

# An Integration of Dynamic MPI Formal Verification Within Eclipse PTP

Alan Humphrey, Chris Derrick, Ganesh Gopalakrishnan

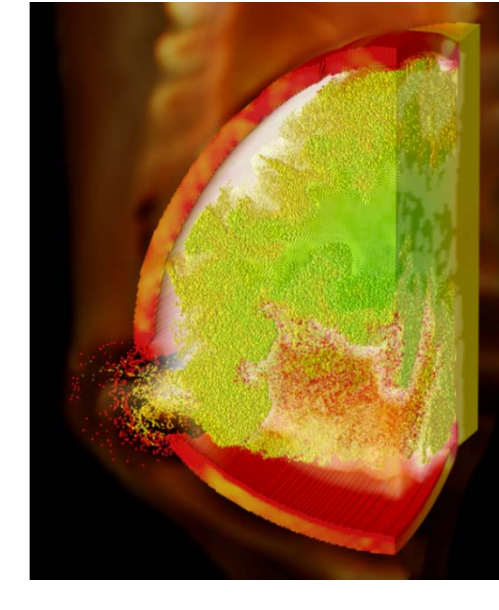
University of Utah School of Computing

## Problem / Motivation

The shift from sequential to parallel and distributed computing is of fundamental importance for the advancement of computing practices. Unfortunately, this makes debugging truly challenging, leading to incorrect simulation results.



IBM Roadrunner at Los Alamos National Laboratory

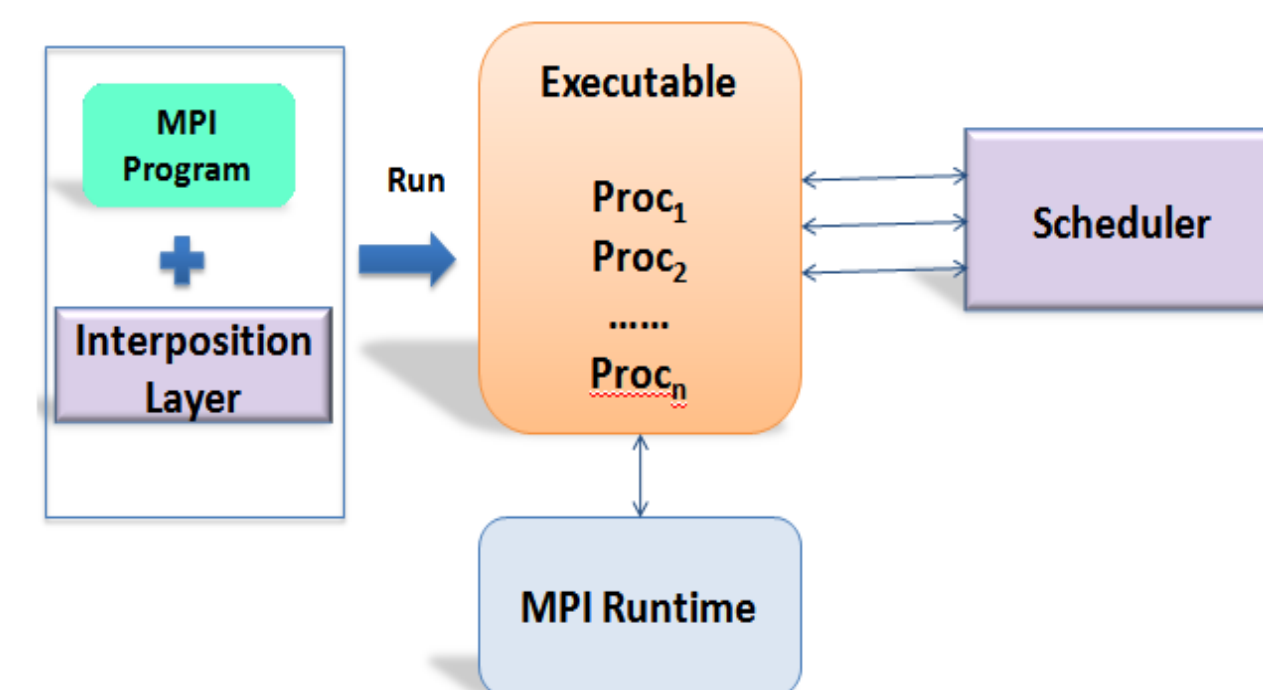


(Image courtesy of Steve Parker, CSAFE, Utah)

- Almost all large-scale parallel scientific simulation programs are written using MPI (Message Passing Interface); examples:
  - Earthquake simulations
  - Chemical reactions
- MPI semantics are complicated
- Difficult to verify
- Few integrated graphical debugging and analysis tools exist

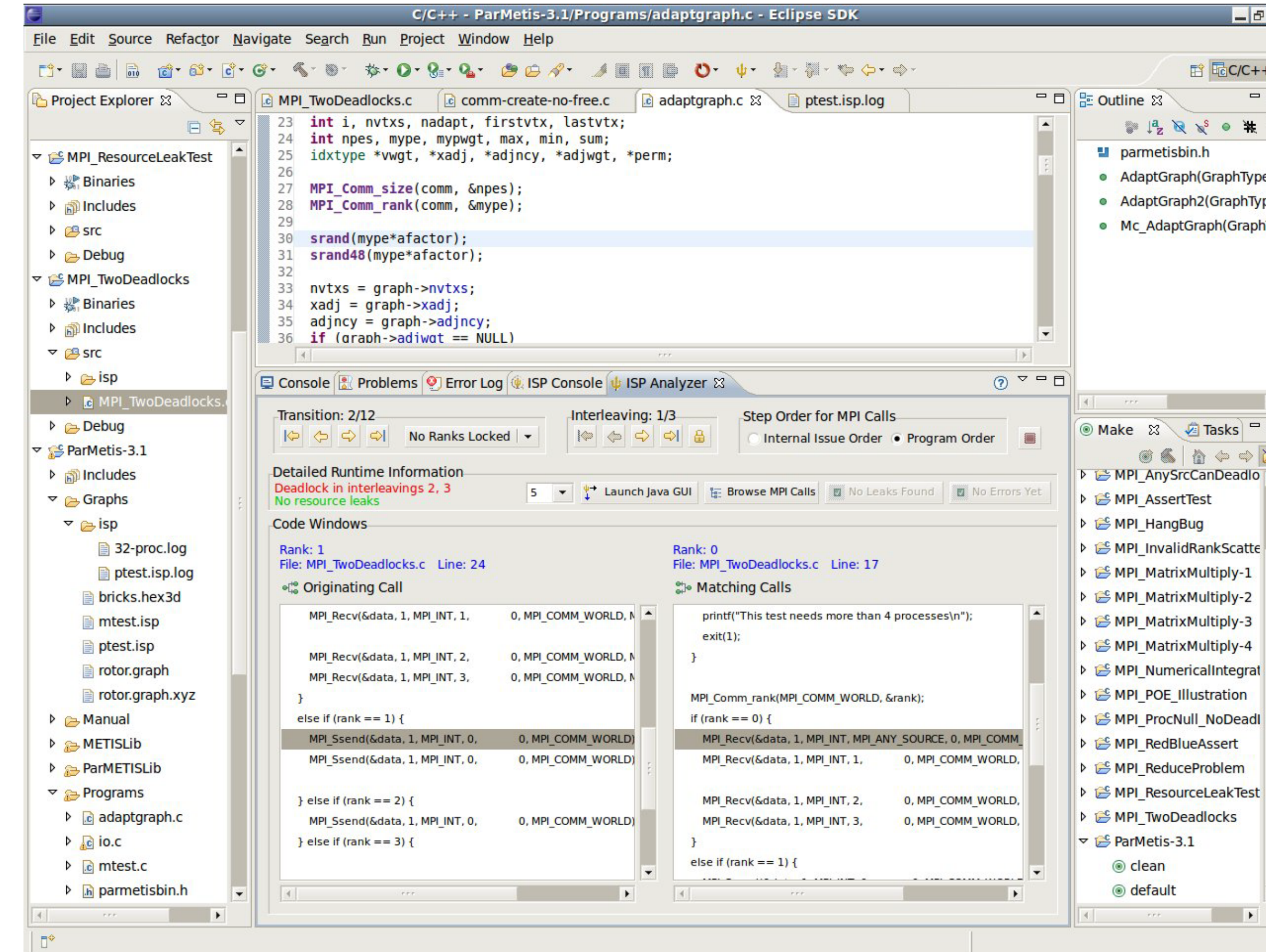
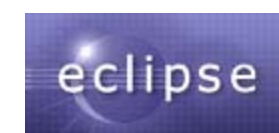
## Approach / Solution

Of particular importance will be the efficient dynamic verification of such parallel applications. This calls for running actual MPI applications under a formal verification scheduler to exercise all relevant interleavings.



Our research goals were to verify practical MPI programs for deadlocks, resource leaks, and assertion violations at the push of a button and be able to easily visualize the results. We also sought to integrate these capabilities with the Eclipse IDE via an Eclipse plug-in for the Parallel Tools Platform (PTP). We present here the result of our work, GEM – Graphical Explorer of MPI.

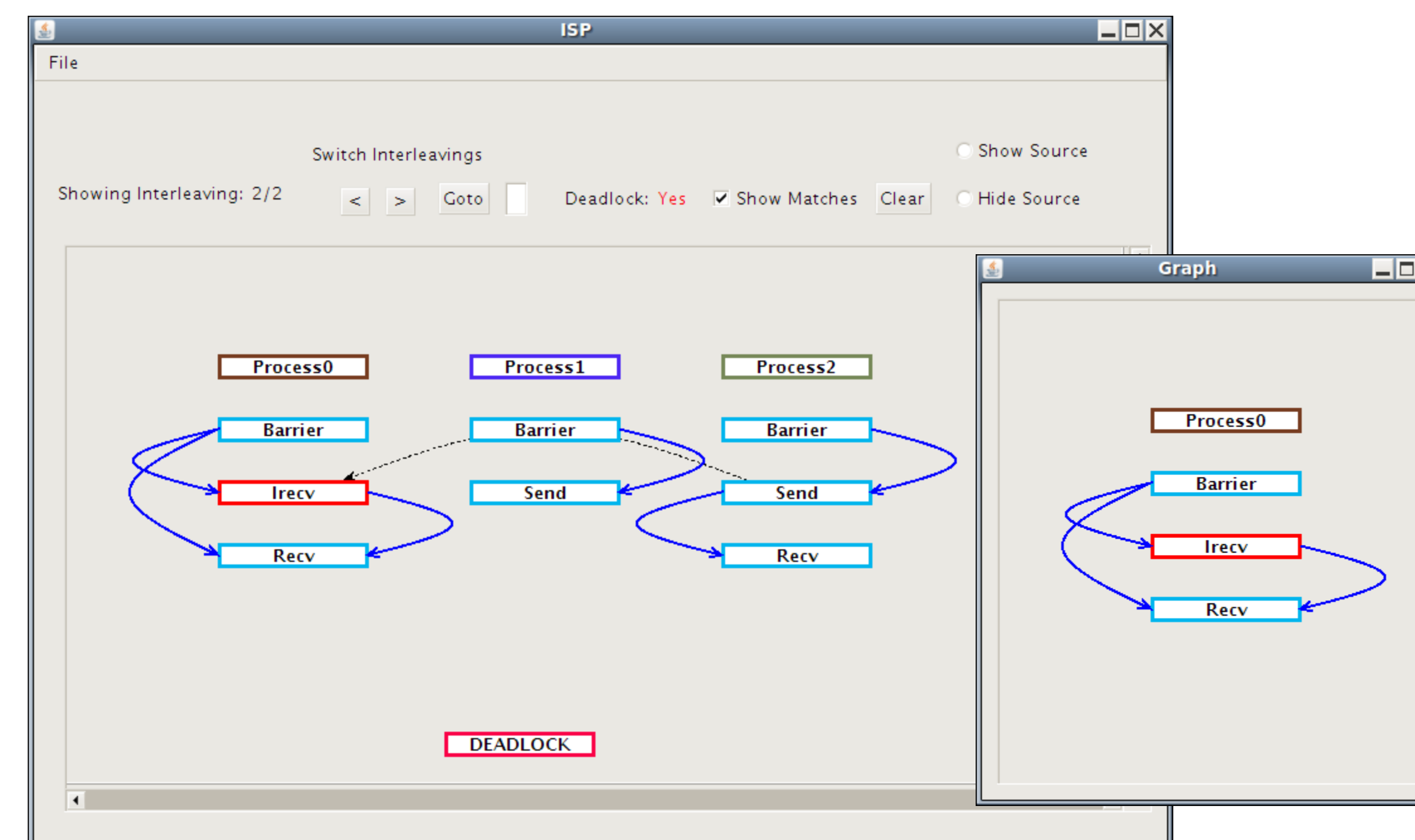
**GEM has been contributed to and is now a part of the Eclipse PTP as of the 3.0 release in December 2009**



### The Analyzer View provides the following features:

Step through the MPI calls in the order issued by ISP's scheduler or in the order the calls appear in the source code.

- Visually displays MPI out-of-order completion semantics
- Optionally focus in on a specific rank (process ID) with the rank lock feature
- Steps through MPI calls for all relevant schedules found by ISP
- Displays program source code and highlights specific MPI calls
- Highlights point-to-point (Send/Recv) and Collective matches



## ISPUI – Happens Before Viewer

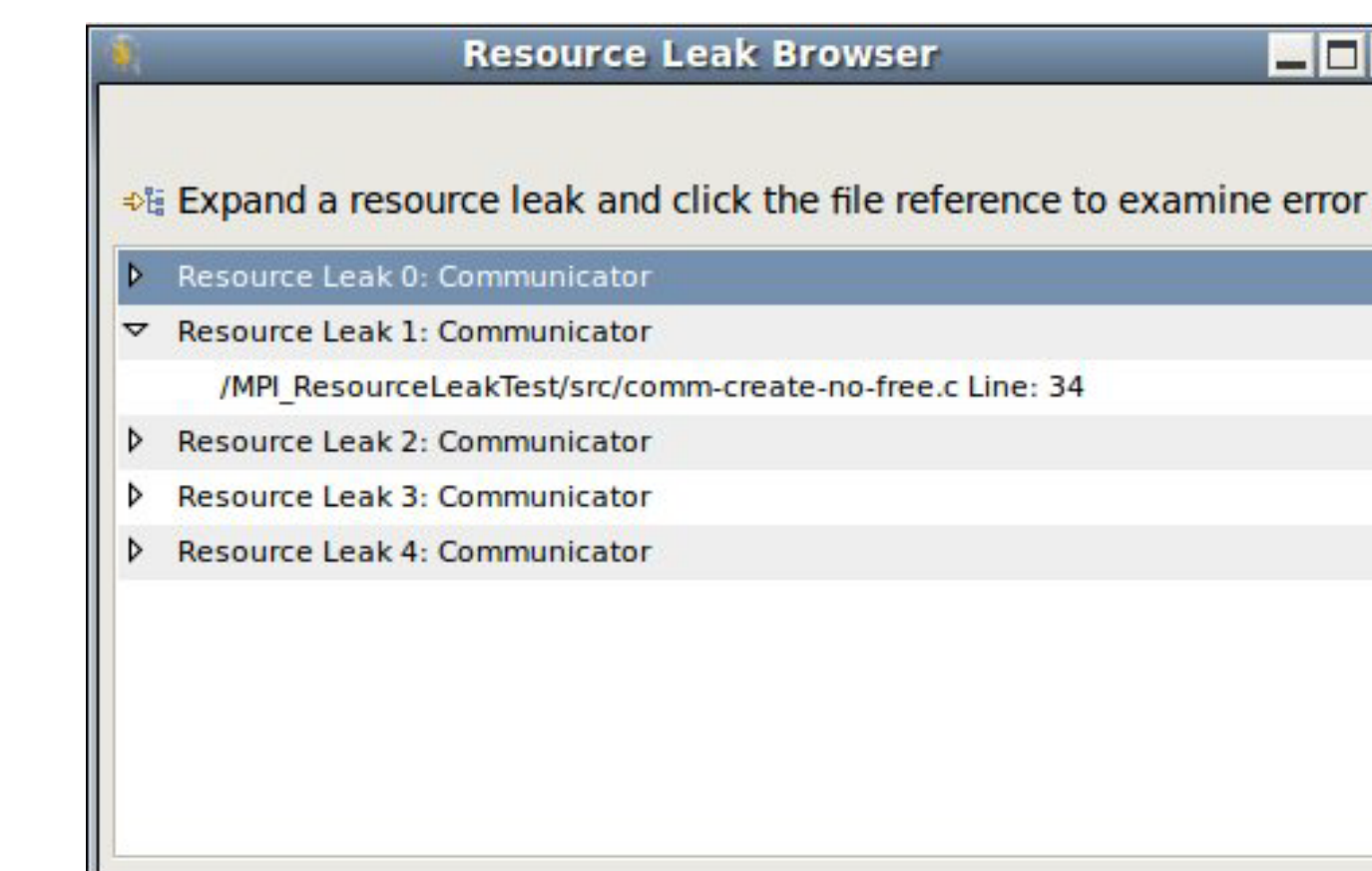
One click in the Analyzer View launches this Java based visualization tool

- Provides a more global view of what ISP found at runtime
- Shows MPI calls as a graph to aid in debugging
- Displays intra/inter process communication
- Lets the user interact with the graph for finer-grained information
- Connects matching deterministic and non-deterministic MPI calls
- Allows the user to look at all processes or focus on a single process
- Displays the Happens-Before relationship between MPI calls

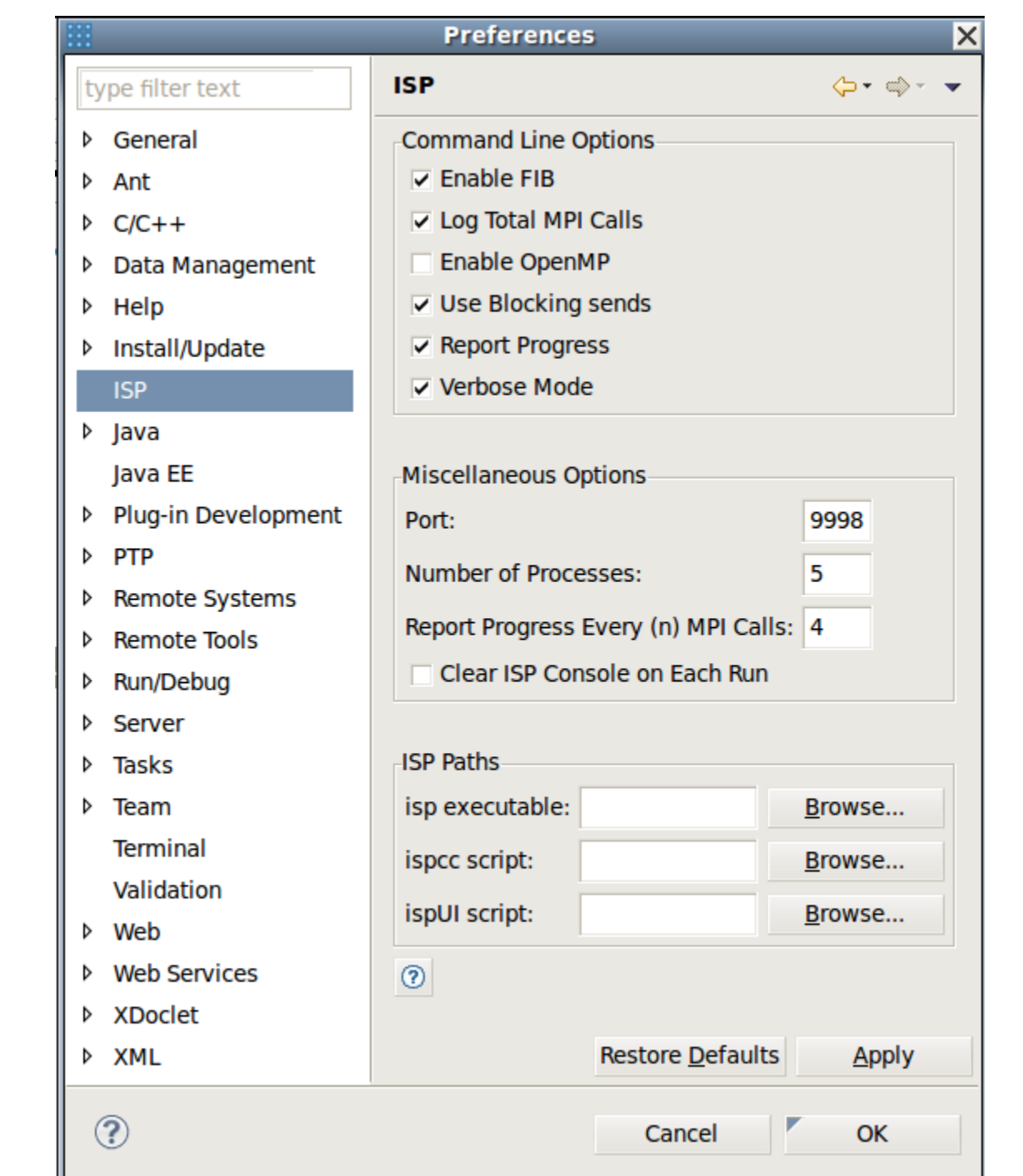
## GEM Overview

GEM provides push button dynamic formal verification for MPI C programs using ISP (In-situ Partial Order), which was developed at the University of Utah. ISP can definitively find deadlocks, resource leaks and local assertion violations in MPI C programs.

- Completely integrated Eclipse plug-in
- Comprehensive help contribution
- Supports Makefile and Managed-Build MPI C projects
- Analyzer View lets the user examine runtime behavior of their code
- Dedicated ISP Console View captures ISP console output



Allows the programmer to visit the line of code that generated an error. A single click opens the Eclipse editor with the offending line highlighted and in view



Easily set ISP command line options with GEM's Eclipse Preference Page

## Acknowledgements

We gratefully acknowledge the work done by the creators of ISP, notably Sarvani Vakkalanka, Anh Vo, and Michael DeLisi. We also acknowledge the contributions of Sriram Ananthakrishnan who created the matches-before viewer of ISP. Thanks also to Beth Tibbitts and Greg Watson of IBM for their mentoring and encouragement to contribute GEM to Eclipse <http://www.eclipse.org/ptp/>

Supported in part by Microsoft, NSF CNS-0509379, CCF-0811429, CCF-0903408



[isp-dev@cs.utah.edu](mailto:isp-dev@cs.utah.edu)