



IBM HPC Development

# Parallel Environment - Today and Tomorrow

SPXXL/Scicomp – San Francisco

May 12, 2010

Chulho Kim

Parallel Environment & HPC Protocols Architect

The futures material presented represents a mix of experimentation, prototyping and development.

While topics discussed may appear in some form in future IBM products there is no guarantee any particular feature will appear precisely as described.

Some work described may never go farther than prototype form.

# Features introduce on PE 5.1 on AIX/pLinux in 11/2008

# Parallel Environment 5.1

- **AIX 5.3, AIX 6.1, SLES 10SP2, SLES11, RHEL5.3 SW support**
  - **Includes HPC Toolkit – performance analysis and tuning tools**
    - **hpmcount is excluded due to perfmon2 patch not in kernel – but downloadable (pLinux)**
  - **Support PDB (new command line parallel debugger)**
    - **Works on AIX and Linux – uses base OS debugger (dbx, gdb)**
      - **POE task group management**
      - **Control input/output and some data aggregation support**
  - **MPI F90 Compile Time Checking (xLF v12)**
  - **Support for User Space network statistics and debug infrastructure (IB only)**
  - **LAPI Multicast APIs**
  - **Small subset of MPI collectives (MPI\_Bcast and MPI\_Barrier ) coded over LAPI active message - performance**

# Parallel Environment PE 5.1

- New Environment variables introduced in service
  - **MP\_SYNC\_QP**
    - Default on PE 5.1 is 'YES'
    - Default on PE 4.3 is 'NO'
  - **MP\_SHM\_ATTACH\_THRESH**
    - Default 32K \* log(common tasks)
  - **MP\_RFIFO\_SIZE**
    - Ability to change default RFIFO size, especially important in sn\_all case and want to save memory. Default is 4MB for < 256 tasks and 16MB for larger task counts. 8 way striping can cause each process to take 32MB. If application does not do too much multi-to-one then setting 2MB can help
  - **FIFO Adapter Affinity default is 'Yes'**
    - **LAPI\_DEBUG\_ENABLE\_AFFINITY** can be used to disable it
  - **MP\_FIFO\_MTU** – default is 2K, Support for 4K MTU on IB
    - User needs to set.
    - Works only if IB switch is setup to handle 4K MTU
  - **IBM HPC Tuning Guide for Power6** was released in April 2009 – will put it up on HPC Central wiki

# HPC LPP Support Withdrawals

- Last releases with support for JS21 Cisco PCIE IB, Power5 Galaxy1 IB and HPS
  - **PE 5.1**
  - **TWS LL 3.5**
  - **Parallel ESSL 3.3/3.3.1**

# SciComp 2009 Input

- 4 requests
  - **Support for multiple level of PE support in the cluster like they have for compilers**
    - Makes the transition support for users more transparent and help users compare old versus new
    - PE can support this but not well documented and prereqs (i.e. LAPI) need to be saved as well
    - **NOTE: This is being investigated**
  - **getrusage() from POE/PMD on per node/task basis (node preferred)**
    - Use new environment variable to have getrusage() info printed out for user to process
    - NERSC states that the information that LL provides is not useful
    - **NOTE: LL uses getrusage() call so would like to understand the issue better**
  - **Better MPI-IO support by having ability to have jobs be able to specify multiple resources (some subset request I/O servers with more memory or I/O) and put MPI-IO agents run on those nodes**
    - DKRZ requirement
    - **NOTE: MPI was enhanced with PE 5.2, users should try and give feedback**
  - **Better MPMD runtime management**
    - This can be satisfied with POE support of dynamic tasking
    - **NOTE: This is being investigated**

# Features introduce on PE 5.2 on AIX in 04/2010

# Parallel Environment 5.2

- AIX 5.3, AIX 6.1, SLES 11SP1 SW support
  - PE MPI supports dynamic process management, the process creation and management capability provided by MPI-2.1 using initial static allocation of resources
  - PE 5.2 provides new environment variables for controlling the number of active I/O agents assigned to a job. Reducing the number of active I/O agents, which require overhead and coordination, should be considered.
  - A new OpenSSH-based user authentication method is provided with PE 5.2. With this release, support for the cluster based security (CTSec) method is removed.
  - PE 5.2 includes improvements in the ability to run POE in large scale environments.
  - In addition to MPI and LAPI, PE 5.2 provides support for other parallel programming APIs such as UPC, X10 and Charm++, and you can mix any combination of these APIs in a single parallel program. Options are also included for specifying how the parallel APIs in your application will use system resources.
  - PE 5.2 supports launching User Space jobs, using the InfiniBand interconnect, without the use of a resource manager, such as LoadLeveler. Instead of using a separate resource manager, POE manages the adapter resources itself, on behalf of the user application.
  - PE 5.2 defines and utilizes a set of common resource management interfaces, for use by LoadLeveler and other resource managers.
  - For AIX users, beginning with PE 5.2, the underlying framework for the checkpoint and restart function has changed. POE now utilizes the *IBM MetaCluster Distributed Checkpoint Restart* (MDCR) function, and its associated components (Application WPAR), to coordinate the checkpointing and restarting of jobs.
  - PDB, PE's command line parallel debugger, has been enhanced for PE 5.2 with several new features, including the ability to run PDB on multiple consoles.

# MPI Process Management

# MPI 2.1 Dynamic Process Management

- New APIs
  - **Creating new processes**
    - **MPI\_COMM\_SPAWN, MPI\_COMM\_SPAWN\_MULTIPLE**
    - **MPI\_COMM\_GET\_PARENT**
  - **Server side Routines**
    - **MPI\_OPEN\_PORT**
    - **MPI\_CLOSE\_PORT**
    - **MPI\_COMM\_PARENT**
    - **MPI\_PUBLISH\_NAME**
    - **MPI\_UNPUBLISH\_NAME**
  - **Client Side connections**
    - **MPI\_COMM\_CONNECT**
    - **MPI\_LOOKUP\_NAME**
  - **Another way to establish connection**
    - **MPI\_COMM\_JOIN**
  - **Releasing Connection**
    - **MPI\_COMM\_DISCONNECT**
  - **New MP Environment variable and poe option**
    - **MP\_WORLD\_SIZES** or **-world\_sizes**

# MPI-IO Enhancements

# MPI-IO Enhancements

- **New MP Environment variable and poe option**
  - **MP\_IOAGENT\_CNT or -ioagent\_cnt**
    - Specify # of I/O agents for a job, default is 1 I/O agent per node
    - Range is 0 to ‘all’
  - **MP\_IOTASKLIST or -iotasklist**
    - Specify which task(s) to be ioagent
    - **Example:** MP\_IOTASKLIST=4:2:4:6:8
      - » First # is the number of I/O agents to specify
      - » Next is list of taskids to be I/O agents – here, 2,4,6,8 rank are I/O agents
    - **NOTE:** When MP\_IOTASKLIST is used MP\_IONODEFILE is ignored
    - Please check PE 5.2.1 Operation and Use Manual for further info

# Misc additions

# Misc. Enhancements

- **Openssh support is added with PE 5.2**
  - CTsec method is withdrawn
- **MP\_MSG\_API is extended to support other parallel programming models.**
  - Example: `MP_MSG_API=mpi,upc`
- **POE runtime enhanced to support launching of User Space jobs without using any resource manager**
  - `MP_RESD=poe` & new environment variable: `MP_POE_LAUNCH`
  - Useful in interactive development environment
  - `/etc/poe.limits` can be modified by System Admin to control
  - `MP_POE_LAUNCH={ip, us, all, none (disables interactive job)}`
- **PE 5.2 includes improvements in the ability to run POE in large scale environments.**
  - POE/PMD are now 64bit binaries
  - `MP_DEBUG_ATTACH={no, yes (default)}`
  - `MP_DEBUG_ATTACH_DIR={/tmp (default), user specified dir}`
  - `MP_INFOLEVEL` debug output is reduced
- **PE 5.2 defines and utilizes a set of common resource management interfaces (Provides interface header file), for use by LoadLeveler and other resource managers.**
- **With PE 5.2, the PE CD now contains LL RM install package to enable third party schedulers to code to new LL RM APIs.**
  - No additional licensing cost to use LL RM with PE
- **`nrt_clean` & `nrt_status` commands are now shipped with with LAPI**

# New Early ship functions

# Parallel Environment PE 5.2.1

- Enhanced OS Jitter co-scheduling function to exploit enhance AIX scheduling hooks
  - AIX 6.1 only, looking for customer willing to work with HPC Performance team to help configure and setup
  - Help minimize OS Jitter

# Parallel Environment Long Range Possibilities

The following slides represent a mix of ideas, prototyping efforts and optimistic looks to the future – None of these are certain

# Huge application Scale Improved User Productivity

- IBM and the Poughkeepsie Lab are committed to two publicly announced Petascale computing programs – These programs dominate our longer range Parallel Environment efforts
  - The Defense Advanced Research Projects (DARPA) program for High Productivity Computing Systems (HPCS)
  - NCSA Blue Waters Petascale Computing System

# Scaling

- The Parallel Environment (POE, MPI, LAPI & Tools) teams are investigating to support hundreds of thousands of tasks
  - smaller, tighter data structures ✓
  - Retain performance at 100s of tasks while making 100s of thousands possible ✓ - ongoing
  - Improved Collective Communications scaling
  - Revised early arrival buffer management ✓
  - Better MPI-IO strategies ✓ - need additional user input
  - Robustness at scale ✓ - ongoing
  - OS Jitter control strategies ✓ - ongoing

# Alternate Programming Models

- MPI Programming is difficult. Many HPC communities are seeking more intuitive ways to exploit parallelism
  - PGAS (Partitioned Global Address space models)
    - Unified Parallel C
    - CoArray Fortran
  - Shared memory models
    - Openshmem (SHMEM™)
  - Potential new programming model from IBM Research
    - Part of DARPA High Productivity Computing Languages effort

# Options for Productivity Tools

- Infrastructures for debugging at huge scale
  - Collaboration with NCSA ✓ - contributed SCI framework to Eclipse PTP project
- Leveraging of Eclipse Parallel Tools Platform
  - An MPI code development assistant ✓
- Watson Research HPC Toolkit for Performance analysis/tuning
  - Part of PE 5.1 ✓
- Implementing MPI 2.0 Process management within pre-defined resources ✓ - Part of PE 5.2

# MPI Forum

- IBM and the MPI team are working with the MPI Forum on MPI 2.1, 2.2 and 3.0
- MPI 2.1 and 2.2 have fairly modest goals. ✓
  - MPI 2.1 will offer a single MPI Standard (1.1 and 2.0 merged and errata corrections formalized). The draft is in the formal approvals process now
  - MPI 2.2 content is being defined now & approval target is very early 2009. Modest API extensions.
    - No changes required for MPI applications
    - Low implementation effort – prompt availability predicted

# MPI 3.0

- Major extensions possible (but not certain)
- Reference implementation required
  - This was part of MPI 1 but not part of MPI 2. The lesson has been learned.
- Target for approval – 2010 – **moving target**
- Implementations may deliver MPI 3.0 in stages
- There is a handful of proposal sets today
  - Some may fall away as they are debated
  - The process is open to new proposals now but presumably will close in 2009. - **moving target**

# MPI 3.0 Proposals

- Enhancements to Collective Communication
  - non-blocking collective operations ✓
    - To allow computation/communication overlap
  - Persistent collectives
    - Allow the MPI implementation to invest in negotiating a plan and amortize the cost over many uses
  - Neighbor communication collectives
    - MPI\_Alltoallv argument lists are too big when the pattern is sparse
    - MPI\_Alltoallv algorithms for dense patterns are not well suited to sparse patterns – hard for libmpi to adapt on the fly

# Runtime Enhancements

- Enhance runtime to support workflow or sub jobs and MPMD runs

## Contact Information

Chulho Kim

IBM Poughkeepsie UNIX  
Development Lab

[chulho@us.ibm.com](mailto:chulho@us.ibm.com)