SHMEM-MT: A Benchmark Suite for Assessing Multi-threaded SHMEM Performance

Hans Weeks and Patrick Bridges
University of New Mexico

Matthew Dosanjh and Ryan Grant
Sandia National Laboratories
Problem

- We want to write performant OpenSHMEM programs
  - How many processing elements?
  - How many threads?
  - Optimal message size?
  - What synchronization methods?

- Very few benchmarks for OpenSHMEM
- None that address multi-threading
Solution

- We need a benchmark suite with:
  - Synchronization methods
  - Multi-threading
  - Variable message sizes

- All these elements are present in RMA-MT
  - Adapt the RMA-MT suite to OpenSHMEM!
RMA-MT Benchmark Suite

- Threaded, 1 sided MPI benchmarks
- Based on:
  - Thakur and Gropp’s multi-threaded benchmarks
  - Sandia Micro Benchmarks
  - Mantevo Mini apps
- 4 synchronization methods
RMA-MT Benchmark Suite

- Microbenchmarks:
  - Latency
  - Bandwidth
  - Message Rate - single direction, halo-exchange

- Mini-apps:
  - HPCCG
  - MiniFE
  - MiniMD
RMA-MT Benchmark Suite

- Microbenchmarks:
  - Latency
  - Bandwidth
  - Message Rate - single direction, halo-exchange

- Mini-apps:
  - HPCCG
  - MiniFE
  - MiniMD

Thakur and Gropp
Sandia Micro Benchmarks

Mantevo Mini Apps
Converting benchmarks to OpenSHMEM

RMA operations:
- `MPI_Get()`
- `MPI_Put()`
- `shmem_getmem()`
- `shmem_putmem()`

Initialization and allocation:
- `MPI_Init_thread()`
- `MPI_Win_create()`
- `shmem_init_thread()`
- `shmem_malloc()`

Synchronizations:
- Fence
- Lock
- Lock-all
- Post/Start/Complete/Wait
- Barrier
- Quiet
Converting benchmarks to OpenSHMEM

- Status
  - Bandwidth and latency converted to passive target
  - Message rate converted to active target
  - 2 mini apps converted to active target with MPI collectives remaining in non-critical paths
  - Coarse granularity of threading

Experimental setup

- Cray XE30m cluster
  - Sandia National Labs Volta cluster
  - Per node: 2x Xeon Ivy Bridge 2.4 GHz 12-core processor with hyper-threading enabled
  - 32 GB RAM
  - Cray Aries network interface
  - Cray-shmem 7.3.2

- 10 runs with 10,000 iterations per run
Micro-benchmarks
Latency

Latency (S) vs. Message Size

- 1 Thread
- 4 Threads
- 16 Threads

Message Size:
- 1B
- 2B
- 4B
- 8B
- 16B
- 32B
- 64B
- 128B
- 256B
- 512B
- 1KiB
- 2KiB
- 4KiB
- 8KiB
- 16KiB
- 32KiB
- 64KiB
- 128KiB
- 256KiB
- 512KiB
- 1MiB
Latency: Sub-optimal Protocol Switch

![Graph showing latency vs message size for different thread counts (1, 4, 16 threads) and message sizes ranging from 1B to 1MiB. The graph demonstrates the latency (in seconds) for various message sizes under sub-optimal protocol switching conditions.]
Bandwidth

![Graph showing bandwidth in MiB/S vs. message size. The x-axis represents message size in bytes, ranging from 1B to 1MiB. The y-axis represents bandwidth in MiB/S, ranging from 0 to 8000. There are three lines for different thread counts: 1 Thread (red), 4 Threads (green), and 16 Threads (brown). Each line shows an increase in bandwidth with increasing message size.]
Bandwidth: 4 thread overtakes single thread
Message Rate: 8 Node Halo Exchange

[Graph showing message rate vs. message size for different thread counts (1, 4, 16, 32) and message sizes ranging from 8B to 1 MiB.]
Mini-apps
Mini-apps

(a) HPCCG

(b) MiniFE
Future work

- Thread-based synchronization methods
- Remove MPI calls from non-critical paths in mini-apps
- Adding multi-threading computation component of mini-apps
- Add mini-apps with asynchronous algorithms
- Investigate SHMEM-MT performance on various network architectures
Acknowledgements

- Oak Ridge Leadership Computing Facility
- Scalable Systems Lab
- Sandia National Laboratory
  - *Sandia National Laboratories is a multiprogram laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.*
Questions?