Process Management Interface – Exascale

Debugger/Tool Interactions

- Query support
  - User-level query of system-level info
- Attach to running job
  - Launch debugger
  - Query status, “published” info
  - Register for event notifications
- Application startup
  - With/without debugger

Portable Interface

Rendezvous

- PMIx server provides tool rendezvous point
  - Only if directed to do so with PMIx_server_init key
  - Standard name/location
    - User-level daemons: $TMPDIR/pmix.tool.<pid>
    - System-level daemons: $TMPDIR/pmix.system.<pid>
      - Only one/node!
    - Unix domain socket comm => pipe
    - Non-usock => file containing URI
- PMIx_tool_init
  - Scan $TMPDIR for standard names
    - Returns error if multiple found
  - Accepts directives
    - pid: Finds specific rendezvous or returns error
    - System vs user-level daemon preference/constraint

Connection Requests

- Possibly still subject to PMIx authentication
  - Depends on configured security mode
- Forwarded to RM
  - Uid/gid of requestor
  - Any provided directives
    - Forward stdin/stdout/stderr
- RM must authorize connection
  - Return success if accepted
  - Return identifier (nspace,rank) assigned to connector
  - RM retains responsibility for authorizing each subsequent request

Query Support

- PMIX_QUERY_AUTHORIZATIONS
  - What operations tool is authorized to perform
- PMIX_QUERY_NAMESPACES
  - List of active jobs
- PMIX_QUERY_JOB_STATUS
  - Given nspace, report current job status
    - Running, paused, queued
- PMIX_QUERY_QUEUE_LIST
  - List of scheduler queues
- PMIX_QUERY_QUEUE_STATUS
  - List of jobs pending on specified queue
- PMIX_QUERY_PROC_TABLE
  - Array of structs containing proc-id, hostname, executable_name, pid, exit-code, and process state

More being defined!

Attach to Running Job

- Depends on environment and launch method
  - Mpirun: pass pid to PMIx_tool_init
  - Auto-connect to system PMIx server
    - PMIx-based launch
      - Tool connects to system server via PMIx to request application launch
    - Custom launch tool (e.g., srun)
- RM responsible for authorization of each operation
- Current use-cases
  - Launch debugger daemons
    - Provide debugger daemon support (e.g., wireup, overlay network)
    - Provide daemons with global/local proc info for specified nspace on launch
    - Specified pattern (1/node, 1/application proc, …), target nspace
  - Query information, status
  - Register for events

Event Registration

- Request to receive callbacks
- Job-related events
  - Job changes state
  - Process failures
  - Allocation received
- Negotiate response
  - Query available actions
  - Direct response

Debugger/App Startup

- Requires system PMIx launch support
- Tool connects to local system PMIx server
  - Packages application and debugger executables into single PMIx_Spawn request
    - Each as separate ”app” entry in array
    - Debugger marked as PMIX_DEBUGGER_DAEMONS
      - Can include debugger directives in application’s info
      - Debugger daemons not included in MPI_COMM_WORLD
  - Submit to server for spawn
- Server returns status of result
  - Includes nspace of application

Status

- Still under design
  - Feedback desired!
- Prototype in OMPI branch
  - Tool connection support already committed
  - Estimate debugger complete in Aug-Sept
- RFC for PMIx
  - Could be ready for end-Aug
    - Pending adoption by debugger community!
  - Coordinate release with debugger, RM, MPI communities