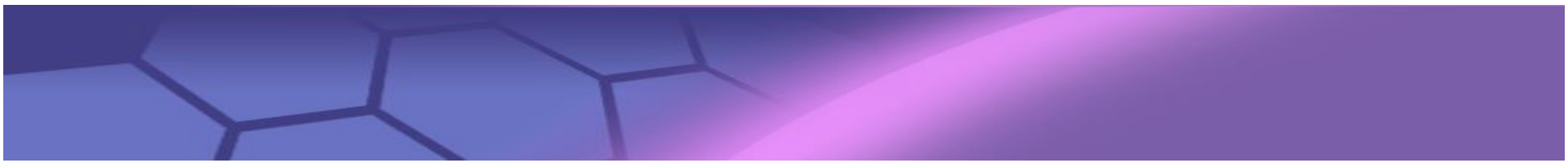
PTP 1.1 Release Review

January, 2007



Agenda

- Features
- Non-Code Aspects
- APIs
- Architectural Issues
- End-of-Life
- Bugzilla
- Standards
- Schedule
- Process
- Community
- IP Issues
- Project Plan

Features

- 1.1 Plan available at:
  - http://wiki.eclipse.org/index.php/PTP/planning/1.1
- Major features:
  - Parallel Language Development Tools. The MPI Language Tools framework has been generalized into a static analysis framework. This will allow other programming models and parallel languages to use the static analysis features.
  - OpenMP Support. Static analysis of OpenMP codes has been added. This provides similar functionality to the MPI support, but includes additional support for detecting certain error conditions in OpenMP pragmas.
  - Debugger Scalability. The debugger has been enhanced to improve scalability on very large parallel machines. It is has been tested on a 1000 process job.

Features (cont…)

- Major features (continued)
  - Debugger UI Performance. Improvements have been made to how variables are fetched and displayed in the UI in order to improve performance.
  - MPICH2 Runtime. Support for the MPICH2 runtime has been added. This allows programs compiled with the MPICH2 runtime to be launched using PTP. Debugging is not supported yet.
- 1.1 release notes available at:
  - http://www.eclipse.org/ptp/docs/releases/release-1.1.php

Non-code Aspects

- User documentation and tutorials
  - http://eclipse.org/ptp/doc.php
  - Using PTP to build, launch and debug an MPI program
  - Using Parallel Language Development Tools
  - Release notes and troubleshooting information
  - *Scientific Application Development using Eclipse and the Parallel Tools Platform*, LACSI 2006 and SC 2006 conferences (will be available for download)
- Design documentation
  - http://wiki.eclipse.org/index.php/PTP/designs
- FAQ
  - http://eclipse.org/ptp/faq.php

Non-code Aspects (cont.)

- Articles
  - *A strategy for Addressing the Needs of Advanced Scientific Computing Using Eclipse as a Parallel Tools Platform* (link from PTP web page)
  - *Developing Scientific Applications Using Eclipse* (appears in July 2006 edition of Computing in Science and Engineering)
  - *A Model-Based Framework for the Integration of Parallel Tools* (to appear in Proceedings of the 2006 IEEE International Conference on Cluster Computing, Sept. 2006)
  - *Weave parallel applications with the Eclipse Parallel Tools Platform* (IBM developerWorks article)

APIs

- Runtime and debugger APIs are still evolving
  - The runtime API for the 1.0 and 1.1 releases are stable
  - The 2.0 release will have significant, breaking, changes, since we're currently implementing considerable new functionality
  - The debug API is also relatively stable, though there were some minor changes between 1.0 and 1.1
  - There should be no API users who are unaware of the changes
- Plugin version numbers have been incremented from 1.0.0 to 1.1.0
- Internal APIs have been marked as such
- Much redundancy and unused code has been removed

Architectural Issues

- The most difficult part is the need to support a large range of machine architectures, operating systems and parallel runtime systems
  - This will be handled by automake/autoconf
  - Next release will be more complicated because target machines will be remote
- Dynamically linked executables have been depreciated
  - Non-java code is now always built from source
  - Build scripts have been improved so this only requires one command

End-of-Life

- No end-of-life issues in this release

Defect Statistics

- 115 bugs opened
- 98 bugs resolved/closed
- 7 bugs deferred
- 17 new bugs
- 0 P1 and P2 bugs outstanding

Standards

- Provides support for MPI-1.1, MPI-2.0 and OpenMP standards
- Fortran 95 support
- Debugger utilizes MI (gdb Machine Interface) for communication to backend debuggers

Schedule

- Original schedule was delayed a month due to:
  - Limited resources
  - Complicated test environment
  - Developers other commitments
- Scheduled release was end of Q4 2006, will now be Q1 2007
- Lessons:
  - Don’t have project leaders’ email blacklisted by eclipse.org
  - Don’t schedule release close to Christmas!

Process

- This release has been developed using open, transparent, and inclusive processes
- This release has followed its charter principles
- The PTP project makes appropriate use of
  - Bugzilla
  - Mailing lists (ptp-dev@eclipse.org)
  - Newsgroups (eclipse.technology.ptp)
  - Regular, monthly, conference calls
  - Wiki (http://wiki.eclipse.org/index.php/Parallel_Tools_Platform)
- There were no commiter elections/removals during the 1.1 release cycle

Community

- Increasing number of users
  - Bugzilla and emails
- Increasing contributions
  - Monthly developer meetings regularly have ~10 participants
- Outreach activities:
  - Tutorial at LACSI 2006 and SC 2006
  - Exhibition booth at SC 2006
  - Workshop presentations
  - Face-to-face meetings

IP Issues

- All plugins contain appropriate about and license files
- All contributions (code, documentation, images, etc) have been committed by individuals who are either Members of the Foundation, or have signed the appropriate Committer Agreement; in either case,these are individuals who have signed, and are abiding by, the Eclipse IP Policy
- Major contributions have followed due diligence process
- All non-committer code contributions, including third-party libraries, have been documented in the release and reviewed by the Foundation's legal staff
- All contribution questionnaires have been completed

Project Plan

- Draft 2.0 plan available at:
  - http://wiki.eclipse.org/index.php/PTP/planning/2.0
- Planned features:
  - Resource Manager Support. This will integrate the parallel model and parallel views with resource management systems. Users will be able to view and query the status of queues, and submit jobs through resource managers.
  - Remote Services. This will allow projects to reside on remote systems, and allow PTP to communicate with remote resource managers for building, launching and debugging applications on remote machines.
  - Parallel Debugger. Improvements to the scalability and functionality of the parallel debugger. New visualization features such as multi-variable and array viewers.