Performance Analysis and Evaluation of InfiniBand FDR and 40GigE RoCE on HPC and Cloud Computing Systems

Jerome Vienne, Jitong Chen, Md. Wasi-ur-Rahman, Nusrat S. Islam,
Hari Subramoni and Dhabaleswar K. (DK) Panda
Department of Computer Science and Engineering, The Ohio State University
{viennej, chenjit, rahmanmd, islamn, subramon, panda}@cse.ohio-state.edu

Abstract

Communication interfaces of high performance computing (HPC) systems and clouds have been continually evolving to meet the ever increasing communication demands being placed on them by HPC applications and cloud computing middleware (e.g., Hadoop). The PCIe interfaces can now deliver speeds up to 128 Gbps (Gen3) and high performance interconnects (10/40 GigE, InfiniBand 32 Gbps QDR, InfiniBand 54 Gbps FDR, 10/40 GigE RDMA over Converged Ethernet) are capable of delivering speeds from 10 to 54 Gbps. However, no previous study has demonstrated how much benefit an end user in the HPC / cloud computing domain can expect by utilizing newer generations of these interconnects over older ones or how one type of interconnect (such as IB) performs in comparison to another (such as RoCE).

In this paper we evaluate various high performance interconnects over the new PCIe Gen3 interface with HPC as well as cloud computing workloads. Our comprehensive analysis done at different levels, provides a global scope of the impact these modern interconnects have on the performance of HPC applications and cloud computing middleware. The results of our experiments show that the latest InfiniBand FDR interconnect gives the best performance for HPC as well as cloud computing applications.

I. Introduction

Clusters based on commodity components continue to be very popular for high-performance computing (HPC) and clouds. High performance scientific computing applications and cloud computing middleware (e.g., Hadoop) have varying computation and communication characteristics. Some applications are sensitive to latency while others are bandwidth hungry. Consequently, the communication interfaces of such systems need to be designed in a high performance and scalable manner. PCI-Express (PCIe) is an industry standard for connecting high performance interconnects with the compute nodes. High performance interconnects such as InfiniBand (IB) and 10/40 GigE have become increasingly popular for deploying modern supercomputing systems as well as clouds. Over the last decade, these communication interfaces have been continually evolving to meet the ever increasing communication demands being placed on them. During the last few years, there is an increasing focus on a new standard called RDMA over Converged Enhanced Ethernet (RoCE). The PCIe interfaces of modern platforms can now deliver speeds up to 128 Gbps (Gen3). Thus, high performance interconnects (10/40 GigE, InfiniBand 32 Gbps QDR, InfiniBand 54 Gbps FDR, and RoCE 10/40 GigE) are capable of delivering speeds from 10 to 54 Gbps [1].

However, no previous study has demonstrated how much benefit an end user in the HPC / cloud computing domain can expect by utilizing newer generations of these interconnects over older ones or how one type of interconnect (such as IB) performs in comparison to another (such as RoCE). This leads us to the following set of questions: (1) How much benefit can the user of a HPC / cloud installation hope to see by utilizing IB FDR / RoCE 40 GigE over IB QDR and RoCE 10 GigE interconnects, respectively? and (2) How does InfiniBand compare with RoCE in terms of performance?

In this paper, we address these questions with a set of well designed and comprehensive set of experiments in the HPC and cloud computing domain. We evaluate various high performance interconnects over the new PCIe Gen3 interface with HPC as well as cloud computing workloads. Our evaluations done at different levels provide a global scope of the impact these modern interconnects have on the performance of HPC applications and cloud computing middleware. Apart from the basic network level characterization of performance, we use MPI level point-to-point as well as collective benchmarks to establish a baseline for comparing the impact various interconnects can have on performance of HPC applications. We also use NAS HPC application benchmark suite [1] to study the impact

*This research is supported in part by U.S. Department of Energy grant #DE-FC02-06ER25755; National Science Foundation grants #CCF-0916302, #OCI-0926691, #CCF-0937842, #OCI-1148371 and #CCF-1213084.

[1] Since different technologies with varying encoding schemes are used in the paper, we refer to payload-level speed (not physical-level speed) demonstrated by these technologies.

978-0-7695-4831-9/12 $26.00 © 2012 IEEE
DOI 10.1109/HOTI.2012.19

IEEE computer society

modern interconnects can have on the performance of end HPC applications. We use the TestDFSIO benchmark to establish the performance of basic HDFS level read and write operations for cloud environments using Hadoop. We then move on to HBase level benchmarks to evaluate the impact of modern interconnects on the performance of typical cloud computing workloads.

Our experimental results show that the latest InfiniBand FDR interconnect gives the best performance in terms of latency and bandwidth for HPC as well as cloud computing applications. We see that IB FDR performs up to 83% better than IB QDR and 38% better than RoCE 40 GigE at the network level. We also see that RoCE 40 GigE performs 3x better than RoCE 10 GigE at the network level. For point-to-point MPI benchmarks, IB FDR performs up to 30% better than both IB QDR and RoCE 40 GigE, while 40 GigE RoCE performs up to 45% better than RoCE 10 GigE. For collectives, IB FDR performs up to 41% better than IB QDR and 10% better than RoCE 40 GigE, while RoCE 40 GigE performs up to 72% better than RoCE 10 GigE. For the NAS parallel benchmarks, IB FDR performs up to 20% better than IB QDR and up to 10% better than RoCE 40 GigE, while RoCE 40 GigE performs up to 46% better than RoCE 10 GigE. For HDFS sequential write, the throughput increases by 11% in IPoIB (FDR) over IPoIB (QDR), by 14% in IPoIB (FDR) over Sockets (40 GigE) and by 31% in Sockets (40 GigE) over Sockets (10 GigE). For a mix of HBase *Get* and *Put* operations, the throughput of IPoIB (FDR) is 9% better than IPoIB (QDR) and 25% better than Sockets (40 GigE), while Sockets (40 GigE) performs 3% better than Sockets (10 GigE).

The rest of the paper is organized as follows. In Section II, we present an overview of InfiniBand and RoCE. Section III provides background on HPC and Cloud Computing applications. In Section IV, we discuss the methodologies of our experiments. In Section V, we present the network level performances of different interconnects and Section VI illustrates the performance of point-to-point, collective and NAS parallel benchmarks. In Section VII, we present the performance of cloud computing middleware. Section VIII summarizes the overall performance of different systems. Related work is discussed in Section IX and in Section X, we present conclusions and future work.

II. Overview of InfiniBand and RoCE

We present an overview of the various network protocol stacks used in this study. As depicted in Figure 1, the network protocols can be broadly divided into two categories: Sockets based and Verbs based. Depending on the type of interconnect being used for data transfer - InfiniBand (IB) or High Speed Ethernet (HSE), each one of these can be further sub-divided into two. If the interconnect is IB, the sockets interface uses the IPoIB driver available with OFED stack [2] and the verbs interface will use the native IB verbs driver for the IB Host Channel Adapter (HCA) being used. If the interconnect is HSE, the sockets interface uses the generic Ethernet driver and the verbs interface uses the RoCE driver available with OFED stack. RoCE is a new protocol that allows to perform native IB communication seamlessly over lossless Ethernet links. RoCE packets are encapsulated into standard Ethernet frames with an IEEE assigned Ethertype, a Global Routing Header (GRH), unmodified InfiniBand transport headers and payload.

InfiniBand software stacks, such as OpenFabrics [2], also provide driver for implementing the IP layer. This makes it possible to use the InfiniBand device like any other network interface available from the system with an IP address. Although the verbs layer in InfiniBand provides OS-bypass, the IP layer does not provide so. This layer is often called “IP-over-IB” or IPoIB for short.

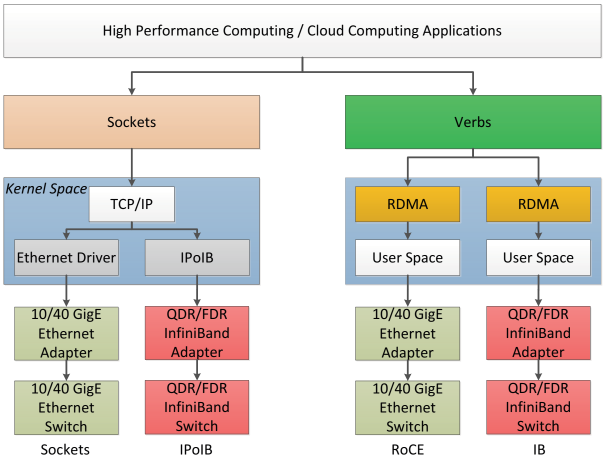

High Performance Computing / Cloud Computing Applications
Sockets
Verbs
Kernel Space
TCP/IP
Ethernet Driver
IPoIB
RDMA
RDMA
User Space
User Space
10/40 GigE Ethernet Adapter
QDR/FDR InfiniBand Adapter
10/40 GigE Ethernet Adapter
QDR/FDR InfiniBand Adapter
10/40 GigE Ethernet Switch
QDR/FDR InfiniBand Switch
10/40 GigE Ethernet Switch
QDR/FDR InfiniBand Switch
Sockets
IPoIB
RoCE
IB

Fig. 1. Overview of network protocol stacks

ConnectX-2 and ConnectX-3 HCAs from Mellanox Technologies can support all these protocols. The ConnectX-2 HCA can be configured to operate either as a 10GigE Ethernet adapter in RoCE mode or as a QDR (Quad Data Rate - 32 Gbps) IB HCA in native IB mode. The newer ConnectX-3 HCA can be configured to operate either as 40GigE Ethernet adapter in RoCE mode or as a FDR (Fourteen Data Rate - 54 Gbps) IB HCA in native IB mode.

Since our study focuses on the performance evaluation and analysis of different IB and RoCE technologies, we use a common computing platform (SandyBridge systems from Intel with support for PCI-3.0 (Gen3). The use of a common computing platform across different networks and protocols isolates the impact of processor and memory speed from the overall performance.

III. HPC and Cloud Computing Applications

A. MPI over IB and HSE

Message Passing Interface (MPI) [3] is one of the most popular programming models for writing parallel applications in cluster computing area. MPI libraries provide basic communication support for a parallel computing job. In particular, several convenient point-to-point and collective communication operations are provided. High performance MPI implementations are closely tied to the underlying network dynamics and try to leverage the best communication performance on the given interconnect. In this paper, we use MVAPICH2 1.8 [4] for our evaluations. However, our observations in this context are quite general and they can be applied to many other high performance MPI libraries as well.

B. Cloud Computing Middleware

Cloud computing economies have gained significant momentum and popularity in today's large-scale computing environments. Cloud computing middleware is mission critical at every juncture, requiring the highest performance and reliability available. Modern clouds are built on commodity components as they are cheap and easily replaceable. In today's cloud computing environments, Apache Hadoop [5] is the most popular framework for running applications on large cluster built of commodity hardware. A brief overview of some of the major components of Hadoop, used in this study, is presented below.

1) HDFS: Hadoop Distributed File System (HDFS) [6], [7] is the underlying file system for Hadoop framework. HDFS is designed for storing very large files on clusters of commodity hardware. In an HDFS cluster, there are two main types of nodes: NameNode and DataNode. The NameNode is responsible for storing and managing the metadata for the files and directories in the file system tree. The DataNodes, on the other hand, act as storage for HDFS files. In HDFS, files are usually divided into fixed-sized (64 MB) blocks and these are stored as independent units. Each block is also replicated to multiple (typically three) DataNodes in order to provide fault tolerance and availability. Most of the applications use Hadoop MapReduce to read from or write data to HDFS. MapReduce is a parallel programming model which uses map and reduce processes to store and retrieve large amount of data. When a client application writes a file to HDFS, each block is sent to a DataNode, which then replicates it to other DataNodes. On the other hand, when a client wants to read a file from HDFS, each block of the file is read from the nearest one among the DataNodes hosting the block replicas. In the existing HDFS, communications among different nodes go over Java socket [8].

2) HBase and YCSB: HBase is a java-based database that runs on top of the Hadoop framework [5]. It is used to host very large tables with many billions of entries and provides capabilities similar to Google's BigTable [9]. It is developed as part of the Apache Software Foundation's [10] Apache Hadoop project [8] and runs on top of HDFS. For performance measurement of basic HBase operations, Yahoo! Cloud Serving Benchmark (YCSB) [11] facilitates performance comparisons of different key/value-pair and cloud data serving systems. It defines a core set of benchmarks for four widely used systems: HBase, Cassandra [12], PNUTS [13] and a simple shared MySQL implementation.

IV. Methodology

The goal of our evaluation is to provide a quantitative analysis of the benefits of both increased data rates from IB QDR to IB FDR and from RoCE 10 GigE to RoCE 40 GigE. In this evaluation, we will only change the network device, leaving all other factors the same.

Our evaluation system consists of four compute nodes featuring the Intel Sandy Bridge-EP platform. Each node has dual Intel Xeon E5-2670 2.6GHz, eight-core processors with 20 MB L3 shared cache. The nodes have 32 GB of main memory. The platform is equipped with one PCIe 3.0 slot. We use RedHat Enterprise Edition 6.1 (RHEL6) with kernel 2.6.32-131.0.15.el6.x86_64 as the operating system on these nodes. Unless mentioned, all applications used in this paper are built with gcc 4.4.6. In addition, Mellanox OpenFabrics Enterprise Edition (MLNX_OFED) 1.5.3-3 [14] is used to provide the InfiniBand interface stack. Nodes are connected to a Mellanox FDR switch SX6036 for all experiments. All the provided results are corresponding to the average of 10 runs.

This study will focus on High-Performance Computing (HPC) and Hadoop Cluster which both require high-speed, high bandwidth, low latency networking infrastructures.

An initial evaluation of native InfiniBand performance on verbs level is done using OFED tools. Then for the HPC evaluation, we use Message Passing Interface (MPI) [3], which is the most prominent parallel programming model used in HPC. As we already mentioned in III-A, we use MVAPICH2 1.8 [4] for our evaluations. However, our observations in this context are quite general and they should be applicable to other high performance MPI libraries as well. The MPI evaluation is decomposed into two parts. First, we evaluate MPI primitives using OSU Micro-Benchmarks [15]. Second, we use NAS Parallel Benchmarks [1] to analyze the benefit of IB FDR and RoCE 40 GigE compared to IB QDR and RoCE 10 GigE.

In this study, we also perform different experiments to demonstrate the impact of increased data rates on various cloud computing applications and middleware using

Hadoop. Our experimental setup consists of three DataNodes, one NameNode and one client. The Hadoop cluster has four compute nodes featuring Intel Sandy Bridge-EP. Three of them are used as DataNodes and one as the NameNode. The client runs in the same node as the NameNode. The replication factor for HDFS is set to three which is the default value of replication in HDFS. For all our experiments, we use hadoop-0.20.2 and hbase-0.90.3. In the experiments with HBase, the number of region servers is set to three.

V. Network Level Performance

The results of network level bandwidth and latency benchmarks run over the various interconnects under consideration (IB FDR, IB QDR, RoCE 40 GigE and RoCE 10 GigE) are presented in this section. Figure 2 shows the performance comparison of various interconnects for network level bandwidth and latency benchmarks. We use these results to establish a baseline for the possible performance improvements that are obtained at higher levels of communication - MPI point-to-point (section VI-A), MPI collectives (section VI-B) and scientific computing applications (section VI-C). We can see that IB FDR gives the best performance in terms of latency and bandwidth of all available interconnects. Although IB QDR gives better latency compared to RoCE 40 GigE, we see that RoCE 40 GigE is able to deliver higher bandwidth than IB QDR. This is due to the difference in the network level encoding used by IB QDR and RoCE 40 GigE. While IB QDR uses a 8/10 encoding (every 10 bits sent carry 8 bits of data) RoCE 40 GigE uses a 64/66 encoding (every 66 bits sent carry 64 bits of data).

VI. MPI Level Performance

In order to investigate the performance benefits brought by IB FDR to MPI level communication, experiments are carried out on four SandyBridge nodes (16 cores/node) with MVAPICH2 1.8 and OSU benchmarks (OMB). OMB is able to evaluate the performance of both point-to-point communication and collective communication operations. We observe obvious performance improvement with IB FDR in comparison to IB QDR. Similarly, we carry out experiments on RoCE 40 GigE and RoCE 10 GigE to evaluate the performance improvement provided by RoCE 40 GigE.

A. Performance of Point-to-Point MPI Operations

Figure 3 compares the performance of various interconnects for MPI level latency and bandwidth benchmarks. We can see that IB FDR delivers the best performance in terms of latency and bandwidth of all available interconnects, which is expected since MPI library is built upon IB network level primitives. RoCE 40 GigE delivers higher bandwidth than IB QDR due to the difference in the network level encoding used by IB QDR and RoCE 40 GigE, as described in Section V. The peak bandwidth provided by IB FDR is about twice that provided by IB QDR while RoCE 40 GigE provides about a four fold improvement over RoCE 10 GigE.

B. Performance of Collective MPI Operations

From the collective communication perspective, all collectives benefit most from IB FDR. We present two collectives here due to the limit of space. As shown in Figure 4 and Figure 5, IB FDR delivers the best performance in terms of MPI_Scatter latency and MPI_Alltoallv latency among all available interconnects. RoCE 40 GigE delivers lower latency than IB QDR in the case of MPI_Scatter and MPI_Alltoallv. Change in performance for MPI_Scatter at 2KBytes suggest that the underlying algorithms can be tuned for IB (QDR) and RoCE (40 GigE and 10 GigE).

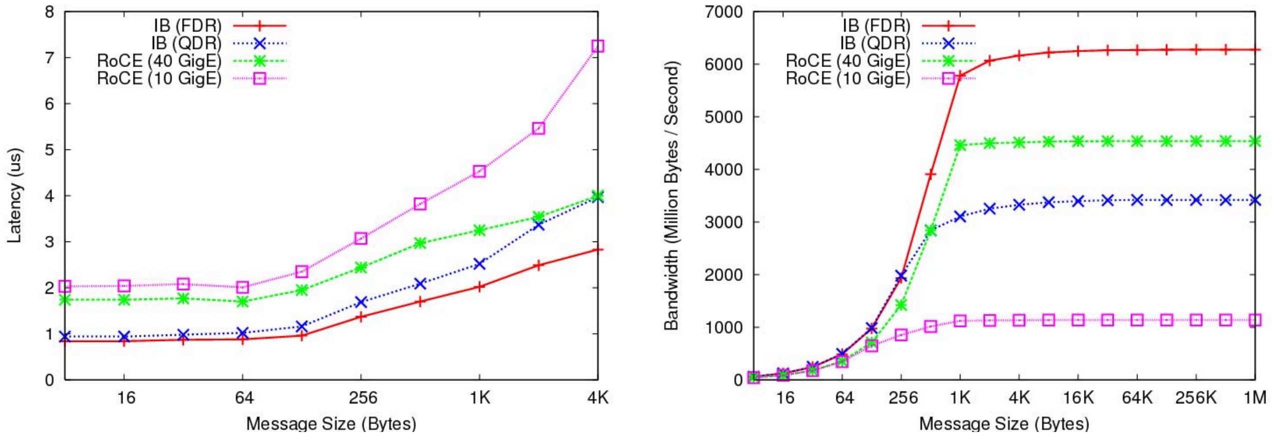
C. Performance of NAS Parallel Benchmarks

The NAS Parallel Benchmarks (NPB) suite contains a set of kernels and pseudo applications designing to mimic the computation and data movement in Computational Fluid Dynamics (CFD) applications. These benchmarks are well-known and already been widely studied in the past [16], [17].

Benchmark	**IB (QDR)**	**IB (FDR)**	**RoCE (10 GigE)**	**RoCE (40 GigE)**
FT	9.96	8.80	14.39	9.71
IS	0.80	0.64	1.32	0.71
MG	2.02	1.98	2.20	1.99
BT	24.79	24.74	26.23	24.83
LU	24.08	24.03	24.51	24.05

TABLE I. Performance (in seconds) of class C NAS benchmarks on 64 processes

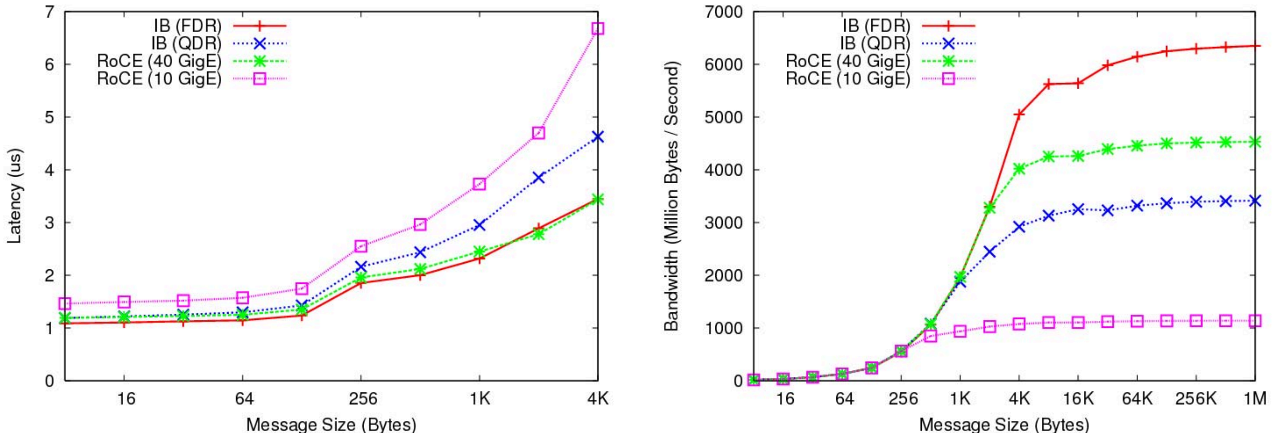
Table I shows the performance of class C NAS parallel benchmarks for 64 processes on different networks. As we can see, Integer Sort (IS) and Fast Fourier Transform (FT) are the benchmarks most impacted by the increased performance offered by IB FDR and RoCE 40 GigE. This is expected as Integer Sort (IS) and Fast Fourier Transform (FT) are known to be communication bound. The performance improvement provided by IB FDR allows IS and FT to perform 20% and 12% faster compared to IB QDR. We get 46% and 32% improvement for IS and FT, respectively, when using RoCE 40 GigE instead of RoCE 10 GigE. On



(a) Latency

(b) Bandwidth

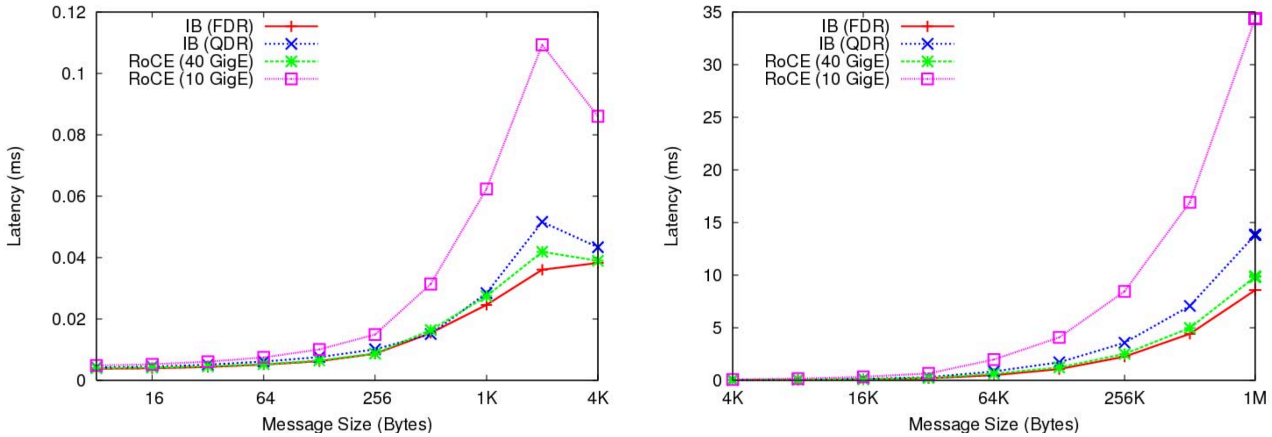
Fig. 2. Network Level Performance



(a) Latency

(b) Bandwidth

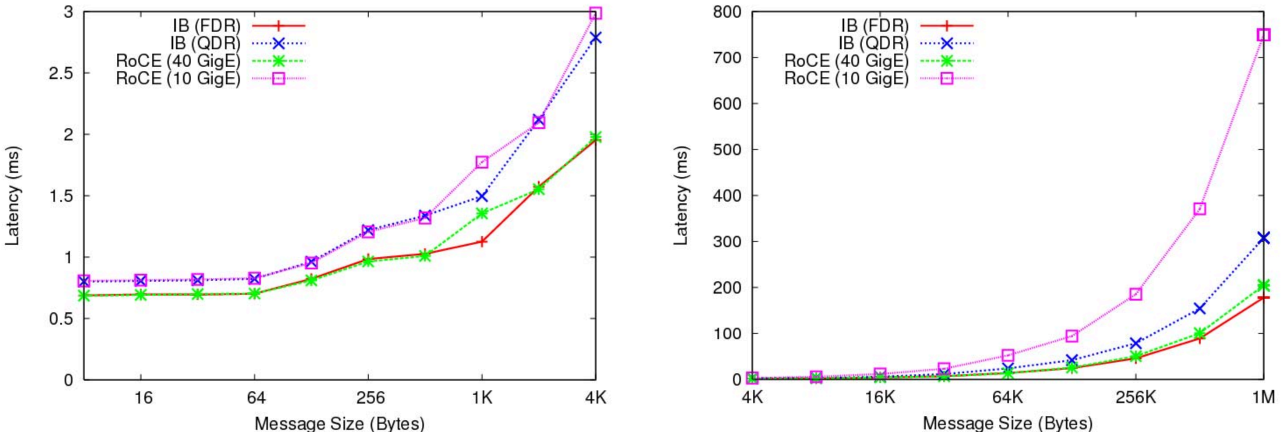
Fig. 3. Point-to-point MPI performance



(a) Small message

(b) Large message

Fig. 4. Performance of MPI_Scatter over 64 cores



(a) Small message

(b) Large message

Fig. 5. Performance of MPI_AlltoAllv over 64 cores

the other hand, Multi-Grid (MG) and Block Tri-diagonal solver (BT) are known to be computation bound and the difference between FDR and QDR performance is small. The impact of the network speed on BT and LU can only be observed on RoCE 10 GigE. The relatively low performance of this network increases the communication time leading to an augmentation of the total execution time.

VII. Impact on Cloud Computing Middlewares

In this section, we characterize the impact of various modern interconnects on the performance of Cloud Computing Middlewares.

A. Performance Evaluation of HDFS Write using TestDFSIO

TestDFSIO is a file system benchmark that measures the I/O performance of HDFS. It is implemented as Hadoop MapReduce job and supports both sequential read and write operations [18]. Since *HDFS Write* is more network sensitive compared to *HDFS Read* (occurs locally in a node [19] in most of the cases), we perform experiments of TestDFSIO sequential write in different systems like Sockets (10 GigE), Sockets (40 GigE), IPoIB (QDR) and IPoIB (FDR). In sequential write, each map task opens a file and writes specific amount of data to the file. A single reduce task aggregates the results of all the map tasks. In our experiments, we start two map tasks each writing a file to three DataNodes. We vary the file size from 1 GB to 10 GB and measure the throughput of sequential write reported by TestDFSIO.

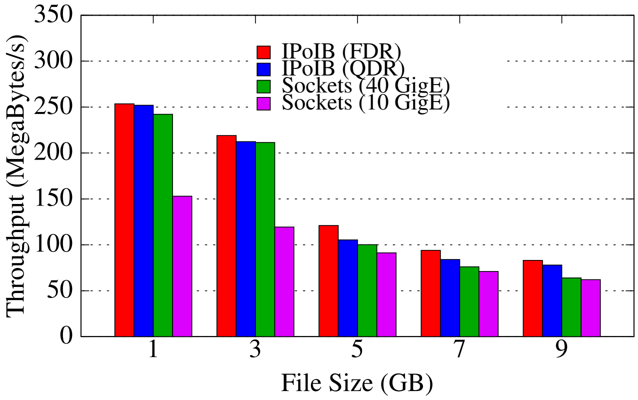


Fig. 6. Performance of *HDFS write* operation

Figure 6 shows the throughput of TestDFSIO sequential write in IPoIB (FDR), IPoIB (QDR), Sockets (40 GigE) and Sockets (10 GigE) systems. Due to the higher bandwidth of IPoIB (FDR) system, sequential write provides better throughput for all the file sizes compared to IPoIB (QDR). We obtain an average gain of 11% in IPoIB (FDR) over IPoIB (QDR). The throughput of sequential write is improved by 31% over Sockets (40 GigE) compared to Sockets (10 GigE). For IPoIB (FDR), we observe an overall benefit of 14% over Sockets (40 GigE).

B. Performance Evaluation of HBase Operations

For HBase, three different sets of operations are performed to find out the impact of IPoIB (FDR) over IPoIB (QDR) and Sockets (40 GigE) over Sockets (10 GigE). Using YCSB as our workload, we perform 100% *Get*, 100% *Put* and a 50% *Get* and *Put* Mix operations. In all these experiments, three regionservers are used. The regionservers communicate with the master (HDFS

NameNode) and the HBaseclient through the underlying interconnect. Usually regionservers are configured to reside in the same nodes as HDFS DataNodes, to improve data locality. We observe the latency and throughput for each operation. In pure HBase *Get* operation, internal MemStore of HBase can provide the desired values, so it requires less network communication. Whereas, in HBase *Put*, all the data are written to both MemStore and HDFS, hence it creates more network traffic. A balanced mix of these two operations generates network traffic for *Get* also, as for each *Put*, some old data are replaced in MemStore by the new ones each time. For these workloads, we have used 320,000 records to be inserted to and read from HBase.

In Figure 7, latency for HBase *Get* and *Put* workloads are shown. We observe 6% benefit in *Get* for Get-Put-Mix workload on IPoIB (FDR) over IPoIB (QDR). For Sockets (40 GigE) interconnect, we observe a benefit of 4% in *Get* over Sockets (10 GigE) for the same workload. For *Put* operation, the benefits are 6% and 1%, respectively.

For 100% *Put* operation, IPoIB (FDR) outperforms IPoIB (QDR) by 9%, whereas Sockets (40 GigE) is better than Sockets (10 GigE) by 1%. Overall, IPoIB (FDR) has a performance benefit of 13% over Sockets (40 GigE) in 100% *Put* operation.

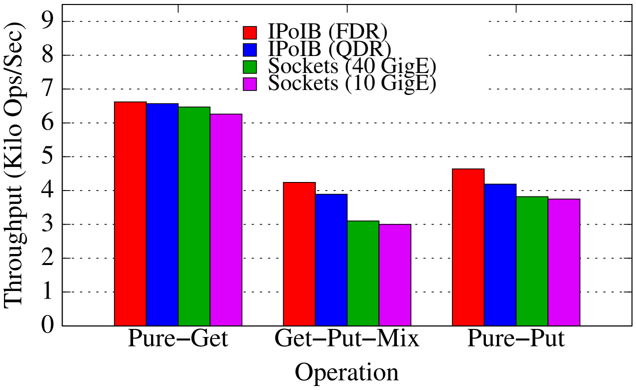


Fig. 8. HBase *Get* and *Put* throughput

Figure 8 shows the throughput for HBase *Get* and *Put* workloads. We observe a 9% benefit in throughput for HBase Get-Put-Mix workload for IPoIB (FDR) over IPoIB (QDR). For 100% *Put*, this benefit is up to 10%. For Sockets (40 GigE), we observe a benefit of up to 3% over Sockets (10 GigE) for both of these workloads. Overall, IPoIB (FDR) gets a performance benefit of 25% over Sockets (40 GigE) in throughput for these workloads.

We observe little performance benefits on IPoIB (FDR) over IPoIB (QDR) and on Sockets (40 GigE) over Sockets (10 GigE). One of the reasons behind this is simultaneous reads and writes occurring in HDFS which causes I/O bottlenecks. The I/O bottlenecks increase with increasing file size causing the throughput of TestDFSIO to drop. One anomaly we observed here is the enhanced performance of IPoIB QDR when compared with RoCE 40 GigE. We believe this could be due to extra optimizations done in the IPoIB stack by the HCA vendor. We are still in the process of investigating this trend.

VIII. Overall Performance Characterization

Figure 9 summarizes the performance improvements observed at the basic network level (section V), HPC applications (section VI) and cloud computing middleware (section VII). As we can see, a combination of the low overhead verbs interface and high throughput enables IB FDR to provide the best performance in all cases. The same factors allow RoCE 40 GigE to deliver better performance when compared to IB QDR in the network level evaluations and for HPC applications. However, IPoIB QDR is able to provide better performance than Sockets 40 GigE for cloud computing middleware. We believe this could be due to extra optimizations done in the IPoIB stack by the HCA vendor. We are still in the process of investigating this trend.

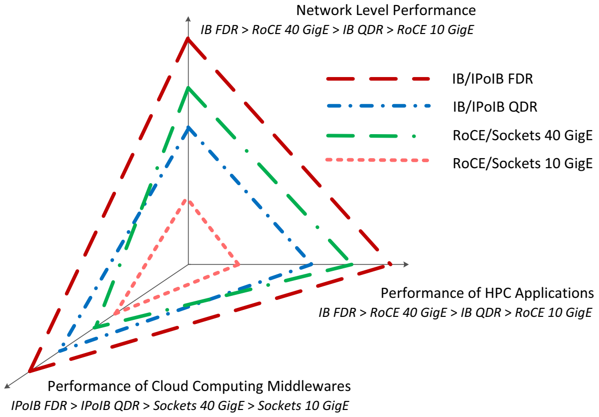
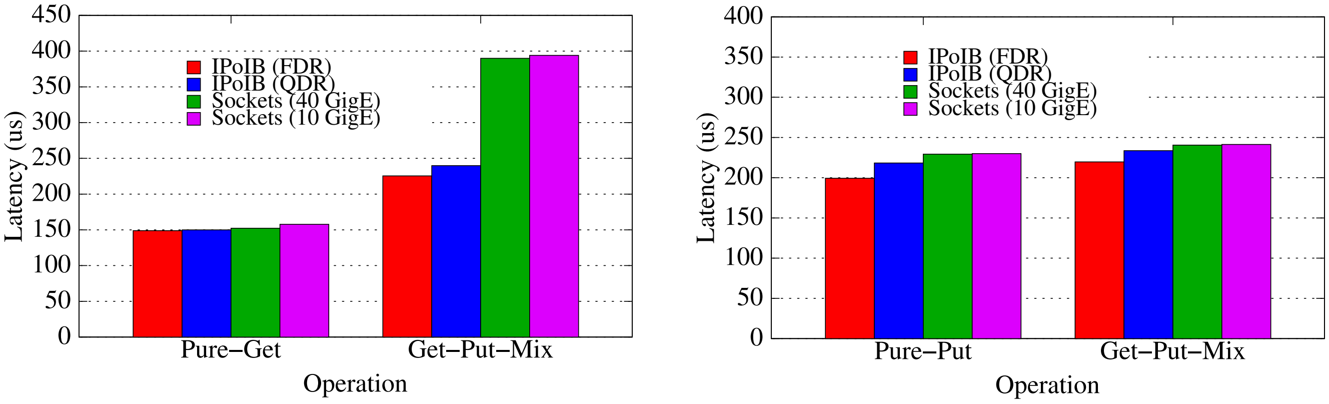


Fig. 9. Performance Characterization

IX. Related Work

Several authors have done evaluations of high performance computing systems and older generation interconnects. The authors in [20] showed the benefits InfiniBand QDR can offer over PCI-Express 2.0 gen2 interface. The authors in [21] and [22] have evaluated the InfiniPath series of InfiniBand interconnects from QLogic. In [23], the authors have demonstrated the improvement in performance that can be obtained with the then state-of-the-art Bensely platform from Intel. The authors in [18] have shown that high performance networks can improve the performance of HDFS in terms of both latency and bandwidth. The



(a) HBase *Get* latency

(b) HBase *Put* latency

Fig. 7. HBase *Get* and *Put* latency

authors in [24] have shown the impact of network on MapReduce performance. The authors in [25] have shown that high performance networks can also improve the performance of different HBase operations.

X. Conclusions and Future Work

In this paper, we have carried out a comprehensive performance evaluation of four possible modes of communication that can be performed using QDR and FDR InfiniBand hardware at network level and middleware-level (MPI and Hadoop - HDFS and HBase). Our experimental results showed that the latest InfiniBand FDR interconnect gives the best performance in terms of latency and bandwidth on HPC as well as cloud computing systems. RoCE 40 GigE delivered better performance when compared to IB QDR in network level evaluations and for HPC applications. However, IPoIB QDR provided better performance than Sockets 40 GigE for cloud computing middleware.

As part of future work, we plan to carry out similar experiments on larger-scale testbeds. We also plan to conduct a thorough performance evaluation of Hadoop components which have been recently designed over native IB. This will allow us to study the impact IB FDR and RoCE 40 GigE have on the performance of cloud computing middleware.

References

[1] D. H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, D. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber, H. D. Simon, V. Venkatakrishnan, and S. K. Weeratunga, "The NAS parallel benchmarks," vol. 5, no. 3, Fall 1991, pp. 63–73.

[2] OpenFabrics Alliance, http://www.openfabrics.org/.

[3] MPI Forum, "MPI: A Message Passing Interface," in *SC'93*, 1993.

[4] MVAPICH2: MPI over InfiniBand, 10GigE/iWARP and RoCE, http://mvapich.cse.ohio-state.edu/.

[5] Apache Hadoop, http://hadoop.apache.org/.

[6] Apache Hadoop Distributed File System, http://hadoop.apache.org/hdfs.

[7] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop Distributed File System," in *MSST'10*, 2010.

[8] The Apache Software Foundation, "The Apache Hadoop Project," http://hadoop.apache.org/.

[9] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. Gruber, "Bigtable: A Distributed Storage System for Structured Data," in *OSDI'06*, 2006.

[10] The Apache Foundation, http://www.apache.org/.

[11] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking Cloud Serving Systems with YCSB," in *SoCC 2010*, 2010.

[12] Apache Cassandra, http://cassandra.apache.org/.

[13] B. F. Cooper, R. Ramakrishnan, R. Sears, U. Srivastava, A. Silberstein, P. Bohannon, H. Jacobsen, N. Puz, D. Weaver, and R. Yerneni, "PNUTS: Yahoo!s Hosted Data Serving Platform," in *34th International Conference on Very Large Data Bases*, 2008.

[14] Mellanox OFED for Linux User Manual, http://www.mellanox.com/related-docs/prod_software/Mellanox%20OFED%20Linux%20User%20Manual%201_5_3-3_0_0.pdf.

[15] OSU Micro-benchmarks, http://mvapich.cse.ohio-state.edu/benchmarks/.

[16] F. C. Wong, R. P. Martin, R. H. Arpaci-Dusseau, and D. E. Culler, "Architectural Requirements and Scalability of the NAS Parallel Benchmarks," in *SC'99*, 1999.

[17] R. Riesen, "Communication Patterns," in *CAC'06*, 2006.

[18] S. Sur, H. Wang, J. Huang, X. Ouyang, and D. K. Panda, "Can High Performance Interconnects Benefit Hadoop Distributed File System?" in *MASVDC Workshop in Conjunction with MICRO 2010*, 2010.

[19] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," in *OSDI'04*, 2004.

[20] M. Koop, W. Huang, K. Gopalakrishnan, and D. K. Panda, "Performance Analysis and Evaluation of PCIe 2.0 and Quad-Data Rate InfiniBand," in *HOTI 16*, 2008.

[21] L. Dickman, G. Lindahl, D. Olson, J. Rubin, and J. Broughton, "PathScale InfiniPath: A First Look," in *HOTI'05*, 2005.

[22] R. Brightwell, D. Doerfler, and K. D. Underwood, "Preliminary Analysis of the InfiniPath and XD1 Network Interfaces," in *CAC'06*, 2006.

[23] M. Koop, W. Huang, A. Vishnu, and D. Panda, "Memory Scalability Evaluation of the Next-Generation Intel Bensley Platform with InfiniBand," in *HOTI 14*, 2006.

[24] Y. Wang, X. Que, W. Yu, D. Goldenberg, and D. Sehgal, "Hadoop acceleration through network levitated merge," in *SC '11*, 2011.

[25] J. Huang, X. Ouyang, J. Jose, M. W. ur Rahman, H. Wang, M. Luo, H. Subramoni, C. Murthy, and D. K. Panda, "High-Performance Design of HBase with RDMA over InfiniBand," in *IPDPS 2011*, 2011.