



IBM Research

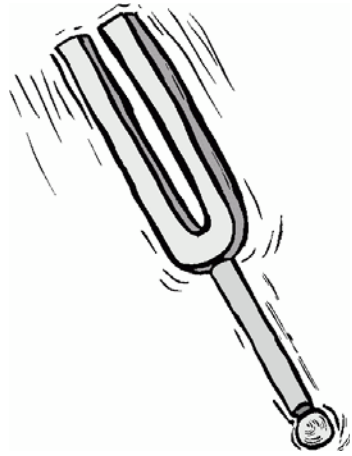
TuningFork: Performance Trace Visualization PTP Performance Framework

Beth Tibbitts
tibbitts@us.ibm.com

May 2007

© 2007 IBM
Corporation

TuningFork

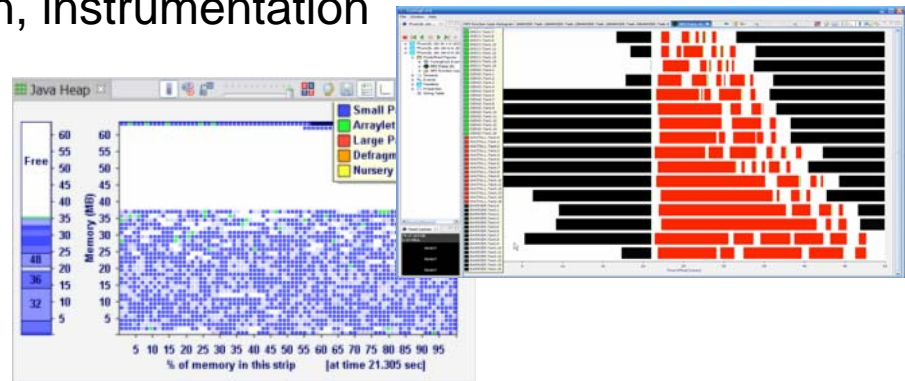
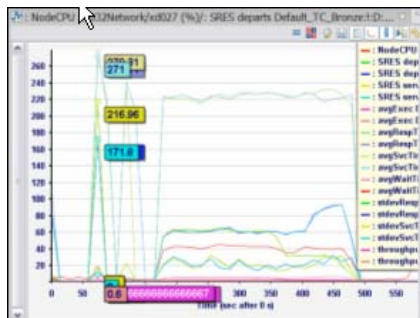


Rich graphical Eclipse RCP (Rich Client Platform) application

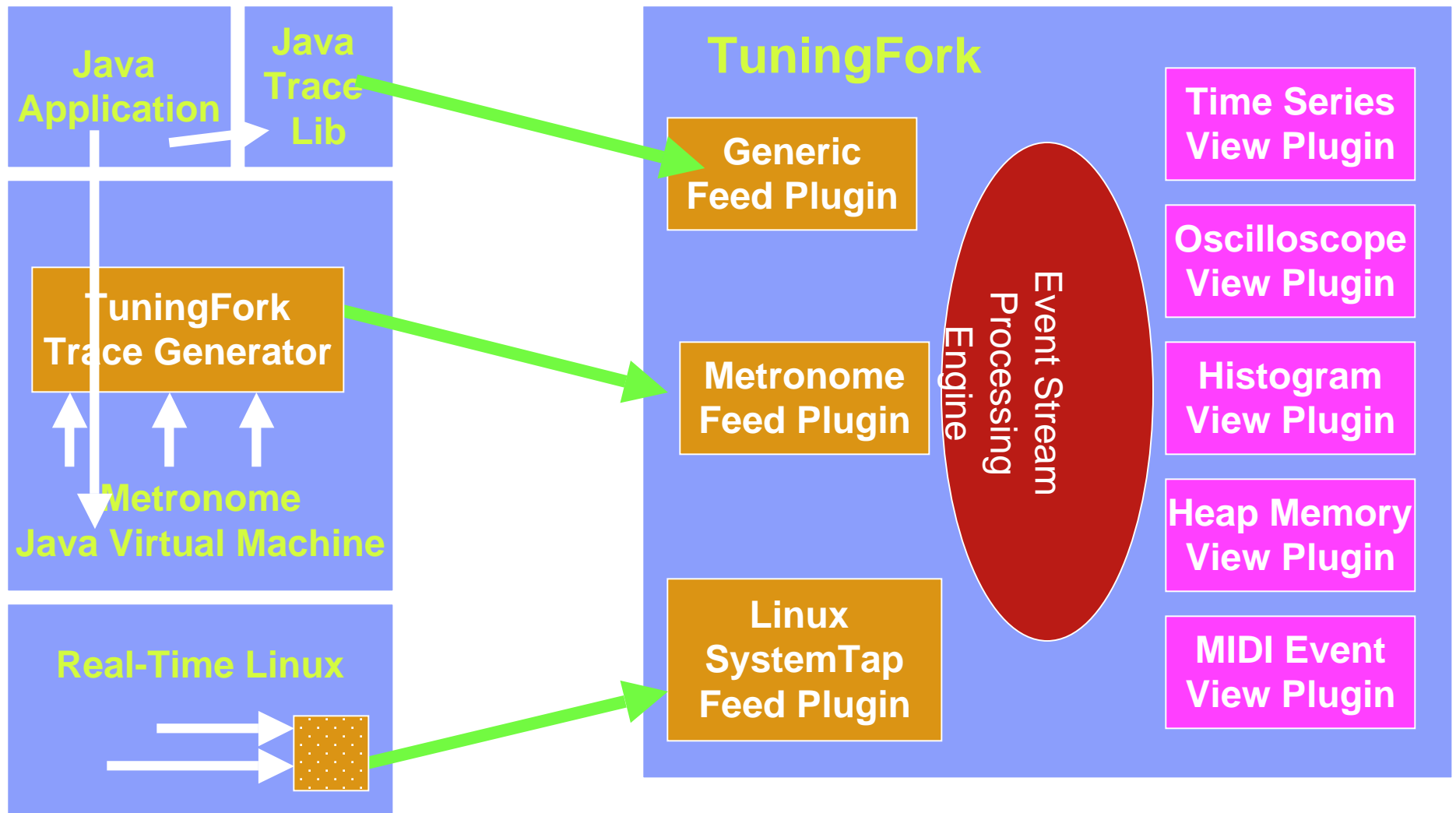
- *Uses OpenGL for rich graphics, but can use SWT at reduced performance*
- *Visualizations of trace information from Java, but being adapted for MPI traces and other HPCS/PERCS uses*
- *Available on alphaWorks now*

<http://www.alphaworks.ibm.com/tech/tuningfork>

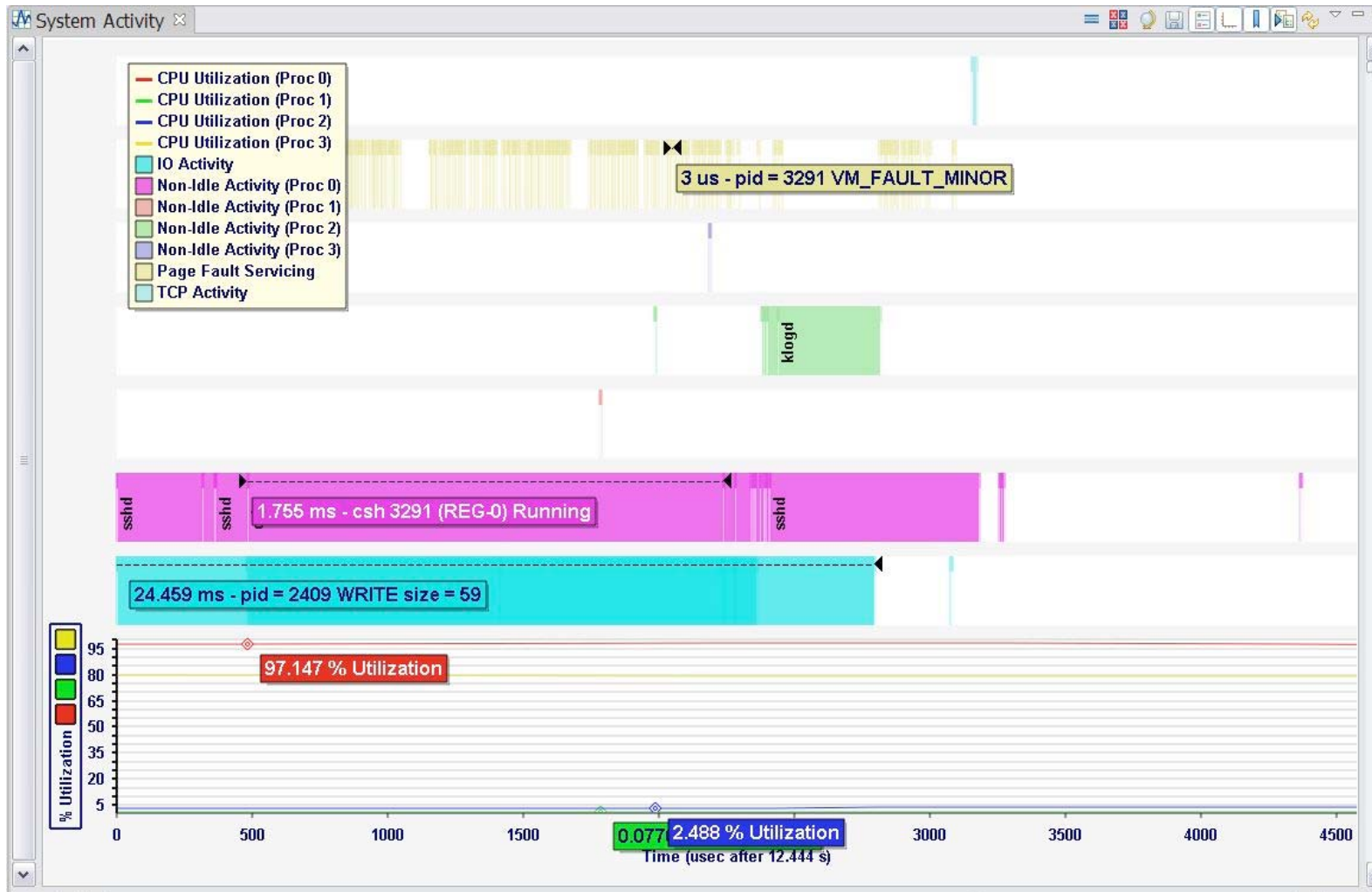
- *Plans: Integrate with IDE, source code integration, instrumentation*



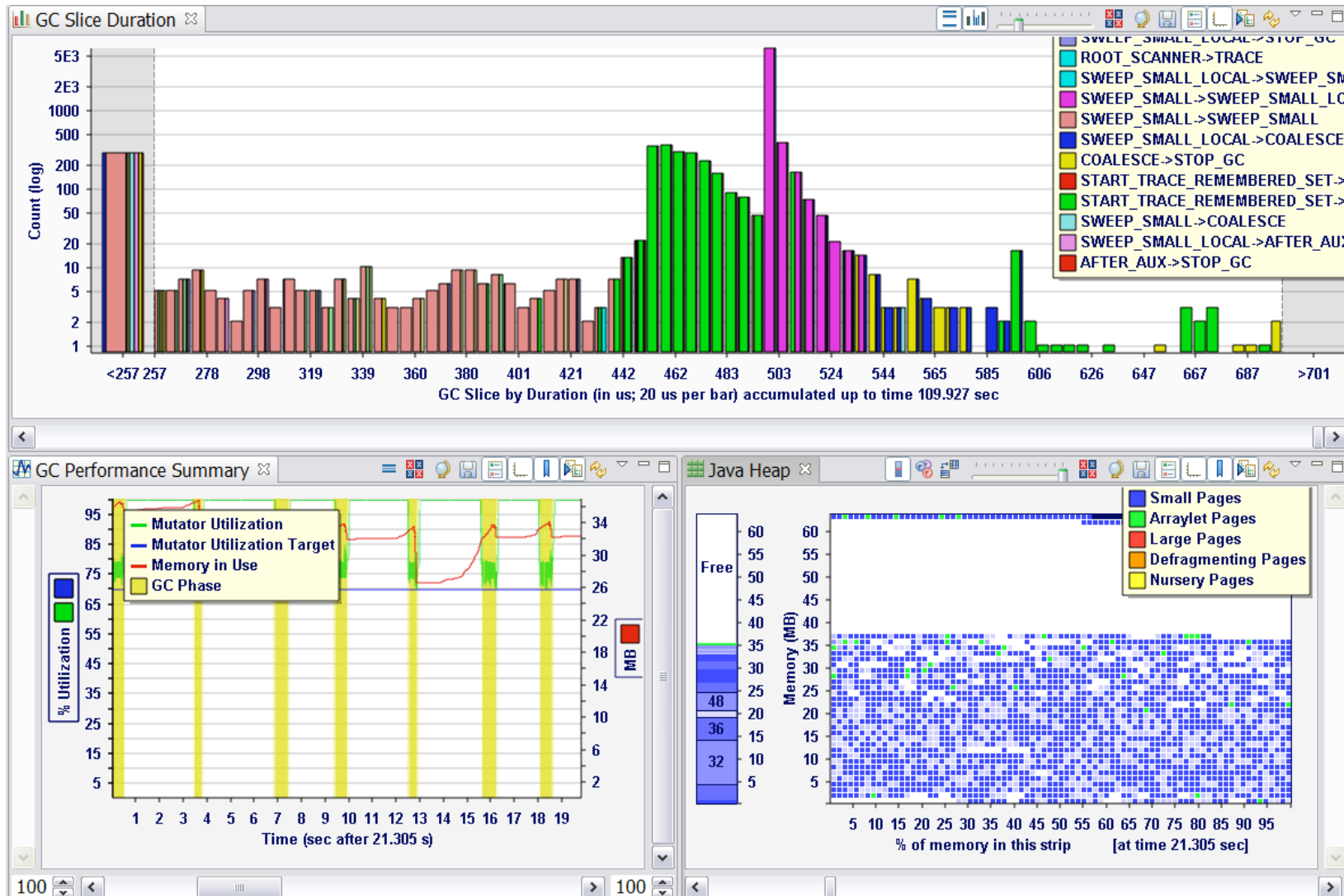
TuningFork Architecture



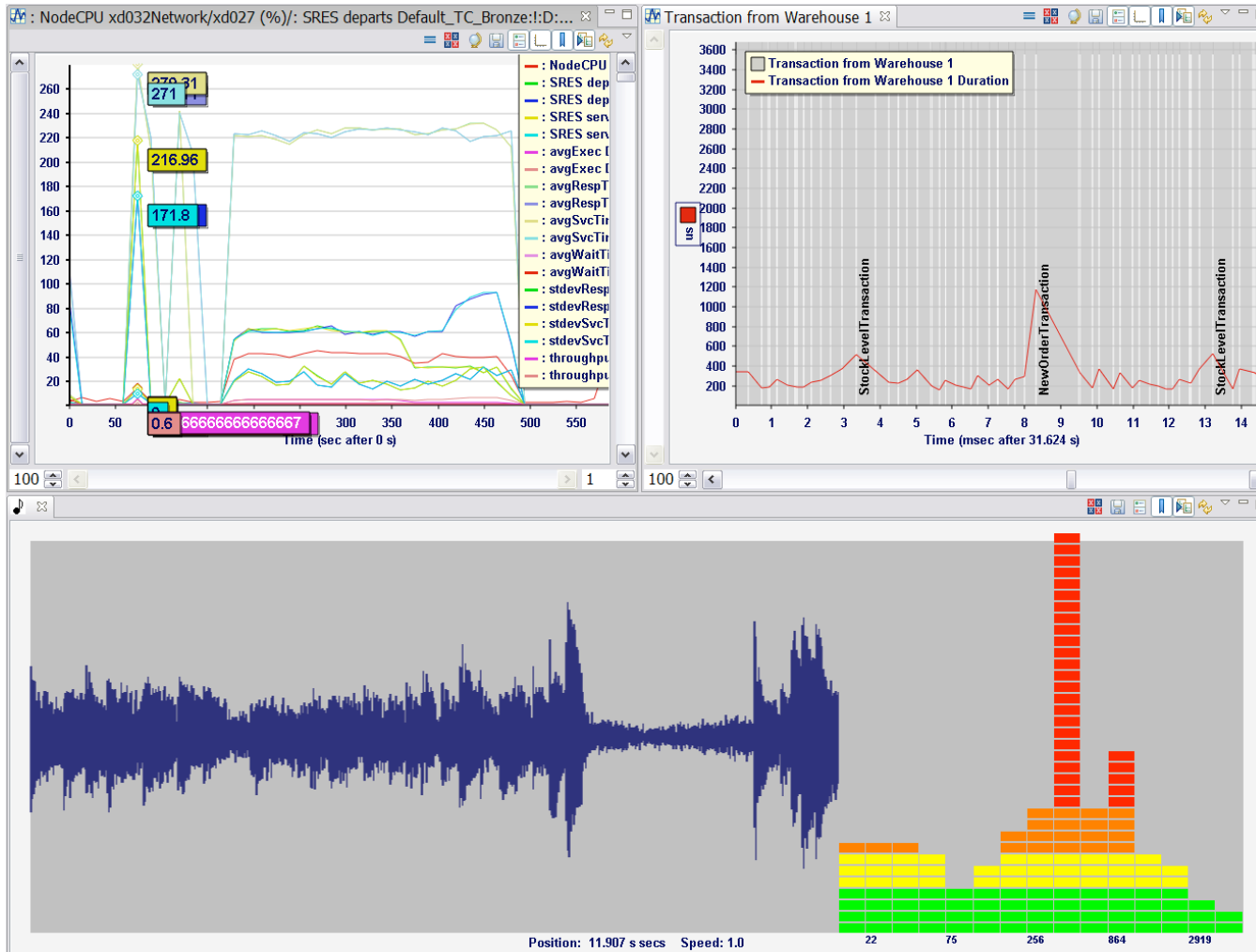
OS-level Activity (Linux)



JVM-level Activity (Real-Time Java)

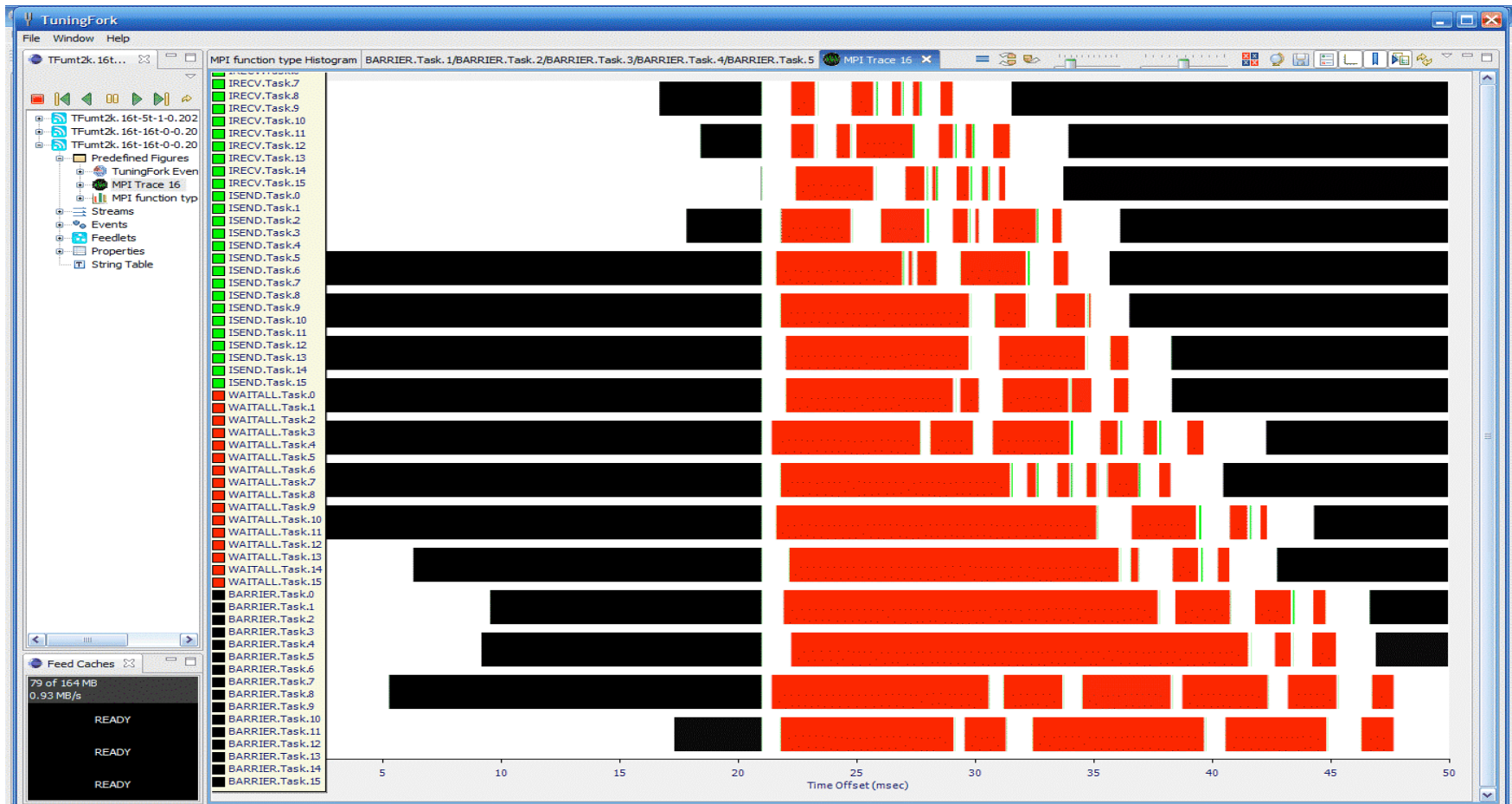


Application-level Activity



TuningFork adaptations for MPI traces

Appended below is a 16 task bluegene trace zoomed in between two barriers. The MPI functions are color coded. The red events are waitalls. The black events are barriers. The green events are irecv, and isend, which take little or no time. This work has not been completed... the attached is to represent work in progress.



PTP Performance Framework : Problem

Problem:

- Performance tools need to be conveniently integrated with Eclipse and PTP
- Each tool provider shouldn't have to duplicate the infrastructure for providing tool launching, trace file handling, and visualization tool integration.
- Tools should have some consistency for PTP integration

Background / Current State:

- TAU initial integration with Eclipse and PTP has been submitted to PTP project; great deal of work to hook to up “eclipse plumbing.” (est: 50% is or could be generic plumbing)
- If the generic parts of the “plumbing” could be generalized into a framework, it would reduce duplication of effort and simplify effort for tool implementers/integrators.
- Requirements for other perf. tools (IBM HPC toolkit, TuningFork, etc.) to be evaluated.
- Feedback from other potential PTP contributors needed

PTP Performance Framework: Proposal

Implementation overview:

- PTP Perf. framework would provide Eclipse *extension points* as the integration point for tool implementers. Potential extension points (generalized) include:
 1. UI for tool setup and interaction with user, storage of settings
 2. Instrumentation tool launch,
 3. Trace file/DB management
 4. Remote execution support (e.g. tasks before/after remote exec launch)
 5. Visualization: graphs, integration with source/machines view, etc.
 6. Analysis – leveraging performance data, link to PLDT tools, etc.
 7. Standalone (e.g. RCP) support for perf. tools
- Tool implementers would provide an Eclipse "extension" implementing any of the above.
- Tools could include both Eclipse plug-ins (e.g. TuningFork) and external tools (e.g. TAU, IBM HPC Toolkit, Paraprof, etc.)