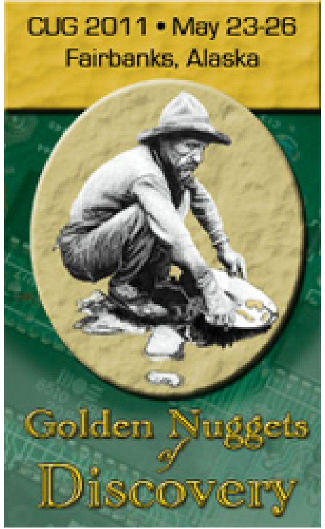
Mitglied der Helmholtz-Gemeinschaft



JÜLICH
FORSCHUNGSZENTRUM

Large-scale performance analysis of *PFLOTRAN* with Scalasca

2011-05-26 | **Brian J. N. Wylie** & Markus Geimer
Jülich Supercomputing Centre

b.wylie@fz-juelich.de

Overview

Dagstuhl challenge

- *PFLOTRAN* at large-scale on Cray XT5 & IBM BG/P

Scalasca performance analysis toolset

- Overview of architecture and usage
- Instrumentation of *PFLOTRAN* & PETSc
- Summary & trace measurement experiments
- Analysis report & trace exploration

Scalasca scalability improvement

- Unification of definition identifiers

Conclusions

Dagstuhl challenge

Demonstrate performance measurement and analysis of *PFLOTRAN* using “2B” problem dataset and more than 10,000 MPI processes

Challenge issued for Dagstuhl seminar 10181 (3-7 May 2010) on “Program development for extreme-scale computing”

- two months notice with download/build/run instructions
- accounts/allocations provided by VI-HPS & INCITE/PEAC
  - *IBM BG/P (**jugene**.fz-juelich.de)*
  - *Cray XT5 (**jaguar**.nccs.ornl.gov)*
- numerous research groups and vendors presented their results and discussed issues encountered
  - *http://www.dagstuhl.de/10181*

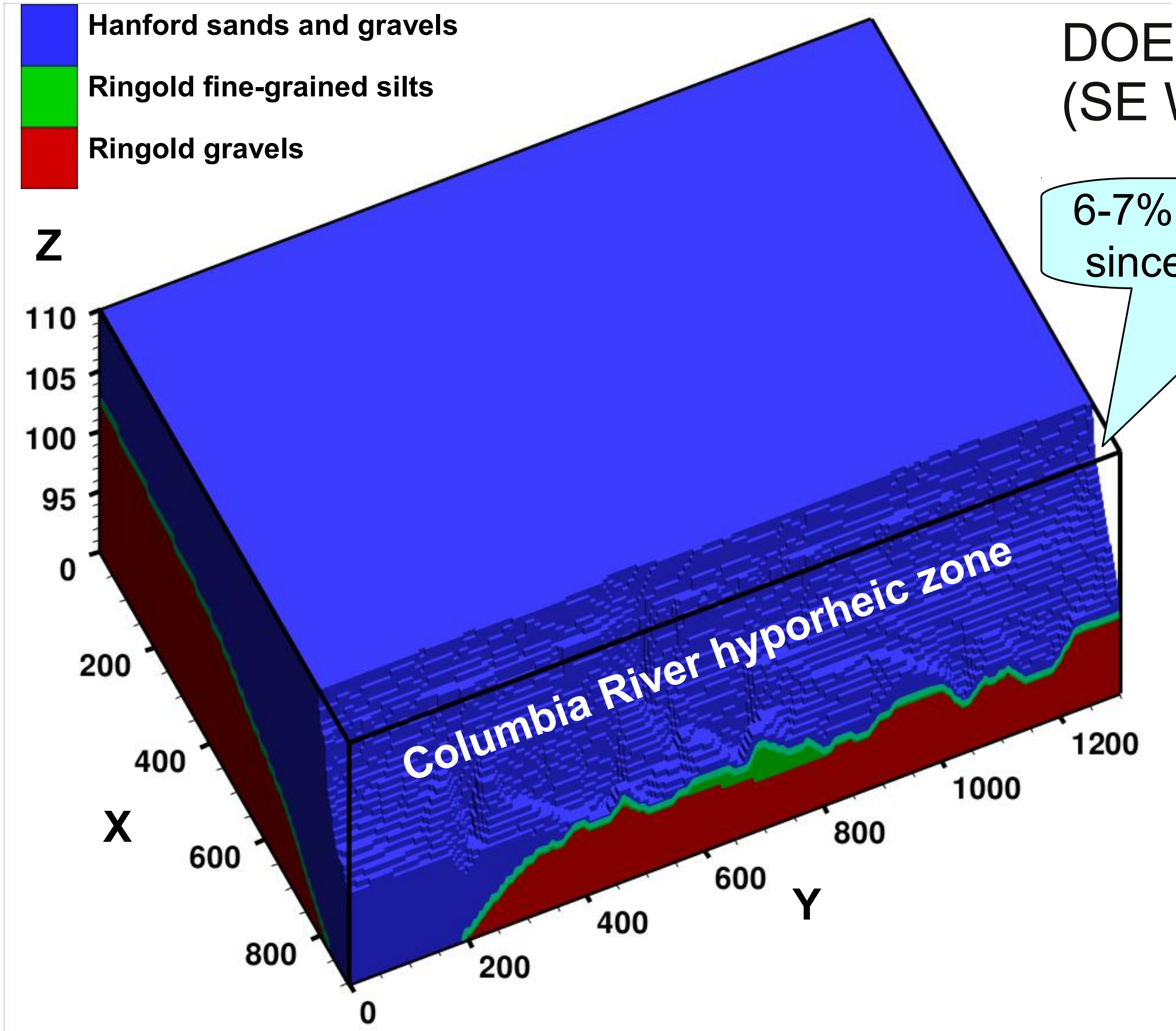
PFLOTRAN

3D reservoir simulator actively developed by LANL/ORNL/PNNL

- approx. 80,000 lines of Fortran9X, combining solvers for
  - *PFLOW non-isothermal, multi-phase groundwater flow*
  - *PTRAN reactive, multi-component contaminant transport*
- employs PETSc, LAPACK, BLAS & HDF5 I/O libraries
  - *87 PFLOTRAN + 789 PETSc source files*
  - *parallel I/O tuning via PERI active liaison*
- “2B” input dataset run for 10 timesteps
  - *uses 3-dimensional (non-MPI) PETSc Cartesian grid*
  - *TRAN(sport) step scales much better than FLOW step*
  - *FLOW step generally faster, but crossover at larger scales*

Courtesy of G. Hammond (PNNL)

PFLOTRAN “2B” test case


🟦 Hanford sands and gravels
🟩 Ringold fine-grained silts
🟥 Ringold gravels
Z
110
105
100
95
0
200
400
X
600
800
0
200
400
600
Y
800
1000
1200
Columbia River hyporheic zone

DOE Hanford 300 Area
(SE Washington, USA)

6-7% of grid cells inactive since within river channel

PFLOTRAN simulation of U(VI) migration

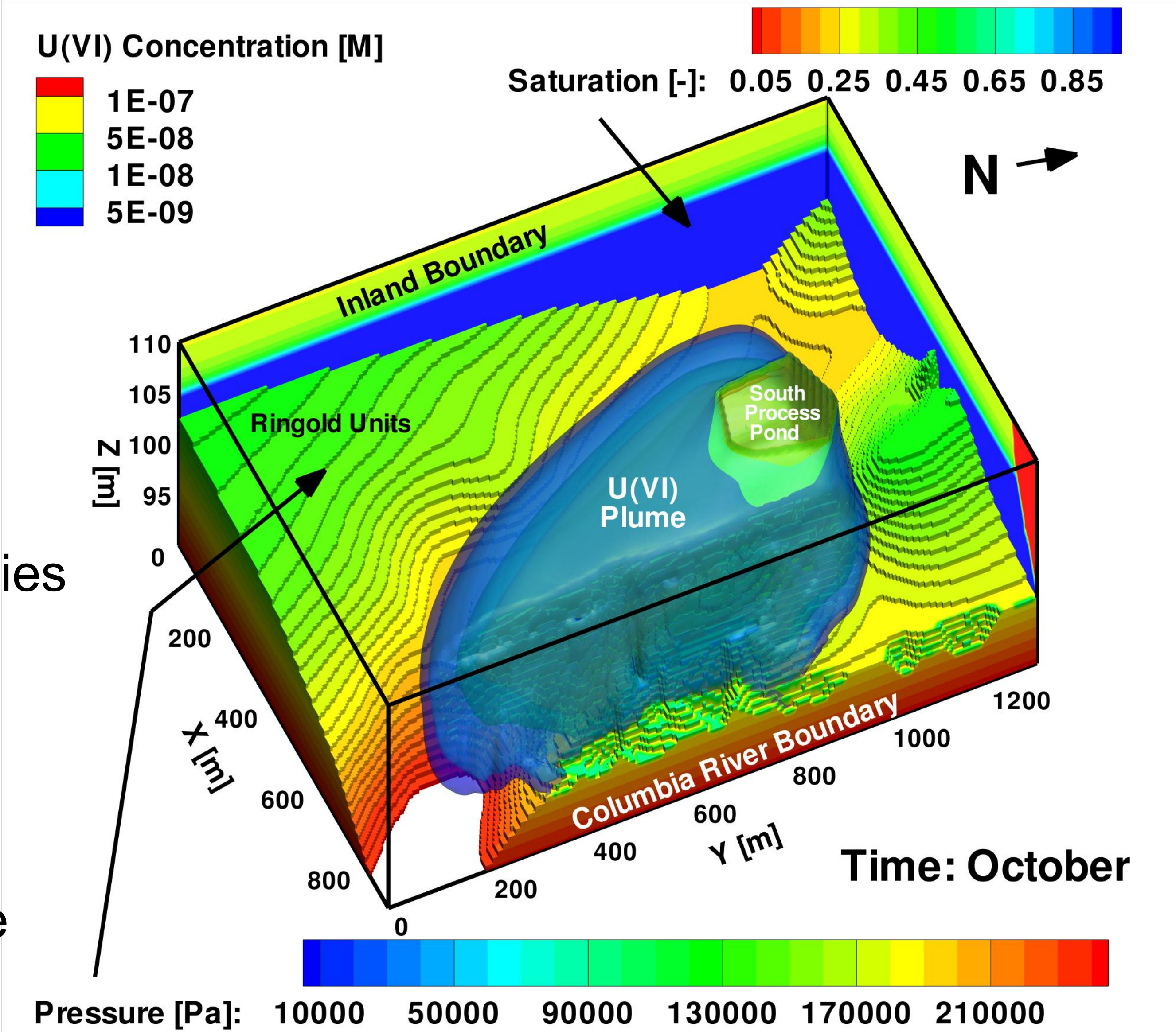
DOE Hanford 300 Area

Problem domain:

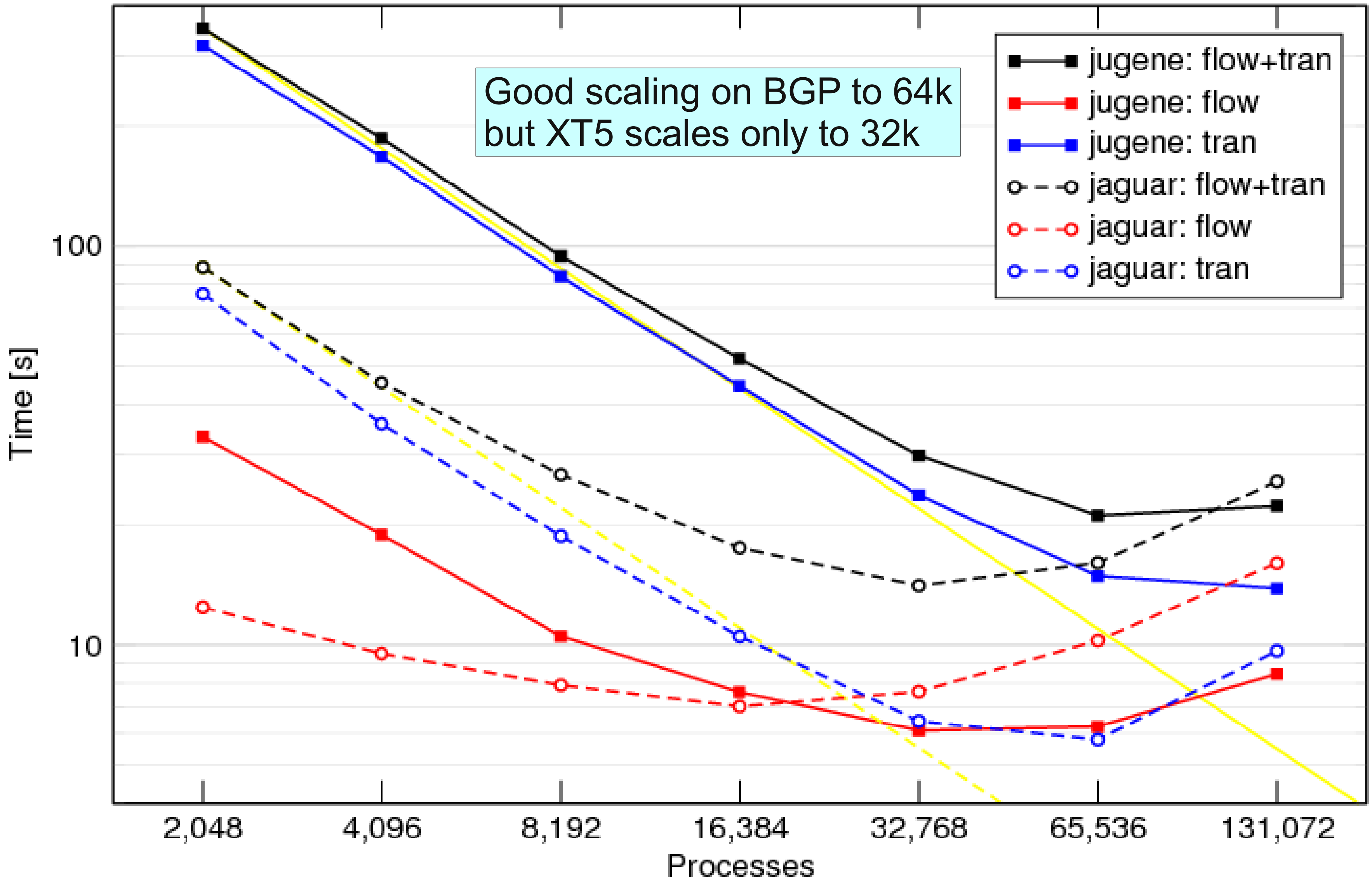
- 900x1300x20m
- Δx/Δy = 5 m
- 1.87M grid cells
- 15 chemical species
- 28M DoF total

1-year simulation:

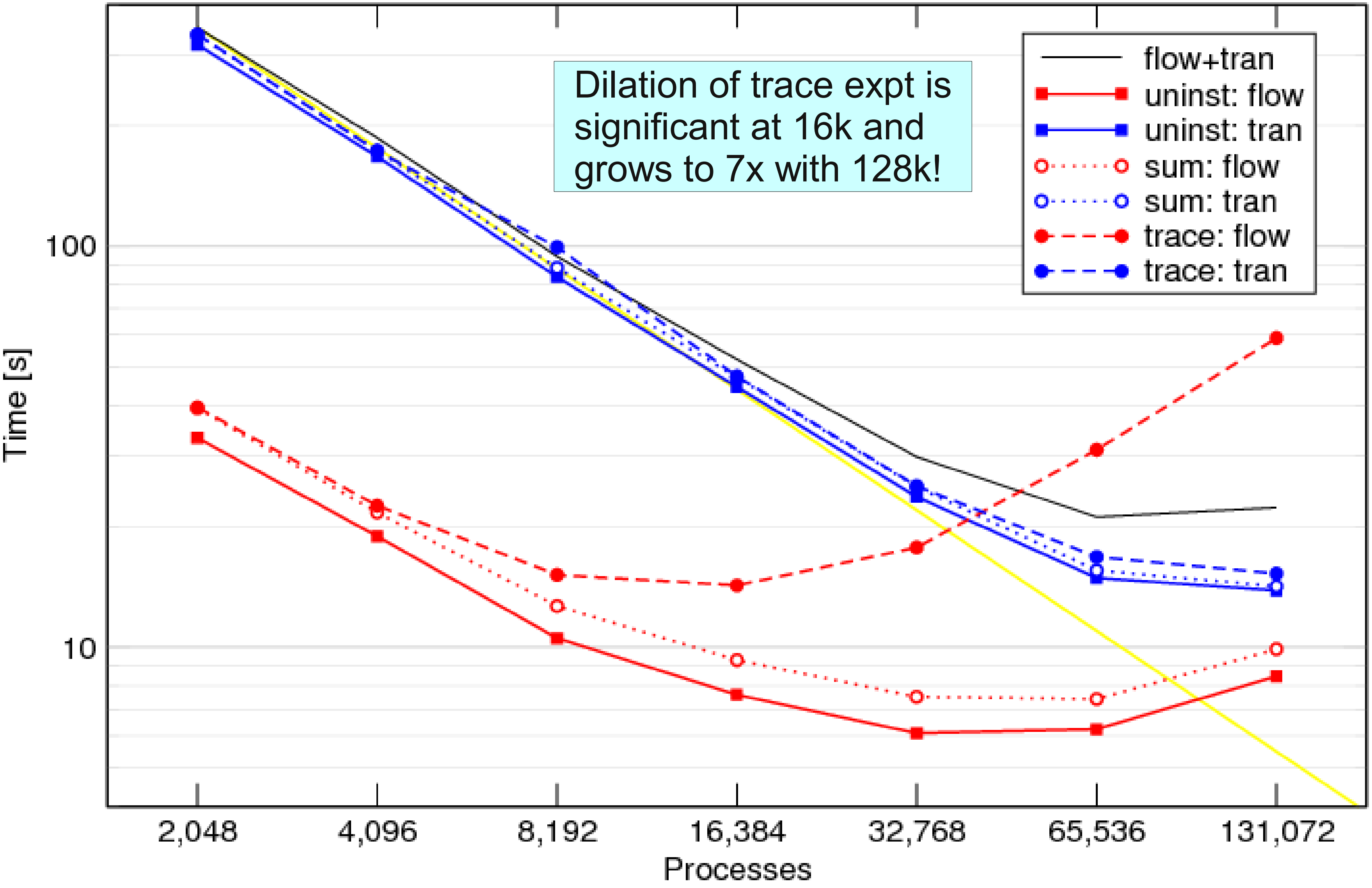
- Δt = 1 hour
- 5-10 hour runtime
- 4096 XT5 cores



PFLOTRAN “2B” strong scalability

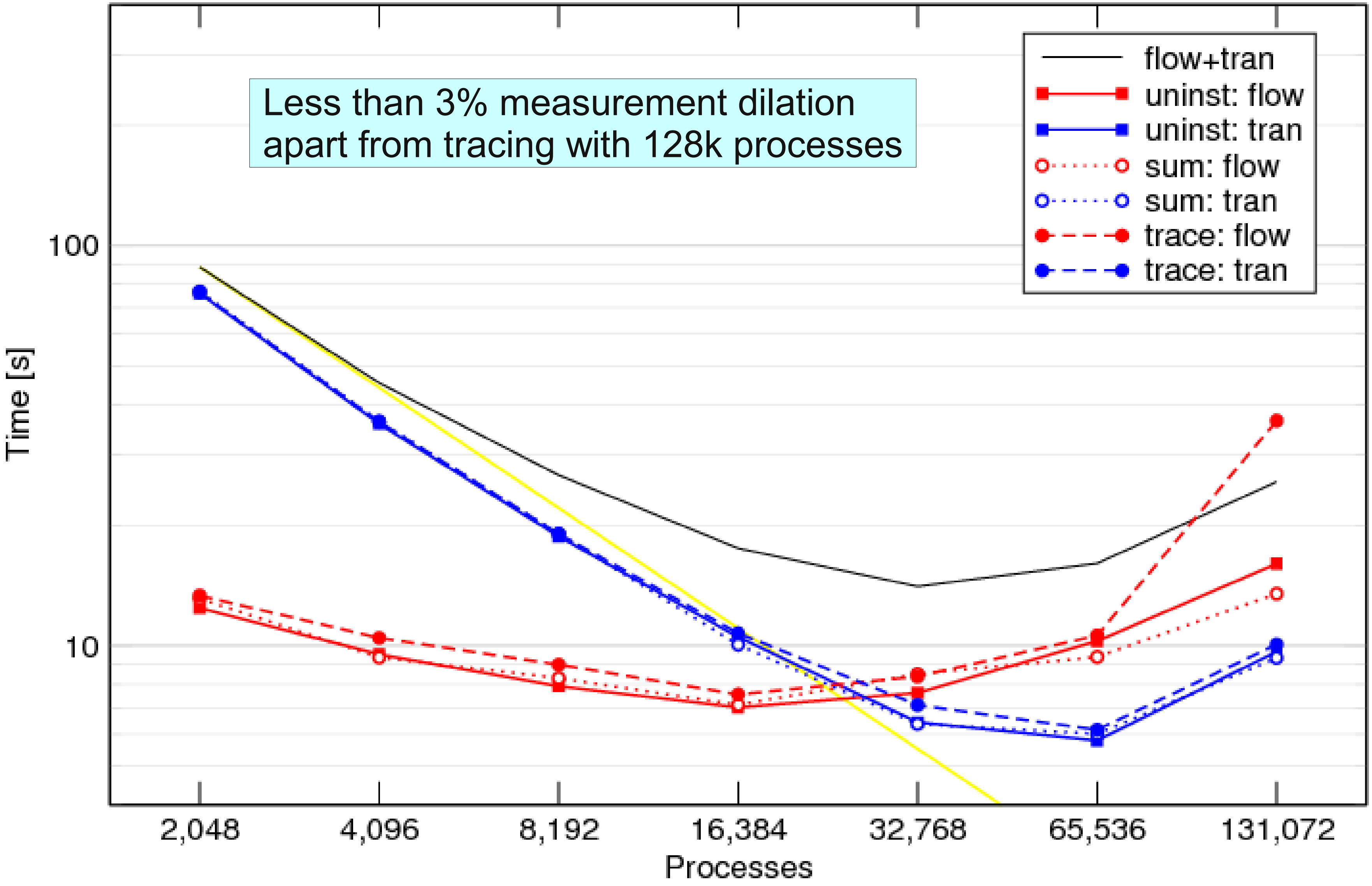


Scalasca *PFLOTRAN* “2B” measurement dilation (BGP)



Dilation of trace expt is significant at 16k and grows to 7x with 128k!

Scalasca *PFLOTRAN* “2B” measurement dilation (XT5)


Less than 3% measurement dilation apart from tracing with 128k processes

Scalasca project

Overview

- Headed by Bernd Mohr (JSC) & Felix Wolf (GRS-Sim)
  - *Helmholtz Initiative & Networking Fund project started in 2006*
- 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 v1.3.3 released in March 2011
- Available for download from www.scalasca.org

Scalasca features

Open source, New BSD license

Portable

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

Supports typical HPC languages & parallel programming paradigms

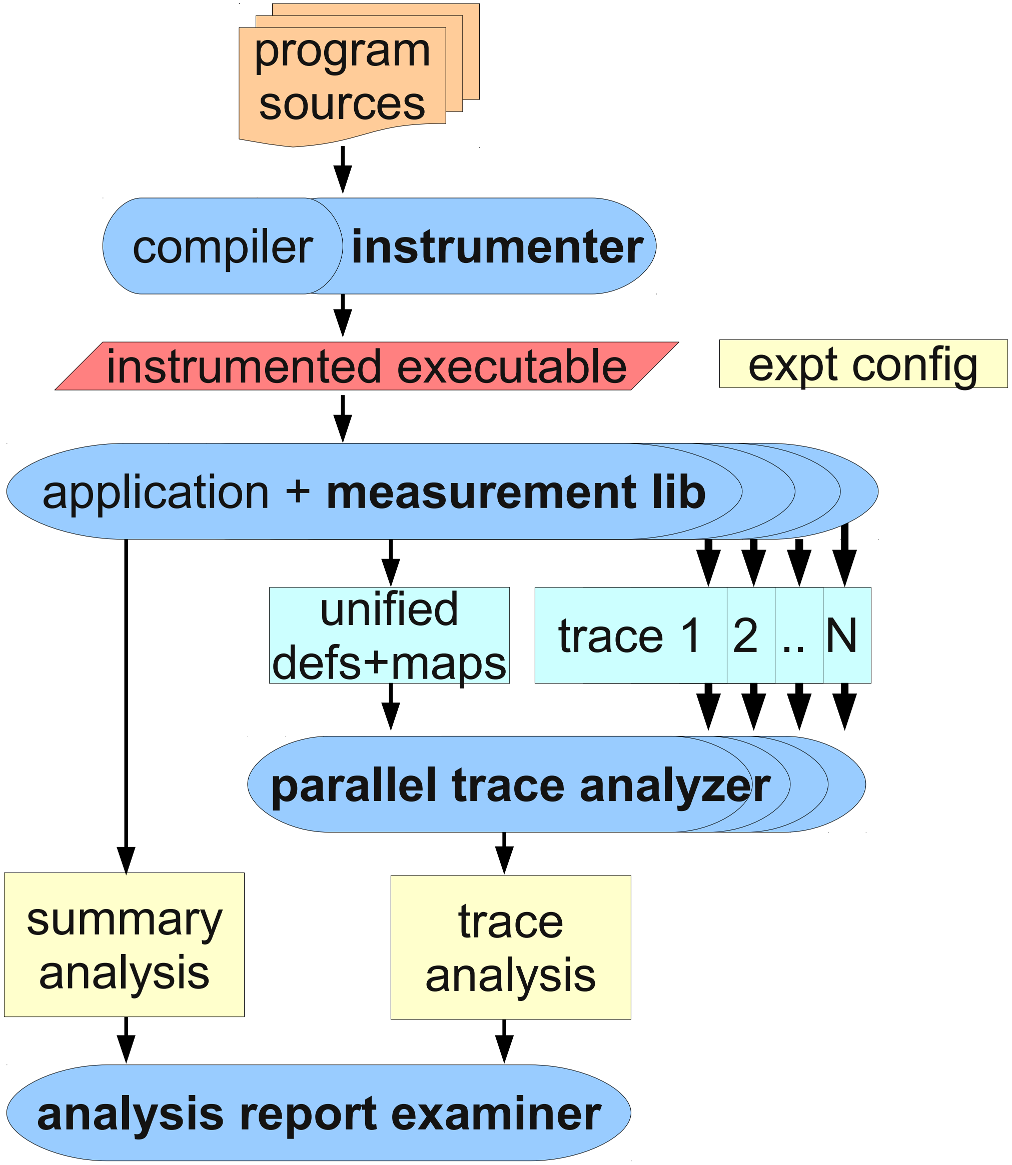
- Fortran, C, C++
- MPI, OpenMP & hybrid MPI/OpenMP

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 (commands)

1. Prepare application objects and executable for measurement:
   - **`scalasca -instrument`** `cc -O3 -c …`
   - **`scalasca -instrument`** `ftn -O3 -o pflotran.exe …`
     - *instrumented executable pflotran.exe produced*
2. Run application under control of measurement & analysis nexus:
   - **`scalasca -analyze`** `aprun -N 12 -n 65536 pflotran.exe …`
     - *epik_pflotran_12p65536_sum* experiment produced
   - **`scalasca -analyze -t`** `aprun -N 12 -n 65536 pflotran.exe …`
     - *epik_pflotran_12p65536_trace* experiment produced

   BATCH JOB
3. Interactively explore experiment analysis report:
   - **`scalasca -examine`** `epik_pflotran_12p65536_trace`
     - *epik_pflotran_12p65536_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 measurement 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

(optional) Customize analysis report and/or instrumentation, e.g.,

- extract key code sections into specific analysis reports
- annotate key code sections with *EPIK* instrumentation API macros

Scalasca usage with *PFLOTRAN*

Automatic instrumentation

- both *PFLOTRAN* application (Fortran) & PETSc library (C)
- USR routines instrumented by IBM XL & PGI compilers
- MPI routine interposition with instrumented library (PMPI)

Initial summary measurements used to define filter files specifying all purely computational routines

Summary & trace experiments collected using filters

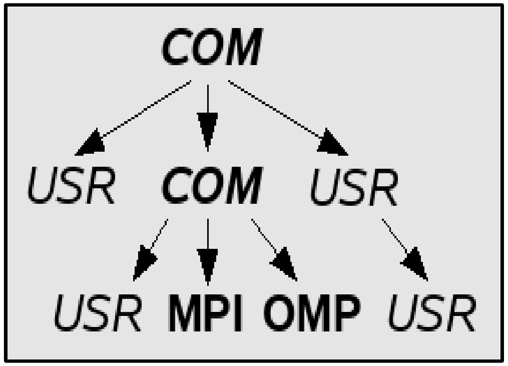
- parallel trace analysis initiated automatically on same partition

Post-processing of analysis reports

- cut to extract timestep loop; incorporation of application topology

Analysis report examination in GUI

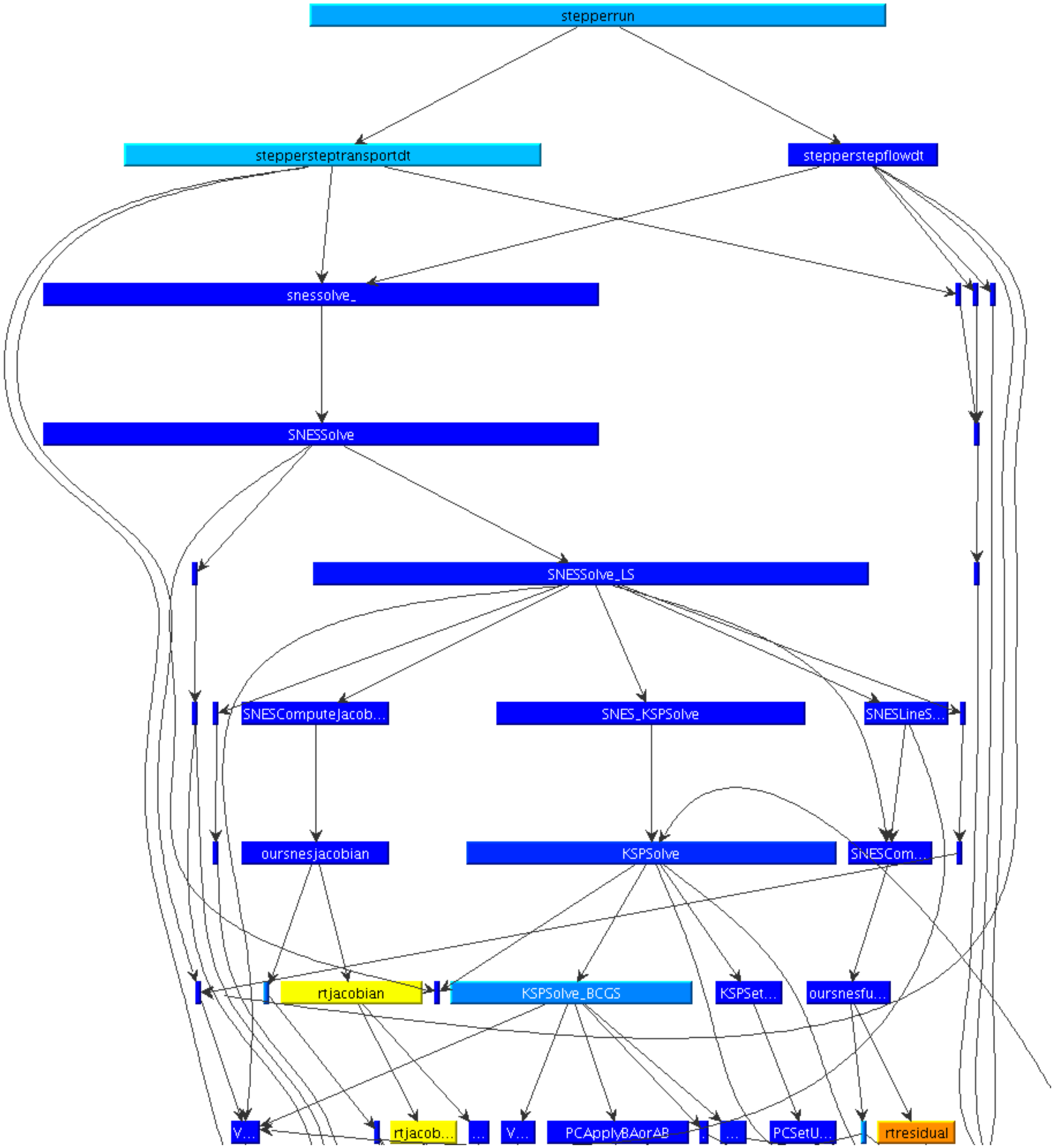
PFLOTRAN structural analysis

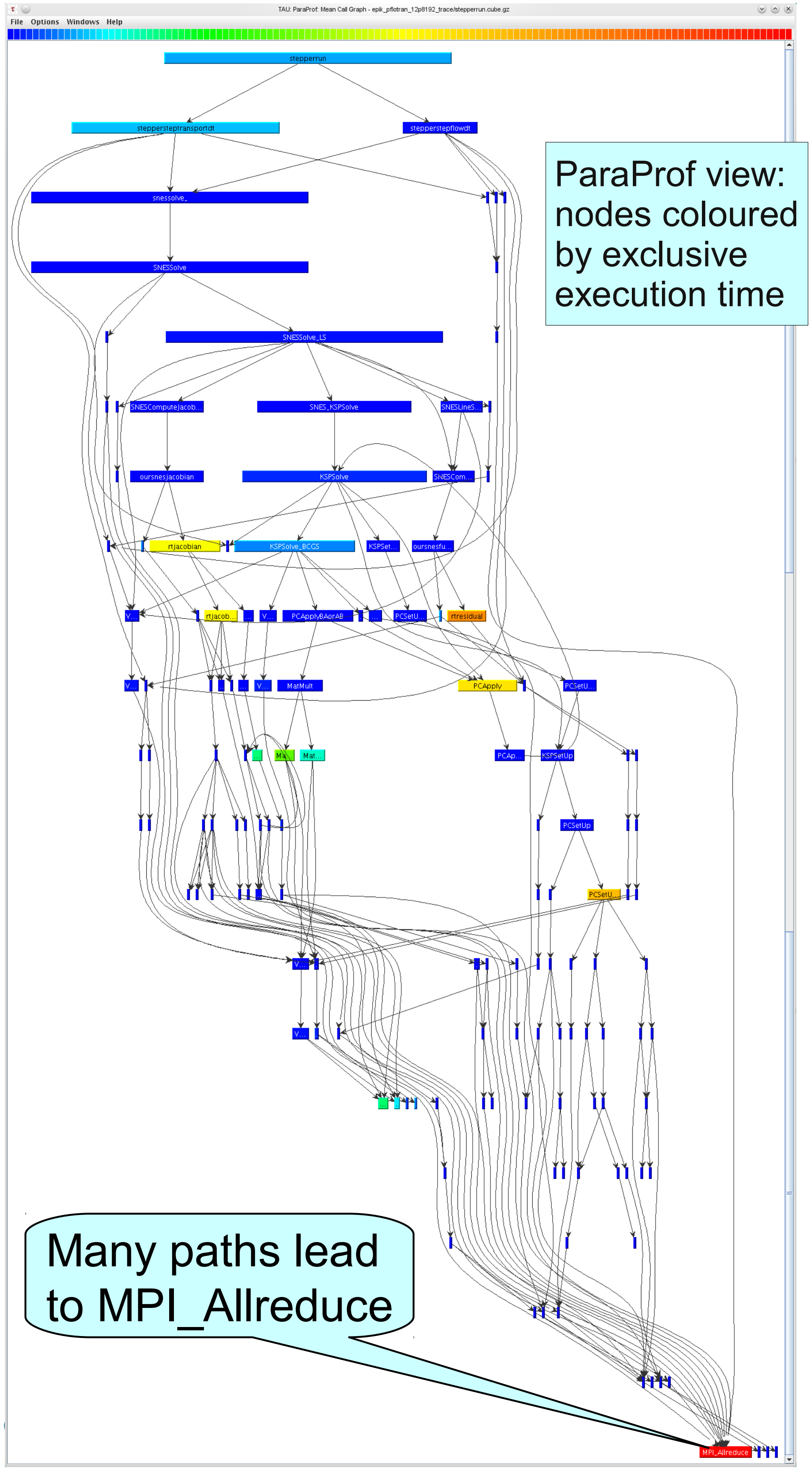


Determined by scoring summary experiment using fully-instrumented application executable

- 29 MPI library routines used
- 1100+ PFLOTRAN & PETSc routines
  - *most not on a callpath to MPI, purely local calculation (USR)*
  - *~250 on callpaths to MPI, mixed calculation & comm. (COM)*
- Using measurement filter listing all USR routines
  - *maximum callpath depth 22 frames*
  - *~1750 unique callpaths (399 in FLOW, 375 in TRAN)*
  - *633 MPI callpaths (121 in FLOW, 114 in TRAN)*
- FLOW callpaths very similar to TRAN callpaths

PFLOTRAN stepperrun callgraph

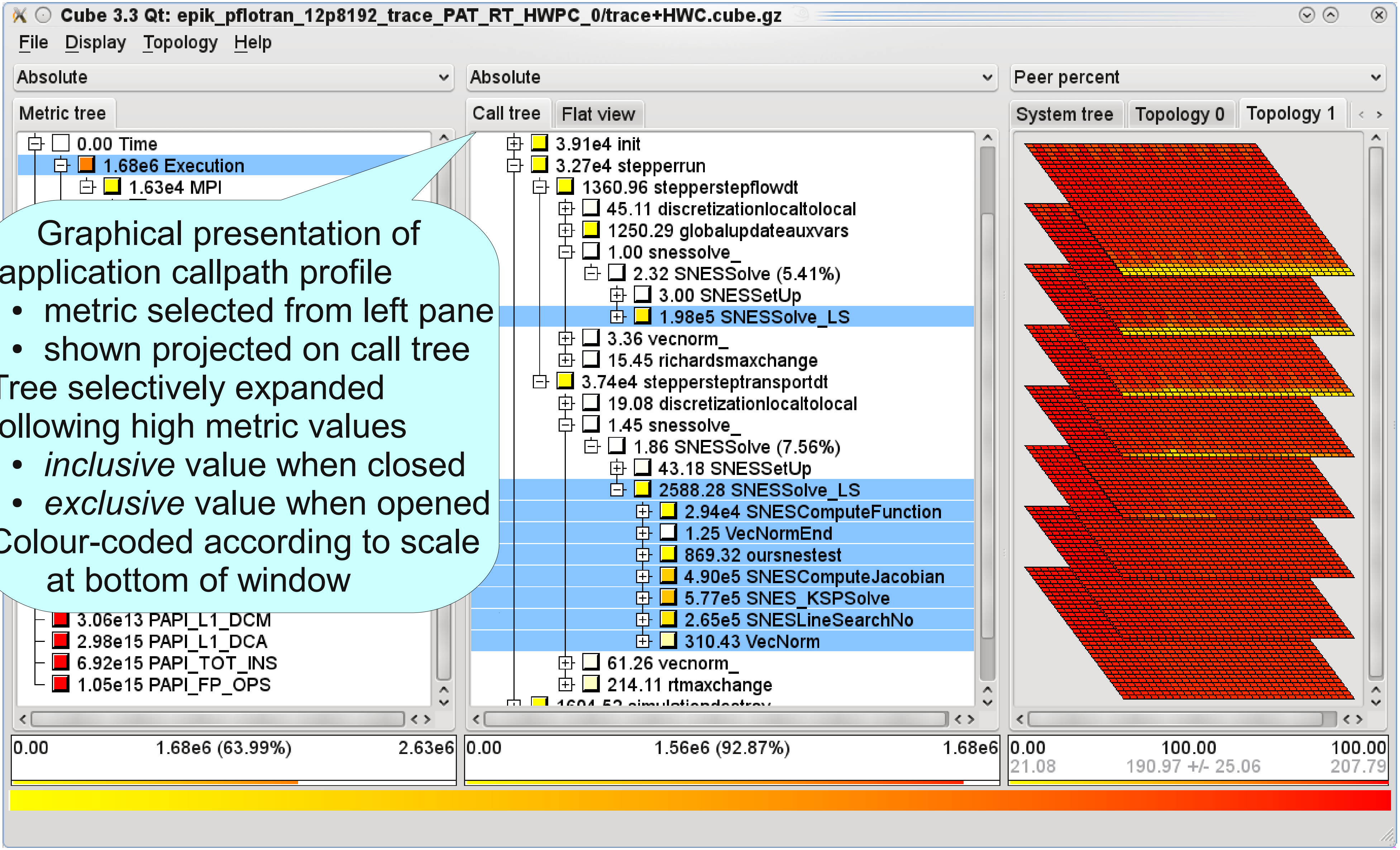




ParaProf view: nodes coloured by exclusive execution time

Many paths lead to MPI_Allreduce

Scalasca analysis report: program call tree



Cube 3.3 Qt: epik_pflotran_12p8192_trace_PAT_RT_HWPC_0/trace+HWC.cube.gz

```
0.00 Time
  1.68e6 Execution
    1.63e4 MPI
3.06e13 PAPI_L1_DCM
2.98e15 PAPI_L1_DCA
6.92e15 PAPI_TOT_INS
1.05e15 PAPI_FP_OPS
```

```
3.91e4 init
3.27e4 stepperrun
  1360.96 stepperstepflowdt
    45.11 discretizationlocaltolocal
    1250.29 globalupdateauxvars
    1.00 snessolve_
      2.32 SNESSolve (5.41%)
        3.00 SNESSetUp
        1.98e5 SNESSolve_LS
    3.36 vecnorm_
    15.45 richardsmaxchange
  3.74e4 steppersteptransportdt
    19.08 discretizationlocaltolocal
    1.45 snessolve_
      1.86 SNESSolve (7.56%)
        43.18 SNESSetUp
        2588.28 SNESSolve_LS
          2.94e4 SNESComputeFunction
          1.25 VecNormEnd
          869.32 oursnestest
          4.90e5 SNESComputeJacobian
          5.77e5 SNES_KSPSolve
          2.65e5 SNESLineSearchNo
          310.43 VecNorm
    61.26 vecnorm_
    214.11 rtmaxchange
1604.52 simulationdestroy
```

0.00	1.68e6 (63.99%)	2.63e6
0.00	1.56e6 (92.87%)	1.68e6
0.00 21.08	100.00 190.97 +/- 25.06	100.00 207.79

Graphical presentation of application callpath profile

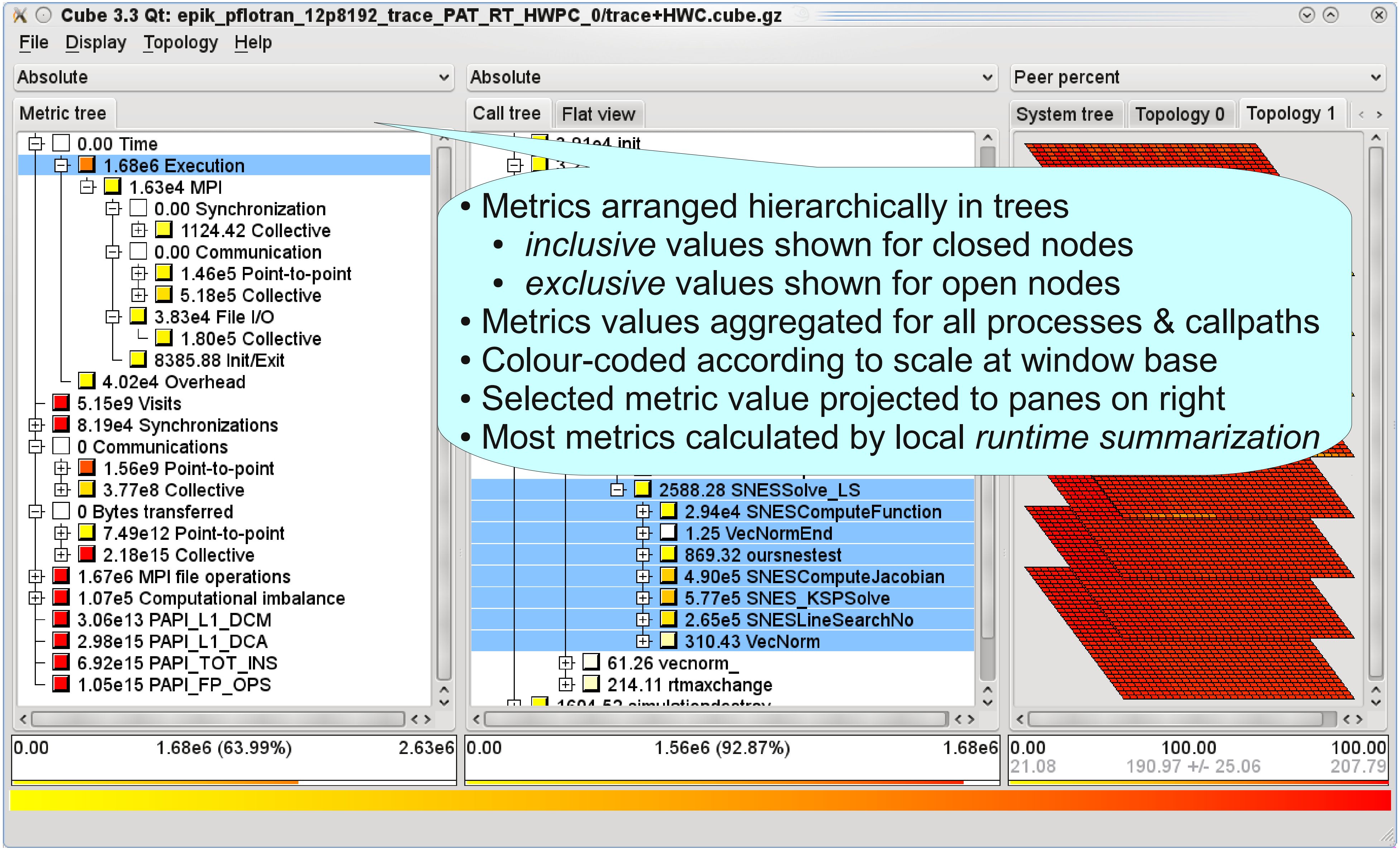
- metric selected from left pane
- shown projected on call tree

Tree selectively expanded following high metric values

- *inclusive* value when closed
- *exclusive* value when opened

Colour-coded according to scale at bottom of window

Scalasca analysis report: performance metric tree



Cube 3.3 Qt: epik_pflotran_12p8192_trace_PAT_RT_HWPC_0/trace+HWC.cube.gz

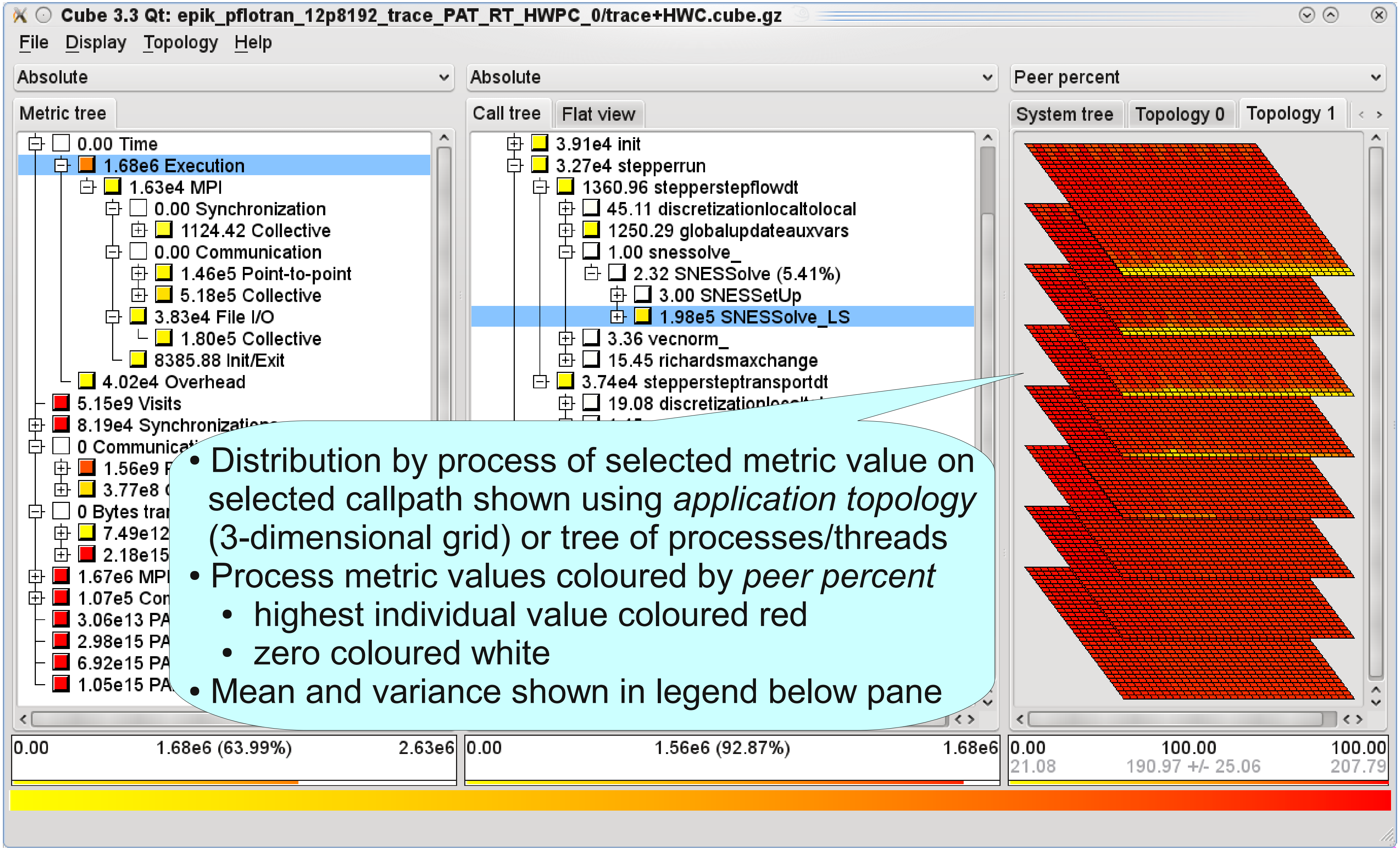
```
⊟ □ 0.00 Time
    ⊟ 1.68e6 Execution
        ⊟ 1.63e4 MPI
            ⊟ □ 0.00 Synchronization
                ⊞ 1124.42 Collective
            ⊟ □ 0.00 Communication
                ⊞ 1.46e5 Point-to-point
                ⊞ 5.18e5 Collective
            ⊟ 3.83e4 File I/O
                1.80e5 Collective
            8385.88 Init/Exit
        4.02e4 Overhead
    5.15e9 Visits
⊞ 8.19e4 Synchronizations
⊟ □ 0 Communications
    ⊞ 1.56e9 Point-to-point
    ⊞ 3.77e8 Collective
⊟ □ 0 Bytes transferred
    ⊞ 7.49e12 Point-to-point
    ⊞ 2.18e15 Collective
⊞ 1.67e6 MPI file operations
⊞ 1.07e5 Computational imbalance
  3.06e13 PAPI_L1_DCM
  2.98e15 PAPI_L1_DCA
  6.92e15 PAPI_TOT_INS
  1.05e15 PAPI_FP_OPS
```

```
    3.01e4 init
⊟ 3
            ⊟ 2588.28 SNESSolve_LS
                ⊞ 2.94e4 SNESComputeFunction
                ⊞ □ 1.25 VecNormEnd
                ⊞ 869.32 oursnestest
                ⊞ 4.90e5 SNESComputeJacobian
                ⊞ 5.77e5 SNES_KSPSolve
                ⊞ 2.65e5 SNESLineSearchNo
                ⊞ 310.43 VecNorm
        ⊞ □ 61.26 vecnorm_
        ⊞ 214.11 rtmaxchange
⊞ 1604.52 simulationdestroy
```

```
0.00        1.68e6 (63.99%)        2.63e6
0.00        1.56e6 (92.87%)        1.68e6
0.00        100.00                 100.00
21.08       190.97 +/- 25.06       207.79
```

- 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*

Scalasca analysis report: system tree/topology



Absolute | Absolute | Peer percent

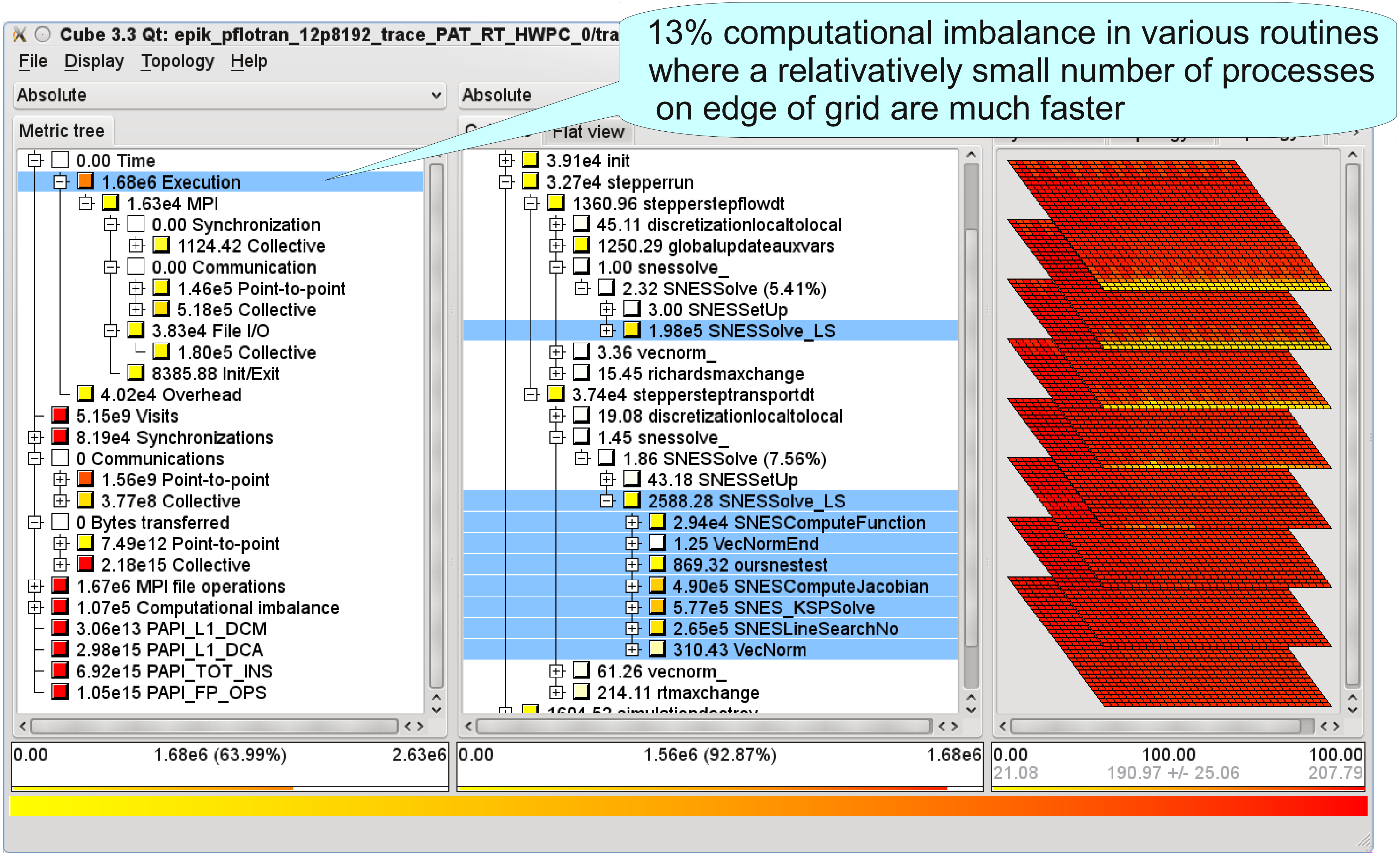
```
Metric tree
0.00 Time
  1.68e6 Execution
    1.63e4 MPI
      0.00 Synchronization
        1124.42 Collective
      0.00 Communication
        1.46e5 Point-to-point
        5.18e5 Collective
      3.83e4 File I/O
        1.80e5 Collective
      8385.88 Init/Exit
  4.02e4 Overhead
5.15e9 Visits
8.19e4 Synchronizatio
0 Communicat
  1.56e9 F
  3.77e8 (
0 Bytes trar
  7.49e12
  2.18e15
1.67e6 MPI
1.07e5 Cor
3.06e13 PA
2.98e15 PA
6.92e15 PA
1.05e15 PA
```

```
Call tree
3.91e4 init
3.27e4 stepperrun
  1360.96 stepperstepflowdt
    45.11 discretizationlocaltolocal
    1250.29 globalupdateauxvars
    1.00 snessolve_
      2.32 SNESSolve (5.41%)
        3.00 SNESSetUp
        1.98e5 SNESSolve_LS
    3.36 vecnorm_
    15.45 richardsmaxchange
  3.74e4 steppersteptransportdt
    19.08 discretizationlocalt
```

```
0.00        1.68e6 (63.99%)        2.63e6
0.00        1.56e6 (92.87%)        1.68e6
0.00        100.00                 100.00
21.08       190.97 +/- 25.06       207.79
```

- Distribution by process of selected metric value on selected callpath shown using *application topology* (3-dimensional grid) or tree of processes/threads
- Process metric values coloured by *peer percent*
  - highest individual value coloured red
  - zero coloured white
- Mean and variance shown in legend below pane

Exclusive Execution time



Cube 3.3 Qt: epik_pflotran_12p8192_trace_PAT_RT_HWPC_0/tra

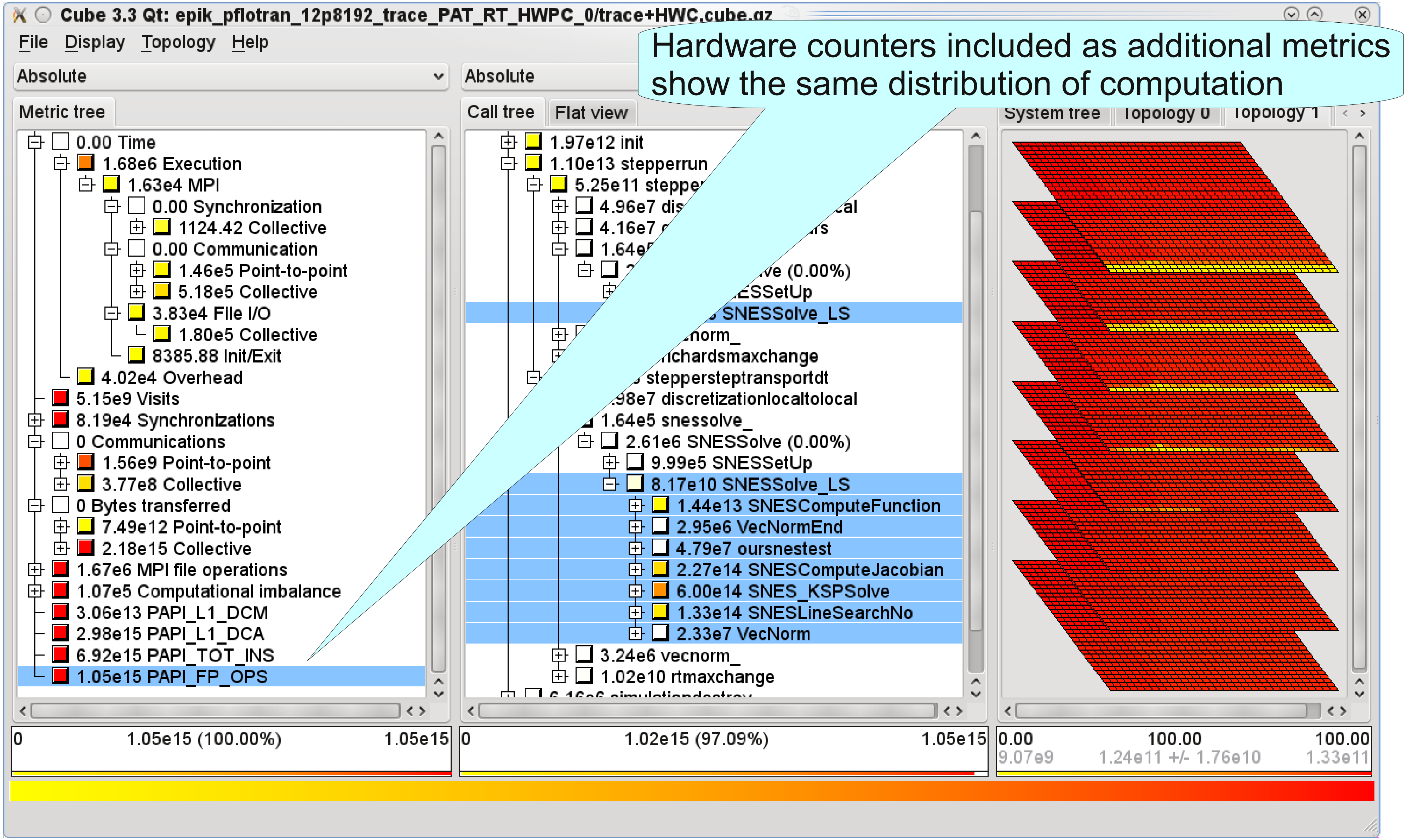
```
⊟ □ 0.00 Time
   ⊟ 🟧 1.68e6 Execution
      ⊟ 🟨 1.63e4 MPI
         ⊟ □ 0.00 Synchronization
            ⊞ 🟨 1124.42 Collective
         ⊟ □ 0.00 Communication
            ⊞ 🟨 1.46e5 Point-to-point
            ⊞ 🟨 5.18e5 Collective
         ⊟ 🟨 3.83e4 File I/O
            └ 🟨 1.80e5 Collective
         🟨 8385.88 Init/Exit
      🟨 4.02e4 Overhead
   🟥 5.15e9 Visits
⊞ 🟥 8.19e4 Synchronizations
⊟ □ 0 Communications
   ⊞ 🟧 1.56e9 Point-to-point
   ⊞ 🟨 3.77e8 Collective
⊟ □ 0 Bytes transferred
   ⊞ 🟨 7.49e12 Point-to-point
   ⊞ 🟥 2.18e15 Collective
⊞ 🟥 1.67e6 MPI file operations
⊞ 🟥 1.07e5 Computational imbalance
   🟥 3.06e13 PAPI_L1_DCM
   🟥 2.98e15 PAPI_L1_DCA
   🟥 6.92e15 PAPI_TOT_INS
   🟥 1.05e15 PAPI_FP_OPS
```

```
⊞ 🟨 3.91e4 init
⊟ 🟨 3.27e4 stepperrun
   ⊟ 🟨 1360.96 stepperstepflowdt
      ⊞ □ 45.11 discretizationlocaltolocal
      ⊞ 🟨 1250.29 globalupdateauxvars
      ⊟ □ 1.00 snessolve_
         ⊟ □ 2.32 SNESSolve (5.41%)
            ⊞ □ 3.00 SNESSetUp
            ⊞ 🟨 1.98e5 SNESSolve_LS
      ⊞ □ 3.36 vecnorm_
      ⊞ □ 15.45 richardsmaxchange
   ⊟ 🟨 3.74e4 steppersteptransportdt
      ⊞ □ 19.08 discretizationlocaltolocal
      ⊟ □ 1.45 snessolve_
         ⊟ □ 1.86 SNESSolve (7.56%)
            ⊞ □ 43.18 SNESSetUp
            ⊟ 🟨 2588.28 SNESSolve_LS
               ⊞ 🟨 2.94e4 SNESComputeFunction
               ⊞ □ 1.25 VecNormEnd
               ⊞ 🟨 869.32 oursnestest
               ⊞ 🟨 4.90e5 SNESComputeJacobian
               ⊞ 🟨 5.77e5 SNES_KSPSolve
               ⊞ 🟨 2.65e5 SNESLineSearchNo
               ⊞ □ 310.43 VecNorm
      ⊞ □ 61.26 vecnorm_
      ⊞ □ 214.11 rtmaxchange
⊞ 🟨 1604.52 simulationdestroy
```

```
0.00          1.68e6 (63.99%)          2.63e6
0.00          1.56e6 (92.87%)          1.68e6
0.00          100.00                   100.00
21.08         190.97 +/- 25.06         207.79
```

13% computational imbalance in various routines where a relativatively small number of processes on edge of grid are much faster

Floating-point operations



Hardware counters included as additional metrics show the same distribution of computation

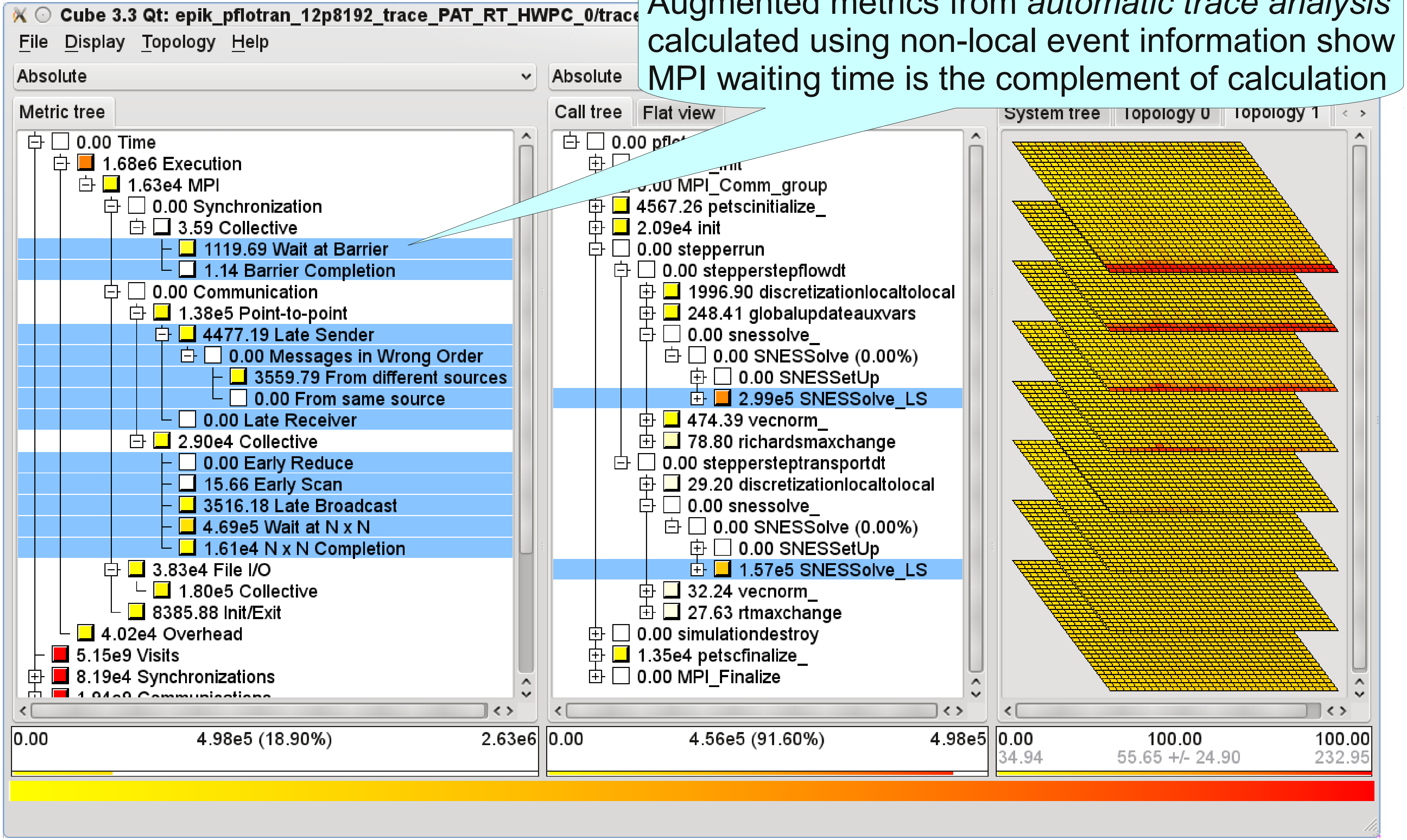
Cube 3.3 Qt: epik_pflotran_12p8192_trace_PAT_RT_HWPC_0/trace+HWC.cube.gz
File Display Topology Help
Absolute
Absolute
Metric tree
Call tree
Flat view
System tree
Topology 0
Topology 1

```
⊟ □ 0.00 Time
   ⊟ 🟧 1.68e6 Execution
      ⊟ 🟨 1.63e4 MPI
         ⊟ □ 0.00 Synchronization
            ⊞ 🟨 1124.42 Collective
         ⊟ □ 0.00 Communication
            ⊞ 🟨 1.46e5 Point-to-point
            ⊞ 🟨 5.18e5 Collective
         ⊟ 🟨 3.83e4 File I/O
            └ 🟨 1.80e5 Collective
         🟨 8385.88 Init/Exit
      🟨 4.02e4 Overhead
🟥 5.15e9 Visits
⊞ 🟥 8.19e4 Synchronizations
⊟ □ 0 Communications
   ⊞ 🟧 1.56e9 Point-to-point
   ⊞ 🟨 3.77e8 Collective
⊟ □ 0 Bytes transferred
   ⊞ 🟨 7.49e12 Point-to-point
   ⊞ 🟥 2.18e15 Collective
⊞ 🟥 1.67e6 MPI file operations
⊞ 🟥 1.07e5 Computational imbalance
🟥 3.06e13 PAPI_L1_DCM
🟥 2.98e15 PAPI_L1_DCA
🟥 6.92e15 PAPI_TOT_INS
🟥 1.05e15 PAPI_FP_OPS
```

```
⊞ 🟨 1.97e12 init
⊟ 🟨 1.10e13 stepperrun
   ⊟ 🟨 5.25e11 steppe
      ⊞ □ 4.96e7 dis                al
      ⊞ □ 4.16e7 c                  rs
      ⊟ □ 1.64e
         ⊟ □ 2                  ve (0.00%)
            ⊞                ESSetUp
                           SNESSolve_LS
      ⊞               norm_
      ⊞               ichardsmaxchange
   ⊟                  steppersteptransportdt
                   98e7 discretizationlocaltolocal
         □ 1.64e5 snessolve_
         ⊟ □ 2.61e6 SNESSolve (0.00%)
            ⊞ □ 9.99e5 SNESSetUp
            ⊟ □ 8.17e10 SNESSolve_LS
               ⊞ 🟨 1.44e13 SNESComputeFunction
               ⊞ □ 2.95e6 VecNormEnd
               ⊞ □ 4.79e7 oursnestest
               ⊞ 🟨 2.27e14 SNESComputeJacobian
               ⊞ 🟧 6.00e14 SNES_KSPSolve
               ⊞ 🟨 1.33e14 SNESLineSearchNo
               ⊞ □ 2.33e7 VecNorm
      ⊞ □ 3.24e6 vecnorm_
      ⊞ □ 1.02e10 rtmaxchange
```

```
0        1.05e15 (100.00%)       1.05e15
0        1.02e15 (97.09%)        1.05e15
0.00     100.00                  100.00
9.07e9   1.24e11 +/- 1.76e10     1.33e11
```

Scalasca trace analysis metrics (waiting time)



Cube 3.3 Qt: epik_pflotran_12p8192_trace_PAT_RT_HWPC_0/trace

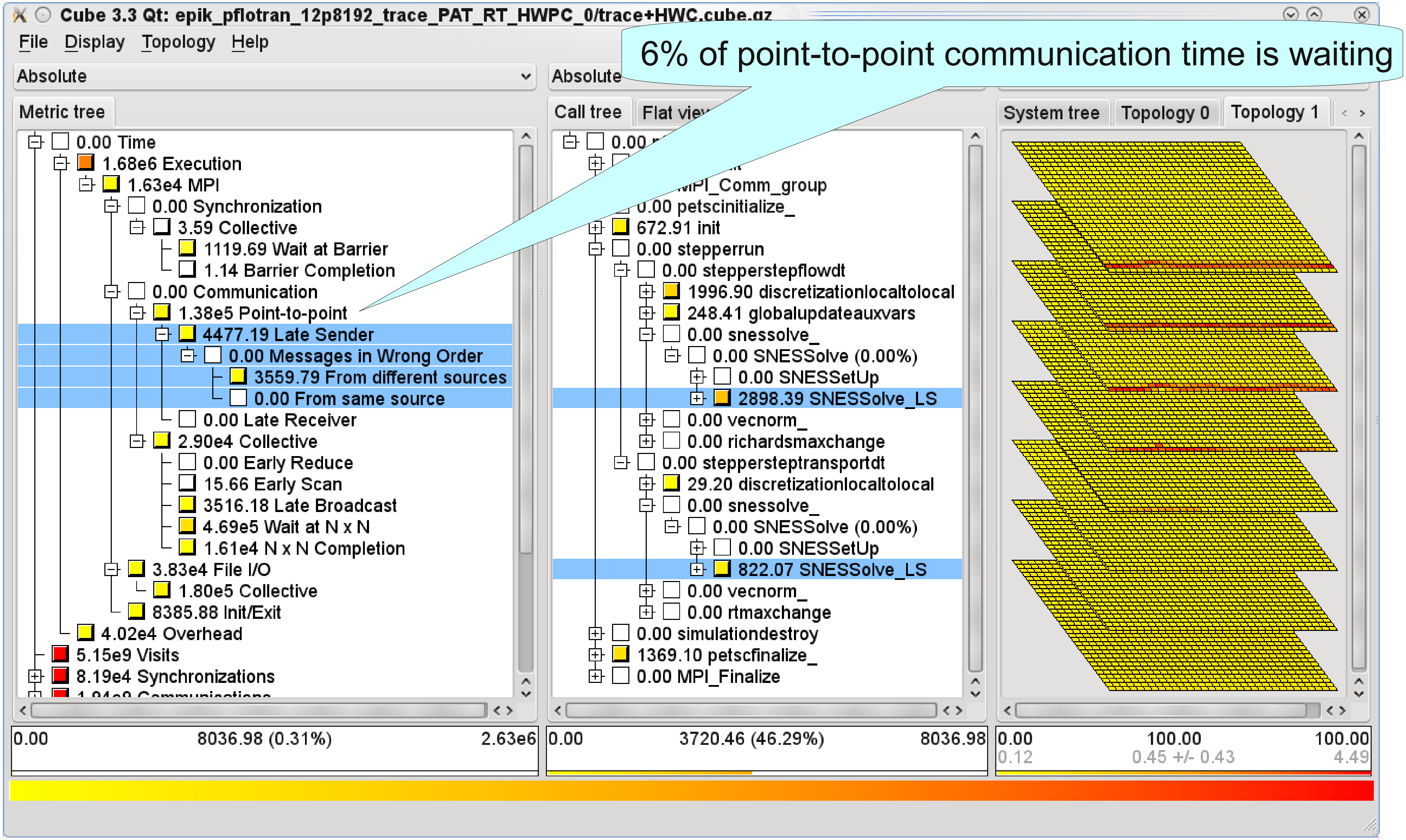
```
0.00 Time
  1.68e6 Execution
    1.63e4 MPI
      0.00 Synchronization
        3.59 Collective
          1119.69 Wait at Barrier
          1.14 Barrier Completion
      0.00 Communication
        1.38e5 Point-to-point
          4477.19 Late Sender
            0.00 Messages in Wrong Order
              3559.79 From different sources
              0.00 From same source
          0.00 Late Receiver
        2.90e4 Collective
          0.00 Early Reduce
          15.66 Early Scan
          3516.18 Late Broadcast
          4.69e5 Wait at N x N
          1.61e4 N x N Completion
      3.83e4 File I/O
        1.80e5 Collective
      8385.88 Init/Exit
    4.02e4 Overhead
5.15e9 Visits
8.19e4 Synchronizations
```

```
0.00 pflot
  _init
  0.00 MPI_Comm_group
  4567.26 petscinitialize_
  2.09e4 init
  0.00 stepperrun
    0.00 stepperstepflowdt
      1996.90 discretizationlocaltolocal
      248.41 globalupdateauxvars
      0.00 snessolve_
        0.00 SNESSolve (0.00%)
          0.00 SNESSetUp
          2.99e5 SNESSolve_LS
      474.39 vecnorm_
      78.80 richardsmaxchange
    0.00 steppersteptransportdt
      29.20 discretizationlocaltolocal
      0.00 snessolve_
        0.00 SNESSolve (0.00%)
          0.00 SNESSetUp
          1.57e5 SNESSolve_LS
      32.24 vecnorm_
      27.63 rtmaxchange
  0.00 simulationdestroy
  1.35e4 petscfinalize_
  0.00 MPI_Finalize
```

```
0.00          4.98e5 (18.90%)          2.63e6
0.00          4.56e5 (91.60%)          4.98e5
0.00          100.00                   100.00
34.94         55.65 +/- 24.90          232.95
```

Augmented metrics from *automatic trace analysis* calculated using non-local event information show MPI waiting time is the complement of calculation

Late Sender time (for early receives)



6% of point-to-point communication time is waiting

Cube 3.3 Qt: epik_pflotran_12p8192_trace_PAT_RT_HWPC_0/trace+HWC.cube.gz

File Display Topology Help

Absolute | Absolute

Metric tree

```
⊟ □ 0.00 Time
   ⊟ 🟧 1.68e6 Execution
      ⊟ 🟨 1.63e4 MPI
         ⊟ □ 0.00 Synchronization
            ⊟ □ 3.59 Collective
               🟨 1119.69 Wait at Barrier
               □ 1.14 Barrier Completion
         ⊟ □ 0.00 Communication
            ⊟ 🟨 1.38e5 Point-to-point
               ⊟ 🟨 4477.19 Late Sender
                  ⊟ □ 0.00 Messages in Wrong Order
                     🟨 3559.79 From different sources
                     □ 0.00 From same source
               □ 0.00 Late Receiver
            ⊟ 🟨 2.90e4 Collective
               □ 0.00 Early Reduce
               □ 15.66 Early Scan
               🟨 3516.18 Late Broadcast
               🟨 4.69e5 Wait at N x N
               🟨 1.61e4 N x N Completion
         ⊟ 🟨 3.83e4 File I/O
            🟨 1.80e5 Collective
         🟨 8385.88 Init/Exit
      🟨 4.02e4 Overhead
   🟥 5.15e9 Visits
⊞ 🟥 8.19e4 Synchronizations
⊞ 🟥 1.94e9 Communications
```

0.00 8036.98 (0.31%) 2.63e6

Call tree | Flat view

```
⊟ □ 0.00
   ⊞ □ ...it
         MPI_Comm_group
      0.00 petscinitialize_
   ⊞ 🟨 672.91 init
   ⊟ □ 0.00 stepperrun
      ⊟ □ 0.00 stepperstepflowdt
         ⊞ 🟨 1996.90 discretizationlocaltolocal
         ⊞ 🟨 248.41 globalupdateauxvars
         ⊟ □ 0.00 snessolve_
            ⊟ □ 0.00 SNESSolve (0.00%)
               ⊞ □ 0.00 SNESSetUp
               ⊞ 🟨 2898.39 SNESSolve_LS
         ⊞ □ 0.00 vecnorm_
         ⊞ □ 0.00 richardsmaxchange
      ⊟ □ 0.00 steppersteptransportdt
         ⊞ 🟨 29.20 discretizationlocaltolocal
         ⊟ □ 0.00 snessolve_
            ⊟ □ 0.00 SNESSolve (0.00%)
               ⊞ □ 0.00 SNESSetUp
               ⊞ 🟨 822.07 SNESSolve_LS
         ⊞ □ 0.00 vecnorm_
         ⊞ □ 0.00 rtmaxchange
   ⊞ □ 0.00 simulationdestroy
   ⊞ 🟨 1369.10 petscfinalize_
   ⊞ □ 0.00 MPI_Finalize
```

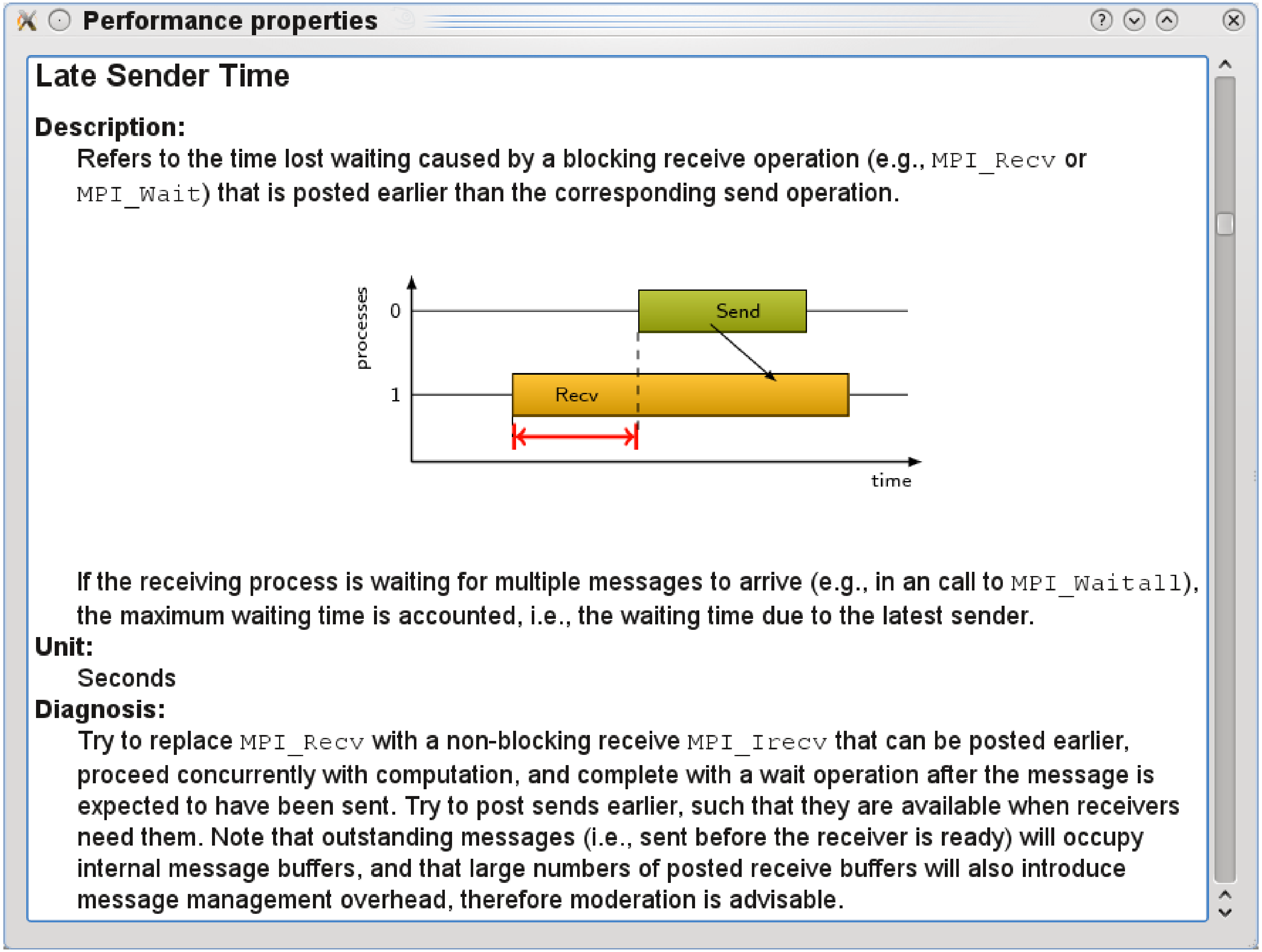
0.00 3720.46 (46.29%) 8036.98

System tree | Topology 0 | Topology 1

0.00 100.00 100.00

0.12 0.45 +/- 0.43 4.49

Scalasca 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:

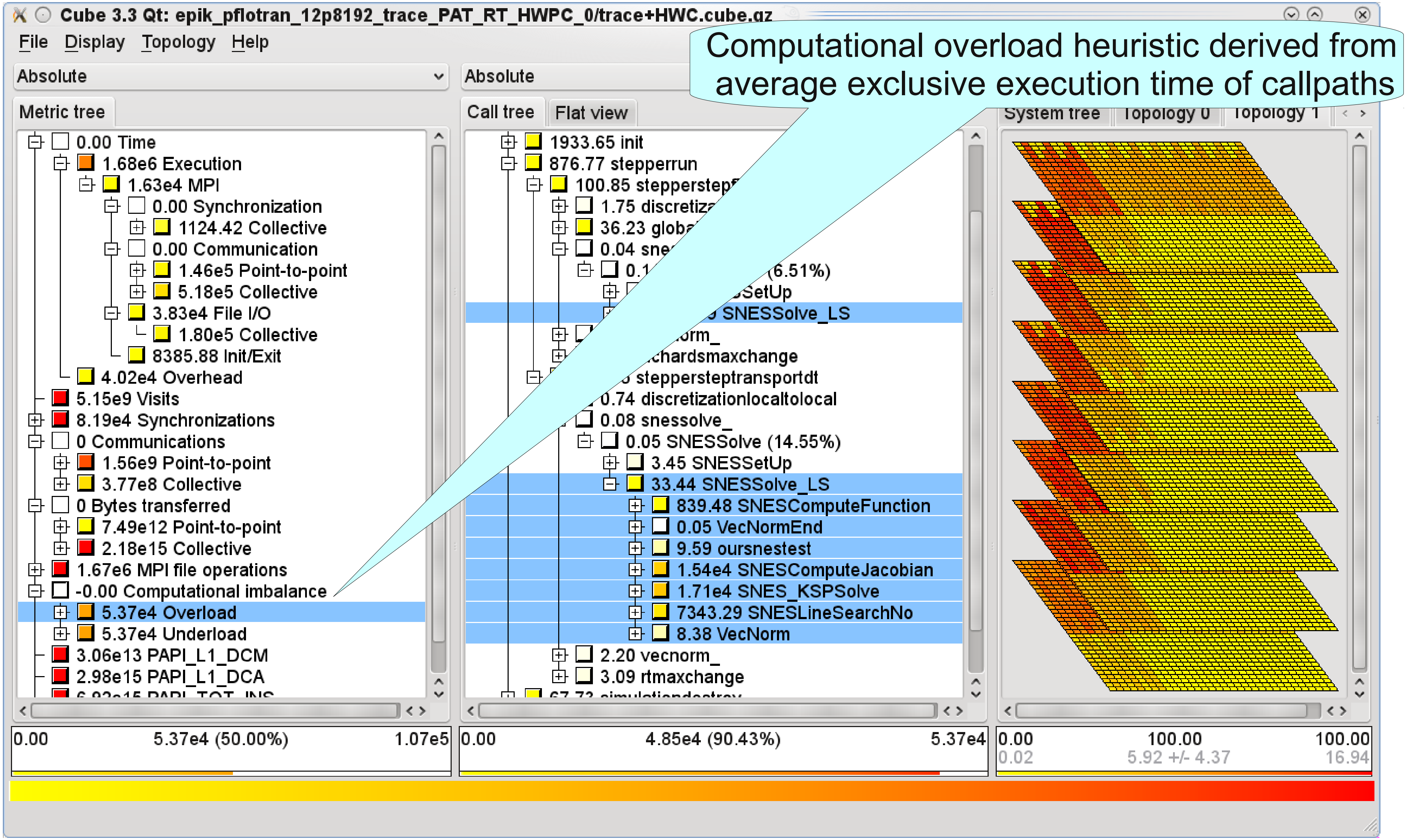
Seconds

Diagnosis:

Try to 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.

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

Computational imbalance: overload



Cube 3.3 Qt: epik_pflotran_12p8192_trace_PAT_RT_HWPC_0/trace+HWC.cube.gz

Computational overload heuristic derived from average exclusive execution time of callpaths

```
⊟ □ 0.00 Time
  ⊟ ■ 1.68e6 Execution
    ⊟ ■ 1.63e4 MPI
      ⊟ □ 0.00 Synchronization
        ⊞ ■ 1124.42 Collective
      ⊟ □ 0.00 Communication
        ⊞ ■ 1.46e5 Point-to-point
        ⊞ ■ 5.18e5 Collective
      ⊟ ■ 3.83e4 File I/O
        └ ■ 1.80e5 Collective
      ■ 8385.88 Init/Exit
    ■ 4.02e4 Overhead
■ 5.15e9 Visits
⊞ ■ 8.19e4 Synchronizations
⊟ □ 0 Communications
  ⊞ ■ 1.56e9 Point-to-point
  ⊞ ■ 3.77e8 Collective
⊟ □ 0 Bytes transferred
  ⊞ ■ 7.49e12 Point-to-point
  ⊞ ■ 2.18e15 Collective
⊞ ■ 1.67e6 MPI file operations
⊟ □ -0.00 Computational imbalance
  ⊞ ■ 5.37e4 Overload
  ⊞ ■ 5.37e4 Underload
■ 3.06e13 PAPI_L1_DCM
■ 2.98e15 PAPI_L1_DCA
■ 6.92e15 PAPI_TOT_INS
```

```
⊞ ■ 1933.65 init
⊟ ■ 876.77 stepperrun
  ⊟ ■ 100.85 stepperstep
    ⊞ □ 1.75 discretiza
    ⊞ ■ 36.23 globa
    ⊟ □ 0.04 sne
      ⊟ □ 0.1                (6.51%)
        ⊞ □                 SetUp
        ⊞ □                 SNESSolve_LS
    ⊞ □                 orm_
    ⊞ □                 chardsmaxchange
  ⊟ □                 stepperstepransportdt
    ⊞ □ 0.74 discretizationlocaltolocal
    ⊟ □ 0.08 snessolve_
      ⊟ □ 0.05 SNESSolve (14.55%)
        ⊞ □ 3.45 SNESSetUp
        ⊟ ■ 33.44 SNESSolve_LS
          ⊞ ■ 839.48 SNESComputeFunction
          ⊞ □ 0.05 VecNormEnd
          ⊞ □ 9.59 oursnestest
          ⊞ ■ 1.54e4 SNESComputeJacobian
          ⊞ ■ 1.71e4 SNES_KSPSolve
          ⊞ ■ 7343.29 SNESLineSearchNo
          ⊞ □ 8.38 VecNorm
    ⊞ □ 2.20 vecnorm_
    ⊞ □ 3.09 rtmaxchange
⊞ ■ 67.73 simulationdestroy
```

0.00	5.37e4 (50.00%)	1.07e5

0.00	4.85e4 (90.43%)	5.37e4

0.00	100.00	100.00
0.02	5.92 +/- 4.37	16.94

PFLOTRAN grid decomposition imbalance

850x1000x160 cells decomposed on 65536=64x64x16 process grid

- *x*-axis: 850/64=13 plus 18 extra cells
- *y*-axs: 1000/64=15 plus 40 extra cells
- *z*-axis: 160/16=10

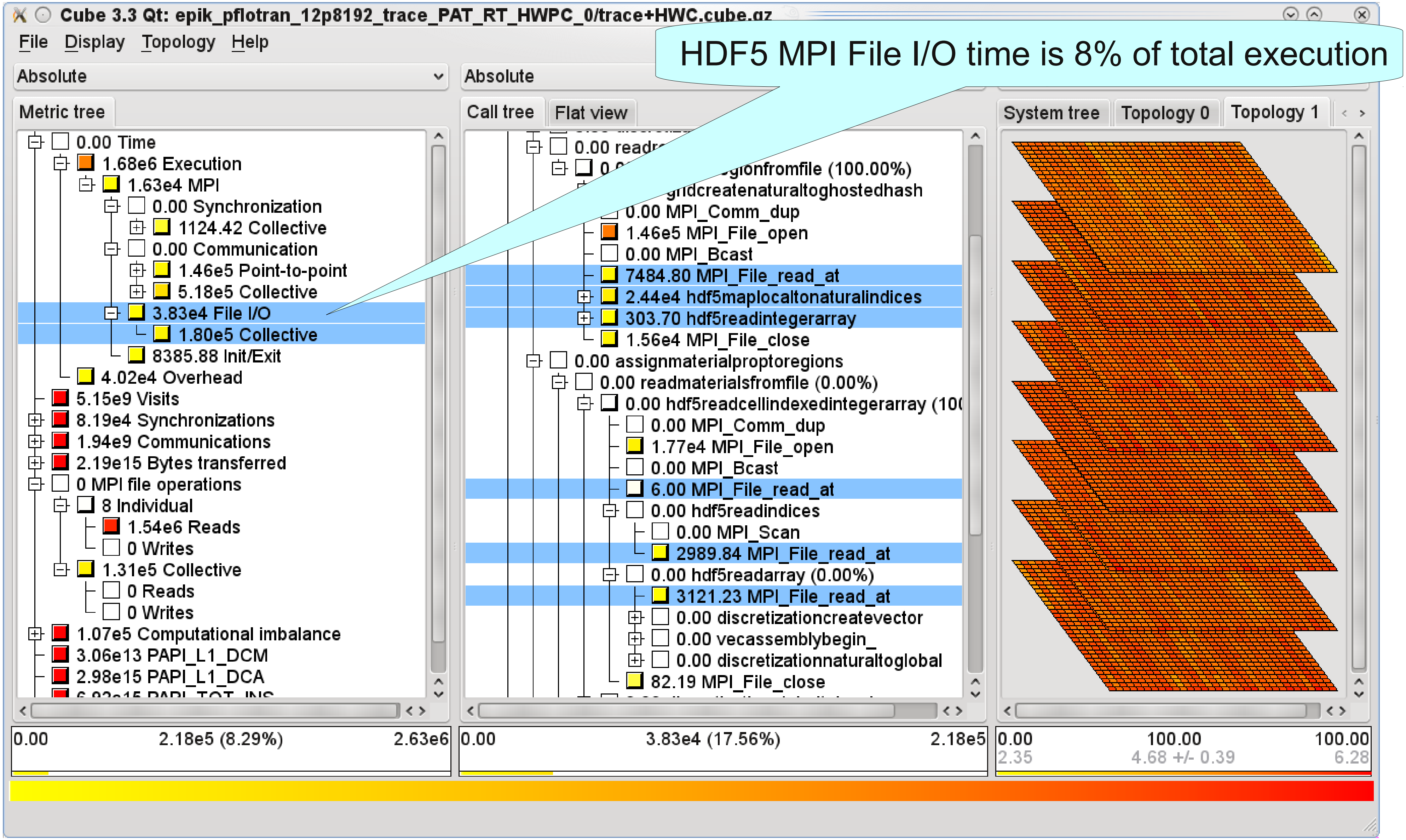
Perfect distribution would be 2075.2 cells per process

- but 20% of processes in each *z*-plane have 2240 cells
- 8% computation overload manifests as waiting times at the next communication/synchronization on the other processes

The problem-specific localized imbalance in the river channel is minor

- reversing the assignments in the *x*-dimension won't help much since some of the *z*-planes have no inactive cells

MPI File I/O time



Cube 3.3 Qt: epik_pflotran_12p8192_trace_PAT_RT_HWPC_0/trace+HWC.cube.gz

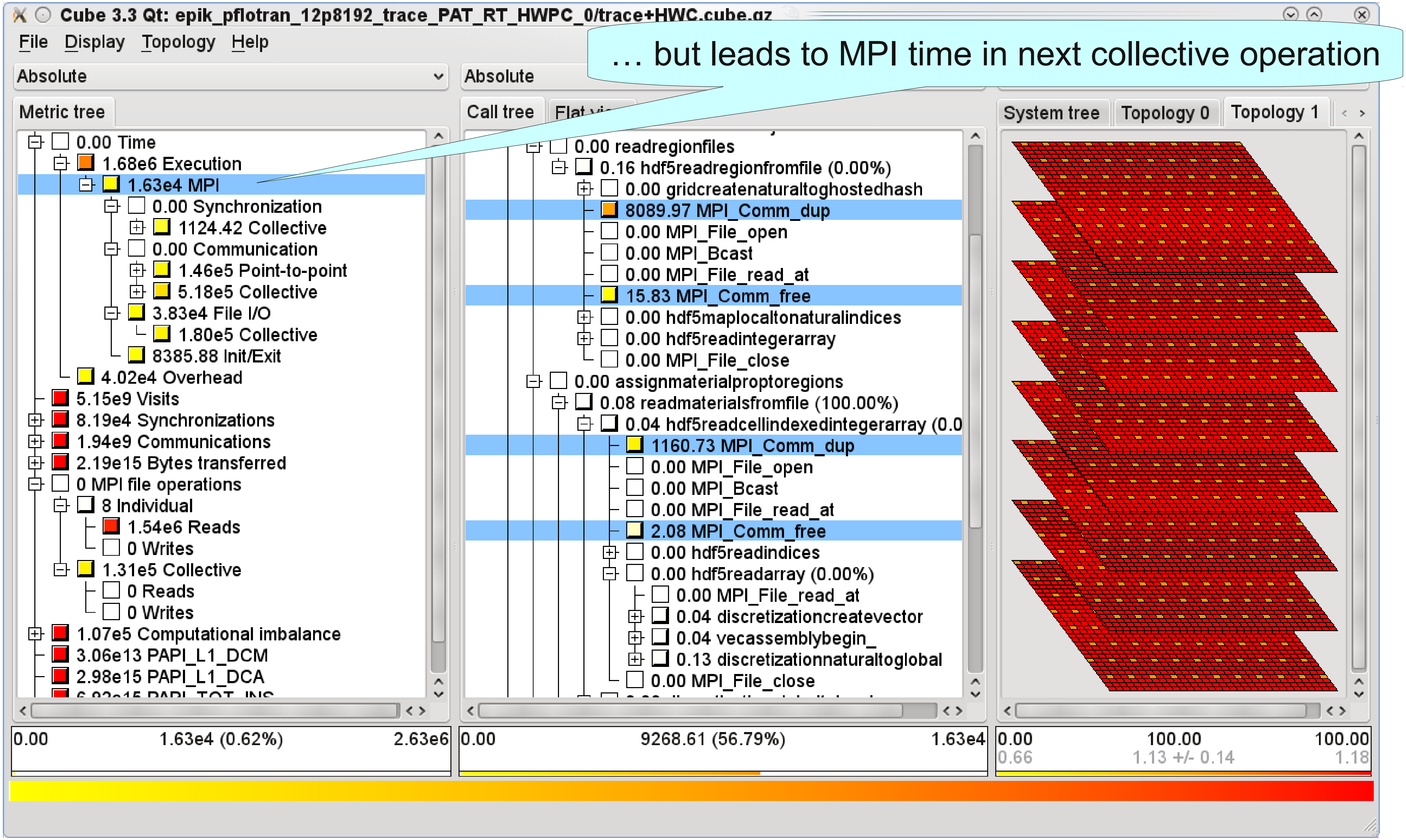
HDF5 MPI File I/O time is 8% of total execution

```
0.00 Time
  1.68e6 Execution
    1.63e4 MPI
      0.00 Synchronization
        1124.42 Collective
      0.00 Communication
        1.46e5 Point-to-point
        5.18e5 Collective
      3.83e4 File I/O
        1.80e5 Collective
      8385.88 Init/Exit
    4.02e4 Overhead
5.15e9 Visits
8.19e4 Synchronizations
1.94e9 Communications
2.19e15 Bytes transferred
0 MPI file operations
  8 Individual
    1.54e6 Reads
    0 Writes
  1.31e5 Collective
    0 Reads
    0 Writes
1.07e5 Computational imbalance
3.06e13 PAPI_L1_DCM
2.98e15 PAPI_L1_DCA
```

```
0.00 readr
  0.0          gionfromfile (100.00%)
               gridcreatenaturaltoghostedhash
    0.00 MPI_Comm_dup
    1.46e5 MPI_File_open
    0.00 MPI_Bcast
    7484.80 MPI_File_read_at
    2.44e4 hdf5maplocaltonaturalindices
    303.70 hdf5readintegerarray
    1.56e4 MPI_File_close
0.00 assignmaterialproptoregions
  0.00 readmaterialsfromfile (0.00%)
    0.00 hdf5readcellindexedintegerarray (10
      0.00 MPI_Comm_dup
      1.77e4 MPI_File_open
      0.00 MPI_Bcast
      6.00 MPI_File_read_at
      0.00 hdf5readindices
        0.00 MPI_Scan
        2989.84 MPI_File_read_at
      0.00 hdf5readarray (0.00%)
        3121.23 MPI_File_read_at
        0.00 discretizationcreatevector
        0.00 vecassemblybegin_
        0.00 discretizationnaturaltoglobal
      82.19 MPI_File_close
```

```
0.00        2.18e5 (8.29%)      2.63e6
0.00        3.83e4 (17.56%)     2.18e5
0.00        100.00              100.00
2.35        4.68 +/- 0.39       6.28
```

MPI Communicator duplication



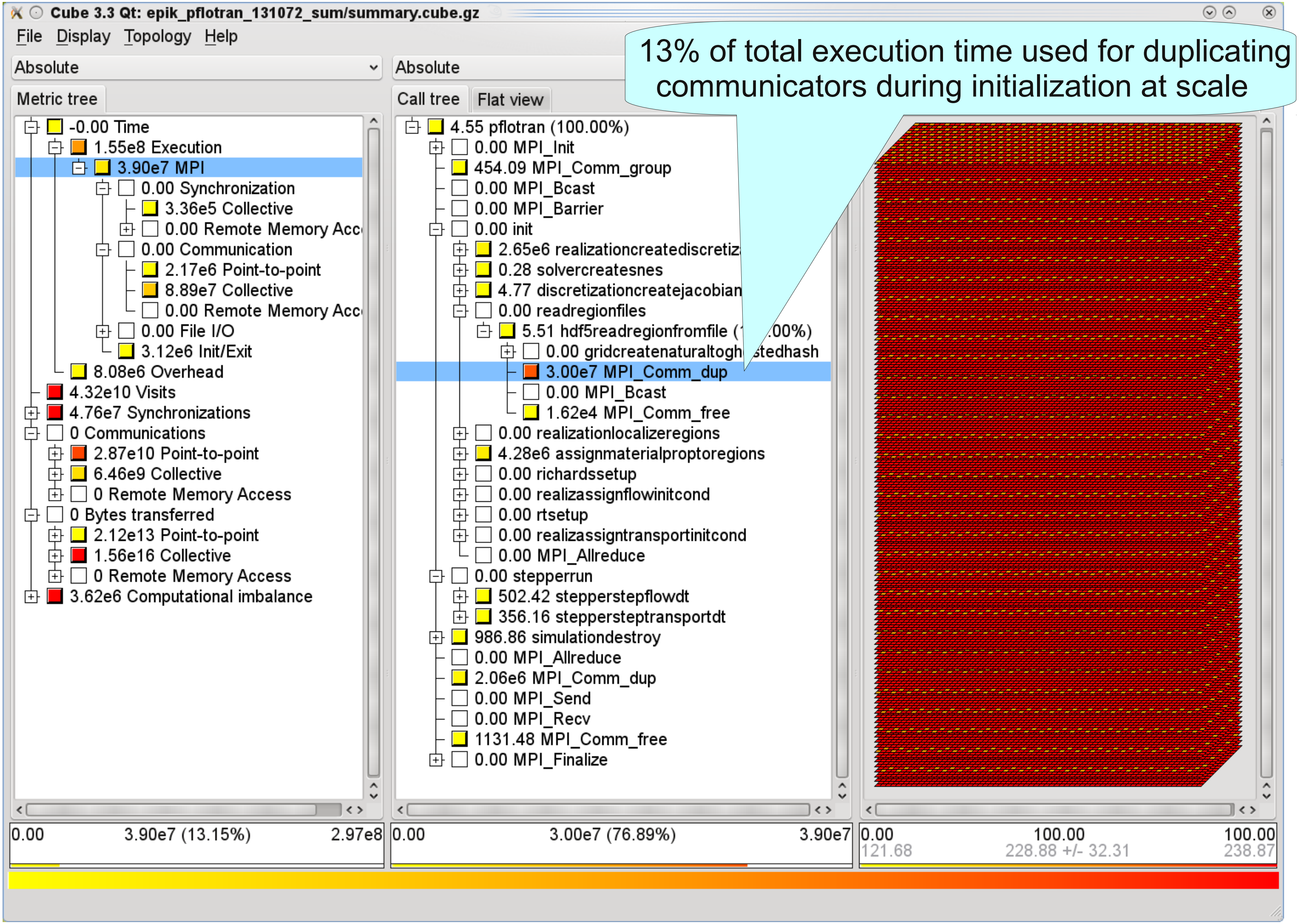
Cube 3.3 Qt: epik_pflotran_12p8192_trace_PAT_RT_HWPC_0/trace+HWC.cube.gz

… but leads to MPI time in next collective operation

```
0.00 Time
  1.68e6 Execution
    1.63e4 MPI
      0.00 Synchronization
        1124.42 Collective
      0.00 Communication
        1.46e5 Point-to-point
        5.18e5 Collective
      3.83e4 File I/O
        1.80e5 Collective
      8385.88 Init/Exit
  4.02e4 Overhead
5.15e9 Visits
8.19e4 Synchronizations
1.94e9 Communications
2.19e15 Bytes transferred
0 MPI file operations
  8 Individual
    1.54e6 Reads
    0 Writes
  1.31e5 Collective
    0 Reads
    0 Writes
1.07e5 Computational imbalance
3.06e13 PAPI_L1_DCM
2.98e15 PAPI_L1_DCA
6.93e15 PAPI_TOT_INS
```

```
0.00 readregionfiles
  0.16 hdf5readregionfromfile (0.00%)
    0.00 gridcreatenaturaltoghostedhash
    8089.97 MPI_Comm_dup
    0.00 MPI_File_open
    0.00 MPI_Bcast
    0.00 MPI_File_read_at
    15.83 MPI_Comm_free
    0.00 hdf5maplocaltonaturalindices
    0.00 hdf5readintegerarray
    0.00 MPI_File_close
0.00 assignmaterialproptoregions
  0.08 readmaterialsfromfile (100.00%)
    0.04 hdf5readcellindexedintegerarray (0.0
      1160.73 MPI_Comm_dup
      0.00 MPI_File_open
      0.00 MPI_Bcast
      0.00 MPI_File_read_at
      2.08 MPI_Comm_free
      0.00 hdf5readindices
      0.00 hdf5readarray (0.00%)
        0.00 MPI_File_read_at
        0.04 discretizationcreatevector
        0.04 vecassemblybegin_
        0.13 discretizationnaturaltoglobal
      0.00 MPI_File_close
```

```
0.00        1.63e4 (0.62%)        2.63e6
0.00        9268.61 (56.79%)      1.63e4
0.00        100.00                100.00
0.66        1.13 +/- 0.14         1.18
```



Cube 3.3 Qt: epik_pflotran_131072_sum/summary.cube.gz

```
-0.00 Time
  1.55e8 Execution
    3.90e7 MPI
      0.00 Synchronization
        3.36e5 Collective
        0.00 Remote Memory Acc
      0.00 Communication
        2.17e6 Point-to-point
        8.89e7 Collective
        0.00 Remote Memory Acc
      0.00 File I/O
      3.12e6 Init/Exit
    8.08e6 Overhead
  4.32e10 Visits
  4.76e7 Synchronizations
  0 Communications
    2.87e10 Point-to-point
    6.46e9 Collective
    0 Remote Memory Access
  0 Bytes transferred
    2.12e13 Point-to-point
    1.56e16 Collective
    0 Remote Memory Access
  3.62e6 Computational imbalance
```

```
4.55 pflotran (100.00%)
  0.00 MPI_Init
  454.09 MPI_Comm_group
  0.00 MPI_Bcast
  0.00 MPI_Barrier
  0.00 init
    2.65e6 realizationcreatediscretiz
    0.28 solvercreatesnes
    4.77 discretizationcreatejacobian
    0.00 readregionfiles
      5.51 hdf5readregionfromfile (1   .00%)
        0.00 gridcreatenaturaltogh   stedhash
        3.00e7 MPI_Comm_dup
        0.00 MPI_Bcast
        1.62e4 MPI_Comm_free
    0.00 realizationlocalizeregions
    4.28e6 assignmaterialproptoregions
    0.00 richardssetup
    0.00 realizassignflowinitcond
    0.00 rtsetup
    0.00 realizassigntransportinitcond
    0.00 MPI_Allreduce
  0.00 stepperrun
    502.42 stepperstepflowdt
    356.16 steppersteptransportdt
  986.86 simulationdestroy
  0.00 MPI_Allreduce
  2.06e6 MPI_Comm_dup
  0.00 MPI_Send
  0.00 MPI_Recv
  1131.48 MPI_Comm_free
  0.00 MPI_Finalize
```

```
0.00      3.90e7 (13.15%)      2.97e8
0.00      3.00e7 (76.89%)      3.90e7
0.00          100.00           100.00
121.68    228.88 +/- 32.31     238.87
```

13% of total execution time used for duplicating communicators during initialization at scale

Complementary analyses and visualizations

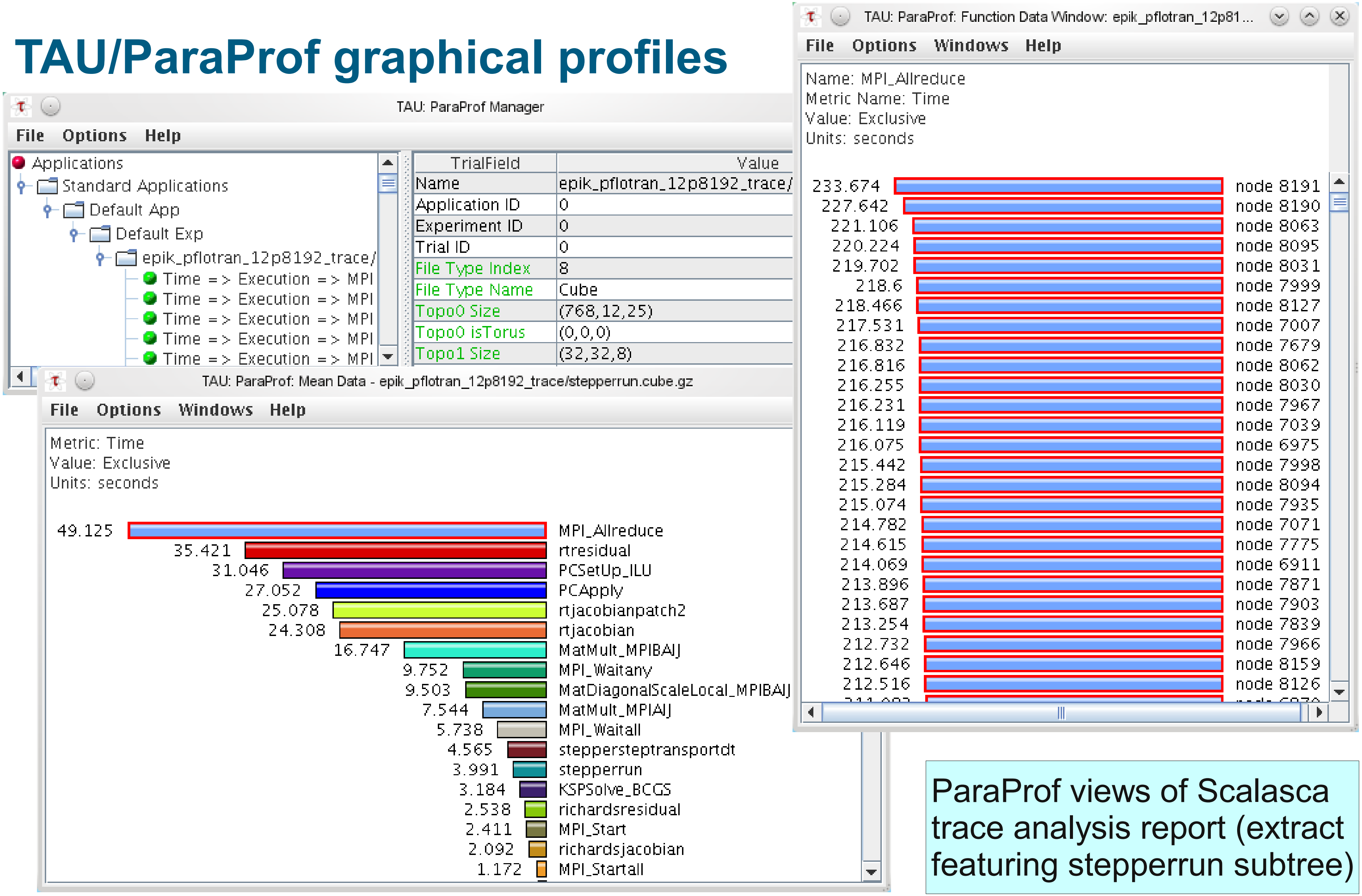
TAU/ParaProf can import Scalasca analysis reports

- part of portable open-source TAU performance system
- provides a callgraph display and various graphical profiles
- may need to extract part of report to fit available memory

Vampir 7 can visualize Scalasca distributed event traces

- part of commercially-licensed Vampir product suite
- provides interactive analysis & visualization of execution intervals
  - *powerful zoom and scroll to examine fine detail*
- avoids need for expensive trace merging and conversion
  - *required for visualization with Paraver & JumpShot*
  - *but often prohibitive for large traces*

TAU/ParaProf graphical profiles



- Applications
  - Standard Applications
    - Default App
      - Default Exp
        - epik_pflotran_12p8192_trace/
          - Time => Execution => MPI
          - Time => Execution => MPI
          - Time => Execution => MPI
          - Time => Execution => MPI
          - Time => Execution => MPI

TrialField	Value
Name	epik_pflotran_12p8192_trace/
Application ID	0
Experiment ID	0
Trial ID	0
File Type Index	8
File Type Name	Cube
Topo0 Size	(768,12,25)
Topo0 isTorus	(0,0,0)
Topo1 Size	(32,32,8)

TAU: ParaProf: Mean Data - epik_pflotran_12p8192_trace/stepperrun.cube.gz
File Options Windows Help

Metric: Time
Value: Exclusive
Units: seconds

49.125	MPI_Allreduce
35.421	rtresidual
31.046	PCSetUp_ILU
27.052	PCApply
25.078	rtjacobianpatch2
24.308	rtjacobian
16.747	MatMult_MPIBAIJ
9.752	MPI_Waitany
9.503	MatDiagonalScaleLocal_MPIBAIJ
7.544	MatMult_MPIAIJ
5.738	MPI_Waitall
4.565	steppersteptransportdt
3.991	stepperrun
3.184	KSPSolve_BCGS
2.538	richardsresidual
2.411	MPI_Start
2.092	richardsjacobian
1.172	MPI_Startall

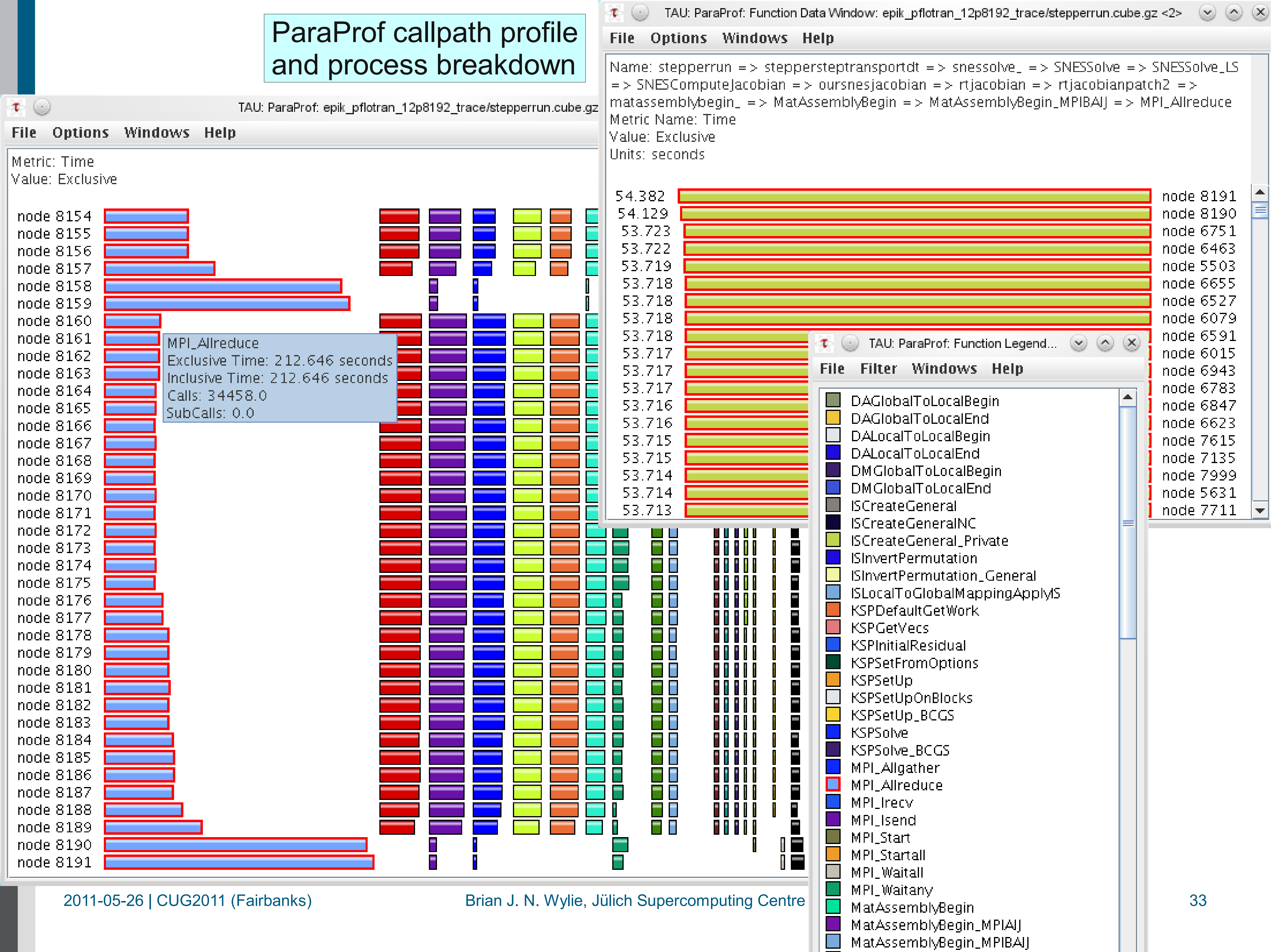
TAU: ParaProf: Function Data Window: epik_pflotran_12p81...
File Options Windows Help

Name: MPI_Allreduce
Metric Name: Time
Value: Exclusive
Units: seconds

233.674	node 8191
227.642	node 8190
221.106	node 8063
220.224	node 8095
219.702	node 8031
218.6	node 7999
218.466	node 8127
217.531	node 7007
216.832	node 7679
216.816	node 8062
216.255	node 8030
216.231	node 7967
216.119	node 7039
216.075	node 6975
215.442	node 7998
215.284	node 8094
215.074	node 7935
214.782	node 7071
214.615	node 7775
214.069	node 6911
213.896	node 7871
213.687	node 7903
213.254	node 7839
212.732	node 7966
212.646	node 8159
212.516	node 8126

ParaProf views of Scalasca trace analysis report (extract featuring stepperrun subtree)

ParaProf callpath profile and process breakdown


TAU: ParaProf: epik_pflotran_12p8192_trace/stepperrun.cube.gz
File Options Windows Help
TAU: ParaProf: Function Data Window: epik_pflotran_12p8192_trace/stepperrun.cube.gz <2>
File Options Windows Help
TAU: ParaProf: Function Legend...
File Filter Windows Help

Metric: Time
Value: Exclusive

```
node 8154
node 8155
node 8156
node 8157
node 8158
node 8159
node 8160
node 8161
node 8162
node 8163
node 8164
node 8165
node 8166
node 8167
node 8168
node 8169
node 8170
node 8171
node 8172
node 8173
node 8174
node 8175
node 8176
node 8177
node 8178
node 8179
node 8180
node 8181
node 8182
node 8183
node 8184
node 8185
node 8186
node 8187
node 8188
node 8189
node 8190
node 8191
```

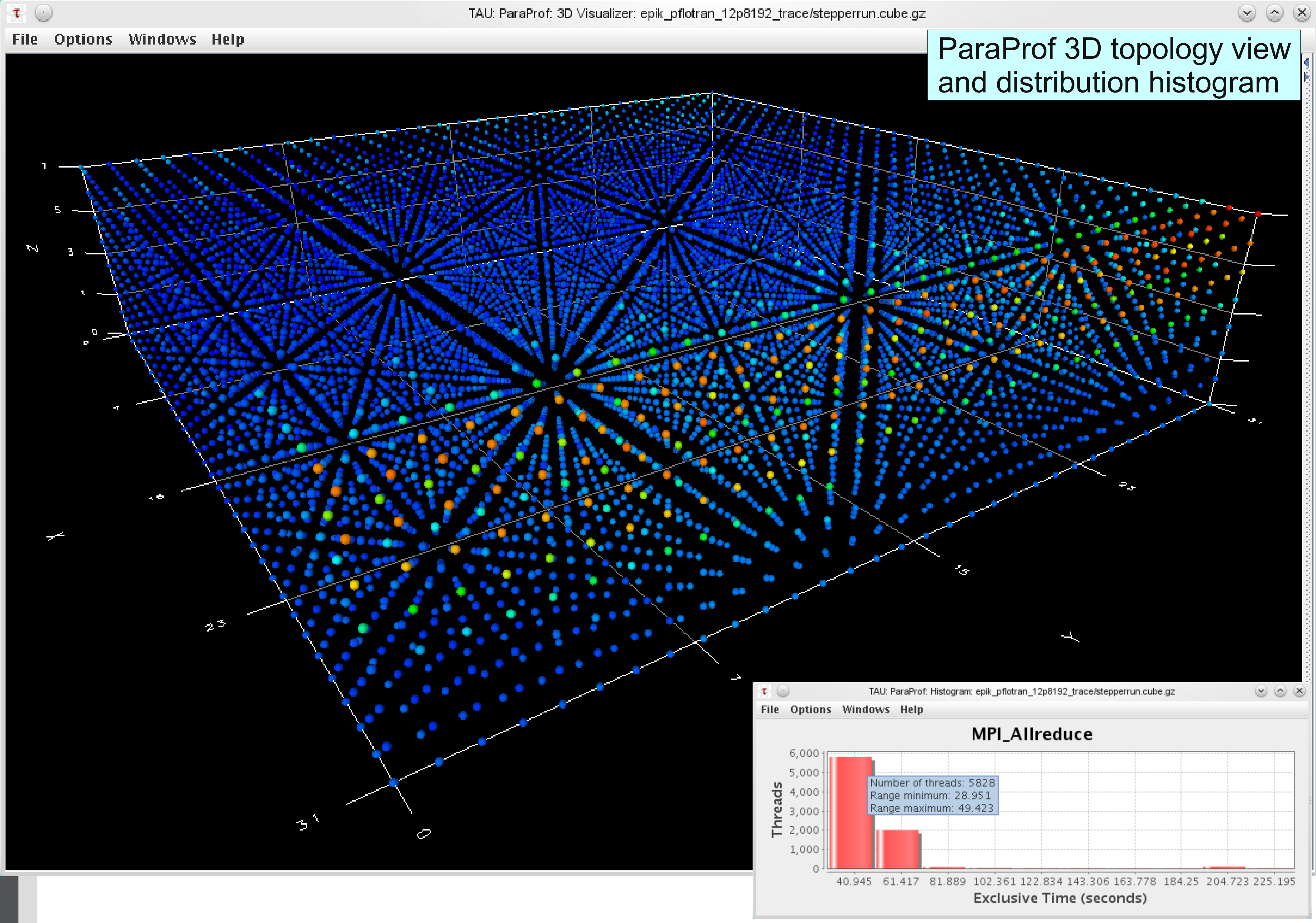
```
MPI_Allreduce
Exclusive Time: 212.646 seconds
Inclusive Time: 212.646 seconds
Calls: 34458.0
SubCalls: 0.0
```

Name: stepperrun => steppersteptransportdt => snessolve_ => SNESSolve => SNESSolve_LS => SNESComputeJacobian => oursnesjacobian => rtjacobian => rtjacobianpatch2 => matassemblybegin_ => MatAssemblyBegin => MatAssemblyBegin_MPIBAIJ => MPI_Allreduce
Metric Name: Time
Value: Exclusive
Units: seconds

```
54.382   node 8191
54.129   node 8190
53.723   node 6751
53.722   node 6463
53.719   node 5503
53.718   node 6655
53.718   node 6527
53.718   node 6079
53.718   node 6591
53.717   node 6015
53.717   node 6943
53.717   node 6783
53.716   node 6847
53.716   node 6623
53.715   node 7615
53.715   node 7135
53.714   node 7999
53.714   node 5631
53.713   node 7711
```

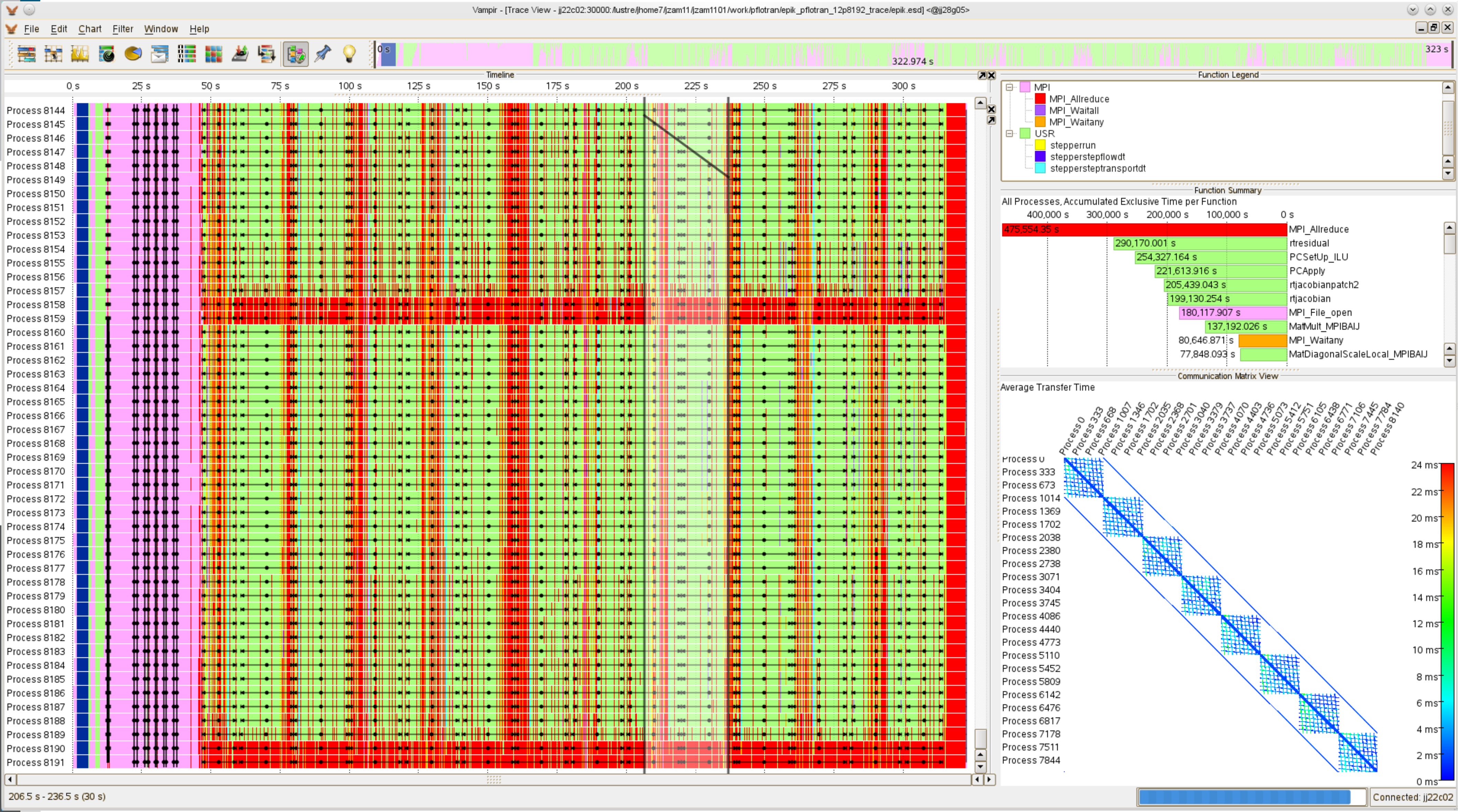
DAGlobalToLocalBegin
DAGlobalToLocalEnd
DALocalToLocalBegin
DALocalToLocalEnd
DMGlobalToLocalBegin
DMGlobalToLocalEnd
ISCreateGeneral
ISCreateGeneralNC
ISCreateGeneral_Private
ISInvertPermutation
ISInvertPermutation_General
ISLocalToGlobalMappingApplyIS
KSPDefaultGetWork
KSPGetVecs
KSPInitialResidual
KSPSetFromOptions
KSPSetUp
KSPSetUpOnBlocks
KSPSetUp_BCGS
KSPSolve
KSPSolve_BCGS
MPI_Allgather
MPI_Allreduce
MPI_Irecv
MPI_Isend
MPI_Start
MPI_Startall
MPI_Waitall
MPI_Waitany
MatAssemblyBegin
MatAssemblyBegin_MPIAIJ
MatAssemblyBegin_MPIBAIJ

ParaProf 3D topology view and distribution histogram

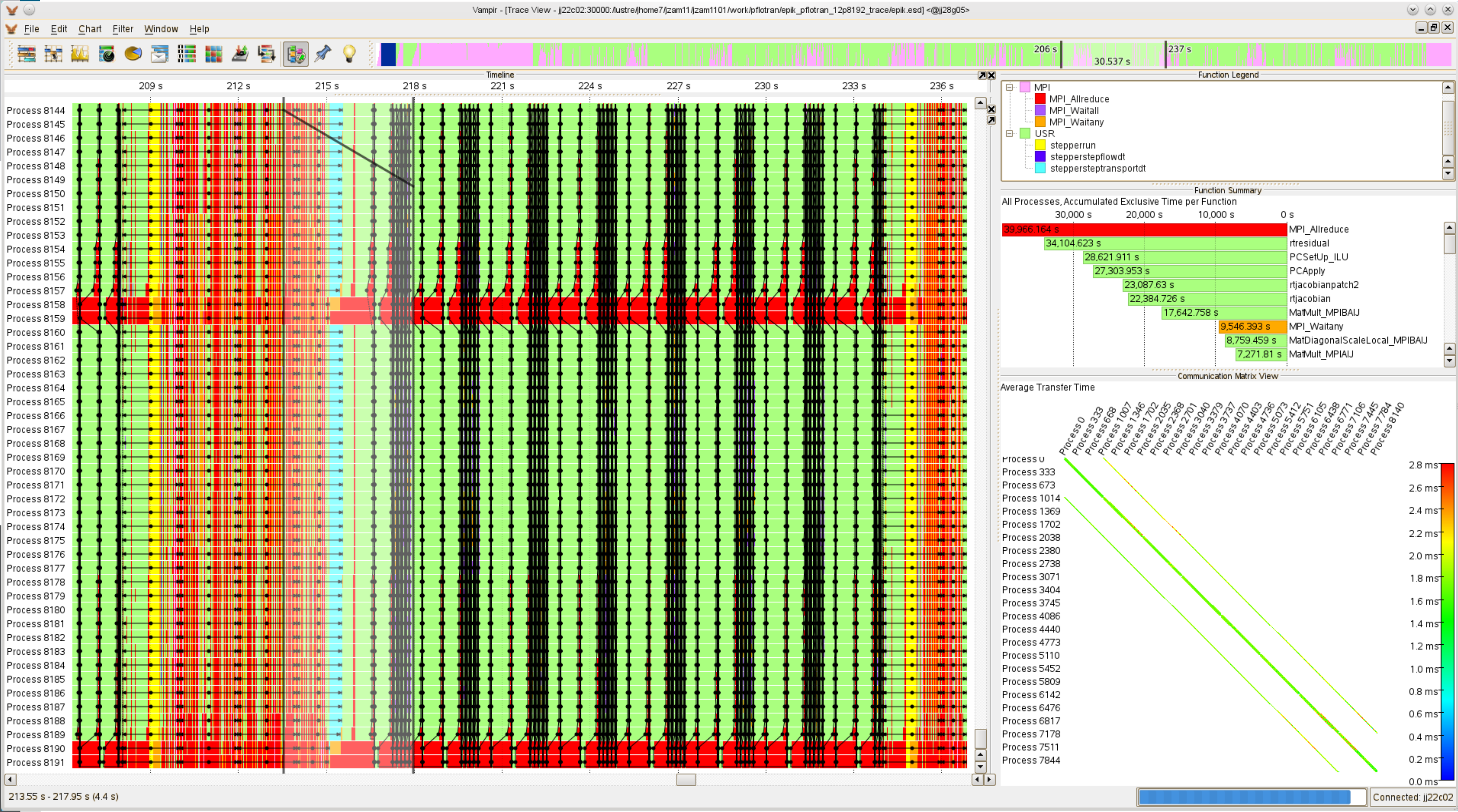


Number of threads: 5828
Range minimum: 28.951
Range maximum: 49.423

Vampir overview of *PFLOTRAN* execution (init + 10 steps)



PFLOTRAN execution timestep 7 (flow + transport)



Vampir - [Trace View - jj22c02:30000:/lustre/jhome7/jzam11/jzam1101/work/pflotran/epik_pflotran_12p8192_trace/epik.esd] <@jj28g05>

Function Legend

- MPI
  - 🟥 MPI_Allreduce
  - 🟪 MPI_Waitall
  - 🟧 MPI_Waitany
- 🟩 USR
  - 🟨 stepperrun
  - 🟦 stepperstepflowdt
  - 🟦 steppersteptransportdt

Function Summary

All Processes, Accumulated Exclusive Time per Function

Time	Function
39,966.164 s	MPI_Allreduce
34,104.623 s	rtresidual
28,621.911 s	PCSetUp_ILU
27,303.953 s	PCApply
23,087.63 s	rtjacobianpatch2
22,384.726 s	rtjacobian
17,642.758 s	MatMult_MPIBAIJ
9,546.393 s	MPI_Waitany
8,759.459 s	MatDiagonalScaleLocal_MPIBAIJ
7,271.81 s	MatMult_MPIAIJ

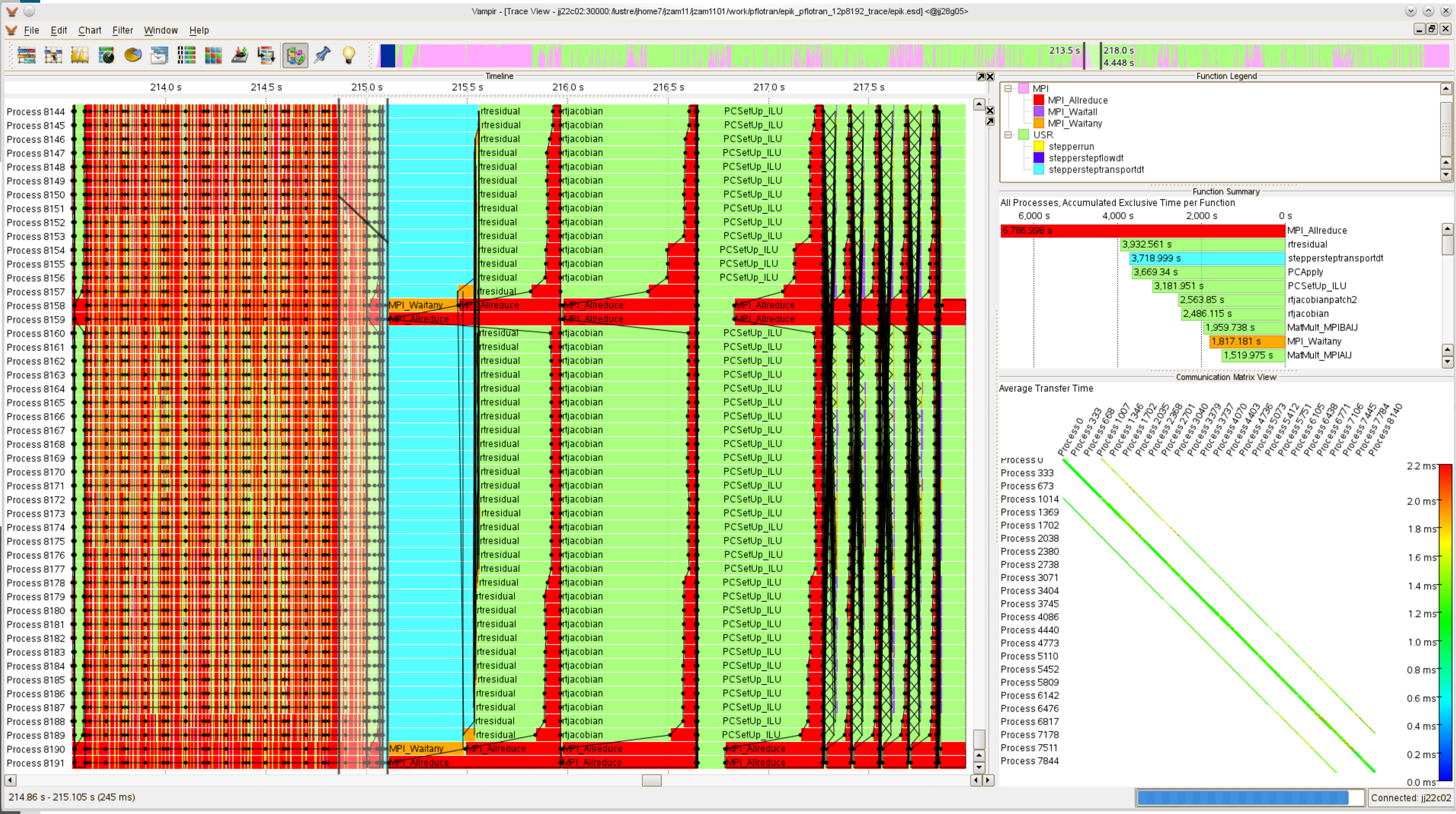
Communication Matrix View

Average Transfer Time

213.55 s - 217.95 s (4.4 s)

Connected: jj22c02

PFLOTRAN execution flow/transport transition



Vampir - [Trace View - jj22c02:30000:/lustre/jhome7/jzam11/jzam1101/work/pflotran/epik_pflotran_12p8192_trace/epik.esd] <@jj28g05>

Function Summary

All Processes, Accumulated Exclusive Time per Function

Time	Function
6,786.998 s	MPI_Allreduce
3,932.561 s	rtresidual
3,718.999 s	steppersteptransportdt
3,669.34 s	PCApply
3,181.951 s	PCSetUp_ILU
2,563.85 s	rtjacobianpatch2
2,486.115 s	rtjacobian
1,959.738 s	MatMult_MPIBAIJ
1,817.181 s	MPI_Waitany
1,519.975 s	MatMult_MPIAIJ

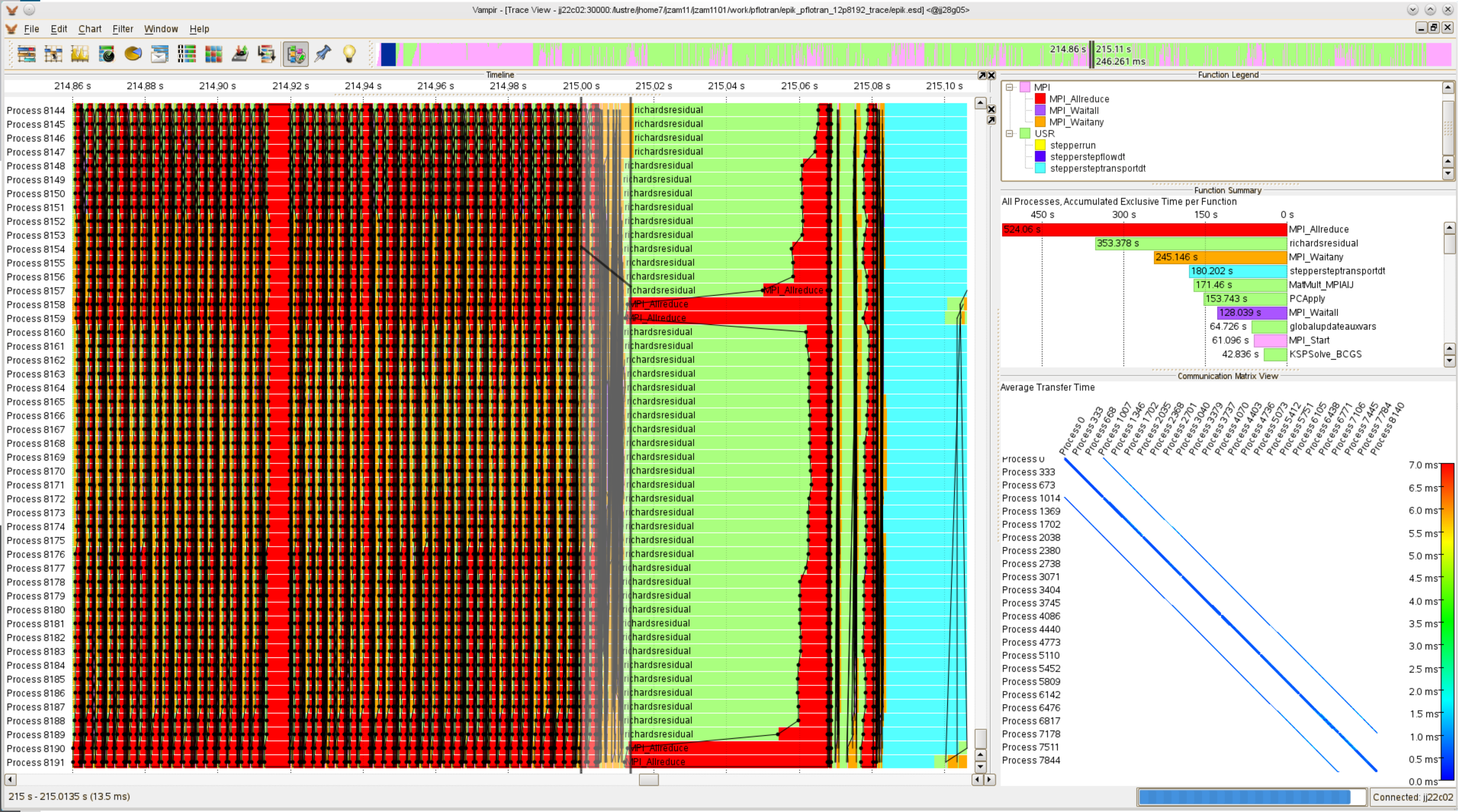
Function Legend

- 🟪 MPI
  - 🟥 MPI_Allreduce
  - 🟪 MPI_Waitall
  - 🟧 MPI_Waitany
- 🟩 USR
  - 🟨 stepperrun
  - 🟦 stepperstepflowdt
  - 🟦 steppersteptransportdt

214.86 s - 215.105 s (245 ms)

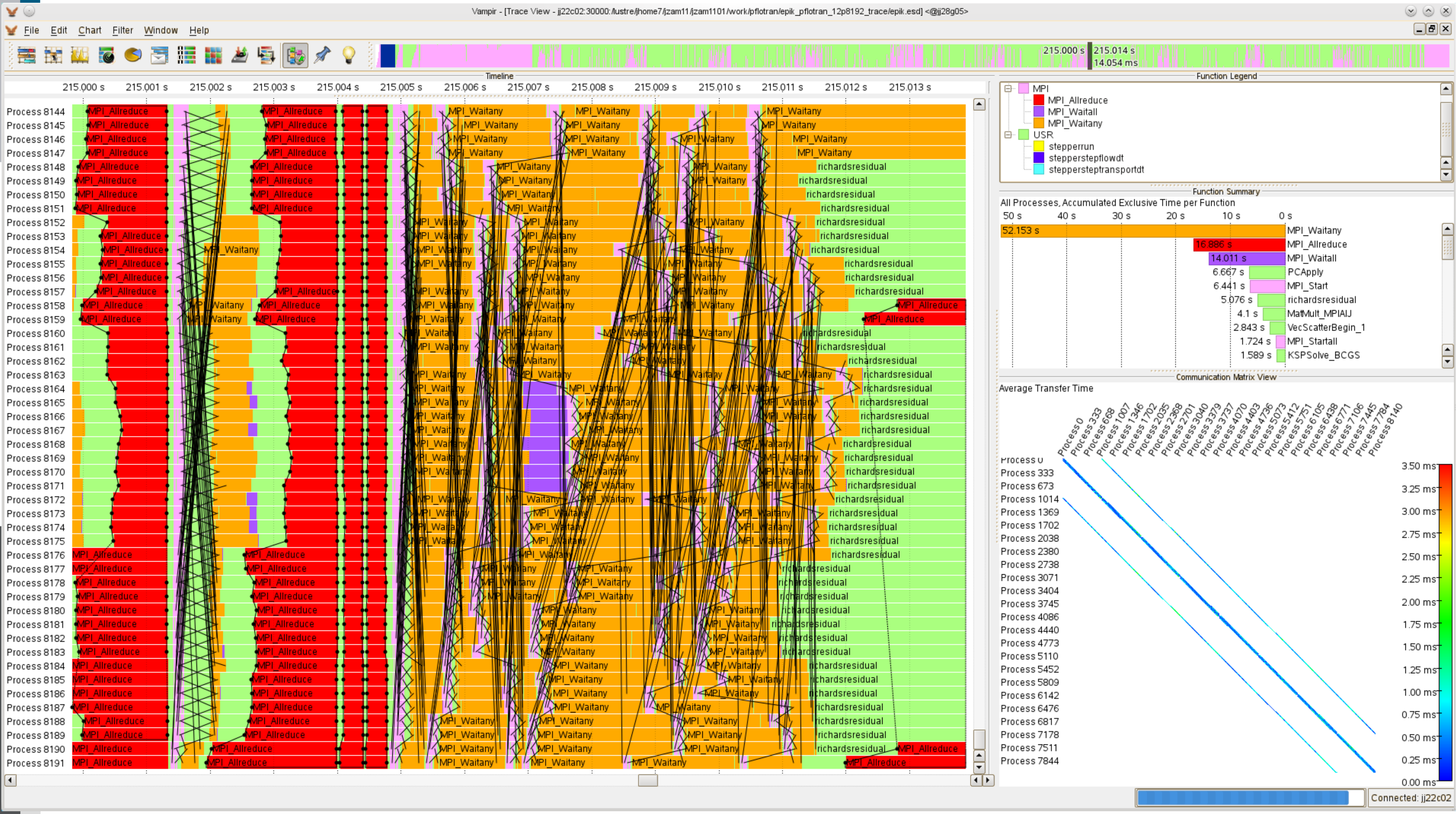
Connected: jj22c02

PFLOTRAN execution at end of flow phase



Vampir - [Trace View - jj22c02:30000:/lustre/jhome7/jzam11/jzam1101/work/pflotran/epik_pflotran_12p8192_trace/epik.esd] <@jj28g05>

PFLOTRAN execution at end of flow phase detail


Vampir - [Trace View - jj22c02:30000:/lustre/jhome7/jzam11/jzam1101/work/pflotran/epik_pflotran_12p8192_trace/epik.esd] <@jj28g05>

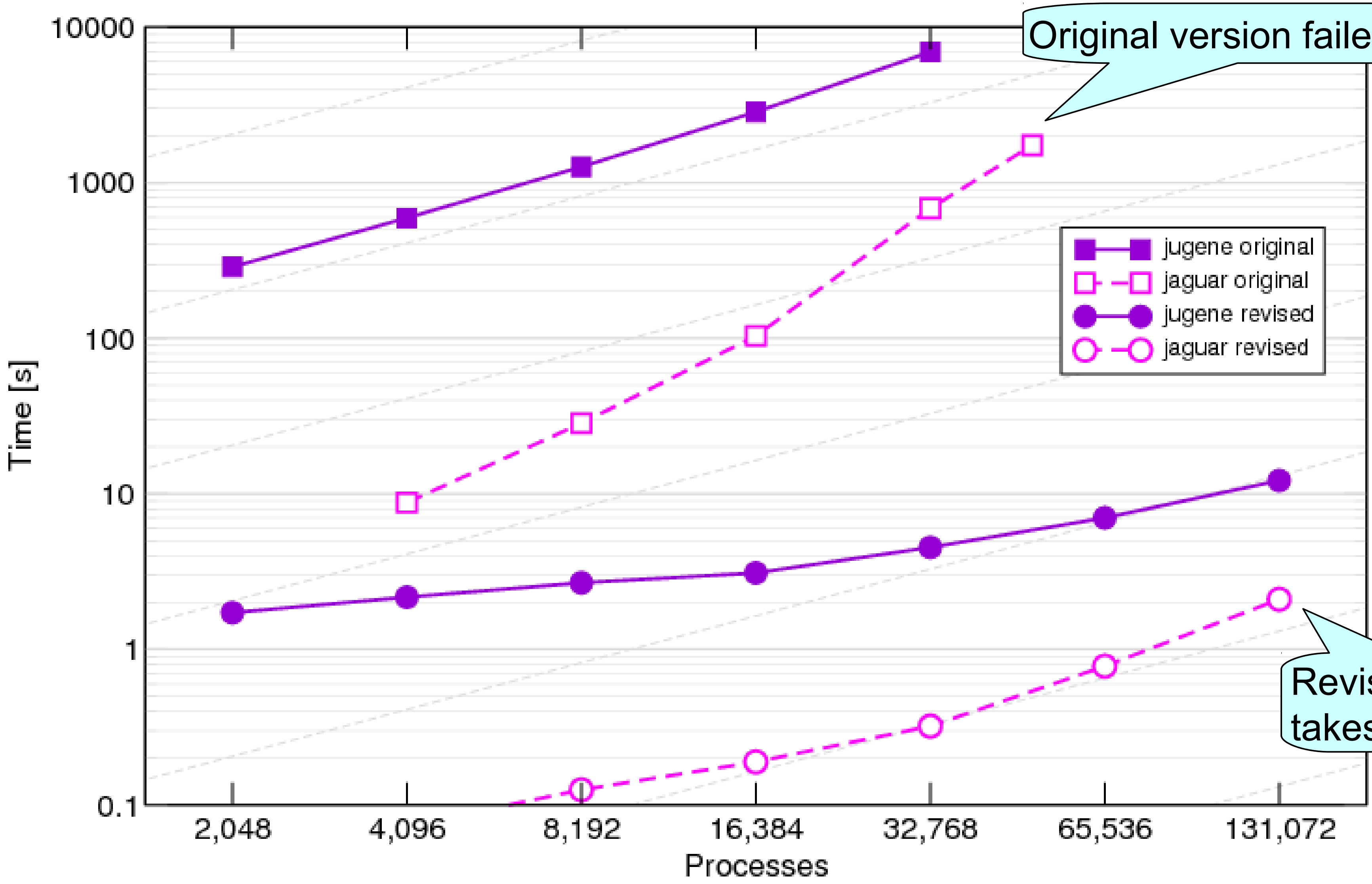
Scalasca/*PFLOTRAN* scalability issues

PFLOTRAN (via PETSc & HDF5) uses lots of MPI communicators

- 19 (MPI_COMM_WORLD) + 5xNPROCS (MPI_COMM_SELF)
  - *time shows up in collective MPI_Comm_dup*
- Not needed for summarization, but essential for trace replay
  - *definition storage & unification time grow accordingly*
- Original 1.3.1 version of Scalasca failed with 48k processes
  - *communicator definitions 1.42 GB, unification over 29 minutes*
- Revised 1.3.2 version resolved scalability bottleneck
  - *custom management of MPI_COMM_SELF communicators*
  - *hierarchical implementation of unification*

Scalasca trace collection and analysis costs increase with scale

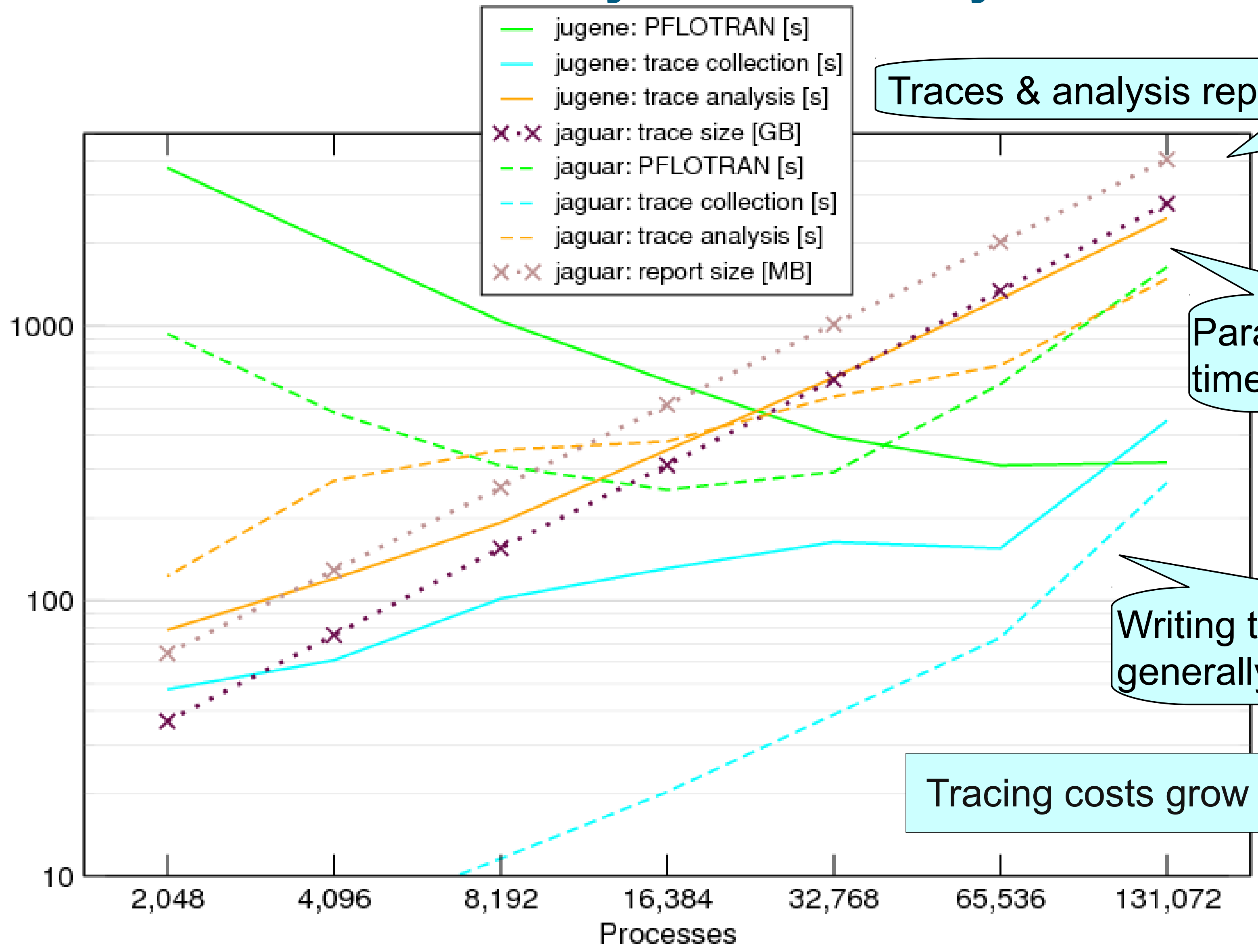
Improved unification of definition identifiers



Original version failed to scale

Revised version takes seconds

Trace collection & analysis scalability



Traces & analysis reports grow linearly

Parallel trace analysis time also mostly linear

Writing traces in parallel generally faster than linear

Tracing costs grow with scale

Conclusions

Very complex applications like *PFLOTRAN* provide significant challenges for performance analysis tools

Scalasca offers a range of instrumentation, measurement & analysis capabilities, with a simple GUI for interactive analysis report exploration

- works across Cray XT/XE and IBM BlueGene systems
- analysis reports and event traces can also be examined with complementary third-party tools such as TAU/ParaProf & Vampir
- convenient automatic instrumentation of applications and libraries must be moderated with selective measurement filtering
    - *analysis reports can still become awkwardly large*

Scalasca is continually improved in response to the evolving requirements of application developers and analysts

Acknowledgments

The application and benchmark developers who generously provided their codes and/or measurement archives

The facilities who made their HPC resources available and associated support staff who helped us use them effectively

- ALCF, BSC, **CSC**, **CSCS**, **EPCC**, JSC, HLRN, **HLRS**, ICL, LRZ, NCAR, **NCCS**, **NICS**, RWTH, RZG, SARA, TACC, ZIH
- Access & usage supported by European Union, German and other national funding organizations

Cray personnel for answers to our detailed questions

Scalasca users who have provided valuable feedback and suggestions for improvements

scalasca

Scalable performance **a**nalysis of **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*

Scalasca training

Tutorials

- full- or part-day hands-on training using POINT/VI-HPS Live DVD
- *HP-SEE/LinkSCEEM/PRACE*, Athens, Greece: 15 July 2011
- *EU/US Summer School*, South Lake Tahoe, USA: 9 Aug 2011
- *PRACE Summer School*, Espoo/Helsinki, Finland: 1 Sep 2011

VI-HPS Tuning Workshops

- 5-day hands-on training with participants' own application codes, in context of Virtual Institute – High Productivity Supercomputing
- *8th VI-HPS Tuning Workshop*, Aachen, Germany: 5-9 Sept 2011

Visit www.vi-hps.org/training for further details & announcements

Scalasca functionality in latest releases

Generic improvements

- Hybrid OpenMP/MPI measurement & analysis
- Selective routine instrumentation with *PDToolkit* instrumenter
- Runtime filtering of routines instrumented by PGI compilers
- Clock synchronization and event timestamp correction
- *SIONlib* mapping of task-local files into physical files
- Separate *CUBE3* GUI package for shared/distributed installations

Enhancements for Cray XT

- Support for multiple programming environments (PrgEnvs)
  - *PGI, GNU, CCE, Intel, Pathscale*
- Capture of mappings of processes to processors in machine

POINT/VI-HPS Linux Live-DVD/ISO

Bootable Linux installation on DVD or USB memory stick

Includes everything needed to try out parallel tools on an x86-architecture notebook computer

- VI-HPS tools: KCachegrind, Marmot, PAPI, Periscope, Scalasca, TAU, VT/Vampir*
- Also: Eclipse/PTP, TotalView*, etc.
  - ** time/capability-limited evaluation licenses provided*
- GCC (w/OpenMP), OpenMPI
- User Guides, exercises & examples

http://www.vi-hps.org/training/material/

Parallel Productivity Tools
Live DVD
Also includes: TotalView, DyninstAPI, PDT, Eclipse PTP, Berkeley UPC, ptoolsrte, Chapel, and much more...

Partners:
ParaTools, Inc.
University of Florida
University of Oregon
Totalview Technologies
RWTH Aachen University
HLRS / University of Stuttgart
Jülich Supercomputing Centre
Technische Universität Dresden
Technische Universität München
University of Wisconsin at Madison
Pittsburgh Supercomputing Center
University of Tennessee at Knoxville
National Center for Supercomputing Applications

September 2010

POINT VI-HPS

http://nic.uoregon.edu/point http://www.vi-hps.org