# Enabling HPC software productivity with the TAU performance system

Jean-Baptiste BESNARD
<jbbesnard@paratools.fr>

PDC Summer School August 2023, KTH, Stockholm, Sweden.

# Runtime Support

**What can be instrumented using TAU ?**

# TAU's Support for Runtime Systems

*MPI*

- PMPI profiling interface
- MPI_T tools interface using performance and control variables

*Pthread*

- Captures time spent in routines per thread of execution

*OpenMP*

- OMPT tools interface to track salient OpenMP runtime events
- Opari source rewriter
- Preloading wrapper OpenMP runtime library when OMPT is not supported

*OpenACC*

- OpenACC instrumentation API
- Track data transfers between host and device (per-variable)
- Track time spent in kernels

# TAU's Support for Runtime Systems (contd.)

*OpenCL*

- OpenCL profiling interface
- Track timings of kernels

*CUDA*

- Cuda Profiling Tools Interface (CUPTI)
- Track data transfers between host and GPU
- Track access to uniform shared memory between host and GPU

*ROCm*

- Rocprofiler and Roctracer instrumentation interfaces
- Track data transfers and kernel execution between host and GPU

*Kokkos*

- Kokkos profiling API
- Push/pop interface for region, kernel execution interface

*Python*

- Python interpreter instrumentation API
- Tracks Python routine transitions as well as Python to C transitions

# Examples of Multi-Level Instrumentation

*MPI + OpenMP*

- MPI_T + PMPI + OMPT may be used to track MPI and OpenMP

*MPI + CUDA*

- PMPI + CUPTI interfaces

*OpenCL + ROCm*

- Rocprofiler + OpenCL instrumentation interfaces

*Kokkos + OpenMP*

- Kokkos profiling API + OMPT to transparently track events

*Kokkos + pthread + MPI*

- Kokkos + pthread wrapper interposition library + PMPI layer

*Python + CUDA*

- Python + CUPTI + pthread profiling interfaces (e.g., Tensorflow, PyTorch)

*MPI + OpenCL*

- PMPI + OpenCL profiling interfaces

# What does TAU support?

C/C++

Fortran

pthreads

Intel GNU

MPC

CUDA UPC

OpenACC

Intel MIC

LLVM PGI Cray

Linux Windows

BlueGene Fujitsu

NVIDIA Power 8

OpenCL

Python

GPI

Java MPI

OpenMP

Sun

AIX

ARM64

OS X

Insert yours here

# Basic usage of TAU

**How to run ? How to visualize ?**

# Simplifying the use of TAU!

Uninstrumented code:

- `$ make`
- `$ mpirun –np 64 ./a.out`

With TAU using event based sampling (EBS):

- `$ mpirun –np 64 tau_exec –ebs ./lu.B.64`
- `$ paraprof (GUI)`
- `$ pprof –a | more`

NOTE: Source code should be compiled with –g for access to symbol table.

# TAU Execution Command (tau_exec)

Uninstrumented execution

- % mpirun -np 256 ./a.out

Track GPU operations

- % mpirun –np 256 tau_exec –rocm ./a.out
- % mpirun –np 256 tau_exec –cupti ./a.out
- % mpirun –np 256 tau_exec –cupti -um ./a.out (for Unified Memory)
- % mpirun –np 256 tau_exec –opencl ./a.out
- % mpirun –np 256 tau_exec –openacc ./a.out

Track MPI performance

- % mpirun -np 256 tau_exec ./a.out

Track I/O, and MPI performance (MPI enabled by default)

- % mpirun -np 256 tau_exec -io ./a.out

Track OpenMP and MPI execution (using OMPT for Intel v19)

- % export TAU_OMPT_SUPPORT_LEVEL=full;
  % export TAU_OMPT_RESOLVE_ADDRESS_EAGERLY=1
- % mpirun –np 256 tau_exec –T ompt,v5,mpi -ompt ./a.out

Track memory operations

- % export TAU_TRACK_MEMORY_LEAKS=1
- % mpirun –np 256 tau_exec –memory_debug ./a.out (bounds check)

Use event based sampling (compile with –g)

- % mpirun –np 256 tau_exec –ebs ./a.out
- Also -ebs_source=<PAPI_COUNTER> -ebs_period=<overflow_count>
  -ebs_resolution=<file | function | line>

# Types of Performance Profiles

*Flat* profiles
- Metric (e.g., time) spent in an event
- Exclusive/inclusive, # of calls, child calls, …

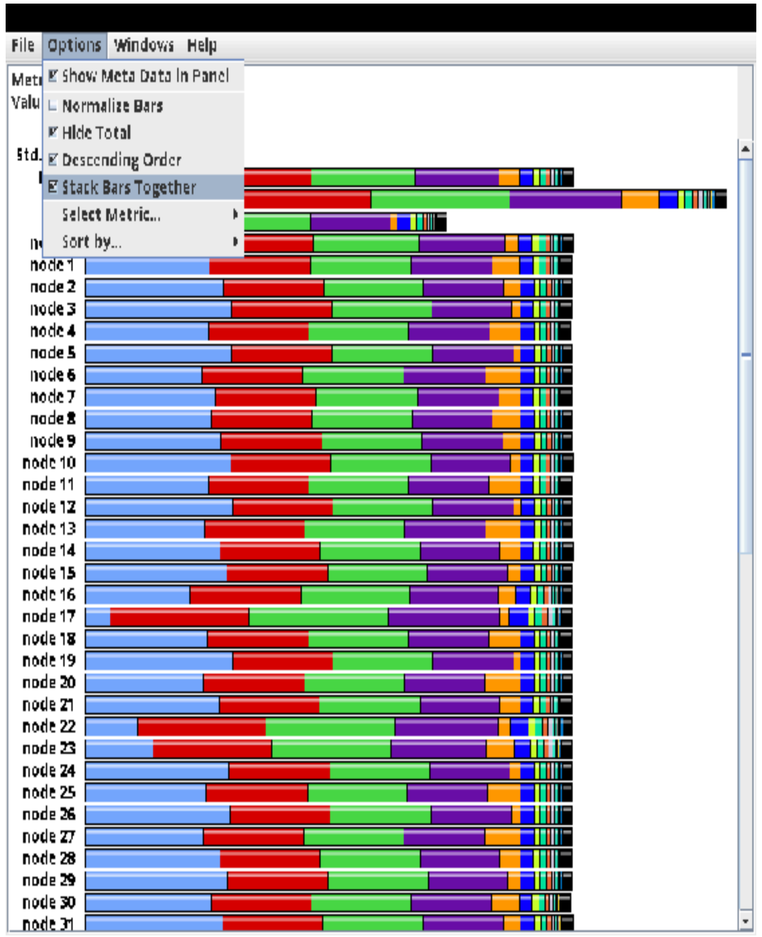
*Callpath* profiles
- Time spent along a calling path (edges in callgraph)
- "*main=> f1 => f2 => MPI_Send*"
- Set the TAU_CALLPATH and TAU_CALLPATH_DEPTH environment variables

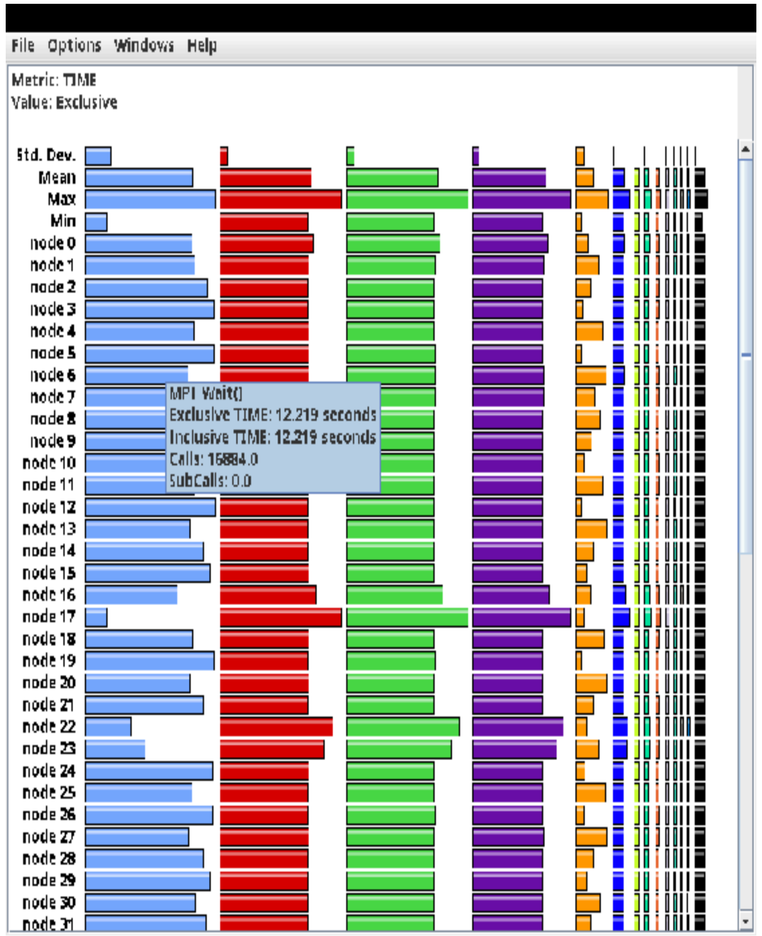
*Callsite* profiles
- Time spent along in an event at a given source location
- Set the TAU_CALLSITE environment variable

*Phase* profiles
- Flat profiles under a phase (nested phases allowed)
- Default "main" phase
- Supports static or dynamic (e.g. per-iteration) phases

Paraprof

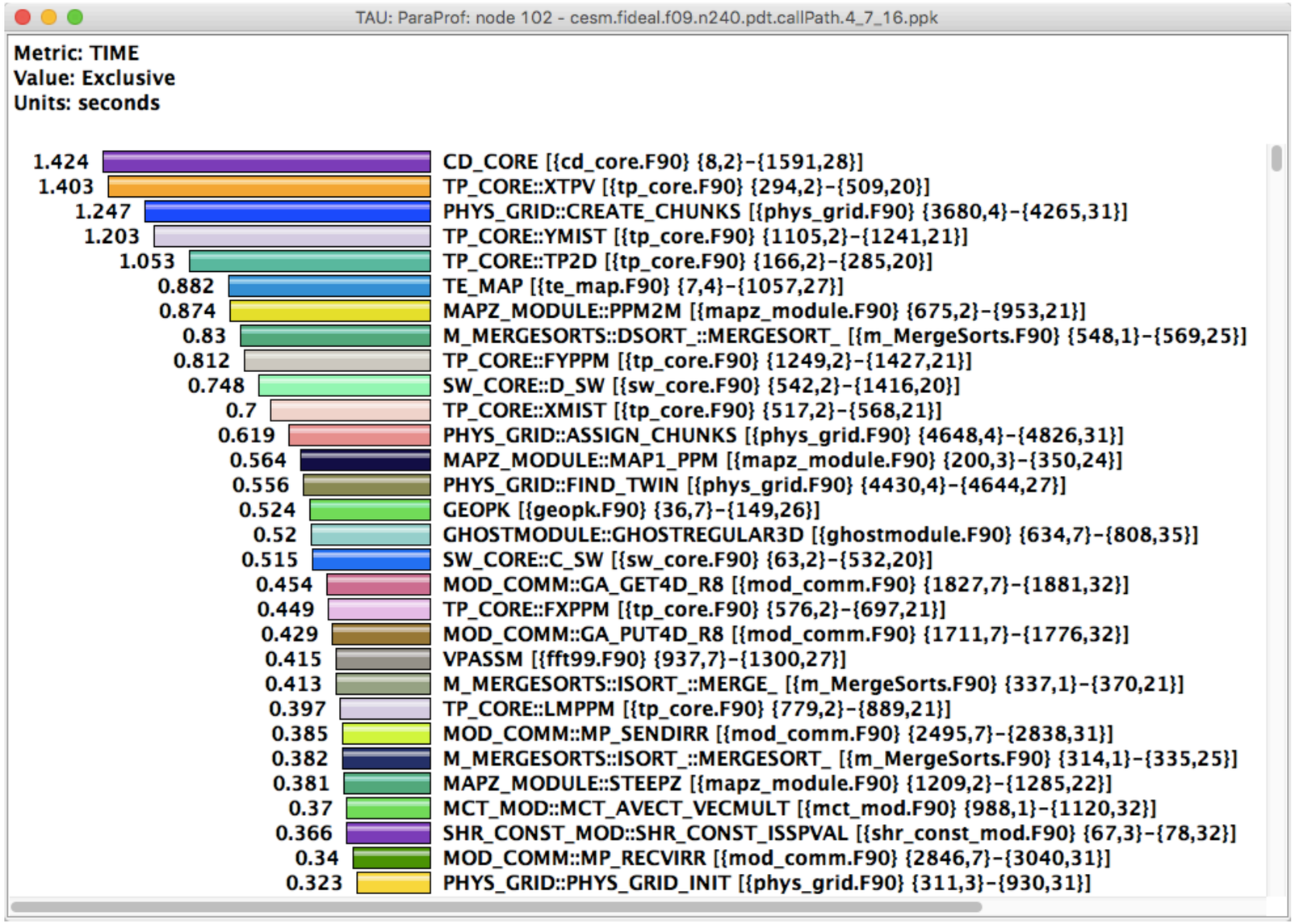
# ParaProf Profile Browser



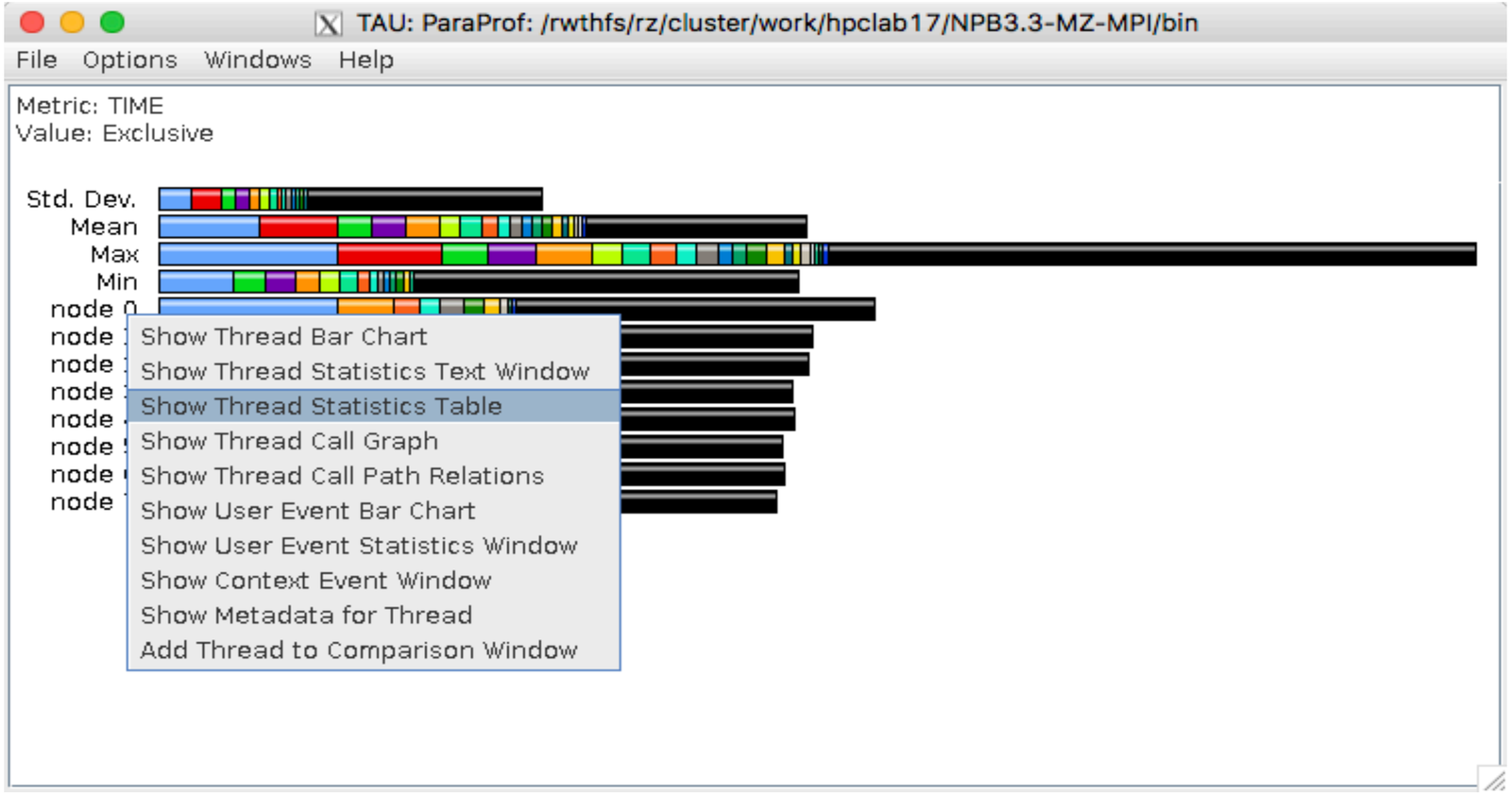




Click “node X” next to see details

# TAU – Flat Profile

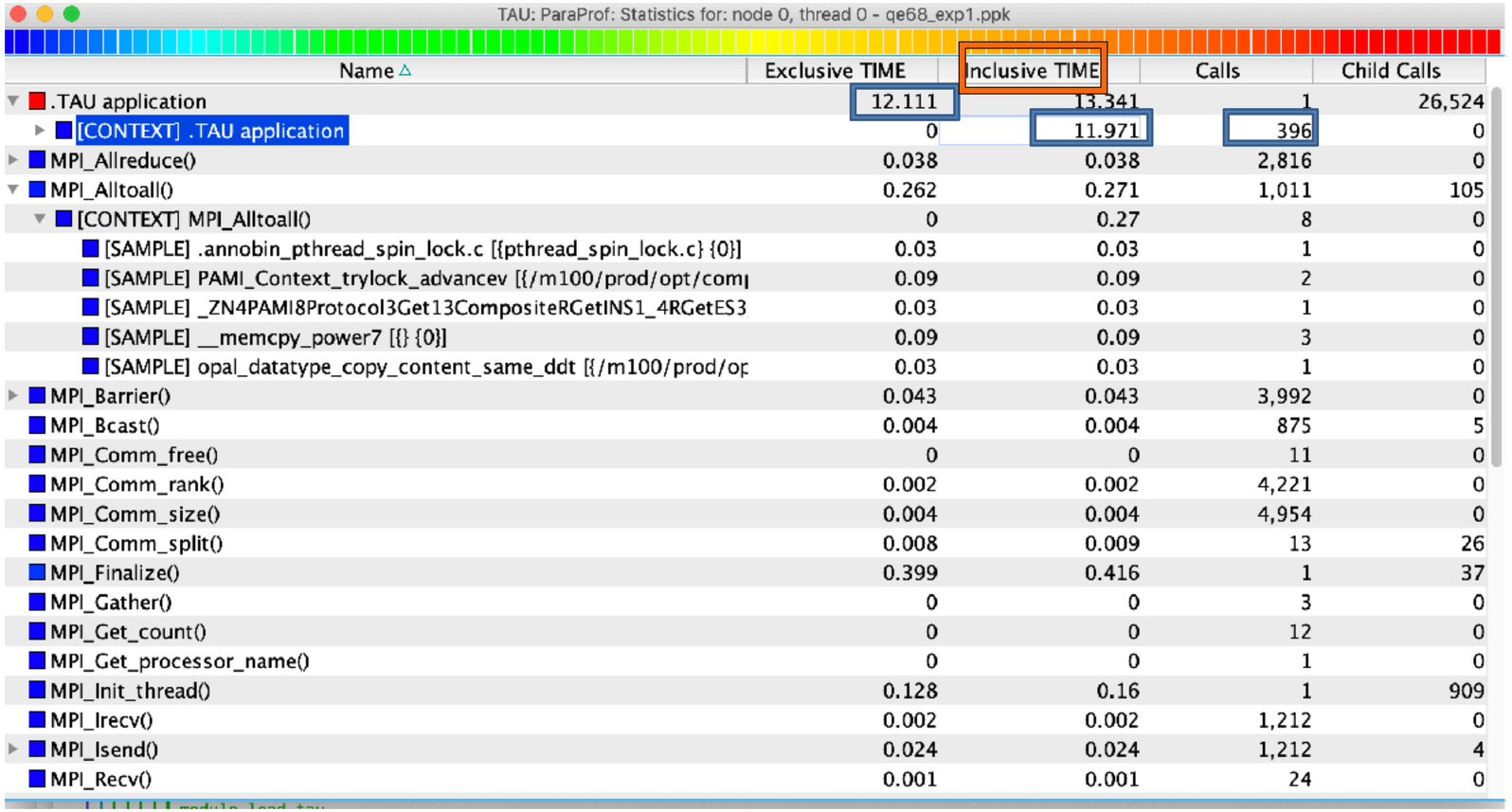


# ParaProf Thread Statistics Table




Right click over “node X” and choose
Show Thread Statistics Table

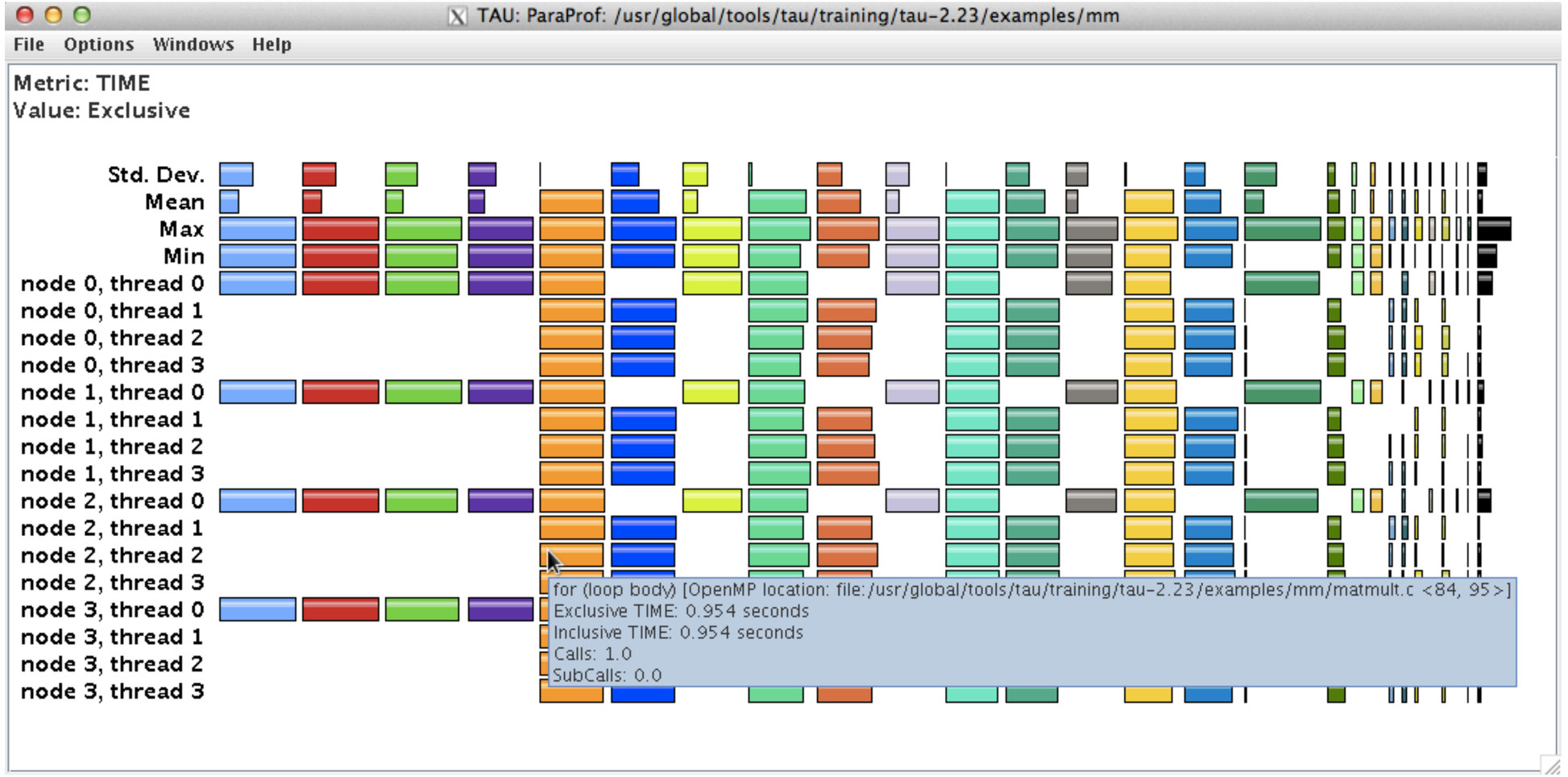
# ParaProf Thread Statistics Table




| Name | Exclusive TIME | Inclusive TIME | Calls | Child Calls |
|---|---|---|---|---|
| .TAU application | 12.111 | 13.341 | 1 | 26,524 |
| [CONTEXT] .TAU application | 0 | 11.971 | 396 | 0 |
| MPI_Allreduce() | 0.038 | 0.038 | 2,816 | 0 |
| MPI_Alltoall() | 0.262 | 0.271 | 1,011 | 105 |
| [CONTEXT] MPI_Alltoall() | 0 | 0.27 | 8 | 0 |
| [SAMPLE] .annobin_pthread_spin_lock.c [{pthread_spin_lock.c} {0}] | 0.03 | 0.03 | 1 | 0 |
| [SAMPLE] PAMI_Context_trylock_advancev [{/m100/prod/opt/com | 0.09 | 0.09 | 2 | 0 |
| [SAMPLE] _ZN4PAMI8Protocol3Get13CompositeRGetINS1_4RGetES3 | 0.03 | 0.03 | 1 | 0 |
| [SAMPLE] __memcpy_power7 [{} {0}] | 0.09 | 0.09 | 3 | 0 |
| [SAMPLE] opal_datatype_copy_content_same_ddt [{/m100/prod/op | 0.03 | 0.03 | 1 | 0 |
| MPI_Barrier() | 0.043 | 0.043 | 3,992 | 0 |
| MPI_Bcast() | 0.004 | 0.004 | 875 | 5 |
| MPI_Comm_free() | 0 | 0 | 11 | 0 |
| MPI_Comm_rank() | 0.002 | 0.002 | 4,221 | 0 |
| MPI_Comm_size() | 0.004 | 0.004 | 4,954 | 0 |
| MPI_Comm_split() | 0.008 | 0.009 | 13 | 26 |
| MPI_Finalize() | 0.399 | 0.416 | 1 | 37 |
| MPI_Gather() | 0 | 0 | 3 | 0 |
| MPI_Get_count() | 0 | 0 | 12 | 0 |
| MPI_Get_processor_name() | 0 | 0 | 1 | 0 |
| MPI_Init_thread() | 0.128 | 0.16 | 1 | 909 |
| MPI_Irecv() | 0.002 | 0.002 | 1,212 | 0 |
| MPI_Isend() | 0.024 | 0.024 | 1,212 | 4 |
| MPI_Recv() | 0.001 | 0.001 | 24 | 0 |

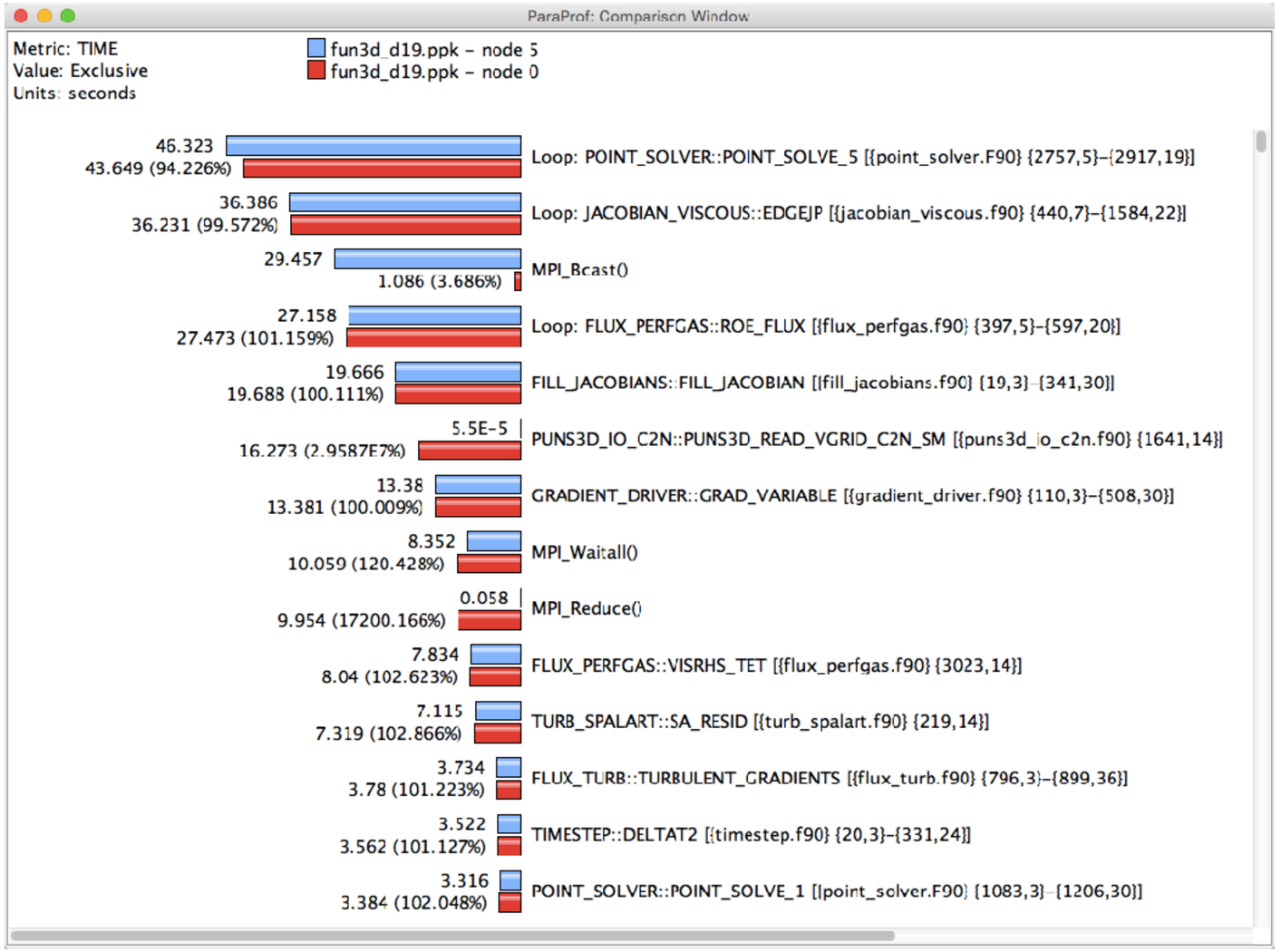
Using sampling, TAU can explain 11.971 seconds out of 12.111 seconds using 396 samples.

# Mixed MPI and OpenMP Instrumentation



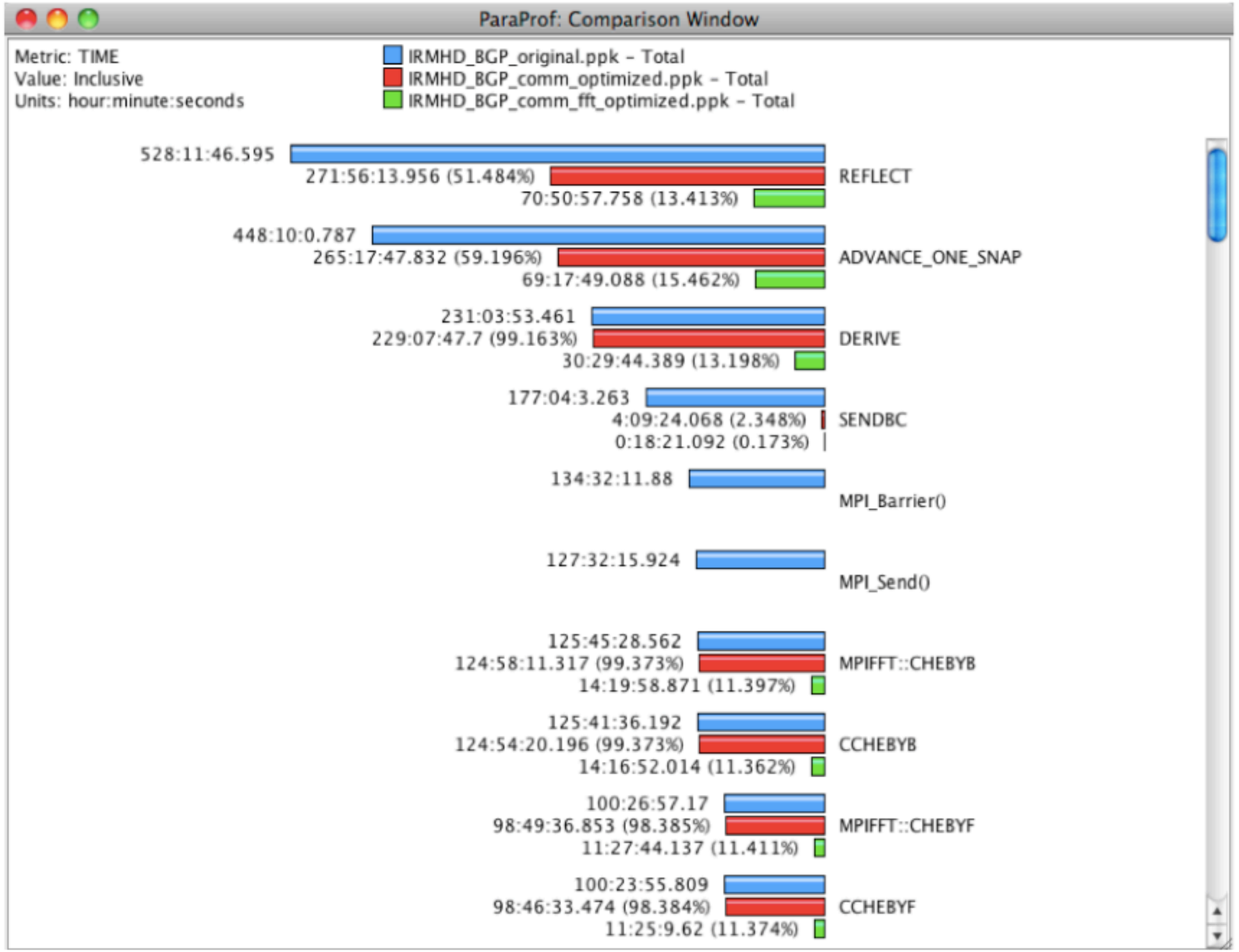

Options -> Uncheck “Stack Bars Together”

# ParaProf Comparison Window

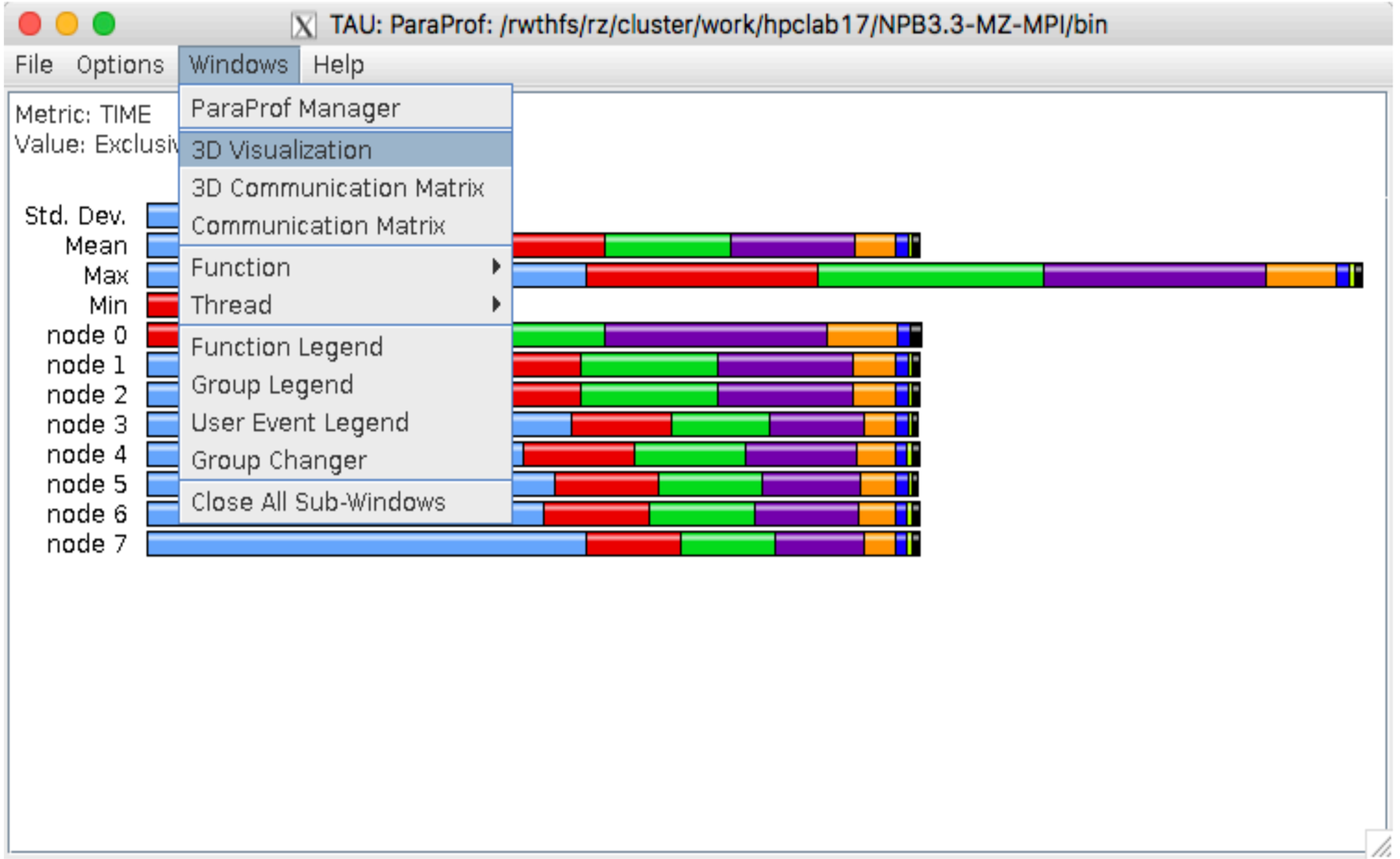



Comparing Rank 0 with 5.
Right click on “node 5” -> Add node to comparison window

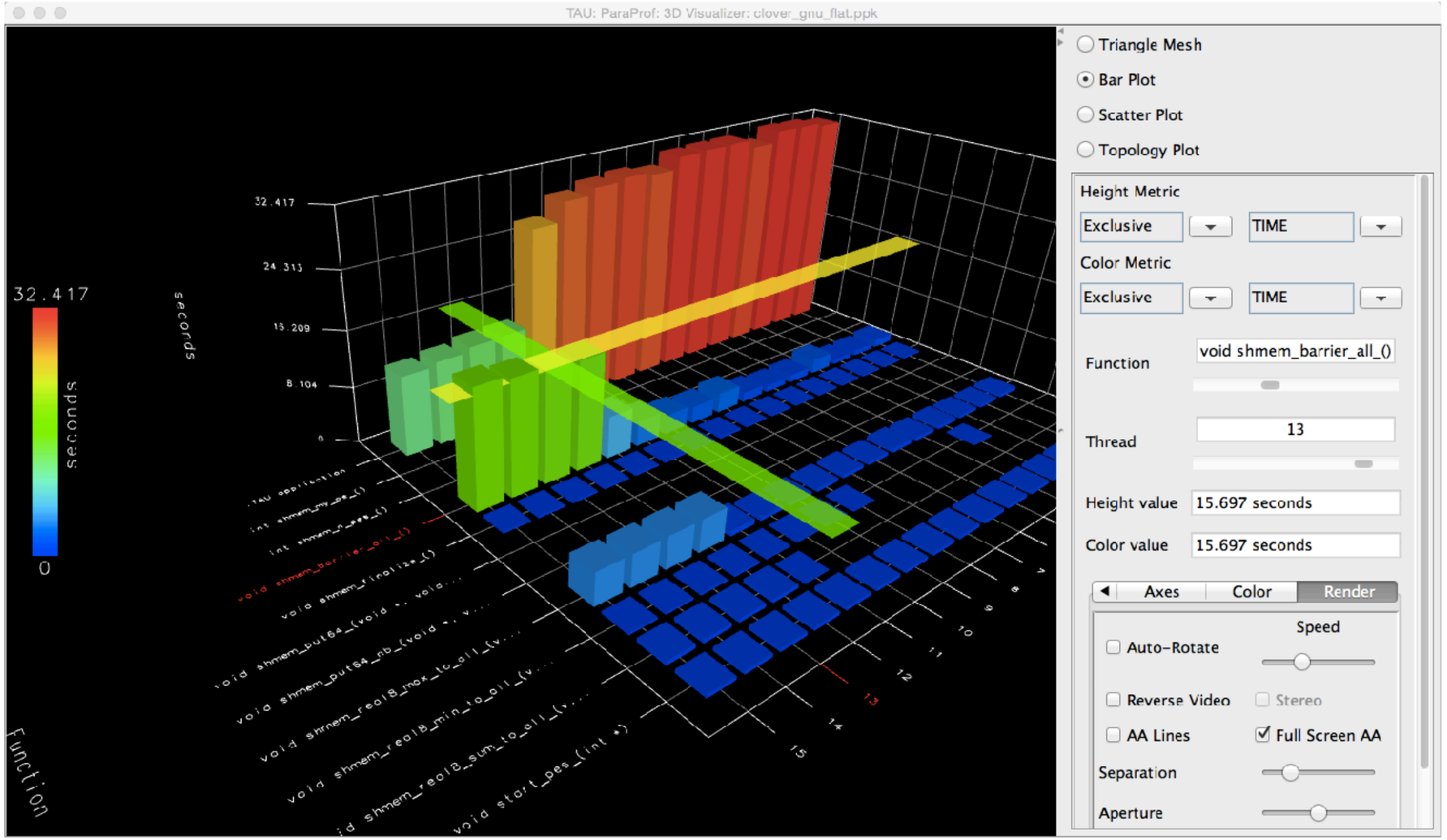
# ParaProf Comparison Window



# 3D Visualization in ParaProf

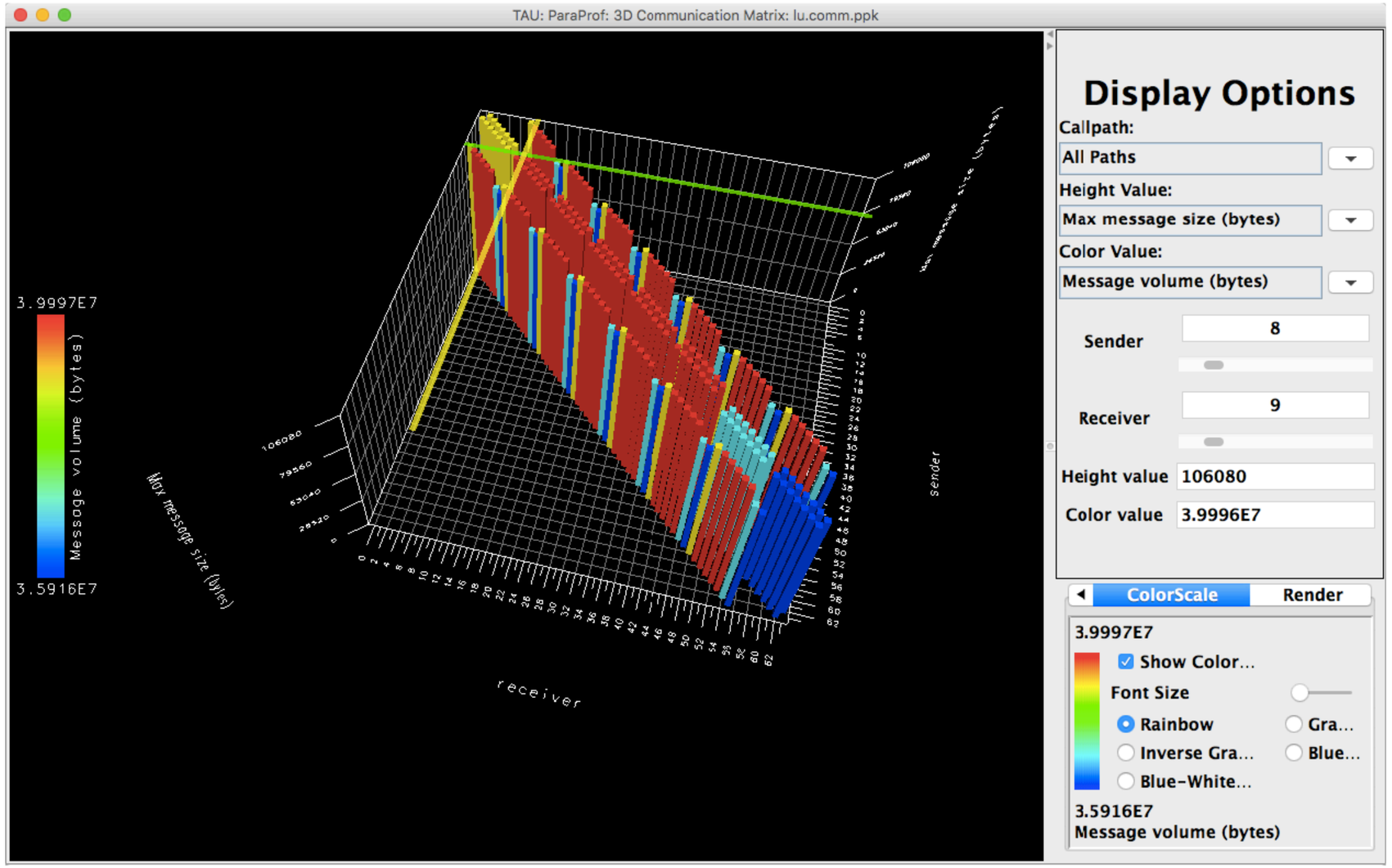


# TAU – ParaProf 3D Visualization



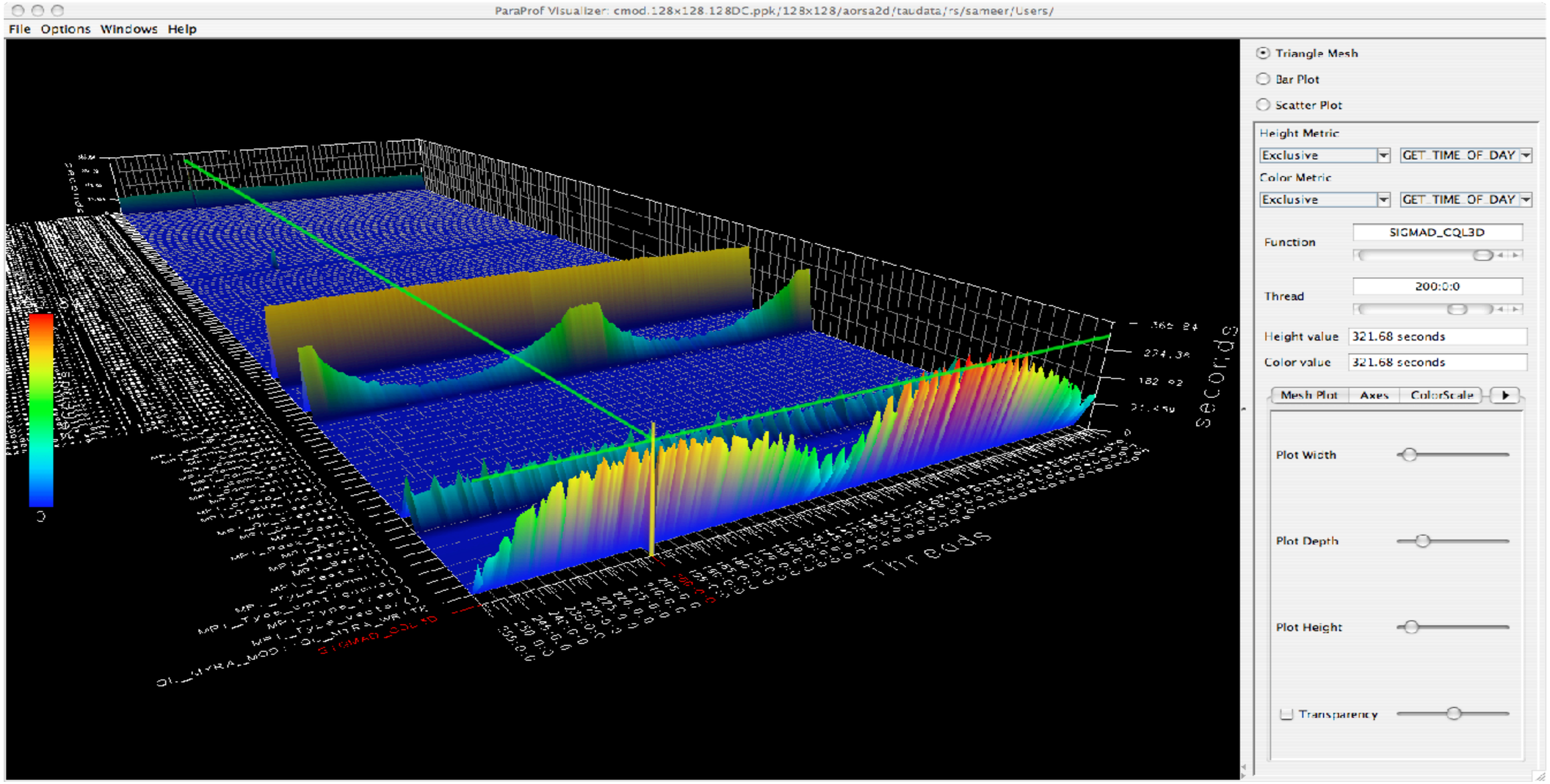

$ paraprof app.ppk
Windows -> 3D Visualization -> Bar Plot (right pane)

# TAU – 3D Communication Window

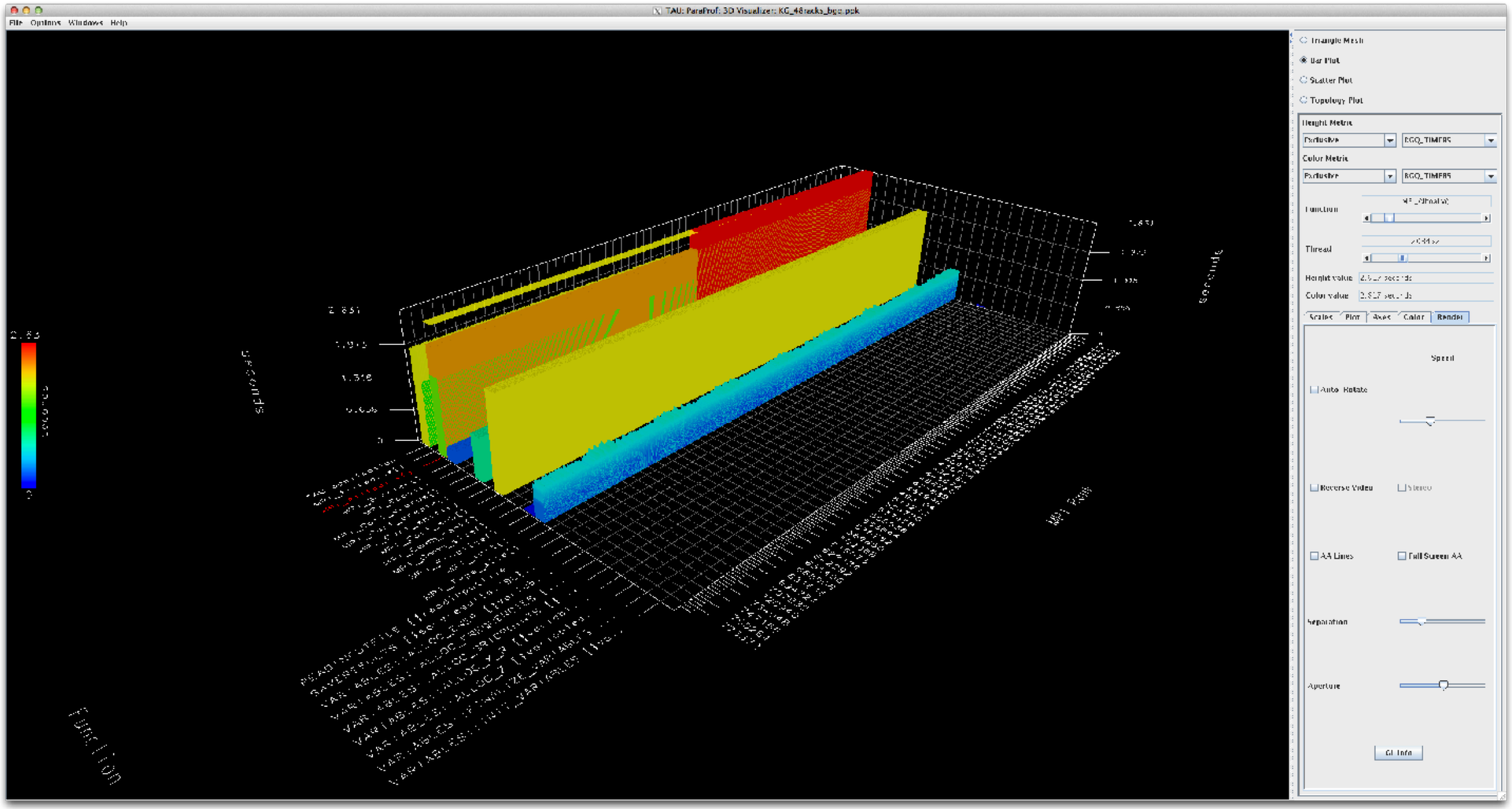



% export TAU_COMM_MATRIX=1; mpirun … tau_exec ./a.out
% paraprof  app.ppk;  Windows -> 3D Communication Matrix

# ParaProf 3D Profile Browser



# ParaProf's Scalable 3D Visualization (BGQ)




786,432 ranks

# Callsite Profiles

# TAU – Callsite Profiling

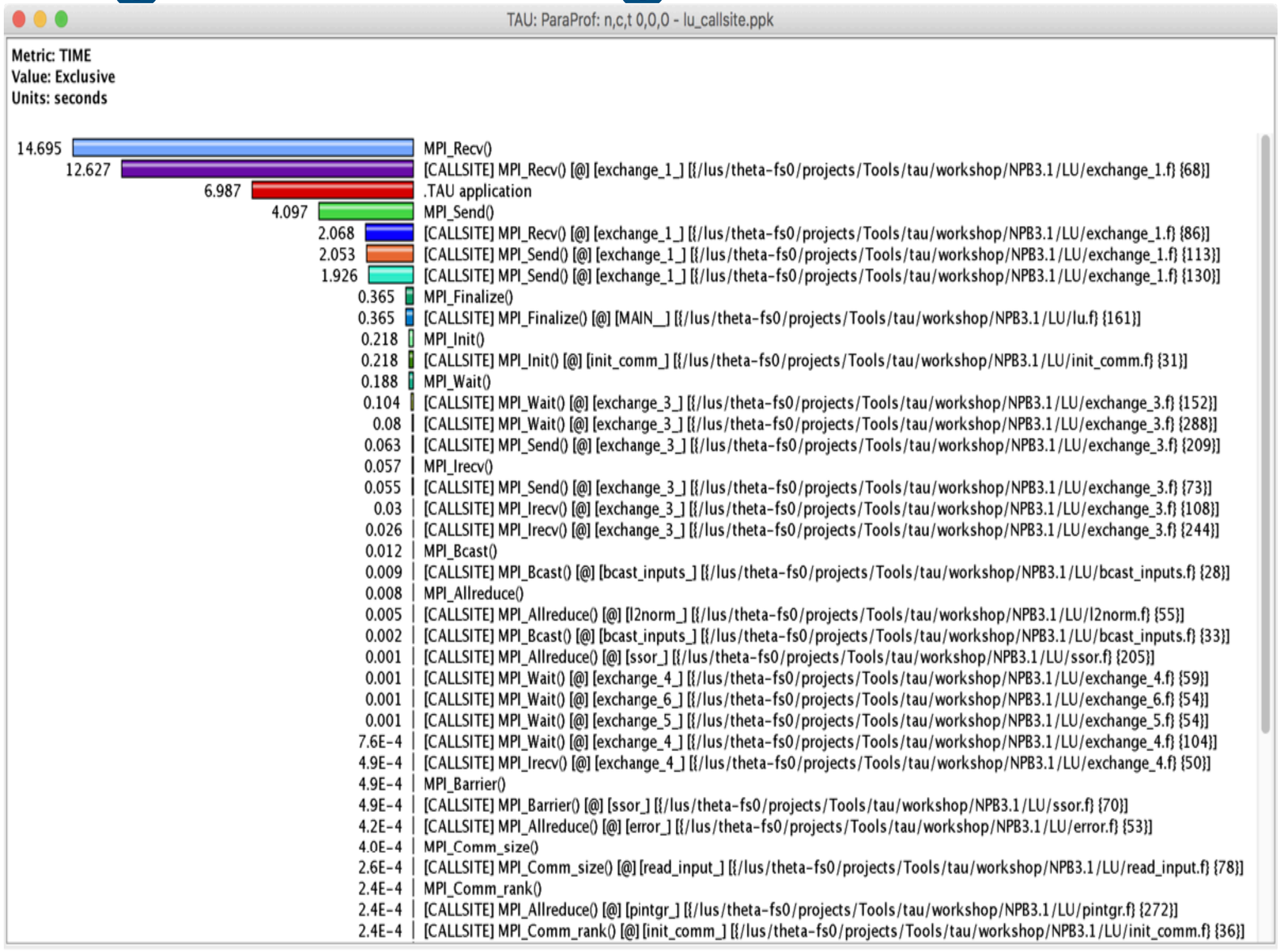
TAU: ParaProf: Statistics for: node 0 - clover_callsite.ppk

| Name | Excl... | Inclu... | Calls | Chil... |
|---|---|---|---|---|
| .TAU application | 6.152 | 8.249 | 1 | 28,383 |
| [CALLSITE] void start_pes_(int *) [@] [/nfsprojects/volta-projects/tau/tau-2.24.1/craycnl/lib/libTAUsh-gnu-papi-shmem-pdt.so] UNRESOLVED ADDR | 0.747 | 0.747 | 1 | 0 |
| void start_pes_(int *) | 0.747 | 0.747 | 1 | 0 |
| void shmem_barrier_all_() | 0.624 | 0.624 | 9,229 | 0 |
| [CALLSITE] void shmem_barrier_all_() [@] [__clover_module_MOD_clover_exchange_message] [{/home/ssshend/CloverLeaf_OpenSHMEM/clover.f90} {572}] | 0.401 | 0.401 | 4,610 | 0 |
| [CALLSITE] void shmem_finalize_() [@] [/nfsprojects/volta-projects/tau/tau-2.24.1/craycnl/lib/libTAUsh-gnu-papi-shmem-pdt.so] UNRESOLVED ADDR | 0.314 | 0.314 | 1 | 0 |
| void shmem_finalize_() | 0.314 | 0.314 | 1 | 0 |
| [CALLSITE] void shmem_barrier_all_() [@] [__clover_module_MOD_clover_exchange_message] [{/home/ssshend/CloverLeaf_OpenSHMEM/clover.f90} {643}] | 0.223 | 0.223 | 4,610 | 0 |
| void shmem_put64_nb_(void *, void *, int *, int *, void *) | 0.159 | 0.159 | 9,220 | 0 |
| void shmem_put64_(void *, void *, int *, int *) | 0.126 | 0.126 | 9,220 | 0 |
| void shmem_real8_max_to_all_(void *, void *, int *, int *, int *, int *, void *, long *) | 0.081 | 0.081 | 400 | 0 |
| [CALLSITE] void shmem_put64_nb_(void *, void *, int *, int *, void *) [@] [__clover_module_MOD_clover_exchange_message] [{/home/ssshend/CloverLeaf_( | 0.07 | 0.07 | 4,610 | 0 |
| [CALLSITE] void shmem_put64_(void *, void *, int *, int *) [@] [__clover_module_MOD_clover_exchange_message] [{/home/ssshend/CloverLeaf_OpenSHME | 0.063 | 0.063 | 4,610 | 0 |
| [CALLSITE] void shmem_real8_max_to_all_(void *, void *, int *, int *, int *, int *, void *, long *) [@] [hydro_] [{/home/ssshend/CloverLeaf_OpenSHMEM/hydr | 0.046 | 0.046 | 200 | 0 |
| [CALLSITE] void shmem_real8_min_to_all_(void *, void *, int *, int *, int *, int *, void *, long *) [@] [/nfsprojects/volta-projects/tau/tau-2.24.1/craycnl/lib/ | 0.04 | 0.04 | 200 | 0 |
| void shmem_real8_min_to_all_(void *, void *, int *, int *, int *, int *, void *, long *) | 0.04 | 0.04 | 200 | 0 |
| [CALLSITE] void shmem_real8_max_to_all_(void *, void *, int *, int *, int *, int *, void *, long *) [@] [hydro_] [{/home/ssshend/CloverLeaf_OpenSHMEM/hydr | 0.036 | 0.036 | 200 | 0 |
| [CALLSITE] void shmem_put64_nb_(void *, void *, int *, int *, void *) [@] [__clover_module_MOD_clover_exchange] [{/home/ssshend/CloverLeaf_OpenSHME | 0.028 | 0.028 | 601 | 0 |

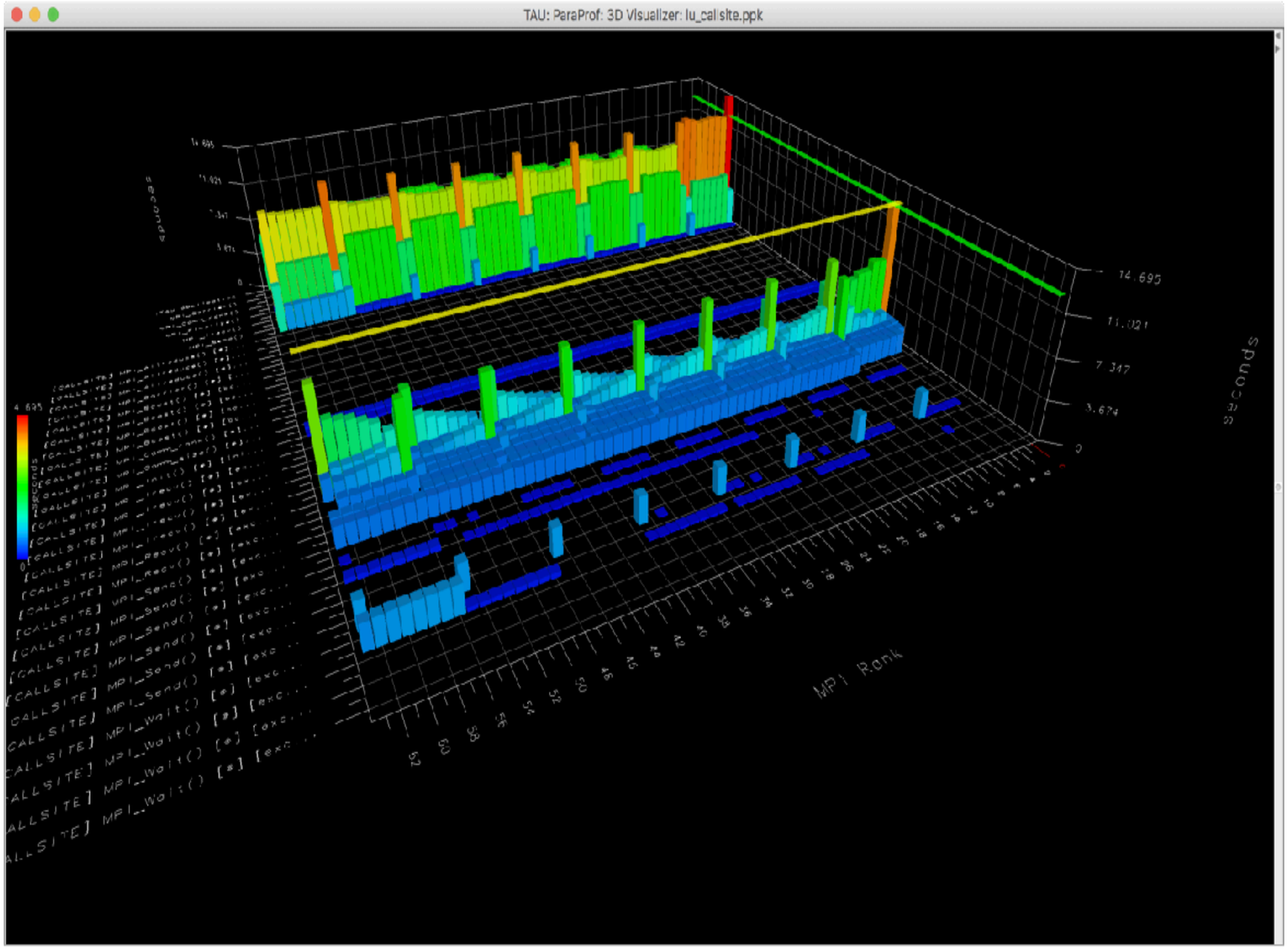
$ export TAU_CALLSITE=1

# Callsite Profiling and Tracing

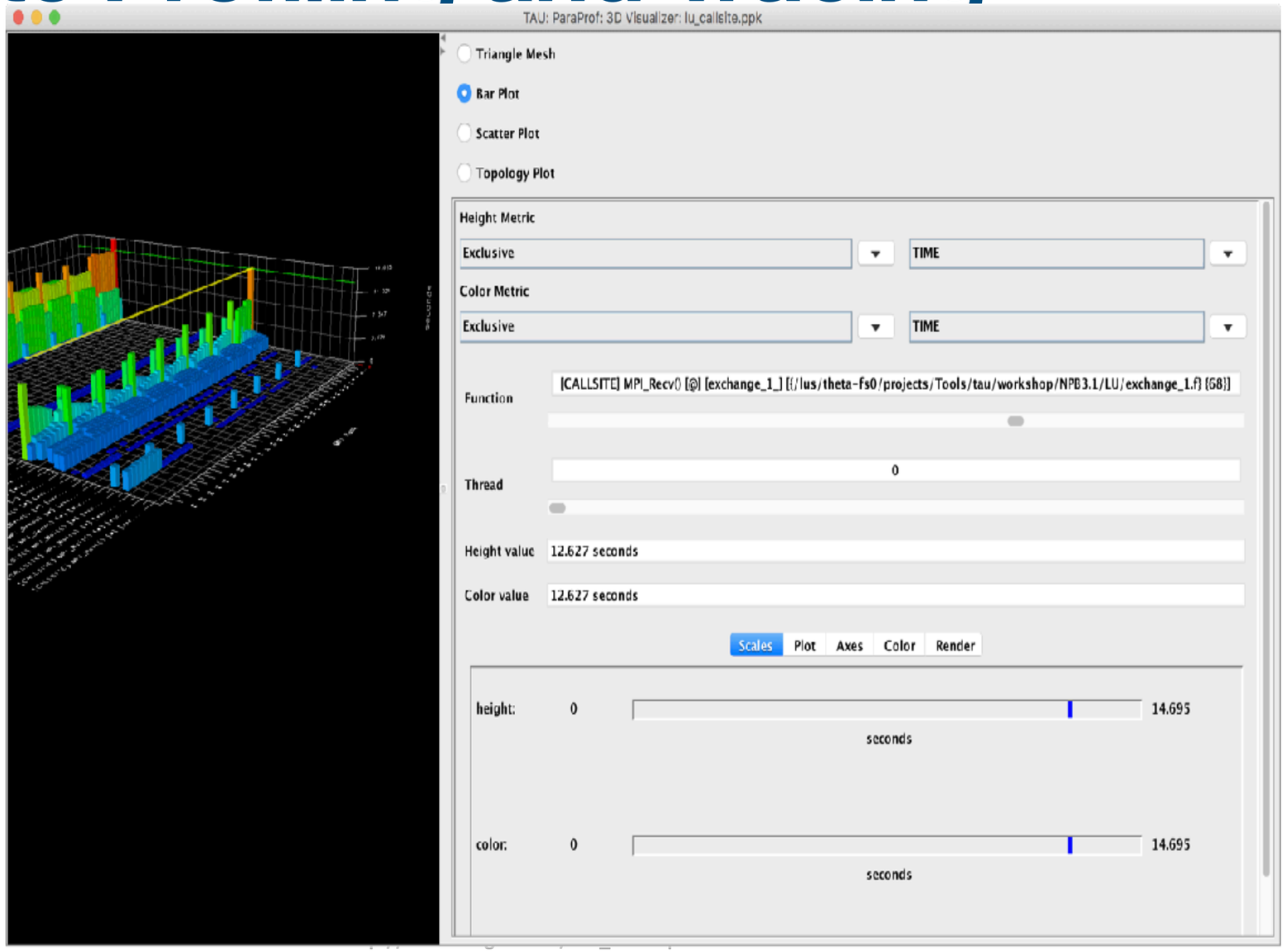
$ export TAU_CALLSITE=1



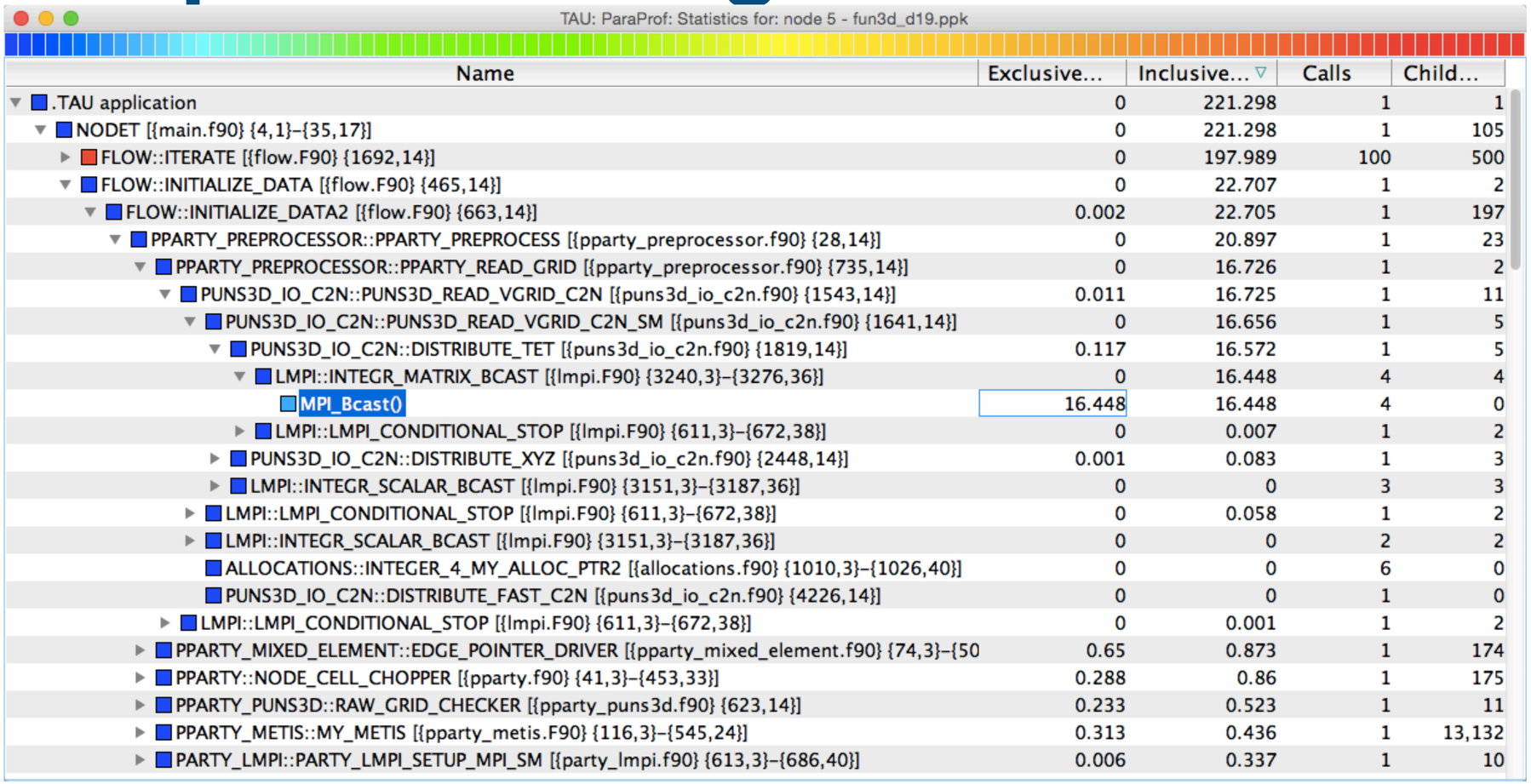
# Callsite Profiling and Tracing



# Callsite Profiling and Tracing



# TAU – Callpath Profiling




| Name | Exclusive... | Inclusive... ▽ | Calls | Child... |
|---|---|---|---|---|
| .TAU application | 0 | 221.298 | 1 | 1 |
| NODET [{main.f90} {4,1}-{35,17}] | 0 | 221.298 | 1 | 105 |
| FLOW::ITERATE [{flow.F90} {1692,14}] | 0 | 197.989 | 100 | 500 |
| FLOW::INITIALIZE_DATA [{flow.F90} {465,14}] | 0 | 22.707 | 1 | 2 |
| FLOW::INITIALIZE_DATA2 [{flow.F90} {663,14}] | 0.002 | 22.705 | 1 | 197 |
| PPARTY_PREPROCESSOR::PPARTY_PREPROCESS [{pparty_preprocessor.f90} {28,14}] | 0 | 20.897 | 1 | 23 |
| PPARTY_PREPROCESSOR::PPARTY_READ_GRID [{pparty_preprocessor.f90} {735,14}] | 0 | 16.726 | 1 | 2 |
| PUNS3D_IO_C2N::PUNS3D_READ_VGRID_C2N [{puns3d_io_c2n.f90} {1543,14}] | 0.011 | 16.725 | 1 | 11 |
| PUNS3D_IO_C2N::PUNS3D_READ_VGRID_C2N_SM [{puns3d_io_c2n.f90} {1641,14}] | 0 | 16.656 | 1 | 5 |
| PUNS3D_IO_C2N::DISTRIBUTE_TET [{puns3d_io_c2n.f90} {1819,14}] | 0.117 | 16.572 | 1 | 5 |
| LMPI::INTEGR_MATRIX_BCAST [{lmpi.F90} {3240,3}-{3276,36}] | 0 | 16.448 | 4 | 4 |
| MPI_Bcast() | 16.448 | 16.448 | 4 | 0 |
| LMPI::LMPI_CONDITIONAL_STOP [{lmpi.F90} {611,3}-{672,38}] | 0 | 0.007 | 1 | 2 |
| PUNS3D_IO_C2N::DISTRIBUTE_XYZ [{puns3d_io_c2n.f90} {2448,14}] | 0.001 | 0.083 | 1 | 3 |
| LMPI::INTEGR_SCALAR_BCAST [{lmpi.F90} {3151,3}-{3187,36}] | 0 | 0 | 3 | 3 |
| LMPI::LMPI_CONDITIONAL_STOP [{lmpi.F90} {611,3}-{672,38}] | 0 | 0.058 | 1 | 2 |
| LMPI::INTEGR_SCALAR_BCAST [{lmpi.F90} {3151,3}-{3187,36}] | 0 | 0 | 2 | 2 |
| ALLOCATIONS::INTEGER_4_MY_ALLOC_PTR2 [{allocations.f90} {1010,3}-{1026,40}] | 0 | 0 | 6 | 0 |
| PUNS3D_IO_C2N::DISTRIBUTE_FAST_C2N [{puns3d_io_c2n.f90} {4226,14}] | 0 | 0 | 1 | 0 |
| LMPI::LMPI_CONDITIONAL_STOP [{lmpi.F90} {611,3}-{672,38}] | 0 | 0.001 | 1 | 2 |
| PPARTY_MIXED_ELEMENT::EDGE_POINTER_DRIVER [{pparty_mixed_element.f90} {74,3}-{50 | 0.65 | 0.873 | 1 | 174 |
| PPARTY::NODE_CELL_CHOPPER [{pparty.f90} {41,3}-{453,33}] | 0.288 | 0.86 | 1 | 175 |
| PPARTY_PUNS3D::RAW_GRID_CHECKER [{pparty_puns3d.f90} {623,14}] | 0.233 | 0.523 | 1 | 11 |
| PPARTY_METIS::MY_METIS [{pparty_metis.F90} {116,3}-{545,24}] | 0.313 | 0.436 | 1 | 13,132 |
| PARTY_LMPI::PARTY_LMPI_SETUP_MPI_SM [{party_lmpi.f90} {613,3}-{686,40}] | 0.006 | 0.337 | 1 | 10 |

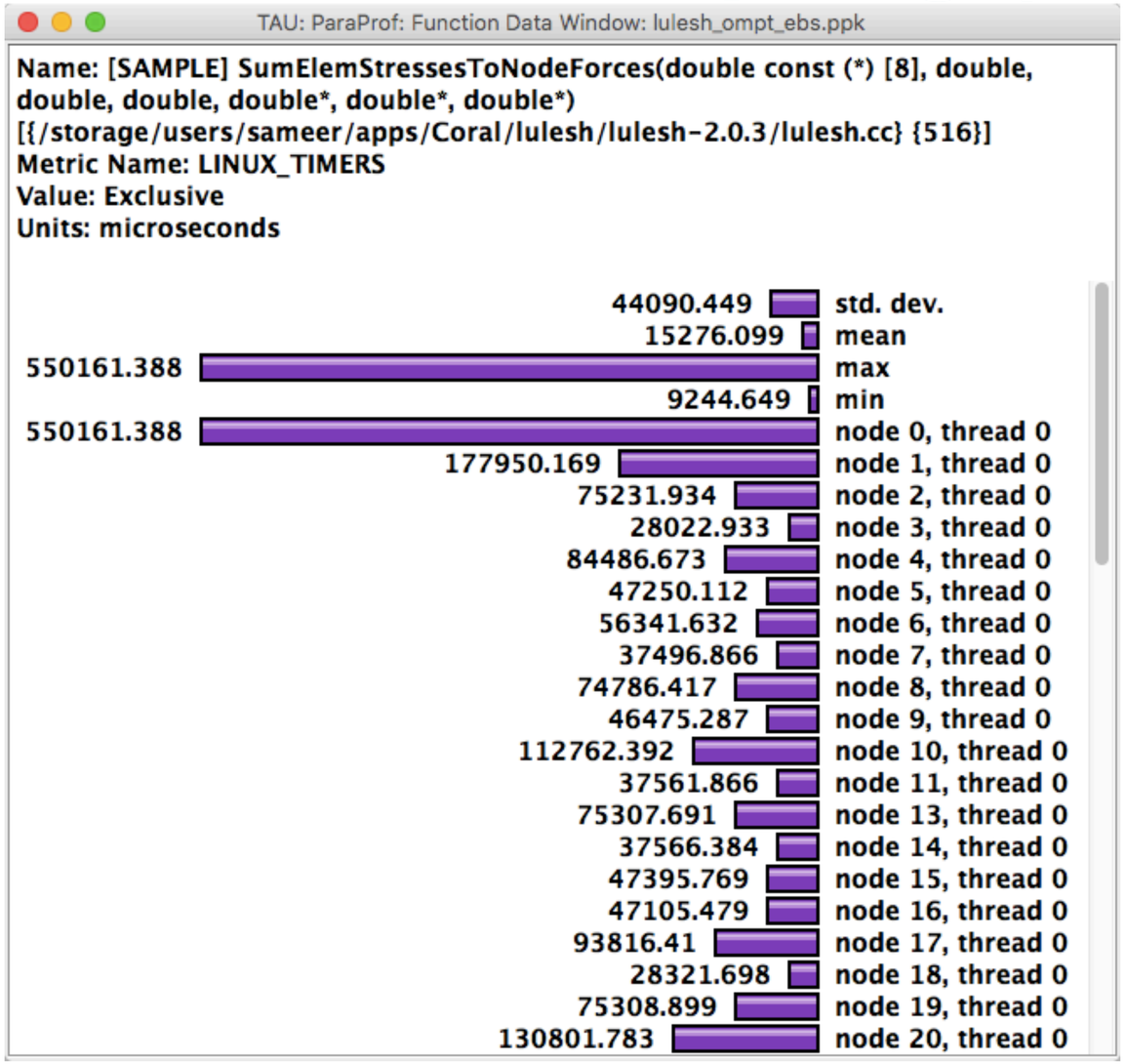
$ export TAU_CALLPATH=1; export TAU_CALLPATH_DEPTH=100

# Sample Callstacks

# TAU – Callstack Sampling


$ export TAU_SAMPLING=1; export TAU_EBS_UNWIND=1

# TAU – Event Based Sampling (EBS)




$ export TAU_SAMPLING=1 (same as -ebs)

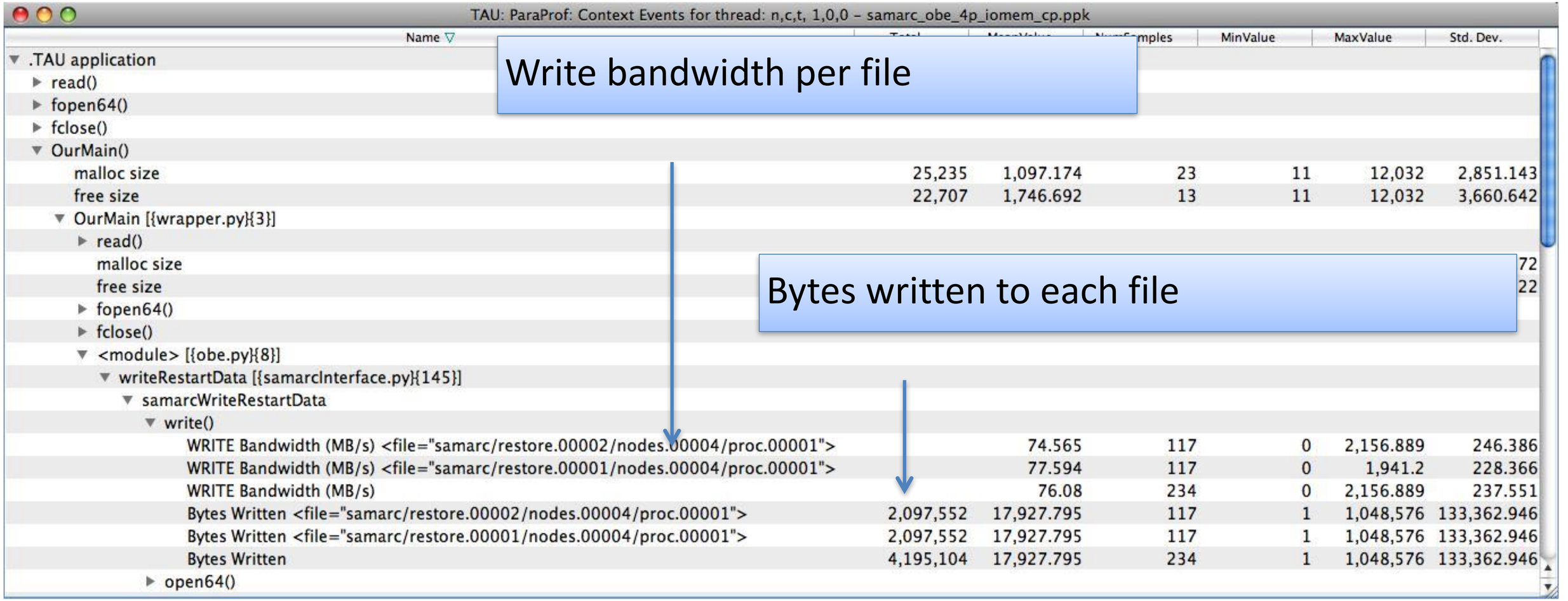
# Atomic Events (incl. I/O)

# TAU Atomic Events

TAU: ParaProf: Context Events for: node 0 - /Users/sameer/tmp

| Name | Total | NumSamples | MaxValue | MinValue | MeanValue | Std. Dev. |
|---|---|---|---|---|---|---|
| Bytes Written <file=stdout> | 911 | 62 | 21 | 1 | 14.694 | 7.441 |
| Bytes Written <file=pipe> | 22 | 22 | 1 | 1 | 1 | 0 |
| Bytes Written <file=Process_Output/VelRsdl.dat> | 7,826 | 100 | 302 | 76 | 78.26 | 22.487 |
| Bytes Written <file=Process_Output/MomRsdl.dat> | 7,826 | 100 | 302 | 76 | 78.26 | 22.487 |
| Bytes Written <file=Process_Output/MassRsdl.dat> | 11,325 | 100 | 435 | 110 | 113.25 | 32.337 |
| Bytes Written <file=Grid_Output/bodyBndry.dat> | 9,724 | 5 | 8,192 | 4 | 1,944.8 | 3,174.201 |
| Bytes Written <file=/home/sameer/apps/sukra/RotCFD_Regression/case_catalog/UNS2D/NA | 45 | 1 | 45 | 45 | 45 | 0 |
| Bytes Written <file=./Restarts/Restart_History//NACA0012_LargeGrid_00010.Rst> | 44,619,720 | 5,484 | 8,192 | 4 | 8,136.346 | 640.325 |
| Bytes Written <file=./Restarts/Restart_History//NACA0012_LargeGrid_00005.Rst> | 44,619,720 | 5,484 | 8,192 | 4 | 8,136.346 | 640.325 |
| Bytes Written <file=./Restarts//NACA0012_LargeGrid.Rst> | 44,619,720 | 5,484 | 8,192 | 4 | 8,136.346 | 640.325 |
| Bytes Written <file=./Process_Output/TurbRsdl.dat> | 4,271 | 72 | 224 | 57 | 59.319 | 19.544 |
| Bytes Written <file=./Process_Output/Solver.out> | 2,039 | 13 | 797 | 43 | 156.846 | 191.359 |
| Bytes Written <file=./Field_Solutions/Solution_History/NACA0012_LargeGrid_00010.Sln> | 4,356,976 | 534 | 8,192 | 4 | 8,159.131 | 501.319 |
| Bytes Written <file=./Field_Solutions/Solution_History/NACA0012_LargeGrid_00005.Sln> | 4,356,976 | 534 | 8,192 | 4 | 8,159.131 | 501.319 |
| Bytes Written <file=./Field_Solutions/NACA0012_LargeGrid.Sln> | 4,356,976 | 534 | 8,192 | 4 | 8,159.131 | 501.319 |
| Bytes Written <file=./Body_Pressure/NACA0012_LargeGrid_00010_body.Prs> | 65,986 | 9 | 8,190 | 1,300 | 7,331.778 | 2,133.204 |
| Bytes Written <file=./Body_Pressure/NACA0012_LargeGrid_00005_body.Prs> | 65,986 | 9 | 8,190 | 1,300 | 7,331.778 | 2,133.204 |
| Bytes Written <file=./Body_Pressure/FrcMnt.out> | 1,497 | 3 | 1,185 | 108 | 499 | 486.656 |
| Bytes Written | 147,107,546 | 18,550 | 8,192 | 1 | 7,930.326 | 1,420.552 |

# TAU – Context Events




| Name | Total | MeanValue | NumSamples | MinValue | MaxValue | Std. Dev. |
|---|---|---|---|---|---|---|
| .TAU application | | | | | | |
| read() | | | | | | |
| fopen64() | | | | | | |
| fclose() | | | | | | |
| OurMain() | | | | | | |
| malloc size | 25,235 | 1,097.174 | 23 | 11 | 12,032 | 2,851.143 |
| free size | 22,707 | 1,746.692 | 13 | 11 | 12,032 | 3,660.642 |
| OurMain [{wrapper.py}{3}] | | | | | | |
| read() | | | | | | |
| malloc size | | | | | | 72 |
| free size | | | | | | 22 |
| fopen64() | | | | | | |
| fclose() | | | | | | |
| <module> [{obe.py}{8}] | | | | | | |
| writeRestartData [{samarcInterface.py}{145}] | | | | | | |
| samarcWriteRestartData | | | | | | |
| write() | | | | | | |
| WRITE Bandwidth (MB/s) <file="samarc/restore.00002/nodes.00004/proc.00001"> | | 74.565 | 117 | 0 | 2,156.889 | 246.386 |
| WRITE Bandwidth (MB/s) <file="samarc/restore.00001/nodes.00004/proc.00001"> | | 77.594 | 117 | 0 | 1,941.2 | 228.366 |
| WRITE Bandwidth (MB/s) | | 76.08 | 234 | 0 | 2,156.889 | 237.551 |
| Bytes Written <file="samarc/restore.00002/nodes.00004/proc.00001"> | 2,097,552 | 17,927.795 | 117 | 1 | 1,048,576 | 133,362.946 |
| Bytes Written <file="samarc/restore.00001/nodes.00004/proc.00001"> | 2,097,552 | 17,927.795 | 117 | 1 | 1,048,576 | 133,362.946 |
| Bytes Written | 4,195,104 | 17,927.795 | 234 | 1 | 1,048,576 | 133,362.946 |
| open64() | | | | | | |

# Trace: Vampir

# Vampir

Alternative and supplement to automatic analysis

Show dynamic run-time behavior graphically at any level of detail

Provide statistics and performance metrics

Timeline charts

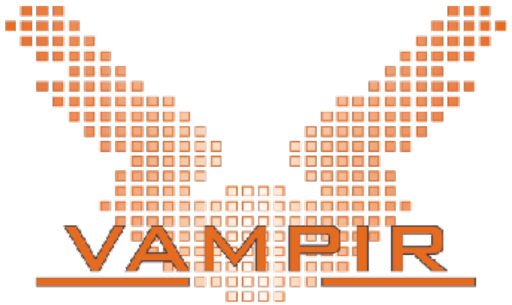
- Show application activities and communication along a time axis

Summary charts

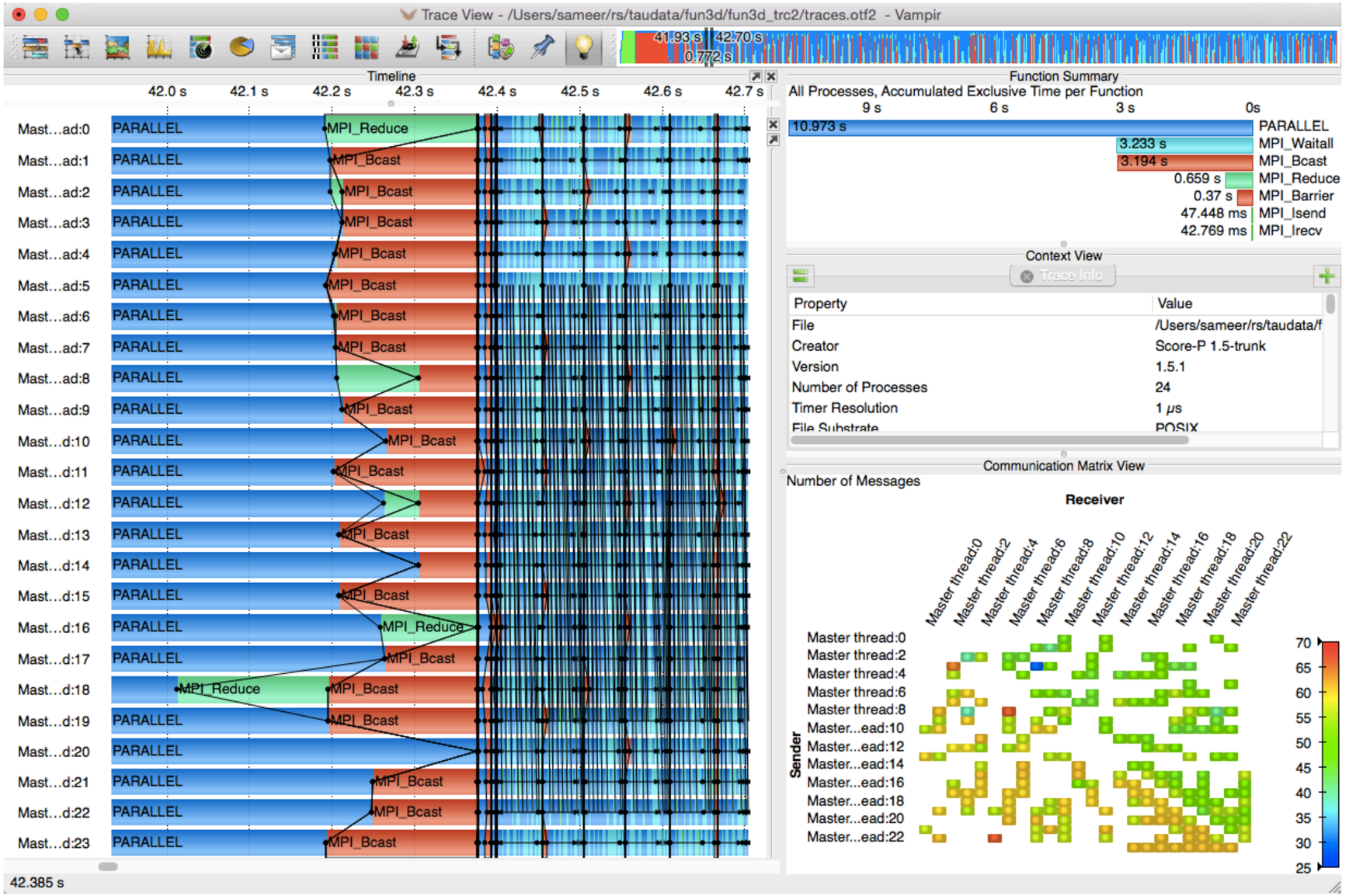
- Provide quantitative results for the currently selected time interval
- Commercial trace visualization tool

*From TU Dresden, Germany*

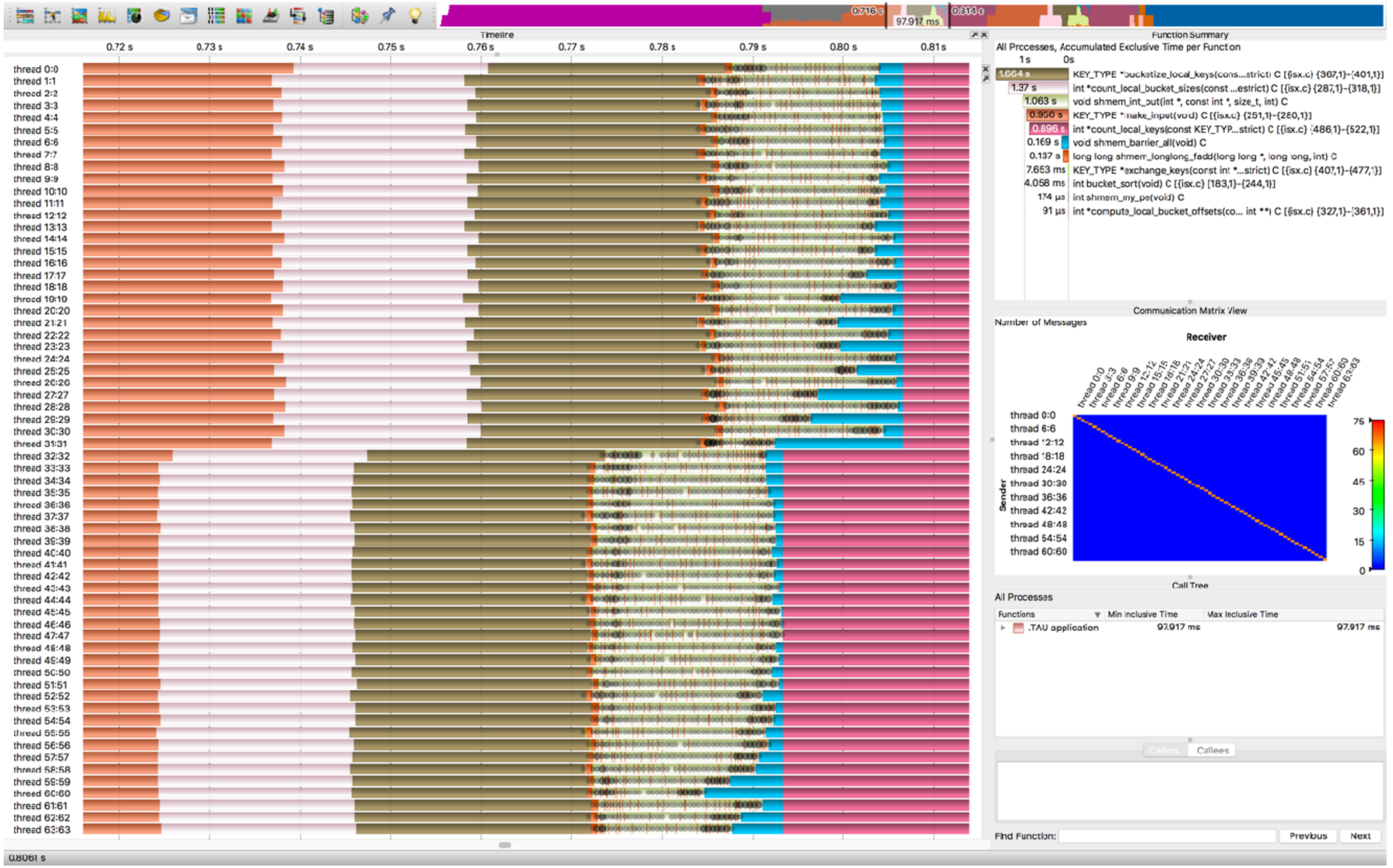
http://www.vampir.eu



# Vampir – Trace Visualization

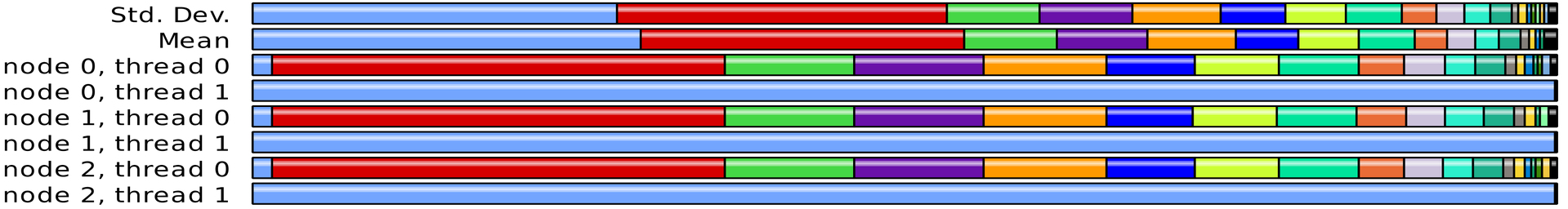


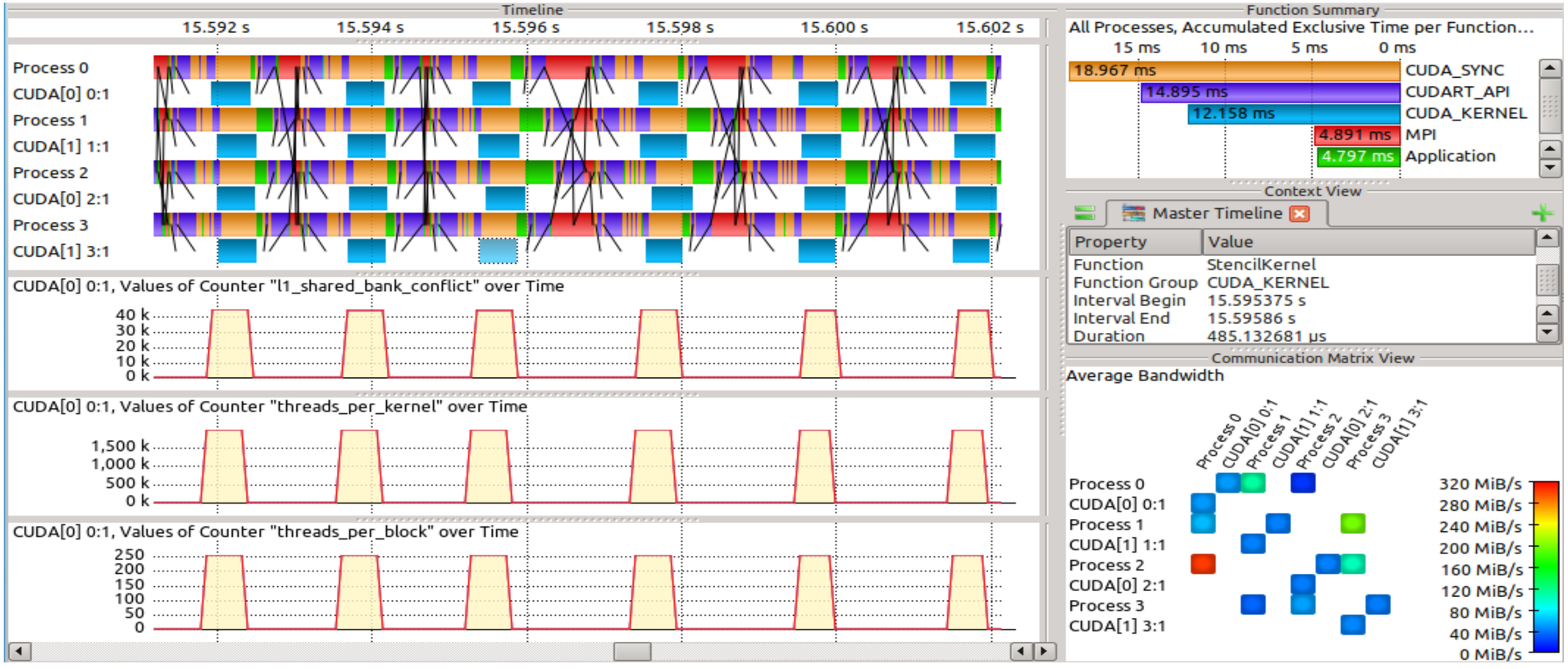
# Vampir – Trace Visualization



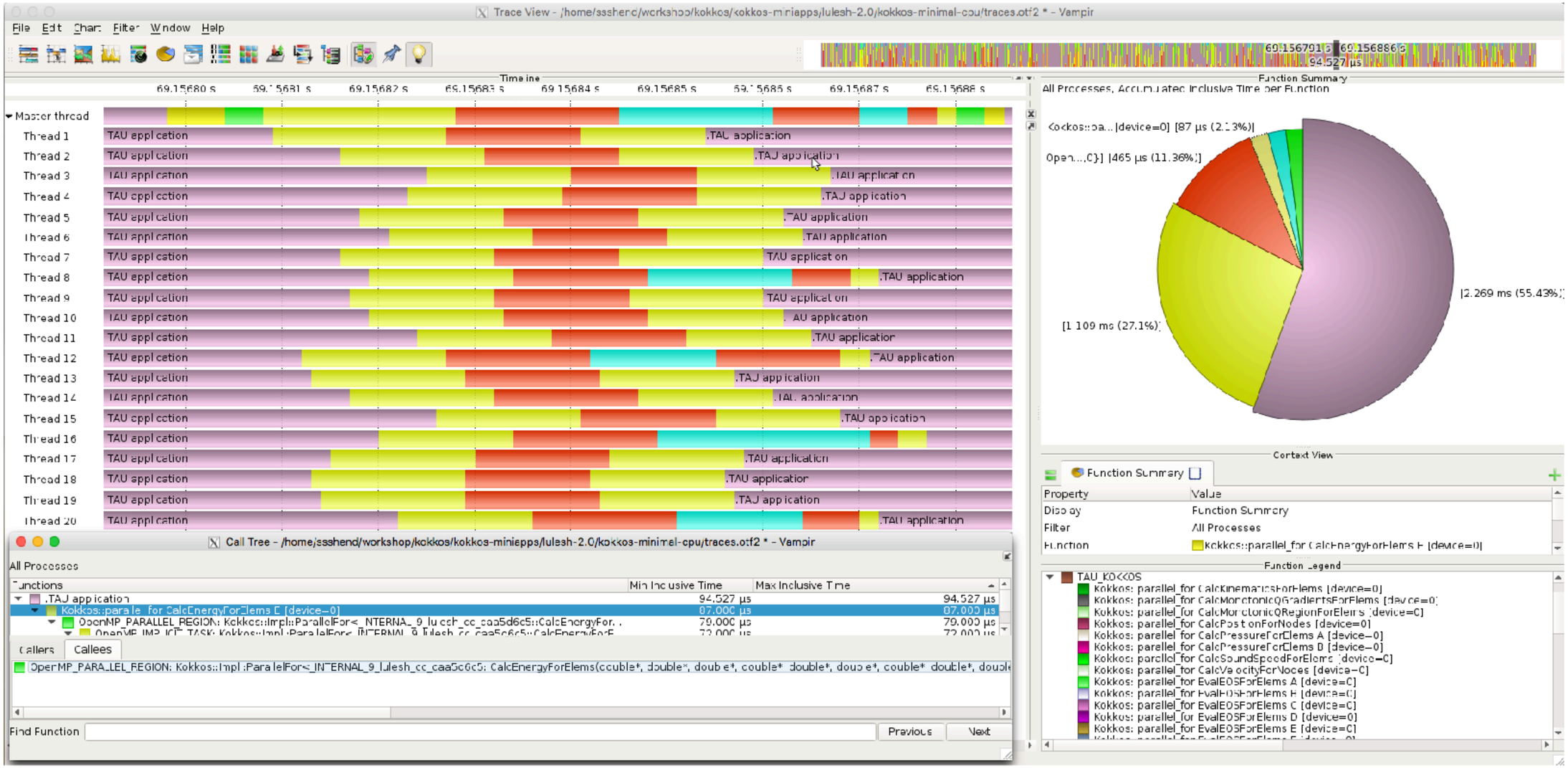
# Stencil2D Parallel Profile / Trace in Vampir

Metric: TAUGPU_TIME
Value: Exclusive



# Vampir – TAU's Kokkos Profiling Interface



# Trace: Jumpshot

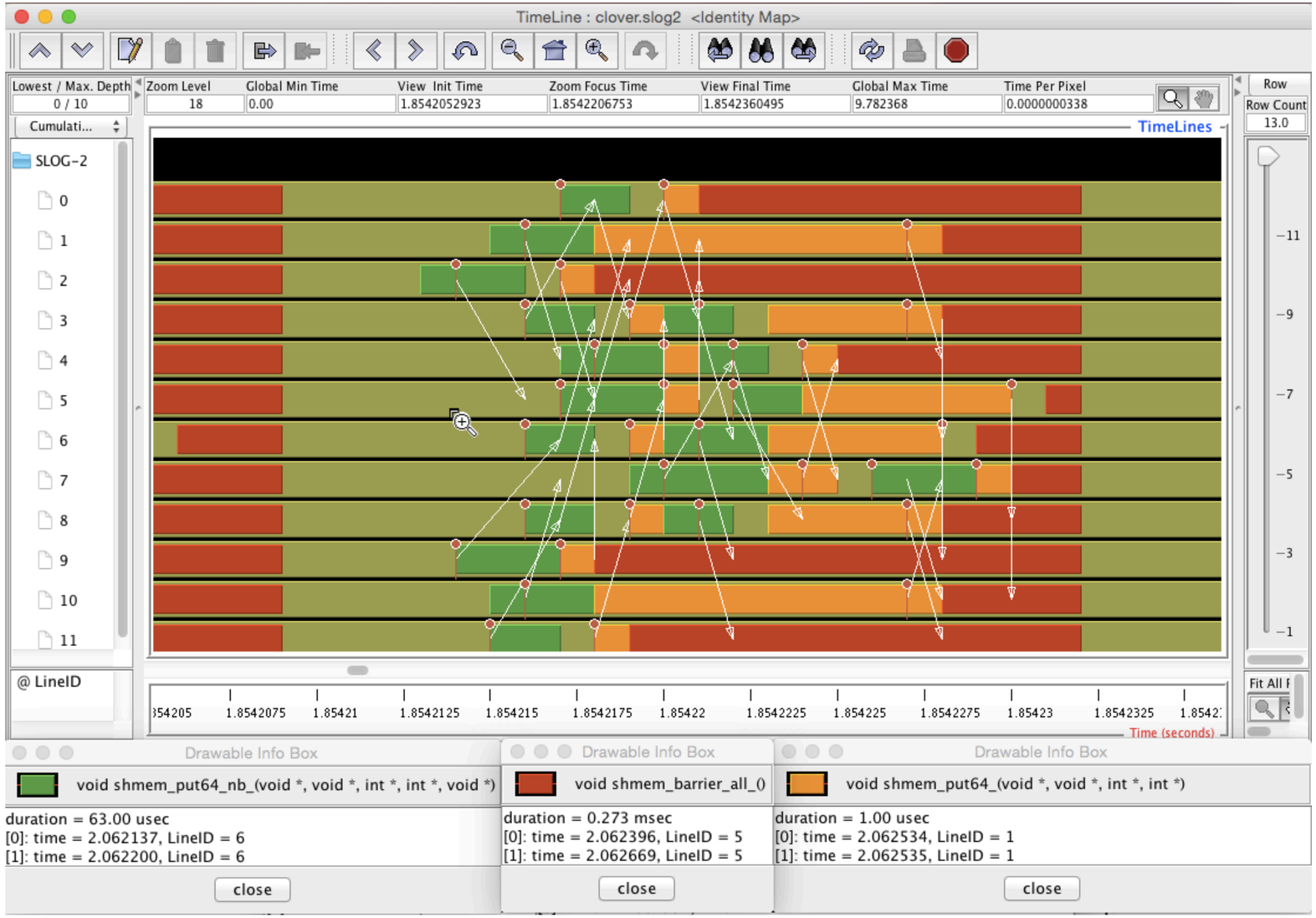
# Jumpshot

- Open source alternative to Vampir
- Developed by Argonne National Laboratory
- Packaged with TAU

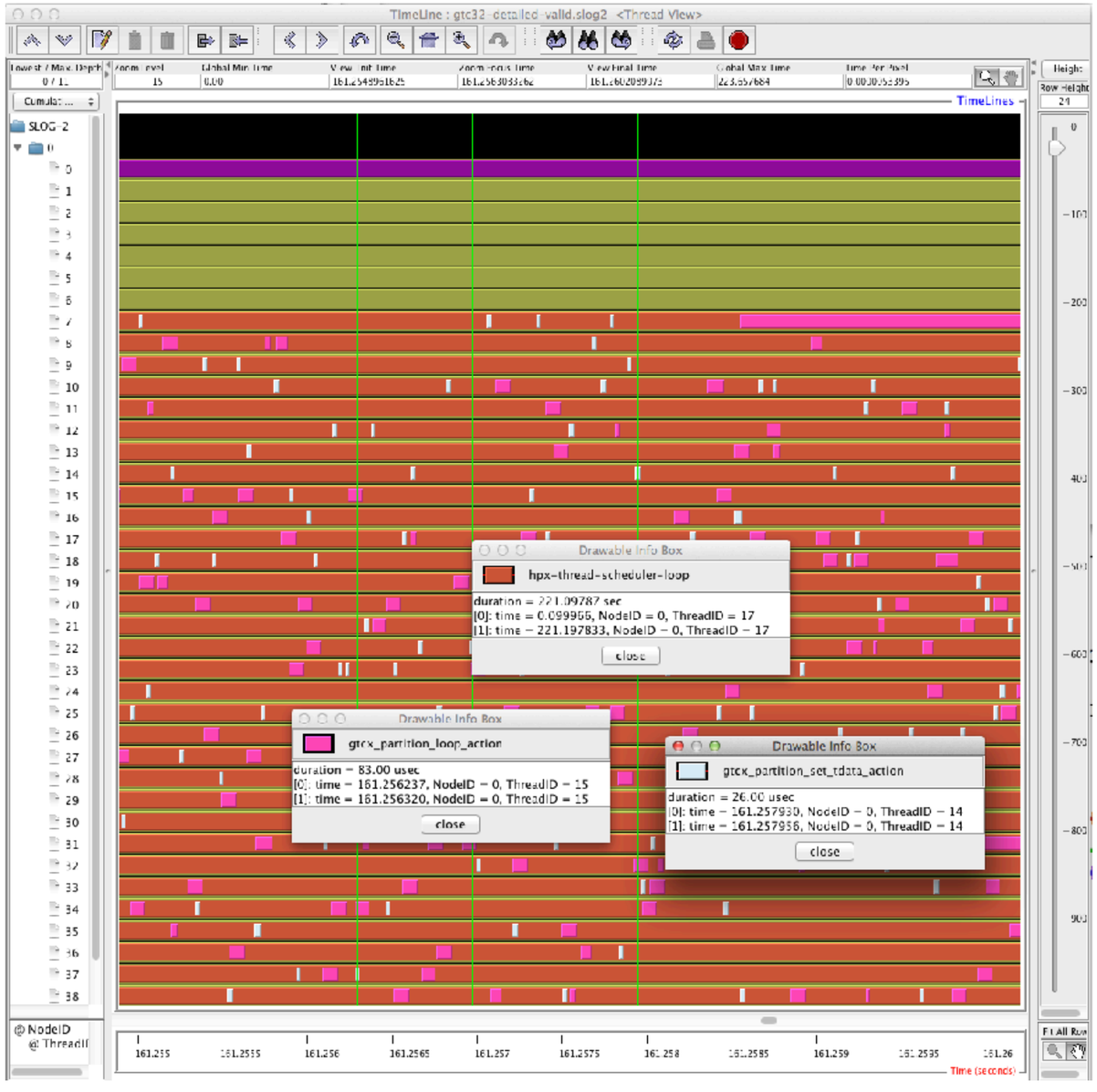
Timeline charts

- Show application activities and communication along a time axis
- Shows boxes within boxes to show nesting of events

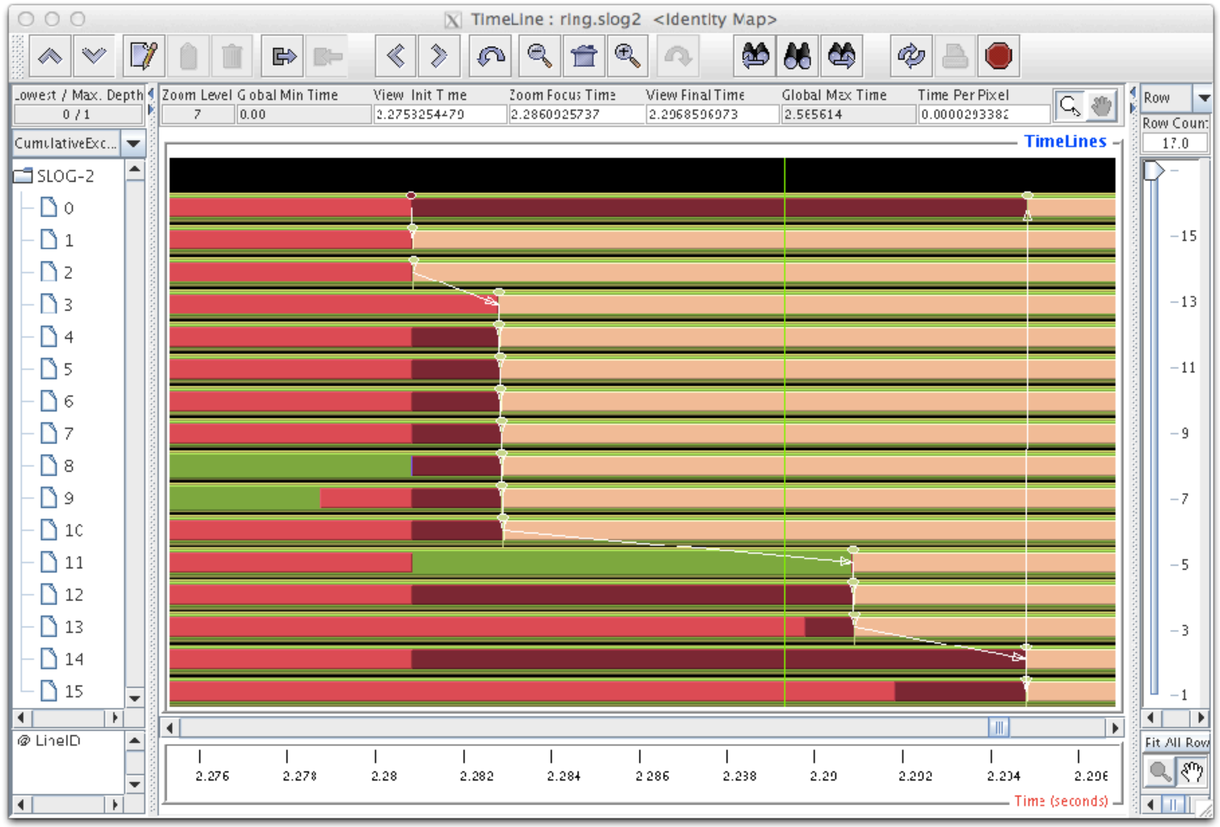
# Jumpshot


TimeLine : clover.slog2 <Identity Map>
Lowest / Max. Depth
0 / 10
Zoom Level
18
Global Min Time
0.00
View Init Time
1.8542052923
Zoom Focus Time
1.8542206753
View Final Time
1.8542360495
Global Max Time
9.782368
Time Per Pixel
0.0000000338
Row
Row Count
13.0
TimeLines
SLOG-2
0
1
2
3
4
5
6
7
8
9
10
11
@ LineID
Time (seconds)
Drawable Info Box
void shmem_put64_nb_(void *, void *, int *, int *, void *)
duration = 63.00 usec
[0]: time = 2.062137, LineID = 6
[1]: time = 2.062200, LineID = 6
close
Drawable Info Box
void shmem_barrier_all_()
duration = 0.273 msec
[0]: time = 2.062396, LineID = 5
[1]: time = 2.062669, LineID = 5
close
Drawable Info Box
void shmem_put64_(void *, void *, int *, int *)
duration = 1.00 usec
[0]: time = 2.062534, LineID = 1
[1]: time = 2.062535, LineID = 1
close

# Jumpshot Trace Visualizer in TAU



# Tracing Communication in Jumpshot




$ export TAU_MAKEFILE=$TAU_MAKEFILE_BASE-icpc-papi-mpi-pdt
$ cmake –DCMAKE_CXX_COMPILER=tau_cxx.sh; make –j 8
$ export TAU_TRACE=1
$ mpirun –np 16 ./a.out ; tau_treemerge.pl; tau2slog2 tau.trc tau.edf –o a.slog2
$ jumpshot a.slog2 &

# Generating Event Traces

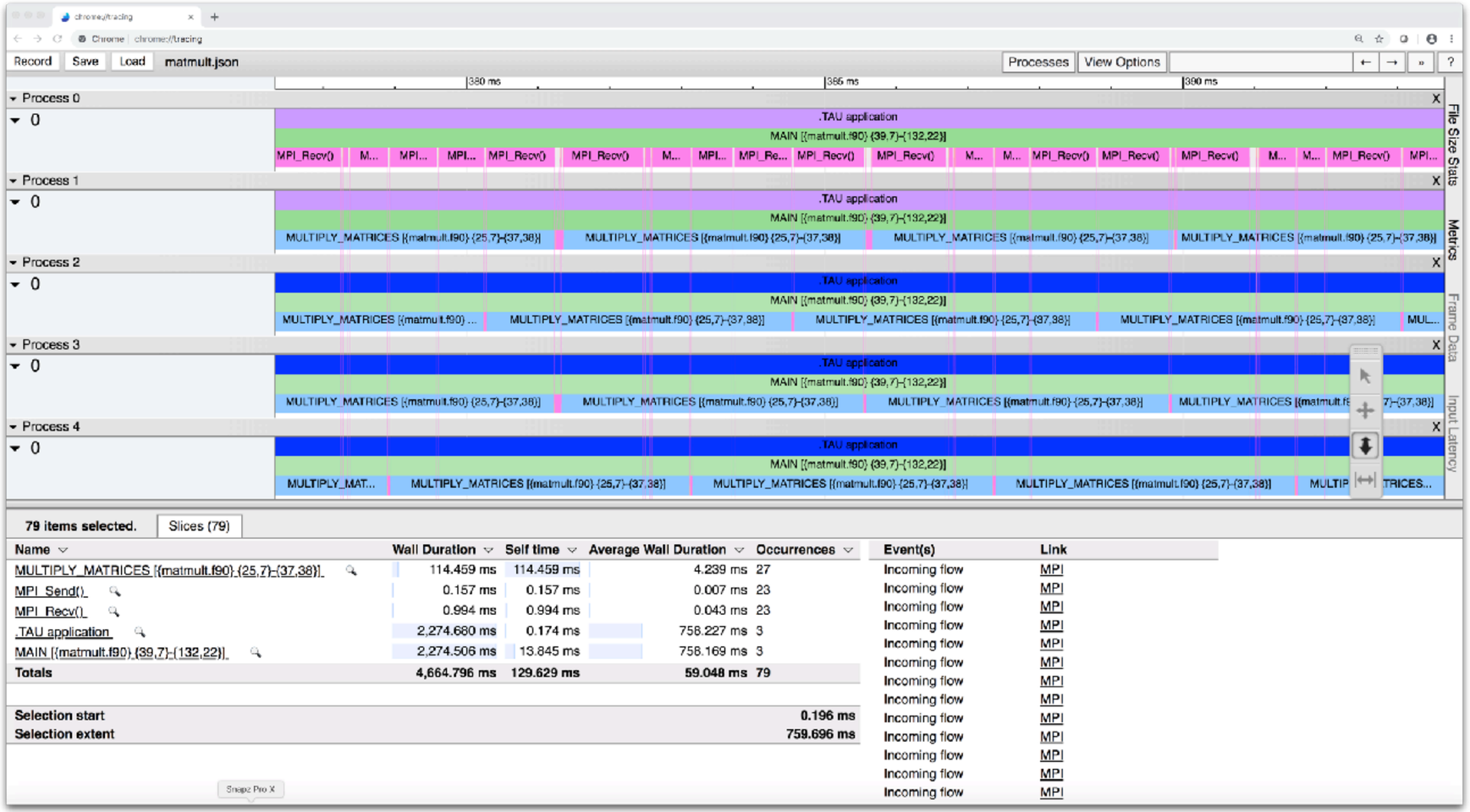
```
% export TAU_MAKEFILE=$TAU/Makefile.tau-icpc-mpi-pdt
% make F90=tau_f90.sh
(Or edit Makefile and change F90=tau_f90.sh)

For Jumpshot:
% export TAU_TRACE=1
% mpirun -np 64   ./a.out

% tau_treemerge.pl
% tau_treemerge; tau2slog2 tau.trc tau.edf -o app.slog2;
% jumpshot app.slog2 &

For Vampir:
% export TAU_TRACE_FORMAT=otf2
# TAU's native OTF2 trace generation capability!
% mpirun -np 64  ./a.out
% vampir traces.otf2 &

For ParaVer:
% tau_convert -paraver tau.trc tau.edf app.prv; paraver app.prv
```

# Trace: Browser

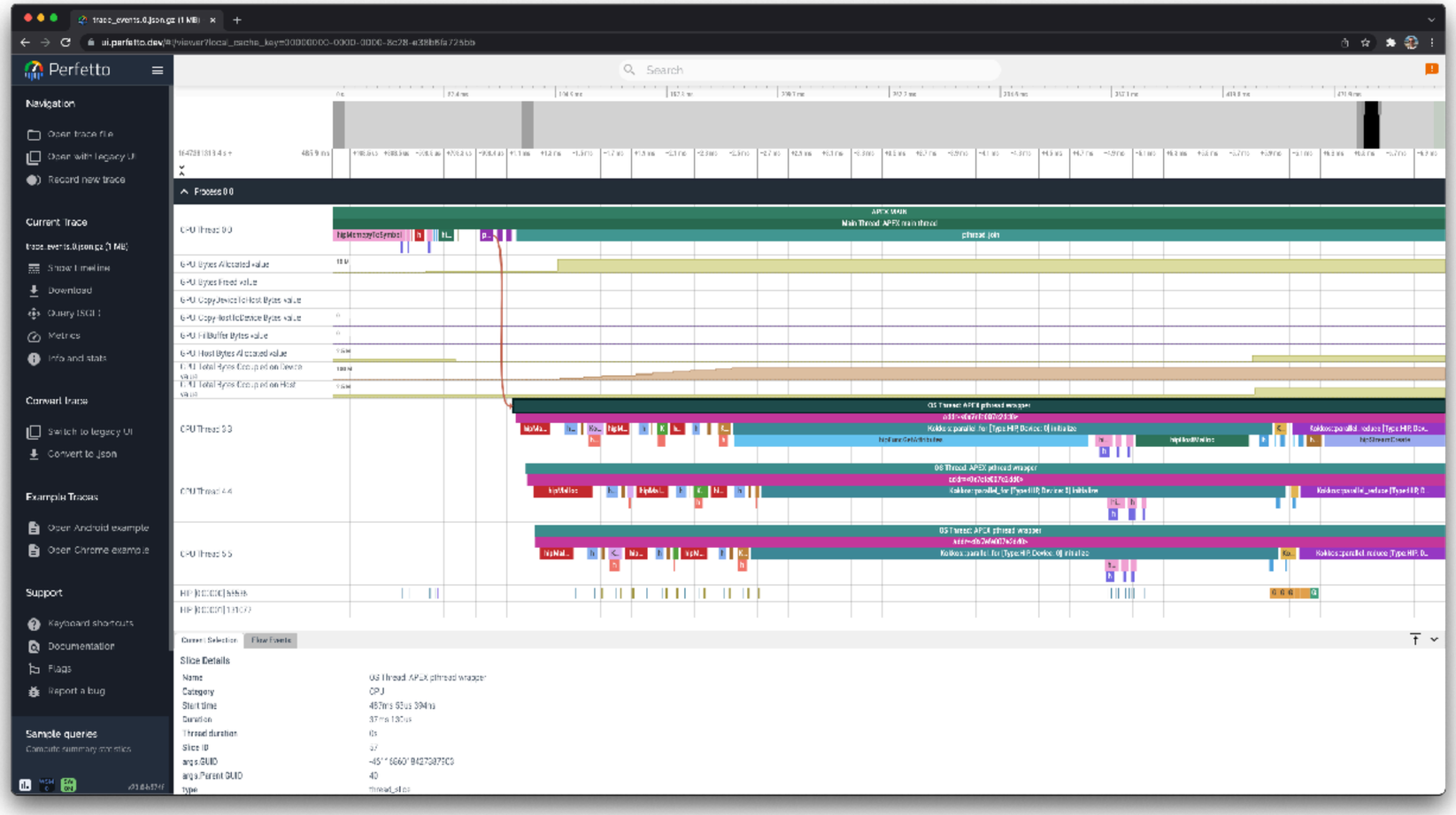
# Chrome Browser

```
% export TAU_MAKEFILE=$TAU/Makefile.tau-icpc-mpi-pdt
% make F90=tau_f90.sh
(Or edit Makefile and change F90=tau_f90.sh)

For Chrome:
% export TAU_TRACE=1
% srun -n 64   ./a.out

% tau_treemerge.pl
% tau_trace2json tau.trc tau.edf -chrome -ignoreatomic -o app.json
Copy app.json to your laptop and launch Chrome browser and in address:
chrome://tracing
Load -> app.json

Newer tool: Perfetto.dev also reads the same json file (or .json.gz)
```

# Chrome Browser

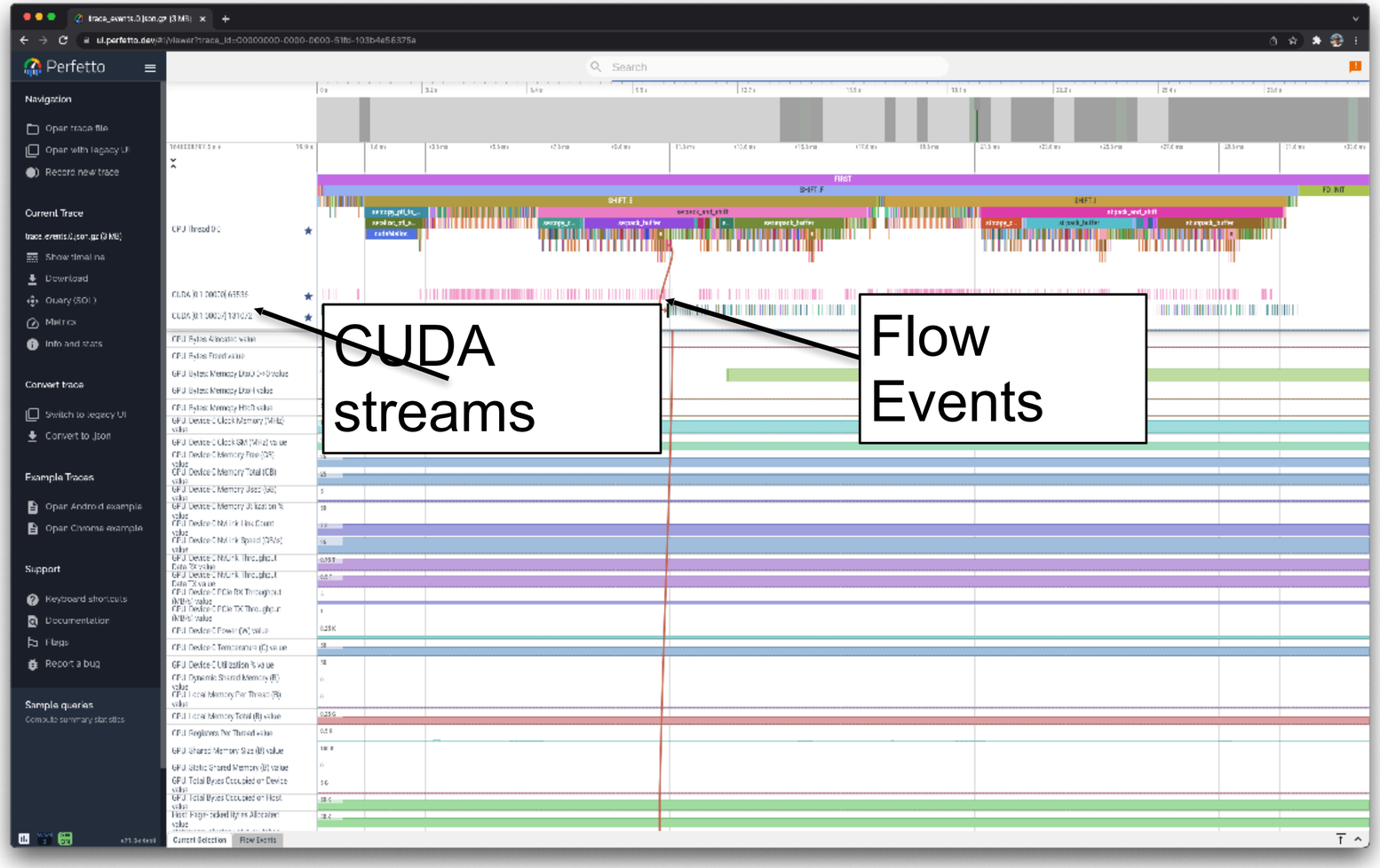


# Perfetto.dev Trace Browser: Kokkos




```
% export TAU_TRACE=1; mpirun -np 64 tau_exec ./a.out;
% tau_treemerge.pl;
% tau_trace2json tau.trc tau.edf -chrome -ignoreatomic -o app.json
```

# Perfetto.dev Trace Browser




```
% export TAU_TRACE=1; mpirun -np 64 tau_exec ./a.out;
% tau_treemerge.pl;
% tau_trace2json tau.trc tau.edf -chrome -ignoreatomic -o app.json
```

# TAU for Heterogeneous Measurement

Multiple performance perspectives

Integrate Host-GPU support in TAU measurement framework

- Enable use of each measurement approach
- Include use of PAPI and CUPTI
- Provide profiling and tracing support

Tutorial

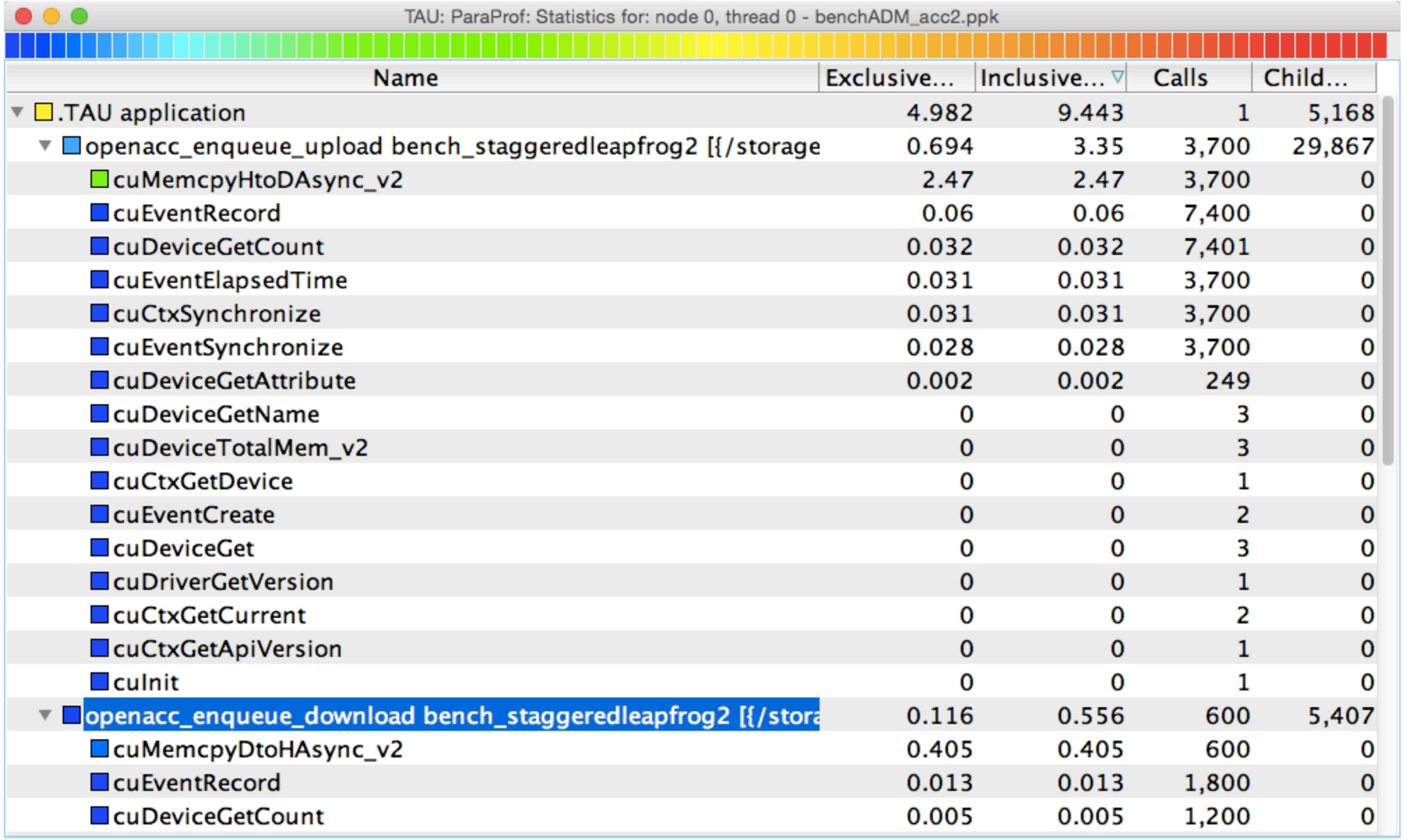
- Use TAU library wrapping of libraries
- Use `tau_exec` to work with binaries

  % ./a.out (uninstrumented)

  % tau_exec –T <configuration tags> –cupti ./a.out

  % paraprof

# OpenACC with PGI compilers




| Name | Exclusive... | Inclusive... ▽ | Calls | Child... |
|---|---|---|---|---|
| .TAU application | 4.982 | 9.443 | 1 | 5,168 |
| openacc_enqueue_upload bench_staggeredleapfrog2 [{/storage | 0.694 | 3.35 | 3,700 | 29,867 |
| cuMemcpyHtoDAsync_v2 | 2.47 | 2.47 | 3,700 | 0 |
| cuEventRecord | 0.06 | 0.06 | 7,400 | 0 |
| cuDeviceGetCount | 0.032 | 0.032 | 7,401 | 0 |
| cuEventElapsedTime | 0.031 | 0.031 | 3,700 | 0 |
| cuCtxSynchronize | 0.031 | 0.031 | 3,700 | 0 |
| cuEventSynchronize | 0.028 | 0.028 | 3,700 | 0 |
| cuDeviceGetAttribute | 0.002 | 0.002 | 249 | 0 |
| cuDeviceGetName | 0 | 0 | 3 | 0 |
| cuDeviceTotalMem_v2 | 0 | 0 | 3 | 0 |
| cuCtxGetDevice | 0 | 0 | 1 | 0 |
| cuEventCreate | 0 | 0 | 2 | 0 |
| cuDeviceGet | 0 | 0 | 3 | 0 |
| cuDriverGetVersion | 0 | 0 | 1 | 0 |
| cuCtxGetCurrent | 0 | 0 | 2 | 0 |
| cuCtxGetApiVersion | 0 | 0 | 1 | 0 |
| cuInit | 0 | 0 | 1 | 0 |
| openacc_enqueue_download bench_staggeredleapfrog2 [{/stora | 0.116 | 0.556 | 600 | 5,407 |
| cuMemcpyDtoHAsync_v2 | 0.405 | 0.405 | 600 | 0 |
| cuEventRecord | 0.013 | 0.013 | 1,800 | 0 |
| cuDeviceGetCount | 0.005 | 0.005 | 1,200 | 0 |

$ configure –c++=pgCC –cc=pgcc –fortran=pgi …
$ tau_exec –T pgi -openacc -cupti ./a.out

# Tracking OpenACC Data Transfers

TAU: ParaProf: Context Events for: node 0, thread 0 - benchADM_acc2.ppk

| Name | Total | NumSamples | MaxValue | MinValue | MeanValue | Std. Dev. |
|---|---|---|---|---|---|---|
| .TAU application | | | | | | |
| openacc_enqueue_upload bench_staggeredleapfrog2 [{/st | | | | | | |
| cuMemcpyHtoDAsync_v2 | | | | | | |
| [GROUP=MAX_MARKER] Bytes copied from Host to De | 512,000 | 1 | 512,000 | 512,000 | 512,000 | 0 |
| Bytes copied from Host to Device | 973,016,000 | 3,700 | 512,000 | 120 | 262,977.297 | 255,846.506 |
| openacc_enqueue_download bench_staggeredleapfrog2 [{ | | | | | | |
| cuMemcpyDtoHAsync_v2 | | | | | | |
| Bytes copied from Device to Host | 307,200,000 | 600 | 512,000 | 512,000 | 512,000 | 0 |
| Bytes copied from Device to Host | 307,200,000 | 600 | 512,000 | 512,000 | 512,000 | 0 |
| Bytes copied from Host to Device | 973,016,000 | 3,700 | 512,000 | 120 | 262,977.297 | 255,846.506 |
| [GROUP=MAX_MARKER] Bytes copied from Host to Device | 512,000 | 1 | 512,000 | 512,000 | 512,000 | 0 |

% configure –c++=pgCC –cc=pgcc –fortran=pgi …
% tau_exec –T pgi -openacc -cupti ./a.out
Context events show extent of variation

# TAU Architecture and Workflow

Instrumentation: Add probes to perform measurements

- Source code instrumentation using pre-processors and compiler scripts
- Wrapping external libraries (I/O, MPI, Memory, CUDA, OpenCL, pthread)
- Rewriting the binary executable

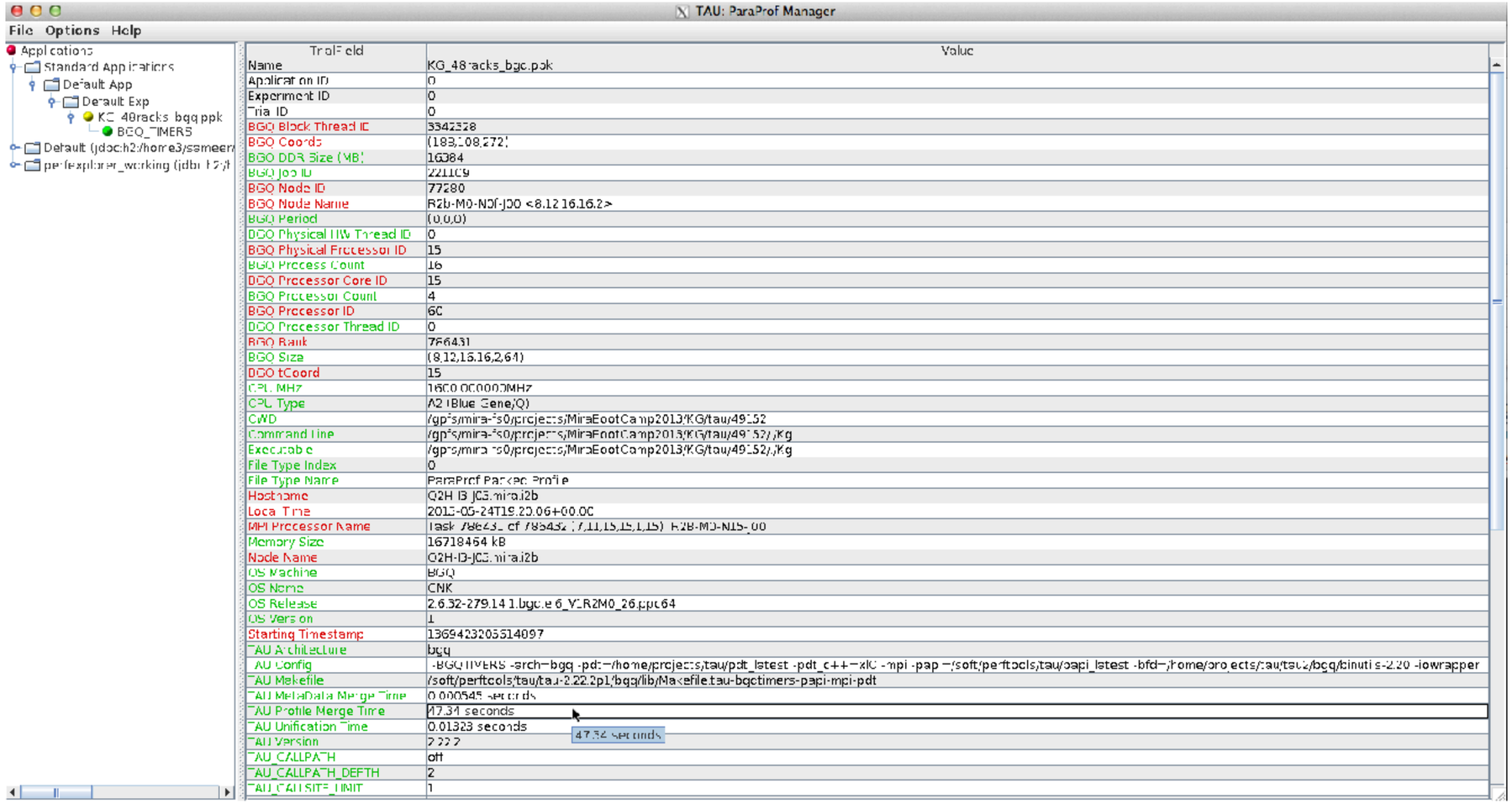
Measurement: Profiling or tracing using various metrics

- Direct instrumentation (Interval events measure exclusive or inclusive duration)
- Indirect instrumentation (Sampling measures statement level contribution)
- Throttling and runtime control of low-level events that execute frequently
- Per-thread storage of performance data
- Interface with external packages (e.g. PAPI hw performance counter library)

Analysis: Visualization of profiles and traces

- 3D visualization of profile data in paraprof or perfexplorer tools
- Trace conversion & display in external visualizers (Vampir, Jumpshot, ParaVer)

# TAU's Runtime Merging of Profile Data




$ export TAU_PROFILE_FORMAT=merged
It took ~48 seconds to merge and write profiles from 786,432 ranks

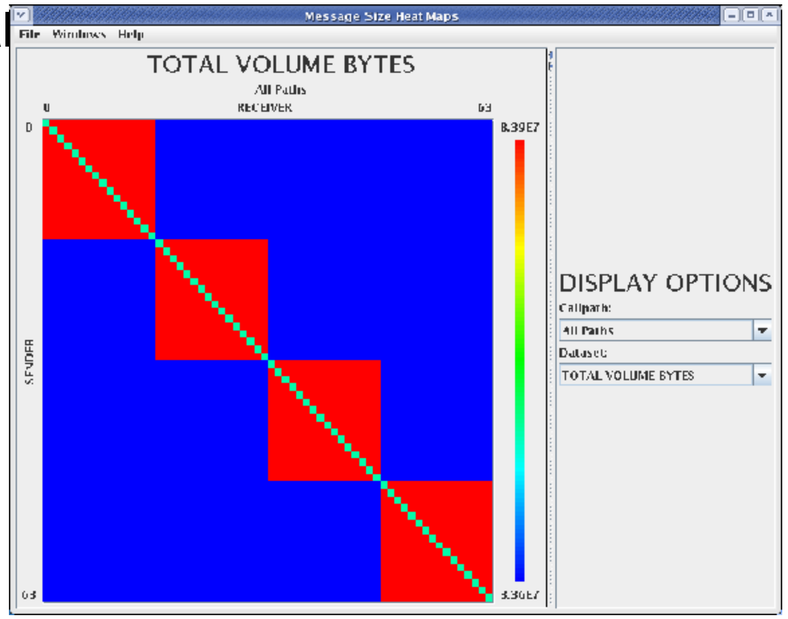
# Communication Matrix

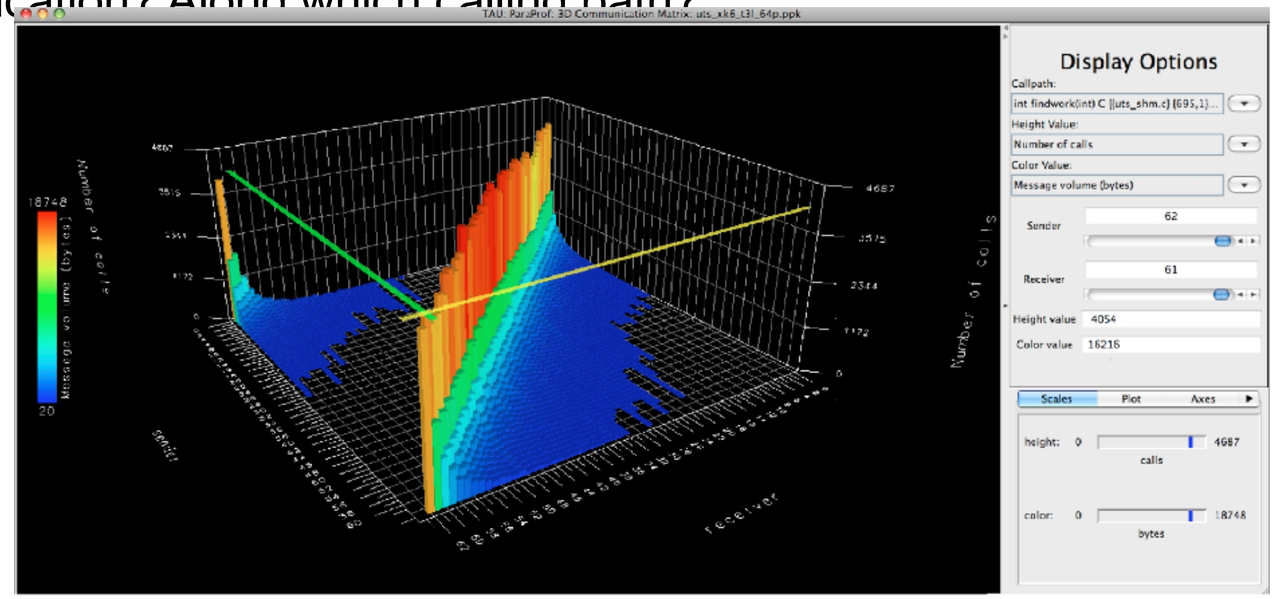
# Generating Communication Matrix

```
% export TAU_MAKEFILE=$TAU_MAKEFILE_BASE-icpc-papi-mpi-pdt
% make F90=tau_f90.sh
(Or edit Makefile and change F90=tau_f90.sh)

% export TAU_COMM_MATRIX=1
% mpirun -np 4    ./a.out

% paraprof
(Windows -> Communication Matrix)
(Windows -> 3D Communication Matrix)
```

# Communication Matrix Display

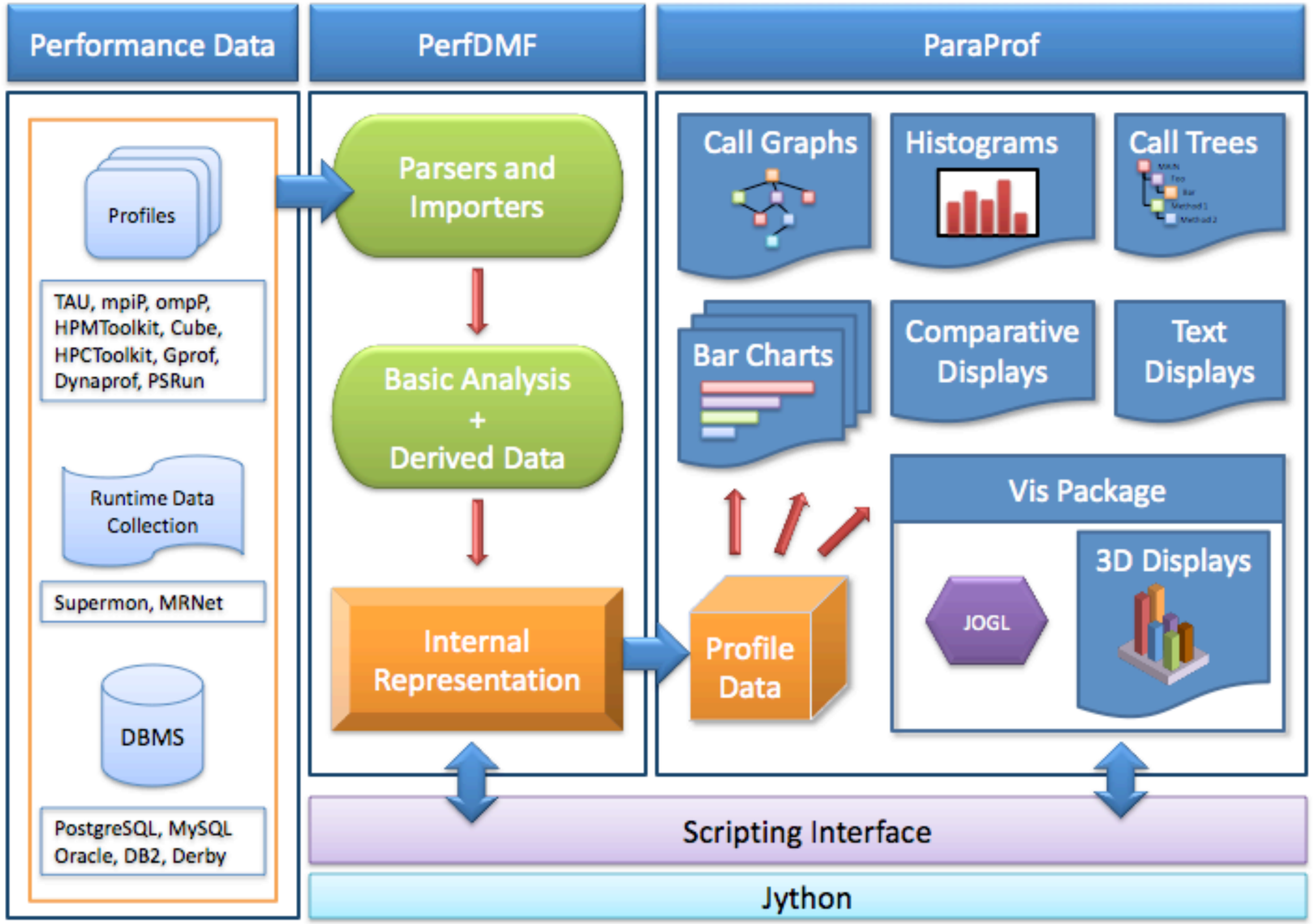
Goal: communication? Along which calling path?



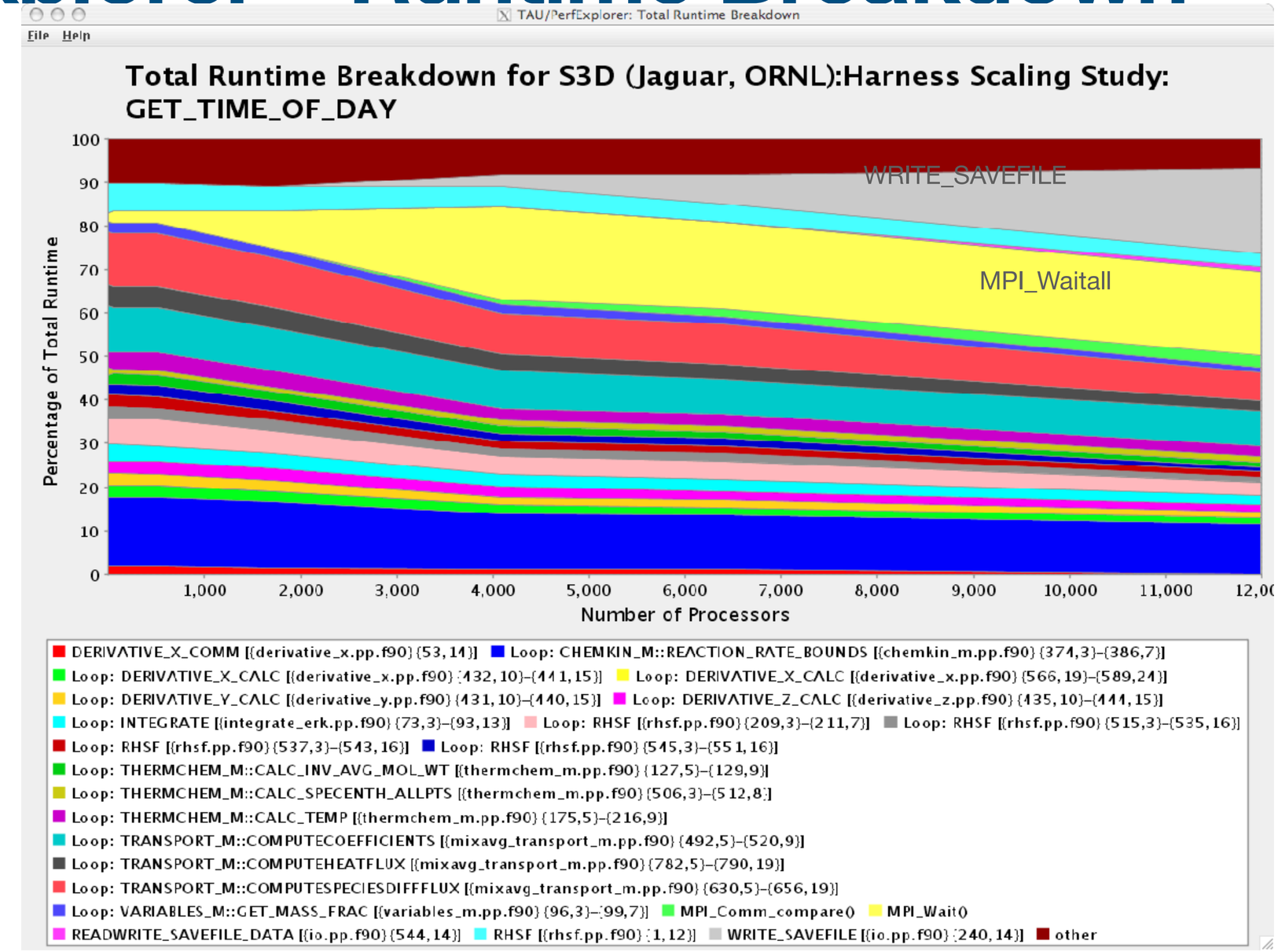



# Perf Explorer

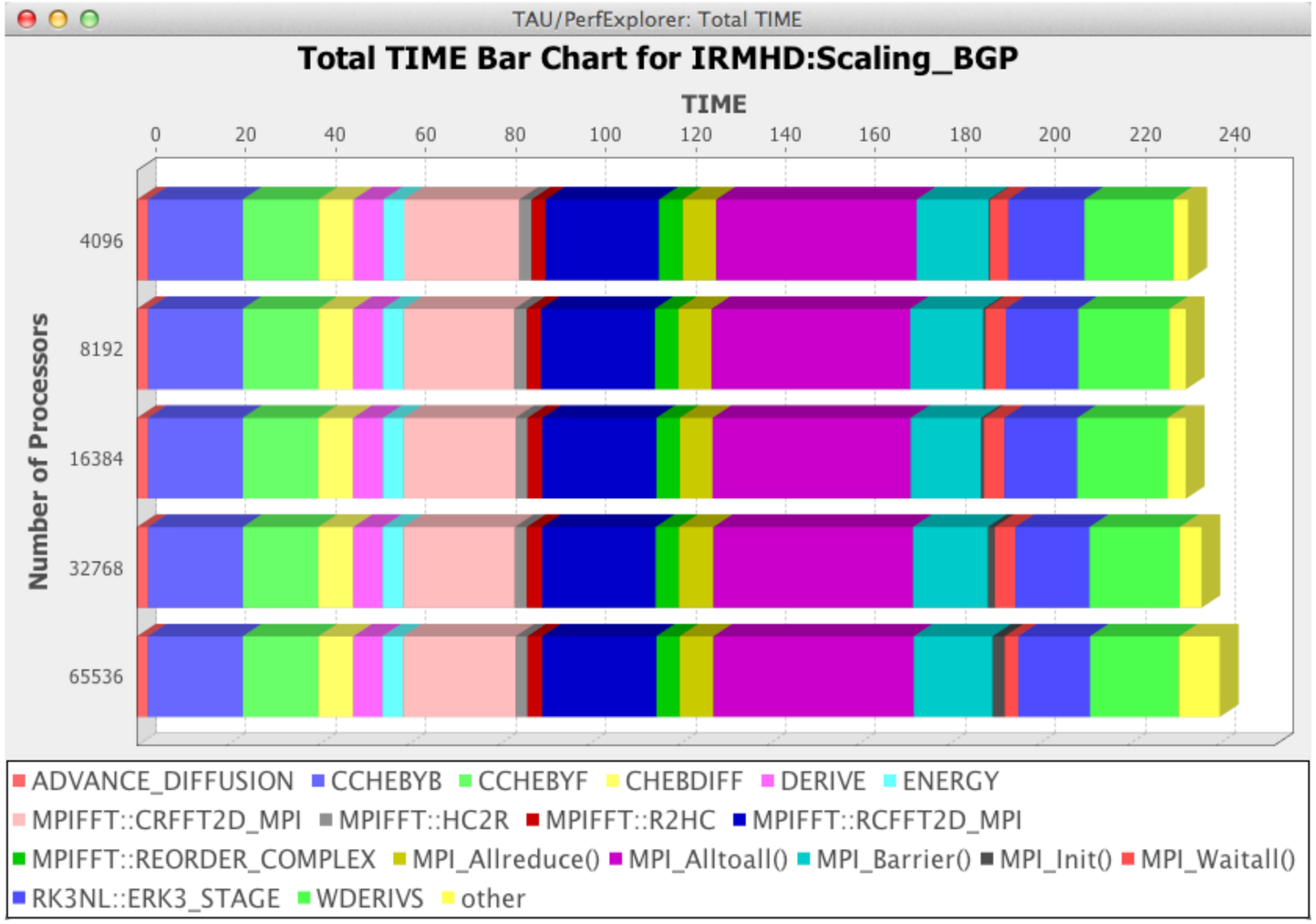
# ParaProf Profile Analysis Framework



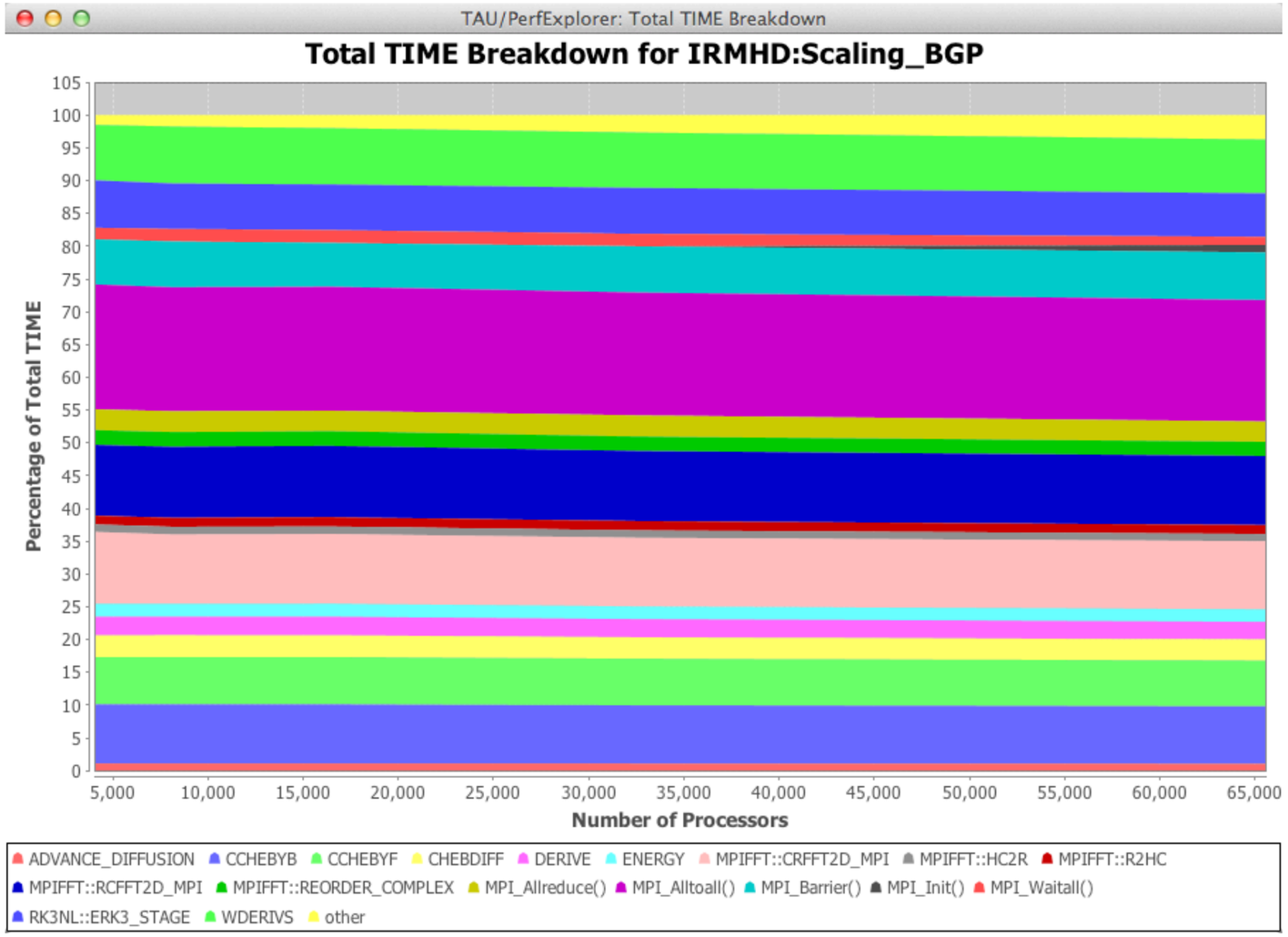
# PerfExplorer – Runtime Breakdown



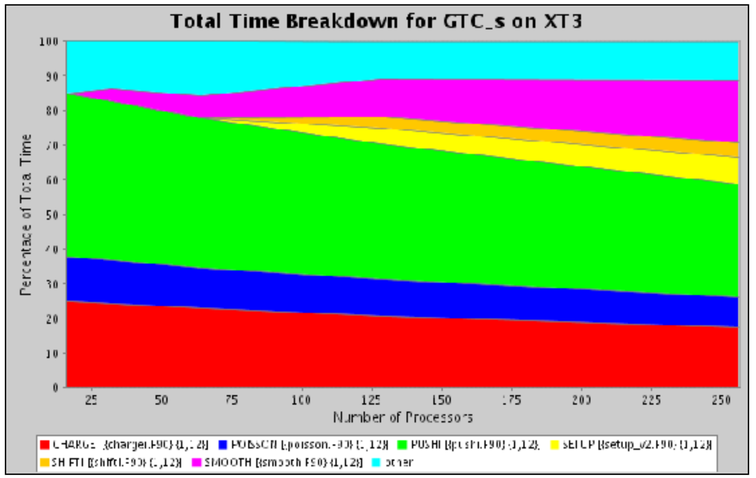
# Evaluate Scalability

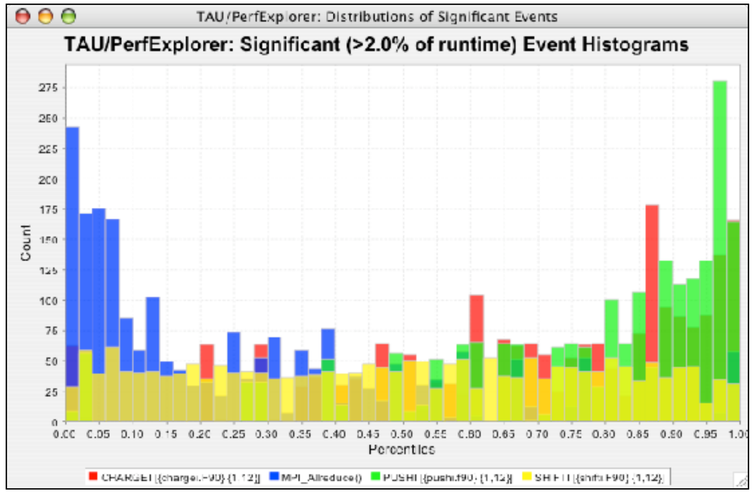


# Runtime Breakdown

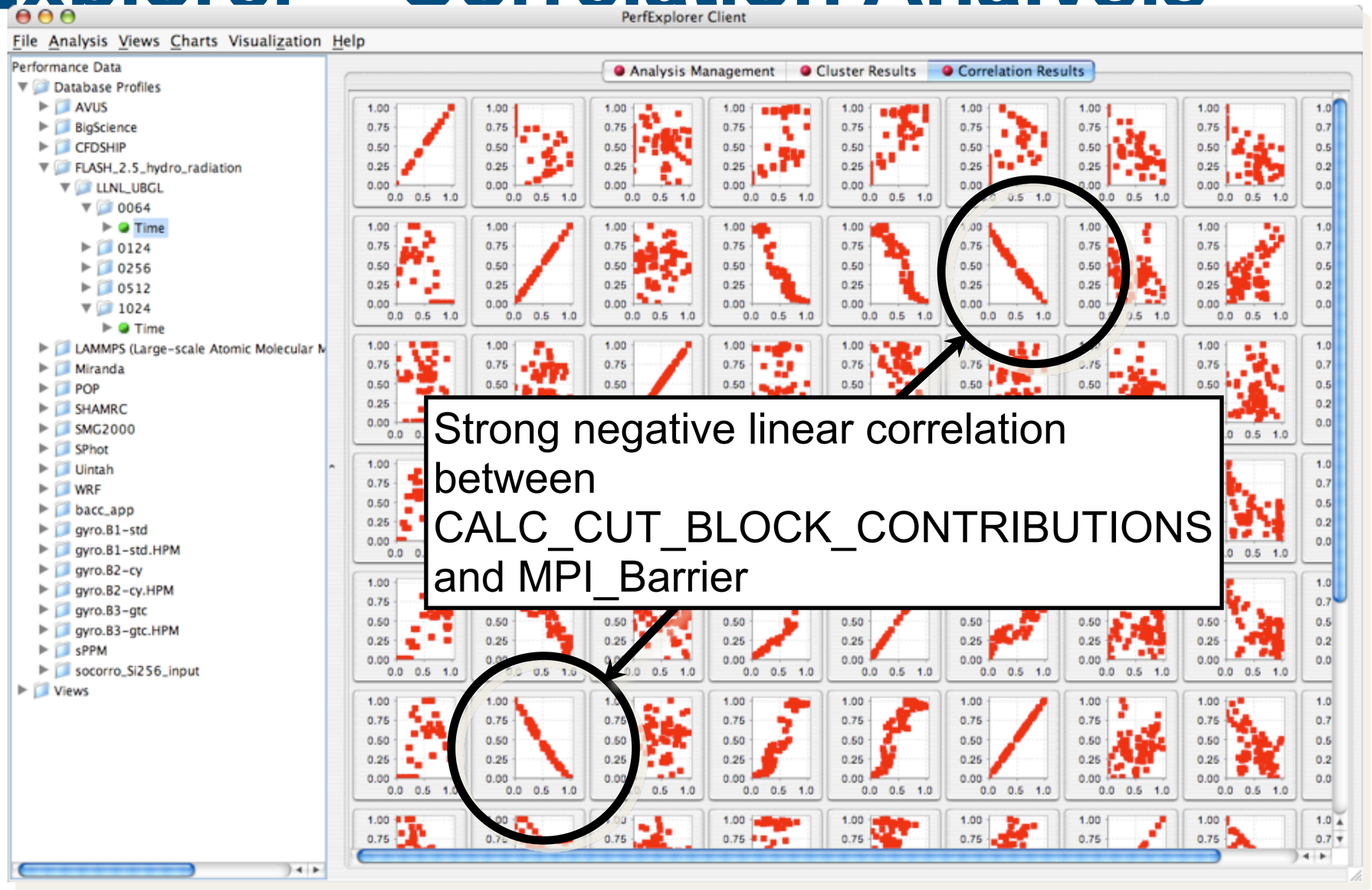


# PerfExplorer – Relative Comparisons



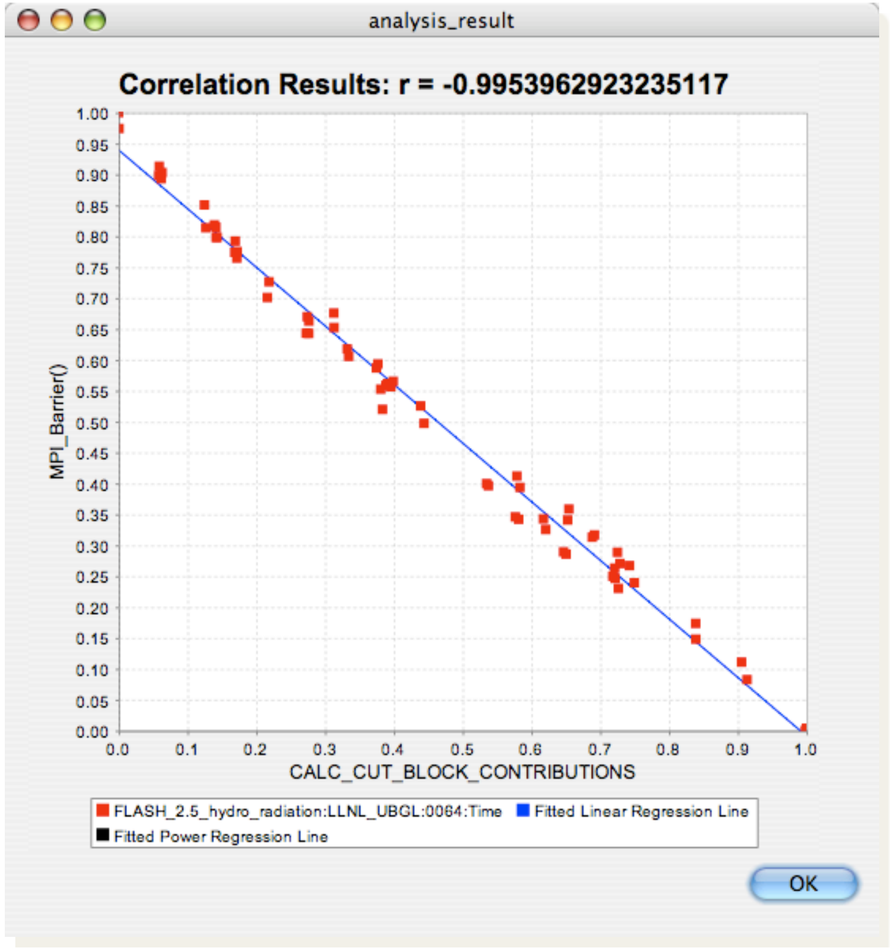



# PerfExplorer – Correlation Analysis

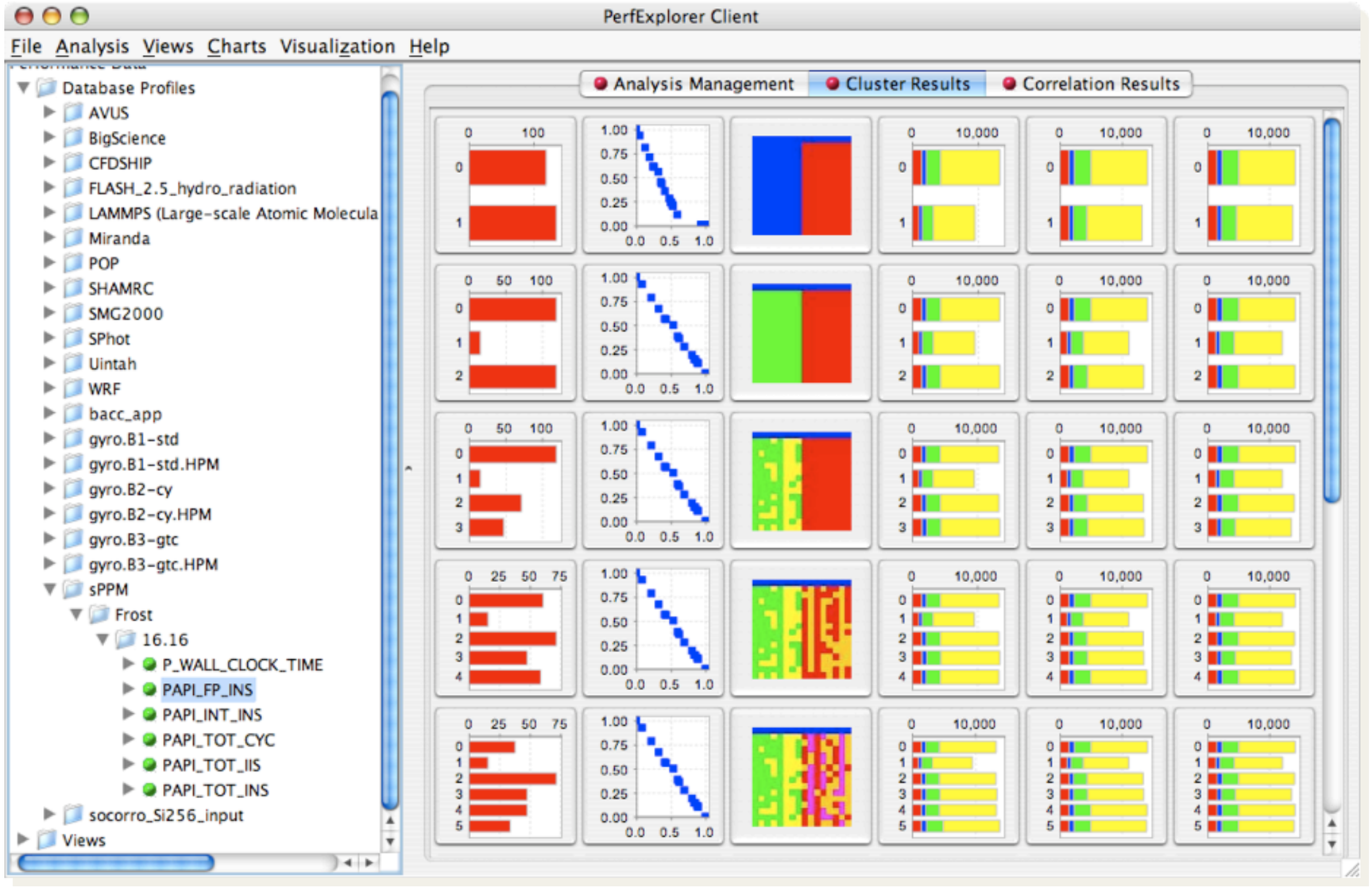



Data: FLASH on BGL(LLNL), 64 nodes

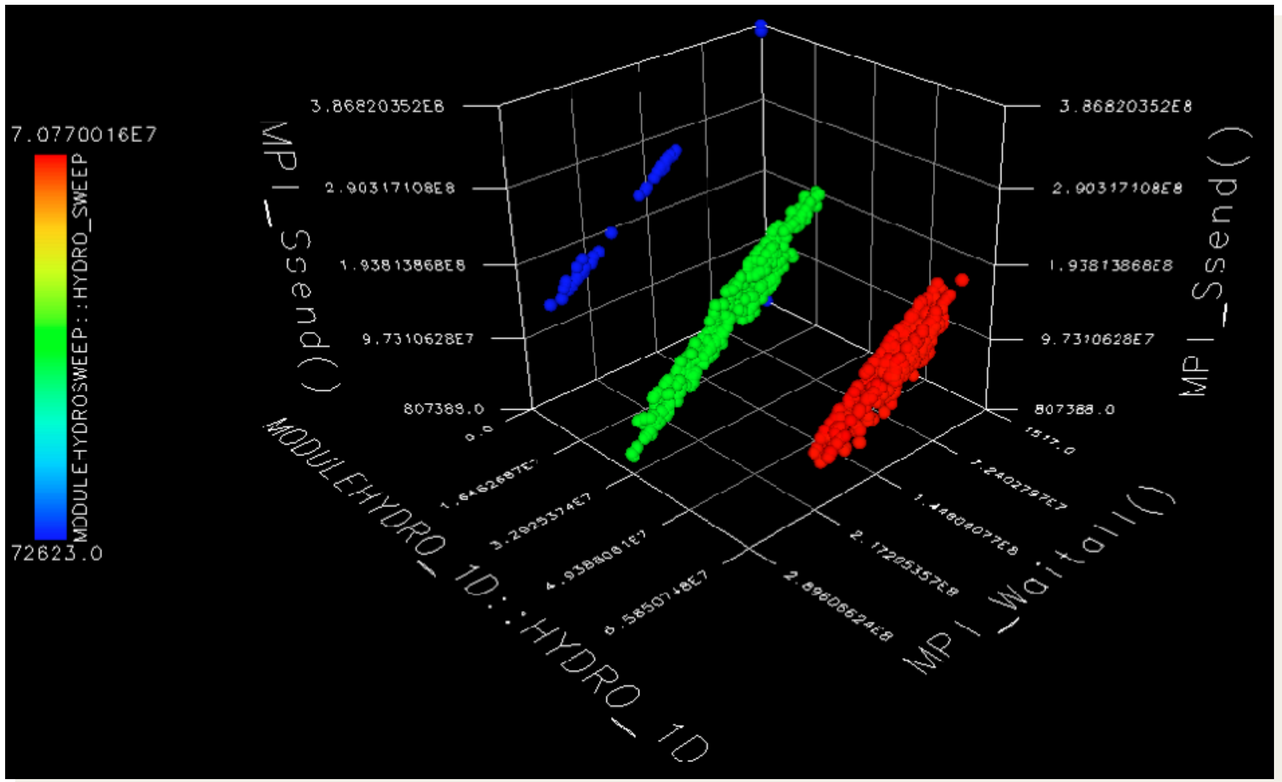
# PerfExplorer – Correlation Analysis



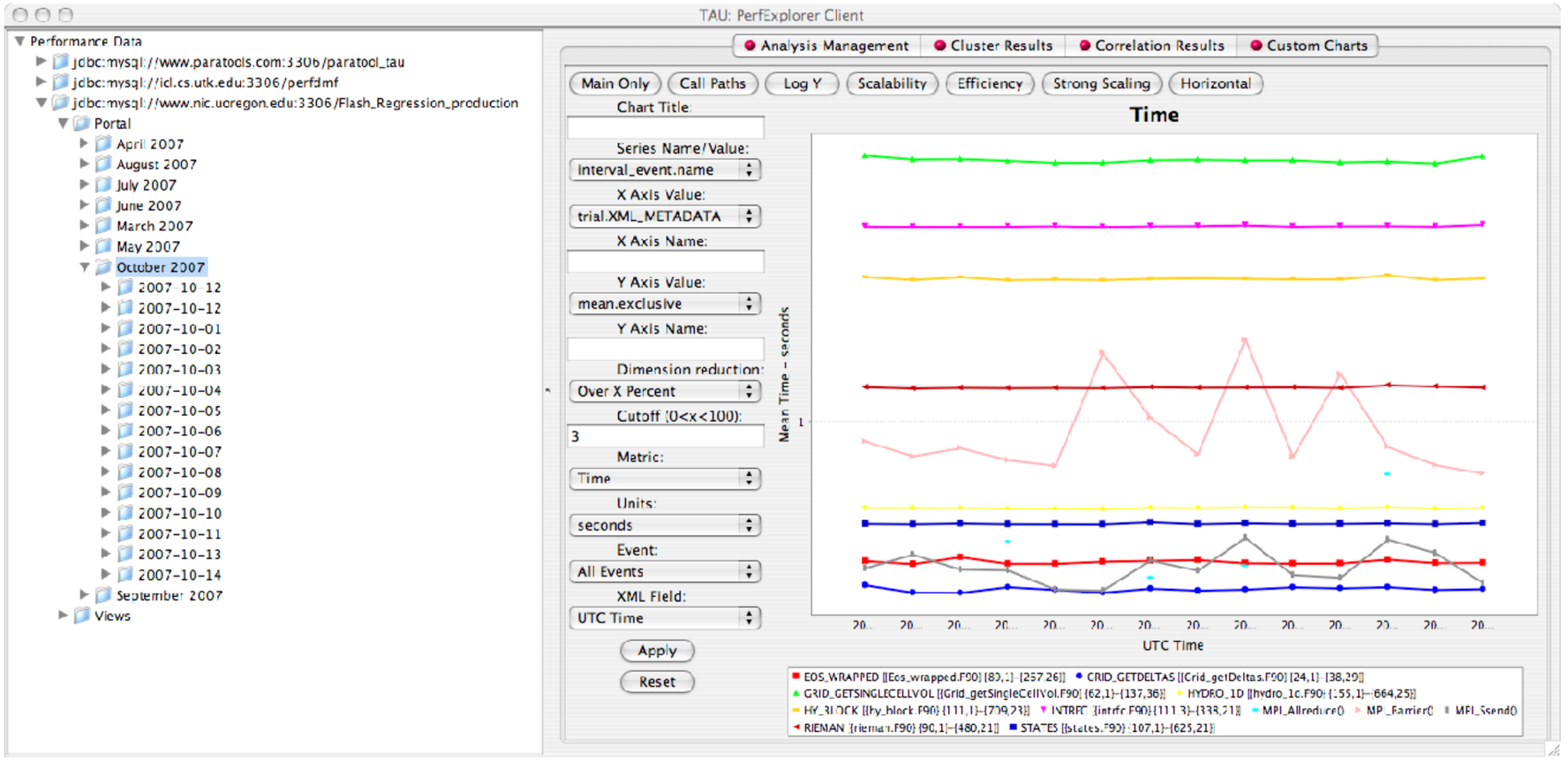
# PerfExplorer – Cluster Analysis



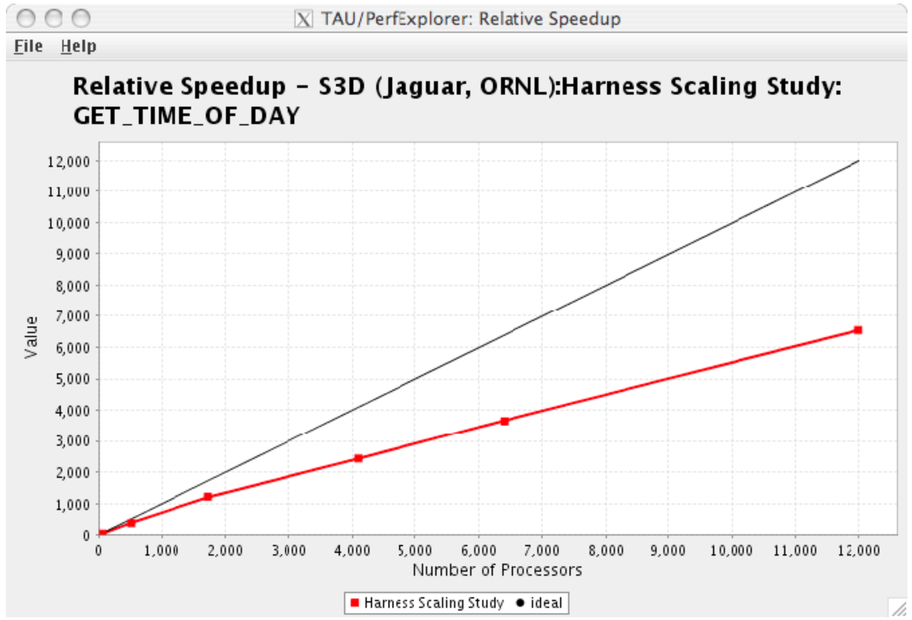
# PerfExplorer – Cluster Analysis

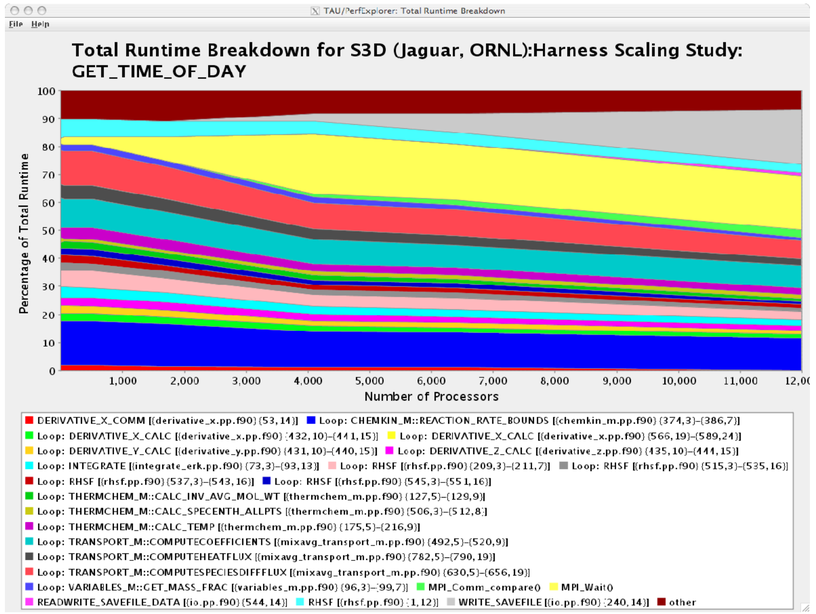


# PerfExplorer – Performance Regression

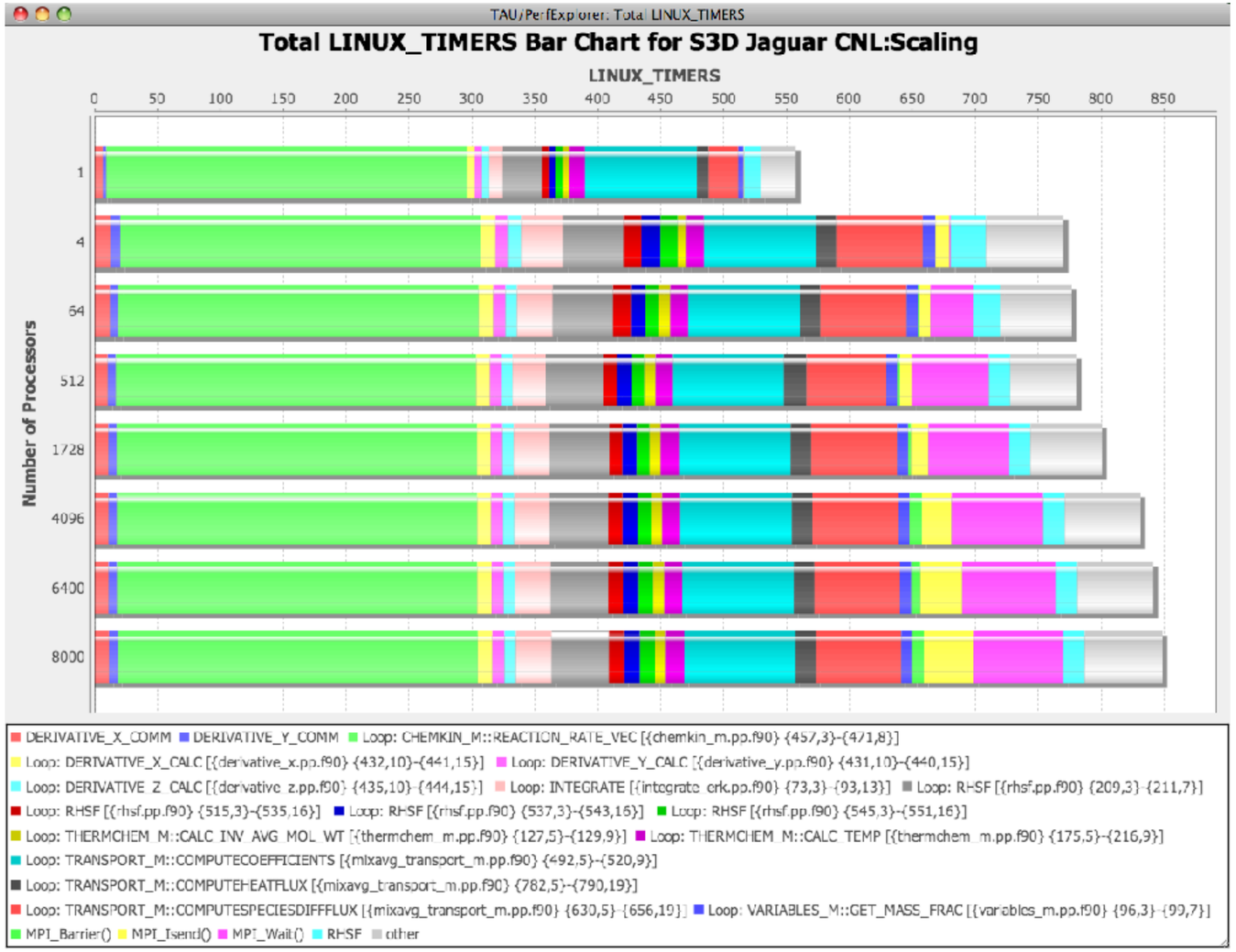


# Evaluate Scalability

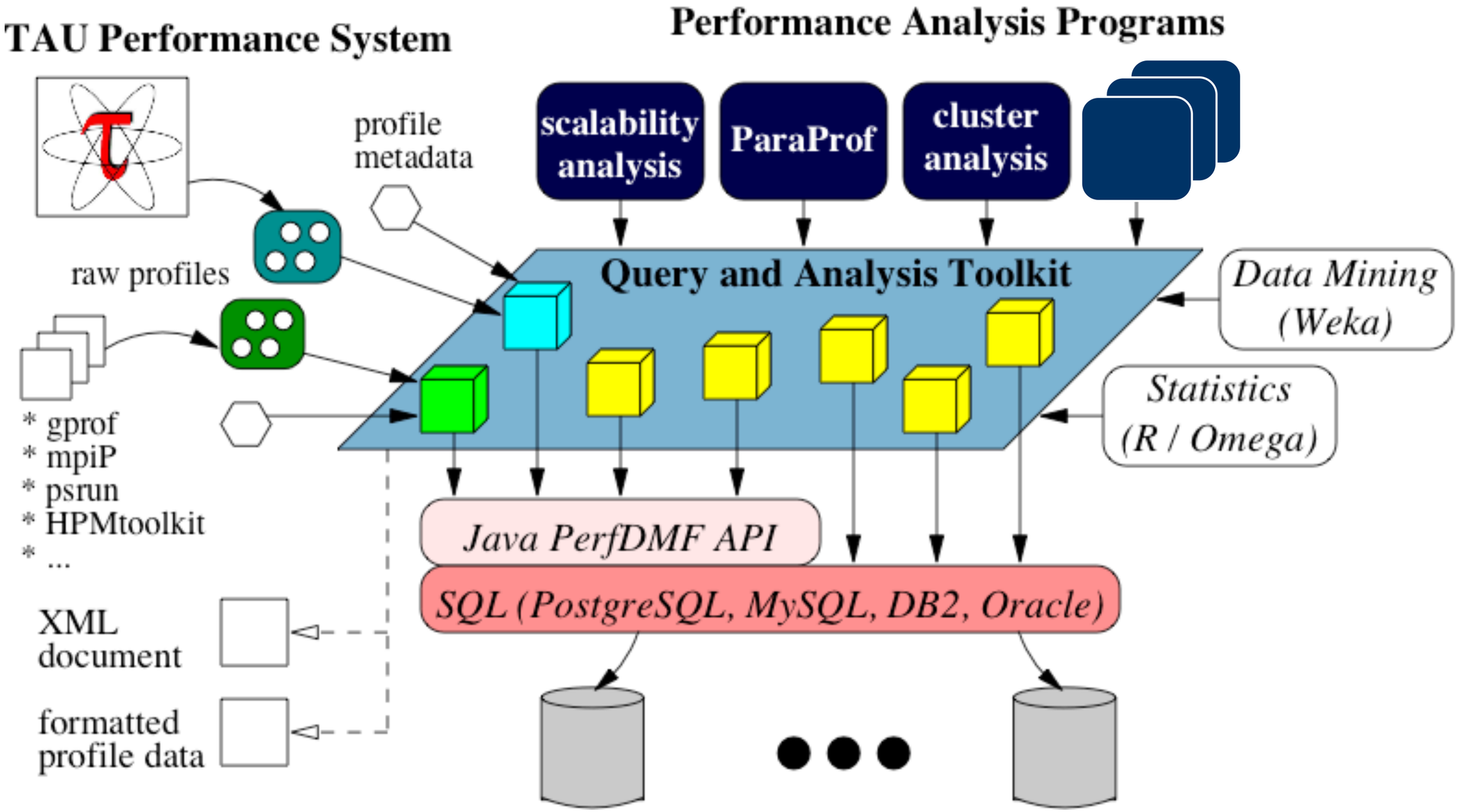


# Usage Scenarios: Evaluate Scalability



# TAUdb: Framework for Managing Performance Data

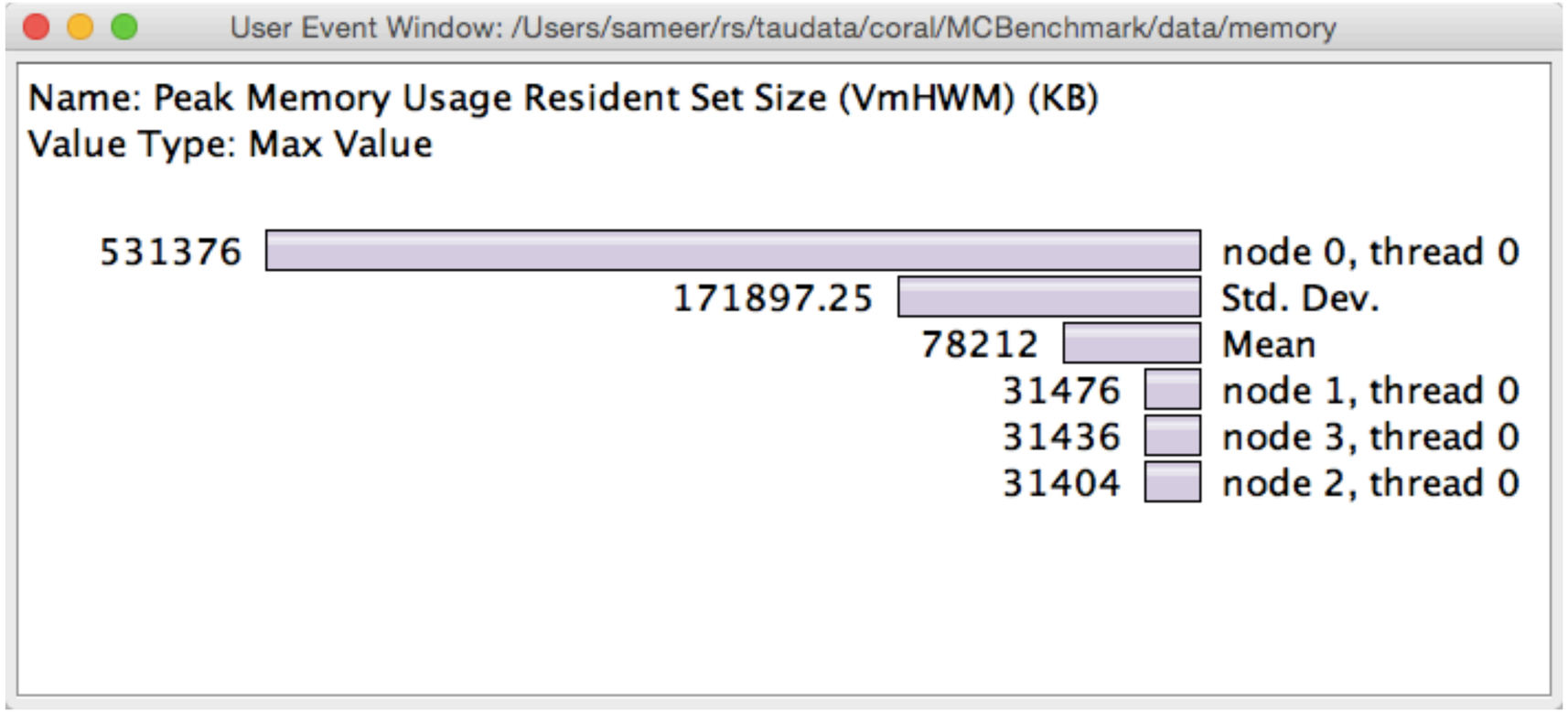


# Evaluate Scalability using PerfExplorer Charts

```
% export TAU_MAKEFILE=$TAU_MAKEFILE_BASE-icpc-papi-mpi-pdt
% make F90=tau_f90.sh
(Or edit Makefile and change F90=tau_f90.sh)
% qsub  run1p.job
% paraprof --pack 1p.ppk
% qsub  run2p.job …
% paraprof --pack 2p.ppk … and so on.
On your client:
% taudb_configure --create-default
% perfexplorer_configure
(Enter, y to load schema, defaults)
% paraprof
(load each trial: DB -> Add Trial -> Type (Paraprof Packed
Profile) -> OK, OR use taudb_loadtrial on the commandline)
% taudb_loadtrial -a App -x MyExp -n 4p 4p.ppk
% perfexplorer
(Charts -> Speedup)
OR:
wget http://tau.uoregon.edu/data.tgz; cat README in data
```

# Debugging and Memory

# Measuring Memory Footprint

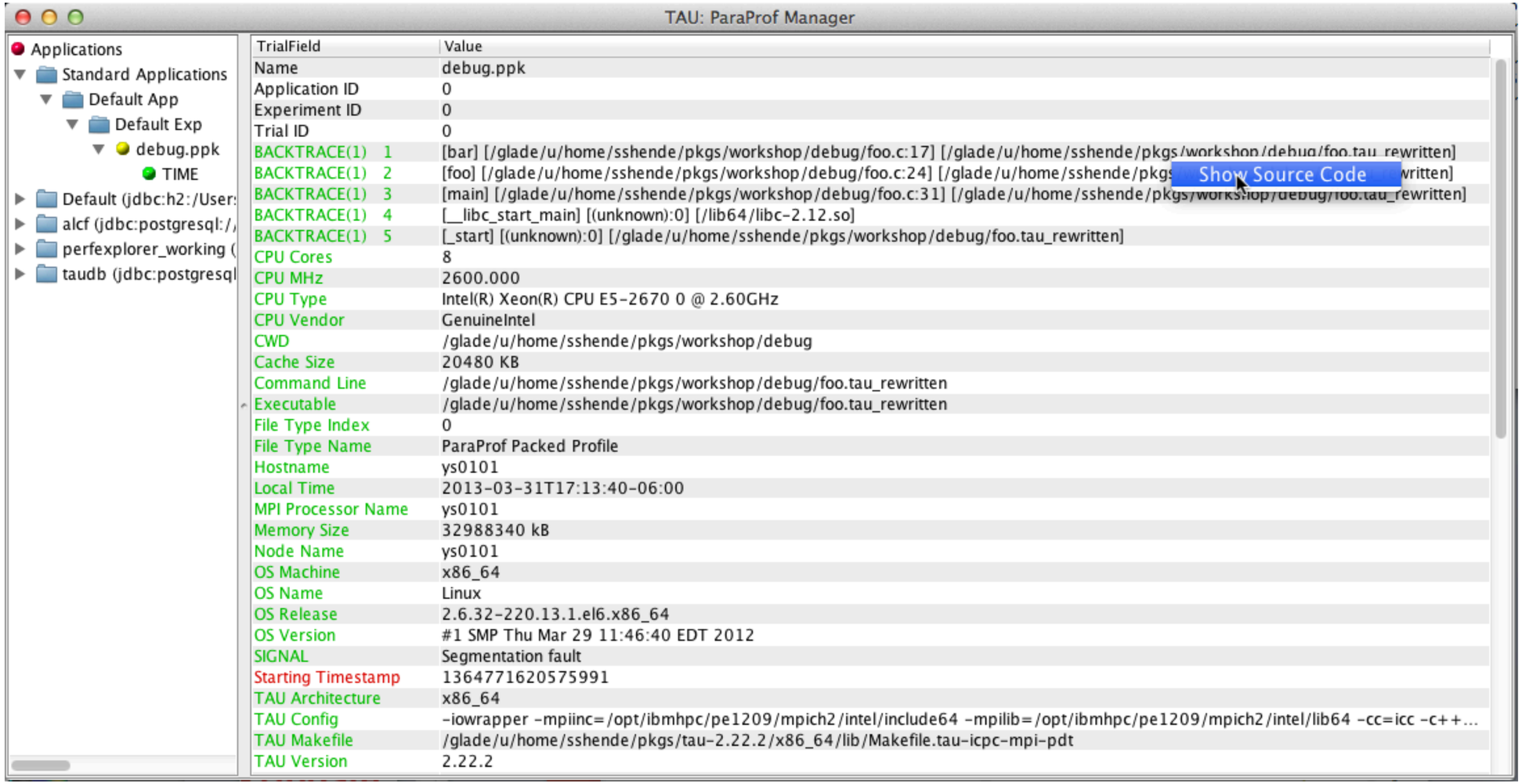



% export TAU_TRACK_MEMORY_FOOTPRINT=1
Paraprof:
Right click on a node -> Show Context Event Window -> see memory events

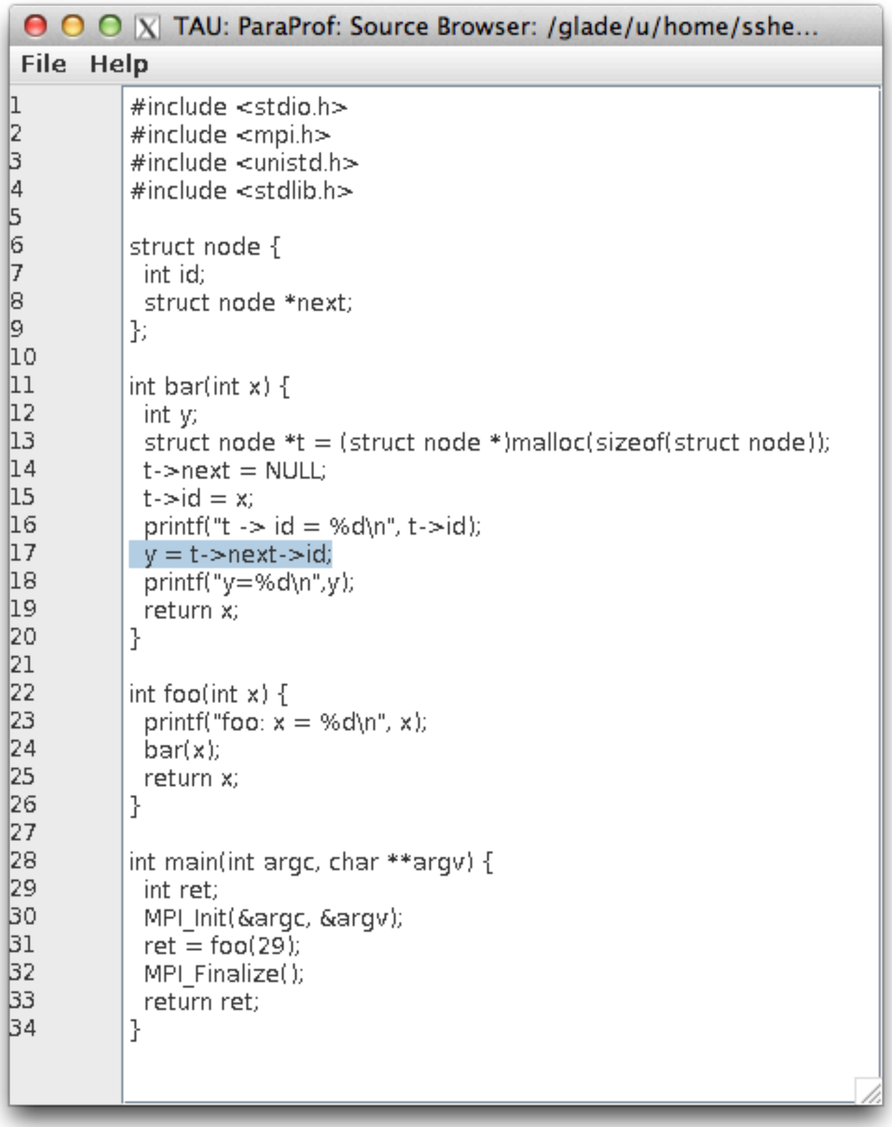
# Application Debugging

```
% export TAU_MAKEFILE=$TAU_MAKEFILE_BASE-icpc-papi-mpi-pdt
% export TAU_OPTIONS='-optMemDbg -optVerbose'
% make F90=tau_f90.sh CC=tau_cc.sh CXX=tau_cxx.sh

% export TAU_MEMDBG_PROTECT_ABOVE=1
% export TAU_MEMDBG_PROTECT_BELOW=1
% export TAU_MEMDBG_PROTECT_FREE=1
% mpirun -np 4   ./matmult
% paraprof
```

# Application « Debugging »



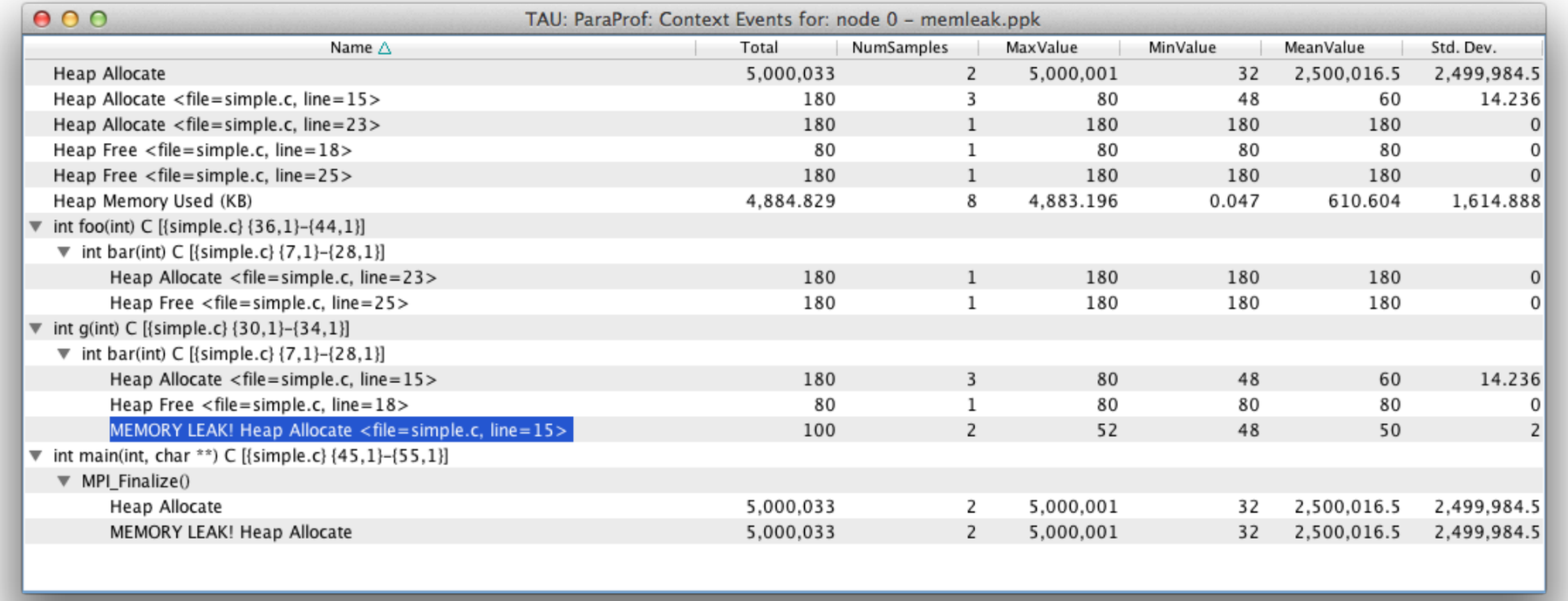
# Location of segmentation violation



# Memory Leak Detection

```
% export TAU_MAKEFILE=$TAU_MAKEFILE_BASE-icpc-papi-mpi-pdt
% export TAU_OPTIONS='-optMemDbg -optVerbose'
% make F90=tau_f90.sh CC=tau_cc.sh CXX=tau_cxx.sh


% export TAU_TRACK_MEMORY_LEAKS=1
% mpirun -np 4   ./matmult
% paraprof
```

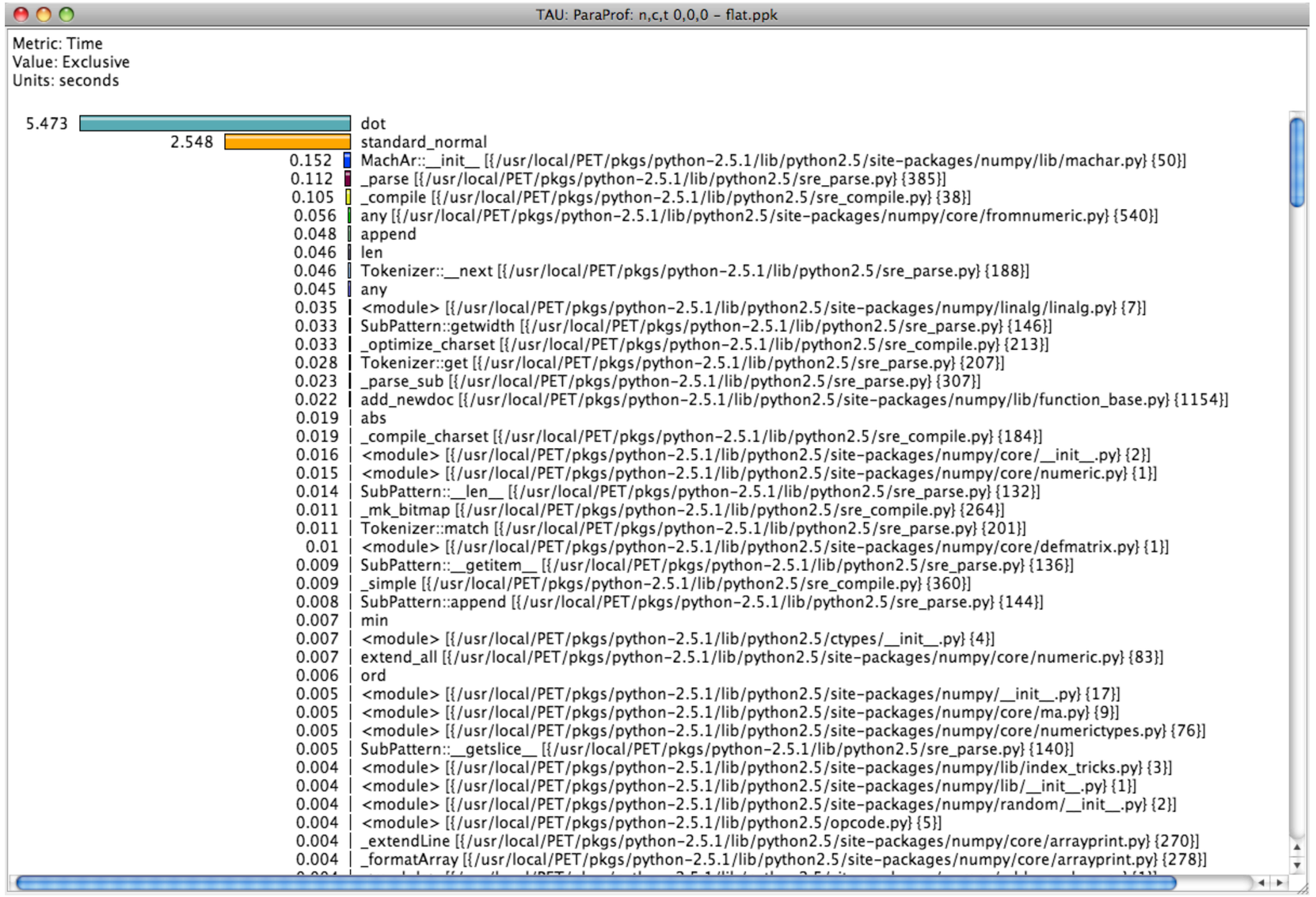
# Memory Leak Detection



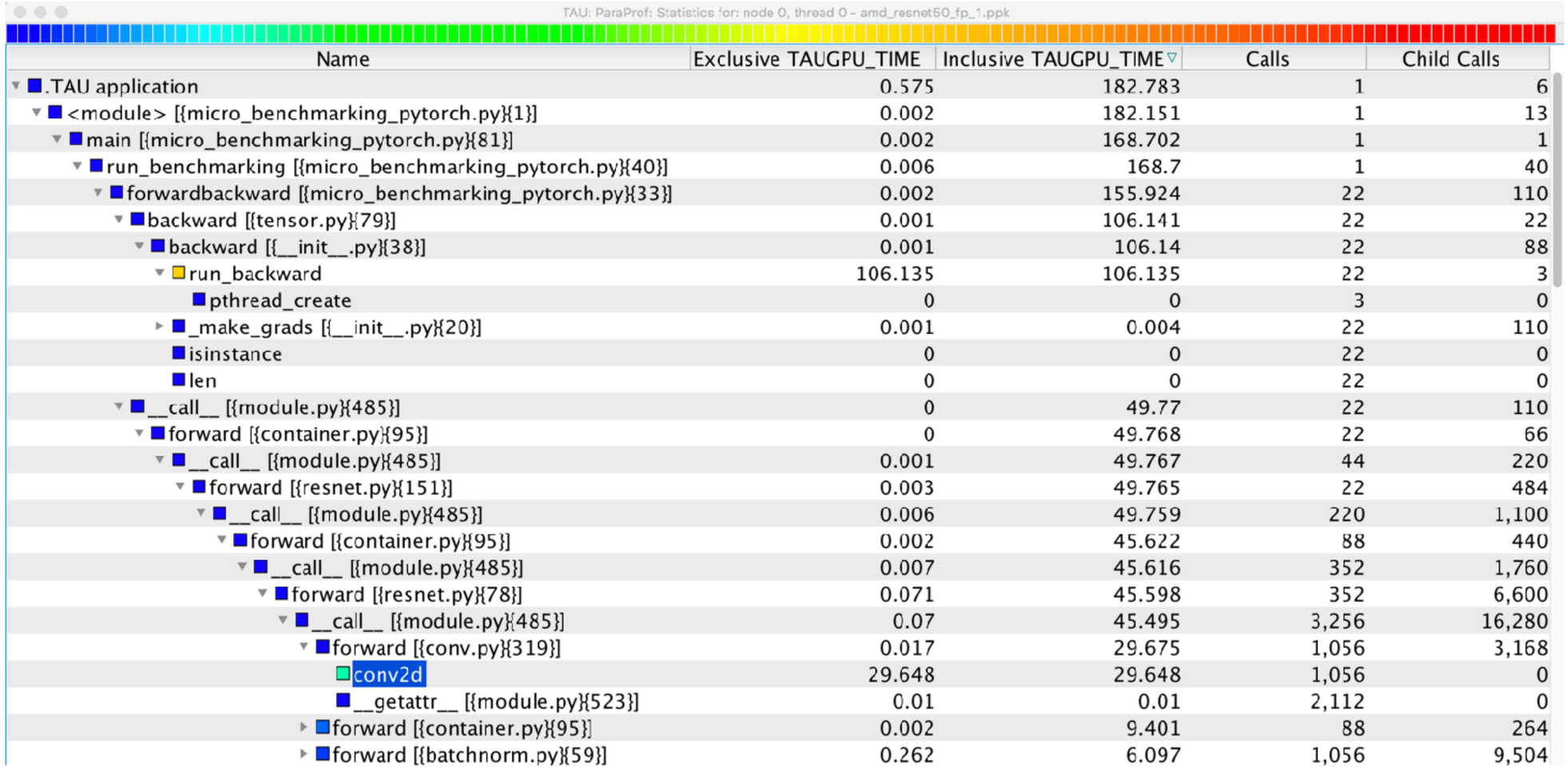

| Name △ | Total | NumSamples | MaxValue | MinValue | MeanValue | Std. Dev. |
|---|---|---|---|---|---|---|
| Heap Allocate | 5,000,033 | 2 | 5,000,001 | 32 | 2,500,016.5 | 2,499,984.5 |
| Heap Allocate <file=simple.c, line=15> | 180 | 3 | 80 | 48 | 60 | 14.236 |
| Heap Allocate <file=simple.c, line=23> | 180 | 1 | 180 | 180 | 180 | 0 |
| Heap Free <file=simple.c, line=18> | 80 | 1 | 80 | 80 | 80 | 0 |
| Heap Free <file=simple.c, line=25> | 180 | 1 | 180 | 180 | 180 | 0 |
| Heap Memory Used (KB) | 4,884.829 | 8 | 4,883.196 | 0.047 | 610.604 | 1,614.888 |
| ▼ int foo(int) C [{simple.c} {36,1}-{44,1}] | | | | | | |
| ▼ int bar(int) C [{simple.c} {7,1}-{28,1}] | | | | | | |
| Heap Allocate <file=simple.c, line=23> | 180 | 1 | 180 | 180 | 180 | 0 |
| Heap Free <file=simple.c, line=25> | 180 | 1 | 180 | 180 | 180 | 0 |
| ▼ int g(int) C [{simple.c} {30,1}-{34,1}] | | | | | | |
| ▼ int bar(int) C [{simple.c} {7,1}-{28,1}] | | | | | | |
| Heap Allocate <file=simple.c, line=15> | 180 | 3 | 80 | 48 | 60 | 14.236 |
| Heap Free <file=simple.c, line=18> | 80 | 1 | 80 | 80 | 80 | 0 |
| MEMORY LEAK! Heap Allocate <file=simple.c, line=15> | 100 | 2 | 52 | 48 | 50 | 2 |
| ▼ int main(int, char **) C [{simple.c} {45,1}-{55,1}] | | | | | | |
| ▼ MPI_Finalize() | | | | | | |
| Heap Allocate | 5,000,033 | 2 | 5,000,001 | 32 | 2,500,016.5 | 2,499,984.5 |
| MEMORY LEAK! Heap Allocate | 5,000,033 | 2 | 5,000,001 | 32 | 2,500,016.5 | 2,499,984.5 |

# Instrument Python

# Profiling Python using tau_python



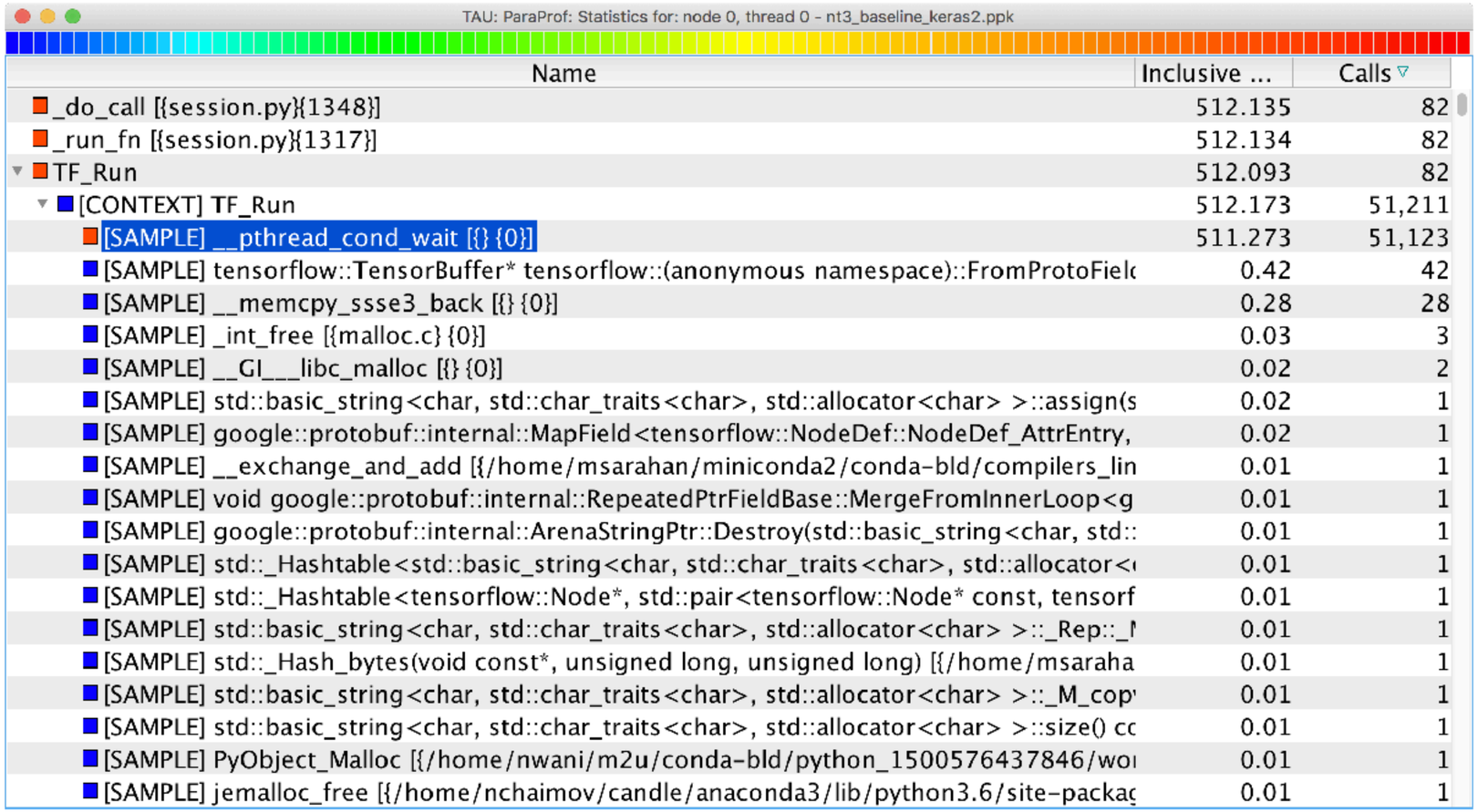
# Python Instrumentation




| Name | Exclusive TAUGPU_TIME | Inclusive TAUGPU_TIME | Calls | Child Calls |
|---|---|---|---|---|
| .TAU application | 0.575 | 182.783 | 1 | 6 |
| <module> [{micro_benchmarking_pytorch.py}{1}] | 0.002 | 182.151 | 1 | 13 |
| main [{micro_benchmarking_pytorch.py}{81}] | 0.002 | 168.702 | 1 | 1 |
| run_benchmarking [{micro_benchmarking_pytorch.py}{40}] | 0.006 | 168.7 | 1 | 40 |
| forwardbackward [{micro_benchmarking_pytorch.py}{33}] | 0.002 | 155.924 | 22 | 110 |
| backward [{tensor.py}{79}] | 0.001 | 106.141 | 22 | 22 |
| backward [{__init__.py}{38}] | 0.001 | 106.14 | 22 | 88 |
| run_backward | 106.135 | 106.135 | 22 | 3 |
| pthread_create | 0 | 0 | 3 | 0 |
| _make_grads [{__init__.py}{20}] | 0.001 | 0.004 | 22 | 110 |
| isinstance | 0 | 0 | 22 | 0 |
| len | 0 | 0 | 22 | 0 |
| __call__ [{module.py}{485}] | 0 | 49.77 | 22 | 110 |
| forward [{container.py}{95}] | 0 | 49.768 | 22 | 66 |
| __call__ [{module.py}{485}] | 0.001 | 49.767 | 44 | 220 |
| forward [{resnet.py}{151}] | 0.003 | 49.765 | 22 | 484 |
| __call__ [{module.py}{485}] | 0.006 | 49.759 | 220 | 1,100 |
| forward [{container.py}{95}] | 0.002 | 45.622 | 88 | 440 |
| __call__ [{module.py}{485}] | 0.007 | 45.616 | 352 | 1,760 |
| forward [{resnet.py}{78}] | 0.071 | 45.598 | 352 | 6,600 |
| __call__ [{module.py}{485}] | 0.07 | 45.495 | 3,256 | 16,280 |
| forward [{conv.py}{319}] | 0.017 | 29.675 | 1,056 | 3,168 |
| conv2d | 29.648 | 29.648 | 1,056 | 0 |
| __getattr__ [{module.py}{523}] | 0.01 | 0.01 | 2,112 | 0 |
| forward [{container.py}{95}] | 0.002 | 9.401 | 88 | 264 |
| forward [{batchnorm.py}{59}] | 0.262 | 6.097 | 1,056 | 9,504 |

$ tau_python ./foo.py

# Identifying Wait States Using EBS

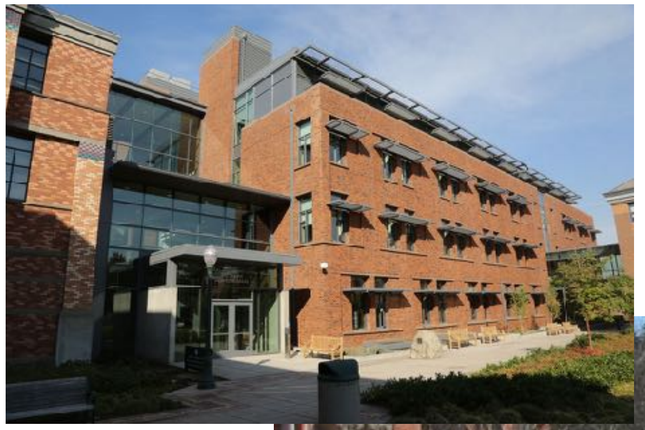


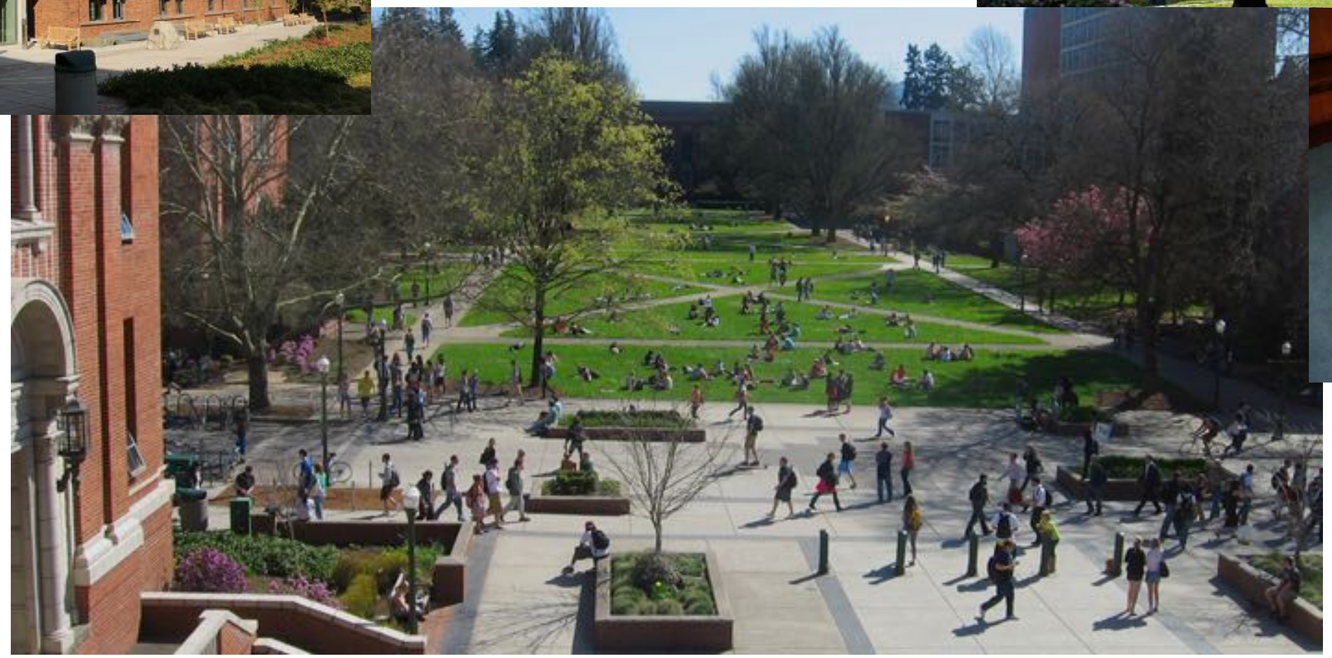

| Name | Inclusive ... | Calls |
|---|---|---|
| _do_call [{session.py}{1348}] | 512.135 | 82 |
| _run_fn [{session.py}{1317}] | 512.134 | 82 |
| TF_Run | 512.093 | 82 |
| [CONTEXT] TF_Run | 512.173 | 51,211 |
| [SAMPLE] __pthread_cond_wait [{} {0}] | 511.273 | 51,123 |
| [SAMPLE] tensorflow::TensorBuffer* tensorflow::(anonymous namespace)::FromProtoFiel | 0.42 | 42 |
| [SAMPLE] __memcpy_ssse3_back [{} {0}] | 0.28 | 28 |
| [SAMPLE] _int_free [{malloc.c} {0}] | 0.03 | 3 |
| [SAMPLE] __GI___libc_malloc [{} {0}] | 0.02 | 2 |
| [SAMPLE] std::basic_string<char, std::char_traits<char>, std::allocator<char> >::assign(s | 0.02 | 1 |
| [SAMPLE] google::protobuf::internal::MapField<tensorflow::NodeDef::NodeDef_AttrEntry, | 0.02 | 1 |
| [SAMPLE] __exchange_and_add [{/home/msarahan/miniconda2/conda-bld/compilers_lin | 0.01 | 1 |
| [SAMPLE] void google::protobuf::internal::RepeatedPtrFieldBase::MergeFromInnerLoop<g | 0.01 | 1 |
| [SAMPLE] google::protobuf::internal::ArenaStringPtr::Destroy(std::basic_string<char, std:: | 0.01 | 1 |
| [SAMPLE] std::_Hashtable<std::basic_string<char, std::char_traits<char>, std::allocator< | 0.01 | 1 |
| [SAMPLE] std::_Hashtable<tensorflow::Node*, std::pair<tensorflow::Node* const, tensorf | 0.01 | 1 |
| [SAMPLE] std::basic_string<char, std::char_traits<char>, std::allocator<char> >::_Rep::_ | 0.01 | 1 |
| [SAMPLE] std::_Hash_bytes(void const*, unsigned long, unsigned long) [{/home/msaraha | 0.01 | 1 |
| [SAMPLE] std::basic_string<char, std::char_traits<char>, std::allocator<char> >::_M_cop | 0.01 | 1 |
| [SAMPLE] std::basic_string<char, std::char_traits<char>, std::allocator<char> >::size() cc | 0.01 | 1 |
| [SAMPLE] PyObject_Malloc [{/home/nwani/m2u/conda-bld/python_1500576437846/wo | 0.01 | 1 |
| [SAMPLE] jemalloc_free [{/home/nchaimov/candle/anaconda3/lib/python3.6/site-packag | 0.01 | 1 |

$ tau_python -ebs ./foo.py

# Acknowledgements

# PRL, University of Oregon, Eugene





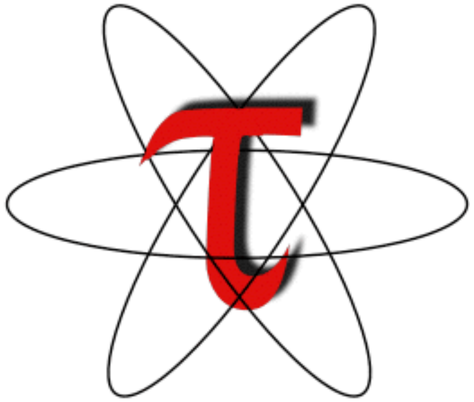

www.uoregon.edu

# Support Acknowledgments

# Acknowledgment

# Cheat-Sheet

# Download TAU from U. Oregon



http://tau.uoregon.edu

http://taucommander.com

http://www.hpclinux.com [OVA for VirtualBox]

https://e4s.io [ Extreme-Scale Scientific Software Stack, Containers for HPC]

Free download, open source, BSD license

# Setup: Installing TAU on Laptops

Prerequisites: Java in your path

Microsoft Windows

- Install Java from Oracle.com
  - http://tau.uoregon.edu/tau.exe
  - Install, click on a ppk file to launch paraprof

macOS (x86_64)

- Install Java 11.0.3:
  - Download and install http://tau.uoregon.edu/java.dmg
  - If you have multiple Java installations, add to your ~/.zshrc (or ~/.bashrc as appropriate):
  - export PATH=/Library/Java/JavaVirtualMachines/jdk-11.0.3.jdk/Contents/Home/bin:$PATH
  - java -version
- Download and install TAU (copy to /Applications from dmg):
  - http://tau.uoregon.edu/tau.dmg
  - export PATH=/Applications/TAU/tau/apple/bin:$PATH
  - paraprof  app.ppk &

macOS (arm64, M1,M2)

  - http://tau.uoregon.edu/java_arm64.dmg
  - http://tau.uoregon.edu/tau_arm64.dmg

Linux (http://tau.uoregon.edu/tau.tgz)

- ./configure -mpi -bfd=download -iowrapper -dwarf=download -unwind=download -otf=download ; make install

# Installing and Configuring TAU

- Installing PDT:
  - wget tau.uoregon.edu/pdt_lite.tgz
  - ./configure; make ; make install
- Installing TAU:
  - wget tau.uoregon.edu/tau.tgz;  tar zxf tau.tgz; cd tau-2.<ver>
  - wget http://tau.uoregon.edu/ext.tgz ; tar xf ext.tgz
  - ./configure -bfd=download -pdt=<dir> -papi=<dir> -mpi –pthread –c++=mpicxx –cc=mpicc –fortran=mpif90 –dwarf=download –unwind=download –otf=download –iowrapper –papi=<dir>
  - make install
- Using TAU:
  - export TAU_MAKEFILE=<taudir>/x86_64/lib/Makefile.tau-<TAGS>
  - make CC=tau_cc.sh  CXX=tau_cxx.sh  F90=tau_f90.sh

# Compile-Time Options

Optional parameters for the TAU_OPTIONS environment variable:
% tau_compiler.sh

-optVerbose Turn on verbose debugging messages
-optCompInst Use compiler based instrumentation
-optNoCompInst Do not revert to compiler instrumentation if source instrumentation fails.
-optTrackIO Wrap POSIX I/O call and calculates vol/bw of I/O operations (Requires TAU to be configured with *–iowrapper*)
-optTrackGOMP Enable tracking GNU OpenMP runtime layer (used without –opari)
-optMemDbg Enable runtime bounds checking (see TAU_MEMDBG_* env vars)
-optKeepFiles Does not remove intermediate .pdb and .inst.* files
-optPreProcess Preprocess sources (OpenMP, Fortran) before instrumentation
-optTauSelectFile=”<file>" Specify selective instrumentation file for *tau_instrumentor*
-optTauWrapFile=”<file>" Specify path to *link_options.tau* generated by *tau_gen_wrapper*
-optHeaderInst Enable Instrumentation of headers
-optTrackUPCR Track UPC runtime layer routines (used with tau_upc.sh)
-optLinking="" Options passed to the linker. Typically $(TAU_MPI_FLIBS) $(TAU_LIBS) $(TAU_CXXLIBS)
-optCompile="" Options passed to the compiler. Typically $(TAU_MPI_INCLUDE) $(TAU_INCLUDE) $(TAU_DEFS)
-optPdtF95Opts="" Add options for Fortran parser in PDT (f95parse/gfparse) …

# Compile-Time Options (contd.)

Optional parameters for the TAU_OPTIONS environment variable:
% tau_compiler.sh

| | |
|---|---|
| -optShared | Use TAU's shared library (libTAU.so) instead of static library (default) |
| -optPdtCxxOpts="" | Options for C++ parser in PDT (cxxparse). |
| -optPdtF90Parser="" | Specify a different Fortran parser |
| -optPdtCleanscapeParser | Specify the Cleanscape Fortran parser instead of GNU gfparser |
| -optTau="" | Specify options to the tau_instrumentor |
| -optTrackDMAPP | Enable instrumentation of low-level DMAPP API calls on Cray |
| -optTrackPthread | Enable instrumentation of pthread calls |

See tau_compiler.sh for a full list of TAU_OPTIONS.

…

# TAU's Runtime Environment Variables

| Environment Variable | Default | Description |
|---|---|---|
| TAU_TRACE | 0 | Setting to 1 turns on tracing |
| TAU_CALLPATH | 0 | Setting to 1 turns on callpath profiling |
| TAU_TRACK_MEMORY_FOOTPRINT | 0 | Setting to 1 turns on tracking memory usage by sampling periodically the resident set size and high water mark of memory usage |
| TAU_TRACK_POWER | 0 | Tracks power usage by sampling periodically. |
| TAU_CALLPATH_DEPTH | 2 | Specifies depth of callpath. Setting to 0 generates no callpath or routine information, setting to 1 generates flat profile and context events have just parent information (e.g., Heap Entry: foo) |
| TAU_SAMPLING | 1 | Setting to 1 enables event-based sampling. |
| TAU_TRACK_SIGNALS | 0 | Setting to 1 generate debugging callstack info when a program crashes |
| TAU_COMM_MATRIX | 0 | Setting to 1 generates communication matrix display using context events |
| TAU_THROTTLE | 1 | Setting to 0 turns off throttling. Throttles instrumentation in lightweight routines that are called frequently |
| TAU_THROTTLE_NUMCALLS | 100000 | Specifies the number of calls before testing for throttling |
| TAU_THROTTLE_PERCALL | 10 | Specifies value in microseconds. Throttle a routine if it is called over 100000 times and takes less than 10 usec of inclusive time per call |
| TAU_CALLSITE | 0 | Setting to 1 enables callsite profiling that shows where an instrumented function was called. Also compatible with tracing. |
| TAU_PROFILE_FORMAT | Profile | Setting to "merged" generates a single file. "snapshot" generates xml format |
| TAU_METRICS | TIME | Setting to a comma separated list generates other metrics. (e.g., ENERGY,TIME,P_VIRTUAL_TIME,PAPI_FP_INS,PAPI_NATIVE_<event>:<subevent>) |

# Runtime Environment Variables

| Environment Variable | Default | Description |
|---|---|---|
| TAU_TRACE | 0 | Setting to 1 turns on tracing |
| TAU_TRACE_FORMAT | Default | Setting to “otf2” turns on TAU’s native OTF2 trace generation (configure with –otf=download) |
| TAU_EBS_UNWIND | 0 | Setting to 1 turns on unwinding the callstack during sampling (use with tau_exec –ebs or TAU_SAMPLING=1) |
| TAU_EBS_RESOLUTION | line | Setting to “function” or “file” changes the sampling resolution to function or file level respectively. |
| TAU_TRACK_LOAD | 0 | Setting to 1 tracks system load on the node |
| TAU_SELECT_FILE | Default | Setting to a file name, enables selective instrumentation based on exclude/include lists specified in the file. |
| TAU_OMPT_SUPPORT_LEVEL | basic | Setting to “full” improves resolution of OMPT TR6 regions on threads 1.. N-1. Also, “lowoverhead” option is available. |
| TAU_OMPT_RESOLVE_ADDRESS_EAGERLY | 1 | Setting to 1 is necessary for event based sampling to resolve addresses with OMPT. Setting to 0 allows the user to do offline address translation. |

# Runtime Environment Variables

| Environment Variable | Default | Description |
|---|---|---|
| TAU_TRACK_MEMORY_LEAKS | 0 | Tracks allocates that were not de-allocated (needs –optMemDbg or tau_exec –memory) |
| TAU_EBS_SOURCE | TIME | Allows using PAPI hardware counters for periodic interrupts for EBS (e.g., TAU_EBS_SOURCE=PAPI_TOT_INS when TAU_SAMPLING=1) |
| TAU_EBS_PERIOD | 100000 | Specifies the overflow count for interrupts |
| TAU_MEMDBG_ALLOC_MIN/MAX | 0 | Byte size minimum and maximum subject to bounds checking (used with TAU_MEMDBG_PROTECT_*) |
| TAU_MEMDBG_OVERHEAD | 0 | Specifies the number of bytes for TAU's memory overhead for memory debugging. |
| TAU_MEMDBG_PROTECT_BELOW/ ABOVE | 0 | Setting to 1 enables tracking runtime bounds checking below or above the array bounds (requires –optMemDbg while building or tau_exec –memory) |
| TAU_MEMDBG_ZERO_MALLOC | 0 | Setting to 1 enables tracking zero byte allocations as invalid memory allocations. |
| TAU_MEMDBG_PROTECT_FREE | 0 | Setting to 1 detects invalid accesses to deallocated memory that should not be referenced until it is reallocated (requires –optMemDbg or tau_exec –memory) |
| TAU_MEMDBG_ATTEMPT_CONTINUE | 0 | Setting to 1 allows TAU to record and continue execution when a memory error occurs at runtime. |
| TAU_MEMDBG_FILL_GAP | Undefined | Initial value for gap bytes |
| TAU_MEMDBG_ALINGMENT | Sizeof(int) | Byte alignment for memory allocations |
| TAU_EVENT_THRESHOLD | 0.5 | Define a threshold value (e.g., .25 is 25%) to trigger marker events for min/max |