Scalable Performance Analysis of Large-scale Applications

Markus Geimer
m.geimer@fz-juelich.de

Forschungszentrum Jülich
in der Helmholtz-Gemeinschaft

Central Institute for Applied Mathematics

- One of the most powerful scientific computing centers in Europe
  - John von Neumann Institute for Computing

- Research
  - Methodological advancement of supercomputing
  - Operation of supercomputers as scientific large-scale devices
  - Essential component: performance analysis tools

Outline

- Motivation
- Performance measurement & analysis
- Addressing scalability: SCALASCA
- Experimental evaluation
- Conclusion
- Outlook

Increasing parallelism

- Advanced numerical simulations harness higher degrees of parallelism
  - Custom-built large-scale systems
  - More CPU cores instead of higher clock speeds



- Scalability is a major concern

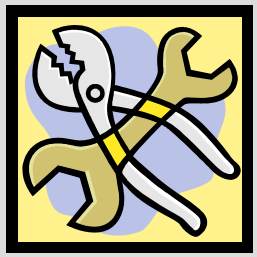
Performance gap

- Available systems are not used efficiently
  - Sustained application performance << peak performance
- Growing size and complexity of platforms and codes
  - Limited parallelism in applications
  - Hierarchies of latencies and bandwidths
  - Remote data accesses
  - Multi-physics applications

Optimization difficult and time consuming

Cost-effective development of efficient code

- Higher degrees of parallelism
  - Also new demands on scalability of software tools
- Traditional tools cease to work in a satisfactory manner for large processor counts
  - Escalating memory requirements, limited I/O bandwidth, etc.



- Scalable performance tools must become an integral part of the software-development environment

Will have significant impact on the overall productivity of high-performance computing systems

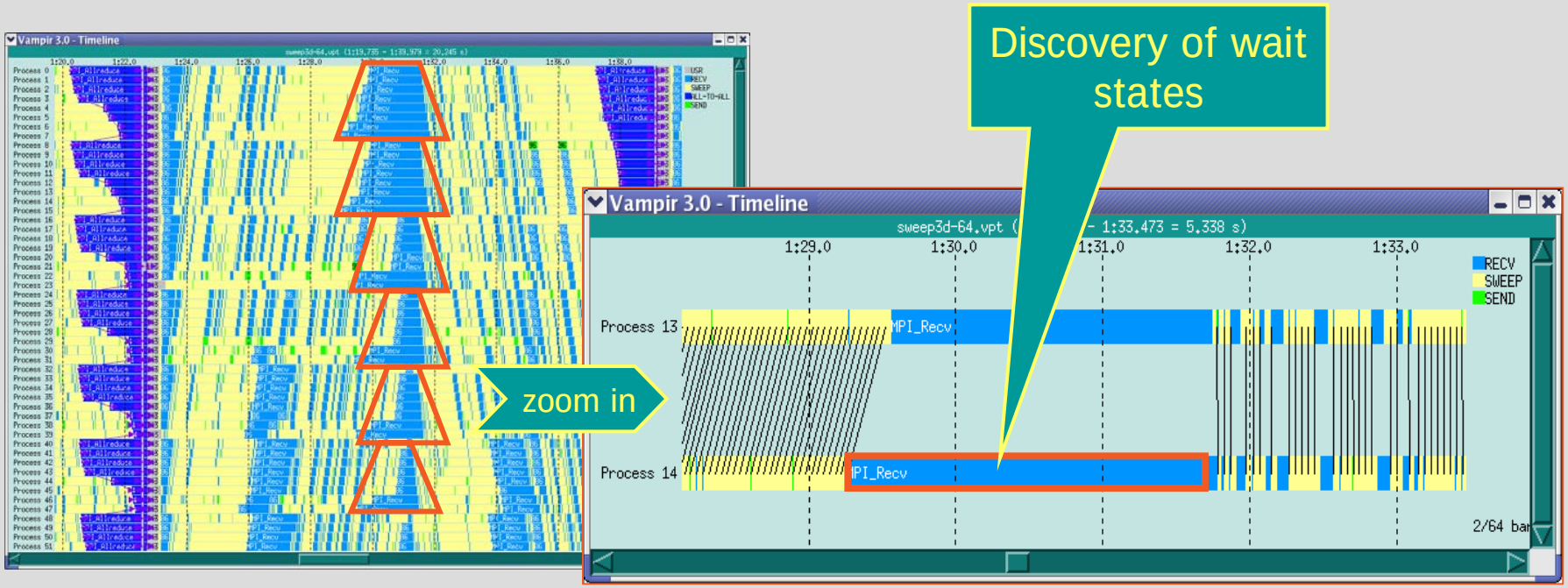
Runtime summarization vs. event tracing

- Process-local calculation of various metrics, e.g.,
  - Time spent in each function
  - Message statistics
  - Hardware counters
- Summarized in single report
- Provides overview of program's execution

- Recording of time-stamped events at runtime, e.g.,
  - Entering / leaving a function
  - Sending / receiving a message
  - Collective operations
- Post-mortem analysis
- High level of detail through reconstruction of dynamic program behavior

Both approaches are valuable and should be tightly integrated to produce the best results possible

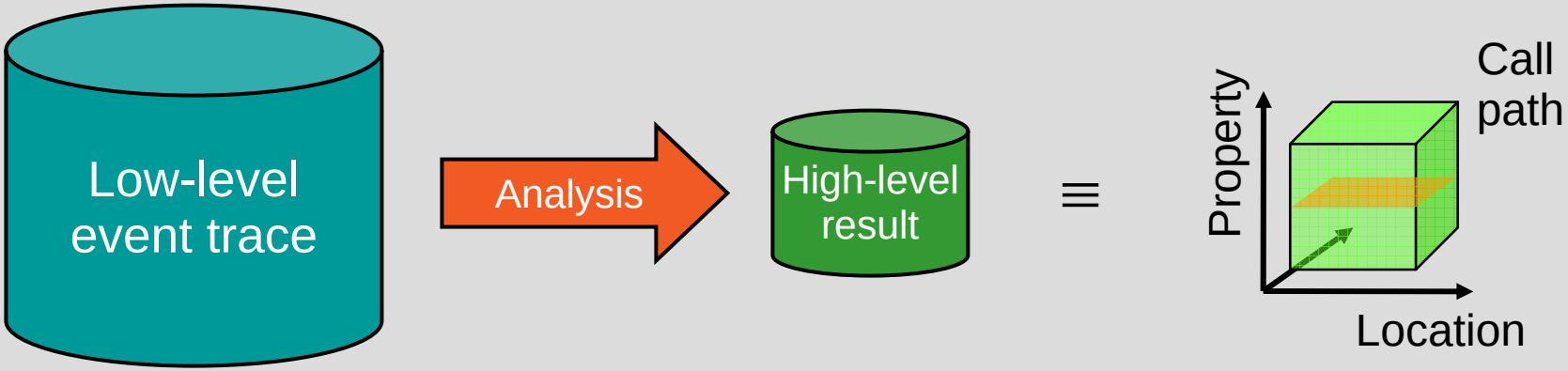
Time-line visualization



- Useful for fine-grained investigation of performance problems
- “Human client”

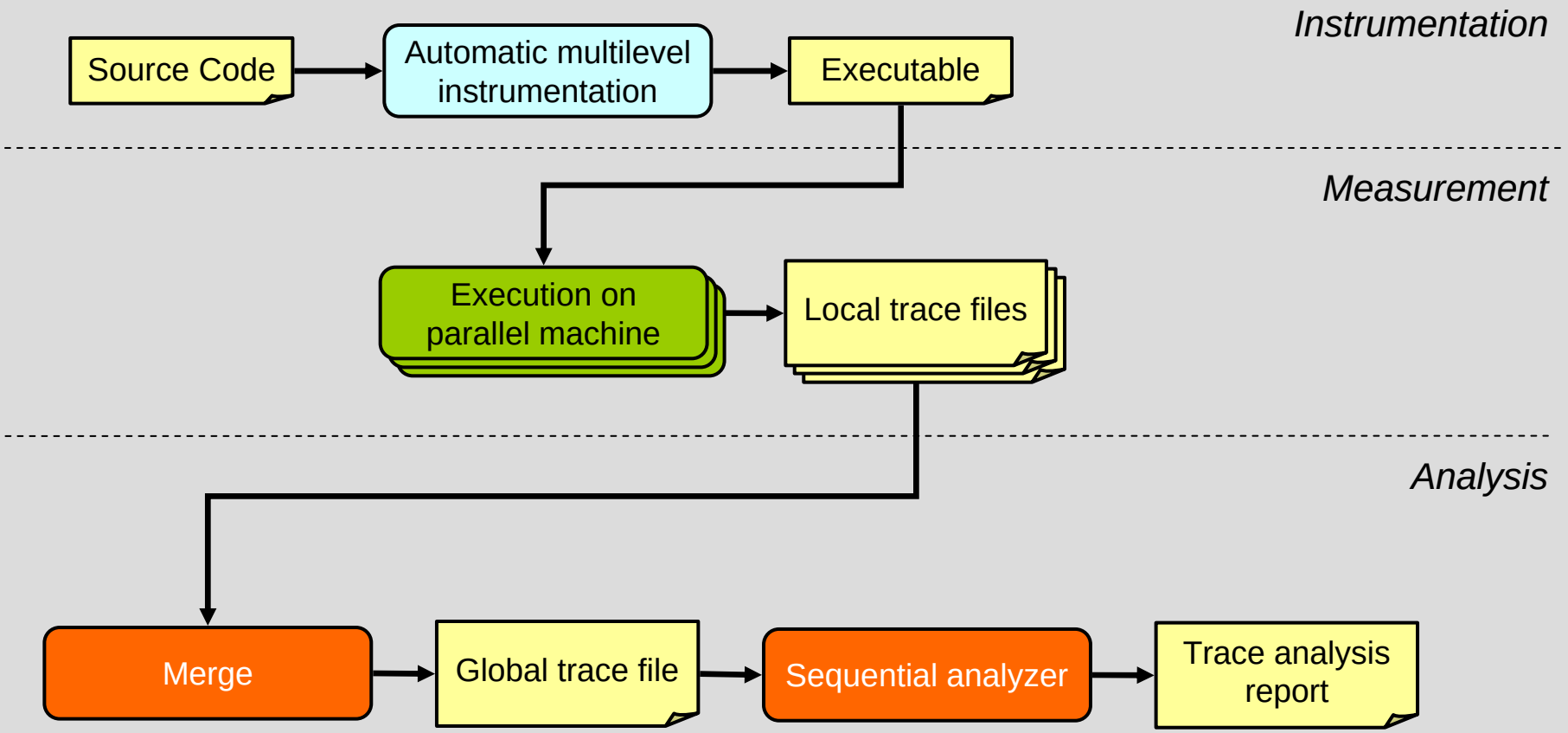
Automatic off-line trace analysis

- Idea
  - Automatic search for *patterns* of inefficient behavior
  - Classification of behavior
  - Quantification of significance



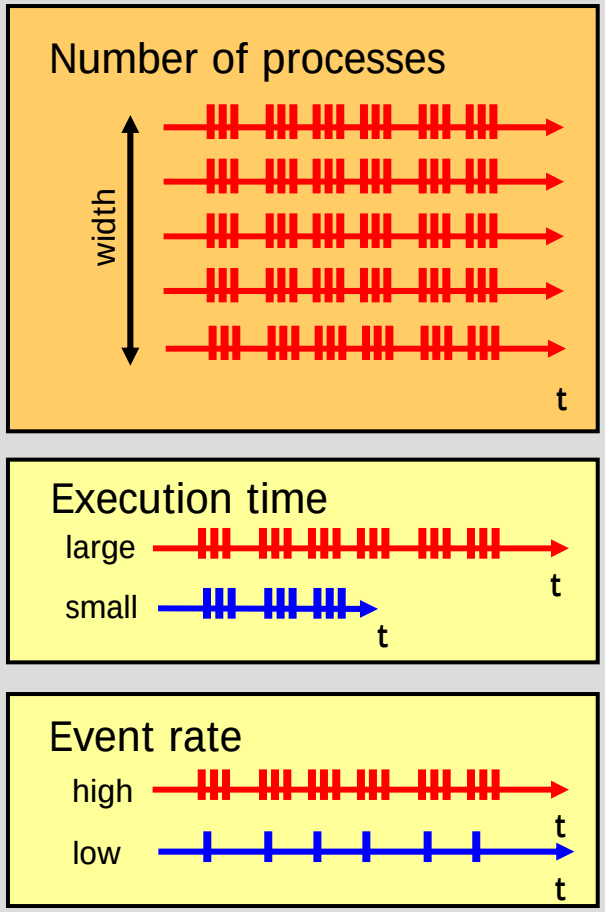
- Guaranteed to cover the entire event trace
- Identifies bottleneck instances
  - Can be used to direct time-line visualization

Sequential trace analysis approach



Trace size limits scalability

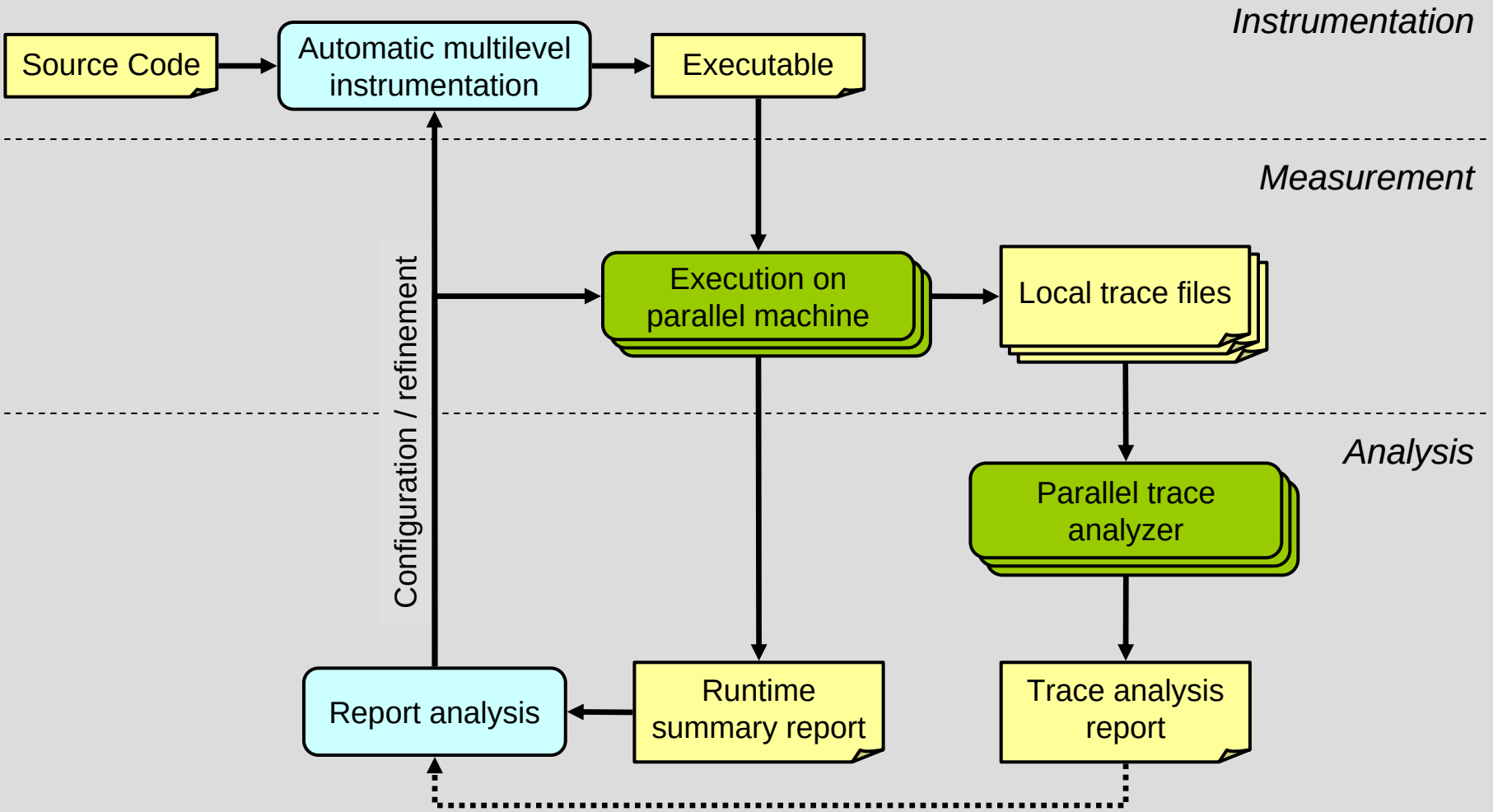
- Serially analyzing a single global trace file does not scale to 1000s of processors
- Main memory might be insufficient to store context of current event
- Amount of trace data might not fit into single file



SCALASCA

- Helmholtz-University Young Investigators Group
  - Started in January 2006
  - Funded by Helmholtz Initiative and Networking Fund
- Objective: develop a scalable performance analysis tool
  - Basic ideas:
    - Parallelization of trace analysis
    - Integration of runtime summarization
  - Current focus: single-threaded MPI-1 applications

SCALASCA's integrated analysis process



Parallel pattern analysis

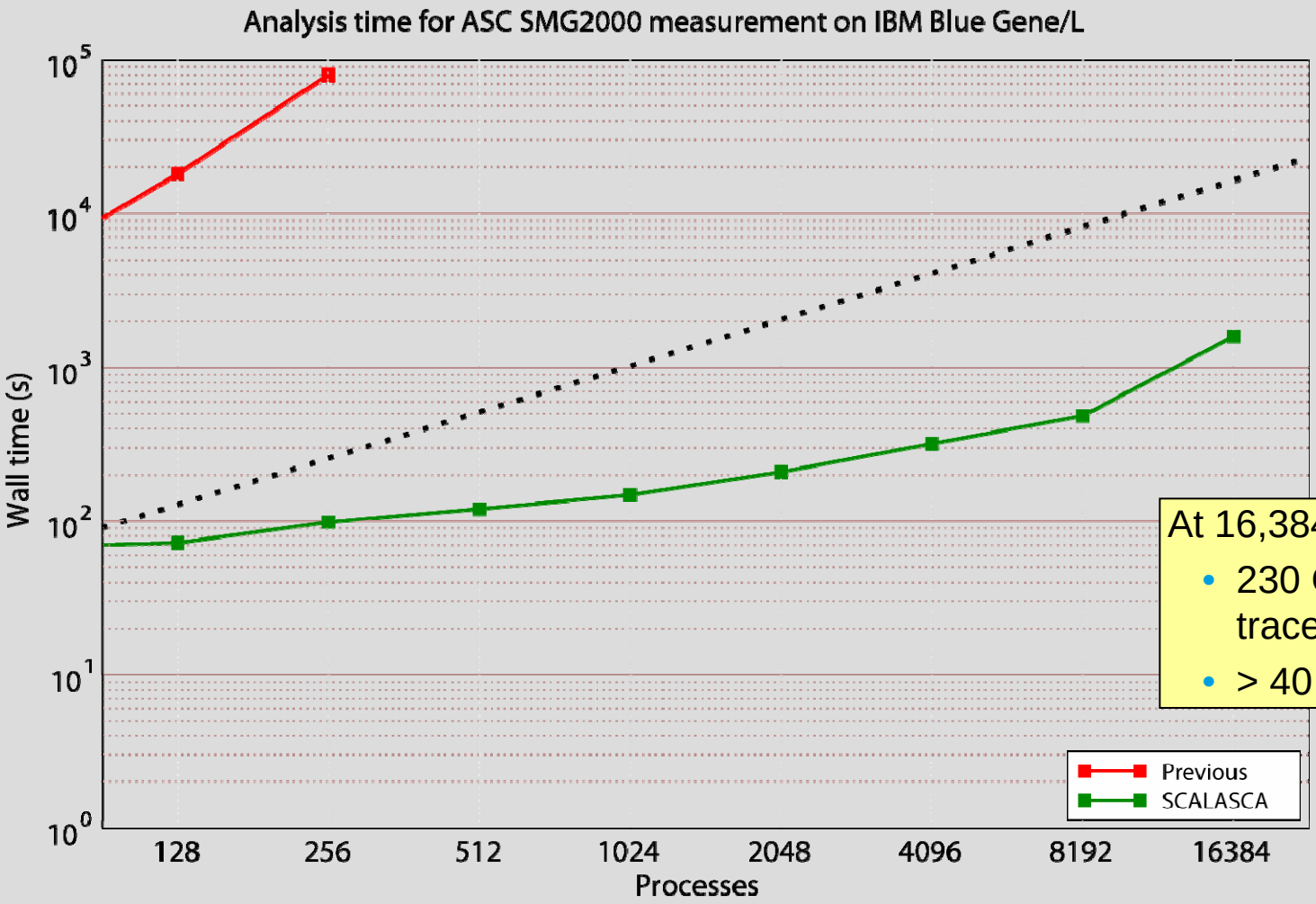
- Analyze separate local trace files in parallel
  - Exploit distributed memory and processing capabilities
  - Often allows keeping whole trace in main memory

- Parallel replay of target application‘s communication behavior
  - Analyze communication with an operation of the same type
  - Traverse local traces in parallel
  - Exchange data at synchronization points of target application

Experimental evaluation

- Scalability test
  - ASC SMG2000 benchmark
  - Semi-coarsening multi-grid solver
  - Fixed problem size per process – weak scaling behavior
- Application analysis
  - XNS fluid dynamics code
  - FE simulation on unstructured meshes
  - Constant overall problem size – strong scaling behavior
- Test platform: IBM Blue Gene/L in Jülich (JUBL)
  8 Racks with 8192 dual-core nodes

Scalability of trace analysis

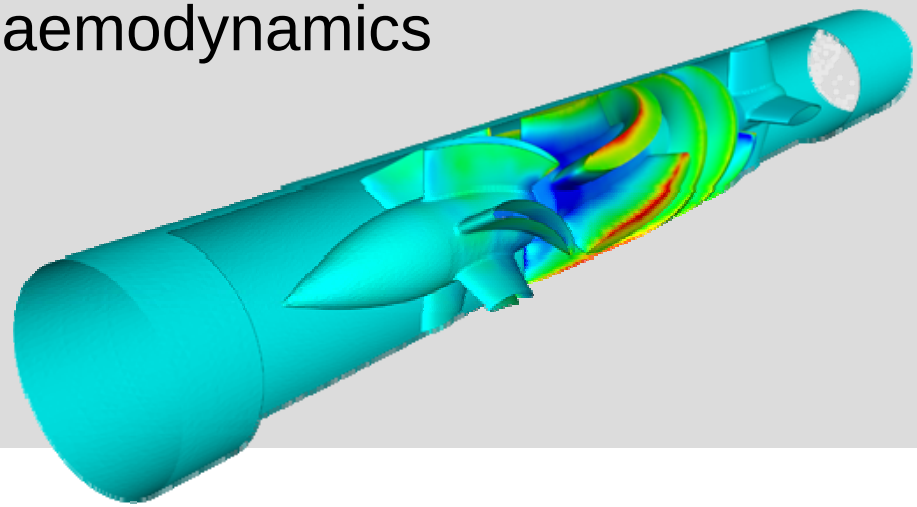


At 16,384 CPUs:

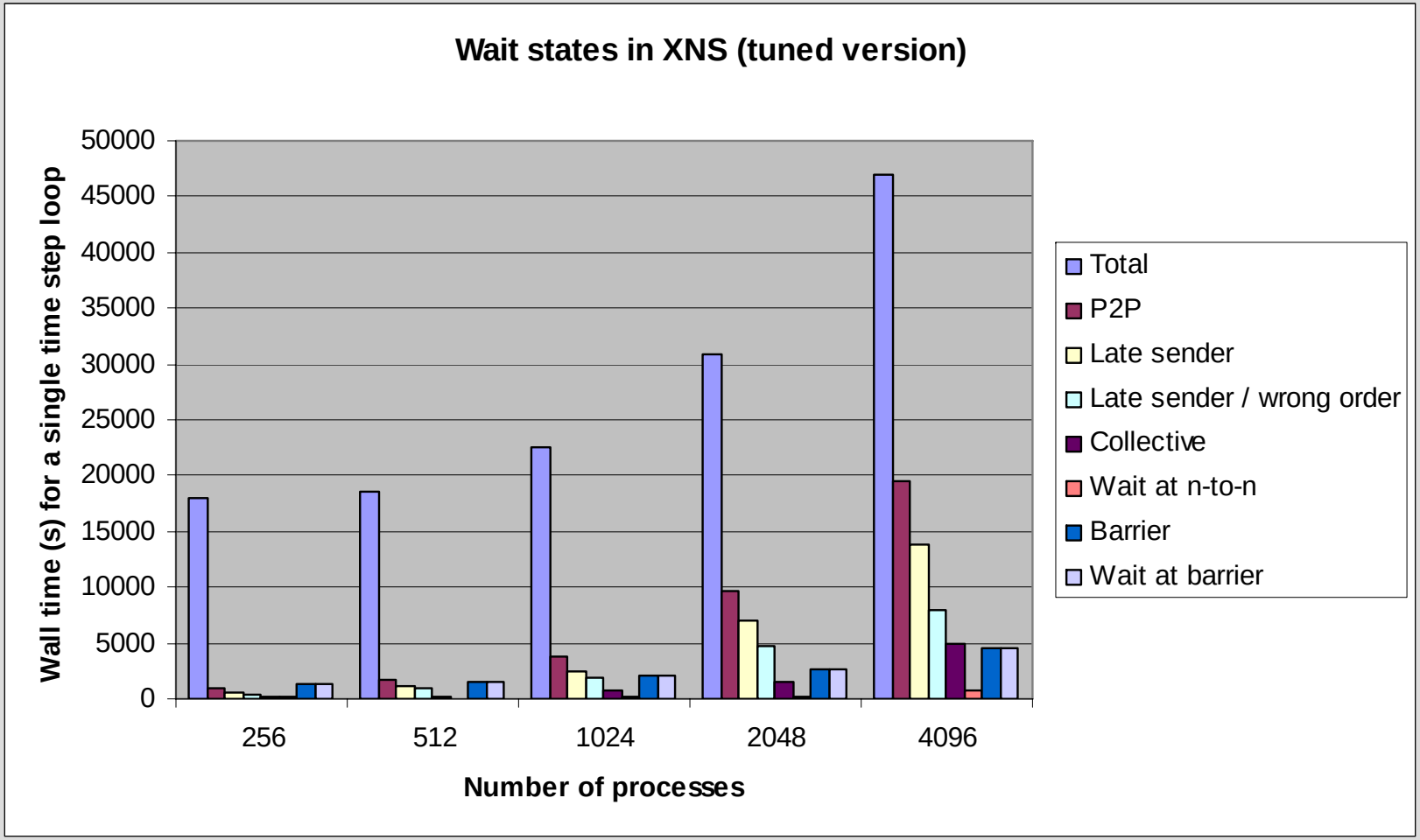
- 230 GB compressed trace data
- > $40 \cdot 10^9$ events

XNS CFD application

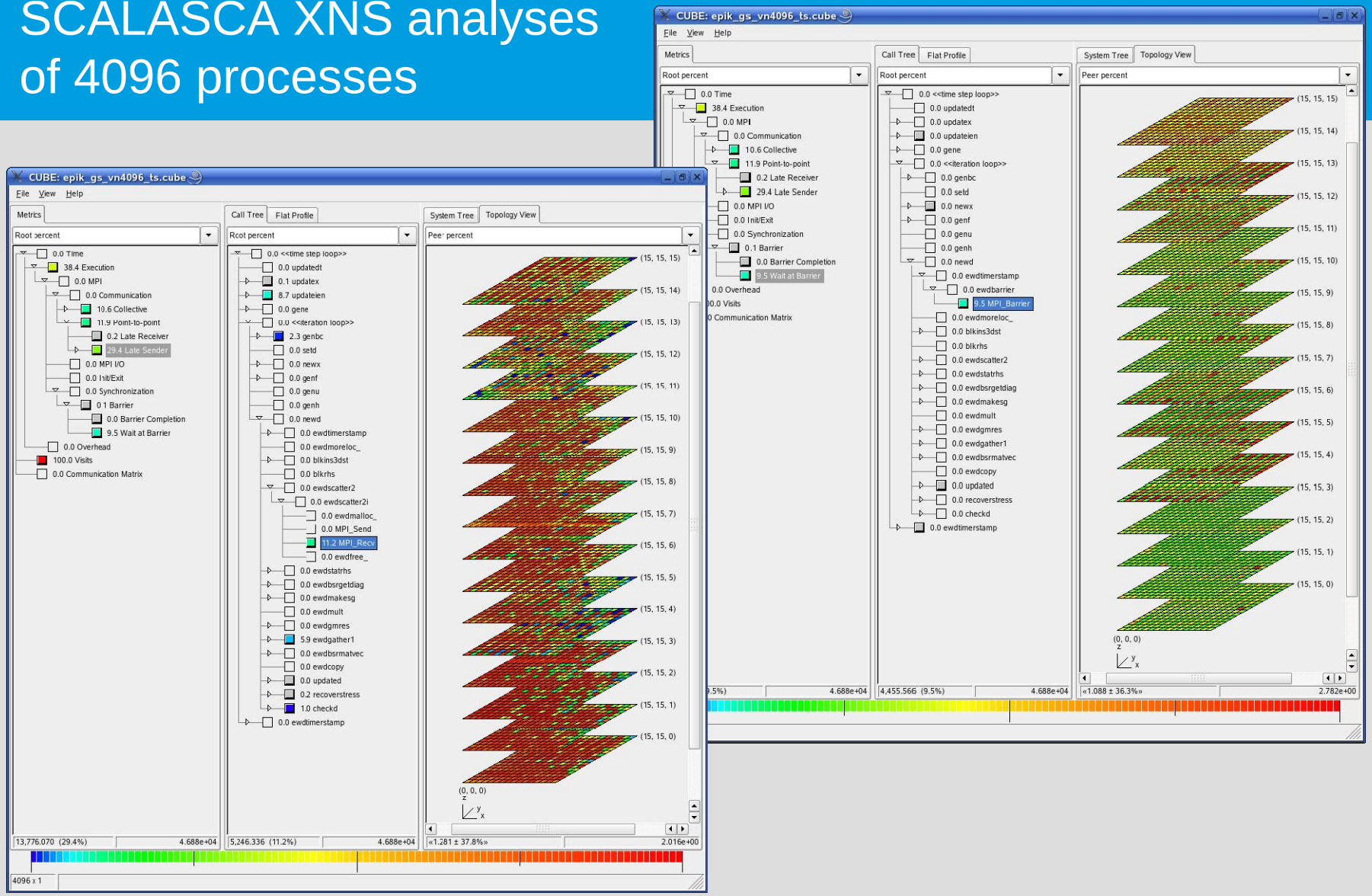
- Academic computational fluid dynamics code for simulation of unsteady flows
  - Developed by group of Marek Behr, Computational Analysis of Technical Systems, RWTH Aachen University
  - Exploits finite-element techniques, unstructured 3D meshes, iterative solution strategies
  - >40,000 lines of Fortran90 using MPI
- Simulation of blood pump haemodynamics



SCALASCA: Wait states in tuned version of XNS



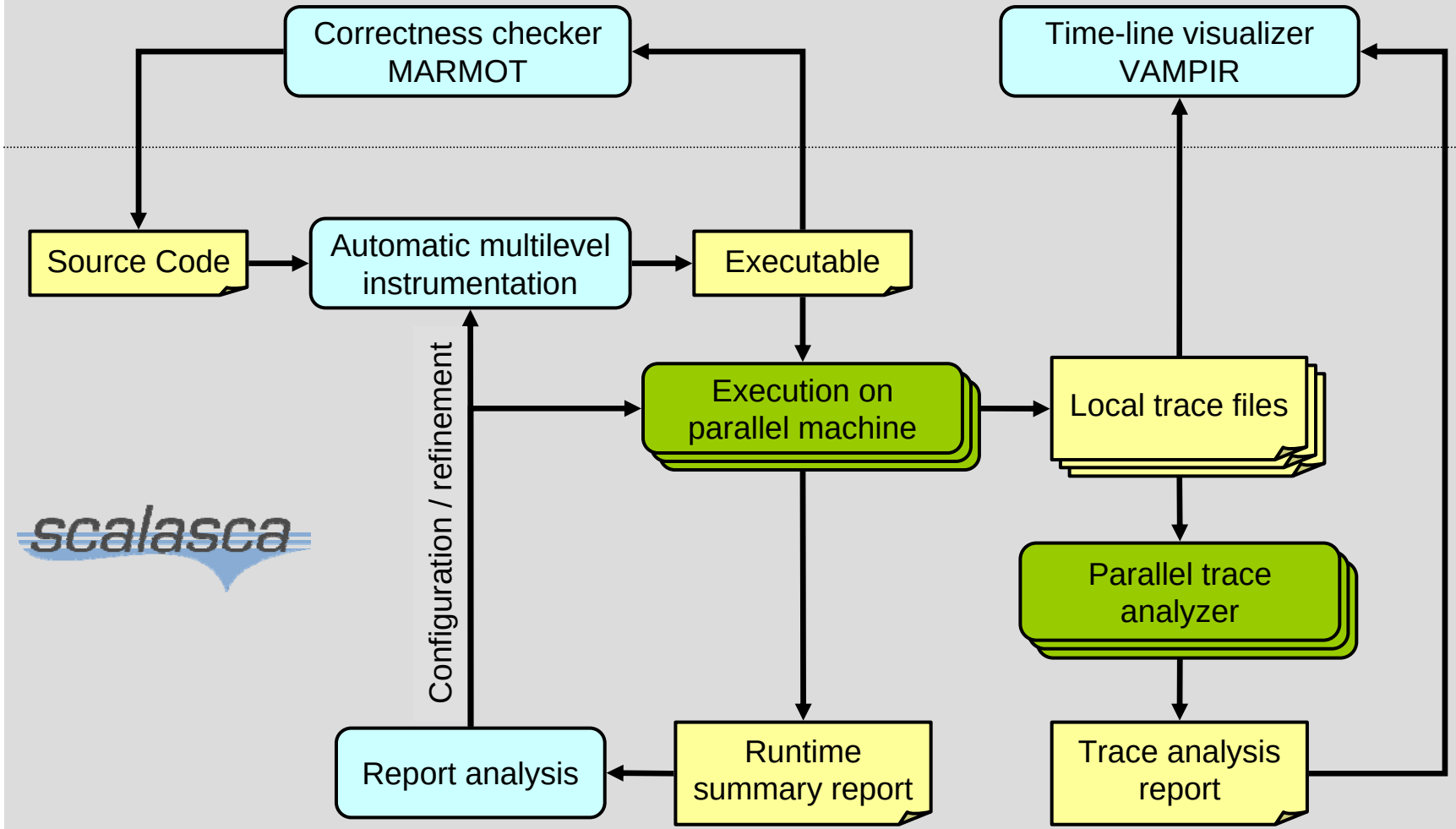
SCALASCA XNS analyses of 4096 processes



Conclusion

- Wait states addressed by our analysis can be significant performance problems – especially at larger scales
- Scalability of the trace analysis can be addressed by parallelization
  - Process local trace files in parallel
  - Replay target applications communication behavior
- Promising results with prototype implementations
  - Analysis scales up to 16,384 processes
  - Enables analyzing traces of previously impractical size

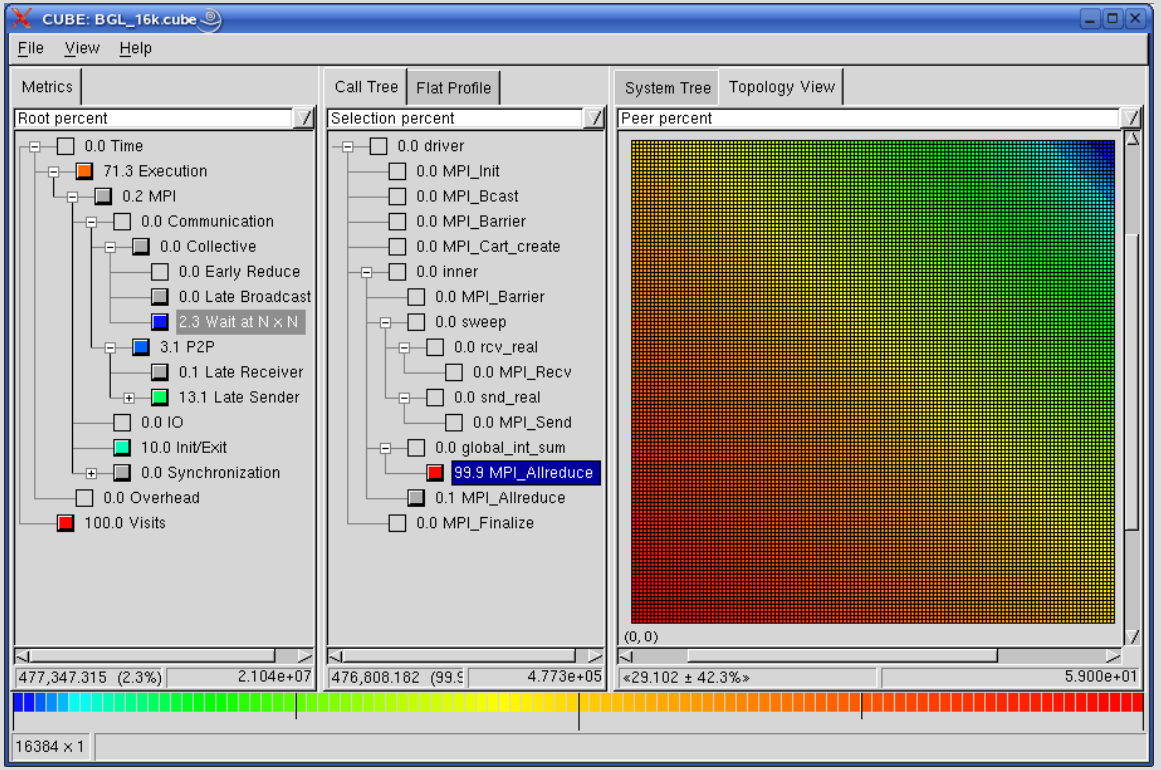
Vision: Integrated tool environment



Thank you!

For more information, visit our project home page:

http://www.scalasca.org



SWEEP3D virtual topology, Wait at NxN, 16K CPUs