Reveal

Heidi Poxon
Cray Inc.

Porting to a Hybrid or Many-core System

A Porting and Optimization Strategy for Hybrid and Many-core Systems

- Maximize on-node communication between MPI ranks
- Relieve on-node shared resource contention by pairing threads or processes that perform different work (for example computation with off-node communication) on the same node
- Add parallelism to MPI ranks to take advantage of cores within a node while minimizing network injection contention
- Accelerate work intensive parallel loops

Approach to Adding Parallelism

1. **Identify possible accelerator kernels**
   - Determine where to add additional levels of parallelism
     - Assumes MPI application is functioning correctly on X86
     - Find top serial work-intensive loops (perftools + CCE loop work estimates)
2. **Perform parallel analysis, scoping and vectorization**
   - Split loop work among threads
     - Do parallel analysis and restructuring on targeted high level loops
     - Use CCE loopmark feedback, Reveal loopmark and source browsing
3. **Move to OpenMP and then to OpenACC**
   - Add parallel directives and acceleration extensions
     - Insert OpenMP directives (Reveal scoping assistance)
     - Run on X86 to verify application and check for performance improvements
     - Convert desired OpenMP directives to OpenACC
4. **Analyze performance from optimizations**

Step 1 - Identify possible accelerator kernels

Loop Work Estimates

- **Helps identify high-level serial loops to parallelize**
  - Based on runtime analysis, approximates how much work exists within a loop
  - Provides min, max and average trip counts that can be used to approximate work and help carve up loop on GPU

Collecting Loop Statistics

- Load PrgEnv-cray module
- Load perftools module

- Compile **AND** link with –h profile_generate

- Instrument binary for tracing
  - pat_build –w my_program

- Run application

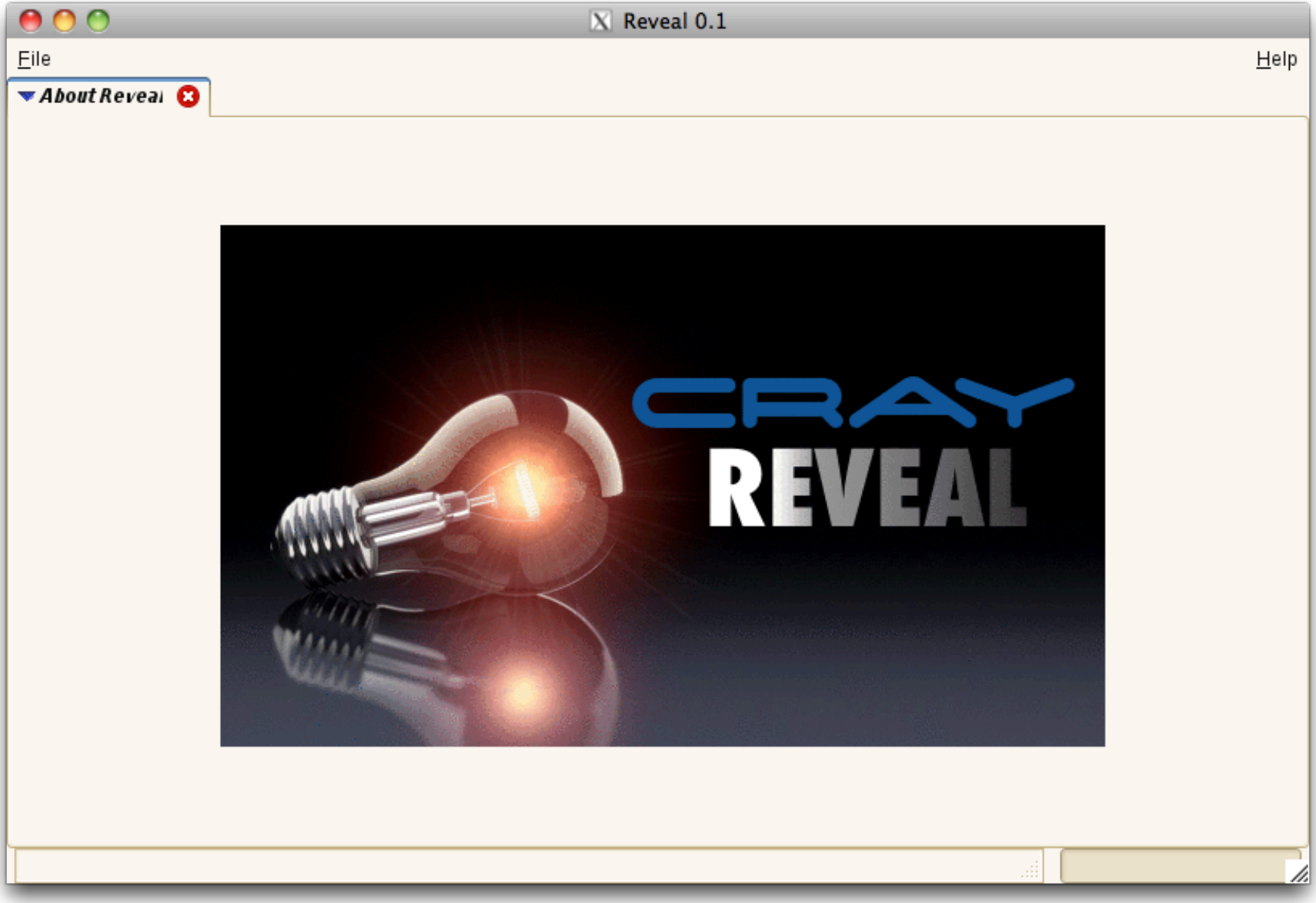
- Create report with loop statistics
  - pat_report my_program.xf > loops_report

Example Report – Inclusive Loop Time

Table 2: Loop Stats by Function (from -hprofile_generate)

Loop Incl Time Total	Loop Hit	Loop Trips Avg	Loop Trips Min	Loop Trips Max	Function=/.LOOP[.] PE=HIDE
8.995914	100	25	0	25	sweepy_.LOOP.1.li.33
8.995604	2500	25	0	25	sweepy_.LOOP.2.li.34
8.894750	50	25	0	25	sweepz_.LOOP.05.li.49
8.894637	1250	25	0	25	sweepz_.LOOP.06.li.50
4.420629	50	25	0	25	sweepx2_.LOOP.1.li.29
4.420536	1250	25	0	25	sweepx2_.LOOP.2.li.30
4.387534	50	25	0	25	sweepx1_.LOOP.1.li.29
4.387457	1250	25	0	25	sweepx1_.LOOP.2.li.30
2.523214	187500	107	0	107	riemann_.LOOP.2.li.63
1.541299	20062500	12	0	12	riemann_.LOOP.3.li.64
0.863656	1687500	104	0	108	parabola_.LOOP.6.li.67

Step 2 - Perform parallel analysis, scoping and vectorization
&
Step 3 - Move to OpenMP and then to OpenACC


Reveal 0.1
File
Help
About Reveal
CRAY
REVEAL

Reveal

New code analysis and restructuring assistant…

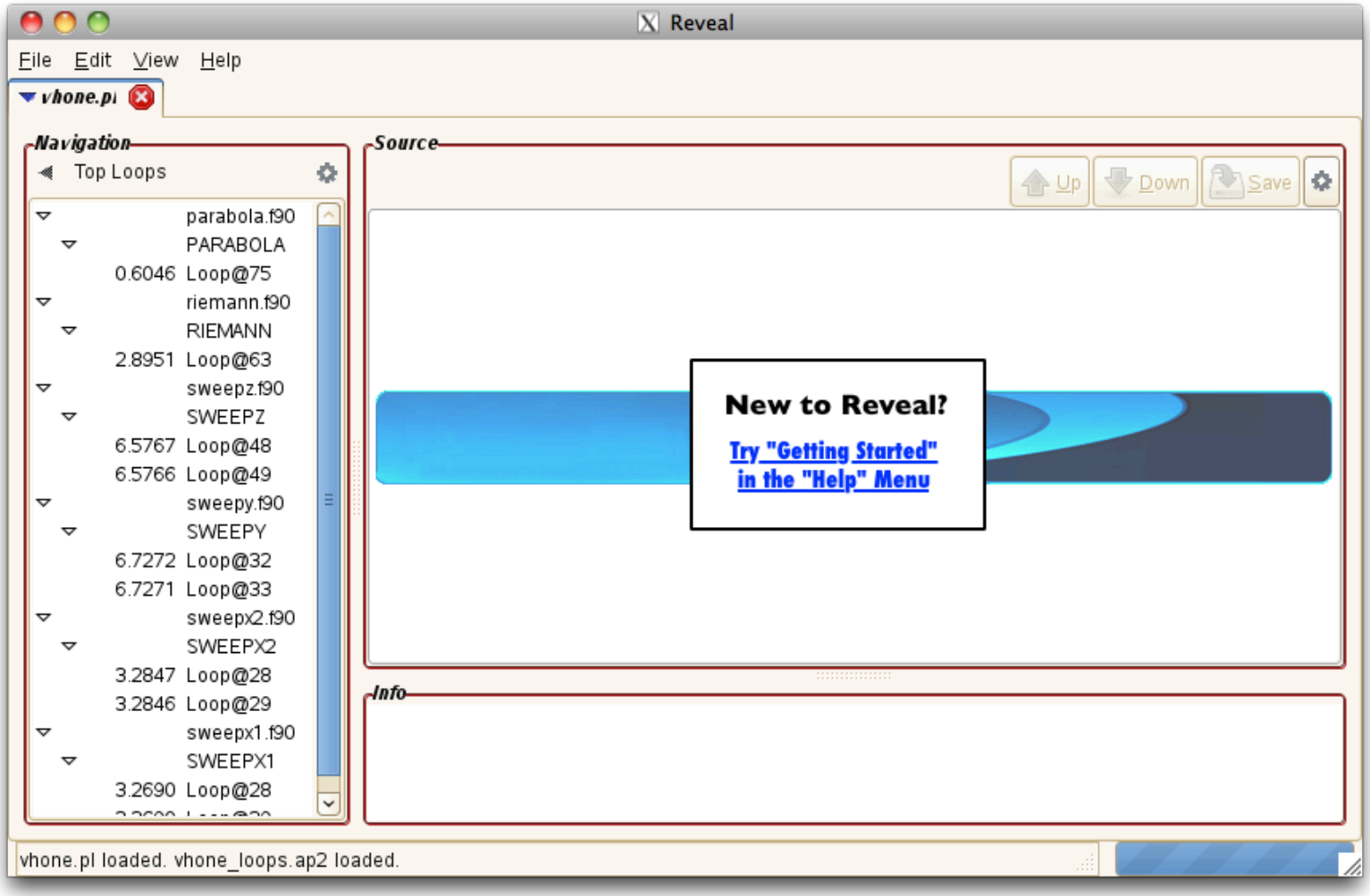
- **Uses both the performance toolset and CCE's program library functionality to provide static and runtime analysis information**

- **Key Features**
  - Annotated source code with compiler optimization information
    - Feedback on critical dependencies that prevent optimizations
  - Scoping analysis
    - Identify, shared, private and ambiguous arrays
      - Allow user to privatize ambiguous arrays
      - Allow user to override dependency analysis
  - Source code navigation based on performance data collected through CrayPat

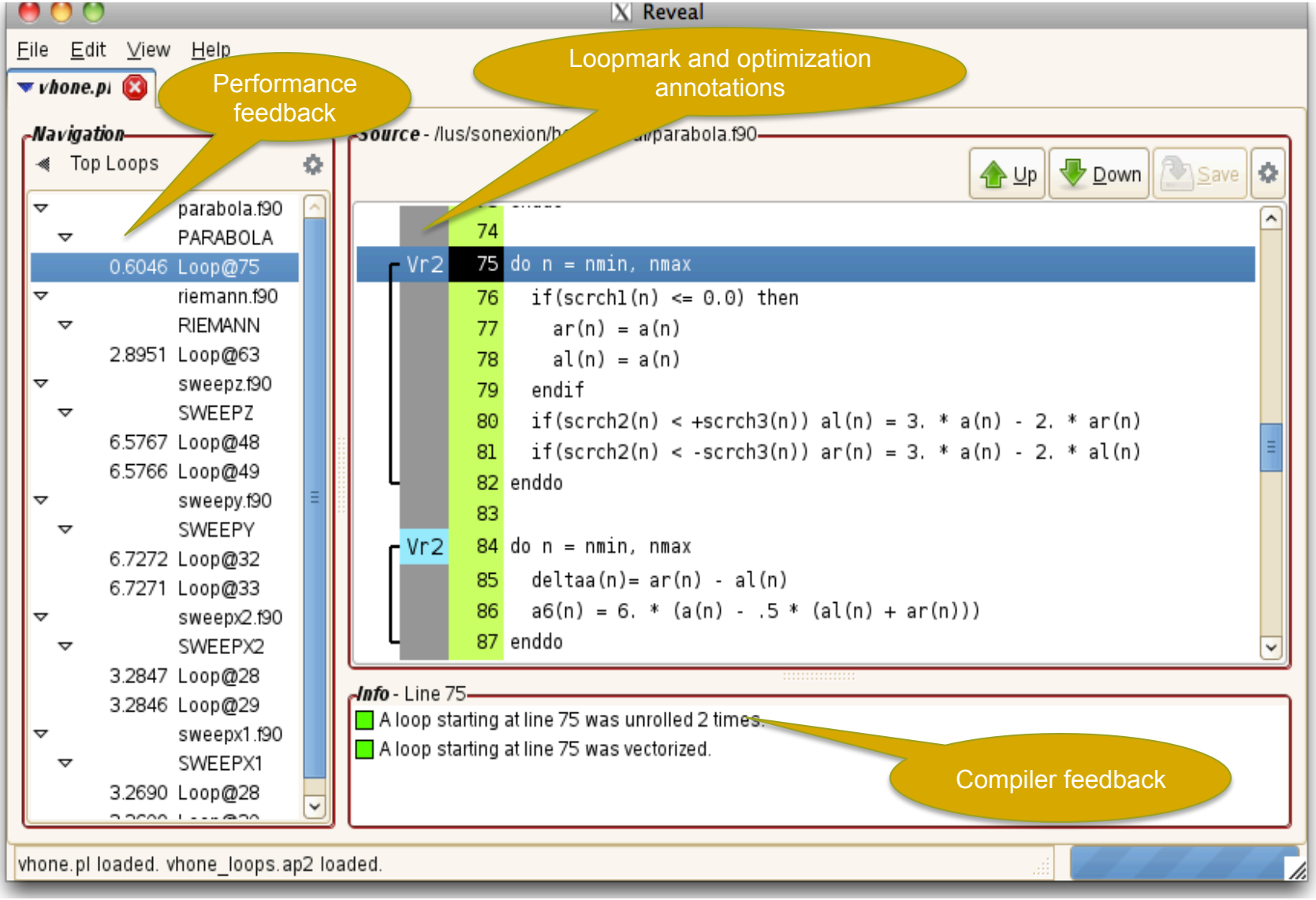
How to Use

- **Optionally create loop statistics using Cray performance tools to determine which loops have the most work**
  - >
- **Compile your application with Cray CCE to generate a program library**
  - > ftn –h pl=vhone.pl –c file1.f90
- **Run reveal**
  - Compiler information only:
    - > reveal vhone.pl
  - Compiler + loop work estimates
    - > reveal vhone.pl vhone_loops.ap2

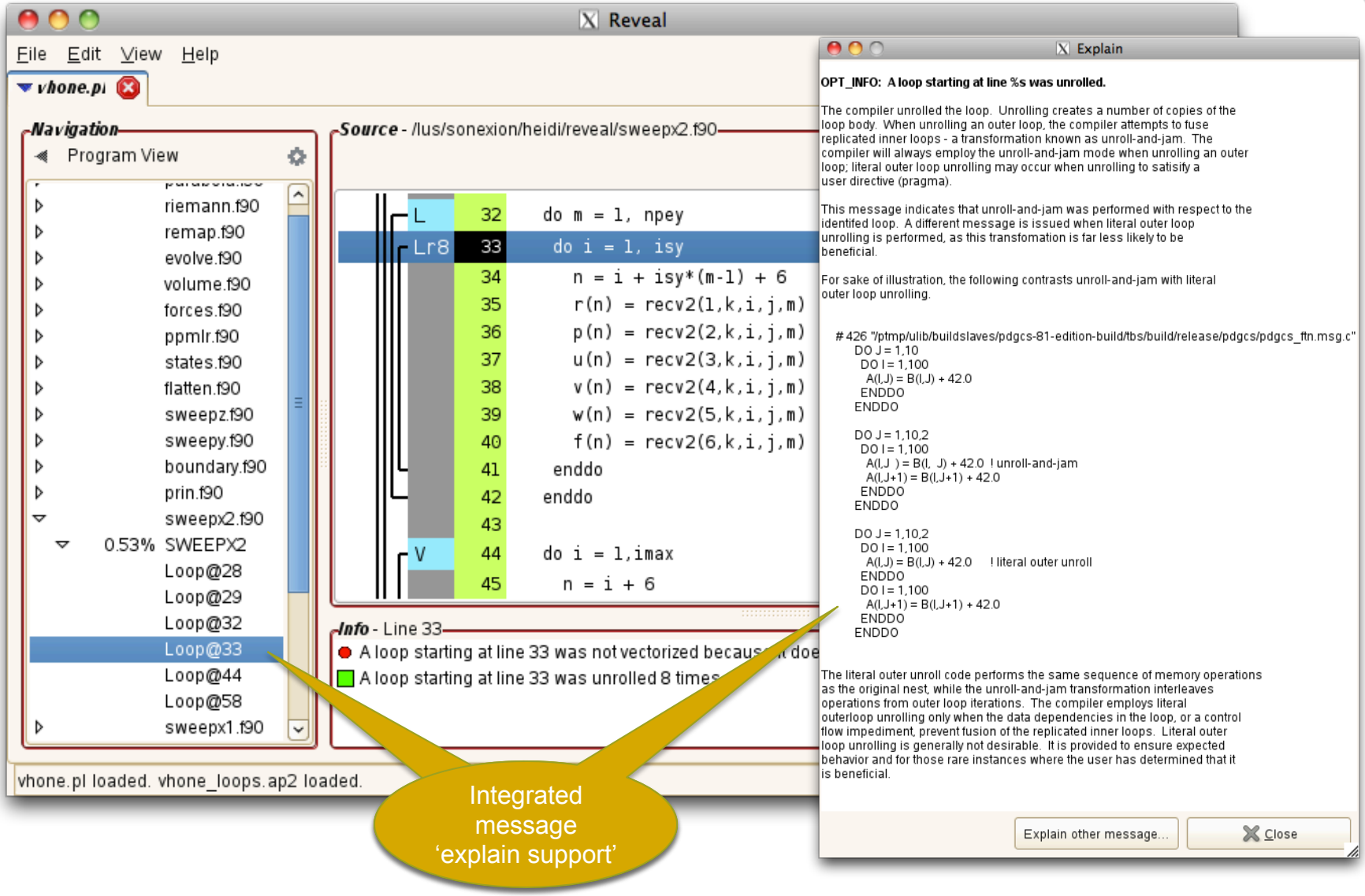
Reveal with Loop Work Estimates



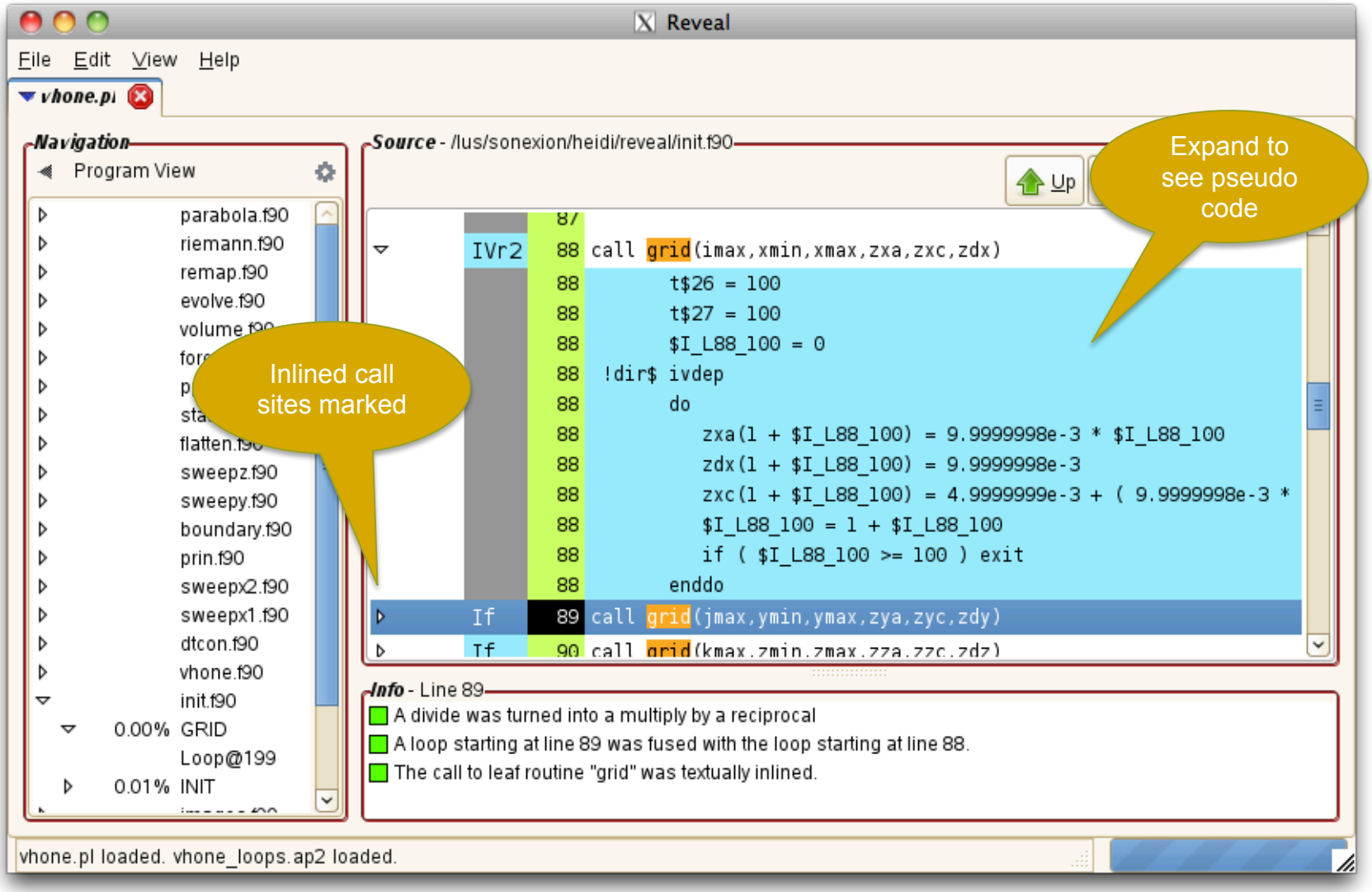
Visualize Loopmark with Performance Information



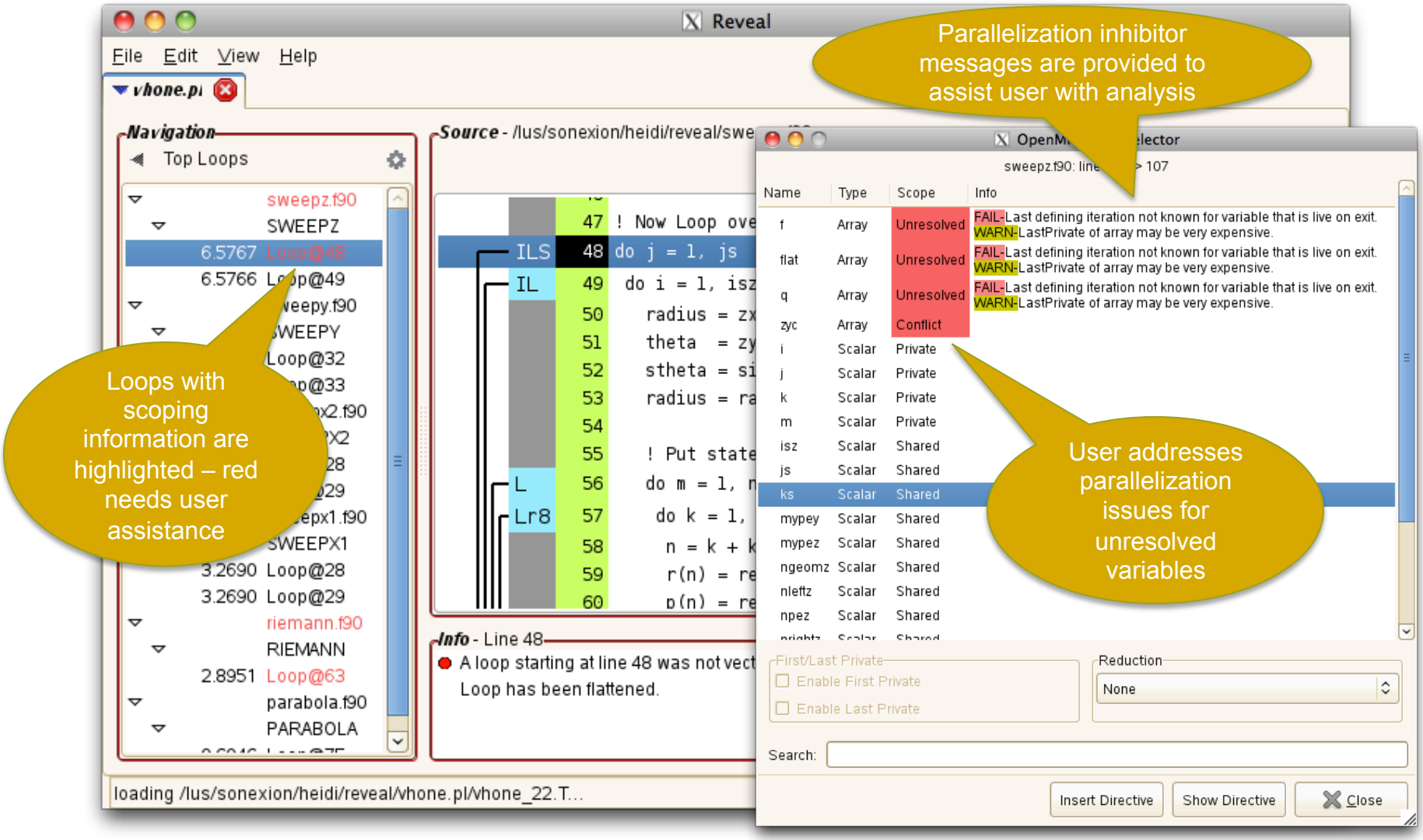
Visualize CCE's Loopmark with Performance Profile (2)



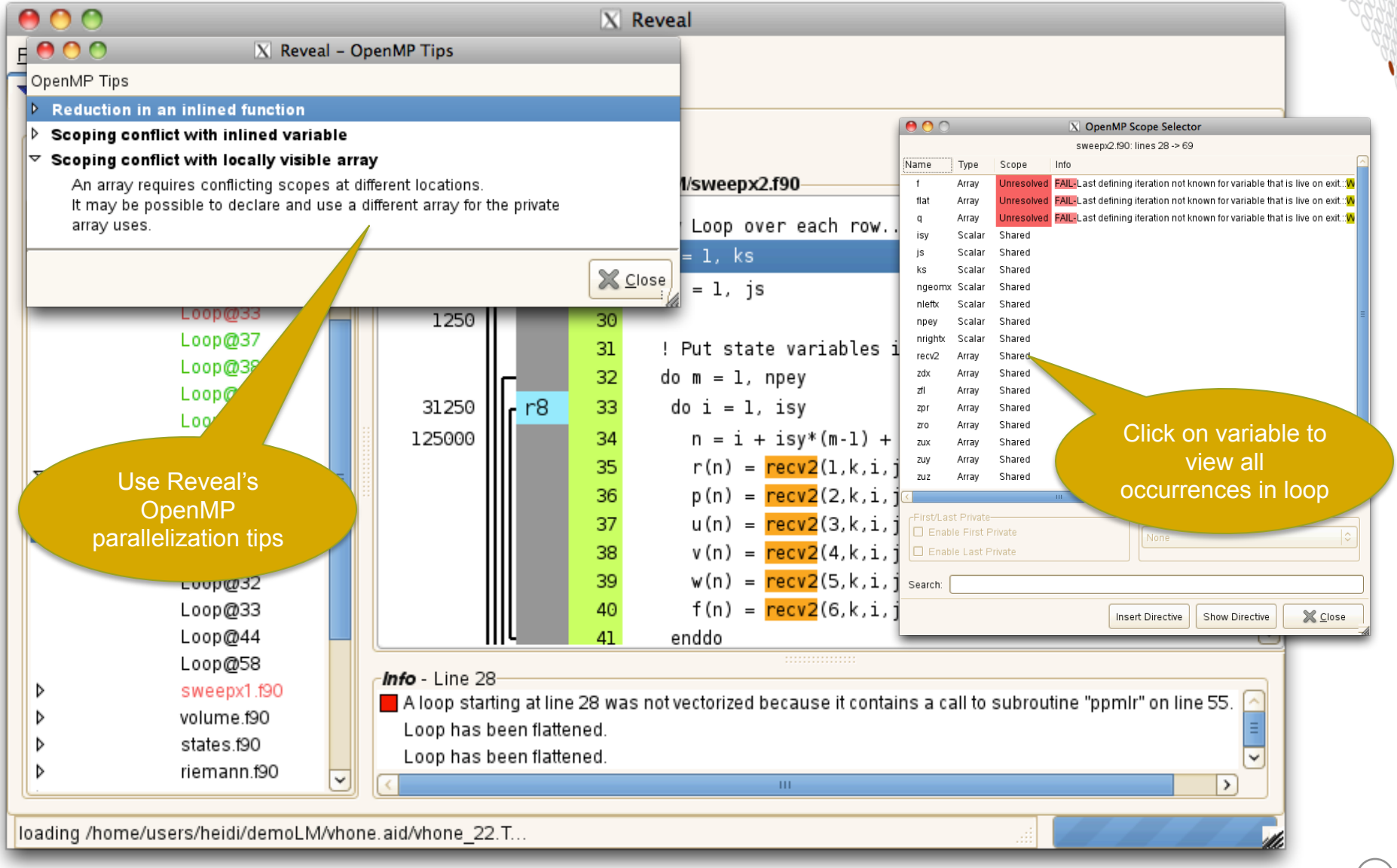
View Pseudo Code for Inlined Functions



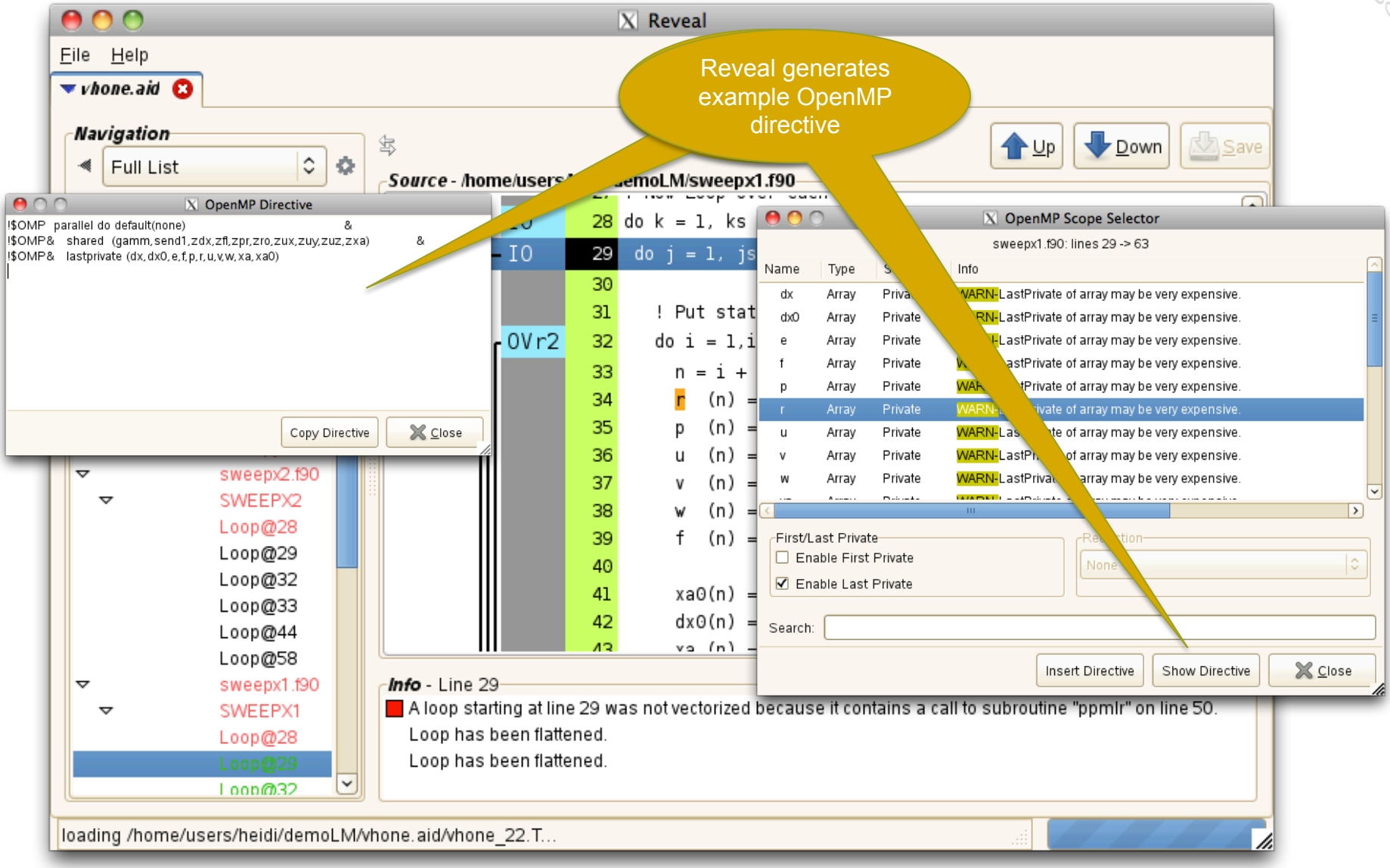
Scoping Assistance – Review Scoping Results



Scoping Assistance – User Resolves Issues



Scoping Assistance – Generate Directive



Questions ?