Improving the Eclipse Parallel Tools Platform in Support of Earth Sciences
High Performance Computing

Jay Alameda
National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign
Wyatt Spear
Department of Computer and Information Science, University of Oregon
Brian F. Jewett
Department of Atmospheric Sciences, University of Illinois at Urbana-Champaign

Software Engineering Assembly Conference 2013
3 April 2013
Acknowledgements

- 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), and Allen Malony (U Oregon). Team members and senior personnel include Beth Tibbitts (IBM), Ralph Johnson (U Illinois), Chris Navarro (NCSA), Sameer Shende (U Oregon), Wyatt Spear (U Oregon), Brian Jewett (U Illinois), Galen Arnold (NCSA), and Rui Liu (NCSA).
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Motivation for Workbench for High Performance Computing (WHPC)
  - 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
  - Earth Science/Weather code example
- Eclipse PTP Resources
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Motivation for Workbench for High Performance Computing (WHPC)
  - 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
  - Earth Science/Weather code example
- 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

![Diagram of Eclipse platform and tools](image)
Eclipse Parallel Tools Platform (PTP)

Coding & Analysis

Performance Tuning

Launching & Monitoring

Debugging
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Motivation for Workbench for High Performance Computing (WHPC)
  - 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
  - Earth Science/Weather code example
- Eclipse PTP Resources
Motivation for Workbench for High Performance Computing (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

- 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, …)
  - Need to cultivate community of users!
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 SR2 Released March 1, 2013

- Foci of improvements
  - Improve usability
  - Improve productivity
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  Run-10
Synchronized Projects

Projects types can be:
- File
- Service
- Index
- Service
- Launch Service
- Build Service
- Debug Service
- Local source code
- Source code copy
- Local Remote Compute
- Edit
- Search/Index Navigation
- Synchronize
- Executable

Synchronized Projects
Integrated OpenACC documentation and PLDT support
(added for BW)

Documentation also available for MPI, OpenMP
After the build, compiler errors, warnings, and loopmark information are shown in the Problems view and source code editor.
The Eclipse Parallel Tools Platform

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 &quot;dual-stream mode,&quot; 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 &quot;single-stream mode,&quot; 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></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.

Advanced Features: NCSA/XSEDE
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!
Improvements in the works...

- External Tools Framework (ETFW) being transitioned to use the resource manager “JAXB” XML descriptions.
  - Enhanced flexibility for tool integration
- Environment Management enhancements
  - Ordering of modules (important for some machines)
  - Connect environment management and scanner-discovery
- Separate Build system from CDT build
  - Cleaner multi-system build management
  - Builds in any language
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Motivation for Workbench for High Performance Computing (WHPC)
  - 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
  - Earth Science/Weather code example
- 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>
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
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)
  - Support for additional command-line tools can be added with XML tool definitions
- 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
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)
- Similar support for SVN, Git, ...
Mylyn Bridge
Tracks tasks, links to source and bug repositories

Connections to Jira, bugzilla, ...
Eclipse Help System – built in and standalone (http://help.eclipse.org)

Parallel Tools Platform

Release 5.0

The Parallel Tools Platform is designed to allow the Eclipse framework to be used for developing applications for parallel computer systems. PTP provides the following functionality:

- tools for developing applications based on the Message Passing Interface (MPI) standard and other environments and APIs including OpenMP, UPC, etc.
- the ability to launch, control and monitor the execution of parallel programs
- the ability to monitor parallel system status information
- an integrated parallel debugger
- a framework for integrating external dynamic tools

In addition, PTP provides a platform for parallel tool developers to integrate their tools within the Eclipse framework. New tools can take advantage of the user interface components and parallel services that are provided by PTP, without the need to develop and support this infrastructure on multiple platforms.

More information and downloads are available at http://eclipse.org/ptp.
Adapting Eclipse Documentation to Other Projects: QMCPack

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

Developers' and users' guides
org.cmscc.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
Consider 2 possible types of users of Eclipse Parallel Tools ...

- Weather code users/modelers
  - Need to build weather code
  - May need to modify weather code (and rebuild)
- Software specialists enabling modeling projects
  - Lots of software engineering concerns
- Next set of slides address some of those concerns.
Weather code users/modelers

- Some of the challenges
  - Complex codes (e.g., WRF)
  - Codes + HPC architectures can be daunting
  - Adding user code not always easy

Navigating Weather Codes

C/C++ - WRF_341/dyn_em/module_initialize_squall2d_x.F - Eclipse

- Project Explorer
  - netcdf4221
  - WRF_341
  - arch
  - Archives
  - Binaries
  - dyn_em
  - dyn_exp
  - dyn_mm
  - external
  - frame
  - inc
  - main
  - netcdf_links
  - phys
  - Registry
  - run
  - share
  - test
  - tools
  - clean
  - compile
  - configure
  - configure.wrf

- Source View

```
!MODEL_LAYER:INITIALIZATION
!
! This MODULE holds the routines which are used to perform
! for the individual domains.
!
! This MODULE CONTAINS the following routines:
!
! initialize_field_test - 1. Set different fields to values. This is only valid if a domain is not found (basen), then a fatal error is issued.

MODULE module_initialize_ideal

USE module_domain
USE module_io_domain
USE module_state_description
USE module_model_constants
```
Navigating Weather Codes

Code navigation
Navigating Weather Codes

Code navigation

Syntax-aware editing (navigate to program units and declarations)
Navigating Weather Codes

- Code navigation
- Code Outline
- Syntax-aware editing (navigate to program units and declarations)
Eclipse aiding in the WRF workflow...

- May want to add a model output variable
- Eclipse PTP makes it easy to navigate source, make changes

**WRF Build**
- Interactive “compile” script – use terminal within Eclipse to complete
- Configure Eclipse to drive “make” to iteratively build after modifications

**WRF Run** – can generate a run configuration for particular system, batch environment
Software Specialists enabling modeling projects

- Need a wider array of software engineering tools
  - Source repository
  - Issue tracking
  - Documentation
  - Performance tuning...

- Eclipse Parallel Tools can help with many of these concerns
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Motivation for Workbench for High Performance Computing (WHPC)
  - 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
  - Earth Science/Weather code example
- 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
    ✦ http://wiki.eclipse.org/PTP/photran/documentation
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!