

# VI-HPS

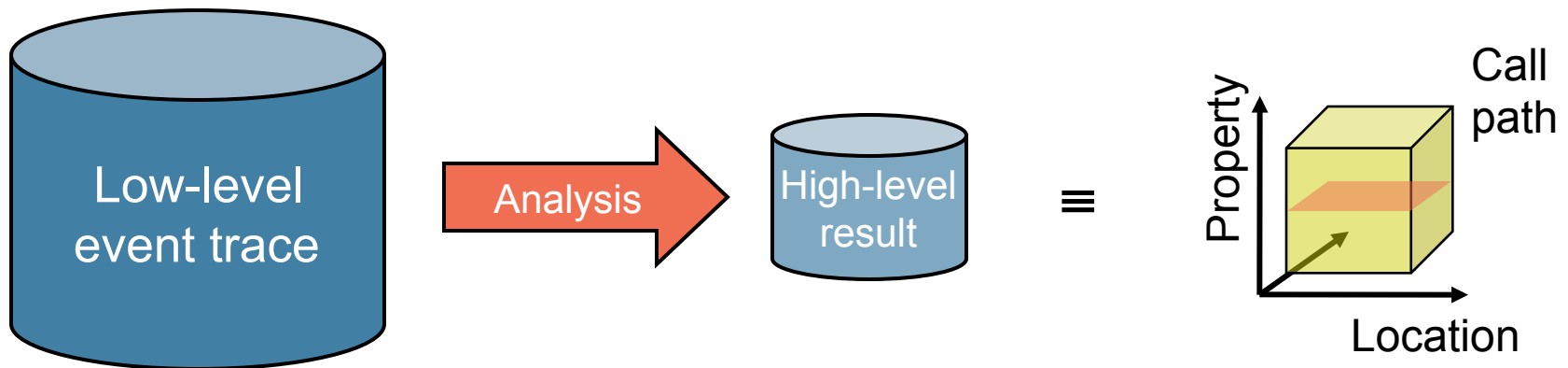


## Automatic trace analysis with Scalasca

Marc-André Hermanns  
JARA-HPC, RWTH Aachen University

scalasca

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

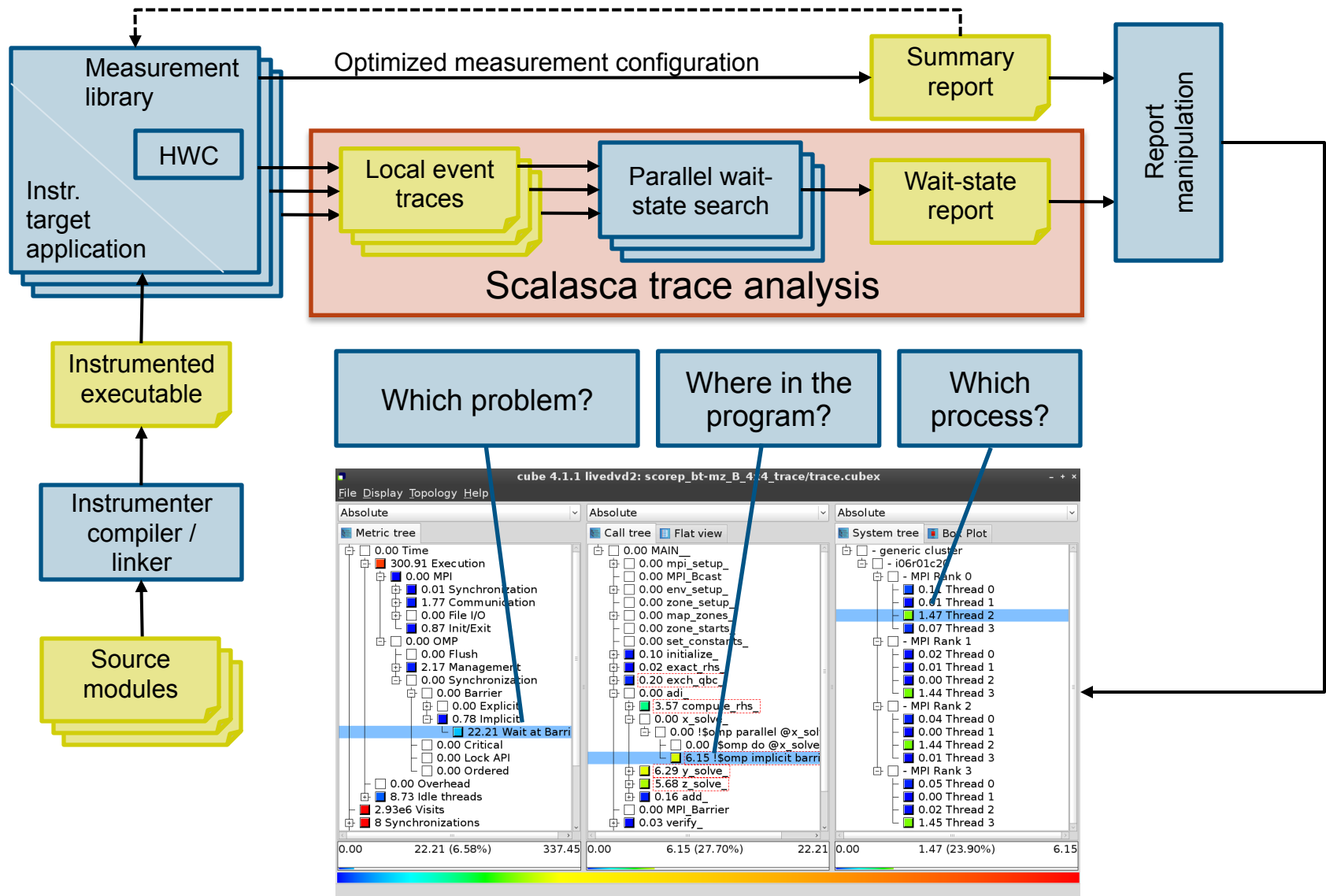


- Guaranteed to cover the entire event trace
- Quicker than manual/visual trace analysis
- Parallel replay analysis exploits available memory & processors to deliver scalability

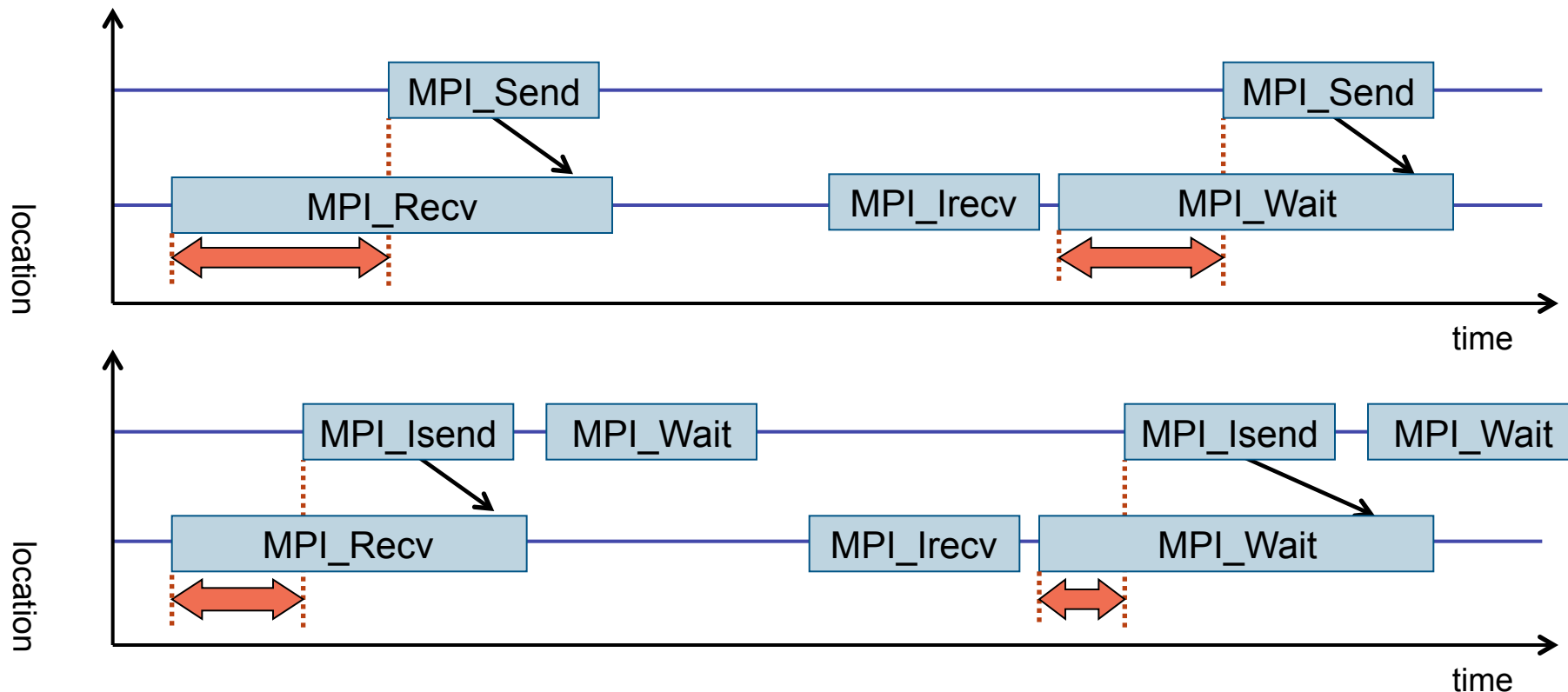
- Project started in 2006
  - Follow-up to pioneering KOJAK project (started 1998)
- Joint development of
  - Jülich Supercomputing Centre
  - German Research School for Simulation Sciences
- Development of a **scalable** performance analysis toolset for most popular parallel programming paradigms
- Specifically targeting **large-scale** parallel applications
  - such as those running on IBM BlueGene or Cray XT systems with one million or more processes/threads
- Latest release:
  - Scalasca v2.1 (August 2014)



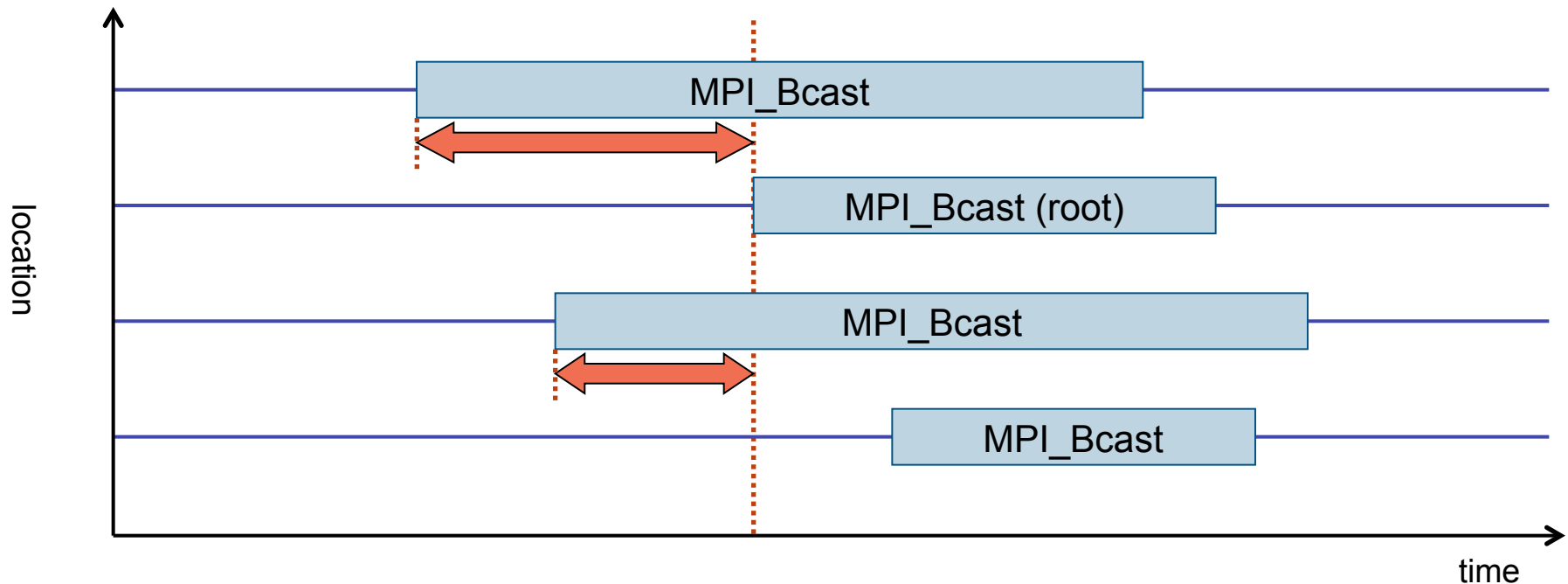
- Open source, BSD 3-clause license
- Fairly portable
  - IBM Blue Gene, IBM SP & blade clusters, Cray XT/XE/XK/XC, SGI Altix, Solaris & Linux clusters, Fujitsu FX10 & K computer, ...
- Uses Score-P instrumenter & measurement libraries
  - Scalasca 2.1 core package focuses on trace-based analyses
  - Supports common data formats
    - Reads event traces in OTF2 format
    - Writes analysis reports in CUBE4 format
- Current limitations:
  - No support for nested OpenMP parallelism and tasking
  - Unable to handle OTF2 traces containing CUDA events



## 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



- 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`



# VI-HPS



## Scalasca Demo: NPB-MZ-MPI / BT

scalasca 



- One command for (almost) everything...

```
% scalasca
Scalasca 2.1
Toolset for scalable performance analysis of large-scale 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

-c, --show-config  show configuration and exit
-h, --help        show this help and exit
-n, --dry-run     show actions without taking them
    --quickref    show quick reference guide and exit
-v, --verbose     enable verbose commentary
-V, --version     show version information and exit
```

- Scalasca application instrumenter

```
% skin
Scalasca 2.1: 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
```

- Deprecated command
  - Provides compatibility with Scalasca 1.x
  - Prints corresponding Score-P instrumenter command
  - Helps in transitioning existing configurations
- Recommended: use Score-P instrumenter directly

- Scalasca measurement collection & analysis nexus

```
% scan
Scalasca 2.1: 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 filtfle : File specifying measurement filter.
    -l lockfile : File that blocks start of measurement.
```

- Scalasca analysis report explorer

```
% square
Scalasca 2.1: analysis report explorer
usage: square [-v] [-s] [-f filtfiler] [-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 filtfiler        : Use specified filter file when doing scoring
  -s                  : Skip display and output textual score report
  -v                  : Enable verbose mode
```

- **scan** configures Score-P measurement by setting some environment variables automatically
  - 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)

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

```
% export SCOREP_EXPERIMENT_DIRECTORY=scorep_bt-mz_W_4x4_sum
% OMP_NUM_THREADS=4 scan mpiexec -np 4 ./bt-mz_W.4
S=C=A=N: Scalasca 2.1 runtime summarization
S=C=A=N: ./scorep_bt-mz_W_4x4_sum experiment archive
S=C=A=N: Thu Jun 12 18:05:17 2014: Collect start
mpiexec -np 4 ./bt-mz_W.4

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:      4

[... More application output ...]

S=C=A=N: Thu Jun 12 18:05:39 2014: Collect done (status=0) 22s
S=C=A=N: ./scorep_bt-mz_W_4x4_sum complete.
```

- Creates experiment directory `./scorep_bt-mz_W_4x4_sum`

- Score summary analysis report

```
% square -s scorep_bt-mz_W_4x4_sum  
INFO: Post-processing runtime summarization result...  
INFO: Score report written to ./scorep_bt-mz_W_4x4_sum/scorep.score
```

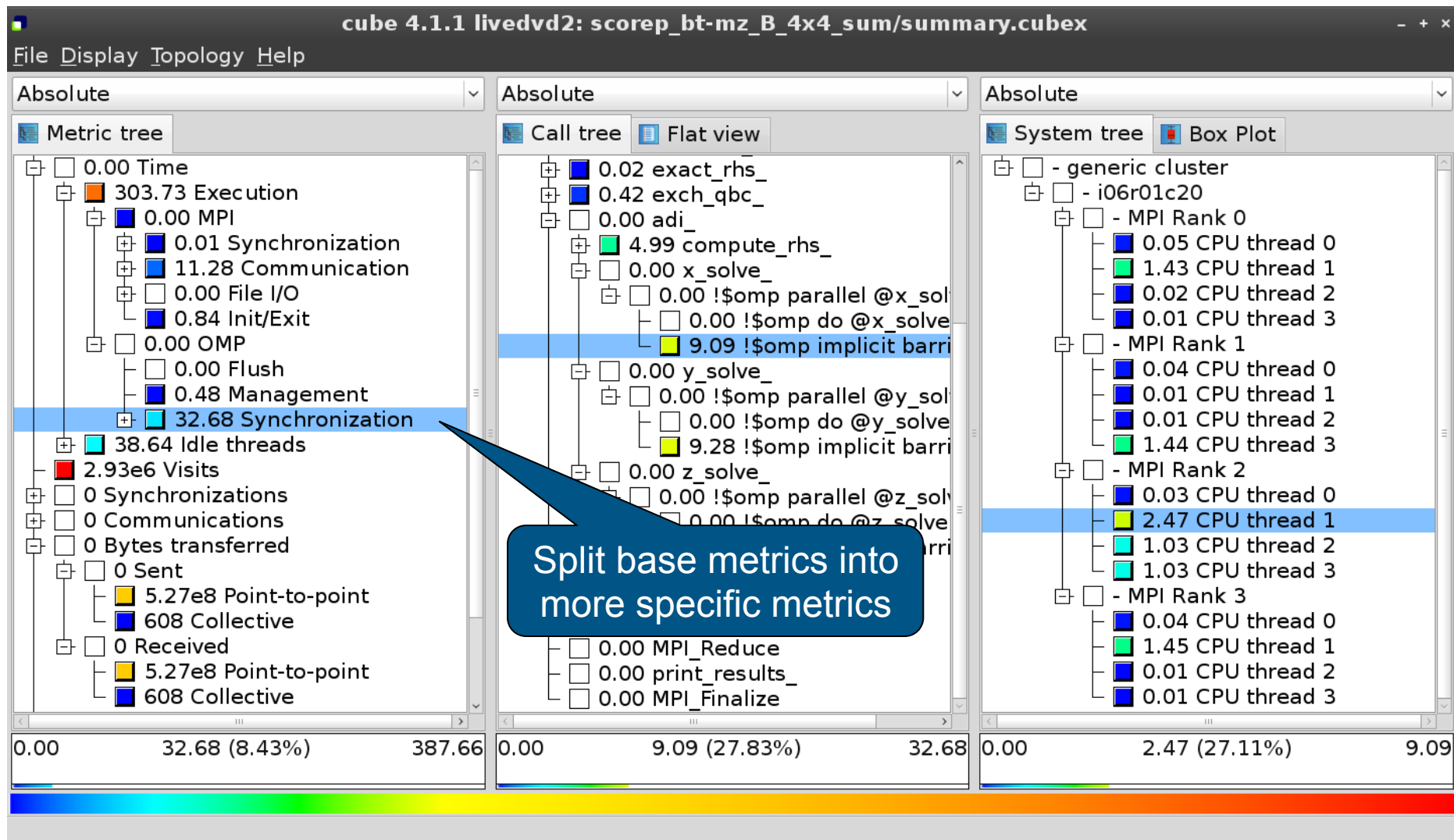
- Post-processing and interactive exploration with CUBE

```
% square scorep_bt-mz_W_4x4_sum  
INFO: Displaying ./scorep_bt-mz_W_4x4_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



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**

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

```
% export SCOREP_EXPERIMENT_DIRECTORY=scorep_bt-mz_W_4x4_trace
% OMP_NUM_THREADS=4 scan -t mpiexec -np 4 ./bt-mz_W.4
S=C=A=N: Scalasca 2.1 trace collection and analysis
S=C=A=N: ./scorep_bt-mz_W_4x4_trace experiment archive
S=C=A=N: Thu Jun 12 18:05:39 2014: Collect start
mpiexec -np 4 ./bt-mz_B.4
  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:      4

      [... More application output ...]

S=C=A=N: Thu Jun 12 18:05:58 2014: Collect done (status=0) 19s
      [... continued ...]
```

- Continues with automatic (parallel) analysis of trace files

```
S=C=A=N: Thu Jun 12 18:05:58 2014: Analyze start
mpiexec -np 4 scout.hyb ./scorep_bt-mz_W_4x4_trace/traces.otf2
SCOUT    Copyright (c) 1998-2012 Forschungszentrum Juelich GmbH
          Copyright (c) 2009-2012 German Research School for Simulation
          Sciences GmbH

Analyzing experiment archive ./scorep_bt-mz_W_4x4_trace/traces.otf2

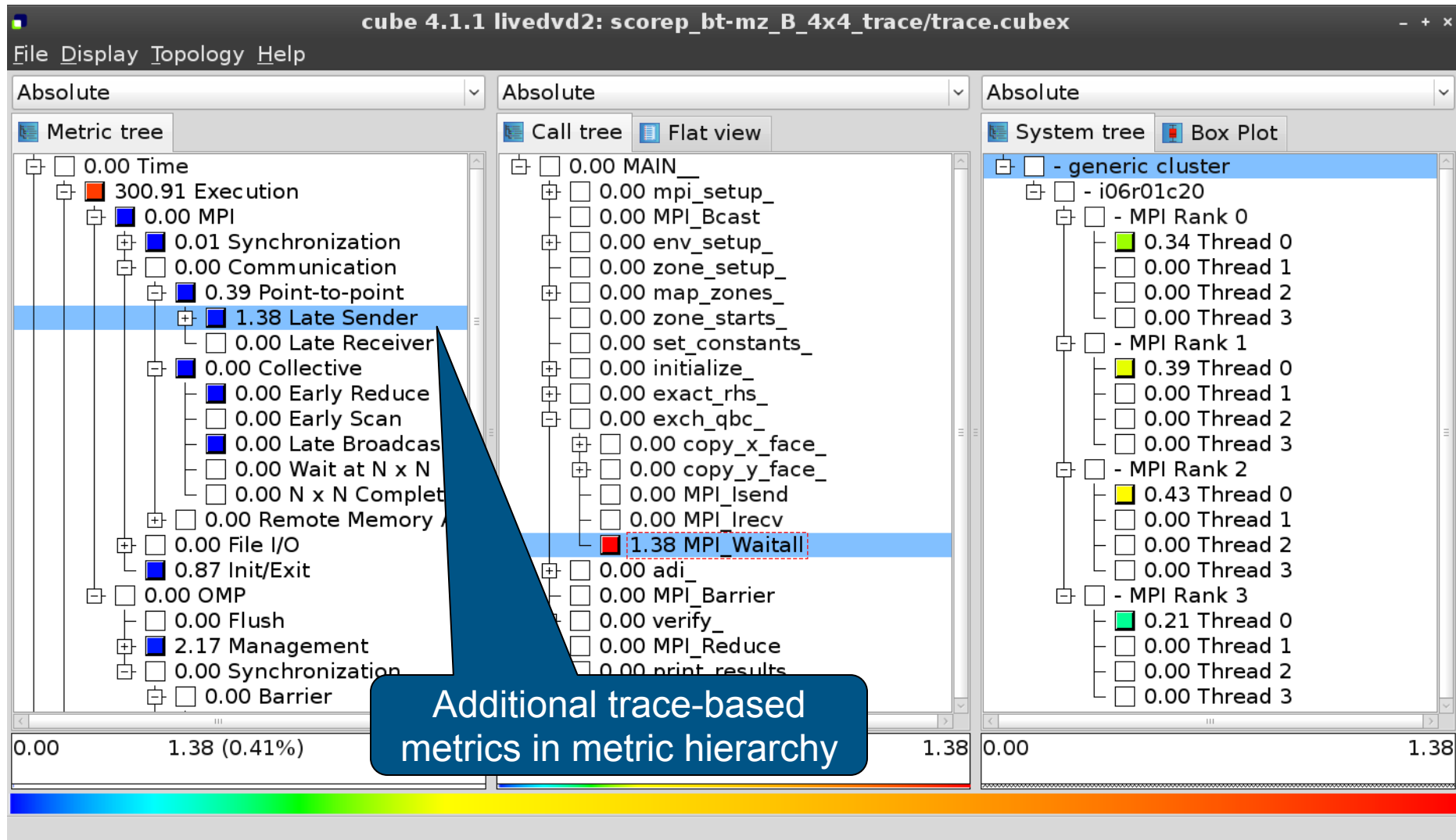
Opening experiment archive ... done (0.002s).
Reading definition data      ... done (0.004s).
Reading event trace data    ... done (0.130s).
Preprocessing                ... done (0.259s).
Analyzing trace data        ...
  Wait-state detection (fwd)  (1/4) ... done (0.575s).
  Wait-state detection (bwd)  (2/4) ... done (0.138s).
  Synchpoint exchange        (3/4) ... done (0.358s).
  Critical-path analysis      (4/4) ... done (0.288s).
done (1.360s).
Writing analysis report      ... done (0.121s).

Total processing time       : 1.924s
S=C=A=N: Thu Jun 12 18:06:00 2014: Analyze done (status=0) 2s
```

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

```
% square scorep_bt-mz_W_4x4_trace  
INFO: Post-processing runtime summarization result...  
INFO: Post-processing trace analysis report...  
INFO: Displaying ./scorep_bt-mz_W_4x4_trace/trace.cubex...  
  
[GUI showing trace analysis report]
```

# Post-processed trace analysis report



The screenshot displays the cube 4.1.1 interface with the title bar "cube 4.1.1 livedvd2: scorep\_bt-mz\_B\_4x4\_trace/trace.cubex". The interface is divided into three main panels: "Metric tree", "Call tree", and "System tree".

- Metric tree:** Shows a hierarchical view of metrics. The "Late Sender" metric (1.38) is selected, and a context menu is open over it. The menu options are: Info, Full info, Online description (highlighted), Expand/collapse, Find items, Find Next, Clear found items, Copy to clipboard, Create derived metric..., Remove metric..., Statistics, and Max severity in trace browser.
- Call tree:** Shows a hierarchical view of calls. The "MAIN\_" call (0.00) is selected.
- System tree:** Shows a hierarchical view of the system. The "generic cluster" is selected, and the "Thread 0" (0.34) is selected.

A blue callout box points to the "Online description" option in the context menu with the text "Access online metric description via context menu".

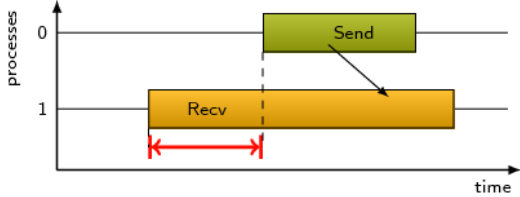




Performance properties

## 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.

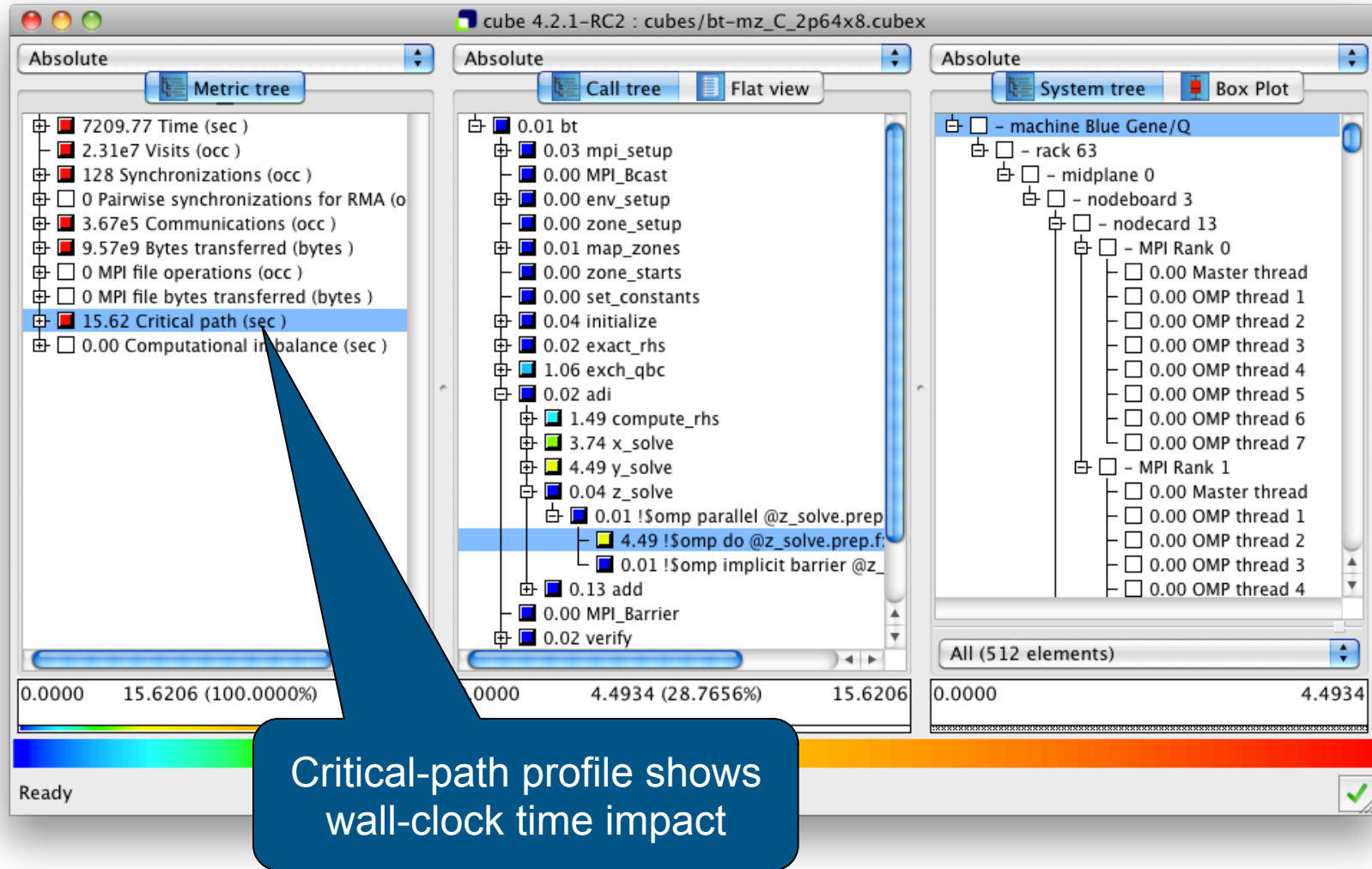
**Unit:**  
Seconds

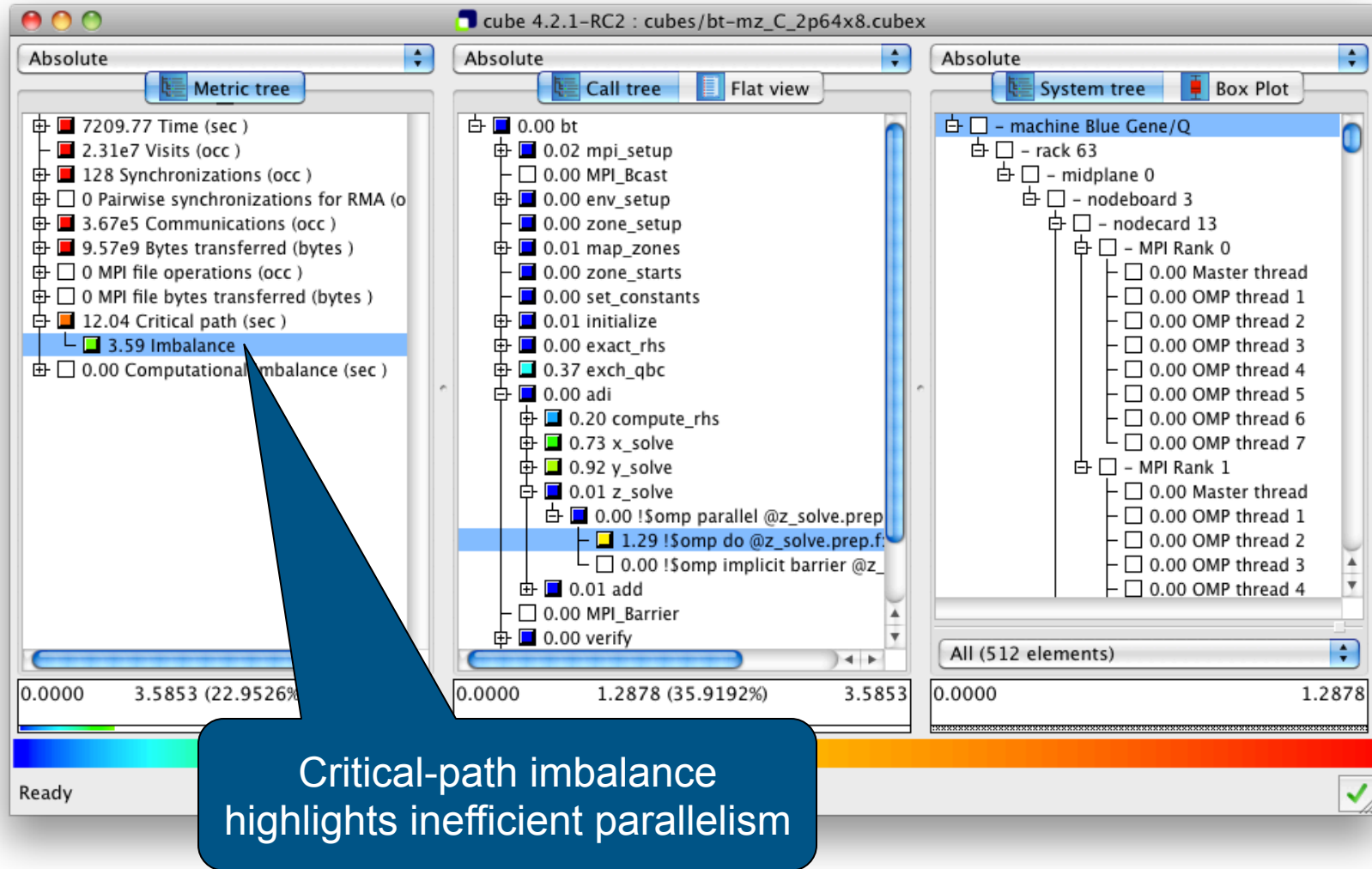
**Diagnosis:**  
Try to replace `MPI_Recv` with a non-blocking receive `MPI_Irecv` that can be posted earlier, proceed concurrently with computation, and complete with a wait operation after the message is expected to have been sent. Try to post sends earlier, such that they are available when receivers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

Close





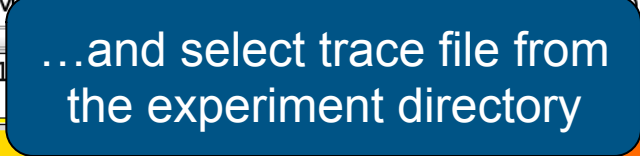
The screenshot displays the VI-HPS interface with the 'cube 4.1.1 livedvd2: scorep\_bt-mz\_B\_4x4\_trace/trace.cubex' file open. The 'Metric tree' on the left shows a hierarchy of metrics, with '1.38 Late Sender' selected. A context menu is open over this metric, listing options such as 'Info', 'Full info', 'Online description', 'Expand/collapse', 'Find items', 'Find Next', 'Clear found items', 'Copy to clipboard', 'Create derived metric...', 'Remove metric...', 'Statistics', and 'Max severity in trace browser'. The 'Statistics' option is highlighted. A callout bubble points to the 'Statistics' option with the text 'Access pattern instance statistics via context menu'. Another callout bubble points to the 'Statistics info' window with the text 'Click to get statistics details'. The 'Statistics info' window displays the following data for the 'mpi\_latesender' pattern:

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%

The 'Statistics info' window also includes a 'To Clipboard' button and a 'Close' button. The 'Statistics' window shows a bar chart with a value of 0.035 on the y-axis and a 'Close' button at the bottom. The bottom of the interface shows a color-coded bar representing metric severity, with a legend indicating 'Shows metric statistics'.



# V-HPS



# Show most severe pattern instances

cube 4.1.1 livedvd2: scorep\_bt-mz\_B\_4x4\_trace/trace.cubex

File Display Topology Help

Absolute Absolute Absolute

Metric tree Call tree Flat view 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 Completion
      - 0.00 Remote Memory Access
    - 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 1.38 (0.41%) 337.45

0.00 MAIN

- 0.00 mpi\_setenv
- 0.00 MPI\_Barrier
- 0.00 env\_set
- 0.00 zone\_set
- 0.00 map\_zone
- 0.00 zone\_start
- 0.00 set\_constants
- 0.00 initialize
- 0.00 exact\_rank
- 0.00 exchange
- 0.00 copy
- 0.00 copy
- 0.00 MPI\_Init
- 1.38 MPI\_Wait**
- 0.00 adi
- 0.00 MPI\_Barrier
- 0.00 verify
- 0.00 MPI\_Reduce
- 0.00 print\_results
- 0.00

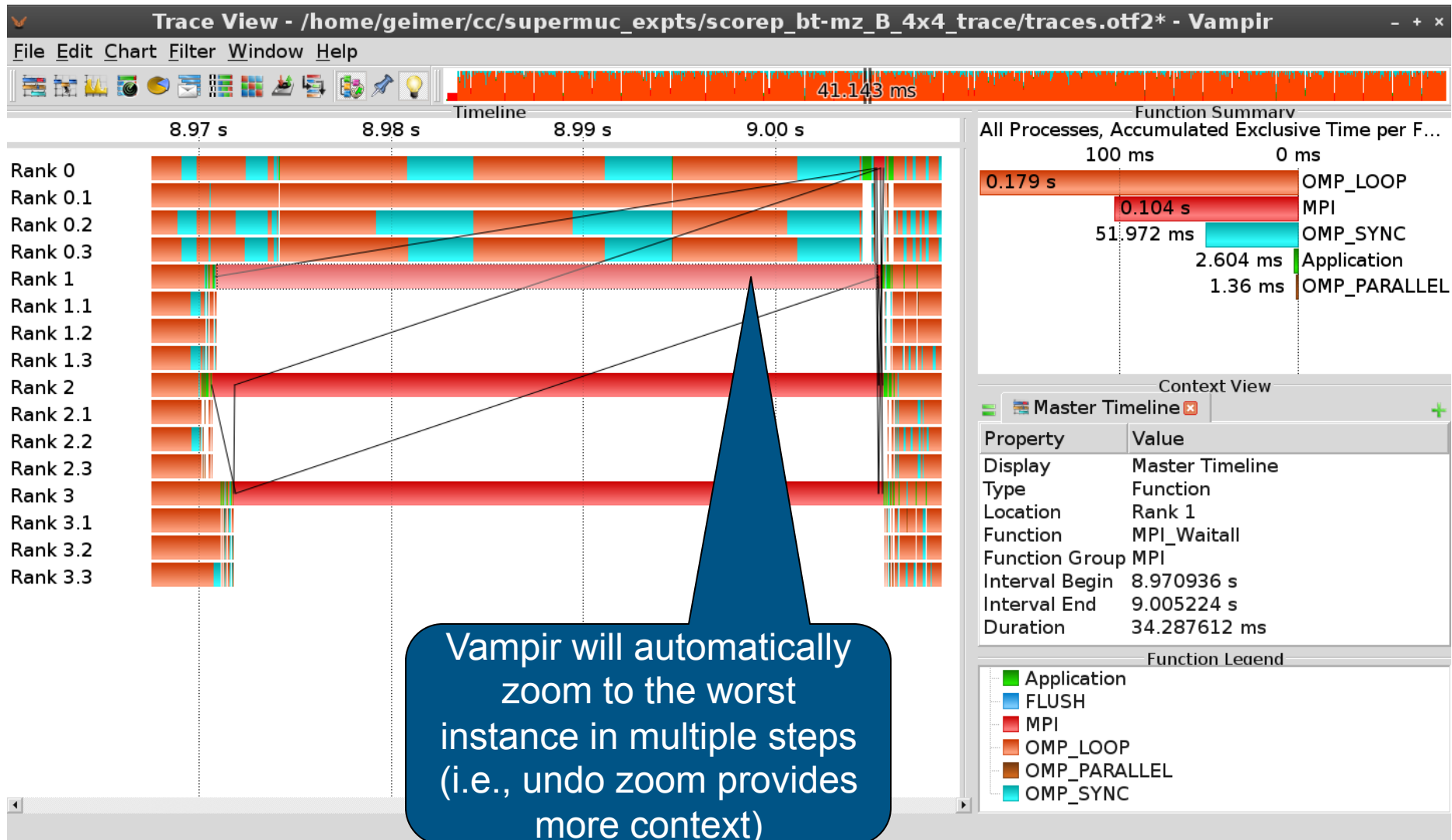
0.00 1.38

Shows the most severe instance of pattern in trace browser

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



# Investigate most severe instance in Vampir





**Website:** [www.scalasca.org](http://www.scalasca.org)  
**User support:** [scalasca@fz-juelich.de](mailto:scalasca@fz-juelich.de)

