Comparative Performance of InfiniBand Architecture and Gigabit Ethernet Interconnects on Intel® Itanium® 2 Microarchitecture-based Clusters

Authors:

Dr. Lars E. Jonsson
Dr. William R. Magro
Intel Americas, Inc., Champaign, Illinois, USA

Correspondence:

Lars E. Jonsson
Intel FX1
1906 Fox Drive
Champaign, IL 61820

Tel: +1-217-356-2288
Fax: +1-217-356-5199
e-mail: lars.e.jonsson@intel.com

Keywords:
MPP-DYNA, parallel performance, speedup, InfiniBand Architecture, Intel® Itanium® Architecture, single precision, scaling

ABSTRACT

The Intel® Itanium® 2 microarchitecture is based on a 64-bit processor architecture that is ideally suited to compute-intensive applications such as LS-DYNA. In fact, Hewlett Packard* has demonstrated outstanding performance of their Itanium 2-based systems on a range of technical computing applications, including LS-DYNA[1]. Given the success of the platform in achieving performance on a per-processor basis, we turned our attention to the speedups achievable with tightly coupled clusters of Itanium architecture-based servers. In this paper, we study the scalability of an Itanium 2-based server cluster consisting of 4-CPU SMP nodes, connected by gigabit Ethernet* and by a high-performance InfiniBand* Architecture interconnect. We study the relative performance of these interconnects, relating the observed application-level performance to the underlying performance characteristics of the interconnect.

INTRODUCTION

The MPP version of LS-DYNA provides significant runtime reductions on many large models, when run on multiple CPUs in large SMP systems or in clusters. The two components of delivered performance are the per-CPU, or architectural, performance and the *speedup* achieved by harnessing the power of many tightly coupled processes. While large SMP systems typically provide very good inter-process communication performance, cluster communication performance is dependent on a number of factors, ranging from the I/O architecture of the node to the interconnect technology to the switching "fabric" that connects the nodes. The goal of this paper is to study the efficacy of a new, high-performance interconnect technology called InfiniBand Architecture in delivering run-time speedups in MPP-DYNA, relative to baseline performance established by a gigabit Ethernet connection.

The key requirements for speedup of clustered parallel applications are an interconnect that allows the cluster nodes to communicate with each other as quickly as possible, both on a one-to-one basis and on a many-to-many basis. The ideal cluster interconnect, then, has the following characteristics:

1. Low latency – the time to send a small message should be correspondingly small. Latencies are typically measured in microseconds (us).
2. High bandwidth – the interconnect should be able to achieve high rate of throughput when large messages are pipelined onto the interconnect fabric. Bandwidth is typically measured in megabytes/second (MB/s).
3. Non-blocking interconnect architecture – the performance of the fabric should not degrade when arbitrary pairs of nodes communicate simultaneously. The typical performance metric is known as bi-sectional bandwidth, and it is measured in megabytes/second.

Gigabit Ethernet has now become commonplace in servers and workstations. While its signaling rate of 1 Gb/s translates to a peak bandwidth of about 120 MB/s, this performance level is rarely achieved in practice. The standard connection-oriented protocol carried on Ethernet is TCP/IP. Because the host CPU is generally responsible for implementing the compute-intensive TCP/IP software stack, one typically observes a relatively large latency, in the range of 50 – 120 us, for zero-byte messages. Further, the CPU time spent processing TCP/IP connections is no longer available to the application software.

The industry-standard InfiniBand Architecture (IBA) was introduced by the InfiniBand Trade Association[2] to address some of these limitations for the data center and high performance computing center environments. IBA is based on a 2.5 Gb/s signaling rate and is available in "widths" of 1x, 4x, and 12x. Each host connects to the IBA switch fabric via a local interface

known as a host channel adapter (HCA). The 4x configuration is the most common, providing link speeds of 10 Gb/s or 1.25 GB/s between hosts.

An important feature of the architecture is its use of "verbs", commands to the HCA that allow it to transfer messages to and from remote nodes' memories on behalf of the host operating system and application programs and without remote host CPU assistance. Because the HCA implements the IBA protocol in hardware and sends and receives directly from system memory, it is possible to achieve a significant fraction of the link speed with very low host CPU utilization. For example, Mellanox* has demonstrated their HCA silicon sending over 800 MB/s with less than 0.5% host CPU utilization on a Linux* server. The HCA-level protocol processing, teamed with other advanced features of the IBA architecture, result in low message latencies, while leaving the host CPU free for use by applications. These capabilities allow overlap of communication progress among nodes with application-level computation.

The final component of interest in a cluster is the interconnect "fabric," typically one or more switches that tie the hosts together. Ethernet protocol packets are routed to their destinations dynamically by switches and routers that build and maintain internal tables of source and destination information. While this architecture is well-suited to the ever-changing multiple connections of the Internet, the time spent looking up addresses in tables can add significant latency to message delivery time at each "hop" through the network. These latencies are typically in the range of a few to a few tens of microseconds. In contrast, routes through an InfiniBand fabric are determined by a subnet manager, and then communicated to the nodes in the network. As a result, the network is encoded with deterministic and quasi-static routing information that allows very fast packet switching. As a result, InfiniBand switching delays are frequently just a small fraction of a microsecond per hop.

Because tightly coupled parallel applications like MPP-DYNA typically involve alternating phases of intense computation and bursts of communication among all nodes, it is important that the switching fabric provide multiple routes connecting each pair of nodes. One design that addresses this need is the so-call "fat tree" topology, as shown in Figure 1.

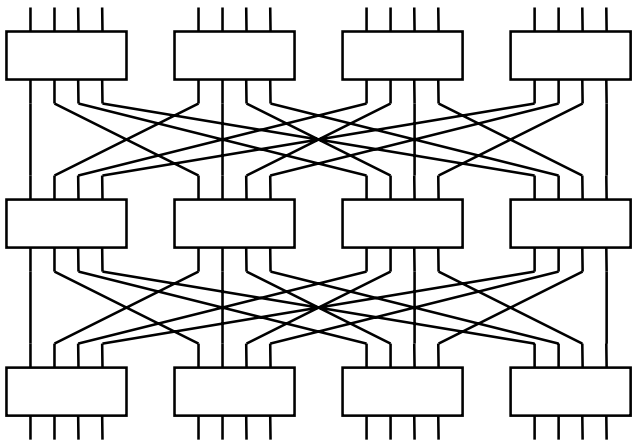

Figure 1 A fat tree switching network with constant bi-sectional bandwidth. The switch is composed of 12 interconnected 8-port switches, providing 32 external ports to cluster nodes.

This topology provides a symmetric view of the cluster to each node. InfiniBand Architecture is ideally suited to such a switching topology, and InfiniBand vendors are beginning to provide high-performance switch products with this very topology. When teamed with a subnet manager that generates appropriate routes, a fat tree network can provide the large and constant bi-sectional bandwidth that maximizes message throughput during the characteristic bursts of MPP-DYNA communication.

The remainder of the paper is organized as follows. We first report the configuration of our test cluster, followed by the performance results obtained on that cluster. We then offer some discussion and performance analysis of those results and conclude with a summary and a number of possible future investigations.

TEST CONFIGURATION

Node Hardware

Our cluster was built by connecting eight Intel Server Platform SR870BN4 compute nodes[3]. Each node was configured with four Intel Itanium 2 processors 1.0 GHz with 3 MB L3 cache and 2 GB of registered ECC CL2.0 DDR PC200 memory.

The SR870BN4 server has a number of relevant performance features. It offers memory bandwidth of 6.4 GB/s, peak performance of 16 GFlop/s per node, a dual-channel SCSI Ultra320 controller for fast disk data transfers, and six separate PCI buses. Three of these buses drive 3 PCI-X 64-bit/133 MHz slots, which are required for fast InfiniBand messaging among nodes.

Interconnect

We used an add-in 32-bit 66 MHz PCI Intel 82540EM gigabit Ethernet controller in each node and wired them to an HP Procurve* switch 2724 24-port gigabit switch with category 5e cables. The switch specifications state a maximum per-hop latency of 12 us.

Each node was also configured with an InfiniBand Architecture 4x host channel adapter, the InfiniCon InfiniServ* 7000. This adapter offers two 10 Gb/s ports, of which we used just one. The InfiniBand HCAs were cabled to an InfiniCon InfinIO* 7000 shared I/O and clustering system. The InfinIO 7000 is a rack mount chassis design that accepts various line cards, including InfiniBand switch modules, gigabit Ethernet gateways, and Fibre Channel I/O gateways. Our InfinIO 7000 was equipped with 2 6-port InfiniFabric* switch modules and one 6-port InfiniBand Expansion line card, for a total of 18 InfiniBand 4x ports. The InfinIO is rated at 0.11 us latency per switch hop, roughly 100x faster than the gigabit Ethernet switch. It is also capable of switching InfiniBand traffic at the full 10 Gb/s line speed.

Software

The nodes ran the following system software:

- RedHat* Advanced Server 2.1
- Linux kernel-2.4.18-tpc.0.18smp
- glibc-2.2.4-31.7
- util-linux-2.11f-20
- elilo-3.3a-1
- Intel gigabit Ethernet driver 4.6.11
- InfiniCon InfiniServ 7000 drivers

We loaded the following software tools and application software:

- Intel® Fortran Compiler 7.1

- MPIch 1.2.3 (for gigabit Ethernet experiments)
- InfiniCon MPI [OSU MVAPICH 0.8 + Argonne MPIch 1.2.3 + Berkeley Lab MVICH 1.0] (for InfiniBand Architecture experiments)
- Scali MPI Connect* 4.0 Beta 3 (for additional InfiniBand Architecture experiments[1])
- MPP-DYNA 960.1715 (built for single precision, using the Intel Fortran compiler and MPI versions above)
- Note: We built the MPI libraries using a configuration that targeted the Intel Fortran compiler. Failure to do so results in libraries targeted at the GNU g77 compiler.

Workload

Our workload was the well-known refined Neon test case, derived from the original NCAC model. The workload represents a 535k element model of a Neon car impacting a solid barrier. The termination time was set to 30 ms. No other modifications were made to the input data set. We specified memory=100M and a pfile that requested default decomposition and declared local scratch and global directories. The global directory was provided by an NFS server.

MEASUREMENTS AND RESULTS

MPP-DYNA uses the community-standard Message Passing Interface (MPI) to send and receive messages to other nodes through the interconnect network. A well-implemented MPI implementation adds little overhead to the latencies of the underlying network and exploits that network's features to deliver fast point-to-point and collective messaging. There are a number of MPI implementations available for both Ethernet and InfiniBand. We began by measuring basic performance characteristics of the networks through MPI.

The key measurements are latency and asymptotic bandwidth. We measured latency as half the round trip delivery time of a zero-size MPI message. The results of MPIch/Ethernet and MVAPICH/InfiniBand are shown in Table 1.

	Latency (us)	Bandwidth (MB/sec.)
Gigabit Ethernet	60-125	45-50
InfiniBand Architecture	13-17	560-610

Table 1 Latency and bandwidth of gigabit Ethernet and InfiniBand 4x connections as measured through MPI.

The best gigabit Ethernet latency measured about 60 us, but we saw frequent variability up to a doubling of that latency. These longer latencies may be attributable to resends of unacknowledged packets. We observed fairly low achieved bandwidth through the Ethernet connection, relative to the theoretical peak value discussed above. There exist many configurable parameters in the Linux networking infrastructure and in the Ethernet drivers, such as increased buffer sizes and larger maximum transmission units. While tuning of these parameters can lead to substantially better performance, we chose to measure using default values, as we do not expect most LS-DYNA users to perform this type of system-level tuning on their clusters.

[1] The InfiniBand performance data presented here used the MVAPICH MPI implementation from Ohio State University. While this is the highest performing InfiniBand MPI implementation of which we are aware, it is not a commercial product. As such, we also built and tested MPP-DYNA over the InfiniBand and Ethernet interconnects with the commercially supported MPI from Scali AS. All tests performed ran correctly and yielded similar parallel speedups to those observed with MVAPICH and MPIch, respectively.

The InfiniBand connection provided more stable results, achieving significantly lower latencies and higher bandwidth. In separate experiments, Intel® Xeon® processor-based systems have achieved InfiniBand latencies as low as 7.5 us and bandwidth exceeding 800 MB/s, using tuned HCA drivers and MVAPICH MPI. The Itanium architecture drivers and MPI have not yet been tuned. It is noteworthy, therefore, that they still achieved excellent performance results. With tuning, we expect to see further improved performance results.

We observe that a switching latency of 12 us represents a significant fraction of the overall observed Ethernet latency, while the sub-microsecond InfiniBand switching latency is insignificantly small. Again, lower latency gigabit Ethernet switches are available, presumably at higher cost.

We next measured elapsed time for the 30 ms simulation of the refined Neon impact on 1, 2, 4, and multiples of 4 CPUs up to the full 32 CPUs in our cluster. We then computed parallel speedups relative to the single-CPU time, which was identical for Ethernet and InfiniBand versions. These speedups are presented, along with ideal speedup, in Figure 2.

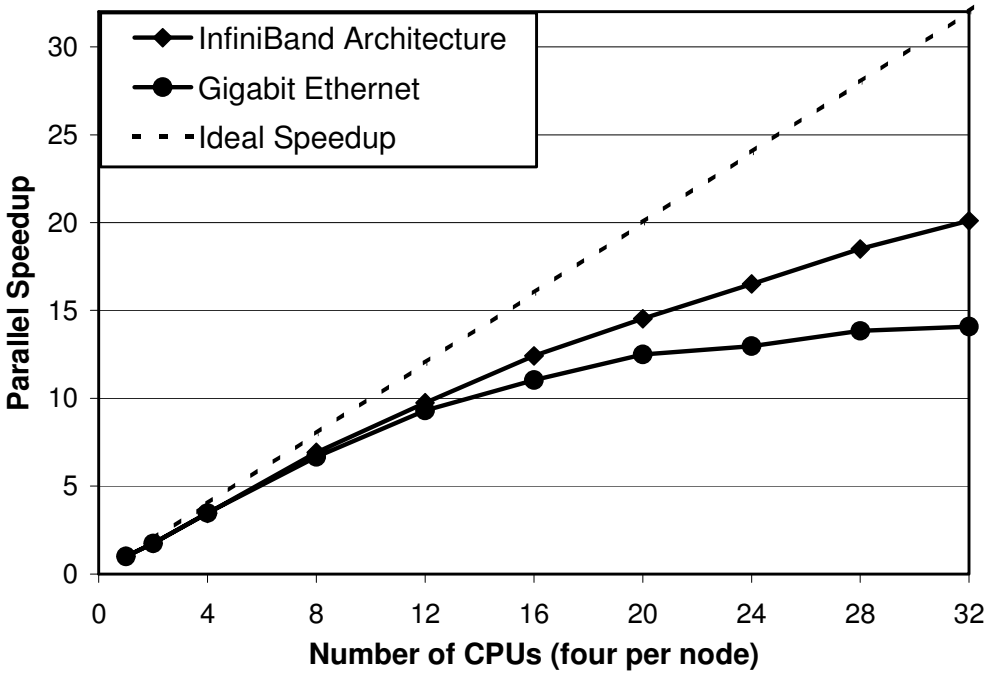

Figure 2 Parallel speedup of refined Neon 535k element 30 ms impact with a solid barrier.

Both the InfiniBand and the gigabit Ethernet performance is monotonically and smoothly increasing up to 8 nodes or 32 processors. However, it is clear that the higher performance InfiniBand interconnect delivers a significant improvement in performance. At 4 nodes, the InfiniBand performance already outperforms Ethernet by a factor of 1.12x. At 8 nodes the gap widens to 1.4x. For larger node counts, we expect the performance difference to grow further.

DISCUSSION

In the largest runs over InfiniBand, the runtime was reduced by a factor of 20x on 32 processors. While impressive, it is well below the ideal speedup of 32x. Because ideal speedup can

only be approached when the communication is very fast and the work is evenly spread across the processors, we decided to investigate the impact of load imbalance on speedup.

Achieving a balanced decomposition of a complex finite element model is very challenging, especially as the number of domains grows. A load imbalance between processes often manifests itself as a variation in reported per-process compute time, since processes that finish their computations early must wait for the others before proceeding to the next time step. The MPI implementations used in this study, however, used a polling, rather than interrupt-driven, algorithm to receive messages. Polling algorithms often deliver the lowest message latency, but fully consume a CPU during otherwise "idle" time waiting for messages.

Because we observed constant CPU utilization across processes, we instead built an instrumented version of the MPIch library that reported selected performance statistics. We observed significant variation in the time spent in the communication routines across processes, which supported the suspicion that the load was imbalanced. The default decomposition strategy, recursive coordinate bisection (RCB), divides the model into patches of roughly equal element counts. Following Roh [4], we created a pfile that requested a custom decomposition:

```
decomposition {
      sy 15
      silist 2,6
}
```

This caused the RCB algorithm to partition the auto model into long, thin domains along the axis of impact. The results of this new decomposition are compared to the default in Figure 3. We ran these experiments on InfiniBand only, to minimize the impact of communications overhead on scalability.

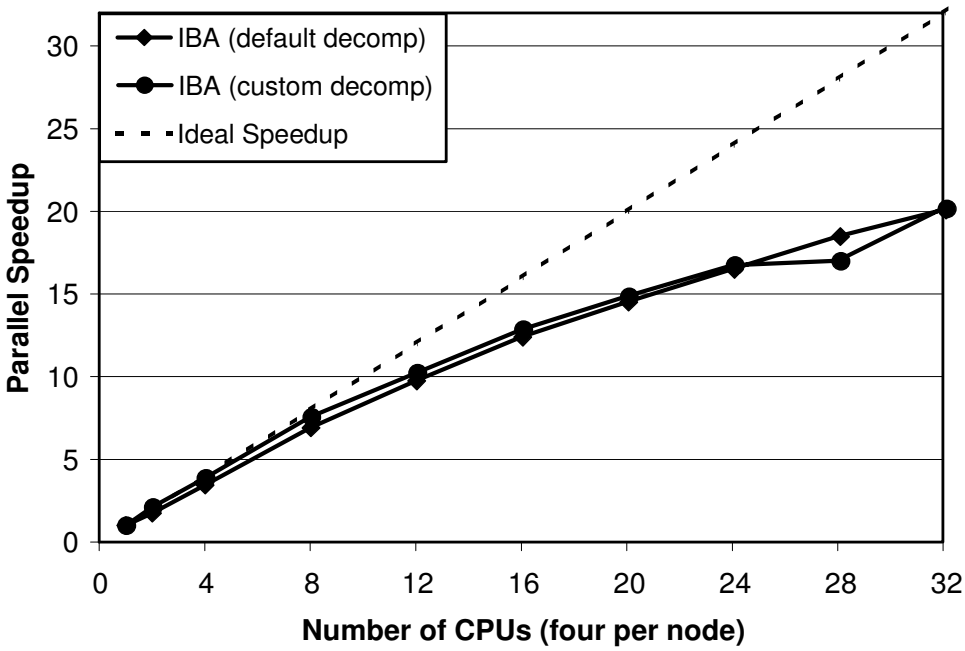

Figure 3 Parallel speedups for the default decomposition and a custom decomposition, as detailed in the text.

The most significant improvements came in the 2- and 4-CPU results. For 2 CPUs, speedup improved from 1.7x to 2.1x. For 4 CPUs, the speedup improved from 3.5x to 3.9x. The 2-CPU result is unsurprising, as it is relatively straightforward to equally divide the model into its left and right halves by partitioning along its rough axis of symmetry. As the number of domains increases, however, the problem becomes much harder, and the benefits of the custom decomposition vanish.

Because the InfiniBand interconnect outperforms gigabit Ethernet in both latency and bandwidth, we performed additional experiments to determine which factor played a larger role in the application level performance difference. Using the instrumented MPI library, we collected a histogram of message counts versus size, presented in Figure 4.

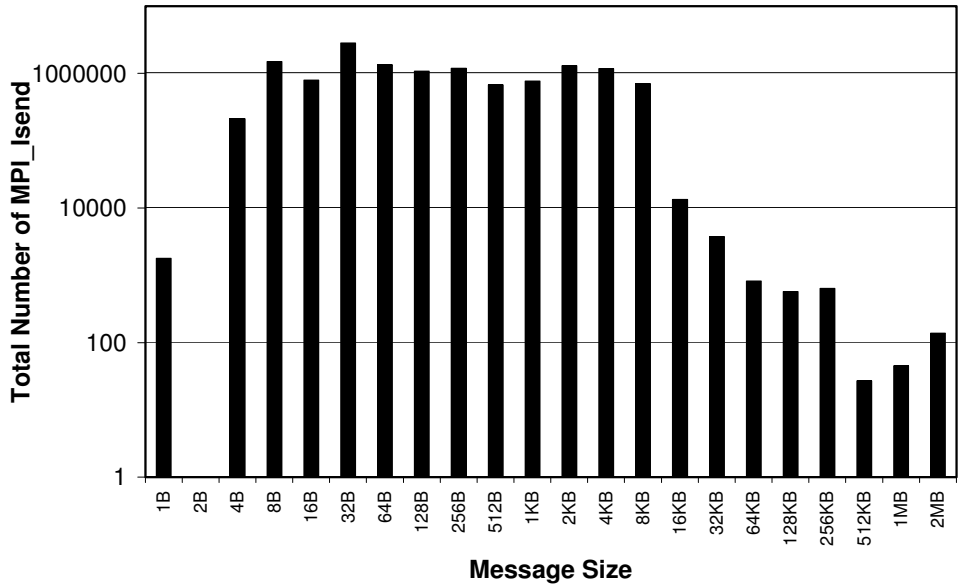

Figure 4 Histogram of MPI_Isend message counts versus size, in bytes, for a 16-cpu run. Note the logarithmic scale in the message size bins.

Noting the logarithmic scale in the message size bins, the communication patterns of MPP-DYNA are dominated by messages under 8 KB in size. Similar measurements of collective communications routines showed that most of those also involve messages of less than 8 KB.

This message pattern indicates that the lower latency of small messages on InfiniBand is the dominant factor in its superior performance on this workload. Nonetheless, there are a significant number of large messages that can take advantage of InfiniBand's higher bandwidth, which increases with message size.

Because further tuning of the Linux TCP/IP settings would improve Ethernet's bandwidth, but likely have little effect on its TCP/IP processing dominated latency, we expect a tuned gigabit Ethernet to have little performance impact at the application level. On the other hand, emerging Ethernet technologies, such as advanced controllers with TCP offload engines (available now) and remote direct memory access (RDMA), as defined by the RDMA Consortium[5], should have lower latencies and yield improved performance.

Measurements also showed that the fraction of run time spent in MPI collective communication operations grows as the number of processors increases. This is another area in which InfiniBand's superior performance to Ethernet contributes to overall performance.

SUMMARY AND CONCLUSIONS

We have presented the comparative performance of MPP-DYNA and the refined Neon workload on a 32-processor Itanium 2 microarchitecture-based cluster connected by gigabit Ethernet and by InfiniBand Architecture interconnects. We find that, while both interconnects yield parallel speedups out to 8 nodes (32 CPUs), the InfiniBand interconnect provides a 40% performance advantage over Ethernet. While InfiniBand is a relatively new technology, we observed very high performance, in terms of low latency, high bandwidth, and reliability. As an industry standard and commodity interconnect technology, we expect InfiniBand interconnects to quickly become a preferred and cost-effective interconnect solution for MPP-DYNA clusters. For very small clusters – say 8 to 16 nodes – gigabit Ethernet appears to provide reasonable performance at potentially lower cost. For larger clusters, especially those serving multiple simultaneous users, InfiniBand should be a more scalable and suitable choice.

During this study, we noted several areas for interesting future work. Because LS-DYNA uses and generates a number of large data files, we would like to measure the impact of a fast, parallel, shared file system such as Lustre* or PVFS. Likewise, the InfiniCon InfinIO 7000 offers the ability to bridge Fibre Channel devices directly into servers on the InfiniBand fabric. We would like to test the efficacy of replacing direct attached disks with large capacity, striped RAID arrays in the network file servers. Finally, as TCP offload engines and remote DMA capabilities become increasingly prevalent in Ethernet adapters, we would like to re-evaluate gigabit Ethernet as a high performance interconnect option for LS-DYNA.

REFERENCES

1. HP Web Site: Technical Performance: http://www.hp.com/products1/itanium/performance/technical/index.html
2. InfiniBand Trade Association: www.infinibandta.org
3. Intel Server Platform SR870BN4: http://intel.com/design/servers/buildingblocks/SR870BN4
4. Youn-Seo Roh, "Scaling Study of LS-DYNA MPP on High Performance Servers", 6th International LS-DYNA Users Conference 2000
5. RDMA Consortium: http://www.rdmaconsortium.org.

ACKNOWLEDGEMENTS

The authors would like to thank the following individuals for their contributions to this work:

- Drs. Chen Tsay and Jason Wang, LSTC, for providing access to LS-DYNA and the appropriate licenses needed in this work.
- Kevin Holohan and Tom Zahniser, InfiniCon Systems, for providing the InfiniBand hardware and software components used in this study.
- Prof. DK Panda, Ohio State University, for providing the MVAPICH implementation of MPI over InfiniBand.
- Scali AS, for providing Scali MPI Connect for testing with multiple interconnects.
- Drs. Tim Prince and George Chaltas of Intel, for their work in porting and validating MPP-DYNA on Linux with Intel compilers.
- Rolf Wilson, for assistance in building, configuring, and running the cluster.

DISCLAIMER

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit http://www.intel.com/performance/resources/limits.htm.

THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN ARE PROVIDED "AS IS" WITH NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. INTEL ASSUMES NO RESPONSIBILITY FOR ANY ERRORS OR OMISSIONS RELATED TO THE INFORMATION IN THIS DOCUMENT, TO PROVIDE UPDATES TO THE INFORMATION, OR TO PROVIDE NOTICE OF ANY CHANGES.

* Other names and brands may be claimed as the property of others

Intel, Itanium and Intel Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.