MPI Optimizations via MXM and FCA for Maximum Performance on LS-DYNA®

Gilad Shainer[1], Tong Liu[1], Pak Lui[1], Todd Wilde[1]
[1]*Mellanox Technologies*

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 hardware configuration for solving the new emerging bottlenecks, in order to maintain high scalability and efficiency. In this paper we cover a new co-design architecture with hardware based accelerations and offloads for MPI collectives communications and how it affects LS-DYNA performance.

Introduction

High-performance computing (HPC) is a critical 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 times 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 large 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.

The recent trends in cluster environments, such as multi-core CPUs, GPUs and new interconnect speeds and offloading capabilities are changing the dynamics of cluster-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.

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 crash 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.

LS-DYNA relies on the Message Passing Interface (MPI) for cluster or node-to-node communications, which is the de-facto messaging library for high performance clusters. Collective communications are the point to multipoint messaging operations frequently used by MPI for operations like broadcasts for sending around initial input data, reductions for consolidating data from multiple sources and barriers for global synchronization. Any collective communication executes some global communication operation by coupling all processes in a given group. As such, collective communications have a crucial impact on the application's scalability. In addition, the explicit and implicit communication coupling, used in high-performance implementations of collective algorithms, tends to magnify the effects of system-noise on application performance further hampering application scalability.

Mellanox 40Gb/s QDR InfiniBand ConnectX-2 adapters address HPC performance and scalability for MPI parallel communications using a unique co-design architecture to extend the capabilities of the network into the communication libraries. This solution is comprised of Mellanox Messaging Accelerations, or MXM to accelerate the underlying send/receive messages, and Fabric Collective Accelerators, or FCA, which accelerate the collective operations within the parallel program.

In this paper we review the most often used collective operations that are found in LS-DYNA, namely MPI AllReduce and MPI Broadcast, and explore how their performance effects LS-DYNA simulations. We will also investigate how the co-design architecture, namely Fabric Collective Accelerations improve the overall performance of LS-DYNA simulations.

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 a 16-node compute cluster, where each server has dual socket/six-core Intel X5670 @ 2.93 GHz CPUs and 24GB memory. The operating system is CentOS-5.4 with InfiniBand drivers OFED 1.5.2. Mellanox ConnectX-2 InfiniBand QDR adapters and InfiniBand QDR switches are used for the cluster interconnect. The MPI version is Open MPI 1.4.3, LS-DYNA LS-DYNA mpp971_s_R5.0 with the 2001 Ford Taurus, detailed model (1,057,113 elements) benchmark.

The Co-Design Architecture Concept

Until recently, the majority of software and hardware developments have been on separate and disparate paths. The software development for interconnect technology and communication interfaces has concentrated on the area of algorithm enhancements while ignoring the underlying networking hardware, while the hardware development has concentrated on solving networking bottlenecks without taking into consideration the software communication libraries (Fig 1). This has led to the creation of performance bottlenecks, as well the inability for the applications to take full advantage of software algorithms running efficiently over new advances in the networking hardware.

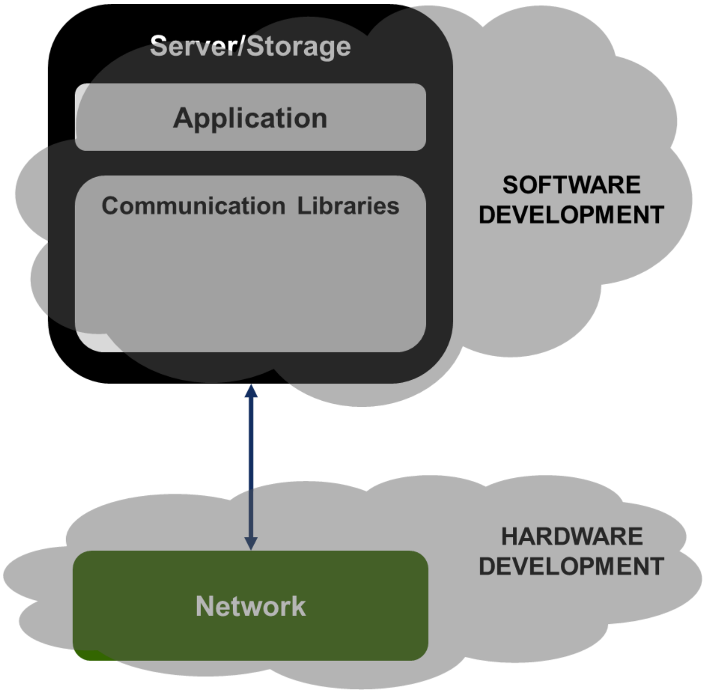


Figure 1 – Disparate development between application programmers and hardware engineers

In order to overcome the artificially created scalability and performance bottlenecks, Mellanox have put in place a new architecture for hardware-software development that enables the extension of the underlying hardware capabilities into the software communication libraries, and a creating of a new application programming interface (API) within the communication libraries to separate the actual communication algorithm and the interconnect implementation, and to add support for specific hardware-based offloading within the communication libraries.

The results of this co-design approach (Fig 2) led to a suite of solutions called ScalableHPC that have been adapted by many parallel programming libraries, including MPI libraries. The two main co-design solutions include Fabric Collective Accelerations, or FCA, which accelerates the underlying the collectives within the communication library, and MXM, which accelerate the underlying message passing within the library.

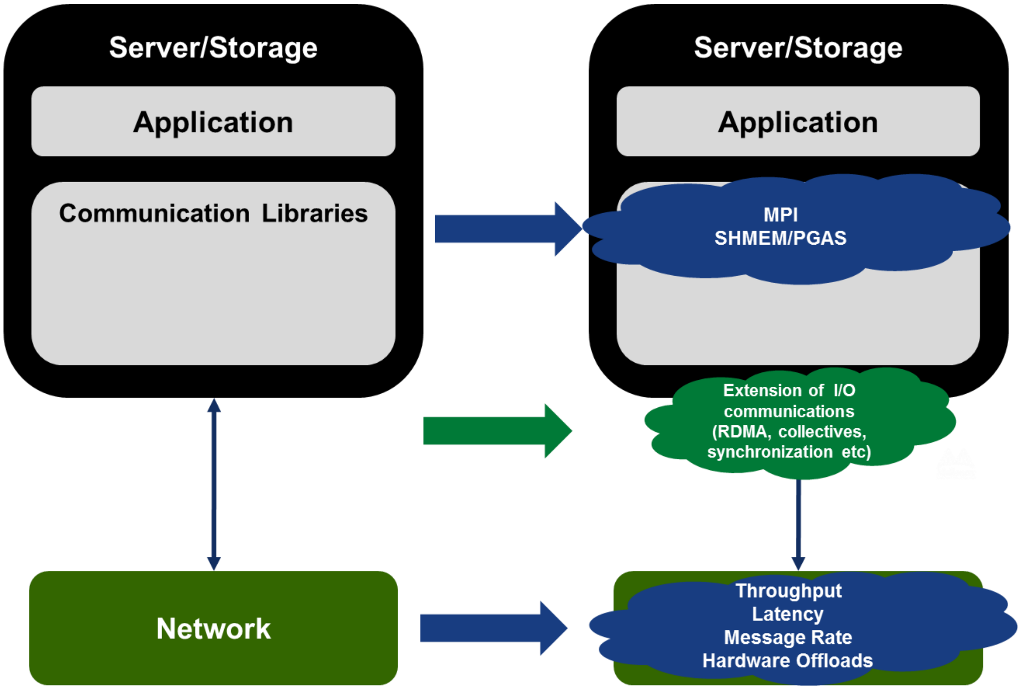


Figure 2 – The Co-Design Architecture

With these two packages incorporated into the parallel programming libraries, the underlying networks unique capabilities and accelerations are utilized properly and to the fullest potential.

Co-Design: Fabric Collective Accelerations

A key technology developed through the co-design initiative is Fabric Collective Accelerations, or FCA. Collective communications are used by many HPC applications and involve all of the processes running in a particular parallel programming job. One example would be a synchronization operation where the processes must wait at a particular point in the program until all processes involved in the program reach this point. Another example is a collective computation, such as a reduction, in which one memory of the group collects data from all other processes and performance a mathematical operation on the data.

To optimize the collective operations running over the high-performance interconnect equipment, we have applied co-design principles to offer a collect acieration library to assure that the hardware and underlying network architecture advancements were being used to the fullest and the collection operation was run efficiently as possible.

Several optimization have been applied within FCA to accurate the collective operation. First, the ability to map the underlying topology of the network and map this to the collective algorithm was implemented. This allowed the algorithm to apply this knowledge to understand the physical location of the various ranks within the collective and to then provide a tree structure to the collective operations. With this tree structure, message coalescing could be utilized at various levels within the topology, dramatically reducing the number of messages required, and also alleviate the potential congestion that would normally occur when running the collective.

Another optimization was to take advantage of the power multicast capabilities supported by the network, and to use these for broadcasts required by the collective. For example, in an Allreduce operation, once the parent of the topology aware collective tree has the final results, it must provide those to all of the ranks in the collective. Instead of sending the message to every rank individually, FCA allows it to send only one message out, and rely on the switch hardware to replicate the results to all of the other processes.

Finally, the hardware based collective offloads (Core-Direct) that run in the HCA hardware has been added to the FCA algorithm. Core-Direct technology offloads the collective operation to the HCA instead of relying on the CPU to compete it. This mitigates the effect of system noise, or “jitter’ than can slow the collective operation, in particular at large scale.

Co-Design: Mellanox Messaging Acceleration

The second element in the co-design architecture is Mellanox Messaging Acceleration, or MXM. While FCA handled the collective operation, MXM the extension for the underlying send/receive messages, including network transport, memory management, matching operations, and more. In particular for the support of the InfiniBand network, MXM extends the management of the network I/O channels into the communication libraries and the ability to optimize the implementation of the memory allocations and usage. MXM also can handle one-sides and two-sided communications. Figure 3 shows the architecture of FCA and MXM within the software stack.

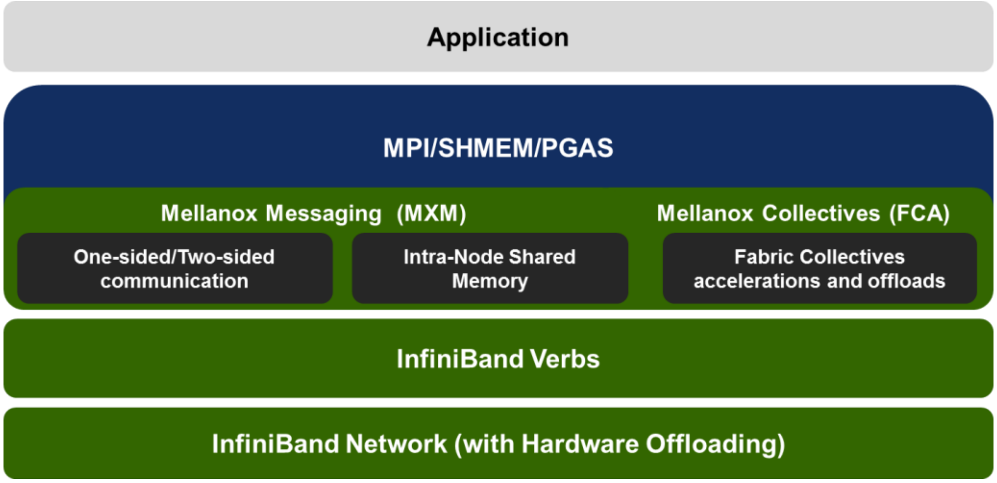
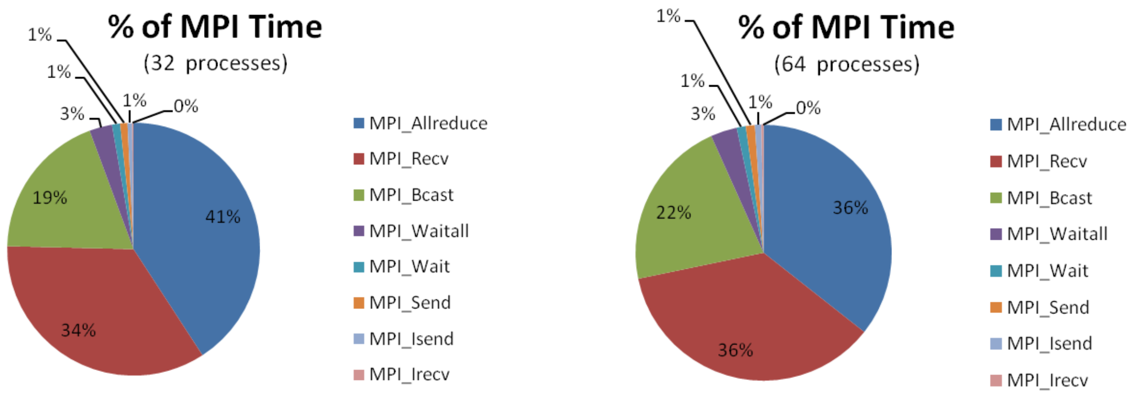

Application
MPI/SHMEM/PGAS
Mellanox Messaging (MXM)
Mellanox Collectives (FCA)
One-sided/Two-sided communication
Intra-Node Shared Memory
Fabric Collectives accelerations and offloads
InfiniBand Verbs
InfiniBand Network (with Hardware Offloading)

Figure 3 – Software layers architecture of the co-design implementation

LS-DYNA MPI Profiling

Profiling the application is essential for understanding its performance dependency on 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 MPI profiling data is presented in Figure 4, which shows the usage of the different MPI communications in several cluster configurations (32-cores, 64-cores 128-cores).



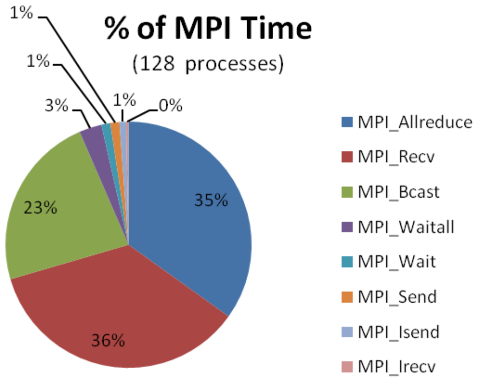


Figure 4 – Distribution of the different MPI communications

From Figure 4 it is clear that the two MPI collectives, MPI_Allreduce and MPI_Bcast (Broadcast) 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.

LS-DYNA Performance Results with FCA

As previously mentioned, FCA provides for accelerated collective operations that run efficienly and at scale with an InfiniBand fabric. FCA also utilizes Core-Direct hardware offload which in particular can provide advantages for systems at large scale by minimizing the effects of system noise and jitter. We have tested a 16-node cluster in order to identify the system size in which FCA will start showing a performance and productivity advantage.

Figure 5 shows the performance results of the Ford Taurus benchmark with and without the Fabric Collectives Acceleration and CORE-Direct offloading hardware. In up to 4 node configuration there are no significant advantages with FCA, but at 8 nodes configurations the performance increase is 5% and at 16 nodes, 192 cores, the performance increase is 15%.

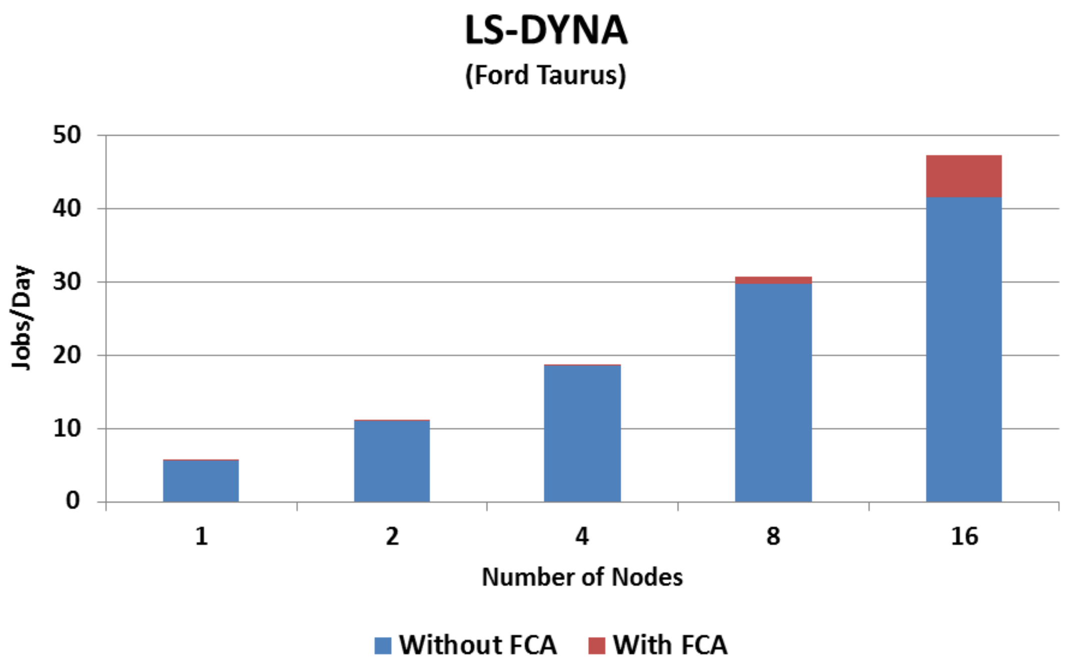


Figure 5 – LS-DYNA performance results with and with FCA

The performance advantage with FCA increases with cluster size, and it is expected to see over 20% at 32 nodes, etc. From the results we can conclude that any system size above 128 cores will benefit from FCA, where the benefit will increase as more cores will be used.

Figure 6 shows the time spent in the MPI library for the MPI All Reduce and MPI Broadcast operations in a 192 core configuration (16 nodes in our setup) with and without FCA. One of the side benefits of FCA is the accelerations of the collective communications which provides performance increases for LS-DYNA even at low scale. According to the results in figure 6, FCA reduces the MPI All Reduce and MPI Broadcast time by more than 10% each. We do expect higher acceleration and reduction in MPI time at a larger core-count.

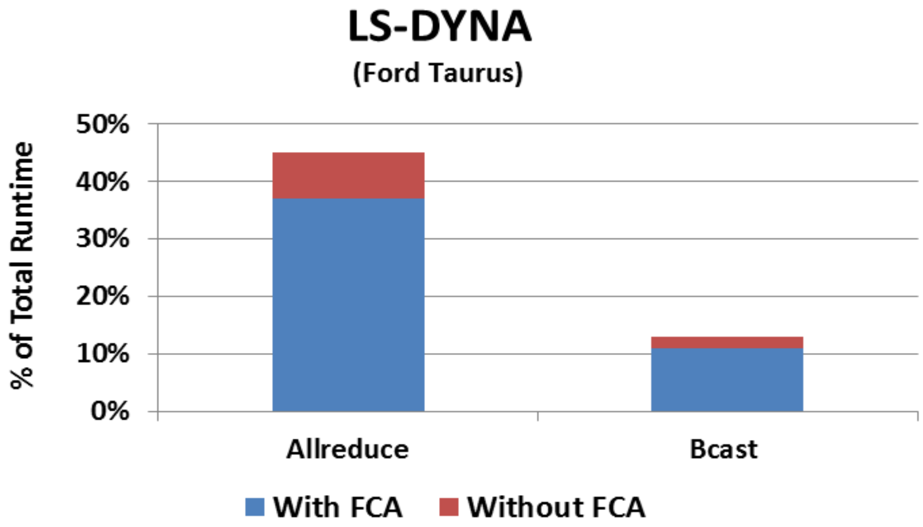


Figure 6 – LS-DYNA MPI time for Allreduce and Broadcast collectives operations

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.

The new concept of the Co-Design architecture, and the collective and message accelerations within parallel programming libraries provided by Mellanox provides performance and scalability increases for MPI applications.

In order to examine the benefits of FCA for commercial HPC, the Livermore Software Technology Corporation (LSTC) LS-DYNA software was investigated. We showed that FCA with CORE-Direct becomes essential for higher productivity beyond 128 processes, and showed a 15% performance increase at 192 processes. We expect to see larger impact on performance as the cluster and job sizes increase. In addition, since FCA with CORE-Direct provides the capability for MPI asynchronous collectives (MPI-3) we expect to see higher gain with the next generation MPI solutions that will be based on the MPI-3 specifications.