2017 25th Euromicro International Conference on Parallel, Distributed and Network-Based Processing

An Intra-Cloud Networking Performance Evaluation on CloudStack Environment

Adriano Vogel*†, Dalvan Griebler*†, Claudio Schepke‡ and Luiz Gustavo Fernandes†

*Laboratory of Advanced Researches on Cloud Computing (LARCC),
Três de Maio Faculty (SETREM), 2405, Santa Rosa Av.– Três de Maio – RS – Brazil
†GMAP Research Group (FACIN/PPGCC),
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), 6681, Ipiranga Av. – Porto Alegre – RS – Brazil
‡Federal University of Pampa (UNIPAMPA), Laboratory of Advanced Studies (LEA) – Alegrete – RS – Brazil
Email: {adriano.vogel,dalvan.griebler}@acad.pucrs.br,
claudioschepke@unipampa.edu.br, luiz.fernandes@pucrs.br

***Abstract*—Infrastructure-as-a-Service (IaaS) is a cloud on-demand commodity built on top of virtualization technologies and managed by IaaS tools. In this scenario, performance is a relevant matter because a set of aspects may impact and increase the system overhead.Specific on the network, the use of virtualized capabilities may cause performance degradation (*eg.,*latency, throughput). The goal of this paper is to contribute to networking performance evaluation, providing new insights for private IaaS clouds. To achieve our goal, we deploy CloudStack environments and conduct experiments with different configurations and techniques. The research findings demonstrate that KVM-based cloud instances have small network performance degradation regarding throughput (about 0.2% for coarse-grained and 6.8% for fine-grained messages) while container-based instances have even better results. On the other hand, the KVM instances present worst latency (about 12.4% on coarse-grained and two times more on fine-grained messages w.r.t. native environment) and better in container-based instances, where the performance results are close to the native environment. Furthermore, we demonstrate a performance optimization of applications running on KVM.**

I. Introduction

Cloud Computing is a flexible paradigm that offers computational resources through service levels and deployment types [1], [2]. Among the cloud service models, IaaS is considered the base layer in cloud architecture, either for public (by providers) and private (inside the corporations) deployments. This is mostly because PaaS (Platform as a Service) and SaaS (Software as a Service) are built on top of it. It implies a large dependency on resources, performance, and reliability [3], [4].

IaaS is a suitable alternative for resources provision with more flexibility, better resources usage and on-demand deployments. A key aspect on cloud environments is the network, as it enables effective resources provision as services to end-users. Nonetheless, the network performance is relevant for application features (distributed processing, replication, load balance, and high availability) [5]. Additionally, communication and data transfers are required for clustering and distributed processing inside the cloud data centers and applications running on cloud instances.

The advantages of cloud computing comes with performance challenges [6] since the virtualization layer induces performance overhead. The cloud network efficiency relies on miscellaneous metrics (*e.g.,* throughput, latency), protocols (*e.g.,* TCP, UDP, SNMP), unique infrastructure environments, and the combination of drivers and underlying technologies. Consequently, the network performance concerns are complex to be addressed and improved. It requires careful planning, analysis, and a deep understanding.

The performance of the cloud environment is an important aspect for providers as well as for end-users. For this paper, we present a network end-to-end (between cloud instances) performance measurement, using network micro-benchmarks and comparing different private cloud deployments.

Our goal is to present performance evaluation for potential optimizations on network intensive workloads. Therefore, we contribute with a network performance evaluation and comparison regarding TCP throughput, latency, and number of connections per second. Only a few papers addressed the open challenges of cloud network performance [7]. In contrast, we considered additional environments, experiments and metrics with respect to the related works. Consequently, this contribution allows cloud administrators to improve the quality of services by optimizing the performance.

This paper is organized as follows. The cloud infrastructure scenario is presented in Section II. Experiments and performance of intra-cloud network are shown in Section III. Section IV discusses results and implications of this paper. This study's related work are presented and contextualized in Section V. The conclusion and future work are presented in Section VI.

II. Scenario

The cloud computing paradigm brought a novel way to design the computational infrastructure and systems provided as services to end-users [8]. The cloud became interesting through the combination of several consolidated technologies (*e.g.,* Clusters, Grids, Networks, Virtualization). On the other hand, there are still several challenges and optimization in order to improve the quality of service and experience.

The combination of network and virtualization introduced the concept of virtual networks, at the software level [9]. The advantages of virtualization provides enough arguments to be a current trend exploit virtualized data center infrastructures,

2377-5750/17 $31.00 © 2017 IEEE
DOI 10.1109/PDP.2017.40

and the size of these environments is growing up. However, it brings an additional complexity regarding resources management, which is a common challenge for efficiency and quality of services.

Moreover, the computing centers are using the cloud model to offer virtualized resources in the form of services to end users. In this approach, we aim at addressing performance aspects on private cloud deployments by using open source solutions. For example, the cloud instances are deployed using KVM and LXC technologies with the Libvirt API [10].

KVM (Kernel-based Virtual Machine) [11] is a popular open source software for full virtualization on Linux systems. KVM enables to run several Virtual Machine (VM) (*e.g.,* Linux, Windows) in the same physical host and each VM owns an isolated computational environment. Figure 1 shows a KVM based environment using the paravirtualized VirtIO driver for I/O operations and network connectivity offering to VMs.

Linux Containers (LXC) [12] aims to offer a computational isolated environment avoiding the overhead of full virtualization. The containers are separated using namespaces and control groups. The network connectivity offered to the containers often uses the Linux Bridges or the native interfaces. Finally, the major difference between KVM and containers is that LXC requires less software abstraction, using the same kernel of the native OS, while KVM emulates the environments for the virtual machines, as shown on Figure 1.

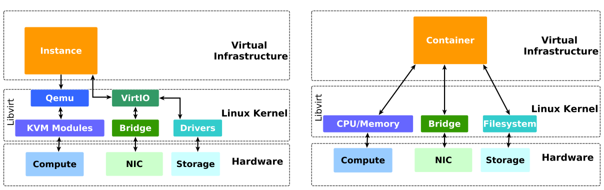

Fig. 1. Host Environment Overview. KVM (Left) and LXC (Right).

III. Network Performance Evaluation

As cloud providers use underlying technologies to enable an easier and on-demand services provision, cloud computing relies in an efficient network on computational centers to offer computational power. In this section, we present a network performance measurement of distinct private cloud deployments by using intensive workloads, we also show a results comparison.

A. Experimental Set-Up

In order to measure the performance aspects on cloud infrastructure, we chose to deploy environments and test the performance on created instances. We used identical machine hardware. Each one has 24 GB of RAM, Intel Xeon X5560 quad-core 2.80GHz CPU socket, and disks SATA II 7200 RPM. We used a Gigabit network.

As the chosen experiments are related to virtual environments, we used Ubuntu Server 14.04 (3.19.0 kernel) as native OS in order to deploy the cloud environments, using CloudStack IaaS tool version 4.8. After we had the cloud management server running, we added the nodes used for the experiments, by creating a cluster of KVM hypervisor and a container-based cluster using LXC.

The deployed environments shared the style of the physical servers hosting one large Linux instance, which had all hardware resources available. We chose such configuration because we tested on an isolated environment to avoid external interferences on running tests. Each environment was formed by two main aspects, the cloud controller (front-end or cloud management) and the compute nodes (back ends). The deployed controller is actually an IaaS platform, which manages the virtual pool (storage, network, users, VMs, among others). The controller allocated the instances in the nodes, dealing with the virtualization layer (LXC, KVM).

The nodes and controller were interconnected by a network switch. The network infrastructure on each environment used Ethernet since it is one of the most exploited network technology on cloud data centers [13]. We selected a flat implementation, using the Linux bridges in order to offer network connectivity to the cloud instances. This set-up was selected due to the small number of hosts on each cloud environment, not requiring advanced features (*e.g.,*OpenStack Neutron).

The experimental methodology relies on passive measurements for evaluating the network performance on private clouds. We chose benchmarks in order to evaluate the network aspects of the deployed environment. We used identical physical servers and network infrastructure to present a fair comparison. The results of each environment are presented below in the form of graphs.

B. Network Intensive Workloads

The cloud paradigm is still pushing for high-speed and efficient network connections, motivating network performance measurements. For evaluating the network infrastructure, we selected NetPIPE (Network Protocol Independent Performance Evaluator) [14] as an intensive benchmark to stress the network using the TCP (Transmission Control Protocol), which is the most used protocol on cloud environments. NetPIPE is used for measuring the Intra-cloud network throughput and latency. We also ran test using the Uperf benchmark rather than the synthetic way of the NetPIPE experiment.

Uperf [15] (Unified Performance tool for networking) is a multi-protocol tool for evaluating network performance aspects. Its experiments are proposed due to the lack of flexibility and real world applications behavior. Uperf ensures to generate custom network traffic and simulate network-based applications characteristics.

The Uperf connection experiments are varying the number of threads. This test started using a single thread and incrementally scales up to 32 threads. The results of the number of connections per second is an average of the communication between one cloud instance acting as a server and another as a client, which generates the data-flow. As a starting point, we used an Uperf "ping-pong" TCP traffic. Such experiment generates send and receive traffic sequences. Also, for experimental purpose, we defined the message size of 64 Bytes and the average is computed over a time interval of 90 seconds. This specific Uperf tests run continuously during a period of time, while other experiments need to be repeated several times, as the Uperf throughput measurements used by [16].

C. Intra-cloud Performance Evaluation

In this part, we present the results of the network measurement tests. The results are measured between two virtual machines launched in each environment. For this scenario, we chose the CentOS as the operating system, and we used results of the native environment as a baseline.

Figure 2 shows the results regarding the throughput metric. The graph presents the throughput of the environments with respect to an increasing size of messages (Bytes). It starts with 1 byte size and goes up to 10 MB. In general, the native environment performed better, followed by the containers performance. The instances using KVM had a slight lower throughput. However, we noticed small differences. A general throughput decrease can be noticed when using the message size around 16 KB, in which we detected as being a particular aspect of the network interfaces.

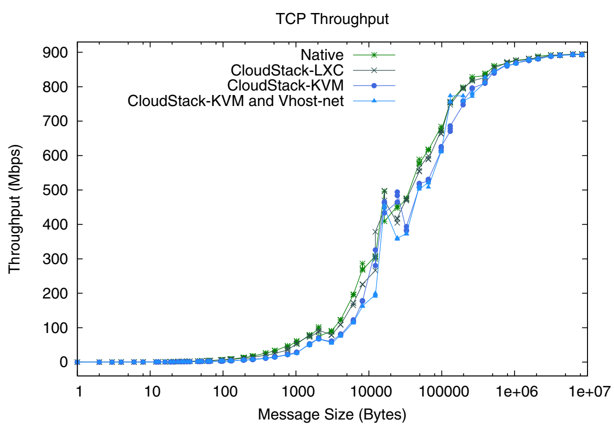

Fig. 2. TCP Throughput between two instances.

Latency is another relevant metric of applications performance. Figure 3 presents the latency results measured between the instances running on deployed environments. We chose the results using the message size between 1 Byte and 100 KBytes, which was considered the most relevant gap. Again, the native environment, as expected, outperforms the cloud instances while the performance of the containers was close to native. The results of instances running on KVM were only near to native using bigger size of the messages. Using sizes between 1 Byte and 1 KByte, we noticed a poor performance on KVM, with latencies more than the double of the time taken on the native environment. Another specific aspect is a noticed variation on results, which is actually something reasonable due to virtualization layer.

The metric of messages transferred per second using Uperf was chosen to compare the different environments. Figure 4 shows significant contrasts comparing the deployments, it represents the number of TCP connections per second as a throughput metric. As expected, the native environment outperforms the others, followed by the containers. The instances running on CloudStack using the KVM Hypervisor shows a

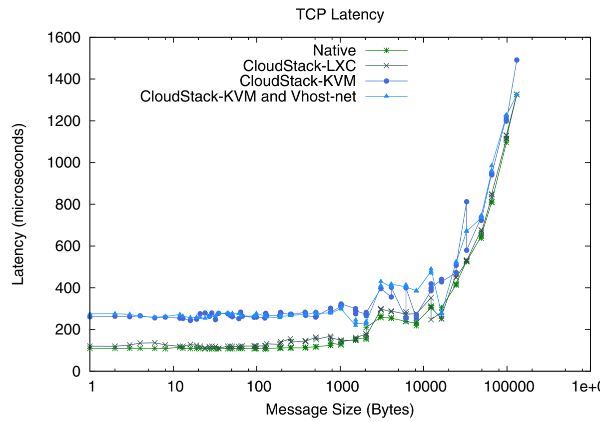

Fig. 3. TCP Latency between two instances.

poor performance, mainly with 8, 16 and 32 threads. The Uperf experiment with small messages size showed a degraded network performance on KVM environment.

Instead of only measuring the performance of the network between cloud instances, we also aim to point alternatives for improvements. As a consequence, we decided to seek improvements on KVM network performance, which raised due to the poor performance noticed on KVM instances. One approach for optimization presented on KVM documentation is the module vhost-net, which promises less overhead, since it decreases the copy operations using minor abstractions between the kernel and network adapters [17]. However, using the CentOS instances we were unable to exploit the vhost-net module. After enable and load the module in the kernel of the cloud services, we switch to Ubuntu Server instances in order to take advantage of the module.

The instances using the vhost-net had a better performance comparing to instances using the VirtIO driver. Such result emphasizes the need for network performance measurement and optimization. However, besides the benchmarks results, we needed a knowledge of the workload traces to allow the performance tuning that may improve the quality of services.

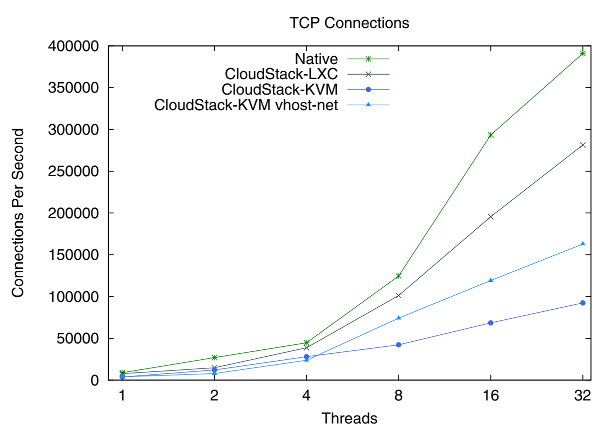

Fig. 4. Uperf Connections.

IV. Discussion of the Results

The intra-cloud network performance is specific to the physical environment and the used version of the native, KVM and LXC environments. Also, the network performance results are restricted to the workloads related with applications that behave similarly.

In this approach, we showed a performance evaluation of cloud deployments. The network throughput results were similar between distinct environments. The combination of the Linux Bridges and paravirtualized VirtIO driver used in KVM instances offered fair throughput rates. On the other hand, the native environment had the higher throughput, and LXC rates were slightly better (between 0.2% and 8%) than KVM environment averages, depending on grain sizes.

Despite the close averages in the throughput experiments, the latency times were distinct among the environments. Mainly, when using small packets the KVM instances had the worst latencies. For instance, using packets size of 64 Bytes the native latency was 109 microseconds while KVM latency was 283. Such results shown the need for performance measurements and evaluation of cloud deployments.

Our Uperf experiments are acting near to real world workload simulating the number of TCP requests per second. As the number of threads increases, more contrasting became the results. As expected, the native environment performed better, and again, the containers had the next best performance, for example, the contrast with 4 threads was 13.1% while with 32 threads increased to 28%. We also found significant performance degradation when testing the KVM instances. However, we introduced a performance optimization example by using the vhost-net module and showing performance improvements. The instances using vhost-net had a significant better performance when the number of threads increased, 43.1% with 8 and 43.2% better using 32 threads. It emphasizes the need for performance tuning options and user expertise to optimize the applications.

Complementing this discussion, LXC is an option for HPC (High Performance Computing), which tends to gain in performance compared to full or para-virtualization technologies. However, the containers are not always the most suitable alternative because of its limitations on flexibility and compatibility aspects. For example, LXC deployment on CloudStack IaaS does not support several native features (VNC console, snapshots, high availability, live migration) often present on CloudStack deployments at the moment being.

The need for low latency on cloud networks is a key challenge to improve the performance of the application running on cloud environments. The cloud deployed based on LXC container presented fair latency averages with small packet sizes. However, such environment suitability requires an analysis. The cloud support for High Performance Computing (HPC) applications relies on an efficient network and the literature is still having efforts towards performance improvements. One example is the study of [18] showing network latency improvements for an OpenStack cloud using the KVM hypervisor. Such approach implemented single root I/O virtualization (SR-IOV) in a network based on InfiniBand interfaces.

V. Related Work

The network role on cloud computing are evaluated mostly in the performance evaluation on public cloud providers. Table I summarizes the related work considered for virtual networking performance. The works of [7] and [21] evaluated network performance aspects on EC2 instances. Also, the study of [7] gave a special attention to the CPU role on network performance. This assumption is motivated by the high impact of the CPU and the communication degradation induced when the system is lacking on CPU power available for processing the network requests. These aspects are either showed in the study of [6], which analyze the Xen hypervisor architecture and the network performance of EC2 instances on geographical distributed area (WAN).

The convenience and growing utilization of public clouds brought an increasing number of related research. However, each provider uses specific virtualization solution (*e.g.,* Xen, KVM, VMWare) to abstract the hardware and place several instances in the same hardware. As a consequence, the performance of hypervisor impacts directly on applications running in public clouds. A set of researches have presented hypervisors performance insights, where we selected the ones that analyzed networking aspects. The study of [19] evaluated the Xen architecture and VMs allocation for performance improvements. A private cloud performance evaluation is shown by [20], which uses Eucalyptus and different deployments by measuring I/O aspects.

The network virtualization solutions are gaining attention for cloud deployments. While [22] presented a survey of these technologies and perspectives, [23] analyzed the architecture of network components for cloud deployments.

In this paper, we presented a network performance evaluation of private IaaS cloud deployments instead of in public providers such as [7], [21] and [6] studies. The performance study of [20] presented a closer approach, however, we considered the CloudStack platform using KVM. Also, our paper introduced the network performance on OS-level virtualization

TABLE I. Cloud Network Related Work Overview.

Paper	Network benchmarks	Network metrics	Environment	Results
[7]	TCPTest and UDPTest	TCP/UDP throughput, latency and packet loss	EC2	Virtualization may induce network performance losses
[19]	HTTP requests	Throughput (req/sec) and resource sharing	Xen	Network performance improvements on VMs allocation.
[6]	IPerf and Sysbench	Bandwidth and RTT latency over TCP and UDP	Xen on Amazon EC2 and WAN	Variability and impact of multi tenancy.
[20]	Netperf	Throughput (TCP)	Eucalyptus on KVM and Xen	Network Performance contrasts and overhead
[21]	Traceroute and files transfer	Network latency and throughput	EC2	Network isolation issues on Xen
This paper	NetPIPE and Uperf	Measurements of throughput, latency and connections	Private Cloud Deployments using KVM and LXC	Network performance evaluation and optimization.

with LXC. All these experiments were taken over a real CloudStack cloud deployment by using the same host OS. In contrast, experiments of [20] used different native OS and hypervisors (Ubuntu with KVM and CentOS with Xen), we used only the Ubuntu Server in the native layer.

The cloud network technologies and considerations presented on surveys [22], [23] concerns to the cloud deployments and architecture. In this approach, we chose to use the Linux Bridges as the virtual interfaces offering connectivity to running VMs. The CPU concerns presented by [7] and [6] are relevant for performance aspects, but we aim to address specific network performance characteristics.

Therefore, to the best of our knowledge, we are the first to present network performance measurement and optimization on private cloud, comparing the KVM and LXC deployments on TCP communication over latency and throughput metrics as well as considering application benchmarking with Uperf.

VI. Conclusion

This paper presented an evaluation regarding network performance of cloud computing. The cloud network infrastructure performance was evaluated by measuring the traffic between cloud instances. One of the main performance impacts of cloud applications are the network throughput and latency [5], [6]. The performance results of the paper are useful either for private and public clouds because the virtualization and cloud technologies deployed are used by several companies and institutions.

We can highlight over our experiments that the cloud instances using the KVM hypervisor achieved fair throughput rates, near the native environment. Such performance results are due to the I/O improvements on virtualization technologies (*e.g.,* KVM virtIO driver). On the other hand, the network latency of cloud instances over KVM may suffer performance degradation. Additionally, we demonstrated a performance optimization example by adjusting KVM modules, which resulted in performance improvements, as shown on Figure 4. Also, despite the fact that LXC had presented a better performance on latency, its poor support and compatibility limitations with IaaS platforms turn it into a drawback when deploying an enterprise and heterogeneous cloud environment.

In short, there is no single recipe to choose a suitable combination of cloud platforms and virtualization technologies. It is often a hard task, requiring a careful analysis mainly concerning the users and applications behavior (*e.g.,* resources, grain, communication). Also, the complexities of deployment and management depend on upon expertise as well as performance optimization.

As future work, we intend to: (I) customize the cloud deployments for testing different network technologies and hypervisors; (II) test the network performance and interference of several VMs running on the same server. (III) evaluate cloud platforms performance.

Acknowledgment

The authors would like to thank for the partial support from HiPerfCloud project and its supporters of SETREM, and PUCRS/FACIN. Also, we would like to thank the anonymous reviewers for their valuable suggestions.

References

[1] R. Buyya, C. Vecchiola, and T. Selvi, *Mastering Cloud Computing*. San Francisco, CA, USA: McGraw Hill, 2013.

[2] P. Mell and T. Grance, "The NIST definition of cloud computing," 2011.

[3] A. Iosup, N. Yigitbasi, and D. Epema, "On the performance variability of production cloud services," in *Cluster, Cloud and Grid Computing (CCGrid), International Symposium on*. IEEE, 2011, pp. 104–113.

[4] A. Vogel, D. Griebler, C. A. F. Maron, C. Schepke, and L. G. L. Fernandes, "Private IaaS Clouds: A Comparative Analysis of OpenNebula, CloudStack and OpenStack," in *24rd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*. Heraklion Crete, Greece: IEEE, Febuary 2016.

[5] L. Chao, *Cloud Computing Networking: Theory, Practice, and Development*. CRC Press, 2015.

[6] R. Shea, F. Wang, H. Wang, and J. Liu, "A deep investigation into network performance in virtual machine based cloud environments," in *INFOCOM*. IEEE, 2014, pp. 1285–1293.

[7] G. Wang and T. E. Ng, "The impact of virtualization on network performance of amazon ec2 data center," in *INFOCOM*. IEEE, 2010, pp. 1–9.

[8] E. Stahl, A. Corona, F. De Gilio, M. Demuro, A. Dowling, L. Duijvestijn, A. Fernandes, D. Jewell, B. Keshavamurthy, S. Markovits *et al.*, *Performance and Capacity Themes for Cloud Computing*. IBM Redbooks, 2013.

[9] A. Pathan, M. Monowar, and Z. Fadlullah, *Building Next-Generation Converged Networks: Theory and Practice*. CRC Press, 2013.

[10] Libvirt, "Libvirt virtualization API," 2016, last access Jul, 2016. [Online]. Available: http://libvirt.org/

[11] KVM, "Kernel Virtual Machine," 2016, last access Jul, 2016. [Online]. Available: http://www.linux-kvm.org

[12] LXC, "Linux Containers (LXC)," 2016, last access Jul, 2016. [Online]. Available: http://linuxcontainers.org/

[13] K. Hwang, J. Dongarra, and G. Fox, *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*. Elsevier Science, 2013.

[14] Q. O. Snell, A. R. Mikler, and J. L. Gustafson, "Netpipe: A network protocol independent performance evaluator," in *IASTED international conference on intelligent information management and systems*, vol. 6. Washington, DC, USA), 1996.

[15] Uperf, "Unified Performance tool for networking (uperf)," 2016, last access Jul, 2016.

[16] P. Rygielski and S. Kounev, "Data center network throughput analysis using queueing petri nets," in *2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW)*. IEEE, 2014, pp. 100–105.

[17] Redhat, "Virtio and vhost-net," 2016, last access Jul, 2016. [Online]. Available: https://access.redhat.com/documentation

[18] V. Mauch, "Deployment of Virtual InfiniBand Clusters with Multitenancy for Cloud Computing," *CLOUD COMPUTING 2015*, p. 81, 2015.

[19] Y. Mei, L. Liu, X. Pu, and S. Sivathanu, "Performance measurements and analysis of network i/o applications in virtualized cloud," in *Cloud Computing (CLOUD) International Conference on*. IEEE, 2010, pp. 59–66.

[20] J. Shafer, "I/O virtualization bottlenecks in cloud computing today," in *Proceedings of the 2nd conference on I/O virtualization*. USENIX Association, 2010, pp. 5–5.

[21] S. K. Barker and P. Shenoy, "Empirical evaluation of latency-sensitive application performance in the cloud," in *Proceedings of conference on Multimedia systems*. ACM, 2010, pp. 35–46.

[22] N. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862–876, 2010.

[23] K. Bakshi, "Network considerations for open source based clouds," in *Aerospace Conference, 2015 IEEE*, March 2015, pp. 1–9.