Improving Performance of OpenSHMEM Reference Library by portable PE Mapping Technique

Swaroop Pophale
University of Houston
Texas, 77204
spophale@cs.uh.edu

Tony Curtis
University of Houston
Texas, 77204
tonyc@cs.uh.edu

Barbara Chapman
University of Houston
Texas, 77204
chapman@cs.uh.edu

ABSTRACT
Reducing data communication cost is a critical performance consideration and the need is more acute when using libraries like the OpenSHMEM Reference library which has to sacrifice some performance optimizations for portability. Being a Partitioned Global Address Space library the OpenSHMEM reference library provides more control over data placement, yet, some communication intensive applications would benefit from the libraries prior knowledge of its communication pattern. In this poster we discuss a low cost portable methodology to provide PE re-numbering to facilitate maximum on-node communication. We validate our method using the well-documented 2D heat transfer application.

Categories and Subject Descriptors
D.2.2 [**Design Tools and Techniques**]: Software libraries

Keywords
Communication optimizations, PGAS, OpenSHMEM

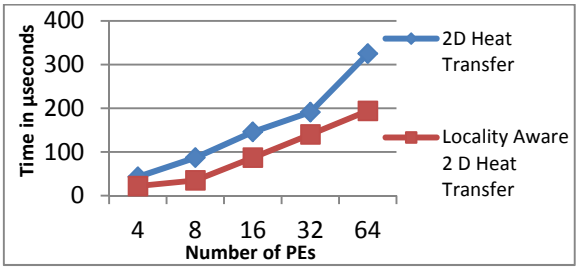
1. INTRODUCTION
OpenSHMEM is a Partitioned Global Address Space library that provides overlap between communication and computation by providing an API for explicit one-sided communication for parallel SPMD applications. Since the computation-communication ratio for different applications is different, a per application analysis of the communication pattern is essential to determine the placement of the PEs across a target architecture. From that profiling information of the application and the topology of the underlying hardware better processor mappings can be obtained. The OpenSHMEM reference library can use this information and provide a low cost process re-naming scheme that will, in turn reduce communication over the network.

2. BACKGROUND, MOTIVATION
Topology mapping is a popular field of research and there are many contributions out there with respect to the optimal process placement. Most of these studies involve specific hardware platforms [1], and/or are limited to the distributed message passing paradigm [2, 3]. For example, several vendor distributed MPI implementations use graph theory algorithms to formulate a near optimum placement of resources. But these solutions are implementation specific. To meet the portable needs of our OpenSHMEM library implementation we have devised our own methodology which can be extended to all other PGAS libraries and language extensions. Moreover since it relies on the communication pattern within the application and the relative communication volume between processes the preliminary run used to collect the required profiling information need not be a production run (since the pattern of communication does not change with change in the amount of data to be processed). As long as the ratio of the individual point-to-point communication is the same the mapping will be optimum for all future runs using the same number of processes.

ICS'13, June 10–14, 2013, Eugene, Oregon, USA.
ACM 978-1-4503-2130-3/13/06.

3. METHODOLOGY
The hardware information is collected prior to execution of the application using the portable **hwloc** [4] package. We use the portable TAU profiling tool [5] to collect the information pertaining to the OpenSHMEM communication events within the application. We compute the communication density matrix by considering the total volume of data exchanged between the PEs. We then place the results of our analysis in the working directory. When the OpenSHMEM library is initialized and the PE numbers are assigned based on the analysis discussed above. This enables optimum placement of PEs.

4. RESULTS
We first adapt the parallel MPI implementation of 2D heat conduction finite difference over a regular domain to use the OpenSHMEM API for computation and communication.



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

6. REFERENCES
1. H. Yu, I.-H. Chung, and J. Moreira. Topology mapping for Blue Gene/L supercomputer. In SC'06, page 116, New York, NY, USA, 2006. ACM.
2. J. L. Träff. Implementing the MPI process topology mechanism. In SC'02, pages 1–14, 2002.
3. T. Hoefler et al. The Scalable Process Topology Interface of MPI 2.2. CCPE, 23(4):293–310, Aug. 2010.
4. Broquedis, F., Clet-Ortega, J., Moreaud et al. hwloc: A Generic Framework for Managing Hardware Affinities in HPC Applications. PDP 2010. February 2010.
5. S. S. Shende and A. D. Malony. "The TAU parallel performance system," Int. J. High Perform. Comput. Appl., vol. 20, no. 2, pp. 287–311, May 2006.