Evaluating InfiniBand Performance with PCI Express

InfiniBand host channel adapters (HCAs) with PCI Express achieve 20 to 30 percent lower latency for small messages compared with HCAs using 64-bit, 133-MHz PCI-X interfaces. PCI Express also improves performance at the MPI level, achieving a latency of 4.1 µs for small messages. It can also improve MPI collective communication and bandwidth-bound MPI application performance.

Jiuxing Liu
IBM T.J. Watson Research Center

Amith Mamidala
Abhinav Vishnu
Dhabaleswar K. Panda
The Ohio State University

The InfiniBand architecture (www.infinibandta.org) is an industry standard that offers low latency and high bandwidth as well as advanced features such as remote direct memory access (RDMA), atomic operations, multicast, and quality of service. InfiniBand products can achieve a latency of several microseconds for small messages and a bandwidth of 700 to 900 Mbytes/s. As a result, it is becoming increasingly popular as a high-speed interconnect technology for building high-performance clusters.

The Peripheral Component Interconnect (PCI, www.pcisig.com) has been the standard local-I/O-bus technology for the last 10 years. However, more applications require lower latency and higher bandwidth than what a PCI bus can provide. As an extension, 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, at most, sustain around a 1 Gbyte/s aggregate bandwidth. However, current 4× InfiniBand host channel adapters (HCAs) have a peak bandwidth of 1 Gbyte/s in each link direction, resulting in an aggregate bandwidth of 2 Gbytes/s for each port. To make matters worse, some of these InfiniBand HCAs have two ports, so can deliver a 4 Gbytes/s combined theoretical bandwidth. Thus, even running at double data rate, PCI-X cannot fully take advantage of the InfiniBand's performance potential. Another issue with PCI and PCI-X buses is that a device can share a bus with other I/O devices. Therefore, I/O operations of other devices on the same bus can adversely affect communication performance.

Recently, PCI Express has become the next-generation local I/O interconnect. Unlike PCI, PCI Express uses a serial point-to-point interface. It can achieve a lower latency than PCI by allowing I/O devices to connect 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 8× PCI Express link can achieve

Published by the IEEE Computer Society

2 Gbytes/s bandwidth in each direction (4 Gbytes/s total), which matches perfectly with the requirement of current InfiniBand HCAs.

The third-generation InfiniBand HCAs from Mellanox support the PCI Express interface. We compared the performance of these HCAs with those using a PCI-X interface and used a set of microbenchmarks at the interconnect level, including latency, bandwidth, and bidirectional bandwidth experiments. We looked at performance results using both ports in the HCAs. Also, we evaluated Message Passing Interface (MPI) performance using microbenchmarks and applications.

Our evaluation shows that InfiniBand HCAs with a PCI Express interface deliver excellent performance, achieving 20 to 30 percent lower latency for small messages than HCAs using PCI-X. The smallest latency obtained is around 3.8 µs. In contrast, HCAs with PCI-X can only achieve a 4.8-µs latency for small messages. By removing the PCI-X bottleneck, HCAs with a PCI Express interface can deliver much higher bandwidth. In bidirectional bandwidth tests, PCI Express can achieve a peak bandwidth of 1,932 Mbytes/s—almost twice the bandwidth delivered by PCI-X. In bandwidth tests using both ports, HCAs cannot increase performance while using PCI-X because the local I/O bus becomes the performance bottleneck. However, PCI Express can deliver significant performance improvements. In one bidirectional bandwidth test, PCI Express HCAs delivered a peak aggregate bandwidth of 2,787 Mbytes/s—2.9 times the bandwidth achievable using PCI-X.

At the MPI level,[1-3] PCI Express also shows excellent performance. We observed a latency of 4.1 µs for small messages. For large messages, HCAs working with PCI Express delivered unidirectional bandwidth of 1,497 Mbytes/s and bidirectional bandwidth of 2,724 Mbytes/s. 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 deliver significantly better performance than PCI-X HCAs for several bandwidth-bound applications in the Nasa Advanced Supercomputing (NAS) Parallel Benchmarks (www.nas.nasa.gov/Software/NPB).

Other literature has reported on high-performance interconnects such as InfiniBand, Myrinet, Quadrics, and 10 gigabit Ethernet.[4-7] Our previous work[8,9] proposed test suites to compare the performance of different virtual interface architectures[10] and InfiniBand implementations. We also evaluated the performance of different high-speed interconnects at the MPI level.[11] Here, we focused on the interaction between the InfiniBand architecture and local-I/O-bus technologies. We want to study how PCI Express can help achieve better communication performance in an InfiniBand cluster.

InfiniBand

The InfiniBand architecture defines a switched network fabric for interconnecting processing and I/O nodes. It provides a communication and management infrastructure for interprocessor communication and I/O. In an InfiniBand network, host channel adapters connect processing nodes to the fabric.

The InfiniBand communication stack consists of different layers. The interface that channel adapters present to users belongs to the transport layer. This interface uses a queue-based model. A queue pair in the InfiniBand architecture consists of two queues: send and receive. The send queue holds instructions to transmit data, and the receive queue holds instructions that describe where to put the received data. Work queue requests, or descriptors, describe communication operations before submission to the work queue. Completion queues report the completion of work queue requests. Once a work queue element finishes, a completion queue entry goes into the associated completion queue. Applications can check the completion queue to see if any work queue request has finished. InfiniBand supports different classes of transport services, although we focus on the reliable connection service.

The InfiniBand architecture supports both channel and memory semantics. Channel semantics uses send and receive operations for communication. To receive a message, the programmer posts a receive descriptor, which describes where the message should go at the receiver side. At the sender side, the programmer initiates the send operation by posting a send descriptor. In memory semantics, InfiniBand supports RDMA operations, including write and read. RDMA operations are one-sided and do not incur software overhead at the remote side. In these operations, the sender (initiator)

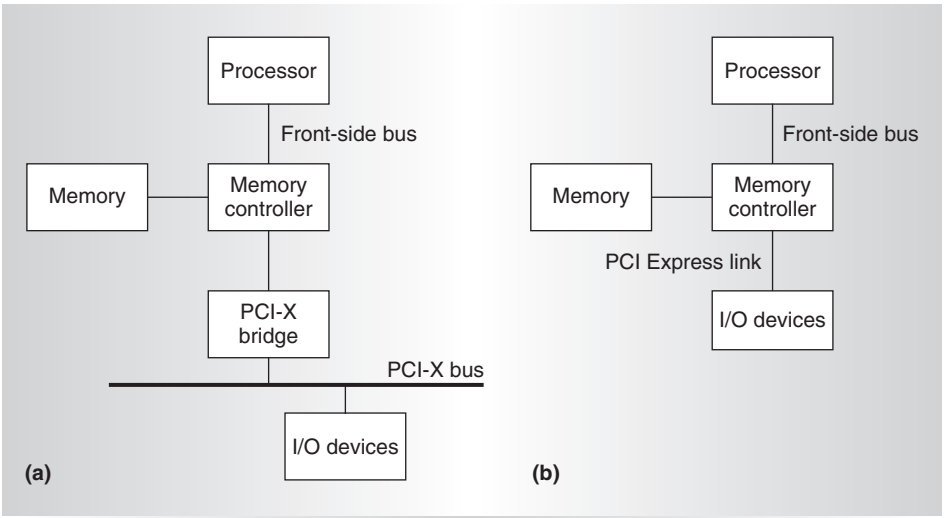

Figure 1. Comparing PCI-X (a) with PCI Express (b).

starts RDMA by posting RDMA descriptors. At the sender side, completion queues report the completion of an RDMA operation. The operation is transparent to the software layer at the receiver (target) side. InfiniBand also supports atomic operations that can carry out certain read-modify-write operations to remote memory locations in an atomic manner.

PCI Express

PCI uses a parallel bus at the physical layer and a load-store-based software usage model. Since PCI's introduction, its bus frequency and width have increased to satisfy the ever-increasing I/O demands of applications. Its extension, PCI-X, is backward compatible with PCI in terms of hardware and software interfaces. PCI-X delivers higher peak I/O performance and efficiency than PCI.

Although PCI Express' physical layer is different, it also maintains compatibility with PCI at the software layer; no changes are necessary for current operating systems and device drivers.

In PCI and PCI-X architectures, signal skews exist in the underlying, parallel, physical interface; these limit bus frequency and width. Further, all the devices connected to a bus share its bandwidth. 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 Mbytes/s of bandwidth in each direction. For example, an 8× (8 lanes in each link) PCI Express channel can achieve 2 Gbytes/s bandwidth in each direction, resulting in an aggregate bandwidth of 4 Gbytes/s.

In PCI or PCI-X based systems, I/O devices typically connect to the memory controller through an additional I/O bridge. In PCI Express-based systems, I/O devices can connect directly to the memory controller through PCI Express links, improving I/O performance. Figure 1 shows a comparison of these two approaches.

Architectures of InfiniBand HCAs

We focus on performance studies for two types of InfiniBand HCAs from Mellanox Technologies: InfiniHost MT25208 and MT23108. MT25208 HCAs are third-generation products from Mellanox; they have 8× PCI Express host interfaces. MT23108 cards are second-generation InfiniBand HCAs; 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. In our experiments, the firmware in MT25208 HCAs runs in a *compatibility mode*, which essentially emulates the MT23108 HCAs and does not activate new features.

We have used the Verbs API (VAPI) as the software interface for accessing InfiniHost

HCAs. Mellanox provides this interface, which is based on the InfiniBand verbs layer. It supports send, receive, and remote direct memory access (RDMA) operations.

Performance

Our experimental testbed is a four-node InfiniBand cluster. Each node has two 3.4-GHz Intel Xeon processors and a 512-Mbytes main memory. The nodes support both 8× PCI Express and PCI-X 64-bit, 133-MHz interfaces and have MT23108 and MT25208 HCAs. An InfiniScale switch connects all the nodes. The operating system we used was Linux with kernel 2.4.21-15.EL.

We evaluated the performance of MT25208 PCI Express HCAs, comparing their performance with MT23108 HCAs, which use PCI-X 64-bit, 133-MHz interfaces. Our evaluation consisted of two parts: performance results at the VAPI and MPI levels.

VAPI-level performance

At the VAPI level, we measured latency, and single- and mutiple-port bandwidth.

InfiniBand operation latency. Our tests measured the latency of various InfiniBand operations such as send, receive, RDMA write, RDMA read, and atomic operations between two processes on different nodes. We carried out experiments for send, receive, and RDMA write in a ping-pong fashion. For send or receive operations, the completion queue checks incoming messages. For RDMA write, the receiver polls the last byte of the destination memory buffer to detect the completion of RDMA communication. In the RDMA read and the atomic experiments, one process acts as the initiator, and the other process acts as the target. The initiator issues RDMA read and atomic operations to buffers in the target's address space and uses the completion queue to detect the completion of these operations. In all the latency experiments, the test programs consist of multiple iterations; the first 1,000 iterations are for warm-up. We reported the average times of the following 10,000 iterations.

Figure 2a compares InfiniBand send and receive latency for PCI Express and PCI-X. The figure shows that PCI Express has better performance: For small messages, it achieves a latency of 4.8 µs compared to PCI-X's 6.9 µs. Figure 2b shows similar results for RDMA write operations. RDMA write has better performance than the send and receive operations, because it incurs less overhead at the receiver side. PCI Express achieves a 3.8-µs latency. The smallest latency

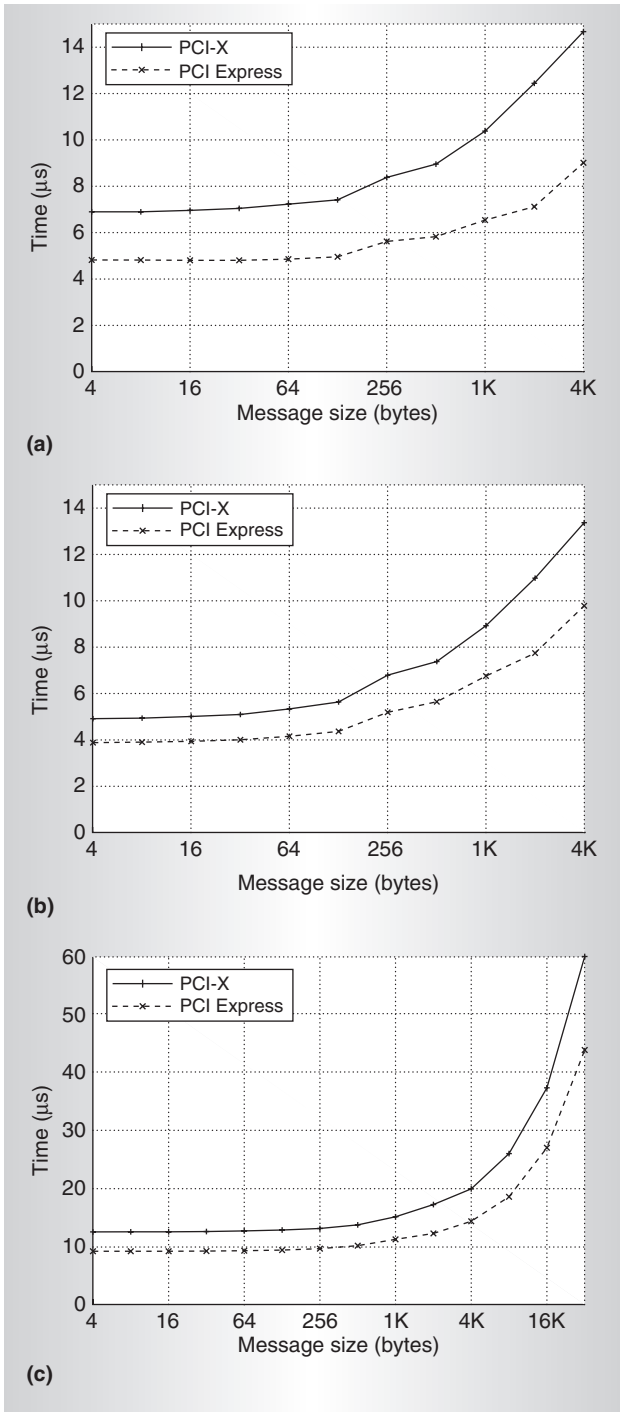

Figure 2. Latencies: send and receive (a), and RDMA write (b) and read (c).

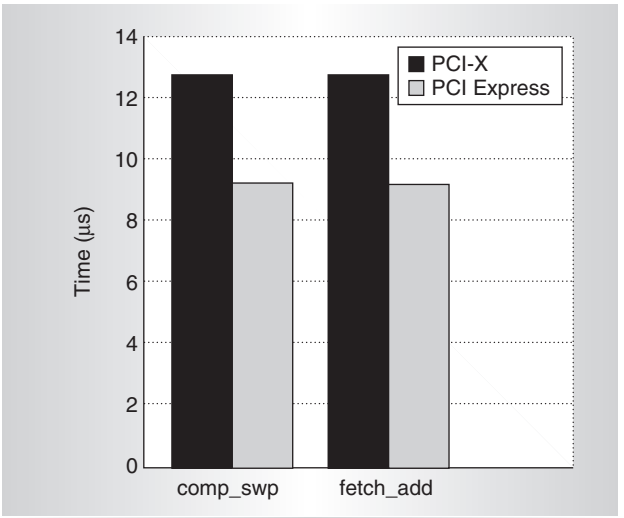

Figure 3. Atomic latency.

for PCI-X is 4.8 µs.

Figure 2c shows the latency performance for RDMA read operations, which achieve a small-message latency of 9.0 µs with PCI Express. Small-message latencies are around 12.4 µs for PCI-X. Figure 3 compares latency performance of InfiniBand atomic operations (fetch_add and comp_swp). The results are similar to those for RDMA read with small messages. Overall, HCAs using PCI Express can improve latency by 20 to 30 percent for small messages.

Single-port bandwidth. In evaluating bandwidth performance for RDMA write operations, we only used one port of each HCA in all the tests. We also used predefined window size W 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. The experiments carried out multiple iterations of that procedure. We used a window size of 64 in our tests. The test's first 10 iterations are for warm-up, and we report the average bandwidths of the following 100 iterations.

Figure 4a shows unidirectional bandwidth performance. PCI Express HCAs perform better than PCI-X for all messages sizes. For large messages, PCI Express delivers a bandwidth of 972 Mbytes/s. PCI Express improves performance by around 24 percent over PCI-X, which has a bandwidth of 781 Mbytes/s for large messages. Figure 4b shows the results of bidirectional bandwidth tests. HCAs with PCI-X achieve a peak aggregate bandwidth of 946 Mbytes/s, which is only slightly higher (21 percent) than the unidirectional bandwidth (781 Mbytes/s). This difference arises mostly from the PCI-X bus limitations. In contrast, PCI Express achieves a peak bidirectional bandwidth of 1,932 Mbytes/s, almost double its unidirectional bandwidth.

Multiple-ports bandwidth. Current Mellanox InfiniBand HCAs have two physical ports. Each port can (in theory) offer 2-Gbyte/s bidirectional bandwidth. However, the PCI-X bus can only achieve around 1 Gbyte/s peak bandwidth. So PCI-X becomes the performance bottleneck if both ports are in use. However, 8× PCI Express offers 4-Gbyte/s theoretical bidirectional bandwidth. Therefore, both ports can achieve higher performance.

We designed a set of microbenchmarks that use both ports of the HCAs and studied their benefits. We considered two cases that take advantage of multiple HCA ports: striping and binding. In striping mode, the message sender divides each message into even pieces and transfers them simultaneously using multiple ports. A *striping threshold* of 8,192 bytes means that the sender does not stripe messages smaller than or equal to 8,192 bytes. In binding mode, the sender never stripes messages. However, communication (send and receive) channels of different processes in a node will use different HCA ports. In striping mode, the communication is not finished until all the stripes arrive at the receiver. To notify the receiver, we send extra control messages via send or receiver operations; these go to all the ports after we send each stripe. The receiver then polls the completion queue to detect the communication's completion.

Figure 4c shows unidirectional bandwidth performance results using both HCA ports operating in striping mode. PCI Express performs significantly better than PCI-X. HCAs with PCI Express can deliver a peak bandwidth of 1,486 Mbytes/s. With PCI-X, we can only achieve around 768 Mbytes/s because of the PCI-X bottleneck. This number is even lower than the peak bandwidth without striping, because of the overhead to divide and reassemble messages. For PCI Express, the bandwidth is not double that of the single port because the HCA hardware is the performance bottleneck.

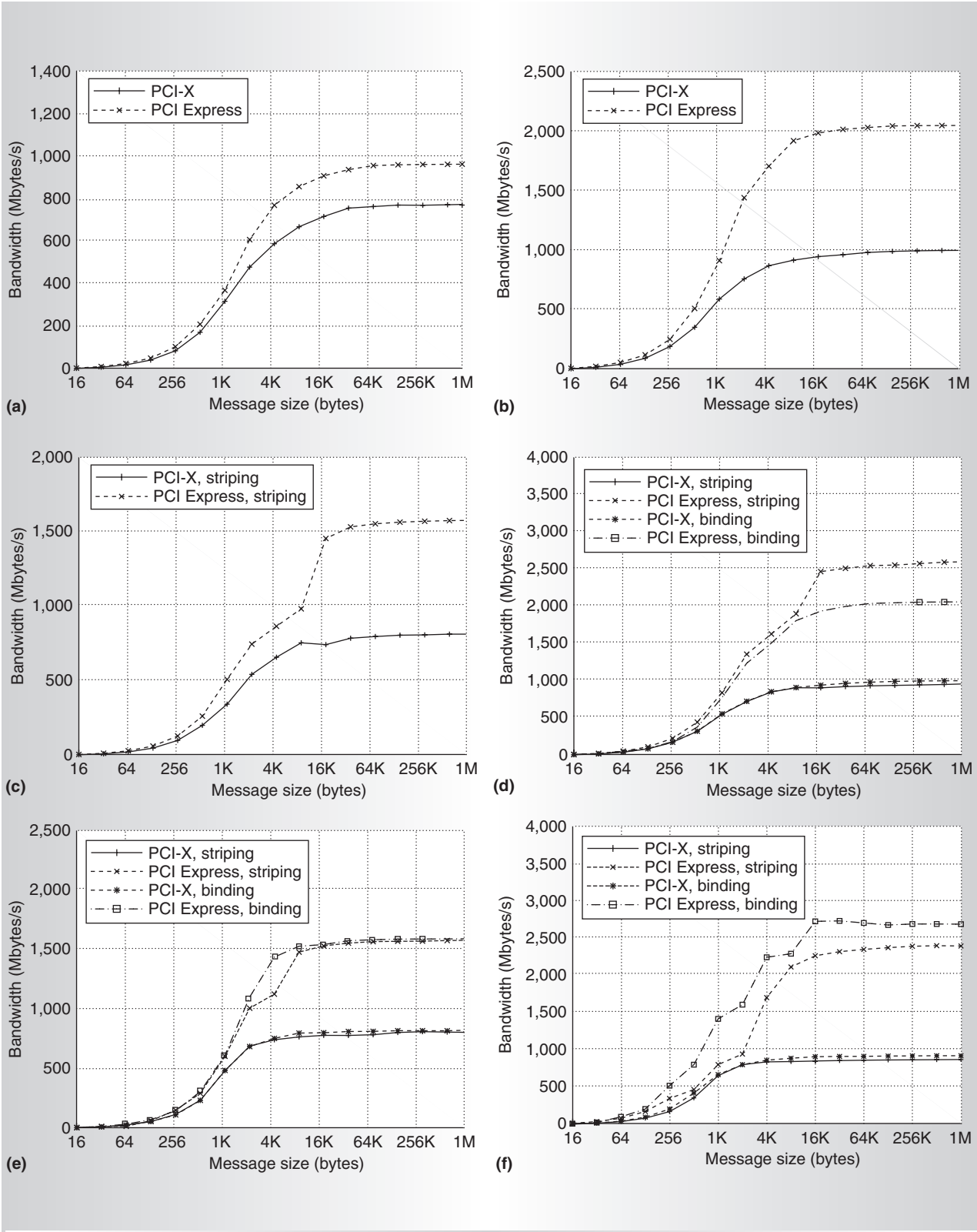

Figure 4. Bandwidth performance: single-port unidirectional (a) and bidirectional (b); dual-port unidirectional (c) and bidirectional (d); and internode communication, unidirectional (e) and bidirectional (f).

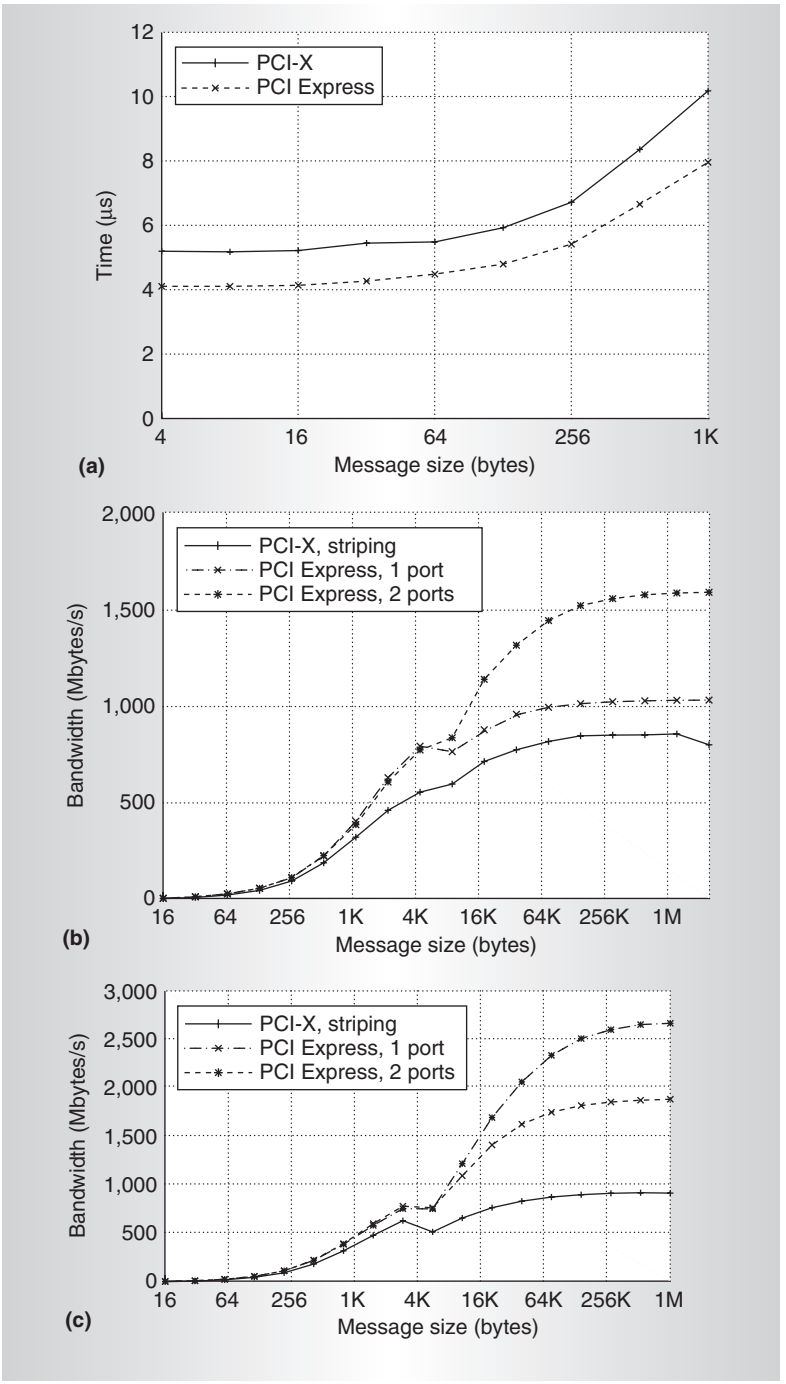

Figure 5. MPI latency (a), and unidirectional (b) and bidirectional (c) bandwidth performance for small messages.

Figure 4d shows the performance measured in bidirectional bandwidth tests using both ports. Striping mode stripes messages larger than 8,192 bytes and transfers them using both ports. In binding mode, the process on the first node uses port 1 to send data; it 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 2,451 Mbytes/s. The peak performance with binding is 1,944 Mbytes/s. Striping performs better than binding in the bidirectional bandwidth test because striping can use both ports in both directions while binding only uses one direction in each port.

In another set of tests, we used two processes, one on each node; each process communicates with the other process on the other node. We carried out both striping and binding tests. In binding mode, each process on the same node uses different ports for sending and receiving.

Figure 4e shows the aggregate bandwidth of two processes in the unidirectional bandwidth tests. With PCI Express, both striping and binding modes can achieve a peak bandwidth of around 1,500 Mbytes/s. Binding mode performs better than striping mode, especially for messages smaller than 8 Kbytes. There are two reasons for this. First, binding mode has less overhead because it does not divide messages. Second, binding mode uses both ports for small messages (less than 8 Kbytes), while striping mode only uses one port. PCI-X achieves only 776 Mbytes/s because its bandwidth is the bottleneck.

Figure 4f shows similar results for the bidirectional cases. PCI-X limits peak bandwidth to around 946 Mbytes/s. PCI Express can achieve much higher aggregate bandwidth. In binding mode, peak bandwidth is 2,745 Mbytes/s, 2.9× the bandwidth of PCI-X. Because of its higher overhead, striping mode's performance is a little worse than that of binding mode. But it can still deliver a peak bandwidth of 2,449 Mbytes/s.

MPI-level performance

We present MPI level results using our enhanced MPI implementation over InfiniBand (known as Mvapich, which stands for MPI-1 over VAPI for Infiniband).[2,3] Our original Mvapich software only uses one port of each HCA. To improve its performance for PCI Express systems, we developed an MPI

implementation that can stripe large messages across both ports. This implementation handles message striping and reassembly completely in the MPI layer; it is transparent to user applications. Our previous work provides details of this implementation.[12] To compile the tests, we used the GCC 3.2 compiler.

Latency and bandwidth. Figure 5a shows MPI latency results for small messages. HCAs with PCI Express can improve performance by around 20 percent. With PCI Express, we can achieve a latency of around 4.1 µs for small messages. PCI-X delivers a latency of 5.1 µs for small messages. Because our new MPI implementation uses both ports and does not stripe small messages, it performance is comparable to that of the old implementation for PCI Express.

Figure 5b shows the performance results for unidirectional bandwidth tests at the MPI level. Our original MPI implementation can achieve a peak bandwidth of 971 Mbytes/s for PCI Express. It delivers around 800-Mbytes/s peak bandwidth for PCI-X. Our new MPI implementation, which stripes data across both ports, can achieve a peak bandwidth of 1,497 Mbytes/s, 86 percent better than the one-port PCI-X implementation and 54 percent better than the one-port PCI Express implementation. In Figures 5a and 5b, the performance drops at an 8-Kbytes message size because of MPI protocol switching and our striping threshold.

Figure 5c shows MPI bidirectional bandwidth performance results. We can see that PCI-X can only achieve a peak bandwidth of 940 Mbytes/s. With PCI Express, we can achieve a bandwidth of 1,927 Mbytes/s for large messages. Using both ports of PCI Express, HCAs achieves 2,721 Mbytes/s, about 2.9× the bandwidth achievable with PCI-X.

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

Collective communication. We use the Pallas MPI Benchmark (www.pallas.com/e/products/pmb/) to compare the performance of MPI collective communication for PCI Express and PCI-X. We use four nodes with one process per node (a 4 × 1 configuration) for our performance evaluation. Figure 6 shows the latency

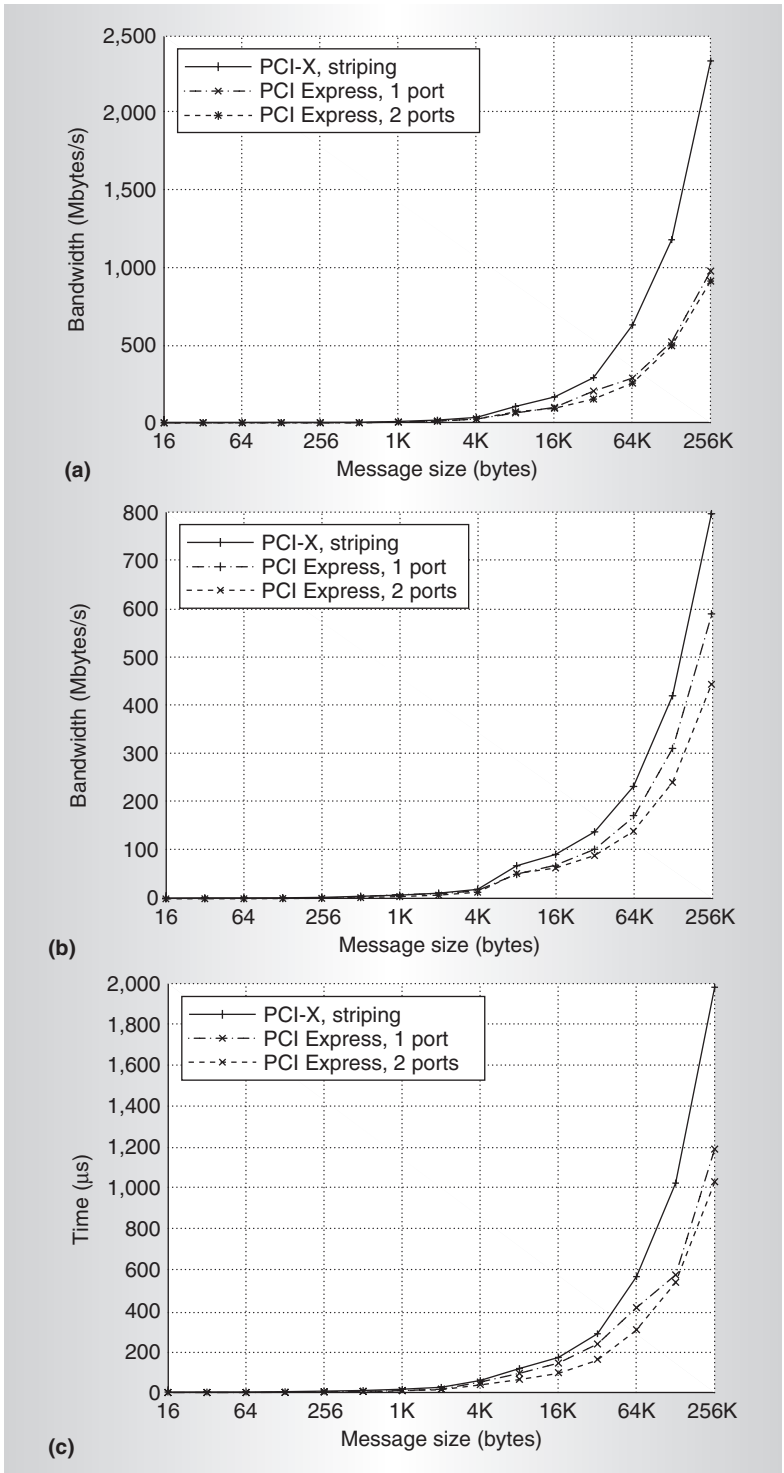

Figure 6. Latencies for MPI_Alltoall (a), MPI_Bcast (b), and MPI_Allgather (c).

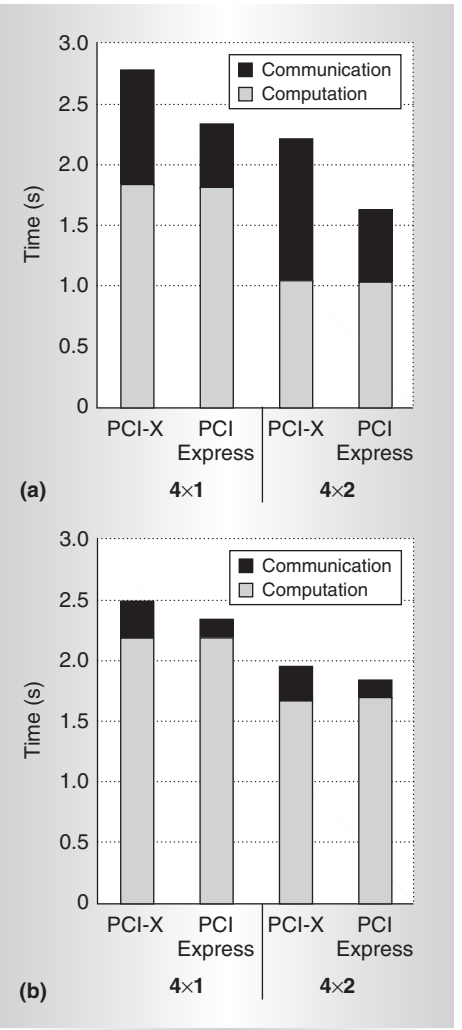

Figure 7. NAS benchmark results for IS class B (a) and FT class A (b).

of three important MPI collective operations: MPI_Alltoall, MPI_Bcast, and MPI_Allgather. MPI with PCI Express can significantly improve performance over that of MPI with PCI-X, even with a single port. The improvements are 47, 34, and 48 percent for MPI_Alltoall, MPI_Bcast, and MPI_Allgather.

Using both ports of the HCAs achieves even higher performance. For MPI_Alltoall, the benefits are small (because of the HCA hardware bottleneck). Performance improvements are more significant for MPI_Bcast (a 27 percent improvement) and MPI_Allgather (25 percent).

Applications. We show the performance of integer sort (IS) and 3D fast-Fourier-Transfer partial-differential equation (FT) applications in the NAS Parallel benchmarks. (We chose Class B for IS and Class A for FT.) Both applications are bandwidth-bound because they use large messages for communication. We use two configurations for running the tests: four nodes, with each node running one (4×1) or two (4×2) processes. Figure 7 shows the performance using both PCI-X and PCI Express. PCI Express can reduce communication time significantly; the improvements are 50 percent for IS and 48 percent for FT. The reductions in communication time also reduce application runtime by 26 percent for IS and 6 percent for FT.

In our performance study of Mellanox InfiniBand HCAs with PCI Express interfaces, we used microbenchmarks and applications at the VAPI and the MPI levels for performance evaluation. This study showed that PCI Express can greatly improve InfiniBand's communication performance.

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

Acknowledgments

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

References

1. W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message, 2nd ed.*, MIT Press, 1999.
2. "MVAPICH: MPI for InfiniBand on VAPI Layer," Network-Based Computing Lab., The Ohio State Univ., http://nowlab.cis.ohio-state.edu/projects/mpi-iba/index.html.
3. J. Liu et al., "High Performance RDMA-Based MPI Implementation over Infini-

Band," *Int'l J. Parallel Programming*, vol. 32, no. 3, June 2004, pp. 167-198.
4. C. Bell et al., "An Evaluation of Current High-Performance Networks," *Proc. Int'l Parallel and Distributed Processing Symp.* (IPDPS 03), IEEE CS Press, 2003, p. 28a.
5. F. Petrini et al., "The Quadrics Network: High-Performance Clustering Technology," *IEEE Micro*, vol. 22, no. 1, Jan.-Feb. 2002, pp. 46-57.
6. J. Liu et al., "Micro-Benchmark Level Performance Comparison of High-Speed Cluster Interconnects," *IEEE Micro*, vol. 24, no. 1, Jan.-Feb. 2004, pp. 42-51.
7. J. Hurwitz and W. Feng, "End-to-End Performance of 10-Gigabit Ethernet on Commodity Systems," *IEEE Micro*, vol. 24, no. 1, Jan.-Feb. 2004, pp. 10-22.
8. M. Banikazemi et al., "VIBe: A Micro-benchmark Suite for Evaluating Virtual Interface Architecture (VIA) Implementations," *Proc. Int'l Parallel and Distributed Processing Symp.* (IPDPS 01), IEEE CS Press, 2001, p. 10024b.
9. B. Chandrasekaran, P. Wyckoff, and D.K. Panda, "A Micro-Benchmark Suite for Evaluating InfiniBand Architecture Implementations," *Proc. Computer Performance Evaluations, Modelling Techniques and Tools, 13th Int'l Conf.* (TOOLS 2003), Lecture Notes in Computer Science 2794, Springer-Verlag, 2003, pp. 29-46.
10. D. Dunning et al., "The Virtual Interface Architecture," *IEEE Micro*, vol. 18, no. 2, Mar.-Apr. 1998, pp. 66-76.
11. J. Liu et al., "Performance Comparison of MPI Implementations over InfiniBand, Myrinet and Quadrics," *Proc. Supercomputing 2003* (SC 03), IEEE CS Press, 2003, p. 58.
12. J. Liu, A. Vishnu, and D.K. Panda, "Building Multirail InfiniBand Clusters: MPI-Level Design and Performance Evaluation," *Proc. Supercomputing 2004* (SC 04), IEEE CS Press, 2004, pp. 33-45.

Jiuxing Liu is a research staff member at the IBM T.J. Watson Research Center. His research interests include high-speed interconnects, cluster computing, server I/O architecture, and storage systems. Liu has a PhD from The Ohio State University. He is a member of the IEEE and the ACM.

Amith Mamidala is a PhD student in Department of Computer Science and Engineering at The Ohio State University. His research interests include collective communication in MPI over InfiniBand. Mamidala has an MS in computer science and engineering from The Ohio State University and a bachelor's degree in computer science and engineering from the Indian Institute of Technology Madras, India. He is a member of the IEEE.

Abhinav Vishnu is a PhD student in Department of Computer Science and Engineering at The Ohio State University. His research interests include message-passing algorithms for high-speed interconnects, cluster computing, and subnet management in InfiniBand. Vishnu has a BS in computer science and engineering from the Institute of Technology, Banaras Hindu University, India. He is a member of the IEEE.

Dhabaleswar K. Panda is a professor of computer science at The Ohio State University. His research interests include parallel computer architecture, high-performance computing, user-level communication protocols, interprocessor communication and synchronization, network-based computing, and quality of service. Panda has a PhD in computer engineering from the University of Southern California. He is a senior member of the IEEE and a member of the ACM.

Direct questions and comments about this article to Jiuxing Liu, IBM T.J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532; jl@us.ibm.com.