

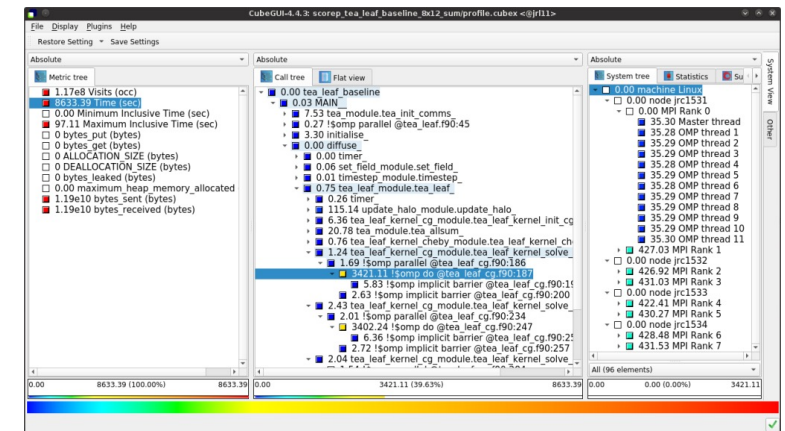
VI-HPS

VIRTUAL INSTITUTE – HIGH PRODUCTIVITY SUPERCOMPUTING

Analysis report examination with Cube



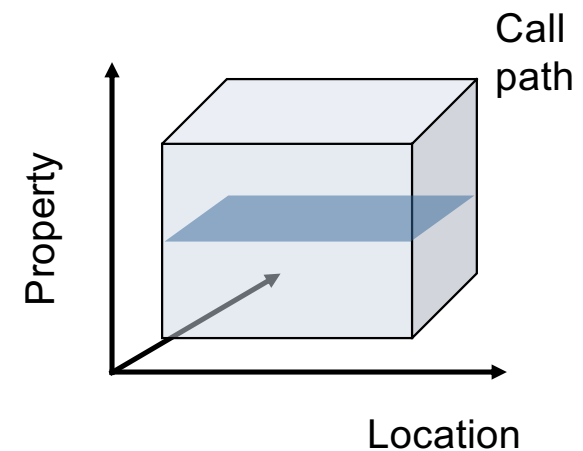
- Parallel program analysis report exploration tools
 - Libraries for XML+binary report reading & writing
 - Algebra utilities for report processing
 - GUI for interactive analysis exploration
 - Requires Qt ≥ 5
- Originally developed as part of the Scalasca toolset
- Now available as separate components
 - Can be installed independently of Score-P and Scalasca, e.g., on laptop/desktop
 - Latest releases: Cube v4.8.2 (Sep 2023)



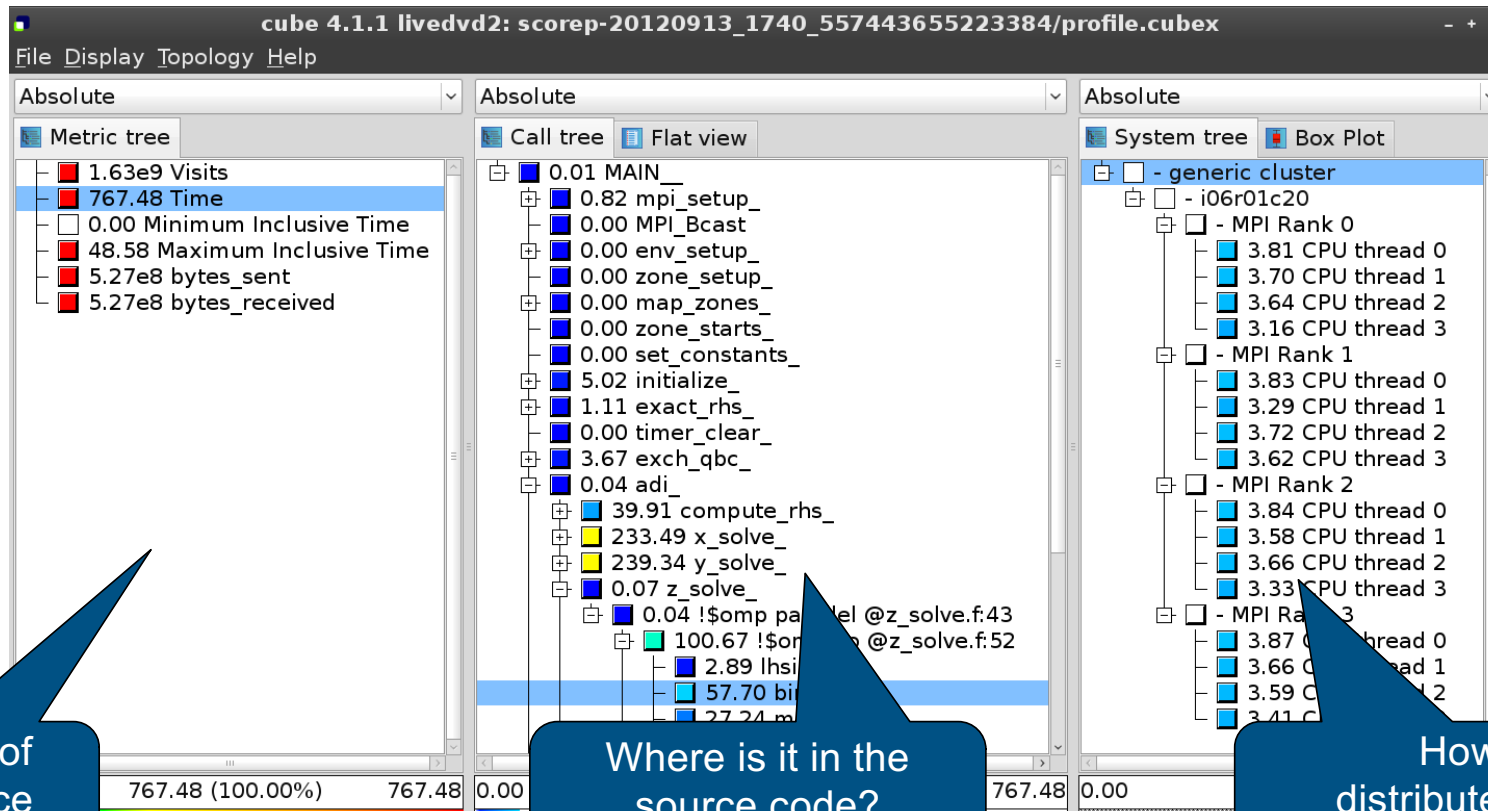
Note: source distribution tarballs for Linux, as well as binary packages provided for Linux, Windows & MacOS, from www.scalasca.org website in Software/Cube 4.x

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

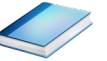


What kind of performance metric?

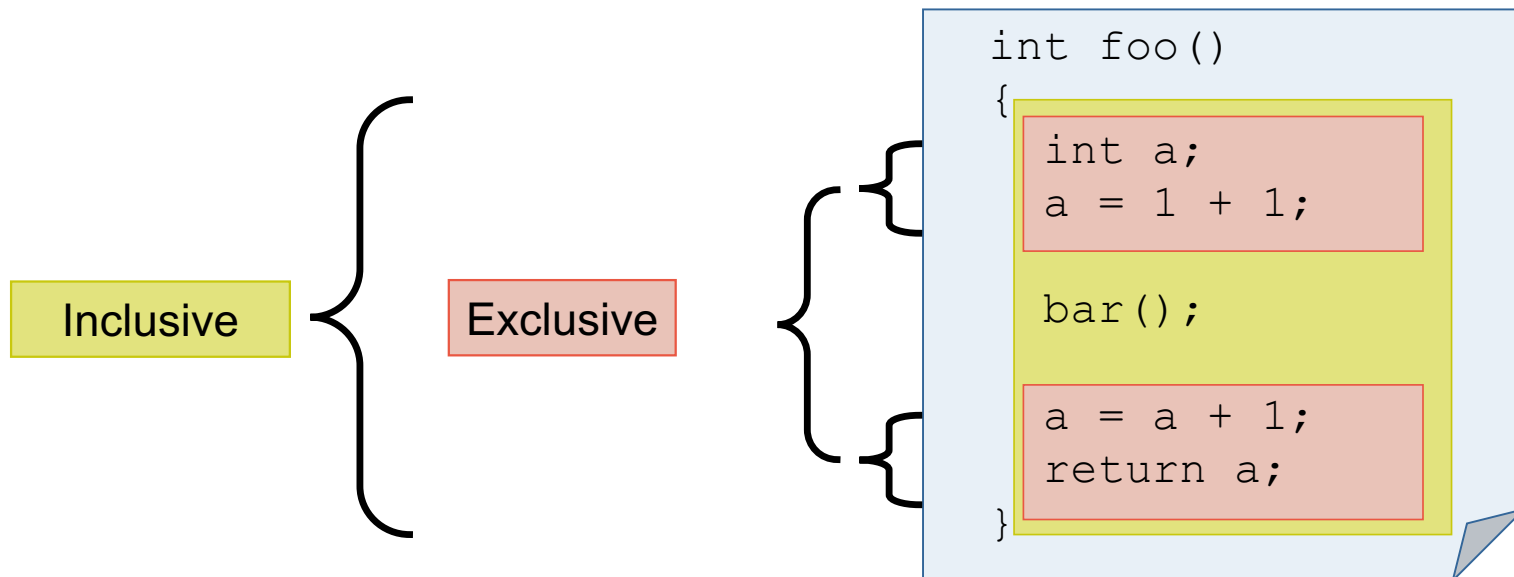
Where is it in the source code? In what context?

How is it distributed across the processes/threads?

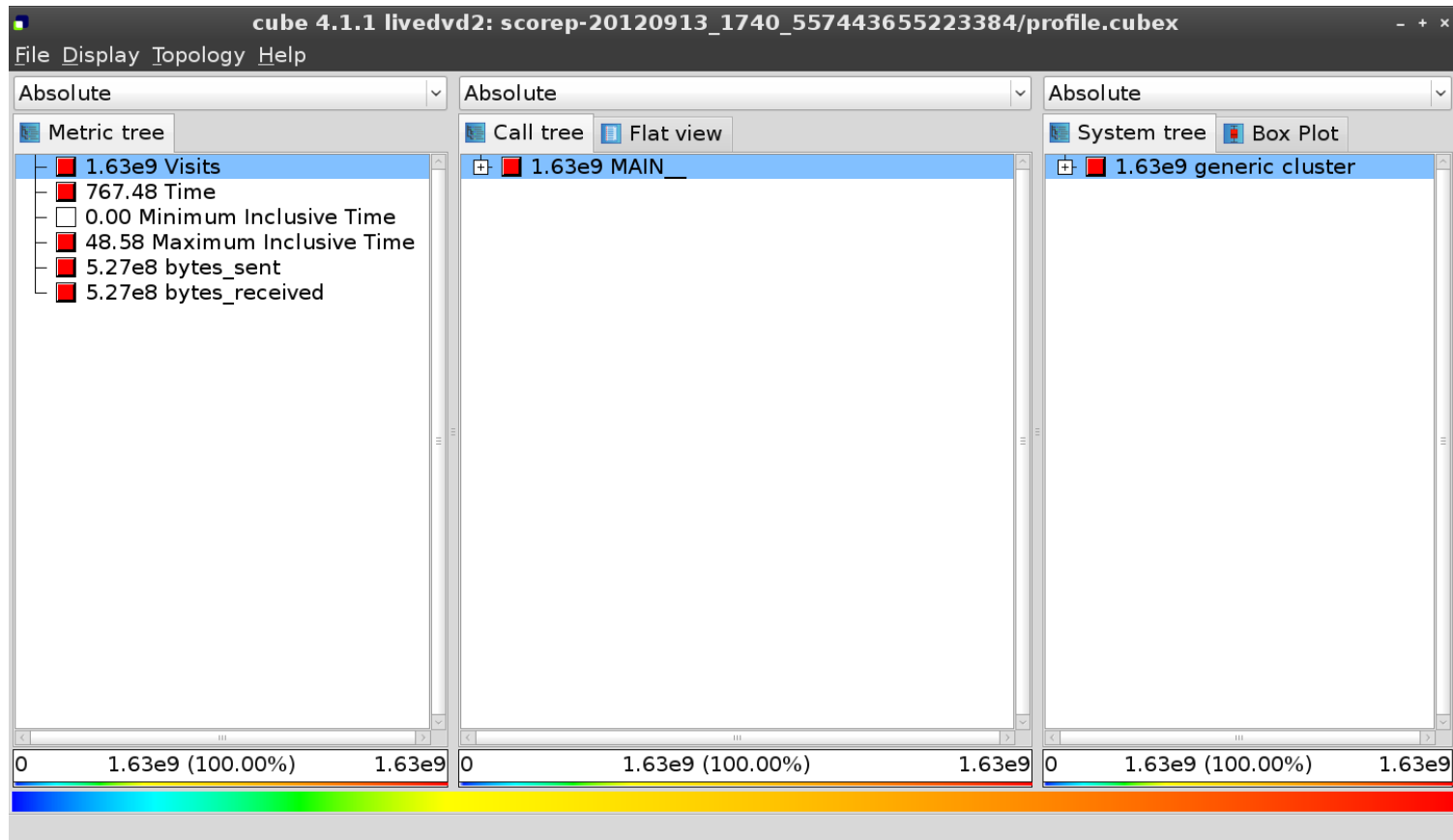
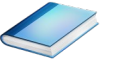
Inclusive vs. exclusive values



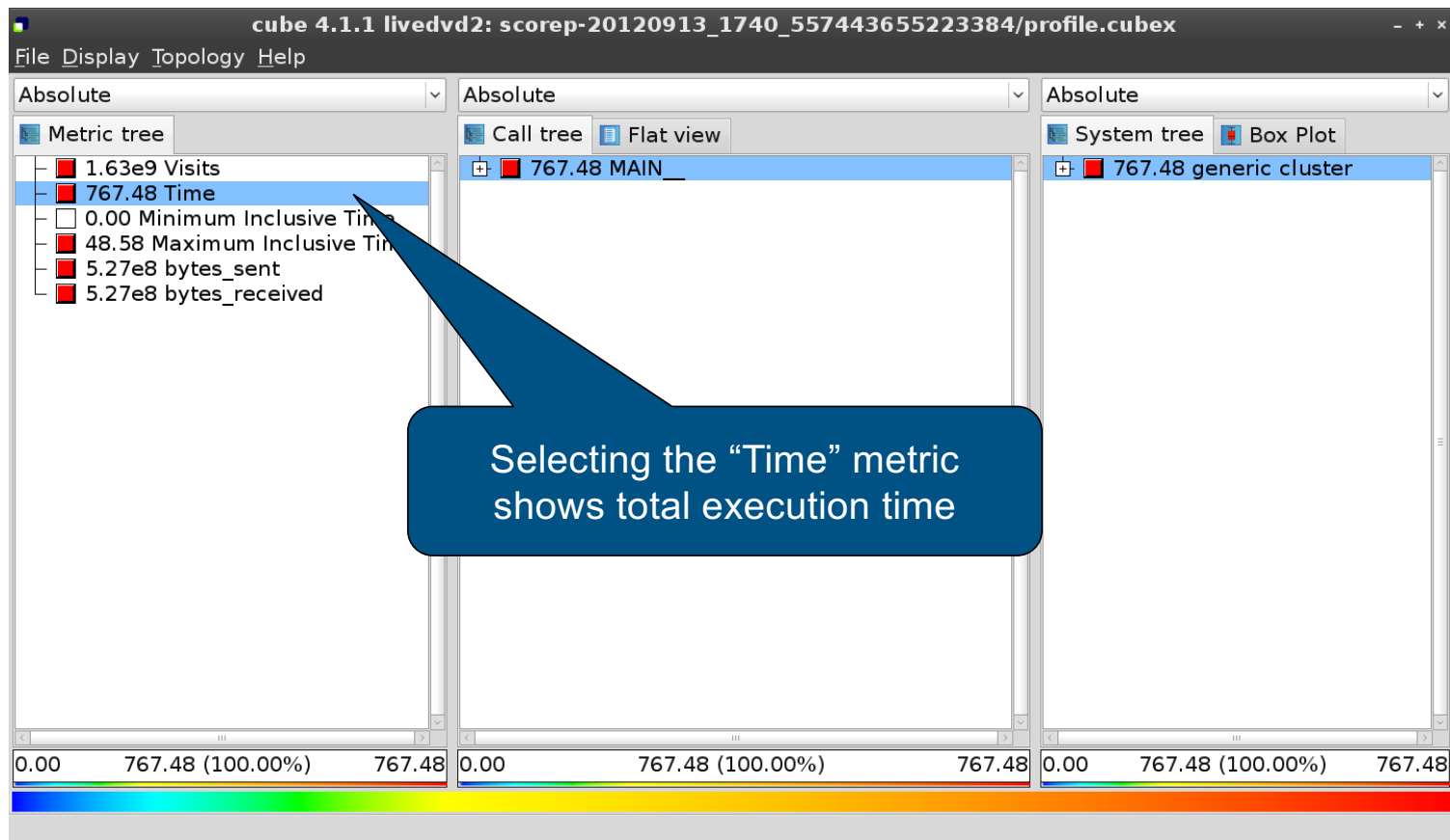
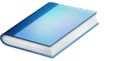
- Inclusive
 - Information of all sub-elements aggregated into single value
- Exclusive
 - Information cannot be subdivided further



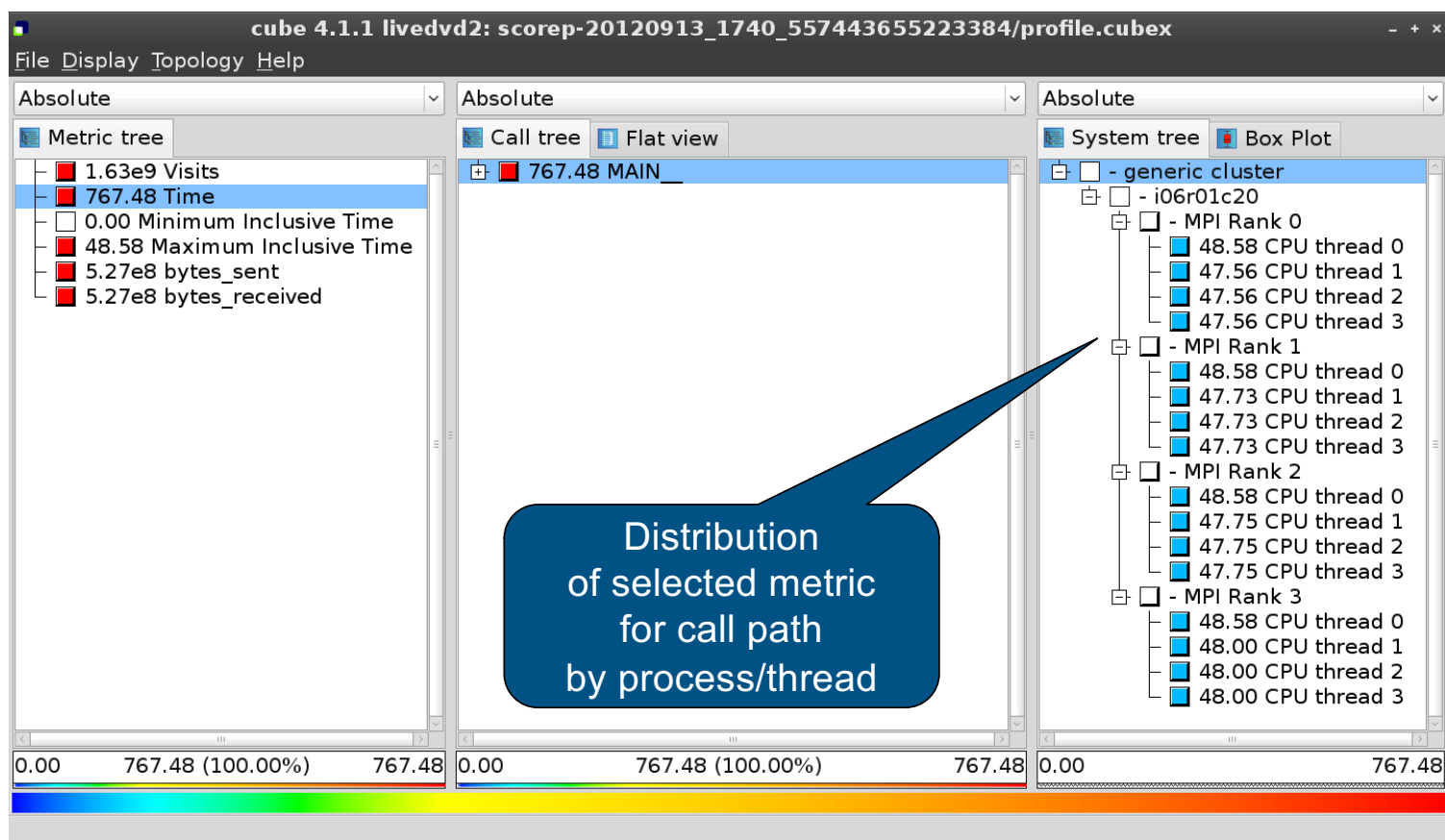
Score-P analysis report exploration (opening view)



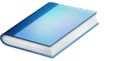
Metric selection



Expanding the system tree

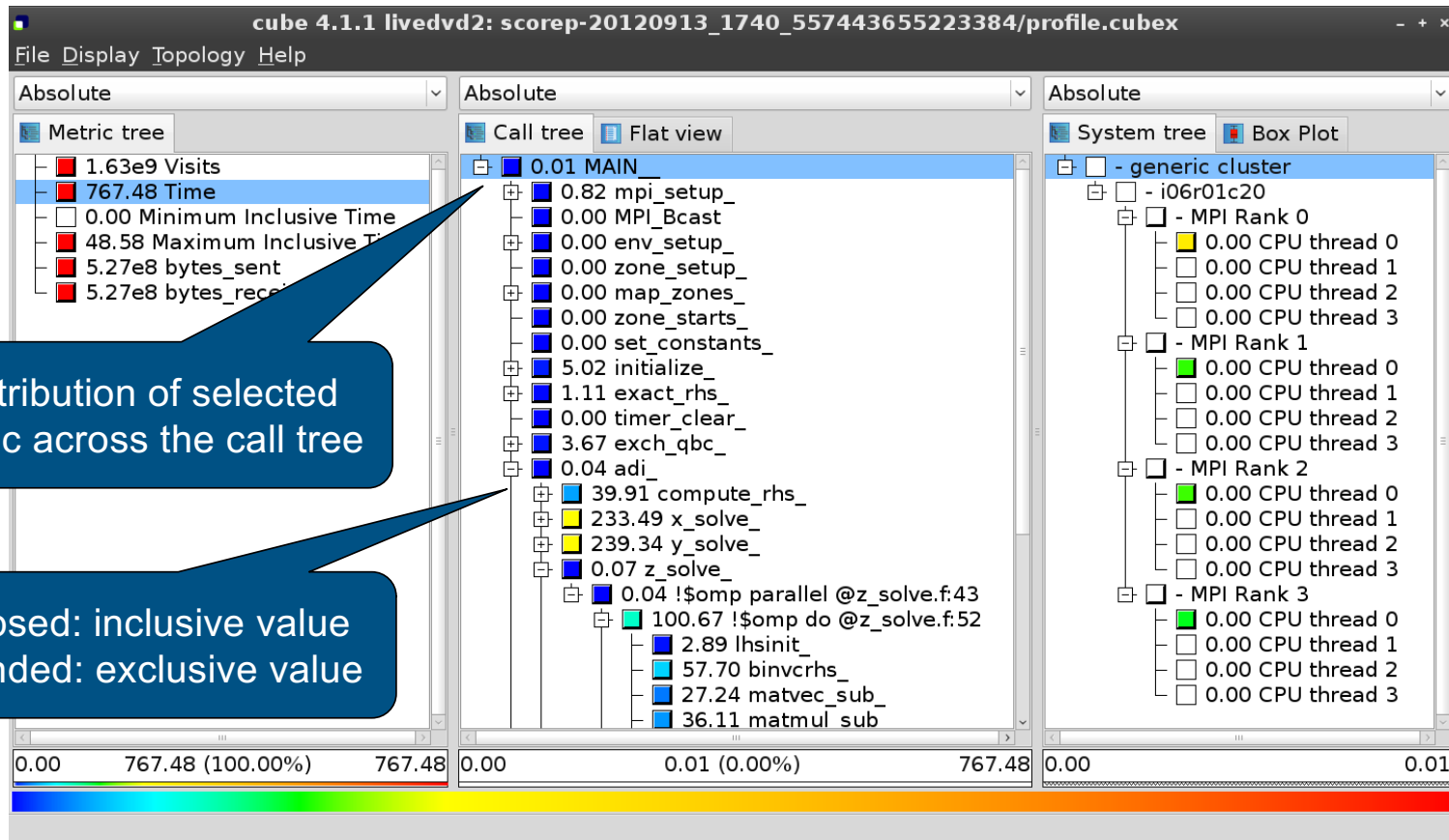


Expanding the call tree

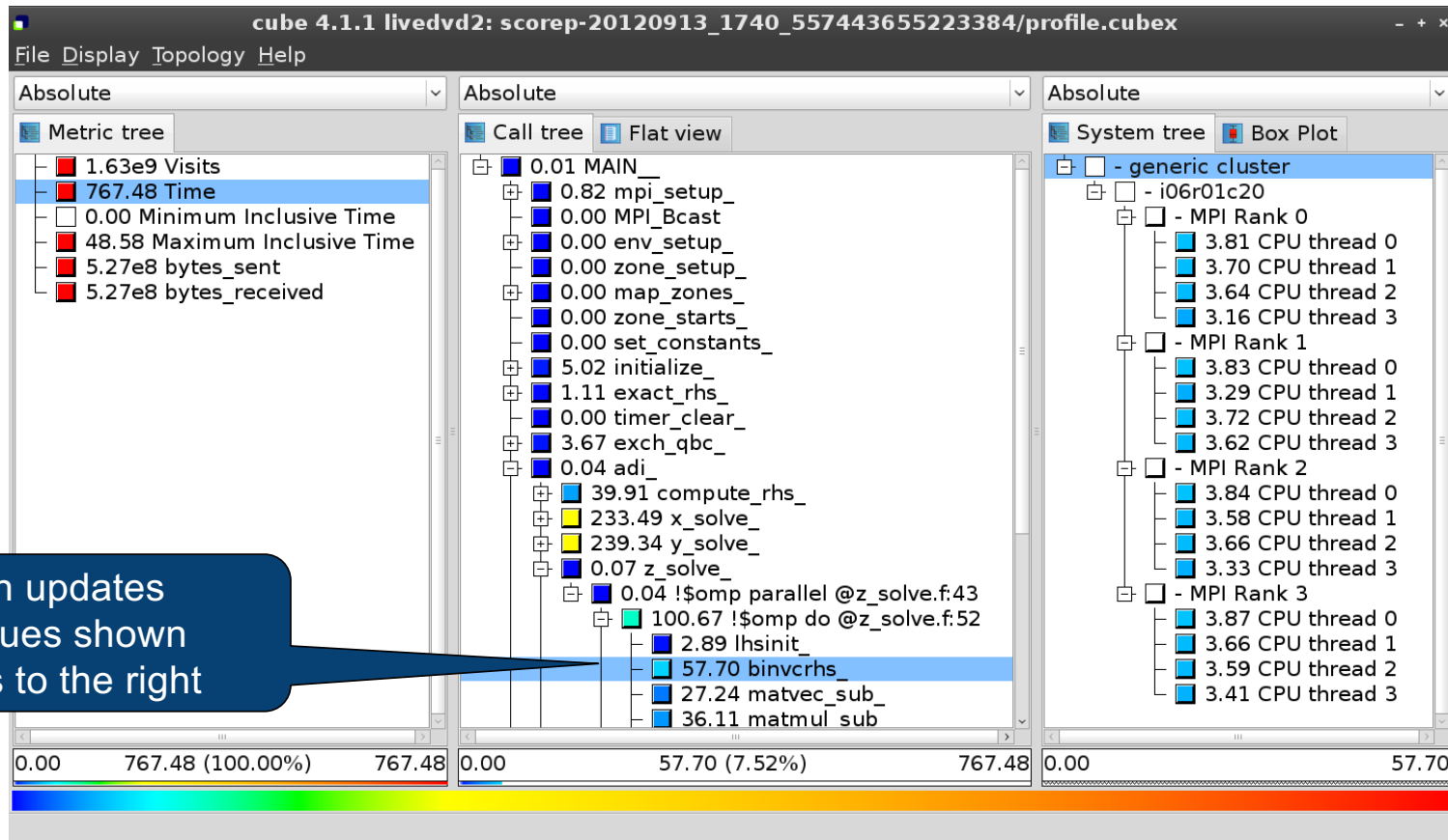
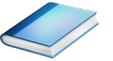


Distribution of selected metric across the call tree

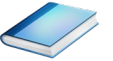
Collapsed: inclusive value
Expanded: exclusive value



Selecting a call path



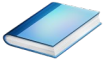
Source-code view via context menu



The screenshot shows the 'cube 4.1.1' interface with three main panels: 'Metric tree', 'Call tree', and 'System tree'. A context menu is open over the '57.70 binvcrhs' node in the call tree. The menu options include 'Call site', 'Called region', 'Expand/collapse', 'Hiding', 'Cut call tree', 'Find items', 'Find Next', 'Clear found items', 'Copy to clipboard', and 'Min/max values'. The 'Source code' option is highlighted. Below the call tree, a source code view is visible, showing the code for the selected function. A blue callout box points to the context menu with the text 'Right-click opens context menu'. At the bottom, a status bar shows 'Shows the source code of the clicked item'.

Metric	Value	Percentage	Parent Value
767.48 Time	767.48	100.00%	767.48
57.70 binvcrhs	57.70	7.52%	767.48

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)

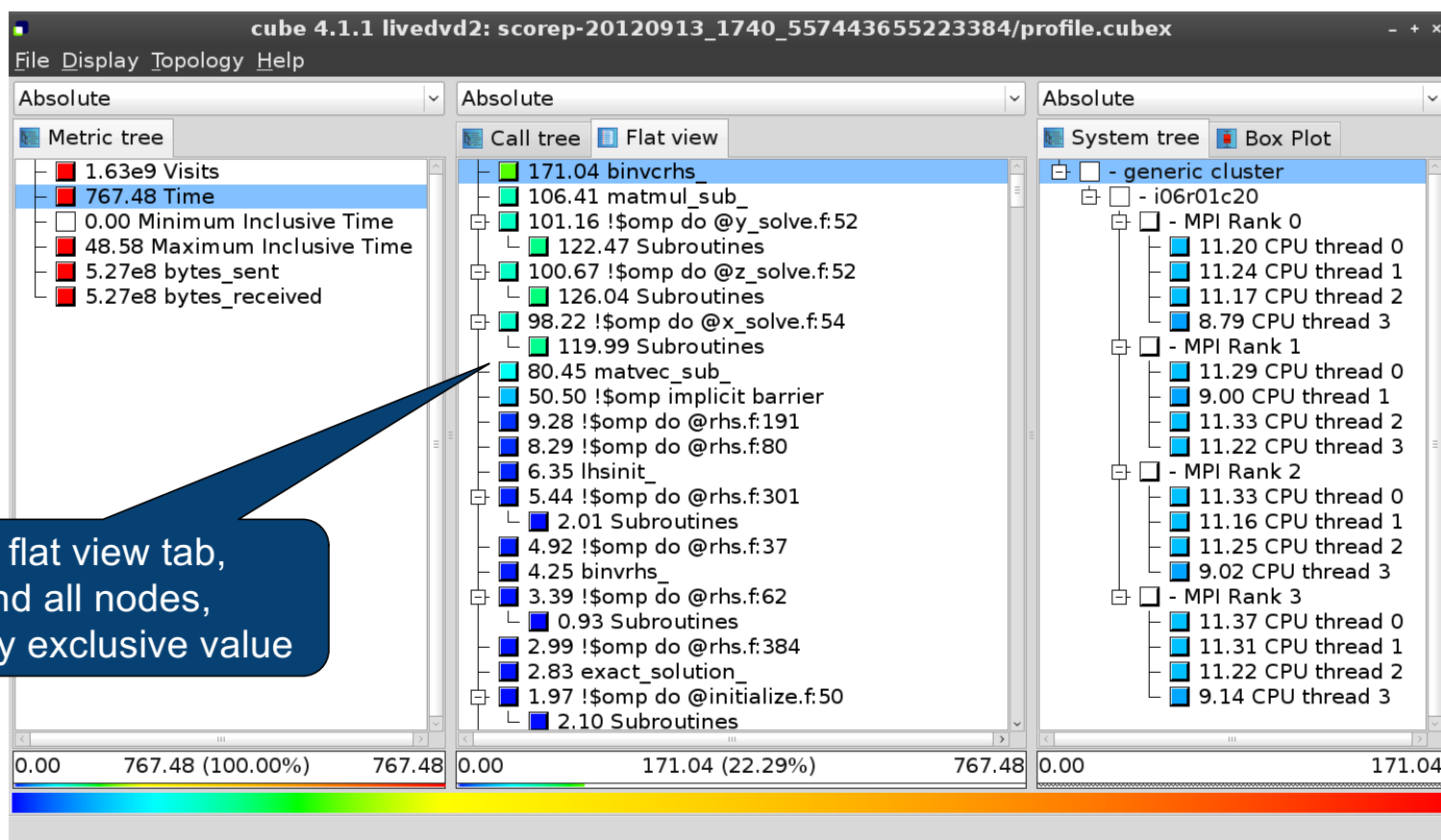
C-----
C
C-----

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
```

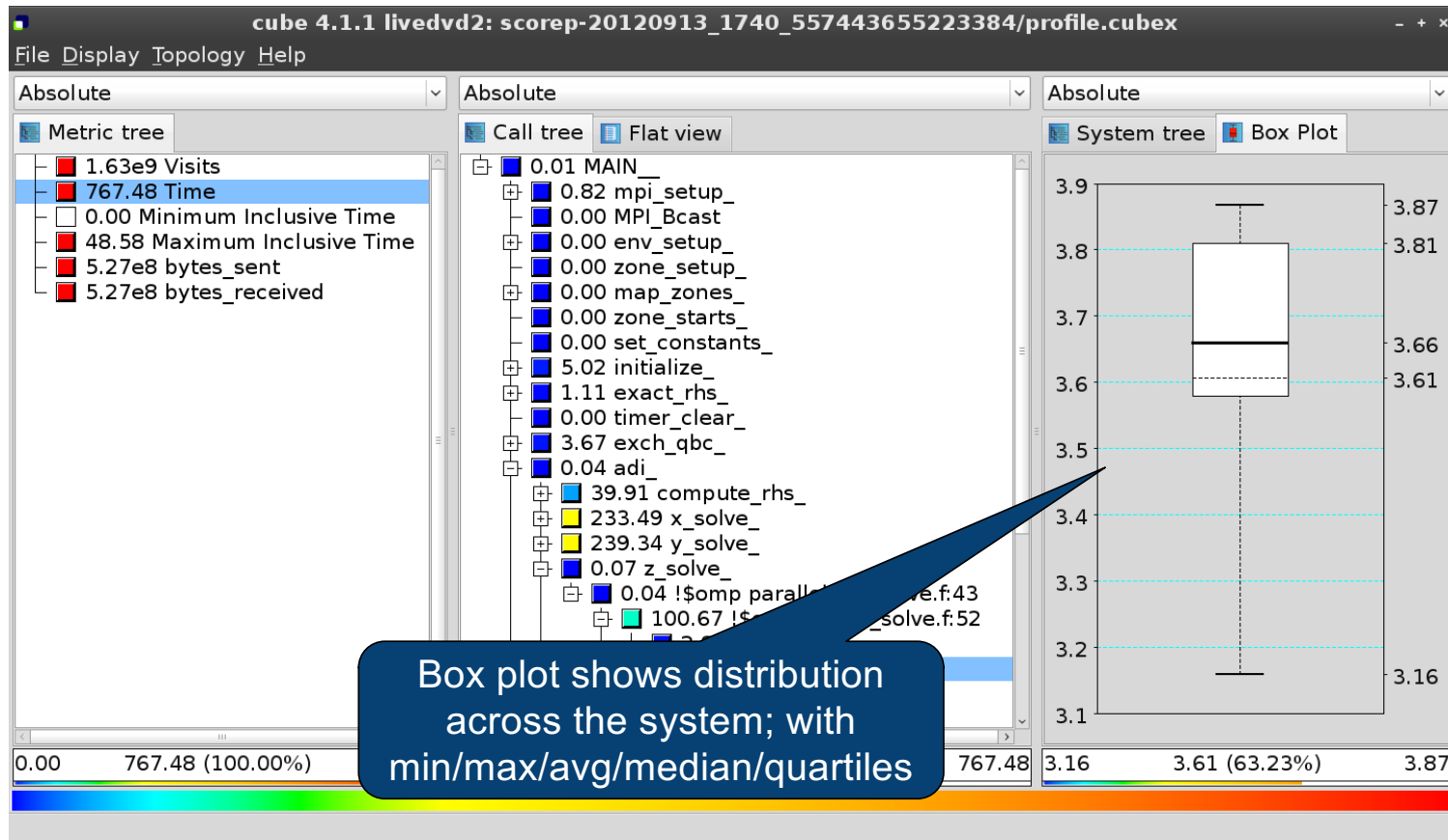
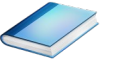
Read only Save Save as Font... Close

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

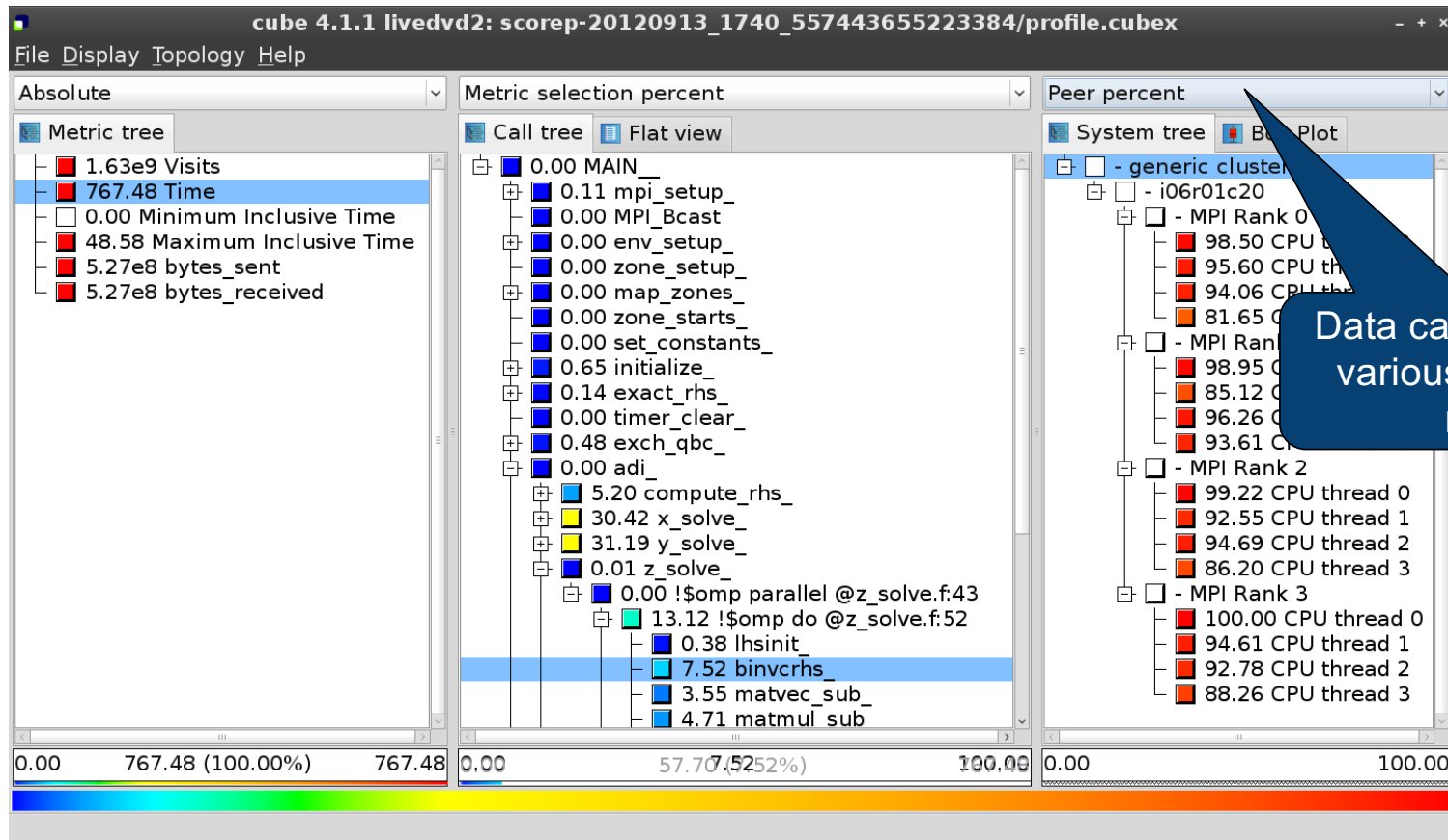
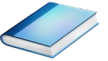
Flat profile view



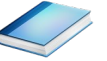
Box plot view



Alternative display modes

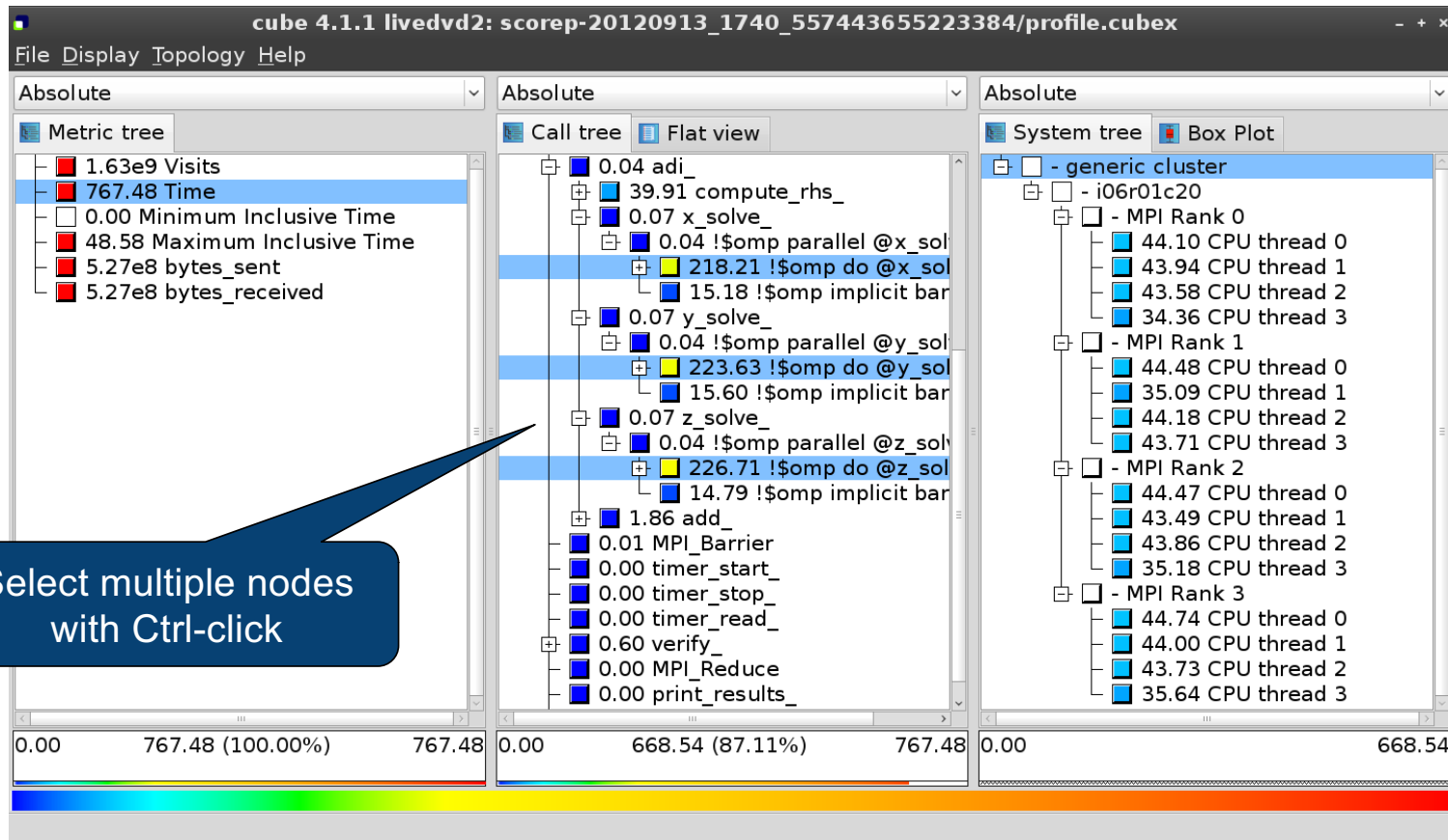
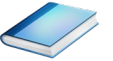


Important display modes



- 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



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
- 5.27e8 byte
- 5.27e8

Getting started
 Mouse and keyboard control
What's This? (Shift+F1)
 About
 Selected metrics description
 Selected regions description

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_
 0.00 print_results_

Absolute

System tree

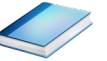
- generic cluster
 - i06r01c20
 - MPI Rank 0
 - 44.10 CPU thread 0
 - 43.94 CPU thread 1
 - 43.58 CPU thread 2
 - 34.36 CPU thread 3
 - MPI Rank 1
 - 44.48 CPU thread 0
 - 35.09 CPU thread 1
 - 44.18 CPU thread 2
 - 43.71 CPU thread 3
 - MPI Rank 2
 - 44.47 CPU thread 0
 - 43.49 CPU thread 1
 - 43.86 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

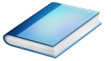
Context-sensitive help available for all GUI items

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



Collection of derived metrics

Parameters of the derived metric

CubePL expression

metric::time()metric::visits(e)

Example: FLOPS based on PAPI_FP_OPS and time



The screenshot displays the Cube-4.3.1 performance analysis tool interface, showing the configuration of a derived metric and its breakdown in three views: Metric tree, Call tree, and System tree.

Left Panel: Edit metric FLOPS (on froggy1)

- Select metric from collection: --- please select ---
- Derived metric type: Postderived metric
- Display name: FLOPS
- Unique name: flops
- Data type: DOUBLE
- Unit of measurement: [Empty]
- URL: [Empty]
- Description: [Empty]
- Calculation: `metric::PAPI_FP_OPS()/metric::time()`

Metric tree (Absolute):

- 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)
- 1.84e9 FLOPS**

Call tree (Absolute):

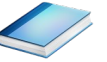
- 3.17e5 MAIN_
 - 7.04e5 mpi_setup_
 - 6.34e4 MPI_Bcast
 - 2.05e5 env_setup_
 - 7.39e5 zone_setup_
 - 9.31e5 map_zones_
 - 9.39e4 zone_starts_
 - 6.16e5 set_constants_
 - 5.91e8 initialize_
 - 0.00 exact_rhs_
 - 145.62 !\$omp parallel @exac...
 - 2.54e4 !\$omp do @exact_r...
 - 9.65e8 !\$omp do @exact_r...**
 - 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_
 - 1.05e5 MPI_Reduce

System tree (Absolute):

 - 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
 - 9.02e8 OMP thread 3

Selected !\$omp do @exact_rhs.f:46"

Iteration profiling

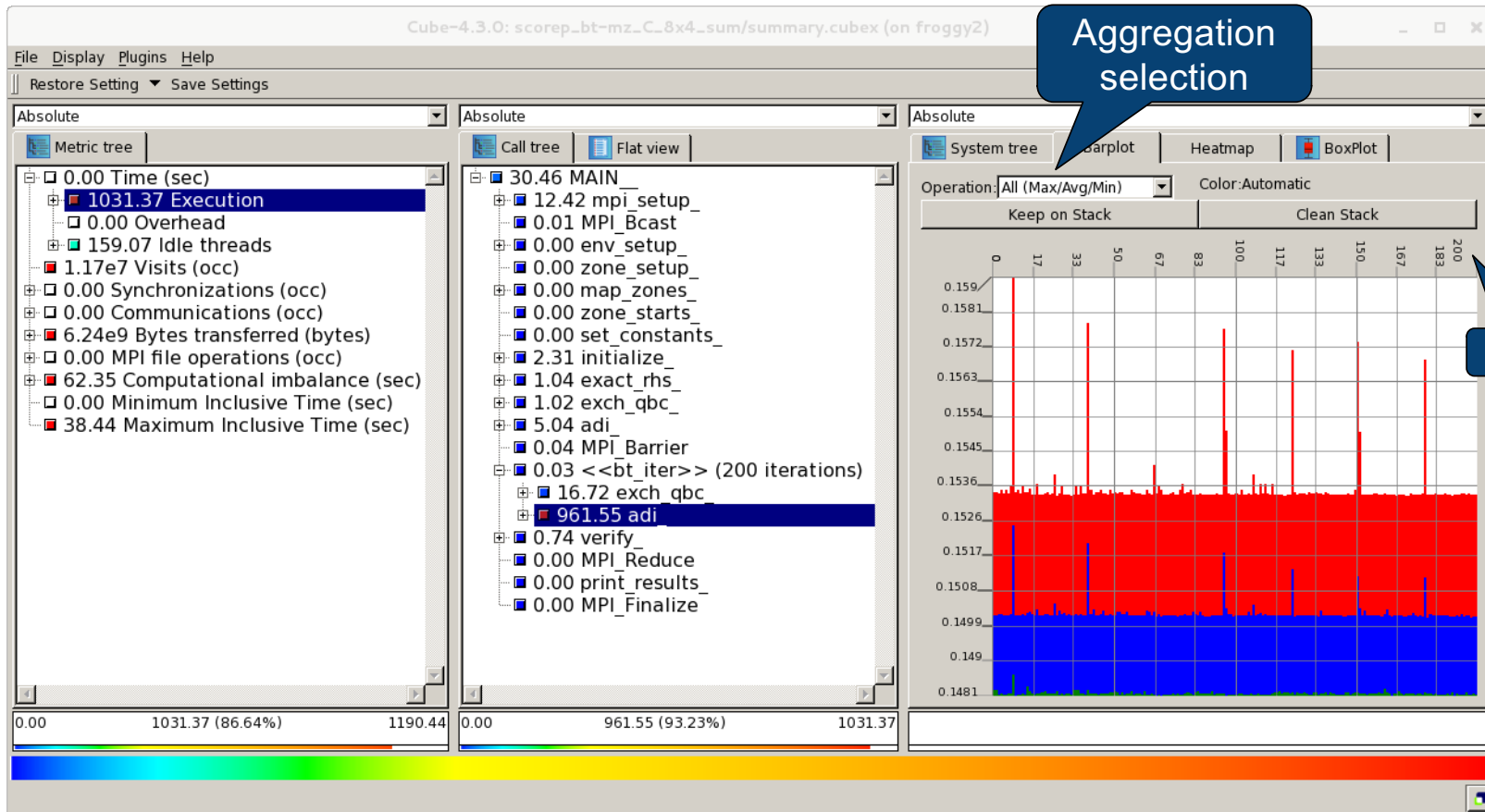
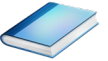


- Show time dependent behavior by “unrolling” iterations
- Preparations:
 - Mark loop body by using Score-P instrumentation API in your source code

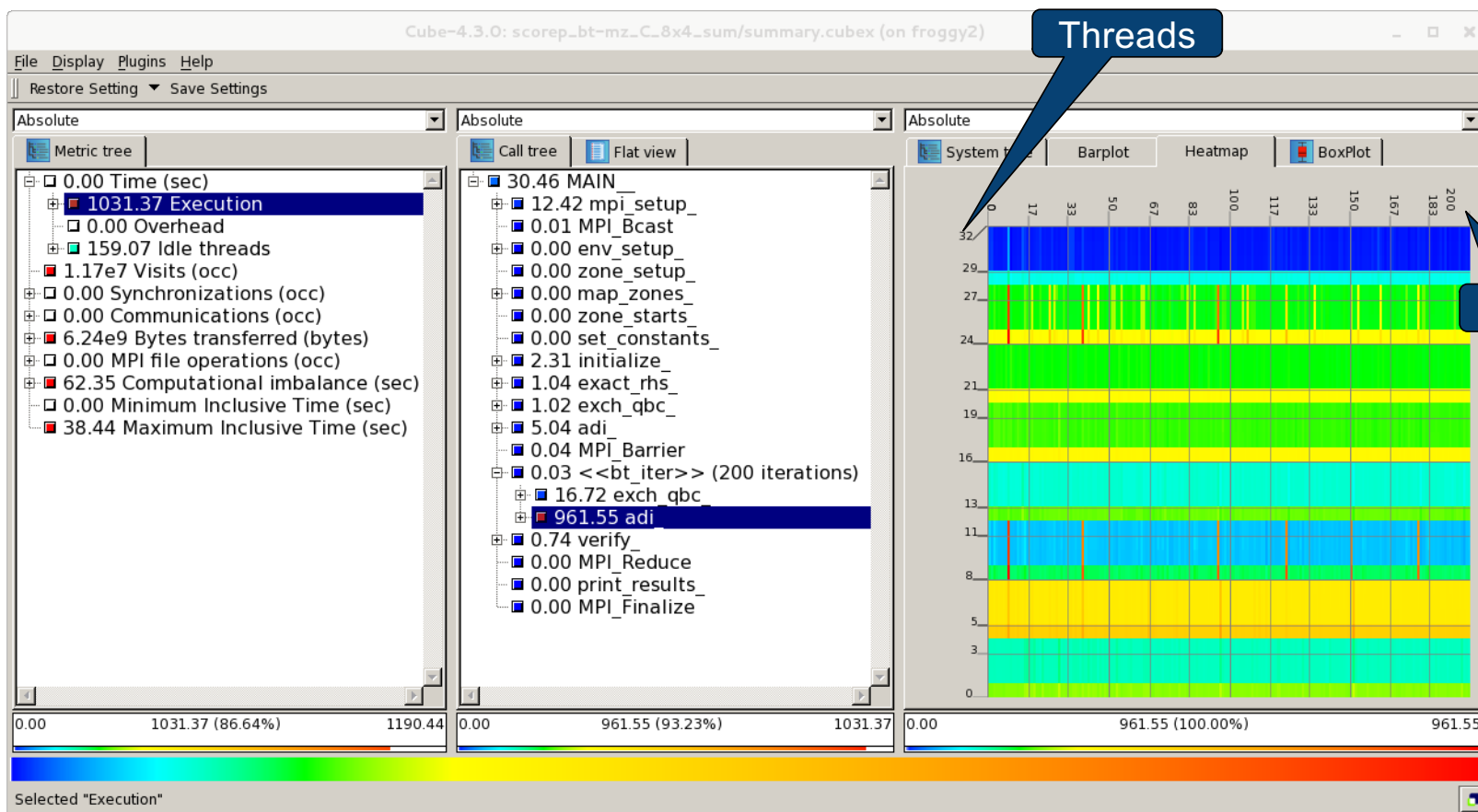
```
SCOREP_USER_REGION_DEFINE( scorep_bt_loop )  
SCOREP_USER_REGION_BEGIN( scorep_bt_loop, "<<bt_iter>>", SCOREP_USER_REGION_TYPE_DYNAMIC )  
SCOREP_USER_REGION_END( scorep_bt_loop )
```

- 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 algebra utilities

- Extracting solver sub-tree from analysis report

```
% cube_cut -r '<<ITERATION>>' scorep_bt-mz_C_8x6_sum/profile.cubex  
Writing cut.cubex... done.
```

- Calculating difference of two reports

```
% cube_diff scorep_bt-mz_C_8x6_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

Square sneak preview

- Scalasca provides **square** to facilitate analysis report exploration
 - `square = scalasca -examine [OPTIONS] (./scorep_expt_sum | ./profile.cubex)`
- Processes intermediate .cubex files produced by Score-P and Scout
 - `profile.cubex -> summary.cubex`
 - `scout.cubex -> trace.cubex`
- and (optionally) starts CUBE GUI with the post-processed file
 - containing additional derived metrics and metric hierarchies

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:
 - <https://www.scalasca.org>
- User guide also part of installation:
 - <prefix>/share/doc/CubeGuide.pdf
- Contact:
 - mailto: scalasca@fz-juelich.de

