scalasca

Scalasca Trace Tools

Scalable performance analysis of large–scale parallel applications

Performance Analysis

Measurement

Tuning

scalasca

Optimized Application
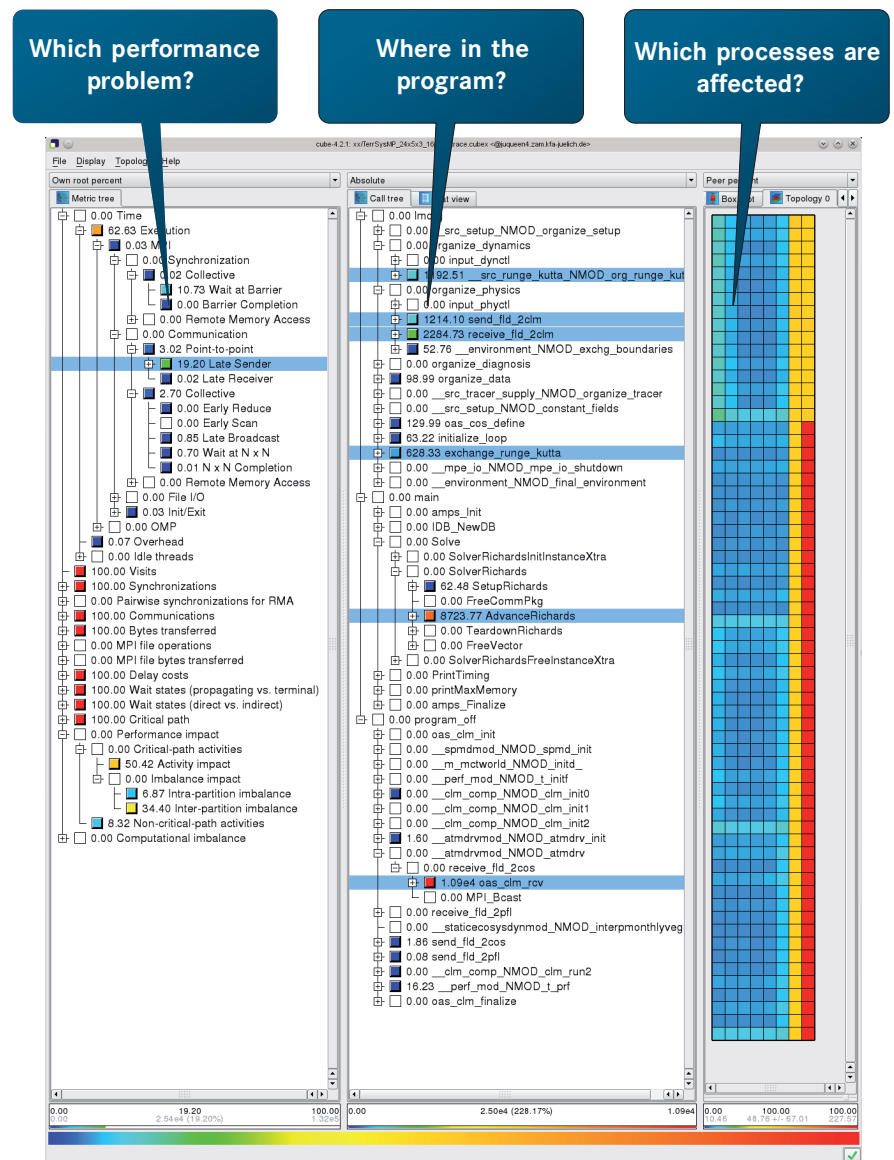
Trace-based performance-analysis software

- Specifically designed for large-scale systems
- In-depth studies of concurrent behavior via event traces
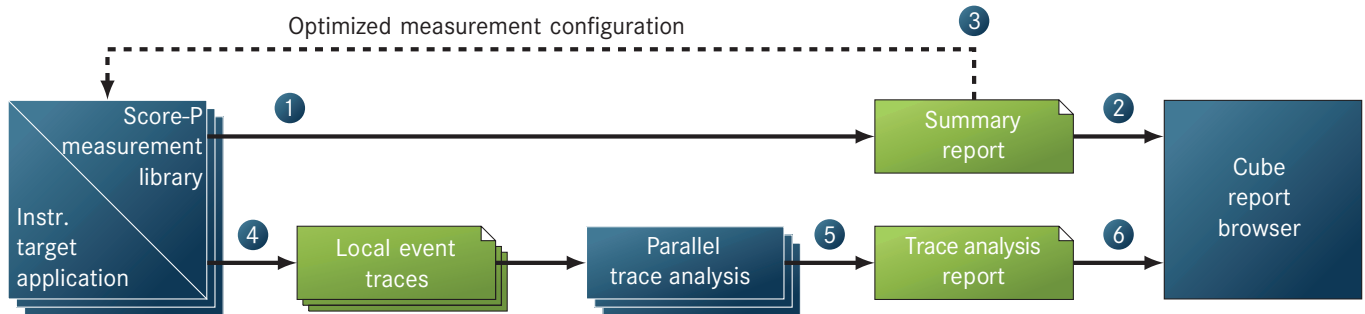- 3-clause BSD open-source licence

Features

- Localization of wait states & their root causes on large processor configurations
- Identification of the critical path
- Support for MPI, OpenMP, Pthreads, and hybrid MPI+OpenMP/Pthreads
- Based on the community–driven instrumentation & measurement infrastructure Score-P
- Uses open data formats OTF2 and CUBE4

Supported Platforms

- IBM BladeCenter & iDataPlex clusters
- HPE Cray XC series (x86, ARM)
- Linux-based clusters (x86, Power, ARM)

Which performance problem?

Where in the program?

Which processes are affected?

www.scalasca.org

## JÜLICH
Forschungszentrum

Scalasca Measurement & Analysis Workflow

- Run Score-P instrumented target application to produce runtime summary ❶
  - Provides initial insight into the application's run-time behavior ❷
  - Allows optimizing the configuration for subsequent measurements ❸
    (e.g., filtering of uncritical code regions, estimation of trace buffer requirements, etc.)
- Generate targeted event traces of critical code regions for closer investigation of concurrent behavior ❹
  - Automatic event trace analysis at the end of measurement searching for inefficiency patterns, wait states, ❺
    and the critical path (using a parallel analysis tool to achieve scalability, executed as part of the same batch job)
- Examine trace analysis results using an intuitive graphical user interface (Cube) ❻

Scalable Automatic Wait-state & Root-cause Analysis

- Replay-based trace analysis searches for wait states
- Attribution of short-term and long-term costs to identify delays as root causes of wait states
- Classification of wait states as propagating or terminal to assess inter-wait state influences
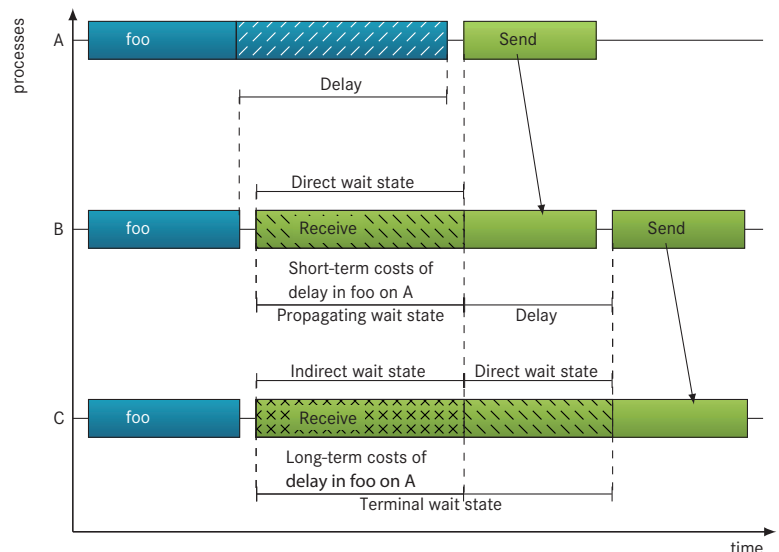
Scalable Critical-path Analysis

- Determines the critical path of the application in a scalable fashion, to help identify
  - Program activities for which optimization will prove worthwhile
  - Parallelization bottlenecks such as load imbalance and serial execution