Documentation, Standards, and other Miscellaneous Stuff

Jeff Squyres

Design Documentation

- Started at beginning of project
  - Effort fell off
  - What currently exists is totally outdated
- Starting a new SVN repository
  - "ompi-design"
  - Fill it up over time

Code Documentation

- Using Doxygen (www.doxygen.org)
  - Formatted comments
- OPAL layer is best documented
- ORTE and OMPI layers "loosely" documented
- Strongly encourage all new functionality to have doxygen comments

User Documentation

- User Guide and Installation Guide
  - Skeletons right now (stolen from LAM/MPI)
  - Need to be filled in
- Man pages
  - mpirun.1
  - …but otherwise non-existent
  - Could certainly use some help here
- Hint hint ☺

User Documentation

- Current main source: web FAQ
  - Easily extensible PHP code
  - Every time we see a question twice, put it on the FAQ
  - Google-able
- Heavily use of mailing lists
  - Web-archives, so also Google-able

Communication

Public Mailing Lists

- announce@open-mpi.org
  - Broadcast only
- users@open-mpi.org
  - User-level questions
- devel@open-mpi.org
  - Developer-level questions
- svn[-full]@open-mpi.org
  - SVN activity

Private Mailing Lists

- admin@open-mpi.org
  - Administrative group
- devel-core@open-mpi.org
  - Private developer list

Developer Communication

- Weekly teleconference
  - Tuesdays, 11am US Eastern
- Telephone
  - Lots and lots of telephone calls
- Instant messenger
  - AOL is IM of choice
  - Lots and lots of IM

Developer Communication

- Quarterly face-to-face meetings
  - Location rotates
  - Sometimes entire group
  - Sometime smaller, targeted meetings
- Virtual is only so good
  - Need real meetings to supplement
  - White boards, etc.

Sub-Projects

Portable Linux Processor Affinity

- Linux processor affinity has changed 3x
  - Same function name; different parameters (!)
  - Depends on glibc, kernel, distro
- Can fix it with "./configure"
  - Does not fix binary compatibility (ISV's care)
- PLPA uses syscall()
  - Probe the running kernel
  - Dispatch to the Right variant

PLPA

- Released v1.0.3
- Trivially small interface
- Will eventually add abstraction layer
  - Map to specific core / socket
- Completely unrelated to Open MPI
- Will be integrated into Open MPI v1.2

MPI Testing Tool

- Test any N MPI implementations
  - Each installed M different ways
    - Against T different test suites
      - Run each R different ways
- Multiplicative effect: N x M x T x R
- Fully automated (run via cron)
- Results go into a centralized database
  - Correctness and performance results
  - Available for historical data mining

MTT

- Supports "disconnected" scenarios
  - Download on one node
  - Compile / install on another
  - Run tests on another [cluster]
- Not yet released
  - Hope to be usable in near future
  - Will first make available to Open MPI members for distributed testing
  - Then open to community

Standards and Conventions

"Minimal" Standards

- None of us could agree on a full set
  - …and even if we did, people would ignore it
- So we created a minimum set
  - Some style
  - Some correctness

Style

- 4 space tabs
  - Spaces, not tabs
- Curly braces on first line of the block
  - if (3 < 4) { …
- Preprocessor macros in all upper case
- That's it (for style)

Correctness

- All blocks use curly braces
  - Even one-line blocks
- Constants on the left side of ==
  - if (NULL == foo) { …
- Functions with no arguments are (void)
- No C++-style comments in C code
  - No GCC extensions except in GCC-only code
- No C++ code in libraries
  - Discouraged in components

Correctness

- Always define preprocessor macros
  - Define logicals to 0 or 1 (vs. define or not define)
  - Use "#if FOO", not "#ifdef FOO"
  - Gives compiler assistance for mistakes
- Not possible for some generated macros
  - Autoconf and friends

#include Statements

- System files are in <>
  - Most should be protected with macros

```
#if HAVE_UNISTD_H
#include <unistd.h>
#endif
```

- OMPI files in ""
  - Always use full pathname

```
#include "opal/mca/base.h"
#include "ompi/group/group.h"
```

Header Files

- Always protect with preprocessor macros

```
#ifndef _THIS_HEADER_FILE_NAME_H
#define _THIS_HEADER_FILE_NAME_H
/* …contents of header file… */
#endif
```

- Only access external symbols through their header files
  - Do not "extern" external variables in .c files
  - Do not prototype external functions in .c files

Windows Compatibility

- OMPI_DECLSPEC
  - Used in header files
  - Before any public symbols
  - Adds in Right keywords for MS C compiler
  - Resolves to blank on POSIX systems

```
OMPI_DECLSPEC extern int ompi_op_foo;
OMPI_DECLSPEC int mca_base_open(void);
```

Compiler Warnings

- Must compile without warnings on all platforms, compilers
- Default GCC developer build
  - Maximum pickyness
- Exceptions granted where warnings cannot be avoided
  - VAPI header files have GCC extensions
  - Flex-generated code

Symbols

- Cannot conflict with user code
- Public files and symbols must be prefixed
  - Library symbols use opal_, orte_, or ompi_
  - Components must adhere to the Prefix Rule
  - One exception: MCA component struct (both covered later)
- All private symbols must be “static”

Prefix Rule

- Files and symbols must be unique
  - Cannot conflict with rest of OMPI/ORTE/OPAL (even file names within a single library)
- Frameworks/components prefix all names
  - <section>_<framework>_<component>_*
  - int ompi_btl_mvapi_foo = 3;
  - int ompi_btl_mvapi_bar(void) { return 2; }
  - [ompi_]btl_mvapi_yow.c (ompi_ optional here)

Note: Section Splits

- OPAL, ORTE, and OMPI split into multiple separate source trees somewhat recent
- Most symbols are “correct”
  - ompi_*(), orte_*(), opal_*()
- But some code has not yet been updated
  - Many frameworks and components are still prefixed with “mca_*” instead of ompi_*, orte_*, opal_*

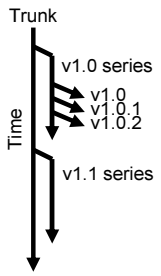
MPI API

- Invoking MPI API is disallowed
  - Required for fault tolerance
  - If need to MPI_SEND, use back-end PML function
- MPI API is usually:
  - Error checking
  - Back-end invocation

→ Show ompi/mpi/c/send.c, bcast.c

Repository

- /trunk is free-for-all
  - Head of development
  - People will yell if you break the trunk
- Release series
  - /branches/v1.0, /branches/v1.1, …
- Stable releases
  - /tags/v1.0, /tags/v1.0.1, …



Repository

- /tmp
  - Free-for-all developer branches
  - Good for short or long-term development that requires breaking things
  - Not open to the public
  - Developers can create / delete whatever they want

Nightly Tests

- Currently testing all 3 active branches:
  - /trunk, /branches/v1.0, /branches/v1.1
- Test what you ship, ship what you test
  - Distribution tarballs made at midnight, US IN
  - Testers download snapshots
  - Do various compile and run tests
  - Send e-mail results
- MTT will be most useful when ready

Version Numbers

- Major.Minor.Release[Qualfier]
- Qualifier
  - aX: alpha
  - bX: beta
  - rcX: release candidate
  - rV: Subversion r number
- Examples
  - 1.2.3a4
  - 4.5.6rc2r9849