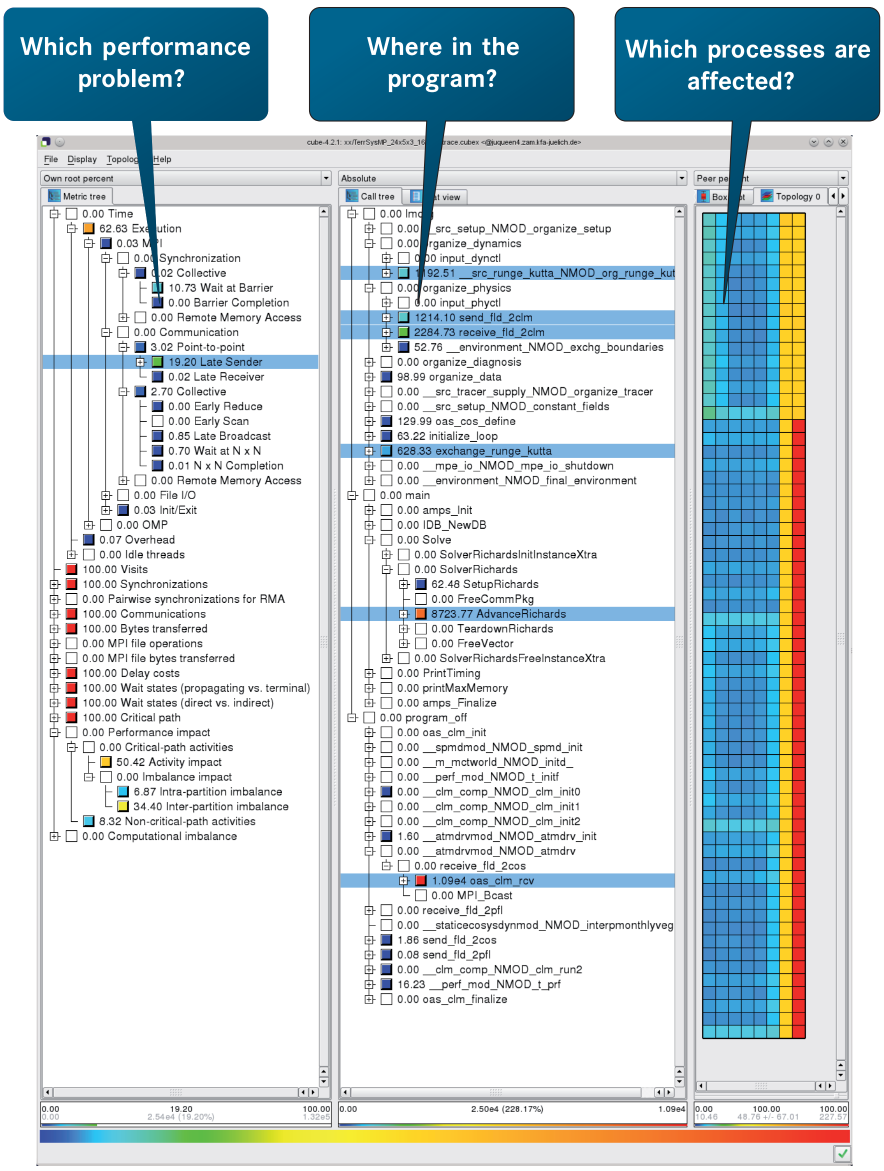
Scalasca Trace Tools

Scalable performance analysis of large-scale parallel applications

Performance Analysis
Tuning
Optimized Application
Measurement
scalasca

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



```
cube-4.2.1: xx/TerrSysMP_24x5x3_16   trace.cubex <@juqueen4.zam.kfa-juelich.de>

0.00 Time
  62.63 Exe ution
    0.03 M I
      0.00 Synchronization
        0.02 Collective
          10.73 Wait at Barrier
          0.00 Barrier Completion
        0.00 Remote Memory Access
      0.00 Communication
        3.02 Point-to-point
          19.20 Late Sender
          0.02 Late Receiver
        2.70 Collective
          0.00 Early Reduce
          0.00 Early Scan
          0.85 Late Broadcast
          0.70 Wait at N x N
          0.01 N x N Completion
        0.00 Remote Memory Access
      0.00 File I/O
      0.03 Init/Exit
    0.00 OMP
  0.07 Overhead
  0.00 Idle threads
100.00 Visits
100.00 Synchronizations
0.00 Pairwise synchronizations for RMA
100.00 Communications
100.00 Bytes transferred
0.00 MPI file operations
0.00 MPI file bytes transferred
100.00 Delay costs
100.00 Wait states (propagating vs. terminal)
100.00 Wait states (direct vs. indirect)
100.00 Critical path
0.00 Performance impact
  0.00 Critical-path activities
    50.42 Activity impact
    0.00 Imbalance impact
      6.87 Intra-partition imbalance
      34.40 Inter-partition imbalance
  8.32 Non-critical-path activities
0.00 Computational imbalance

0.00      19.20                 100.00
0.00      2.54e4 (19.20%)       1.32e5
```

```
0.00 Imc
  0.00 _src_setup_NMOD_organize_setup
  0.00 rganize_dynamics
    0 00 input_dynctl
    1 92.51 __src_runge_kutta_NMOD_org_runge_kut
  0.00 organize_physics
    0.00 input_phyctl
    1214.10 send_fld_2clm
    2284.73 receive_fld_2clm
    52.76 __environment_NMOD_exchg_boundaries
  0.00 organize_diagnosis
  98.99 organize_data
  0.00 __src_tracer_supply_NMOD_organize_tracer
  0.00 __src_setup_NMOD_constant_fields
  129.99 oas_cos_define
  63.22 initialize_loop
  628.33 exchange_runge_kutta
  0.00 __mpe_io_NMOD_mpe_io_shutdown
  0.00 __environment_NMOD_final_environment
0.00 main
  0.00 amps_Init
  0.00 IDB_NewDB
  0.00 Solve
    0.00 SolverRichardsInitInstanceXtra
    0.00 SolverRichards
      62.48 SetupRichards
      0.00 FreeCommPkg
      8723.77 AdvanceRichards
      0.00 TeardownRichards
      0.00 FreeVector
    0.00 SolverRichardsFreeInstanceXtra
  0.00 PrintTiming
  0.00 printMaxMemory
  0.00 amps_Finalize
0.00 program_off
  0.00 oas_clm_init
  0.00 __spmdmod_NMOD_spmd_init
  0.00 __m_mctworld_NMOD_initd_
  0.00 __perf_mod_NMOD_t_initf
  0.00 __clm_comp_NMOD_clm_init0
  0.00 __clm_comp_NMOD_clm_init1
  0.00 __clm_comp_NMOD_clm_init2
  1.60 __atmdrvmod_NMOD_atmdrv_init
  0.00 __atmdrvmod_NMOD_atmdrv
    0.00 receive_fld_2cos
      1.09e4 oas_clm_rcv
      0.00 MPI_Bcast
  0.00 receive_fld_2pfl
  0.00 __staticecosysdynmod_NMOD_interpmonthlyveg
  1.86 send_fld_2cos
  0.08 send_fld_2pfl
  0.00 __clm_comp_NMOD_clm_run2
  16.23 __perf_mod_NMOD_t_prf
  0.00 oas_clm_finalize

0.00      2.50e4 (228.17%)      1.09e4
```

```
0.00      100.00                100.00
10.46     48.76 +/- 67.01       227.57
```

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

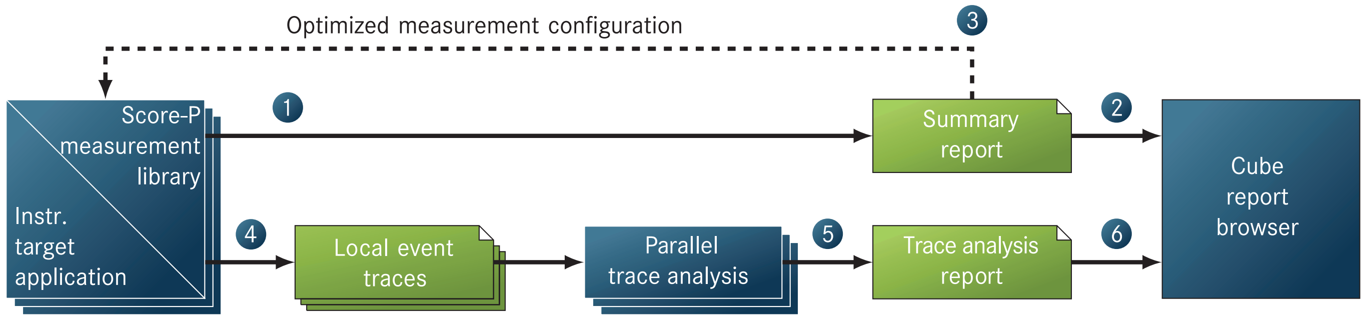
Supported Platforms

- Cray XT/XE/XK/XC
- IBM Blue Gene
- IBM SP & Blade clusters
- Linux-based clusters (x86, Power, ARM)
- Tianhe-1A & 2
- SGI Altix (incl. ICE + UV)
- Fujitsu FX / K computer
- Intel Xeon Phi (native mode only)

www.scalasca.org

JÜLICH
FORSCHUNGSZENTRUM

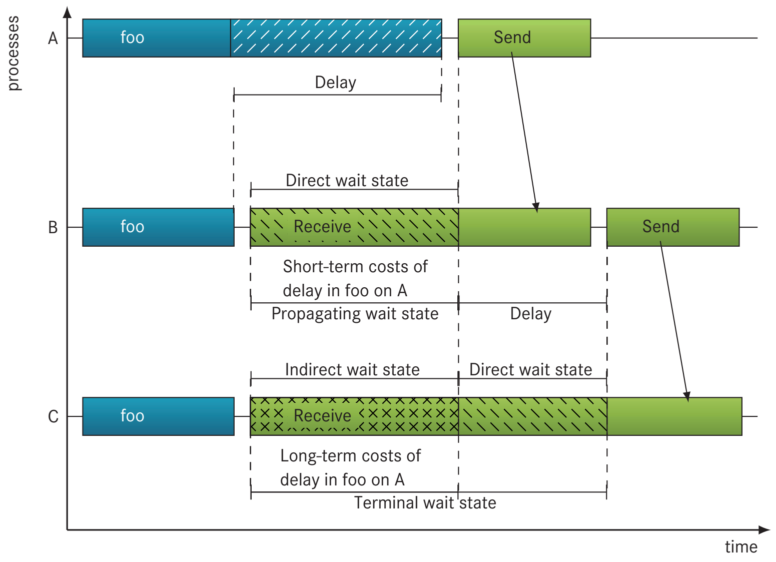
Scalasca Measurement & Analysis Workflow



- Run Score-P instrumented target application to produce runtime summary 1
  - Provides initial insight into the application's run-time behavior 2
  - Allows optimizing the configuration for subsequent measurements 3 (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 4
  - Automatic event trace analysis at the end of measurement searching for inefficiency patterns, wait states, 5 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) 6

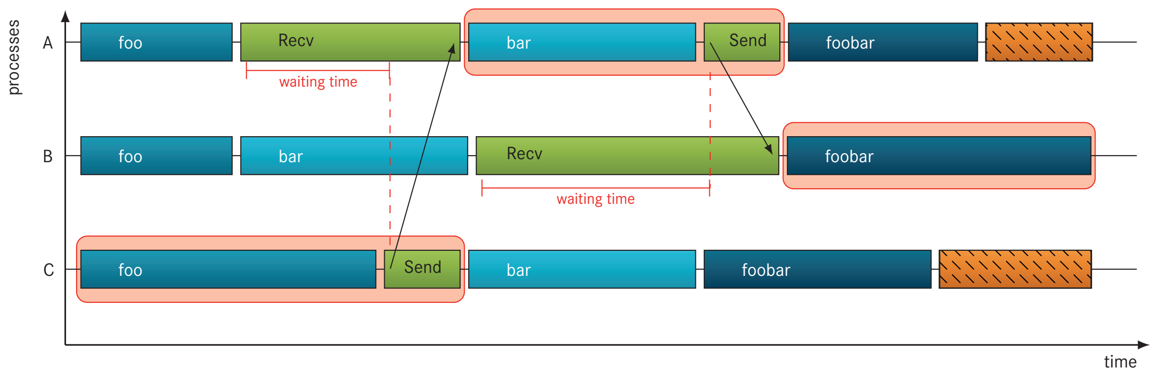
Scalable Automatic Wait-state & Root-cause Analysis

- 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


processes
A
B
C
foo
Recv
waiting time
bar
Send
foobar
foo
bar
Recv
waiting time
foobar
foo
Send
bar
foobar
time