Improving the Eclipse Parallel Tools Platform to Create an Effective Workbench for High Performance Computing

Jay Alameda
National Center for Supercomputing Applications
1st CHANGES Workshop, Jülich
5 September 2012
Portions of this material are supported by or based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under its Agreement No. HR0011-07-9-0002, the United States Department of Energy under Contract No. DE-FG02-06ER25752, the Blue Waters sustained petascale computing project, which is supported by the National Science Foundation under award number OCI 07-25070, and the SI2-SSI Productive and Accessible Development Workbench for HPC Applications, which is supported by the National Science Foundation under award number OCI 1047956.

The SI2-SSI team is lead by Jay Alameda (NCSA), Greg Watson (IBM), Steven Brandt (LSU), Marc Snir (U Illinois), and Allen Malony (U Oregon). Team members and senior personnel include Beth Tibbitts (IBM), Ralph Johnson (U Illinois), Albert Rossi (NCSA), Rick Kufrin (NCSA), Sameer Shende (U Oregon), Wyatt Spear (U Oregon), Bety Rodriguez-Milla (LSU), Brian Jewett (U Illinois), Galen Arnold (NCSA), and Rui Liu (NCSA).
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - Improvements to Eclipse PTP
- Software Engineering Practices Enabled by Eclipse PTP
  - Code visibility
  - Multi-system build management
  - Performance tuning
  - Source code control
  - Issue Tracking
  - Documentation
- Eclipse PTP Resources
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - Improvements to Eclipse PTP
- Software Engineering Practices Enabled by Eclipse PTP
  - Code visibility
  - Multi-system build management
  - Performance tuning
  - Source code control
  - Issue Tracking
  - Documentation
- Eclipse PTP Resources
What is Eclipse?

- A vendor-neutral open-source workbench for multi-language development
- A extensible platform for tool integration
- Plug-in based framework to create, integrate and utilize software tools
Eclipse Parallel Tools Platform (PTP)

Coding & Analysis

Performance Tuning

Launching & Monitoring

Debugging
Parallel Tools Platform (PTP)

- The Parallel Tools Platform aims to provide a highly integrated environment specifically designed for parallel application development.
- Features include:
  - An integrated development environment (IDE) that supports a wide range of parallel architectures and runtime systems.
  - A scalable parallel debugger.
  - Parallel programming tools (MPI, OpenMP, UPC, etc.).
  - Support for the integration of parallel tools.
  - An environment that simplifies the end-user interaction with parallel systems.
- [http://www.eclipse.org/ptp](http://www.eclipse.org/ptp)
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - Improvements to Eclipse PTP
- Software Engineering Practices Enabled by Eclipse PTP
  - Code visibility
  - Multi-system build management
  - Performance tuning
  - Source code control
  - Issue Tracking
  - Documentation
- Eclipse PTP Resources
Why WHPC?

- Stable, portable platform for tool development
  - Focus on tool functionality, manage rapid evolution of HPC platforms
  - Encourage consistent tool look and feel
  - Support for HPC application development practices
    - Edit, build, test, debug, maintain, for maximum developer productivity
    - Remote development, batch execution mandatory
  - Track, store, search, browse code artifact provenance
  - Share tool functionality through an integration framework
  - Maintain tool identity
    - Provides for independent tool development pathways and funding
Why Parallel Tools Platform?

- High potential to meet needs of a WHPC.
- Target next generation of HPC developers growing up with IDEs (Eclipse, Visual Studio, ...)
- For PTP to become a WHPC need to:
  - Cultivate community of users
  - Make substantial improvements to PTP around two themes:
    - Improving usability
    - Improving productivity
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - Improvements to Eclipse PTP
- Software Engineering Practices Enabled by Eclipse PTP
  - Code visibility
  - Multi-system build management
  - Performance tuning
  - Source code control
  - Issue Tracking
  - Documentation
- Eclipse PTP Resources
Requirements and Impact

- **Application-centric approach**
  - Use real application codes, with PTP, on production computational resources
    - Identify specific goals to accomplish with each application
    - Use Eclipse PTP to accomplish the goals
    - Identify shortcomings in Eclipse PTP that need to be rectified for Eclipse PTP to be effective with that application workplan
  - This is part of our project team’s responsibility
  - Work with application community and learn from their experience with Eclipse PTP
Requirements and Impact (2)

- **Application-centric approach**
  - Work with application community and learn from their experience with Eclipse PTP
  - Bridge to TeraGrid and (now) XSEDE Advanced User Support
  - Work with targeted organizations to assist with adoption of PTP
  - Monthly user calls
  - Annual user group meeting
  - Hands on tutorials
  - Conference Birds of a Feather
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - Improvements to Eclipse PTP
- Software Engineering Practices Enabled by Eclipse PTP
  - Code visibility
  - Multi-system build management
  - Performance tuning
  - Source code control
  - Issue Tracking
  - Documentation
- Eclipse PTP Resources
Improvements

- Work within Eclipse release cycle
  - Major (API-breaking) improvements with coordinated June release
    - Last major release Eclipse 4.2 “Juno” released June 27, 2012
  - Minor enhancements and bug-fixes with two coordinated service releases in September and February
    - Eclipse 4.2 SR1 due out Sept 26, 2012

- Foci of improvements
  - Improve usability
  - Improve productivity
Improve Usability

- Remote support and scalability enhancements
  - Broaden support of remote capabilities to full PTP
  - Provide for easy platform configuration management
  - Provide additional remote features
    - Automatic remote service deployment
    - Multiple authentication mechanism
    - Support wide range of resource managers
    - Full remote debug support
Improve Usability

- Integration with other tools
  - Improve External Tools Framework (ETFw)
    - Full remote support
    - Integration of tool output with Eclipse views
- Improve and broaden parallel paradigm support
  - Driven by user needs and feedback
Improve Productivity

- Provide support for performance driven refactoring
- Track source and executable code provenance
Significant Recent Improvements

- User-configurable machine configuration
- Wide variety of configurations now available:
- Documentation, tutorial at
Scalable System Monitoring

- System view
- Jobs running on system
- Active jobs
- Inactive jobs
- Messages
- Console

Running an Application
Synchronized Projects

Projects types can be:
- File Service
- Index Service
- Launch Service
- Build Service
- Debug Service

Local source code

Source code copy

Edit

Search/Index Navigation

Run

Debug

Build

Synchronize

Local

Remote

Sync-20
Blue Waters Enhancements

- **Blue Waters**: Cray XE6/XK7 at NCSA
- PTP did not work with Crays “out of the box”
  - Could not submit jobs with appropriate aprun options
  - Could not monitor status of compute nodes
  - Could not set environment modules for build
  - Did not recognize Cray, PGI compilers’ errors messages
  - Did not support OpenACC
  - ...
- Less than 6 months to fix these for PTP 6.0 (!)
Integrated OpenACC documentation and PLDT support (added for BW)

Documentation also available for MPI, OpenMP

**OpenACC™ parallel directive**

Delineates a block of code that will be executed on an accelerator device.

```
!$acc parallel [clause [, clause ...]]
block
 !$acc end parallel
```

```
#pragma acc parallel [clause [, clause ...]]
block
```

Supported clauses are `if`, `async`, `num_gangs`, `num_workers`, `vector_length`, `reduction`, `copy`, `copyin`, `copyout`, `create`, `present`, `present_or_copy`, `present_or_copyin`, `present_or_copyout`, `present_or_create`, `deviceptr`, `private`, `firstprivate`. 
Code completion for OpenACC directives (added for BW)
After the build, compiler errors, warnings, and loopmark information are shown in the Problems view and source code editor.

(Cray, PGI support added for BW)
Graphical interface for launching a job (customized for BW)

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total MPI Tasks:</td>
<td>32</td>
<td>Each XE6 node has two AMD Interlagos CPUs for a total of 32 integer cores and 16 floating point units per node. Therefore, the product of the number of MPI tasks per node and the number of OpenMP threads per task must be less than or equal to 32 (or 16 if running in single-stream mode). The number of MPI tasks per node must not exceed the total number of MPI tasks.</td>
</tr>
<tr>
<td>MPI Tasks per Node:</td>
<td>32</td>
<td>XE6 nodes are normally run in “dual-stream mode,” where every integer core is allocated one task (i.e., one MPI task or one OpenMP thread). However, this means that every two tasks share a floating point unit. Some floating-point-intensive computations may need to run in “single-stream mode,” where every other integer core is idle but every task has exclusive access to a floating point unit.</td>
</tr>
<tr>
<td>OpenMP Threads per Process:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run in Dual-Stream Mode:</td>
<td>✓</td>
<td>The name assigned to the job by the qsub or qalter command.</td>
</tr>
<tr>
<td>Job Name:</td>
<td>ptp_job</td>
<td>Account to which to charge this job.</td>
</tr>
<tr>
<td>Account:</td>
<td></td>
<td>Designation of the queue to which to submit the job.</td>
</tr>
<tr>
<td>Queue:</td>
<td></td>
<td>Maximum amount of memory used by all concurrent processes in the job.</td>
</tr>
<tr>
<td>Total Memory Needed:</td>
<td></td>
<td>Wallclock Time: 00:30:00 Maximum amount of real time during which the job can be in the running state.</td>
</tr>
<tr>
<td>When to Send E-mail:</td>
<td></td>
<td>Whether email should be sent.</td>
</tr>
<tr>
<td>Send E-mail:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Additional Plug-ins from NCSA

- NCSA publishes additional plug-ins can be added onto an existing PTP installation
  - [http://forecaster.ncsa.uiuc.edu/help/index.jsp](http://forecaster.ncsa.uiuc.edu/help/index.jsp)
- Contribute a **System** menu to the menu bar with XSEDE- and NCSA-specific commands
System Menu

- Open Web content in Eclipse:
  - Open XSEDE User Portal
  - Open User Guide for a machine
  - Open an SSH terminal (as an Eclipse view)

Eclipse-integrated SSH terminals are provided by the Remote System Explorer (RSE), one of the features that is included in the Eclipse for Parallel Application Developers package.
System Menu

Shortcuts for common PTP tasks:

- **Add Remote Environment** adds a Remote Tools connection for a particular machine.
- **Add System Monitor** opens the System Monitoring perspective and begins monitoring a particular machine.
System Menu

- The plug-in is preconfigured with information about XSEDE and NCSA resources
- The bottom four commands generally prompt for a system
- **Select System** can be used to eliminate this prompt, so these commands always act on a particular system
MyProxy Logon

- **MyProxy Logon** allows you to authenticate with a MyProxy server
  - Often myproxy.teragrid.org
  - It stores a “credential,” which is usually valid for 12 hours
  - During these 12 hours, SSH connections to XSEDE resources will not require a password; they can use the stored credential
  - However, you **must** enter the correct username for that machine!
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - Improvements to Eclipse PTP
- Software Engineering Practices Enabled by Eclipse PTP
  - Code visibility
  - Multi-system build management
  - Performance tuning
  - Source code control
  - Issue Tracking
  - Documentation
- Eclipse PTP Resources
Software Engineering

። Code Visibility
Software Engineering

✦ Code Visibility

Code navigation
Software Engineering

✨ Code Visibility

- Code navigation
- Syntax-aware editing (navigate to program units and declarations)
Software Engineering

Code Visibility

- Code navigation
- Syntax-aware editing (navigate to program units and declarations)

Code Outline
Would like to understand call hierarchy of this code in relation to "main()" in startup.c.
Software Engineering: Call Hierarchy (C/C++)

- After selecting main, right click and select <Open Call Hierarchy>
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - Improvements to Eclipse PTP
- Software Engineering Practices Enabled by Eclipse PTP
  - Code visibility
  - Multi-system build management
  - Performance tuning
  - Source code control
  - Issue Tracking
  - Documentation
- Eclipse PTP Resources
Multi-machine build management

- **Local**
  - Source is located on local machine, builds happen locally

- **Synchronized**
  - Source is local, then synchronized with remote machine(s)
  - Building and launching happens remotely (can also happen locally)

- **Remote**
  - Source is located on remote machine(s), build and launch takes place on remote machine(s)
Synchronized Projects

Projects types can be:

- File Service
- Index Service
- Launch Service
- Build Service
- Debug Service

Local source code

Source code copy

Compute

Executable

Edit

Search/Index Navigation

Synchronize

Local

Remote

parallel tools platform
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - Improvements to Eclipse PTP
- Software Engineering Practices Enabled by Eclipse PTP
  - Code visibility
  - Multi-system build management
  - Performance tuning
  - Source code control
  - Issue Tracking
  - Documentation
- Eclipse PTP Resources
Performance Tuning: PTP TAU plug-ins

http://www.cs.uoregon.edu/research/tau

- TAU (Tuning and Analysis Utilities)
- First implementation of External Tools Framework (ETFw)
- Eclipse plug-ins wrap TAU functions, make them available from Eclipse
- Full GUI support for the TAU command line interface
- Performance analysis integrated with development environment
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - Improvements to Eclipse PTP
- Software Engineering Practices Enabled by Eclipse PTP
  - Code visibility
  - Multi-system build management
  - Performance tuning
  - Source code control
  - Issue Tracking
  - Documentation
- Eclipse PTP Resources
Source Code Control: “Team” Features

- Eclipse supports integration with multiple version control systems (VCS)
  - CVS, SVN, Git, and others
  - Collectively known as “Team” services
- Many features are common across VCS
  - Compare/merge
  - History
  - Check-in/check-out
- Some differences
  - Version numbers
  - Branching
CVS Features

- Shows version numbers next to each resource
- Marks resources that have changed
  - Can also change color (preference option)
- Context menu for Team operations
- Compare to latest, another branch, or history
- Synchronize whole project (or any selected resources)
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - Improvements to Eclipse PTP
- Software Engineering Practices Enabled by Eclipse PTP
  - Code visibility
  - Multi-system build management
  - Performance tuning
  - Source code control
  - Issue Tracking
  - Documentation
- Eclipse PTP Resources
Issue Tracking

- Mylyn Bridge
  - Tracks tasks, links to source and bug repositories
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - Improvements to Eclipse PTP
- Software Engineering Practices Enabled by Eclipse PTP
  - Code visibility
  - Multi-system build management
  - Performance tuning
  - Source code control
  - Issue Tracking
  - Documentation
- Eclipse PTP Resources
Eclipse Documentation

Eclipse Help System – built in and standalone (http://help.eclipse.org)
Adapting Eclipse Documentation to Other Projects: QMCPack

See http://code.google.com/p/qmcpack-doc/

Developers' and users' guides

org.cmcc.qmcpack.doc is developed as an eclipse plug-in for QMCPACK help page. If all goes well, a help document with
• build instructions
• doxygen code documentation
• other materials on wiki

can be downloaded as an eclipse plug-in.

Licensed under UIUC/NCSA open-source license

See more on UIUC/NCSA license

Instructions for viewing help page in eclipse
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - Improvements to Eclipse PTP
- Software Engineering Practices Enabled by Eclipse PTP
  - Code visibility
  - Multi-system build management
  - Performance tuning
  - Source code control
  - Issue Tracking
  - Documentation
- Eclipse PTP Resources
Online Information

- Information about PTP
  - Main web site for downloads, documentation, etc.
    - http://eclipse.org/ptp
  - Wiki for designs, planning, meetings, etc.
    - http://wiki.eclipse.org/PTP
  - Articles and other documents
    - http://wiki.eclipse.org/PTP/articles

- Information about Photran
  - Main web site for downloads, documentation, etc.
    - http://eclipse.org/photran
  - User’s manuals
Mailing Lists

- **PTP Mailing lists**
  - Major announcements (new releases, etc.) - low volume
    - http://dev.eclipse.org/mailman/listinfo/ptp-announce
  - User discussion and queries - medium volume
    - http://dev.eclipse.org/mailman/listinfo/ptp-user
  - Developer discussions - high volume
    - http://dev.eclipse.org/mailman/listinfo/ptp-dev

- **Photran Mailing lists**
  - User discussion and queries
    - http://dev.eclipse.org/mailman/listinfo/photran
  - Developer discussions –
    - http://dev.eclipse.org/mailman/listinfo/photran-dev
Getting Involved

- See http://eclipse.org/ptp
- Read the developer documentation on the wiki
- Join the mailing lists
- Attend the monthly developer meetings
  - Conf Call Monthly: Second Tuesday, 1:00 pm ET
  - Details on the PTP wiki
- Attend the monthly user meetings
  - Teleconference Monthly
  - Each 4th Wednesday, 2:00 pm ET
  - Details on the PTP wiki

PTP will only succeed with your participation!