OAK RIDGE National Laboratory

Using PMIx with OpenSHMEM at ORNL

Thomas Naughton

ECP PMIx BoF

March 31, 2021

ORNL Languages team (contributions from):
Ferrol Aderholdt, Matt Baker, Swen Boehm, Ed D'Azevedo, Oscar Hernandez, Graham Lopez, Thomas Naughton, Swaroop Pophale, Pavel (Pasha) Shamis, Manjunath Gorentla Venkata, Aaron Welch

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

OpenSHMEM

- OpenSHMEM
  - Standardized library interface for Parallel Global Address Space (PGAS) programming
  - Several implementations exist

    e.g., Cray shmem/oshmemx, Open MPI, OpenSHMEM.org reference, etc.

- ORNL OpenSHMEM implementation
  - Research vehicle for design & development of specification
    - Last update had most of v1.4 spec (all but C11 support)
  - Threading
  - Communication contexts
  - Modular communication conduits
    - UCX (default)
    - Libfabric (experimental)

How do you use PMIx?

- Past iterations for process management
  - Initially based on Gasnet runtime
  - Moved to custom runtime abstraction library ("librte")
    - Intended to provide more generic interface to different runtimes

- Revised approach for process management
  - In practice, often used OpenRTE (ORTE)
  - Moving toward OpenPMIx's reference runtime (PRRTE)
  - Desire to reduce maintenance of local "librte" interface
  - Move to PMIx for runtime abstraction

☺ *Initial PMIx support by Swen Boehm!*

PMIx usage

- Wireup modules: librte and PMIx

int shmemi_rte_init()	Inititialize library - set 'mype', 'numpes', - register info on memory segments (rkey, base), - "worker_address" for UCX, etc.
int shmemi_rte_finalize()	Finalize shmem library & call PMIx_Finalize
int shmemi_rte_abort(int)	Cleanup shmem & call PMIx_Abort
int shmemi_rte_barrier_all()	Used during shmem exit/abort on error path
int shmemi_rte_commit(void)	PMIx_Commit at end of comms_init
uint32_t shmemi_rte_my_pe()	Returns 'myproc.rank'
uint32_t shmemi_rte_group_size()	Get PMIX_JOB_SIZE for myproc

Questions posed for BoF…

- *What does PMIx make easier?*
  - Reduce OSH project's runtime maintenance
  - (I think) Streamline support for debuggers
- *What features are needed?*
  - For OSH, seem to be fine
  - Maybe: Interoperability with other programming models (e.g., SHMEM+X)
  - Maybe: Hardware topology details (e.g., hierarchical memory on nodes)
- *What would help?*
  - Reference implementations are extremely useful!
    - More frequent "stable" tags, or full release would be helpful

Acknowledgements

This work supported by the Exascale Computing Project (ECP), Project Number: 17-SC-20-SC, under the Open MPI for Exascale (OMPI-X) effort.

The ORNL OpenSHMEM work was supported by the United States Department of Defense (DoD) and used resources of the Computational Research and Development Programs at Oak Ridge National Laboratory.