PMIx: Enabling Application-driven Execution at Exascale

Ralph H. Castain

PMIx $10^{18}$
Collaborative open source effort led by Intel, Mellanox Technologies, IBM, Adaptive Computing, and SchedMD. 

New collaborators are most welcome!
Motivation

• Exascale launch times are a hot topic
  ▪ Desire: reduce from many minutes to few seconds
  ▪ Target: $O(10^6)$ MPI processes on $O(10^5)$ nodes thru MPI_Init in < 30 seconds

• New programming models are exploding
  ▪ Driven by need to efficiently exploit scale vs. resource constraints
  ▪ Characterized by increased app-RM integration
Launch

Initialization

Exchange MPI contact info

Setup MPI structures

barrier

mpi_init completion

barrier

RRZ, 16-nodes, 8ppn, rank=0
What Is Being Shared?

- **Job Info (~90%)**
  - Names of participating nodes
  - Location and ID of procs
  - Relative ranks of procs (node, job)
  - Sizes (#procs in job, #procs on each node)

- **Endpoint info (~10%)**
  - Contact info for each supported fabric
Stage I

**MPI_Init**

- Launch
- Initialization
- Exchange MPI contact info
- Setup MPI structures
- mpi_init completion

**MPI_Finalize**

- Provide method for RM to share job info
- Work with fabric and library implementers to compute endpt from RM info

RRZ, 16-nodes, 8ppn, rank=0

Time (µsec)
Stage II

Launch
Initialization
Exchange MPI contact info
Setup MPI structures
mpi_init completion
mpi_barrier

MPI_Init

Add on 1st communication (non-PMIx)

RRZ, 16-nodes, 8ppn, rank=0
Stage III

Launch

Initialization

Exchange MPI contact info

Setup MPI structures

mpi_init completion

mpi_finalize

Use HSN

RRZ, 16-nodes, 8ppn, rank=0

Time (µsec)
How You Can Help

• Build OpenMPI
  ▪ Master or 2.x

• Run scaling test script
  ▪ contrib/scaling/scaling.pl
  ▪ README for instructions

• Email results
  ▪ PMIx or OMPI-devel mailing lists
  ▪ rhc@open-mpi.org
Changing Needs

- Notifications/response
  - Errors, resource changes
  - Negotiated response
- Request allocation changes
  - shrink/expand
- Workflow management
  - Steered/conditional execution
- QoS requests
  - Power, file system, fabric

Multiple, use-specific libs?
(difficult for RM community to support)

Single, multi-purpose lib?
Objectives

• Establish an independent, open community
  ▪ Industry, academia, lab
• Standalone client/server libraries
  ▪ Ease adoption, enable broad/consistent support
  ▪ Open source, non-copy-left
  ▪ Transparent backward compatibility
• Support evolving programming requirements
• Enable “Instant On” support
  ▪ Eliminate time-devouring steps
  ▪ Provide faster, more scalable operations
PMIx: Status

• Version 1.1 release
  ▪ Production version
  ▪ Released Nov 2015

• Server integrations underway
  ▪ SLURM
  ▪ Moab
  ▪ LSF
  ▪ ORTE/ORCM
  ▪ Others pending
PMIx v1.1 features

- Data scoping with 3 levels of locality:
  - local, remote, global.

- Communication scoping
  - PMIx_Fence across arbitrary subset of processes.

- *Point-to-point* "direct" data retrieval
  - Suited for applications with sparse communication graphs.

- Full support for non-blocking operations.

- Support for “binary blobs”
  - Reduces intra-node exchanges and encoding/decoding overhead

- Full support for MPI dynamic process management
Goal for SC’15

- Inform the community
- Solicit your input on the roadmap
- Get you a little excited
- Encourage participation

https://pmix.github.io/master
https://github.com/pmix

BoF: Thurs @ 12:15-1:15pm
Room 15