HPC Advisory Council - HPCIScale Special Interest Group

Accelerating High Performance Computing Applications Through MPI Offloading

HPC ADVISORY COUNCIL
NETWORK OF EXPERTISE

WHITE PAPER

Gilad Shainer, Tong Liu, Pak Lui, Richard Graham
{HPC Advisory Council}

Abstract

In the past, performance tuning of parallel applications could be fairly accomplished by separately optimizing their algorithms, communication, and computational aspects. However, as we continue to scale future larger machines, these issues become co-mingled and must be addressed comprehensively. MPI collectives communications are frequently being used for processes synchronization and explicit or implicit coupling between processes and their performance is critical for scalable, high-performance applications. Optimizing collectives communication performance can be achieved by offloading these communication to the network therefore minimizing the negative effect of system noise and jitter as well as separating them from the rest of the CPU activities. Throughout application profiling we can identify applications which will greatly benefit from such offloading and can determine the associate performance and productivity benefits.

Introduction

For many years optimizing high-performance computing applications could be done simply by separately optimizing their algorithms, communication, and computational aspects. As we move into the many-core and many-node compute environments, these issues must be addressed comprehensively. According to the June 2011 TOP500 supercomputers list, we have ushered into the PetaScale era and all top 10 systems have demonstrated above Petaflop performance. Multiple systems node count has exceeded ten thousand of nodes, and the number of cores is in the tens (or hundreds in cases of GPGPUs). The Message Passing Interface library (MPI) or the Shared Memory (SHMEM) environments are a few examples of libraries that provide implementations of collectives communications for the usage of HPC applications. Collectives communications have a crucial impact on the engineering and scientific application's performance and scalability as they are frequently being used for operations such as broadcast for sending around initial input data, reductions for consolidating data from multiple sources and barriers for global synchronization. This behavior tends to have the most significant negative 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.

A recent development, which is the result of a collaboration between HPC researchers at Oak Ridge National Laboratory and the InfiniBand vendor, Mellanox, addresses several aspects of the MPI collective communication performance and scalability by offloading the MPI collective communications from the host CPU to the network. This solution provides the mechanism needed to support not only computation and communication overlap (allowing the communication to progress asynchronously in hardware as being specified by the MPI Forum for MPI-3), but also supports simultaneous computations processed by the CPU for higher application performance. This reduces the negative effects of systems noise and jitter and the effect of the non-application compute CPU activities. Offloading of the MPI communication semantics from the software MPI to network provides a comprehensive solution for emerging scalability and performance challenges, as well as enables the usage of "smart" clustering elements beyond the CPU for next-generation productive HPC.

MPI Collectives Offloads

The recent InfiniBand interconnect solutions include new hardware technology to support offloading communication management (Figure 1). The new technology defines a general purpose mechanism for coordinating multiple network operations. In the design process, care was taken to ensure this supports effective implementation of asynchronous collective communications (MPI, SHMEM and others) used by scientific applications. The goal of these enhancements is to relieve communication management workload from the CPU and to enhance the scalability of applications on ultra-scale computer systems.

The new offloading technology, named CORE-Direct®, includes Management-Queue, Multiple Work Request, and the wait task functionality. It is designed to support arbitrary communication patterns and to manage the data dependencies between tasks in these patterns. This was added specifically to support collective operations. The intent is to offload the progression of collective operations to the network, with the CPU being involved only in the completion of the collective communication. MPI collective operations are implemented using an interdependent sequence of network operations executed by each process.

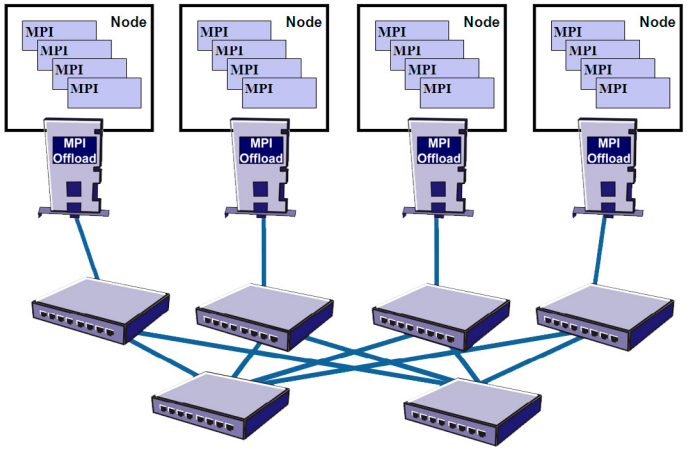

Figure 1 – HPC system architecture (cluster) with MPI offloading

Applications Communications Profiling

In order to determine which applications can benefit from MPI collectives offloads, we need to review the communications patterns of each application, or application groups with similar network characteristics. MPI profiling can be done via dedicated tools either open source or commercial. For example: mpiP for MVAPICH MPI and IPM for Platform MPI. Throughout the MPI profiling we could determine how much of MPI processed communications is being done via MPI collectives communications and what is the associated overhead. The higher the usage of the MPI collective semantics, the higher the benefit from using MPI offloading. The current MPI collectives offloads mechanism support offloading of MPI Barrier, MPI Broadcast, MPI AllReduce, MPI Reduce, MPI AllGather and MPI AllgatherV collectives communications. The rest of the communications are being handled via the CPU.

OpenFOAM Computational Fluids Dynamics (CFD) Application

From concept to engineering, and from design to test and manufacturing, engineering relies on powerful virtual development solutions. CFD is performed in an effort to secure quality and speed up the development process. The OpenFOAM (Open Field Operation and Manipulation) CFD Toolbox is a free, open source CFD software package produced by a commercial company, OpenCFD Ltd. It has a large user base across most areas of engineering and science, from both commercial and academic organizations. OpenFOAM has an extensive range of features to solve anything from complex fluid flows involving chemical reactions, turbulence and heat transfer, to solid dynamics and electromagnetics.

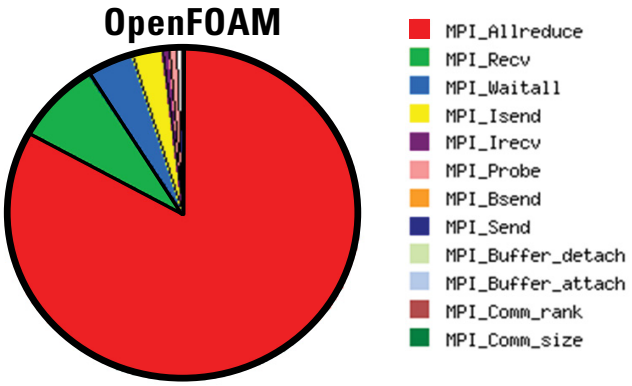

Figure 2 – OpenFOAM MPI Profiling Information

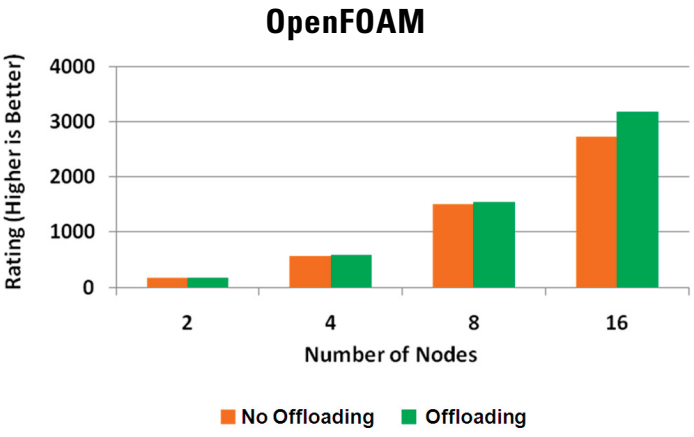

Figure 3 – OpenFOAM Performance with and without MPI offloads

OpenFOAM communication profiling is presented in Figure 2 for a 16-node cluster, 192-coreconfiguration. The main collective communication used is MPI Allreduce, which is responsible for 80% of the MPI communications. The performance advantage of the MPI collectives offloads by using the cluster network to offload the MPI collectives communications from the CPU provides more than 20% performance increase, as presented in Figure 3.

Amber Molecular Dynamic Application

Amber refers to two things: a set of molecular mechanical force fields for the simulation of biomolecules (which are in the public domain, and are used in a variety of simulation programs) and a package of molecular simulation programs which includes source code and demos. The current version of the code is Amber version 11, which is distributed by UCSF. Amber is one of the most widely used programs for biomolecular studies, with an extensive user base. It is being used for classical molecular dynamics simulations (NVT, NPT, etc), force field for biomolecular simulations, combined Quantum Mechanics/Molecular Mechanics (QM/MM) implementation and more.

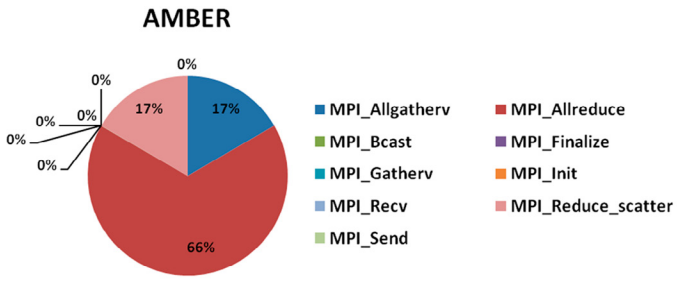

Figure 4 – Amber MPI Profiling Information

Amber communication profiling is presented in Figure 4 for a 16-node cluster, 192-core configuration. The main collective communications used are MPI Allreduce and MPI Allgatherv, which are responsible for 83% of the MPI communications. The performance advantage of the MPI collectives offloads by using the cluster network to offload the MPI collectives communications from the CPU provides more than 30% performance increase as presented in Figure 5.

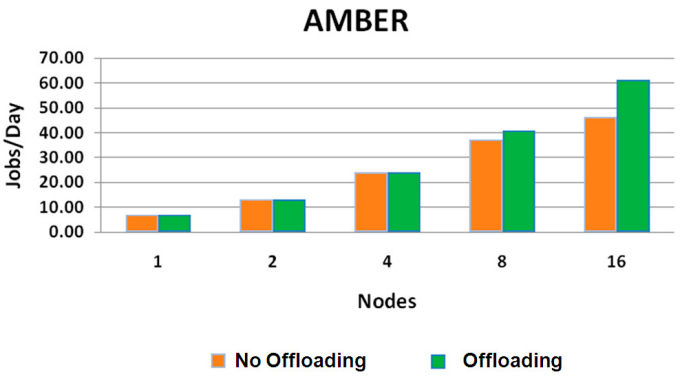

Figure 5 – Amber Performance with and without MPI offloads

CPMD Car-Parrinello Electronic Structure and Molecular Dynamic Application

Car-Parrinello Molecular Dynamics (CPMD) is an ab initio electronic structure and molecular dynamics (MD) simulation software that provides a powerful way to perform molecular dynamic simulations from first principles, using a plane wave/pseudopotential implementation of density functional theory. The CPMD code has been used to examine systems including protein active sites, liquid-surface interactions, and surface catalysts. The ability to examine interactions on the nanoscale makes this approach ideal for studying systems where chemical and biological interactions are critical.

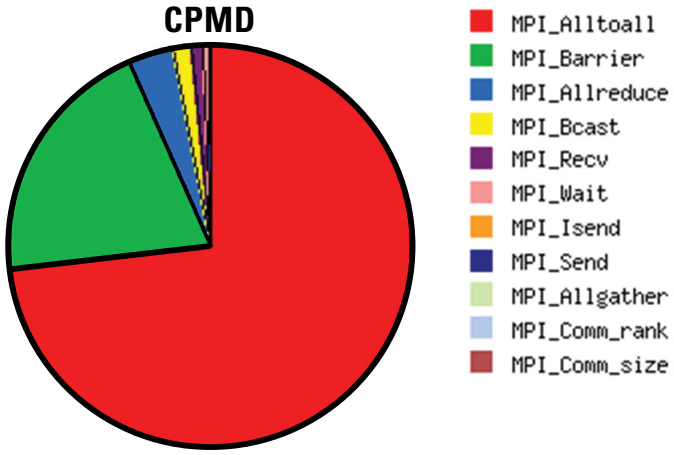

Figure 6 – CPMD MPI Profiling Information

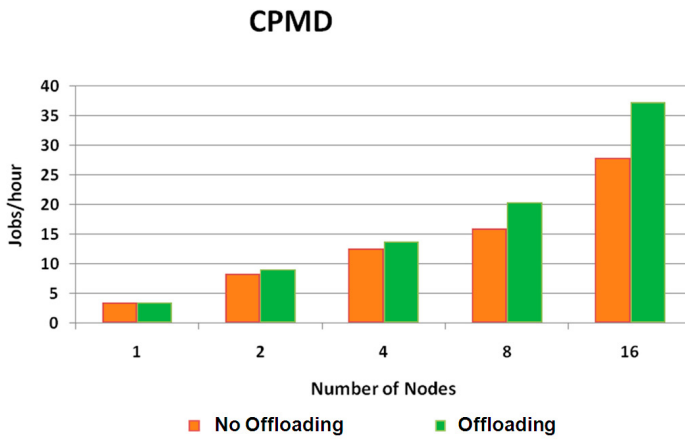

Figure 7 – CPMD Performance with and without MPI offloads

CPMD communication profiling is presented in Figure 6 for a 16-node cluster, 192-core configuration. The main collective communications used are MPI Alltoall, MPI Allreduce and MPI Barrier, which are responsible for 90% of the MPI communications. The performance advantage of the MPI collectives offloads by using the cluster network to offload the MPI collectives communications from the CPU provides more than 35% performance increase as presented in Figure 7.

Summary

In this paper we explored the advantages of offloading MPI collectives communications from the CPU to the cluster interconnect on various applications' performance. Offloading the MPI collectives enables faster execution of the collectives communications, higher overlapping of computations and communications, and reduces the load from the CPU. The later hides two benefits – first it enables more CPU cycles to be dedicated to the application and second it minimizes the negative effect on the critical processes communications – the collectives operations.

We explored three open source applications by profiling their communications to examine their usage of the collective operations, and then reviewed the performance benefits of using collective offloads. As expected, as more collectives communication were in use, the higher the performance gain we saw. Our testing were limited to a small size of a cluster – 16 nodes, or 192 cores, therefore we expect to see higher performance benefits at larger cluster sizes.

Acknowledgment

We would like to thank the HPC Advisory Council for providing access time to the council compute center for conducting the described tests.

Authors

Gilad Shainer, Tong Liu, Pak Lui, Richard Graham

{HPC Advisory Council}

HPC ADVISORY COUNCIL
NETWORK OF EXPERTISE

350 Oakmead Pkwy, Sunnyvale, CA 94085
Tel: 408-970-3400 • Fax: 408-970-3403
www.hpcadvisorycouncil.com