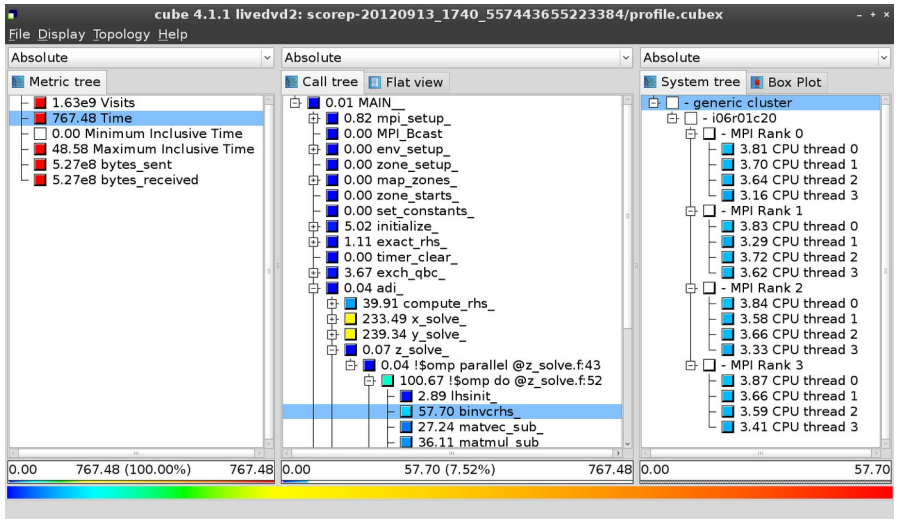
Analysis report examination with Cube

Anke Visser
Jülich Supercomputing Centre

cube
scalasca

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 v4.6 (April 2021)



Note:
Binary packages provided for Windows & MacOS, from **www.scalasca.org** website in software/Cube-4x

Cube GUI (marenostrum)

mailto: **scalasca@fz-juelich.de**

cube scalasca

- Run ***remote*** *(often convenient)*
  - start X server (e.g., Xming) locally
  - connect to marenostrum with X forwarding enabled
    - **-Y** may be faster but is insecure!
  - load cube module and start cube remotely

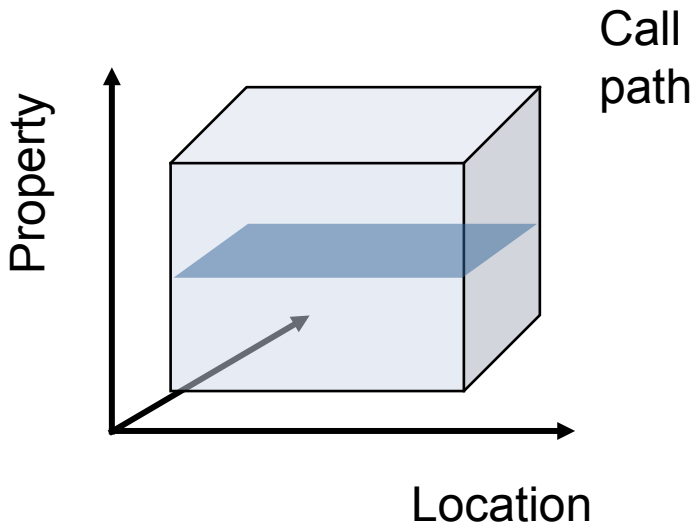
```
desk$ ssh -X userid@mn2.bsc.es
mn2$ PUBLIC=/gpfs/projects/nct00/nct00005/public
mn2$ module use $PUBLIC/software/modules
mn2$ module load cubegui/4.6
mn2$ cube ./scorep_sum/profile.cubex
```

- Install & run ***local*** *(recommended)*
  - install Cube GUI locally on desktop
    - binary packages available for MacOS & Windows and externally provided by OpenHPC and various Linux distributions
    - source package available for Linux, requires Qt
      - configure/build/install manually or use your favourite framework (e.g. Spack or EasyBuild)
  - copy .cubex file (or entire scorep directory) to desktop from remote system
    OR locally mount remote filesystem
  - start cube locally

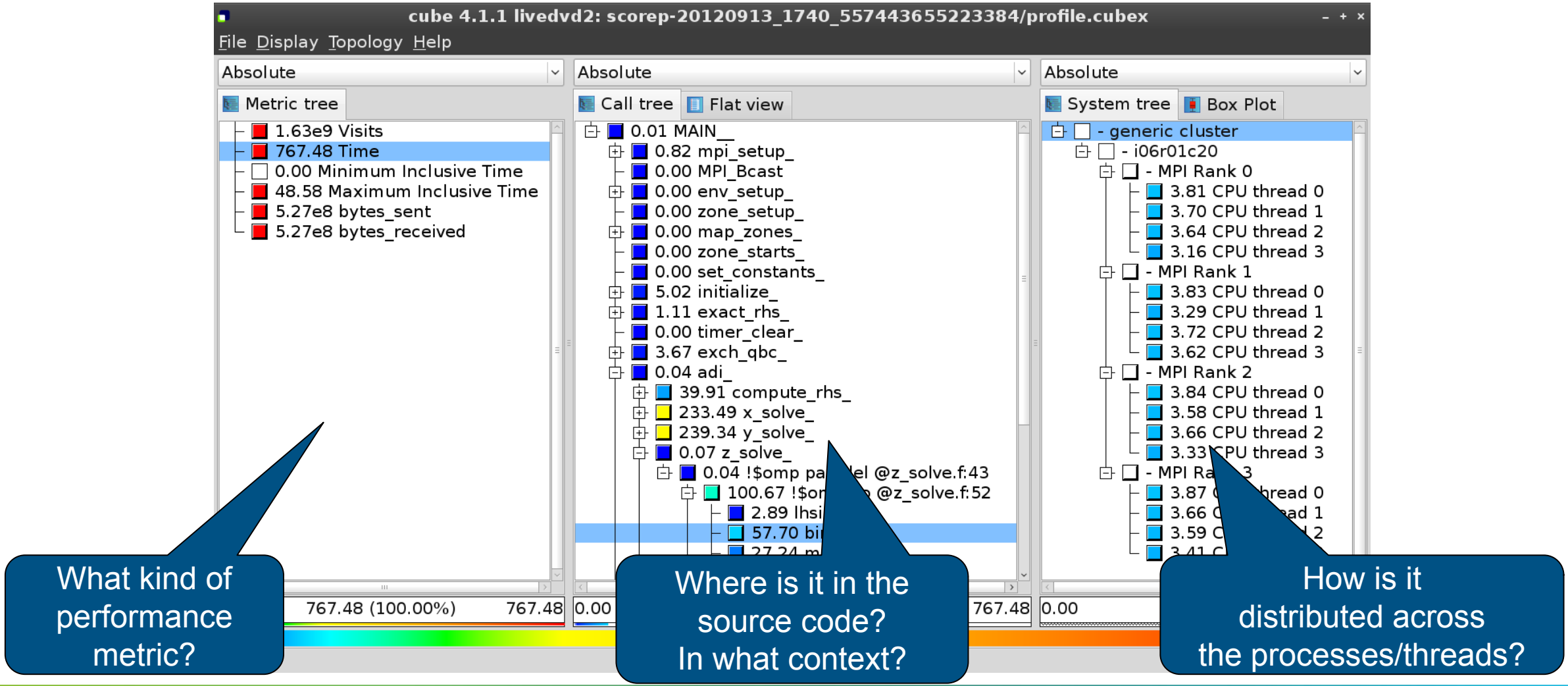
```
desk$ mkdir $HOME/mnt
desk$ sshfs [user@]mn2.bsc.es:[dir] $HOME/mnt
desk$ cd $HOME/mnt
desk$ cube ./scorep_sum/profile.cubex
```

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

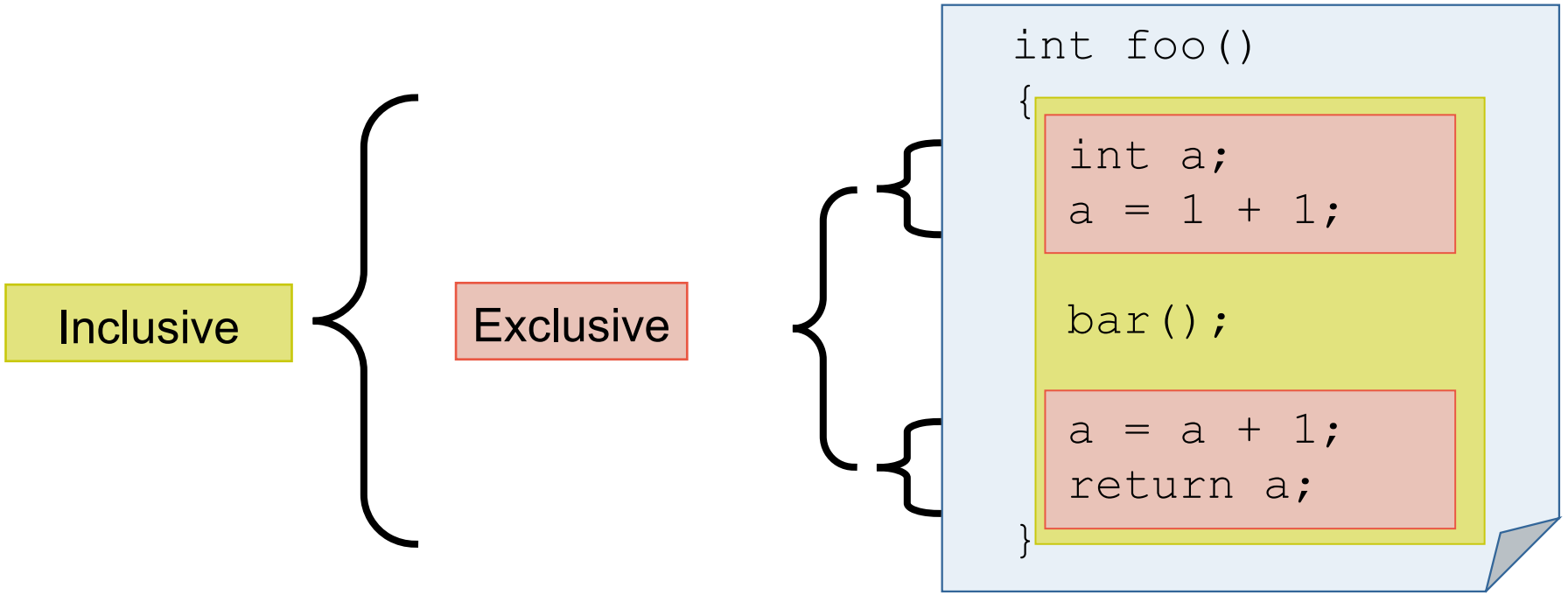


Analysis presentation

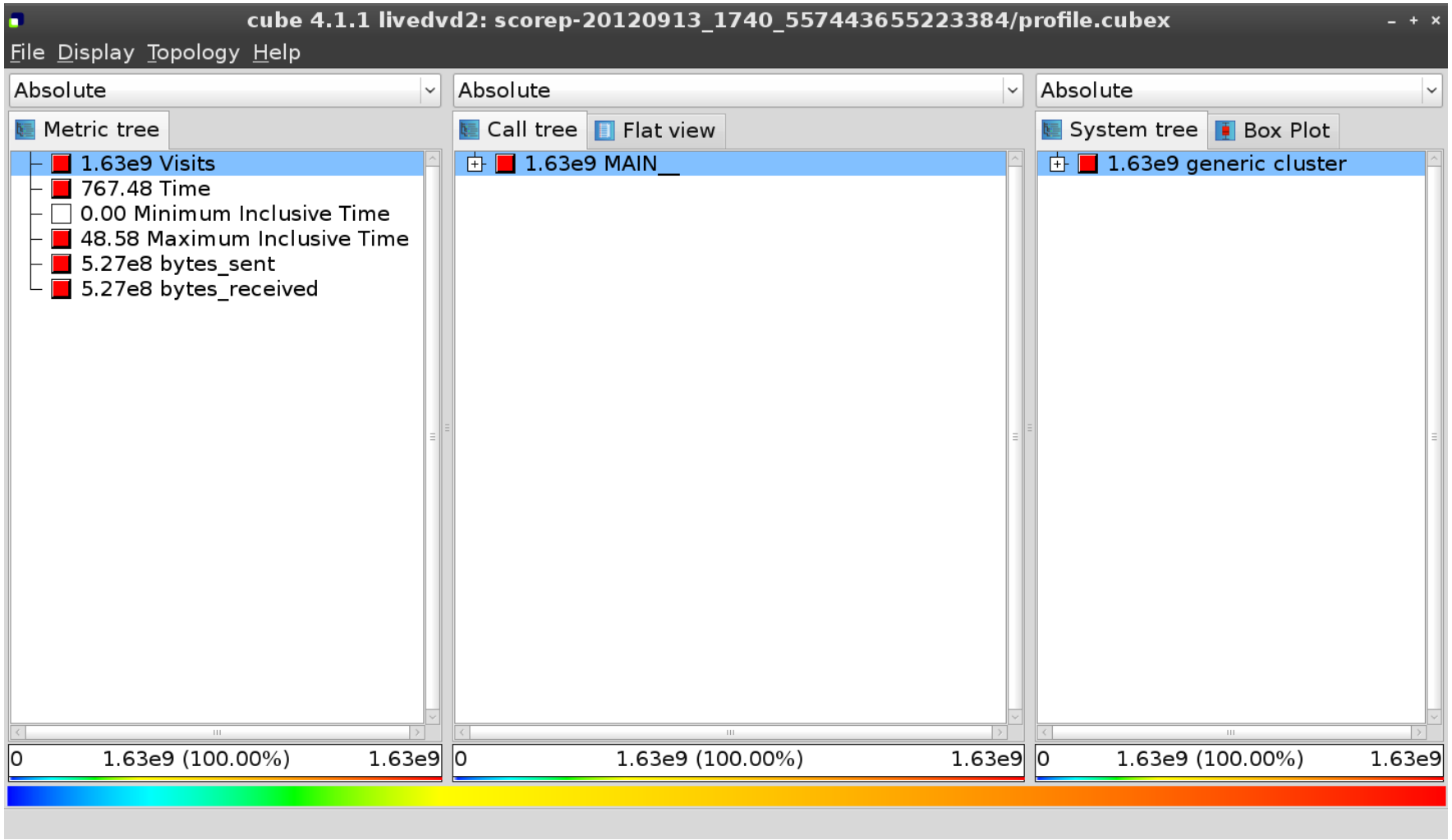


Inclusive vs. exclusive values

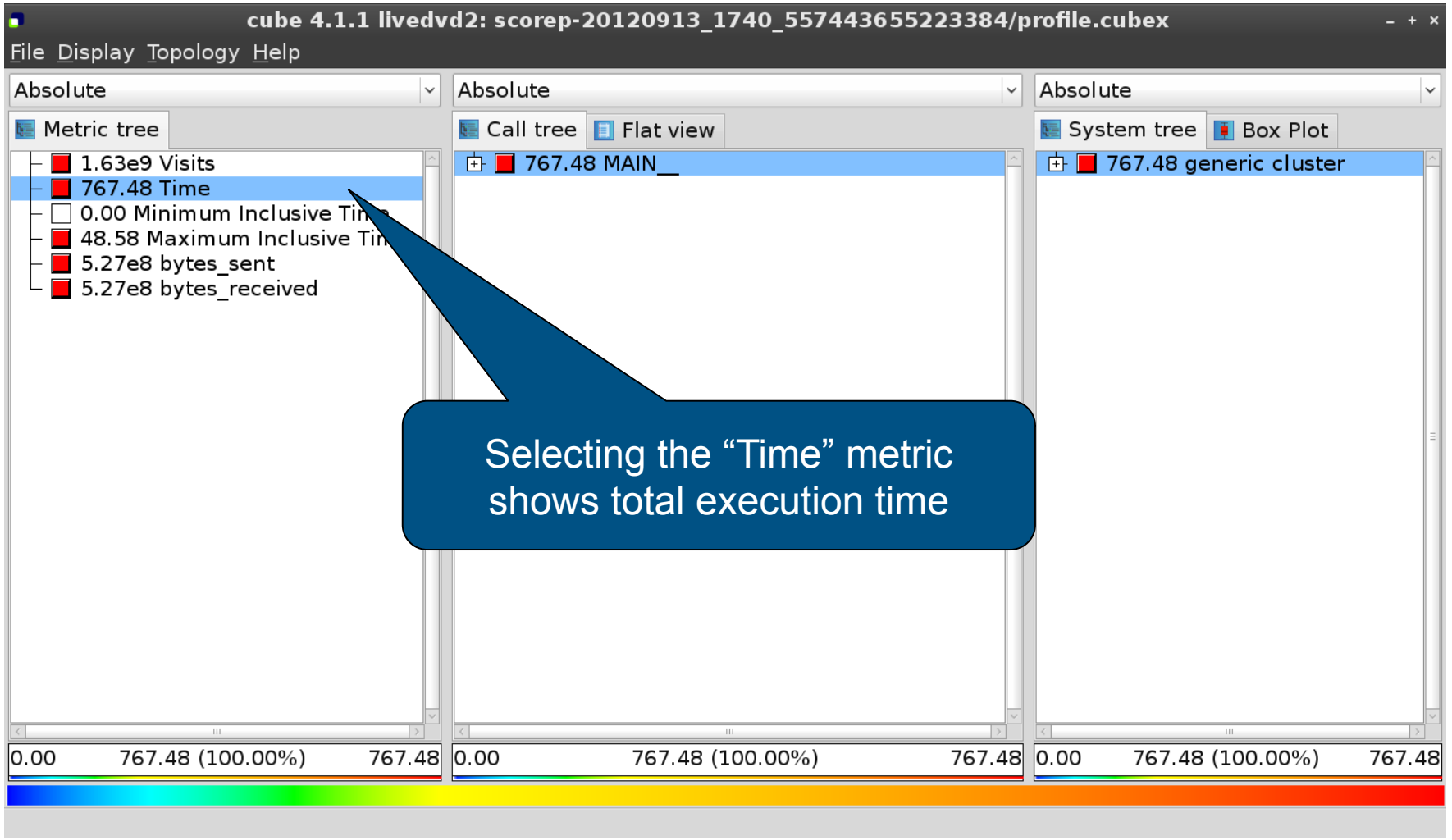
- Inclusive
  - Information of all sub-elements aggregated into single value
- Exclusive
  - Information cannot be subdivided further



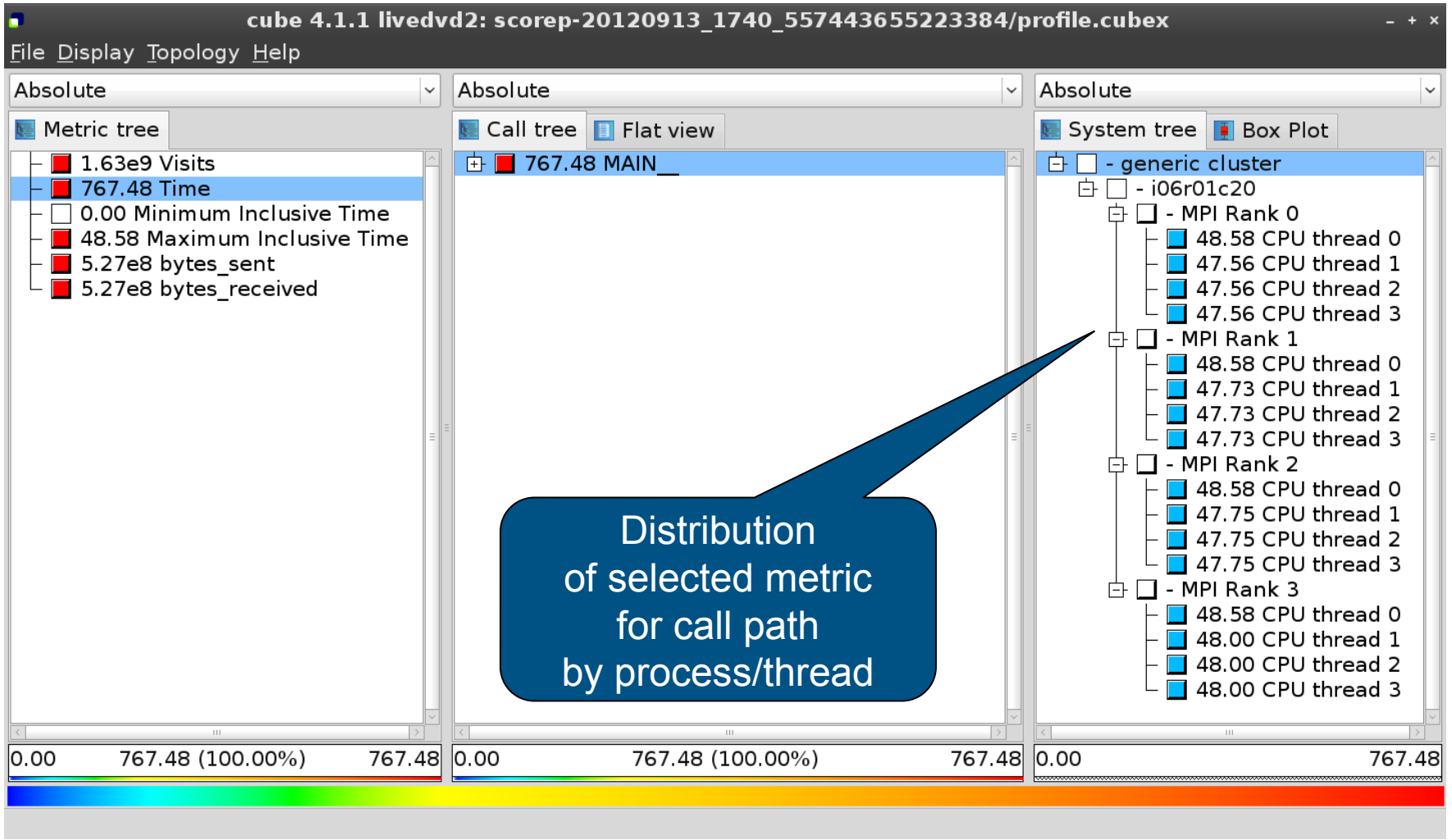
Score-P analysis report exploration (opening view)



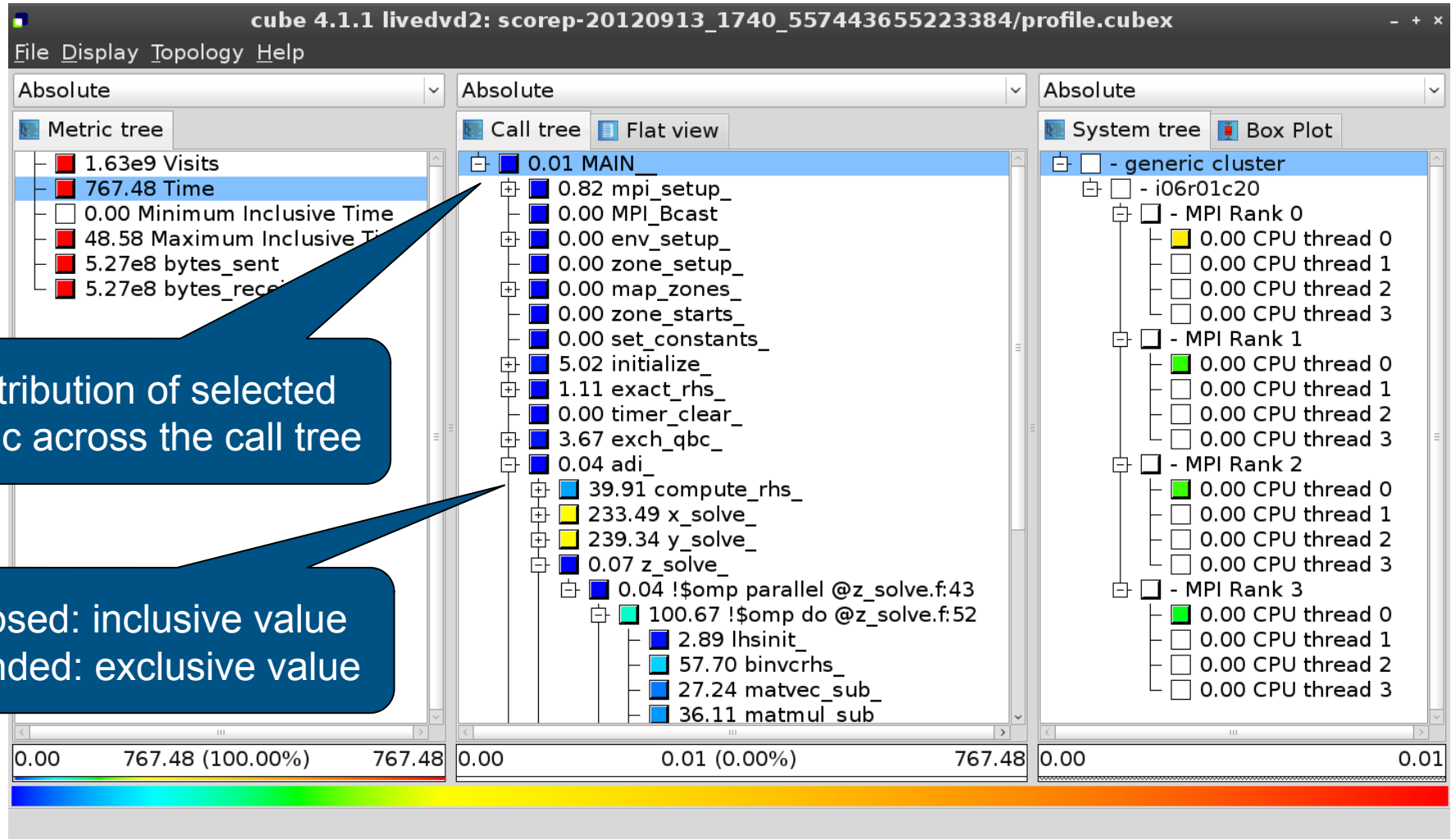
Metric selection



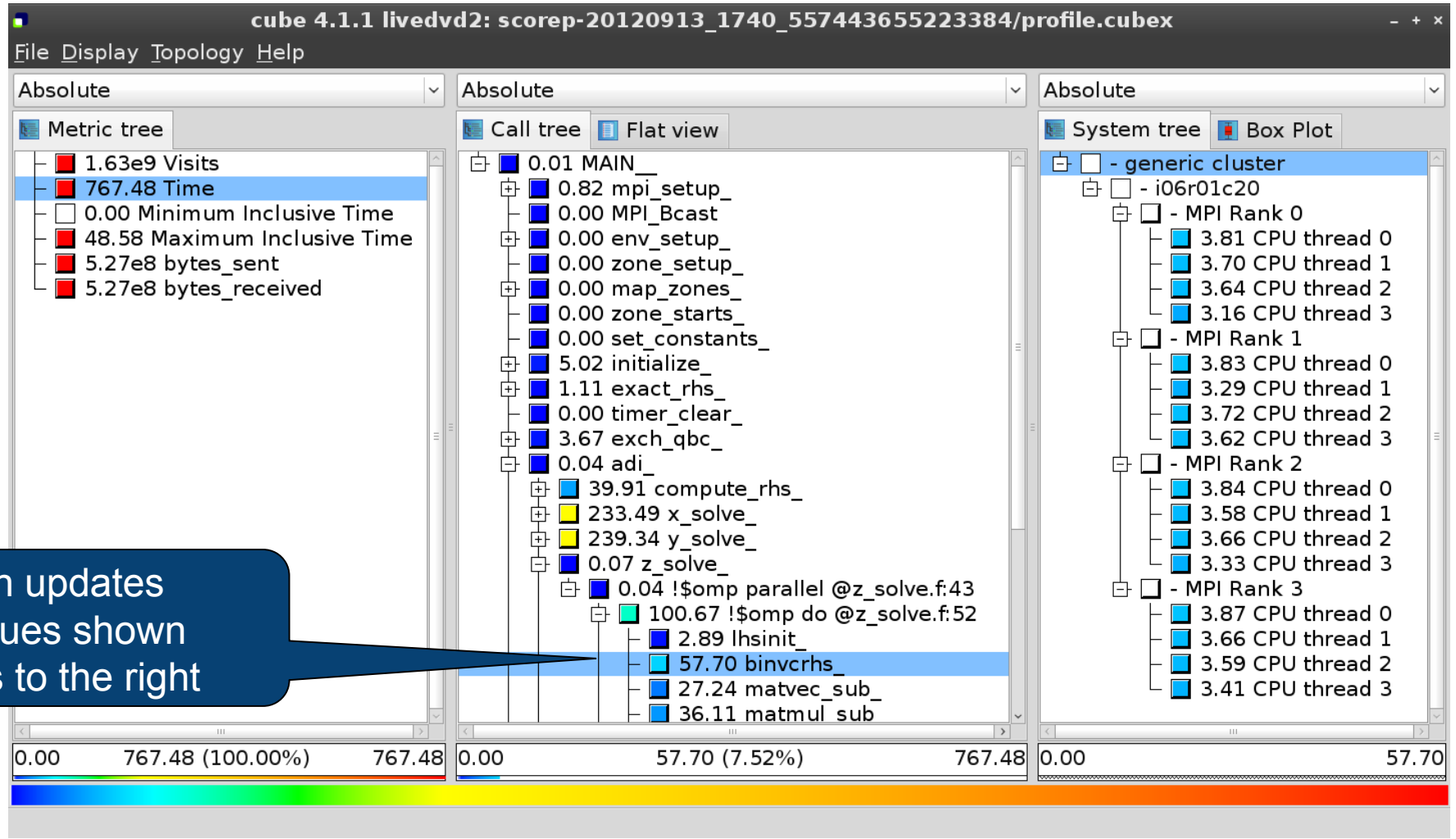
Expanding the system tree



Expanding the call tree

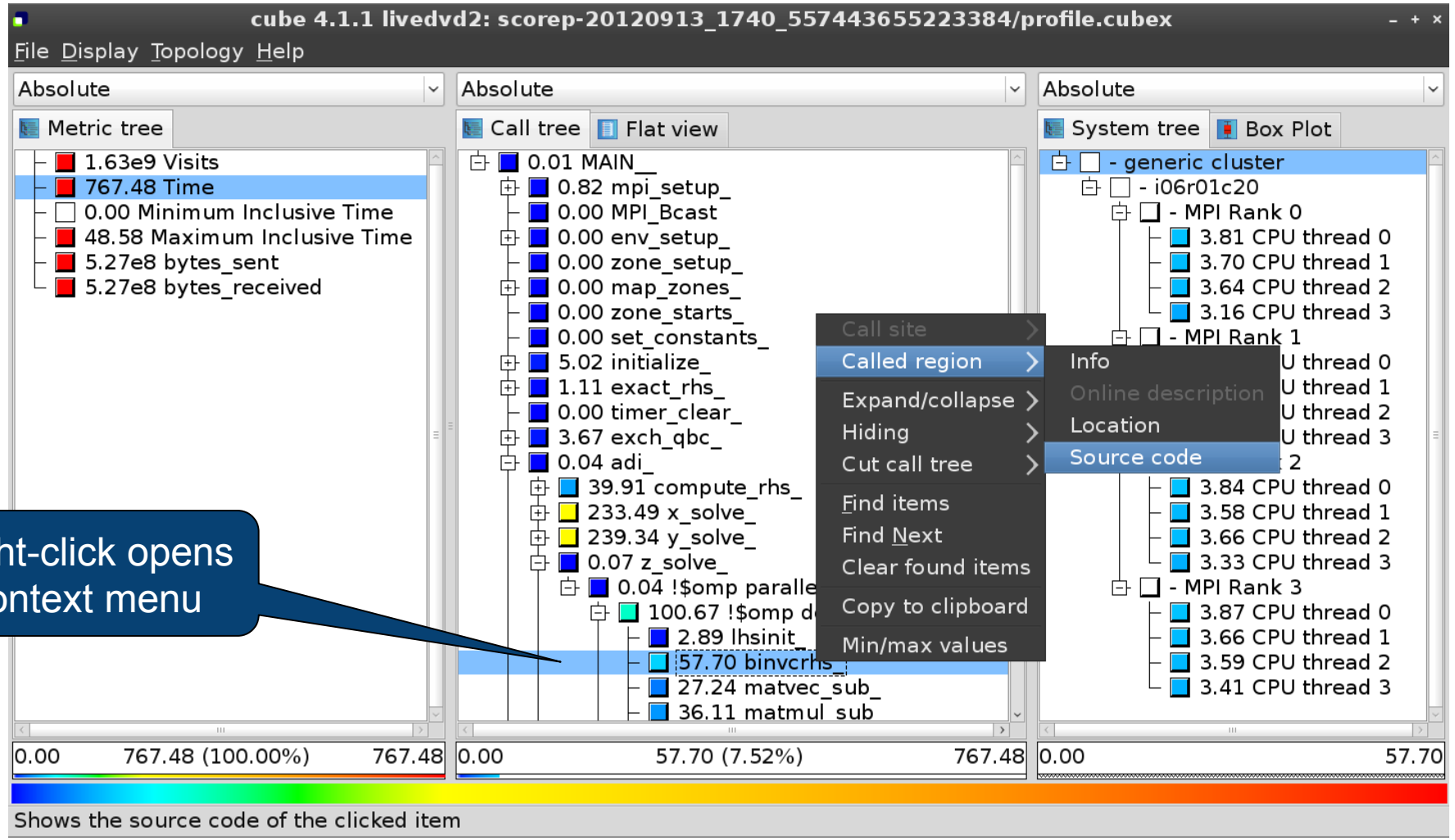


Selecting a call path

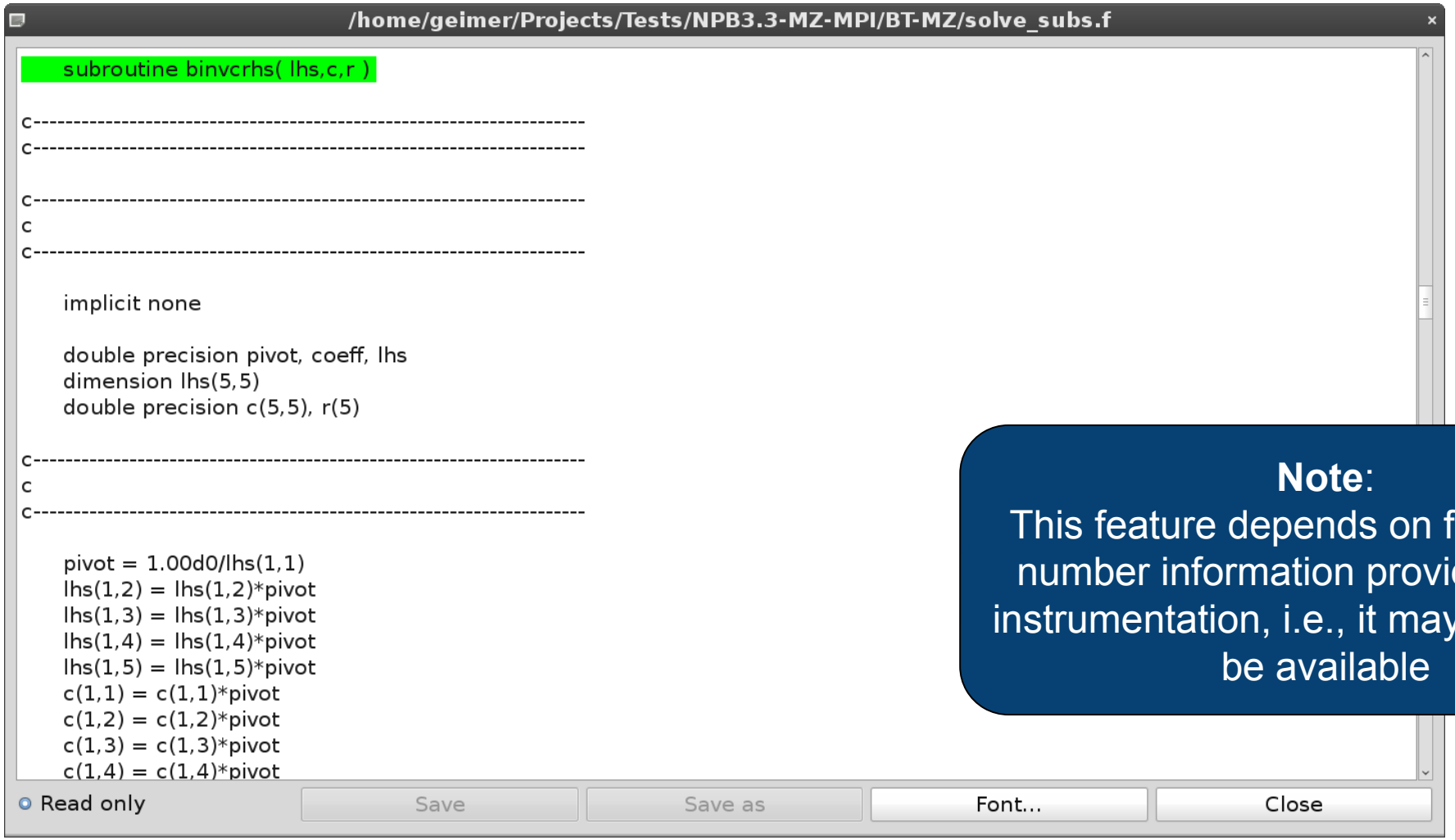


Selection updates metric values shown in columns to the right

Source-code view via context menu

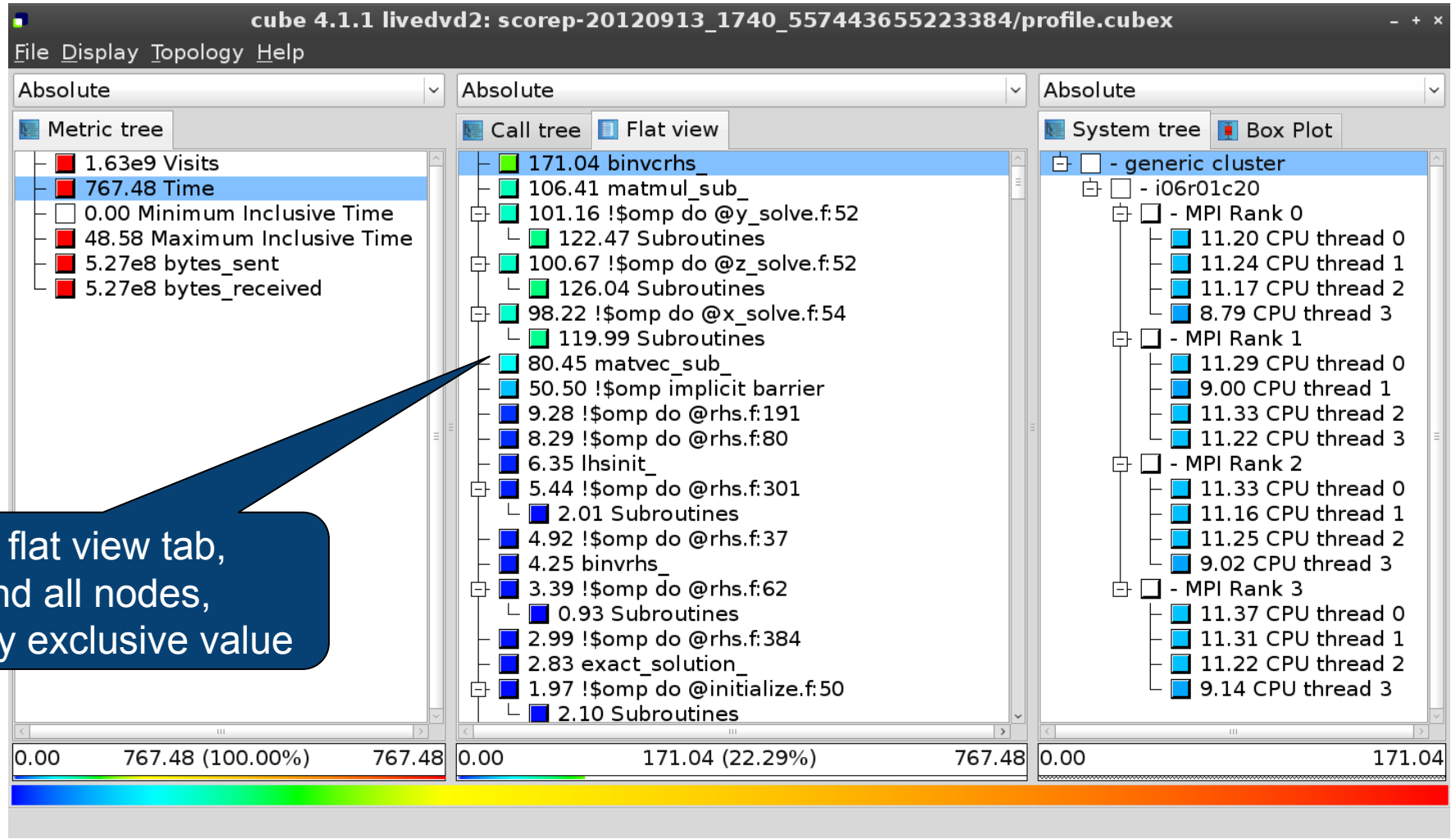


Source-code view

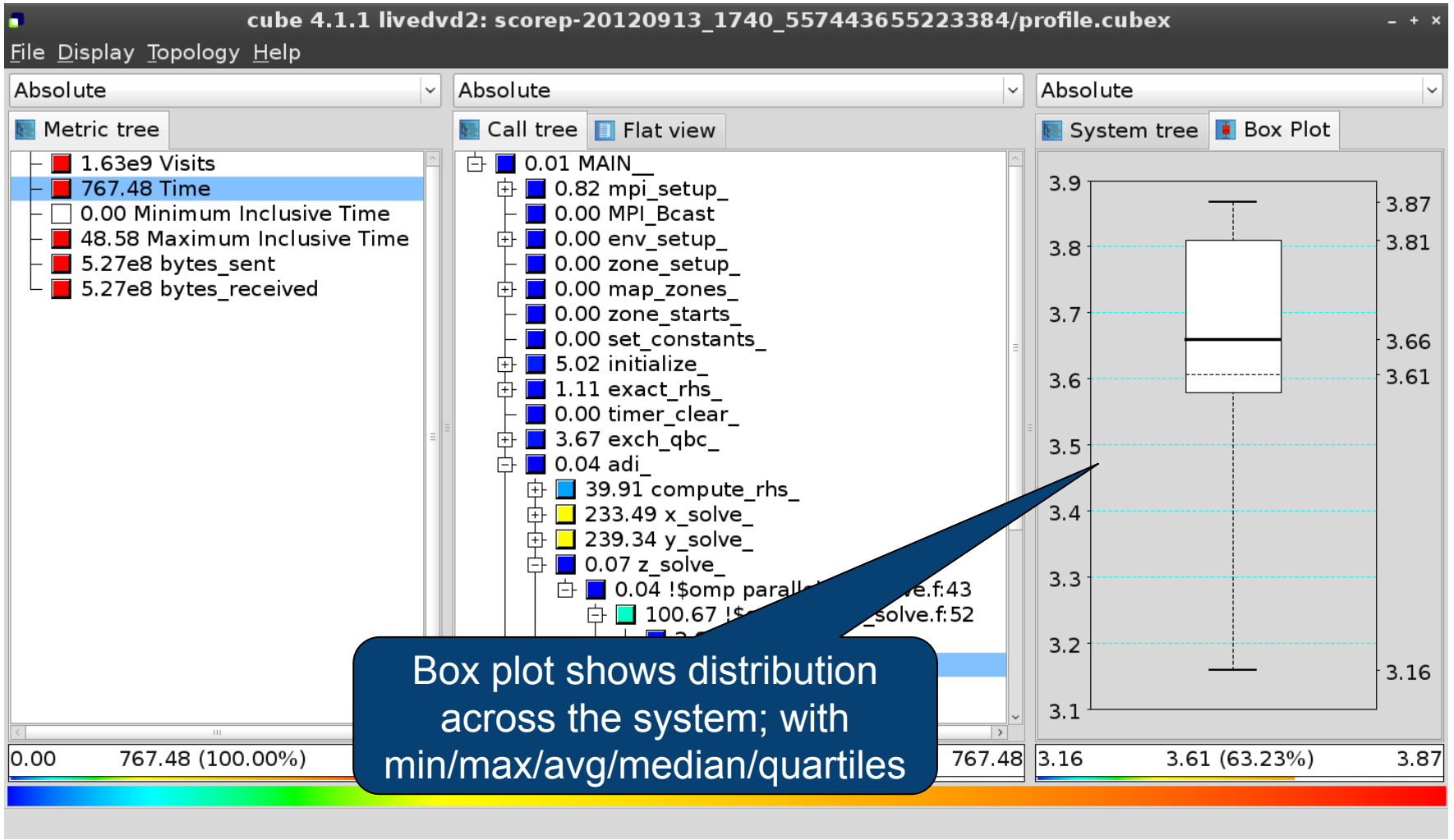


Note:
This feature depends on file and line number information provided by the instrumentation, i.e., it may not always be available

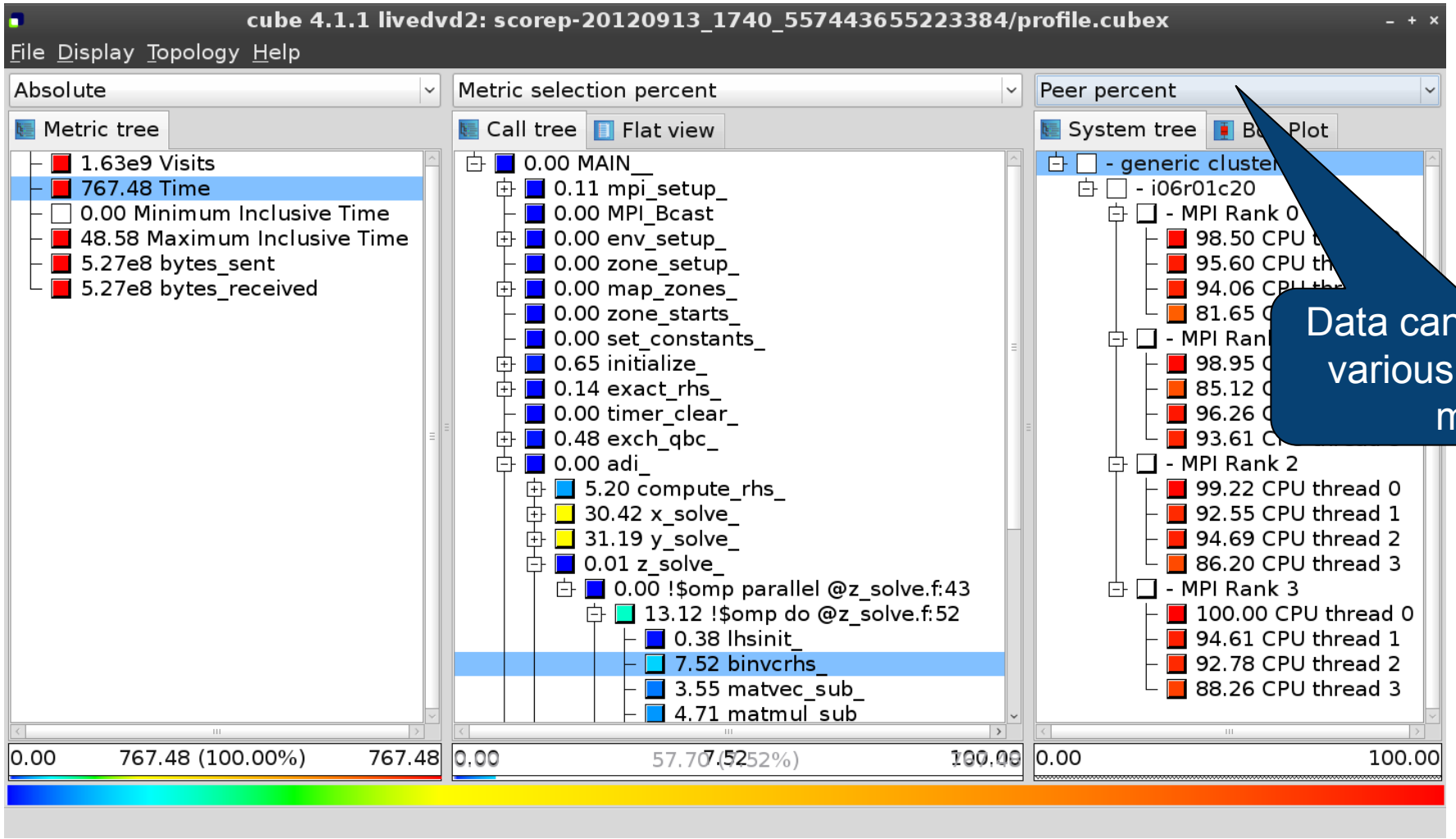
Flat profile view



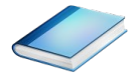
Box plot view



Alternative display modes

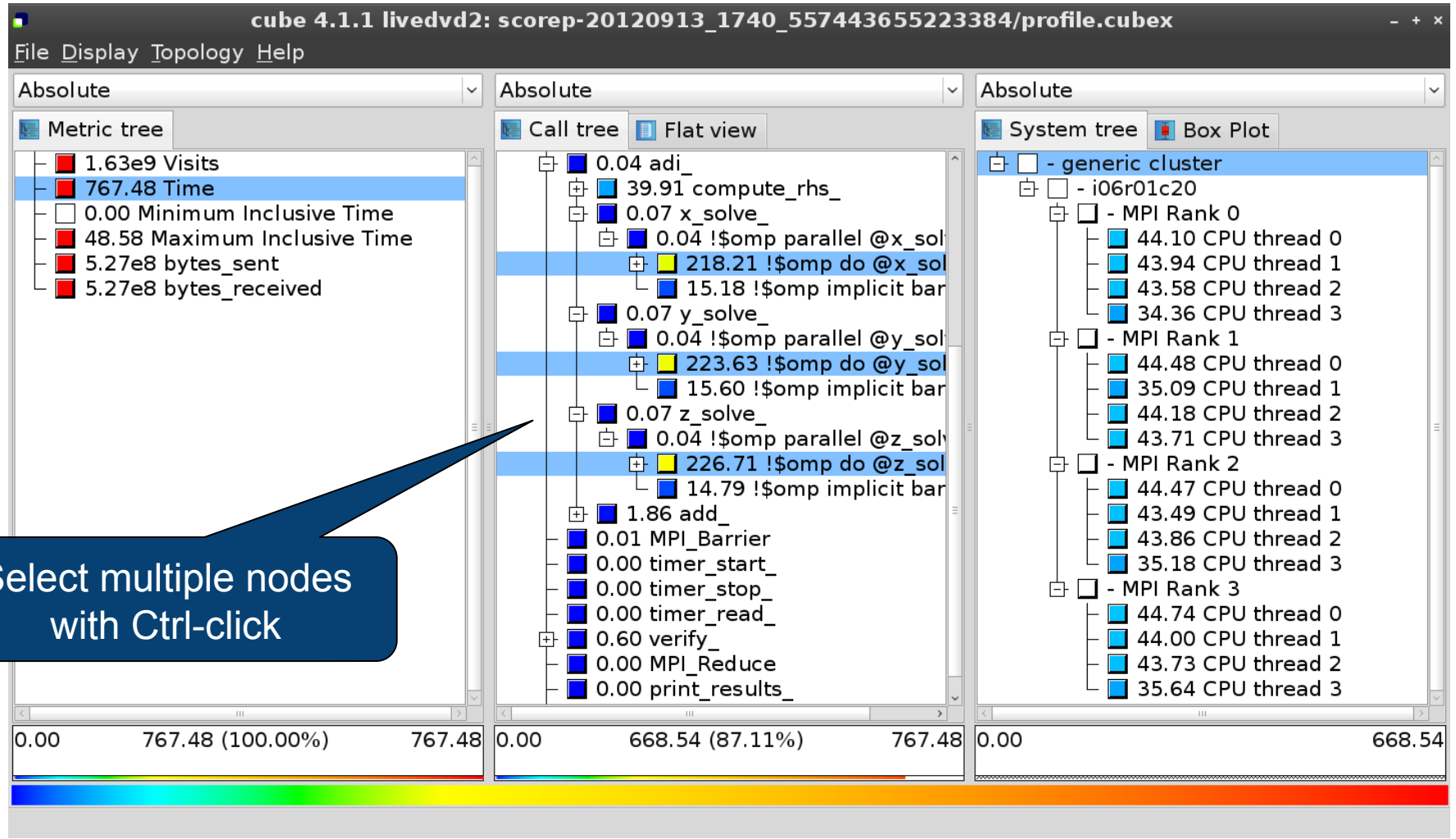


Important display modes

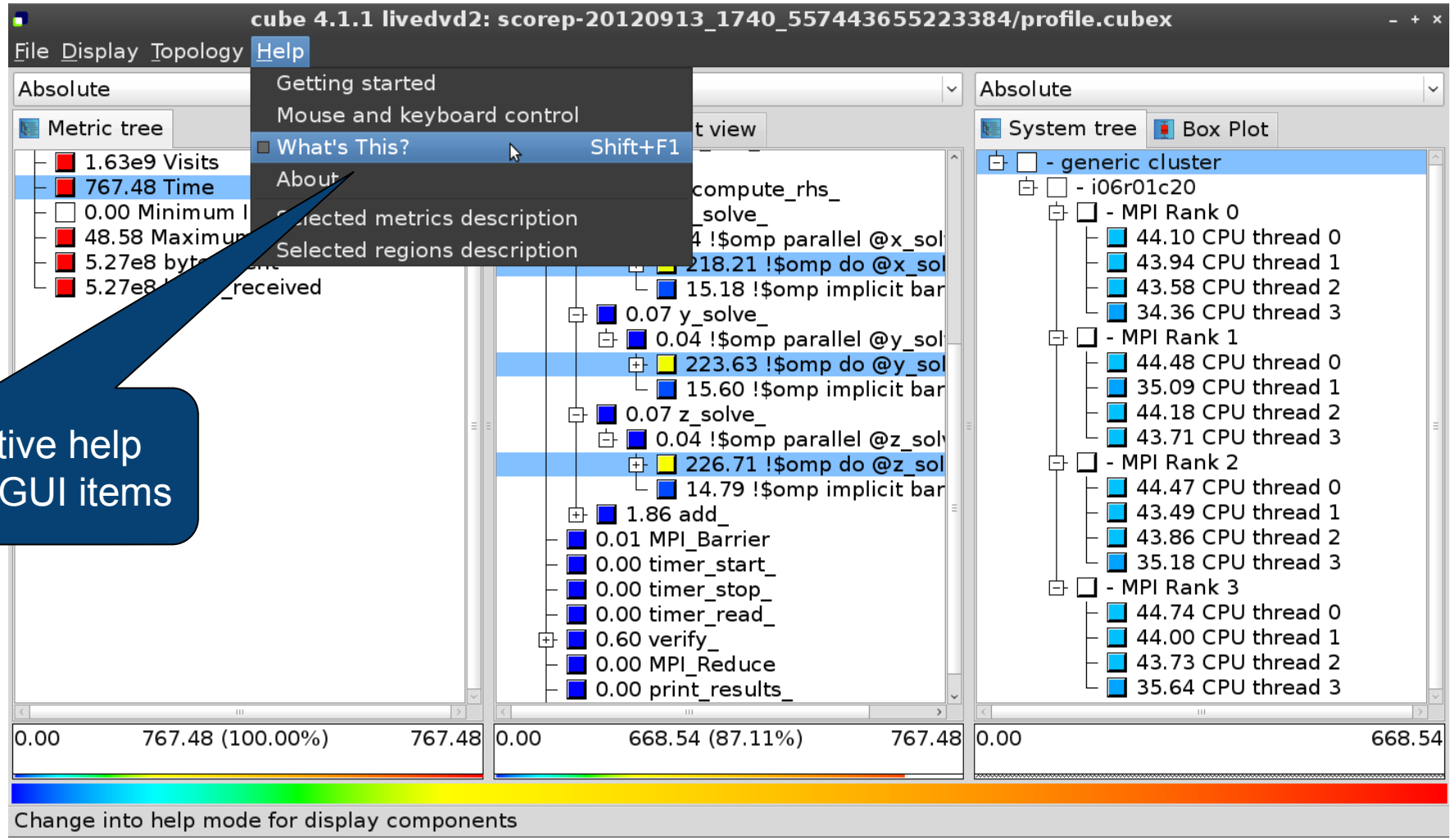


- Absolute
  - Absolute value shown in seconds/bytes/counts
- Selection percent
  - Value shown as percentage w.r.t. the selected node "on the left" (metric/call path)
- Peer percent (system tree only)
  - Value shown as percentage relative to the maximum peer value

Multiple selection

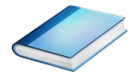


Context-sensitive help



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

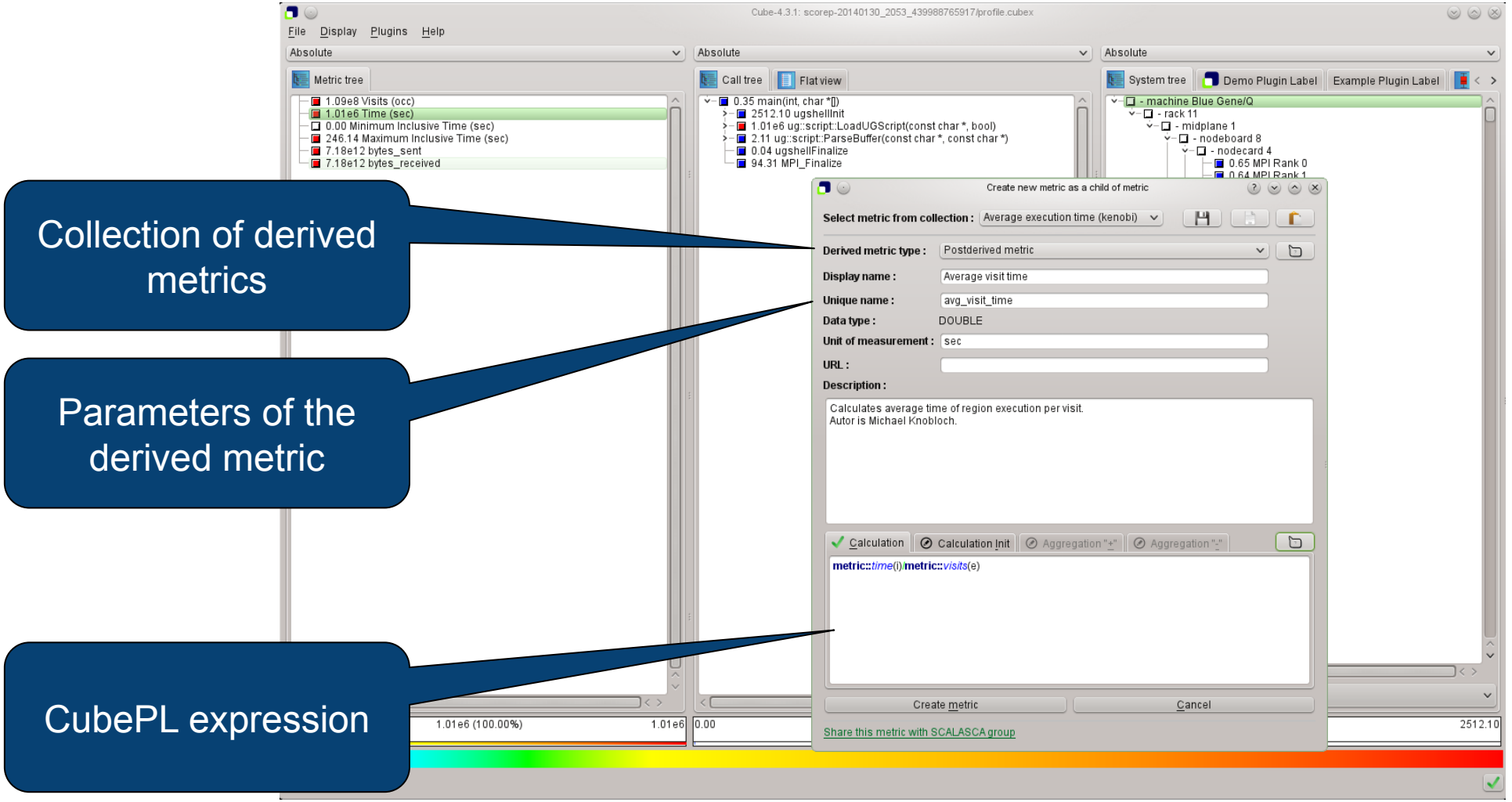
- Examples:
  - "Average execution time": Postderived metric with expression

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

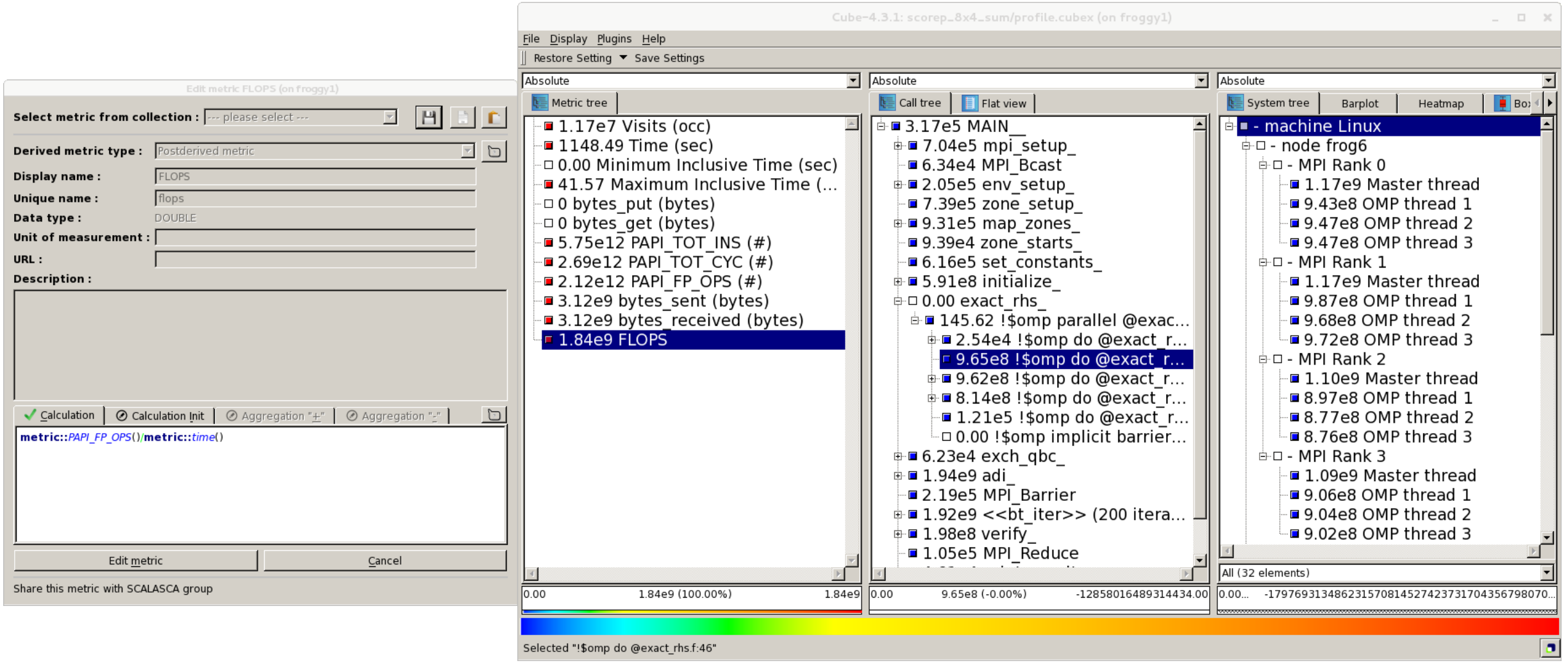
  - "Number of FLOP per second": Postderived metric with expression

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

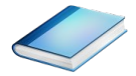
Derived metrics in Cube GUI



Example: FLOPS based on PAPI_FP_OPS and time



Iteration profiling

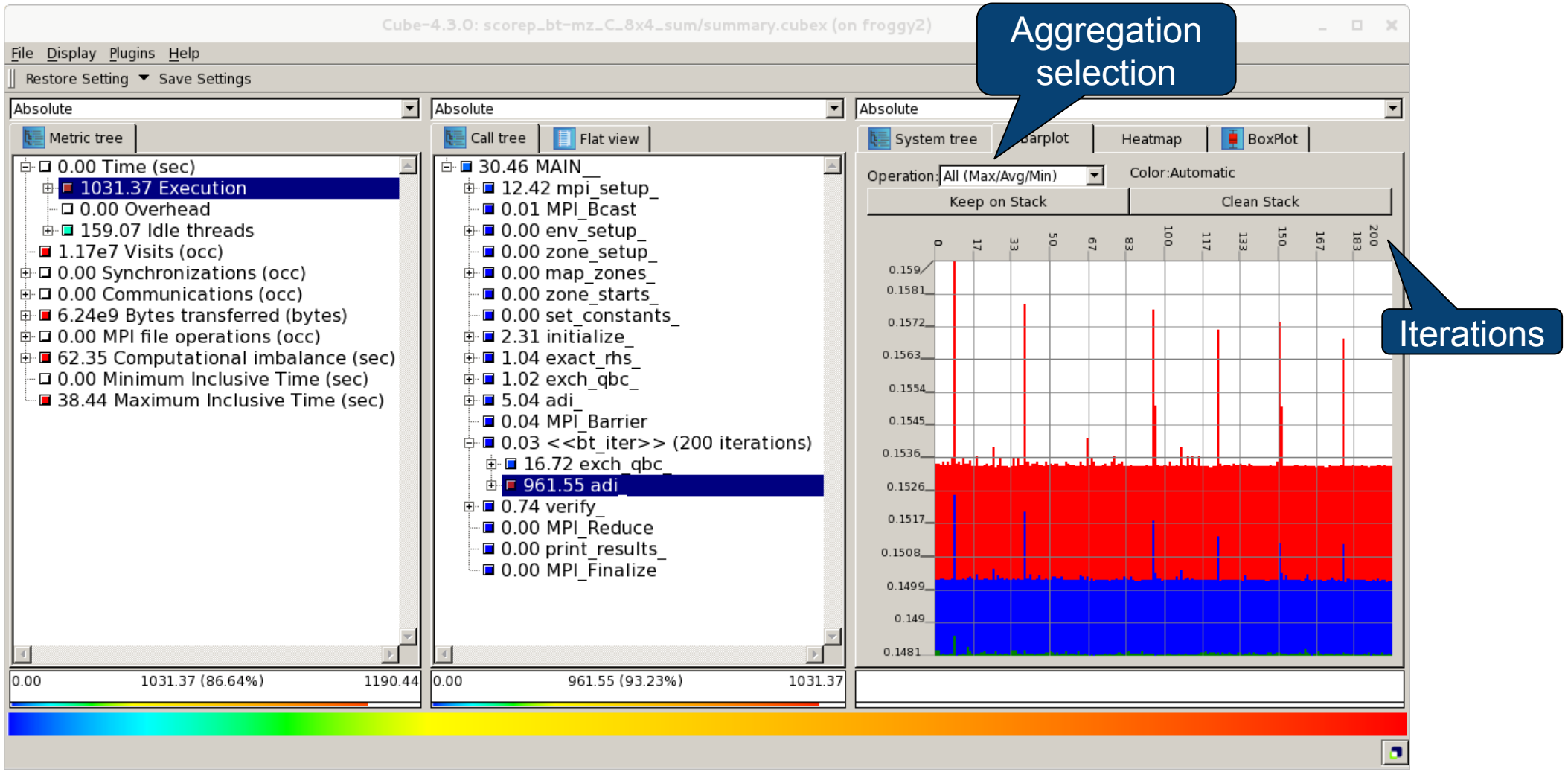


- Show time dependent behavior by “unrolling” iterations
- Preparations:
  - Mark loop body by using Score-P instrumentation API in your source cod
  - ```
    SCOREP_USER_REGION_DEFINE( scorep_bt_loop )
    SCOREP_USER_REGION_BEGIN( scorep_bt_loop, "<<bt_iter>>", SCOREP_USER_REGION_TYPE_DYNAMIC )
    SCOREP_USER_REGION_END( scorep_bt_loop )
    ```
  - Iterations shown as separate call trees
  - ➢Useful for checking results for specific iterations

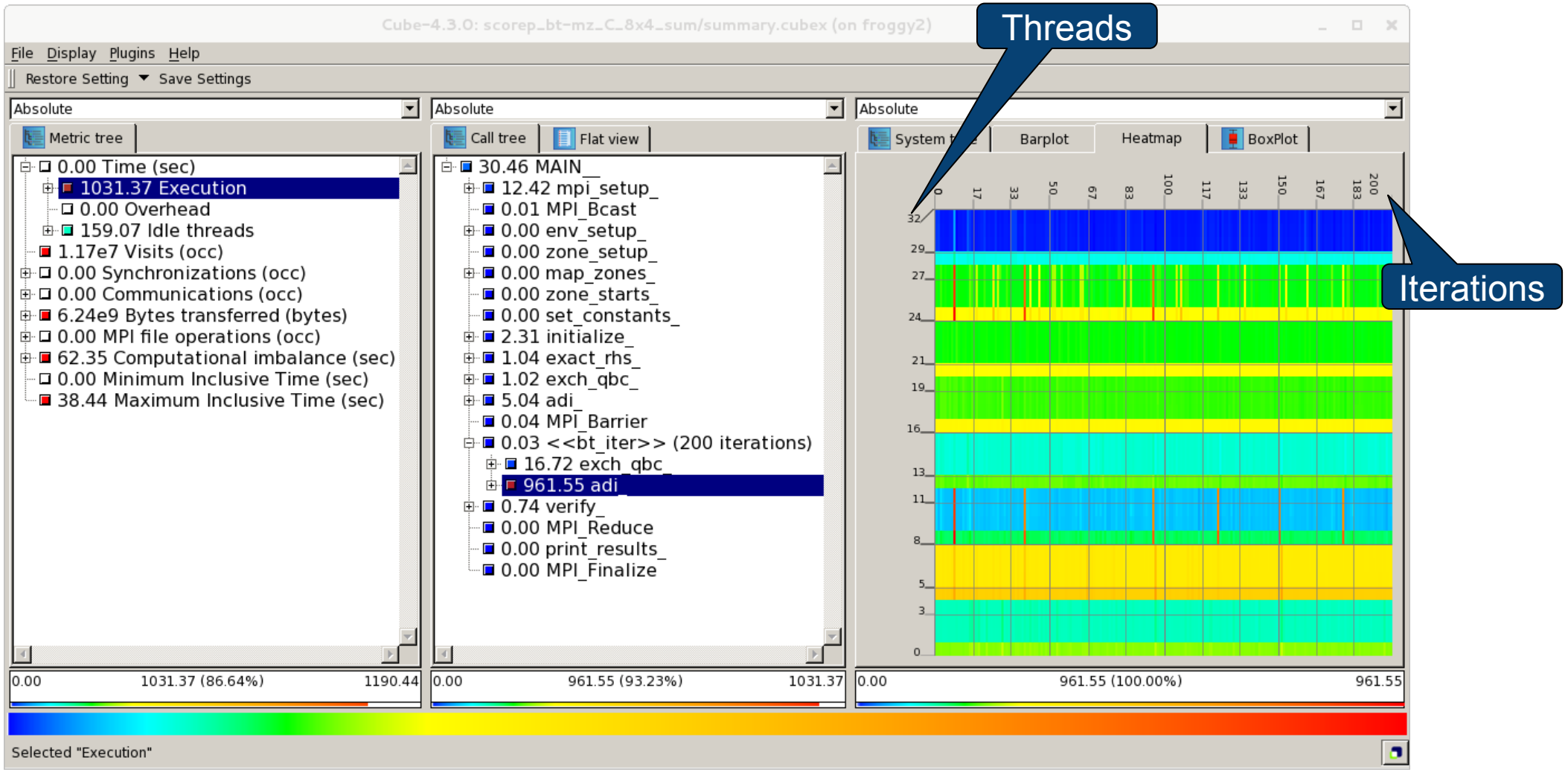
  or

  - 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_16x8_sum/profile.cubex
Writing cut.cubex... done.
```

- Calculating difference of two reports

```
% cube_diff  scorep_bt-mz_C_16x8_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

trace tools scalasca

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:
  - <prefix>/share/doc/CubeGuide.pdf
- Contact:
  - mailto: scalasca@fz-juelich.de