Performance Evaluation of InfiniBand with PCI Express

Jiuxing Liu
Server Technology Group
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
jl@us.ibm.com

Amith Mamidala, Abhinav Vishnu, and Dhabaleswar K. Panda
Department of Computer Science and Engineering
The Ohio State University
Columbus, OH 43210
{mamidala,vishnu,panda}@cse.ohio-state.edu

Abstract— **In this paper, we present an initial performance evaluation of InfiniBand HCAs from Mellanox with PCI Express interfaces. We compare the performance with HCAs using PCI-X interfaces. Our results show that InfiniBand HCAs with PCI Express can achieve significant performance benefits. Compared with HCAs using 64 bit/133 MHz PCI-X interfaces, they can achieve 20%–30% lower latency for small messages. The small message latency obtained with PCI Express is around 3.8 μs, compared with 4.8 μs with PCI-X. For large messages, HCAs with PCI Express using a single port can deliver unidirectional bandwidth up to 972 MB/s and bidirectional bandwidth up to 1932 MB/s, which are 1.24 and 2.04 times of the peak bandwidths achieved by HCAs with PCI-X, respectively. When both the ports of the HCAs are activated, PCI Express can deliver a peak aggregate bidirectional bandwidth of 2787 MB/s, which is 2.95 times of the peak bandwidth obtained using HCAs with PCI-X.**

PCI Express also improves performance at the MPI level. A latency of 4.1 μs with PCI Express is achieved for small messages. And for large messages, uni-directional bandwidth of 1497 MB/s and bi-directional bandwidth of 2721 MB/s are observed. Our evaluation also shows that PCI Express can significantly improve the performance of MPI collective communication and bandwidth-bound MPI applications.

I. Introduction

InfiniBand Architecture [1] is an industry standard which offers low latency and high bandwidth as well as many advanced features such as Remote Direct Memory Access (RDMA), atomic operations, multicast and QoS. Currently, InfiniBand products in the market can achieve a latency of several microseconds for small messages and a bandwidth of up to 700-900 MB/s. (Note that unless otherwise stated, the unit MB in this paper is an abbreviation for 10^6 bytes and GB is an abbreviation for 10^9 bytes.) As a result, it is becoming increasingly popular as a high speed interconnect technology for building high performance clusters.

PCI [2] has been the standard local I/O bus technology for the last ten years. However, currently more and more applications require lower latency and higher bandwidth than what a PCI bus can provide. As an extension to PCI, PCI-X offers higher peak performance and efficiency. However, it can still become a bottleneck for today's demanding applications, especially for those running over InfiniBand. For example, a 64 bit/133 MHz PCI-X bus can sustain around 1 GB/s aggregate bandwidth at most. However, current 4x InfiniBand HCAs have a peak bandwidth of 1 GB/s in each link direction, resulting in an aggregate bandwidth of 2 GB/s for each port. To make matters worse, some of these InfiniBand HCAs have two ports which can deliver up to 4 GB/s combined theoretical bandwidth. Even PCI-X with Double Data Rate (DDR) cannot fully take advantage of their performance potential. Another issue with PCI and PCI-X buses is that a device can share a bus with other I/O devices. Therefore, communication performance can be adversely affected by I/O operations of other devices on the same bus.

Recently, PCI Express [3] has been introduced as the next generation local I/O interconnect. Unlike PCI, PCI Express uses a serial, point-to-point interface. Compared with PCI, PCI Express can achieve lower latency by allowing I/O devices to be connected directly to the memory controller. More importantly, it can deliver scalable bandwidth by using multiple lanes in each point-to-point link. For example, an 8x PCI Express link can achieve 2 GB/s bandwidth in each direction (4 GB/s total), which matches perfectly with the requirement of current InfiniBand HCAs.

In this work, we present an initial performance evaluation of the third generation InfiniBand HCAs from Mellanox, which support PCI Express interface. We compare the performance of these HCAs with those using PCI-X interface. Our performance evaluation consists of a set of microbenchmarks at the interconnect level, including latency, bandwidth, and bi-directional bandwidth experiments. Performance results using both ports in the HCAs are also presented. In addition, we have carried out MPI level performance evaluation using both micro-benchmarks and applications.

Our performance evaluation shows that InfiniBand HCAs with PCI Express interface deliver excellent performance. Compared with HCAs using PCI-X, they can achieve 20%–30% lower latency for small messages. The smallest latency obtained is around 3.8 μs. In contrast, HCAs with PCI-X can only achieve a latency of 4.8 μs for small messages. By removing the PCI-X bottleneck, HCAs with PCI Express interface can deliver much higher bandwidth. In the bi-directional bandwidth tests, PCI Express can achieve a peak bandwidth of 1932 MB/s, which is almost twice the bandwidth delivered by PCI-X. In bandwidth tests using both ports, HCAs with PCI-X cannot achieve much performance gain due to local I/O bus being the performance bottleneck. However, PCI

Express can deliver significant performance improvements. In one bi-directional bandwidth test, PCI Express HCAs have been shown to deliver a peak aggregate bandwidth of 2787 MB/s, which is 2.95 times the bandwidth achievable using PCI-X.

At the MPI level [4] [8] [9], PCI Express also shows excellent performance. For small messages, a latency of 4.1 μs was observed. For large messages, uni-directional bandwidth of 1497 MB/s and bi-directional bandwidth of 2724 MB/s were observed. PCI Express also improves performance for MPI collective operations such as MPI_Alltoall, MPI_Bcast, and MPI_Allgather. At the application level, PCI Express HCAs have been shown to deliver significantly better performance than PCI-X HCAs for several bandwidth-bound applications in the NAS Parallel Benchmarks [5].

The remaining part of the paper is organized as follows: In Section II, we provide a brief overview of InfiniBand and PCI Express. In Section III, we describe the architectures of InfiniBand HCAs. Performance evaluations and discussions are presented in Section IV. We present related work in Section V and conclusions in Section VI.

II. Background

In this section, we provide background information for our work. First, we give a brief introduction to InfiniBand. Then, we introduce the PCI Express architecture and compare it with existing PCI buses.

A. InfiniBand

The InfiniBand Architecture (IBA) [1] defines a switched network fabric for interconnecting processing nodes and I/O nodes. It provides a communication and management infrastructure for inter-processor communication and I/O. In an InfiniBand network, processing nodes and I/O nodes are connected to the fabric by Channel Adapters (CA). The Host Channel Adapters (HCAs) are used in processing nodes.

The InfiniBand communication stack consists of different layers. The interface presented by Channel adapters to consumers belongs to the transport layer. A queue-based model is used in this interface. A Queue Pair in InfiniBand Architecture consists of two queues: a send queue and a receive queue. The send queue holds instructions to transmit data and the receive queue holds instructions that describe where received data is to be placed. Communication operations are described in Work Queue Requests (WQR), or descriptors, and submitted to the work queue. The completion of WQRs is reported through Completion Queues (CQs). Once a work queue element is finished, a completion queue entry is placed in the associated completion queue. Applications can check the completion queue to see if any work queue request has been finished. InfiniBand supports different classes of transport services. In this paper, we focus on the Reliable Connection (RC) service. InfiniBand Architecture supports both channel and memory semantics. In channel semantics, send/receive operations are used for communication. To receive a message, the programmer posts a receive descriptor which describes where the message should be put at the receiver side. At the sender side, the programmer initiates the send operation by posting a send descriptor. In memory semantics, InfiniBand supports Remote Direct Memory Access (RDMA) operations, including RDMA write and RDMA read. RDMA operations are one-sided and do not incur software overhead at the remote side. In these operations, the sender (initiator) starts RDMA by posting RDMA descriptors. At the sender side, the completion of an RDMA operation can be reported through CQs. The operation is transparent to the software layer at the receiver side. InfiniBand also supports atomic operations that can carry out certain read-modify-write operations to remote memory locations in an atomic manner.

B. PCI Express

PCI [2] has been the standard local I/O bus technology for the last ten years. It uses a parallel bus at the physical layer and a load/store based software usage model. Since its introduction, both PCI bus frequency and bus width have been increased to satisfy the ever-increasing I/O demand of applications. Later, PCI-X [2] was introduced as an extension to PCI. PCI-X is backward compatible with PCI in terms of both hardware and software interfaces. It delivers higher peak I/O performance and efficiency compared with PCI.

Recently, PCI Express [3] technology was introduced as the next generation I/O interconnect. Unlike traditional I/O buses such as PCI, PCI Express uses a high performance, point-to-point, and serial interface. Although the physical layer is different, PCI Express maintains compatibility with PCI at the software layer and no changes are necessary for current operating systems and device drivers.

In PCI and PCI-X architectures, bus frequency and width are limited due to signal skews in the underlying parallel physical interface. Further, a bus is shared among all devices connected to it. Therefore, PCI and PCI-X have limited bandwidth scalability. To achieve better scalability, PCI Express links can have multiple lanes, with each lane delivering 250 MB/s bandwidth in each direction. For example, an 8x (8 lanes in each link) PCI Express channel can achieve 2 GB/s bandwidth in each direction, resulting in an aggregate bandwidth of 4 GB/s. In comparison, a 64 bit/133 MHz PCI-X bus can only achieve around 1 GB/s bandwidth at most.

In PCI or PCI-X based systems, I/O devices are typically connected to the memory controller through an additional I/O bridge. In PCI Express based systems, I/O devices can be connected directly to the memory controller through PCI Express links. This can result in improved I/O performance. A comparison of these two approaches is shown in Figure 1.

III. Architectures of InfiniBand HCAs

In this work, we focus on performance studies of two kinds of InfiniBand HCAs from Mellanox Technologies: InfiniHost MT25208 HCAs [6] and InfiniHost MT23108 HCAs [7]. InfiniHost MT25208 HCAs are the third generation products from Mellanox which has 8x PCI Express host interfaces.

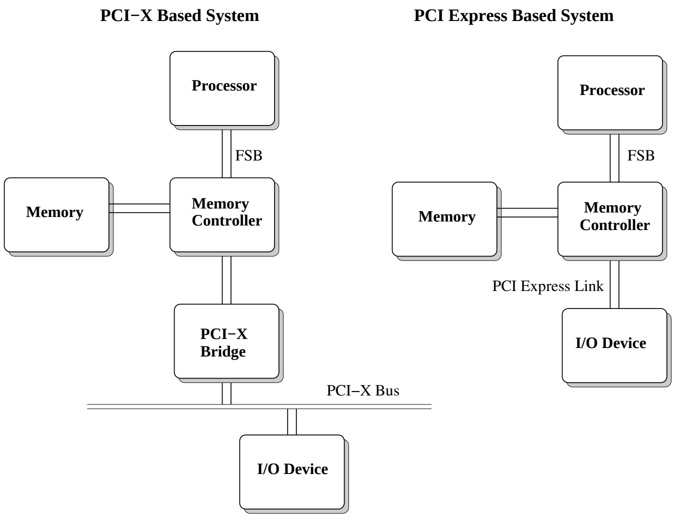

Fig. 1. Comparing PCI (PCI-X) with PCI Express

InfiniHost MT23108 cards are the second generation InfiniBand HCAs from Mellanox. They have PCI-X 64 bit/133 MHz interfaces to connect to the host. Both MT25208 and MT23108 HCAs have two physical ports. Although the major difference between MT25208 and MT23108 HCAs is the host I/O interface, MT25208 HCAs also include other enhancements such as improved internal caching and prefetching mechanisms and additional CPU offload capabilities [6]. In our experiments, the firmware in MT25208 HCAs runs in a “compatibility mode” which essentially emulates the MT23108 HCAs and the new features are not activated.

We have used VAPI as the software interface for accessing InfiniHost HCAs. This interface is provided by Mellanox and based on the InfiniBand verbs layer. It supports both send/receive operations and remote direct memory access (RDMA) operations.

IV. Performance

In this section, we present performance evaluation of Mellanox InfiniHost MT25208 PCI Express HCAs. We compare their performance with MT23108 HCAs which use PCI-X 64 bit/133 MHz interfaces. Our evaluation consists of two parts. In the first part, we show performance results at the VAPI level. In the second part, we present MPI level performance.

A. Experimental Testbeds

Our experimental testbed is a four-node InfiniBand cluster. Each node of the cluster has two 3.4 GHz Intel Xeon processors and 512 MB main memory. The nodes support both 8x PCI Express and 64 bit/133 MHz interfaces and are equipped with both MT23108 and MT25208 HCAs. An InfiniScale switch is used to connect all the nodes. The operating system used were Linux with kernel 2.4.21-15.EL.

B. VAPI Level Performance

At the VAPI level, we first show latency results of InfiniBand send/receive, RDMA write, and RDMA read operations. We also measure the cost of InfiniBand atomic operations. Then we present bandwidth numbers for tests using a single HCA port. After that, we show the results of a set of experiments that use both HCA ports. GCC 3.2 were used to compile all the test programs.

1) Latency: In this subsection, we present latency results of various InfiniBand operations such as send/receive, RDMA write, RDMA read, and atomic operations between two processes on two different nodes. Experiments for send/receive and RDMA write were carried out in a ping-pong fashion. For send/receive operations, CQ is used to check incoming messages. For RDMA write, the receiver polls on the last byte of destination memory buffer to detect the completion of RDMA communication. In the RDMA read and atomic experiments, one process acts as the initiator and the other process acts as the target. The initiator process issues RDMA read and atomic operations to buffers in the address space of the target process and uses CQ to detect completion of these operations. In all the latency experiments, the test programs consists of multiple iterations. The first 1000 iterations are used for warm-up. The average times of the following 10,000 iterations are reported.

Figure 2 compares InfiniBand send/receive latency with PCI Express and PCI-X. (Note that in the x axis of the figures, unit K is an abbreviation for 2^{10} and M is an abbreviation for 2^{20}.) We can see that PCI Express has better performance. For small messages, PCI Express achieves a latency of 4.8 μs while PCI-X achieves 6.9 μs. Figure 3 shows the results for RDMA write operations. RDMA write has better performance than send/receive operations since they incur less overhead at the receiver side. We can see that with PCI Express, we can achieve a latency of 3.8 μs. The smallest latency for PCI-X is 4.8 μs. Figure 4 shows the latency performance for RDMA read operations. With PCI Express, a latency of 9.0 μs is achieved for small messages. Latencies are around 12.4 μs with PCI-X for small messages. Figure 5 compares latency performance of InfiniBand atomic operations (Fetch-and-Add and Compare-and-Swap). The results are similar to RDMA read for small messages. Overall, we can see that HCAs using PCI Express can improve latency performance by 20%–30% for small messages.

2) Single Port Bandwidth: In this subsection we focus on bandwidth performance of InfiniBand RDMA write operations. Only one port of each HCA is used in all the tests. Results of bandwidth experiments using both ports are included in the next subsection.

A pre-defined window size *W* is used in all the bandwidth tests. In each test, the sender will issue *W* back-to-back messages to the receiver. The receiver waits for all *W* messages and then sends back a small reply message. Multiple iterations of the above procedure are carried out in the experiments. We have used a window size of 64 in our tests. The first 10 iterations of the tests are used for warm-up and the average bandwidths of the following 100 iterations are reported.

Figure 6 shows uni-directional bandwidth performance results. We can see that PCI Express HCAs perform better than

PCI-X for all messages sizes. For large messages, PCI Express delivers a bandwidth of 972 MB/s. Compared with PCI-X which has a bandwidth of 781 MB/s for large messages, PCI Express improves performance by around 24%.

Figure 7 shows the results of bi-directional bandwidth tests. HCAs with PCI-X achieve a peak aggregate bandwidth of 946 MB/s, which is only slightly higher (21%) than the uni-directional bandwidth (781 MB/s). This is mostly due to the limitation of PCI-X bus. In contract, PCI Express achieves a peak bi-directional bandwidth of 1932 MB/s, which almost doubles its uni-directional bandwidth.

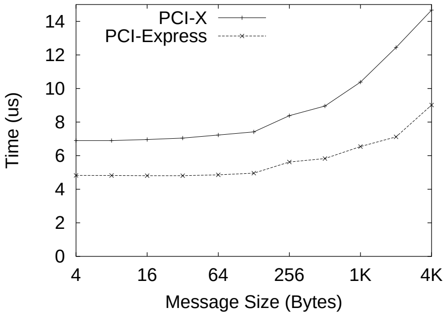

Fig. 2. Send/Receive Latency

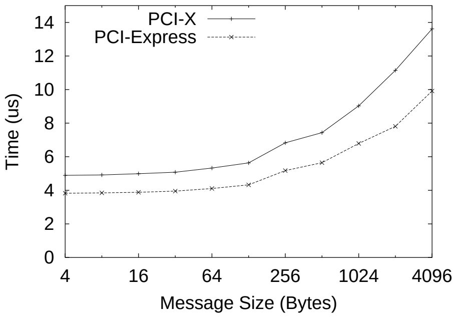

Fig. 3. RDMA Write Latency

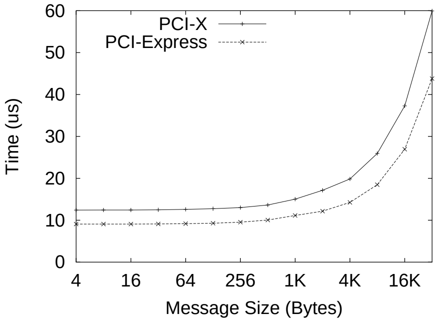

Fig. 4. RDMA Read Latency

3) Multiple Ports Bandwidth: Current Mellanox InfiniBand HCAs have two physical ports. Each port can (in theory) offer 2 GB/s bi-directional bandwidth. However, the PCI-X bus can only achieve around 1 GB/s peak bandwidth. As a result, PCI-X becomes the performance bottleneck if both ports are used. However, 8x PCI Express offers 4 GB/s theoretical bi-directional bandwidth. Therefore, both ports can be used to achieve higher performance.

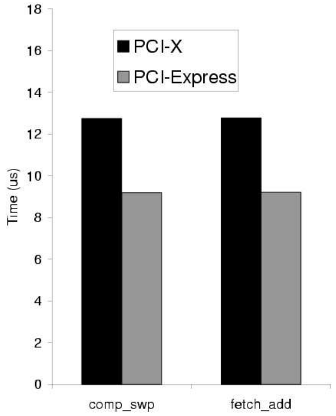

Fig. 5. Atomic Latency

We have designed a set of microbenchmarks that use both ports of the HCAs and study their benefits. We have considered two cases to take advantage of multiple HCA ports: *striping* and *binding*. In the striping mode, each message is divided into even pieces and transferred simultaneously using multiple ports. A striping threshold of 8192 bytes is used which means that messages smaller than or equal to 8192 bytes are not striped. In the binding mode, messages are never striped. However, communication channels (send channel and receive channel) of different processes in a node will use different ports of the HCA. In the striping mode, the communication is not finished until all stripes arrive at the receiver side. To notify the receiver, we send extra control messages using send/receiver operations through all the ports after sending each stripe. The receiver then polls the CQ to detect the completion of communication.

Figure 8 shows uni-directional bandwidth performance results using both HCA ports. Only striping mode is used in this test. We can see that PCI Express performs significantly better than PCI-X. HCAs with PCI Express can deliver a peak bandwidth of 1486 MB/s. With PCI-X, we can only achieve around 768 MB/s because of the PCI-X bottleneck. This number is even lower than the peak bandwidth without striping, due to the overhead of dividing and reassembling messages. For PCI Express, the bandwidth is not doubled compared to the single port case due to the HCA hardware being the performance bottleneck.

In Figure 9, we show the performance of bi-directional bandwidth tests using both ports. In the striping mode, each messages (larger than 8192 bytes) are striped and transferred using both ports. In the binding mode, the process on the first node uses port 1 to send data and uses port 2 to receive data from the process on the second node. Still we can see that PCI Express performs much better than PCI-X. We also notice that striping mode performs better than binding mode in this test for large messages. With striping, PCI Express can achieve a peak bandwidth of 2451 MB/s. The peak performance with binding is 1944 MB/s. The reason why striping performs better than binding in the bi-directional bandwidth test is that striping can utilize both ports in both directions while binding only uses one direction in each port.

In another set of tests, we have used two processes on each node with each process doing inter-node communication with another process on the other node. Both striping and binding tests have been carried out. In the binding mode, each process on the same node uses different ports for sending and receiving.

Figure 10 shows the aggregate bandwidth of two processes in the uni-directional bandwidth tests. We can see that with PCI Express, both striping and binding modes can achieve a peak bandwidth of around 1500 MB/s. The binding mode performs better than the striping mode, especially for messages smaller than 8 KB. There are two reasons for this. First, the binding mode has less overhead because it does not divide messages. Second, for small messages (less than 8 KB), both ports can be used, while the striping mode only uses one port. With PCI-X, only 776 MB/s can be achieved due to PCI-X bandwidth being the bottleneck.

In Figure 11, we show similar results for the bi-directional cases. With PCI-X, peak bandwidth is limited to around 946 MB/s. PCI Express can achieve much higher aggregate bandwidth. In the binding mode, peak bandwidth is 2745 MB/s, which is 2.9 times the bandwidth achieved by PCI-X. Due to its higher overhead, the striping mode performs a little worse than the binding mode. But it can still deliver a peak bandwidth of 2449 MB/s.

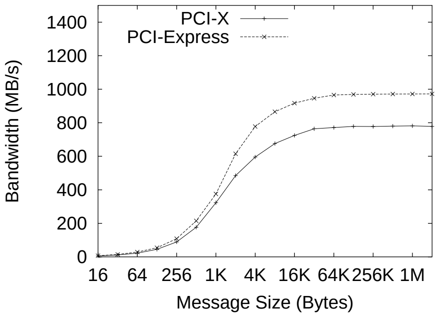

Fig. 6. Uni-Directional Bandwidth

C. MPI Level Performance

In this subsection we present MPI level results results using our enhanced MPI implementation over InfiniBand (MVA-

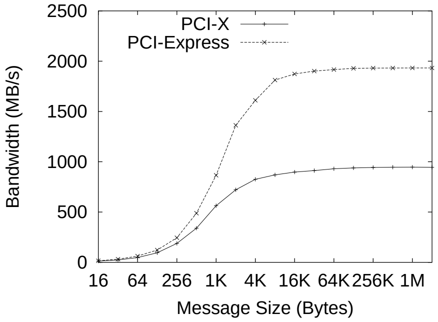

Fig. 7. Bi-Directional Bandwidth

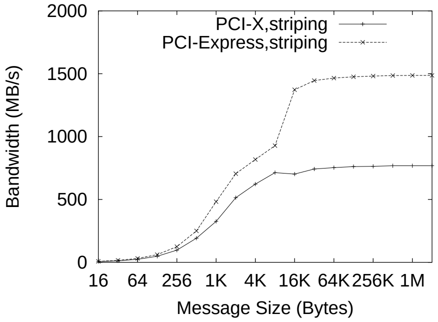

Fig. 8. Uni-Directional Bandwidth (Two Ports)

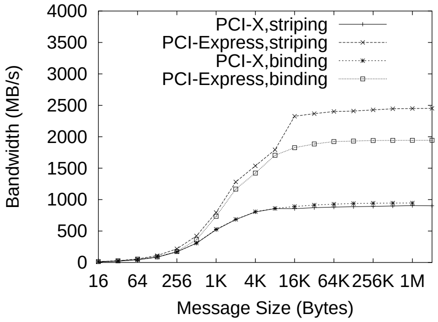

Fig. 9. Bi-Directional Bandwidth (Two Ports)

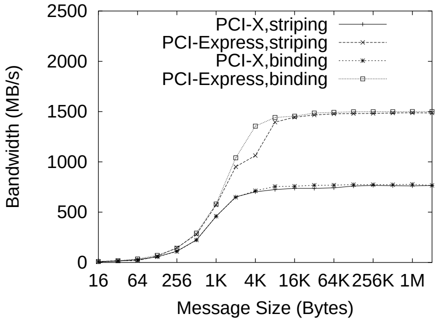

Fig. 10. Uni-Directional Bandwidth (Two Processes)

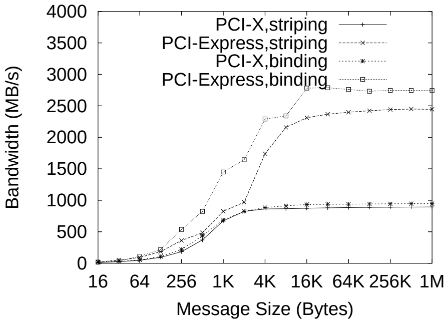

Fig. 11. Bi-Directional Bandwidth (Two Processes)

PICH) [8], [9]. Our original MVAPICH software only uses one port of each HCA. To improve its performance for PCI Express systems, we have developed an MPI implementation which can stripe large messages across both ports. In this implementation, message striping and reassembling are handled completely in the MPI layer and transparent to user applications. Details of this implementation can be found in [10]. For compiling tests, we have used GCC 3.2 compiler.

1) Latency and Bandwidth: Figure 12 shows MPI latency results for small messages. We can observe that HCAs with PCI Express can improve performance by around 20%. With PCI Express, we can achieve a latency of around 4.1 μs for small messages. In comparison, PCI-X delivers a latency of 5.1 μs for small messages. Since small messages are not striped in our new MPI implementation which uses both ports, it performs comparable to the old implementation for PCI Express.

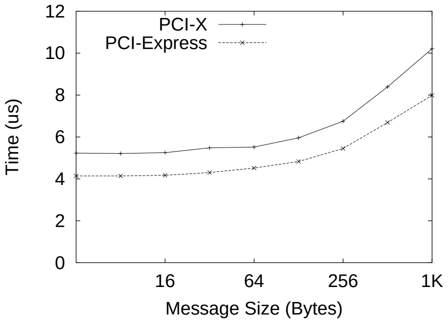

Fig. 12. MPI Latency

Figure 13 shows the performance results for the uni-directional bandwidth tests at the MPI level. Our original MPI implementation can achieve a peak bandwidth of 971 MB/s for PCI Express. It delivers around 800 MB/s peak bandwidth for PCI-X. With our new MPI implementation that stripes data across both ports, we can achieve a peak bandwidth of 1497 MB/s, which is 86% better than the one port implementation with PCI-X and 54% better than the one port implementation with PCI Express The performance drops around 8 KB in the figures is because of both protocol switch in MPI and our striping threshold.

Figure 14 shows MPI bi-directional bandwidth performance results. We can see that PCI-X can only achieve a peak bandwidth of 940 MB/s. With PCI Express, we can achieve a bandwidth of 1927 MB/s for large messages. By using both ports of PCI Express HCAs, we are able to get 2721 MB/s, which is around 2.9 times the bandwidth we can achieve with PCI-X.

We have noticed that in some cases, MPI level bandwidth is slightly higher than the VAPI level bandwidth. One reason for this is that in the VAPI tests, we have used send/receive operations to send control and synchronization messages while our optimized MPI implementation is based on RDMA operations, which have higher performance and lower overhead.

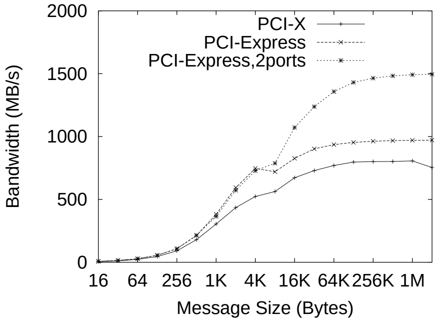

Fig. 13. MPI Bandwidth

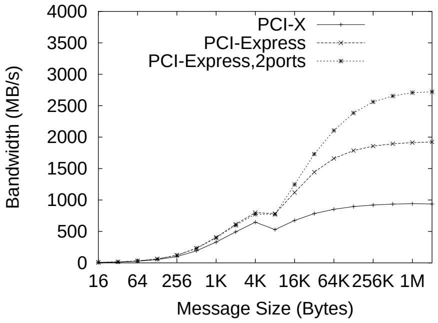

Fig. 14. MPI Bi-Directional Bandwidth

2) Collective Communication: In this subsection, we use the Pallas MPI Benchmark [11] to compare the performance of MPI collective communication for PCI Express and PCI-X. We use (4x1)(one process per node) configuration for the performance evaluation. Figures 15, 16 and 17 show the latency of three important MPI collective operations: MPI_Alltoall, MPI_Bcast and MPI_Allgather. We can see that compared with MPI running over PCI-X, MPI with PCI Express can significantly improve performance even with a single port. The improvements are up to 47%, 34%, and 48% for MPI_Alltoall, MPI_Bcast and MPI_Allgather, respectively. Further performance benefits are achieved by utilizing both ports of the HCAs. Although in the case of MPI_Alltoall this benefits are small (due to the bottleneck of HCA hardware), they are more significant for MPI_Bcast (up to 27%) and

MPI_Allgather(up to 25%).

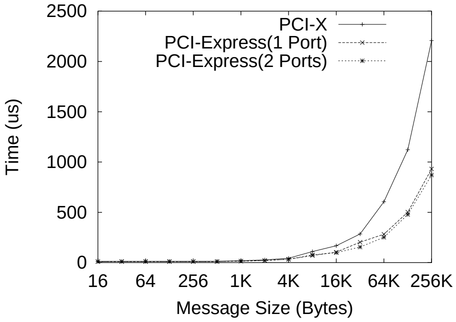

Fig. 15. MPI Alltoall Latency

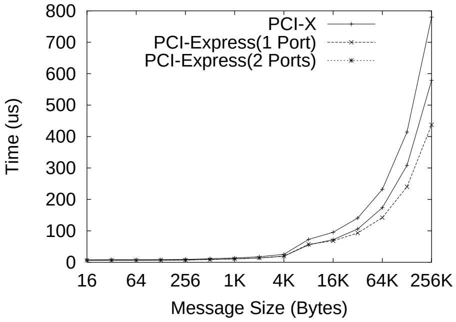

Fig. 16. MPI Broadcast Latency

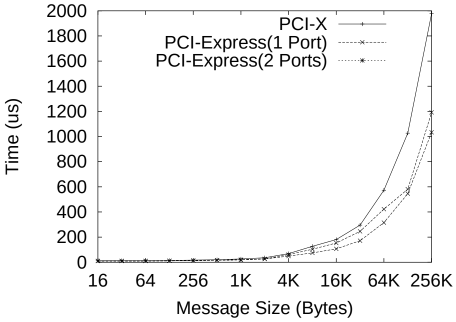

Fig. 17. MPI Allgather Latency

3) Applications: In this subsection, we show the performance of IS and FT applications in the NAS Parallel Benchmarks [5]. (We have chosen Class B for IS and Class A for FT.) Both applications are bandwidth-bound because they use large message for communication. Two configurations are used for running the tests: one process per node (4x1) and two processes per node (4x2). We show the performance using both PCI-X and PCI Express. The results are presented in Figures 18 and 19. We can see that PCI Express can reduce communication time significantly. The improvements are up to 50% for IS and up to 48% for FT. The reductions in communication time also result in improvements in application running time, which are up to 26% for IS and up to 6% for FT.

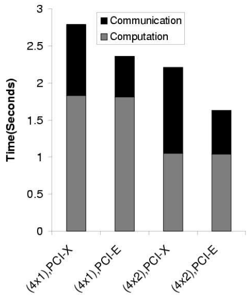

Fig. 18. IS (Class B) Results

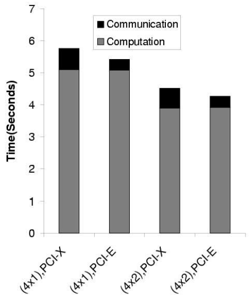

Fig. 19. FT (Class A) Results

V. Related Work

Studies on the performance of high performance interconnects including InfiniBand, Myrinet, Quadrics, and 10 gigabit ethernet have been carried out in the literature [12], [13], [14], [15]. Our previous work [16], [17] proposed test suites to compare performance of different VIA [18] and InfiniBand implementations. We have also conducted performance evaluation of different high speed interconnects at the MPI level [19]. In this paper, we focus on the interaction between InfiniBand Architecture and local I/O bus technologies. Our objective is to study how PCI Express can help us achieve better communication performance in an InfiniBand cluster.

VI. Conclusions and Future Work

In this work, we have conducted a performance study of Mellanox InfiniBand HCAs with PCI Express interfaces. We have used microbenchmarks and applications at the interconnect level and the MPI level for our performance evaluation. Our results show that PCI Express can greatly improve the communication performance of InfiniBand. Compared with HCAs with PCI-X 64 bit/133 MHz interfaces, HCAs with PCI Express can improve small message latency by 20%–30%. For large messages, HCAs with PCI Express can achieve up to 2.9 times the bandwidth compared with PCI-X.

In future, we plan to continue our evaluation of PCI Express technology by using more application level benchmarks and large scale systems. In this paper, we have shown that we can achieve much higher bandwidth at the MPI level by utilizing both ports of the HCAs. We are currently working on enhancing our MPI implementation to efficiently support different ways of transferring messages through multiple HCA ports as well as multiple HCAs for both point-to-point and collective communication.

Acknowledgments

This research is supported in part by Department of Energy's Grant #DE-FC02-01ER25506 and National Science Foundation's grants #CNS-0204429 and #CCR-0311542.

References

[1] InfiniBand Trade Association, “InfiniBand Architecture Specification, Release 1.2,” http://www.infinibandta.org, September 2004.

[2] PCI-SIG, “PCI and PCI-X,” http://www.pcisig.com.

[3] PCI-SIG, “PCI Express Architecture,” http://www.pcisig.com.

[4] W. Gropp and E. Lusk and A. Skjellum, “Using MPI: Portable Parallel Programming with the Message, 2nd edition,” MIT Press, 1999.

[5] NASA, “NAS Parallel Benchmarks”, http://www.nas.nasa.gov/Software/NPB

[6] Mellanox Technologies, “Mellanox InfiniBand InfiniHost III Ex MT25208 Adapters,” http://www.mellanox.com, February 2004.

[7] Mellanox Technologies, “Mellanox InfiniBand InfiniHost MT23108 Adapters,” http://www.mellanox.com, July 2002.

[8] Network-Based Computing Laboratory, The Ohio State University, “MVAPICH: MPI for InfiniBand on VAPI Layer,” http://nowlab.cis.ohio-state.edu/projects/mpi-iba/index.html.

[9] J. Liu, J. Wu, S. P. Kini, P. Wyckoff, and D. K. Panda, “High Performance RDMA-Based MPI Implementation over InfiniBand,” in *17th Annual ACM International Conference on Supercomputing (ICS ’03)*, June 2003. An extened version is in *International Journal of Parallel Programming (IJPP)*, vol. 32, no. 3, pp. 167–198, 2004.

[10] J. Liu, A. Vishnu, and D. K. Panda, “Building Multirail InfiniBand Clusters: MPI-Level Design and Performance Evaluation,” in *SuperComputing 2004 (SC ’04)*, November 2004.

[11] Intel Corporation, “Pallas MPI Benchmarks,” http://www.pallas.com/e/products/pmb/

[12] C. Bell, D. Bonachea, Y. Cote, J. Duell, P. Hargrove, P. Husbands, C. Iancu, M. Welcome, and K. Yelick, “An Evaluation of Current High-Performance Networks,” in *International Parallel and Distributed Processing Symposium (IPDPS’03)*, April 2003.

[13] F. Petrini, W. Feng, A. Hoisie, S. Coll, and E. Frachtenberg, “The Quadrics Network: High-Performance Clustering Technology,” *IEEE Micro*, vol. 22, no. 1, pp. 46–57, 2002.

[14] J. Liu, B. Chandrasekaran, W. Yu, J. Wu, D. Buntinas, S. P. Kinis, P. Wyckoff, and D. K. Pand, “Micro-Benchmark Level Performance Comparison of High-Speed Cluster Interconnects,” *IEEE Micro*, vol. 24, no. 1, pp. 42–51, 2004.

[15] J. Hurwitz and W. Feng, “End-to-End Performance of 10-Gigabit Ethernet on Commodity Systems,” *IEEE Micro*, vol. 24, no. 1, pp. 10–22, 2004.

[16] M. Banikazemi, J. Liu, S. Kutlug, A. Ramakrishna, P. Sadayappan, H. Shah, and D. K. Panda, “VIBe: A Micro-benchmark Suite for Evaluating Virtual Interface Architecture (VIA) Implementations,” in *Int’l Parallel and Distributed Processing Symposium (IPDPS ’01)*, April 2001.

[17] B. Chandrasekaran, P. Wyckoff, and D. K. Panda, “A Micro-benchmark Suite for Evaluating InfiniBand Architecture Implementations,” in *Performance TOOLS 2003 (part of the 2003 Illinois International Multiconference on Measurement, Modeling, and Evaluation of Computer-Communication Systems)*, September 2003.

[18] D. Dunning, G. Regnier, G. McAlpine, D. Cameron, B. Shubert, F. Berry, A. Merritt, E. Gronke, and C. Dodd, “The Virtual Interface Architecture,” *IEEE Micro*, pp. 66–76, March/April 1998.

[19] J. Liu, B. Chandrasekaran, J. Wu, W. Jiang, S. Kini, W. Yu, D. Buntinas, P. Wyckoff, and D. K. Panda, “Performance Comparison of MPI Implementations over InfiniBand, Myrinet and Quadrics,” in *SuperComputing 2003 (SC ’03)*, November 2003.