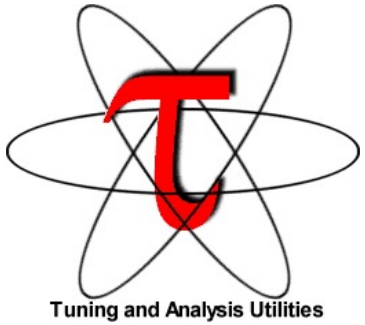
# ECP Community BOF: Observing GPU Performance Using the TAU Performance System®

March 30, 2021, 10am PT
https://www.exascaleproject.org/event/ecp-community-bof-days-2021/




Sameer Shende, Kevin Huck, Allen Malony, Wyatt Spear, Camille Coti
Performance Research Laboratory, OACISS, University of Oregon
http://tau.uoregon.edu/TAU_BoF_Mar21.pdf

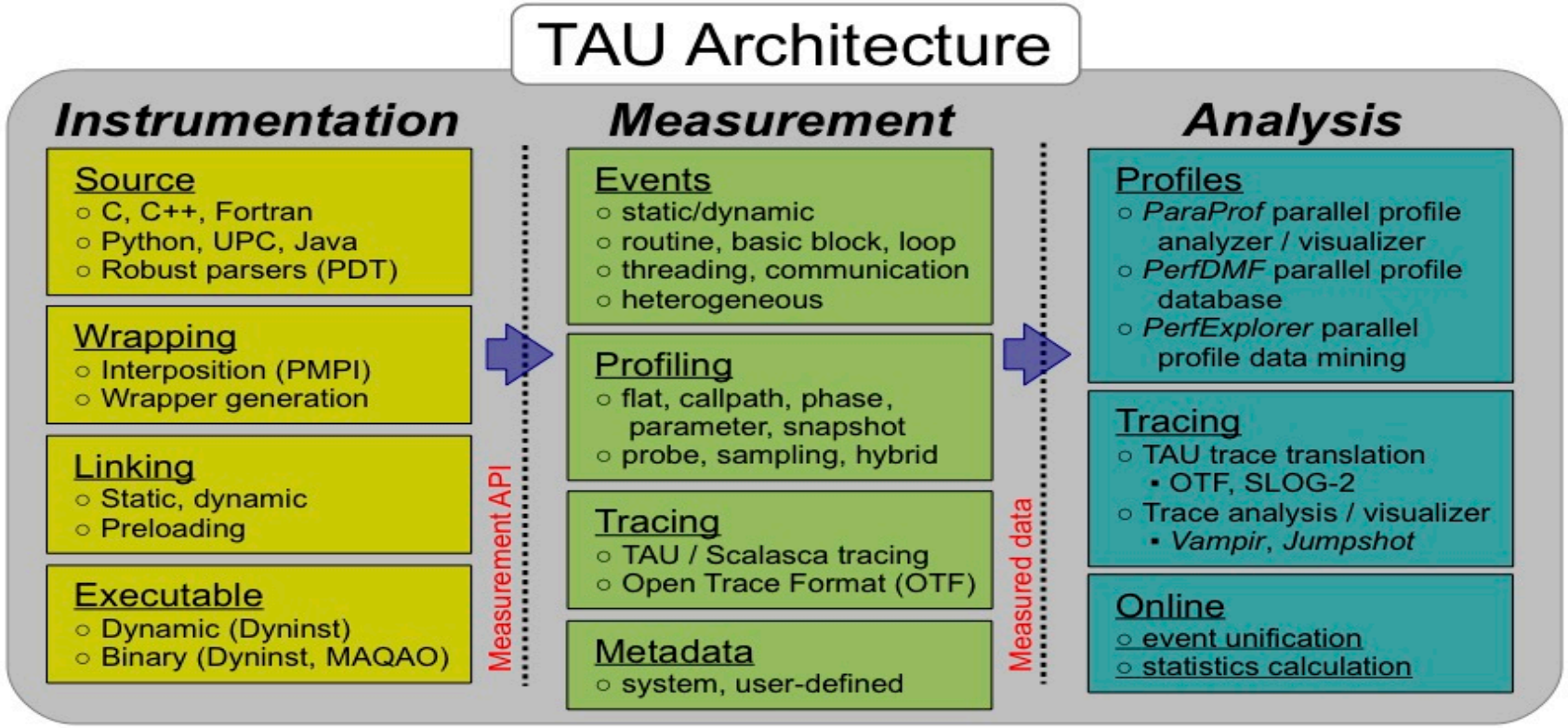
# Challenges

- With growing hardware complexity, it is getting harder to accurately measure and optimize the performance of our HPC and AI/ML workloads.
- TAU Performance System®:
  - Deliver a scalable, portable, performance evaluation toolkit for HPC and AI/ML workloads
  - http://tau.uoregon.edu

# TAU Performance System®

## Parallel performance framework and toolkit

Supports all HPC platforms, compilers, runtime system
Provides portable instrumentation, measurement, analysis



# TAU Performance System

Instrumentation
- Fortran, C++, C, UPC, Java, Python, Chapel, Spark
- Automatic instrumentation

Measurement and analysis support
- MPI, OpenSHMEM, ARMCI, PGAS, DMAPP
- pthreads, OpenMP, OMPT interface, hybrid, other thread models
- GPU, ROCm, CUDA, OpenCL, OpenACC
- Parallel profiling and tracing

Analysis
- Parallel profile analysis (ParaProf), data mining (PerfExplorer)
- Performance database technology (TAUdb)
- 3D profile browser

# Application Performance Engineering using TAU

- How much time is spent in each application routine and outer *loops*? Within loops, what is the contribution of each *statement*? What is the time spent in OpenMP loops? In kernels on GPUs. How long did it take to transfer data between host and device (GPU)?
- How many instructions are executed in these code regions?
  Floating point, Level 1 and 2 *data cache misses*, hits, branches taken? What is the extent of vectorization for loops?
- What is the memory usage of the code? When and where is memory allocated/de-allocated? Are there any memory leaks? What is the memory footprint of the application? What is the memory high water mark?
- How much energy does the application use in Joules? What is the peak power usage?
- What are the I/O characteristics of the code?  What is the peak read and write *bandwidth* of individual calls, total volume?
- How does the application *scale*? What is the efficiency, runtime breakdown of performance across different core counts?

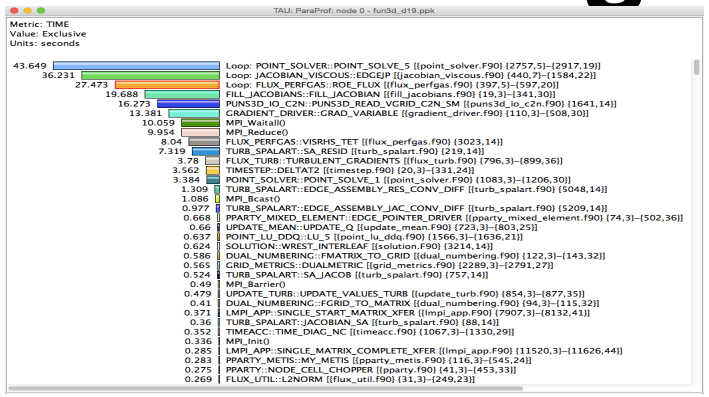
# Instrumentation

## Add hooks in the code to perform measurements

- **Source instrumentation using a preprocessor**
  - Add timer start/stop calls in a copy of the source code.
  - Use Program Database Toolkit (PDT) for parsing source code.
  - Requires recompiling the code using TAU shell scripts (tau_cc.sh, tau_f90.sh)
  - Selective instrumentation (filter file) can reduce runtime overhead and narrow instrumentation focus.
- **Compiler-based instrumentation**
  - Use system compiler to add a special flag to insert hooks at routine entry/exit.
  - Requires recompiling using TAU compiler scripts (tau_cc.sh, tau_f90.sh…)
- **Runtime preloading of TAU's Dynamic Shared Object (DSO)**
  - No need to recompile code! Use **mpirun tau_exec ./app** with options.

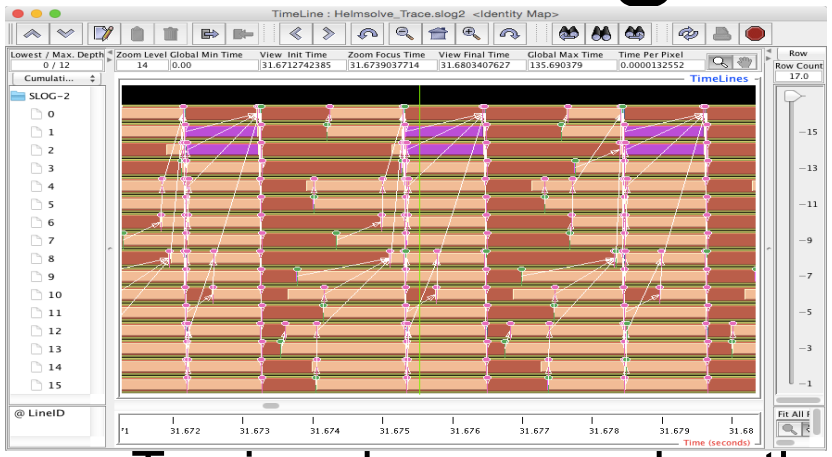
# Profiling and Tracing

## Profiling




- **Profiling** shows you **how much** (total) time was spent in each routine

## Tracing




- Tracing shows you when the events take place on a timeline

- Profiling and tracing

  **Profiling** shows you **how much** (total) time was spent in each routine

  **Tracing** shows you **when** the events take place on a timeline

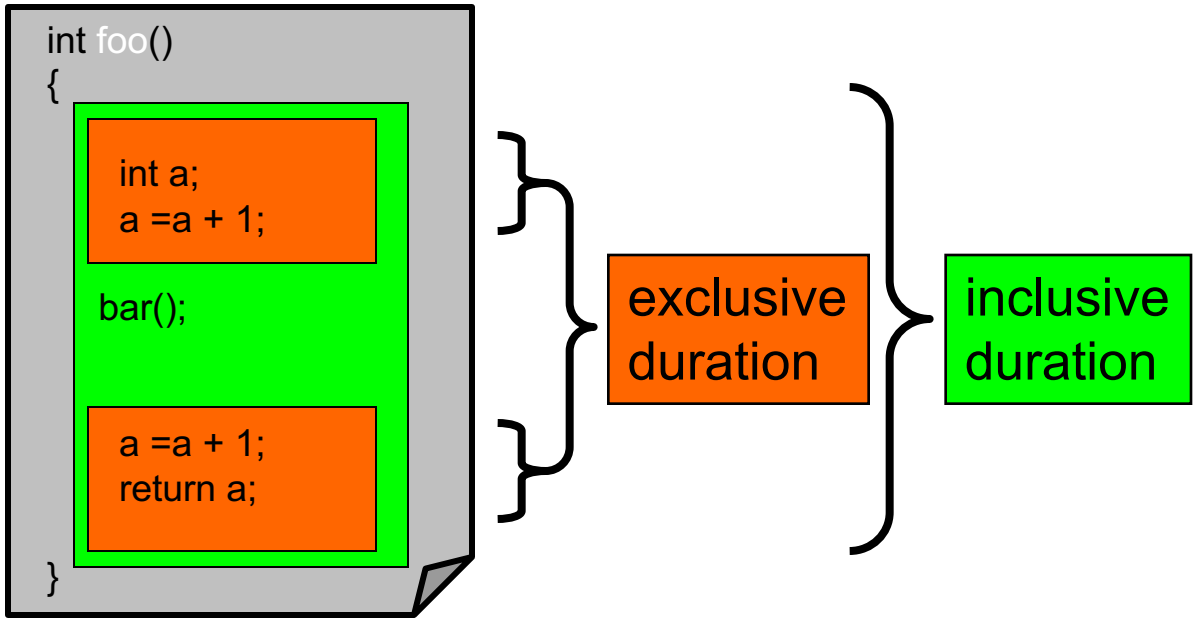
# Instrumentation

- Direct and indirect performance observation
- Instrumentation invokes performance measurement
- Direct measurement with *probes*
- Indirect measurement with periodic sampling or hardware performance counter overflow interrupts
- Events measure performance data, metadata, context, etc.

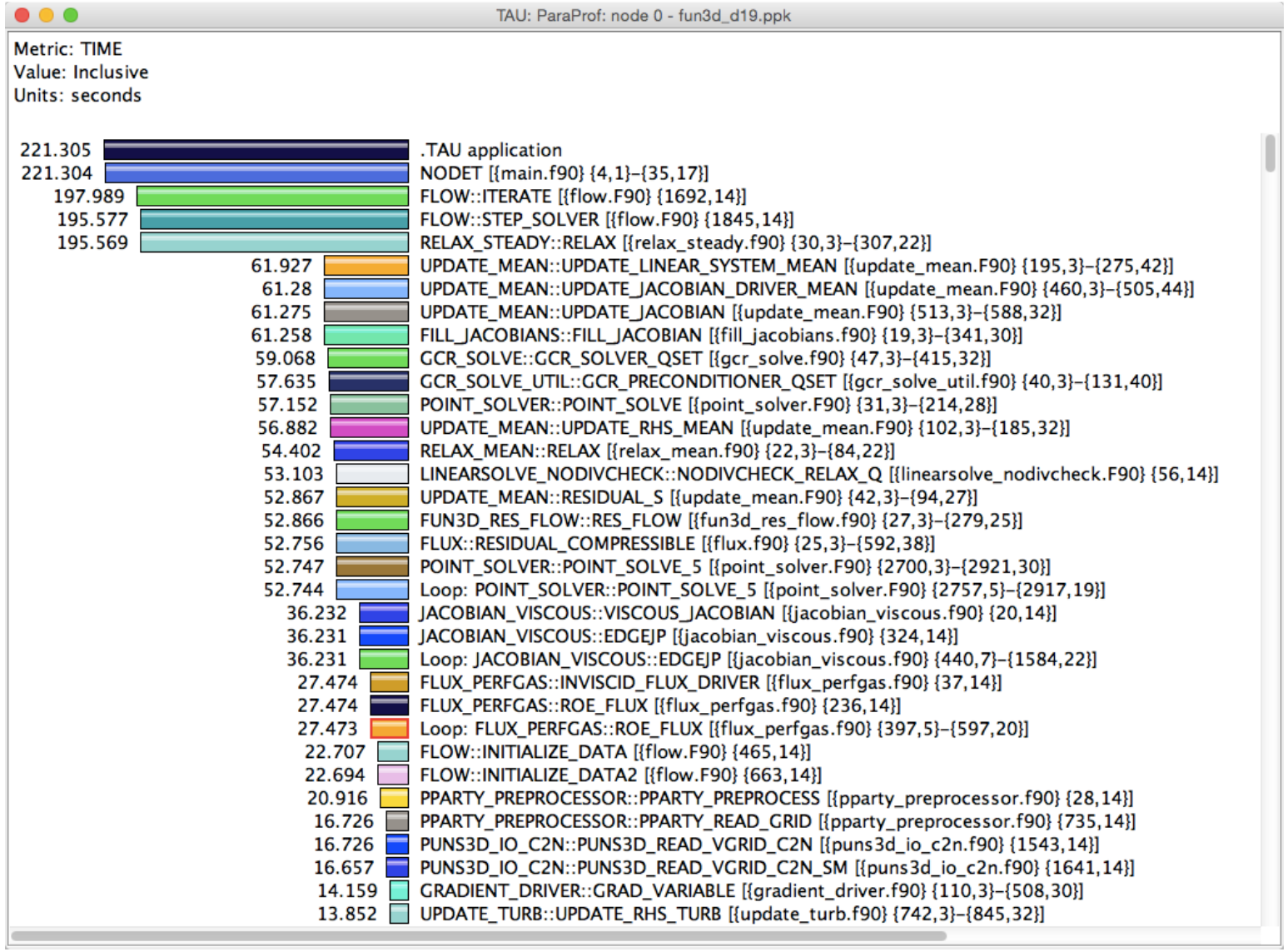
- User-defined events
  - ***Interval*** (start/stop) events to measure exclusive & inclusive duration
  - ***Atomic events*** take measurements at a single point
    - Measures total, samples, min/max/mean/std. deviation statistics
  - ***Context events*** are atomic events with executing context
    - Measures above statistics for a given calling path

# Inclusive vs. Exclusive Measurements

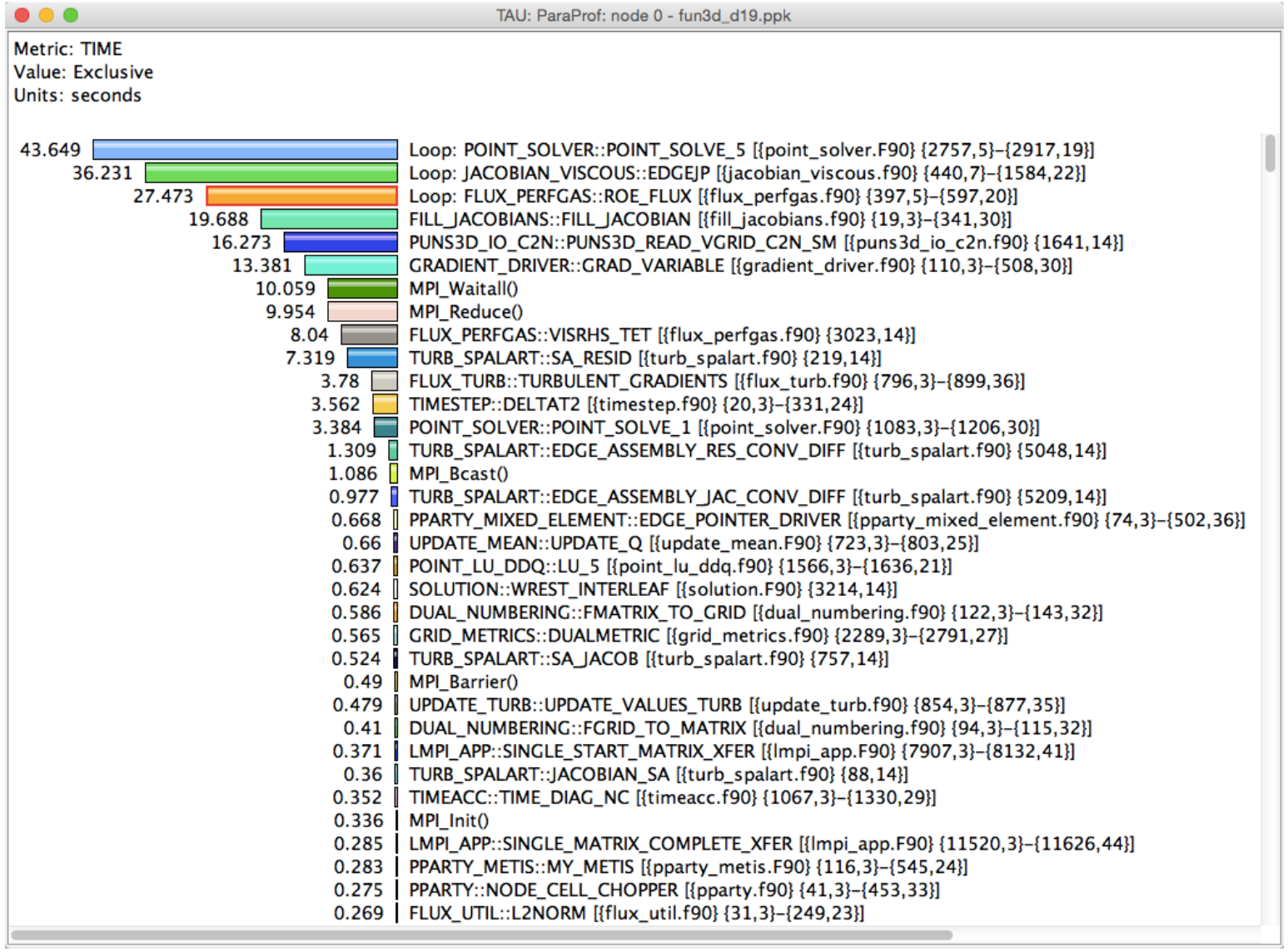
- Performance with respect to code regions
- Exclusive measurements for region only
- Inclusive measurements includes child regions



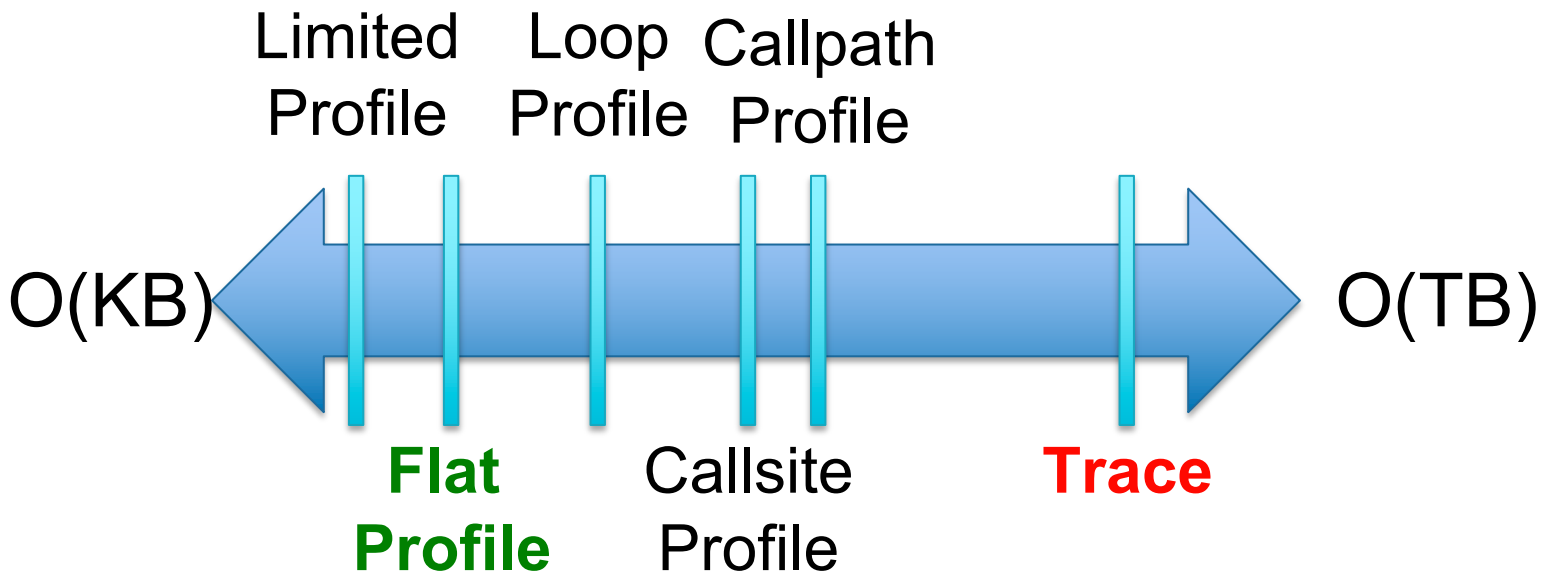
# Inclusive Measurements



# Exclusive Time



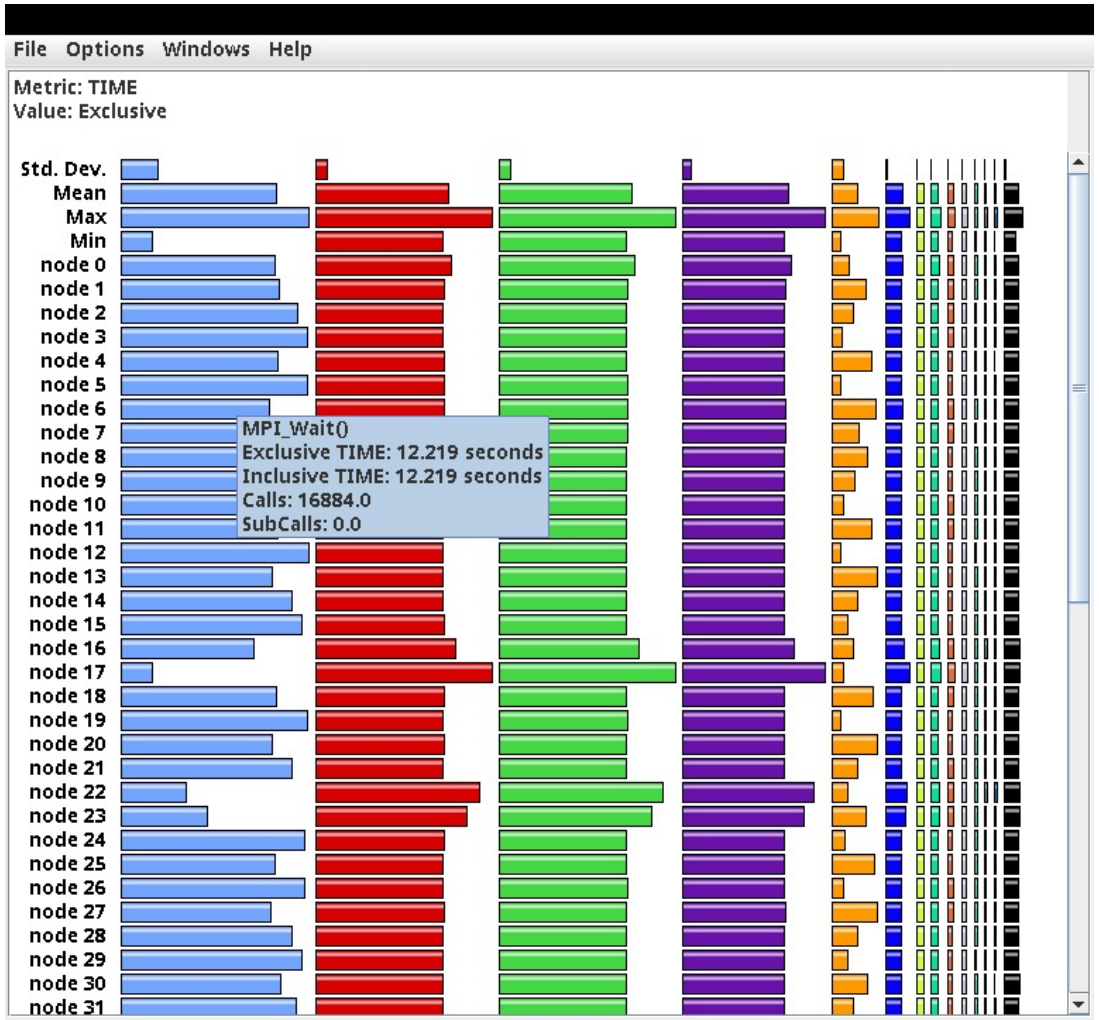
# How much data do you want?



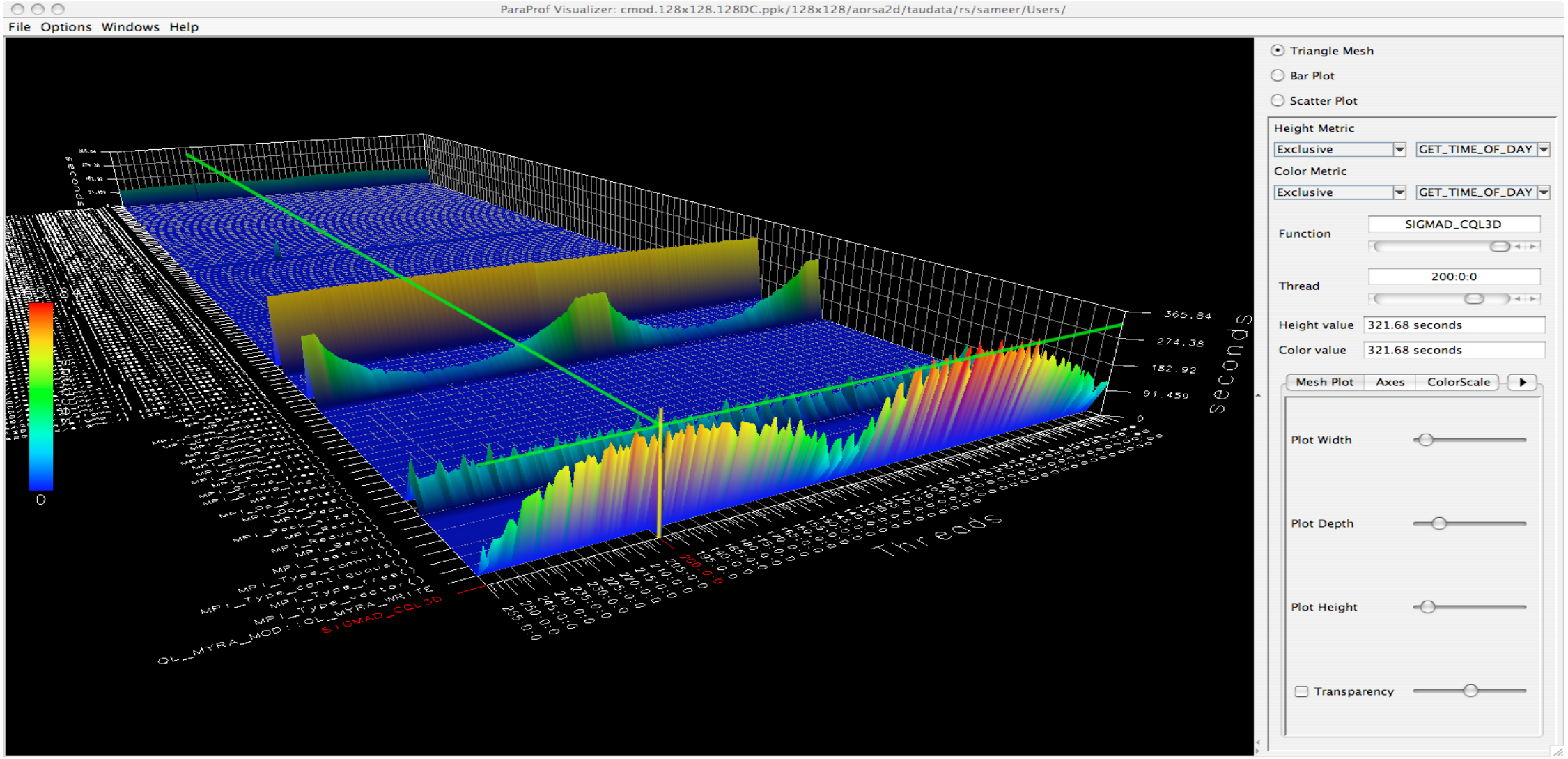
# ParaProf Profile Browser

```
% paraprof
```

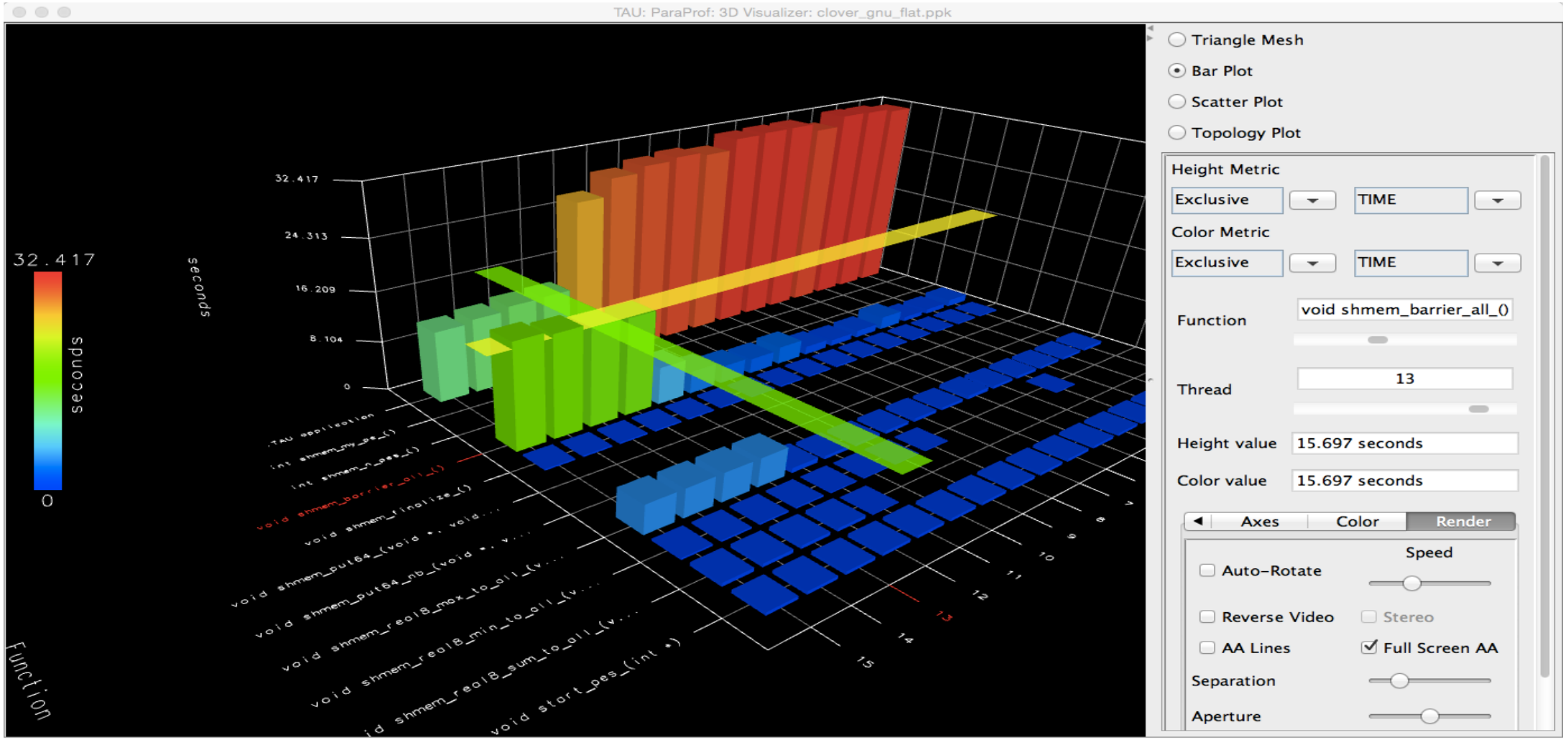
# ParaProf Profile Browser

# ParaProf 3D Profile Browser

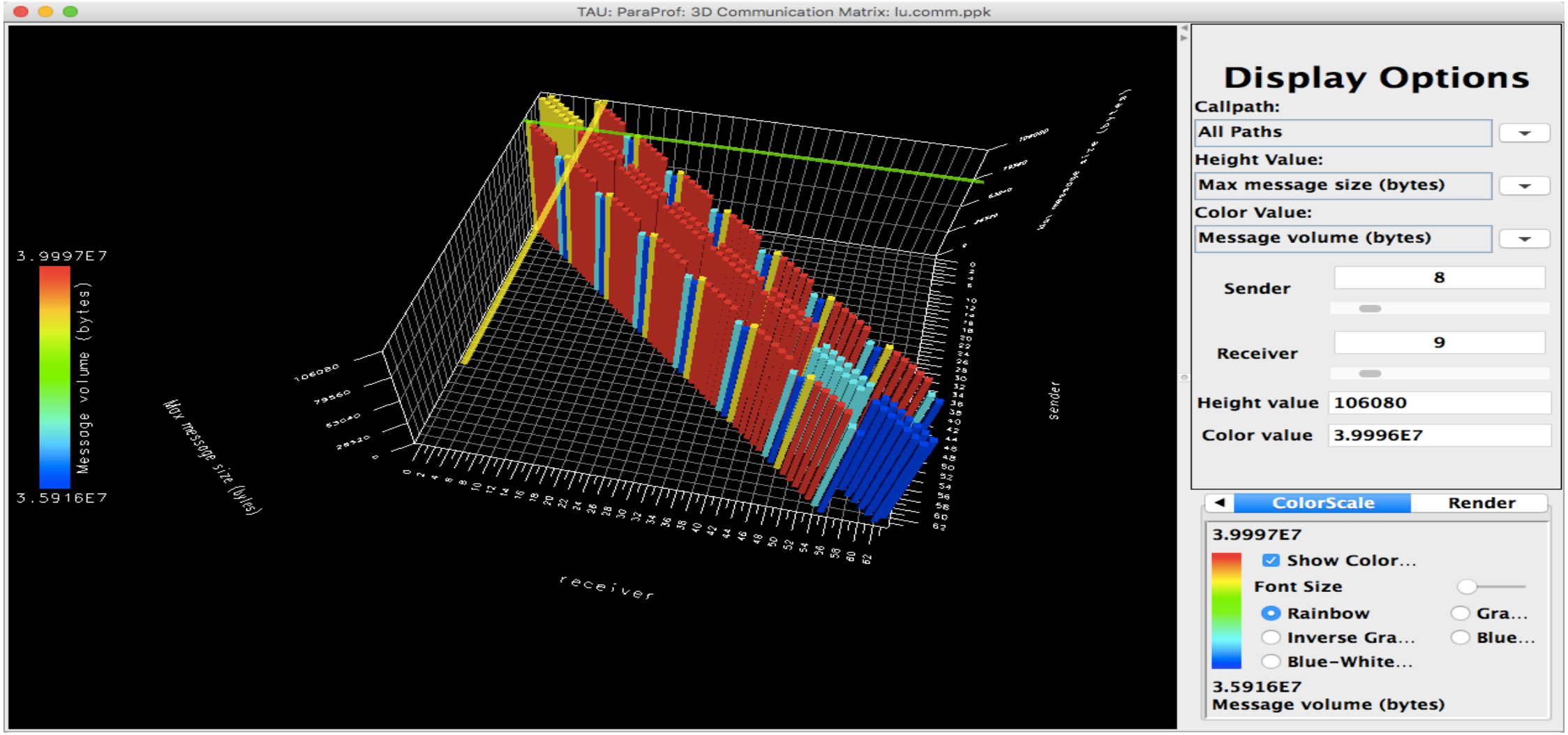


# TAU – ParaProf 3D Visualization



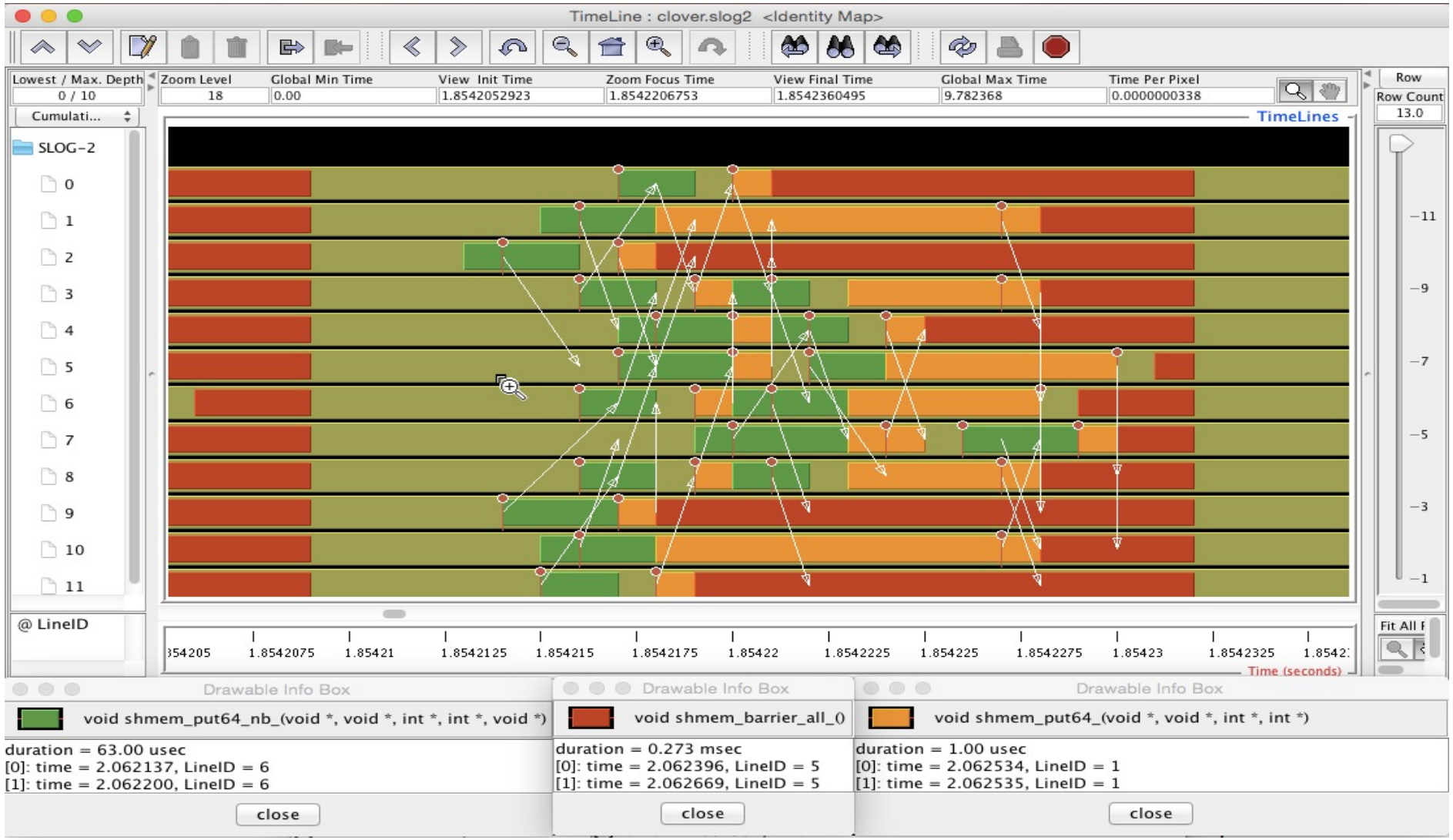

% paraprof app.ppk
Windows -> 3D Visualization -> Bar Plot (right pane)

# TAU – 3D Communication Window

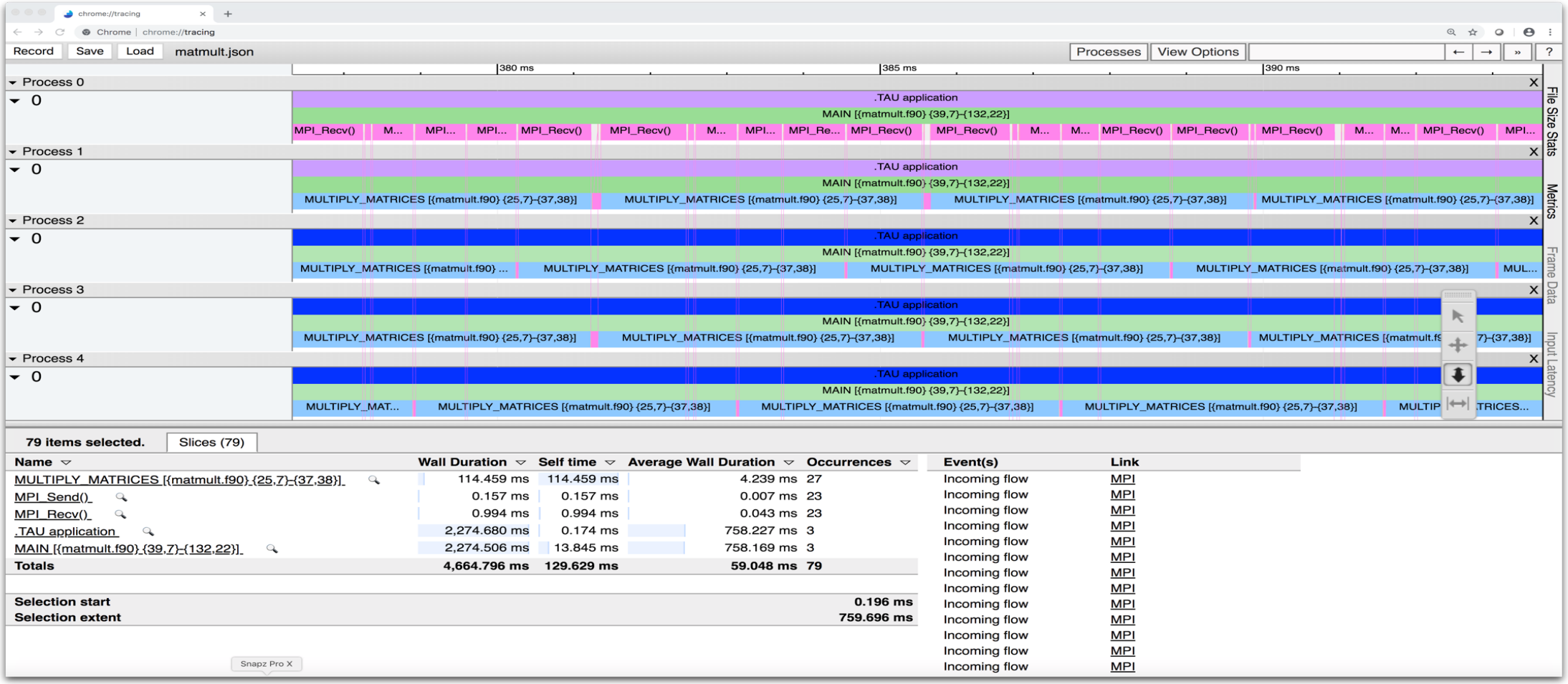



% export TAU_COMM_MATRIX=1; mpirun … tau_exec ./a.out
% paraprof ;  Windows -> 3D Communication Matrix

# Tracing: Jumpshot (ships with TAU)



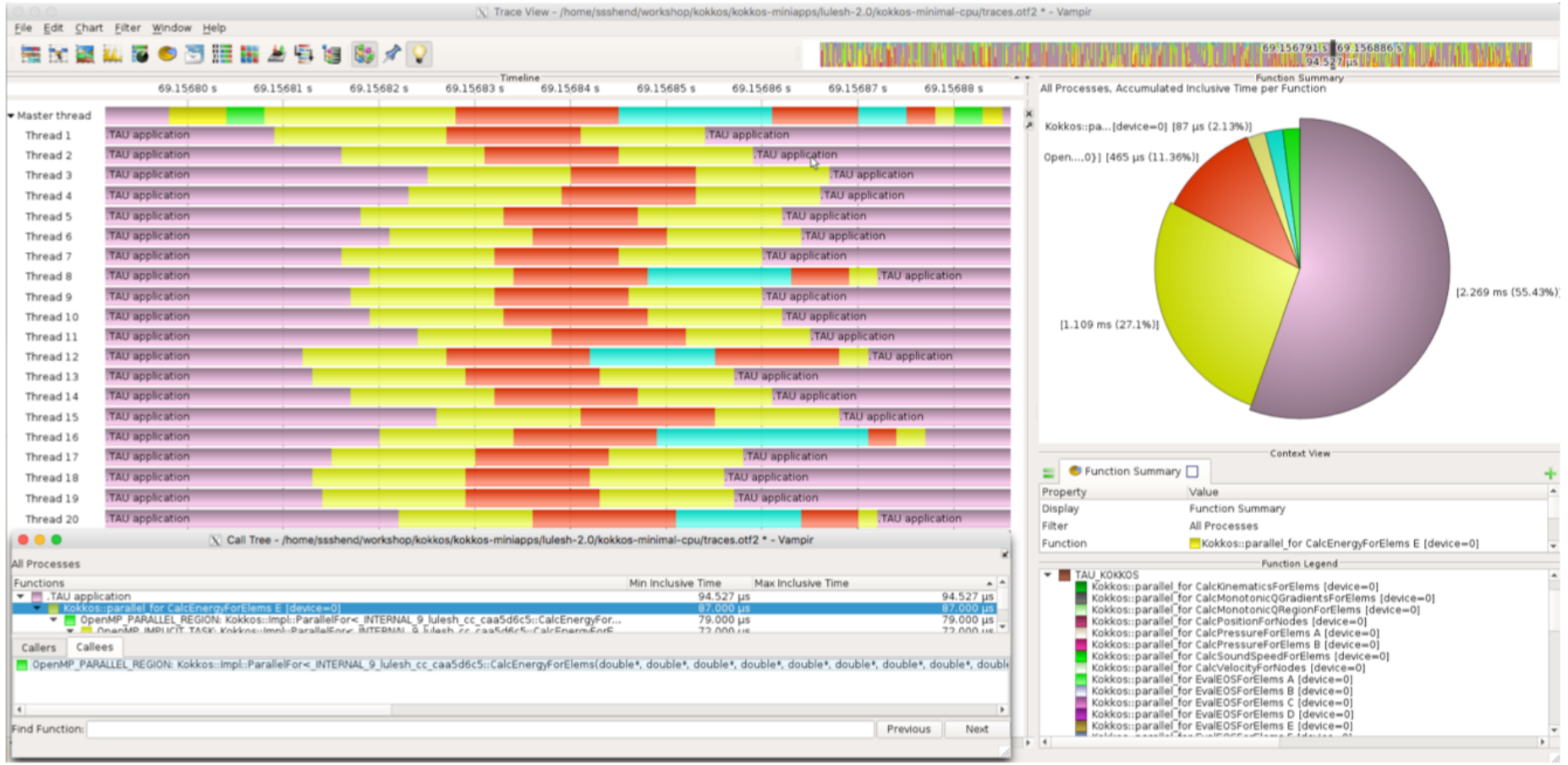
# Tracing: Chrome Browser




**% export TAU_TRACE=1**
**% mpirun –np 256 tau_exec ./a.out**
**% tau_treemerge.pl; tau_trace2json tau.trc tau.edf –chrome –ignoreatomic –o app.json**

**Chrome browser: chrome://tracing (Load -> app.json)**

# Vampir [TU Dresden] Timeline: Kokkos




```
% export TAU_TRACE=1; export TAU_TRACE_FORMAT=otf2
% tau_exec –ompt  ./a.out
% vampir traces.otf2 &
```

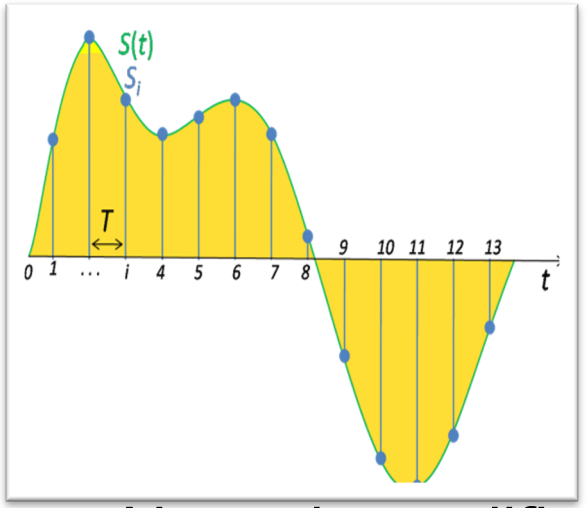
# Performance Data Measurement

## Direct via Probes

```
Call START('potential')
// code
Call STOP('potential')
```

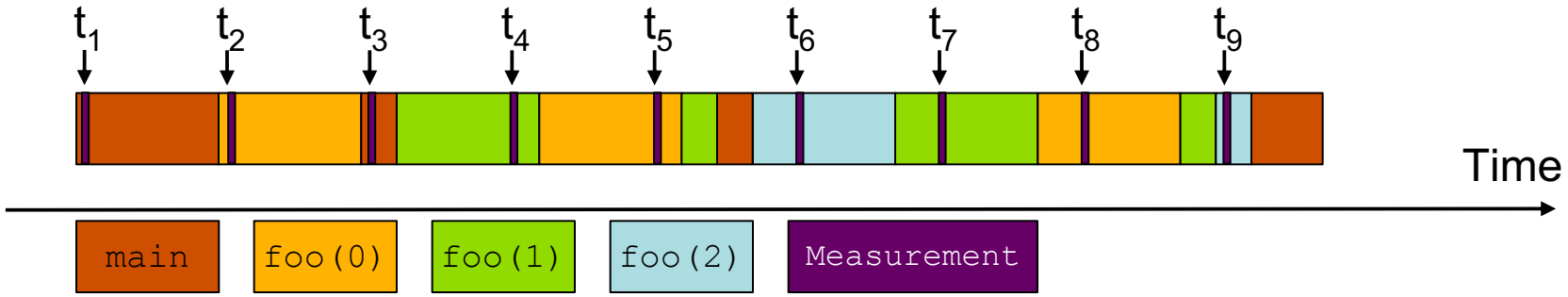
- Exact measurement
- Fine-grain control
- Calls inserted into code

## Indirect via Sampling




- No code modification
- Minimal effort
- Relies on debug symbols (**-g**)

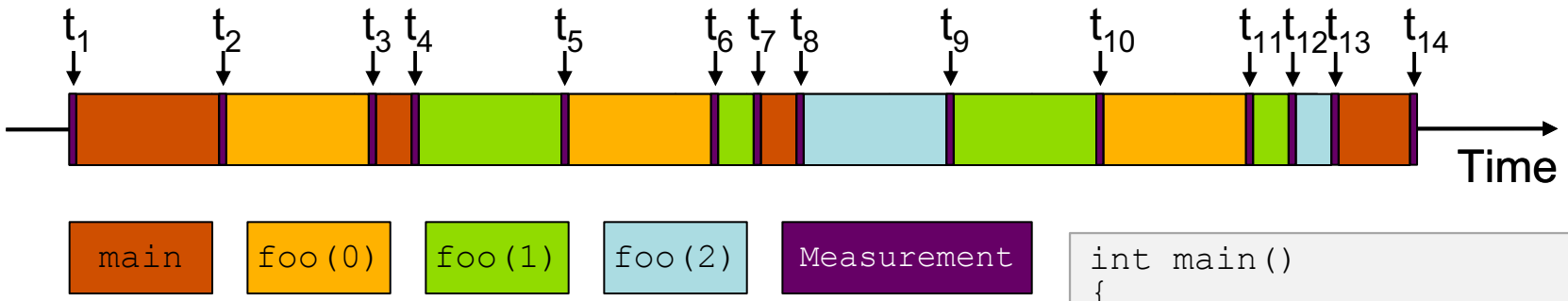
# Sampling




- Running program is periodically interrupted to take measurement
  - Timer interrupt, OS signal, or HWC overflow
  - Service routine examines return-address stack
  - Addresses are mapped to routines using symbol table information
- Statistical inference of program behavior
  - Not very detailed information on highly volatile metrics
  - Requires long-running applications
- Works with unmodified executables

```
int main()
{
  int i;

  for (i=0; i < 3; i++)
    foo(i);

  return 0;
}

void foo(int i)
{

  if (i > 0)
    foo(i - 1);

}
```

# Instrumentation




- Measurement code is inserted such that every event of interest is captured directly
  - Can be done in various ways
- Advantage:
  - Much more detailed information
- Disadvantage:
  - Processing of source-code / executable necessary
  - Large relative overheads for small functions

```
int main()
{
  int i;
  Start("main");
  for (i=0; i < 3; i++)
    foo(i);
  Stop("main");
  return 0;
}

void foo(int i)
{
  Start("foo");
  if (i > 0)
    foo(i - 1);
  Stop("foo");
}
```

# Using TAU's Runtime Preloading Tool: tau_exec

Preload a wrapper that intercepts the runtime system call and substitutes with another

**MPI**

**OpenMP**

**POSIX I/O**

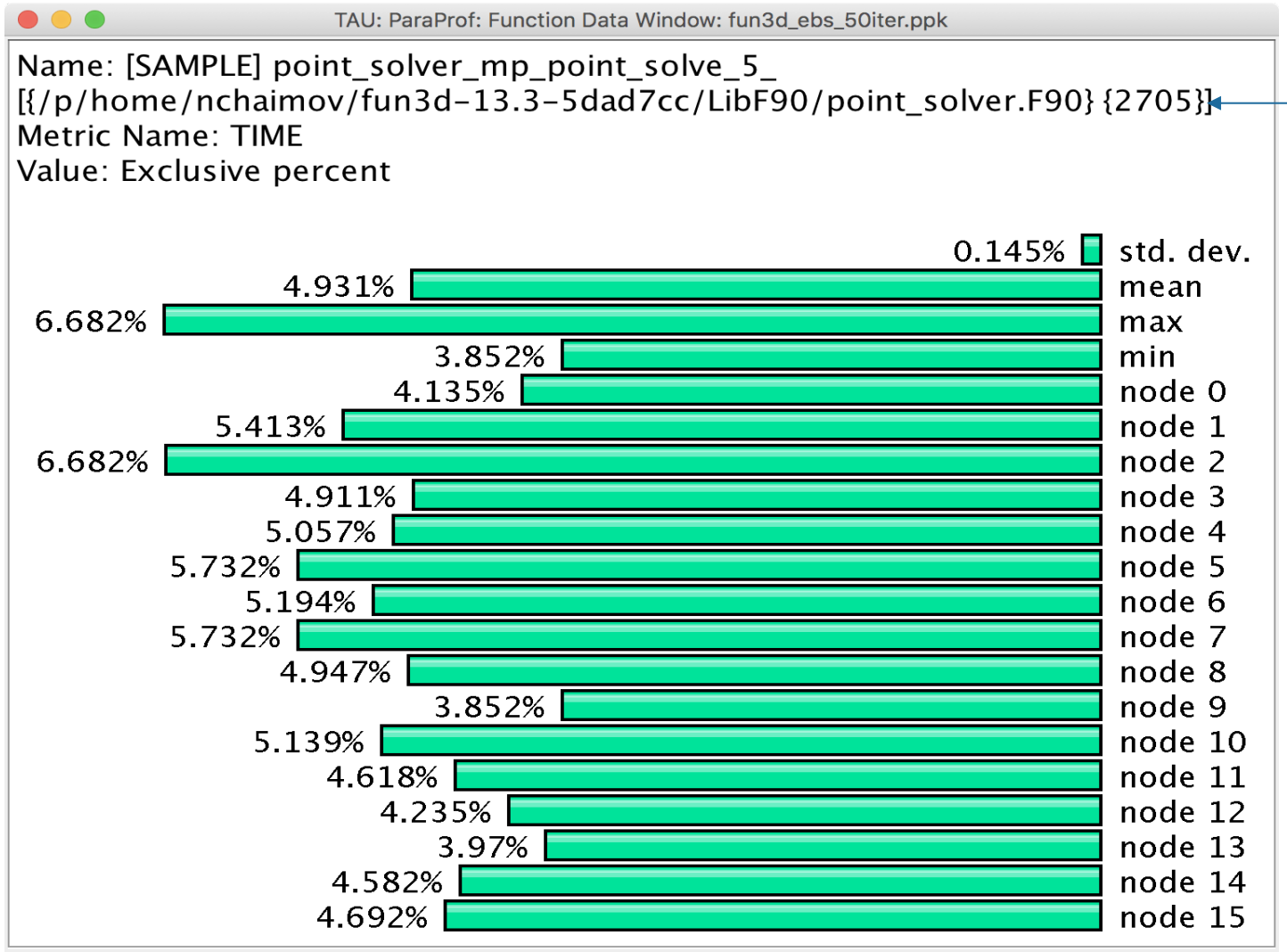
**Memory allocation/deallocation routines**

**Wrapper library for an external package**

No modification to the binary executable!

Enable other TAU options (communication matrix, OTF2, event-based sampling)

# Event Based Sampling (EBS)




File: point_solver.F90
Line: 2705

Uninstrumented!

```
% mpirun -n 16 tau_exec –ebs a.out
```

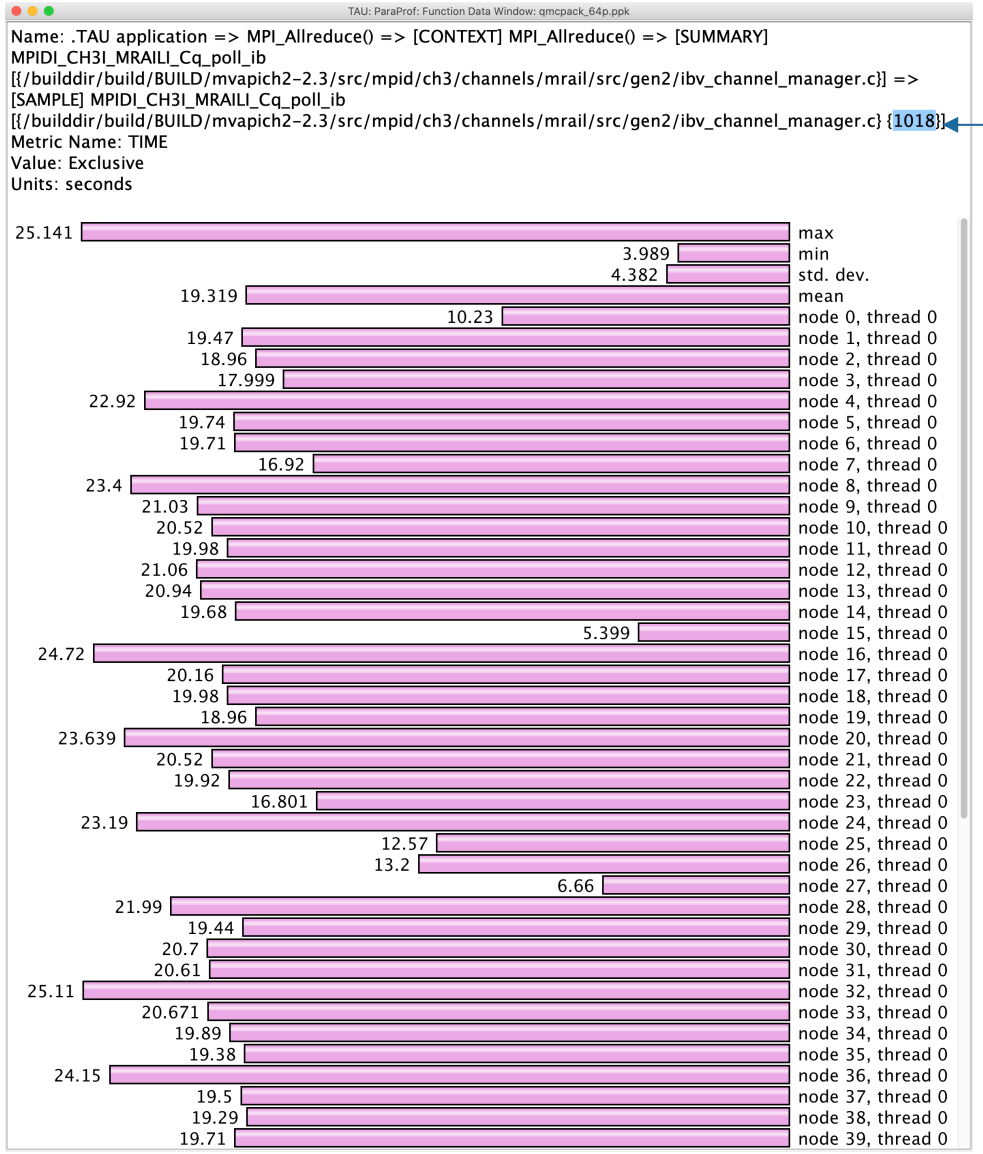
# Event Based Sampling (EBS): QMCPack

TAU: ParaProf: Statistics for: node 0, thread 0 - qmcpack_64p.ppk

| Name | Exclusive TIME | Inclusive TIME ▽ | Calls | Child Calls |
|---|---|---|---|---|
| .TAU application | 979.071 | 1,066.528 | 1 | 493 |
| [CONTEXT] .TAU application | 0 | 975.151 | 32,481 | 0 |
| MPI_Allreduce() | 35.346 | 35.346 | 52 | 0 |
| [CONTEXT] MPI_Allreduce() | 0 | 35.37 | 1,179 | 0 |
| [SUMMARY] MPIDI_CH3I_MRAILI_Get_next_vbuf [{/builddir/build/BUILD/mvapich2-2.3/src/n | 16.109 | 16.109 | 537 | 0 |
| [SUMMARY] MPIDI_CH3I_MRAILI_Cq_poll_ib [{/builddir/build/BUILD/mvapich2-2.3/src/mpid | 10.71 | 10.71 | 357 | 0 |
| [SAMPLE] MPIDI_CH3I_MRAILI_Cq_poll_ib [{/builddir/build/BUILD/mvapich2-2.3/src/mpid/ | 10.23 | 10.23 | 341 | 0 |
| [SAMPLE] MPIDI_CH3I_MRAILI_Cq_poll_ib [{/builddir/build/BUILD/mvapich2-2.3/src/mpid/ | 0.27 | 0.27 | 9 | 0 |
| [SAMPLE] MPIDI_CH3I_MRAILI_Cq_poll_ib [{/builddir/build/BUILD/mvapich2-2.3/src/mpid/ | 0.09 | 0.09 | 3 | 0 |
| [SAMPLE] MPIDI_CH3I_MRAILI_Cq_poll_ib [{/builddir/build/BUILD/mvapich2-2.3/src/mpid/ | 0.03 | 0.03 | 1 | 0 |
| [SAMPLE] MPIDI_CH3I_MRAILI_Cq_poll_ib [{/builddir/build/BUILD/mvapich2-2.3/src/mpid/ | 0.03 | 0.03 | 1 | 0 |
| [SAMPLE] MPIDI_CH3I_MRAILI_Cq_poll_ib [{/builddir/build/BUILD/mvapich2-2.3/src/mpid/ | 0.03 | 0.03 | 1 | 0 |
| [SAMPLE] MPIDI_CH3I_MRAILI_Cq_poll_ib [{/builddir/build/BUILD/mvapich2-2.3/src/mpid/ | 0.03 | 0.03 | 1 | 0 |
| [SUMMARY] MPIDI_CH3I_SMP_read_progress [{/builddir/build/BUILD/mvapich2-2.3/src/mpi | 1.98 | 1.98 | 66 | 0 |
| [SUMMARY] MPIDI_CH3I_SMP_pull_header [{/builddir/build/BUILD/mvapich2-2.3/src/mpid/ | 1.77 | 1.77 | 59 | 0 |
| [SAMPLE] GetSeqNumVbuf [{/builddir/build/BUILD/mvapich2-2.3/src/mpid/ch3/channels/n | 1.23 | 1.23 | 41 | 0 |
| [SAMPLE] UNRESOLVED /usr/lib64/libmlx5.so.1.0.0 | 1.05 | 1.05 | 35 | 0 |
| [SAMPLE] ibv_poll_cq [{/usr/include/infiniband/verbs.h} {2456}] | 0.96 | 0.96 | 32 | 0 |
| [SUMMARY] MPIDI_CH3I_Progress [{/builddir/build/BUILD/mvapich2-2.3/src/mpid/ch3/cha | 0.6 | 0.6 | 20 | 0 |
| [SUMMARY] MPIDI_CH3I_MRAILI_Test_pkt [{/builddir/build/BUILD/mvapich2-2.3/src/mpid/c | 0.24 | 0.24 | 8 | 0 |
| [SUMMARY] MPIDU_Sched_progress [{/builddir/build/BUILD/mvapich2-2.3/src/mpid/comm | 0.21 | 0.21 | 7 | 0 |
| [SAMPLE] pthread_spin_lock [{/usr/lib64/libpthread-2.17.so} {0}] | 0.18 | 0.18 | 6 | 0 |
| [SUMMARY] MPIDI_CH3I_read_progress [{/builddir/build/BUILD/mvapich2-2.3/src/mpid/ch3 | 0.15 | 0.15 | 5 | 0 |
| [SUMMARY] MPIDU_Sched_progress_state [{/builddir/build/BUILD/mvapich2-2.3/src/mpid/ | 0.09 | 0.09 | 3 | 0 |
| [SUMMARY] MPIDI_CH3I_SHMEM_COLL_GetShmemBuf [{/builddir/build/BUILD/mvapich2-2.3 | 0.09 | 0.09 | 3 | 0 |
| MPI_Reduce() | 31.996 | 31.996 | 9 | 0 |
| MPI_Gatherv() | 9.433 | 9.433 | 230 | 0 |
| MPI_Bcast() | 5.452 | 5.452 | 71 | 0 |
| MPI_Init() | 2.356 | 2.356 | 1 | 2 |
| MPI_Finalize() | 1.333 | 1.351 | 1 | 4 |

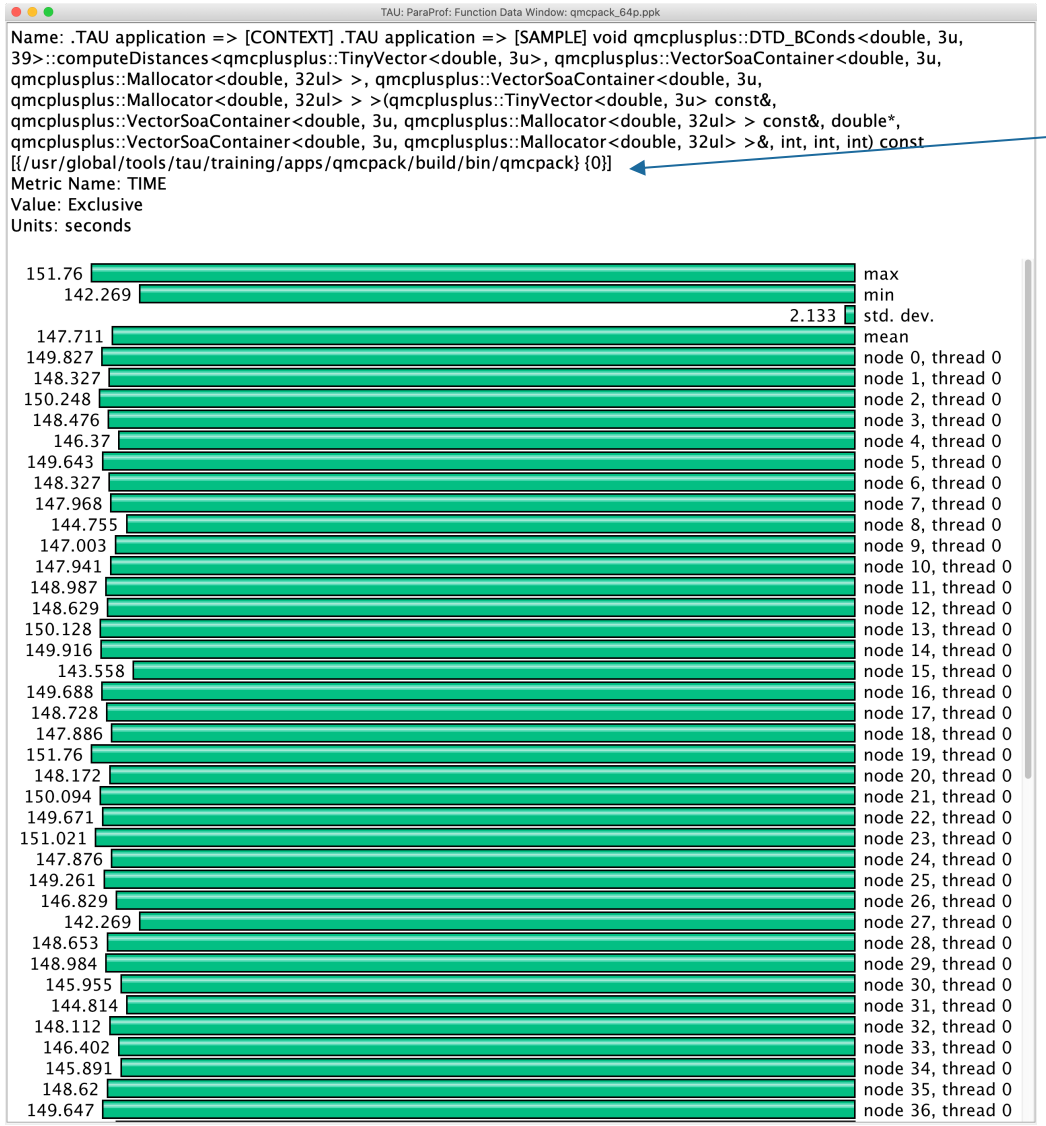
Ran for 1066.528 seconds. Outside of MPI calls, TAU can explain 975.151 seconds out of 979.071 seconds of exclusive time using EBS!

# Event Based Sampling (EBS) shows statement level information




File: ibv_channel_manager.c
Line: 1018

# Event Based Sampling without symbol information (-g): QMCPack




Line number is 0!

~147.71 seconds are spent in Qmcplusplus:DTD_Bconds:: computeDistances method

# EBS introspection in system libraries

TAU: ParaProf: Statistics for: node 1, thread 0 - hdf5_ex1.ppk

| Name | Exclusive TIME | Inclusive TIME ▽ | Calls | Child Calls |
|---|---|---|---|---|
| .TAU application | 0.097 | 9.271 | 1 | 4,979 |
| MPI_Barrier() | 4.561 | 4.561 | 5 | 0 |
| [CONTEXT] MPI_Barrier() | 0 | 4.59 | 153 | 0 |
| [SAMPLE] pthread_spin_lock [{/usr/lib64/libpthread-2.17.so} {0}] | 1.89 | 1.89 | 63 | 0 |
| [SAMPLE] PAMI_Context_advancev [{/autofs/nccs-svm1_sw/summit/.swci | 0.96 | 0.96 | 32 | 0 |
| [SAMPLE] start_libcoll_blocking_collective [{/autofs/nccs-svm1_sw/summ | 0.6 | 0.6 | 20 | 0 |
| [SAMPLE] PAMI::Device::IBV::Device::advance() [{/autofs/nccs-svm1_sw/su | 0.48 | 0.48 | 16 | 0 |
| [SAMPLE] UNRESOLVED /usr/lib64/libmlx5.so.1.0.0 | 0.18 | 0.18 | 6 | 0 |
| [SAMPLE] pthread_spin_unlock [{/usr/lib64/libpthread-2.17.so} {0}] | 0.18 | 0.18 | 6 | 0 |
| [SUMMARY] LIBCOLL_Advance_pami [{/__SMPI_build_dir_____ | 0.15 | 0.15 | 5 | 0 |
| [SAMPLE] verbs_get_exp_ctx [{pami.cc} {0}] | 0.09 | 0.09 | 3 | 0 |
| [SUMMARY] LIBCOLL_Advance [{/__SMPI_build_dir_____ | 0.06 | 0.06 | 2 | 0 |
| MPI_Bcast() | 2.089 | 2.089 | 4,509 | 0 |
| [CONTEXT] MPI_Bcast() | 0 | 2.07 | 69 | 0 |
| [SAMPLE] pthread_spin_lock [{/usr/lib64/libpthread-2.17.so} {0}] | 0.66 | 0.66 | 22 | 0 |
| [SAMPLE] PAMI_Context_advancev [{/autofs/nccs-svm1_sw/summit/.swci | 0.51 | 0.51 | 17 | 0 |
| [SAMPLE] start_libcoll_blocking_collective [{/autofs/nccs-svm1_sw/summ | 0.27 | 0.27 | 9 | 0 |
| [SAMPLE] PAMI::Device::IBV::Device::advance() [{/autofs/nccs-svm1_sw/su | 0.21 | 0.21 | 7 | 0 |
| [SAMPLE] UNRESOLVED /usr/lib64/libmlx5.so.1.0.0 | 0.15 | 0.15 | 5 | 0 |
| [SAMPLE] pthread_spin_unlock [{/usr/lib64/libpthread-2.17.so} {0}] | 0.12 | 0.12 | 4 | 0 |
| [SUMMARY] LIBCOLL_Advance_pami [{/__SMPI_build_dir_____ | 0.09 | 0.09 | 3 | 0 |
| [SAMPLE] verbs_get_exp_ctx [{pami.cc} {0}] | 0.03 | 0.03 | 1 | 0 |
| [SUMMARY] LIBCOLL_Advance [{/__SMPI_build_dir_____ | 0.03 | 0.03 | 1 | 0 |
| MPI_Finalize() | 0.624 | 0.681 | 1 | 55 |

# 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
- *Intel® OneAPI*
  - Level Zero
  - Track time spent in kernels executing on GPU
  - Track time spent in OneAPI runtime calls
- *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
- *MPI + Intel ® OneAPI DPC++/SYCL*
- PMPI + Level Zero 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 + MPI*
  - Python + CUPTI + pthread profiling interfaces (e.g., Tensorflow, PyTorch) + MPI
- *MPI + OpenCL*
  - PMPI + OpenCL profiling interfaces

# Kokkos

- Provides abstractions for node level parallelism (X in MPI+X)
- Productive, portable, and performant shared-memory programming model
- Helps you create single source performance portable codes
- Provides data abstractions
- C++ API for expressing parallelism in your program
- Aggressive compiler transformations using C++ templates
- Low level code targets backends such as OpenMP, Pthread, CUDA
- Creates a problem for performance evaluation tools
- Gap: performance data and higher-level abstractions
- Solution: Kokkos profiling API for mapping performance data
- https://kokkos.org  Sandia National Laboratories, NM

# Kokkos API use in ExaMiniMD

20. sameer@pegasus:~/pkgs/ORNL/DEMO/BUILD/ExaMiniMD-pthread/ExaMiniMD/src/comm_types (ssh)

```
void CommMPI::update_halo() {

  Kokkos::Profiling::pushRegion("Comm::update_halo");

  N_ghost = 0;
  s=*system;

  pack_buffer_update = t_buffer_update((T_X_FLOAT*)pack_buffer.data(),pack_indicies_all.extent(1));
  unpack_buffer_update = t_buffer_update((T_X_FLOAT*)unpack_buffer.data(),pack_indicies_all.extent(1));

  for(phase = 0; phase<6; phase++) {
    pack_indicies = Kokkos::subview(pack_indicies_all,phase,Kokkos::ALL());
    if(proc_grid[phase/2]>1) {

      Kokkos::parallel_for("CommMPI::halo_update_pack",
         Kokkos::RangePolicy<TagHaloUpdatePack, Kokkos::IndexType<T_INT> >(0,proc_num_send[phase]),
         *this);
      MPI_Request request;
      MPI_Status status;
      MPI_Irecv(unpack_buffer.data(),proc_num_recv[phase]*sizeof(T_X_FLOAT)*3/sizeof(int),MPI_INT, proc_neighbors_recv[phase],100002,MPI_COMM_WORLD,&request);
      MPI_Send (pack_buffer.data(),proc_num_send[phase]*sizeof(T_X_FLOAT)*3/sizeof(int),MPI_INT, proc_neighbors_send[phase],100002,MPI_COMM_WORLD);
      s = *system;
      MPI_Wait(&request,&status);
      const int count = proc_num_recv[phase];
      if(unpack_buffer_update.extent(0)<count) {
        unpack_buffer_update = t_buffer_update((T_X_FLOAT*)unpack_buffer.data(),count);
      }
      Kokkos::parallel_for("CommMPI::halo_update_unpack",
                Kokkos::RangePolicy<TagHaloUpdateUnpack, Kokkos::IndexType<T_INT> >(0,proc_num_recv[phase]),
                *this);

    } else {
      //printf("HaloUpdateCopy: %i %i %i\n",phase,proc_num_send[phase],pack_indicies.extent(0));
      Kokkos::parallel_for("CommMPI::halo_update_self",
        Kokkos::RangePolicy<TagHaloUpdateSelf, Kokkos::IndexType<T_INT> >(0,proc_num_send[phase]),
        *this);
    }
    N_ghost += proc_num_recv[phase];
  }

  Kokkos::Profiling::popRegion();
};
```

pushRegion("Comm::update_halo")

Kokkos::parallel_for

popRegion

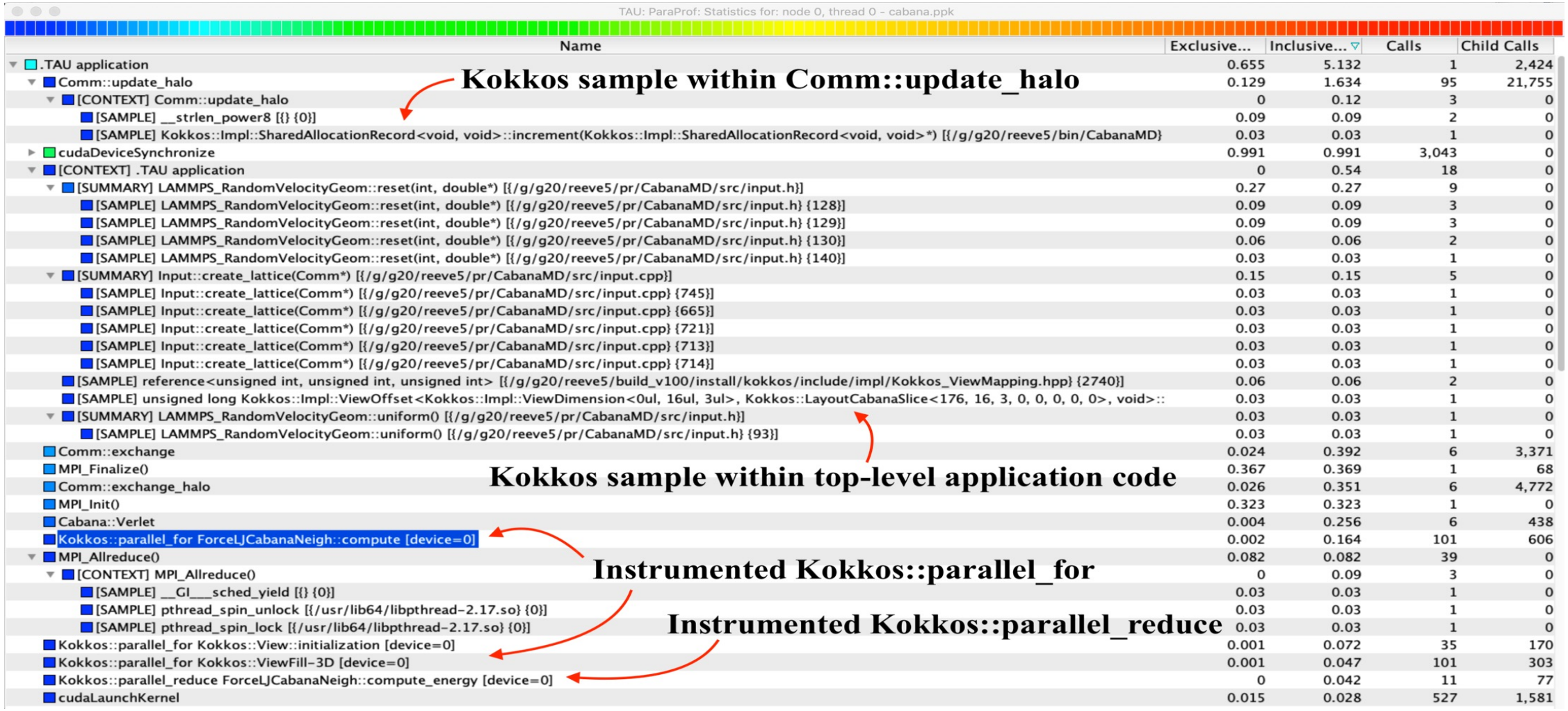
# ExaMiniMD: TAU Phase

TAU: ParaProf: Statistics for: node 0, thread 0 - examinimd_ompt_phase.ppk

| Name | Exclusive TIME | Inclusive TIME | Calls | Child Calls |
|---|---|---|---|---|
| .TAU application | 0.143 | 96.743 | 1 | 832 |
| Comm::exchange | 0.001 | 0.967 | 6 | 142 |
| Comm::exchange_halo | 0.001 | 4.702 | 6 | 184 |
| Comm::update_halo | 0.004 | 31.347 | 95 | 1,330 |
| Kokkos::parallel_for CommMPI::halo_update_pack [device=0] | 0.002 | 0.506 | 190 | 190 |
| Kokkos::parallel_for CommMPI::halo_update_self [device=0] | 0.003 | 0.597 | 380 | 380 |
| Kokkos::parallel_for CommMPI::halo_update_unpack [device=0] | 0.002 | 0.97 | 190 | 190 |
| MPI_Irecv() | 0.001 | 0.001 | 190 | 0 |
| MPI_Send() | 29.268 | 29.268 | 190 | 0 |
| MPI_Wait() | 0.001 | 0.001 | 190 | 0 |
| OpenMP_Implicit_Task | 0.041 | 1.985 | 760 | 760 |
| OpenMP_Parallel_Region parallel_for<Kokkos::RangePolicy<CommMPI::Ta | 0 | 0.504 | 190 | 190 |
| OpenMP_Parallel_Region parallel_for<Kokkos::RangePolicy<CommMPI::Ta | 0.08 | 0.968 | 190 | 190 |
| OpenMP_Parallel_Region void Kokkos::parallel_for<Kokkos::RangePolicy< | 0.001 | 0.594 | 380 | 380 |
| OpenMP_Sync_Region_Barrier parallel_for<Kokkos::RangePolicy<CommMF | 0.489 | 0.489 | 190 | 0 |
| OpenMP_Sync_Region_Barrier parallel_for<Kokkos::RangePolicy<CommMF | 0.875 | 0.875 | 190 | 0 |
| OpenMP_Sync_Region_Barrier void Kokkos::parallel_for<Kokkos::RangePol | 0.58 | 0.58 | 380 | 0 |

Comm::update_halo phase in TAU ParaProf's Thread Statistics Table

# Event-based Sampling (EBS): CabanaMD on an IBM AC922 with NVIDIA V100 GPUs




| Name | Exclusive... | Inclusive... | Calls | Child Calls |
|---|---|---|---|---|
| .TAU application | 0.655 | 5.132 | 1 | 2,424 |
| Comm::update_halo | 0.129 | 1.634 | 95 | 21,755 |
| [CONTEXT] Comm::update_halo | 0 | 0.12 | 3 | 0 |
| [SAMPLE] __strlen_power8 [{} {0}] | 0.09 | 0.09 | 2 | 0 |
| [SAMPLE] Kokkos::Impl::SharedAllocationRecord<void, void>::increment(Kokkos::Impl::SharedAllocationRecord<void, void>*) [{/g/g20/reeve5/bin/CabanaMD} | 0.03 | 0.03 | 1 | 0 |
| cudaDeviceSynchronize | 0.991 | 0.991 | 3,043 | 0 |
| [CONTEXT] .TAU application | 0 | 0.54 | 18 | 0 |
| [SUMMARY] LAMMPS_RandomVelocityGeom::reset(int, double*) [{/g/g20/reeve5/pr/CabanaMD/src/input.h}] | 0.27 | 0.27 | 9 | 0 |
| [SAMPLE] LAMMPS_RandomVelocityGeom::reset(int, double*) [{/g/g20/reeve5/pr/CabanaMD/src/input.h} {128}] | 0.09 | 0.09 | 3 | 0 |
| [SAMPLE] LAMMPS_RandomVelocityGeom::reset(int, double*) [{/g/g20/reeve5/pr/CabanaMD/src/input.h} {129}] | 0.09 | 0.09 | 3 | 0 |
| [SAMPLE] LAMMPS_RandomVelocityGeom::reset(int, double*) [{/g/g20/reeve5/pr/CabanaMD/src/input.h} {130}] | 0.06 | 0.06 | 2 | 0 |
| [SAMPLE] LAMMPS_RandomVelocityGeom::reset(int, double*) [{/g/g20/reeve5/pr/CabanaMD/src/input.h} {140}] | 0.03 | 0.03 | 1 | 0 |
| [SUMMARY] Input::create_lattice(Comm*) [{/g/g20/reeve5/pr/CabanaMD/src/input.cpp}] | 0.15 | 0.15 | 5 | 0 |
| [SAMPLE] Input::create_lattice(Comm*) [{/g/g20/reeve5/pr/CabanaMD/src/input.cpp} {745}] | 0.03 | 0.03 | 1 | 0 |
| [SAMPLE] Input::create_lattice(Comm*) [{/g/g20/reeve5/pr/CabanaMD/src/input.cpp} {665}] | 0.03 | 0.03 | 1 | 0 |
| [SAMPLE] Input::create_lattice(Comm*) [{/g/g20/reeve5/pr/CabanaMD/src/input.cpp} {721}] | 0.03 | 0.03 | 1 | 0 |
| [SAMPLE] Input::create_lattice(Comm*) [{/g/g20/reeve5/pr/CabanaMD/src/input.cpp} {713}] | 0.03 | 0.03 | 1 | 0 |
| [SAMPLE] Input::create_lattice(Comm*) [{/g/g20/reeve5/pr/CabanaMD/src/input.cpp} {714}] | 0.03 | 0.03 | 1 | 0 |
| [SAMPLE] reference<unsigned int, unsigned int, unsigned int> [{/g/g20/reeve5/build_v100/install/kokkos/include/impl/Kokkos_ViewMapping.hpp} {2740}] | 0.06 | 0.06 | 2 | 0 |
| [SAMPLE] unsigned long Kokkos::Impl::ViewOffset<Kokkos::Impl::ViewDimension<0ul, 16ul, 3ul>, Kokkos::LayoutCabanaSlice<176, 16, 3, 0, 0, 0, 0, 0>, void>:: | 0.03 | 0.03 | 1 | 0 |
| [SUMMARY] LAMMPS_RandomVelocityGeom::uniform() [{/g/g20/reeve5/pr/CabanaMD/src/input.h}] | 0.03 | 0.03 | 1 | 0 |
| [SAMPLE] LAMMPS_RandomVelocityGeom::uniform() [{/g/g20/reeve5/pr/CabanaMD/src/input.h} {93}] | 0.03 | 0.03 | 1 | 0 |
| Comm::exchange | 0.024 | 0.392 | 6 | 3,371 |
| MPI_Finalize() | 0.367 | 0.369 | 1 | 68 |
| Comm::exchange_halo | 0.026 | 0.351 | 6 | 4,772 |
| MPI_Init() | 0.323 | 0.323 | 1 | 0 |
| Cabana::Verlet | 0.004 | 0.256 | 6 | 438 |
| Kokkos::parallel_for ForceLJCabanaNeigh::compute [device=0] | 0.002 | 0.164 | 101 | 606 |
| MPI_Allreduce() | 0.082 | 0.082 | 39 | 0 |
| [CONTEXT] MPI_Allreduce() | 0 | 0.09 | 3 | 0 |
| [SAMPLE] __GI___sched_yield [{} {0}] | 0.03 | 0.03 | 1 | 0 |
| [SAMPLE] pthread_spin_unlock [{/usr/lib64/libpthread-2.17.so} {0}] | 0.03 | 0.03 | 1 | 0 |
| [SAMPLE] pthread_spin_lock [{/usr/lib64/libpthread-2.17.so} {0}] | 0.03 | 0.03 | 1 | 0 |
| Kokkos::parallel_for Kokkos::View::initialization [device=0] | 0.001 | 0.072 | 35 | 170 |
| Kokkos::parallel_for Kokkos::ViewFill-3D [device=0] | 0.001 | 0.047 | 101 | 303 |
| Kokkos::parallel_reduce ForceLJCabanaNeigh::compute_energy [device=0] | 0 | 0.042 | 11 | 77 |
| cudaLaunchKernel | 0.015 | 0.028 | 527 | 1,581 |

Event-based sampling (EBS) with Kokkos API

# 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 –l0      ./a.out
% mpirun –np 256  tau_exec –cupti ./a.out
% 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+ or Clang 8+)

```
% export TAU_OMPT_SUPPORT_LEVEL=full;
% 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  export TAU_METRICS=TIME,PAPI_L1_DCM…  -ebs_resolution=<file | function | line>
```

# AMD HIP: Kernel execution on GPUs: rochpcg

TAU: ParaProf: Statistics for: node 0, thread 6 - rochpcg_amd.ppk

| Name | Exclusive TAUGPU_... | Inclusive TAUGPU_T... | Calls | Child Calls |
|---|---|---|---|---|
| .TAU application | 235.631 | 252.534 | 1 | 500,993 |
| void kernel_symgs_sweep<128u>(int, int, int, int, int, int const*, double const*, do | 8.692 | 8.692 | 164,953 | 0 |
| void kernel_backward_sweep_0<128u>(int, int, int, int, int const*, double const*, ir | 3.261 | 3.261 | 109,748 | 0 |
| void kernel_forward_sweep_0<128u>(int, int, int, int, int const*, double const*, int | 2.948 | 2.948 | 110,058 | 0 |
| void kernel_fused_restrict_spmv<1024u>(int, int const*, double const*, int, int, int, | 0.633 | 0.633 | 11,799 | 0 |
| void kernel_waxpby<512u>(int, double, double const*, double, double const*, doul | 0.296 | 0.296 | 7,987 | 0 |
| void kernel_symgs_halo<128u>(int, int, int, int, int const*, int const*, double const | 0.285 | 0.285 | 11,799 | 0 |
| void kernel_dot2_part1<256u>(int, double const*, double const*, double*) [clone .k | 0.216 | 0.216 | 7,910 | 0 |
| void kernel_fused_waxpby_dot_part1<256u>(int, double, double const*, double*, d | 0.149 | 0.149 | 3,953 | 0 |
| void kernel_gather<128u>(int, double const*, int const*, int const*, double*) [clone | 0.124 | 0.124 | 27,601 | 0 |
| void kernel_spmv_halo<128u>(int, int, int, int const*, int const*, double const*, int | 0.093 | 0.093 | 4,038 | 0 |
| void kernel_prolongation<1024u>(int, int const*, double const*, double*, int const* | 0.077 | 0.077 | 11,775 | 0 |
| void kernel_pointwise_mult<256u>(int, double const*, double const*, double*) [clo | 0.057 | 0.057 | 15,731 | 0 |
| void kernel_dot_part2<256u>(double*) [clone .kd] | 0.025 | 0.025 | 7,986 | 0 |
| void kernel_fused_waxpby_dot_part2<256u>(int, double*) [clone .kd] | 0.013 | 0.013 | 3,951 | 0 |
| __amd_rocclr_copyBuffer.kd | 0.007 | 0.007 | 358 | 0 |
| __amd_rocclr_fillBuffer.kd | 0.004 | 0.004 | 120 | 0 |
| void kernel_jpl<27u, 16u>(int, int const*, int, int, char const*, int const*, int*) [clon | 0.003 | 0.003 | 16 | 0 |
| void kernel_setup_halo<27u, 16u>(int, int, int, int, int, int, int, bool, bool, bool, int, | 0.002 | 0.002 | 4 | 0 |
| void kernel_to_ell_val<27u, 32u>(int, int, double const*, double*) [clone .kd] | 0.002 | 0.002 | 4 | 0 |
| void kernel_perm_cols<32u, 16u>(int, int, int, int const*, int*, double*) [clone .kd] | 0.002 | 0.002 | 4 | 0 |
| void kernel_generate_problem<27u, 16u>(int, int, int, int, int, long long, long long, | 0.002 | 0.002 | 4 | 0 |
| void kernel_permute_ell_rows<1024u>(int, int, int const*, double const*, int const* | 0.002 | 0.002 | 108 | 0 |
| void kernel_dot1_part1<256u>(int, double const*, double*) [clone .kd] | 0.002 | 0.002 | 83 | 0 |
| void rocprim::detail::sort_and_scatter_kernel<256u, 15u, 6u, false, long long*, long | 0.001 | 0.001 | 72 | 0 |
| void kernel_to_ell_col<27u, 32u>(int, int, int const*, int*, int*, int*) [clone .kd] | 0.001 | 0.001 | 4 | 0 |
| void rocprim::detail::sort_and_scatter_kernel<256u, 15u, 7u, false, long long*, long | 0.001 | 0.001 | 48 | 0 |
| void rocprim::detail::sort_and_scatter_kernel<256u, 17u, 6u, false, int*, int*, rocpri | 0.001 | 0.001 | 48 | 0 |

% mpirun –np 64 tau_exec –rocm ./a.out

# Intel Level Zero (TigerLake Gen12LP integrated CPUs or DG1)

TAU: ParaProf: Statistics for: node 0, thread 0 - ze_gemm_4096.ppk

| Name | Exclusive TAUGPU_T... | Inclusive TAUGPU_TI... | Calls | Child Calls |
|---|---|---|---|---|
| .TAU application | 117,876 | 30,283,630 | 1 | 256 |
| zeCommandQueueSynchronize | 29,877,963 | 29,877,963 | 4 | 0 |
| [CONTEXT] zeCommandQueueSynchronize | 0 | 29,905,688 | 997 | 0 |
| [SAMPLE] __GI___sched_yield [{/lib64/libc-2.26.so} | 25,765,719 | 25,765,719 | 859 | 0 |
| [SAMPLE] UNRESOLVED /soft/libraries/intel-level-z | 4,139,969 | 4,139,969 | 138 | 0 |
| zeCommandQueueExecuteCommandLists | 186,203 | 186,203 | 4 | 0 |
| zeModuleCreate | 98,896 | 98,896 | 1 | 0 |
| zeCommandListAppendMemoryCopy | 1,410 | 1,410 | 12 | 0 |
| zeCommandQueueDestroy | 321 | 321 | 4 | 0 |
| zeDriverAllocDeviceMem | 137 | 137 | 12 | 0 |
| zeEventPoolDestroy | 128 | 128 | 20 | 0 |
| zeDriverFreeMem | 96 | 96 | 12 | 0 |
| zeCommandListCreate | 89 | 89 | 4 | 0 |
| zeCommandQueueCreate | 82 | 82 | 4 | 0 |
| zeCommandListDestroy | 71 | 71 | 4 | 0 |
| zeKernelSetArgumentValue | 43 | 43 | 16 | 0 |
| zeDeviceGetProperties | 38 | 38 | 26 | 0 |
| zeCommandListClose | 35 | 35 | 4 | 0 |
| zeEventCreate | 30 | 30 | 4 | 0 |
| zeEventDestroy | 30 | 30 | 24 | 0 |
| zeEventGetTimestamp | 28 | 28 | 48 | 0 |
| pthread_create | 26 | 26 | 1 | 0 |
| zeEventPoolCreate | 20 | 20 | 4 | 0 |
| zeKernelDestroy | 20 | 20 | 1 | 0 |
| zeModuleDestroy | 17 | 17 | 1 | 0 |
| zeCommandListAppendLaunchKernel | 15 | 15 | 4 | 0 |
| zeCommandListAppendBarrier | 13 | 13 | 8 | 0 |
| zeKernelSuggestGroupSize | 12 | 12 | 4 | 0 |
| zeEventQueryStatus | 11 | 11 | 20 | 0 |
| zeKernelCreate | 11 | 11 | 1 | 0 |
| zeKernelSetGroupSize | 5 | 5 | 4 | 0 |
| zeDeviceGet | 2 | 2 | 2 | 0 |
| zeInit | 2 | 2 | 1 | 0 |
| zeDriverGet | 0 | 0 | 2 | 0 |

Units: microseconds

TAU: ParaProf: Statistics for: node 0, thread 2 - ze_gemm_4096.ppk

| Name | Exclusive TAU... | Inclusive TAUG... | Calls | Child Calls |
|---|---|---|---|---|
| .TAU application | 0.131 | 29.88 | 1 | 24 |
| <Barrier> | 0 | 0 | 8 | 0 |
| <MemoryCopy> | 0.049 | 0.049 | 12 | 0 |
| GEMM | 29.7 | 29.7 | 4 | 0 |

Units: seconds

Time spent in GEMM kernel

% mpirun –np 64 tau_exec –l0 ./a.out

# CUPTI (CUDA Profiling Tools Interface)

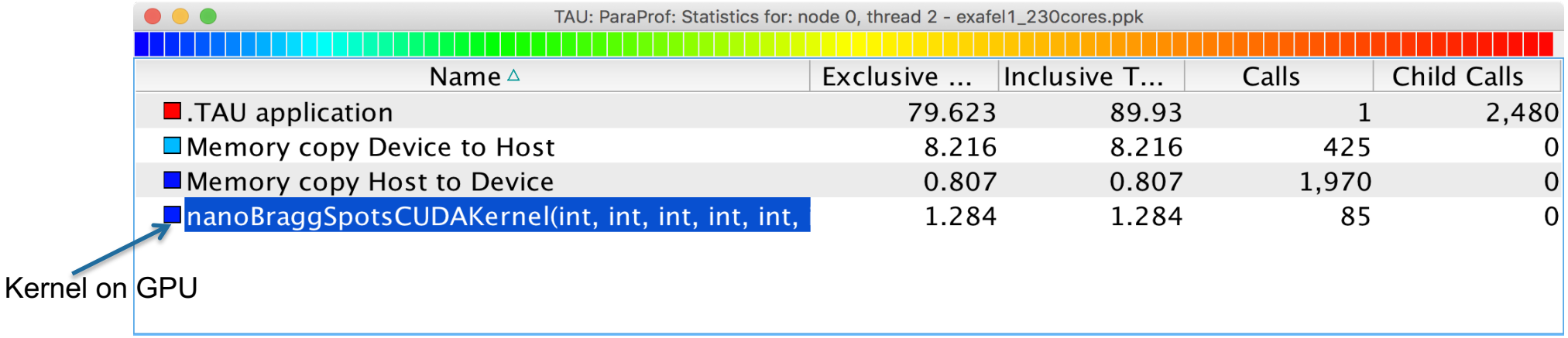
TAU: ParaProf: Statistics for: node 0, thread 0 - exafel1_230cores.ppk

| Name | Exclusive... | Inclusive ... | Calls | Child Calls |
|---|---|---|---|---|
| __init__ [{from_scatterers_fft.py}{13}] | 20.036 | 20.362 | 303 | 10,914 |
| run_sim2smv [{step5_pad.py}{138}] | 16.78 | 134.9 | 1 | 1,066 |
| __init__ [{__init__.py}{150}] | 11.669 | 15.909 | 101 | 1,010 |
| channel_pixels [{step5_pad.py}{79}] | 11.029 | 107.657 | 100 | 13,358 |
| [CONTEXT] channel_pixels [{step5_pad.py}{79}] | 0 | 9.345 | 312 | 0 |
| [SAMPLE] nanoBraggSpotsCUDA [{/autofs/nccs-svm1_home1/iris/adse13_161/psana-legion/simtbx/sun | 4.755 | 4.755 | 159 | 0 |
| [SAMPLE] simtbx::nanoBragg::nanoBragg::add_nanoBragg_spots_cuda() [{/autofs/nccs-svm1_home1/iris/ | 4.08 | 4.08 | 136 | 0 |
| [SAMPLE] __memset_power8 [{} {0}] | 0.3 | 0.3 | 10 | 0 |
| [SAMPLE] UNRESOLVED /usr/lib64/libc-2.17.so | 0.181 | 0.181 | 6 | 0 |
| [SUMMARY] Tau_handle_driver_api_memcpy(void*, CUpti_CallbackDomain, unsigned int, CUpti_CallbackDa | 0.03 | 0.03 | 1 | 0 |
| cuMemcpyDtoH_v2 | 9.483 | 9.483 | 500 | 0 |
| expand_to_p1_iselection [{__init__.py}{1376}] | 7.349 | 7.35 | 101 | 606 |
| load | 7.004 | 7.009 | 2 | 2,251 |
| reset_wavelength [{util_fmodel.py}{121}] | 6.197 | 6.553 | 100 | 47,550 |
| is_unique_set_under_symmetry [{__init__.py}{790}] | 5.913 | 5.915 | 202 | 808 |
| __import__ | 5.782 | 15.766 | 382 | 78 |
| fp_fdp_at_wavelength [{fdp_plot.py}{44}] | 5.616 | 5.723 | 800 | 1,600 |
| MPI_Init_thread() | 4.987 | 4.987 | 1 | 0 |
| cuDevicePrimaryCtxRetain | 4.735 | 4.735 | 2 | 0 |
| <module> [{__init__.py}{1}] | 4.255 | 23.888 | 85 | 756 |
| MPI_Finalize() | 3.829 | 3.829 | 1 | 1 |
| match_bijvoet_mates [{__init__.py}{1032}] | 3.146 | 3.684 | 101 | 707 |
| bcast | 3.073 | 3.448 | 1 | 9 |
| __init__ [{__init__.py}{20}] | 3.011 | 3.399 | 101 | 149,196 |
| compute_f_mask [{__init__.py}{299}] | 2.897 | 18.853 | 101 | 707 |

Python, MPI, CUDA, and samples from DSOs are all integrated in a single view
% mpirun –np 64 tau_python –cupti ./exafel.py

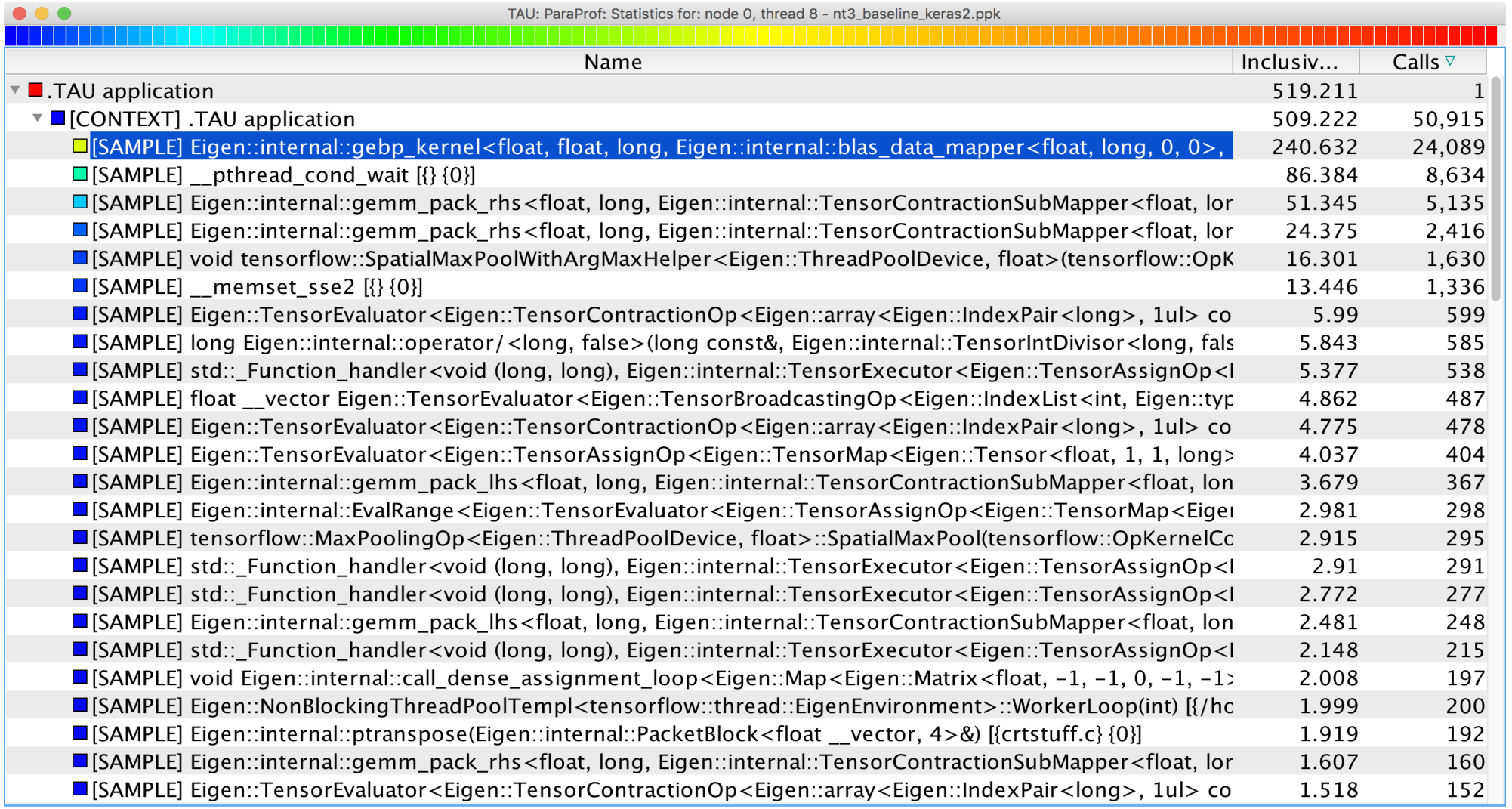
# TAU supports Python, MPI, and CUDA

Without any modification to the source code or DSOs or interpreter, it instruments and samples the application using Python, MPI, and CUDA instrumentation.




% mpirun –np 230 tau_python –T cupti,mpi,pdt –ebs –cupti ./exafel.py
Instead of:
% mpirun –np 230 python ./exafel.py

# Deep Learning: Tensorflow




| Name | Inclusiv... | Calls |
|---|---|---|
| .TAU application | 519.211 | 1 |
| [CONTEXT] .TAU application | 509.222 | 50,915 |
| [SAMPLE] Eigen::internal::gebp_kernel<float, float, long, Eigen::internal::blas_data_mapper<float, long, 0, 0>, | 240.632 | 24,089 |
| [SAMPLE] __pthread_cond_wait [{} {0}] | 86.384 | 8,634 |
| [SAMPLE] Eigen::internal::gemm_pack_rhs<float, long, Eigen::internal::TensorContractionSubMapper<float, lor | 51.345 | 5,135 |
| [SAMPLE] Eigen::internal::gemm_pack_rhs<float, long, Eigen::internal::TensorContractionSubMapper<float, lor | 24.375 | 2,416 |
| [SAMPLE] void tensorflow::SpatialMaxPoolWithArgMaxHelper<Eigen::ThreadPoolDevice, float>(tensorflow::OpK | 16.301 | 1,630 |
| [SAMPLE] __memset_sse2 [{} {0}] | 13.446 | 1,336 |
| [SAMPLE] Eigen::TensorEvaluator<Eigen::TensorContractionOp<Eigen::array<Eigen::IndexPair<long>, 1ul> co | 5.99 | 599 |
| [SAMPLE] long Eigen::internal::operator/<long, false>(long const&, Eigen::internal::TensorIntDivisor<long, fals | 5.843 | 585 |
| [SAMPLE] std::_Function_handler<void (long, long), Eigen::internal::TensorExecutor<Eigen::TensorAssignOp<I | 5.377 | 538 |
| [SAMPLE] float __vector Eigen::TensorEvaluator<Eigen::TensorBroadcastingOp<Eigen::IndexList<int, Eigen::typ | 4.862 | 487 |
| [SAMPLE] Eigen::TensorEvaluator<Eigen::TensorContractionOp<Eigen::array<Eigen::IndexPair<long>, 1ul> co | 4.775 | 478 |
| [SAMPLE] Eigen::TensorEvaluator<Eigen::TensorAssignOp<Eigen::TensorMap<Eigen::Tensor<float, 1, 1, long> | 4.037 | 404 |
| [SAMPLE] Eigen::internal::gemm_pack_lhs<float, long, Eigen::internal::TensorContractionSubMapper<float, lon | 3.679 | 367 |
| [SAMPLE] Eigen::internal::EvalRange<Eigen::TensorEvaluator<Eigen::TensorAssignOp<Eigen::TensorMap<Eiger | 2.981 | 298 |
| [SAMPLE] tensorflow::MaxPoolingOp<Eigen::ThreadPoolDevice, float>::SpatialMaxPool(tensorflow::OpKernelCo | 2.915 | 295 |
| [SAMPLE] std::_Function_handler<void (long, long), Eigen::internal::TensorExecutor<Eigen::TensorAssignOp<I | 2.91 | 291 |
| [SAMPLE] std::_Function_handler<void (long, long), Eigen::internal::TensorExecutor<Eigen::TensorAssignOp<I | 2.772 | 277 |
| [SAMPLE] Eigen::internal::gemm_pack_lhs<float, long, Eigen::internal::TensorContractionSubMapper<float, lon | 2.481 | 248 |
| [SAMPLE] std::_Function_handler<void (long, long), Eigen::internal::TensorExecutor<Eigen::TensorAssignOp<I | 2.148 | 215 |
| [SAMPLE] void Eigen::internal::call_dense_assignment_loop<Eigen::Map<Eigen::Matrix<float, -1, -1, 0, -1, -1> | 2.008 | 197 |
| [SAMPLE] Eigen::NonBlockingThreadPoolTempl<tensorflow::thread::EigenEnvironment>::WorkerLoop(int) [{/ho | 1.999 | 200 |
| [SAMPLE] Eigen::internal::ptranspose(Eigen::internal::PacketBlock<float __vector, 4>&) [{crtstuff.c} {0}] | 1.919 | 192 |
| [SAMPLE] Eigen::internal::gemm_pack_rhs<float, long, Eigen::internal::TensorContractionSubMapper<float, lor | 1.607 | 160 |
| [SAMPLE] Eigen::TensorEvaluator<Eigen::TensorContractionOp<Eigen::array<Eigen::IndexPair<long>, 1ul> co | 1.518 | 152 |

```
% tau_python -ebs nt3_baseline_keras2.py
```

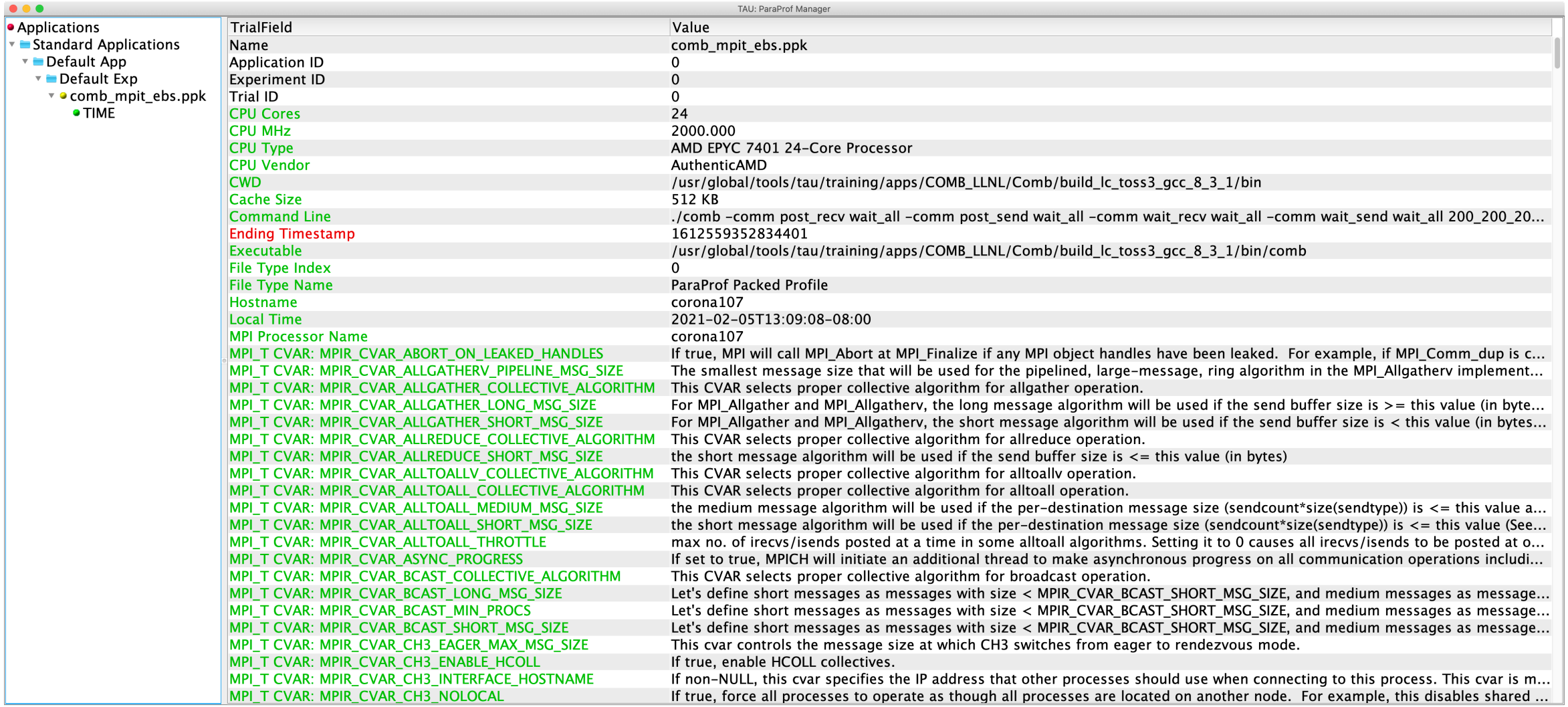
# MPI Tools Interface: PVARs and CVARs

TAU: ParaProf: Context Events for: node 2, thread 0 - comb_mpit_ebs.ppk

| Name | Total | NumSamples | MaxValue | MinValue | MeanValue | Std. Dev. |
|---|---|---|---|---|---|---|
| Message size for broadcast | 4 | 1 | 4 | 4 | 4 | 0 |
| Message size for reduce | 608 | 20 | 48 | 8 | 30.4 | 15.513 |
| mpit_progress_poll (CH3 RDMA progress engine polling count) | 474,499 | 1 | 474,499 | 474,499 | 474,499 | 0 |
| mv2_coll_allgather_bytes_recv (Number of bytes recv by default algorithm of allgathe | 612 | 1 | 612 | 612 | 612 | 0 |
| mv2_coll_allgather_bytes_send (Number of bytes send by default algorithm of allgath | 252 | 1 | 252 | 252 | 252 | 0 |
| mv2_coll_allgather_count_recv (Count of messages recv by default algorithm of allgat | 21 | 1 | 21 | 21 | 21 | 0 |
| mv2_coll_allgather_count_send (Count of messages send by default algorithm of allga | 21 | 1 | 21 | 21 | 21 | 0 |
| mv2_coll_allgather_rd (Number of times recursive doubling Allgather was invoked) | 7 | 1 | 7 | 7 | 7 | 0 |
| mv2_coll_allgather_rd_bytes_recv (Number of bytes recv by rd algorithm of allgather | 612 | 1 | 612 | 612 | 612 | 0 |
| mv2_coll_allgather_rd_bytes_send (Number of bytes send by rd algorithm of allgather | 252 | 1 | 252 | 252 | 252 | 0 |
| mv2_coll_allgather_rd_count_recv (Count of messages recv by rd algorithm of allgath | 21 | 1 | 21 | 21 | 21 | 0 |
| mv2_coll_allgather_rd_count_send (Count of messages send by rd algorithm of allgat | 21 | 1 | 21 | 21 | 21 | 0 |
| mv2_coll_allreduce_2lvl (Number of times MV2 two-level allreduce algorithm was invc | 5 | 1 | 5 | 5 | 5 | 0 |
| mv2_coll_allreduce_bytes_recv (Number of bytes recv by allreduce collective) | 7,080 | 1 | 7,080 | 7,080 | 7,080 | 0 |
| mv2_coll_allreduce_bytes_send (Number of bytes send by allreduce collective) | 7,080 | 1 | 7,080 | 7,080 | 7,080 | 0 |
| mv2_coll_allreduce_count_recv (Count of messages recv by allreduce collective) | 42 | 1 | 42 | 42 | 42 | 0 |
| mv2_coll_allreduce_count_send (Count of messages send by allreduce collective) | 42 | 1 | 42 | 42 | 42 | 0 |
| mv2_coll_allreduce_pt2pt_rd_bytes_recv (Number of bytes recv by pt2pt rd algorithm | 7,080 | 1 | 7,080 | 7,080 | 7,080 | 0 |
| mv2_coll_allreduce_pt2pt_rd_bytes_send (Number of bytes send by pt2pt rd algorithn | 7,080 | 1 | 7,080 | 7,080 | 7,080 | 0 |
| mv2_coll_allreduce_pt2pt_rd_count_recv (Count of messages recv by pt2pt rd algorith | 42 | 1 | 42 | 42 | 42 | 0 |
| mv2_coll_allreduce_pt2pt_rd_count_send (Count of messages send by pt2pt rd algorit | 42 | 1 | 42 | 42 | 42 | 0 |
| mv2_coll_allreduce_shm_intra (Number of times MV2 shm intra allreduce algorithm w | 3 | 1 | 3 | 3 | 3 | 0 |
| mv2_coll_allreduce_shm_rd (Number of times MV2 shm rd allreduce algorithm was in | 14 | 1 | 14 | 14 | 14 | 0 |
| mv2_coll_allreduce_subcomm (Number of times MV2 allreduce was invoked at a sub- | 10 | 1 | 10 | 10 | 10 | 0 |
| mv2_coll_barrier_count_recv (Count of messages recv by barrier collective) | 24 | 1 | 24 | 24 | 24 | 0 |
| mv2_coll_barrier_count_send (Count of messages send by barrier collective) | 24 | 1 | 24 | 24 | 24 | 0 |
| mv2_coll_barrier_pairwise (Number of times pairwise barrier was invoked) | 8 | 1 | 8 | 8 | 8 | 0 |
| mv2_coll_barrier_pairwise_count_recv (Count of messages recv by pairwise algorithm | 24 | 1 | 24 | 24 | 24 | 0 |
| mv2_coll_barrier_pairwise_count_send (Count of messages send by pairwise algorithn | 24 | 1 | 24 | 24 | 24 | 0 |

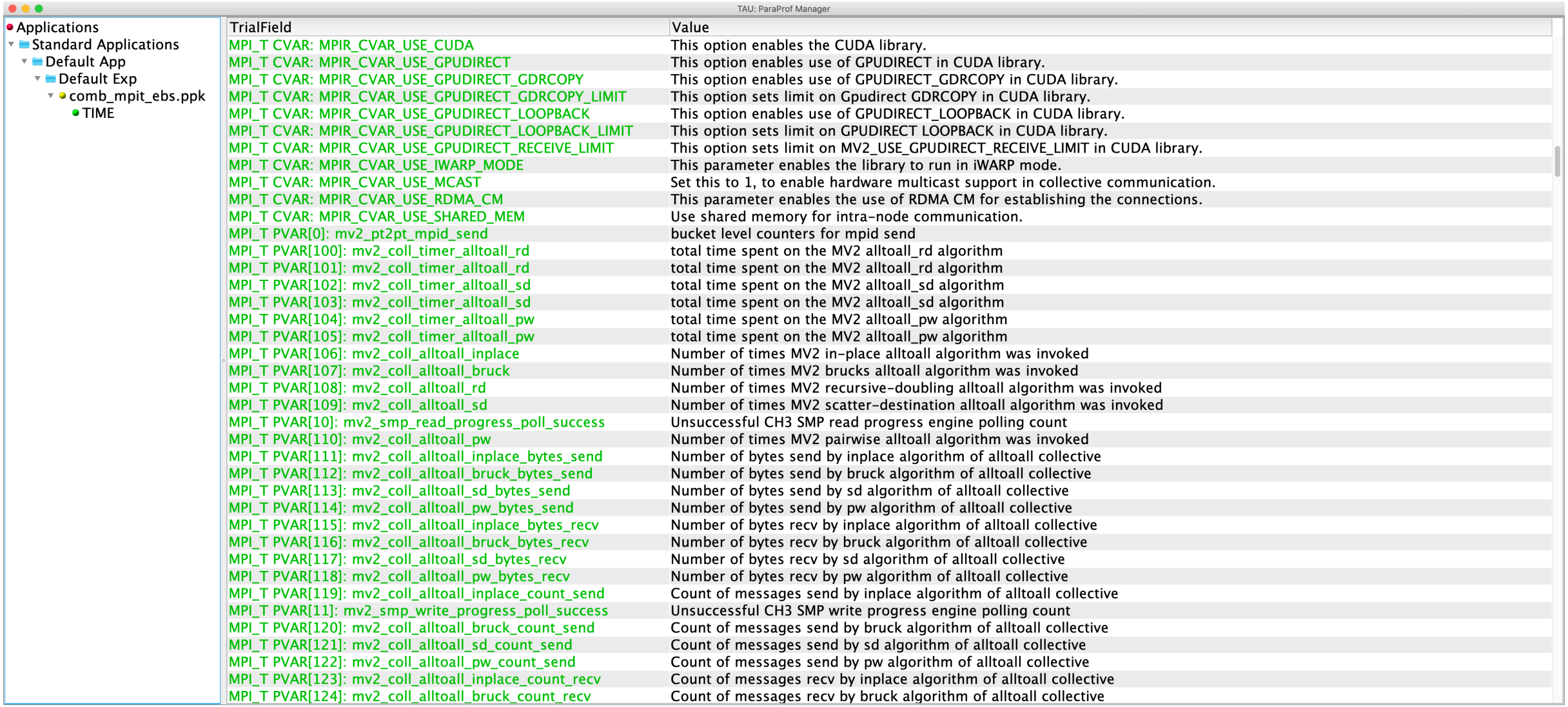
```
% export TAU_TRACK_MPI_T_PVARS=1
% mpirun –np 64 tau_exec –T mpit ./a.out
```

# MPI Tools Interface: Control Variables (CVARs)




| TrialField | Value |
|---|---|
| Name | comb_mpit_ebs.ppk |
| Application ID | 0 |
| Experiment ID | 0 |
| Trial ID | 0 |
| CPU Cores | 24 |
| CPU MHz | 2000.000 |
| CPU Type | AMD EPYC 7401 24-Core Processor |
| CPU Vendor | AuthenticAMD |
| CWD | /usr/global/tools/tau/training/apps/COMB_LLNL/Comb/build_lc_toss3_gcc_8_3_1/bin |
| Cache Size | 512 KB |
| Command Line | ./comb -comm post_recv wait_all -comm post_send wait_all -comm wait_recv wait_all -comm wait_send wait_all 200_200_20... |
| Ending Timestamp | 1612559352834401 |
| Executable | /usr/global/tools/tau/training/apps/COMB_LLNL/Comb/build_lc_toss3_gcc_8_3_1/bin/comb |
| File Type Index | 0 |
| File Type Name | ParaProf Packed Profile |
| Hostname | corona107 |
| Local Time | 2021-02-05T13:09:08-08:00 |
| MPI Processor Name | corona107 |
| MPI_T CVAR: MPIR_CVAR_ABORT_ON_LEAKED_HANDLES | If true, MPI will call MPI_Abort at MPI_Finalize if any MPI object handles have been leaked. For example, if MPI_Comm_dup is c... |
| MPI_T CVAR: MPIR_CVAR_ALLGATHERV_PIPELINE_MSG_SIZE | The smallest message size that will be used for the pipelined, large-message, ring algorithm in the MPI_Allgatherv implement... |
| MPI_T CVAR: MPIR_CVAR_ALLGATHER_COLLECTIVE_ALGORITHM | This CVAR selects proper collective algorithm for allgather operation. |
| MPI_T CVAR: MPIR_CVAR_ALLGATHER_LONG_MSG_SIZE | For MPI_Allgather and MPI_Allgatherv, the long message algorithm will be used if the send buffer size is >= this value (in byte... |
| MPI_T CVAR: MPIR_CVAR_ALLGATHER_SHORT_MSG_SIZE | For MPI_Allgather and MPI_Allgatherv, the short message algorithm will be used if the send buffer size is < this value (in bytes... |
| MPI_T CVAR: MPIR_CVAR_ALLREDUCE_COLLECTIVE_ALGORITHM | This CVAR selects proper collective algorithm for allreduce operation. |
| MPI_T CVAR: MPIR_CVAR_ALLREDUCE_SHORT_MSG_SIZE | the short message algorithm will be used if the send buffer size is <= this value (in bytes) |
| MPI_T CVAR: MPIR_CVAR_ALLTOALLV_COLLECTIVE_ALGORITHM | This CVAR selects proper collective algorithm for alltoallv operation. |
| MPI_T CVAR: MPIR_CVAR_ALLTOALL_COLLECTIVE_ALGORITHM | This CVAR selects proper collective algorithm for alltoall operation. |
| MPI_T CVAR: MPIR_CVAR_ALLTOALL_MEDIUM_MSG_SIZE | the medium message algorithm will be used if the per-destination message size (sendcount*size(sendtype)) is <= this value a... |
| MPI_T CVAR: MPIR_CVAR_ALLTOALL_SHORT_MSG_SIZE | the short message algorithm will be used if the per-destination message size (sendcount*size(sendtype)) is <= this value (See... |
| MPI_T CVAR: MPIR_CVAR_ALLTOALL_THROTTLE | max no. of irecvs/isends posted at a time in some alltoall algorithms. Setting it to 0 causes all irecvs/isends to be posted at o... |
| MPI_T CVAR: MPIR_CVAR_ASYNC_PROGRESS | If set to true, MPICH will initiate an additional thread to make asynchronous progress on all communication operations includi... |
| MPI_T CVAR: MPIR_CVAR_BCAST_COLLECTIVE_ALGORITHM | This CVAR selects proper collective algorithm for broadcast operation. |
| MPI_T CVAR: MPIR_CVAR_BCAST_LONG_MSG_SIZE | Let's define short messages as messages with size < MPIR_CVAR_BCAST_SHORT_MSG_SIZE, and medium messages as message... |
| MPI_T CVAR: MPIR_CVAR_BCAST_MIN_PROCS | Let's define short messages as messages with size < MPIR_CVAR_BCAST_SHORT_MSG_SIZE, and medium messages as message... |
| MPI_T CVAR: MPIR_CVAR_BCAST_SHORT_MSG_SIZE | Let's define short messages as messages with size < MPIR_CVAR_BCAST_SHORT_MSG_SIZE, and medium messages as message... |
| MPI_T CVAR: MPIR_CVAR_CH3_EAGER_MAX_MSG_SIZE | This cvar controls the message size at which CH3 switches from eager to rendezvous mode. |
| MPI_T CVAR: MPIR_CVAR_CH3_ENABLE_HCOLL | If true, enable HCOLL collectives. |
| MPI_T CVAR: MPIR_CVAR_CH3_INTERFACE_HOSTNAME | If non-NULL, this cvar specifies the IP address that other processes should use when connecting to this process. This cvar is m... |
| MPI_T CVAR: MPIR_CVAR_CH3_NOLOCAL | If true, force all processes to operate as though all processes are located on another node. For example, this disables shared ... |

# MPI Tools Interface: Performance Variables (PVARs)




| TrialField | Value |
|---|---|
| MPI_T CVAR: MPIR_CVAR_USE_CUDA | This option enables the CUDA library. |
| MPI_T CVAR: MPIR_CVAR_USE_GPUDIRECT | This option enables use of GPUDIRECT in CUDA library. |
| MPI_T CVAR: MPIR_CVAR_USE_GPUDIRECT_GDRCOPY | This option enables use of GPUDIRECT_GDRCOPY in CUDA library. |
| MPI_T CVAR: MPIR_CVAR_USE_GPUDIRECT_GDRCOPY_LIMIT | This option sets limit on Gpudirect GDRCOPY in CUDA library. |
| MPI_T CVAR: MPIR_CVAR_USE_GPUDIRECT_LOOPBACK | This option enables use of GPUDIRECT_LOOPBACK in CUDA library. |
| MPI_T CVAR: MPIR_CVAR_USE_GPUDIRECT_LOOPBACK_LIMIT | This option sets limit on GPUDIRECT LOOPBACK in CUDA library. |
| MPI_T CVAR: MPIR_CVAR_USE_GPUDIRECT_RECEIVE_LIMIT | This option sets limit on MV2_USE_GPUDIRECT_RECEIVE_LIMIT in CUDA library. |
| MPI_T CVAR: MPIR_CVAR_USE_IWARP_MODE | This parameter enables the library to run in iWARP mode. |
| MPI_T CVAR: MPIR_CVAR_USE_MCAST | Set this to 1, to enable hardware multicast support in collective communication. |
| MPI_T CVAR: MPIR_CVAR_USE_RDMA_CM | This parameter enables the use of RDMA CM for establishing the connections. |
| MPI_T CVAR: MPIR_CVAR_USE_SHARED_MEM | Use shared memory for intra-node communication. |
| MPI_T PVAR[0]: mv2_pt2pt_mpid_send | bucket level counters for mpid send |
| MPI_T PVAR[100]: mv2_coll_timer_alltoall_rd | total time spent on the MV2 alltoall_rd algorithm |
| MPI_T PVAR[101]: mv2_coll_timer_alltoall_rd | total time spent on the MV2 alltoall_rd algorithm |
| MPI_T PVAR[102]: mv2_coll_timer_alltoall_sd | total time spent on the MV2 alltoall_sd algorithm |
| MPI_T PVAR[103]: mv2_coll_timer_alltoall_sd | total time spent on the MV2 alltoall_sd algorithm |
| MPI_T PVAR[104]: mv2_coll_timer_alltoall_pw | total time spent on the MV2 alltoall_pw algorithm |
| MPI_T PVAR[105]: mv2_coll_timer_alltoall_pw | total time spent on the MV2 alltoall_pw algorithm |
| MPI_T PVAR[106]: mv2_coll_alltoall_inplace | Number of times MV2 in-place alltoall algorithm was invoked |
| MPI_T PVAR[107]: mv2_coll_alltoall_bruck | Number of times MV2 brucks alltoall algorithm was invoked |
| MPI_T PVAR[108]: mv2_coll_alltoall_rd | Number of times MV2 recursive-doubling alltoall algorithm was invoked |
| MPI_T PVAR[109]: mv2_coll_alltoall_sd | Number of times MV2 scatter-destination alltoall algorithm was invoked |
| MPI_T PVAR[10]: mv2_smp_read_progress_poll_success | Unsuccessful CH3 SMP read progress engine polling count |
| MPI_T PVAR[110]: mv2_coll_alltoall_pw | Number of times MV2 pairwise alltoall algorithm was invoked |
| MPI_T PVAR[111]: mv2_coll_alltoall_inplace_bytes_send | Number of bytes send by inplace algorithm of alltoall collective |
| MPI_T PVAR[112]: mv2_coll_alltoall_bruck_bytes_send | Number of bytes send by bruck algorithm of alltoall collective |
| MPI_T PVAR[113]: mv2_coll_alltoall_sd_bytes_send | Number of bytes send by sd algorithm of alltoall collective |
| MPI_T PVAR[114]: mv2_coll_alltoall_pw_bytes_send | Number of bytes send by pw algorithm of alltoall collective |
| MPI_T PVAR[115]: mv2_coll_alltoall_inplace_bytes_recv | Number of bytes recv by inplace algorithm of alltoall collective |
| MPI_T PVAR[116]: mv2_coll_alltoall_bruck_bytes_recv | Number of bytes recv by bruck algorithm of alltoall collective |
| MPI_T PVAR[117]: mv2_coll_alltoall_sd_bytes_recv | Number of bytes recv by sd algorithm of alltoall collective |
| MPI_T PVAR[118]: mv2_coll_alltoall_pw_bytes_recv | Number of bytes recv by pw algorithm of alltoall collective |
| MPI_T PVAR[119]: mv2_coll_alltoall_inplace_count_send | Count of messages send by inplace algorithm of alltoall collective |
| MPI_T PVAR[11]: mv2_smp_write_progress_poll_success | Unsuccessful CH3 SMP write progress engine polling count |
| MPI_T PVAR[120]: mv2_coll_alltoall_bruck_count_send | Count of messages send by bruck algorithm of alltoall collective |
| MPI_T PVAR[121]: mv2_coll_alltoall_sd_count_send | Count of messages send by sd algorithm of alltoall collective |
| MPI_T PVAR[122]: mv2_coll_alltoall_pw_count_send | Count of messages send by pw algorithm of alltoall collective |
| MPI_T PVAR[123]: mv2_coll_alltoall_inplace_count_recv | Count of messages recv by inplace algorithm of alltoall collective |
| MPI_T PVAR[124]: mv2_coll_alltoall_bruck_count_recv | Count of messages recv by bruck algorithm of alltoall collective |

# 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>) |

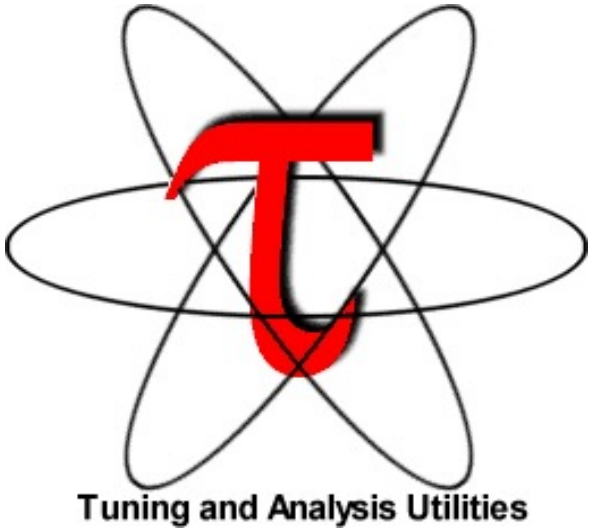
# Runtime Environment Variables

| 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 |

# Download TAU from U. Oregon



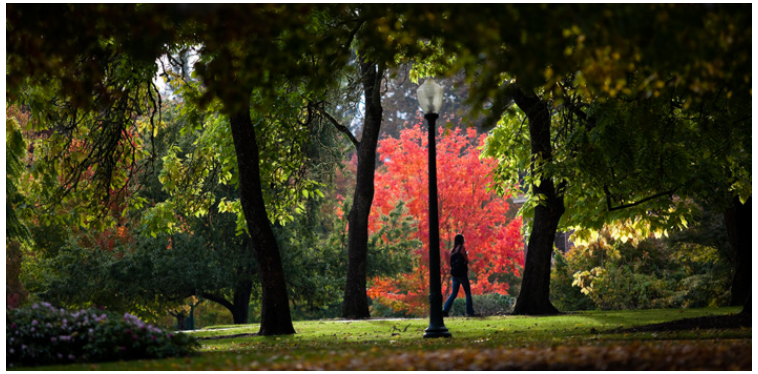

http://tau.uoregon.edu

https://e4s.io [TAU in Docker/Singularity containers]

**for more information**

**Free download, open source, BSD license**

# Performance Research Laboratory, University of Oregon, Eugene




www.uoregon.edu

# Support Acknowledgements


- US Department of Energy (DOE)
  - ANL
  - Office of Science contracts, ECP
  - SciDAC, LBL contracts
  - LLNL-LANL-SNL ASC/NNSA contract
  - Battelle, PNNL and ORNL contract
- Department of Defense (DoD)
  - PETTT, HPCMP
- National Science Foundation (NSF)
  - SI2-SSI, Glassbox
- NASA
- CEA, France
- Partners:
  - –University of Oregon
  - –The Ohio State University
  - –ParaTools, Inc.
  - –University of Tennessee, Knoxville
  - –T.U. Dresden, GWT
  - –Jülich Supercomputing Center


# Acknowledgment


“This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nation’s exascale computing imperative.”