Using PMIx to Help Replace MPI_Init

14th November 2018

Dan Holmes, Howard Pritchard, Nathan Hjelm

LA-UR-18-30830

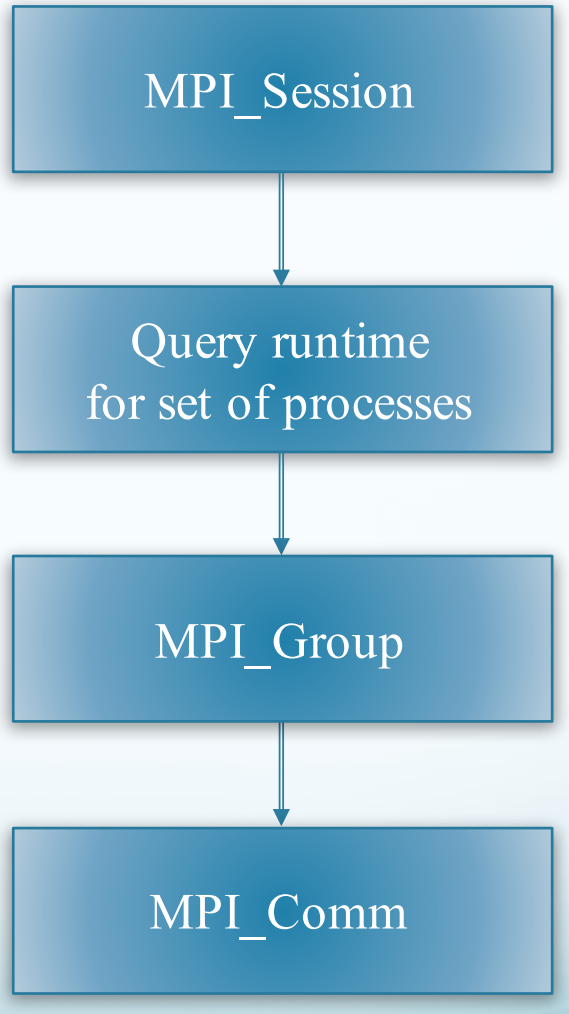
Problems with MPI_Init

- 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 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
- Create an MPI_GROUP from a process set name:
  - MPI_GROUP_CREATE_FROM_SESSION
- Create an MPI_COMM from an MPI_GROUP:
  - MPI_COMM_CREATE_FROM_GROUP

PMIx groups helps here

MPI_COMM_CREATE_FROM_GROUP

```
MPI_Create_comm_from_group(IN  MPI_Group  group,
                                                    IN cc
*uri,
                                                    IN MP
info,
                                                    IN
MPI_Erhandler  hndl,
                                                    OUT M
*comm);
```

The ‘uri’ is supplied by the application.

Implementation challenge: ‘group’ is a local object. Need some way to synchronize with other “joiners” to the communicator. The ‘uri’ is different than a process set name.

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 pro
                                    const pmix_info_t in
                                    Size_t ninfo);
```

- '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
  - mpiexec -np 2 --pset user://ocean ocean.x : \
    -np 2 --pset user://atmosphere atmosphere.x

Work in progress

MPI Sessions Prototype Status

- All "MPI_*_from_group" functions have been implemented
  - Only pml/ob1 supported at this time
- Working on MPI_Group creation (from PSET) now
  - Will start with mpi://world and mpi://self
  - User-defined process sets need additional support from PMIx
- Up next: break apart MPI initialization
  - Goal is to reduce startup time and memory footprint

Summary

- PMIx Groups provides an OOB mechanism for MPI processes to bootstrap the formation of a communication context (MPI Communicator) from a group of MPI processes
- Functionality for future work
  - Handling (unexpected) process exit
  - User-defined process sets
  - Group expansion

Funding Acknowledgments