Presented at Workshop on Parallel Programming on Accelerator Clusters (PPAC), held in conjunction with IEEE Cluster, September 2011

MPI Alltoall Personalized Exchange on GPGPU Clusters: Design Alternatives and Benefits

A. K. Singh, S. Potluri, H. Wang, K. Kandalla, S. Sur, D. K. Panda

Department of Computer Science and Engineering

The Ohio State University

{singhas, potluri, wangh, kandalla, ssur, panda}@cse.ohio-state.edu

Abstract

General Purpose Graphics Processing Units (GPGPUs) are rapidly becoming an integral part of highperformance system architectures. The Tianhe-1A and Tsubame systems have received significant attention for their architectures that leverage GPGPUs. Increasingly many scientific applications that were originally written for CPUs using MPI for parallelism are being ported to these hybrid CPU-GPU clusters. In the traditional sense, CPUs perform computation while the MPI library takes care of communication. When computation is performed on GPGPUs, the data has to be moved from device memory to main memory before it can be used in communication. Though GPGPUs provide huge compute potential, the data movement to and from GPGPUs is both a performance and a productivity bottleneck. Recently, the MVAPICH2 MPI library has been modified to directly support point-to-point MPI communication from the GPU memory. Using this support, programmers do not need to explicitly move data to main memory before using MPI. This feature also enables performance improvement due to tight integration of GPU data movement and MPI internal protocols. Typically, sceintific applications spend a considerable portion of their execution time in collective communication. Hence, optimizing peformance of collectives has a significant impact on their performance. MPI Alltoall is a heavily used collective that has O(N2) communication, for N processes. In this paper, we outline the major design alternatives for MPI Alltoall collective communication operation on GPGPU clusters. We propose three design alternatives and provide a corresponding performance analysis. Using our dynamic staging techniques, the latency of MPI Alltoall on GPU clusters can be improved by 44% over a user level implementation and 31% over a send-recv based implementation for 256 KByte messages on 8 processes.