SANDIA REPORT

SAND2005-5199
Unlimited Release
Printed August 2005

An Analysis of the Pathscale Inc. InfiniBand Host Channel Adapter, InfiniPath

Douglas W. Doerfler

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2005-5199
Unlimited Release
Printed August 2005

An Analysis of the Pathscale Inc. InfiniBand Host Channel Adapter, InfiniPath

Douglas W. Doerfler
Scalable Systems Integration
Sandia National Laboratories
P.O. Box 5800, MS-0817
Albuquerque, New Mexico 87185

Abstract

The use of the InfiniBand Network Architecture (IB) in high performance computing (HPC) is a growing market. Several HPC vendors are offering IB as a high-speed message passing interconnect option in their product line today and it is anticipated that the number of installations will continue to grow over the next decade. At this time, it's predominately being used in capacity type systems. Its use in higher end capability systems is limited at this time, primarily due to the immaturity of the system software and inherent scalability limitations. Pathscale Inc. is addressing some of the key scalability issues, low-latency, message throughput and connectionless based protocols in its InfiniPath product. This report evaluates the performance of the InfiniPath product by analyzing message passing microbenchmarks, Sandia Applications, and comparison to other systems using alternative high performance messaging interconnects.

Acknowledgments

I would like to thank the staff of Pathscale for the use of their CBC Cluster. My requests for access were met in a timely fashion and the support was professional. Without their support this report would not have been possible. I would also like to thank the Pathscale team for reviewing early drafts of the report and provided technical guidance, for example their suggestions that led to the effective latency analysis in Section 5.

Contents

Figures

Tables

1. Introduction

The InfiniBand Network Architecture (IB) is being deployed in a variety of high performance computing (HPC) platforms as the high-speed messaging network for Message Passing Interface (MPI) based applications. Several vendors have committed to providing IB hardware and software for the HPC market. Mellanox is the primary provider of IB silicon, and they also provide host channel adapters and switches. Vendors using Mellanox silicon and/or hardware include Voltaire, Cisco Systems (formerly Topspin Communications) and SilverStorm Technologies (formerly InfiniCon Systems). These vendors also provide their own value added intellectual property by engineering high-port count switches and high performance IB software stacks.

Pathscale has recently entered the InfiniBand market space by introducing an IB host channel adapter (HCA), which they call InfiniPath. The performance of the InfiniPath interconnect is evaluated and compared with the performance of other high-performance interconnects used in HPC. The Platforms tested include Pathscale's CBC cluster, Sandia National Laboratories (SNL) Red Squall cluster using Quadrics' Elan4 interconnect technology, and SNL's Escher cluster utilizing Voltaire's 4X InfiniBand product. The performance is evaluated using MPI microbenchmarks and SNL's LAMMPS application. Other SNL codes were not tested due to export control restrictions that do not allow distribution and installation on the open network CBC cluster. Due to the relatively small size of the CBC and Escher clusters, scaling results are limited to a size of 16-nodes and 8-nodes respectively. Although this scale limits the ability to study large scale issues, trends can be identified which may affect larger scale performance. In addition to single processor per node performance, dual processor per node performance was also investigated.

2. Pathscale's InfiniPath

The InfiniPath HBA uses a HyperTransport interface to the host processor. Most cluster network interface controllers (NICs) use the PCI-X or the PCI-Express (PCI-XorE) interface and communicate with the host processor using a host processor bus to PCI-XorE bridge chip. With the InfiniPath HBA tied directly to the host processor, a lower latency transaction is possible than can be achieved via a PCI-XorE bridge chip. The InfiniPath HBAs network fabric connection is a 4X IB link and is compatible with standard IB switches at the physical and link layers. It also uses existing IB fabric management solutions. However, since the protocol stack used by InfiniPath is optimized for MPI it is not plug-and-play compatible with other vendors IB HBAs and software stacks. However, Pathscale has announced their intent to support the OpenIB software stack.

The InfiniPath HBA does not contain an embedded processor and all control and protocol stack operations are performed on the host processor. The advantage of this architecture is the host processor is much more powerful computationally and hence makes it possible to handle protocol computations much faster than an embedded processor. The disadvantage is that while the host is processing communication protocols it is not available to perform application processing.

3. Test Platforms

In order to understand the performance of the InfiniPath solution, it is necessary to compare and contrast it to other platforms using alternative interconnects. Microbenchmark and application performance is dependent on many system level factors. The interconnect is one of the key factors, but several other factors such as main memory, bridge chip, and host processor performance also play a role. Hence, it is not possible to directly compare interconnects contained in dissimilar platforms. However, it is possible to draw conclusions on the performance by analyzing trends and scaling information. For this study, the test platforms are listed in Table 1.

Table 1: Test Platform Summary

	CBC Cluster	Red Squall Cluster	Escher Cluster
Interconnect	4x InfiniBand	Elan4 (QsNetII)	4x InfiniBand
Host Processor Interface	HyperTransport	133MHz PCI-X	x8 PCI-Express
Link Peak BW (Aggregate)	2 GB/sec	2.133 GB/sec	2 GB/sec
Host Interface Peak BW	3.2 GB/sec	1.064 GB/sec	4 GB/sec
Host Processor	dual 2.6 GHz Opteron	dual 2.2 GHz Opteron	dual 3.4 GHz Xeon EM64T
Memory Subsystem	dual DDR-400 (2 x 3.2 GB/sec)	dual DDR-333 (2 x 2.67 GB/sec)	dual DDR2-400 (2 x 3.2 GB/sec)
Mother Board	Microway HTX Series	HP DL145	SuperMicro X6DAE-G2
Operating System	Red Hat Fudora Core 3	SuSE Linux Professional 9.1	SuSE Linux Professional 9.1
Compiler Suite (C, C++ and Fortran)	Pathscale 2.1	Pathscale 2.1	Pathscale 2.1
MPI Release		MPICH QSNET 1.24-43	MVAPICH 0.9.2

The CBC Cluster is a test and evaluation cluster administered by Pathscale for customer evaluations. It is a 16-node cluster. The Red Squall cluster is a research and development platform at SNL. It is a 256-node cluster, although scaling results are limited to 16-nodes for this study since that is the size of the CBC system. The Escher cluster is also a research and development platform at SNL. It is an 8-node cluster. Larger IB clusters exist at SNL, but they were not available at the time of this study.

4. MPI Microbenchmark Results

4.1. Point-to-Point

4.1.1. Ping-Pong & Streaming

Traditionally, in performance comparisons ping-pong latency and bandwidth tests are performed. Sometimes the bandwidth test is performed using the streaming method (multiple outstanding sends/receives issued using non-blocking MPI calls) because it produces better results. The two types of tests should be performed at all times as they show two different characteristics of how a network interface performs. Both tests are valid and both types of communication are performed in real applications. The mpi_bw benchmark (developed at SNL by the author) was used to gather the data for this section.

Figure 1 plots the latency for each platform using both ping-pong and streaming tests. For all platforms, the effective latency of the streaming test is reduced as compared to the ping-pong test. Table 2 shows the latency for a zero data byte message and includes the percentage reduction in time for the latency of the streaming test as compared to the ping-pong test. Note that the InfiniPath cluster is significantly more efficient in the streaming test, where messages are queued up for transmission using non-blocking calls.

Another way to interpret streaming latency is as the number of messages per second for a given message size. Table 3 shows the messaging rate for an eight byte message.

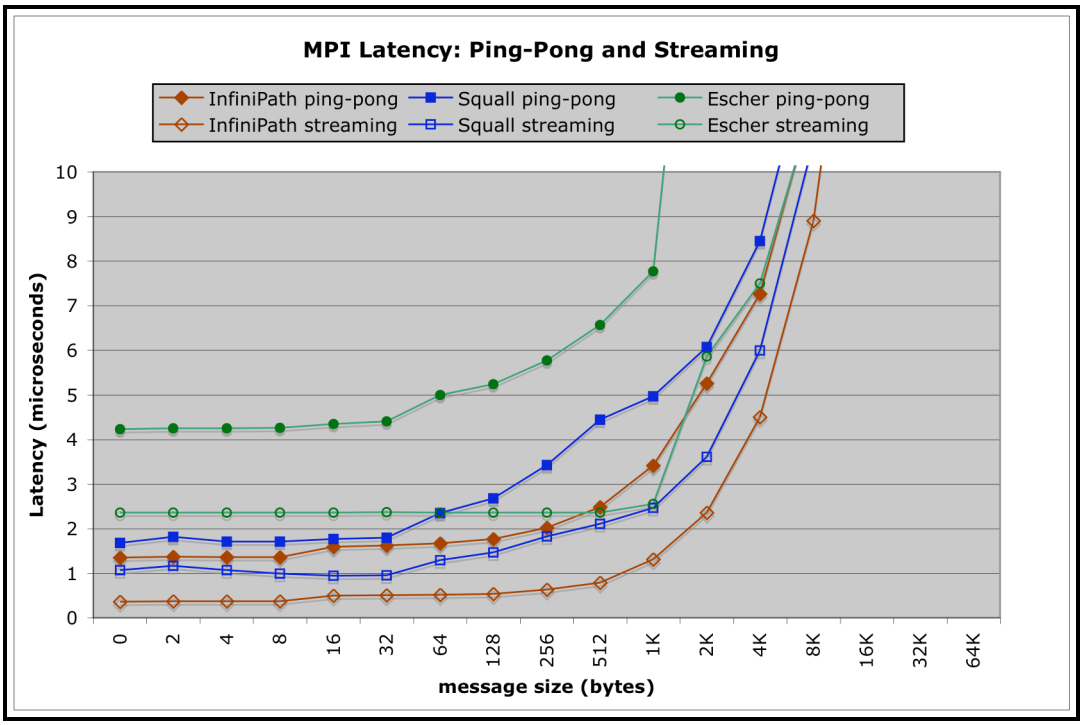

Figure 1: Ping-Pong and Streaming Latency

Table 2: Zero Data Byte Latency

Platform	Ping-Pong	Streaming	% reduction
InfiniPath	1.35	0.36	73.3%
Squall	1.68	1.07	36.3%
Escher	4.23	2.23	47.3%

Table 3: Eight Data Byte Messaging Rate (10^6 messages per sec)

Platform	Streaming
InfiniPath	2.70
Squall	1.01
Escher	0.45

Figure 2[1] plots the bandwidth for each platform. The IB based clusters are able to achieve a higher bandwidth than the Elan4 based Squall cluster. Quadrics theoretical peak bandwidth is lower than that of the IB clusters, but it also uses the less efficient PCI-X bus as opposed to the PCI-Express bus used by the Escher cluster and the HyperTransport bus of the InfiniPath cluster. However, the interesting observation is the difference in the rate at which the bandwidth increases as a function of message size. Table 4 shows the peak bandwidth. Table 5 shows the message size at which one-half of peak bandwidth is achieved.[2] The InfiniPath interconnect performs significantly better with a streaming test as opposed to a ping-pong test.

[1] In this document, for bandwidth measurements 1 MiB = 10^6 bytes. However, for message sizes; 1 KB = 1024 bytes and 1 MB = 1024^2 bytes. The PMB benchmarks report bandwidth in units of 1 MB/s = 1024^2 bytes per second. For this study, all PMB bandwidth results are translated to 1 MiB/s.

[2] Half-bandwidth was determined using a simple linear interpolation of four data points (two on each side) of the half-bandwidth point on the curve.

The Escher cluster also shows a significant improvement when requests are queued. The Squall cluster shows improvement, but it is not as dramatic as the other two clusters.

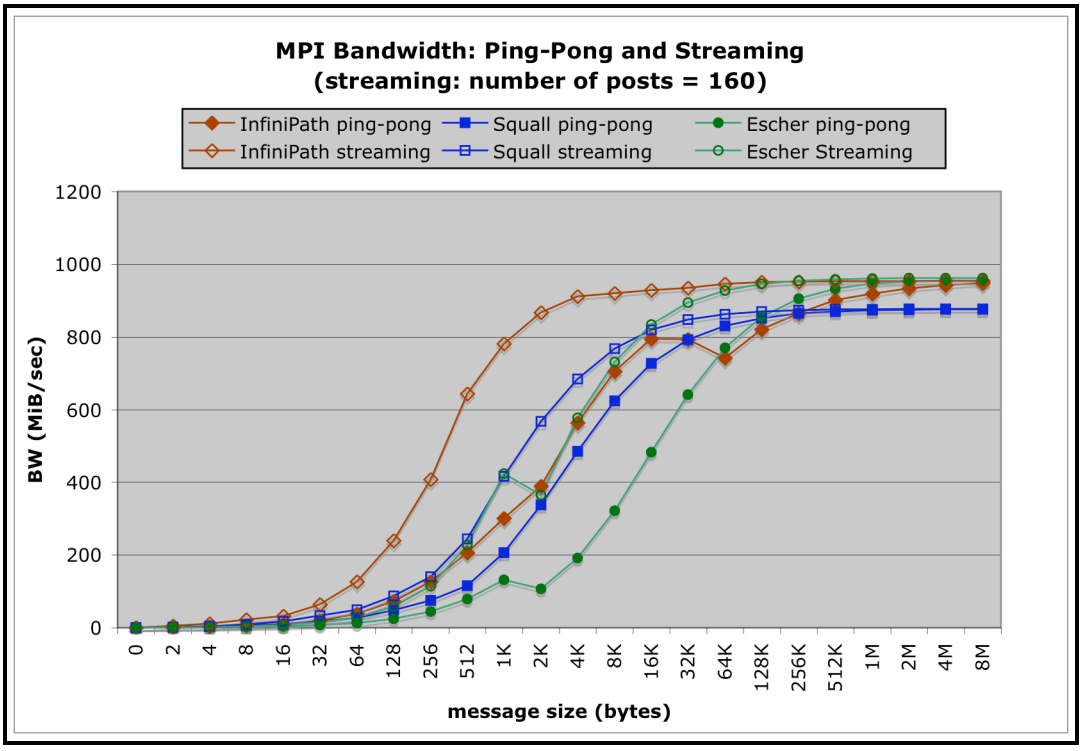

Figure 2: Ping-Pong and Streaming Bandwidth

Table 4: Peak Bandwidth (MiB/sec)

Platform	Ping-Pong	Streaming
InfiniPath	949	954
Squall	876	877
Escher	955	962

Table 5: One-Half Peak Bandwidth Message Size (bytes)

Platform	Ping-Pong	Streaming	% Increase
InfiniPath	3579	417	859%
Squall	3393	1614	210%
Escher	19663	3570	551%

4.1.2. PMB SendRecv

The Pallas MPI Benchmarks version 2.2.1 (PMB) were used to measure MPI_Sendrecv() bandwidth. The results are shown in Figure 3. This benchmark reports the aggregate bandwidth. For the Squall cluster, the PCI-X bus is limiting the bandwidth, and provides evidence that the PCI-X bus is limiting the uni-

directional bandwidth reported in the previous section. The peak rate is slightly less, due to the overhead associated with negotiating the PCI-X bus. Peak values are reported in Table 6.

The InfiniPath cluster shows a change in protocol between message sizes of 32K and 64K bytes. The author does not have knowledge of the different protocol levels, but it is interesting to see that the InfiniPath results correlate very closely with the Escher results for large message sizes. [3] This was also seen in the uni-directional ping-pong tests. It is also interesting that for large messages, the Escher results show a slight advantage over the InfiniPath cluster.

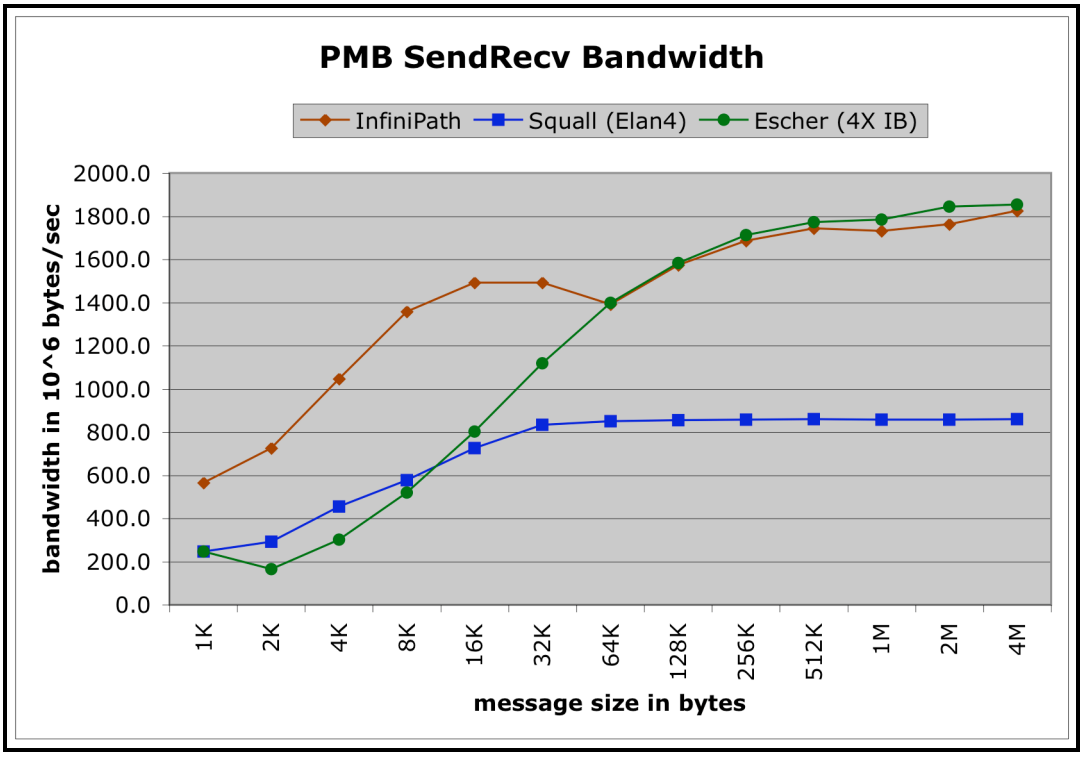

Figure 3: PMB SendRecv Bandwidth

Table 6: PMB SendRecv Peak Bandwidth

Platform	Ping-Pong
InfiniPath	1826
Squall	860
Escher	1855

4.2. 16-Node Collective Results

The PMB benchmark was used to measure collective performance. The Exchange, Allreduce, Alltoall and Broadcast benchmarks were analyzed. The Exchange benchmark exchanges data with its MPI neighbors, performing a non-blocking send to nodes N-1 and N+1, then performing simultaneous non-blocking receives for each neighbor. Hence, it's a measure of bi-directional bandwidth of the node. The Allreduce benchmark measures the performance of the MPI_Allreduce() call. Likewise for Alltoall and Broadcast.

[3] Pathscale has indicated that newer versions of their MPI software smoothes out the "notch" seen between 16K byte and 128K byte message sizes.

The respective results are shown in Figures 4, 5, 6 and 7. The Escher cluster is not included due to its size of 8 nodes.

The InfiniPath results for the Exchange benchmark show good scaling up to the 64K message size, where the performance falls and then the bandwidth increases as a function of message size, which correlates with its ping-pong test characteristics. The Squall results show an increase in bandwidth as the message grows until the PCI-X bus is saturated at a message size of 32K. The MPI_Allreduce() algorithm is a tree based algorithm and hence represents the performance of the uni-directional send and receive calls, which are similar for the two interconnects. Due to its lower latency and higher peak bandwidth, the InfiniPath cluster performs slightly better. The Alltoall benchmark demonstrates a significant advantage, approximately 10X, for the InfiniPath cluster for small messages. This advantage lessens as the message size increases. The smaller message size advantage is most likely due to the streaming performance of the InfiniPath interconnect, which can be utilized in an MPI_Alltoall() call. The Broadcast benchmark shows an advantage of approximately 3X for small messages. At a message size of 1K bytes, the two interconnects perform similarly.

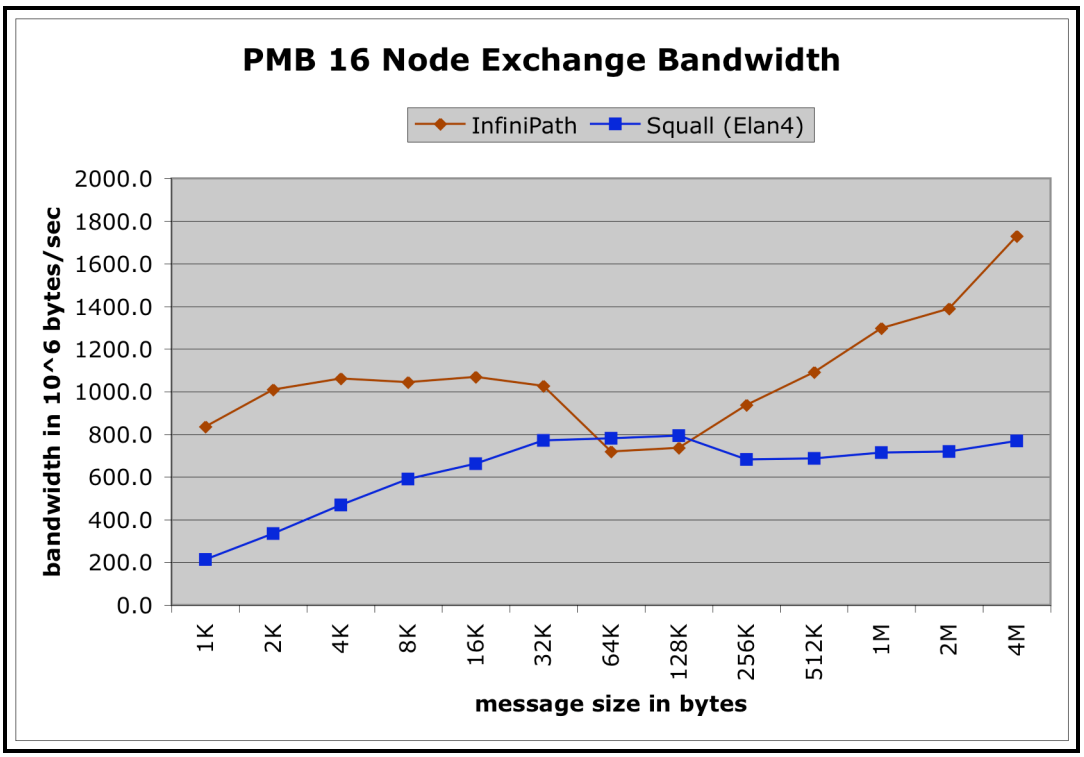

Figure 4: PMB Exchange Results

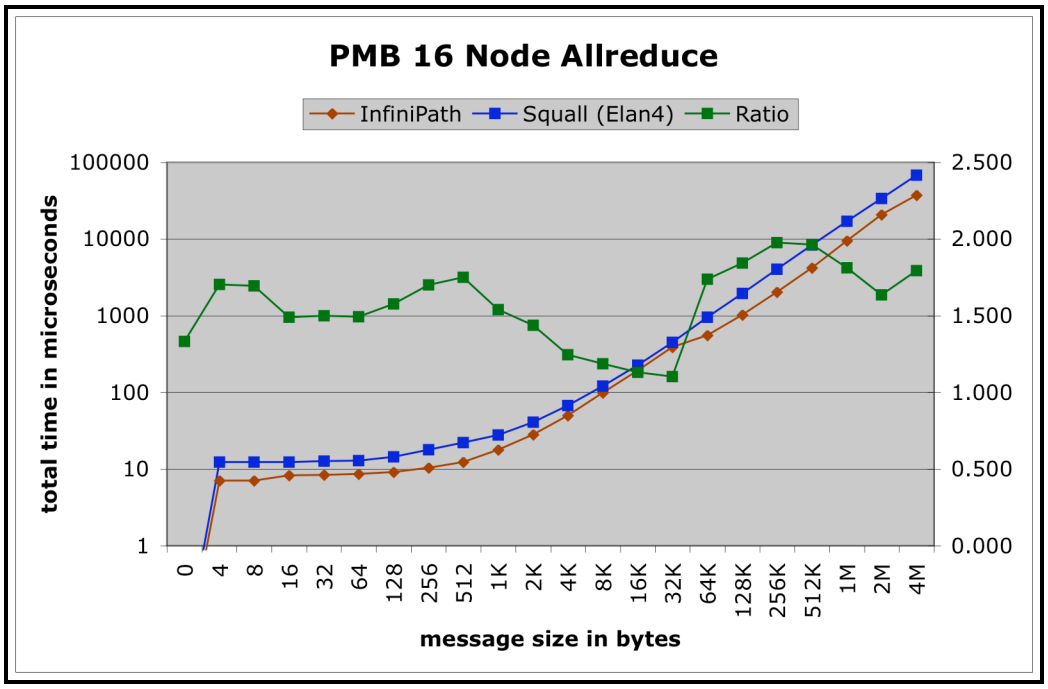

Figure 5: PMB Allreduce Results

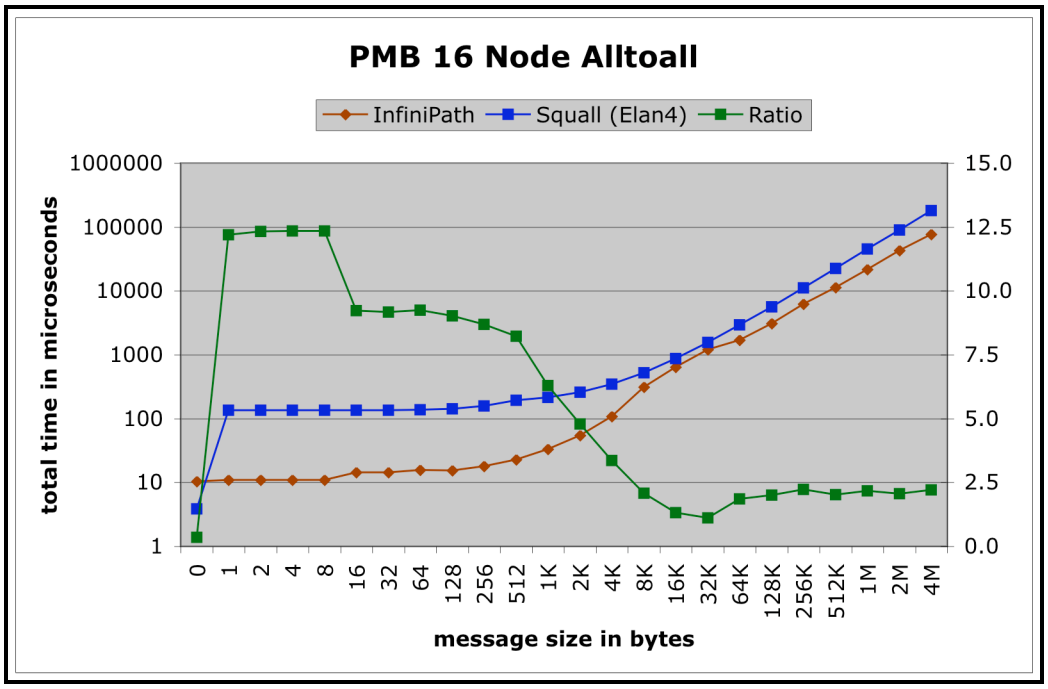

Figure 6: PMB Alltoall Results

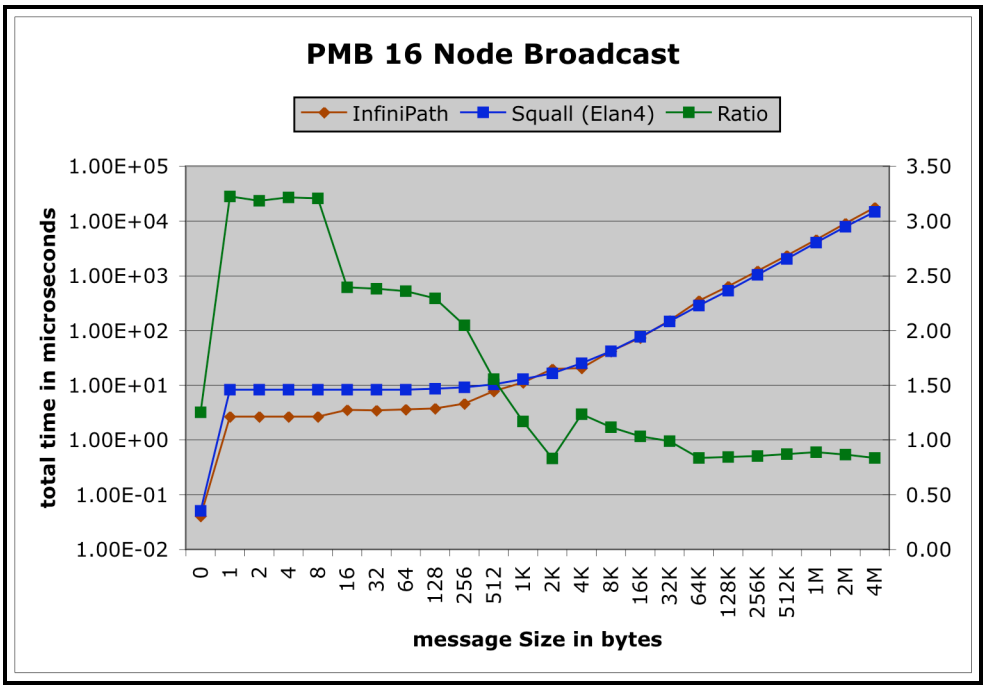

Figure 7: PMB Broadcast Results

4.3. Two Process per Node (2 PPN) Results

The MPI performance for the case of two processes per node was also analyzed. Most clusters are purchased with 2-way nodes and scheduled with two MPI processes, one for each CPU, so it is important to understand how an interconnect performs when two application processes are contending for the same resource. Only the MPI_Allreduce() collective performance is discussed as it is the collective dominant in the LAMMPS results presented in the next section. [1]

4.3.1. PMB PingPong 2 PPN

Figure 8 plots the latency of two MPI processes on the same node, i.e. intra-node latency. It is assumed that all three interconnects use a shared memory message passing protocol for intra-node communications. All three demonstrate approximately 1 microsecond performance for small message sizes. Figure 9 plots the bandwidth achieved with intra-node communications. The InfiniPath cluster's performance increases steadily as the size of the message increases. Squall's bandwidth increases in a similar fashion and peaks at around the 256K byte message size, and then decreases for message sizes larger than 256K bytes. Both of the interconnects demonstrate better performance than the inter-node performance using the interconnects link. Escher's intra-node bandwidth plot provides lower performance for all message sizes, with the exception of those messages that are 64K to 128K bytes in size. This characteristic is very strange and further analysis of the protocol stack is required to understand this behavior. It should be noted that the software revision of the MPI software stack used on Escher is dated, and newer implementations should be investigated before any conclusions are made.

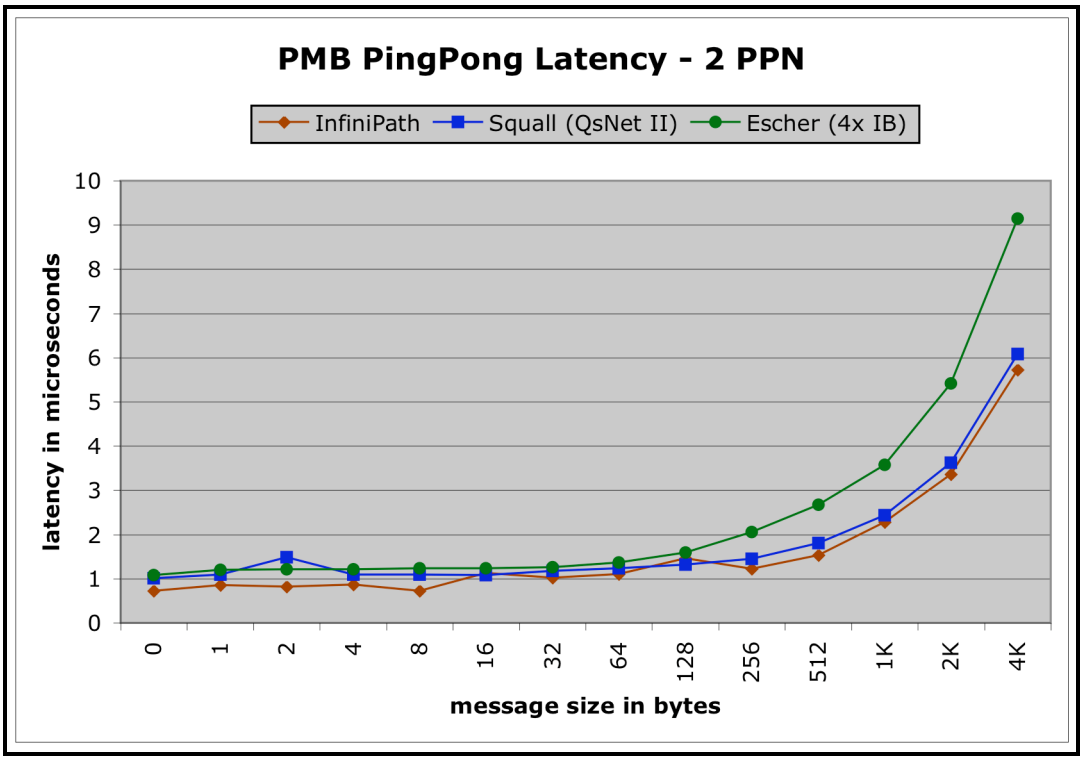

Figure 8: 2 PPN PMB PingPong Latency

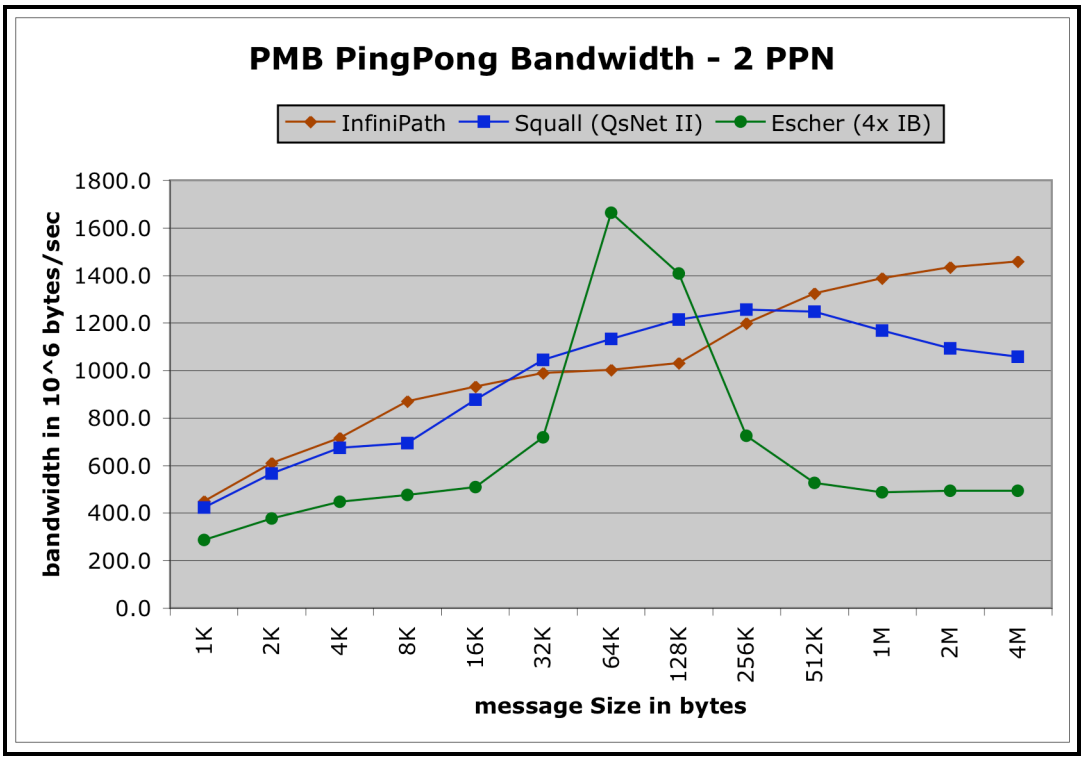

Figure 9: 2 PPN PMB PingPong Bandwidth

Table 7: 2 PPN PMB PingPong Zero Byte Latency

Platform	Zero Byte Latency
InfiniPath	0.72
Squall	1.01
Escher	1.08

Table 8: 2 PPN PMB PingPong Peak Bandwidth

Platform	Peak BW
InfiniPath	1458
Squall	1257
Escher	1664

4.3.2. PMB SendRecv 2 PPN

The PMB SendRecv bandwidth is shown in Figure 10, with peak values tabulated in Table 9. As with the ping-pong results, the InfiniPath cluster demonstrates a steady increase in bandwidth as the message size increases. The Squall cluster also demonstrates performance similar to the ping-pong tests, but the performance is erratic between message sizes of 4K bytes and 64K bytes. The Escher cluster is able to achieve better small message performance, relative to it's ping-pong results, but still demonstrates poor performance for message sizes larger than 128K bytes in size.

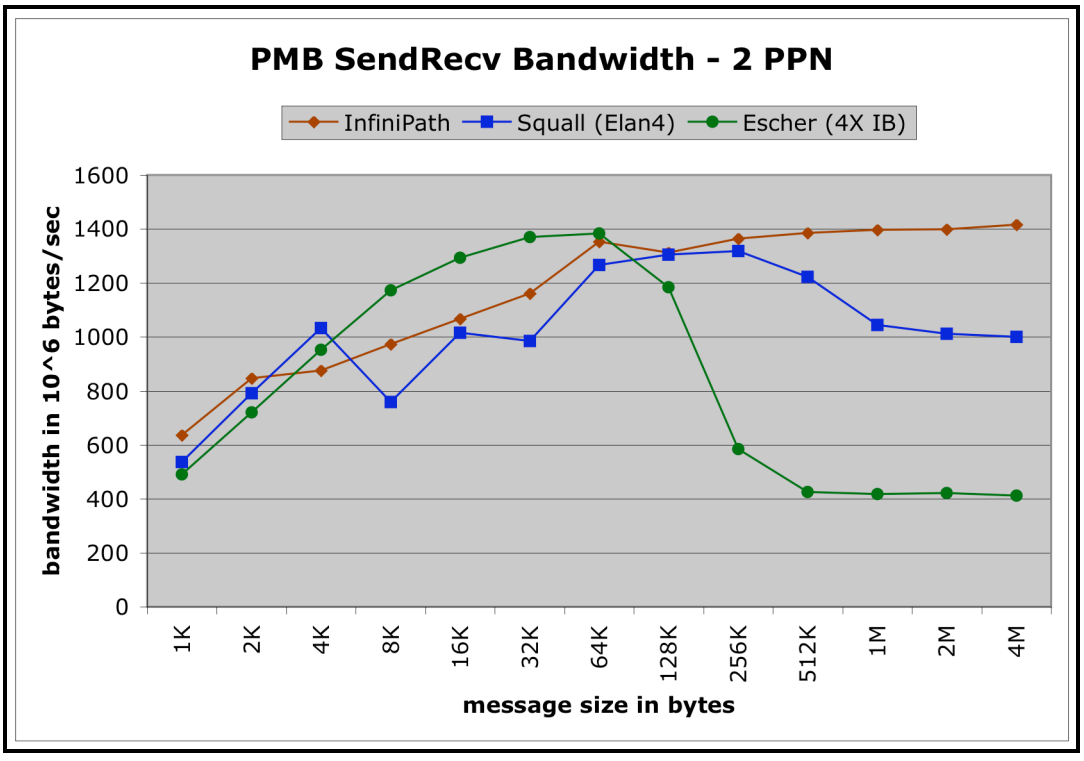

Figure 10: 2 PPN PMB SendRecv Bandwidth

Table 9: 2PPN PMB SendRecv Peak Bandwidth

Platform	Peak BW
InfiniPath	1415
Squall	1318
Escher	1383

4.3.3. PMB Allreduce 2 PPN

The Allreduce performance in 2 PPN mode is essentially equal for the two interconnects, except for the message size range of 4KB to 32KB where the Squall cluster shows a slight advantage. Note that this result is in contrast to the 1 PPN results in which the InfiniPath results showed an advantage for all message sizes. This is an indication that for the LAMMPS application in 2 PPN mode, the InfiniPath cluster may not scale as well in 2 PPN mode as it does in 1 PPN mode.

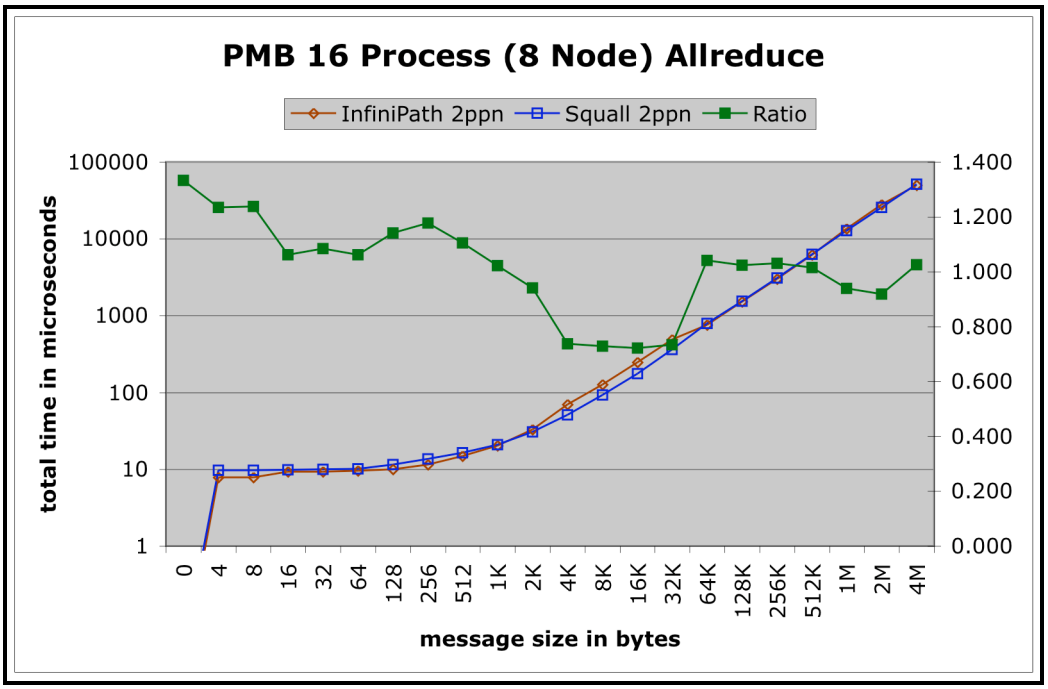

Figure 11: 2 PPN PMB Allreduce

5. Overlapping Computation and Communication

A key attribute for a message passing interface is its ability to offload communication processing from the host processing. I.e. it's ability to overlap application computation with application communication. The InfiniPath HCA architecture does not contain a programmable offload processor, but instead relies on the power of the host processor to manage the HCA. In order to better understand this tradeoff, a benchmark was developed to characterize a NICs ability to perform application computation and communication overlap.

In addition to the three clusters investigated in the other sections of this study, the Myrinet 2000 interface using the GM Myrinet Control Program was also analyzed. The Myrinet cluster was not included in the other sections due to its limited bandwidth, and hence it is not that interesting for comparison. However, this particular test is not a latency, bandwidth, etc performance test, it's a test of a network cards ability to offload the host processor. In this context, Myrinet is a viable test platform. The Myrinet NIC running the GM protocol is widely used in today's commodity clusters and serves as a good "industry standard" protocol for comparison purposes. The Myrinet test cluster for this study is the Spirit Cluster at SNL. In addition to using Myrinet 2000 and GM, Spirit uses Hewlett-Packard DL135 servers with an Intel 3.4 GHz EM64T processor.

The benchmark is modeled after the ping-pong benchmark for MPI, but instead of simply passing a message of size N between the two nodes, one node performs "work" for a period of time between issuing a send (or receive, the benchmark can measure both directions) with a non-blocking call and then checking for completion of the operation with the MPI_Wait() call. [4] The work interval is increased until the total time for the message transmission begins to increase, i.e. until the time to do work begins to impact the overall transmission time. This work interval is then timed without messaging. The overhead time then becomes the difference between the nominal transmission time and the work time. Another way to interpret the overhead time is to view it as the effective latency associated with a message transfer when it is overlapped with computation. I.e. if overhead time during the transfer is 0, then the effective latency seen by the application for that transfer is 0. As the overhead time increases the effective latency seen by the application increases and equals the overhead time.

The results of the MPI overhead test are shown in Figures 12, 13 and 14. Figure 12 plots the overhead time, or effective latency, as a function of message size. In Figures 13 and 14, the host availability is plotted as a function of message size. Host availability is defined as the percentage of the total transfer time that the application is able to perform other work.

For small message sizes the InfiniPath interconnect shows the lowest effective latency. The Escher and Spirit clusters perform similarly and the Squall cluster shows the largest effective latency. However, as the protocols switch from a short message protocol to a long message protocol the Squall cluster has increasingly less host involvement as compared to the other clusters. The results of the receive operation are similar. Thus, if an application passes large messages between nodes and attempts to overlap the communication with the computation, the Squall cluster may provide the best performance even though it's absolute bandwidth is less. But of course, this depends on the message size and the amount of overlap achieved by the application.

[4] It should be noted that returning from an MPI_Wait() call only indicates that the message buffer passed to the non-blocking call can be reused. For a send operation, returning from MPI_Wait() does not imply that the message has been received by destination process.

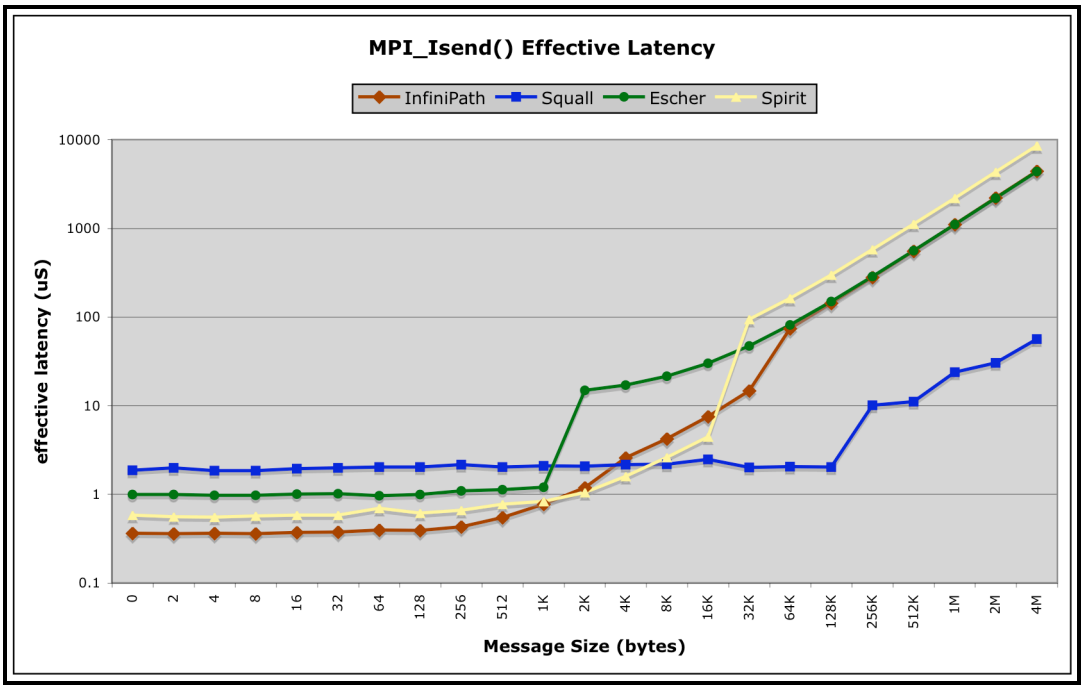

Figure 12: Effective Send Latency

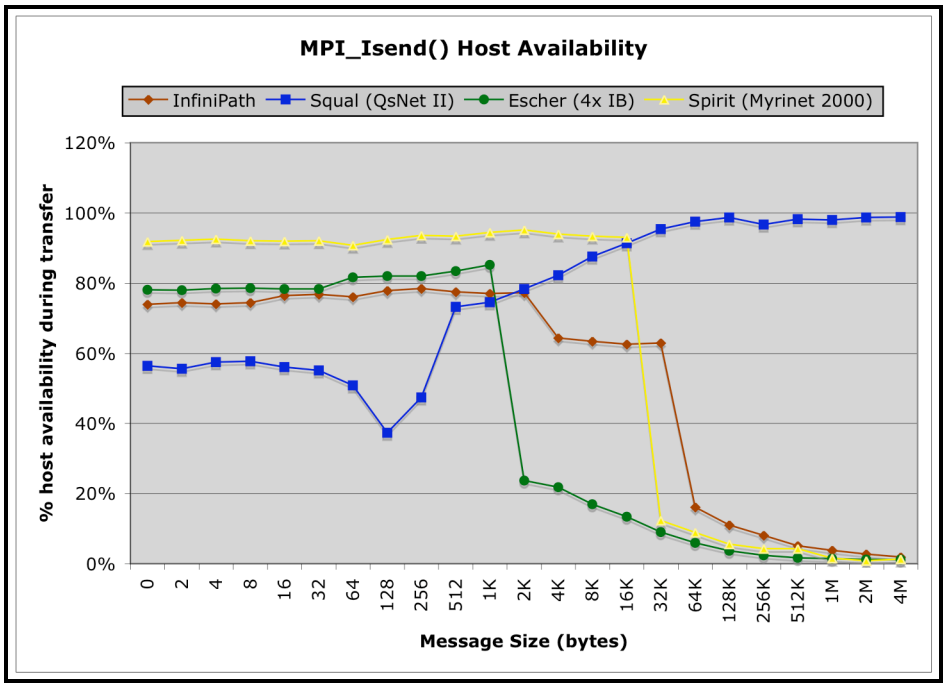

Figure 13: Host Availability during an MPI_Isend()

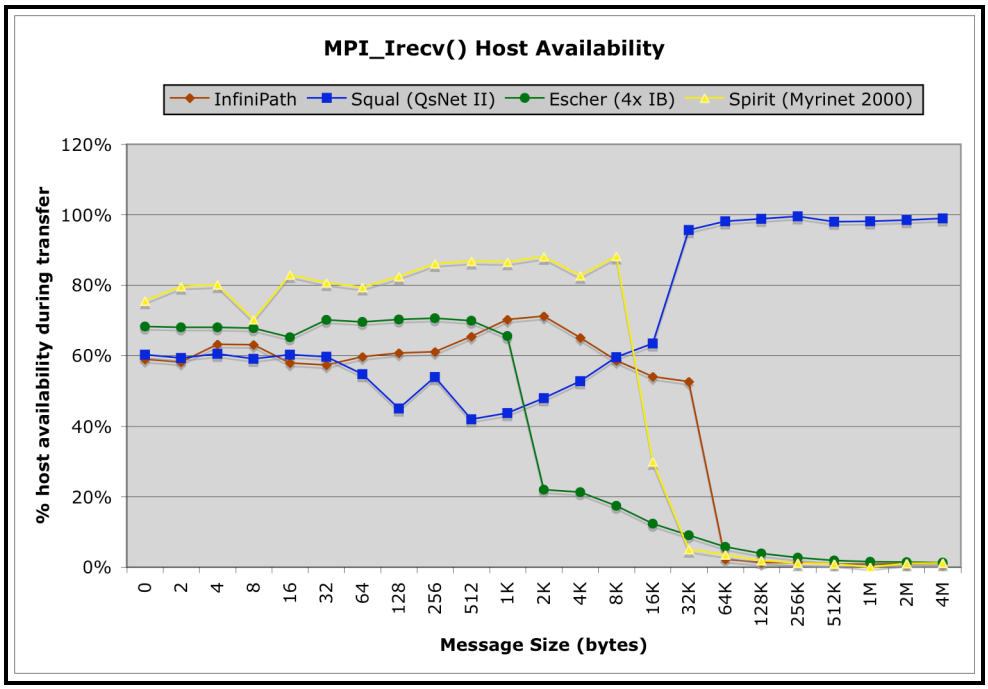

Figure 14: Host Availability during an MPI_Irecv()

6. Application Results

The LAMMPS application is the only Sandia application that was benchmarked in this study. The majority of SNL applications have an export control restriction and hence it was not possible to benchmark them on the InfiniPath cluster as it resides on an open network at the Pathscale site. However, the LAMMPS application is a good application for a study in that numerous benchmark problem sets are provided. In addition, the benchmark comes in a Fortran 90 version, LAMMPS 2001, and a C++ version, LAMMPS 17Jan05. Of the available benchmark problem sets, the Stouch and LJ problems were chosen because they provide a good balance of computation with communication and hence will demonstrate scalability problems if the interconnect does not perform well.

Although this study is focused on communication performance, it should be noted that the single process Opteron runtimes for the InfiniPath and Squall clusters is significantly better than that of the EM64T processor used in the Escher cluster. Despite the EM64T having a clock rate of 3.4 GHz and the InfiniPath and Squall Opteron clock rates of 2.6 GHz and 2.2 GHz respectively.

6.1. LAMMPS 2001 Stouch Study

The results of the Stouch study are shown in Figure 15 for two modes of operation, 1 PPN and 2 PPN. A few interesting observations are evident. The first is that the Squall cluster performs very well in 2 PPN mode, were as the other clusters show better performance in 1 PPN mode. The 1 PPN and 2 PPN results for the Squall and Escher clusters parallel each other as the size of the problem set grows, i.e. the difference for a given job size remains relatively constant. However, the difference between the 1 PPN and 2 PPN InfiniPath results diverge at a constant rate up to 16 nodes.

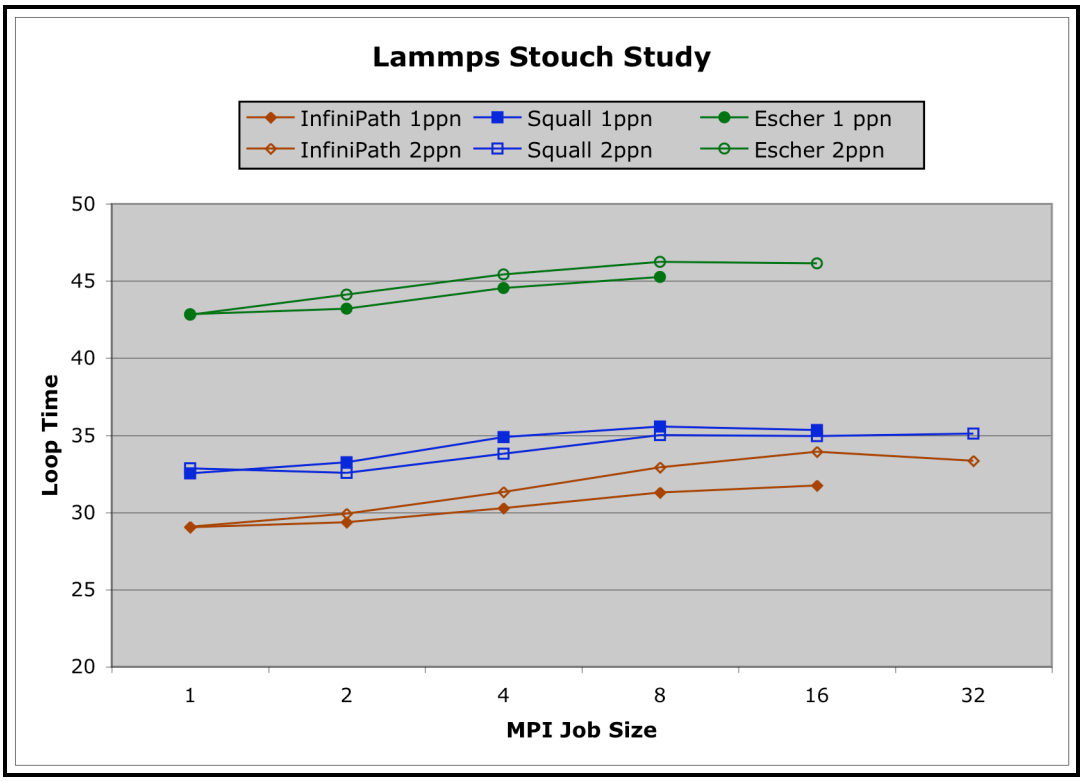

Figure 15: LAMMPS 2001 Stouch Study

6.2. LAMMPS 17Jan05 LJ Study

The results of the LJ study are shown in Figure 16. Again, the 2 PPN results for Squall show better runtimes than the 1 PPN results. This combined with the Stouch results demonstrates that its intra-node performance is more efficient than the inter-node performance. Note that for this problem, the InfiniPath cluster also shows better performance in 2 PPN mode. This is in contrast to the Stouch study.

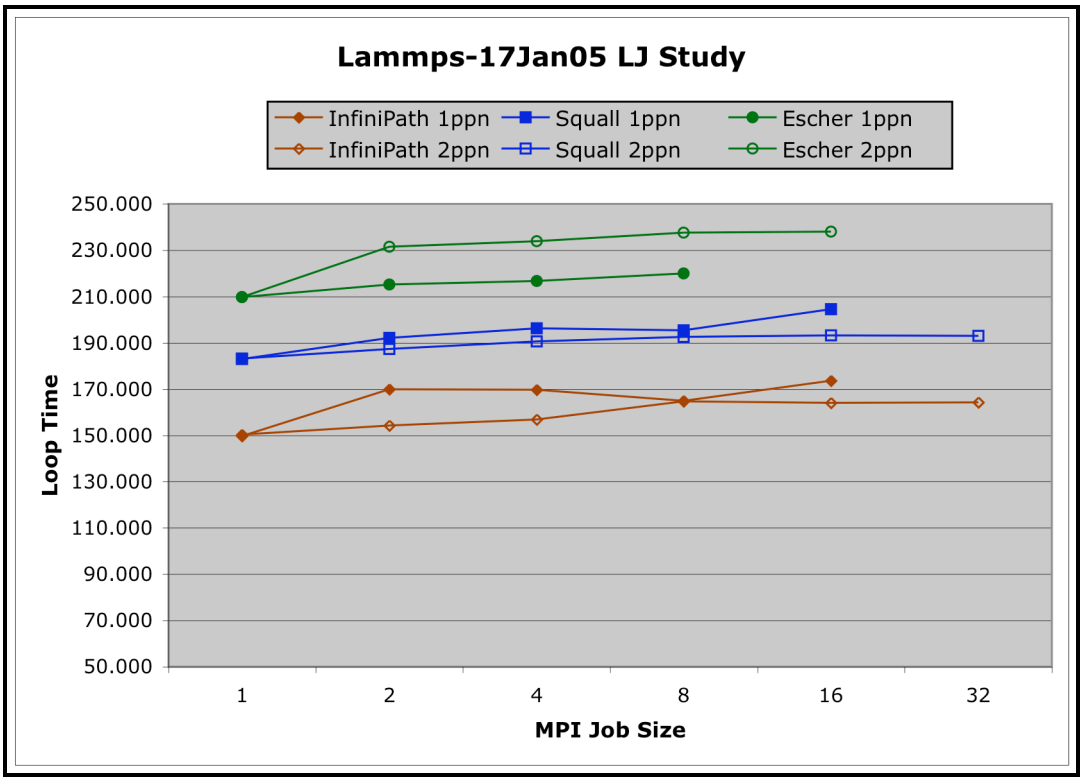

Figure 16: LAMMPS-17Jan05 LJ Study

7. Conclusions

The InfiniPath cluster exhibited the best performance in absolute zero-byte latency and peak bandwidth, when compared to other clusters using Quadrics Elan4 (Elite II) and Voltaire's 4X IB high-speed interconnects. Both ping-pong and streaming message passing benchmarks were performed and the InfiniPath interconnect is optimized for streaming performance, showing a significant improvement of it's ping-pong performance on the streaming benchmarks. For the collective benchmarks, the InfiniPath interconnect worked well and showed an advantage over the other networks in most cases. And in particular the PMB Alltoall benchmark for small messages.

The ability to overlap computation and communication was analyzed for the three test clusters, and in addition Myrinet 2000 with GM was added as a test platform. The InfiniPath interconnect does a good job of offloading the host CPU for small message sizes, but the host CPU becomes less available when long message protocols are used. This was seen on the IB and Myrinet interconnects also. The Elan4 interconnect has more overhead for small message sizes, but for large messages host availability is high. This test shows that an application that implements overlap of computation and communication and moves it's data using large message transfers will perform well on an Elan4 based cluster.

The only SNL application benchmark suitable for use on the open network CBC cluster is LAMMPS. For the Stouch study, the CBC cluster demonstrated scaling issues when run in the 2 PPN mode. In this mode the runtime showed a steady increase as the job size increased. This was not seen in the other clusters. In fact, for the Squall cluster the 2 PPN results scaled better than the 1 PPN results. However, for the LJ study, the CBC cluster showed very good 2 PPN scaling. In fact, it followed the trend of the Squall cluster, with the 2 PPN results scaling better than the 1 PPN results.

8. Follow-on Work

The usefulness of analyzing an interconnects ability to scale by running benchmarks on small clusters is limited. Trends can be surmised, but it's hard to tell if it's a scaling issue with the interconnect or an artifact of the application. Including results from other clusters for comparison helps to eliminate the later concern, if the benchmark shows good scaling results on a well characterized platform. It would be beneficial to repeat the tests on InfiniPath clusters of larger scale, as then all of the interactions of the network and the protocols can be analyzed, for example routing and congestion.

The use of the Voltaire 4X IB cluster in this study is somewhat flawed in that the software revision used in the study is dated. It would be more beneficial to obtain data using more recent software stacks. Which have most likely improved over the last six months.

The study is lacking a detailed analysis of the LAMMPS message passing characteristics. If this was better characterized, then it would allow better correlations, hopefully, between microbenchmark and application benchmark results.

9. References

[1] Ron Brightwell, Sue Goudy, Arun Rodrigues and Keith Underwood, "Implications of Application Usage Characteristics for Collective Communication Offload", International Journal of High Performance Computing and Networking, April 2005.
[2] Pathscale Inc. http://www.pathscale.com
[3] Quadrics Ltd. http://www.quadrics.com
[4] Voltaire Inc. http://www.voltaire.com

Distribution:

1	MS-0316	Sudip Dosanjh	01420	
1	MS-0321	Bill Camp	01400	
1	MS-0376	Ted Blacker	01426	
1	MS-0816	Jim Ang	01422	
1	MS-0822	David White	01427	
1	MS-0823	Carl Leishman	04324	
1	MS-0823	John Zepper	04320	
1	MS-1094	Rob Leland	04300	
1	MS-1110	Neil Pundit	01423	
1	MS-9158	Mitchel Sukalski	08961	
1	MS-9018	Central Technical Files		8945-1
2	MS-0899	Technical Library		9616