Virtual Institute – High Productivity Supercomputing

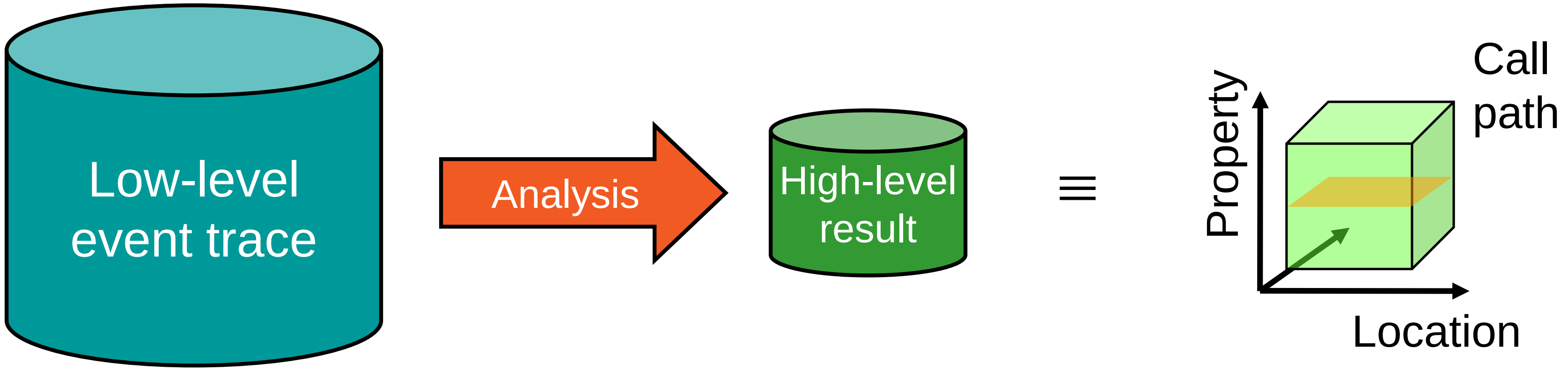
Automatic trace analysis with Scalasca

Ilya Zhukov

Jülich Supercomputing Centre

Automatic trace analysis

- Idea
  - Automatic search for patterns of inefficient behaviour
  - Classification of behaviour & 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: Overview

- Project started in 2006
  - Initial funding by Helmholtz Initiative & Networking Fund
  - Many follow-up projects
- Follow-up to pioneering KOJAK project (started 1998)
  - Automatic pattern-based trace analysis
- Now joint development of
  - Jülich Supercomputing Centre
  - German Research School for Simulation Sciences

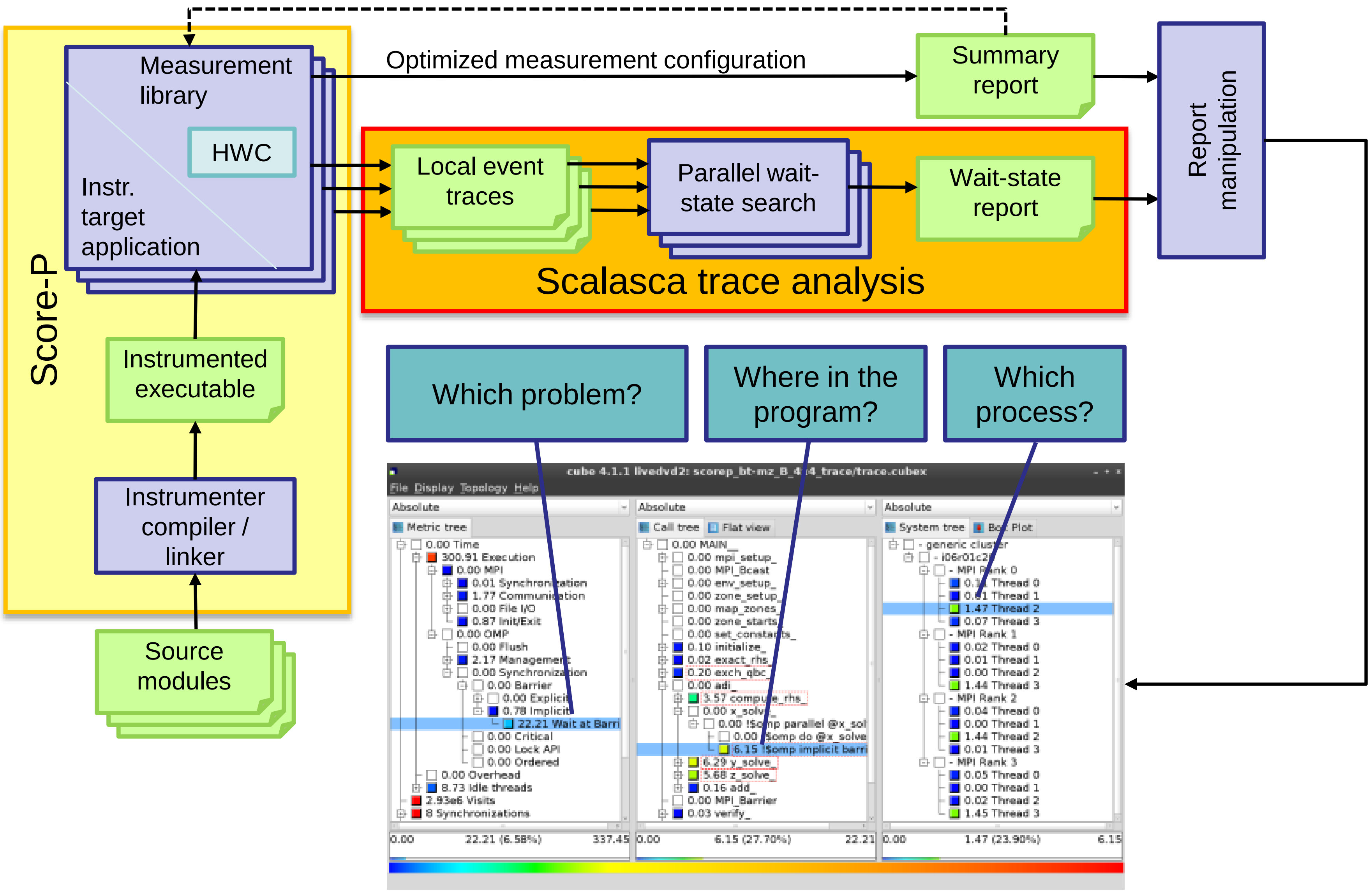
The Scalasca project: Objective

- 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 BlueGene or Cray systems with one million or more processes/threads
- Latest release:
  - Scalasca v2.2 coordinated with Score-P v1.4 (January 2015)
    - *initial support for Intel Xeon Phi (native mode only)*
    - full support for traces in SIONlib format (if configured for OTF2)
    - basic support for POSIX threads and OpenMP tasking
    - added lock contention and root-cause/delay analysis

Scalasca 2.2 features

- Open source, New BSD license
- Fairly portable
  - IBM Blue Gene, Cray XT/XE/XK/XC, SGI Altix, *Fujitsu FX10/100 & K computer*, Linux clusters, Intel Xeon Phi (native MIC) ...
- 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



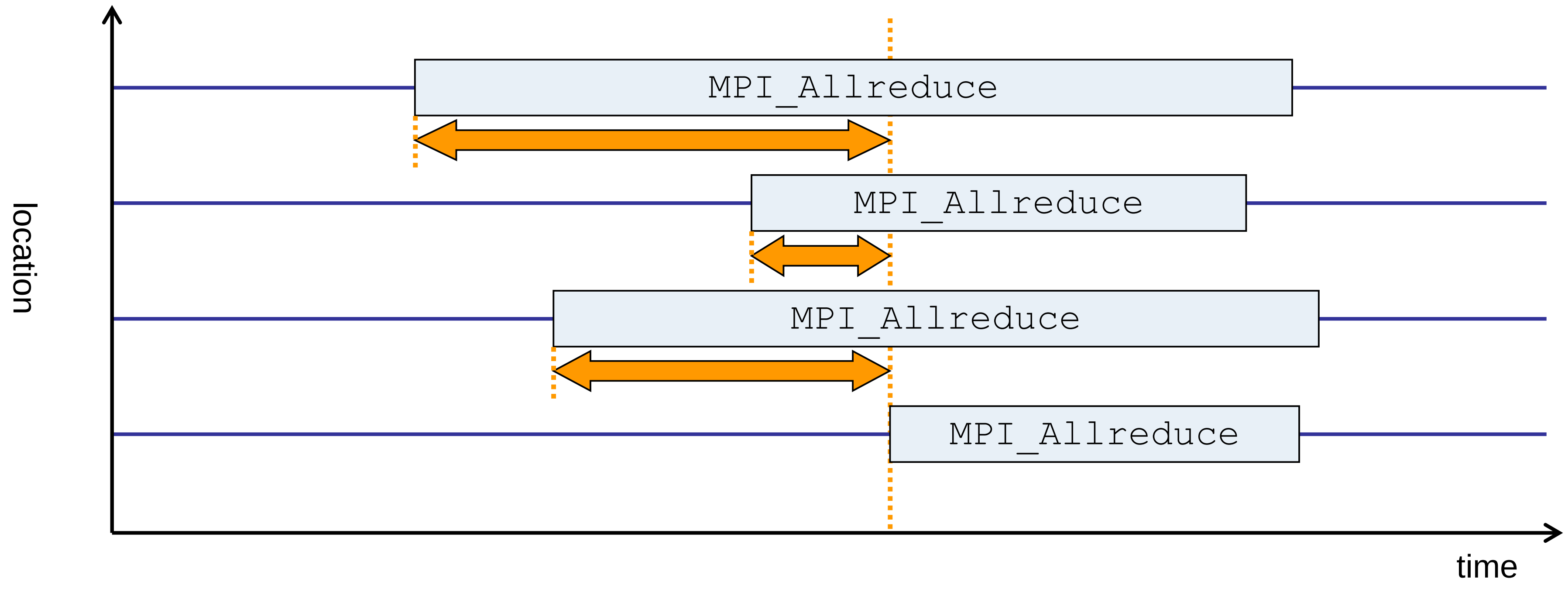
```
cube 4.1.1 livedvd2: scorep_bt-mz_B_4x4_trace/trace.cubex
File  Display  Topology  Help
```

```
Absolute
Metric tree
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
```

```
Absolute
Call tree   Flat view
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
```

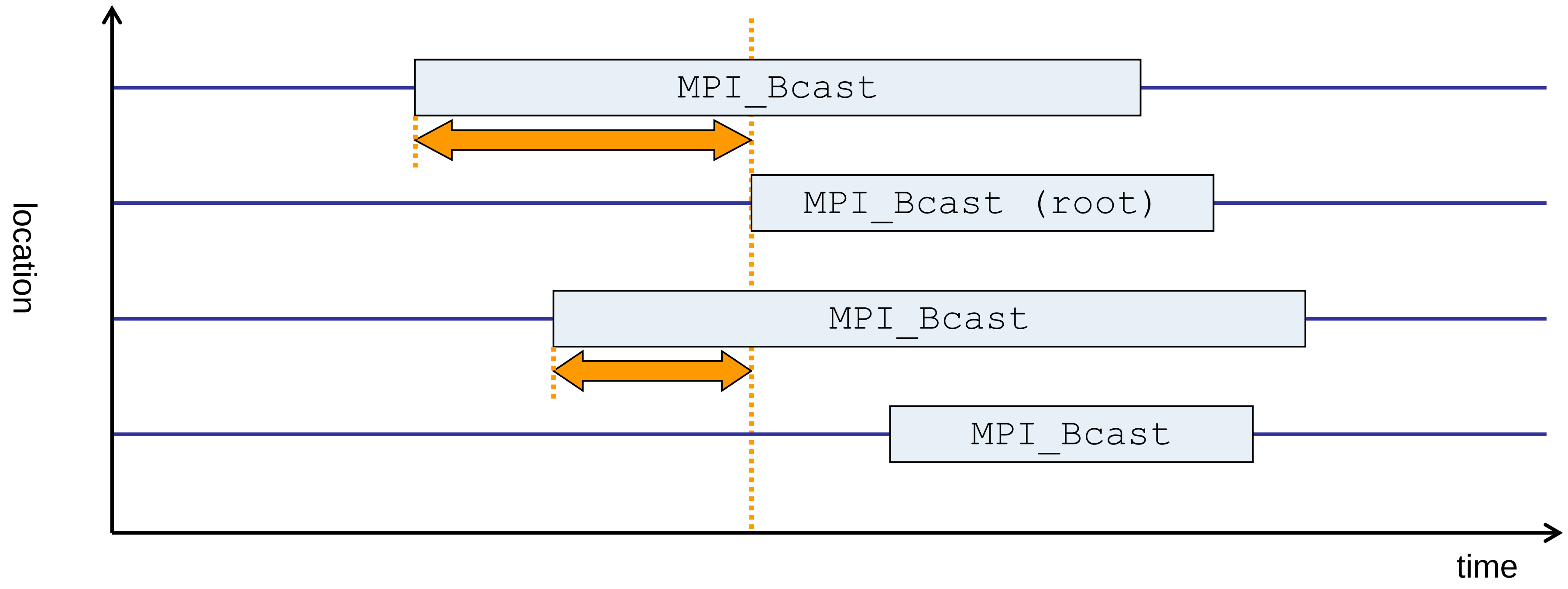
```
Absolute
System tree   Box Plot
- generic cluster
  - i06r01c28
    - MPI Rank 0
      0.11 Thread 0
      0.01 Thread 1
      1.47 Thread 2
      0.07 Thread 3
    - MPI Rank 1
      0.02 Thread 0
      0.01 Thread 1
      0.00 Thread 2
      1.44 Thread 3
    - MPI Rank 2
      0.04 Thread 0
      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: Wait at NxN



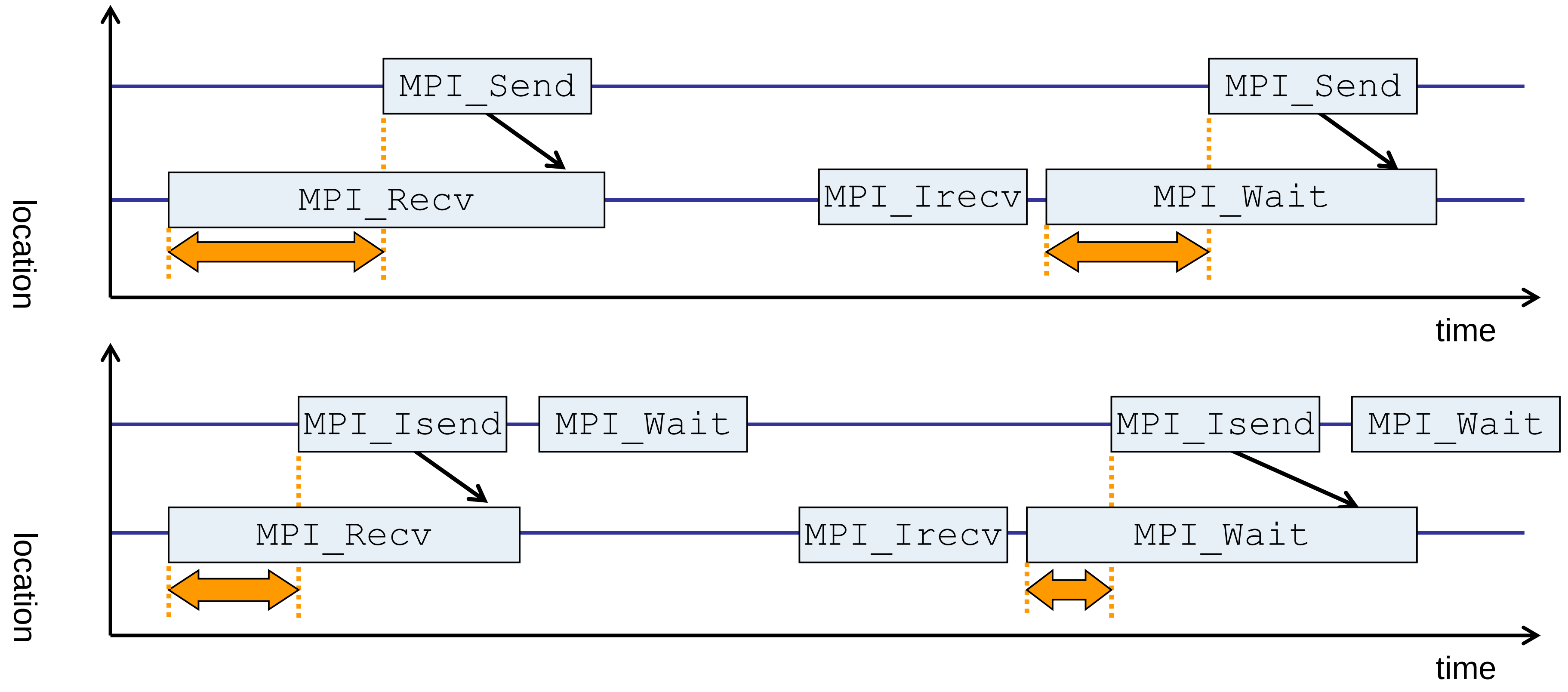
- Time spent waiting in front of synchronizing collective operation until the last process reaches the operation
- Applies to: MPI_Allgather, MPI_Allgatherv, MPI_Alltoall, MPI_Reduce_scatter, MPI_Reduce_scatter_block, MPI_Allreduce

Example: Late Broadcast



- Waiting times if the destination processes of a collective 1-to-N operation enter the operation earlier than the source process (root)
- Applies to: MPI_Bcast, MPI_Scatter, MPI_Scatterv

Example: Late Sender



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

Hands-on: NPB-MZ-MPI / BT

Scalasca command

- One command for (almost) everything…

```
% scalasca
Scalasca 2.2
Toolset for scalable performance analysis of large-scale applications
usage: scalasca [-v][-n][c] {action}
    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

   -v, --verbose      enable verbose commentary
   -n, --dry-run      show actions without taking them
   -c, --show-config  show configuration 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*

- Scalasca application instrumenter

```
% skin
Scalasca 2.2: application instrumenter using scorep
usage: skin [-v] [-comp] [-pdt] [-pomp] [-user] <compile-or-link-cmd>
   -comp={all|none|...}: routines to be instrumented by compiler
          (... custom instrumentation specification for compiler)
   -pdt:  process source files with PDT 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
```

  – Provides compatibility with Scalasca 1.x
  – Recommended: use Score-P instrumenter directly

Scalasca convenience command: *scan*

- Scalasca measurement collection & analysis nexus

```
% scan
Scalasca 2.2: 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.
```

Scalasca advanced command: *scout*

- Scalasca automatic trace analyzer

```
% mpiexec -np 1 scout.hyb --help
SCOUT   Copyright (c) 1998-2014 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
  --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*

- Scalasca analysis report explorer

```
% square
Scalasca 2.2: analysis report explorer
usage: square [-v] [-s] [-f filtfile] [-F] <experiment archive
                                             | cube file>
   -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
```

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

- Load module

```
% module load scalasca/2.2
```

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

BT-MZ summary measurement collection...

- Change to directory with executable and edit job script

```
% cd bin.scorep
% cp ../jobscript/hornet/scalasca2.pbs .
% vi scalasca2.pbs

 [...]

export OMP_NUM_THREADS=6
CLASS=C
NPROCS=8
EXE=./bt-mz_$CLASS.$NPROCS

#export SCOREP_FILTERING_FILE=../config/scorep.filt
#export SCOREP_TOTAL_MEMORY=78M

scalasca -analyze -s  aprun -n $NPROCS -d $OMP_NUM_THREADS  $EXE
```

- Submit the job

```
% qsub scalasca2.pbs
```

BT-MZ summary measurement

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

```
% OMP_NUM_THREADS=6
% scan -f ../config/scorep.filt  mpiexec -np 8 ./bt-mz_C.8
S=C=A=N: Scalasca 2.2 runtime summarization
S=C=A=N: ./scorep_bt-mz_C_8x6_sum experiment archive
S=C=A=N: Thu Sep 13 18:05:17 2012: Collect start
mpiexec -np 8 ./bt-mz_C.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: Thu Sep 13 18:05:39 2012: Collect done (status=0) 22s
S=C=A=N: ./scorep_bt-mz_C_8x6_sum complete.
```

- Creates experiment directory ./scorep_bt-mz_C_8x6_sum

BT-MZ summary analysis report examination

- Score summary analysis report

```
% square -s  scorep_bt-mz_C_8x6_sum
INFO: Post-processing runtime summarization result...
INFO: Score report written to ./scorep_bt-mz_C_8x6_sum/scorep.score
```

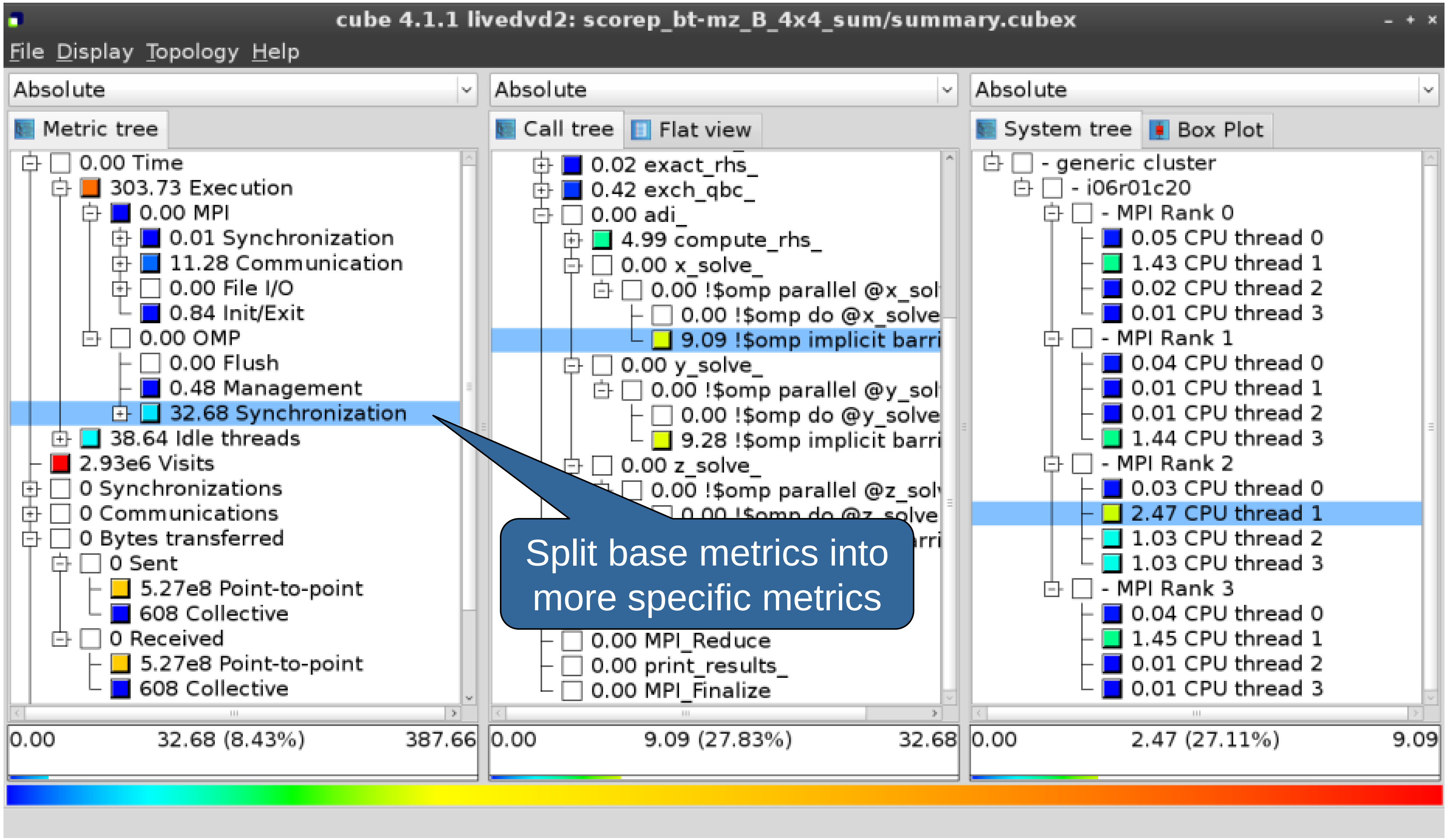
- Post-processing and interactive exploration with CUBE

```
% square  scorep_bt-mz_C_8x6_sum
INFO: Displaying ./scorep_bt-mz_C_8x6_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



cube 4.1.1 livedvd2: scorep_bt-mz_B_4x4_sum/summary.cubex

File Display Topology Help

Absolute | Absolute | Absolute

Metric tree
```
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

Call tree | Flat view
```
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 !$omp parallel @z_solv
      0.00 !$omp do @z_solve
0.00 MPI_Reduce
0.00 print_results_
0.00 MPI_Finalize
```
0.00 9.09 (27.83%) 32.68

System tree | Box Plot
```
- 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

Split base metrics into more specific metrics

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...

- Change to directory with executable and edit job script

```
% cd bin.scorep
% cp ../jobscript/hornet/scalasca2.pbs .
% vi scalasca2.pbs

 [...]

export OMP_NUM_THREADS=6
CLASS=C
NPROCS=8
EXE=./bt-mz_$CLASS.$NPROCS

export SCOREP_FILTERING_FILE=../config/scorep.filt
export SCOREP_TOTAL_MEMORY=78M

scalasca -analyze -t  aprun -n $NPROCS -d $OMP_NUM_THREADS  $EXE
```

- Submit the job

```
% qsub scalasca2.pbs
```

BT-MZ trace analysis

- Automatic trace analysis of existing experiment archives

```
% OMP_NUM_THREADS=6  scan -a  mpiexec -np 8 ./bt-mz_C.8
S=C=A=N: Scalasca 2.2 trace analysis
S=C=A=N: Fri Sep 20 15:09:59 2013: Analyze start
mpiexec -np 8 scout.hyb ./scorep_bt-mz_C_8x6_trace/traces.otf2

Analyzing experiment archive ./scorep_bt-mz_C_8x6_trace/traces.otf2

Opening experiment archive ... done (0.019s).
Reading definition data    ... done (0.178s).
Reading event trace data   ... done (2.068s).
Preprocessing              ... done (3.789s).
Analyzing trace data       ...
  Wait-state detection (fwd)       (1/4) ... done (2.889s).
  Wait-state detection (bwd)       (2/4) ... done (1.136s).
  Synchpoint exchange (fws)        (3/4) ... done (0.813s).
  Critical-path & delay analysis   (4/4) ... done (0.568s).
done (5.413s).
Writing analysis report    ... done (1.994s).

Total processing time: 34.812s
S=C=A=N: Fri Sep 20 15:10:16 2013: Analyze done (status=0) 39s
```

BT-MZ trace measurement ... collection

- Nexus can also combine trace collection and automated analysis using the same execution configuration
- Starts measurement with collection of trace files …

```
% SCOREP_FILTERING_FILE=../config/scorep.filt
% OMP_NUM_THREADS=6  scan -t  mpiexec -np 8 ./bt-mz_C.8
S=C=A=N: Scalasca 2.2 trace collection and analysis
S=C=A=N: Fri Sep 20 15:09:59 2013: Collect start
mpiexec -np 8 ./bt-mz_C.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: Fri Sep 20 15:10:16 2013: Collect done (status=0) 28s
```

BT-MZ trace measurement ... analysis

- Continues with automatic (parallel) analysis of trace files

```
S=C=A=N: Fri Sep 20 15:09:59 2013: Analyze start
mpiexec -np 8 scout.hyb ./scorep_bt-mz_C_8x6_trace/traces.otf2

Analyzing experiment archive ./scorep_bt-mz_C_8x6_trace/traces.otf2

Opening experiment archive ... done (0.019s).
Reading definition data    ... done (0.178s).
Reading event trace data   ... done (2.068s).
Preprocessing              ... done (3.789s).
Analyzing trace data       ...
  Wait-state detection (fwd)      (1/4) ... done (2.889s).
  Wait-state detection (bwd)      (2/4) ... done (1.136s).
  Synchpoint exchange (fws)       (3/4) ... done (0.813s).
  Critical-path & delay analysis  (4/4) ... done (0.568s).
done (5.413s).
Writing analysis report    ... done (1.994s).

*** WARNING ***: 916 clock condition violations detected:
This usually leads to inconsistent analysis results.

Total processing time: 34.812s
S=C=A=N: Fri Sep 20 15:10:16 2013: Analyze done (status=0) 39s
```

BT-MZ trace measurement & time-corrected analysis

- Generating a time-corrected trace and its analysis

```
% SCAN_TRACE_ANALYZER=none  scan -t  mpiexec -np 8 ./bt-mz_C.8
S=C=A=N: Scalasca 2.2 trace collection and analysis
Info: Automatic trace analysis will be skipped!
...
S=C=A=N: Fri Mar 21: 18:00:56 2014: Collect done (status=0) 28s
S=C=A=N: ./scorep_bt-mz_C_8x6_trace complete.
% cd scorep_bt-mz_C_8x6_trace
% mpiexec -np 8 clc_synchronize.hyb ./traces.otf2

        # passes          : 1
        # violated        : 3362
        # corrected       : 1610977
        # reversed-p2p    : 233
        # reversed-coll   : 0
        # reversed-omp    : 3129
        # events          : 6287852
        max. error        : 0.000112 [s]
        error at final.   : 0.000118 [%]
        Max slope         : 0.010000000
% scan -a -e ./clc_sync  mpiexec -np 8 ../bt-mz_C.8
S=C=A=N: Scalasca 2.2 trace analysis
...
S=C=A=N: Fri Mar 21 18:29:29 2014: Analyze done (status=0) 39s
S=C=A=N: ./clc_sync complete
```

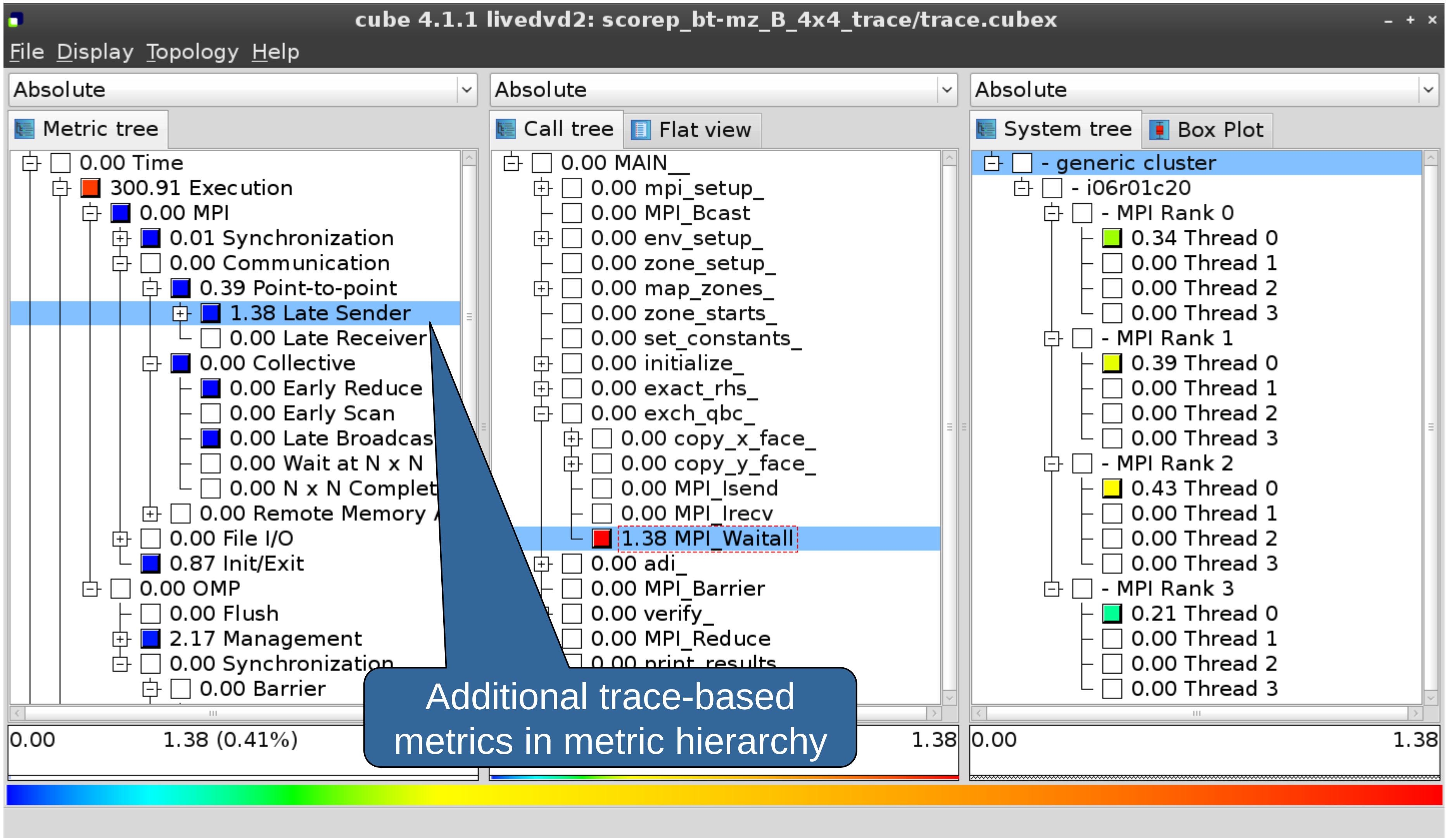
BT-MZ trace analysis report exploration

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

```
% square  scorep_bt-mz_C_8x6_trace
INFO: Post-processing runtime summarization result...
INFO: Post-processing trace analysis report...
INFO: Displaying ./scorep_bt-mz_C_8x6_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

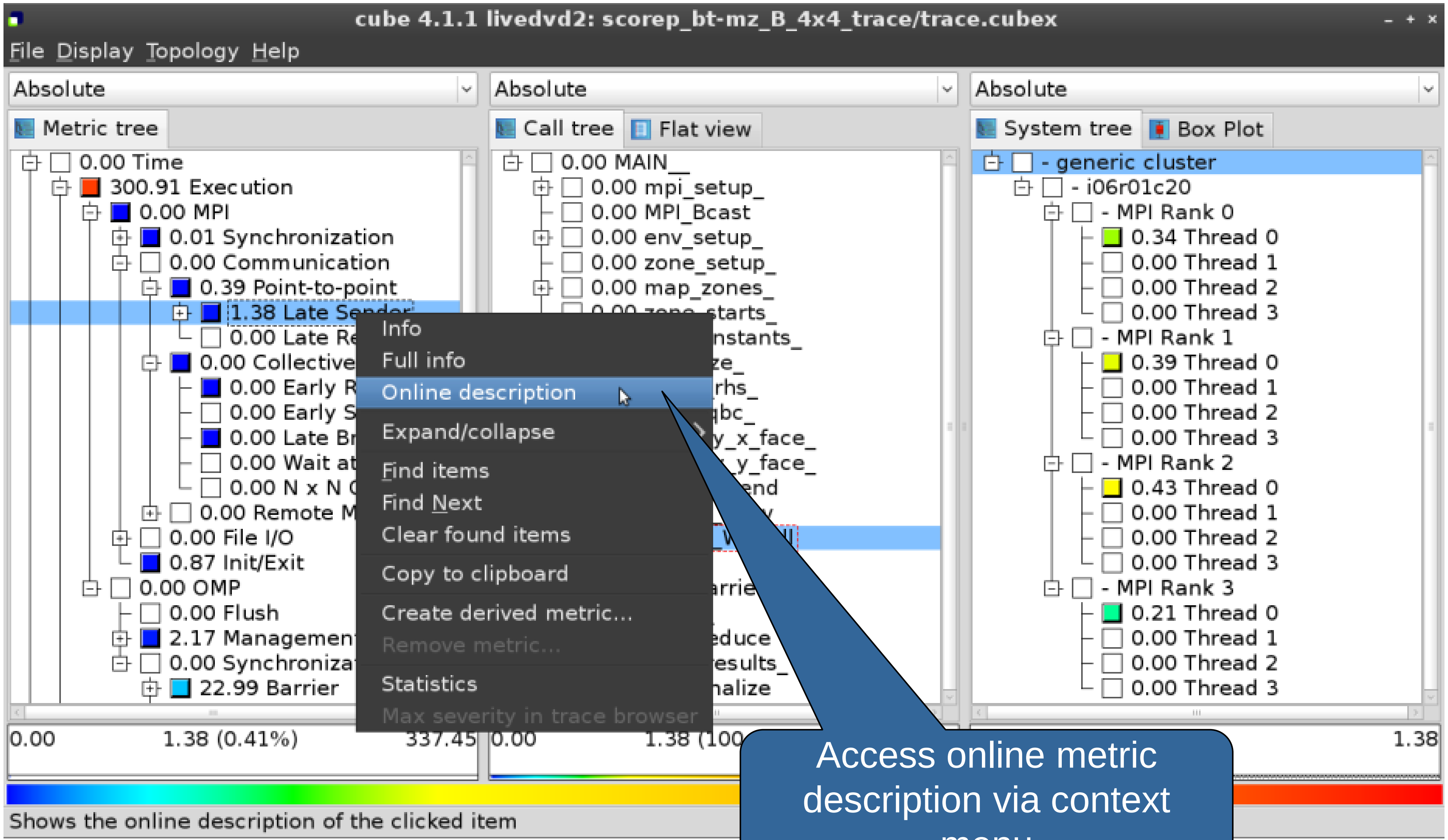
File Display Topology Help

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 Broadcas	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 Complet	0.00 MPI_Isend	0.43 Thread 0
0.00 Remote Memory	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 Thread 3

0.00 1.38 (0.41%) 1.38 0.00 1.38

Additional trace-based metrics in metric hierarchy

Online metric description



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
      0.00 Communication
        0.39 Point-to-point
          1.38 Late Sender
          0.00 Late Re
        0.00 Collective
          0.00 Early R
          0.00 Early S
          0.00 Late Br
          0.00 Wait at
          0.00 N x N C
        0.00 Remote M
      0.00 File I/O
      0.87 Init/Exit
    0.00 OMP
      0.00 Flush
      2.17 Managemen
      0.00 Synchroniza
        22.99 Barrier
```

```
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_
```

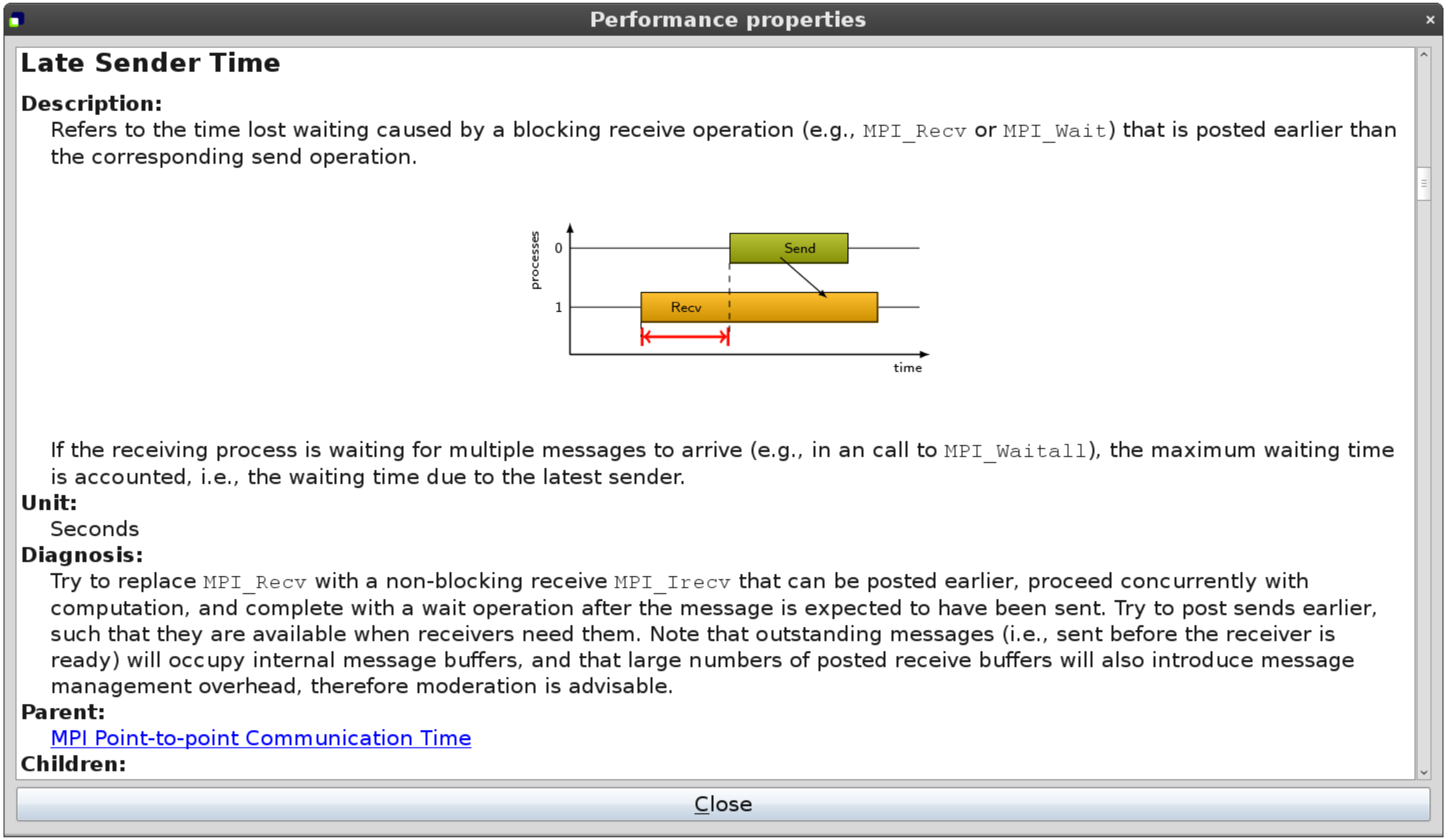
```
- generic cluster
  - i06r01c20
    - MPI Rank 0
      0.34 Thread 0
      0.00 Thread 1
      0.00 Thread 2
      0.00 Thread 3
    - MPI Rank 1
      0.39 Thread 0
      0.00 Thread 1
      0.00 Thread 2
      0.00 Thread 3
    - MPI Rank 2
      0.43 Thread 0
      0.00 Thread 1
      0.00 Thread 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 (0.41%) 337.45 | 0.00 1.38 (100 | 1.38

Shows the online description of the clicked item

Access online metric description via context menu

Online metric description

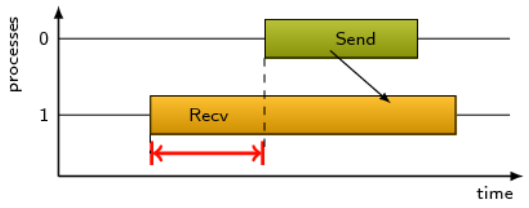


Performance properties

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:

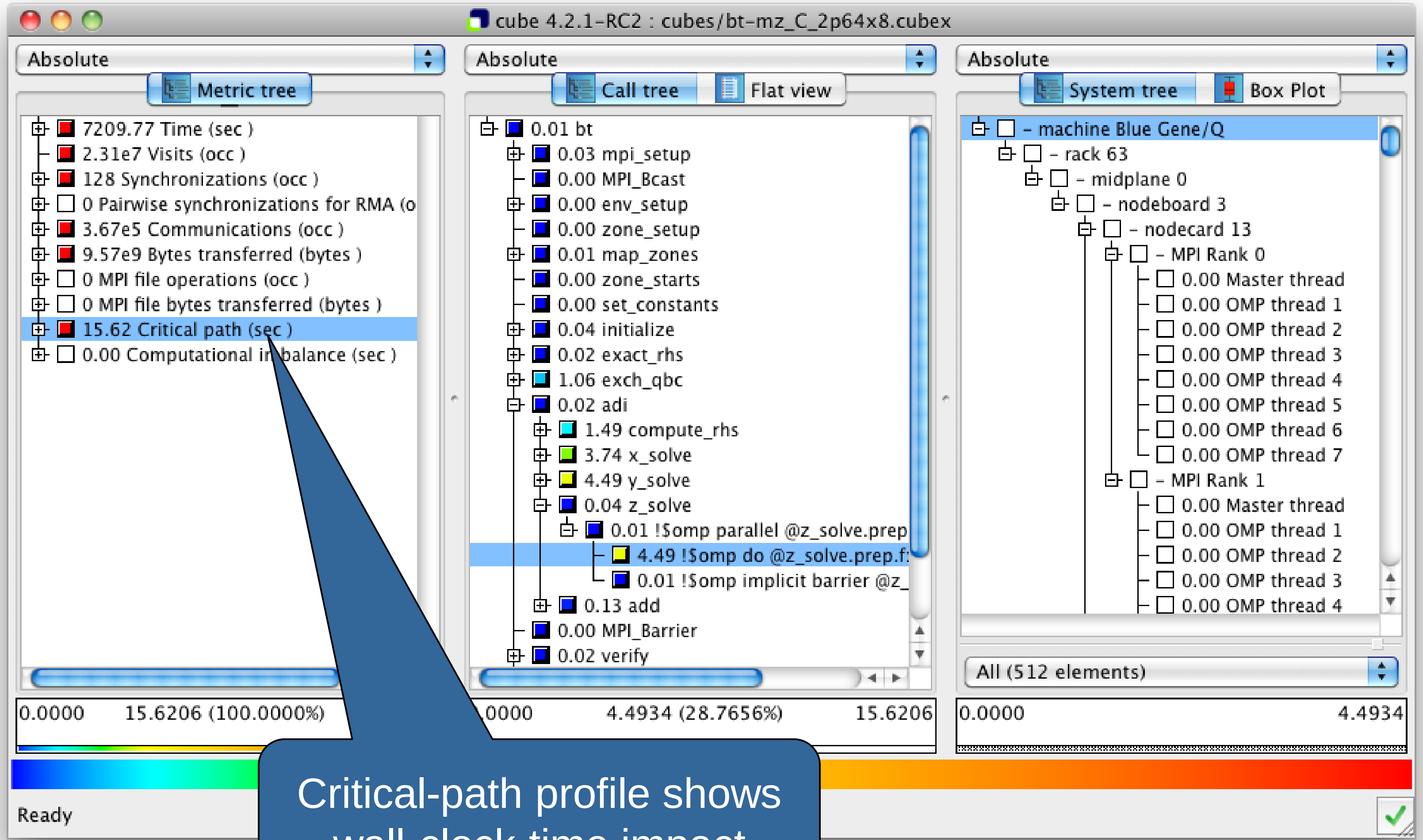
Try to replace `MPI_Recv` with a non-blocking receive `MPI_Irecv` that can be posted earlier, proceed concurrently with computation, and complete with a wait operation after the message is expected to have been sent. Try to post sends earlier, such that they are available when receivers need them. Note 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



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

Absolute | Absolute | 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 )
```

Call tree:
```
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
```

System tree:
```
- 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 15.6206 (100.0000%)

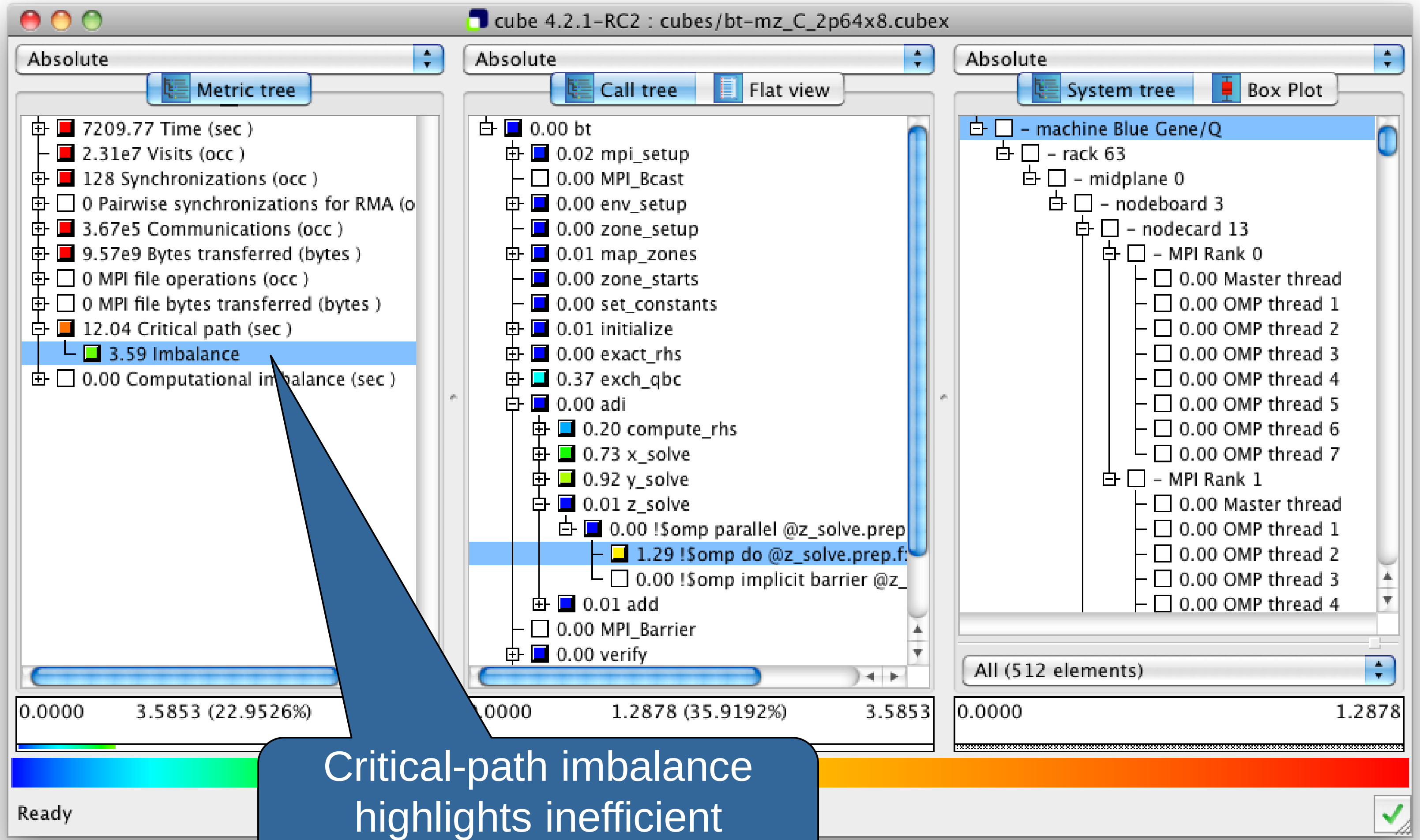
.0000 4.4934 (28.7656%) 15.6206

0.0000 4.4934

Ready

Critical-path profile shows wall-clock time impact

Critical-path analysis



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

Absolute | Absolute | 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)

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

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

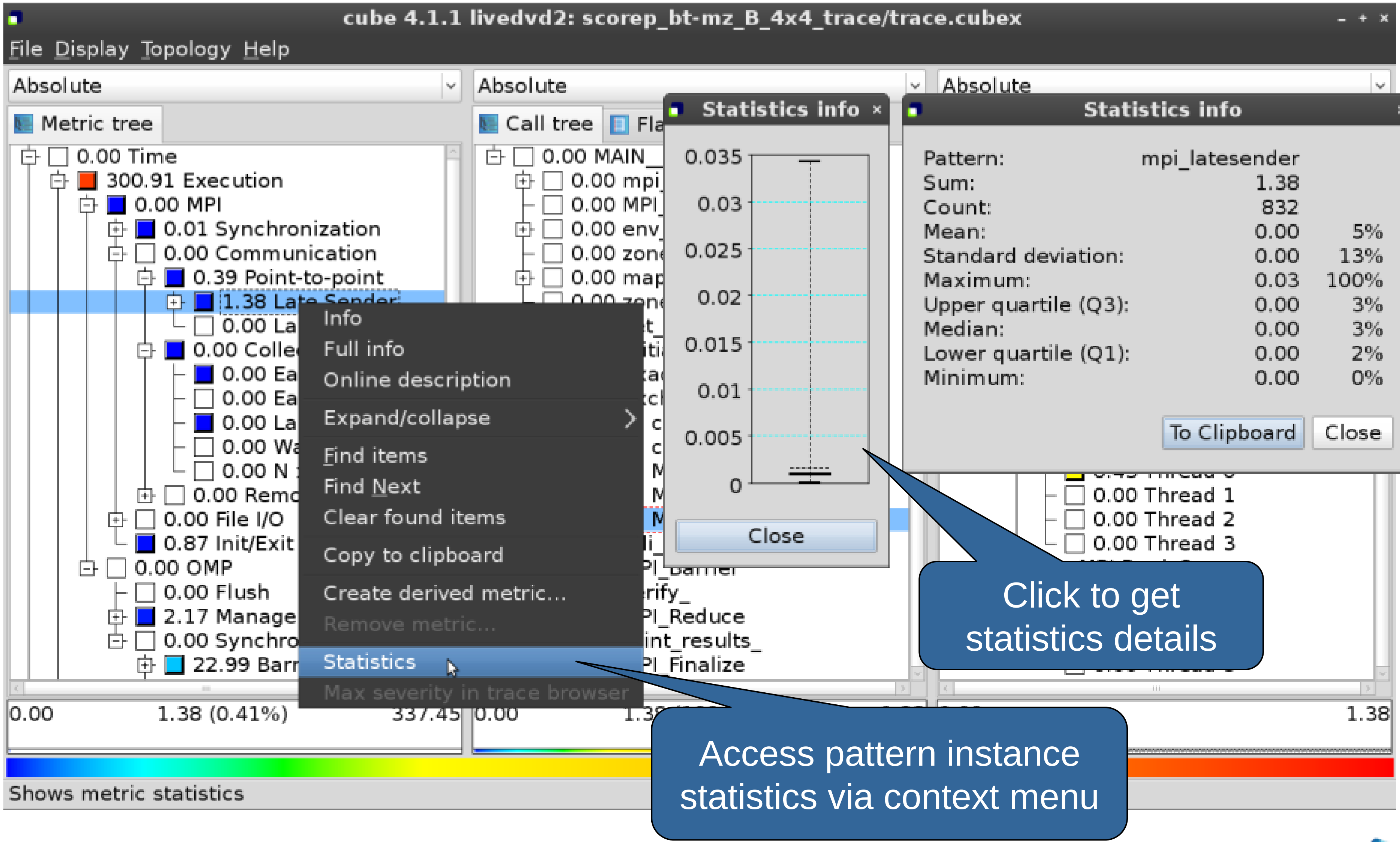
All (512 elements)

0.0000 3.5853 (22.9526%) | .0000 1.2878 (35.9192%) 3.5853 | 0.0000 1.2878

Ready

Critical-path imbalance highlights inefficient parallelism

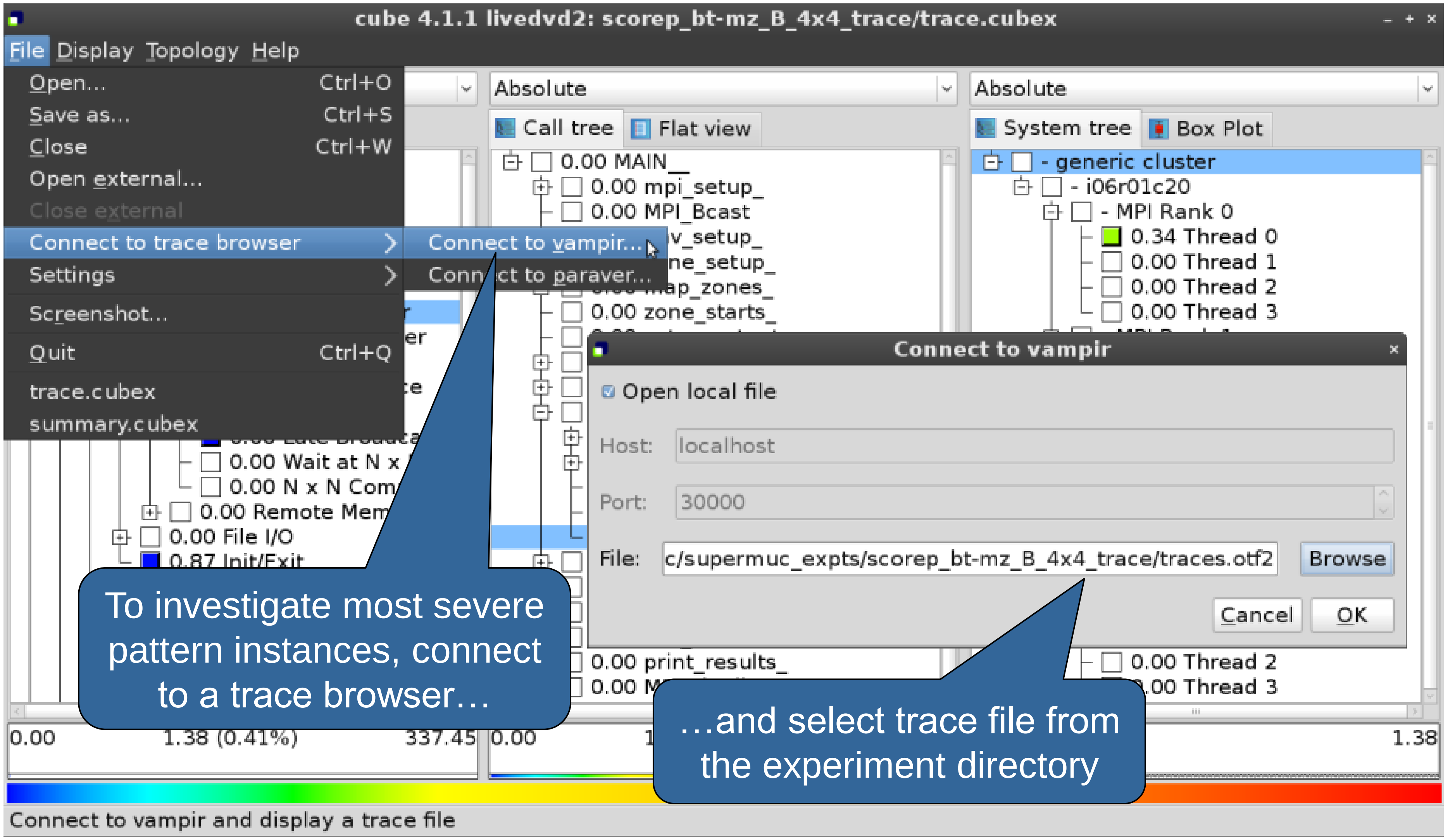
Pattern instance statistics



```
0.00 Time
  300.91 Execution
    0.00 MPI
      0.01 Synchronization
      0.00 Communication
        0.39 Point-to-point
          1.38 Late Sender
```

```
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%
```

Connect to Vampir trace browser



Connect to vampir

☑ Open local file

Host: localhost

Port: 30000

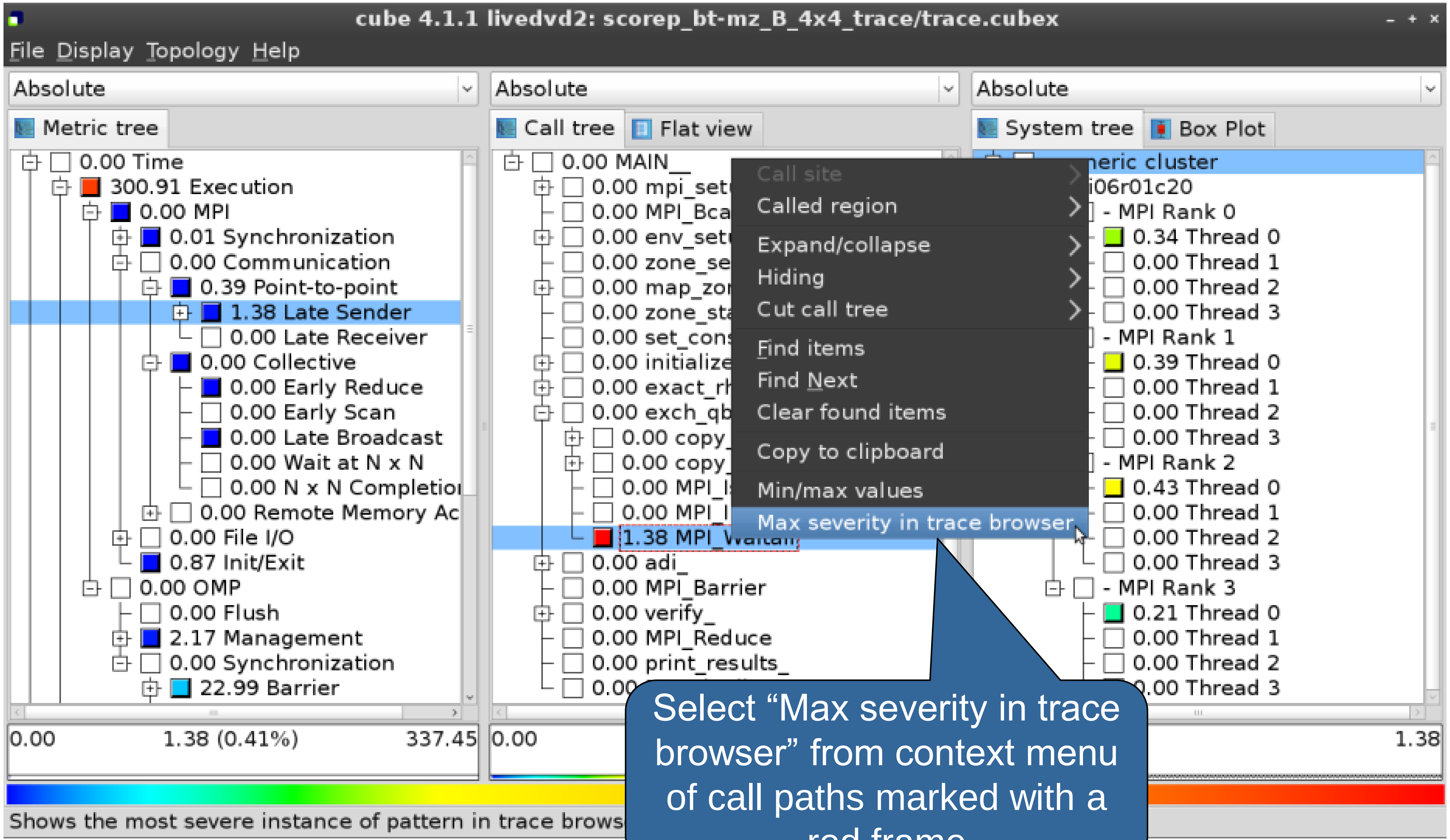
File: c/supermuc_expts/scorep_bt-mz_B_4x4_trace/traces.otf2

Browse | Cancel | OK

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

…and select trace file from the experiment directory

Show most severe pattern instances



```
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
        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_zor
  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
```

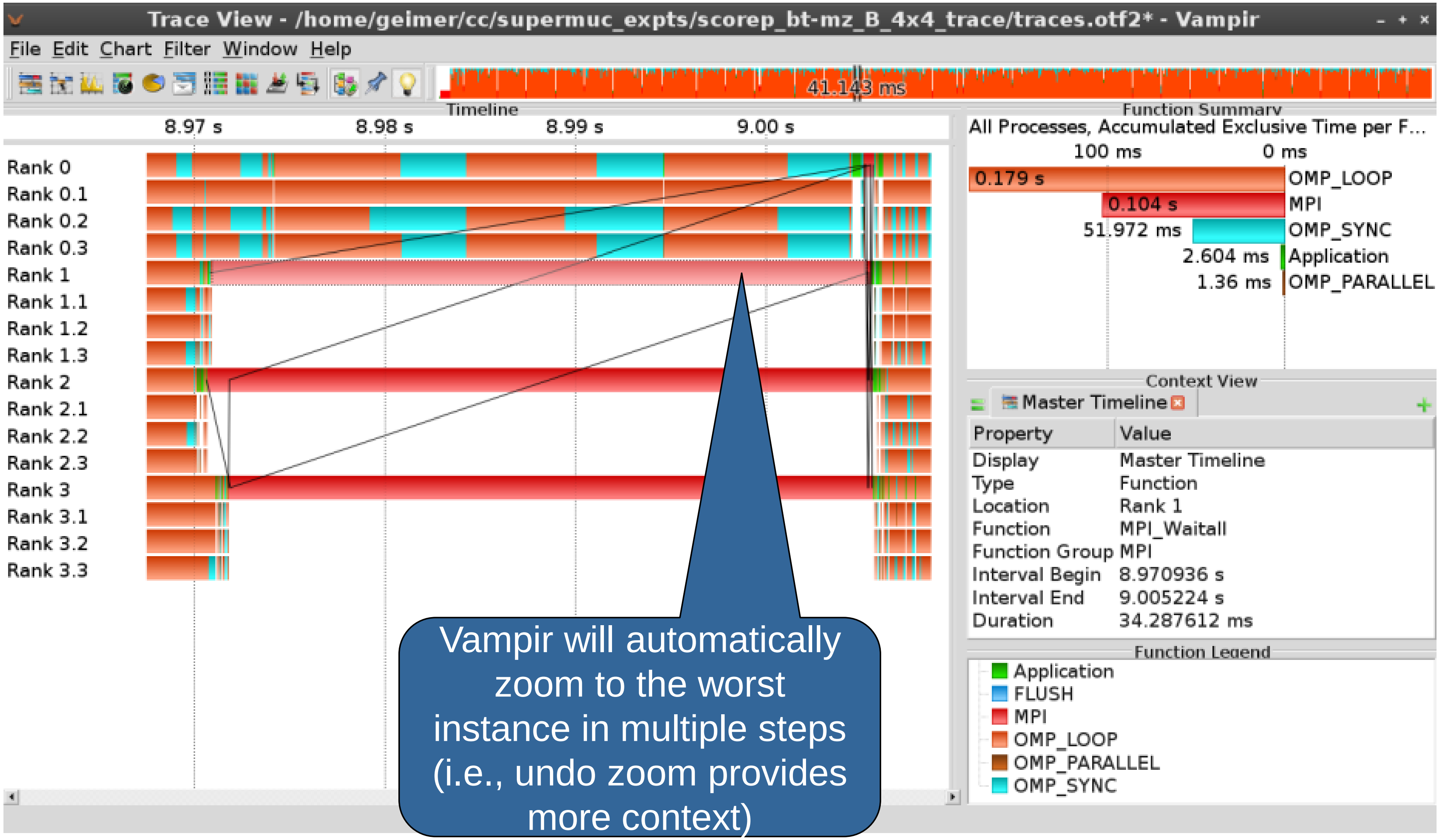
```
eric cluster
i06r01c20
  - MPI Rank 0
    0.34 Thread 0
    0.00 Thread 1
    0.00 Thread 2
    0.00 Thread 3
  - MPI Rank 1
    0.39 Thread 0
    0.00 Thread 1
    0.00 Thread 2
    0.00 Thread 3
  - MPI Rank 2
    0.43 Thread 0
    0.00 Thread 1
    0.00 Thread 2
    0.00 Thread 3
  - MPI Rank 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

Shows the most severe instance of pattern in trace brows

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

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

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-**sca**le parallel **a**pplications

– toolset for scalable performance measurement & analysis of MPI, OpenMP & hybrid parallel applications
– supporting most popular HPC computer systems
– available under New BSD open-source license
– sources, documentation & publications:
  - http://www.scalasca.org
  - mailto: scalasca@fz-juelich.de