Python Performance Evaluation with the TAU Performance System

John C. Linford, Sameer Shende, Allen Malony
{jlinford,sameer,malony}@paratools.com
ParaTools, Inc.

2 July 2015, EMiT'15
www.paratools.com/emit15/TAU

Tutorial Overview

- Performance optimization of Python applications
- We will cover:
  - Profiling and debugging via the TAU Performance System
  - Performance analysis of Python, C/C++, Fortran
  - Python+X analysis
  - MPI and/or OpenMP analysis
  - Memory debugging
  - Hardware performance counters (PAPI)

Schedule

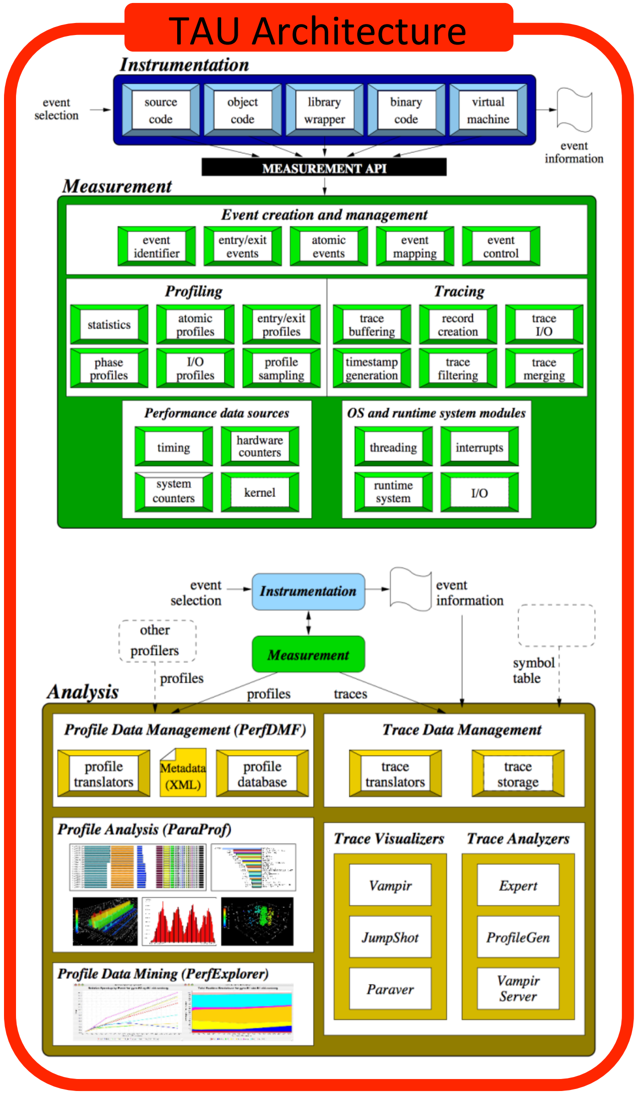
- The TAU Performance System from 10,000 feet
- Live demonstration of TAU + Python
- Hands-on TAU with:
  - Simple pure Python
  - Python + X
  - Let’s build a CTM…
    - With Ipython!

Python Performance Evaluation

THE TAU PERFORMANCE SYSTEM

The TAU Performance System®

- *Integrated toolkit* for performance problem solving
  - Instrumentation, measurement, analysis, visualization
  - Portable profiling and tracing
  - Performance data management and data mining
- Direct and indirect measurement
- *Free, open source, BSD license*
- Available on all HPC platforms (and many non-HPC)
- http://tau.uoregon.edu/



The TAU Performance System®

- 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

Questions TAU Can Answer

- **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, *vector instructions*?
- 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 **time spent waiting for collectives**?
- How does the application **scale**?

TAU Supports All HPC Platforms

C/C++ CUDA UPC Python

GPI

Fortran

OpenACC Java MPI

pthreads

Intel MIC OpenMP

Intel GNU

Sun

LLVM PGI Cray

MinGW

AIX

Linux Windows

Insert
yours
here

BlueGene Fujitsu ARM

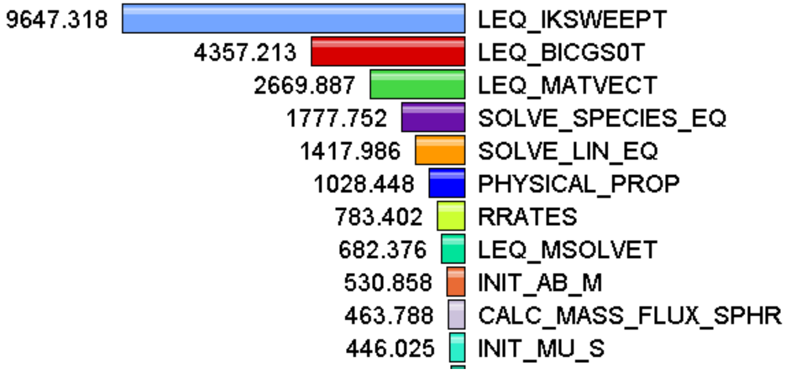
Android MPC OS X

Python Performance Evaluation

VOCABULARY

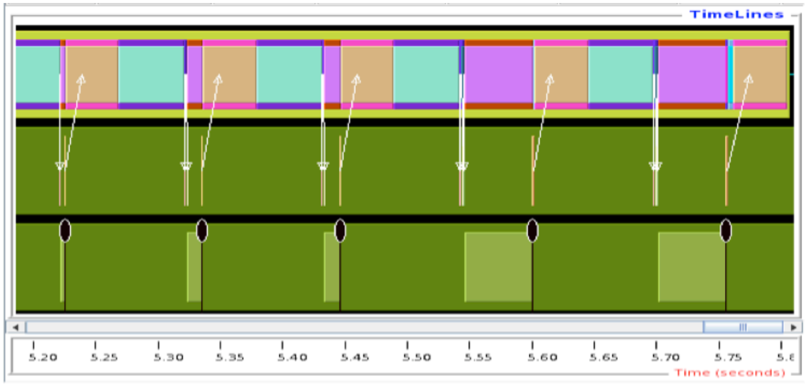
Measurement Approaches

Profiling



Shows **how much** time was spent in each routine

Tracing

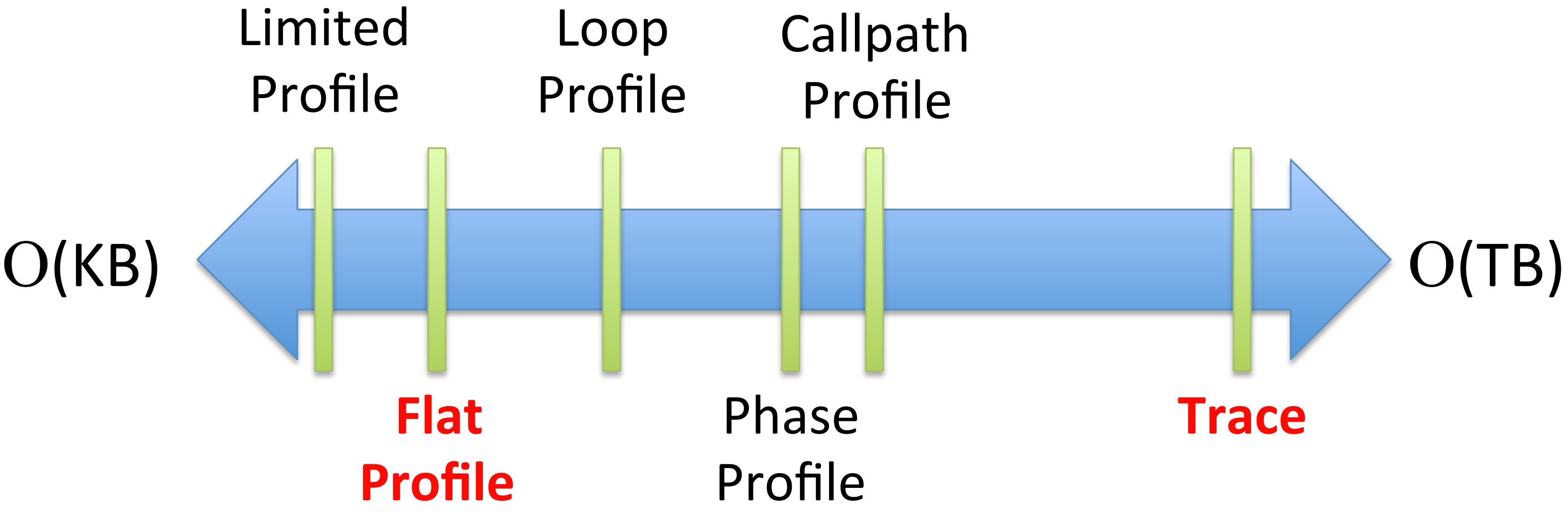


Shows **when** events take place on a timeline

Types of Performance Profiles

- *Flat* profiles
  - Metric (e.g., time) spent in an event
  - Exclusive/inclusive, # of calls, child calls, …
- *Callpath* profiles
  - Time spent along a calling path (edges in callgraph)
  - *"main=> f1 => f2 => MPI_Send"*
  - Set the **TAU_CALLPATH_DEPTH** environment variable
- *Phase* profiles
  - Flat profiles under a phase (nested phases allowed)
  - Default "main" phase
  - Supports static or dynamic (e.g. per-iteration) phases

How much data do you want?



All levels support multiple metrics/counters

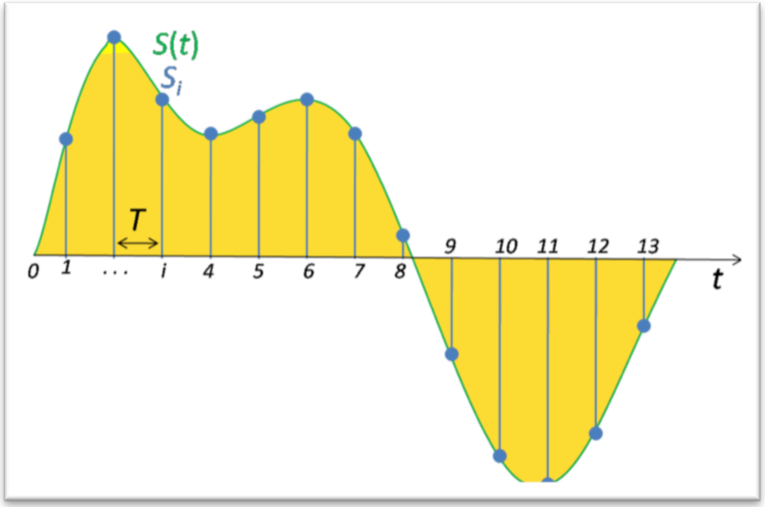
Performance Data Measurement

Direct via Probes

```
call TAU_START('potential')
// code
call TAU_STOP('potential')
```

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

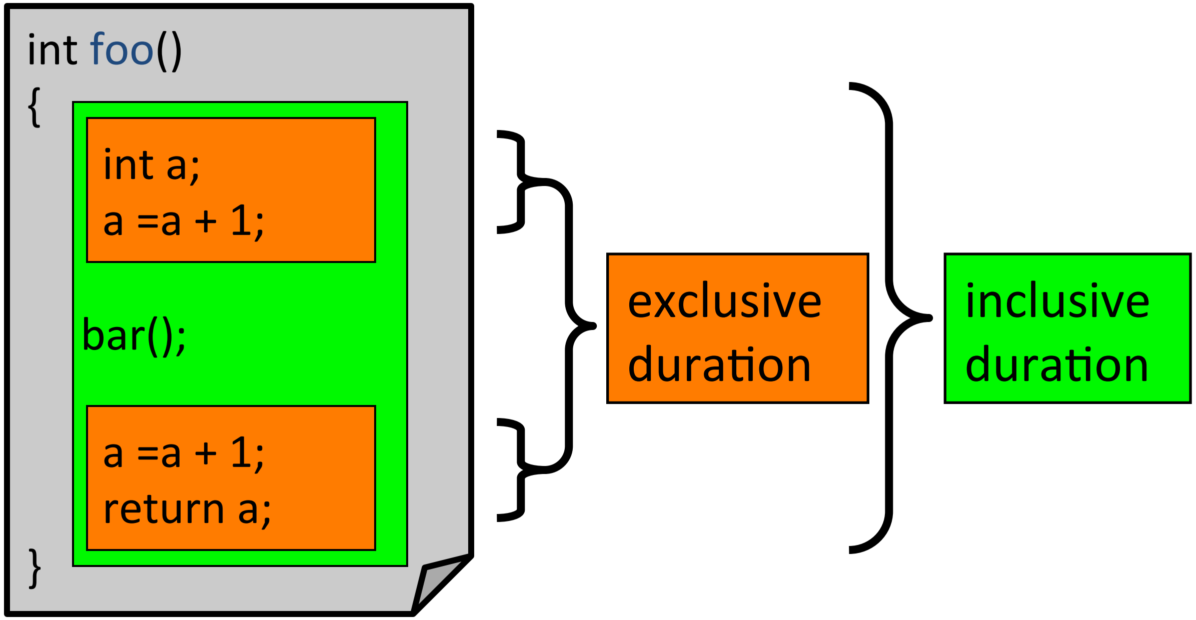
Indirect via Sampling



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

Inclusive vs. Exclusive Measurements

- **Exclusive** measurements for **region only**
- **Inclusive** measurements **includes child** regions



```
int foo()
{
    int a;
    a =a + 1;

    bar();

    a =a + 1;
    return a;
}
```

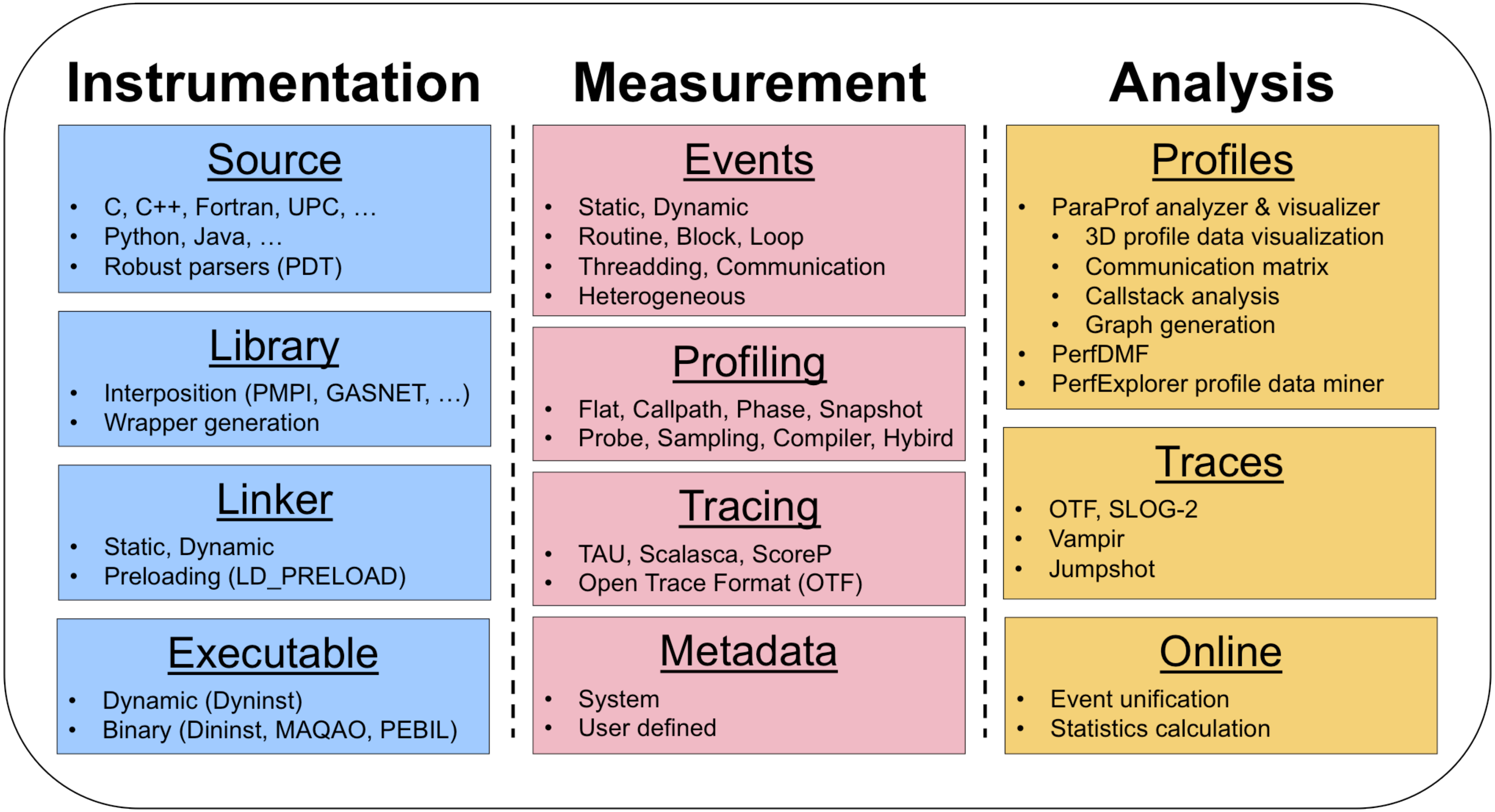
exclusive duration

inclusive duration

Python Performance Evaluation

PERFORMANCE ANALYSIS WORKFLOW

TAU Architecture and Workflow



Instrumentation

Source

- C, C++, Fortran, UPC, …
- Python, Java, …
- Robust parsers (PDT)

Library

- Interposition (PMPI, GASNET, …)
- Wrapper generation

Linker

- Static, Dynamic
- Preloading (LD_PRELOAD)

Executable

- Dynamic (Dyninst)
- Binary (Dininst, MAQAO, PEBIL)

Measurement

Events

- Static, Dynamic
- Routine, Block, Loop
- Threadding, Communication
- Heterogeneous

Profiling

- Flat, Callpath, Phase, Snapshot
- Probe, Sampling, Compiler, Hybird

Tracing

- TAU, Scalasca, ScoreP
- Open Trace Format (OTF)

Metadata

- System
- User defined

Analysis

Profiles

- ParaProf analyzer & visualizer
  - 3D profile data visualization
  - Communication matrix
  - Callstack analysis
  - Graph generation
- PerfDMF
- PerfExplorer profile data miner

Traces

- OTF, SLOG-2
- Vampir
- Jumpshot

Online

- Event unification
- Statistics calculation

Instrument: Add Probes

- *Source code* instrumentation
  - PDT parsers, pre-processors
- *Wrap* external libraries
  - I/O, MPI, Memory, CUDA, OpenCL, pthread
- *Rewrite* the binary executable
  - Dyninst, MAQAO

Insert TAU API Calls Automatically

- Use TAU's compiler wrappers
  - Replace C++ compiler with `tau_cxx.sh`, etc.
  - Automatically instruments source code, links with TAU libraries.
- Use `tau_cc.sh` for C, `tau_f90.sh` for Fortran, etc.

Makefile without TAU

```
CXX = mpicxx
F90 = mpif90
CXXFLAGS =
LIBS = -lm
OBJS = f1.o f2.o f3.o … fn.o

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

Makefile with TAU

```
CXX = tau_cxx.sh
F90 = tau_f90.sh
CXXFLAGS =
LIBS = -lm
OBJS = f1.o f2.o f3.o … fn.o

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

Measure: Gather Data

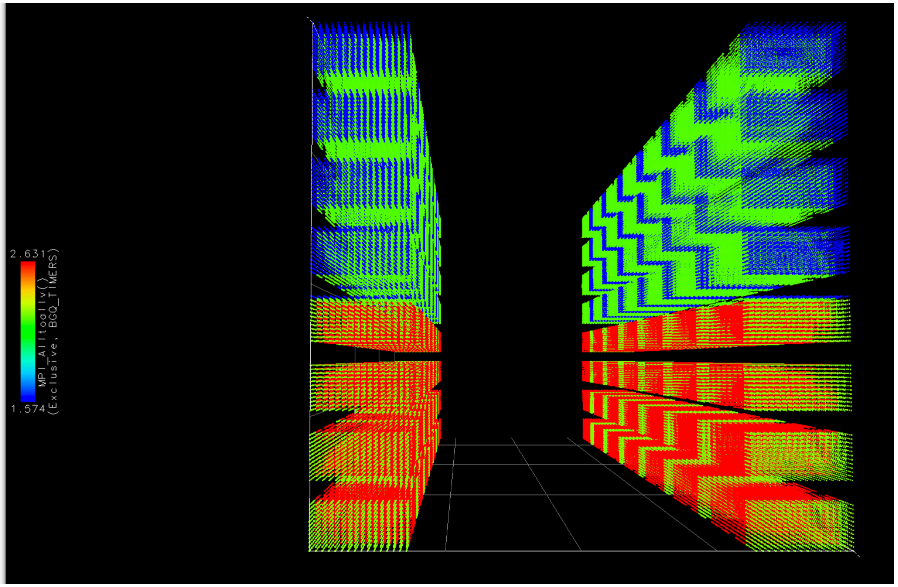
- Direct measurement via *probes*
- Indirect measurement via *sampling*
- Throttling and runtime control
- Interface with external packages (PAPI)

Direct Observation Events

- Interval events (begin/end events)
  - Measures exclusive & inclusive durations between events
  - Metrics monotonically increase
  - Example: Wall-clock timer
- Atomic events (trigger with data value)
  - Used to capture performance data state
  - Shows extent of variation of triggered values (min/max/mean)
  - Example: heap memory consumed at a particular point
- Code events
  - Routines, classes, templates
  - Statement-level blocks, loops
  - Example: for-loop begin/end

Analyze: Synthesize Knowledge

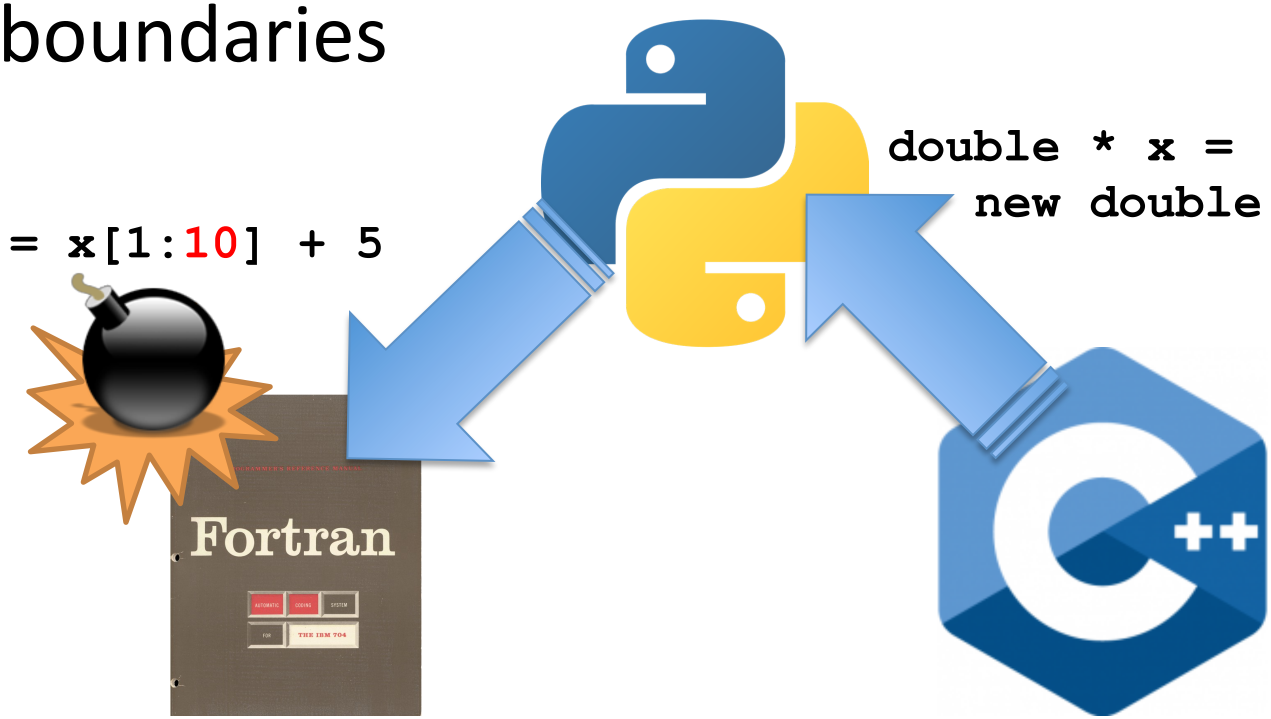
- Data *visualization*



- Data *mining*
- Statistical analysis
- Import/export performance data

Multi-Language Debugging

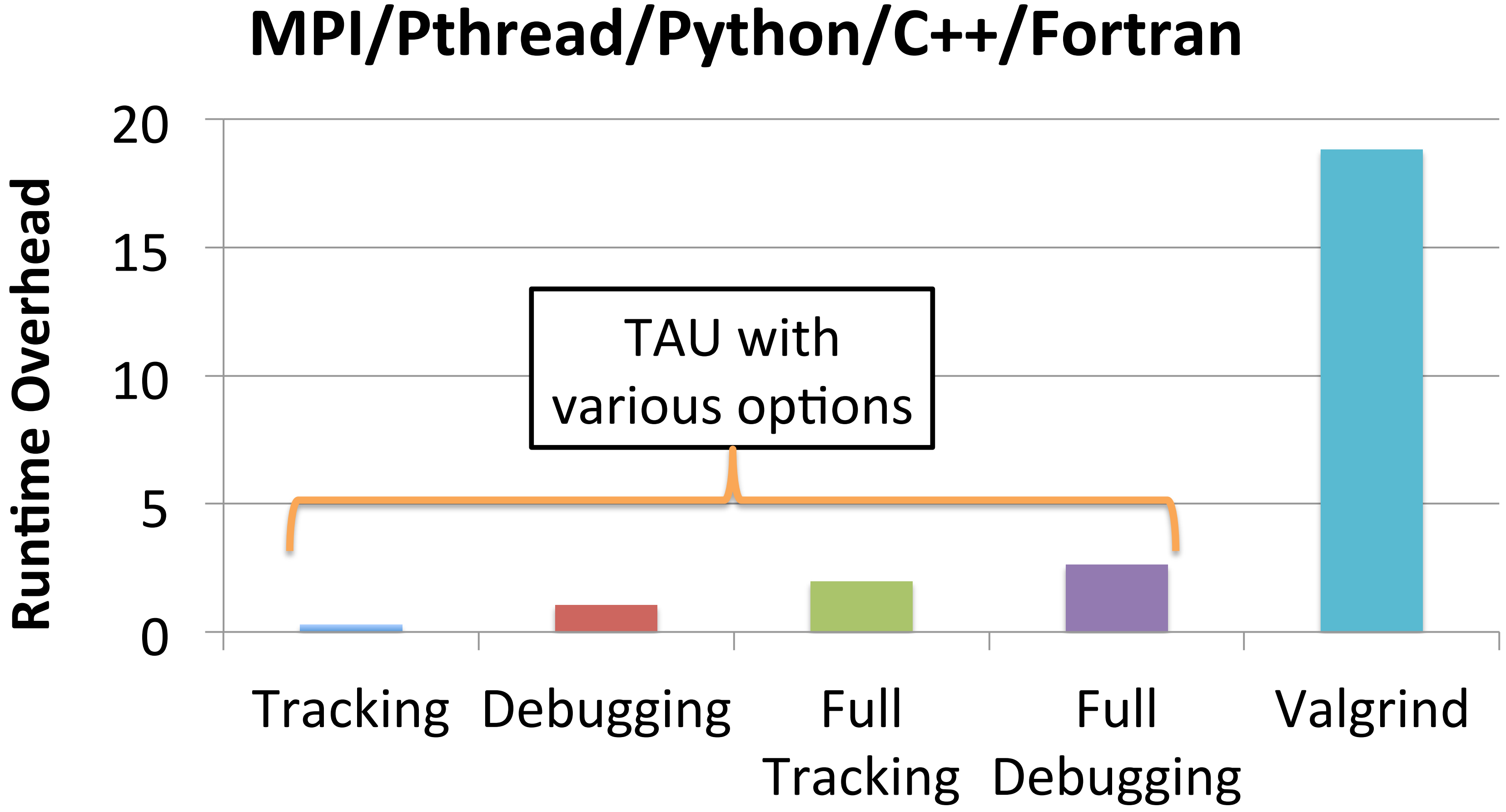
- Identify the source location of a crash by unwinding the system callstack
- Identify memory errors (off-by-one, etc.) across language boundaries



```
A = x[1:10] + 5
```

```
double * x =
    new double[5];
```

Memory debugging

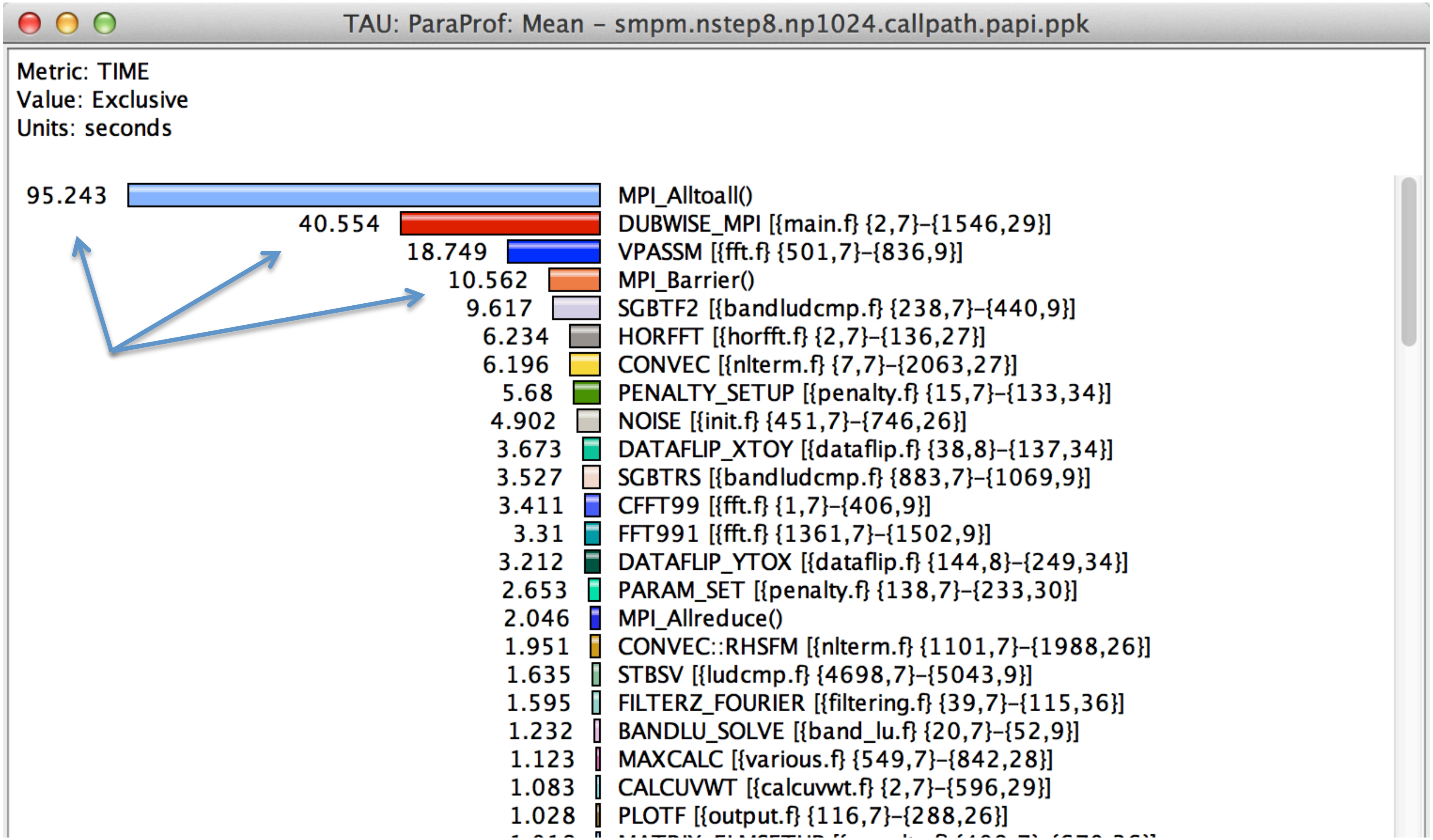


Note: Requires working mprotect() so BGQ not supported

Python Performance Evaluation

ANALYSIS EXAMPLES

How Much Time per Code Region?



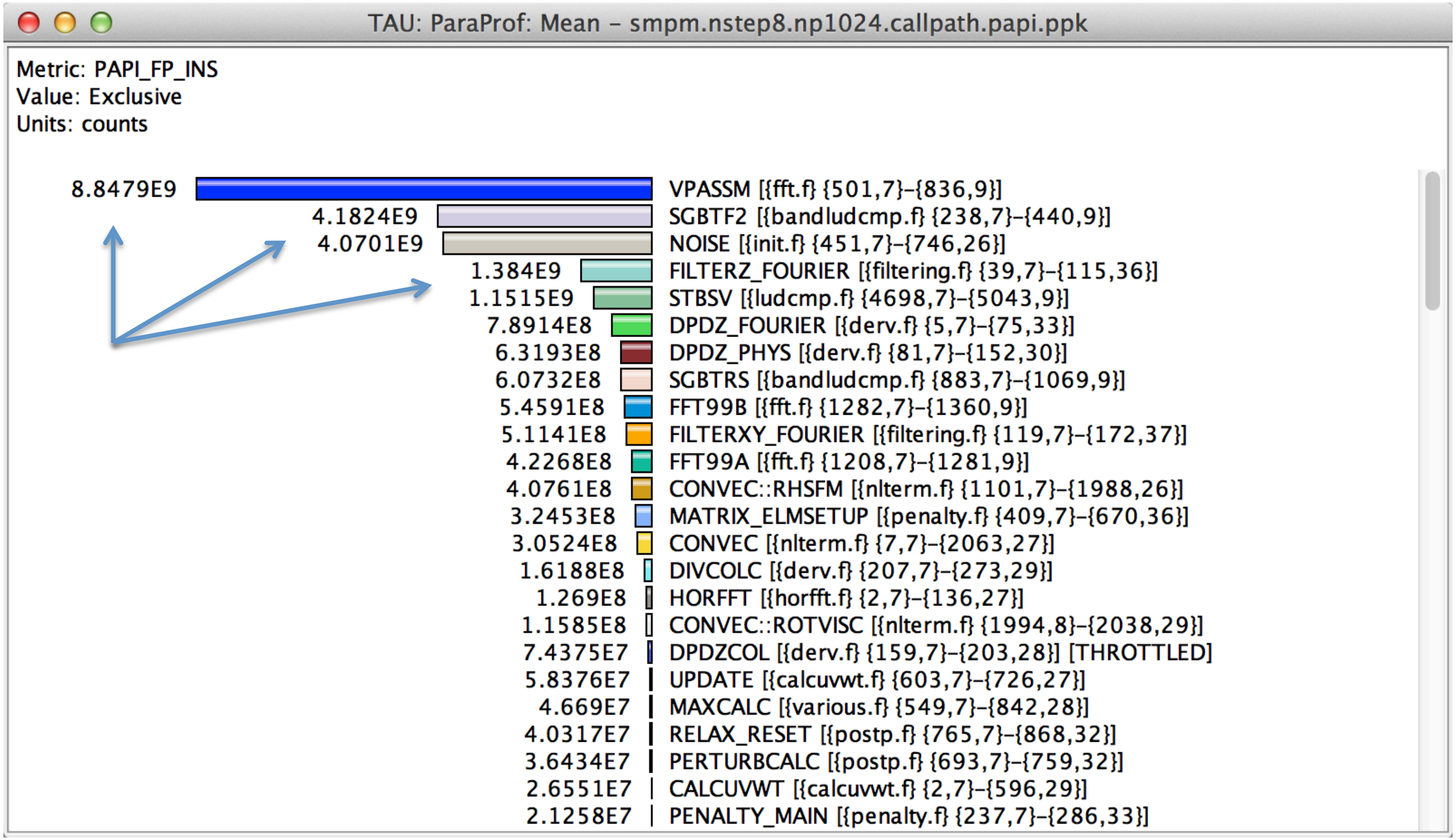
TAU: ParaProf: Mean – smpm.nstep8.np1024.callpath.papi.ppk

Metric: TIME
Value: Exclusive
Units: seconds

```
95.243  MPI_Alltoall()
40.554  DUBWISE_MPI [{main.f} {2,7}-{1546,29}]
18.749  VPASSM [{fft.f} {501,7}-{836,9}]
10.562  MPI_Barrier()
 9.617  SGBTF2 [{bandludcmp.f} {238,7}-{440,9}]
 6.234  HORFFT [{horfft.f} {2,7}-{136,27}]
 6.196  CONVEC [{nlterm.f} {7,7}-{2063,27}]
 5.68   PENALTY_SETUP [{penalty.f} {15,7}-{133,34}]
 4.902  NOISE [{init.f} {451,7}-{746,26}]
 3.673  DATAFLIP_XTOY [{dataflip.f} {38,8}-{137,34}]
 3.527  SGBTRS [{bandludcmp.f} {883,7}-{1069,9}]
 3.411  CFFT99 [{fft.f} {1,7}-{406,9}]
 3.31   FFT991 [{fft.f} {1361,7}-{1502,9}]
 3.212  DATAFLIP_YTOX [{dataflip.f} {144,8}-{249,34}]
 2.653  PARAM_SET [{penalty.f} {138,7}-{233,30}]
 2.046  MPI_Allreduce()
 1.951  CONVEC::RHSFM [{nlterm.f} {1101,7}-{1988,26}]
 1.635  STBSV [{ludcmp.f} {4698,7}-{5043,9}]
 1.595  FILTERZ_FOURIER [{filtering.f} {39,7}-{115,36}]
 1.232  BANDLU_SOLVE [{band_lu.f} {20,7}-{52,9}]
 1.123  MAXCALC [{various.f} {549,7}-{842,28}]
 1.083  CALCUVWT [{calcuvwt.f} {2,7}-{596,29}]
 1.028  PLOTF [{output.f} {116,7}-{288,26}]
```

`% paraprof` **(Click on label, e.g. "Mean" or "node 0")**

How Many Instructions per Code Region?



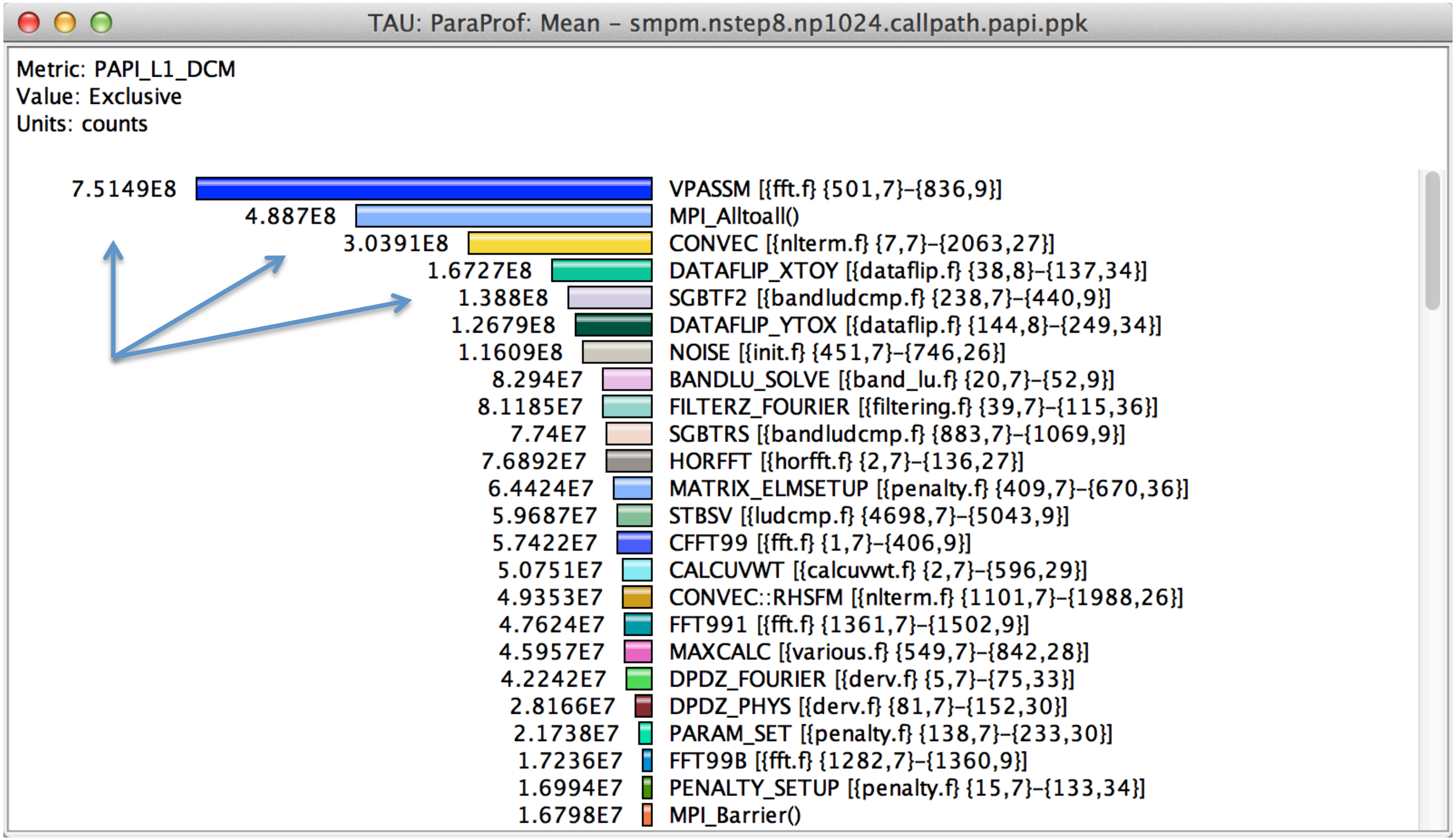
TAU: ParaProf: Mean – smpm.nstep8.np1024.callpath.papi.ppk

Metric: PAPI_FP_INS
Value: Exclusive
Units: counts

```
8.8479E9   VPASSM [{fft.f} {501,7}-{836,9}]
4.1824E9   SGBTF2 [{bandludcmp.f} {238,7}-{440,9}]
4.0701E9   NOISE [{init.f} {451,7}-{746,26}]
1.384E9    FILTERZ_FOURIER [{filtering.f} {39,7}-{115,36}]
1.1515E9   STBSV [{ludcmp.f} {4698,7}-{5043,9}]
7.8914E8   DPDZ_FOURIER [{derv.f} {5,7}-{75,33}]
6.3193E8   DPDZ_PHYS [{derv.f} {81,7}-{152,30}]
6.0732E8   SGBTRS [{bandludcmp.f} {883,7}-{1069,9}]
5.4591E8   FFT99B [{fft.f} {1282,7}-{1360,9}]
5.1141E8   FILTERXY_FOURIER [{filtering.f} {119,7}-{172,37}]
4.2268E8   FFT99A [{fft.f} {1208,7}-{1281,9}]
4.0761E8   CONVEC::RHSFM [{nlterm.f} {1101,7}-{1988,26}]
3.2453E8   MATRIX_ELMSETUP [{penalty.f} {409,7}-{670,36}]
3.0524E8   CONVEC [{nlterm.f} {7,7}-{2063,27}]
1.6188E8   DIVCOLC [{derv.f} {207,7}-{273,29}]
1.269E8    HORFFT [{horfft.f} {2,7}-{136,27}]
1.1585E8   CONVEC::ROTVISC [{nlterm.f} {1994,8}-{2038,29}]
7.4375E7   DPDZCOL [{derv.f} {159,7}-{203,28}] [THROTTLED]
5.8376E7   UPDATE [{calcuvwt.f} {603,7}-{726,27}]
4.669E7    MAXCALC [{various.f} {549,7}-{842,28}]
4.0317E7   RELAX_RESET [{postp.f} {765,7}-{868,32}]
3.6434E7   PERTURBCALC [{postp.f} {693,7}-{759,32}]
2.6551E7   CALCUVWT [{calcuvwt.f} {2,7}-{596,29}]
2.1258E7   PENALTY_MAIN [{penalty.f} {237,7}-{286,33}]
```

`% paraprof` **(Options → Select Metric... → Exclusive... → PAPI_FP_INS)**

How Many L1 or L2 Cache Misses?



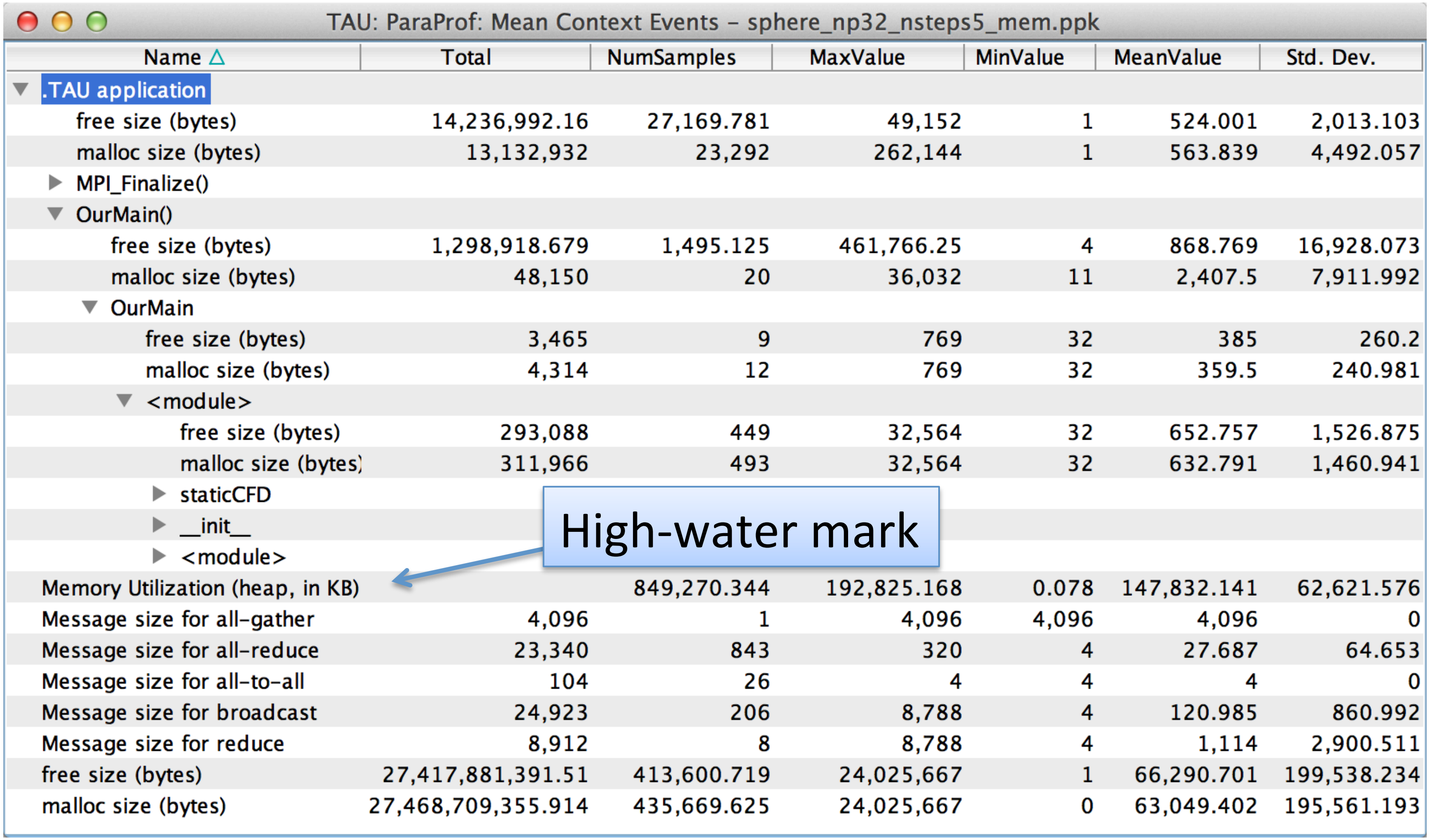
TAU: ParaProf: Mean – smpm.nstep8.np1024.callpath.papi.ppk

Metric: PAPI_L1_DCM
Value: Exclusive
Units: counts

```
7.5149E8   VPASSM [{fft.f} {501,7}-{836,9}]
4.887E8    MPI_Alltoall()
3.0391E8   CONVEC [{nlterm.f} {7,7}-{2063,27}]
1.6727E8   DATAFLIP_XTOY [{dataflip.f} {38,8}-{137,34}]
1.388E8    SGBTF2 [{bandludcmp.f} {238,7}-{440,9}]
1.2679E8   DATAFLIP_YTOX [{dataflip.f} {144,8}-{249,34}]
1.1609E8   NOISE [{init.f} {451,7}-{746,26}]
8.294E7    BANDLU_SOLVE [{band_lu.f} {20,7}-{52,9}]
8.1185E7   FILTERZ_FOURIER [{filtering.f} {39,7}-{115,36}]
7.74E7     SGBTRS [{bandludcmp.f} {883,7}-{1069,9}]
7.6892E7   HORFFT [{horfft.f} {2,7}-{136,27}]
6.4424E7   MATRIX_ELMSETUP [{penalty.f} {409,7}-{670,36}]
5.9687E7   STBSV [{ludcmp.f} {4698,7}-{5043,9}]
5.7422E7   CFFT99 [{fft.f} {1,7}-{406,9}]
5.0751E7   CALCUVWT [{calcuvwt.f} {2,7}-{596,29}]
4.9353E7   CONVEC::RHSFM [{nlterm.f} {1101,7}-{1988,26}]
4.7624E7   FFT991 [{fft.f} {1361,7}-{1502,9}]
4.5957E7   MAXCALC [{various.f} {549,7}-{842,28}]
4.2242E7   DPDZ_FOURIER [{derv.f} {5,7}-{75,33}]
2.8166E7   DPDZ_PHYS [{derv.f} {81,7}-{152,30}]
2.1738E7   PARAM_SET [{penalty.f} {138,7}-{233,30}]
1.7236E7   FFT99B [{fft.f} {1282,7}-{1360,9}]
1.6994E7   PENALTY_SETUP [{penalty.f} {15,7}-{133,34}]
1.6798E7   MPI_Barrier()
```

`% paraprof` **(Options → Select Metric... → Exclusive... → PAPI_L1_DCM)**

How Much Memory Does the Code Use?



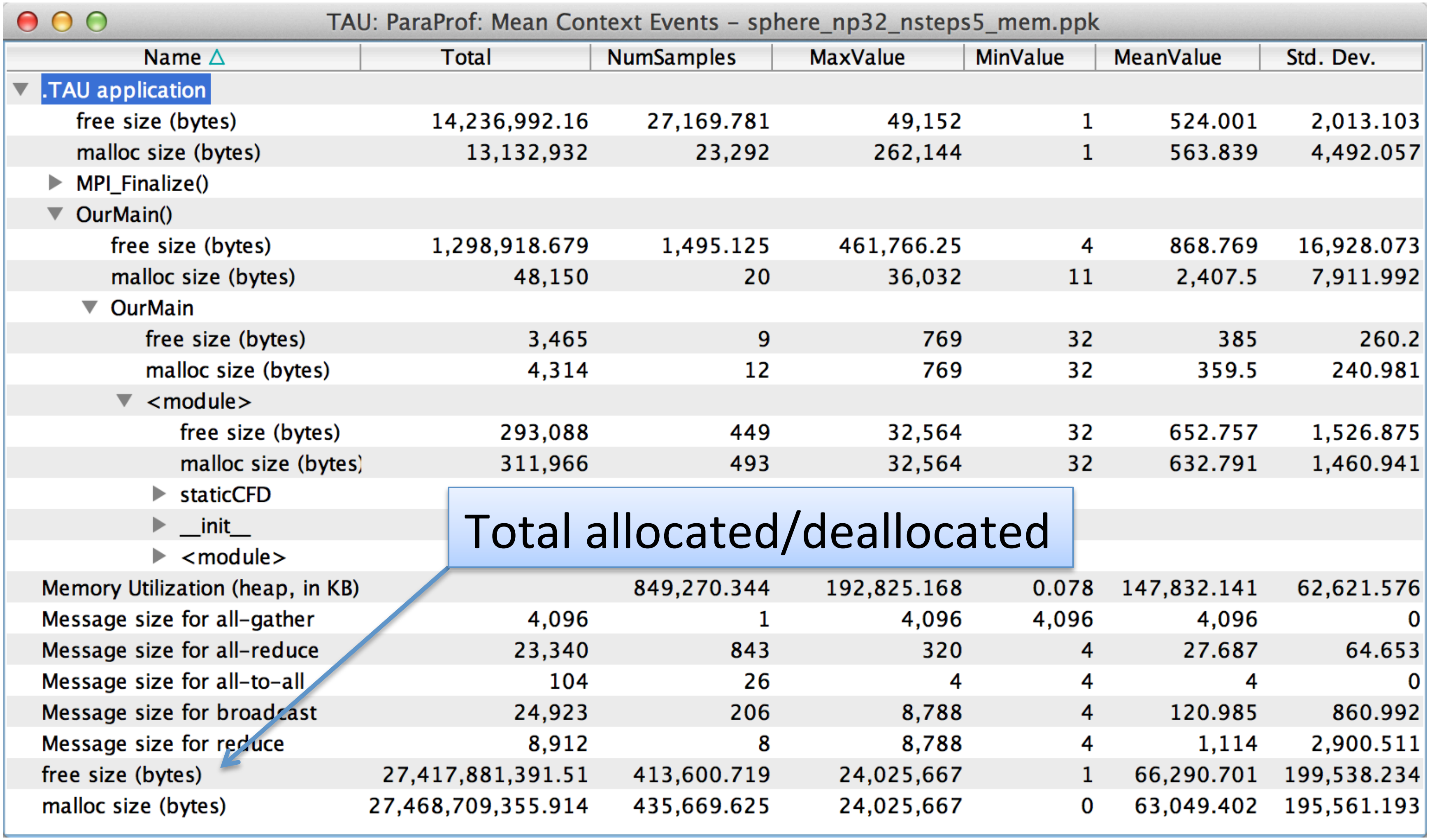
TAU: ParaProf: Mean Context Events – sphere_np32_nsteps5_mem.ppk

Name △	Total	NumSamples	MaxValue	MinValue	MeanValue	Std. Dev.
▼ .TAU application						
free size (bytes)	14,236,992.16	27,169.781	49,152	1	524.001	2,013.103
malloc size (bytes)	13,132,932	23,292	262,144	1	563.839	4,492.057
▶ MPI_Finalize()						
▼ OurMain()						
free size (bytes)	1,298,918.679	1,495.125	461,766.25	4	868.769	16,928.073
malloc size (bytes)	48,150	20	36,032	11	2,407.5	7,911.992
▼ OurMain						
free size (bytes)	3,465	9	769	32	385	260.2
malloc size (bytes)	4,314	12	769	32	359.5	240.981
▼ <module>						
free size (bytes)	293,088	449	32,564	32	652.757	1,526.875
malloc size (bytes)	311,966	493	32,564	32	632.791	1,460.941
▶ staticCFD						
▶ __init__						
▶ <module>						
Memory Utilization (heap, in KB)		849,270.344	192,825.168	0.078	147,832.141	62,621.576
Message size for all-gather	4,096	1	4,096	4,096	4,096	0
Message size for all-reduce	23,340	843	320	4	27.687	64.653
Message size for all-to-all	104	26	4	4	4	0
Message size for broadcast	24,923	206	8,788	4	120.985	860.992
Message size for reduce	8,912	8	8,788	4	1,114	2,900.511
free size (bytes)	27,417,881,391.51	413,600.719	24,025,667	1	66,290.701	199,538.234
malloc size (bytes)	27,468,709,355.914	435,669,625	24,025,667	0	63,049.402	195,561.193

High-water mark

% **`paraprof`** **(Right-click label [e.g "node 0"] → Show Context Event Window)**

How Much Memory Does the Code Use?



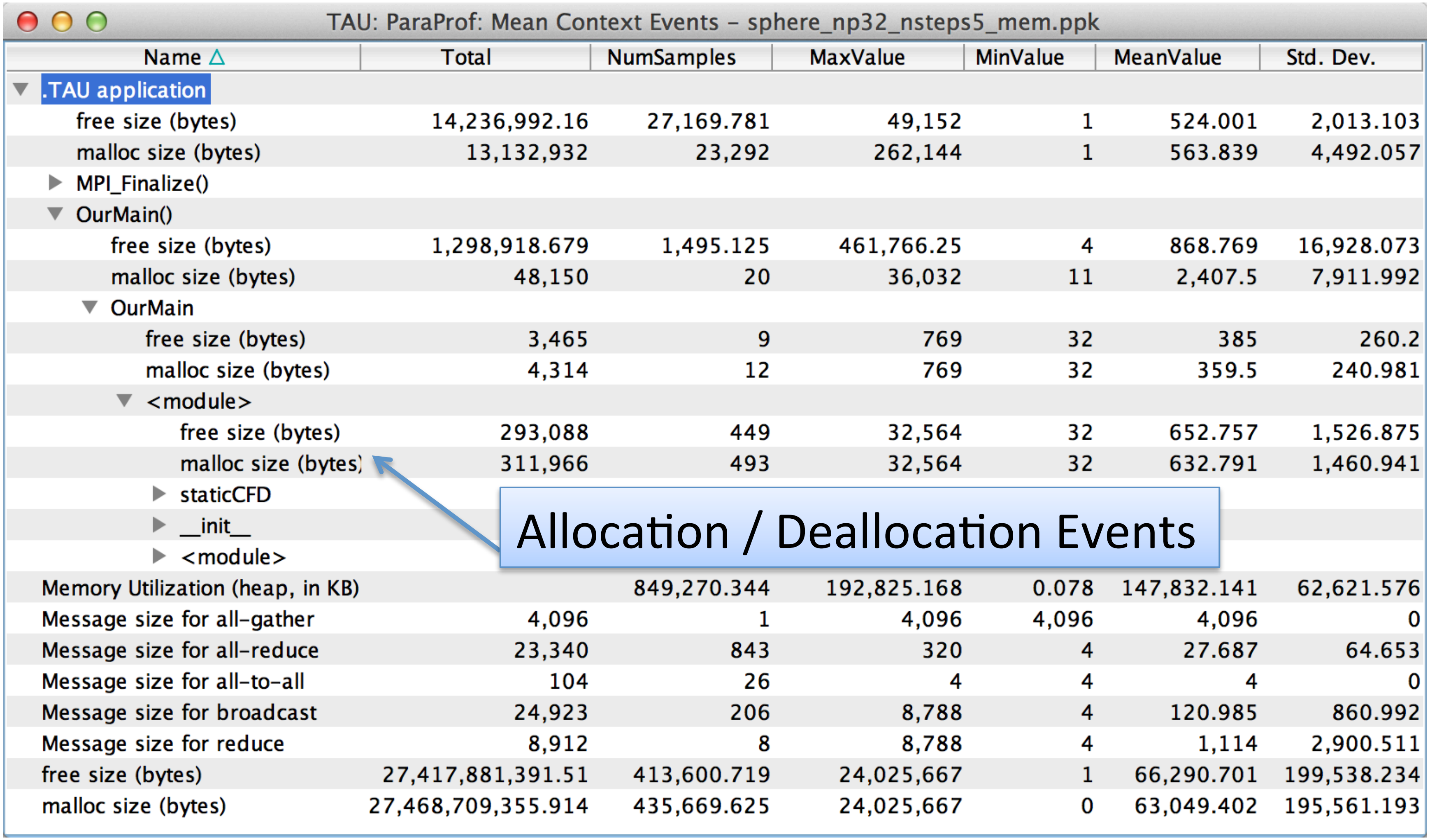
TAU: ParaProf: Mean Context Events – sphere_np32_nsteps5_mem.ppk

Name △	Total	NumSamples	MaxValue	MinValue	MeanValue	Std. Dev.
▼ .TAU application						
free size (bytes)	14,236,992.16	27,169.781	49,152	1	524.001	2,013.103
malloc size (bytes)	13,132,932	23,292	262,144	1	563.839	4,492.057
▶ MPI_Finalize()						
▼ OurMain()						
free size (bytes)	1,298,918.679	1,495.125	461,766.25	4	868.769	16,928.073
malloc size (bytes)	48,150	20	36,032	11	2,407.5	7,911.992
▼ OurMain						
free size (bytes)	3,465	9	769	32	385	260.2
malloc size (bytes)	4,314	12	769	32	359.5	240.981
▼ <module>						
free size (bytes)	293,088	449	32,564	32	652.757	1,526.875
malloc size (bytes)	311,966	493	32,564	32	632.791	1,460.941
▶ staticCFD						
▶ __init__						
▶ <module>						
Memory Utilization (heap, in KB)		849,270.344	192,825.168	0.078	147,832.141	62,621.576
Message size for all-gather	4,096	1	4,096	4,096	4,096	0
Message size for all-reduce	23,340	843	320	4	27.687	64.653
Message size for all-to-all	104	26	4	4	4	0
Message size for broadcast	24,923	206	8,788	4	120.985	860.992
Message size for reduce	8,912	8	8,788	4	1,114	2,900.511
free size (bytes)	27,417,881,391.51	413,600.719	24,025,667	1	66,290.701	199,538.234
malloc size (bytes)	27,468,709,355.914	435,669.625	24,025,667	0	63,049.402	195,561.193

Total allocated/deallocated

`% paraprof` (Right-click label [e.g “node 0”] → Show Context Event Window)

Where is Memory Allocated / Deallocated?



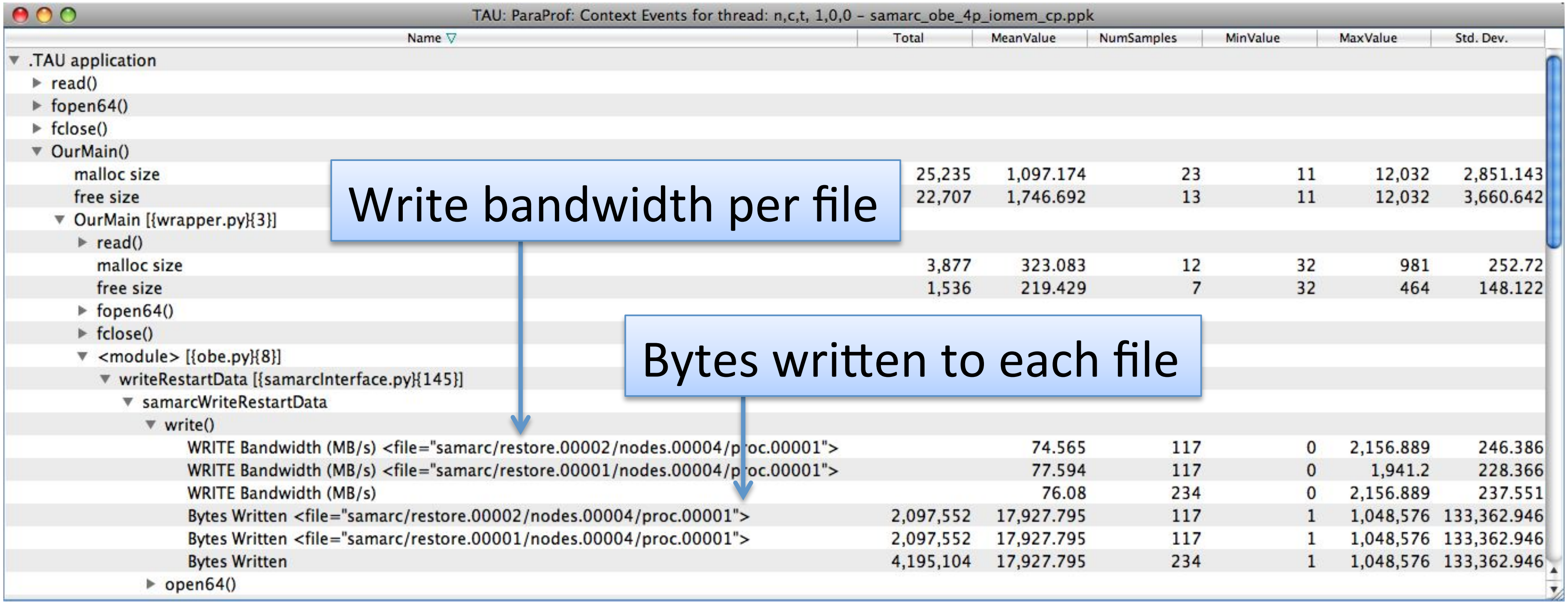
TAU: ParaProf: Mean Context Events – sphere_np32_nsteps5_mem.ppk

Name △	Total	NumSamples	MaxValue	MinValue	MeanValue	Std. Dev.
▼ .TAU application						
free size (bytes)	14,236,992.16	27,169.781	49,152	1	524.001	2,013.103
malloc size (bytes)	13,132,932	23,292	262,144	1	563.839	4,492.057
▶ MPI_Finalize()						
▼ OurMain()						
free size (bytes)	1,298,918.679	1,495.125	461,766.25	4	868.769	16,928.073
malloc size (bytes)	48,150	20	36,032	11	2,407.5	7,911.992
▼ OurMain						
free size (bytes)	3,465	9	769	32	385	260.2
malloc size (bytes)	4,314	12	769	32	359.5	240.981
▼ <module>						
free size (bytes)	293,088	449	32,564	32	652.757	1,526.875
malloc size (bytes)	311,966	493	32,564	32	632.791	1,460.941
▶ staticCFD						
▶ __init__						
▶ <module>						
Memory Utilization (heap, in KB)		849,270.344	192,825.168	0.078	147,832.141	62,621.576
Message size for all-gather	4,096	1	4,096	4,096	4,096	0
Message size for all-reduce	23,340	843	320	4	27.687	64.653
Message size for all-to-all	104	26	4	4	4	0
Message size for broadcast	24,923	206	8,788	4	120.985	860.992
Message size for reduce	8,912	8	8,788	4	1,114	2,900.511
free size (bytes)	27,417,881,391.51	413,600.719	24,025,667	1	66,290.701	199,538.234
malloc size (bytes)	27,468,709,355.914	435,669.625	24,025,667	0	63,049.402	195,561.193

Allocation / Deallocation Events

`% paraprof` (Right-click label [e.g "node 0"] → Show Context Event Window)

What are the I/O Characteristics?



TAU: ParaProf: Context Events for thread: n,c,t, 1,0,0 – samarc_obe_4p_iomem_cp.ppk

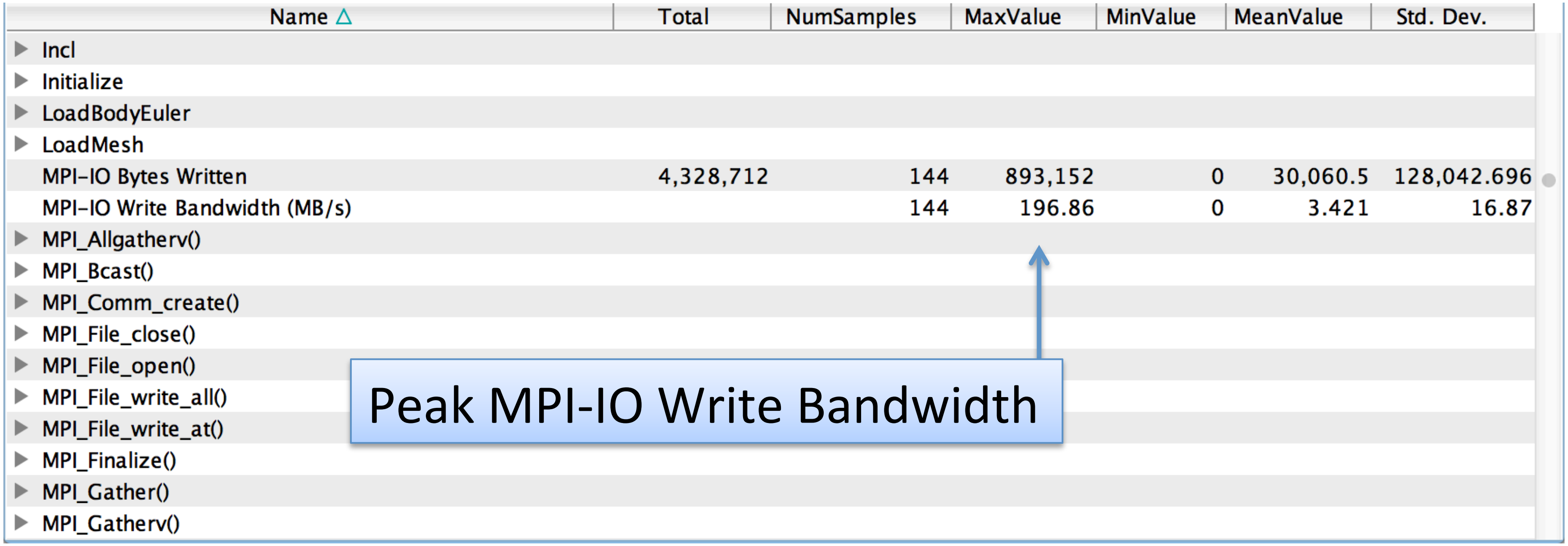
Name ▽	Total	MeanValue	NumSamples	MinValue	MaxValue	Std. Dev.
▼ .TAU application						
▶ read()						
▶ fopen64()						
▶ fclose()						
▼ OurMain()						
malloc size	25,235	1,097.174	23	11	12,032	2,851.143
free size	22,707	1,746.692	13	11	12,032	3,660.642
▼ OurMain [{wrapper.py}{3}]						
▶ read()						
malloc size	3,877	323.083	12	32	981	252.72
free size	1,536	219.429	7	32	464	148.122
▶ fopen64()						
▶ fclose()						
▼ <module> [{obe.py}{8}]						
▼ writeRestartData [{samarcInterface.py}{145}]						
▼ samarcWriteRestartData						
▼ write()						
WRITE Bandwidth (MB/s) <file="samarc/restore.00002/nodes.00004/proc.00001">		74.565	117	0	2,156.889	246.386
WRITE Bandwidth (MB/s) <file="samarc/restore.00001/nodes.00004/proc.00001">		77.594	117	0	1,941.2	228.366
WRITE Bandwidth (MB/s)		76.08	234	0	2,156.889	237.551
Bytes Written <file="samarc/restore.00002/nodes.00004/proc.00001">	2,097,552	17,927.795	117	1	1,048,576	133,362.946
Bytes Written <file="samarc/restore.00001/nodes.00004/proc.00001">	2,097,552	17,927.795	117	1	1,048,576	133,362.946
Bytes Written	4,195,104	17,927.795	234	1	1,048,576	133,362.946
▶ open64()						

Write bandwidth per file

Bytes written to each file

% `paraprof` (Right-click label [e.g “node 0”] → Show Context Event Window)

What are the I/O Characteristics?

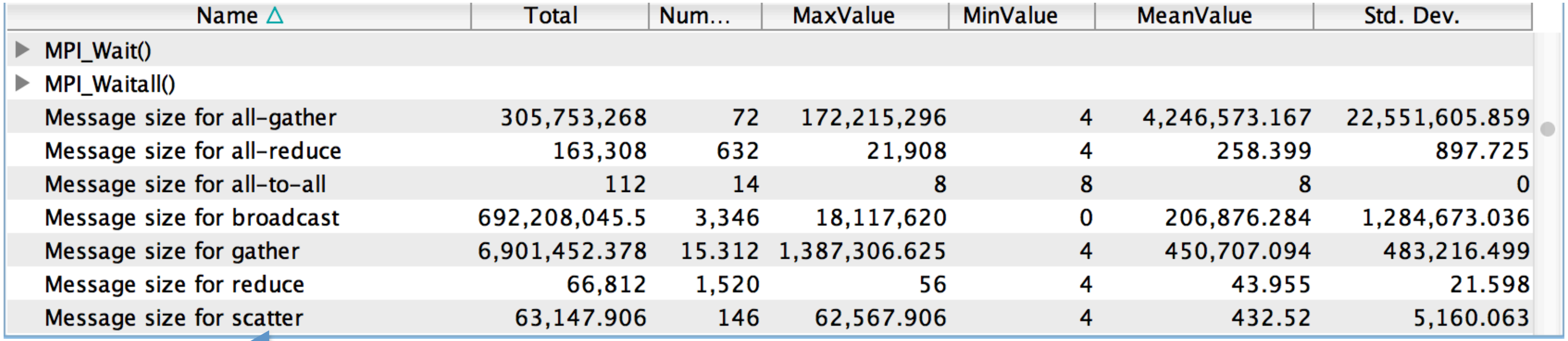


Name △	Total	NumSamples	MaxValue	MinValue	MeanValue	Std. Dev.
▶ Incl						
▶ Initialize						
▶ LoadBodyEuler						
▶ LoadMesh						
MPI–IO Bytes Written	4,328,712	144	893,152	0	30,060.5	128,042.696
MPI–IO Write Bandwidth (MB/s)		144	196.86	0	3.421	16.87
▶ MPI_Allgatherv()						
▶ MPI_Bcast()						
▶ MPI_Comm_create()						
▶ MPI_File_close()						
▶ MPI_File_open()						
▶ MPI_File_write_all()						
▶ MPI_File_write_at()						
▶ MPI_Finalize()						
▶ MPI_Gather()						
▶ MPI_Gatherv()						

Peak MPI-IO Write Bandwidth

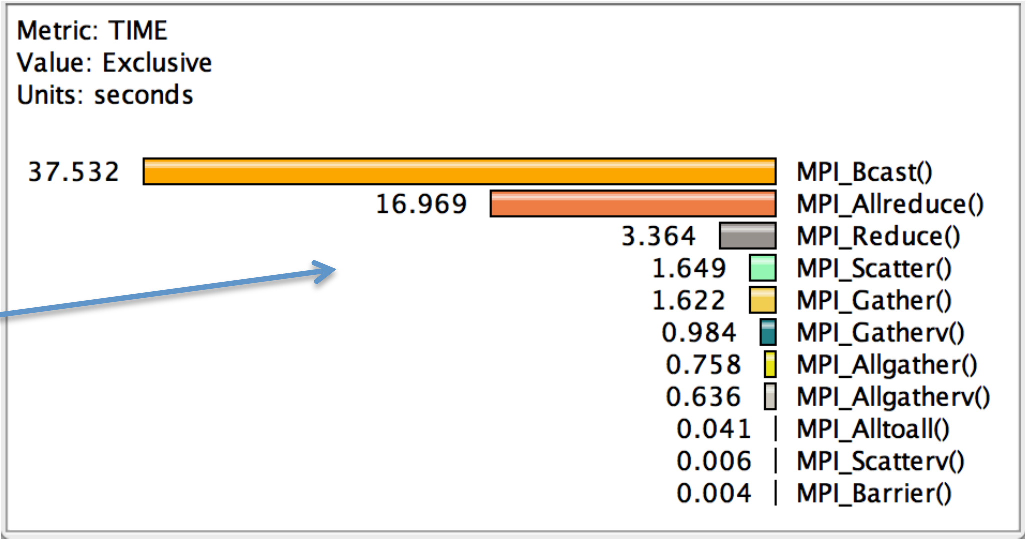
% **`paraprof`** **(Right-click label [e.g "node 0"] → Show Context Event Window)**

How Much Time is spent in Collectives?



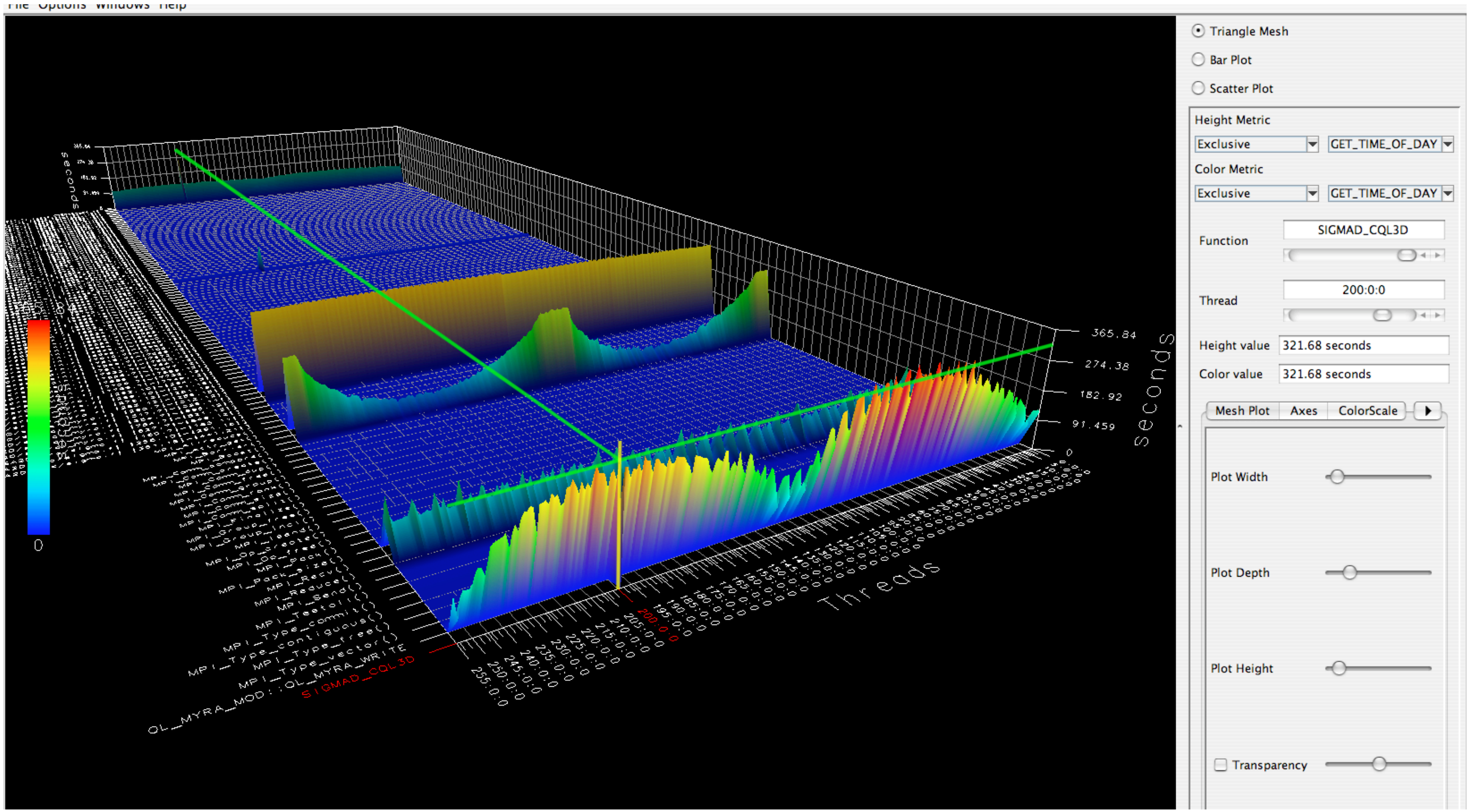
Name △	Total	Num...	MaxValue	MinValue	MeanValue	Std. Dev.
▶ MPI_Wait()						
▶ MPI_Waitall()						
Message size for all-gather	305,753,268	72	172,215,296	4	4,246,573.167	22,551,605.859
Message size for all-reduce	163,308	632	21,908	4	258.399	897.725
Message size for all-to-all	112	14	8	8	8	0
Message size for broadcast	692,208,045.5	3,346	18,117,620	0	206,876.284	1,284,673.036
Message size for gather	6,901,452.378	15.312	1,387,306.625	4	450,707.094	483,216.499
Message size for reduce	66,812	1,520	56	4	43.955	21.598
Message size for scatter	63,147.906	146	62,567.906	4	432.52	5,160.063

Message sizes



Time spent in collectives

3D Profile Visualization



● Triangle Mesh

○ Bar Plot

○ Scatter Plot

Height Metric

Exclusive | GET_TIME_OF_DAY

Color Metric

Exclusive | GET_TIME_OF_DAY

Function: SIGMAD_CQL3D

Thread: 200:0:0

Height value: 321.68 seconds

Color value: 321.68 seconds

Plot Width

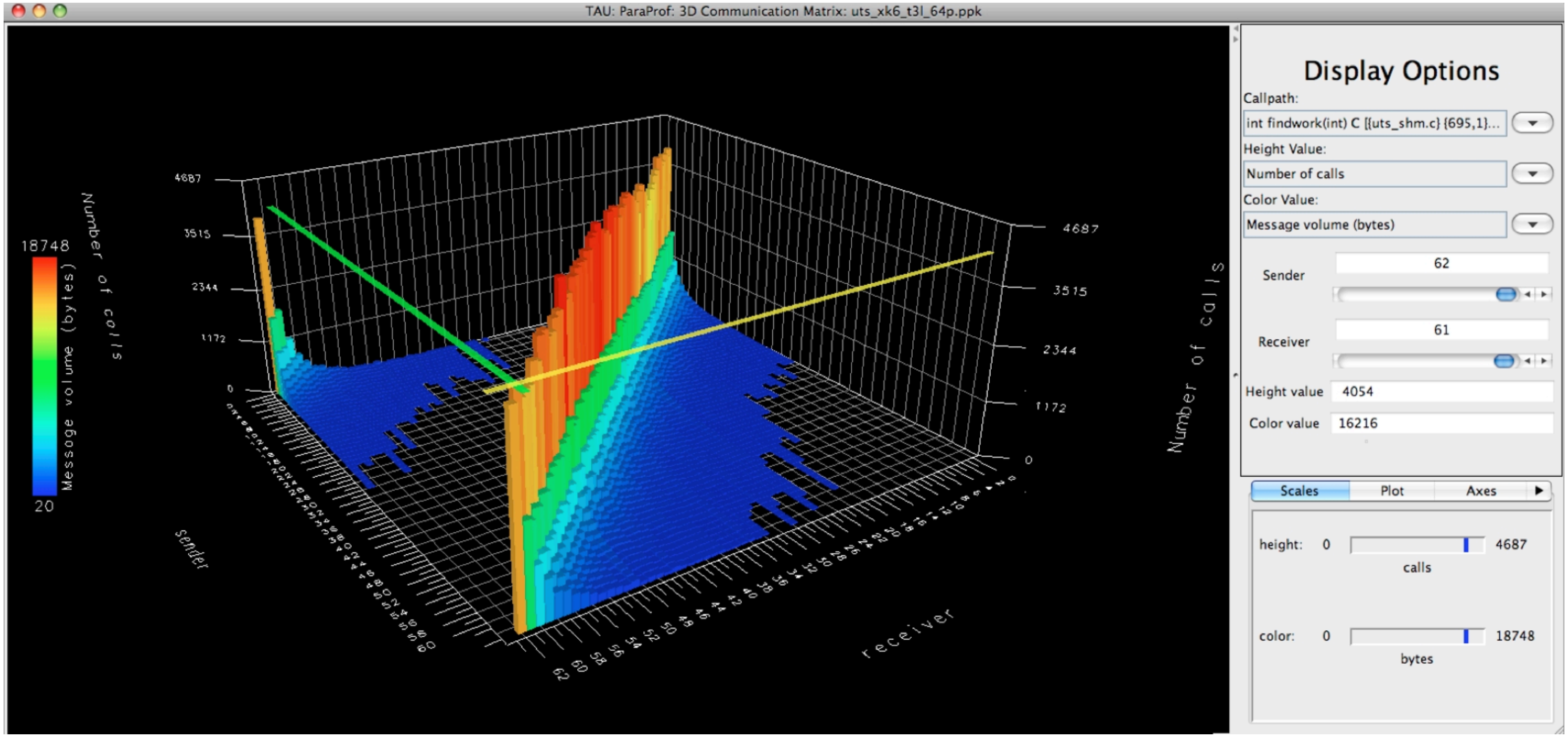
Plot Depth

Plot Height

□ Transparency

`% paraprof` (Windows → 3D Visualization)

3D Communication Visualization



TAU: ParaProf: 3D Communication Matrix: uts_xk6_t3l_64p.ppk

Display Options

Callpath: int findwork(int) C [{uts_shm.c} {695,1}...

Height Value: Number of calls

Color Value: Message volume (bytes)

Sender: 62

Receiver: 61

Height value: 4054

Color value: 16216

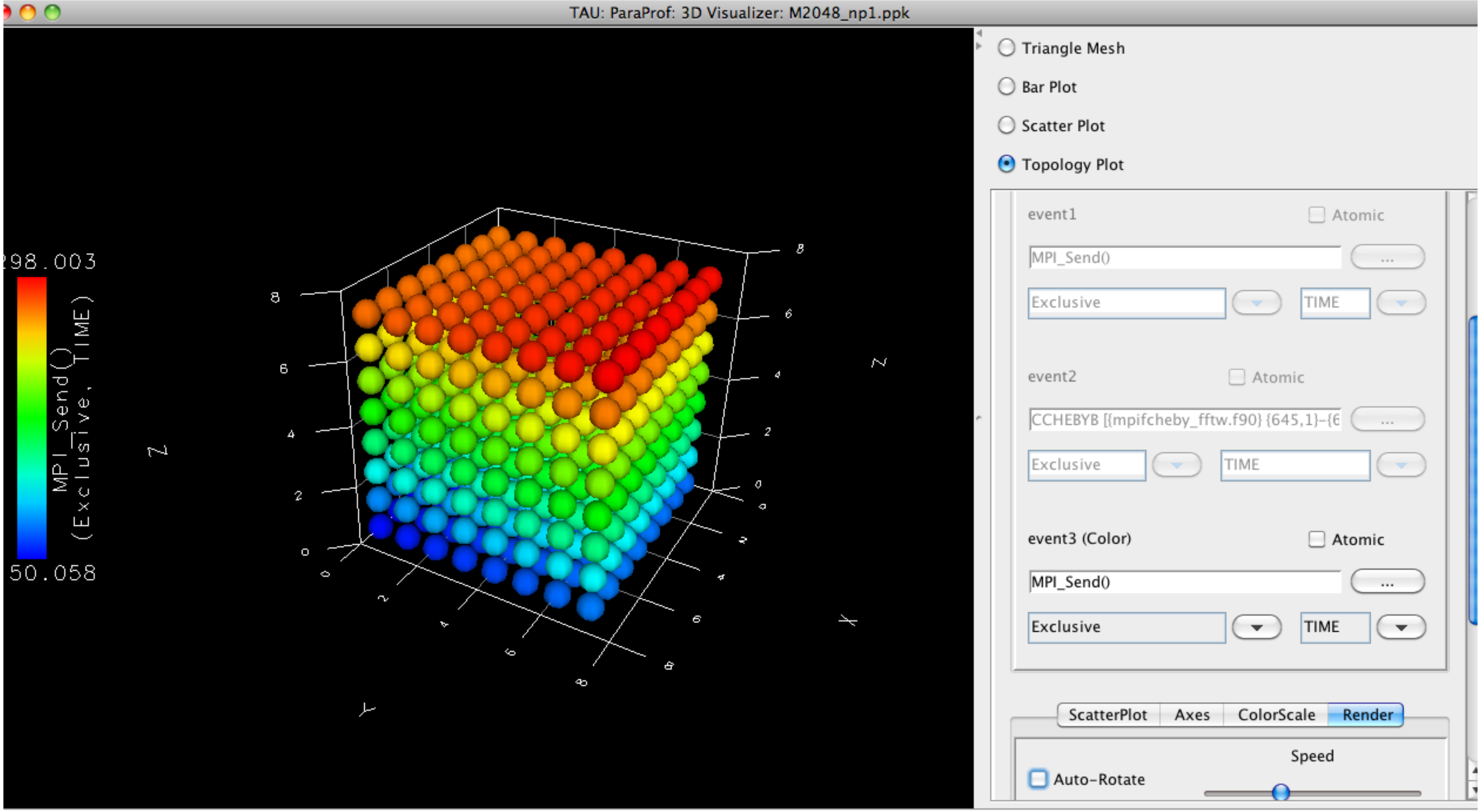
Scales | Plot | Axes

height: 0 – 4687 calls

color: 0 – 18748 bytes

```
% qsub –env TAU_COMM_MATRIX=1 …
% paraprof (Windows → 3D Communication Matrix)
```

3D Topology Visualization



TAU: ParaProf: 3D Visualizer: M2048_np1.ppk

○ Triangle Mesh

○ Bar Plot

○ Scatter Plot

● Topology Plot

event1 □ Atomic

MPI_Send()

Exclusive TIME

event2 □ Atomic

CCHEBYB [{mpifcheby_fftw.f90} {645,1}-{6

Exclusive TIME

event3 (Color) □ Atomic

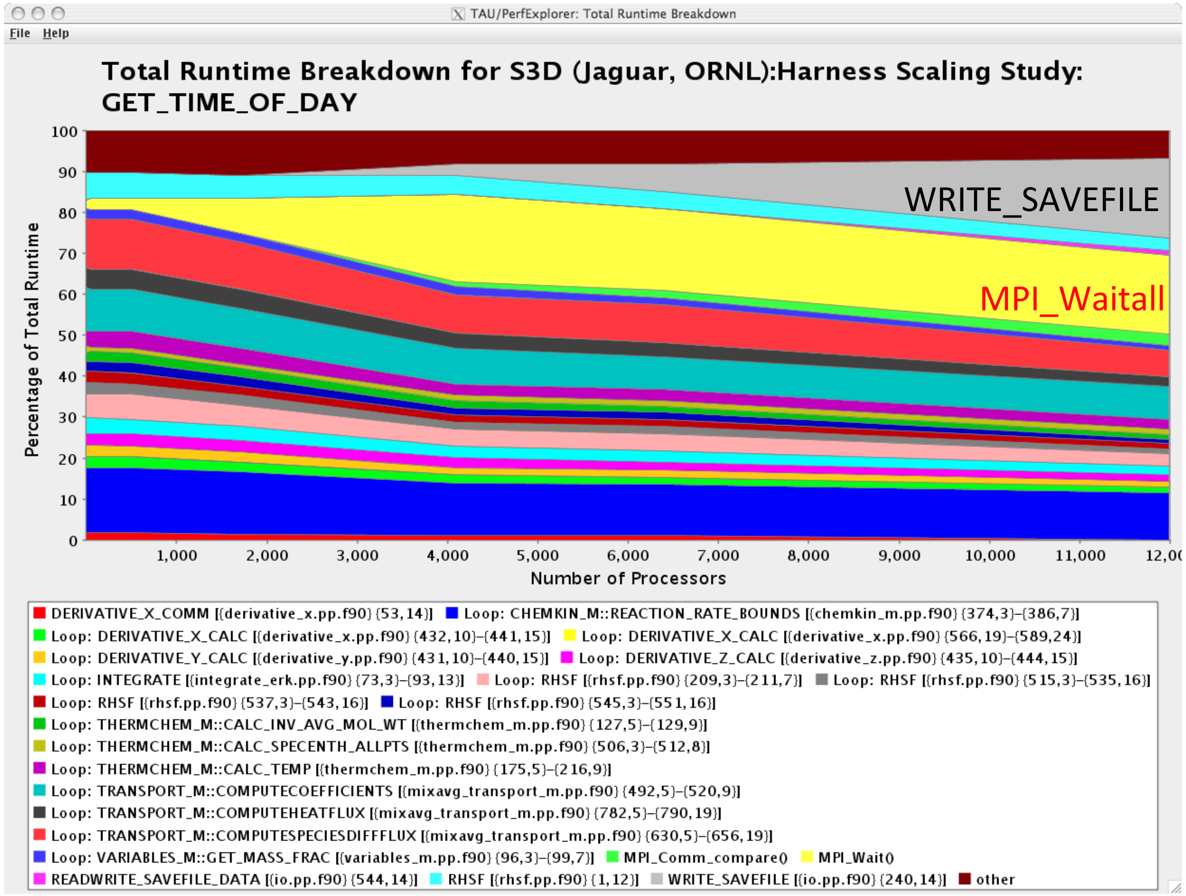
MPI_Send()

Exclusive TIME

□ Auto-Rotate

```
% paraprof  (Windows → 3D Visualization → Topology Plot)
```

How Does Each Routine Scale?

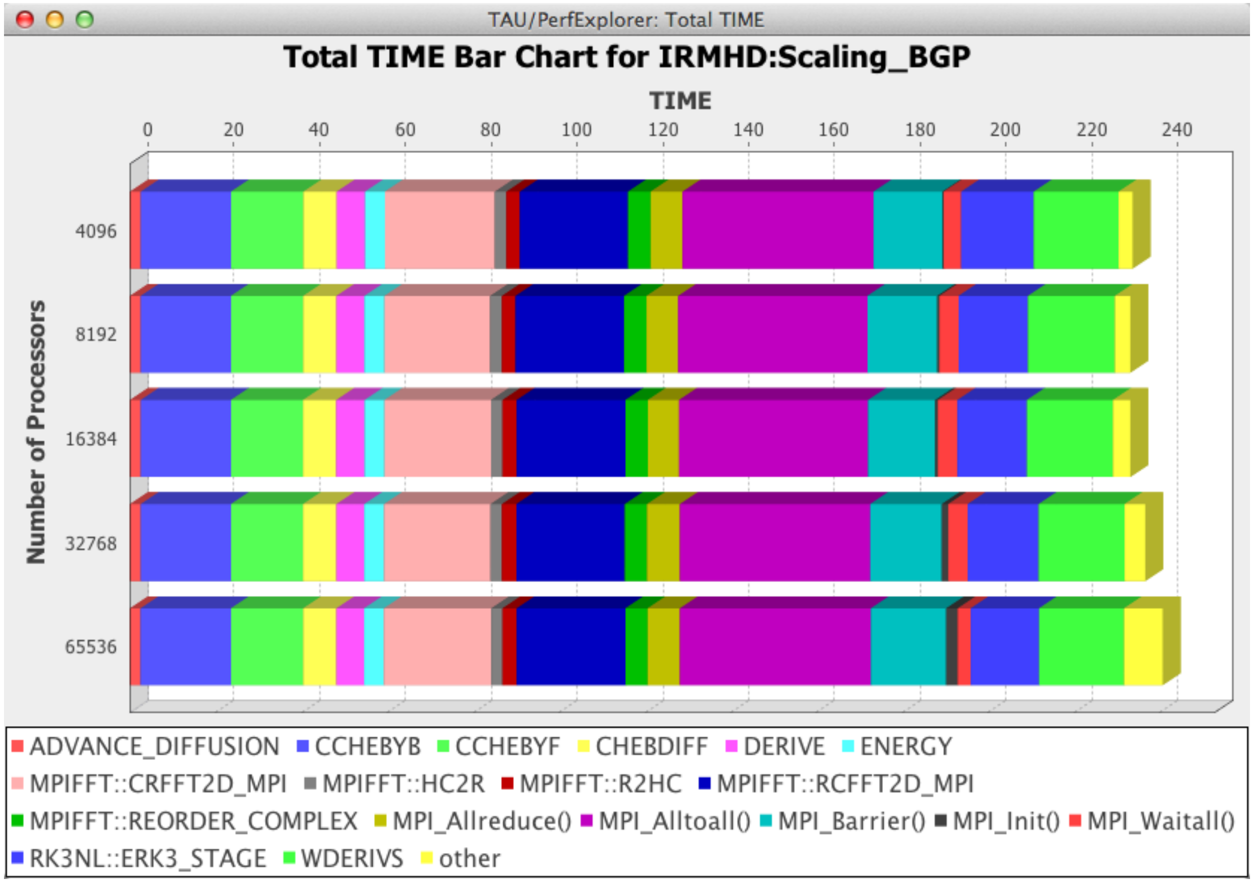


WRITE_SAVEFILE

MPI_Waitall

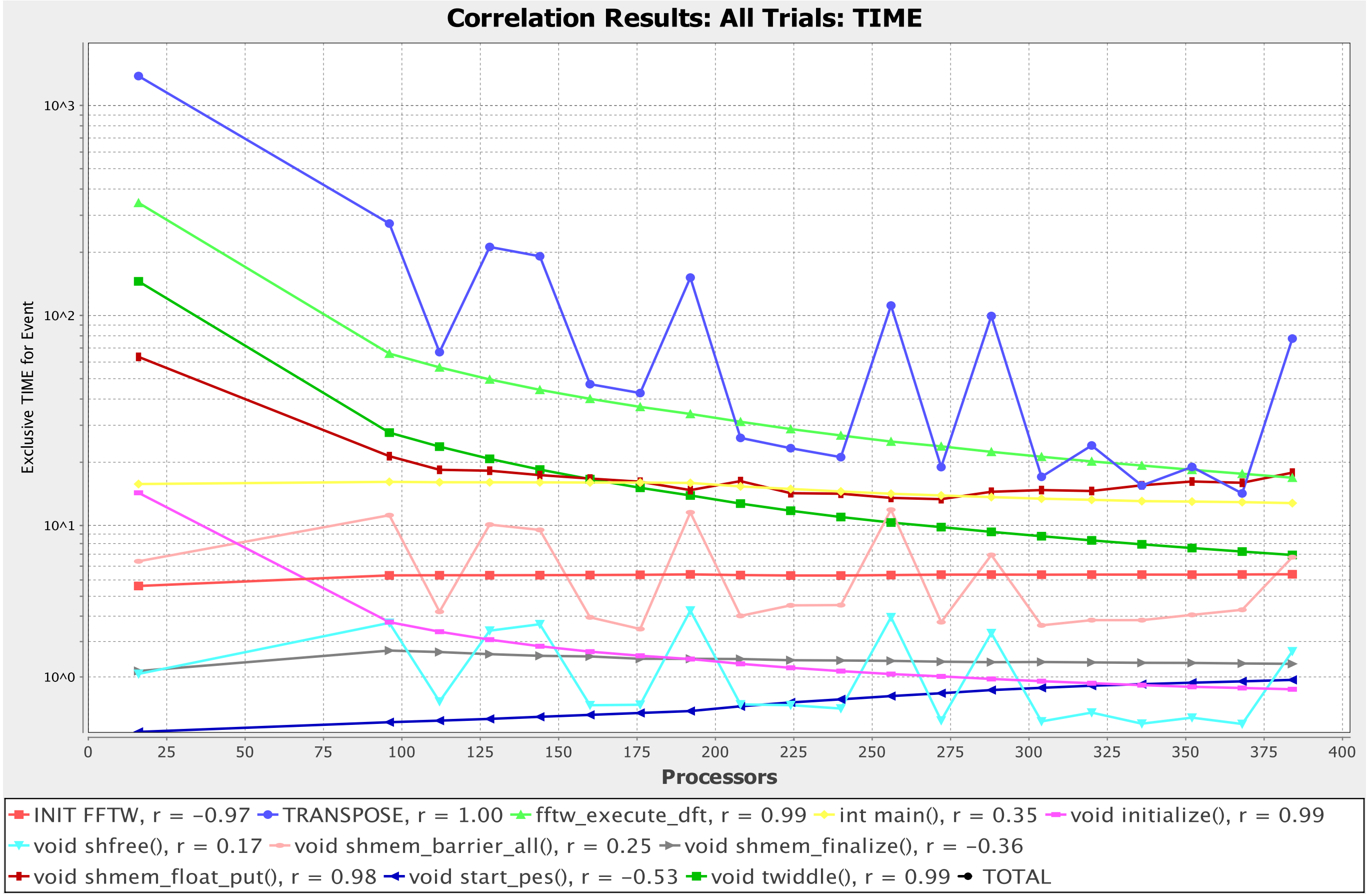
`% perfexplorer` **(Charts → Runtime Breakdown)**

How Does Each Routine Scale?



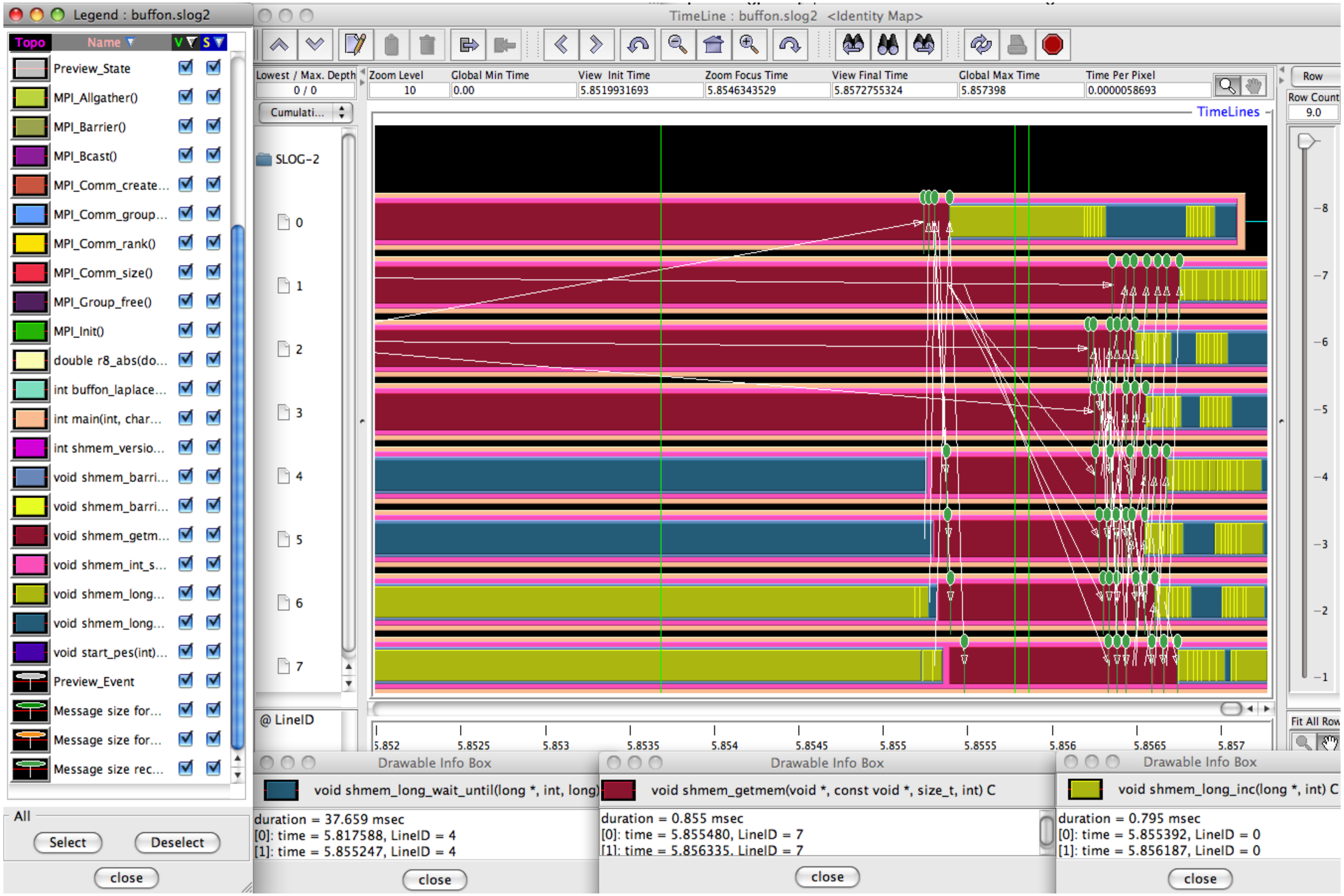
`% perfexplorer` (Charts → Stacked Bar Chart)

Which Events Correlate with Runtime?



`% perfexplorer` (Charts → Correlate Events with Total Runtime)

When do Events Occur?



Legend : buffon.slog2

Topo	Name	V	S
	Preview_State	☑	☑
	MPI_Allgather()	☑	☑
	MPI_Barrier()	☑	☑
	MPI_Bcast()	☑	☑
	MPI_Comm_create...	☑	☑
	MPI_Comm_group...	☑	☑
	MPI_Comm_rank()	☑	☑
	MPI_Comm_size()	☑	☑
	MPI_Group_free()	☑	☑
	MPI_Init()	☑	☑
	double r8_abs(do...	☑	☑
	int buffon_laplace...	☑	☑
	int main(int, char...	☑	☑
	int shmem_versio...	☑	☑
	void shmem_barri...	☑	☑
	void shmem_barri...	☑	☑
	void shmem_getm...	☑	☑
	void shmem_int_s...	☑	☑
	void shmem_long...	☑	☑
	void shmem_long...	☑	☑
	void start_pes(int)...	☑	☑
	Preview_Event	☑	☑
	Message size for...	☑	☑
	Message size for...	☑	☑
	Message size rec...	☑	☑

TimeLine : buffon.slog2 <Identity Map>

Lowest / Max. Depth	Zoom Level	Global Min Time	View Init Time	Zoom Focus Time	View Final Time	Global Max Time	Time Per Pixel
0 / 0	10	0.00	5.8519931693	5.8546343529	5.8572755324	5.857398	0.0000058693

Row Count: 9.0

Drawable Info Box

void shmem_long_wait_until(long *, int, long)

duration = 37.659 msec
[0]: time = 5.817588, LineID = 4
[1]: time = 5.855247, LineID = 4

Drawable Info Box

void shmem_getmem(void *, const void *, size_t, int) C

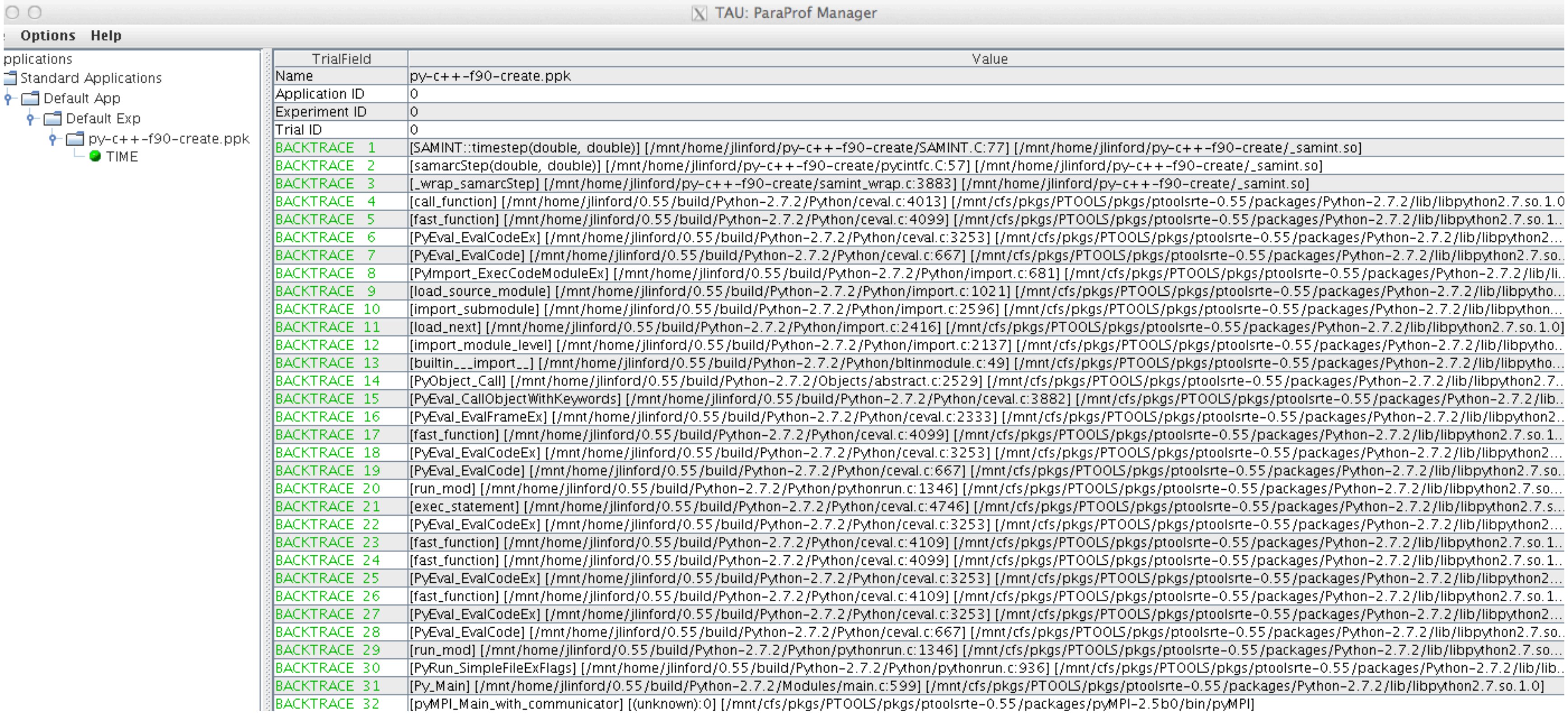
duration = 0.855 msec
[0]: time = 5.855480, LineID = 7
[1]: time = 5.856335, LineID = 7

Drawable Info Box

void shmem_long_inc(long *, int) C

duration = 0.795 msec
[0]: time = 5.855392, LineID = 0
[1]: time = 5.856187, LineID = 0

What Caused My Application to Crash?

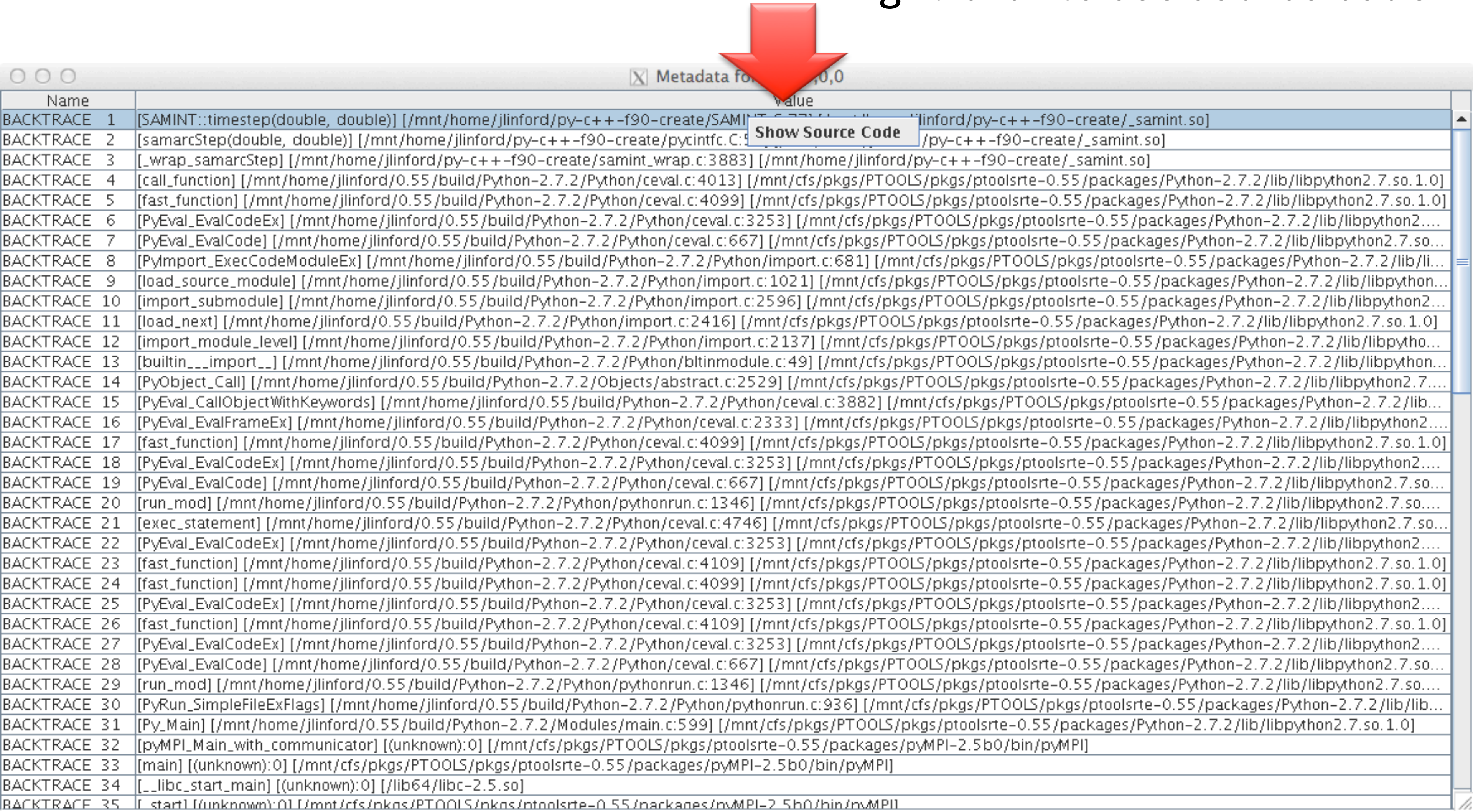


TrialField	Value
Name	py-c++-f90-create.ppk
Application ID	0
Experiment ID	0
Trial ID	0
BACKTRACE 1	[SAMINT::timestep(double, double)] [/mnt/home/jlinford/py-c++-f90-create/SAMINT.C:77] [/mnt/home/jlinford/py-c++-f90-create/_samint.so]
BACKTRACE 2	[samarcStep(double, double)] [/mnt/home/jlinford/py-c++-f90-create/pycintfc.C:57] [/mnt/home/jlinford/py-c++-f90-create/_samint.so]
BACKTRACE 3	[_wrap_samarcStep] [/mnt/home/jlinford/py-c++-f90-create/samint_wrap.c:3883] [/mnt/home/jlinford/py-c++-f90-create/_samint.so]
BACKTRACE 4	[call_function] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:4013] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so.1.0
BACKTRACE 5	[fast_function] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:4099] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so.1..
BACKTRACE 6	[PyEval_EvalCodeEx] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:3253] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2...
BACKTRACE 7	[PyEval_EvalCode] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:667] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so..
BACKTRACE 8	[PyImport_ExecCodeModuleEx] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/import.c:681] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/li..
BACKTRACE 9	[load_source_module] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/import.c:1021] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpytho...
BACKTRACE 10	[import_submodule] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/import.c:2596] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython...
BACKTRACE 11	[load_next] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/import.c:2416] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 12	[import_module_level] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/import.c:2137] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpytho..
BACKTRACE 13	[builtin___import__] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/bltinmodule.c:49] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpytho...
BACKTRACE 14	[PyObject_Call] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Objects/abstract.c:2529] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7..
BACKTRACE 15	[PyEval_CallObjectWithKeywords] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:3882] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib..
BACKTRACE 16	[PyEval_EvalFrameEx] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:2333] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2..
BACKTRACE 17	[fast_function] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:4099] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so.1..
BACKTRACE 18	[PyEval_EvalCodeEx] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:3253] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2...
BACKTRACE 19	[PyEval_EvalCode] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:667] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so..
BACKTRACE 20	[run_mod] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/pythonrun.c:1346] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so...
BACKTRACE 21	[exec_statement] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:4746] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.s...
BACKTRACE 22	[PyEval_EvalCodeEx] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:3253] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2...
BACKTRACE 23	[fast_function] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:4109] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so.1..
BACKTRACE 24	[fast_function] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:4099] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so.1..
BACKTRACE 25	[PyEval_EvalCodeEx] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:3253] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2...
BACKTRACE 26	[fast_function] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:4109] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so.1..
BACKTRACE 27	[PyEval_EvalCodeEx] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:3253] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2...
BACKTRACE 28	[PyEval_EvalCode] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:667] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so..
BACKTRACE 29	[run_mod] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/pythonrun.c:1346] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so...
BACKTRACE 30	[PyRun_SimpleFileExFlags] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/pythonrun.c:936] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/lib..
BACKTRACE 31	[Py_Main] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Modules/main.c:599] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 32	[pyMPI_Main_with_communicator] [(unknown):0] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/pyMPI-2.5b0/bin/pyMPI]

```
% export TAU_TRACK_SIGNALS=1
% paraprof
```

What Caused My Application to Crash?

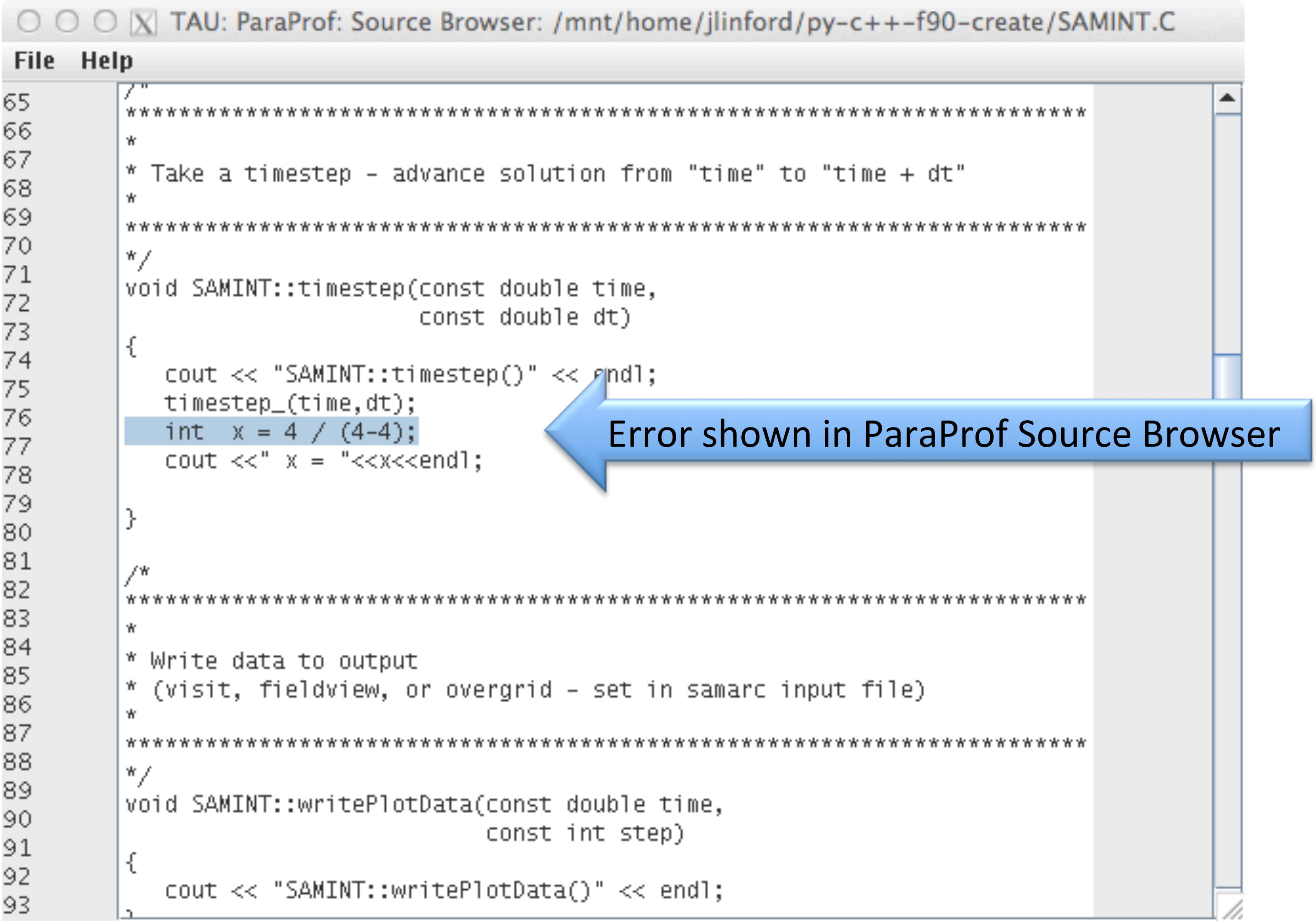
Right-click to see source code



Name	Value
BACKTRACE 1	[SAMINT::timestep(double, double)] [/mnt/home/jlinford/py-c++-f90-create/SAMI linford/py-c++-f90-create/_samint.so]
BACKTRACE 2	[samarcStep(double, double)] [/mnt/home/jlinford/py-c++-f90-create/pycintfc.C:5 /py-c++-f90-create/_samint.so]
BACKTRACE 3	[_wrap_samarcStep] [/mnt/home/jlinford/py-c++-f90-create/samint_wrap.c:3883] [/mnt/home/jlinford/py-c++-f90-create/_samint.so]
BACKTRACE 4	[call_function] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:4013] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 5	[fast_function] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:4099] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 6	[PyEval_EvalCodeEx] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:3253] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2....
BACKTRACE 7	[PyEval_EvalCode] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:667] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so...
BACKTRACE 8	[PyImport_ExecCodeModuleEx] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/import.c:681] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/li...
BACKTRACE 9	[load_source_module] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/import.c:1021] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython...
BACKTRACE 10	[import_submodule] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/import.c:2596] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2...
BACKTRACE 11	[load_next] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/import.c:2416] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 12	[import_module_level] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/import.c:2137] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpytho...
BACKTRACE 13	[builtin___import__] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/bltinmodule.c:49] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython...
BACKTRACE 14	[PyObject_Call] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Objects/abstract.c:2529] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7....
BACKTRACE 15	[PyEval_CallObjectWithKeywords] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:3882] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib...
BACKTRACE 16	[PyEval_EvalFrameEx] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:2333] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2....
BACKTRACE 17	[fast_function] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:4099] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 18	[PyEval_EvalCodeEx] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:3253] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2....
BACKTRACE 19	[PyEval_EvalCode] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:667] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so...
BACKTRACE 20	[run_mod] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/pythonrun.c:1346] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so....
BACKTRACE 21	[exec_statement] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:4746] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so...
BACKTRACE 22	[PyEval_EvalCodeEx] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:3253] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2....
BACKTRACE 23	[fast_function] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:4109] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 24	[fast_function] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:4099] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 25	[PyEval_EvalCodeEx] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:3253] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2....
BACKTRACE 26	[fast_function] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:4109] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 27	[PyEval_EvalCodeEx] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:3253] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2....
BACKTRACE 28	[PyEval_EvalCode] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/ceval.c:667] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so...
BACKTRACE 29	[run_mod] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/pythonrun.c:1346] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so....
BACKTRACE 30	[PyRun_SimpleFileExFlags] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Python/pythonrun.c:936] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/lib...
BACKTRACE 31	[Py_Main] [/mnt/home/jlinford/0.55/build/Python-2.7.2/Modules/main.c:599] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/Python-2.7.2/lib/libpython2.7.so.1.0]
BACKTRACE 32	[pyMPI_Main_with_communicator] [(unknown):0] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/pyMPI-2.5b0/bin/pyMPI]
BACKTRACE 33	[main] [(unknown):0] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/pyMPI-2.5b0/bin/pyMPI]
BACKTRACE 34	[__libc_start_main] [(unknown):0] [/lib64/libc-2.5.so]
BACKTRACE 35	[_start] [(unknown):0] [/mnt/cfs/pkgs/PTOOLS/pkgs/ptoolsrte-0.55/packages/pyMPI-2.5b0/bin/pyMPI]

ParaTools

What Caused My Application to Crash?


TAU: ParaProf: Source Browser: /mnt/home/jlinford/py-c++-f90-create/SAMINT.C
File
Help

```
/*
************************************************************************
*
* Take a timestep - advance solution from "time" to "time + dt"
*
************************************************************************
*/
void SAMINT::timestep(const double time,
                      const double dt)
{
   cout << "SAMINT::timestep()" << endl;
   timestep_(time,dt);
   int  x = 4 / (4-4);
   cout <<" x = "<<x<<endl;

}

/*
************************************************************************
*
* Write data to output
* (visit, fieldview, or overgrid - set in samarc input file)
*
************************************************************************
*/
void SAMINT::writePlotData(const double time,
                           const int step)
{
   cout << "SAMINT::writePlotData()" << endl;
}
```

Error shown in ParaProf Source Browser

Python Performance Evaluation

HANDS-ON

ParaTools Training Cluster

```
ssh -XY livetau@cerberus.nic.uoregon.edu
```

Password: ***********

Pick a number **XX** from [1, 39]

```
cd studentXX
tar xvzf ~/workshop-python.tgz
```

Training materials

- ~livetau/workshop-python.tgz
- https://github.com/jlinford/workshop-python
- http://www.paratools.com/emit15/TAU

Getting Started with TAU

- Each configuration of TAU corresponds to a unique stub makefile (*TAU_MAKEFILE*) in the TAU installation directory

```
% ls $TAU/Makefile.*
Makefile.tau-icpc-papi-mpi-pdt
Makefile.tau-icpc-papi-ompt-mpi-pdt-openmp
Makefile.tau-icpc-papi-ompt-pdt-openmp
…
Makefile.tau-mpi-pthread-python-pdt
Makefile.tau-mpi-python-pdt
Makefile.tau-mpi-python-pdt-openmp
Makefile.tau-pthread-python-pdt
Makefile.tau-python-pdt
```

`19 TAU Makefiles on cerberus.nic.uoregon.edu`

Basic TAU Workflow

1. Choose your `TAU_MAKEFILE`:

```
$ export TAU_MAKEFILE=
          $TAU/Makefile.tau-mpi-python-pdt
```

2. Use `tau_f90.sh`, `tau_cxx.sh`, etc. as compiler:

```
$ mpif90 foo.f90
```

   changes to

```
$ tau_f90.sh foo.f90
```

3. Edit Makefile or set compilers on command line:

```
$ make CC=tau_cc.sh
```

4. Execute application

5. Analyze performance data:

`pprof`	(for text based profile display)
`paraprof`	(for GUI)

TAU with Pure Python

```
$ cd workshop-python/00_matmult.py
$ python mm.py
```

Run with tau_python to generate profiles:

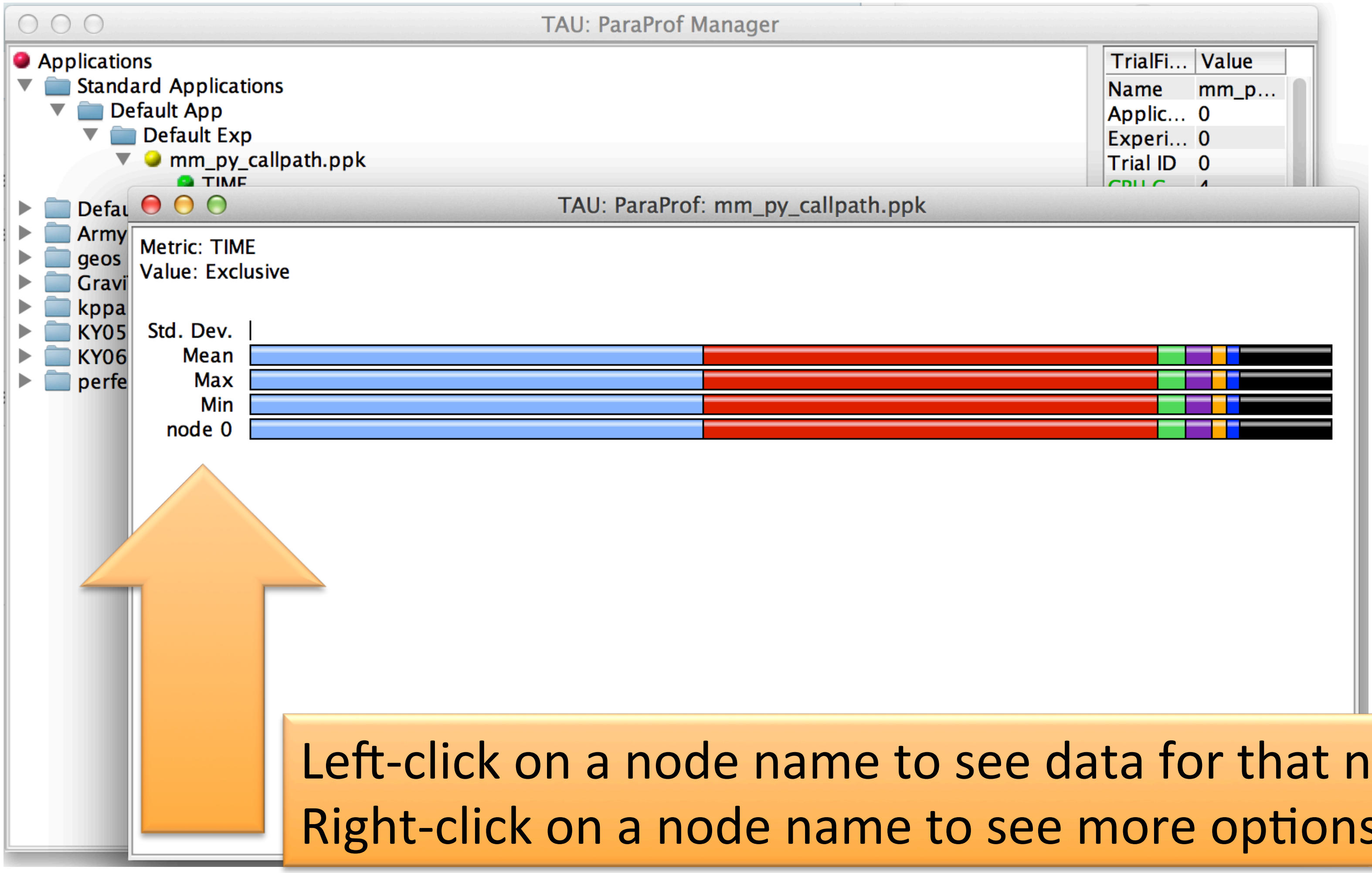
```
$ tau_python mm.py
$ ls profile.*              # shows profile.0.0.0
$ paraprof --pack mm_py_flat.ppk
```

View the profiles:

```
$ pprof -a | less           #Command line
$ paraprof                  #GUI (Java, X11)
```

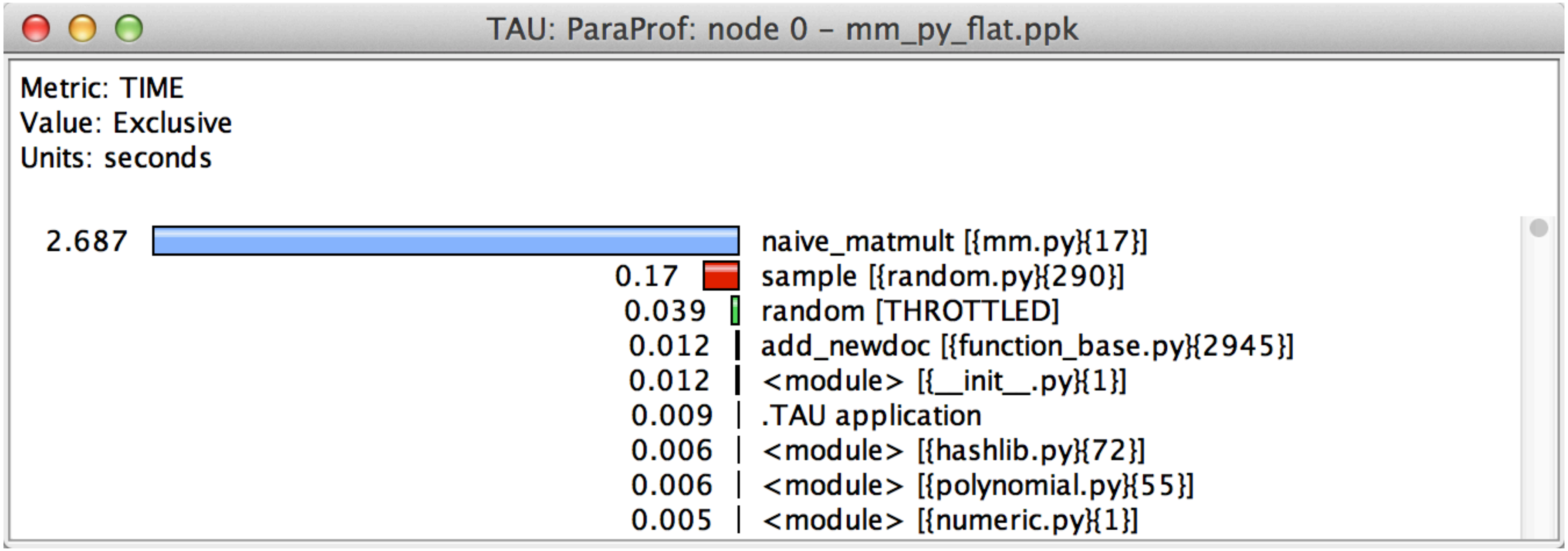
ParaProf Profile Visualizer

```
$ paraprof 00_matmult.py/analysis/mm_py_flat.ppk
```



Left-click on a node name to see data for that node
Right-click on a node name to see more options

Exclusive Time in ParaProf

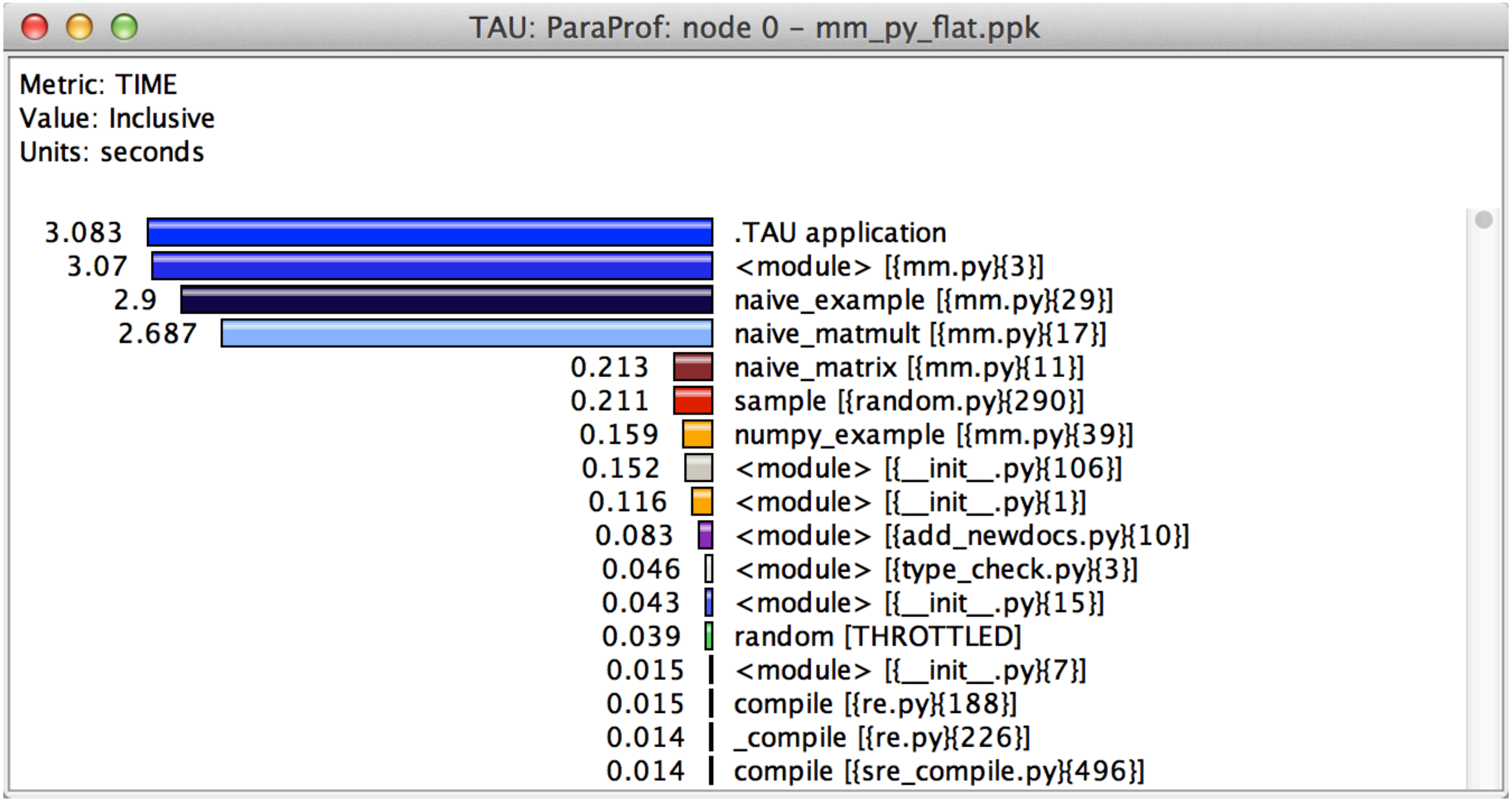


```
Metric: TIME
Value: Exclusive
Units: seconds

2.687  naive_matmult [{mm.py}{17}]
0.17   sample [{random.py}{290}]
0.039  random [THROTTLED]
0.012  add_newdoc [{function_base.py}{2945}]
0.012  <module> [{__init__.py}{1}]
0.009  .TAU application
0.006  <module> [{hashlib.py}{72}]
0.006  <module> [{polynomial.py}{55}]
0.005  <module> [{numeric.py}{1}]
```

```
$ paraprof 00_matmult.py/analysis/mm_py_flat.ppk
```

Inclusive Time in ParaProf



TAU: ParaProf: node 0 – mm_py_flat.ppk

Metric: TIME
Value: Inclusive
Units: seconds

3.083 .TAU application
3.07 <module> [{mm.py}{3}]
2.9 naive_example [{mm.py}{29}]
2.687 naive_matmult [{mm.py}{17}]
0.213 naive_matrix [{mm.py}{11}]
0.211 sample [{random.py}{290}]
0.159 numpy_example [{mm.py}{39}]
0.152 <module> [{__init__.py}{106}]
0.116 <module> [{__init__.py}{1}]
0.083 <module> [{add_newdocs.py}{10}]
0.046 <module> [{type_check.py}{3}]
0.043 <module> [{__init__.py}{15}]
0.039 random [THROTTLED]
0.015 <module> [{__init__.py}{7}]
0.015 compile [{re.py}{188}]
0.014 _compile [{re.py}{226}]
0.014 compile [{sre_compile.py}{496}]

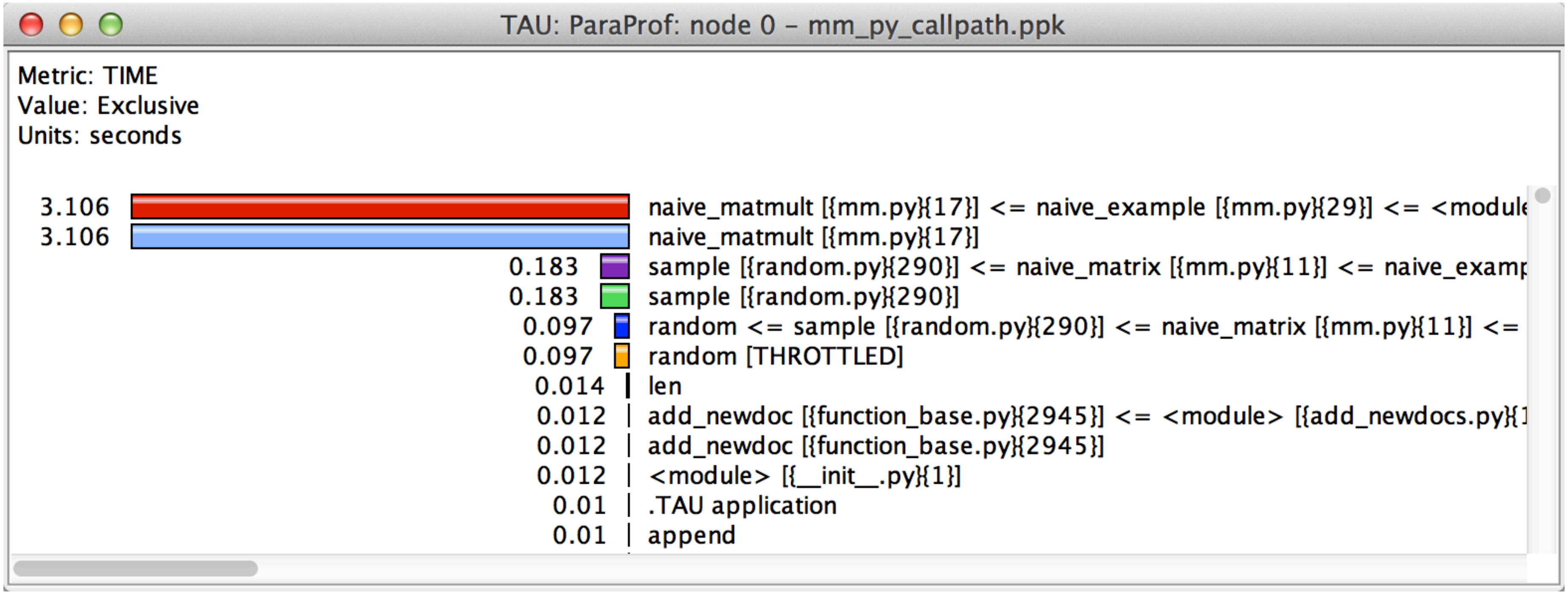
Callpath Profiles with Pure Python

For callpath profiles:

```
$ export TAU_CALLPATH=1
$ export TAU_CALLPATH_DEPTH=10
$ tau_python mm.py
```

TAU_CALLPATH_DEPTH controls the depth of the recorded callpath. "10" is usually more than enough.

Callpath Profiles in ParaProf



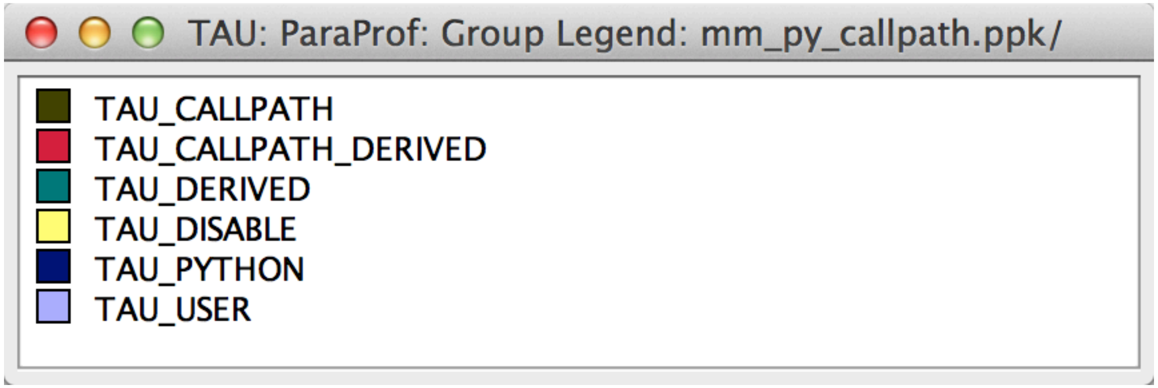
TAU: ParaProf: node 0 – mm_py_callpath.ppk

```
Metric: TIME
Value: Exclusive
Units: seconds

3.106   naive_matmult [{mm.py}{17}] <= naive_example [{mm.py}{29}] <= <module
3.106   naive_matmult [{mm.py}{17}]
0.183   sample [{random.py}{290}] <= naive_matrix [{mm.py}{11}] <= naive_examp
0.183   sample [{random.py}{290}]
0.097   random <= sample [{random.py}{290}] <= naive_matrix [{mm.py}{11}] <=
0.097   random [THROTTLED]
0.014   len
0.012   add_newdoc [{function_base.py}{2945}] <= <module> [{add_newdocs.py}{1
0.012   add_newdoc [{function_base.py}{2945}]
0.012   <module> [{__init__.py}{1}]
 0.01   .TAU application
 0.01   append
```

Windows | Group Legend →

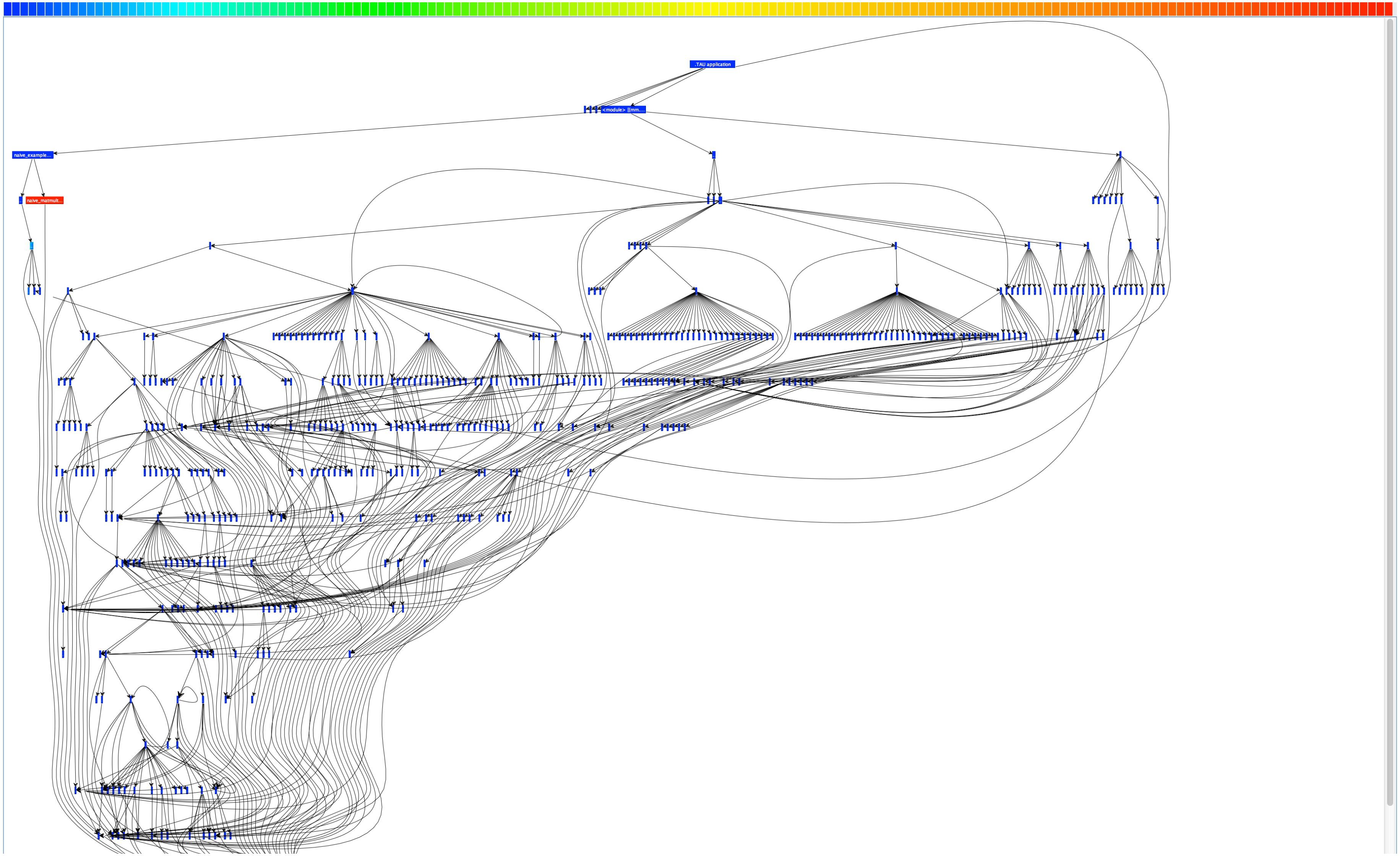
Right-click to hide groups



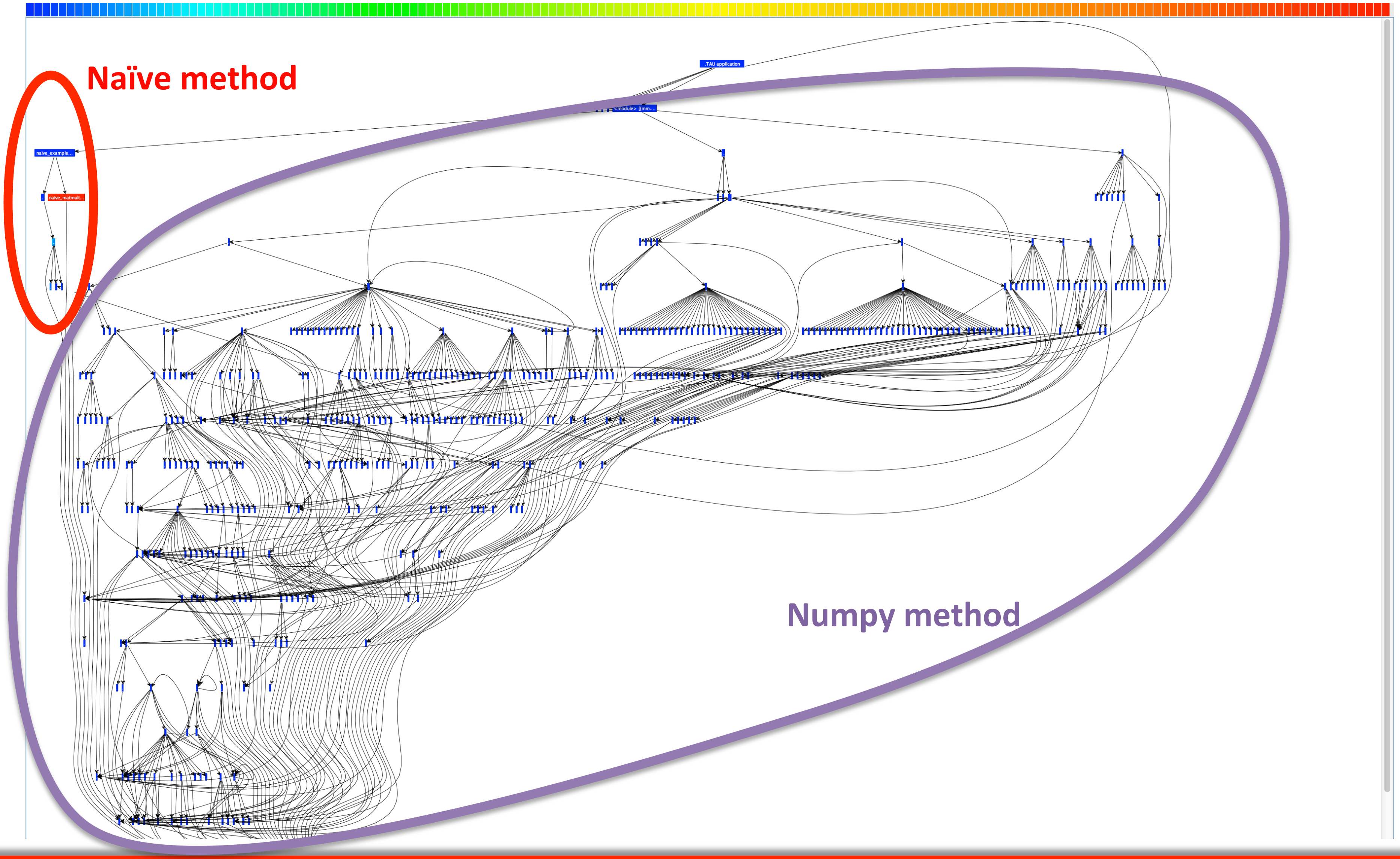
TAU: ParaProf: Group Legend: mm_py_callpath.ppk/

- 🟫 TAU_CALLPATH
- 🟥 TAU_CALLPATH_DERIVED
- 🟩 TAU_DERIVED
- 🟨 TAU_DISABLE
- 🟦 TAU_PYTHON
- 🟪 TAU_USER

Callgraph in ParaProf



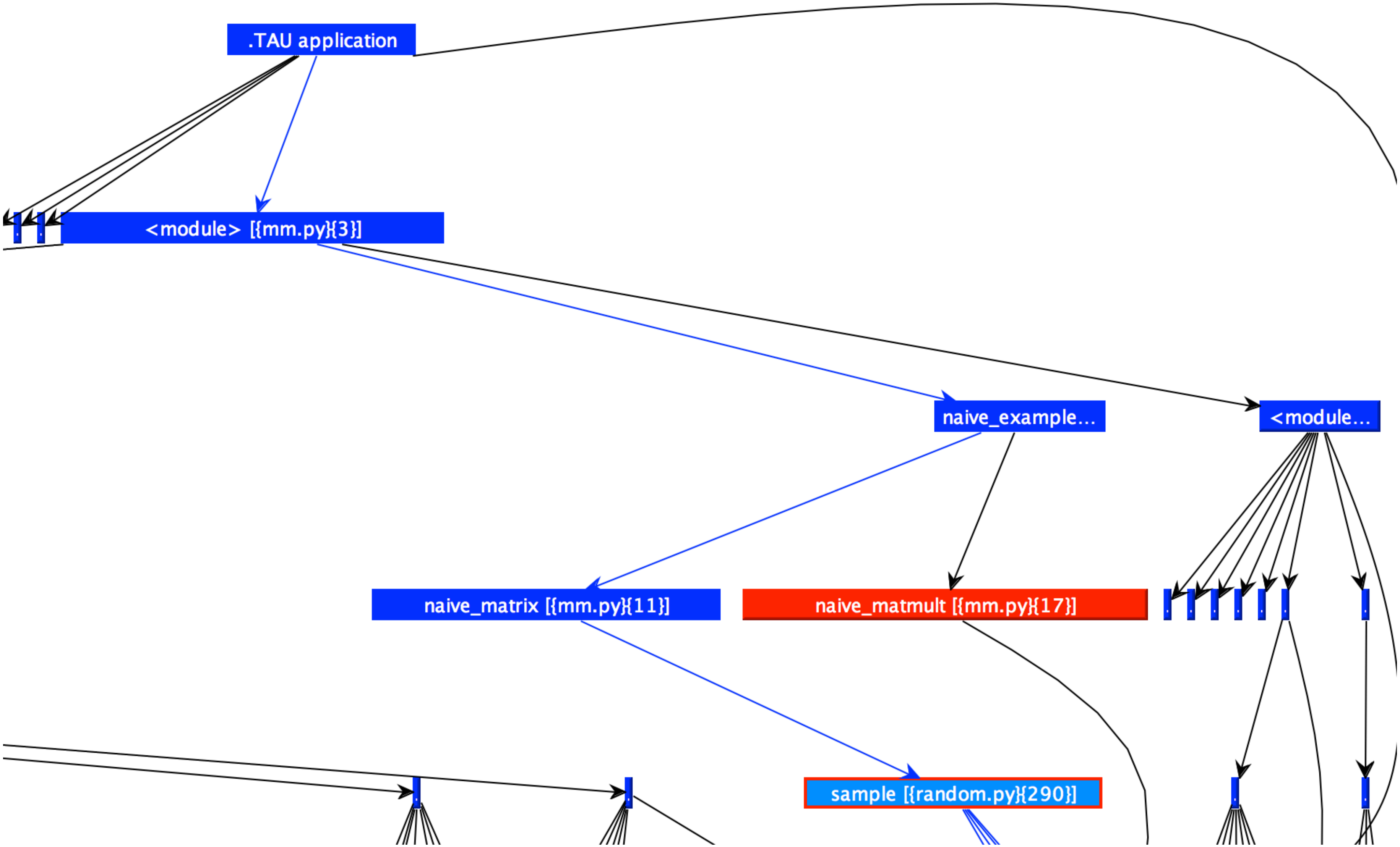
Callgraph in ParaProf



Naïve method

Numpy method

Callgraph in ParaProf


.TAU application
<module> [{mm.py}{3}]
naive_example…
<module…
naive_matrix [{mm.py}{11}]
naive_matmult [{mm.py}{17}]
sample [{random.py}{290}]

Traces with Pure Python

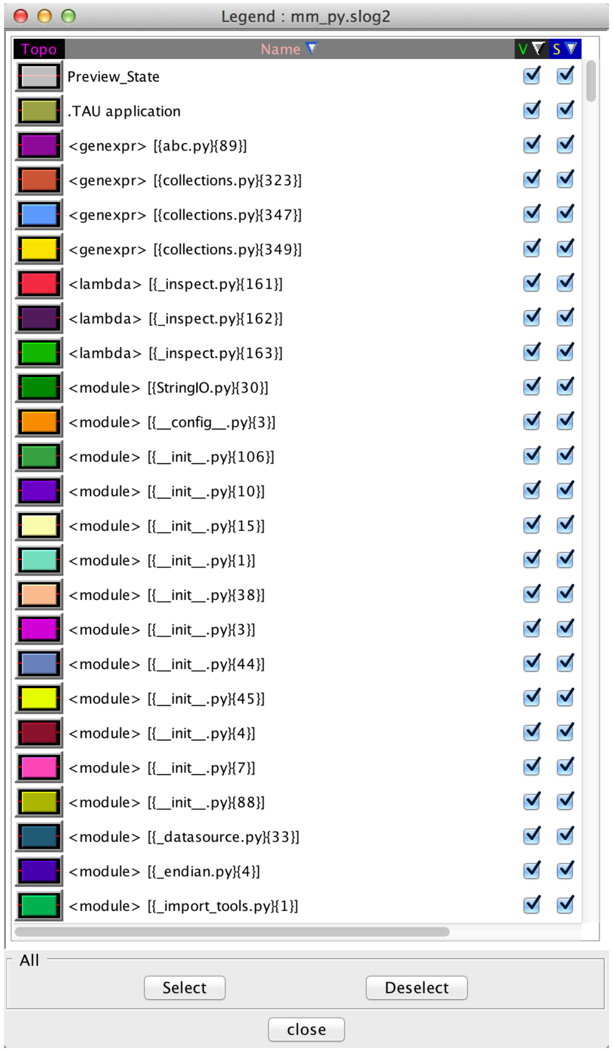
To generate traces:

```
$ unset TAU_CALLPATH    #recommended
$ export TAU_TRACE=1
$ tau_python mm.py
```

Trace files must be post-processed:

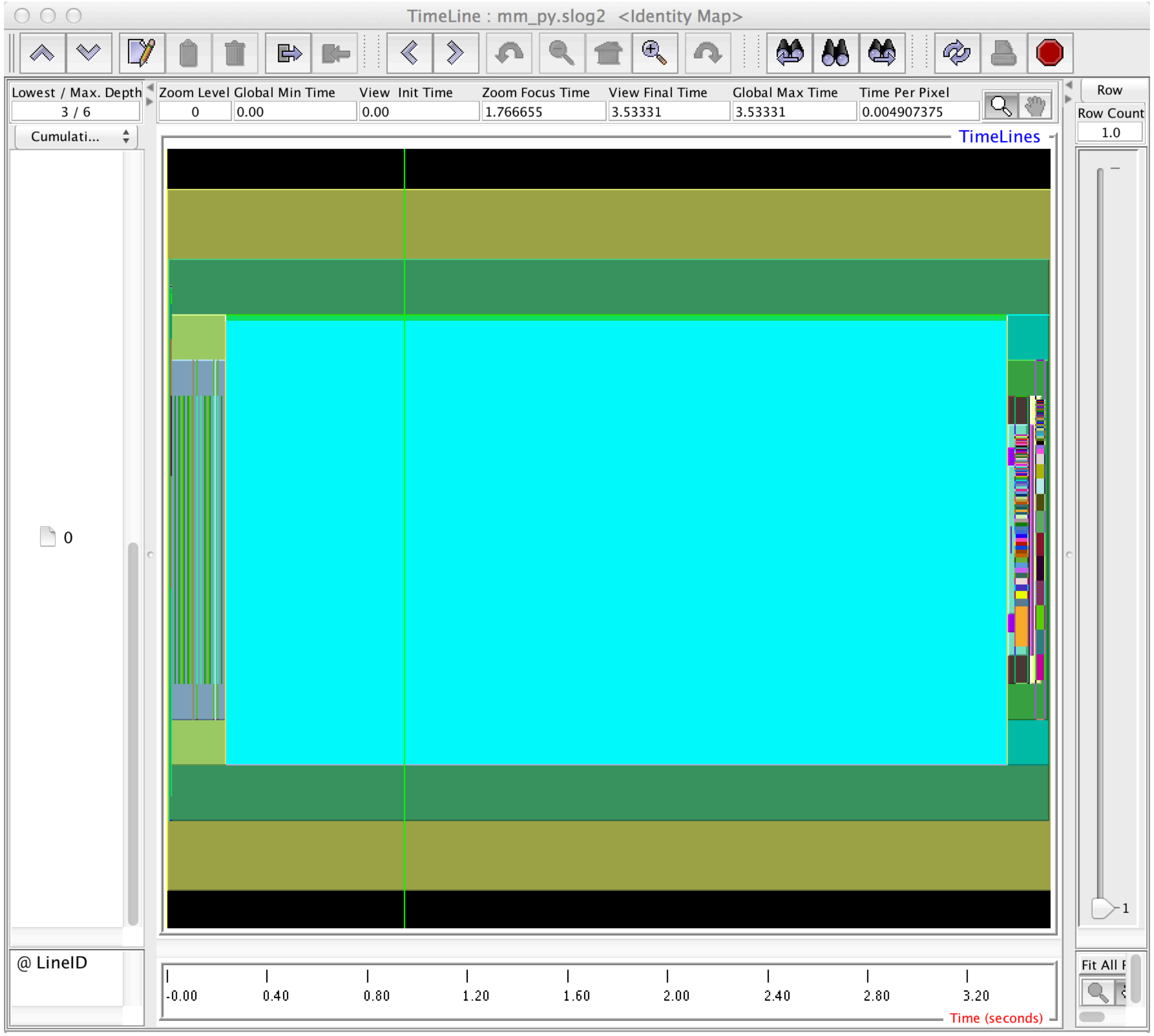
```
$ tau_treemerge.pl
$ tau2slog2 tau.trc tau.edf -o \
        mm_py.slog2
$ jumpshot mm_py.slog2
```

Jumpshot Trace Viewer



Legend : mm_py.slog2

Topo	Name	V	S
■	Preview_State	☑	☑
■	.TAU application	☑	☑
■	<genexpr> [{abc.py}{89}]	☑	☑
■	<genexpr> [{collections.py}{323}]	☑	☑
■	<genexpr> [{collections.py}{347}]	☑	☑
■	<genexpr> [{collections.py}{349}]	☑	☑
■	<lambda> [{_inspect.py}{161}]	☑	☑
■	<lambda> [{_inspect.py}{162}]	☑	☑
■	<lambda> [{_inspect.py}{163}]	☑	☑
■	<module> [{StringIO.py}{30}]	☑	☑
■	<module> [{__config__.py}{3}]	☑	☑
■	<module> [{__init__.py}{106}]	☑	☑
■	<module> [{__init__.py}{10}]	☑	☑
■	<module> [{__init__.py}{15}]	☑	☑
■	<module> [{__init__.py}{1}]	☑	☑
■	<module> [{__init__.py}{38}]	☑	☑
■	<module> [{__init__.py}{3}]	☑	☑
■	<module> [{__init__.py}{44}]	☑	☑
■	<module> [{__init__.py}{45}]	☑	☑
■	<module> [{__init__.py}{4}]	☑	☑
■	<module> [{__init__.py}{7}]	☑	☑
■	<module> [{__init__.py}{88}]	☑	☑
■	<module> [{_datasource.py}{33}]	☑	☑
■	<module> [{_endian.py}{4}]	☑	☑
■	<module> [{_import_tools.py}{1}]	☑	☑



TimeLine : mm_py.slog2 <Identity Map>

Lowest / Max. Depth	Zoom Level	Global Min Time	View Init Time	Zoom Focus Time	View Final Time	Global Max Time	Time Per Pixel	Row Count
3 / 6	0	0.00	0.00	1.766655	3.53331	3.53331	0.004907375	1.0

Public Service Annoucement

Don't forget to clean your environment!

(Some folks write scripts)

Show all TAU environment variables:

```
$ env | grep TAU
```

Unset the ones you don't need anymore:

```
$ unset TAU_TRACE
$ unset TAU_CALLPATH
```

etc.

Python Performance Evaluation

HANDS-ON: NATIVE LANGUAGES

TAU with C/C++

```
$ cd workshop-python/01_matmult.c
$ make CC=tau_cc.sh
```

Run normally to generate profiles:

```
$ mpirun -np 4 ./matmult
$ ls profile.*                # Shows four files
$ paraprof --pack mm_c_flat.ppk
```

View the profiles:

```
pprof -a | less               #Command line
paraprof                      #GUI (Java, X11)
```

TAU with Fortran

```
$ cd workshop-python/02_matmult.f90
$ make
```

Run normally to generate profiles:

```
$ mpirun -np 4 ./matmult
$ ls profile.*              # Shows four files
$ paraprof --pack mm_f90_flat.ppk
```

View the profiles:

```
pprof -a | less             #Command line
paraprof                    #GUI (Java, X11)
```

Python Performance Evaluation

HANDS-ON: PYTHON+MPI (MPI4PY)

FIXEDGRID

A simple chemical transport model in Python

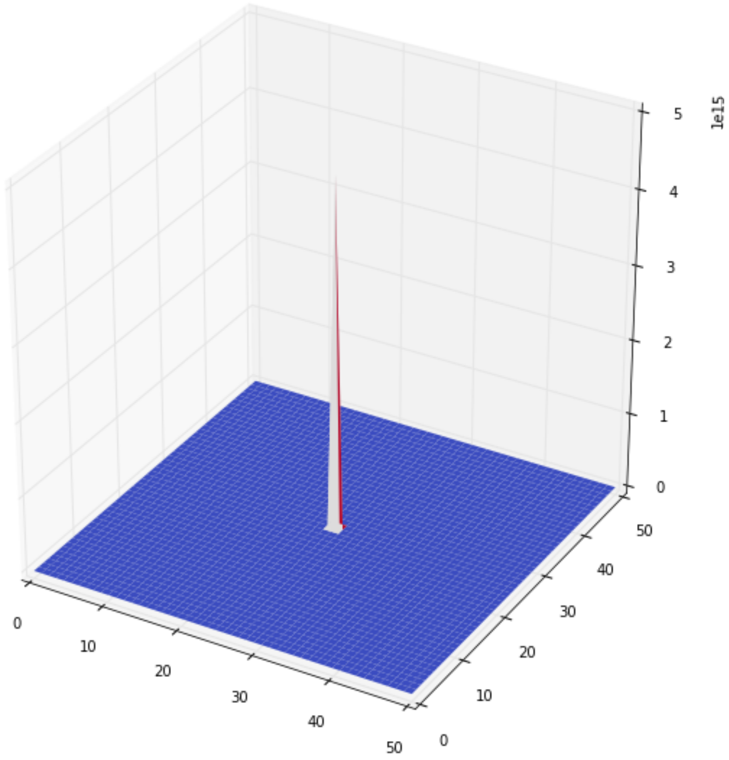
$$\frac{\partial \mathbf{c}_x^t}{\partial t} = \sum_{k=1}^{d} \left[\frac{\partial}{\partial x_k} \left(d_k(x,t) \frac{\partial \mathbf{c}_x^t}{\partial x_k} - a_k(x,t)\mathbf{c}_x^t \right) \right] + F$$

Advection: Upwind-biased 2nd order finite differences

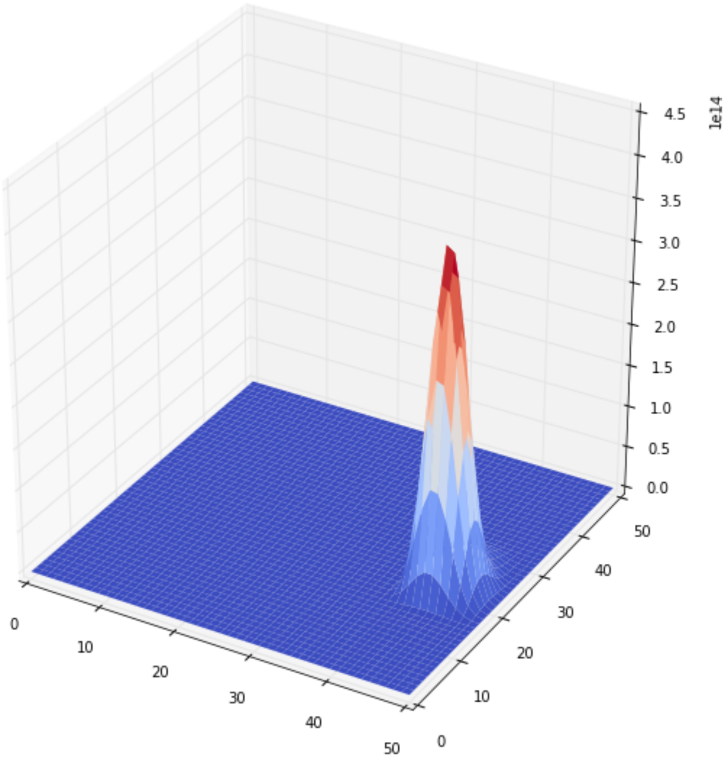
Diffusion: 3rd order finite differences

Chemistry: Rosenbrock time-stepping integrator

FIXEDGRID

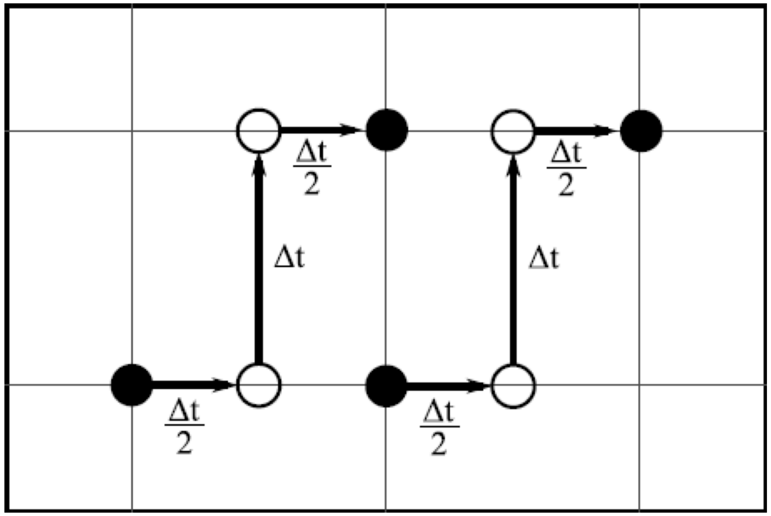


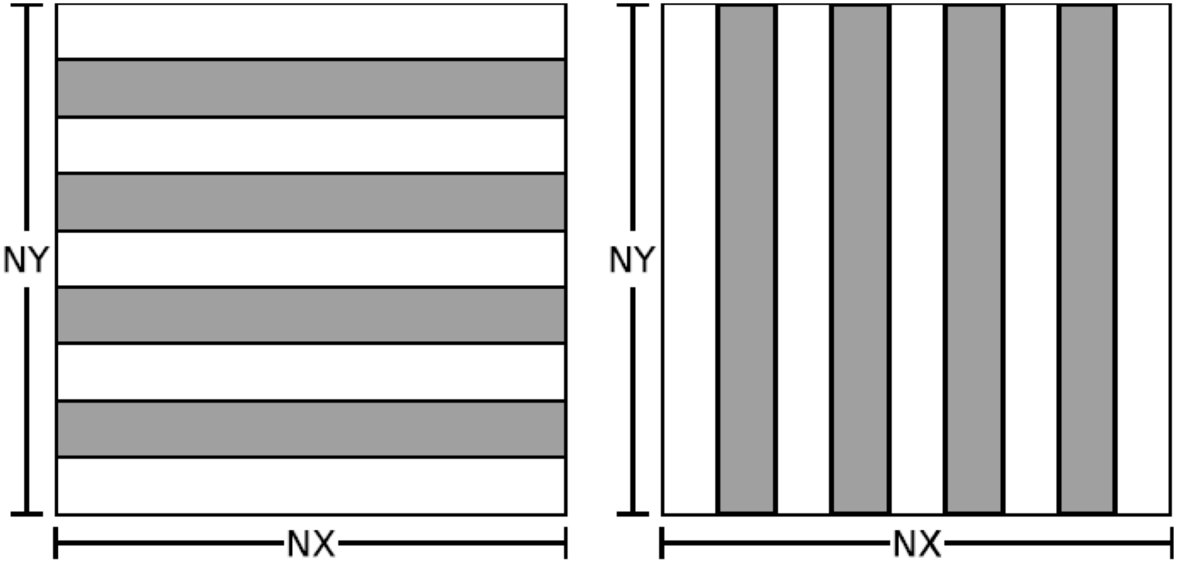
TIME = 0



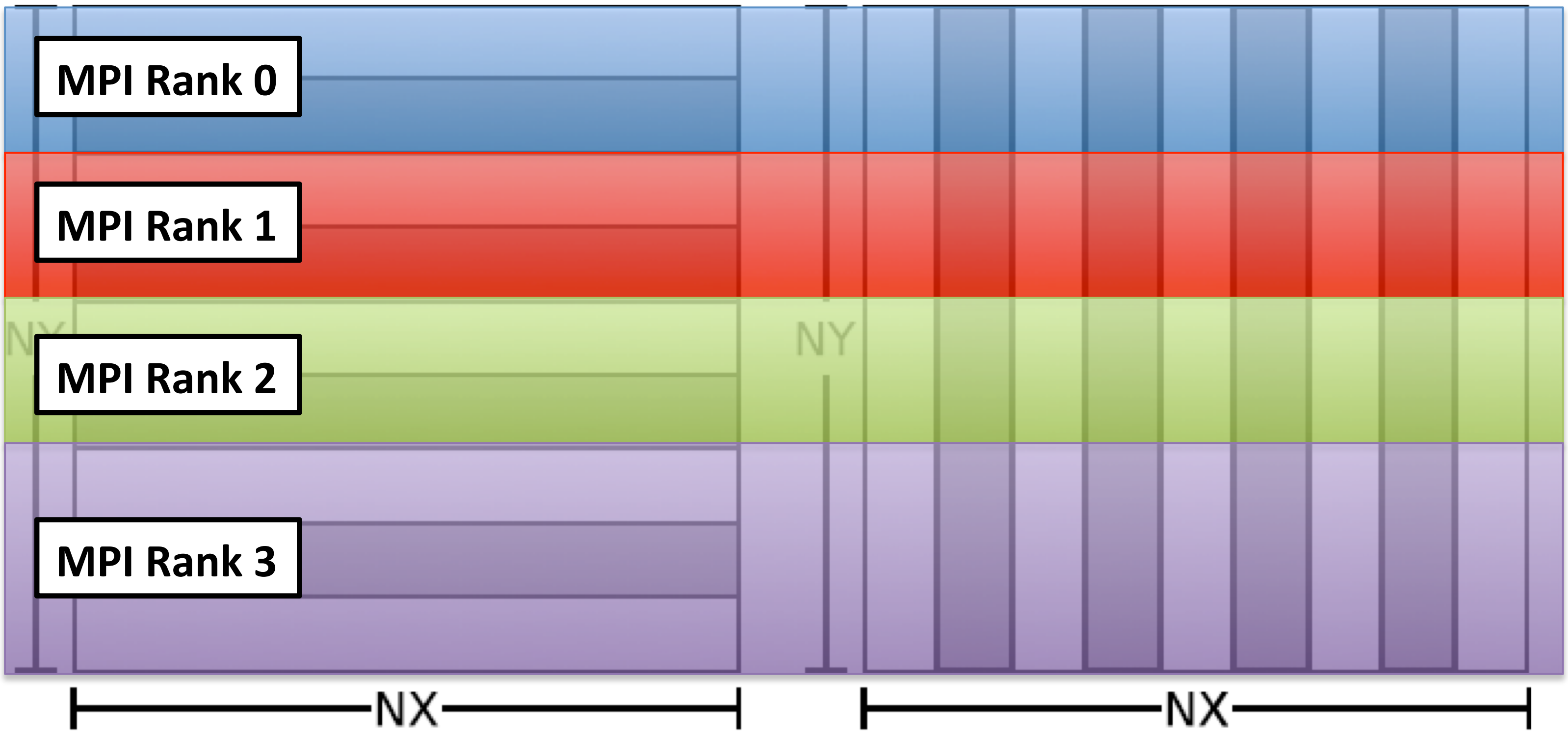
TIME = 900 seconds

2nd Order Dimension Split in FIXEDGRID





MPI in FIXEDGRID



TAU with mpi4py

```
$ cd 04_fixedgrid-mpi.py
$ mpirun -np 4 python fixedgrid.py
```

Run with tau_exec and wrapper.py to generate profiles:

```
$ mpirun -np 4 tau_exec -T python,mpi \
      python wrapper.py
```

View the profiles:

```
pprof -a | less          #Command line
paraprof                 #GUI (Java, X11)
```

Multiple Layers of Instrumentation

```
$ mpirun -np 4 \
    tau_exec –T python,mpi \
    python wrapper.py
```

a.k.a

$ “Run my code” \

“Use TAU to measure MPI” \

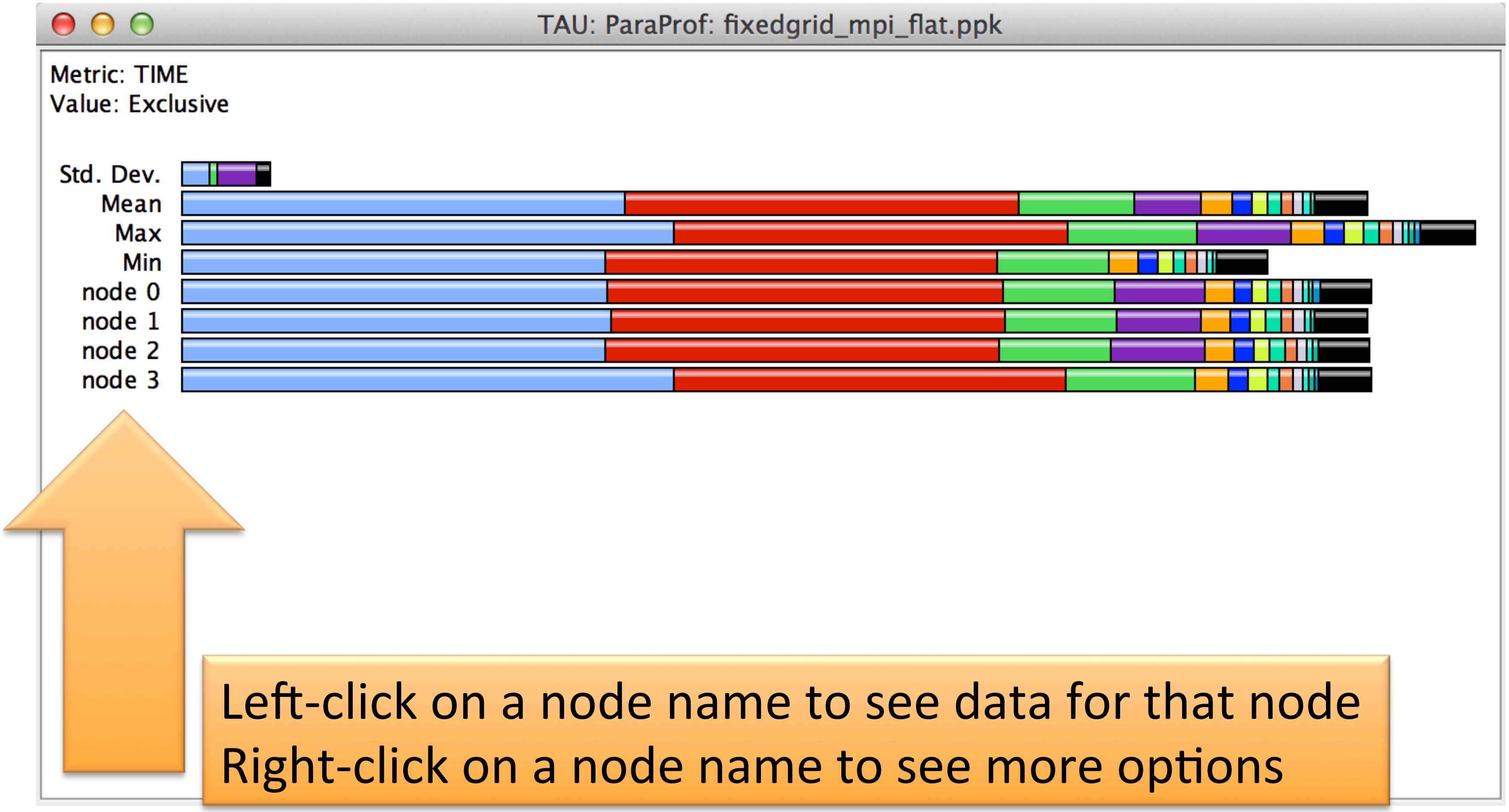
“Within that TAU instance, instrument python”

wrapper.py for Python Instrumentation

```
$ cat wrapper.py
import tau
tau.run('import fixedgrid')
```

This approach works for many Python packages, not just mpi4py

FIXEDGRID Profile

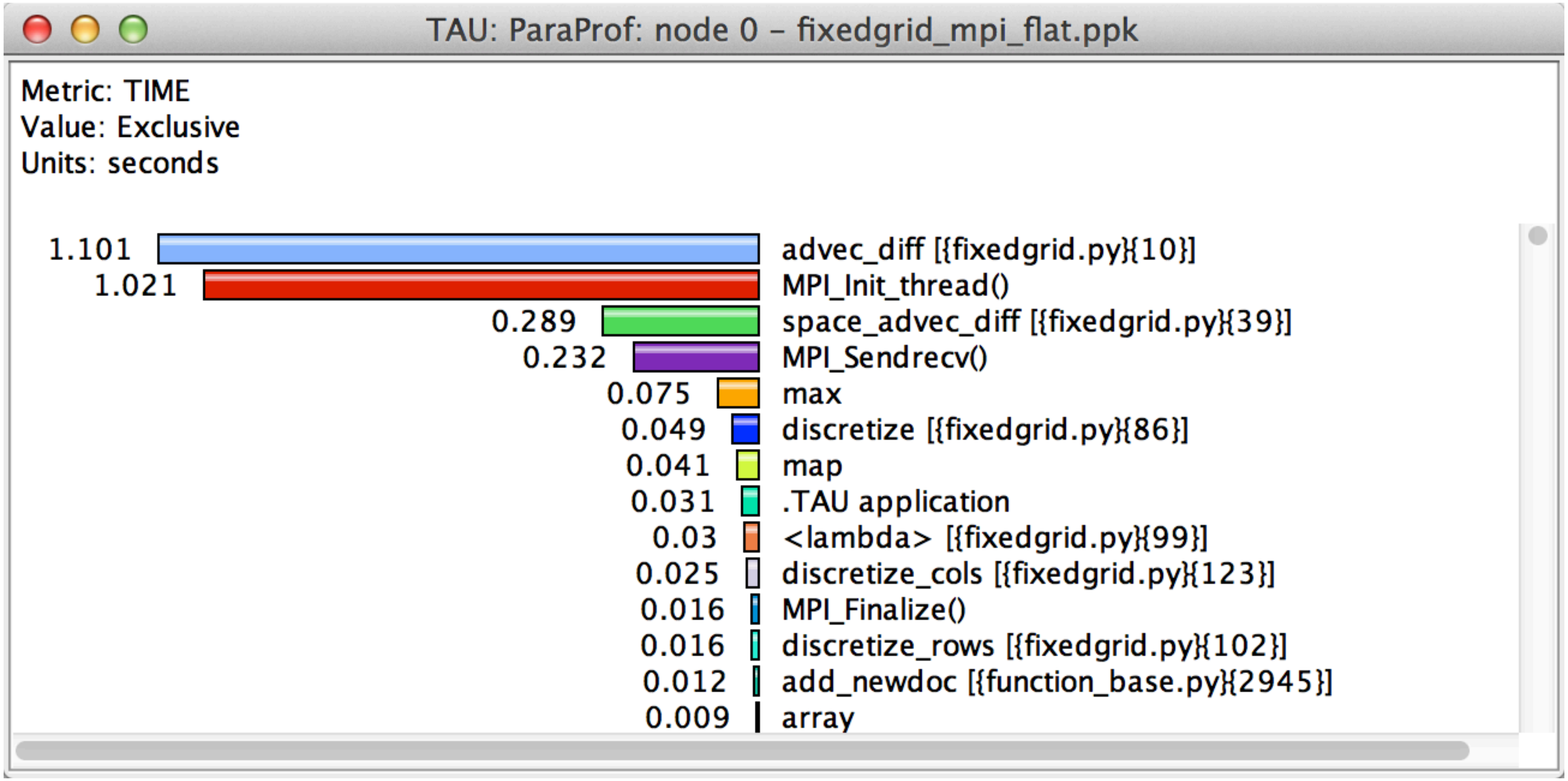


TAU: ParaProf: fixedgrid_mpi_flat.ppk

Metric: TIME
Value: Exclusive

Left-click on a node name to see data for that node
Right-click on a node name to see more options

FIXEDGRID Profile



TAU: ParaProf: node 0 – fixedgrid_mpi_flat.ppk

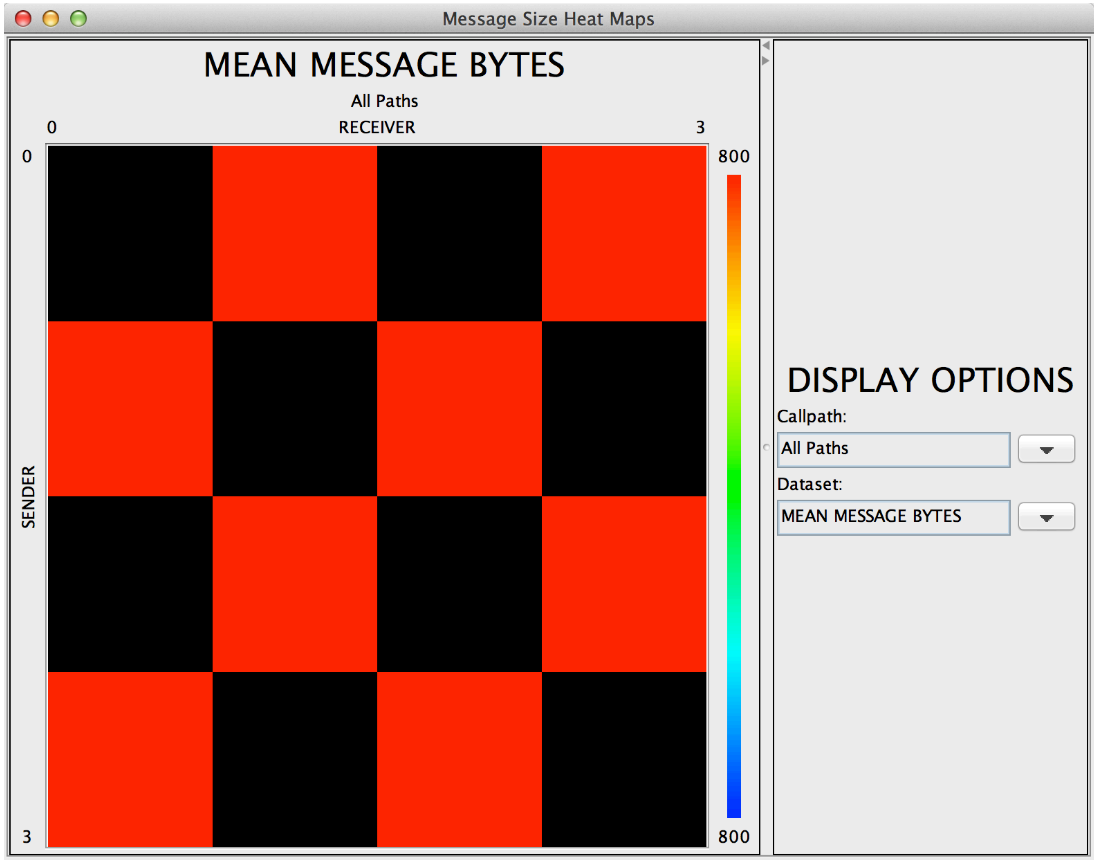
Metric: TIME
Value: Exclusive
Units: seconds

1.101	advec_diff [{fixedgrid.py}{10}]
1.021	MPI_Init_thread()
0.289	space_advec_diff [{fixedgrid.py}{39}]
0.232	MPI_Sendrecv()
0.075	max
0.049	discretize [{fixedgrid.py}{86}]
0.041	map
0.031	.TAU application
0.03	<lambda> [{fixedgrid.py}{99}]
0.025	discretize_cols [{fixedgrid.py}{123}]
0.016	MPI_Finalize()
0.016	discretize_rows [{fixedgrid.py}{102}]
0.012	add_newdoc [{function_base.py}{2945}]
0.009	array

FIXEDGRID Communication Matrix

```
$ export TAU_COMM_MATRIX=1
$ mpirun -np 4 tau_exec -T python,mpi python wrapper.py
```

In Paraprof: **Windows | Communication Matrix**

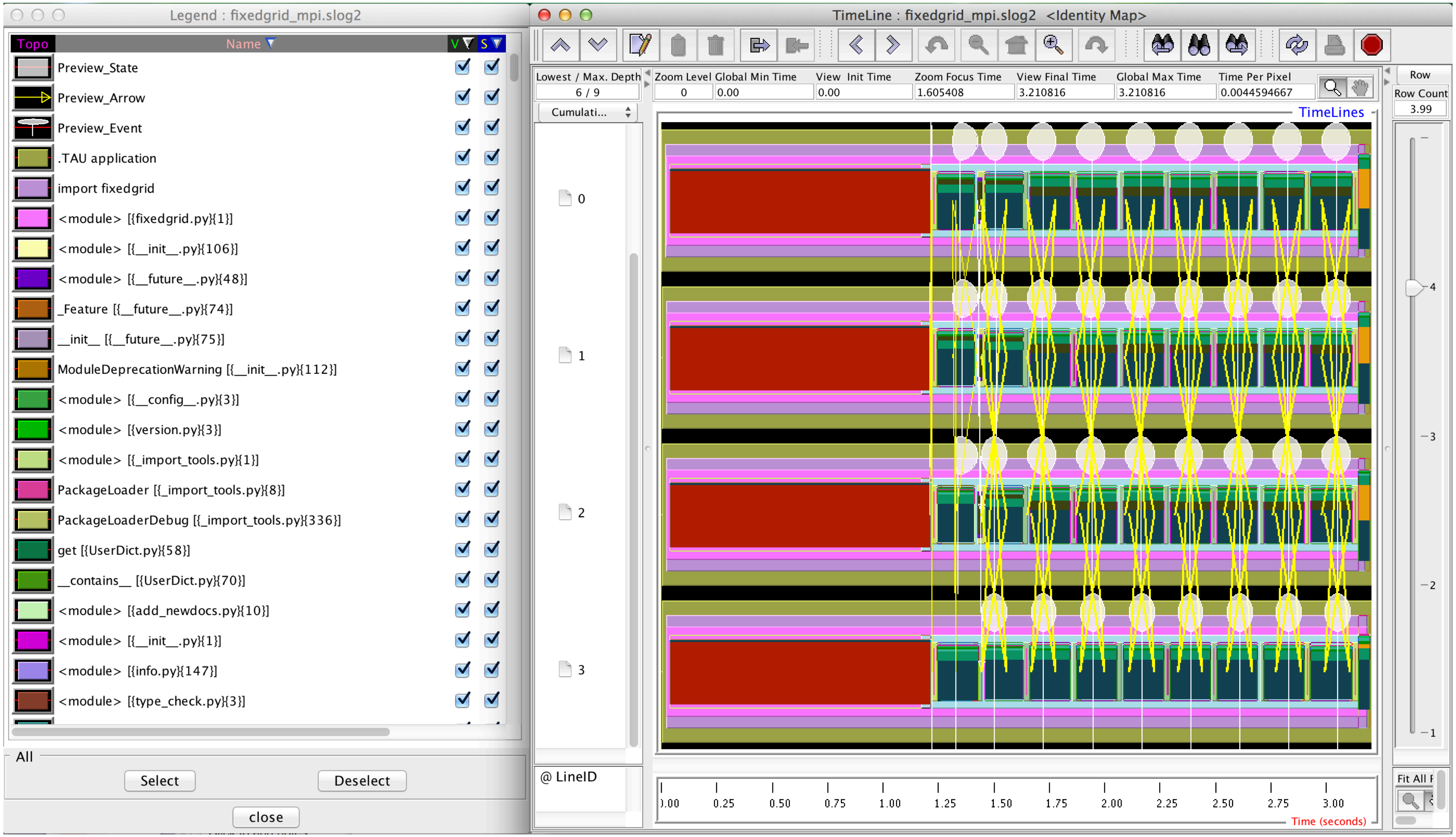


Callpath: All Paths

Dataset: MEAN MESSAGE BYTES

FIXEDGRID Trace Shows Communication

```
$ jumpshot fixedgrid_mpi.slog2
```



Legend : fixedgrid_mpi.slog2

Topo	Name	V	S
■	Preview_State	☑	☑
■	Preview_Arrow	☑	☑
■	Preview_Event	☑	☑
■	.TAU application	☑	☑
■	import fixedgrid	☑	☑
■	<module> [{fixedgrid.py}{1}]	☑	☑
■	<module> [{__init__.py}{106}]	☑	☑
■	<module> [{__future__.py}{48}]	☑	☑
■	_Feature [{__future__.py}{74}]	☑	☑
■	__init__ [{__future__.py}{75}]	☑	☑
■	ModuleDeprecationWarning [{__init__.py}{112}]	☑	☑
■	<module> [{__config__.py}{3}]	☑	☑
■	<module> [{version.py}{3}]	☑	☑
■	<module> [{_import_tools.py}{1}]	☑	☑
■	PackageLoader [{_import_tools.py}{8}]	☑	☑
■	PackageLoaderDebug [{_import_tools.py}{336}]	☑	☑
■	get [{UserDict.py}{58}]	☑	☑
■	__contains__ [{UserDict.py}{70}]	☑	☑
■	<module> [{add_newdocs.py}{10}]	☑	☑
■	<module> [{__init__.py}{1}]	☑	☑
■	<module> [{info.py}{147}]	☑	☑
■	<module> [{type_check.py}{3}]	☑	☑

All

TimeLine : fixedgrid_mpi.slog2 <Identity Map>

Lowest / Max. Depth	Zoom Level	Global Min Time	View Init Time	Zoom Focus Time	View Final Time	Global Max Time	Time Per Pixel
6 / 9	0	0.00	0.00	1.605408	3.210816	3.210816	0.0044594667

Row Count: 3.99

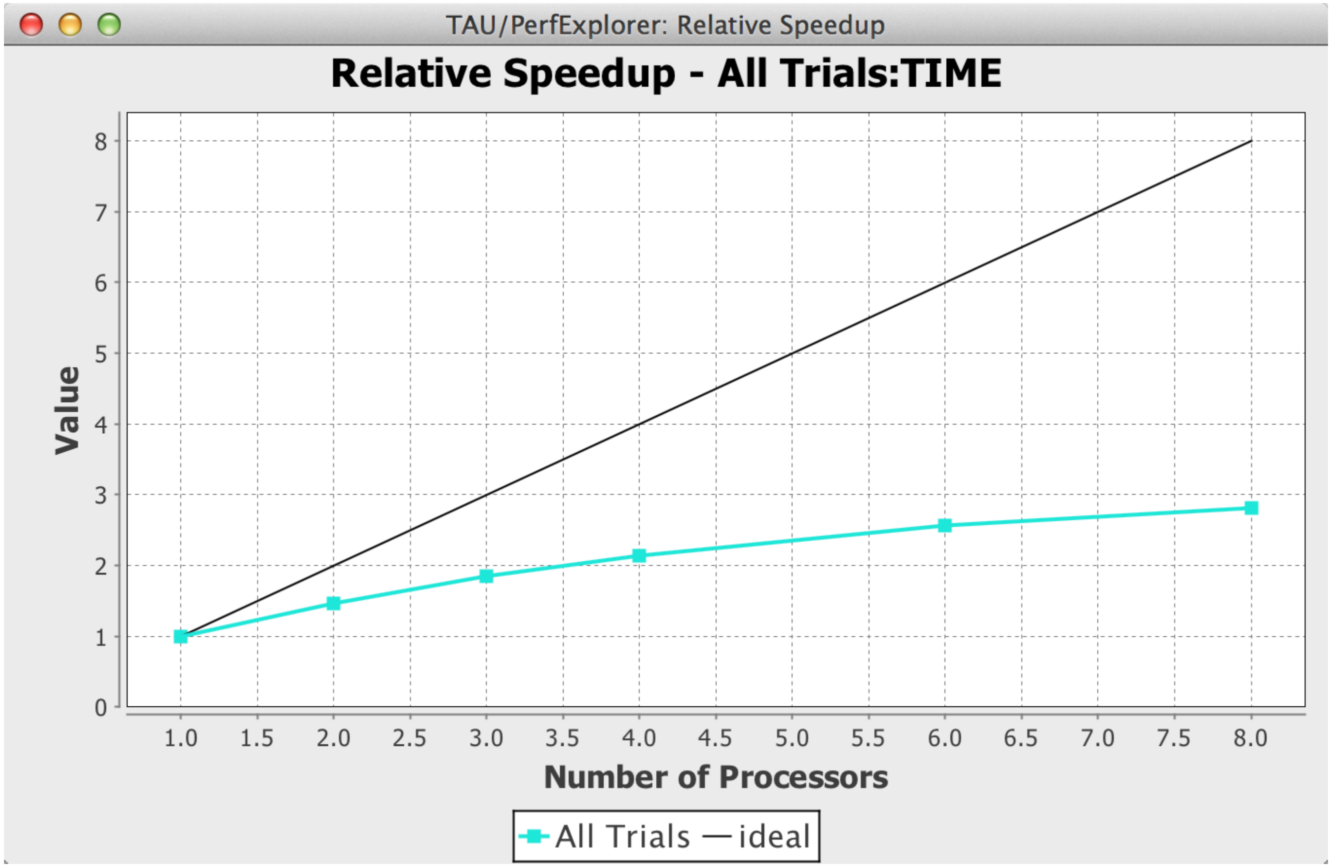
Cumulati...

PerfExplorer

```
$ cd 04_fixedgrid-mpi.py/analysis
$ taudb_configure --create-default
$ taudb_loadtrial fixedgrid_np1.ppk
$ taudb_loadtrial fixedgrid_np2.ppk
$ taudb_loadtrial fixedgrid_np3.ppk
…
$ perfexplorer
```

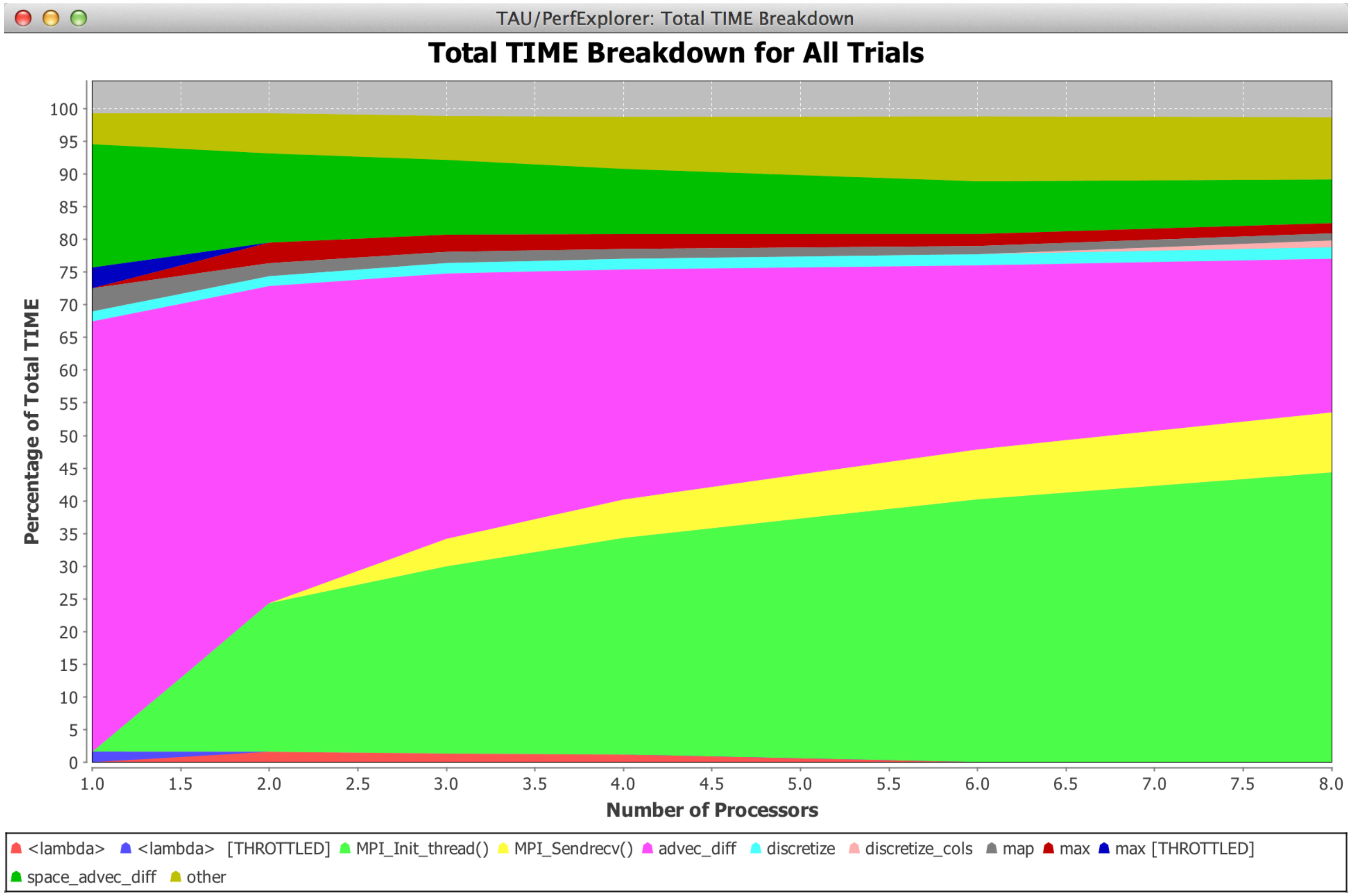
Relative Speedup Chart

- In PerfExplorer: **Charts | Relative Speedup**



Runtime Breakdown Chart

- In PerfExplorer: **Charts | Runtime Breakdown**

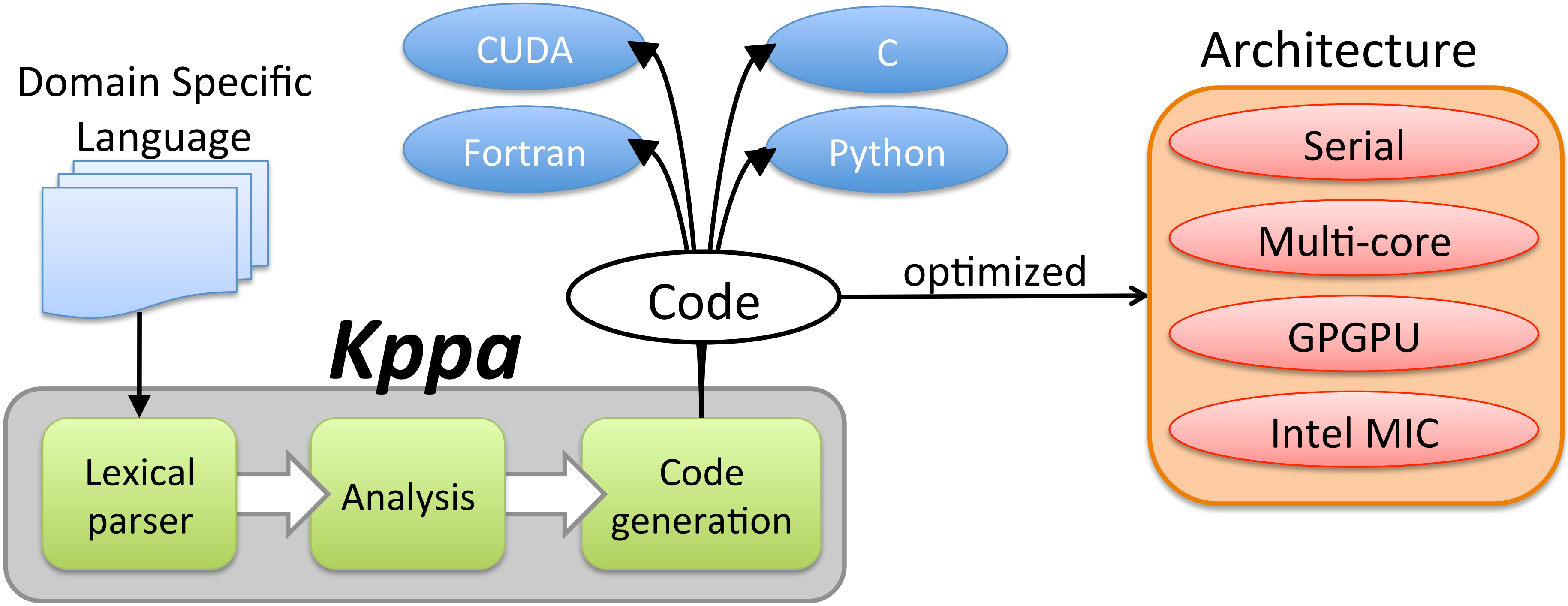


Python Performance Evaluation

HANDS-ON: PYTHON+X

(BECAUSE WE CAN)

Kppa: The Kinetic PreProcessor Accelerated



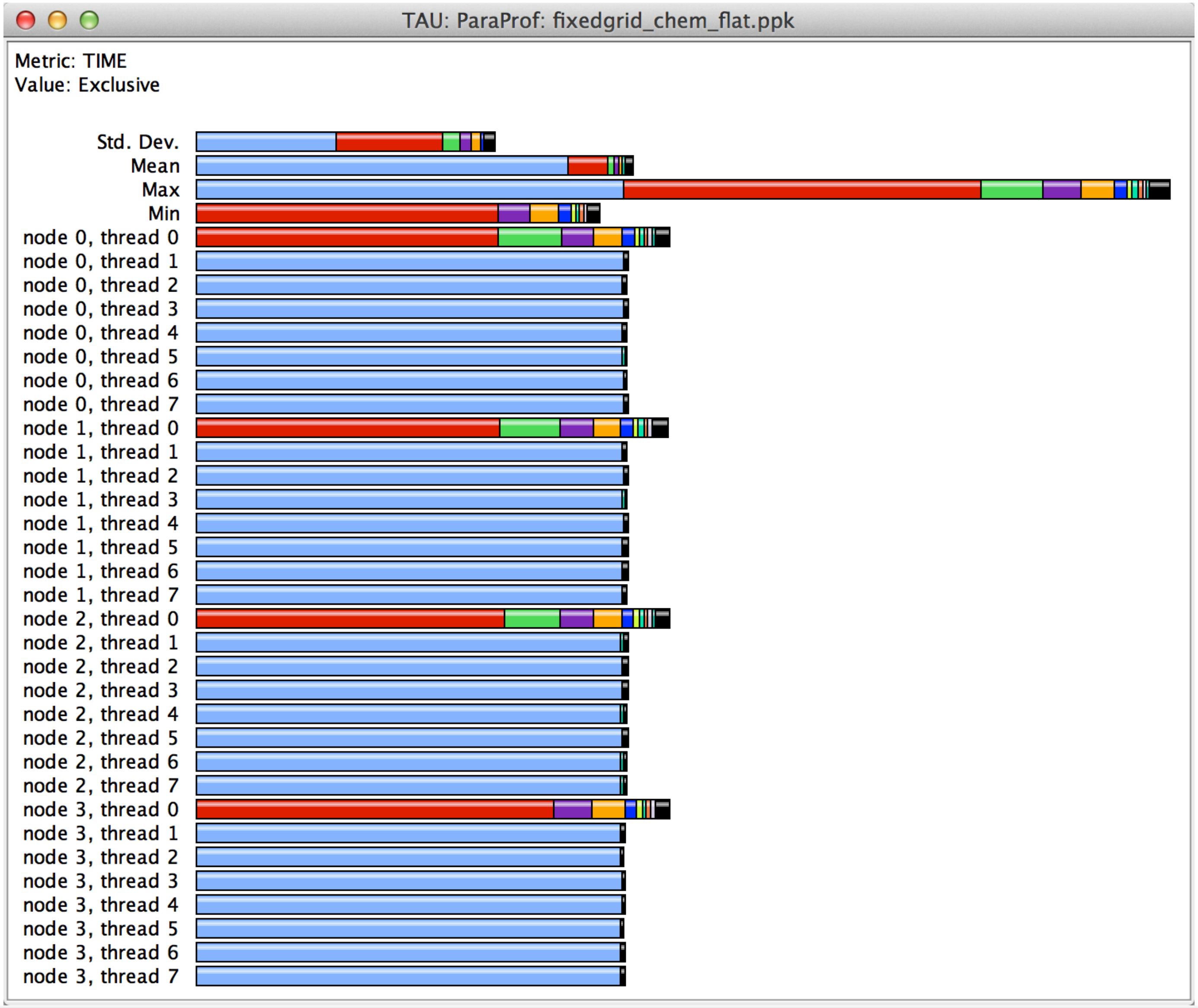
TAU + Python + mpi4py + C + OpenMP

```
$ cd 05_fixedgrid-chem.c_py
$ make
$ mpirun -np 4 python fixedgrid.py
```

Run with tau_exec and wrapper.py to generate profiles:

```
$ make clean
$ make CC=tau_cc.sh
$ mpirun -np 4 tau_exec -T python,mpi,openmp \
       python wrapper.py
```

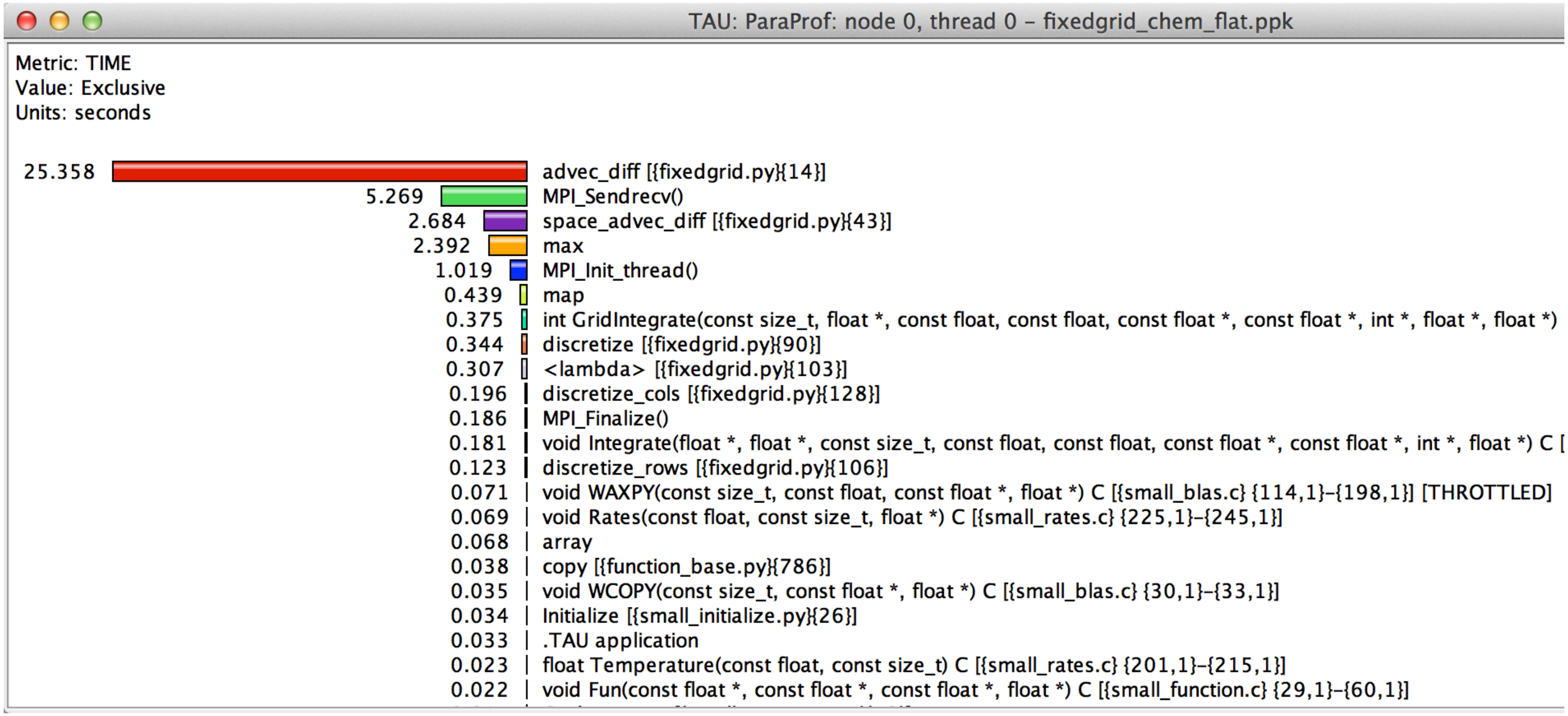
MPI + OpenMP Profiles



TAU: ParaProf: fixedgrid_chem_flat.ppk

Metric: TIME
Value: Exclusive

Rank 0, Thread 0

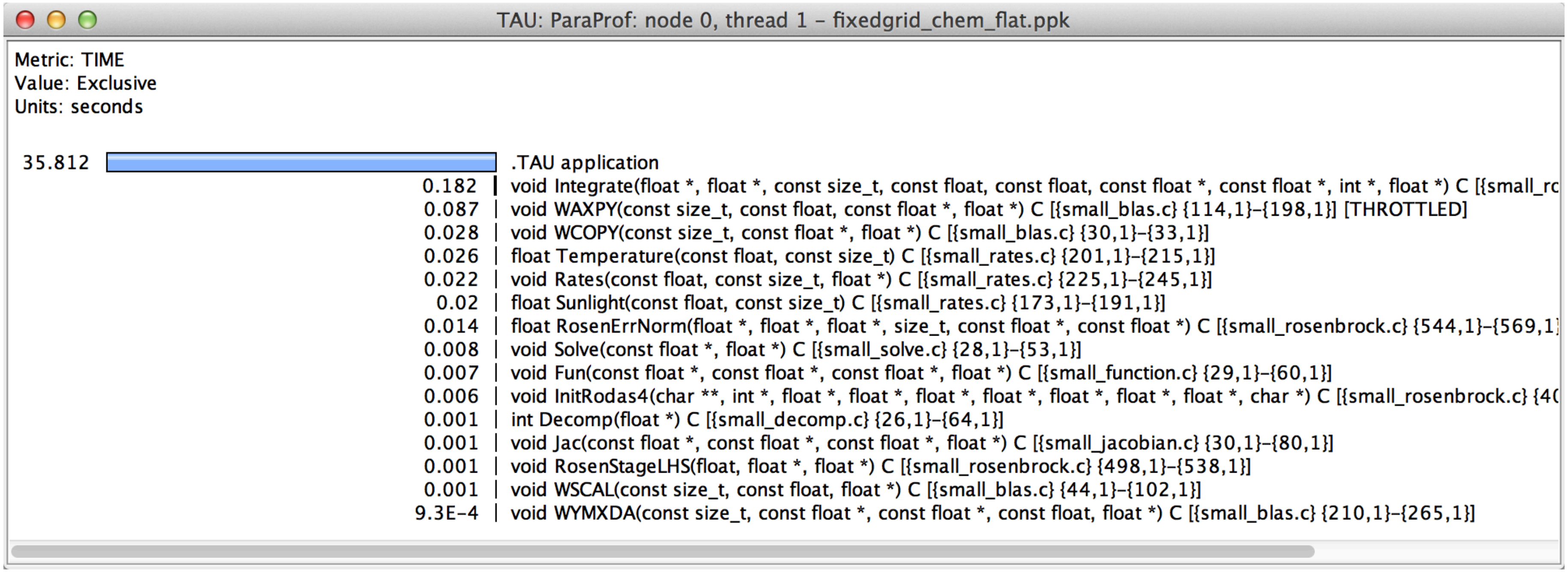


TAU: ParaProf: node 0, thread 0 – fixedgrid_chem_flat.ppk

Metric: TIME
Value: Exclusive
Units: seconds

```
25.358  advec_diff [{fixedgrid.py}{14}]
 5.269  MPI_Sendrecv()
 2.684  space_advec_diff [{fixedgrid.py}{43}]
 2.392  max
 1.019  MPI_Init_thread()
 0.439  map
 0.375  int GridIntegrate(const size_t, float *, const float, const float, const float *, const float *, int *, float *, float *)
 0.344  discretize [{fixedgrid.py}{90}]
 0.307  <lambda> [{fixedgrid.py}{103}]
 0.196  discretize_cols [{fixedgrid.py}{128}]
 0.186  MPI_Finalize()
 0.181  void Integrate(float *, float *, const size_t, const float, const float, const float *, const float *, int *, float *) C [
 0.123  discretize_rows [{fixedgrid.py}{106}]
 0.071  void WAXPY(const size_t, const float, const float *, float *) C [{small_blas.c} {114,1}-{198,1}] [THROTTLED]
 0.069  void Rates(const float, const size_t, float *) C [{small_rates.c} {225,1}-{245,1}]
 0.068  array
 0.038  copy [{function_base.py}{786}]
 0.035  void WCOPY(const size_t, const float *, float *) C [{small_blas.c} {30,1}-{33,1}]
 0.034  Initialize [{small_initialize.py}{26}]
 0.033  .TAU application
 0.023  float Temperature(const float, const size_t) C [{small_rates.c} {201,1}-{215,1}]
 0.022  void Fun(const float *, const float *, const float *, float *) C [{small_function.c} {29,1}-{60,1}]
```

Rank 0, Thread 1



TAU: ParaProf: node 0, thread 1 – fixedgrid_chem_flat.ppk

```
Metric: TIME
Value: Exclusive
Units: seconds

35.812   .TAU application
 0.182   void Integrate(float *, float *, const size_t, const float, const float, const float *, const float *, int *, float *) C [{small_rc
 0.087   void WAXPY(const size_t, const float, const float *, float *) C [{small_blas.c} {114,1}-{198,1}] [THROTTLED]
 0.028   void WCOPY(const size_t, const float *, float *) C [{small_blas.c} {30,1}-{33,1}]
 0.026   float Temperature(const float, const size_t) C [{small_rates.c} {201,1}-{215,1}]
 0.022   void Rates(const float, const size_t, float *) C [{small_rates.c} {225,1}-{245,1}]
  0.02   float Sunlight(const float, const size_t) C [{small_rates.c} {173,1}-{191,1}]
 0.014   float RosenErrNorm(float *, float *, float *, size_t, const float *, const float *) C [{small_rosenbrock.c} {544,1}-{569,1}
 0.008   void Solve(const float *, float *) C [{small_solve.c} {28,1}-{53,1}]
 0.007   void Fun(const float *, const float *, const float *, float *) C [{small_function.c} {29,1}-{60,1}]
 0.006   void InitRodas4(char **, int *, float *, float *, float *, float *, float *, float *, float *, char *) C [{small_rosenbrock.c} {4(
 0.001   int Decomp(float *) C [{small_decomp.c} {26,1}-{64,1}]
 0.001   void Jac(const float *, const float *, const float *, float *) C [{small_jacobian.c} {30,1}-{80,1}]
 0.001   void RosenStageLHS(float, float *, float *) C [{small_rosenbrock.c} {498,1}-{538,1}]
 0.001   void WSCAL(const size_t, const float, float *) C [{small_blas.c} {44,1}-{102,1}]
9.3E-4   void WYMXDA(const size_t, const float *, const float *, const float, float *) C [{small_blas.c} {210,1}-{265,1}]
```

Python Performance Evaluation

HANDS-ON: DEBUGGING

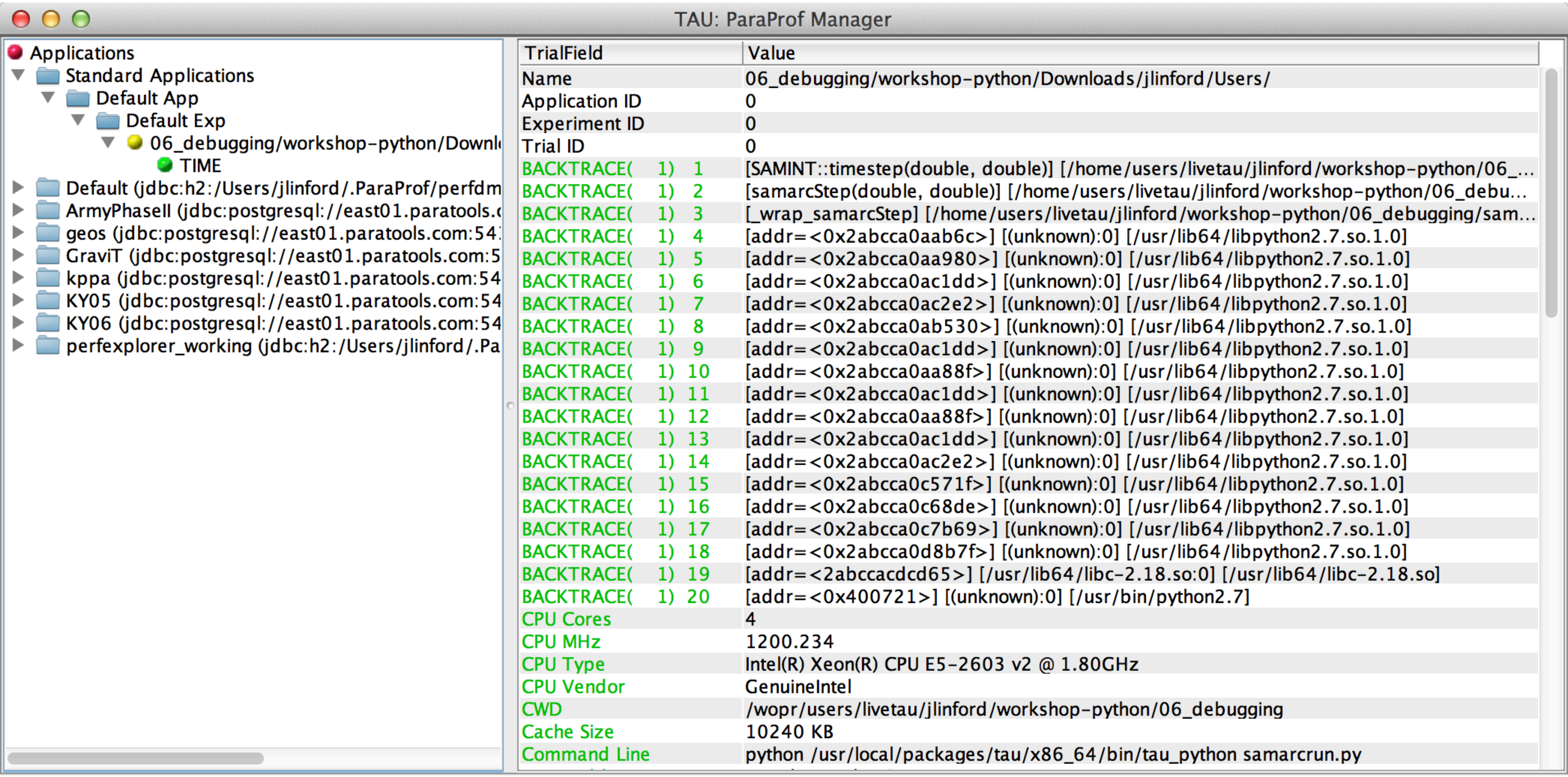
TAU + Python + mpi4py + C + OpenMP

```
$ cd 06_debugging
$ make
$ tau_python samarcrun.py
TAU: Caught signal 8 (Floating point exception), …
```

To see stack trace on command line:

```
$ paraprof -d | grep BACKTRACE
```

Backtrace Shown in ParaProf



TAU: ParaProf Manager

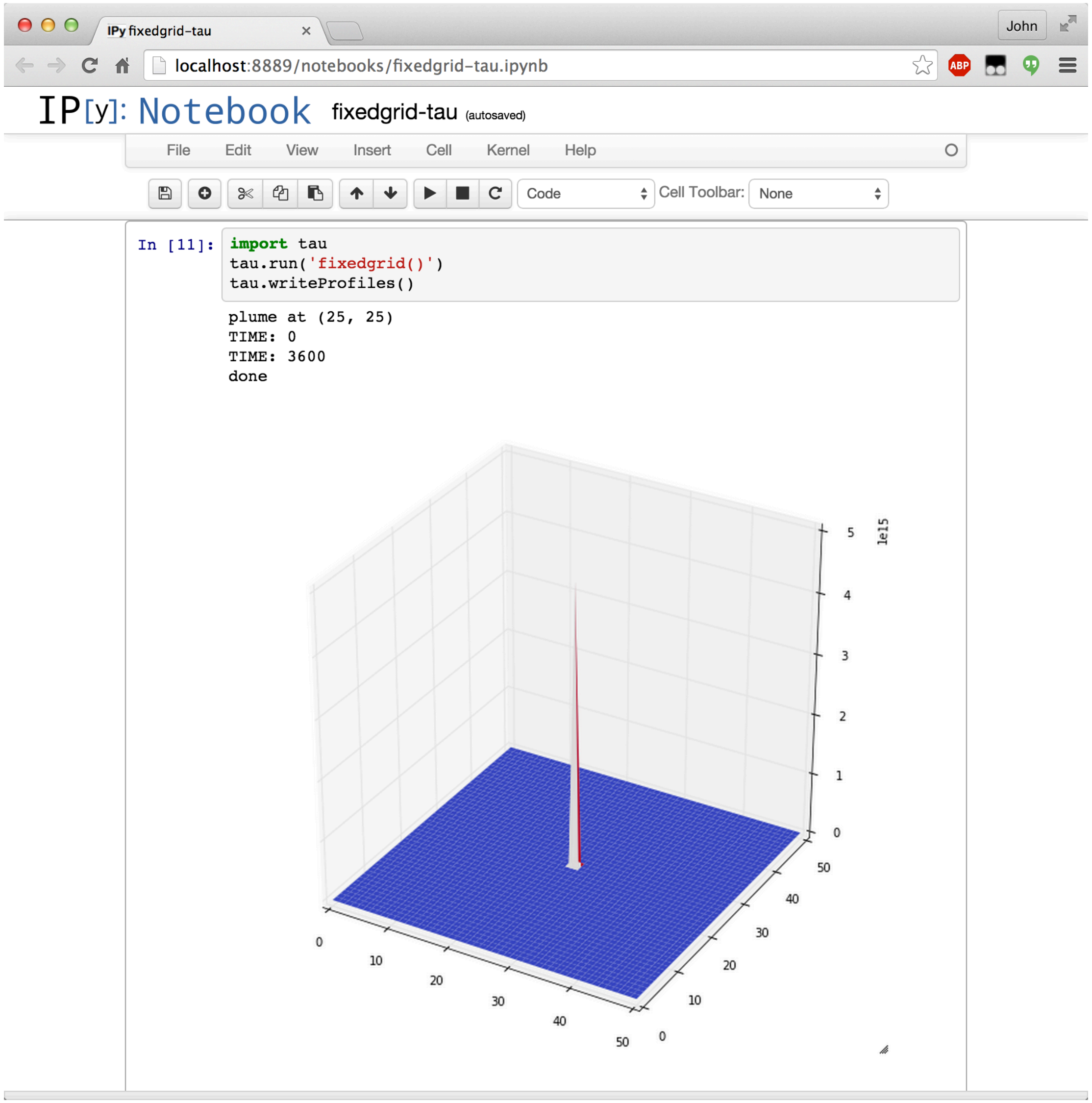
- Applications
  - Standard Applications
    - Default App
      - Default Exp
        - 06_debugging/workshop-python/Downl
          - TIME
- Default (jdbc:h2:/Users/jlinford/.ParaProf/perfdm
- ArmyPhasell (jdbc:postgresql://east01.paratools.c
- geos (jdbc:postgresql://east01.paratools.com:54
- GraviT (jdbc:postgresql://east01.paratools.com:5
- kppa (jdbc:postgresql://east01.paratools.com:54
- KY05 (jdbc:postgresql://east01.paratools.com:54
- KY06 (jdbc:postgresql://east01.paratools.com:54
- perfexplorer_working (jdbc:h2:/Users/jlinford/.Pa

TrialField	Value
Name	06_debugging/workshop-python/Downloads/jlinford/Users/
Application ID	0
Experiment ID	0
Trial ID	0
BACKTRACE(1) 1	[SAMINT::timestep(double, double)] [/home/users/livetau/jlinford/workshop-python/06_...
BACKTRACE(1) 2	[samarcStep(double, double)] [/home/users/livetau/jlinford/workshop-python/06_debu...
BACKTRACE(1) 3	[_wrap_samarcStep] [/home/users/livetau/jlinford/workshop-python/06_debugging/sam...
BACKTRACE(1) 4	[addr=<0x2abcca0aab6c>] [(unknown):0] [/usr/lib64/libpython2.7.so.1.0]
BACKTRACE(1) 5	[addr=<0x2abcca0aa980>] [(unknown):0] [/usr/lib64/libpython2.7.so.1.0]
BACKTRACE(1) 6	[addr=<0x2abcca0ac1dd>] [(unknown):0] [/usr/lib64/libpython2.7.so.1.0]
BACKTRACE(1) 7	[addr=<0x2abcca0ac2e2>] [(unknown):0] [/usr/lib64/libpython2.7.so.1.0]
BACKTRACE(1) 8	[addr=<0x2abcca0ab530>] [(unknown):0] [/usr/lib64/libpython2.7.so.1.0]
BACKTRACE(1) 9	[addr=<0x2abcca0ac1dd>] [(unknown):0] [/usr/lib64/libpython2.7.so.1.0]
BACKTRACE(1) 10	[addr=<0x2abcca0aa88f>] [(unknown):0] [/usr/lib64/libpython2.7.so.1.0]
BACKTRACE(1) 11	[addr=<0x2abcca0ac1dd>] [(unknown):0] [/usr/lib64/libpython2.7.so.1.0]
BACKTRACE(1) 12	[addr=<0x2abcca0aa88f>] [(unknown):0] [/usr/lib64/libpython2.7.so.1.0]
BACKTRACE(1) 13	[addr=<0x2abcca0ac1dd>] [(unknown):0] [/usr/lib64/libpython2.7.so.1.0]
BACKTRACE(1) 14	[addr=<0x2abcca0ac2e2>] [(unknown):0] [/usr/lib64/libpython2.7.so.1.0]
BACKTRACE(1) 15	[addr=<0x2abcca0c571f>] [(unknown):0] [/usr/lib64/libpython2.7.so.1.0]
BACKTRACE(1) 16	[addr=<0x2abcca0c68de>] [(unknown):0] [/usr/lib64/libpython2.7.so.1.0]
BACKTRACE(1) 17	[addr=<0x2abcca0c7b69>] [(unknown):0] [/usr/lib64/libpython2.7.so.1.0]
BACKTRACE(1) 18	[addr=<0x2abcca0d8b7f>] [(unknown):0] [/usr/lib64/libpython2.7.so.1.0]
BACKTRACE(1) 19	[addr=<2abccacdcd65>] [/usr/lib64/libc-2.18.so:0] [/usr/lib64/libc-2.18.so]
BACKTRACE(1) 20	[addr=<0x400721>] [(unknown):0] [/usr/bin/python2.7]
CPU Cores	4
CPU MHz	1200.234
CPU Type	Intel(R) Xeon(R) CPU E5-2603 v2 @ 1.80GHz
CPU Vendor	GenuineIntel
CWD	/wopr/users/livetau/jlinford/workshop-python/06_debugging
Cache Size	10240 KB
Command Line	python /usr/local/packages/tau/x86_64/bin/tau_python samarcrun.py

Python Performance Evaluation

HANDS-ON: TAU AND IPYTHON

TAU in IPython Notebook



```
import tau
tau.run('fixedgrid()')
tau.writeProfiles()
```

```
plume at (25, 25)
TIME: 0
TIME: 3600
done
```

Python Performance Evaluation

CONCLUSION

Download TAU from U. Oregon

http://tau.uoregon.edu

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

Free download, open source, BSD license

Acknowledgements

- Department of Energy
  - Office of Science
  - Argonne National Laboratory
  - Oak Ridge National Laboratory
  - NNSA/ASC Trilabs (SNL, LLNL, LANL)
- HPCMP DoD PETTT Program
- National Science Foundation
  - Glassbox, SI-2
- University of Tennessee
- University of New Hampshire
  - Jean Perez, Benjamin Chandran
- University of Oregon
  - Allen D. Malony, Sameer Shende
  - Kevin Huck, Wyatt Spear
- TU Dresden
  - Holger Brunst, Andreas Knupfer
  - Wolfgang Nagel
- Research Centre Jülich
  - Bernd Mohr
  - Felix Wolf

TAU Performance System

REFERENCE

Online References

- PAPI:
  - PAPI documentation is available from the PAPI website:

    http://icl.cs.utk.edu/papi/
- TAU:
  - TAU Users Guide and papers available from the TAU website:
    http://tau.uoregon.edu/
- VAMPIR:
  - VAMPIR website:

    http://www.vampir.eu/
- Scalasca:
  - Scalasca documentation page:

    http://www.scalasca.org/
- Eclipse PTP:
  - Documentation available from the Eclipse PTP website:

    http://www.eclipse.org/ptp/

Compiling Fortran Codes with TAU

- **If your Fortran code uses free format in .f files (fixed is default for .f):**
  % export TAU_OPTIONS='-optPdtF95Opts="-R free" -optVerbose'

- **To use the compiler based instrumentation instead of PDT (source-based):**
  % export TAU_OPTIONS='-optCompInst -optVerbose'

- **If your Fortran code uses C preprocessor directives (#include, #ifdef, #endif):**
  % export TAU_OPTIONS='-optPreProcess –optVerbose'

- **To use an instrumentation specification file:**
  % export TAU_OPTIONS=
  '-optTauSelectFile=select.tau -optVerbose -optPreProcess’

Example select.tau file

```
BEGIN_INSTRUMENT_SECTION
loops file="*" routine="#"
memory file="foo.f90" routine=”#"
io file="abc.f90" routine="FOO"
END_INSTRUMENT_SECTION
```

Generate a PAPI profile with 2 or more counters

```
% export TAU_MAKEFILE=$TAU/Makefile.tau-papi-mpi-pdt
% export TAU_OPTIONS=‘-optTauSelectFile=select.tau –optVerbose’
% cat select.tau
  BEGIN_INSTRUMENT_SECTION
  loops routine=“#”
  END_INSTRUMENT_SECTION

% export PATH=$TAU_ROOT/bin:$PATH
% make F90=tau_f90.sh
(Or edit Makefile and change F90=tau_f90.sh)
%
% export TAU_METRICS=TIME:PAPI_FP_INS:PAPI_L1_DCM
% mpirun -np 4 ./a.out
% paraprof --pack app.ppk
  Move the app.ppk file to your desktop.
% paraprof app.ppk
  Choose Options -> Show Derived Metrics Panel -> “PAPI_FP_INS”, click
   “/”, “TIME”, click “Apply”  and choose the derived metric.
```

Tracking I/O

```
% export TAU_MAKEFILE=$TAU/Makefile.tau-papi-mpi-pdt
% export PATH=$TAU_ROOT/bin:$PATH
% export TAU_OPTIONS=‘-optTrackIO -optVerbose’
% make CC=tau_cc.sh CXX=tau_cxx.sh F90=tau_f90.sh
% mpirun -n 4 ./a.out
% paraprof -pack ioprofile.ppk
% export TAU_TRACK_IO_PARAMS=1
% mpirun -n 4 ./a.out
```

Installing and Configuring TAU

- Installing PDT:
  - wget http://tau.uoregon.edu/pdt.tgz
  - ./configure –prefix=<dir>; make ; make install
- Installing TAU:
  - wget http://tau.uoregon.edu/tau.tgz
  - ./configure -bfd=download -pdt=<dir> -papi=<dir> ...
  - make install
- Using TAU:
  - export TAU_MAKEFILE=<taudir>/<arch>/lib/Makefile.tau-<TAGS>
  - make CC=tau_cc.sh   CXX=tau_cxx.sh   F90=tau_f90.sh

Compile-Time Options (TAU_OPTIONS)

% tau_compiler.sh

-optVerbose	Turn on verbose debugging messages
-optCompInst	Use compiler based instrumentation
-optNoCompInst	Do not revert to compiler instrumentation if source instrumentation fails.
-optTrackIO	Wrap POSIX I/O call and calculates vol/bw of I/O operations
-optMemDbg	Runtime bounds checking (see TAU_MEMDBG_* env vars)
-optKeepFiles	Does not remove intermediate .pdb and .inst.* files
-optPreProcess	Preprocess sources (OpenMP, Fortran) before instrumentation
-optTauSelectFile=”<file>"	Specify selective instrumentation file for *tau_instrumentor*
-optTauWrapFile=”<file>"	Specify path to *link_options.tau* generated by *tau_gen_wrapper*
-optHeaderInst	Enable Instrumentation of headers
-optTrackUPCR	Track UPC runtime layer routines (used with tau_upc.sh)
-optPdtF95Opts=""	Add options for Fortran parser in PDT (f95parse/gfparse) …

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_LEAKS	0	Setting to 1 turns on leak detection (for use with –optMemDbg or tau_exec)
TAU_MEMDBG_PROTECT_ABOVE	0	Setting to 1 turns on bounds checking for dynamically allocated arrays. (Use with –optMemDbg or tau_exec –memory_debug).
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_TRACK_IO_PARAMS	0	Setting to 1 with –optTrackIO or tau_exec –io captures arguments of I/O calls
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. Enabled by default to remove 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_COMPENSATE	0	Setting to 1 enables runtime compensation of instrumentation overhead
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., TIME:P_VIRTUAL_TIME:PAPI_FP_INS:PAPI_NATIVE_<event>\\:<subevent>)