PMIx: Process Management for Exascale Environments

$\text{PMIx} \times 10^{18}$
Agenda

• State of the Community
  ▪ Ralph H. Castain (Intel)

• Scaled Performance
  ▪ Aurelien Bouteiller (UTK)

• PMIx Standards Document
  ▪ Josh Hursey (IBM)

• Q&A
Where Are We?

• Launch scaling
  - Wireup enhancements complete
  - Fabric “instant on” enablement underway

• Results
  - Tracks spawn propagation time
  - Exascale in < 5 seconds
  - 3rd party confirmation
Stage 1

*RM daemon, mpirun-daemon, etc.
PMIx Launch Sequence

Stage 1
- Job script
  - FS
    - Files & retrieval times
  - WLM
    - PMIx Server

Stage 2
- Job Script
  - FS
    - Request file caching
    - Notify caching complete
  - WLM
    - Obtain fabric configuration
  - Fabric Mgr

Stage 3
- PMIx Server
  - Launch Cmd
  - RM*
    - Create shared memory storage for job info & topology
  - SM
    - Info as requested
    - SM location
    - Spawn
  - Fabric
    - Configure local fabric interfaces
  - NIC
    - Setup MPI interfaces
  - PMIx Client

Stage 4
- Cache
  - Files served from cache
  - Proc

*RM daemon, mpirun-daemon, etc.
Current Support (I)

- Typical startup operations
  - Put, get, commit, barrier, spawn, [dis]connect, publish/lookup
- Tool connections
  - Debugger, job submission, query
- Generalized query support
  - Job status, layout, system data, resource availability
- Event notification
  - App, system generated
  - Subscribe, chained
  - Preemption, failures, timeout warning, …
- Logging
  - Status reports, error output
- Flexible allocations
  - Release resources, request resources
Current Support (II)

• Network support
  ▪ Security keys, pre-spawn local driver setup
• Obsolescence protection
  ▪ Automatic cross-version compatibility
  ▪ Container support
• Job control
  ▪ Pause, kill, signal, heartbeat, resilience support (C/R coordination)
• Async definition of process groups
  ▪ Rolling startup/teardown
In Pipeline

- **Network support**
  - Fabric topology and status, traffic reports, fabric manager interaction
- **MPI Sessions support**
  - Rolling startup/teardown
- **Generalized data store**
  - Distributed key-value storage
- **Security**
  - Obtain and validate credentials for application/SMS
- **File system support**
  - Dependency detection
  - Tiered storage caching strategies
- **Debugger/tool support**
  - Automatic rendezvous
  - Single interface to all launchers
  - Co-launch daemons
  - Access fabric info, etc.
- **Cross-library interoperation**
  - OpenMP/MPI coordination
Three Distinct Entities

- **PMIx Standard**
  - Defined set of APIs, attribute strings
  - Nothing about implementation

- **PMIx Reference Library**
  - A full-featured implementation of the Standard
  - Intended to ease adoption

- **PMIx Reference Server**
  - Full-featured “shim” to a non-PMIx RM
  - Provides development environment
Adoption

• RMs
  ▪ SLURM, JSM complete – Fujitsu underway
  ▪ Altair ramping up

• Libraries
  ▪ OpenMPI, OSHMEM, SOS complete
  ▪ GASNet, ORNLshm – in PR
  ▪ MPICH to come (1Q2018?)

• Tools
  ▪ Debugger integration under development
Agenda

• State of the Community
  ▪ Ralph H. Castain (Intel)

• Scaled Performance
  ▪ Aurelien Bouteiller (UTK)

• PMIx Standards Document
  ▪ Josh Hursey (IBM)

• Q&A
Agenda

• State of the Community
  ▪ Ralph H. Castain (Intel)

• Scaled Performance
  ▪ Aurelien Bouteiller (UTK)

• PMIx Standards Document
  ▪ Josh Hursey (IBM)

• Q&A
Agenda

- State of the Community
  - Ralph H. Castain (Intel)
- Scaled Performance
  - Aurelien Bouteiller (UTK)
- PMIx Standards Document
  - Josh Hursey (IBM)
- Q&A