TAU Performance System®

Sameer Shende
University of Oregon and ParaTools, Inc.
sameer@cs.uoregon.edu
http://tau.uoregon.edu

ACB, TACC, Austin, Texas, Thursday, September 28, 1pm – 3pm

Download slides from:

http://tau.uoregon.edu/tau.pdf

Stampede2: tar xf ~tg457572/workshop.tgz
cp ~tg457572/tau.pdf .

Acknowledgements

- TACC, U. Texas
- University of Oregon
- ParaTools, Inc.
- ParaTools, SAS, France
- The Ohio State University
- NSF (SI2, XSEDE)
- U.S. Department of Energy
  - LLNL, LANL, SNL
  - ALCF/ANL, PNNL, ORNL, NERSC/LBL
- U.S. Department of Defense
- NASA

Outline

- Introduction to TAU Commander
- Instrumentation: PDT, MPI, OpenMP OMPT
- Event-based sampling
- PDT based source instrumentation
- Selective instrumentation
- Compiler-based Instrumentation
- Event-tracing with Jumpshot and Vampir
- Conclusions

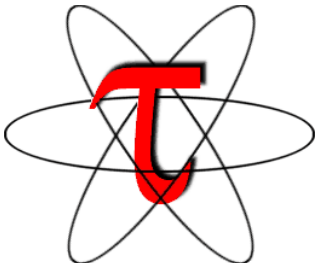
Tutorial Goals

This tutorial is an introduction to portable performance evaluation tools. You should leave here with a better understanding of…

- Concepts and steps involved in performance evaluation
- Understanding key concepts in understanding code performance
- How to collect and analyze data from hardware performance counters (PAPI)
- How to instrument your programs with TAU
- Measurement options provided by TAU
- Environment variables used for choosing metrics, generating performance data
- How to use ParaProf, TAU's profile browser
- General familiarity with TAU use for Fortran, C++, C, and mixed language
- How to generate trace data in different formats

TAU Performance System®

http://tau.uoregon.edu



- **Tuning and Analysis Utilities (20+ year project)**
- **Comprehensive performance profiling and tracing**
  - Integrated, scalable, flexible, portable
  - Targets all parallel programming/execution paradigms
- **Integrated performance toolkit**
  - Instrumentation, measurement, analysis, visualization
  - Widely-ported performance profiling / tracing system
  - Performance data management and data mining
  - Open source (BSD-style license)
- **Integrates with application frameworks**

Understanding Application Performance using TAU

- **How much time** is spent in each application routine and outer *loops*? Within loops, what is the contribution of each *statement*?
- **How many instructions** are executed in these code regions? Floating point, Level 1 and 2 *data cache misses*, hits, branches taken?
- **What is the memory usage** of the code? When and where is memory allocated/de-allocated? Are there any memory leaks?
- **What are the I/O characteristics** of the code? What is the peak read and write *bandwidth* of individual calls, total volume?
- **What is the contribution of each *phase*** of the program? What is the time wasted/spent waiting for collectives, and I/O operations in Initialization, Computation, I/O phases?
- **How does the application *scale***? What is the efficiency, runtime breakdown of performance across different core counts?

TAU Commander

Universal tool or integrated toolkit

Unbiased, accurate measurements

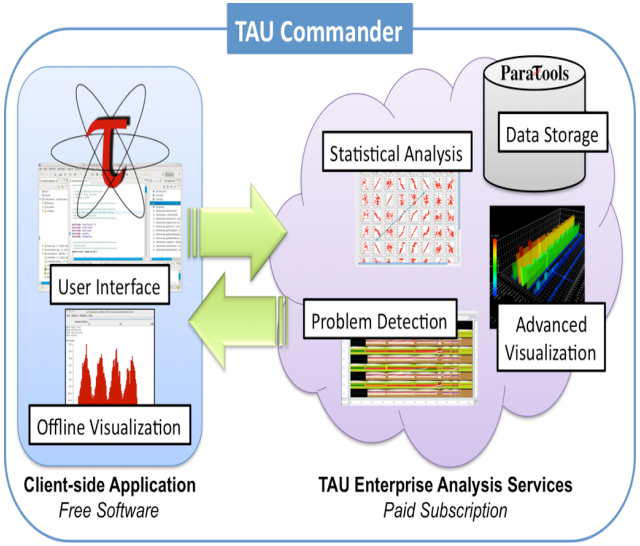
- File I/O: serial and parallel
- Communication: inter- and intra-node
- Memory: allocation and access
- CPU: vectorization, cache utilization, etc.

Minimal overhead

- Provide multiple measurement methods
- Focus on one performance aspect at a time

Easy to use

- Intuitive, systematic, and well documented
- Easy to understand and configure
- BSD style license, Github open source
- ***http://www.taucommander.com***

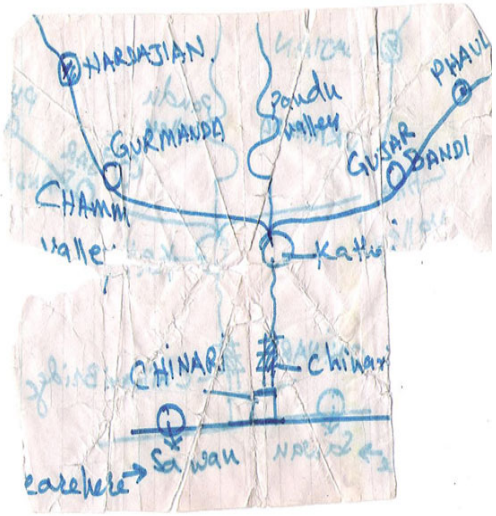


TAU Commander's Approach

Say where you're going, not how to get there

Experiments give context to the user's actions

- Defines desired metrics and measurement approach
- Defines operating environment
- Establishes a baseline for error checking



VS.

T-A-M Model for Performance Engineering

Target

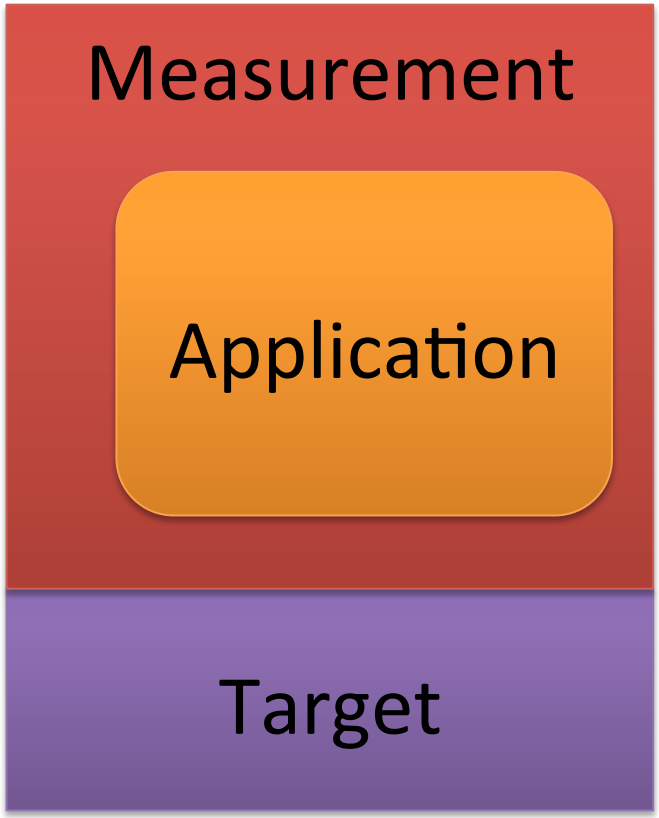
- Installed software
- Available compilers
- Host architecture/ OS

Application

- MPI, OpenMP, CUDA, OpenACC, etc.

Measurement

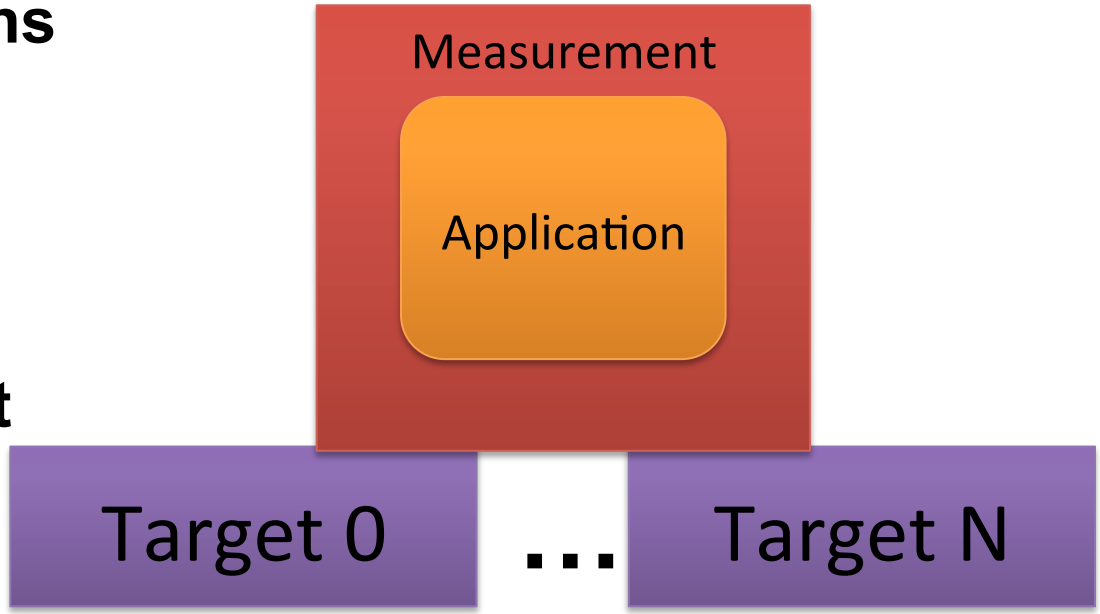
- Profile, trace, or both
- Sample, source inst…



Experiment = (Target, Application, Measurement)

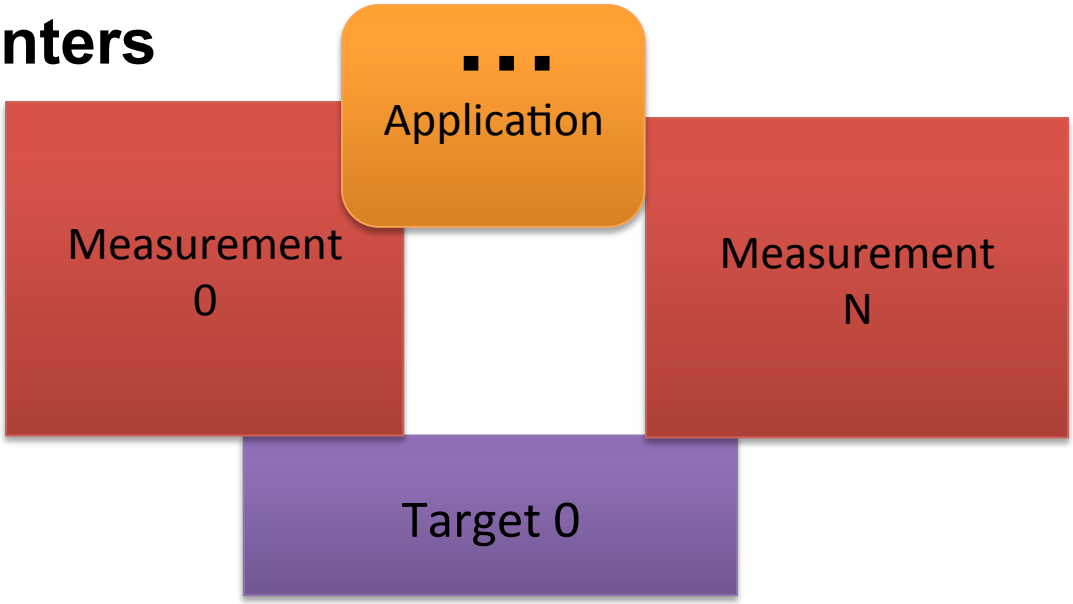
Which platform is best for my application?

- Many targets:
  - Different MPI implementations
  - Different CPU architectures
  - GPU vs MIC
  - Cray vs SGI
- One measurement
- One application



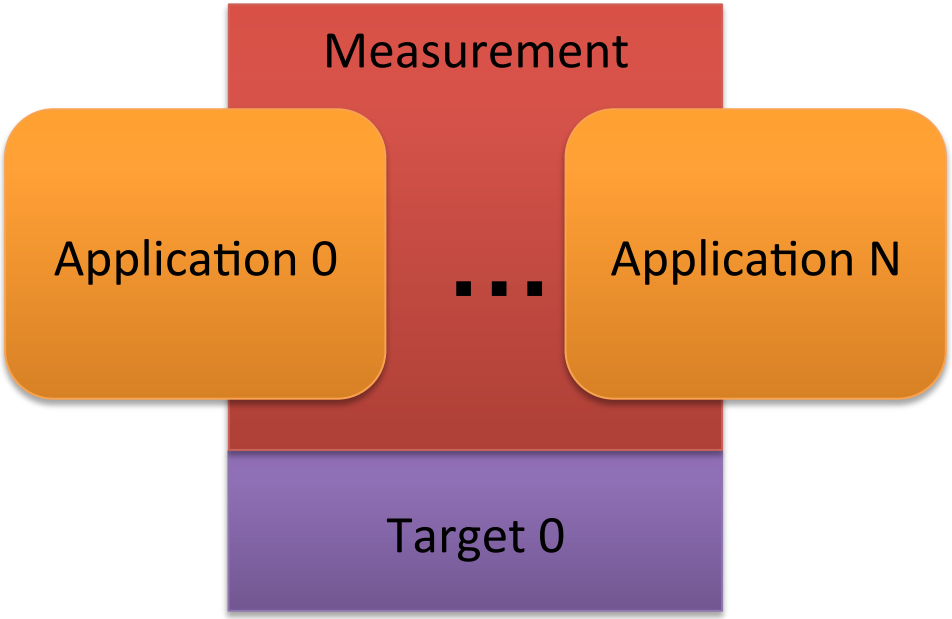
What are the performance characteristics of my application?

- One target
- Many measurements:
  - File I/O
  - Communication
  - Memory allocation
  - Performance counters
  - Vectorization
- One application



How well does my target perform various tasks?

- One target
- One measurement
- Many applications:
  - Compute bound
    - Dense LA
  - Memory bound
    - Sparse LA
    - Graph
  - Scaling
    - Thread-level
    - Process-level



Getting Started with TAU Commander

1. tau initialize
2. tau oshf90 *.f90 -o foo
3. tau oshrun -np 64 ./foo
4. tau show

Just put `tau` in front of everything and see what happens.

This works on any supported system, even if TAU is not installed or has not been configured appropriately.

TAU and all its dependencies will be downloaded and installed if required.

TAU Commander Online Help

jlinford — ssh cori.nersc.gov — 80×47

```
jlinford@cori09 ~/workspace/openshmem17/applications/ISx $ tau --help
usage: tau [arguments] <subcommand> [options]

TAU Commander 1.0a [ www.taucommander.com ]

Positional Arguments:
  <subcommand>   See subcommand descriptions below.
  [options]      Options to be passed to <subcommand>.

Optional Arguments:
  -V, --version  Show program's version number and exit.
  -h, --help     Show this help message and exit.
  -q, --quiet    Suppress all output except error messages.
  -v, --verbose  Show debugging messages.

Configuration Subcommands:
  application     Create and manage application configurations.
  experiment      Create and manage experiments.
  measurement     Create and manage measurement configurations.
  project         Create and manage project configurations.
  target          Create and manage target configurations.
  trial           Create and manage experiment trials.

Subcommands:
  build           Instrument programs during compilation and/or linking.
  configure       Configure TAU Commander.
  dashboard       Show all project components.
  help            Show help for a command or suggest actions for a file.
  initialize      Initialize TAU Commander.
  select          Create a new experiment or select an existing experiment.

Shortcuts:
  tau <compiler>  Execute a compiler command
                  - Example: tau gcc *.c -o a.out
                  - Alias for 'tau build <compiler>'
  tau <program>   Gather data from a program
                  - Example: tau ./a.out
                  - Alias for 'tau trial create <program>'
  tau metrics     Show metrics available in the current experiment
                  - Alias for 'tau target metrics'
  tau select      Select configuration objects to create a new experiment
                  - Alias for 'tau experiment create'
  tau show        Show data from the most recent trial
                  - Alias for 'tau trial show'

See 'tau help <subcommand>' for more information on <subcommand>.
jlinford@cori09 ~/workspace/openshmem17/applications/ISx $
```

jlinford — ssh cori.nersc.gov — 80×35

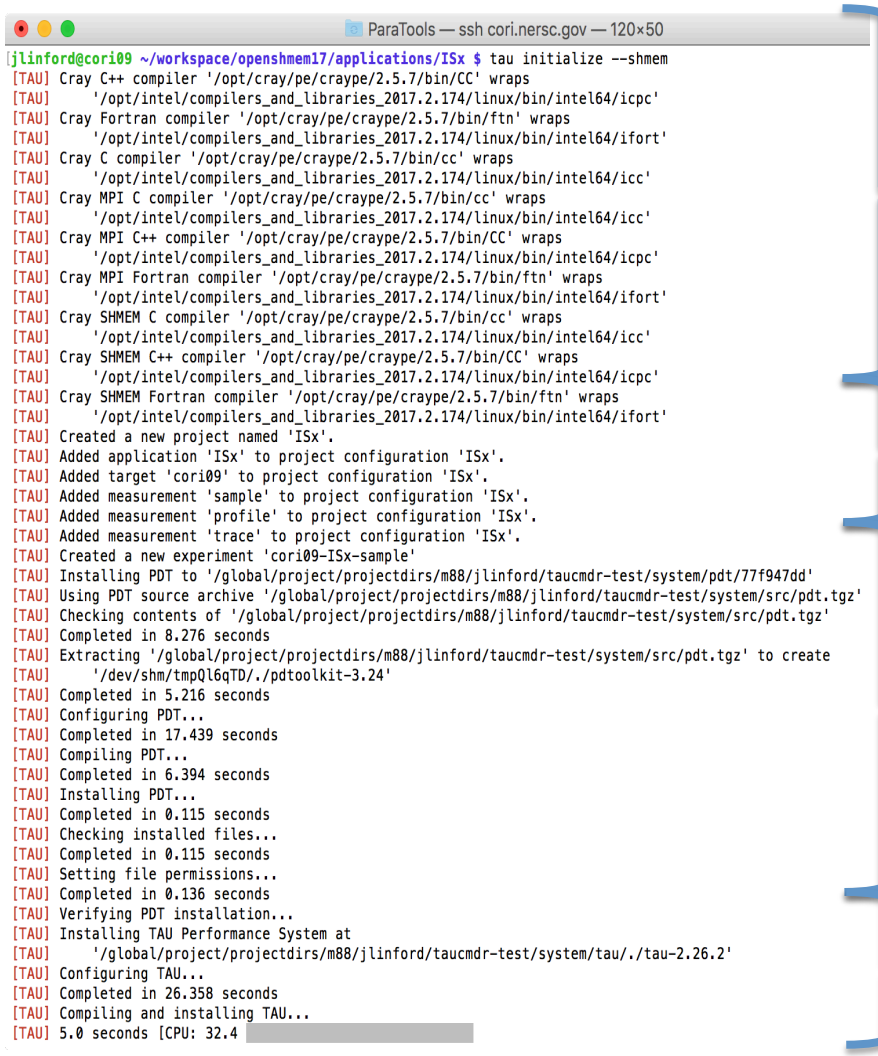
```
jlinford@cori09 ~/workspace/openshmem17/applications/ISx $ tau app cre --help
usage: tau application create <application_name> [arguments]

Create application configurations.

Optional Arguments:
  -@ <level>           Create the application at the specified storage
                       level.
                         - <level>: project, user, system
                         - default: project
  -h, --help           Show this help message and exit.

Application Arguments:
  <application_name>   Application configuration name.
  --cuda [T/F]         Application uses NVIDIA CUDA.
                         - default: False
  --linkage <linkage>  Application linkage.
                         - <linkage>: static, dynamic
                         - default: static
  --mpc [T/F]          Application uses MPC.
                         - default: False
  --mpi [T/F]          Application uses MPI.
                         - default: False
  --opencl [T/F]       Application uses OpenCL.
                         - default: False
  --openmp [T/F]       Application uses OpenMP.
                         - default: False
  --pthreads [T/F]     Application uses pthreads.
                         - default: False
  --select-file path   Specify selective instrumentation file.
  --shmem [T/F]        Application uses SHMEM.
                         - default: False
  --tbb [T/F]          Application uses Thread Building Blocks (TBB).
                         - default: False
jlinford@cori09 ~/workspace/openshmem17/applications/ISx $
```

Step 1: Initialize TAU Project

```
$ tau initialize --mpi --compilers Intel
                 --mpi-compilers   Intel
$ tau init --shmem
```

- Creates a new project configuration using defaults
- Project files exist in a directory named ".tau"
- Like git, all directories below the directory containing the ".tau" directory can access the project
  - E.g. `tau dashboard` works in miniapp1/baseline

Project Initialization

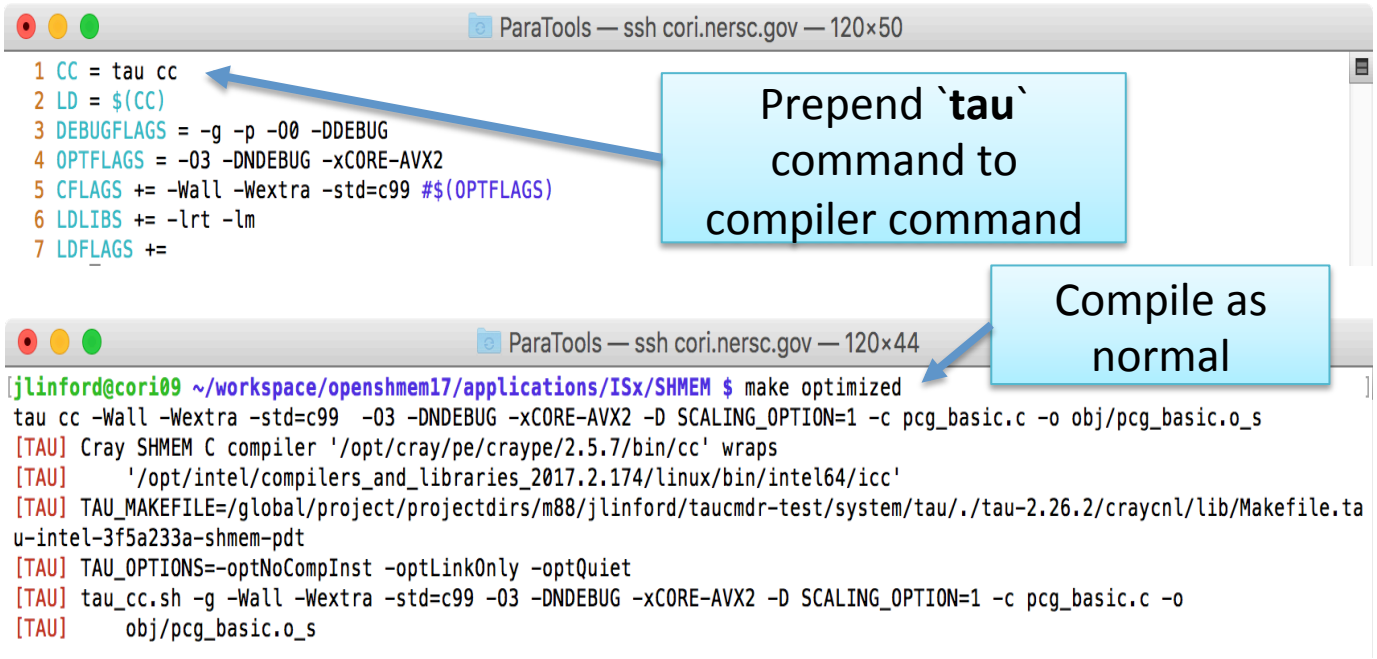


Project Dashboard (`tau dashboard`)

ParaTools — ssh cori.nersc.gov — 120×50

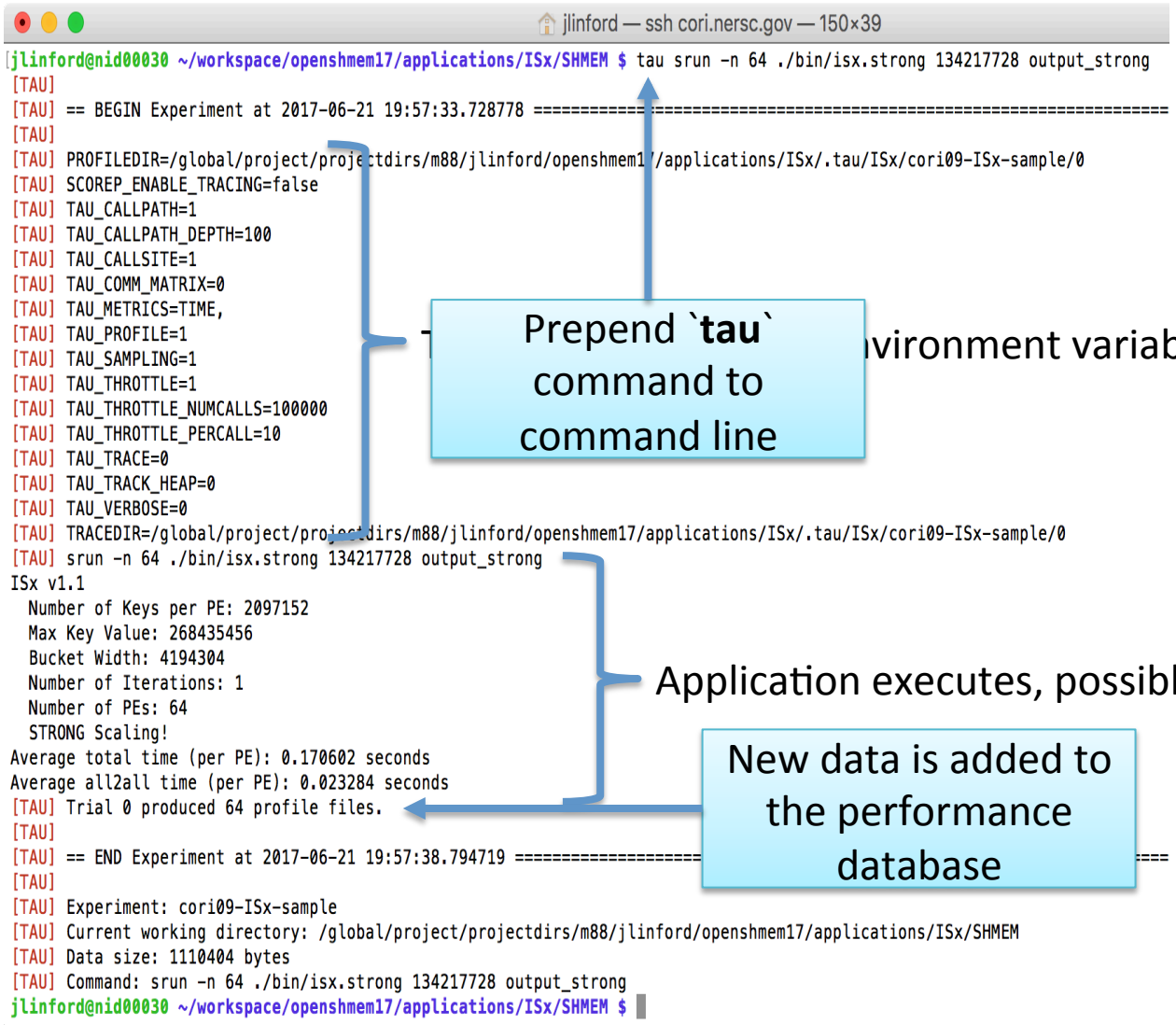
```
== Project Configuration (/global/project/projectdirs/m88/jlinford/openshmem17/applications/ISx/.tau/project.json) ==

+------+---------+--------------+---------------------+---------------+
| Name | Targets | Applications |     Measurements    | # Experiments |
+======+=========+==============+=====================+===============+
| ISx  | cori09  |      ISx     | sample, profile, trace |       1       |
+------+---------+--------------+---------------------+---------------+

== Targets in project 'ISx' ===================================================================================

+--------+---------+-----------+-----------------+-----------------+-----------------+
|  Name  | Host OS | Host Arch | Host Compilers | MPI Compilers | SHMEM Compilers |
+========+=========+===========+=================+=================+=================+
| cori09 |   CNL   |   x86_64  |       Cray      |       Cray      |       Cray      |
+--------+---------+-----------+-----------------+-----------------+-----------------+

== Applications in project 'ISx' ==============================================================================

+------+---------+--------+---------+-----+-----+------+--------+--------+-----+
| Name | Linkage | OpenMP | Pthreads | TBB | MPI | CUDA | OpenCL | SHMEM | MPC |
+======+=========+========+==========+=====+=====+======+========+========+=====+
| ISx  | static  |   No   |    No    | No  | No  |  No  |   No   |  Yes  | No  |
+------+---------+--------+----------+-----+-----+------+--------+--------+-----+

== Measurements in project 'ISx' ==============================================================================

+---------+---------+-------+--------+--------------+---------------+---------+------+-----+-----+--------+
|  Name   | Profile | Trace | Sample | Source Inst. | Compiler Inst. | OpenMP | CUDA | I/O | MPI | SHMEM |
+=========+=========+=======+========+==============+================+========+======+=====+=====+========+
| sample  |   tau   | none  |  Yes   |    never     |     never      | ignore |  No  | No  | No  |  Yes   |
+---------+---------+-------+--------+--------------+----------------+--------+------+-----+-----+--------+
| profile |   tau   | none  |   No   |  automatic   |     never      | ignore |  No  | No  | No  |  Yes   |
+---------+---------+-------+--------+--------------+----------------+--------+------+-----+-----+--------+
|  trace  |  none   | slog2 |   No   |  automatic   |     never      | ignore |  No  | No  | No  |  Yes   |
+---------+---------+-------+--------+--------------+----------------+--------+------+-----+-----+--------+

== Experiments in project 'ISx' ===============================================================================

+-----------------+--------+-----------+---------+-------------+-------------+-----------------+
|      Name       | Trials | Data Size |  Target | Application | Measurement |  TAU Makefile   |
+=================+========+===========+=========+=============+=============+=================+
| cori09-ISx-     |   0    |   0.0B    | cori09  |     ISx     |   sample    | Makefile.tau-   |
|      sample     |        |           |         |             |             |    intel-       |
|                 |        |           |         |             |             |    3f5a233a-    |
|                 |        |           |         |             |             |    shmem-pdt    |
+-----------------+--------+-----------+---------+-------------+-------------+-----------------+

Selected Experiment: cori09-ISx-sample
```

Step 2: Use `tau` to compile



- TAU Commander constructs a new compilation command line.
  - May replace compiler commands with TAU's compiler wrapper scripts.
  - May set environment variables, parse configuration files, etc.
  - If no changes are required then nothing is changed.

Step 3: Use `tau` to run



Step 4: Use `tau` to view data (`tau show`)

TAU: ParaProf: Mean Statistics - cori09-ISx-sample.trial0.ppk

Name	Exclusive TIME	Inclusive TIME ▽	Calls	Child Calls
.TAU application	0.306	1.347	1	326
void shmem_init(void) C	0.498	0.498	1	0
void shmem_finalize(void) C	0.462	0.462	1	0
[CONTEXT] .TAU application	0	0.32	6.406	0
[SUMMARY] main [{/global/project/projectdirs/m88/jlinford/openshmem17/applications/ISx/SHMEM/isx.c}]	0.262	0.262	5.172	0
[SAMPLE] main [{/global/project/projectdirs/m88/jlinford/openshmem17/applications/ISx/SHMEM/isx.c} {497}]	0.149	0.149	2.922	0
[SAMPLE] main [{/global/project/projectdirs/m88/jlinford/openshmem17/applications/ISx/SHMEM/isx.c} {374}]	0.041	0.041	0.812	0
[SAMPLE] main [{/global/project/projectdirs/m88/jlinford/openshmem17/applications/ISx/SHMEM/isx.c} {376}]	0.036	0.036	0.719	0
[SAMPLE] main [{/global/project/projectdirs/m88/jlinford/openshmem17/applications/ISx/SHMEM/isx.c} {378}]	0.011	0.011	0.219	0
[SAMPLE] main [{/global/project/projectdirs/m88/jlinford/openshmem17/applications/ISx/SHMEM/isx.c} {260}]	0.008	0.008	0.172	0
[SAMPLE] main [{/global/project/projectdirs/m88/jlinford/openshmem17/applications/ISx/SHMEM/isx.c} {381}]	0.005	0.005	0.094	0
[SAMPLE] main [{/global/project/projectdirs/m88/jlinford/openshmem17/applications/ISx/SHMEM/isx.c} {476}]	0.004	0.004	0.078	0
[SAMPLE] main [{/global/project/projectdirs/m88/jlinford/openshmem17/applications/ISx/SHMEM/isx.c} {379}]	0.004	0.004	0.078	0
[SAMPLE] main [{/global/project/projectdirs/m88/jlinford/openshmem17/applications/ISx/SHMEM/isx.c} {380}]	0.004	0.004	0.078	0
[SUMMARY] pcg32_boundedrand_r [{/global/project/projectdirs/m88/jlinford/openshmem17/applications/ISx/SHMEM/pcg_basic.c}]	0.057	0.057	1.203	0
[SAMPLE] __close_nocancel [{/home/abuild/rpmbuild/BUILD/glibc-2.19/nptl/../sysdeps/unix/syscall-template.S} {81}]	0.001	0.001	0.016	0
[SAMPLE] __wrap_shmem_n_pes [{/global/project/projectdirs/m88/jlinford/openshmem17/applications/ISx/SHMEM/bin/isx.strong} {0}]	0.001	0.001	0.016	0
void shmem_int_put(int *, const int *, size_t, int) C	0.037	0.037	126	0
long long shmem_longlong_fadd(long long *, long long, int) C	0.018	0.018	128	0
void *shmem_malloc(size_t) C	0.015	0.015	16	0
void shmem_barrier_all(void) C	0.009	0.009	27	0
void shmem_fcollect64(void *, const void *, size_t, int, int, int, long *) C	0.001	0.001	7	0
void shmem_collect32(void *, const void *, size_t, int, int, int, long *) C	0	0	1	0
void shmem_longlong_sum_to_all(long long *, const long long *, size_t, int, int, int, long long *, long *) C	0	0	1	0
int shmem_my_pe(void) C	0	0	9	0
void shmem_free(void *) C	0	0	8	0
int shmem_n_pes(void) C	0	0	1	0
[CALLSITE] void shmem_init(void) C	0.996	0.996	2	0
[CONTEXT] [CALLSITE] void shmem_init(void) C	0	0.481	1.688	0
[SAMPLE] __ioctl [{/home/abuild/rpmbuild/BUILD/glibc-2.19/misc/../sysdeps/unix/syscall-template.S} {81}]	0.473	0.473	1.344	0
[SAMPLE] _pmi_smp_barrier_join [{/usr/src/packages/BUILD/cray-pmi-5.0.10/src/pmi_core/smp_barrier.c} {70}]	0.006	0.006	0.281	0
[SAMPLE] Tau_lite_stop_timer [{/global/project/projectdirs/m88/jlinford/taucmdr-test/system/tau/tau-2.26.2/src/Profile/TauCAPI.cpp}	0.002	0.002	0.047	0
[SAMPLE] _dmappi_sheap_alloc [{/home/abuild/rpmbuild/BUILD/cray-dmapp-7.1.1/src/dmapp_sheap.c} {318}]	0.001	0.001	0.016	0

Create a New Experiment

Select a new measurement to create a new experiment

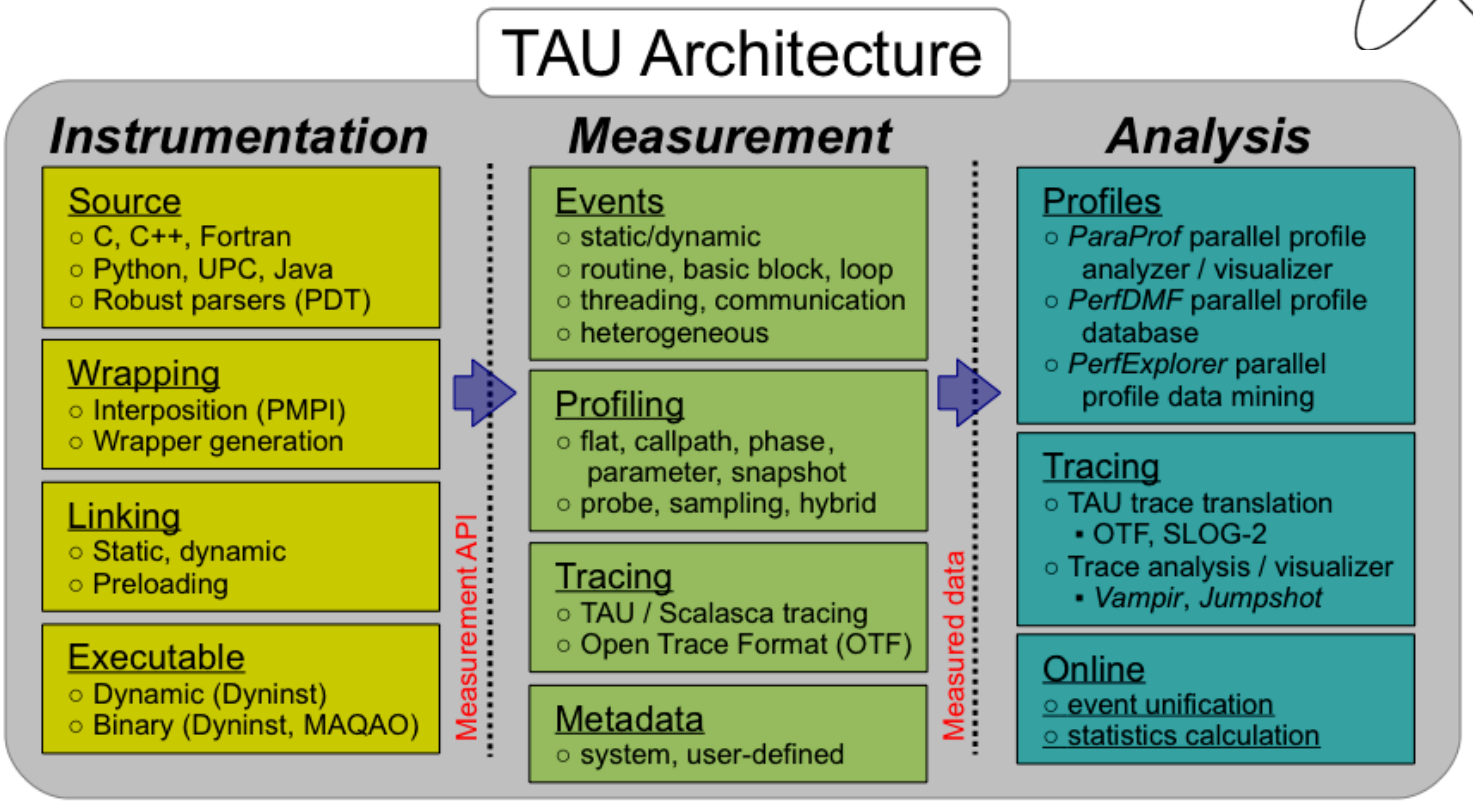
ParaTools — ssh cori.nersc.gov — 120×17

```
jlinford@nid00073 ~/workspace/openshmem17/applications/ISx/SHMEM $ tau select profile
[TAU] Created a new experiment 'cori09-ISx-profile'
[TAU] Installing TAU Performance System at
[TAU]     '/global/project/projectdirs/m88/jlinford/taucmdr-test/system/tau/./tau-2.26.2'
[TAU] Configuring TAU...
[TAU] Completed in 155.459 seconds
[TAU] Compiling and installing TAU...
[TAU] Completed in 48.596 seconds
[TAU] Checking installed files...
[TAU] Completed in 10.551 seconds
[TAU] Setting file permissions...
[TAU] Completed in 2.556 seconds
[TAU] Verifying TAU Performance System installation...
[TAU] Selected experiment 'cori09-ISx-profile'.
[TAU] Application rebuild required:
[TAU]   - source_inst changed from 'never' to 'automatic'
jlinford@nid00073 ~/workspace/openshmem17/applications/ISx/SHMEM $
```

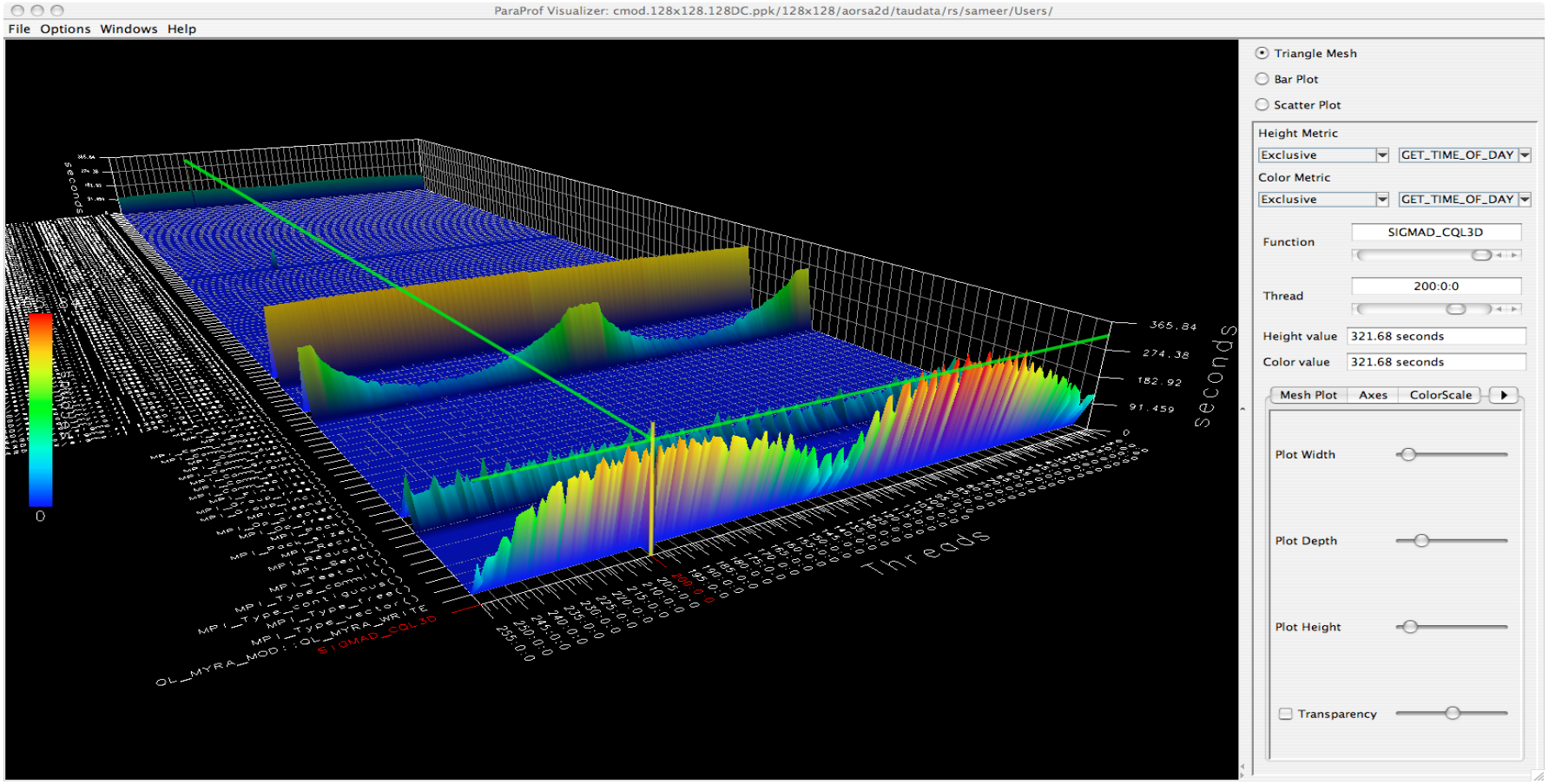
TAU Performance System® automatically reconfigured and recompiled.

User advised that an application rebuild is required to use source-based instrumentation.

TAU Architecture and Workflow



ParaProf 3D Profile Browser



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**

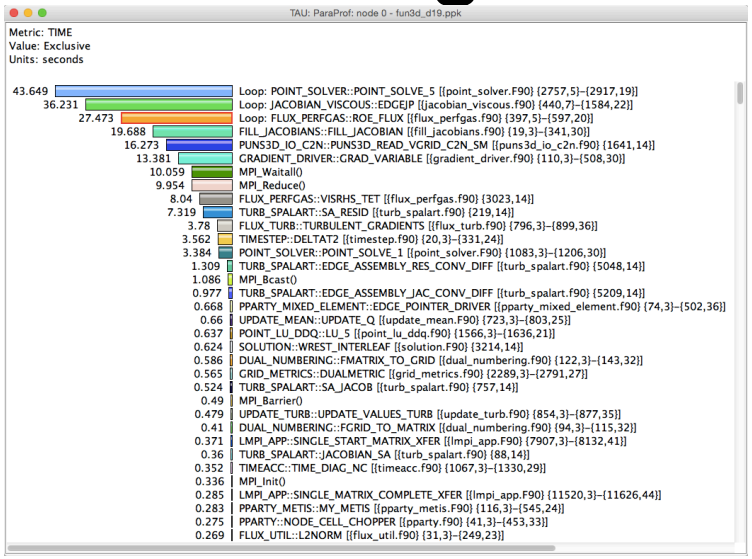
- Direct instrumentation (Interval events measure exclusive or inclusive duration)
- Indirect instrumentation (Sampling measures statement level contribution)
- Throttling and runtime control of low-level events that execute frequently
- Per-thread storage of performance data
- Interface with external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

Analysis: **Visualization of profiles and traces**

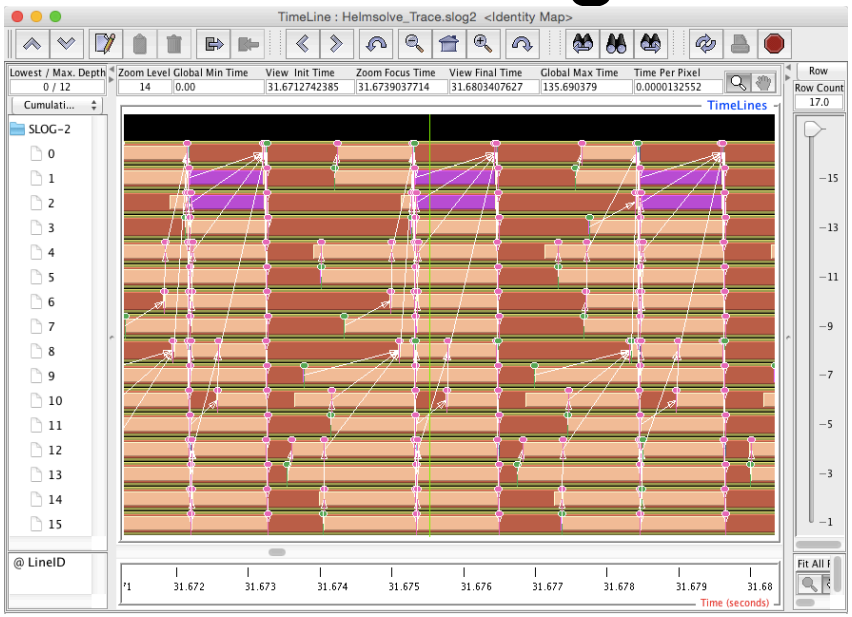
- 3D visualization of profile data in paraprof or perfexplorer tools
- Trace conversion & display in external visualizers (Vampir, Jumpshot, ParaVer)

Profiling and Tracing

Profiling



Tracing



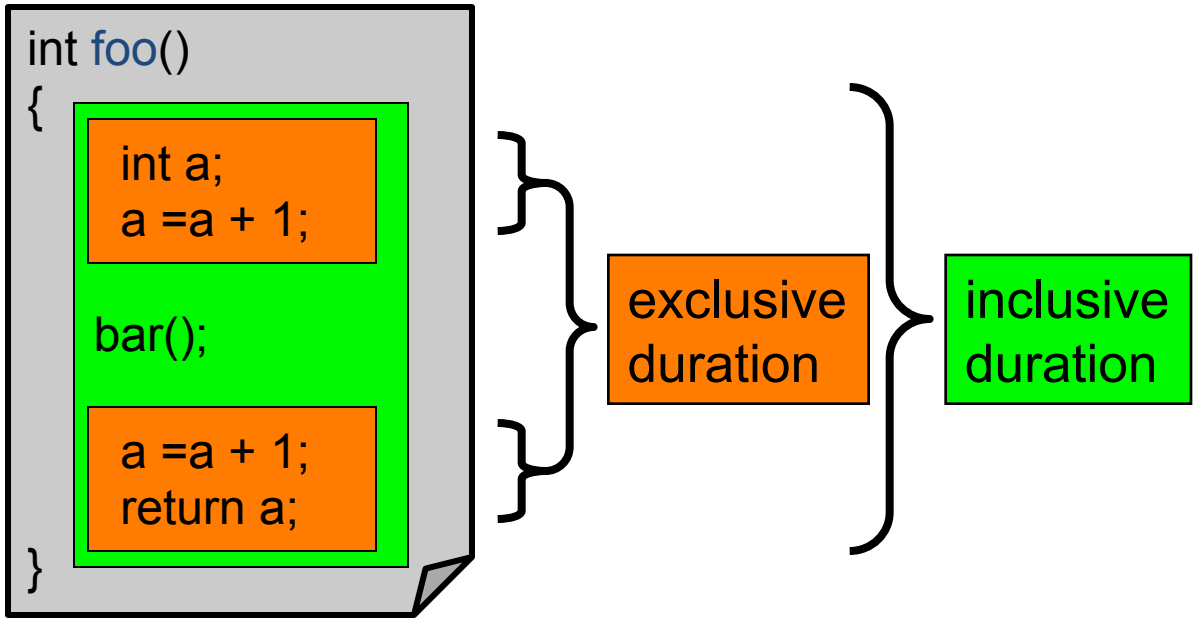
- Profiling and tracing

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

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

Inclusive vs. Exclusive Measurements

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



Instrumentation

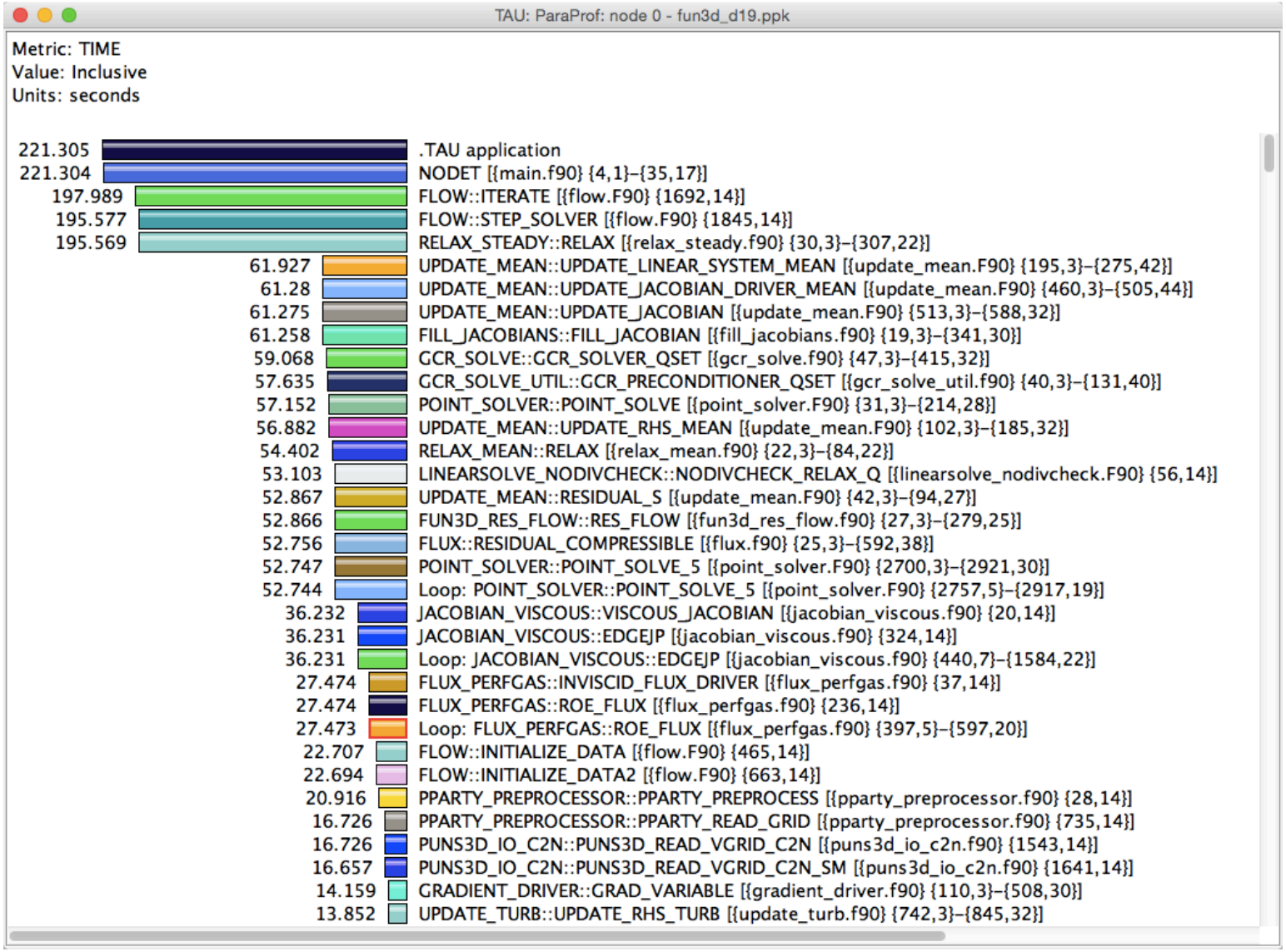
Direct and indirect performance observation

- 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.

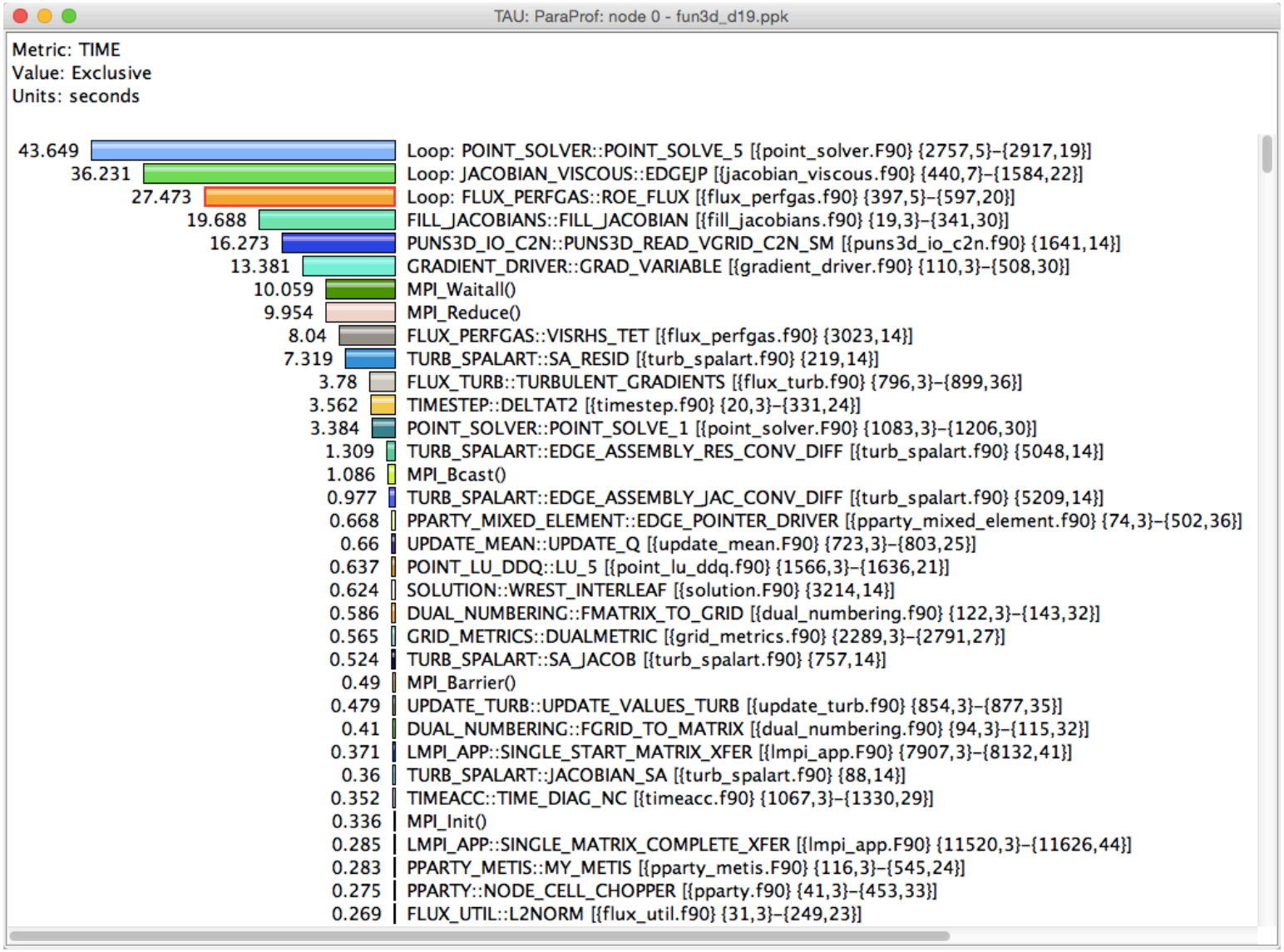
User-defined events

- ***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

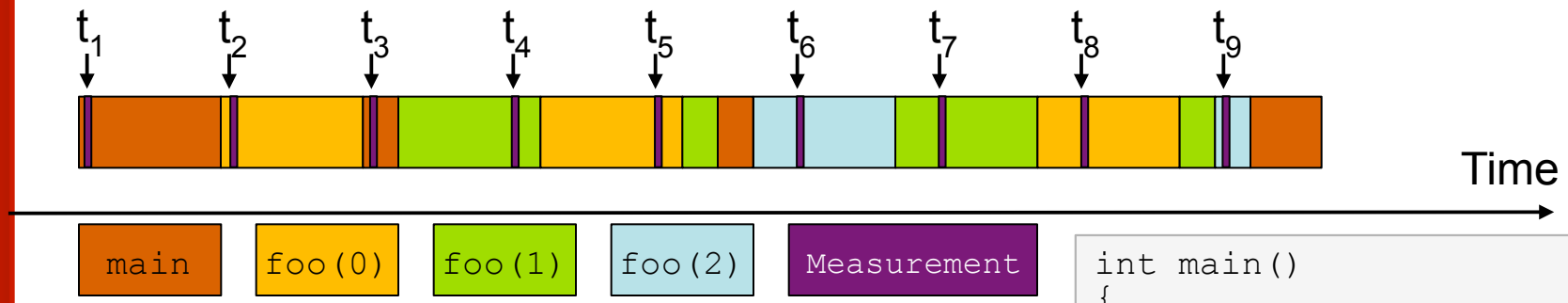
Inclusive Measurements



Exclusive Time



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

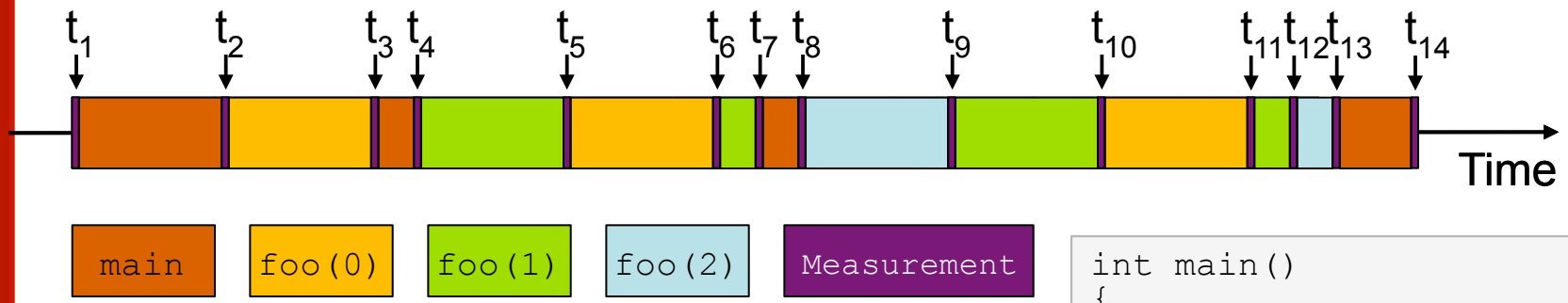
Statistical inference of program behavior

- 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");
}
```

Examples

Setting up Accounts, Examples at TACC

```
On Stampede2.tacc.utexas.edu
% tar zxf ~tg457572/workshop.tgz
% idev –m 50
% source ~tg457572/tau.impi.bashrc
% which tau
% cd workshop/mm
% tau init --mpi --compilers Intel --mpi-compilers Intel
% make clean;
% make
% tau mpirun –np 64 ./matmult
% tau show
And try the examples. Try:
% tau --help
% tau meas edit --help
```

Simplifying the use of TAU!

Uninstrumented code:

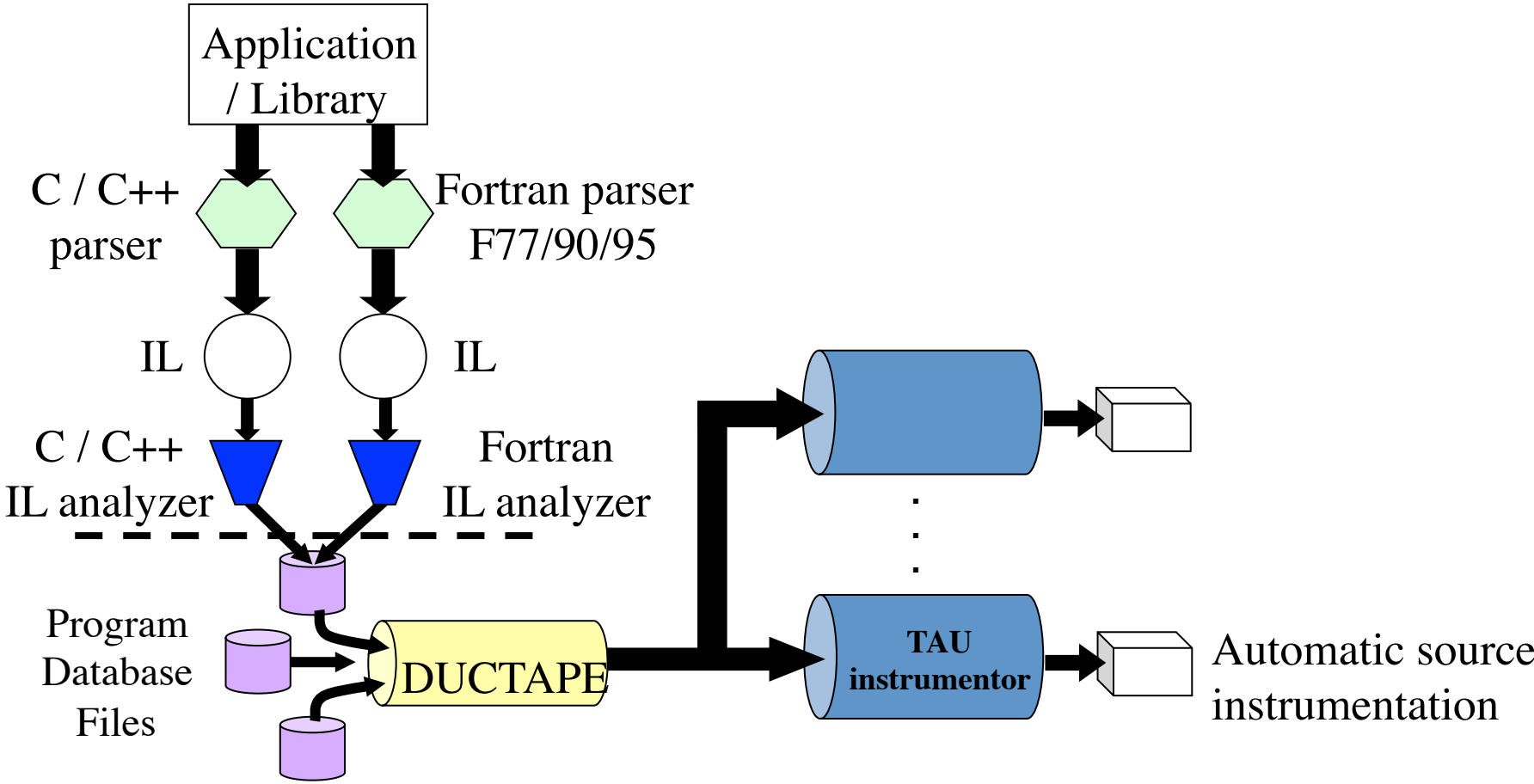
- % make
- % mpirun –np 256 ./a.out

With TAU using event based sampling (EBS):

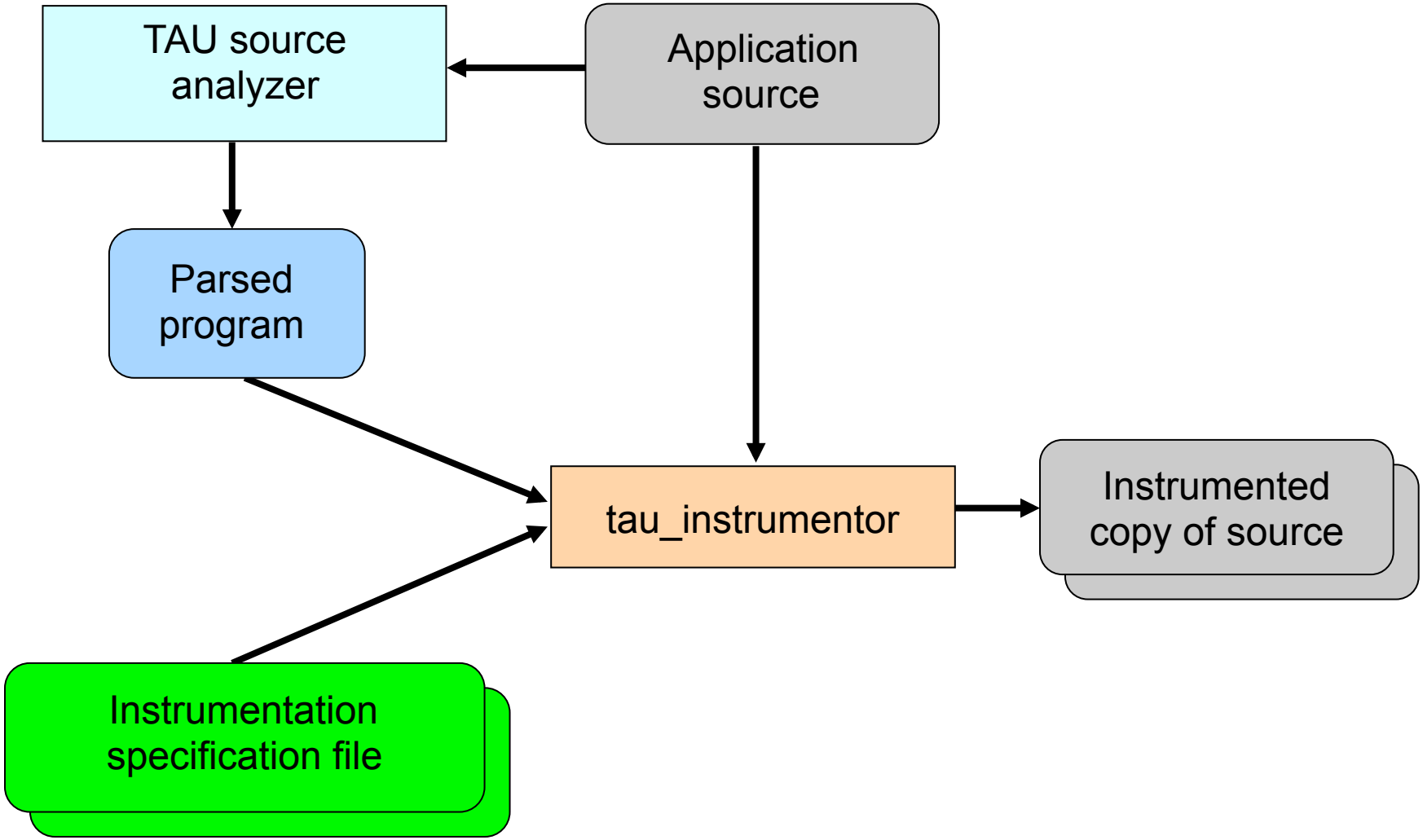
- % tau init --mpi --mpi-compilers Intel --compilers Intel
- % tau mpirun –np 256 ./a.out
- % tau show

Automatic Source Instrumentation

TAU's Static Analysis System: Program Database Toolkit (PDT)



Automatic Source Instrumentation using PDT



Automatic Source Instrumentation

- **Use TAU's compiler wrappers**
- **% tau select profile**
  - Simply replace `CXX` with `tau <compiler>`
  - Automatically instruments source code, links with TAU libraries.

Before

```
CXX = mpiicpc
F90 = mpiifort
CXXFLAGS =
LIBS = -lm
OBJS = f1.o f2.o f3.o … fn.o

app: $(OBJS)
     $(CXX) $(LDFLAGS) $(OBJS) -o $@
     $(LIBS)
.cpp.o:
     $(CXX) $(CXXFLAGS) -c $<
```

After

```
CXX = tau mpiicpc
F90 = tau mpiifort
CXXFLAGS =
LIBS = -lm
OBJS = f1.o f2.o f3.o … fn.o

app: $(OBJS)
     $(CXX) $(LDFLAGS) $(OBJS) -o $@
     $(LIBS)
.cpp.o:
     $(CXX) $(CXXFLAGS) -c $<
```

TAU and PDT for Source Instrumentation

```
On Stampede2.tacc.utexas.edu
% tar zxf ~tg457572/workshop.tgz
% idev -m 50
% source ~tg457572/tau.impi.bashrc
% which tau
% cd workshop/mm
% tau init --mpi --compilers Intel --mpi-compilers Intel
% tau dash
% tau select profile
% make clean
% make CC='tau mpiicc'
% tau mpirun -np 64 ./matmult
% tau show
```

Selective Instrumentation File

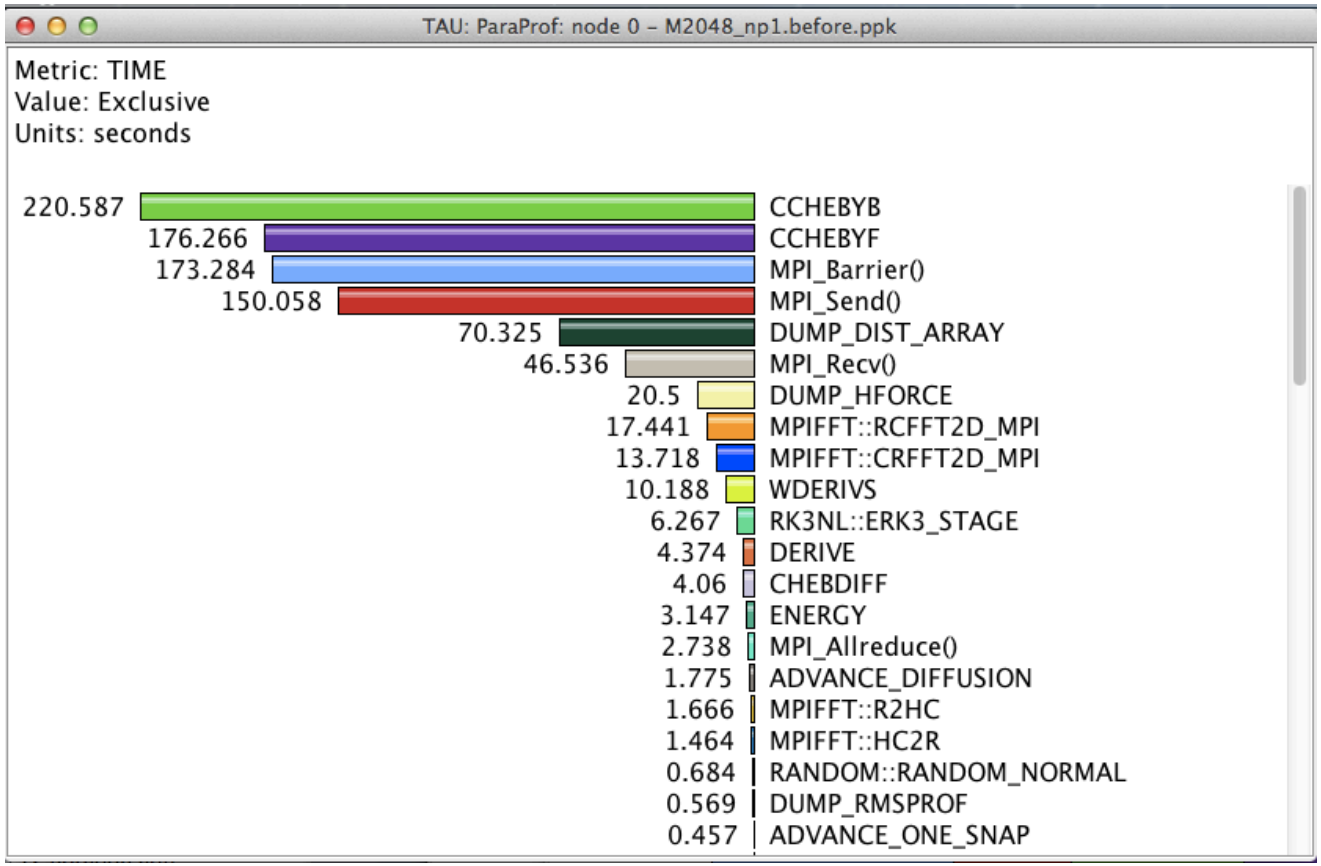
```
% tau dash
% tau application edit <app_name> --select-file select.tau
% cat select.tau
BEGIN_INCLUDE_LIST
int main#
int dgemm#
END_INCLUDE_LIST
BEGIN_FILE_INCLUDE_LIST
Main.c
Blas/*.f77
END_FILE_INCLUDE_LIST
# replace include with exclude list (BEGIN_EXCLUDE_LIST/END…)

BEGIN_INSTRUMENT_SECTION
loops routine="foo"
loops routine="int main#"
END_INSTRUMENT_SECTION
% export TAU_SELECT_FILE=select.tau    (to use at runtime)
```

TAU and PDT for Source Instrumentation

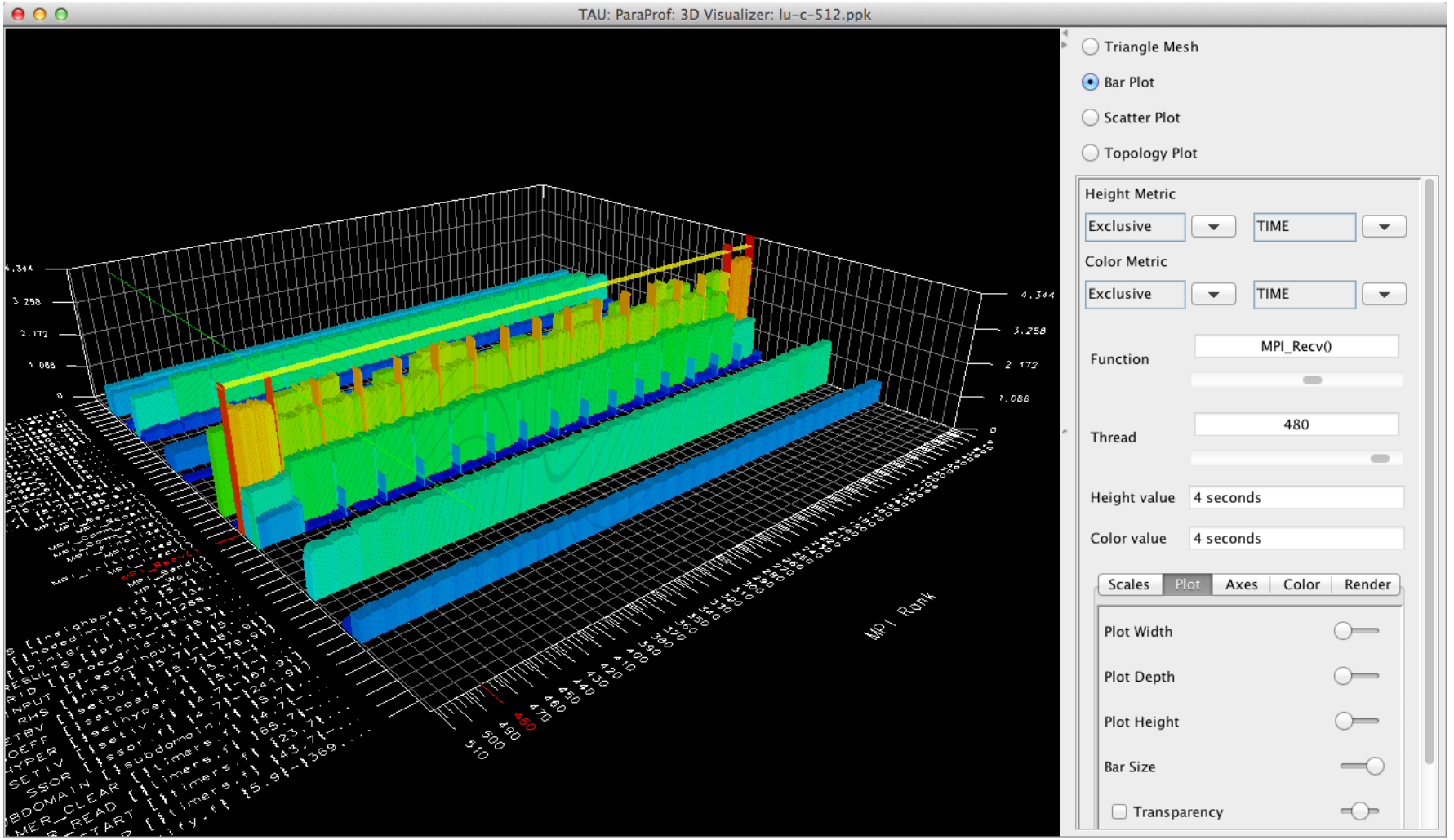
```
On Stampede2.tacc.utexas.edu
% tar zxf ~tg457572/workshop.tgz
% idev -m 50
% source ~tg457572/tau.impi.bashrc
% which tau
% cd workshop/mm
% tau init --mpi --compilers Intel --mpi-compilers Intel
% tau select profile
% cat select.tau
% tau application edit mm --select-file select.tau
% make CC='tau mpiicc'
% tau mpirun -np 64 ./matmult
% tau show
```

Routine Level Profile

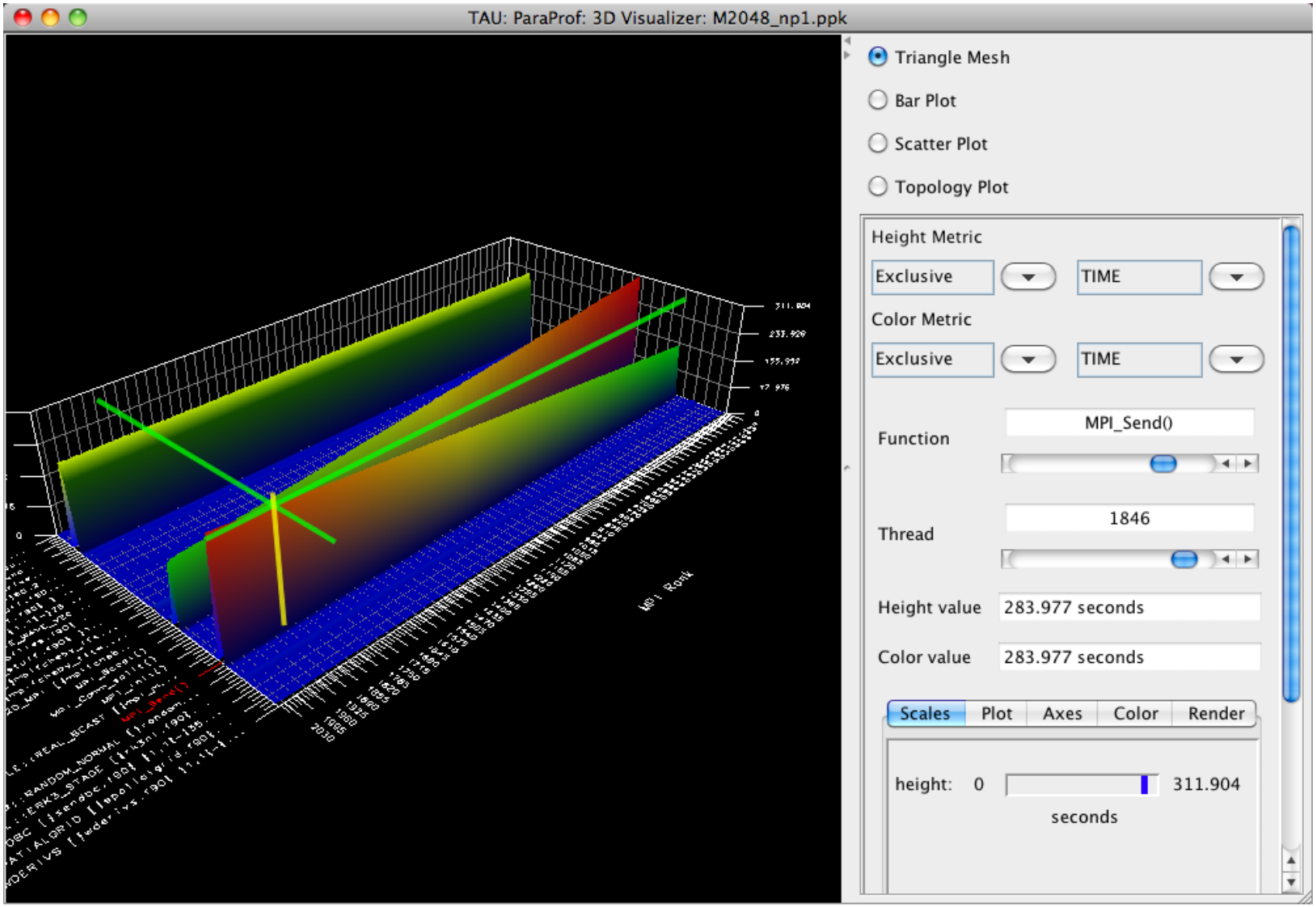
How much time is spent in each application routine?



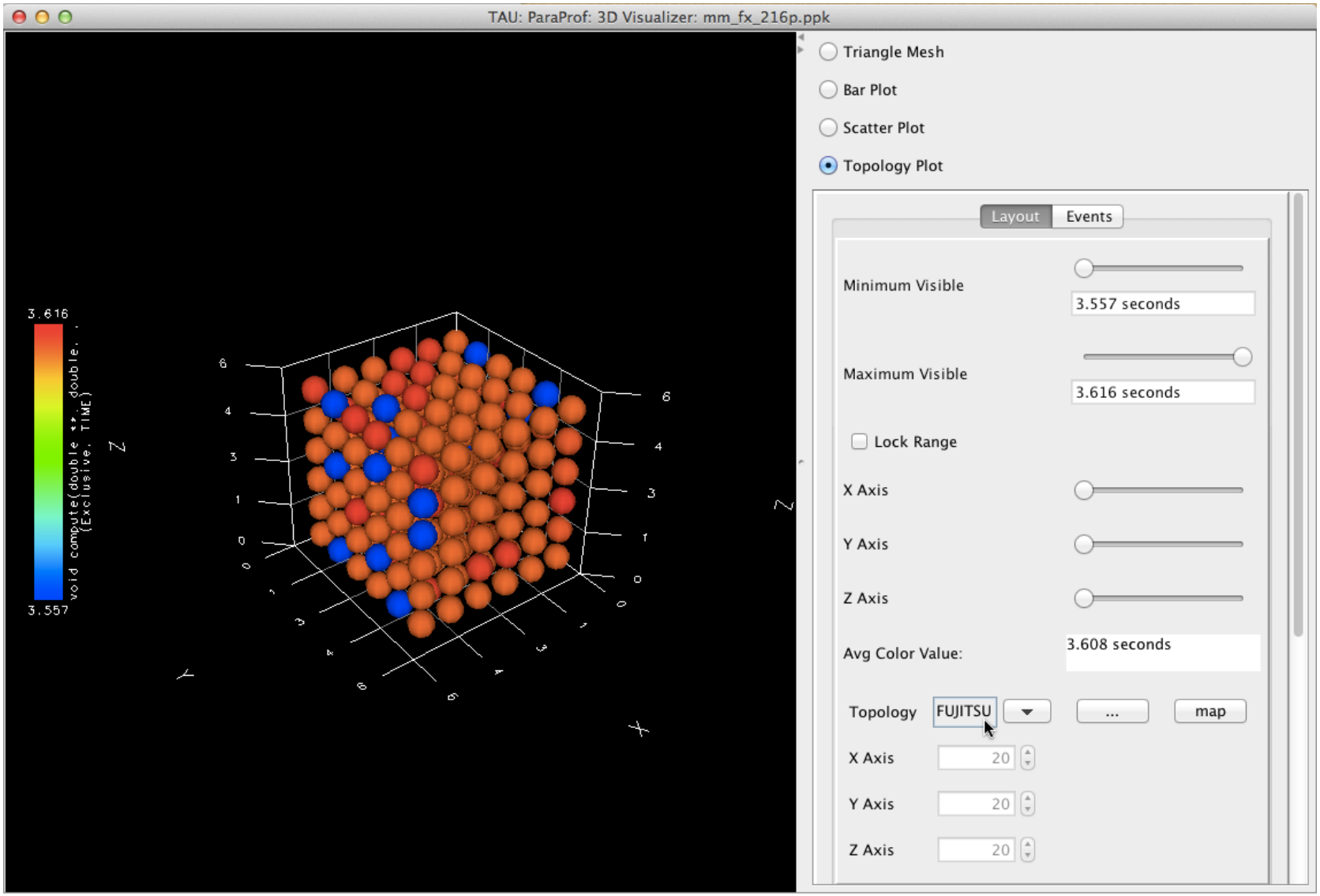
ParaProf 3D Profile Browser



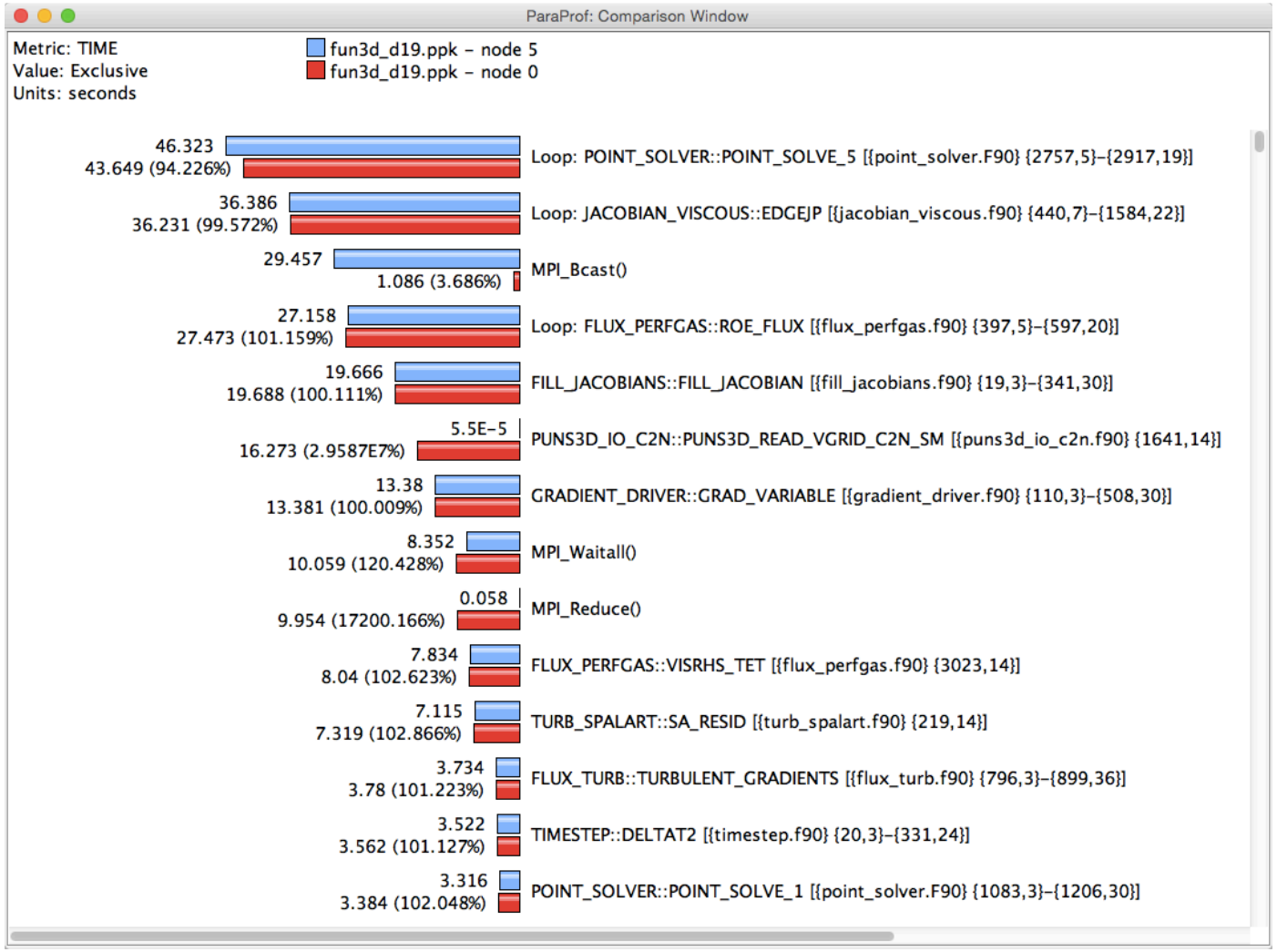
ParaProf


TAU: ParaProf: 3D Visualizer: M2048_np1.ppk
Triangle Mesh
Bar Plot
Scatter Plot
Topology Plot
Height Metric
Exclusive
TIME
Color Metric
Exclusive
TIME
Function
MPI_Send()
Thread
1846
Height value
283.977 seconds
Color value
283.977 seconds
Scales
Plot
Axes
Color
Render
height: 0
311.904
seconds
MPI Rank

ParaProf 3D Topology Display



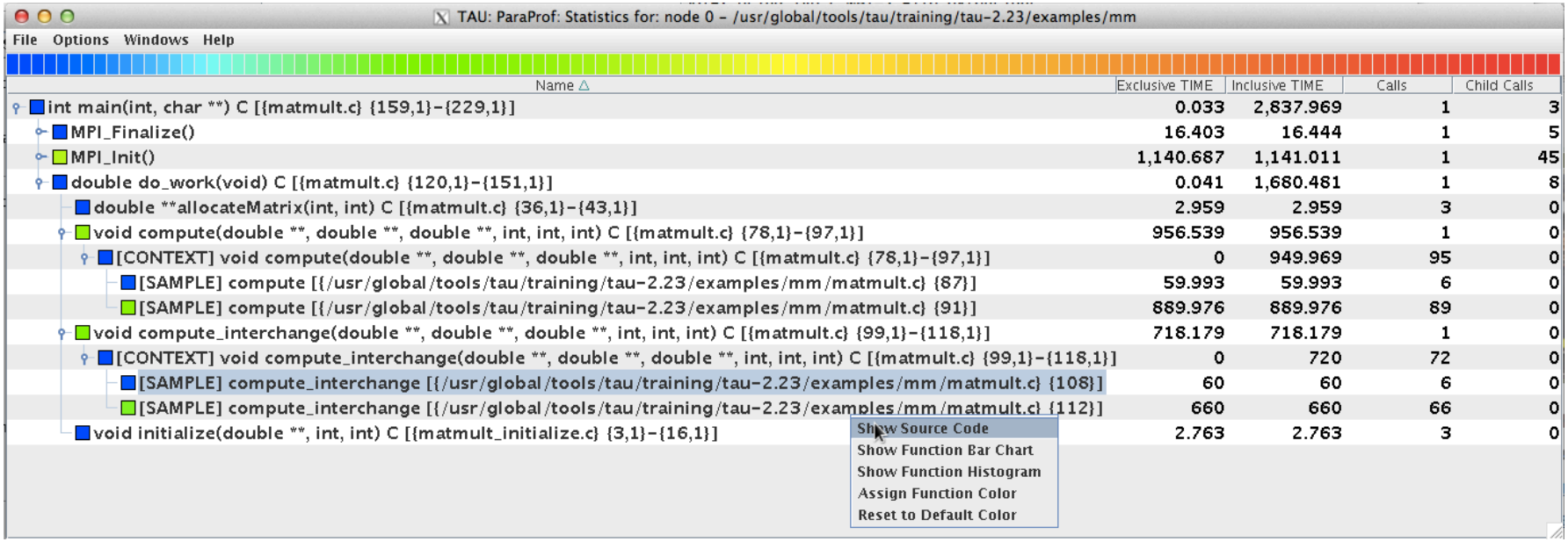
ParaProf Comparison Window



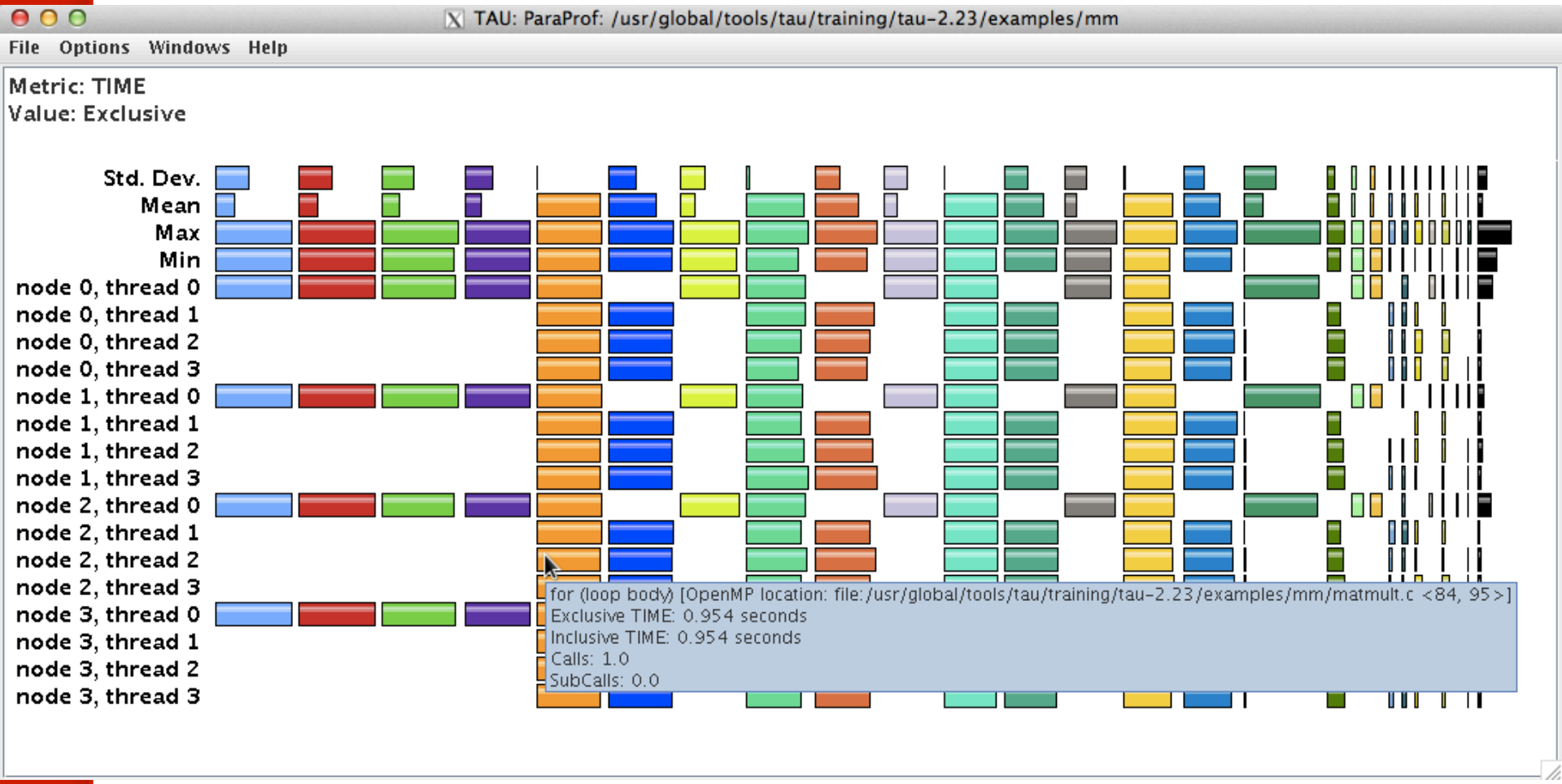
Comparing Rank 0 with 5.

Right click on “node 5” -> Add node to comparison window

Event Based Sampling in TAU

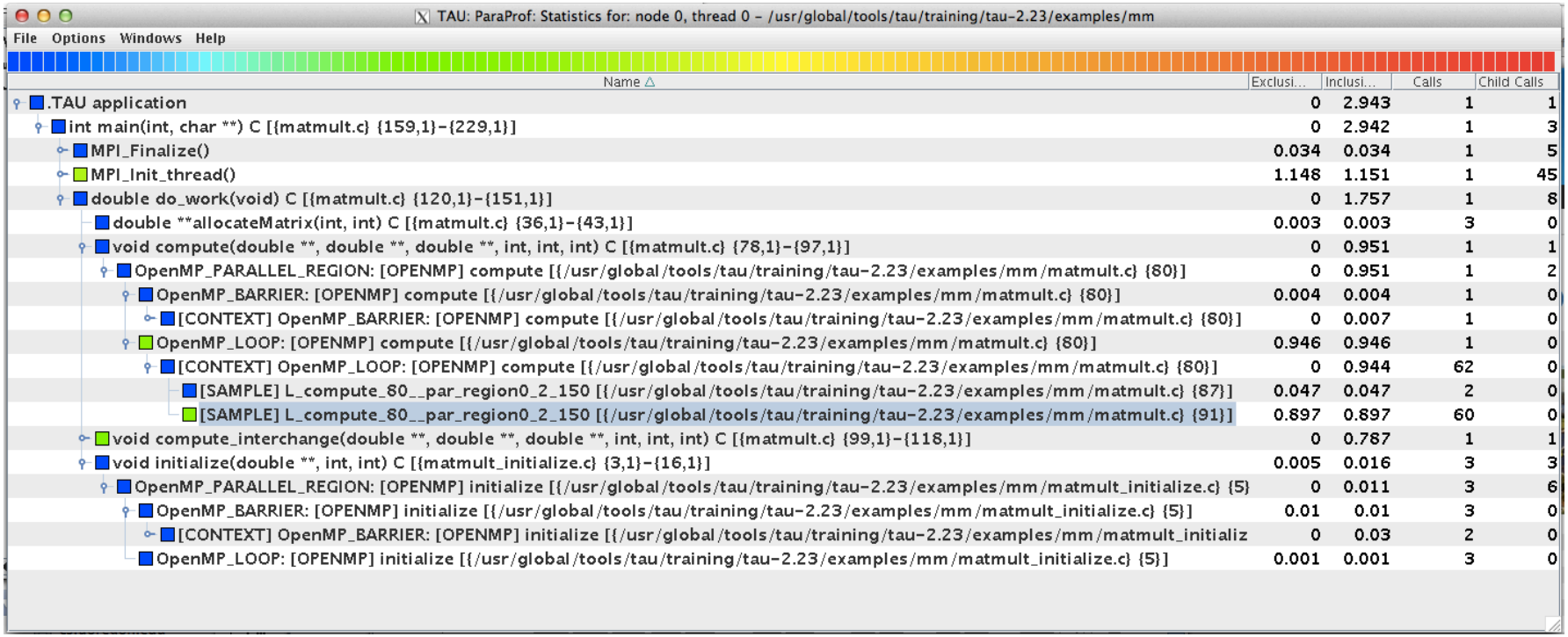


Mixed MPI and OpenMP Instrumentation



Options -> Uncheck “Stack Bars Together”

TAU's support for Intel and GNU OMPT



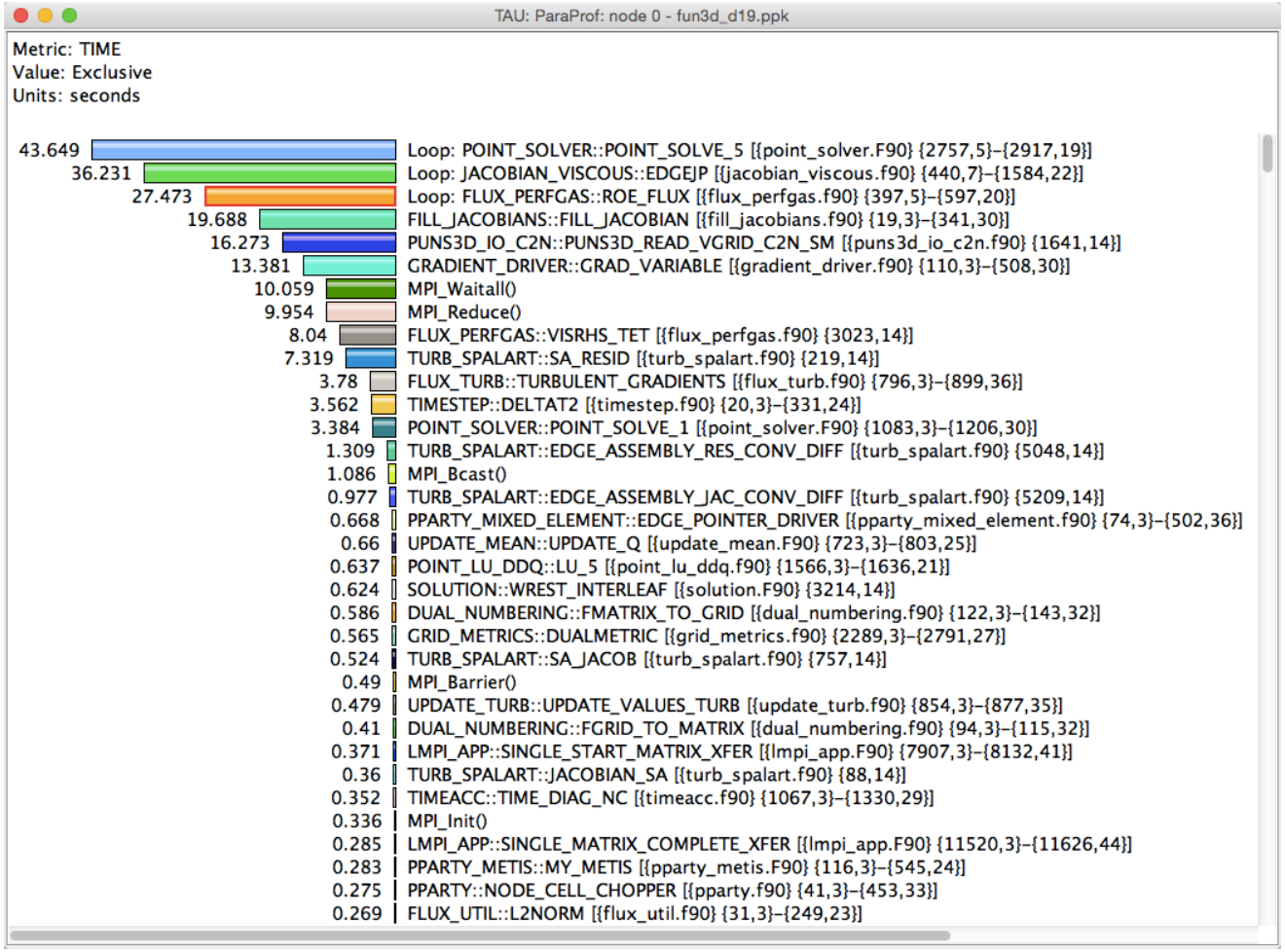
Name	Exclusi...	Inclusi...	Calls	Child Calls
.TAU application	0	2.943	1	1
int main(int, char **) C [{matmult.c} {159,1}-{229,1}]	0	2.942	1	3
MPI_Finalize()	0.034	0.034	1	5
MPI_Init_thread()	1.148	1.151	1	45
double do_work(void) C [{matmult.c} {120,1}-{151,1}]	0	1.757	1	8
double **allocateMatrix(int, int) C [{matmult.c} {36,1}-{43,1}]	0.003	0.003	3	0
void compute(double **, double **, double **, int, int, int) C [{matmult.c} {78,1}-{97,1}]	0	0.951	1	1
OpenMP_PARALLEL_REGION: [OPENMP] compute [{/usr/global/tools/tau/training/tau-2.23/examples/mm/matmult.c} {80}]	0	0.951	1	2
OpenMP_BARRIER: [OPENMP] compute [{/usr/global/tools/tau/training/tau-2.23/examples/mm/matmult.c} {80}]	0.004	0.004	1	0
[CONTEXT] OpenMP_BARRIER: [OPENMP] compute [{/usr/global/tools/tau/training/tau-2.23/examples/mm/matmult.c} {80}]	0	0.007	1	0
OpenMP_LOOP: [OPENMP] compute [{/usr/global/tools/tau/training/tau-2.23/examples/mm/matmult.c} {80}]	0.946	0.946	1	0
[CONTEXT] OpenMP_LOOP: [OPENMP] compute [{/usr/global/tools/tau/training/tau-2.23/examples/mm/matmult.c} {80}]	0	0.944	62	0
[SAMPLE] L_compute_80__par_region0_2_150 [{/usr/global/tools/tau/training/tau-2.23/examples/mm/matmult.c} {87}]	0.047	0.047	2	0
[SAMPLE] L_compute_80__par_region0_2_150 [{/usr/global/tools/tau/training/tau-2.23/examples/mm/matmult.c} {91}]	0.897	0.897	60	0
void compute_interchange(double **, double **, double **, int, int, int) C [{matmult.c} {99,1}-{118,1}]	0	0.787	1	1
void initialize(double **, int, int) C [{matmult_initialize.c} {3,1}-{16,1}]	0.005	0.016	3	3
OpenMP_PARALLEL_REGION: [OPENMP] initialize [{/usr/global/tools/tau/training/tau-2.23/examples/mm/matmult_initialize.c} {5}	0	0.011	3	6
OpenMP_BARRIER: [OPENMP] initialize [{/usr/global/tools/tau/training/tau-2.23/examples/mm/matmult_initialize.c} {5}]	0.01	0.01	3	0
[CONTEXT] OpenMP_BARRIER: [OPENMP] initialize [{/usr/global/tools/tau/training/tau-2.23/examples/mm/matmult_initializ	0	0.03	2	0
OpenMP_LOOP: [OPENMP] initialize [{/usr/global/tools/tau/training/tau-2.23/examples/mm/matmult_initialize.c} {5}]	0.001	0.001	3	0

```
% tau meas edit profile −openmp ompt
% tau select profile
% make clean
% make CC='tau mpiicc'
% tau mpirun −np 16 ./matmult
% tau show
```

Loop Level Instrumentation

Goal: What loops account for the most time? How much?

Flat profile with wallclock time with loop instrumentation:

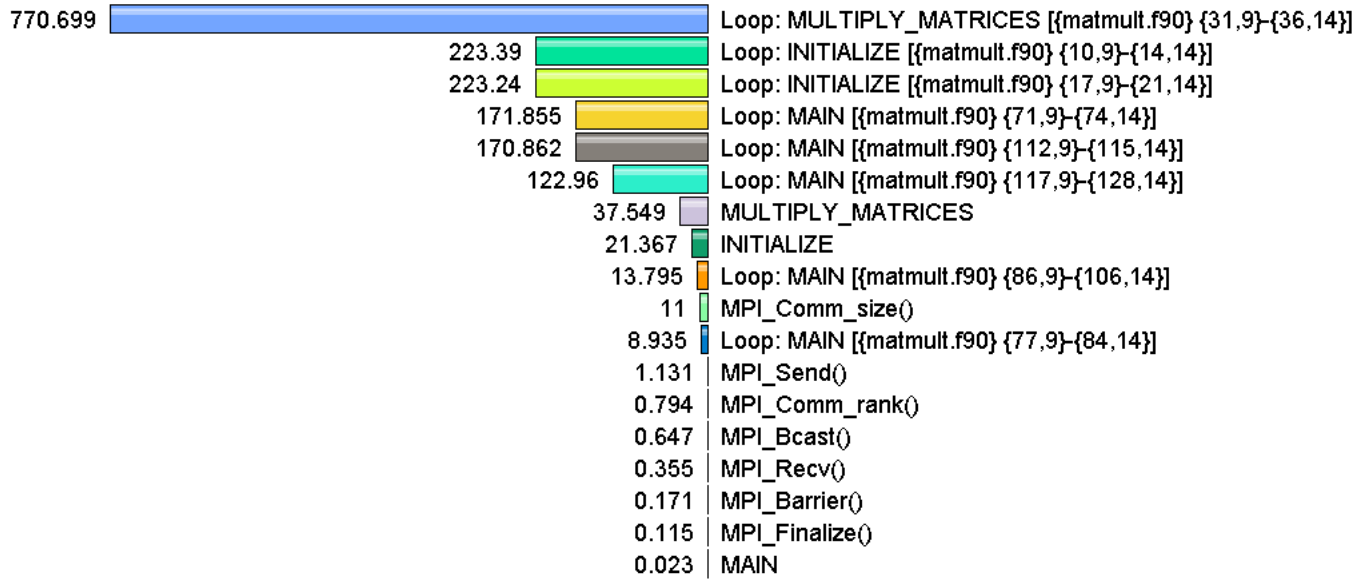


Computing FLOPS per loop

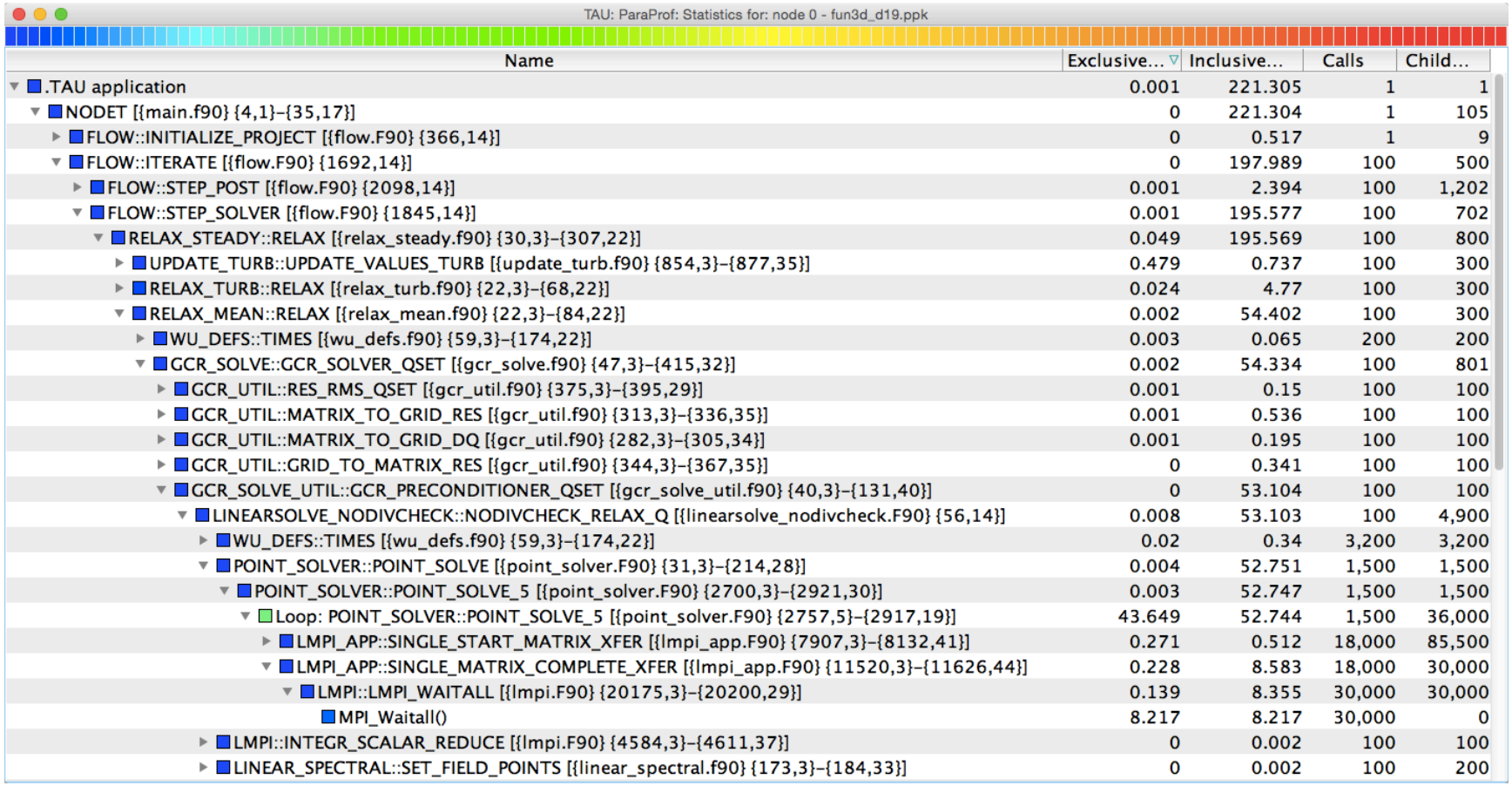
Goal: What is the execution rate of my loops in MFLOPS?

Flat profile with PAPI_FP_INS and time with loop instrumentation:

Metric: PAPI_FP_INS / GET_TIME_OF_DAY
Value: Exclusive
Units: Derived metric shown in microseconds format

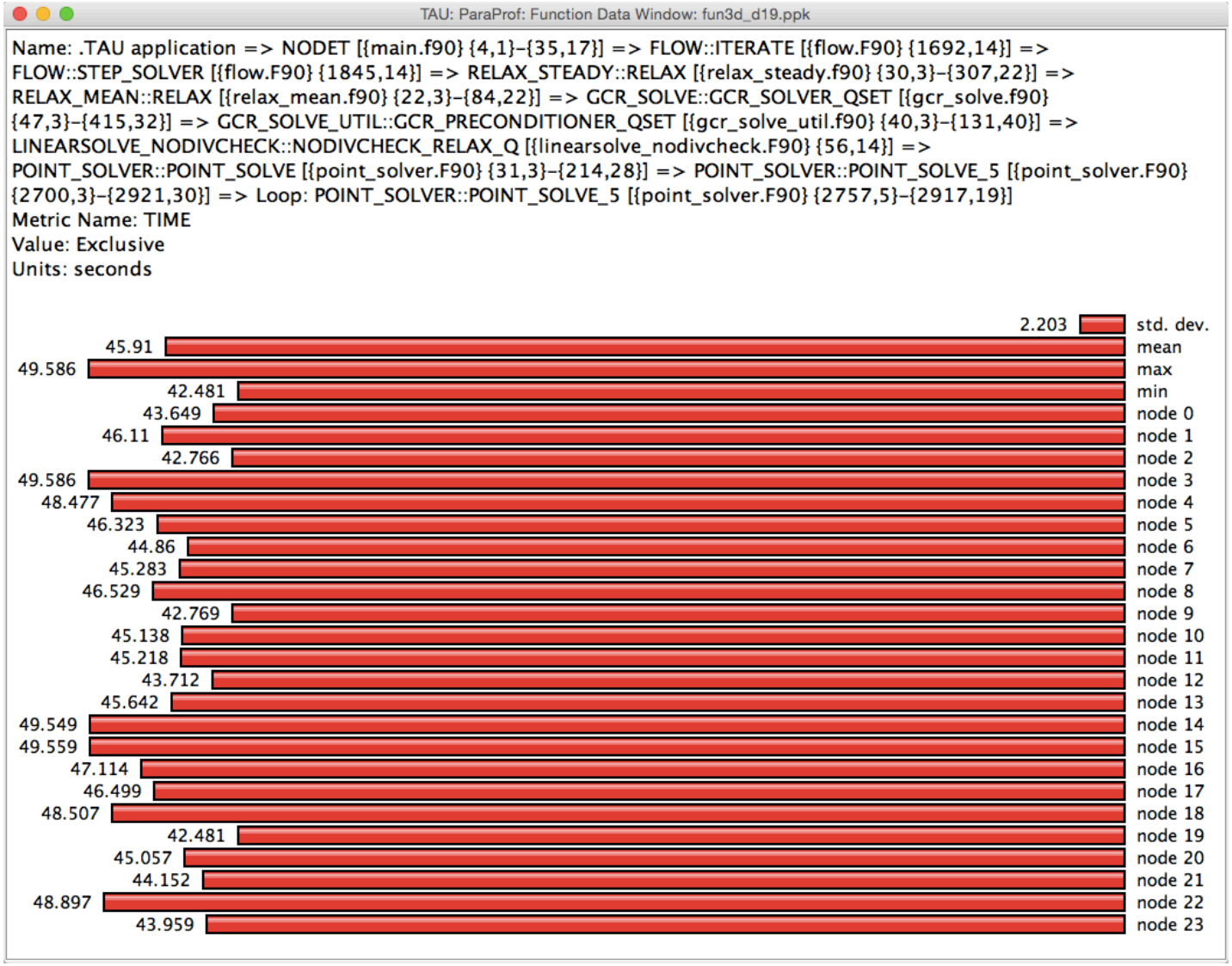


Callpath Profiling: FUN3D



Name	Exclusive...	Inclusive...	Calls	Child...
.TAU application	0.001	221.305	1	1
NODET [{main.f90} {4,1}-{35,17}]	0	221.304	1	105
FLOW::INITIALIZE_PROJECT [{flow.F90} {366,14}]	0	0.517	1	9
FLOW::ITERATE [{flow.F90} {1692,14}]	0	197.989	100	500
FLOW::STEP_POST [{flow.F90} {2098,14}]	0.001	2.394	100	1,202
FLOW::STEP_SOLVER [{flow.F90} {1845,14}]	0.001	195.577	100	702
RELAX_STEADY::RELAX [{relax_steady.f90} {30,3}-{307,22}]	0.049	195.569	100	800
UPDATE_TURB::UPDATE_VALUES_TURB [{update_turb.f90} {854,3}-{877,35}]	0.479	0.737	100	300
RELAX_TURB::RELAX [{relax_turb.f90} {22,3}-{68,22}]	0.024	4.77	100	300
RELAX_MEAN::RELAX [{relax_mean.f90} {22,3}-{84,22}]	0.002	54.402	100	300
WU_DEFS::TIMES [{wu_defs.f90} {59,3}-{174,22}]	0.003	0.065	200	200
GCR_SOLVE::GCR_SOLVER_QSET [{gcr_solve.f90} {47,3}-{415,32}]	0.002	54.334	100	801
GCR_UTIL::RES_RMS_QSET [{gcr_util.f90} {375,3}-{395,29}]	0.001	0.15	100	100
GCR_UTIL::MATRIX_TO_GRID_RES [{gcr_util.f90} {313,3}-{336,35}]	0.001	0.536	100	100
GCR_UTIL::MATRIX_TO_GRID_DQ [{gcr_util.f90} {282,3}-{305,34}]	0.001	0.195	100	100
GCR_UTIL::GRID_TO_MATRIX_RES [{gcr_util.f90} {344,3}-{367,35}]	0	0.341	100	100
GCR_SOLVE_UTIL::GCR_PRECONDITIONER_QSET [{gcr_solve_util.f90} {40,3}-{131,40}]	0	53.104	100	100
LINEARSOLVE_NODIVCHECK::NODIVCHECK_RELAX_Q [{linearsolve_nodivcheck.F90} {56,14}]	0.008	53.103	100	4,900
WU_DEFS::TIMES [{wu_defs.f90} {59,3}-{174,22}]	0.02	0.34	3,200	3,200
POINT_SOLVER::POINT_SOLVE [{point_solver.F90} {31,3}-{214,28}]	0.004	52.751	1,500	1,500
POINT_SOLVER::POINT_SOLVE_5 [{point_solver.F90} {2700,3}-{2921,30}]	0.003	52.747	1,500	1,500
Loop: POINT_SOLVER::POINT_SOLVE_5 [{point_solver.F90} {2757,5}-{2917,19}]	43.649	52.744	1,500	36,000
LMPI_APP::SINGLE_START_MATRIX_XFER [{lmpi_app.F90} {7907,3}-{8132,41}]	0.271	0.512	18,000	85,500
LMPI_APP::SINGLE_MATRIX_COMPLETE_XFER [{lmpi_app.F90} {11520,3}-{11626,44}]	0.228	8.583	18,000	30,000
LMPI::LMPI_WAITALL [{lmpi.F90} {20175,3}-{20200,29}]	0.139	8.355	30,000	30,000
MPI_Waitall()	8.217	8.217	30,000	0
LMPI::INTEGR_SCALAR_REDUCE [{lmpi.F90} {4584,3}-{4611,37}]	0	0.002	100	100
LINEAR_SPECTRAL::SET_FIELD_POINTS [{linear_spectral.f90} {173,3}-{184,33}]	0	0.002	100	200

ParaProf Function Window



ParaProf Callpath Thread Relations Window

TAU: ParaProf: Call Path Data n,c,t, 5,0,0 - fun3d_d19.ppk

Metric Name: TIME
Sorted By: Exclusive
Units: seconds

	Exclusive	Inclusive	Calls/Tot.Calls	Name[id]
	46.323	53.014	1500/1500	POINT_SOLVER::POINT_SOLVE_5 [{point_solver.F90} {2700,3}-{2921,30}]
-->	46.323	53.014	1500	Loop: POINT_SOLVER::POINT_SOLVE_5 [{point_solver.F90} {2757,5}-{2917,19}]
	0.16	6.379	18000/30100	LMPI_APP::SINGLE_MATRIX_COMPLETE_XFER [{lmpi_app.F90} {11520,3}-{11626,44}]
	0.166	0.311	18000/30100	LMPI_APP::SINGLE_START_MATRIX_XFER [{lmpi_app.F90} {7907,3}-{8132,41}]
	36.386	36.386	78/78	JACOBIAN_VISCOUS::EDGEJP [{jacobian_viscous.f90} {324,14}]
-->	36.386	36.386	78	Loop: JACOBIAN_VISCOUS::EDGEJP [{jacobian_viscous.f90} {440,7}-{1584,22}]
	4.7E-4	4.7E-4	45/27956	PPARTY_METIS::MY_METIS [{pparty_metis.F90} {116,3}-{545,24}]
	0.242	0.242	777/27956	LMPI::LOGICL_SCALAR_BCAST [{lmpi.F90} {3691,3}-{3727,36}]
	0.002	0.002	2/27956	LMPI::DOUBLE_TENSOR_BCAST [{lmpi.F90} {4185,3}-{4214,36}]
	16.689	16.689	31/27956	LMPI::INTEGR_MATRIX_BCAST [{lmpi.F90} {3240,3}-{3276,36}]
	0.013	0.013	48/27956	LMPI::INTEG8_VECTOR_BCAST [{lmpi.F90} {3376,3}-{3412,36}]
	0.587	0.587	489/27956	LMPI::INTEGR_VECTOR_BCAST [{lmpi.F90} {3196,3}-{3232,36}]
	0.003	0.003	1197/27956	LMPI::CHARACTER_BCAST [{lmpi.F90} {3100,3}-{3136,32}]
	1.521	1.521	5371/27956	LMPI::DOUBLE_VECTOR_BCAST [{lmpi.F90} {4096,3}-{4132,36}]
	1.2E-5	1.2E-5	2/27956	LMPI::LOGICL_VECTOR_BCAST [{lmpi.F90} {3736,3}-{3772,36}]
	0.012	0.012	1040/27956	LMPI::DOUBLE_SCALAR_BCAST [{lmpi.F90} {4051,3}-{4087,36}]
	0.055	0.055	3/27956	LMPI::INTEG8_SCALAR_BCAST [{lmpi.F90} {3331,3}-{3367,36}]
	10.328	10.328	18941/27956	LMPI::INTEGR_SCALAR_BCAST [{lmpi.F90} {3151,3}-{3187,36}]
	0.005	0.005	10/27956	LMPI::DOUBLE_MATRIX_BCAST [{lmpi.F90} {4140,3}-{4176,36}]
-->	29.457	29.457	27956	MPI_Bcast()
	27.158	27.158	100/100	FLUX_PERFGAS::ROE_FLUX [{flux_perfgas.f90} {236,14}]
-->	27.158	27.158	100	Loop: FLUX_PERFGAS::ROE_FLUX [{flux_perfgas.f90} {397,5}-{597,20}]

Shows the contribution of parents and children for each routine (marked by an arrow)

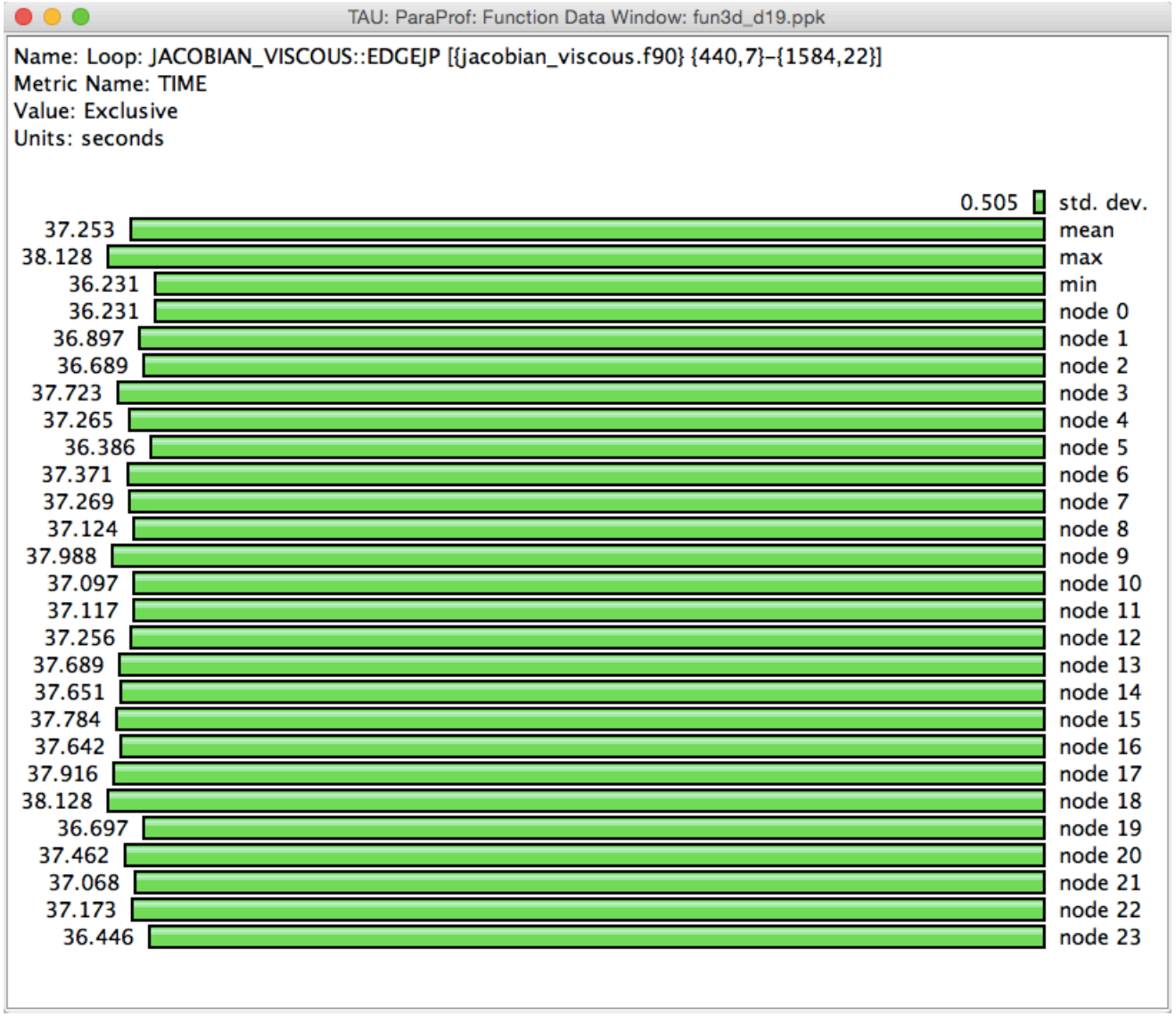
ParaProf Callpath Thread Relations Window

TAU: ParaProf: Call Path Data n,c,t, 13,0,0 - fun3d_d19.ppk

Metric Name: TIME
Sorted By: Exclusive
Units: seconds

	Exclusive	Inclusive	Calls/Tot.Calls	Name[id]
	45.642	52.774	1500/1500	POINT_SOLVER::POINT_SOLVE_5 [{point_solver.F90} {2700,3}-{2921,30}]
-->	45.642	52.774	1500	Loop: POINT_SOLVER::POINT_SOLVE_5 [{point_solver.F90} {2757,5}-{2917,19}]
	0.299	6.259	18000/30100	LMPI_APP::SINGLE_MATRIX_COMPLETE_XFER [{lmpi_app.F90} {11520,3}-{11626,44}]
	0.6	0.873	18000/30100	LMPI_APP::SINGLE_START_MATRIX_XFER [{lmpi_app.F90} {7907,3}-{8132,41}]
	37.689	37.689	78/78	JACOBIAN_VISCOUS::EDGEJP [{jacobian_viscous.f90} {324,14}]
-->	37.689	37.689	78	Loop: JACOBIAN_VISCOUS::EDGEJP [{jacobian_viscous.f90} {440,7}-{1584,22}]
	28.431	28.431	100/100	FLUX_PERFGAS::ROE_FLUX [{flux_perfgas.f90} {236,14}]
-->	28.431	28.431	100	Loop: FLUX_PERFGAS::ROE_FLUX [{flux_perfgas.f90} {397,5}-{597,20}]
	0.003	0.003	1197/27956	LMPI::CHARACTER_BCAST [{lmpi.F90} {3100,3}-{3136,32}]
	0.542	0.542	489/27956	LMPI::INTEGR_VECTOR_BCAST [{lmpi.F90} {3196,3}-{3232,36}]
	0.033	0.033	3/27956	LMPI::INTEG8_SCALAR_BCAST [{lmpi.F90} {3331,3}-{3367,36}]
	0.005	0.005	10/27956	LMPI::DOUBLE_MATRIX_BCAST [{lmpi.F90} {4140,3}-{4176,36}]
	16.724	16.724	31/27956	LMPI::INTEGR_MATRIX_BCAST [{lmpi.F90} {3240,3}-{3276,36}]
	0.032	0.032	1040/27956	LMPI::DOUBLE_SCALAR_BCAST [{lmpi.F90} {4051,3}-{4087,36}]
	1.48	1.48	5371/27956	LMPI::DOUBLE_VECTOR_BCAST [{lmpi.F90} {4096,3}-{4132,36}]
	1.5E-5	1.5E-5	2/27956	LMPI::LOGICL_VECTOR_BCAST [{lmpi.F90} {3736,3}-{3772,36}]
	0.002	0.002	2/27956	LMPI::DOUBLE_TENSOR_BCAST [{lmpi.F90} {4185,3}-{4214,36}]
	0.013	0.013	48/27956	LMPI::INTEG8_VECTOR_BCAST [{lmpi.F90} {3376,3}-{3412,36}]
	6.1E-4	6.1E-4	45/27956	PPARTY_METIS::MY_METIS [{pparty_metis.F90} {116,3}-{545,24}]
	5.481	5.481	18941/27956	LMPI::INTEGR_SCALAR_BCAST [{lmpi.F90} {3151,3}-{3187,36}]
	0.243	0.243	777/27956	LMPI::LOGICL_SCALAR_BCAST [{lmpi.F90} {3691,3}-{3727,36}]
-->	24.557	24.557	27956	MPI_Bcast()
	20.045	61.19	78/78	UPDATE_MEAN::UPDATE_JACOBIAN [{update_mean.F90} {513,3}-{588,32}]
-->	20.045	61.19	78	FILL_JACOBIANS::FILL_JACOBIAN [{fill_jacobians.f90} {19,3}-{341,30}]
	1.4E-4	1.4E-4	78/78	SOURCE::SOURCE_JACOBIAN [{source.f90} {93,3}-{168,32}]
	0.006	2.491	3822/16665	LMPI::LMPI_CONDITIONAL_STOP [{lmpi.F90} {611,3}-{672,38}]
	0.003	0.003	3822/8622	BC_NAMES::BC_HAS_PRESSURE_CLOSURE [{bc_names.f90} {1618,3}-{1693,38}]
	0.008	0.008	7644/17444	BC_NAMES::ELEMENT_BASED_BC [{bc_names.f90} {1390,3}-{1439,31}]
	3.2E-4	37.689	78/78	JACOBIAN_VISCOUS::VISCOUS_JACOBIAN [{jacobian_viscous.f90} {20,14}]
	0.443	0.445	78/123	TIMEACC::TIME_DIAG_NC [{timeacc.f90} {1067,3}-{1330,29}]

ParaProf Function Window

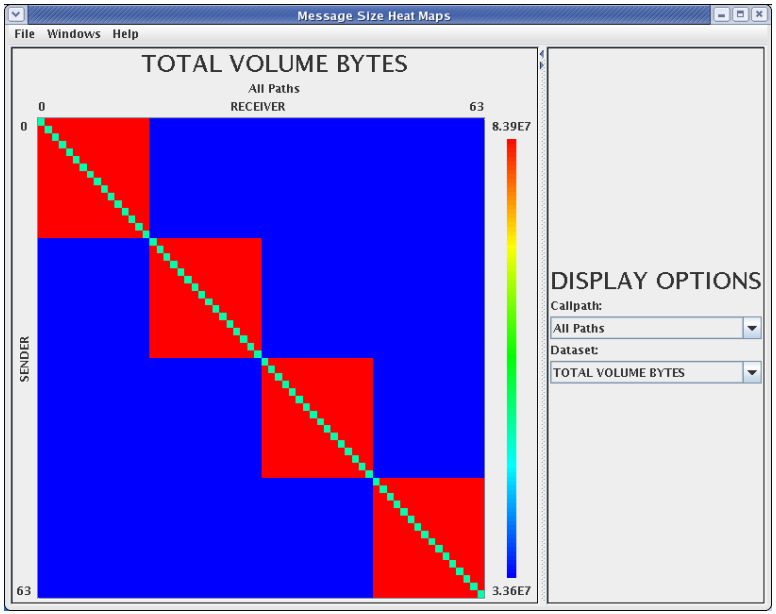


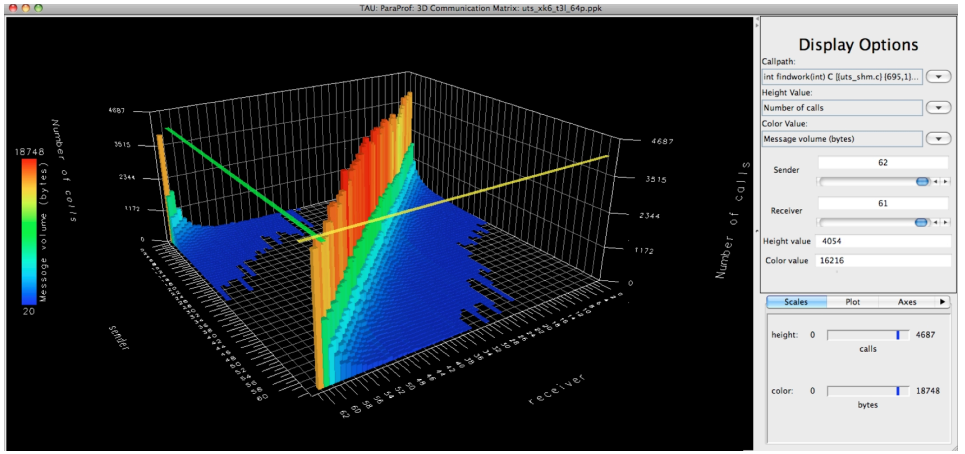
Generating Communication Matrix

```
% tau init --mpi --mpi-compilers Intel --compilers Intel
% tau meas edit sample --comm-matrix
% tau mpirun -np 16 ./a.out
% tau show
```

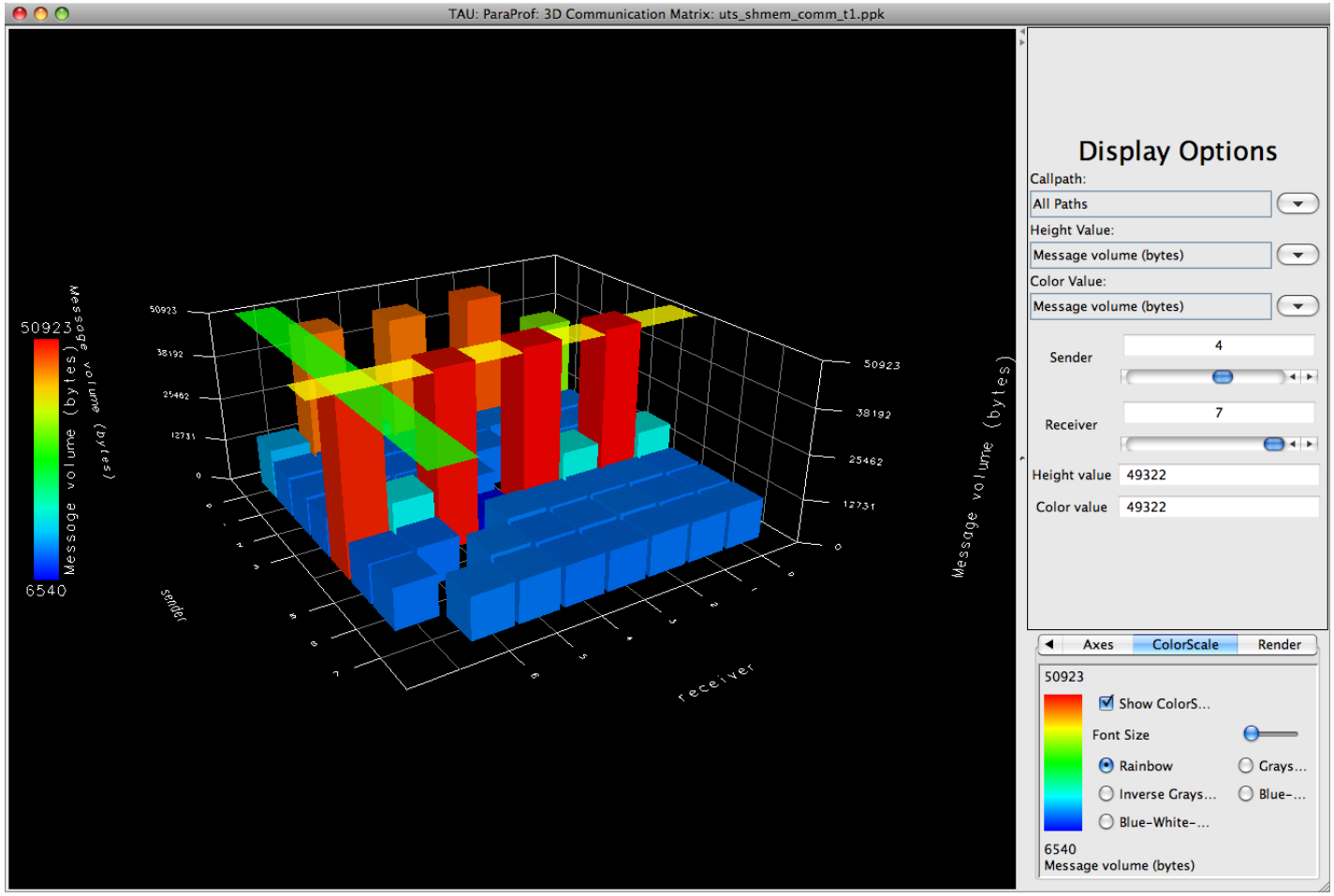
Communication Matrix Display

Goal: What is the volume of inter-process communication? Along which calling path?





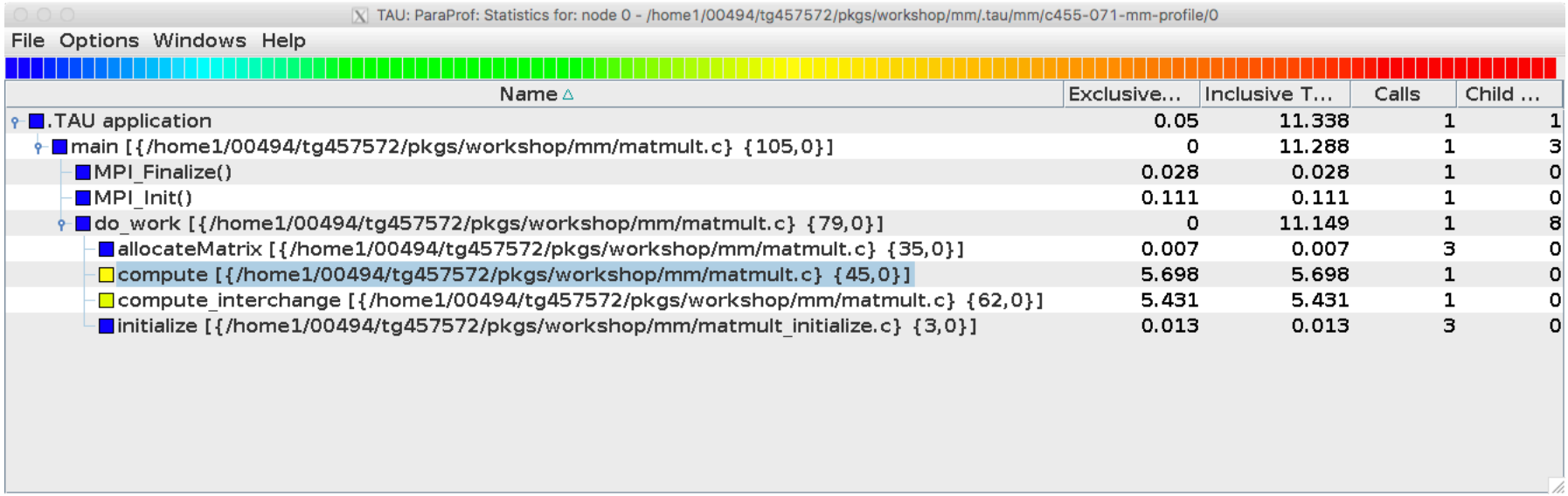
SHMEM Communication Matrix



Use Compiler-Based Instrumentation

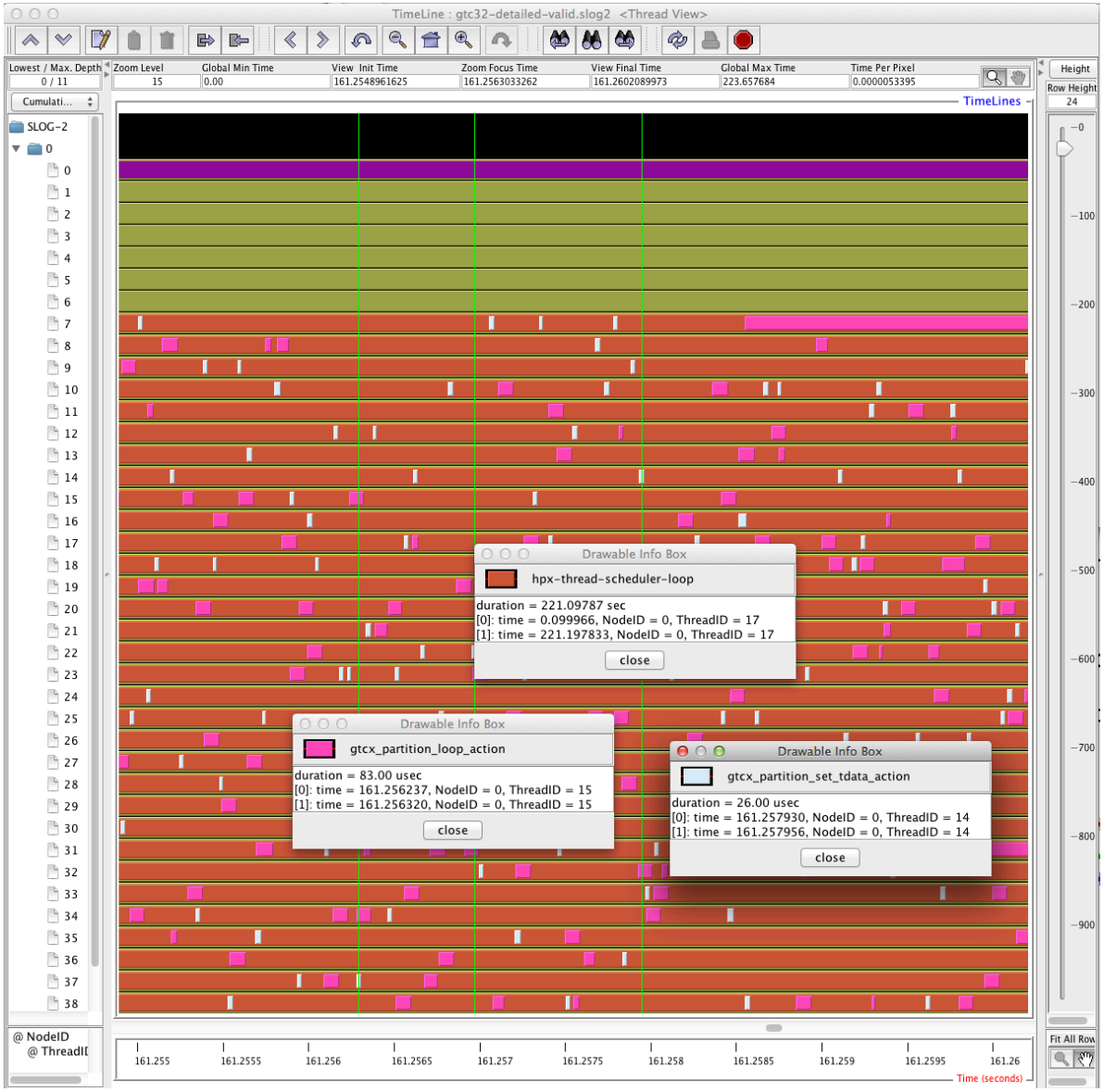
```
% tau init --mpi --mpi-compilers Intel --compilers Intel
% tau meas edit profile --compiler-inst always
% tau select profile
% make CC=‘tau mpiicc’
% tau mpirun –np 16 ./a.out
% tau show
```

Compiler-based instrumentation

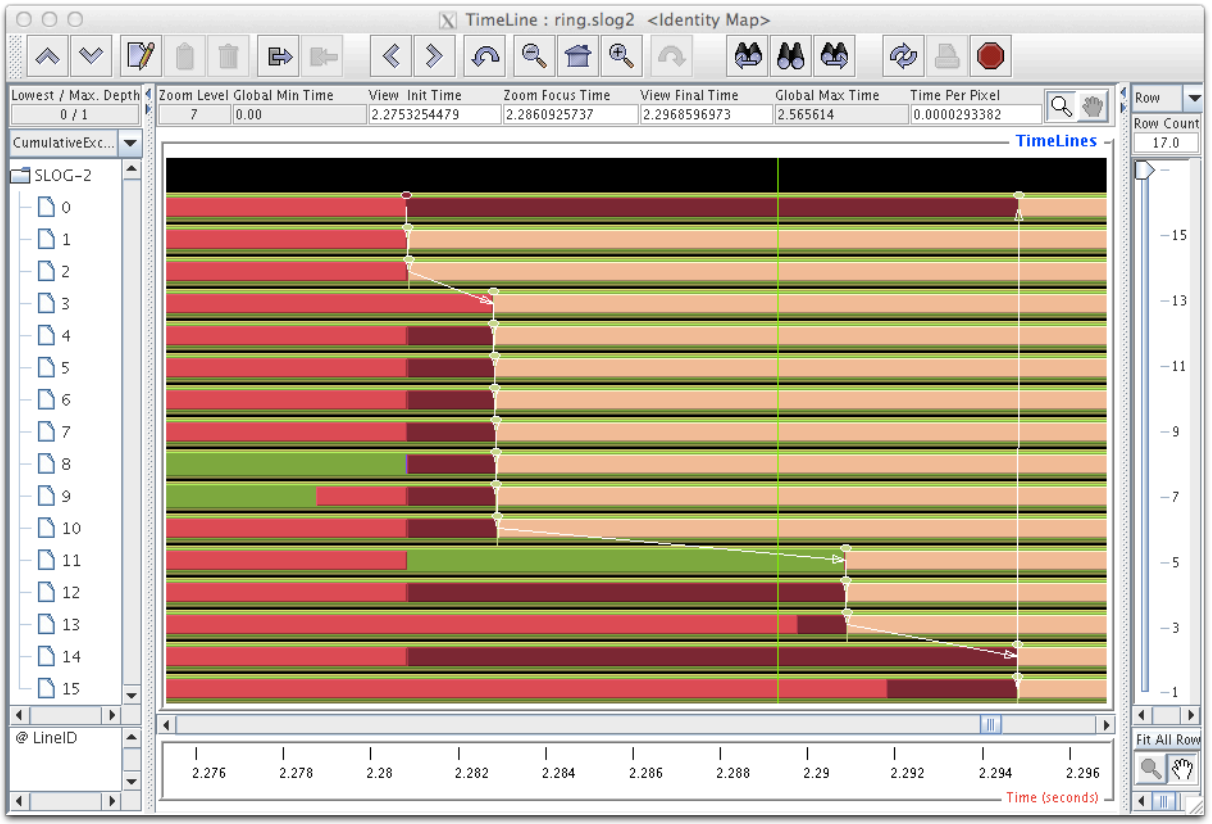


Name	Exclusive...	Inclusive T...	Calls	Child ...
.TAU application	0.05	11.338	1	1
main [{/home1/00494/tg457572/pkgs/workshop/mm/matmult.c} {105,0}]	0	11.288	1	3
MPI_Finalize()	0.028	0.028	1	0
MPI_Init()	0.111	0.111	1	0
do_work [{/home1/00494/tg457572/pkgs/workshop/mm/matmult.c} {79,0}]	0	11.149	1	8
allocateMatrix [{/home1/00494/tg457572/pkgs/workshop/mm/matmult.c} {35,0}]	0.007	0.007	3	0
compute [{/home1/00494/tg457572/pkgs/workshop/mm/matmult.c} {45,0}]	5.698	5.698	1	0
compute_interchange [{/home1/00494/tg457572/pkgs/workshop/mm/matmult.c} {62,0}]	5.431	5.431	1	0
initialize [{/home1/00494/tg457572/pkgs/workshop/mm/matmult_initialize.c} {3,0}]	0.013	0.013	3	0

Jumpshot Trace Visualizer in TAU

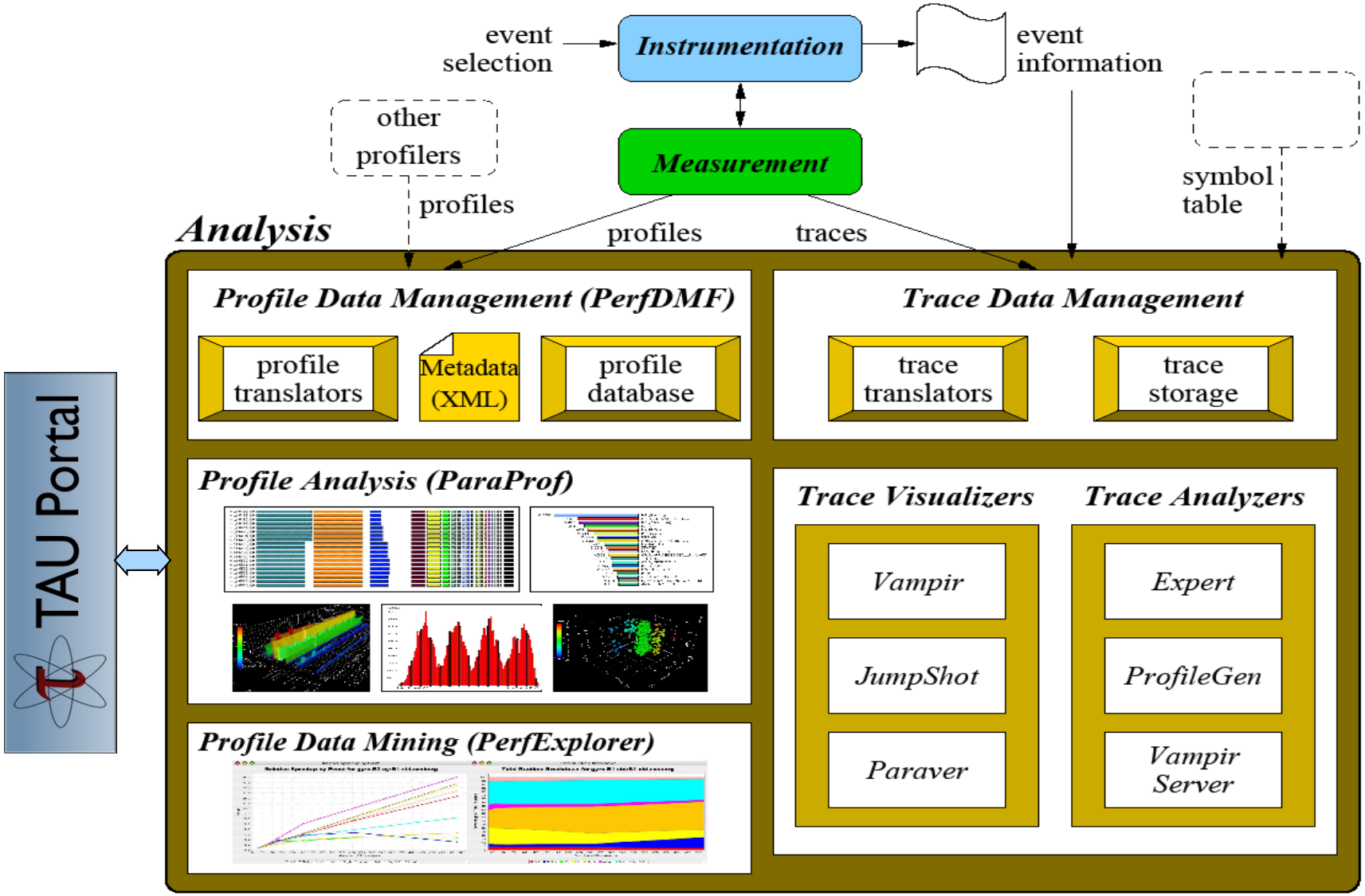


Tracing Communication in Jumpshot



% tau init

Performance Analysis



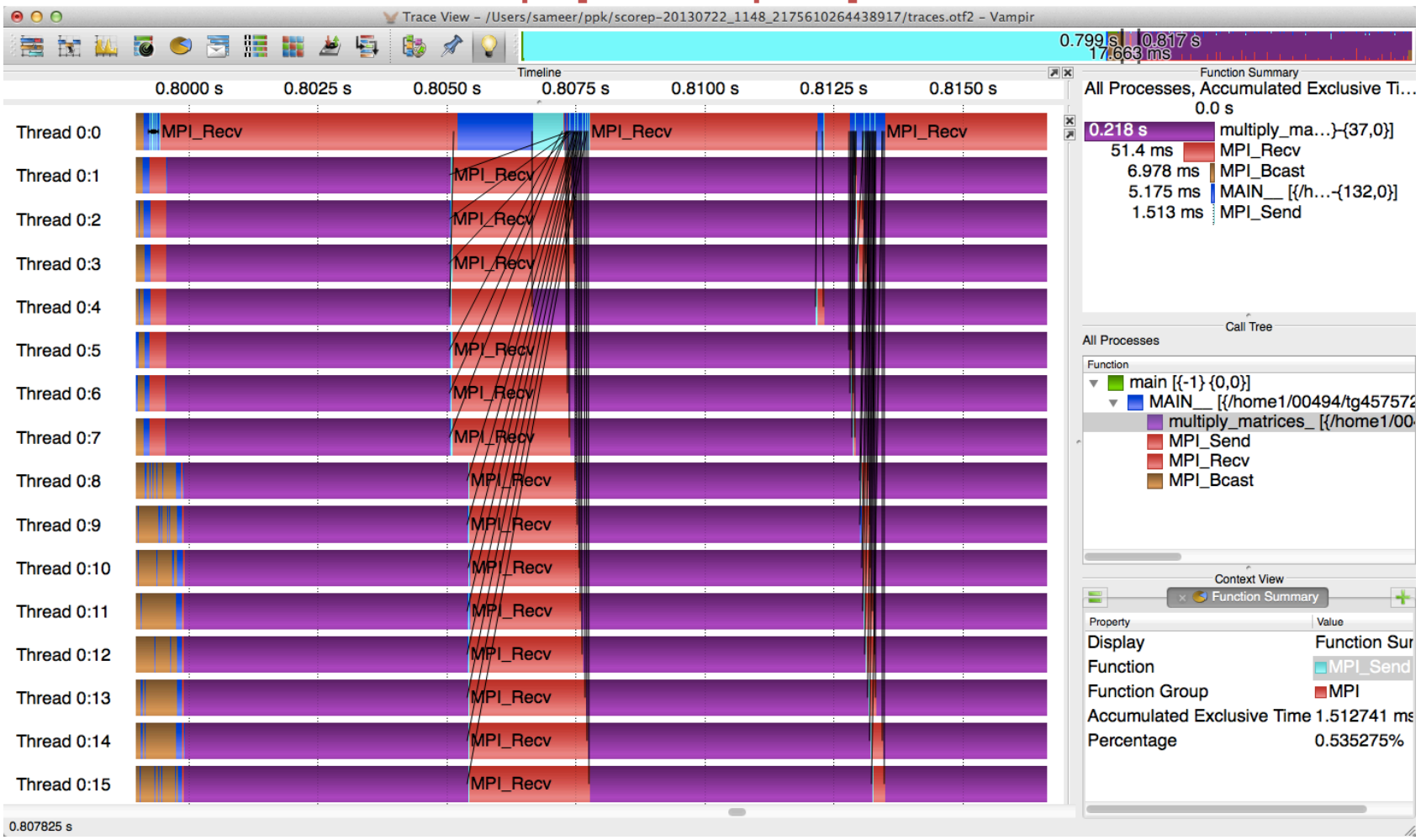
Generating Event Traces

```
% cd workshop/mstmult
% tau init --mpi --mpi-compilers Intel --compilers Intel
% tau meas edit profile --trace slog2
    (if it is profiling is being used in another experiment, you may
    have to delete it:
    tau experiment delete <exp_name>
    and retry

% make CC=‘tau mpiicc’
% tau mpirun -np 16 ./matmult
% tau show

NOTE: For OTF2 traces replace slog2 with otf2.
```

Visualizing Event Traces with Vampir

Goal: Identify the temporal aspect of performance. What happens in my code at a given time? When?

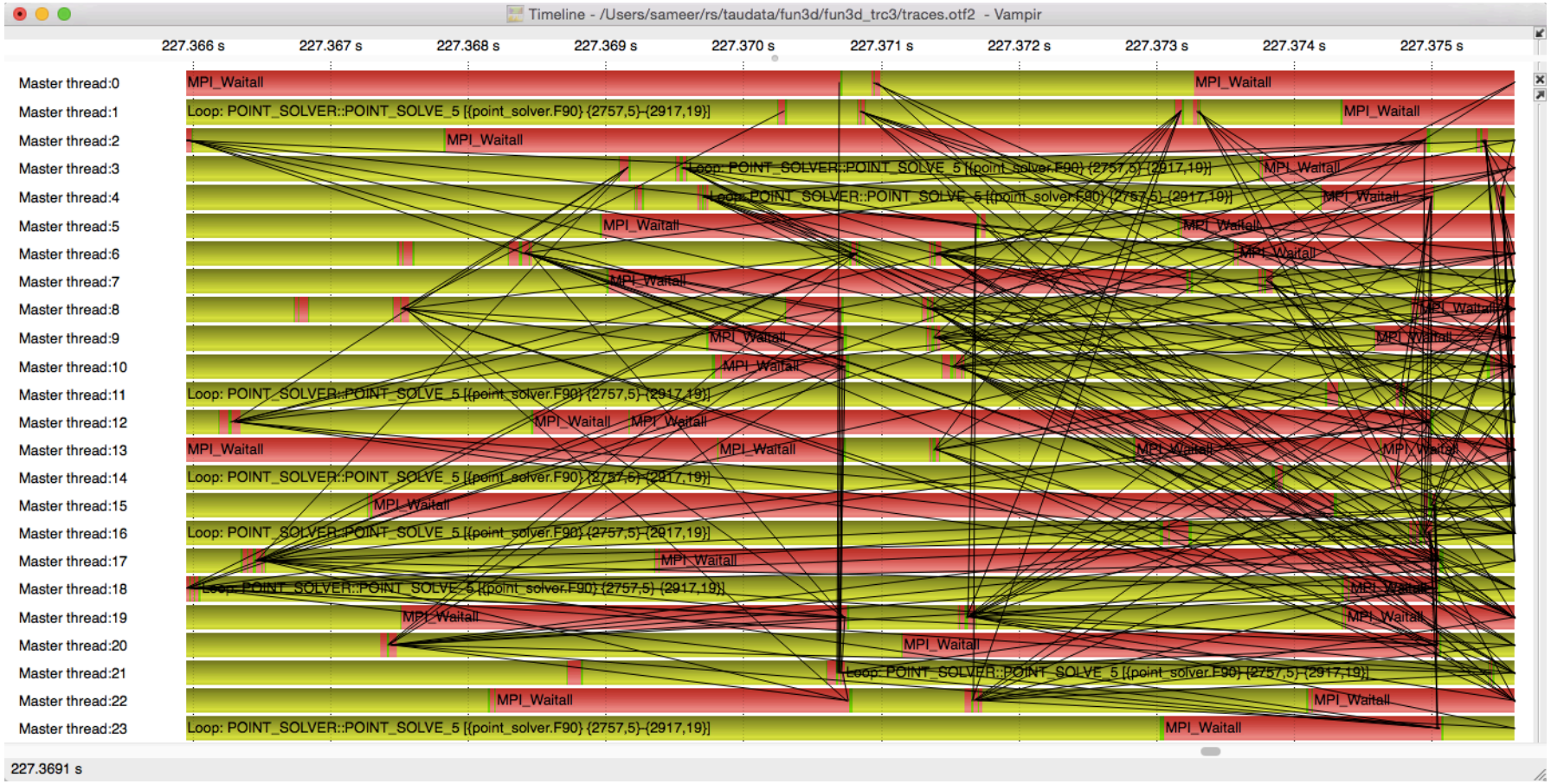
Event trace visualized in Vampir [www.vampir.eu]



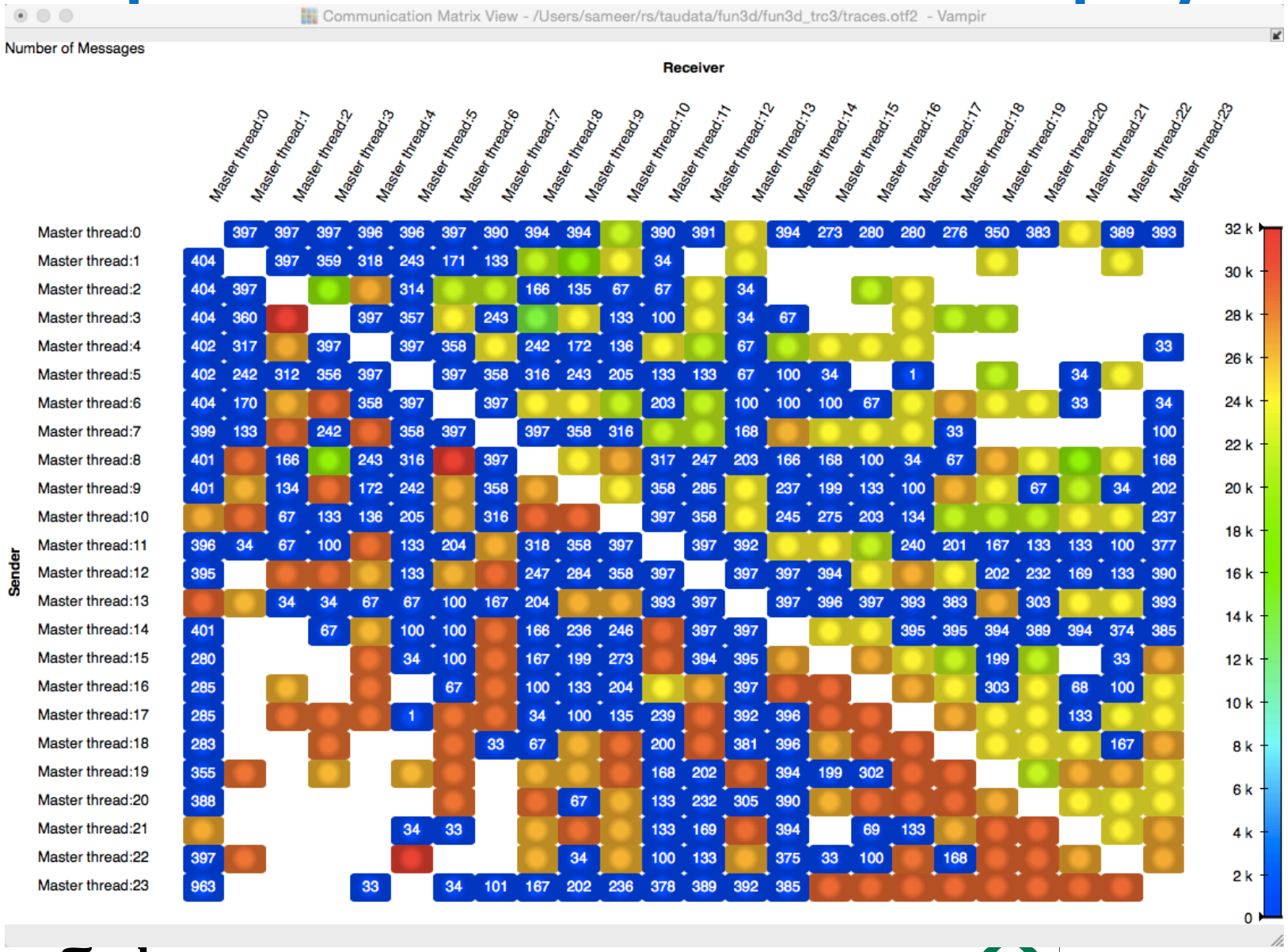
Vampir [T.U. Dresden] Timeline Display

Rank 0 performs I/O while all other ranks wait in MPI_Bcast()

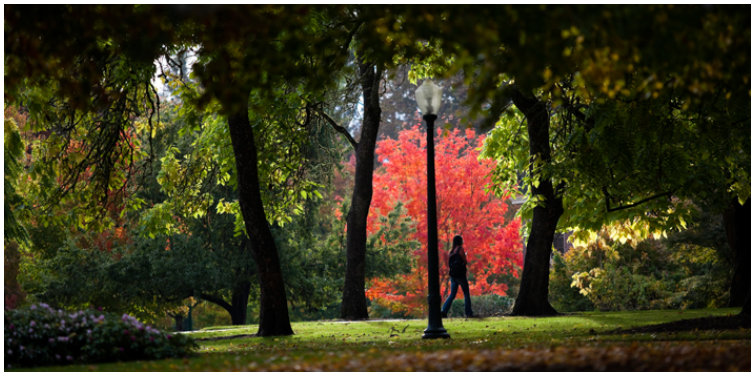
Vampir Timeline Display



Vampir Communication Matrix Display



PRL, University of Oregon, Eugene



www.uoregon.edu

Support Acknowledgments

US Department of Energy (DOE)

- Office of Science contracts
- SciDAC, LBL contracts
- LLNL-LANL-SNL ASC/NNSA contract
- Battelle, PNNL contract
- ANL, ORNL contract

Department of Defense (DoD)

- PETTT, HPCMP

National Science Foundation (NSF)

- Glassbox, SI-2

NASA

Partners:

University of Oregon

ParaTools, Inc.

University of Tennessee, Knoxville

T.U. Dresden, GWT

Juelich Supercomputing Center







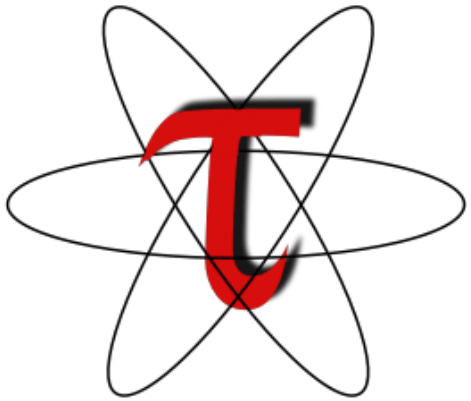









Download TAU from U. Oregon



http://www.hpclinux.com [LiveDVD]

Free download, open source, BSD license