LA-UR-20-24996
Approved for public release; distribution is unlimited.

Title: Open MPI Sessions Prototype and PMIx

Author(s): Pritchard, Howard Porter Jr.
Holmes, Daniel

Intended for: presentation to a PMIx WG

Issued: 2020-07-08

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Open MPI Sessions Prototype and PMIx

Daniel Holmes (EPCC)
Howard Pritchard (LANL)

7/8/20

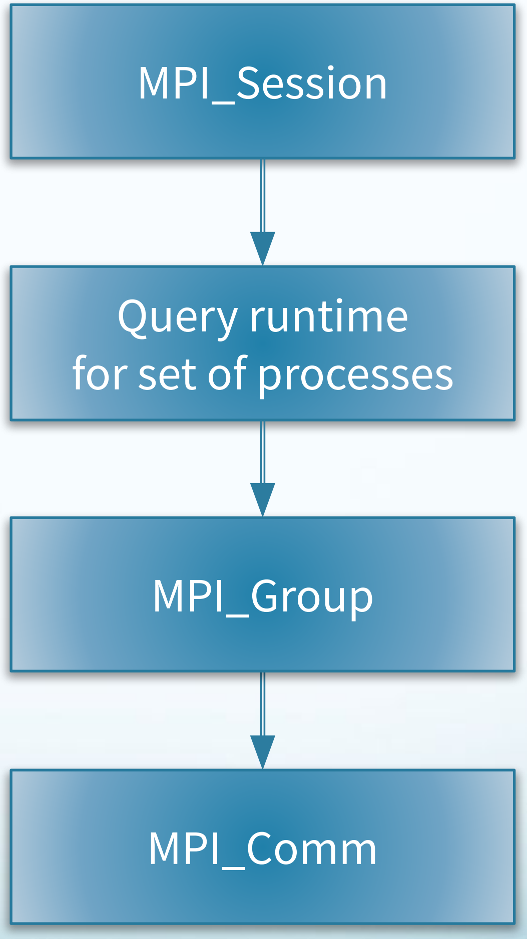
Problems with MPI_Init

- All MPI processes must initialize MPI exactly once
- MPI cannot be initialized within an MPI process from different application components without coordination
- MPI cannot be re-initialized after MPI is finalized
- Error handling for MPI initialization cannot be specified

Sessions – a new way to start MPI

- General scheme:
  - Query the underlying run-time system
    - Get a “set” of processes
  - Determine the processes you want
    - Create an MPI_Group
  - Create a communicator with just those processes
    - Create an MPI_Comm

Could be PMIx



MPI_Session

Query runtime for set of processes

MPI_Group

MPI_Comm

MPI Sessions proposed API

- Create (or destroy) a session:
  - MPI_SESSION_INIT (and MPI_SESSION_FINALIZE)
- Get names of sets of processes:
  - MPI_SESSION_GET_NUM_PSETS, MPI_SESSION_GET_NTH_PSET

PMIx PSETs used here

- Create an MPI_GROUP from a process set name:
  - MPI_GROUP_FROM_SESSION_PSET

PMIx PSETs used here

- Create an MPI_COMM from an MPI_GROUP:
  - MPI_COMM_CREATE_FROM_GROUP

PMIx groups used here

MPI_COMM_CREATE_FROM_GROUP

```
MPI_Comm_create_from_group(IN MPI_Group group,
                           IN const char *uri,
                           IN MPI_Info info,
                           IN MPI_Erhandler hndl,
                           OUT MPI_Comm *comm);
```

- The *uri* is supplied by the application.
- *uri* is different from the process name.
- Implementation challenge: *group* is a local object.
- Need some way to synchronize with other “joiners” to the communicator.

Using PMIx Groups

- PMIx Groups - a collection of processes desiring a unified identifier for purposes such as passing events or participating in PMIx fence operations
  - Invite/join/leave semantics
- Sessions prototype implementation currently uses PMIX_Group_construct/PMIX_Group_destruct
- Can be used to generate a “unique” 64-bit identifier for the group. Used by the sessions prototype to generate a communicator ID.
- Useful options for future work
  - Timeout for processes joining the group
  - Asynchronous notification when a process leaves the group

Using PMIx_Group_Construct

```
PMIx_Group_Construct(const char id[],
                     const pmix_proc_t procs[],
                     const pmix_info_t info[],
                     size_t ninfo,
                     pmix_info_t **results,
                     size_t nresults);
```

- ‘id’ maps to/from the ‘uri’ in MPI_Comm_create_from_group (plus additional Open MPI internal info)
- ‘procs’ array comes from information previously supplied by PMIx
  - “mpi://world” and “mpi://self” already available
  - prun -n 2 --pset user://ocean ocean.x : \
    -n 2 --pset user://atmosphere atmosphere.x
  - Have an issue with generic case (a little more later)

MPI Sessions Prototype Status

- Fully functional - implements the MPI Sessions functionality to appear in MPI 4.0 standard
- C and Fortran interfaces implemented
- Currently prototype only supports Sessions API for the PML/OB1 messaging component
- https://github.com/hpc/ompi/tree/sessions_new

MPI Sessions Prototype TODOs

- Add support for using Sessions over other network APIs.
  - First target is OFI libfabric
- Address various outstanding issues: https://github.com/hpc/ompi/issues
- Prepare pull request to merge into Open MPI master (post branch of next major Open MPI release)

Future work

- Address potential scalability issues (both on PMIx and OMPI sides)
  - Procs arg to PMIx_Group_construct (PMIx)
  - Per OMPI proc memory needed for extended CID handling (OMPI)
- Procs need to be associated with multiple PMIX_PSET_NAMEs (PMIx)
- Enhance mechanism for creating PMIx PSETs (PMIx)
- Handling (unexpected) process exit (OMPI)
- Group expansion (PMIx)
- Investigate use of Sessions in various workflows (tried DASK) (OMPI/PMIx)

Funding Acknowledgments

ECP

EXASCALE COMPUTING PROJECT

EPiGRAM HS