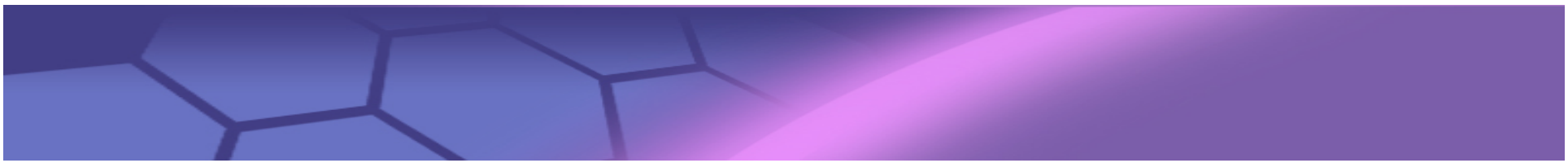
Parallel Tools Platform Project

August 15, 2006



Introduction

- Major project milestones
  - Proposed November 2004
  - Created February 2005
  - Fortran merge with Photran project April 2005
  - IBM contribution November 2005
  - Oregon contribution May 2006
  - Stable build March 2006
- Continuing to expand community
  - Contributions from LANL, IBM, Monash, and Oregon
  - Participation from Absoft, Cray, UBC, TUM
  - Terra Soft Solutions plan to bundle with their HPC release

PTP 1.0 Features

- 1.0 Plan available at: http://www.eclipse.org/ptp/docs/plan.html
- Parallel Runtime
  - Support for controlling and monitoring machines and jobs via the Open Runtime Environment (OpenRTE).
  - Views for dynamically displaying machine/node status, job status and process detail information.
- Parallel Launch
  - Provides the ability to specify parallel executable, debugger and number of processes for a parallel job launch.

PTP 1.0 Features (cont.)

- Parallel Debugger
  - Extends the CDT debug model to provide support for parallel debugging.
  - New user interface components are available to support the extended model and allow control of multiple processes simultaneously.
  - A high level debug interface has been designed to allow communication to arbitrary back-end debuggers.
  - A prototype back-end debugger implementation has been provided.

PTP 1.0 Features (cont.)

- MPI Development Tools
  - Static analysis tool specifically for MPI programming. Provides MPI artifact annotation in the editor, an artifact view for easy navigation, content assist and context sensitive help.
  - MPI artifacts are extracted automatically from the MPI header files. Extensive help information has been provided.
- Fortran Development Tools
  - Extends CDT to support Fortran programs, including a Fortran editor, tool chain support and error parsers for Fortran compilers. Also adds support for mixed-language projects (Fortran and C/C++).
  - This was a preliminary implementation that has now been merged with the Photran project.

Non-code Aspects

- User documentation and tutorials
  - http://eclipse.org/ptp/doc.php
  - Using PTP to build, launch and debug an MPI program
  - Using MPI Development Tools
  - Release notes and troubleshooting information
- Design documentation
  - http://wiki.eclipse.org/index.php/PTP/designs
- FAQ
  - http://eclipse.org/ptp/faq.php
- Website has transitioned to Phoenix

Non-code Aspects (cont.)

- Whitepapers/Publications
  - *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)

Non-code Aspects (cont.)

- Press
  - *Parallel systems development eyed by Eclipse,* InfoWorld, March 2005
  - *Parallel Tools Platform,* Eclipse Review, April 2005
  - *LANL, Eclipse Intro Parallel Tools Platform,* HPC wire, April 2005
  - *Eclipse For Clusters*, ClusterWorld, April 2005
  - *Eclipse focuses on tools for parallel systems development*, InfoWorld, April 2005
  - *Eclipse Project Takes on Parallel Computing*, Application Development Trends, April 2005

Community Feedback

- Support for additional runtimes (e.g. MPICH)
- Support for resource managers
- Ensuring PTP is scalable to large systems (particularly UI)
- Improving debugger functionality

API: 1.0 Status

- Runtime API
  - 3 clients in PTP
  - Javadoc
  - Test cases
- Debugger API
  - 2 clients in PTP
  - Javadoc
  - Test cases
- No previous releases, so no compatibility issues

Tool Usability

- Minimize configuration
  - Once architecture specific fragment is installed, plugin will autoconfigure as much as possible
- Delay startup until needed
  - Since loading initial model is time consuming, it will only be done when actually needed - when selecting a PTP perspective or when a parallel job is launched
- Real-time feedback
  - In order to give developer a sense of interaction with the parallel system
- Emphasize integration
  - Switching between tools should be as easy as e.g. debugging in CDT

Architectural Issues

- Architecture Support
  - The most difficult part is the need to support a large range of machine architectures, operating systems and parallel runtime systems.

	Arch		*O/S*		*Runtime*		*Resource Manager*	
V1.0	x86 x86_64 ppc	X	Linux MacOSX	X	OpenRTE			= 6
V2.0	x86 x86_64 ppc	X	Linux MacOSX Windows	X	OpenRTE MPICH2	X	LSF SLURM LoadLeveler	= 54

Architectural Issues (cont.)

- Dynamically linked executables
  - Since we have no control over runtime version installed, must ship dynamically linked executables.
  - Not a perfect solution, since library version changes may be too great.
    - E.g. OMPI 1.0.2 on RH9 vs OMPI 1.0.2 on FC4
  - In this case, even with the same version of OMPI, changes to libc may still cause problems
  - Only choice is to ship executables for every major O/S version, or force installer to build from source
  - Neither are ideal

Defect Statistics

- As of August 2006:
  - 12 NEW
  - 4 ASSIGNED
  - 1 REOPENED
  - 13 RESOLVED
  - 43 VERIFIED
  - 16 CLOSED
- Release exit criteria
  - 0 high severity defects
  - 100% test pass (as per PTP 1.0 Release Test Plan)

Standards

- Provides support for MPI-1.1 and MPI-2.0 standards
- Fortran 95 support
- Debugger utilizes MI (gdb Machine Interface) for communication to backend debuggers
- There are not many standards in parallel computing!
  - which is one of the major problems PTP is trying to address

Committers and Contributors

- 7 committers from 4 organizations
- Direct contributions from 2 other organizations
- Emphasis on growing the project
- All design documents available on the wiki
  - Design discussions are held via mailing list and phone calls
- Monthly development calls
  - Opportunity to review status
  - Developer/design discussions
- Release planning
  - Release documents maintained on wiki
  - Conference calls used to refine releases
- Face-to-face meetings at EclipseCon and SuperComputing

Community

- Slow uptake expected
  - Community skeptical of IDEs, very easy to turn off
  - Require solid, stable implementation
  - Require stable Fortran support
  - Beginning to see some user activity
- Download activity
  - ~1400 downloads in last 30 days
- Outreach activity
  - Talks at EclipseCon 2005 & 2006
  - Tutorial at LACSI 2005, planned for LACSI 2006 and SC 2006
  - Publications in CiSE Magazine, IBM developerWorks, and Cluster 2006 conference
  - Face-to-face meetings

Intellectual Property

- All contributions to date have been made under EPL 1.0
- A small amount of GPL code that was uploaded to CVS during initial import has been removed and replaced with code re-written from scratch and licensed under EPL 1.0
- All plugins contain appropriate license files
- All committers have completed Eclipse Committer Agreements and have been approved by the PMC
- Major contributions have followed due diligence process
- Project log available at
  - http://dev.eclipse.org/viewcvs/index.cgi/%7Echeckout%7E/org.eclipse.ptp/doc/project_log.csv?cvsroot=Technology_Project

Future Plans

- PTP 1.1 Release
  - Release/feature planning underway
  - Future major releases to align with Europa
- Closely aligned with CDT and Photran projects
  - Possible move to Tools project
- Working with DSDP/TM project
- New language tools
  - Support for OpenMP
  - Parallel checking and reporting tools
- Additional runtime/debugging support (MPICH2)
- Resource manager integration
- Remote project/build/launch/debug
- Parallel debugger enhancements (many)
- Integration with parallel performance tools
  - TAU, PE, …
- Support for parallel languages
  - UPC, Co-array Fortran, others …