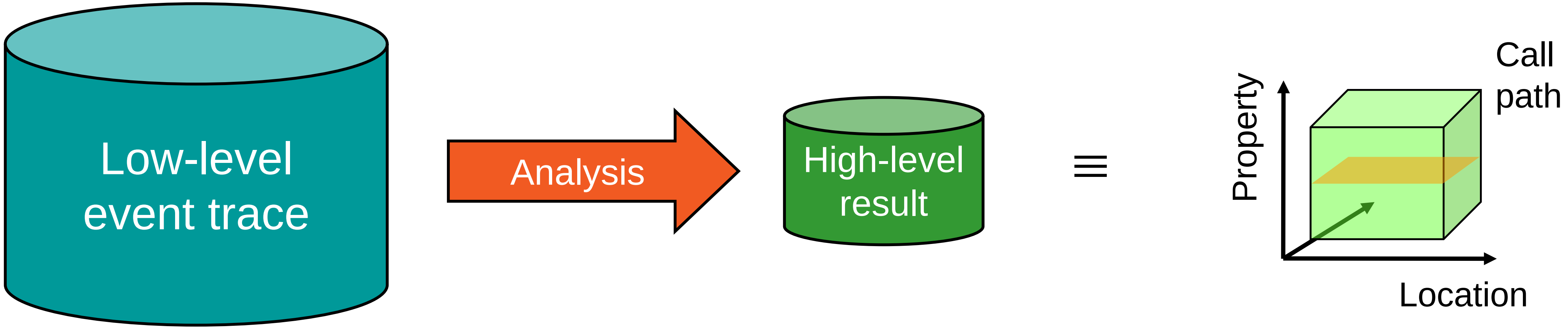
VIRTUAL INSTITUTE – HIGH PRODUCTIVITY SUPERCOMPUTING

Automatic trace analysis with Scalasca

Marc-André Hermanns
Jülich Supercomputing Centre

Automatic trace analysis

- Idea
  - Automatic search for patterns of inefficient behavior
  - Classification of behavior & quantification of significance



  - Guaranteed to cover the entire event trace
  - Quicker than manual/visual trace analysis
  - Parallel replay analysis exploits available memory & processors to deliver scalability

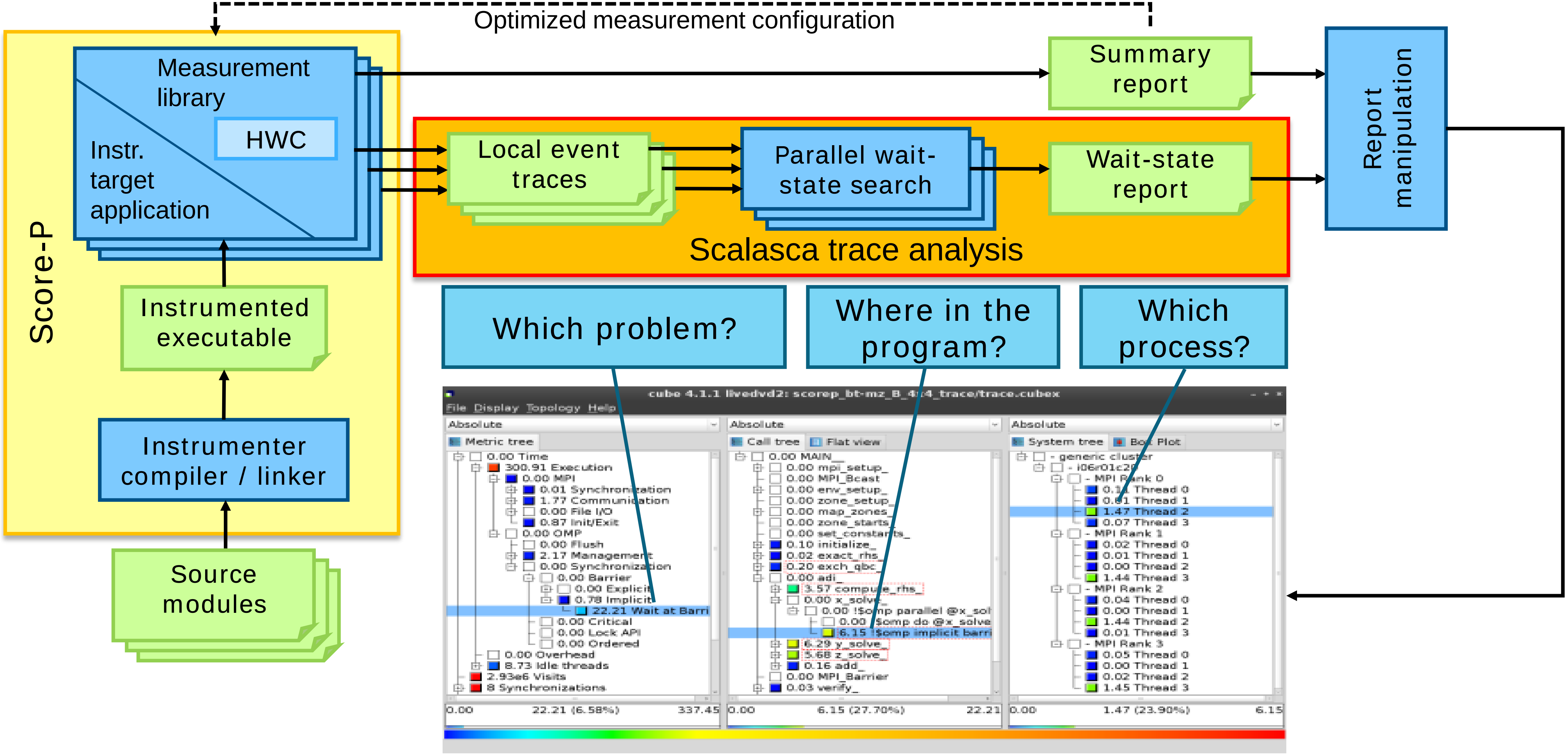
The Scalasca project: Objective

- Development of a **scalable** performance analysis toolset for most popular parallel programming paradigms
- Specifically targeting **large-scale** parallel applications
  - Such as those running on IBM Blue Gene or Cray systems with one million or more processes/threads
- Latest release:
  - Scalasca v2.3.1 coordinated with Score-P v2.0.2 (May 2016)

Scalasca features

- Open source, 3-clause BSD license
- Fairly portable
  - IBM Blue Gene, Cray XT/XE/XK/XC, SGI Altix, Fujitsu FX10/100 & K computer, Linux clusters (x86, Power, ARM), Intel Xeon Phi, ...
- Uses Score-P instrumenter & measurement libraries
  - Scalasca 2 core package focuses on trace-based analyses
  - Supports common data formats
    - Reads event traces in OTF2 format
    - Writes analysis reports in CUBE4 format
- Current limitations:
  - Unable to handle traces containing CUDA or SHMEM events, or OpenMP nested parallelism
  - PAPI/rusage metrics for trace events are ignored

Scalasca workflow



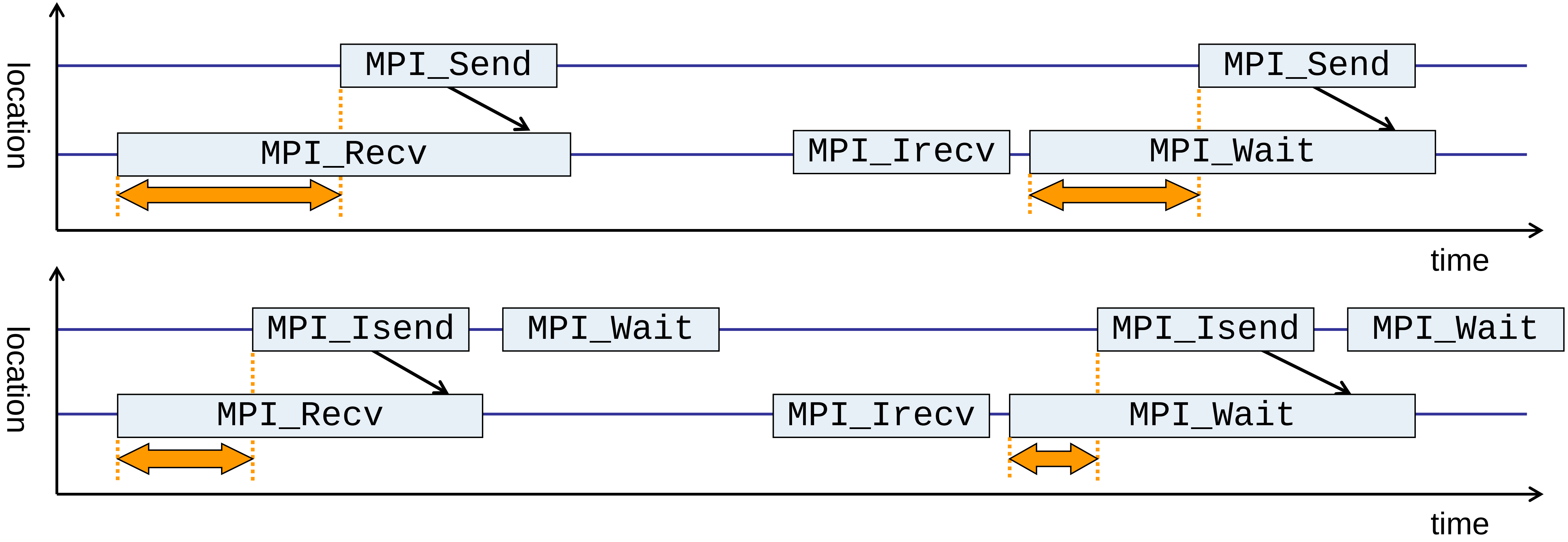
```
cube 4.1.1 livedvd2: scorep_bt-mz_B_4x4_trace/trace.cubex
```

```
0.00 Time
  300.91 Execution
    0.00 MPI
      0.01 Synchronization
      1.77 Communication
      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 Explicit
          0.78 Implicit
            22.21 Wait at Barri
        0.00 Critical
        0.00 Lock API
        0.00 Ordered
    0.00 Overhead
    8.73 Idle threads
  2.93e6 Visits
  8 Synchronizations
0.00          22.21 (6.58%)          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_
  0.10 initialize_
  0.02 exact_rhs_
  0.20 exch_qbc_
  0.00 adi_
    3.57 compute_rhs_
    0.00 x_solve_
      0.00 !$omp parallel @x_sol
        0.00 !$omp do @x_solve
        6.15 !$omp implicit barri
    6.29 y_solve_
    5.68 z_solve_
    0.16 add_
  0.00 MPI_Barrier
  0.03 verify_
0.00          6.15 (27.70%)          22.21
```

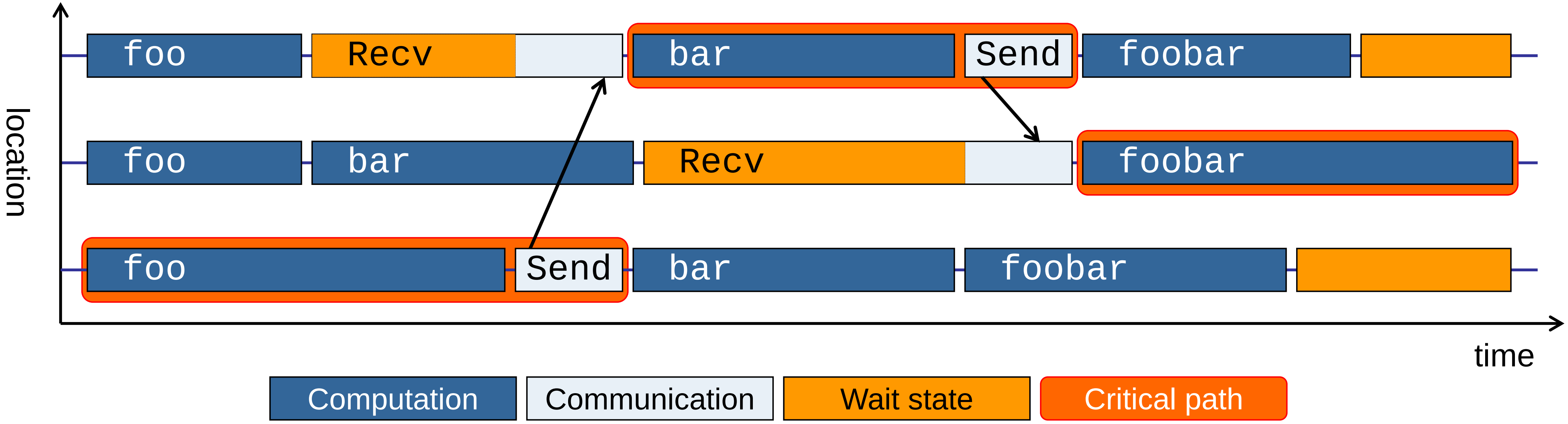
```
- generic cluster
  - i06r01c20
    - MPI Rank 0
      0.11 Thread 0
      0.01 Thread 1
      1.47 Thread 2
      0.07 Thread 3
    - MPI Rank 1
      0.02 Thread 0
      0.01 Thread 1
      0.00 Thread 2
      1.44 Thread 3
    - MPI Rank 2
      0.04 Thread 0
      0.00 Thread 1
      1.44 Thread 2
      0.01 Thread 3
    - MPI Rank 3
      0.05 Thread 0
      0.00 Thread 1
      0.02 Thread 2
      1.45 Thread 3
0.00          1.47 (23.90%)          6.15
```

Example: “*Late Sender*” wait state



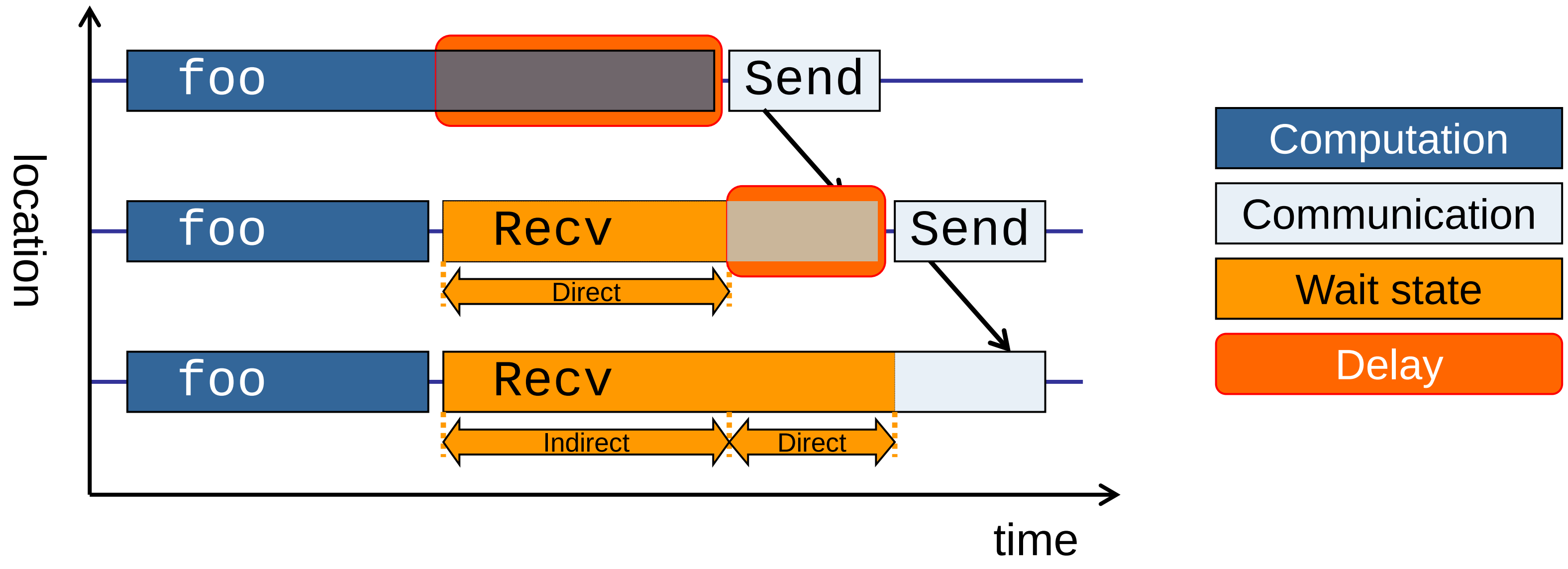
- Waiting time caused by a blocking receive operation posted earlier than the corresponding send
- Applies to blocking as well as non-blocking communication

Example: Critical path



- Shows call paths and processes/threads that are responsible for the program’s wall-clock runtime
- Identifies good optimization candidates and parallelization bottlenecks

Example: Root-cause analysis



- Classifies wait states into direct and indirect (i.e., caused by other wait states)
- Identifies *delays* (excess computation/communication) as root causes of wait states
- Attributes wait states as *delay costs*

Hands-on:
NPB-MZ-MPI / BT

scalasca

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 Summary measurement collection with filtering
- 2.2 Filtered summary analysis report examination

- 3.0 Event trace collection
- 3.1 Event trace examination & analysis

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)

Setup environment

- Remember load modules to add local tool installations to $PATH

```
% module purge
% module load default-impi-LATEST
% module load cube scorep/2.0.2/intel-impi-latest scalasca/2.3.1/intel-impi-latest
```

- Change to directory containing NPB3.3-MZ-MPI sources
- Existing instrumented executable in bin.scorep/ directory can be reused

```
% cd $HOME/scratch/NPB3.3-MZ-MPI
```

BT-MZ summary measurement collection...

```
% cd bin.scorep
% cp ../jobscript/darwin/scalasca.sbatch .
% vi scalasca.sbatch

 [...]

export SCOREP_FILTERING_FILE=../config/scorep.filt
#export SCOREP_TOTAL_MEMORY=21MB

# Scalasca configuration
#export SCAN_ANALYZE_OPTS=“--time-correct”

scalasca -analyze mpirun –np 8 ./bt-mz_${CLASS}.${PROCS}
```

- Change to directory with the executable and edit the job script

```
% sbatch scalasca.sbatch
```

- Submit the job

BT-MZ summary measurement

```
S=C=A=N: Scalasca 2.3.1 runtime summarization
S=C=A=N: ./scorep_bt-mz_B_8x4_sum experiment archive
S=C=A=N: Sat Jul  2 16:12:23 2016: Collect start
    mpirun -np 8 ./bt-mz_B.8

 NAS Parallel Benchmarks (NPB3.3-MZ-MPI) - BT-MZ MPI+OpenMP \
 Benchmark

 Number of zones:   8 x   8
 Iterations: 200    dt:   0.000300
 Number of active processes:     8

 [... More application output ...]

S=C=A=N: Sat Jul  2 16:12:36 2016: Collect done (status=0) 13s
S=C=A=N: ./scorep_bt-mz_B_8x4_sum complete.
```

- Run the application using the Scalasca measurement collection & analysis nexus prefixed to launch command
- Creates experiment directory: ./scorep_bt-mz_B_8x4_sum

BT-MZ summary analysis report examination

- Score summary analysis report

```
% square -s  scorep_bt-mz_B_8x4_sum
INFO: Post-processing runtime summarization result...
INFO: Score report written to ./scorep_bt-mz_B_8x4_sum/scorep.score
```

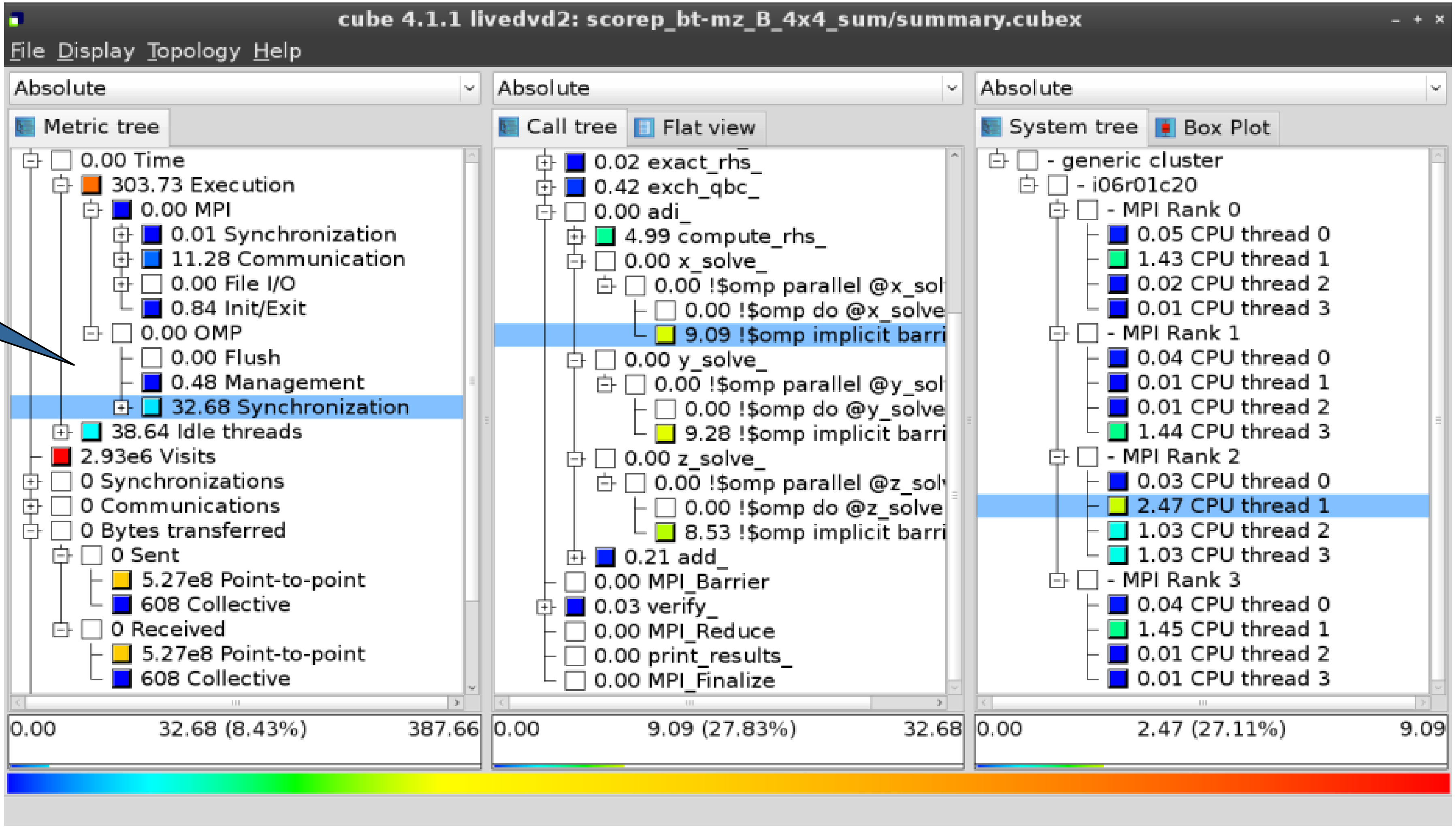
- Post-processing and interactive exploration with Cube

```
% square  scorep_bt-mz_B_8x4_sum
INFO: Displaying ./scorep_bt-mz_B_8x4_sum/summary.cubex...

                    [GUI showing summary analysis report]
```

- The post-processing derives additional metrics and generates a structured metric hierarchy

Post-processed summary analysis report

Split base metrics into more specific metrics



cube 4.1.1 livedvd2: scorep_bt-mz_B_4x4_sum/summary.cubex

```
0.00 Time
  303.73 Execution
    0.00 MPI
      0.01 Synchronization
      11.28 Communication
      0.00 File I/O
      0.84 Init/Exit
    0.00 OMP
      0.00 Flush
      0.48 Management
      32.68 Synchronization
  38.64 Idle threads
2.93e6 Visits
0 Synchronizations
0 Communications
0 Bytes transferred
  0 Sent
    5.27e8 Point-to-point
    608 Collective
  0 Received
    5.27e8 Point-to-point
    608 Collective

0.00          32.68 (8.43%)          387.66
```

```
0.02 exact_rhs_
0.42 exch_qbc_
0.00 adi_
  4.99 compute_rhs_
  0.00 x_solve_
    0.00 !$omp parallel @x_sol
      0.00 !$omp do @x_solve
      9.09 !$omp implicit barri
  0.00 y_solve_
    0.00 !$omp parallel @y_sol
      0.00 !$omp do @y_solve
      9.28 !$omp implicit barri
  0.00 z_solve_
    0.00 !$omp parallel @z_sol
      0.00 !$omp do @z_solve
      8.53 !$omp implicit barri
  0.21 add_
0.00 MPI_Barrier
0.03 verify_
0.00 MPI_Reduce
0.00 print_results_
0.00 MPI_Finalize

0.00          9.09 (27.83%)          32.68
```

```
- generic cluster
  - i06r01c20
    - MPI Rank 0
      0.05 CPU thread 0
      1.43 CPU thread 1
      0.02 CPU thread 2
      0.01 CPU thread 3
    - MPI Rank 1
      0.04 CPU thread 0
      0.01 CPU thread 1
      0.01 CPU thread 2
      1.44 CPU thread 3
    - MPI Rank 2
      0.03 CPU thread 0
      2.47 CPU thread 1
      1.03 CPU thread 2
      1.03 CPU thread 3
    - MPI Rank 3
      0.04 CPU thread 0
      1.45 CPU thread 1
      0.01 CPU thread 2
      0.01 CPU thread 3

0.00          2.47 (27.11%)          9.09
```

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 Summary measurement collection with filtering
- 2.2 Filtered summary analysis report examination

- 3.0 Event trace collection
- 3.1 Event trace examination & analysis

BT-MZ trace measurement collection...

```
% cd bin.scorep
% cp ../jobscript/darwin/scalasca.sbatch .
% vi scalasca.sbatch

 [...]

export SCOREP_FILTERING_FILE=../config/scorep.filt
export SCOREP_TOTAL_MEMORY=21MB

# Scalasca configuration
export SCAN_ANALYZE_OPTS=“--time-correct”

scalasca –analyze -t mpirun –np 8 ./bt-mz_${CLASS}.${PROCS}
```

- Change to directory with the executable and edit the job script

```
% sbatch scalasca.sbatch
```

- Submit the job

BT-MZ trace measurement ... collection

```
S=C=A=N: Scalasca 2.3.1 trace collection and analysis
S=C=A=N: ./scorep_bt-mz_B_8x4_trace experiment archive
S=C=A=N: Sat Jul  2 16:20:56 2016: Collect start \
  mpirun -np 8 ./bt-mz_B.8


 NAS Parallel Benchmarks (NPB3.3-MZ-MPI) - BT-MZ MPI+OpenMP \
 Benchmark

 Number of zones:   8 x   8
 Iterations: 200    dt:   0.000300
 Number of active processes:     8

 [... More application output ...]

S=C=A=N: Sat Jul  2 16:21:11 2016: Collect done (status=0) 15s
```

- Starts measurement with collection of trace files ...

BT-MZ trace measurement ... analysis

```
S=C=A=N: Sat Jul  2 16:21:11 2016: Analyze start \
 mpirun -np 8 scout.hyb ./scorep_bt-mz_B_8x4_trace/traces.otf2
SCOUT   Copyright (c) 1998-2016 Forschungszentrum Juelich GmbH
        Copyright (c) 2009-2014 German Research School for Simulation
                                Sciences GmbH

Analyzing experiment archive ./scorep_bt-mz_B_8x4_trace/traces.otf2

Opening experiment archive ... done (0.073s).
Reading definition data    ... done (0.053s).
Reading event trace data   ... done (0.106s).
Preprocessing              ... done (0.125s).
Analyzing trace data       ... done (2.778s).
Writing analysis report    ... done (0.260s).

Max. memory usage         : 171.227MB

Total processing time     : 3.436s
S=C=A=N: Sat Jul  2 16:21:16 2016: Analyze done (status=0) 5s
```

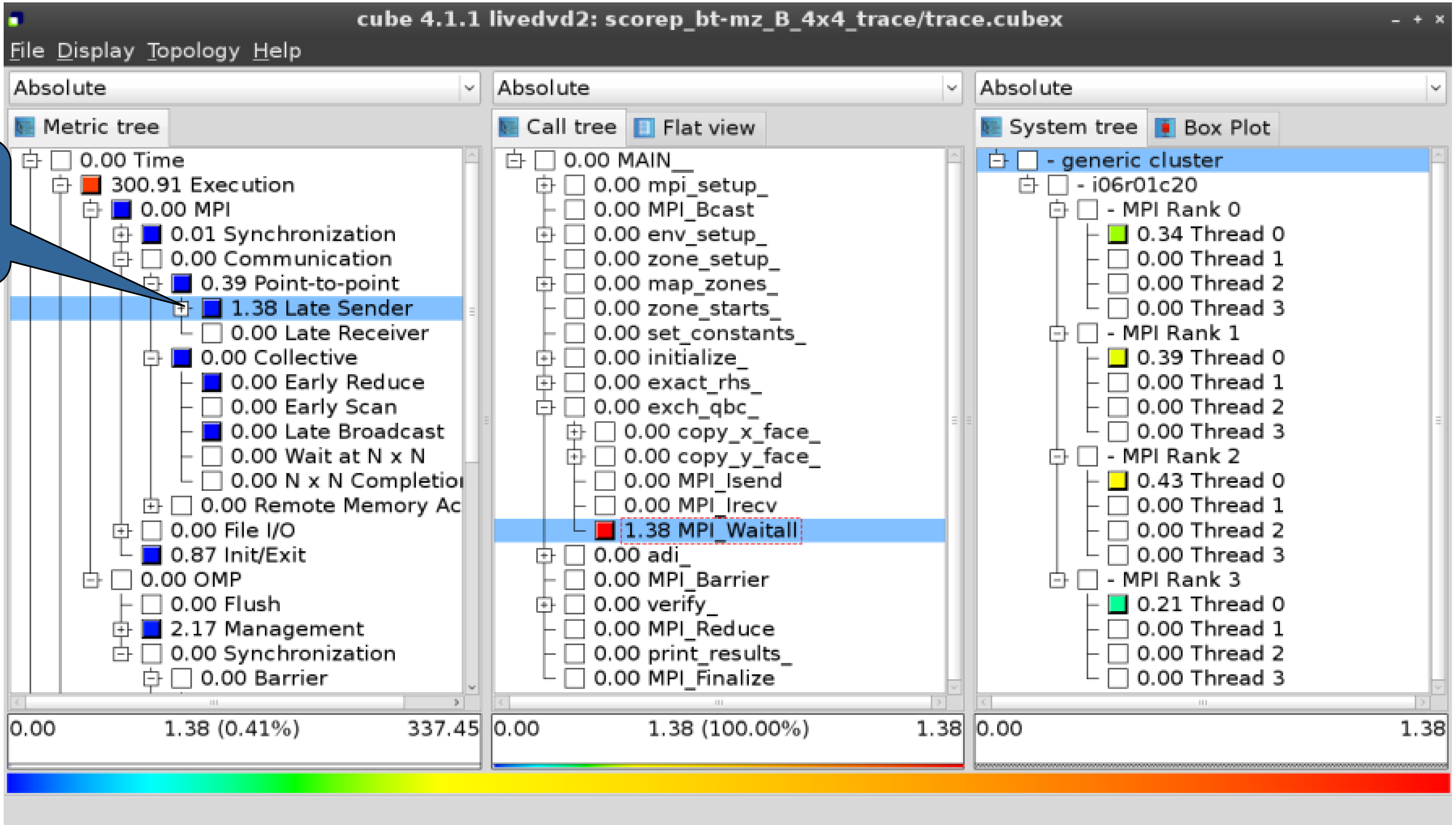
- Continues with automatic (parallel) analysis of trace files

BT-MZ trace analysis report exploration

- Produces trace analysis report in the experiment directory containing trace-based wait-state metrics

```
% square  scorep_bt-mz_B_8x4_trace
INFO: Post-processing runtime summarization result...
INFO: Post-processing trace analysis report...
INFO: Displaying ./scorep_bt-mz_B_8x4_trace/trace.cubex...

               [GUI showing trace analysis report]
```

Post-processed trace analysis report



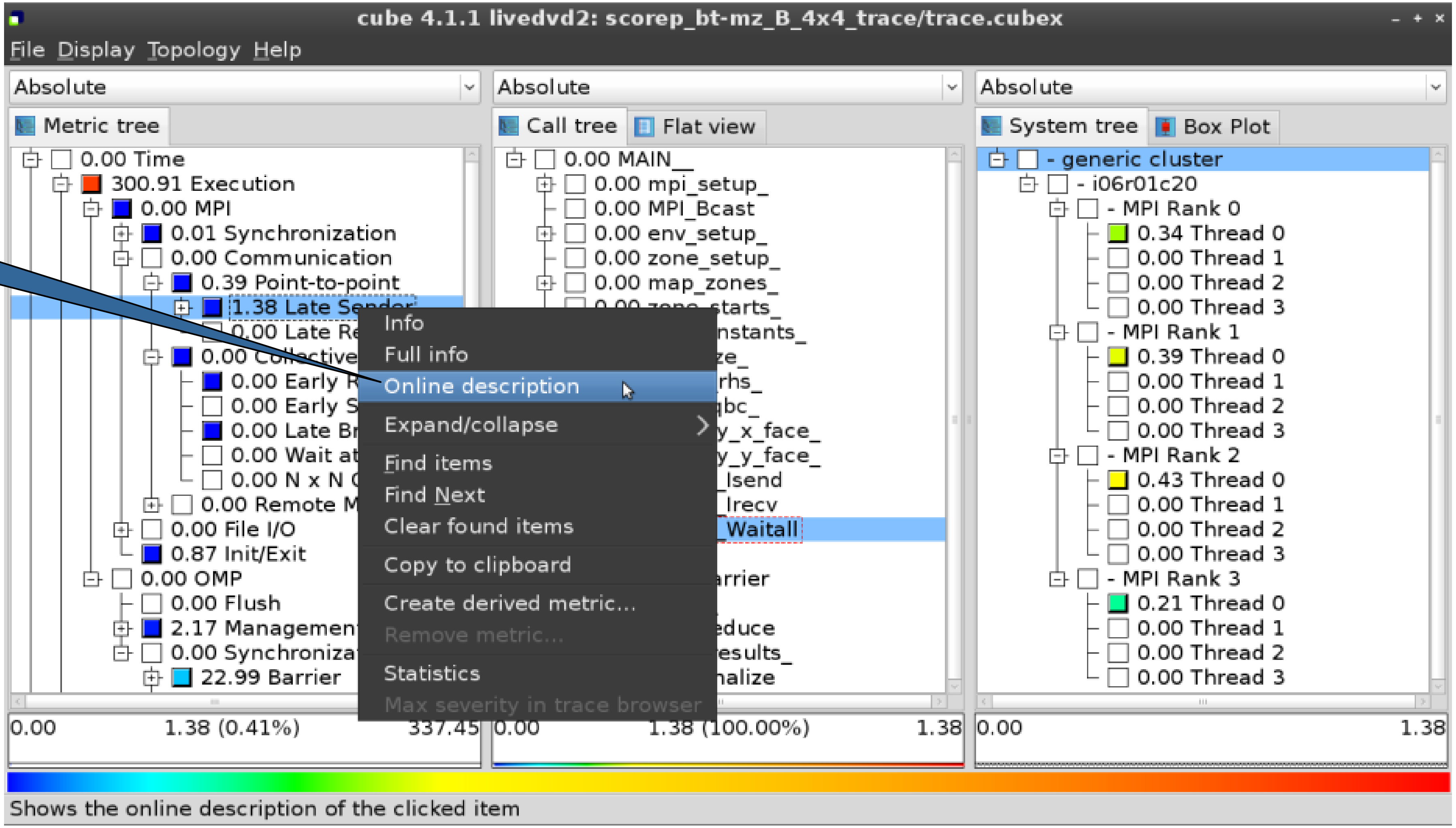
cube 4.1.1 livedvd2: scorep_bt-mz_B_4x4_trace/trace.cubex

```
Metric tree                         Call tree                      System tree
0.00 Time                           0.00 MAIN__                    - generic cluster
  300.91 Execution                    0.00 mpi_setup_                - i06r01c20
    0.00 MPI                          0.00 MPI_Bcast                   - MPI Rank 0
      0.01 Synchronization            0.00 env_setup_                    0.34 Thread 0
      0.00 Communication              0.00 zone_setup_                   0.00 Thread 1
        0.39 Point-to-point           0.00 map_zones_                    0.00 Thread 2
          1.38 Late Sender            0.00 zone_starts_                  0.00 Thread 3
          0.00 Late Receiver          0.00 set_constants_              - MPI Rank 1
        0.00 Collective               0.00 initialize_                   0.39 Thread 0
          0.00 Early Reduce           0.00 exact_rhs_                    0.00 Thread 1
          0.00 Early Scan             0.00 exch_qbc_                     0.00 Thread 2
          0.00 Late Broadcast           0.00 copy_x_face_                0.00 Thread 3
          0.00 Wait at N x N            0.00 copy_y_face_              - MPI Rank 2
          0.00 N x N Completioi         0.00 MPI_Isend                   0.43 Thread 0
        0.00 Remote Memory Ac           0.00 MPI_Irecv                   0.00 Thread 1
      0.00 File I/O                     1.38 MPI_Waitall                 0.00 Thread 2
      0.87 Init/Exit                  0.00 adi_                          0.00 Thread 3
    0.00 OMP                          0.00 MPI_Barrier                 - MPI Rank 3
      0.00 Flush                      0.00 verify_                       0.21 Thread 0
      2.17 Management                 0.00 MPI_Reduce                    0.00 Thread 1
      0.00 Synchronization            0.00 print_results_                0.00 Thread 2
        0.00 Barrier                  0.00 MPI_Finalize                  0.00 Thread 3

0.00   1.38 (0.41%)   337.45       0.00   1.38 (100.00%)   1.38        0.00   1.38
```

Additional trace-based metrics in metric hierarchy

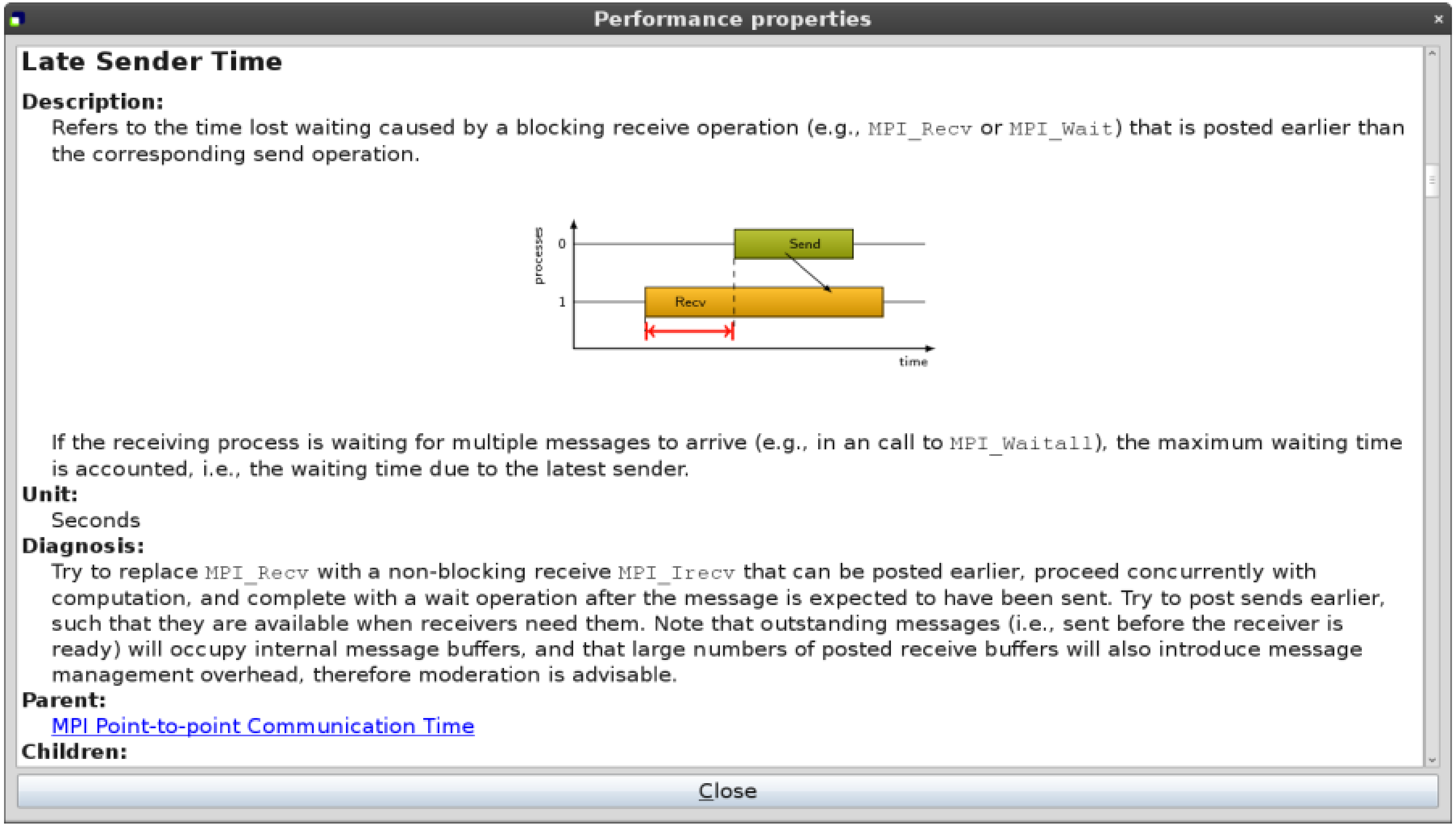
Online metric description

Access online metric description via context menu



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.

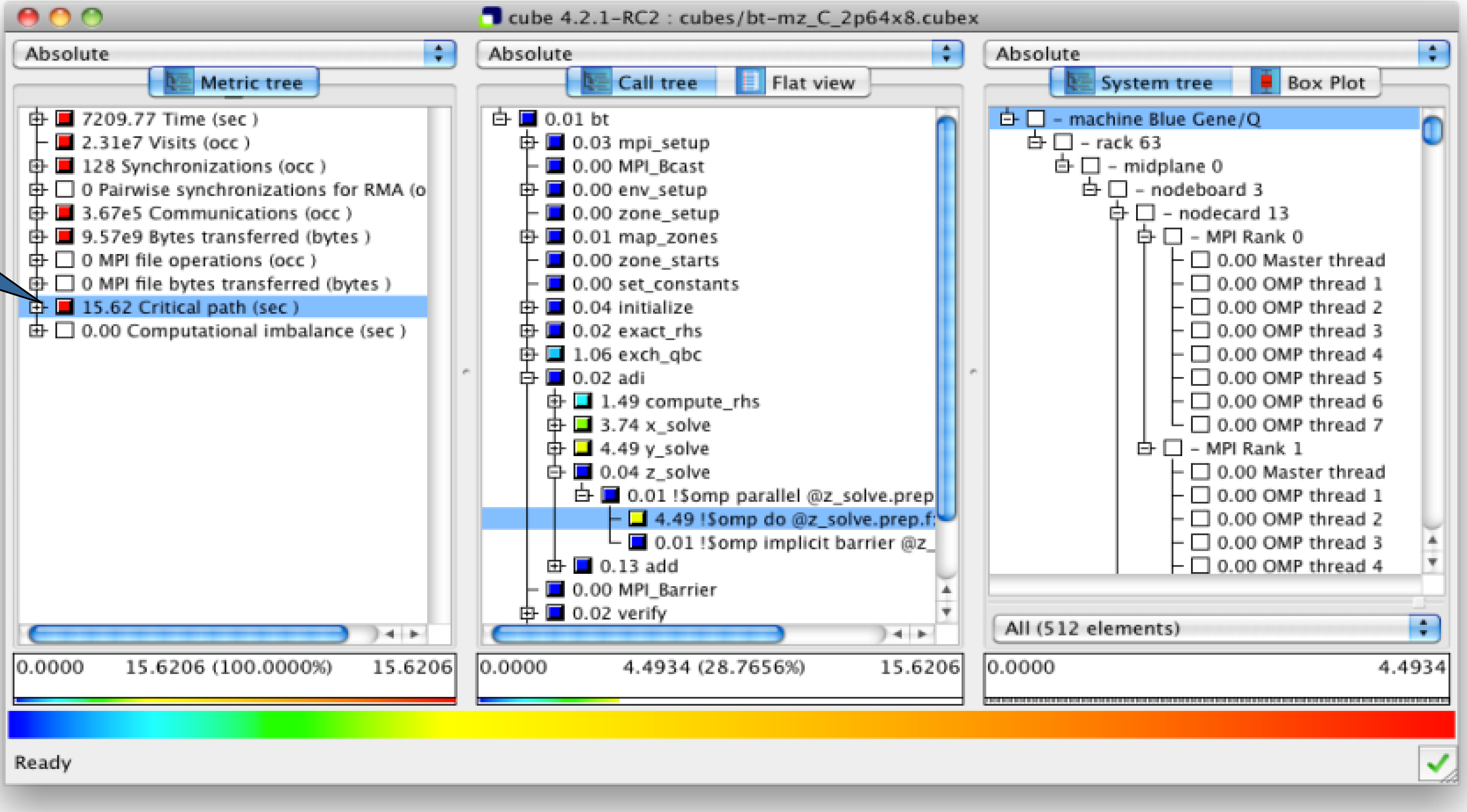
Parent:

MPI Point-to-point Communication Time

Children:

Critical-path analysis

Critical-path profile shows wall-clock time impact

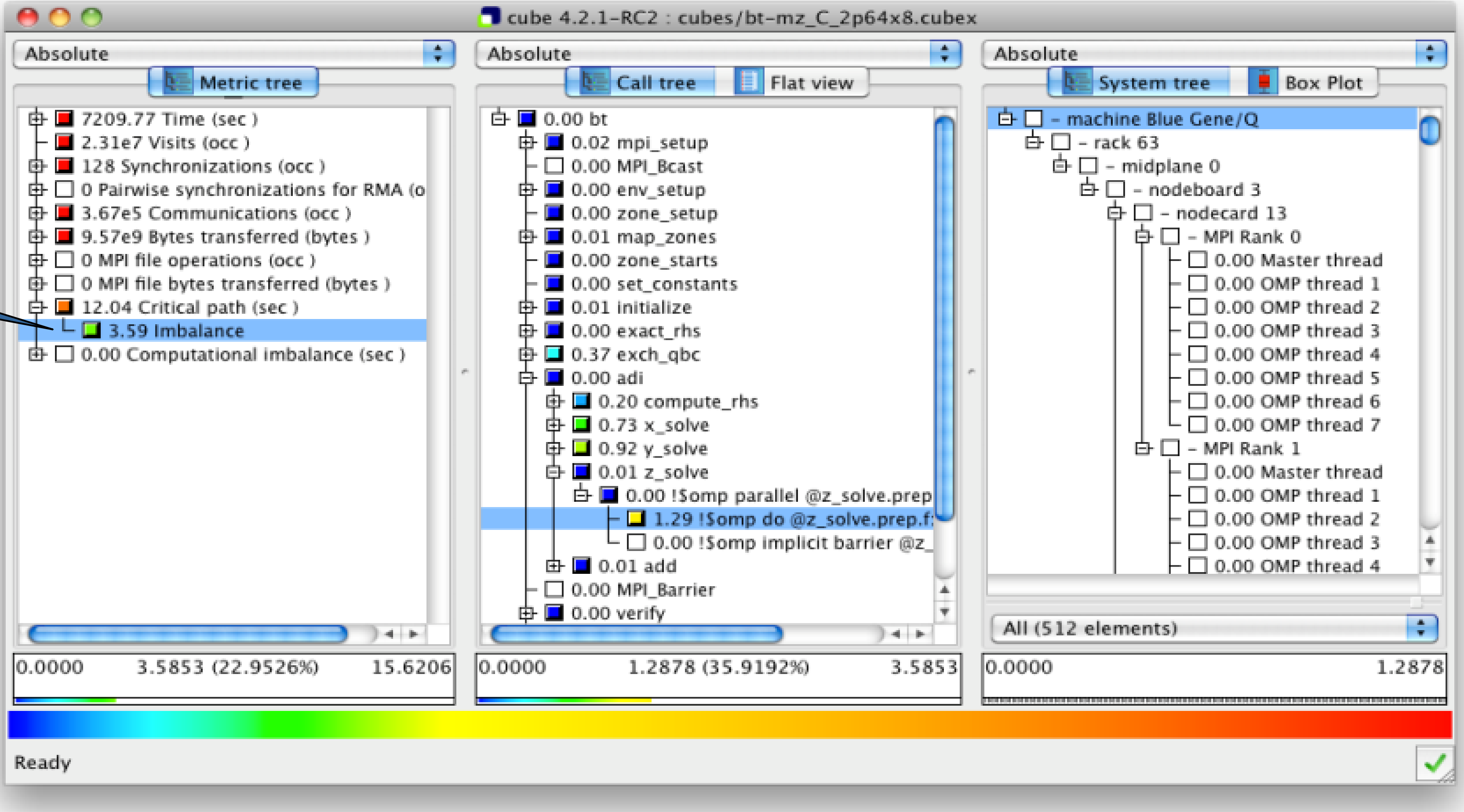


cube 4.2.1-RC2 : cubes/bt-mz_C_2p64x8.cubex

```
Absolute                          Absolute                                Absolute
Metric tree                       Call tree | Flat view                   System tree | Box Plot

7209.77 Time (sec )               0.01 bt                                 - machine Blue Gene/Q
2.31e7 Visits (occ )                0.03 mpi_setup                          - rack 63
128 Synchronizations (occ )         0.00 MPI_Bcast                            - midplane 0
0 Pairwise synchronizations for RMA (o   0.00 env_setup                         - nodeboard 3
3.67e5 Communications (occ )        0.00 zone_setup                               - nodecard 13
9.57e9 Bytes transferred (bytes )   0.01 map_zones                                  - MPI Rank 0
0 MPI file operations (occ )        0.00 zone_starts                                  0.00 Master thread
0 MPI file bytes transferred (bytes )   0.00 set_constants                           0.00 OMP thread 1
15.62 Critical path (sec )          0.04 initialize                                   0.00 OMP thread 2
0.00 Computational imbalance (sec ) 0.02 exact_rhs                                    0.00 OMP thread 3
                                    1.06 exch_qbc                                     0.00 OMP thread 4
                                    0.02 adi                                          0.00 OMP thread 5
                                      1.49 compute_rhs                                0.00 OMP thread 6
                                      3.74 x_solve                                    0.00 OMP thread 7
                                      4.49 y_solve                                  - MPI Rank 1
                                      0.04 z_solve                                    0.00 Master thread
                                        0.01 !$omp parallel @z_solve.prep             0.00 OMP thread 1
                                          4.49 !$omp do @z_solve.prep.f:              0.00 OMP thread 2
                                          0.01 !$omp implicit barrier @z_             0.00 OMP thread 3
                                      0.13 add                                        0.00 OMP thread 4
                                    0.00 MPI_Barrier
                                    0.02 verify                           All (512 elements)

0.0000   15.6206 (100.0000%)   15.6206   0.0000   4.4934 (28.7656%)   15.6206   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

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 )
12.04 Critical path (sec )
  3.59 Imbalance
0.00 Computational imbalance (sec )
```

0.0000   3.5853 (22.9526%)   15.6206

Absolute | Call tree | Flat view

```
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
```

0.0000   1.2878 (35.9192%)   3.5853

Absolute | System tree | Box Plot

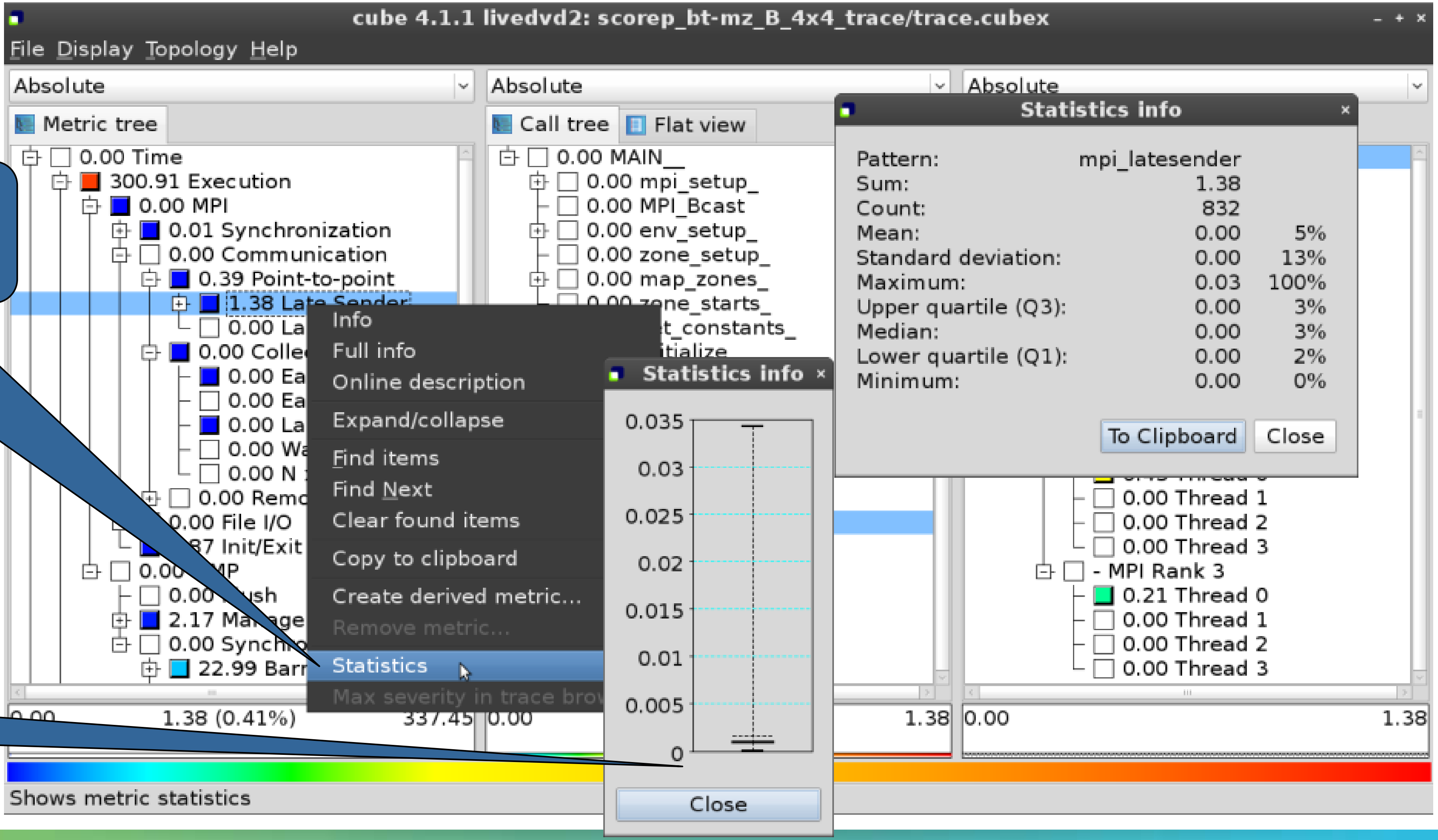
```
- machine Blue Gene/Q
  - rack 63
    - midplane 0
      - nodeboard 3
        - nodecard 13
          - MPI Rank 0
            0.00 Master thread
            0.00 OMP thread 1
            0.00 OMP thread 2
            0.00 OMP thread 3
            0.00 OMP thread 4
            0.00 OMP thread 5
            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   1.2878

Ready

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%

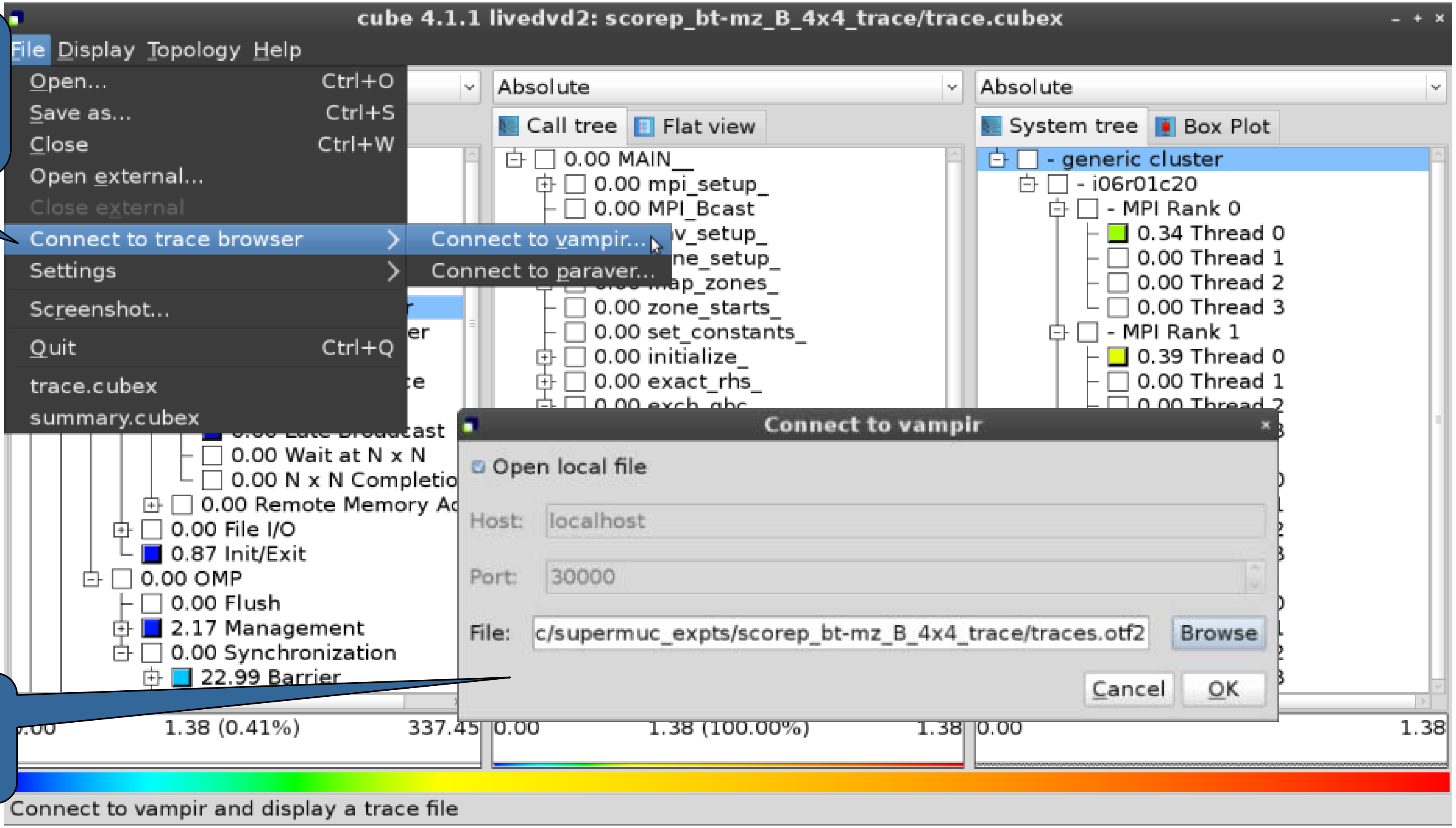
To Clipboard Close

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…



cube 4.1.1 livedvd2: scorep_bt-mz_B_4x4_trace/trace.cubex

Connect to vampir

☑ Open local file

Host: localhost

Port: 30000

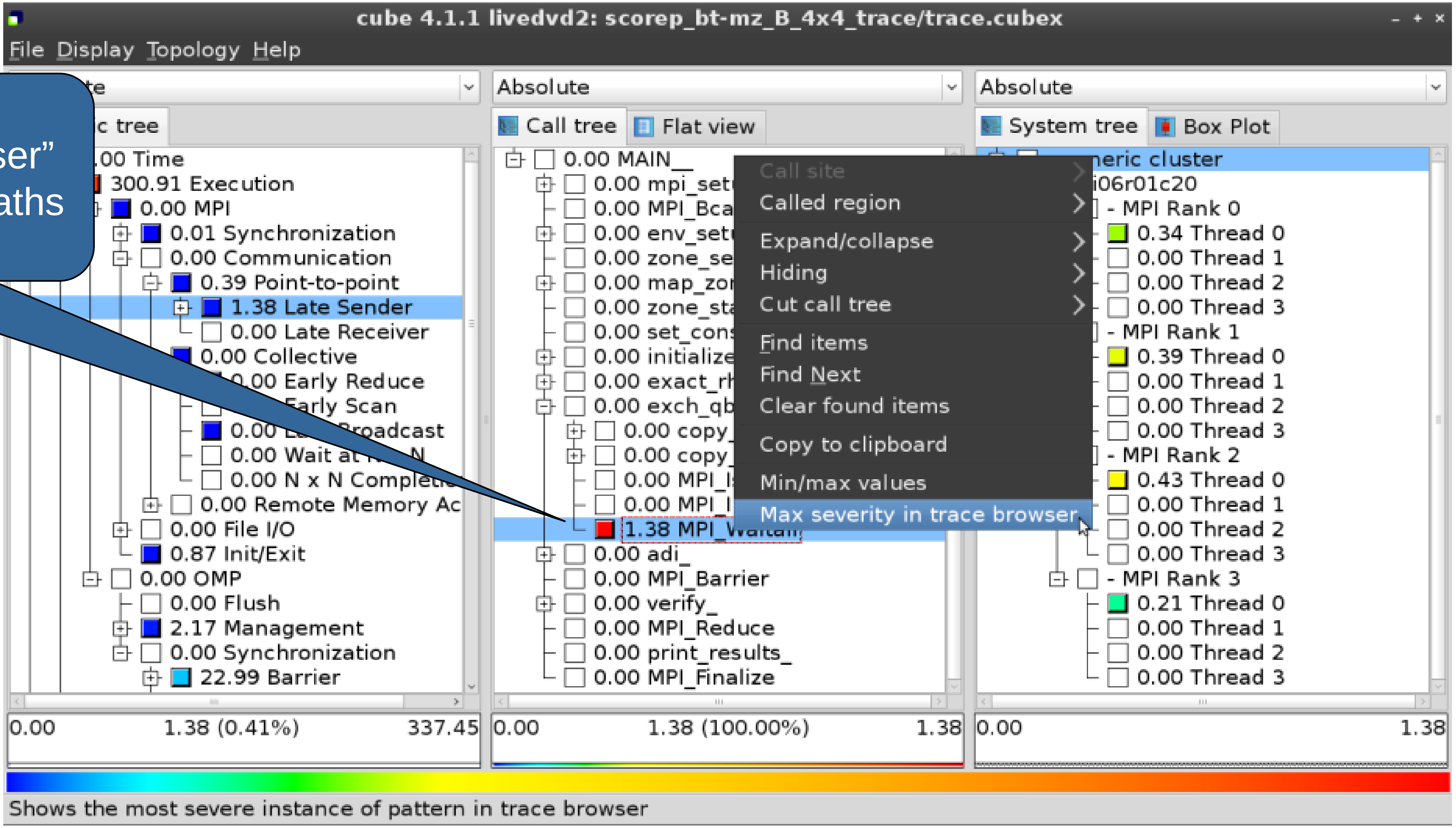
File: c/supermuc_expts/scorep_bt-mz_B_4x4_trace/traces.otf2

Connect to vampir and display a trace file

…and select trace file from the experiment directory

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

```
...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
          ...arly Scan
          0.00 L... Broadcast
          0.00 Wait at ... N
          0.00 N x N Complet...
        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
        22.99 Barrier
```

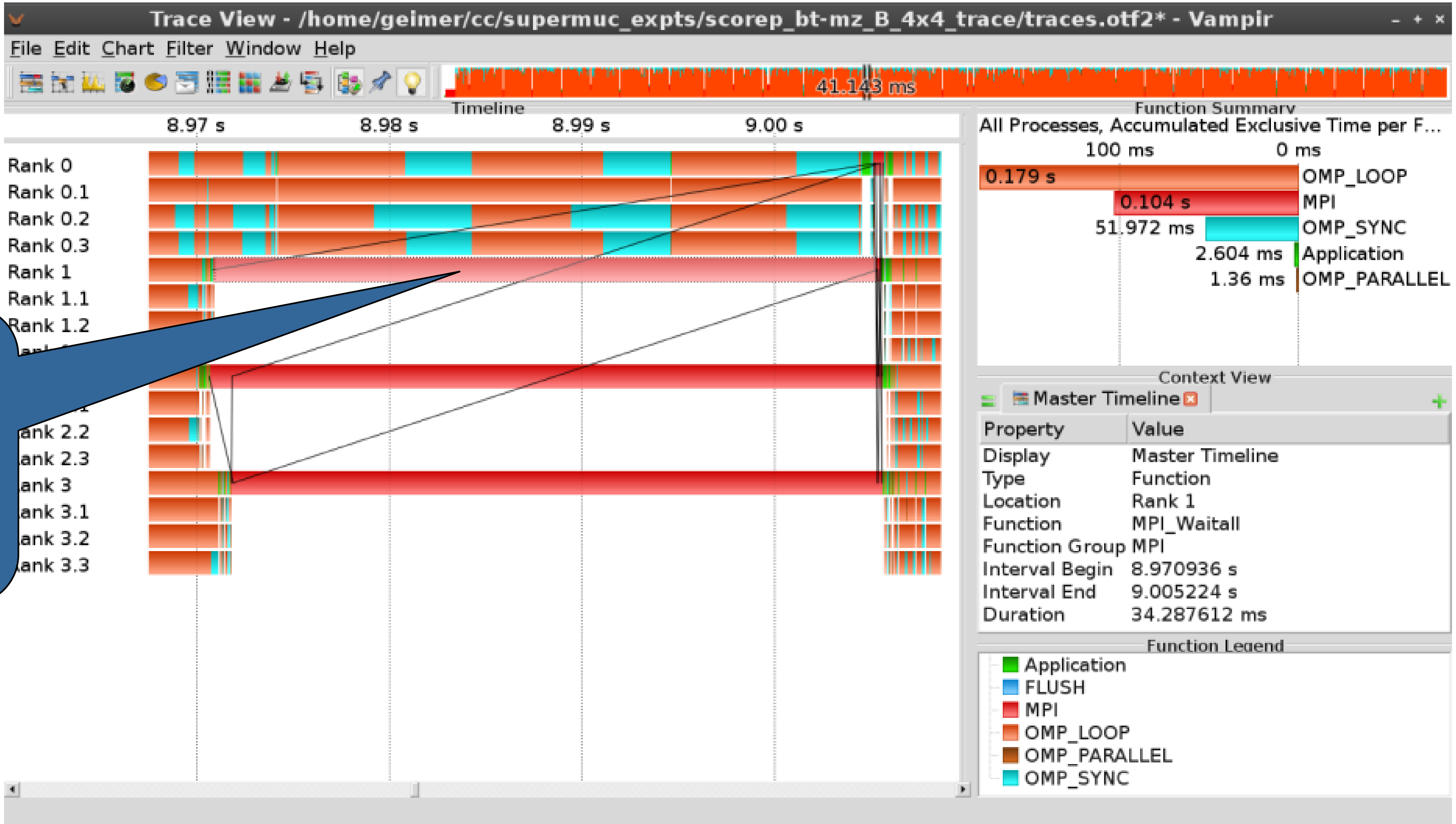
```
0.00 MAIN__
  0.00 mpi_set...
  0.00 MPI_Bca...
  0.00 env_set...
  0.00 zone_se...
  0.00 map_zo...
  0.00 zone_sta...
  0.00 set_cons...
  0.00 initialize...
  0.00 exact_rh...
  0.00 exch_qb...
    0.00 copy...
    0.00 copy...
    0.00 MPI_I...
    0.00 MPI_I...
    1.38 MPI_Waitall
  0.00 adi_
  0.00 MPI_Barrier
  0.00 verify_
  0.00 MPI_Reduce
  0.00 print_results_
  0.00 MPI_Finalize
```

```
...neric 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 (0.41%) 337.45 | 0.00 1.38 (100.00%) 1.38 | 0.00 1.38

Shows the most severe instance of pattern in trace browser

Investigate most severe instance in Vampir



Trace View - /home/geimer/cc/supermuc_expts/scorep_bt-mz_B_4x4_trace/traces.otf2* - Vampir

Function Summary

All Processes, Accumulated Exclusive Time per F...

0.179 s	OMP_LOOP
0.104 s	MPI
51.972 ms	OMP_SYNC
2.604 ms	Application
1.36 ms	OMP_PARALLEL

Context View: Master Timeline

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)

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