

## Analysis report examination with Cube

---

**Markus Geimer**  
Jülich Supercomputing Centre

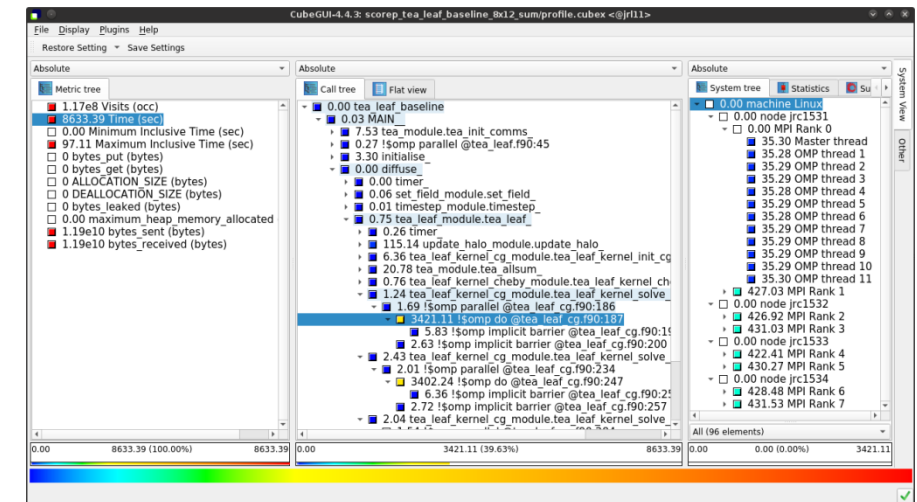


# Cube

CubeLib DOI 10.5281/zenodo.1248078

CubeGUI DOI 10.5281/zenodo.1248087

- 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  $\geq$  5
- Originally developed as part of the Scalasca toolset
- Now available as a separate component
  - Can be installed independently of Score-P and Scalasca, e.g., on laptop or desktop
  - Latest release: Cube v4.8.2 (September 2023)

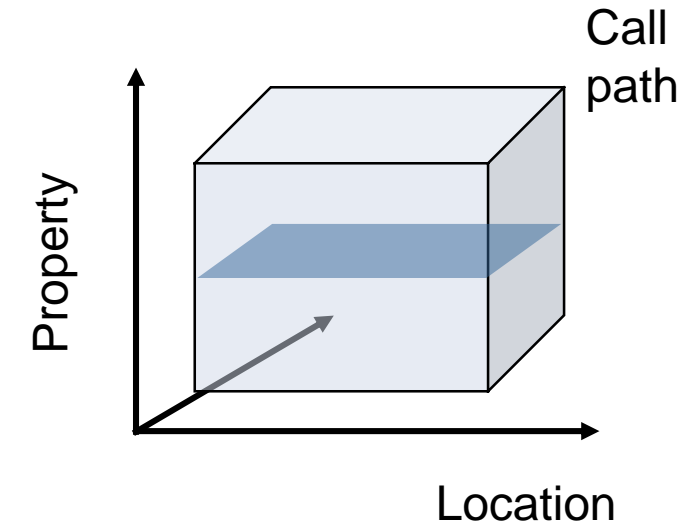


**Note:** source distribution tarballs for Linux, as well as binary packages provided for Windows & MacOS, from [www.scalasca.org](http://www.scalasca.org) website in software/Cube-4x

# Analysis presentation and exploration

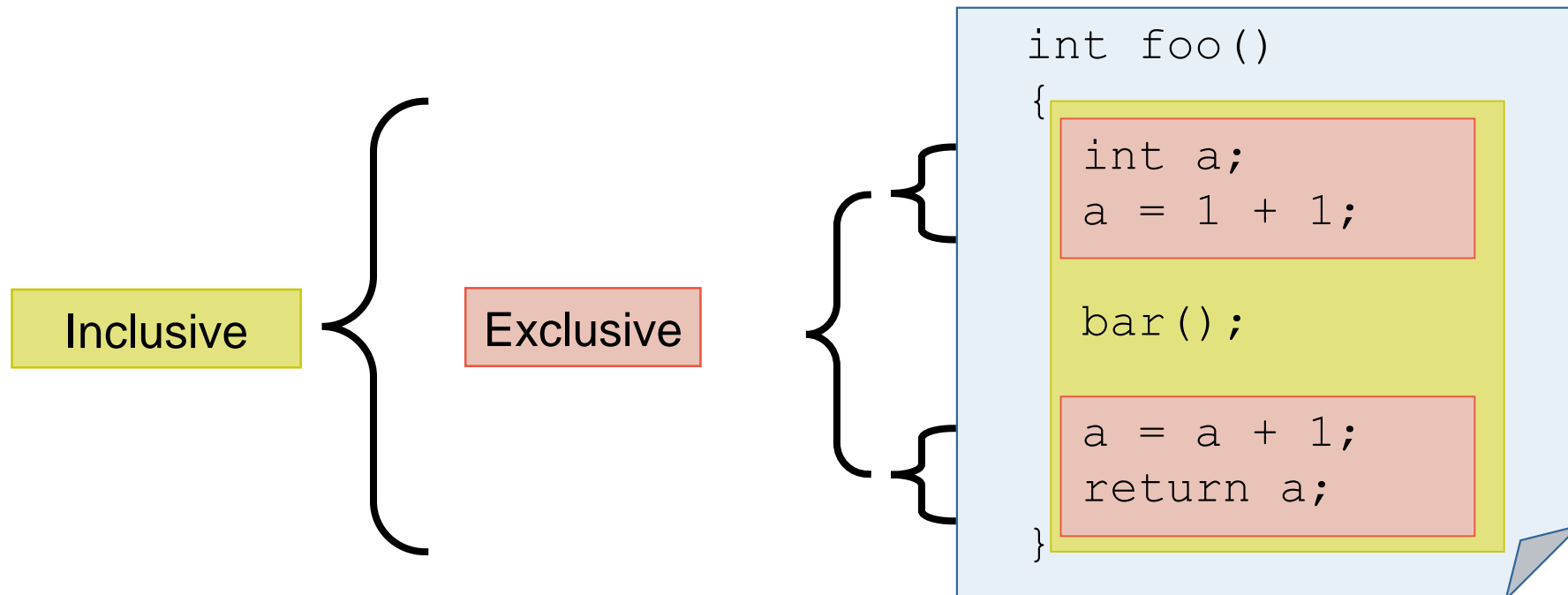
---

- 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



# Score-P analysis report exploration (opening view)

CubeGUI-4.4.3: scorep\_tea\_leaf\_baseline\_8x12\_sum/profile.cubex <@jrl11>

File Display Plugins Help

Restore Setting Save Settings

Absolute

Metric tree

- 1.17e8 Visits (occ)
- 8633.39 Time (sec)
- 0.00 Minimum Inclusive Time (sec)
- 97.11 Maximum Inclusive Time (sec)
- 0 bytes\_put (bytes)
- 0 bytes\_get (bytes)
- 0 ALLOCATION\_SIZE (bytes)
- 0 DEALLOCATION\_SIZE (bytes)
- 0 bytes\_leaked (bytes)
- 0.00 maximum\_heap\_memory\_allocated (bytes)
- 1.19e10 bytes\_sent (bytes)
- 1.19e10 bytes\_received (bytes)

Absolute

Call tree Flat view

- 1.17e8 tea leaf baseline

Absolute

System tree Statistics Sunburst Pr

- 1.17e8 machine Linux

System View Other

All (96 elements)

1.17e8 (100.00%) 1.17e8 0 1.17e8 (100.00%)

What kind of performance metric?

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

How is it distributed across the processes/threads?

# Metric selection

CubeGUI-4.4.3: scorep\_tea\_leaf\_baseline\_8x12\_sum/profile.cubex <@jrl11>

File Display Plugins Help

Restore Setting Save Settings

Absolute

Metric tree

- 1.17e8 Visits (occ)
- 8633.39 Time (sec)
- 0.00 Minimum Inclusive Time (sec)
- 97.11 Maximum Inclusive Time (sec)
- 0 bytes\_put (bytes)
- 0 bytes\_get (bytes)
- 0 ALLOCATION\_SIZE (bytes)
- 0 DEALLOCATION\_SIZE (bytes)
- 0 bytes\_leaked (bytes)
- 0.00 maximum\_heap\_memory\_allocated (bytes)
- 1.19e10 bytes\_sent (bytes)
- 1.19e10 bytes\_received (bytes)

Absolute

Call tree Flat view

8633.39 tea leaf baseline

Absolute

System tree Statistics Sunburst Pr

8633.39 machine Linux

System View Other

0.00 8633.39 (100.00%) 8633.39

0.00 8633.39 (100.00%) 8633.39

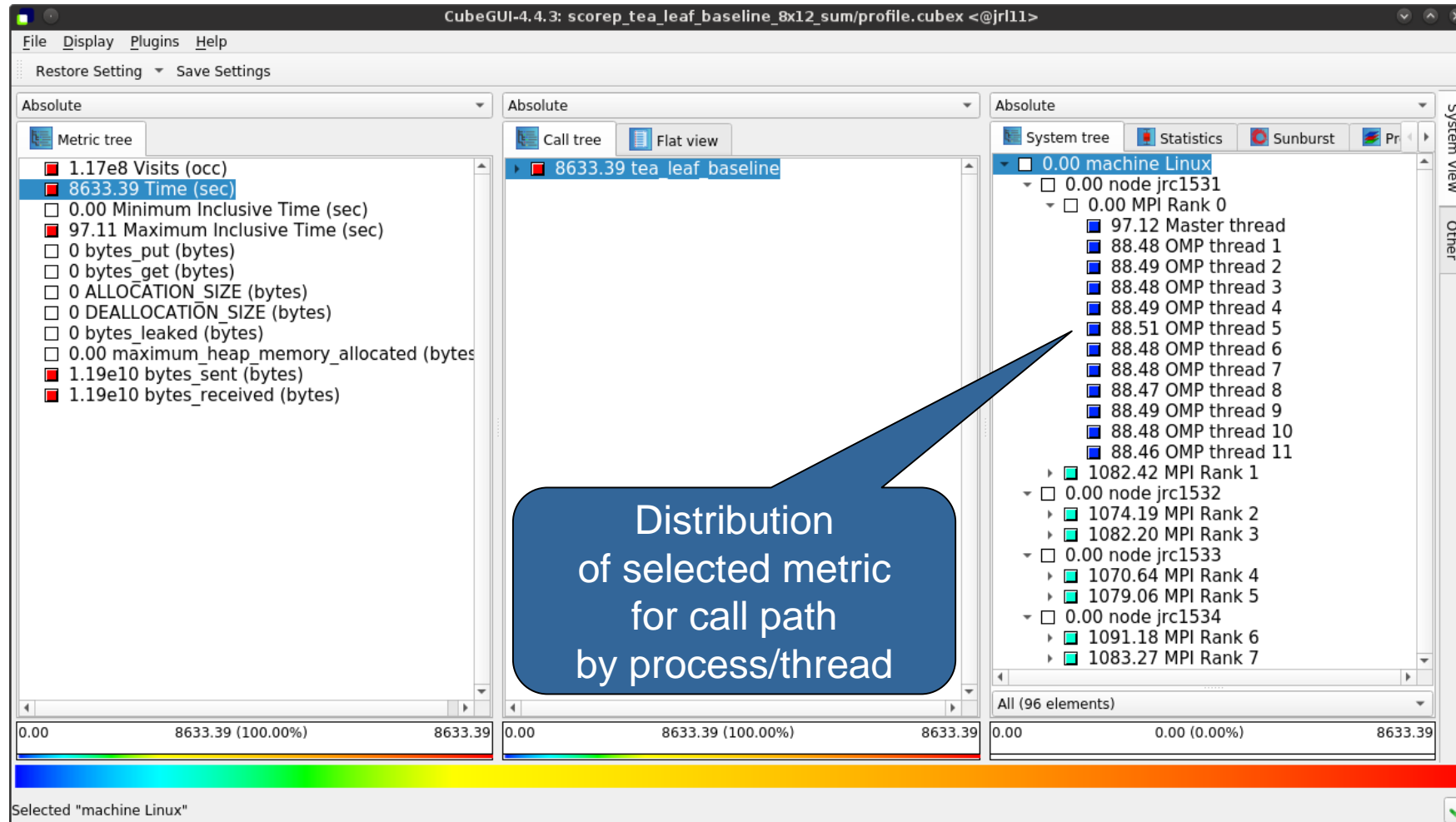
0.00 8633.39 (100.00%) 8633.39

All (96 elements)

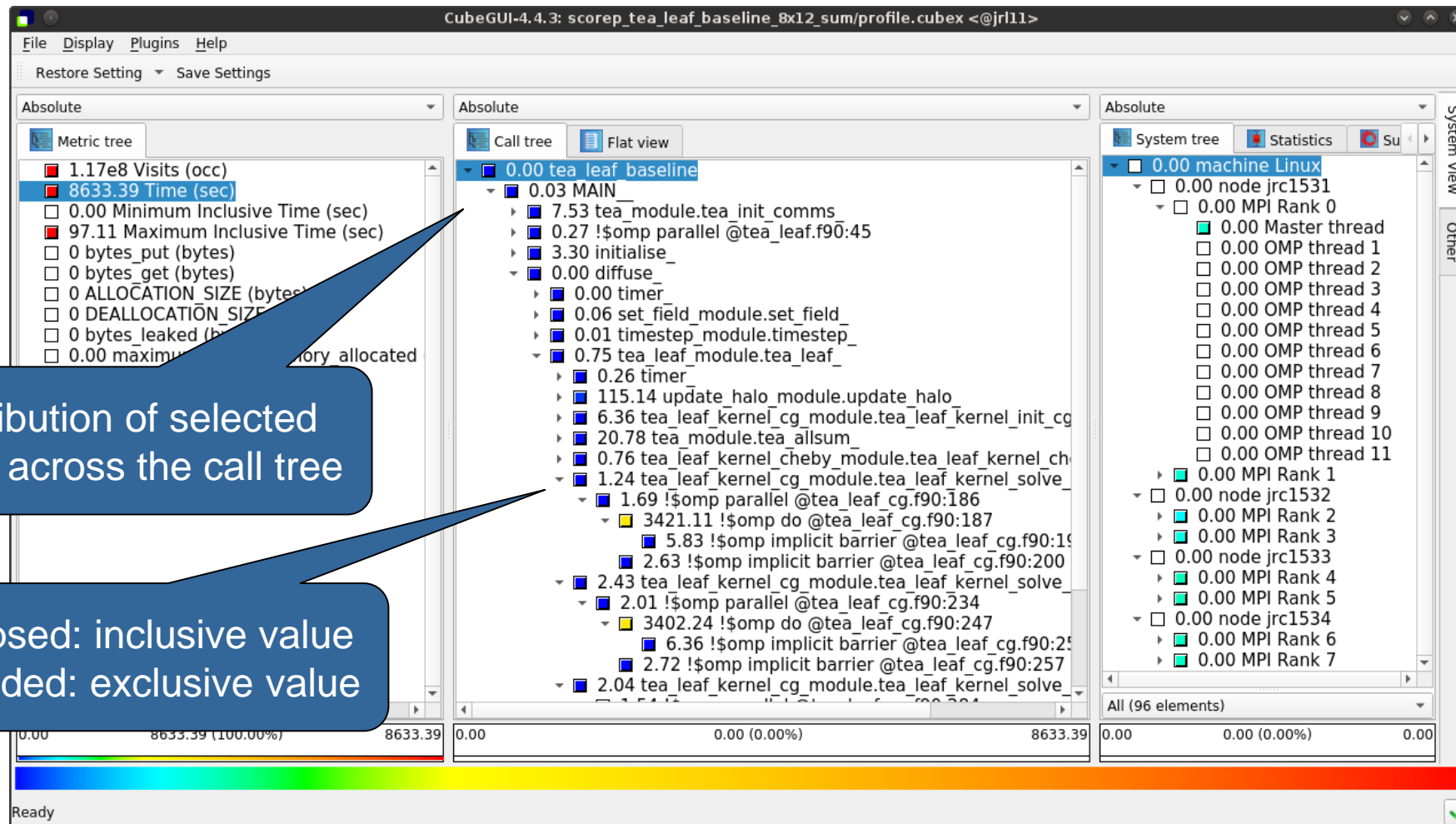
Selected "Time"

Selecting the "Time" metric shows total execution time

# Expanding the system tree



# Expanding the call tree





# Selecting a call path

The screenshot displays the CubeGUI-4.4.3 interface for a performance profile. The window title is "CubeGUI-4.4.3: scorep\_tea\_leaf\_baseline\_8x12\_sum/profile.cubex <@jrl11>". The interface is divided into three main panels:

- Metric tree (Left):** Shows a list of performance metrics. The "8633.39 Time (sec)" metric is highlighted in blue. Other metrics include Visits (occ), Minimum Inclusive Time (sec), Maximum Inclusive Time (sec), bytes\_put (bytes), bytes\_get (bytes), ALLOCATION\_SIZE (bytes), DEALLOCATION\_SIZE (bytes), bytes\_leaked (bytes), maximum\_heap\_memory\_allocated, bytes\_sent (bytes), and bytes\_received (bytes).
- Call tree (Middle):** Shows a hierarchical view of the call paths. The path "0.03 MAIN\_ > 0.27 !\$omp parallel @tea\_leaf.f90:45 > 3.30 initialise\_ > 0.00 diffuse\_ > 0.06 set\_field\_module.set\_field\_ > 0.01 timestep\_module.timestep\_ > 0.75 tea\_leaf\_module.tea\_leaf\_ > 0.26 timer\_ > 115.14 update\_halo\_module.update\_halo\_ > 6.36 tea\_leaf\_kernel\_cg\_module.tea\_leaf\_kernel\_init\_cg\_ > 20.78 tea\_module.tea\_allsum\_ > 0.76 tea\_leaf\_kernel\_cheby\_module.tea\_leaf\_kernel\_ch\_ > 1.24 tea\_leaf\_kernel\_cg\_module.tea\_leaf\_kernel\_solve\_ > 1.69 !\$omp parallel @tea\_leaf\_cg.f90:186 > 3421.11 !\$omp do @tea\_leaf\_cg.f90:187" is highlighted in blue. A callout box points to this path with the text: "Selection updates metric values shown in columns to the right".
- System tree (Right):** Shows a hierarchical view of the system components. The path "0.00 machine Linux > 0.00 node jrc1531 > 0.00 MPI Rank 0" is highlighted in blue. Other components include MPI Ranks 1 through 7, Master thread, and OMP threads 1 through 11.

At the bottom of the interface, there are three columns of performance data corresponding to the selected paths:

Metric	Value	Percentage
Time (sec)	8633.39	100.00%
Time (sec)	3421.11	39.63%
Time (sec)	0.00	0.00%

Selection updates  
metric values shown  
in columns to the right

# Multiple selection

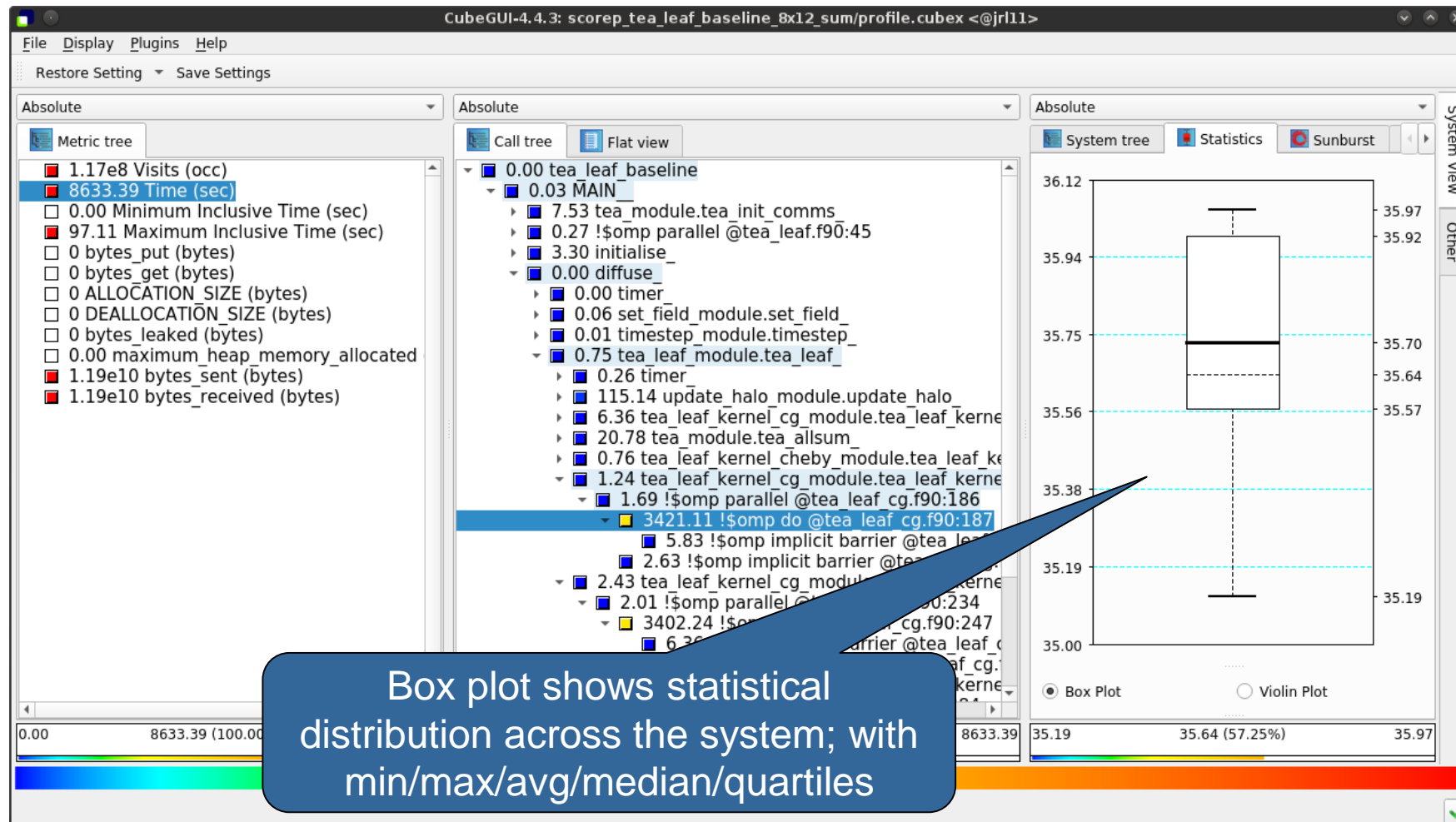
The screenshot displays the CubeGUI-4.4.3 interface for a performance profile. The window title is "CubeGUI-4.4.3: scorep\_tea\_leaf\_baseline\_8x12\_sum/profile.cubex <@jrl11>". The interface is divided into three main panels:

- Metric tree (left):** Shows various performance metrics. The "8633.39 Time (sec)" metric is highlighted in blue.
- Call tree (middle):** Shows a hierarchical view of the application's execution. Several nodes are selected with blue highlights, including:
  - 0.75 tea\_leaf\_module.tea\_leaf\_
  - 1.24 tea\_leaf\_kernel\_cg\_module.tea\_leaf\_kernel\_solve\_
  - 1.69 !\$omp parallel @tea leaf cg.f90:186
  - 3421.11 !\$omp do @tea leaf cg.f90:187
  - 2.63 !\$omp implicit barrier @tea leaf cg.f90:200
  - 2.43 tea\_leaf\_kernel\_cg\_module.tea\_leaf\_kernel\_solve\_
  - 2.01 !\$omp parallel @tea leaf cg.f90:234
  - 3402.24 !\$omp do @tea leaf cg.f90:247
  - 2.72 !\$omp implicit barrier @tea leaf cg.f90:257
  - 2.04 tea\_leaf\_kernel\_cg\_module.tea\_leaf\_kernel\_solve\_
  - 1.54 !\$omp parallel @tea leaf cg.f90:284
  - 1580.11 !\$omp do @tea leaf cg.f90:294
  - 40.82 !\$omp implicit barrier @tea leaf cg.f90:302
  - 3.24 !\$omp implicit barrier @tea leaf cg.f90:302
  - 1.37 tea\_leaf\_kernel\_module.tea\_leaf\_kernel\_finalise\_
  - 0.25 field\_summary\_
- System tree (right):** Shows the system hierarchy, including nodes and MPI ranks. The "0.00 machine Linux" node is expanded, showing various MPI ranks and threads.

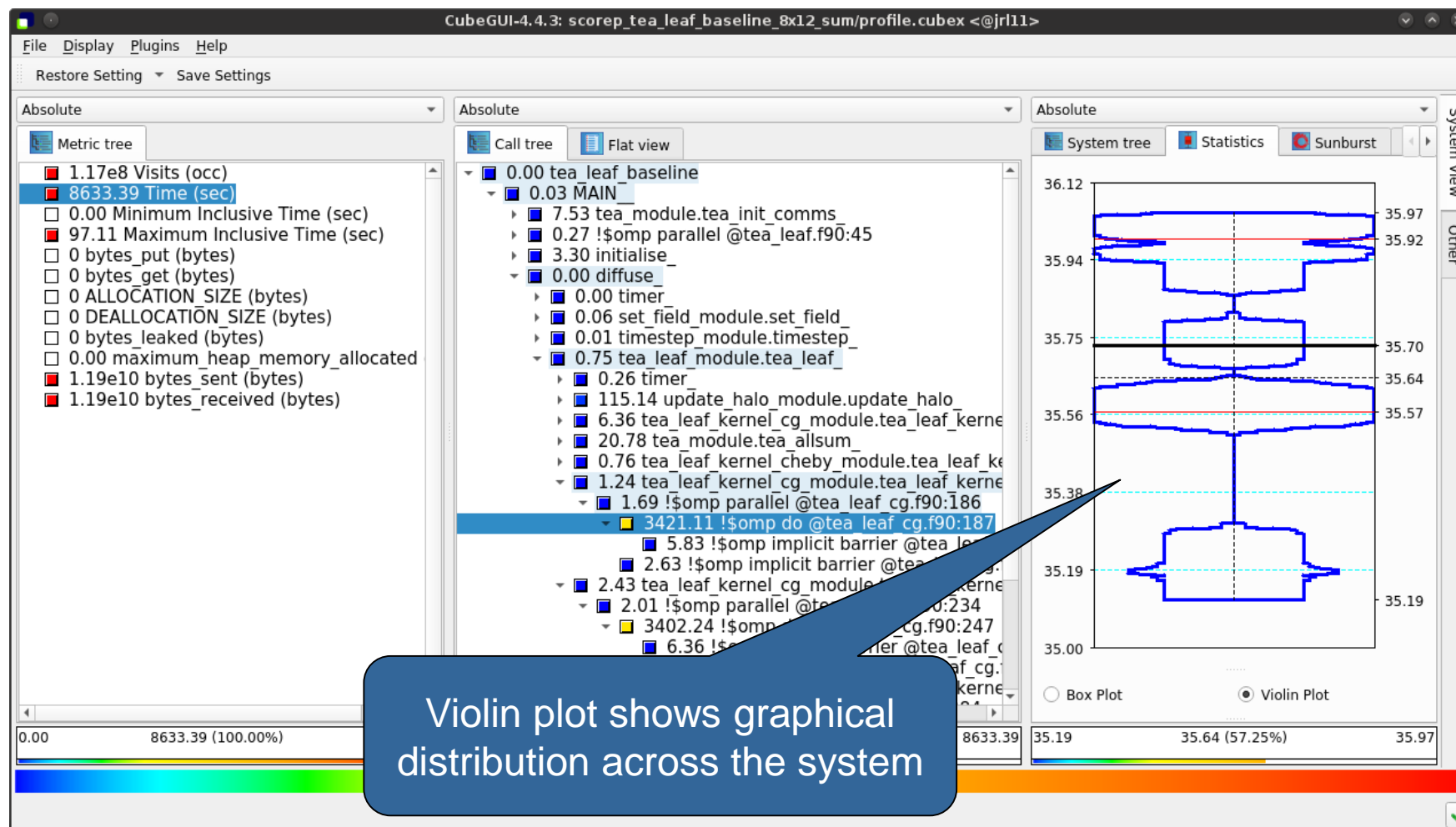
A blue callout box with a white border and a blue gradient background points to the selected nodes in the Call tree. The text inside the callout reads: "Select multiple nodes with Ctrl-click".

At the bottom of the interface, there are three progress bars showing the percentage of nodes selected in each view: Metric tree (100.00%), Call tree (97.34%), and System tree (0.00%).

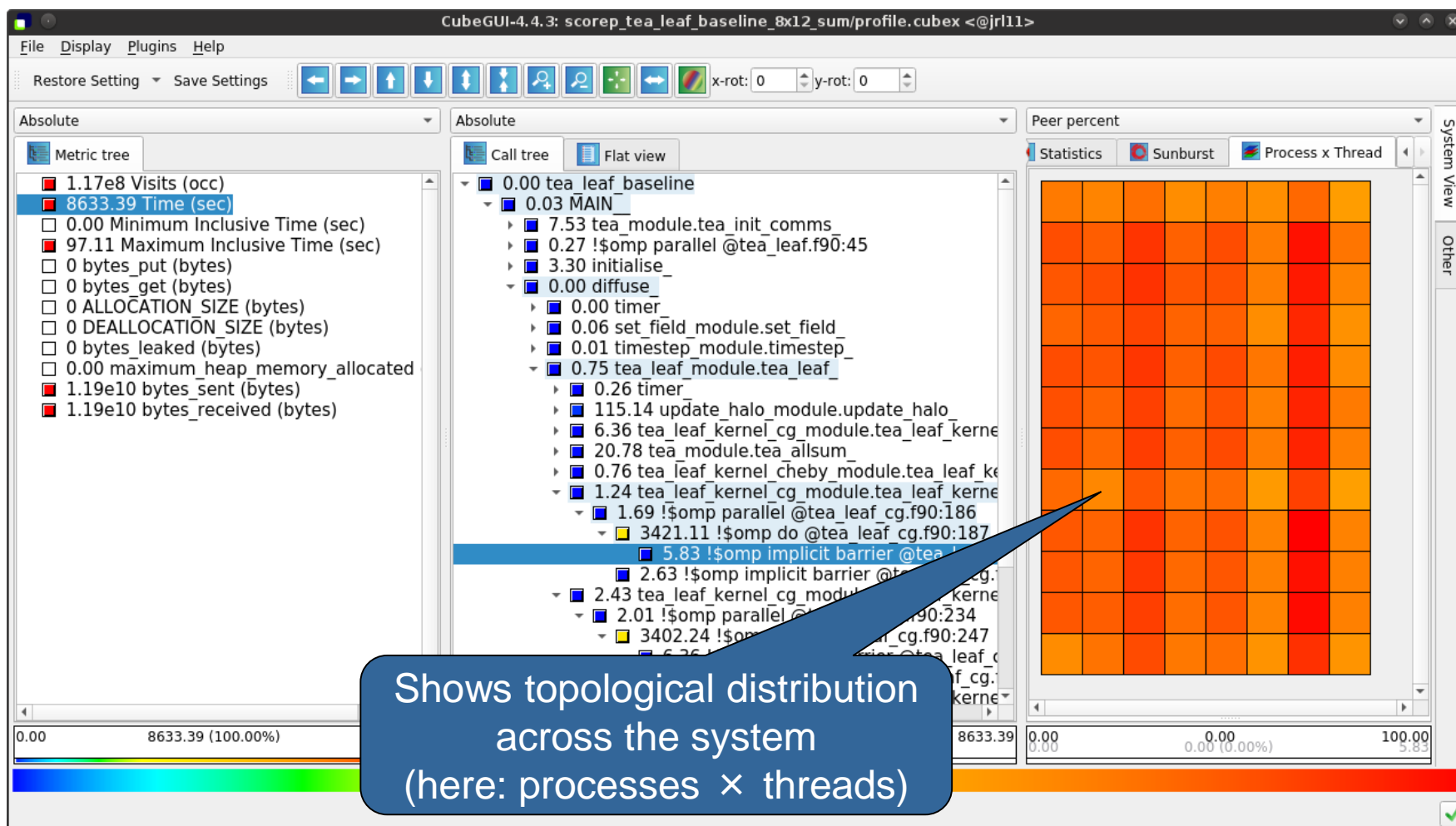
# Box plot view



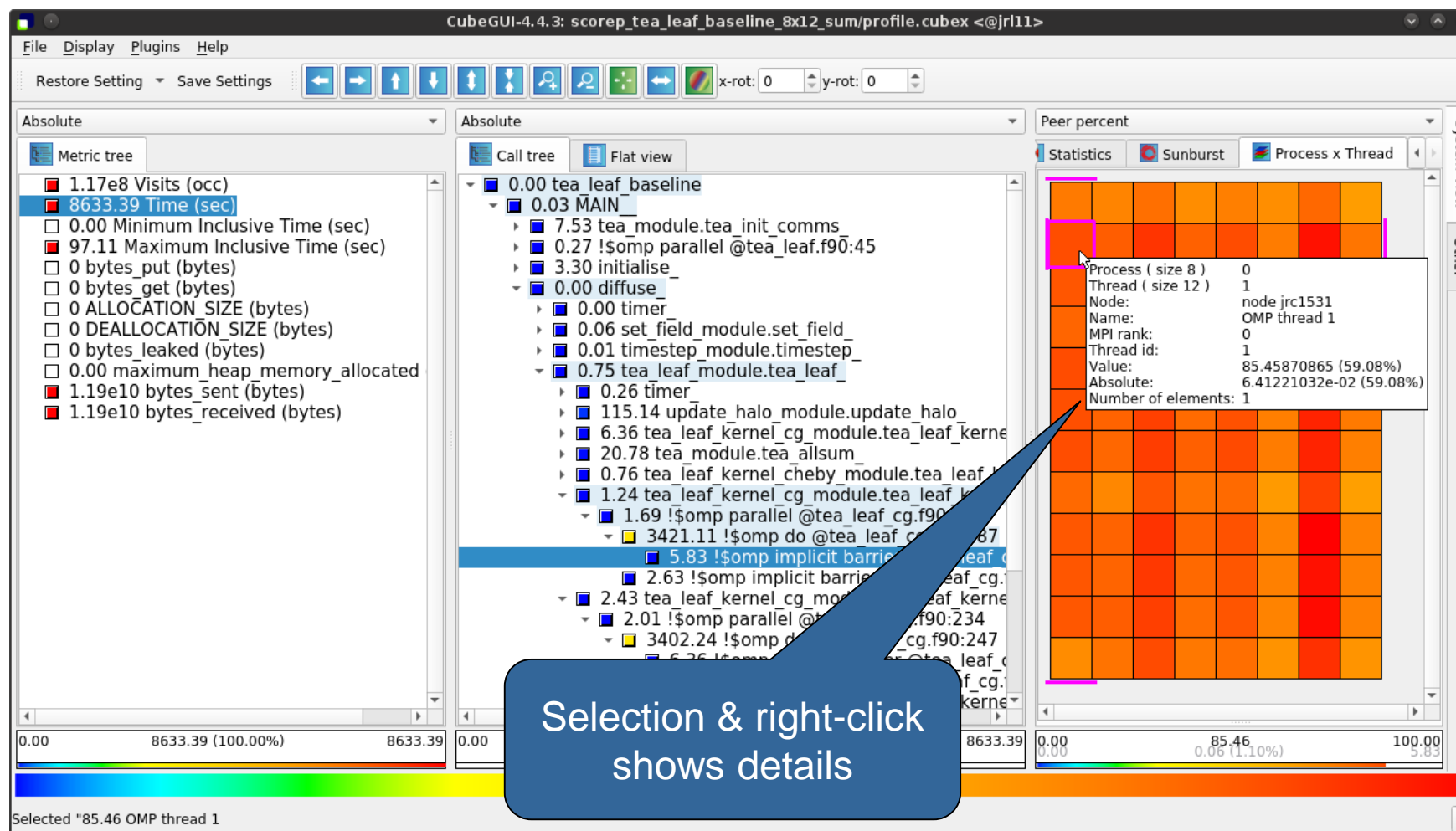
# Violin plot view



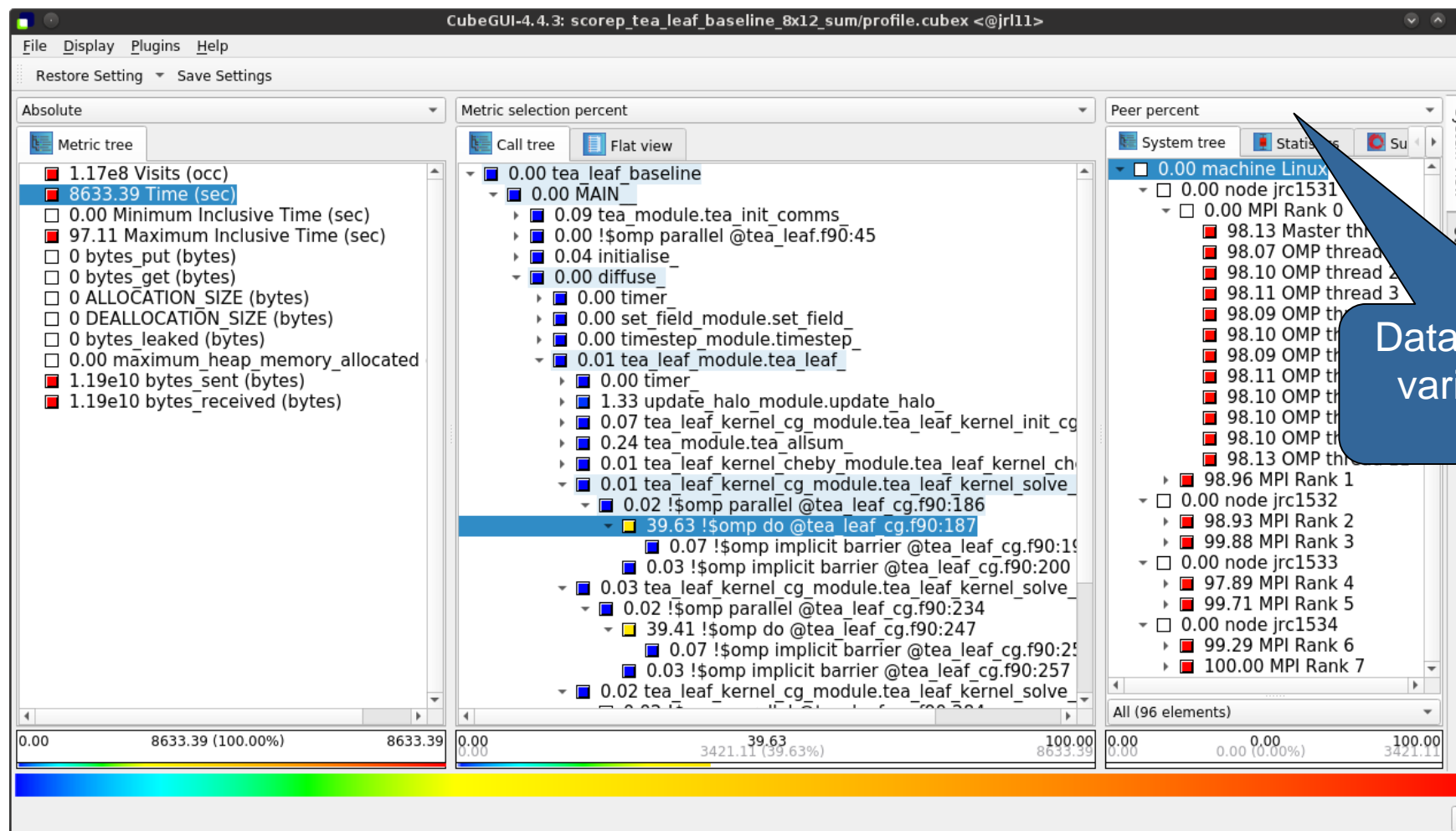
# Topology view



# Topology view (cont.)



# 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



# Source-code view via context menu

The screenshot displays the CubeGUI-4.4.3 interface with three main panels: Metric tree, Call tree, and System tree. The Call tree panel is active, showing a hierarchical view of the execution profile. A context menu is open over the item `3421.11 !$omp do @tea_leaf_cg.f90:18`. The menu options include: Info, Documentation, Set as loop, Expand/collapse, Hiding, Cut call tree, Find items, Clear found items, Sort tree items..., Min/max values, Copy to clipboard, Show max severity information, and Mark this item. A blue callout box with a speech bubble points to the context menu, containing the text: "Right-click opens context menu".

Right-click opens context menu

# Source-code view

The screenshot displays the CubeGUI-4.4.3 interface for the file `scorep_tea_leaf_baseline_8x12_sum/profile.cubex`. The 'Source' tab is selected, showing Fortran code for the subroutine `tea_leaf_kernel_solve_cg_fortran_calc_ur(x, m)`. The code includes OpenMP directives and a nested loop for a reduction operation. The 'Call tree' shows the execution flow, with the `!$omp do @tea_leaf_cg.f90` node highlighted, indicating it accounts for 3421.11 (39.63%) of the total execution time. The 'Metric tree' on the left lists various performance metrics, with `8633.39 Time (sec)` being the most significant.

Select  
"Source" tab

Note:

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

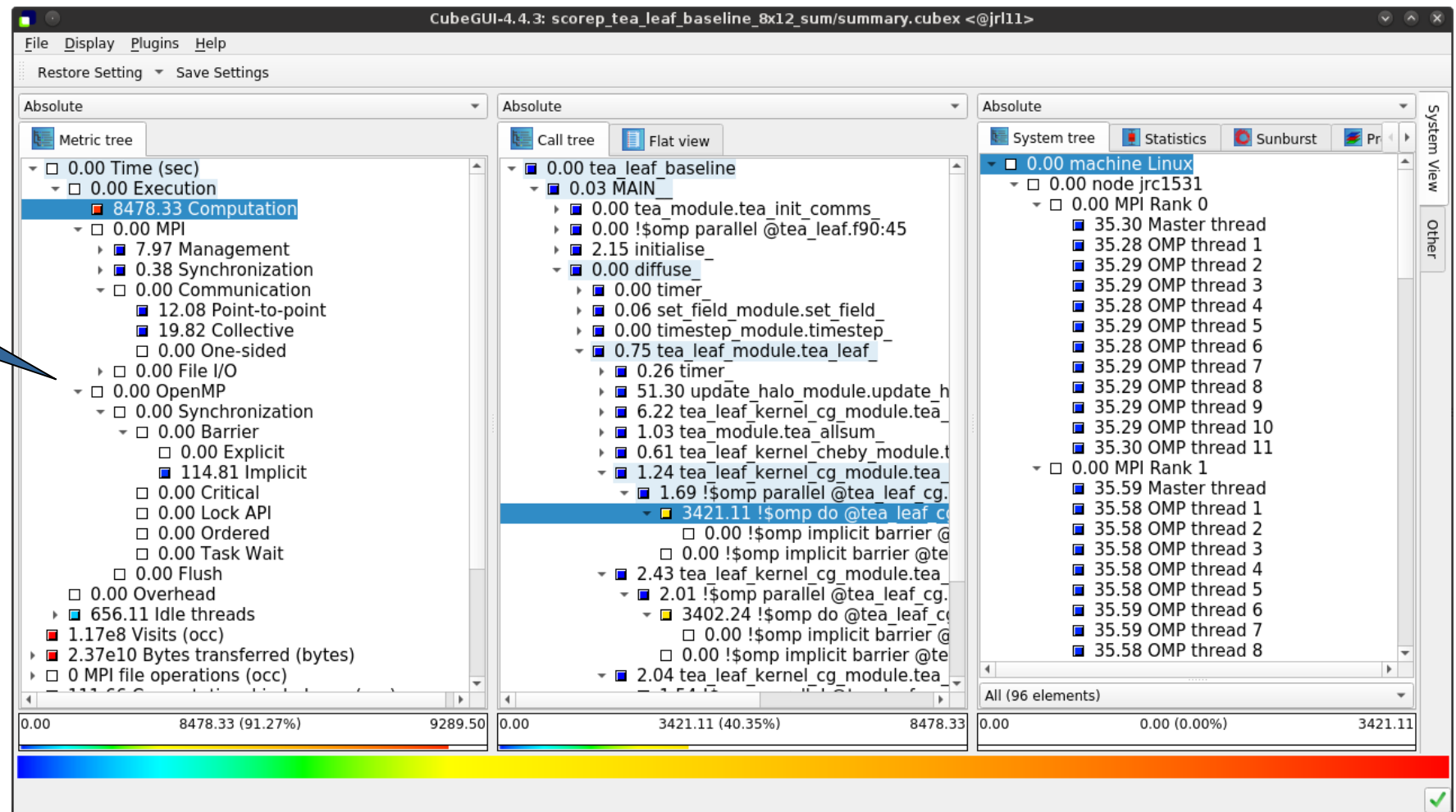
# Context-sensitive help

The screenshot displays the CubeGUI-4.4.3 interface with the 'Help' menu open. The 'What's This?' option is selected, and a blue callout box points to it with the text: 'Context-sensitive help available for all GUI items'. The main window shows a hierarchical tree of metrics and system components. The selected metric is '39.63 !\$omp do @tea\_leaf\_cg.f90:187'. The interface includes a 'Metric tree' on the left, a 'System tree' on the right, and a 'Statistics' panel at the bottom. A color bar at the bottom indicates the range of values for the selected metric.

Change into help mode for display components

# Post-processed summary analysis report

Split base metrics into more specific metrics



## 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/cubegui/CubeUserGuide.pdf`
- Contact:
  - mailto: [scalasca@fz-juelich.de](mailto:scalasca@fz-juelich.de)



## Case study: TeaLeaf

---



# TeaLeaf

---

- HPC mini-app developed by the UK Mini-App Consortium
  - Solves the linear 2D heat conduction equation on a spatially decomposed regular grid using a 5 point stencil with implicit solvers
  - Part of the Mantevo 3.0 suite
  - Available on GitHub: <http://uk-mac.github.io/TeaLeaf/>
- Measurements of TeaLeaf reference v1.0 taken on (previous) Jureca cluster @ JSC
  - Using Intel 19.0.3 compilers, Intel MPI 2019.3, and Score-P 5.0
  - Run configuration
    - 8 MPI ranks with 12 OpenMP threads each
    - Distributed across 4 compute nodes (2 ranks per node)
    - Test problem "5": 4000 × 4000 cells, CG solver

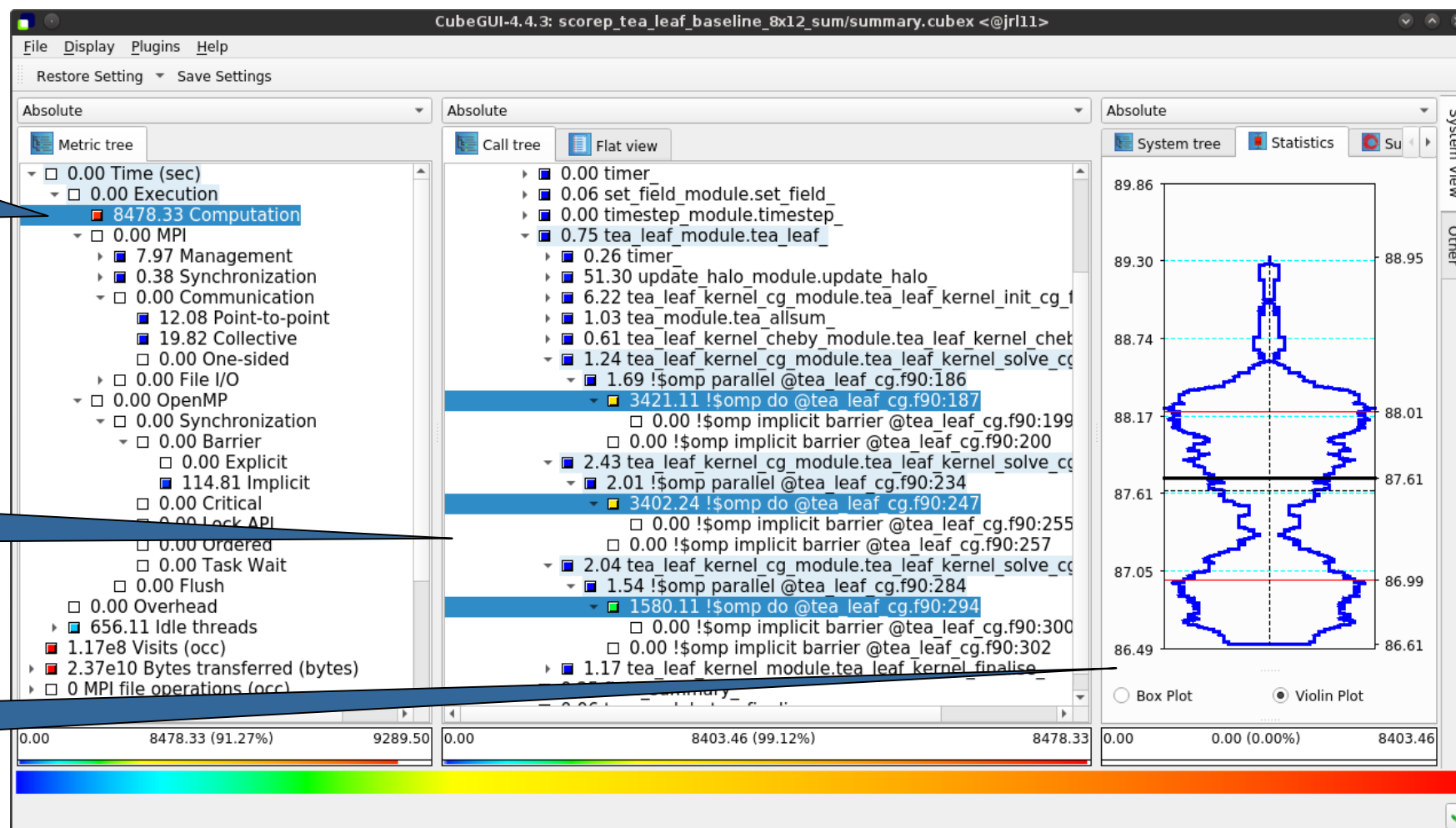


# TeaLeaf summary report analysis (I)

91% of the execution time is computation...

...almost entirely spent in 3 OpenMP do loops...

...with a slight imbalance across ranks & threads

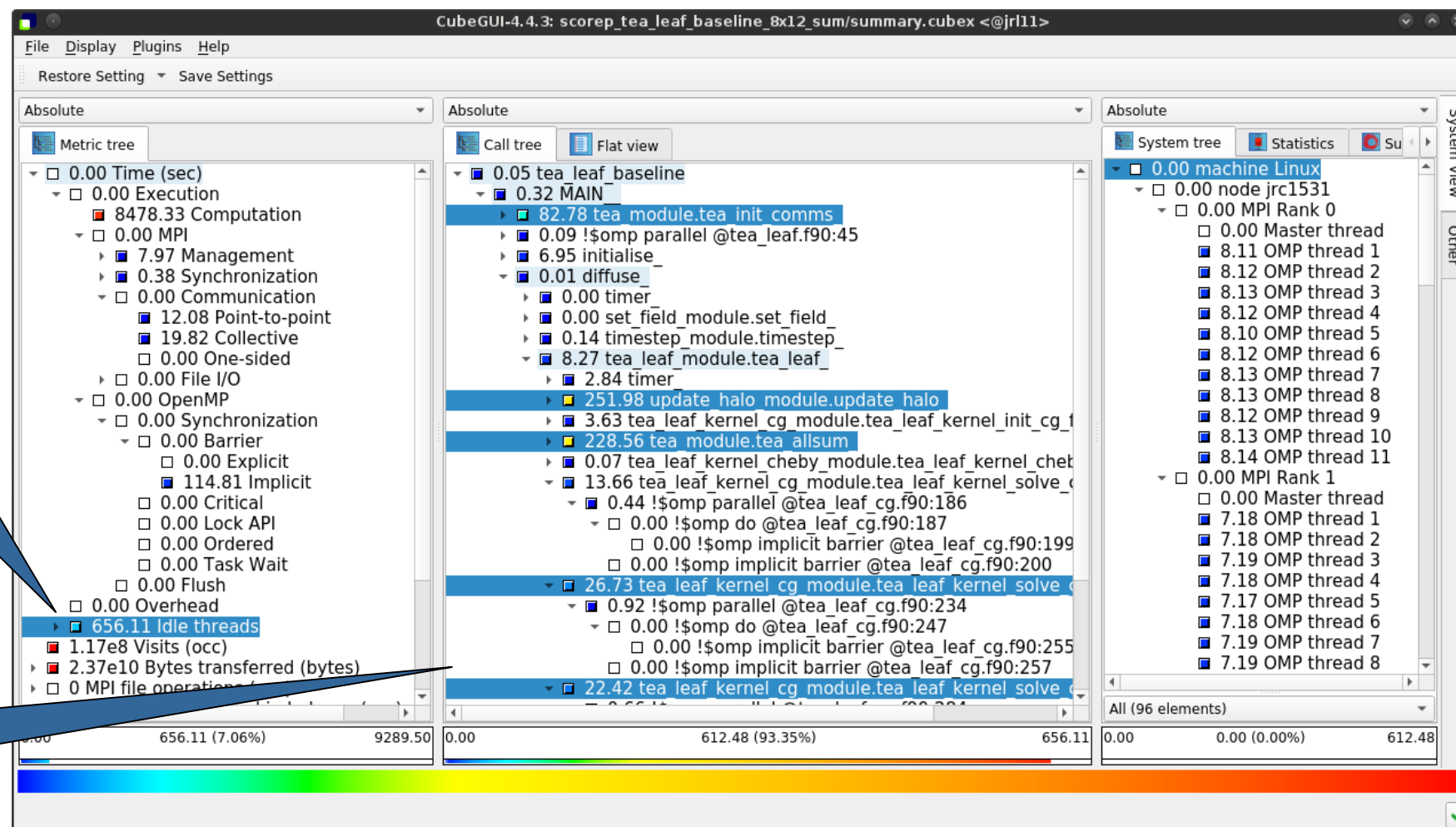




# TeaLeaf summary report analysis (II)

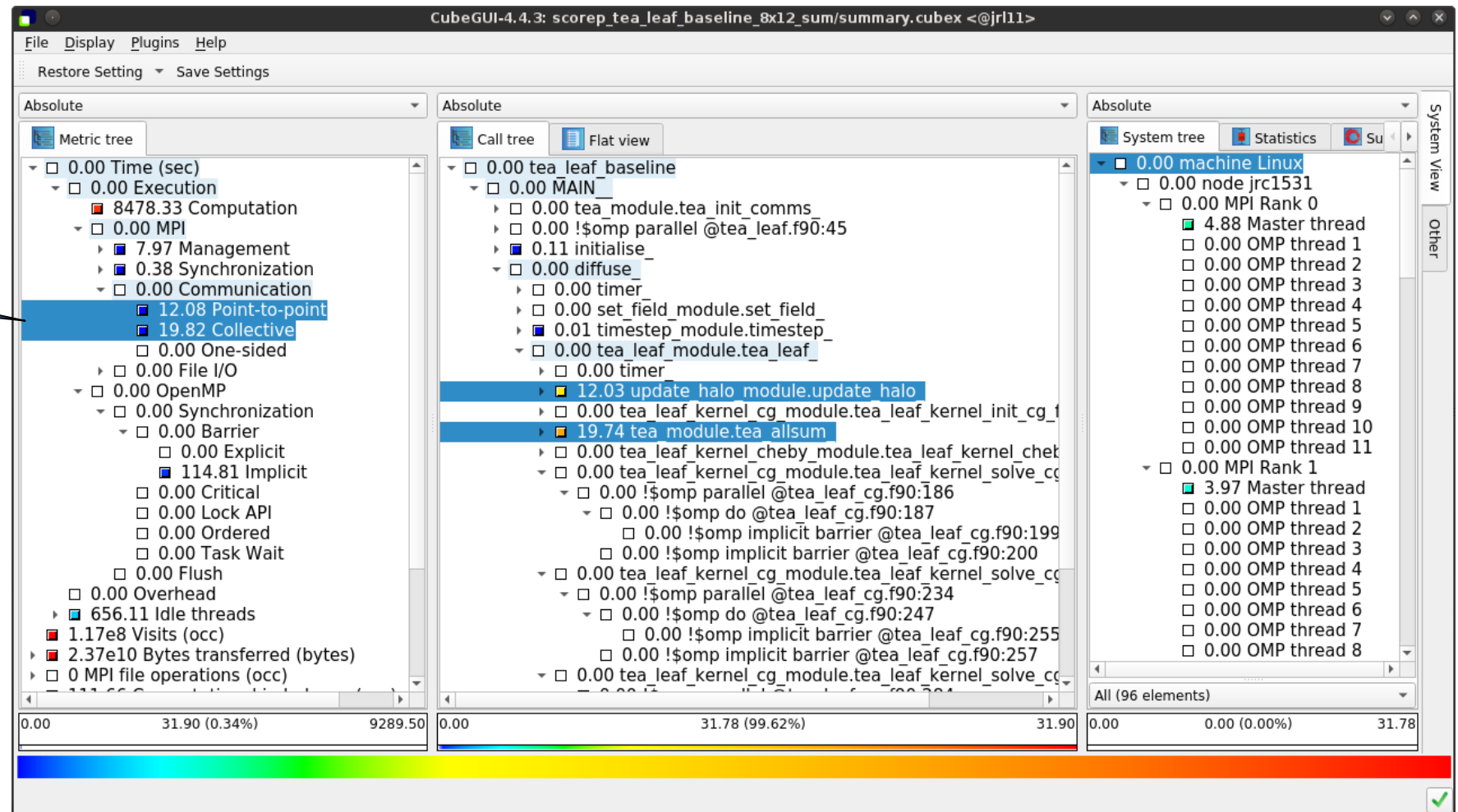
7% of the execution time are lost due to idle threads...

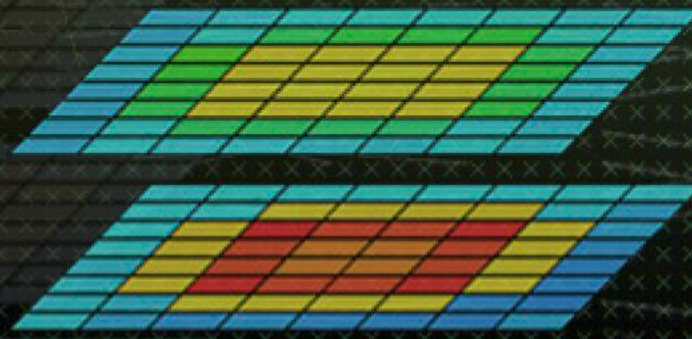
...in non-OpenMP parallelized code regions



# TeaLeaf summary report analysis (III)

MPI communication time is negligible (0.34%); communication is only on the master threads (MPI\_THREAD\_FUNNELED)



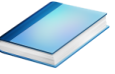


## Reference material



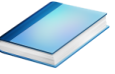
## Derived metrics

---



- 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



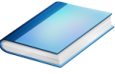
Collection of derived metrics

Parameters of the derived metric

CubePL expression

The screenshot shows the Cube GUI interface with a dialog box for creating a new derived metric. The dialog is titled "Create new metric as a child of metric". It features several input fields and a description area. The "Select metric from collection" dropdown is set to "Average execution time (kenobi)". The "Derived metric type" is "Postderived metric". The "Display name" is "Average visit time", the "Unique name" is "avg\_visit\_time", the "Data type" is "DOUBLE", and the "Unit of measurement" is "sec". The "Description" field contains the text: "Calculates average time of region execution per visit. Autor is Michael Knobloch." At the bottom of the dialog, there is a "Calculation" section with a checked checkbox and a text area containing the CubePL expression: `metric::time()/metric::visits(e)`. The background shows a performance profile with a metric tree on the left and a call tree in the center.

# Example: FLOPS based on PAPI\_FP\_OPS and time



Cube-4.3.1: scorep\_8x4\_sum/profile.cubex (on froggy1)

File Display Plugins Help  
Restore Setting Save Settings

**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:

URL:

Description:

Calculation Calculation Init Aggregation "+" Aggregation "-"

`metric::PAPI_FP_OPS()/metric::time()`

Edit metric Cancel

Share this metric with SCALASCA group

**Absolute** Metric tree

- 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**

**Absolute** Call tree Flat view

- 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

**Absolute** System tree Barplot Heatmap

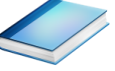
    - machine Linux
      - node frog6
        - MPI Rank 0
          - 1.17e9 Master thread
          - 9.43e8 OMP thread 1
          - 9.47e8 OMP thread 2
          - 9.47e8 OMP thread 3
        - MPI Rank 1
          - 1.17e9 Master thread
          - 9.87e8 OMP thread 1
          - 9.68e8 OMP thread 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

All (32 elements)

0.00 1.84e9 (100.00%) 1.84e9 0.00 9.65e8 (-0.00%) -12858016489314434.00 0.00... -179769313486231570814527423731704356798070...

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

# CUBE algebra utilities



- Extracting solver sub-tree from analysis report

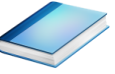
```
% cube_cut -r '<<ITERATION>>' scorep_bt-mz_C_32x4_sum/profile.cubex  
Writing cut.cubex... done.
```

- Calculating difference of two reports

```
% cube_diff scorep_bt-mz_C_32x4_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



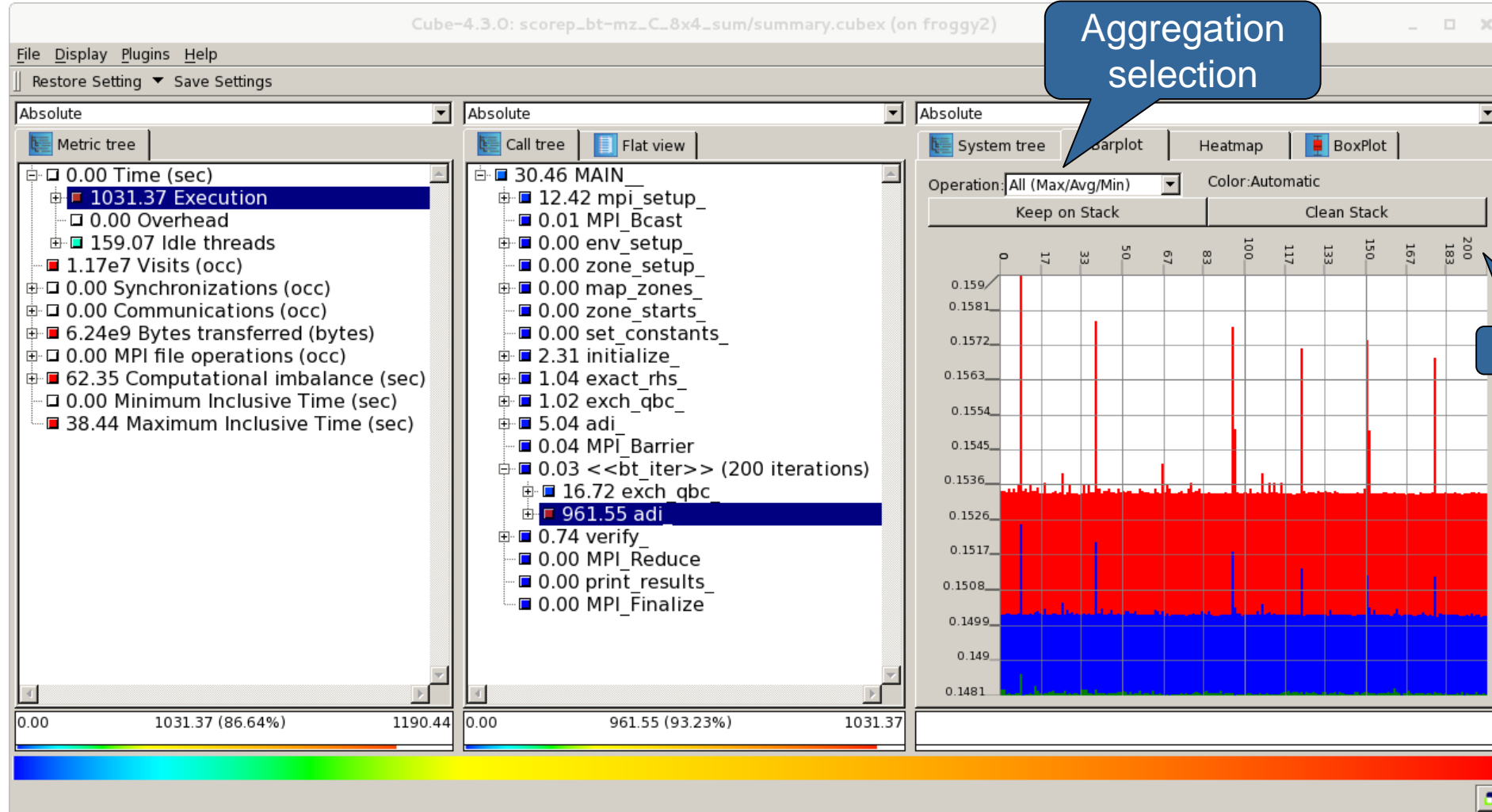
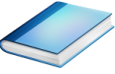
- Show time dependent behavior by “unrolling” iterations
- Preparations:
  - Mark loop body by using Score-P instrumentation API in your source code

```
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

