Developing Parallel Applications with the Eclipse Parallel Tools Platform

Greg Watson
IBM STG
grw@us.ibm.com
Parallel Tools Platform

*Enabling Parallel Application Development*

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

Coding & Static Analysis

Application Execution

Dynamic & Performance Analysis

Application Debugging
Coding & Static Analysis

- Eclipse provides a wide variety of coding assistance tools
  - Project management, Editing and formatting, Navigation, Advanced searching, Refactoring, Version control

- C/C++ Development Tools (CDT)
  - Standard (Makefile) and managed builders, Support for arbitrary toolchains, Visual debugging using GDB, High level views (outline view, call hierarchy, type hierarchy, include browser), Refactorings

- Parallel Tools Platform (PTP)
  - Fortran, New project wizards (MPI, OpenMP) Content Assist, Hover help, Built-in API descriptions (MPI, OpenMP, LAPI, UPC), Location of parallel “artifacts” in code (MPI, OpenMP, PAMI, and UPC), Barrier analysis, Deadlock detection
Coding & Static Analysis

Parallel Language Development Tools (PLDT)

- Assistance tools to increase productivity of parallel programmers
  - New project wizards (MPI, OpenMP)
  - Content Assist (command/API completion), hover help, built-in API help descriptions in an html help “view” (MPI, OpenMP, LAPI, UPC)
  - Location of parallel “artifacts” in code: MPI, OpenMP, LAPI APIs, and UPC
Coding & Static Analysis

Parallel Language Development Tools (2)

- Static analysis tools
- Provides “advanced” error checking
- Not always 100% accurate
- MPI analysis
  - Barrier deadlock detection
- OpenMP analysis
  - Concurrency analysis
  - Common OpenMP problems
Fortran Development Tools

- **Photran project**
  - http://eclipse.org/photran
  - Tech lead at UIUC, Jeff Overbey
  - More UIUC students are contributing
  - Merged with PTP in 2009
  - Photran 8.0 released in Jun ’12 with PTP 6.0 / Eclipse Juno

- **Photran features:**
  - Supports Fortran 77-2008
  - Syntax-highlighting editor
  - GUI interface to *gdb*
  - Makefile-based compilation
  - Compiler error extraction
  - Outline view
  - Open declaration
  - Fortran refactorings
  - C preprocessor support
Fortran Refactoring

Some samples:

- **Rename**
  - Change name of every use of a variable, function, etc.
  - Only proceeds if new name will be legal

- **Extract procedure**
  - Moves statements into a new subroutine, replacing statements with a call
  - Local variables are passed as arguments

- **Introduce implicit none**
  - Adds an ‘implicit none’ statement
  - Adds explicit variable declarations

- **Photran 7.0: 31 refactorings**
PTP Application Development Cycle

Coding & Static Analysis

Application Execution

Dynamic & Performance Analysis

Application Debugging
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 Execution

Target Configuration Framework

- Extensible framework for launching & monitoring
  - System and node status information
  - Job status (e.g. position in queue) & application status
  - Job submission & control
  - Debugger launch

- Configuration files to support different resource managers
  - Job schedulers (LoadLeveler, PBS, Torque, SLURM, GridEngine)
  - Interactive runtimes (e.g. PE, Open MPI, MPICH2, MVAPICH)
  - Systems (AIX, Linux, Power, x86, BG/Q, Cray)

- Local or remote system support
  - Command-line tools executed locally or via ssh connection
PTP Application Development Cycle

Coding & Static Analysis  
Dynamic & Performance Analysis

Application Debugging

Application Execution
Application Debugging

PTP 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
Application Debugging

IParallel Debugger Architecture

- Scalable debugger using multicast reduction network
- Integrated with PTP and launched using target configurations
- Supports basic debug commands
- Uses gdb on backend
PTP Application Development Cycle

Coding & Static Analysis

Application Execution

Dynamic & Performance Analysis

Application Debugging
Dynamic & Performance Analysis

**Dynamic Analysis Tools**

- 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
Dynamic & Performance Analysis

Tools Integrated with PTP

- **Tuning and Analysis Utilities (TAU)**
  - Instrumentation and transparent re-build of application executable
  - Execution of profiled application and collect performance data
  - Performance data visible in UI
  - Launches paraprof visualization client from Eclipse

- **Graphical Explorer of MPI Programs (GEM)**
  - Formal Dynamic Verification of MPI Applications
  - Detects all deadlocks, assert violations, MPI object leaks, and default safety properties
  - Matches sends and receives
  - Allows post-verification review of highlighted bugs
  - Works with a variety of MPI implementations
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

- **It’s Free!**
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)
Community

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

- **Meetings**
  - Monthly developer conference calls
    - Announced on ptp-dev list
  - Monthly user conference calls
    - Announced on ptp-user list
  - Annual developer/user meeting
    - Tentatively scheduled for Sept 18/19 in Chicago
Demo
Thank You