

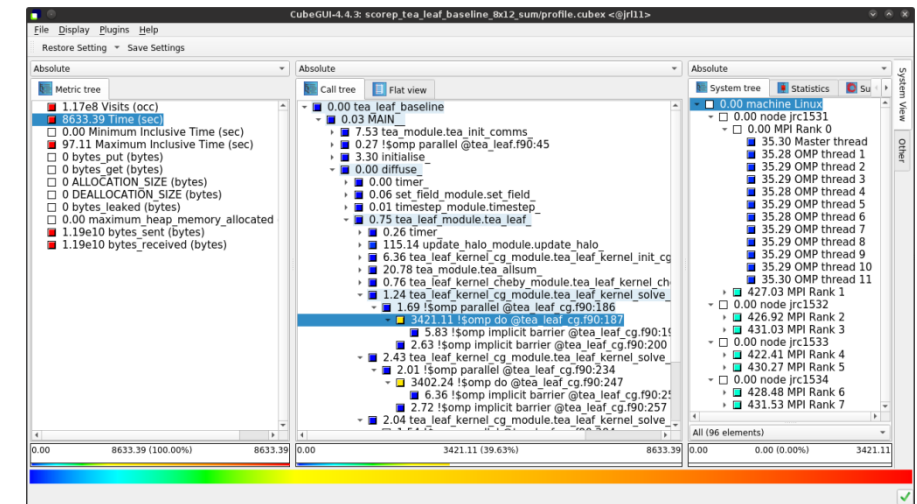
Analysis report examination with Cube

Brian Wylie
Jülich Supercomputing Centre



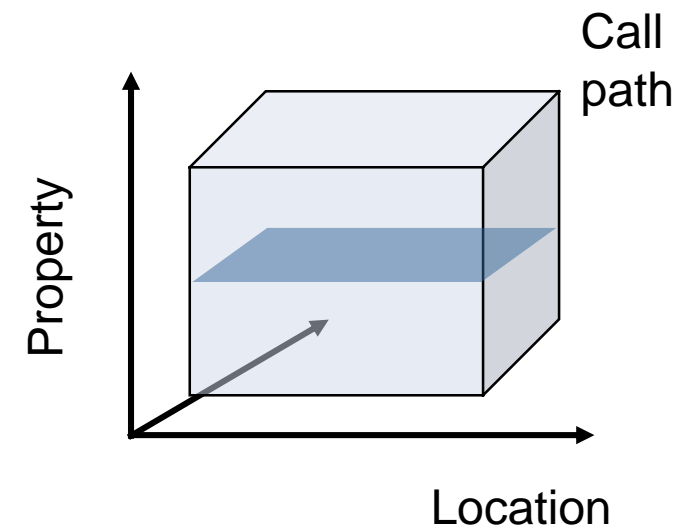
Cube

- 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 components
 - Can be installed independently of Score-P, e.g., on laptop or desktop
 - Latest release: Cube v4.4.x (March 2019)



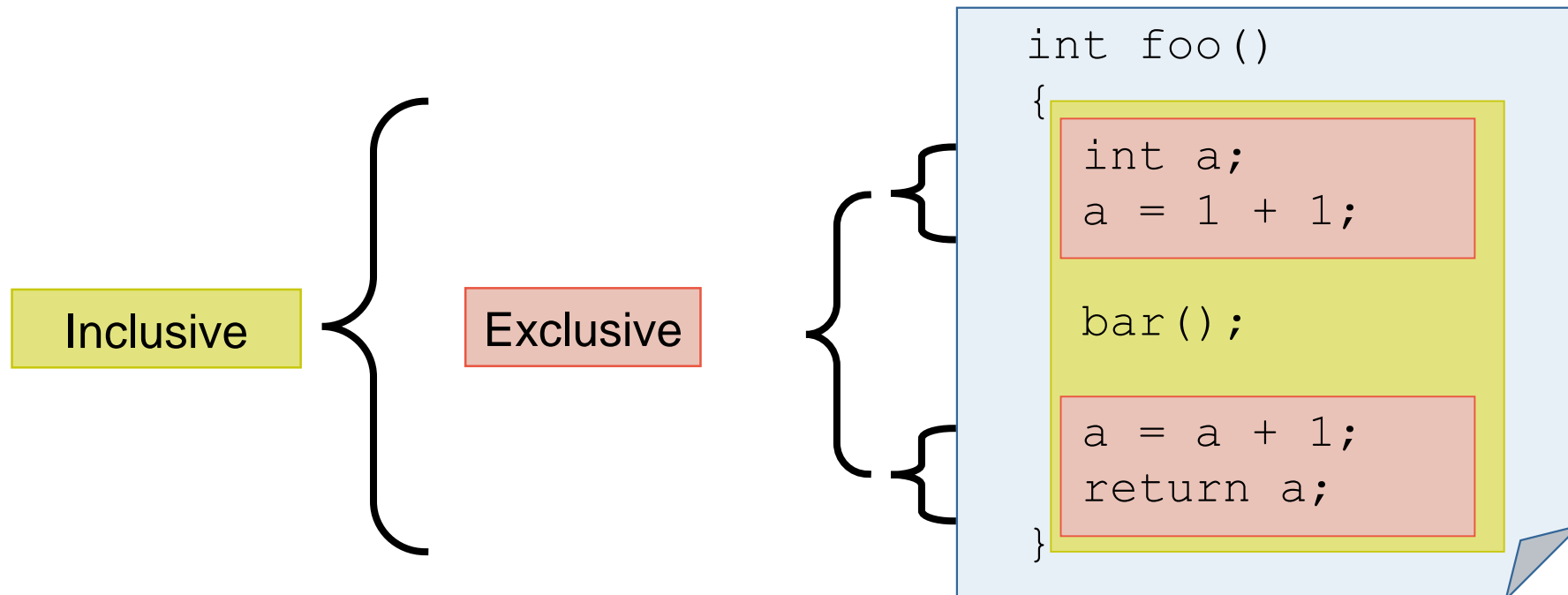
Analysis presentation and exploration

- 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



Case study: TeaLeaf



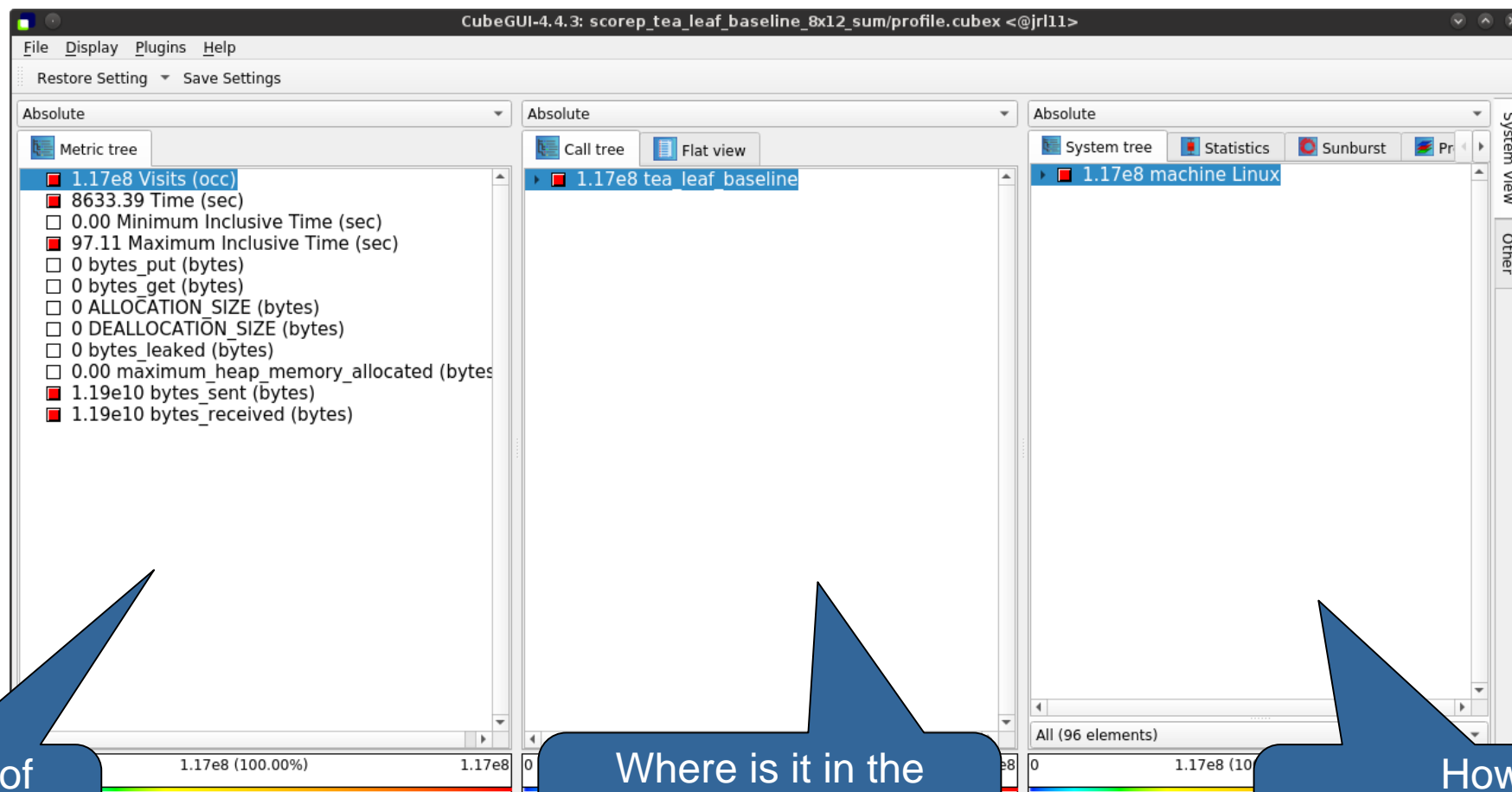
Case study: TeaLeaf

- 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 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



```
% cp -r /p/scratch/share/VI-HPS/examples/TeaLeaf .  
% cd TeaLeaf  
% cube scorep_tea_leaf_baseline_8x12_sum/profile.cubex  
[GUI showing summary analysis report]
```

Score-P analysis report exploration (opening view)



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

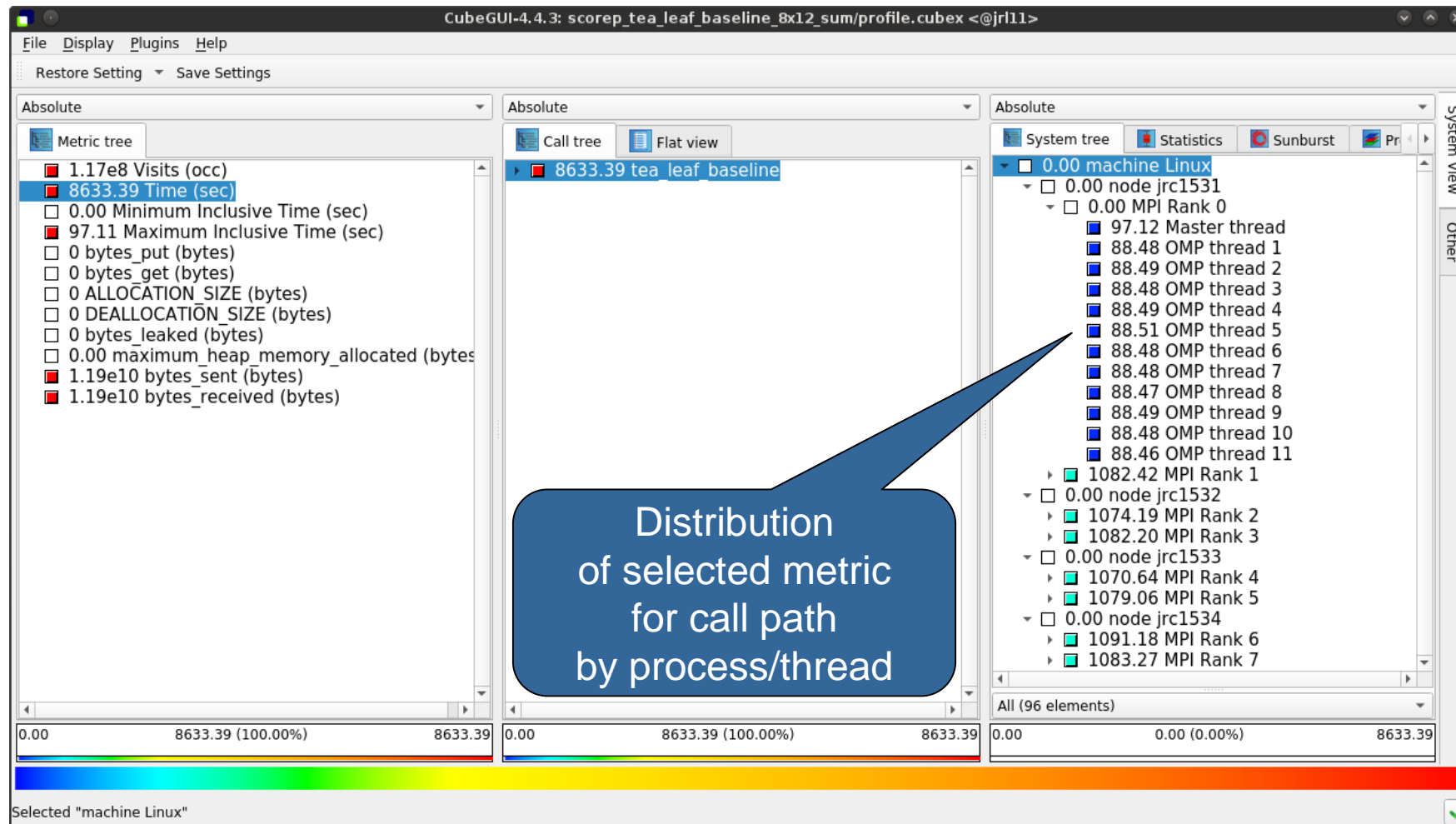
All (96 elements)

0.00 8633.39 (100.00%) 8633.39 0.00 8633.39 (100.00%) 8633.39 0.00 8633.39 (100.00%) 8633.39

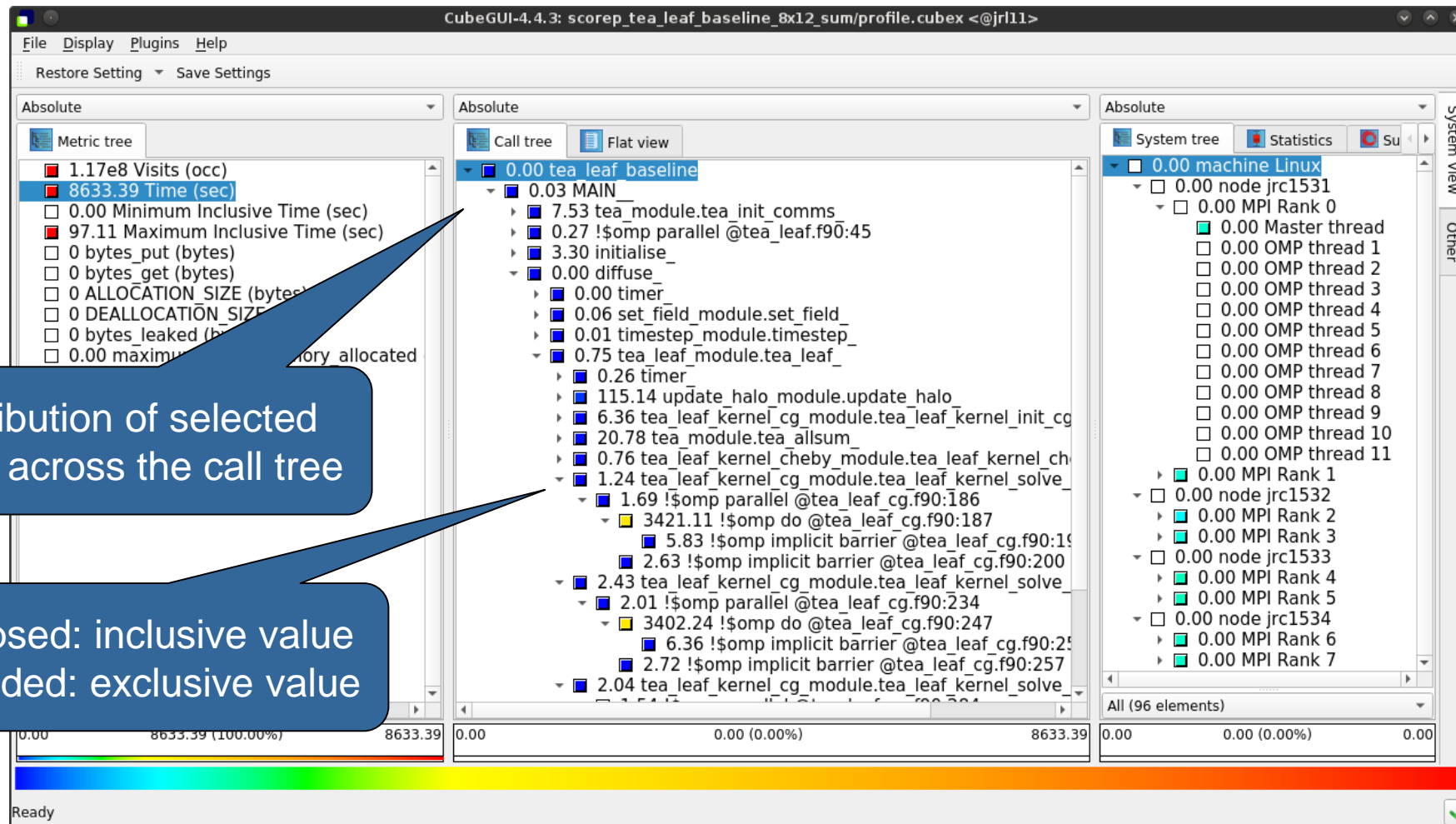
Selected "Time"

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 main window is titled "CubeGUI-4.4.3: scorep_tea_leaf_baseline_8x12_sum/profile.cubex <@jrl11>". It features three main panels:

- Metric tree (Left):** Lists various performance metrics. The "8633.39 Time (sec)" metric is highlighted in blue.
- Call tree (Center):** Shows a hierarchical view of the application's execution. The path "0.03 MAIN_ > 0.27 !\$omp parallel @tea_leaf.f90:45 > 0.00 diffuse_ > 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" is selected and highlighted in blue.
- System tree (Right):** Shows the system's execution tree. The path "0.00 machine Linux > 0.00 node jrc1531 > 0.00 MPI Rank 0" is selected.

At the bottom, a color-coded bar indicates the relative execution time of the selected path. The bar is divided into three segments: a blue segment (0.00), a green segment (8633.39 (100.00%)), and a red segment (8633.39). The total time is 8633.39. The selected path's time is 3421.11 (39.63%), and the remaining time is 8633.39.

A callout box with a blue background and white text points to the selected path in the call tree, stating: "Selection updates metric values shown in columns to the right".

Multiple selection

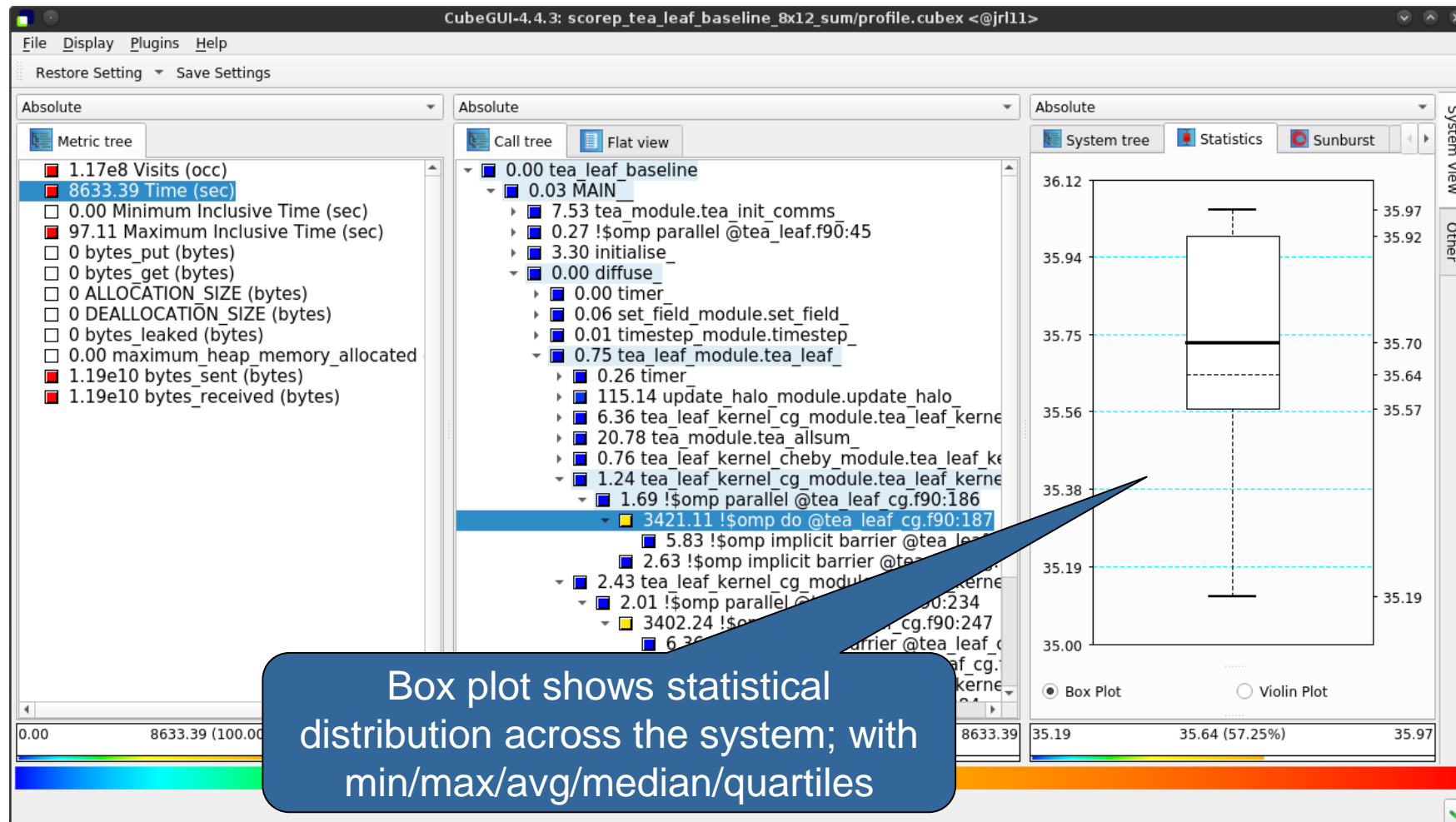
The screenshot displays the CubeGUI-4.4.3 interface for a performance profile. The window title is "CubeGUI-4.4.3: scorep_tea_leaf_baseline_8x12_sum/profile.cubex <@jrl11>". The interface is divided into three main panels:

- 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
 - 5.83 !\$omp implicit barrier @tea leaf cg.f90:19
 - 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
 - 6.36 !\$omp implicit barrier @tea leaf cg.f90:25
 - 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.f90:3
 - 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 like "0.00 machine Linux", "0.00 node jrc1531", and various MPI Ranks and threads.

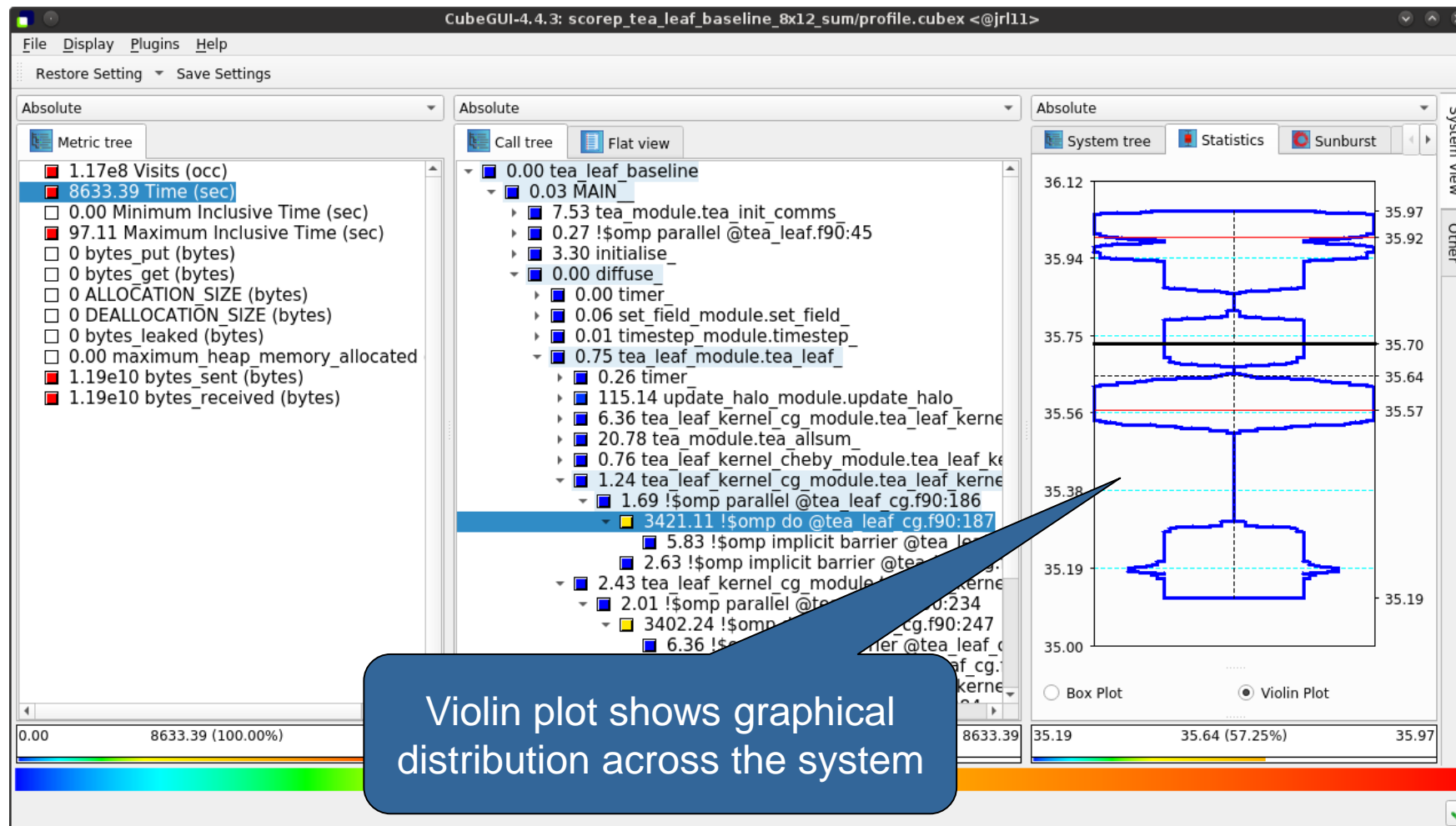
A blue callout box with a white border and a pointer to the selected nodes in the Call tree contains the text: "Select multiple nodes with Ctrl-click".

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

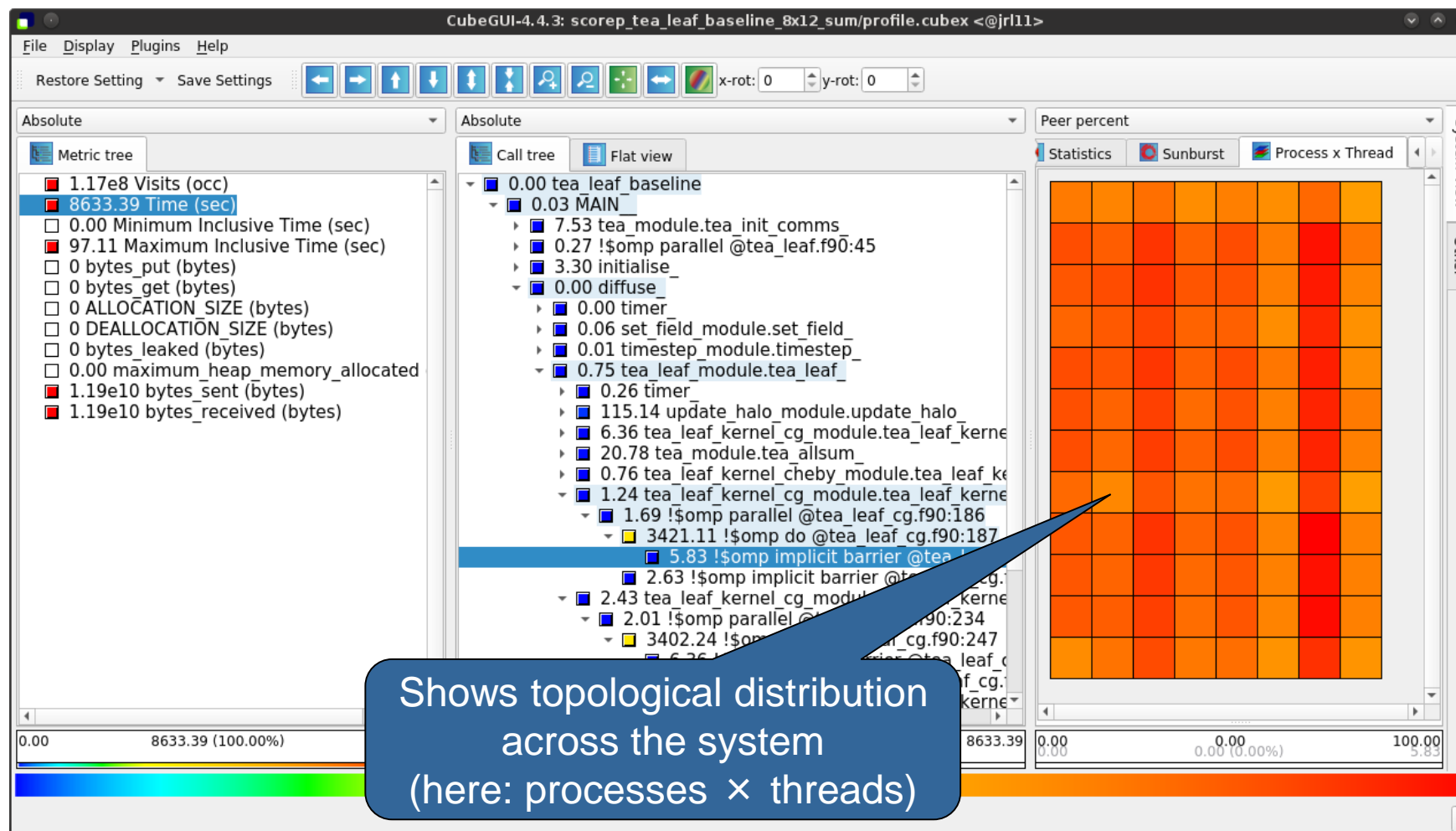
Box plot view



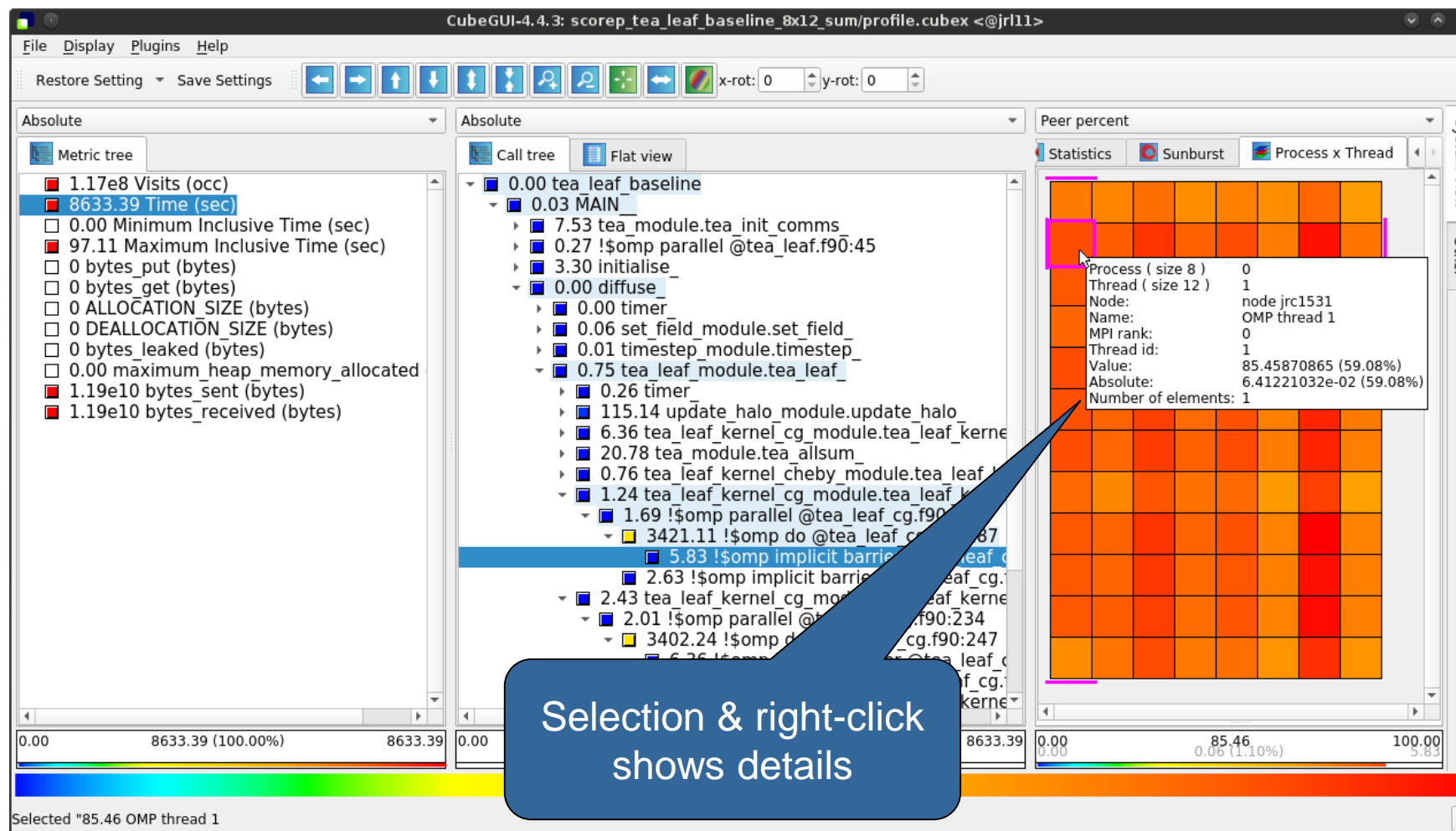
Violin plot view



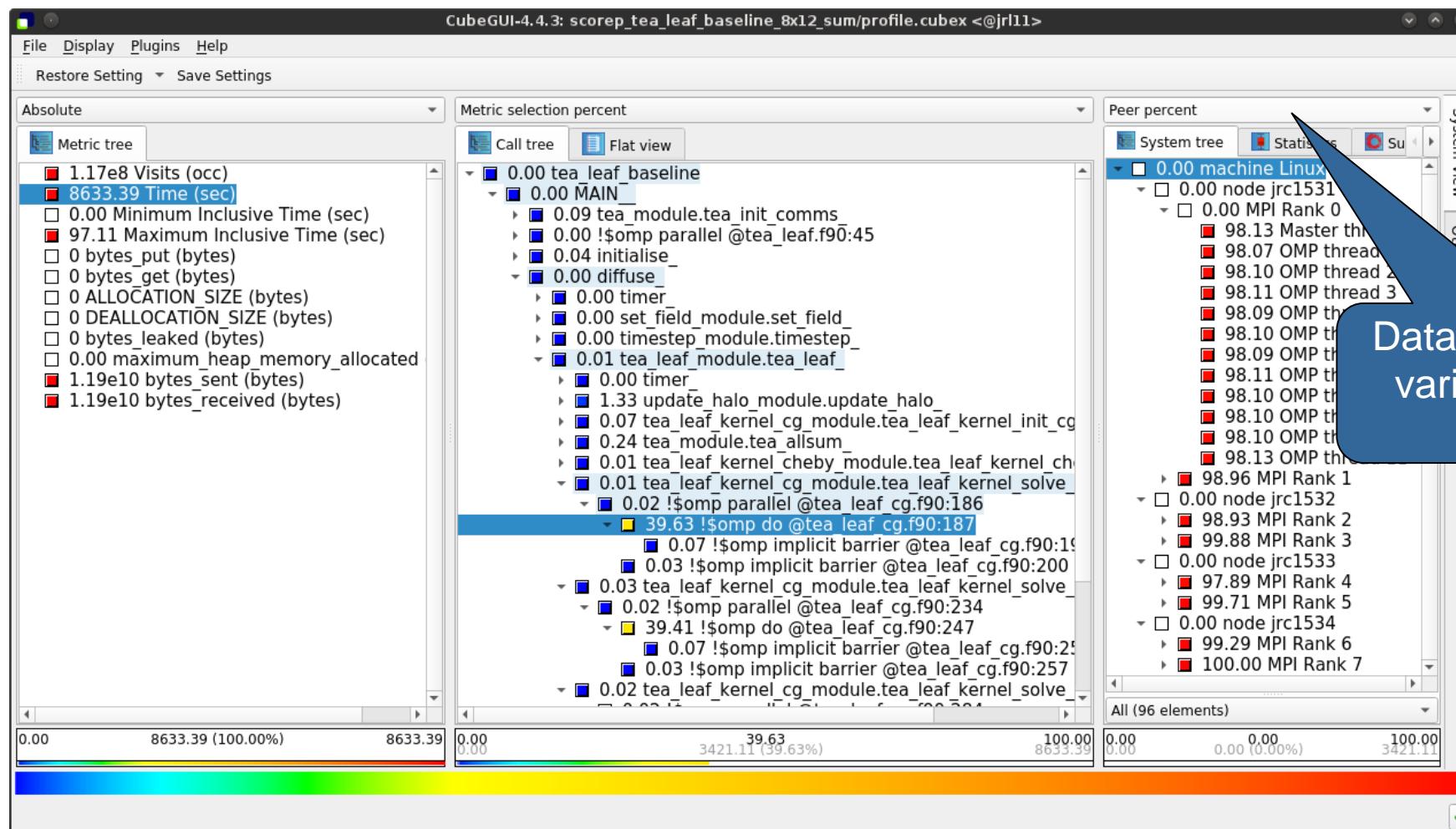
Topology view



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.f90:186`. 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

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
- 1.19e10 bytes_sent (bytes)
- 1.19e10 bytes_received (bytes)

Absolute

Call tree Flat view

- 0.00 tea_leaf_baseline
 - 0.03 MAIN_
 - 7.53 tea_module.tea_init_comms
 - 0.27 !\$omp parallel @tea_leaf.f90:45
 - 3.30 initialise_
 - 0.00 diffuse_
 - 0.00 timer_
 - 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_h
 - 6.36 tea_leaf_kernel_cg_module.tea_le
 - 20.78 tea_module.tea_allsum_
 - 0.76 tea_leaf_kernel_cheby_module.te
 - 1.24 tea_leaf_kernel_cg_module.tea_le
 - 1.69 !\$omp parallel @tea_leaf_cg.f
 - 3421.11 !\$omp do @tea_leaf_cg.f
 - 5.83 !\$omp implicit barrier @t
 - 2.63 !\$omp implicit barrier @tea
 - 2.43 tea_leaf_kernel_cg_module.tea_le
 - 2.01 !\$omp parallel @tea_leaf_cg.f
 - 3402.24 !\$omp do @tea_leaf_cg.f
 - 6.36 !\$omp implicit barrier @t
 - 2.72 !\$omp implicit barrier @tea
 - 2.04 tea_leaf_kernel_cg_module.tea_le
 - 2.54 timer_

Score-P Configuration Source Info

System View Other

```

170
171 IMPLICIT NONE
172
173 INTEGER(KIND=4):: x_min,x_max,y_min,y_max
174 REAL(KIND=8), DIMENSION(x_min-2:x_max+2,y_min-2:y
175 REAL(KIND=8), DIMENSION(x_min-2:x_max+2,y_min-2:y
176 REAL(KIND=8), DIMENSION(x_min-2:x_max+2,y_min-2:y
177 REAL(KIND=8), DIMENSION(x_min-2:x_max+2,y_min-2:y
178
179 REAL(KIND=8) :: rx, ry
180
181 INTEGER(KIND=4) :: j,k,n
182 REAL(kind=8) :: pw
183
184 pw = 0.0_08
185
186 !$OMP PARALLEL
187 !$OMP DO REDUCTION(+:pw)
188 DO k=y_min,y_max
189   DO j=x_min,x_max
190     w(j, k) = (1.0_8
191       + ry*(Ky(j, k+1) + Ky(j, k))
192       + rx*(Kx(j+1, k) + Kx(j, k))*p(j, k)
193       - ry*(Ky(j, k+1)*p(j, k+1) + Ky(j, k)*p(j, k-1)) &
194       - rx*(Kx(j+1, k)*p(j+1, k) + Kx(j, k)*p(j-1, k))
195     )
196     pw = pw + w(j, k)*p(j, k)
197   ENDDO
198 ENDDO
199 !$OMP END DO
200 !$OMP END PARALLEL
201
202 END SUBROUTINE tea_leaf_kernel_solve_cg_fortran_calc_w
203
204 SUBROUTINE tea_leaf_kernel_solve_cg_fortran_calc_ur(x m

```

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

33.39 0.00 3421.11 (39.63%) 8633.39

Select
"Source" tab

Context-sensitive help

The screenshot displays the CubeGUI-4.4.3 interface. The 'Help' menu is open, showing options like 'Getting started', 'User Guide', and 'What's This?'. A blue callout box points to the 'What's This?' option, stating: 'Context-sensitive help available for all GUI items'. The main window shows a 'Metric tree' on the left, a central 'Flat view' of the metric hierarchy, and a 'System tree' on the right. The 'Flat view' shows a hierarchy of metrics, with the selected item being '39.63 !\$omp do @tea_leaf_cg.f90:187'. The 'System tree' shows a hierarchy of nodes and MPI ranks. A color bar at the bottom indicates the range of values for the selected metric, from 0.00 to 100.00.

0.00 8633.39 (100.00%) 8633.39 0.00 39.63 (39.63%) 100.00 0.00 0.00 (0.00%) 100.00

Change into help mode for display components

Scalasca report post-processing

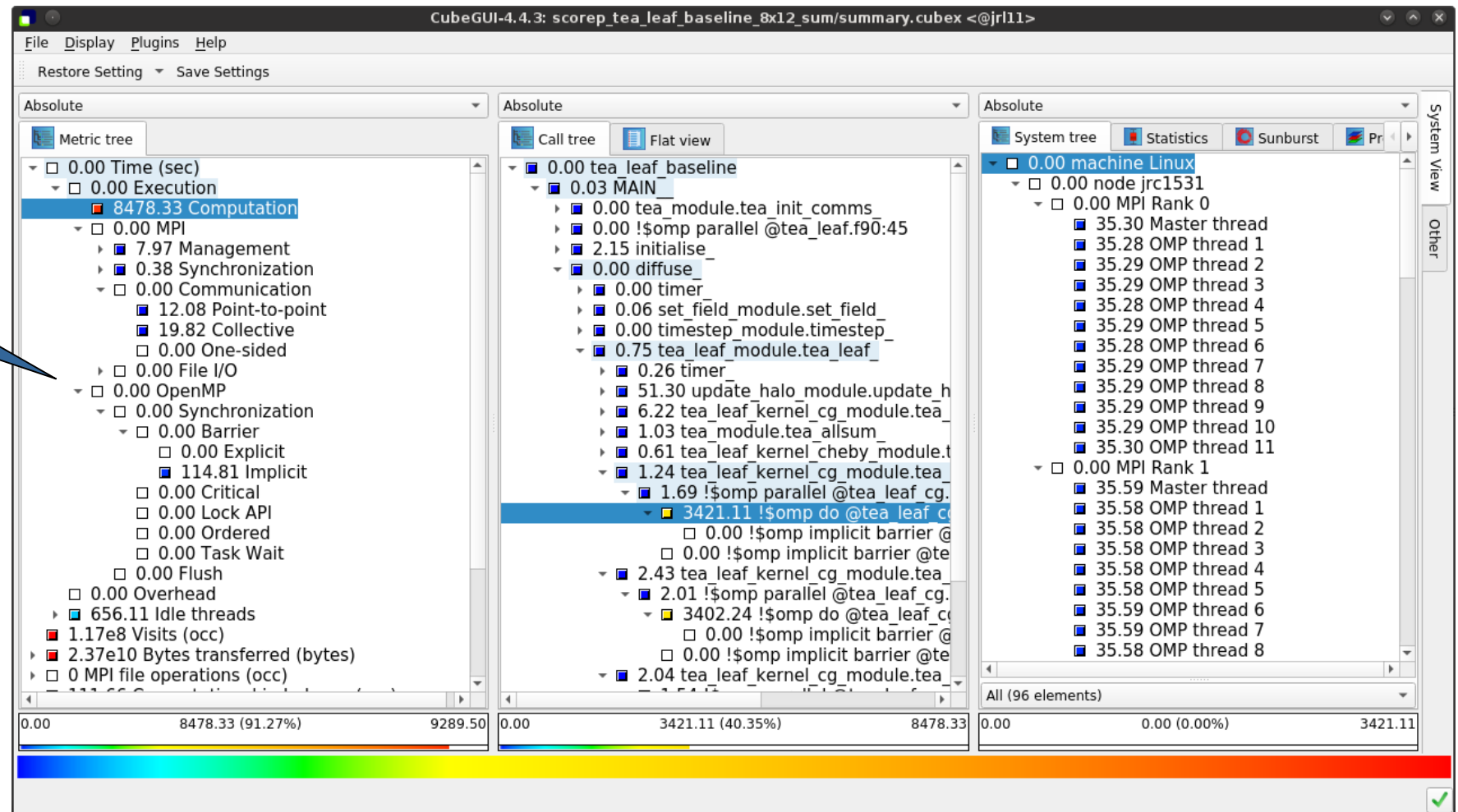
- Scalasca's report post-processing derives additional metrics and generates a structured metric hierarchy
- Automatically run (if needed) when using the **square** convenience command:

```
% square scorep_tea_leaf_baseline_8x12_sum  
INFO: Post-processing runtime summarization report (profile.cubex)...  
INFO: Displaying ./scorep_tea_leaf_baseline_8x12_sum/summary.cubex...
```

[GUI showing post-processed summary analysis report]

Post-processed summary analysis report

Split base metrics into more specific metrics

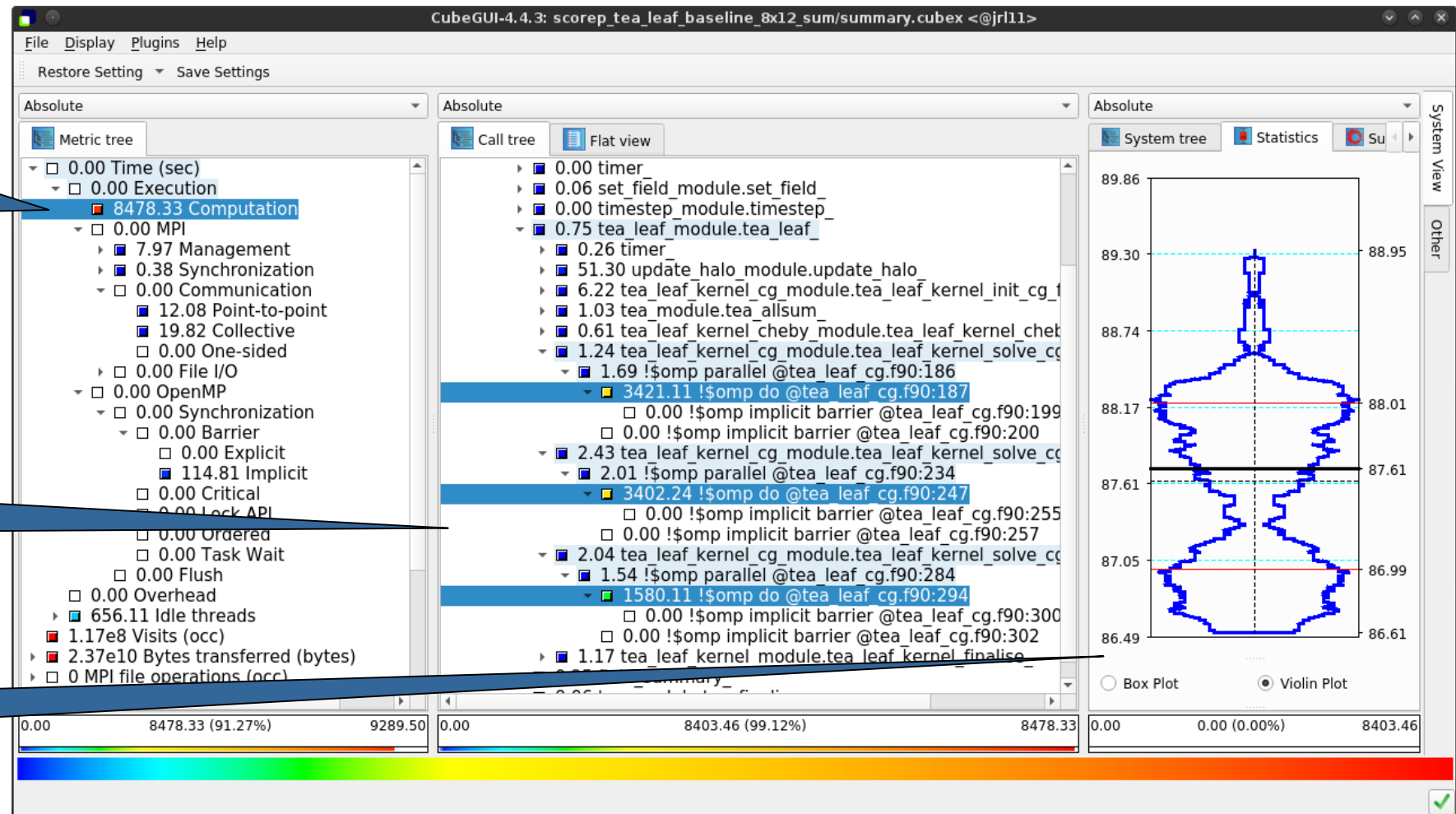


TeaLeaf summary report analysis (I)

91% of the execution time is computation...

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

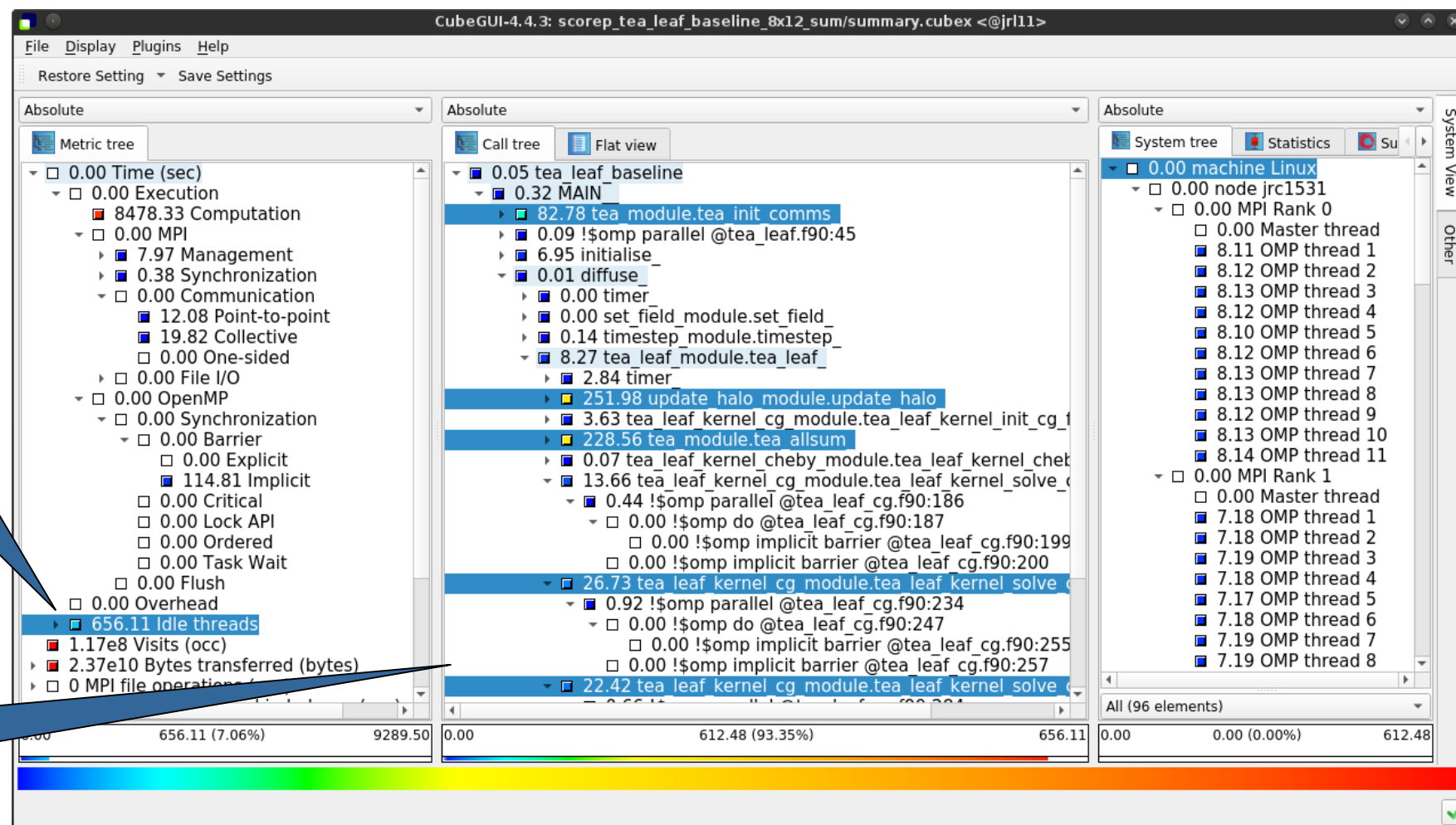
...with a slight imbalance across ranks & threads



TeaLeaf summary report analysis (II)

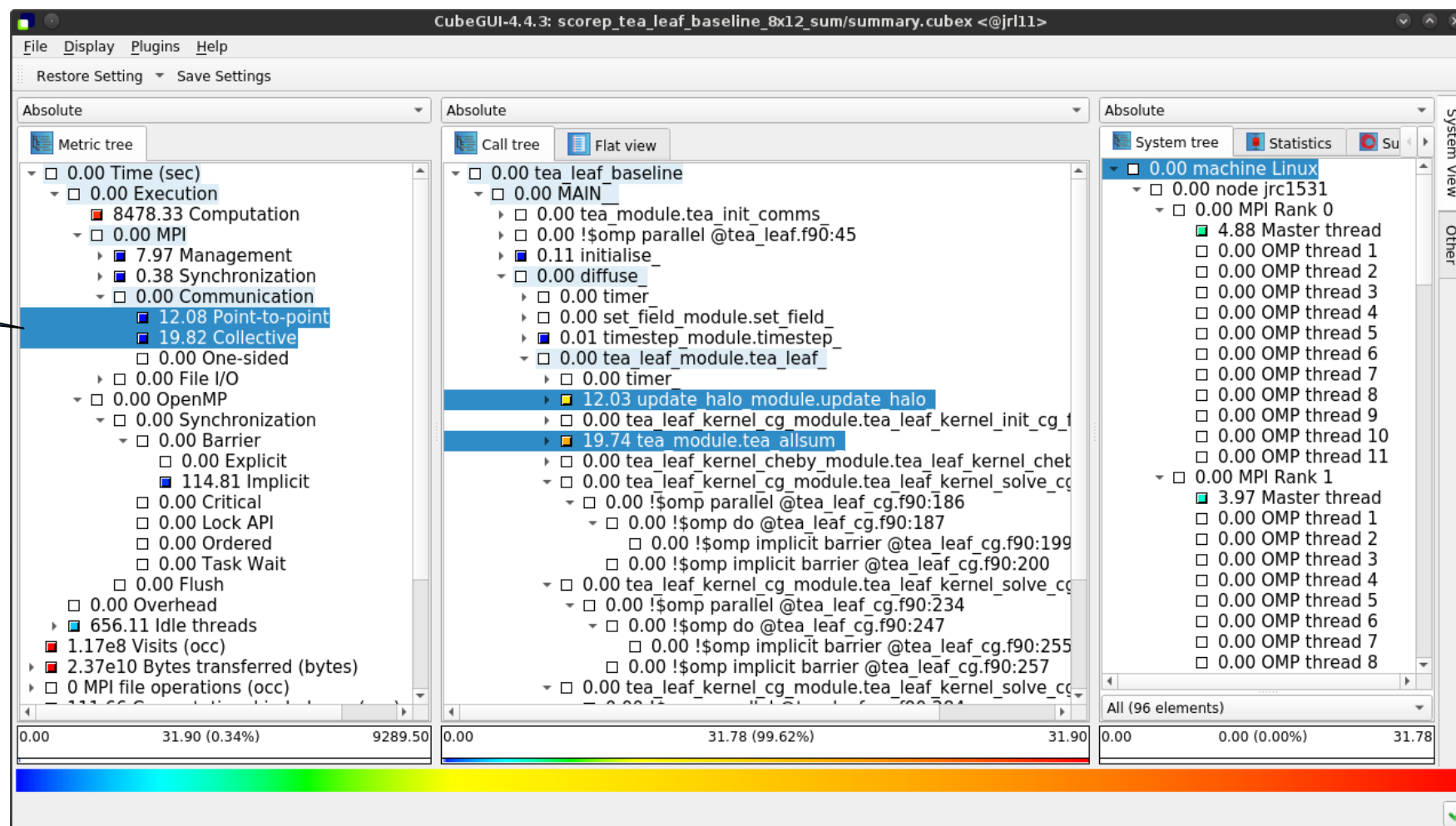
7% of the execution time are lost due to idle threads...

...in non-OpenMP parallelized code regions



TeaLeaf summary report analysis (III)

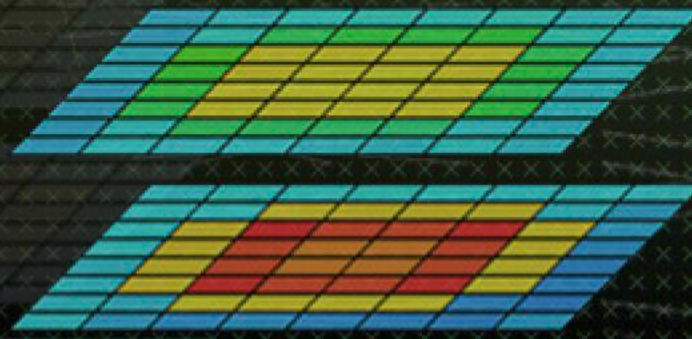
MPI communication time is negligible (0.34%); communication is only on the master threads (MPI_THREAD_FUNNELED)



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/cubegui/CubeUserGuide.pdf`
- Contact:
 - mailto: scalasca@fz-juelich.de





Reference material

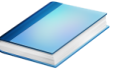


Derived metrics



- 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

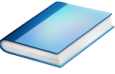
Parameters of the derived metric

CubePL expression

1.01e6 (100.00%) 1.01e6 0.00 2512.10

```
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 calculation of FLOPS based on PAPI_FP_OPS and time. The interface is divided into several panels:

- Edit metric FLOPS (on froggy1):** A dialog box for configuring the metric. It shows:
 - 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: $\text{metric::PAPI_FP_OPS()} / \text{metric::time()}$
- Metric tree:** A hierarchical view of metrics. The selected metric is **1.84e9 FLOPS**. Other visible metrics include:
 - 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)
- Call tree:** A hierarchical view of call trees. The selected call is **9.65e8 !\$omp do @exact_r...**. Other visible call trees include:
 - 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... (Selected)
 - 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:** A hierarchical view of the system tree. The selected system is **- machine Linux**. Other visible system trees include:
 - 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

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

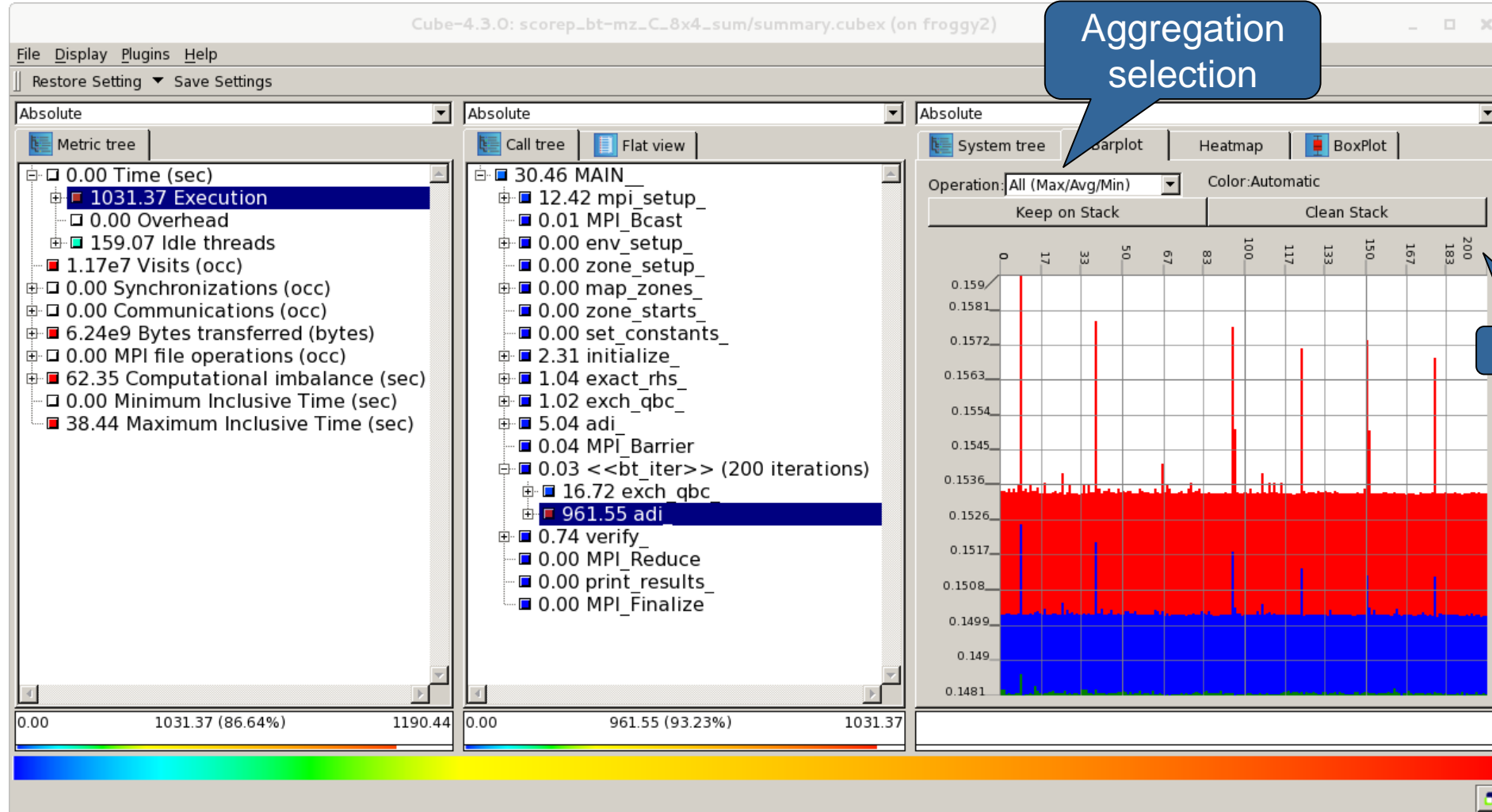
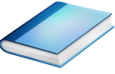


- Show time dependent behavior by “unrolling” iterations
- Preparations:
 - Mark loop body by using Score-P instrumentation API in your source code

```
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

