Improving Performance of OpenSHMEM Reference Library by portable PE Mapping Technique

Swaroop Pophale, University of Houston, Texas

This work is supported by the United States Department of Defense & used resources of the Extreme Scale Systems Center at Oak Ridge National Laboratory.

Abstract

OpenSHMEM is a Partitioned Global Address Space library that provides greater overlap between communication and computation by providing an API for explicit one-sided communication. Although the PGAS programming model provides more control over data placement, some communication intensive library operations like collectives would benefit from prior knowledge of the relative placement of the processing elements (PEs) participating in the program. Since the computation-communication ratio for different applications is different, we think that per application analysis of the communication pattern is essential in determining the placement of the PEs across a target architecture. From that profiling information and topology of the underlying hardware better mappings can be obtained and collectives that are aware of these mappings will, in turn, reduce network traffic thus reducing the cost of data movement. This poster presents key concepts involved in the communication cost modeling and efficient ways of representing, storing and using this information in the collective operations like *broadcast, barrier, reductions* collectives to reduce the overall execution time of the application.

Introduction

What is OpenSHMEM?

- OpenSHMEM Specification is an effort to create a standardized SHMEM library API by making the process open to the community for reviews and contributions.
- OpenSHMEM is a Partitioned Global Address Space (PGAS) library .
- A simple library API for C, C++ and Fortran programs that supports the Single Program Multiple Data (SPMD) style of programming .
- The OpenSHMEM API supplies routines for remote data transfer, remote atomic memory operations, with a simple set of ordering, locking, point to point synchronization and **collectives** (broadcast, reduction, group synchronization, collection).
- The processors participating in an application using the OpenSHMEM library are referred to as processing elements (PEs).
- PEs communicate with each other through one-sided updates to **symmetric** data elements.
- Symmetric data can be statically allocated, or dynamically allocated at run-time using a symmetric memory allocator provided by SHMEM.
- Symmetric data has the same size, type, and relative address on all PEs.
- Explicit synchronization required to guarantee completion of one-sided put operations.
- SGI's SHMEM API is the baseline for OpenSHMEM Specification 1.0
- The HPC Tools group at the University of Houston in collaboration with Oak Ridge are working on an OpenSHMEM Reference Implementation and Specification development.

OpenSHMEM Reference Library

- Our portable SHMEM library in accordance with OpenSHMEM specification 1.0.
- Uses GASNet communication library to interface with the underlying network interconnects.
- Current library implementation is layered over GASNet for efficient one-sided operations.
- Reference library implementation aims to provide multiple algorithms for collectives which are optimized for a range of different execution scenarios.

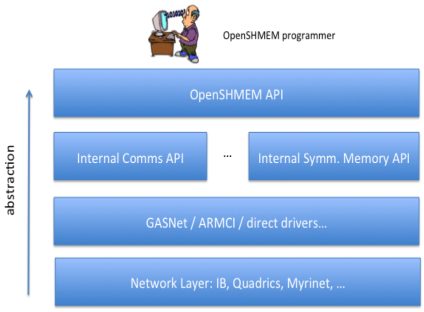
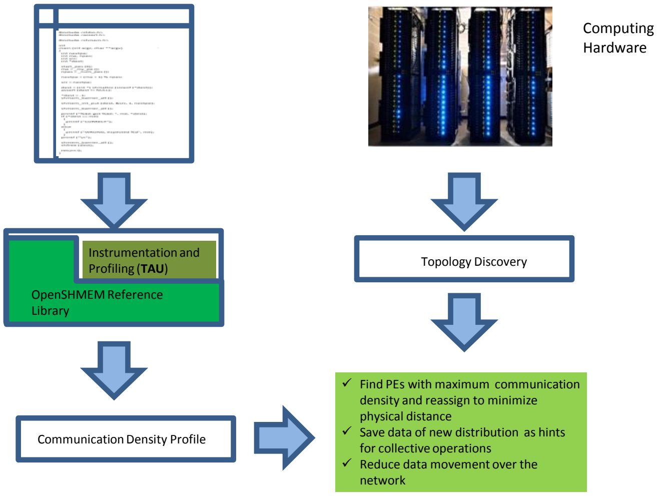

OpenSHMEM programmer
OpenSHMEM API
Internal Comms API
...
Internal Symm. Memory API
GASNet / ARMCI / direct drivers...
Network Layer: IB, Quadrics, Myrinet, ...
abstraction

Figure 1. OpenSHMEM Reference Library

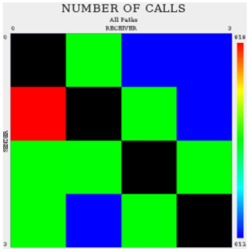
Key Concepts

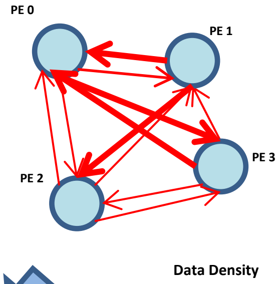

Computing Hardware
Instrumentation and Profiling (TAU)
OpenSHMEM Reference Library
Topology Discovery
Communication Density Profile

- ✓ Find PEs with maximum communication density and reassign to minimize physical distance
- ✓ Save data of new distribution as hints for collective operations
- ✓ Reduce data movement over the network

Case Study

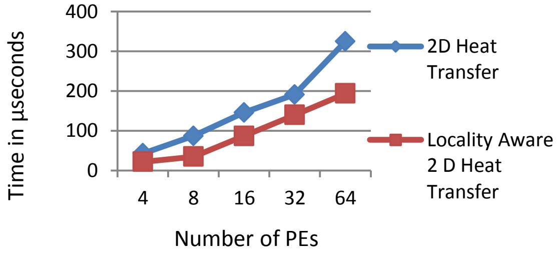
Application: 2D-Heat Transfer
Application for 2D heat transfer modeling using different methods. Adapted from the parallel MPI implementation of 2D heat conduction finite difference over a regular domain using the following methods, Jacobi, Gauss-Siedel and SOR.





ORIGINAL PE ASSIGNMENTS	NEW PE ASSIGNMENTS
PE 0	PE 0
PE 1	PE 3
PE 2	PE 2
PE 3	PE 1

- The new PE assignments are based on the frequency of communication as well as the size of the data communicated.
- We assume that the actual allocation of PE to a computing core is transparent to the user , but is assigned in a sequential fashion (which is usually the case)
- New PE assignments are recorded for collective operations to enable usage of locality information.


Time in µseconds
Number of PEs
2D Heat Transfer
Locality Aware 2 D Heat Transfer