International HPC Summer School 2021: Performance analysis and optimization

Vampir, TAU, Extra-P, Darshan

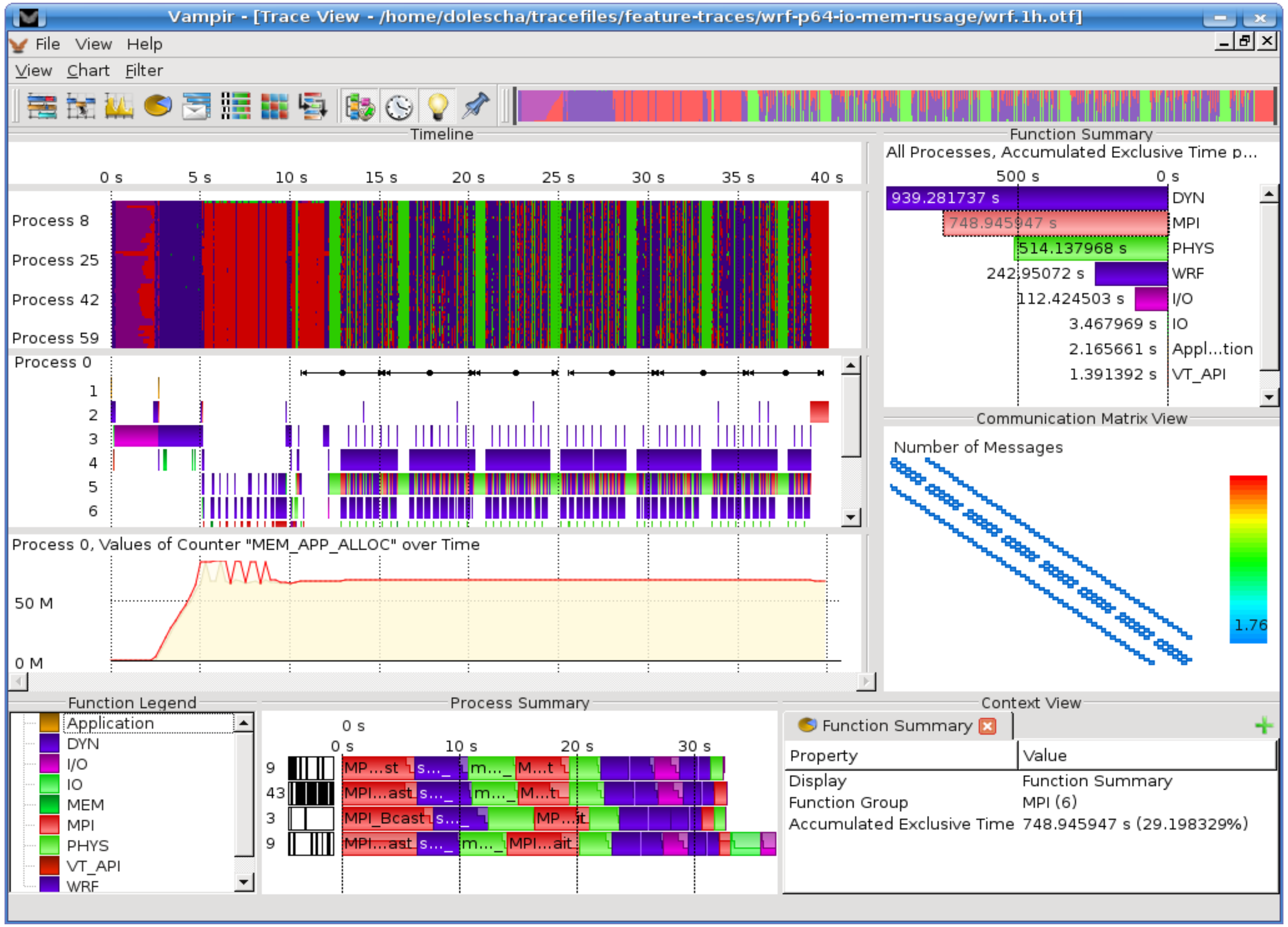
VI-HPS Team
Ilya Zhukov – Jülich Supercomputing Centre

Vampir Event Trace Visualizer

- Offline trace visualization for Score-P's OTF2 trace files
- Visualization of MPI, OpenMP and application events:
  - All diagrams highly customizable (through context menus)
  - Large variety of displays for ANY part of the trace
- http://www.vampir.eu

- Advantage:
  - Detailed view of dynamic application behavior
- Disadvantage:
  - Requires event traces (huge amount of data)
  - Completely manual analysis

Vampir Displays



Property	Value
Display	Function Summary
Function Group	MPI (6)
Accumulated Exclusive Time	748.945947 s (29.198329%)

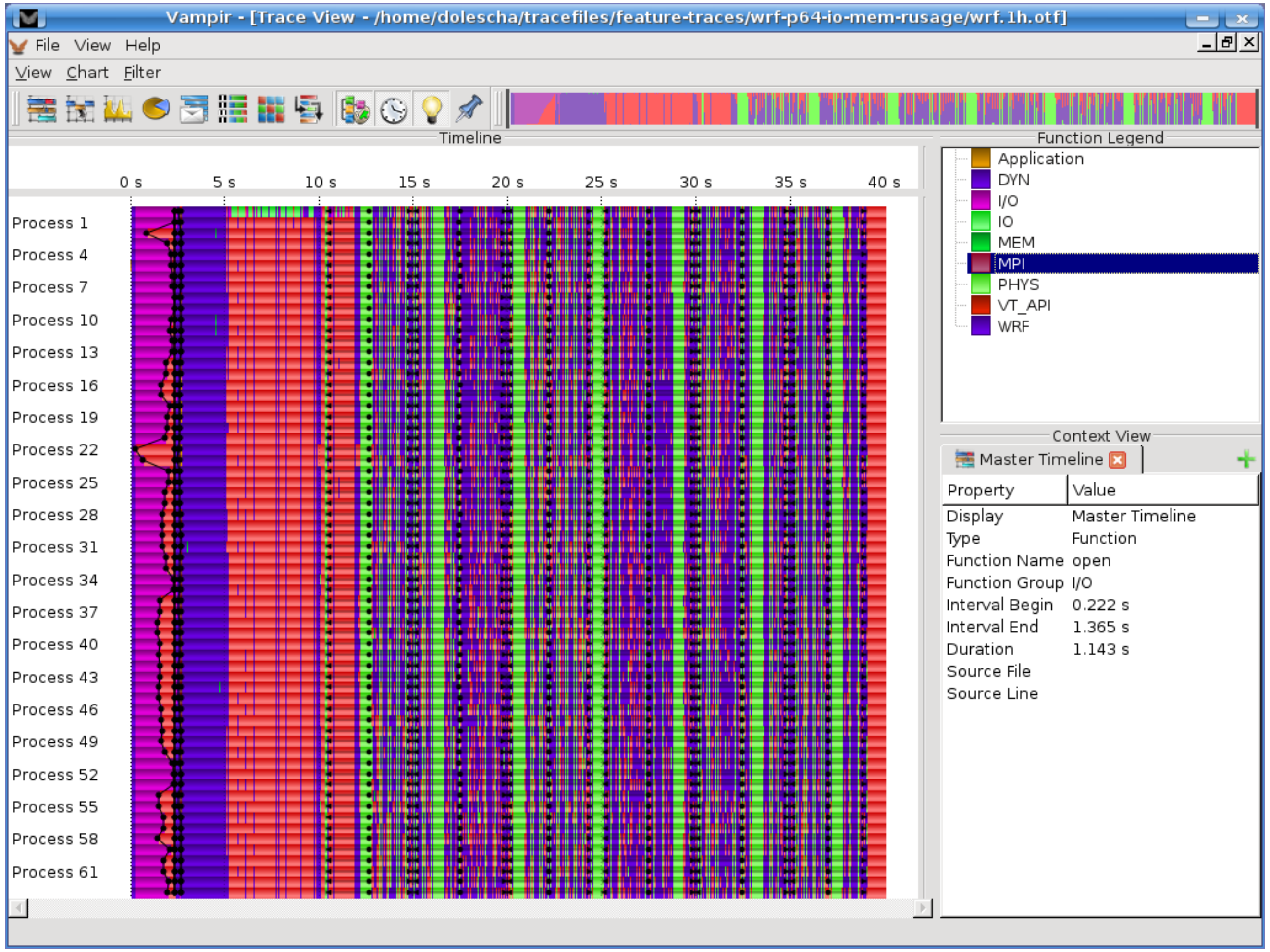
Vampir: Timeline Diagram

- Functions organized into groups

- Coloring by group

- Message lines can be colored by tag or size

- Information about states, messages, collective and I/O operations available through clicking on the representation

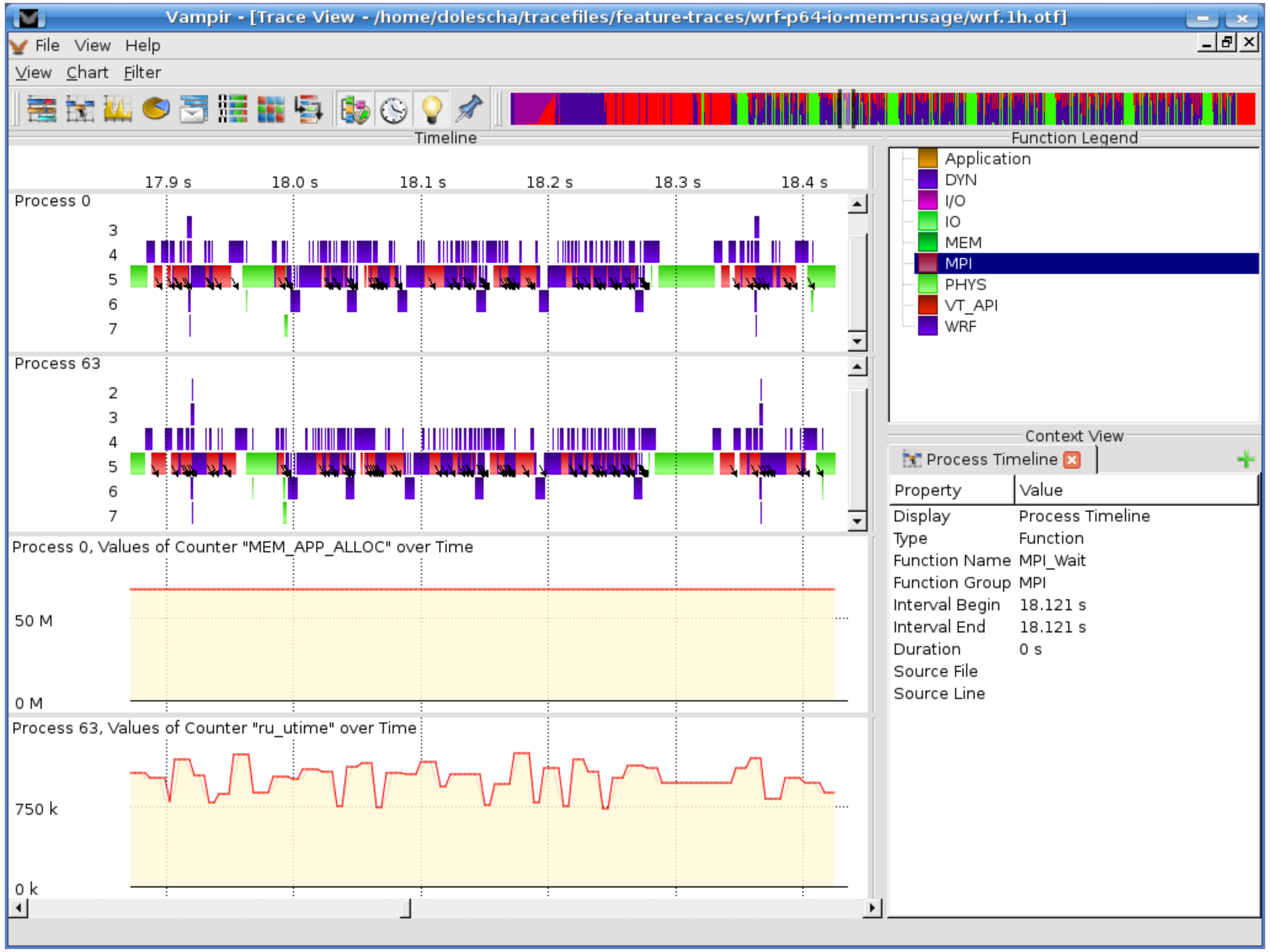


Property	Value
Display	Master Timeline
Type	Function
Function Name	open
Function Group	I/O
Interval Begin	0.222 s
Interval End	1.365 s
Duration	1.143 s
Source File	
Source Line	

Vampir: Process and Counter Timelines

- Process timeline
  show call stack nesting

- Counter timelines
  for hardware and
  software counters


Vampir - [Trace View - /home/dolescha/tracefiles/feature-traces/wrf-p64-io-mem-rusage/wrf.1h.otf]
File View Help
View Chart Filter
Timeline
17.9 s
18.0 s
18.1 s
18.2 s
18.3 s
18.4 s
Process 0
3
4
5
6
7
Process 63
2
3
4
5
6
7
Process 0, Values of Counter "MEM_APP_ALLOC" over Time
50 M
0 M
Process 63, Values of Counter "ru_utime" over Time
750 k
0 k
Function Legend
Application
DYN
I/O
IO
MEM
MPI
PHYS
VT_API
WRF
Context View
Process Timeline

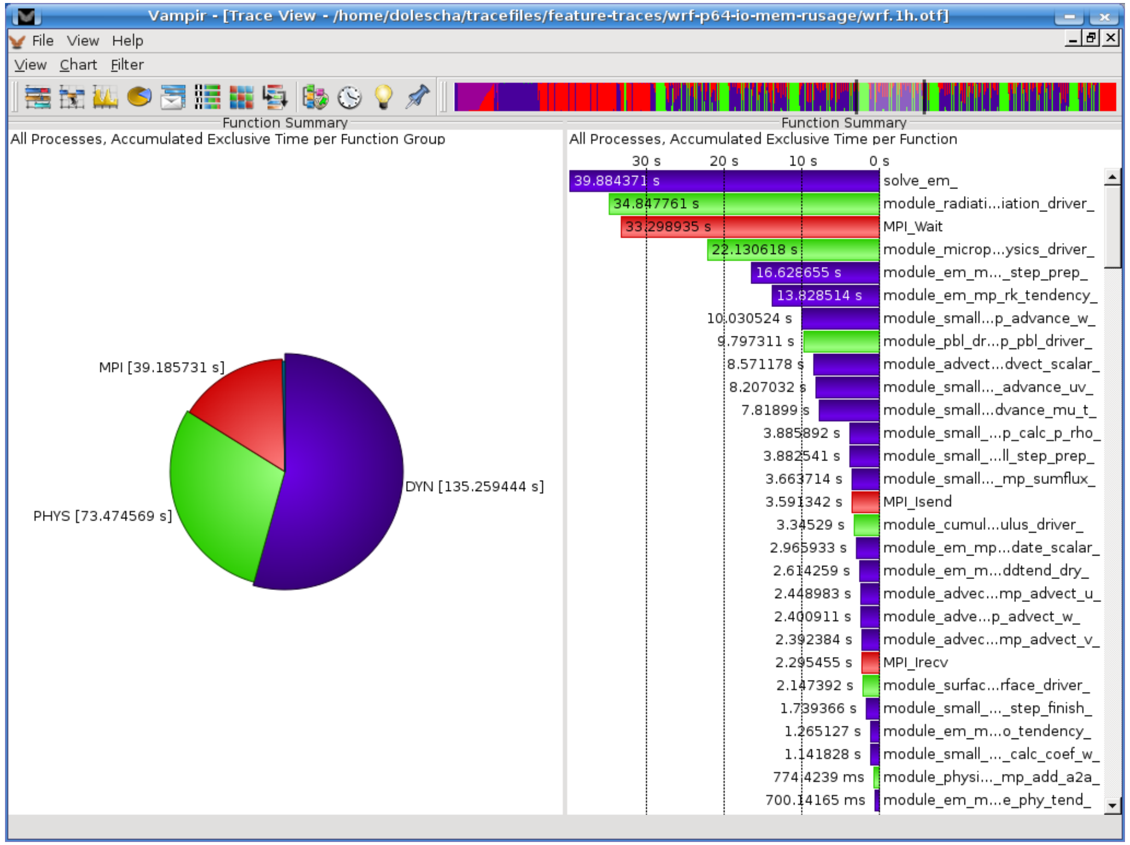
Property	Value
Display	Process Timeline
Type	Function
Function Name	MPI_Wait
Function Group	MPI
Interval Begin	18.121 s
Interval End	18.121 s
Duration	0 s
Source File	
Source Line	

Vampir: Execution Statistics

- Aggregated profiling information: execution time, number of calls, inclusive/exclusive

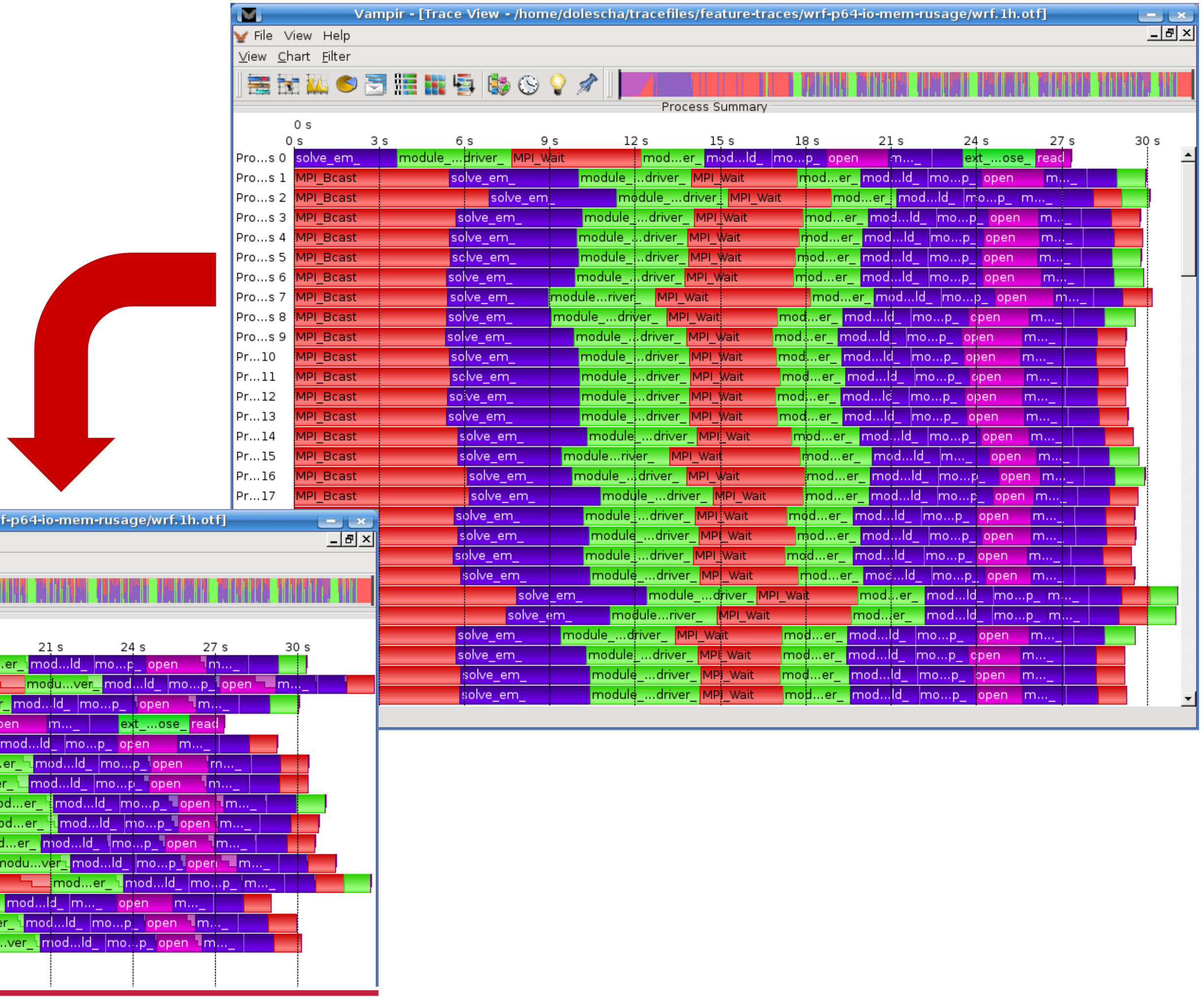
- Available for all / any group (activity) or all routines (symbols)

- Available for any part of the trace
  ⇨ selectable through time line diagram


Vampir - [Trace View - /home/dolescha/tracefiles/feature-traces/wrf-p64-io-mem-rusage/wrf.1h.otf]
File View Help
View Chart Filter
Function Summary
All Processes, Accumulated Exclusive Time per Function Group
MPI [39.185731 s]
DYN [135.259444 s]
PHYS [73.474569 s]
Function Summary
All Processes, Accumulated Exclusive Time per Function
30 s
20 s
10 s
0 s
39.884371 s solve_em_
34.847761 s module_radiati...iation_driver_
33.298935 s MPI_Wait
22.130618 s module_microp...ysics_driver_
16.628655 s module_em_m..._step_prep_
13.828514 s module_em_mp_rk_tendency_
10.030524 s module_small...p_advance_w_
9.797311 s module_pbl_dr...p_pbl_driver_
8.571178 s module_advect...dvect_scalar_
8.207032 s module_small..._advance_uv_
7.81899 s module_small...dvance_mu_t_
3.885892 s module_small_...p_calc_p_rho_
3.882541 s module_small_...ll_step_prep_
3.663714 s module_small..._mp_sumflux_
3.591342 s MPI_Isend
3.34529 s module_cumul...ulus_driver_
2.965933 s module_em_mp...date_scalar_
2.614259 s module_em_m...ddtend_dry_
2.448983 s module_advec...mp_advect_u_
2.400911 s module_adve...p_advect_w_
2.392384 s module_advec...mp_advect_v_
2.295455 s MPI_Irecv
2.147392 s module_surfac...rface_driver_
1.739366 s module_small_..._step_finish_
1.265127 s module_em_m...o_tendency_
1.141828 s module_small_..._calc_coef_w_
774.4239 ms module_physi..._mp_add_a2a_
700.14165 ms module_em_m...e_phy_tend_

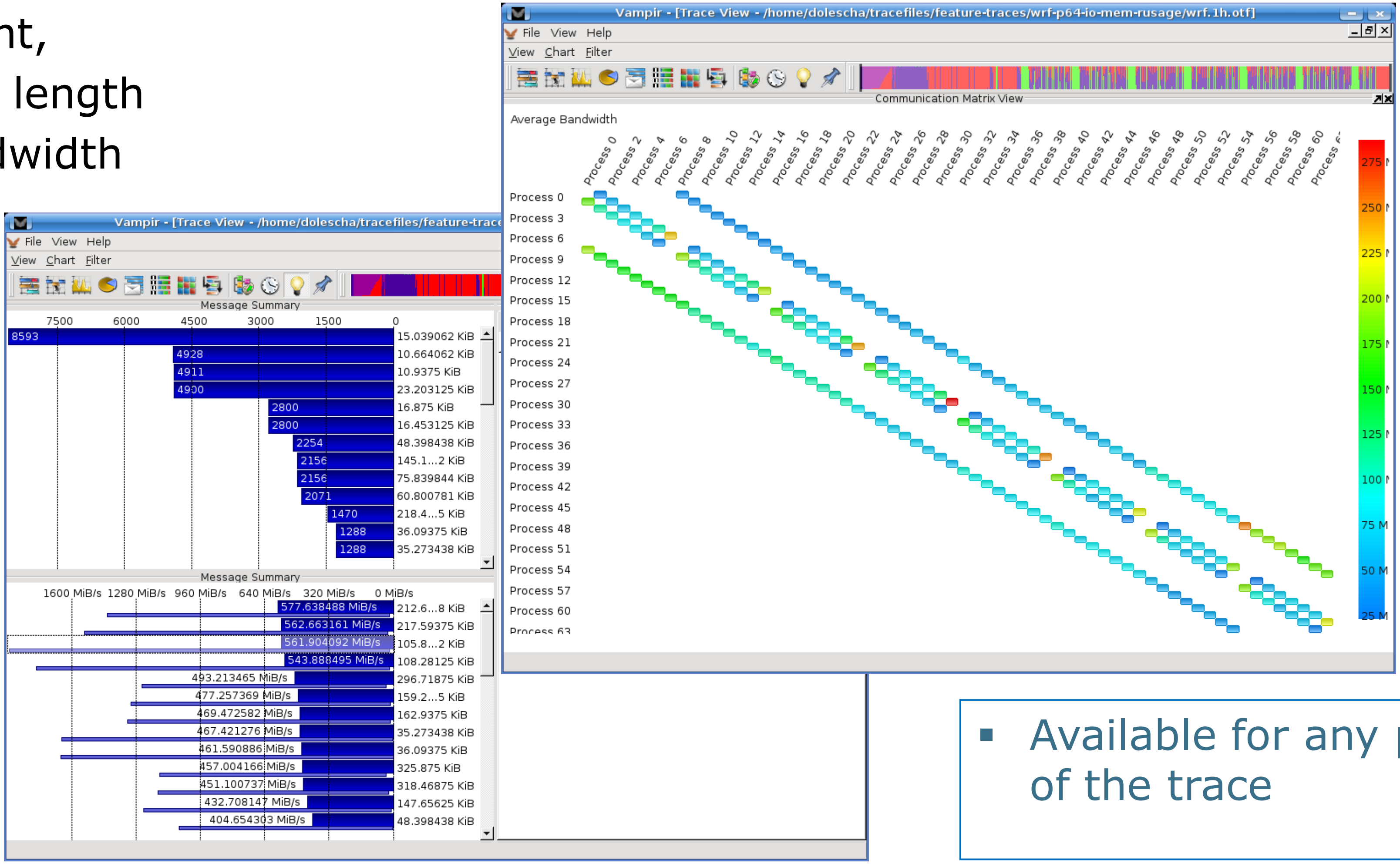
Vampir: Process Summary

- Execution statistics over all processes for comparison
- Clustering mode available for large process counts



Vampir: Communication Statistics

- Byte and message count, min/max/avg message length and min/max/avg bandwidth for each process pair
- Message length statistics

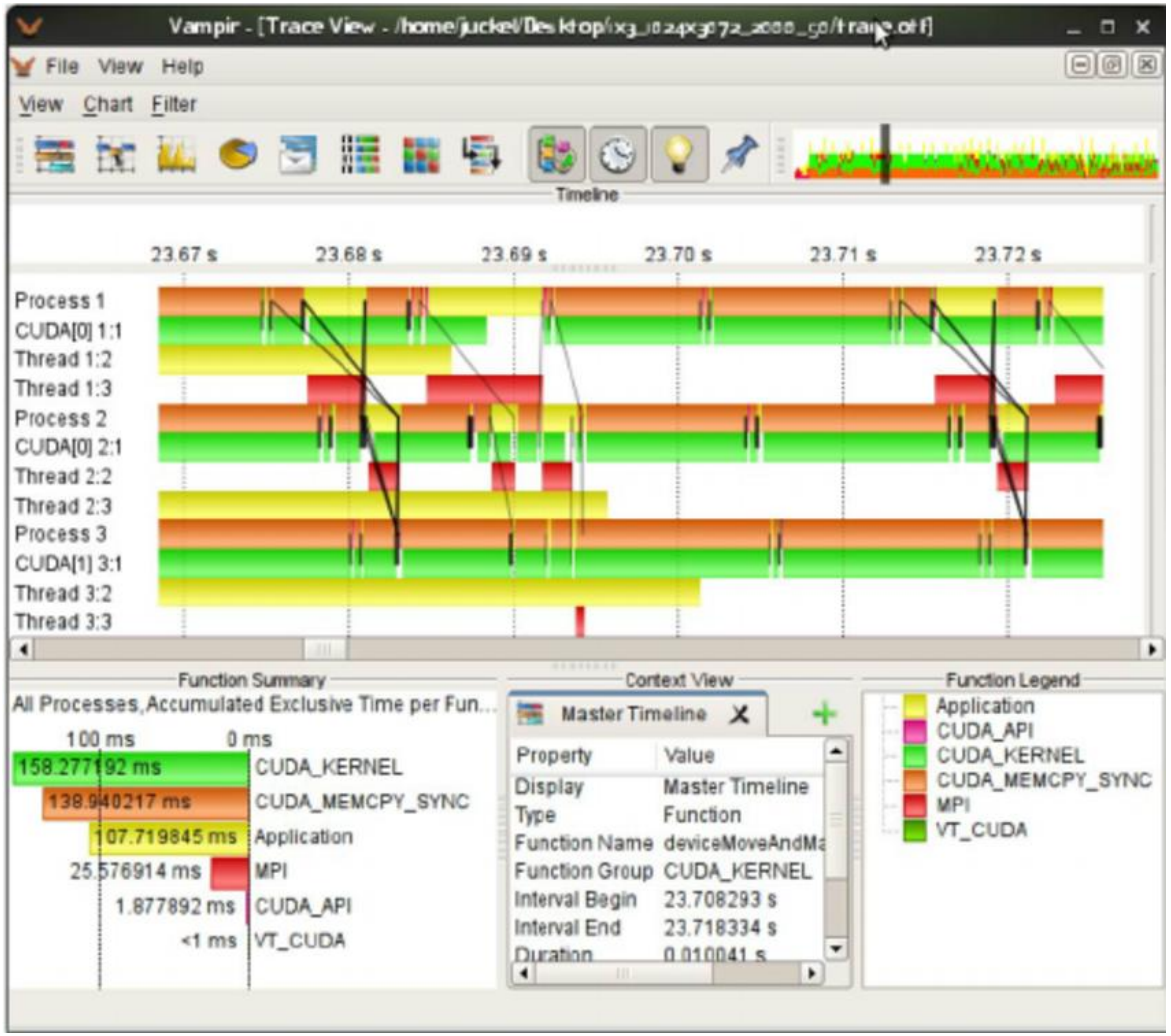


- Available for any part of the trace

Vampir: CUDA Example

- Detailed information on kernel execution and memory transfers

- All statistics and displays also available for CUDA events



Property	Value
Display	Master Timeline
Type	Function
Function Name	deviceMoveAndMa
Function Group	CUDA_KERNEL
Interval Begin	23.708293 s
Interval End	23.718334 s
Duration	0.010041 s

TAU

- Very portable tool set for instrumentation, measurement and analysis of parallel multi-threaded applications
- http://tau.uoregon.edu/

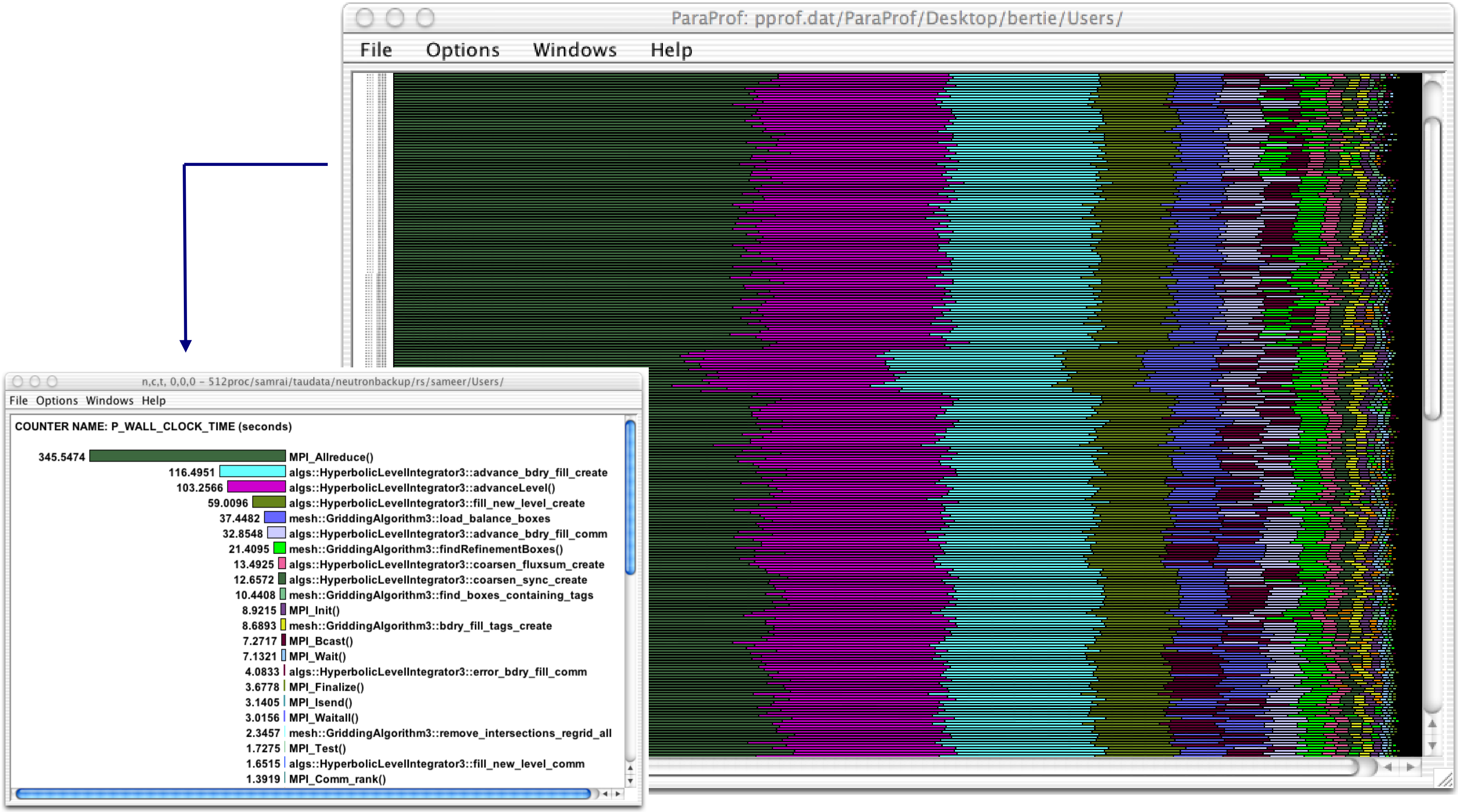
- Supports
  - Various profiling modes and tracing
  - Various forms of code instrumentation
  - C, C++, Fortran, Java, Python
  - MPI, multi-threading (OpenMP, Pthreads, …)
  - Accelerators

Tuning and Analysis Utilities

TAU: Instrumentation

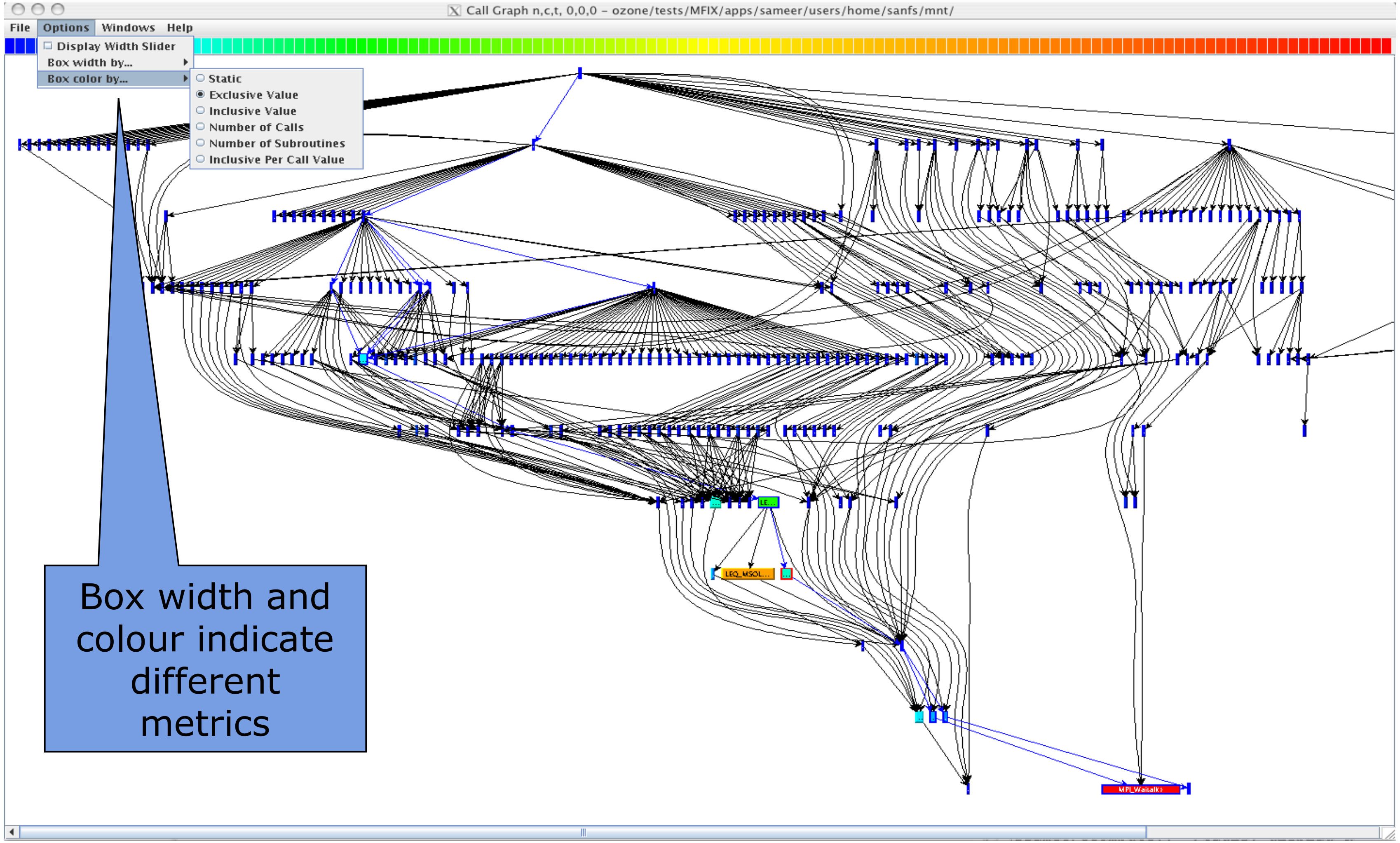
- Flexible instrumentation mechanisms at multiple levels
  - Source code
    - manual
    - automatic
      - C, C++, F77/90/95 (Program Database Toolkit (PDT))
      - OpenMP (directive rewriting with Opari)
  - Object code
    - pre-instrumented libraries (e.g., MPI using PMPI)
    - statically-linked and dynamically-loaded (e.g., Python)
  - Executable code
    - dynamic instrumentation (pre-execution) (DynInst)
    - virtual machine instrumentation (e.g., Java using JVMPI)
- Support for performance mapping
- Support for object-oriented and generic programming

TAU: Basic Profile View


ParaProf: pprof.dat/ParaProf/Desktop/bertie/Users/
File Options Windows Help
n,c,t, 0,0,0 – 512proc/samrai/taudata/neutronbackup/rs/sameer/Users/
File Options Windows Help

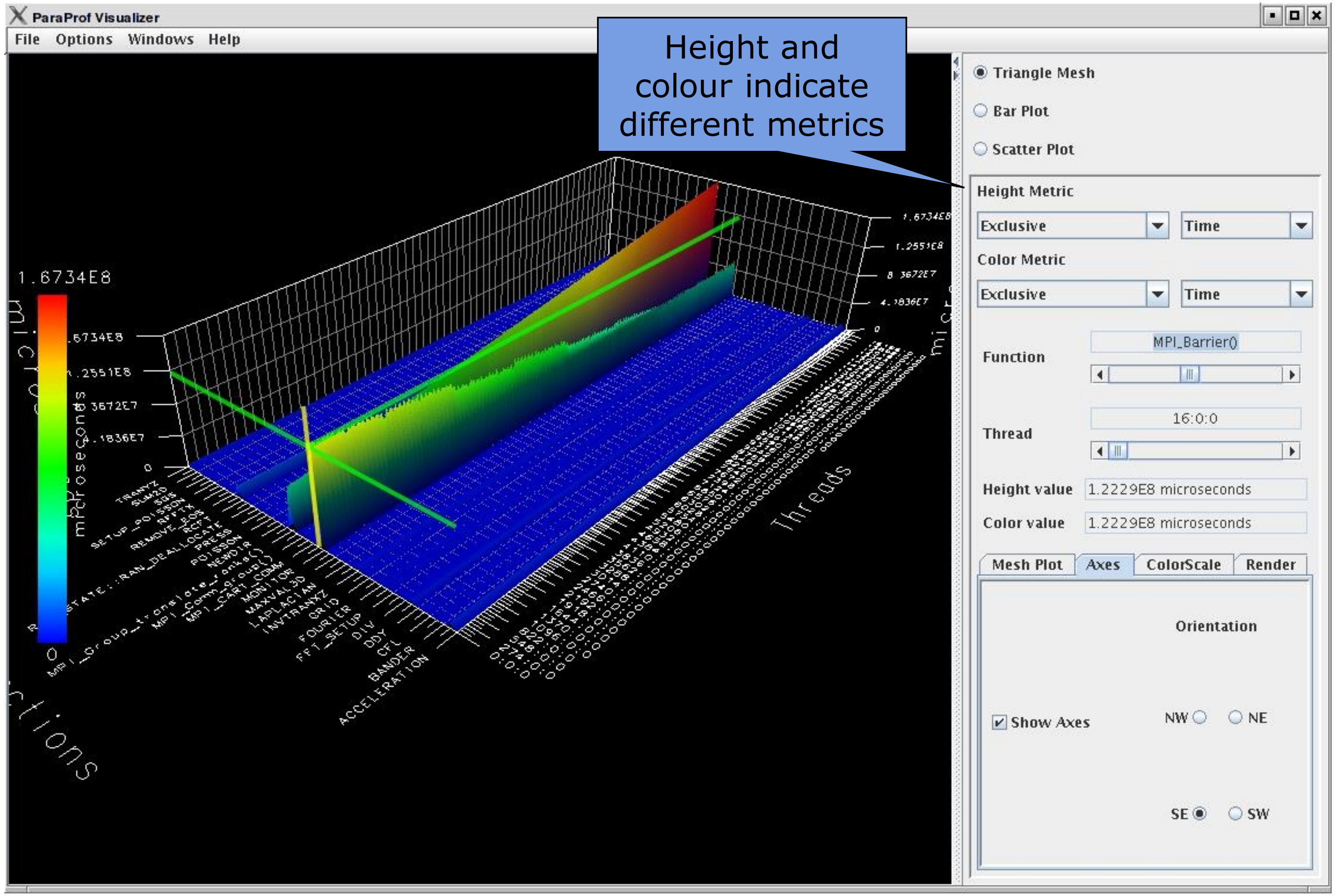
```
COUNTER NAME: P_WALL_CLOCK_TIME (seconds)

345.5474  MPI_Allreduce()
116.4951  algs::HyperbolicLevelIntegrator3::advance_bdry_fill_create
103.2566  algs::HyperbolicLevelIntegrator3::advanceLevel()
 59.0096  algs::HyperbolicLevelIntegrator3::fill_new_level_create
 37.4482  mesh::GriddingAlgorithm3::load_balance_boxes
 32.8548  algs::HyperbolicLevelIntegrator3::advance_bdry_fill_comm
 21.4095  mesh::GriddingAlgorithm3::findRefinementBoxes()
 13.4925  algs::HyperbolicLevelIntegrator3::coarsen_fluxsum_create
 12.6572  algs::HyperbolicLevelIntegrator3::coarsen_sync_create
 10.4408  mesh::GriddingAlgorithm3::find_boxes_containing_tags
  8.9215  MPI_Init()
  8.6893  mesh::GriddingAlgorithm3::bdry_fill_tags_create
  7.2717  MPI_Bcast()
  7.1321  MPI_Wait()
  4.0833  algs::HyperbolicLevelIntegrator3::error_bdry_fill_comm
  3.6778  MPI_Finalize()
  3.1405  MPI_Isend()
  3.0156  MPI_Waitall()
  2.3457  mesh::GriddingAlgorithm3::remove_intersections_regrid_all
  1.7275  MPI_Test()
  1.6515  algs::HyperbolicLevelIntegrator3::fill_new_level_comm
  1.3919  MPI_Comm_rank()
```

TAU: Callgraph Profile View



Box width and colour indicate different metrics

TAU: 3D Profile View



Height and colour indicate different metrics

● Triangle Mesh

○ Bar Plot

○ Scatter Plot

Height Metric: Exclusive, Time

Color Metric: Exclusive, Time

Function: MPI_Barrier()

Thread: 16:0:0

Height value: 1.2229E8 microseconds

Color value: 1.2229E8 microseconds

☑ Show Axes

Orientation: NW ○ NE ○ SE ● SW ○

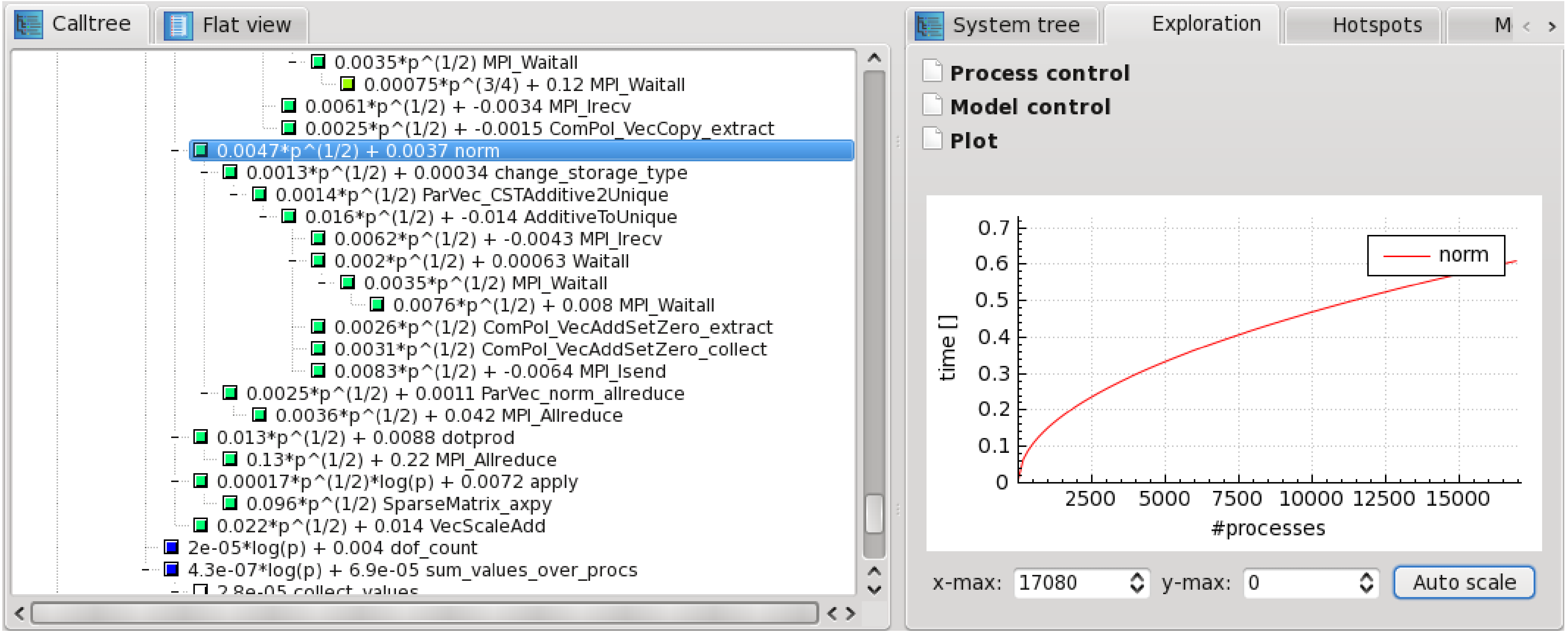
Extra-P

- **Goal**: identification of parts of the program which scaling behavior is unintentionally poor (much worse than expected) by means of automatic performance-modeling

- Supports Linux (x86x86_64/IA64/PPC/Power), Mac OS X (x86_64)
- Accepts input files in the Cube format and processes them into a condensed Cube format containing functions for each metric and call path

- http://www.scalasca.org
- Open Source: BSD 3-Clause License

Interactive exploration of performance models in Extra-P



```
              0.0035*p^(1/2) MPI_Waitall
                0.00075*p^(3/4) + 0.12 MPI_Waitall
            0.0061*p^(1/2) + -0.0034 MPI_Irecv
            0.0025*p^(1/2) + -0.0015 ComPol_VecCopy_extract
        0.0047*p^(1/2) + 0.0037 norm
          0.0013*p^(1/2) + 0.00034 change_storage_type
            0.0014*p^(1/2) ParVec_CSTAdditive2Unique
              0.016*p^(1/2) + -0.014 AdditiveToUnique
                0.0062*p^(1/2) + -0.0043 MPI_Irecv
                0.002*p^(1/2) + 0.00063 Waitall
                  0.0035*p^(1/2) MPI_Waitall
                    0.0076*p^(1/2) + 0.008 MPI_Waitall
                0.0026*p^(1/2) ComPol_VecAddSetZero_extract
                0.0031*p^(1/2) ComPol_VecAddSetZero_collect
                0.0083*p^(1/2) + -0.0064 MPI_Isend
          0.0025*p^(1/2) + 0.0011 ParVec_norm_allreduce
            0.0036*p^(1/2) + 0.042 MPI_Allreduce
        0.013*p^(1/2) + 0.0088 dotprod
          0.13*p^(1/2) + 0.22 MPI_Allreduce
        0.00017*p^(1/2)*log(p) + 0.0072 apply
          0.096*p^(1/2) SparseMatrix_axpy
        0.022*p^(1/2) + 0.014 VecScaleAdd
      2e-05*log(p) + 0.004 dof_count
      4.3e-07*log(p) + 6.9e-05 sum_values_over_procs
        2.8e-05 collect_values
```

x-max: 17080 y-max: 0

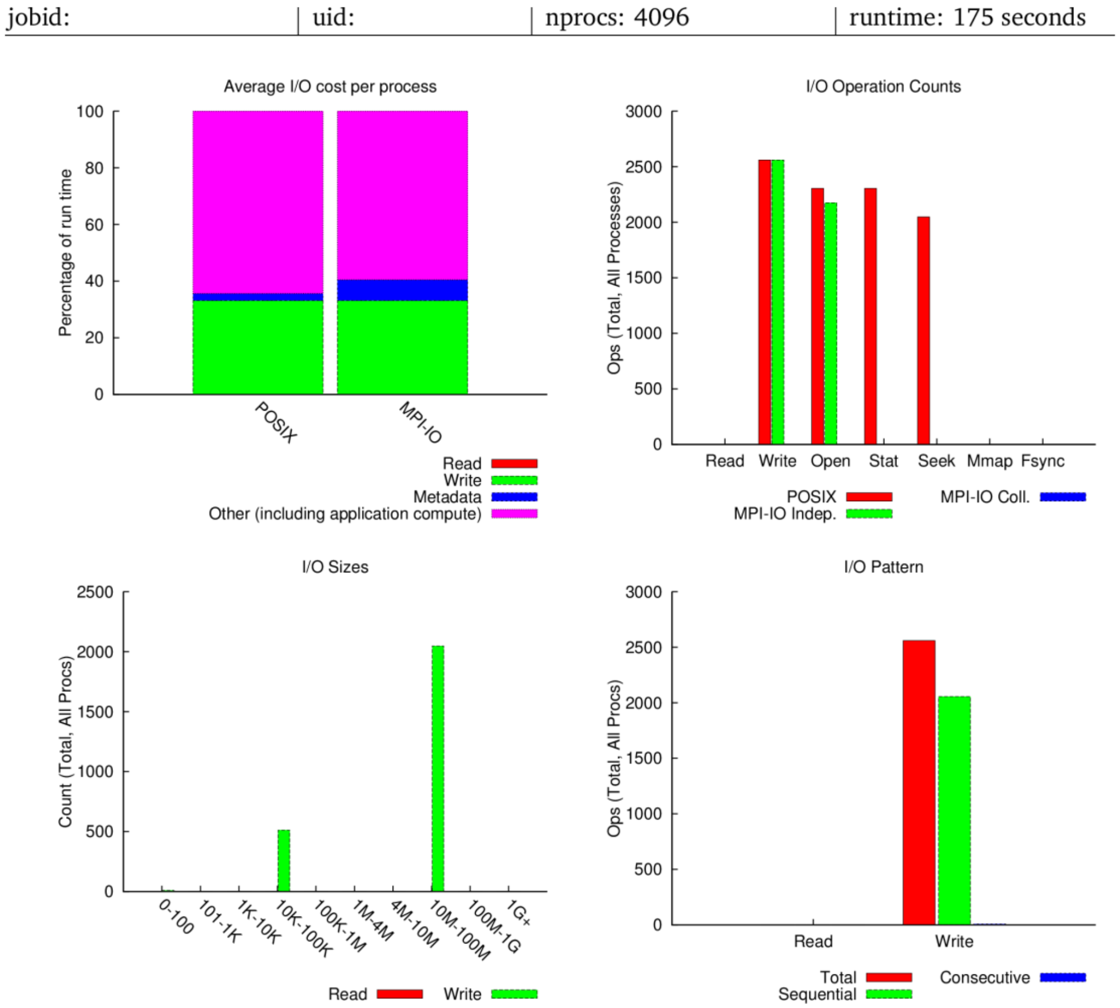
Darshan

- I/O characterization tool logging parallel application file access
- Summary report provides quick overview of performance issues
- Works on unmodified, optimized executables
- Shows counts of file access operations, times for key operations, histograms of accesses, etc.

- Supports POSIX, MPI-IO, HDF5, PnetCDF, …
  - Doesn’t support mpif90 on BlueGene systems (use mpif77)
- Binary log file written at exit post-processed into PDF report

- http://www.mcs.anl.gov/research/projects/darshan/
- Open Source: installed on many HPC systems

Example Darshan report extract



jobid:	uid:	nprocs: 4096	runtime: 175 seconds