Low-Latency Software Defined Network for High Performance Clouds

Paul Rad[1], Rajendra V. Boppana[2], Palden Lama[2], Gilad Berman[3] and Mo Jamshidi[1]

[1]Department of Computer Engineering, University of Texas at San Antonio, USA
[2]Department of Computer Science, University of Texas at San Antonio, USA
[3]Technology and Engineering, Mellanox Technologies, USA
Email addresses: {paul.rad, rajendra.boppana, palden.lama}@utsa.edu, giladb@mellanox.com, moj@wacong.org

***Abstract*—Multi-tenant clouds with resource virtualization offer elasticity of resources and elimination of initial cluster setup cost and time for applications. However, poor network performance, performance variation and noisy neighbors are some of the challenges for execution of high performance applications on public clouds. Utilizing these virtualized resources for scientific applications, which have complex communication patterns, require low latency communication mechanisms and rich set of communication constructs. To minimize the virtualization overhead, a novel approach for low latency network for HPC Clouds is proposed and implemented over a multi-technology software defined network. The efficiency of the proposed low-latency Software Defined Networking is analyzed and evaluated for high performance applications. The results of the experiments show that the latest Mellanox FDR InfiniBand interconnect and Mellanox OpenStack plugin gives the best performance for implementing VM-based high performance clouds with large message sizes.**

Keywords—InfiniBand; SR-IOV; Software Defined Networking; Cloud; HPC; OpenStack

I. INTRODUCTION

Clusters of independent processors are used for parallelization in High Performance Computing (HPC) environment. HPC typically utilizes the Message Passing Interface (MPI) protocol to communicate between the processes. In the traditional approach, these applications are executed on compute clusters, super computers or Grid Infrastructure [7], [15] where the availability of resources is limited. High performance computing employs fast interconnect technologies to provide low communication and network latencies for tightly coupled parallel compute jobs. The compute clusters are typically linked by high-speed network using gigabit network switches or InfiniBand. Contemporary HPC grids and clusters have a fixed capacity and static runtime environment; they can neither elastically adapt to dynamic workloads nor allocate resources efficiently and concurrently among multiple smaller parallel computing applications [7].

Cloud technology uses an infrastructure that involves a large number of computers connected through a network. Cloud service allows users to provision resources easily and quickly by paying only for their usage of the resources. Cloud computing offers the benets of utility and elastic pool of resources, plus it eliminates the initial cluster setup cost and time [9]. However, poor network performance, virtualization overhead that causes performance degradation, quality of service, and noisy neighbor issues are some of the challenges for execution of real-time, high performance, tightly coupled, parallel applications on Cloud.

A low latency and reliable network with Software Defined Networking among cloud servers is a key element for a cloud infrastructure to be capable of running scientific applications, facilitating the transfer of data and communication between cloud servers [13]. InfiniBand is an interesting technology since it offers one of the highest throughputs and lowest latency, guaranteeing link Quality of Service (QoS) and scalability. It is often used in supercomputers and high performance computing [8]. One major challenge to overcome in the deployment of high-performance cloud network is the overhead introduced by virtual switches and virtual devices used and shared by the cloud servers. The Single Root I/O Virtualization (SR-IOV) interface, an extension to the PCI Express (PCIe) specification, overcomes the virtualization overhead by providing device virtualization through virtual functions that reside in the device [18], [19]. This model allows the hypervisor to simply map virtual functions to cloud servers, which can achieve the native device performance even without using pass through [6], [12].

The characterization of the InfiniBand in bare-metal and virtualized environments have been thoroughly evaluated by the HPC and Virtualization communities [6], [16], [22]. However, a comprehensive solution to support HPC applications with low-latency communication at the scale of Cloud systems is lacking. **In this paper, we present a novel solution to enable scalable and low-latency networking capability in the Cloud. Our solution integrates InfiniBand Software Defined Networking with an open-source Cloud management software, OpenStack. Such integration will truly bring the benefits of software defined networking, and InfiniBand to the cloud**. It provides a scalable architecture for high speed networking in the Cloud, and also facilitates network programmability of OpenStack networking using APIs.

The contribution of this paper is twofold. First, we enable a dynamic configuration of InfiniBand software defined networking with SR-IOV virtualization using OpenStack neutron plugin in a cloud environment. To the best of our knowledge this is the first paper to present a dynamic flexible low-latency Software Defined Networking architecture for cloud to support high performance computing. Second, we conduct extensive

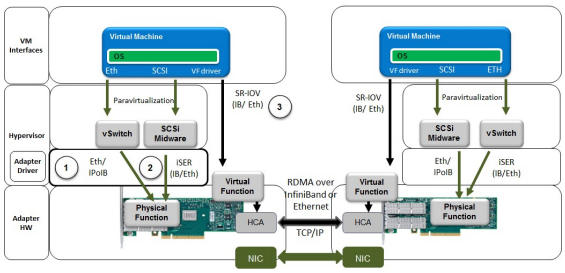


Fig. 1. Three ways to leverage RDMA in a cloud environment

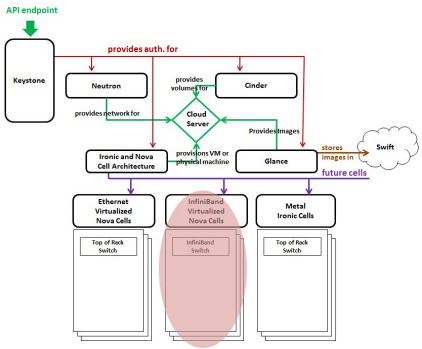


Fig. 2. OpenStack Architecture with InfiniBand Virtualized Nova Cell

performance evaluation of the proposed architecture using micro benchmarks and an HPC computation library. In order to understand the latency and bandwidth performance implication of the proposed approaches on cloud resources, a broad performance analysis has been conducted using OpenStack based cloud and low latency Software Defined Networking with Mellanox neutron plugin. Throughout the paper, latency and bandwidth efficiency is defined as the percentage of latency and bandwidth in the virtualized environment compared with the non-virtualized environment with the same resources. To measure the performance and efficiency, first we measured individual characterization such as bandwidth and latency using the IB-verbs and the Intel MPI micro benchmarks [15] with different communication and computation characteristics. Second, we used an application level benchmark such as the HPL Linpack to measure the efficiency and the overall performance of a typical scientific application. Our results show that, when large messages used for communication among cloud servers with SR-IOV virtualization, the performance degradation due to network virtualization overhead is low (less than 5%). However, when small message sizes are used for communication, a reduction in performance can be expected compared to the standard HPC grid configuration.

The remainder of the paper is organized as follows. Section 2 provides the background information, an overview of related work and our approach for a low latency software defined network for HPC clouds. Section 3 presents a brief introduction to the benchmarks we used and the results of our evaluations. Section 4 concludes the paper with directions for future work.

II. BACKGROUND AND RELATED WORK

High performance computing in the cloud has gained significant research attention in recent years [10], [11], [14], [17], [20]. Marathe et al. [17] evaluated the cloud against traditional high performance clusters along turnaround time and cost.

The characterization of the InfiniBand in bare-metal and virtualized environments have been thoroughly evaluated by the HPC and Virtualization communities [6], [16], [22]. However, to the best of our knowledge, this is the first paper that offers dynamic configuration of InniBand software defined networking with SR-IOV in a cloud environment. Our design is based on several building blocks, which we introduce in this section. Further, we present related work, such as concepts for low latency software defined networking, InfiniBand, and OpenStack.

A. OpenStack Cloud Architecture

OpenStack is an open-source cloud management software, which consists of several loosely coupled services, designed to deliver a massively scalable cloud operating system for building public or private clouds. To achieve this, all of the constituent services are designed to work together to provide a complete Infrastructure as a Service (IaaS). All the services collaborate to offer flexible and scalable cloud solution using API [3].

Rackspace and NASA announced the OpenStack project in July of 2010. The OpenStack software consists of several loosely coupled services with well-defined APIs. While these APIs allow each of the services to use any of the other services, it also allows an implementer to switch out any service as long as they maintain the API.

B. Software Defined Networking

Software Defined Networking is an emerging architecture, which decouples the network control and the flow of packets in the data plane [4], [21]. This new approach makes network management dynamic and adoptable for the high-bandwidth and dynamic nature of todays highly scalable applications. It is a network technology that allows centralized programmable control plane to manage the entire data plane. It allows open API communication between the hardware and the operating system, and also between network elements (Physical and Virtualized) and operating system. However, integration of Software Defined Networking with an open source Cloud management software, OpenStack, has not been explored, and evaluated so far.

C. The InfiniBand and SR-IOV Architecture

In this section we provide a short overview of InfiniBand followed by a description of the SR-IOV in the context of our research.

1) InfiniBand Overview: InfiniBand is a high-performance network technology, which is in widespread use in low latency clusters [15]. Compared to network technologies such as Ethernet, IB has a substantial performance advantage through aggressive protocol offloading; all layers up to the transport layer are handled completely in network adapters with Remote Direct Memory Access (RDMA) over InfiniBand. RDMA is a zero-copy, CPU bypass technology for data transfer and is

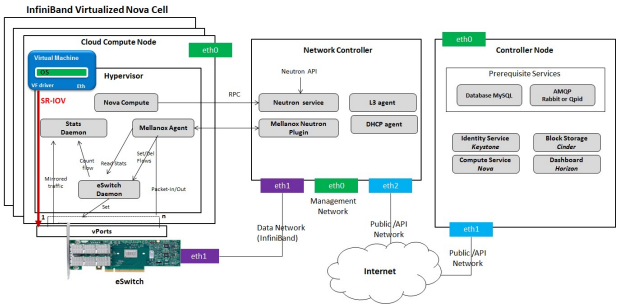


Fig. 3. Integration of Mellanox Neutron Plugin

supported over standard interconnect protocols. RDMA allows applications to transfer data directly to the buffer of a remote application and therefore provides extremely low latency data transfers. The Operating System (OS) is involved only in establishing connections and registering memory buffers to ensure protection. Applications bypass the OS to trigger actual communication operations and poll for their completion, by directly accessing device memory. As a result, an application can handle complete send/receive cycles independently and without added latency from the intervention of the OS.

Another capability of the InfiniBand stack, shown in Figure 1, is IP over InfiniBand (IPoIB) protocol. IPoIB protocol defines how to send IP packets using InfiniBand by creating a normal IP network interface [2]. IPoIB impacts the functionalities and performance of the InfiniBand protocol stack in spite of this the user can deploy widespread set of TCP/IP based applications. IPoIB impacts the functionalities and performance of the InfiniBand protocol stack in spite of this the user can deploy widespread set of TCP/IP based applications.

2) SR-IOV Overview: Single Root IO Virtualization (SR-IOV) allows a physical PCIe device to present itself as a multiple devices on the PCIe bus. This technology enables a single adapter to provide multiple virtual instances of the device with separate resources. Mellanox ConnectX-3 adapters are capable of exposing virtual instances, called Virtual Functions, which can be provisioned separately. Each Virtual Function (VF) can be viewed as an additional device associated with the Physical Function (PF). In the host hypervisor, these VFs appear as a set of independent PCIe InfiniBand devices.

SR-IOV is commonly used in conjunction with the SR-IOV enabled hypervisor to provide virtual machines with direct hardware access to network resources, thereby improving performance. SR-IOV enables close to zero virtualization latency penalties through RDMA and Hypervisor bypass.

III. Integration of InfiniBand Software Defined Networking with OpenStack for low-latency HPC Cloud

We now present our work on the integration of InfiniBand Software Defined Networking with OpenStack. The implementation described in this paper is based on the Havana release of the OpenStack distribution.

A. Multi-cell Cloud Infrastructure

To support a heterogeneous infrastructure for researching and examining multi-technology, software defined networking; we implemented a multi-cell cloud infrastructure as shown in Figure 2. The cell structure enables distributed tree topology with two-degree-of-freedoms with respect to cell and compute nodes scheduling.

B. Integration of Software Defined Networking controller into OpenStack Neutron

As shown in Fig 3, we integrated Software Defined Networking controller into OpenStack Neutron using Mellanox plug-ins. Such integration truly brings the benefits of software defined networking, and InfiniBand to the Cloud. It provides a scalable architecture for high speed networking in the Cloud, and also facilitates network programmability of OpenStack networking using APIs.

Mellanox Neutron plugin provides the required integration of Mellanox devices with Neutron service. The Mellanox Neutron plugin creates and configures hardware vNICs based on SR-IOV virtual functions and enables each Virtual Machine vNIC to have its unique connectivity, security, and QoS attributes. The Neutron plugin enables the switching in a Mellanox embedded switch in the Virtual Path Identifier (VPI) Ethernet/InfiniBand network adapter. Hardware vNICs are mapped to the guest VMs, through para-virtualization using a TAP device or directly as a Virtual PCI device to the guest via SR-IOV, allowing higher performance and advanced features such as remote direct memory access (RDMA).

OpenStack Neutron controller with Mellanox plugin comprises of following elements:

1) Neutron-server, a python daemon, is the main process of the OpenStack Networking that runs on the OpenStack Network Controller.
2) Mellanox OpenStack Neutron Agent runs on each compute node, mapping between a VM vNIC (VIF) and an embedded switch port, thus enabling the VM network connectivity.
3) The Mellanox Nova VIF driver is used together with the Mellanox Neutron plugin. This driver supports the VIF plugin by binding vNIC para-virtualized or SR-IOV with optional RDMA guest access to the embedded switch port.
4) DHCP agent, a part of Neutron, provides DHCP services to tenant networks. This agent maintains the required DHCP configuration.
5) L3 agent: This agent is responsible for providing Layer 3 and NAT forwarding to gain external access for virtual machines on tenant networks.

Mellanox Neutron plugin is part of the upstream OpenStack release and provides unique value added features such as transparent InfiniBand network management and configuration. Mellanox Neutron plugin automatically configures the IB SR-IOV interface for the virtual machines and also assign Pkey to the interface to provide multi tenancy and network isolation support.

In our InfiniBand Nova cell, each VF is directly associated to a cloud server that has the exclusive use of that function

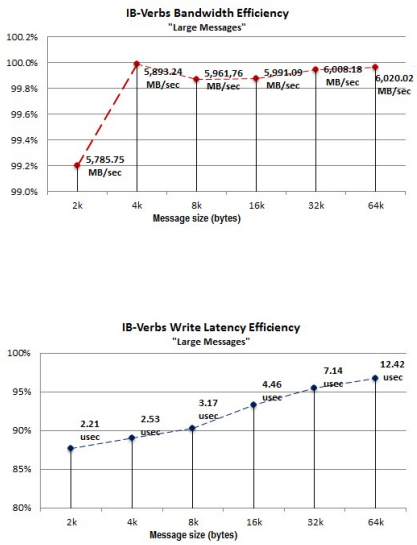


Fig. 4. IB-Verbs latency and bandwidth efficiency for large messages

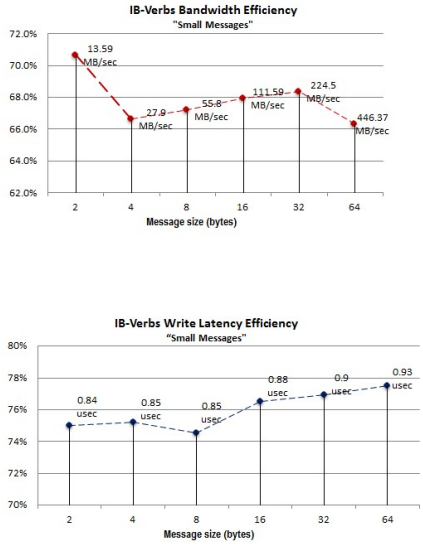


Fig. 5. IB-Verbs latency and bandwidth efficiency for small messages

without using any device emulation in hypervisor [5]. And each PF and VF receives a unique PCI Express Requester ID that allows the I/O Memory Management Unit (IOMMU) to differentiate the traffic among VFs.

The eSwitch, Mellanox ConnectX-3 adapters are equipped with onboard-embedded switch (eSwitch) and are capable of performing layer-2 switching for the different virtual machines running on the server. Higher performance levels can be achieved using eSwitch, since the switching is handled in hardware and reduces CPU overhead.

IV. Performance Evaluation

A. Testbed Setup

To evaluate Low-Latency Software Defined Network properties and performance for HPC clouds, we set up two child

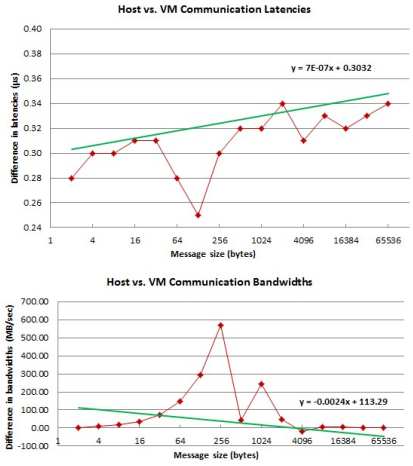


Fig. 6. Host vs. VM communication latencies and bandwidths

cloud cells, an InfiniBand Virtualized Nova Cloud cell and an Ethernet Virtualized Nova Cloud cell, under the top-level Open Cloud cell, in our Cloud Data Center. The InfiniBand cell and the Ethernet cloud cell for the evaluation comprises of 16 high-performance Open Compute servers with ConnectX-3 IB adapters and interconnected by FDR 56Gigabit InfiniBand Mellanox switch. The InfiniBand cell comprises of 8 servers with two 10-core Intel(R) Xeon(R) CPU E5-2670 v2 @ 2.50GHz processors and 256 GB RAM. All servers run Cent OS 6.5 with Linux kernel 2.6 and KVM hypervisor kvm-kmod-3.2. We use the Havana version of OpenStack, whereby 8 servers are configured as OpenStack compute servers as shown in Figures 2 and 3.

The Open Cloud cell runs a nova-api service. Each child cell runs all of the typical nova-* services in a regular OpenStack cloud except for nova-api. Each child cell has its own scheduler, database server and message queue broker. The nova-cells service handles communication between cells and selecting a cell for new instances. The communication between cells is pluggable via Remote Procedure Call (RPC). Once a cell has been selected and the new instance request has reached its nova-cells service, it will be sent over to the node scheduler in that cell and the build proceeds as it does normally. We maintain only one cloud server per hypervisor. To achieve 1:1 virtual to physical ratio and control the placement of cloud servers in each cell, we leverage InstanceGroupApiExtension API.

B. Benchmarks

We have used IB-Verbs benchmarks for network level experiments. All MPI level experiments were run using the Intel MPI benchmark 4.1.3.048 [1]. Then we used application level benchmark such as the HPL Linpack to measure the efficiency and the overall performance of a typical scientific application. We report results that were averaged across multiple runs to ensure fair comparisons.

C. Network Level Performance Evaluation

In this section, first we present the performance evaluation of proposed architecture compared to bare-metal servers. We

use IB-Verbs and the Intel MPI Benchmark with different communication and computation characteristics to understand the overhead of cloud resources with pass-through drivers.

Figure 4 shows virtualization efficiency calculated from the ratio of bandwidth and latency measurements of IB-Verbs communication between two cloud servers in different hosts and separate measurements of direct IB channel between two hosts for 2K bytes or larger messages. For larger message sizes, the difference becomes insignificant and the overhead in latency and bandwidth diminishes to nearly zero with very large message sizes. In this scenario the results are extraordinary, with cloud servers achieving the same network throughput and latency of the host.

Figure 5 shows the virtualization efficiency calculated from the ratio of bandwidth and latency measurements of IB-Verbs communication between two cloud servers in different hosts and separate measurements of direct IB channel between two hosts for less than 64 bytes message sizes. When using the IB-verbs benchmark, we witness a big difference for small messages. For messages less than 64bytes, the extra latency caused by virtualization and SR-IOV is on the order of 30%.

Figure 6 plots the differences in VM-to-VM and host-to-host latencies for messages with sizes from 2 bytes to 64 KB. The trend line indicates that there is an additional message startup overhead of 0.303 microseconds with SR-IOV and virtualization. The per-byte overhead due to SR-IOV and virtualization is negligible: less than a picosecond. This data clearly demonstrates that with appropriate isolation of resources allocated to users, the impact of virtualization can be lower as the message sizes increase.

D. MPI Level Performance Evaluation

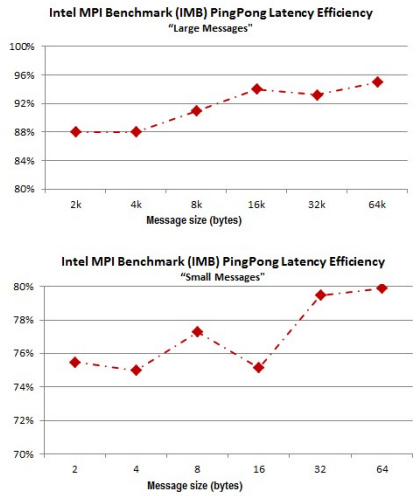
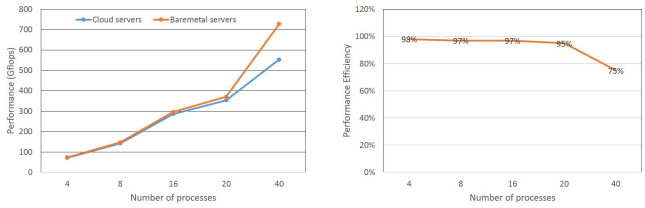


Fig. 7. Intel MPI Benchmark (IMB) PingPong Latency Efficiency

After micro benchmark IB-Verbs diagnostics tests between hosts and cloud servers, to scale the tests up and evaluate if the efficiency with the IB-Verbs is continued, we used the Intel MPI Benchmark to measure latency and bandwidth. Figure 7 represents virtualization efficiency calculated from the ratio of bandwidth and latency measurements of the Intel MPI Benchmarks between two cloud servers in different hosts and separate measurements of direct IB channel between two hosts with different message sizes. The results are similar to that of IB-Verbs latency. For a less than 64 bytes message sizes, the latency efficiency is on the order of 25% to 35%.



(a) Performance comparison. (b) Performance efficiency of cloud servers.

Fig. 8. HPL Linpack benchmark performance with increasing number of processes.

E. HPC Application Level Performance Evaluation

After testing the performance of the network using micro benchmarks, we used HPL Linpack, an application level benchmark, to measure the efficiency and the overall performance of a typical scientific application. To test the worst case scenario in our experiment, we deployed one cloud server per hypervisor and increase the number of processes while the HPL matrix size is kept constant.

Figure 8(a) compares the performance of HPL Linpack benchmark when running on two cloud servers versus when running on two bare-metal servers. As the number of processes increase, the performance increases in both cases. The performance of cloud servers, and bare-metal servers increase at similar rates initially. However, as the number of processes exceed a certain limit, the performance of bare-metal servers increase more rapidly than that of the cloud servers. As a result, the performance efficiency of our HPC cloud decreases as shown in Figure 8(b).

The performance trend observed in Figure 8 is due to the following reasons. HPL performance is influenced by the amount of computation per process and message sizes. As the number of processes increase, the fraction of computation decreases linearly, while the message sizes also decrease proportionately. This is due to the fact that the input matrix size (the amount of computation) is kept constant as the number of processes is increased. Due to the decrease in the fraction of computation, communication efficiency has a greater impact on performance, as the number of processes increase. Furthermore, due to the decrease in message sizes with increasing number of processes, the communication efficiency itself goes down. This phenomenon is also illustrated in the network level experiments using IB-Verbs benchmark. Therefore, the increase in the number of processes has a double-whammy effect on the performance of HPL Linpack benchmark.

In a preliminary experiment, when large number of processes (40 processes) deployed on two cloud servers, we observed that performance efficiency is only about 75% of the performance with HPC grid configuration due to message size impact illustrated in IB-Verbs experiments.

OpenStack Neutron plugin and SR-IOV are still in their early stages. Nevertheless, this low-latency Software Defined

Networking technology, with very little tuning, delivers almost 100% of efficiency in bandwidth and latency for large-message communications. However, the network virtualization and cloud management software will need to be tuned carefully to achieve high communication efficiency.

V. CONCLUSION AND FUTURE WORK

Both cloud platforms and traditional grid/cluster systems have different advantages and disadvantages in support of HPC applications. In this paper, InfiniBand Low Latency Software Defined Network for high performance cloud was studied in order to understand the performances of I/O bound scientific workloads deployed on public or private cloud infrastructures.

The InfiniBand Low Latency Software Defined Network combined with SR-IOV is a new architecture proposed for the next-generation High Performance Clouds. This involves two important ideas. First, centralized programmable network control plane architectures with multi-protocol low latency plugins will replace todays proprietary network architectures for hyper-scale infrastructures. Second, cloud computing architectures are used for high performance computing. To the best of our knowledge this is the first paper to present a dynamic flexible low-latency networking architecture for clouds to support high performance computing. Another major contribution of this paper is the evolution of the proposed architecture with micro and application level benchmarks.

Our experimental results show that Software Defined Networking can provide operators and providers with unprecedented flexibility and centralized control in building hyper-scale high performance cloud infrastructure required for scientific applications. The result of the experiment is exceptional, with cloud servers achieving the same network throughput and latency of the hypervisor for large message transfers. Our results for micro benchmarks show that there is about 0.3 microseconds of overhead for message set up introduced by SR-IOV Virtual Function; the additional cost per byte is negligible. Also, the overhead of the network control plane is negligible since it is required at the beginning to set up the SR-IOV VF functionality. Our Linpack experiments show that when computation is high per proses and message sizes are large, the proposed cloud provides more than 95% of the bare-metal performance, which is consistent with the results from micro benchmarks. However the cloud performance can decrease measurably when the numbers of messages increase while message sizes decrease.

The proposed architecture is our first attempt to make cloud architecture suitable for HPC applications that requires high performance communication support. In future, we will be investigating further the impact of the entire network stack at a even larger scale.

REFERENCES

[1] Intel mpi benchmarks. https://software.intel.com/en-us/articles/intel-mpi-benchmarks/.

[2] Ip over infiniband. http://www.ietf.org/html.charters/ipoib-charter.html.

[3] Openstack: Open source cloud computing software. https://www.openstack.org.

[4] A. Devlic, W. John, and P. Skoldstrom. A use-case based analysis of network management functions in the onf sdn model. In *Proc. IEEE European Workshop on Software Defined Networking*, pages 85–90, 2012.

[5] I. L. A. Division. Pci-sig sr-iov primer: An introduction to sr-iov technology. White paper.

[6] Y. Dong, X. Yang, J. Li, G. Liao, K. Tian, and H. Guan. High performance network virtualization with sr-iov. *Journal of Parallel Distributed Computing*, 72(11):1471–1480, 2012.

[7] I. Foster, Y. Zhao, I. Raicu, and S. Lu. Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop*, 2008.

[8] P. Grun. Introduction to infiniband for end users. White paper, InfiniBand Trade Association.

[9] A. Gupta and M. Dejan. Evaluation of hpc applications on cloud. In *Proc. Open Cirrus Summit (OCS)*, 2011.

[10] A. Gupta, L. V. Kale, D. Milojicic, P. Faraboschi, and S. M. Balle. Hpc-aware vm placement in infrastructure clouds. In *Proc. of the 2013 IEEE International Conference on Cloud Engineering*, 2013.

[11] A. Gupta, O. Sarood, L. V. Kale, and D. Milojicic. Improving hpc application performance in cloud through dynamic load balancing. In *IEEE/ACM Int'L Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, 2013.

[12] J. Huang, X. Ouyang, J. Jose, M. Wasi-ur Rahman, H. Wang, M. Luo, H. Subramoni, C. Murthy, and D. K. Panda. High-performance design of hbase with rdma over infiniband. In *Proc. IEEE 26th International Symposium on Parallel and Distributed Processing(IPDPS)*, pages 774–785, 2012.

[13] K. Hwang, G. C. Fox, and J. J. Dongarra. *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*. Morgan Kaufmann, 2012.

[14] A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema. Performance analysis of cloud computing services for many-tasks scientific computing. *IEEE Transactions on Parallel and Distributed Systems*, 22(6):931–945, 2011.

[15] J. Jose, M. Li, X. Lu, K. C. Kandalla, M. D. Arnold, and D. K. Panda. Sr-iov support for virtualization on infiniband clusters: Early experience. In *Proc. IEEE/ACM 13th International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, 2013.

[16] M. J. Koop, J. Sridhar, and D. K. Panda. Tupleq: Fully-asynchronous and zero-copy mpi over infiniband. In *Proc. IEEE International Symposium on Parallel Distributed Processing (IPDPS)*, pages 1–8, 2009.

[17] A. Marathe, R. Harris, D. K. Lowenthal, B. R. de Supinski, B. Rountree, M. Schulz, and X. Yuan. A comparative study of high-performance computing on the cloud. In *Proc. Int'l Symposium on High-performance Parallel and Distributed Computing (HPDC)*, 2013.

[18] L. Ramakrishnan, R. S. Canon, K. Muriki, I. Sakrejda, and N. J. Wright. Evaluating interconnect and virtualization performance for high performance computing. In *Proc. Int'l Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computing Systems*, 2011.

[19] J. R. Santos, Y. Turner, G. Janakiraman, and I. Pratt. Bridging the gap between software and hardware techniques for i/o virtualization. In *Proc. USENIX Annual Technical Conference (ATC)*, pages 29–42, 2008.

[20] J. Shafer. I/o virtualization bottlenecks in cloud computing today. In *Proc. Conference on I/O Virtualization*, 2010.

[21] M. Shin, K. Nam, and H. Kim. Software-defined networking (sdn): A reference architecture and open apis. In *Proc. IEEE International Conference on ICT Convergence (ICTC)*, pages 360–361, 2012.

[22] J. Vienne, J. Chen, M. W. Rahman, N. S. Islam, H. Subramoni, and D. K. Panda. Performance analysis and evaluation of infiniband fdr and 40gige roce on hpc and cloud computing systems. In *Proc. IEEE 20th Annual Symposium on High-Performance Interconnects (HOTI)*, 2012.