Parallel Tools Platform Update
A Development Environment for Scientific Applications

Greg Watson
IBM
Application Development in HPC

- Developing scalable applications becoming more difficult
  - Systems are becoming more complex
  - Applications are becoming more complex

- Tools are not keeping pace with main stream development technologies
  - Difficult to learn
  - Inconsistent interfaces
  - Low productivity
  - Expensive to make
  - Slow to exploit new technology (e.g. refactoring)
Petascale Challenges

- Must be able to exploit ultra scalability
  - Maximize application performance
  - Minimize time to develop and port applications
  - Lower entry bar for new HPC developers
  - Maximize developer productivity

- Requires coordinated effort
  - Better languages
  - Better Libraries
  - Better Tools

- Need to avoid same development problems
Parallel Tools Platform Goals

- Collaborative Ecosystem Based on Eclipse
  - Parallel Tools Platform
  - Integrated Tools

- Multi Language Support
  - C, C++, UPC, Fortran, X10

- Platform Independence

- Highly Scalable

- Enhanced Productivity
  - Lower Entry Point for New Users
Introduction

What is Eclipse?

- Cross-platform open source framework for highly integrated state-of-the-art development tools
- Integrated tools include:
  - Project management
  - Advanced editing
  - Automated build system
  - Revision control (CVS, SVN)
  - Visual debugging
  - Much more…
- Designed to be robust, scalable, extensible, commercial quality
- Available for Linux, Unix and Windows
- Multi-language support for Java, C, C++, Fortran, Python, Perl, PHP, and others
Parallel Tools Platform

Enabling Parallel Application Development

- Best practice tools for experienced parallel programmers
- Tools to assist new breed of programmers to develop parallel programs
- Leverage Eclipse ecosystem and community for development and support
- Improve parallel tools and the productivity of tool developers
- Provide focal point for parallel tool development for a broad range of architectures
PTP Application Development Cycle

Coding & Static Analysis

Dynamic & Performance Analysis

Application Execution

Application Debugging
Coding & Static Analysis

- **Eclipse**
  - Advanced editing support, content assist, context sensitive help, searching, etc.
- **C/C++ Development Tools (CDT)**
  - Simple refactorings (e.g. extract constant, extract function, extract local variable, hide method)
- **Fortran Development Tools (Photran)**
  - Many complex refactorings (e.g. replace old-style Do-Loops, safe delete internal subprogram, permute subroutine arguments, tile loops, fuse loops)
- **Parallel Language Development Tools (PLDT)**
  - MPI Content Assist
  - Locate MPI artifacts
  - MPI barrier deadlock detection
  - OpenMP concurrency analysis
  - OpenMP common coding problems
Application Execution

Launching & Monitoring

- Improves visibility into target system
- Single point of interface for launching and control
- Manages interaction with different runtime systems and job schedulers
Application Debugging

Parallel Debugger

- Mid-scale integrated debugger
- Tightly integrated with Eclipse
- Supports debugging multiple jobs simultaneously
- Utilizes backend debugger (e.g. gdb) for low level operations
- Targeted at SPMD programming models
- Supports mixed MPI & thread debugging
- Single process and group operations
- Platform for building new debugging paradigms
- IBM Parallel Debugger
- Scalable up to 1M tasks
Dynamic & Performance Analysis

- Perform analysis on the running application using external tools
- Generate results that must be brought back into Eclipse as part of the development workflow
- May require external tool for visualization or other purposes
- TAU, GEM, PPW
- IBM HPC Toolkit
PTP Roadmap

PTP 5.0 is part of the Eclipse “Indigo” simultaneous release of 41 Eclipse projects
New Features

- Remote Project Synchronization
- Configurable Resource Manager
- Scalable System Monitoring
- PE Developer Edition
Remote Project Synchronization (IBM/ORNL)

- Eclipse runs on local workstation, application source resides both locally and remotely
- Pluggable synchronization adapters
  - Initial implementation uses GIT
- All Eclipse features available using local source copy
  - Allows off-line development
- Source synchronized with remote system
  - Configurable; on save and prior to build
- Builds are performed on remote system
  - Multiple build hosts supported
- Launching and debugging on remote system
  - Possibly different from the build system
Remote Project Synchronization
Configurable Resource Manager (IBM/NCSA)

- Interface to remote batch submission or runtime system
- Completely specified by XML definition file
- No coding necessary to add support for new system
- XML definition files can be importing into workspace and edited like normal Eclipse projects
- Implementation available for PBS
- Working on:
  - LoadLeveler
  - Parallel Environment
  - Open MPI/MPICH2
Configurable Resource Manager

Create, manage, and run configurations

Create a configuration to launch a parallel application in Parallel Perspective

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job Name</td>
<td></td>
<td>The name assigned to the job by the qsub or qsh command.</td>
</tr>
<tr>
<td>Queue</td>
<td>normal</td>
<td>Designation of the queue to which to submit the job.</td>
</tr>
<tr>
<td>Number of nodes</td>
<td></td>
<td>Number and/or type of nodes to be reserved for exclusive use by the job.</td>
</tr>
<tr>
<td>Total Memory Needed</td>
<td></td>
<td>Maximum amount of memory used by all concurrent processes in the job.</td>
</tr>
<tr>
<td>Wallclock Time</td>
<td></td>
<td>Maximum amount of real time during which the job can be in the running state.</td>
</tr>
<tr>
<td>MPI Command</td>
<td>mpirun</td>
<td>Which mpirun command to use.</td>
</tr>
<tr>
<td>MPI Number of Cores</td>
<td>100</td>
<td>the &quot;-np&quot; value.</td>
</tr>
</tbody>
</table>

Remote Output Path: Enable read upon job termination
Remote Error Path: Enable read upon job termination
Scalable Batch System Monitoring (IBM/Jülich)

- Based on Jülich’s LLview system
- Scalable monitoring of virtually any size system
- Currently monitors job status, and limited system status
  - Could be extended to provide additional system monitoring
- Supports PBS and LL
  - Being extended to other job schedulers/runtime systems
- Two different polling options
  - Per user; simple to install, potentially heavy load on system
  - Per system; requires sys admin to install, minimal system load
PE Developer Edition

- PTP 5.0 (4.0) depending on licensing issues
- Desktop support:
  - Windows XP & 7
  - Mac OS X 10.6
  - RHEL & SLES 11
- Support for PE 5.1, 5.2, PE Runtime Edition 1.1
- Includes HPC Toolkit
- Future versions IBM Parallel Debugger
PTP Committers

- Tu Hong Jun, IBM
- Alan Humphrey, Utah
- Jie Jiang, NUDT
- Claudia Knobloch, JSC
- Vivian Kong, IBM
- Dieter Krachtus, UTK
- Mike Kucera, IBM
- John Liu, IBM
- Chris Recoskie, IBM

- Randy Roberts, LANL
- Albert L. Rossi, NCSA
- Roland Schulz, ORNL
- Wyatt Spear, Oregon
- Beth Tibbitts, IBM
- Greg Watson, IBM
- David Wootton, IBM
- Wolfgang Frings, JSC (pending)
Conclusion

- **Complexity of peta-scale application development is becoming clearer to developers**
  - IDEs are starting to be seen as key to improving HPC developer productivity
  - IDEs are becoming more accepted in the HPC community

- **Eclipse is now a very mature platform**
  - Has more users than VS
  - Hundreds of commercial products based on Eclipse

- **PTP is the only available platform on which to build an HPC development environment**
  - PTP developer community is starting to reach critical mass
  - Beginning to see a diverse user base forming

- **PTP will provide a robust, extensible, and scalable platform for peta-scale application development**
Online Information

- **Information about PTP**
  - Main web site for downloads, documentation, etc.
    - [http://eclipse.org/ptp](http://eclipse.org/ptp)
  - Developers’ wiki for designs, planning, meetings, etc.
  - Articles and other documents
    - [http://wiki.eclipse.org/PTP/articles](http://wiki.eclipse.org/PTP/articles)
Thank You