International HPC Summer School 2016: Scoring-based measurement configuration and automatic trace analysis with Scalasca

Christian Feld
Jülich Supercomputing Centre

Score-P scalasca

Recap: Local installation

- VI-HPS tools not yet installed system-wide
  - Source provided shell code snippet to add local tool installations to $PATH
  - Required for each shell session

```
% source /home/roessel/ihpcss16/tools/source.me.gcc-openmpi
```

- Copy tutorial sources to your working directory, ideally on a parallel file system (recommended: $SCRATCH)

% cd $HOME
% ~~tar zxvf /home/roessel/ihpcss16/tutorial/NPB3.3-MZ-MPI.tar.gz~~
% cd NPB3.3-MZ-MPI

Recap: BT-MZ summary analysis report examination

```
% cd bin.scorep
% ls -1
bt-mz_C.8
scorep.sbatch.C.8
scorep-C.8-<jobid>.err
scorep-C.8-<jobid>.out
scorep_bt-mz_C.8x7.<jobid>

% ls scorep_bt-mz_C.8x7.<jobid>
profile.cubex  scorep.cfg


laptop> scp userid@bridges.psc.edu:~/NPB3.3-MZ-MPI/bin.scorep/\
scorep_bt-mz_C.8x7.<jobid>/profile.cubex .

laptop> paraprof profile.cubex

       [Paraprof GUI showing summary analysis report]
       [You can use Cube on profile.cubex as well]
```

- Creates experiment directory including
  - A record of the measurement configuration (scorep.cfg)
  - The analysis report that was collated after measurement (profile.cubex)
- Interactive exploration with Paraprof

Hint:
Copy ‘profile.cubex’ to your laptop using ‘scp’ to improve responsiveness of GUI

Congratulations!?

- If you made it this far, you successfully used Score-P to
  - instrument the application
  - analyze its execution with a summary measurement, and
  - examine it with one the interactive analysis report explorer GUIs
- ... revealing the call-path profile annotated with
  - the “Time” metric
  - Visit counts
  - MPI message statistics (bytes sent/received)
  - PAPI hardware-counters
- ... but how ***good*** was the measurement?
  - The measured execution produced the desired valid result
  - however, the execution took rather longer than expected!
    - even when ignoring measurement start-up/completion, therefore
    - it was probably dilated by instrumentation/measurement overhead

Performance Analysis Steps

- 0.0 Reference preparation for validation

- 1.0 Program instrumentation
- 1.1 Summary measurement collection
- 1.2 Summary analysis report examination

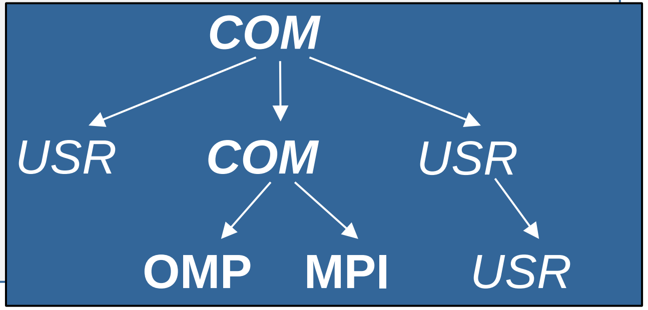
- 2.0 Summary experiment scoring
- 2.1 Event trace collection with filtering
- 2.2 Event trace examination & analysis

BT-MZ Summary Analysis Result Scoring

```
% scorep-score scorep_bt-mz_C.8x7.<jobid>/profile.cubex

Estimated aggregate size of event trace:                   159GB
Estimated requirements for largest trace buffer (max_buf): 20GB
Estimated memory requirements (SCOREP_TOTAL_MEMORY):       20GB
(warning: The memory requirements cannot be satisfied by Score-P to avoid
 intermediate flushes when tracing. Set SCOREP_TOTAL_MEMORY=4G to get the
 maximum supported memory or reduce requirements using USR regions filters.)

flt     type     max_buf[B]        visits time[s] time[%] time/visit[us]  region
         ALL 21,389,438,207 6,557,153,121 1934.82   100.0           0.30  ALL
         USR 21,309,225,314 6,537,020,537  835.65    43.2           0.13  USR
         OMP     76,450,336    19,013,888 1087.70    56.2          57.21  OMP
         COM      3,525,730     1,084,840    2.20     0.1           2.03  COM
         MPI        236,827        33,856    9.28     0.5         274.03  MPI
```

- Report scoring as textual output

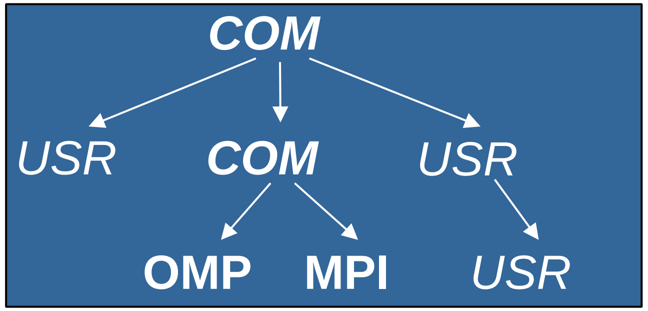
159 GB total memory
20 GB per rank!



- Region/callpath classification
  - **MPI** pure MPI functions
  - **OMP** pure OpenMP regions
  - **USR** user-level computation
  - **COM** “combined” USR+OpenMP/MPI
  - **ANY/ALL** aggregate of all region types

BT-MZ Summary Analysis Report Breakdown

```
% scorep-score -r scorep_bt-mz_C.8x7.<jobid>/profile.cubex
  [...]
  [...]
flt     type     max_buf[B]          visits time[s] time[%] time/visit[us]  region
         ALL 21,389,438,207 6,557,153,121 1934.82   100.0           0.30  ALL
         USR 21,309,225,314 6,537,020,537  835.65    43.2           0.13  USR
         OMP     76,450,336    19,013,888 1087.70    56.2          57.21  OMP
         COM      3,525,730     1,084,840    2.20     0.1           2.03  COM
         MPI        236,827        33,856    9.28     0.5         274.03  MPI

         USR  6,883,222,086 2,110,313,472  381.69    19.7           0.18  binvcrhs_
         USR  6,883,222,086 2,110,313,472  163.88     8.5           0.08  matvec_sub_
         USR  6,883,222,086 2,110,313,472  262.46    13.6           0.12  matmul_sub_
         USR    293,617,584    87,475,200    9.83     0.5           0.11  binvrhs_
         USR    293,617,584    87,475,200   14.86     0.8           0.17  lhsinit_
         USR    101,320,128    31,129,600    2.37     0.1           0.08  exact_solution_
```



More than
19 GB just for these 6 regions

BT-MZ Summary Analysis Score

- Summary measurement analysis score reveals
  - Total size of event trace would be ~159 GB
  - Maximum trace buffer size would be ~20 GB per rank
    - smaller buffer would require flushes to disk during measurement resulting in substantial perturbation
  - 99.7% of the trace requirements are for USR regions
    - purely computational routines never found on COM call-paths common to communication routines or OpenMP parallel regions
  - These USR regions contribute around 43% of total time
    - however, much of that is very likely to be measurement overhead for frequently-executed small routines
- Advisable to tune measurement configuration
  - Specify an adequate trace buffer size
  - Specify a filter file listing (USR) regions not to be measured

BT-MZ Summary Analysis Report Filtering

```
% cat ../config/scorep.filt
SCOREP_REGION_NAMES_BEGIN EXCLUDE
binvcrhs*
matmul_sub*
matvec_sub*
exact_solution*
binvrhs*
lhs*init*
timer_*


% scorep-score -f ../config/scorep.filt [-c 2] \
> scorep_bt-mz_C.8x7.<jobid>/profile.cubex

Estimated aggregate size of event trace:                   612MB
Estimated requirements for largest trace buffer (max_buf): 77MB
Estimated memory requirements (SCOREP_TOTAL_MEMORY):       91MB
(hint: When tracing set SCOREP_TOTAL_MEMORY=91MB to avoid
 intermediate flushes or reduce requirements using USR
 regions filters.)
```

- Report scoring with prospective filter listing 6 USR regions

612 MB of memory in total,
91 MB per rank!

(Add space for metric values using `-c #`)

BT-MZ Summary Analysis Report Filtering

```
% scorep-score -r -f ../config/scorep.filt [-c 2] \
> scorep_bt-mz_C.8x7.<jobid>/profile.cubex

flt     type     max_buf[B]         visits time[s] time[%] time/visit[us]  region
 -       ALL 21,389,438,207 6,557,153,121 1934.82   100.0           0.30  ALL
 -       USR 21,309,225,314 6,537,020,537  835.65    43.2           0.13  USR
 -       OMP     76,450,336    19,013,888 1087.70    56.2          57.21  OMP
 -       COM      3,525,730     1,084,840    2.20     0.1           2.03  COM
 -       MPI        236,827        33,856    9.28     0.5         274.03  MPI

 *       ALL     80,212,945    20,132,593 1099.74    56.8          54.62  ALL-FLT
 +       FLT 21,309,225,262 6,537,020,528  835.08    43.2           0.13  FLT
 -       OMP     76,450,336    19,013,888 1087.70    56.2          57.21  OMP-FLT
 *       COM      3,525,730     1,084,840    2.20     0.1           2.03  COM-FLT
 -       MPI        236,827        33,856    9.28     0.5         274.03  MPI-FLT
 *       USR             52             9    0.57     0.0       63057.18  USR-FLT

 +       USR  6,883,222,086 2,110,313,472  381.69    19.7           0.18  binvcrhs_
 +       USR  6,883,222,086 2,110,313,472  163.88     8.5           0.08  matvec_sub_
 +       USR  6,883,222,086 2,110,313,472  262.46    13.6           0.12  matmul_sub_
 +       USR    293,617,584    87,475,200    9.83     0.5           0.11  binvrhs_
 +       USR    293,617,584    87,475,200   14.86     0.8           0.17  lhsinit_
 +       USR    101,320,128    31,129,600    2.37     0.1           0.08  exact_solution_
```

- Score report breakdown by region

Filtered routines marked with '+'

BT-MZ Filtered Summary Measurement

```
% cd bin.scorep
% cp ../jobscript/bridges/scalasca.sbatch.C.8 .
% less scalasca.sbatch.C.8



…
# Score-P measurement configuration
export SCOREP_EXPERIMENT_DIRECTORY=scalasca_bt-mz_${CLASS}.${PROCS}x${OMP_NUM_THREADS}.${SLURM_JOB_ID}
export SCOREP_FILTERING_FILE=../config/scorep.filt
export SCOREP_TOTAL_MEMORY=91M
#export SCOREP_METRIC_PAPI=PAPI_TOT_INS,PAPI_TOT_CYC
#export SCOREP_ENABLE_TRACING=true

# Scalasca2 configuration
#export SCAN_ANALYZE_OPTS="--time-correct"
NEXUS="scalasca -analyze -t"

$NEXUS mpirun --report-bindings -np $SLURM_NTASKS $EXE

% sbatch scalasca.sbatch.C.8
```

- Set new experiment directory and re-run measurement with new filter configuration

- Submit new job

BT-MZ Summary Analysis Report Examination

```
% ls
…
scalasca.sbatch.C.8              scalasca-C.8-<jobid2>.err
scalasca-C.8-<jobid2>.out        scalasca_bt-mz_C.8x7.<jobid2>/

% ls scalasca_bt-mz_C.8x7.<jobid2>
scorep.filt     scorep.cfg
traces/         traces.def
profile.cubex   traces.otf2
scorep.log      trace.stat
scout.cubex     scout.err
scout.log

% square -s scalasca_bt-mz_C.8x7.<jobid2>/
INFO: Post-processing runtime summarization report...
INFO: Post-processing trace analysis report...
…
```

- Creates experiment directory
  - The analysis report that was collated after measurement (profile.cubex)
  - A trace analysis was performed after the measurement (scout.cubex)
- Post-processing with square -s (scalasca -examine -s)

BT-MZ Summary Analysis Report Examination (cont.)

```
% ls scalasca_bt-mz_C.8x7.<jobid2>
scorep.filt       scorep.cfg
traces/           traces.def
profile.cubex     traces.otf2
scorep.log        trace.stat
scout.cubex       scout.err
scout.log         summary.cubex
trace.cubex       scorep.score



laptop> scp userid@bridges.psc.edu:~/NPB3.3-MZ-MPI/bin.scorep/\
> scalasca_bt-mz_C.8x7.<jobid2>/trace.* .

laptop> cube trace.cubex

       [cube GUI showing trace analysis report]
```

- Creates experiment directory
  - A record of the measurement configuration (scorep.cfg)
  - The analysis report that was collated after measurement (profile.cubex)
- Interactive exploration with CUBE

Hint:
Copy '*.cubex' to your laptop using 'scp' to improve responsiveness of GUI

Scalasca: Reference material

Scalasca command – One command for (almost) everything

```
% scalasca
Scalasca 2.3.1
Toolset for scalable performance analysis of large-scale parallel applications
usage: scalasca [OPTION]... ACTION <argument>...
    1. prepare application objects and executable for measurement:
       scalasca -instrument <compile-or-link-command> # skin (using scorep)
    2. run application under control of measurement system:
       scalasca -analyze <application-launch-command> # scan
    3. interactively explore measurement analysis report:
       scalasca -examine <experiment-archive|report>  # square

Options:
   -c, --show-config     show configuration summary and exit
   -h, --help            show this help and exit
   -n, --dry-run         show actions without taking them
       --quickref        show quick reference guide and exit
       --remap-specfile  show path to remapper specification file and exit
   -v, --verbose         enable verbose commentary
   -V, --version         show version information and exit
```

- The ‘`scalasca -instrument`’ command is deprecated and only provided for backwards compatibility with Scalasca 1.x., recommended: use Score-P instrumenter directly

Scalasca compatibility command: skin

```
% skin
Scalasca 2.3.1: application instrumenter (using Score-P instrumenter)
usage: skin [-v] [-comp] [-pdt] [-pomp] [-user] [--*] <compile-or-link-command>
  -comp={all|none|...}: routines to be instrumented by compiler [default: all]
                 (... custom instrumentation specification depends on compiler)
  -pdt:  process source files with PDT/TAU instrumenter
  -pomp: process source files for POMP directives
  -user: enable EPIK user instrumentation API macros in source code
  -v:    enable verbose commentary when instrumenting

  --*:   options to pass to Score-P instrumenter
```

- Scalasca application instrumenter
  - Provides compatibility with Scalasca 1.x
  - **Deprecated! Use Score-P instrumenter directly.**

Scalasca convenience command: scan

```
% scan
Scalasca 2.3.1: measurement collection & analysis nexus
usage: scan {options} [launchcmd [launchargs]] target [targetargs]
      where {options} may include:
  -h    Help: show this brief usage message and exit.
  -v    Verbose: increase verbosity.
  -n    Preview: show command(s) to be launched but don't execute.
  -q    Quiescent: execution with neither summarization nor tracing.
  -s    Summary: enable runtime summarization. [Default]
  -t    Tracing: enable trace collection and analysis.
  -a    Analyze: skip measurement to (re-)analyze an existing trace.
  -e exptdir   : Experiment archive to generate and/or analyze.
                 (overrides default experiment archive title)
  -f filtfile  : File specifying measurement filter.
  -l lockfile  : File that blocks start of measurement.
  -m metrics   : Metric specification for measurement.
```

- Scalasca measurement collection & analysis nexus

Scalasca advanced command: scout - Scalasca automatic trace analyzer

```
% scout.hyb --help
SCOUT   Copyright (c) 1998-2016 Forschungszentrum Juelich GmbH
        Copyright (c) 2009-2014 German Research School for Simulation
                                Sciences GmbH

Usage: <launchcmd> scout.hyb [OPTION]... <ANCHORFILE | EPIK DIRECTORY>
Options:
  --statistics       Enables instance tracking and statistics [default]
  --no-statistics    Disables instance tracking and statistics
  --critical-path    Enables critical-path analysis [default]
  --no-critical-path Disables critical-path analysis
  --rootcause        Enables root-cause analysis [default]
  --no-rootcause     Disables root-cause analysis
  --single-pass      Single-pass forward analysis only
  --time-correct     Enables enhanced timestamp correction
  --no-time-correct  Disables enhanced timestamp correction [default]
  --verbose, -v      Increase verbosity
  --help             Display this information and exit
```

- Provided in serial (.ser), OpenMP (.omp), MPI (.mpi) and MPI+OpenMP (.hyb) variants

Scalasca advanced command: clc_synchronize

- Scalasca trace event timestamp consistency correction

```
Usage: <launchcmd> clc_synchronize.hyb <ANCHORFILE | EPIK_DIRECTORY>
```

  - Provided in MPI (.mpi) and MPI+OpenMP (.hyb) variants
  - Takes as input a trace experiment archive where the events may have timestamp inconsistencies
    - E.g., multi-node measurements on systems without adequately synchronized clocks on each compute node
  - Generates a new experiment archive (always called ./clc_sync) containing a trace with event timestamp inconsistencies resolved
    - E.g., suitable for detailed examination with a time-line visualizer

Scalasca convenience command: square

```
% square
Scalasca 2.3.1: analysis report explorer
usage: square [-v] [-s] [-f filtfile] [-F] <experiment archive | cube file>
   -c <none | quick | full> : Level of sanity checks for newly created reports
   -F                       : Force remapping of already existing reports
   -f filtfile              : Use specified filter file when doing scoring
   -s                       : Skip display and output textual score report
   -v                       : Enable verbose mode
   -n                       : Do not include idle thread metric
```

- Scalasca analysis report explorer

Automatic measurement configuration

- scan configures Score-P measurement by automatically setting some environment variables and exporting them
  - E.g., experiment title, profiling/tracing mode, filter file, …
  - Precedence order:
    - Command-line arguments
    - Environment variables already set
    - Automatically determined values
- Also, scan includes consistency checks and prevents corrupting existing experiment directories
- For tracing experiments, after trace collection completes then automatic parallel trace analysis is initiated
  - Uses identical launch configuration to that used for measurement (i.e., the same allocated compute resources)

Further information

Scalable performance **a**nalysis of **la**rge-**sc**ale parallel **a**pplications

- Toolset for scalable performance measurement & analysis of MPI, OpenMP & hybrid parallel applications
- Supporting most popular HPC computer systems
- Available under 3-clause BSD open-source license
- Sources, documentation & publications:
  - http://www.scalasca.org
  - mailto: scalasca@fz-juelich.de

scalasca

International HPC Summer School 2016: Analysis report examination with Cube

Christian Feld
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

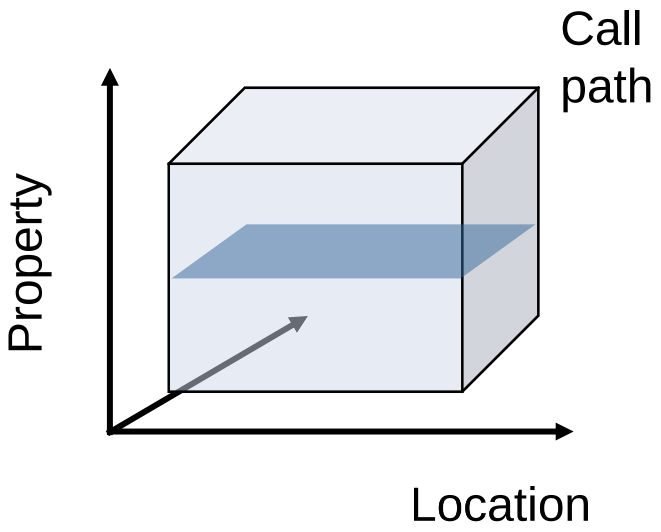
- Now available as a separate component
  - Can be installed independently of Score-P, e.g., on laptop or desktop
  - Latest release: Cube 4.3.4 (April 2016)

Analysis presentation and exploration

- Representation of values (severity matrix) on three hierarchical axes
  - Performance property (metric)
  - Call path (program location)
  - System location (process/thread)

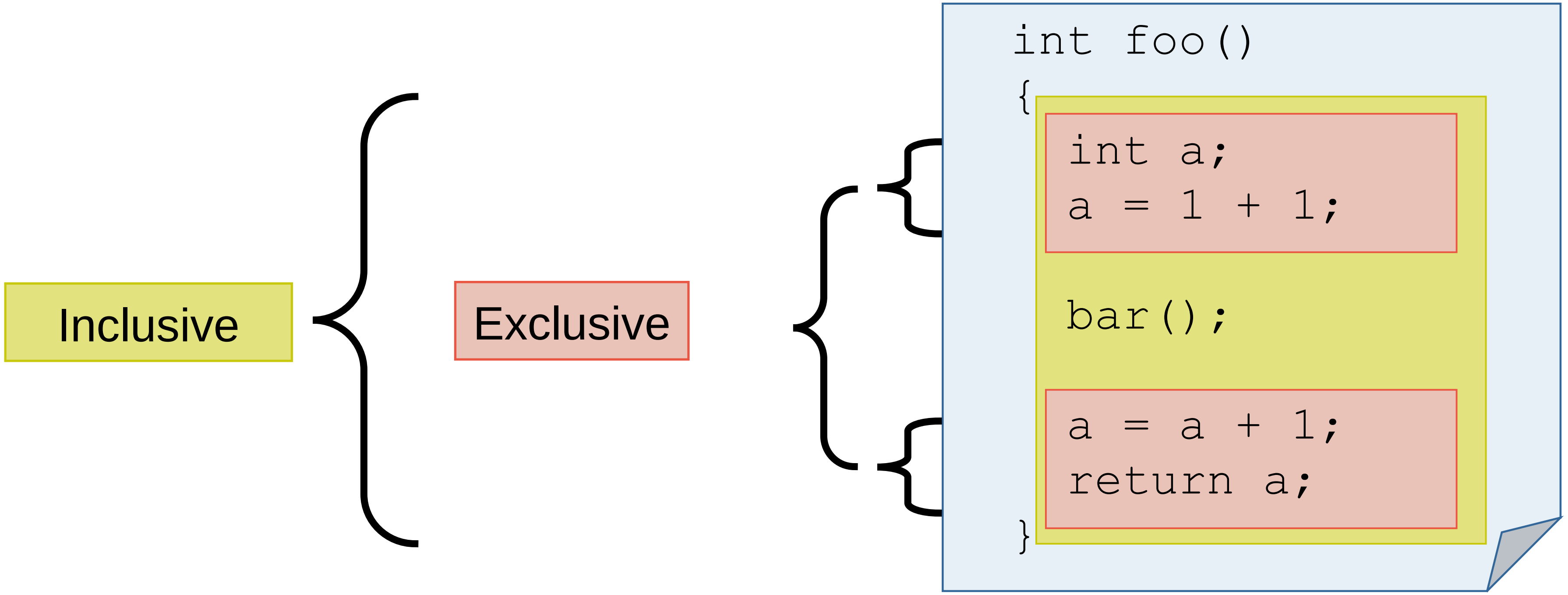
- Three coupled tree browsers

- 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



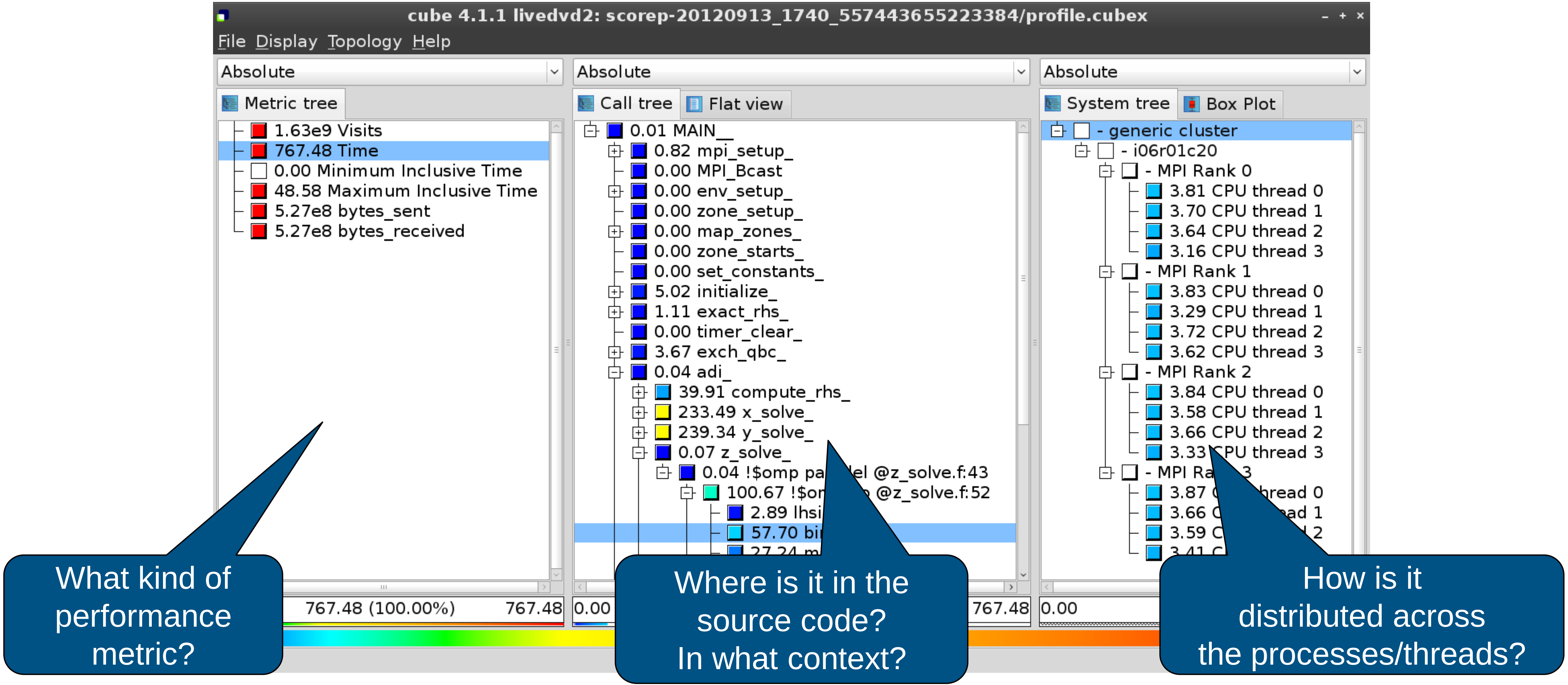
Inclusive vs. exclusive values

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



```
int foo()
{
  int a;
  a = 1 + 1;

  bar();

  a = a + 1;
  return a;
}
```

Analysis presentation



cube 4.1.1 livedvd2: scorep-20120913_1740_557443655223384/profile.cubex

```
1.63e9 Visits
767.48 Time
0.00 Minimum Inclusive Time
48.58 Maximum Inclusive Time
5.27e8 bytes_sent
5.27e8 bytes_received
```

```
0.01 MAIN__
  0.82 mpi_setup_
  0.00 MPI_Bcast
  0.00 env_setup_
  0.00 zone_setup_
  0.00 map_zones_
  0.00 zone_starts_
  0.00 set_constants_
  5.02 initialize_
  1.11 exact_rhs_
  0.00 timer_clear_
  3.67 exch_qbc_
  0.04 adi_
    39.91 compute_rhs_
    233.49 x_solve_
    239.34 y_solve_
    0.07 z_solve_
      0.04 !$omp pa   lel @z_solve.f:43
        100.67 !$or    @z_solve.f:52
          2.89 lhsi
          57.70 bi
          27.24 m
```

```
- generic cluster
  - i06r01c20
    - MPI Rank 0
      3.81 CPU thread 0
      3.70 CPU thread 1
      3.64 CPU thread 2
      3.16 CPU thread 3
    - MPI Rank 1
      3.83 CPU thread 0
      3.29 CPU thread 1
      3.72 CPU thread 2
      3.62 CPU thread 3
    - MPI Rank 2
      3.84 CPU thread 0
      3.58 CPU thread 1
      3.66 CPU thread 2
      3.33 CPU thread 3
    - MPI Ra   3
      3.87 C    hread 0
      3.66 C    ead 1
      3.59 C      2
      3.41 C
```

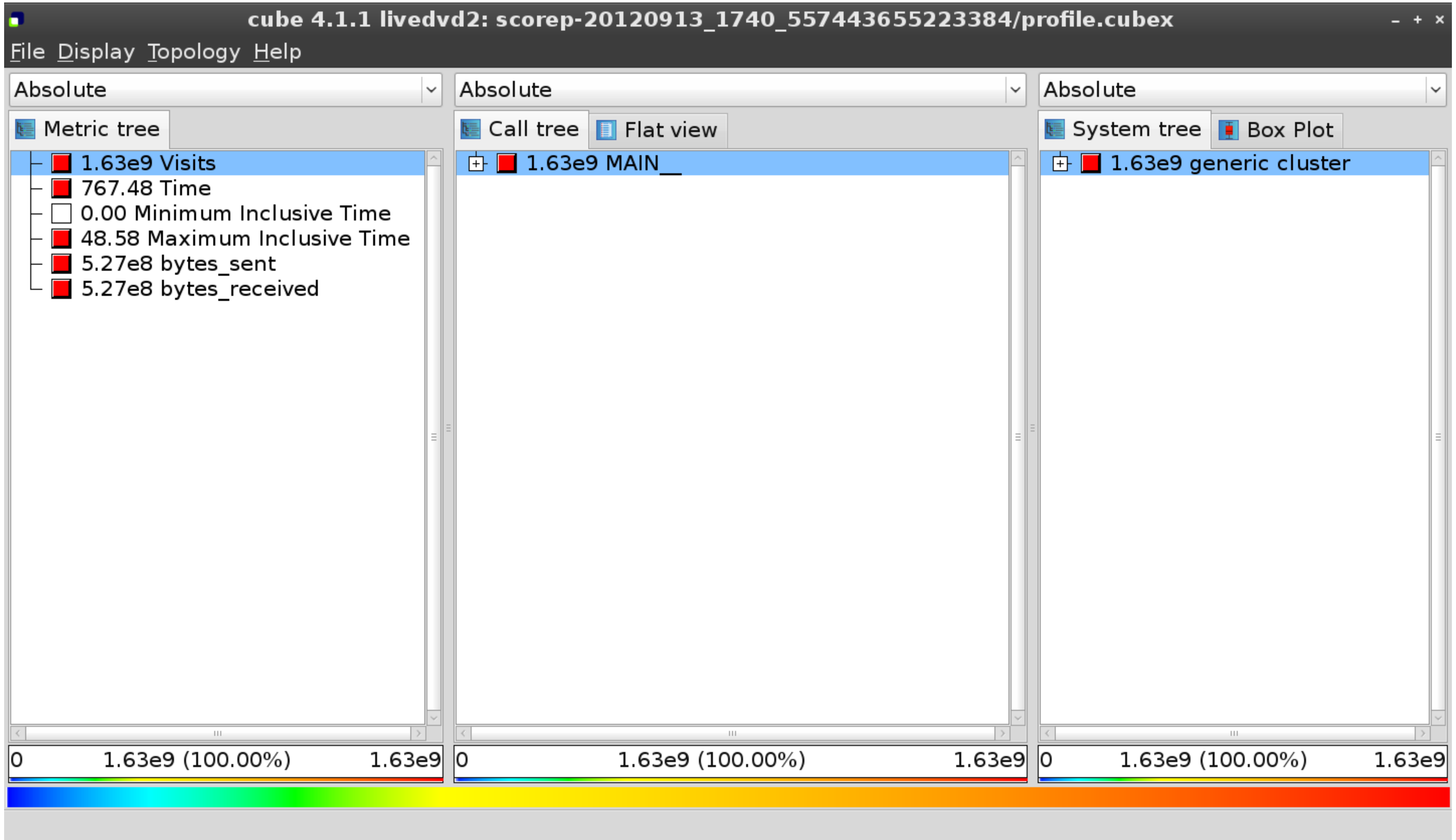
767.48 (100.00%) 767.48 | 0.00 767.48 | 0.00

What kind of performance metric?

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

How is it distributed across the processes/threads?

Score-P analysis report exploration (opening view)



cube 4.1.1 livedvd2: scorep-20120913_1740_557443655223384/profile.cubex

File Display Topology Help
Absolute
Metric tree
Absolute
Call tree
Flat view
Absolute
System tree
Box Plot

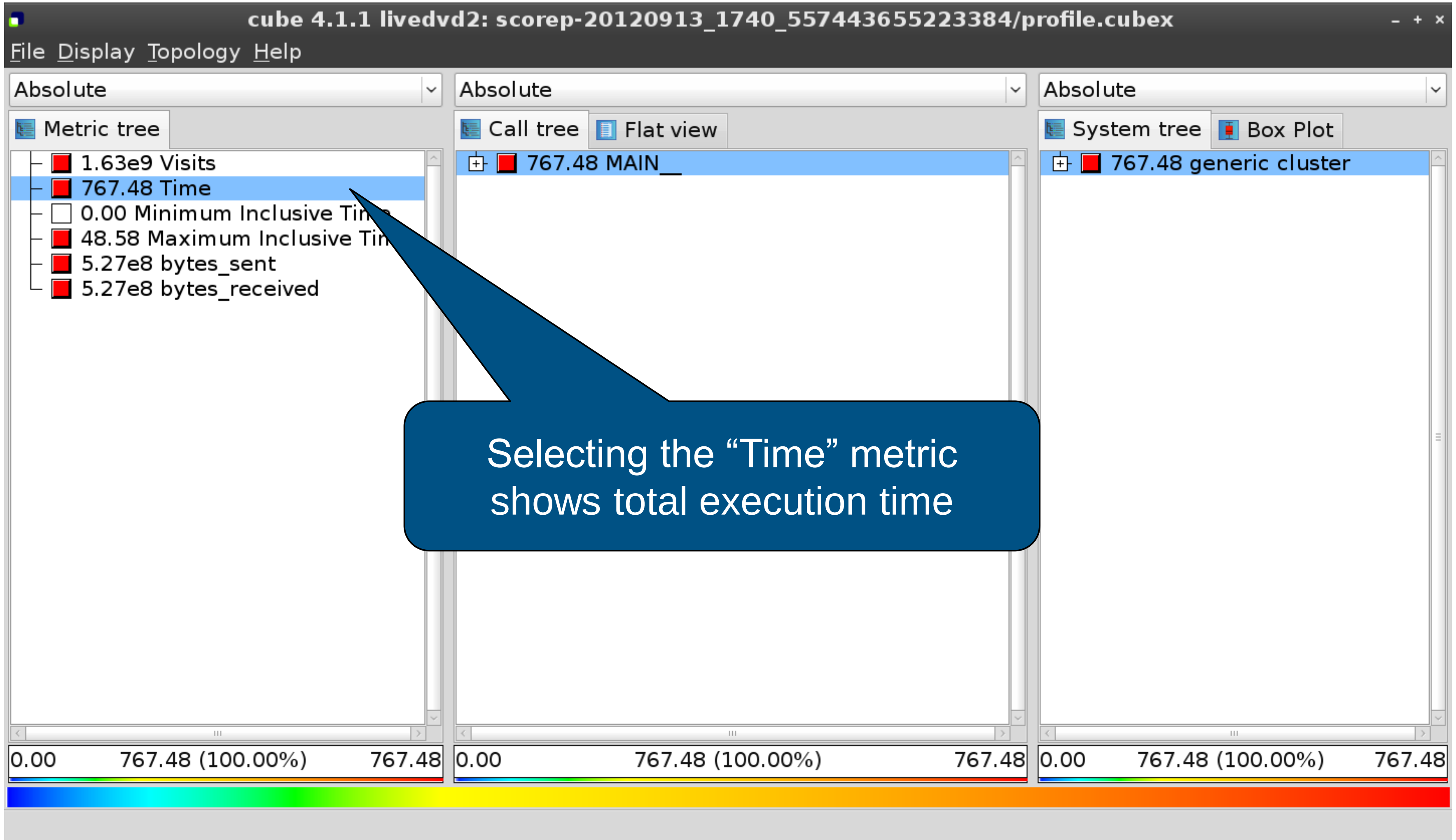
Metric tree:
- 🟥 1.63e9 Visits
- 🟥 767.48 Time
- □ 0.00 Minimum Inclusive Time
- 🟥 48.58 Maximum Inclusive Time
- 🟥 5.27e8 bytes_sent
- 🟥 5.27e8 bytes_received

Call tree:
- ⊞ 🟥 1.63e9 MAIN__

System tree:
- ⊞ 🟥 1.63e9 generic cluster

0 1.63e9 (100.00%) 1.63e9
0 1.63e9 (100.00%) 1.63e9
0 1.63e9 (100.00%) 1.63e9

Metric selection



cube 4.1.1 livedvd2: scorep-20120913_1740_557443655223384/profile.cubex

- 🟥 1.63e9 Visits
- 🟥 767.48 Time
- □ 0.00 Minimum Inclusive Time
- 🟥 48.58 Maximum Inclusive Time
- 🟥 5.27e8 bytes_sent
- 🟥 5.27e8 bytes_received

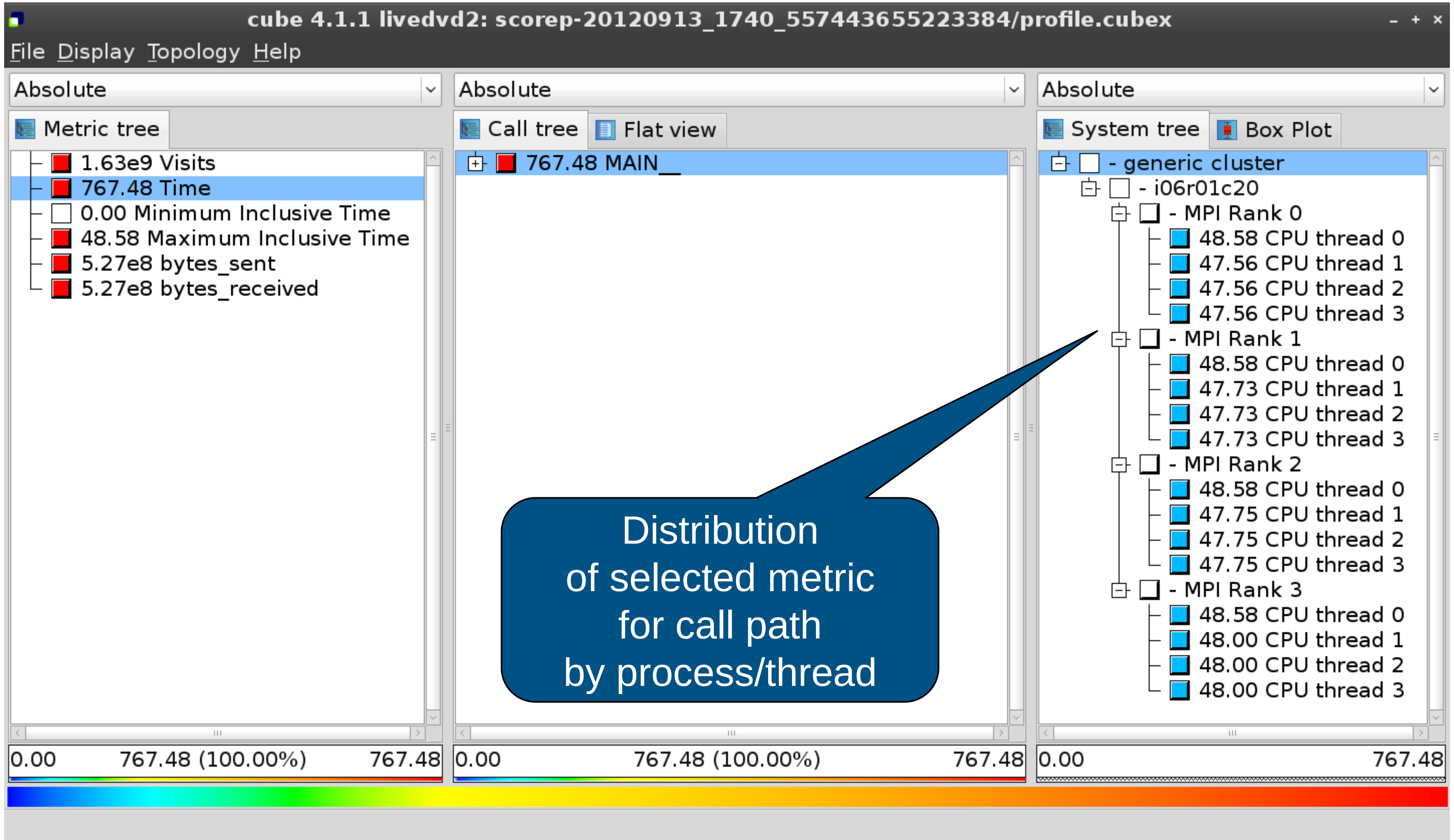
⊞ 🟥 767.48 MAIN__

⊞ 🟥 767.48 generic cluster

0.00 767.48 (100.00%) 767.48

Selecting the “Time” metric shows total execution time

Expanding the system tree



cube 4.1.1 livedvd2: scorep-20120913_1740_557443655223384/profile.cubex

Metric tree:

- 🟥 1.63e9 Visits
- 🟥 767.48 Time
- □ 0.00 Minimum Inclusive Time
- 🟥 48.58 Maximum Inclusive Time
- 🟥 5.27e8 bytes_sent
- 🟥 5.27e8 bytes_received

Call tree:

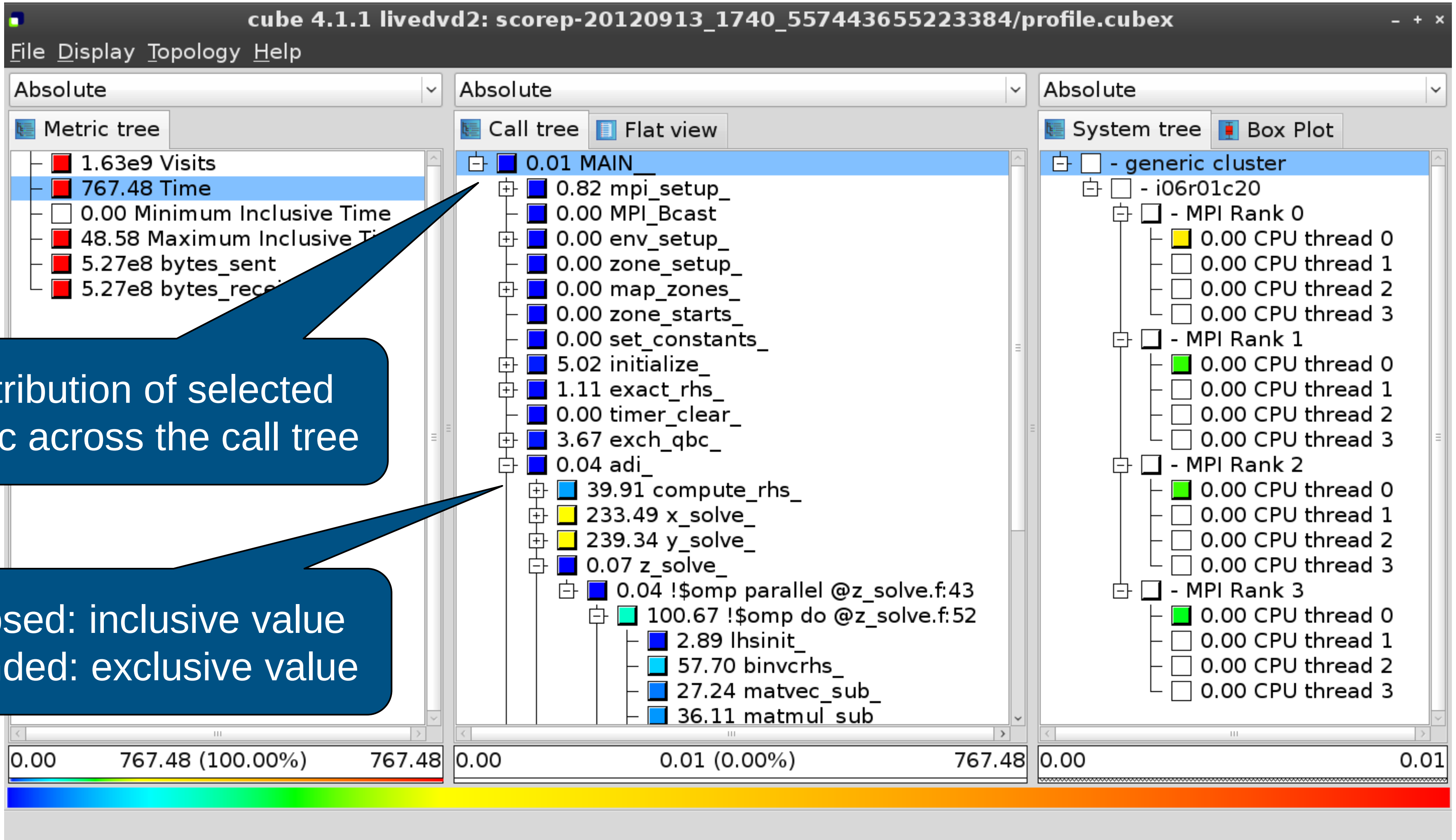
- ⊞ 🟥 767.48 MAIN__

System tree:

- □ - generic cluster
  - □ - i06r01c20
    - □ - MPI Rank 0
      - 🟦 48.58 CPU thread 0
      - 🟦 47.56 CPU thread 1
      - 🟦 47.56 CPU thread 2
      - 🟦 47.56 CPU thread 3
    - □ - MPI Rank 1
      - 🟦 48.58 CPU thread 0
      - 🟦 47.73 CPU thread 1
      - 🟦 47.73 CPU thread 2
      - 🟦 47.73 CPU thread 3
    - □ - MPI Rank 2
      - 🟦 48.58 CPU thread 0
      - 🟦 47.75 CPU thread 1
      - 🟦 47.75 CPU thread 2
      - 🟦 47.75 CPU thread 3
    - □ - MPI Rank 3
      - 🟦 48.58 CPU thread 0
      - 🟦 48.00 CPU thread 1
      - 🟦 48.00 CPU thread 2
      - 🟦 48.00 CPU thread 3

Distribution of selected metric for call path by process/thread

Expanding the call tree



cube 4.1.1 livedvd2: scorep-20120913_1740_557443655223384/profile.cubex

Metric tree:

- 1.63e9 Visits
- 767.48 Time
- 0.00 Minimum Inclusive Time
- 48.58 Maximum Inclusive Ti
- 5.27e8 bytes_sent
- 5.27e8 bytes_rece

Call tree:

- 0.01 MAIN__
  - 0.82 mpi_setup_
  - 0.00 MPI_Bcast
  - 0.00 env_setup_
  - 0.00 zone_setup_
  - 0.00 map_zones_
  - 0.00 zone_starts_
  - 0.00 set_constants_
  - 5.02 initialize_
  - 1.11 exact_rhs_
  - 0.00 timer_clear_
  - 3.67 exch_qbc_
  - 0.04 adi_
    - 39.91 compute_rhs_
    - 233.49 x_solve_
    - 239.34 y_solve_
    - 0.07 z_solve_
      - 0.04 !$omp parallel @z_solve.f:43
        - 100.67 !$omp do @z_solve.f:52
          - 2.89 lhsinit_
          - 57.70 binvcrhs_
          - 27.24 matvec_sub_
          - 36.11 matmul_sub_

System tree:

- generic cluster
  - i06r01c20
    - MPI Rank 0
      - 0.00 CPU thread 0
      - 0.00 CPU thread 1
      - 0.00 CPU thread 2
      - 0.00 CPU thread 3
    - MPI Rank 1
      - 0.00 CPU thread 0
      - 0.00 CPU thread 1
      - 0.00 CPU thread 2
      - 0.00 CPU thread 3
    - MPI Rank 2
      - 0.00 CPU thread 0
      - 0.00 CPU thread 1
      - 0.00 CPU thread 2
      - 0.00 CPU thread 3
    - MPI Rank 3
      - 0.00 CPU thread 0
      - 0.00 CPU thread 1
      - 0.00 CPU thread 2
      - 0.00 CPU thread 3

0.00 767.48 (100.00%) 767.48

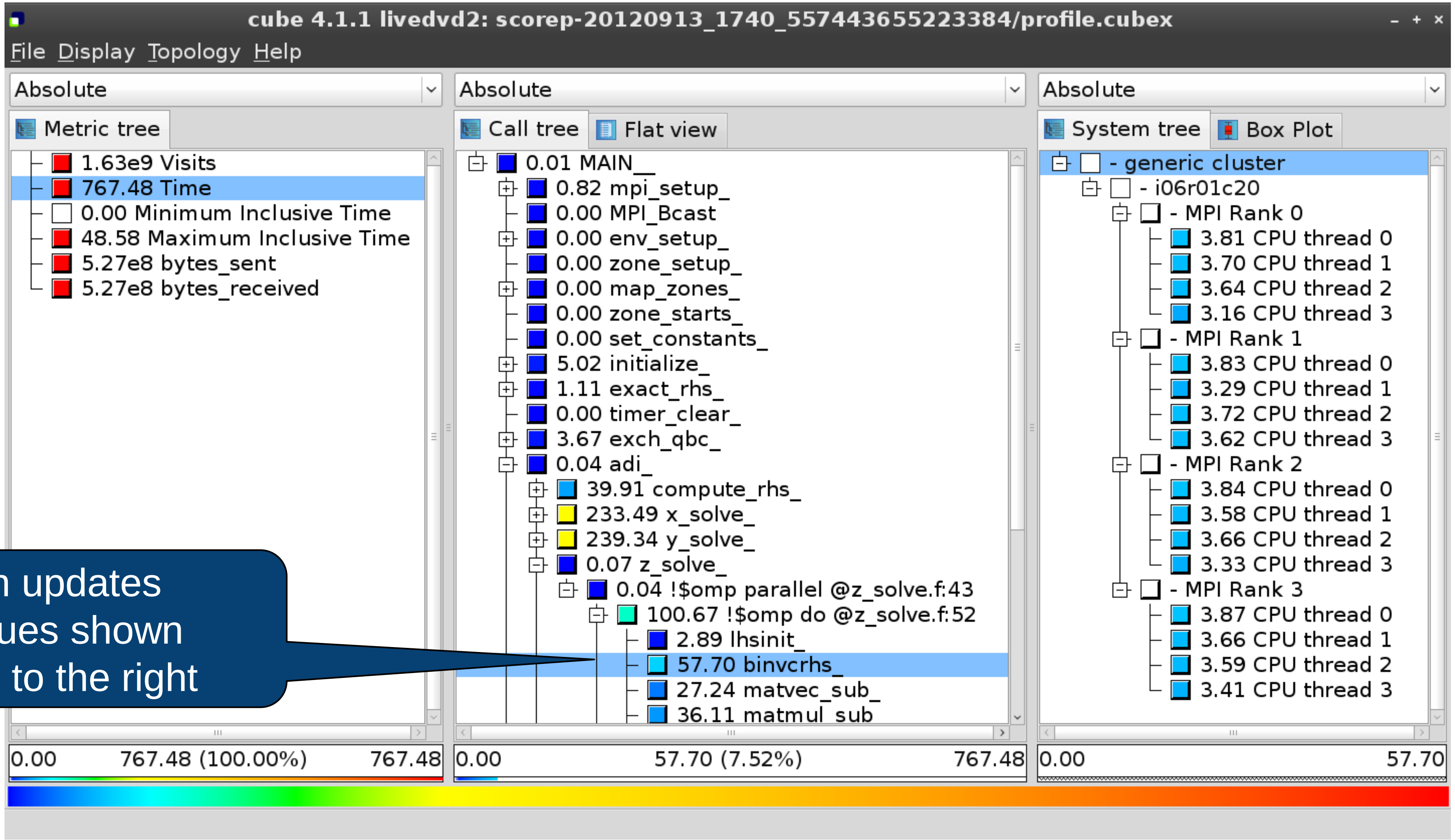
0.00 0.01 (0.00%) 767.48

0.00 0.01

Distribution of selected metric across the call tree

Collapsed: inclusive value
Expanded: exclusive value

Selecting a call path


cube 4.1.1 livedvd2: scorep-20120913_1740_557443655223384/profile.cubex
File Display Topology Help
Absolute
Metric tree
Absolute
Call tree
Flat view
Absolute
System tree
Box Plot

Metric tree:

```
├ 1.63e9 Visits
├ 767.48 Time
├ 0.00 Minimum Inclusive Time
├ 48.58 Maximum Inclusive Time
├ 5.27e8 bytes_sent
└ 5.27e8 bytes_received
```

Call tree:

```
0.01 MAIN__
  0.82 mpi_setup_
  0.00 MPI_Bcast
  0.00 env_setup_
  0.00 zone_setup_
  0.00 map_zones_
  0.00 zone_starts_
  0.00 set_constants_
  5.02 initialize_
  1.11 exact_rhs_
  0.00 timer_clear_
  3.67 exch_qbc_
  0.04 adi_
    39.91 compute_rhs_
    233.49 x_solve_
    239.34 y_solve_
    0.07 z_solve_
      0.04 !$omp parallel @z_solve.f:43
        100.67 !$omp do @z_solve.f:52
          2.89 lhsinit_
          57.70 binvcrhs_
          27.24 matvec_sub_
          36.11 matmul_sub_
```

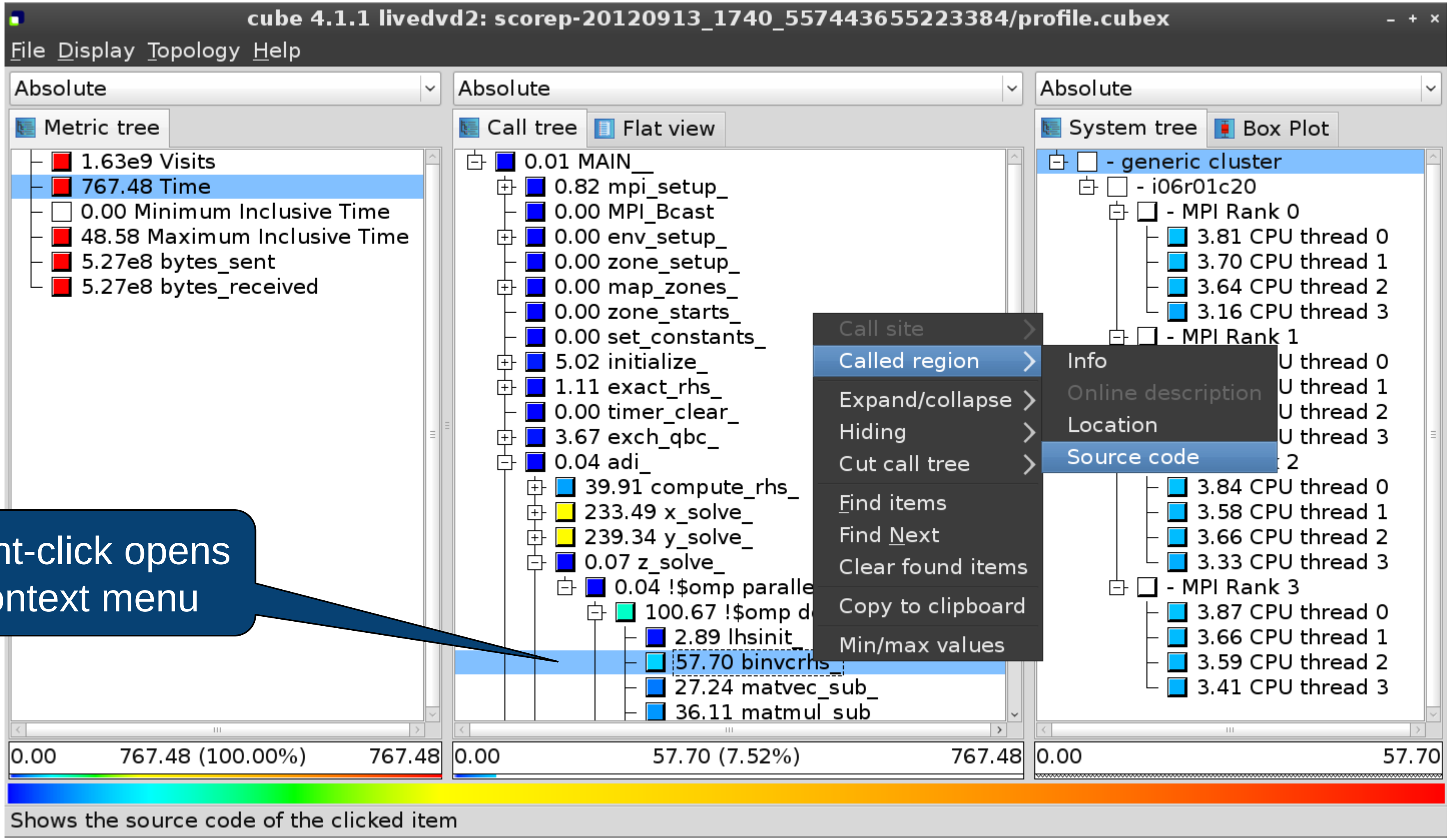
System tree:

```
- generic cluster
  - i06r01c20
    - MPI Rank 0
      3.81 CPU thread 0
      3.70 CPU thread 1
      3.64 CPU thread 2
      3.16 CPU thread 3
    - MPI Rank 1
      3.83 CPU thread 0
      3.29 CPU thread 1
      3.72 CPU thread 2
      3.62 CPU thread 3
    - MPI Rank 2
      3.84 CPU thread 0
      3.58 CPU thread 1
      3.66 CPU thread 2
      3.33 CPU thread 3
    - MPI Rank 3
      3.87 CPU thread 0
      3.66 CPU thread 1
      3.59 CPU thread 2
      3.41 CPU thread 3
```

```
0.00   767.48 (100.00%)   767.48
0.00   57.70 (7.52%)   767.48
0.00   57.70
```

Selection updates metric values shown in columns to the right

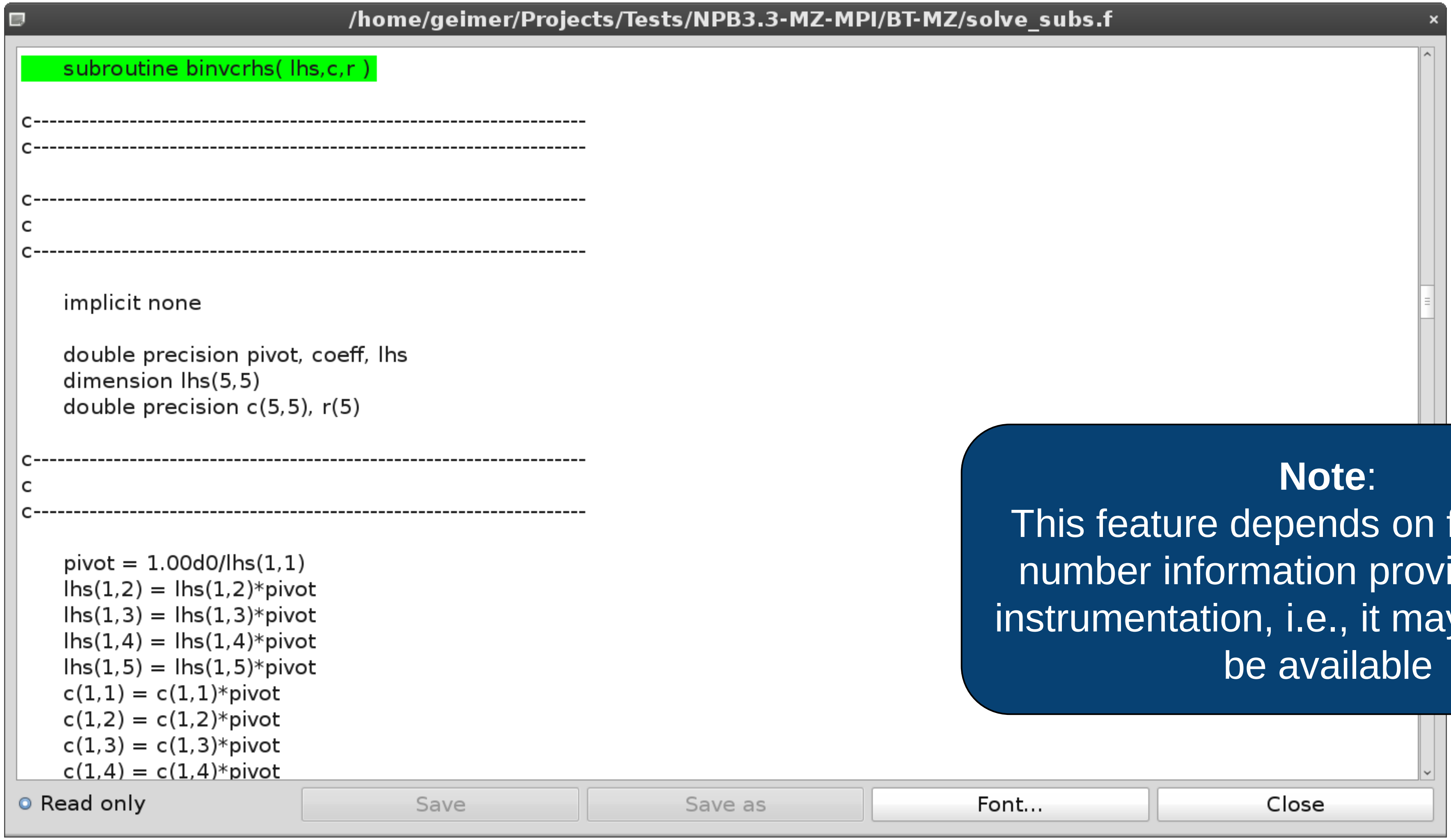
Source-code view via context menu



Shows the source code of the clicked item

Right-click opens context menu

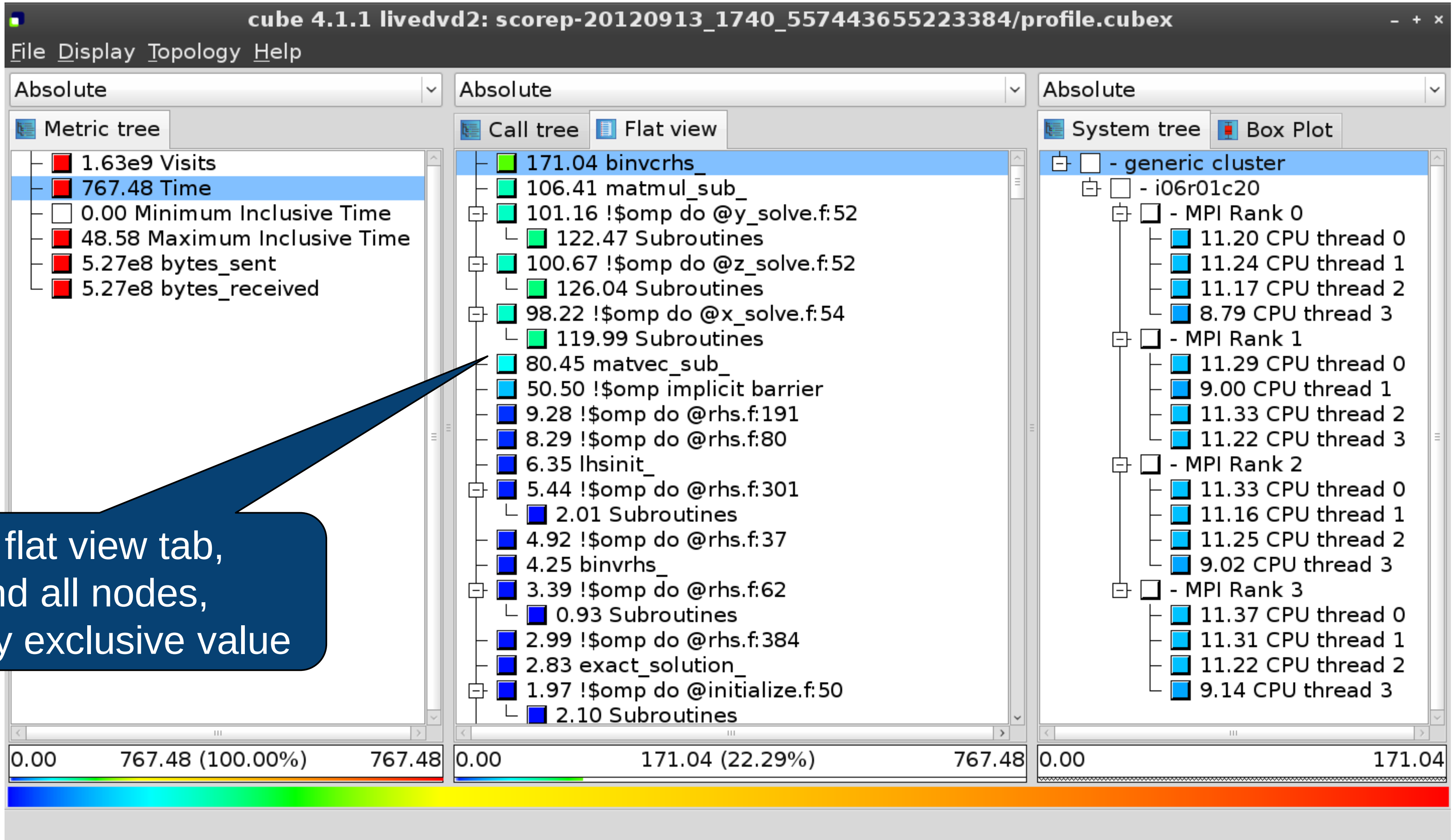
Source-code view


/home/geimer/Projects/Tests/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f
○ Read only
Save
Save as
Font...
Close

```
      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
```

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



cube 4.1.1 livedvd2: scorep-20120913_1740_557443655223384/profile.cubex

```
1.63e9 Visits
767.48 Time
0.00 Minimum Inclusive Time
48.58 Maximum Inclusive Time
5.27e8 bytes_sent
5.27e8 bytes_received
```

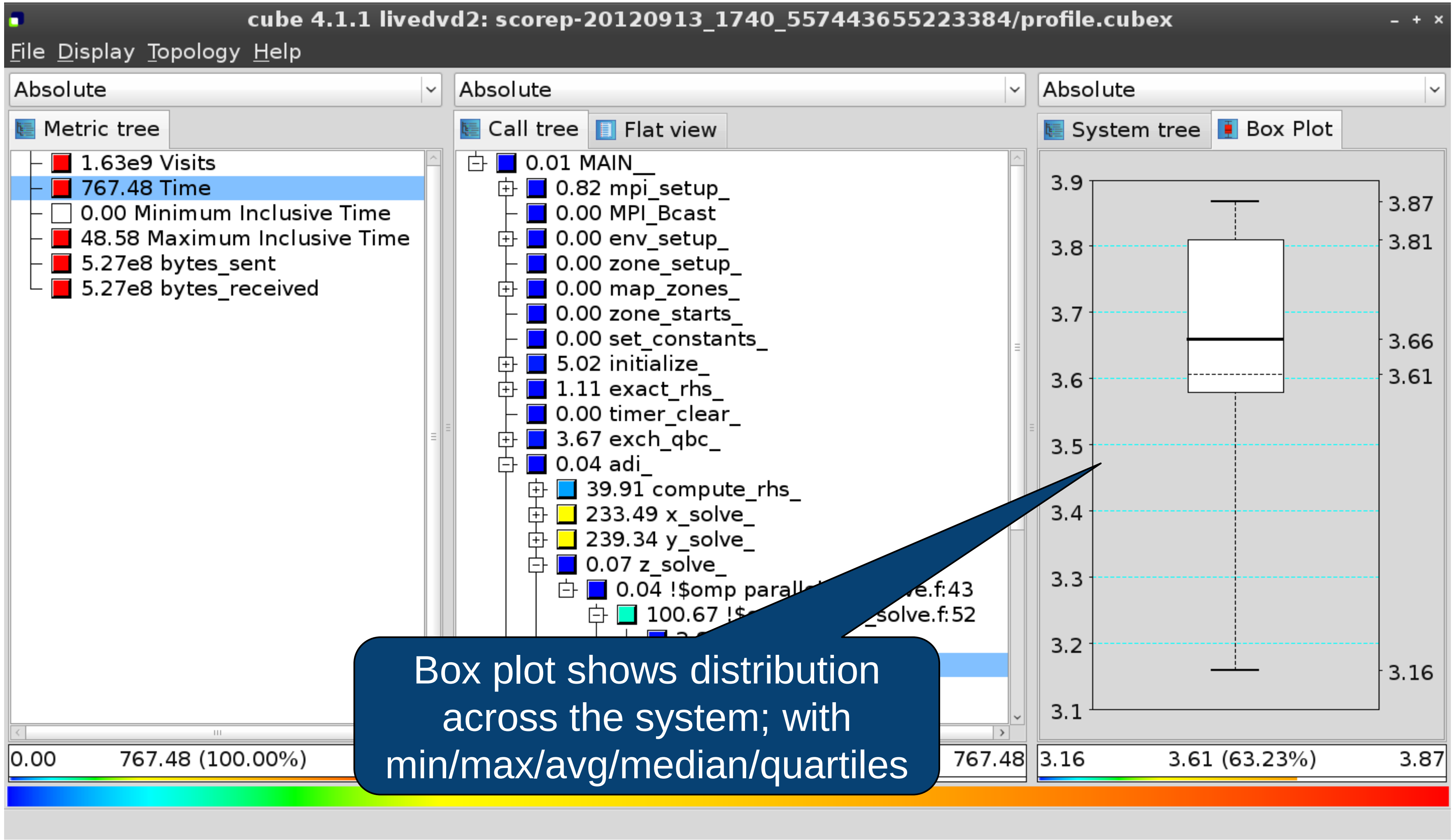
```
171.04 binvcrhs_
106.41 matmul_sub_
101.16 !$omp do @y_solve.f:52
  122.47 Subroutines
100.67 !$omp do @z_solve.f:52
  126.04 Subroutines
98.22 !$omp do @x_solve.f:54
  119.99 Subroutines
80.45 matvec_sub_
50.50 !$omp implicit barrier
9.28 !$omp do @rhs.f:191
8.29 !$omp do @rhs.f:80
6.35 lhsinit_
5.44 !$omp do @rhs.f:301
  2.01 Subroutines
4.92 !$omp do @rhs.f:37
4.25 binvrhs_
3.39 !$omp do @rhs.f:62
  0.93 Subroutines
2.99 !$omp do @rhs.f:384
2.83 exact_solution_
1.97 !$omp do @initialize.f:50
  2.10 Subroutines
```

```
- generic cluster
  - i06r01c20
    - MPI Rank 0
      11.20 CPU thread 0
      11.24 CPU thread 1
      11.17 CPU thread 2
      8.79 CPU thread 3
    - MPI Rank 1
      11.29 CPU thread 0
      9.00 CPU thread 1
      11.33 CPU thread 2
      11.22 CPU thread 3
    - MPI Rank 2
      11.33 CPU thread 0
      11.16 CPU thread 1
      11.25 CPU thread 2
      9.02 CPU thread 3
    - MPI Rank 3
      11.37 CPU thread 0
      11.31 CPU thread 1
      11.22 CPU thread 2
      9.14 CPU thread 3
```

```
0.00    767.48 (100.00%)    767.48
0.00    171.04 (22.29%)     767.48
0.00                        171.04
```

Select flat view tab,
expand all nodes,
and sort by exclusive value

Box plot view



cube 4.1.1 livedvd2: scorep-20120913_1740_557443655223384/profile.cubex

File Display Topology Help

Absolute | Absolute | Absolute

Metric tree

- 🟥 1.63e9 Visits
- 🟥 767.48 Time
- ⬜ 0.00 Minimum Inclusive Time
- 🟥 48.58 Maximum Inclusive Time
- 🟥 5.27e8 bytes_sent
- 🟥 5.27e8 bytes_received

Call tree | Flat view

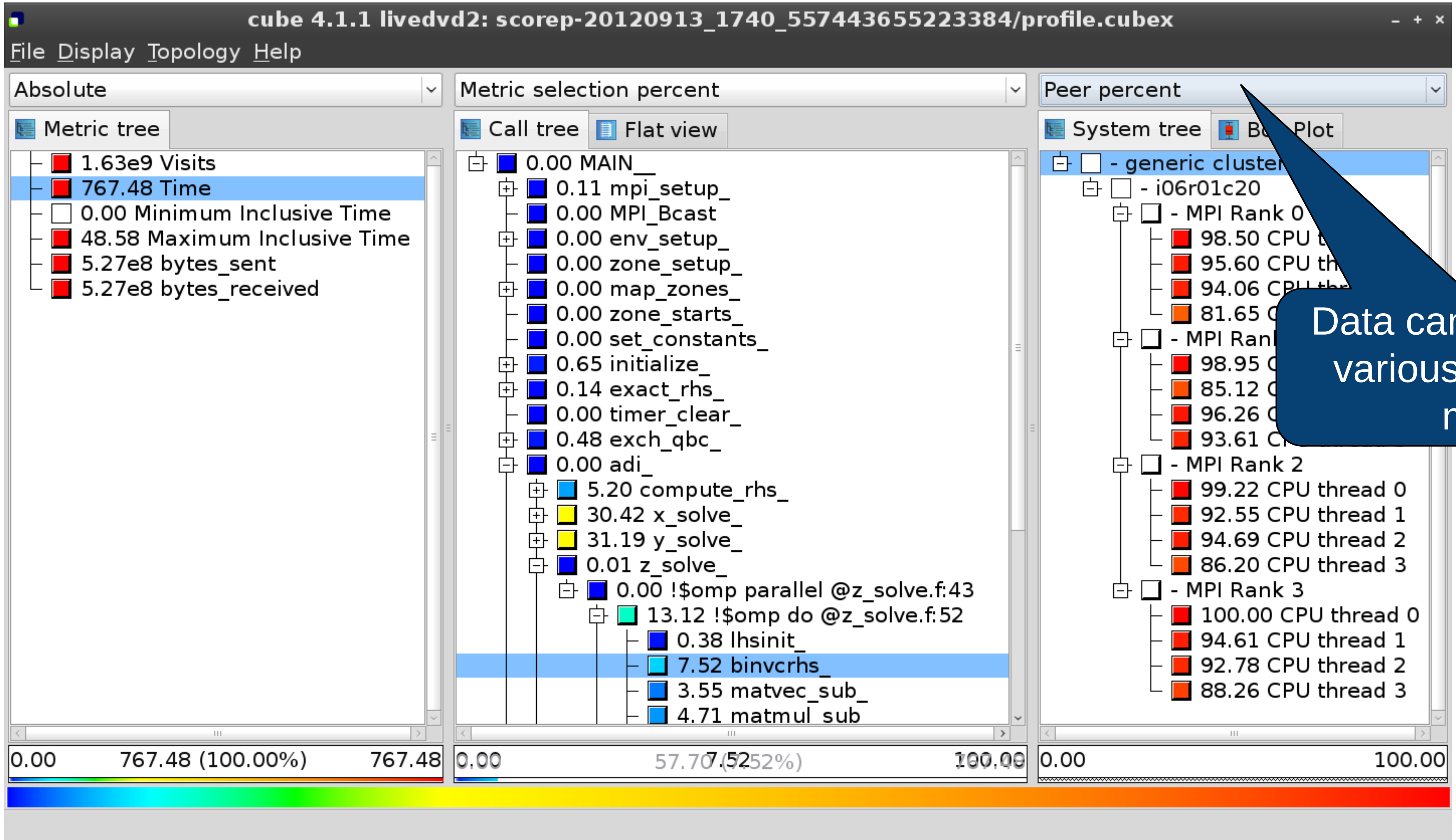
```
⊟ 🟦 0.01 MAIN__
   ⊞ 🟦 0.82 mpi_setup_
     🟦 0.00 MPI_Bcast
   ⊞ 🟦 0.00 env_setup_
     🟦 0.00 zone_setup_
   ⊞ 🟦 0.00 map_zones_
     🟦 0.00 zone_starts_
     🟦 0.00 set_constants_
   ⊞ 🟦 5.02 initialize_
   ⊞ 🟦 1.11 exact_rhs_
     🟦 0.00 timer_clear_
   ⊞ 🟦 3.67 exch_qbc_
   ⊟ 🟦 0.04 adi_
      ⊞ 🟦 39.91 compute_rhs_
      ⊞ 🟨 233.49 x_solve_
      ⊞ 🟨 239.34 y_solve_
      ⊟ 🟦 0.07 z_solve_
         ⊟ 🟦 0.04 !$omp parall      e.f:43
            ⊟ 🟩 100.67 !$          solve.f:52
```

System tree | Box Plot

0.00 767.48 (100.00%) | 767.48 | 3.16 3.61 (63.23%) 3.87

Box plot shows distribution across the system; with min/max/avg/median/quartiles

Alternative display modes



cube 4.1.1 livedvd2: scorep-20120913_1740_557443655223384/profile.cubex

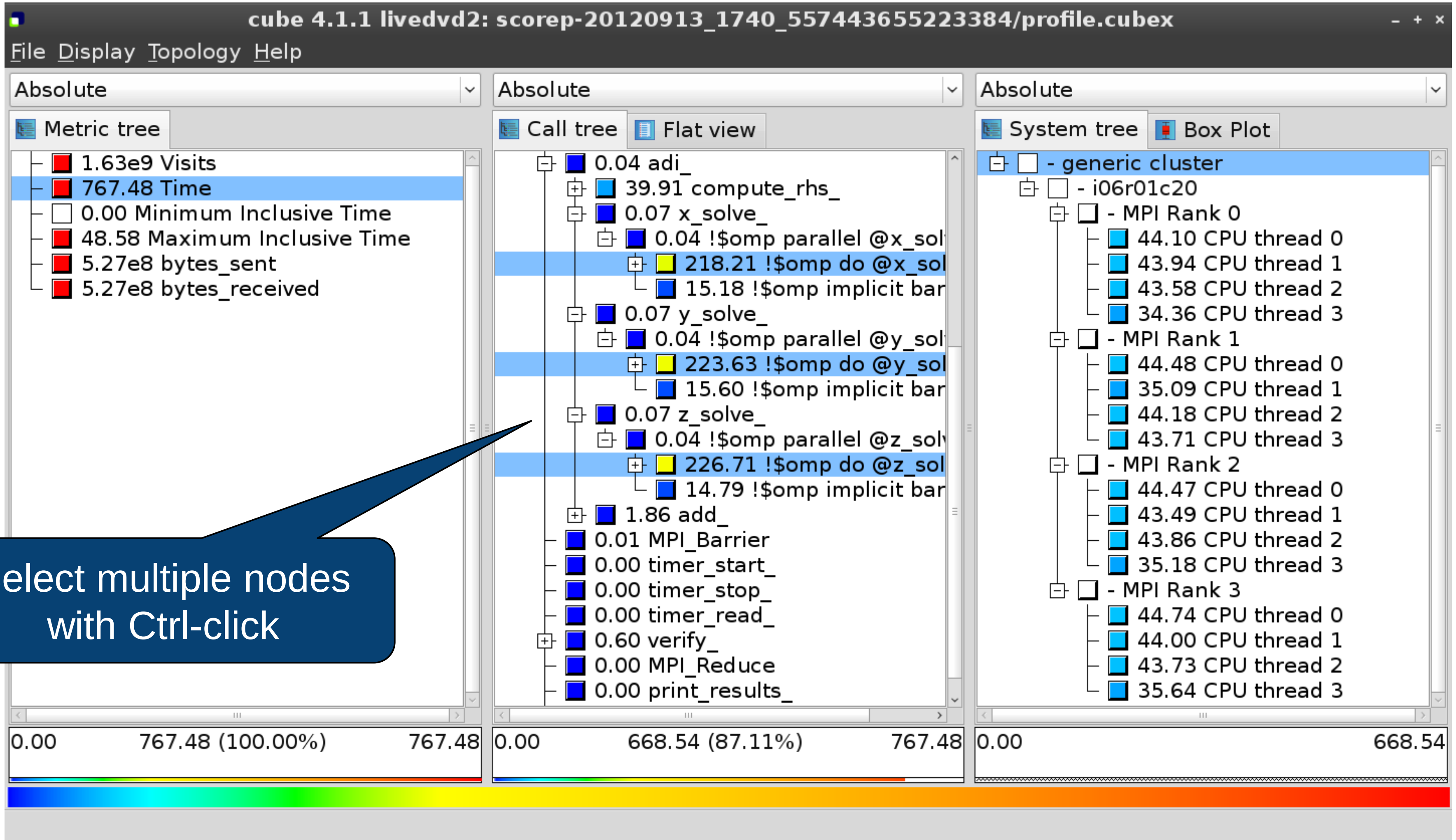
```
Metric tree:
├ 1.63e9 Visits
├ 767.48 Time
├ 0.00 Minimum Inclusive Time
├ 48.58 Maximum Inclusive Time
├ 5.27e8 bytes_sent
└ 5.27e8 bytes_received

Call tree:
0.00 MAIN__
  0.11 mpi_setup_
  0.00 MPI_Bcast
  0.00 env_setup_
  0.00 zone_setup_
  0.00 map_zones_
  0.00 zone_starts_
  0.00 set_constants_
  0.65 initialize_
  0.14 exact_rhs_
  0.00 timer_clear_
  0.48 exch_qbc_
  0.00 adi_
    5.20 compute_rhs_
    30.42 x_solve_
    31.19 y_solve_
    0.01 z_solve_
      0.00 !$omp parallel @z_solve.f:43
        13.12 !$omp do @z_solve.f:52
          0.38 lhsinit_
          7.52 binvcrhs_
          3.55 matvec_sub_
          4.71 matmul_sub

System tree:
- generic cluster
  - i06r01c20
    - MPI Rank 0
      98.50 CPU t
      95.60 CPU th
      94.06 CPU thr
      81.65 C
    - MPI Ran
      98.95 C
      85.12 C
      96.26 C
      93.61 C
    - MPI Rank 2
      99.22 CPU thread 0
      92.55 CPU thread 1
      94.69 CPU thread 2
      86.20 CPU thread 3
    - MPI Rank 3
      100.00 CPU thread 0
      94.61 CPU thread 1
      92.78 CPU thread 2
      88.26 CPU thread 3

0.00   767.48 (100.00%)   767.48
0.00   57.70 (7.52%) 7.52   100.00
0.00   100.00
```

Data can be shown in various percentage 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

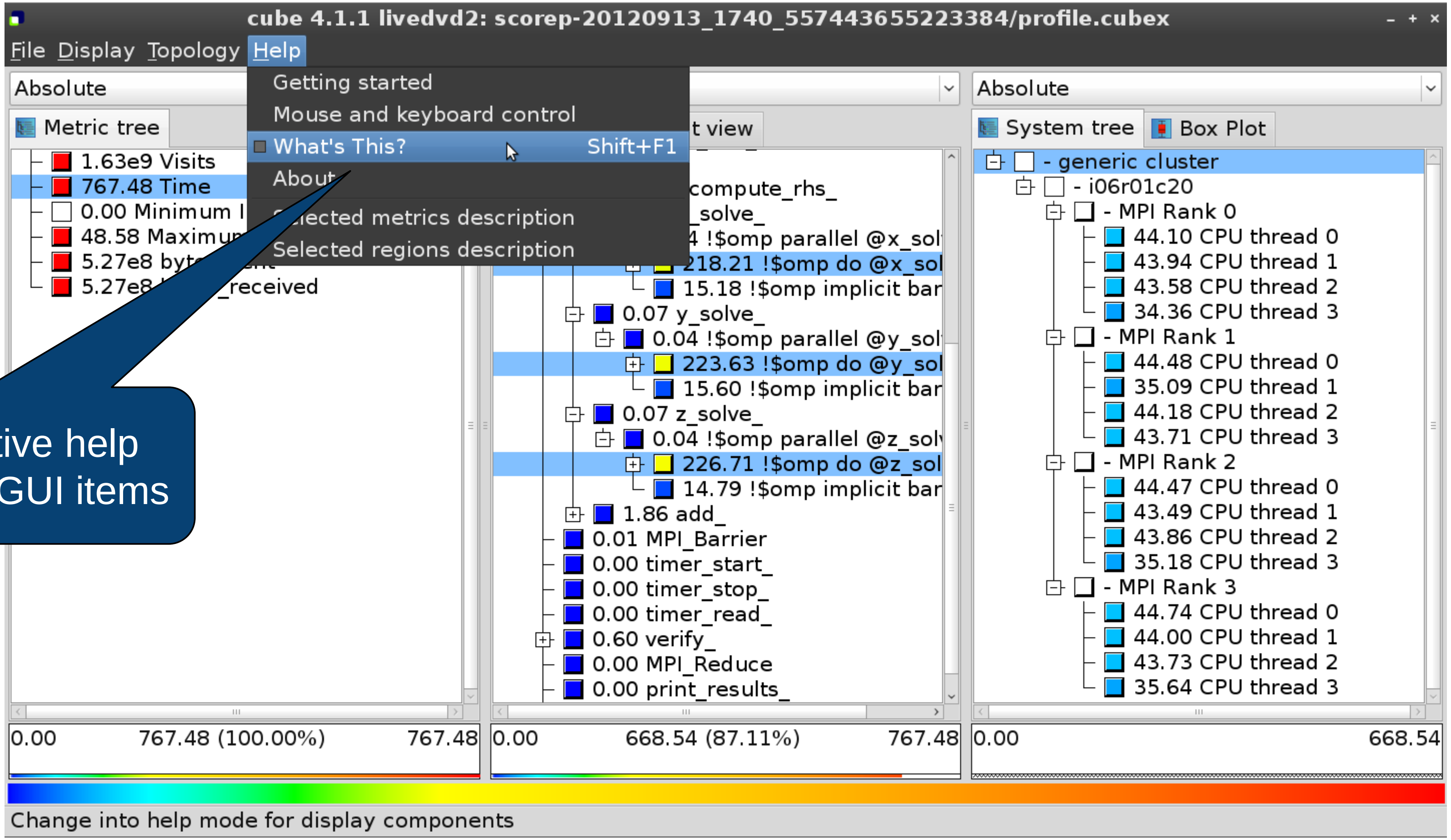


cube 4.1.1 livedvd2: scorep-20120913_1740_557443655223384/profile.cubex

```
Metric tree
1.63e9 Visits
767.48 Time
0.00 Minimum Inclusive Time
48.58 Maximum Inclusive Time
5.27e8 bytes_sent
5.27e8 bytes_received

Call tree
0.04 adi_
  39.91 compute_rhs_
  0.07 x_solve_
    0.04 !$omp parallel @x_sol
      218.21 !$omp do @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 !$omp parallel @z_solv
      226.71 !$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_

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
```

Select multiple nodes with Ctrl-click

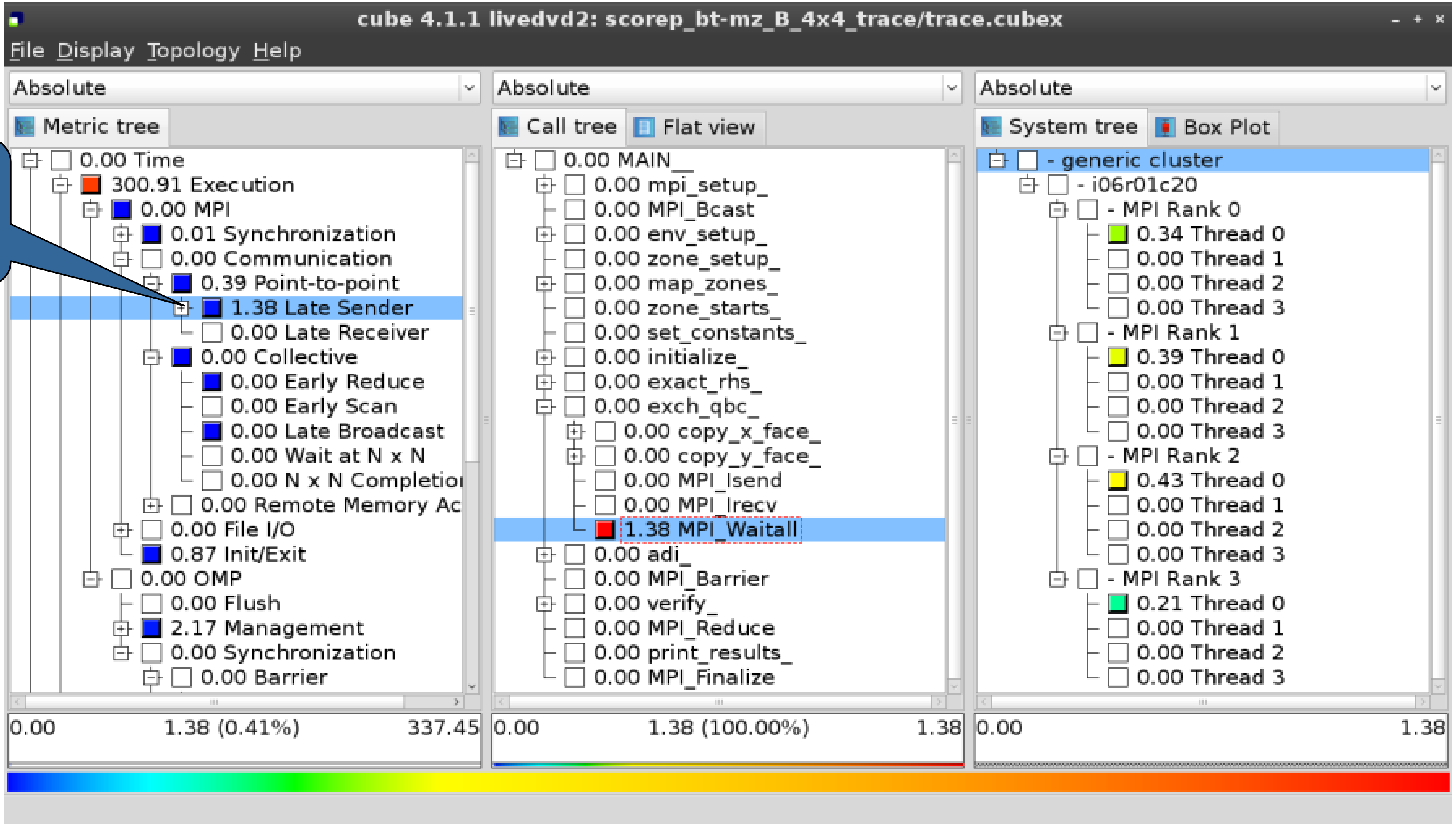
Context-sensitive help



Context-sensitive help available for all GUI items

Post-processed trace analysis report

Additional trace-based metrics in metric hierarchy


cube 4.1.1 livedvd2: scorep_bt-mz_B_4x4_trace/trace.cubex
File Display Topology Help
Absolute
Metric tree
Absolute
Call tree
Flat view
Absolute
System tree
Box Plot

```
⊟ □ 0.00 Time
  ⊟ 🟥 300.91 Execution
    ⊟ 🟦 0.00 MPI
      ⊞ 🟦 0.01 Synchronization
      ⊟ □ 0.00 Communication
        ⊟ 🟦 0.39 Point-to-point
          ⊞ 🟦 1.38 Late Sender
            □ 0.00 Late Receiver
        ⊟ 🟦 0.00 Collective
            🟦 0.00 Early Reduce
            □ 0.00 Early Scan
            🟦 0.00 Late Broadcast
            □ 0.00 Wait at N x N
            □ 0.00 N x N Completio
        ⊞ □ 0.00 Remote Memory Ac
      ⊞ □ 0.00 File I/O
        🟦 0.87 Init/Exit
    ⊟ □ 0.00 OMP
        □ 0.00 Flush
      ⊞ 🟦 2.17 Management
      ⊟ □ 0.00 Synchronization
        ⊟ □ 0.00 Barrier
```

```
0.00          1.38 (0.41%)          337.45
```

```
⊟ □ 0.00 MAIN__
  ⊞ □ 0.00 mpi_setup_
    □ 0.00 MPI_Bcast
  ⊞ □ 0.00 env_setup_
    □ 0.00 zone_setup_
  ⊞ □ 0.00 map_zones_
    □ 0.00 zone_starts_
    □ 0.00 set_constants_
  ⊞ □ 0.00 initialize_
  ⊞ □ 0.00 exact_rhs_
  ⊟ □ 0.00 exch_qbc_
    ⊞ □ 0.00 copy_x_face_
    ⊞ □ 0.00 copy_y_face_
      □ 0.00 MPI_Isend
      □ 0.00 MPI_Irecv
      🟥 1.38 MPI_Waitall
  ⊞ □ 0.00 adi_
    □ 0.00 MPI_Barrier
  ⊞ □ 0.00 verify_
    □ 0.00 MPI_Reduce
    □ 0.00 print_results_
    □ 0.00 MPI_Finalize
```

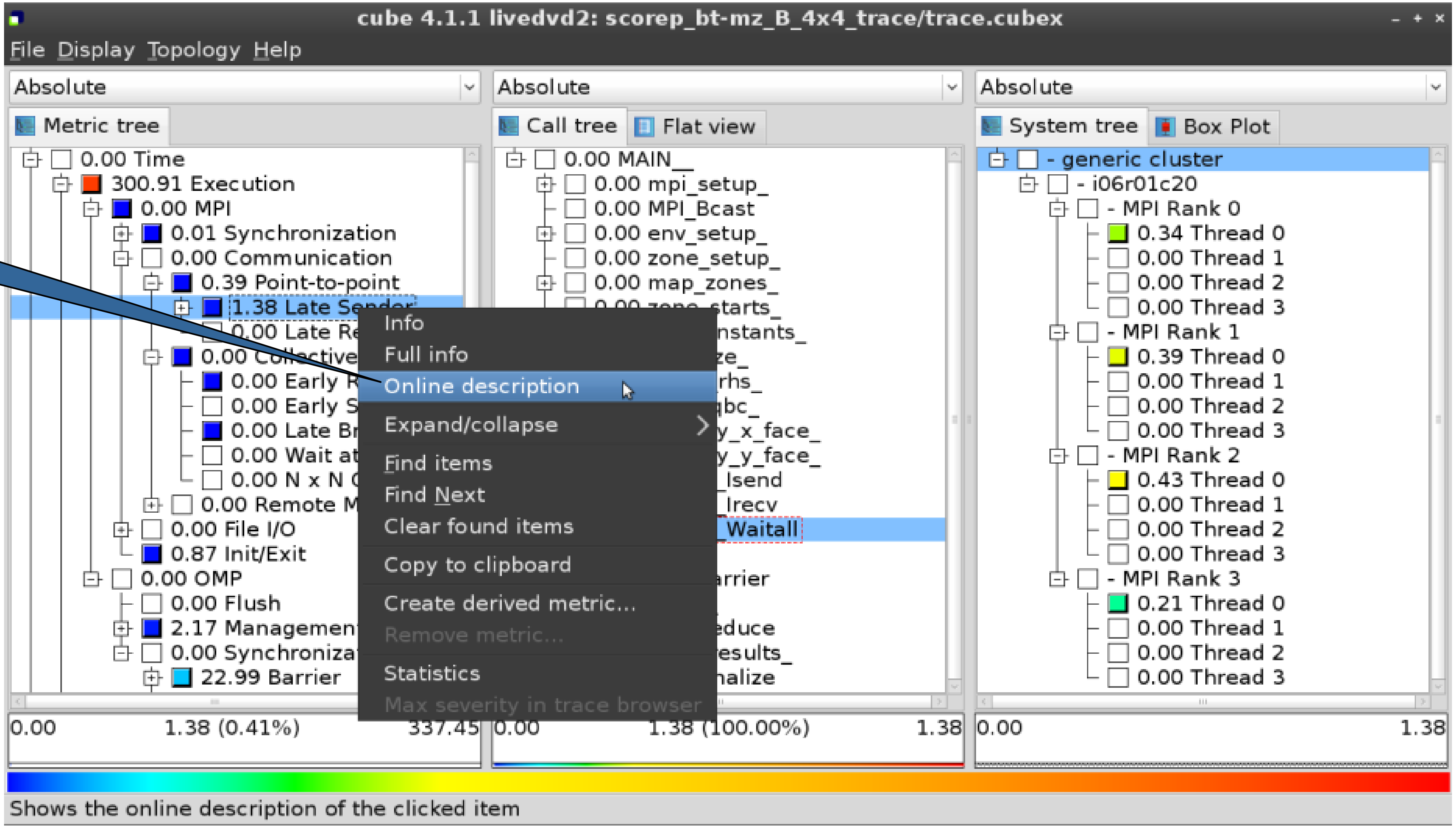
```
0.00          1.38 (100.00%)          1.38
```

```
⊟ □ - generic cluster
  ⊟ □ - i06r01c20
    ⊟ □ - MPI Rank 0
        🟩 0.34 Thread 0
        □ 0.00 Thread 1
        □ 0.00 Thread 2
        □ 0.00 Thread 3
    ⊟ □ - MPI Rank 1
        🟨 0.39 Thread 0
        □ 0.00 Thread 1
        □ 0.00 Thread 2
        □ 0.00 Thread 3
    ⊟ □ - MPI Rank 2
        🟨 0.43 Thread 0
        □ 0.00 Thread 1
        □ 0.00 Thread 2
        □ 0.00 Thread 3
    ⊟ □ - MPI Rank 3
        🟩 0.21 Thread 0
        □ 0.00 Thread 1
        □ 0.00 Thread 2
        □ 0.00 Thread 3
```

```
0.00                                  1.38
```

Online metric description

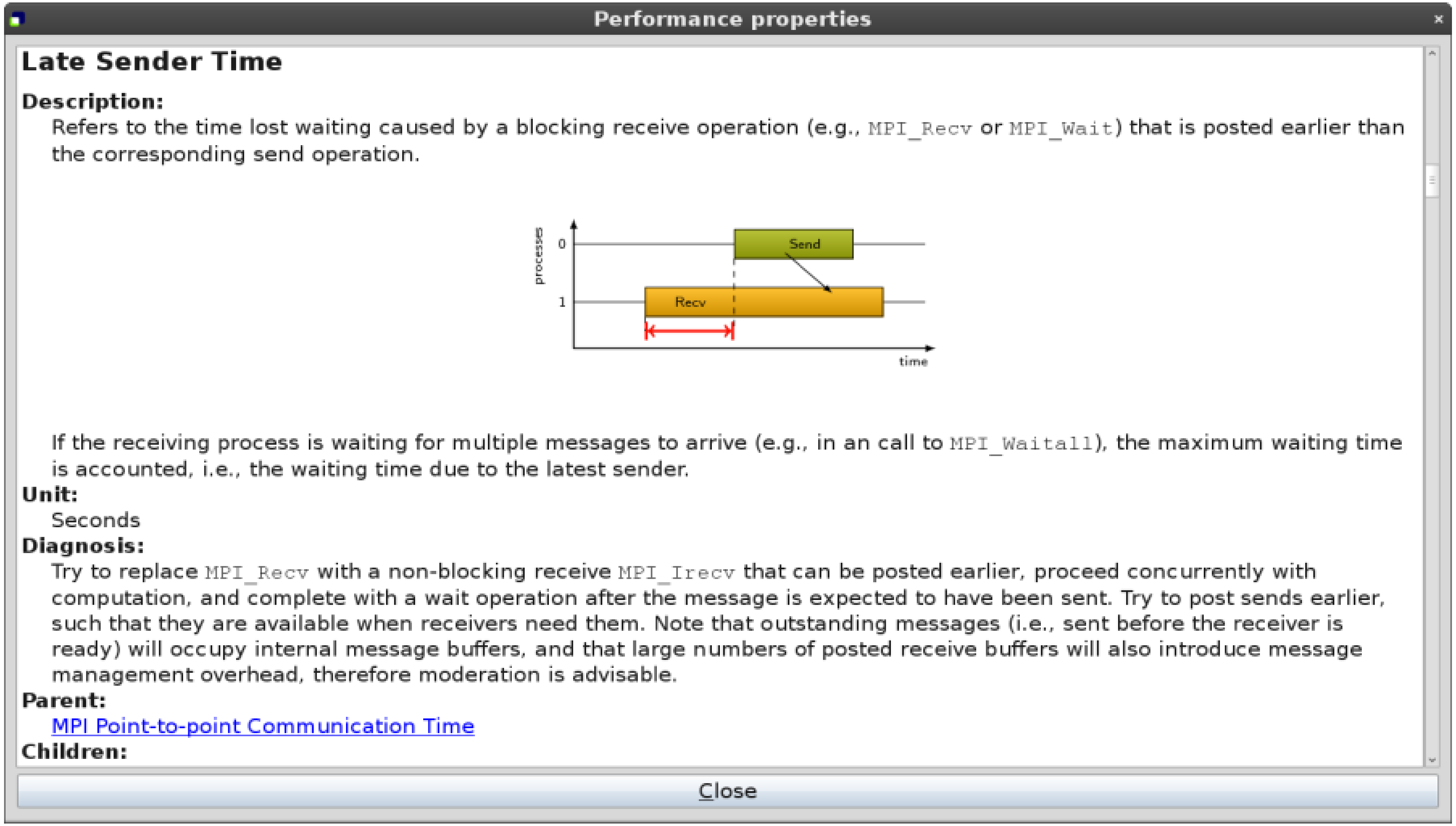
Access online metric description via context menu



cube 4.1.1 livedvd2: scorep_bt-mz_B_4x4_trace/trace.cubex

Shows the online description of the clicked item

Online metric description



Late Sender Time

Description:

Refers to the time lost waiting caused by a blocking receive operation (e.g., `MPI_Recv` or `MPI_Wait`) that is posted earlier than the corresponding send operation.

If the receiving process is waiting for multiple messages to arrive (e.g., in an call to `MPI_Waitall`), the maximum waiting time is accounted, i.e., the waiting time due to the latest sender.

Unit:

Seconds

Diagnosis:

Try to replace `MPI_Recv` with a non-blocking receive `MPI_Irecv` that can be posted earlier, proceed concurrently with computation, and complete with a wait operation after the message is expected to have been sent. Try to post sends earlier, such that they are available when receivers need them. Note that outstanding messages (i.e., sent before the receiver is ready) will occupy internal message buffers, and that large numbers of posted receive buffers will also introduce message management overhead, therefore moderation is advisable.

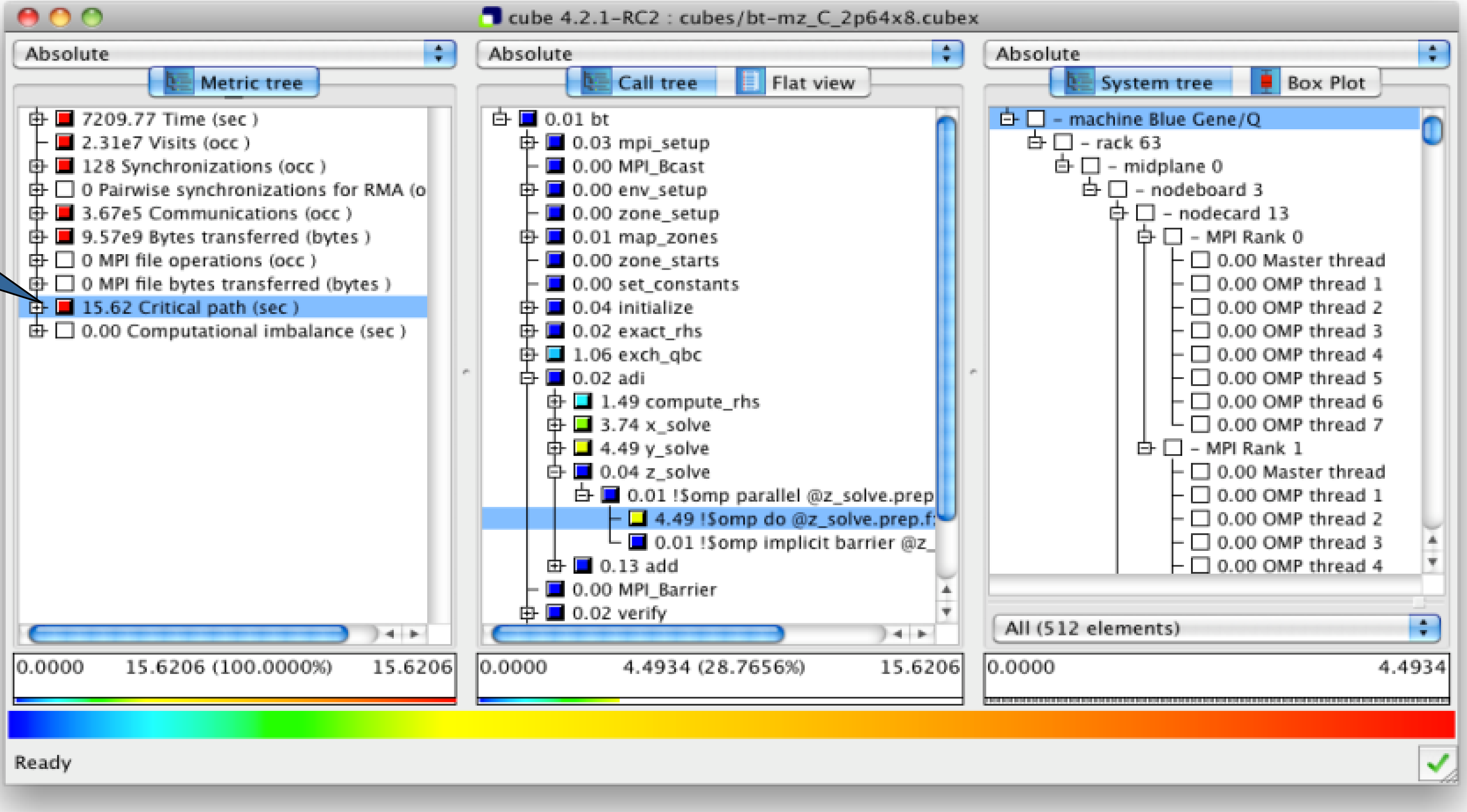
Parent:

MPI Point-to-point Communication Time

Children:

Critical-path analysis

Critical-path profile shows wall-clock time impact



cube 4.2.1-RC2 : cubes/bt-mz_C_2p64x8.cubex

Absolute | Metric tree

```
⊞ 7209.77 Time (sec )
  2.31e7 Visits (occ )
⊞ 128 Synchronizations (occ )
⊞ 0 Pairwise synchronizations for RMA (o
⊞ 3.67e5 Communications (occ )
⊞ 9.57e9 Bytes transferred (bytes )
⊞ 0 MPI file operations (occ )
⊞ 0 MPI file bytes transferred (bytes )
⊞ 15.62 Critical path (sec )
⊞ 0.00 Computational imbalance (sec )
```

0.0000 15.6206 (100.0000%) 15.6206

Absolute | Call tree | Flat view

```
⊟ 0.01 bt
  ⊞ 0.03 mpi_setup
    0.00 MPI_Bcast
  ⊞ 0.00 env_setup
    0.00 zone_setup
  ⊞ 0.01 map_zones
    0.00 zone_starts
    0.00 set_constants
  ⊞ 0.04 initialize
  ⊞ 0.02 exact_rhs
  ⊞ 1.06 exch_qbc
  ⊟ 0.02 adi
    ⊞ 1.49 compute_rhs
    ⊞ 3.74 x_solve
    ⊞ 4.49 y_solve
    ⊟ 0.04 z_solve
      ⊟ 0.01 !$omp parallel @z_solve.prep
          4.49 !$omp do @z_solve.prep.f
          0.01 !$omp implicit barrier @z_
    ⊞ 0.13 add
    0.00 MPI_Barrier
  ⊞ 0.02 verify
```

0.0000 4.4934 (28.7656%) 15.6206

Absolute | System tree | Box Plot

```
⊟ □ - machine Blue Gene/Q
  ⊟ □ - rack 63
    ⊟ □ - midplane 0
      ⊟ □ - nodeboard 3
        ⊟ □ - nodecard 13
          ⊟ □ - MPI Rank 0
              □ 0.00 Master thread
              □ 0.00 OMP thread 1
              □ 0.00 OMP thread 2
              □ 0.00 OMP thread 3
              □ 0.00 OMP thread 4
              □ 0.00 OMP thread 5
              □ 0.00 OMP thread 6
              □ 0.00 OMP thread 7
          ⊟ □ - MPI Rank 1
              □ 0.00 Master thread
              □ 0.00 OMP thread 1
              □ 0.00 OMP thread 2
              □ 0.00 OMP thread 3
              □ 0.00 OMP thread 4
```

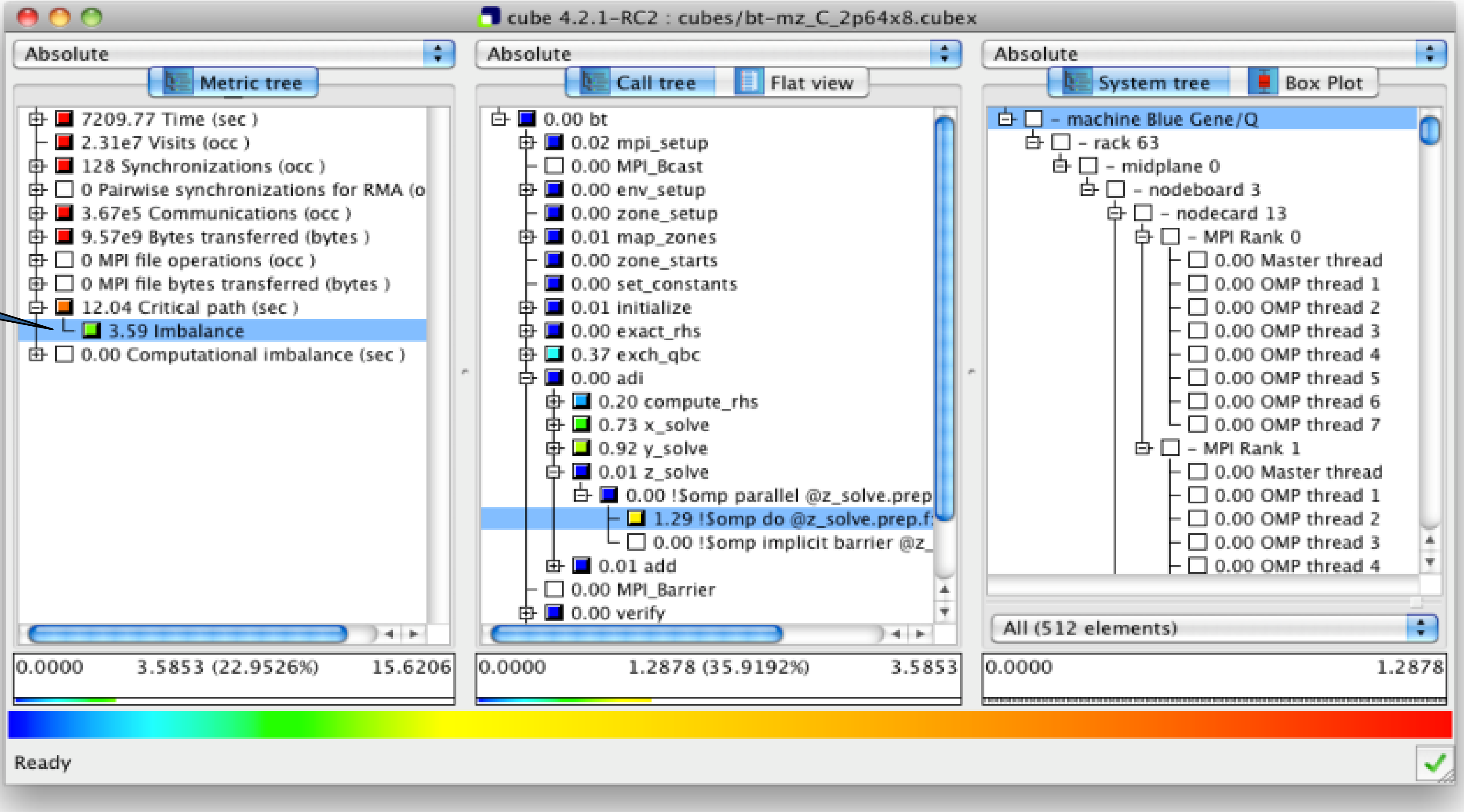
All (512 elements)

0.0000 4.4934

Ready

Critical-path analysis

Critical-path imbalance highlights inefficient parallelism



cube 4.2.1-RC2 : cubes/bt-mz_C_2p64x8.cubex

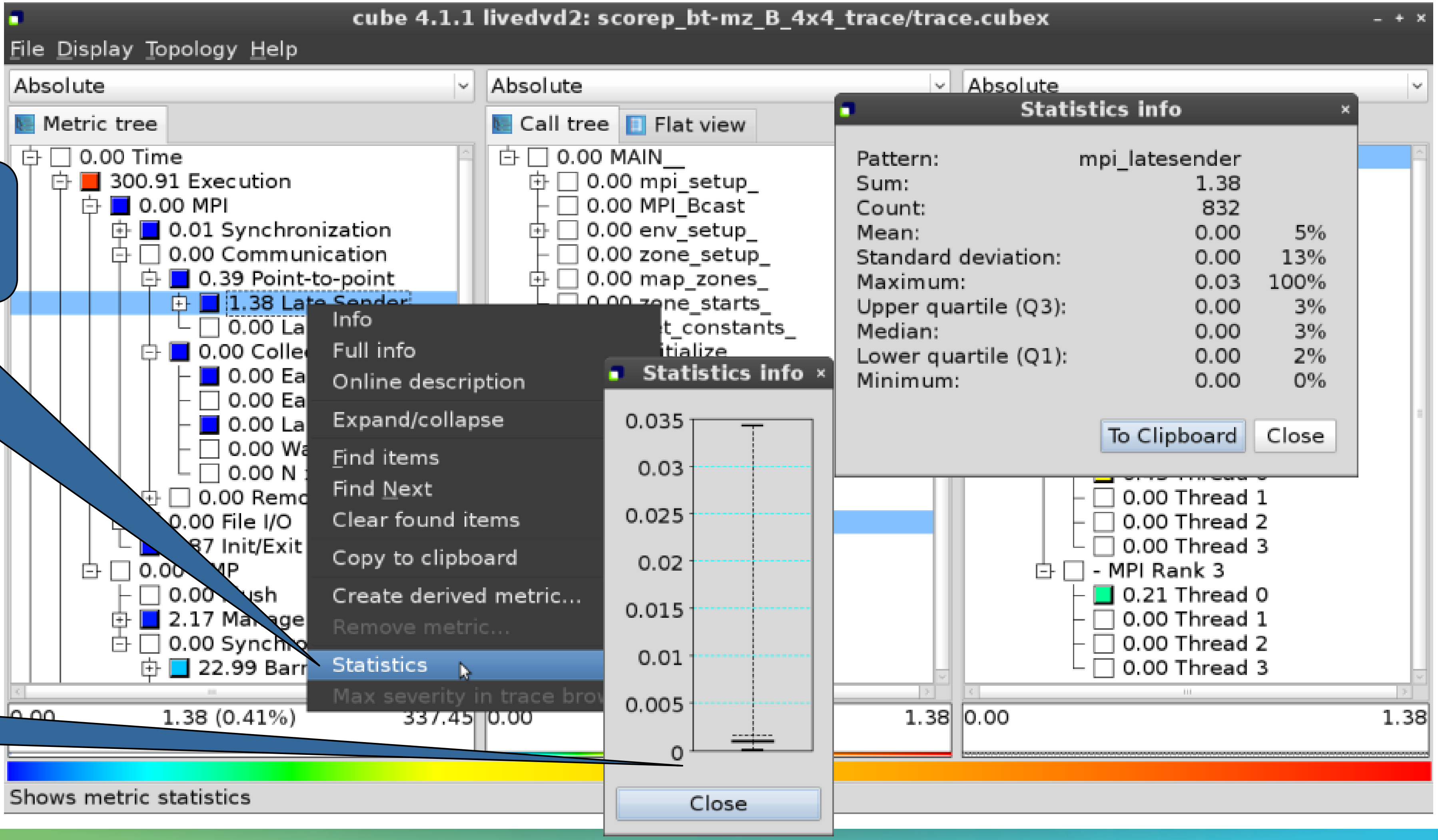
```
7209.77 Time (sec )
2.31e7 Visits (occ )
128 Synchronizations (occ )
0 Pairwise synchronizations for RMA (o
3.67e5 Communications (occ )
9.57e9 Bytes transferred (bytes )
0 MPI file operations (occ )
0 MPI file bytes transferred (bytes )
12.04 Critical path (sec )
  3.59 Imbalance
0.00 Computational imbalance (sec )
```

```
0.00 bt
  0.02 mpi_setup
  0.00 MPI_Bcast
  0.00 env_setup
  0.00 zone_setup
  0.01 map_zones
  0.00 zone_starts
  0.00 set_constants
  0.01 initialize
  0.00 exact_rhs
  0.37 exch_qbc
  0.00 adi
    0.20 compute_rhs
    0.73 x_solve
    0.92 y_solve
    0.01 z_solve
      0.00 !$omp parallel @z_solve.prep
        1.29 !$omp do @z_solve.prep.f
        0.00 !$omp implicit barrier @z_
    0.01 add
  0.00 MPI_Barrier
  0.00 verify
```

```
- machine Blue Gene/Q
  - rack 63
    - midplane 0
      - nodeboard 3
        - nodecard 13
          - MPI Rank 0
            0.00 Master thread
            0.00 OMP thread 1
            0.00 OMP thread 2
            0.00 OMP thread 3
            0.00 OMP thread 4
            0.00 OMP thread 5
            0.00 OMP thread 6
            0.00 OMP thread 7
          - MPI Rank 1
            0.00 Master thread
            0.00 OMP thread 1
            0.00 OMP thread 2
            0.00 OMP thread 3
            0.00 OMP thread 4
```

```
0.0000   3.5853 (22.9526%)   15.6206
0.0000   1.2878 (35.9192%)   3.5853
0.0000                       1.2878
```

Pattern instance statistics



Statistics info

Pattern:	mpi_latesender	
Sum:	1.38	
Count:	832	
Mean:	0.00	5%
Standard deviation:	0.00	13%
Maximum:	0.03	100%
Upper quartile (Q3):	0.00	3%
Median:	0.00	3%
Lower quartile (Q1):	0.00	2%
Minimum:	0.00	0%

To Clipboard   Close

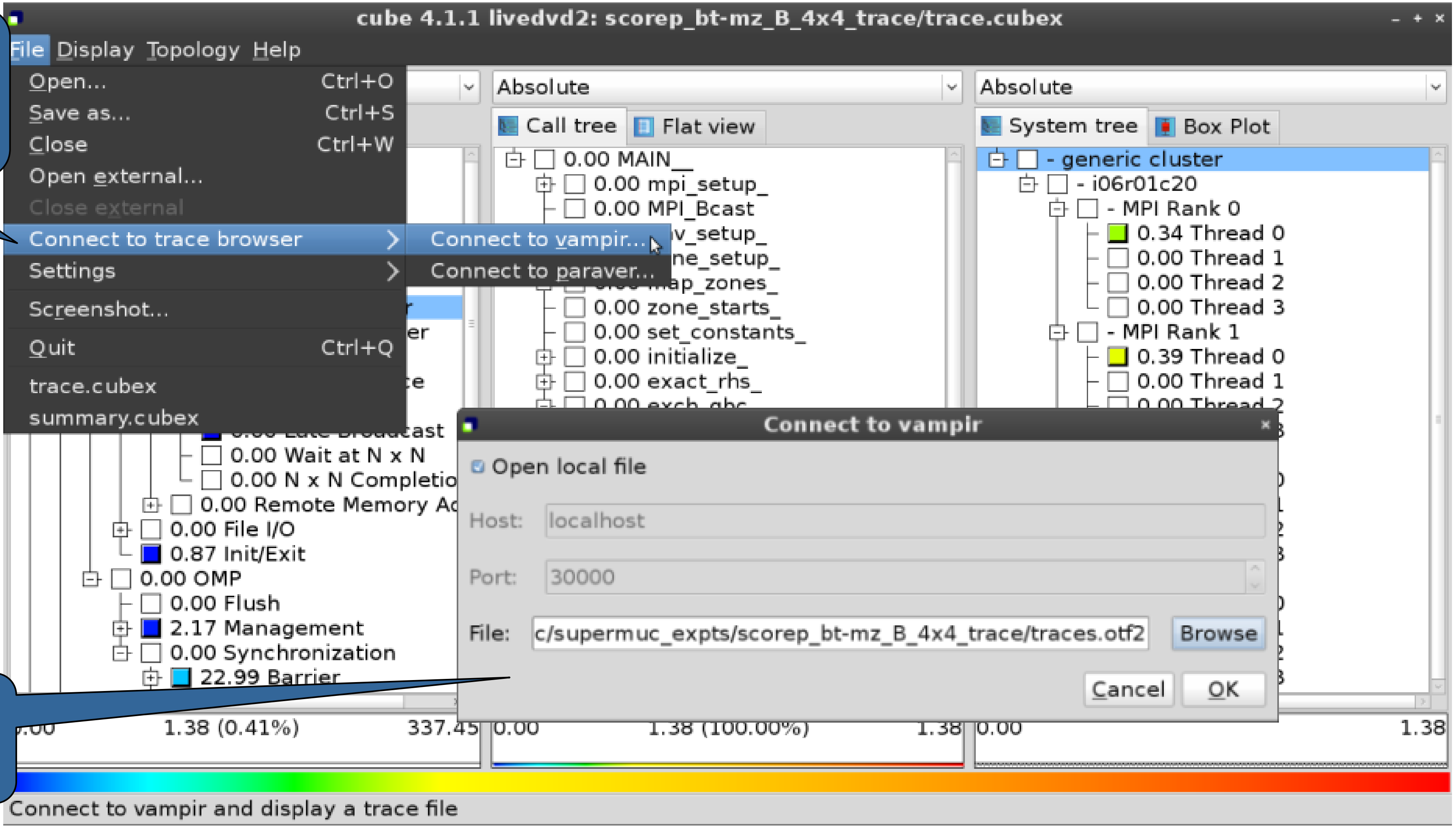
Access pattern instance statistics via context menu

Click to get statistics details

Connect to Vampir trace browser

To investigate most severe pattern instances, connect to a trace browser…

…and select trace file from the experiment directory



Connect to vampir

☑ Open local file

Host: localhost

Port: 30000

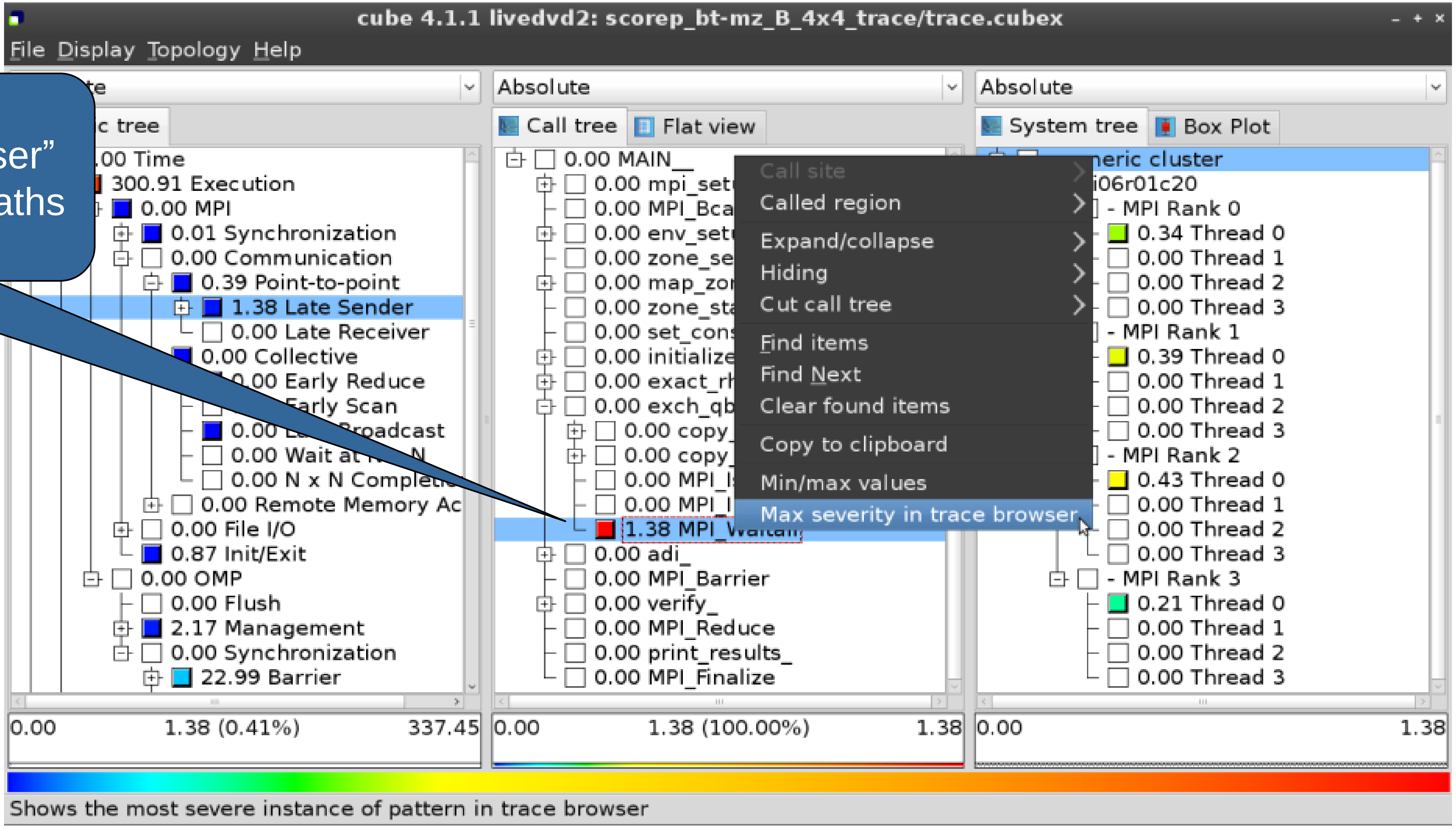
File: c/supermuc_expts/scorep_bt-mz_B_4x4_trace/traces.otf2

Browse

Cancel OK

Show most severe pattern instances

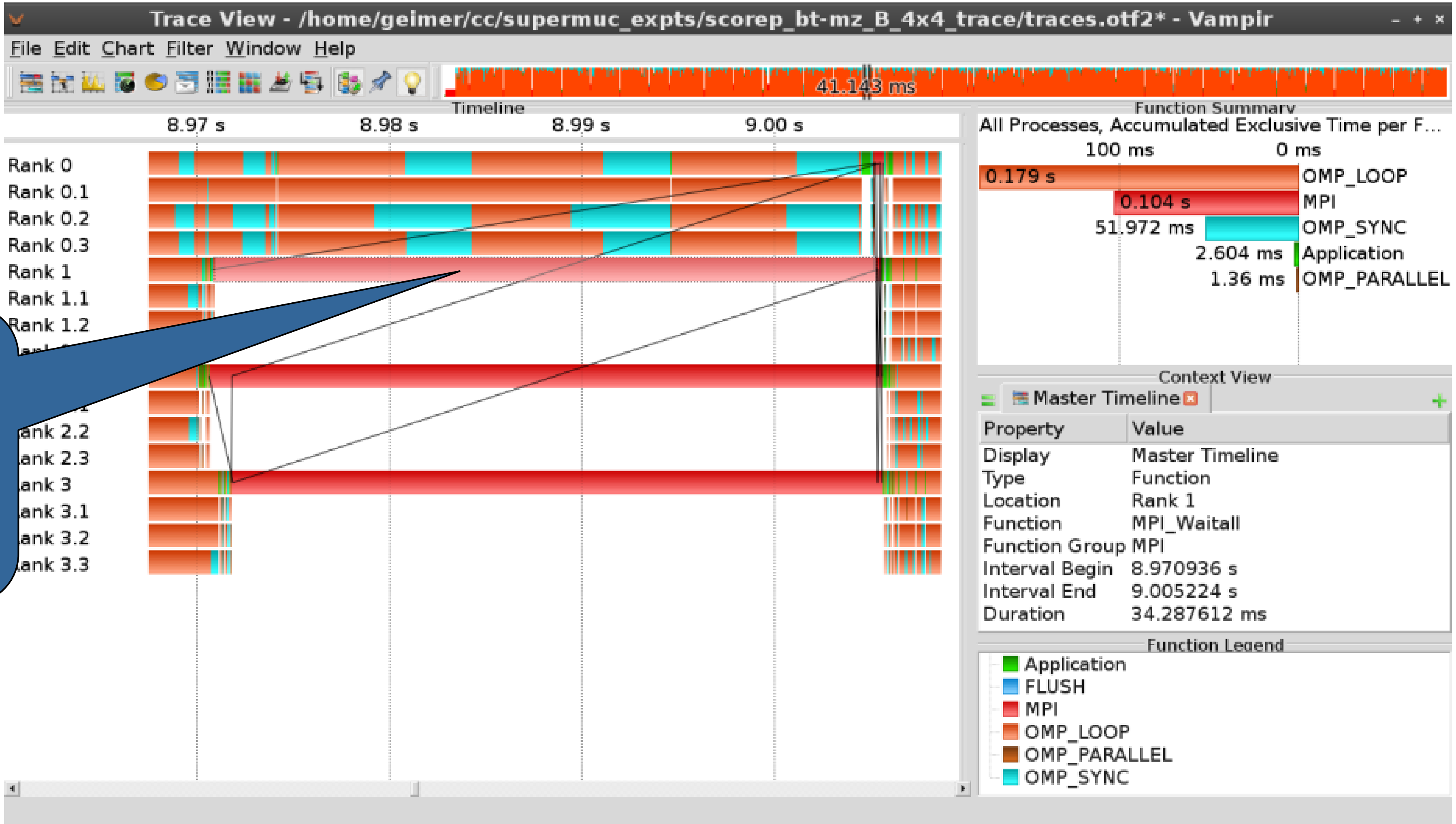
Select
“Max severity in trace browser”
from context menu of call paths
marked with a red frame



cube 4.1.1 livedvd2: scorep_bt-mz_B_4x4_trace/trace.cubex

Shows the most severe instance of pattern in trace browser

Investigate most severe instance in Vampir



Property	Value
Display	Master Timeline
Type	Function
Location	Rank 1
Function	MPI_Waitall
Function Group	MPI
Interval Begin	8.970936 s
Interval End	9.005224 s
Duration	34.287612 ms

Function Legend

- 🟩 Application
- 🟦 FLUSH
- 🟥 MPI
- 🟧 OMP_LOOP
- 🟫 OMP_PARALLEL
- 🟦 OMP_SYNC

Vampir will automatically zoom to the worst instance in multiple steps (i.e., undo zoom provides more context)

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

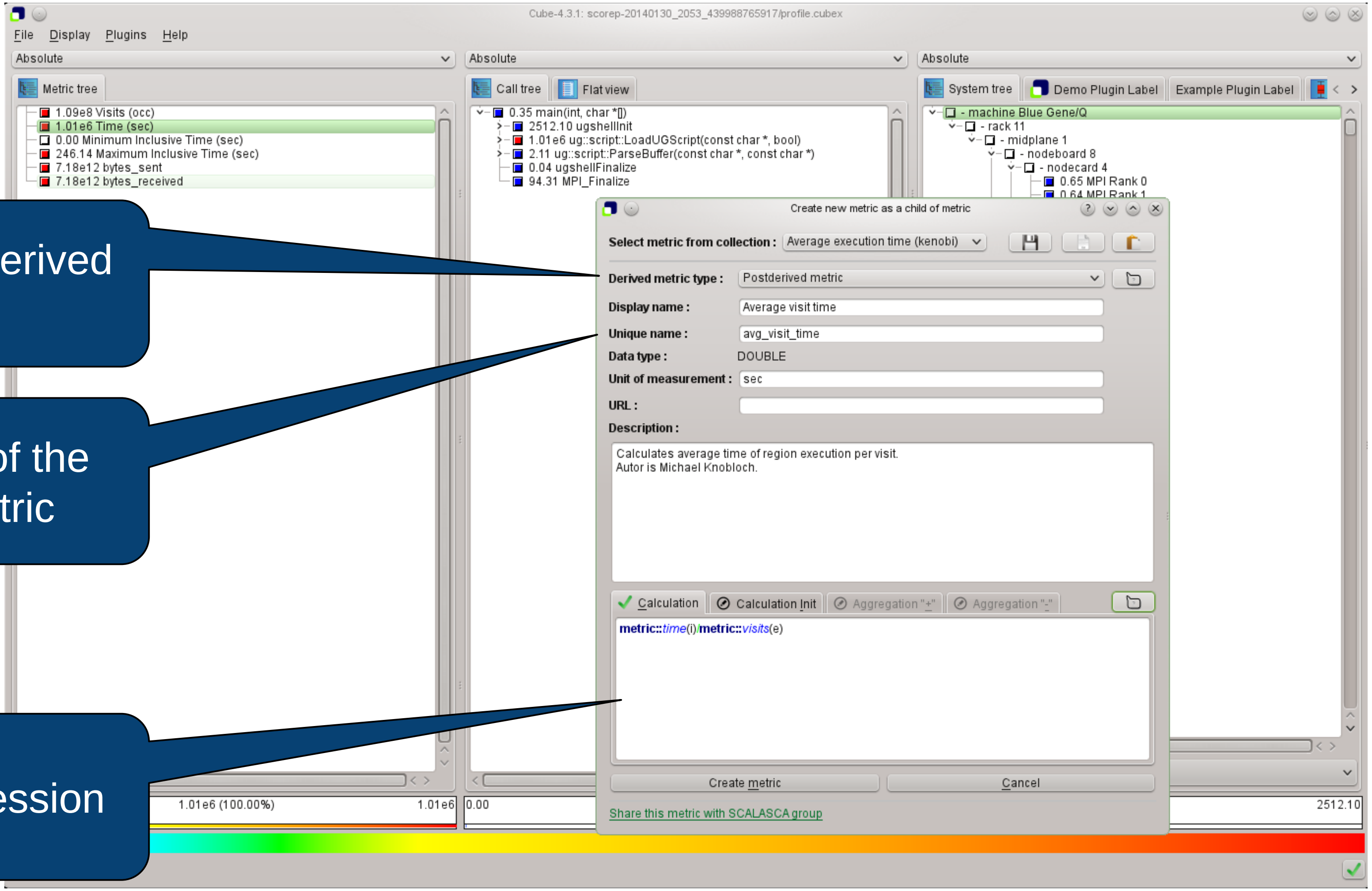
- Examples:
  - "Average execution time": Postderived metric with expression

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

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

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

Derived metrics in Cube GUI



Create new metric as a child of metric

Select metric from collection : Average execution time (kenobi)

Derived metric type : Postderived metric

Display name : Average visit time

Unique name : avg_visit_time

Data type : DOUBLE

Unit of measurement : sec

URL :

Description :

Calculates average time of region execution per visit.
Autor is Michael Knobloch.

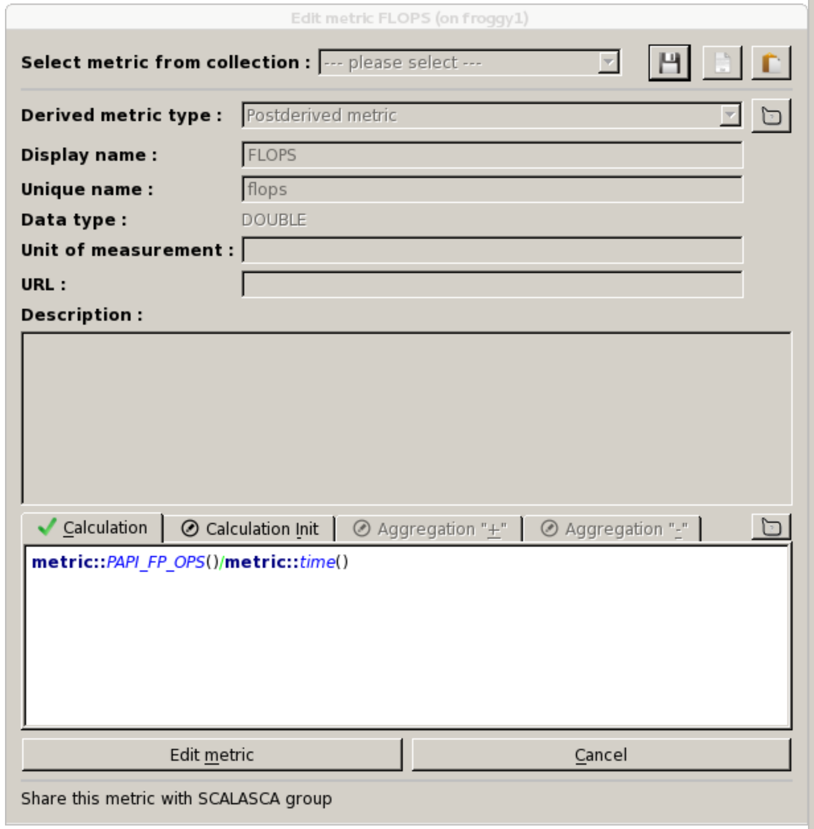
```
metric::time(i)/metric::visits(e)
```

Collection of derived metrics

Parameters of the derived metric

CubePL expression

Example: FLOPS based on PAPI_FP_OPS and time



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 :

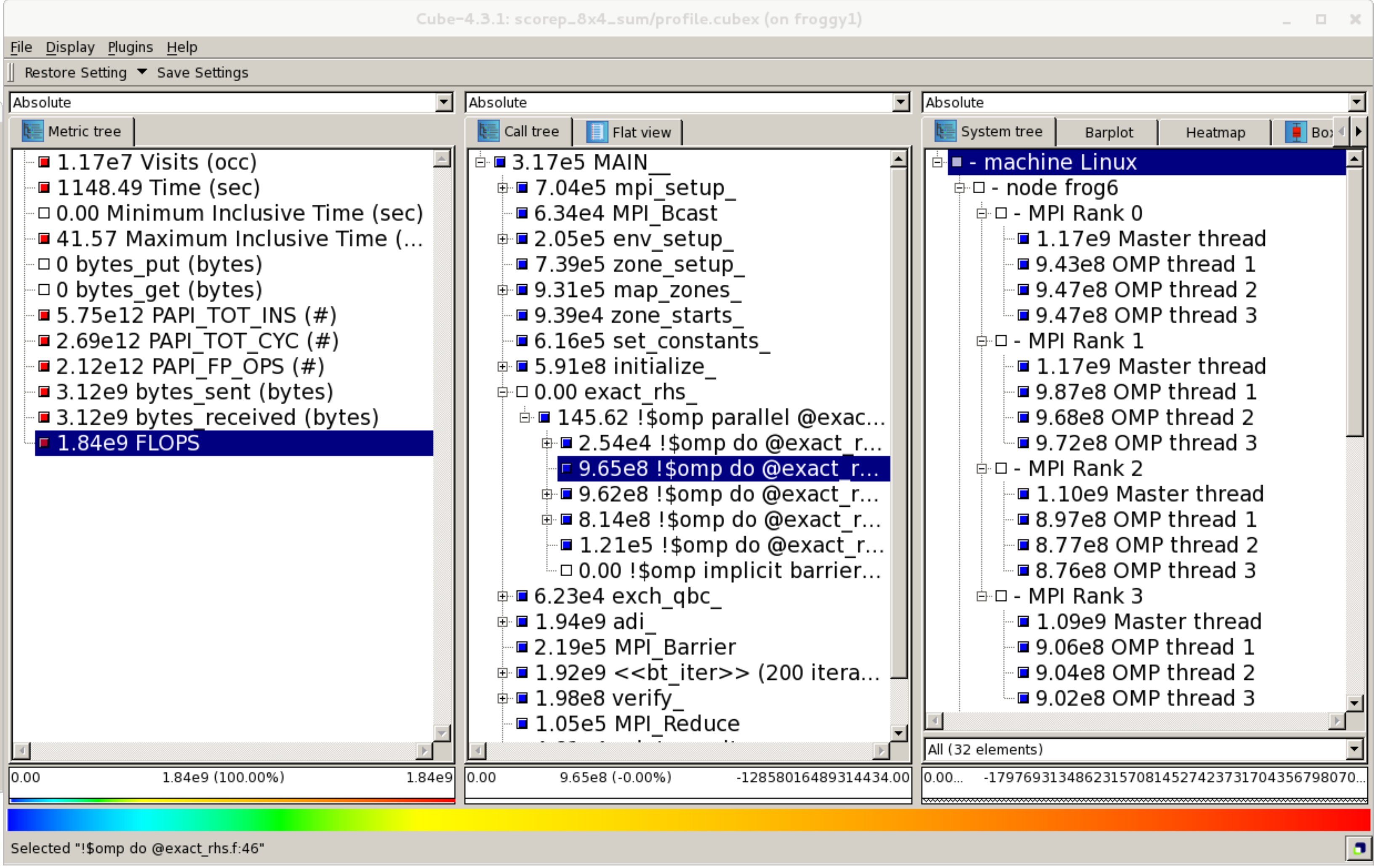
Description :

Calculation | Calculation Init | Aggregation "+" | Aggregation "-"

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

Edit metric | Cancel

Share this metric with SCALASCA group



Cube-4.3.1: scorep_8x4_sum/profile.cubex (on froggy1)

File Display Plugins Help
Restore Setting ▾ Save Settings
Absolute | Absolute | Absolute
Metric tree | Call tree | Flat view | System tree | Barplot | Heatmap | Box

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

Call tree:

```
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:

```
- 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_B_mic15p30x4_sum/profile.cubex
Writing cut.cubex... done.
```

- Calculating difference of two reports

```
% cube_diff  scorep_bt-mz_B_mic15p30x4_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

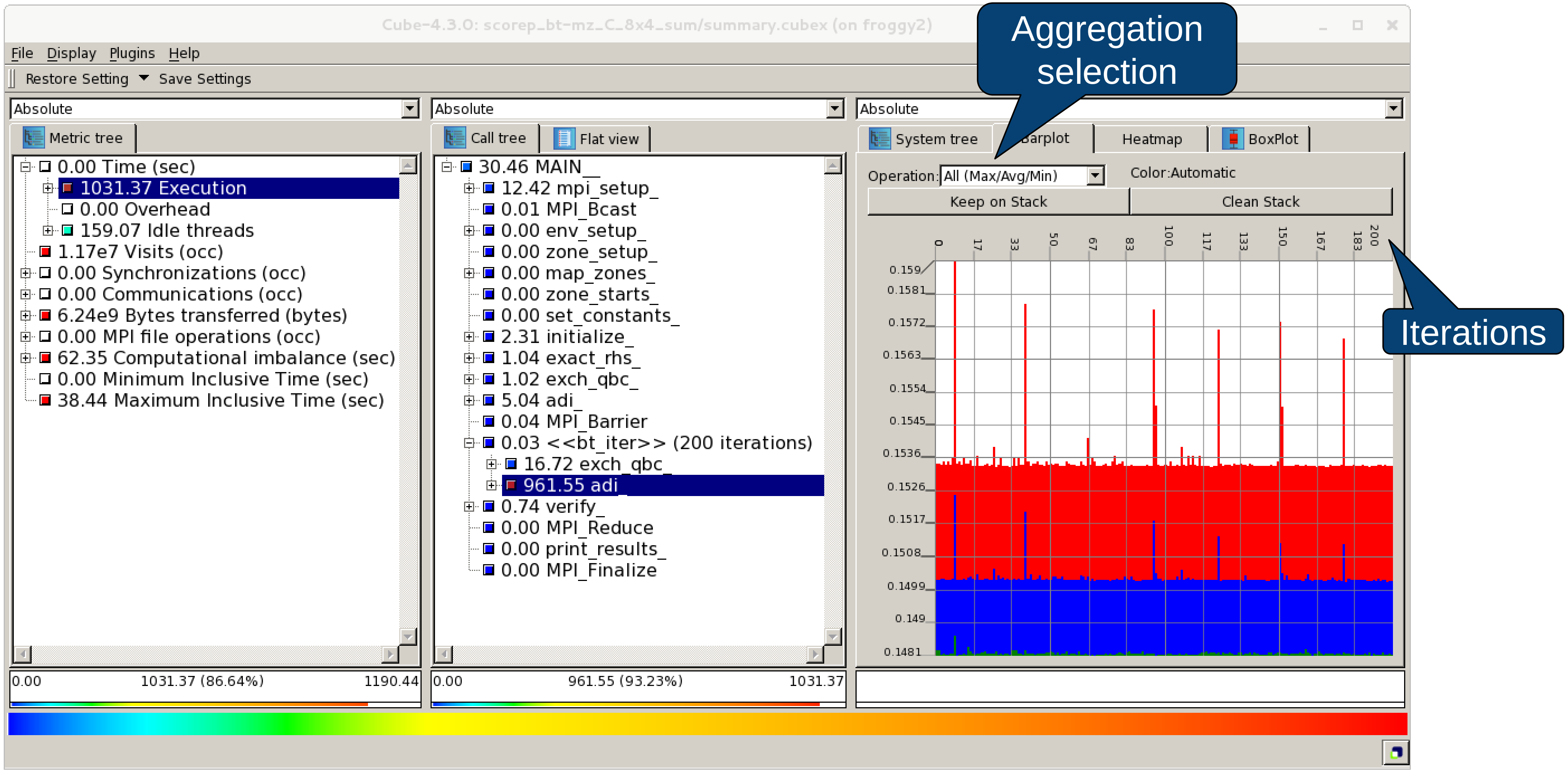
```
SCOREP_USER_REGION_DEFINE( scorep_bt_loop )
SCOREP_USER_REGION_BEGIN( scorep_bt_loop, "<<bt_iter>>", SCOREP_USER_REGION_TYPE_DYNAMIC )
SCOREP_USER_REGION_END( scorep_bt_loop )
```

- 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



Cube-4.3.0: scorep_bt-mz_C_8x4_sum/summary.cubex (on froggy2)

File Display Plugins Help

Restore Setting Save Settings

Absolute | Absolute | Absolute

Metric tree

- 0.00 Time (sec)
  - 1031.37 Execution
  - 0.00 Overhead
  - 159.07 Idle threads
- 1.17e7 Visits (occ)
- 0.00 Synchronizations (occ)
- 0.00 Communications (occ)
- 6.24e9 Bytes transferred (bytes)
- 0.00 MPI file operations (occ)
- 62.35 Computational imbalance (sec)
- 0.00 Minimum Inclusive Time (sec)
- 38.44 Maximum Inclusive Time (sec)

Call tree | Flat view

- 30.46 MAIN_
  - 12.42 mpi_setup_
  - 0.01 MPI_Bcast
  - 0.00 env_setup_
  - 0.00 zone_setup_
  - 0.00 map_zones_
  - 0.00 zone_starts_
  - 0.00 set_constants_
  - 2.31 initialize_
  - 1.04 exact_rhs_
  - 1.02 exch_qbc_
  - 5.04 adi_
  - 0.04 MPI_Barrier
  - 0.03 <<bt_iter>> (200 iterations)
    - 16.72 exch_qbc_
    - 961.55 adi_
  - 0.74 verify_
  - 0.00 MPI_Reduce
  - 0.00 print_results_
  - 0.00 MPI_Finalize

System tree | Barplot | Heatmap | BoxPlot

Operation: All (Max/Avg/Min) Color:Automatic

Keep on Stack | Clean Stack

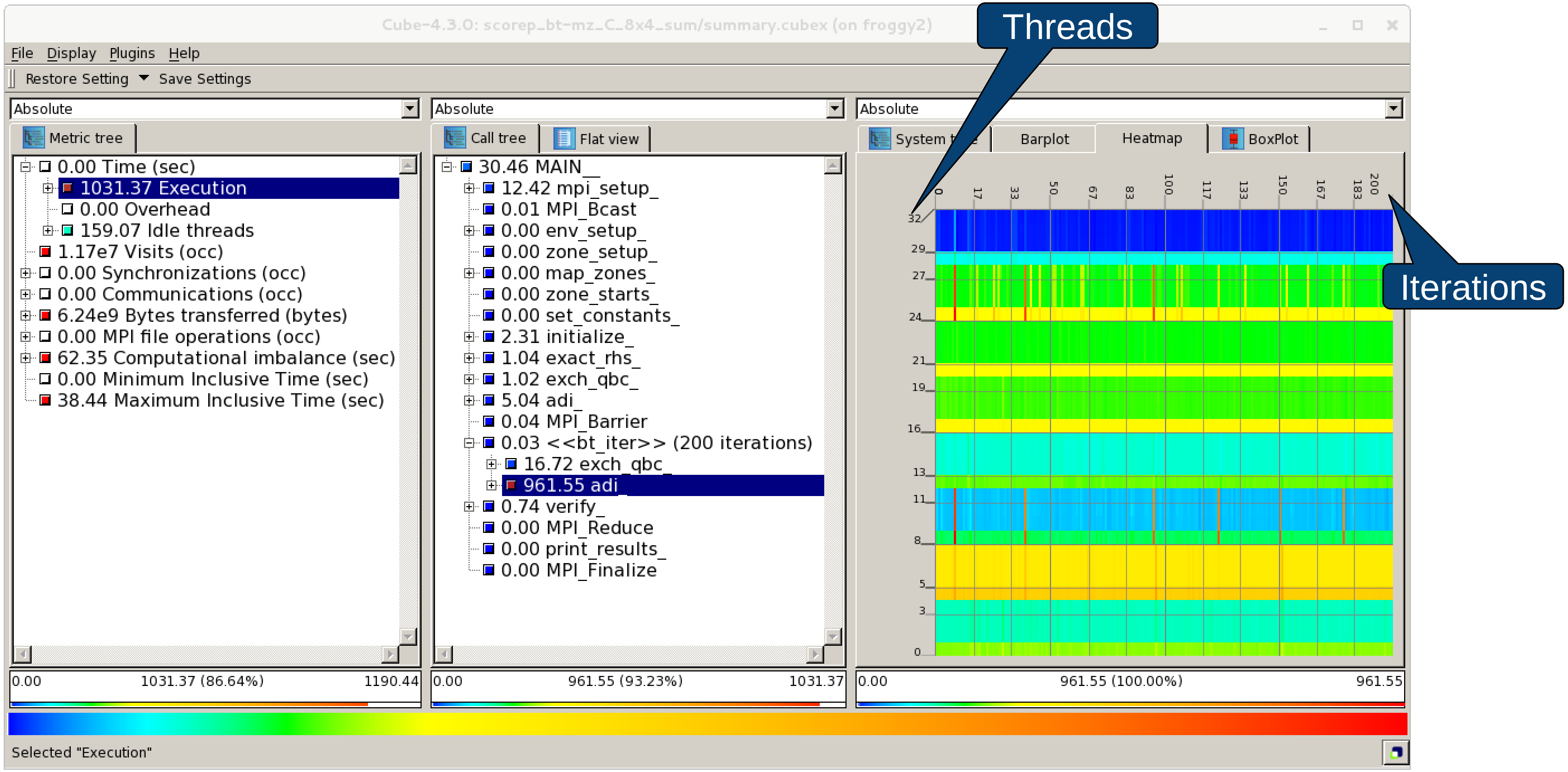
0.00 1031.37 (86.64%) 1190.44

0.00 961.55 (93.23%) 1031.37

Aggregation selection

Iterations

Iteration profiling: Heatmap



Cube-4.3.0: scorep_bt-mz_C_8x4_sum/summary.cubex (on froggy2)

File Display Plugins Help

Restore Setting ▾ Save Settings

Absolute | Absolute | Absolute

Metric tree | Call tree | Flat view | System tree | Barplot | Heatmap | BoxPlot

```
⊟ □ 0.00 Time (sec)
   ⊞ ■ 1031.37 Execution
     □ 0.00 Overhead
   ⊞ ■ 159.07 Idle threads
  ■ 1.17e7 Visits (occ)
⊞ □ 0.00 Synchronizations (occ)
⊞ □ 0.00 Communications (occ)
⊞ ■ 6.24e9 Bytes transferred (bytes)
⊞ □ 0.00 MPI file operations (occ)
⊞ ■ 62.35 Computational imbalance (sec)
  □ 0.00 Minimum Inclusive Time (sec)
  ■ 38.44 Maximum Inclusive Time (sec)
```

```
⊟ ■ 30.46 MAIN__
   ⊞ ■ 12.42 mpi_setup_
     ■ 0.01 MPI_Bcast
   ⊞ ■ 0.00 env_setup_
     ■ 0.00 zone_setup_
   ⊞ ■ 0.00 map_zones_
     ■ 0.00 zone_starts_
     ■ 0.00 set_constants_
   ⊞ ■ 2.31 initialize_
   ⊞ ■ 1.04 exact_rhs_
   ⊞ ■ 1.02 exch_qbc_
   ⊞ ■ 5.04 adi_
     ■ 0.04 MPI_Barrier
   ⊟ ■ 0.03 <<bt_iter>> (200 iterations)
      ⊞ ■ 16.72 exch_qbc_
      ⊞ ■ 961.55 adi_
   ⊞ ■ 0.74 verify_
     ■ 0.00 MPI_Reduce
     ■ 0.00 print_results_
     ■ 0.00 MPI_Finalize
```

```
0.00   1031.37 (86.64%)   1190.44
0.00   961.55 (93.23%)   1031.37
0.00   961.55 (100.00%)   961.55
```

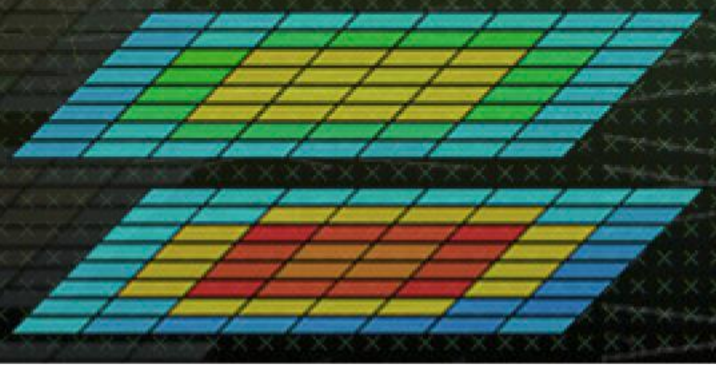
Selected "Execution"

Threads

Iterations

Cube: Further information

- Parallel program analysis report exploration tools
  - Libraries for XML 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:
  - \`cube-config --cube-dir\`/share/doc/CubeGuide.pdf
- Contact:
  - mailto: scalasca@fz-juelich.de



International HPC Summer School 2016: Performance analysis and optimization Tools overview

VI-HPS Team

Christian Feld – Jülich Supercomputing Centre

Virtual Institute – High Productivity Supercomputing

- **Goal**: Improve the quality and accelerate the development process of complex simulation codes running on highly-parallel computer systems
- Start-up funding (2006–2011) by Helmholtz Association of German Research Centres
- Activities
  - Development and integration of HPC programming tools
    - Correctness checking & performance analysis
  - Academic workshops
  - Training workshops
  - Service
    - Support email lists
    - Application engagement

HELMHOLTZ | ASSOCIATION

http://www.vi-hps.org

VI-HPS partners (founders)

Forschungszentrum Jülich

- Jülich Supercomputing Centre

RWTH Aachen University

- Centre for Computing & Communication

Technische Universität Dresden

- Centre for Information Services & HPC

University of Tennessee (Knoxville)

- Innovative Computing Laboratory

VI-HPS partners (cont.)

Barcelona Supercomputing Center

- Centro Nacional de Supercomputación

Lawrence Livermore National Lab.

- Center for Applied Scientific Computing

Technical University of Darmstadt

- Laboratory for Parallel Programming

Technical University of Munich

- Chair for Computer Architecture

University of Oregon

- Performance Research Laboratory

University of Stuttgart

- HPC Centre

University of Versailles St-Quentin

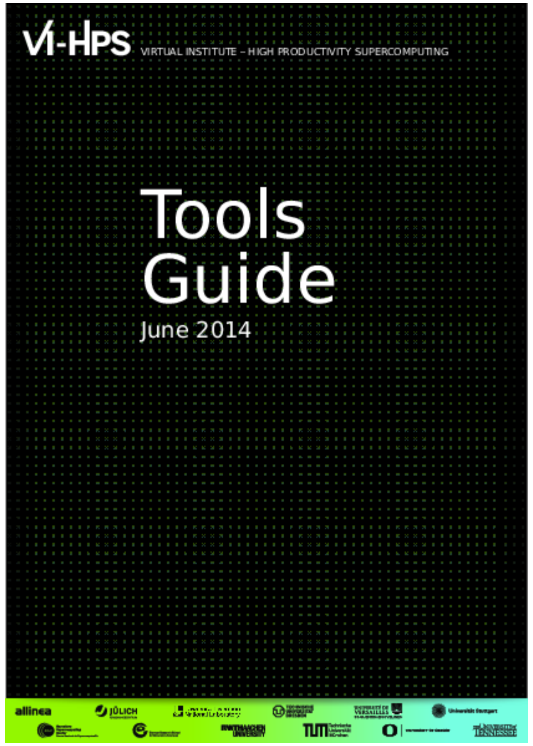
- LRC ITACA

Allinea Software Ltd

Productivity tools

- MUST
  - MPI usage correctness checking
- PAPI
  - Interfacing to hardware performance counters
- Periscope Tuning Framework
  - Automatic analysis and Tuning
- **Scalasca**
  - Large-scale parallel performance analysis
- **TAU**
  - Integrated parallel performance system
- Vampir
  - Interactive graphical trace visualization & analysis
- **Score-P**
  - Community-developed instrumentation & measurement infrastructure

For a brief overview of tools consult the VI-HPS Tools Guide:



VI-HPS VIRTUAL INSTITUTE – HIGH PRODUCTIVITY SUPERCOMPUTING

Tools Guide

June 2014

Productivity tools (cont.)

- DDT/MAP/PR: Parallel debugging, profiling & performance reports
- Extra-P: Automated performance modelling
- Kcachegrind: Callgraph-based cache analysis [x86 only]
- MAQAO: Assembly instrumentation & optimization [x86-64 only]
- mpiP/mpiPview: MPI profiling tool and analysis viewer
- Open MPI: Integrated memory checking
- Open|SpeedShop: Integrated parallel performance analysis environment
- Paraver/Dimemas/Extrae: Event tracing and graphical trace visualization & analysis
- Rubik: Process mapping generation & optimization [BG only]
- SIONlib/Spindle: Optimized native parallel file I/O & shared library loading
- STAT: Stack trace analysis tools
- SysMon: Batch system monitor plugin for Eclipse PTP

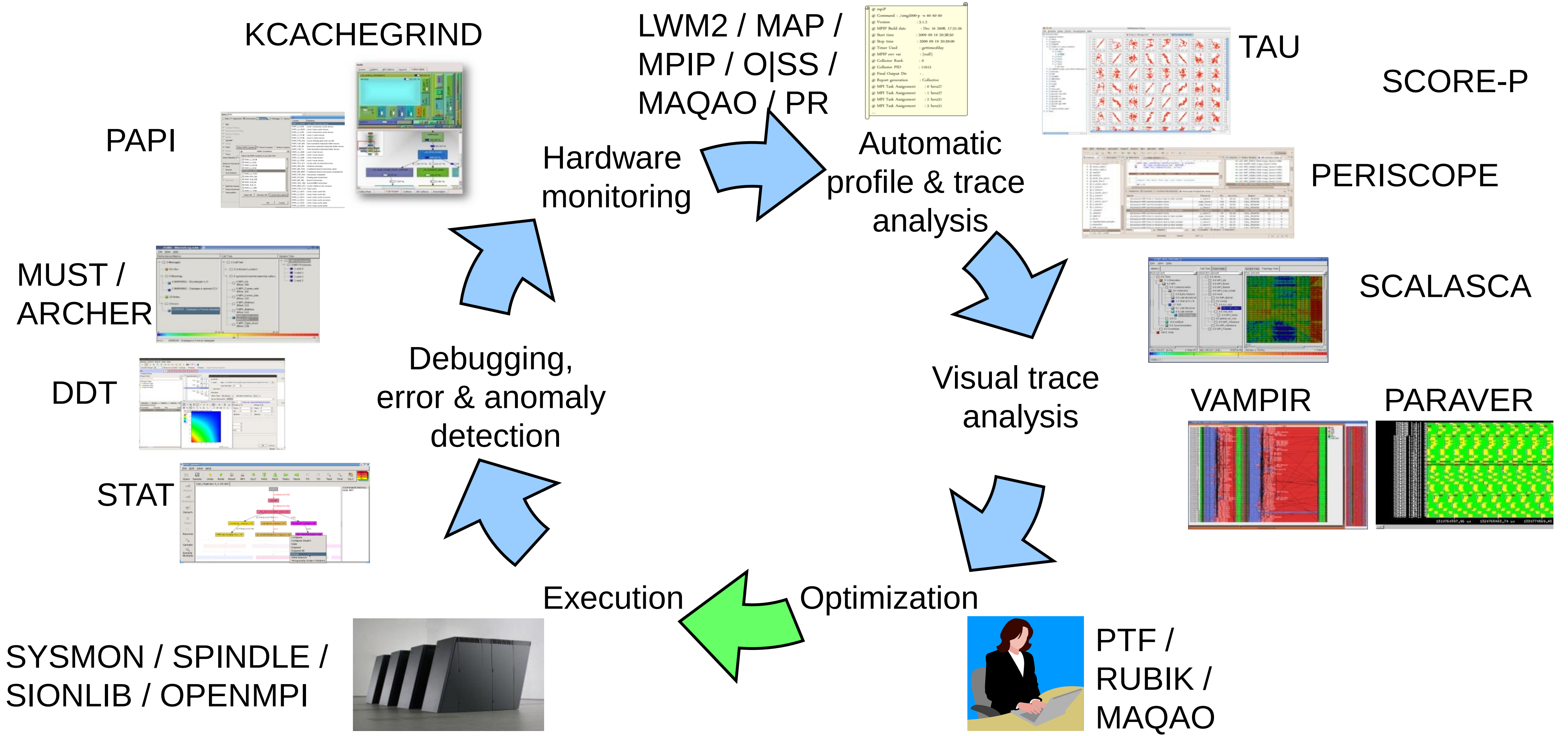
Non VI-HPS performance tools

- HPC Toolkit (Rice University): http://hpctoolkit.org/
- PerfExpert (TACC): https://www.tacc.utexas.edu/research-development/tacc-projects/perfexpert
- Likwid (University of Erlangen-Nuremberg): https://github.com/RRZE-HPC/likwid/wiki
- …

Commercial tools:

- CrayPat (Cray)
- Intel VTune Amplifier XE: https://software.intel.com/en-us/intel-vtune-amplifier-xe
- …

Technologies and their integration



Hardware monitoring

Automatic profile & trace analysis

Visual trace analysis

Optimization

Execution

Debugging, error & anomaly detection

KCACHEGRIND

PAPI

LWM2 / MAP / MPIP / O|SS / MAQAO / PR

TAU

SCORE-P

PERISCOPE

SCALASCA

VAMPIR

PARAVER

PTF / RUBIK / MAQAO

SYSMON / SPINDLE / SIONLIB / OPENMPI

STAT

DDT

MUST / ARCHER

Workshops/Tutorials

- Tuning Workshop Series
  - Three to five days *bring-your-own-code* workshops at HPC centres
  - Usually free of charge
  - http://www.vi-hps.org/training/tws/

- Tutorials at various conferences
  - E.g., SC16: Hands-on Practical Hybrid Parallel Application Performance Engineering

Performance Audits/Plans/Proof-of-concepts

- Performance Optimisation and Productivity (POP)
  - Offers performance optimisation and productivity services
  - Time-limited offer/project
  - Using VI-HPS tools
  - Funded by European Unions Horizon 2020 research and innovation programme
  - https://pop-coe.eu/services

- They help you fix your code, for free!!!