

The External Tools Framework, Performance Tuning and Analysis Tools

Wyatt Spear, University of Oregon

Tools Demo + Q&A

PTP/External Tools Framework

formerly "Performance Tools Framework"

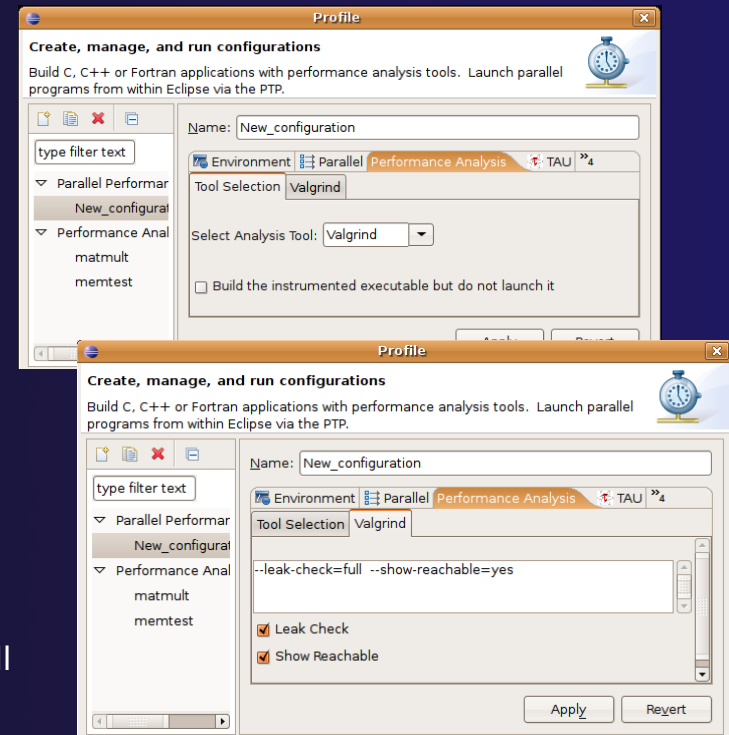
Goal:

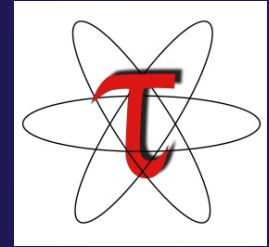
- ✦ Reduce the "eclipse plumbing" necessary to integrate tools
- ✦ Provide integration for instrumentation, measurement, and analysis for a variety of performance tools
 - ✦ Dynamic Tool Definitions: Workflows & UI
 - ✦ Tools and tool workflows are specified in an XML file
 - ✦ Tools are selected and 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" separatewith=" ">
        <togoption label="Leak Check" optname="--leak-check=full" tooltip="Leak Check" />
        <togoption label="Show Reachable" optname="--show-reachable=yes" tooltip="Show Reachable" />
      </optionpane>
    </utility>
  </execute>
</tool>

```





PTP TAU plug-ins

<http://www.cs.uoregon.edu/research/tau>

- ★ TAU (Tuning and Analysis Utilities)
- ★ First implementation of External Tools Framework (ETFw)
- ★ Eclipse plug-ins wrap TAU functions, make them available from Eclipse
- ★ Full GUI support for the TAU command line interface
- ★ Performance analysis integrated with development environment

The image displays several screenshots from the Eclipse IDE illustrating the TAU configuration and analysis process:

- Configuration Dialogs:** Two screenshots show the 'Create, manage, and run configurations' dialog. The first shows the 'Analysis Options' tab with 'Tau Compiler' and 'Selective Instrumentation' selected. The second shows the 'PAPI Counters' dialog where various PAPI counter categories like 'PAPL_L1_DCM', 'PAPL_L1_TCM', and 'PAPL_L1_LDM' are being selected.
- Code Editor:** A screenshot shows a C++ code snippet for an MPI barrier function:


```
MPI_Barrier(MPI_COMM_WORLD);
for (i=0; i<S; ++i) {
  if (me==0) {
    MPI_Send(field, SIZE, MPI_INT, 1, 4711, MPI_COMM_WORLD);
    MPI_Recv(field, SIZE, MPI_INT, proc-1, 4711, MPI_COMM_WORLD, &status);
  }
  else {
    MPI_Recv(field, SIZE, MPI_INT, me-1, 4711, MPI_COMM_WORLD, &status);
    MPI_Send(field, SIZE, MPI_INT, (me+1)%proc, 4711, MPI_COMM_WORLD);
  }
}
```
- Performance Analysis:** A screenshot shows a 3D heatmap visualization of performance data, with axes representing different variables and a color scale from blue (low) to red (high).
- IDE Context Menu:** A screenshot shows a context menu over a code block with 'Selective Instrumentation' highlighted, indicating the integration of TAU analysis into the development environment.