SRC: OpenSHMEM Library Development

Swaroop Pophale
University of Houston
4800 Calhoun Road, Houston, Texas.
1-832-818-5558
spophale@cs.uh.edu

ABSTRACT
OpenSHMEM is a PGAS programming library implementing an RMA-based point-to-point and collective communication paradigm which decouples data motion from synchronization. This results in a more scalable programming model than more common two-sided paradigms such as MPI. The OpenSHMEM project arose in an effort to standardize among several implementations of the decade-old SHMEM API, which exhibited subtle differences in the API and underlying semantics, inhibiting portability between implementations. In collaboration with Oak Ridge National Laboratory, the University of Houston is preparing an API specification and a portable, scalable, observable OpenSHMEM reference implementation.

Categories and Subject Descriptors
D.3.2 [**Language Classifications**]: *Concurrent, distributed, and parallel languages*; D.3.4 [**Processors***]: Run-time environments*; D.2.4 [**Software/Program Verification**]: *Validation*

General Terms
Performance, Languages, Standardization, Verification

Keywords
ICS Poster, SHMEM, OpenSHMEM, PGAS

1. EXTENDED ABSTRACT
The Symmetric Hierarchical MEMory (SHMEM) API allows a programmer to write parallel applications using a Partitioned Global Address Space (PGAS) programming model. This model permits an explicit decoupling of data motion and synchronization to enhance application scalability. SHMEM is an SPMD programming model in which the parallel processes communicate through one-sided updates to the symmetric memory spaces of the participating processing elements (PEs) via the API's support for point-to-point (put/get) operations, remote atomic memory operations, broadcasts, and reductions with a simple set of ordering, locking, and synchronization primitives. The key concept in SHMEM is the use of data structures stored in symmetric memory, which is accessible to other PEs.

Such symmetric variables can be either statically allocated or dynamically allocated at run-time using a symmetric memory allocator. Thus data structures allocated from symmetric memory will have symmetric addresses on each PE. This makes it easy to address remote symmetric variables via locally generated addresses, and when enabled by the underlying hardware, allows SHMEM to do true "one-sided communication" with very low latency.

The SHMEM library has a long history as a parallel programming library. The SHMEM library was first introduced in 1993 by Cray Research Inc for their T3D systems. SHMEM was later adapted by SGI for its products based on the Numa-Link architecture and included in the Message Passing Toolkit (MPT). Other SHMEM implementations grew out of the SGI and Cray implementations. These implementations of the SHMEM API support C, C++, and Fortran programs; however, the differences between SHMEM implementations' semantics and APIs are subtle, resulting in portability and correctness issues.

Recently, the U.S. Department of Defense funded a collaboration between Oak Ridge National Laboratory and the University of Houston to develop a specification for a uniform SHMEM API. The OpenSHMEM specification was announced to address the divergence of the SHMEM APIs. In this poster we present our ongoing work on a reference implementation of OpenSHMEM and the challenges faced therein. We introduce the OpenSHMEM programming model and then touch on the key concepts of OpenSHMEM, our library development process, and our implementation based on the portable OpenSHMEM specification. Our goal is to make this library implementation observable and scalable by interfacing with profiling tools and making use of locality information. We will also use benchmarks to observe the speed-up obtained while using our library implementation and identify the strengths and weaknesses of our approach.

2. ACKNOWLEDGMENTS
I would like to thank Tony Curtis, Dr. B. Chapman, Jeffery Kuehn, and Stephen Poole for their guidance and support. Lastly, I would like to thank the Department of Defense for making this project possible.

Copyright is held by the author/owner(s).
ICS'11, May 31–June 4, 2011, Tucson, Arizona, USA.
ACM 978-1-4503-0102-2/11/05.