PMIx: Bridging the Container Boundary

Ralph H. Castain
Intel
Origin: Changing Landscape

Launch time limiting scale

Programming model & runtime proliferation

Legion

Hybrid applications

Model-specific tools

Container technologies

MPI

OpenMP

Spark

docker
Resolve launch scaling

- Pre-load information known to RM/scheduler
- Pre-assign communication endpoints
- Eliminate data exchange during init
- Orchestrate launch procedure
PMIx Launch Sequence

*RM daemon, mpirun-daemon, etc.*
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 RTE
  ▪ Full-featured “shim” to a non-PMIx RM
  ▪ Provides development environment
Where Is It Used?

- Libraries
  - OMPI, MPICH, Intel MPI, HPE-MPI, Spectrum MPI, Fujitsu MPI
  - OSHMEM, SOS, OpenSHMEM, …

- RMs
  - Slurm, Fujitsu, IBM’s JSM, PBSPro (2019), Kubernetes(?)
  - Slurm enhancement (LANL/ECP)

- New use-cases
  - Spark, TensorFlow
  - Debuggers (TotalView, DDT)
  - MPI
    - Re-ordering for load balance (UTK/ECP)
    - Fault management (UTK)
    - On-the-fly session formation/teardown (MPIF)
  - Logging information
  - Containers
    - Singularity, Docker, Amazon
Async event notification

Cross-model notification
- Announce model type, characteristics
- Coordinate resource utilization, programming blocks

Generalized tool support
- Co-launch daemons with job
- Forward stdio channels
- Query job, system info, network traffic, process counters, etc.
- Standardized attachment, launch methods
Sprinkle Some Magic Dust

- **Allocation support**
  - Dynamically add/remove/loan nodes
  - Register pre-emption acceptance, handshake

- **Dynamic process groups**
  - Async group construct/destruct
  - Notification of process departure/failure

- **File system integration**
  - Pre-cache files, specify storage strategies
PMIx-SMS Interactions

- OpenMP
- APP
- MPI
- PMIx Client
- PMIx Server
- Orchestration Requests
- Responses
- System Management Stack
  - FS
  - Fabric Mgr
  - Fabric
  - NIC
  - RAS
  - Job Script
  - Tool Support
PMIx-SMS Interactions

- PMIx Client
- PMIx Server
- OpenMP
- MPI
- APP
- Orchestration Requests
- Responses
- System Management Stack
  - FS
  - Fabric Mgr
  - Fabric
  - NIC
  - RAS
  - Job Script
  - Tool Support

Container!
Container Issues

- Version tracking across container boundary
  - Different pieces moving at different rates
- Container managers vs HPC schedulers
  - Dynamic, service related vs static, application focus
- Uneven adoption rates
  - Different environments adopt features at different times, different combinations
Version Tracking

- Auto-negotiate messaging protocol
- Client starts
  - Envar indicates server capabilities
  - Select highest support in common
  - Convey selection in connection handshake
- Server follows client’s lead
  - Per-client messaging protocol
  - Support mix of client versions
Container Issues

• Version tracking across container boundary
  ▪ Different pieces moving at different rates

• Uneven adoption rates
  ▪ Different environments adopt features at different times, different combinations

• Container managers vs HPC schedulers
  ▪ Dynamic, service related vs static, application focus
  ▪ Mismatched capabilities
- PMIx relay daemon/server
- Integrated into container
- Sense what SMS supports
  - From nothing to everything
- Supported requests
  - Relay requests/responses
- Unsupported requests
  - Execute internally
  - Return “not supported”
• PMIx relay daemon/server
• Integrated into container
• Sense what SMS supports
  • From nothing to everything
• Supported requests
  • Relay requests/responses
• Unsupported requests
  • Execute internally
  • Return “not supported”

*RM can perform request, but doesn’t have PMIx backend support for it
Call intrinsic APIs to execute PMIx requests from client

Treat the RM as an equal member of SMS

Pros

- Allows more transparent movement of containers across systems
- Reduces obstacles

Cons

- Reduces pressure on SMS vendors to integrate
• Allocation request
  ▪ Stabilize allocation for some period of time

• Event notification
  ▪ Handshake need to break commitment
  ▪ Notify when restored
  ▪ Use new FT/Sessions methods for flexible members
Services on HPC Systems

- **Job control API**
  - Register as a service
  - Request auto-restart, multiple replicas
  - Setup parallel duplicate IO streams

- **IO Forwarding APIs**
  - Construct data flows between processes

- **Storage APIs**

- **Publish/Lookup APIs**
  - Service discovery, rendezvous
Why Enable This?

Interchangeable
Summary

- Avoid having to write entire runtimes just to do something a little different
- Portability (HPC ↔ Service Mgrs)
- Generalized tools
- Scalable operations
- Async event notification
- Full system orchestration
Come Join Us!

Q&A

Useful Links:

General Information: https://pmix.org/
PMIx Library: https://github.com/pmix/pmix
PMIx Reference RunTime Environment (PRRTE): https://github.com/pmix/prrte
PMIx Standard: https://github.com/pmix/pmix-standard
Slack: pmix-workspace.slack.com

https://doi.org/10.1016/j.parco.2018.08.002