Integrating a tool with the Eclipse UI using the ETFw

Wyatt Spear
wspear@cs.uoregon.edu
University of Oregon

PTP User-Developer Workshop, Sept 18-20,2012

Introduction

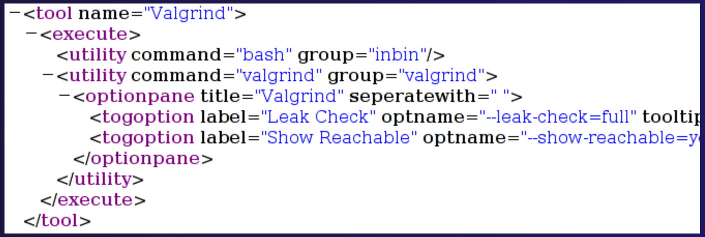
- Objective
  - Become familiar with the procedure for wrapping external command-line based utilities in the Eclipse/PTP UI
- Contents
  - Overview of ETFw and TAU
  - Description of ETFw workflow definition format
  - Background implementation details
  - Overview of continuing work

PTP/External Tools Framework

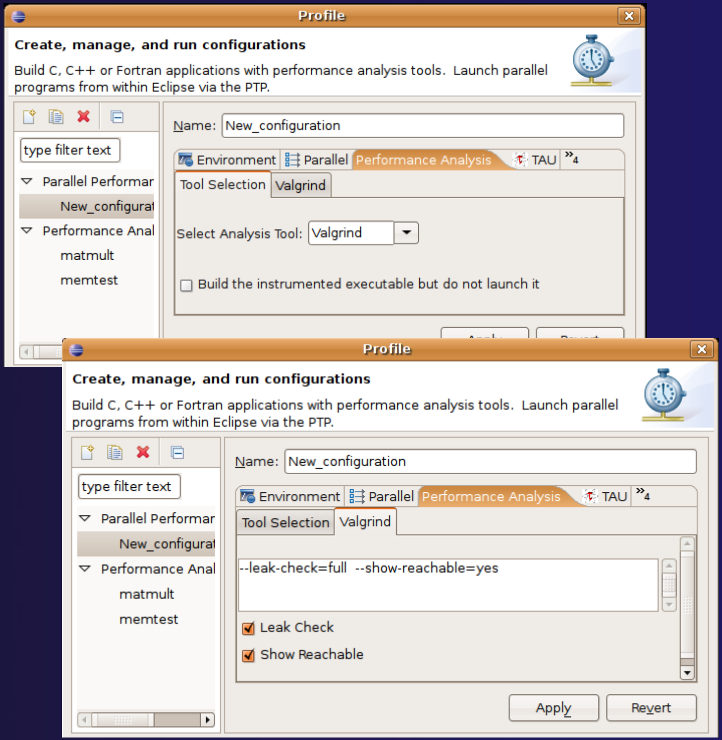
formerly "Performance Tools Framework"

Goal:

- **Reduce the "eclipse plumbing" necessary to integrate tools**
- Provide integration for instrumentation, measurement, and analysis for a variety of performance tools
    - Dynamic Tool Definitions: Workflows & UI
    - Tools and tool workflows are specified in an XML file
    - Tools are selected and configured in the launch configuration window
    - Output is generated, managed and analyzed as specified in the workflow
    - One-click 'launch' functionality
    - Support for development tools such as TAU, PPW and others.
    - Adding new tools is much easier than developing a full Eclipse plug-in



```
-<tool name="Valgrind">
 -<execute>
    <utility command="bash" group="inbin"/>
  -<utility command="valgrind" group="valgrind">
    -<optionpane title="Valgrind" seperatewith=" ">
        <togoption label="Leak Check" optname="--leak-check=full" tooltip
        <togoption label="Show Reachable" optname="--show-reachable=y
     </optionpane>
    </utility>
  </execute>
</tool>
```



Profile

Create, manage, and run configurations

Build C, C++ or Fortran applications with performance analysis tools. Launch parallel programs from within Eclipse via the PTP.

Name: New_configuration

Select Analysis Tool: Valgrind

☐ Build the instrumented executable but do not launch it

Profile

Create, manage, and run configurations

Build C, C++ or Fortran applications with performance analysis tools. Launch parallel programs from within Eclipse via the PTP.

Name: New_configuration

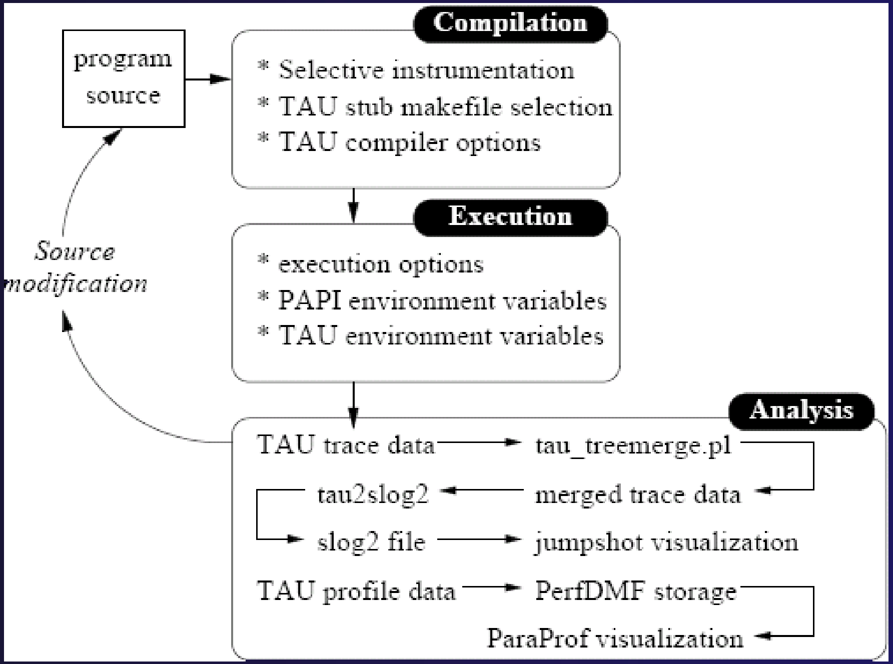
--leak-check=full --show-reachable=yes

☑ Leak Check

☑ Show Reachable

TAU Integration with PTP

- TAU: Tuning and Analysis Utilities
  - Performance data collection and analysis for HPC codes
  - Numerous features
  - Command line interface
- The TAU Workflow:
  - Instrumentation
  - Execution
  - Analysis



Generalized Solution: ETFw

- Describe individual steps/applications in tool workflow as compilers or utilities
  - Individual utilities have arguments, either hard coded or with tool-pane UI elements
  - Tool input can be customized between files
  - UI definitions relatively trivial
- Workflow sequences can be arbitrarily complex
  - Multiple build/execute/analysis steps
  - Preliminary support for logical branching

ETFw Examples/Resources

- TAU examples in <tau2>/tools/srs/eclipse or http://nic.uoregon.edu/~wspear/etfw_tool_xml
- TAU Plugin example in PTP GIT repository: org.eclipse.ptp.etfw.tau/toolxml
- PTP Wiki: http://wiki.eclipse.org/PTP/ETFw/PTP_External_Tools_Framework

ETFw Internal Structure

- Individual tools parsed into Build/Exec/PostProc tools with associated ToolPane UI elements
- UI element settings populate launch configuration values
- At launch individual utility elements are iterated through, arguments are populated from launch configuration and the final adjusted commands are issued in sequence

ETFw Extension Points

- org.eclipse.ptp.etfw.dataManagers
  - Custom definition of data-management and post-processing operations
  - Also useful for defining arbitrary commands to run at any time during the workflow
  - Accessible from custom workflow definitions
- org.eclipse.ptp.etfw.workflows
  - Used to include a workflow xml inside a plugin
- org.eclipse.ptp.etfw.toolUITabs
  - Defines an individual ui tab associated with a tool
- org.eclipse.ptp.etfw.configurationTabs
  - Deprecated
  - Used to define a top-level tab containing a tool's UI

Coming Attractions

- Better integration with editor
  - Source highlighting/markup
  - Generalized source tree context capabilities
- Generalized PAPI hardware counter functionality
- Improved scaling study support