The MPI Scalability Challenge

Collective functions in the MPI API involve communication between all processes in a process group (which can mean the entire process pool or a program-defined subset). These types of calls are often useful at the beginning or end of a large distributed calculation, where each processor operates on a part of the data and then combines it into a result. Examples of popular collective functions are MPI_Barrier, MPI_Broadcast, MPI_Reduce and MPI_Allreduce.

The performance of collective communication operations is known to have a significant impact on the scalability of some applications. Indeed, the global, synchronous nature of some collective operations directly implies that they will become the bottleneck when scaling to thousands of ranks (where a rank is an MPI process, typically running on a single core). This fact has led many researchers to try to improve the efficiency of collective operations.

Several challenges face today’s collective function communication for applications scaling beyond a few dozens of cores. The impact of these challenges significantly increases as multi-core environments continue to dominate HPC, and the number of cores per node continues to grow.

- In large fabrics, servers are typically connected through several tiers of switches, yet all traffic passes through all tiers because the switches are not capable of handling offloading or aggregation. More tiers and longer routes add latency and jitter that rapidly accumulate and significantly impact runtime.
- In many cases a single rank receives data from many ranks at once, significantly increasing the risk of congestion.
- Use of unicast instead of multicast significantly increases latency and congestion when distributing results from a single rank to thousands of ranks.
- The actual physical connectivity topology of the fabric and process placement is not taken into account in processing the collective function.
- Server OS “noise” quickly accumulates on large, synchronous operations, severely impacting overall latency as well as predictability (jitter).

Features

- Offload collective function communication & computation from Platform MPI process into Voltaire switches
- Efficient collective communication flow optimized to job and topology
- Monitor performance of collective operations

Benefits

- Significantly reduce Platform MPI job runtime (up to 40%)
- Improve collective function scalability above and beyond any proprietary interconnect
- Eliminate congestion caused by collective function calls
- No additional hardware to install or manage
- No space/power/cooling penalty
- Zero provisioning penalty (parallel to job scheduler initialization)

Supported MPI Collectives

- Allreduce, Reduce, Barrier, BCast
- Open MPI 1.4.1 and up
- Platform MPI 8.0 and up
Breakthrough Performance with Fabric-based Collective Offload

With the aim of scaling out fabrics and improving performance from an application perspective, not just a server/network perspective, Voltaire has designed a unique patent-pending technology called Fabric Collective Accelerator™ (FCA™) software. FCA significantly reduces the runtime of collective operations on any fabric, and is available as an add-on to Voltaire’s Unified Fabric Manager™ (UFM™) software. FCA addresses a wide range of applications with out-of-the-box integration with Platform MPI, and requires no changes to the application.

The FCA algorithm uses data from the job scheduler and UFM to establish a topology map as it relates to a specific job, as well as the processing power of Voltaire’s switches to offload significant parts of the computation.

Using the FCA algorithm with Voltaire switches and UFM ensures a single message per physical wire for any collective function, as opposed to potentially hundreds or thousands of messages per wire using traditional algorithms for collective function handling. This non-blocking collective architecture finally allows InfiniBand to scale collective communication to thousands of nodes better than any interconnect (standard or proprietary) in the market.

As a result, this well integrated, application-oriented, fabric-wide solution can reduce the runtime of collective operations by more than 90%, resulting in an up to 40% reduction in total MPI job runtime.

Reaching such significant improvements requires new intelligence in several parts of the fabric.

- First, full information regarding topology and job parameters needs be obtained, processed and translated into a collective map that can then be programmed to all compute entities in the fabric. This is handled by Voltaire UFM.
- Second, each compute entity, be it a server or a CPU on a switch, needs to be able to perform an efficient collective computation on the ranks below it in the tree, and forward the result to the next level. Platform MPI ensures first tier local offload for all cores in a server, while Voltaire switches offload computation at the switching tiers of the fabric.
- Finally, the computation result is distributed to all ranks using InfiniBand (hardware-based) multicast coupled with a reliable message delivery protocol.

Initialization takes no more than 2-3 seconds even on clusters with tens of thousands of ranks, and occurs in parallel with the job scheduler provisioning/resource allocation phase—so there is no penalty on the job start/runtime.
Accelerating Platform MPI Performance

Platform MPI 8.0 is a high performance, production-quality implementation of the Message Passing Interface (MPI). It is widely used in the high performance computing industry and is considered the de facto standard for developing scalable, parallel applications.

Platform MPI 8.0 maintains full backward compatibility with HP-MPI and Platform MPI 7.0 applications and incorporates advanced CPU affinity features, dynamic selection of interface libraries, superior workload manager integrations and better performance and scalability. Platform MPI supports the broadest range of industry standard platforms, interconnects and operating systems helping ensure that parallel applications can run anywhere.

Voltaire FCA integrates with Platform MPI to enable unlimited scalability and consistently improved performance that is determined by the number of switch hops, which remains nearly constant, rather than the number of ranks. These results are achieved without making any changes to the application code.

Benchmark Comparisons by HP

HP is a leader in high performance computing, delivering fully integrated systems tuned for the best performance. HP conducted benchmark tests using HP ProLiant BL460c G7 Blade Servers (each with two Intel® Xeon® CPU X5670 @ 2.93 GHz) in order to measure the impact of FCA and Platform MPI on FLUENT V12.1 and STAR-CD V4.12. Platform MPI instrumentation was used to measure user and MPI time usage, with FCA enabled vs. disabled. FCA reduced MPI time and increased user time, increasing both application performance and cluster scalability.

- 80 x HP ProLiant BL460c G7 Blade Servers each with two Intel® Xeon® CPU X5670 @ 2.93 GHz
- Voltaire QDR InfiniBand
- Platform MPI 8.0
- Fluent version 12.1
- StarCD version 4.12
Maximize Platform MPI Performance with Voltaire® Fabric Collective Accelerator™ (FCA™) and HP

ANSYS Fluent benchmarks on 192 cores show 14-30% gain in runtime acceleration

Star-CD A-Class benchmark on 192 cores shows 10% runtime reduction