JÜLICH
FORSCHUNGSZENTRUM

Performance analysis of Sweep3D on Blue Gene/P with Scalasca

2010-04-23 | **Brian J. N. Wylie**, David Böhme, Bernd Mohr, Zoltán Szebenyi & Felix Wolf
Jülich Supercomputing Centre

b.wylie@fz-juelich.de

Overview

Introduction

- Scalasca performance analysis toolset
- Blue Gene/P supercomputer
- Sweep3D compact application

Measurements & Analyses

- Default & improved Sweep3D configurations
- Refined instrumentation & analyses

Scalasca scalability

Conclusions

Scalasca project

Overview

- Helmholtz Initiative & Networking Fund project started in 2006
  - *Headed by Prof. Felix Wolf (RWTH Aachen, GRS & JSC)*
- Follow-up to pioneering KOJAK project (started 1998)
  - *Automatic pattern-based trace analysis*

Objective

- Development of a **scalable** **performance analysis** toolset
- Specifically targeting **large-scale** **parallel applications**

Status

- Scalasca v1.3 released in March 2010
- Available for download from www.scalasca.org

Supercomputers at Jülich Supercomputing Centre

Period	Machine	#Cores
2003-8	"jump" IBM p690 cluster	1,312
2006-8	"jubl" IBM Blue Gene/L	16,384
2009-	"juropa" Bull/Sun/Intel blades	26,304
2009-	"jugene" IBM Blue Gene/P	294,912

[jugene is the largest system in the Top500 list, Jun/Nov 2009]

JUGENE - IBM Blue Gene/P system

- 72 racks with 32 nodecards of 32 compute nodes (multiple networks)
  - *Quad-core 850 MHz PowerPC 450 processors*
  - *2 Gbytes main memory (aggregate 144 TB)*
- Proprietary Linux microkernel, MPI, XL compilers, GPFS filesystem

Large-scale parallel applications (* analyzed by Scalasca)

Selected applications from Jülich Blue Gene Scaling Workshops:

- 294,912 (288k) cores
  - *GENE (MPI-RZG/D) gyrokinetic turbulence**
  - *KKRnano (FZJ/D) electronic structure**
  - *MUPHY/PEBS (Harvard/USA) MD/evolutionary biology*
  - *OSIRIS (IST/PT) 3D relativistic plasma**
  - *PHASTA (RPI/USA) unsteady fluid dynamics**
  - *UNIC (ANL/USA) neutron transport in fast reactors*
  - *(various) lattice QCD*
- 262,144 (256k) cores
  - *MP2C (FZJ/D) mesoscale particle dynamics**
  - *NEK5000 (ANL/USA) CFD**

Sweep3D

Ubiquitous ASCI benchmark code from Los Alamos National Laboratory

- 3-dimensional neutron transport simulation
- direct order solve uses diagonal wavefront sweeps over grid cells combined with pipelining of blocks of *k*-planes and octants
- execution performance extensively modeled & analyzed

MPI parallel version using 2D domain decomposition

- ~2,000 lines of code (12 source modules), mostly Fortran77
- very portable, and highly scalable
- tunable via input deck, e.g., number of *k*-planes in blocks (MK)
- benchmark configuration does 12 iterations
  - *flux correction 'fixups' applied after 7th iteration*

Scalasca features

Open source, New BSD license

Portable

- IBM BlueGene/L, BlueGene/P, SP & blade clusters, Cray XT4/5, NEC SX, SGI Altix, SiCortex, Linux cluster® (SPARC, x86-64), ...

Supports typical HPC languages & parallel programming paradigms

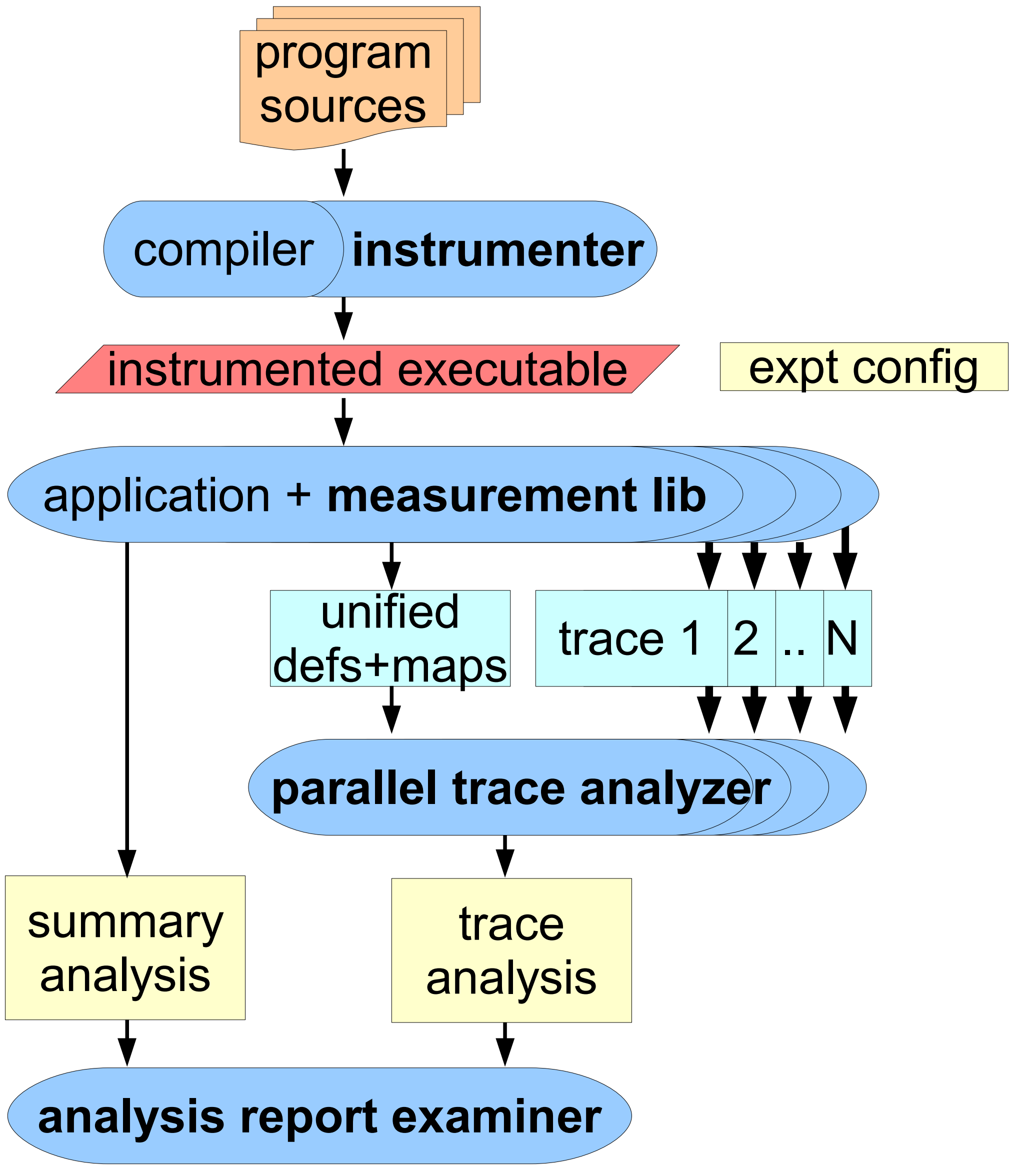
- Fortran, C, C++
- MPI, OpenMP & hybrid MPI/OpenMP

Integrated instrumentation, measurement & analysis toolset

- Customizable automatic/manual instrumentation
- Runtime summarization (*aka* profiling)
- Automatic event trace analysis

Scalasca components

- Automatic program instrumenter creates instrumented executable
- Unified measurement library supports both
  - *runtime summarization*
  - *trace file generation*
- Parallel, replay-based event trace analyzer invoked automatically on set of traces
- Common analysis report explorer & examination/processing tools



Scalasca usage

1. Prepare application objects and executable for measurement:
   - `scalasca -instrument  mpixlf77 -O3 -qarch=450 -qtune=450 …`
2. Run application under control of measurement nexus:
   - `scalasca -analyze     mpirun -mode VN -np 294912 sweep3d`
     - *epik_sweep3d_vn294912_sum* experiment produced
   - `scalasca -analyze -t  mpirun -mode VN -np 294912 sweep3d`
     - *epik_sweep3d_vn294912_trace* experiment produced
3. Interactively explore measurement analysis report
   - `scalasca -examine     epik_sweep3d_vn294912_trace`
     - *epik_sweep3d_vn294912_trace/trace.cube.gz* presented

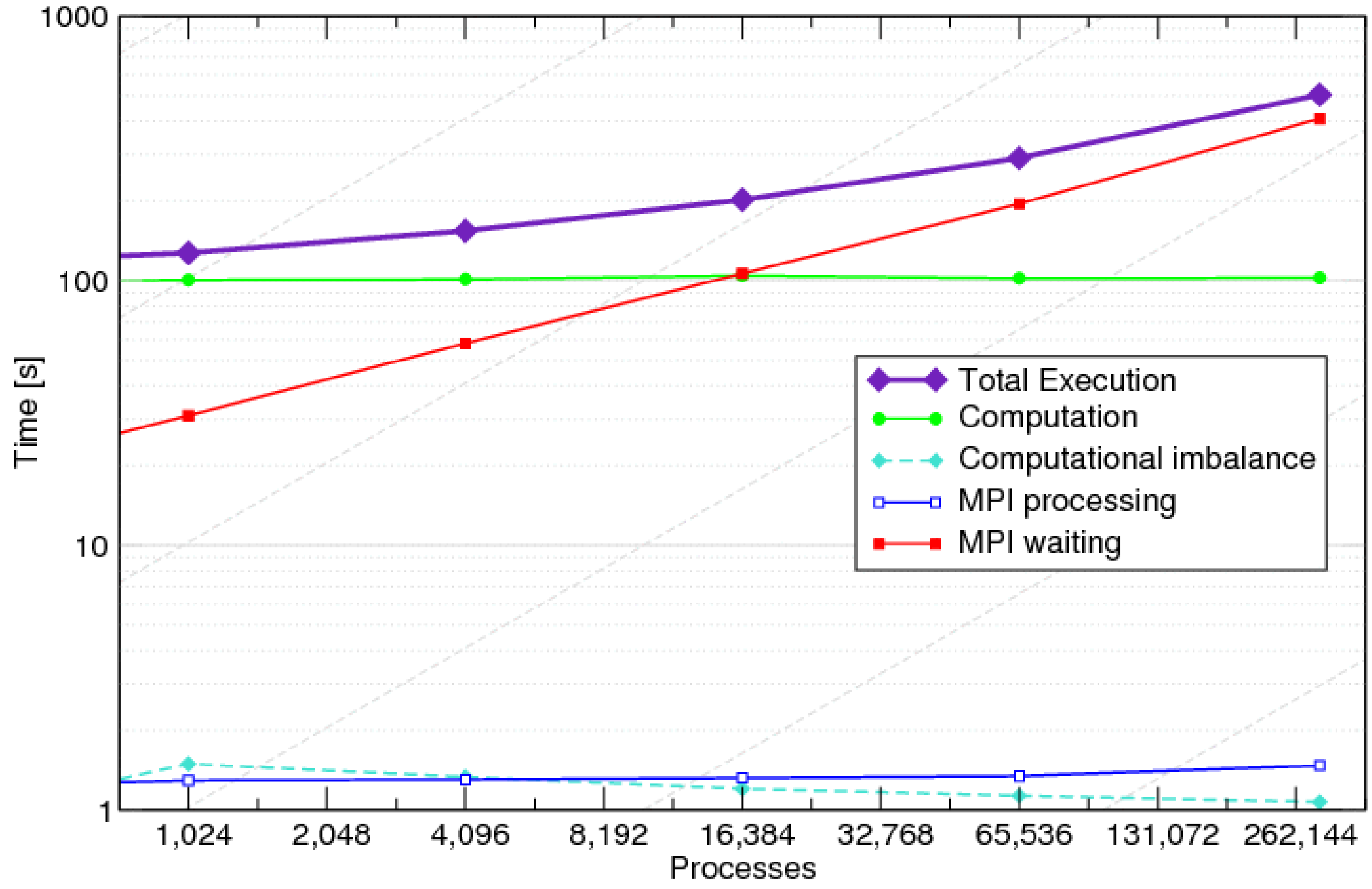
Measurement & analysis methodology

1. Run uninstrumented/optimized version (as reference for validation)
   - determine memory available for measurement
2. Run automatically-instrumented version collecting runtime summary
   - determine functions with excessive overheads
     - *examine distortion and trace buffer capacity requirement*
   - if necessary, prepare filter file and repeat measurement
3. Reconfigure measurement to collect and automatically analyze traces
4. Refine instrumentation by manually annotating key code sections
   - use EPIK instrumentation API macros

Compare original MK=10 and alternative MK=1 Sweep3D configurations

- varies computation wavefront pipelining concurrency

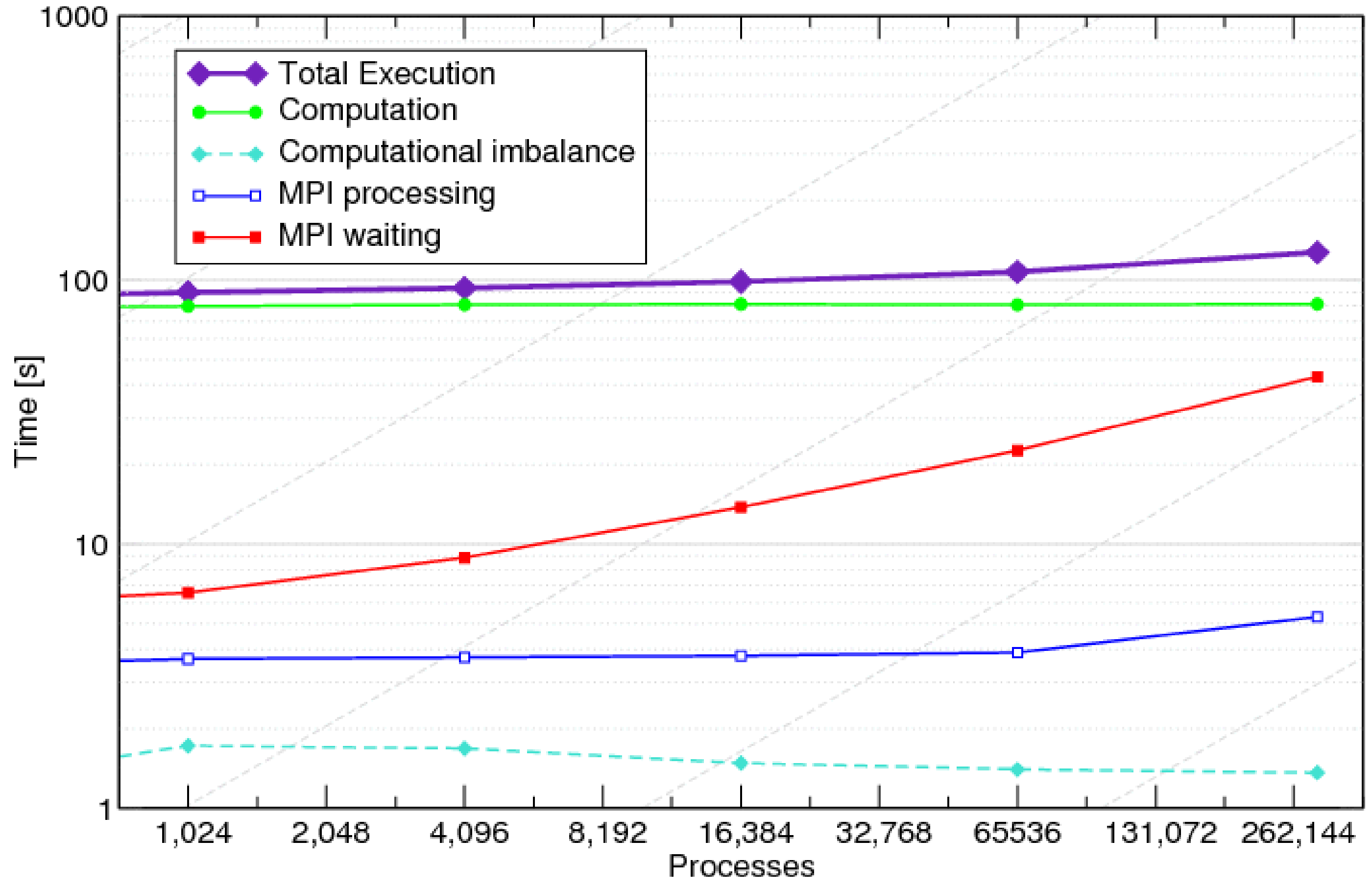
Sweep3D execution time scaling: original MK=10



- Default input deck settings
- *Weak scaling* due to fixed problem size per process
- Reasonable scalability(?)
- Constant time for *computation*
- Rapid growth in *MPI* time, which is almost all *waiting* time

[Replication of scaling characteristics reported by others]

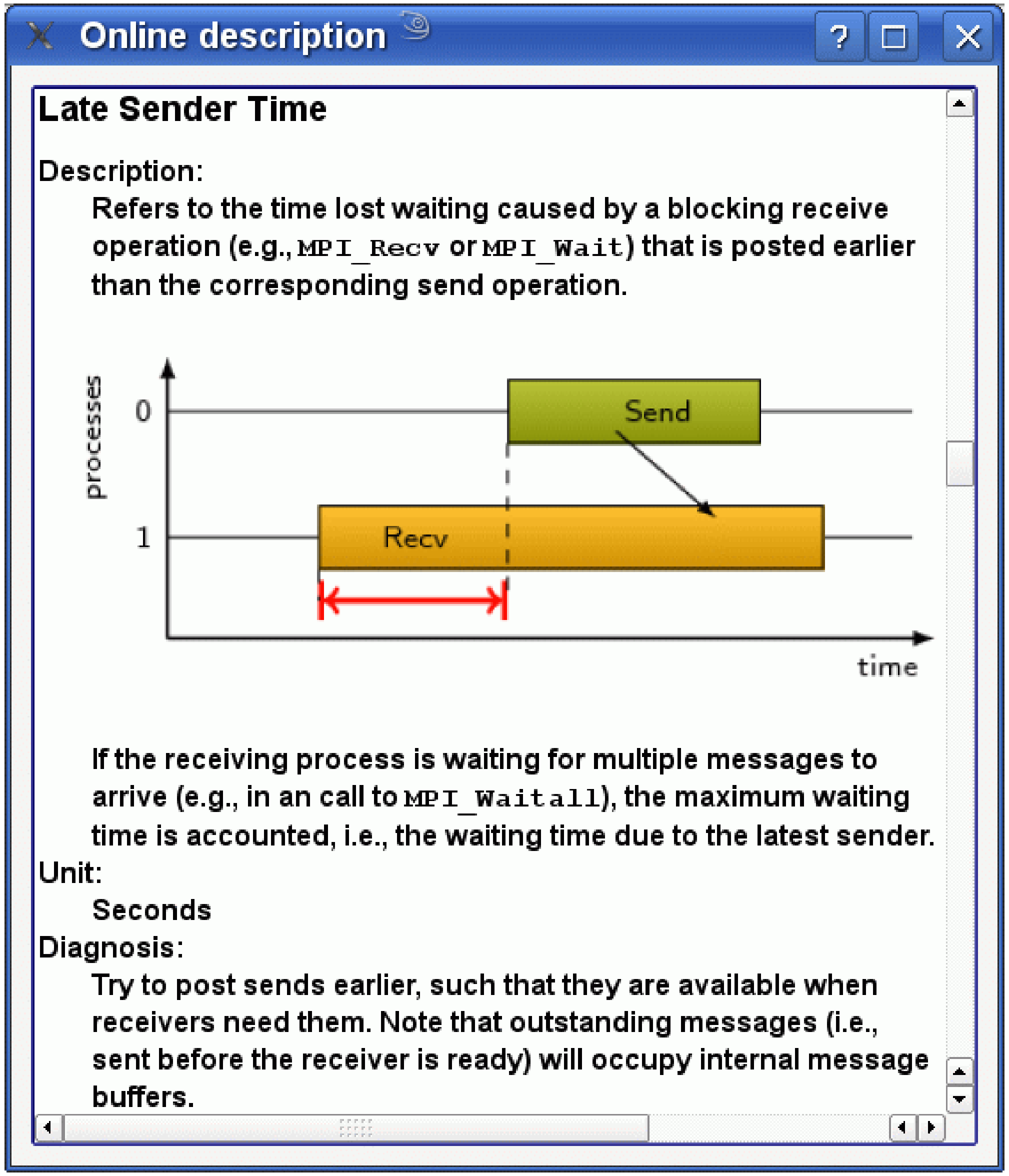
Sweep3D execution time scaling: improved MK=1



- Much improved performance and scalability
- *Computation* 20% faster (better caching?)
- *MPI processing* time increased (10x messages)
- *MPI waiting* time significantly reduced, though still growing markedly

[Single *k*-planes rather than default batches of 10]

Scalasca GUI description for *Late Sender* metric



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.

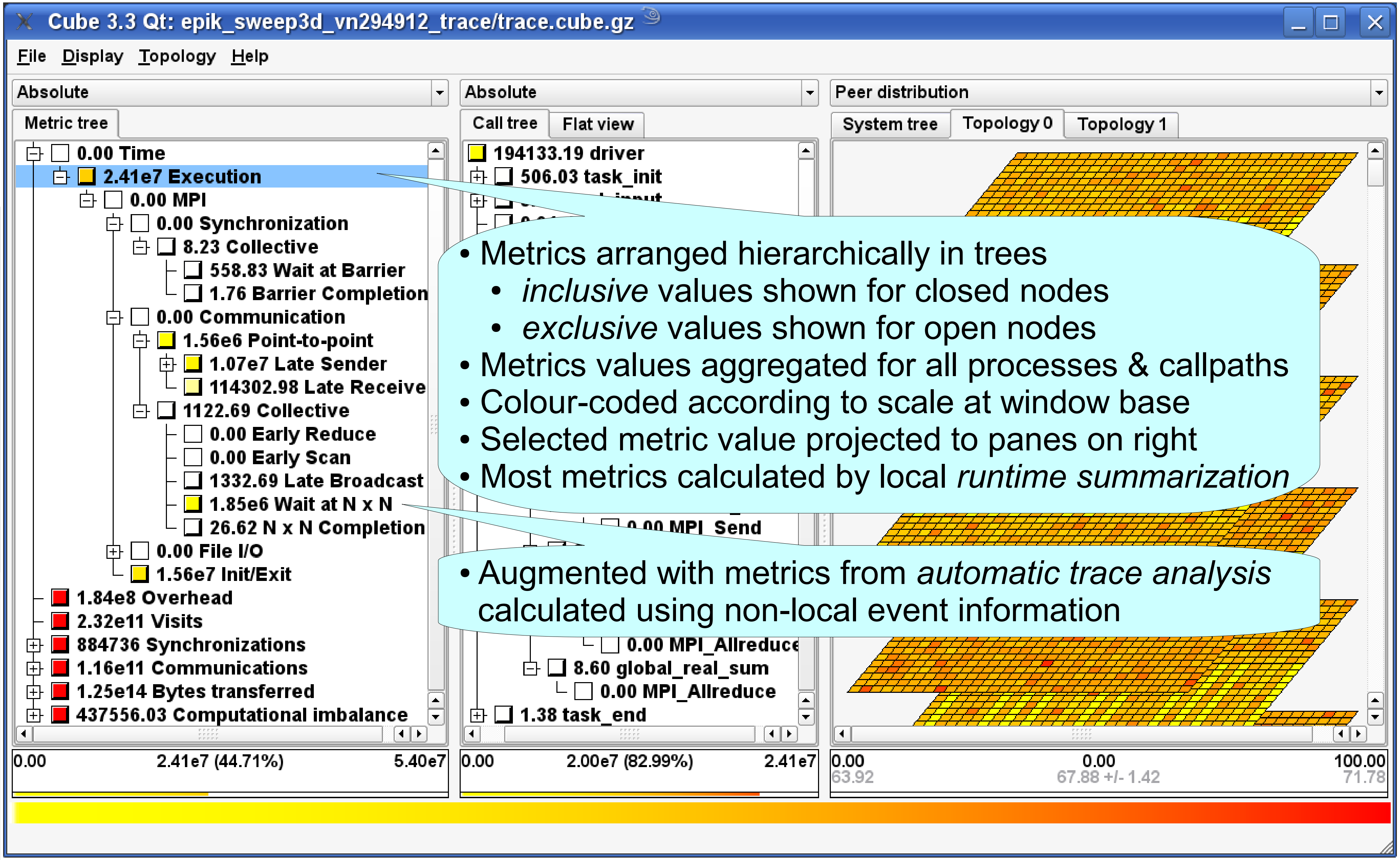
If the receiving process is waiting for multiple messages to arrive (e.g., in an call to `MPI_Waitall`), the maximum waiting time is accounted, i.e., the waiting time due to the latest sender.

Unit:
Seconds

Diagnosis:
Try to 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.

- Analysis report explorer GUI provides hyperlinked online descriptions of metrics
- Diagnosis hints suggest how to refine diagnosis of performance problems and possible remediation

Scalasca analysis report: performance metric tree



Cube 3.3 Qt: epik_sweep3d_vn294912_trace/trace.cube.gz

File Display Topology Help

Absolute | Absolute | Peer distribution

Metric tree | Call tree | Flat view | System tree | Topology 0 | Topology 1

```
0.00 Time
  2.41e7 Execution
    0.00 MPI
      0.00 Synchronization
        8.23 Collective
          558.83 Wait at Barrier
          1.76 Barrier Completion
      0.00 Communication
        1.56e6 Point-to-point
          1.07e7 Late Sender
          114302.98 Late Receive
        1122.69 Collective
          0.00 Early Reduce
          0.00 Early Scan
          1332.69 Late Broadcast
          1.85e6 Wait at N x N
          26.62 N x N Completion
      0.00 File I/O
      1.56e7 Init/Exit
1.84e8 Overhead
2.32e11 Visits
884736 Synchronizations
1.16e11 Communications
1.25e14 Bytes transferred
437556.03 Computational imbalance
```

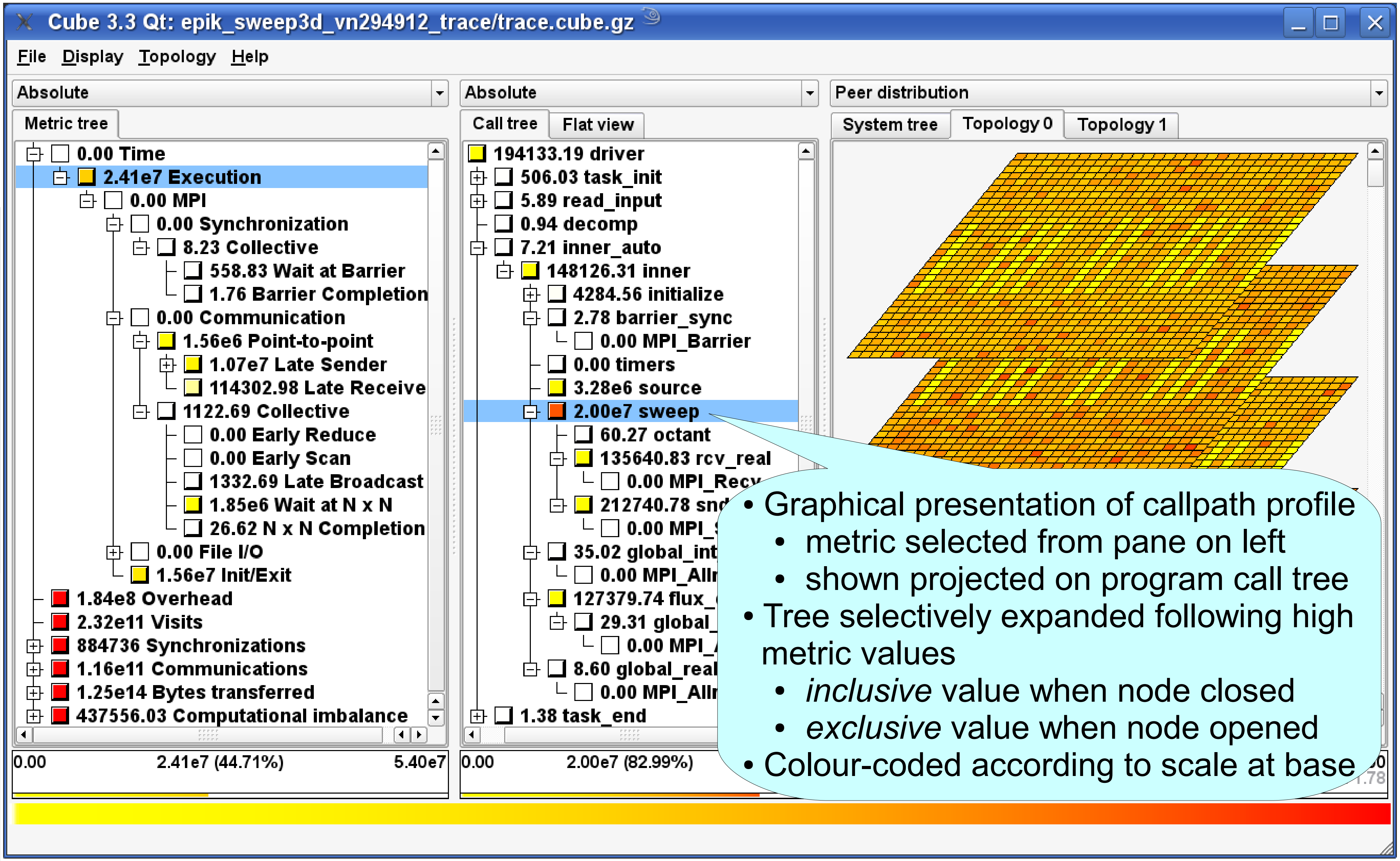
```
194133.19 driver
  506.03 task_init
  ...
  0.00 MPI_Send
  0.00 MPI_Allreduce
  8.60 global_real_sum
    0.00 MPI_Allreduce
  1.38 task_end
```

0.00 2.41e7 (44.71%) 5.40e7 | 0.00 2.00e7 (82.99%) 2.41e7 | 0.00 63.92 0.00 67.88 +/- 1.42 100.00 71.78

- Metrics arranged hierarchically in trees
  - *inclusive* values shown for closed nodes
  - *exclusive* values shown for open nodes
- Metrics values aggregated for all processes & callpaths
- Colour-coded according to scale at window base
- Selected metric value projected to panes on right
- Most metrics calculated by local *runtime summarization*

- Augmented with metrics from *automatic trace analysis* calculated using non-local event information

Scalasca analysis report: program call tree



Cube 3.3 Qt: epik_sweep3d_vn294912_trace/trace.cube.gz

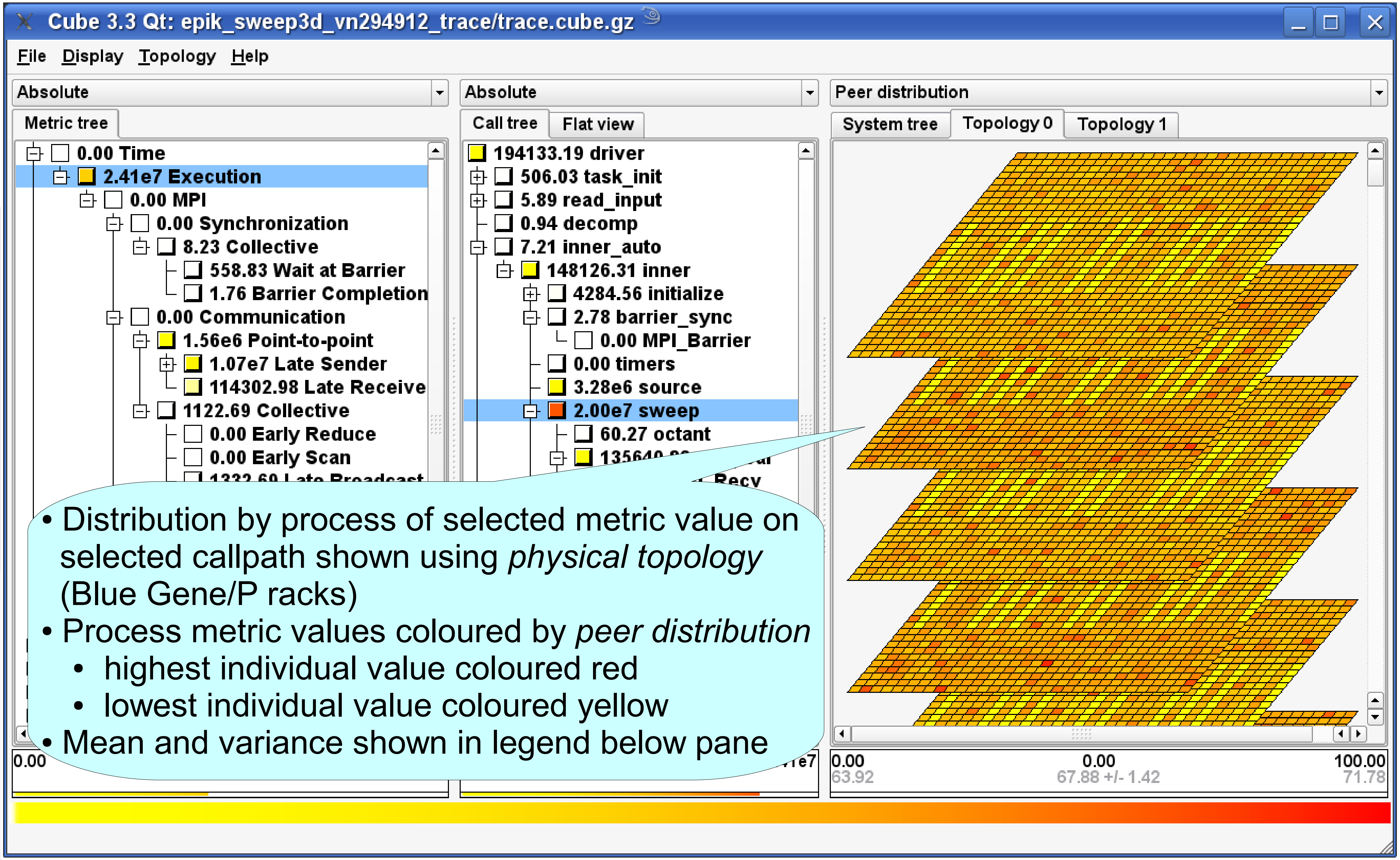
```
0.00 Time
  2.41e7 Execution
    0.00 MPI
      0.00 Synchronization
        8.23 Collective
          558.83 Wait at Barrier
          1.76 Barrier Completion
      0.00 Communication
        1.56e6 Point-to-point
          1.07e7 Late Sender
          114302.98 Late Receive
        1122.69 Collective
          0.00 Early Reduce
          0.00 Early Scan
          1332.69 Late Broadcast
          1.85e6 Wait at N x N
          26.62 N x N Completion
      0.00 File I/O
      1.56e7 Init/Exit
1.84e8 Overhead
2.32e11 Visits
884736 Synchronizations
1.16e11 Communications
1.25e14 Bytes transferred
437556.03 Computational imbalance
```

```
194133.19 driver
  506.03 task_init
  5.89 read_input
  0.94 decomp
  7.21 inner_auto
    148126.31 inner
      4284.56 initialize
      2.78 barrier_sync
        0.00 MPI_Barrier
      0.00 timers
      3.28e6 source
      2.00e7 sweep
        60.27 octant
        135640.83 rcv_real
          0.00 MPI_Recv
        212740.78 snd
          0.00 MPI_
      35.02 global_int
        0.00 MPI_Allr
      127379.74 flux_
        29.31 global_
          0.00 MPI_
      8.60 global_real
        0.00 MPI_Allr
  1.38 task_end
```

```
0.00          2.41e7 (44.71%)          5.40e7
0.00          2.00e7 (82.99%)
```

- Graphical presentation of callpath profile
  - metric selected from pane on left
  - shown projected on program call tree
- Tree selectively expanded following high metric values
  - *inclusive* value when node closed
  - *exclusive* value when node opened
- Colour-coded according to scale at base

Scalasca analysis report: system tree/topology



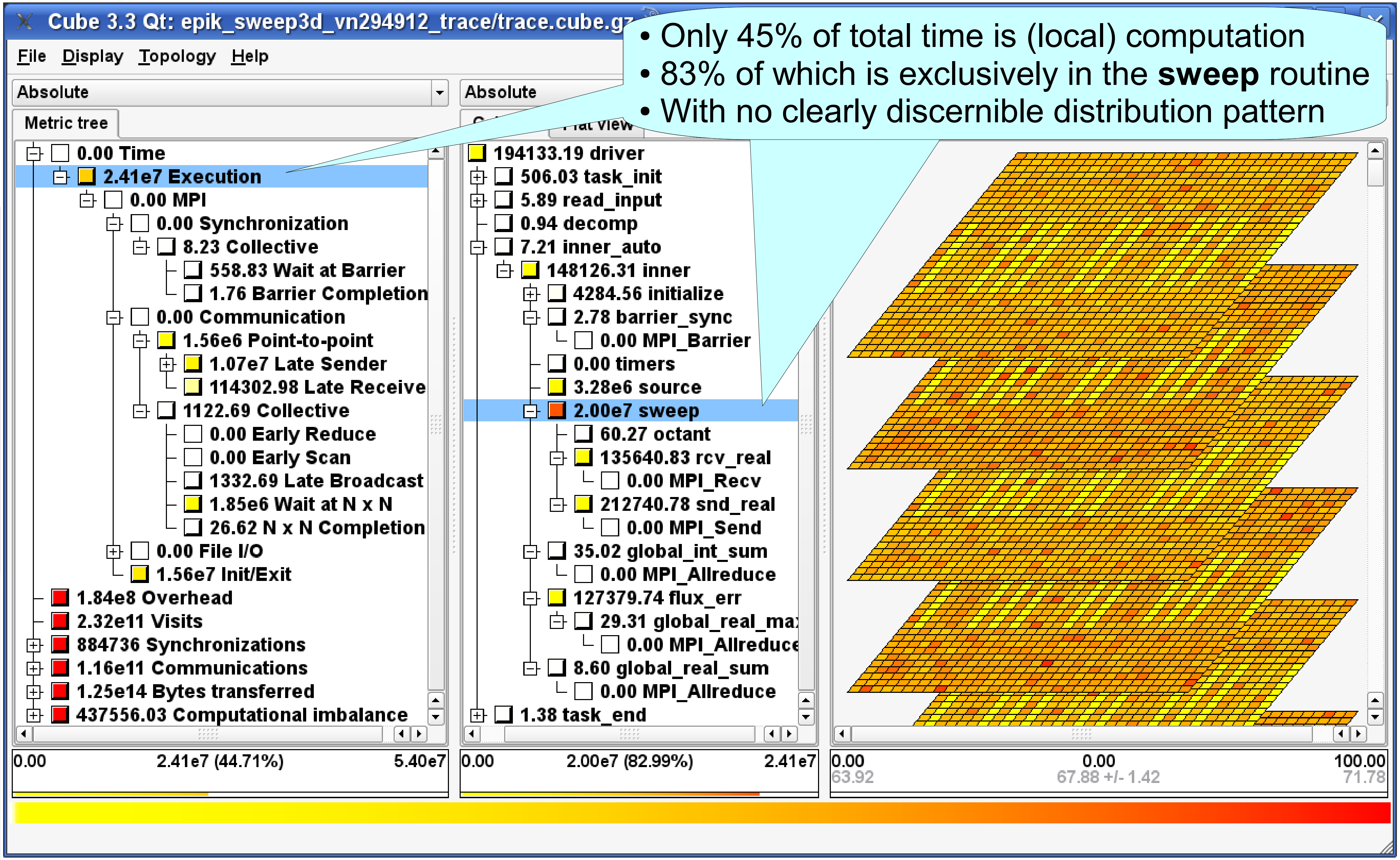
```
0.00 Time
  2.41e7 Execution
    0.00 MPI
      0.00 Synchronization
        8.23 Collective
          558.83 Wait at Barrier
          1.76 Barrier Completion
      0.00 Communication
        1.56e6 Point-to-point
          1.07e7 Late Sender
          114302.98 Late Receive
        1122.69 Collective
          0.00 Early Reduce
          0.00 Early Scan
          1332.60 Late Broadcast
```

```
194133.19 driver
  506.03 task_init
  5.89 read_input
  0.94 decomp
  7.21 inner_auto
    148126.31 inner
      4284.56 initialize
      2.78 barrier_sync
        0.00 MPI_Barrier
      0.00 timers
      3.28e6 source
      2.00e7 sweep
        60.27 octant
        135640.0…
        …Recv
```

```
0.00                                   0.00                 100.00
63.92                            67.88 +/- 1.42              71.78
```

- Distribution by process of selected metric value on selected callpath shown using *physical topology* (Blue Gene/P racks)
- Process metric values coloured by *peer distribution*
  - highest individual value coloured red
  - lowest individual value coloured yellow
- Mean and variance shown in legend below pane

Scalasca analysis report: sweep computation time



Cube 3.3 Qt: epik_sweep3d_vn294912_trace/trace.cube.gz

File Display Topology Help

Absolute | Absolute

Metric tree | Flat view

```
0.00 Time
  2.41e7 Execution
    0.00 MPI
      0.00 Synchronization
        8.23 Collective
          558.83 Wait at Barrier
          1.76 Barrier Completion
      0.00 Communication
        1.56e6 Point-to-point
          1.07e7 Late Sender
          114302.98 Late Receive
        1122.69 Collective
          0.00 Early Reduce
          0.00 Early Scan
          1332.69 Late Broadcast
          1.85e6 Wait at N x N
          26.62 N x N Completion
      0.00 File I/O
      1.56e7 Init/Exit
1.84e8 Overhead
2.32e11 Visits
884736 Synchronizations
1.16e11 Communications
1.25e14 Bytes transferred
437556.03 Computational imbalance
```

```
194133.19 driver
  506.03 task_init
  5.89 read_input
  0.94 decomp
  7.21 inner_auto
    148126.31 inner
      4284.56 initialize
      2.78 barrier_sync
        0.00 MPI_Barrier
      0.00 timers
      3.28e6 source
      2.00e7 sweep
        60.27 octant
        135640.83 rcv_real
          0.00 MPI_Recv
        212740.78 snd_real
          0.00 MPI_Send
      35.02 global_int_sum
        0.00 MPI_Allreduce
      127379.74 flux_err
        29.31 global_real_max
          0.00 MPI_Allreduce
      8.60 global_real_sum
        0.00 MPI_Allreduce
  1.38 task_end
```

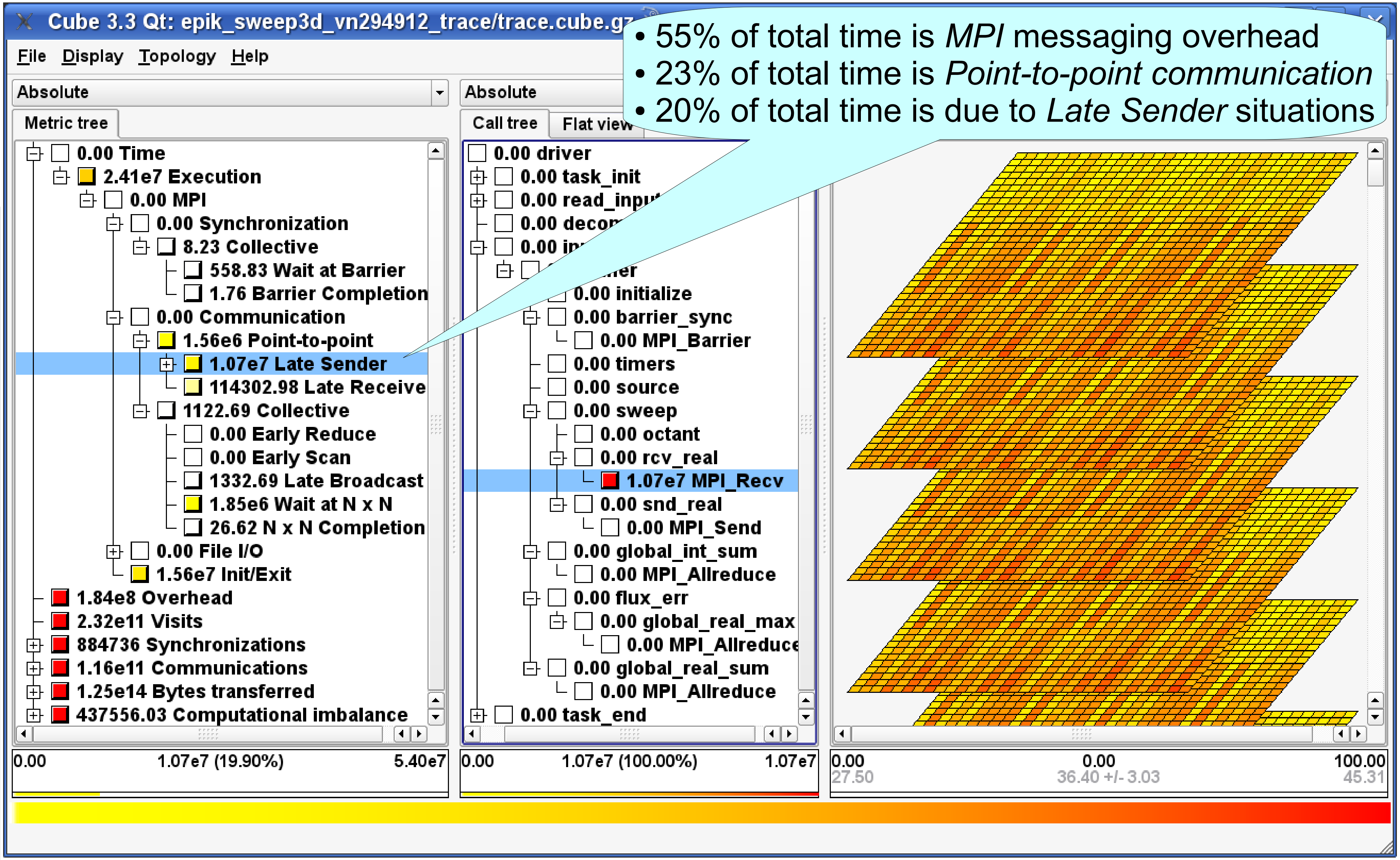
0.00 | 2.41e7 (44.71%) | 5.40e7

0.00 | 2.00e7 (82.99%) | 2.41e7

0.00 | 0.00 | 100.00
63.92 | 67.88 +/- 1.42 | 71.78

- Only 45% of total time is (local) computation
- 83% of which is exclusively in the **sweep** routine
- With no clearly discernible distribution pattern

Scalasca analysis report: *Late Sender* time



- 55% of total time is *MPI* messaging overhead
- 23% of total time is *Point-to-point communication*
- 20% of total time is due to *Late Sender* situations

Cube 3.3 Qt: epik_sweep3d_vn294912_trace/trace.cube.gz

File Display Topology Help

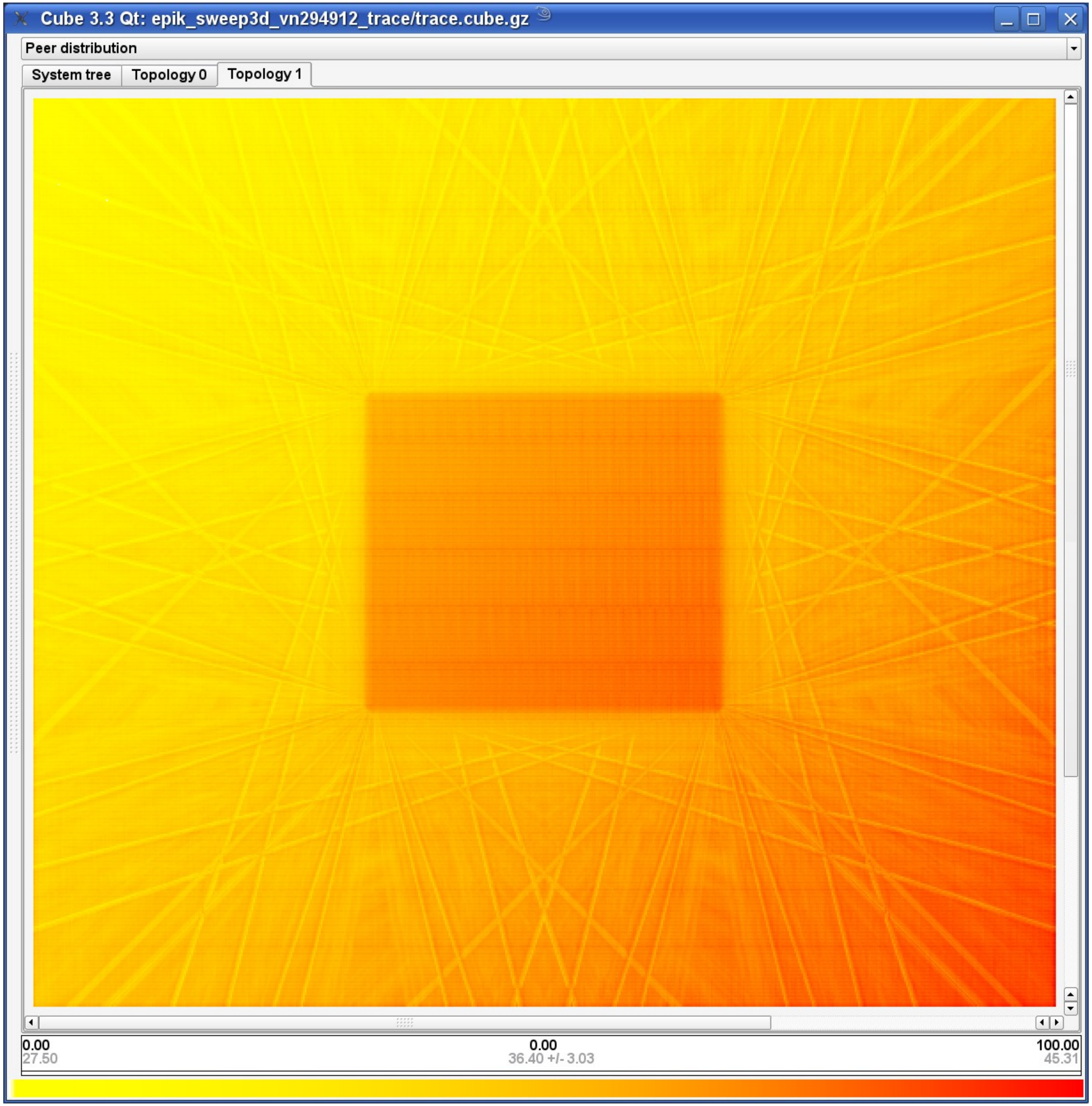
Absolute | Absolute

Metric tree | Call tree | Flat view

```
0.00 Time
  2.41e7 Execution
    0.00 MPI
      0.00 Synchronization
        8.23 Collective
          558.83 Wait at Barrier
          1.76 Barrier Completion
      0.00 Communication
        1.56e6 Point-to-point
          1.07e7 Late Sender
          114302.98 Late Receive
        1122.69 Collective
          0.00 Early Reduce
          0.00 Early Scan
          1332.69 Late Broadcast
          1.85e6 Wait at N x N
          26.62 N x N Completion
      0.00 File I/O
      1.56e7 Init/Exit
1.84e8 Overhead
2.32e11 Visits
884736 Synchronizations
1.16e11 Communications
1.25e14 Bytes transferred
437556.03 Computational imbalance
```

```
0.00 driver
  0.00 task_init
  0.00 read_input
  0.00 decom
  0.00 in
    ...ner
      0.00 initialize
      0.00 barrier_sync
        0.00 MPI_Barrier
      0.00 timers
      0.00 source
      0.00 sweep
        0.00 octant
        0.00 rcv_real
          1.07e7 MPI_Recv
        0.00 snd_real
          0.00 MPI_Send
      0.00 global_int_sum
        0.00 MPI_Allreduce
      0.00 flux_err
        0.00 global_real_max
          0.00 MPI_Allreduce
      0.00 global_real_sum
        0.00 MPI_Allreduce
  0.00 task_end
```

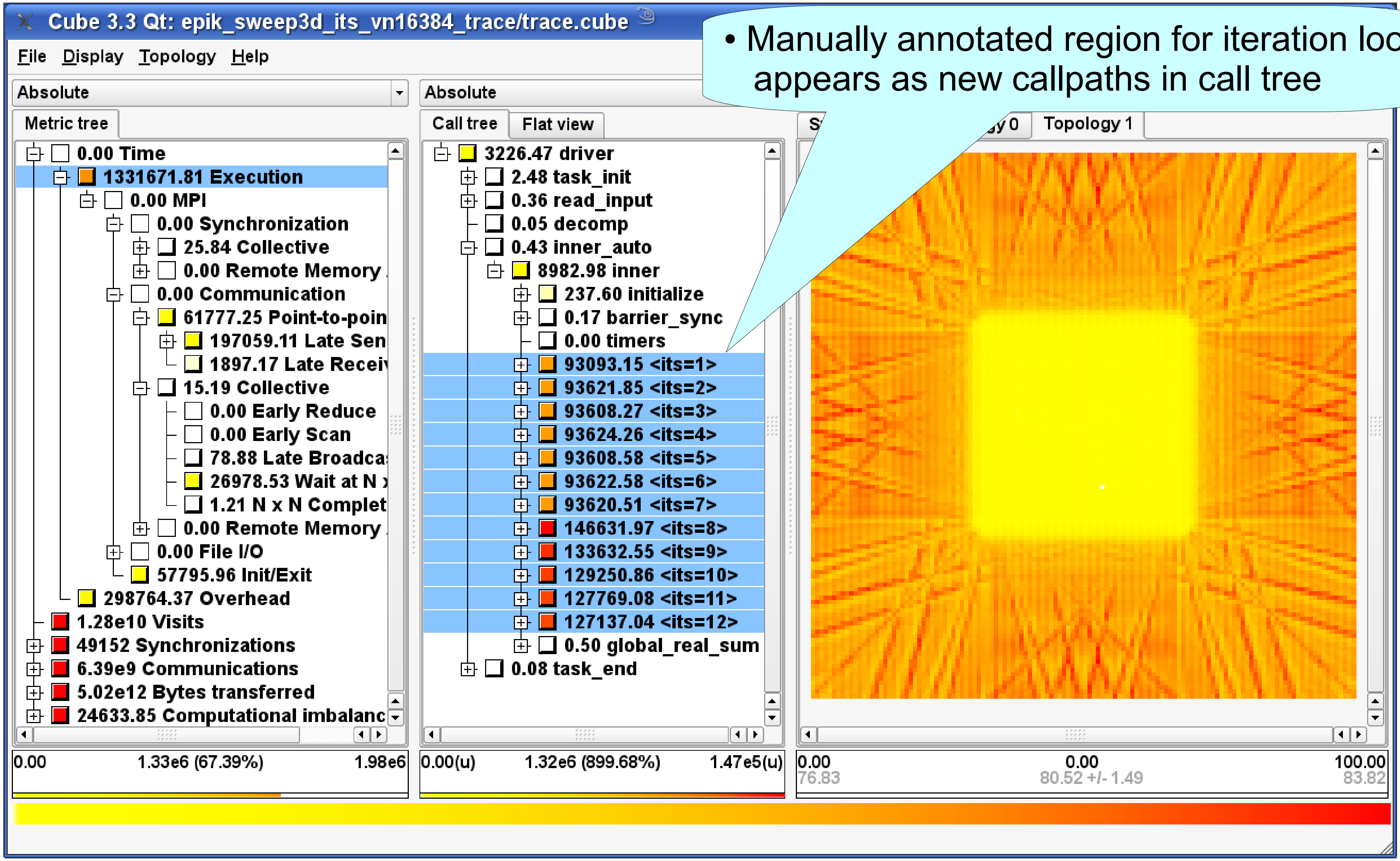
```
0.00      1.07e7 (19.90%)      5.40e7
0.00      1.07e7 (100.00%)     1.07e7
0.00            0.00           100.00
27.50     36.40 +/- 3.03       45.31
```



Sweep3D grid topology view

- Application's 2D topology of 576x512 procs
- Reveals clear pattern in the distribution of *Late Sender* metric values
- Arises from superimposing octant sweeps with imbalanced computation

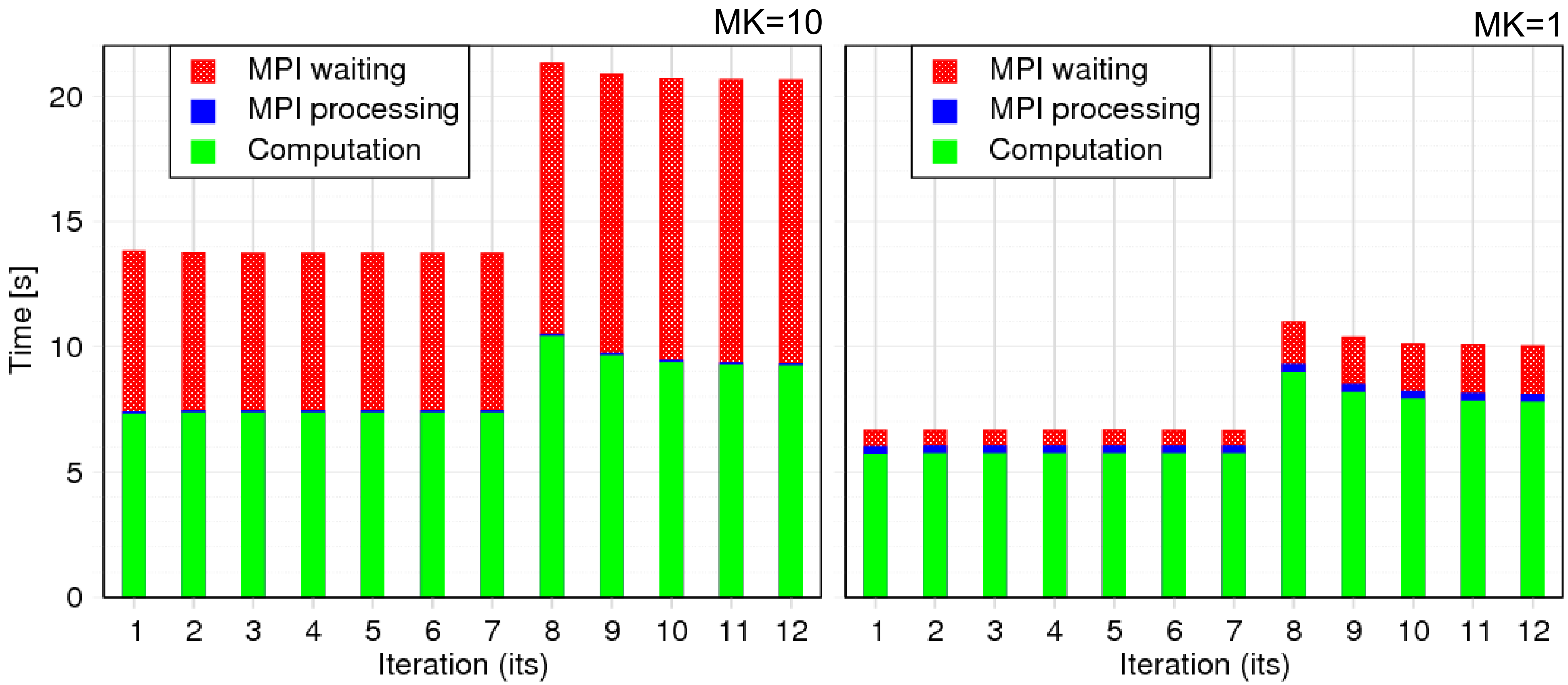
Sweep3D computation time by iteration (16,384 processes)



- Manually annotated region for iteration loop appears as new callpaths in call tree

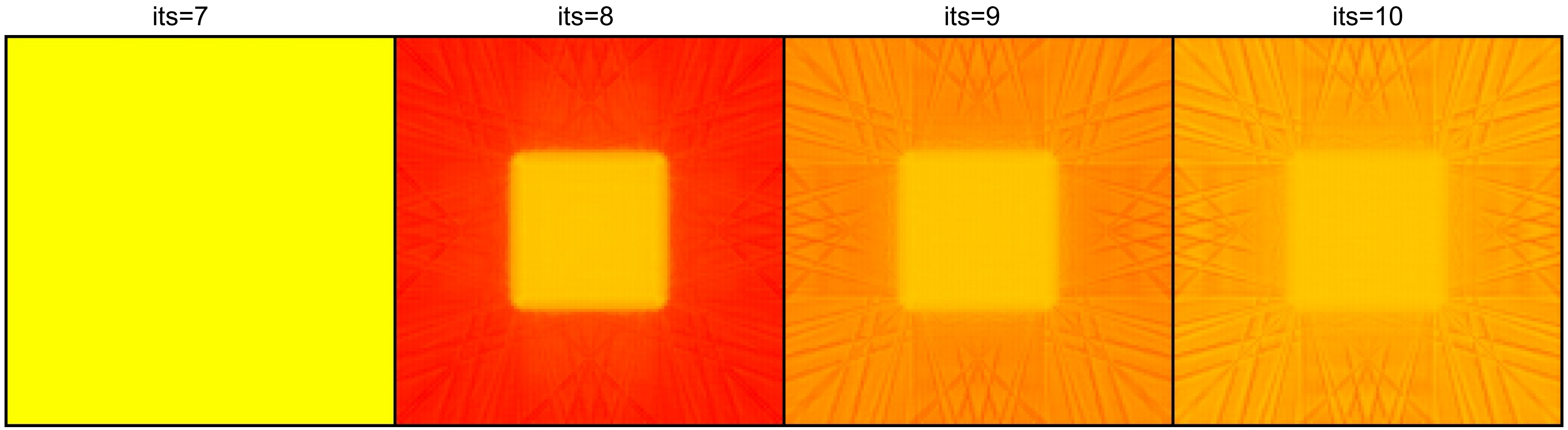
```
Cube 3.3 Qt: epik_sweep3d_its_vn16384_trace/trace.cube
File  Display  Topology  Help

Absolute                          Absolute
Metric tree                       Call tree | Flat view            ... gy 0 | Topology 1

0.00 Time                         3226.47 driver
  1331671.81 Execution              2.48 task_init
    0.00 MPI                        0.36 read_input
      0.00 Synchronization          0.05 decomp
        25.84 Collective            0.43 inner_auto
        0.00 Remote Memory            8982.98 inner
      0.00 Communication                237.60 initialize
        61777.25 Point-to-poin          0.17 barrier_sync
          197059.11 Late Sen            0.00 timers
          1897.17 Late Receiv           93093.15 <its=1>
        15.19 Collective                93621.85 <its=2>
          0.00 Early Reduce             93608.27 <its=3>
          0.00 Early Scan               93624.26 <its=4>
          78.88 Late Broadcas           93608.58 <its=5>
          26978.53 Wait at N x          93622.58 <its=6>
          1.21 N x N Complet            93620.51 <its=7>
        0.00 Remote Memory              146631.97 <its=8>
      0.00 File I/O                     133632.55 <its=9>
      57795.96 Init/Exit                129250.86 <its=10>
  298764.37 Overhead                    127769.08 <its=11>
1.28e10 Visits                          127137.04 <its=12>
49152 Synchronizations                  0.50 global_real_sum
6.39e9 Communications               0.08 task_end
5.02e12 Bytes transferred
24633.85 Computational imbalanc

0.00   1.33e6 (67.39%)   1.98e6   0.00(u)   1.32e6 (899.68%)   1.47e5(u)   0.00        0.00           100.00
                                                                          76.83   80.52 +/- 1.49   83.82
```

Iteration execution time breakdown (16,384 processes)



- Initial 7 (non-fixup) iterations faster than later 5 (fixup) iterations
- *MPI waiting* time dominates original; amplified by fixup imbalance
- *MPI processing* time a fixed cost in each iteration

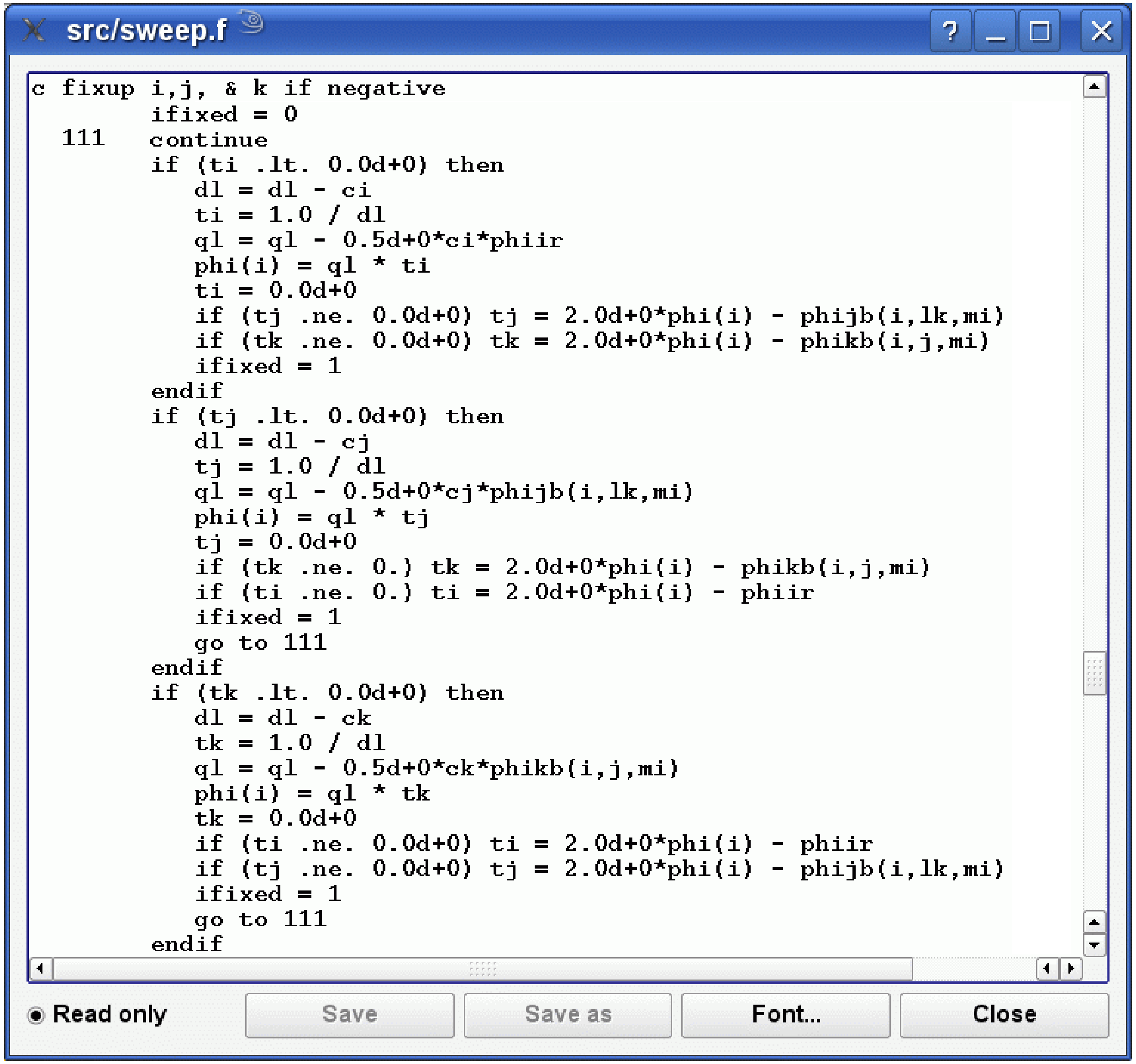
Sweep3D computation time distribution evolution



[Values coloured on scale 5 to 10 seconds]

- Initial 7 (non-fixup) iterations are very well balanced
- Fixups introduced after iteration 7 are computationally imbalanced
- Iteration 8 clearly shows central rectangular imbalance feature
- Subsequent iterations have less imbalance in the rectangular core, however, pronounced symmetric oblique lines of excess computation

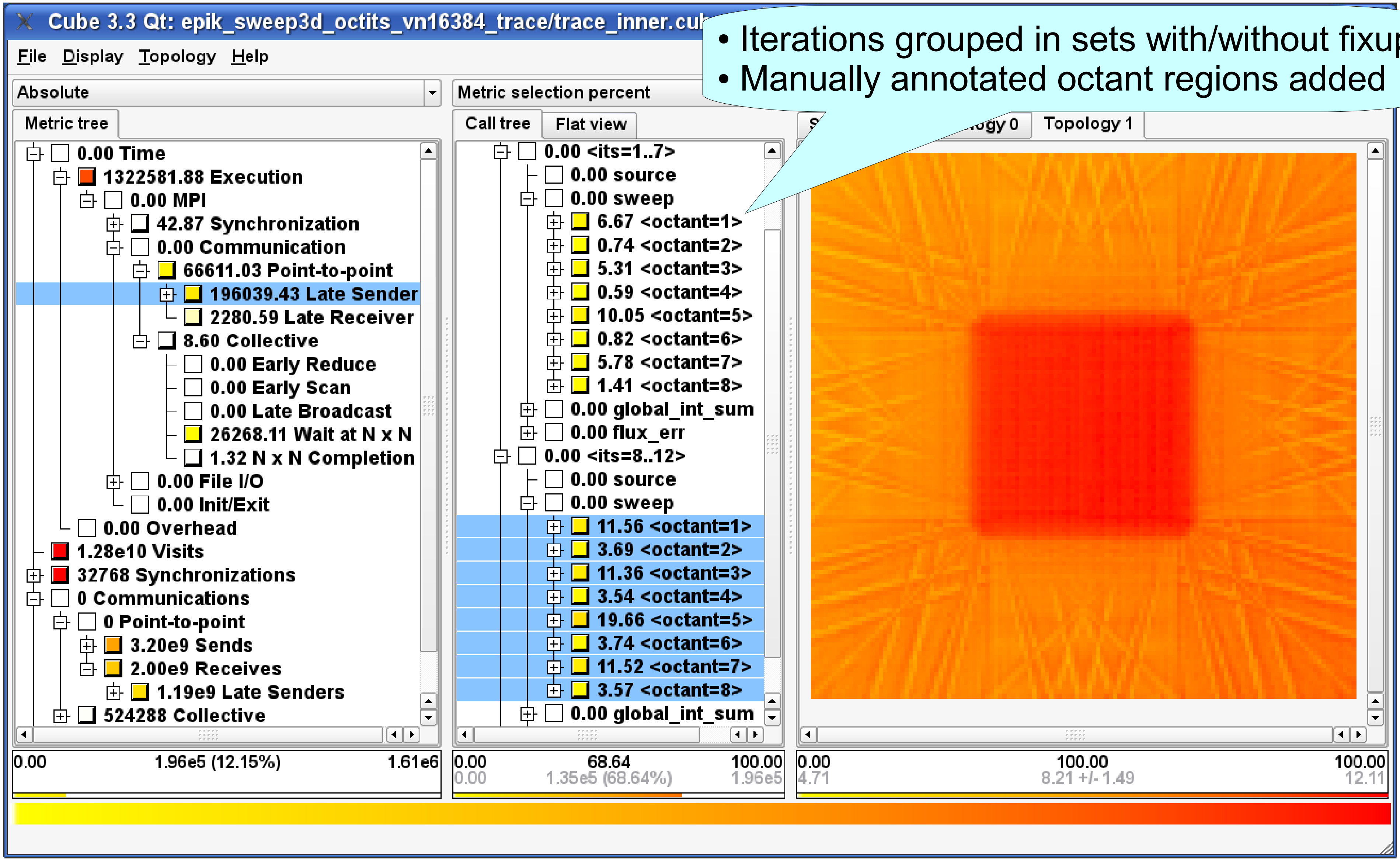
Scalasca source code browser



```
c fixup i,j, & k if negative
         ifixed = 0
  111    continue
         if (ti .lt. 0.0d+0) then
            dl = dl - ci
            ti = 1.0 / dl
            ql = ql - 0.5d+0*ci*phiir
            phi(i) = ql * ti
            ti = 0.0d+0
            if (tj .ne. 0.0d+0) tj = 2.0d+0*phi(i) - phijb(i,lk,mi)
            if (tk .ne. 0.0d+0) tk = 2.0d+0*phi(i) - phikb(i,j,mi)
            ifixed = 1
         endif
         if (tj .lt. 0.0d+0) then
            dl = dl - cj
            tj = 1.0 / dl
            ql = ql - 0.5d+0*cj*phijb(i,lk,mi)
            phi(i) = ql * tj
            tj = 0.0d+0
            if (tk .ne. 0.) tk = 2.0d+0*phi(i) - phikb(i,j,mi)
            if (ti .ne. 0.) ti = 2.0d+0*phi(i) - phiir
            ifixed = 1
            go to 111
         endif
         if (tk .lt. 0.0d+0) then
            dl = dl - ck
            tk = 1.0 / dl
            ql = ql - 0.5d+0*ck*phikb(i,j,mi)
            phi(i) = ql * tk
            tk = 0.0d+0
            if (ti .ne. 0.0d+0) ti = 2.0d+0*phi(i) - phiir
            if (tj .ne. 0.0d+0) tj = 2.0d+0*phi(i) - phijb(i,lk,mi)
            ifixed = 1
            go to 111
         endif
```

- Source file **sweep.f** contains 625-line **sweep** flow routine
- Computationally imbalanced *i*-line section where corrective 'fixups' for negative fluxes are recursively applied in *i*, *j*, *k* grid directions
- Hotspot also typically identified by profilers based on sampling

MPI *Late Sender* time variation by sweep octant



Cube 3.3 Qt: epik_sweep3d_octits_vn16384_trace/trace_inner.cub

- Iterations grouped in sets with/without fixups
- Manually annotated octant regions added

Absolute | Metric selection percent

Metric tree:

```
0.00 Time
  1322581.88 Execution
    0.00 MPI
      42.87 Synchronization
      0.00 Communication
        66611.03 Point-to-point
          196039.43 Late Sender
          2280.59 Late Receiver
        8.60 Collective
          0.00 Early Reduce
          0.00 Early Scan
          0.00 Late Broadcast
          26268.11 Wait at N x N
          1.32 N x N Completion
      0.00 File I/O
      0.00 Init/Exit
    0.00 Overhead
1.28e10 Visits
32768 Synchronizations
0 Communications
  0 Point-to-point
    3.20e9 Sends
    2.00e9 Receives
      1.19e9 Late Senders
  524288 Collective
```

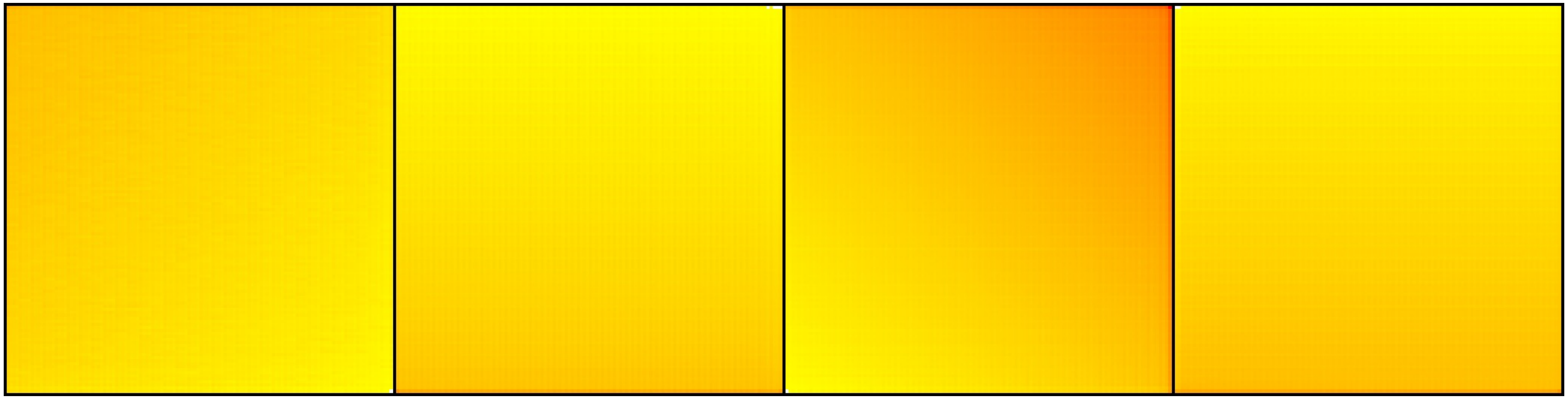
Call tree:

```
0.00 <its=1..7>
  0.00 source
  0.00 sweep
    6.67 <octant=1>
    0.74 <octant=2>
    5.31 <octant=3>
    0.59 <octant=4>
    10.05 <octant=5>
    0.82 <octant=6>
    5.78 <octant=7>
    1.41 <octant=8>
  0.00 global_int_sum
  0.00 flux_err
0.00 <its=8..12>
  0.00 source
  0.00 sweep
    11.56 <octant=1>
    3.69 <octant=2>
    11.36 <octant=3>
    3.54 <octant=4>
    19.66 <octant=5>
    3.74 <octant=6>
    11.52 <octant=7>
    3.57 <octant=8>
  0.00 global_int_sum
```

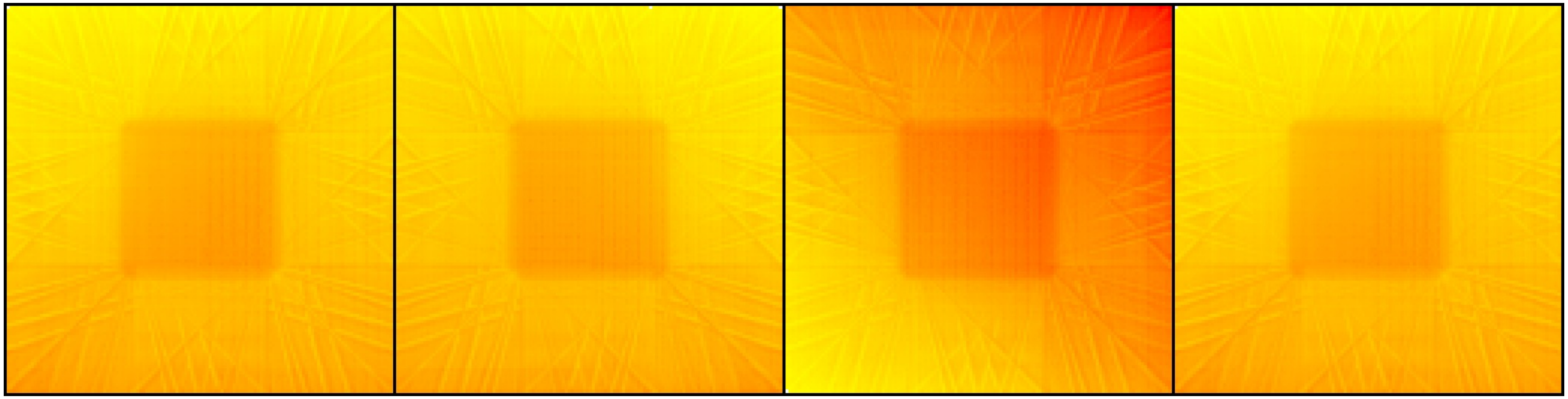
```
0.00    1.96e5 (12.15%)    1.61e6
0.00    68.64              100.00
0.00    1.35e5 (68.64%)    1.96e5
0.00    100.00             100.00
4.71    8.21 +/- 1.49      12.11
```

MPI *Late Sender* time distributions for octant pairs

Iterations without fixups:



Iterations with fixups:

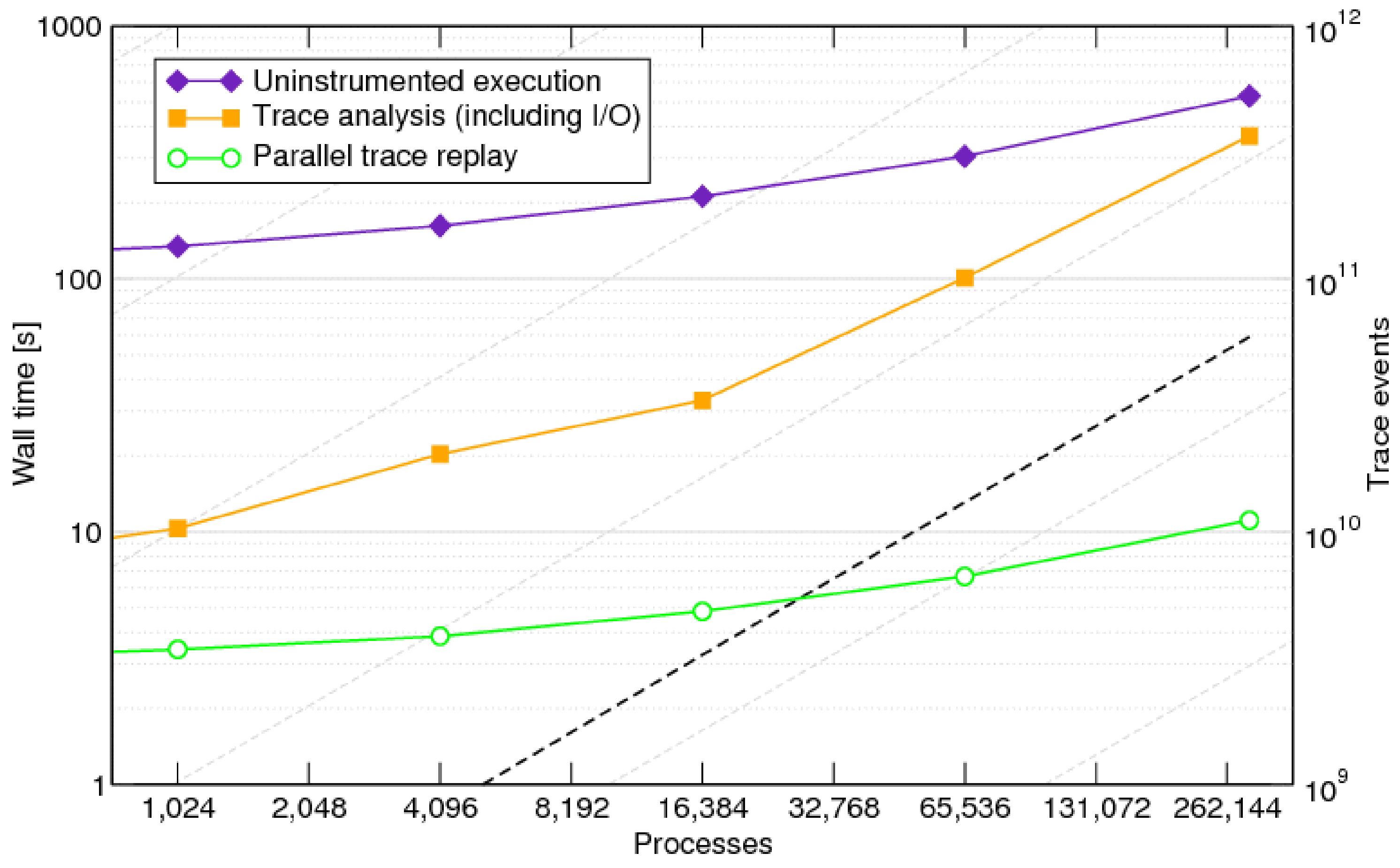


octant=1+2 octant=3+4 octant=5+6 octant=7+8

Scalasca experiment statistics: sweep3d_vn294912_trace

Scalasca	*v1.2*	*v1.3+*
Sweep3d size of *k*-blocks MK	10	1
Sweep3d elapsed time [s]	505	129
Measurement dilation [%]	5	3
Unification time [mins]	43	41
Trace event records [G]	59	510
Trace buffer content [MB]	2.75	27
Trace total size [TB]	0.79	7.6
Trace (physical) files [#]	294912	576
Trace open/create time [mins]	86	10
Trace flush rate [GB/s]	3.7	19.1
Trace analysis time [s]	368	546
- Trace analysis collation [s]	162	91
- Trace analysis replay [s]	11	74

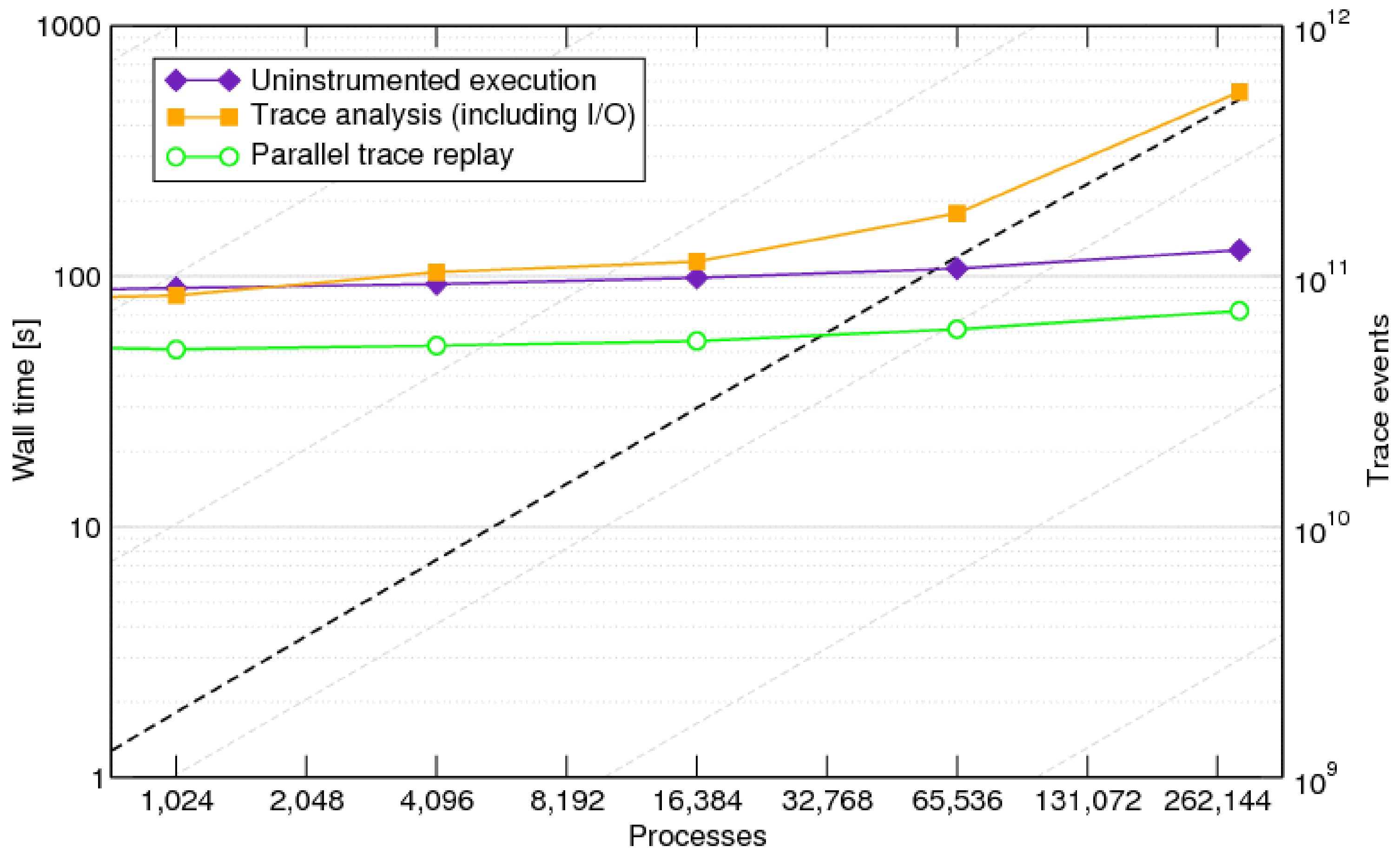
Scalasca trace analysis time scaling: original MK=10



- 2.75MB of trace event records per process (determined from summary)
- Total trace size (---) increases linearly to 790GB for 59G events
- Trace replay time scales with application execution time

[Scalasca v1.2]

Scalasca trace analysis time scaling: improved MK=1



[Scalasca v1.3 using SIONlib & prototype binary-format report]

- 10x larger traces due to 10x messages
- 27MB of trace event records per process
- Total trace size (---) increased to 7.6TB for 510G events
- 10x longer trace replay time still scales with application execution time

Scalasca scalability issues/optimizations

Time for unification of identifiers grows linearly with number of processes

- use *hierarchical* unification scheme

Creating individual traces files for each process is prohibitive

- use SIONlib *multi-files* (e.g., one per Blue Gene/P IONode)

Analysis reports are large and slow to collate

- *binary format* for metric value data plus *separate* XML metadata

Analysis presentation is slow and requires lots of memory

- store metrics as *inclusive* values and load them *incrementally*

Full analysis presentation requires very large, high-resolution screens

- :-)

Conclusions

Blue Gene/P efficiently supports application executions at extreme scale

Sweep3D is scalable to 294,912 MPI processes on Blue Gene/P

- appropriate input configuration/tuning is necessary
- 'fixups' are computationally imbalanced and amplify waiting times
  - *they therefore need to be used sparingly*
- there might be further opportunities for performance improvement
  - *e.g., with non-default mapping of processes to cores*

Scalasca has demonstrated performance measurement/analysis at unprecedented scale

- however, further significant improvements are in progress scalasca

Scalable performance **a**nalysis of *extremely* **la**rge-**sca**le parallel **a**pplications

- portable toolset for scalable performance measurement & analysis of MPI, OpenMP & hybrid OpenMP/MPI parallel applications
- supporting most popular HPC computer systems
- available under New BSD open-source license
- distributed on POINT/VI-HPS Parallel Productivity Tools Live-DVD
- sources, documentation & publications:
  - *http://www.scalasca.org*
  - *mailto: scalasca@fz-juelich.de*