LS-DYNA® Best-Practices: Networking, MPI and Parallel File System Effect on LS-DYNA Performance

Gilad Shainer[1], Tong Liu[2], Jeff Layton[3], Onur Celebioglu[3]
[1]*HPC Advisory Council* [2]*Mellanox Technologies* [3]*Dell, Inc.*

Abstract

From concept to engineering, and from design to test and manufacturing, the automotive industry relies on powerful virtual development solutions. CFD and crash simulations are performed in an effort to secure quality and accelerate the development process. The recent trends in cluster environments, such as multi-core CPUs, GPUs, cluster file systems and new interconnect speeds and offloading capabilities are changing the dynamics of clustered-based simulations. Software applications are being reshaped for higher parallelism and multi-threads, and hardware configuration for solving the new emerging bottlenecks, in order to maintain high scalability and efficiency. In this paper we cover best practices for achieving maximum productivity through MPI optimizations, efficient networking utilization and usage of parallel file systems.

Introduction

High-performance computing (HPC) is a crucial tool for automotive design and manufacturing. It is used for computer-aided engineering (CAE) from component-level to full vehicle analyses: crash simulations, structure integrity, thermal management, climate control, engine modeling, exhaust, acoustics and much more. HPC helps drive faster speed to market, significant cost reductions, and tremendous flexibility. The strength in HPC is the ability to achieve best sustained performance by driving the CPU performance towards its limits. The motivation for high-performance computing in the automotive industry has long been its tremendous cost savings and product improvements - the cost of a high-performance compute cluster can be just a fraction of the price of a single crash test, while providing a system that can be used for every test simulation going forward.

LS-DYNA software from Livermore Software Technology Corporation is a general purpose structural and fluid analysis simulation software package capable of simulating complex real world problems. It is widely used in the automotive industry for crashworthiness analysis, occupant safety analysis, metal forming and much more. In most cases, LS-DYNA is being used in cluster environments as these environments provide better flexibility, scalability and efficiency for such simulations.

Cluster productivity, sometimes not measured by just how fast an application runs, is the most important factor for cluster hardware and software configuration. Achieving the maximum number of jobs executed per day is of higher importance than the wall clock time of a single job. Maximizing productivity in today’s cluster platforms requires using enhanced messaging techniques even on a single server platform. These techniques also help with parallel simulations by using efficient cluster interconnects.

For LS-DYNA, users can either choose to use local disks for the storage solution, or take advantage of parallel file system. With parallel file all nodes may be accessing the same files at the same time, concurrently reading and writing. One example for parallel file system is Lustre - Lustre is an object-based, distributed file system, generally used for large scale cluster computing. Efficient usage of Lustre can increase LS-DYNA performance, in particular when the compute system is connected via a high speed network such as InfiniBand.

HPC Clusters

LS-DYNA simulations are typically carried out on high-performance computing (HPC) clusters based on industry-standard hardware connected by a private high-speed network. The main benefits of clusters are affordability, flexibility, availability, high-performance and scalability. A cluster uses the aggregated power of compute server nodes to form a high-performance solution for parallel applications such as LS-DYNA. When more compute power is needed, it can sometimes be achieved simply by adding more server nodes to the cluster.

The manner in which HPC clusters are architected has a huge influence on the overall application performance and productivity – number of CPUs, usage of GPUs, the storage solution and the cluster interconnect. By providing low-latency, high-bandwidth and extremely low CPU overhead, InfiniBand has become the most deployed high-speed interconnect for HPC clusters, replacing proprietary or low-performance solutions. The InfiniBand Architecture (IBA) is an industry-standard fabric designed to provide high-bandwidth, low-latency computing, scalability for ten-thousand nodes and multiple CPU cores per server platform and efficient utilization of compute processing resources.

This study was conducted at the HPC Advisory Council systems center (www.hpcadvisorycouncil.com) on an Intel Cluster Ready certified cluster comprised of Dell™ PowerEdge™ M610 16-node cluster, each node with quad-core Intel® Xeon® processors X5570 at 2.93 GHz (2 CPU sockets per node or total of 8 cores per node), Mellanox ConnectX®-2 40Gb/s InfiniBand mezzanine card and Mellanox M3601Q 36-Port 40Gb/InfiniBand Switch. Each node had 24GB of memory. The Operating system used was RHEL5U3, the InfiniBand driver version was OFED 1.5, File system - Lustre 1.8.2, MPI libraries used were Open MPI 1.3.3, HP-MPI 2.7.1, Platform MPI 5.6.7 and Intel MPI 4.0, the LS-DYNA version was LS-DYNA MPP971_s_R4.2.1 and the benchmark Workload was the Three Vehicle Collision Test simulation.

The Importance of the Cluster Interconnect

The cluster interconnect is very critical for efficiency and performance of the application in the multi-core era. When more CPU cores are present, the overall cluster productivity increases only in the presence of a high-speed interconnect. We have compared the elapsed time with LS-DYNA using 40Gb/s InfiniBand and Gigabit Ethernet. Figure 1 below shows the elapsed time for these interconnects for a range of core/node counts for the Three Vehicle Collision case.

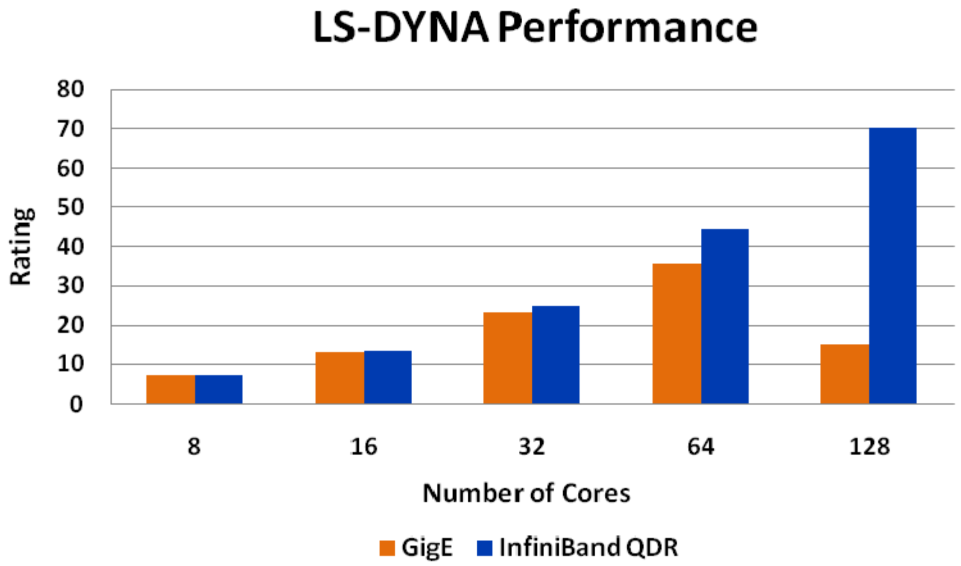

Figure 1 – Interconnect comparison with Three Vehicle Collision

InfiniBand delivered superior scalability in performance, resulting in faster run time, providing the ability to run more jobs per day. The 40Gb/s InfiniBand-based simulation performance measured as ranking (number of jobs per day) was 365% higher compared to GigE at 16 nodes. While Ethernet showed a loss of performance (increase in run time) beyond 8 nodes, InfiniBand demonstrated good scalability throughout the various tested configurations. LS-DYNA uses MPI for the interface between the application and the networking layer, and as such, requires scalable and efficient send-receive semantics, as well as good scalable collective operations. While InfiniBand provides an effective way for those operations, the Ethernet TCP stack which leads to CPU overheads that translate to higher network latency, reduces the cluster efficiency and scalability.

LS-DYNA MPI Profiling

Profiling the application is essential for understanding its performance dependency on the various cluster subsystems. In particular, application communication profiling can help in choosing the most efficient interconnect and MPI library, and in identifying the critical communication sensitivity points that greatly influence the application's performance, scalability and productivity.

LS-DYNA 3 Vehicle Collision MPI profiling data is presented in Figures 2 and 3 which show the run-time distribution between computation and communications and usage of the different MPI communications in several cluster configurations (32-cores or 4 nodes, 64-cores or 8 nodes and 128-cores or 16 nodes).

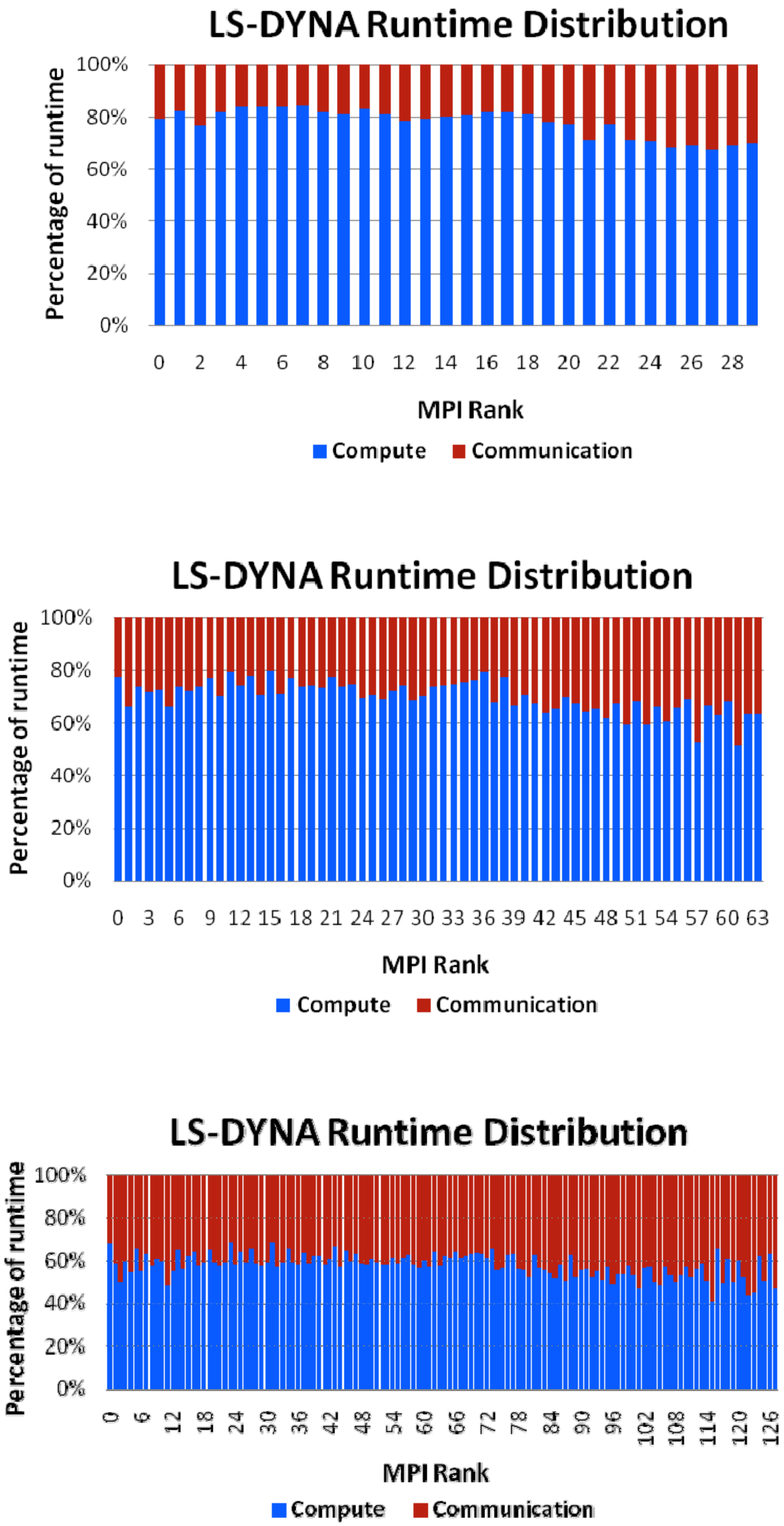

Figure 2 – Run-time distribution between computation and communications

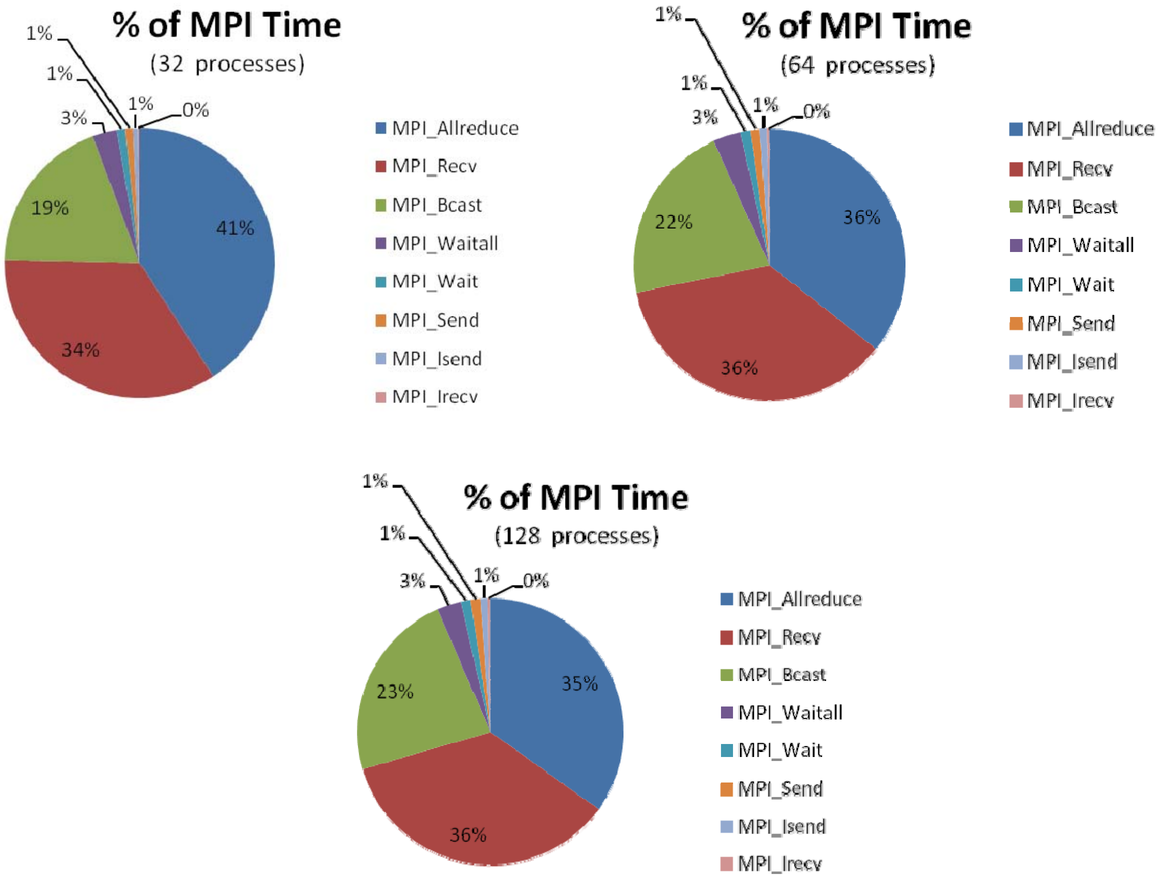

Figure 3 – Distribution of the different MPI communications

As seen in figure 2, the percentage of communication time versus computation time increases as cluster size scales from 35% at 32 processes (4-nodes) to 55% at 128 processes (16-nodes) which demonstrated the need for a low-latency interconnect solution that delivers the increased throughput required for the node or core communications.

From figure 3 it is clear that the two MPI collectives, MPI_Allreduce and MPI_Bcast consume most of the total MPI time and hence is critical to LS-DYNA performance. MPI libraries and offloading related to those two collectives operation will greatly influence the system performance. In this paper we will review the difference between the different MPI libraries. We will review the MPI collectives offloading capabilities presented within the Mellanox ConnectX-2 adapters in a follow-up paper.

MPI Library Comparisons

We have compared the performance of three MPI libraries – Open MPI, HP MPI and Platform MPI and two compilers – Intel compiler and PGI compiler. The results are presented in figure 4.

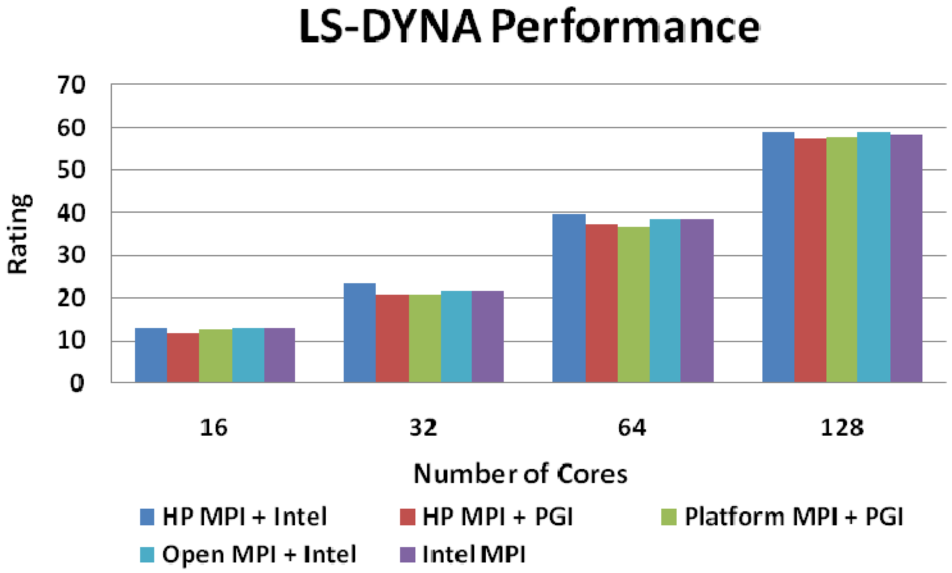

Figure 4 – MPI libraries and compilers comparison

The combination of HP MPI with the Intel compiler has demonstrated the highest performance but in general the difference between the different configurations was not dramatic. All MPI options showed comparable performance in the four cluster sizes that were tested.

Parallel File System

There are three main categories of file systems for HPC system I/O - the Network File System (NFS), storage area network (SAN) file systems, and parallel file systems. The most scalable solution with the highest performance is the parallel file systems. In this solution, a few nodes connected to the storage (I/O nodes) serve data to the rest of the cluster. The main advantages a parallel file system can provide include a global name space, scalability, and the capability to distribute large files across multiple nodes.
Generally, a parallel file system includes a metadata server (MDS), which contains information about the data on the I/O nodes. Metadata is the information about a file (its name, location and owner). Some parallel file systems use a dedicated server for the MDS, while other parallel file systems distribute the functionality of the MDS across the I/O nodes.

Lustre is an open source parallel file system for Linux clusters. It appears to work like a traditional UNIX file system (similar to GPFS), distributed Object Storage Targets are responsible for actual file-to-disk transactions and a user level library is available to allow application I/O requests to be translated into LUSTRE calls. As with other parallel file systems, data striping from concurrently running nodes is the main performance enhancing factor. Metadata is provided from a separate server. Lustre stores file system metadata on a cluster of MDSs and stores file data as objects on object storage targets (OSTs), which directly interface with object-based disks (OBDs). The MDSs maintain a transactional record of high-level file and file system changes. They support all file system namespace operations such as file lookups, file creation, and file and directory attribute manipulation. In our test environment, native InfiniBand-based storage was used for Lustre as described in figure 5.

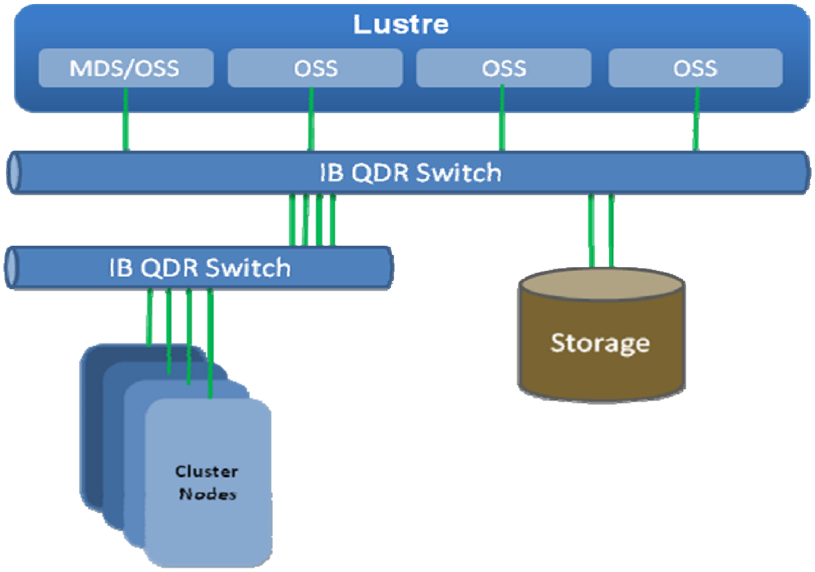

Figure 5 – Test system environment with native InfiniBand Lustre parallel file system

With the availability of native InfiniBand Lustre parallel file system, we have compared LS-DYNA performance with Lustre versus using the local disk for storing the data files. The results of the two cases are presented in figure 6.

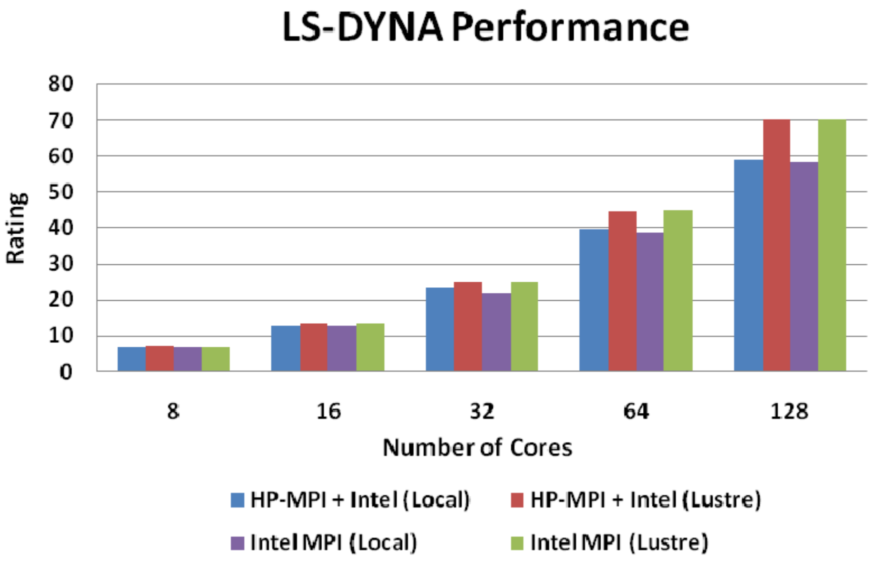

Figure 6 – LS-DYNA performance with Lustre versus local disk

Due to the parallel file system capabilities and the usage of InfiniBand as the network for Lustre, using Lustre instead of the local disk increased LS-DYNA performance 20% in average for both HP MPI and Intel MPI. Intel MPI has native Lustre support (command line mpiexec -genv I_MPI_ADJUST_BCAST 5 -genv I_MPI_EXTRA_FILESYSTEM on -genv I_MPI_EXTRA_FILESYSTEM_LIST luster).

Future work

Future investigation is required on the benefits of MPI collectives communication offloads supported in the Mellanox ConnectX-2 adapters. From figure 3 it is clear that the MPI collectives MPI_Allreduce and MPI_Bcast consume most of the total MPI time and offloading them to the network is expected to increase LS-DYNA performance.

Conclusions

From concept to engineering and from design to test and manufacturing; engineering relies on powerful virtual development solutions. Finite Element Analysis (FEA) and Computational Fluid Dynamics (CFD) are used in an effort to secure quality and speed up the development process. Cluster solutions maximize the total value of ownership for FEA and CFD environments and extend innovation in virtual product development.

HPC cluster environments impose high demands for cluster connectivity throughput, low-latency, low CPU overhead, network flexibility and high-efficiency in order to maintain a balanced system and to achieve high application performance and scaling. Low-performance interconnect solutions, or lack of interconnect hardware capabilities will result in degraded system and application performance.

Livermore Software Technology Corporation (LSTC) LS-DYNA software was investigated. In all InfiniBand-based cases, LS-DYNA demonstrated high parallelism and scalability, which enabled it to take full advantage of multi-core HPC clusters. Moreover, according to the results, a lower-speed interconnect, such as Ethernet is ineffective on mid to large cluster size, and can cause a dramatically reduction in performance beyond 8 server nodes (i.e. the application run time actually gets slower).

We have profiled the communication over the network of LS-DYNA software to determine LS-DYNA sensitivity points, which is essential in order to estimate the influence of the various cluster components, both hardware and software. We evidenced the importance of providing high-performance MPI_AllReduce and MPI_Bcast collectives operations and the performance differences between the different MPI libraries and compilers.

We have also investigated the benefits of using parallel file systems instead of the local disk and have shown that using InfiniBand for both the MPI and the storage can provide performance benefits as well as cost savings by using a single network for all cluster purposes – network consolidations.