

Evaluation of a Cloud Management System for Live Migrations

Md Israfil Biswas* 0, Philip Morrow, Sally McClean

School of Computing, Ulster University, Coleraine, Northern Ireland, UK Email: *mi.biswas@ulster.ac.uk, pj.morrow@ulster.ac.uk, si.mcclean@ulster.ac.uk

How to cite this paper: Biswas, M.I., Morrow, P. and McClean, S. (2019) Evaluation of a Cloud Management System for Live Migrations. *Journal of Computer and Communications*, **7**, 121-137. https://doi.org/10.4236/jcc.2019.711009

Received: September 6, 2019 Accepted: November 26, 2019 Published: November 29, 2019

Copyright © 2019 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0). http://creativecommons.org/licenses/by/4.0/

C ① Open Access

Abstract

IT infrastructures have been widely deployed in datacentres by cloud service providers for Infrastructure as a Service (IaaS) with Virtual Machines (VMs). With the rapid development of cloud-based tools and techniques, IaaS is changing the current cloud infrastructure to meet the customer demand. In this paper, an efficient management model is presented and evaluated using our unique Trans-Atlantic high-speed optical fibre network connecting three datacentres located in Coleraine (Northern Ireland), Dublin (Ireland) and Halifax (Canada). Our work highlights the design and implementation of a management system that can dynamically create VMs upon request, process live migration and other services over the high-speed inter-networking Datacentres (DCs). The goal is to provide an efficient and intelligent on-demand management system for virtualization that can make decisions about the migration of VMs and get better utilisation of the network.

Keywords

Cloud Computing, Virtual Machine, Live Migration, Network Management

1. Introduction

With the incredible growth of both Cloud and wireless networks, new resource management and routing techniques have been proposed to cope with highbandwidth applications and improve the Quality of Service (QoS) [1]. Cloud computing has unprecedented advantages like on-demand resource provisioning, usage-based network access, remote resource management with ultra-resilience flexibility, universal data access with independent geographical locations and avoidance of capital expenditure on resources, etc. that consider cloud as a next generation technology [2]. Hence, this has resulted in the establishment of large-scale datacentres across the world centralising or outsourcing into the Cloud, consisting of millions of servers. One of the current cloud computing paradigms is Virtualization that provides VM migration among physical servers in cloud datacentres with benefits like load balancing, server consolidation, online maintenance and proactive fault tolerance among datacentres [3]. To extensively assist large-scale data analytics and rapid big data innovation, virtualisation also provides large scale on-demand and elastic computation and storage capabilities. Hence, the resources are partitioned into VMs through virtualisation with advantages such as isolation, consolidation and multiplexing of resources. The resources on the same physical machine in the virtual machine environments are shared by multiple VMs. An application can run in multiple VMs whereas each VM can run one or more applications.

Resource management is becoming a fundamental concern for resource optimisation with this exceptional growth in the cloud computing paradigm. Live Migration (LM) is a key feature for continuous management and maintenance of datacentres that attracts considerable interest in datacentre management and cluster computing. In cloud management, many techniques have been proposed over the last years to solve the migration strategy available for a variety of application cases concerning the issues of live VM migration. The migration problems for cloud management have been investigated but these techniques are more restricted for live VM migration. Assuming no consideration for multiple users that may be connected to VMs locally or remotely, also not considering 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 network.

This paper proposes and evaluates an efficient on-demand management model for inter-datacentre virtualization that checks CPU utilisation status for live migration then contacts a suitable host to submit a job e.g., migration request. Any VM migration is done based on a policy manager or Service Level Agreement (SLA). Our evaluation of the model found benefits showing reasonable utilisation of all hosts between datacentres through experimental results. The model also provides benefit by supporting flexible management access for remote hosts with reduced migration time and latency. In summary, the paper is unique for the following aspects:

- An on-demand virtualization architecture is designed and evaluated for inter-datacentre networks considering live migrations;
- Provides performance improvement *i.e.*, CPU utilization, SLA violation, etc using the proposed VM management system;
- Shows an efficient VM management technique using various VM loads including remote locations.

The paper is organised as follows: Section II describes cloud management systems in general, Section III illustrates our proposed Management System, which is connected to a high-speed fibre optic infrastructure; Section IV presents the experimental setup and configuration for this work; Section V evaluates our proposed system through experimentation in a developed testbed. Finally, the paper concludes with a view of future work.

2. Cloud Management

The high demand for cloud computing-based resources introduces complex datacentre architectures that require highly efficient and optimised operating and monitoring applications, which is the main goal of cloud management software design. Services residing in the cloud require efficient management and management tools help ensure cloud computing-based resources are working optimally and properly interacting with users and other services.

2.1. Cloud Management Policies

Cloud management policies are vital as cloud computing grows more complex and a wide variety of cloud-based systems and infrastructure (e.g., private, hybrid, public, etc) exist. The policies may include various aspects including security, monitoring and emergency plans. Hence, cloud management tools need to be flexible and scalable within a cloud computing strategy. Cloud management software must have the following features for users [4]:

- A common and similar understanding of virtualized resources irrespective of the virtualization platform;
- Full life cycle management of VMs: this includes dynamic network setup, storage requirements, etc.;
- Provision of resource allocation: able to configurable policies to achieve goals like high availability, server consolidation to minimize power usage, etc.;
- Resource management: acclimate to an organization's resource needs (changing resources, including addition or failure of physical resources), especially in critical hours.

2.2. Service Level Agreement (SLA)

The guarantee of the Service Level Agreement (SLA) for cloud and large-scale applications is the key to the dynamic workload of applications and sharing the resources in datacentre networks. VMs receive the resources requested by users and are then placed in different hosts based on resource utilization or SLA requirements. The cloud facilitates such resource virtualisation with flexible resource utilisation that dynamically regulates the resource allocation for a VM to support the application resource demands. Meeting the SLA requirements between cloud providers and users also enables improved resource consumption, power energy efficiency and reducing the number of VM migrations in popular places.

An SLA includes the management of various integrated process at different levels *i.e.*, SLAs from the business level to service level and network level management [5]. The SLA consists of several functions starting with SLA creation, contract, provisioning, monitoring, maintenance, notifications and assessment, which is shown in Figure 1.

Guaranteeing the SLAs of applications in the cloud is critical for VM management. The increased workload in the VMs may cause one or multiple resources



Figure 1. Illustration of cloud-based SLA architecture.

including CPU, memory, I/O and network bandwidth on the physical resources to become overloaded. However, the performance of all the VMs is often degraded due to overloading and hence increases the job completion time (*i.e.*, for batch data processing and the response time of interactive applications). Any overloading must be migrated from the overloaded physical machines to underutilised ones in order to eliminate such hotspots. VM migration is a complicated and challenging problem for resource management like CPU, memory, I/O and network for individual VMs and physical machine management.

Hence, the resource requirement of the VM has to be considered and matched with the available resources on the physical machines to decide to which physical machine a VM should be migrated. Another issue for VM management is the overhead of VM migration itself that can severely affect application performance and should be optimised. Reducing the overhead of VM migrations reduces the performance degradation caused by the migrations. Moreover, dynamic VM management is necessary, which is difficult in an environment where workload dynamically changes. Hence, it is not efficient to make the migration decision only based on the current state of the system.

3. Related Work

A number of methods have been proposed to address VM management issues in the context of the rapid growth of cloud computing. Cloud providers now are paying more attention to this by providing high quality services. Therefore, many researchers have begun to study VM management methods.

One approach is to check current resource utilisation measurements within given thresholds and decide where the VM should be migrated based on the dynamically assigned different weights to different resources. However, the problem with this method is the time delay in responding to load imbalances and inefficient management issues due to the dynamic load changes. A comprehensively surveyed based on the challenges like memory data migration; storage data migration and network connection continuity are discussed in [6], where the works on quantitative analysis of VM migration performance are also elaborated. Another method of allocating VM resources dynamically is based on resource utilisations of VMs and physical machines [7]. The problem in this approach is over-estimation or under-estimation regarding resource allocation that may cause waste of resources leading to significant SLA violations [8].

T. Maoz *et al.* proposed a VM migration approach that helps in the migration of groups of processes and parallel jobs among different clusters in a multicluster or a Grid in [9]. They evaluated their proposed technique called Jobrun in real HPC applications and presented detailed measurements of the performance.

An adaptive energy-efficient and threshold-based heuristic algorithm is suggested in [10] [11], which controls virtual machine migration by monitoring the resource utilization rate. However, threshold-based migration strategies have a problem in predicting the possible workload. Hence, this may trigger unnecessary and wasteful migrations for the host machine.

A load balancing VM migration framework based on a new metric for quantifying virtualized server load is presented in [12] based on the variation in load measured on the machines. The load balancing algorithm chooses the VM migration that achieves the greatest improvement on this imbalance metric in this framework. The migration process is considered as a multi-objective problem and a novel migration policy is proposed [13]. To evaluate different objectives simultaneously, it utilises a new elastic multi-objective optimisation strategy.

A model for intelligent decision over migration time of VMs across heterogeneous physical nodes of a cluster server is presented in [14]. The method provides an algorithm for making a multi criteria decision method to migrate VMs between cluster nodes and improve performance. To reduce the time and cost to achieve load balance, Chen *et al.* propose RIAL [15] in which different weights are dynamically assigned to different resources based on their usage intensity in the physical machines to determine the destination of the VM to migrate.

A system named Sandpiper is proposed in [16], which automates the detection of hotspots and management of VMs including live migrations. The resource utilisation on the physical machine and comparison is done by Sandpiper and a threshold is used to determine an unbalanced location. The volume, defined as the product of CPU, network and memory loads to capture the combined load on multidimensional resources, is used in this system. Volume-to-size ratio (VSR) is used to measure the volume per unit byte moved.

The live migration strategy of multiple virtual machines with different resource reservation methods is presented in [17]. Considering the communication dependencies among VMs of a multi-tier enterprise application, the underlying datacentre network topology, as well as the capacity limits of the physical servers in datacentres, an application aware virtual machine migration scheme is proposed in [18].

A parallel migration is proposed in [19] to speed up the load balancing process, which migrates multiple VMs in parallel from overloaded hosts to underutilised hosts. However, this work is unique to the following aspects:

- It proposes and evaluated an efficient approach that not only shows improve utilisation of the resources but also reduces migration time and latency, which is absent from most previous works in Cloud computing environments;
- This work evaluates SLA violation to manage customer satisfaction, which is also not found in many previous works;
- The proposed system evaluates overall networks performance, which is very important for customer satisfaction;
- This work proposes a dynamic management system that adopts resource pool e.g., LM with remote datacentres and evaluates the proposed system, while most previous work deal with fixed size of the resource pools with simulations.

4. The Proposed Approach

The proposed cloud management model for LM is a managed system that is connected to remote but inter-networking datacentres and is based on two processes: 1) VM-Request process and 2) Migration process. Service: The VM-Request allows users to submit or receive a VM creation request, specifying the details e.g., VM identification number, memory requirements, remote host to be used, etc. For a VM creation, the system has shared storage facilities within a Network File System (NFS).

In the second step, the Migration process migrates a VM to a local or remote. In the system, the users can specify a path to a directory containing files needed for the request to avoid file access through NFS or to avoid NFS enslavement. **Figure 2** illustrates the Management System and its interaction with Host VMs.



Figure 2. The proposed cloud management system overview.

The system checks all the possible hosts' CPU utilisation status for LM, contacts a suitable host and submits the LM request. Once the VM has been launched by the system, it sends the VM to the Migration process for migration and receives the confirmation of an accepted destination host. When the associated VM is migrated, the running system is also informed, so that it can track the status of the CPU during the migration. As soon as a VM is created and launched, a Policy Manager inside the management system runs the LM submitted request and track the migration status as well. The Policy Manager is able to edit or modify the migration rule or the SLA that will be performed from the source host to the targeted destination host.

The management system also able to replicate the VM image in a cache repository attached to a local disk in order to optimise the performance and reduce the overall run-time for future migrations. This is really helpful to avoid network congestion in a network although any high storage systems like the proposed model have the ability to load several VM images or associated files to the memory cache. As the developed architecture is connected to a repository service on NFS, the files can be decompressed directly to the disk when creating a new VM by eliminating redundant duplications. This also can reduce disk performance or avoid extra strain on network resources. The cached VM memory image is also used when the system instructs the Migration process to create a new VM locally and send it to a remote host prior to its execution. The required VM image can be taken directly from the memory cache and sent compressed to the remote host, where it is uncompressed directly to the disk. Moreover, the goal is to reduce the creation time by eliminating the usage of the local disk and reducing the transfer time of the files to the remote host.

During the Migration process, it can inform the management system to create a VM and migrate it to a remote host before execution. This is required as a remote host may not support the creation of VMs but may be able to host guest VMs or if some user specified files are not accessible from the remote host. The Migration process is in charge of contacting the host for the live migration and relay the request. Furthermore, it migrates VMs to other hosts by dumping the VM to the disk and sending the resulting files to the Migration process running on the remote host, when requested by the management system.

The Migration process is also responsible for managing VMs including cleaning up, deleting VMs, resuming any remote VMs or configuring network parameters, etc. The Migration process disposes of the local files upon success. If there is any error during the migration process *i.e.*, a security issue or high latency, the VM is resumed to ensure the continued execution of the migration process. The Migration process ensures the running and monitoring of a migration by configuring and obtaining the request details from the Migration running on the hosting node. During the migration it also monitors its run to relay input and output from and to the user. Moreover, it updates the host Migration to shut the VM down and deletes it from the host when the migration is completed. It may instruct the VM to bring processes to the source when the VM is about to migrate and resume these processes when the migration is completed. **Figure 3** illustrates the flowchart for a live migration process.

The System Configuration and Setup

We have developed a testbed that connects three Cable Landing Stations (CLS), designed and implemented three simulated datacentres with a dynamic management system for resource provisioning between the datacentres, specifically for live VM migrations. The cloud testbed is designed around a fibre optic ring network connecting Coleraine (Northern Ireland), Dublin (Ireland) and Halifax (Canada) CLSs [20], [21]. The proposed management system with inter-datacentre networking provides a direct high-speed fibre optic interface connecting the Ulster University at Coleraine through a fibre patch panel, where the other two datacentres in Dublin CLS and Halifax CLS are also connected with the ring network.

Figure 4 represents the architecture of the developed testbed used for the management system. The details of the management system and the datacentre configuration are given in Table 1.





Physical machines	Server	CPU (GHz)	Memory (GB)	Operating system	Network interfaces (Gbps)	Live migration
The management system	Intel (R) Core (TM) i7-4770	3.40	16	Ubuntu desktop v 16.04	1	-
Dublin-CLS	Dell PowerEdge R815	3.6	128	Ubuntu KVM 16.04	10	QEMU version 2.6.0, libvirt 1.2.2
Halifax-CLS	Dell PowerEdge R815	3.6	128	Ubuntu KVM 16.04	10	QEMU version 2.6.0, libvirt 1.2.2
Coleraine (Local)-CLS	Dell PowerEdge R815	3.6	128	Ubuntu KVM 16.04	10	QEMU version 2.6.0, libvirt 1.2.2

 Table 1. The cloud management system and datacenter configuration.



Figure 4. The architecture of the interconnected datacentre testbed.

The management system is at the Ulster University NetCom lab is connected with a Fibre Channel (FC) switch to manage the three physical machines in Coleraine, Dublin and Halifax. An ADSL connection for Internet access between the CLSs is also provided with the developed testbed. This is to provide a gateway to the outside world for secure software development and updates between the CLSs without interfering with the academic JANET network. Hence, this is not an extension of the high-speed network and the interface is only installed to operate at around 76 Mbits/sec, which is purely for server firmware updates and to comply with UK JISC rules.

The NFS Server that is configured for seamless and secure virtualisation is not required to be transferred during the migration of the disk images. The images can be accessible by the same path from hosts; therefore, the shared storage is set up and mounted on all the available hosts during the LM.

Migration only transfers in-memory state of a running domain for example memory, CPU state, etc. by default. The NFS server with its shared storage exports a directory that is mounted at a common place on all hosts. We consider that all the hosts are running in the same network *i.e.*, a LAN environment connecting remote datacentres and the DNS configuration for the consistency of associated files across all the hosts are confirmed.

libvirt [22] is an open-source API, daemon and management tool that is popular for cloud virtualization management and supports various virtualization technologies including KVM. In this work, we have used libvirt for live migration by developing a Linux based tool to manage the network and configured this by enabling associated files to make the hypervisor listen for TCP communication with authentication. For cloud security, the firewall configuration and associated files are also configured to allow libvirt to listen on a TCP port and to allow a record accepting KVM communication on the TCP port within the sync range. The SSH keys for authentication are strongly recommended as authentication is set to NONE. By default, pre-copy is enabled for live migration in Ubuntu 16.04. However, we have configured libvirt with post-copy live migration capability and used it for VM migrations in this work as our previous works show that post-copy live migration results in lower downtime.

VM host detection is very important as underutilization of a host can be used to migrate all the VMs unless the host goes to sleep or in shutdown mode. However, if any migrated VMs are over-utilized, the under-loaded host can be used. Hence, any over-utilized host can be considered as a destination host during the migration process. In this paper we have used CPU utilization based "averaging threshold-based algorithm (THR)" [23] that computes the mean of the n last CPU utilization values and compares it to the previously defined threshold. In this algorithm a threshold is specified, and an underload state is detected if the average of the n last CPU utilization measurements is lower than the specified threshold.

5. Evaluation of the System and Experiment Results

We have measured the real throughput during LM over high-speed 10Gbps fibre optical links using the iperf tool [24] and **Table 2** provides some of the test results. In this section, we evaluate the performance of the proposed management approach during live migration of VMs. The performance metric used for the experiments is the CPU utilisation (as a percentage).

CPU queue length depends on the number of processes waiting to execute in a queue and usually a load on a host at any given time was described by the queue length. However, the queue length does not reflect directly memory utilization as other system resources waiting for execution are not included. Therefore, the system statistics such as CPU utilization and memory utilization of a node changes during the process's execution *i.e.*, the CPU utilization may be high for a period of time but low in the next time interval. Therefore, in this work, we have

Datacentres	Average speed (Gbps)	Average latency (ms)
Coleraine-CLS to Dublin-CLS	7.4	5.5
Coleraine-CLS to Halifax-CLS	3.5	52.5
Coleraine (Local)-CLS	8.0	0.68

Table 2. The initial tests results between data centres.

calculated the average statistics for an overall period of time and have considered the fact that the CPU usage during any VM migrations is dependent on the activity of the VM. If the VM is very active or fully loaded, the CPU usage is higher. Hence, this work analyses the system performance to determine the impact for host CPU utilisation over the high-speed network.

The first step in the migration process is to determine and classify the load for the hosts. Considering CPU utilization, we have classified two kinds of loads, where a stress tool is used to generate a stable memory load during migration based on the threshold value. The size of stressed memory is tuned with the VMs so that the stress tool can consume the threshold values for calculation:

Lightly loaded VMs: VMs were running Ubuntu 16.04 LTS and not consuming more than 20% of the VM's memory.

Fully loaded VMs: when VMs with high memory-intensive load or, VM running in Ubuntu 16.04 LTS and consumes more than 75% of the VM's memory.

5.1. Evaluation

In this subsection, we have evaluated our proposed system comparing it against when the system is not used. Figure 5(a) shows LM between the datacentres without the proposed management system for lightly loaded VMs. We found not much difference in utilisation while the network load is less than 500 Mbps. However, for a higher network load *i.e.* 3Gbits/s, the remote host with higher latency shows up to 20% lower utilisation of its CPU compared to other low latency or local hosts (e.g., the CPU utilization during live migration between Coleraine and Halifax host shows lower utilisation compared to the utilization among local hosts in Coleraine). Figure 5(b) shows the improvement using our proposed management system that is using lightly loaded VMs during LM between datacentres. The figure clearly indicates only less than 5% difference in CPU utilization during the LM.

For experiments with fully loaded VMs, we found better results using our proposed system. LM between datacentres shows better utilization regardless of the latency. **Figure 6(a)** shows that without using the proposed system local hosts in Coleraine highly utilize their CPU during LM compared to LM between high latency hosts. For example, LM between local ColeraineVMs utilizes 40% more than LM between Coleraine and Halifax.

However, we observed better utilisation using the proposed system between datacentres as shown in **Figure 6(b)**. For example, less than 7% difference is observed in CPU utilization during live migrations between Coleraine local servers and between Coleraine to Halifax.



Figure 5. Lightly loaded VM migration (a) without the management system, (b) with the management system



Figure 6. Fully loaded VM migration (a) without the management system, (b) with the management system.

5.2. SLA Violation Metrics

Managing resources for optimised results and deals at various service levels are very important requirements where SLA plays an important role. A SLA contains the information for optimised capacities of CPU, RAM, storage and bandwidth. The service provider will be responsible for any breach of the SLA and charged to pay to the other contracted provider. For example, the CPU usage fluctuates over time and the usage could be agreed through an SLA for a host. The host has to pay a fine when it is oversubscribed (*i.e.*, maximum allowed CPU usage is requested by all the VMs of the host and hence the total CPU demand will exceed the capacity of the CPU). Therefore, the violation of the SLA is caused between the service provider and the customer while the CPU performance is exceeding the total capacity.

In order to evaluate the proposed system, we consider SLA violation metric used in [11]. This is to evaluate the level of SLA violation caused by the system defined as follows:

SLA violations (SLAv) =
$$\frac{1}{n} \sum_{h=1}^{n} \frac{Tf}{Ta} * \frac{1}{Vm} \sum_{i=1}^{Vm} \frac{Ei}{Ci}$$

Here, *n* is the number of hosts, *Tf* is the total time considering 100% utilisation of host *h* experience leading to an SLA violation. *Ta* is the total active time of the host *h* while serving the VMs. The number of VMs is denoted by *Vm*; *Ei* is the estimate of the performance degradation of the VM*i* caused by migrations and *Ei* is estimated by 10% of the CPU utilization in MIPS during all migrations of the VM*i*; *Ci* is the total CPU requested by VM*i* during its life time.

VM selection algorithms are important as 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, we have used two types of VM selection algorithms: 1) *Minimum Migration Time* (*MMT*) [10] and 2) *Random Choice* (*RC*) [23] to compare our proposed system.

Figure 5(a) shows that *mmt*VM selection outperforms *rc* in term of SLA violation with all the host detection algorithms without the proposed system. However, using the proposed system as shown in **Figure 7(b)**, the VM selection algorithms show more improvement in term of SLA violation with all the host detection algorithms.

5.3. Overall Network Performance

In this subsection, we have analysed the experimental results to evaluate the performance of the high-speed network using the proposed system with a network load of 3 Gbps considering lightly and fully loaded VMs.

Figure 8(a) shows that our proposed system can improve the overall network performance for lightly loaded VMs regardless of network latency. For example, our results found that without our proposed system live migration within Coleraine local servers can have overutilization up to 90% and less than 20% utilization during LM between Coleraine and Halifax. However, the proposed system



Figure 7. SLA violation (a) without and (b) with the proposed management system.

effectively uses the threshold that found less than 70% between local servers and more than 40% utilisation between Coleraine and Halifax.

Similar performance was observed for fully load VMs as shown in **Figure 8(b)**, where without our proposed system live migration within Coleraine local servers can have overutilization for more than 95% (with some packet drops) and less than 20% for Coleraine to Halifax VM migrations. The proposed system found less than 75% utilization during live migration between local servers in Coleraine and more than 30% utilisation for live migration between Coleraine to Halifax VMs.

In summary, the results illustrate the fact that lightly loaded VMs are more reasonably utilised for all the hosts regardless of the latency using the proposed system.



Figure 8. Overall network performance with and without the proposed system (a) lightly loaded, (b) fully loaded.

6. Conclusions

This work proposed and evaluated a cloud management framework that dynamically provides live migration to remote datacentres considering various types of load. Our results with the developed management system show improvement in CPU utilisation during VM migrations. The management model can be improved by introducing new techniques considering a multi-domain environment.

This work focused on CPU utilisation, but further development is possible considering other parameters such as CPU queue length, size of the process, dependency on host and time required for migration, etc. Future work will extend this management system by evaluating real-world applications, the resource use assumption and the trigger points or events for the live migrations.

Acknowledgements

This research is partly supported by the "Agile Cloud Service Delivery Using Integrated Photonics Networking" project funded under the US-Ireland Programme NSF (US), SFI (Ireland) and DfE (N. Ireland). The authors would also like to acknowledge support from the BT-Ireland Innovation Centre (BTIIC).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- Al-Dubai, A.Y., Zhao, L., Zomaya, A.Y. and Min, G. (2015) QoS-Aware Inter-Domain Multicast for Scalable Wireless Community Networks. *IEEE Transactions on Parallel and Distributed Systems*, 26, 3136-3148. https://doi.org/10.1109/TPDS.2014.2365190
- Biswas, M.I., Parr, G., McClean, S., Morrow, P. and Scotney, B. (2014) SLA-Based Scheduling of Applications for Geographically Secluded Clouds. 1st Workshop on Smart Cloud Networks & Systems (SCNS14), Paris, France, 3-5 December 2014, 1-8. <u>https://doi.org/10.1109/NOF.2014.7119800</u>
- [3] Yi, B., Wang, X., Das, S.K., Li, K. and Huang, M. (2018) A Comprehensive Survey of Network Function Virtualization. *Computer Networks*, 133, 212-262. <u>http://www.sciencedirect.com/science/article/pii/S1389128618300306</u> <u>https://doi.org/10.1016/j.comnet.2018.01.021</u>
- [4] Sotomayor, B., Montero, R.S., Llorente, I.M. and Foster, I. (2009) Virtual Infrastructure Management in Private and Hybrid Clouds. *IEEE Internet Computing*, 13, 14-22. <u>https://doi.org/10.1109/MIC.2009.119</u>
- [5] Lee, H.J., Kim, M.S. and Hong, J.W. (2002) Mapping between Qos Parameters and Network Performance Metrics for SLA Monitoring. Asia-Pacific Network Operations and Management Symposium (APNOMS) Conference. <u>http://dpe.postech.ac.kr/papers/APNOMS/02/sla-monitoring.pdf</u>
- [6] Zhang, F., Liu, G., Fu, X. and Yahyapour, R. (2018) A Survey on Virtual Machine Migration: Challenges, Techniques, and Open Issues. *IEEE Communications Surveys & Tutorials*, 20, 1206-1243. <u>https://doi.org/10.1109/COMST.2018.2794881</u>
- Herrera, J.G. and Botero, J.F. (2016) Resource Allocation in NFV: A Comprehensive Survey. *IEEE Transactions on Network and Service Management*, 13, 518-532. https://doi.org/10.1109/TNSM.2016.2598420
- [8] Xiao, Z., Song, W. and Chen, Q. (2013) Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment. *IEEE Transactions on Parallel and Distributed Systems*, 24, 1107-1117. https://doi.org/10.1109/TPDS.2012.283
- [9] Maoz, T., Barak, A. and Amar, L. (2008) Combining Virtual machine Migration with Process Migration for HPC on Multi-Clusters and Grids. 2008 *IEEE International Conference on Cluster Computing*, Tsukuba, Japan, 29 September-1 October 2008, 89-98. <u>https://doi.org/10.1109/CLUSTR.2008.4663759</u>
- [10] Beloglazov, A. and Buyya, R. (2015) OpenStack Neat: A Framework for Dynamic and Energy-Efficient Consolidation of Virtual Machines in OpenStack Clouds. *Concurrency and Computation: Practice and Experience*, 27, 1310-1333. <u>https://doi.org/10.1002/cpe.3314</u>

- [11] Beloglazov, A. (2013) Energy-Efficient Management of Virtual Machines in Data Centers for Cloud Computing. Ph.D. Thesis, The University of Melbourne, Melbourne.
- [12] Arzuaga, E. and Kaeli, D.R. (2010) Quantifying Load Imbalance on Virtualized Enterprise Servers. In: *Proceedings of the First Joint WOSP/SIPEW International Conference on Performance Engineering*, ACM, New York, 235-242. <u>https://doi.org/10.1145/1712605.1712641</u>
- [13] Sallam, A. and Li, K. (2014) A Multi-Objective Virtual Machine Migration Policy in Cloud Systems. *The Computer Journal*, 57, 195-204. https://doi.org/10.1093/comjnl/bxt018
- [14] Tarighi, M., Motamedi, S.A. and Sharifian, S. (2010) A New Model for Virtual Machine Migration in Virtualized Cluster Server Based on Fuzzy Decision Making. arXiv Preprint arXiv:1002.3329.
- [15] Chen, L., Shen, H. and Sapra, K. (2014) RIAL: Resource Intensity Aware Load Balancing in Clouds. *IEEE INFOCOM* 2014-*IEEE Conference on Computer Communications*, Toronto, 27 April-2 May 2014, 1294-1302. https://doi.org/10.1109/INFOCOM.2014.6848062
- [16] Wood, T., Shenoy, P., Venkataramani, A. and Yousif, M. (2009) Sandpiper: Black-Box and Gray-Box Resource Management for Virtual Machines. *Computer Networks*, 53, 2923-2938. <u>https://doi.org/10.1016/j.comnet.2009.04.014</u>
- [17] Ye, K., Jiang, X., Huang, D., Chen, J. and Wang, B. (2011) Live Migration of Multiple Virtual Machines with Resource Reservation in Cloud Computing Environments. 2011 *IEEE 4th International Conference on Cloud Computing*, Washington DC, 4-9 July 2011, 267-274. <u>https://doi.org/10.1109/CLOUD.2011.69</u>
- Shrivastava, V., Zerfos, P., Lee, K.-W., Jamjoom, H., Liu, Y.-H. and Banerjee, S. (2011) Application-Aware Virtual Machine Migration in Data Centers. 2011 *Proceedings IEEE INFOCOM*, Shanghai, 10-15 April 2011, 66-70. https://doi.org/10.1109/INFCOM.2011.5935247
- [19] Chen, K.-T., Chen, C. and Wang, P.-H. (2014) Network Aware Load Balancing via Parallel VM Migration for Data Centers. 2014 23rd International Conference on Computer Communication and Networks, Shanghai, 4-7 August 2014, 1-8. https://doi.org/10.1109/ICCCN.2014.6911748
- [20] Biswas, I., Parr, G., McClean, S., Morrow, P. and Scotney, B. (2016) An Analysis of Live Migration in Openstack using High Speed Optical Network. 2016 SAI Computing Conference, London, UK, 13-15 July 2016, 1267-1272. https://doi.org/10.1109/SAI.2016.7556142
- [21] Biswas, I., Parr, G., McClean, S., Morrow, P. and Scotney, B. (2016) A Practical Evaluation in Openstack Live Migration of VMs Using 10Gb/s Interfaces. 2016 *IEEE Symposium on Service-Oriented System Engineering*, Oxford, UK, 29 March-2 April 2016, 346-351. https://doi.org/10.1109/SOSE.2016.35
- [22] Libvirt Wiki (2019). https://wiki.libvirt.org/page/Main_Page
- [23] Beloglazov, A. and Buyya, R. (2012) Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers. *Concurrency and Computation: Practice and Experience*, 24, 1397-1420. <u>https://doi.org/10.1002/cpe.1867</u>
- [24] iPerf (2019) iPerf—The Ultimate Speed Test Tool for TCP, UDP and SCTP. https://iperf.fr/en/