Open MPI Community Meeting

Jeff Squyres

Rainer Keller

International Supercomputer Conference (ISC) 2007
Dresden, Germany

Overview

• Introduction to Open MPI
• Current status
• Future directions
• Audience feedback
Open MPI Is…

- Open source
  - Started with expertise from 4 MPI implementations
  - Has grown into a full community
- Features of Open MPI:
  - Full MPI-2 implementation
  - Fast, reliable and extensible
  - Production-grade code quality as a base for research
  - BSD license

Why Does Open MPI Exist?

- Maximize all MPI expertise
  - Research / academia
  - Industry
  - …elsewhere
- Capitalize on [literally] years of MPI research and implementation experience
- The sum is greater than the parts
Why Separate From MPICH / MVAPICH?

- Open, inclusive community
- Support for more networks
- Support for many resource managers
- MPICH / MVAPICH have different project goals
  - They both chose to remain separate

Current Membership

- 14 members, 6 contributors
  - 4 US DOE labs
  - 8 universities
  - 7 vendors
  - 1 individual
Sponsors

Current Status

- Stable release version: v1.2.3
- Source code
  - tarballs
  - SRPM
  - Subversion repository
- Binaries available for
  - OpenSuse
  - Mandriva

- Binaries included in
  - RHEL, Fedora, Scientific Linux, ...
  - Debian (just saw posting this past weekend)
  - Gentoo
  - OFED
  - Sun ClusterTools 7
  - OS X Leopard (*)
Current Status

- Networks
  - Shared memory
  - Infiniband:
    - OpenFabrics
    - UDAPL
    - mVAPI (deprecated)
  - InfiniPath
  - Myrinet
    - gm
    - MX
  - Portals
  - TCP

- Resource managers
  - Clustermatic Bproc
  - LoadLeveler
  - PBS / Torque
  - POE
  - rsh/ssh
  - SGE / N1GE
  - SLURM
  - Xgrid
  - LSF (coming soon)

Features

- Plugins: “MCA”
  - Plugins auto-select based on environment
  - Selectable by user/admin

- ISVs may
  - Distribute binary plugins
  - Redistribute Open MPI

- Run-time tunable values
  - MPI layer parameters
  - Per plugin parameters
  - Change behavior of code at run-time
  - Does not require recompiling / re-linking

- Simple example
  - Choose which network to use for MPI communications
Point to Point Architecture

• Now MPI_SEND is fantastically complex!
  ▪ Fragment the message
  ▪ Select which device(s) to use
  ▪ Send each fragment on an available device
  ▪ Be careful with resource usage…etc.

<table>
<thead>
<tr>
<th>MPI-Layer</th>
<th>PML</th>
<th>BML</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenIB</td>
<td>RDMA MPool</td>
<td>MX BTL</td>
</tr>
<tr>
<td>BTL</td>
<td>RDMA MPool</td>
<td>SM BTL</td>
</tr>
<tr>
<td>Rcache</td>
<td>Rcache</td>
<td>Rcache</td>
</tr>
</tbody>
</table>

Configuration

• “Normal” GNU installation
  ```
  shell$ configure && make all install
  ```

• Can easily adapt for your site:
  ▪ Select which plugins to be compiled
  ▪ Build static libraries (including plugins)
  ▪ Deselect optional features (C++/F90 bindings)
  ▪ Enable tracing based on PERUSE
  ▪ …etc.
Open MPI IB DDR Performance

<table>
<thead>
<tr>
<th></th>
<th>µs</th>
<th>MB/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open MPI IB</td>
<td>3.23</td>
<td>1467</td>
</tr>
<tr>
<td>MvApich 0.9.7</td>
<td>3.15</td>
<td>1425</td>
</tr>
<tr>
<td>Open MPI (tcp)</td>
<td>62.6</td>
<td>221</td>
</tr>
</tbody>
</table>

Open MPI-trunk~r14000 (BTL)
ofed-1.1-stack
MvApich-0.9.7
NetPipe-3.6.2
HCA: MT25204
1 mem, 4x, 5Gbps = 20 Gbps, 8x PCIe

Open MPI Myri-10G Performance

<table>
<thead>
<tr>
<th></th>
<th>µs</th>
<th>MB/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPIch-mx</td>
<td>2.62</td>
<td>1055</td>
</tr>
<tr>
<td>Open MPI (BTL mx)</td>
<td>3.34</td>
<td>1053</td>
</tr>
<tr>
<td>Open MPI (MTL mx)</td>
<td>2.83</td>
<td>1055</td>
</tr>
</tbody>
</table>

Open MPI-trunk~r14000 (BTL)
MPIch-MX-1.2.7..1
mx-1.2.0i
NetPipe-3.6.2
NIC: Myri-10GE, 2MB mem, 8x PCIe
Success Stories

- Achieved #6 slot on Sandia Thunderbird
  - 53 tflops
  - November 2006 Top500 list
- Vendor support
  - Sun ClusterTools 7
  - OpenFabrics vendors / OFED
- Integrated in many Linux distros

COMMUNITY

Roadmap

- v1.2 series
  - Current stable version: v1.2.3
  - v1.2.4 is possible (minor bug fixes)
- v1.3 series
  - “Expected” towards end of 2007
  - Difficult to exactly predict timelines with multi-organization open source projects
Possible Upcoming Features

• v1.3 *may possibly* contain:
  ▪ Checkpoint / restart functionality
  ▪ Better mapping of IB HCA ports to processes
  ▪ Add the Portable Linux Processor Affinity (PLPA) support to portably pin processes to specific cores
  ▪ End-to-end data reliability
  ▪ Memory debugging features
  ▪ Symbol visibility, compiler attributes, Fortran fixes

• Something down the road:
  ▪ Windows CCS support
  ▪ More forms of fault tolerance

Valgrind Memory Debugging

• Work by HLRS
• Check of Open MPI memory failures:
  ▪ Parameters passed to MPI
  ▪ Definedness of MPI-internal structures
• Check of application’s MPI-conformance:
  ▪ MPI-buffers passed to MPI_Irecv, …

```c
MPI_Irecv (buffer, ... &req);
buffer[n] = 1;
MPI_Wait (&req, &status);
```
**PERUSE**

- Work by HLRS, U. Tennessee
- Give tools insight to MPI-internal state

Visualization:
Paraver@BSC

- # of fragments/second - congestion (top)
- # of physical concurrent transfers (bottom)

**Checkpoint / Restart**

- Work by Indiana University
- Added much infrastructure to Open MPI
  - Next generation beyond LAM/MPI
  - Generic process and parallel job FT support
  - Foundation for many other forms of fault tolerance
- First: LAM/MPI-like coordinated checkpoint
  - Uses BLCR or “self” plugins
OpenFabrics Features

- Work by OpenFabrics vendors, Livermore
- Better mapping of cores to HCAs (NUMA)
- Better multi-NIC fragment scheduling
- Support for asynchronous events
- Small message aggregation
- RDMA connection manager (iWARP)
- Threaded progress
- Unreliable datagram support (?)

What do You Want From MPI?

(audience -- you talk now)
How Important Is…

• Thread safety
  ▪ Multiple threads making simultaneous MPI calls
• Parallel I/O
  ▪ Working with parallel file systems
• Dynamic processes
  ▪ Spawn, connect / accept
• One-sided operations
  ▪ Put, get, accumulate
• Multi-core operations
  ▪ Fine-grained process affinity
  ▪ Internal host topology awareness

Come Join Us!

http://www.open-mpi.org/