

# The Eclipse Parallel Tools Platform Project

An Eclipse Foundation Technology Project

<http://eclipse.org/ptp>

# 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)
- Dominant debugger is TotalView (proprietary)
  - Some use DDT (but guess what? - it's proprietary)
  - No widely used open-source parallel debugger
- Plethora of stand-alone tools
  - Platform/vendor specific, e.g. DCPI
  - Open source, e.g. TAU, HPCToolkit
  - Proprietary, e.g. Vampir, Assure

# Parallel Development Tools Limitations

---

- Many tools are specific to only one platform or vendor
- They do not interoperate, and never will
  - No integrated UI
  - No ability to share data
  - Functionality limited to that provided by tool
- They do not scale
  - Fine for 1990's machines
  - New machines will have 10,000+ processors
- Few high quality open-source tools in wide use
  - Increases difficulty of adopting new architectures
  - Can lead to vendor lock-in

# Parallel Development Tools

## Industry Best Practice (for everyone else)

---

- Integrated development environment (IDE)
  - Combines editor, compiler, debugger and other tools into a single consistent user interface
- Integrated management
  - Change control, build management, software quality policies
- Integrated testing
  - Automated unit testing, verification and validation activities
- Integrated documentation
  - On-the-fly documentation generation

# Parallel Development Tools

## Why Change?

---

- Reinforce good software engineering practices
- Strengthen auditing
- Enhance work-flow
- Increase productivity
- Improve documentation
- Reduce time-to-delivery

= Reduced development costs

# 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 Development Tools

## How will developer productivity be improved?

---

- Eliminates cost of switching between tools
- Uniform user interface for consistency and ease of use
- All the tools necessary to do the job on hand
- Team activities are integrated
- Tools can interoperate, sharing data and functionality
- Many manual activities (such as documentation) can be automated

= Improved work flow in development cycle

# Parallel Development Tools

## How will end-user productivity be improved?

---

- Hides complexity of parallel systems
- Generic interface to job scheduler/runtime system
- Visual indication of system and job status
- Can utilize functionality that will benefit end-user
- Opportunity to simplify data management
- Can be used to facilitate activities such as v&v

= Simplified interaction with parallel machines



# Eclipse Foundation History

---

- Originally developed by Object Technology International (OTI) and purchased by IBM for use by internal developers
- Released to open-source community in 2001, managed by consortium
  - Eclipse Public License (EPL)
  - Based on IBM Common Public License (CPL)
- Consortium reorganized into independent not-for-profit corporation, the Eclipse Foundation, in early 2004
  - Participants from over 85 companies

# Eclipse Foundation Strategic Members

---

- Actuate Corporation
- BEA
- Borland
- Computer Associates
- Hewlett Packard
- IBM
- Intel
- MontaVista Software
- SAP AG
- Scapa Technologies
- Serena Software
- Sybase, Inc.
- Wind River

# Eclipse Foundation

## Members

---

- Accelerated Technology
- Acucorp
- Advanced Systems Concepts
- Agitar
- Aldon
- Aonix
- AvantSoft
- Catalyst Systems Corporation
- CollabNet
- Compuware
- Discovery Machine
- Embarcadero Technologies
- ENEA
- Ericsson
- ETRI
- Exadel
- Fujitsu
- Genuitec
- Hitachi
- ILOG
- INNOOPRACT
- Inpriva
- Instantiations
- JBoss
- Kinzan
- Klocwork
- Logic Library
- Lombardi Software
- M1 Global
- M7 Corporation
- Mercury
- META-1
- Micro Focus
- MKS
- mValent
- Novell
- NTT
- Comware
- OC Systems
- Omondo
- Optena Corporation
- Oracle
- PalmSource
- Panscopic
- Parasoft Corporation
- PureEdge
- QNX Software Systems
- Real-Time Innovations
- Red Hat
- SAS
- SlickEdit
- Soft Landing Systems
- Teamstudio
- Telelogic
- Tensilica
- Texas Instruments
- THALES
- TimeSys
- Unisys
- VA Software
- Wasabi Systems
- webMethods

# Eclipse Foundation

## Commercial Tools Based on Eclipse

---

- Exadel Struts Studio and JSF Studio
- Genuitec MyEclipse
- IBM WebSphere Studio
- Intel C++ Compiler 8.1 for Linux
- Kinzan Studio
- M7 NitroX
- Mentor Graphics Nucleus Edge
- Monta Vista Dev Rocket
- Novell/SuSE SDK
- PalmOS Dev Suite
- Parasoft Jtest
- PureEdge Designer
- QNX Momentics
- Red Hat Developer Suite
- SAP NetWeaver Studio
- Tensilica Xtensa Xplorer IDE
- TimeSys TimeStorm IDE
- Wind River Workbench

# Eclipse Foundation

## Example Applications of Eclipse

---

- University of Washington
  - Integrated land use, transportation and environmental modeling
- Electronics and Telecommunications Research Institute
  - Embedded software development tool
- DaimlerChrysler
  - Business solution for processing geographic data (GDF)
- Hewlett Packard
  - Web services development
  - Software development kits
- Bank SinoPac
  - Model-driven business process integration and management

# Eclipse Foundation

## Eclipse Robustness?

---

- Used for key commercial applications
  - Reliability is essential
- Designed for very large projects
  - Regularly used to compile Linux kernel (~6M lines of code)
  - Legacy COBOL programs contain 10's of thousands of variables
- Large support base
  - Currently in excess of 50 FTEs working on core, 100's on other components
  - Over 85 companies contributing resources
  - Global open-source community

# Parallel Tools Platform

# 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
4. Exploit enhanced capabilities to develop a new generation of parallel tools



# Parallel Tools Platform Benefits

---

- Not just “another parallel tool”
  - A commercial quality framework for integrating existing and new tools
  - Designed for scalability and reliability
- Provides a strategy for tool development
  - Combats the existing proliferation of tools
  - A focal point for future tool development
- In the long run, saves money
  - Reduces work required to build tools
  - Reinforces software engineering practices
  - Simplifies adoption of new architectures
  - Enhances work flow, and hence productivity

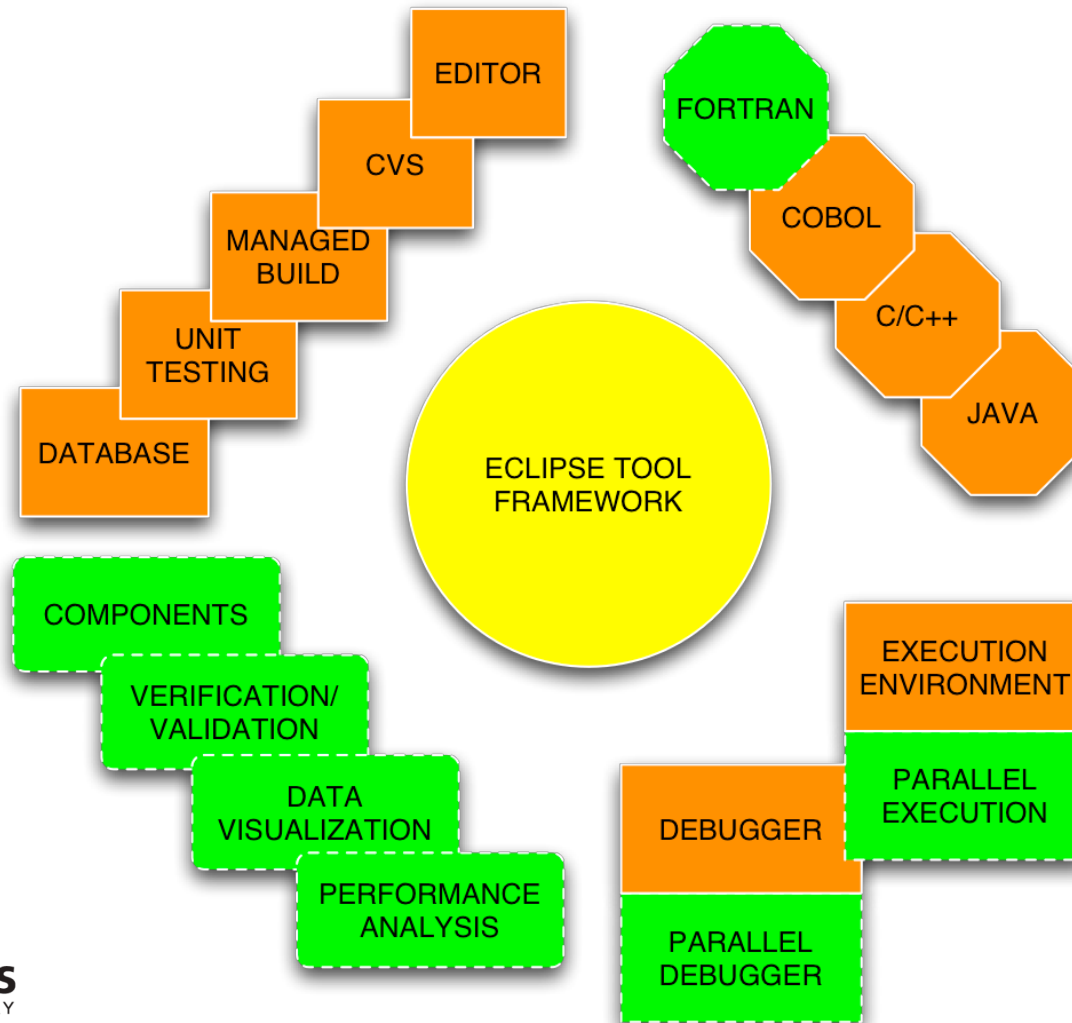
# Parallel Tools Platform Project Supporters

---

- Altera Corporation
- AWE, PLC
- Etnus, LLC
- IBM Research
- Intel Corporation
- ITACA
- Louisiana State University
- Monash University
- Open HPC Inc.
- Open MPI
- Rice University
- Scalable Systems Pte Ltd
- Technische Universitaet Muenchen
- Terra Soft Solutions
- University of Oregon
- University of Tennessee

# Parallel Tools Platform Components

---

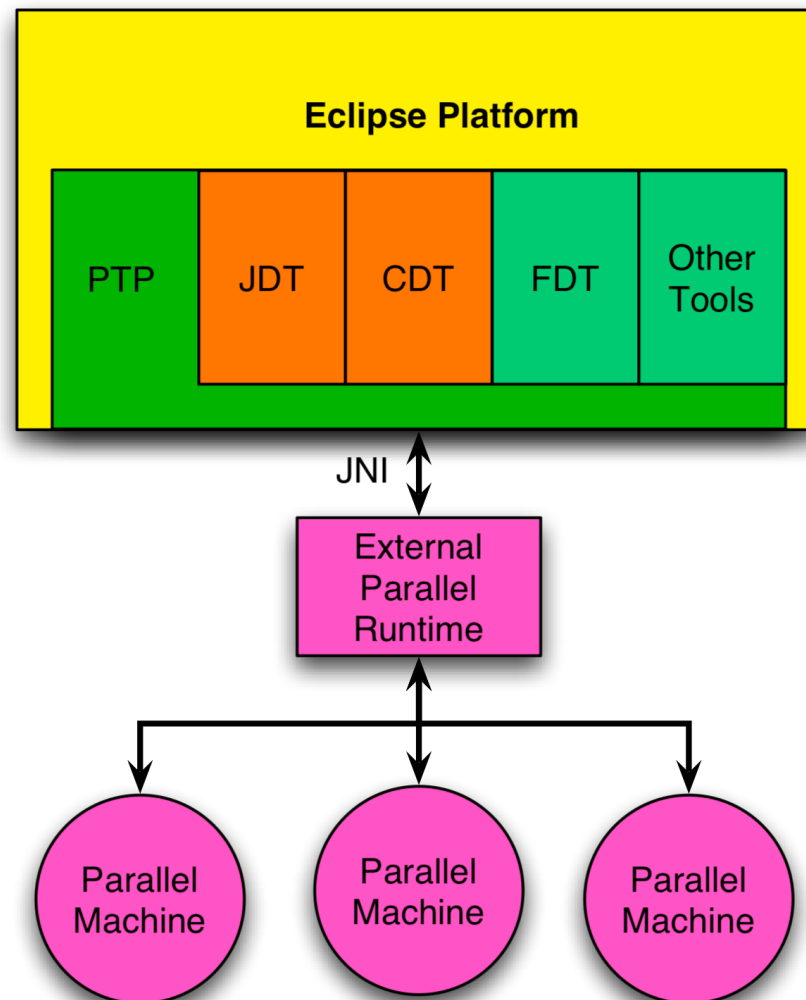


# Parallel Tools Platform Components

---

- Parallel Execution Environment
  - Extends existing execution environment to support parallel programs
- Parallel Debugger
  - Adds parallel debugging support to Eclipse
- Tools Integration
  - Support the integration of a variety of parallel tools, e.g. performance, verification, visualization, components
- Fortran
  - Adds Fortran support to a similar level as C/C++

# Parallel Tools Platform Architecture

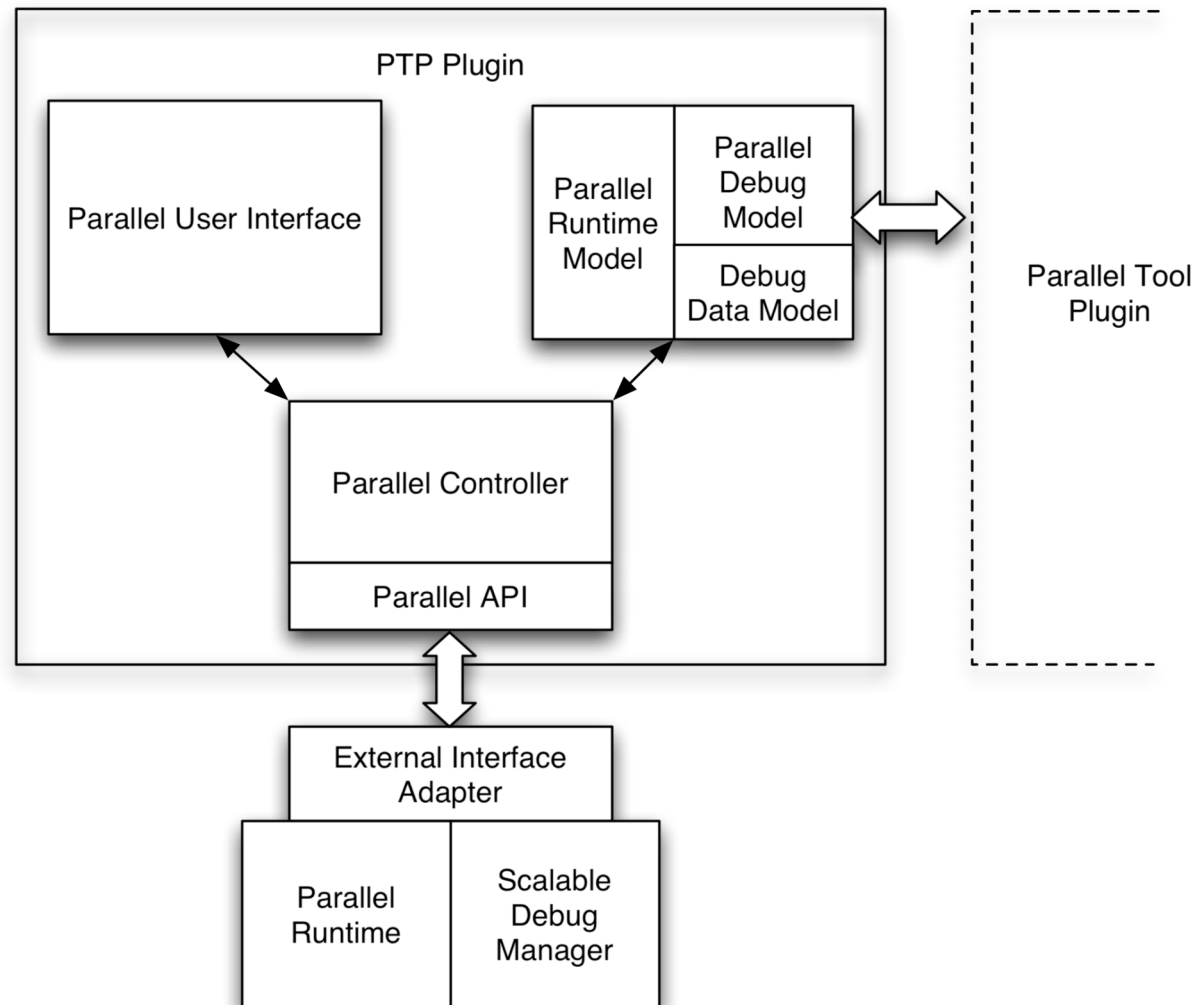


# Parallel Tools Platform Architecture

---

- 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
  - Initially target OpenRTE (independent component of OpenMPI)
  - Will support other runtime systems (requires additional work)

# Parallel Tools Platform Plugin Detail

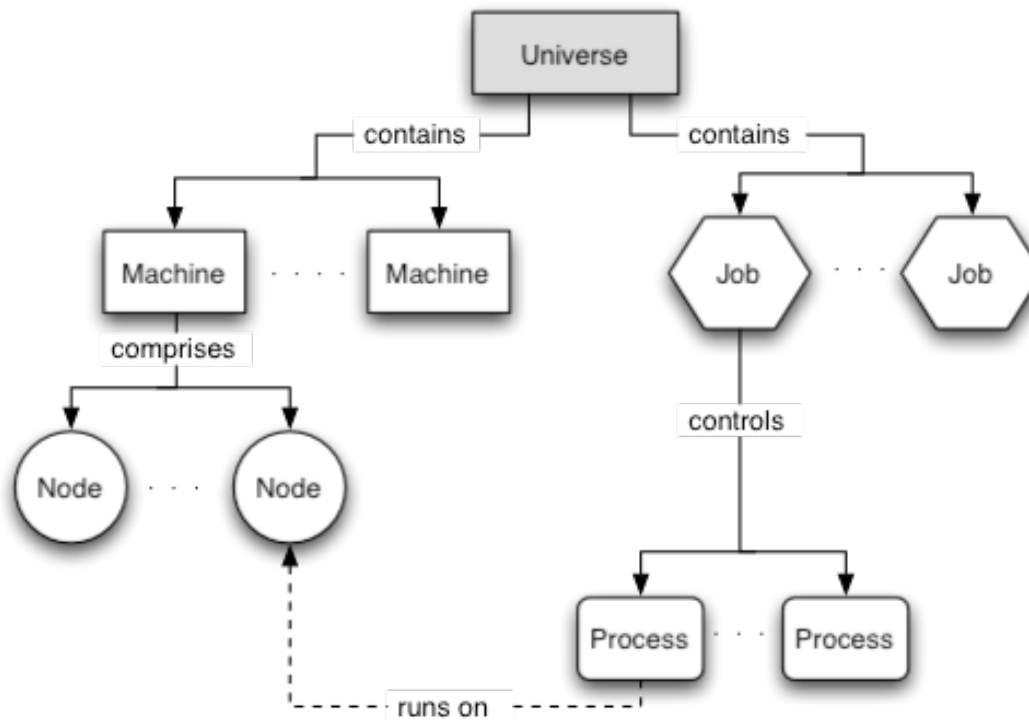


# Parallel Tools Platform

## Parallel Model

---

- Runtime Model
  - Defines notion of *universe*, parallel machines, jobs, processes, etc.





# Parallel Tools Platform

## Parallel Model

---

- Debug Model
  - Extends platform/CDT debug model to support parallel processes
  - Support for group operations
  - Scalable event management
- Debug data model
  - Efficient data handling and internal data manipulation
  - Data representation is not UI centric
  - Language independent (support for *arbitrary data types*)
  - Efficient conversion to/from Java native types
  - Efficient manipulation in intermediate form
  - Value caching and lazy evaluation

# Parallel Tools Platform

## Parallel Controller

---

- Controls interaction between PTP components and external runtime
- Manages debugger specific operations and event handling

# 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, array viewing, and new visualization tools
- Launch Configuration
  - Specify resource requirements (e.g. number of processes, execution time, etc.) and interface to launch services

# 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

---

- External framework
- Manages scalable and efficient debugging of enormous parallel programs
  - Launching processes under debug control
  - Communication with large numbers of processes
  - Efficient data transfers
  - Event management

# Parallel Tools Platform

## Tool Integration

---

- Runtime Services
  - A range of 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
- Tool Specific Services
  - Still being defined

# 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
  - Simplify interaction with runtime/job scheduler
- Allow user to concentrate on doing science, rather than how to run parallel applications

# Parallel Tools Platform

## Fortran Development Tools

---

- Still the predominant language for parallel scientific computing
- Support is essential for adoption of Eclipse in the parallel computing community
- Moving target:
  - Many standards: 66, 77, 90, 95, 2003
  - Substantial differences between versions
  - Then there's the extensions: HPF, Cray, VAX, IBM, KAP, LS, etc.
- Advanced features such as code refactoring will be available



# Parallel Tools Platform

## Future Work

---

- New parallel tools
  - What new possibilities become available when tools are tightly integrated?
- Advanced debugging
  - Can the close integration of tools and debuggers lead to new debugging techniques?
- Lightweight tools
  - Can the developer or end-user benefit from an array of simple, useful tools?

# Conclusion

---

- Unique opportunity to move parallel development to best practice
- Designed to address scalability and performance issues from the beginning
- Potential to push the development of parallel tools into new areas
- Growing interest from the parallel scientific computing community
- Seeking your support and involvement!