A09

Visual Performance Analyzer

Luc Smolders

IBM SYSTEM p, AIX 5L
and LINUX TECHNICAL
UNIVERSITY
Sept 11 - 15, 2006

Las Vegas, NV

charts to be available at:
http://www.ibm.com/developerworks/blogs/page/Systemptechuniv

Agenda

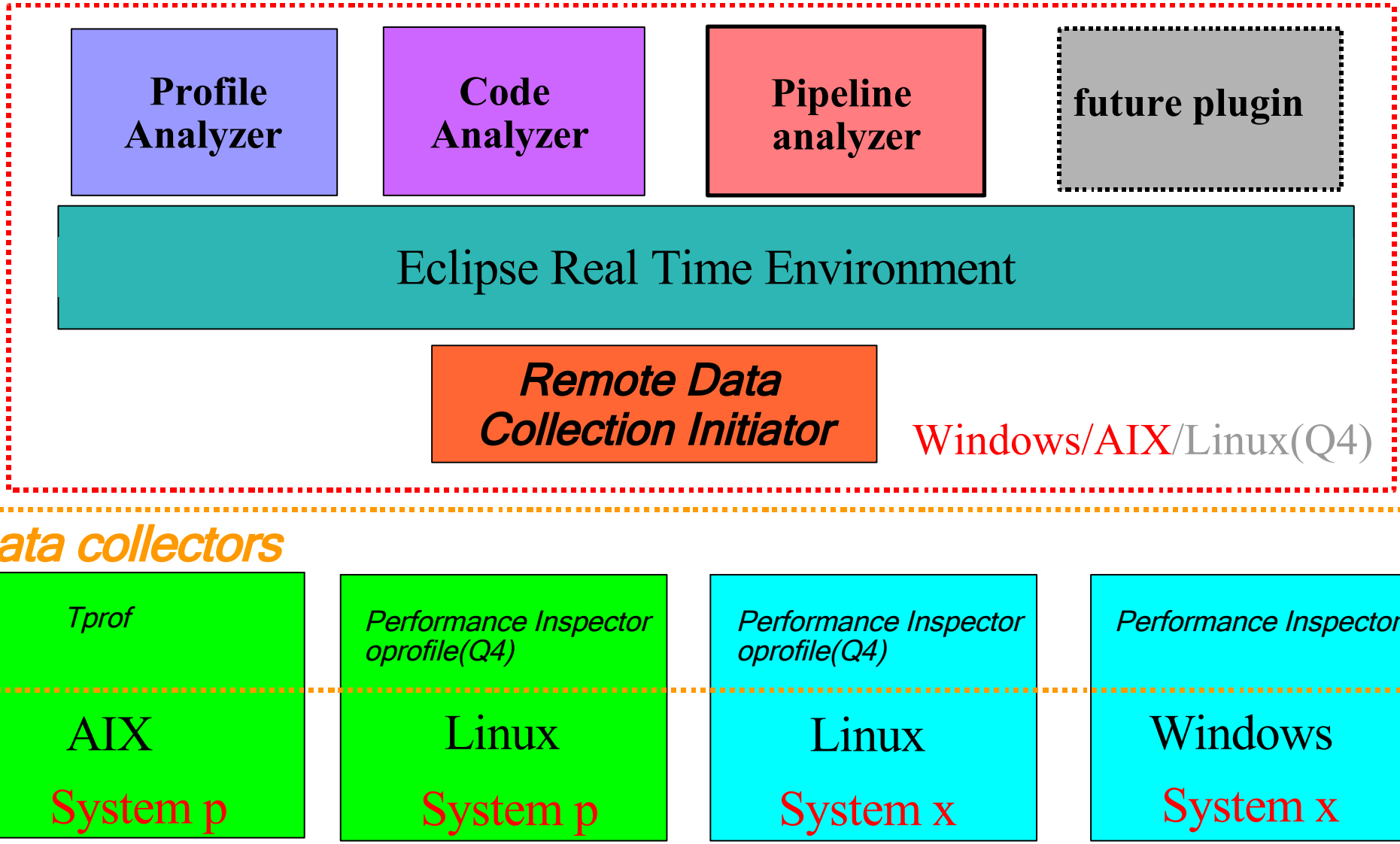
- Visual Performance analyzer
  - goals and components
- Profile Analyzer overview
  - components, selected features and sample views
- Code Analyzer overview
  - motivations, features and sample views
- Pipeline Analyzer overview
  - features and sample views
- Code availability
  - Alphaworks and Source Forge
- Future plans
- Recorded demo

Visual Performance Analyzer - goals

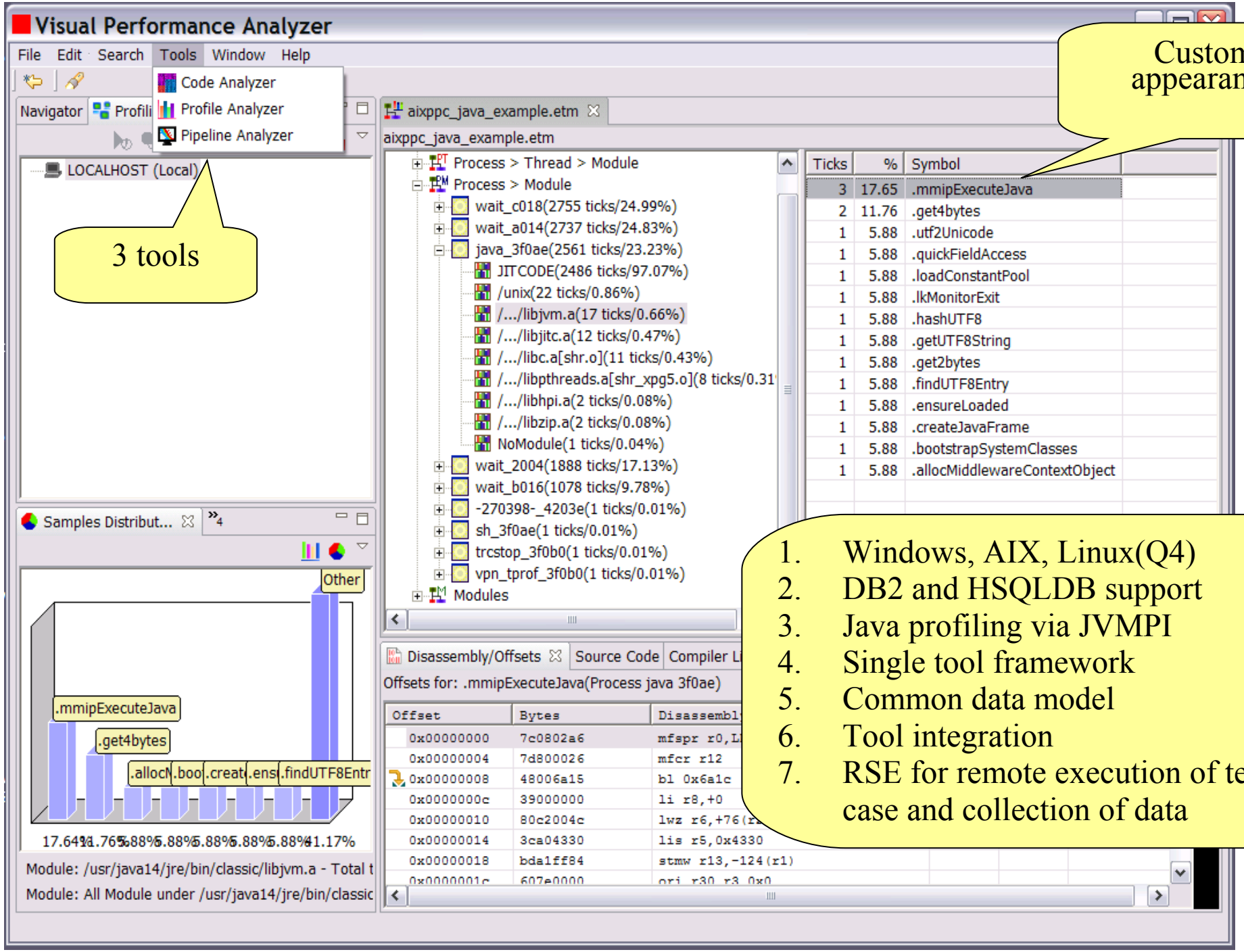
- Provide platform independent easy to use integrated set of graphical application performance analysis tools
- Target audience includes
  - performance analysts
  - application developers
  - compiler developers
- Leverage existing platform specific non-GUI performance analysis tools to provide comprehensive set of features
  - Performance Inspector on Windows and Linux
  - tprof on AIX
  - oProfile on Linux
  - FDPR on Linux and AIX
- Create consistent set of integrated tools to provide platform independent drill down performance analysis experience

Visual Performance Analyzer - components

- The Visual Performance Analyzer is an Eclipse based tool set, initially including three plug-in applications working collaboratively: Profile Analyzer, Code Analyzer and Pipeline Analyzer

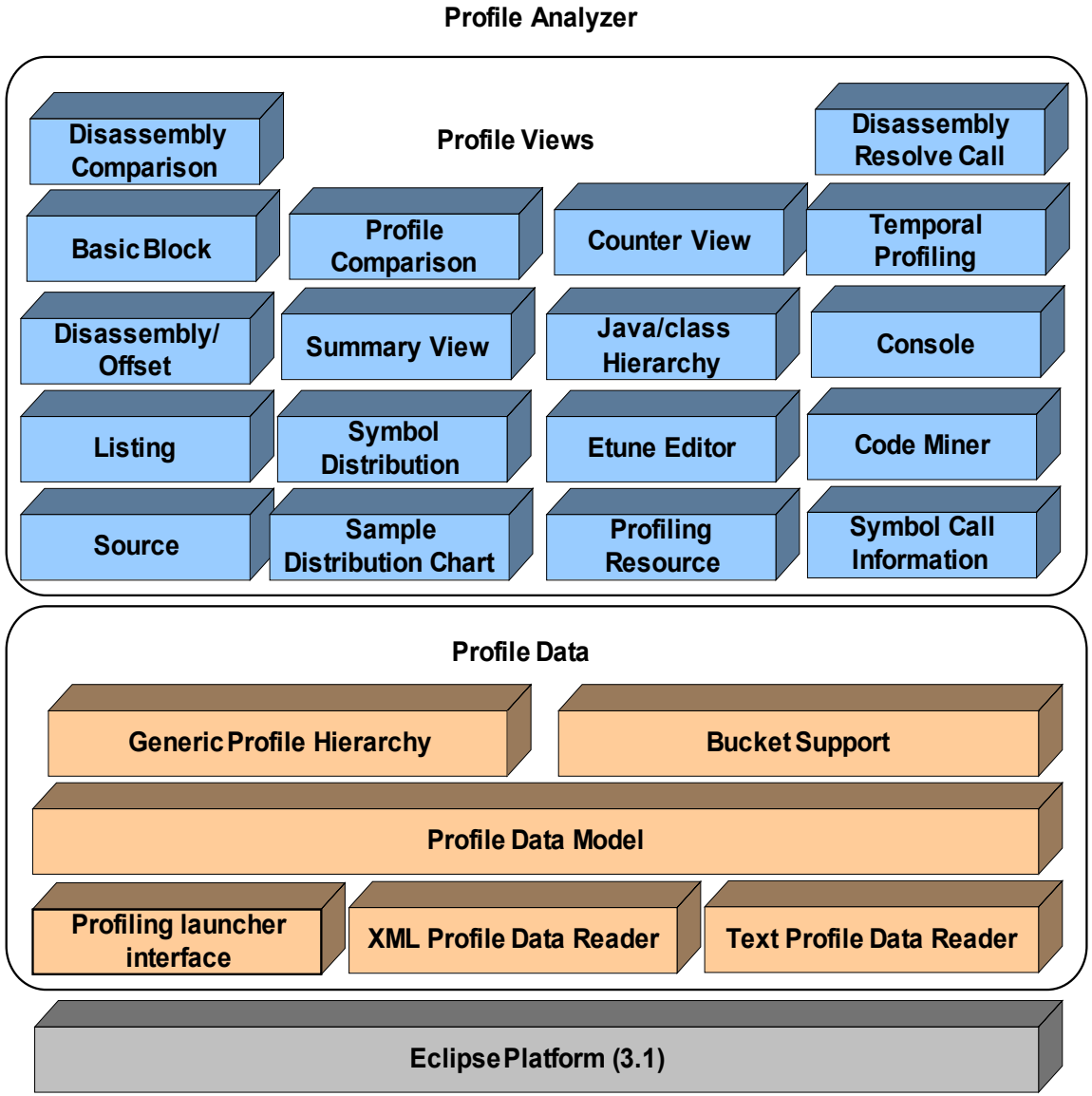


Visual Performance Analyzer - current look



Profile Analyzer - components

- Uses data from numerous platform specific profiling tools, including:
  - AIX tprof XML output data
  - Linux and Windows profiles produced by Performance Inspector's tprof
  - IBM JRE Java profile data
  - Linux profiles produced by oprofile via XML output option
    - coming in the next version(Q4)
  - profiles from additional IBM systems platforms
    - internal version only for now
- Provides many analysis views and features to help identify CPU performance bottlenecks



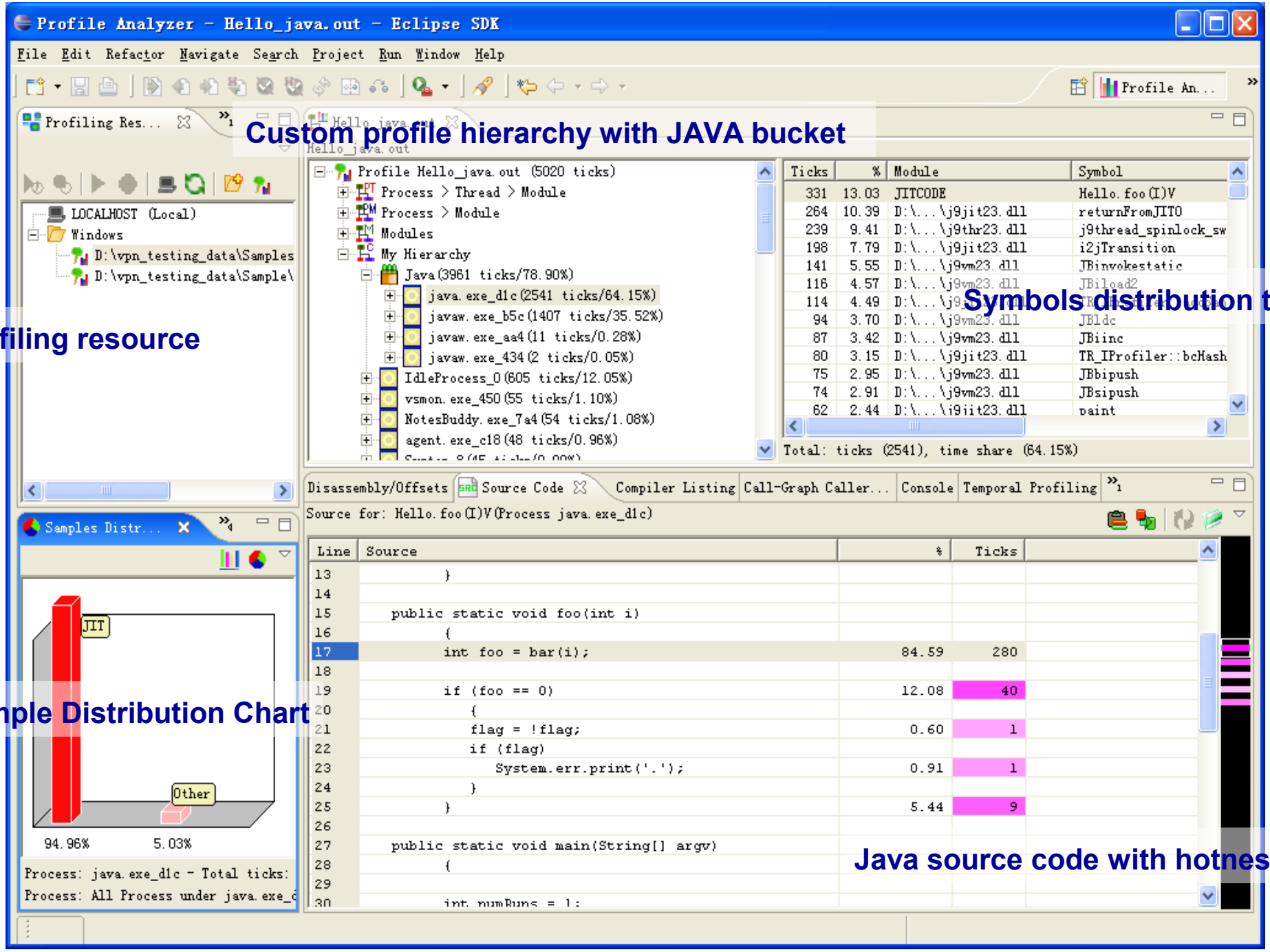
Profile Analyzer - selected features

- Customizable profile hierarchies
  - by default uses generic hierarchies
  - includes JAVA class hierarchy view
  - allows user-defined hierarchical views
    - includes bucketized view
- Built-in multi-platform disassembler
  - supports PowerPC, IA32 and AMD64
  - provides annotated instruction view
- Source annotation
  - synchronized with disassembly view
- Basic blocks view
  - automatically splits instructions into basic blocks and annotate them with profiling data
- Listing files annotation
  - limited to .lst files for now
- Temporal profiles
  - shows evolution of samples distribution over time
  - works on process, threads, objects/modules and on symbols

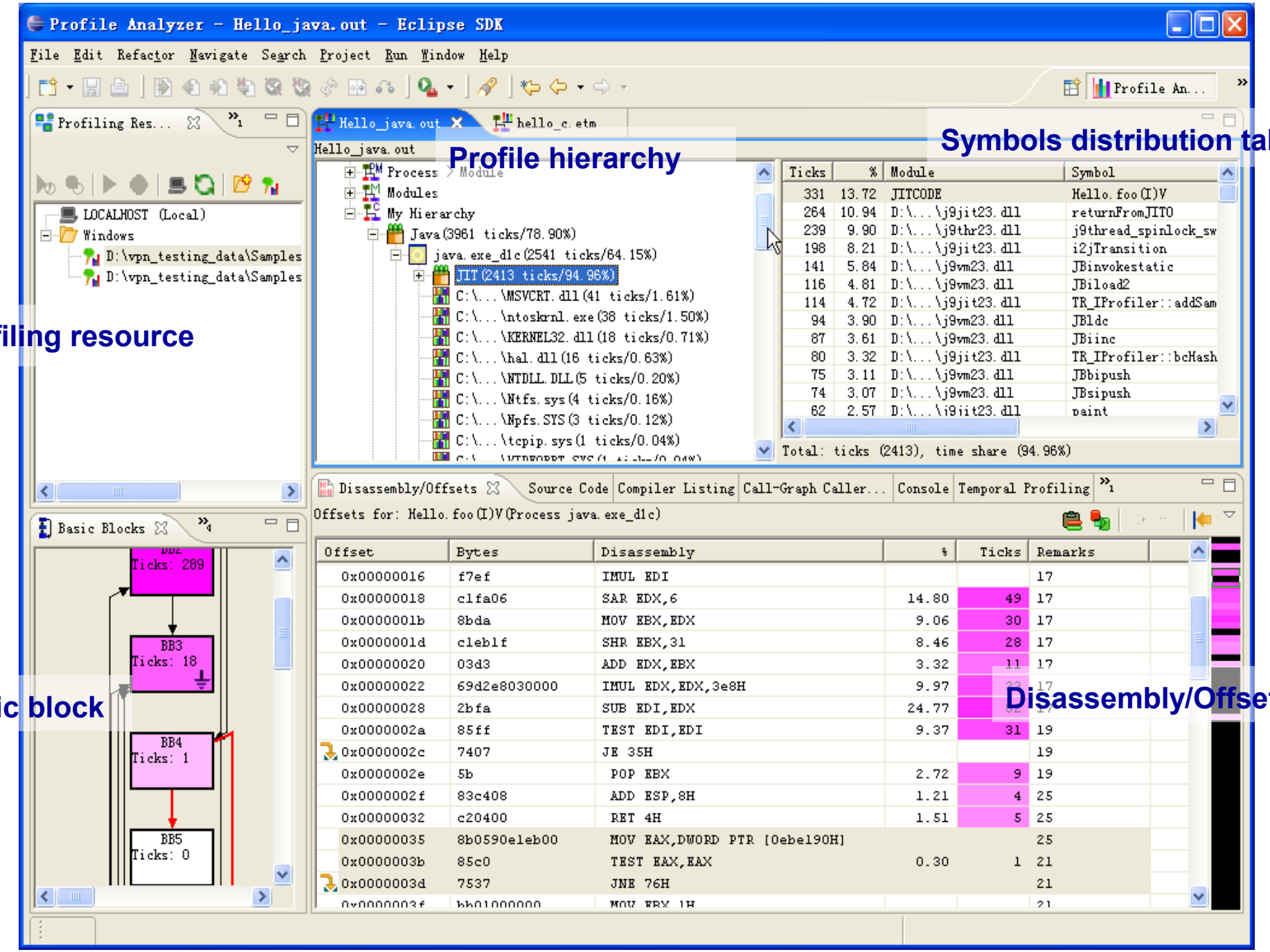
Profile Analyzer - selected features*(cont)*

- Profile comparison and merging
  - includes normalization
    - on transaction rate
    - cpu busyness
    - number of cpus
- Remote capture of profiles
  - uses RSE plugin to
    - start remote test execution
    - capture profile data
- Hardware events support
- Code Miner
  - wizard populates SQL database with profile information
    - supports DB2 and HSQLDB
  - queries allow detection of performance patterns hard to find with traditional profilers, e.g.
    - find hottest pairs of sequential instructions
    - find all symbols that contain a particular instruction sequence
    - all symbols that are hotter than a certain threshold and that have a certain pattern in their name
  - ideal for analysis of flat profile
    - where no single symbol uses more than a fractional percentage of total ticks

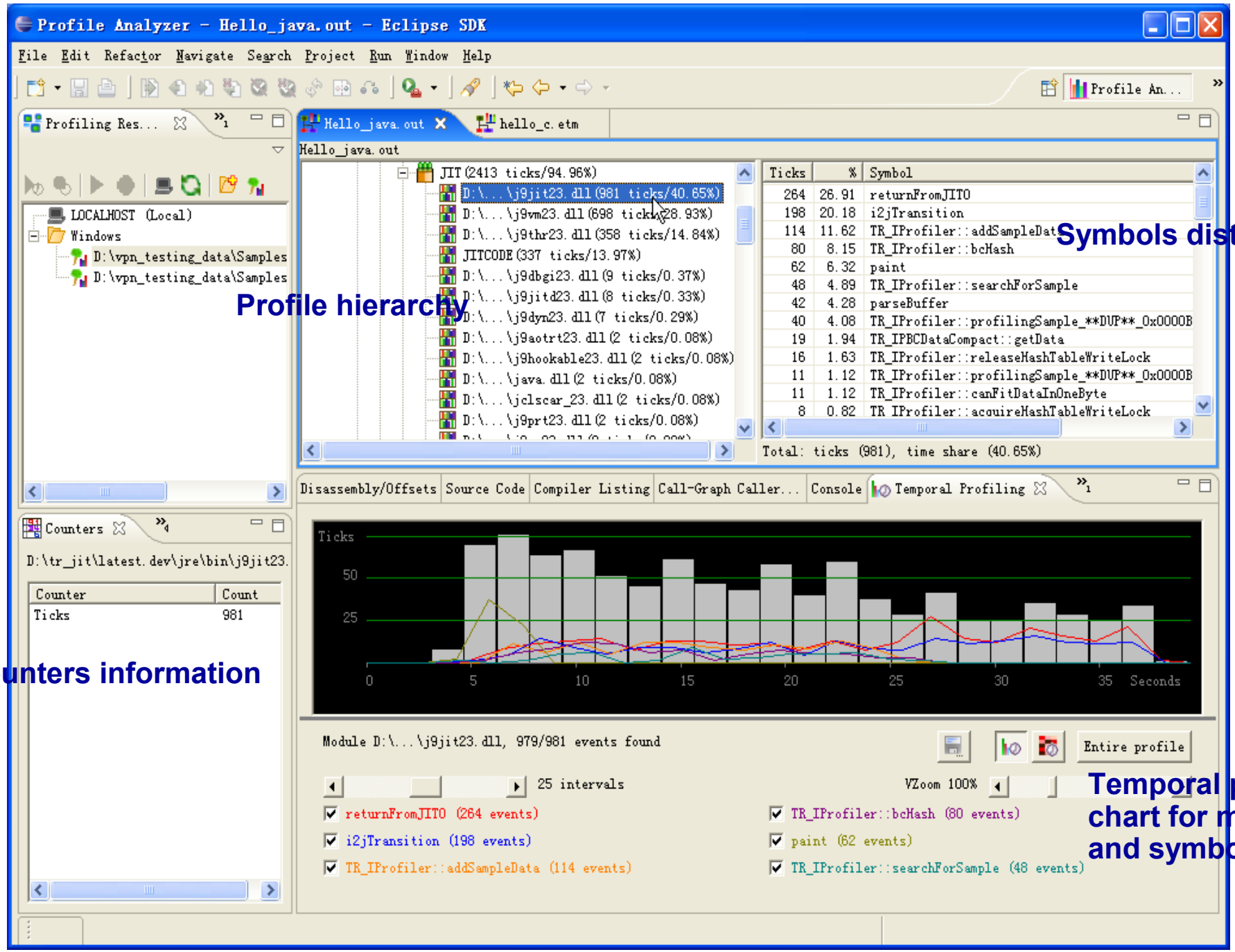
Profile Analyzer - custom hierarchy and Java support



Profile Analyzer - symbol distribution and disassembly



Profile Analyzer - temporal profiling



Profile hierarchy

Symbols distribution

Counters information

Temporal profiling chart for modules and symbols

Code analyzer – motivations

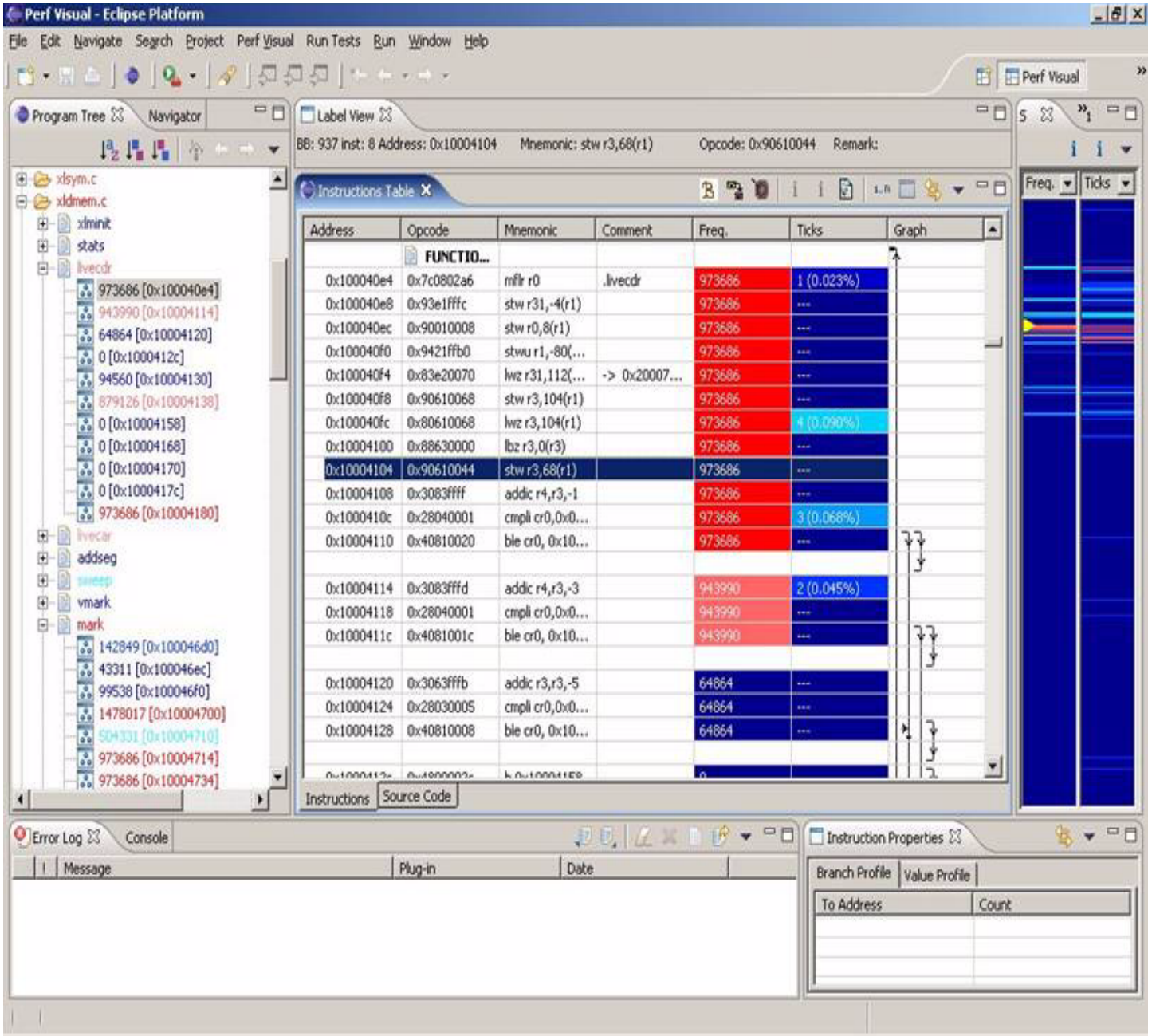
- Architectures are becoming more and more complex
- Using only hardware simulators to detect information about potential instruction level performance bottlenecks in a given program is very hard
- There is a need for performance tools that can statically analyze and visualize programs structure and characteristics for a specific platform design
- Target audience includes
  - hardware architects
  - compiler writers
  - application developers

Code Analyzer - overview

- Code Analyzer is an eclipse plugin which performs a comprehensive static analysis on executable files and libraries/DLLs
  - relies on the FDPR-Pro tool for the analysis phase
- Displays the analyzed information together with optional profiling data collected by
  - tprof (AIX only)
  - FDPR-Pro (AIX and Linux PPC) phase 2 output
    - fdpr -12 -p binary -x binary binary_options
- The code is then colored according to:
  - frequency counters
    - gathered by FDPR-Pro
      - can also collect value profiles
  - hardware event ticks
    - gathered by tprof

Code Analyzer – overview*(cont)*

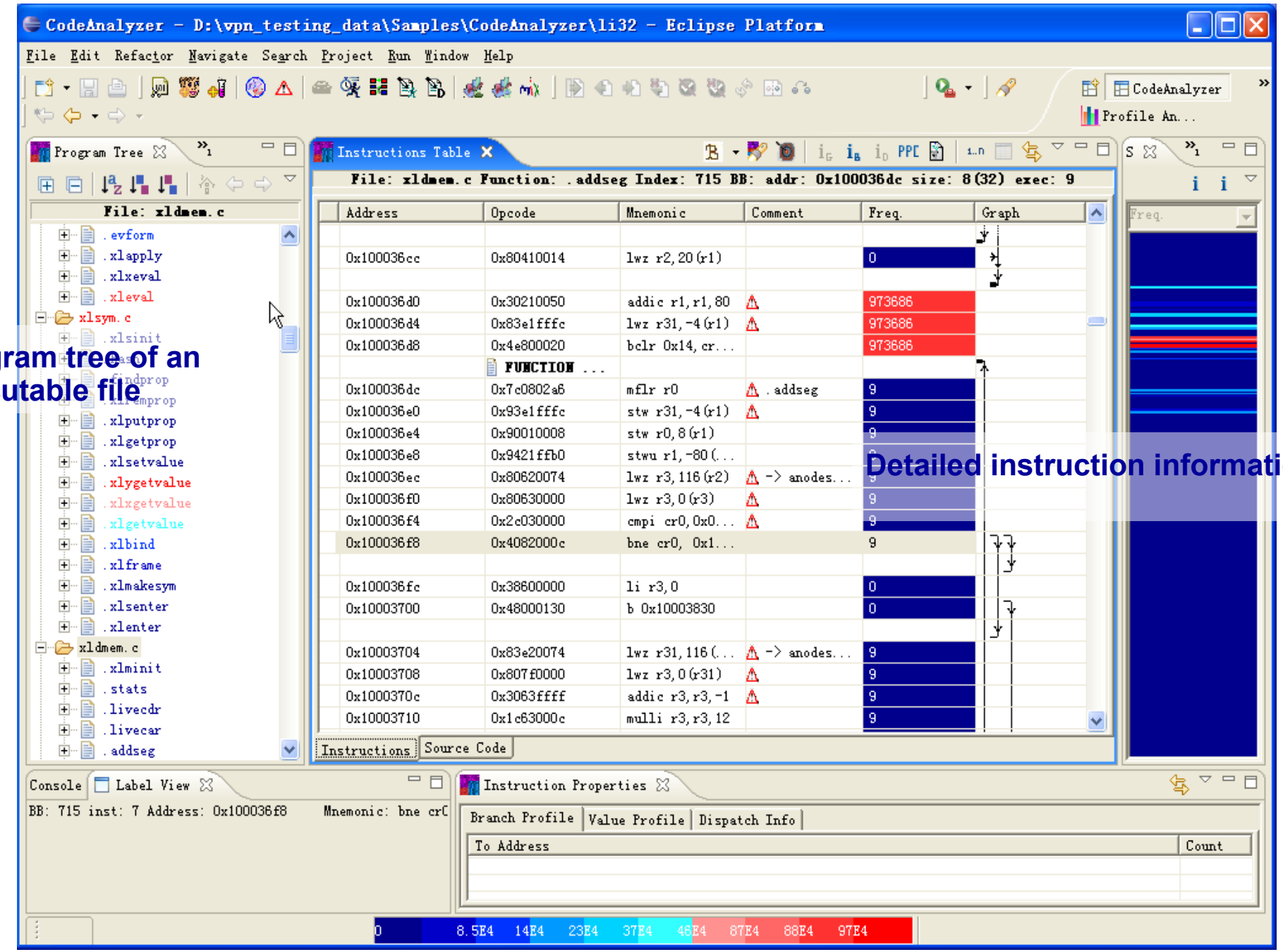
- Provides several views of the input binary
  - assembly instructions
  - basic blocks
  - procedures
  - CSECT modules
  - control flow graph
  - hot loops
  - call graph
  - annotated source code
  - dispatch group formation
  - pipeline slots and functional units



Code Analyzer - features

- Analysis features provided
  - easy code navigation
    - basic block, function, hot loops, ...
  - architectural comments generation
    - based on static analysis of code structure
  - performance comments generation
    - when profile information is available
  - dispatch group and pipeline functional units mapping
  - source code mapping of executable
  - statistical views
- Binary types currently supported
  - AIX XCOFF
  - Linux ELF
  - JITed code
    - embedded in AIX tprof XML output file and
    - J2N files produced by Performance Inspector's jprof

Code Analyzer - views



Program tree of an executable file

Detailed instruction information

Code Analyzer - architecture comments

- Three types of comments
  - Power5
  - Power6
  - general

- Examples of comments
  - pipeline stalls (bubbles) for the Power6 architecture
  - unresolved dependencies
  - unaligned code
  - unreachable code and non-used data
  - architectural hazards, for instance
    - load after store

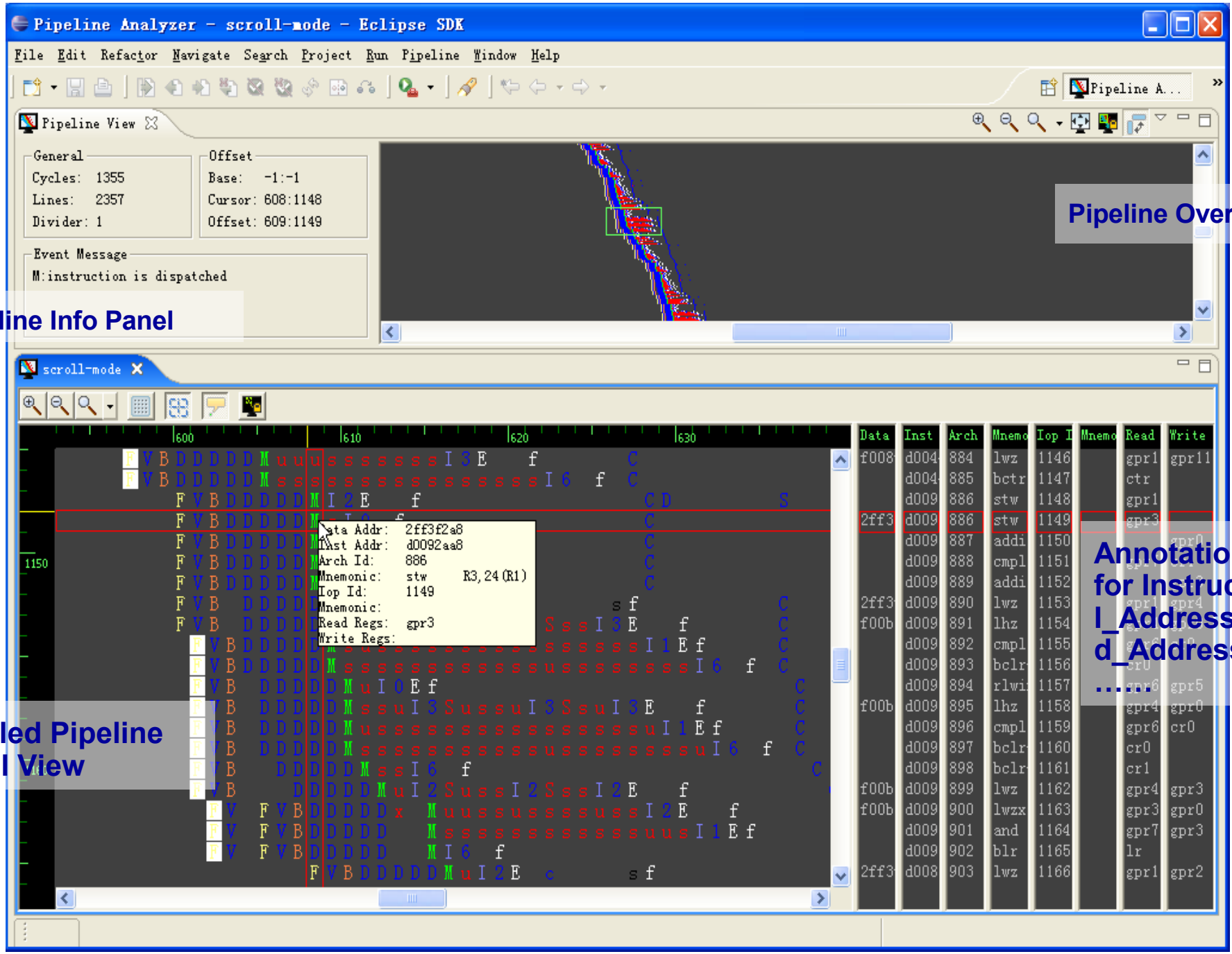
Code Analyzer - performance comments

- Profile-based comments
  - non-variant instructions within Hot loops
  - hot function calls preceeded by overwriting non-volatile registers
  - hot saves and restores of registers which could be relocated to cold spill areas
  - hot instructions that could be scheduled to colder areas in the code
  - removable hot branches
  - dead code and non-used identification
  - hot direct unconditional branches
  - hot direct conditional branches that are taken, which have a colder fallthru
  - hot call sites that are appropriate candidates for function inlining
  - hot call sites that are appropriate for function specialization
  - hot loops that are appropriate for loop unrolling
  - performance bottlenecks identified with collected hardware events
  - hot TOC load instructions that can be replaced by immediate add instructions

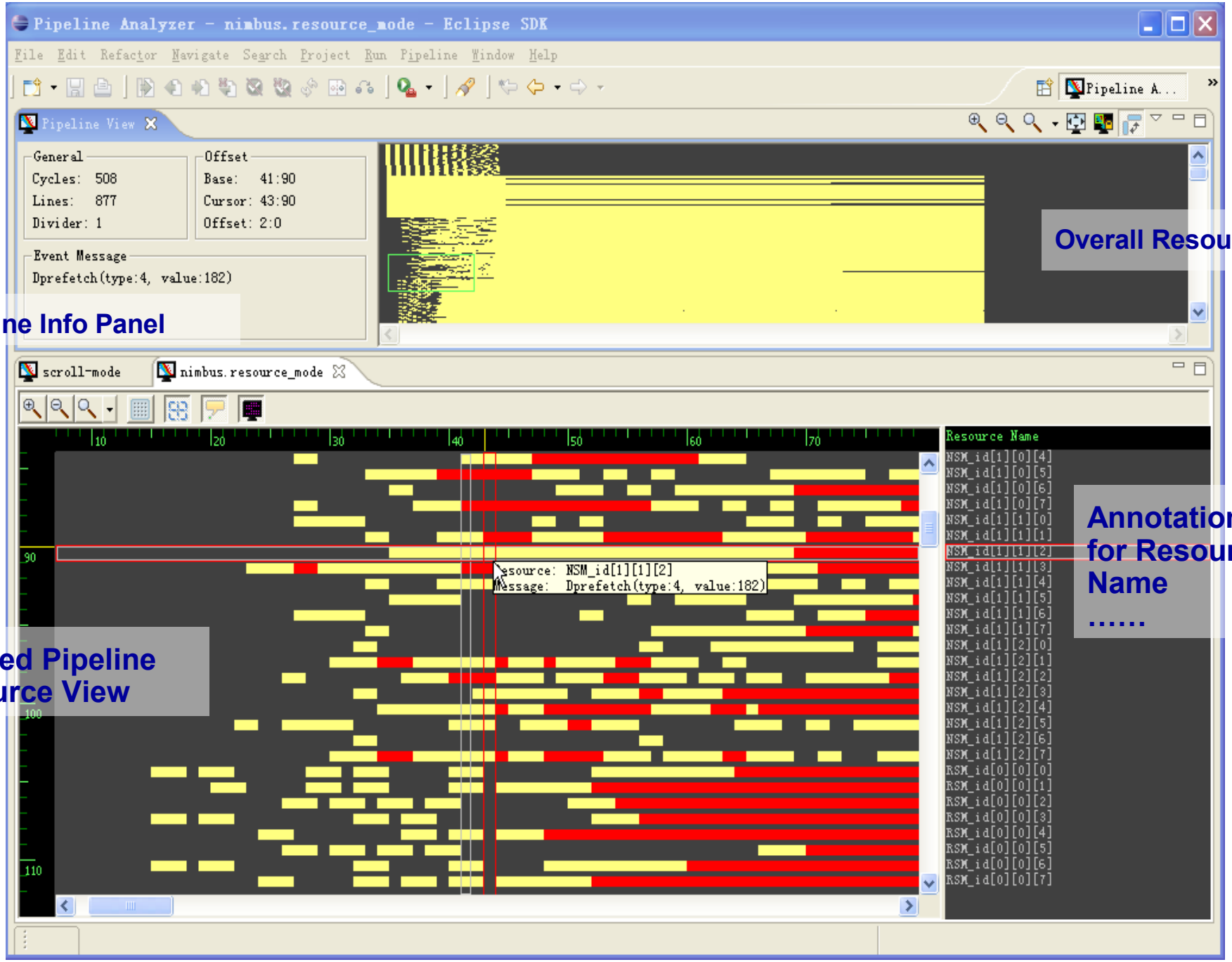
Pipeline Analyzer - features

- Pipeline visualization tool for POWER series processor
  - pipeline data comes from processor timing models which is fed an instruction trace of a program execution
- Provides views in two modes:
  - The "scroll" mode shows how instructions are executed in the pipeline
    - various pipeline events are represented by different symbols in different colors
  - The "resource" mode shows how resources are utilized during instruction execution
    - in this view, a continuous bar in a color means during this cycle period a particular resource is being utilized.
- Provides graphical views of pipeline data, easy data navigation and customization for GUI elements.
- Very detailed hardware performance data is provided
  - useful for hardware performance design or critical library or module analysis.

Pipeline Analyzer – view in scroll mode



Pipeline Analyzer – view in resource mode



Code Availability

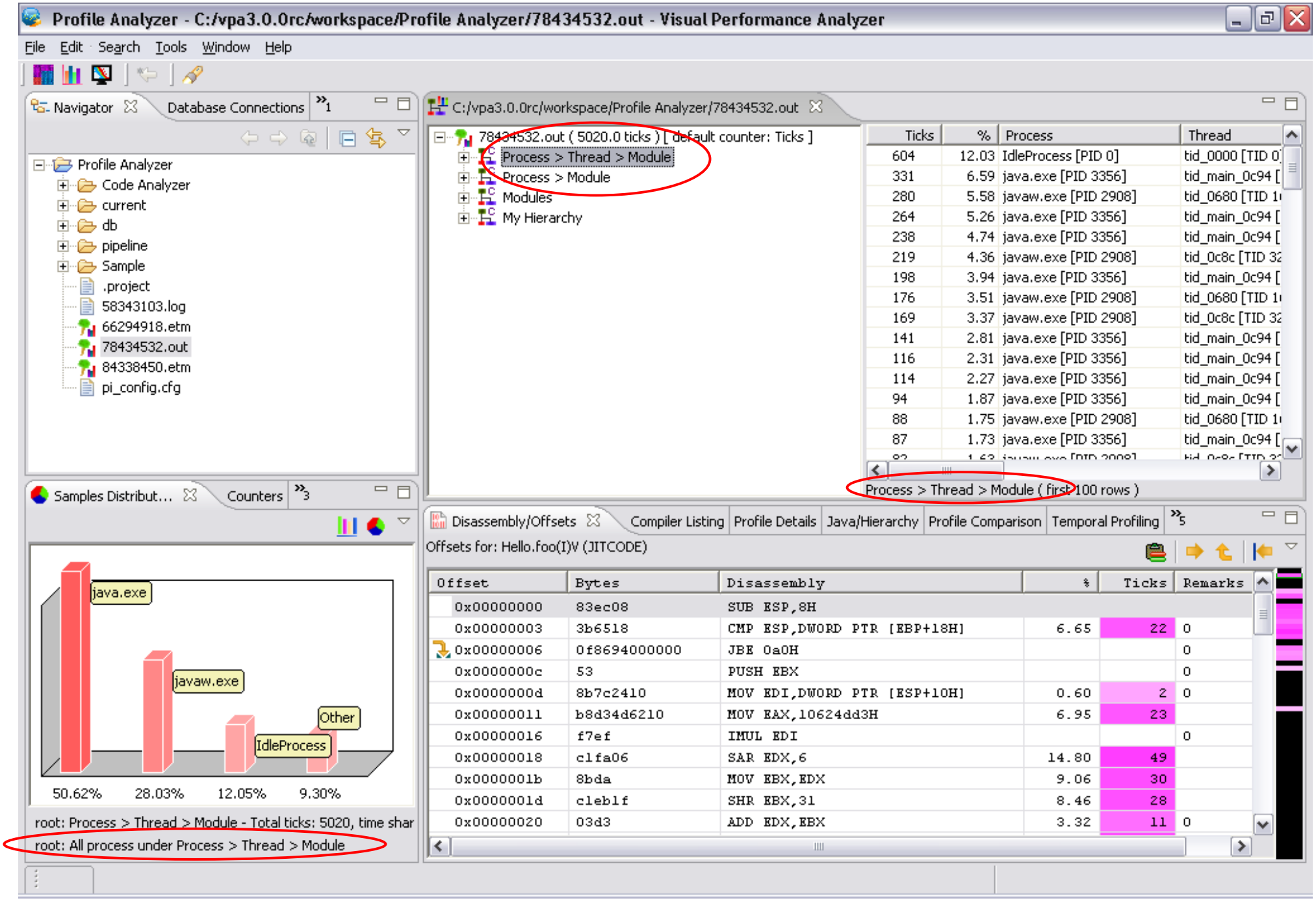
- AIX Profile Analyzer support
  - tprof -X option(undocumented) produces XML file loadable into VPA
    - included with AIX 5.3 TL5
- Alphaworks (www.alphaworks.ibm.com)
  - Virtual Performance Analyzer
    - Windows and AIX binaries
  - Performance Inspector for Windows
    - profiler for Windows
  - Post-Link Optimization for Linux on Power
    - FDPR for Linux PowerPC
  - Full-System Simulator for IBM PowerPC 970
    - can generate instruction traces to be used as input for processor timing models
  - IBM Performance Simulator for Linux on POWER
    - timing models for most recent PowerPC processors
      - output feeds into Pipeline Analyzer
    - also includes older standalone version of Pipeline Analyzer
- Sourceforge
  - Linux Performance Inspector
    - support Intel, AMD and PPC64
    - includes itrace tool which feeds into processor timing models
  - http://perfinsp.sourceforge.net/

Future plans

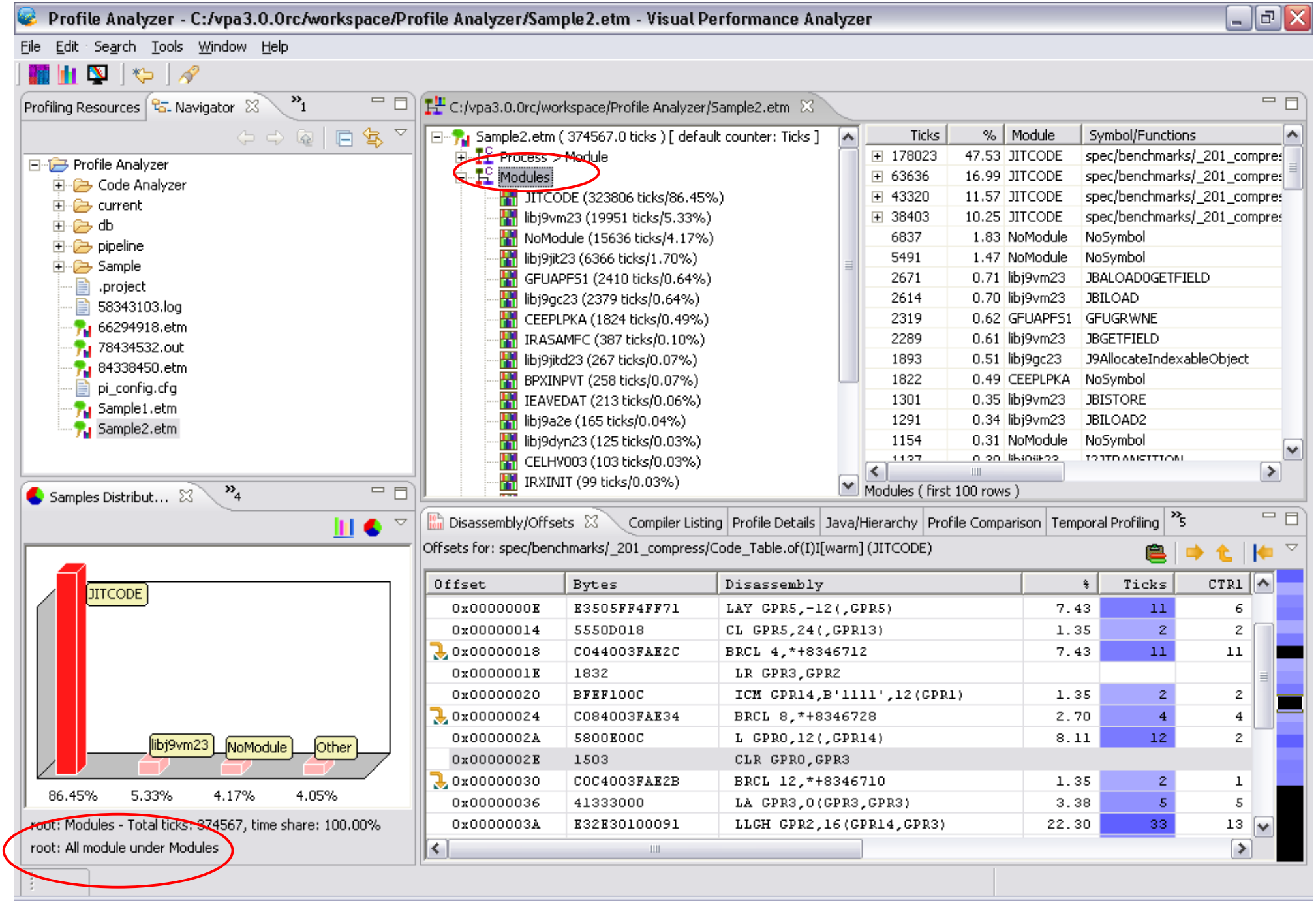
- Better platform coverage
  - Linux PPC and Intel VPA binaries
    - should be available in the next release scheduled for Q4
  - goal is to cover all of IBM's server platforms and Operating Systems
    - internal version already provides wider coverage
  - full static and dynamic code coverage
- Better plugin interaction
  - goal is to provide true point and click drill down experience
  - fully functional remote data collector
- Future plugins
  - PM data postprocessing
  - Trace visualization
  - built-in instruction trace collection and processor models
- Feedback from alphaworks

Recorded demo

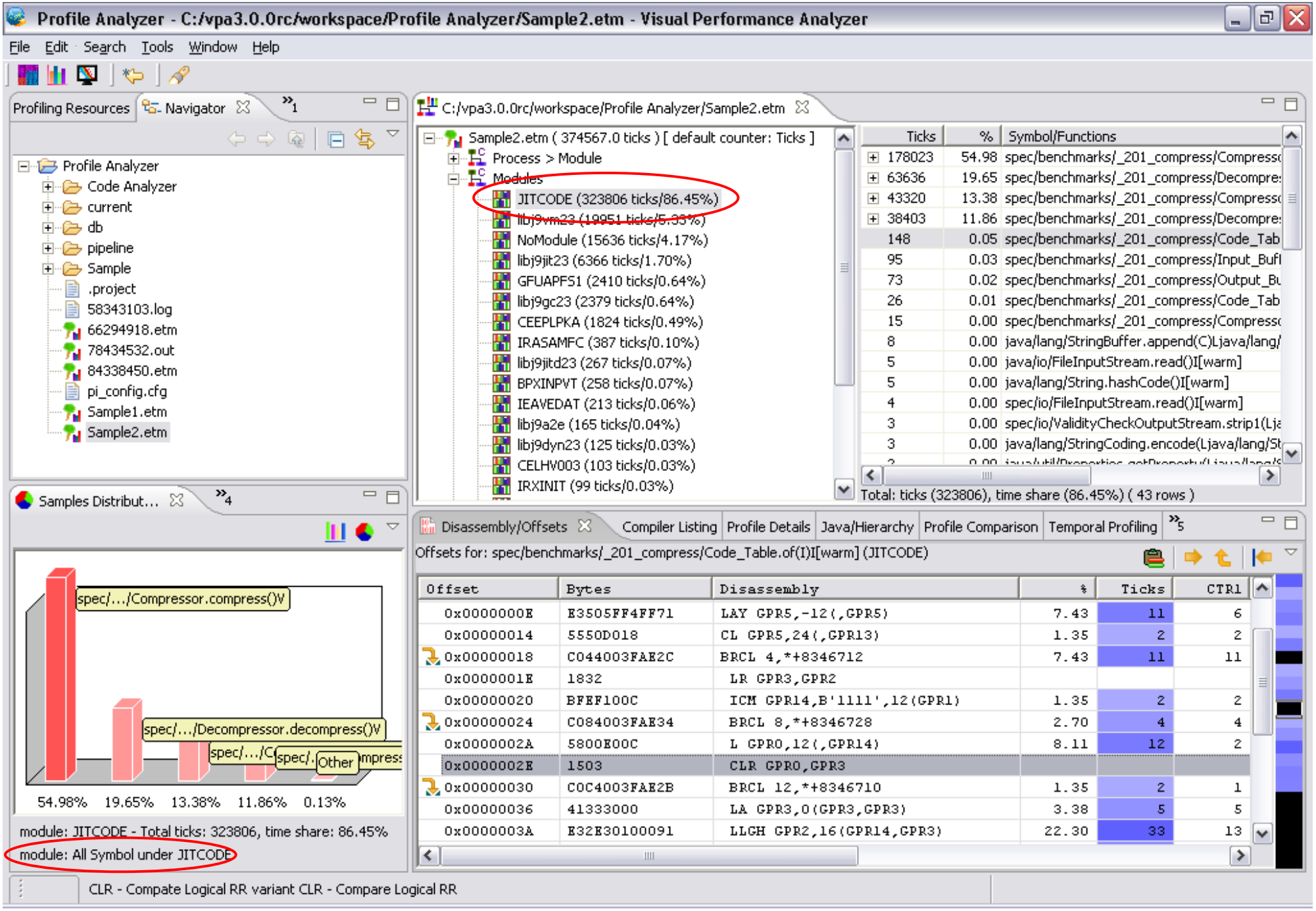
Profile Analyzer - initial view of a profile



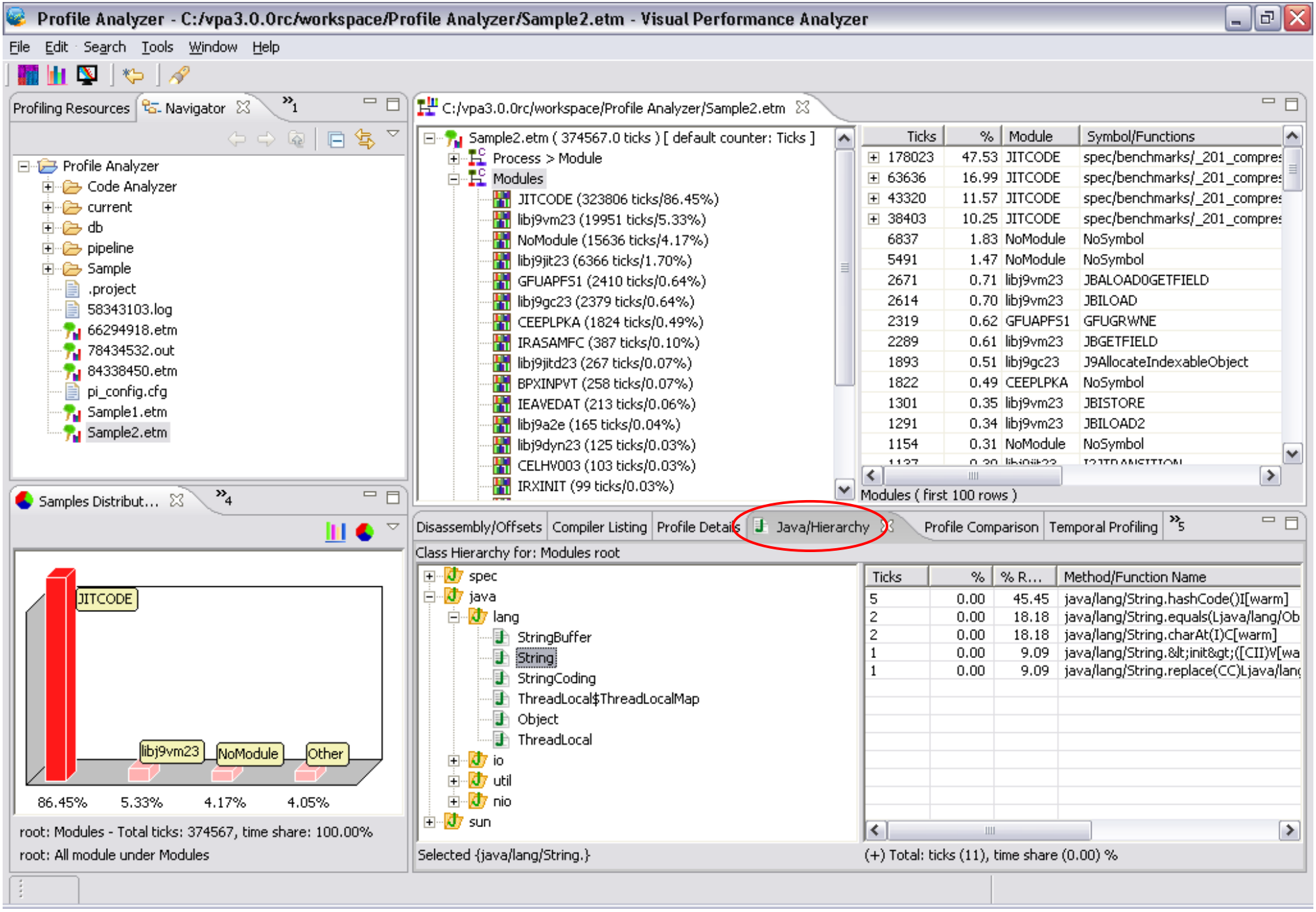
Profile Analyzer - module view



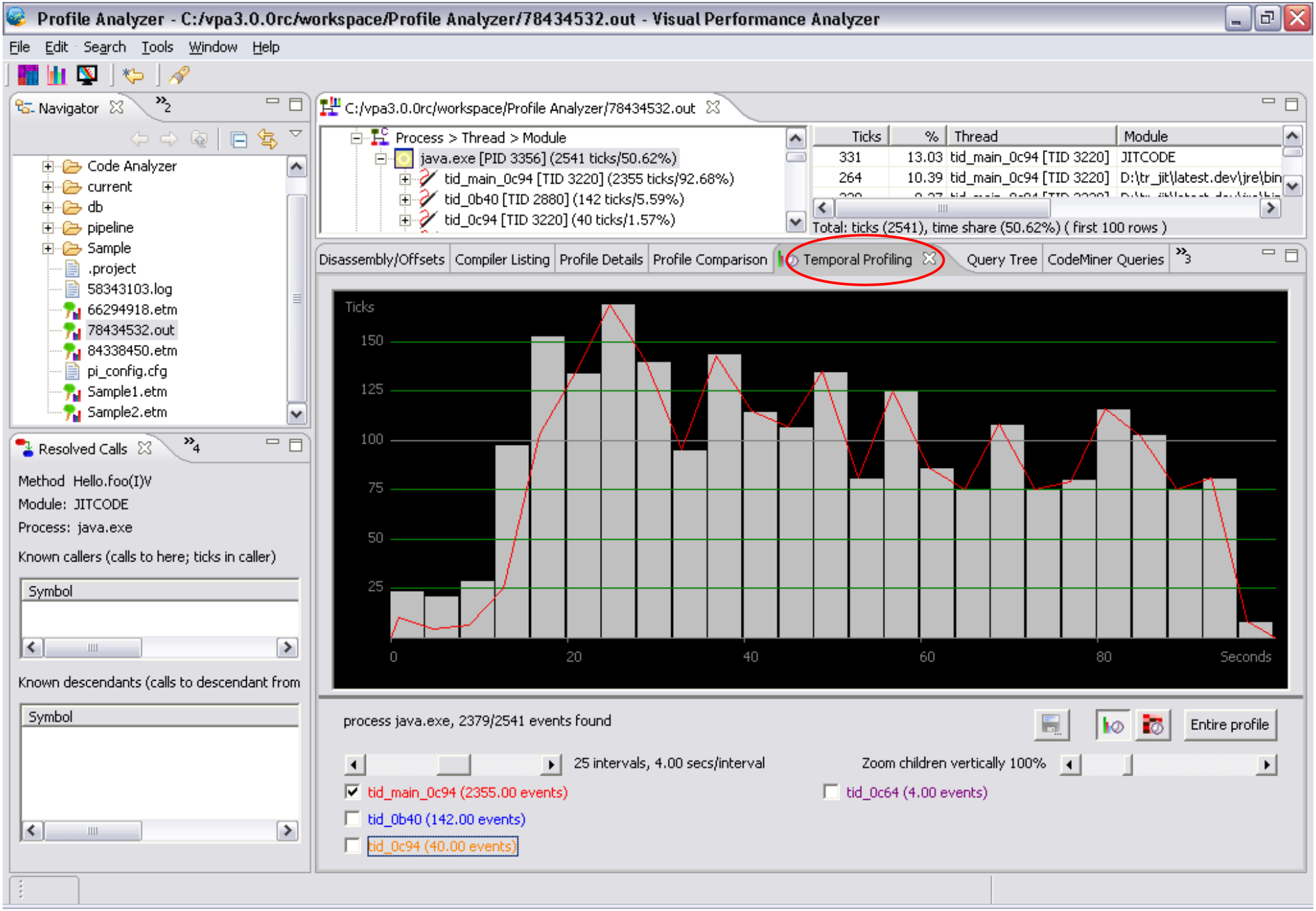
Profile Analyzer – context-sensitive module graph



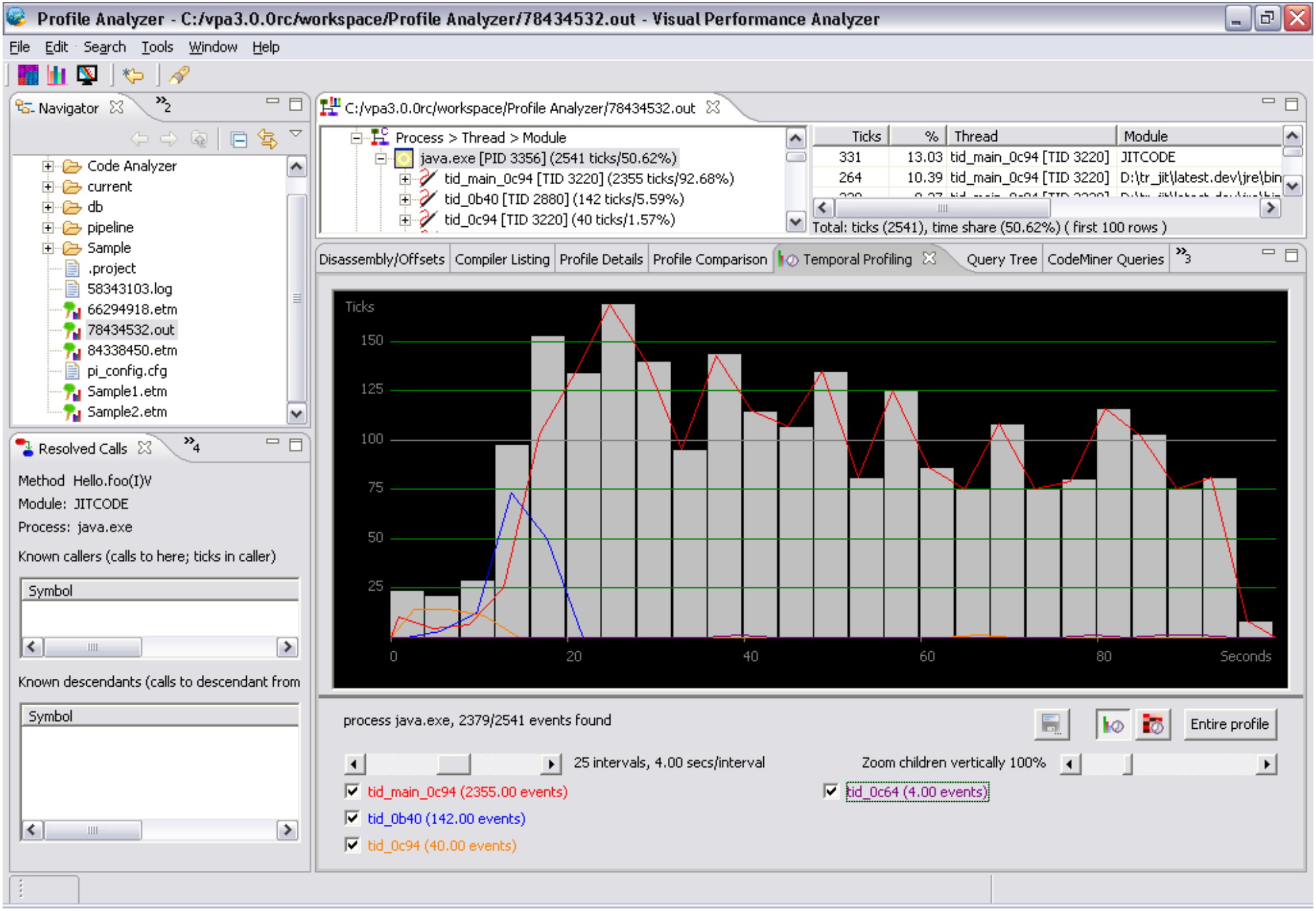
Profile Analyzer - Java class hierarchy view



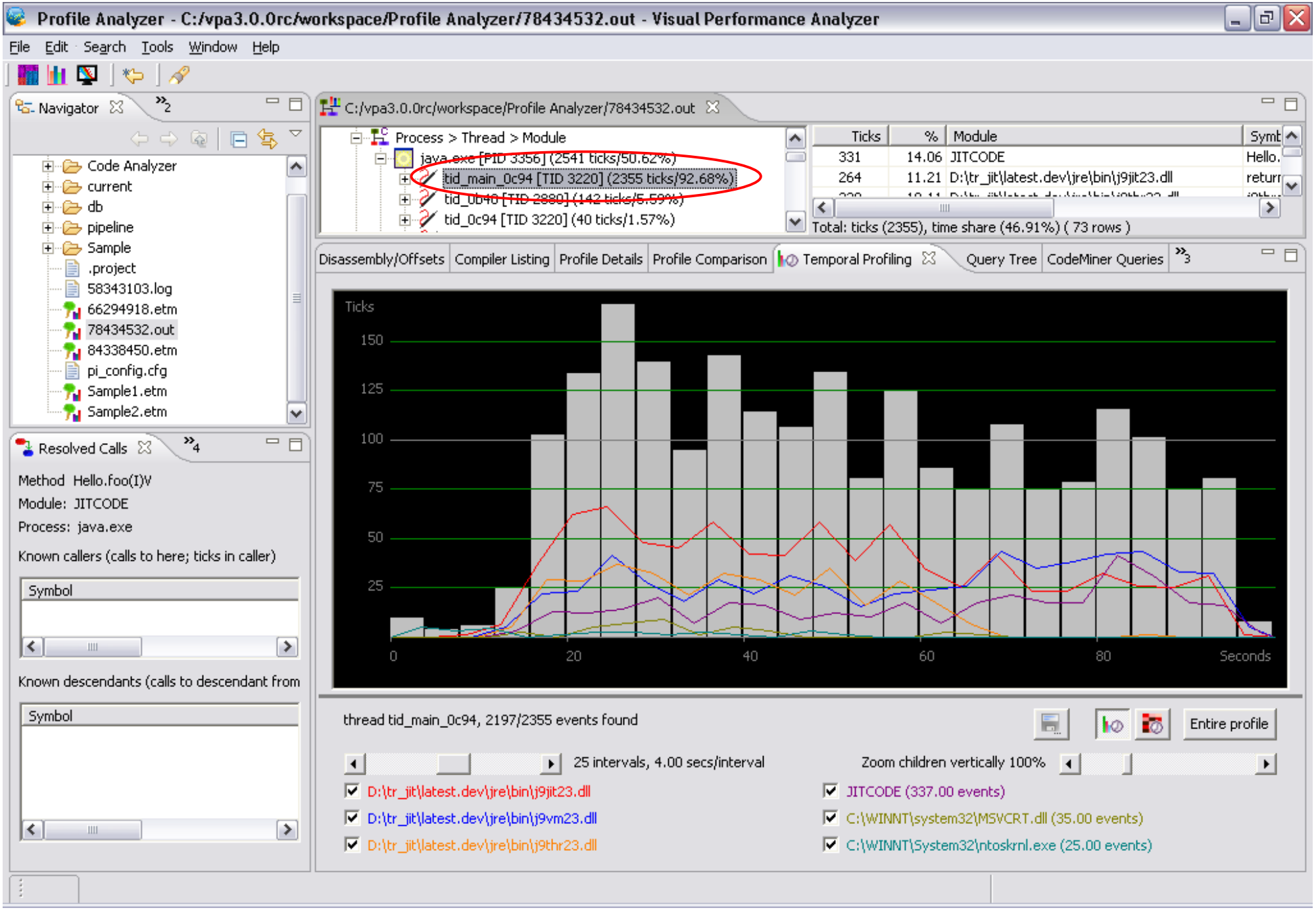
Profile Analyzer - temporal profiling: main process



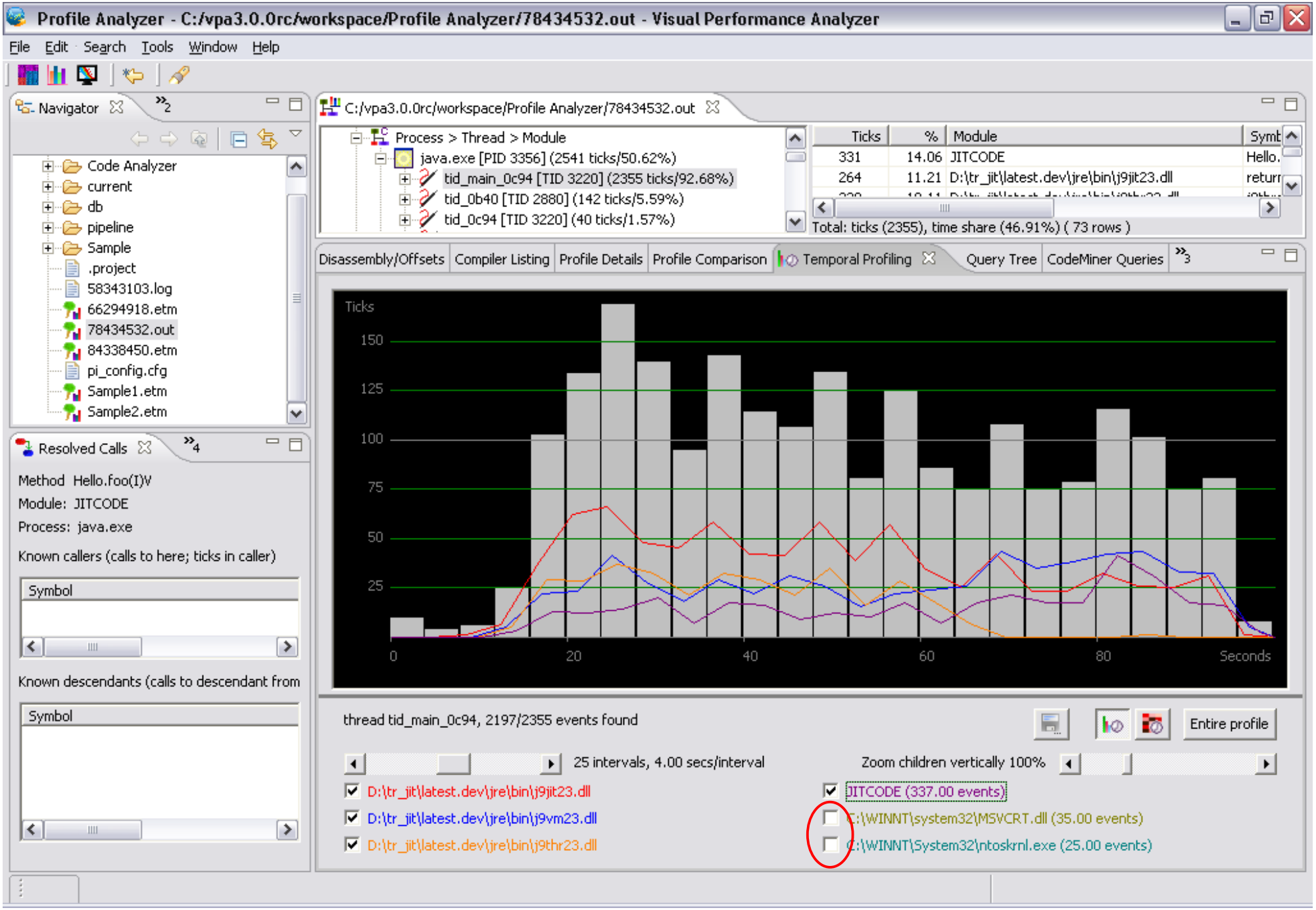
Profile Analyzer - main process with children threads



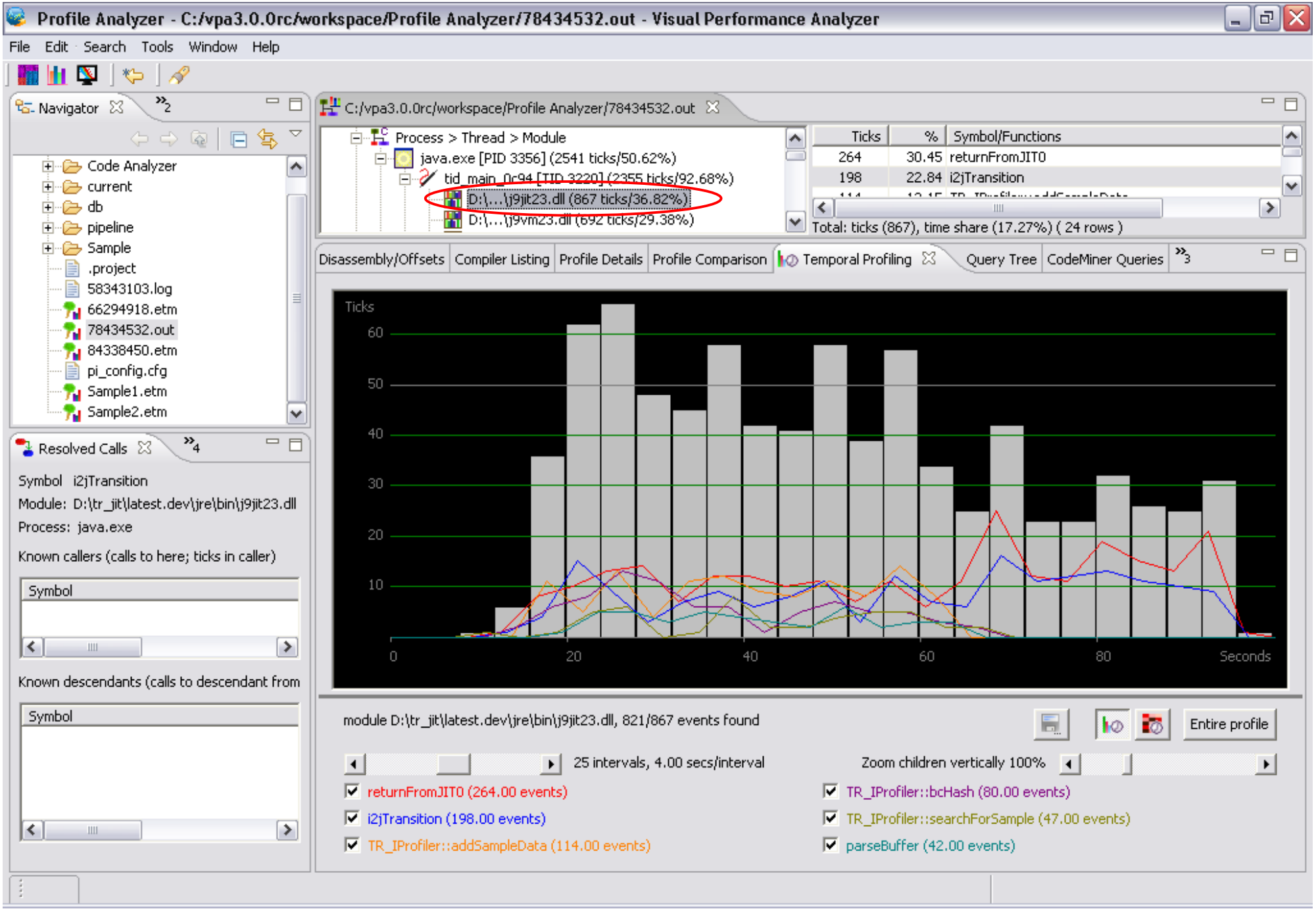
Profile Analyzer - top thread with children modules



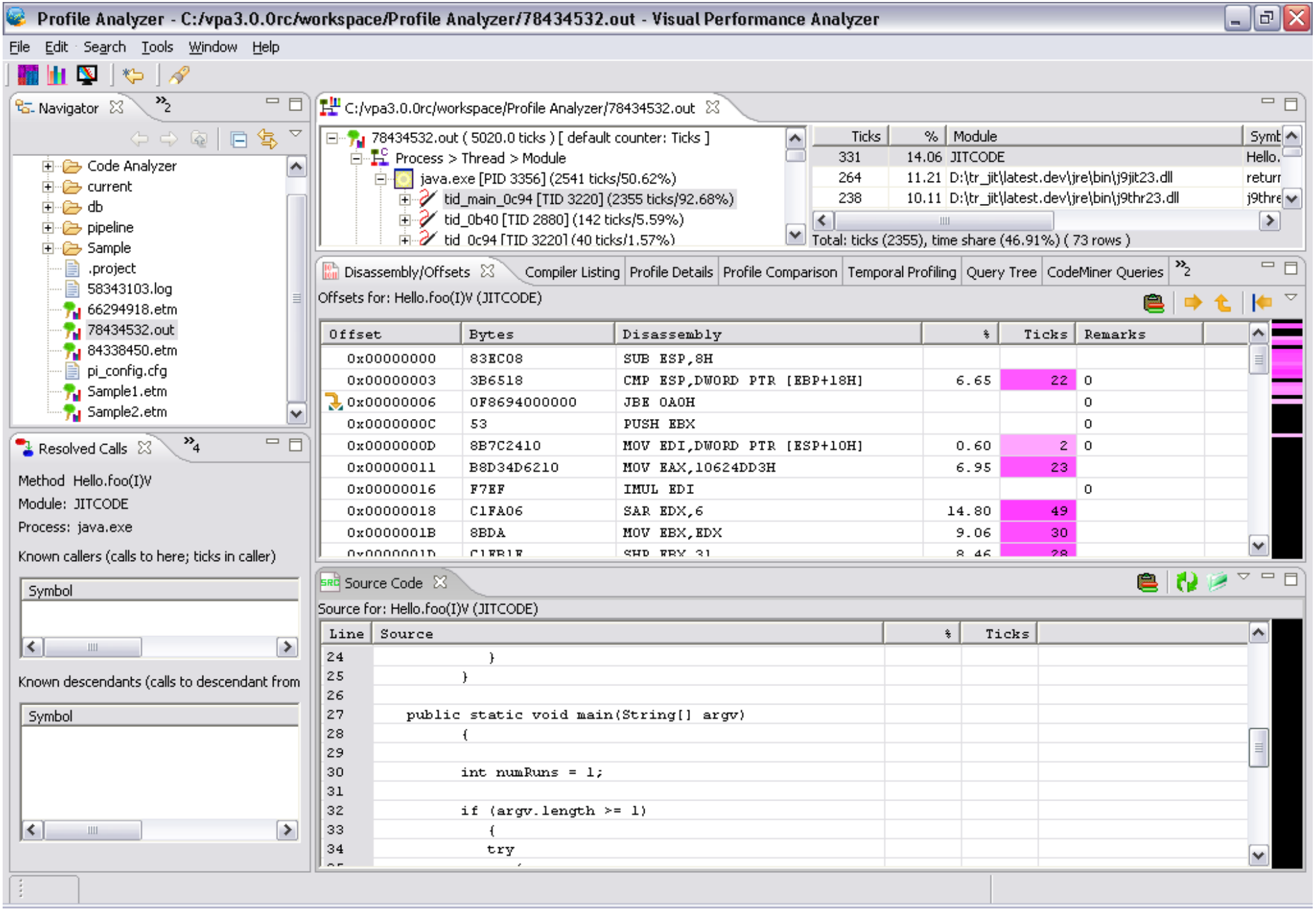
Profile Analyzer - hide lower-hit children



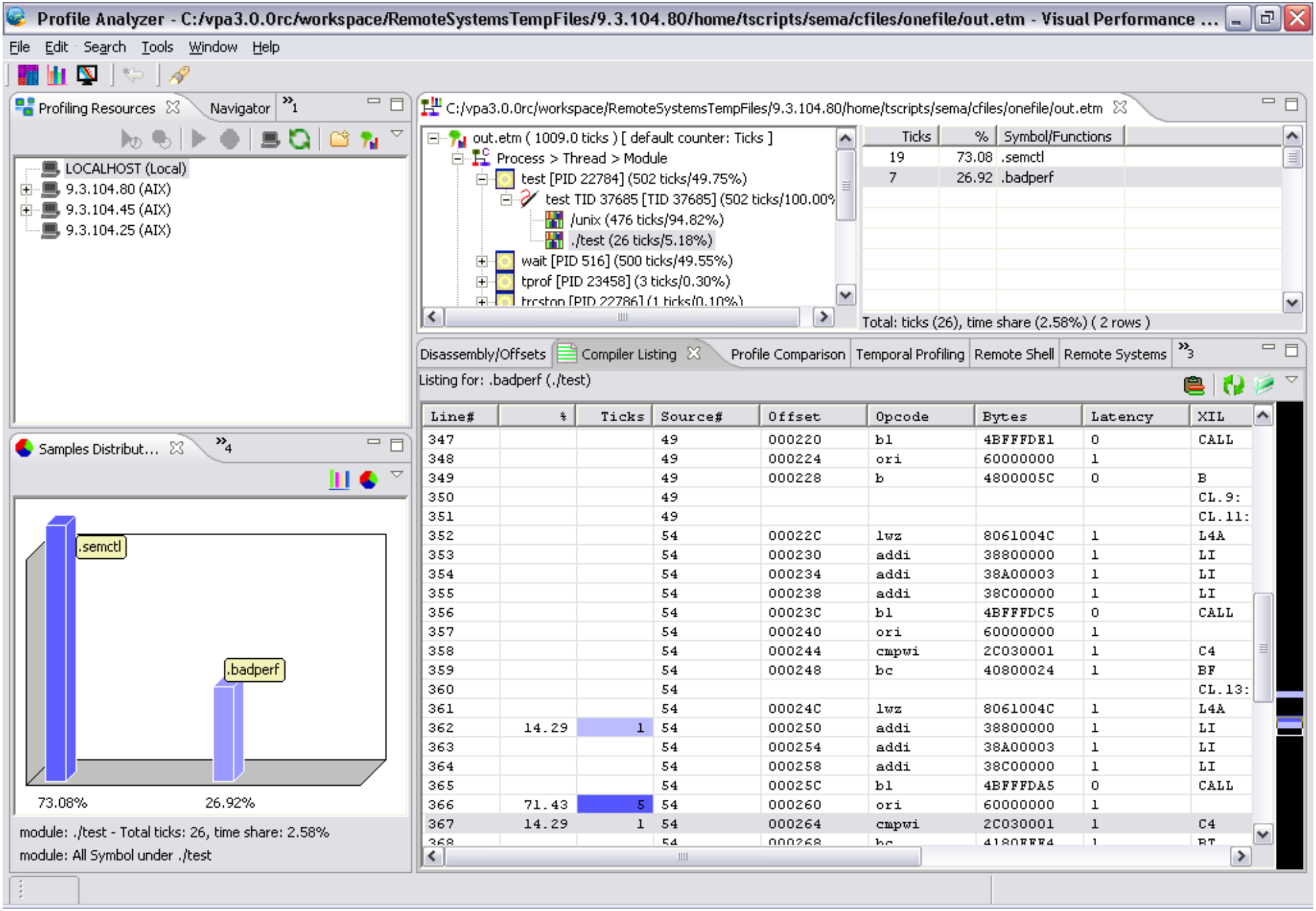
Profile Analyzer - symbols in a module



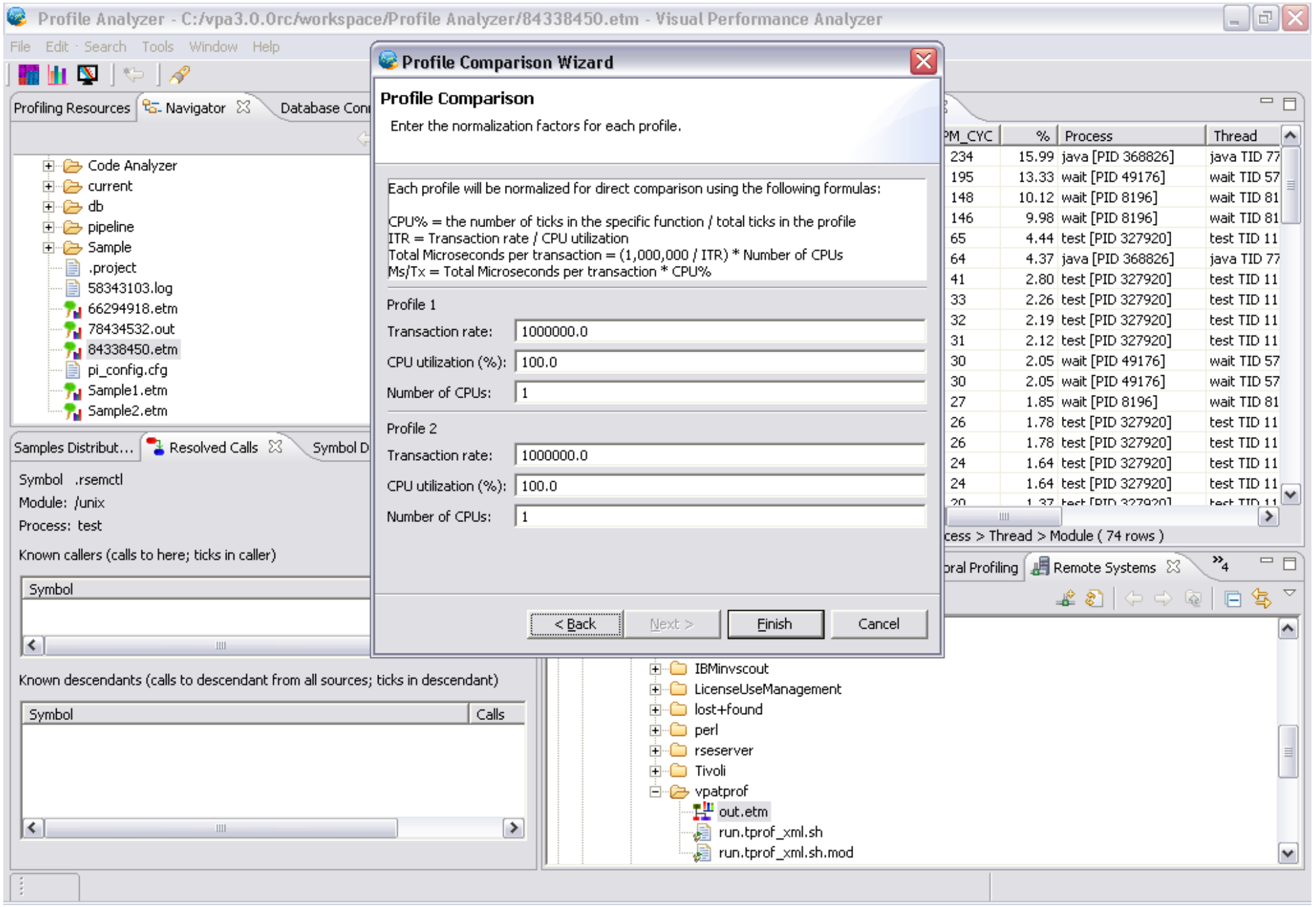
Profile Analyzer - source and disassembly view



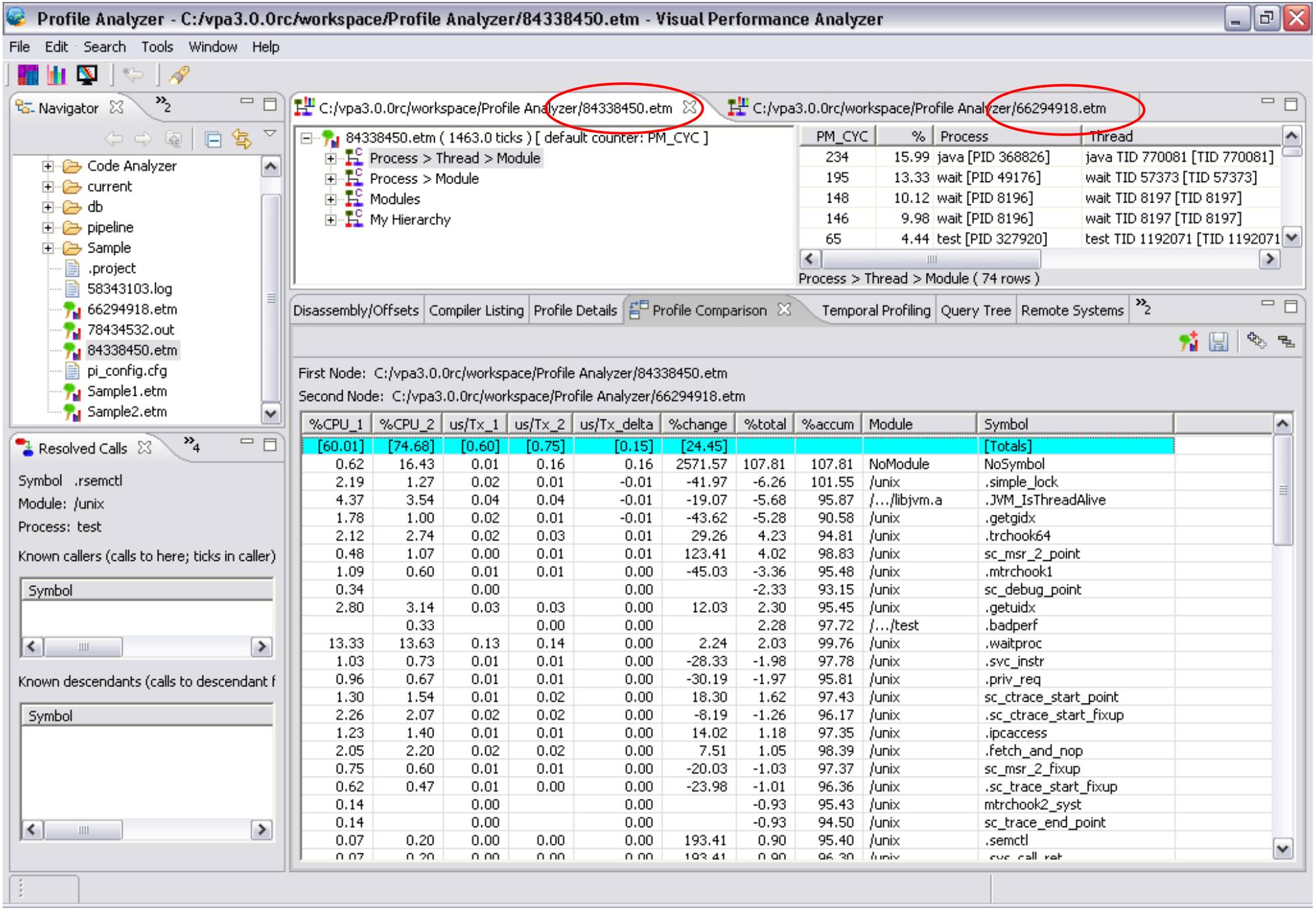
Profile Analyzer – compiler listing view



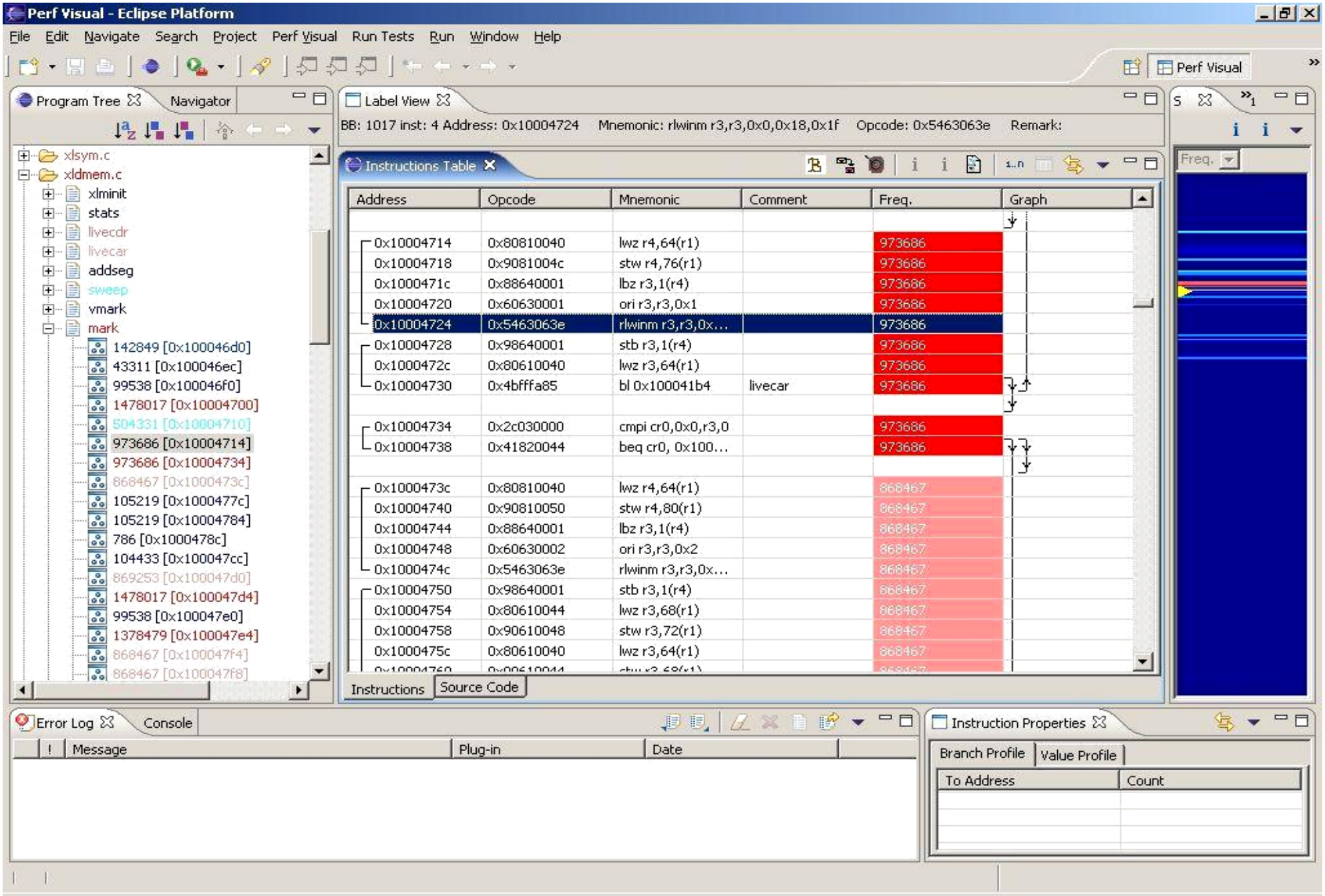
Profile Analyzer - normalization factors for comparisons



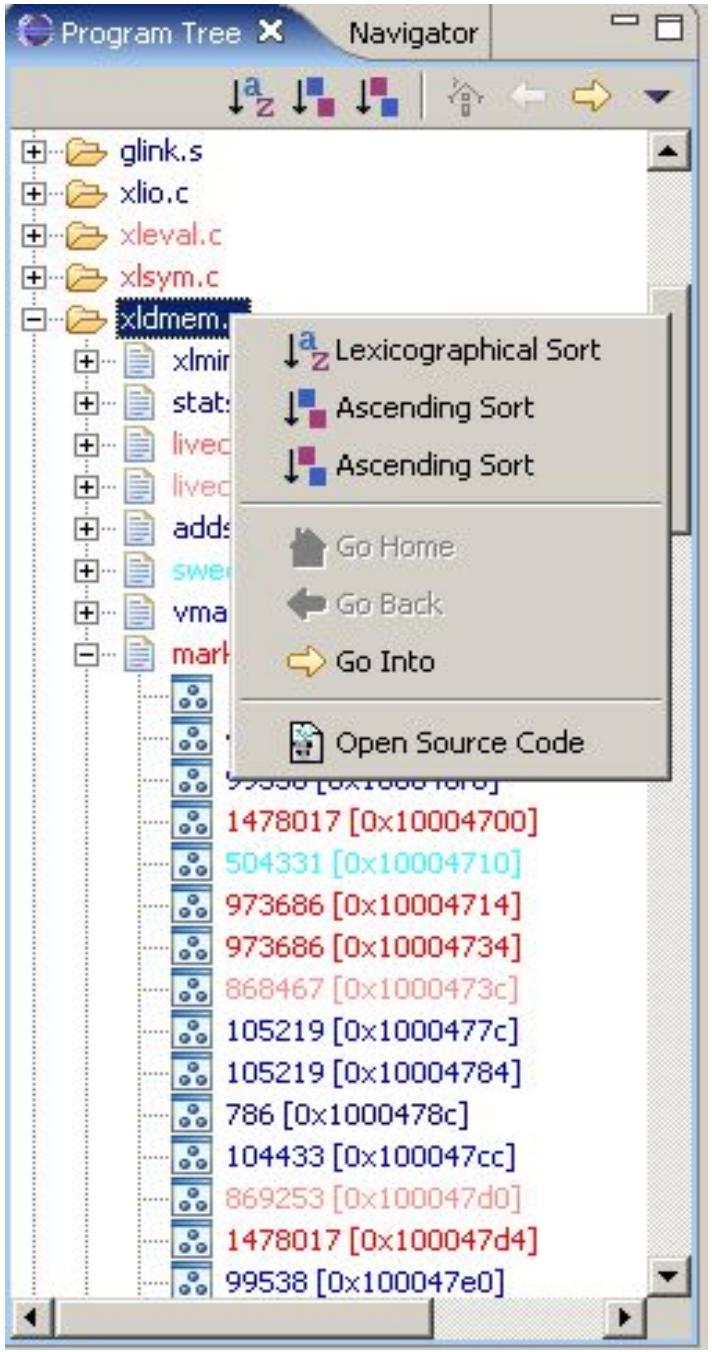
Profile Analyzer - profile comparison chart



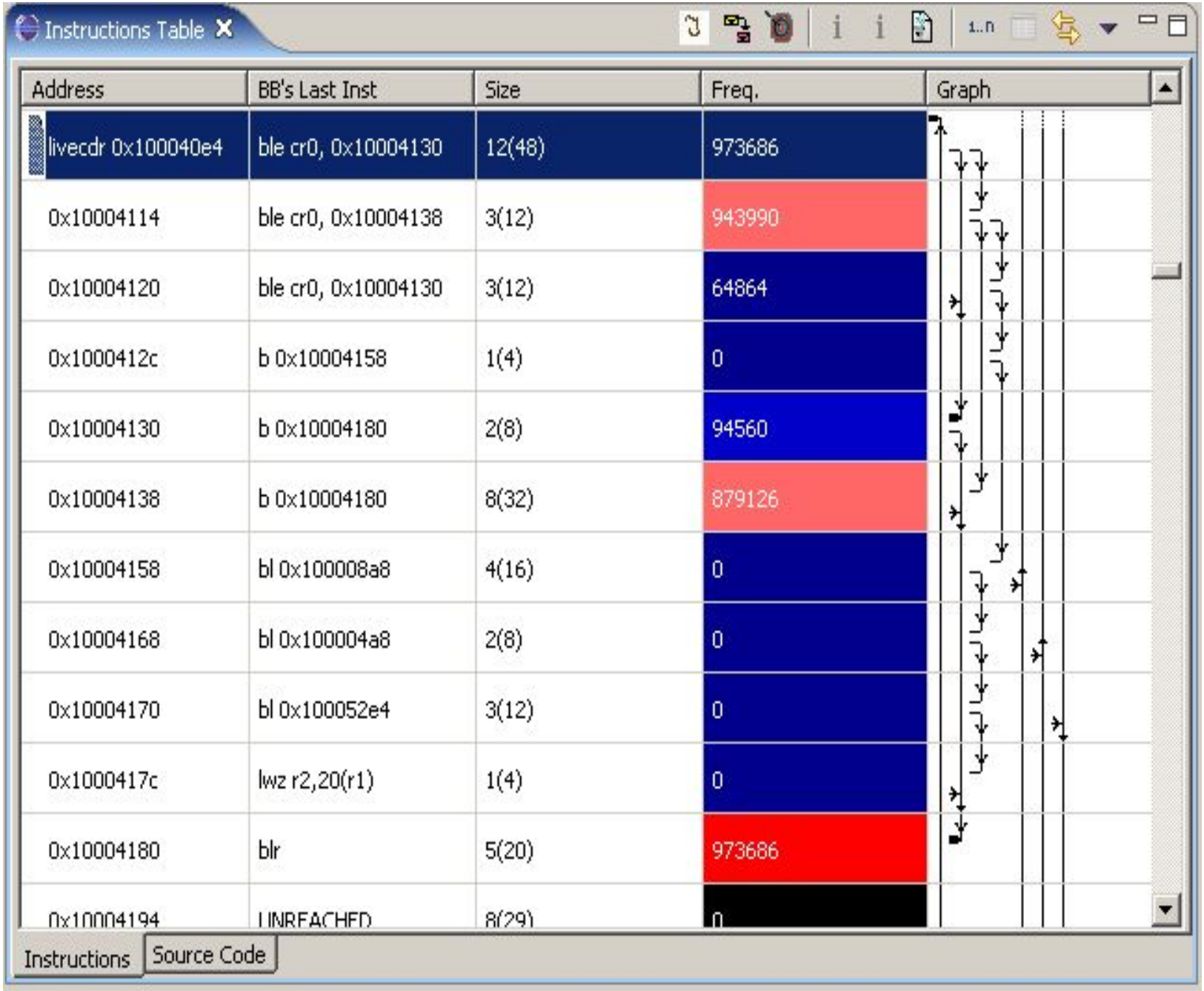
Code Analyzer - main view



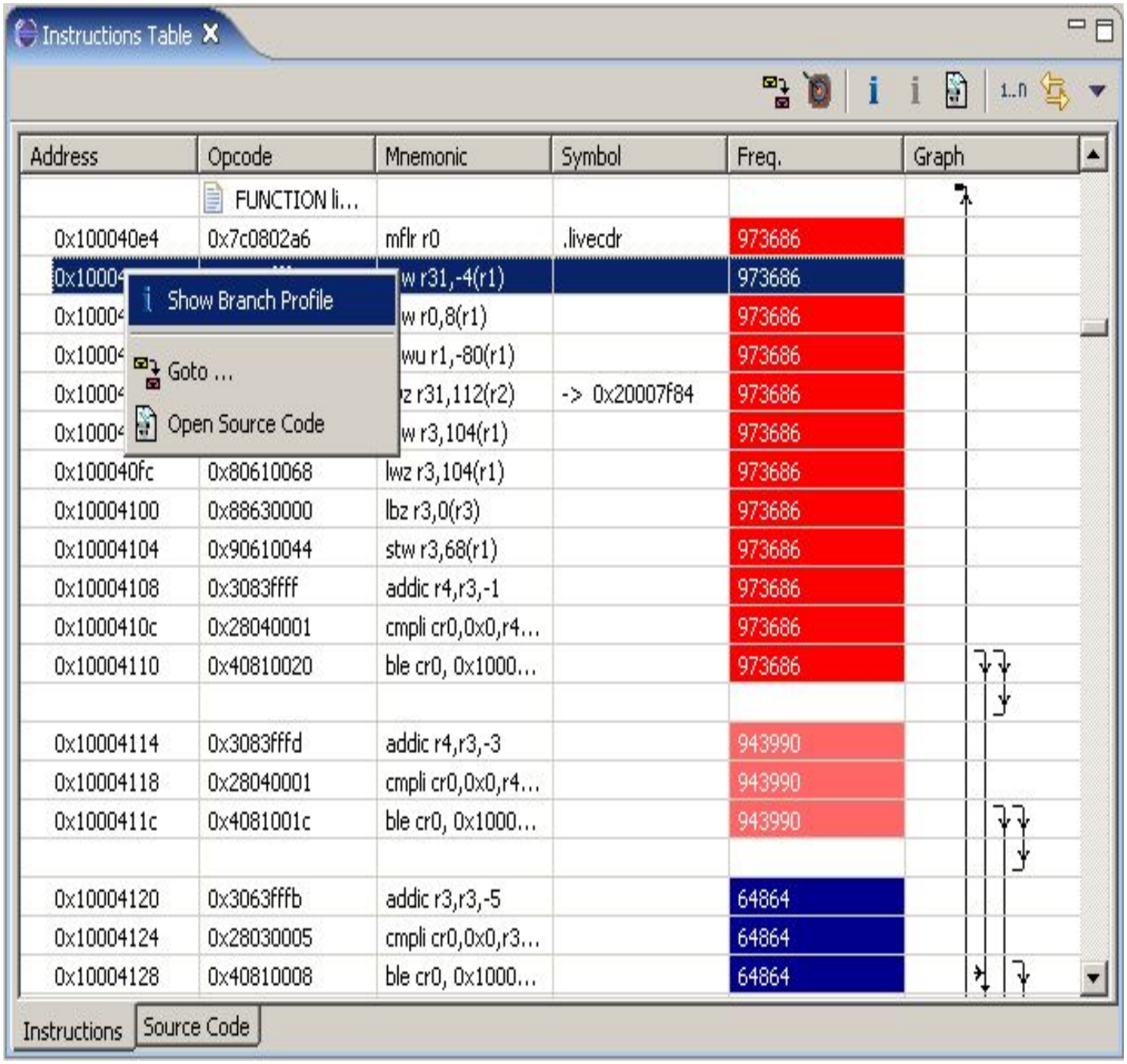
Code Analyzer - tree view

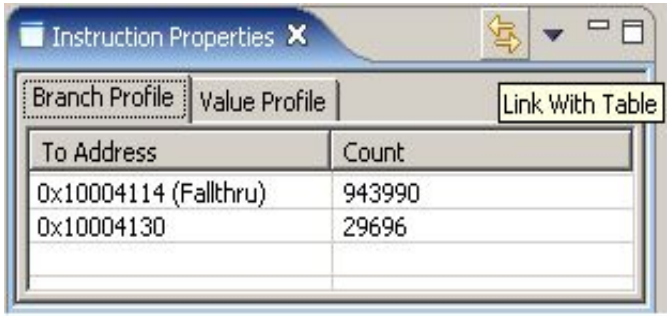


Code Analyzer - basic block view

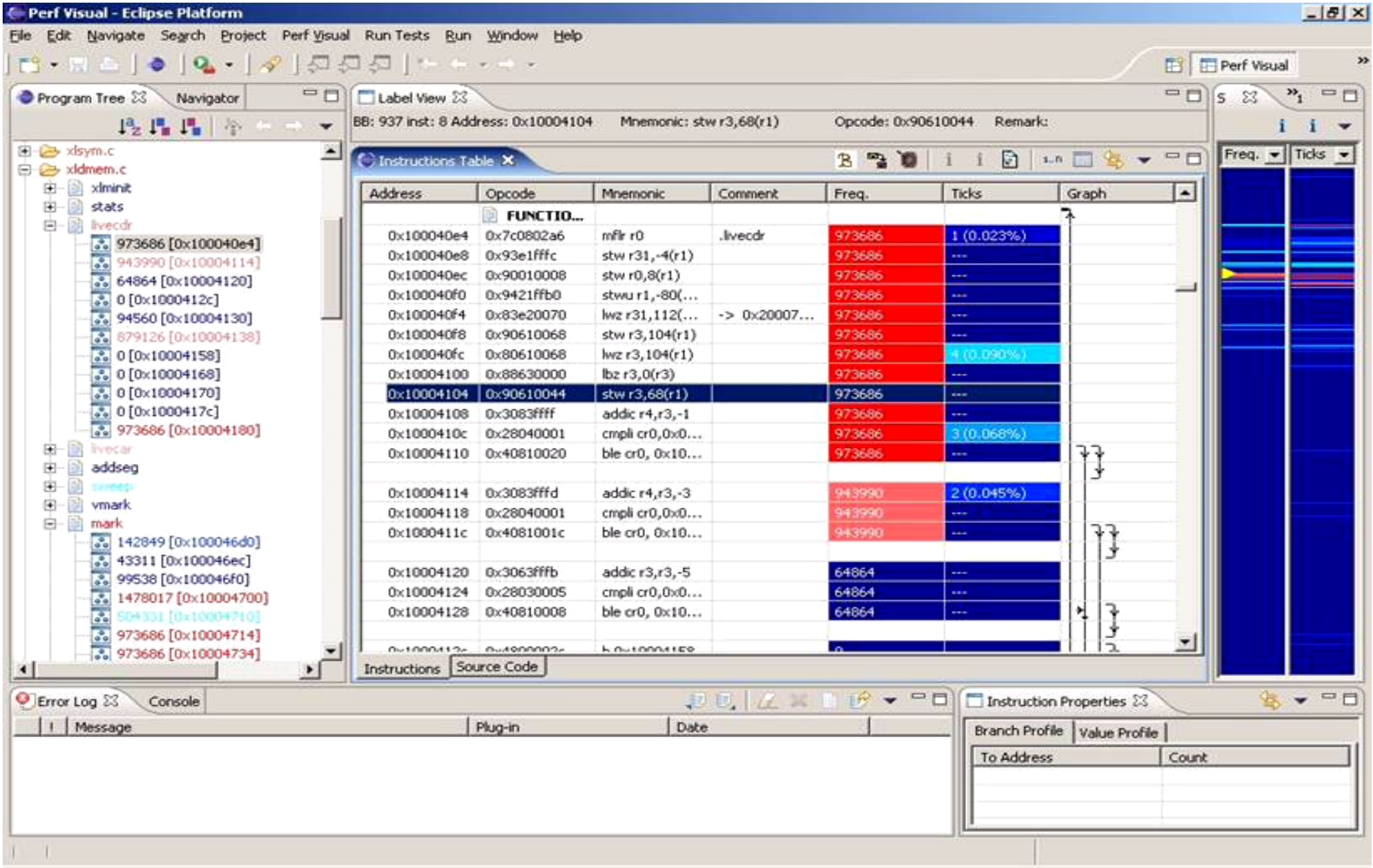


Code Analyzer - branch profiling

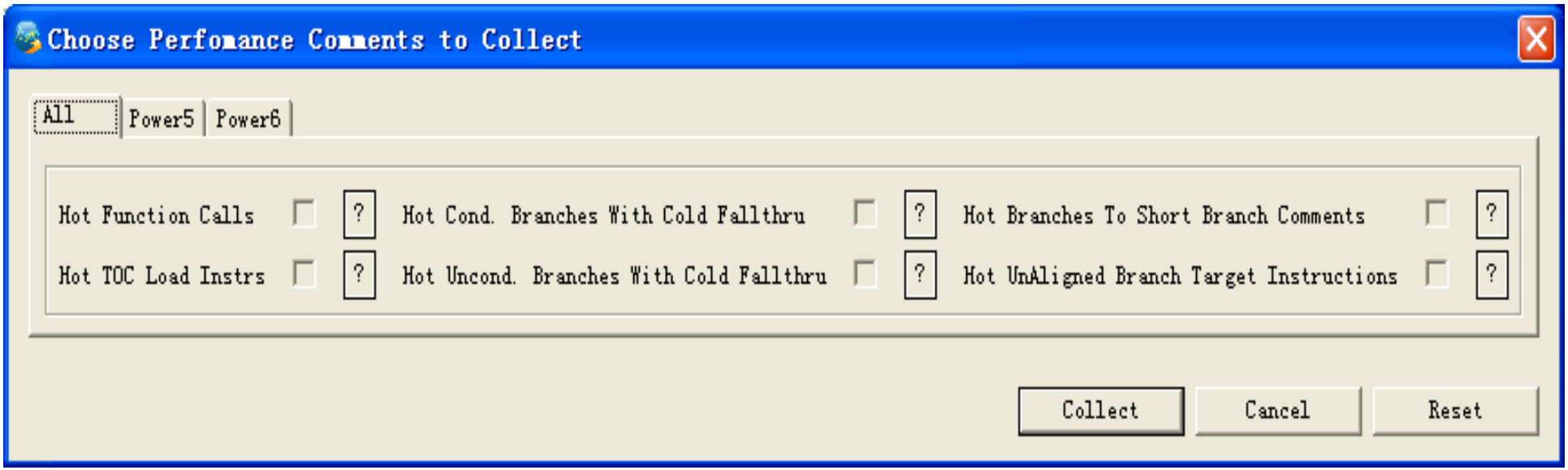




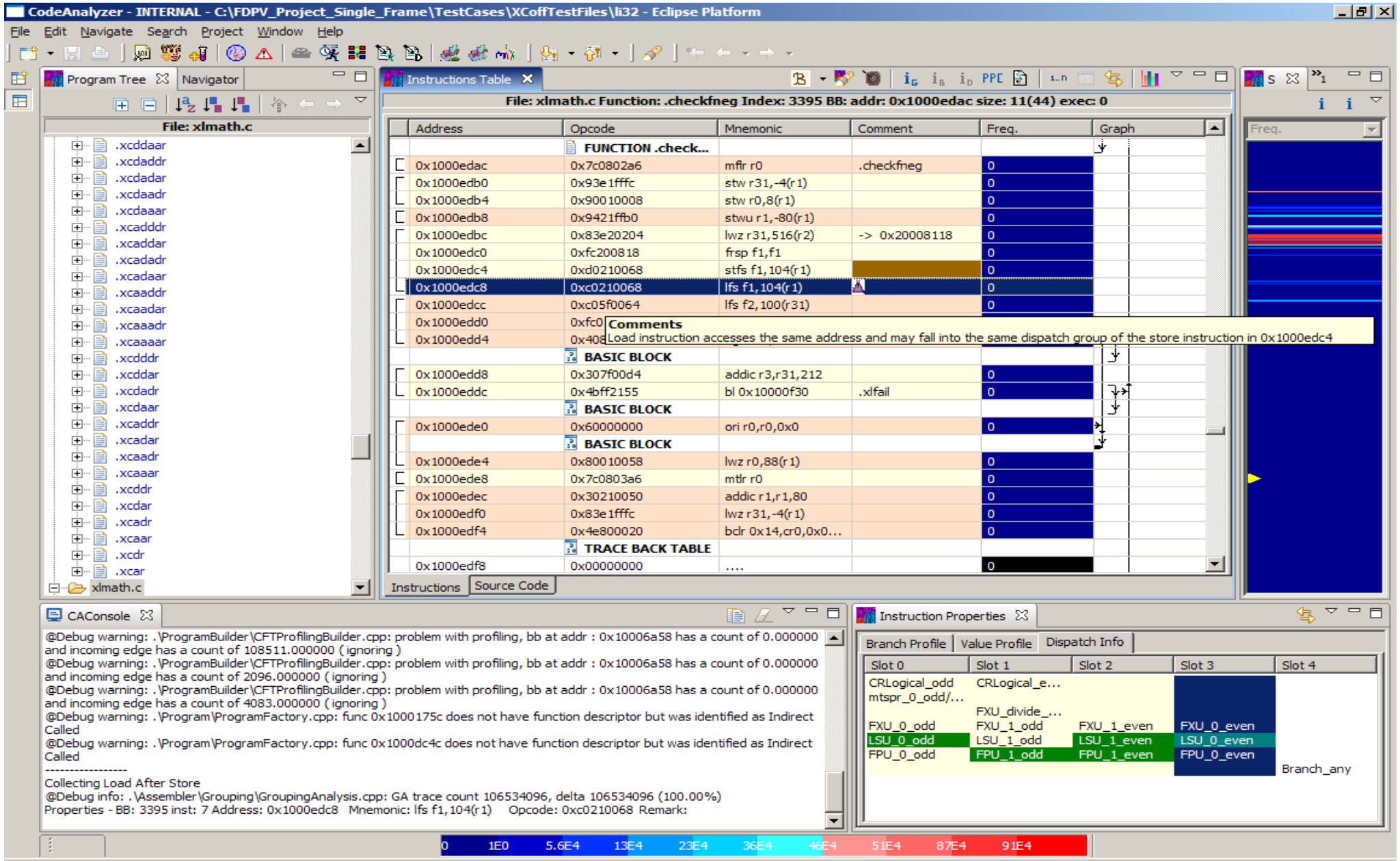
Code Analyzer - frequency and tprof profiling view



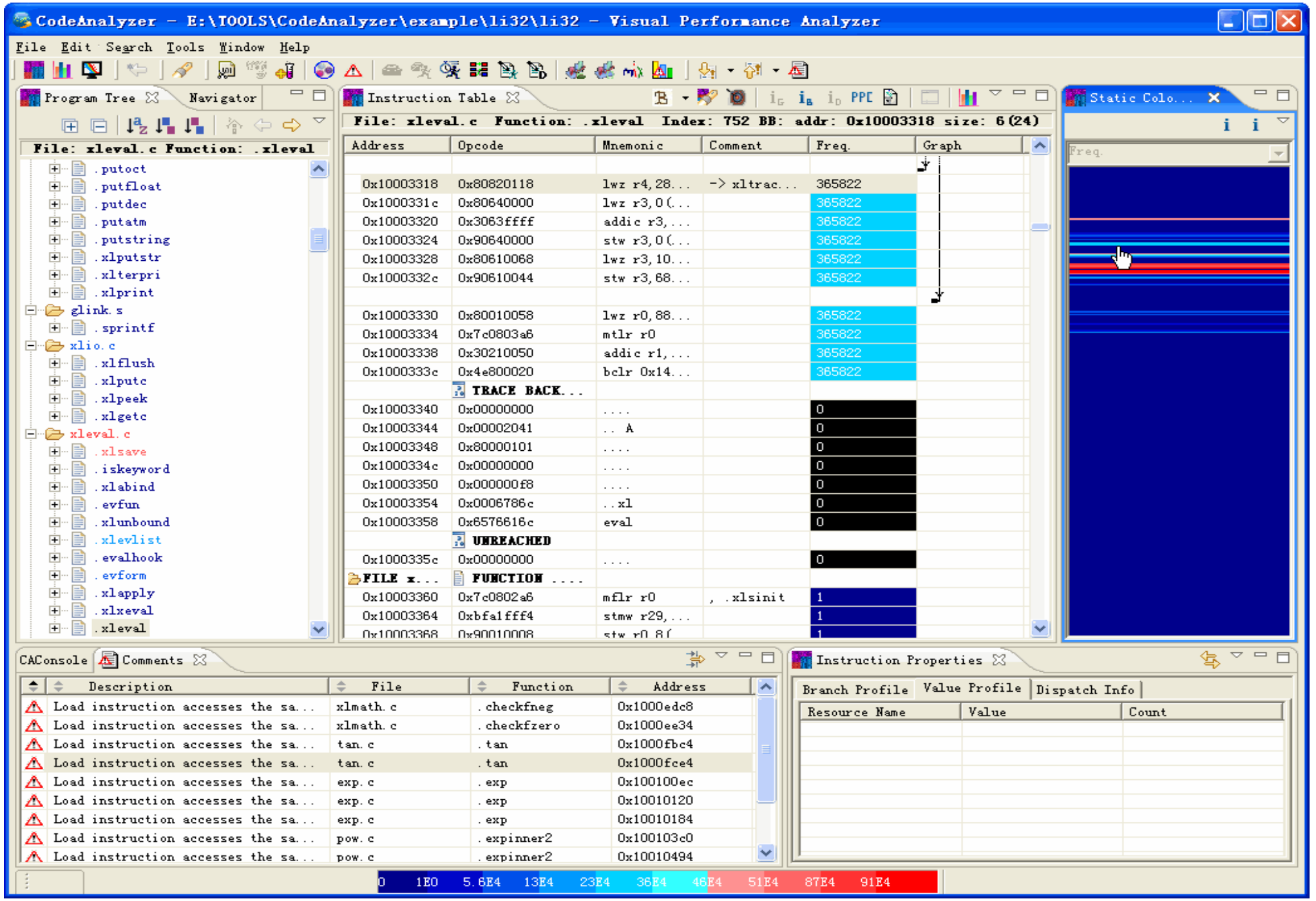
Code Analyzer – collecting hazard information



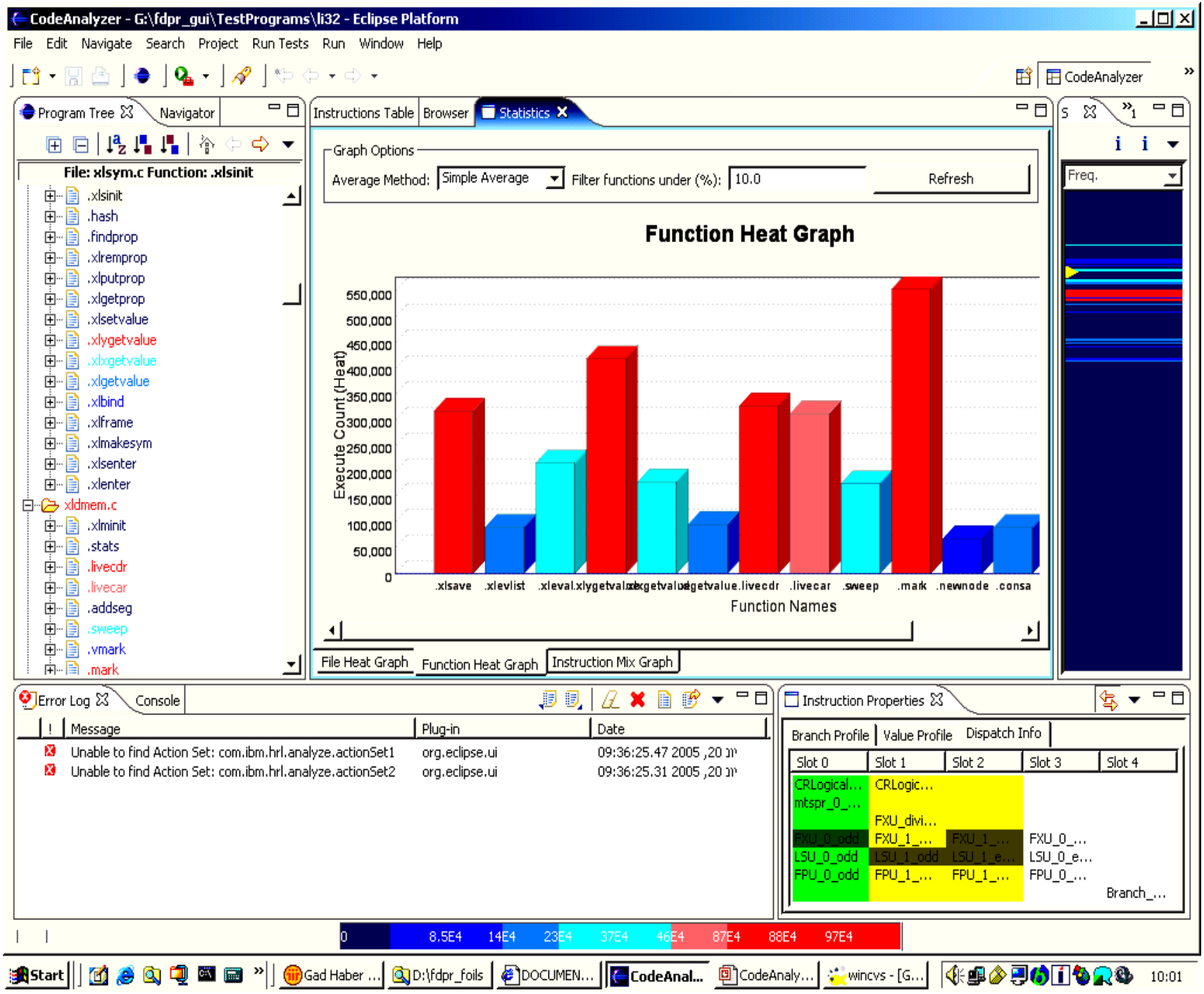
Code Analyzer – comment and instruction grouping view



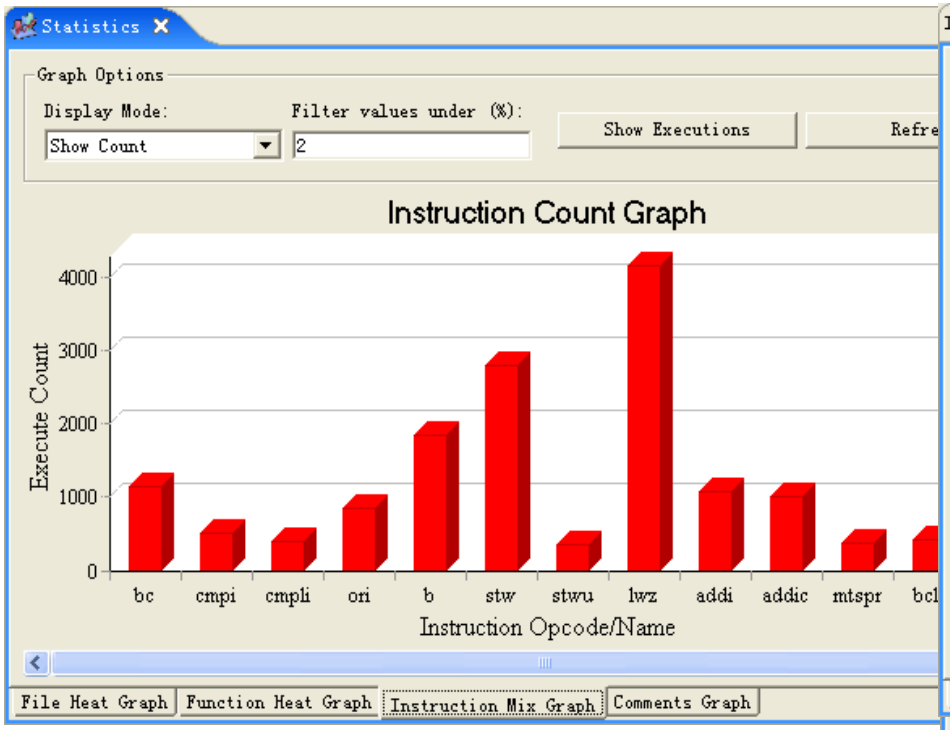
Code Analyzer - comments and instructions properties

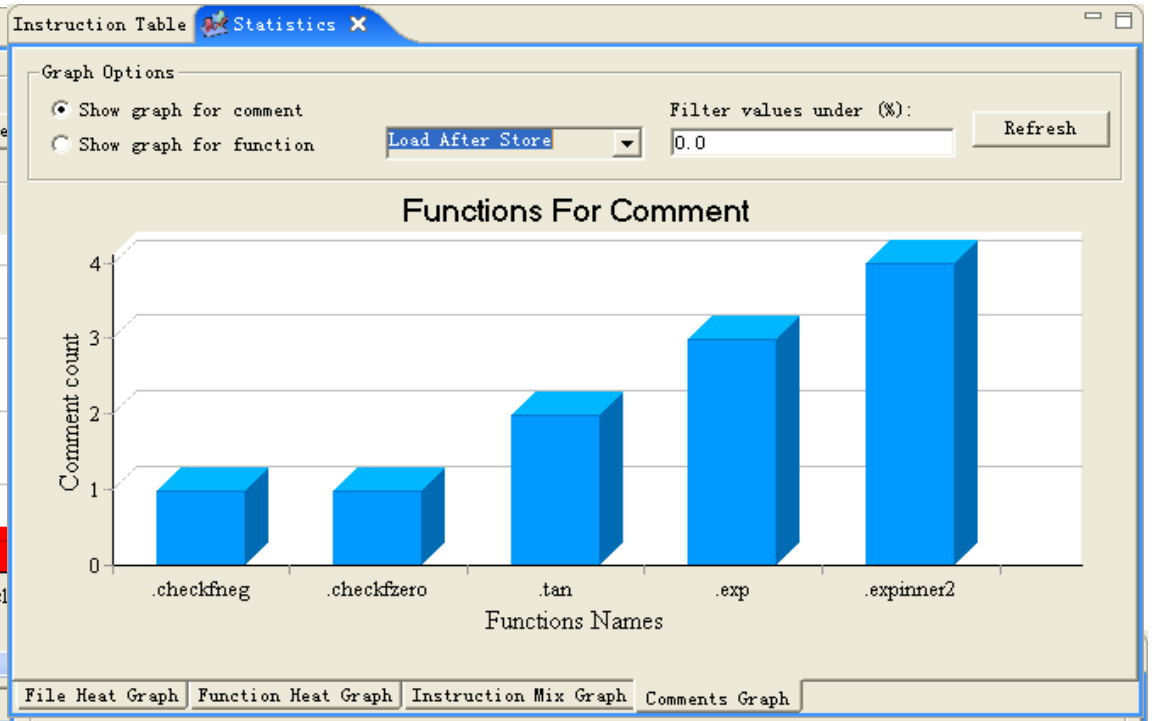


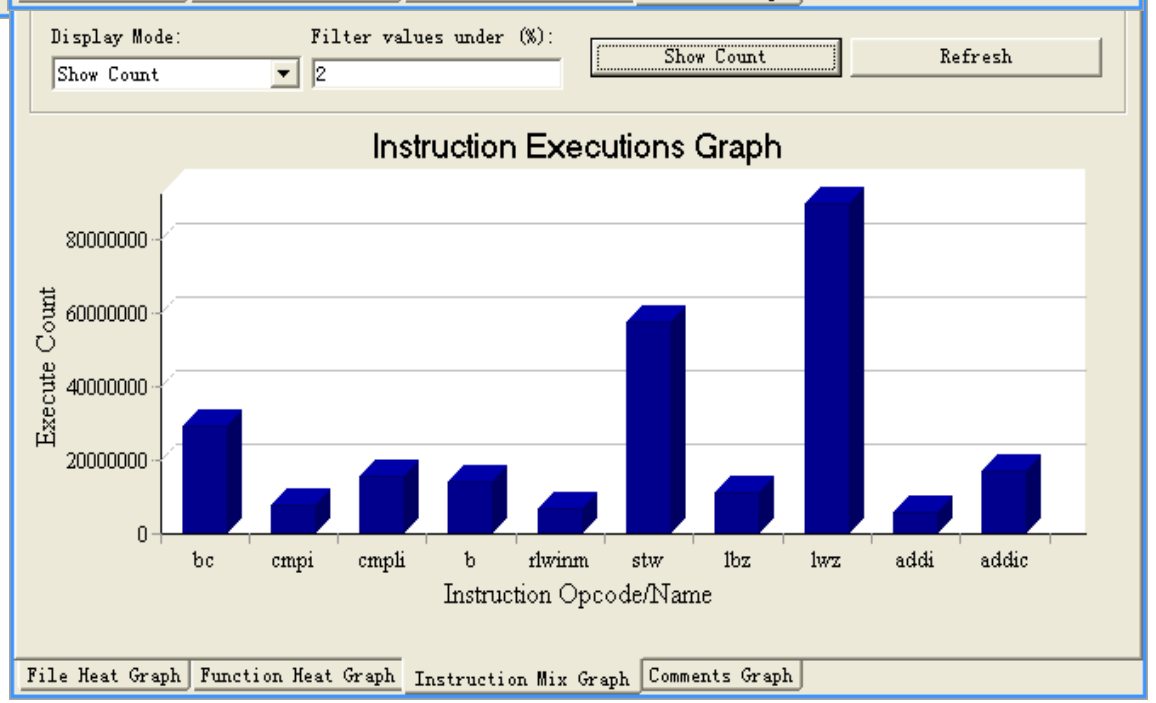
Code Analyzer - statistics view



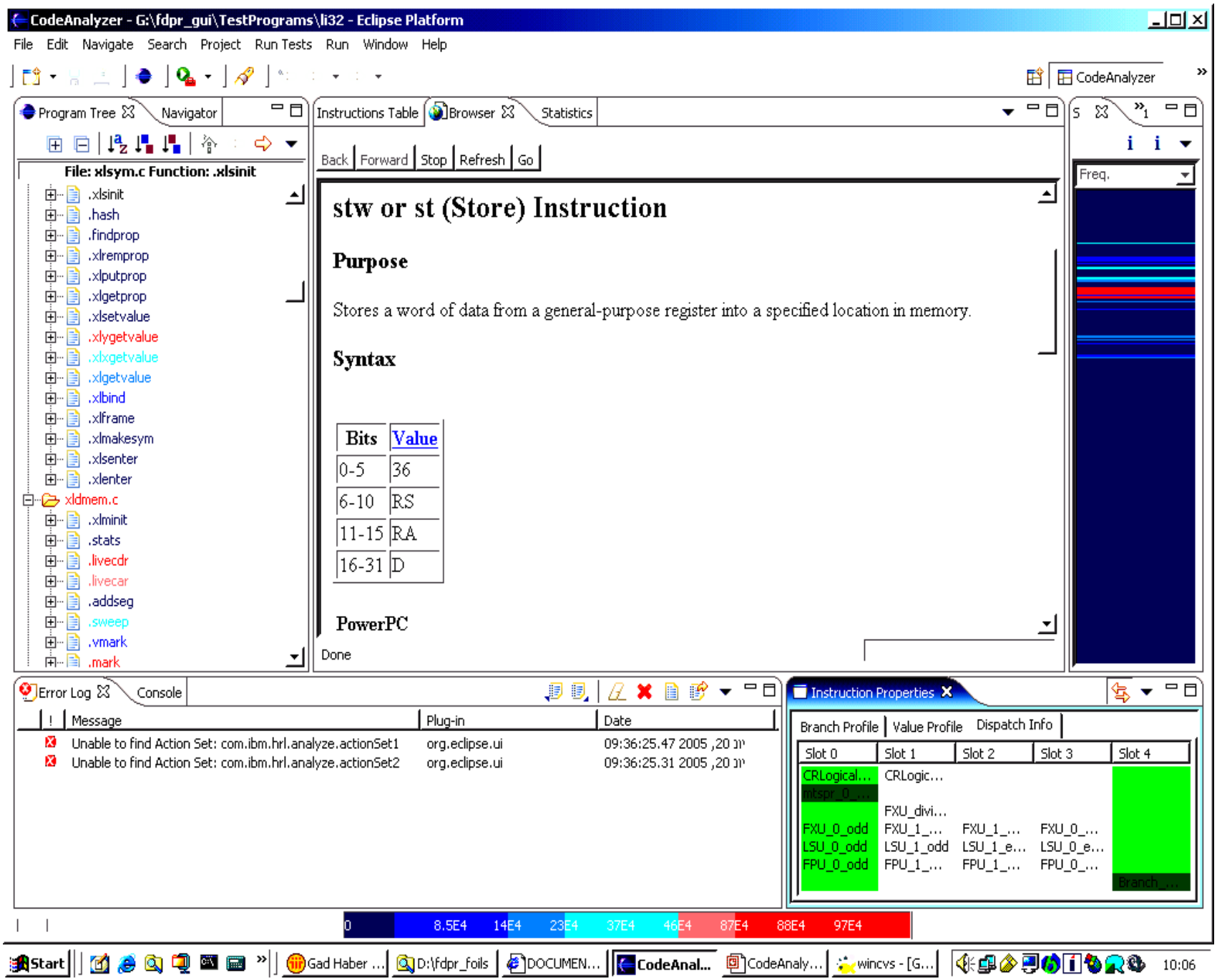
Code Analyzer – more statistics views



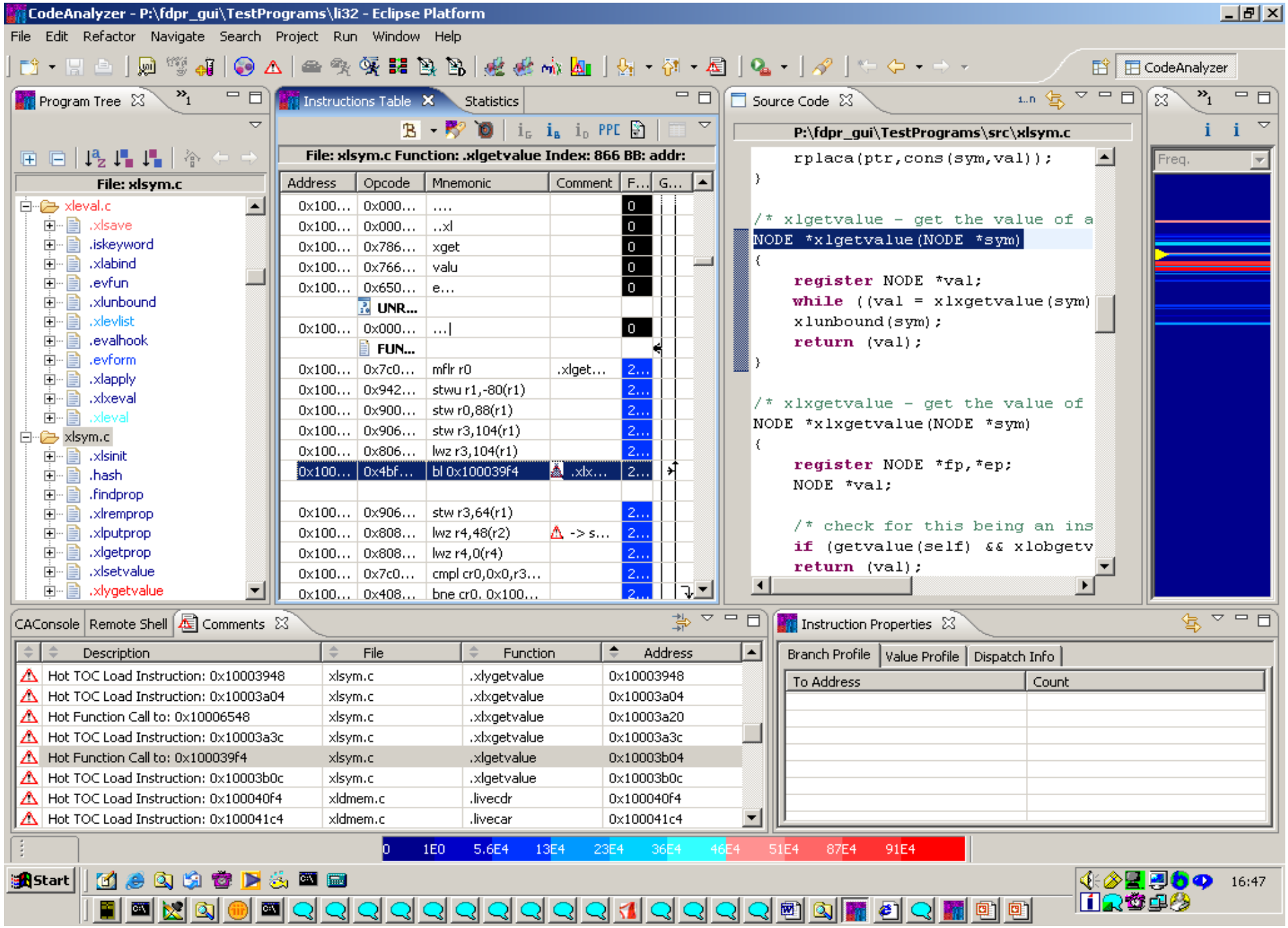




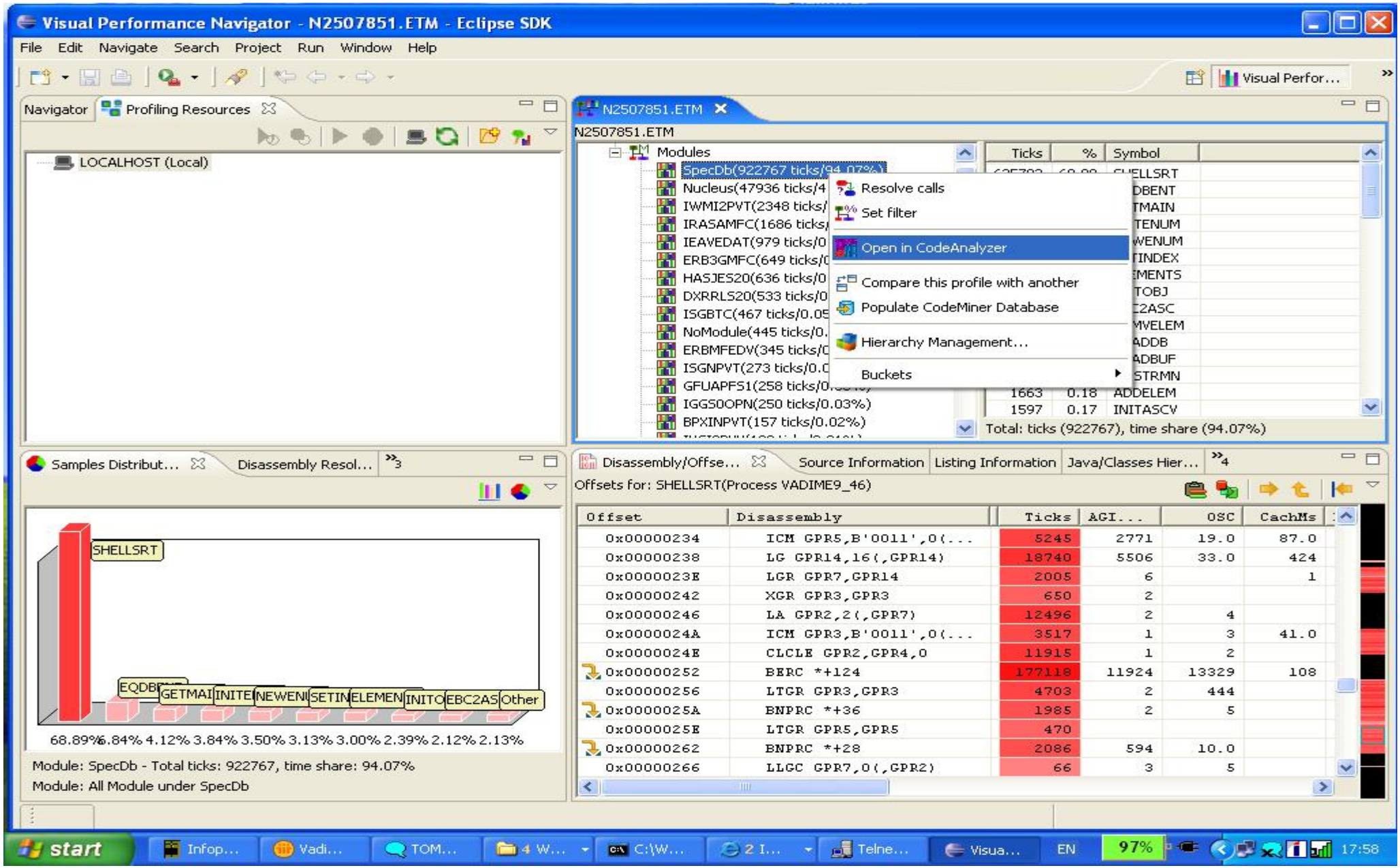
Code Analyzer - PowerPC instructions help view



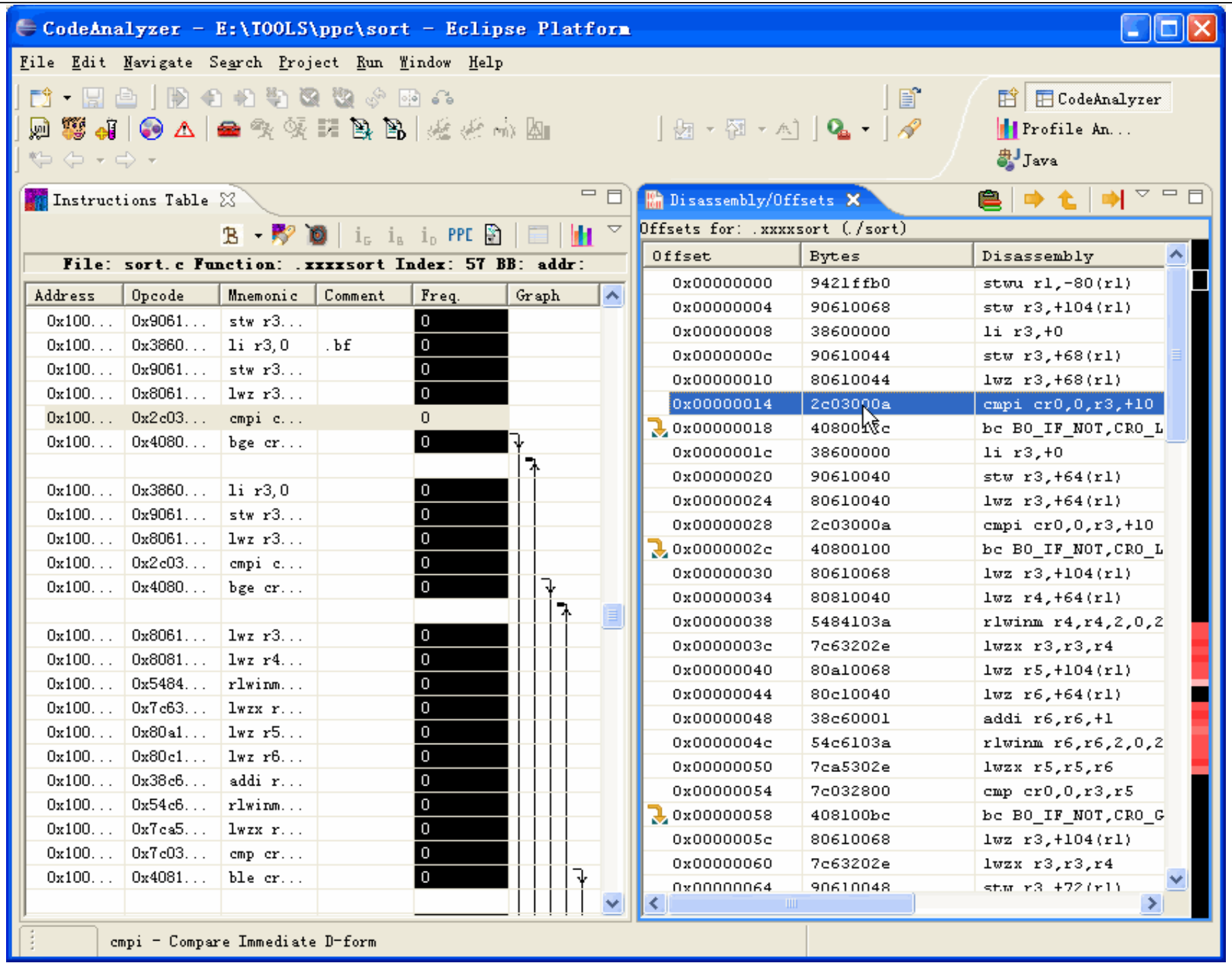
Code Analyzer - instructions, source & comments view



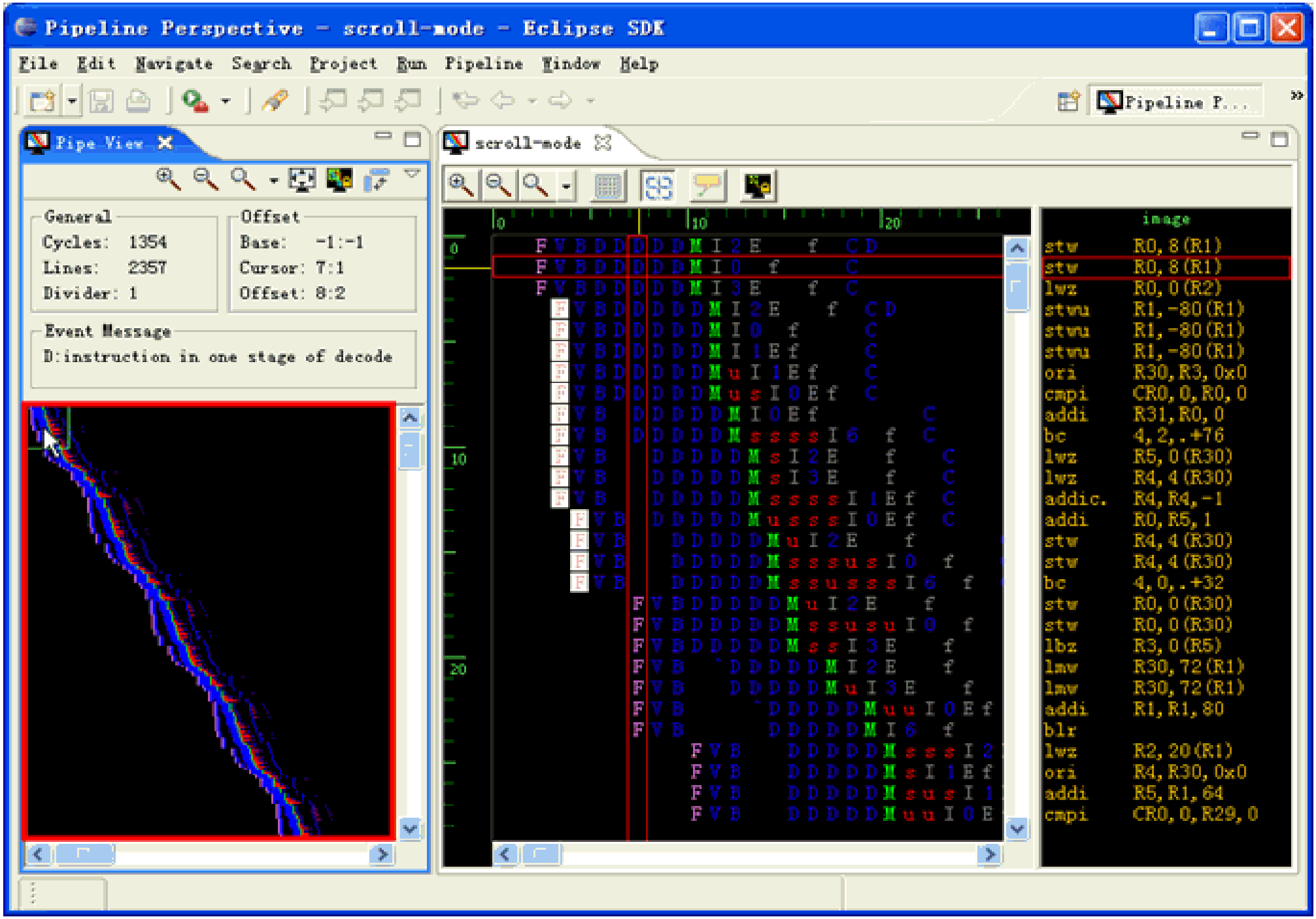
Invoking Code Analyzer from Profile Analyzer



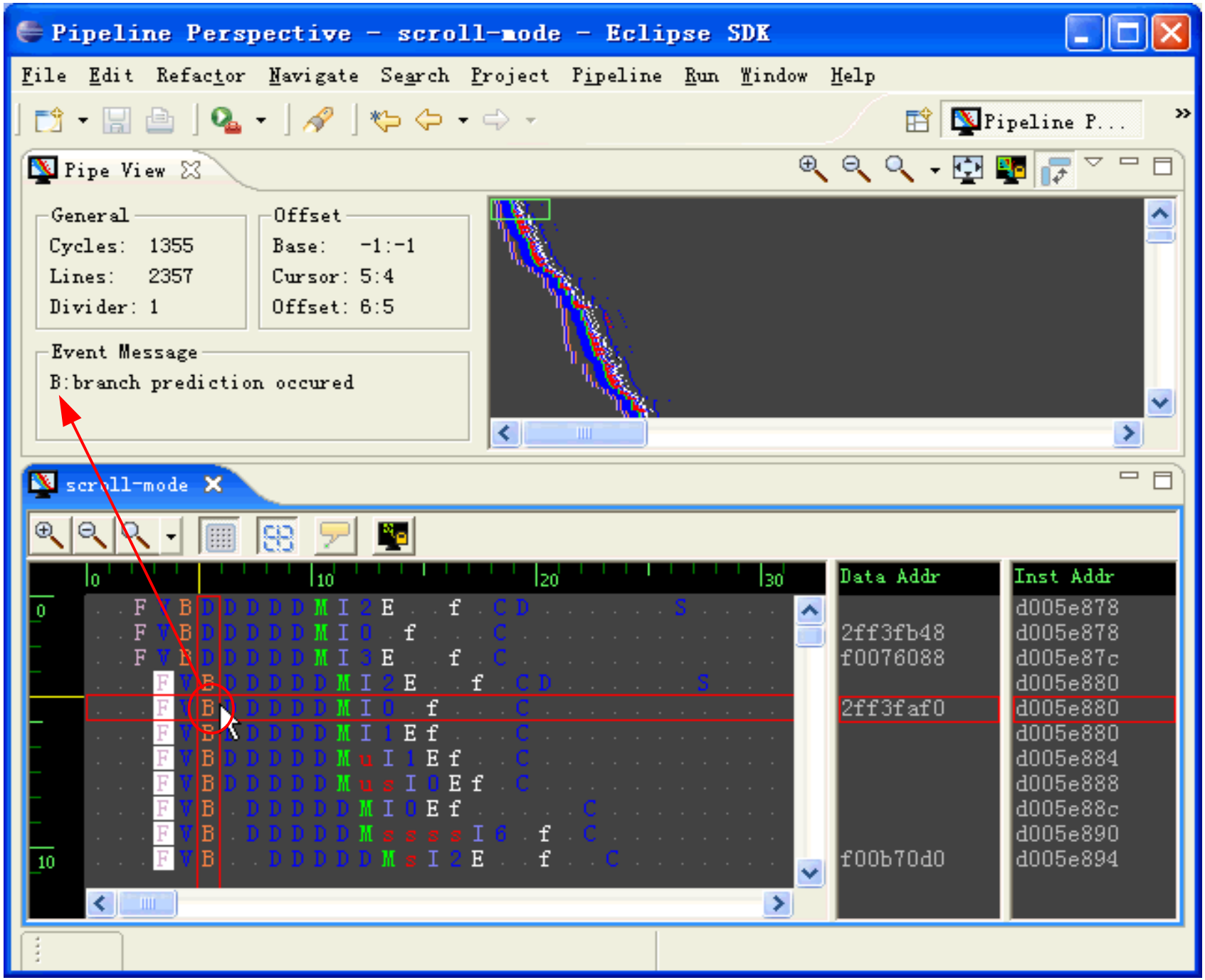
Code Analyzer – combined view with diassembly



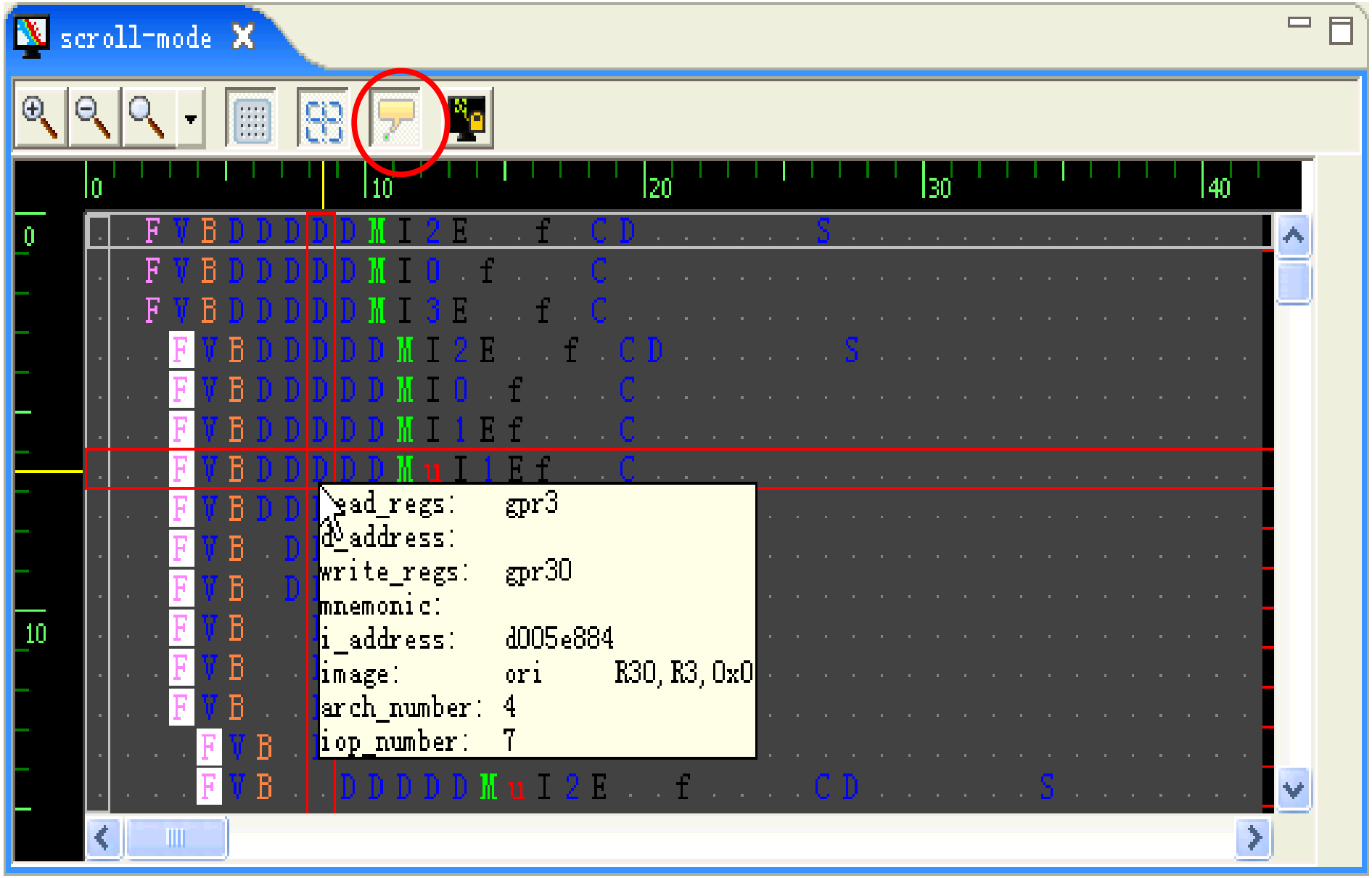
Pipeline Analyzer – scroll mode



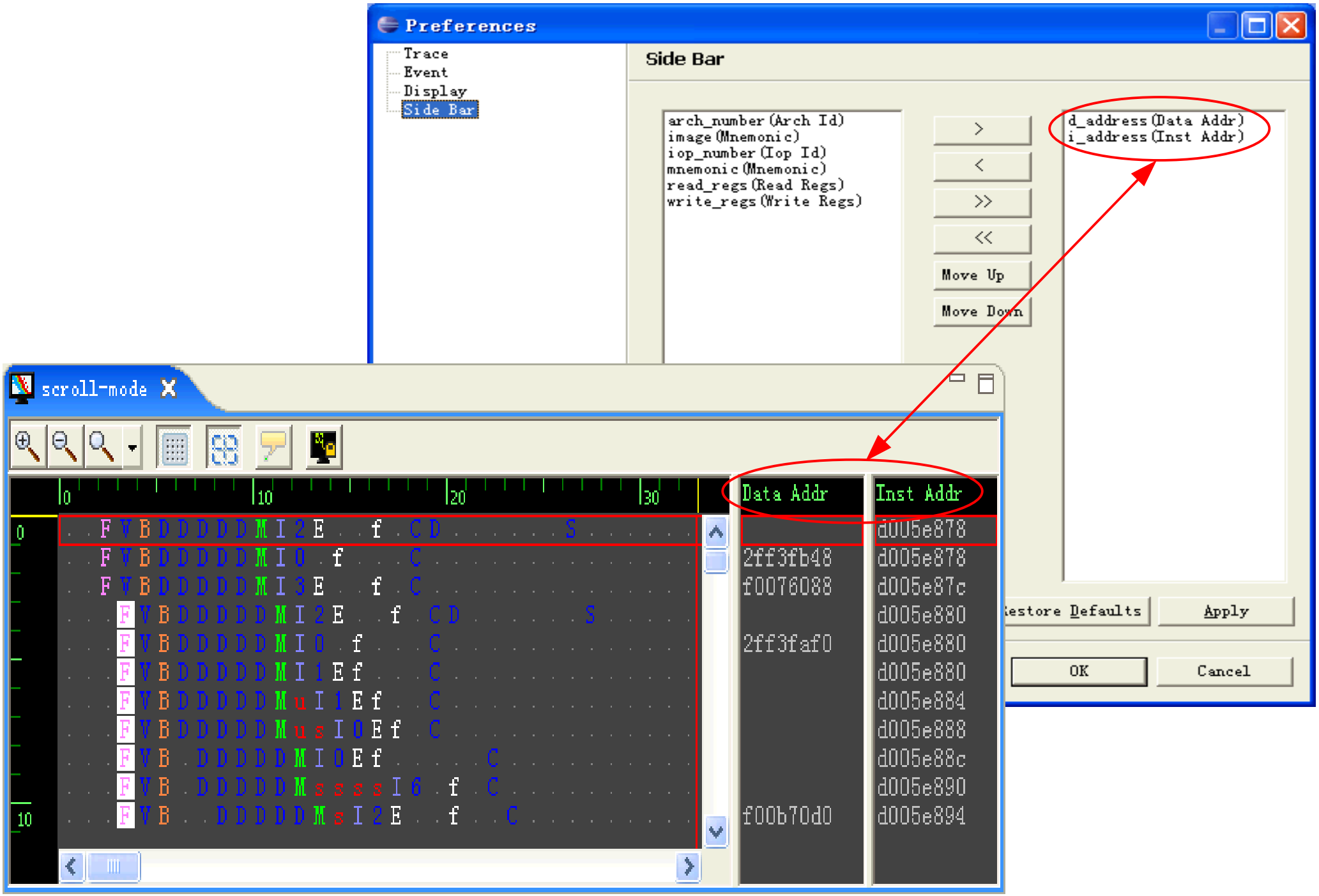
Pipeline Analyzer – scroll mode navigation



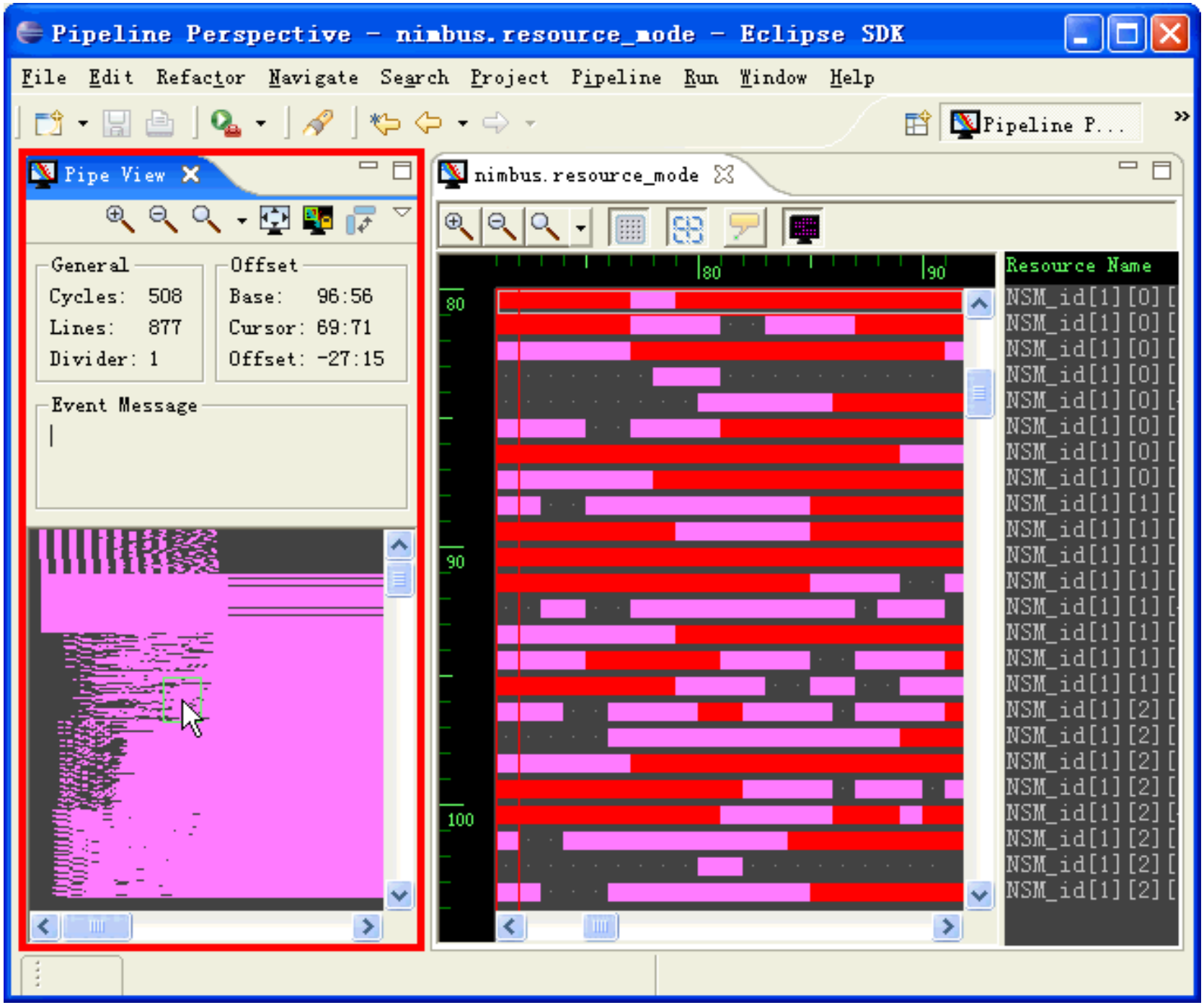
Pipeline Analyzer – scroll mode navigation



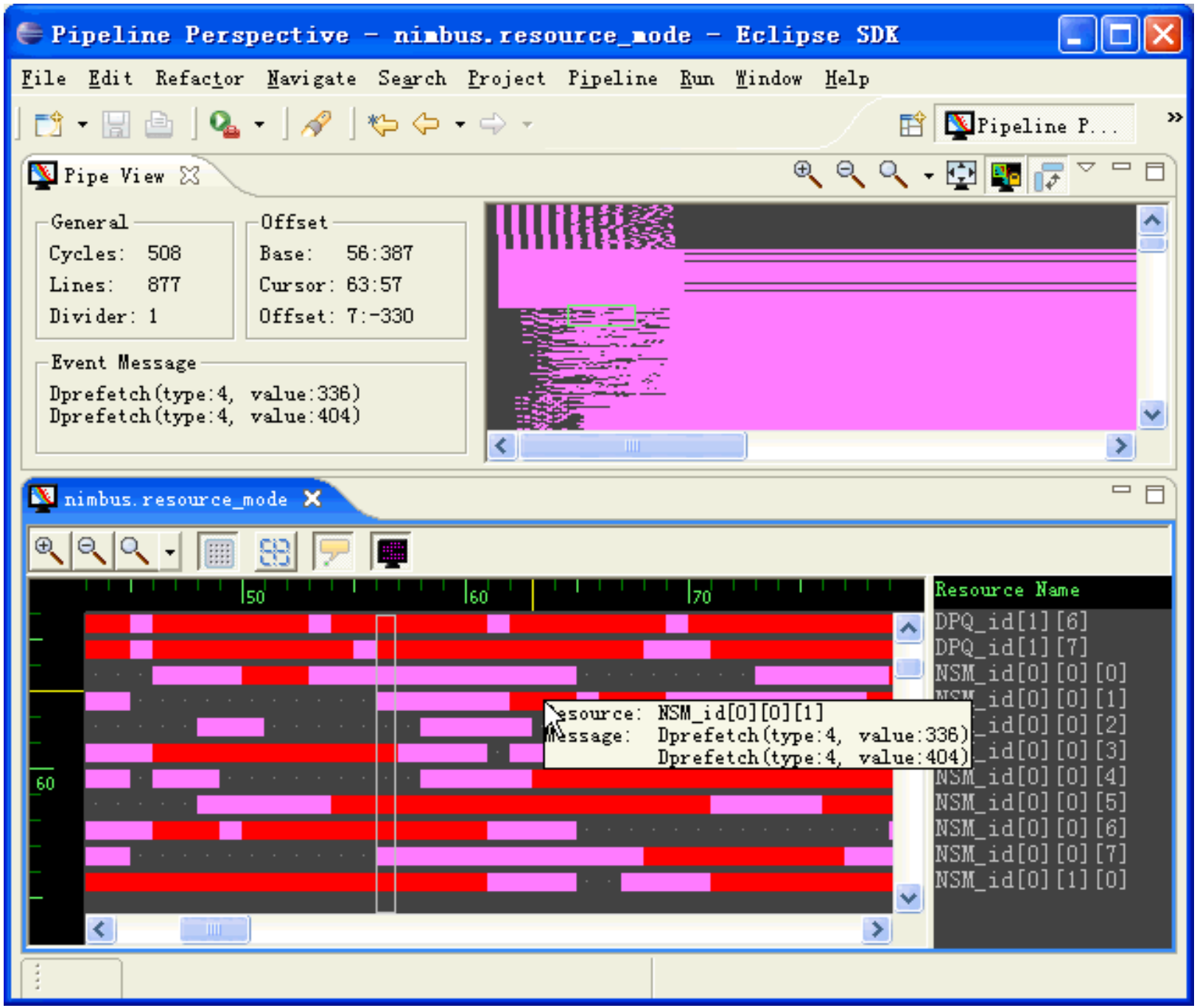
Pipeline Analyzer – scroll mode navigation



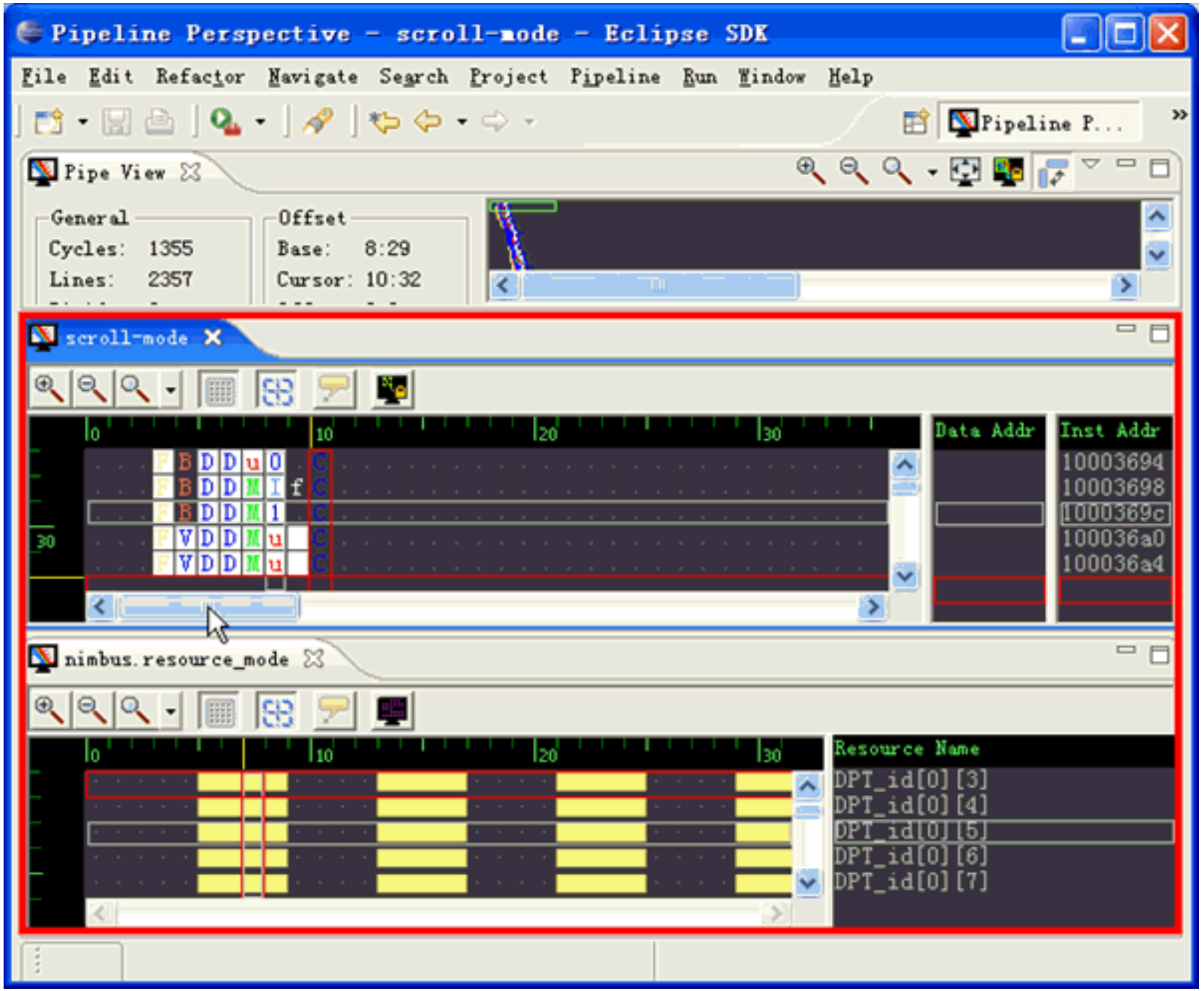
Pipeline Analyzer – resource editor



Pipeline Analyzer – resource editor



Pipeline Analyzer – tie cycle control



Thank you

charts to be available at:

http://www.ibm.com/developerworks/blogs/page/Systemptechuniv