JÜLICH
FORSCHUNGSZENTRUM

The Score-P ecosystem and Scalasca

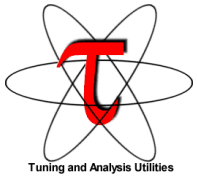
2015-06-24 | Markus Geimer
Jülich Supercomputing Centre
m.geimer@fz-juelich.de

Challenges of increasing parallelism

- Optimization of applications more difficult
  - More CPUs / multi-core / heterogeneity
  - Every doubling of scale reveals new bottlenecks
- Also new demands on performance tools
  - ***Scalable*** to keep up with HPC systems and applications
  - ***Efficient*** to meet performance expectations
  - ***Effective to use*** so that programmer productivity is maximized
  - ***Interoperable*** to perform different analyses on a single measurement data set

HPC performance tool landscape

- **TAU**
  - University of Oregon, US
  - http://tau.uoregon.edu



- **Scalasca**
  - JSC, GRS, TU Darmstadt, Germany
  - http://www.scalasca.org

- **HPCToolkit**
  - Rice University, US
  - http://hpctoolkit.org

- **Open|SpeedShop**
  - Krell Institute, US
  - http://www.openspeedshop.org

- **Extrae/Paraver**
  - BSC, Spain
  - http://www.bsc.es/paraver

- **Periscope**
  - TU Munich, Germany
  - http://www.lrr.in.tum.de/~periscop/

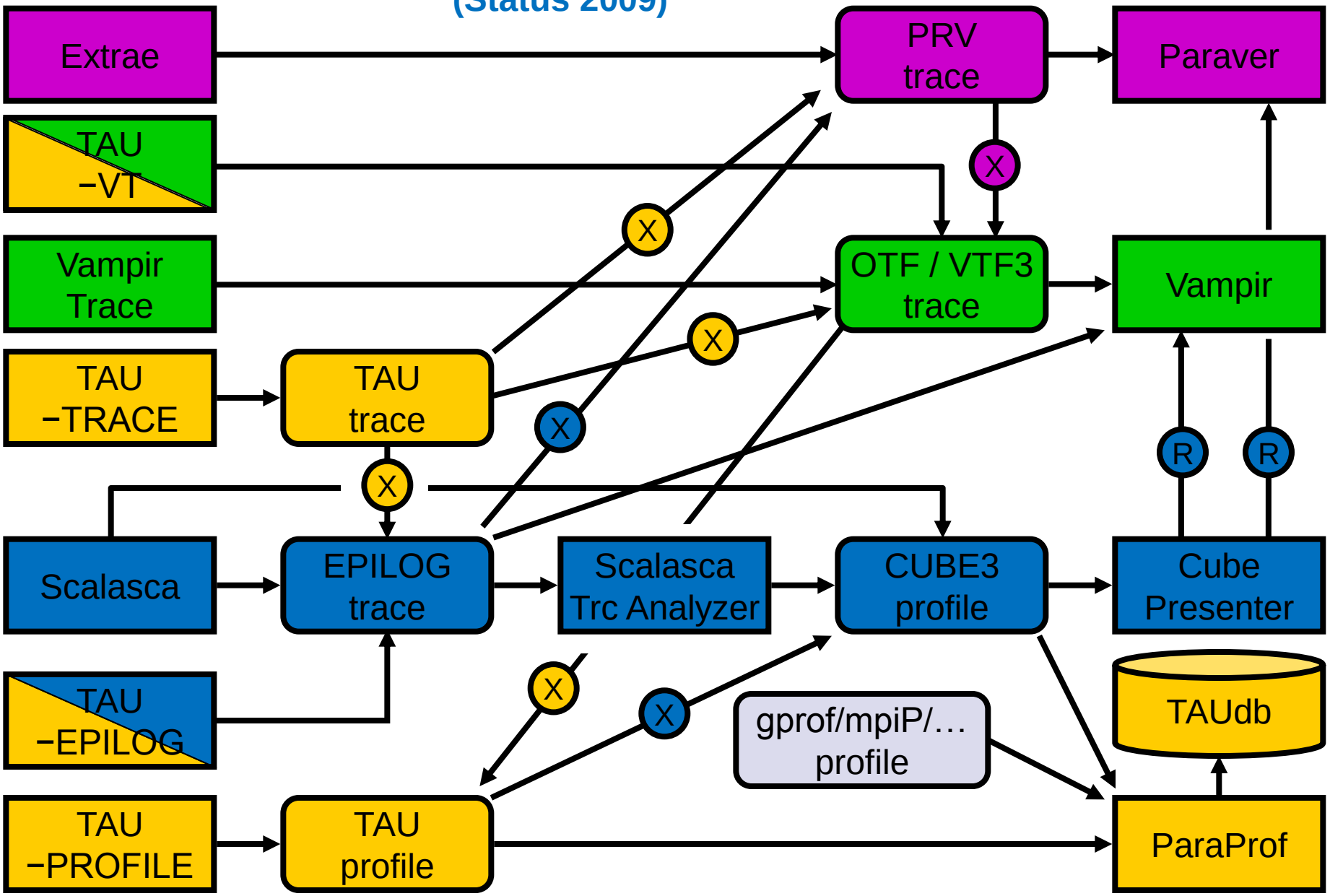
- **VampirTrace/Vampir**
  - TU Dresden, Germany
  - http://www.vampir.eu

- **… and many others**
  - Open source
  - Vendor specific

Scalasca ⇔ TAU ⇔ Vampir ⇔ Paraver

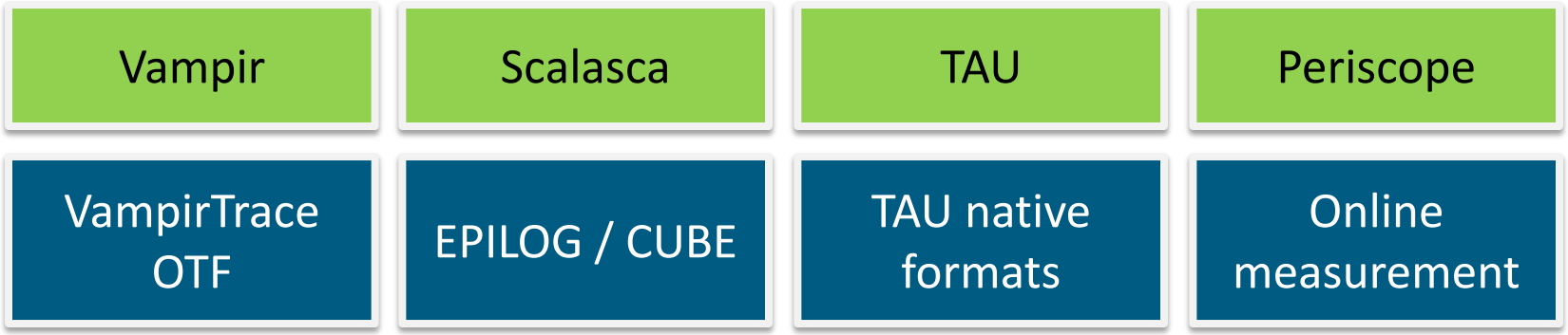
JÜLICH FORSCHUNGSZENTRUM

(Status 2009)



Fragmentation of tools landscape

- Several performance tools co-exist
- Separate measurement systems and output formats
- Complementary features and overlapping functionality
- Redundant effort for development and maintenance
- Limited or expensive interoperability
- Complications for user experience, support, training



Integration projects

- SILC (2009-2011)
  - Develop a common, next-gen measurement infrastructure for Scalasca, Vampir and Periscope incl. open data formats
  - Partners: JSC, GRS, TUD, TUM, RWTH, GNS

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

- PRIMA (2009-2013)
  - Reengineering core components of TAU and Scalasca
  - Tighter integration and improved interoperability
  - Partners: UOregon, JSC

Joining forces

- Partners of both consortia quickly agreed to collaborate
  - Recognized overlapping goals & benefits from sharing resources, ideas, and designs
  - Key: Early communication

- Funding agencies were open-minded, too
  - BMBF approved UOregon as associated partner in SILC
  - DOE did not object either

JÜLICH
FORSCHUNGSZENTRUM

A community instrumentation and measurement infrastructure

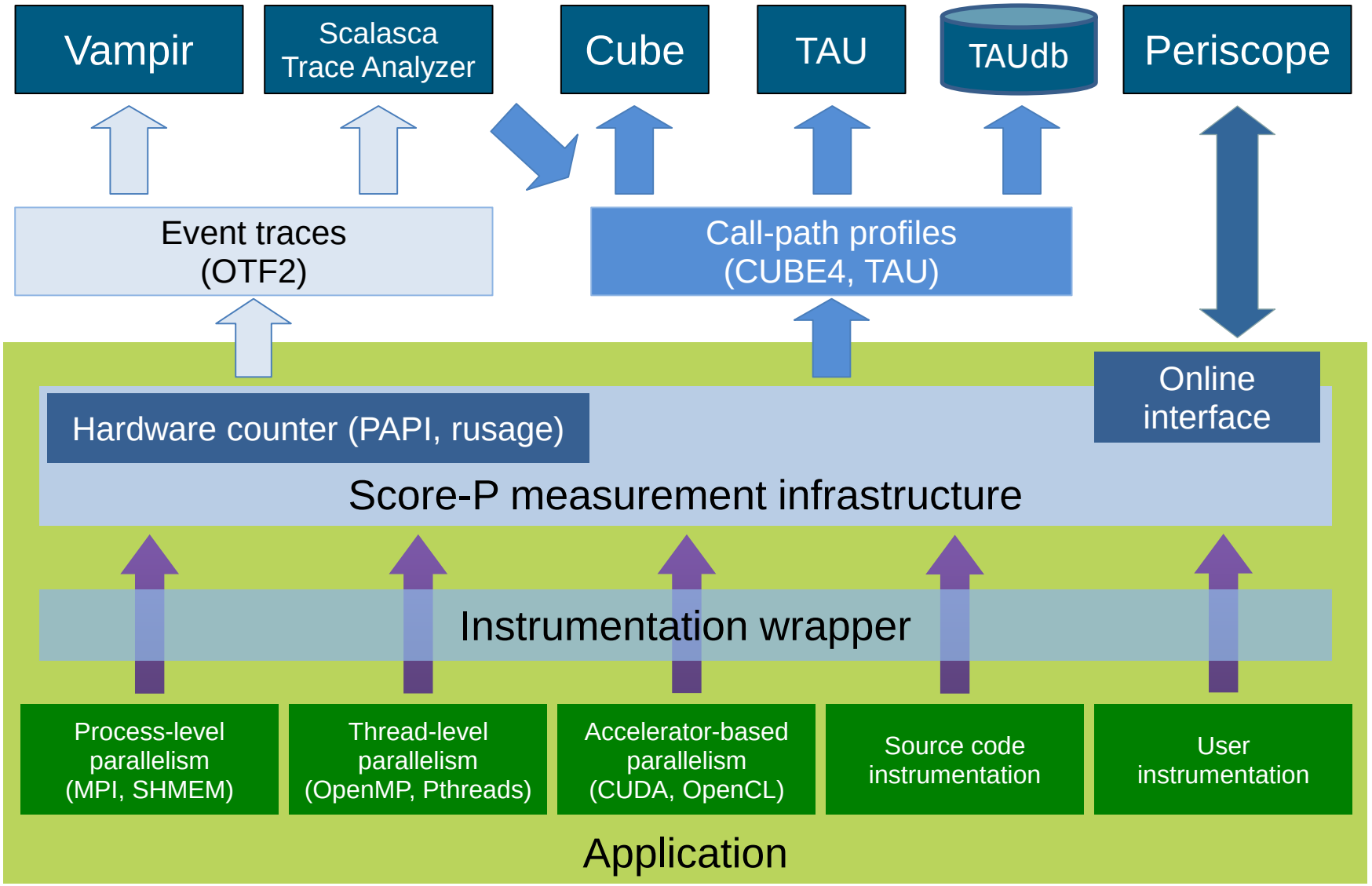
The Score-P idea

- Start a community effort for a common infrastructure
  - Score-P instrumentation and measurement system
  - Common data formats OTF2 and CUBE4
  - Initial scalability target: Petascale systems
- Developer perspective:
  - Save manpower by sharing development resources
  - Invest in new analysis functionality and scalability
  - Save efforts for maintenance, testing, porting, support, training
- User perspective:
  - Single learning curve
  - Single installation, fewer version updates
  - Interoperability and data exchange

Score-P functionality

- Provide typical functionality for HPC performance tools
- Instrumentation (various methods)
  - User code (automatic, manual)
  - Multi-process paradigms (MPI, SHMEM)
  - Thread-parallel paradigms (OpenMP, POSIX threads)
  - Accelerator-based paradigms (CUDA, OpenCL)
  - And their combination
- Flexible measurement configuration without re-compilation
  - Basic and advanced profile generation
  - Event trace recording
  - Online access to profiling data
- Highly scalable I/O functionality
- Support all fundamental concepts of partner's tools

Score-P ecosystem



Score-P features and status

- Open source (3-clause BSD license)
- Fairly portable
  - IBM Blue Gene, Cray XC series, Fujitsu FX & K computer, IBM SP & blade clusters, SGI Altix, Linux clusters, Xeon Phi
- Supports common parallel programming paradigms & languages
  - Fortran, C, C++
  - MPI 2.2, OpenMP 3.0, POSIX threads, SHMEM, CUDA, OpenCL and combinations
- Scalable generation of call-path profiles and event traces
  - Successful profile measurements of ~1 million concurrent threads
  - Successful trace measurements of ~650k concurrent threads
- Current release: 1.4.2 (June 2015)

http://www.score-p.org

Future features and management

- Scalability to maximum available CPU core count
- Support for sampling & binary instrumentation
- Support for new programming models, e.g., PGAS
- Support for new architectures

- Ensure a single official release version at all times which will always work with the tools
- Allow experimental versions for new features or research

- Commitment to joint long-term cooperation
  - Development based on meritocratic governance model
  - Open for contributions and new partners

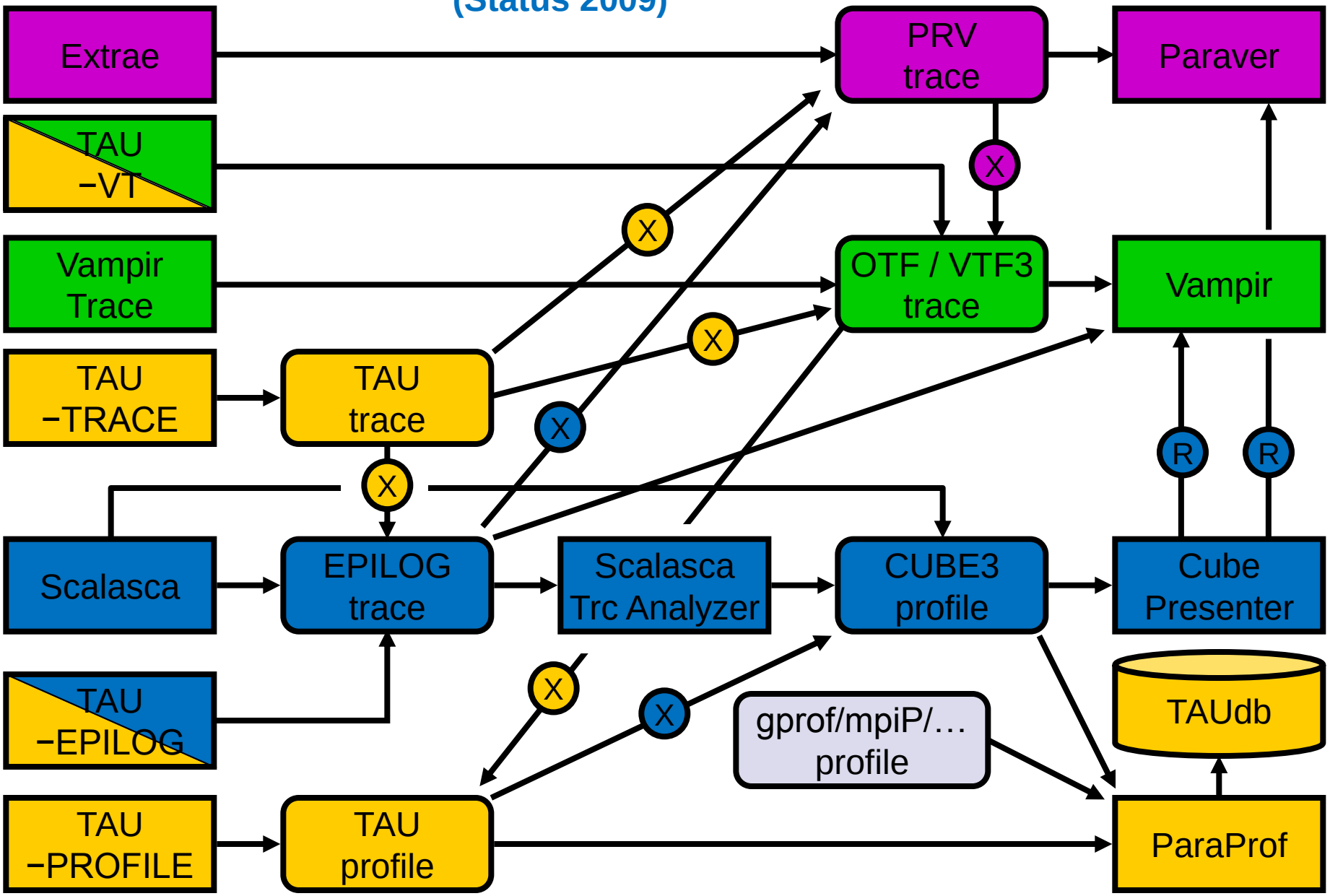
Current Score-P consortium partners

- Forschungszentrum Jülich GmbH (JSC)
- German Research School for Simulation Sciences
- Gesellschaft für numerische Simulation mbH
- RWTH Aachen University
- Technische Universität Dresden
- Technische Universität Darmstadt
- Technische Universität München
- University of Oregon

Scalasca ⇔ TAU ⇔ Vampir ⇔ Paraver

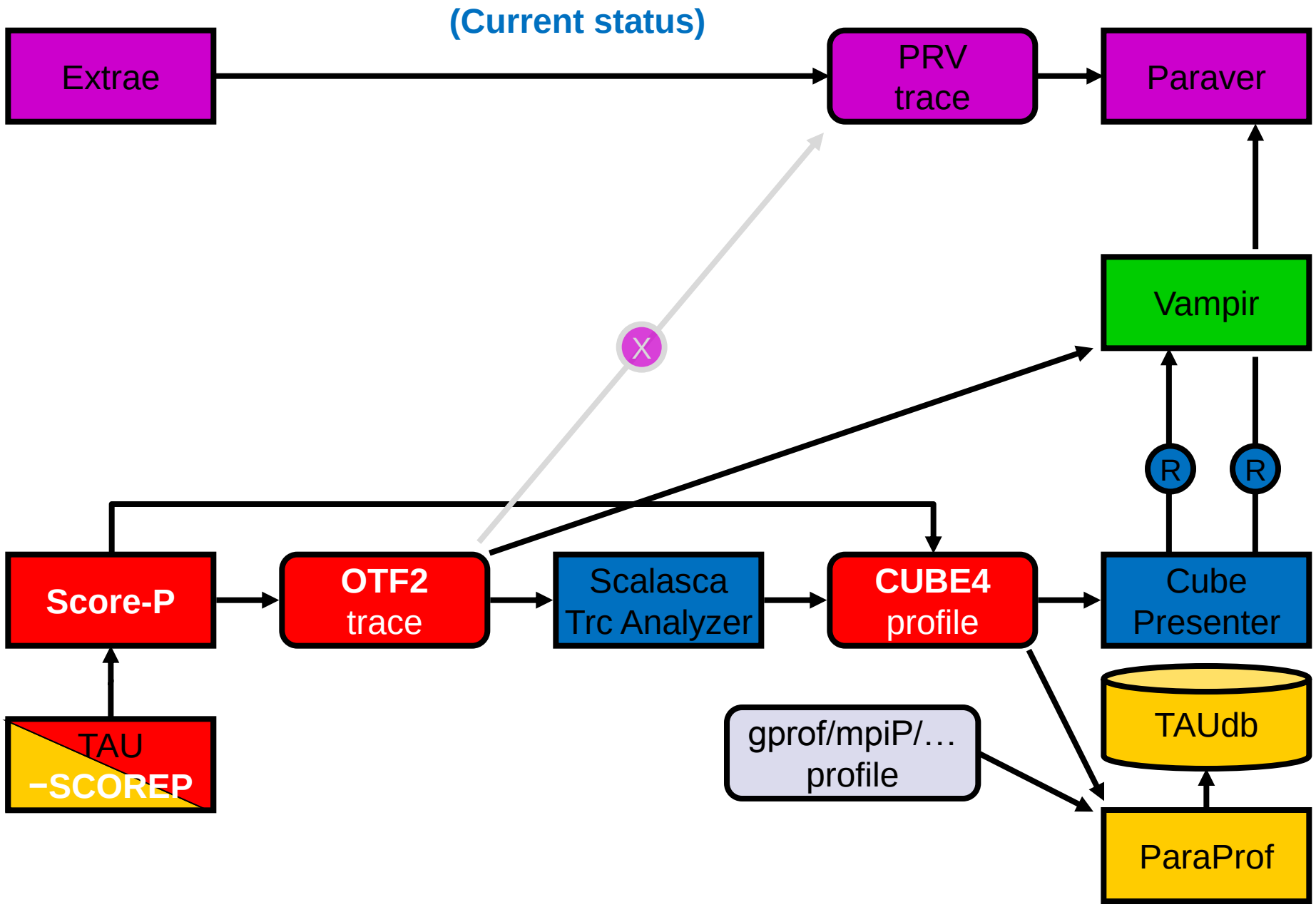
JÜLICH FORSCHUNGSZENTRUM

(Status 2009)



Score-P ⇔ Scalasca ⇔ TAU ⇔ Vampir ⇔ Paraver

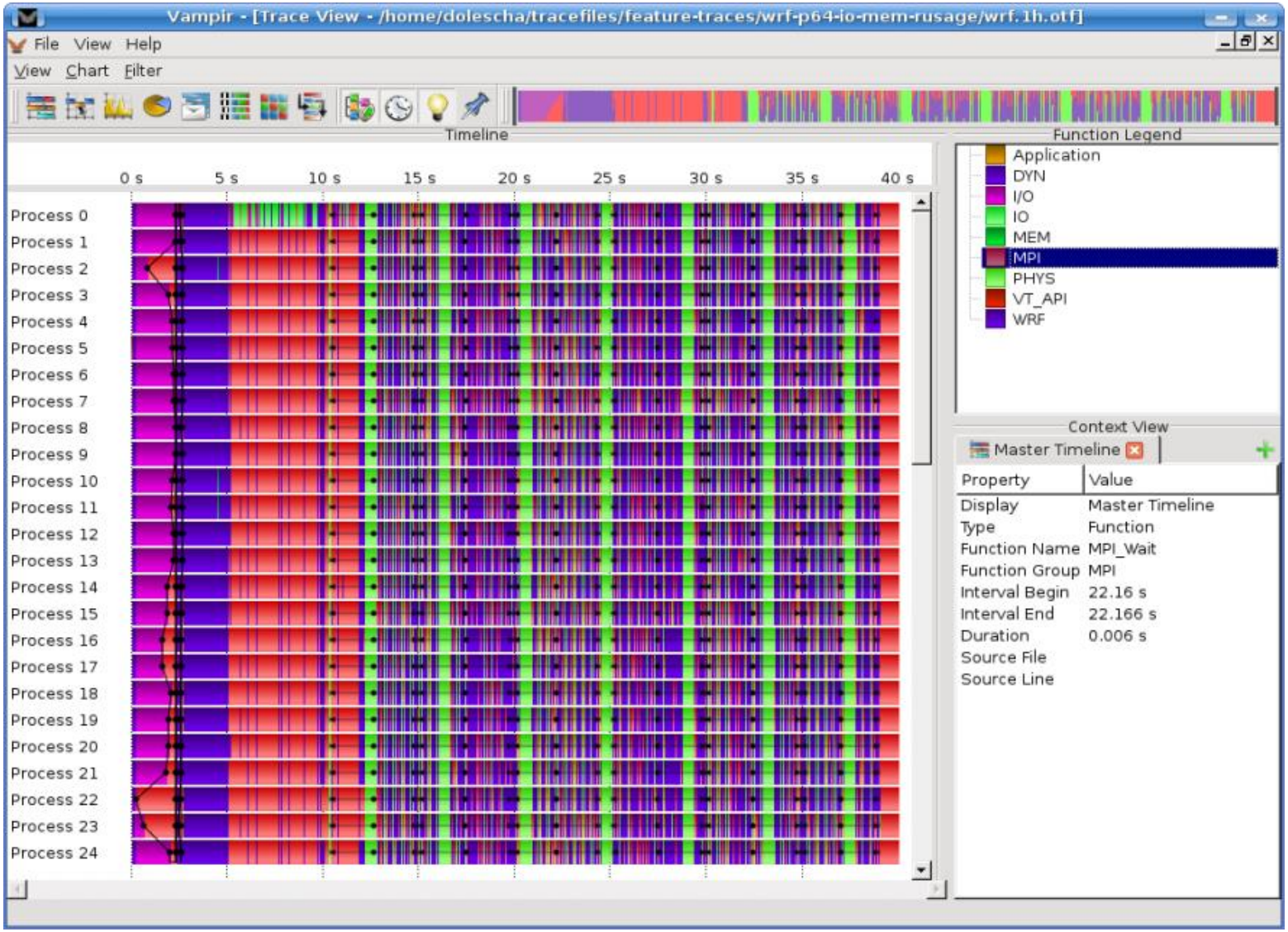
JÜLICH FORSCHUNGSZENTRUM



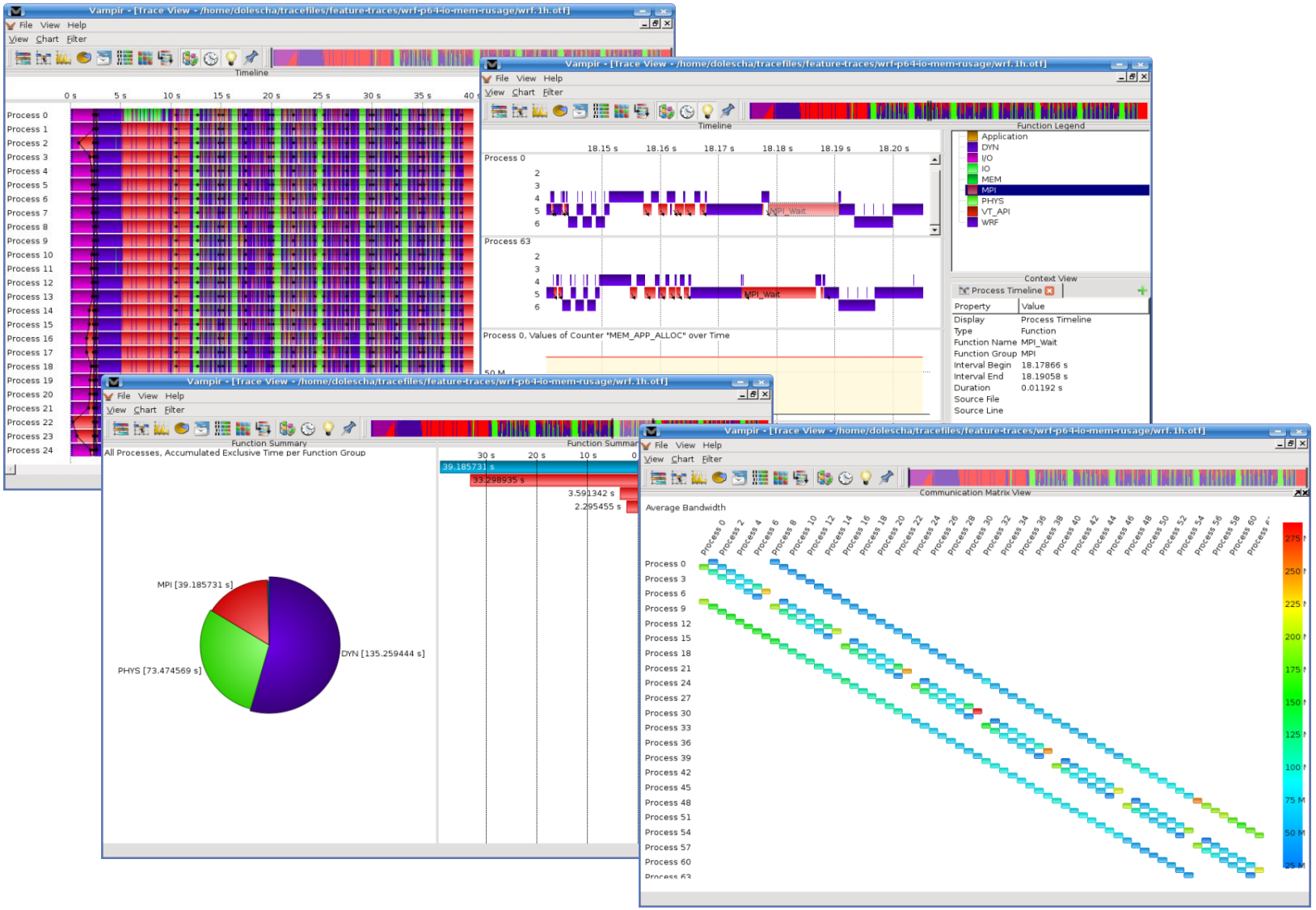
JÜLICH
FORSCHUNGSZENTRUM

A scalable performance analysis toolset

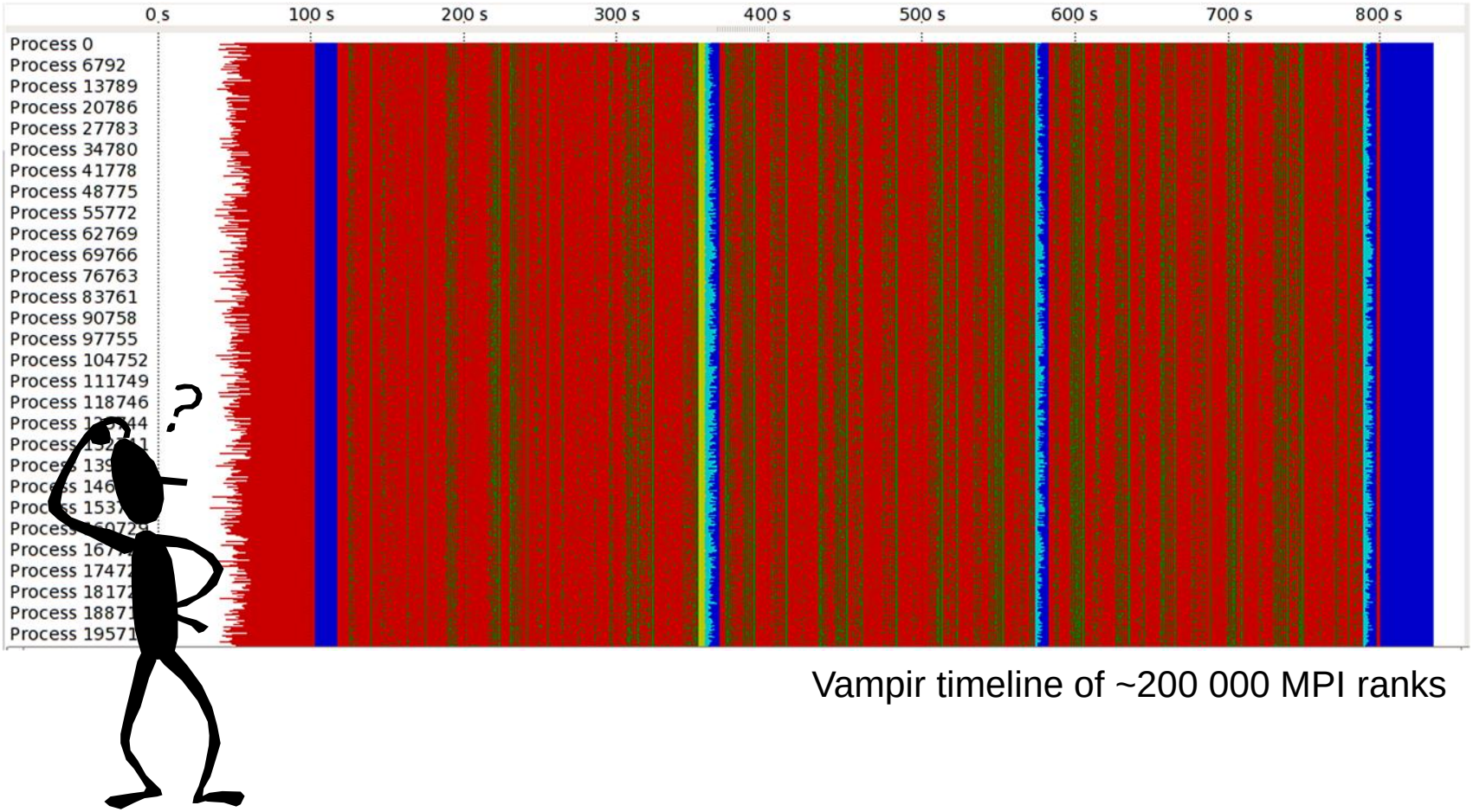
A picture is worth a thousand words…



But how do I locate the bottleneck?



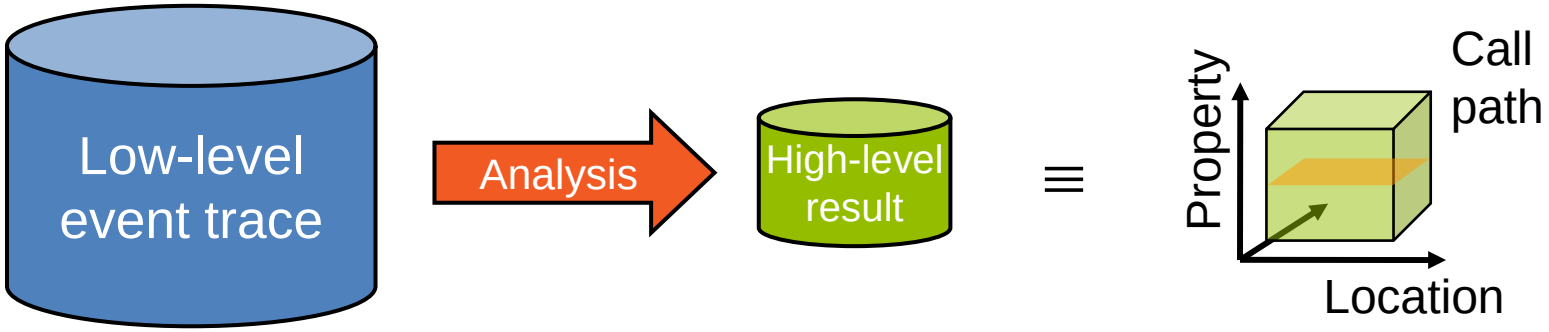
Especially in *really huge* traces?



Vampir timeline of ~200 000 MPI ranks

Automatic trace analysis

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



- Advantages
  - Guaranteed to cover the entire event trace
  - Quicker than manual/visual trace analysis
  - Helps to identify hot-spots for in-depth manual analysis
    - Complements the functionality of other tools

The Scalasca project: Overview

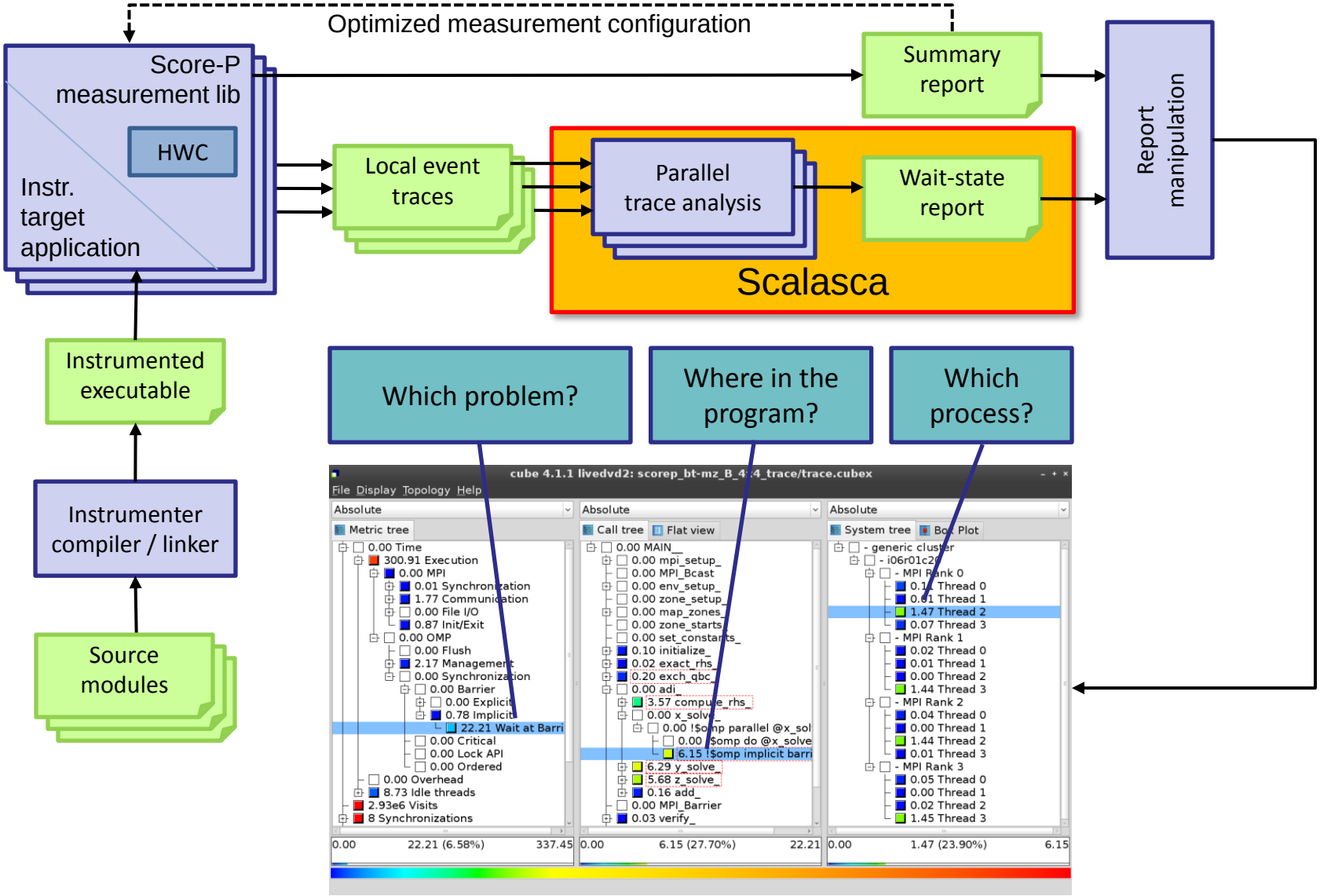
- Project started in 2006
  - Initial funding by Helmholtz Initiative & Networking Fund
  - Many follow-up projects
- Follow-up to pioneering KOJAK project (started 1998)
  - Automatic pattern-based trace analysis
- Now joint development of
  - Jülich Supercomputing Centre
  - German Research School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

Scalasca features and status

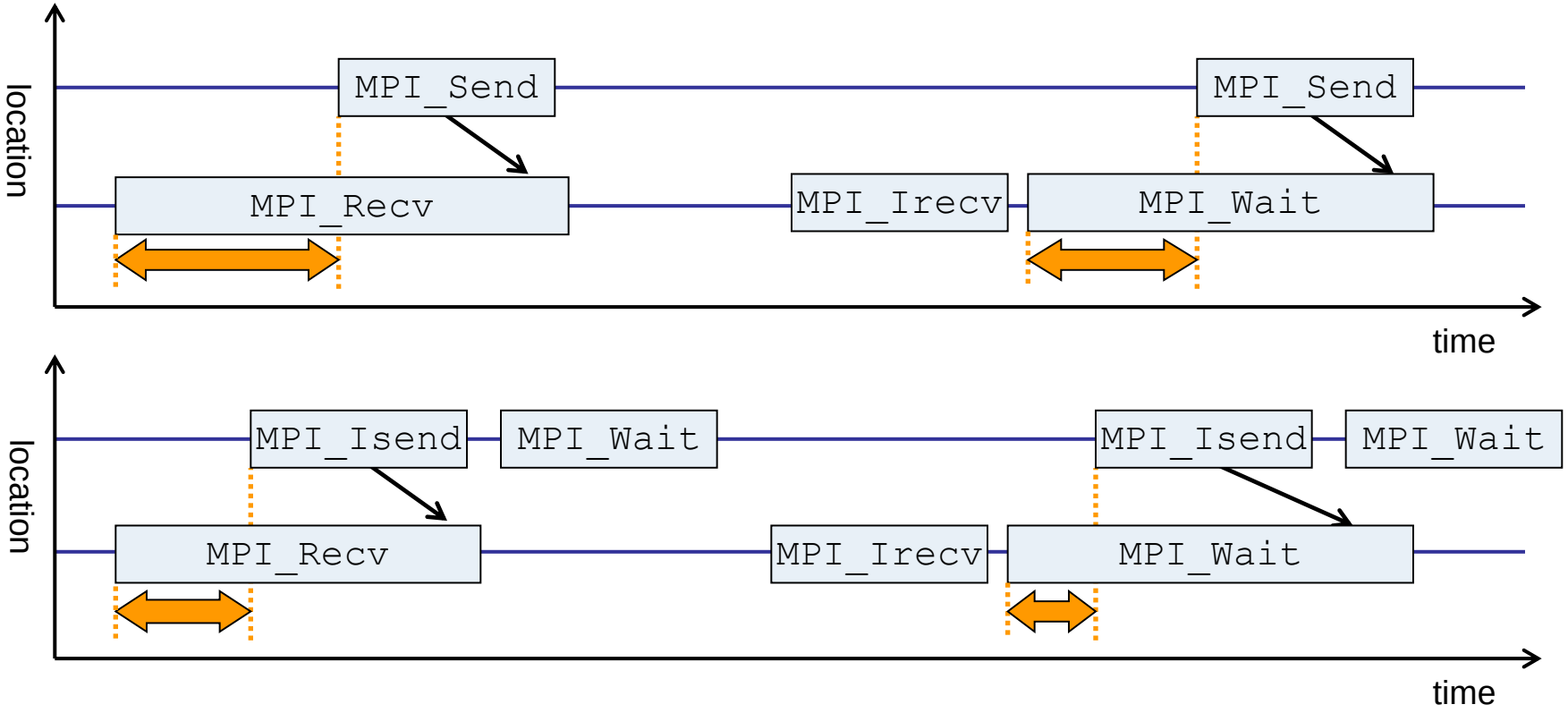
- Open source (3-clause BSD license)
- Fairly portable
  - IBM Blue Gene, Cray XC series, Fujitsu FX & K computer, IBM SP & blade clusters, SGI Altix, Linux clusters, Xeon Phi
- **Unique:** Scalable automatic trace analysis
  - Automatic wait-state search
  - Identification of delays as root causes of wait states
  - Critical-path analysis
  - Parallel replay exploits memory & processors to deliver scalability
- Current release: 2.2.2 (June 2015)

http://www.scalasca.org

Scalasca workflow



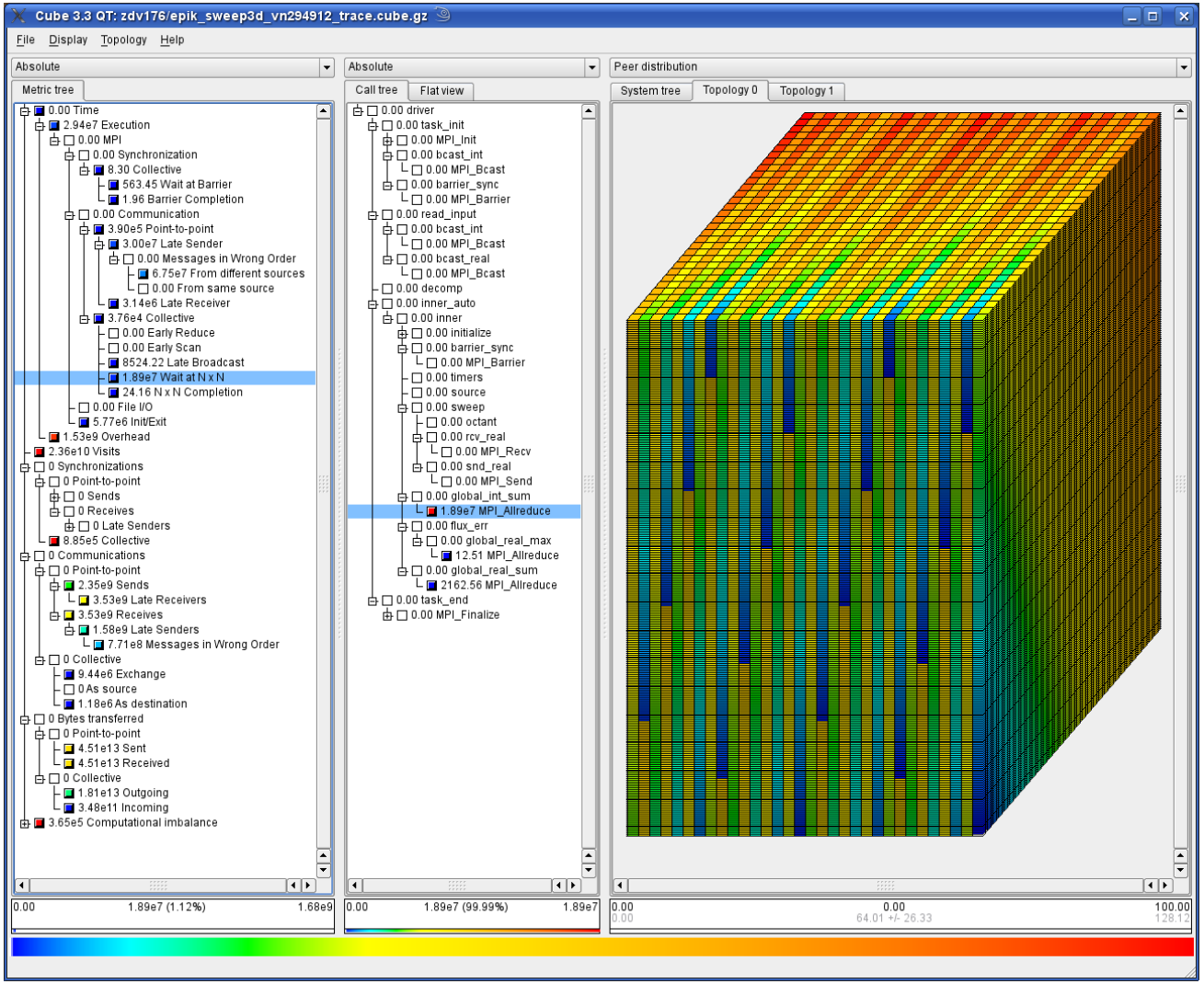
Example: Late Sender



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

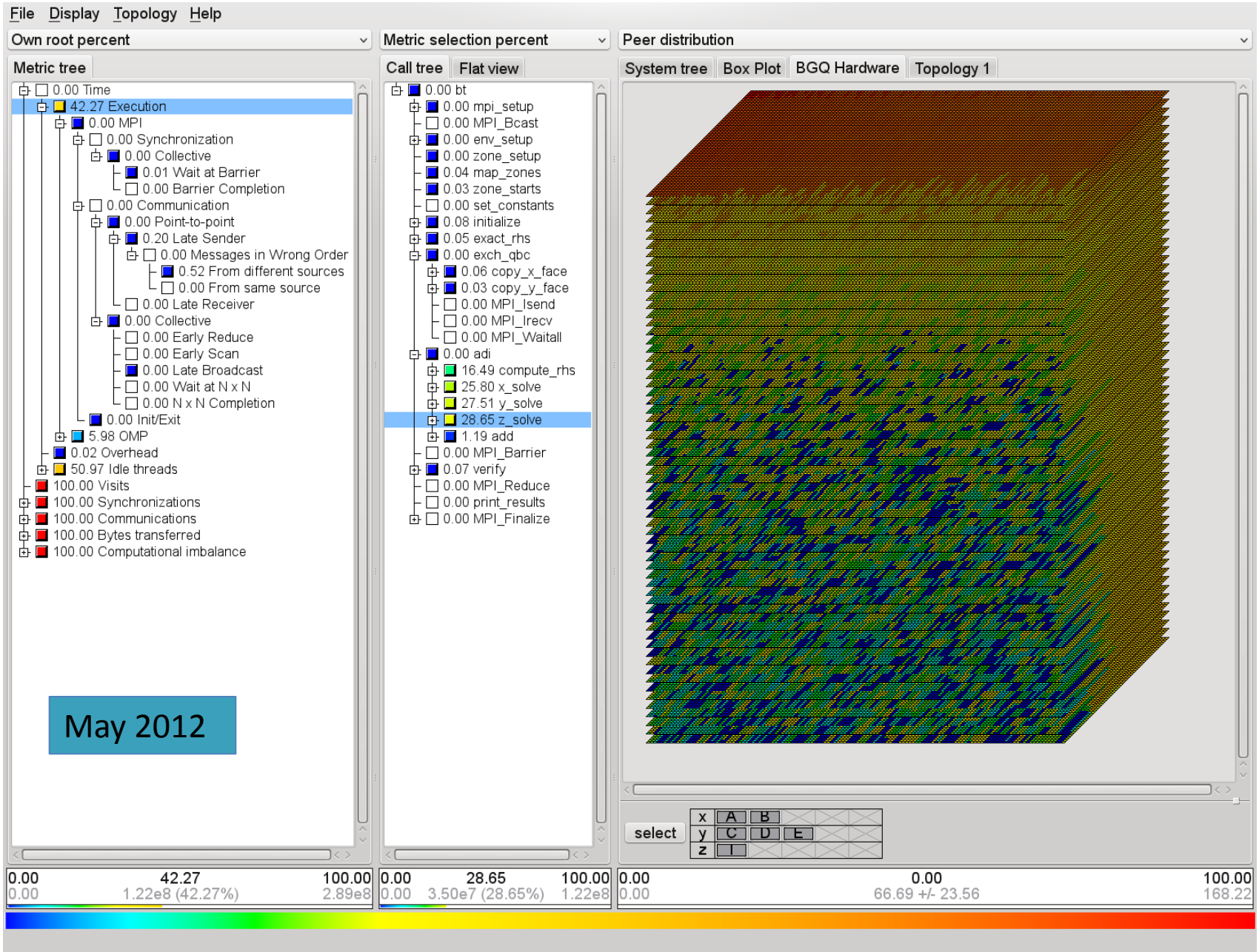
Scalasca trace analysis: Sweep3D@294,912 BG/P

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

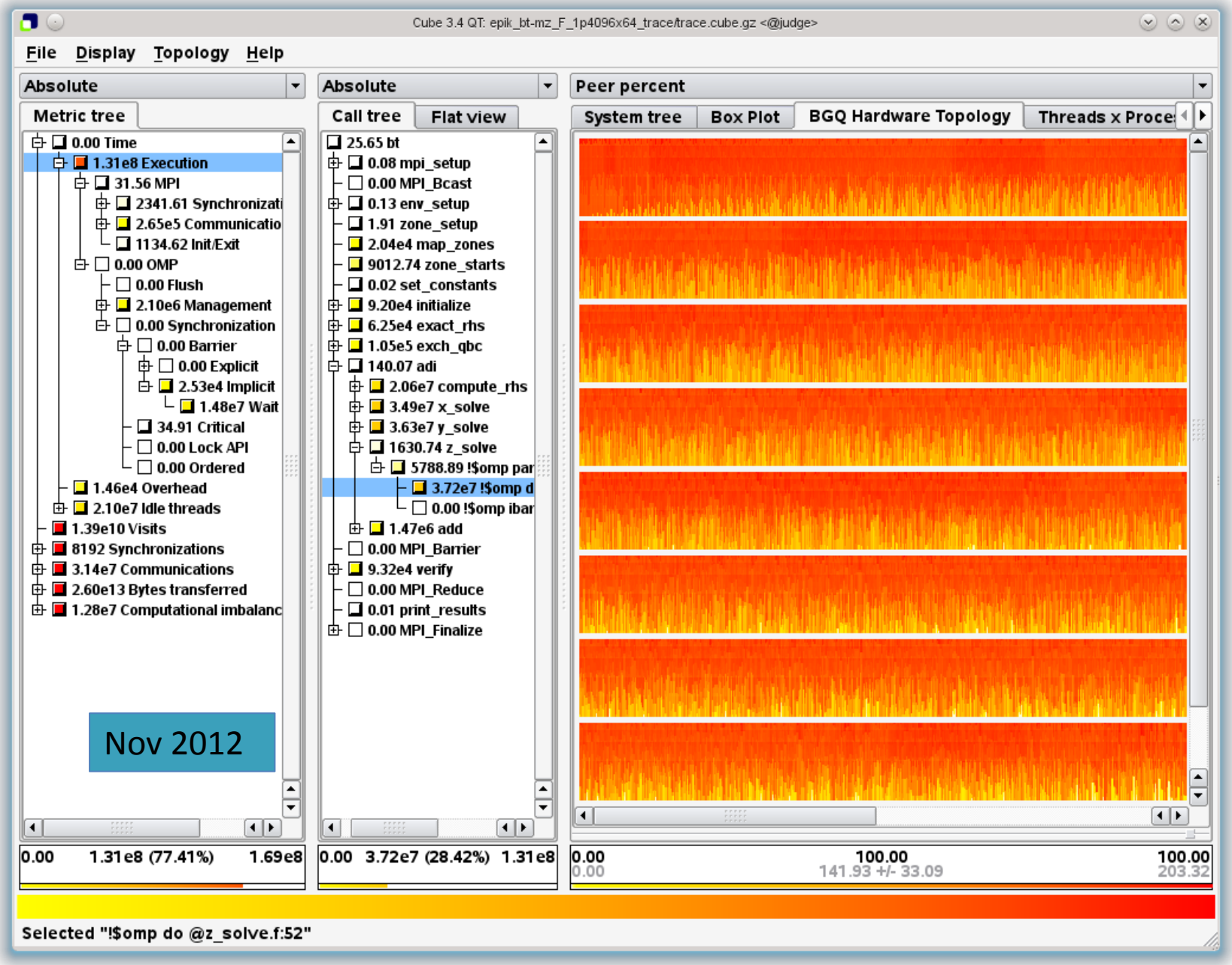


B. J. N. Wylie, M. Geimer, B. Mohr, D. Böhme, Z.Szebenyi, F. Wolf: Large-scale performance analysis of Sweep3D with the Scalasca toolset. Parallel Processing Letters, 20(4):397-414, 2010.

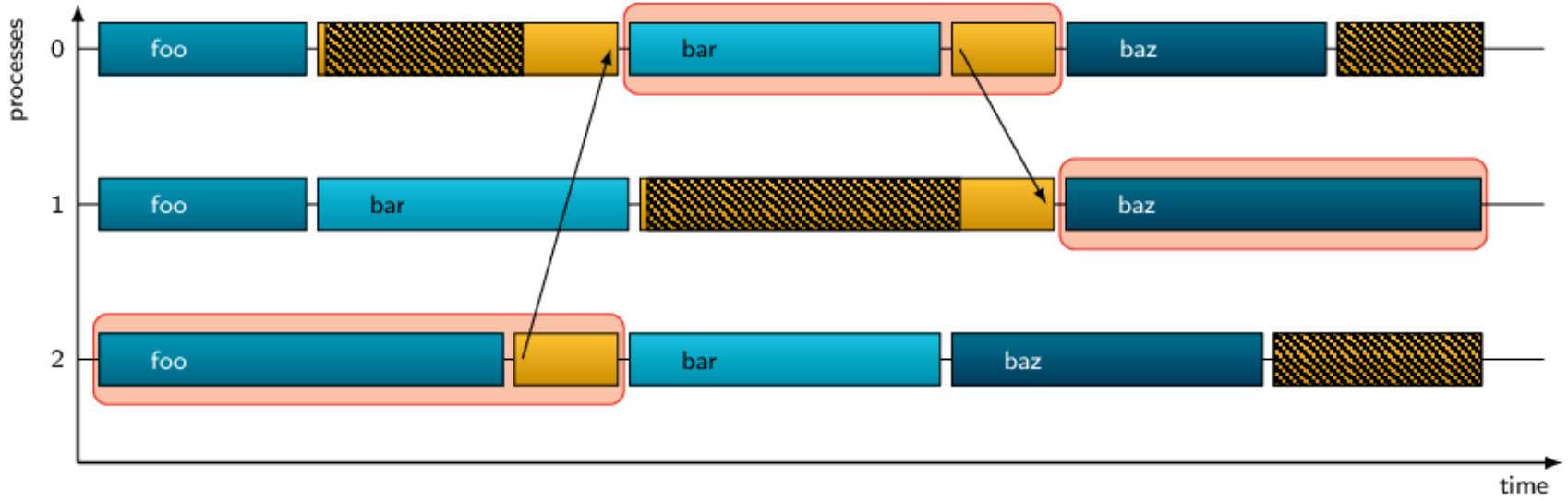
Scalasca trace analysis: bt-mz@524,288 BG/Q



Scalasca trace analysis: bt-mz@1,048,704 BG/Q



Critical-path analysis

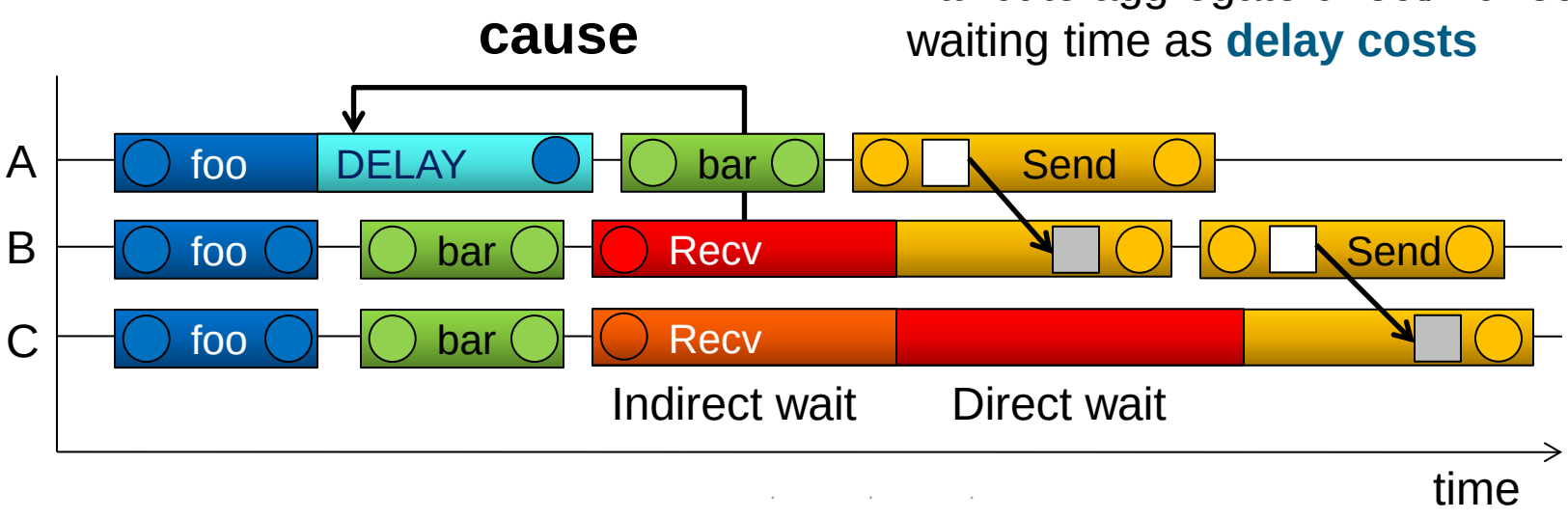


- Follows the causality chain from the last active process/thread back to the program start
  - ➔ **Backward replay** (reversing communication direction)
- Identifies candidate regions for which optimization will prove worthwhile

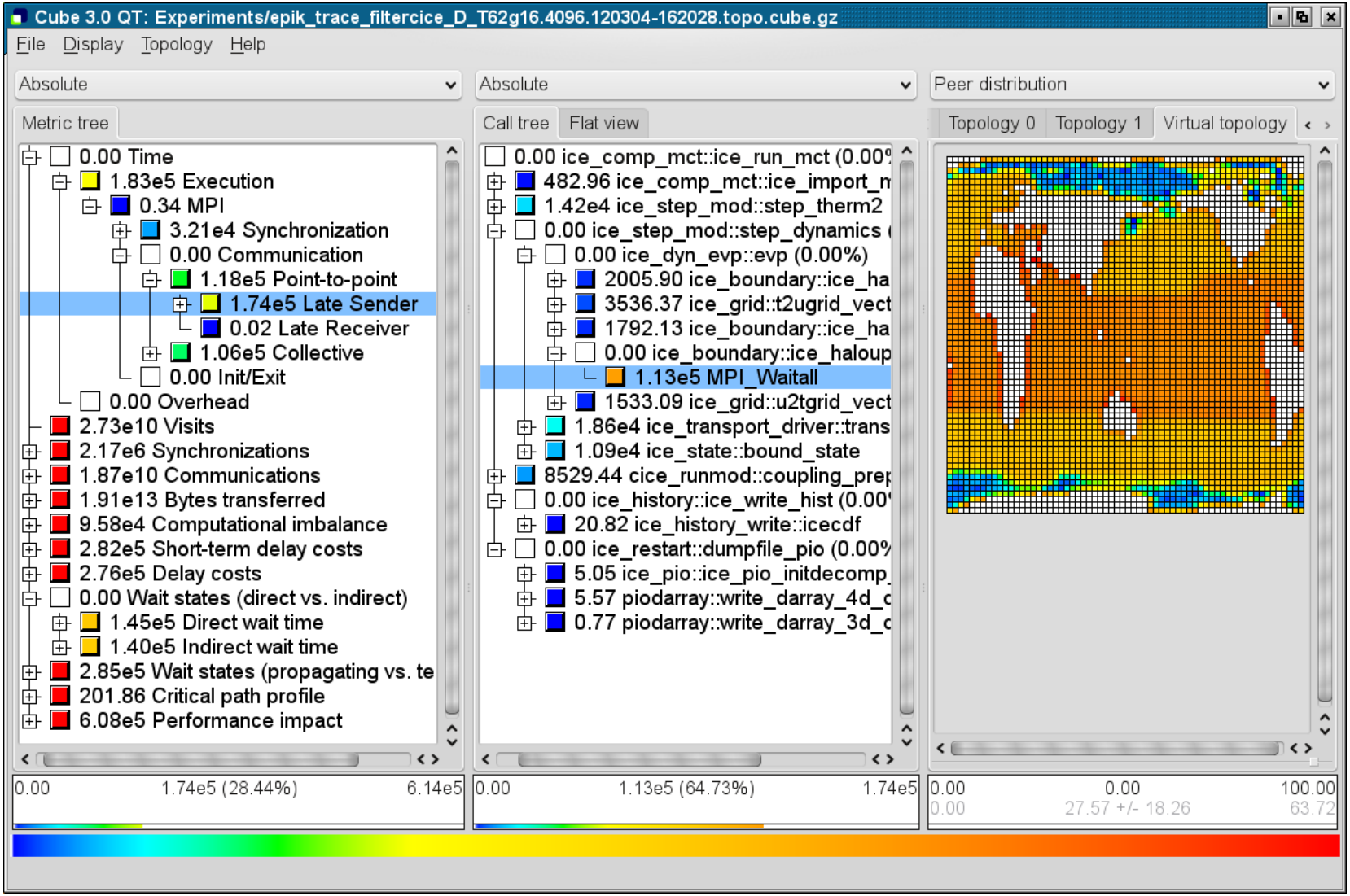
Delay / root-cause analysis

- Wait states typically caused by load or communication imbalances earlier in the program
- Waiting time can also propagate (e.g., indirect waiting time)
- Goal: Find the root cause of wait states

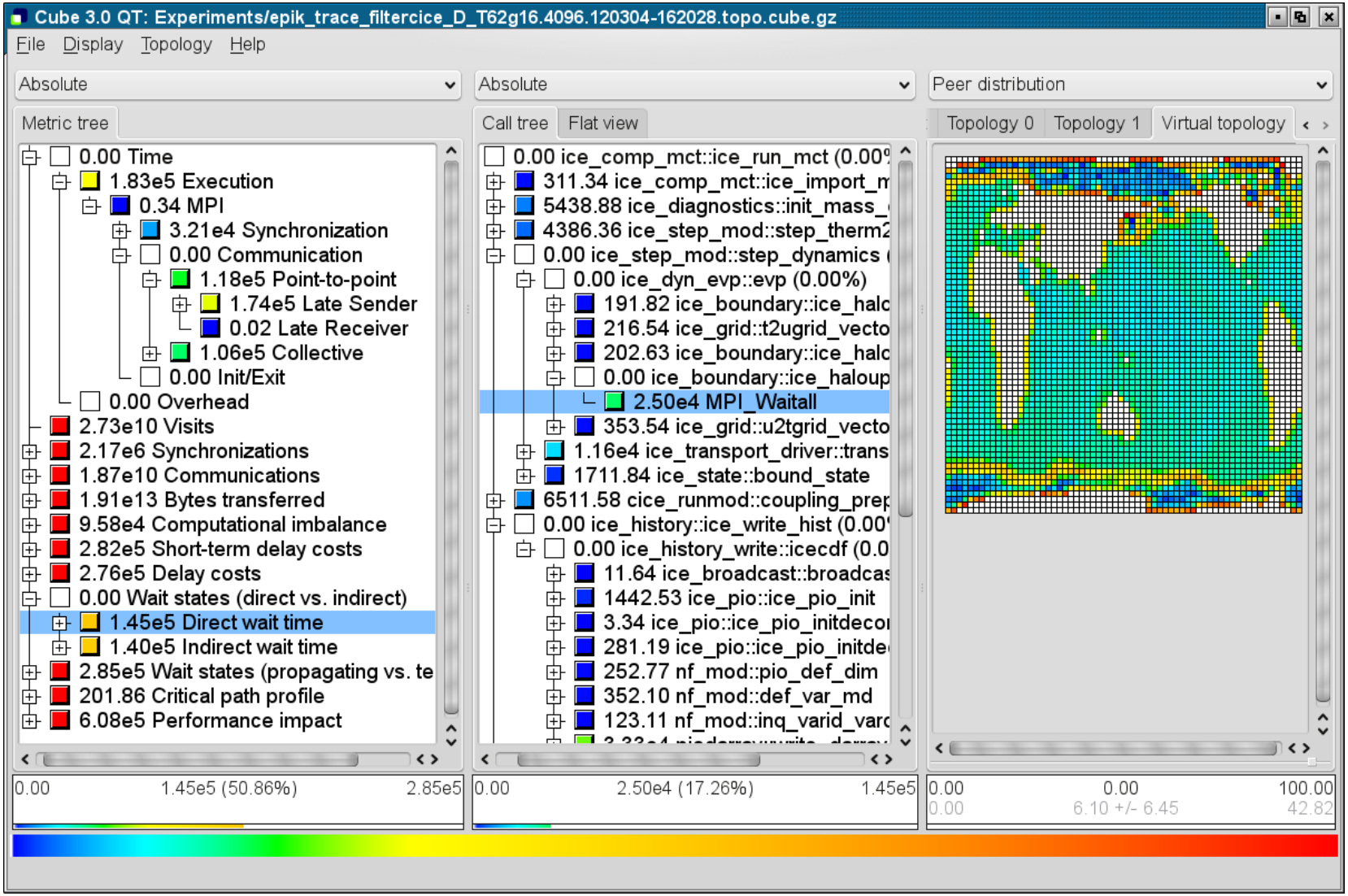
- Distinguish between direct and indirect waiting time
- Identify call path/process combinations delaying other processes and causing first order waiting time
- Identify original **delay**
- Attribute aggregate direct/indirect waiting time as **delay costs**



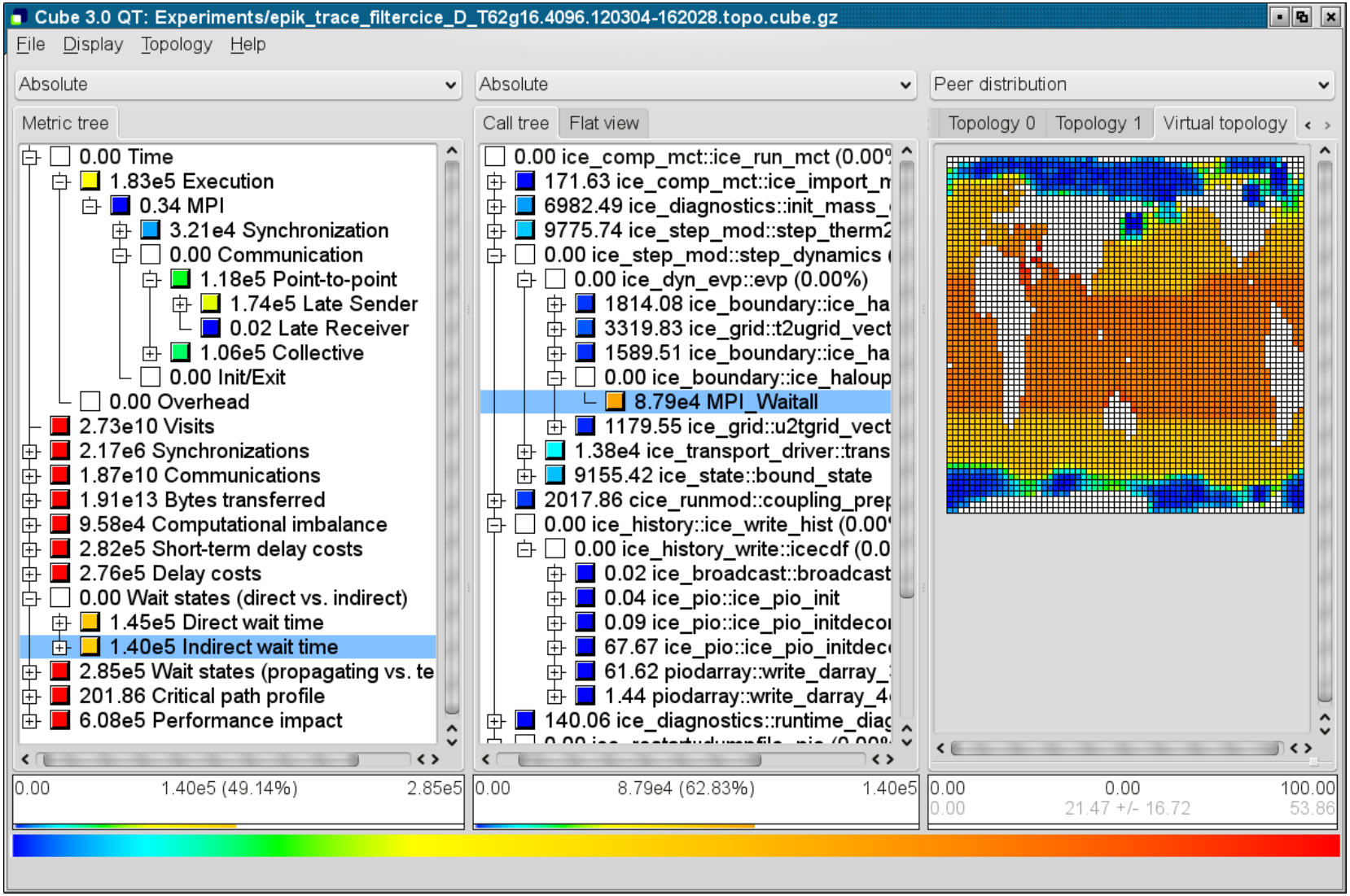
CESM Sea Ice Module: Late Sender analysis



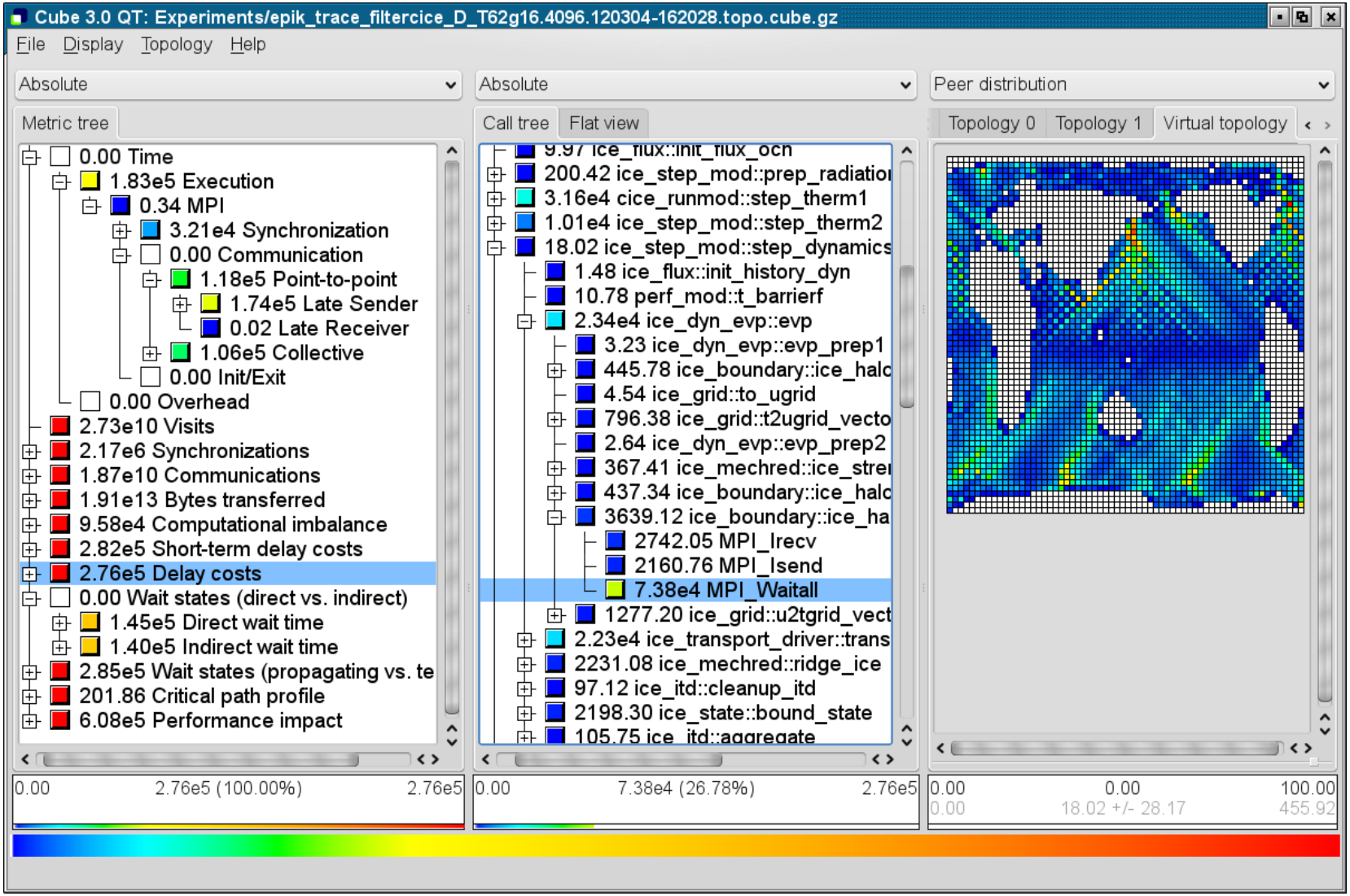
CESM Sea Ice Module: Direct wait time



CESM Sea Ice Module: Indirect wait time

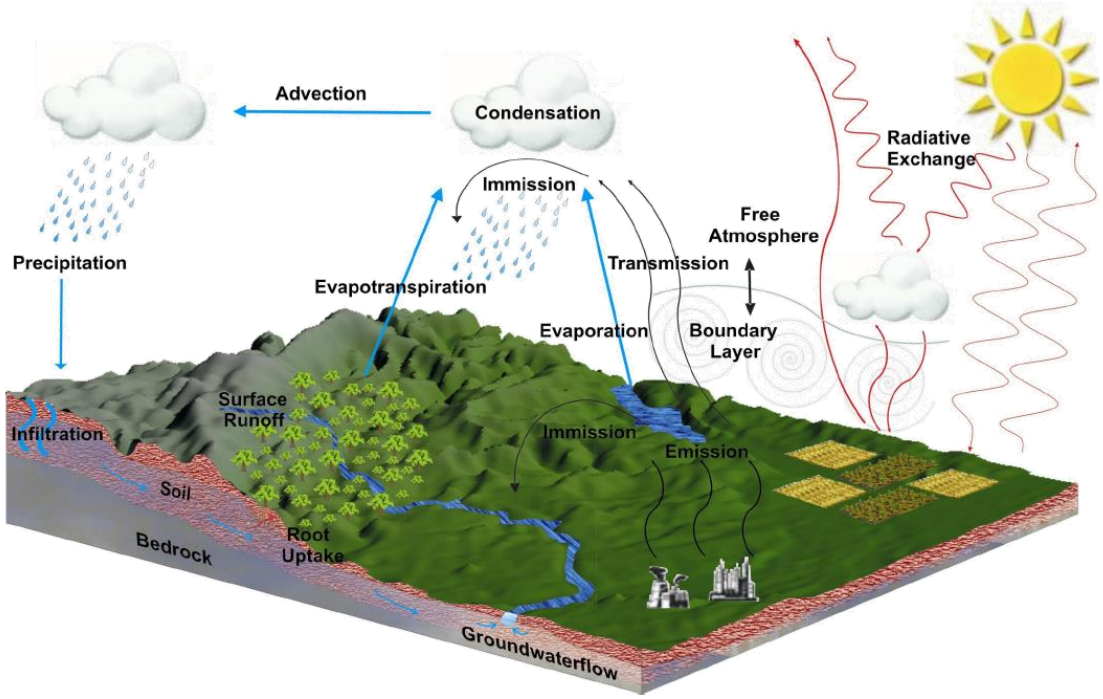


CESM Sea Ice Module: Delay costs



Success story: TerrSysMP

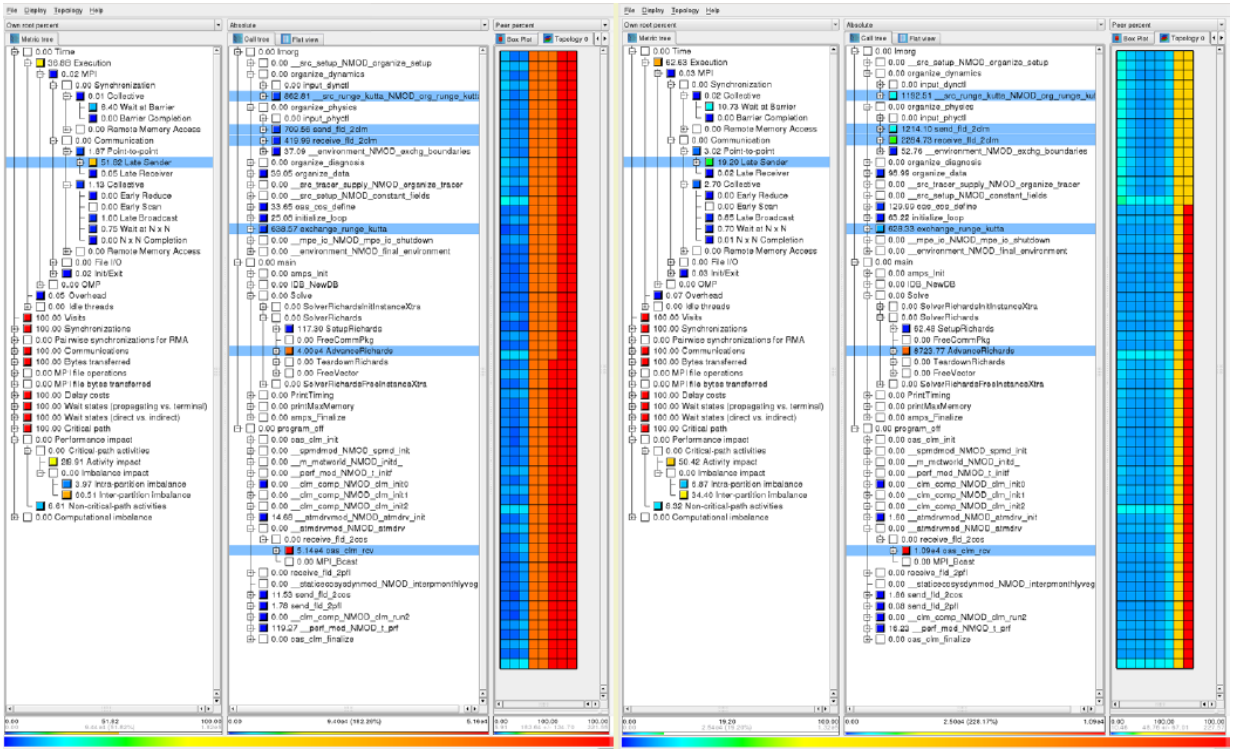
- Scale-consistent
  highly modular
  integrated
  multi-physics
  sub-surface/surface
  hydrology-vegetation
  atmosphere
  modelling system



- Fully-coupled MPMD simulation consisting of
  - COSMO (Weather prediction)
  - CLM (Community Land Model)
  - ParFlow (Parallel Watershed Flow)
  - OASIS coupler

Success story: TerrSysMP

- Identified several sub-components bottlenecks
  - Inefficient communication patterns
  - Unnecessary/inefficient code blocks
  - Inefficient data structures
- Performance of sub-components improved by factor of 2!
- Scaling improved from 512 to 32768 cores!
- Critical-path analysis helped to improve MPMD balancing



Summary

- Application optimization is getting more and more difficult
- A variety of performance analysis tools exist to assist
- The **Score-P** community project aims to
  - Lower the entry threshold
  - Unify instrumentation and measurement for multiple higher-level analysis tools
- **Scalasca** provides a scalable in-depth analysis to
  - Identify communication/synchronization wait states and their root causes
  - Identify the critical path and thus optimization candidates

Acknowledgements

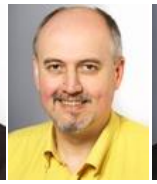
Scalasca team (JSC)


Christian Feld


Markus Geimer


Marc-André Hermanns


Michael Knobloch


Bernd Mohr


Pavel Saviankou


Marc Schlütter


Alexandre Strube


Anke Visser


Brian Wylie


Ilja Zhukov

(TU Darmstadt)


Alexandru Calotoiu


Daniel Lorenz


Aamer Shah


Sergei Shudler


Felix Wolf

Sponsors

HELMHOLTZ ASSOCIATION

ITEA 2
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

SIEMENS

Deutsche Forschungsgemeinschaft DFG

Bundesministerium für Bildung und Forschung

JÜLICH
FORSCHUNGSZENTRUM

Thank you!