HPC Workbench
A Development Environment for Scientific Applications

Ted Hoover, Greg Watson, Beth Tibbitts
HPC Development
Application Development in HPC

- Developing Scalable Applications Becomes More Complex Each Generation
- Inconsistent Interfaces
- Difficult to Learn
- Expensive to Make
- Slow to Keep Pace with Mainstream Application Development Environment Technologies
- Low Productivity
Petascale Challenges

- How Can We…?
  - Enable Ultra Scalability
  - Minimize Time to Develop and Port Applications
  - Lower Entry Bar for New HPC Developers
  - Maximize Productivity

- With Better Languages… Better Libraries… Better Tools

Better Application Development Environment
PERCS Goals

- Collaborative Ecosystem Based on Eclipse
  - Parallel Tools Platform
  - Integrated Tools

- Multi Language Support
  - C, C++, UPC, Fortran, X10

- Platform Independence

- Highly Scalable

- Enhanced Productivity
  - Lower Entry Point for New Users
Introduction

What is Eclipse?

- Cross-platform open source framework for highly integrated state-of-the-art development tools
- Integrated tools include:
  - Project management
  - Advanced editing
  - Automated build system
  - Revision control (CVS, SVN)
  - Visual debugger
  - Much more…
- Designed to be robust, scalable, extensible, commercial quality
- Available for Linux, Unix and Windows
- Multi-language support for Java, C, C++, Fortran, Python, Perl, PHP, and others
Parallel Tools Platform

Enabling Parallel Application Development

- Best practice tools for experienced parallel programmers
- Tools to assist new breed of programmers to develop parallel programs
- Leverage Eclipse ecosystem and community for development and support
- Provide focal point for parallel tool development for a broad range of architectures
- Improve parallel tools and the productivity of tool developers
PTP Components

- **PTP Core**
  - Runtime, debugging and static analysis tools

- **Remote Development Tools (since PTP 2.1)**
  - Adds functionality to enable the remote development of applications

- **Photran (since PTP 3.0)**
  - Adds Fortran support to Eclipse

- **PLDT**
  - MPI, OpenMP, LAPI, UPC aware tools for development and analysis

- **ISP (since PTP 3.0)**
  - Adds In-situ Partial Order analysis of MPI programs – deadlock detection etc.
PTP Timeline

PTP 4.0 will be part of the Eclipse “Helios” Simultaneous release of 39 Eclipse projects – a first for PTP
PTP Contributors

- IBM
- Los Alamos National Laboratory
- Oak Ridge National Laboratory
- University of Tennessee at Knoxville
- University of Oregon
- Monash University
- National University of Defense Technology
- NCSA
- University of Illinois
- University of Utah
- Munich University of Technology
- Forschungszentrum Juelich
Collaborations

- A Scalable Development Environment for Peta-Scale Computing
  - DOE Office of Science funded project
  - IBM, ORNL, LANL, JSC, Monash U.
- Blue Waters Advanced Tools Collaboration
  - IBM, NCSA, UIUC, LSU, RENCI
HPC Workbench

- IBM’s PERCS deliverable for Eclipse tooling
- Builds on Eclipse and PTP
- Adds IBM proprietary tools
  - HPC Toolkit for performance analysis and optimization
  - Parallel Environment
- Focus on scalability and usability
  - Must scale to PERCS and BW size machines
  - Must ensure that tools are all tightly integrated
  - Must improve developer workflow
HPC Workbench Application Development Cycle

Coding & Static Analysis

Application Execution

Dynamic & Performance Analysis

Application Debugging
Coding & Static Analysis

Eclipse Platform

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

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)
- Advanced searching (types, functions, variables, declaration, reference, etc.)
- Content assist (automatic completion in the editor)
- Context sensitive help
- Simple refactorings (rename, extract constant, more coming…)
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
- MPI analysis
  - Barrier deadlock detection
- OpenMP analysis
  - Concurrency analysis
  - Common OpenMP problems
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

Resource Manager 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

- Plug-in adapters to support different resource managers
  - Job schedulers (e.g. LoadLeveler)
  - Interactive runtime systems (e.g. PE or OpenMPI)

- Local or remote system support
  - Command-line tools executed locally or via ssh connection
  - Remote proxy agent (can tunnel over ssh)
Eclipse Parallel Environment (PE) Plug-In

- Support for running interactive Parallel Environment applications within the Eclipse PTP framework
  - with/without LoadLeveler
- Support for submitting batch LoadLeveler jobs
- Monitor status of running applications
- Integrates with existing PTP resource scheduler views
Eclipse PTP Perspective Running PE Application

Resource Manager View: Configured resource manager

Machine view: Nodes in use

Jobs view: Currently running app

Jobs view: Application tasks

Sample Message Passing Program - Heat Transfer

Rectangular grid - temperature profile on left edge

top and bottom edges are insulating

right edge is zero degrees

each row (horizontal slice) is assigned to a task

each row exchanges data with row above and below

---

#include <mpi.h>
#include <stdlib.h>
#include <math.h>

number of x cells set by environment variable
N = mpirights; Number of cells = N + 1

time cells set by environment variable
N = NPROCS-1; Number of cells = N + 1

heat_equation:
T(j, i, k+1) = T(j, i, k) + h^2(T(j+1, i) + T(j-1, i) - 2T(j, i))

for non-exit cells, where

---
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
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

External Tools Framework

- Provides integration for instrumentation, measurement, and analysis for a variety of performance tools
- Developed to reduce the “plumbing” necessary to integrate external tools
- Integration points include
  - Instrumentation; automatic and selective
  - Build; may be transparent to user
  - Application launch with instrumentation
  - Management of profile/trace data
  - External tool launch
  - Visual feedback from external tool execution
  - Feedback View maps tool results to source code lines
Dynamic & Performance Analysis

**HPC Toolkit**

- Integrated with Eclipse and PTP
- Application and tools launched using PE resource manager
- HPC Toolkit
  - Provides an integrated framework for performance analysis
  - Looks at all aspects of performance (communication, memory, processor, I/O, etc) from within a single interface
  - Operates on the binary and yet provides reports in terms of source-level symbols
  - Full source code traceback capability
Dynamic & Performance Analysis

Other Tools supported by 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

- In-Situ Partial Order (ISP) Analysis
  - 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

Note: These tools are not included with PTP, but the eclipse integration is.
Remote Development

- Key requirement for HPC development
- Need Eclipse running on local workstation to avoid network latency
- Some development activities can be local or remote, but must be transparent to the developer
Remote Development Tools (RDT)

- Adds transparent remote development capability to CDT
- Builds on changes introduced into CDT 5.0 that allow projects to reside remotely
- Uses PTP *service model* to allow development services (e.g. build, index, launch, debug, etc.) to be independently selected as local or remote
- Other parts of PTP also use the service model to ensure uniformity and ensure usability
- Includes a new remote project creation wizard
Fully Remote

- All services are provided by remote systems
- Project source is physically located and indexed remotely
- Builds are performed on remote system
- Launching and debugging on remote system (possibly different from the project/build system)
Partially Remote (Off-line Development) – Future Release

- Application source resides locally
- Uses normal local indexing service
- Source synchronized with remote system prior to build
- Builds are performed on remote system
- Launching and debugging on remote system (possibly different from the build system)
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 5.0 released in Fall ‘09
  - Photran 6.0 to be released in Jun ’10 with PTP 4.0 / Eclipse Helios

- Photran features:
  - Supports Fortran 77-2003
  - Photran 6.0: Fortran 2008 support
  - Syntax-highlighting editor
  - GUI interface to gdb
  - Makefile-based compilation
  - Compiler error extraction
  - Outline view
  - Open declaration
  - Fortran refactorings: this is the research focus of the UIUC group
  - 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 6.0: 16 refactorings**

Image courtesy Jeff Overbey, UIUC
Current Development Activities

- **IBM**
  - Scalability improvements in the runtime and debugger
  - Improvements and enhancements to RDT
  - Ongoing CDT development
  - Feedback View improvements to further integrate external tool-generated information with user interface and source code annotations

- **University of Oregon**
  - Improvements to TAU integration
  - External tools framework enhancements

- **Oak Ridge National Laboratory**
  - Usability enhancements

- **Los Alamos National Laboratory**
  - Core PTP scalability issues

- **NCSA**
  - PBS job scheduler
  - X.509 authentication

- **Monash University**
  - New debugging techniques
  - Improvements and enhancements to RDT
  - Debugger scalability

- **UIUC**
  - CPP support for Fortran
  - Additional Fortran refactorings

- **Jülich Supercomputer Centre**
  - Scalability of UI, particularly job schedulers

- **University of Defense Technology**
  - SLURM improvements and enhancements

- **University of Utah**
  - GEM (Graphical Environment for MPI) enhances ISP
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

- **HPC Workbench will provide a robust, extensible, and scalable platform for peta-scale application development**
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)

- **Information about Photran**
  - Main web site for downloads, documentation, etc.
    - [http://eclipse.org/photran](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](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)

- **Photran Mailing lists**
  - User discussion and queries
    - [http://dev.eclipse.org/mailman/listinfo/photran](http://dev.eclipse.org/mailman/listinfo/photran)
  - Developer discussions –
    - [http://dev.eclipse.org/mailman/listinfo/photran-dev](http://dev.eclipse.org/mailman/listinfo/photran-dev)
Thank You