

A Model-Based Framework for the Integration of Parallel Tools

Greg Watson
Nathan DeBardeleben



Eclipse

What is Eclipse?

- A vendor-neutral open source development platform
 - Integrated development environment
 - Wide variety of tools available
- A universal platform for tools integration
 - Designed to support the integration of tools
 - Provides a uniform interface across a wide variety of tools
- Plug-in based framework to create, integrate and utilize software tools
 - Designed to enable the development of integrated tools
 - Flexibility through plug-in architecture
 - Plug-in development uses Eclipse

Eclipse

Main Features (1)

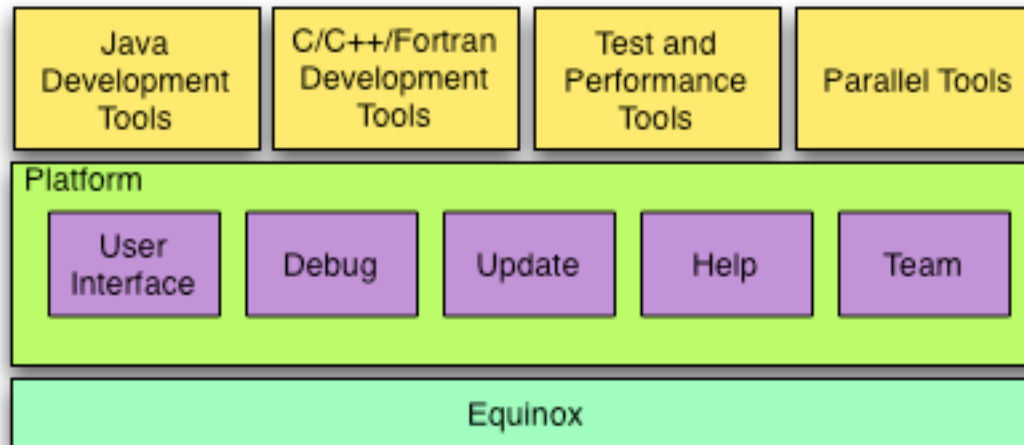
- Rich User Interface
 - Perspectives
 - Tabbed Views
 - Editors
 - Toolbars
 - Preferences
- Plug-in Architecture
 - Ability to load new functionality
 - Manage plug-ins through update wizard
 - Updates can be loaded directly from a web site
 - A plug-in can contribute any functionality
 - Eclipse is built entirely using plug-ins

Eclipse

Main Features (2)

- Rich Set of Core Functionality
 - Integrated help system
 - Team versioning environment
 - Update management
 - Language independent debug framework
 - Build support
- Broad User Community
 - Open source license (EPL)
 - 60% Java, 40% C/C++ developers
 - Hundreds of third party plug-ins available
 - The basis of many commercial products
 - Tool developers, application developers, application users

Eclipse Architecture



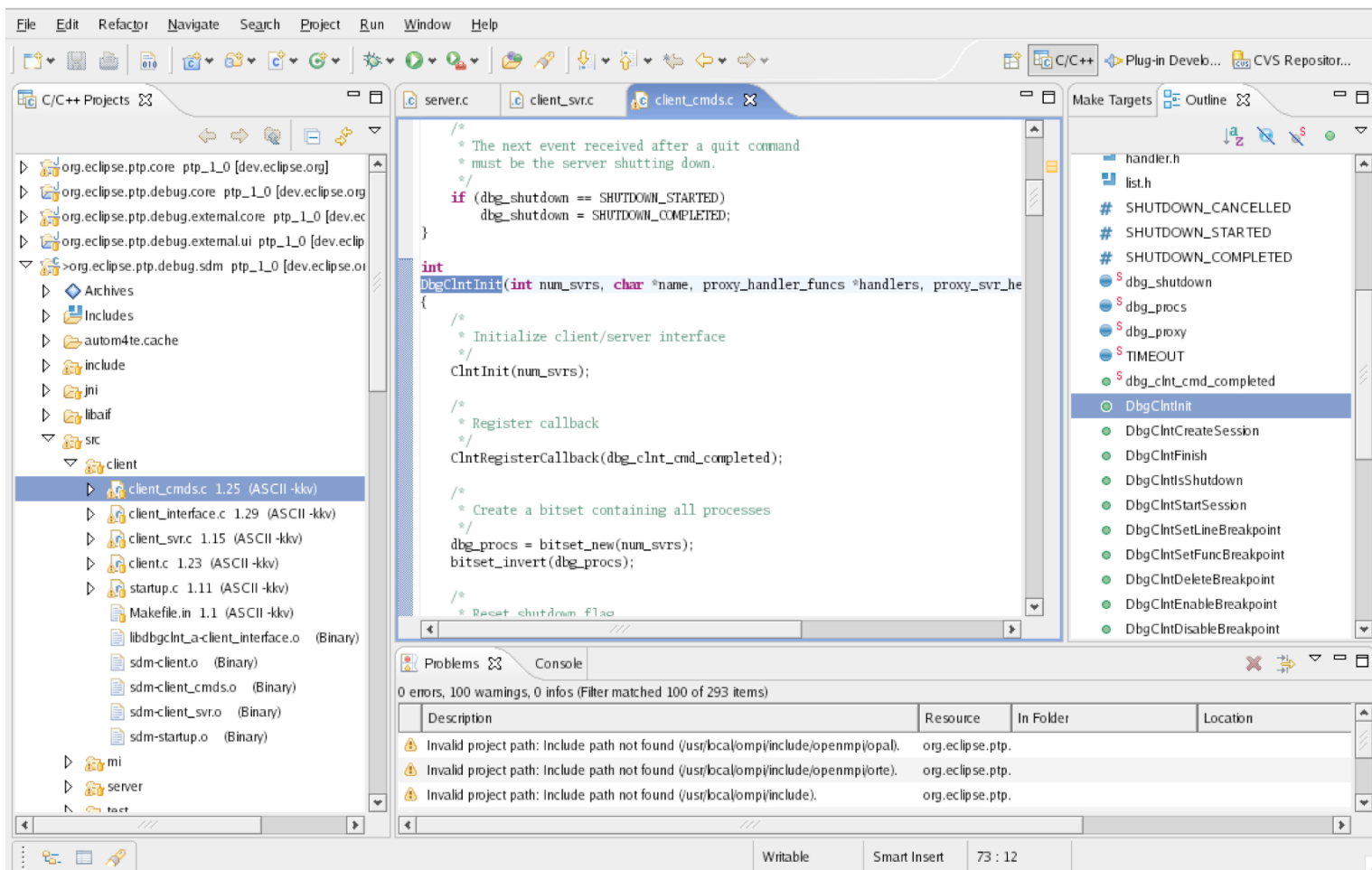
Eclipse Equinox

- OSGi framework implementation model
 - Formerly known as the Open Services Gateway initiative
 - Standard for application lifecycle management
- Provides the most fundamental Eclipse infrastructure
 - Plug-ins (known as a bundle)
 - Bundle install, update and uninstall
 - Bootstrap and launching
 - Extension registry
- Introduced in Eclipse 3.0

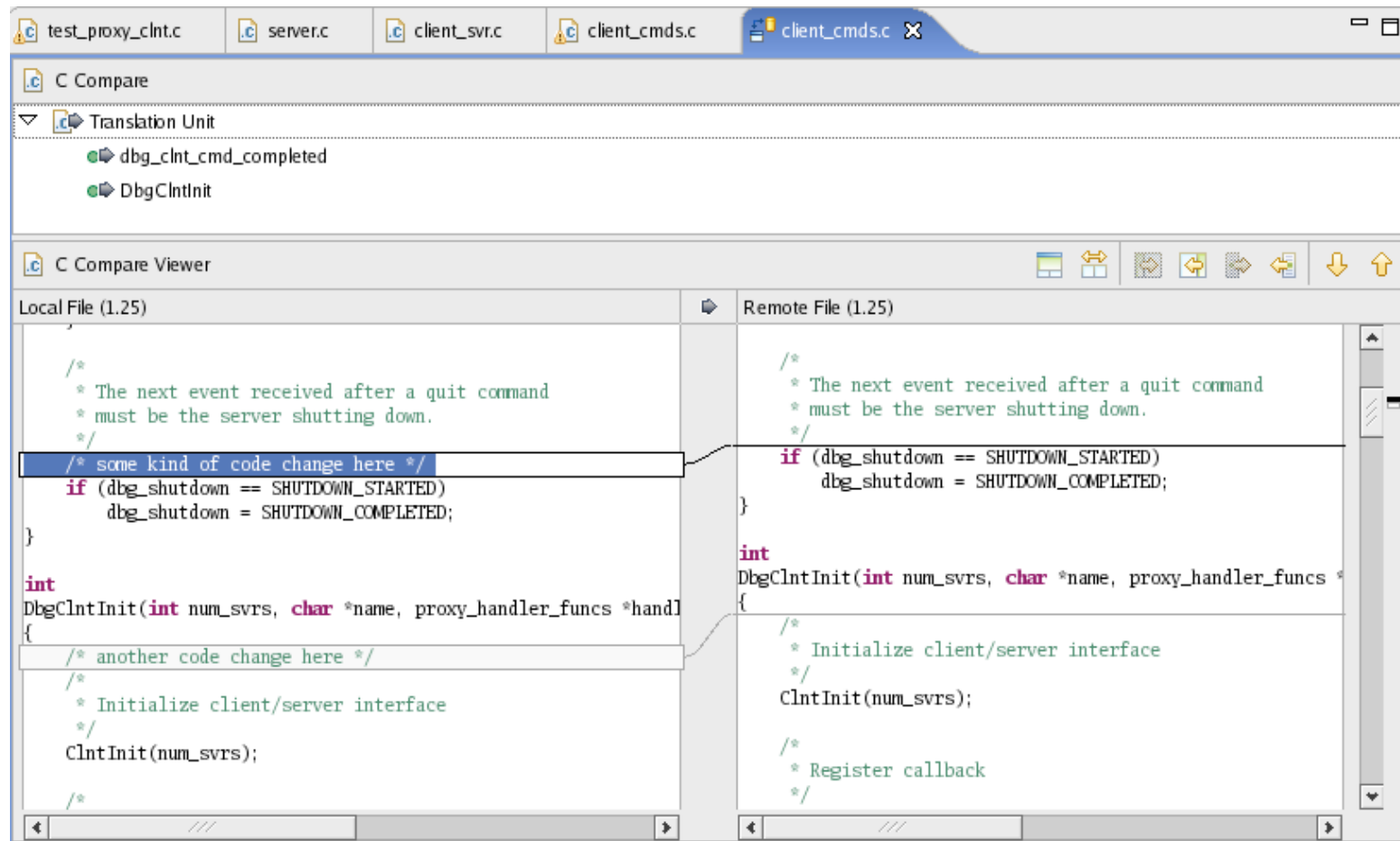
Eclipse Platform

- Core frameworks and services up which all plug-in extensions are created
- Represent the common facilities required by most tool builders:
 - Workbench user interface
 - Project model for resource management
 - Portable user interface libraries (SWT and JFace)
 - Automatic resource delta management for incremental compilers and builders
 - Language-independent debug infrastructure
 - Distributed multi-user versioned resource management
 - Dynamic update/install service

Eclipse User Interface



Eclipse Team Support



Eclipse Plug-ins

- Java Development Tools (JDT)
- Plug-in Development Environment (PDE)
- C/C++ Development Tools (CDT)
- Parallel Tools (PTP)
- Test and Performance Tools (TPTP)
- Business Intelligence and Reporting Tools (BIRT)
- Web Tools (WTP)
- Data Tools (DTP)
- Device Software Development (DSDP)
- Many more...

Parallel Tools Platform

Project Objectives

1. Extend Eclipse to support parallel development tools
2. Equip Eclipse with key tools needed to start developing parallel codes
3. Encourage existing parallel tool projects to support Eclipse by providing integration services
4. Exploit enhanced capabilities to develop a new generation of parallel tools

Parallel Tools Platform Participants

- LANL
- IBM
- University of Oregon
- University of Tennessee
- Monash University

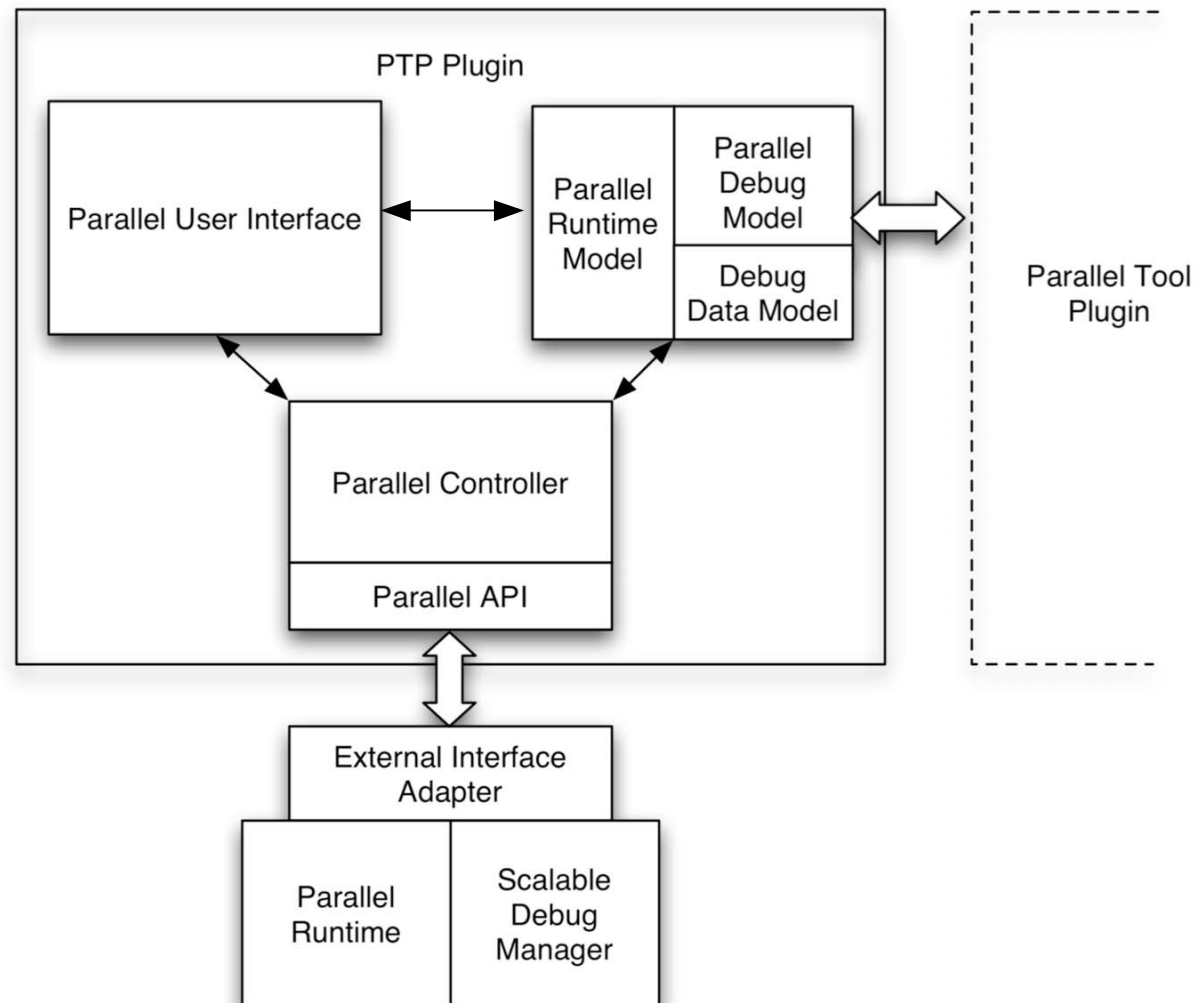
Parallel Tools Platform Components

- Parallel Model
 - Abstract representation of a parallel machine
 - Abstract representation of a running job
- Parallel Execution Environment
 - Supports launching, controlling, and debugging parallel programs on arbitrary parallel machines
- Parallel Debugger
 - A scalable parallel debugging framework
- Tool Integration
 - Support the integration of a variety of parallel tools, e.g. performance, verification, visualization, components

Parallel Tools Platform Architecture

- Parallel Tools Platform Plug-in
 - Model/view/controller (MVC) architecture
 - Extends existing components where necessary (e.g. debug model)
 - Adds new parallel functionality (e.g. parallel launch wizard, user interface components)
 - Utilizes existing language support (e.g. CDT)
 - Provides infrastructure to support other parallel tools
 - Interfaces to external parallel runtime systems
- External Parallel Runtime and Scalable Debug Manager
 - Targets OpenMPI runtime (*not* dependent on MPI)
 - Will support other runtime systems
 - MPICH2 support under development

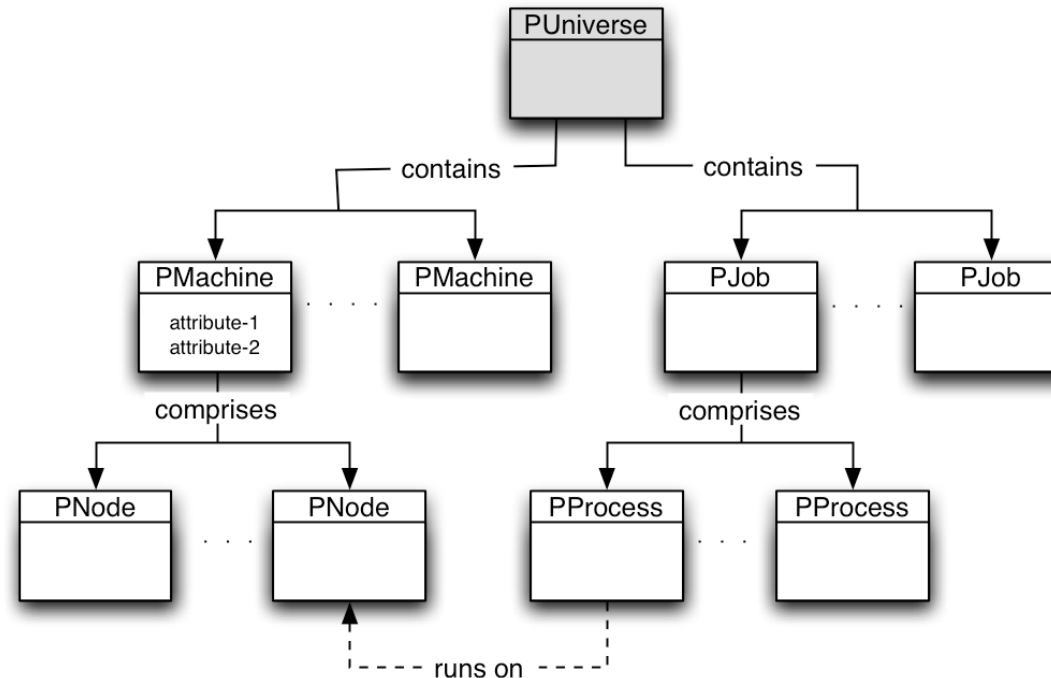
Parallel Tools Platform Plugin Detail



Parallel Tools Platform

Parallel Model

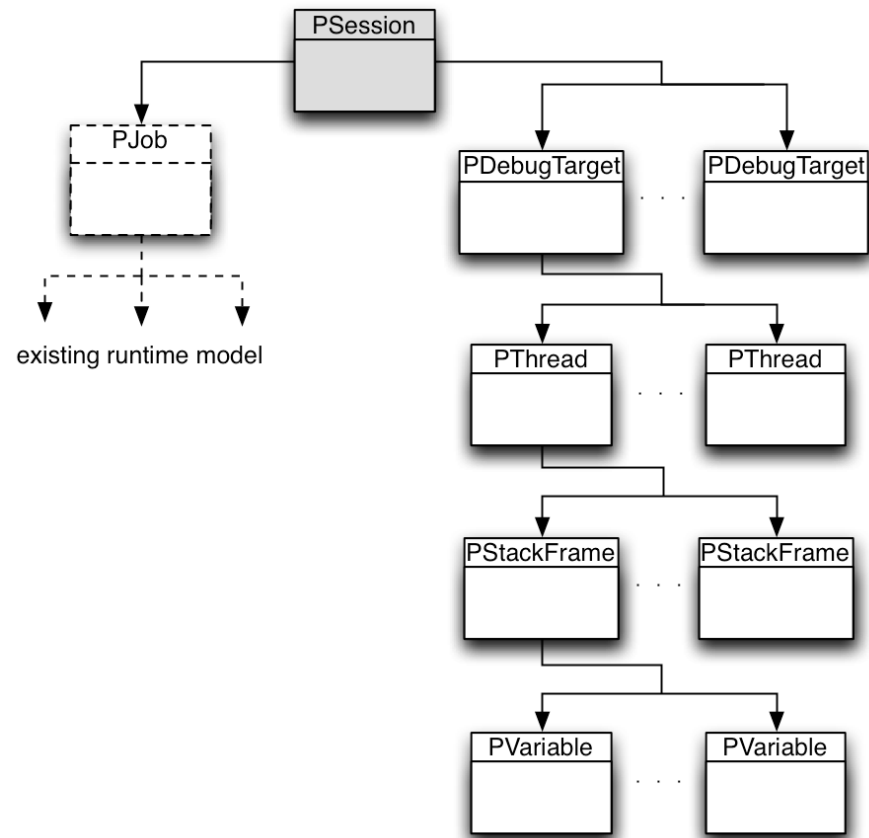
- Runtime Model
 - Defines notion of *universe*, parallel machines, jobs, processes, etc.
 - Model elements support arbitrary attributes



Parallel Tools Platform

Parallel Model

- Debug Model
 - Extends the Eclipse platform debug model to support parallel processes
 - Support for grouping of debug operations
 - Provides scalable event management
 - Efficient data handling and internal data manipulation



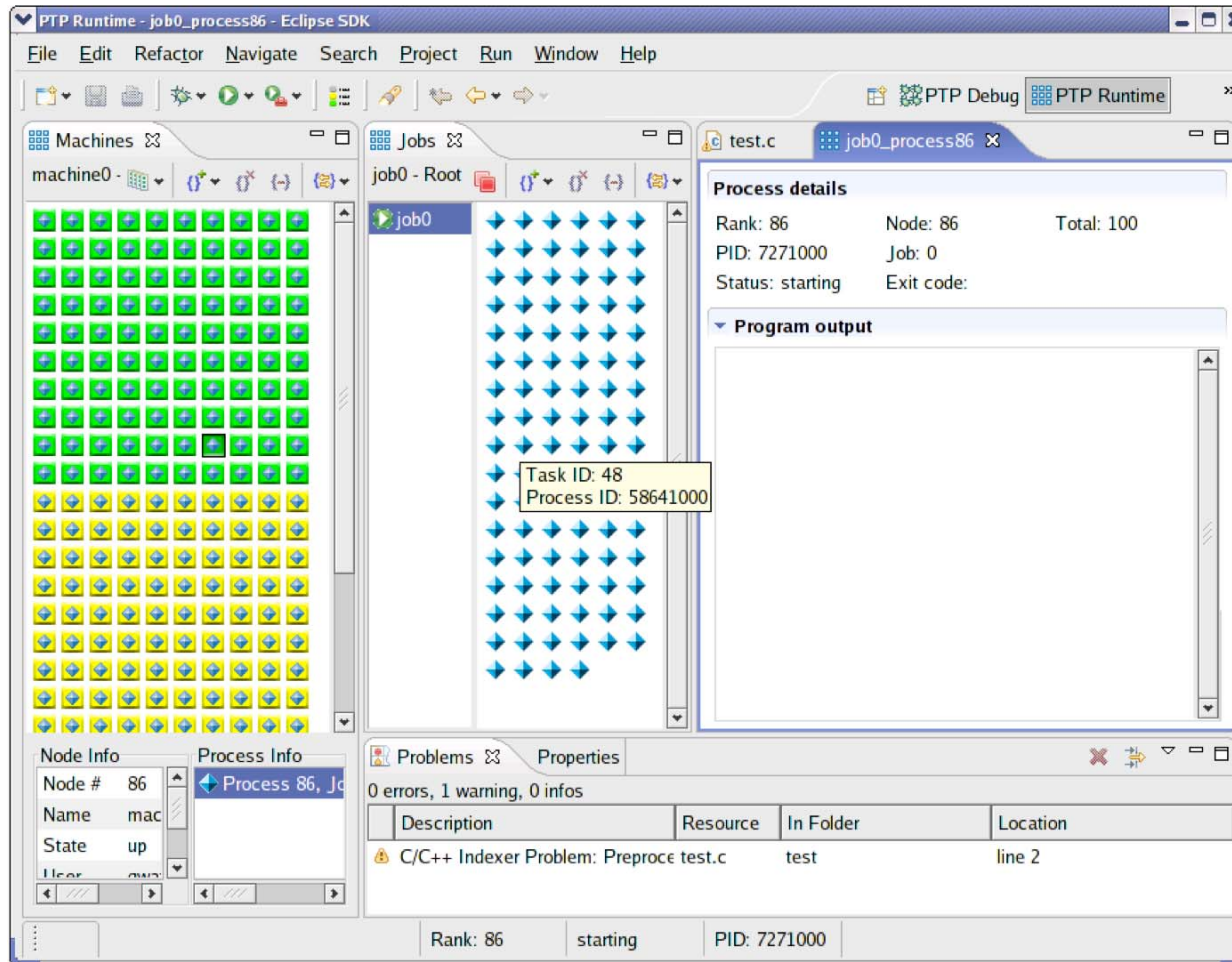
Parallel Tools Platform

Parallel User Interface

- Specifically designed to provide compact and scalable interface to parallel components
- Extends existing debug interface to support parallel processes
 - Infrastructure for array viewing and new visualization tools
- Parallel Launch Configurations
 - Specify resource requirements (e.g. number of processes, execution time, etc.) and interface to launch services

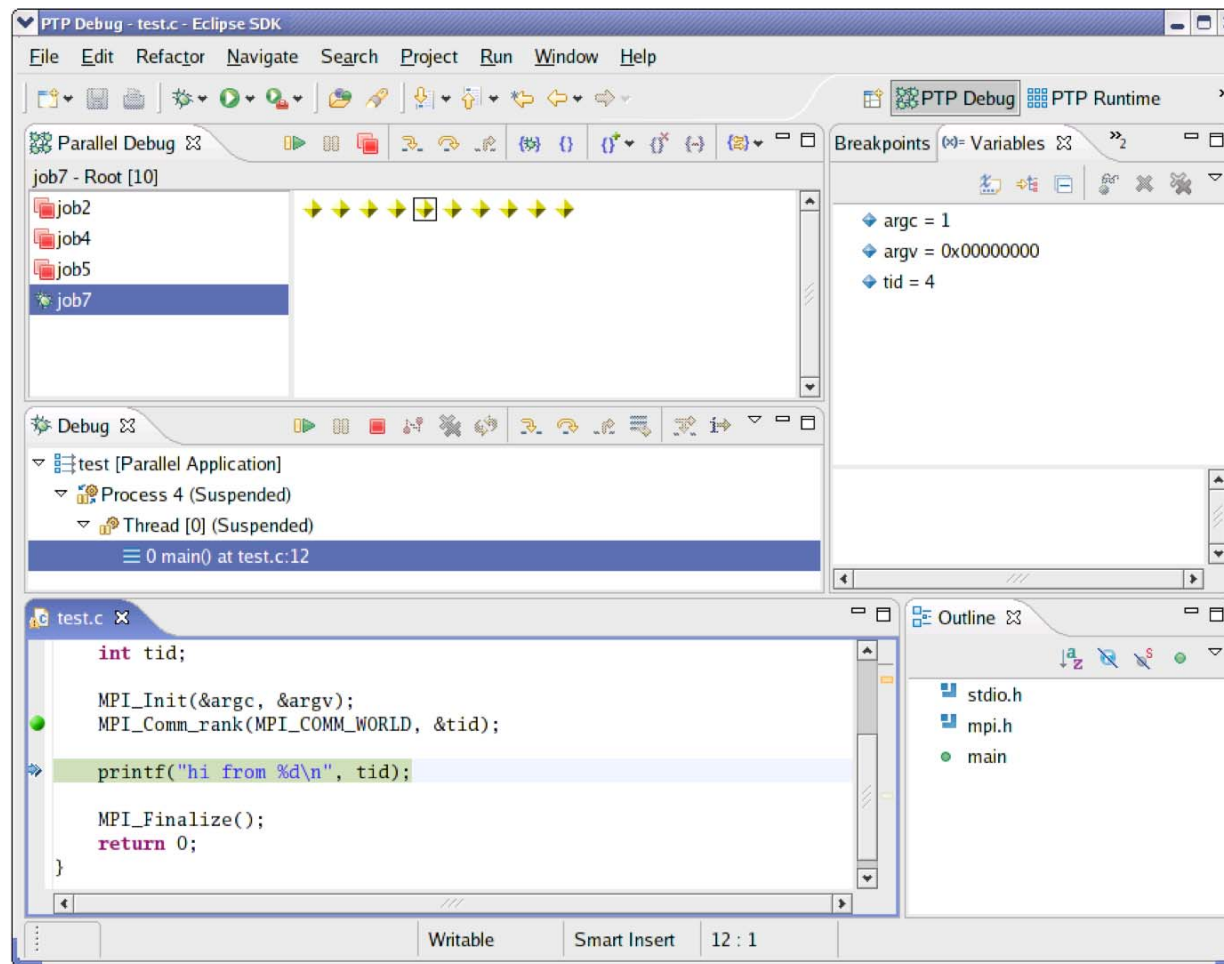
Parallel Tools Platform

PTP Runtime Perspective



Parallel Tools Platform

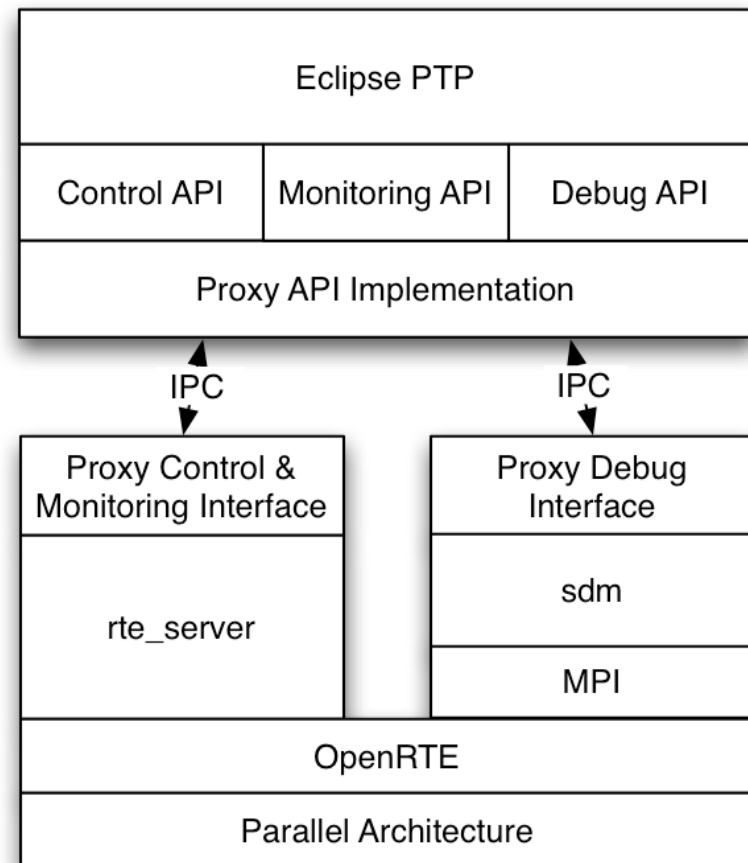
PTP Debug Perspective



Parallel Tools Platform

Parallel API

- High level interface for interacting with parallel machines, managing jobs, etc.
- High level API for managing debug operations
- Allows PTP to support different parallel runtime systems and parallel debug managers



Parallel Tools Platform

Scalable Debug Manager

- Manages scalable and efficient debugging of enormous parallel programs
- Uses standard MPI startup mechanism
- Broadcast topology for efficient communication with large numbers of processes
- Event coalescing to manage large volumes of events

Parallel Tools Platform

Tool Integration

- Parallel Model
 - Available for other tools to use
 - Generic attributes allow other tools to annotate model
- Runtime Services
 - Services for launching, running and controlling parallel programs
- Debug Services
 - Services for managing parallel programs under debugger control
- User Interface Components
 - Reusable user interface components designed to be compact and scalable

Parallel Tools Platform

Future Work

- Remote system support
 - Allow Eclipse running on desktop to monitor/control/debug jobs running on remote parallel machines
- Resource manager support
 - Provide monitoring and submission of jobs through remote resource management systems, such as SLURM, LSF, etc.
- Advanced debugging
 - Develop new debugging techniques for very large programs (10's or 100's of thousands of processes)
- Parallel language tools
 - Use static analysis framework to provide support for developing MPI, OpenMP and other types of parallel languages

Conclusion

- Leverages the worlds most widely used open-source IDE
- Provides core services for developing, running and debugging parallel programs using Eclipse
- Provides a parallel debugger framework that can be used as a platform for developing new debugger technology
- Encourages existing and new tool developers to integrate their parallel tools into a world-class platform
- Source available at
 - <http://eclipse.org/ptp>