

Analysis report examination with Cube

Marc-André Hermanns
Jülich Supercomputing Centre

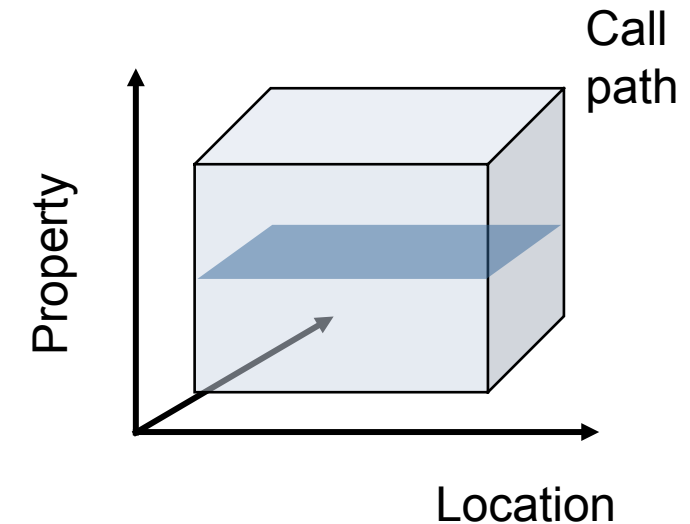


Cube

- Parallel program analysis report exploration tools
 - Libraries for XML+binary report reading & writing
 - Algebra utilities for report processing
 - GUI for interactive analysis exploration
 - Requires Qt4 ≥ 4.6 or Qt 5
- Originally developed as part of the Scalasca toolset
- Now available as a separate component
 - Can be installed independently of Score-P, e.g., on laptop or desktop
 - Latest release: Cube 4.3.4 (April 2016)

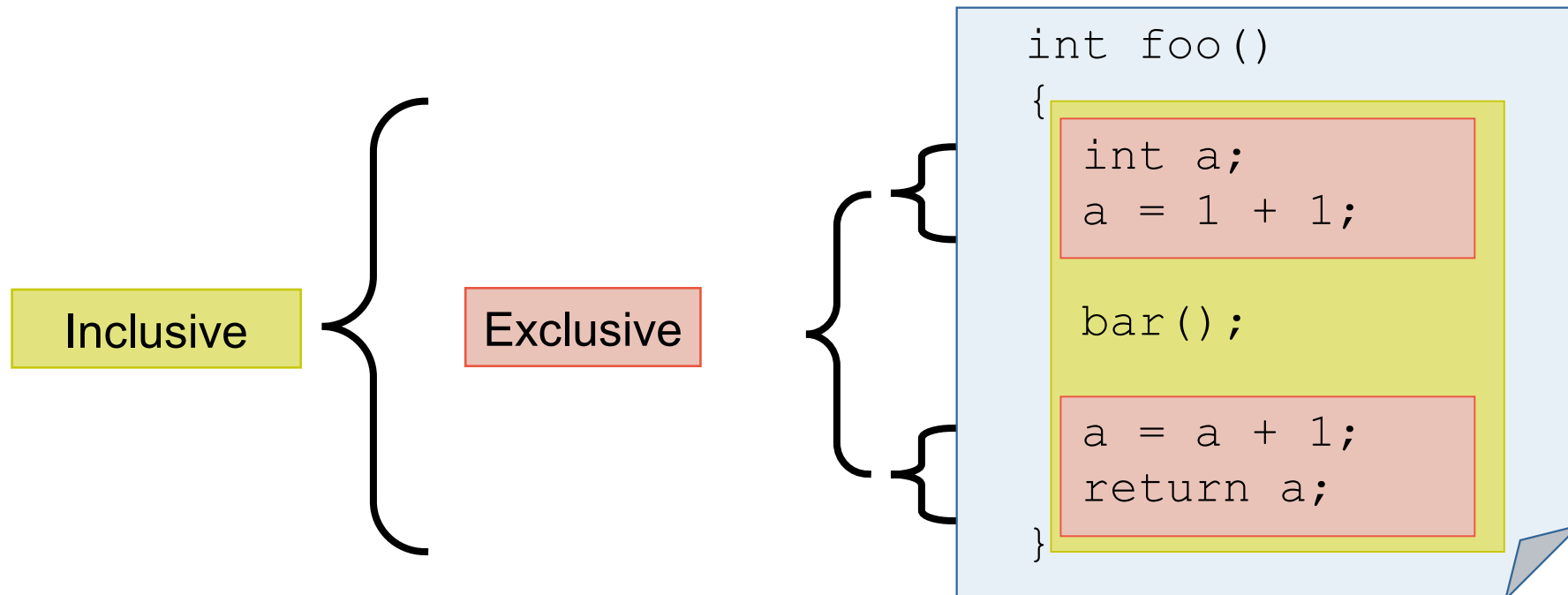
Analysis presentation and exploration

- 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 color: for easy identification of hotspots
 - Inclusive value when closed & exclusive value when expanded
 - Customizable via display modes

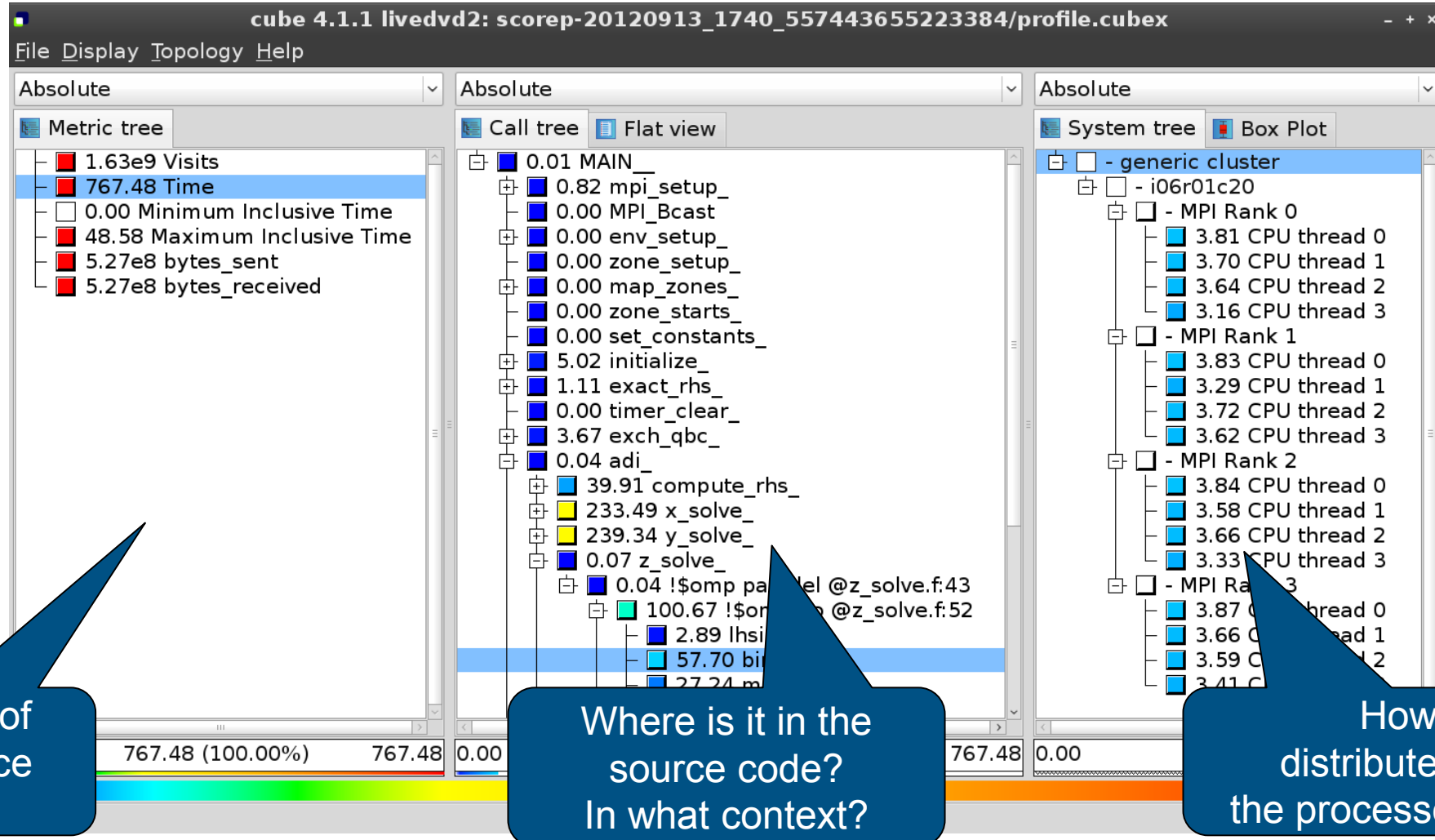


Inclusive vs. exclusive values

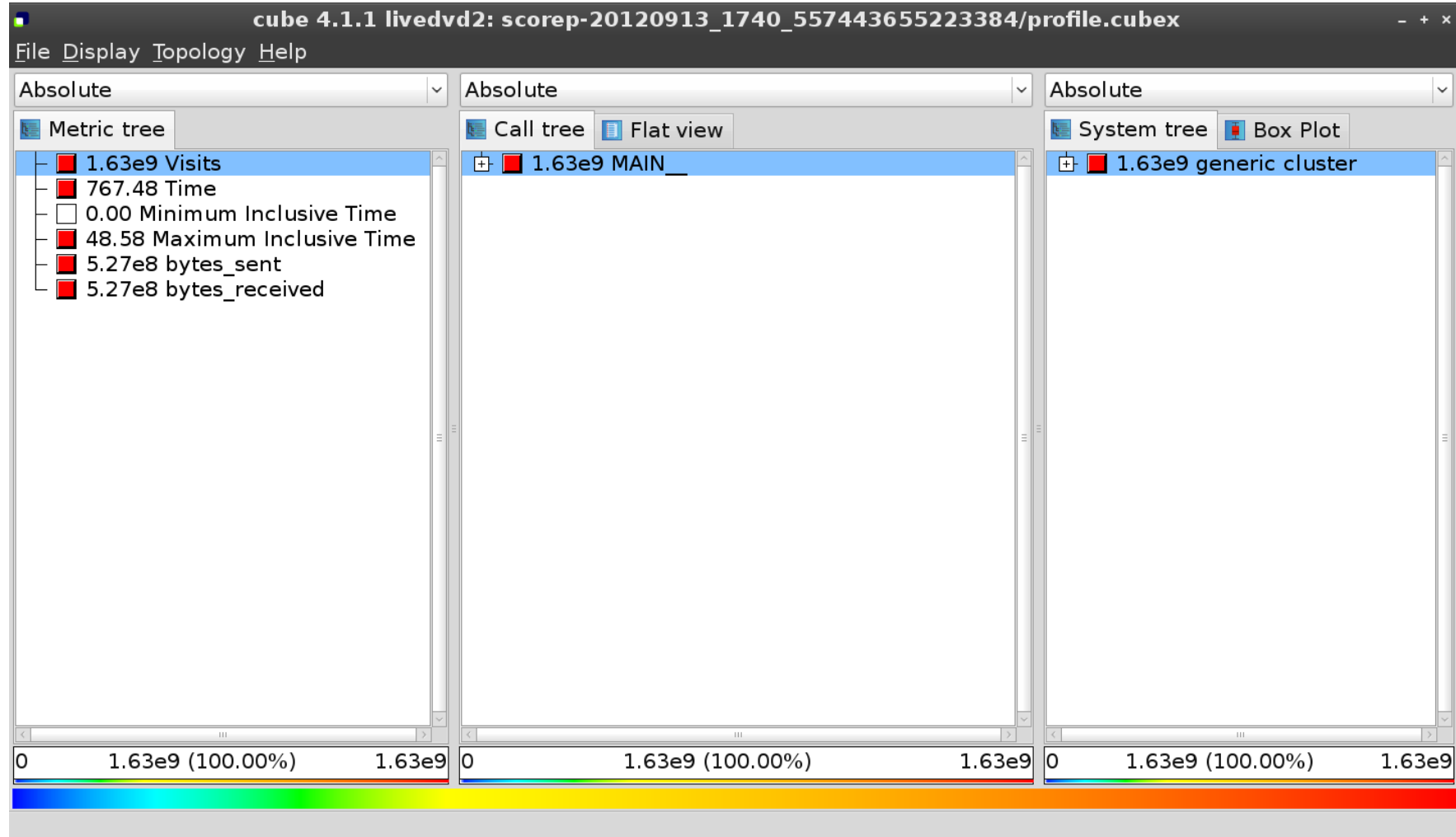
- Inclusive
 - Information of all sub-elements aggregated into single value
- Exclusive
 - Information cannot be subdivided further



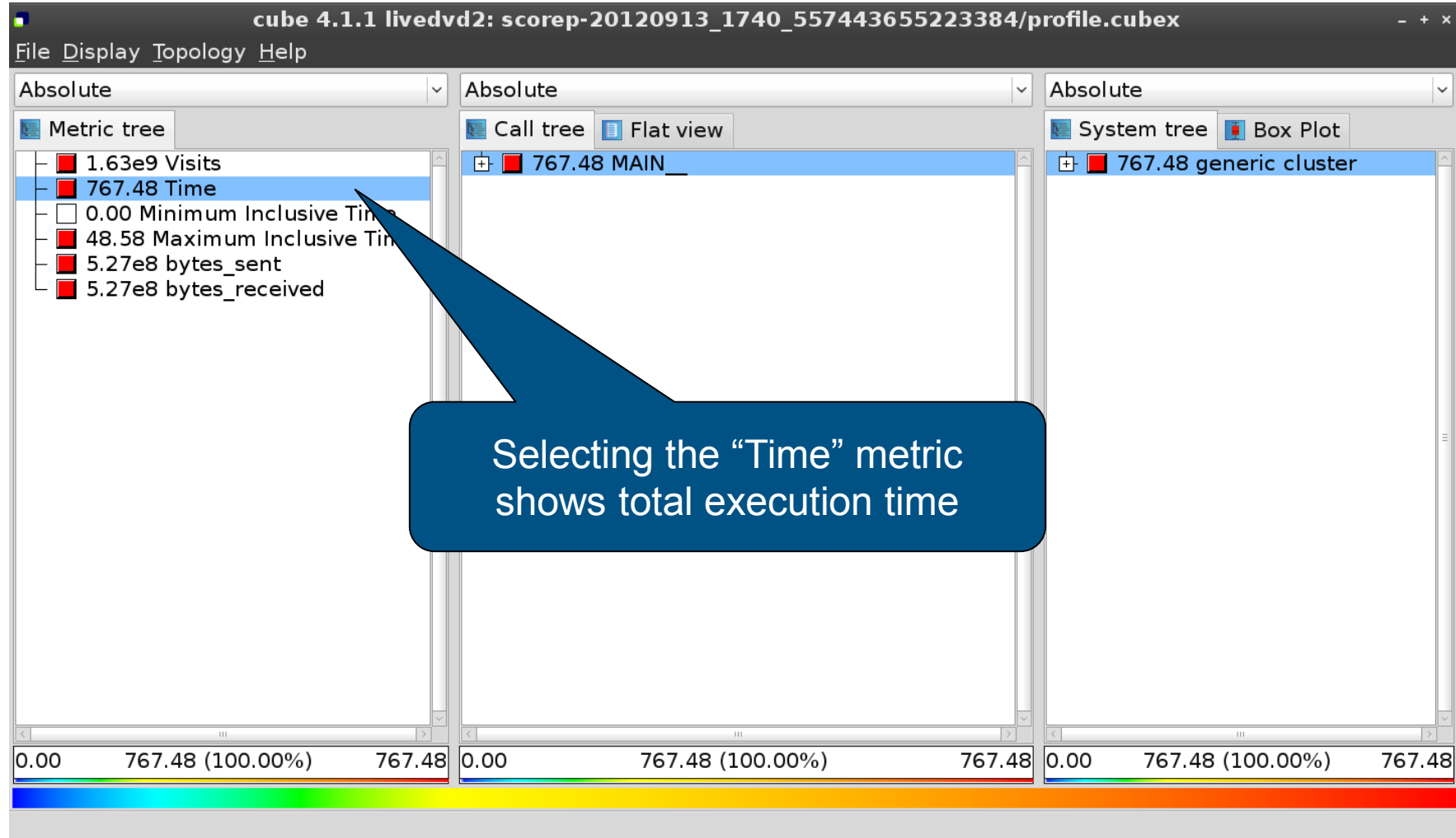
Analysis presentation



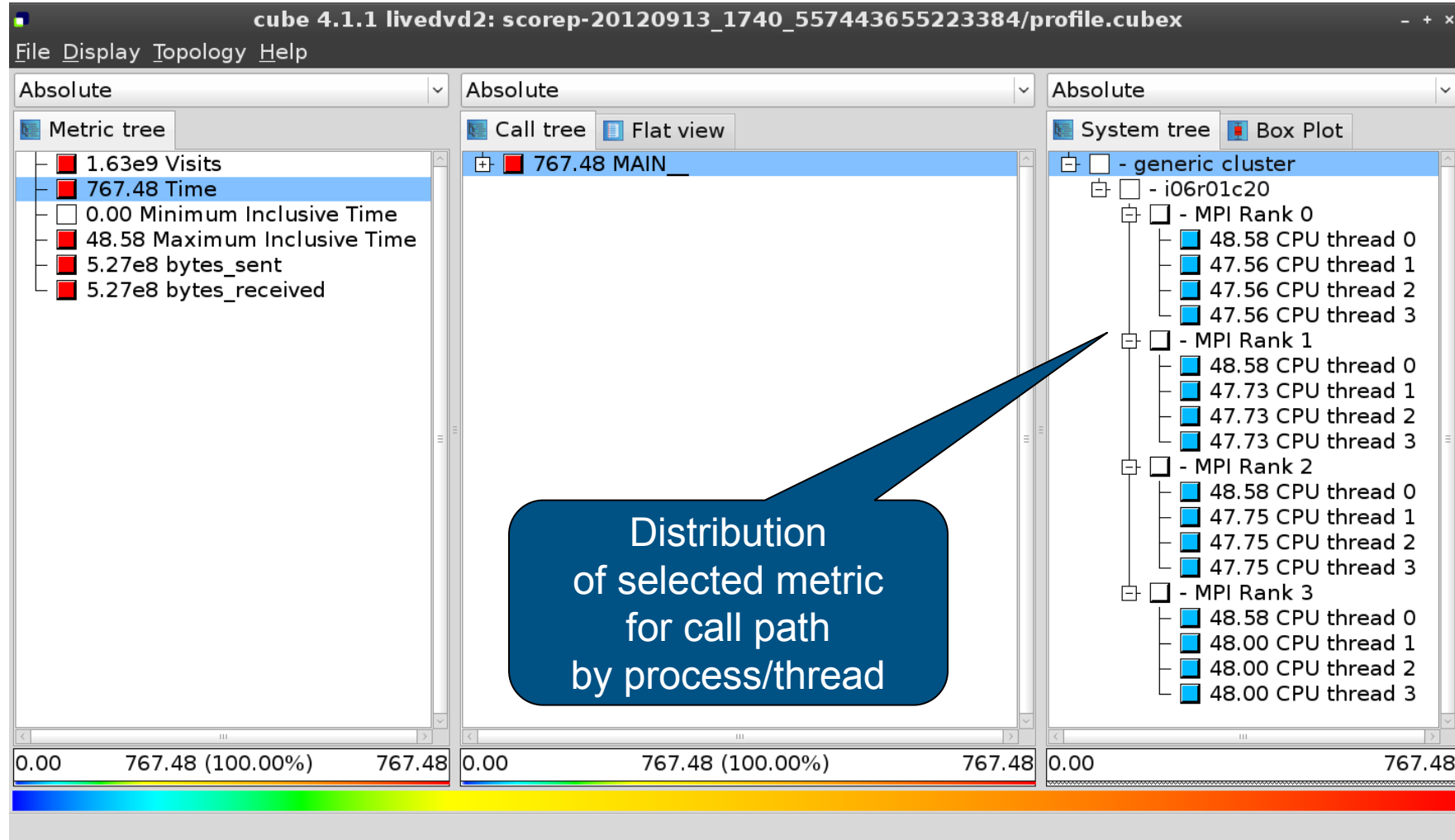
Score-P analysis report exploration (opening view)



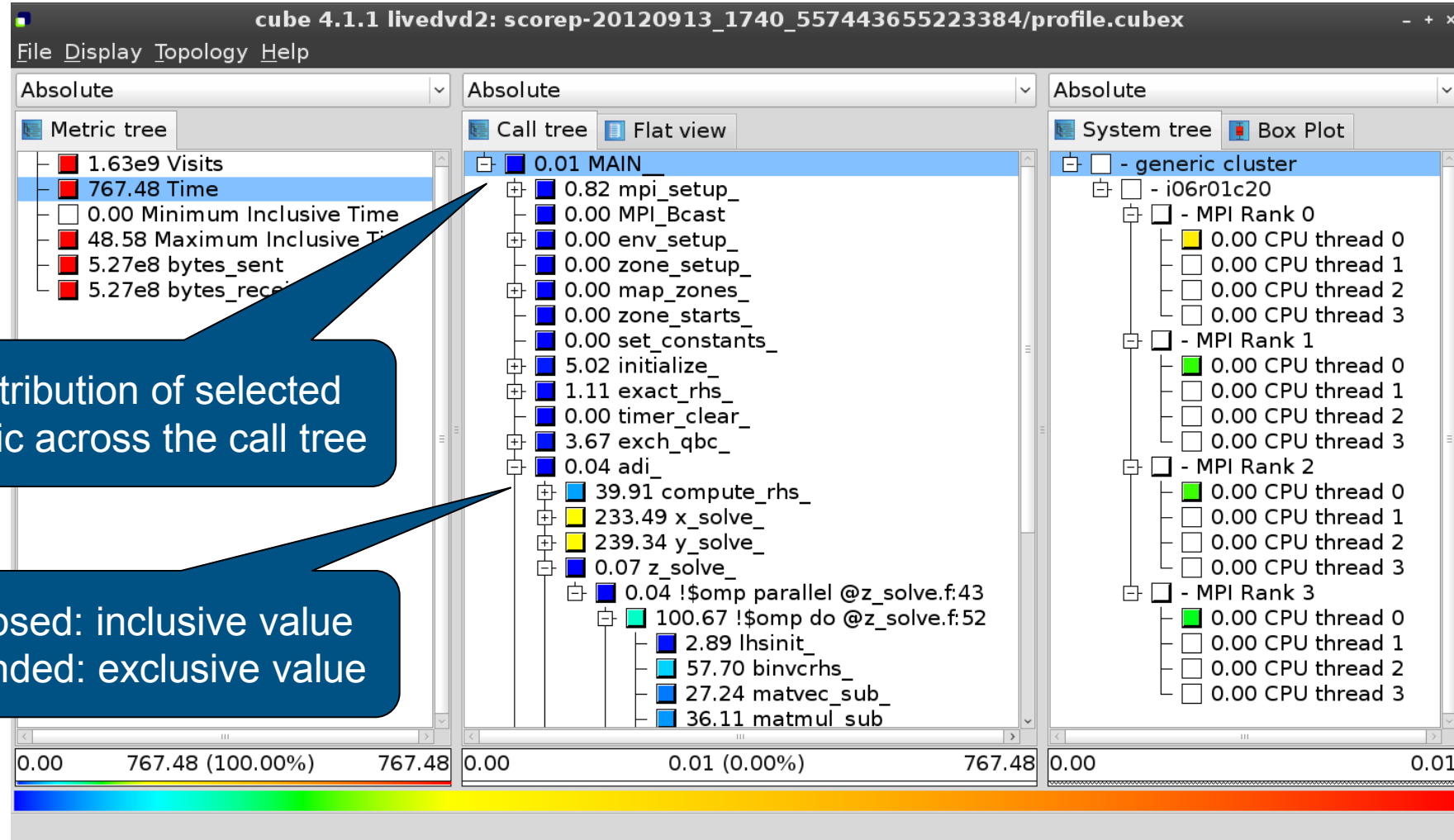
Metric selection



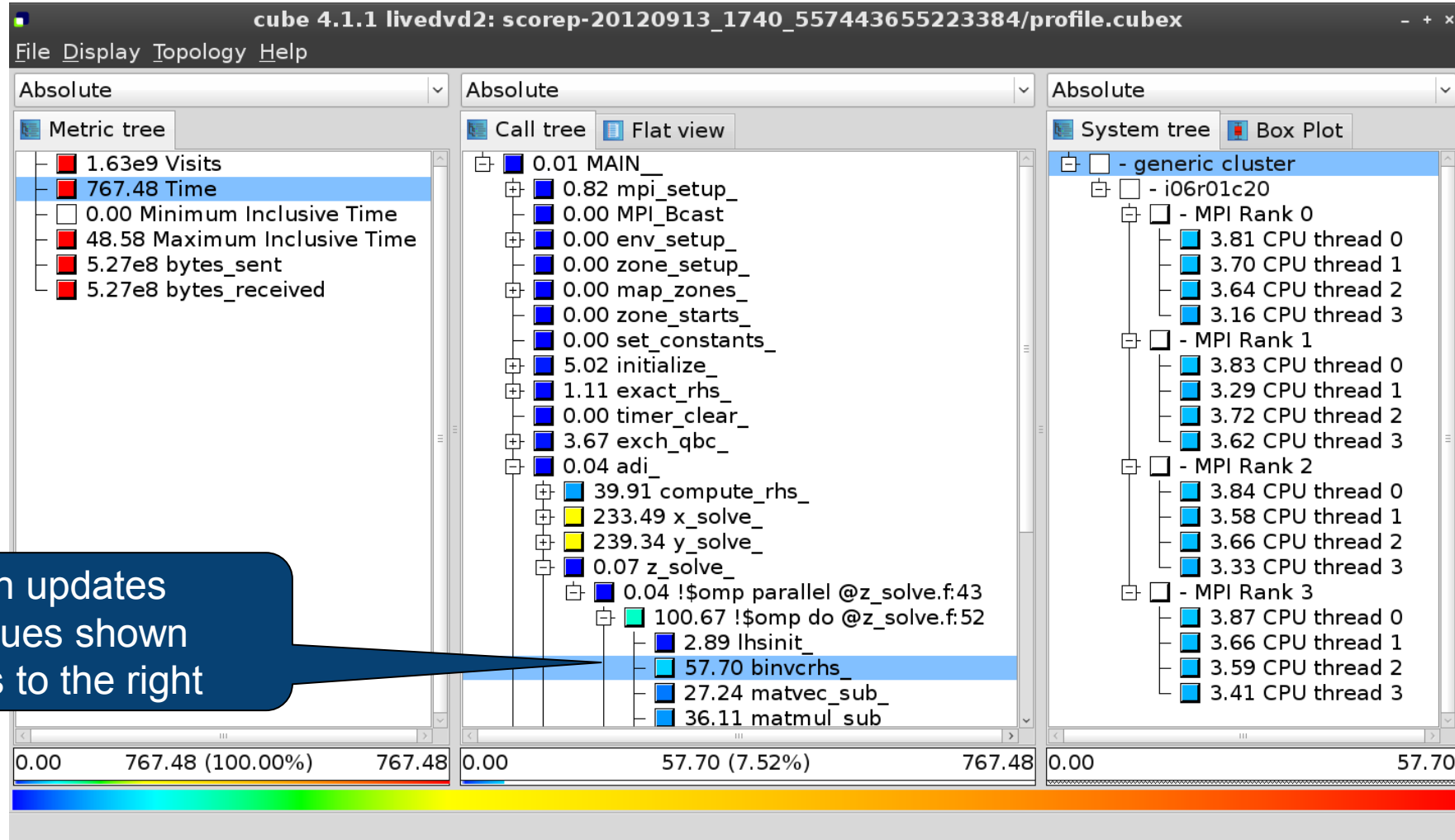
Expanding the system tree



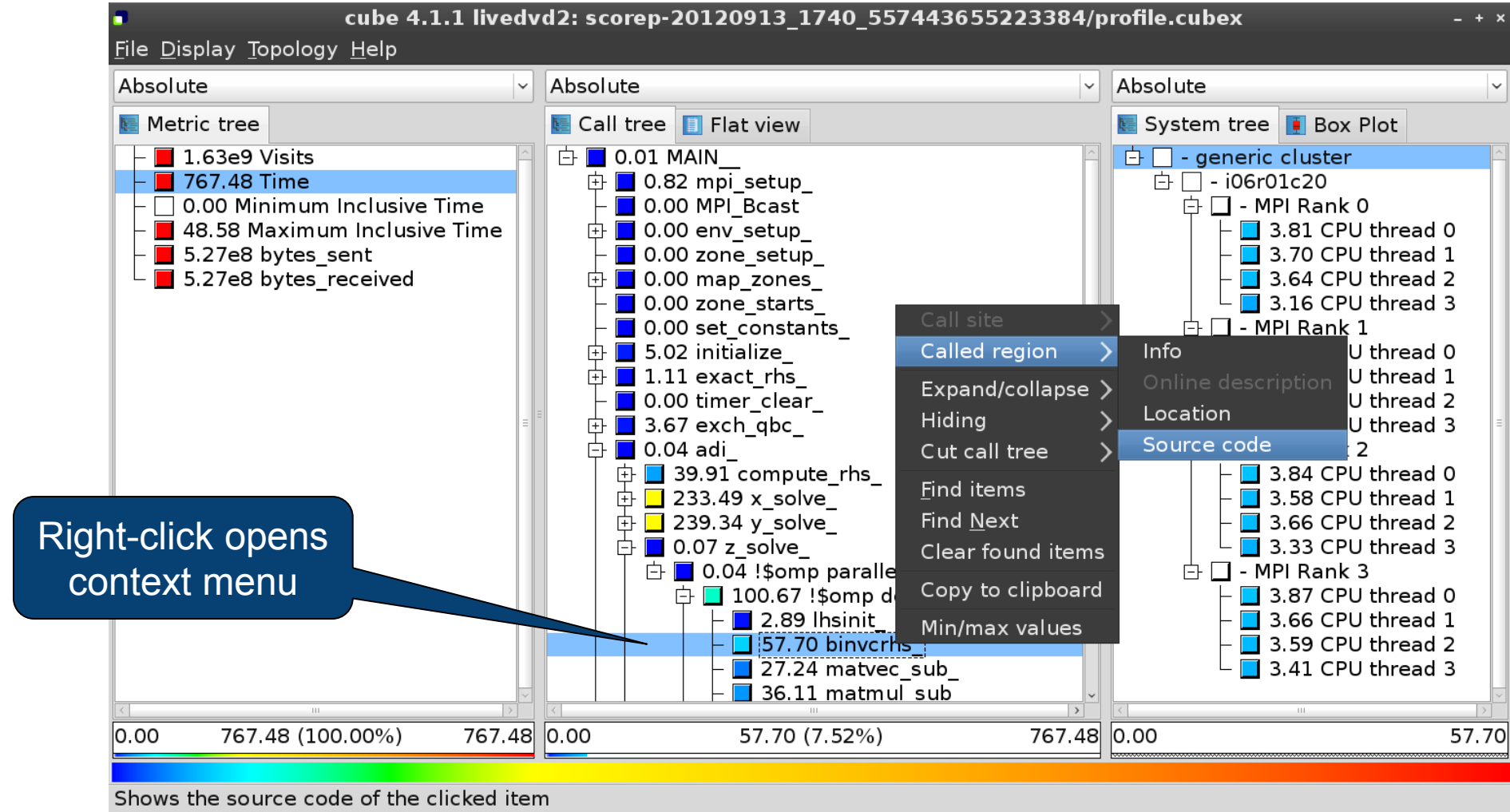
Expanding the call tree



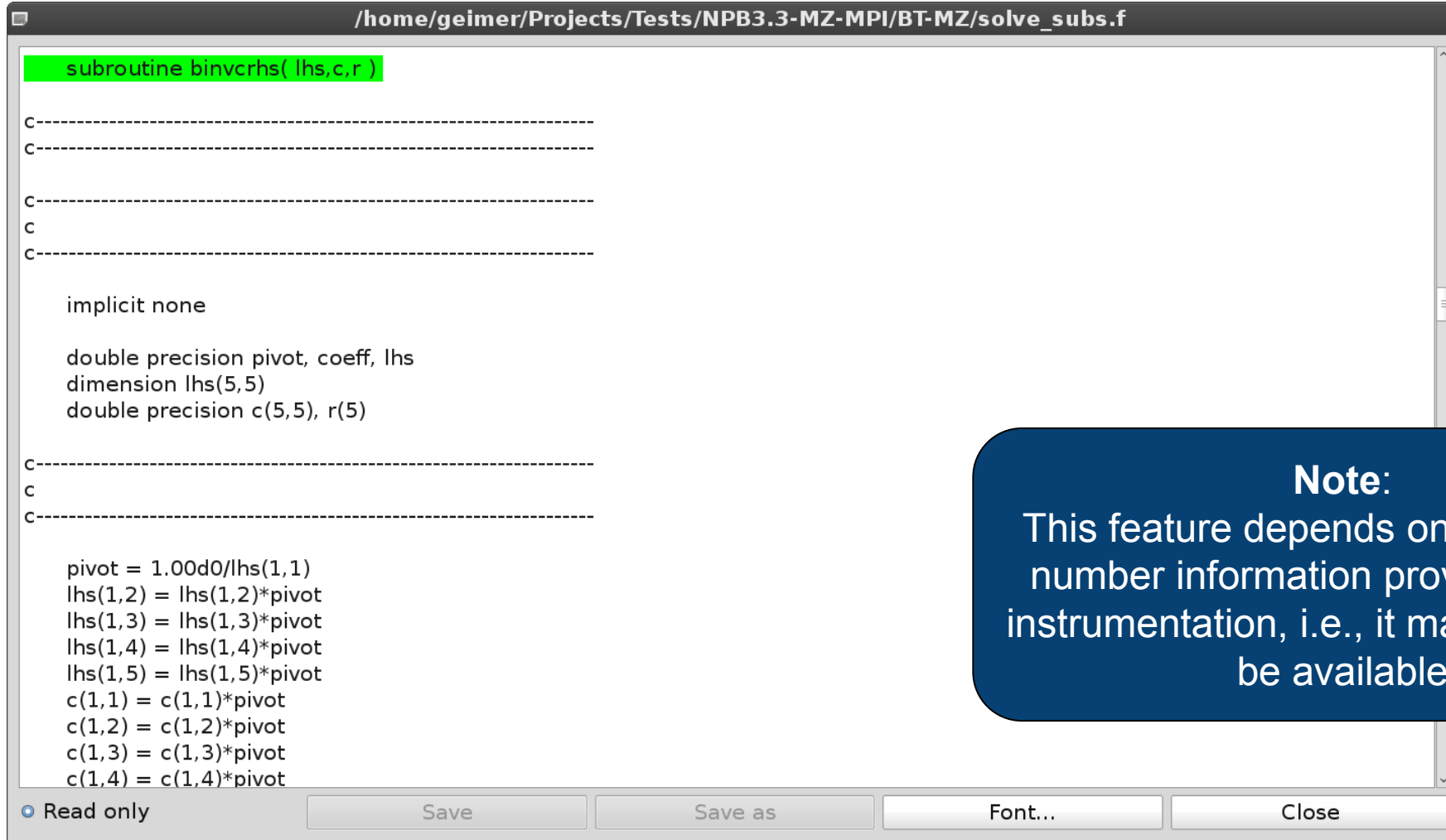
Selecting a call path



Source-code view via context menu



Source-code view



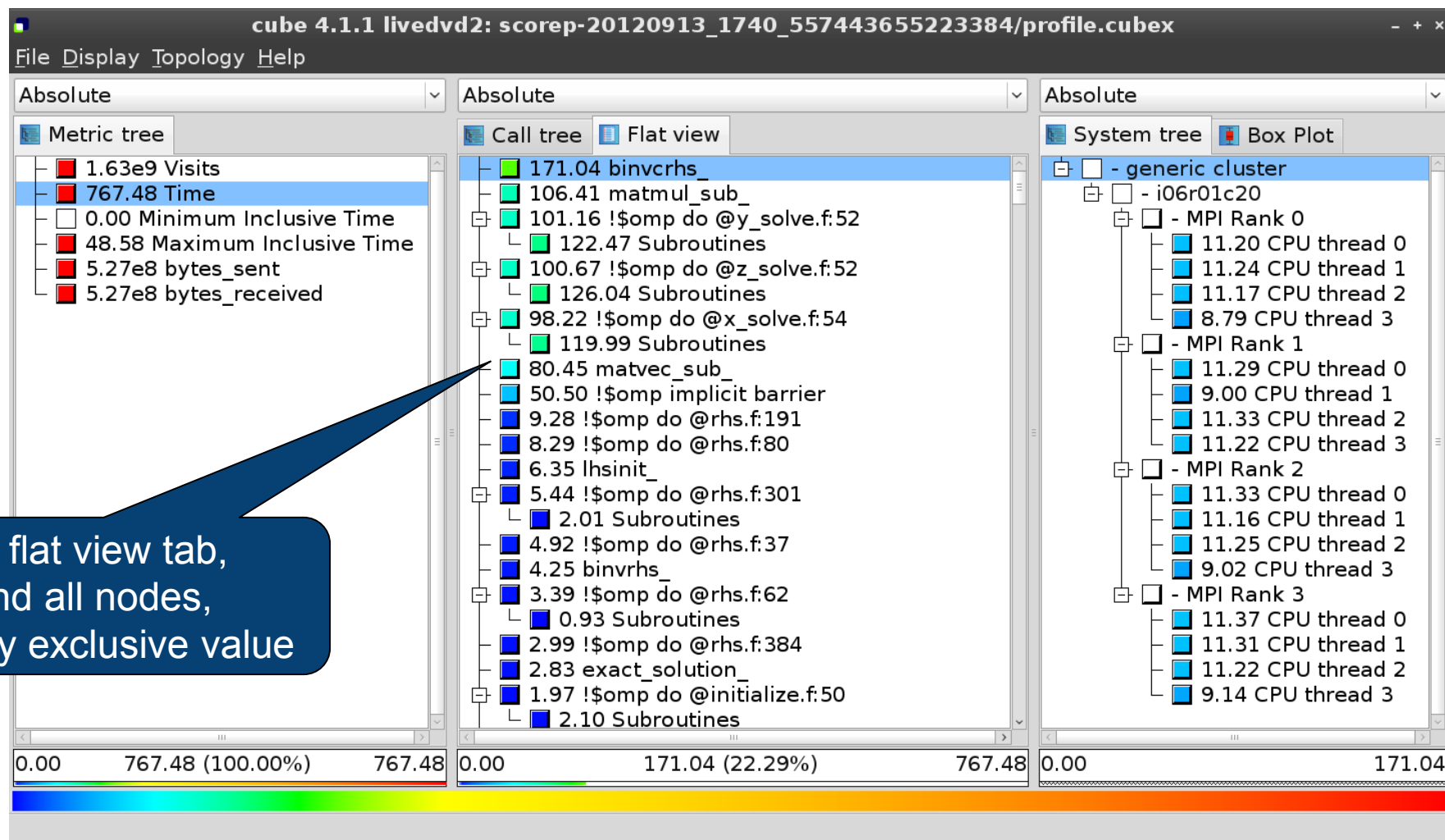
```
subroutine binvrhs( lhs,c,r )  
  
C-----  
C-----  
  
C-----  
C  
C-----  
  
implicit none  
  
double precision pivot, coeff, lhs  
dimension lhs(5,5)  
double precision c(5,5), r(5)  
  
C-----  
C  
C-----  
  
pivot = 1.00d0/lhs(1,1)  
lhs(1,2) = lhs(1,2)*pivot  
lhs(1,3) = lhs(1,3)*pivot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

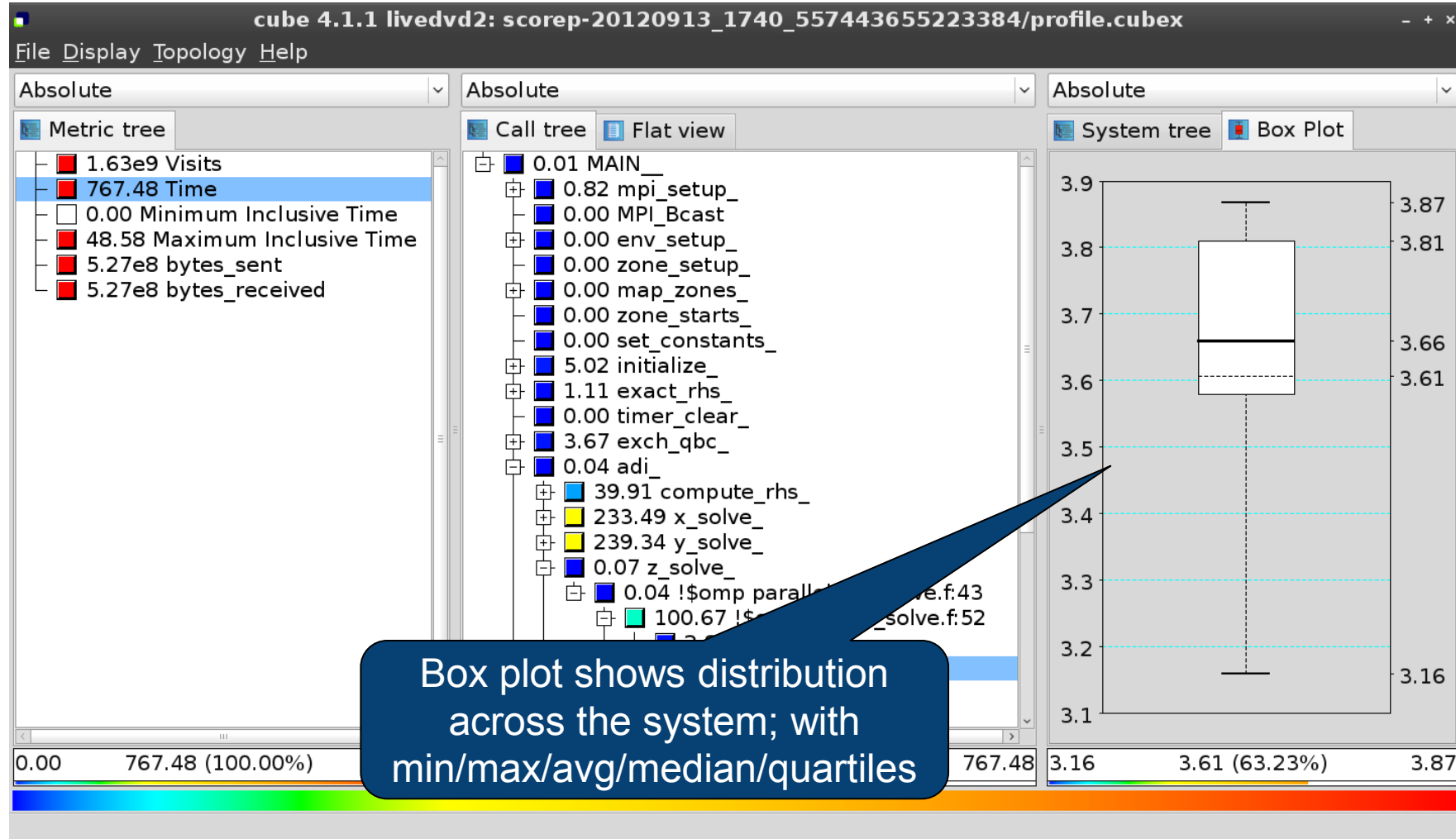
Note:

This feature depends on file and line number information provided by the instrumentation, i.e., it may not always be available

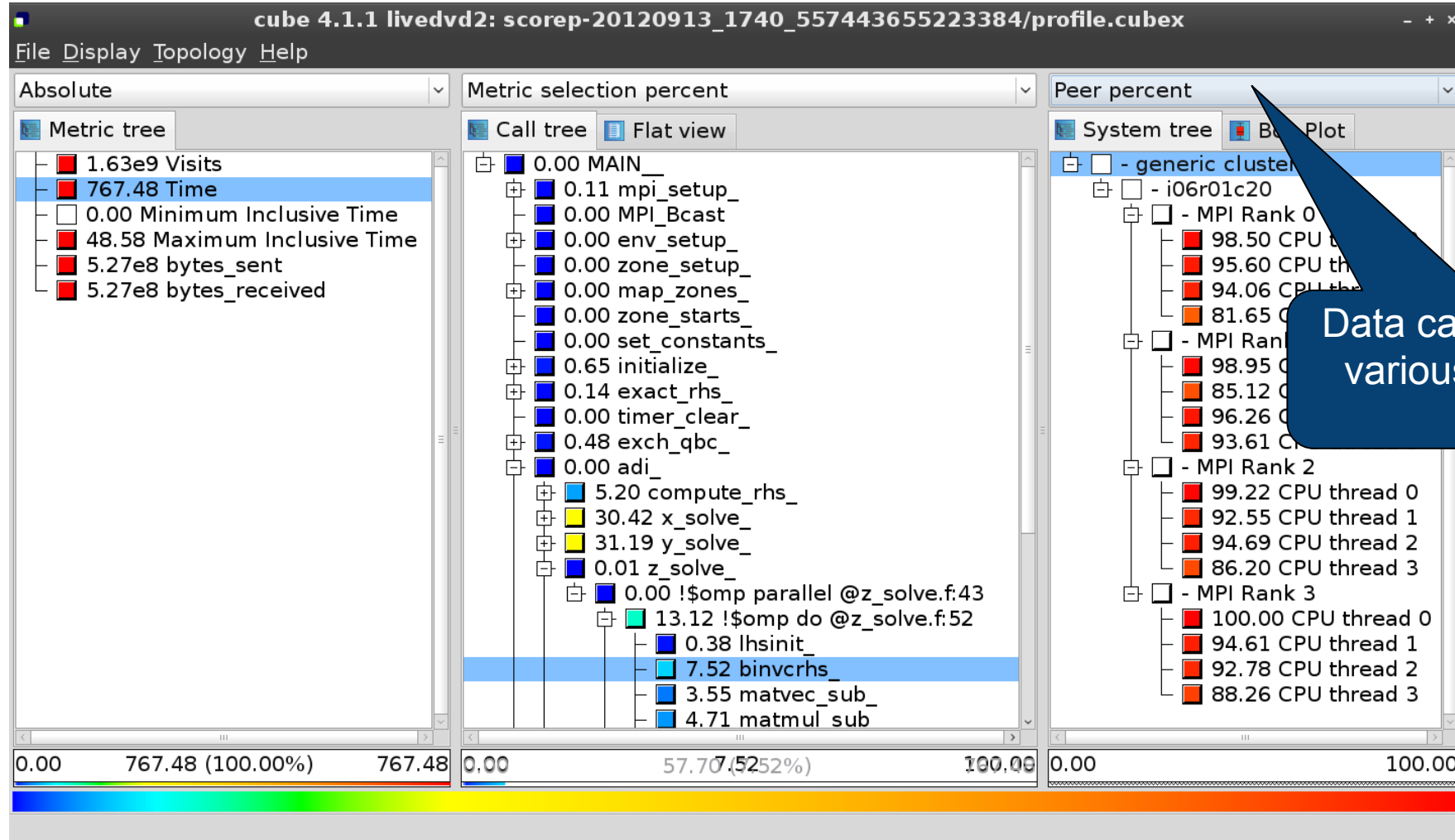
Flat profile view



Box plot view

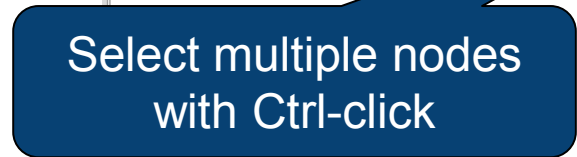


Alternative display modes



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



Context-sensitive help

The screenshot displays the cube 4.1.1 GUI window titled "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 mouse cursor and the keyboard shortcut "Shift+F1"), and "About". The "What's This?" option is being hovered over, revealing a sub-menu with "Selected metrics description" and "Selected regions description".

The GUI is divided into several panels:

- Metric tree:** A list of metrics with checkboxes. The "767.48 Time" metric is selected and highlighted in blue.
- System tree:** A hierarchical tree view showing the system structure. The "generic cluster" is selected, and the "i06r01c20" node is expanded, showing MPI Ranks 0 through 3 and their respective CPU threads.
- Bottom status bar:** Displays performance metrics and a color-coded progress bar. The first section shows "0.00 767.48 (100.00%) 767.48". The second section shows "0.00 668.54 (87.11%) 767.48". The third section shows "0.00 668.54".

A blue callout box with a white arrow points to the "What's This?" menu item, containing the text: "Context-sensitive help available for all GUI items".

Change into help mode for display components

Derived metrics

- Derived metrics are defined using CubePL expressions, e.g.:
`metric::time(i)/metric::visits(e)`
- Values of derived metrics are not stored, but calculated on-the-fly
- Types of derived metrics:
 - Prederived: evaluation of the CubePL expression is performed before aggregation
 - Postderived: evaluation of the CubePL expression is performed after aggregation
- Examples:
 - “Average execution time”: Postderived metric with expression
`metric::time(i)/metric::visits(e)`
 - “Number of FLOP per second”: Postderived metric with expression
`metric::FLOP()/metric::time()`

Derived metrics in Cube GUI

The screenshot shows the Cube GUI interface with the 'Create new metric as a child of metric' dialog box open. The dialog box contains the following fields and options:

- Select metric from collection:** Average execution time (kenobi)
- Derived metric type:** Postderived metric
- Display name:** Average visit time
- Unique name:** avg_visit_time
- Data type:** DOUBLE
- Unit of measurement:** sec
- URL:**
- Description:** Calculates average time of region execution per visit. Autor is Michael Knobloch.
- Calculation:** ☒ Calculation ☐ Calculation Init ☐ Aggregation "*" ☐ Aggregation "/"
- Calculation expression:** `metric::time()|metric::visits(e)`
- Buttons:** Create metric, Cancel
- Link:** [Share this metric with SCALASCA group](#)

The background shows the 'Metric tree' panel with a list of metrics:

- 1.09e8 Visits (occ)
- 1.01e6 Time (sec)
- 0.00 Minimum Inclusive Time (sec)
- 246.14 Maximum Inclusive Time (sec)
- 7.18e12 bytes_sent
- 7.18e12 bytes_received

Example: FLOPS based on PAPI_FP_OPS and time

The screenshot displays the Cube-4.3.1 software interface, which is used for analyzing performance metrics. It is divided into several panels:

- Edit metric FLOPS (on froggy1):** This panel on the left allows for configuring a derived metric. It shows the selection of 'PAPI_FP_OPS' and 'time' as the base metrics, with the formula `metric::PAPI_FP_OPS()/metric::time()` entered. The display name is set to 'FLOPS', the unique name to 'flops', and the data type to 'DOUBLE'.
- Metric tree:** This panel shows a list of available metrics. The 'FLOPS' metric is highlighted, showing a value of 1.84e9.
- Call tree:** This panel shows a hierarchical view of the execution. The 'MAIN' node is expanded, showing various MPI and OpenMP operations. The 'exact_rhs' node is highlighted, showing a value of 9.65e8.
- System tree:** This panel shows the system configuration, including the node 'frog6' and the MPI ranks. The 'MPI Rank 0' node is expanded, showing the master thread and three OpenMP threads.

The bottom of the interface features a color-coded bar and a status bar indicating the selected metric and its value.

CUBE algebra utilities

- Extracting solver sub-tree from analysis report

```
% cube_cut -r '<<ITERATION>>' scorep_bt-mz_B_mic15p30x4_sum/profile.cubex  
Writing cut.cubex... done.
```

- Calculating difference of two reports

```
% cube_diff scorep_bt-mz_B_mic15p30x4_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

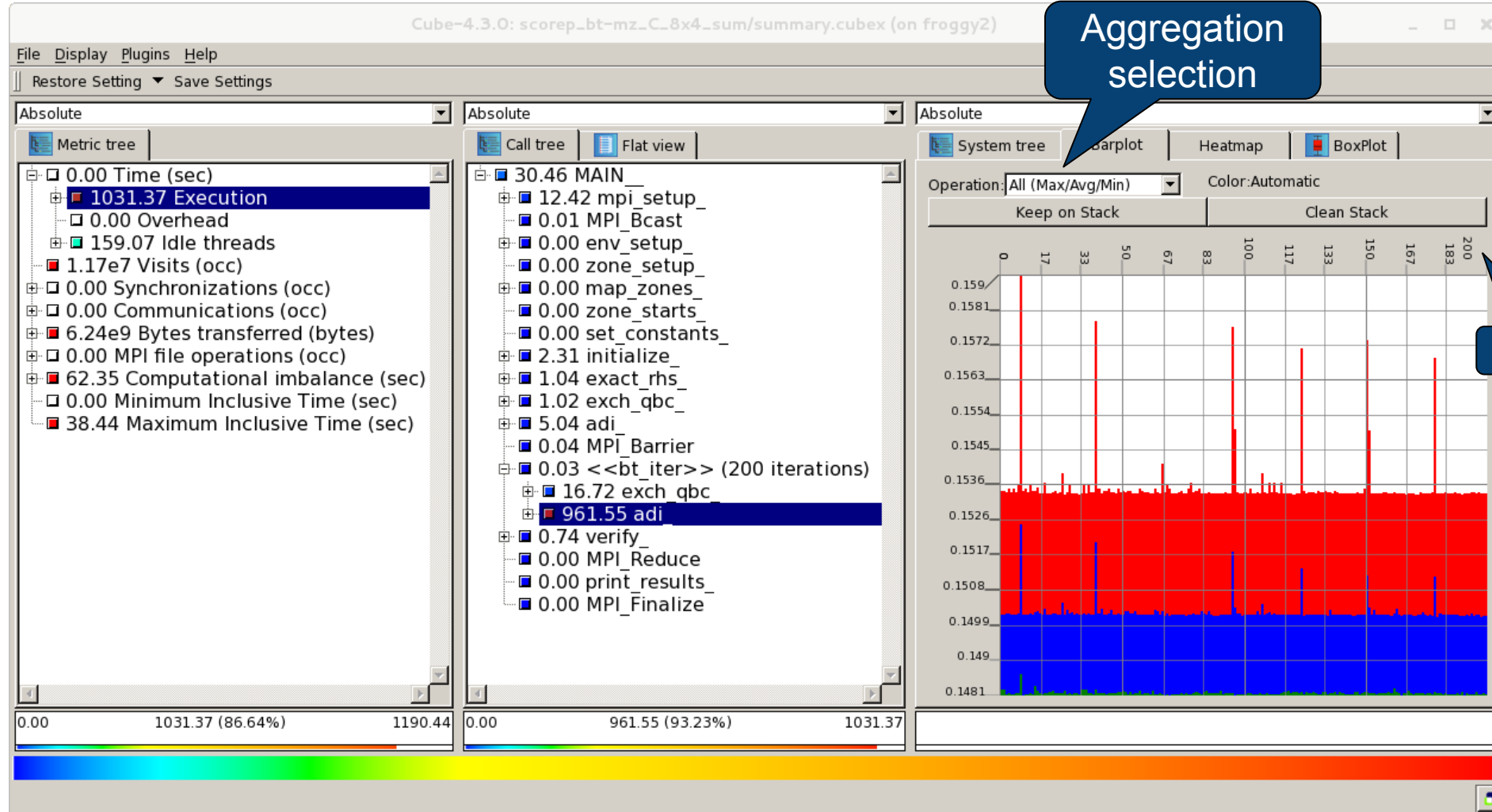
Iteration profiling

- Show time dependent behavior by “unrolling” iterations
- Preparations:
 - Mark loop body by using Score-P instrumentation API in your source code

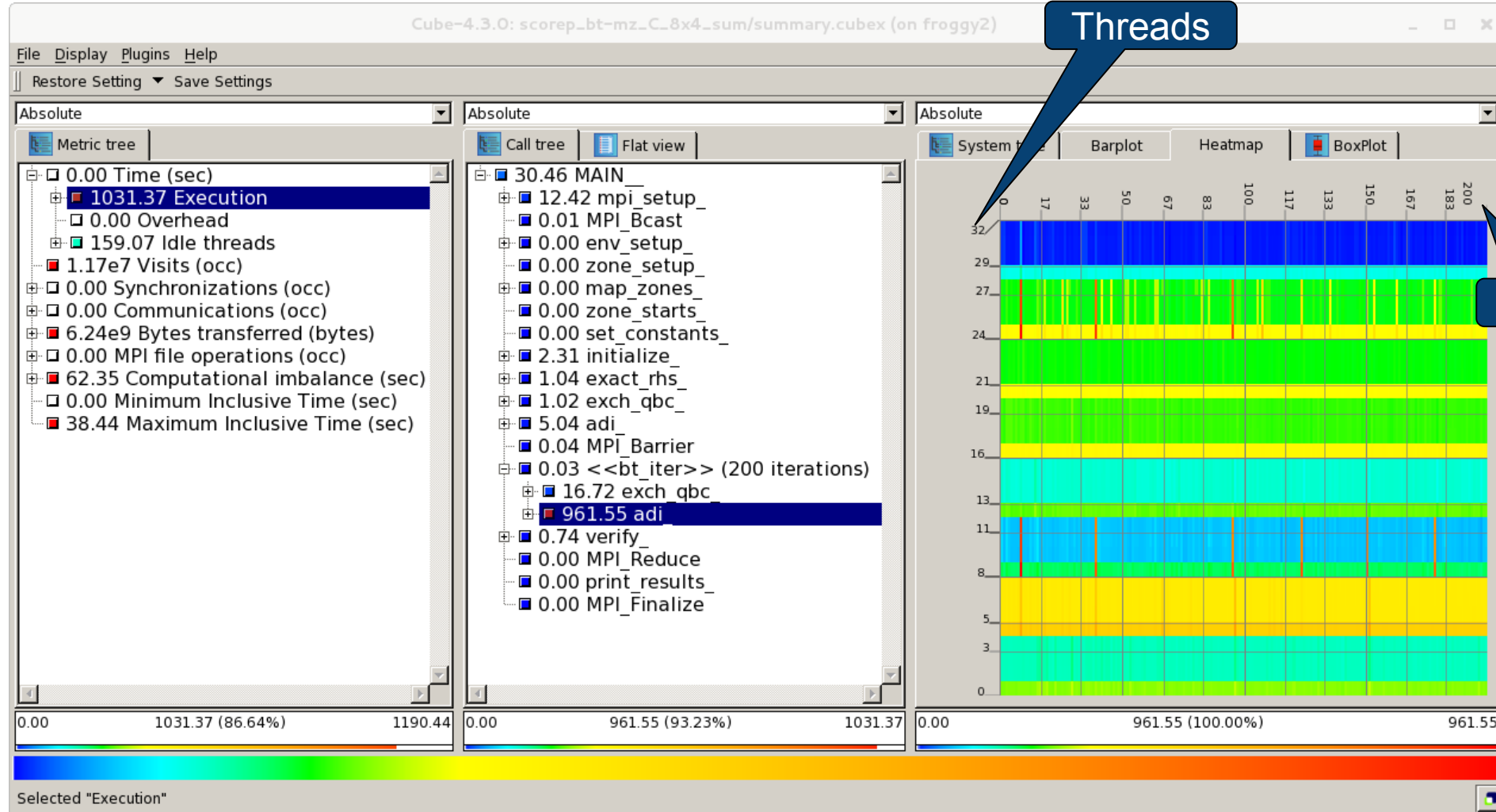
```
SCOREP_USER_REGION_DEFINE( scorep_bt_loop )  
SCOREP_USER_REGION_BEGIN( scorep_bt_loop, "<<bt_iter>>", SCOREP_USER_REGION_TYPE_DYNAMIC )  
SCOREP_USER_REGION_END( scorep_bt_loop )
```

- 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

Iteration profiling: Barplot



Iteration profiling: Heatmap



Cube: Further information

- Parallel program analysis report exploration tools
 - Libraries for XML report reading & writing
 - Algebra utilities for report processing
 - GUI for interactive analysis exploration
- Available under 3-clause 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

