The Eclipse Framework for Tool Development

Vivek Sarkar
IBM T.J. Watson Research Center
vsarkar@us.ibm.com
May 3 - 4, 2005

This work has been supported in part by the Defense Advanced Research Projects Agency (DARPA) under contract No. NBCH30390004.
Acknowledgments

• IBM PERCS team members
  - PERCS Tools
    • Marina Biberstein, Robert Fuhrer, Emmanuel Geay, Matthias Hauswirth, Peter Sweeney, Beth Tibbitts, Frank Tip, Mandana Vaziri
  - PERCS Productivity
    • Catalina Danis, Kemal Ebcioğlu, Jason Ellis, Christine Halverson
  - PERCS leaders in STG product group
    • Mootaz Elnozahy (PI), Rama Govindaraju (Software lead)
• Eclipse Parallel Tools Platform (PTP) team at LANL
  - Contact: Greg Watson, PTP team lead
• IBM Rational Software
  - Contact: John Kellerman, IBM Eclipse Product Manager
Outline

1. What is Eclipse?
2. Eclipse Parallel Tools Platform (PTP) project
3. PERCS Productivity Tools
What Is Eclipse?

• A software platform …
  - Eclipse is a universal platform of frameworks and exemplary tools that make it easy and cost-effective to build and deploy software in today’s connected and unconnected world.

• … and a community (eclipse.org)
  - Eclipse is a consortium of major software vendors, solution providers, corporations, educational and research institutions and individuals working together to create an eco-system that enhances, promotes and cultivates the Eclipse open platform with complementary products, services and capabilities.
Eclipse Architecture

Runtime (OSGi)

A component runtime
Eclipse Architecture

UI (Generic Workbench)

JFace

SWT

An extensible application platform
Eclipse Architecture

Many plug-ins built on top

Help (optional)  Update (optional)  Text (optional)  IDE  Resources

IDE

IDE Text  Compare  Debug  Search  Team/CVS

UI (Generic Workbench)

JFace

SWT

Runtime (OSGi)

A suite of tools and extensions
Each Eclipse component is a Plug-in

Eclipse platform and plug-ins are implemented in Java
Plug-in Architecture

- **Plug-in**
  - smallest unit of eclipse functionality
  - plugins can add code, define extension points, and contribute to extension points

- **Extension point**
  - named entity for collecting contributions
  - Defines API contract
  - example: extension point to add menu actions

- **Feature**
  - Group of plug-ins that form a single installable chunk
Key Eclipse Functionalities (Incomplete List)

- Standard Widget Toolkit (SWT)
  - Leverages native widgets with platform-independent API
- JFace
  - UI framework built on top of SWT, with support for trees, tables, lists, etc.
- Workbench
  - Views, editors, perspectives
- Graphical Editing Framework (GEF)
- Scalable lazy loading of plug-ins
- Extensive use of background jobs for UI responsiveness
- Update manager
- Extensive help system
- Cheat sheets
- Code assist
- Built-in support for CVS
- Plug-in Development Environment (PDE)
- Ultra-customizable user experience
- ...
Eclipse is ultra-customizable
e.g., Java Development Toolkit
Eclipse is ultra-customizable
e.g., C/C++ Development Toolkit

- Editing and Navigation
  - Syntax highlighting,
  - outline view, search,
  - content assist
- Debug
  - GDB Integrated
  - Extensible Debug Interface
- Build
  - Standard Make for
  - projects with existing build infrastructure
  - Managed Build
- Extensibility
Eclipse is ultra-customizable
e.g., Fortran plug-in

NOTE: Photran project is led by Ralph Johnson, UIUC. See www.photran.org for details
Eclipse is ultra-customizable
e.g., tool for Geographic Data Files
Eclipse Eco-System:
Eclipse Powered Commercial Offerings

• Change Management
  Aldon
  CA AllFusion Harvest Change Manager
  Interwoven TeamSite
  Sarena PVCS Version Manager, Dimensions, Tracker
  Minerva SoftCare Change
  MKS Source Integrity Enterprise
  Serena ChangeManDS
  SoftLanding TurnOver
  Borland Starbase StarTeam
  Telelogic CMSynergy

• Asset Management & Reuse
  CAST Enlighten, Envision
  Flashline CMEE
  LogicLibrary Logidex
  Business Process Integration
  CommerceQuest Business Process Integrator
  SOA
  Systinet WASP Developer

• Build/Deploy
  Catalyst Systems OpenMake
  Genuitec EASIE Plug-in Suite for J2EE, Plug-in for WSAD
  ObjectEdge WebLogic Plug-in for WSAD

• Design & Programming
  Embarcadero Describe
  Advanced SystemConcepts
  SWTWorkbench
  Bowstreet Factory
  Canoo ULC Launcher, JVE
  Extensions
  Dassault CAA
  Exaltec B+
  Instantiations CodePro Studio, SWT Designer
  Lansa
  M7
  Persistence EdgeExtend
  Versant enJin
  Visual SlickEdit

• Development Environments
  VA Software SourceForge
  LegacyJ PERCobol
  MontaVista DevRocket
  QNX Momentics
  SAP NetWeaver Developer Studio
  Sybase EASStudio

• Quality
  CAST Enforce
  Candle PathWAI Performance Analyzer
  Crunchy Technology
  PageScreamer
  Ensemble Systems Glider
  Mid-Comp Oyster
  OpenDemand OpenLoad
  Parasoft jtest, webking, soaptest
  Quest Jprobe
  QASystems Qstudio
  Sanctum
  Scapa Technologies StressTest
  TeamStudio for Java
Eclipse: the proof is in the numbers

- Eclipse Platform download requests topped 3.1 Million in first year
- To date over 39 Million download requests
  - Over ~640K developers, companies, or organizations from over 125 countries
  - 15% Linux downloads, 80% Windows downloads, 3% Solaris
- Number of open source projects: 36
- Number of third party plug-ins: 680+
- Number of books published about Eclipse: 19
  - Books in German, Japanese, Korean
- Eclipse related projects listed on SourceForge: 500+
  Broad community growth
  Over 20 Eclipse information sites worldwide

*Download data as of 1Q05*
Outline

1. What is Eclipse?
2. Eclipse Parallel Tools Platform (PTP) project
3. PERCS Productivity Tools
Parallel Development Tools
State of the Art

- Command-line compilers for Fortran and C/C++
  - Sometimes wrapped in a GUI
- Editors are vi, emacs and FRED (vintage 1960’s)
- Plethora of stand-alone tools
  - Platform/vendor specific, e.g. DCPI
  - Open source, e.g. TAU, HPCToolkit
  - Proprietary, e.g. Vampir, Assure, TotalView
- Many tools are specific to only one platform or vendor, and they do not interoperate
- No integrated UI
  - No ability to share data
- Current tools do not scale
  - New machines will have 10,000+ processors
Parallel Development Tools
Barriers To Change

- Must support a range of architectures/platforms
- Must honor existing practices/processes
- Must be scalable and reliable
- Must provide core functionality
- Must be easy to adopt and support
- Must be future-proof
Parallel Tools Platform (PTP) Project

Eclipse PTP project initiated in Feb 2005

• Project leader: Greg Watson, LANL
• Multi-organization open source project --- partners & contributors are welcome!

First release planned in Sep 2005

PTP 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
4. Exploit enhanced capabilities to develop a new generation of parallel tools
Parallel Tools Platform Architecture

Eclipse Platform

PTP  JDT  CDT  FDT  Other Tools

External Parallel Runtime

Parallel Machine  Parallel Machine  Parallel Machine
Parallel Tools Platform Architecture

- Parallel Tools Platform Plug-in
  - 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
  - Target OpenMPI runtime
  - Should support other runtime systems (requires work)
Parallel Tools Platform
Parallel Runtime Model

- Defines notion of *universe*, parallel machines, jobs, processes using Eclipse Modeling Framework (EMF)
Parallel Tools Platform
Parallel Debug Model

- **Debug Model**
  - Extends platform/CDT debug model to support parallel processes
  - Support for group operations
  - Scalable event management

- **Debug data model**
  - Language independent (support for *arbitrary data types*)
  - Efficient conversion to/from Java native types
  - Value caching and lazy evaluation

- **Scalable Debug Manager**
  - Manages scalable and efficient debugging of enormous parallel programs
    - Launching processes under debug control
    - Communication with large numbers of processes
    - Efficient data transfers and event management
Parallel Tools Platform
End-User Support

• Many components of PTP support *running* and *managing*, rather than *developing* parallel programs

• Utilize RCP to provide an environment that targets the end-user, rather than the developer

• Deal with issues that are usually ignored:
  - How to manage program input and output data
  - Pre- and post-processing of data
  - Visualization, model coupling, etc.
Outline

1. What is Eclipse?
2. Eclipse Parallel Tools Platform (PTP) project
3. PERCS Productivity Tools
IBM PERCS project
(Produ"cive Easy-to-use Reliable Computing Systems)

Increase overall productivity

Increase performance of applications

Increase execution productivity

PERCS Programming Tools
performance-guided parallelization and transformation, static & dynamic checking, separation of concerns --- all integrated into a single development environment (Eclipse)

X10 Programming Model
OpenMP
MPI

Static and Dynamic Compilers + Language Runtimes
Mature languages: C/C++, Fortran, Java
Emerging language: UPC
Experimental language: X10

Continuous Optimization + Communication Libraries (MPI/LAPI) + Parallel Environment

PERCS System Software (K42, vHype)

PERCS System Hardware

IBM Petascale Tools Strategy Workshop
Slide 28
PERCS Programming Model, Tools and Compilers: Overall Architecture

X10 source code

Java source code (w/ threads & conc util) => X10 Development Toolkit

C/C++ source code (w/ MPI, OpenMP, UPC) => Java Development Toolkit

Fortran source code (w/ MPI, OpenMP) => C/C++ Development Toolkit + MPI tools

Java Compiler

C/C++ Compiler + UPC support

Fortran Compiler

X10 Components

Java components

Fast extern interface

C/C++ components

Fortran components

X10 runtime

Java runtime

C/C++ runtime

Fortran runtime

Text in blue identifies PERCS extensions

Dynamic Compilation + Continuous Program Optimization

Integrated Concurrency Library: messages, threads, synchronization

IBM Petascale Tools Strategy Workshop

Slide 29
PERCS Programming Model & Tools: Summary of Productivity Approach

1. **Integrated Development Environment**
   - Enhance Eclipse PTP to obtain a customized Workbench for HPC users (C/Fortran build support, MPI code assist, etc.)

2. **Debugging tools**
   - Integrate current tools (pdbx, ...) into Eclipse
   - Advanced static & dynamic program analysis for detection & analysis of data races, deadlock, memory leaks, ...

3. **Component Libraries**
   - Tools to support application development with commonly used HPC libraries (MPI, ESSL, LAPACK, SCALAPACK, ...)
     - Code assist, detection of common errors, ...

4. **Separation of Concerns**
   - Morphogenic software: exploit aspect-oriented software development for separation of concerns between domain experts and performance experts

5. **Performance tuning**
   - Focus on performance analysis and tuning across multiple system layers (PEM infrastructure, Performance Explorer tool)

6. **New languages and programming models**
   - Use experimental language (X10) and related tools (X10DT) to demonstrate how changes in programming model can address productivity obstacles
Conclusion

- Productivity will be the biggest bottleneck in future petascale systems
- Unique opportunity to move parallel development to best practices
  - Approach should be designed to address scalability and performance issues from the start
- Eclipse and the Eclipse Parallel Tools Platform project offer a promising foundation for petascale tools