



IBM Advanced Clustering Technology Team

Parallel Environment MPI - Today and Tomorrow

ScicomP 14 - Poughkeepsie

May 22, 2008

Dick Treumann - MPI Team

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.

Recent enhancements in PE

Enhancements: Parallel Environment 4.3.2

- PE now also supports AIX Version 6.1, or later (on stand-alone servers, running in IP mode only). PE supports SLES9 and now SLES10
- PE now provides task affinity support for OpenMP applications running over AIX.
- PE Remote Direct Memory Access (RDMA) function can now be used with the InfiniBand interconnect on both AIX and Linux
- The ability to manage task affinity, which was previously only supported for AIX, has been extended to Linux users.
- Support for Clustered IBM BladeCenter® Model JS21/JS22 servers with Infiniband user space communication. (AIX & Linux)

Parallel Environment Long Range Possibilities

The following slides represent a mix of ideas, prototyping efforts and optimistic gazing into the future

Huge application Scale Improved User Productivity

- In IBM and the Poughkeepsie Lab, we are excited about our involvement in 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 1000s of tasks while making 100s of thousands possible
 - Improved Collective Communications scaling
 - Revised early arrival buffer management
 - Better MPI-IO strategies
 - Robustness at scale
 - OS Jitter control strategies

Alternate Programming Models

- MPI Programming is considered difficult. Many HPC communities are seeking more intuitive ways to exploit parallelism
 - PGAS (Partitioned Global Address space models)
 - Unified Parallel C
 - CoArray Fortran as defined by Fortran 2000 standard
 - Hybrid models
 - OpenMP + MPI
 - Potential new programming model from IBM Research
 - Part of DARPA High Productivity Computing Languages effort

Options for Productivity Tools

Leveraging of Eclipse Parallel Tools Platform

- Infrastructure for debugging at petascale
- An MPI code development assistant
- Watson Research HPC Toolkit for Performance analysis/tuning
- Plug-ins for running PE/LL under Eclipse PTP
- Static analysis of parallel applications

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
 - Forum “hopes” all MPI implementations will comply ASAP
 - MPI 2.2 content is being defined now & approval target is very early 2009. Modest API extensions.
 - No changes required for MPI applications
 - Modest implementation effort – prompt and full availability predicted (should not require release staging)

MPI 3.0

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

MPI 3.0 Proposals

- **Fault Tolerant MPI**
 - Mixed goals in working group but most desire a way for an MPI job to survive the loss of a task. The application would need to adapt, not expect transparent recovery.
- **MPI Application Binary Interface**
 - The ABI is defined per-platform. It would allow an MPI application to use a different MPI implementation without being recompiled. An ABI is attractive to many ISVs.

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

MPI 3.0 Proposals

- New Fortran Bindings
 - use the intrinsic ISO_C_BINDING module in Fortran 2003
 - Resolves F90 interface explosion problem – solid compile time call checking becomes practical
- New Remote Memory Address operations
 - Existing MPI 1sided is complex & often non-intuitive
 - Truly high performance implementation and exploitation of present RMA model has been elusive

MPI 3.0 Proposals

- MPI subsetting – members of working group have various goals. Here are sample arguments but goals need to be pinned down
 - MPI standard's complexity intimidates people
 - Having all of MPI claimed to cause memory footprint or performance problems – maybe subsets can help
 - Implementations that do not have everything have no defined way to state what they do have
 - Make PMPI optional so some MPI routines can be in-lined.

MPI 3.0 Proposals

- Generalized Request enhancements
 - Main focus is to allow the application to specify a progress function which would be run transparently
- Point to Point enhancements
 - Concrete proposals not on record yet.
 - Some suggestions for a dynamic size receive but little general enthusiasm

To observe or join in visit: <http://meetings.mpi-forum.org/>

Contact Information

Richard Treumann

IBM Poughkeepsie UNIX
Development Lab

treumann@us.ibm.com