
scalasca

# Tutorial Exercise NPB-MZ-MPI/BT

Brian Wylie & Markus Geimer
Jülich Supercomputing Centre
scalasca@fz-juelich.de

August 2012

# Performance analysis steps


0. Reference preparation for validation
1. Program instrumentation: `skin`
2. Summary measurement collection & analysis: `scan [-s]`
3. Summary analysis report examination: `square`
4. Summary experiment scoring: `square -s`
5. Event trace collection & analysis: `scan -t`
6. Event trace analysis report examination: `square`

- Configuration & customization
  - Instrumentation, Measurement, Analysis, Presentation

- Intermediate-level tutorial example
- Available in MPI, OpenMP & **hybrid OpenMP/MPI** variants
  - also MPI File I/O variants (collective & individual)
- Summary measurement collection & analysis
  - Automatic instrumentation
    - OpenMP, MPI & application functions
  - Summary analysis report examination
  - PAPI hardware counter metrics
- Trace measurement collection & analysis
  - Filter determination, specification & configuration
  - Automatic trace analysis report patterns
- Manual and PDT instrumentation
- Measurement configuration
- Analysis report algebra

# Load the Scalasca module


- Load the module

```
% module load UNITE
UNITE loaded
% module load scalasca
scalasca/1.4.2 loaded
```

- ... and run `scalasca` for brief usage information

```
% scalasca
Scalasca 1.4.2
Toolset for scalable performance analysis of large-scale applications
usage: scalasca [-v][-n] {action}
    1. prepare application objects and executable for measurement:
       scalasca -instrument <compile-or-link-command>       # skin
    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

   -v: enable verbose commentary
   -n: show actions without taking them
   -h: show quick reference guide (only)
```

# NPB instrumentation

- Prefix compile/link commands in Makefile definitions (config/make.def) with the Scalasca instrumenter

```
MPIF77 = scalasca -instrument mpif77
FLINK = $(MPIF77)
FFLAGS = -O -fopenmp

bt-mz: $(OBJECTS)
        $(FLINK) $(FFLAGS) -o bt-mz $(OBJECTS)
.f.o:
        $(MPIF77) $(FFLAGS) -c $<
```

- or use PREP macro as customizable preparation preposition

```
MPIF77 = $(PREP) mpif77
```

  - By default, PREP macro is not set and no instrumentation is performed for a regular “production” build
  - Specifying a PREP value in the Makefile or on the make command line uses it as a preposition, e.g., for instrumentation
    - % make PREP=“scalasca -instrument” ...

```
scalasca -instrument  mpif77 -O -fopenmp -c bt.f
```

# NPB-MZ-MPI/BT instrumented build

- Return to root directory and clean-up

```
% make clean
```

- Re-build specifying Scalasca instrumenter as PREP

```
% make bt-mz CLASS=B NPROCS=4 PREP="scalasca -instrument"
cd BT-MZ; make CLASS=B NPROCS=4 VERSION=
gmake: Entering directory 'BT-MZ'
cd ../sys; cc  -o setparams setparams.c
../sys/setparams bt-mz 4 B
scalasca -instrument mpif77 -c  -O -fopenmp  bt.f
...
scalasca -instrument mpif77 -c  -O -fopenmp  setup_mpi.f
cd ../common;  scalasca -instrument mpif77 -c -O -fopenmp timers.f
scalasca -instrument mpif77 -O -fopenmp -o ../bin.scalasca/bt-mz_B.4 \
bt.o make_set.o initialize.o exact_solution.o exact_rhs.o \
set_constants.o adi.o define.o copy_faces.o rhs.o solve_subs.o \
x_solve.o y_solve.o z_solve.o add.o error.o verify.o setup_mpi.o \
../common/print_results.o ../common/timers.o
INFO: Instrumented executable for OMP+MPI measurement
gmake: Leaving directory 'BT-MZ'
```

# BT-MZ summary measurement

- Run the application using the Scalasca measurement collection & analysis nexus prefixed to launch command

```
% cd bin.scalasca
% OMP_NUM_THREADS=4  scalasca -analyze  mpiexec -np 4 ./bt-mz_B.4
S=C=A=N: Scalasca 1.3 runtime summarization
S=C=A=N: ./epik_bt-mz_B_4x4_sum experiment archive
S=C=A=N: Sun Mar 29 16:36:31 2009: Collect start
mpiexec -np 4 ./bt-mz_B.4
[00000]EPIK: Created new measurement archive ./epik_bt-mz_B_4x4_sum
[00000]EPIK: Activated ./epik_bt-mz_B_4x4_sum [NO TRACE] (0.006s)

        [... Application output ...]

[00000]EPIK: Closing experiment ./epik_bt-mz_B_4x4_sum
[00000]EPIK: 164 unique paths (178 max paths, 7 max frames, 0 unknown)
[00000]EPIK: Unifying... done (0.023s)
[00000]EPIK: Collating... done (0.049s)
[00000]EPIK: Closed experiment ./epik_bt-mz_B_4x4_sum (0.073s)
S=C=A=N: Sun Mar 29 16:36:45 2009: Collect done (status=0) 57s
S=C=A=N: ./epik_bt-mz_B_4x4_sum complete.
```

- Produces experiment directory ./epik_bt-mz_B_4x4_sum

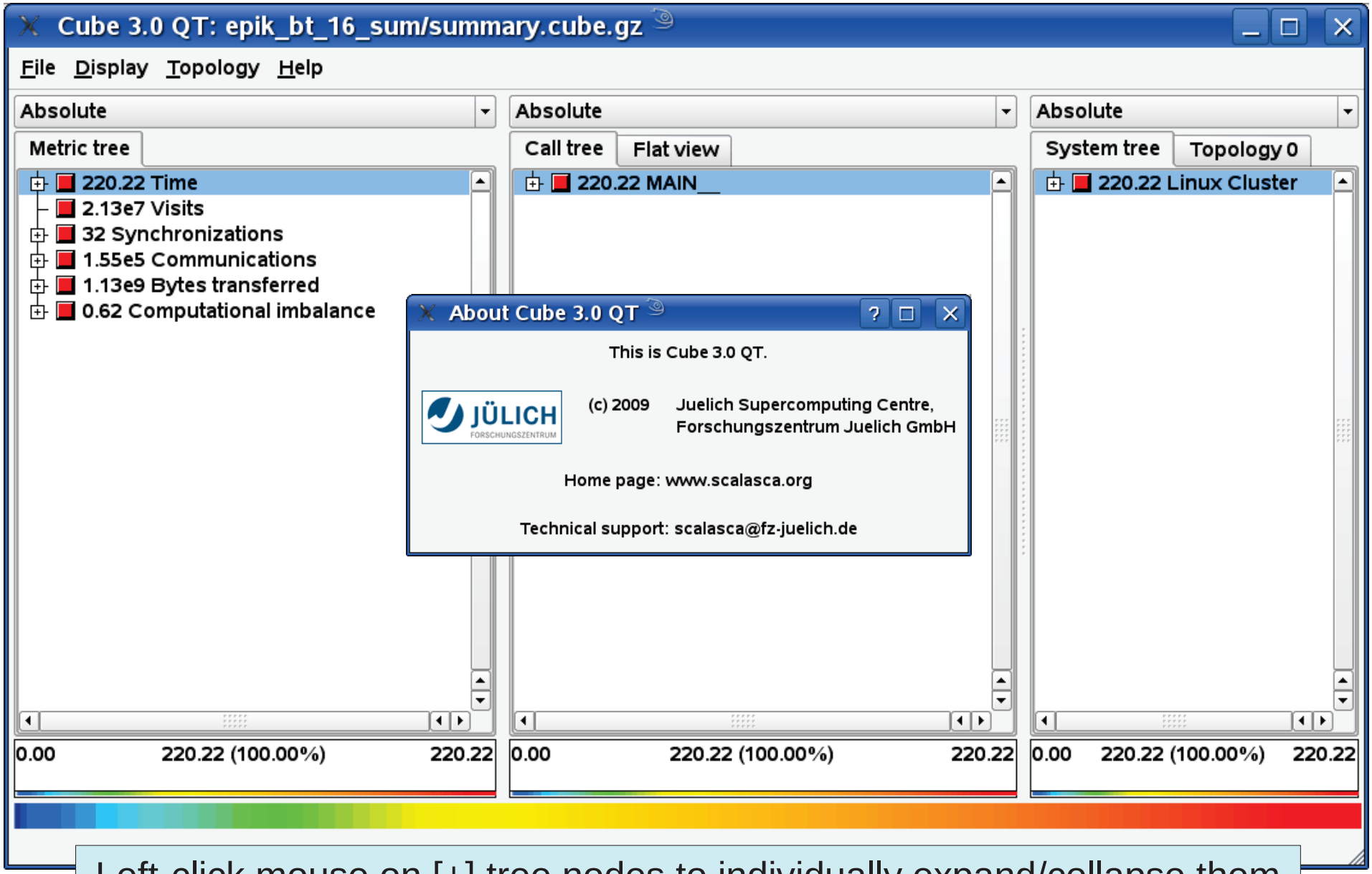
# BT-MZ summary analysis report examination

- Interactive exploration with Scalasca GUI

```
% scalasca -examine  epik_bt-mz_B_4x4_sum
INFO: Post-processing runtime summarization result...
INFO: Displaying ./epik_bt-mz_B_4x4_sum/summary.cube...

                  [GUI showing summary analysis report]
```

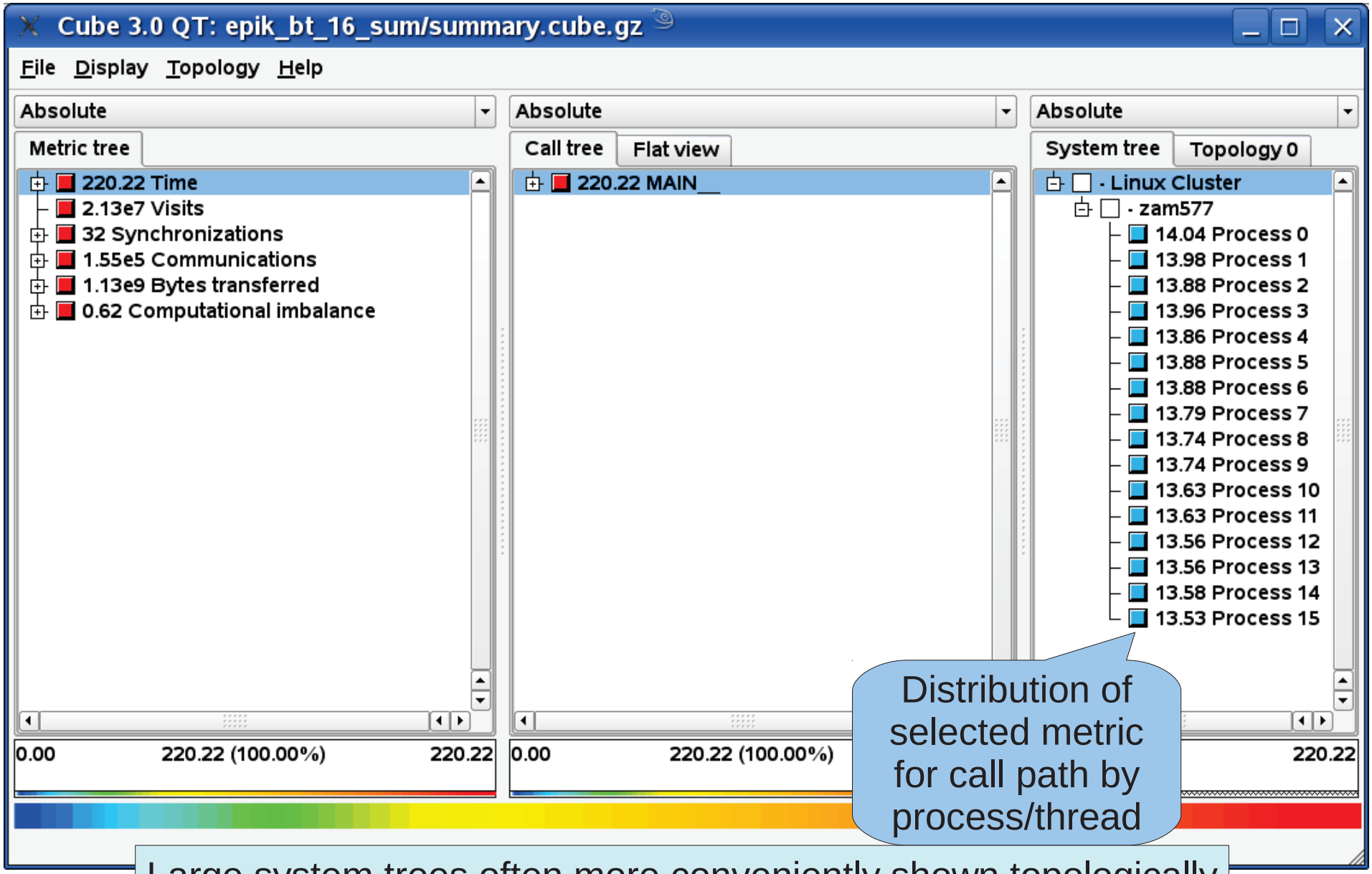
- The measurement archive directory ultimately contains
  - a copy of the execution output (epik.log)
  - a record of the measurement configuration (epik.conf)
  - the basic analysis report that was collated after measurement (epitome.cube)
  - the complete analysis report produced during post-processing (summary.cube.gz)

# Analysis report exploration (opening view)



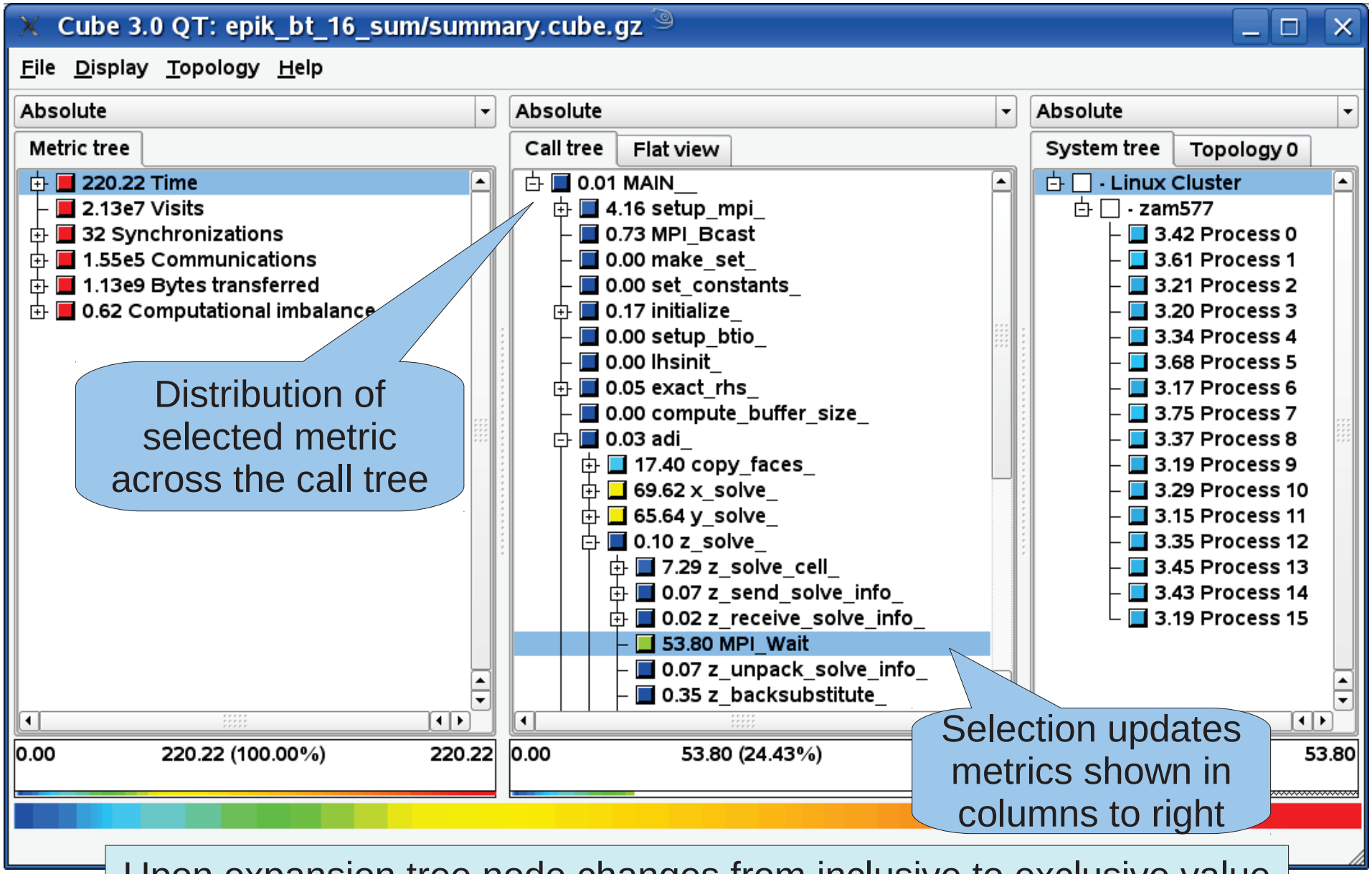

Left-click mouse on [+] tree nodes to individually expand/collapse them

# Analysis report exploration (system tree)

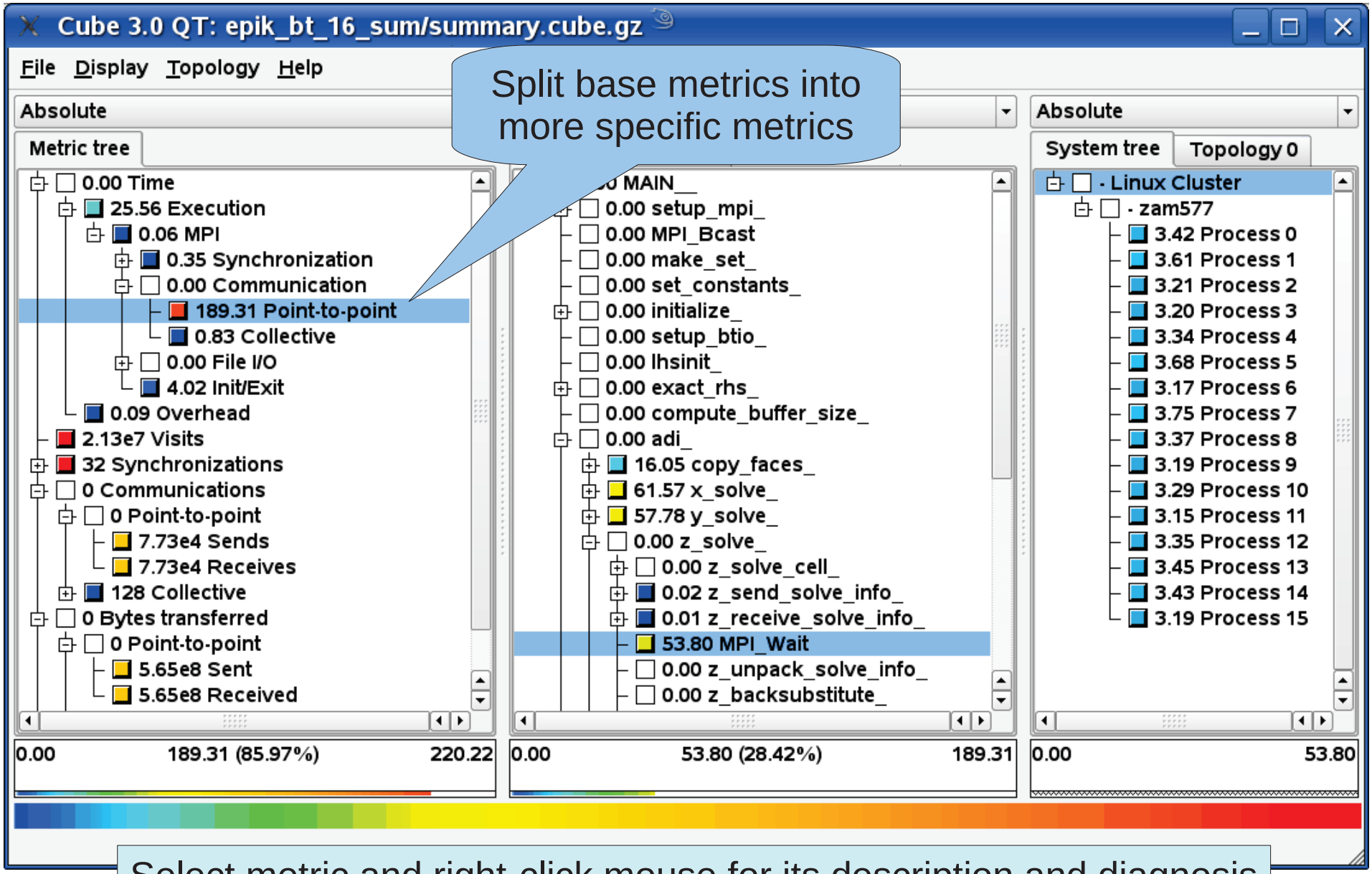



Large system trees often more conveniently shown topologically

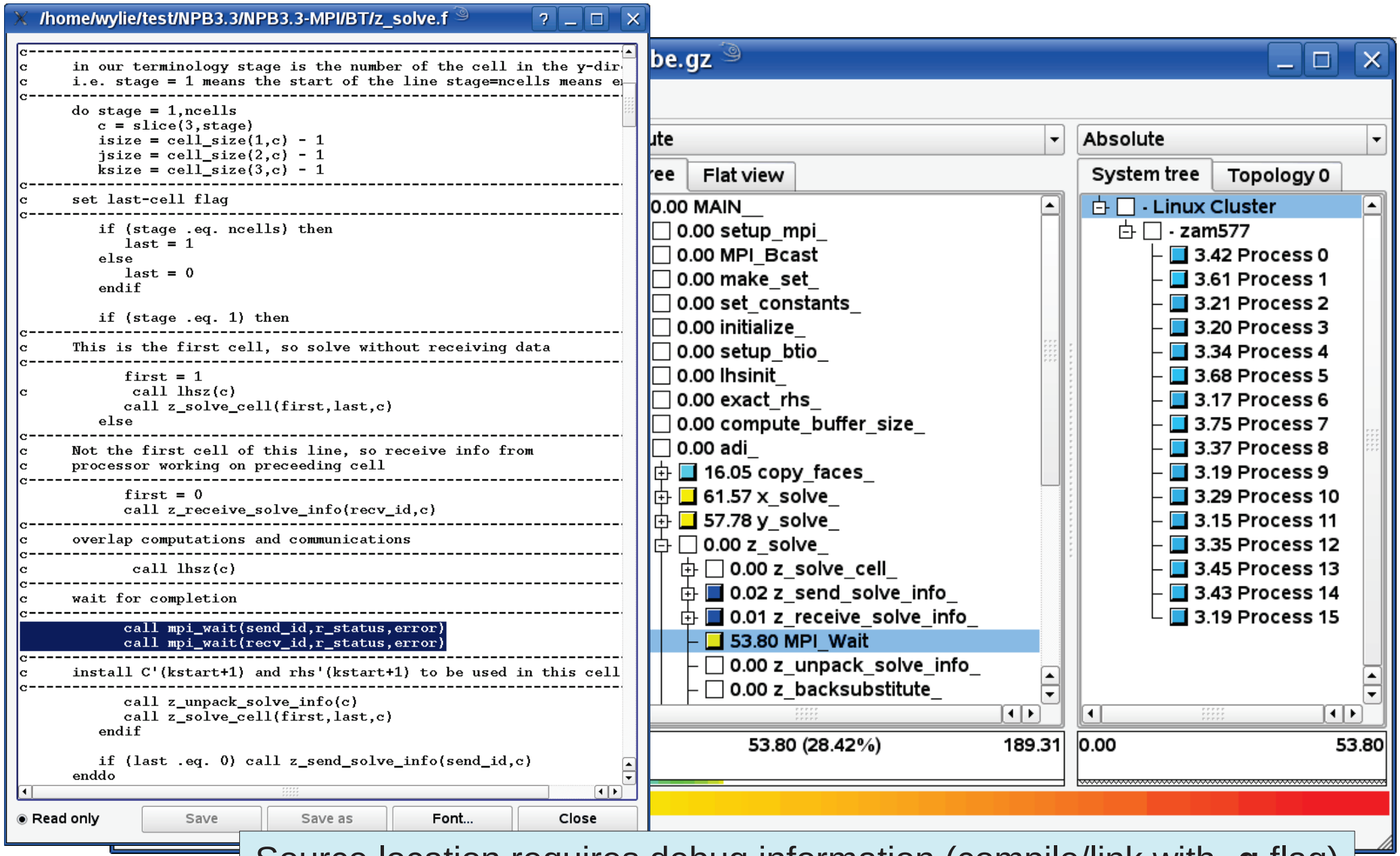
# Analysis report exploration (call tree)



# Analysis report exploration (metric tree)



# Analysis report exploration (source browser)




Source location requires debug information (compile/link with **-g** flag)

# Congratulations!?

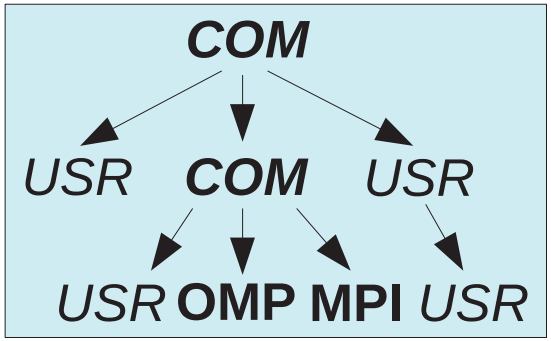

- If you made it this far, you successfully used Scalasca to
  - *instrument* the application
  - *analyze* its execution with a summary measurement, and
  - *examine* it with the interactive analysis report explorer GUI
- ... revealing the call-path profile annotated with
  - Time metrics (including MPI & OpenMP times)
  - Visit counts
  - MPI message statistics (sends/receives, bytes sent/received)
  - Computational imbalance
- ... 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

# BT-MZ summary analysis report scoring

- Report scoring as textual output

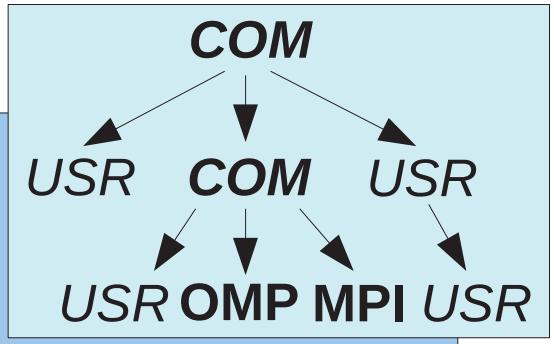
```
% scalasca -examine -s  epik_bt-mz_B_4x4_sum
cube3_score -r ./epik_bt-mz_B_4x4_sum/summary.cube
Reading ./epik_bt-mz_B_4x4_sum/summary.cube... done.
Est. aggregate size of event trace (total_tbc):  39,231,218,072 bytes
Est. size of largest process trace (max_tbc):     2,632,541,576 bytes
(When tracing set ELG_BUFFER_SIZE to avoid intermediate flushes or
 reduce requirements using filter file listing names of USR regions.)

INFO: Score report written to ./epik_bt-mz_B_4x4_sum/epik.score
```

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



# BT-MZ summary analysis report breakdown

- Score report breakdown by region




```
% less  epik_bt-mz_B_4x4_sum/epik.score

flt  type      max_tbc     time        % region
      ANY   2632541576   871.73   100.00 (summary) ALL
      MPI        73064    13.27     1.52 (summary) MPI
      OMP      5186496   496.36    56.94 (summary) OMP
      COM      1087824     3.15     0.36 (summary) COM
      USR   2626194144   358.88    41.17 (summary) USR

      USR    841575744   109.22    12.53 matmul_sub_
      USR    841575744   168.61    19.34 binvcrhs_
      USR    841575744    68.95     7.91 matvec_sub_
      USR     37120680     5.14     0.59 binvrhs_
      USR     37120680     4.10     0.47 lhsinit_
      USR     29960856     2.85     0.33 exact_solution_
      COM       308736     0.82     0.09 copy_x_face_
      COM       308736     0.81     0.09 copy_y_face_
      OMP       283008     2.07     0.24 !$omp parallel @exch_qbc.f:204
      OMP       283008     2.02     0.23 !$omp parallel @exch_qbc.f:215
      OMP       283008     2.16     0.25 !$omp parallel @exch_qbc.f:244
      OMP       283008     2.01     0.23 !$omp parallel @exch_qbc.f:255
...
```

# BT-MZ summary analysis score

- Summary measurement analysis score reveals
  - Total size of event trace would be almost 40GB
  - Maximum trace buffer size would be over 2.5GB per thread
    - smaller buffer would require flushes to disk during measurement resulting in substantial perturbation
  - 99.76% of the trace requirements are for USR regions
    - purely computational routines never found on COM call-paths common to communication routines
  - These USR regions contribute around 40% of total time
    - however, much of that is very likely to be measurement overhead for a few 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

- Report scoring with prospective filter listing USR regions

```
% scalasca -examine -s -f bt.filt  epik_bt-mz_B_4x4_sum
cube3_score -r -f bt.filt ./epik_bt-mz_B_4x4_sum/summary.cube
Applying filter "./bt.filt":
Estimated aggregate size of event trace (total_tbc):  16,852,888 bytes
Estimated size of largest process trace (max_tbc):     1,053,304 bytes

INFO: Score report written to ./epik_bt-mz_B_4x4_sum/epik.score_bt.filt

% less epik_bt-mz_B_4x4_sum/epik.score_bt.filt
flt   type    max_tbc      time       % region
  +    FLT 2626190016   358.88   41.17 (summary) FLT
  *    ANY    6351584   512.85   58.83 (summary) ALL-FLT
  -    MPI      73064    13.27    1.52 (summary) MPI-FLT
  -    OMP    5186496   496.36   56.94 (summary) OMP-FLT
  *    COM    1087824     3.15    0.36 (summary) COM-FLT
  *    USR       4152     0.00    0.00 (summary) USR-FLT

  +    USR  841575744   109.22   12.53 matmul_sub_
  +    USR  841575744   168.61   19.34 binvcrhs_
  +    USR  841575744    68.95    7.91 matvec_sub_
  +    USR   37120680     5.14    0.59 binvrhs_
  +    USR   37120680     4.10    0.47 lhsinit_
  +    USR   29960856     2.85    0.33 exact_solution_
...
```

Filtered routines marked with '+'

```
% cat bt.filt
# bt-mz filter
matmul_sub_
binvcrhs_
matvec_sub_
binvrhs_
lhsinit_
exact_solution_
timer_*
```

# BT-MZ filtered summary measurement

- Rename former measurement archive directory, set new filter configuration and re-run the measurement

```
% mv epik_bt-mz_B_4x4_sum epik_bt-mz_B_4x4_sum.nofilt
% export EPK_FILTER=bt.filt
% OMP_NUM_THREADS=4  scalasca -analyze  mpiexec -np 4 ./bt-mz_B.4
S=C=A=N: Scalasca 1.3 runtime summarization
S=C=A=N: ./epik_bt-mz_4x4_sum experiment archive
S=C=A=N: Sun Mar 29 16:58:34 2009: Collect start
mpiexec -np 4 ./bt-mz_B.4
[00000.0]EPIK: Created new measurement archive ./epik_bt-mz_B_4x4_sum
[00000.0]EPIK: EPK_FILTER “bt.filt” filtered 10 of 113 functions
[00000.0]EPIK: Activated ./epik_bt-mz_B_4x4_sum [NO TRACE] (0.071s)

        [... Application output ...]

[00000.0]EPIK: Closing experiment ./epik_bt-mz_B_4x4_sum
[00000.0]EPIK: 134 unique paths (148 max paths, 7 max frames, 0 unkns)
[00000.0]EPIK: Unifying... done (0.014s)
[00000.0]EPIK: Collating... done (0.059s)
[00000.0]EPIK: Closed experiment ./epik_bt-mz_B_4x4_sum (0.075s)
S=C=A=N: Sun Mar 29 16:58:41 2009: Collect done (status=0) 36s
S=C=A=N: ./epik_bt-mz_B_4x4_sum complete.
```

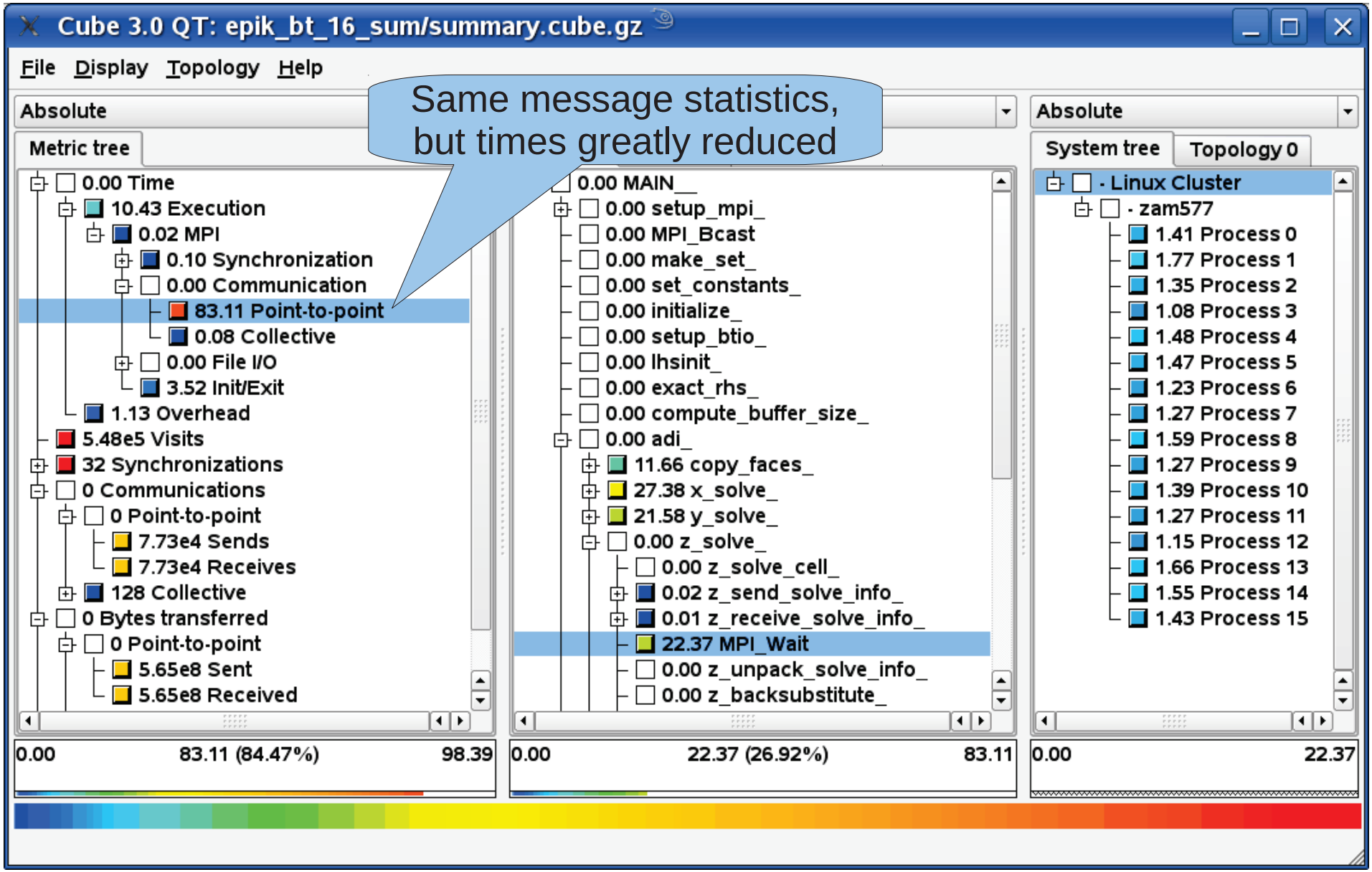
# BT-MZ filtered summary analysis report score

- Scoring of new analysis report as textual output

```
% scalasca -examine -s  epik_bt-mz_B_4x4_sum
INFO: Post-processing runtime summarization result...
cube3_score ./epik_bt-mz_B_4x4_sum/summary.cube
Estimated aggregate size of event trace (total_tbc):  83,920,952 bytes
Estimated size of largest process trace (max_tbc):     6,351,584 bytes
...

INFO: Score report written to ./epik_bt-mz_B_4x4_sum/epik.score

flt  type       max_tbc         time       % region
      ANY       6351584       531.69  100.00 (summary) ALL
      MPI         73064        13.27    2.50 (summary) MPI
      OMP       5186496       515.11   96.88 (summary) OMP
      COM       1087824         3.22    0.61 (summary) COM
      USR          4152         0.00    0.00 (summary) USR
```

- Significant reduction in runtime (measurement overhead)
  - Not only reduced time for USR regions, but OMP reduced too!
- Further measurement tuning (filtering) may be appropriate
  - e.g., "`timer_*`" filters `timer_start_`, `timer_read_`, etc.

# Summary analysis report exploration (filtered)



# BT-MZ trace measurement collection ...

- Re-run the application using Scalasca nexus with “-t” flag

```
% OMP_NUM_THREADS=4 scalasca -analyze -t  mpiexec -np 4 ./bt-mz_B.4
S=C=A=N: Scalasca trace collection and analysis
S=C=A=N: ./epik_bt-mz_B_4x4_trace experiment archive
S=C=A=N: Sun Apr  5 18:50:57 2009: Collect start
mpiexec -np 4 ./bt-mz_B.4
[00000.0]EPIK: Created new measurement archive ./epik_bt-mz_B_4x4_trace
[00000.0]EPIK: EPK_FILTER "npb.filt" filtered 10 of 113 functions
[00000.0]EPIK: Activated ./epik_bt-mz_B_4x4_trace [10000000 bytes] (0.051s)

          [... Application output ...]

[00000.0]EPIK: Closing experiment ./epik_bt-mz_B_4x4_trace [0.069GB] (max 18466028)
[00000.0]EPIK: Flushed 6351570 bytes to file ./epik_bt-mz_B_4x4_trace/ELG/00000
[00000.0]EPIK: 134 unique paths (148 max paths, 7 max frames, 0 unknowns)
[00000.0]EPIK: Unifying... done (0.021s)
[00003.0]EPIK: Flushed 6351570 bytes to file ./epik_bt-mz_B_4x4_trace/ELG/00003
...
[00001.0]EPIK: Flushed 6351570 bytes to file ./epik_bt-mz_B_4x4_trace/ELG/00001
[00000.0]EPIK: 1flush=0.001GB@2.582MB/s, Pflush=0.015GB@35.458MB/s
[00000.0]EPIK: Closed experiment ./epik_bt-mz_B_4x4_trace (0.178s)
S=C=A=N: Sun Apr  5 18:51:05 2009: Collect done (status=0) 41s
[.. continued ...]
```

- Separate trace file per MPI rank written straight into new experiment directory ./epik_bt-mz_B_4x4_trace

# BT-MZ trace measurement … analysis

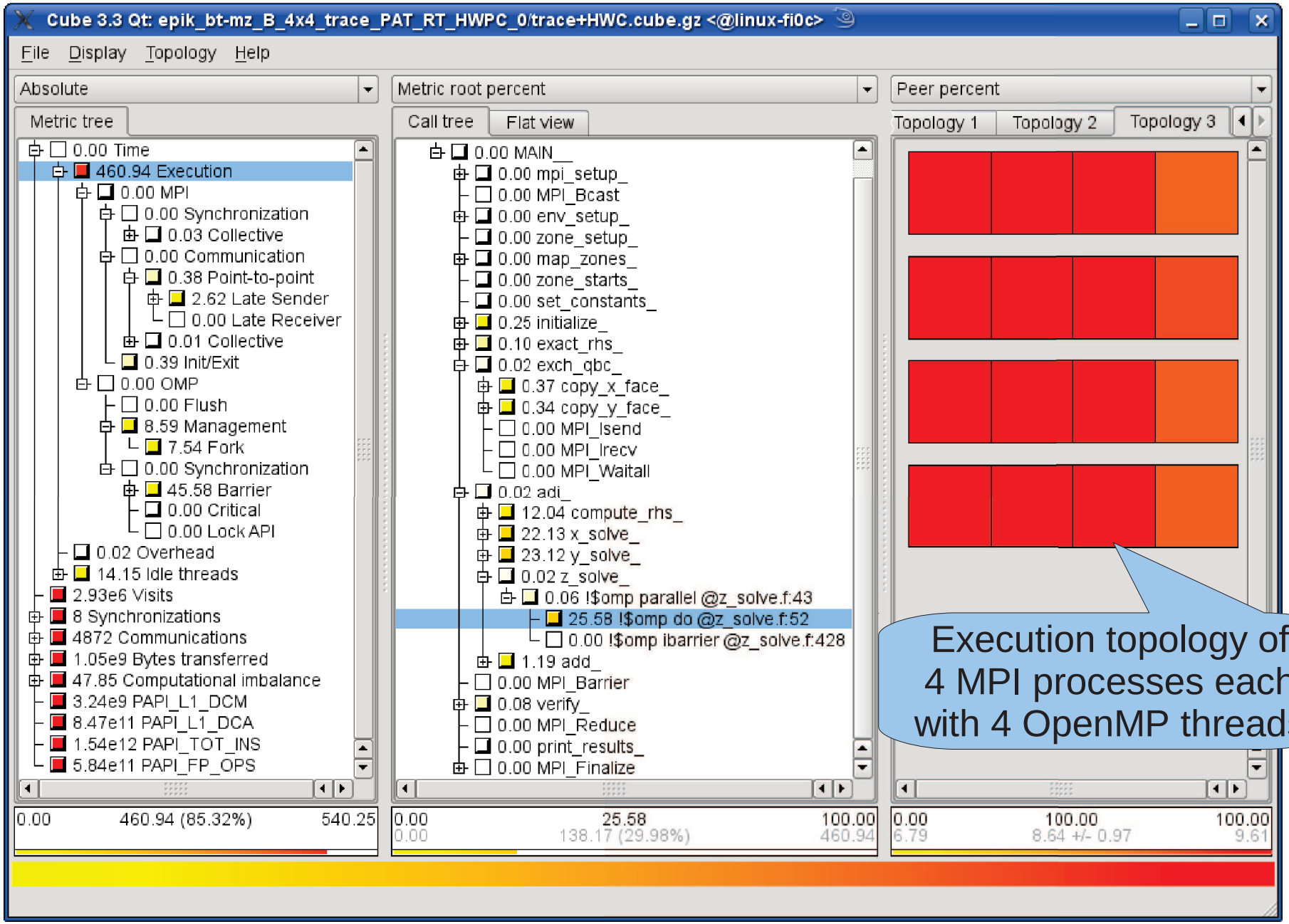

- Continues with automatic (parallel) analysis of trace files

```
S=C=A=N: Sun Apr  5 18:51:05 2009: Analyze start
mpiexec -np 4 scout.hyb ./epik_bt-mz_B_4x4_trace
SCOUT   Copyright (c) 1998-2009 Forschungszentrum Juelich GmbH

Analyzing experiment archive ./epik_bt-mz_B_4x4_trace

Reading definitions file  ... done (0.563s).
Reading event trace files ... done (0.495s).
Preprocessing             ... done (0.134s).
Analyzing event traces    ... done (2.186s).
Writing CUBE report       ... done (0.160s).

Total processing time     : 3.737s
Max. memory usage         : 47.504MB

S=C=A=N: Sun Apr  5 18:51:09 2009: Analyze done (status=0) 4s
S=C=A=N: ./epik_bt-mz_B_4x4_trace complete.
```

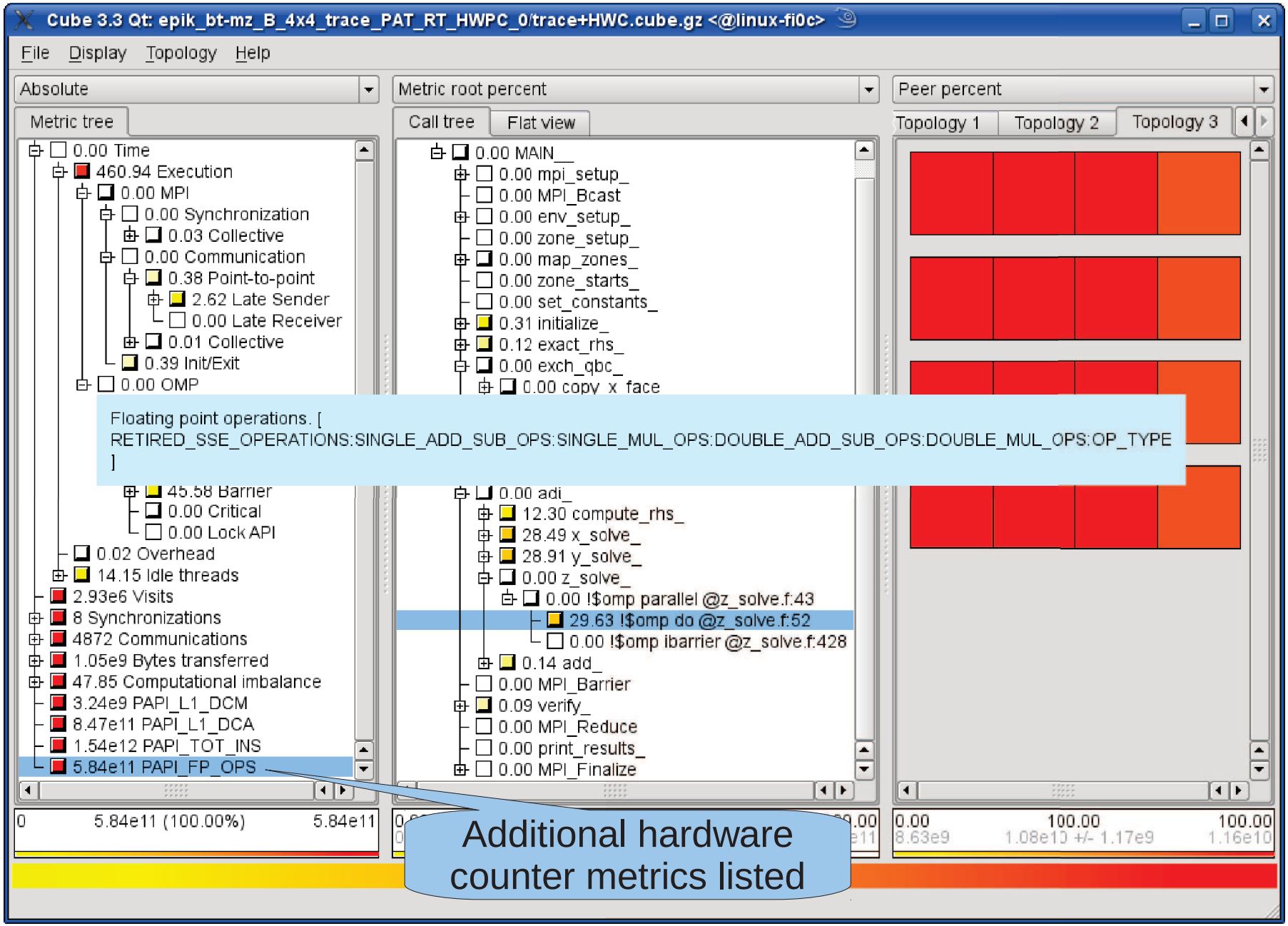
- Produces trace analysis report in experiment directory

```
% scalasca -examine epik_bt-mz_B_4x4_trace
INFO: Post-processing runtime summarization result...
INFO: Post-processing trace analysis report ...
INFO: Displaying ./epik_bt-mz_B_4x4_trace/trace.cube...

                [GUI showing trace analysis report]
```

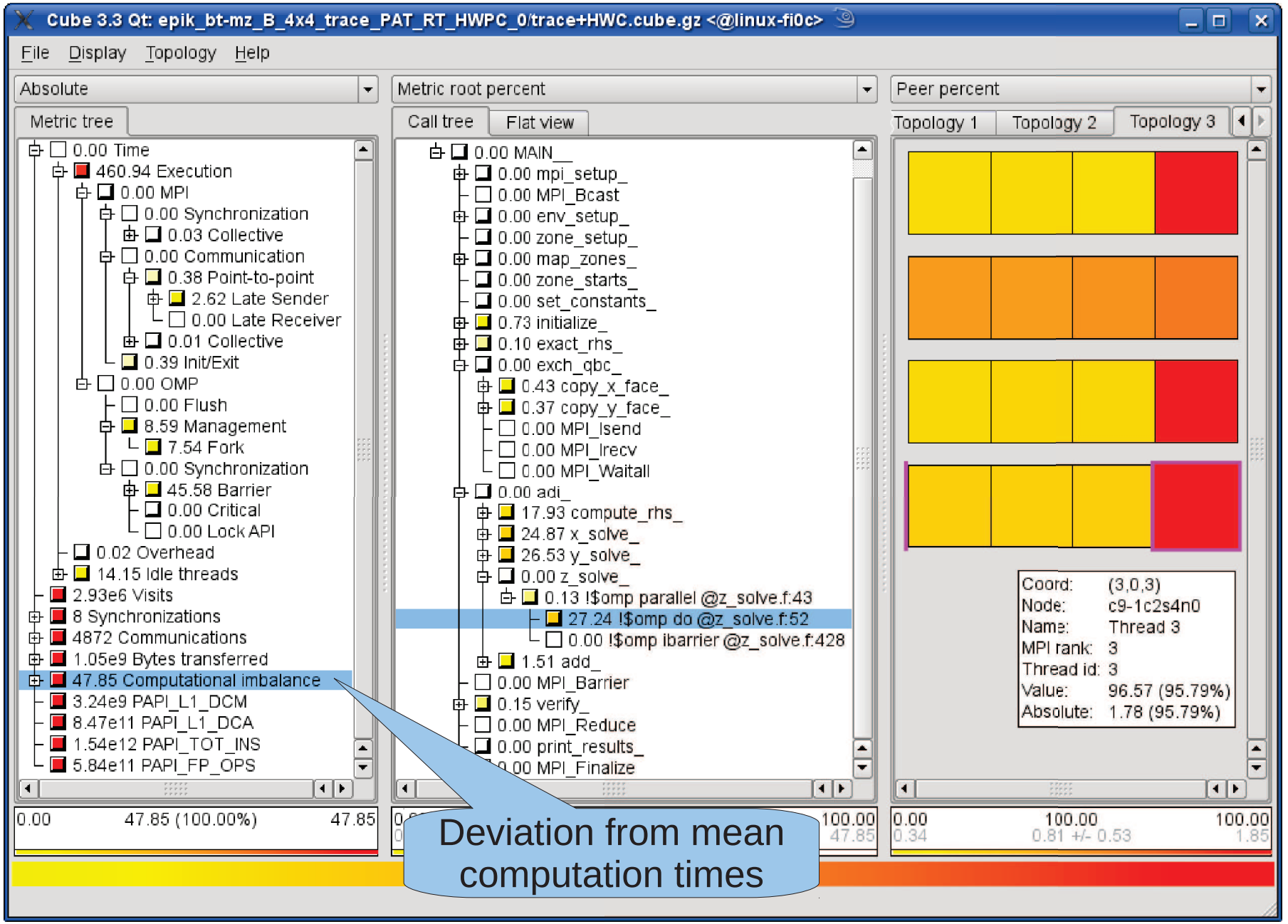
# Scalasca topological presentation



# Scalasca hardware counter metrics



# Scalasca computational imbalance heuristic



# Scalasca trace analysis report exploration