

Effective Live Cloud Migration

Ibrahim Ejdayid A.Mansour
Dept. of Computing & Informatics
Bournemouth University
Bournemouth, UK
imansour@bournemouth.ac.uk

Kendra Cooper
Dept. of Computer Science
The University of Texas at Dallas
Richardson, U.S.A
kendra.m.cooper@gmail.com

Hamid Bouchachia
Dept. of Computing & Informatics
Bournemouth University
Bournemouth, UK
abouchachia@bournemouth.ac.uk

Abstract—Cloud providers offer their IaaS services based on virtualization to enable multi-tenant and isolated environments for cloud users. Currently, each provider has its own proprietary virtual machine (VM) manager, called the hypervisor. This has resulted in tight coupling of VMs to their underlying hardware hindering live migration of VMs to different providers. A number of user-centric approaches have been proposed from both academia and industry to solve this issue. However, these approaches suffer limitations in terms of performance (migration downtime), flexibility (decoupling VMs from underlying hardware) and security (secure live migration). This paper proposes LivCloud to overcome such limitations. An open-source cloud orchestrator, a developed transport protocol, overlay network and secured migration channel are crucial parts of LivCloud to achieve effective live cloud migration. Moreover, an initial evaluation of LAN live migration in nested virtualization environment and between different hypervisors has been considered to show the migration impact on network throughput, network latency and CPU utilization. The evaluation has demonstrated the need for optimization within the LAN environment.

Keywords—Virtualization; Network Virtualization; Nested Virtualization; Live Cloud Migration; Software Defined Network (SDN); Network Function Virtualization (NFV); Cloud infrastructure

I. INTRODUCTION

Live cloud migration of VMs at IaaS is an active research area, working to overcome the lack of cloud interoperability among providers. Virtualization is the foundation of the cloud IaaS. It allows cloud users to exploit multi-tenant resources (compute, network and storage) from a secure Cloud IaaS [2]. Virtualization is the conversion of a physical machine to individual isolated spaces (VMs) that can be used by multiple users as per their needs. The isolation and resources provision is provided by hypervisor [6]. Public cloud IaaS is often described as a heterogeneous environment due to the fact that each cloud provider has their own hypervisor. Providers such as Amazon EC2 and Rackspace use the hypervisor Xen; while, Fractus and Google Compute Engine rely on KVM. Windows Azure, on the other hand, uses the Microsoft hypervisor, Hyper-V [1].

Despite that many providers leverage the same hypervisors for virtualization, for example Google and HP both use KVM, live cloud migration of VMs between those providers is still challenging [1]. Every provider has been developing their own APIs and proprietary features to their selected hypervisor. This has made it difficult for cloud users to live migrate VMs to

other providers - one aspect of vendor lock-in with substantial consequences [7]. In 2013, for example, Amazon's US-EAST availability region remained unavailable for 59 minutes, resulting in users in U.S.A. and Canada who could not access Amazon.com and Audible.com. The reported loss was about \$1,100 in net sales per second [31]. If cloud users were able to utilize resources and services from various providers, then many benefits would be achieved [4][7]:

- 1) Achieving high flexibility to change service providers, thereby, alleviating vendor lock-in.
- 2) Live cloud migration can take advantage of low-price services offered by certain providers.
- 3) Offering service continuity in case of ceasing operation or natural disasters.
- 4) Reducing latency by connecting cloud users to the nearest datacenter, regardless of the provider.
- 5) Processing sensitive data on a private trusted cloud, while processing less sensitive on a public cloud.
- 6) Borrowing resources from different providers in case of over-utilization or limited resources.

Many user-centric approaches have been proposed to decouple VMs from hypervisors and migrate them with minimal service interruption [1][2]. Three user-centric approaches are evaluated in [4] based on live cloud migration criteria to show deficiency with respect to performance, flexibility and security. Although, these approaches are able to deploy nested virtualization (e.g. Xen-Blanket [3] and the hypervisor, HVX [8]) they suffer significant performance degradation and limitation of VMs to support different operating systems. Moreover, securing live migration has not been considered due to the extra performance degradation caused by security mechanisms, such as IPsec VPN [5]. This paper introduces LivCloud to address the limitations in the proposed approaches with respect to performance, flexibility and security. LivCloud proposes to use different technologies, some of which have never been used in live cloud migration, such as the User Datagram Protocol based data transfer, -known as UDP-based data transfer or UDT and inter Software Defined Network (SDN) controller communication (ODL SDNi) [17][19]. Moreover, it uses KVM for the first time to enable nested virtualization on the cloud IaaS as well as securing the migration channel, which has not been considered in live cloud migration [4]. A preliminary experimental study is presented to validate the results.

The rest of the paper is organized as follows: Section II presents a brief summary of related work highlighting existing techniques to achieve nested virtualization on the cloud IaaS. Section III introduces the LivCloud architecture for live cloud migration. Section IV explains the current evaluation of LivCloud, including initial results. The conclusion and future work are presented in Section V.

II. RELATED WORK

Nested virtualization has been used to decouple the VM from public IaaS [1][2][8]. Nested virtualization is configuring one hypervisor (in the upper layer) within a virtual machine hosted by another hypervisor [26]. Most of legacy hypervisors, such as KVM, Xen, and VMware can run nested virtualization [2][3][8]. However, public cloud hypervisors do not allow running nested virtualization [1]. Two main techniques have been used to enable nested virtualization on the top of cloud IaaS, paravirtualization and binary translation. The Xen hypervisor can be configured to run paravirtualization concept, while VMware and hypervisor, HVX run binary translation [3][8]. KVM is limited in running paravirtualization. However, OPENFV has been developing KVM for running Network Function Virtualization (NFV), which will help overcoming KVMs limitations [12]. A brief discussion of both paradigms is presented in the following.

A. Paravirtualization

Paravirtualization allows different hardware architecture to be exposed to the VM. Subsequently, the VMs kernel has to be modified prior to OS installation, thus it does not support Windows OS [3]. Xen-Blanket is an academic approach is designed using Xen hypervisor. paravirtulaization significantly helped Xen-Blanket enabling nested virtualization on Amazon EC2 instance. Xen-Blanket has been used by many academic live cloud migration approaches like in [1][9][10]. These approaches inherited drawbacks of Xen-Blanket, including, significant downtime during live migration (1.4 seconds) [4], overhead (~ 30%) on drivers I/O [11] and difficulty to run unmodified OS (Windows) [3]. Furthermore, securing live migration has not been taken into account due to extra latency caused by encryption and authentication [5].

B. Binary Translation

It transfers the VM instructions directly to the underlying system and dynamically converts them to native x86 during runtime. As instance of binary translation, HVX is a proprietary hypervisor designed by Ravello systems [8]. This hypervisor enables nested virtualization on the top of Amazon EC2 and Google Compute engine [8]. The main drawback of HVX is its proprietary status which hinders evaluating its performance. Many experts are sceptical about the performance of binary translation, because it imposes extra overhead on the guest kernel [1]. Moreover, security which has not yet been implemented could be done using IPsec VPN [8].

III. LIVCLOUD ARCHITECTURE

LivCloud aims to achieve effective live cloud migration for VMs at cloud IaaS with minimal services interruption. It is similar to NFV Hypervisor-KVM Architecture using KVM as hypervisor and ODL as the SDN controller [12]. Figure 1 illustrates LivCloud architecture.

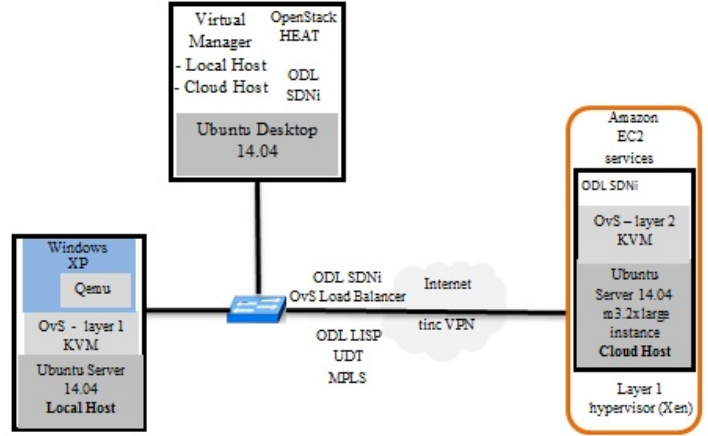


Fig. 1. LivCloud Architecture

LivCloud is designed based on previously proposed criteria [4], which are in three general categories: performance, flexibility and security. There are three performance criteria, denoted as- P1 live migration must be imperceptible to the migrated VM and its users; P2 predicting the required resources to decide whether or not to proceed with live migration; P3 monitoring resource utilization to avoid over utilization and to predict any possible failure. There are two flexibility criteria; F1 decoupling the migrated VM from underlying system by supporting wide range of hardware drivers, such as CPU drivers; F2 supporting various OS on the migrated VM, for instance, Windows. With respect to security, there are security criteria, S1 maintaining data privacy during live migration using encryption; S2 imposing authentication during migration.

To support effective live cloud migration, the design needs a foundation that supports nested virtualization to decouple VMs from the cloud IaaS and connect hypervisors on the IaaS in order to facilitate live migration back and forth. In addition to this, the design needs to optimize live migration performance, prevent any potential failure, and protect the process against hijacking and penetration.

A. Design Foundation: Nested Virtualization and Network Connectivity

Two fundamental features are necessary to establish network connectivity between LivCloud and the cloud IaaS. Firstly, Nested virtualization needs a particular hypervisor installed and configured on source and destination machines. LivCloud uses KVM as a Layer 1 hypervisor on the source and the destination and as a Layer 2 installed on certain VMs. Linux

virtual manager is a user interface for managing virtual machines mainly on KVM. Any physical or virtual machine that has KVM configured can be connected locally or remotely over SSH to virtual manager [27]. In LivCloud, two physical machines and three VMs are connected to the virtual manager installed on LivClouds management machine.

To fully achieve nested virtualization, KVM as a Layer 2 hypervisor must be enabled on Amazon Ubuntu m3.2xlarge instance. Secondly, both sides must be connected to LivClouds virtual manager in order to live migrate VMs between LivCloud and Amazon instance. KVM supports running modified and unmodified OS [2]. This step is currently underway, working closely with a cloud provider to enable nested KVM. The functional requirements help to fulfill F1, F2, and P1.

B. Design Optimization: Performance, Flexibility, and Security

The optimizations are related to performance, flexibility and security. LivCloud should guarantee enhancing network throughput, maintaining existing VMs connections, reserving required migration resources and securing the process which are briefly explained in the following:

1) *Enhancing network throughput*: Various technologies are leveraged to help to fulfill P1, including:

a) *OpenvSwitch (OvS)*: OvS has flow classification, caching and better performance over the traditional Linux Bridge. Moreover, it has its own load balancer, which is used to distribute loads across available routes [18][30].

b) *UDT*: UDT is used as the transport protocol between LivCloud and Amazon instance instead of TCP. The protocol's developers claim that UDT has throughput about five times more than TCPs and is as reliable and secure as TCP. Until 2009, UDT was able to win *Supercomputing 2009 Bandwidth Challenge Winner* [19]. Moreover, in [32], a comparison study is conducted between TCP and UDT, which shows that UDT performance is far better performance than TCP, especially in Long Flat Network (LFN) which is known as Bandwidth-Delayed Network. In 2013, VMware announced that UDT can be used to speed up VMs live migration between Datacenters that deploy VMware vCloud Air [33].

c) *Multiprotocol Label Switching (MPLS)*: MPLS has improved QoS attributes over WAN connections and recently it has been integrated into SDN controllers [20]. LivCloud incorporates MPLS into ODL controller to improve the network performance.

2) *Maintaining VMs connections and configurations*: Live migration of a VM from one site to another over the Internet needs to keep existing connections and configurations, such as, Access Control List (ACL) and DNS records. The following technologies used to maintain these configurations:

a) *ODL Inter SDN controller communication (ODL SDNi)*: Different ODL controllers can use this feature to instantly communicate to each other and pass any changes in the topology to each other, such as VMs relocation [13][17]. Two ODLs are configured on LivCloud and Amazon instance to deploy SDNi between both sites.

b) *ODL Locator/Identifier Separation Protocol (LISP)*: It is integrated into ODLs on both sides. LISP builds a routed layer on IP using two addresses. These two addresses, Endpoint Identifier (EIDs) and Routing Locator's (RLOCs) are used to decouple the VM from its fixed IP address and keep the existing connections when the VM is migrated [18].

3) *Reserving resources and prediction of potential failure*: To help fulfill P2 and P3, LivCloud aims to utilize OpenStack orchestrator, HEAT and a plug-in coded in Python [14][21]. These components will help to reserve enough resources for migration, finish the migration within predefined time, avoid any potential failure and prevent therefore throttling QoS attributes, such as Service Level Agreement (SLA).

4) *Securing the migration channel*: Due to the extra overhead processing and migration downtime added by security mechanism, such as IPsec to live migration, it has been avoided in many live cloud migration approaches [4]. In many cases, the downtime is increased about 5 times when IPsec added to live migration as the study in [5] shows. The study illustrates the increase of both migration downtime and total time migration, from less than two seconds to almost 8 seconds downtime when IPsec VPN is implemented. Moreover, in studies [1], [3], [9] and [10] the live migration is between a local deployment and Amazon services and there is no security mechanism used, despite the fact that Amazon offers load balanced IPsec VPN between its VPC and cloud users' local IaaS [16]. Data have to be encrypted during migration, thus it is protected from any penetration. Also, during migration, authentication has to be imposed in order to prevent any potential hijacking [28]. To maintain encryption (S1) and authentication (S2), LivCloud uses tinc VPN, which is able to provide encryption using Advanced Encryption Standard (AES) and authentication using Hash-based Message Authentication Code (HMAC-SHA1) [1].

IV. PRELIMINARY EXPERIMENTAL EVALUATION

A. Experiment Setup

The experiment aims to evaluate LivCloud within a LAN environment. Thus, the lab setup consists of four HP Z440 workstations are connected through Cisco L2 switch providing a 100 Mbps. Each machine has 32 GB of RAM, 1TB disk and 4-core 2.8GHz Intel(R) Xeon(R) E5-1603 v3 CPU. Ubuntu Server 14.04 LTS and KVM (a Layer 1 hypervisor) are installed and configured on two machines, H1 and H2. Using KVM, VMs, H3 is created on H1 and H4 on H2. KVM (a Layer 2 hypervisor) is configured on H3 and H4. Microsoft Windows 10 and VMware workstation 12 are installed on the third machine. Using VMware, H5 is created and equipped with Ubuntu Server 14.04 as well as KVM. The last machine is configured as a NFS server (FreeNAS 9.3) for LivCloud. Any VM can be configured with a local disk or a disk hosted on the NFS server. H3 and H4 have their disks hosted on the NFS server. Hosts H1, H2, H3, H4 and H5 are connected to the virtual manager as shown in Figure 2. A VM with Windows XP used as the migrated VM between all hosts. It has 1GB of RAM and 2vCPUs. The connection between

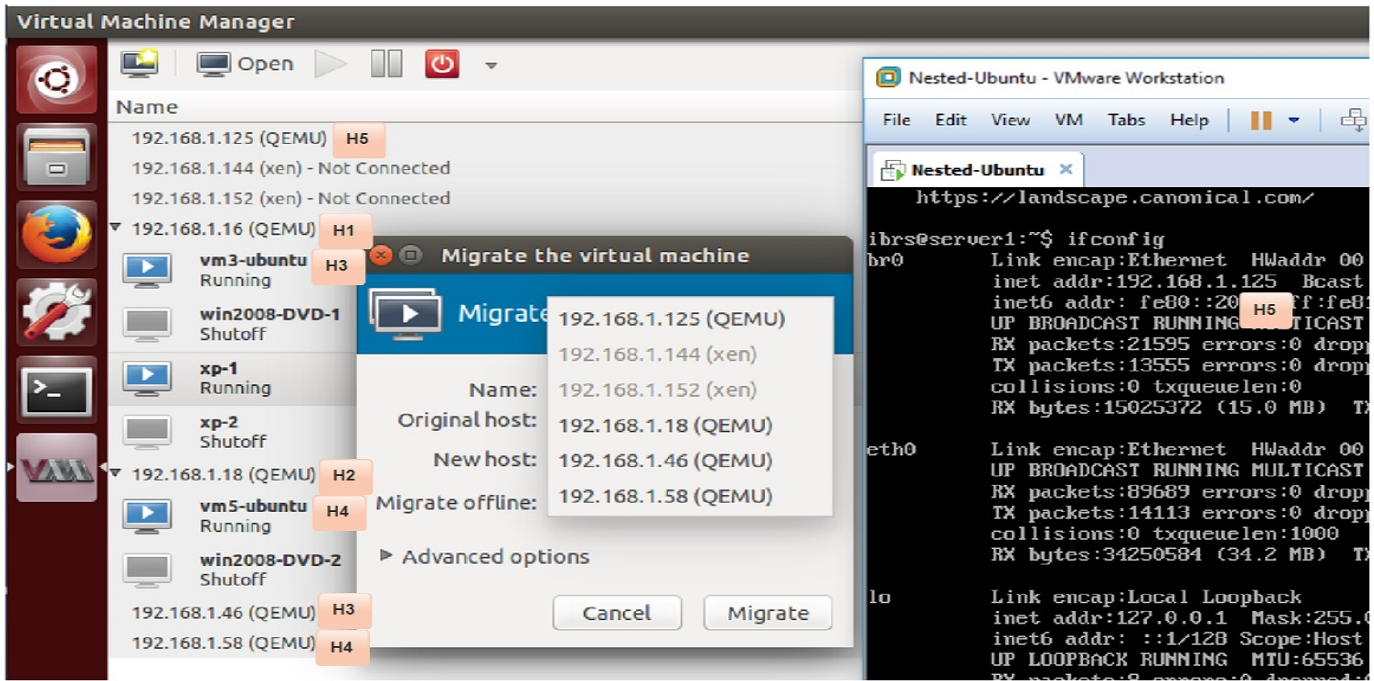


Fig. 2. Virtual Manager's connection to all hosts

LivCloud and Amazon instance is an essential part of live cloud migration. This connection is established through a VPN. Figure 3 shows how both sides are connected [16]. VPC is Amazon Virtual Network that helps building user-defined private network subnets inside the cloud in order to facilitate controlling IP address changes.

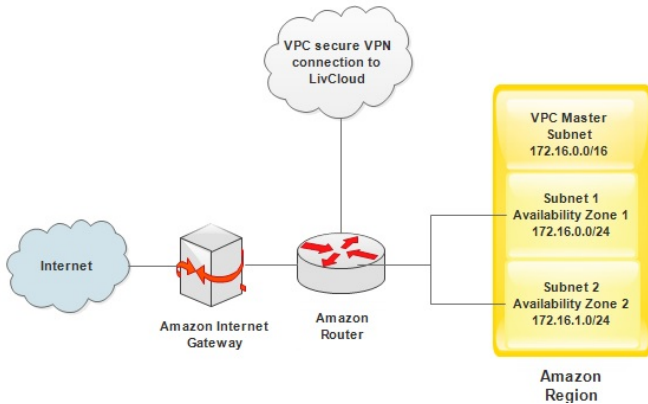


Fig. 3. LivCloud's connection to Amazon

B. Experiment Motivation

The main motivation behind conducting KVM live migration within the LAN environment is to illustrate that despite the LAN resources are less affected by the migration than the WAN, there is still a notable impact of the process on the LAN without the proposed criteria [4]. Network throughput, CPU utilization, network latency, migration downtime and disk I/O performance are the main parameters used to analyze

the live migration impact. Network throughput is measured using iPerf version3 [23] and migration downtime is measured by using Wireshark protocol analyzer [24], whereas disk I/O performance is tested using hdparm command [29].

C. Experiment Results Discussion

Figure 5, 4, 6 and 7 illustrate the experimental results. In particular, Figure 4 shows that there is downtime (0.07 seconds) during live migration between H5 and H1 (VMware to KVM) because of that there is no network latency. A

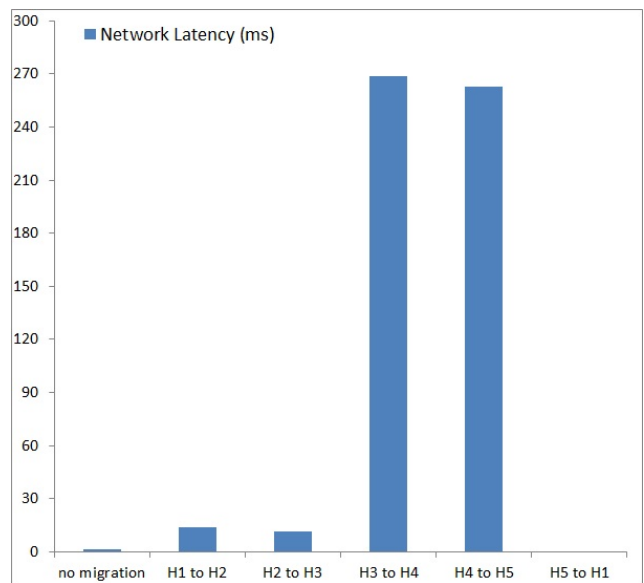


Fig. 4. Network latency statistics

remarkable increase in network latency is obtained when migrating from H3 to H4 and from H4 and H5 (a Layer 2 hypervisor). Figure 5 shows that the network throughput is highly affected, in particular when migrating between H3 and H4. In effect, the throughput decreases $\sim 25\%$ compared to the case of no migration. In Figure 6, the CPU load increases

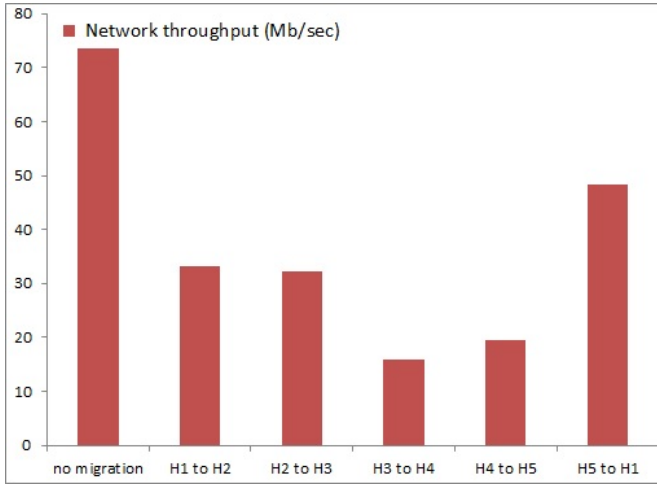


Fig. 5. Network throughput statistics

by almost 20% during live migration between H1 and H5 (KVM to VMware). Figure 7 shows that the I/O performance of disks that are hosted on the NFS server (H3 and H4) are severely affected by the migration. These hosts access their disks through the network to write and read data. The impact is notable in accessing the disks through the LAN. Therefore, it will be more pronounced in the case of WAN and live cloud migration.

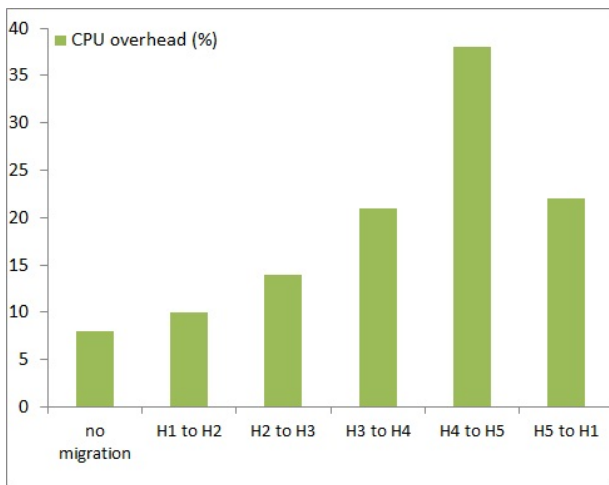


Fig. 6. CPU overhead statistics

D. Additional Discussion: VMs Disk Migration

Live migration of VMs disks has been considered in many studies [2][21]. However, it is considered to be unreliable and needs synchronization between CPU processing speed and

network bandwidth [25]. Moreover, many cloud users prefer keeping VMs disks in-house for more control and privacy [1][9][10]. As mentioned earlier, LivCloud uses KVM that has a live block migration feature that allows migrating the disks state [25]. However, during the initial evaluation of LivCloud, this feature showed instability and the process crashed many times. On account of this, the disk live migration is cloud users decision to use this feature or leave the disk on the shared storage in LivCloud.

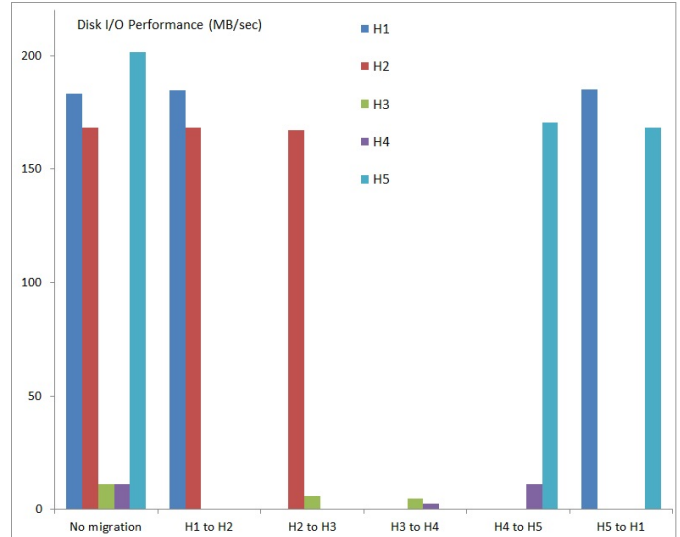


Fig. 7. Disk I/O performance

V. CONCLUSION

Given the requirements of cloud users, such as, cloud service continuity and data privacy, there is a clear need for live migration of VMs at IaaS. The current cloud providers IaaS is heterogeneous and hence, hinders live migration of VMs. Every provider deploys their own developed virtualization platforms. Many user-centric approaches to overcome virtualization heterogeneity, including Xen-Blanket and HVX, attempted to achieve the migration with minimal service disruption. While they have managed to devise a customized nested virtualization such as paravirtualization and binary translation, they have shown shortfalls in terms of migration downtime, decoupling VMs from underlying systems and securing the live migration.

LivCloud is designed to address the existing issue using different components, including KVM, OpenDaylight controller, UDT and OpenStack orchestrator, HEAT. The initial evaluation shows that live migration impacts the LAN resources, including network throughput and latency as well as CPU utilization and disk I/O performance before and during live migration. Optimization is needed to tackle the migration negative impact, in particular when live migrating between different hypervisors (KVM and VMware) and when VMs disks are hosted on an NFS server.

The next step of this research is to enable nested virtualization on Amazon IaaS using KVM customization. This step is

underway, working closely with a cloud provider to achieve it.

REFERENCES

- [1] Qin Jia, Zhiming Shen, Weijia Song, Robbert van Renesse, and Hakim Weatherspoon. Supercloud: Opportunities and Challenges. *SIGOPS Oper. Syst. Rev.* 49, 1, January 2015, 137-141.
- [2] Kaveh Razavi, Ana Ion, Genc Tato, Kyuho Jeong, Renato Figueiredo, Guillaume Pierre and Thilo Kielmann. Kangaroo: A Tenant-Centric Software-Defined Cloud Infrastructure. *IEEE International Conference on Cloud Computing (IC2E)*, March, 2015. DOI: 10.1109/IC2E.2015.19
- [3] Dan Williams, Hani Jamjoom, and Hakim Weatherspoon. The Xen-Blanket: virtualize once, run everywhere. *The 7th ACM European Conference on Computer Systems (EuroSys '12)*. ACM, New York, NY, USA, 2012, 113-126. DOI=10.1145/2168836.2168849
- [4] Ibrahim Mansour, Reza Sahandi, Kendra Cooper and Adrian Warman. Interoperability in the Heterogeneous Cloud Environment: A Survey of Recent User-centric Approaches. *ICC2016*, March 2016. DOI: <http://dx.doi.org/10.1145/2896387.2896447>
- [5] Anupam Tamrakar. Security in Live Migration of Virtual Machine with Automated Load Balancing. *International Journal of Engineering Research & Technology (IJERT)*, December, 2014. ISSN: 2278-0181
- [6] Apprenda. Server Virtualization. Available at: <https://apprenda.com/library/glossary/definition-server-virtualization/>. (Accessed: 06-04-2016)
- [7] Adel Nadjaran Toosi, Rodrigo N. Calheiros, and Rajkumar Buyya. Interconnected Cloud Computing Environments: Challenges, Taxonomy, and Survey. *ACM Comput. Surv.* 47, 1, Article 7, May 2014, 47 pages. DOI=10.1145/2593512
- [8] Alex Fishman, Mike Rapoport, Evgeny Budilovsky and Izik Eidus. HVX: Virtualizing the Cloud. 5th USENIX Workshop on Hot Topics in Cloud Computing, 2013. USENIX Association, Berkeley, CA, USA.
- [9] Dan Williams, Hani Jamjoom, Zhefu Jiang and Hakim Weatherspoon. VirtualWires for Live Migration Virtual Networks across Clouds. Report by Cornell University and IBM T.J Watson Research Center, New York, April 2013.
- [10] Dan Williams, Hani Jamjoom, and Hakim Weatherspoon. Software defining system devices with the 'Banana' double-split driver model. *The 6th USENIX conference on Hot Topics in Cloud Computing (HotCloud'14)*, 2014. USENIX Association, Berkeley, CA, USA, 5-5.
- [11] Mark Shtern, Bradley Simmons, Michael Smit and Marin Litoiu. An architecture for Overlaying Private Clouds on Public Providers. *The 8th International Conference and Workshop on Network and Service Management (CNSM) and System Virtualization Management (SVM)*, 2012. E-ISBN: 978-1-4673-3134-0
- [12] OPENFV. Available at: <https://www.opnfv.org/>. (Accessed: 06-04-2016)
- [13] G. Carrozzo, R. Monno, B. Belter, R. Krzywania, K. Pentikousis, M. Broadben, T. Kudoh, A. Takefusa, A. Vico-Oton, C. Fernandez and B. Puype J. Tanak. Large-scale SDN Experiments in Federated Environments. *IEEE International Conference on Smart Communications in Network Technologies (SaCoNeT)*, May, 2014. DOI: 10.1109/SaCoNeT.2014.6867764.
- [14] OpenStack Foundation. OpenStack Conceptual Architecture. Available at: http://docs.openstack.org/juno/install-guide/install/apt/content/ch_overview.html. (Accessed: 06-04-2016).
- [15] OPENDAYLIGHT. OPENDAYLIGHT, available at: <https://www.opendaylight.org/> (Accessed: 06-04-2016)
- [16] Amazon Web Services. Amazon Virtual Private Cloud Connectivity Options. Available at: http://media.amazonwebservices.com/AWS_Amazon_VPC_Connectivity_Options.pdf (Accessed: 06-04-2016)
- [17] OpenDaylight Summit. Inter SDN Controller Communication (SDNi). Available at: http://events.linuxfoundation.org/sites/events/files/slides/ODL-SDNi_0.pdf (Accessed:06-04-2016)
- [18] OpenDaylight. Whats New in Lithium. Available at: <https://www.opendaylight.org/whats-new-lithium> (Accessed: 06-04-2016)
- [19] UDT UDP-based Data Transfer. UDT: Breaking the Data Transfer Bottleneck. Available at: <http://udt.sourceforge.net/> (Accessed: 07-04-2016)
- [20] Gabriele Lospoto , Massimo Rimondini , Benedetto Gabriele Vignoli and Giuseppe Di Battista. Rethinking Virtual Private Networks in the Software-Defined Era. *IEEE International Symposium on Integrated Network Management (IM)*, May, 2015. DOI: 10.1109/INM.2015.7140314
- [21] Konstantinos Tsakalozos, Vasilis Verroios, Mema Roussopoulos and Alex Delis. Time-Constrained Live VM Migration in Share-Nothing IaaS-Clouds. *The 7th IEEE International Conference on Cloud Computing (CLOUD)* (July, 2014). DOI: 10.1109/CLOUD.2014.18.
- [22] Yong Cui, Yusong Lin, Yi Guo, Runzhi Li and Zongmin Wang. Optimizing Live Migration of Virtual Machines with Context Based Prediction Algorithm. *Advances in Intelligent Systems Research*, October, 2013. DOI: 10.2991/ccis-13.2013.102
- [23] iPerf3. Available at: <https://iperf.fr/iperf-download.php> (Accessed: 07-04-2016)
- [24] Wireshark. Available at: <https://www.wireshark.org/> (Accessed: 07-04-2016)
- [25] OpenStack. Issues with Live Migration. Available at: http://docs.openstack.org/openstack-ops/content/compute_nodes.html (Accessed:07-04-2016)
- [26] Zhenhao Pan, Qing He, Wei Jiang, Yu Chen and Yaozu Dong. NetCloud: Towards Practical Nested Virtualization. *IEEE international Conference on Cloud and Service Computing (CSC)*, December, 2011. DOI:10.1109/CSC.2011.6138541
- [27] Virtual Machine Manager. Available at: <https://virt-manager.org/> (Accessed: 13-04-2016)
- [28] Mahdi Aiash, Glenford Mapp and Orhan Gemikonakli. Secure Live Virtual Machines Migration: Issues and Solutions. *The 28th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, May, 2014. DOI: 10.1109/WAINA.2014.35.
- [29] hdparm. Available at: <https://wiki.archlinux.org/index.php/hdparm> (Accessed: 13-04-2016)
- [30] Ben Pfaff, Justin Petti, Teemu Koponen, Ethan J. Jackson, Andy Zhou, Jarno Rajahalme, Jesse Gross, Alex Wang, Jonathan Stringer, Pravin Shelar, Keith Amidon and Martin Casado. The Design and Implementation of Open vSwitch. *The 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI15)*, May, 2015. Oklsnd, CA, USA. ISBN 987-1-931971-21.
- [31] Whittaker, Z." Amazon Web Services suffers outage, takes down Vine, Instagram, others with it. *ZDNet*". Available at: <http://www.zdnet.com/article/amazon-web-services-suffers-outage-takes-down-vine-instagram-others-with-it/>. 2013. (Accessed: 01-06-2016).
- [32] Tatikayala Sai Gopal, Neha Gupta, Rahul Jain,Srinivasa Reddy Kamatham and Reddy Lakshmi Eswari P."Experiences in porting TCP application over UDT and their Performance Analysis". *IEEE International Conference in MOOC Innovation and Technology in Education (MITE)*, December, 2013. Jaipu, India. DOI:10.1109/MITE.2013.6756370
- [33] Prasenjit Sarkar. "Improving Performance with Path Optimization and UDT - VMware vCC 2.5". Available at: <http://stretch-cloud.info/2013/06/improving-performance-with-path-optimization-and-udt-vmware-vcc-2-5/>. 2013. (Accessed:01-06-2016)