The Eclipse Parallel Tools Platform

Toward an Integrated Development Environment for Improved Software Engineering on Crays
Agenda

1. What is the Eclipse Parallel Tools Platform (PTP)
2. Tour of features available in Eclipse/PTP
   • Features added to support Blue Waters
3. How could Cray-PTP integration be improved
What is the Eclipse Parallel Tools Platform?

• Eclipse
  • Multi-platform integrated development environment
  • Extremely popular as a Java IDE
  • Excellent support for C/C++, UPC, Fortran, Python
  • Extensible via third-party plug-ins
    • Actually, *everything* is a plug-in!
    • Java, C/C++, Fortran support are all plugged in
    • CVS, Subversion, Git support are plugged in
    • What about plug-ins to support HPC?
What is the Eclipse Parallel Tools Platform?

- **Parallel Tools Platform (PTP)**
  - Set of Eclipse plug-ins
  - Adds support for HPC development to Eclipse
    - Write code on your laptop; compile on an HPC resource
    - Submit jobs to a batch scheduler; monitor jobs
    - Debug remote MPI applications (parallel debugger)
    - Get assistance with MPI, OpenMP development
  - Current release: about 50,000 downloads
Supporting Blue Waters

- **Blue Waters**: Cray XE6/XK6 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 (!)
The Eclipse Parallel Tools Platform

File Navigation
Syntax-aware Editing
Code Outline
Static analyses built in, available in real time while coding

< Static call hierarchy
Integrated OpenACC documentation (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.

```fortran
!$acc parallel loop
  do i = 1, 1000
    c(:, :) = (a(:, :) + b(:, :)) / 2.0
    a(:, :) = (a(:, :) + c(:, :) / 2.0) / 2.0
    b(:, :) = (b(:, :) + c(:, :) / 2.0)
  end do
!$acc end parallel loop
```

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

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)
• Source code editing
• Code search/navigation
• Static analysis

• Compilation
• Running and debugging
• Performance tuning

Local Source Code → Synchronize → Copy of Source Code
Configuring environment modules for build (added for BW)
Build is performed on remote machine (via SSH)
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)
Graphical interface for system monitoring
Enabling Better Software Engineering

- Integrated static analyses, available in real time
- Language-aware code completion
- Language-aware code searching and navigation
- Automated refactoring
- Integrated documentation, available in real time
- Easy-to-use, graphical interfaces for
  - Version control (CVS, Subversion, Git)
  - Issue tracking (Bugzilla, Jira)
Toward Better Cray Support

• PTP parallel debugger does not yet work
  • DDT does not have Eclipse integration, either

• Craypat could be integrated with PTP
  • TAU integrated using PTP’s *External Tools Framework* (*ETFw*)

• Refactorings could be built for OpenACC

• Loopmark information could be better integrated
  • E.g., used to suggest automated refactorings
  • *ETFw Feedback View* is designed to make this possible
Building a Cray-PTP Community

- Try PTP
  - See www.eclipse.org/ptp
  - Tutorials given at Supercomputing, XSEDE
    - Slides online: see wiki.eclipse.org/PTP/tutorials
- Ask questions and give feedback
  - Join the ptp-user mailing list
- Join the open source developer community
  - Join the ptp-dev mailing list
- How could PTP benefit your organization?