JÜLICH
FORSCHUNGSZENTRUM

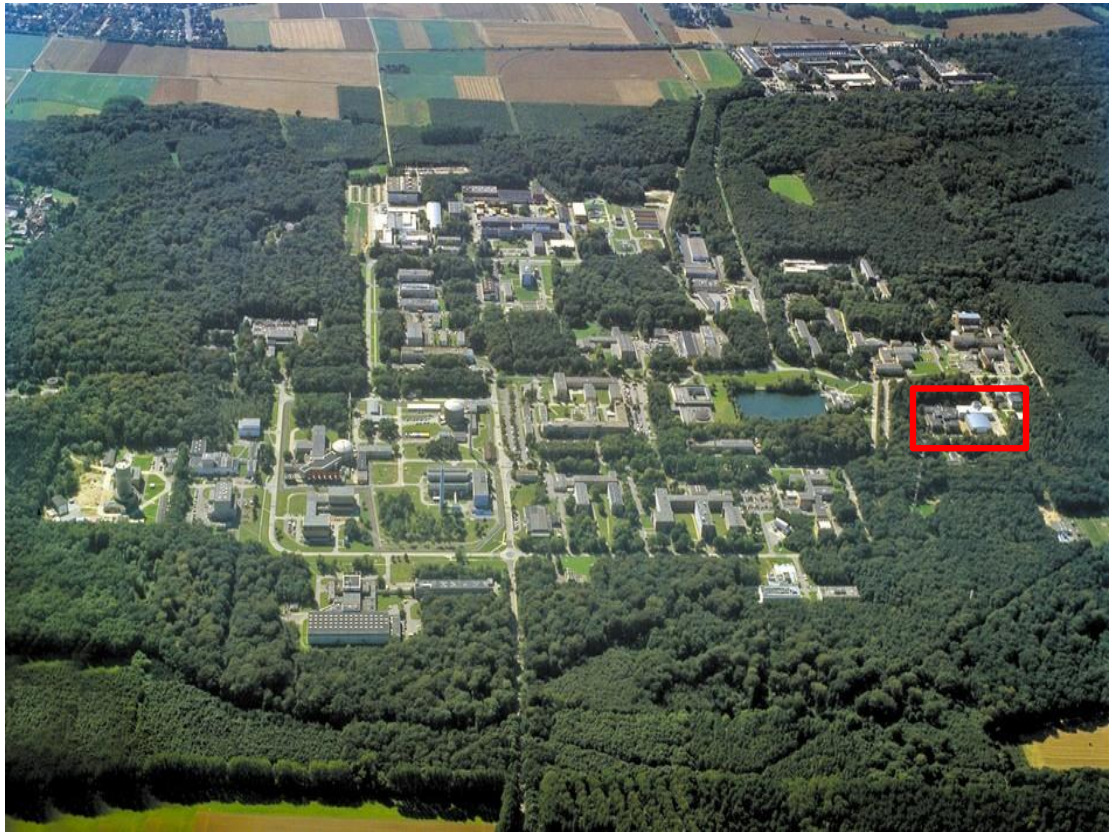
Multicore Performance Analysis at Scale: From two Embedded Cores to one Million HPC Cores

Bernd Mohr
Jülich Supercomputing Centre, Germany

Forschungszentrum Jülich

JÜLICH SUPERCOMPUTING CENTRE

Forschungszentrum Jülich GmbH



- Germany's largest national laboratory
- About 5600 employees
- Research areas
  - Information technology
  - Health (Neuroscience / brain research)
  - Energy and environment

Jülich Supercomputing Centre (JSC)

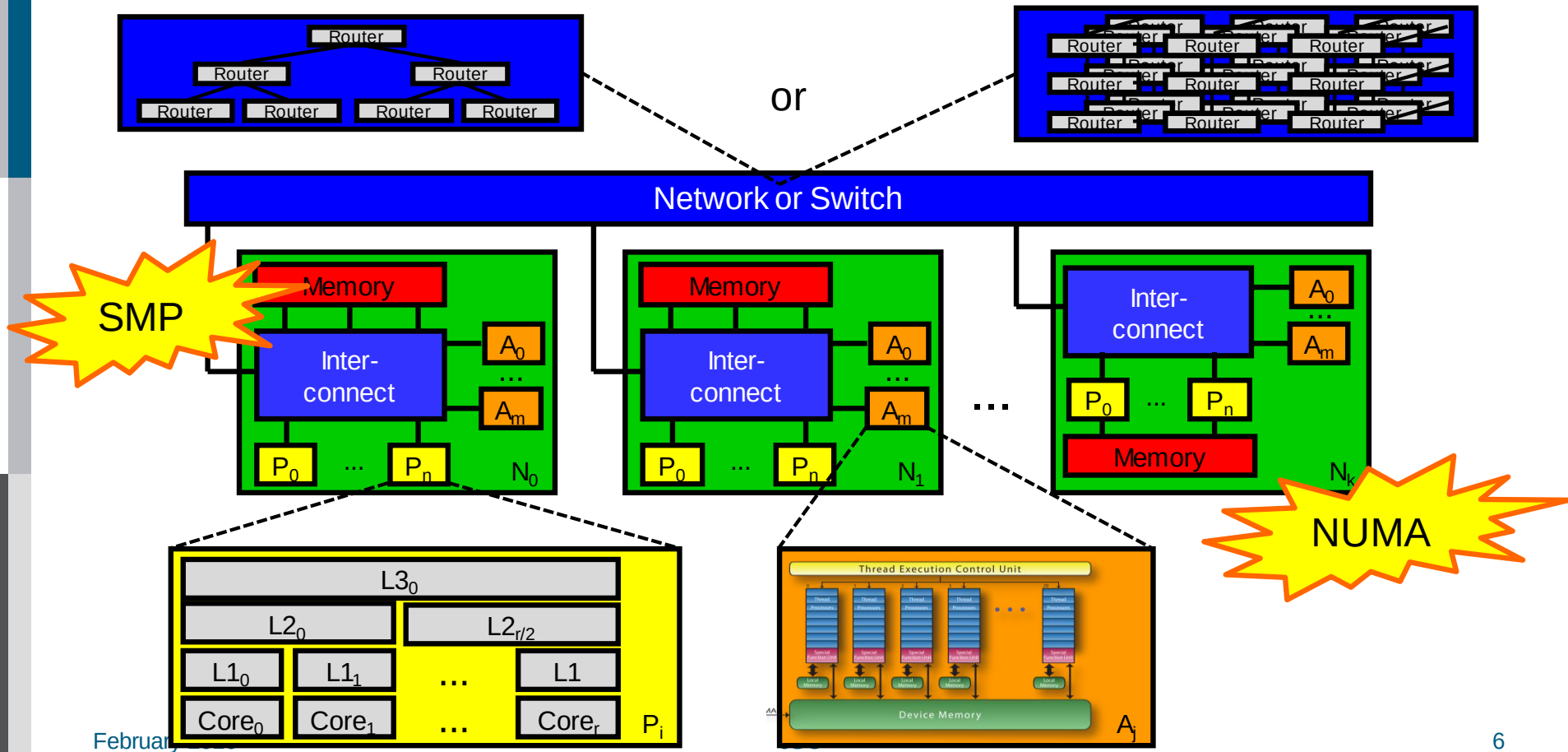
HPC Centre for

- Forschungszentrum Jülich
- Jülich Aachen Research Alliance (JARA)
- Germany as GCS (1 of 3 German National Centres)
- Europe (1st European Centre inside PRACE)

How and Why

MULTICORE PERFORMANCE ANALYSIS

HPC Architectures: State of the Art



Performance Challenges for HPC Systems

JÜLICH FORSCHUNGSZENTRUM

- HPC systems consist of
  - **Complex** configurations
  - With a **huge** number of components
    - Very likely **heterogeneous**
  - With never enough memory
  - **Dynamically changing** configuration due to fault recovery + power saving

⇨ Deep software **hierarchies of large, complex software** components are needed to make use of such systems

⇨ Sophisticated integrated performance measurement, analysis, and optimization capabilities are required to efficiently operate an HPC system

Performance Challenges for Multicore Systems

JÜLICH FORSCHUNGSZENTRUM

- Multicore Systems consist of
  - **Complex** configurations
  - With a ~~**huge**~~ number of components
    - Very likely **heterogeneous**
  - With never enough memory
  - **Dynamically changing** configuration due to ~~fault recovery~~ + power saving

⇨ Deep software **hierarchies of large, complex** software components are needed to make use of such systems

⇨ Sophisticated integrated performance measurement, analysis, and optimization capabilities are required to efficiently operate an Multicore Systems

Measurement Methods: Profiling

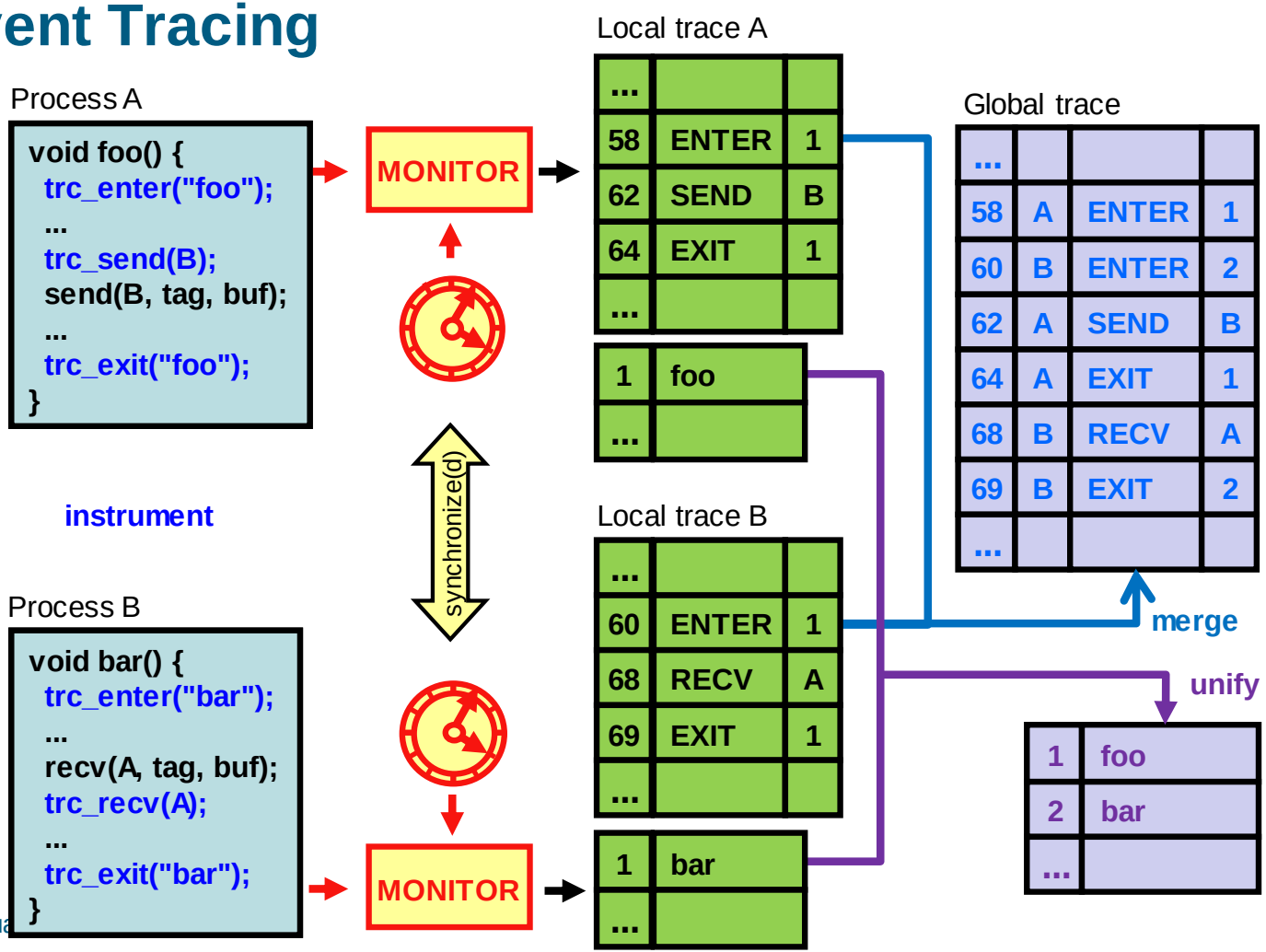
- Recording of **aggregated information**
  - Time
  - Counts
    - Calls
    - Hardware counters
- **about program and system entities**
  - Functions, call sites, loops, basic blocks, …
  - Processes, threads
- **Statistical information**
  - Min, max, mean and total number of values

Measurement Methods: Tracing

- Recording **information about** significant points (**events**) during execution of the program
  - Enter/leave a code region (function, loop, …)
  - Send/receive a message ...
- Save information in **event record**
  - Timestamp, location ID, event type
  - plus event specific information
- **Event trace** := stream of event records sorted by time
- Can be used to reconstruct the **dynamic behavior**

  ⇨ Abstract execution model on level of defined events

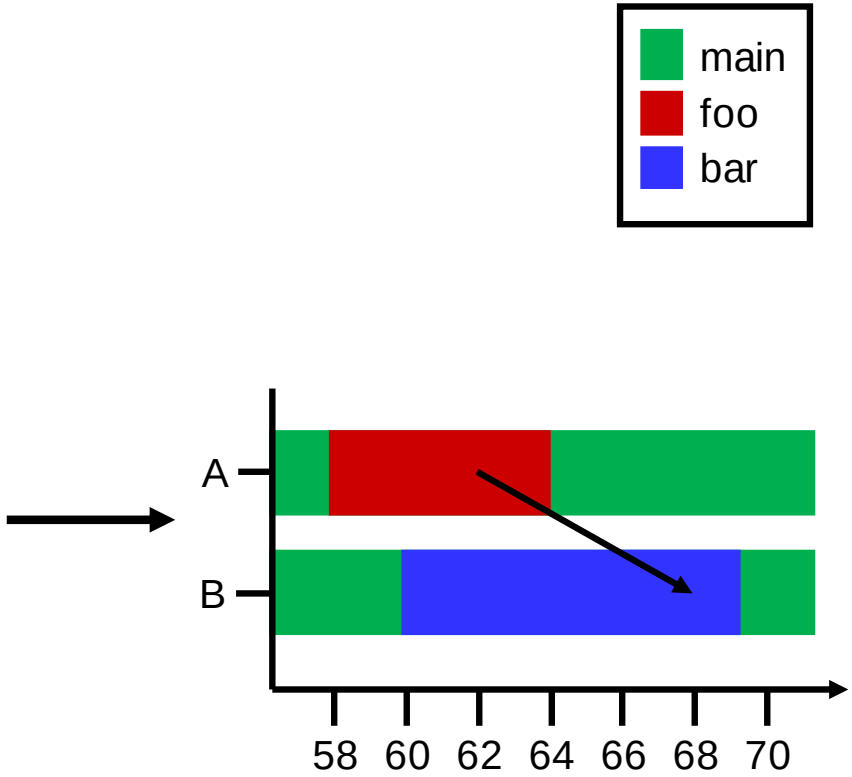
Event Tracing



Event Tracing: “Timeline” Visualization

1	foo
2	bar
3	…

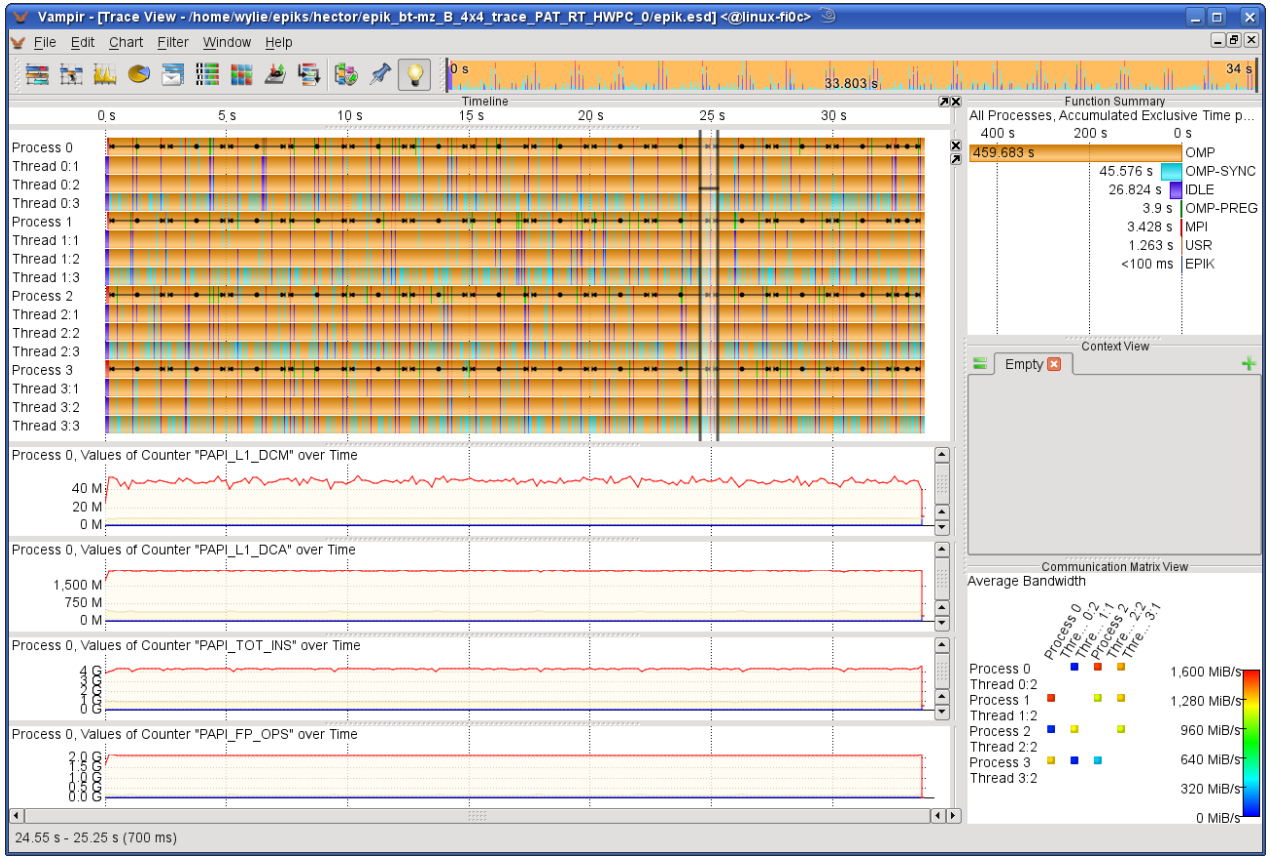
…			
58	A	ENTER	1
60	B	ENTER	2
62	A	SEND	B
64	A	EXIT	1
68	B	RECV	A
69	B	EXIT	2
…			



More than numbers and diagrams

INSIGHTFULNESS

Interactive Event Trace Analysis: Vampir

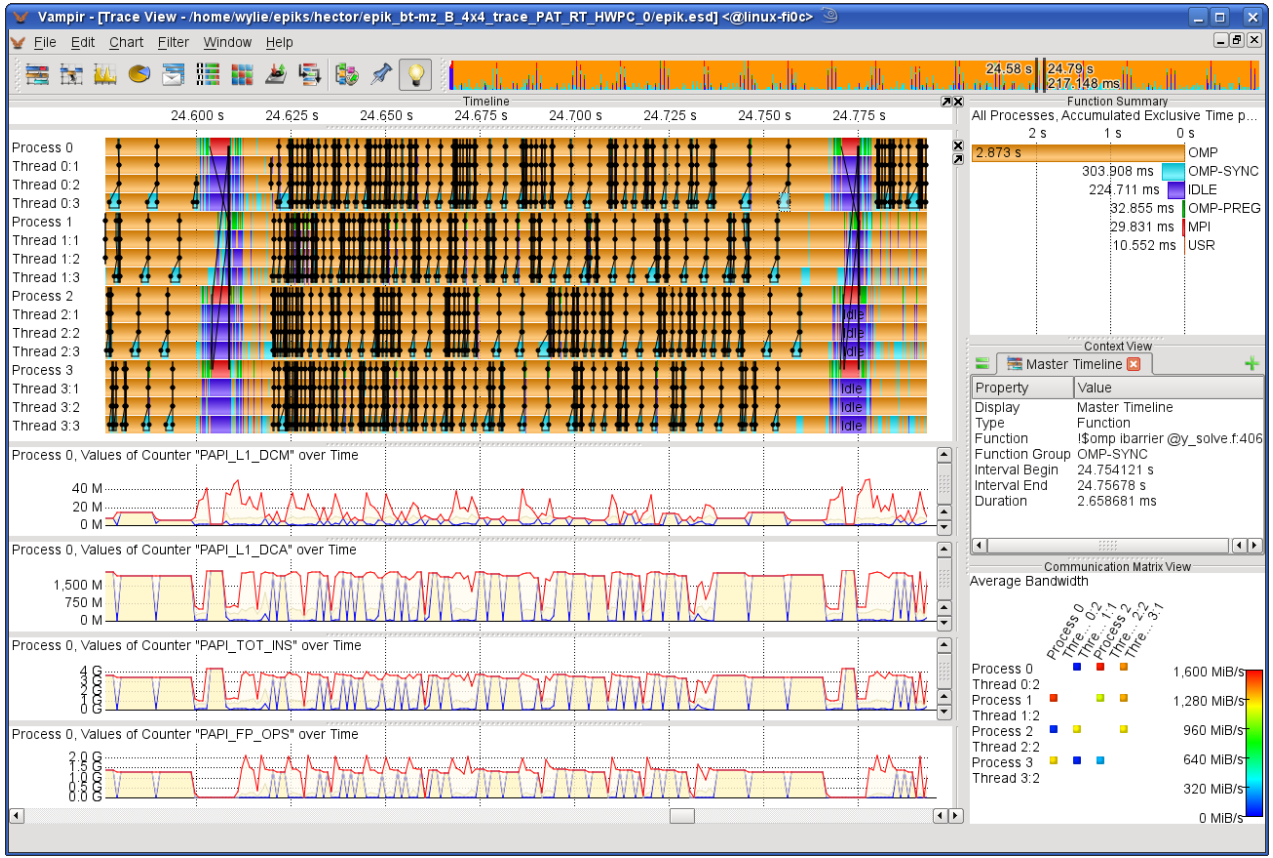


Visual presentation of dynamic runtime behaviour

- Event timeline chart for states & interactions of processes/threads
- Communication statistics, summaries & more

http://www.vampir.eu/

Vampir GUI (zoom)



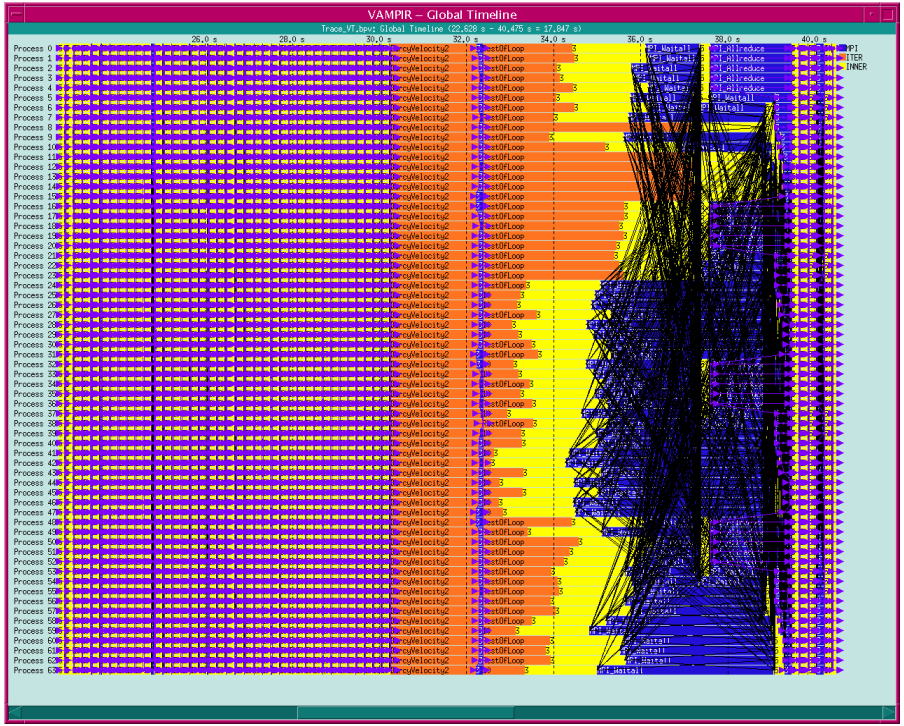
Interactive browsing, zooming, selecting

- Linked displays & statistics adapt to selected time interval

Trace formats

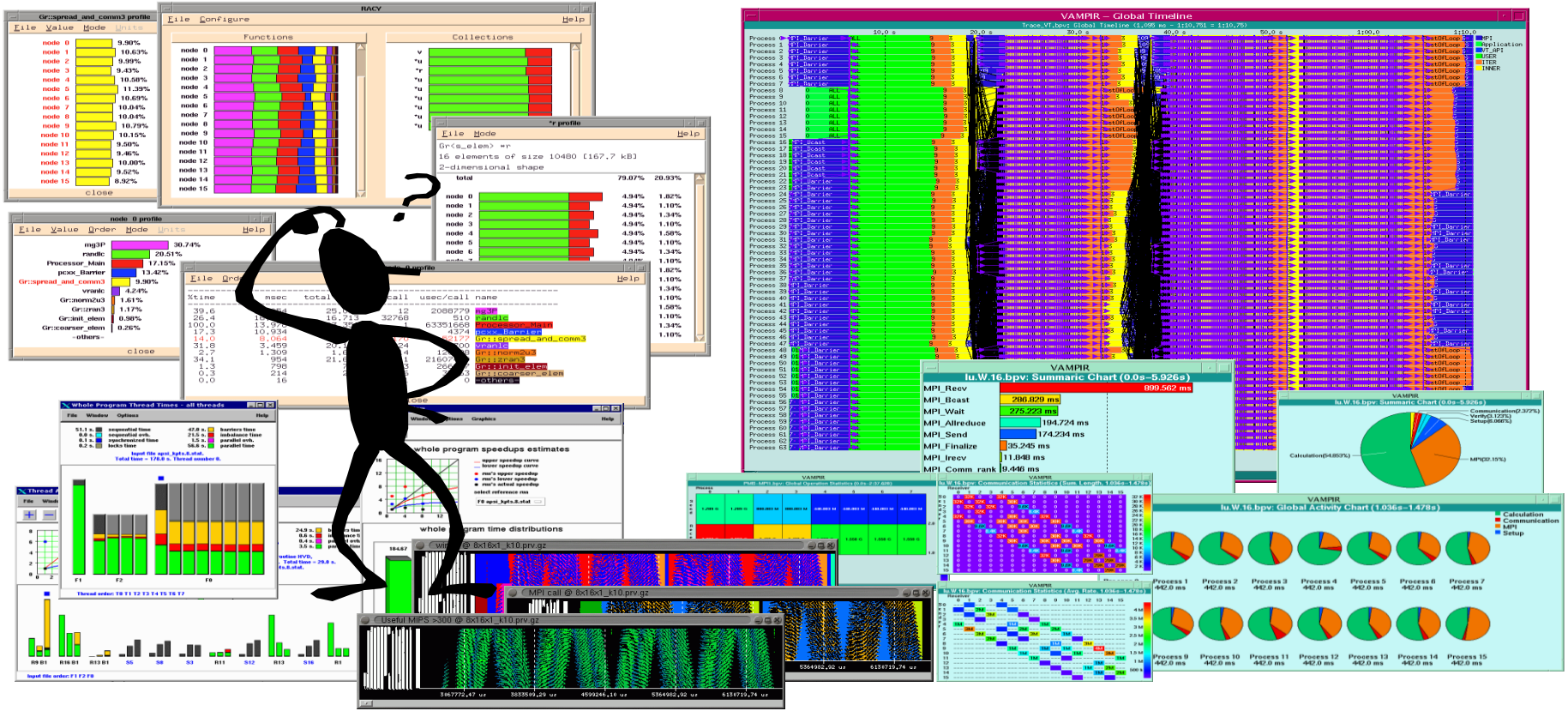
- OTF (VampirTrace)
- OTF2 (Score-P)
- EPIK (Scalasca1)

“A picture is worth 1000 words…”

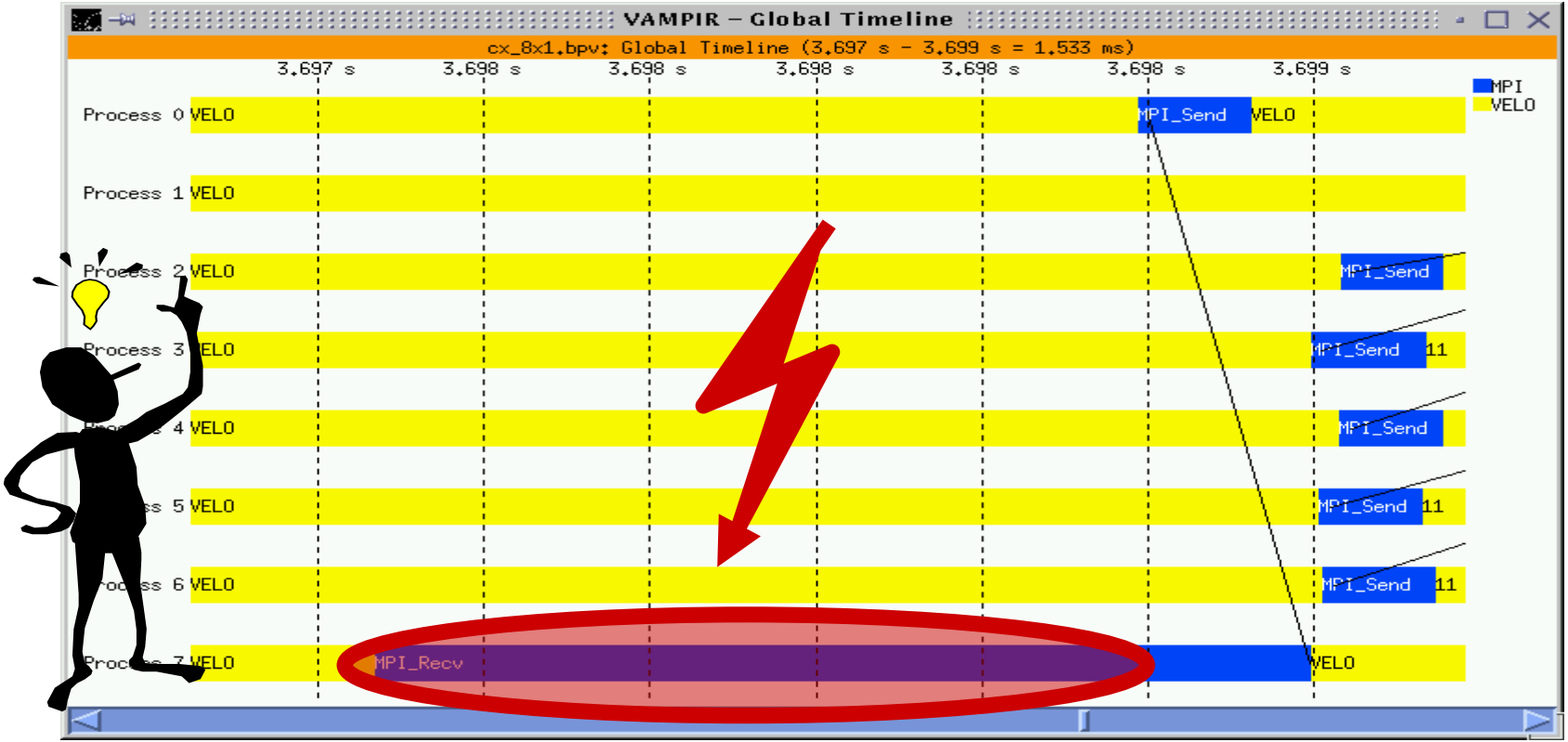


- MPI ring program
- “Real world” example

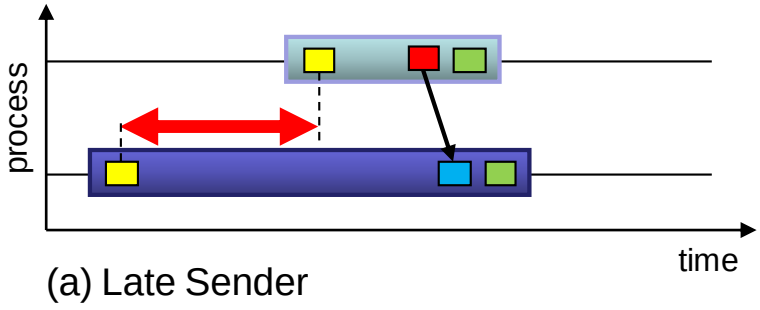
“What about 1000’s of pictures?” (with 100’s of menu options)



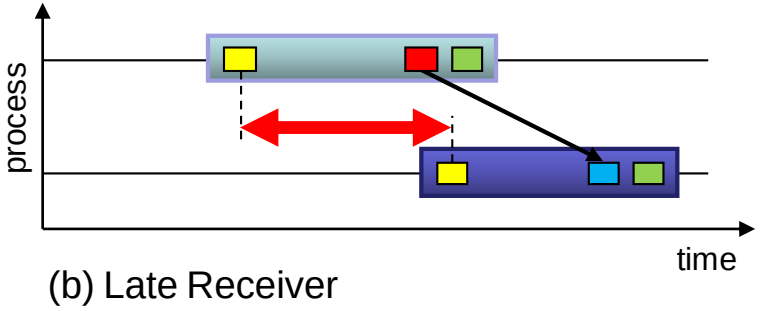
Example Automatic Analysis: Late Sender



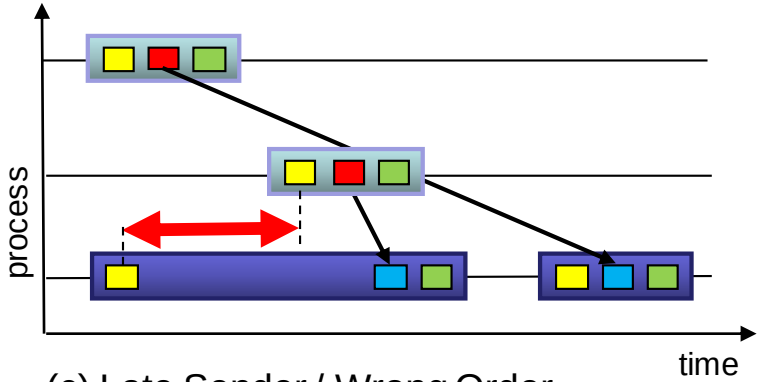
Example MPI Wait States



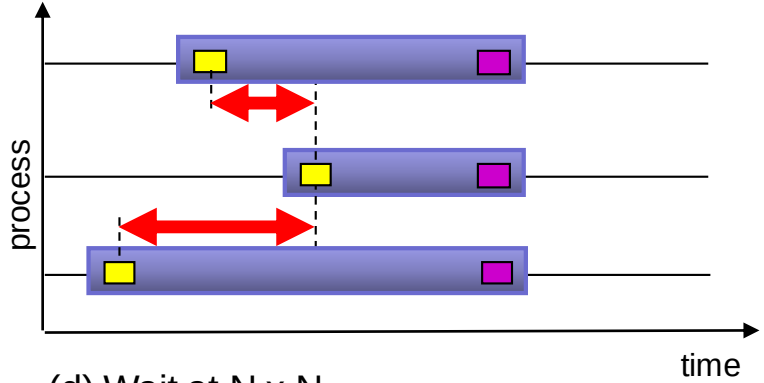
(a) Late Sender



(b) Late Receiver



(c) Late Sender / Wrong Order



(d) Wait at N x N

ENTER EXIT SEND RECV COLLEXIT

Scalasca

JÜLICH FORSCHUNGSZENTRUM

scalasca

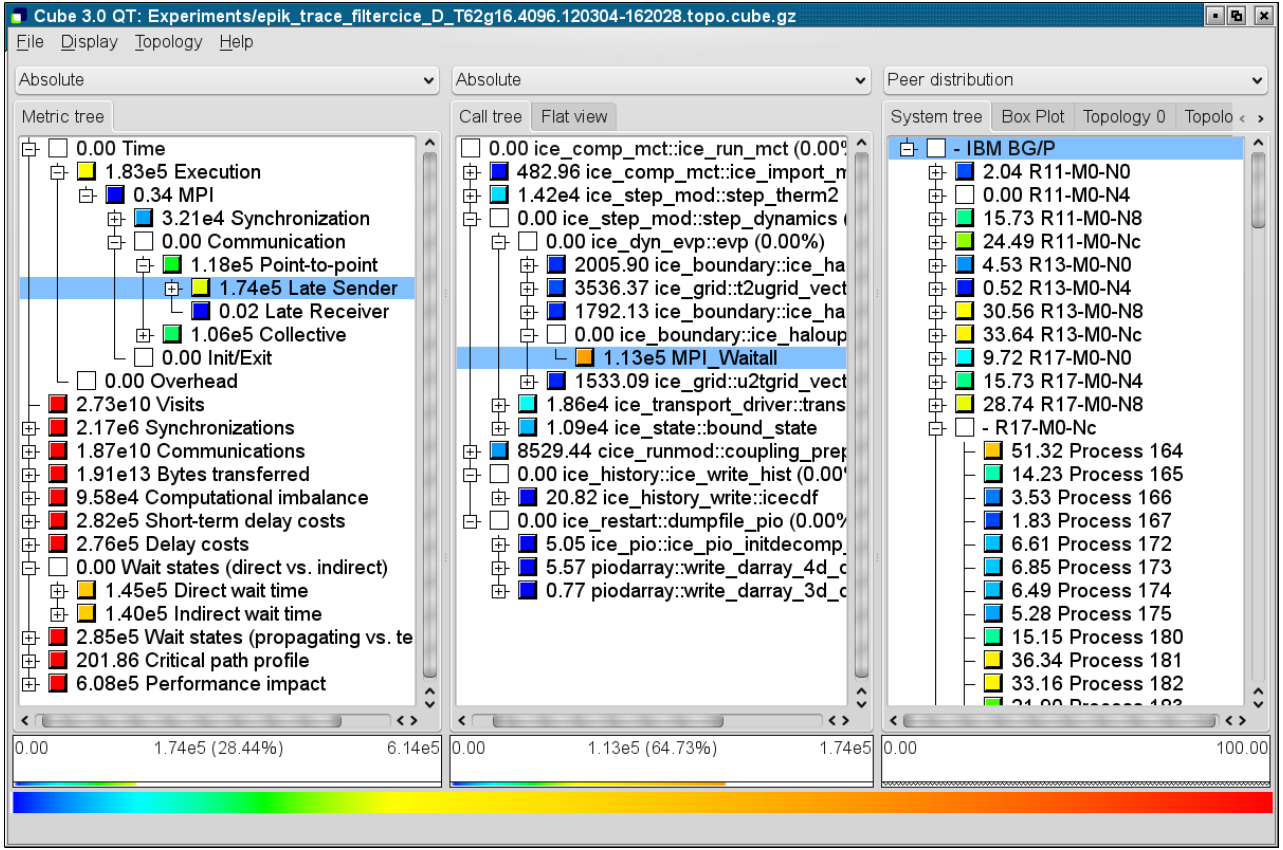
http://www.scalasca.org/

- Scalable Analysis of Large Scale Applications
- Approach
  - **Instrument** C, C++, and Fortran parallel applications
  - Option 1: **scalable call-path profiling**
  - Option 2: **scalable event trace analysis**
    - **Collect** event traces
    - **Process** trace **in parallel**
      - Wait-state analysis
      - Delay and root-cause analysis
      - Critical path analysis
    - **Categorize and rank** results

Scalasca Example: CESM Sea Ice Module

Late Sender Analysis

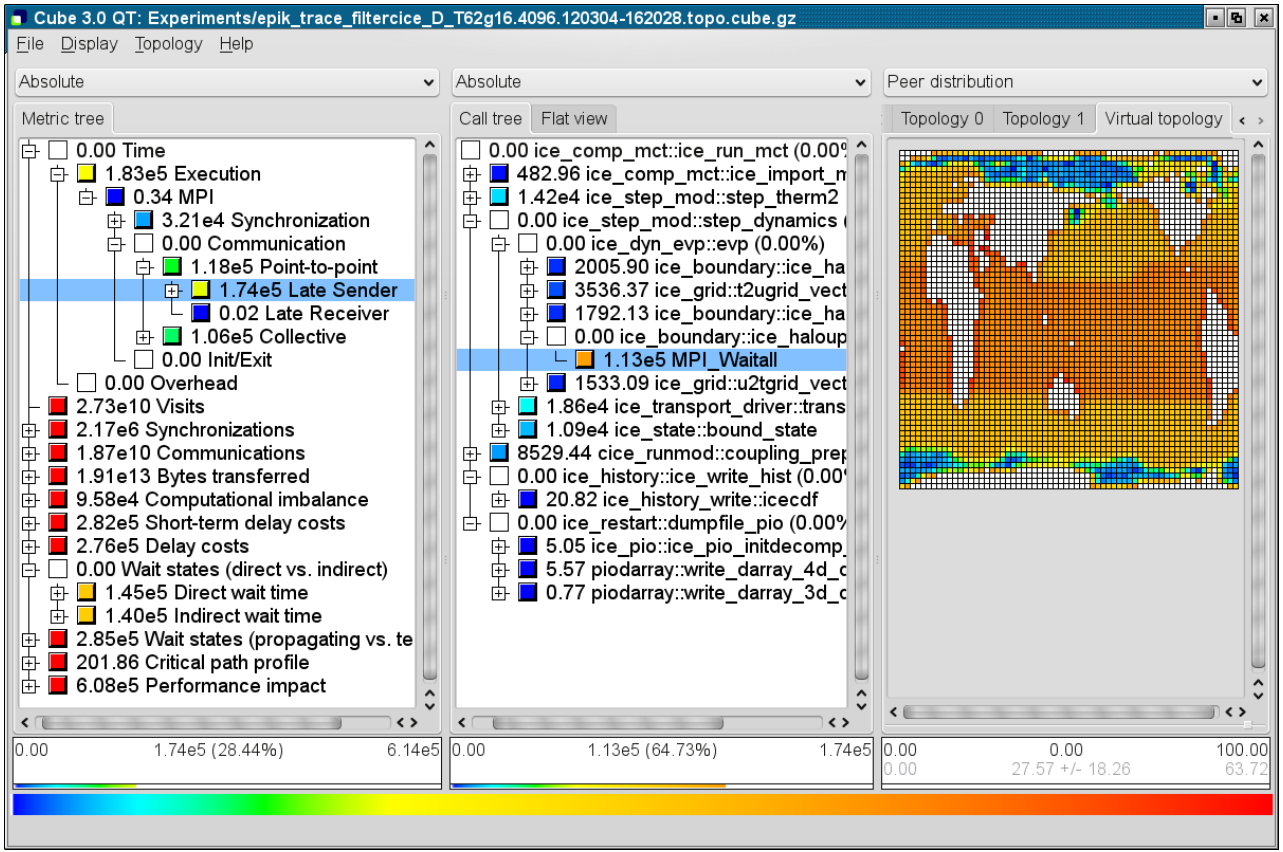
- Finds waiting at MPI_Waitall() inside ice boundary halo update
- Shows distribution of imbalance across system and ranks



Scalasca Example: CESM Sea Ice Module

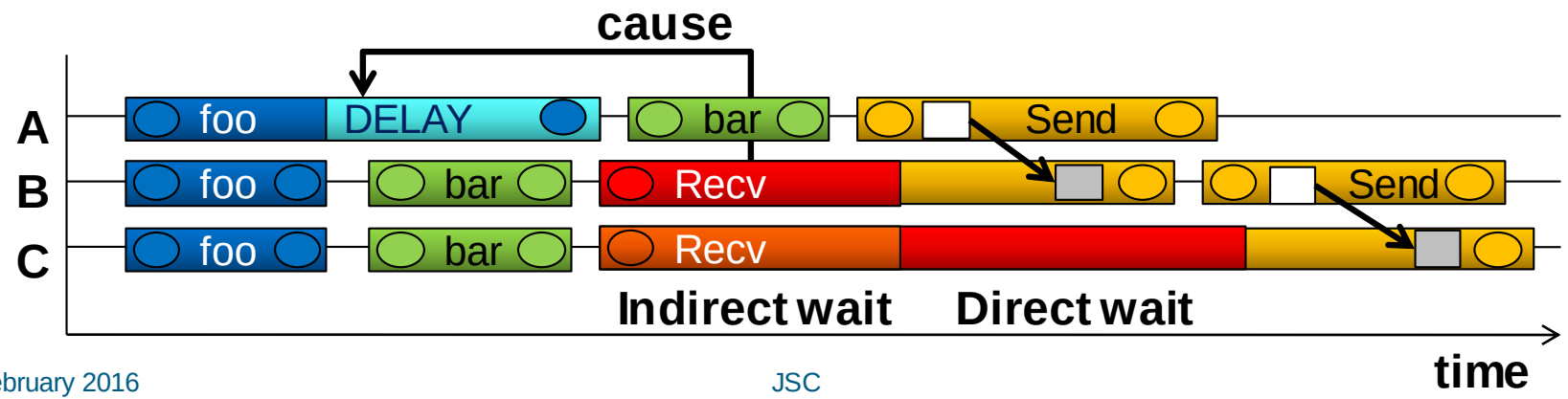
Late Sender Analysis + Application Topology

- Shows distribution of imbalance over topology
- MPI topologies are automatically captured



Scalasca Root Cause Analysis

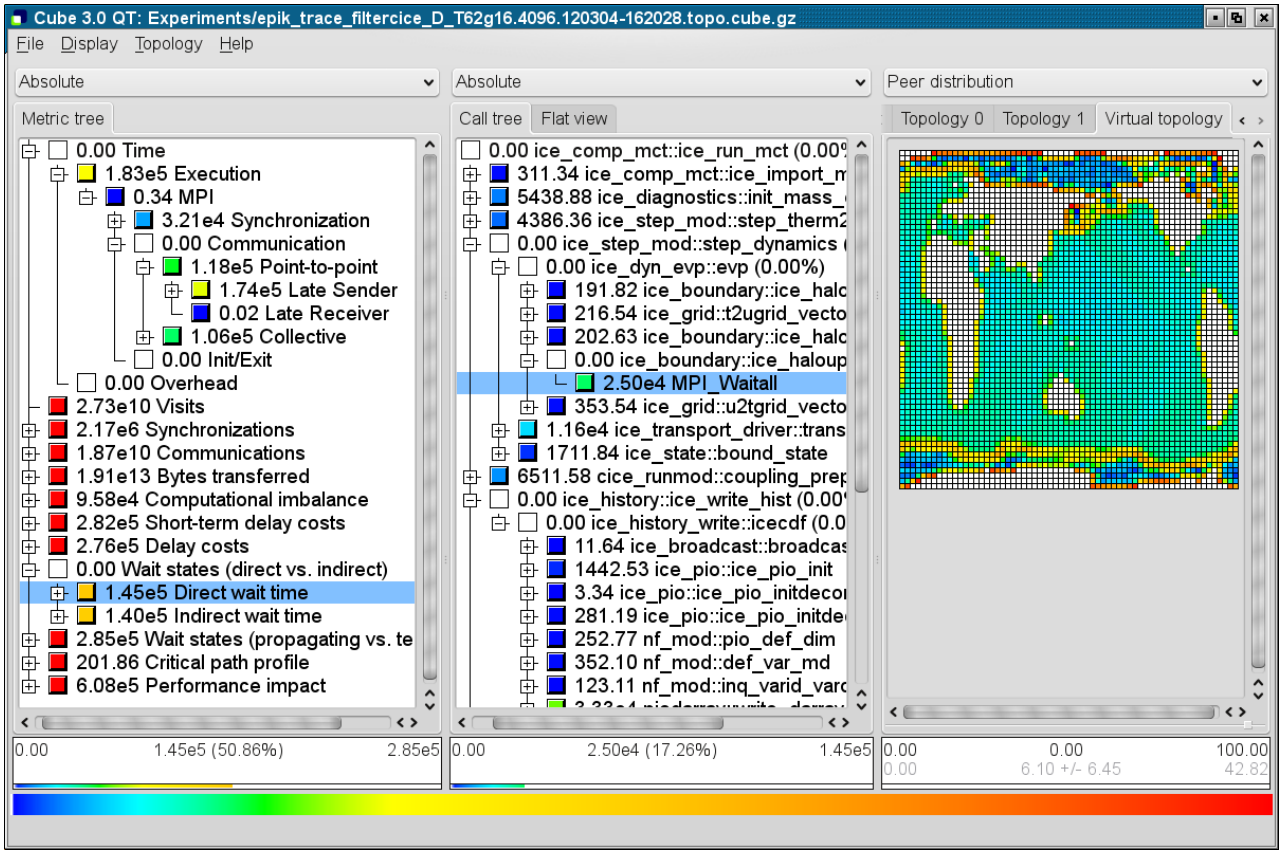
- **Root-cause analysis**
  - Wait states typically caused by load or communication imbalances earlier in the program
  - Waiting time can also propagate (e.g., indirect waiting time)
  - Enhanced performance analysis to find the root cause of wait states
- **Approach**
  - Distinguish between direct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



Scalasca Example: CESM Sea Ice Module

Direct Wait Time Analysis

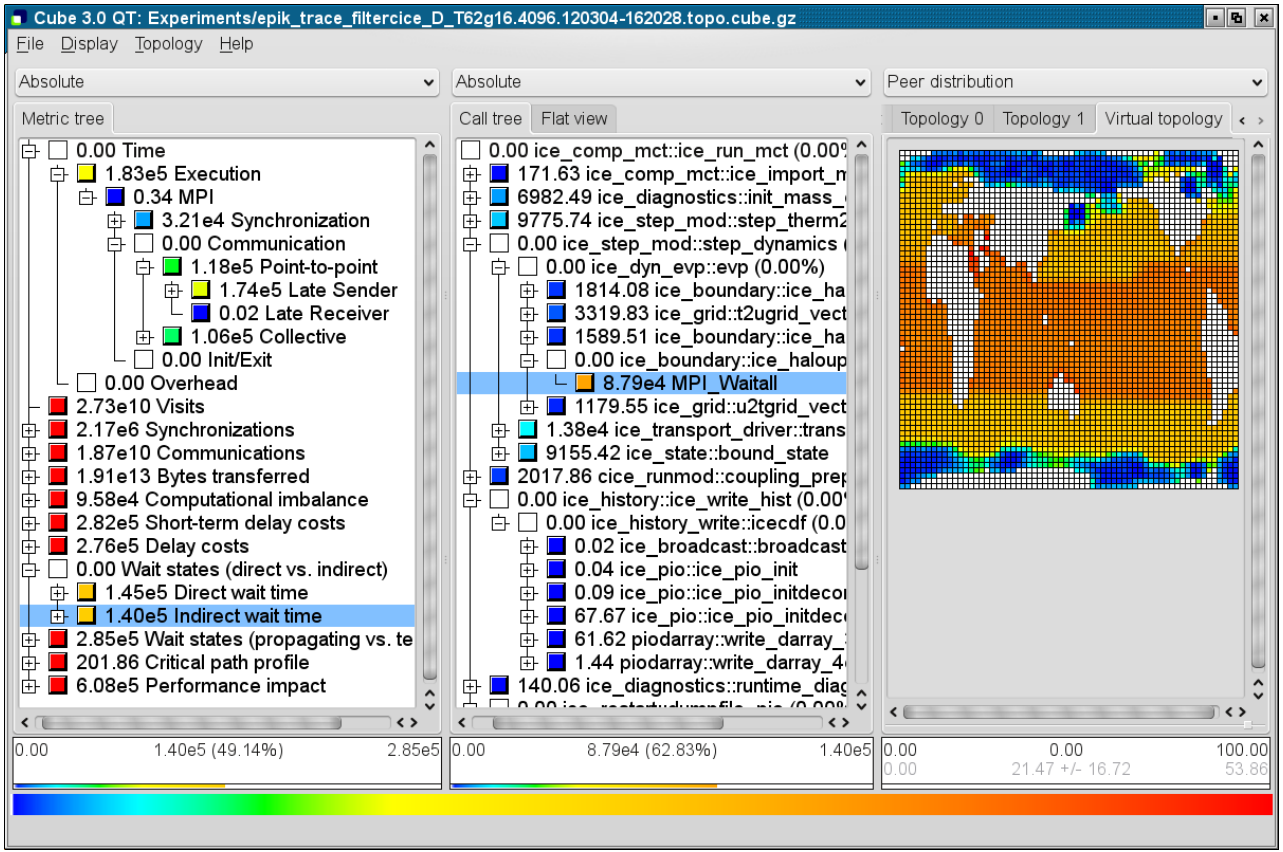
- Direct wait caused by ranks processing areas near the north and south ice borders



Scalasca Example: CESM Sea Ice Module

Indirect Wait Time Analysis

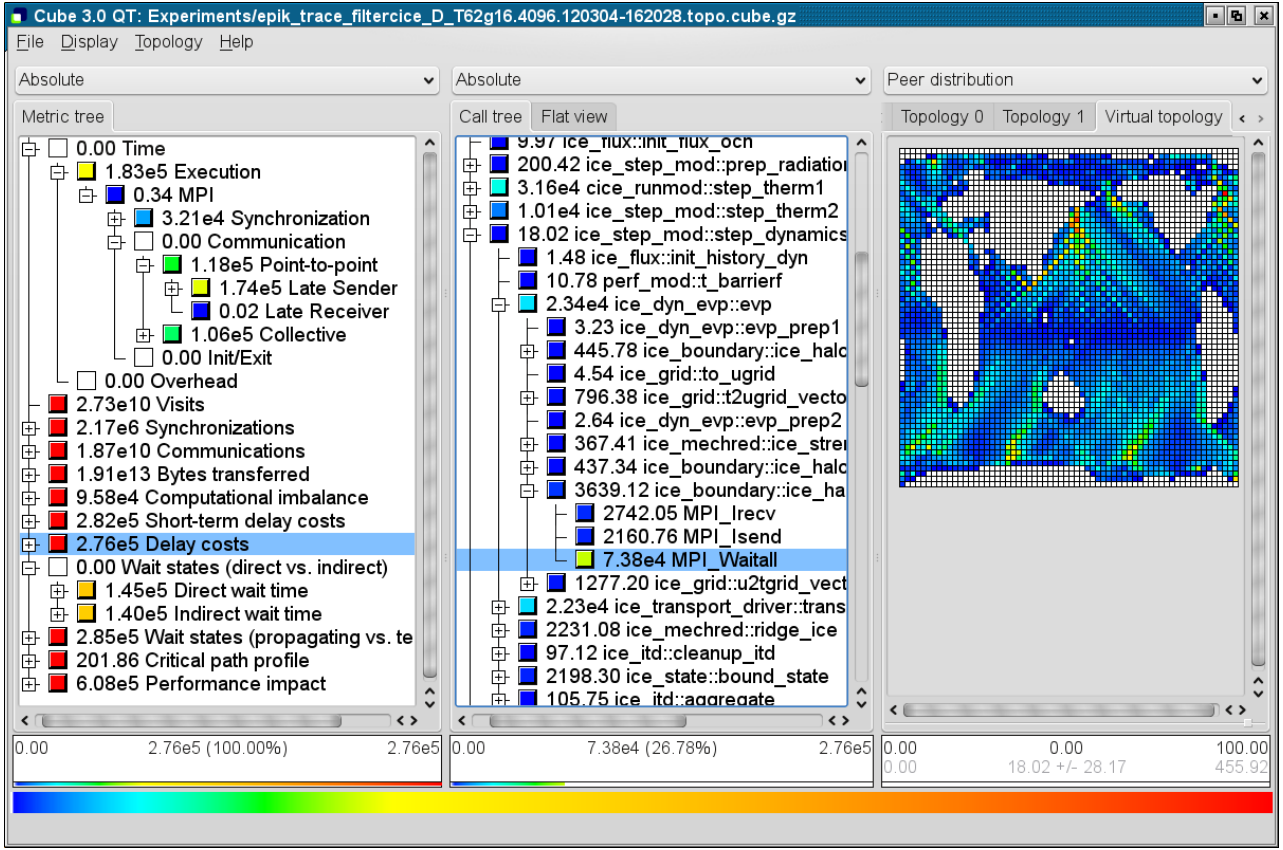
- Indirect waits occurs for ranks processing warmer areas



Scalasca Example: CESM Sea Ice Module

Delay Costs Analysis

- Delays **NOT** caused on ranks processing ice!



YOU KNOW YOU MADE IT …

… IF LARGE COMPANIES “COPY” YOUR STUFF

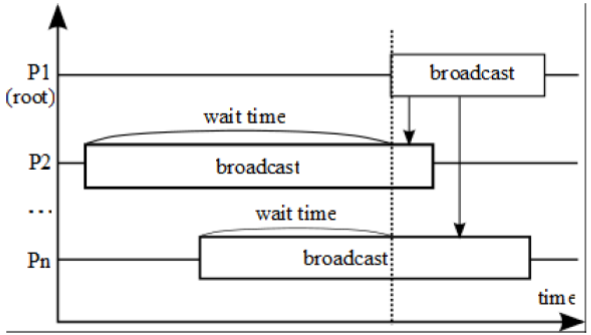
Introducing the Intel® Trace Analyzer and Collector Performance Assistant

Motivation: Improve method of performance analysis via the GUI

Solution:

- Define common/known performance problems
- Automate detection via the Intel® Trace Analyzer

Example: A “Late Broadcast” is not easy to identify with existing views

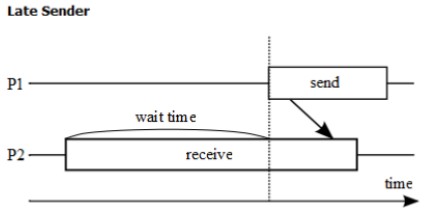


Source:
https://software.intel.com/en-us/videos/quickly-discover-performance-issues-with-the-intel-trace-analyzer-and-collector-90-beta

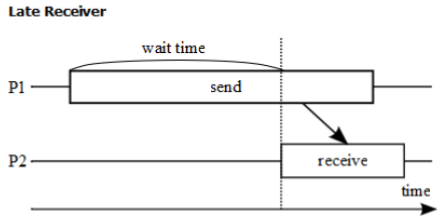
Which Performance Issues are automatically identified?

Point-to-point exchange problems:

- Late Sender

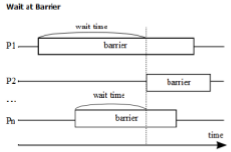


- Late Receiver

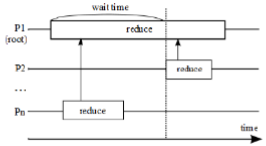


Problems with global collective operation performance:

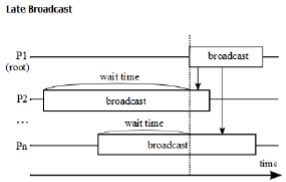
- Wait at Barrier



- Early Reduce



- Late Broadcast



Source: https://software.intel.com/en-us/videos/quickly-discover-performance-issues-with-the-intel-trace-analyzer-and-collector-90-beta

Together we are strong

INTEGRATION

Integration

- Need integrated tool (environment) **for all levels of parallelization**
  - Inter-node (MPI, PGAS, SHMEM)
  - Intra-node (OpenMP, multi-threading, multi-tasking)
  - Accelerators (OpenACC, CUDA, OpenCL)
- Integration with **performance modeling and prediction**
- No tool fits all requirements
  - **Interoperability of tools**
  - Integration via open interfaces

JÜLICH FORSCHUNGSZENTRUM

JÜLICH FORSCHUNGSZENTRUM

German Research School for Simulation Sciences

gns

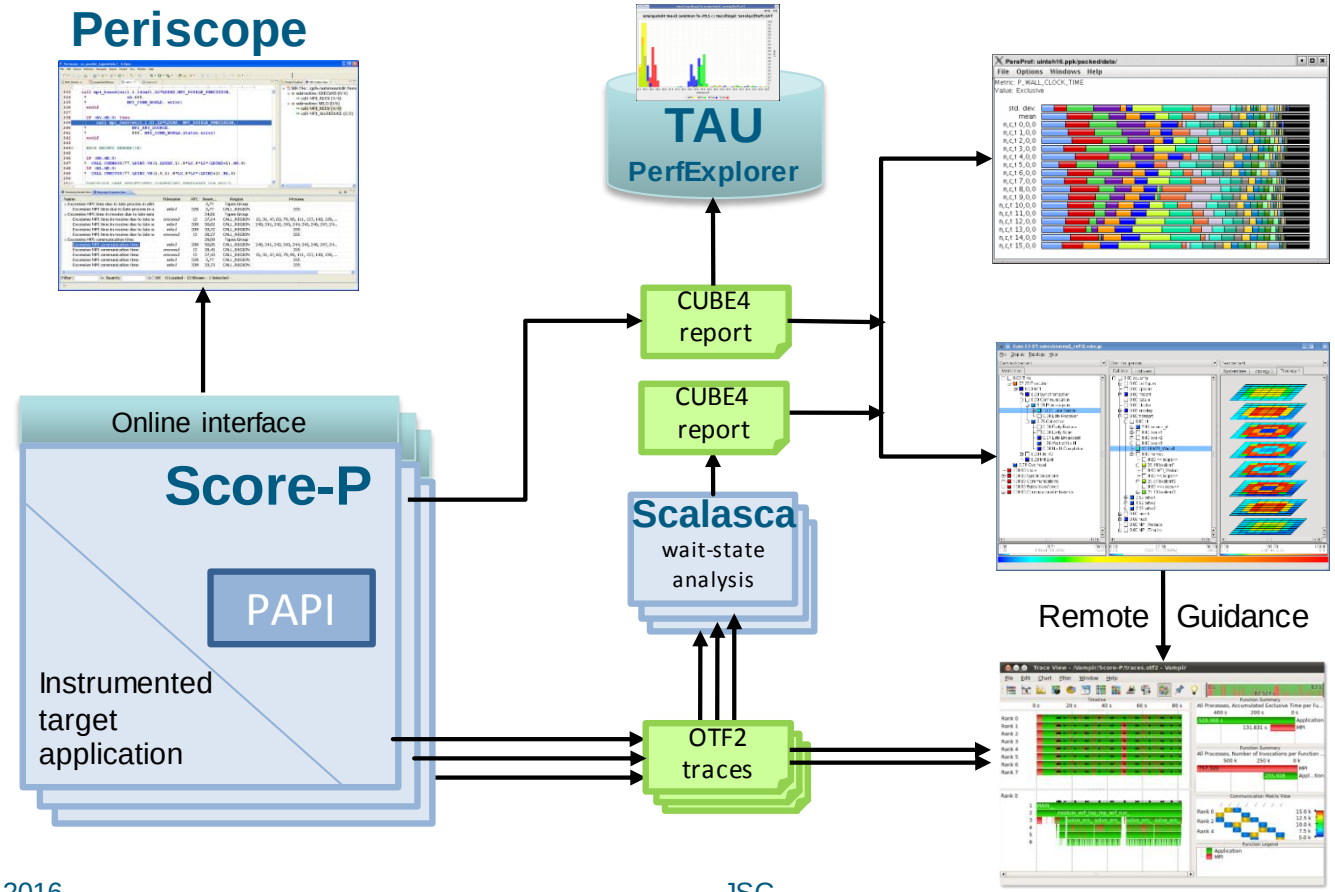
Technische Universität München

RWTH AACHEN UNIVERSITY

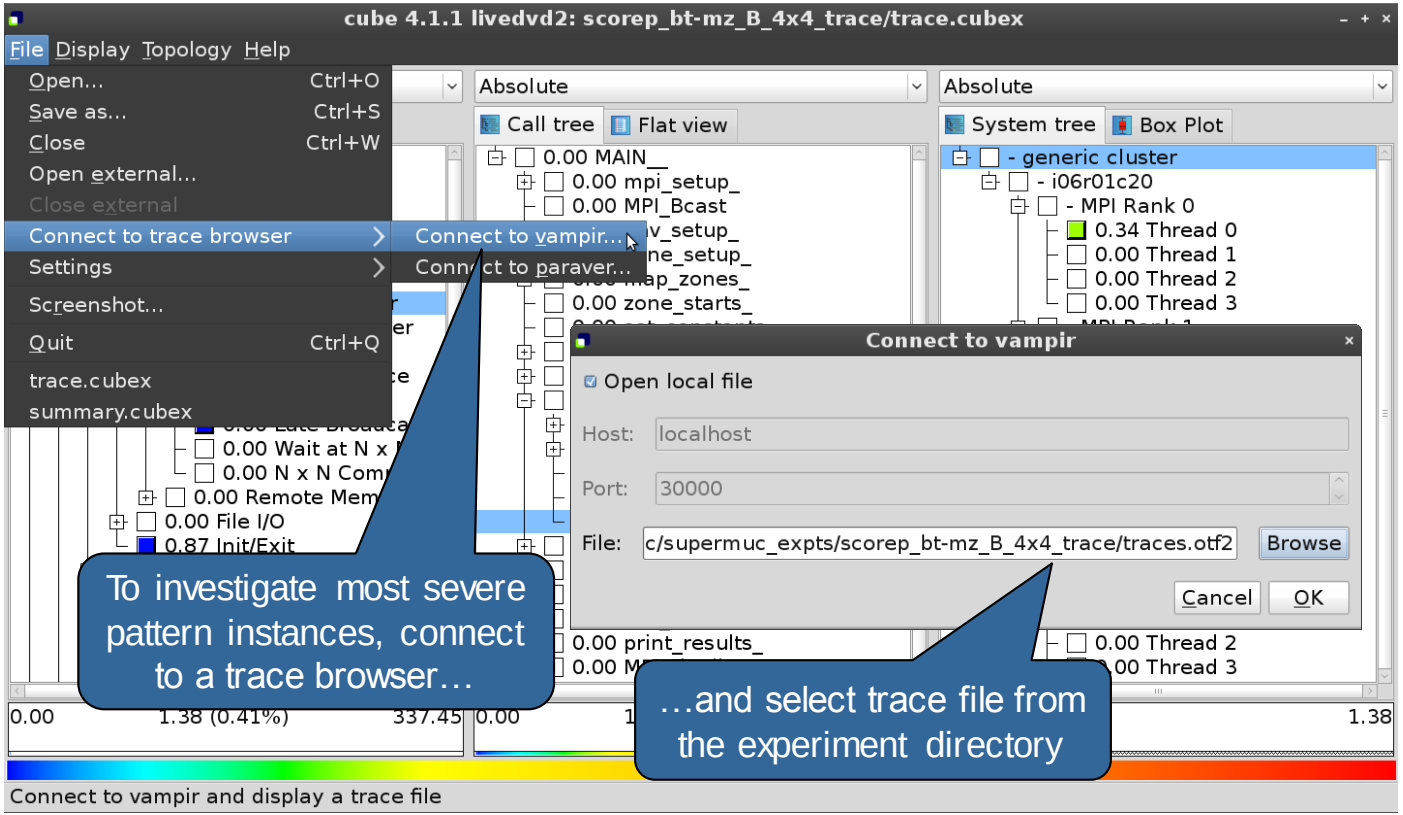
TECHNISCHE UNIVERSITÄT DRESDEN

UNIVERSITY OF OREGON

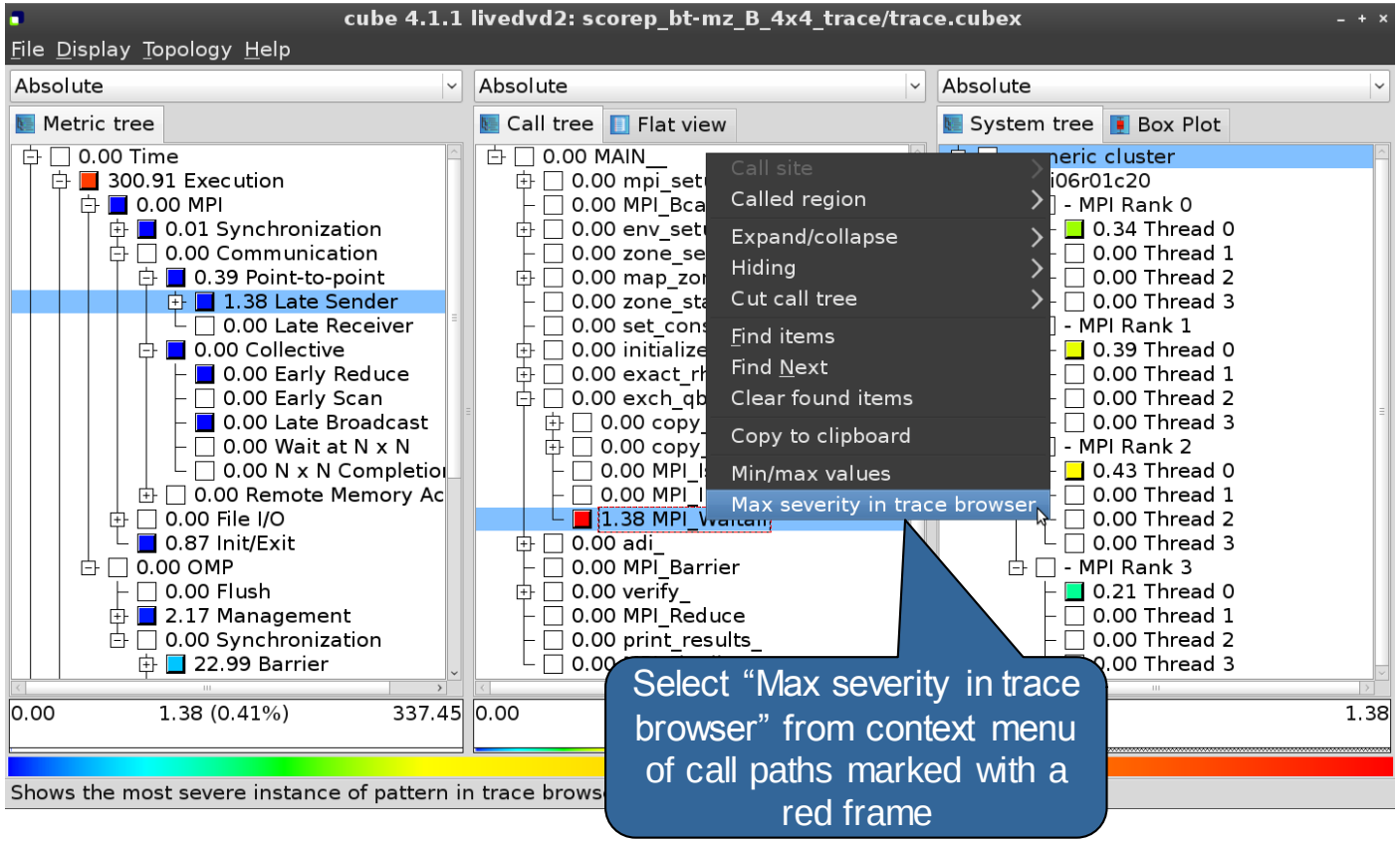
Score-P Tool Ecosystem



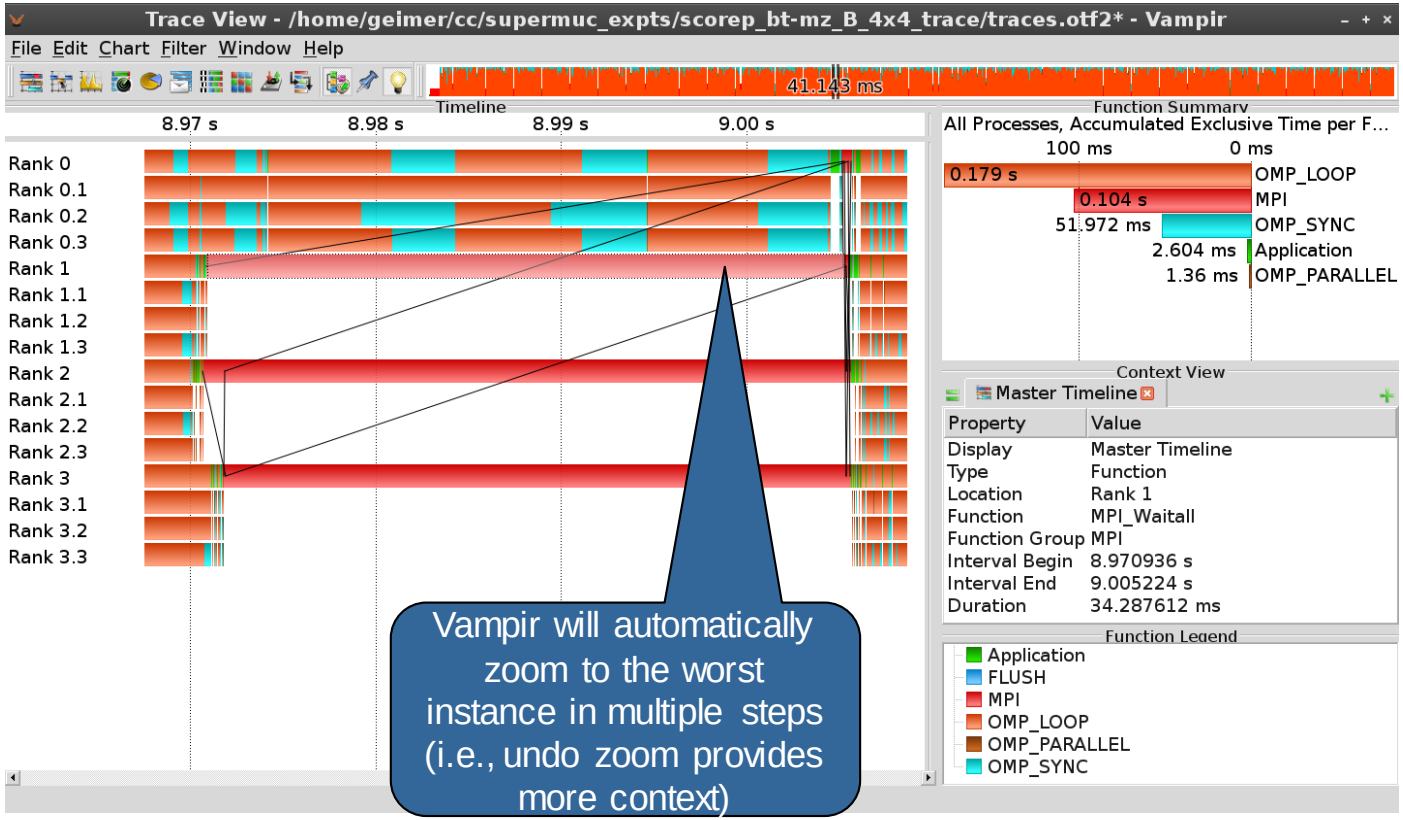
Connect to Vampir Trace Browser



Show most Severe Pattern Instances



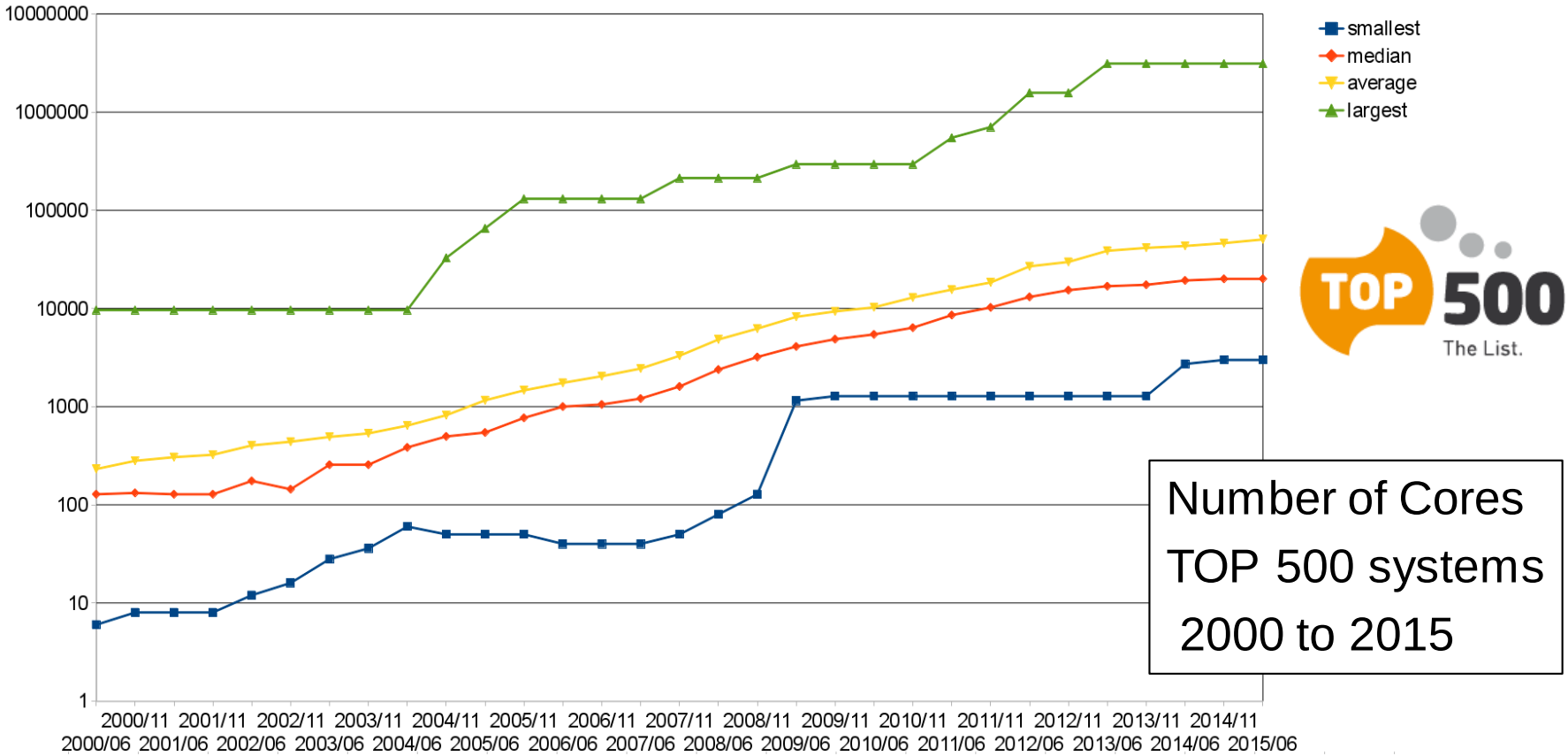
Investigate most Severe Instance in Vampir



To infinity and beyond

EXTREME CONCURRENCY

Typical HPC System Size



Number of Cores
TOP 500 systems
2000 to 2015

Personal Motivation

JuQueen

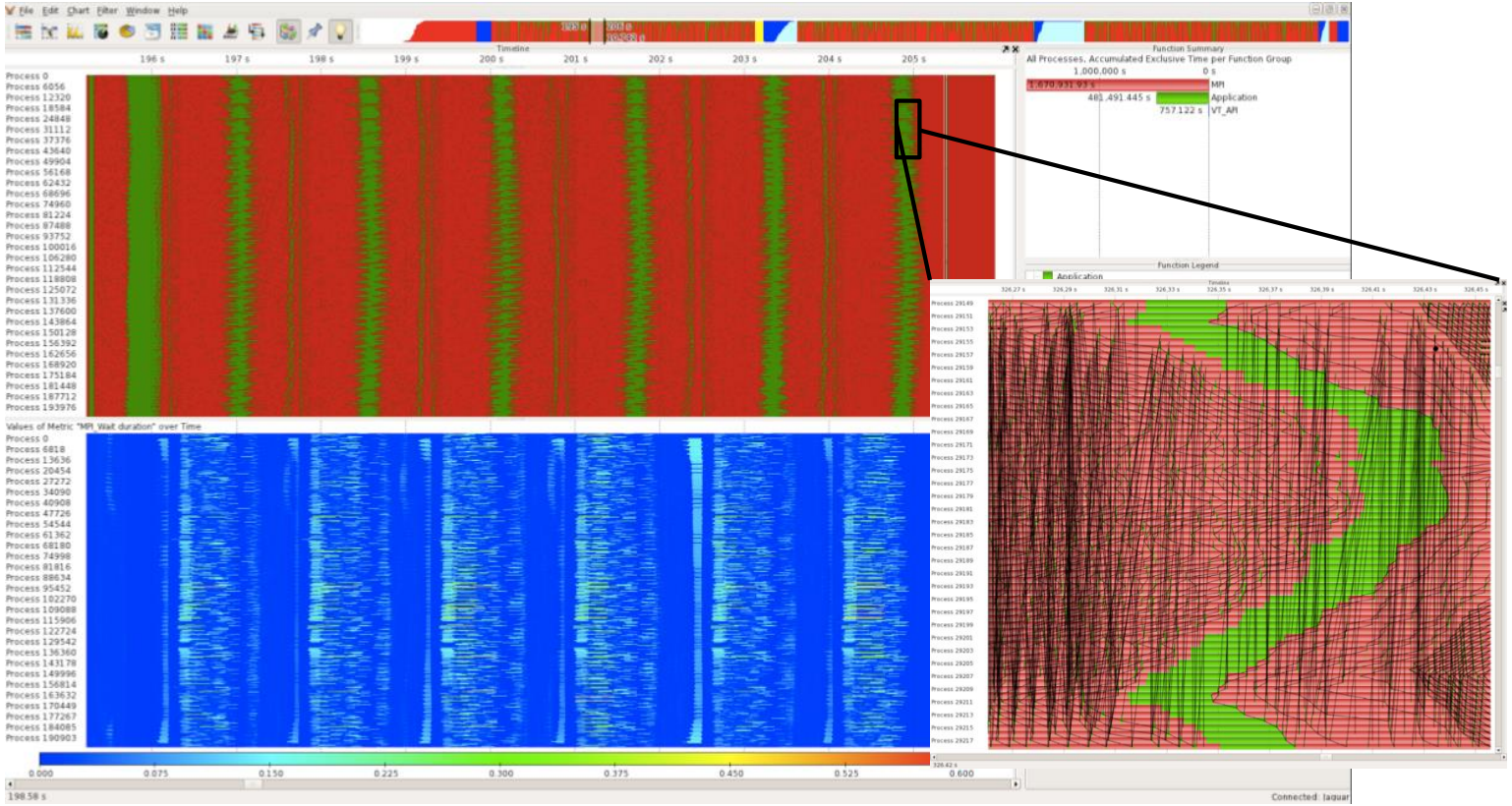
IBM BlueGene/Q
28 racks
458,752 cores
1,835,008 HW threads

2012/2013:
Most powerful machine in Europe!

Most **parallel** machine in Europe!

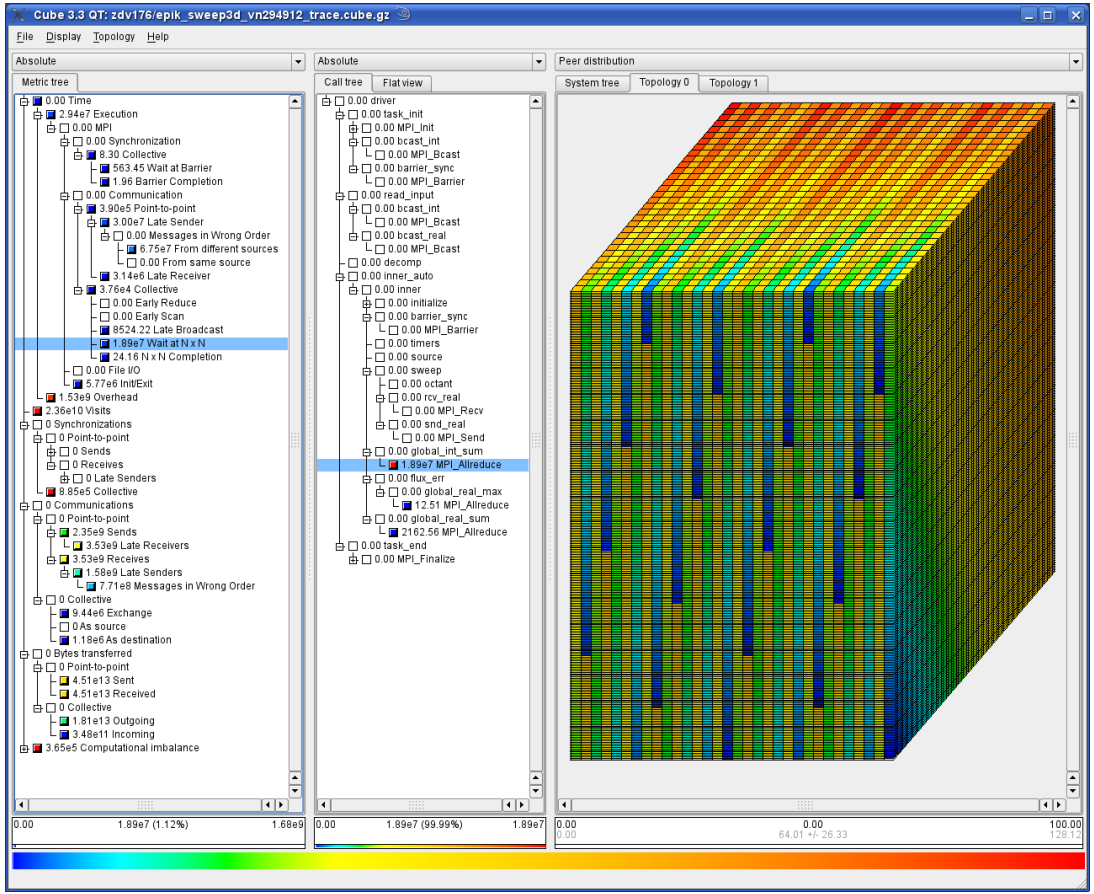
VampirServer: Trace Visualization S3D@200,448

- OTF trace 4.5 TB
- Vampir Server running with 20,000 analysis processes



Scalasca Trace Analysis sweep3D@294,912 BGP

- 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.

Performance Tool Scaling: Scalasca

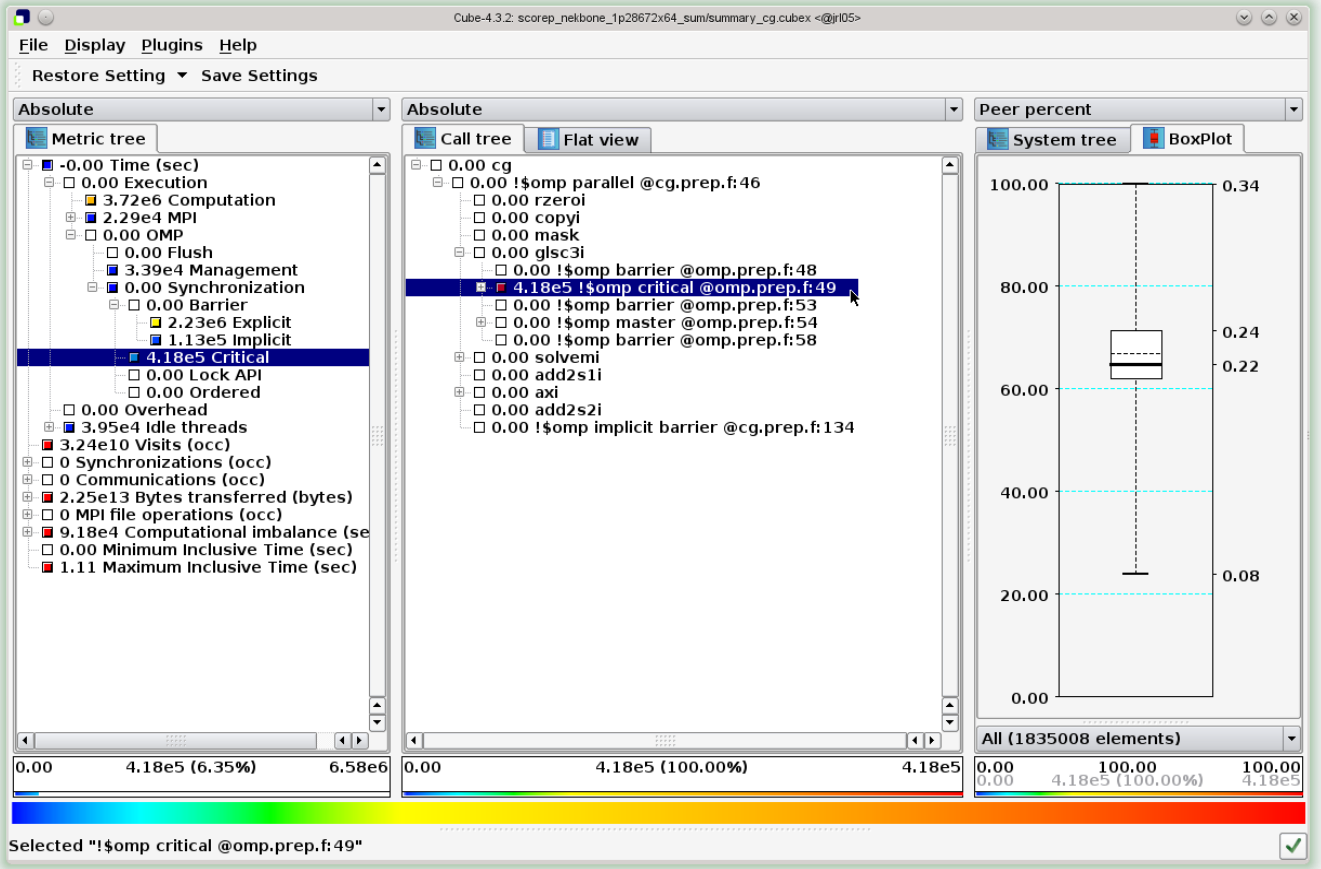
JÜLICH FORSCHUNGSZENTRUM

- **Latest full application test case**
  - Granular Dynamics Simulation
  - Based on Physics Engine (PE) Framework (Erlangen), MPI only
  - PRACE @ ISC Award winner
- **Scalasca 1.x** experiments on JUQUEEN
  - Full machine experiment: 28,672 nodes x 32 MPI ranks
    - **917,504 processes** [Limit: Memory / System metadata]
  - Largest number of threads: 20,480 nodes x 64 MPI ranks
    - **1,310,720 processes** [Limit: Memory / System metadata]
- **Scalasca 2.x / Score-P 1.4.1** Nekbone (CORAL benchmark) on JUQUEEN
  - Profiles: 28,672 x 64 = **1,835,008 threads !!!**
  - Traces: 10,240 x 64 = **655,360 threads** [Limit: OTF2]

Scalasca: 1,835,008 Threads Test Case

- Nekbone
- CORAL benchmark
- JuQueen experiment
- 28,672 x 64 = 1,835,008 threads

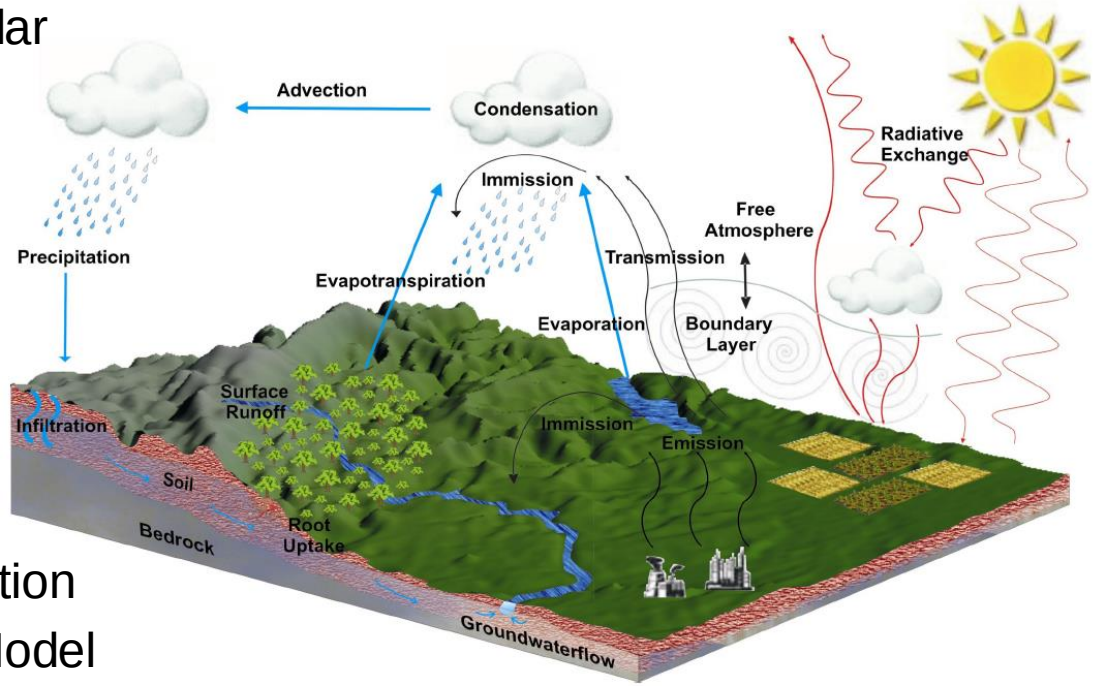
- Load imbalance at OpenMP critical section



USE CASES

Success Story: TerrSysMP

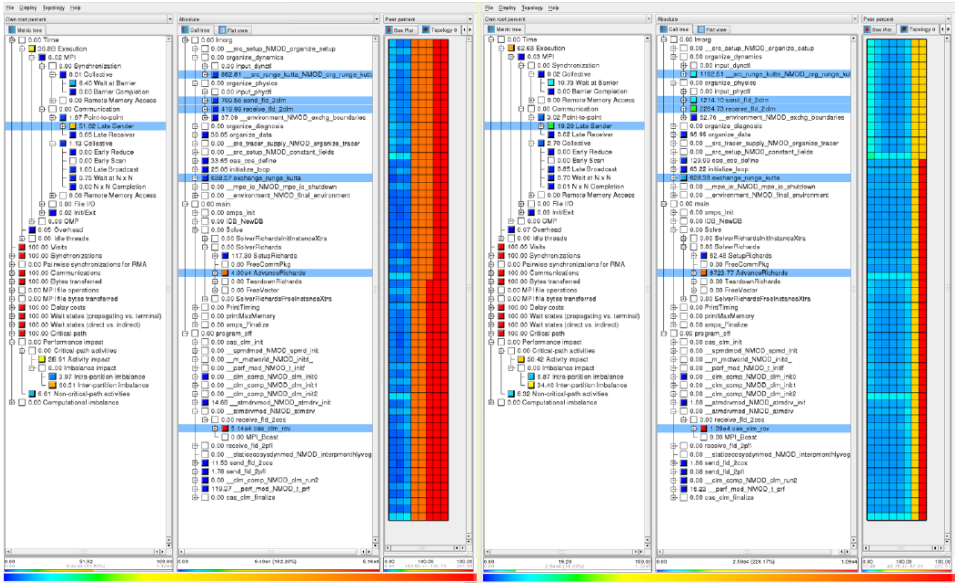
- Scale-consistent highly modular integrated multi-physics sub-surface/surface hydrology-vegetation atmosphere modelling system



- **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!**



ITEA 2

INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

H4H

Optimize Industrial HPC Applications on Heterogeneous Architectures

H4H Project Review#3, Repsol, Madrid, Spain 19th September 2013

H4H Application and Tool Coverage Overview

Company	Area	Performance Analysis	Programming Tools	High level tools
ATEME	Video compression	ThreadSpotter™ Scalasca, VampirTrace	HMPP	
Dassault Aviation	Simulation of aircraft design	ThreadSpotter™ Scalasca, Vampir PAS2P	HMPP	SAMG Scilab LAMA
Efield	Electromagnetic fields modeling and simulation	ThreadSpotter™ Scalasca, Vampir Marmot	HMPP	SAMG CuBLAS LAMA
GNS	Metal forming processes simulation	Scalasca/Score-P VampirTrace	HMPP	SAMG
INTES	Implicit finite element analysis system	VampirTrace		
MAGMA	Casting process simulation	ThreadSpotter™ Scalasca		SAMG, LAMA
RECOM	3D combustion	VampirTrace	OpenACC	
Repsol	Seismic imaging and reservoir simulation	PAS2P		Scilab

Main Achievements (Code Optimizations)

- **Significant performance improvements for Applications** through performance analysis, code restructuring / porting on GPU
- DASSAULT:
  - LAMA ⇨ sparse linear equations solver GPU-MPI ↑**3x to 4x faster**
  - Scilab/GPGPU ⇨ processing electromagnetic measurements ↑**3x faster**
  - Scilab/MPI ⇨ **very good scalability** reached in inverse design code
- RECOM:
  - Optimized particle deposition algorithm : ↑**7.9x** GPU vs CPU
- MAGMA:
  - Sparse linear equations solver on GPU **with SAMG** ↑**2x to 2.65x faster**
  - Sparse linear equations solver **on GPU with LAMA CG solver** ↑**3x faster**
  - MPI-GPU solver kernel version with CUDA yielded a **3x speedup** (8 CPU MPI vs 8 CPU / 1 GPU version)

Main Achievements (Code Optimizations)

- GWT:
  - IUMD code ➔ **13x speedup** on GPU compared 16 OpenMP Threads Intel Xeon E5 CPU
- INTES:
  - XPU (SMP+GPU) 🡅**2.5x faster than pre-existing** 16-core SMP version
  - Combined with additional DMP parallelization, a static analysis of a motor block benchmark even showed a **speedup of 68.6** against the single core time (total job) → Turnaround time for typical simulation cycles **reduced from 1 week to about 2 hours** (SMP+XPU+DMP).
- REPSOL:
  - 🡅**1.9x** Speedup for seismic code with GPUs and linear scaling with MPI
  - 🡅**2.6x** Speedup for Reservoir Simulator with GPUs speed up of reservoir

SIEMENS

JÜLICH
FORSCHUNGSZENTRUM

Runtime **A**nalysis of **P**arallel applications for **I**ndustrial software **D**evelopment

- **Collaboration** between
  - Corporate Technology Multicore Expert Center of Siemens AG
  - Jülich Supercomputing Centre
- **Results**
  - Tool support (Score-P, Scalasca, Vampir) for
    - Threading models (POSIX, QT, ACE, Windows)
    - MTAPI (Multicore Association Task API)
  - Windows port of basic Score-P and Cube

SIEMENS

Corporate Technology | Siemens AG

ScoreP, Scalasca and Vampir @ Siemens AG

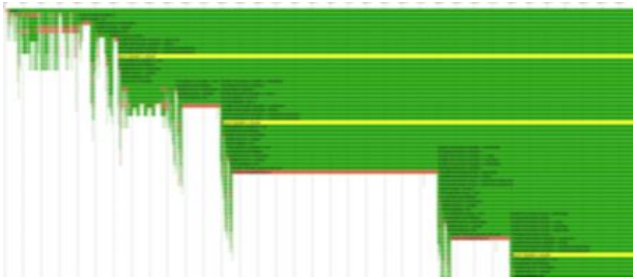
Multicore
Expert Center

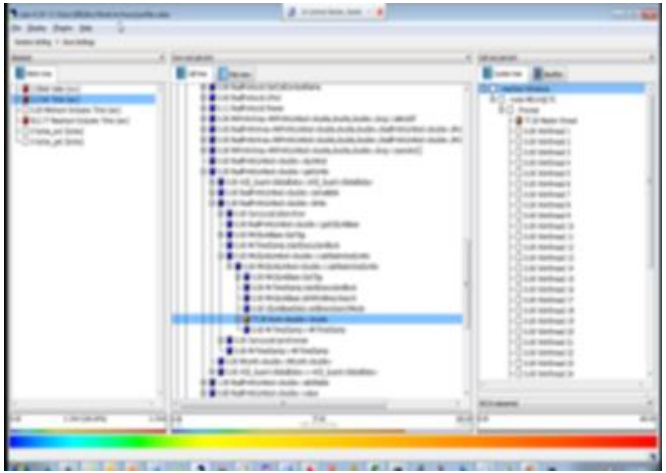
Targeted Analysis using adapted ScoreP allows relevant focussing

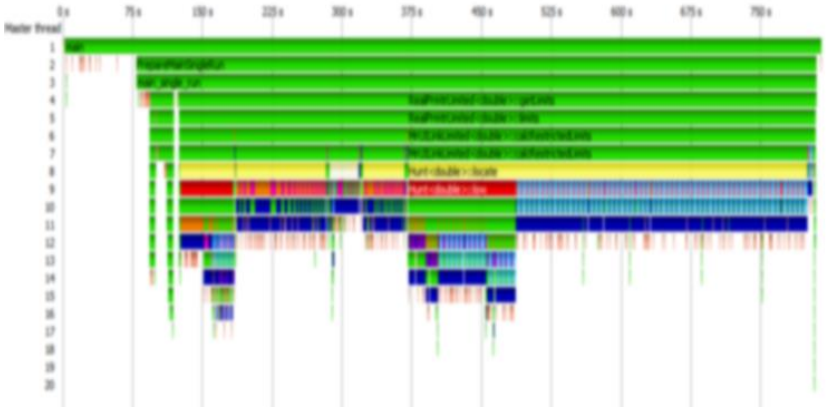
SIEMENS

Via ScoreP, Cube, Vampir

- Effective (sub-)hotspot localization (80%)
- Identified legacy threads
- Gained insight into call hierarchy/stack







Questions?

scalasca

- http://www.scalasca.org
- scalasca@fz-juelich.de

Score-P
Scalable performance measurement infrastructure for parallel codes

- http://www.score-p.org
- support@score-p.org