

# Analysis report examination with Cube

---

Marc Schlütter  
Jülich Supercomputing Centre

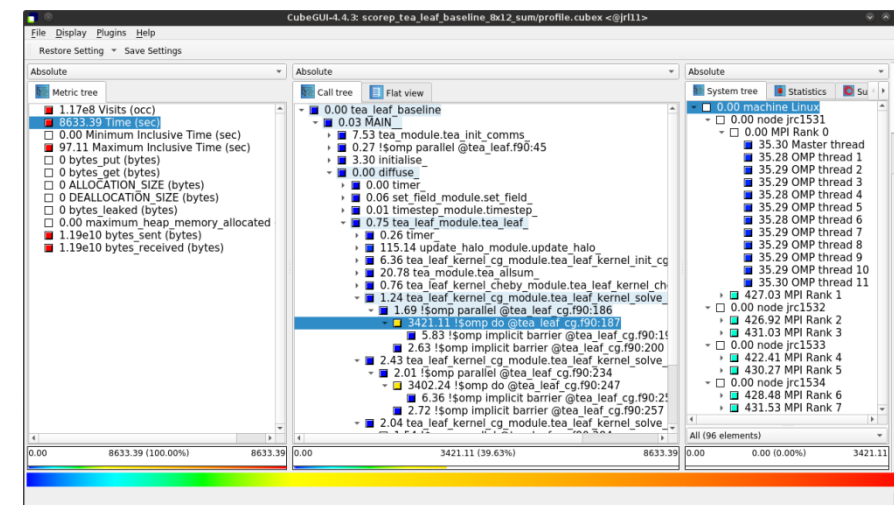


# Cube

CubeLib DOI 10.5281/zenodo.15051777

CubeGUI DOI 10.5281/zenodo.15051823

- 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 components
  - Can be installed independently of Score-P, e.g., on laptop or desktop
  - Latest release: Cube v4.9 (March 2025)



**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

# Cube GUI

mailto: [scalasca@fz-juelich.de](mailto:scalasca@fz-juelich.de)



## ▪ Run **remote** (*ssh*)

- start X server (e.g., Xming) locally
- connect to system with X forwarding enabled
  - Login node: *login1.amplitude.uni-due.de*
- load cube module and start cube remotely

```
desk$ ssh -X <yourid>@amplitude  
  
[login1~]$ module load cube/4.9-gcc-11.4.1  
[login1~]$ cube ./scorep-*/profile.cubex
```

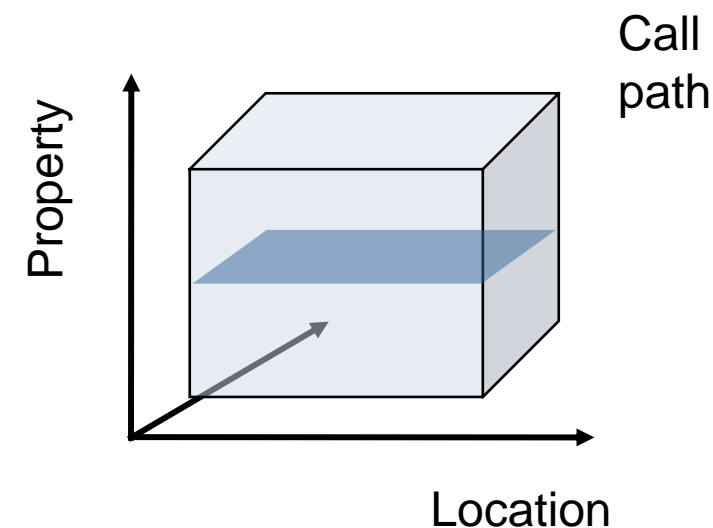
## ▪ Install & run **local**

- 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@]remote.sys:[dir] $HOME/mnt  
desk$ cd $HOME/mnt  
desk$ cube ./scorep-*/profile.cubex
```

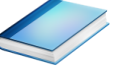
# Analysis presentation and exploration

- 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*

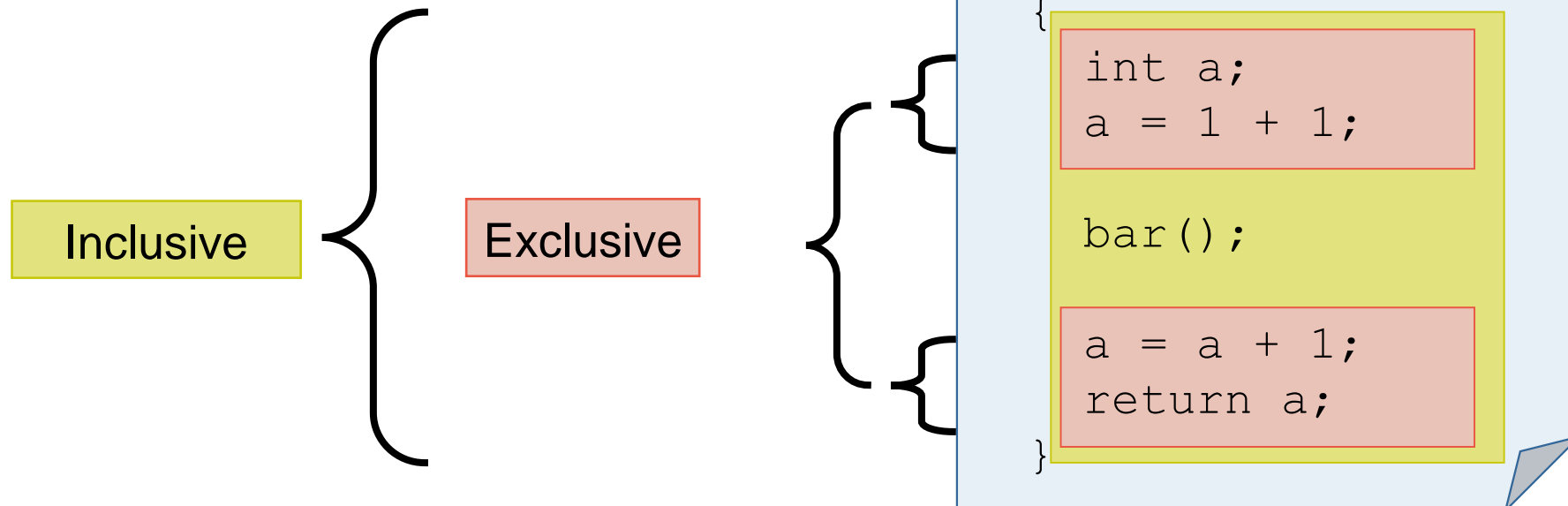




# Inclusive vs. exclusive values



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



# Demo: TeaLeaf case study

---



## Case study: TeaLeaf

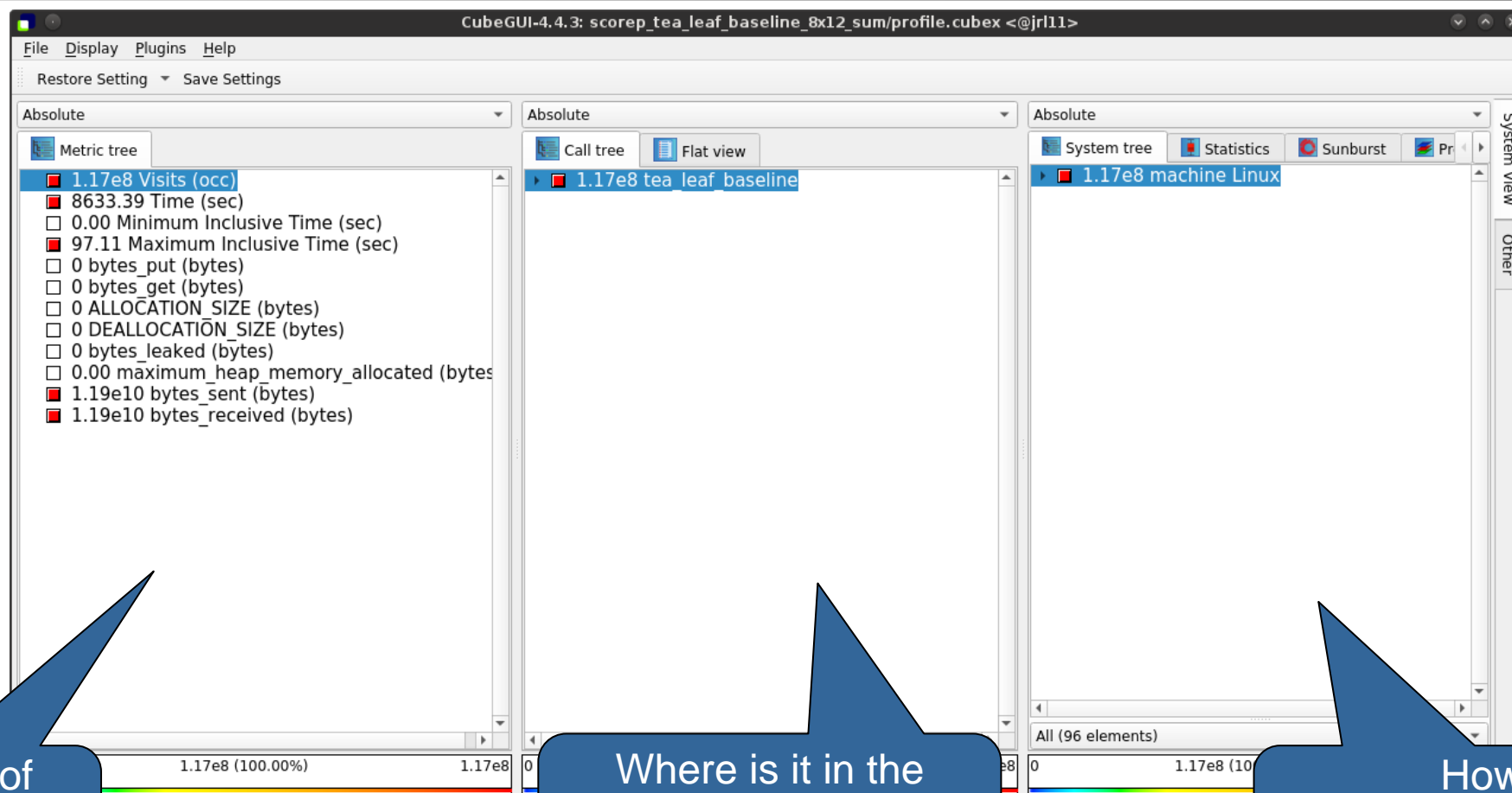
---

- 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



```
% cd ~/workshop-vihps/Experiments
% cube scorep_tea_leaf_baseline_8x12_sum/profile.cubex
[GUI showing summary analysis report]
```

# Score-P analysis report exploration (opening view)



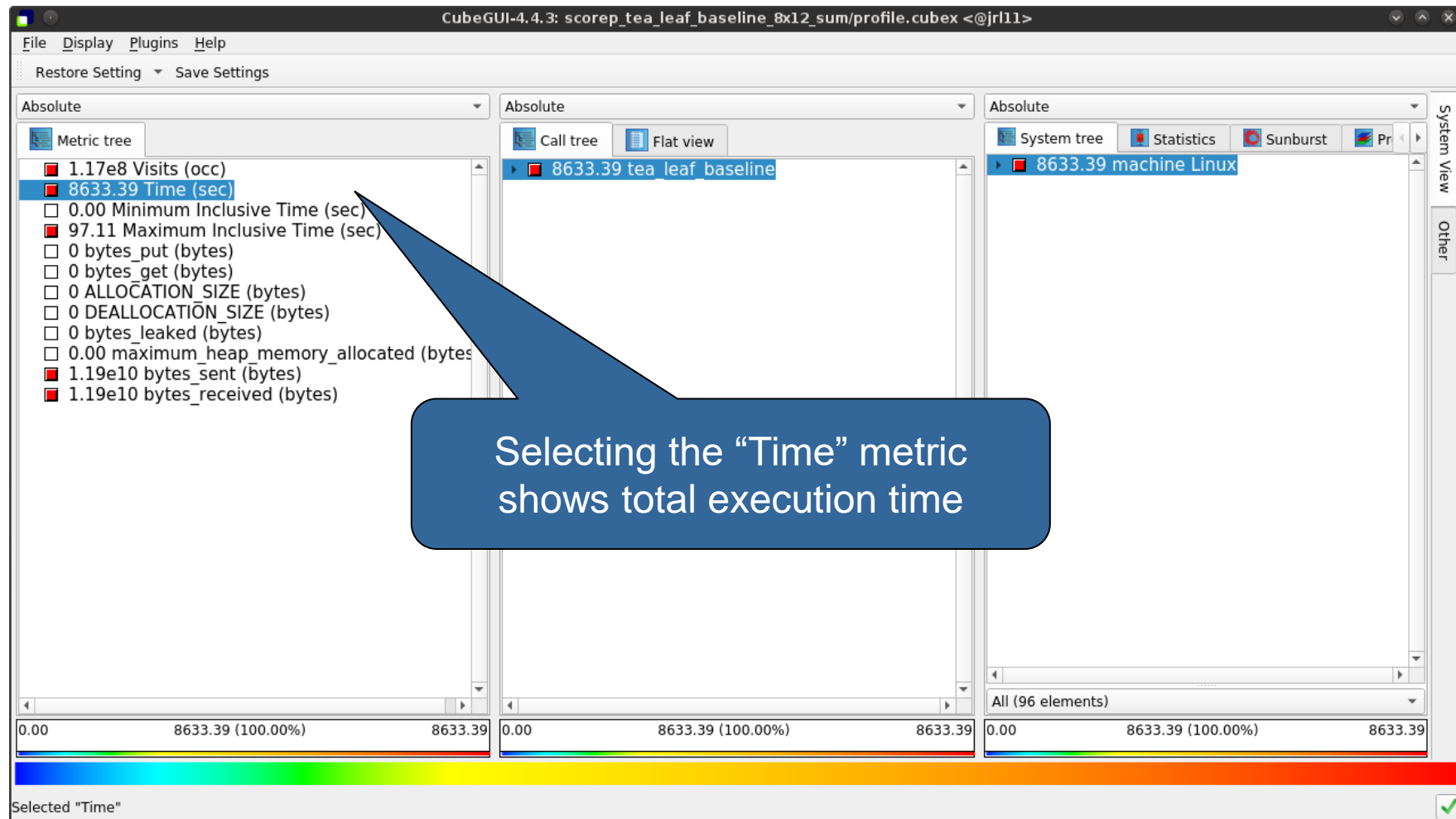
What kind of performance metric?

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

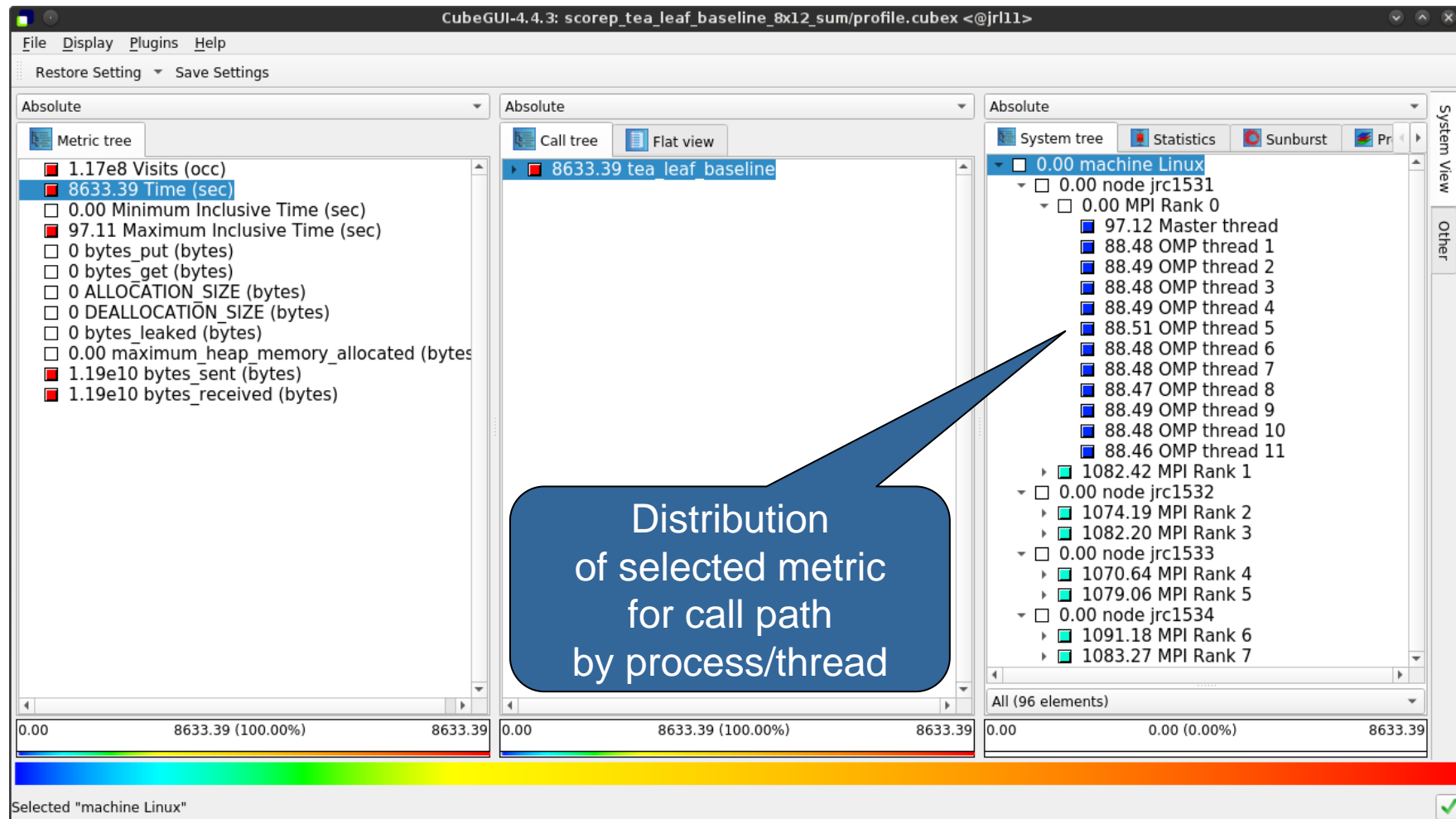
How is it distributed across the processes/threads?



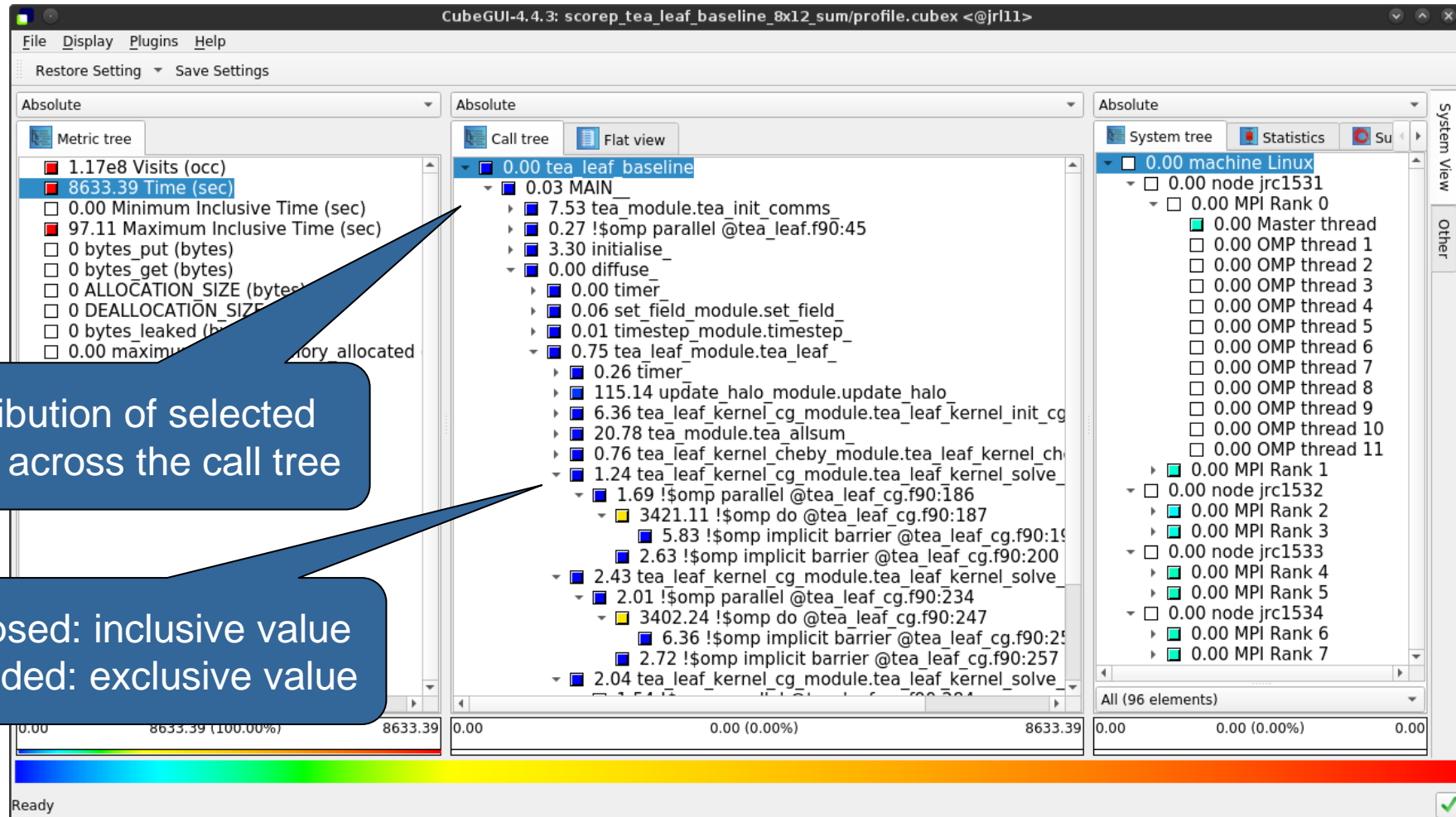
# Metric selection



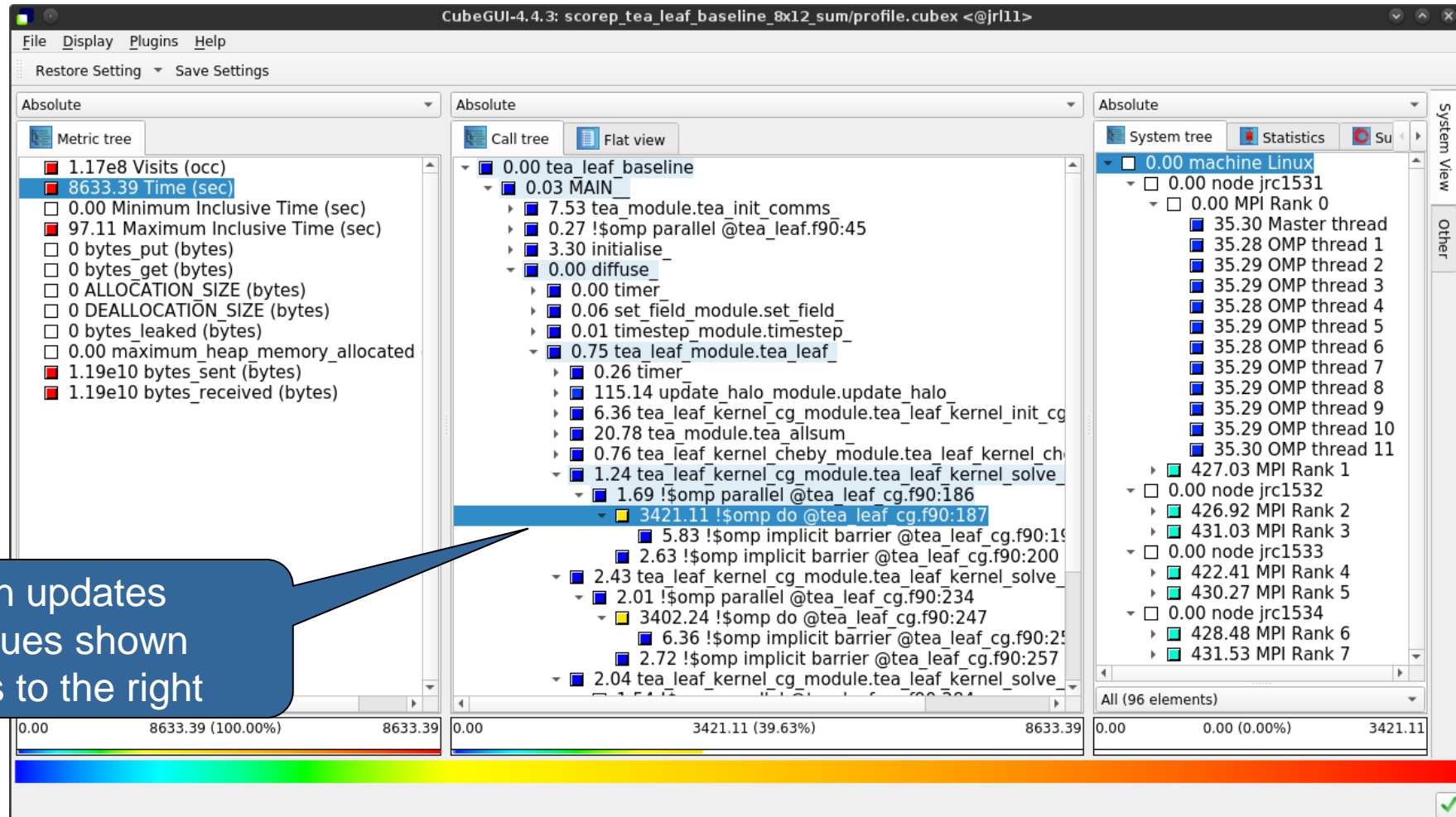
# Expanding the system tree



# Expanding the call tree



# Selecting a call path





# Multiple selection

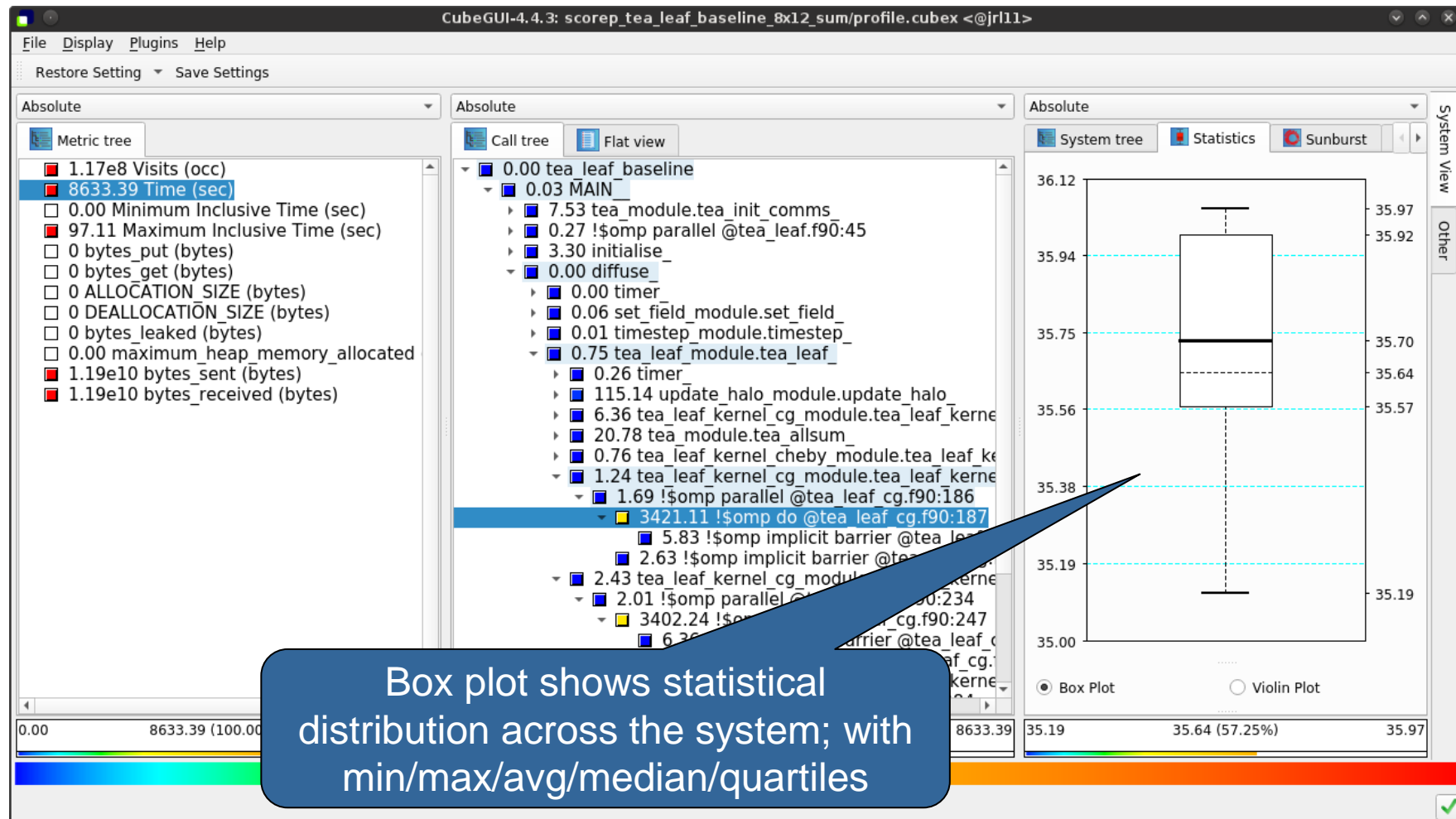
The screenshot displays the CubeGUI-4.4.3 interface with the title bar "CubeGUI-4.4.3: scorep\_tea\_leaf\_baseline\_8x12\_sum/profile.cubex <@jrl11>". The interface is divided into three main panels: "Metric tree", "Call tree", and "System tree".

- Metric tree:** Shows various performance metrics. The "8633.39 Time (sec)" metric is highlighted in blue.
- Call tree:** Displays a hierarchical view of function calls. Several nodes are selected with blue highlights, including "0.75 tea\_leaf\_module.tea\_leaf", "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", "6.36 !\$omp implicit barrier @tea\_leaf\_cg.f90:257", "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", and "0.25 field\_summary".
- System tree:** Shows the system hierarchy. The "0.00 machine Linux" node is selected, and its sub-nodes, including "0.00 node jrc1531", "0.00 MPI Rank 0", and various MPI ranks, are listed.

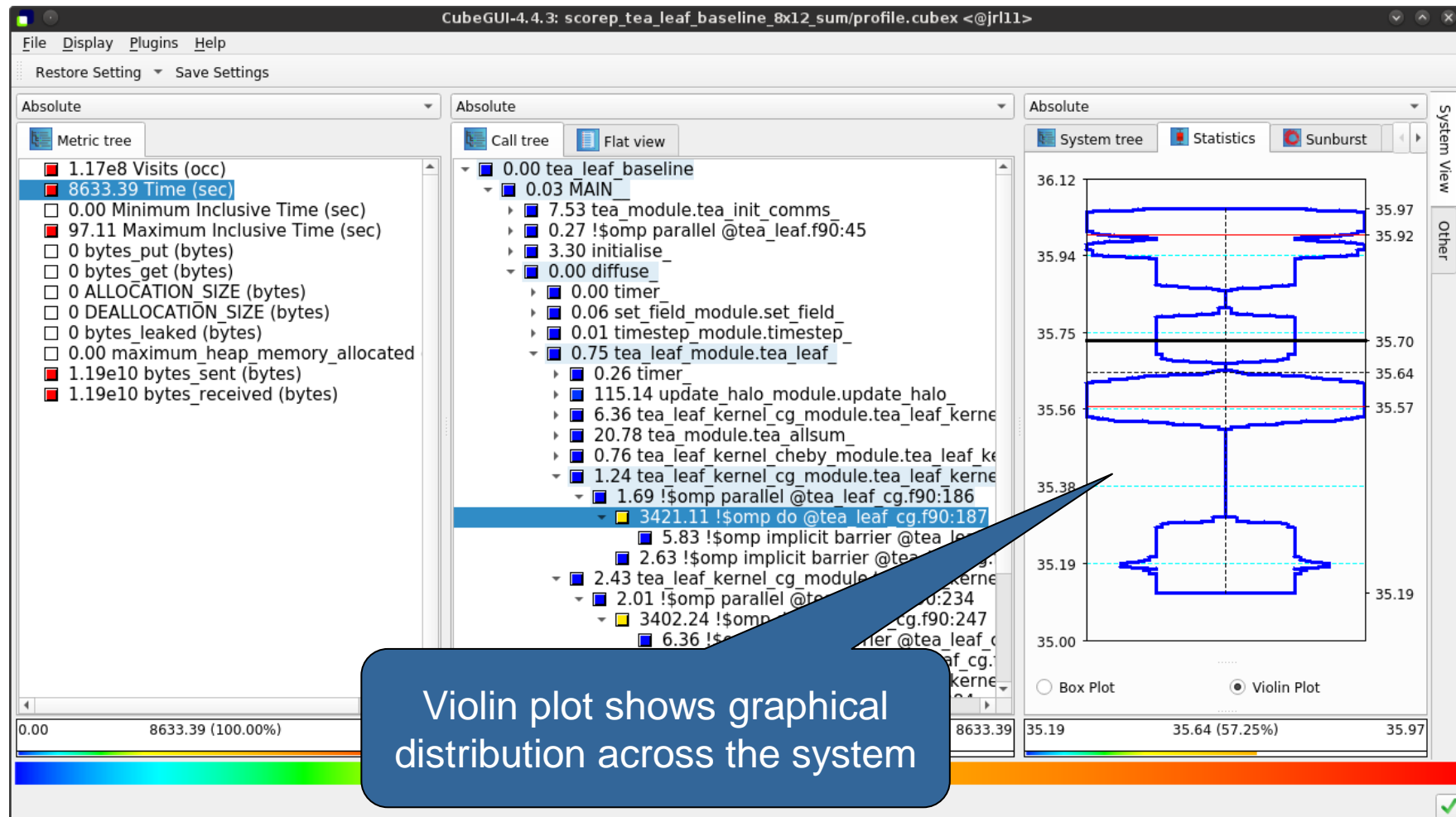
A blue callout box with a speech bubble icon points to the "System tree" panel, containing the text: "Select multiple nodes with Ctrl-click".

At the bottom of the interface, there are three progress bars showing the selection status of the metrics, call tree, and system tree. The first bar shows "0.00 8633.39 (100.00%) 8633.39". The second bar shows "0.00 8403.46 (97.34%) 8633.39". The third bar shows "0.00 0.00 (0.00%) 8403.46".

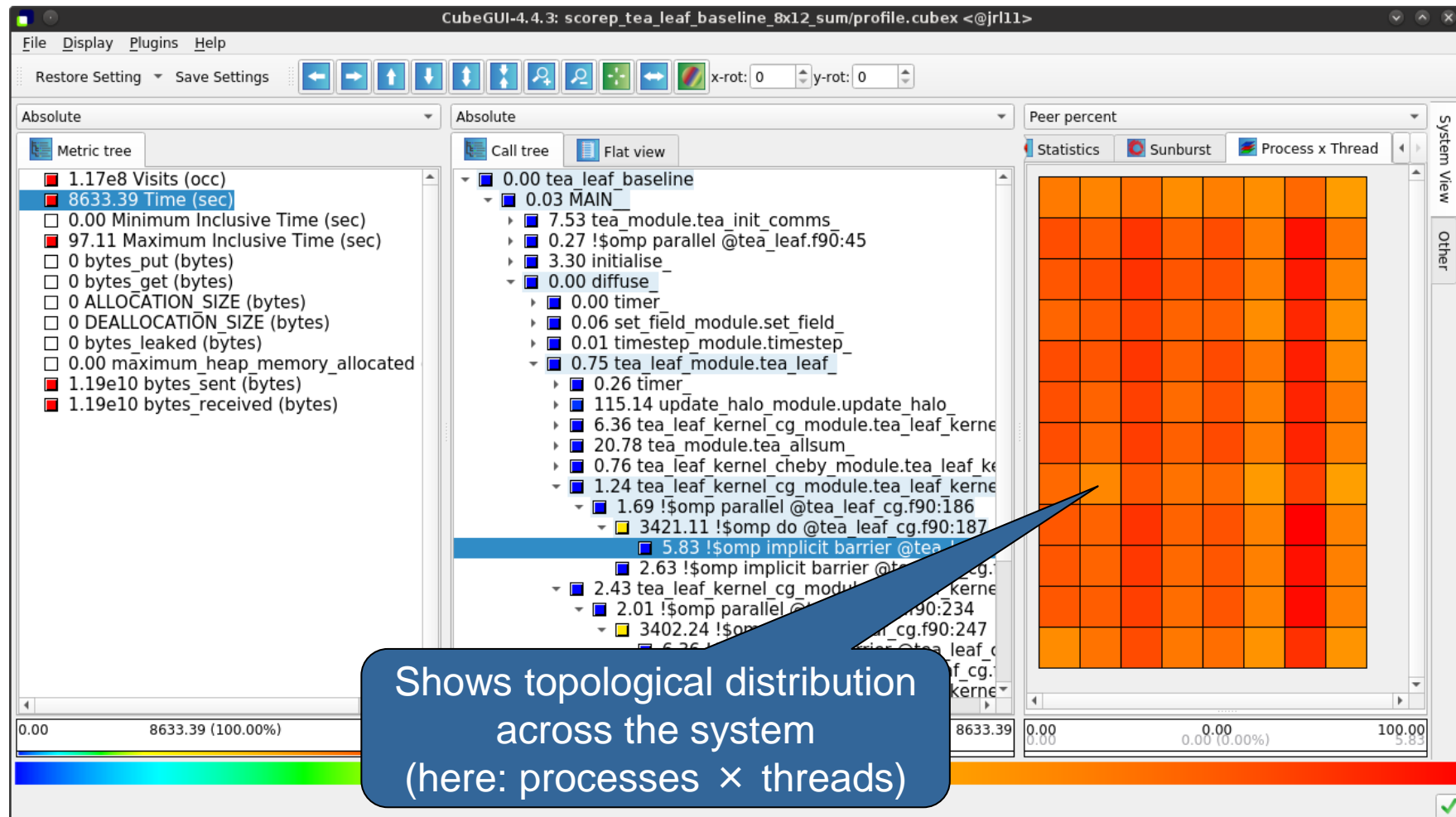
# Box plot view



# Violin plot view

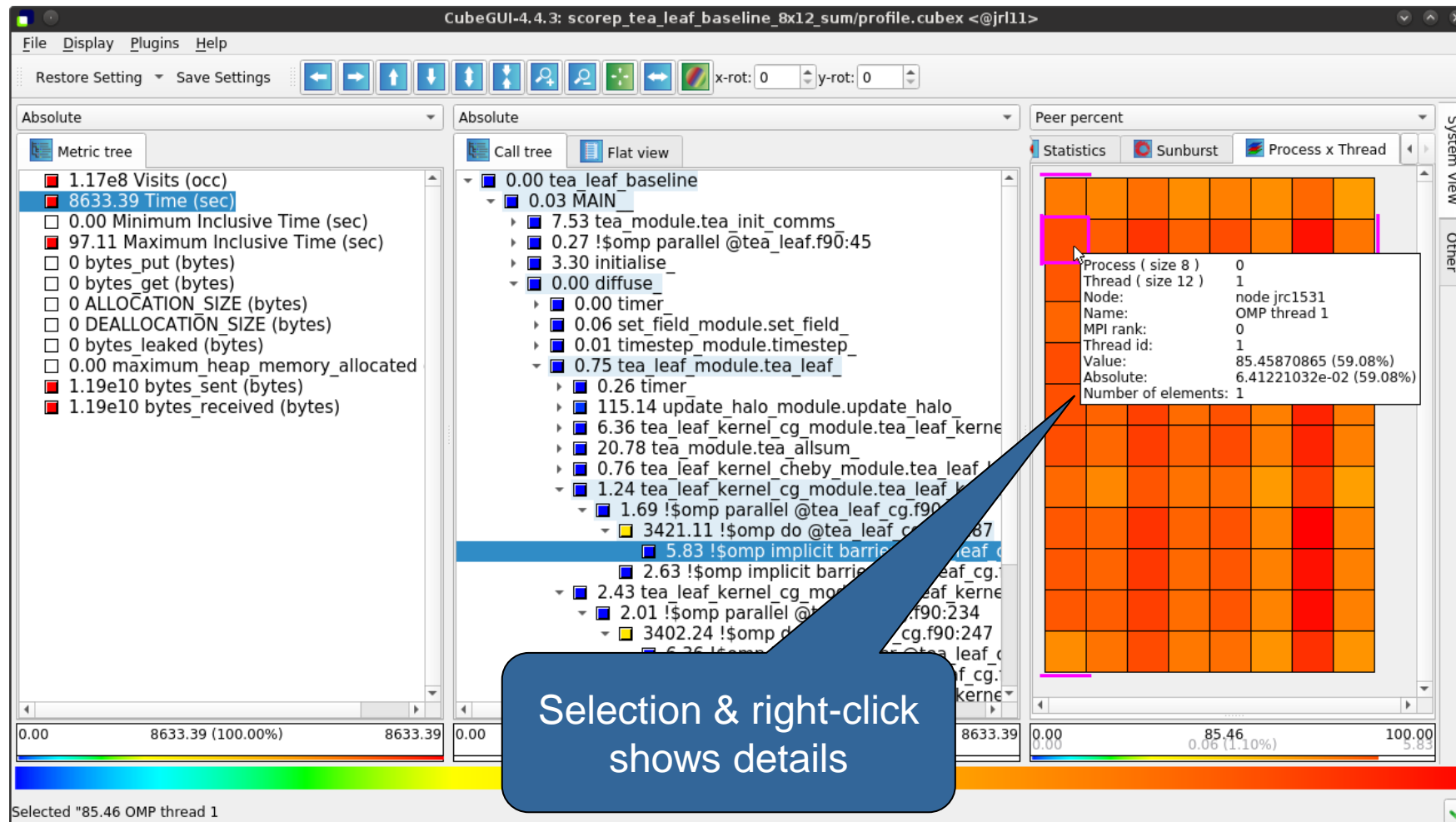


# Topology view

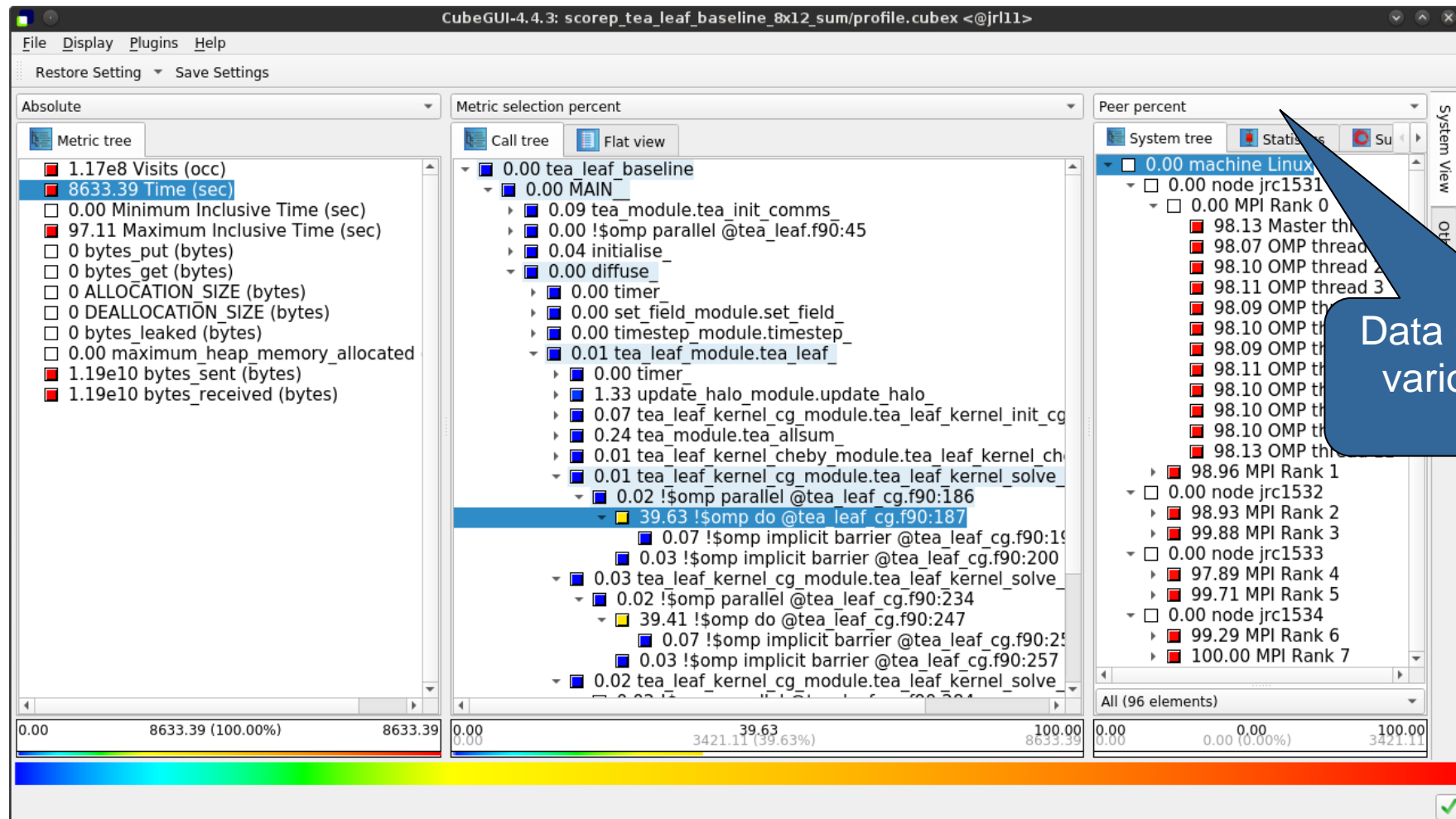




## Topology view (cont.)



# 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 right-click context menu is open over the selected 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 the text "Right-click opens context menu" points to the right-click action. The bottom of the interface shows a progress bar and a status message: "Shows a short description of the clicked item".

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

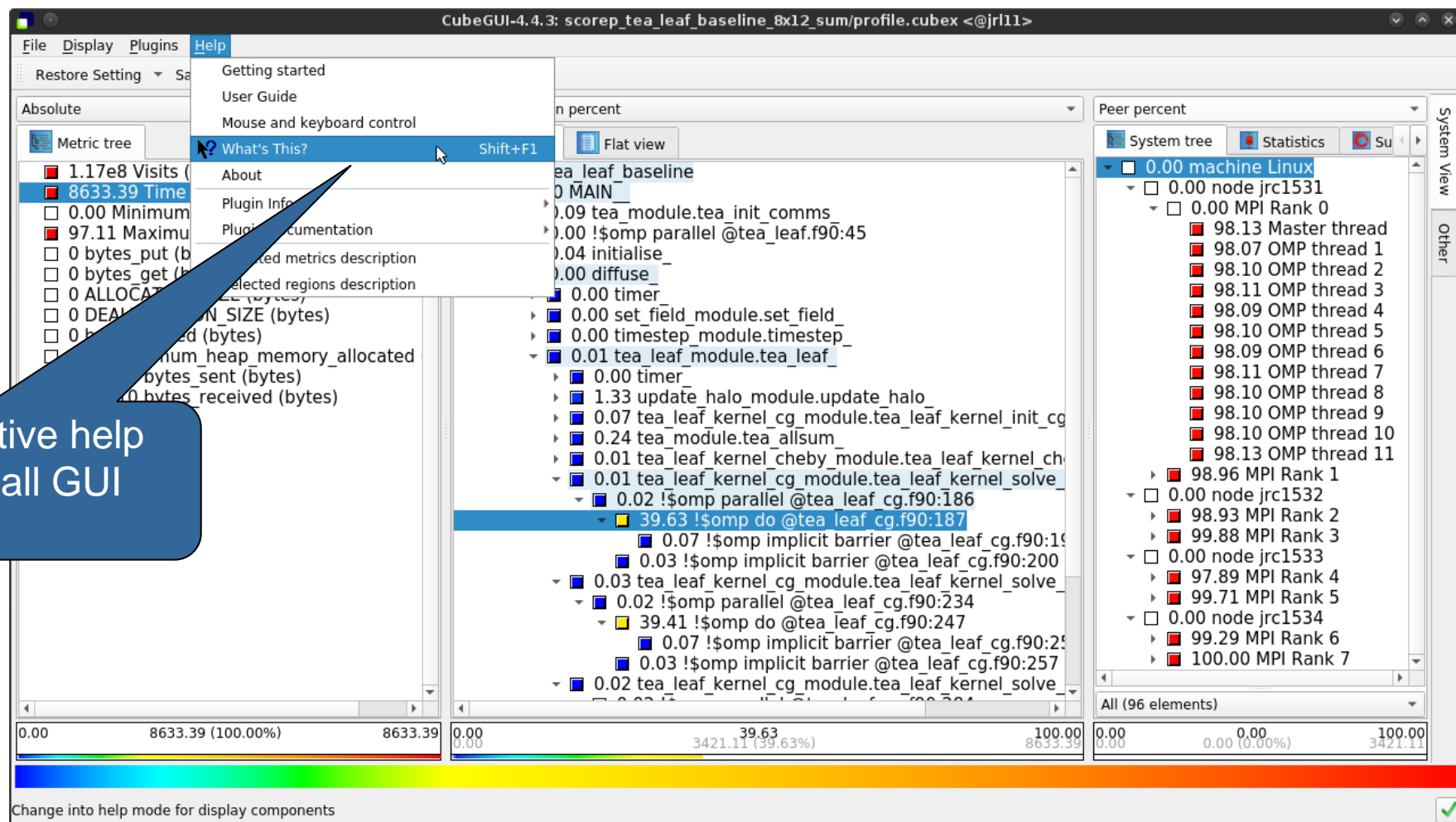
- 0.00 tea\_leaf\_baseline
  - 0.03 MAIN
    - 7.53 tea\_module.tea\_init\_comms
    - 0.27 !\$omp parallel @tea\_leaf.f90:45
    - 3.30 initialise
    - 0.00 diffuse
      - 0.00 timer
      - 0.06 set\_field\_module.set\_field
      - 0.01 timestep\_module.timestep
      - 0.75 tea\_leaf\_module.tea\_leaf
        - 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.f90
            - 3421.11 !\$omp do @tea\_leaf\_cg.f90
            - 5.83 !\$omp implicit barrier @tea
            - 2.63 !\$omp implicit barrier @tea
            - 2.43 tea\_leaf\_kernel\_cg\_module.tea\_le
              - 2.01 !\$omp parallel @tea\_leaf\_cg.f90
                - 3402.24 !\$omp do @tea\_leaf\_cg.f90
                - 6.36 !\$omp implicit barrier @tea
                - 2.72 !\$omp implicit barrier @tea
                - 2.04 tea\_leaf\_kernel\_cg\_module.tea\_le

Score-P Configuration Source Info

```
170 IMPLICIT NONE
171
172 INTEGER(KIND=4):: x_min,x_max,y_min,y_max
173 REAL(KIND=8), DIMENSION(x_min-2:x_max+2,y_min-2:y
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
178 REAL(KIND=8) :: rx, ry
179
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 the availability of the source code, as well as file and line number information provided by the instrumentation, i.e., it may not always be available

# Context-sensitive help



## Scalasca report post-processing

---

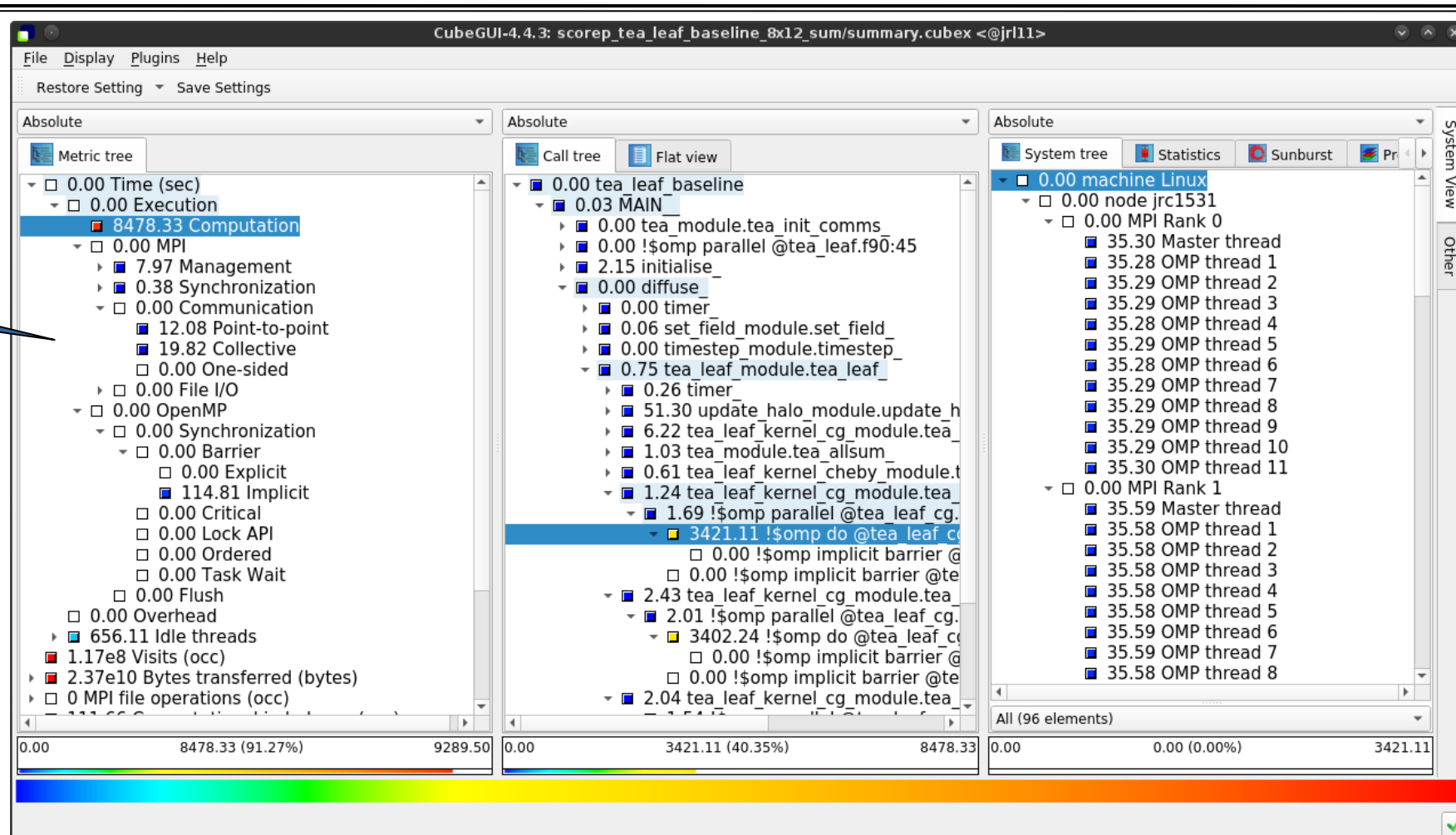
- Scalasca's report post-processing derives additional metrics and generates a structured metric hierarchy
- Automatically run (if needed) when using the **square** convenience command:

```
% 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]
```

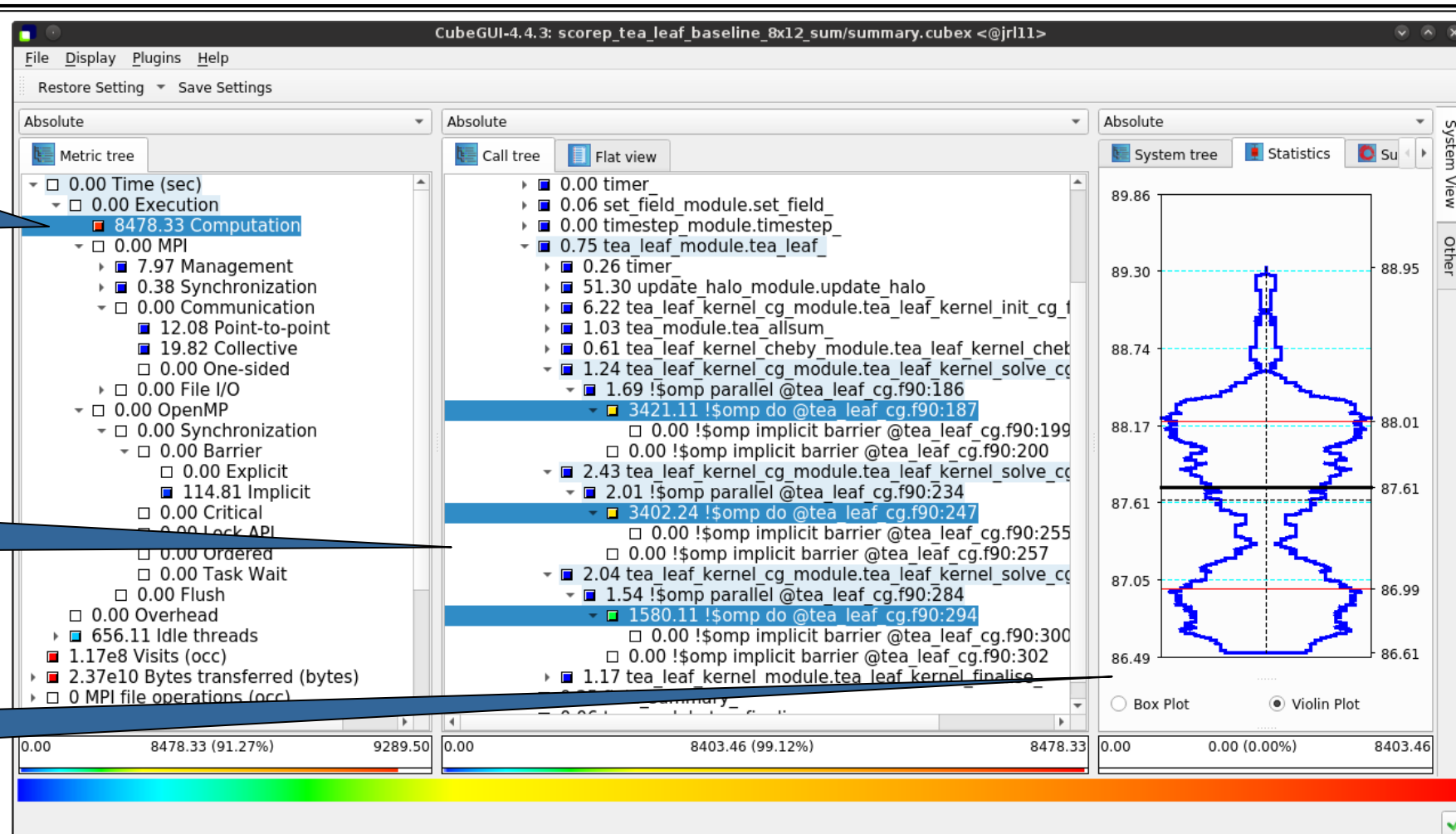
# Post-processed summary analysis report

Split base metrics into more specific metrics, e.g. computation vs parallelization costs





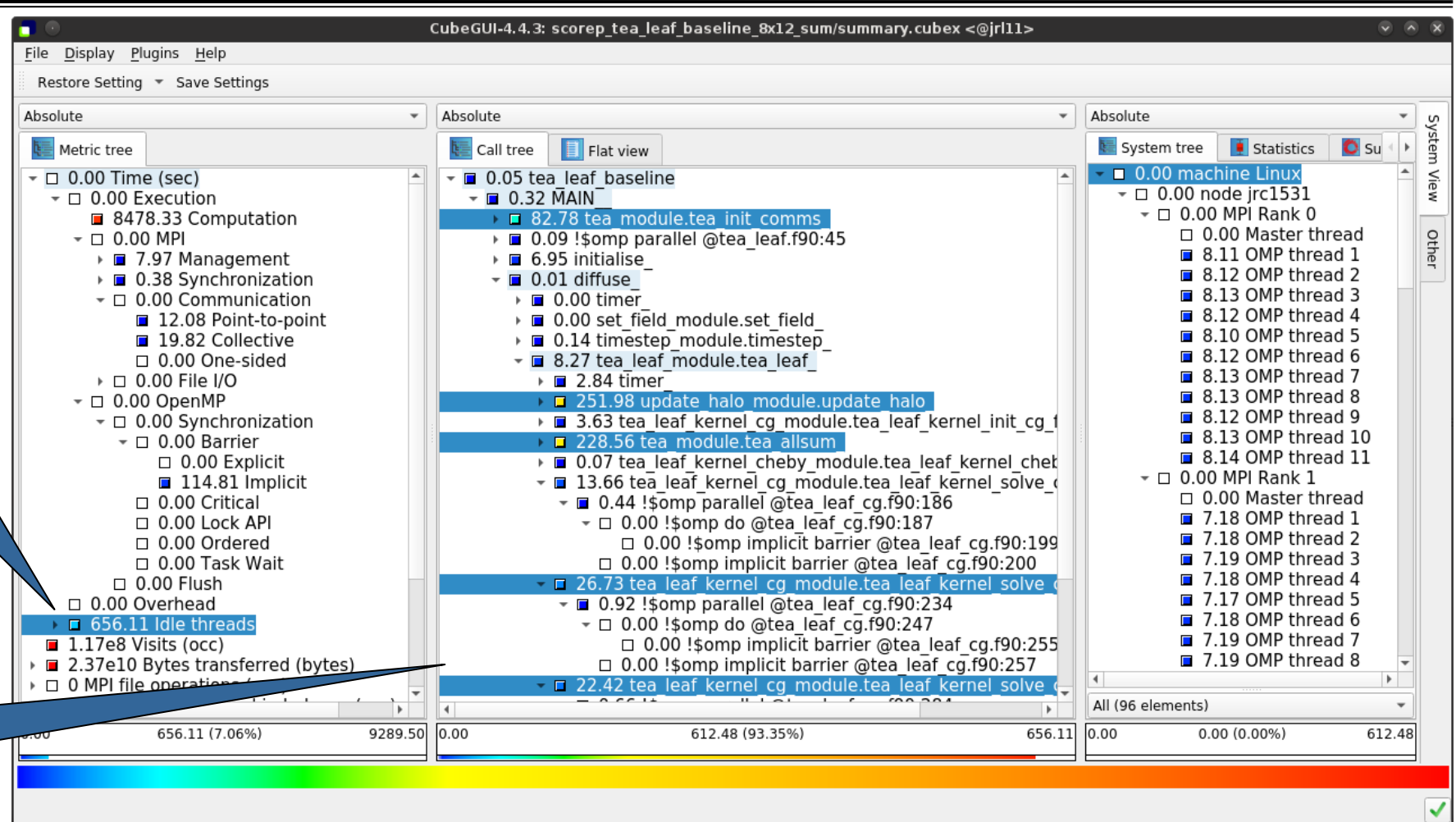
# TeaLeaf summary report analysis (I)



## TeaLeaf summary report analysis (II)

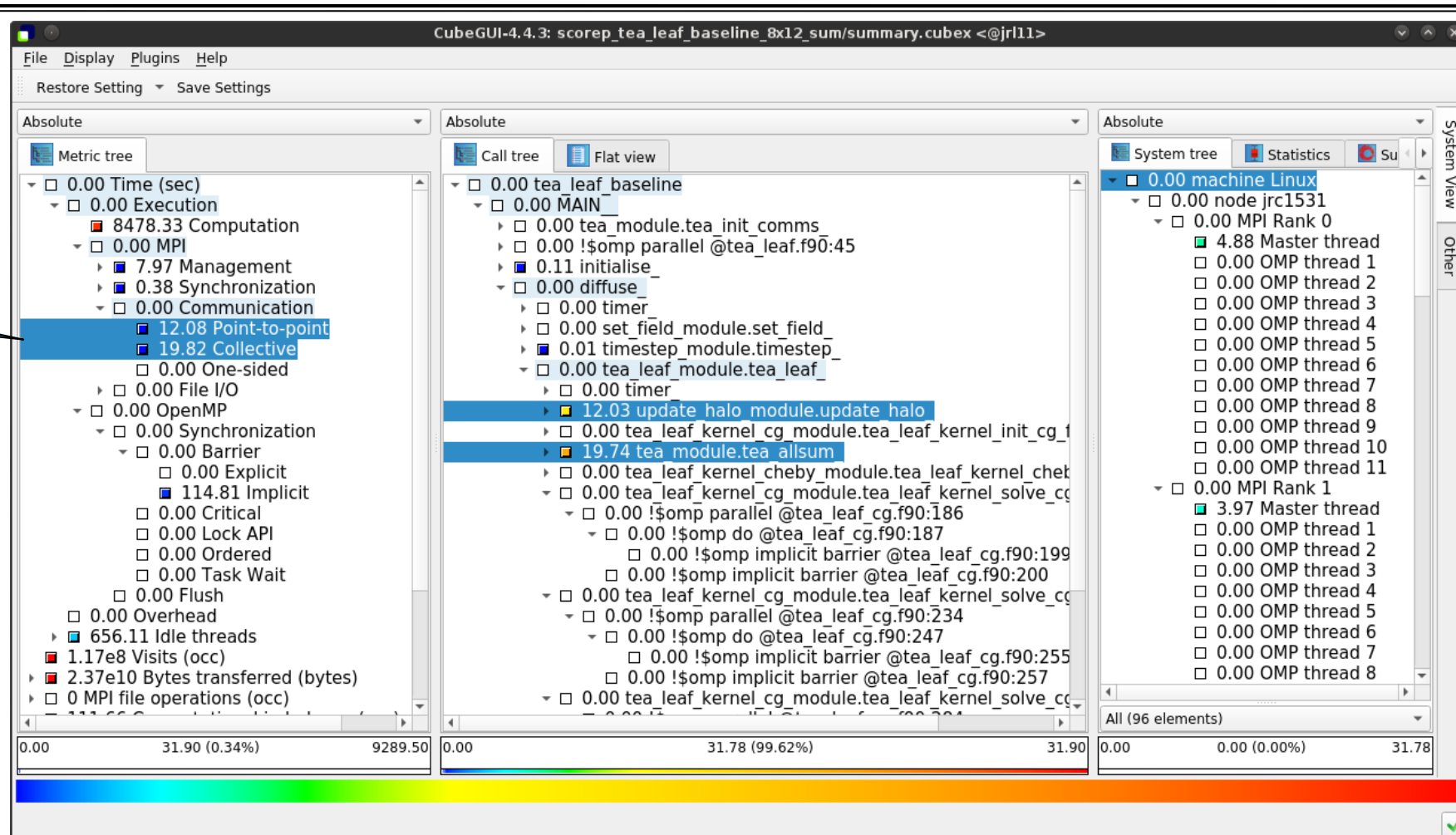
7% of the total CPU execution time is due to "idle threads"...

... when not within OpenMP-parallelized code regions



## TeaLeaf summary report analysis (III)

MPI communication time is negligible (0.34%); but 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:
  - <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)

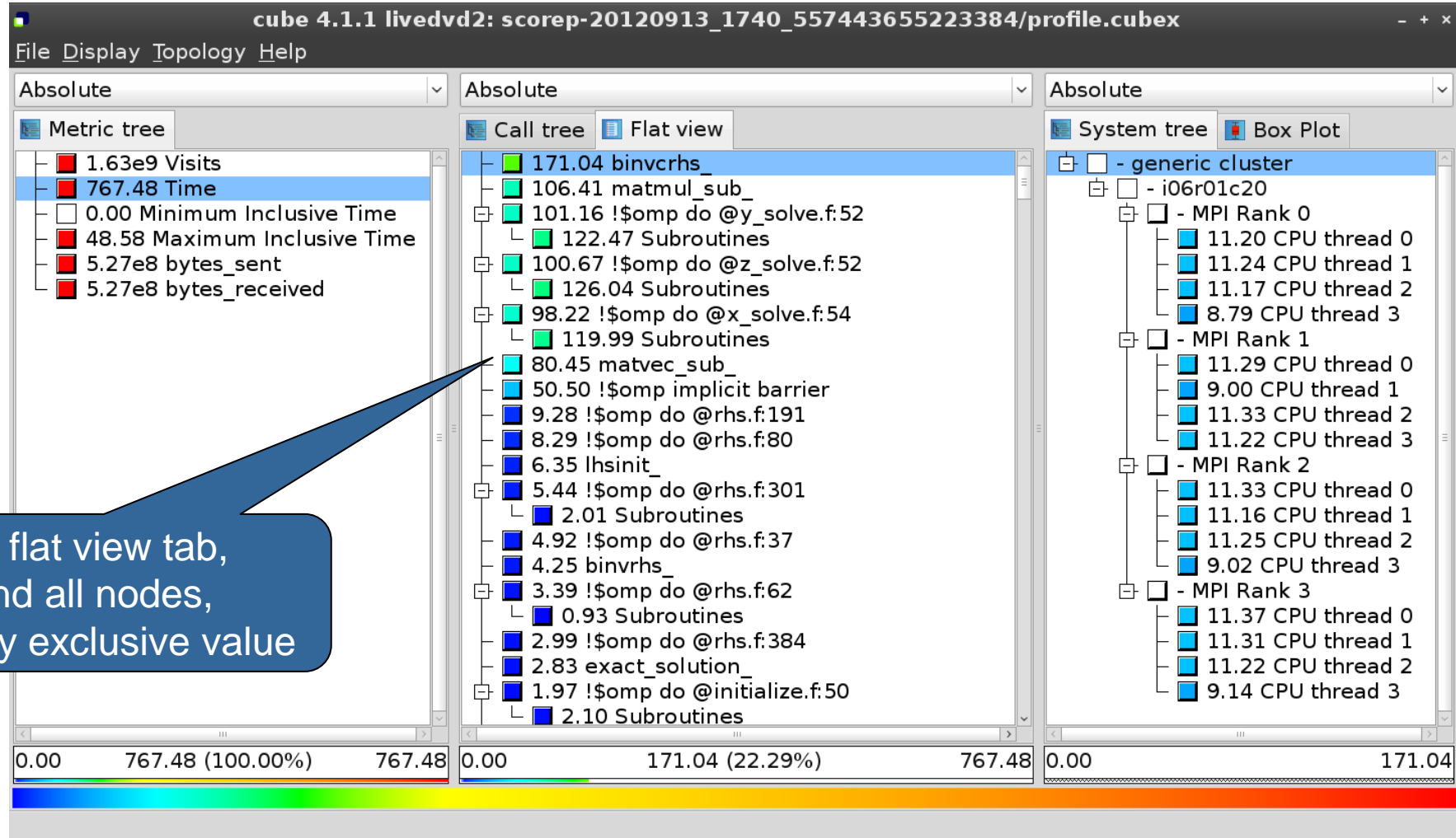
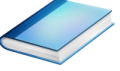




## Reference material



# Flat profile view



## 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)`**

# 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



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



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

The image displays three screenshots from the Cube-4.3.1 software interface, illustrating the configuration and analysis of FLOPS metrics.

**Left Screenshot: Edit metric FLOPS (on froggy1)**

This window shows the configuration for a derived metric named "FLOPS".

- 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:**

The calculation formula is defined as:

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

Buttons: Calculation, Calculation Init, Aggregation "+", Aggregation "-", Edit metric, Cancel.

Share this metric with SCALASCA group

**Metric tree (Middle Screenshot):**

The metric tree shows the following metrics (values are in scientific notation):

- 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** (highlighted)

**Call tree (Right Screenshot):**

The call tree shows the following metrics (values are in scientific notation):

- 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...** (highlighted)
- 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 (Far Right Screenshot):**

The system tree shows the following metrics (values are in scientific notation):

- 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"

# 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

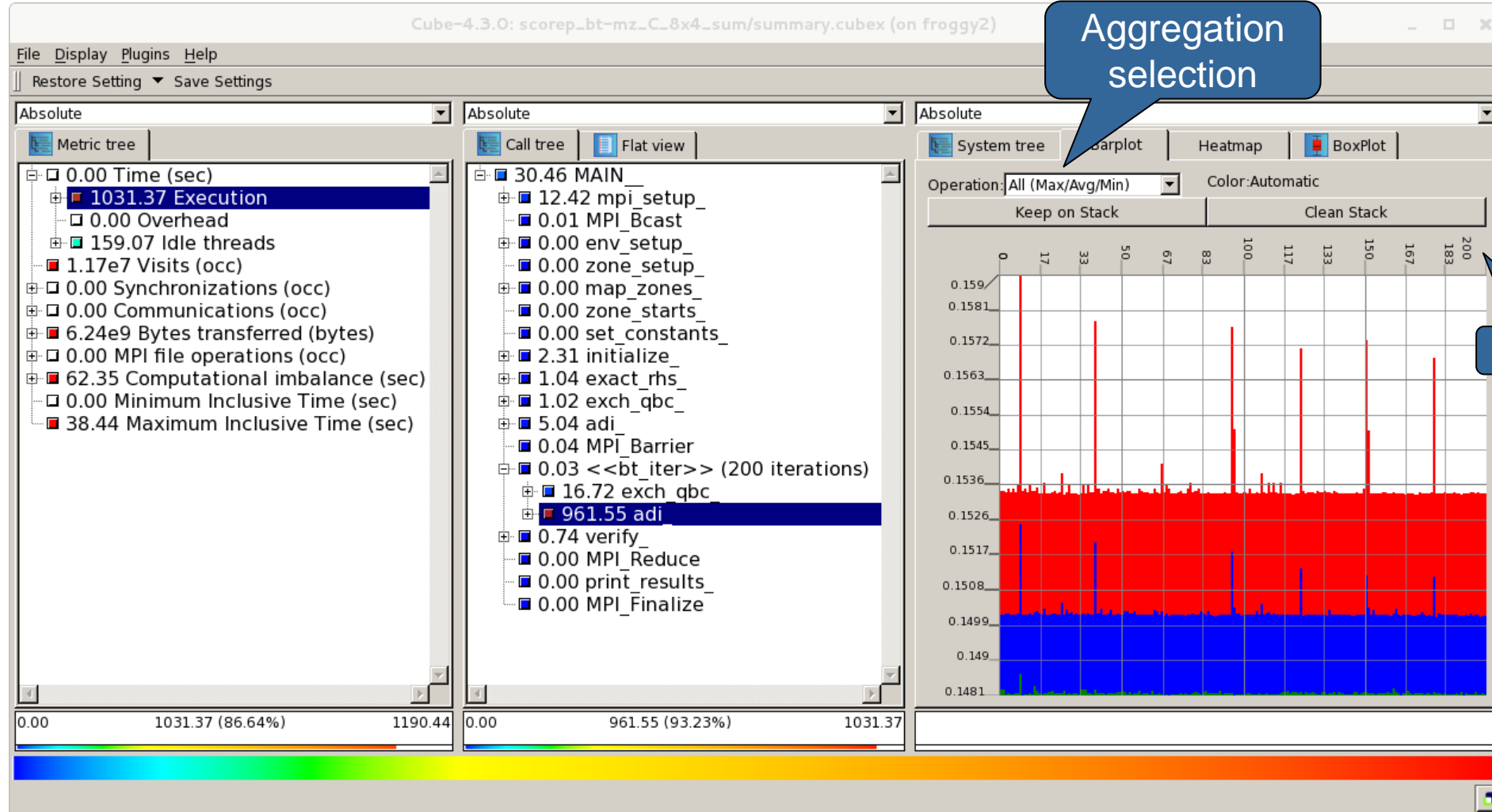


- Show time dependent behavior by “unrolling” iterations
- Preparations:
  - Mark loop body by using Score-P instrumentation API in your source code

```
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

