

Analysis report examination with CUBE

Brian Wylie
Jülich Supercomputing Centre



CUBE

Parallel program analysis report exploration tools

- Libraries for XML report reading & writing
- Algebra utilities for report processing
- GUI for interactive analysis exploration
 - requires Qt4.6 or later

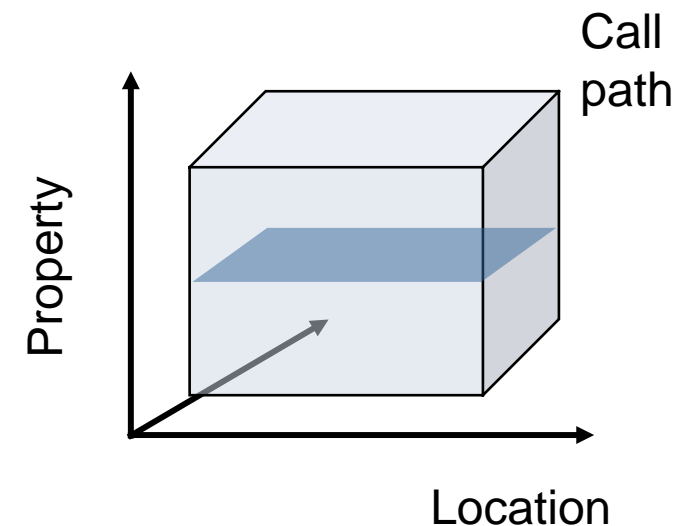
Originally developed as part of Scalasca toolset

Now available as a separate component

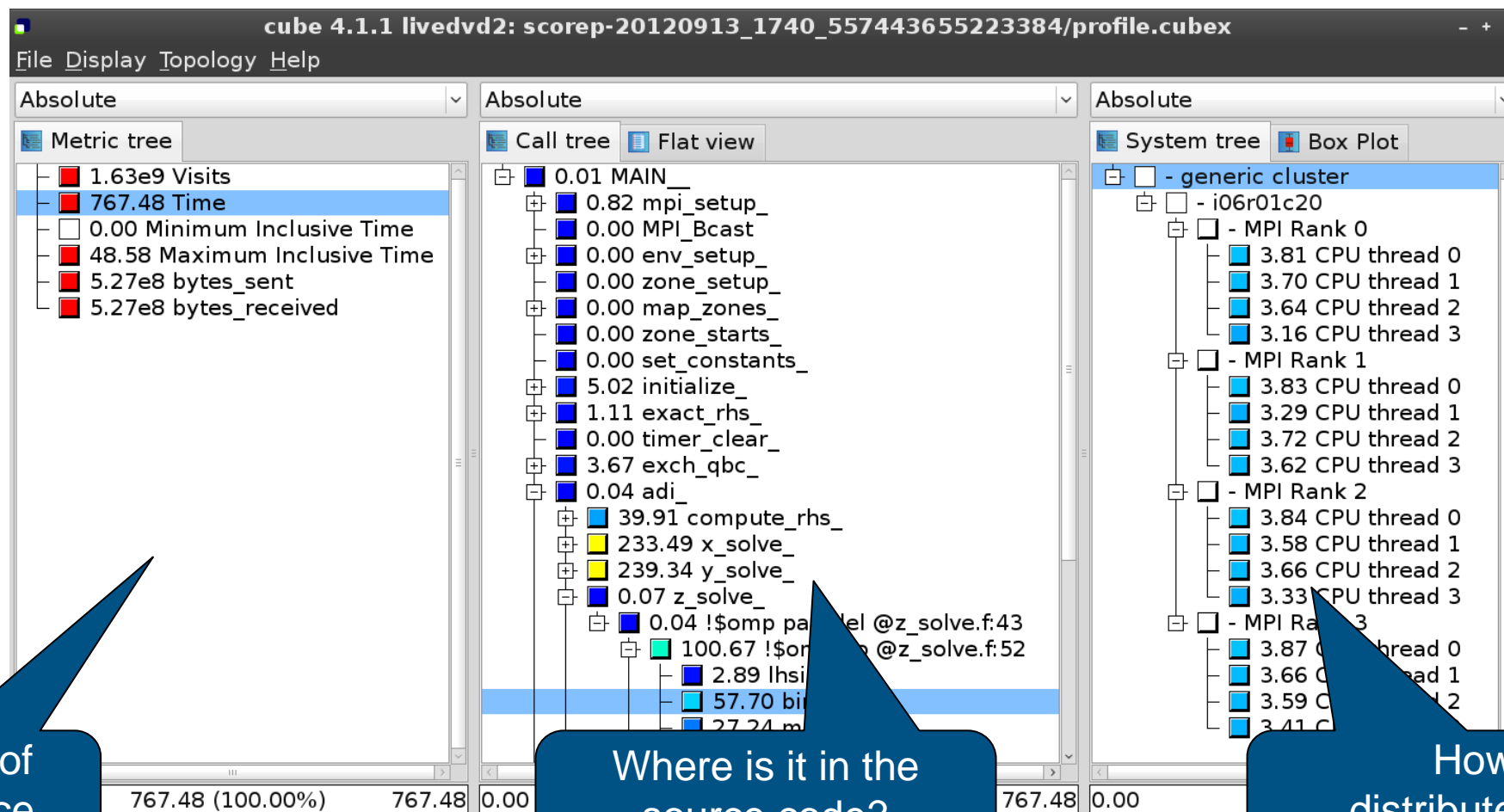
- Can be installed independently of Score-P, e.g., on laptop or desktop
- Latest release: CUBE 4.3.3 (December 2015)

Analysis presentation and exploration

- Representation of values (severity matrix) on three hierarchical axes
 - Performance property (metric)
 - Call path (program location)
 - System location (process/thread)
- Three coupled tree browsers
- CUBE displays severities
 - As value: for precise comparison
 - As colour: for easy identification of hotspots
 - Inclusive value when closed & exclusive value when expanded
 - Customizable via display modes



Analysis presentation

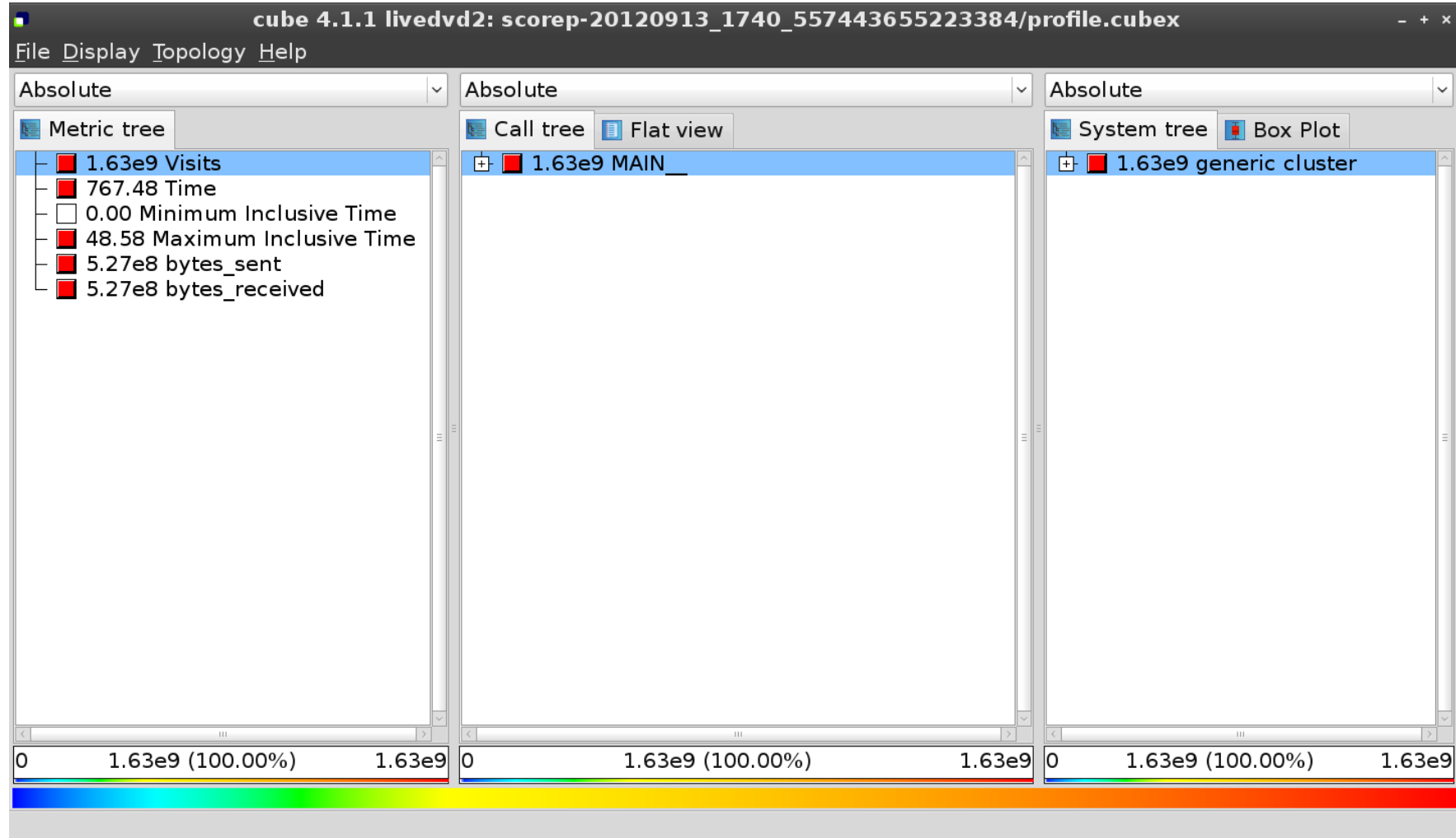


What kind of performance metric?

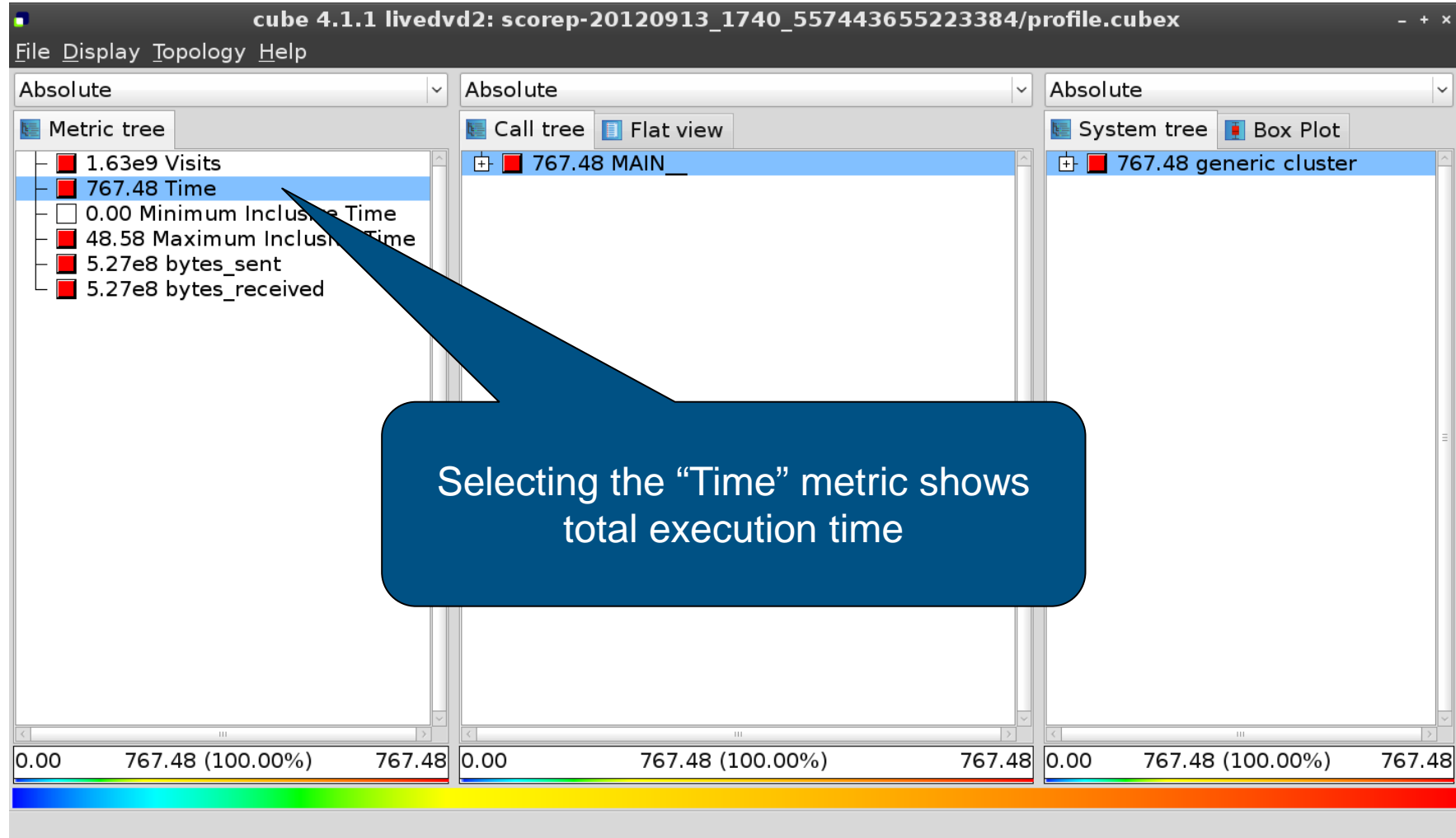
Where is it in the source code? In what context?

How is it distributed across the processes/threads?

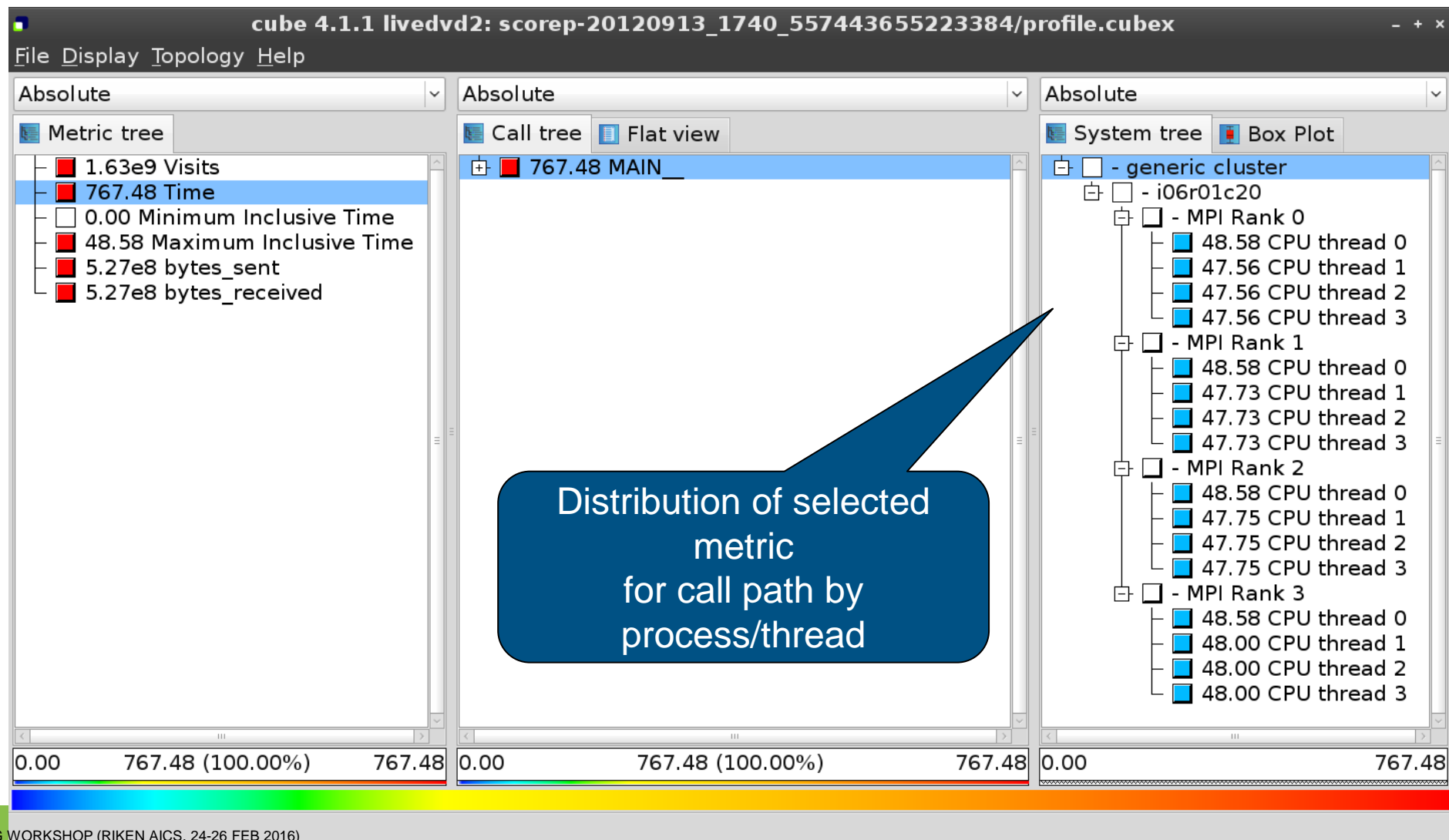
Analysis report exploration (opening view)



Metric selection



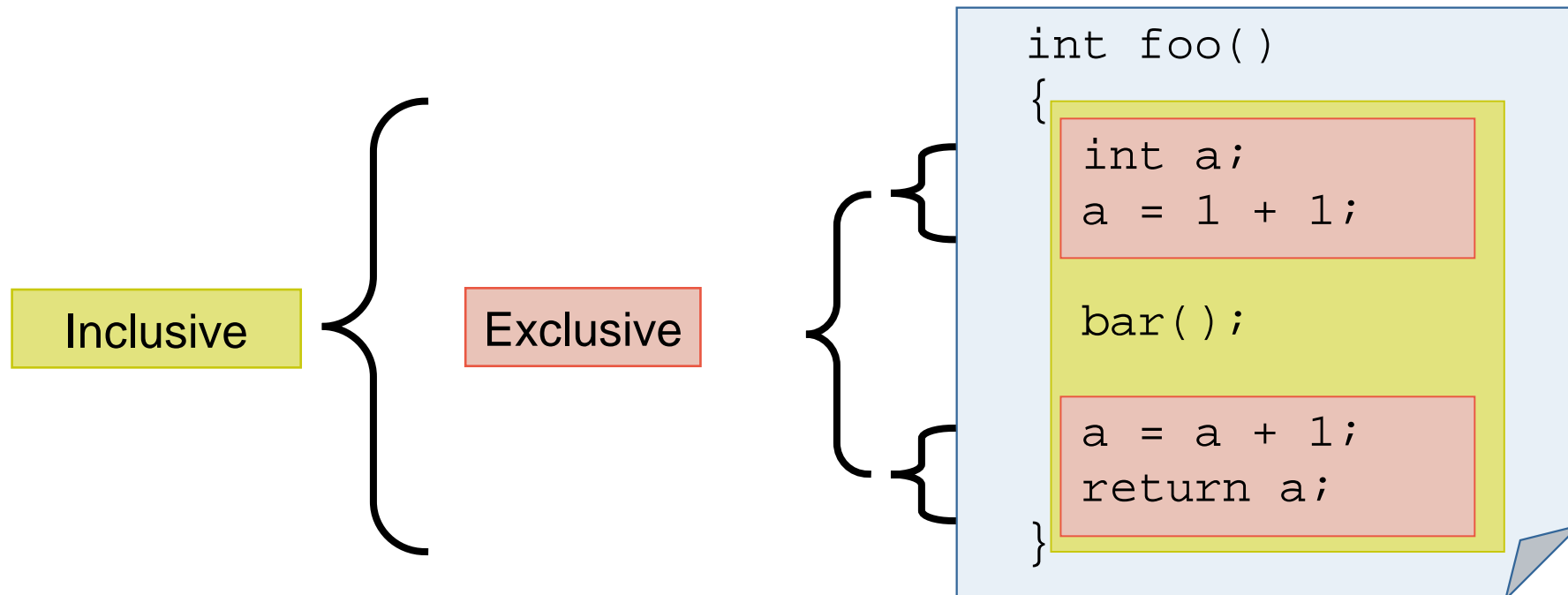
Expanding the system tree



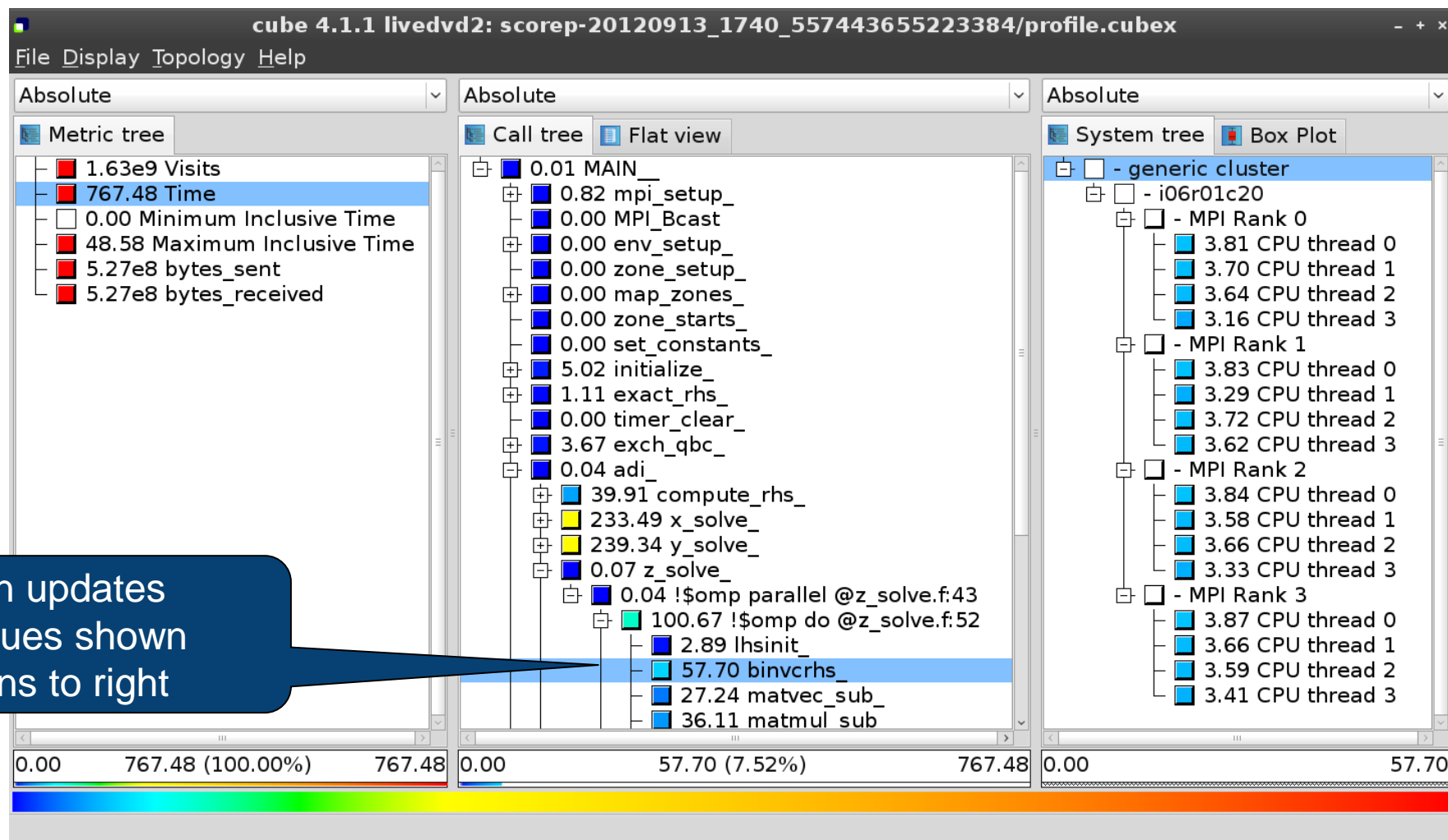


Inclusive vs. Exclusive values

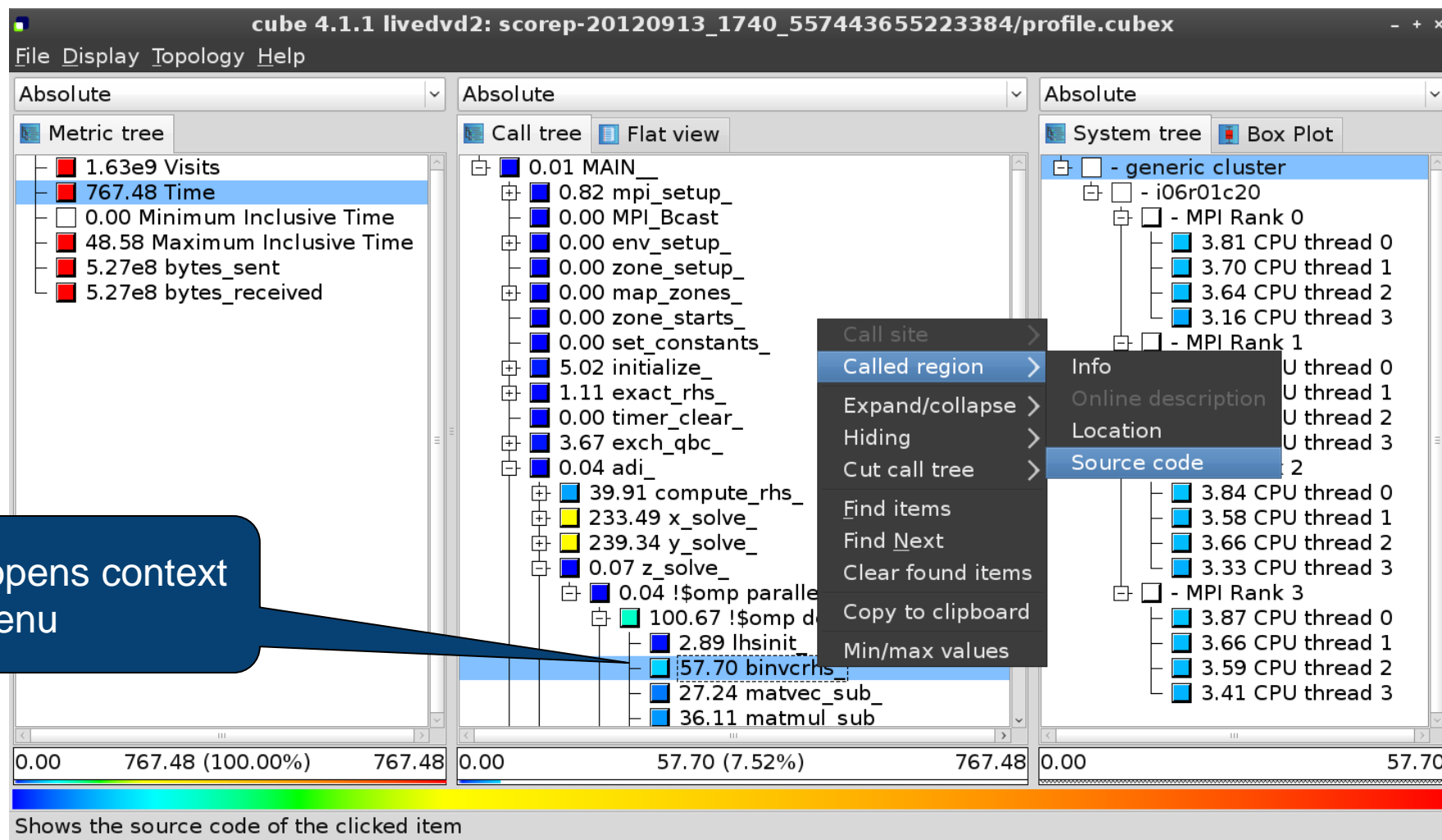
- Inclusive
 - Information of all sub-elements aggregated into single value
- Exclusive
 - Information cannot be subdivided further



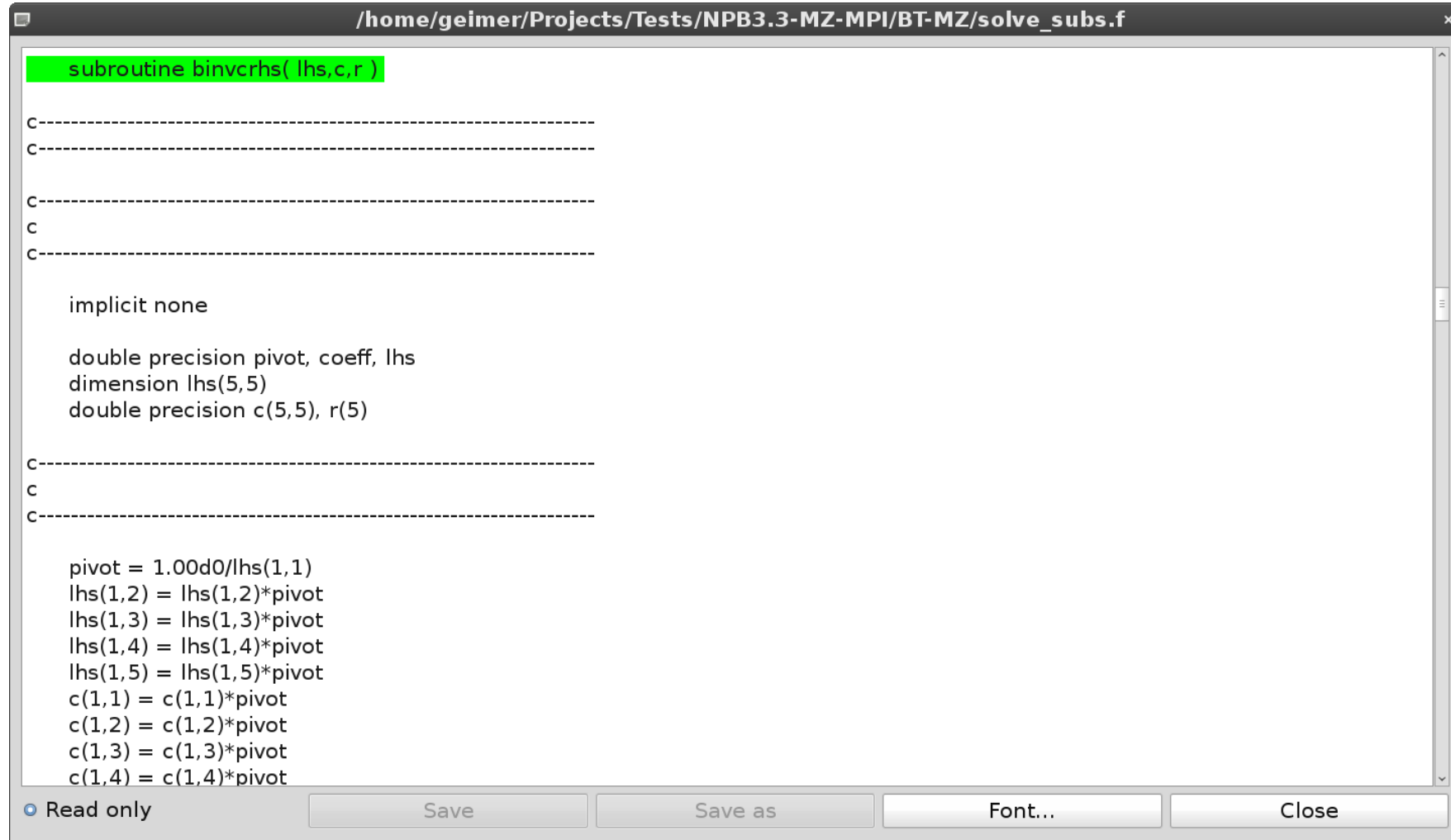
Selecting a call path



Source-code view via context menu



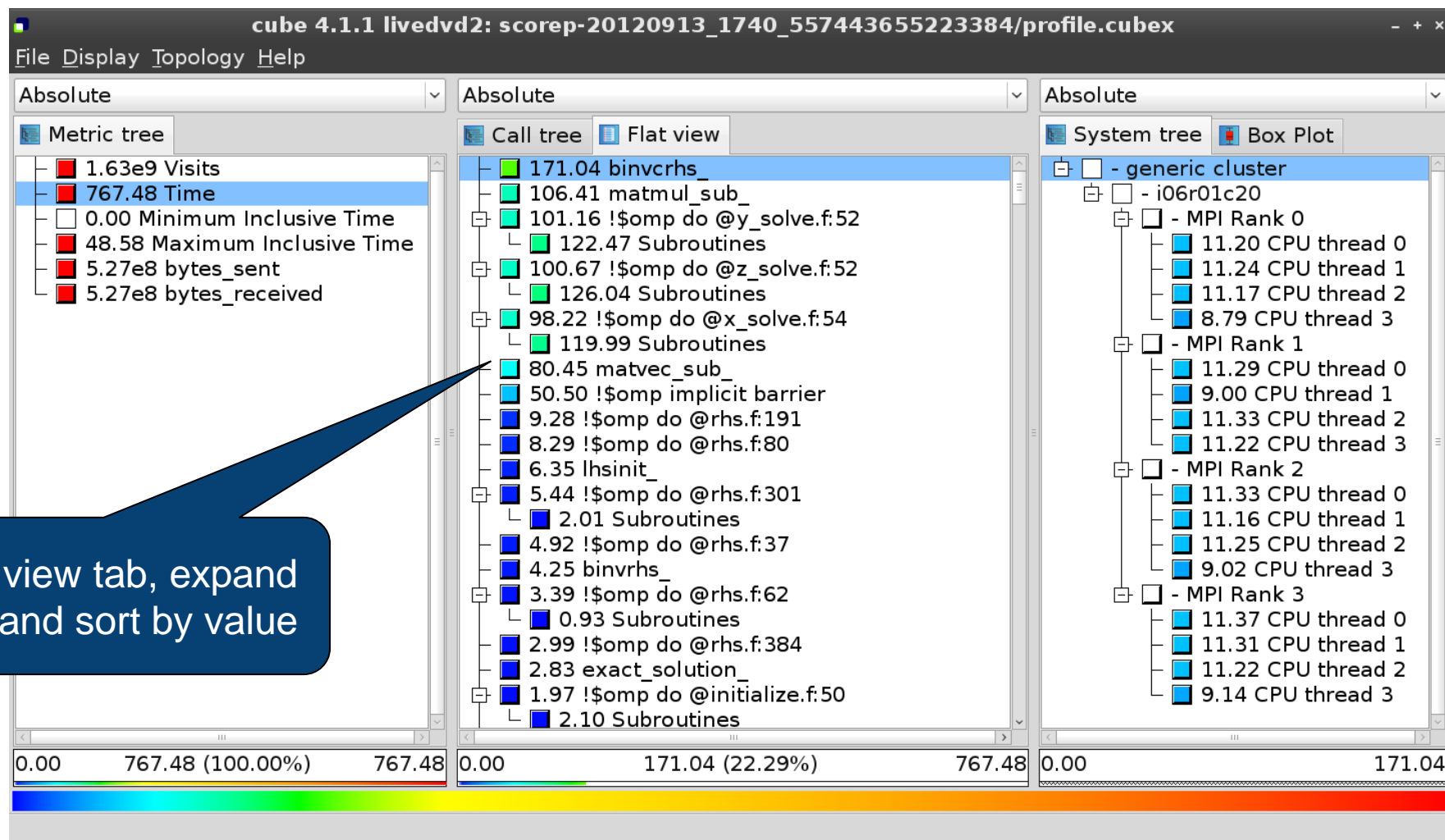
Source-code view



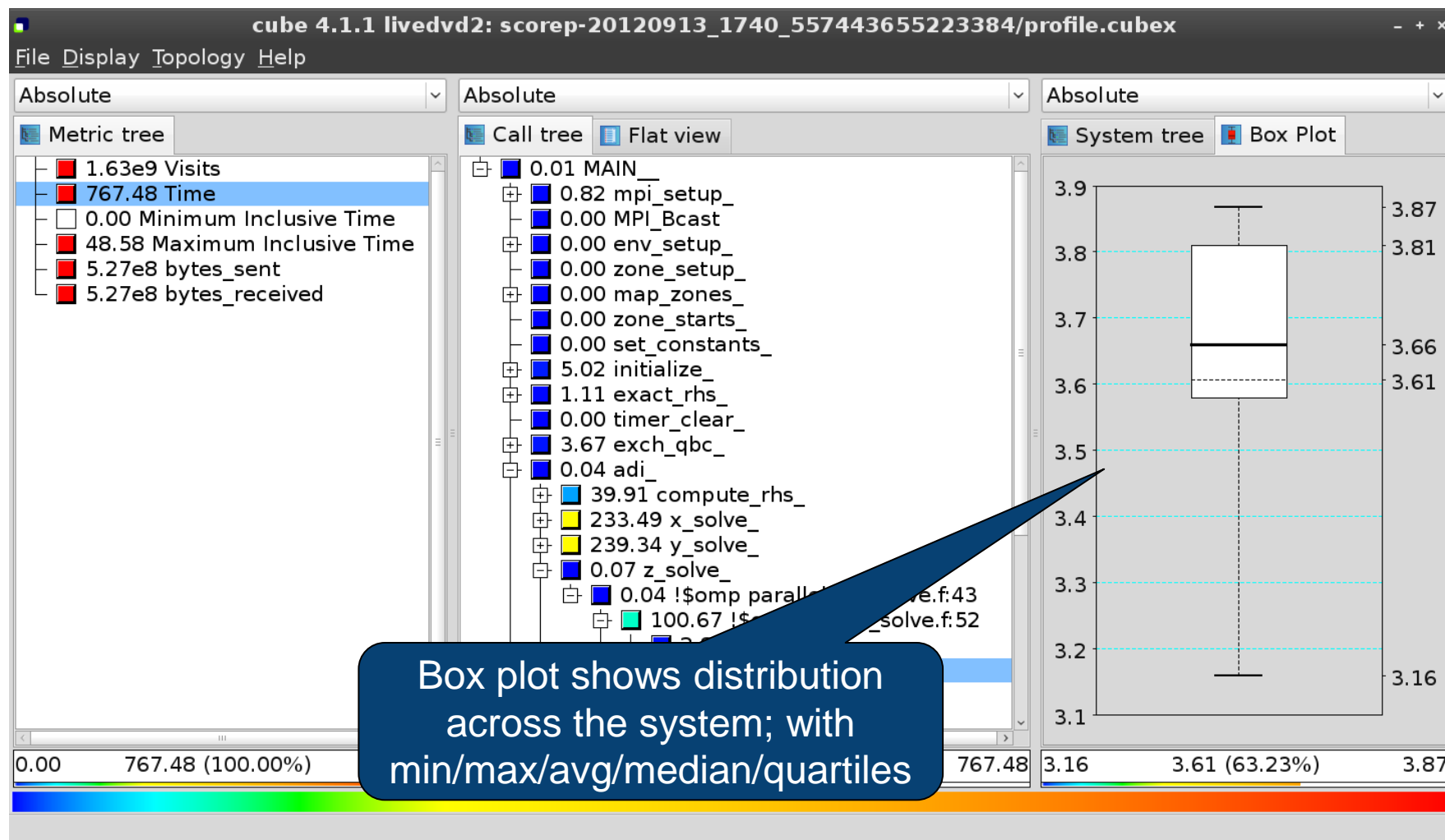
```
subroutine binvcrhs( lhs,c,r )  
  
C-----  
C-----  
  
C-----  
C  
C-----  
  
implicit none  
  
double precision pivot, coeff, lhs  
dimension lhs(5,5)  
double precision c(5,5), r(5)  
  
C-----  
C  
C-----  
  
pivot = 1.00d0/lhs(1,1)  
lhs(1,2) = lhs(1,2)*pivot  
lhs(1,3) = lhs(1,3)*pivot  
lhs(1,4) = lhs(1,4)*pivot  
lhs(1,5) = lhs(1,5)*pivot  
c(1,1) = c(1,1)*pivot  
c(1,2) = c(1,2)*pivot  
c(1,3) = c(1,3)*pivot  
c(1,4) = c(1,4)*pivot
```

☒ Read only Save Save as Font... Close

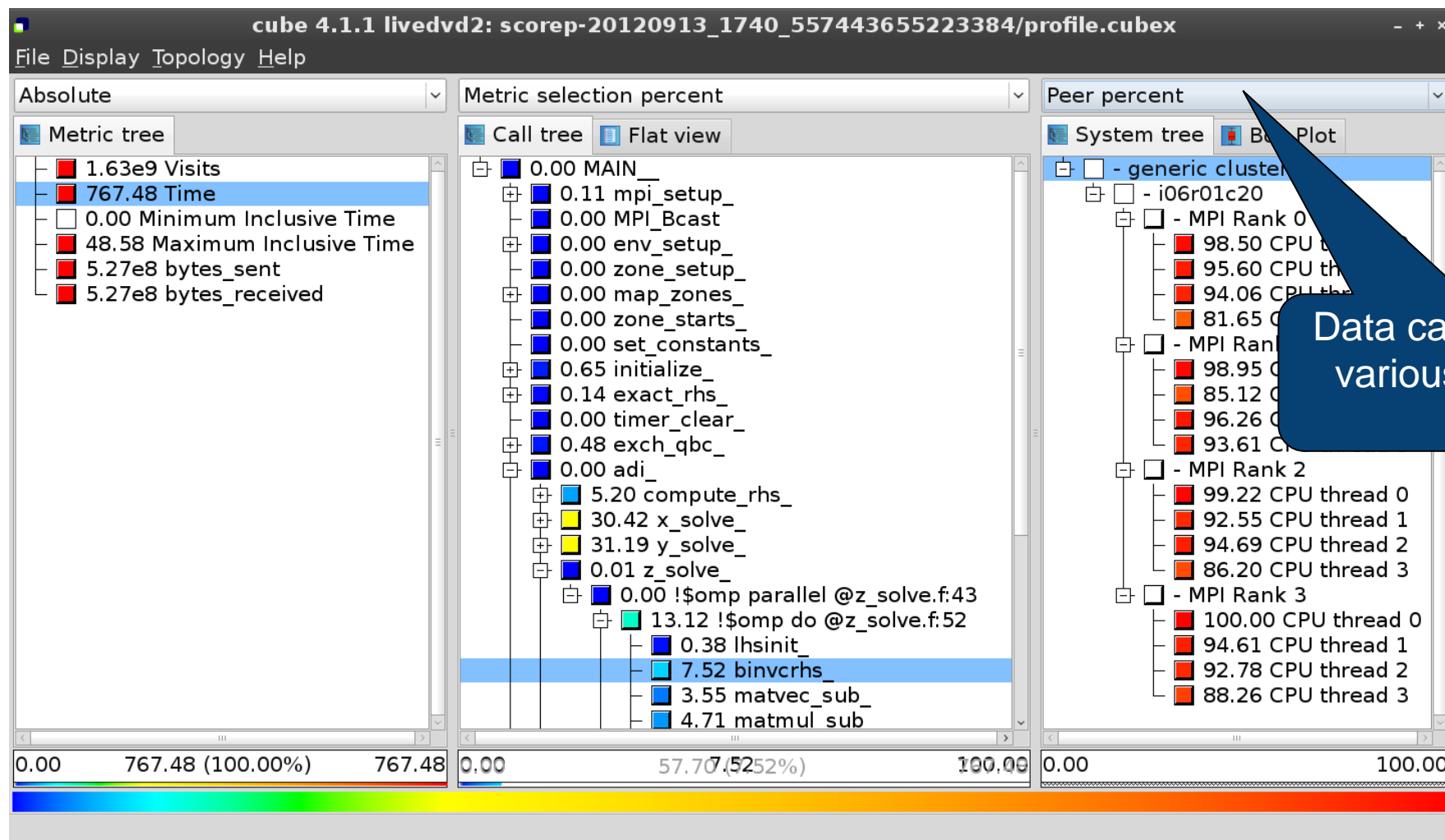
Flat profile view



Box plot view



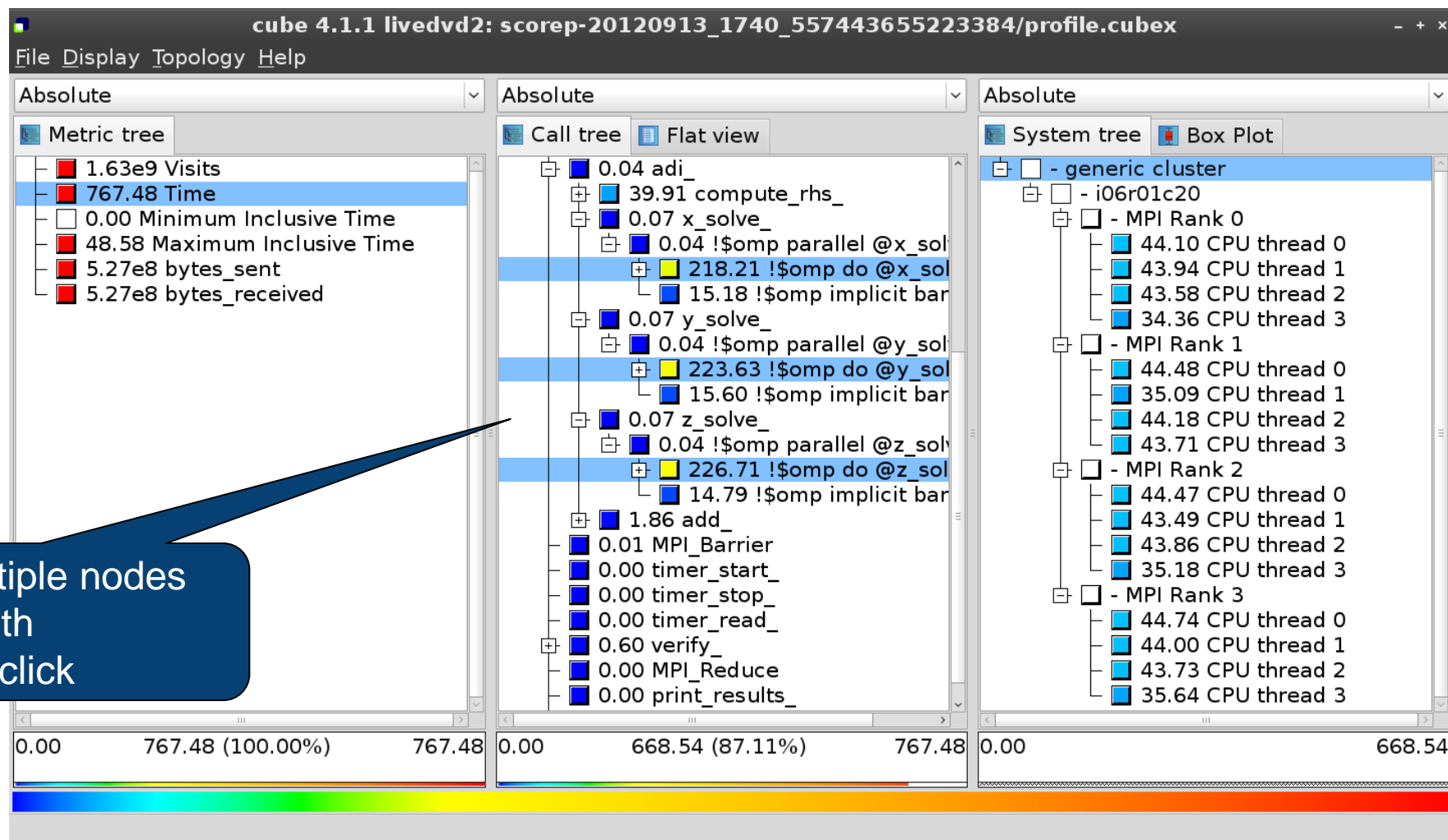
Alternative display modes



Important display modes

- Absolute
 - Absolute value shown in seconds/bytes/counts
- Selection percent
 - Value shown as percentage w.r.t. the selected node
“on the left” (metric/call path)
- Peer percent (system tree only)
 - Value shown as percentage relative to the maximum peer value

Multiple selection



Derived metrics in Cube

- Value of the derived metric is not stored, but **calculated** on-the-fly
- One defines an CubePL expression, e.g.:

`metric::time(i)/metric::visits(e)`

- Types of derived metrics:
 - **Prederived**: evaluation of the CubePL expression is done before the aggregation
 - **Postderived**: evaluation of the CubePL expression is performed after the aggregation

- Examples:

- “Average execution time” Postderived metric with an expression:

`metric::time(i)/metric::visits(e)`

- “Number of FLOP per second” Postderived metric with an expression:

`metric::FLOP()/metric::time()`

Derived metrics in Cube GUI

The screenshot displays the Cube GUI interface with a dialog box for creating a new derived metric. The dialog box is titled "Create new metric as a child of metric" and contains the following fields and options:

- Select metric from collection:** A dropdown menu showing "Average execution time (kenobi)".
- Derived metric type:** A dropdown menu showing "Postderived metric".
- Display name:** A text input field containing "Average visit time".
- Unique name:** A text input field containing "avg_visit_time".
- Data type:** A dropdown menu showing "DOUBLE".
- Unit of measurement:** A text input field containing "sec".
- URL:** An empty text input field.
- Description:** A text area containing "Calculates average time of region execution per visit. Autor is Michael Knobloch."
- Calculation:** A tab with a checked icon, showing a CubePL expression: `metric::time(i)/metric::visits(e)`.
- Calculation Init:** A tab with an unchecked icon.
- Aggregation "+":** A tab with an unchecked icon.
- Aggregation "-":** A tab with an unchecked icon.

The background shows the main Cube GUI interface with three panels:

- Metric tree:** A list of metrics including "1.09e8 Visits (occ)", "1.01e6 Time (sec)", "0.00 Minimum Inclusive Time (sec)", "246.14 Maximum Inclusive Time (sec)", "7.18e12 bytes_sent", and "7.18e12 bytes_received".
- Call tree:** A list of function calls including "0.35 main(int, char *[])", "2512.10 ugshellInit", "1.01e6 ug::script::LoadUGScript(const char *, bool)", "2.11 ug::script::ParseBuffer(const char *, const char *)", "0.04 ugshellFinalize", and "94.31 MPI_Finalize".
- System tree:** A list of system components including "machine Blue Gene/Q", "rack 11", "midplane 1", "nodeboard 8", "nodecard 4", "0.65 MPI Rank 0", and "0.64 MPI Rank 1".

Collection of derived metrics

Parameters of the derived metric

CubePL expression

20

Context-sensitive help

The screenshot displays the 'cube 4.1.1' application window with the title bar 'cube 4.1.1 livedvd2: scorep-20120913_1740_557443655223384/profile.cubex'. The 'Help' menu is open, showing options: 'Getting started', 'Mouse and keyboard control', 'What's This?' (highlighted with a blue bar and 'Shift+F1'), and 'About'. A blue callout bubble points to the 'What's This?' option with the text 'Context-sensitive help available for all GUI items'. The main window is divided into three panes. The left pane, titled 'Metric tree', shows a list of metrics: '1.63e9 Visits', '767.48 Time' (highlighted), '0.00 Minimum I', '48.58 Maximum I', '5.27e8 byt', and '5.27e8'. The middle pane shows a hierarchical tree of operations: 'compute_rhs_', 'solve_', '218.21 !\$omp parallel @x_sol', '15.18 !\$omp implicit bar', '0.07 y_solve_', '0.04 !\$omp parallel @y_sol', '223.63 !\$omp do @y_sol', '15.60 !\$omp implicit bar', '0.07 z_solve_', '0.04 !\$omp parallel @z_sol', '226.71 !\$omp do @z_sol', '14.79 !\$omp implicit bar', '1.86 add_', '0.01 MPI_Barrier', '0.00 timer_start_', '0.00 timer_stop_', '0.00 timer_read_', '0.60 verify_', '0.00 MPI_Reduce', and '0.00 print_results_'. The right pane, titled 'System tree', shows a hierarchical tree of system components: '- generic cluster', '- i06r01c20', '- MPI Rank 0' (with 4 CPU threads: 44.10, 43.94, 43.58, 34.36), '- MPI Rank 1' (with 4 CPU threads: 44.48, 35.09, 44.18, 43.71), '- MPI Rank 2' (with 4 CPU threads: 44.47, 43.49, 43.86, 35.18), and '- MPI Rank 3' (with 4 CPU threads: 44.74, 44.00, 43.73, 35.64). At the bottom, there are three progress bars. The first bar shows '0.00 767.48 (100.00%) 767.48'. The second bar shows '0.00 668.54 (87.11%) 767.48'. The third bar shows '0.00 668.54'. A status bar at the bottom reads 'Change into help mode for display components'.

CUBE algebra utilities

- Extracting solver sub-tree from analysis report

```
% cube_cut -r '<<ITERATION>>' scorep_bt-mz_B_8x8_sum/profile.cubex  
Writing cut.cubex... done.
```

- Calculating difference of two reports

```
% cube_diff scorep_bt-mz_B_8x8_sum/profile.cubex cut.cubex  
Writing diff.cubex... done.
```

- Additional utilities for merging, calculating mean, etc.
- Default output of cube_utility is a new report utility.cubex
- Further utilities for report scoring & statistics
- Run utility with “-h” (or no arguments) for brief usage info

Loop Unrolling

- Show time dependent behavior by unrolling iterations

- Preparations:

- Mark loops by using Score-P user instrumentation in your source code

```
SCOREP_USER_REGION_BEGIN( scorep_bt_loop, "<<bt_iter>>", SCOREP_USER_REGION_TYPE_DYNAMIC )
```

- Result in the CUBE profile:

- Iterations shown as separate call trees

- Useful for checking results for specific iterations

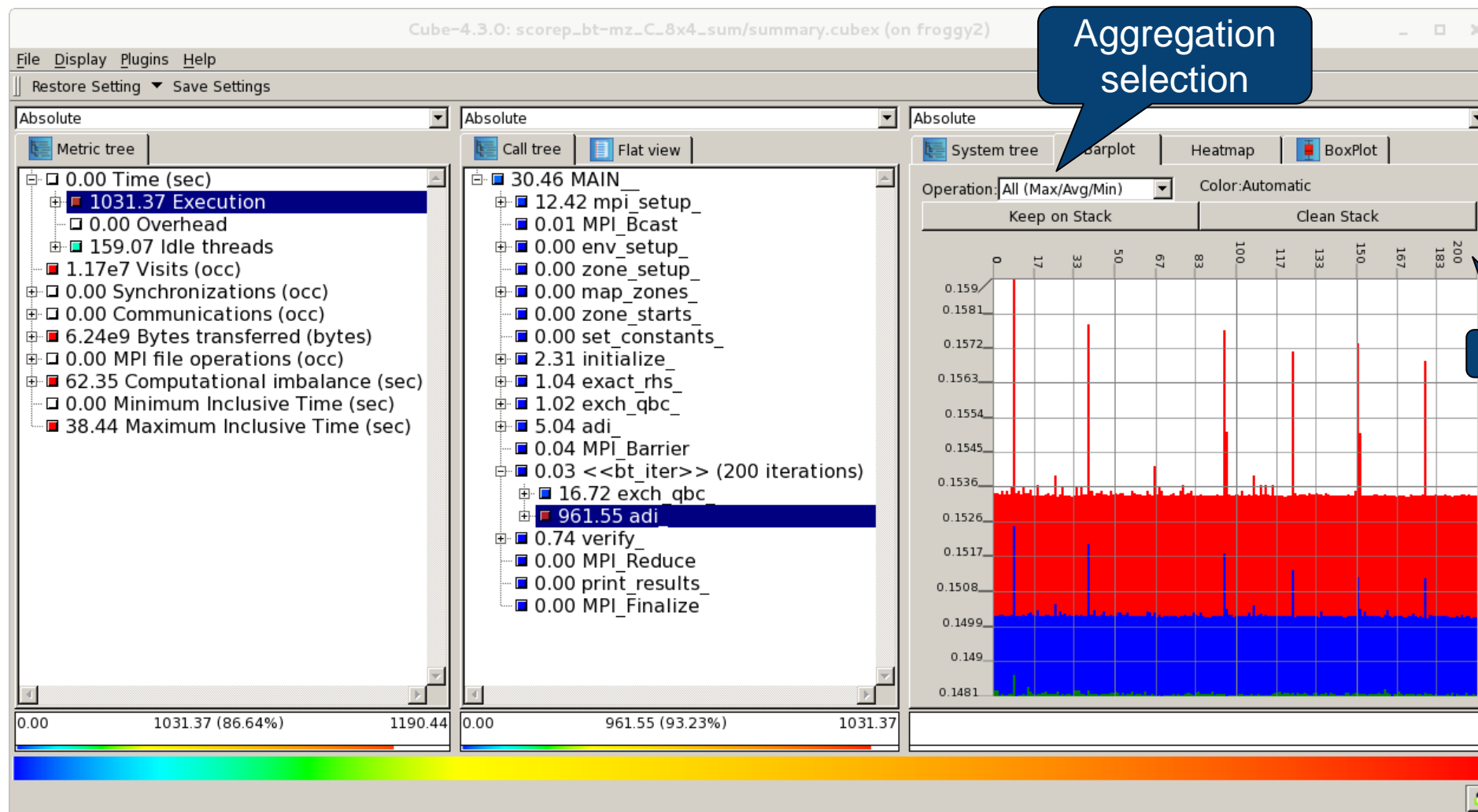
or

- Select your user instrumented region and mark it as loop

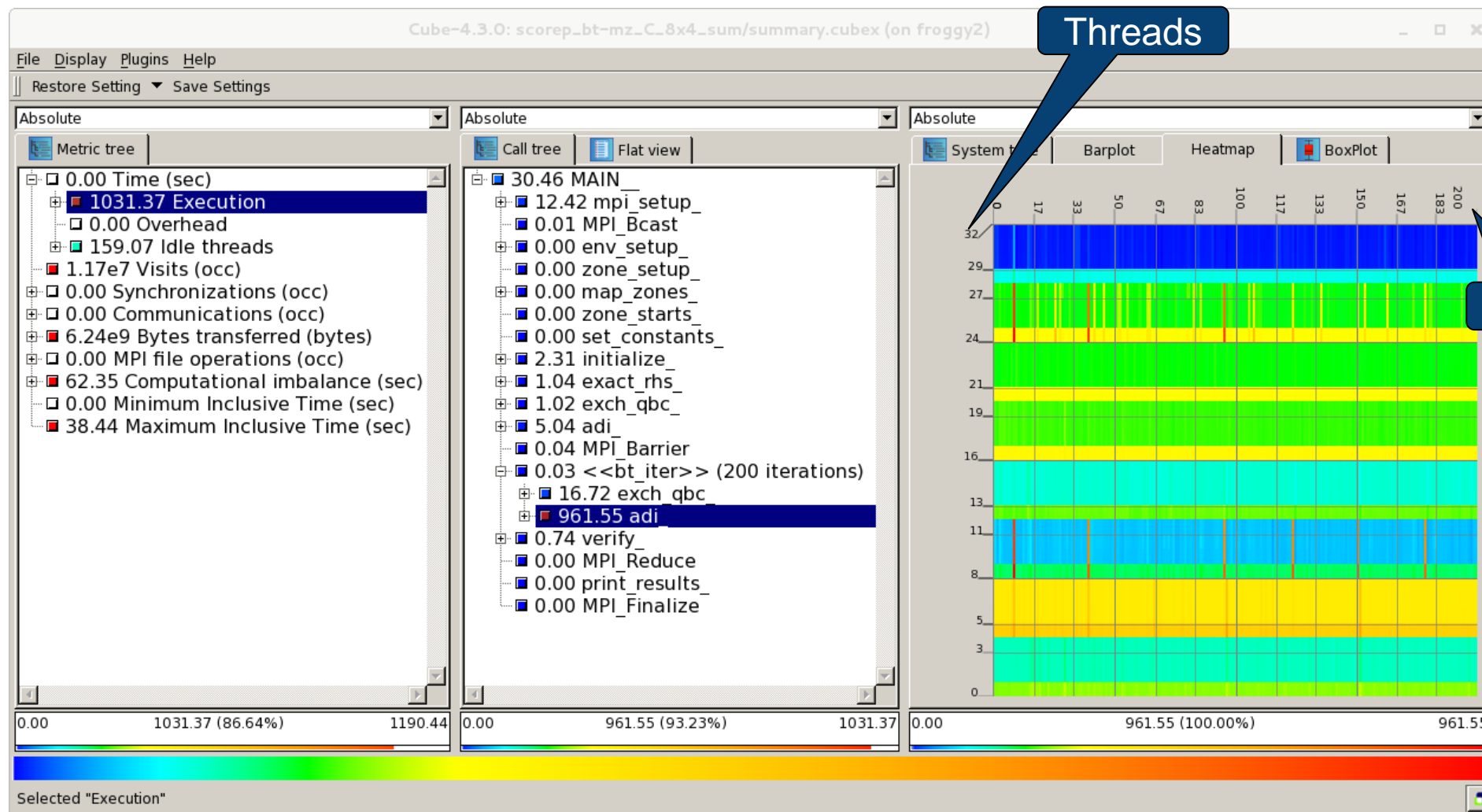
- Choose hide iterations

- View the Barplot statistics or the (thread x iterations) Heatmap

Loop Unrolling - Barplot



Loop Unrolling – Heatmap



Further information

CUBE

- Parallel program analysis report exploration tools
 - Libraries for XML report reading & writing
 - Algebra utilities for report processing
 - GUI for interactive analysis exploration
- Available under New BSD open-source license
- Documentation & sources:
 - <http://www.scalasca.org>
- User guide also part of installation:
 - ``cube-config --cube-dir` /share/doc/CubeGuide.pdf`
- Contact:
 - mailto: scalasca@fz-juelich.de

