are required in case of WAN migration between the data centers.

ACKNOWLEDGEMENT

The authors would like to acknowledge the financial support provided by Natural Sciences and Engineering Research Council (NSERC) in collaboration with Cistech Limited, Canada.

REFERENCES

[1] W. Voorsluys, J. Broberg, S. Venugopal and R. Buyya. “Cost of virtual machine live migration in clouds: A performance evaluation,” In Cloud Computing, Springer, pp. 254-265, 2009.

[2] S. B. Melhem, A. Agarwal, N. Goel and M. Zaman, “Selection process approaches in live migration: A comparative study,” 8th IEEE International Conference on Information and Communication Systems (ICICS), pp. 23-28, 2017.

[3] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt and A. Warfield. “Live migration of virtual machines,” In Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation, vol. 2, pp. 273-286, 2005.

[4] R. Bradford, E. Kotsovinos, A. Feldmann and H. Schiöberg. “Live wide-area migration of virtual machines including local persistent state,” In Proceedings of the 3rd international conference on Virtual execution environments, pp. 169-179, 2007.

[5] E. Silvera, G. Sharaby, D. Lorenz and I. Shapira. “IP mobility to support live migration of virtual machines across subnets,” In Proceedings of SYSTOR 2009, The Israeli Experimental Systems Conference, pp. 1-13, 2009.

[6] T. Wood, K. Ramakrishnan, P. Shenoy and J. Van der Merwe. “CloudNet: Dynamic pooling of cloud resources by live WAN migration of virtual machines,” In ACM Sigplan Notices, vol. 46, no. 7, pp. 121-132, 2011.

[7] M. Mishra, A. Das, P. Kulkarni and A. Sahoo. “Dynamic resource management using virtual machine migrations,” Communications Magazine, IEEE, vol. 50, no. 9, pp. 34-40, 2012.

[8] T. Hirofuchi, H. Ogawa, H. Nakada, S. Itoh and S. Sekiguchi. “A live storage migration mechanism over WAN for relocatable virtual machine services on clouds,” In Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 460-465, 2009.

[9] H. Watanabe, T. Ohigashi, T. Kondo, K. Nishimura and R. Aibara. “A performance improvement method for the global live migration of virtual machine with IP mobility,” In Proceedings of the Fifth International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2010), pp. 194-199, 2010.

[10] HP White Paper. “Live migration across data centers and disaster tolerant virtualization architecture with HP StorageWorks Cluster Extension and Microsoft Hyper-V,” 2010.

[11] A. Manzalini, R. Minerva, F. Callegati, W. Cerroni and A. Campi. “Clouds of virtual machines in edge networks,” Communications Magazine, IEEE, vol. 51, no. 7, pp. 63–70, 2013.

[12] T. Wood, K. Ramakrishnan. "CloudNet: Dynamic Pooling of Cloud Resources by Live WAN Migration of Virtual Machines," In IEEE/ACM Transactions on Networking, vol. 23, no. 5, pp. 1568-1583, 2015.

[13] U. Kalim, M. K. Gardner, E. J. Brown and W. Feng. “Seamless migration of virtual machines across networks,” In Computer Communications and Networks (ICCCN), 2013 22nd International Conference, pp. 1-7, IEEE, 2013.

[14] M. V. Bicakci and T. Kunz. “TCP-freeze: Beneficial for virtual machine live migration with IP address change?” In Wireless Communications and Mobile Computing Conference (IWCMC), 8th IEEE International, pp. 136-141, 2012.

[15] S. Kuribayashi. “Improving quality of service and reducing power consumption with WAN accelerator in cloud computing environments,” International Journal of Computer Networks & Communications (IJCNC), vol. 5, no.1, pp. 41-52, 2013.

[16] A. Beloglazov and R. Buyya, “Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers,” Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science, p. 4, 2010.

[17] A. Beloglazov and R. Buyya. “OpenStack neat: A framework for dynamic and energy-efficient consolidation of virtual machines in OpenStack clouds,” In Concurrency and Computation: Practice and Experience, vol. 27, no. 5, pp. 1310-1333, 2015.

[18] A. Beloglazov and R. Buyya. “Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers,” In Concurrency and Computation: Practice and Experience, vol. 24, no. 13, pp.1397-1420, 2012.

[19] A. Beloglazov. “Energy-efficient management of virtual machines in data centers for cloud computing,” Ph.D. thesis, The University of Melbourne, 2013.

[20] A. Beloglazov and R. Buyya. “Energy efficient allocation of virtual machines in cloud data centers,” 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), pp. 577-578, 2010.

[21] S. Mustafa, B. Nazir, A. Hayat, and S. A. Madani “Resource management in cloud computing: Taxonomy, prospects, and challenges,” Computers & Electrical Engineering, vol. 47, pp. 186-203, 2015.

[22] X. Fan, W.D Weber, L.A Barroso “Power provisioning for a warehouse-sized computer,” ACM SIGARCH Computer Architecture News, vol. 35, no. 2, pp 13-23, 2007.

[23] U. Mandal, P. Chowdhury, M. Tornatore, C. U. Martel, & B. Mukherjee.. “Bandwidth provisioning for virtual machine migration in cloud: Strategy and application,” IEEE Transactions on Cloud Computing, 2016.