Performance Characterization of Hadoop Workloads on SR-IOV-enabled Virtualized InfiniBand Clusters [*]

Shashank Gugnani
Department of Computer Science and Engineering
The Ohio State University
Columbus, OH USA 43210
gugnani.2@osu.edu

Xiaoyi Lu
Department of Computer Science and Engineering
The Ohio State University
Columbus, OH USA 43210
lu.932@osu.edu

Dhabaleswar K. (DK) Panda
Department of Computer Science and Engineering
The Ohio State University
Columbus, OH USA 43210
panda.2@osu.edu

ABSTRACT

Big Data Systems are becoming increasingly complex and generally have very high operational costs. Cloud computing offers attractive solutions for managing large scale systems. However, one of the major bottlenecks in VM performance is virtualized I/O. Since Big Data applications and middleware rely heavily on high performance interconnects such as InfiniBand, the performance of virtualized InfiniBand interfaces is vital. Single Root I/O Virtualization (SR-IOV) is a hardware based approach which offers significant performance benefits as compared to software based I/O virtualization. With the increasing adoption of InfiniBand network for cloud computing, it is important to evaluate the performance benefits of SR-IOV for InfiniBand networks; especially to see the performance characteristics of Big Data applications and middleware under different scenarios. We characterize the main performance factors for different workloads through this study (such as map task scheduling, I/O, data replication, etc.). Our experimental evaluations show that the performance difference for a wide set of Big Data benchmarks and applications over SR-IOV with InfiniBand using RDMA-enabled Hadoop as compared to native InfiniBand network is just 5 - 15%. In addition, with RDMA-enabled Hadoop, we see 20.9 - 81.6% performance improvement for RDMA as compared to IPoIB.

Keywords

Virtualization, SR-IOV, Hadoop, Big Data, InfiniBand

1. INTRODUCTION

Over the years, there has been a tremendous rise in the demand for computational power. To meet this ever-growing demand, modern High-Performance Computing (HPC) clusters and Big Data Systems have become very complex and large in size. With the prevalence of high-speed networking interconnects and multi-core processors, efficient sharing of resources is becoming more and more important. Also, a large number of users, particularly enterprise users, experience highly variable workloads. For such users, cloud-based systems and clusters are attractive options that offer scalable, reliable and flexible services [20]. Cloud computing relies on sharing of resources to provide coherence and save power and money.

Although several improvements have been made in virtualization technology over the years [17], [39], running Big Data applications on VMs still remains a major challenge. This is because while the current virtualization technology can deliver near native CPU performance, the I/O performance in virtualized systems is a major bottleneck [35]. Since many applications and middleware in the Big Data community rely heavily on the features and performance offered by modern networking interconnects, the I/O performance in virtualized environments will be a major driver in the adoption of cloud computing for Big Data applications.

I/O virtualization technologies can broadly be classified as either hardware-based or software-based. In software-based approaches [36], the virtual machine monitor (VMM) usually emulates the network interface controller (NIC) to provide virtualized I/O access points to the user. In such approaches, the VM cannot directly access the physical device and must go through the VMM. This causes a lot of context-switching and results in significant performance degradation. Many approaches have been proposed to allow the VM to directly access physical devices [32], [9]. Such hardware-based approaches completely bypass the VMM and can potentially deliver higher performance. Single-Root I/O Virtualization (SR-IOV) [5] is one such approach. With SR-IOV, a PCI Express (PCIe) device can present itself as multiple virtual devices, where each virtual device can be assigned to a VM. Recent studies [18], [26] have shown that SR-IOV is significantly better than software-based approaches and can deliver near native I/O performance.

In recent years, modern interconnects such as InfiniBand [4], have seen increased usage for HPC and Big Data Systems. InfiniBand offers several advantages over the traditional Ethernet technology, such as Remote Direct Memory Access (RDMA), low latency and high bandwidth. Moreover, socket based applications can also be run with InfiniBand hardware using the IP over InfiniBand (IPoIB) protocol [15]. Apache Hadoop [1] is one of the most popular open-source Big Data stacks currently available. In the last few years, it has become the de-facto package for analysis of Big Data. RDMA-Hadoop [6], based on Apache Hadoop, is a publicly available stack for InfiniBand clusters. It provides RDMA

[*] This research is supported in part by National Science Foundation grants #CNS-1419123, #IIS-1447804, #ACI-1450440, #CNS-1513120, and #IIS-1636846.

BDCAT'16, December 06-09, 2016, Shanghai, China
© 2016 ACM. ISBN 978-1-4503-4617-7/16/12…$15.00
DOI: http://dx.doi.org/10.1145/3006299.3006313

enhanced designs in Hadoop. The emergence of I/O virtualization technologies like SR-IOV and the increased adoption of InfiniBand networks for cloud deployments leads us to a broad question: *Is SR-IOV support for InfiniBand networks ready for "Prime-Time" Big Data workloads?*

	Evaluation Platform			Hadoop Distribution	
	VM	**SR-IOV**	**InfiniBand**	**Hadoop**	**RDMA-Hadoop**
[19], [38], [31]	×	×	√	√	×
[33]	×	×	√	√	√
[13], [21], [41]	√	×	×	√	×
[37], [24]	√	√	√	×	×
This paper	√	√	√	√	√

Table 1: Comparison with existing studies

The performance characteristics of InfiniBand native hardware have been thoroughly evaluated by the Big Data community. However, performance evaluation of InfiniBand in virtualized environments with SR-IOV, has not been systematically carried out, particularly for Big Data applications. We summarize existing studies in this area in Table 1. Studies [19], [38], [31] present their evaluations with InfiniBand and Hadoop, but don't use either VMs or RDMA-Hadoop for their evaluations. One study [33] focuses on InfiniBand and Apache and RDMA-Hadoop. However, it presents no results with VMs. Several studies [13], [21], [41] use VMs and Apache Hadoop for evaluations, but do not consider SR-IOV or use InfiniBand. Some studies [37], [24] present evaluations with SR-IOV-enabled VMs on InfiniBand, but focus on MPI based workloads rather than Big Data workloads. Therefore, there exist relatively few evaluations which consider all aspects covered in Table 1. In this paper, we present an in-depth study of all these important aspects to understand the trade-offs and performance attributes of using SR-IOV on virtualized InfiniBand clusters for Big Data applications. To summarize, we address the following critical problems:

1. Big Data workloads are very diverse. What are the trade-offs and performance attributes when using SR-IOV, compared to native InfiniBand hardware, for different Big Data applications and data sizes?

2. Modern processors with multi-cores can run VMs with various subscription policies (VM per node, VM per socket, and VM per core). What is the impact of such policies on the performance of Big Data applications?

3. Socket based applications can run on InfiniBand hardware using IPoIB. What are the performance characteristics of benchmark applications for different InfiniBand communication modes (IPoIB and RDMA)?

We carry out multiple experiments to find answers to these critical problems.

The main contributions of this paper are as follows:

1. Provide a comprehensive evaluation of the performance of Hadoop workloads and applications on SR-IOV-enabled InfiniBand clusters

2. Understand the critical factors as well as the best configuration for the execution of different Hadoop workloads and applications in virtualized environments (Table 3)

3. Provide an answer to the question: Is SR-IOV support for InfiniBand networks ready for Hadoop workloads and applications?

Our evaluations show that the performance overhead of Big Data workloads with SR-IOV over InfiniBand is within 5 - 15% of native InfiniBand hardware performance. Also, with RDMA-Hadoop, we see 20.9 - 81.6% performance improvement for RDMA as compared to IPoIB.

The rest of this paper is organized as follows. Section 2 presents an overview of SR-IOV, InfiniBand, and Apache and RDMA-Hadoop. Section 3 describes the methodology used for evaluation and Section 4 presents our performance evaluation results. We discuss related work in Section 5 and conclude our work in Section 6.

2. BACKGROUND

In this section, we provide an overview of SR-IOV, InfiniBand, and Apache and RDMA-Hadoop.

2.1 Single Root I/O Virtualization (SR-IOV)

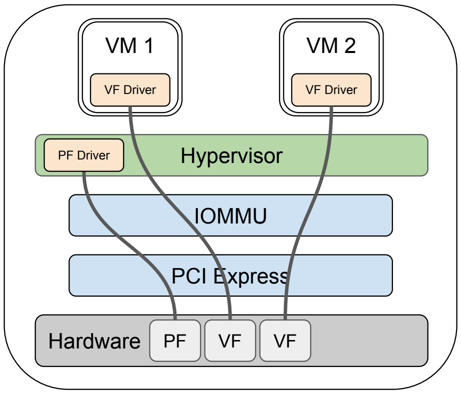


Figure 1: Overview of SR-IOV

Single Root I/O Virtualization (SR-IOV) [5] is a standard for PCI Express (PCIe), which specifies the native I/O virtualization capabilities in PCIe adapters. By using SR-IOV, a single physical device or Physical Function (PF), can be presented as multiple virtual devices or Virtual Functions (VFs). As is evident by the solid lines in Figure 1, a single VM can be assigned a virtual device through PCI pass-through, which allows direct access to the VF from each VM. SR-IOV is a hardware-based approach for implementing I/O virtualization. Thus, the drivers of the PF can also be used for VFs, and its performance is generally higher than the traditional software-based I/O virtualization methods.

2.2 InfiniBand

InfiniBand [4] is a high-performance networking interconnect that is widely used for high-performance computing on supercomputers. The latest TOP500 [8] rankings released in June 2016 indicate that more than 40% of the top 500 supercomputers are using InfiniBand as their primary interconnect. Remote Direct Memory Access (RDMA), one of the main features of InfiniBand, allows a node to directly access the CPU memory of another remote node without any involvement from the remote node. InfiniBand communication is carried out in userspace and in a 'zero-copy' manner. In addition, InfiniBand uses hardware offload for all protocol processing. This results in low latency and high bandwidth communication. In addition, Traditional socket (TCP/IP) based applications can be run over InfiniBand hardware using the IP over InfiniBand (IPoIB) protocol.

2.3 Apache Hadoop

MapReduce [16] is a framework introduced by Google which is used for large-scale parallel processing of data. Apache Hadoop, an open-source implementation of the MapReduce framework, has become extremely popular in recent times for Big Data processing. The Apache Hadoop framework is composed of the following modules:

1. **Hadoop Common** contains utilities and libraries that are used by other modules.
2. **Hadoop Distributed File System (HDFS)** is a distributed filesystem which offers fault-tolerant and high-throughput access to data.
3. **Hadoop YARN** is a resource management module that also handles job scheduling.
4. **Hadoop MapReduce** is an implementation of the MapReduce framework for parallel processing of data.

2.4 RDMA-Hadoop

RDMA-Hadoop [6] is a publicly available package that is built on Apache Hadoop. It can be used to exploit the advantages of InfiniBand on RDMA-enabled clusters for Big Data applications. RDMA-Hadoop provides advanced designs for HDFS [23], MapReduce [40] and Remote Procedure Call (RPC) [27] components which are optimized for RDMA-enabled clusters and deliver high performance. The HDFS plugin supports multiple modes of operation: **HHH** - The default mode, **HHH-M** - Adds support for in-memory I/O operations, and **HHH-L** - For use with Lustre Filesystem. It also has policies which make efficient use of heterogeneous storage devices (SSD, HDD, RAM Disk, and Lustre). The MapReduce plugin has an advanced design that provides with RDMA-based shuffle, prefetching of map output and optimized overlapping of the different stages in MapReduce. The RPC plugin offers JVM-bypassed buffer management with smart buffer allocation and RDMA-based data transmission.

3. EVALUATION METHODOLOGY

In this section, we discuss the different factors of using InfiniBand in virtualized environments for Big Data. We use different dimensions for evaluating the performance characteristics of using SR-IOV with InfiniBand for Big Data, as shown in Figure 2. Experimental results for these dimensions are presented in Section 4.

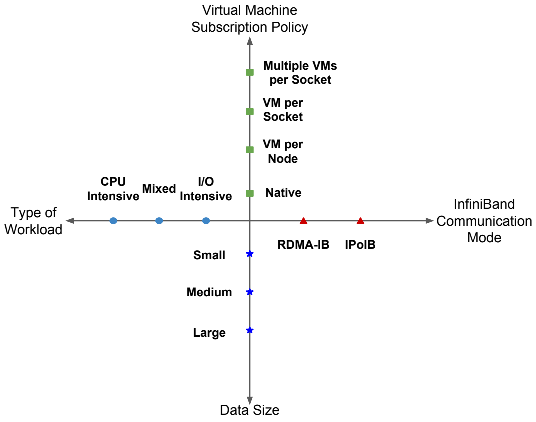


Figure 2: Evaluation Dimensions

3.1 Virtual Machine Subscription Policy

Multi-core architectures allow for the execution of multiple parallel jobs to improve CPU utilization and network resources. In such cases, using virtual machines can offer performance isolation and easy resource management. However, the performance characteristics can vary significantly based on the virtual machine subscription policy. In Section 4.2, we present the performance characteristics of running different number of VMs per node. Specifically, we evaluate the performance for VM per CPU node, VM per CPU socket (NUMA-node), and multiple VMs per CPU socket configurations.

3.2 InfiniBand Communication Mode

As introduced in Section 2.2, InfiniBand offers multiple communication modes. The performance of these modes has been widely evaluated with native InfiniBand hardware. In Section 4.4, we present the performance evaluation of different communication modes for Big Data workloads with SR-IOV over InfiniBand. Specifically, we evaluate the performance of RDMA and IPoIB modes on SR-IOV and compare it to that of native hardware.

3.3 Data Size

Volume (or data size) is one of the most important aspects of Big Data (hence the name Big Data), which is generally ignored during performance evaluations. For our performance evaluations, we use 3 data sizes - Small, Medium, and Large. For TestDFSIO, the values of these sizes are 48 GB, 72 GB, and 96 GB, respectively. For all other benchmarks, the values are 20 GB, 40 GB, and 60 GB, respectively.

3.4 Type of Big Data Workload

Big Data Applications are very diverse in terms of their CPU and I/O utilization. Since SR-IOV is an approach to improve I/O performance, the comparison of different workload execution using SR-IOV is important to analyze. We select different types of workloads (CPU Intensive, I/O Intensive and Mixed) and applications, and evaluate their performance trade-offs with SR-IOV as compared to native hardware. The workloads and applications we have used are presented in Table 2. Results are presented in Sections 4.3 and 4.5.

4. PERFORMANCE EVALUATION

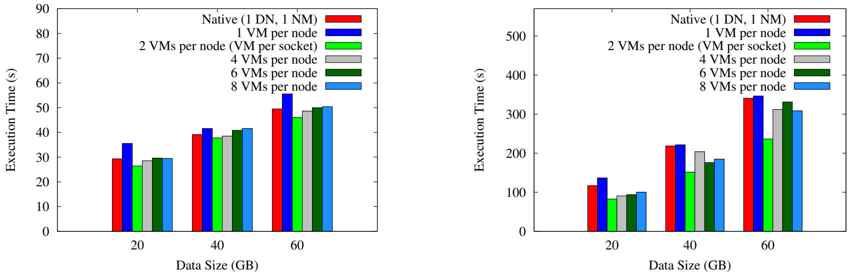
4.1 Experiment Setup

Our testbed consists of nine physical nodes on the Chameleon Cloud [2], where each node has a 24-core 2.3 GHz Intel Xeon E5-2670 (Haswell) processor with 128 GB main memory and equipped with Mellanox ConnectX-3 FDR (56 Gbps) HCAs and PCI Gen3 interfaces. We use CentOS Linux 7.1.1503 (Core) with kernel 3.10.0-229.el7.x86_64. In addition, we use the Mellanox OpenFabrics Enterprise Distribution MLNX_OFED_LINUX-3.0-1.0.1 to provide the InfiniBand interface with SR-IOV support, OpenJDK 1.7.0_91 as the Java package, and KVM as the Virtual Machine Monitor (VMM). For consistency, we use the same OS and software versions for the virtual machines as well.

For efficiently building a large scale Big Data Cloud Testbed using bare metal InfiniBand nodes, we created a dedicated appliance, as shown in Figure 3. The appliance has IOMMU and SR-IOV enabled by default for hardware-based virtualization. It also has the Open Fabrics Enterprise Distribution (OFED) [30] stack and all necessary virtualization packages pre-installed. Inside this appliance, a small VM image is provided along with a VM launch script which launches VMs on all necessary bare metal nodes and runs an

Benchmark/Application	Type of workload	Description
TestDFSIO (Read)	I/O Intensive (Read)	Benchmark for testing the read throughput of the Hadoop cluster
TeraGen	I/O Intensive (Write)	Benchmark to generate desired rows of data to be later used by TeraSort
TeraSort	Mixed	Benchmark to sort rows of data
Sort	Mixed	Benchmark to sort text input data
Wordcount	CPU Intensive	Benchmark to count the occurrences of each word in multiple text files
CloudBurst [34]	-	MapReduce-based read-mapping algorithm optimized for mapping of sequence data
MR-MSPolygraph [25, 14]	-	MapReduce-based implementation of algorithm for peptide identification from mass spectrometry data
Self-join [10]	-	MapReduce-based algorithm for generating association among given fields

Table 2: Benchmarks and Applications used for evaluation



(a) TeraGen (RDMA)

(b) TeraSort (RDMA)

Figure 4: Impact of Virtual Machine Subscription Policy

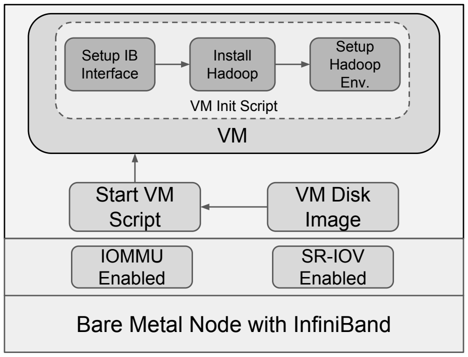


Figure 3: RDMA-Hadoop Appliance

initialization script inside each VM on launch. This initialization script configures InfiniBand, installs Hadoop and then configures the necessary environment inside the VM to run Hadoop jobs. This script also configures hostnames, DNS resolution files, and ssh to ensure Hadoop works correctly. In addition, these scripts have been written to take advantage of the large number of cores in modern processors by launching and configuring VMs in parallel, which greatly reduces the system setup time.

Using this appliance, users can easily setup a Big Data cloud system to run Hadoop applications. The user just has to launch bare metal nodes using the appliance image and then run the launch script on one of the nodes. This appliance is publicly available on the Chameleon Cloud [7].

We have used the standard benchmark suite that comes with Apache Hadoop (v 2.7.1) for our experiments. All benchmarks are run using RDMA-Hadoop 0.9.9 (based on Apache Hadoop 2.7.1). The results have been averaged over three runs to ensure a fair comparison.

All experiments are performed on the same physical set of nodes with a total of 96 maps and 48 reduces. 70% of the RAM disk is used for data storage. HDFS block size is kept to 256 MB. The NameNode runs on a different node of the Hadoop cluster and the benchmark is run in the NameNode. Each NodeManager is configured to assign a minimum of 4 GB memory per container. This ensures that in all cases the total physical resources used are the same.

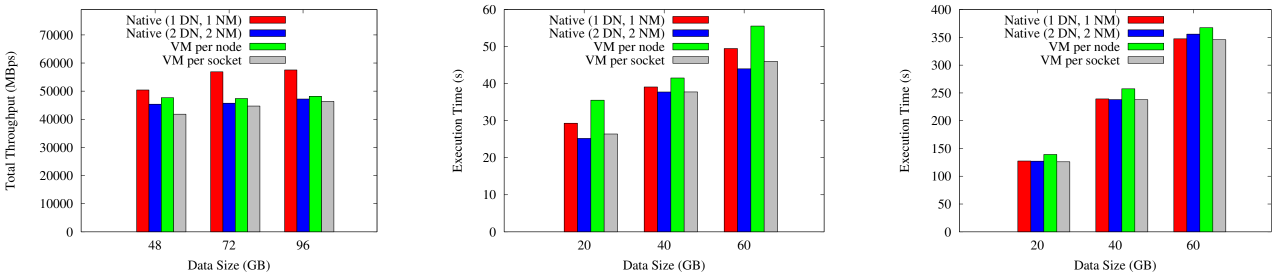
All native experiments are performed with 8 physical nodes. Each node has a single 230 GB HDD. We evaluate 2 cases for native experiments - 1 DataNode, 1 NodeManager per node, which is the default case, and 2 DataNodes, 2 NodeManagers per node (henceforth referred to as Native (1 DN, 1 NM) and Native (2 DN, 2 NM), respectively). Each NodeManager is configured to run with 12 and 6 concurrent containers, respectively. For all VM experiments, we use the same nodes as the native experiments. We also make sure that the total number of containers launched per physical node is 12 for all cases.

4.2 Impact of Virtual Machine Subscription Policy

To understand the performance characteristics of running multiple VMs per node, we conduct a VM subscription policy study for TeraGen and TeraSort. We go from 1 VM per node to 8 VMs per node, and compare the performance with Native (1 DN, 1NM) case. Figure 4 shows the results of this study. We observe that with 2 VMs per node (VM per socket), we achieve the best performance. Increasing the number of VMs beyond that only decreases the performance. This is because the overhead of running multiple VMs outweighs the benefits of having multiple VMs per node. Since the VM per node configuration has the least overhead for VM deployments, and the performance of VM per socket is the best among all cases, we only use VM per node and VM per socket VM configurations for all further experiments. For VM per socket, on each physical node, 2 NodeManagers, and 2 DataNodes are running. To see if this is the reason that VM per socket performance is the best, we compare both Native (1 DN, 1 NM) and Native (2 DN, 2 NM) performance with the performance of different VM configurations.

4.3 Impact of type of Workload

In this section, we present the evaluation results for different Hadoop workloads. We evaluate the performance of different Hadoop

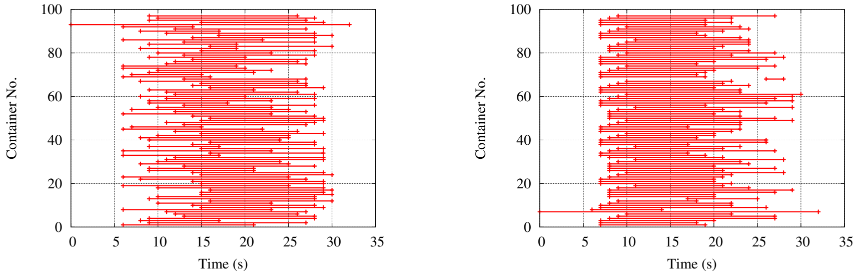


(a) TestDFSIO Read Throughput (RDMA)

(b) TeraGen Execution Time (RDMA)

(c) Wordcount Execution Time (RDMA)

Figure 5: Performance Characterization of CPU and I/O Intensive Workloads on SR-IOV enabled InfiniBand Clusters



(a) Native (1 DN, 1 NM)

(b) VM per socket

Figure 6: Container Launch Time Analysis for TeraGen - 60 GB (RDMA)

benchmarks for VM per host node and VM per host CPU socket configurations. We compare the results with those of Native (1 DN, 1 NM) and Native (2 DN, 2 NM) configurations. We choose TestDFSIO, TeraGen, TeraSort, Sort and Wordcount benchmarks as representatives of different Big Data workloads. TestDFSIO and TeraGen are I/O intensive workloads, while Wordcount is CPU intensive, and Sort and TeraSort are mixed workloads. The performance evaluation results are presented in Figures 5 and 7. In all cases, we observe that the VM per node configuration is only slightly worse than Native (1 DN, 1 NM) configuration (about 15% or less).

4.3.1 CPU Intensive Workloads

Results for Wordcount are presented in Figure 5(c). We observe that Native (1 DN, 1 NM), Native (2 DN, 2 NM), and VM per socket performance is very similar. VM per node performance is only slightly worse that Native (1 DN, 1 NM). This is because Wordcount is a CPU intensive workload, thus the performance of different configurations will be similar.

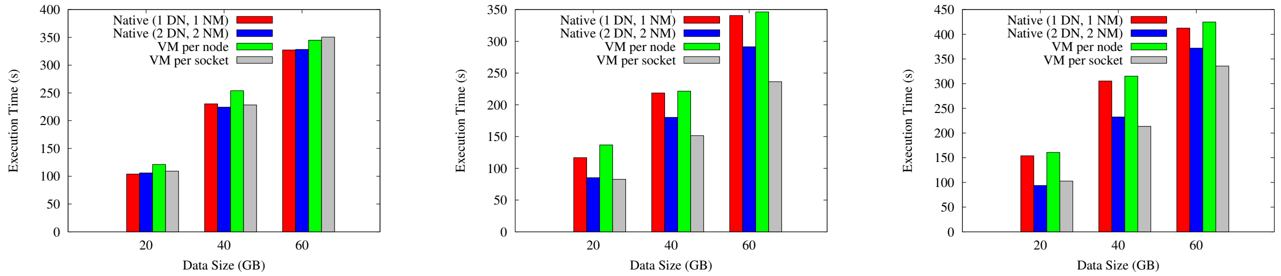
4.3.2 I/O Intensive Workloads

The performance evaluation results are presented in Figure 5(a) and Figure 5(b). We notice that compared to Native (1 DN, 1 NM) performance, VM per socket performance is better for TeraGen and worse for TestDFSIO Read. We also observe that Native (2 DN, 2 NM) performance is comparable to VM per socket case. This is because, for both cases, the same number of Hadoop daemons are running per node. For TestDFSIO Read, most of the reads are local, which is why running multiple DataNodes and NodeManagers on each node reduces performance. On the contrary, this increases performance for TeraGen. This is because the data generated needs to be replicated to multiple nodes, which can be done faster when running 2 DataNodes per node as more daemons are available to send data to other DataNodes.

For VM per socket and Native (2 DN, 2 NM) configurations, 2 NodeManagers are running per physical node and 1 NodeManager is running for the Native (1 DN, 1 NM) configuration. To see how this impacts performance, we analyzed the container launch times for TeraGen (60 GB). Figure 6 shows the results of this analysis. In each graph, we show the running time for each container launched by the Hadoop framework. The first point on each line denotes the container launch time and the second point denotes the container completion time. The line that starts from time 0 signifies the Application Master Container. The running time for this container gives us an estimate of the runtime of the Hadoop application. We observe that although the average container runtime for Native configuration is smaller than that of VM per socket configuration, the containers are launched much faster for the latter case. All containers are launched and running by the 11^{th} second for VM per socket configuration, while containers are launched as late as the 17^{th} second for Native (1 DN, 1 NM) configuration. In addition, most containers complete earlier for VM per socket configuration. This also explains why Native (2 DN, 2 NM) and VM per socket performance is better than or similar to Native (1 DN, 1 NM) performance for most cases.

4.3.3 Mixed Workloads

From Figure 7(a) we see that for Sort, the performance of Native (1 DN, 1 NM), Native (2 DN, 2 NM) and VM per socket is comparable. However, for TeraSort (Figure 7(b)) we see that VM per socket performance is the best and Native (2 DN, 2 NM) is better than Native (1 DN, 1 NM). To see whether these results were because of the RDMA-enhanced designs in RDMA-Hadoop, we also analyzed the performance of TeraSort for IPoIB (Figure 7(c)). We observe the same trend for IPoIB as well. Thus, we can conclude that some system level factor is the reason for this trend. To further

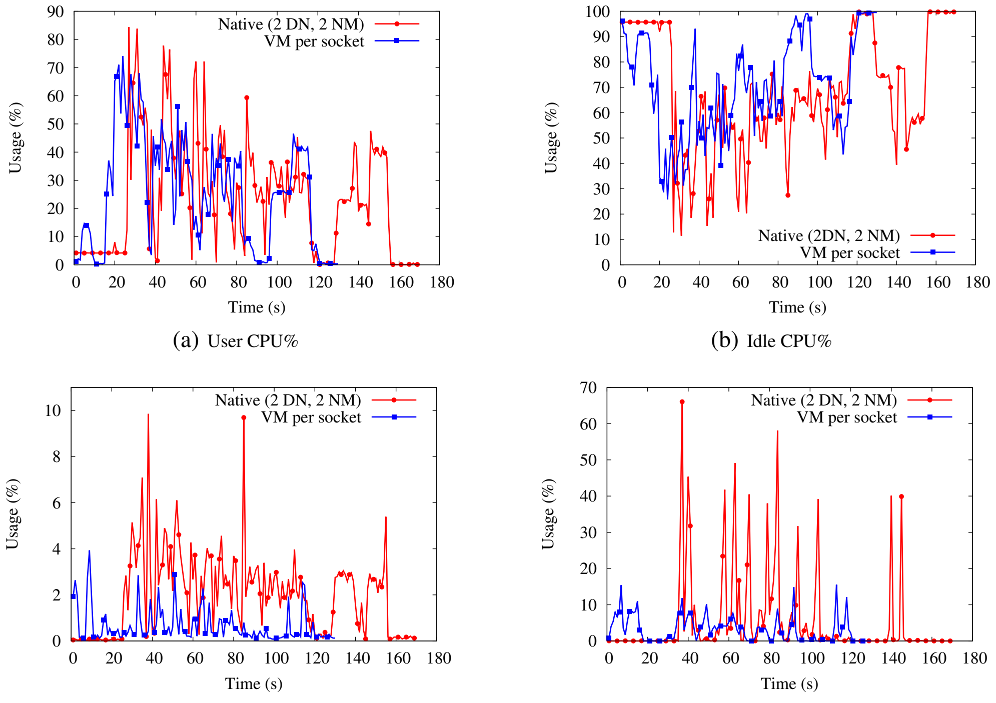


(a) Sort Execution Time (RDMA)

(b) TeraSort Execution Time (RDMA)

(c) TeraSort Execution Time (IPoIB)

Figure 7: Performance Characterization of Mixed Workloads on SR-IOV enabled InfiniBand Clusters



(a) User CPU%

(b) Idle CPU%

(c) System CPU%

(d) I/O CPU%

Figure 8: CPU Utilization Analysis for TeraSort - 40 GB (RDMA)

investigate this reason for this trend, we conduct a CPU utilization analysis for TeraSort with Native (2 DN, 2 NM) and VM per socket configurations. Figure 8 shows the results of this analysis. We can see that the main reason that performance of Native (2 DN, 2 NM) is worse than VM per socket configuration is I/O wait time. As discussed in [3], the hypervisor does buffering for I/O in VMs, which is why I/O in VMs is more efficient than in the Native mode for this case. We also observe that system CPU usage is higher for Native (2 DN, 2 NM). This is because the CPU usage statistics were taken on the physical hosts and the system usage inside the VMs is reported as user CPU usage by the host. We did a similar analysis for Sort, and we observed the same trend as for TeraSort. However, since Sort uses a replication factor of 3 and TeraSort uses a replication factor of 1, there is significantly more I/O involved for Sort. This reduces the difference between native and VM I/O performance and thus for Sort, the performance of different configuration modes is comparable.

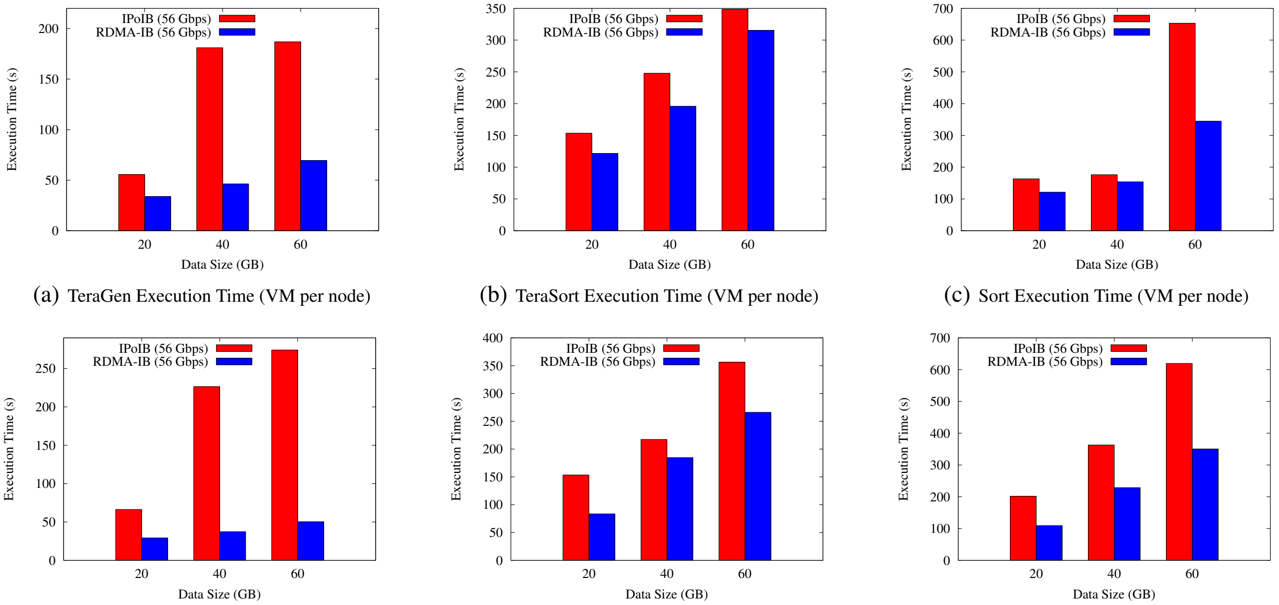
These trends indicate that by carefully selecting the VM subscription policy, we can achieve near native performance, and in some cases even better than native performance.

4.4 Impact of InfiniBand Communication Mode

Traditional socket (TCP/IP) based applications can be run over InfiniBand hardware using a protocol known as IP over InfiniBand (IPoIB). In this section, we present performance evaluation results when using RDMA compared to IPoIB for virtual environments with SR-IOV.

Figure 9 shows the results of our evaluation. We notice that RDMA is better than IPoIB for all benchmarks. For Sort and TeraGen, we see significant performance difference between RDMA and IPoIB. The maximum improvement we see is 74.3% for Sort, 20.9% for TeraSort, and 47.1% for Sort with VM per node. The maximum improvements for VM per socket are 81.6% for Sort, 45.6% for TeraSort, and 45.9% for Sort.

From Figures 5 and 7, we see that the performance difference between Native (1 DN, 1 NM) and VM per node modes is the most



(a) TeraGen Execution Time (VM per node)

(b) TeraSort Execution Time (VM per node)

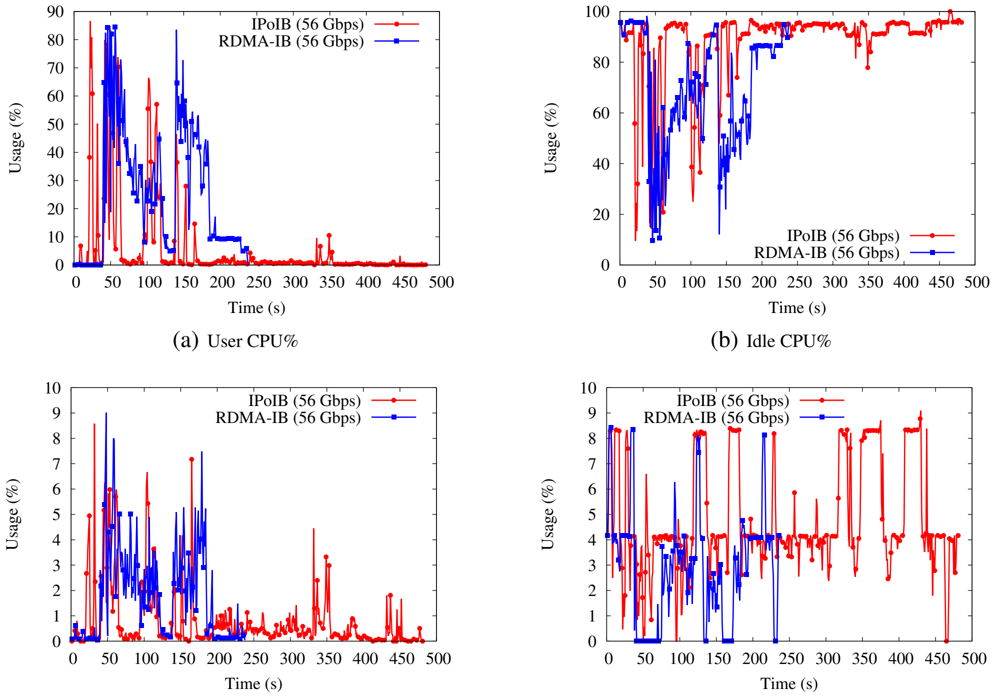
(c) Sort Execution Time (VM per node)

(d) TeraGen Execution Time (VM per socket)

(e) TeraSort Execution Time (VM per socket)

(f) Sort Execution Time (VM per socket)

Figure 9: Impact of InfiniBand Communication Mode (IPoIB v/s RDMA-IB)



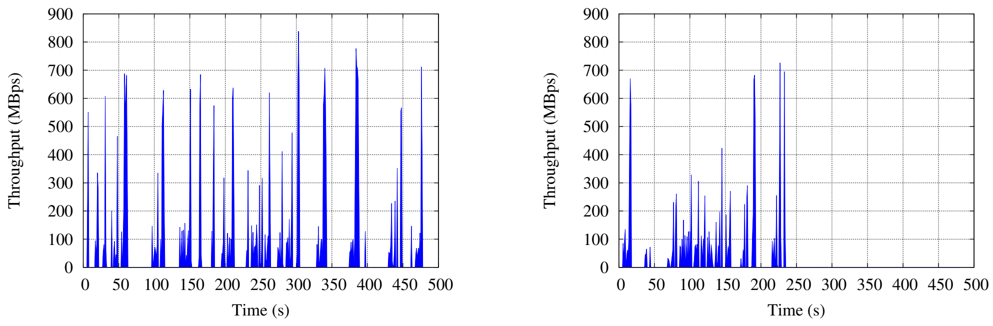
(a) User CPU%

(b) Idle CPU%

(c) System CPU%

(d) I/O CPU%

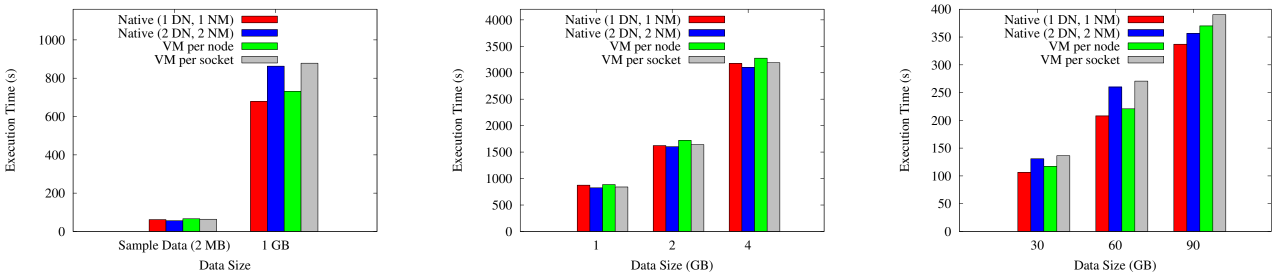
Figure 10: CPU Utilization Analysis for Sort - 60 GB (VM per node)



(a) IPoIB

(b) RDMA-IB

Figure 11: Disk I/O Analysis for Sort - 60 GB (VM per node)



(a) CloudBurst

(b) MR-MSPolygraph

(c) Self-join

Figure 12: Application Level Evaluations

for TestDFSIO and TeraGen, especially for the largest data size. For all other workloads, the size of data doesn't affect the performance of different modes. This trend indicates that disk I/O is the main bottleneck for virtual machines when compared with native execution.

We also observe that for large data sizes, there is a significant performance difference between IPoIB and RDMA for TeraGen and Sort. To see where RDMA is winning over IPoIB, we did a performance analysis for Sort (60 GB). Figures 10 and 11 show the CPU utilization analysis and disk I/O analysis, respectively. We see that the user CPU utilization is very high for RDMA as compared to IPoIB. This is mainly because we use polling for receiving messages when using RDMA, which increases the user CPU utilization and increases performance. Considering Disk I/O, we observe that for RDMA, there is significantly less I/O than IPoIB. RDMA-Hadoop uses optimized storage policies for HDFS in RDMA mode [22], which try to keep important data in memory and make use of RAM Disk for storing some of the application data, thereby reducing the amount of Disk I/O required.

4.5 Evaluation with Applications

To see whether running Hadoop on SR-IOV-enabled InfiniBand clusters is actually practical, we did evaluations at the application level. We evaluated with CloudBurst, MR-MSPolygraph, and Self-join which are MapReduce-based applications widely used by scientists. Figure 12 shows the results of our evaluation. For all applications, we observe that the VM per node performance is only slightly worse than Native (1 DN, 1 NM) (only 7.1%, 3.4%, and 8.7% overhead for CloudBurst, MR-MSPolygraph, and Self-join, respectively). In addition, Native (2 DN, 2 NM) performance is comparable to VM per socket performance. This indicates that SR-IOV enabled InfiniBand clusters can be used for Big Data applications with minimal overhead.

4.6 Summary of Evaluation

Table 3 shows the summary of our evaluation of the different benchmarks and applications. For TestDFSIO (Read), Native (1 DN, 1 NM) gives the best performance. Overall, we observe that with multiple DataNodes and NodeManagers per physical node, we get better I/O and task scheduling performance. However, the CPU performance suffers because of the additional overhead of running multiple DataNodes and NodeManagers. This leads us to conclude that I/O is the main bottleneck, which makes sense, since all reads are local. However, for Teragen, we observe that Native (2 DN, 2 NM) provides the best performance. Here, the data needs to be replicated to multiple nodes. Thus, the faster we can start generating the data and copying it, the better our execution time will be, which is why the factor most affecting performance is map task scheduling and data replication. VM per socket performance is the best for TeraSort. This implies that data replication is the critical factor here, since, we have multiple DataNodes running on each physical node, and TeraSort requires data to be replicated to multiple nodes. Both the Native cases give the best performance for Sort and Wordcount. Since, adding more DataNodes and NodeManagers does not improve performance here, we conclude that the main bottleneck here is CPU performance. For applications MR-MSPolygraph and Self-join, Native (2 DN, 2 NM) gives the best performance, and VM per socket performance is better than VM per node performance. So, the factor most affecting performance here is map/reduce task scheduling and data replication. For CloudBurst, Native (1 DN, 1 NM) and VM per node deliver the best performance for Native and VM Modes, respectively. Thus, CPU

Benchmark/Application	Type of workload	Factor most critical to performance	Native Mode with best performance	VM Mode with best performance	% Overhead
TestDFSIO (Read)	I/O Intensive (Read)	I/O	Native (1 DN, 1 NM)	VM per node	12.8%
TeraGen	I/O Intensive (Write)	Map Task Scheduling & Data Replication	Native (2 DN, 2 NM)	VM per node	3%
TeraSort	Mixed	Data Replication	Native (2DN, 2 NM)	VM per socket	-12.5%
Sort	Mixed	Raw CPU Performance	Native (both cases)	VM per socket	4.5%
Wordcount	CPU Intensive	Raw CPU Performance	Native (both cases)	VM per socket	0.3%
CloudBurst	-	Raw CPU Performance	Native (1 DN, 1 NM)	VM per node	7.1%
MR-MSPolygraph	-	Map/Reduce Task Scheduling & Data Replication	Native (2 DN, 2 NM)	VM per socket	2.4%
Self-join	-	Map/Reduce Task Scheduling & Data Replication	Native (2 DN, 2 NM)	VM per socket	8.7%

Table 3: Evaluation Summary of Benchmarks and Applications

performance is the main factor for this application. Although Native performance is the best in most cases, the overhead with the VM modes is minimal.

5. RELATED WORK

I/O virtualization technologies have been widely studied and evaluated under different scenarios. Studies [28], [11] present performance evaluations of different software-based approaches using the Xen virtualization environment [12]. There have been several studies [18], [26], [29] that demonstrate the superiority of SR-IOV as compared to software-based approaches for Ethernet and InfiniBand networks. However, there are relatively few studies which focus on SR-IOV, InfiniBand and Big Data.

Fadika et al. [19] explore the performance of Hadoop for data intensive scientific applications with IPoIB. They present the impact of network, file system and programming modes on application performance. Another study by Saxena et al. [33] presents an in-depth study of Hadoop and RDMA-Hadoop. They consider performance trade-offs when using SSDs for data storage as compared to HDDs. Jose et al. [24] and Tatineni et al. [38] evaluate the performance of MPI collectives and point-to-point operations on SR-IOV-enabled InfiniBand clusters. Results show that by using SR-IOV, near native performance can be achieved. A white paper by VMWare [13] presents a benchmarking case study of Hadoop on virtual machines. Their results show that in some cases, by tuning the number of VMs per node, better than native performance can be achieved.

None of the studies mentioned before present systematic evaluations for Big Data applications using SR-IOV on InfiniBand networks. Therefore, our study is unique in terms of providing a comprehensive performance evaluation of Hadoop in a virtualized cluster environment using SR-IOV and InfiniBand.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented performance evaluation results of running Big Data workloads on SR-IOV-enabled virtualized InfiniBand clusters. We explored different dimensions for evaluating performance such as virtual machine configuration, data size, type of communication mode, and type of workload. We also presented system performance evaluation for some cases to gain deeper insight into the performance characteristics of these workloads.

Our experimental evaluations show that the performance of Big Data workloads and applications over SR-IOV with InfiniBand is comparable to that of native InfiniBand hardware, with an overhead of less than 15%. Also, by carefully selecting the right virtual machine configuration mode, we can get near native performance and in some cases even better than native performance for SR-IOV over InfiniBand. In the future, we plan to evaluate with larger data and cluster sizes with more diverse workloads. We also plan to use more Big Data benchmarks and applications to carry out additional performance evaluations on SR-IOV-enabled virtualized InfiniBand clusters.

7. REFERENCES

[1] Apache Hadoop. http://www.hadoop.apache.org.

[2] Chameleon. http://chameleoncloud.org/.

[3] Comparing Filesystem performance in Virtual Machines. http://mitchellh.com/comparing-lesystem-performance-in-virtual-machines.

[4] InfiniBand Trade Association. http://www.infinibandta.com.

[5] PCI-SIG Single-Root I/O Virtualization Specification. http://www.pcisig.com/specifications/iov/.

[6] RDMA Hadoop. http://hibd.cse.ohio-state.edu/overview/.

[7] RDMA-Hadoop Appliance. https://www.chameleoncloud.org/appliances/17/.

[8] TOP500 Supercomputing Sites. http://www.top500.org/.

[9] D. Abramson, J. Jackson, S. Muthrasanallur, G. Neiger, G. Regnier, R. Sankaran, I. Schoinas, R. Uhlig, B. Vembu, and J. Wiegert. Intel Virtualization Technology for Directed I/O. *Intel technology journal*, 10(3), 2006.

[10] F. Ahmad, S. Lee, M. Thottethodi, and T. Vijaykumar. PUMA: Purdue MapReduce Benchmarks Suite. 2012.

[11] P. Apparao, S. Makineni, and D. Newell. Characterization of Network Processing Overheads in Xen. In *Proceedings of the 2nd international Workshop on Virtualization Technology in Distributed Computing*, page 2. IEEE Computer Society, 2006.

[12] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the Art of Virtualization. *ACM SIGOPS Operating Systems Review*, 37(5):164–177, 2003.

[13] J. Buell. A Benchmarking Case Study of Virtualized Hadoop Performance on VMware vSphere 5. *technical white paper. VMware, Inc*, 2011.

[14] W. R. Cannon, M. M. Rawlins, D. J. Baxter, S. J. Callister, M. S. Lipton, and D. A. Bryant. Large Improvements in MS/MS-Based Peptide Identification Rates Using a Hybrid Analysis. *Journal of proteome research*, 10(5):2306–2317, 2011.

[15] J. Chu and V. Kashyap. Transmission of IP over InfiniBand (IPoIB). Technical report, 2006.

[16] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, OSDI'04, pages 10–10, Berkeley, CA, USA, 2004. USENIX Association.

[17] A. M. Devices. AMD, Secure Virtual Machine Architecture Reference Manual, 2005.

[18] Y. Dong, X. Yang, J. Li, G. Liao, K. Tian, and H. Guan. High Performance Network Virtualization with SR-IOV. *Journal of Parallel and Distributed Computing*, 72(11):1471 – 1480, 2012. Communication Architectures for Scalable Systems.

[19] Z. Fadika, M. Govindaraju, R. Canon, and L. Ramakrishnan. Evaluating Hadoop for Data-Intensive Scientific Operations. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 67–74. IEEE, 2012.

[20] R. Grossman. The Case for Cloud Computing. *IT Professional*, 11(2):23–27, March 2009.

[21] S. Ibrahim, H. Jin, L. Lu, L. Qi, S. Wu, and X. Shi. Evaluating MapReduce on Virtual Machines: The Hadoop Case. In *Cloud Computing*, pages 519–528. Springer, 2009.

[22] N. S. Islam, X. Lu, M. Wasi-ur Rahman, D. Shankar, and D. K. Panda. Triple-H: A Hybrid Approach to Accelerate HDFS on HPC Clusters with Heterogeneous Storage Architecture. In *Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on*, pages 101–110. IEEE, 2015.

[23] N. S. Islam, M. W. Rahman, J. Jose, R. Rajachandrasekar, H. Wang, H. Subramoni, C. Murthy, and D. K. Panda. High Performance RDMA-based Design of HDFS over InfiniBand. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, page 35. IEEE Computer Society Press, 2012.

[24] 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 *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, pages 385–392. IEEE, 2013.

[25] A. Kalyanaraman, W. R. Cannon, B. Latt, and D. J. Baxter. MapReduce Implementation of a Hybrid Spectral Library-Database Search Method for Large-Scale Peptide Identification. *Bioinformatics*, 27(21):3072–3073, 2011.

[26] J. Liu. Evaluating Standard-Based Self-Virtualizing Devices: A Performance Study on 10 GBE NICs with SR-IOV Support. In *Parallel Distributed Processing (IPDPS), 2010 IEEE International Symposium on*, pages 1–12, April 2010.

[27] X. Lu, N. S. Islam, M. Wasi-ur Rahman, J. Jose, H. Subramoni, H. Wang, and D. K. Panda. High-Performance Design of Hadoop RPC with RDMA over InfiniBand. In *Parallel Processing (ICPP), 2013 42nd International Conference on*, pages 641–650. IEEE, 2013.

[28] A. Menon, J. R. Santos, Y. Turner, G. J. Janakiraman, and W. Zwaenepoel. Diagnosing Performance Overheads in the Xen Virtual Machine Environment. In *Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments*, pages 13–23. ACM, 2005.

[29] M. Musleh, V. Pai, J. Walters, A. Younge, and S. Crago. Bridging the Virtualization Performance Gap for HPC Using SR-IOV for InfiniBand. In *Cloud Computing (CLOUD), 2014 IEEE 7th International Conference on*, pages 627–635, June 2014.

[30] Open Fabrics Enterprise Distribution. http://www.openfabrics.org/.

[31] N. Poggi and D. Carrera. Evaluating the Impact of SSDs and InfiniBand in Hadoop Cluster Performance and Costs. *technical white paper. Barcelona Supercomputing Center*, 2015.

[32] J. R. Santos, Y. Turner, G. J. Janakiraman, and I. Pratt. Bridging the Gap between Software and Hardware Techniques for I/O Virtualization. In *USENIX Annual Technical Conference*, pages 29–42, 2008.

[33] P. Saxena and P. Kumar. Performance evaluation of HDD and SSD on 10GigE, IPoIB & RDMA-IB with Hadoop Cluster Performance Benchmarking System. In *Confluence The Next Generation Information Technology Summit (Confluence), 2014 5th International Conference-*, pages 30–35. IEEE, 2014.

[34] M. C. Schatz. CloudBurst: Highly Sensitive Read Mapping with MapReduce. *Bioinformatics*, 25(11):1363–1369, 2009.

[35] J. Shafer. I/O Virtualization Bottlenecks in Cloud Computing Today. In *Proceedings of the 2nd conference on I/O virtualization*, pages 5–5. USENIX Association, 2010.

[36] J. Sugerman, G. Venkitachalam, and B.-H. Lim. Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor. In *USENIX Annual Technical Conference, General Track*, pages 1–14, 2001.

[37] R. Takano, Y. Tanimura, A. Oota, H. Oohashi, K. Yusa, and Y. Tanaka. AIST Super Green Cloud: Lessons Learned from the Operation and the Performance Evaluation of HPC cloud. In *International Symposium on Grids and Clouds*, volume 15, 2015.

[38] M. Tatineni, J. Greenberg, R. Wagner, E. Hocks, and C. Irving. Hadoop Deployment and Performance on Gordon Data Intensive Supercomputer. In *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery*, page 45. ACM, 2013.

[39] R. Uhlig, G. Neiger, D. Rodgers, A. Santoni, F. Martins, A. Anderson, S. Bennett, A. Kagi, F. Leung, and L. Smith. Intel Virtualization Technology. *Computer*, 38(5):48–56, May 2005.

[40] M. Wasi-ur Rahman, N. S. Islam, X. Lu, J. Jose, H. Subramoni, H. Wang, and D. K. Panda. High-Performance RDMA-Based Design of Hadoop MapReduce over InfiniBand. In *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International*, pages 1908–1917. IEEE, 2013.

[41] G. Xu, F. Xu, and H. Ma. Deploying and Researching Hadoop in Virtual Machines. In *Automation and Logistics (ICAL), 2012 IEEE International Conference on*, pages 395–399. IEEE, 2012.