The Eclipse Parallel Tools Platform Project

Creation Review

18 February 2005

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
Main Components

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

Parallel Tools Platform
FY05 LANL Resources

- Greg Watson (PTP Lead)
- Nathan DeBardeleben (Parallel Execution)
- Craig Rasmussen (Fortran - 50%)
- Jeff Brown (Parallel Debugger Design - 25%)
- Contractor (Parallel Debugger)

Parallel Tools Platform Interested Parties

PTP - Parallel Tools Platform
PEX - Parallel Execution Environment
PDB - Parallel Debugger
PTI - Parallel Tools Integration
FDT - Fortran Development Tools

What	Who	Interested	Potential Committer?
PTP	Altera	Y	
PTP	AWE		Y
PDB	Etnus	Y	
PDB, FDT	Intel		Y
PTP	ITACA	Y	
PTP	LLNL	Y	
PDB, PEX	Monash University		Y
PTP	OpenHPC	Y	
PTP	OpenMPI	Y	
PTI, FDT	Rice University		Y
PTI, FDT	TUM	Y	
PTP	Terra Soft Solutions	Y	
PTI	University of Oregon		Y
PEX, PDB, PTI	University of Tennessee	Y	

Parallel Tools Platform
FY05 Development Plan

Parallel Execution Environment
Parallel Debugger
Parallel Tool Integration
Fortran Development Tools

3/05	Design and implement parallel runtime model Initial Fortran language plugin (tool chain, managed build, Fortran nature, Fortran perspective)
4/05	Design and implement high level parallel API to interface to external runtime Design and implement parallel debug model Managed build support for performance analysis tool
6/05	Design and implement parallel runtime UI components Design and implement parallel data model
7/05	Integrate parallel debug model and parallel data model into platform debug model Automatic instrumentation for performance analysis tool
8/05	Design and implement parallel controller Design and implement scalable debug manager, debugger UI components Extended Fortran language plugin (debugging, syntax highlighting) UI for performance analysis tool
9/05	Version 1.0 “Friendly user” deployment

Parallel Tools Platform
FY06 Development Plan Milestones

- Additional functionality in parallel runtime (generic resource scheduler interface, connect/disconnect from running program, etc.)
- Extend parallel debugger to support complete set of debug services, provide parallel-specific data views (array viewer, etc.)
- Provide a minimal implementation that supports a simplified parallel runtime interface for the end-user
- Extend Fortran integration to support indexing, searching and code completion
- Integration with at least one existing parallel tool

Parallel Tools Platform
FY07 Development Plan Milestones

- Support for future parallel architectures
- Advanced debugging features such as relative debugging, automatic debugging, visualization
- Complete, simple, interface to parallel systems for end-users, including support for pre- and post-processing of data, visualization, etc
- Full Fortran parser integration with internal AST, support for most Fortran standards
- Support for AJAX, HPF, Co-array Fortran, and other data parallel languages
- A wide range of parallel tools integrated into Eclipse
- A new generation of parallel tools that utilize the integrated nature of the Eclipse Framework

Parallel Tools Platform Accomplishments

9/04	Demonstrated compiling, launching, and monitoring a parallel program using Eclipse
10/04	Demonstrated attaching Eclipse debugger to single parallel process running on a cluster
11/04	Project declaration
1/05	Draft design document published
2/05	Runtime model completed
2/05	First Fortran program compiled using managed build system