JÜLICH
FORSCHUNGSZENTRUM

Scalasca: A Scalable Portable Integrated Performance Measurement and Analysis Toolset

CHANGES 2012 | Bernd Mohr

Scalasca Overview

JÜLICH FORSCHUNGSZENTRUM

- **Goal**
  - Measure and analyze performance of parallel programs
- **Context**
  - “Real-world” HPC applications
  - Batch processing, space sharing
  - Large distributed memory machines
- **Our focus**
  - Extreme scalability
  - Portability
  - Integration

Increasing Importance of Scaling

- Number of Cores share for TOP 500 June 2012

NCore	Count	Share	∑Rmax	Share	∑NCore
1025-2048	1	0.2%	122 TF	0.1%	1,280
2049-4096	9	1.8%	623 TF	0.5%	30,520
4097-8192	85	17.0%	7,500 TF	6.1%	581,728
8193-16384	268	53.6%	24,852 TF	20.1%	3,319,798
> 16384	137	27.4%	90,376 TF	73.2%	9,519,131
Total	500	100%	123,473 TF	100%	13,452,457

- **Average** system size: **26,904 cores**
- **Median** system size: **13,104 cores**

Portability: Supported Platforms

JÜLICH FORSCHUNGSZENTRUM

- **Instrumentation and measurement only (visual analysis on front-end or workstation)**
  - Cray XT3/XT4/XT5, XE6, XK6
  - IBM BlueGene/L, BlueGene/P, BlueGene/Q
  - NEC SX8 and SX9
  - K Machine?

- **Full support (instrumentation, measurement, and automatic analysis)**
  - Linux IA32, IA64, x86_64, and PPC based clusters
  - IBM AIX Power3/4/5/6/7 based clusters
  - SGI Linux IA64 and x86_64 based clusters
  - SUN/Oracle Solaris Sparc and x86/x86_64 based clusters

Known Installations of Scalasca

JÜLICH FORSCHUNGSZENTRUM

Companies

- Bull (France)
- Dassault Aviation (France)
- EDF (France)
- GNS (Germany)
- MAGMA (Germany)
- RECOM (Germany)
- Shell (Netherlands)
- Sun Microsystems (USA)
- Qontix (UK)

Research / HPC Centres

- ANL (USA)
- BSC (Spain)
- CEA (France)
- CERFACS (France)
- CINECA (Italy)
- CSC (Finland)
- CSCS (Switzerland)

Research / HPC Centres (cont.)

- DLR (Germany)
- DKRZ (Germany)
- EPCC (UK)
- HLRN (Germany)
- HLRS (Germany)
- ICHEC (Ireland)
- IDRIS (France)
- JSCC (Russia)
- LLNL (USA)
- LRZ (Germany)
- MSU (Russia)
- NCAR (USA)
- NCSA (USA)
- NSCC (China)
- ORNL (USA)
- PSC (USA)
- RZG (Germany)

Research / HPC Centres (cont.)

- SARA (Netherlands)
- SAITC (Bulgaria)
- TACC (USA)

Universities

- RPI (USA)
- RWTH (Germany)
- TUD (Germany)
- UOregon (USA)
- UTK (USA)

DoD Computing Centers (USA)

- AFRL DSRC
- ARL DSRC
- ARSC DSRC
- ERDC DSRC
- Navy DSRC
- MHPCC DSRC
- SSC-Pacific

Integration

- Need integrated tool (environment)
  for all levels of parallelization
  - Inter-node (MPI)
  - Intra-node (OpenMP, task-based programming)
  - Accelerators (CUDA, OpenCL)

- Integration with **performance modeling and prediction**

- No tool fits all requirements
  - **Interoperability of tools**
  - Integration via open interfaces

The Scalasca Project

JÜLICH FORSCHUNGSZENTRUM

- Scalable Analysis of Large Scale Applications

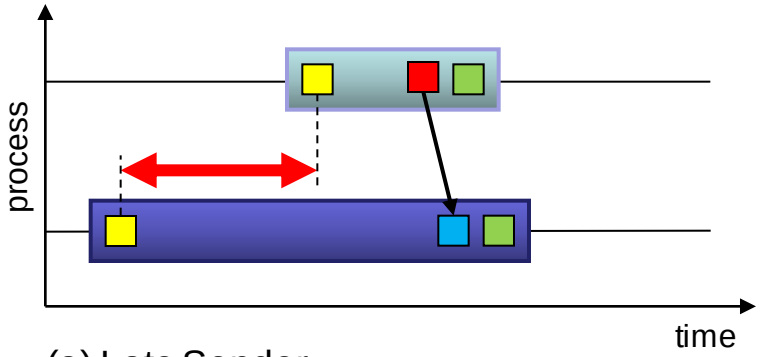
http://www.scalasca.org/

- Approach
  - **Instrument** C, C++, and Fortran parallel applications
    - Based on MPI, OpenMP, SHMEM, or hybrid
  - **Option 1: scalable call-path profiling**
  - **Option 2: scalable event trace analysis**
    - **Collect** event traces
    - **Search** trace for event patterns representing inefficiencies
    - **Categorize and rank** inefficiencies found
- Supports MPI 2.2 (P2P, collectives, RMA, IO) and OpenMP 3.0 (exception: nesting)

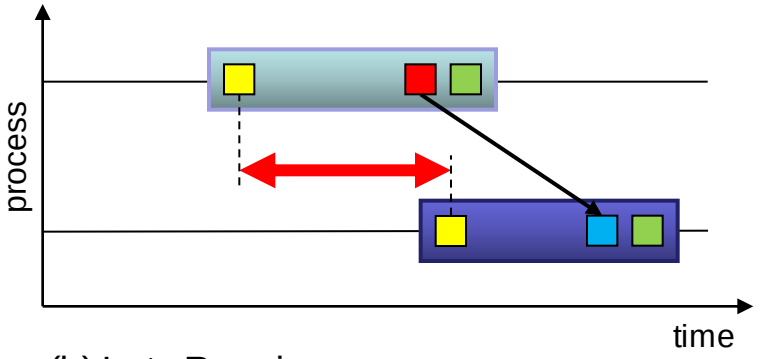
JÜLICH FORSCHUNGSZENTRUM

German Research School for Simulation Sciences

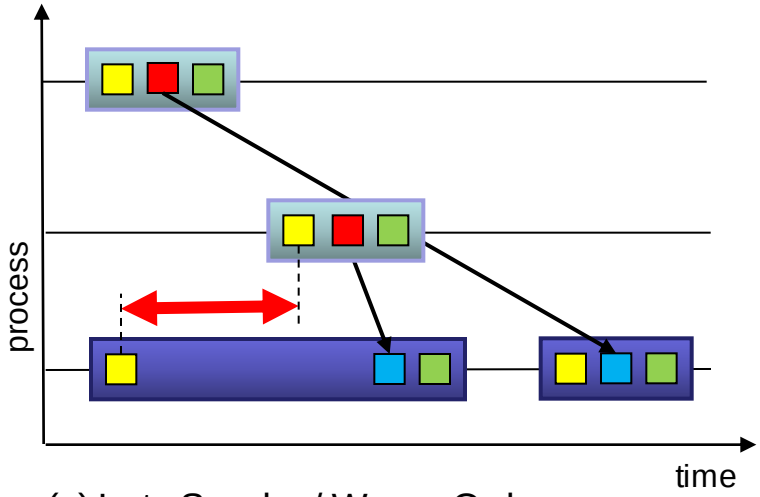
Example Patterns



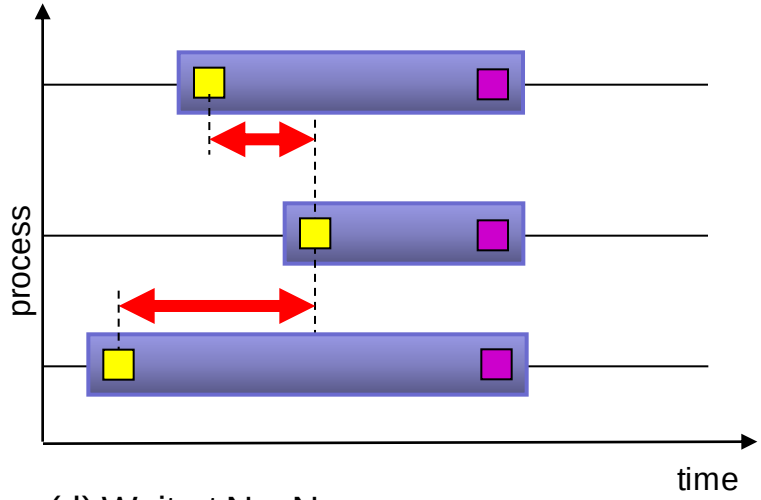
(a) Late Sender



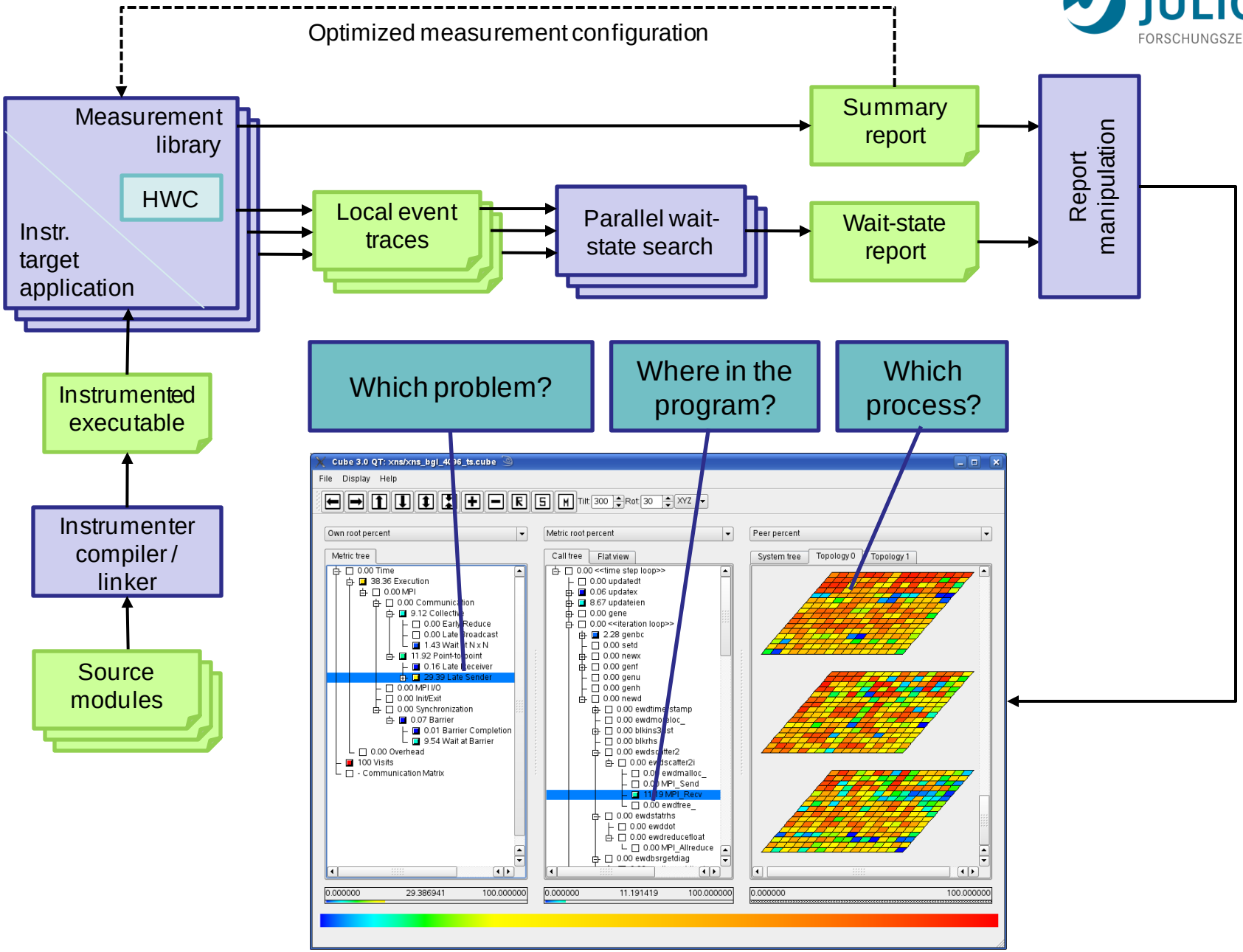
(b) Late Receiver



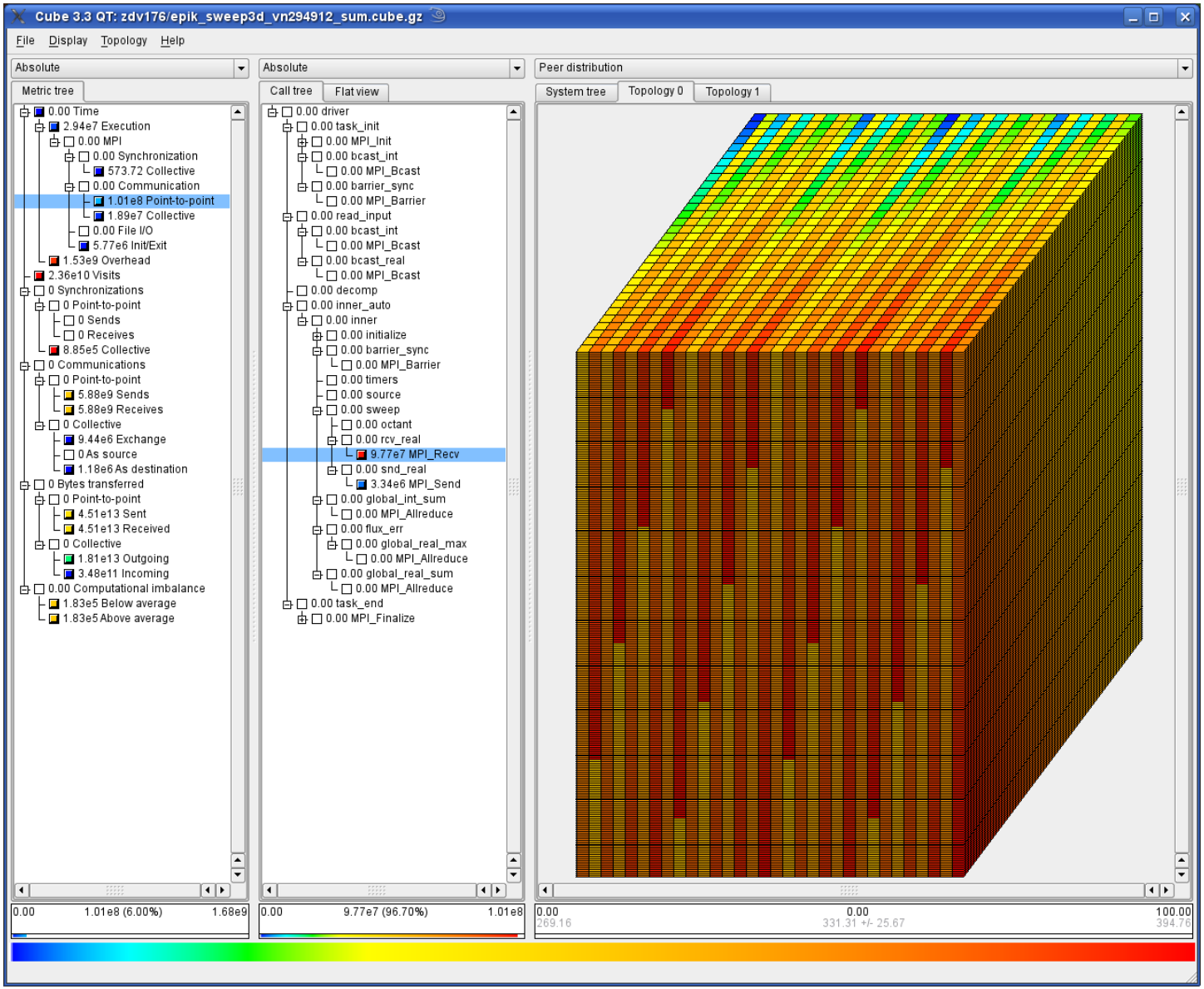
(c) Late Sender / Wrong Order



(d) Wait at N x N

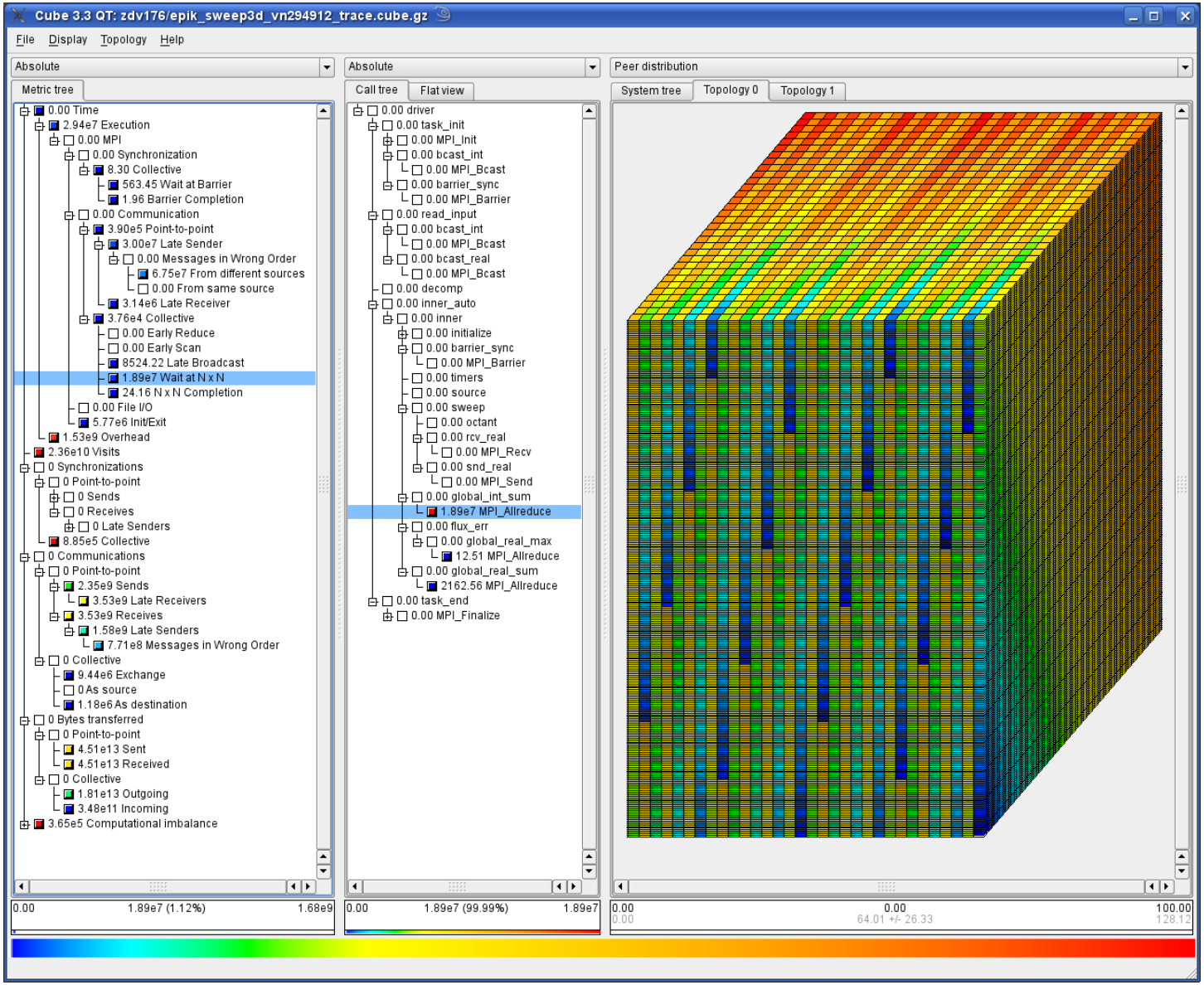

Optimized measurement configuration
Measurement library
HWC
Instr. target application
Local event traces
Parallel wait-state search
Summary report
Wait-state report
Report manipulation
Instrumented executable
Instrumenter compiler / linker
Source modules
Which problem?
Where in the program?
Which process?
Cube 3.0 QT: xns/xns_bgl_4096_ts.cube
File Display Help
Own root percent
Metric root percent
Peer percent
Metric tree
Call tree
Flat view
System tree
Topology 0
Topology 1
0.00 Time
38.36 Execution
0.00 MPI
0.00 Communication
9.12 Collective
0.00 Early Reduce
0.00 Late Broadcast
1.43 Wait at N x N
11.92 Point-to-point
0.16 Late Receiver
29.39 Late Sender
0.00 MPI I/O
0.00 Init/Exit
0.00 Synchronization
0.07 Barrier
0.01 Barrier Completion
9.54 Wait at Barrier
0.00 Overhead
100 Visits
- Communication Matrix
0.00 <<time step loop>>
0.00 updatedt
0.06 updatex
8.67 updateien
0.00 gene
0.00 <<iteration loop>>
2.28 genbc
0.00 setd
0.00 newx
0.00 genf
0.00 genu
0.00 genh
0.00 newd
0.00 ewdtimestamp
0.00 ewdmoreloc_
0.00 blkins3dst
0.00 blkrhs
0.00 ewdscatter2
0.00 ewdscatter2i
0.00 ewdmalloc_
0.00 MPI_Send
11.19 MPI_Recv
0.00 ewdfree_
0.00 ewdstatrhs
0.00 ewddot
0.00 ewdreducefloat
0.00 MPI_Allreduce
0.00 ewdbsrgetdiag
0.000000 29.386941 100.000000
0.000000 11.191419 100.000000
0.000000 100.000000

Summary analysis sweep3D@294,912 BGP



Trace analysis sweep3D@294,912 BGP

- 10 min sweep3D runtime
- 11 sec replay
- 4 min trace data write/read (576 files)
- 7.6 TB buffered trace data
- 510 billion events

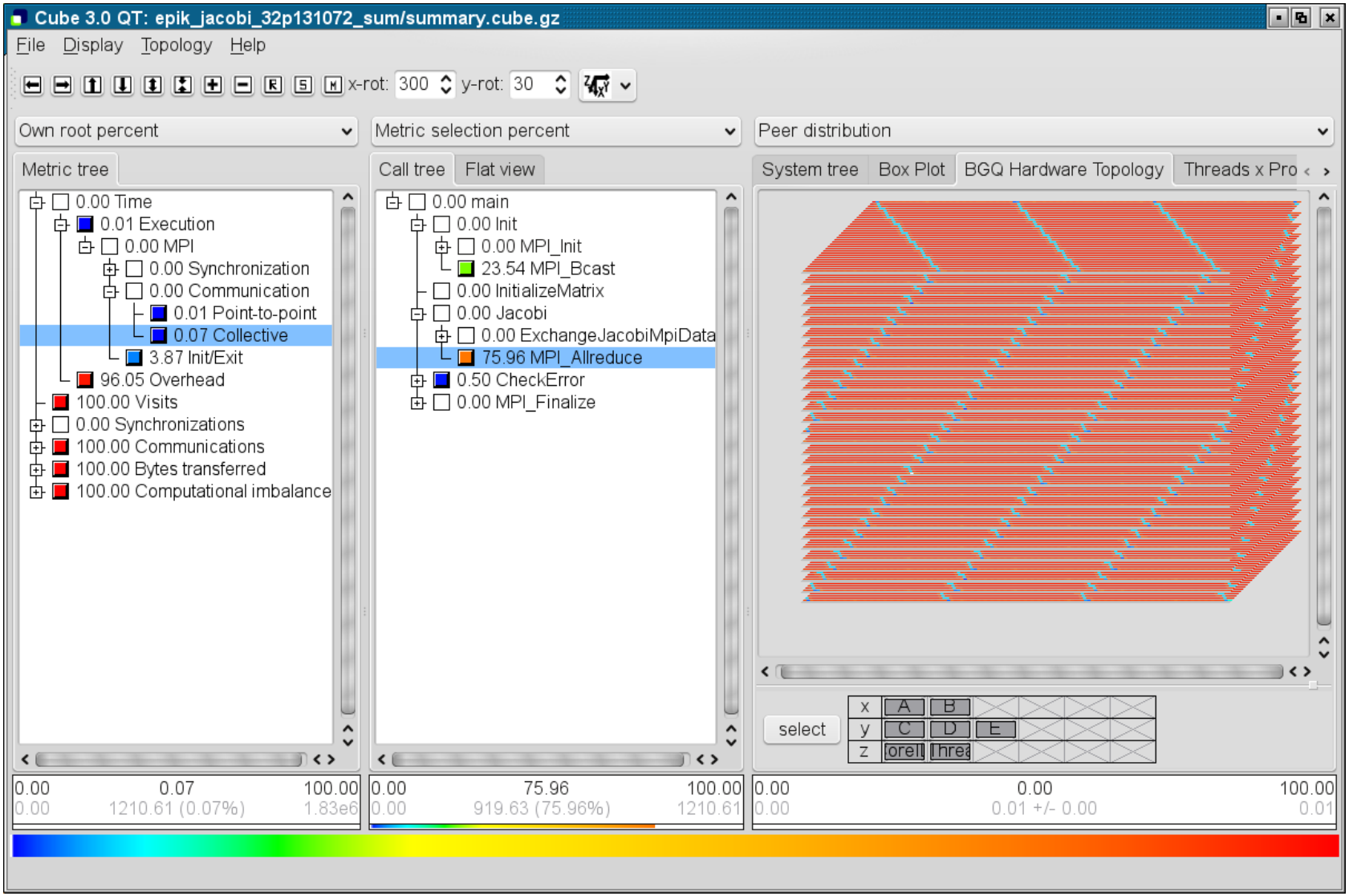


B. J. N. Wylie, M. Geimer, B. Mohr, D. Böhme, Z.Szebenyi, F. Wolf: Large-scale performance analysis of Sweep3D with the Scalasca toolset. Parallel Processing Letters, 20(4):397-414, 2010.

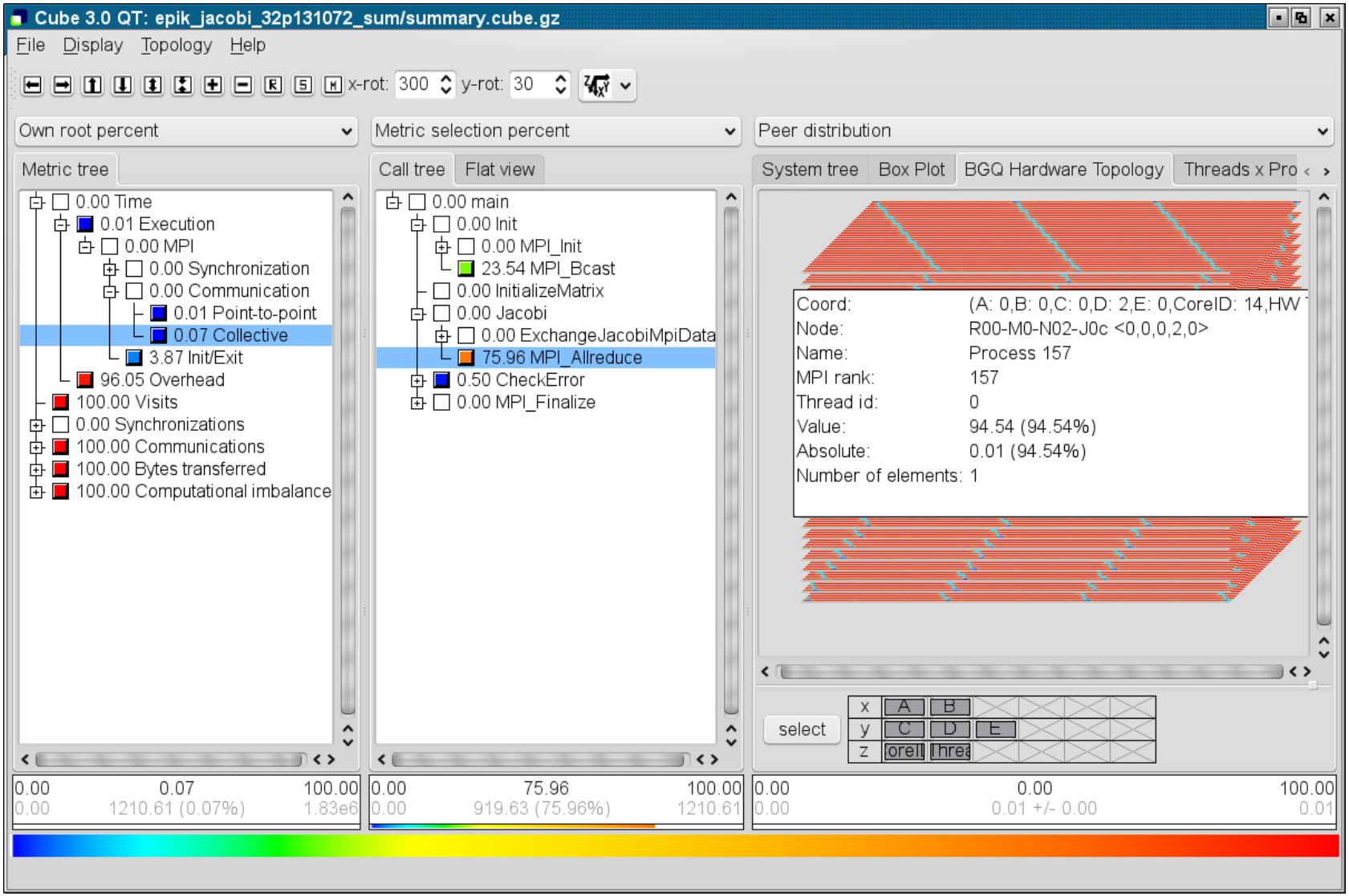
Current research

- Work in progress:
  - Further **scalability improvements**
  - Integration with trace visualizers (Paraver, Vampir) and Rogue Wave TreadSpotter
  - Integration with MAQAO binary instrumenter
- Medium term:
  - Further **scalability improvements**
  - **Phase analysis**
  - **Root cause analysis**
  - Support for **emerging programming paradigms**
    - **Asynchronous tasks (CUDA, OpenCL, HMPP, OMPSs…)**
    - **One-sided communication (PGAS, armci, shmem…)**

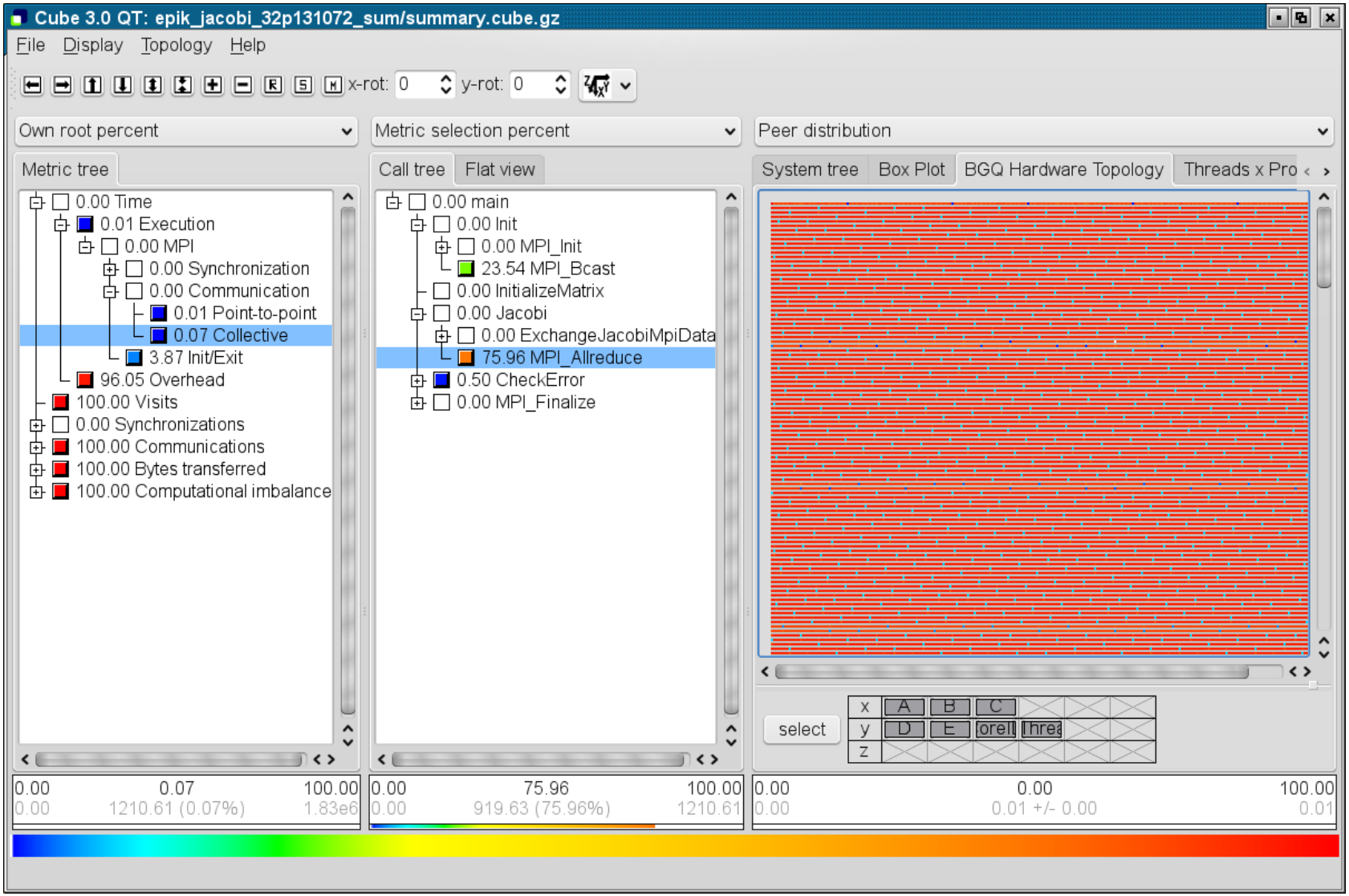
CUBE: > 3D ⇨ Folding (e.g. AB-CDE-CT)



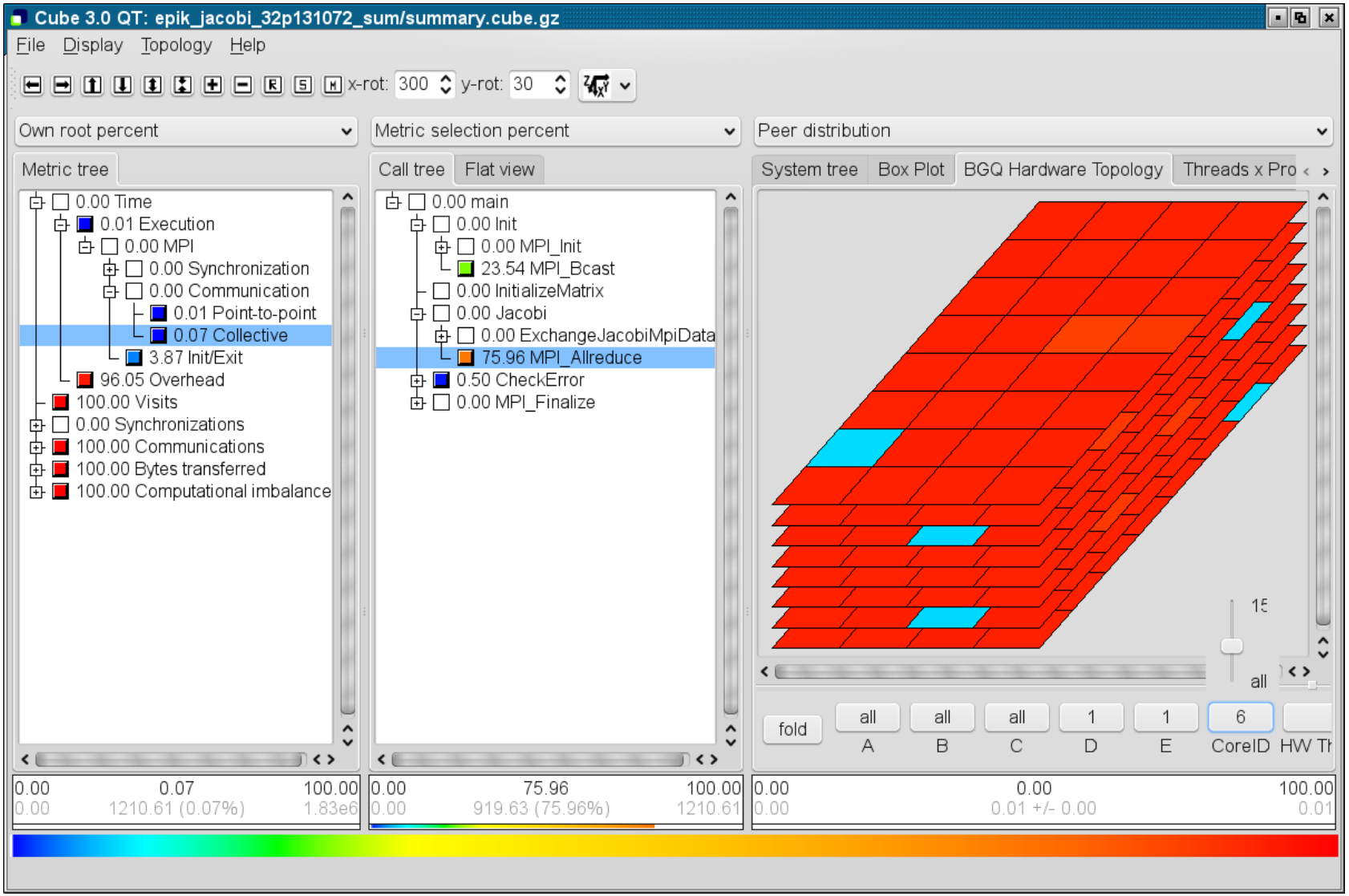
CUBE: > 3D ⇨ Folding (e.g. AB-CDE-CT)



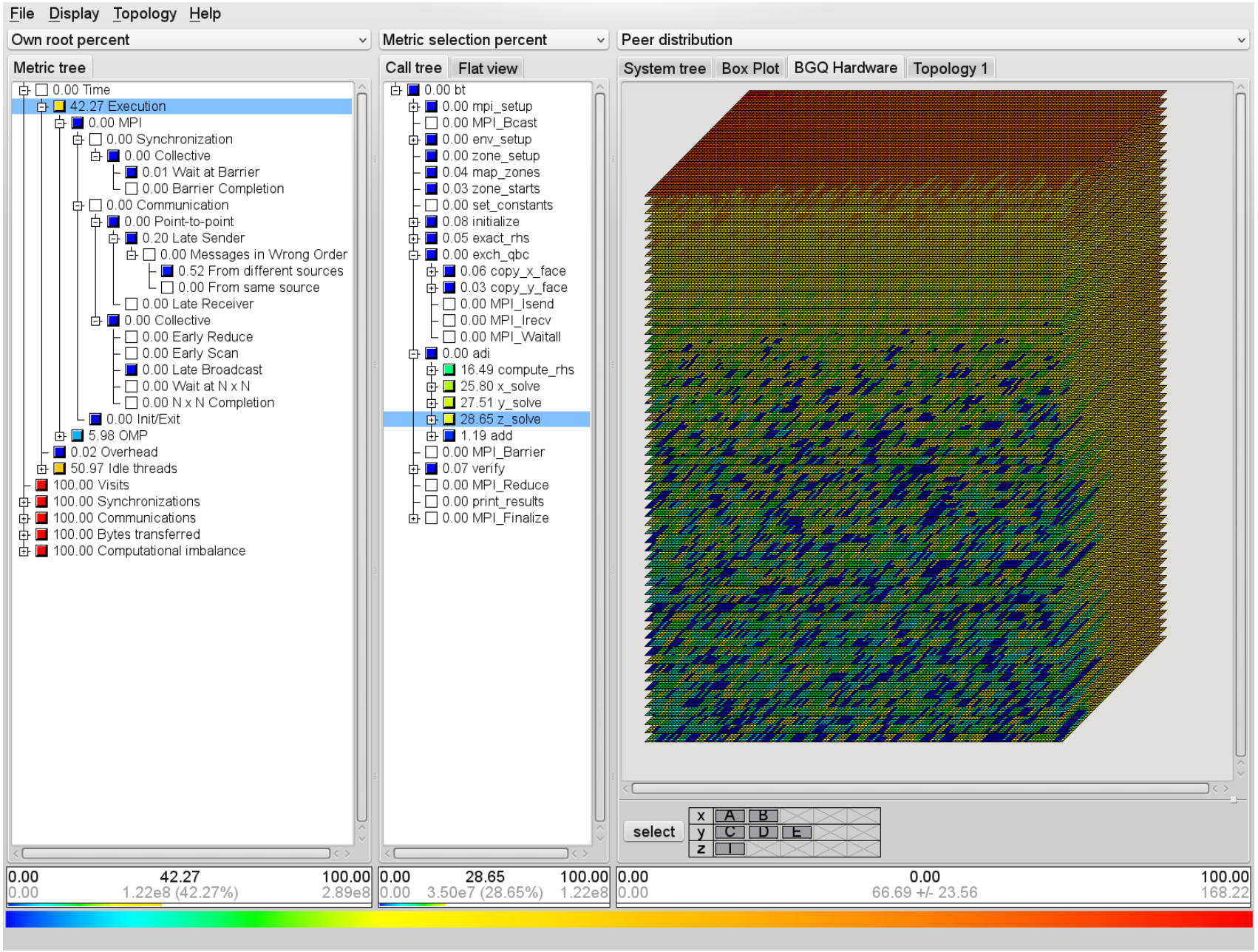
CUBE: > 3D ⇨ Folding (e.g. ABC-DECT)



CUBE: > 3D ⇨ Selecting (e.g. A,B,C,D1,E1,C6,T0)



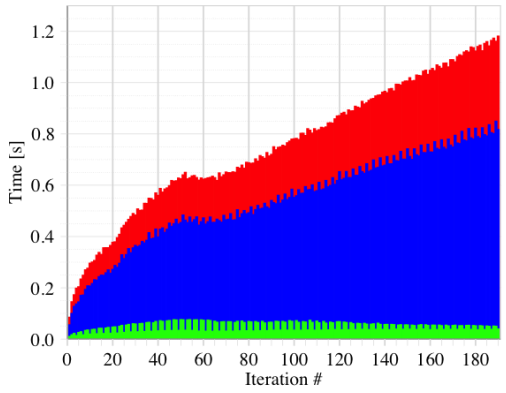
Trace analysis bt-mz@524,288 BGQ



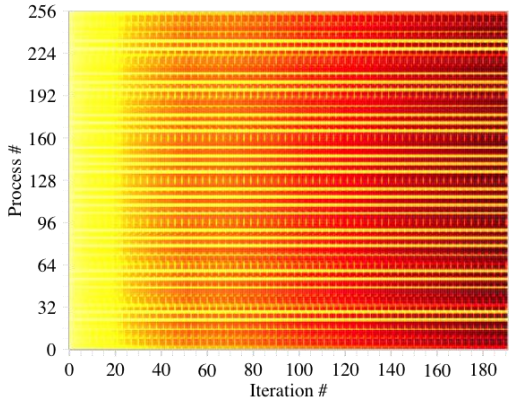
Time-series profiling

- Most simulation codes work iteratively
- Growing complexity of codes makes performance behavior more dynamic
  - Different frequency of certain calculations
  - Adaptation to changing state of computation
  - Changing load distribution
  - External influence (e.g., dynamic reconfiguration)

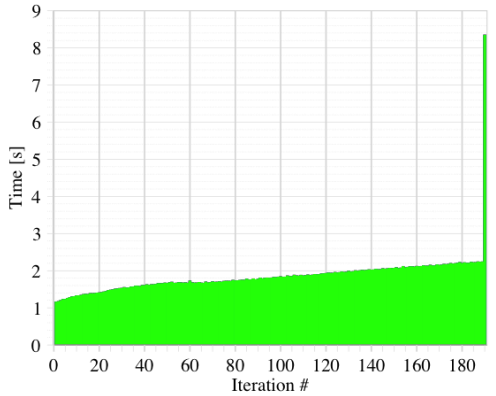
MPI point-to-point



MPI point-to-point



Execution



Time-series profiling

- Challenge – profile size grows with iteration count
- Solution – online compression

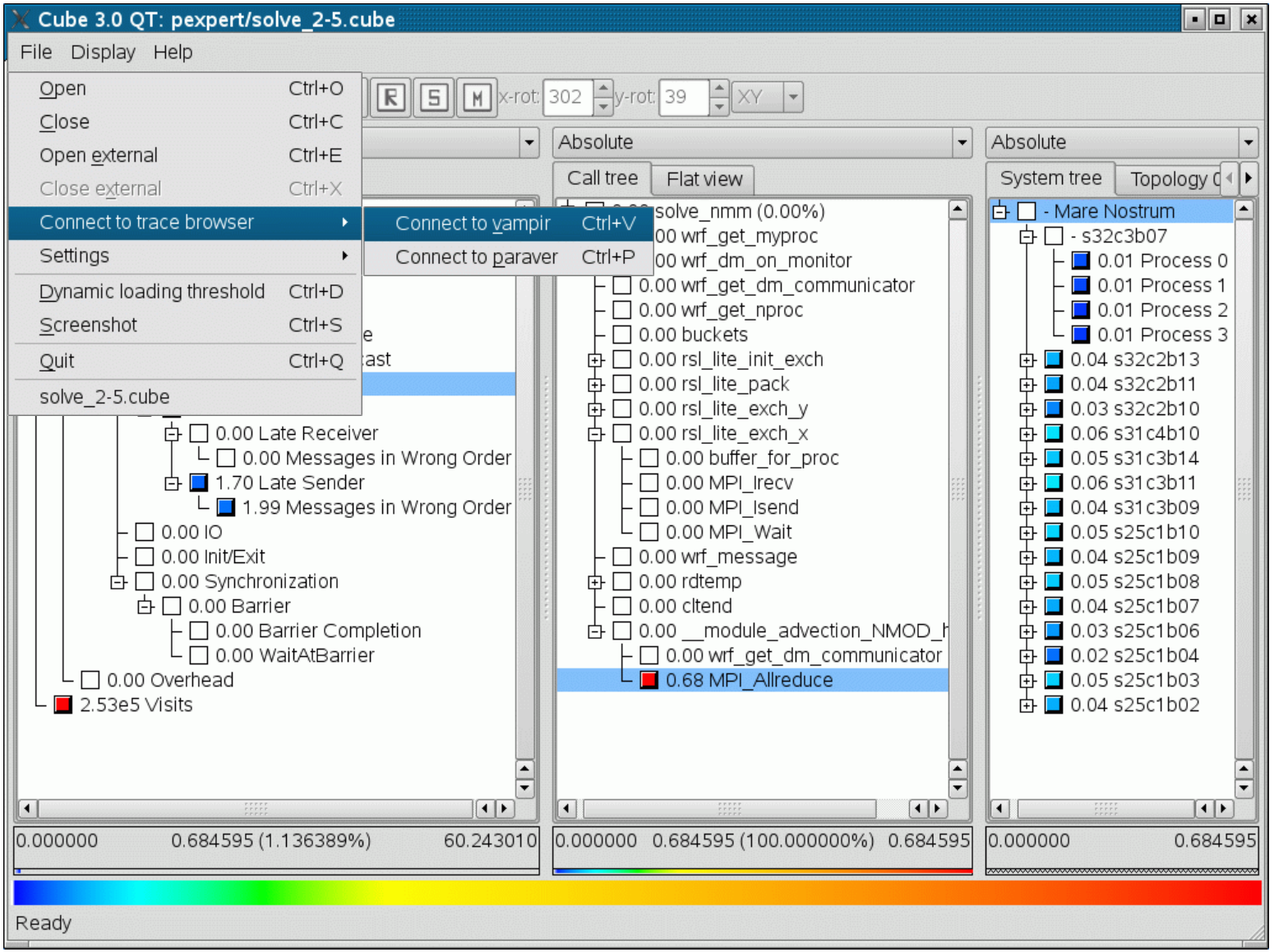
- Previous work
  - Prototypical online implementation for MPI
- Current objective
  - Create hybrid online version
- Status
  - Needs more work

Scalasca ⇨ Vampir/Paraver integration

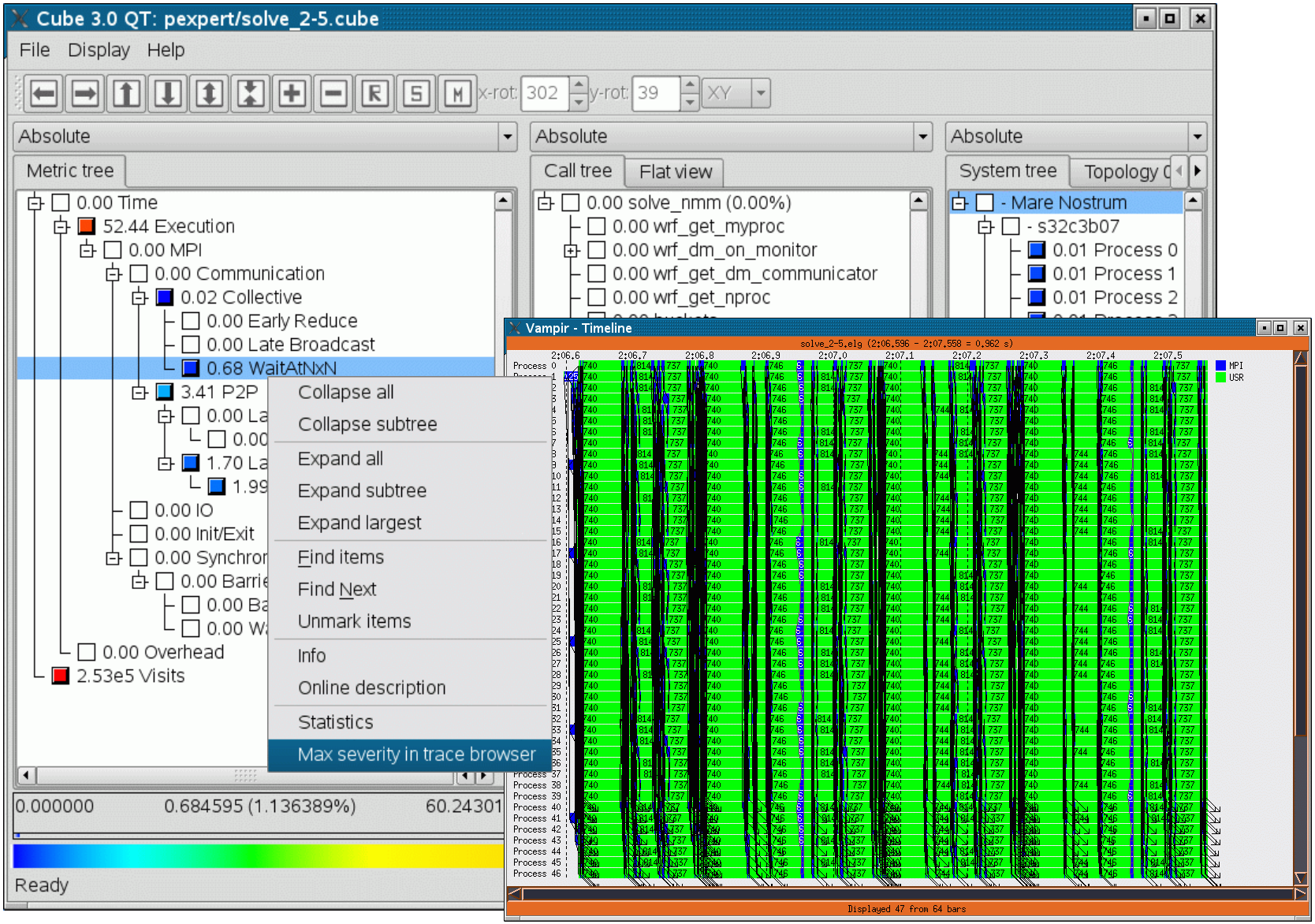
JÜLICH FORSCHUNGSZENTRUM

- Objective
  - Integration of Scalasca's interactive report explorer with Vampir or Paraver
  - Will allow the detailed investigation of the most severe instances of inefficiency patters identified by Scalasca
- Status
  - First version available as of Jul 2012 (V1.4.2)

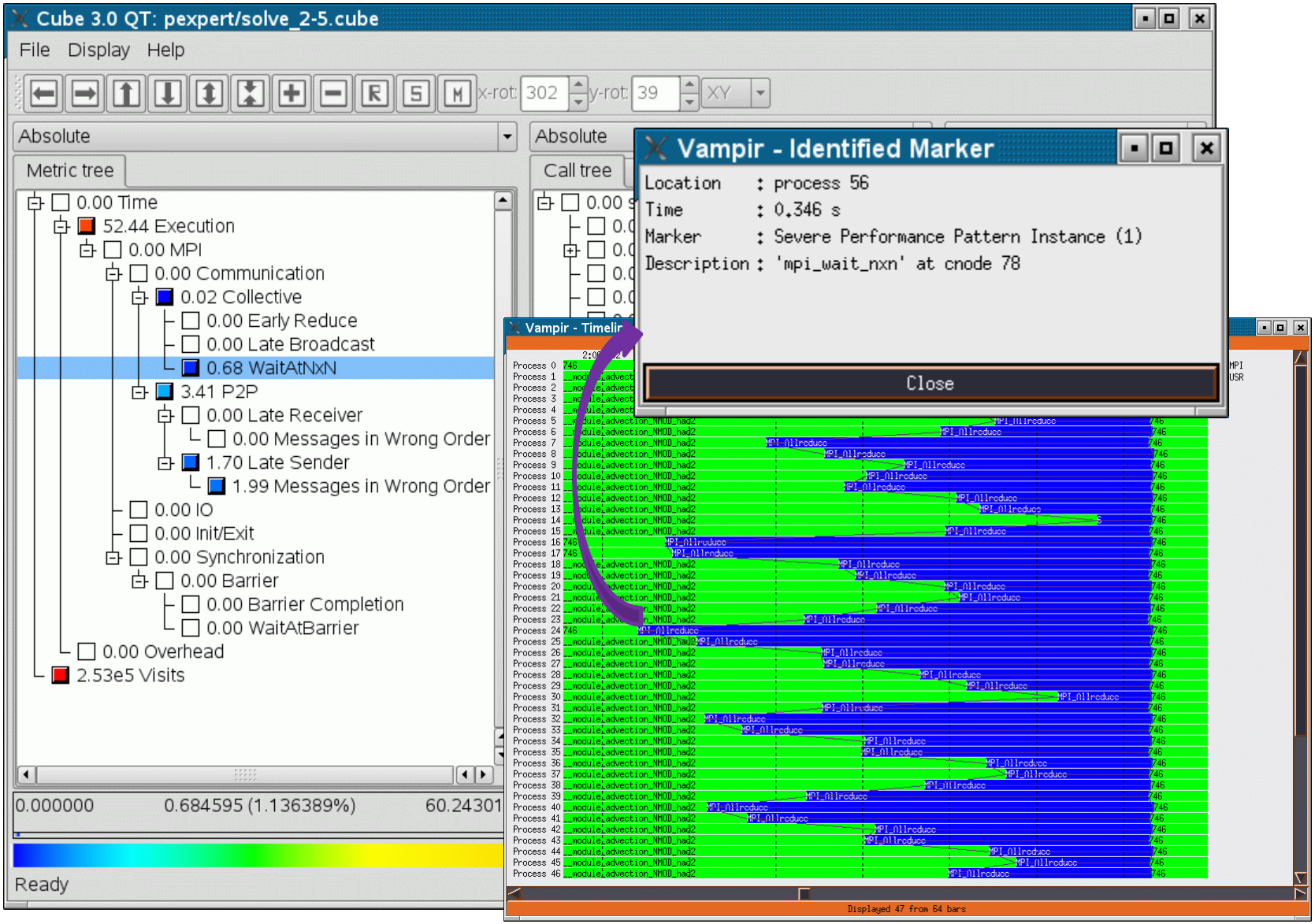
Scalasca ⇨ Vampir/Paraver integration



Scalasca ⇨ Vampir/Paraver integration

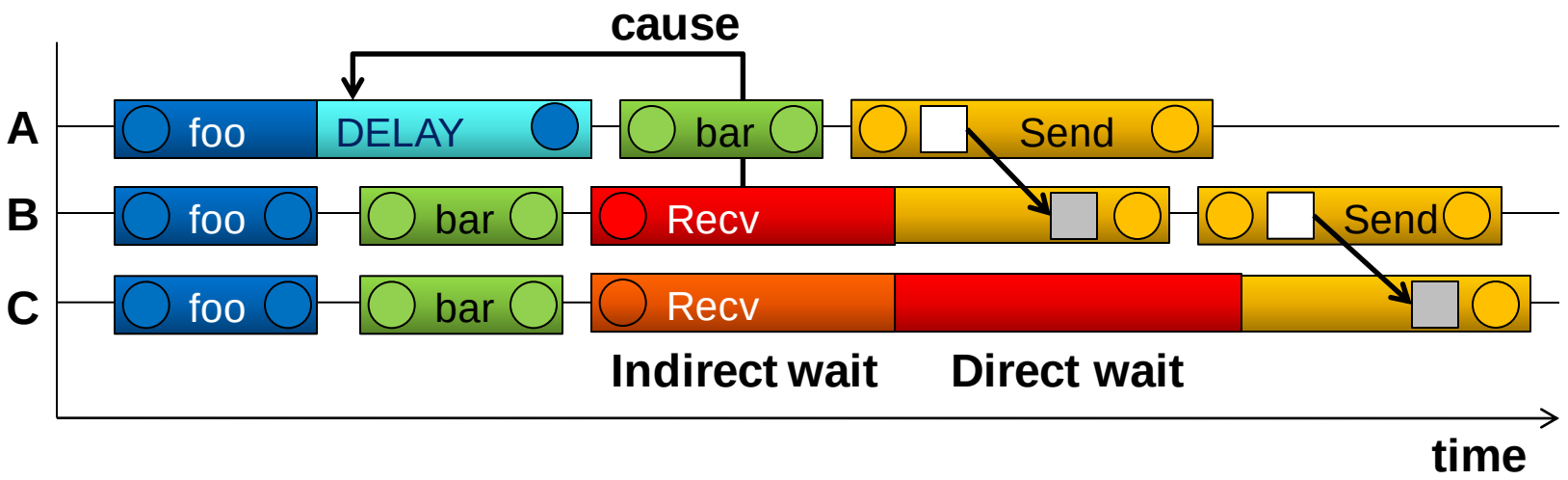


Scalasca ⇨ Vampir/Paraver integration

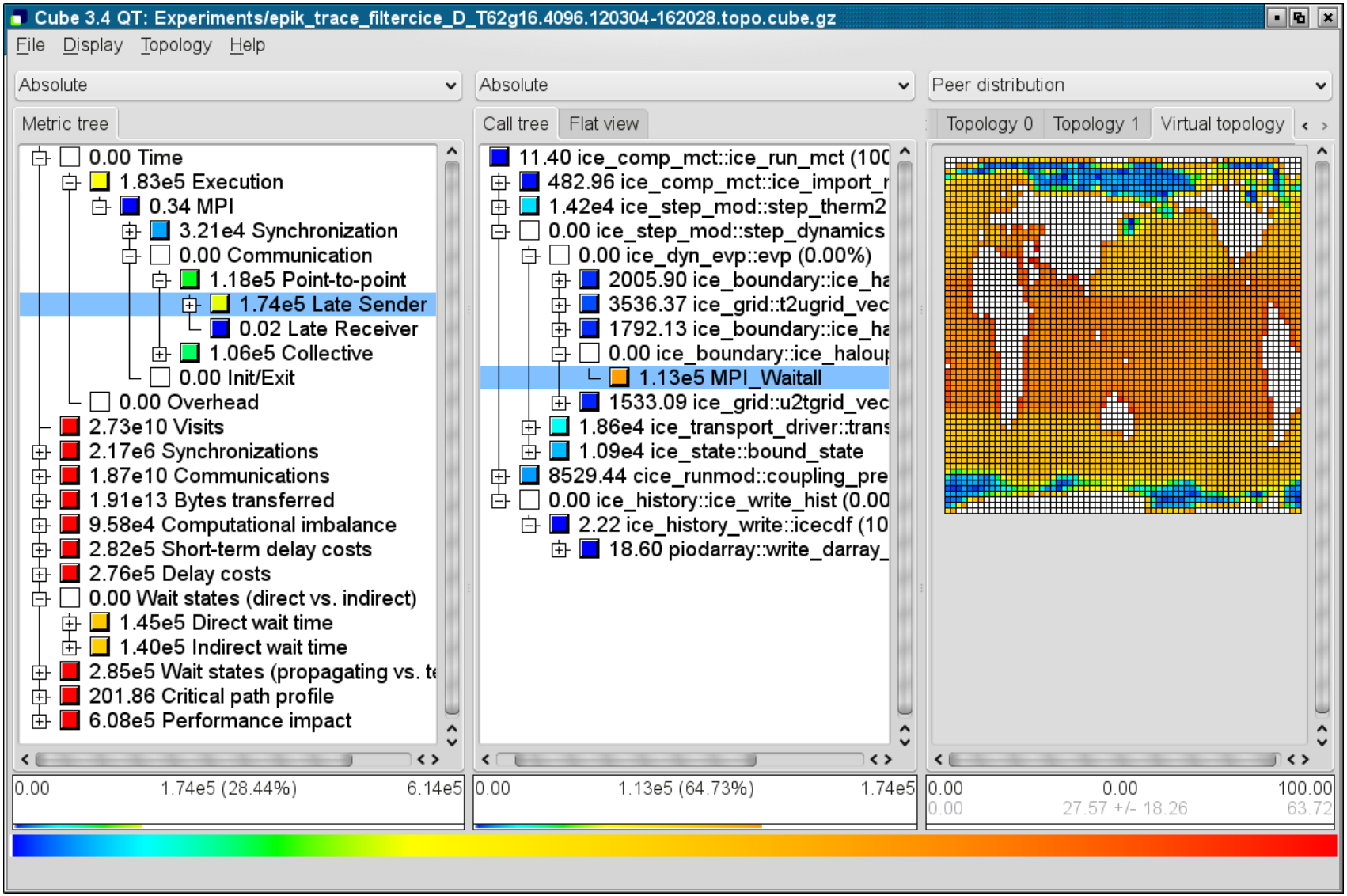


Root Cause Analysis

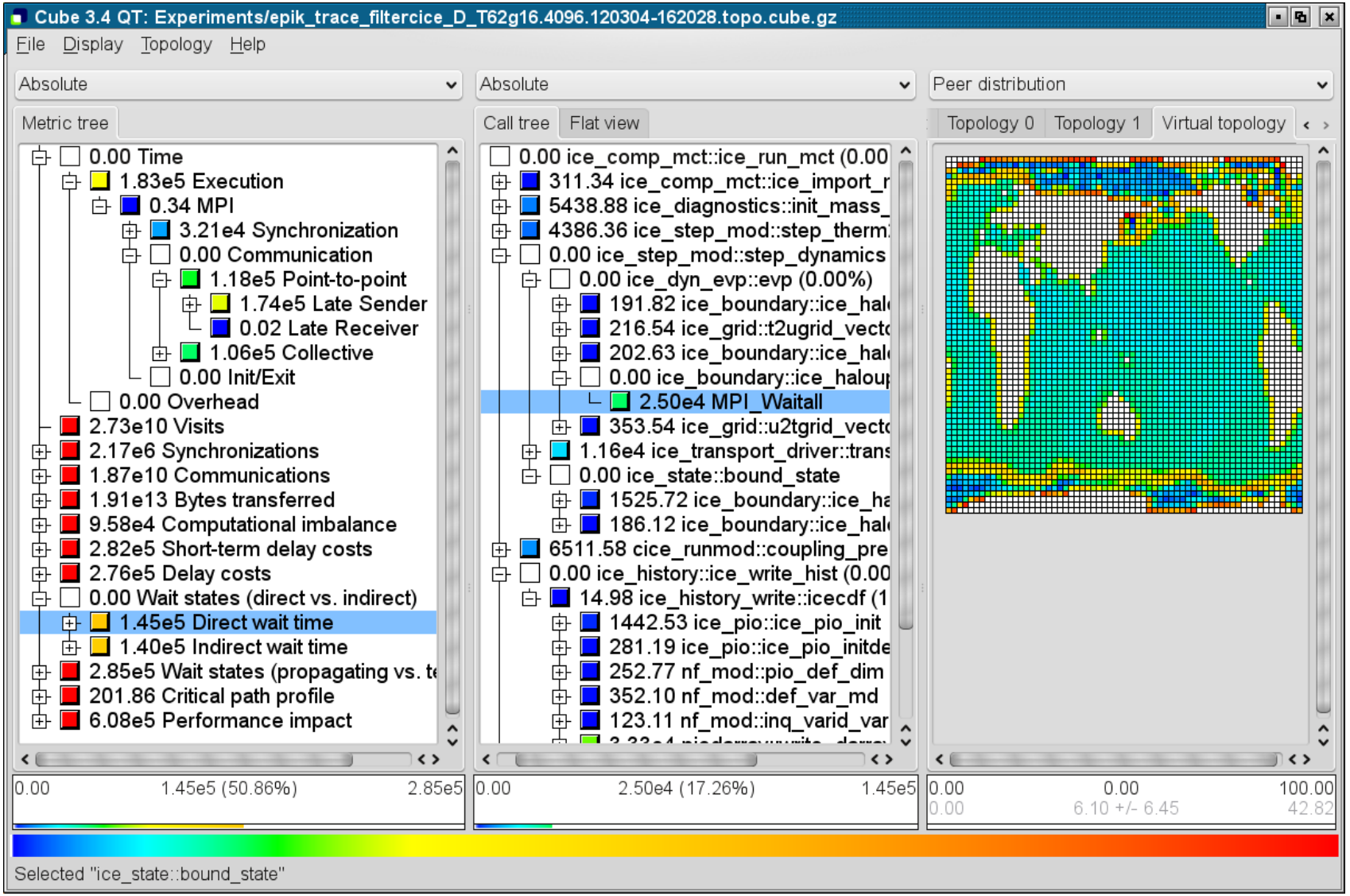
- **Root-cause analysis**
  - Wait states typically caused by load or communication imbalances earlier in the program
  - Waiting time can also propagate (e.g., indirect waiting time)
  - Goal: Enhance performance analysis to find the root cause of wait states
- **Approach (work in progress)**
  - Distinguish between direct and indirect waiting time
  - Identify call path/process combinations delaying other processes and causing first order waiting time
  - Identify original **delay**



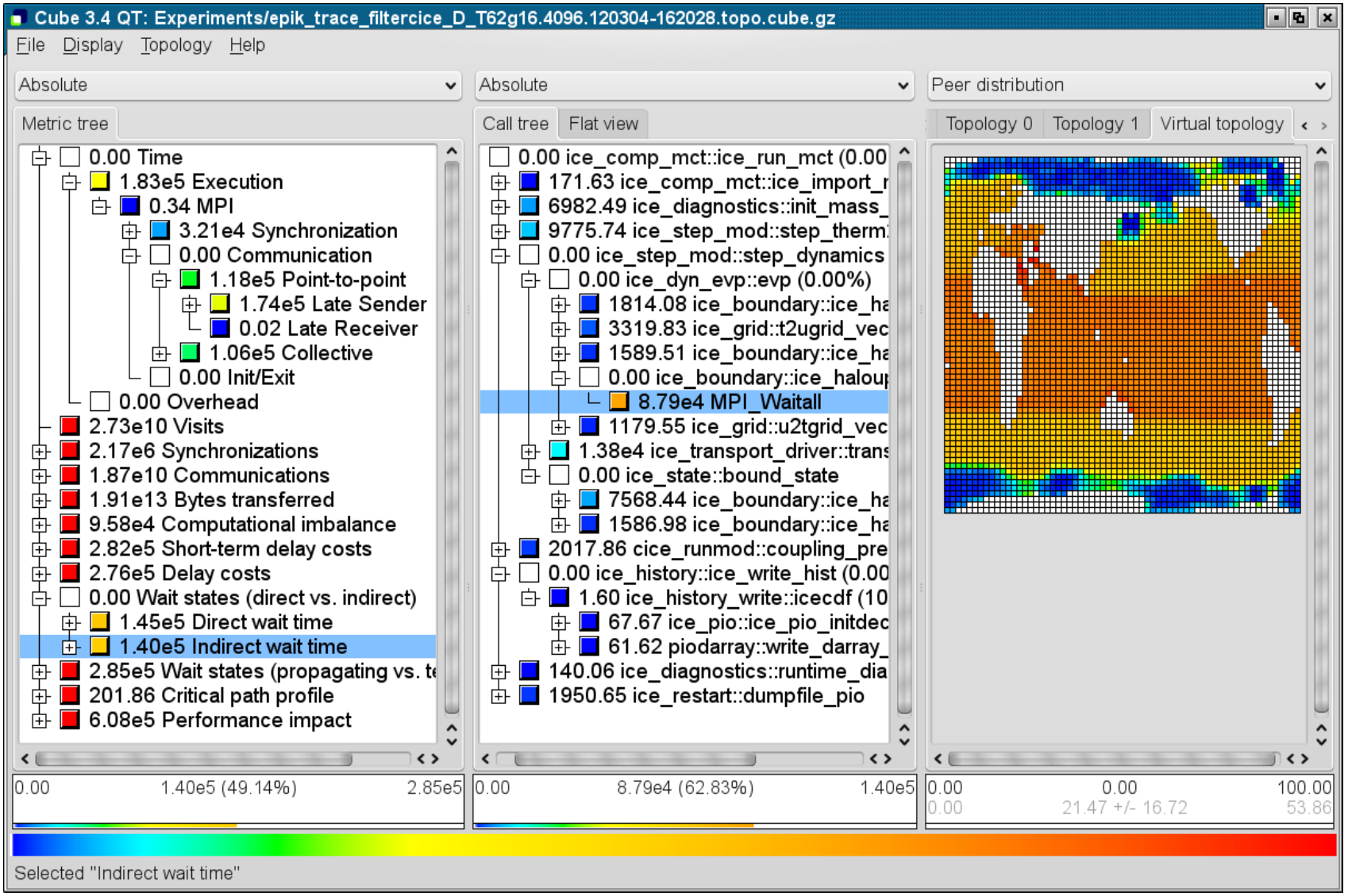
Example: CESM Sea Ice Module Late Sender



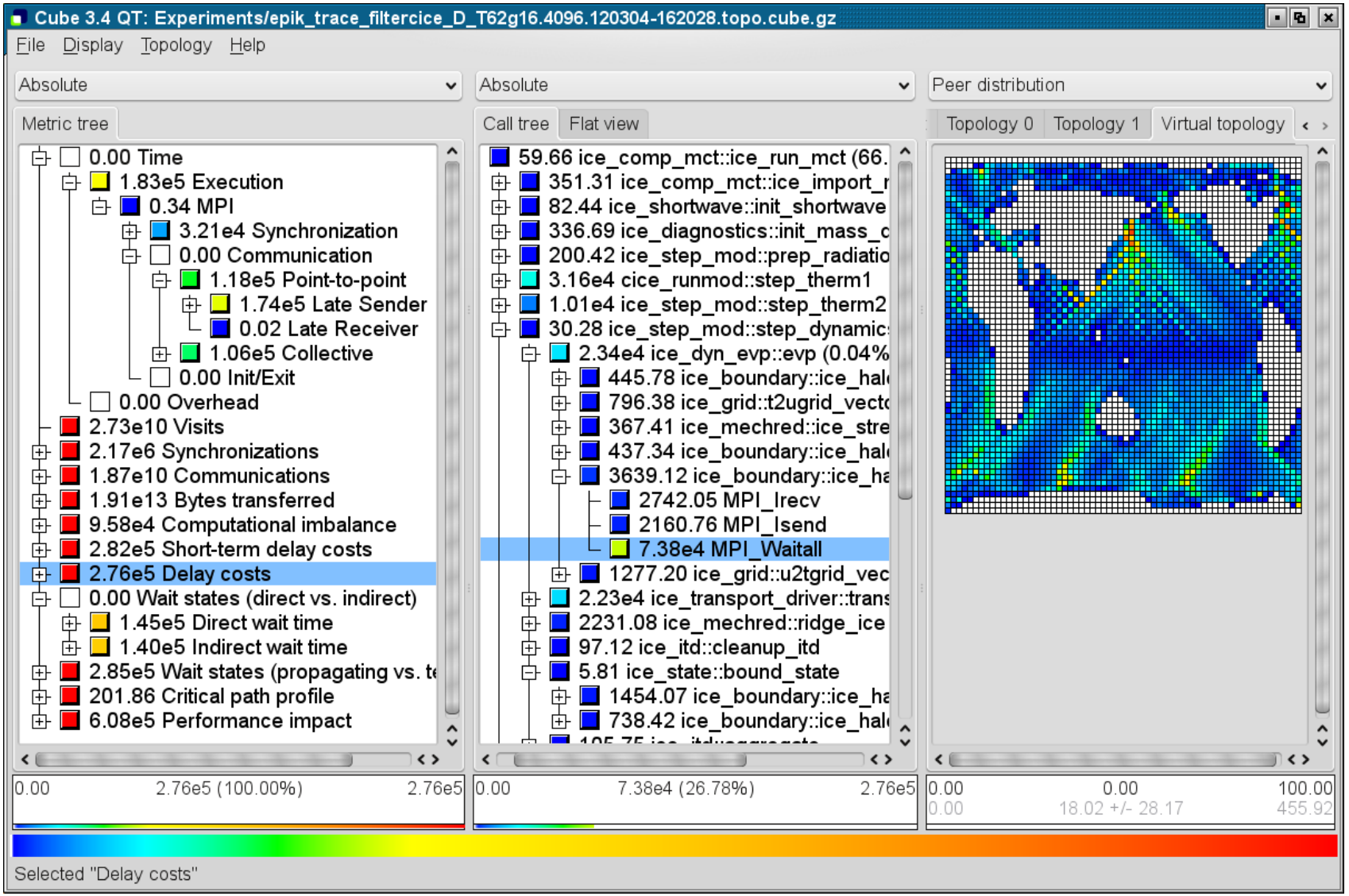
Example: CESM Sea Ice Module Direct Wait Time



Example: CESM Sea Ice Module Indirect Wait Time



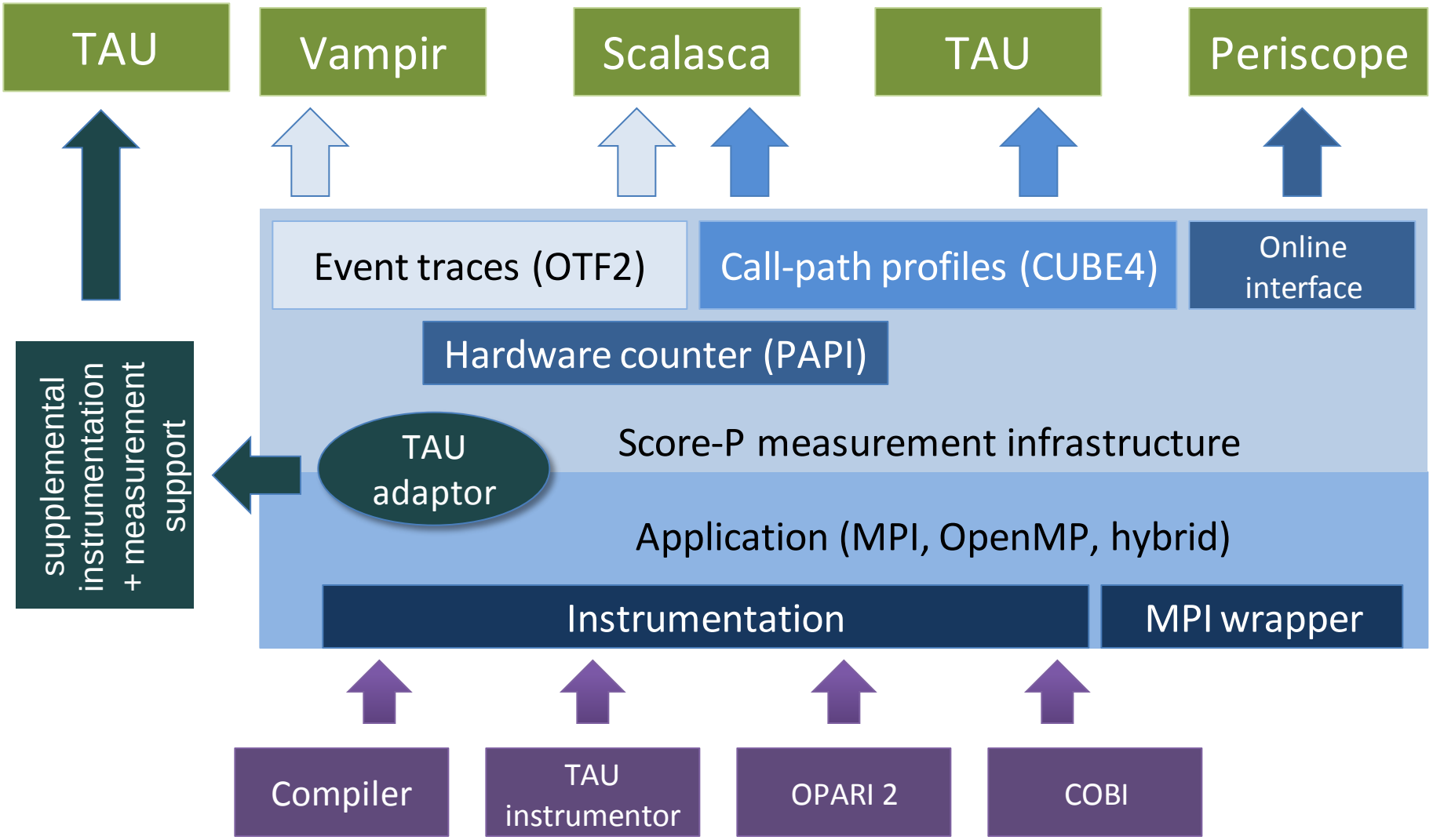
Example: CESM Sea Ice Module Delay Costs



Tool Integration: Score-P Objectives

- Mainly funded by SILC, PRIMA, LMAC projects
- Make common part of Periscope, Scalasca, TAU, and Vampir a community effort
  - **Score-P measurement system**
- **Functional requirements**
  - Performance data: profiles (CUBE4), traces (OTF2)
  - Initially direct instrumentation, later also sampling
  - Offline and online access
  - Metrics: time, communication metrics and hardware counters
  - Initially MPI 2 and OpenMP 3, later also CUDA and OpenCL
- Current release: V1.0.2 of June 2012 (1.1 soon)

Score-P Architecture



Score-P Partners

- Forschungszentrum Jülich, Germany
- German Research School for Simulation Sciences, Aachen, Germany
- Gesellschaft für numerische Simulation mbH Braunschweig, Germany
- RWTH Aachen, Germany
- Technische Universität Dresden, Germany
- Technische Universität München, Germany
- University of Oregon, Eugene, USA

Funded Integration Projects

- **SILC (01/2009 to 12/2011)**
  - Unified measurement system (Score-P) for Vampir, Scalasca, Periscope
- **PRIMA (08/2009 to 08/2012)**
  - Integration of TAU and Scalasca
- **LMAC (08/2011 to 07/2013)**
  - Evolution of Score-P
  - Analysis of performance dynamics
- **H4H (10/2010 to 09/2013)**
  - Hybrid programming for heterogeneous platforms
- **HOPSA (02/2011 to 01/2013)**
  - Integration of system and application monitoring

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

ITEA 2

INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

SEVENTH FRAMEWORK PROGRAMME

MINISTRY OF EDUCATION AND SCIENCE OF THE RUSSIAN FEDERATION

The Scalasca Team

- **JSC**


Markus Geimer


Christian Heinrich


Michael Knobloch


Daniel Lorenz


Bernd Mohr


Peter Philippen


Christian Rössel


Marc Schlütter


Pavel Saviankou


Alexandre Strube


Brian Wylie


Anke Visser


Ilja Zhukov

GRS


David Böhme


Marc-André Hermanns


Zoltán Szebenyi


Felix Wolf

- **Sponsors**

HELMHOLTZ ASSOCIATION

ITEA 2
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

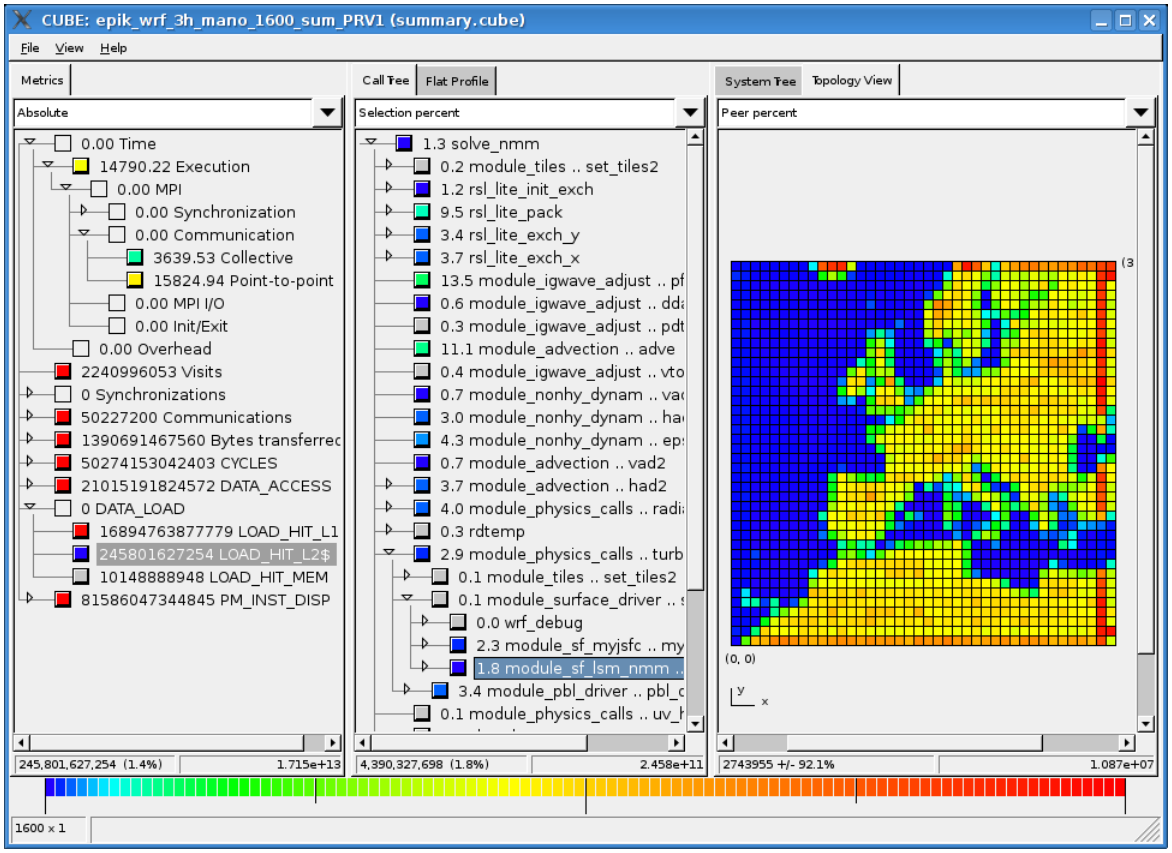
SEVENTH FRAMEWORK PROGRAMME

Deutsche Forschungsgemeinschaft DFG

Bundesministerium für Bildung und Forschung

DEPARTMENT OF ENERGY · UNITED STATES OF AMERICA

Thank you!



WRF-NMM weather prediction code on MareNostrum @ 1600 CPUs

scalasca

http://www.scalasca.org

scalasca@fz-juelich.de

Questions?

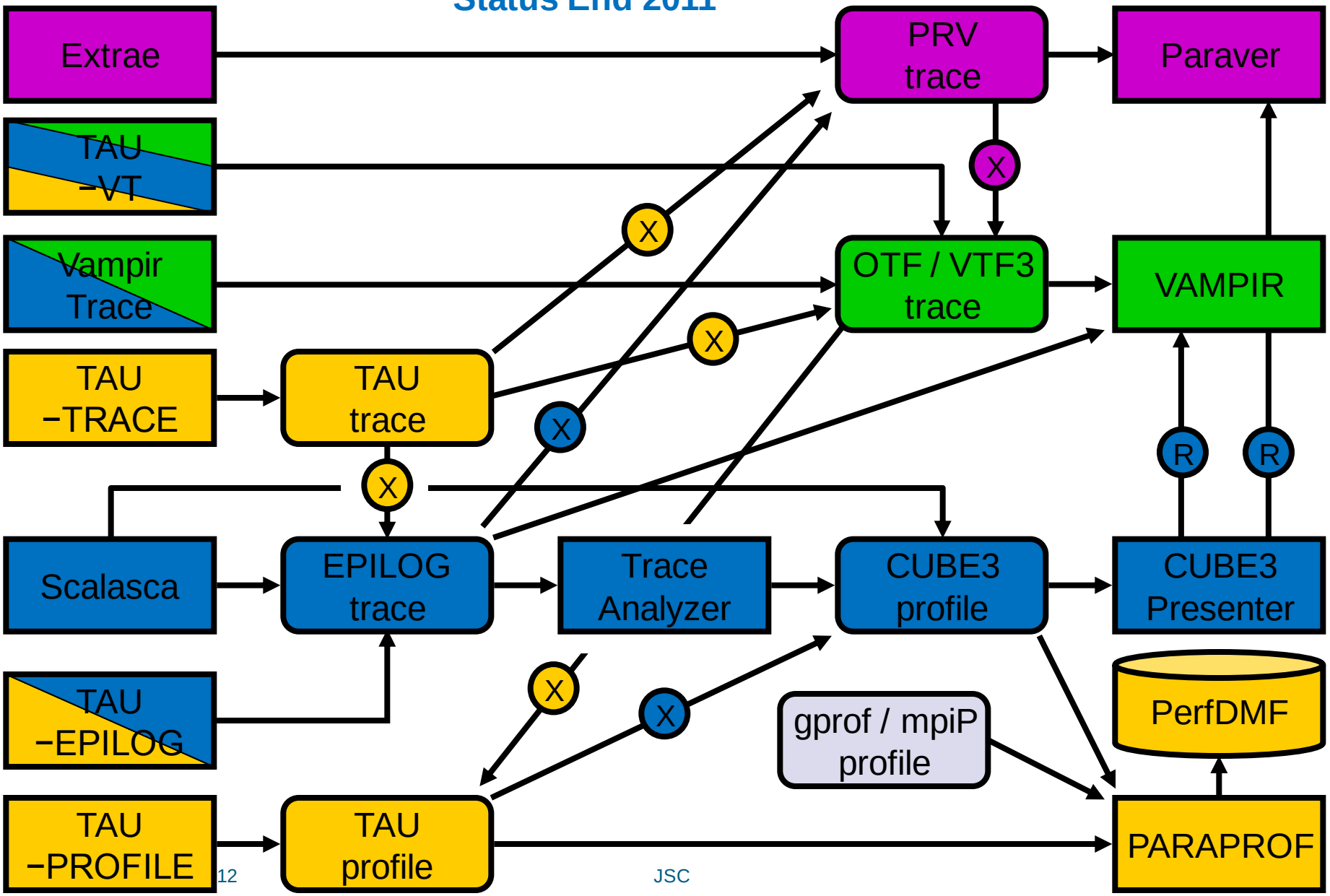
JÜLICH
FORSCHUNGSZENTRUM

BACKUP SLIDES

Scalasca ⇔ TAU ⇔ VAMPIR ⇔ Paraver

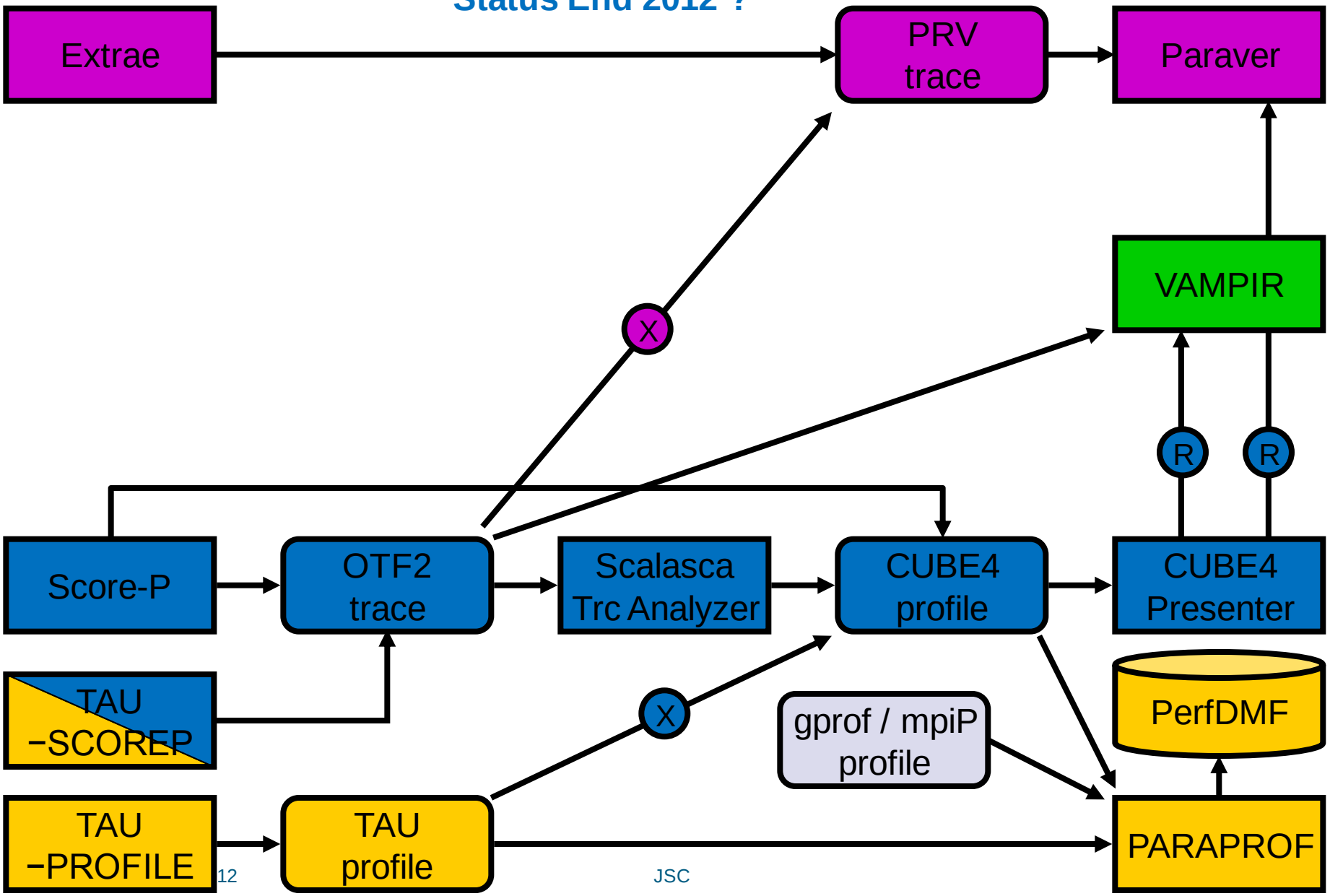
JÜLICH FORSCHUNGSZENTRUM

Status End 2011



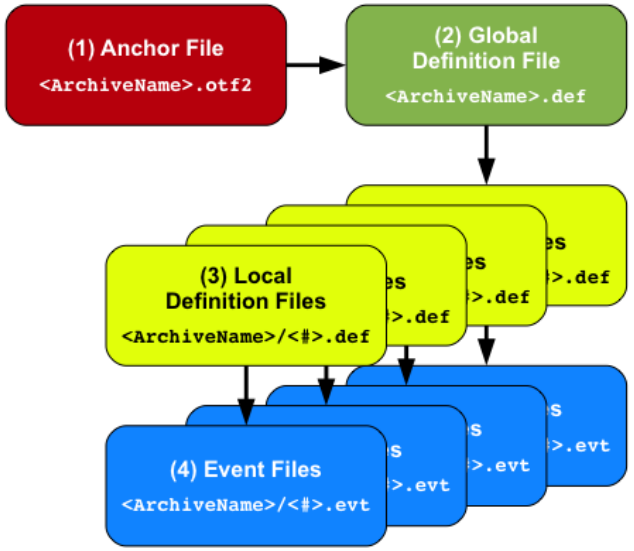
Scalasca ⇔ TAU ⇔ VAMPIR ⇔ Paraver

Status End 2012 ?



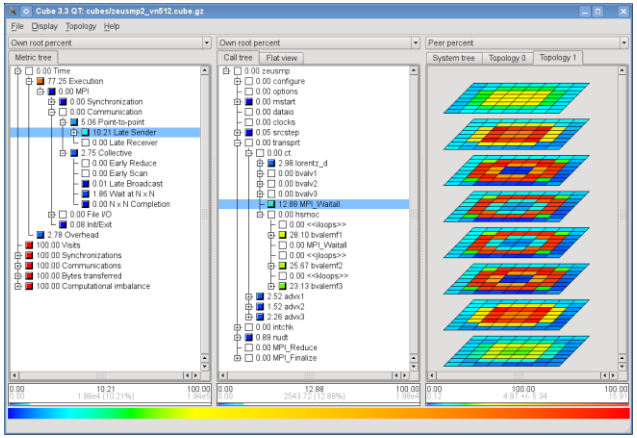
OTF-2 Tracing Format

- Successor to OTF and EPILOG
- Same basic structure as OTF, EPILOG, or other formats
- Design goals
  - High scalability
  - Low overhead (storage space and processing time)
  - Good read/write performance
    - Reduced number of files during initial writing via SIONlib
  - Compatibility reader for OTF and Epilog formats
  - Extensibility



CUBE-4 Profiling Format

- Latest version of a family of profiling formats
  - Still under development, to be released soon
- Representation of three-dimensional performance space
  - Metric, call path, process or thread
- File organization
  - Metadata stored as XML file
  - Metric values stored in binary format
    - Two files per metric:
      data + index for storage-efficient
      sparse representation
- Optimized for
  - High write bandwidth
  - Fast interactive analysis through incremental loading



II-40