

# Analysis report examination with Cube

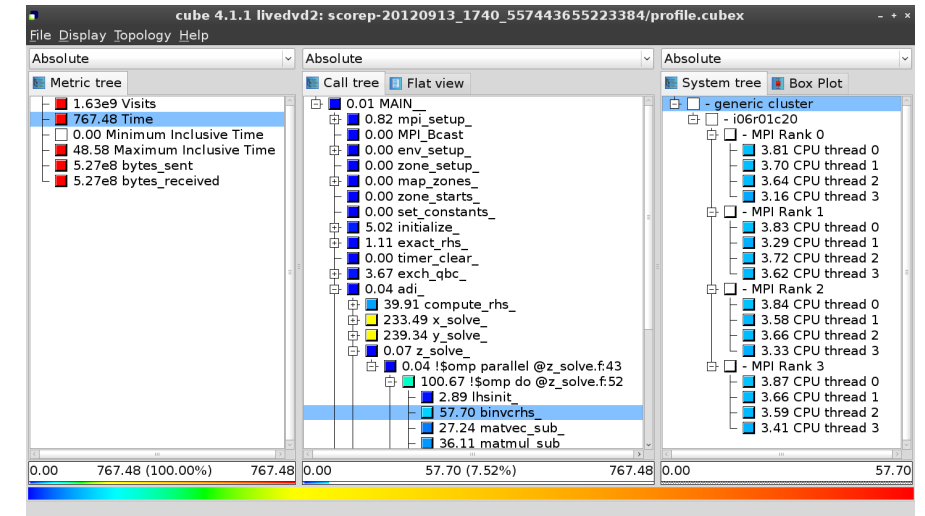
---

Brian Wylie  
Jülich Supercomputing Centre



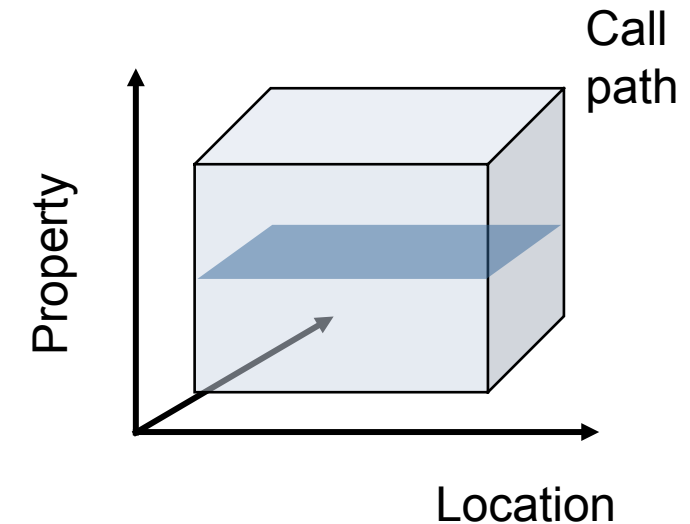
# Cube

- 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  $\geq 4.6$  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 notebook or desktop computers
  - Latest release: Cube 4.3.4 (April 2016)

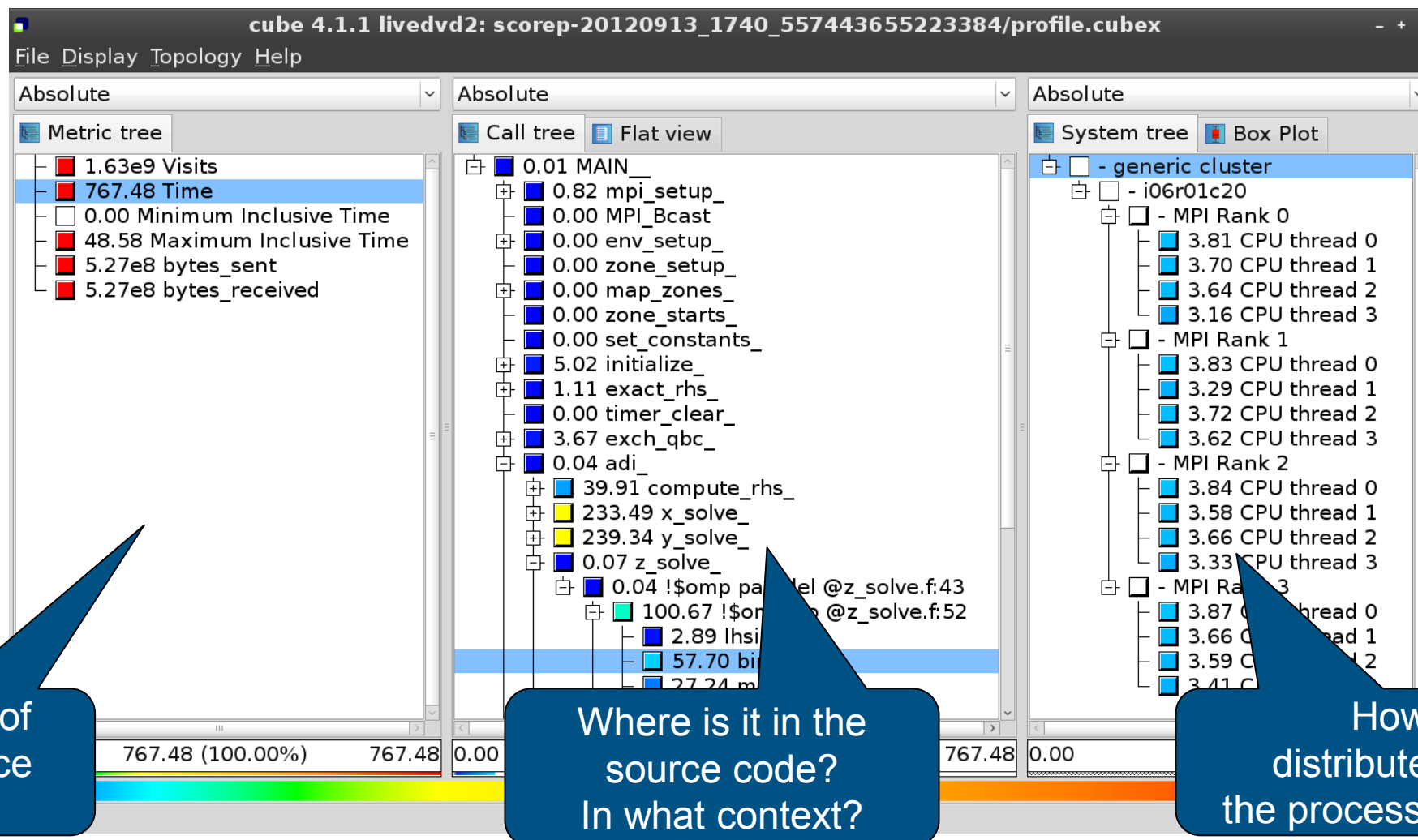


# Analysis presentation and exploration

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

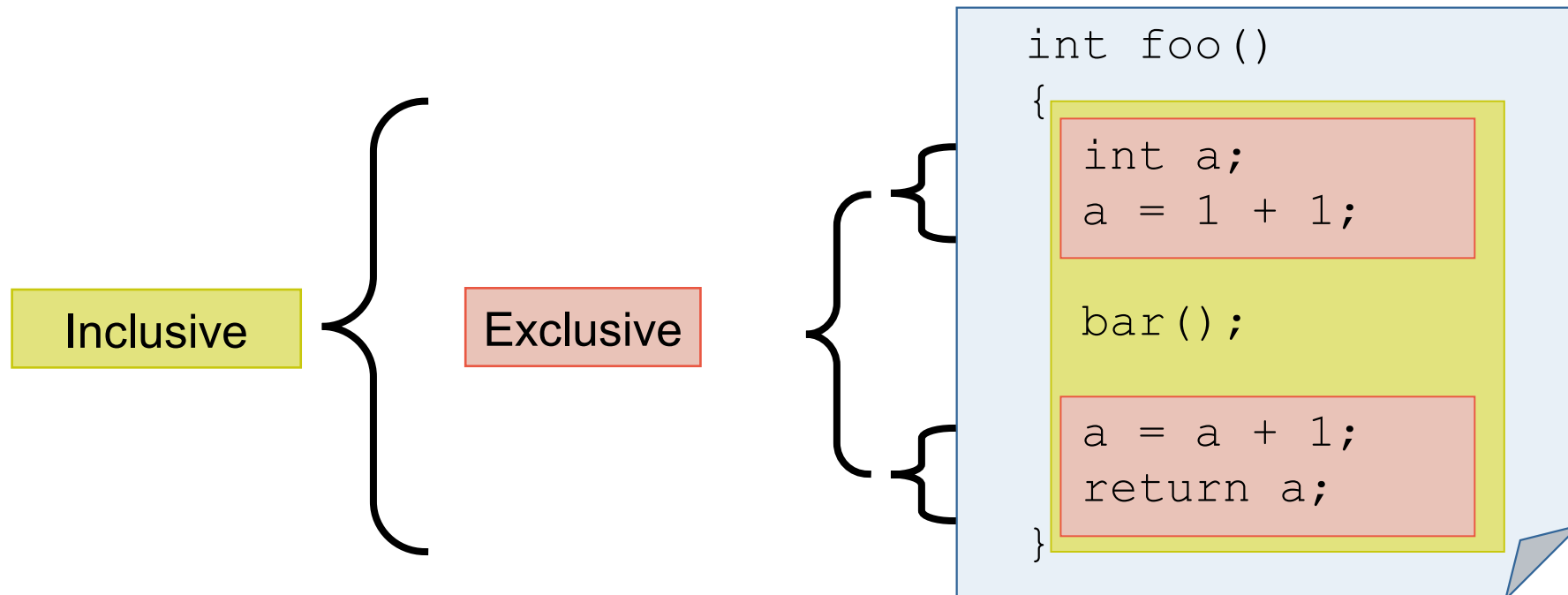


# Analysis presentation



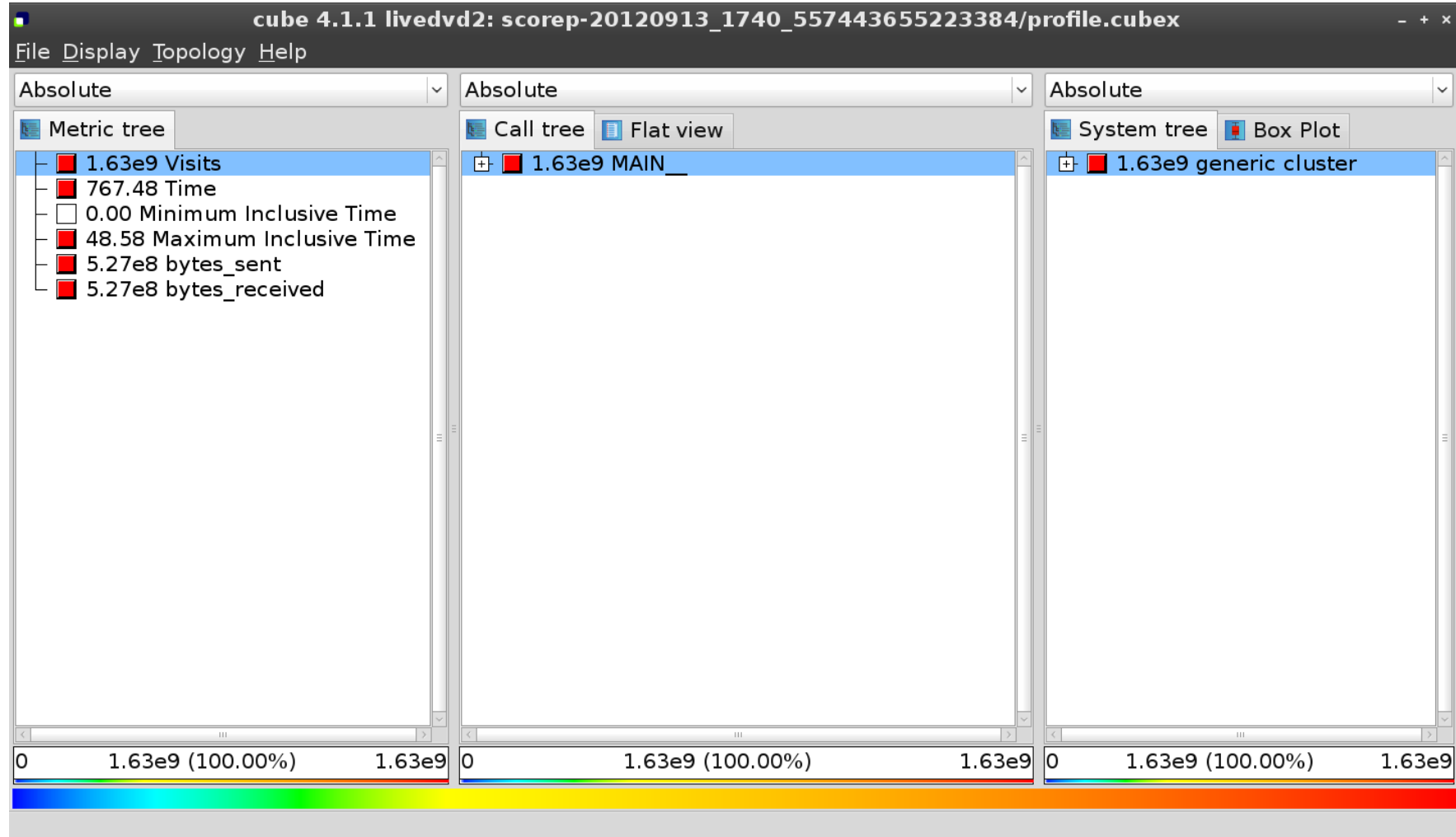
# Inclusive vs. exclusive values

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

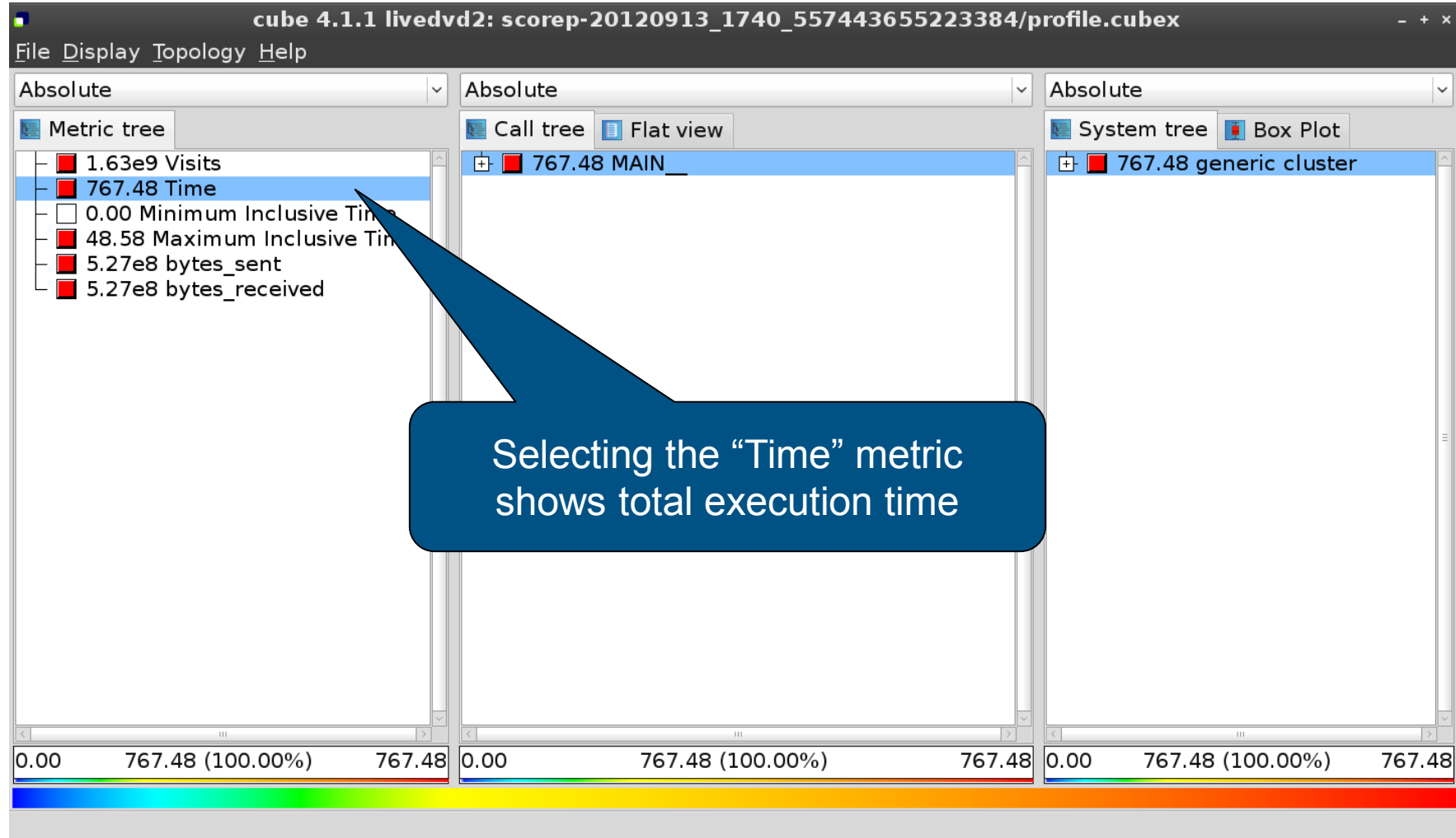




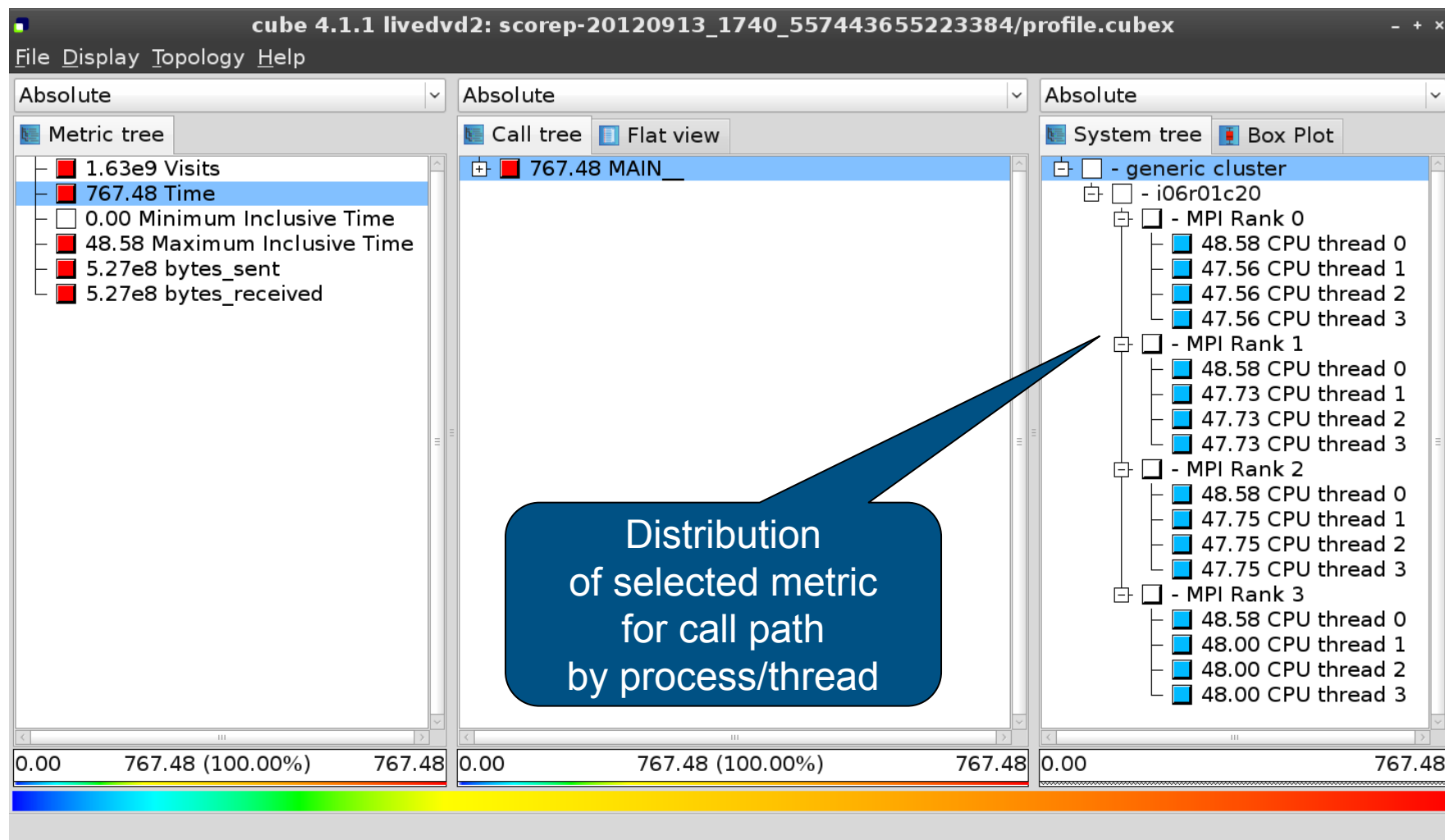
# Score-P analysis report exploration (opening view)



# Metric selection

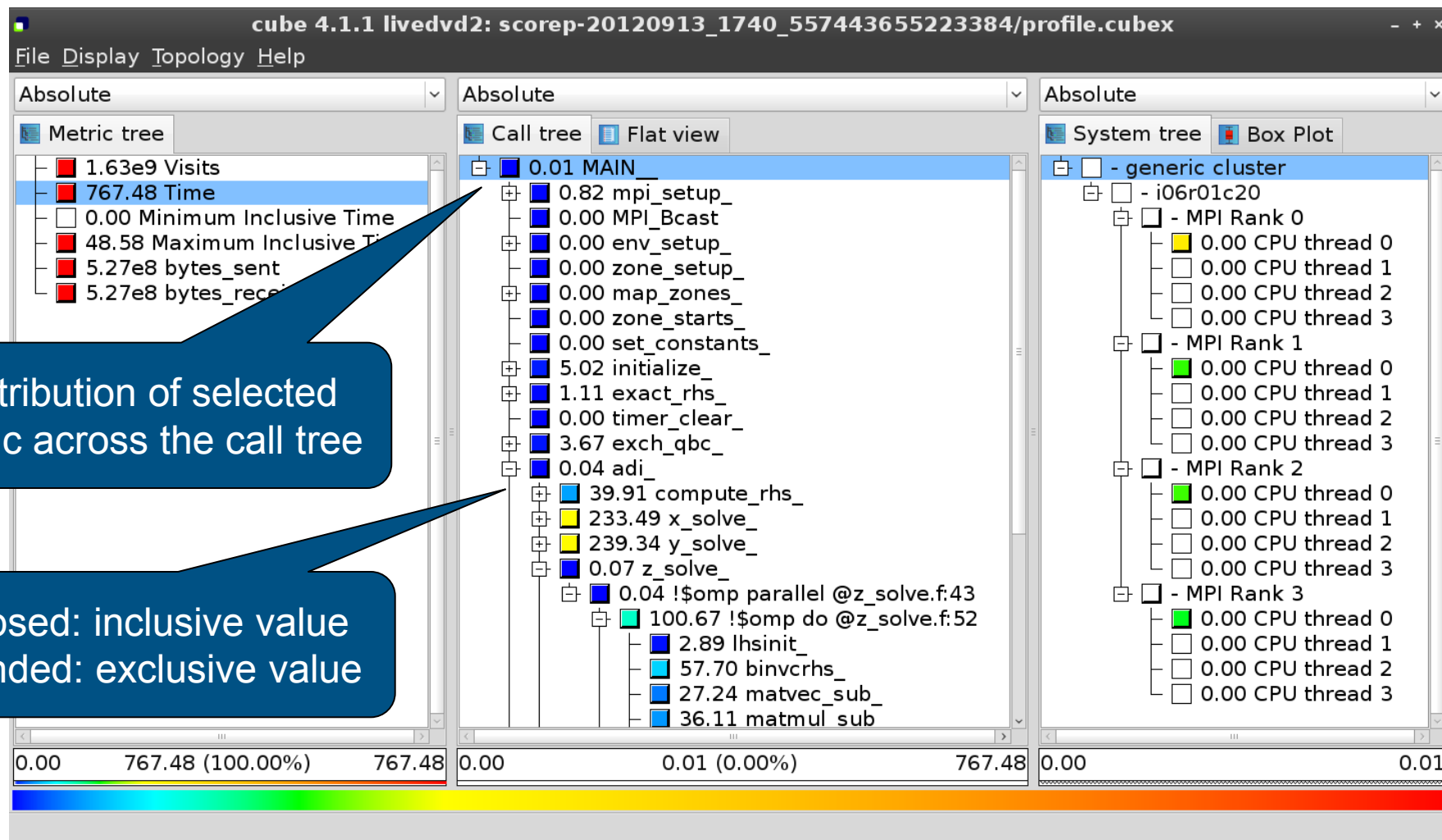


# Expanding the system tree

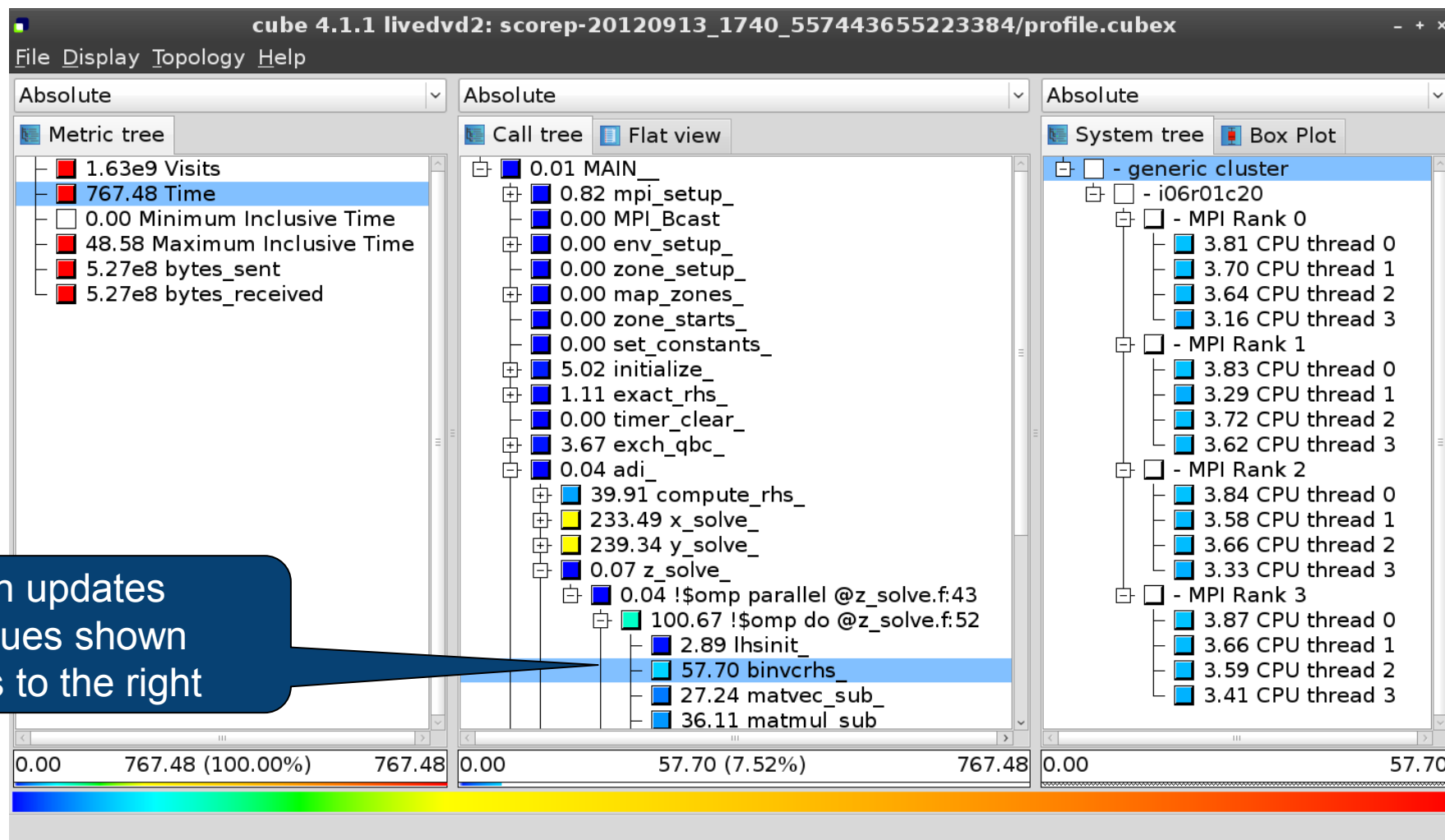




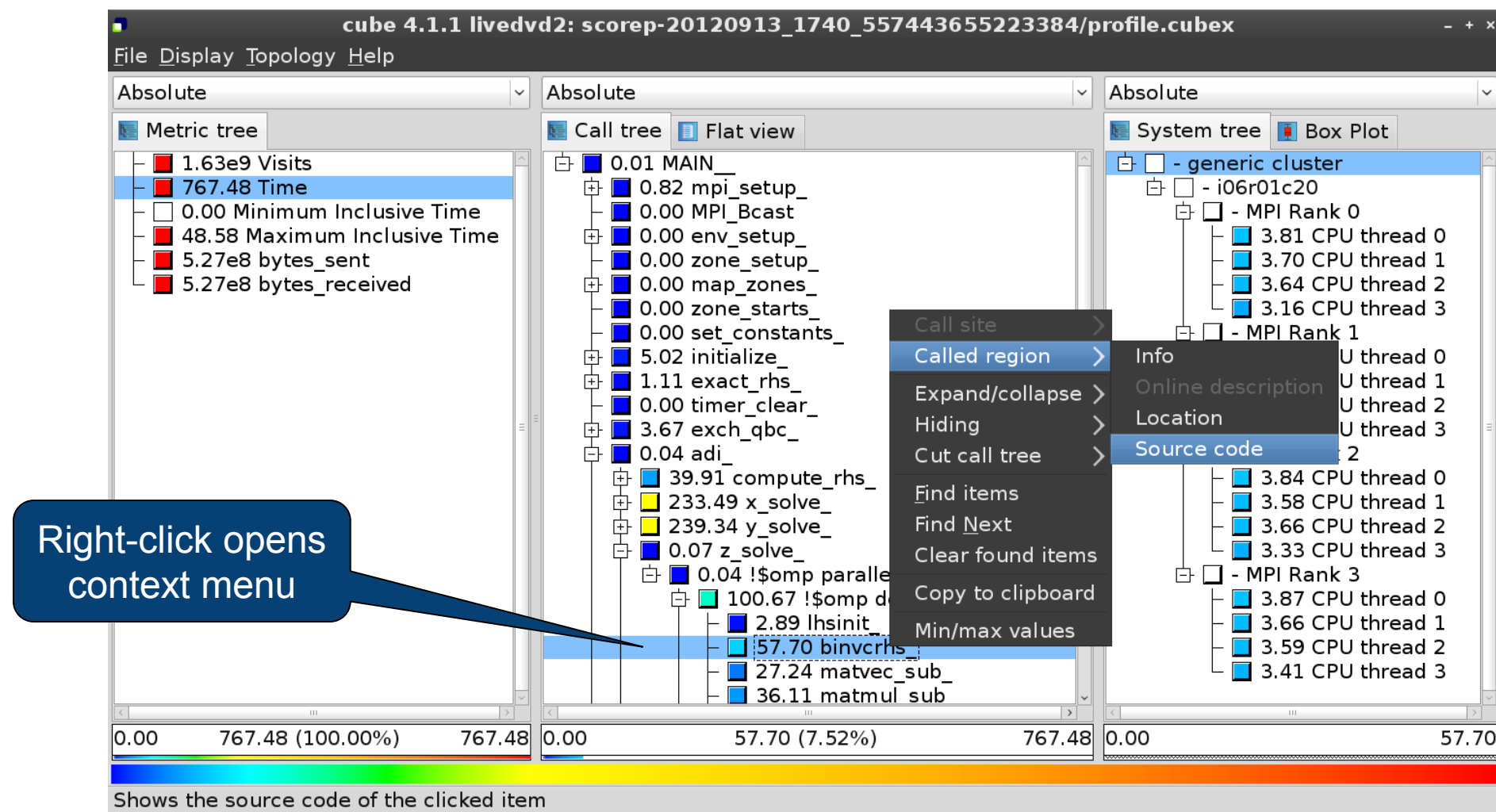
# Expanding the call tree



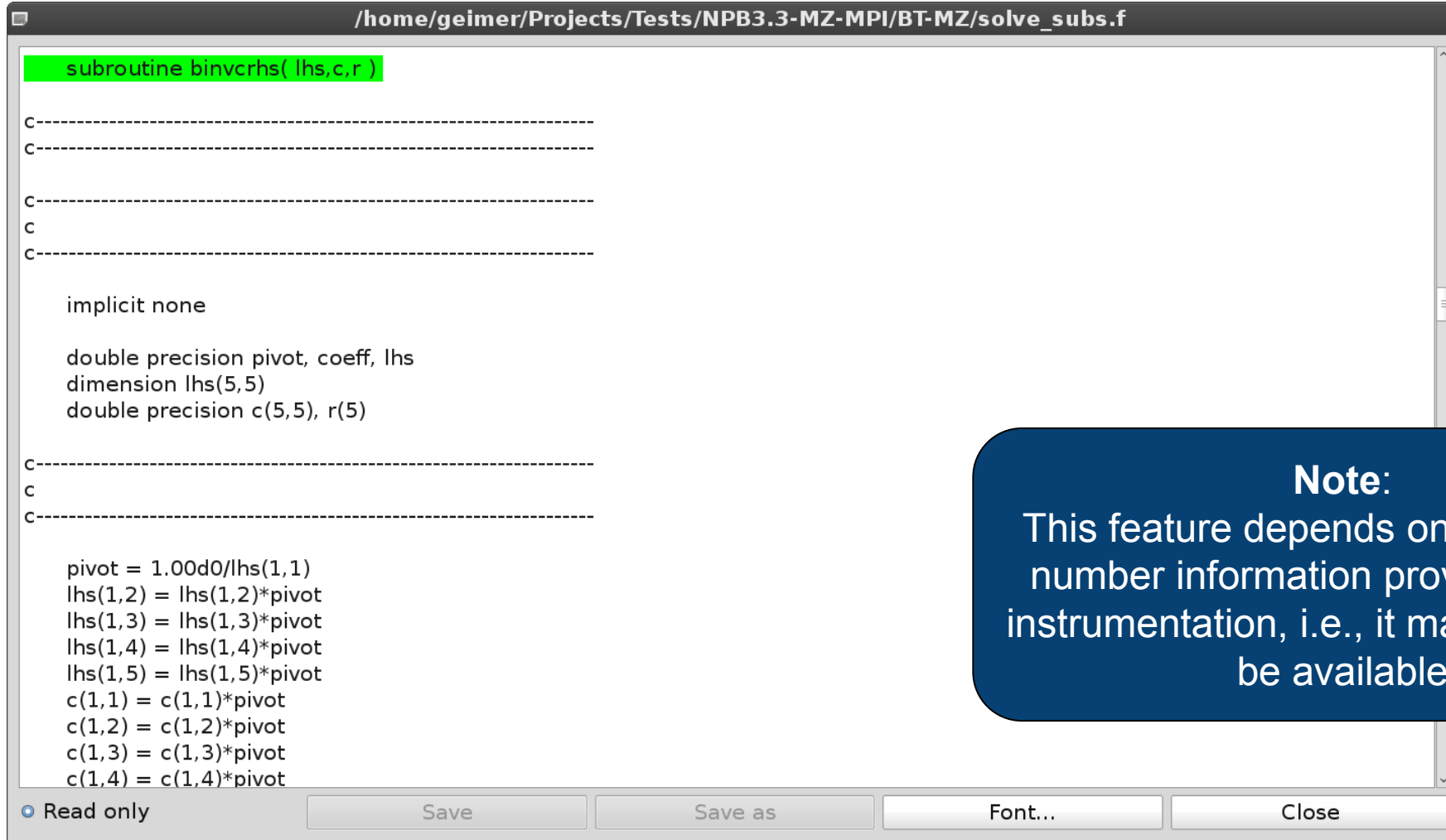
# Selecting a call path



# Source-code view via context menu



# Source-code view



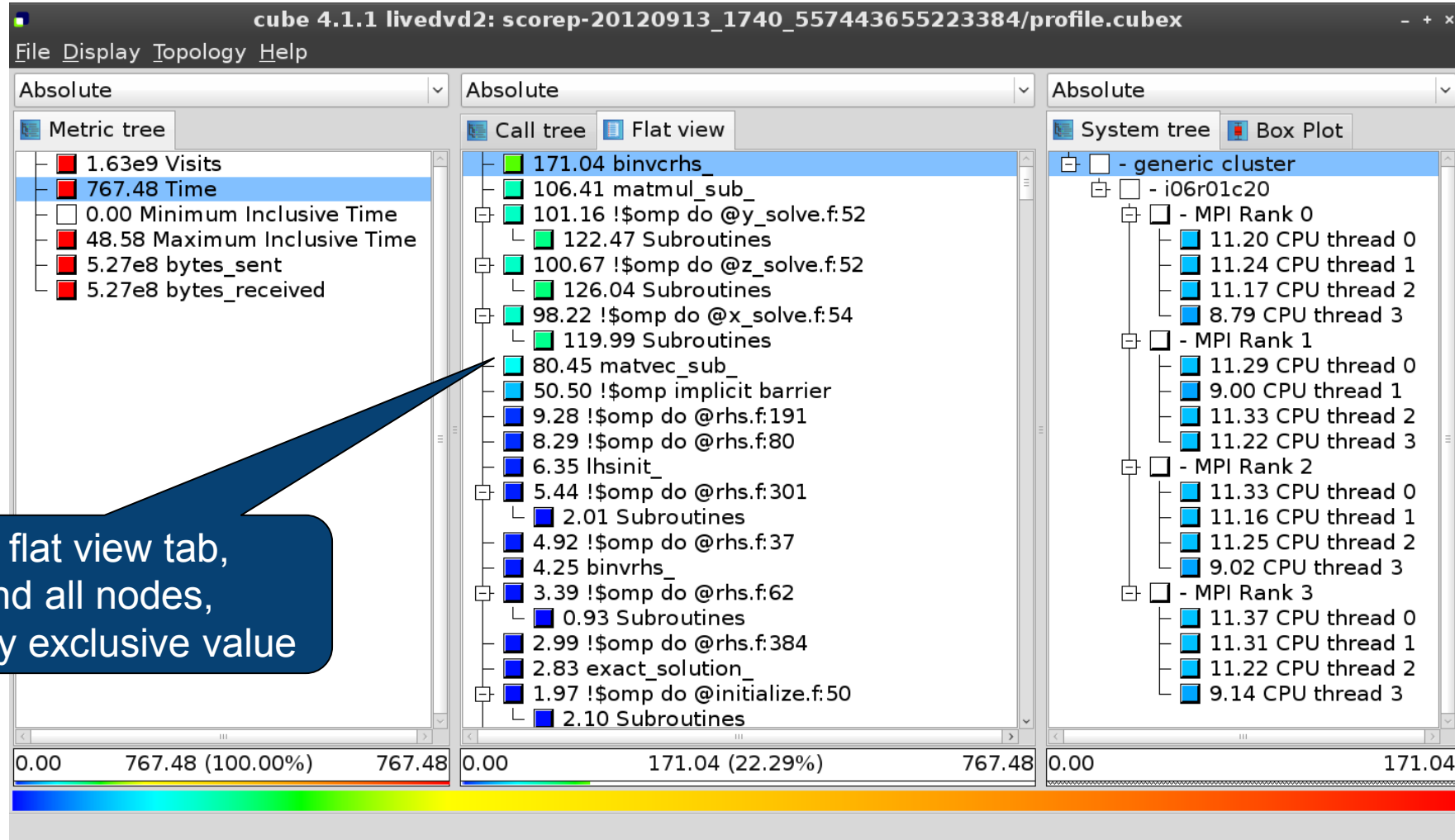
```
subroutine binvrhs( lhs,c,r )  
  
C-----  
C-----  
  
C-----  
C  
C-----  
  
implicit none  
  
double precision pivot, coeff, lhs  
dimension lhs(5,5)  
double precision c(5,5), r(5)  
  
C-----  
C  
C-----  
  
pivot = 1.00d0/lhs(1,1)  
lhs(1,2) = lhs(1,2)*pivot  
lhs(1,3) = lhs(1,3)*pivot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

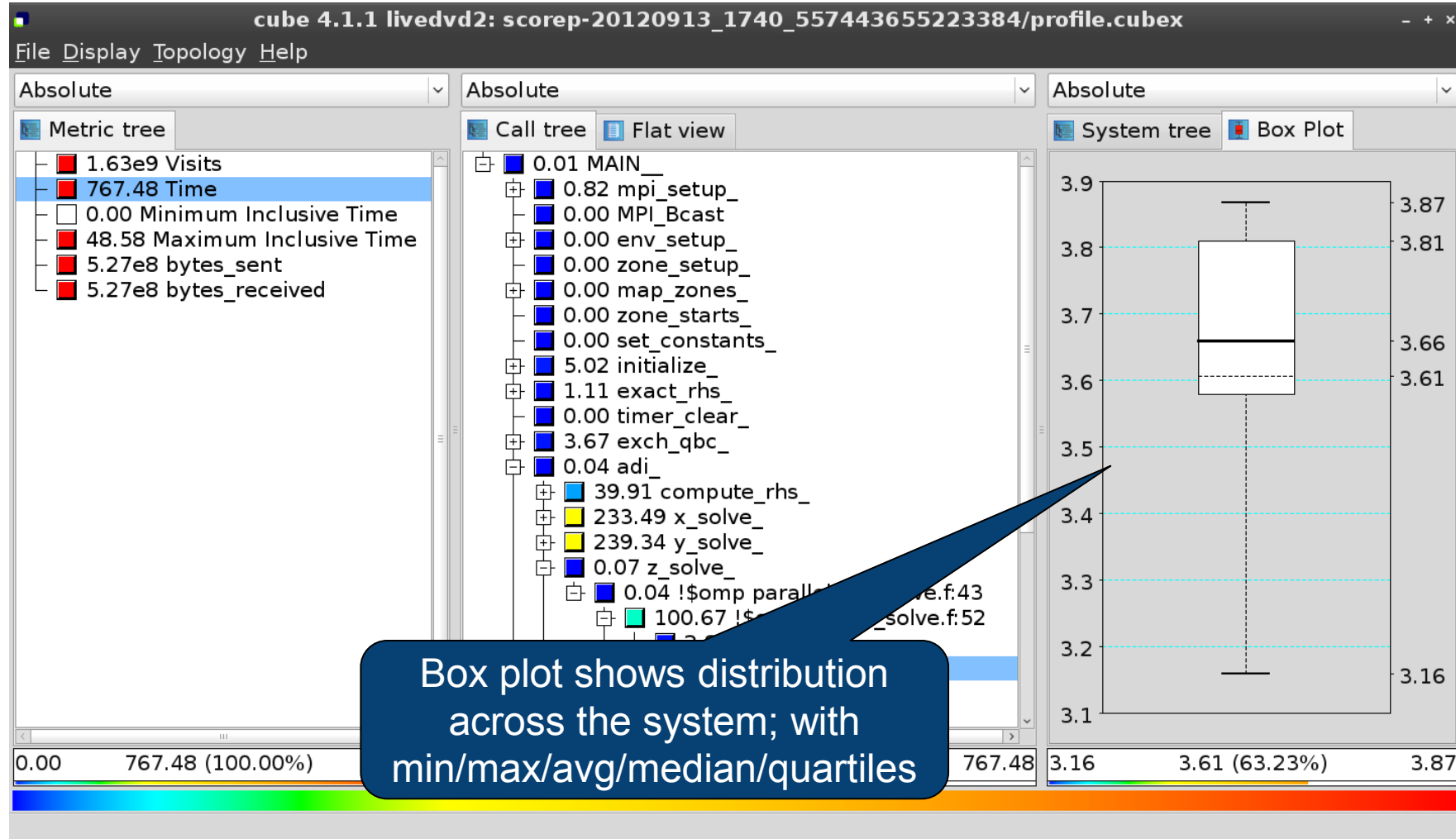
This feature depends on file and line number information provided by the instrumentation, i.e., it may not always be available

# Flat profile view

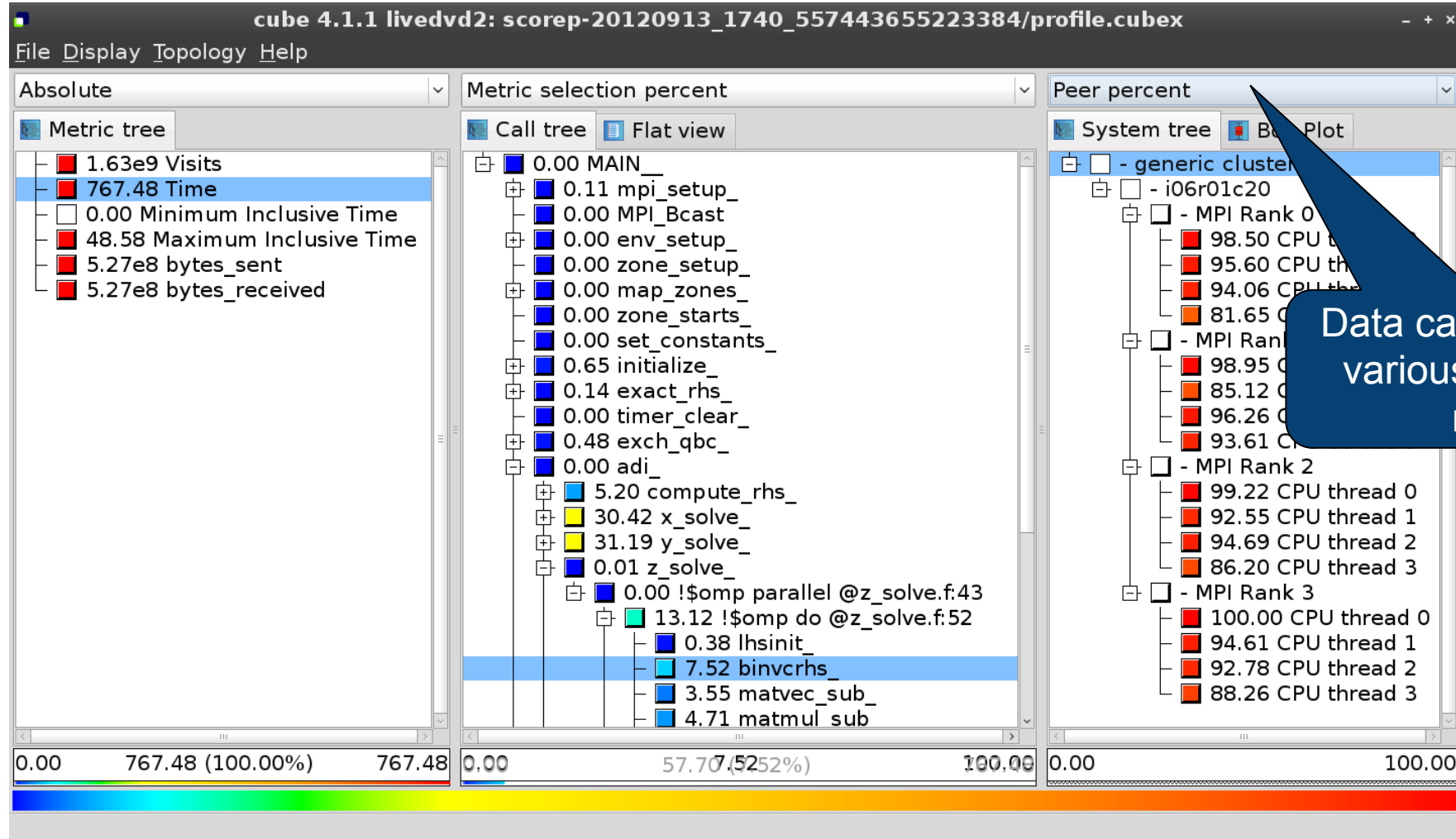




# Box plot view



# Alternative display modes

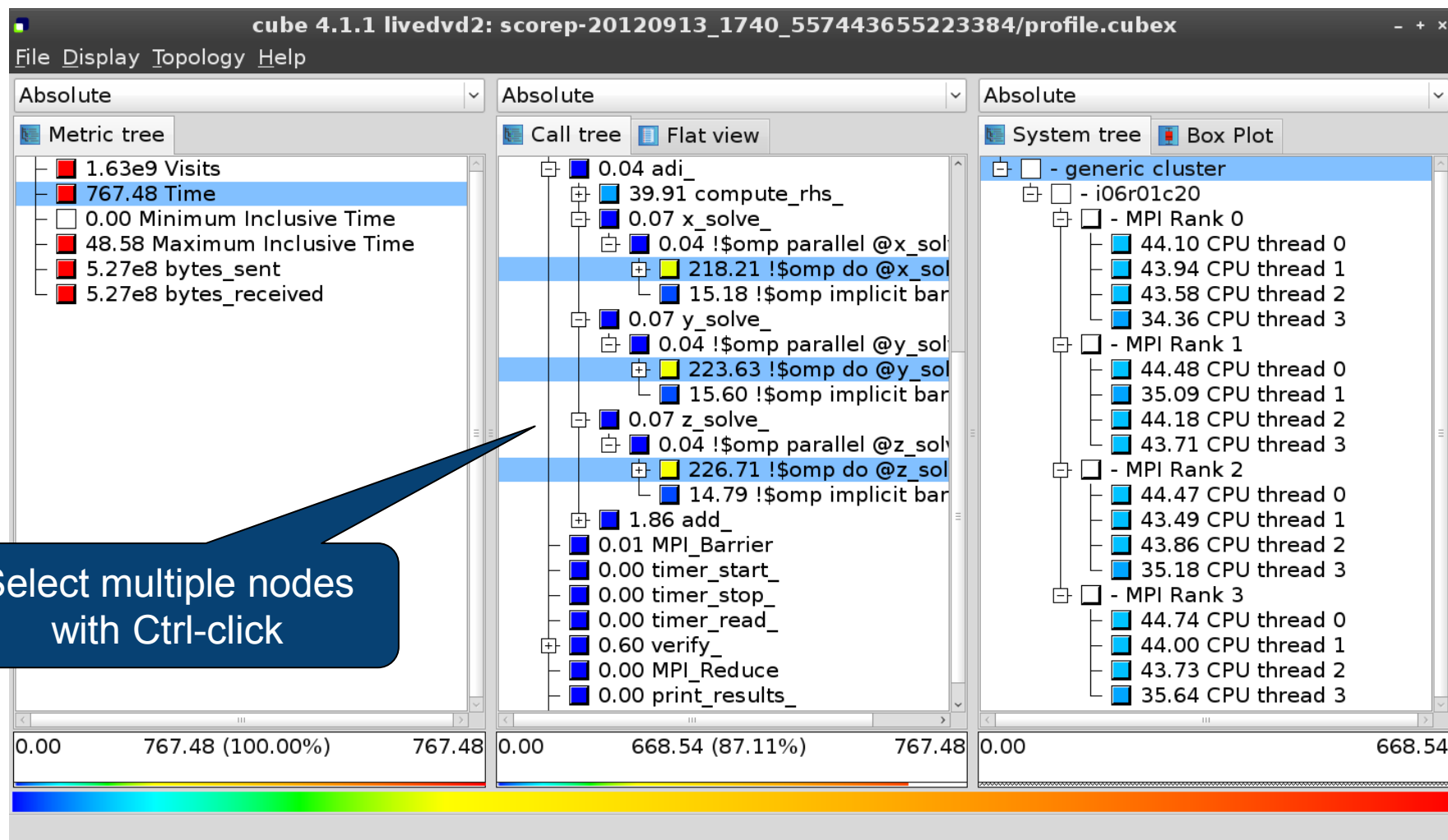


# Important display modes

---

- 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

# Multiple selection



# 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 mouse cursor and 'Shift+F1'), and 'About'. The 'Metric tree' on the left lists metrics like '1.63e9 Visits', '767.48 Time', and '0.00 Minimum I'. The central 'System tree' shows a hierarchical view of the application's execution, including 'compute\_rhs\_', 'solve\_', and 'verify\_'. The right panel shows a 'Box Plot' of CPU threads for MPI Ranks 0, 1, 2, and 3. A blue callout box points to the 'What's This?' menu item with the text 'Context-sensitive help available for all GUI items'. At the bottom, a status bar shows progress bars for three components: '0.00 767.48 (100.00%) 767.48', '0.00 668.54 (87.11%) 767.48', and '0.00 668.54'. A legend at the bottom indicates 'Change into help mode for display components'.

cube 4.1.1 livedvd2: scorep-20120913\_1740\_557443655223384/profile.cubex

File Display Topology Help

Absolute

Metric tree

- 1.63e9 Visits
- 767.48 Time
- 0.00 Minimum I
- 48.58 Maximum I
- 5.27e8 bytes
- 5.27e8 bytes received

Getting started

Mouse and keyboard control

What's This? Shift+F1

About

Selected metrics description

Selected regions description

compute\_rhs\_

solve\_

!\$omp parallel @x\_sol

218.21 !\$omp do @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

0.00 print\_results\_

System tree

Box Plot

- generic cluster

- i06r01c20
  - MPI Rank 0
    - 44.10 CPU thread 0
    - 43.94 CPU thread 1
    - 43.58 CPU thread 2
    - 34.36 CPU thread 3
  - MPI Rank 1
    - 44.48 CPU thread 0
    - 35.09 CPU thread 1
    - 44.18 CPU thread 2
    - 43.71 CPU thread 3
  - MPI Rank 2
    - 44.47 CPU thread 0
    - 43.49 CPU thread 1
    - 43.86 CPU thread 2
    - 35.18 CPU thread 3
  - MPI Rank 3
    - 44.74 CPU thread 0
    - 44.00 CPU thread 1
    - 43.73 CPU thread 2
    - 35.64 CPU thread 3

0.00 767.48 (100.00%) 767.48

0.00 668.54 (87.11%) 767.48

0.00 668.54

Change into help mode for display components

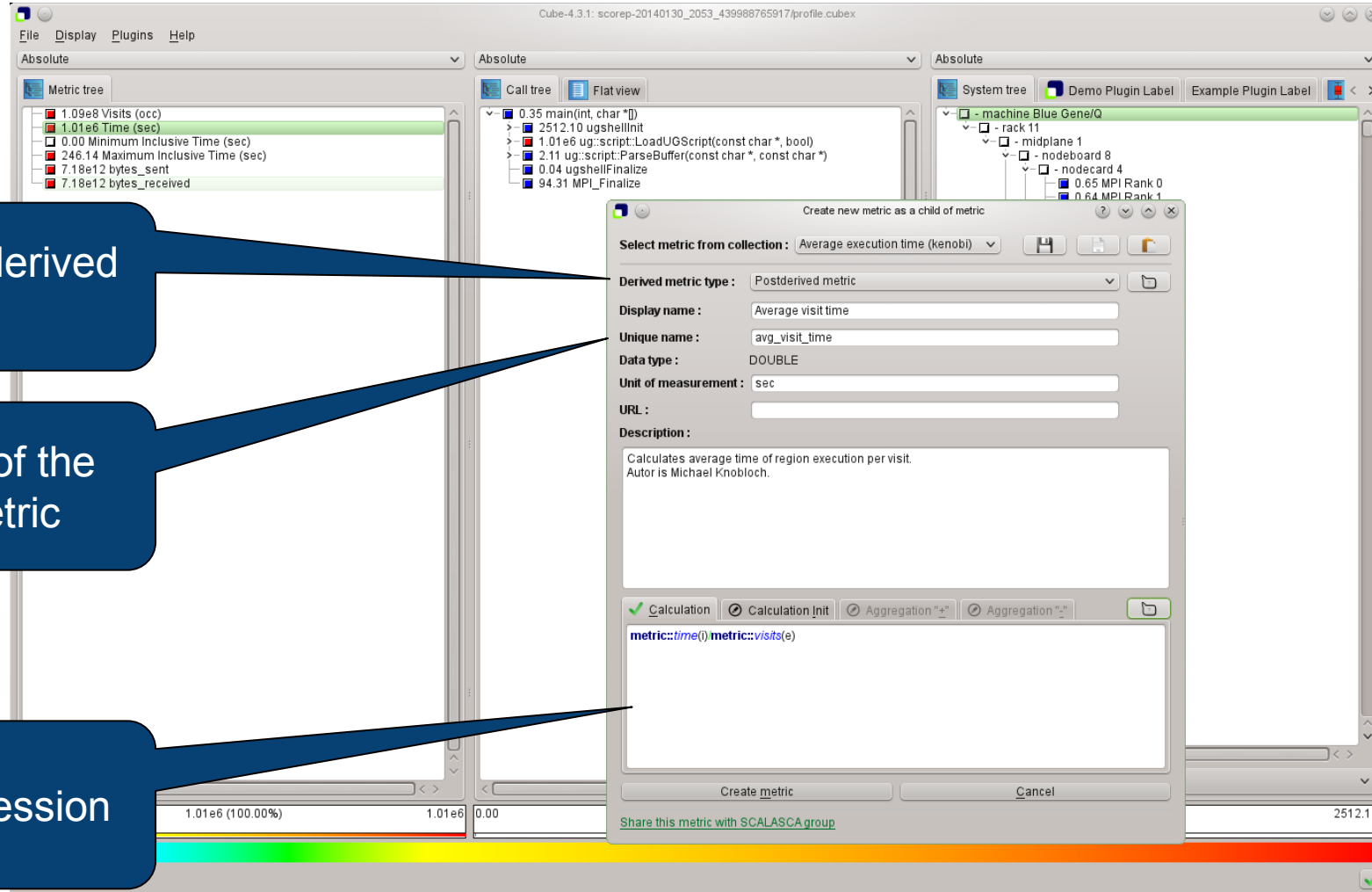


## Derived metrics

---

- 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:
  - Pre-derived: evaluation of the CubePL expression is performed before aggregation
  - Post-derived: evaluation of the CubePL expression is performed after aggregation
- Examples:
  - “Average execution time”: Post-derived metric with expression  
**`metric::time(i)/metric::visits(e)`**
  - “Number of FLOP per second”: Post-derived metric with expression  
**`metric::FLOP()/metric::time()`**

# Derived metrics in Cube GUI



Collection of derived metrics

Parameters of the derived metric

CubePL expression

# 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 panes:

- Left Pane (Edit metric FLOPS):** This pane allows for configuring a derived metric. The 'Select metric from collection' dropdown is set to '--- please select ---'. The 'Derived metric type' is 'Postderived metric'. The 'Display name' is 'FLOPS', the 'Unique name' is 'flops', and the 'Data type' is 'DOUBLE'. The 'Unit of measurement' is empty. The 'URL' is empty. The 'Description' field is empty. At the bottom, there are buttons for 'Calculation', 'Calculation Init', 'Aggregation "+"', and 'Aggregation "-"'. The 'Calculation' button is selected. Below these buttons, the formula `metric::PAPI_FP_OPS()/metric::time()` is entered. At the very bottom, there are buttons for 'Edit metric' and 'Cancel'.
- Metric tree (Absolute):** This pane shows a list of metrics. The 'FLOPS' metric is highlighted in blue. The list includes: 1.17e7 Visits (occ), 1148.49 Time (sec), 0.00 Minimum Inclusive Time (sec), 41.57 Maximum Inclusive Time (...), 0 bytes\_put (bytes), 0 bytes\_get (bytes), 5.75e12 PAPI\_TOT\_INS (#), 2.69e12 PAPI\_TOT\_CYC (#), 2.12e12 PAPI\_FP\_OPS (#), 3.12e9 bytes\_sent (bytes), 3.12e9 bytes\_received (bytes), and 1.84e9 FLOPS.
- Call tree (Absolute):** This pane shows a hierarchical view of the metrics. The 'MAIN' metric is highlighted in blue. The tree includes: 3.17e5 MAIN, 7.04e5 mpi\_setup\_, 6.34e4 MPI\_Bcast, 2.05e5 env\_setup\_, 7.39e5 zone\_setup\_, 9.31e5 map\_zones\_, 9.39e4 zone\_starts\_, 6.16e5 set\_constants\_, 5.91e8 initialize\_, 0.00 exact\_rhs\_, 145.62 !\$omp parallel @exac..., 2.54e4 !\$omp do @exact\_r..., 9.65e8 !\$omp do @exact\_r..., 9.62e8 !\$omp do @exact\_r..., 8.14e8 !\$omp do @exact\_r..., 1.21e5 !\$omp do @exact\_r..., 0.00 !\$omp implicit barrier..., 6.23e4 exch\_qbc\_, 1.94e9 adi\_, 2.19e5 MPI\_Barrier, 1.92e9 <<bt\_iter>> (200 itera..., 1.98e8 verify\_, and 1.05e5 MPI\_Reduce.
- System tree (Absolute):** This pane shows a hierarchical view of the system. The 'machine Linux' metric is highlighted in blue. The tree includes: - machine Linux, - node frog6, - MPI Rank 0, 1.17e9 Master thread, 9.43e8 OMP thread 1, 9.47e8 OMP thread 2, 9.47e8 OMP thread 3, - MPI Rank 1, 1.17e9 Master thread, 9.87e8 OMP thread 1, 9.68e8 OMP thread 2, 9.72e8 OMP thread 3, - MPI Rank 2, 1.10e9 Master thread, 8.97e8 OMP thread 1, 8.77e8 OMP thread 2, 8.76e8 OMP thread 3, - MPI Rank 3, 1.09e9 Master thread, 9.06e8 OMP thread 1, 9.04e8 OMP thread 2, and 9.02e8 OMP thread 3.

At the bottom of the interface, there is a status bar showing the selected metric: 'Selected !\$omp do @exact\_rhs.f:46'.

# 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

## Cube: Further information

---

- Parallel program analysis report exploration tools
  - Libraries for Cube 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](mailto:scalasca@fz-juelich.de)

