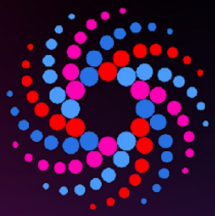


SC22
Dallas, TX | hpc accelerates.

Birds of a Feather: Charting the PMIx Roadmap

Joshua Hursey — IBM Corporation
Thomas Naughton — Oak Ridge National Laboratory (ORNL)
Ralph Castain — Nanook Consulting
Aurelien Bouteiller — University of Tennessee

Agenda & Logistics

- **Agenda**
  - 5 min: Introduction & Gathering
  - 10 min: PMIx Standard and Working Group updates
  - 10 min: Open PMIx projects update
  - 5-8 min: Short talk: Thomas Naughton (ORNL)
  - 5-8 min: Short talk: Howard Pritchard (LANL)
  - 10-15 min: Q&A and Open Discussion

Slack:
https://pmix-workspace.slack.com
#events

Mailing List:
https://groups.google.com/g/pmix-forum

You can find these slides after SC on the wiki:
https://github.com/pmix/pmix-standard/wiki#events

Presenters / organizers

Aurelien Bouteiller
The university of Tennessee
PMIx co-chair

Thomas Naughton
Oak Ridge National Lab.
PMIx secretary

Ralph H. Castain
Nanook Consulting
Open PMIx/PRTE lead

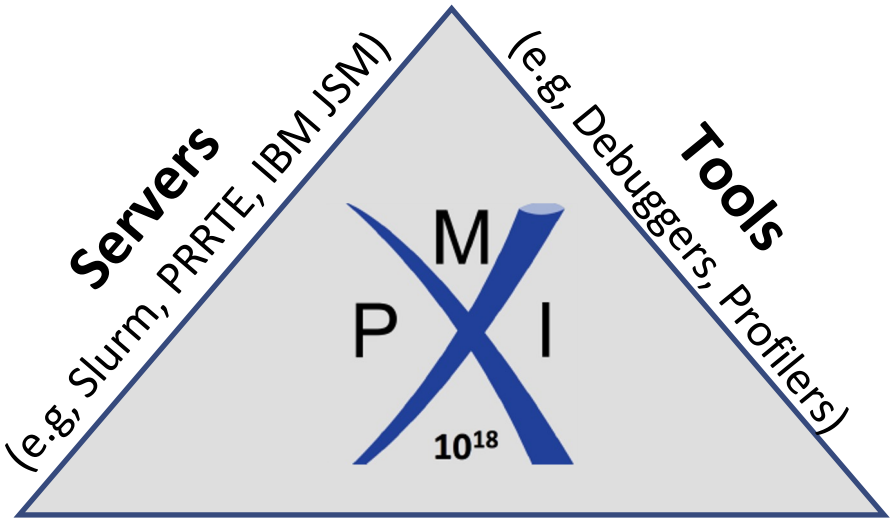
Joshua Hursey
IBM
PMIx co-chair

Howard Pritchard
Los Alamos National Lab.
MPI Sessions WG chair

What is PMIx?

PMIx is a standard API providing libraries and programming models with portable and well-defined access to commonly available system services.

- PMIx is a messenger between these pieces of software, not a doer.
  - Facilitates the interaction between applications, tools, runtime environments.
- Open, community driven standard.
- Use cases: (summarized list)
  - **Process wire-up** via either business card exchange or “instant on” (where supported)
  - **Tool connections** including debugger support
  - **Event notification** used by fault tolerant libraries
  - Application/Job/Node **environment discovery**
  - Job scheduler interaction



Clients

(e.g, MPI, OpenSHMEM)

https://github.com/pmix/pmix-standard
https://github.com/pmix/governance

PMIx Standard and Working Group Updates

November 17, 2022

PMIx Administrative Steering Committee (ASC)

The open governance body for the PMIx Standard with broad representation.

- Mission:
  - Define the mission of the PMIx Standard
  - Plan release timelines for versions of the PMIx Standard
  - Assemble working groups for areas of interest
  - Vote on all issues (1 vote per Member organization)
  - Vote on ASC leadership positions
- Meetings:
  - Quarterly ASC meetings to vote on items that move the PMIx Standard forward.
    - One quarter per year may be face-to-face.
  - Regular PMIx Standard teleconference to drive progress between quarterly meetings.
  - Working Group meetings to drive specific areas of interest and need.

PMIx Standard Recent Activities

- Releases:
  - PMIx 4.1 – Released Oct. 2021
  - PMIx 4.2 – Expected 1Q 2023
    - Errata and provisional items
    - Release Managers: Joshua Hursey (IBM) & Ralph Castain (Nanook Consulting)
  - PMIx 5.0 – Expected 1Q 2023
    - Large text changes, Use cases, ABI
    - Release Managers: Ken Raffenetti (ANL) & David Solt (IBM)
- Working Groups:
  - **Client Separation / Implementation Agnostic Document**
    - WG Champion: David Solt (IBM)
  - **Tools & Dynamic Workflows**
    - WG Champion: Isaías A. Comprés Ureña (TU Munich)

Implementation Agnostic Working Group

- **Goal**: Review the PMIx standard and rework text which assumes or requires a specific implementation of the standard
  - Multiple implementations of PMIx should be encouraged
  - The text should not assume a particular implementation architecture or design
  - Make a clear division between client/tool interfaces and system management software interfaces
  - Broaden the scope of PMIx from HPC only to distributed computing
- **Current activities:**
  - Re-work of how PMIx_Publish/Lookup/Unpublish work using a new API interface
  - Continue working through the standard (currently on Event Notification chapter)

Mailing List:
https://groups.google.com/g/pmix-forum-wg-impl-agnostic

Meeting Infomation:
Monday's 1 pm (US Central)
https://github.com/pmix/pmix-standard/wiki

Tools & Dynamic Workflows

- **Goals**:
  - Tools: Enhance shared-memory and distributed-memory tools' support and interoperability in supercomputing systems.
  - Dynamic Workflows: Extend current support for dynamic allocations and introduce new APIs that enable emerging dynamic processes workloads.
- **Current activities:**
  - Improving allocation requests support from PMIx clients.
  - Revising feedback and control use cases implemented with PMIx.
    - May lead to proposals in 2023
  - Working on a proposal for "resource advertisements" for dynamic workflows.
  - Resource acquisition, release, and cancelling ongoing acquisitions

Mailing List:
https://groups.google.com/g/pmix-forum-wg-tools

Meeting Infomation:
First Wed. 5 pm (Berlin time)
https://github.com/pmix/pmix-standard/wiki

Tools & Dynamic Workflows

- **Close collaboration with EuroHPC projects**:
  - Expanding existing support of PMIx in well established and emerging workload managers
  - Using PMIx as a common interface across multiple programming model's runtime systems:
    - OmpSs
    - Gaspi/GPI-2
    - GPI-Space
    - MPI: ParaStation MPI, MPICH, Open MPI
  - Using PMIx as a common interface across tools:
    - Shared memory: MemAxes, MUSA, PROFET
    - Distributed memory: Extrae, Paraver, Score-P, Extra-P, Scalasca
    - System and job monitoring: DCDB, LLview

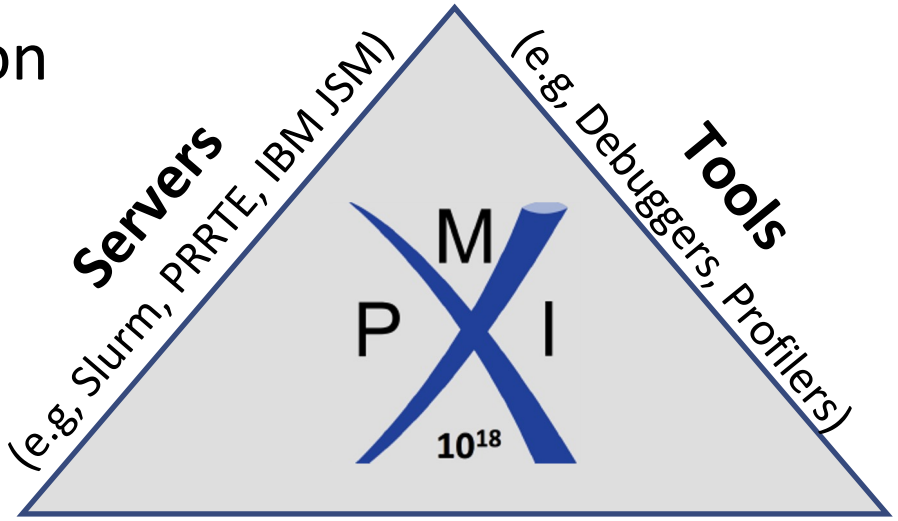

Open PMIx Projects Update

November 17, 2022

What is OpenPMIx?

OpenPMIx is a feature complete implementation of the PMIx standard.

- OpenPMIx provides the implementation to connect PMIx-enabled clients (like Open MPI) with PMIx-enabled Tools (like debuggers) and PMIx-enabled Servers (like PRRTE, SLURM, IBM JSM)
  - Works primarily as a works as a messenger between these pieces of software, not a doer.
- Open, community supported, scalable implementation
  - OpenPMIx releases tied to corresponding PMIx Standard releases
  - Proving ground for new PMIx standard additions
  - Currently used on many large scale HPC systems including all of the top 3 systems in the Top500 Nov. 2020 list.
  - Cross-version compatibility allowing clients to use a different version of OpenPMIx than the server or tool.



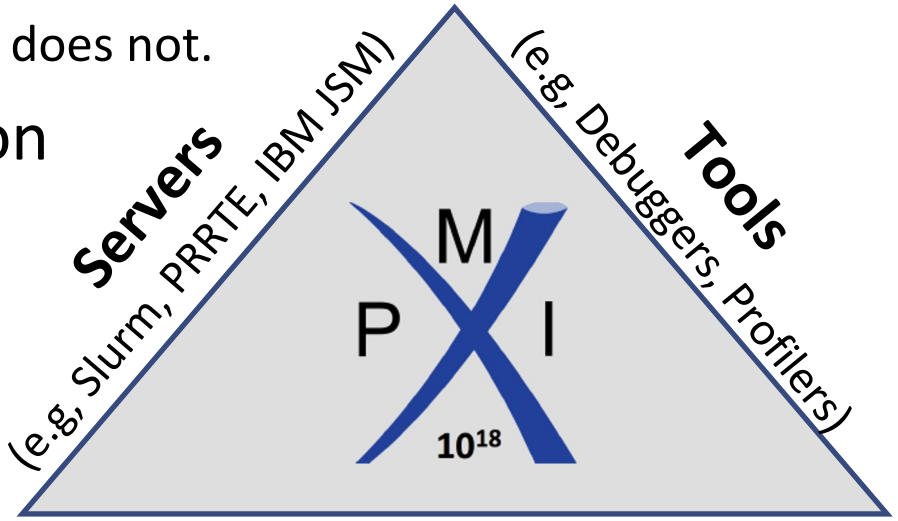
OpenPMIx Status

- v4.2.2 released
  - Minimum version required to support PRRTE v3.0 and above
  - Exposed broad range of infrastructure for reuse by PRRTE
    - Utility functions, including cmd line processing
    - Plugin and framework infrastructure
    - Class system and atomics
  - Fix handling of realm-specific info
    - Session, app, and node arrays
    - Retrieval rules
- v5.0 (mid-2023)
  - New shared memory subsystem
  - Scheduler-to-RTE integration points

What is PRRTE?

PMIx Reference RunTime Environment (PRRTE) is a featureful, scalable, PMIx-enabled runtime environment.

- PRRTE supports the PMIx standard interfaces needed for PMIx-enabled clients (like Open MPI) and PMIx-enabled Tools (like debuggers) to interact across HPC systems through the PMIx interface.
  - PRRTE provides a PMIx environment even if the host environment does not.
- Open, community supported, scalable implementation
  - Proving ground for new PMIx standard additions
  - Supports one-off jobs via prterun and multiple jobs via prte/prun
  - PMIx tool interface support (replacement for MPIR)
- Evolution of the ORTE runtime from Open MPI into a standalone project.



Clients

(e.g, MPI, OpenSHMEM)

PRRTE Status

- v3.0.0 released
- Major refactoring
  - Large code reduction by reuse of OpenPMIx internals
- Multi-purpose
  - Adopted by Open MPI for its internal RTE
  - Utilized by Altair's OpenPBS and PBS Professional
    - Support Open MPI direct launch (i.e., outside of "mpirun")
  - Still utilized as DVM by multiple organizations
    - As a “shim” to less supporting RM environments
    - Support workload management projects

Next stages: Scheduler integration support

PRRTE v3.0 Highlights

- Added one-shot operation option
  - “prterun” – start DVM and execute single application, then shutdown DVM
- Introduction of personalities for customized cmd line definitions
  - “prte” (default), “ompi” – soon to come include “srun” and “hydra”
  - Allows “prun” and “prterun” to proxy other native cmd line definitions
  - Passed to PMIx for lower-level support
    - Allows PMIx to level programming model/library support across RMs
- Addition of new command line directives
  - --output
    - tag stdout/stderr streams in variety of ways, redirect to files
  - --display
    - display allocation, bindings, placement map, node topologies
  - --runtime-options
    - launch and error termination behavior, exec agent definitions, debugger controls, forward local environment

PRRTE v3.0 Highlights (continued)

- Extensive “--help” support system
  - Replaces “man” page system
  - Hierarchical
    - “--help” provides overview of options for that cmd
    - “--help <foo>” provides detailed help on the “foo” cmd line option
  - Added non-cmd option topics
    - “--help placement” to access process placement explanation, examples
- Reduction in the “--rank-by” options
  - Replaced “--rank-by object” with “fill” and “span”
- Many, many bug fixes

Supporting many-task Python workflows with PMIx

Thomas Naughton, Wael Elwasif (ORNL)

Matt Baker (Voltron Data)

November 17, 2022

Many-task Ensembles in HPC

- Executing ensembles of tasks is a core part of scientific workflows, but face challenges
  - Use platform specific commands
  - Often limited to shell utilities
    - Example: Arbitrary delays between task launches

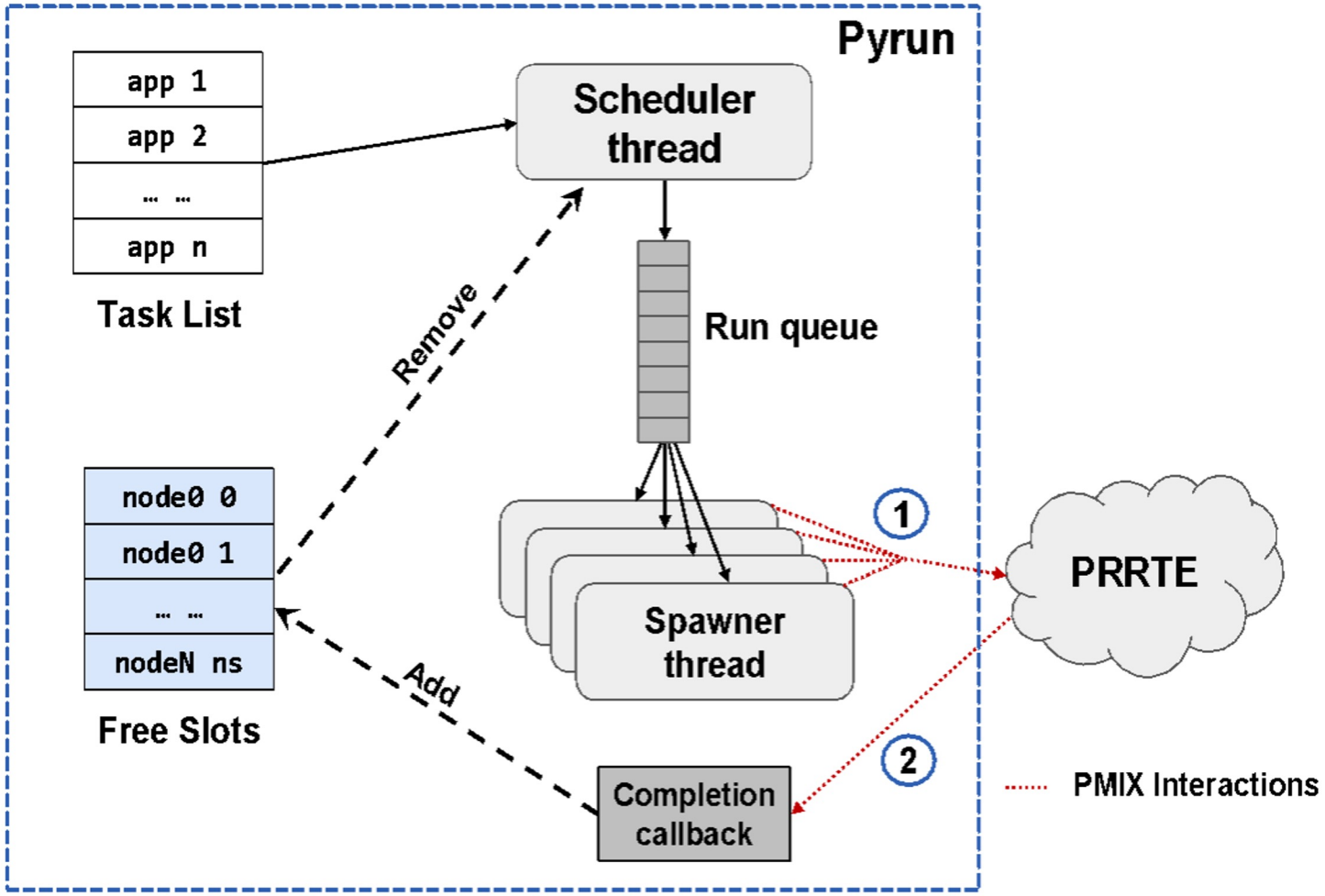
- PMIx can help
  - Offers a uniform interface across systems
  - Improved insights & handling of process lifecycle
  - Python binding more attractive option than C interface for these workflows

Recent work

- Python bindings added in PMIx v4.0
  - Capable to use full set of Client/Tool functionality
- Helping to harden support in OpenPMIx
  - Resolve threading issue with Python GIL
  - Resolve some type conversion issues
- Developed *pyrun* prototype to explore capabilities
  - Python tool to execute ensemble of tasks
  - Focused on user-driven scheduling

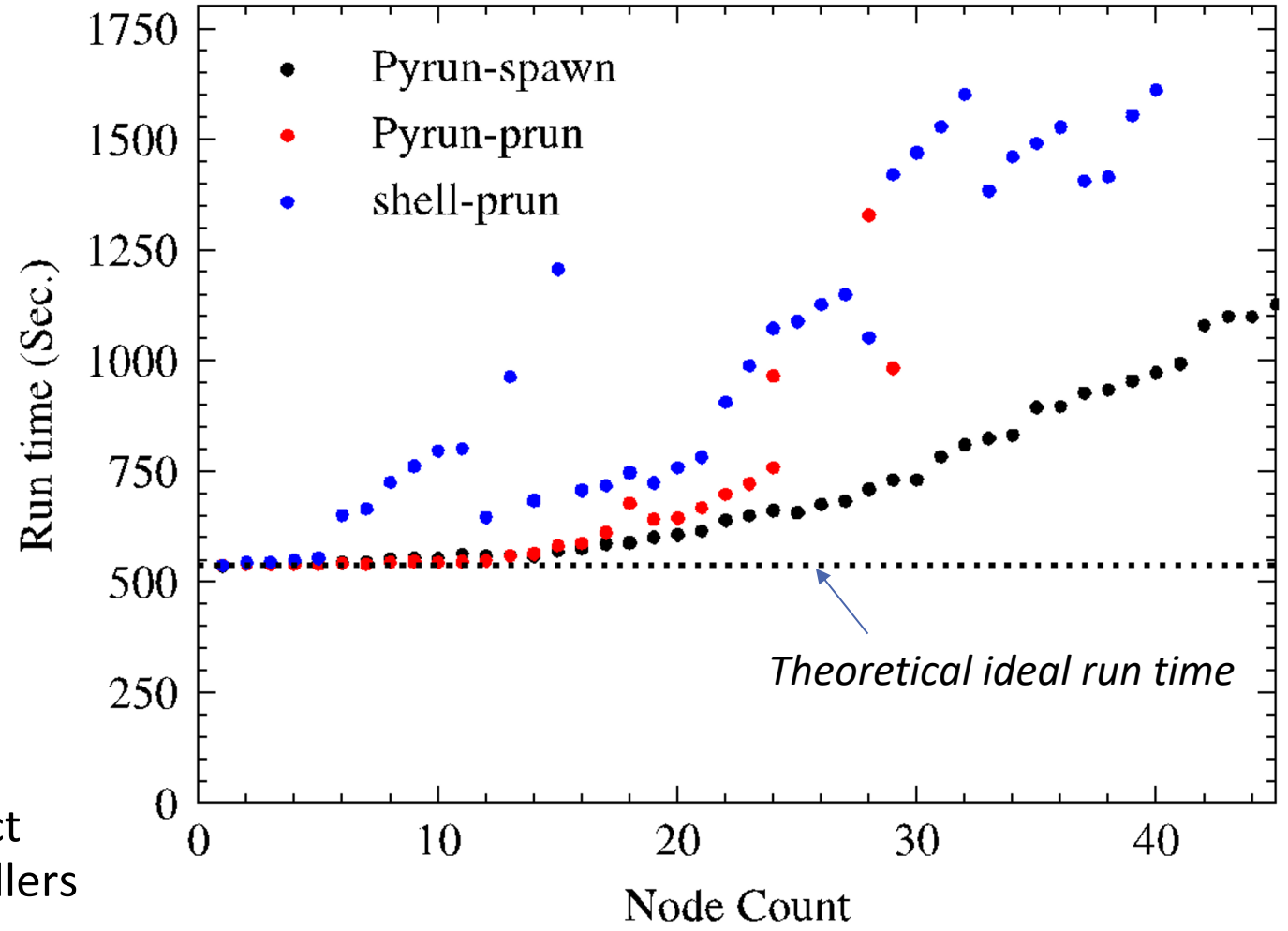
User-driven Scheduling with Pyrun

- Task List (applications)
  - Executable & arguments
  - Number of processes
    - Ex. MPI processes
- Scheduler
  - Free slot tracking
    - Generic counter, or Specific node(s)
  - FIFO queue with back-fill
  - Spawner threads consume tasks
  - Add/Remove slot tracking
- PMIx standard
  - Spawn tasks (1) (2)
  - Callbacks & Events



Comparison with command-line tools

- Execution <*driver*>-<*method*>
  - Shell script w/ prun
  - Pyrun w/ popen of prun
  - Pyrun w/ direct spawn()
- Workload
  - MPI “sleeper” (times 150-180s)
  - #Tasks = 3 x (#Nodes x 168 cores)
- Remarks
  - Shell-prun need arbitrary delays
  - Pyrun spread work over more nodes b/c start when jobs end
  - Pyrun-prun has more overhead than direct interaction w/ Pyrun-spawn & event handlers



Summit @ OLCF, Using OpenPMIx/PRTE from Open MPI 5.0.0rc6

Summary

- PMIx offers standard interface for process management
- Foundations are there for wider adoption using Python
- Ability for user-driven scheduling of without resorting to arbitrary delays
- Encourage further support of PMIx functionality by system vendors & workload management software

Acknowledgements

- **OpenPMIx**: Ralph Castain & Danielle Sikich with support from Intel and Argonne Natioinal Laboratory

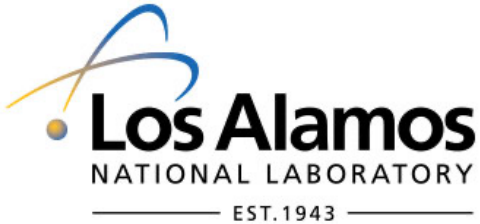
- **OLCF**: This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

- **ECP**: This research was partially supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

PMIx 10^18

Short Talk:
PMIx and MPI Sessions, etc.

Howard Pritchard (LANL)

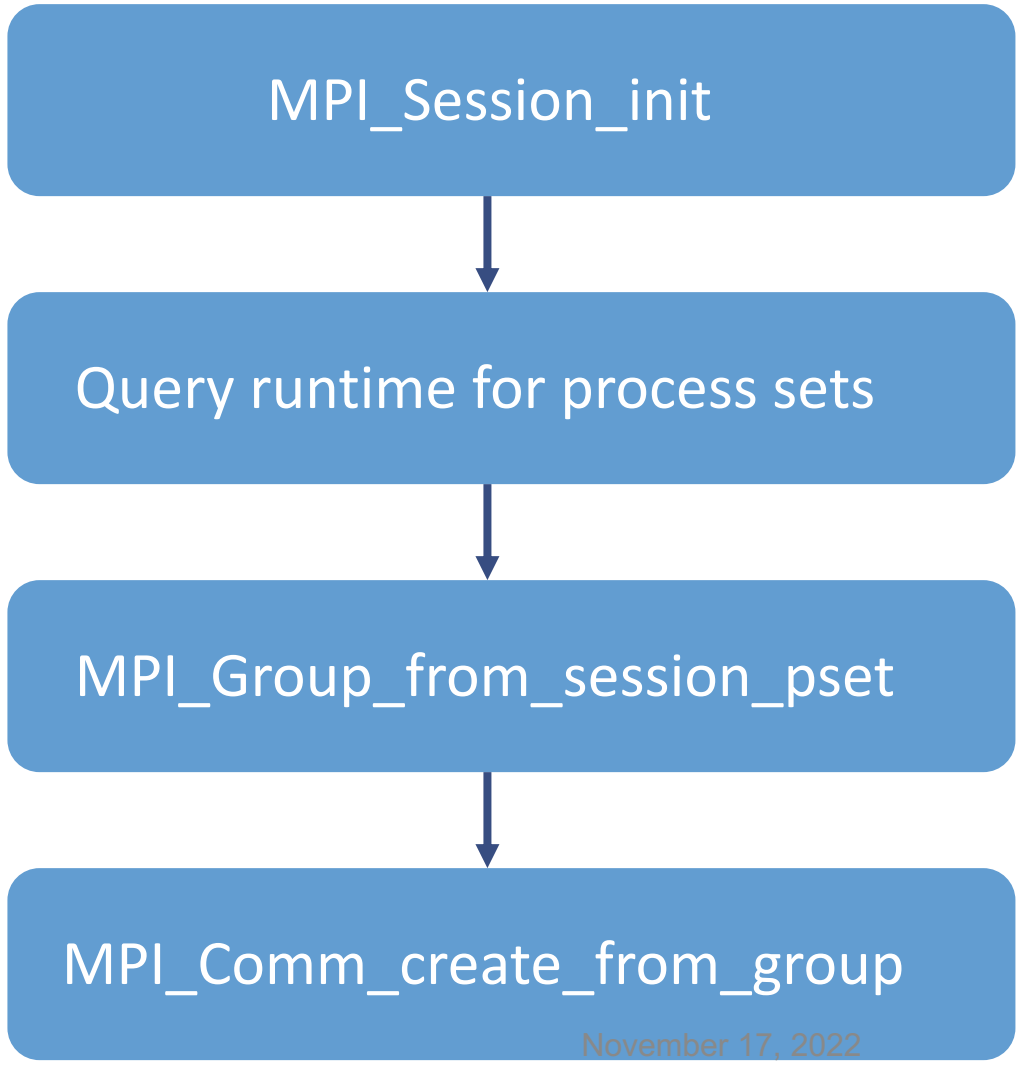


LA-UR-22-32131

November 17, 2022

MPI Sessions (4.0 version) and PMIx

PMIx calls used currently (Open MPI)



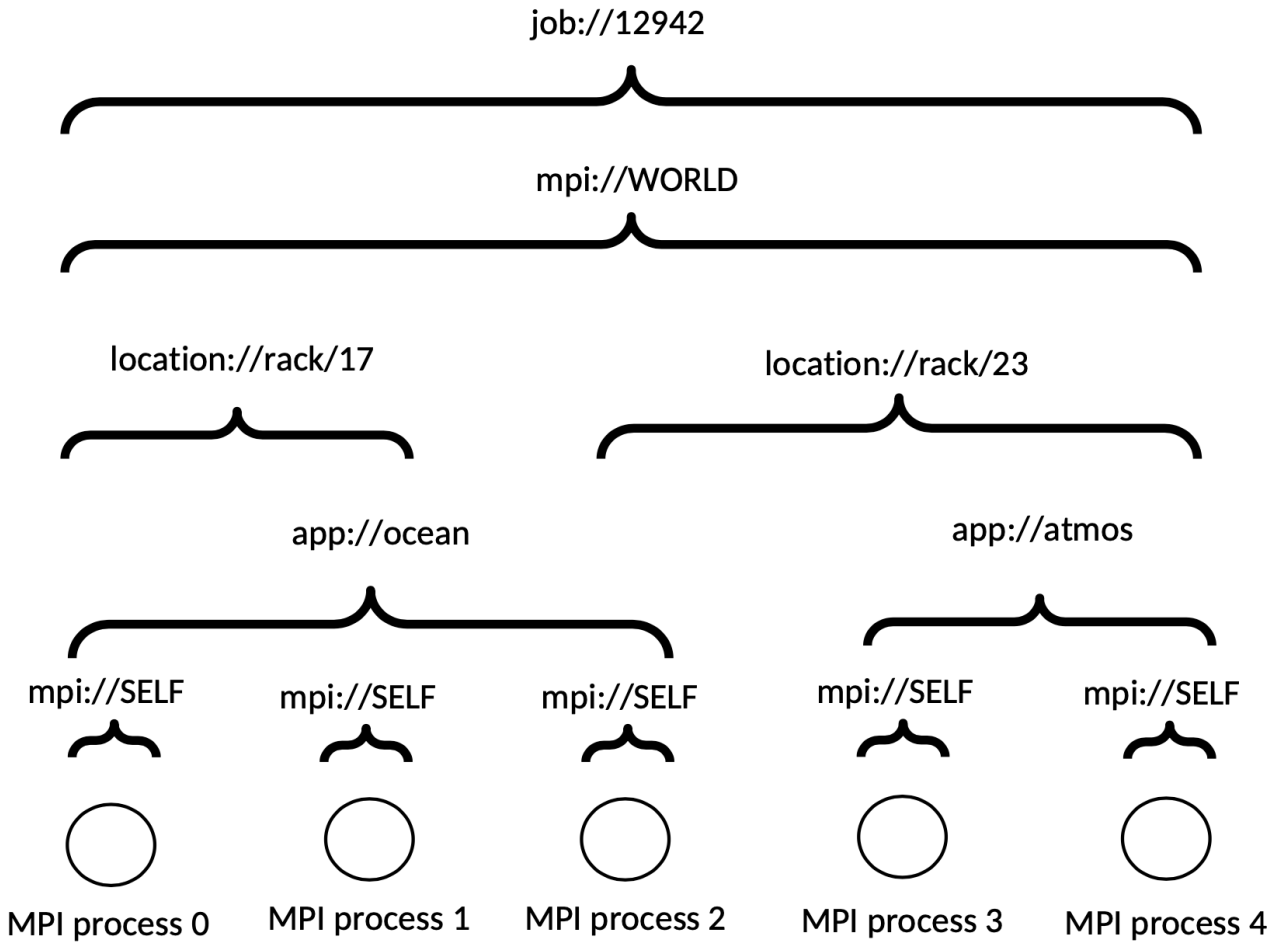
PMIx_Query_info – PMIX_QUERY_NUM_PSETS, PMIX_QUERY_PSET_NAMES

PMIx_Query_info - PMIX_QUERY_PSET_MEMBERSHIP (not for mpi://world and mpi://self)

PMIx_Group_construct/destruct

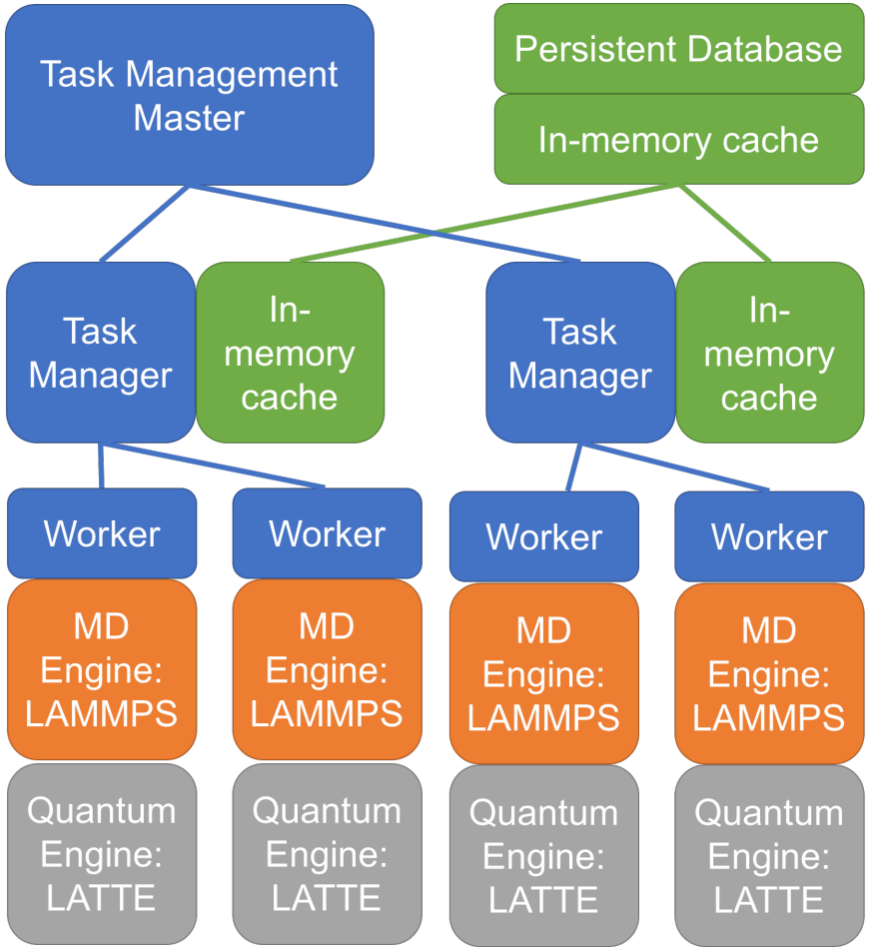
MPI Sessions (4.0 version) and PMIx Process Sets

- Figure from the MPI 4.0 standard – illustrates possible process sets defined by the runtime at application launch
- This maps well to the PMIx Process Set definition (sec. 13.1 of PMIx 4.1 std)
- MPI Standard does have wording to indicate a runtime can create additional process sets after application launch
- *Upshot is PMIx Process Set definition does not map directly to the process set terminology in the MPI standard. This is okay.*



MPI 5 and Better Support for Malleable Applications

- Increasing number of HPC workflows could benefit from a more elastic runtime and resource scheduling environment
- MPI Sessions working group is exploring approaches within the context of MPI to expose capabilities of such a more elastic runtime to the application without introducing too much complexity



Exaalt infrastructure (ECP ADSE04)

PMIx Support for Malleable MPI Applications

PMIx 4 defines methods that, in principle, would provide much of the functionality needed to support approaches the MPI Sessions WG is considering for MPI5:

- Job management including resource allocation, job control, etc.
- Group management methods, e.g. *PMIx_Group_construct*, *PMIx_Group_invite*
  - PMIx groups can be mapped to MPI process sets
- Process creation – *PMIx_Spawn*

Numerous challenges, however, including PMIx server support for this functionality, limitations in resource management systems, etc.

What functionality would we want to expose through MPI verses more runtime oriented interfaces?

Some related Presentations at SC22

- Martin Schulz gave a talk - ***Adding Malleability to MPI: Opportunities and Challenges*** at the **ESPM2 2022** workshop on Monday
- A BOF **- Enabling I/O and Computation Malleability in High-Performance Computing -** which took place on Wednesday
- Some presentations at WORKS22 workshop on Monday

Question & Answer
Open Discussion

November 17, 2022

Thank you for participating today!

Next PMIx Standard ASC Monthly Teleconf:
Thurs., Dec. 8, 2022
https://github.com/pmix/pmix-standard
https://github.com/pmix/governance

Next PMIx Standard ASC Quarterly Meeting:
Tues., Jan. 24 & 26
Mailing List: https://groups.google.com/g/pmix-forum
Slack: https://pmix-workspace.slack.com