Using the Eclipse Parallel Tools Platform to Assist Earth Science Model Development and Optimization on High Performance Computers

Jay Alameda
National Center for Supercomputing Applications
UCAR Software Engineering Assembly
22 February 2012
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), 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

Launching & Monitoring

Performance Tuning

Debugging
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
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
  - Minor enhancements and bug-fixes with two coordinated service releases in September and February
    - We are working towards a new major release now (Eclipse 4.2/Juno, released June 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
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
Code Visibility

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

Star 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
- Remote source code

Local source code

Edit

Search/Index Navigation

Synchronize

Run

Debug

Build

Executable

Local

Remote
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)

Parallel Tools Platform

**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.cmssc.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!