Developing Parallel Applications: The PTP's PLDT

Parallel Language Development Tools (PLDT):
Syntactic and Static Analysis
Of C/C++/Fortran code with the Parallel Tools Platform

Beth Tibbitts

IBM T.J. Watson Research Center

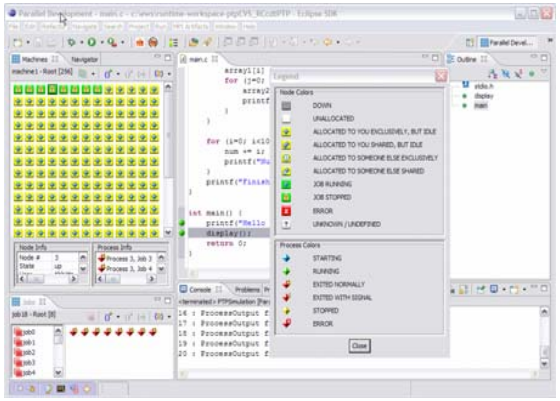
tibbitts@us.ibm.com

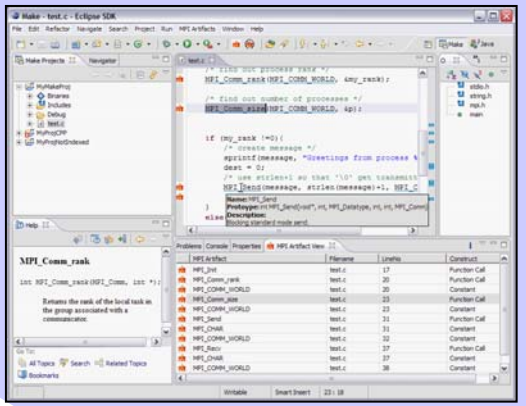
This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under its Agreement No. HR0011-07-9-0002

PTP: Parallel Tools Platform

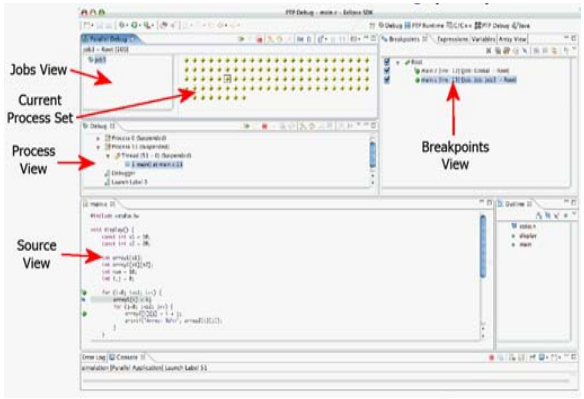
http://eclipse.org/ptp

Parallel Runtime



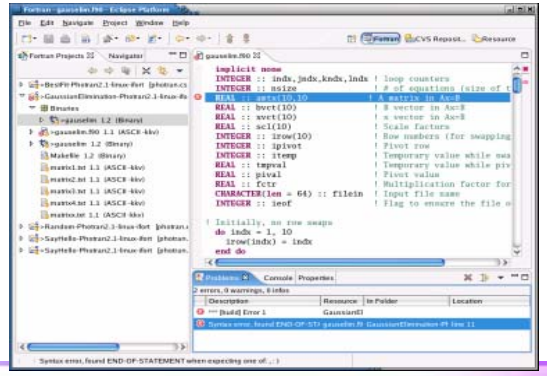


Parallel Language Dev. Tools (PLDT)



Parallel Debugger

Fortran Tools



Performance Tools*

Based on TuningFork:
http://www.alphaworks.ibm.com/tech/tuningfork

Based on
C/C++ Development Tools: CDT
http://eclipse.org/cdt

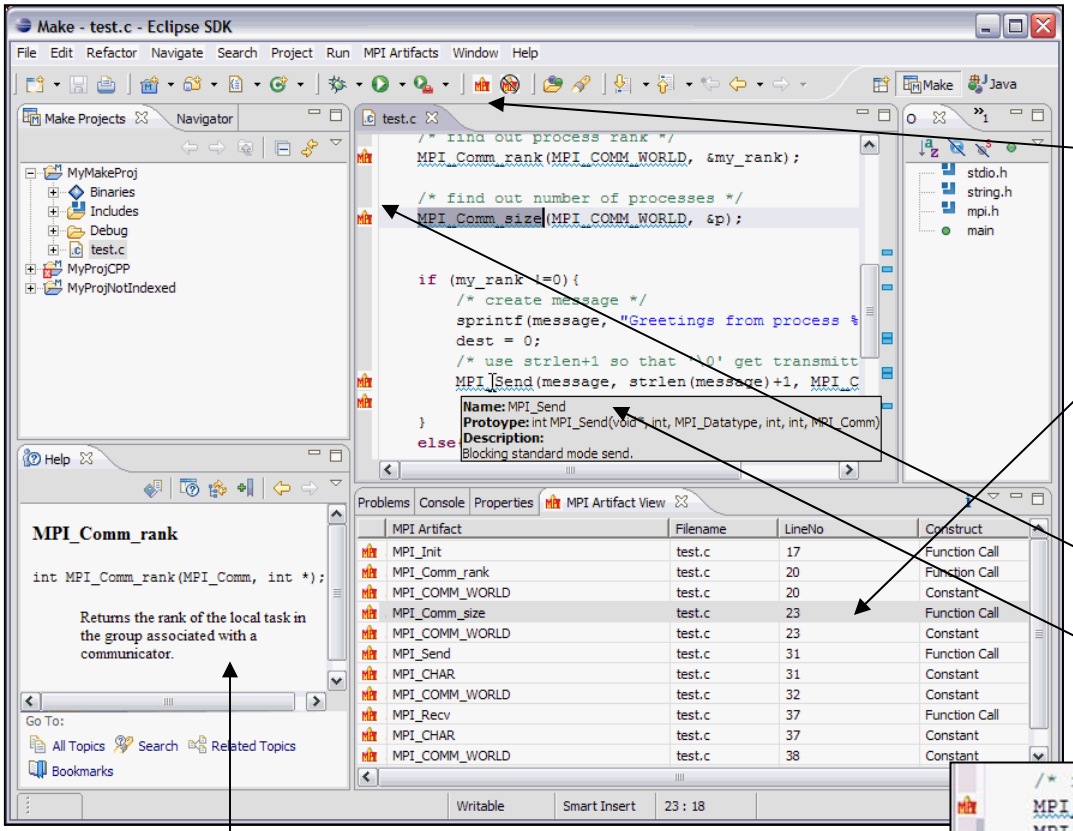
* Not yet publicly available on eclipse.org

Parallel Tools Platform
Parallel Language Development Tools

- PLDT is a subset of PTP, and can stand alone
  - Syntactic and static analysis of MPI and OpenMP code in C/C++/Fortran: general framework extensible for other parallel tools and languages
  - Simple development assistance (content assist, hover help, F1 Help view) similar to other eclipse projects
  - Some special parallel programming features:
    - Location of MPI Artifacts, with source code navigation
    - New Project wizard helpers
    - MPI Barrier Analysis
    - OpenMP Concurrency analysis
    - OpenMP Identification of common errors
    - Show #pragma regions in OpenMP
  - Some Marker views are easily reusable by other Eclipse plug-ins
- PTP Core requires Linux™ or MacOS (for OpenMPI); PLDT can run on Windows®

Parallel Language Development Tools: MPI Development Tools

Based on the CDT (C/C++ Development Toolkit),



Actions to find MPI Artifacts via Static analysis

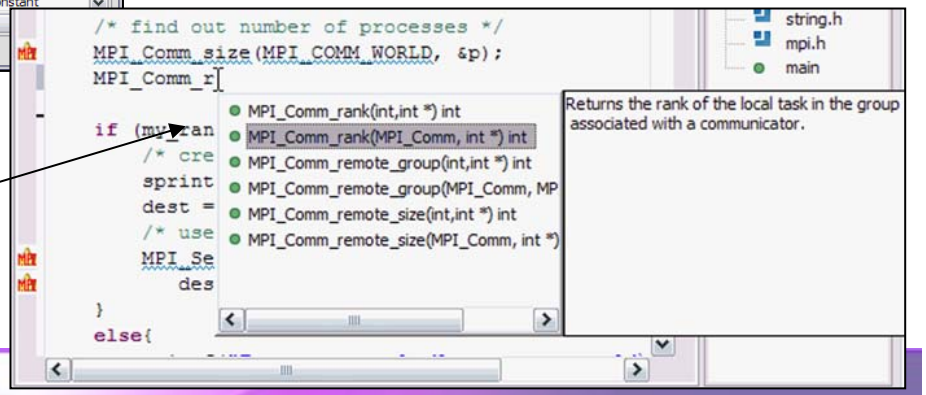
MPI Artifacts found by analysis

Source Markers for Navigation & ID

Mouse hover help And content assist

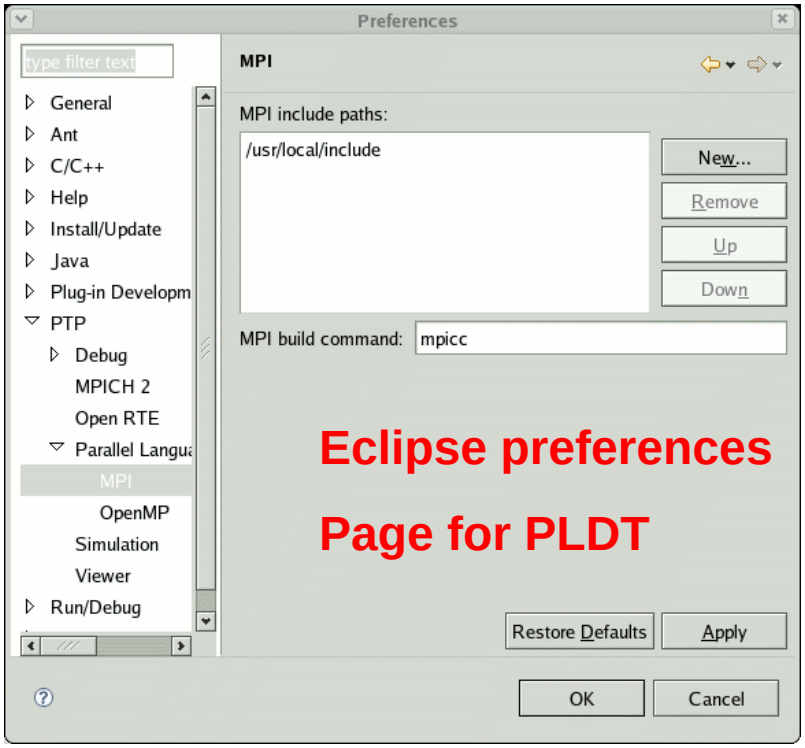
Context Sensitive Help (F1) provides API info

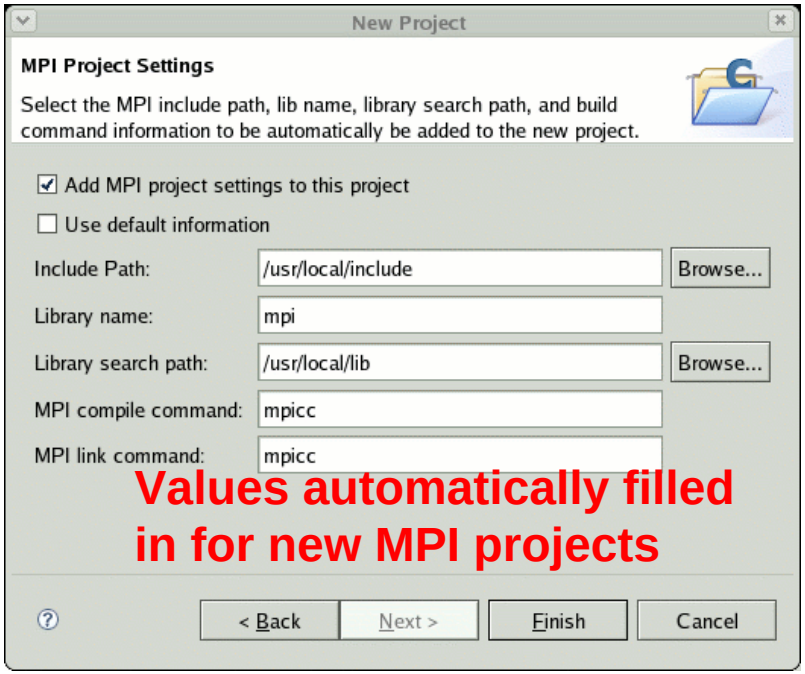
Content Assist Ctrl-space Suggests completions



Preferences / New MPI Project Wizard

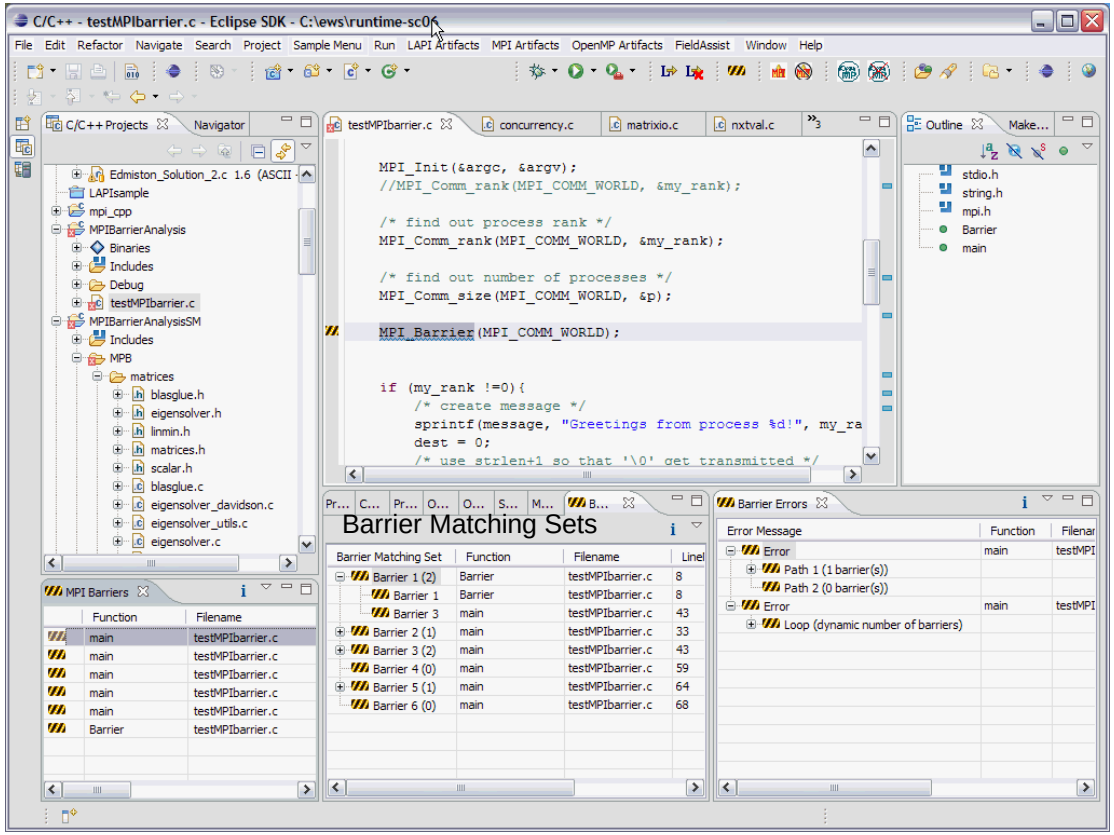
Set pointer to include path once, and new project setup is simplified.





Look for this in PTP 2.0

PTP PLDT: MPI Barrier Analysis



Verify barrier synchronization in C/MPI programs

- Interprocedural static analysis. Output is:

1) For verified programs, lists barrier statements that synchronize (match)

2) For synchronization errors, reports counter example that illustrates and explains the error.

CDT enhancements for This analysis:
Bottom-up AST traversal
Visitor: visit()
leave()

new

Reference: Yuan Zhang and Evelyn Duesterwald. Barrier matching for programs with textually unaligned barriers. In Proceedings of the Symposium on **Principles and Practice of Parallel Programming**, March 14-17, 2007, San Jose, CA

OpenMP Tools

OpenMP - Simple, Portable, Scalable SMP Programming An API for multi-platform shared-memory parallel programming in C/C++ and Fortran.

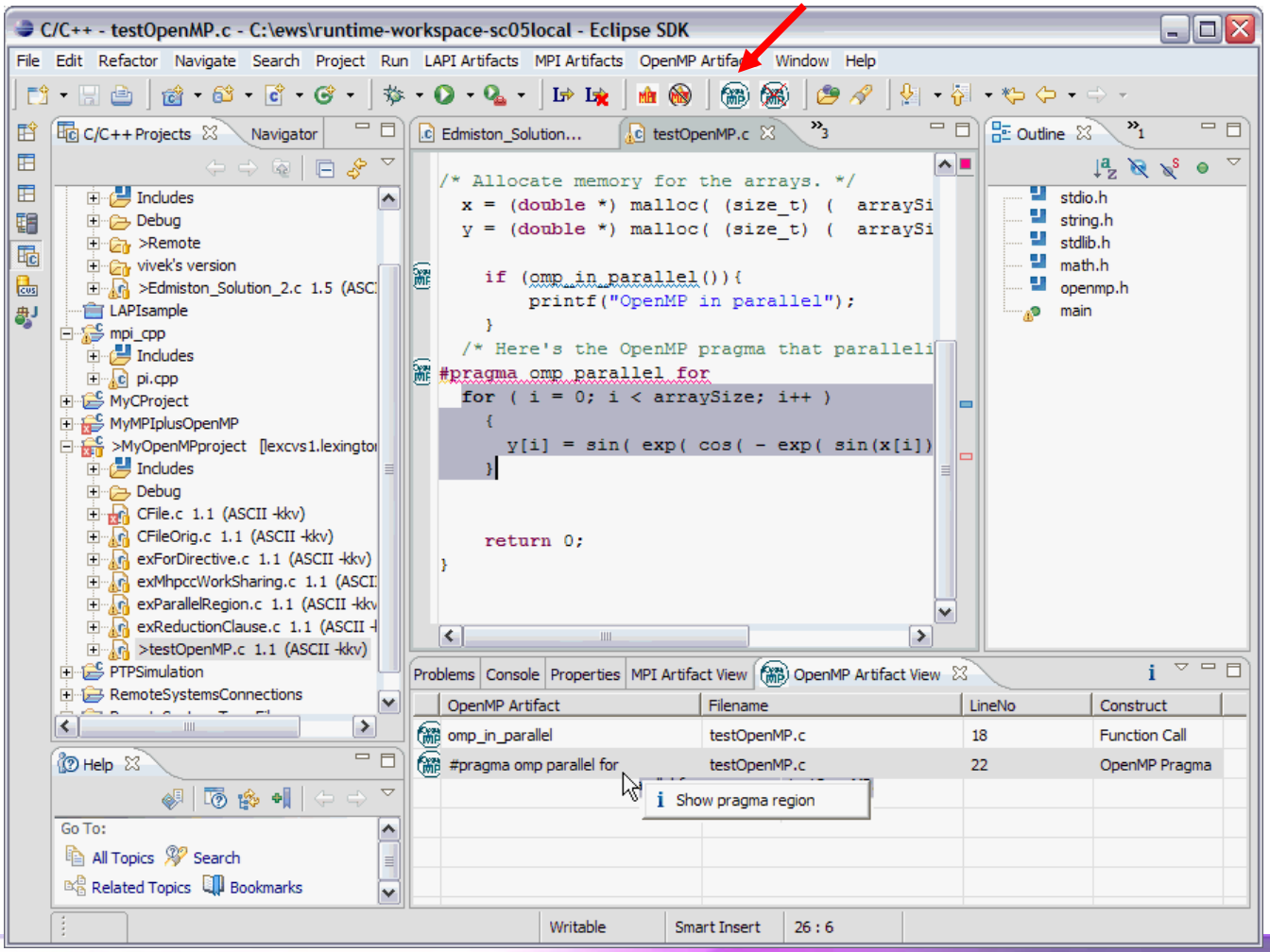
- Identify constructs
- List OpenMP constructs
- Link to source code
- Help: hover, content assist…

Analysis

- identify scope of #pragma

Next slides:

- Identify common problems
- Concurrency analysis

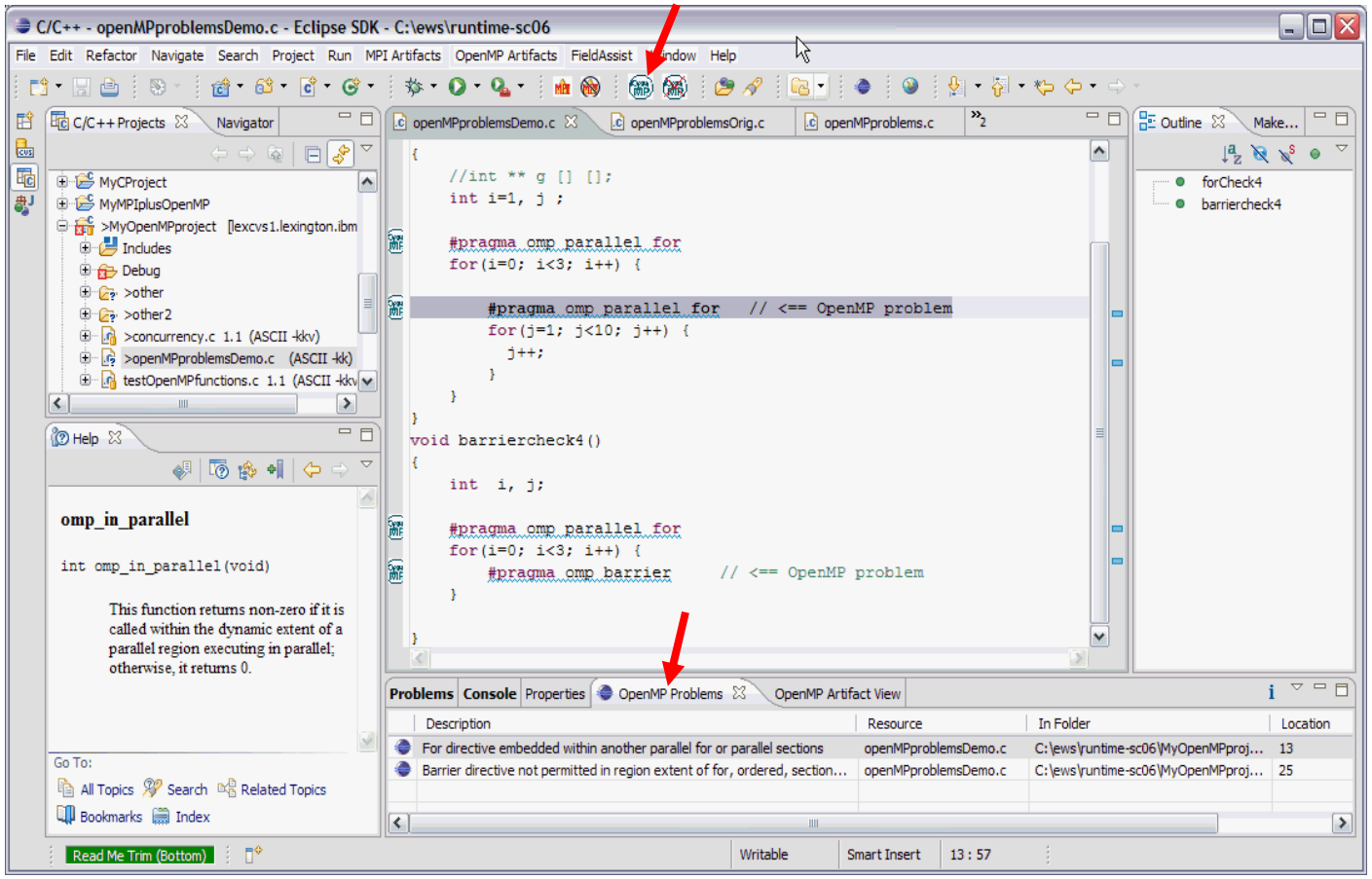


OpenMP Tools

OpenMP - Simple, Portable, Scalable SMP Programming An API for multi-platform shared-memory parallel programming in C/C++ and Fortran.

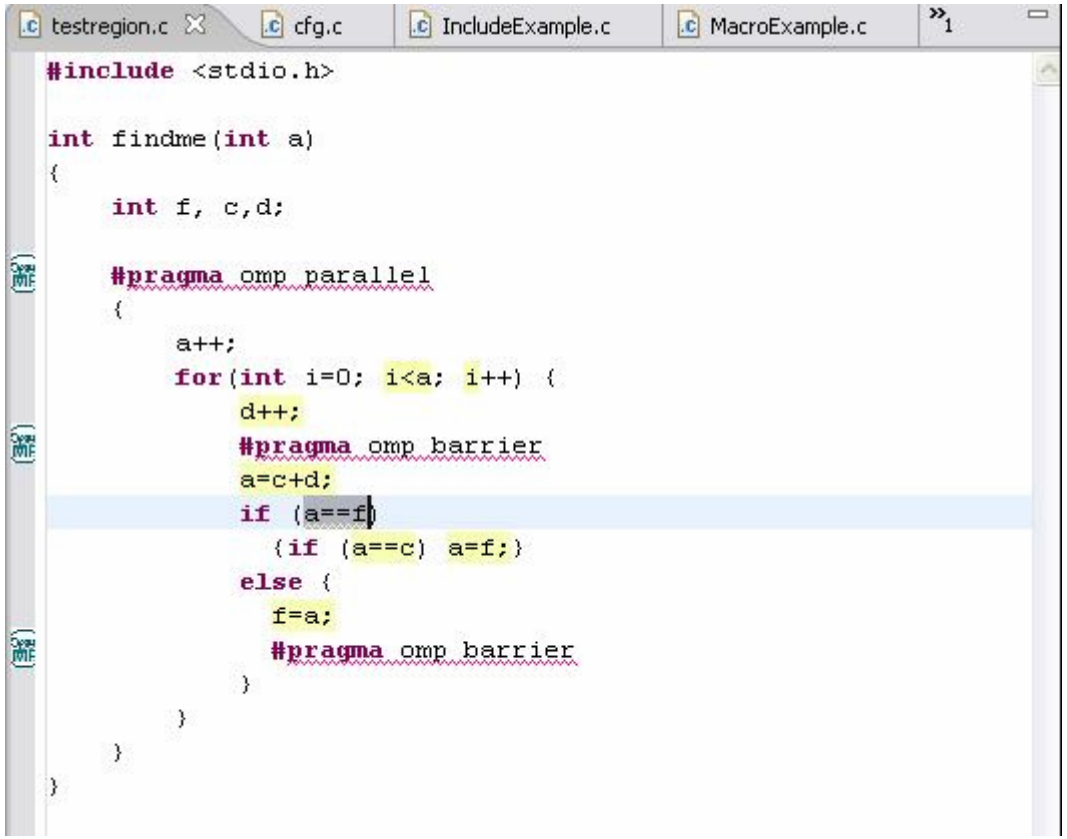
- **Identify common problems in OpenMP programming**

Control flow graph constructed for analysis



See next slide ….

OpenMP Concurrency Analysis



- Analysis of which statements could execute in parallel (based on concurrency analysis of Yuan Lin)

Possible future extension:

- Analysis to develop strategy for parallelizing

Need user feedback on OpenMP features especially!

Performance Visualization Tools

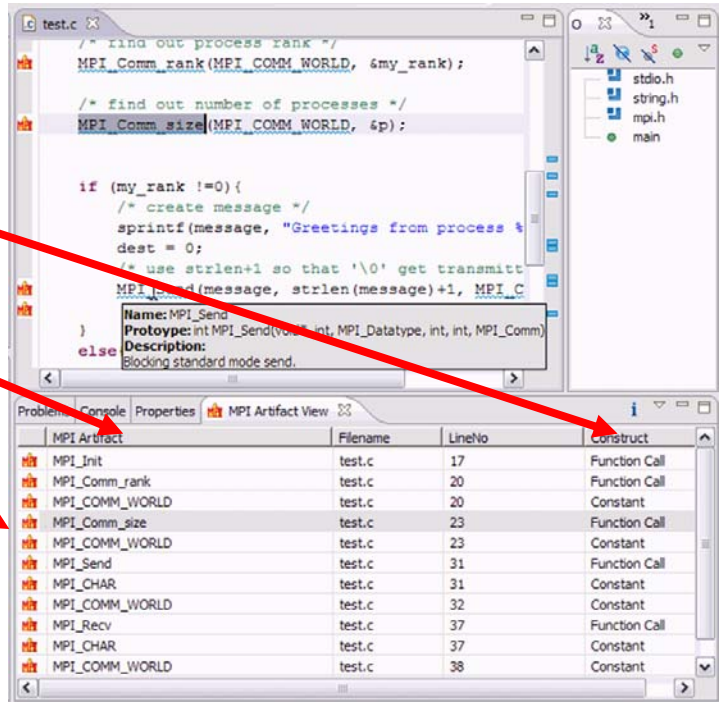
Being developed for visualization of performance trace data

Based on TuningFork: http://www.alphaworks.ibm.com/tech/tuningfork

Handy reusable view class: org.eclipse.ptp.pldt.common.SimpleTableMarkerView

```
public class MPITableView extends
SimpleTableMarkerView {
  public MPITableView() {
        super(MpiPlugin.getDefault(),
        "MPI Artifact",
        "MPI Artifacts","Construct",
        MpiIDs.MARKER_ID);
        }
}
```

(plus plugin.xml marker image info)



Shows marker instances in a view;
Provides source code navigation, sorting of marker info, resource change management, etc.

Hierarchical (2-level) version too for tree table

PTP-Related EclipseCon opportunities

- PTP and related Long Talks:
  - Parallel Tools Platform: Now and Future (Greg Watson, Wednesday 11:10 am, Grand Ballroom C)
  - C/C++ Source Code Introspection Using The CDT (Chris Recoskie and Beth Tibbitts, Wednesday 11:10 am, Great America Ballroom JK
- PTP-related Short Talks
  - *Developing Parallel Programs – PTP's PLDT (Beth Tibbitts, Wednesday 2:00 pm, Room 210)*
  - Performance Analysis of Parallel C/C++ and Fortran Applications in Eclipse (Wyatt Spear, Wednesday 10:50 am, Room 210)
  - HPCVision: An Interactive Tool for Scalable Analysis of Parallel Performance Profiles (Adam Bordelon, Thursday 1:30 pm, Room 207)
- PTP BOF: Tuesday, 7:30 pm, Room 209

Legal Notices

- IBM and alphaWorks are registered trademarks of International Business Corp. in the United States and other countries
- Windows is a registered trademark of Microsoft Corporation in the United States, other countries, or both.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- Other company, product, or service names may be trademarks or service marks of others

