Profiling and Debugging with TAU Commander

Sameer Shende

University of Oregon and ParaTools, Inc.

sameer@cs.uoregon.edu

NCAR Center Green Campus, CG South Auditorium, Friday, April 6, 2018

Download slides from:

http://tau.uoregon.edu/sea18.pdf

Workshop hands-on:

% cp ~sshende/pkgs/workshop.tgz .

Acknowledgments

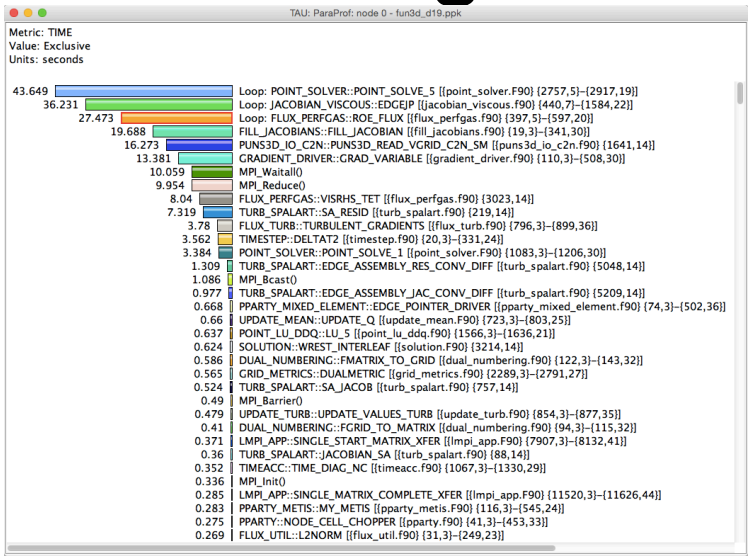
- University of Oregon
- ParaTools, Inc.
- The Ohio State University
- NSF (SI2, XSEDE)
- ParaTools, SAS, France
- U.S. Department of Energy
  - LLNL, LANL, SNL
  - ALCF/ANL, PNNL, ORNL, NERSC/LBL
- U.S. Department of Defense
- CEA, France
- NASA

Outline

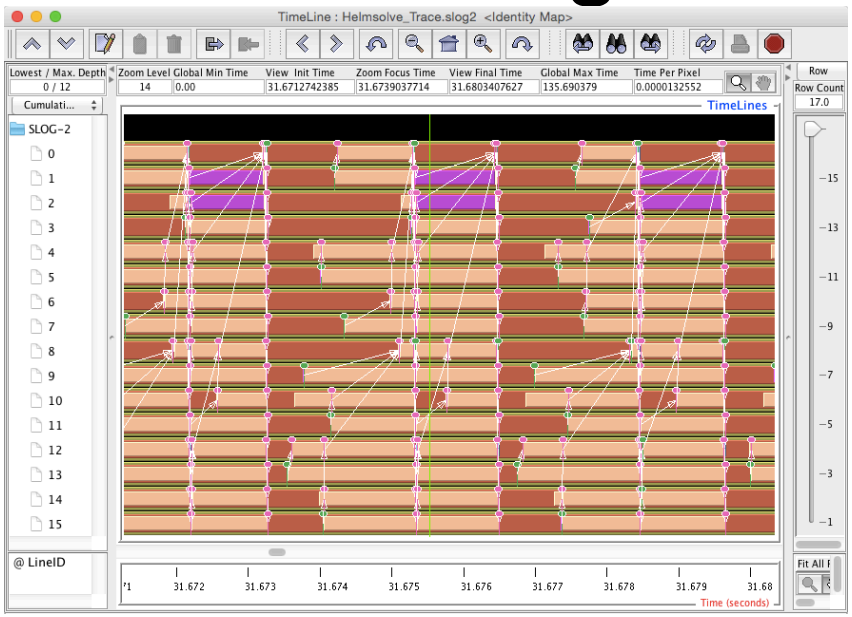
- Core concepts in performance evaluation tools
- Tools: PAPI, Vampir, Jumpshot, TAU
- Instrumentation: PDT, MPI, OpenMP OMPT
- Event-based sampling
- PDT based source instrumentation
- Selective instrumentation
- Compiler-based Instrumentation
- Event-tracing with Jumpshot and Vampir
- Introduction to TAU Commander
- Conclusions

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

Instrumentation

Direct and indirect performance observation

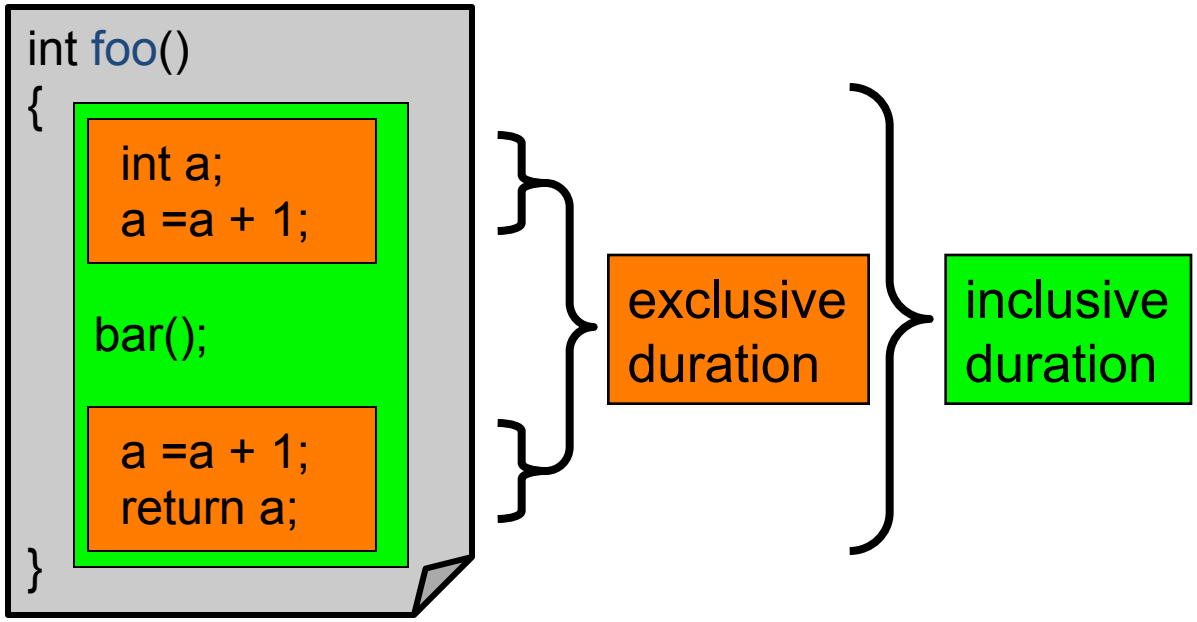
- Instrumentation invokes performance measurement
- Direct measurement with *probes*
- Indirect measurement with periodic sampling or hardware performance counter overflow interrupts
- Events measure performance data, metadata, context, etc.

User-defined events

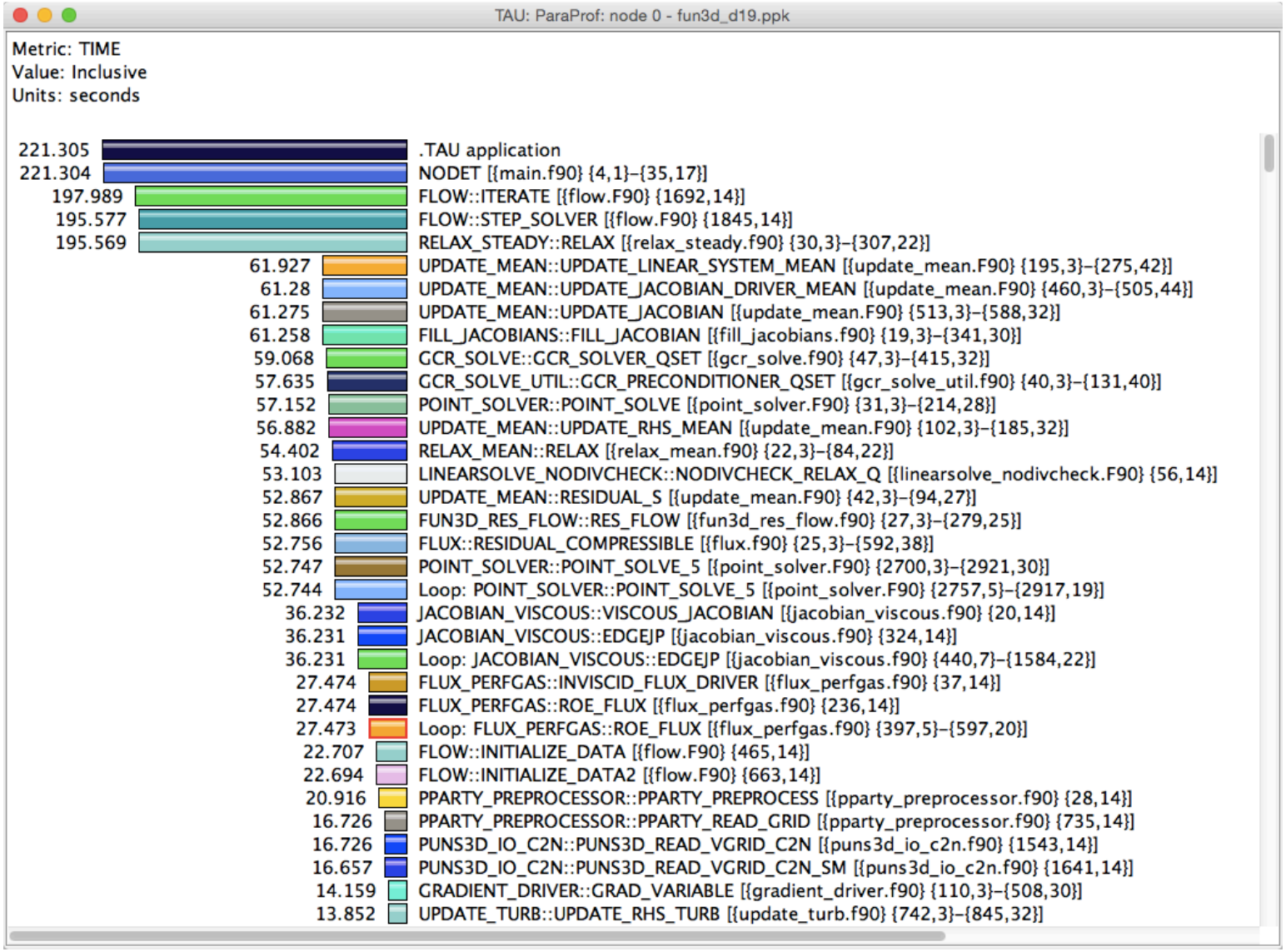
- ***Interval*** (start/stop) events to measure exclusive & inclusive duration
- ***Atomic events*** take measurements at a single point
  - Measures total, samples, min/max/mean/std. deviation statistics
- ***Context events*** are atomic events with executing context
  - Measures above statistics for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

Inclusive vs. Exclusive Measurements

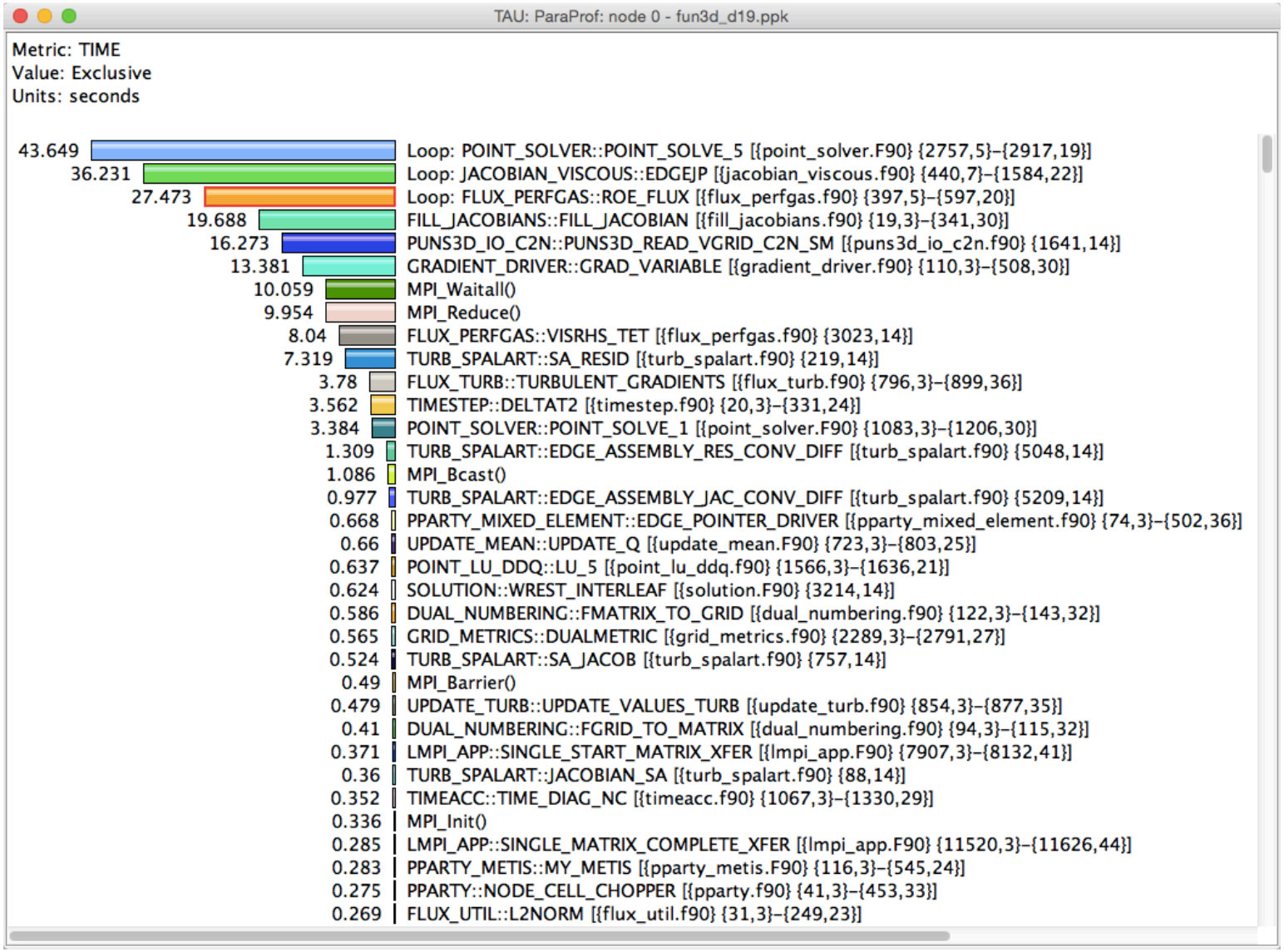
- Performance with respect to code regions
- Exclusive measurements for region only
- Inclusive measurements includes child regions



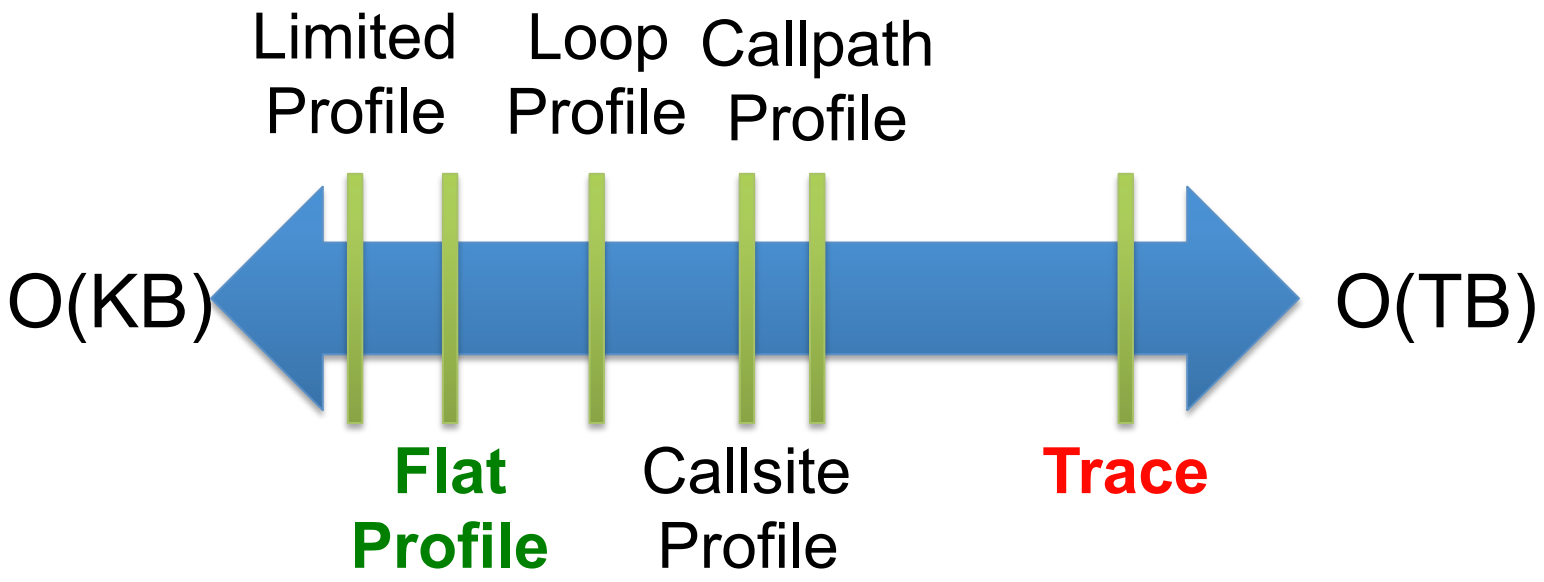
Inclusive Measurements



Exclusive Time



How much data do you want?



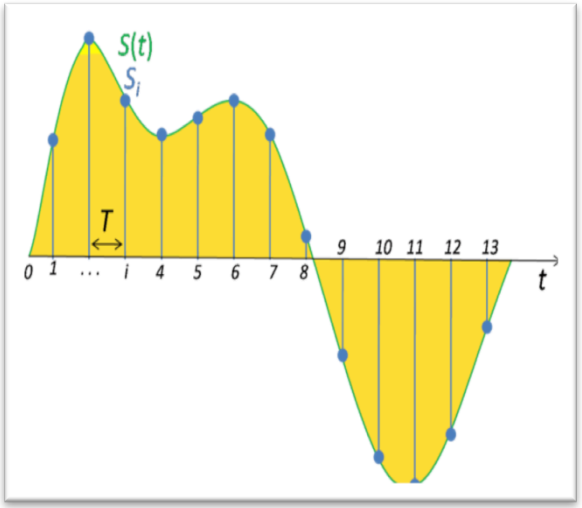
Performance Data Measurement

Direct via Probes

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

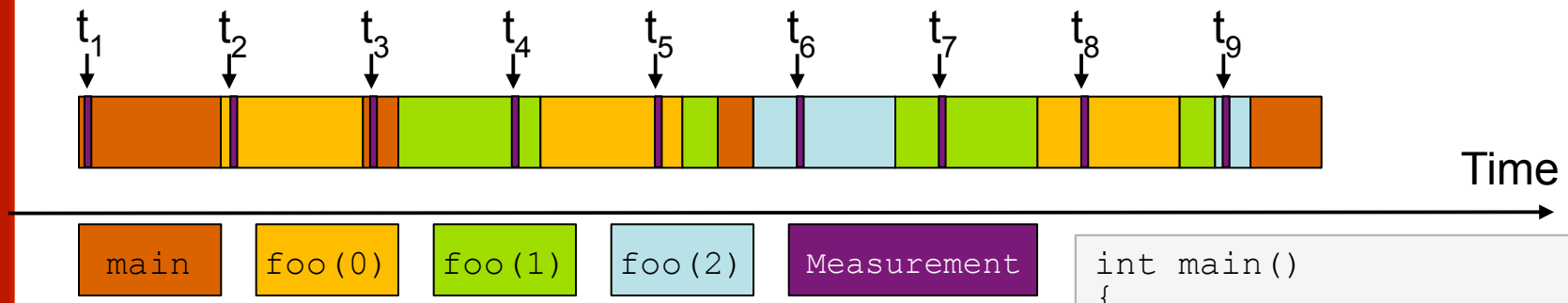
- Exact measurement
- Fine-grain control
- Calls inserted into code

Indirect via Sampling



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

Sampling



Running program is periodically interrupted to take measurement

- Timer interrupt, OS signal, or HWC overflow
- Service routine examines return-address stack
- Addresses are mapped to routines using symbol table information

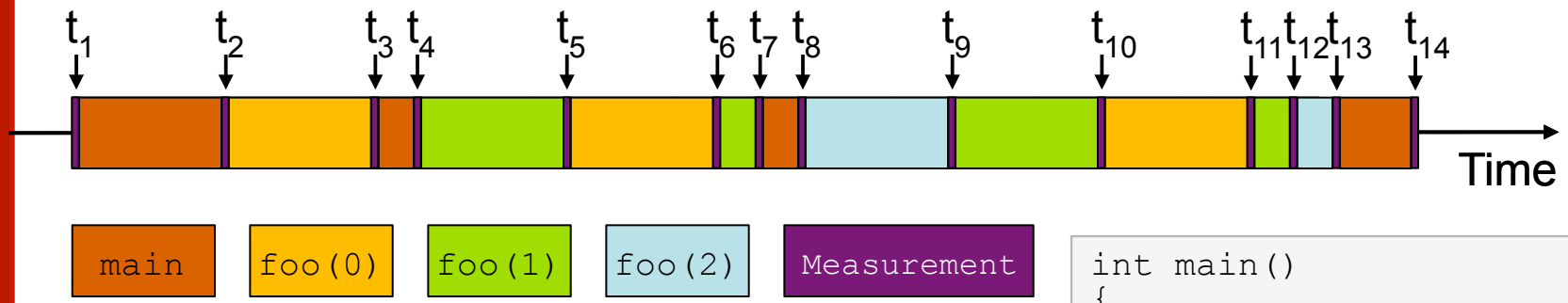
Statistical inference of program behavior

- Not very detailed information on highly volatile metrics
- Requires long-running applications

Works with unmodified executables

```
int main()
{
  int i;

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

  return 0;
}

void foo(int i)
{

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

}
```

Instrumentation



Measurement code is inserted such that every event of interest is captured directly

- Can be done in various ways

Advantage:

- Much more detailed information

Disadvantage:

- Processing of source-code / executable necessary
- Large relative overheads for small functions

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

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

Tools: PAPI

PAPI

- Consistent interface to hardware performance counters
- Preset and native events
- Higher level tools use PAPI
  - TAU, Score-P, HPCToolkit, VampirTrace, Open|SpeedShop
- API, Library, Tools
  - papi_avail - shows the list of preset counters
  - papi_native_avail – shows the list of native processor specific counters
  - papi_event_chooser – allows you to find a list of compatible events
  - papi_decode – shows how a preset event is comprised of native events
  - Components – perf, RAPL for energy, network counters
- ***From University of Tenessee, Knoxville***
  - http://icl.cs.utk.edu/papi

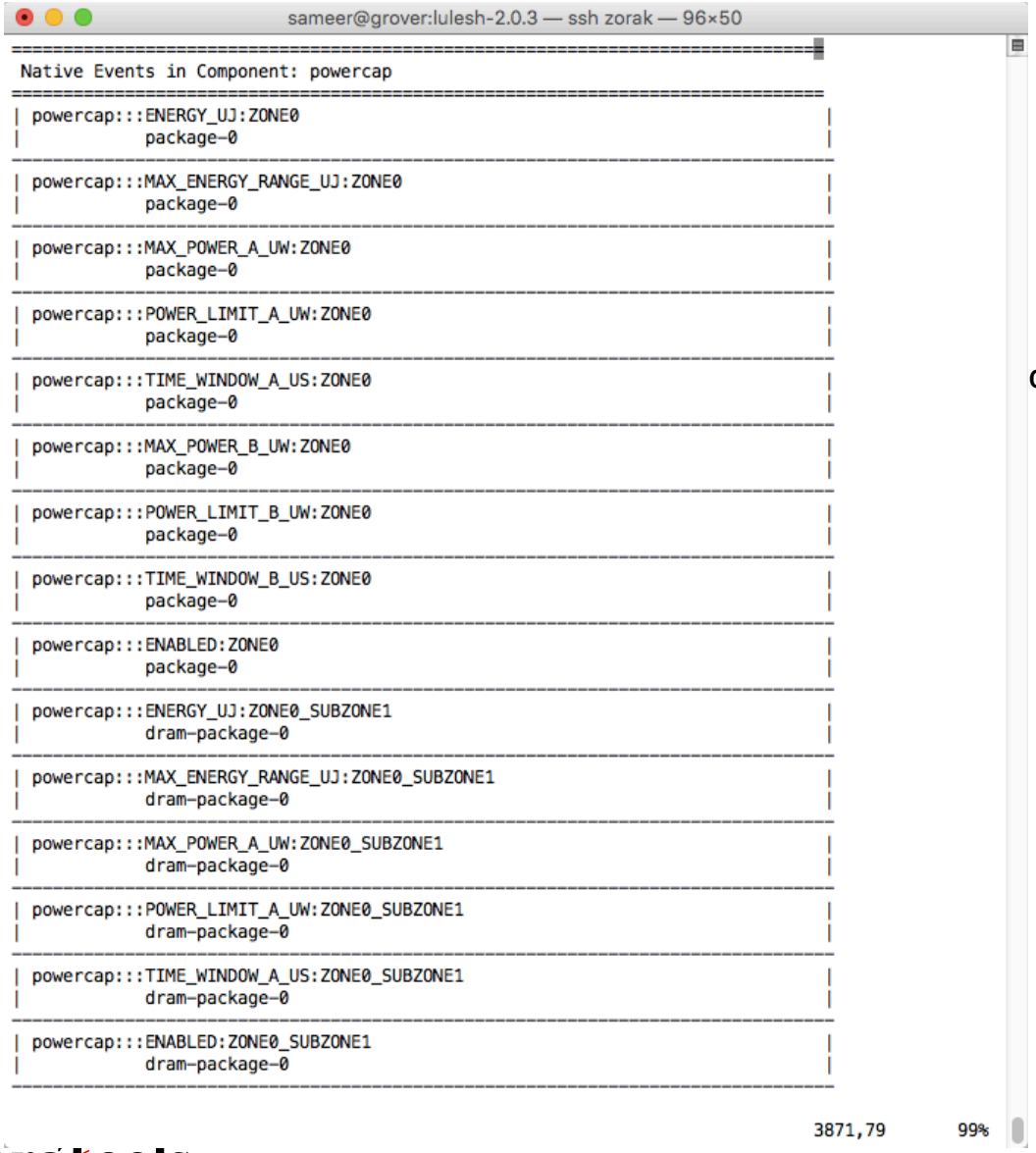
PAPI's Preset Cache Events on KNL

tg457572@c455-073.stampede2:~/tmp/mm — ssh stampede — 84×47

```
c455-073[knl](24)$ papi_avail --ca
Available PAPI preset and user defined events plus hardware information.
--------------------------------------------------------------------------------
PAPI Version             : 5.5.1.0
Vendor string and code   : GenuineIntel (1)
Model string and code    : Intel(R) Xeon Phi(TM) CPU 7250 @ 1.40GHz (87)
CPU Revision             : 1.000000
CPUID Info               : Family: 6  Model: 87  Stepping: 1
CPU Max Megahertz        : 1600
CPU Min Megahertz        : 1000
Hdw Threads per core     : 4
Cores per Socket         : 68
Sockets                  : 1
NUMA Nodes               : 1
CPUs per Node            : 272
Total CPUs               : 272
Running in a VM          : no
Number Hardware Counters : 5
Max Multiplex Counters   : 384
--------------------------------------------------------------------------------

================================================================================
  PAPI Preset Events
================================================================================
    Name        Code    Deriv Description (Note)
PAPI_L1_DCM  0x80000000  No   Level 1 data cache misses
PAPI_L1_ICM  0x80000001  No   Level 1 instruction cache misses
PAPI_L1_TCM  0x80000006  Yes  Level 1 cache misses
PAPI_L2_TCM  0x80000007  No   Level 2 cache misses
PAPI_TLB_DM  0x80000014  No   Data translation lookaside buffer misses
PAPI_L1_LDM  0x80000017  No   Level 1 load misses
PAPI_L2_LDM  0x80000019  No   Level 2 load misses
PAPI_STL_ICY 0x80000025  No   Cycles with no instruction issue
PAPI_BR_UCN  0x8000002a  Yes  Unconditional branch instructions
PAPI_BR_CN   0x8000002b  No   Conditional branch instructions
PAPI_BR_TKN  0x8000002c  No   Conditional branch instructions taken
PAPI_BR_NTK  0x8000002d  Yes  Conditional branch instructions not taken
PAPI_BR_MSP  0x8000002e  No   Conditional branch instructions mispredicted
PAPI_TOT_INS 0x80000032  No   Instructions completed
PAPI_LD_INS  0x80000035  No   Load instructions
PAPI_SR_INS  0x80000036  No   Store instructions
PAPI_BR_INS  0x80000037  No   Branch instructions
PAPI_RES_STL 0x80000039  No   Cycles stalled on any resource
PAPI_TOT_CYC 0x8000003b  No   Total cycles
PAPI_LST_INS 0x8000003c  Yes  Load/store instructions completed
PAPI_L1_DCA  0x80000040  Yes  Level 1 data cache accesses
PAPI_L1_ICH  0x80000049  No   Level 1 instruction cache hits
```

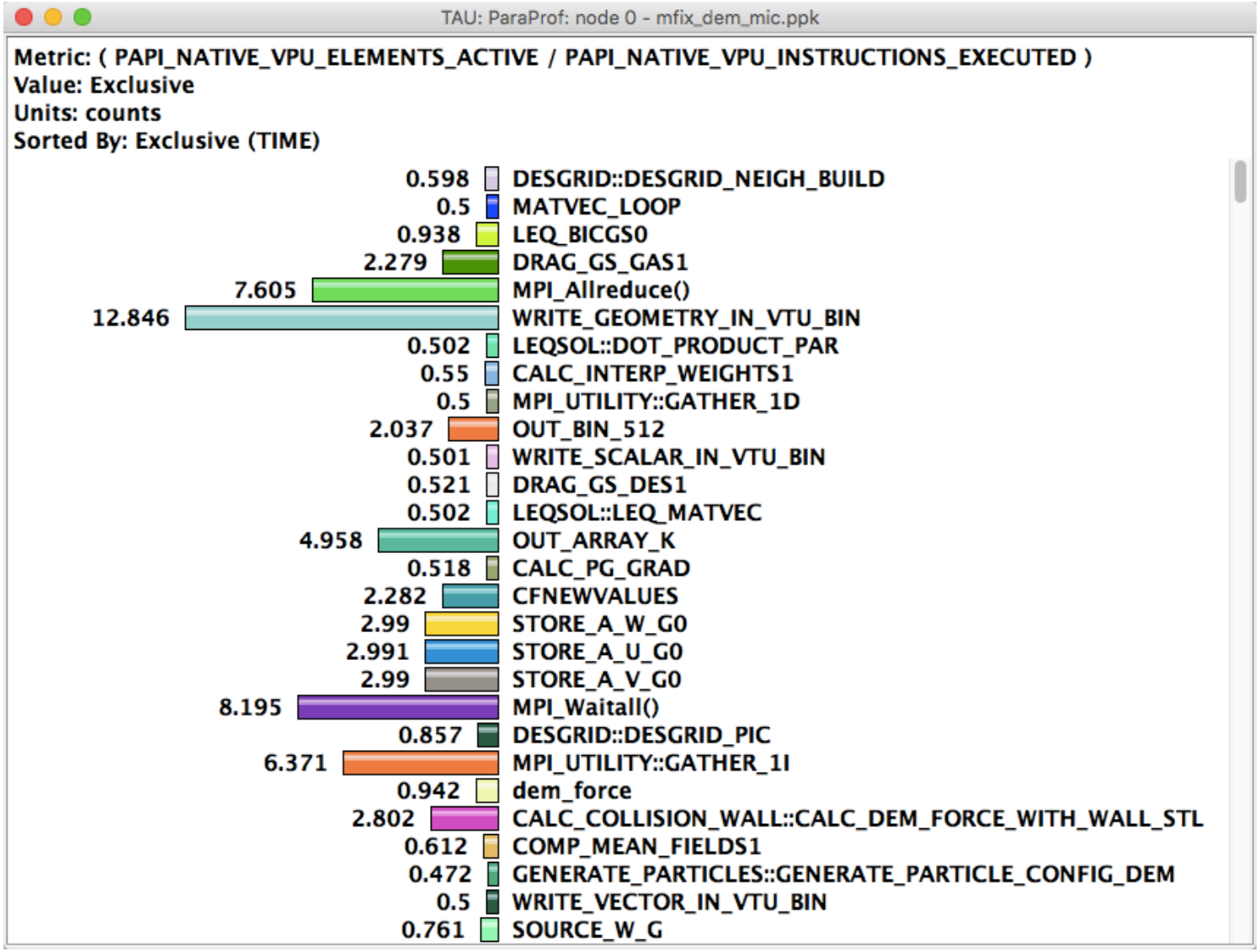
% papi_avail −ca

PAPI's Native Events for powercap

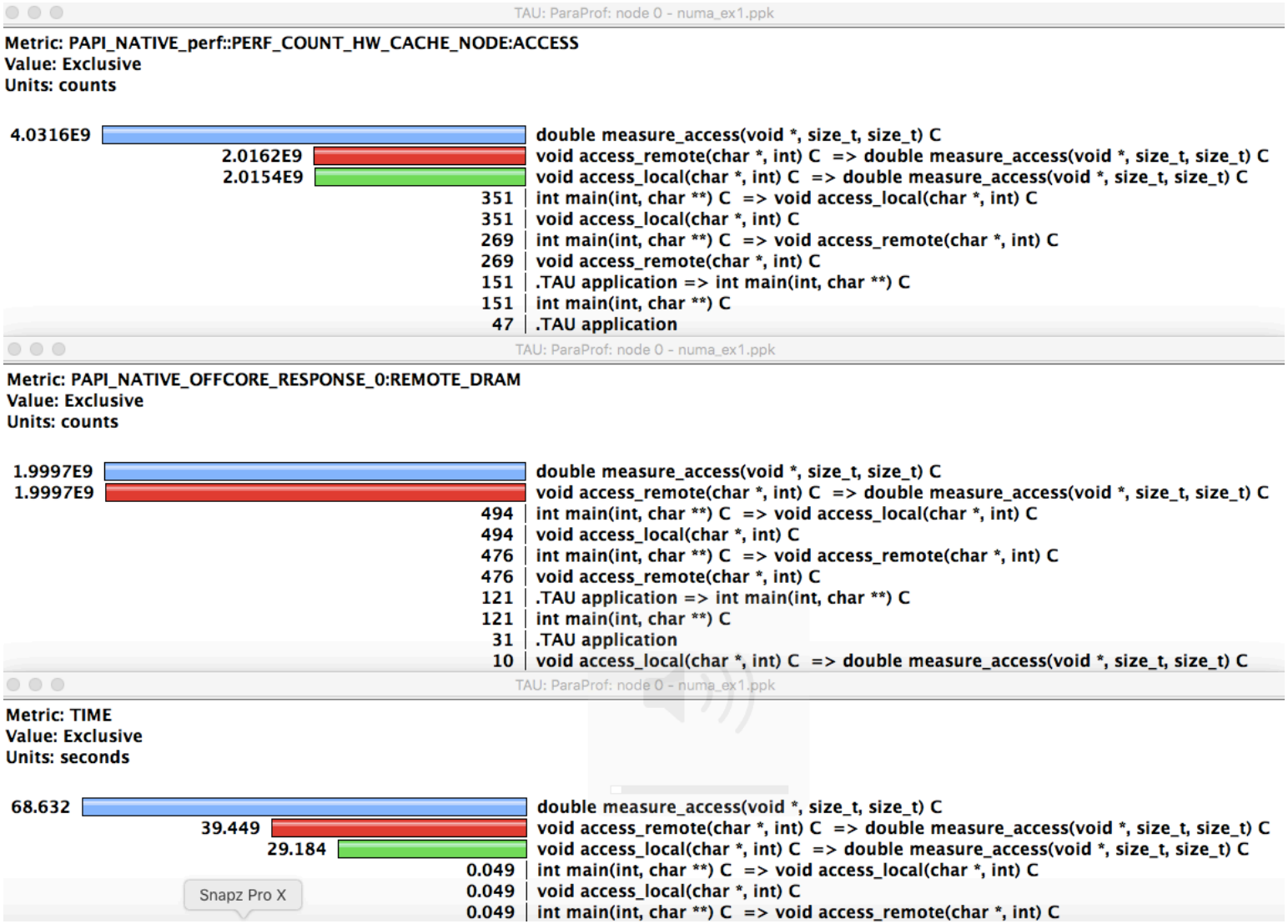


% papi_native_avail

TAU – PAPI Vectorization Intensity



TAU NUMA events from PAPI



Tools: Vampir

Vampir

Alternative and supplement to automatic analysis

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

Provide statistics and performance metrics

Timeline charts

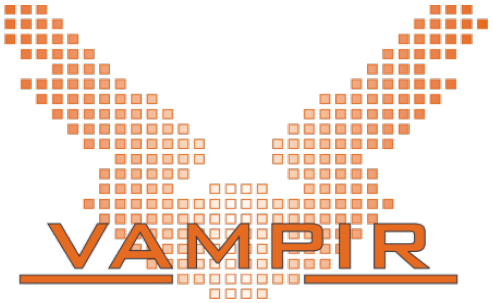
- **Show application activities and communication along a time axis**

Summary charts

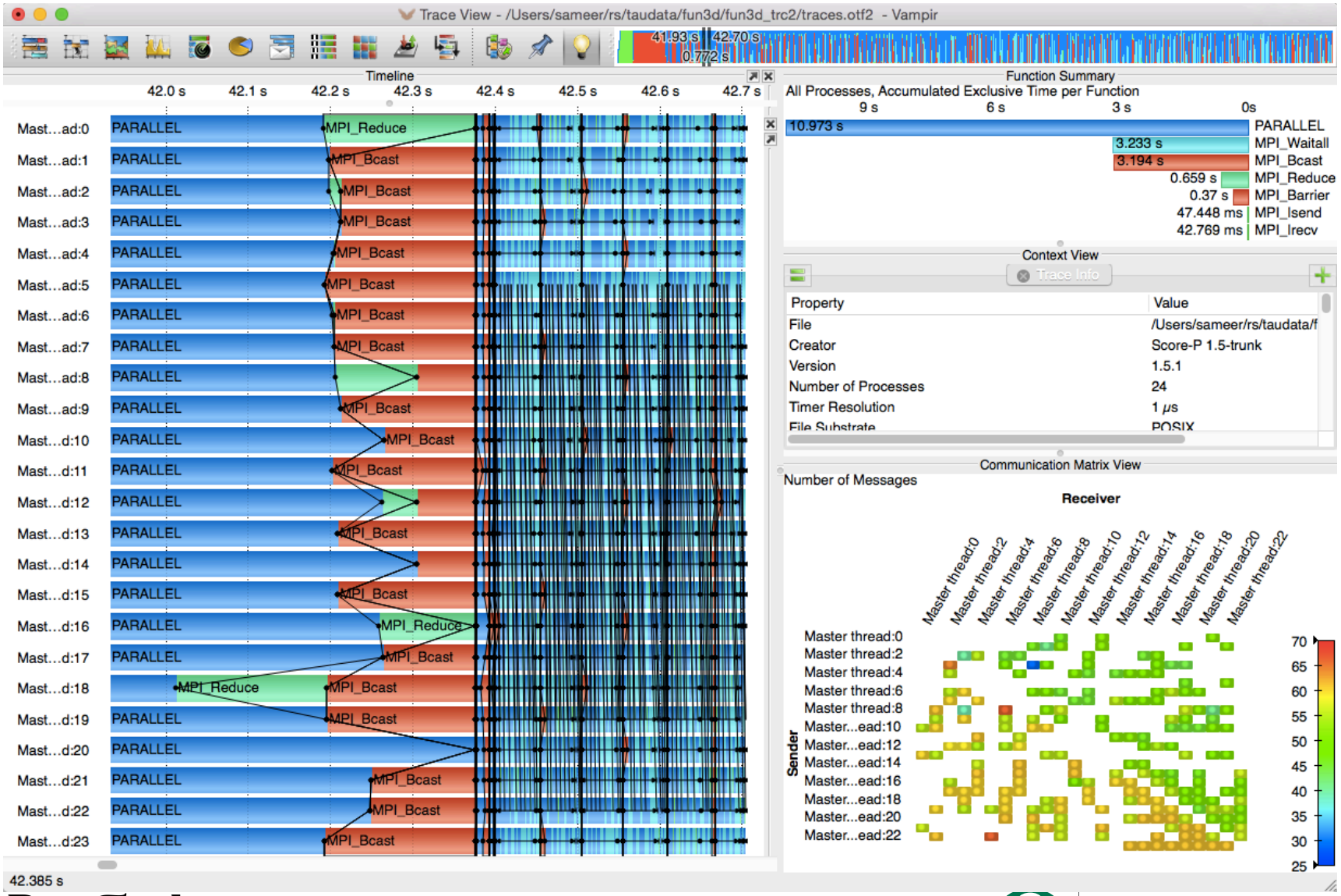
- **Provide quantitative results for the currently selected time interval**
- **Commercial trace visualization tool**

From TU Dresden, Germany

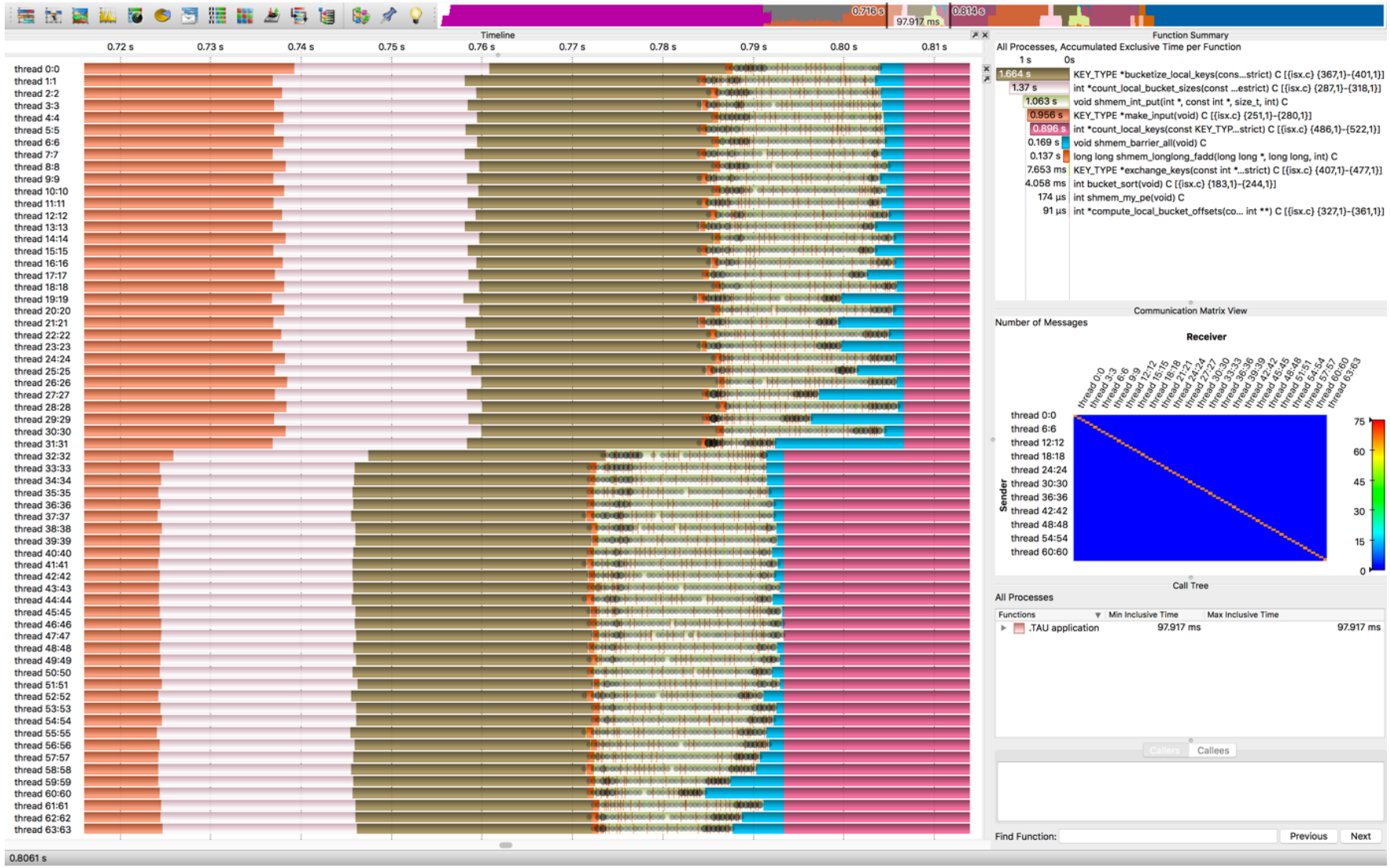
http://www.vampir.eu



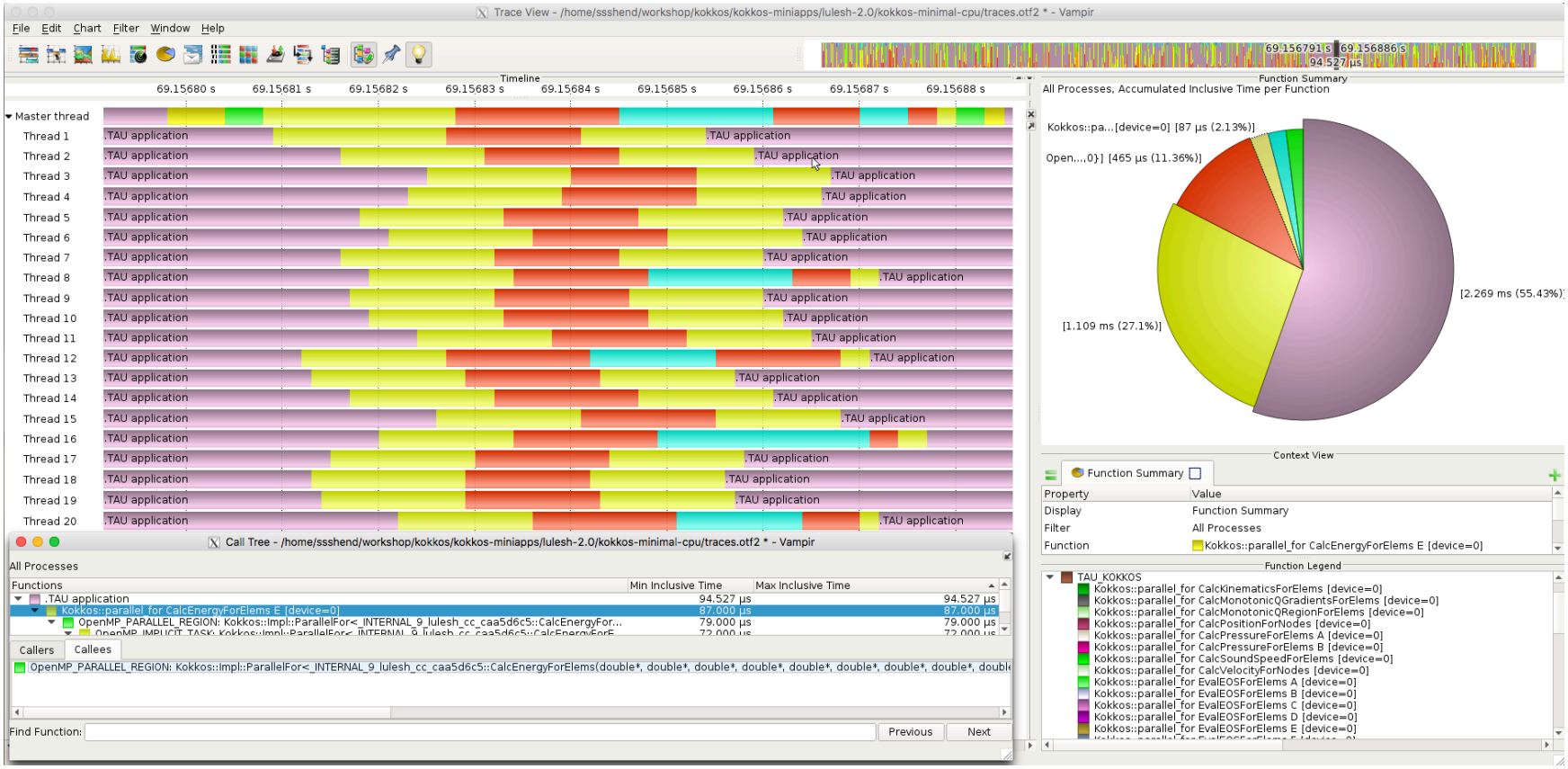
Vampir – Trace Visualization



Vampir – Trace Visualization



Vampir – TAU's Kokkos Profiling Interface



Tools: Jumpshot

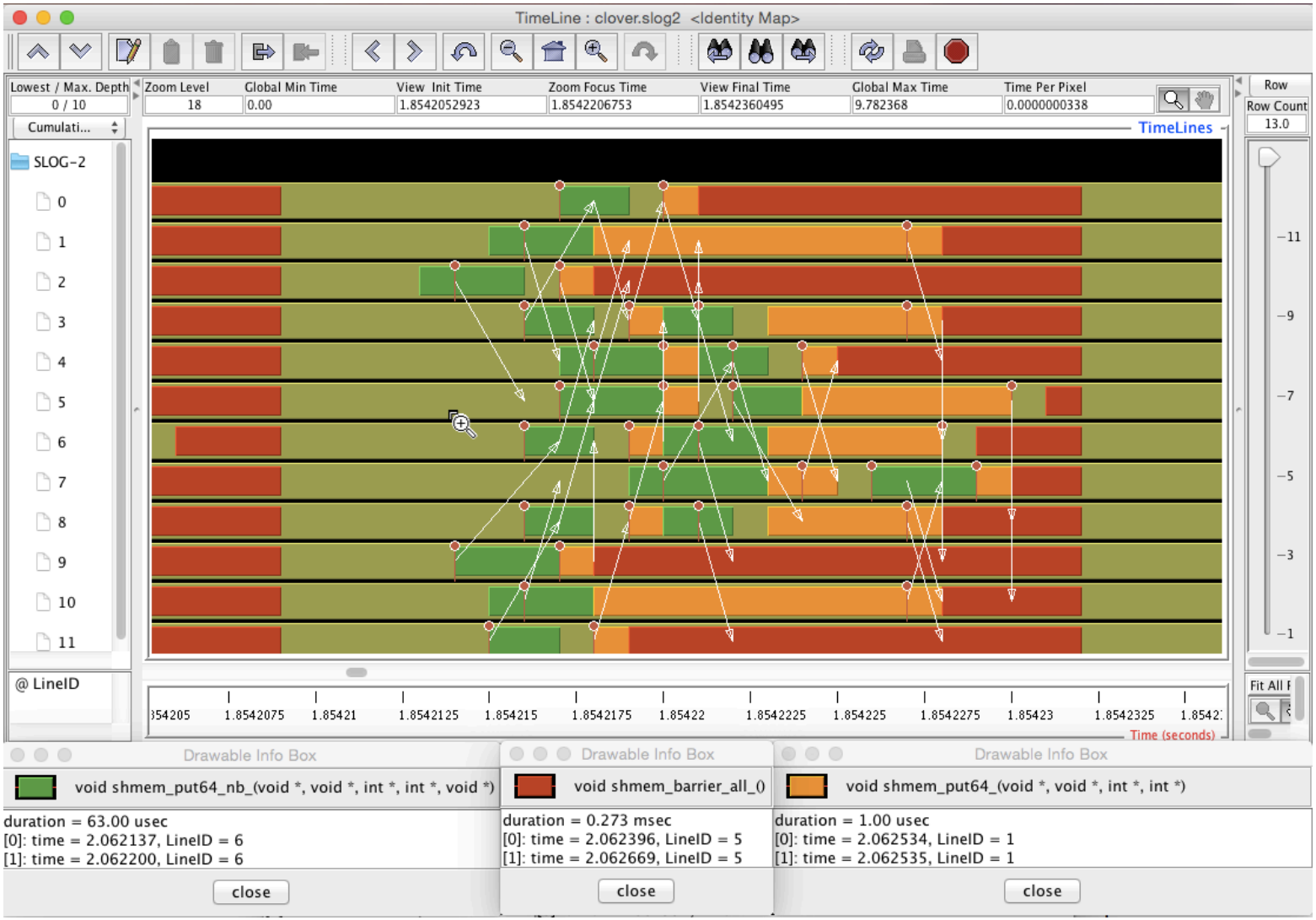
Jumpshot

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

Timeline charts

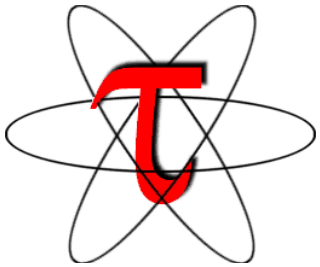
- Show application activities and communication along a time axis
- Shows boxes within boxes to show nesting of events

Jumpshot


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

Tools: TAU

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**
  - http://tau.uoregon.edu

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?

New Features in TAU

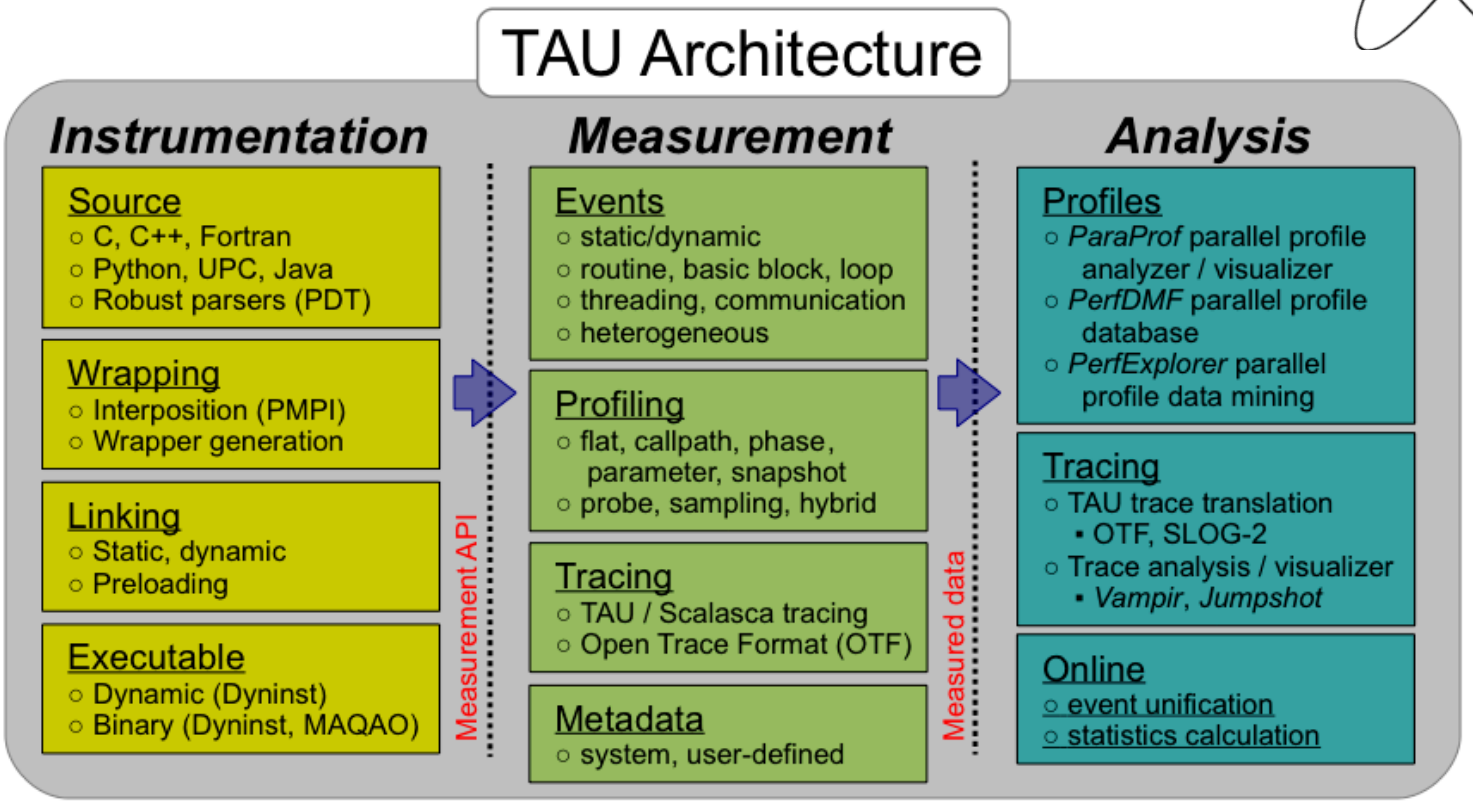
- Native OTF2 traces in TAU (TAU_TRACE_FORMAT=otf2)
- Callsite profiling (TAU_CALLSITE=1), also with tracing (TAU_TRACE=1)
- *tau_exec*: A tool to simplify TAU's usage
- Track system load (TAU_TRACK_LOAD=1) and memory footprint
- Support for accelerators: CUDA, OpenCL, Intel Xeon Phi Co-processor
- Event based sampling
- OpenMP instrumentation using OpenMP Tools Interface with Intel compilers (OMPT)
- Support for Intel TBB, CILK, and MPC [paratools.com/mpc]
- ParaProf 3D topology displays
- TAUdb using H2 database

What does TAU support?

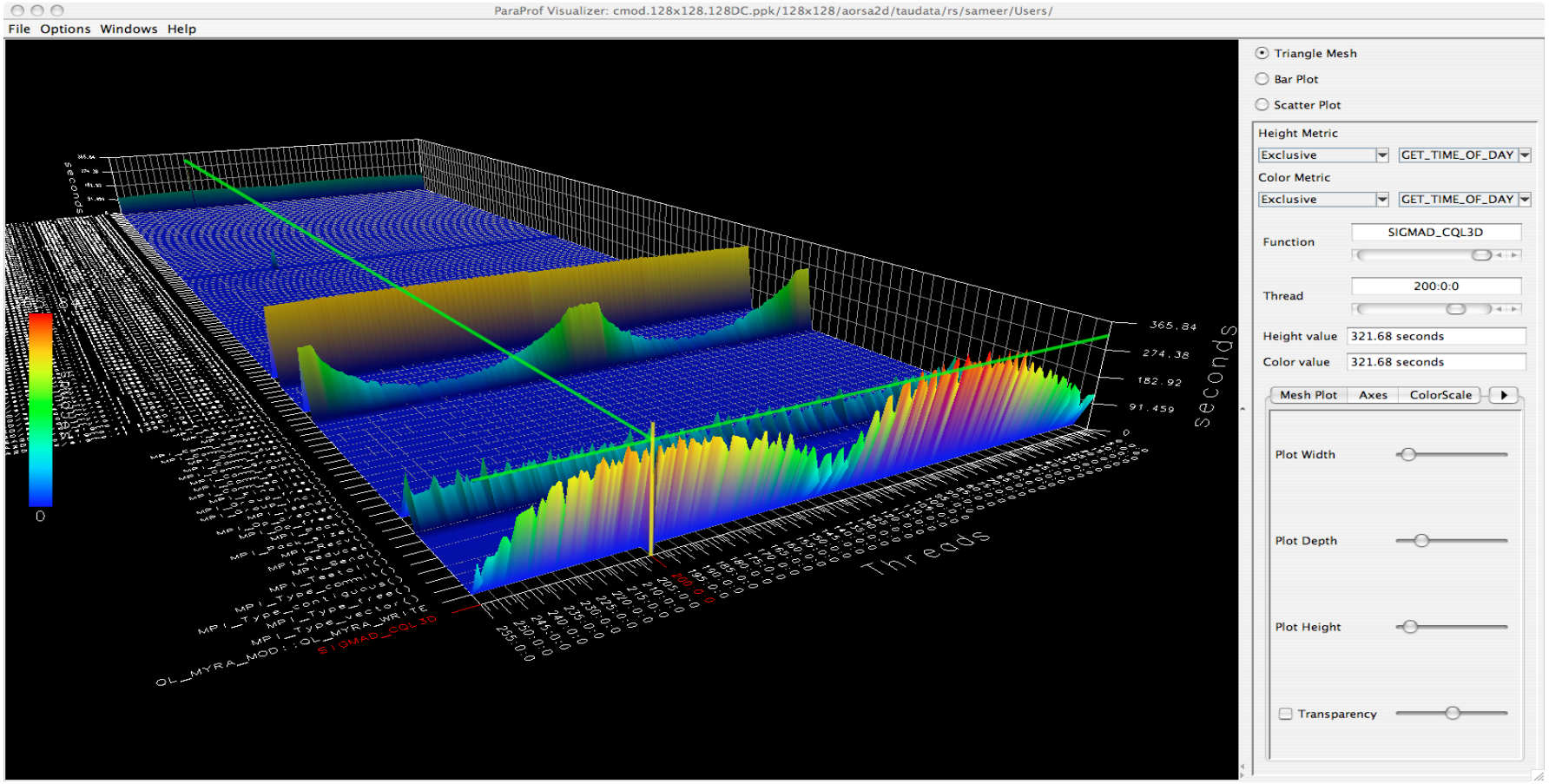
C/C++
Fortran
pthreads
Intel
GNU
MPC
CUDA
UPC
OpenACC
Intel MIC
LLVM
PGI
Linux
BlueGene
NVIDIA
OpenCL
Python
GPI
Java
MPI
OpenMP
Cray
Sun
AIX
Windows
Fujitsu
ARM64
Power 8
OS X

Insert yours here

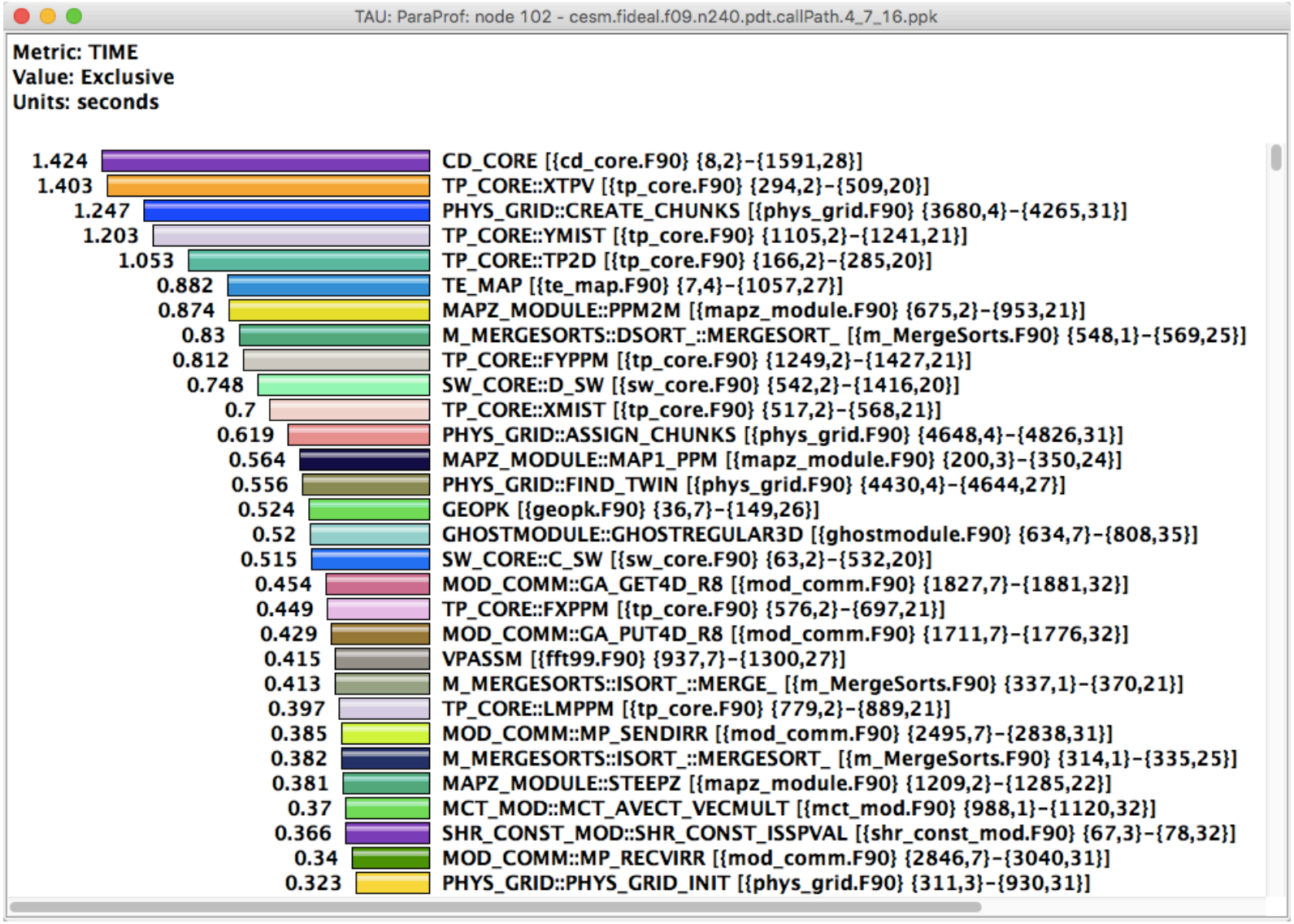
TAU Architecture and Workflow



ParaProf 3D Profile Browser



TAU – Flat Profile



TAU – Callsite Profiling

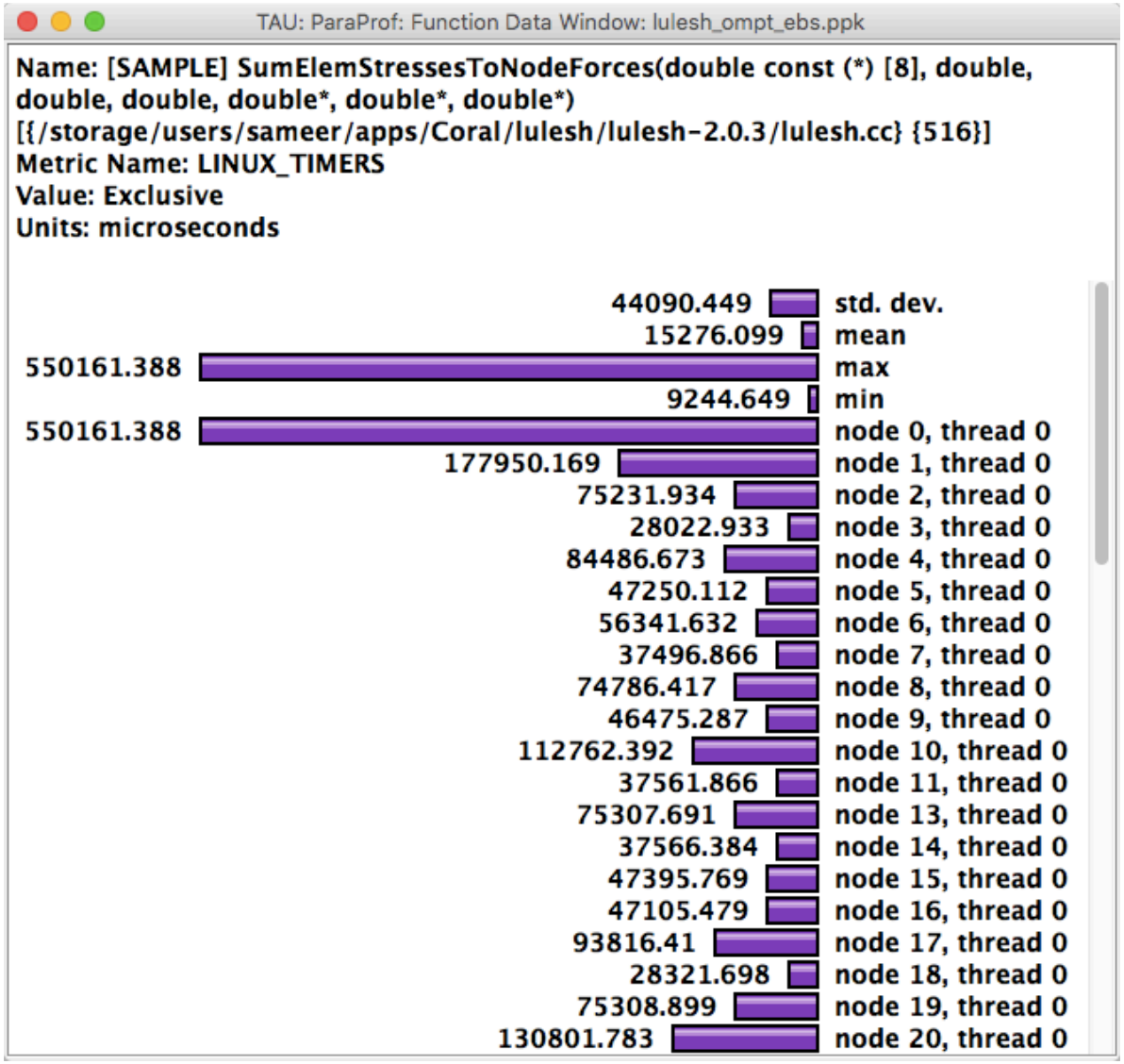
TAU: ParaProf: Statistics for: node 0 - clover_callsite.ppk

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

TAU – Callstack Sampling

Name	Inclusive...	Calls
.TAU application	34.979	1
[CONTEXT] .TAU application	31.647	632
void shmem_barrier_all_()	1.219	46,029
[CONTEXT] void shmem_barrier_all_()	1.599	32
[UNWIND] [/home/ssshend/CloverLeaf_OpenSHMEM/clover_leaf.f90.41] [@] UNRESOLVED /lib64/libc-2.11.3.so	1.599	32
[UNWIND] /home/ssshend/CloverLeaf_OpenSHMEM/hydro.f90.62 [@] main [{/home/ssshend/CloverLeaf_OpenSHMEM/clover_leaf.f90} {41}]	0.85	17
[UNWIND] /home/ssshend/CloverLeaf_OpenSHMEM/advection.f90.102 [@] hydro_ [{/home/ssshend/CloverLeaf_OpenSHMEM/hydro.f90} {62}]	0.55	11
[UNWIND] /home/ssshend/CloverLeaf_OpenSHMEM/update_halo.f90.36 [@] __advection_module_MOD_advection [{/home/ssshend/CloverLeaf_Ope	0.55	11
[UNWIND] /home/ssshend/CloverLeaf_OpenSHMEM/clover.f90.292 [@] __update_halo_module_MOD_update_halo [{/home/ssshend/CloverLeaf_O	0.5	10
[UNWIND] /home/ssshend/CloverLeaf_OpenSHMEM/clover.f90.572 [@] __clover_module_MOD_clover_exchange [{/home/ssshend/CloverLeaf_(	0.5	10
[UNWIND] UNRESOLVED [@] __clover_module_MOD_clover_exchange_message [{/home/ssshend/CloverLeaf_OpenSHMEM/clover.f90} {572}]	0.5	10
[UNWIND] [/notbackedup/tmp/ulib/mpt/nightly/7.2/062215-RC/sma_dmapp/src/shmem_opt_barrier.c.118] [@] UNRESOLVED /nfsproje	0.45	9
[SAMPLE] _smai_smp_barrier_in [{/notbackedup/tmp/ulib/mpt/nightly/7.2/062215-RC/sma_dmapp/src/shmem_opt_barrier.c} {118}]	0.45	9
[UNWIND] [/notbackedup/tmp/ulib/mpt/nightly/7.2/062215-RC/sma_dmapp/src/shmem_internal.h.88] [@] UNRESOLVED /nfsprojects/v	0.05	1
[UNWIND] /home/ssshend/CloverLeaf_OpenSHMEM/clover.f90.461 [@] __update_halo_module_MOD_update_halo [{/home/ssshend/CloverLeaf_O	0.05	1
[UNWIND] /home/ssshend/CloverLeaf_OpenSHMEM/advection.f90.72 [@] hydro_ [{/home/ssshend/CloverLeaf_OpenSHMEM/hydro.f90} {62}]	0.15	3
[UNWIND] /home/ssshend/CloverLeaf_OpenSHMEM/advection.f90.55 [@] hydro_ [{/home/ssshend/CloverLeaf_OpenSHMEM/hydro.f90} {62}]	0.15	3
[UNWIND] /home/ssshend/CloverLeaf_OpenSHMEM/hydro.f90.52 [@] main [{/home/ssshend/CloverLeaf_OpenSHMEM/clover_leaf.f90} {41}]	0.5	10
[UNWIND] /home/ssshend/CloverLeaf_OpenSHMEM/hydro.f90.54 [@] main [{/home/ssshend/CloverLeaf_OpenSHMEM/clover_leaf.f90} {41}]	0.25	5
void start_pes_(int *)	0.508	1
void shmem_real8_max_to_all_(void *, void *, int *, int *, int *, int *, void *, long *)	0.325	2,000
[CONTEXT] void shmem_real8_max_to_all_(void *, void *, int *, int *, int *, int *, void *, long *)	0.5	10
[UNWIND] [/home/ssshend/CloverLeaf_OpenSHMEM/clover_leaf.f90.41] [@] UNRESOLVED /lib64/libc-2.11.3.so	0.5	10
[UNWIND] /home/ssshend/CloverLeaf_OpenSHMEM/hydro.f90.58 [@] main [{/home/ssshend/CloverLeaf_OpenSHMEM/clover_leaf.f90} {41}]	0.45	9
[UNWIND] /home/ssshend/CloverLeaf_OpenSHMEM/PdV.f90.107 [@] hydro_ [{/home/ssshend/CloverLeaf_OpenSHMEM/hydro.f90} {58}]	0.45	9
[UNWIND] /home/ssshend/CloverLeaf_OpenSHMEM/clover.f90.740 [@] __pdv_module_MOD_pdv [{/home/ssshend/CloverLeaf_OpenSHMEM/PdV.f90	0.45	9
[UNWIND] UNRESOLVED [@] __clover_module_MOD_clover_check_error [{/home/ssshend/CloverLeaf_OpenSHMEM/clover.f90} {740}]	0.45	9
[UNWIND] [/notbackedup/tmp/ulib/mpt/nightly/7.2/062215-RC/sma_dmapp/src/shmem_reduction.h.207] [@] UNRESOLVED /nfsprojects/vol	0.45	9
[UNWIND] /notbackedup/tmp/ulib/mpt/nightly/7.2/062215-RC/sma_dmapp/src/shmem_opt_reduction.h.788 [@] pshmem_double_max_tc	0.45	9
[UNWIND] /notbackedup/tmp/ulib/mpt/nightly/7.2/062215-RC/sma_dmapp/src/shmem_opt_reduction.h.107 [@] _smai_opt_double_ma	0.45	9
[SAMPLE] _smai_smp_reduce_double_max [{/notbackedup/tmp/ulib/mpt/nightly/7.2/062215-RC/sma_dmapp/src/shmem_opt_reduct	0.45	9
[UNWIND] /home/ssshend/CloverLeaf_OpenSHMEM/hydro.f90.54 [@] main [{/home/ssshend/CloverLeaf_OpenSHMEM/clover_leaf.f90} {41}]	0.05	1

TAU – Event Based Sampling

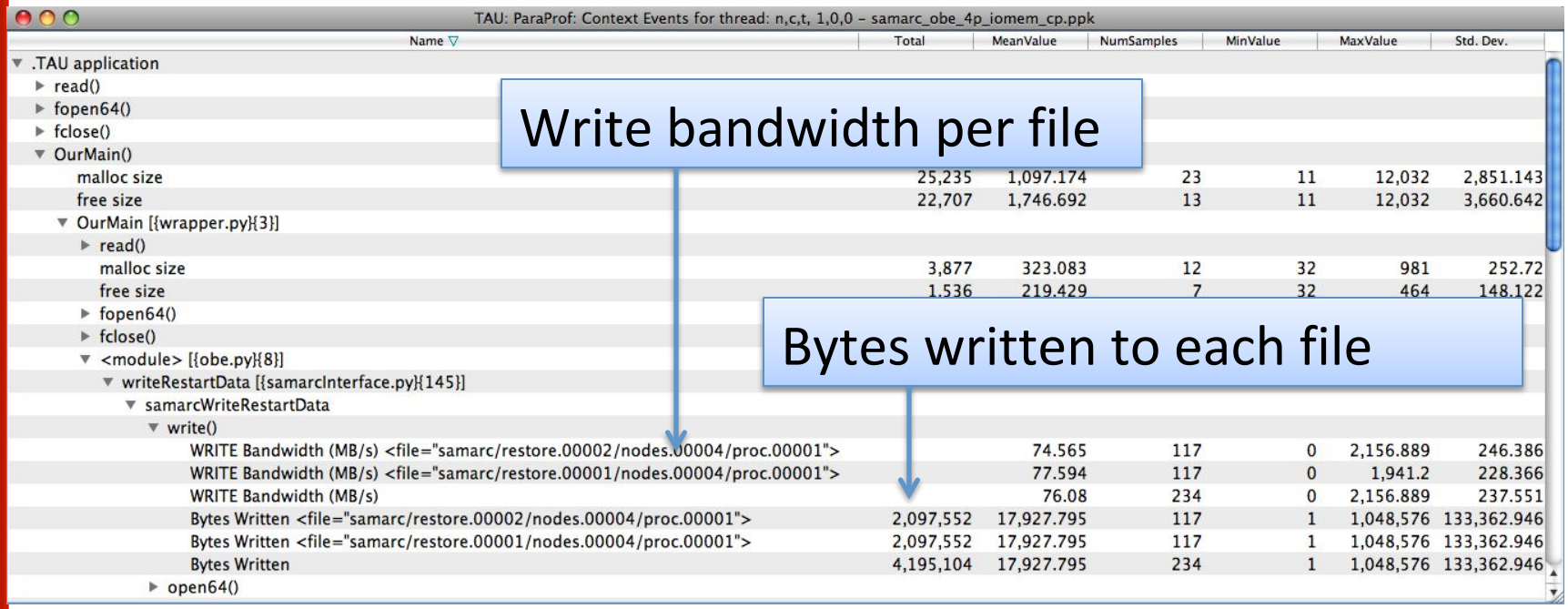


TAU Atomic Events

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

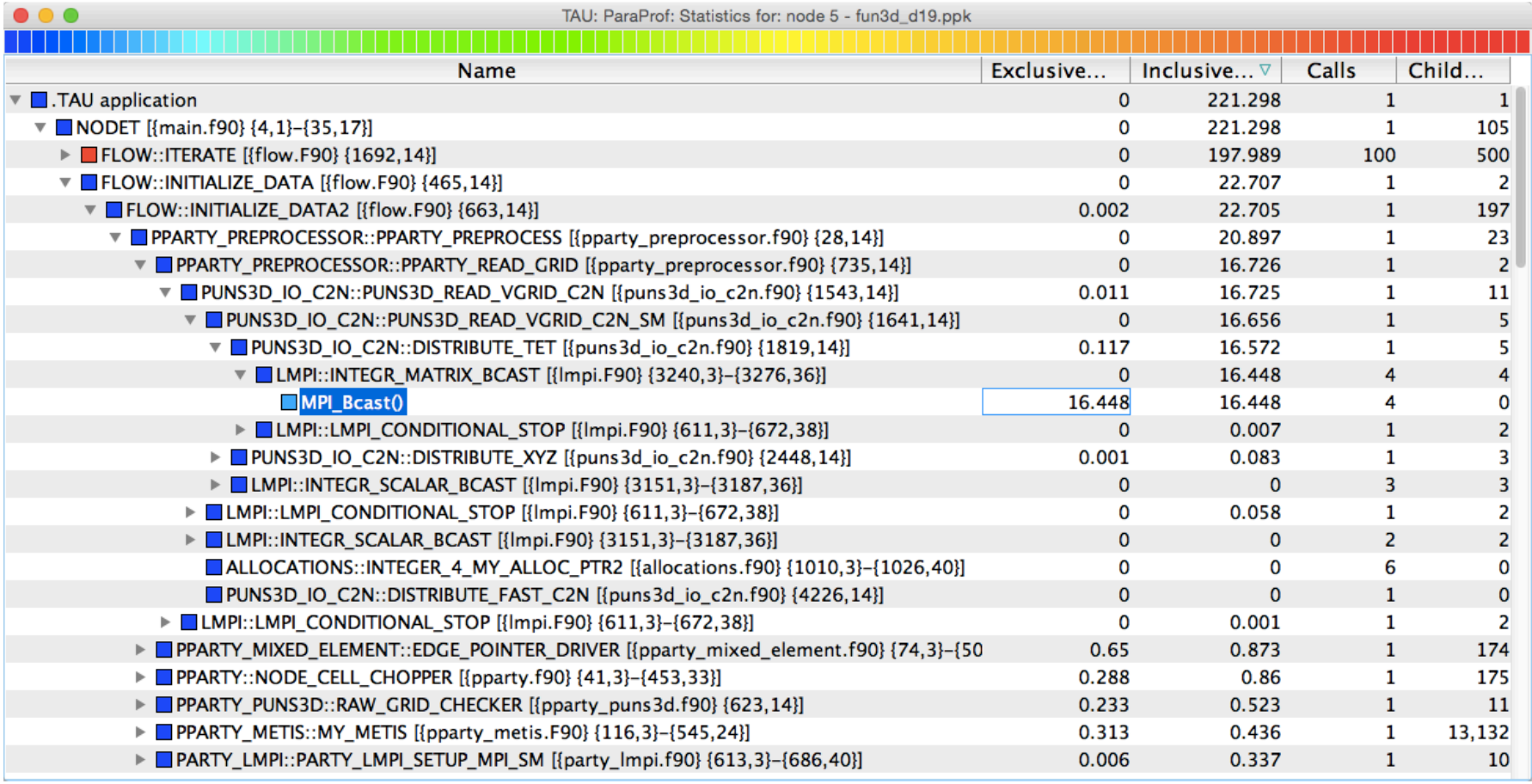
Name	Total	NumSamples	MaxValue	MinValue	MeanValue	Std. Dev.
Bytes Written <file=stdout>	911	62	21	1	14.694	7.441
Bytes Written <file=pipe>	22	22	1	1	1	0
Bytes Written <file=Process_Output/VelRsdl.dat>	7,826	100	302	76	78.26	22.487
Bytes Written <file=Process_Output/MomRsdl.dat>	7,826	100	302	76	78.26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

TAU – Context Events



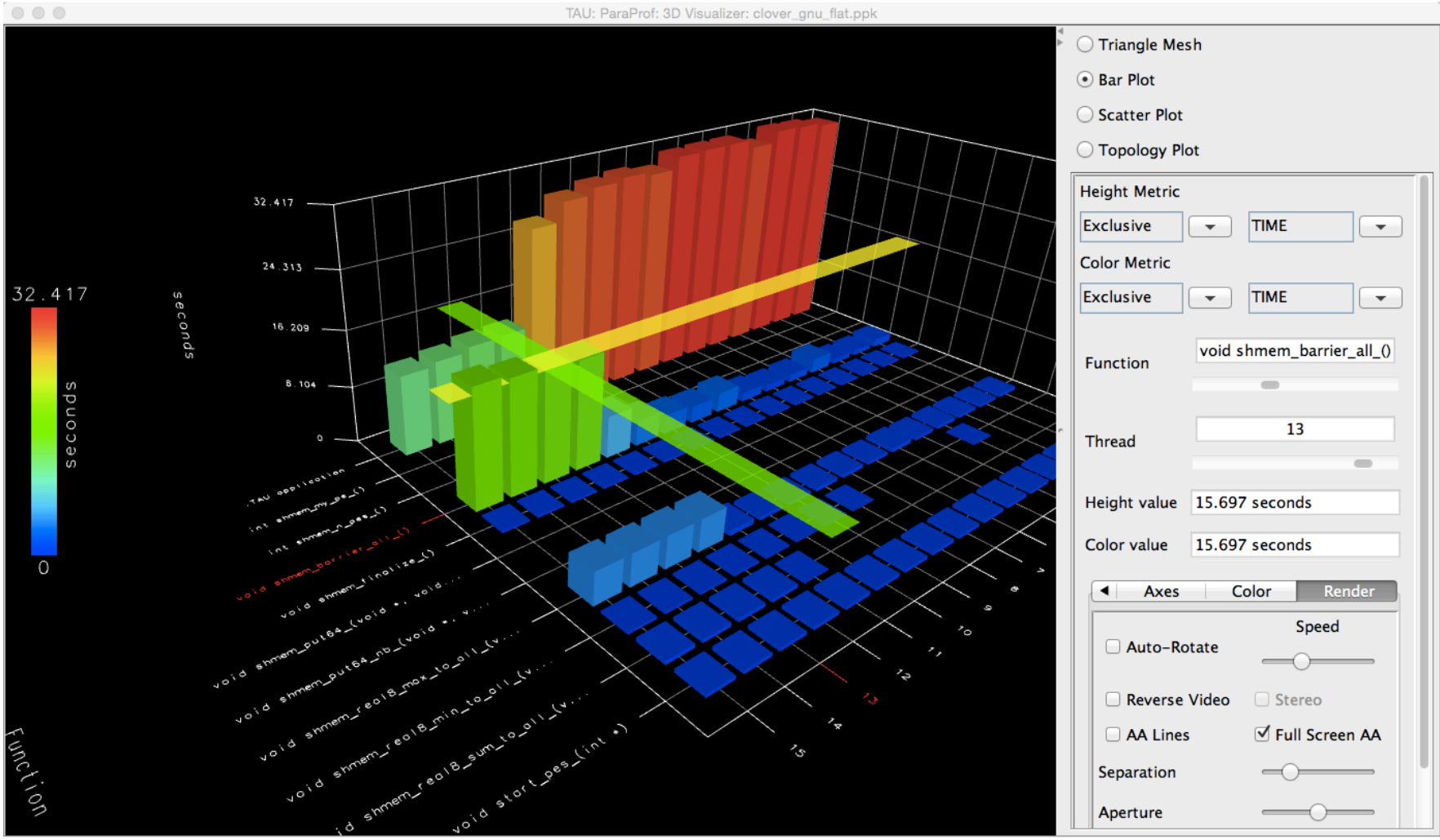
Name	Total	MeanValue	NumSamples	MinValue	MaxValue	Std. Dev.
.TAU application						
read()						
fopen64()						
fclose()						
OurMain()						
malloc size	25,235	1,097.174	23	11	12,032	2,851.143
free size	22,707	1,746.692	13	11	12,032	3,660.642
OurMain [{wrapper.py}{3}]						
read()						
malloc size	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()						

TAU – Callpath Profiling

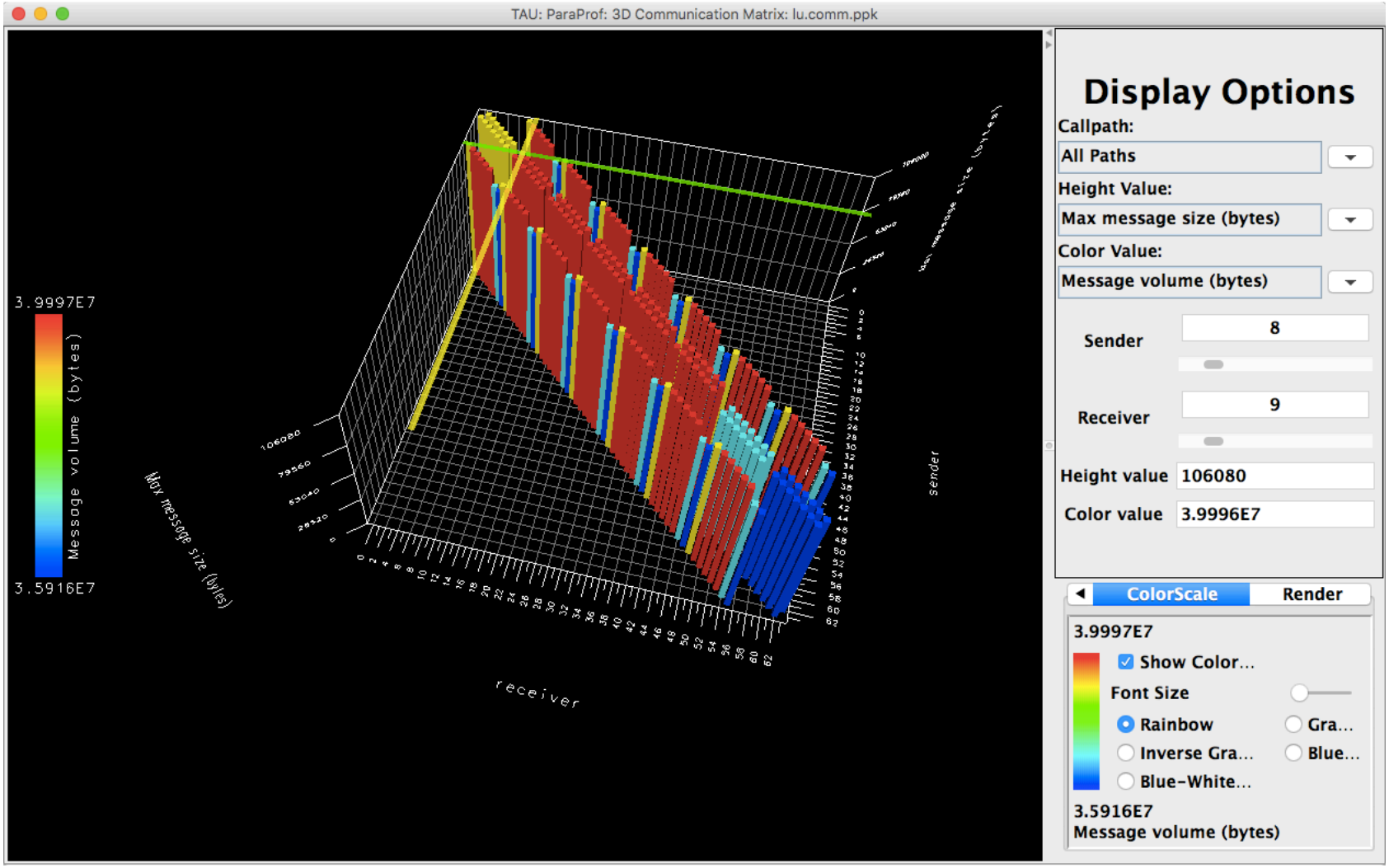


Name	Exclusive...	Inclusive... ▽	Calls	Child...
.TAU application	0	221.298	1	1
NODET [{main.f90} {4,1}-{35,17}]	0	221.298	1	105
FLOW::ITERATE [{flow.F90} {1692,14}]	0	197.989	100	500
FLOW::INITIALIZE_DATA [{flow.F90} {465,14}]	0	22.707	1	2
FLOW::INITIALIZE_DATA2 [{flow.F90} {663,14}]	0.002	22.705	1	197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

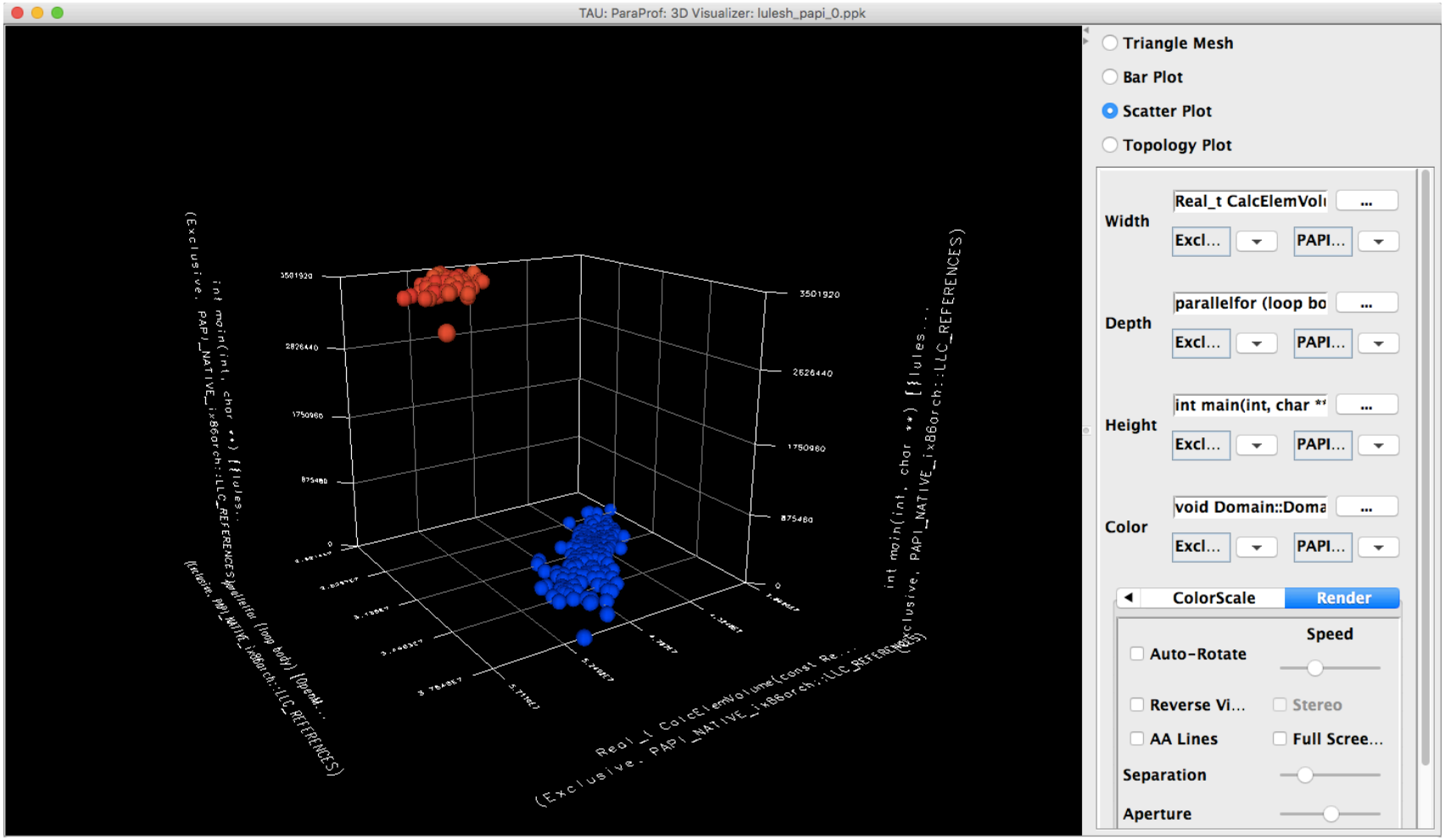
TAU – ParaProf 3D Visualization



TAU – 3D Communication Window



TAU – 3D Scatter Plot



Examples

Simplifying the use of TAU!

Uninstrumented code:

- % make
- % mpirun –np 4 ./lu.W.8

With TAU using event based sampling (EBS):

- % mpirun -np 4 tau_exec -ebs ./lu.W.8
- % paraprof
- % pprof -a | more

TAU Demo on Cheyenne, NCAR

```
On Cheyenne.ucar.edu,
% tar zxf ~sshende/pkgs/workshop.tgz
% source ~sshende/pkgs/tau.bashrc
% cd workshop/NPB3.1; make clean; make suite; cd bin
% qsub -I -A <Project_ID> -l walltime=1:30:00
                  -l select=2:ncpus=36:mpiprocs=36 -q regular
(OR: % getnode -A <Project_ID> )
% mpirun -np 8 ./lu.W.8
% mpirun -np 8 tau_exec -ebs ./lu.W.8
% paraprof


Binary rewriting (Beta):
% tau_rewrite lu.W.8 lu.i
% mpirun -np 8 ./lu.i
% paraprof
% pprof
```

And try the examples.

Please see runtime environment variables on the last 2 slides!

TAU for Heterogeneous Measurement

Multiple performance perspectives

Integrate Host-GPU support in TAU measurement framework

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

Tutorial

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

  % ./a.out (uninstrumented)

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

  % paraprof

TAU Execution Command (tau_exec)

Uninstrumented execution

- % mpirun -np 256 ./a.out

Track GPU operations

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

Track MPI performance

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

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

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

Track memory operations

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

Use event based sampling (compile with –g)

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

Binary Rewriting Instrumentation

- Support for both **static and dynamic** executables
- Specify a list of routines to instrument
- Specify the TAU measurement library to be injected
- **Dyninst [U. Wisconsin, U. Maryland]**:

```
% tau_run -T [tags] a.out -o a.inst
```

- **MAQAO [Intel Exascale Labs, UVSQ]**:

```
% tau_rewrite -T [tags] a.out -o a.inst
```

- **Pebil [SDSC]**:

```
% tau_pebil_rewrite -T [tags] a.out \
  -o a.inst
```

- Execute the application to get measurement data:

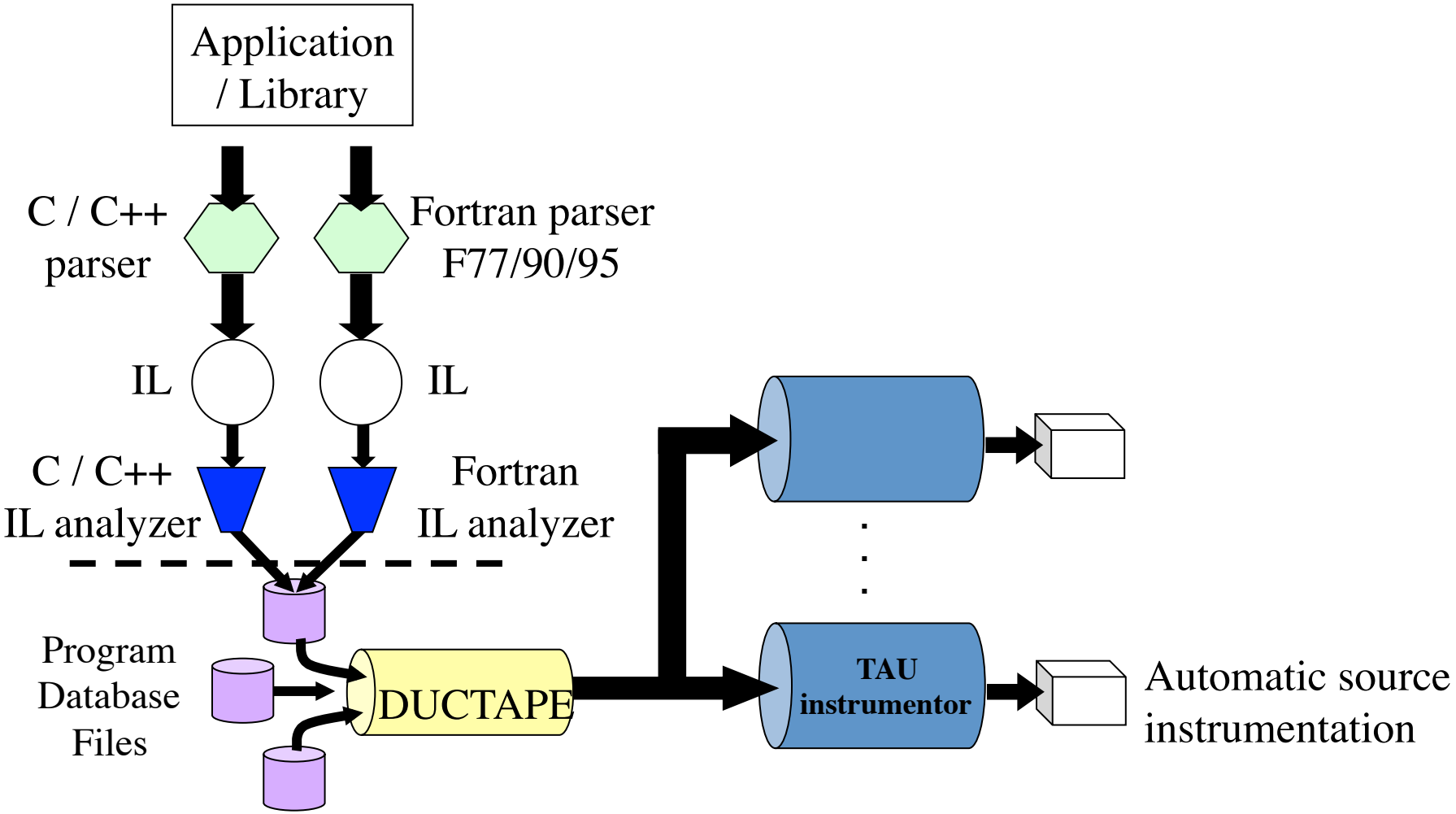
```
% mpirun -np 4  ./a.inst
```

Binary Rewriting Instrumentation

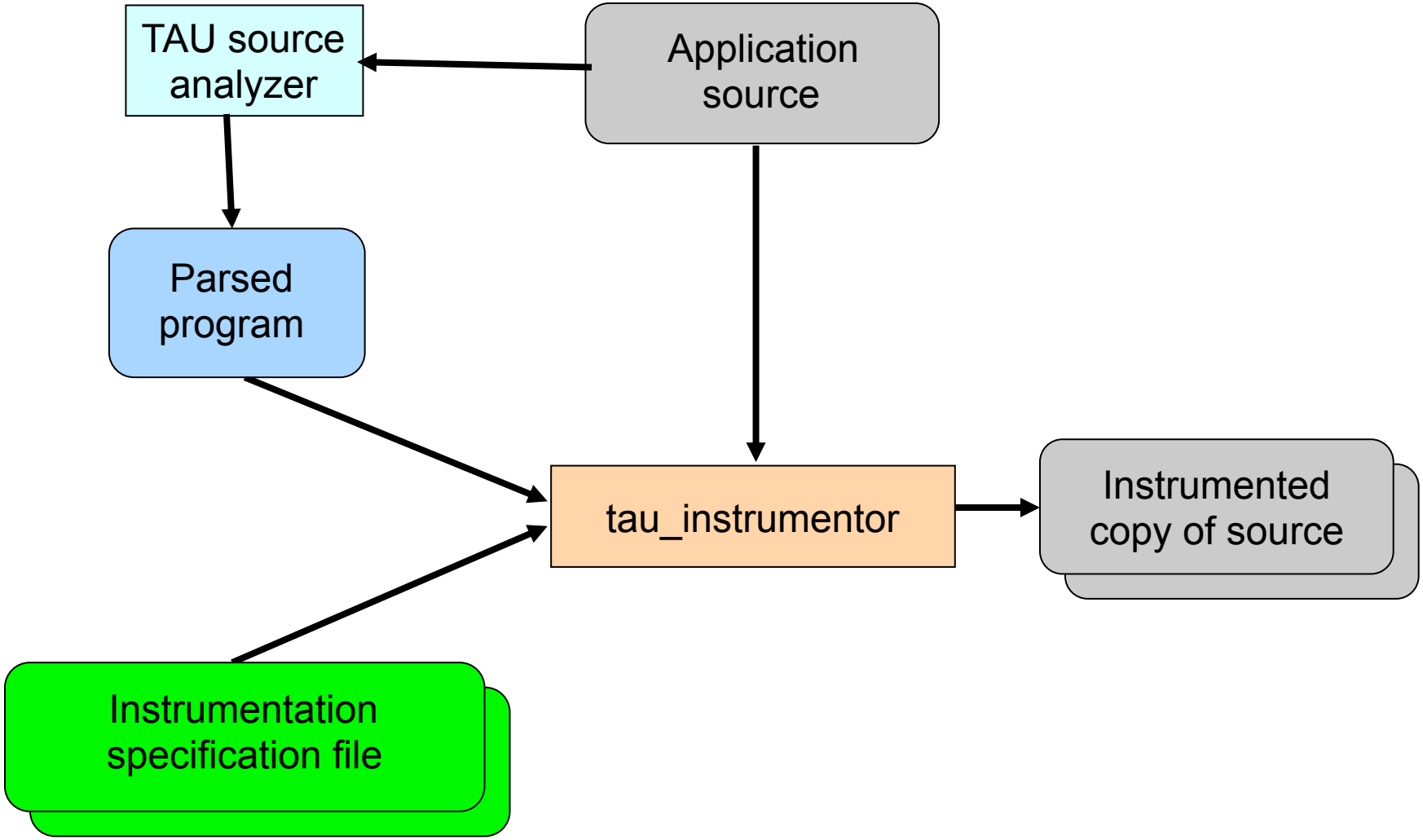
```
% mpif90 -g matmult.f90 -o matmult
% tau_rewrite matmult matmult.i

Or use a selective instrumentation file (include/exclude lists)
% tau_rewrite -f select.tau -T icpc,papi ./matmult -o matmult.i
% mpirun -np 256  ./matmult.i
% paraprof
```

Automatic Source Instrumentation

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



PDT: automatic source instrumentation



Installing TAU

Installing PDT:

- wget http://tau.uoregon.edu/pdt_lite.tgz
- ./configure –prefix=<dir>; make ; make install

Installing TAU:

- wget http://tau.uoregon.edu/tau.tgz
- ./configure -mpi –pdt=<dir> -bfd=download –unwind=download –iowrapper …
- make install

Using TAU:

- export TAU_MAKEFILE=<taudir>/x86_64/lib/Makefile.tau-<TAGS>
- make CC=tau_cc.sh CXX=tau_cxx.sh F90=tau_f90.sh

Installing TAU on Laptops

Installing TAU under Mac OS X:

- wget http://tau.uoregon.edu/tau.dmg
- Install tau.dmg

Installing TAU under Windows

- http://tau.uoregon.edu/tau.exe

Installing TAU under Linux

- http://tau.uoregon.edu/tau.tgz
- ./configure; make install
- export PATH=<taudir>/x86_64/bin:$PATH

Source Instrumentation in TAU

TAU supports several compilers, measurement, and thread options

Intel compilers, profiling with hardware counters using PAPI, MPI library, OpenMP…

Each measurement configuration of TAU corresponds to a unique stub makefile (configuration file) and library that is generated when you configure it

To instrument source code automatically using PDT

Choose an appropriate TAU stub makefile in <arch>/lib:

% module load UNITE tau

% export TAU_MAKEFILE=<tauroot>/x86_64/lib/Makefile.tau-mpt-icpc-papi-mpi-pdt

% export TAU_OPTIONS= '-optVerbose …' (see tau_compiler.sh)

Use tau_f90.sh, tau_cxx.sh, tau_upc.sh, tau_caf.sh, or tau_cc.sh as F90, C++, UPC, CAF, or C compilers respectively:

% mpif90 foo.f90 changes to

% tau_f90.sh foo.f90

Set runtime environment variables, execute application and analyze performance data:

% pprof (for text based profile display)

% paraprof (for GUI)

Different Makefiles for TAU Compiler

```
%  source ~sshende/pkgs/tau.bashrc
% ls $TAU/Makefile.*
/glade/u/home/sshende/pkgs/tau-2.27/x86_64/lib/Makefile.tau
/glade/u/home/sshende/pkgs/tau-2.27/x86_64/lib/Makefile.tau-icpc-papi-pthread-pdt
/glade/u/home/sshende/pkgs/tau-2.27/x86_64/lib/Makefile.tau-impi-icpc-papi-mpi-pdt
/glade/u/home/sshende/pkgs/tau-2.27/x86_64/lib/Makefile.tau-impi-icpc-papi-mpi-pthread-pdt
/glade/u/home/sshende/pkgs/tau-2.27/x86_64/lib/Makefile.tau-impi-icpc-papi-ompt-mpi-pdt-
openmp
/glade/u/home/sshende/pkgs/tau-2.27/x86_64/lib/Makefile.tau-mpt-icpc-papi-mpi-pdt
/glade/u/home/sshende/pkgs/tau-2.27/x86_64/lib/Makefile.tau-mpt-icpc-papi-mpi-pthread-pdt
/glade/u/home/sshende/pkgs/tau-2.27/x86_64/lib/Makefile.tau-mpt-icpc-papi-ompt-mpi-pdt-openmp
```

For an SGI MPT MPI+F90 application with Intel Compilers, you may choose

```
Makefile.tau-mpt-icpc-papi-mpi-pdt
```

Supports MPI instrumentation & PDT for automatic source instrumentation

```
% export TAU_MAKEFILE=$TAU/Makefile.tau-mpt-icpc-papi-mpi-pdt
% tau_f90.sh app.f90 -o app; mpirun -n 256 ./app; paraprof
```

Configuration tags for tau_exec

```
% ./configure -pdt=<dir> -mpi -papi=<dir>; make install
```
Creates in $TAU:
```
Makefile.tau-papi-mpi-pdt(Configuration parameters in stub makefile)
shared-papi-mpi-pdt/libTAU.so

% ./configure -pdt=<dir> -mpi; make install  creates
Makefile.tau-mpi-pdt
shared-mpi-pdt/libTAU.so
```
To explicitly choose preloading of shared-<options>/libTAU.so change:
```
% mpirun -np   256 ./a.out     to
% mpirun -np   256  tau_exec -T <comma_separated_options> ./a.out

% mpirun -np   256  tau_exec -T papi,mpi,pdt ./a.out
```
Preloads `$TAU/shared-papi-mpi-pdt/libTAU.so`
```
% mpirun -np   256  tau_exec -T papi ./a.out
```
Preloads `$TAU/shared-papi-mpi-pdt/libTAU.so by matching.`
```
% mpirun -np 256 tau_exec -T papi,mpi,pdt -s ./a.out
```
Does not execute the program. Just displays the library that it will preload if executed without the –s option.
NOTE: -mpi configuration is selected by default. Use –T serial for
Sequential programs.

Compile-Time Options

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

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

Compile-Time Options (contd.)

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

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

See tau_compiler.sh for a full list of TAU_OPTIONS.

…

Compiling Fortran Codes with TAU

If your Fortran code uses free format in .f files (fixed is default for .f), you may use:

% 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 -optDetectMemoryLeaks'

To use an instrumentation specification file:

% export TAU_OPTIONS= '-optTauSelectFile=select.tau -optVerbose -optPreProcess'

% cat select.tau

```
BEGIN_INSTRUMENT_SECTION
loops  routine="#"
# this statement instruments all outer loops in all routines. # is wildcard as well as comment in first column.
END_INSTRUMENT_SECTION
```

Selective Instrumentation File With Program Database Toolkit (PDT)

To use an instrumentation specification file for source instrumentation:

% export TAU_OPTIONS= '-optTauSelectFile=/path/to/select.tau -optVerbose '

% cat select.tau

```
BEGIN_EXCLUDE_LIST
BINVCRHS
MATMUL_SUB
MATVEC_SUB
EXACT_SOLUTION
BINVRHS
LHS#INIT
TIMER_#
END_EXCLUDE_LIST
```

NOTE: paraprof can create this file from an earlier execution for you.

File -> Create Selective Instrumentation File -> save

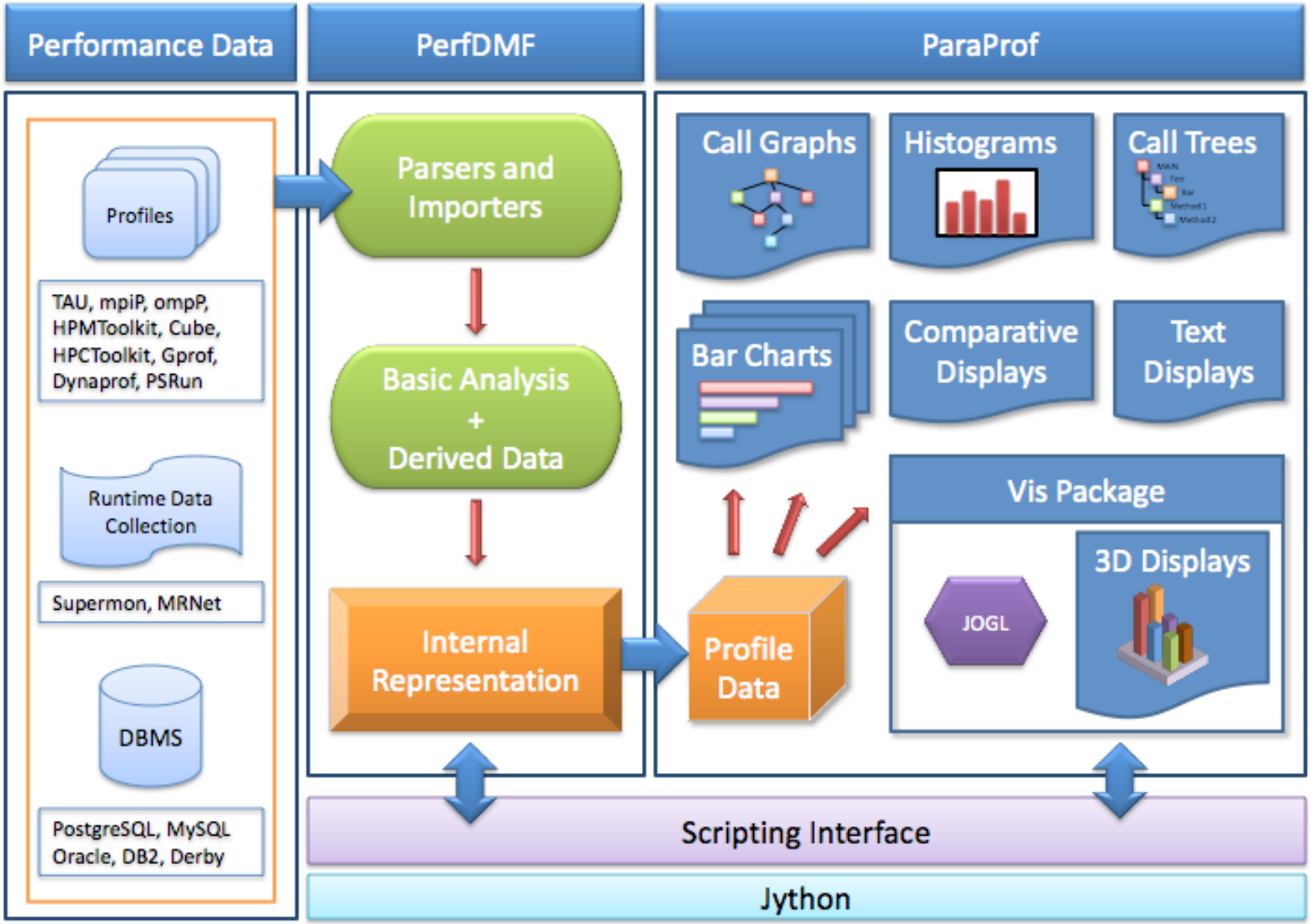
Selective instrumentation at runtime:

% export TAU_SELECT_FILE=select.tau

Using CoArray Fortran with TAU

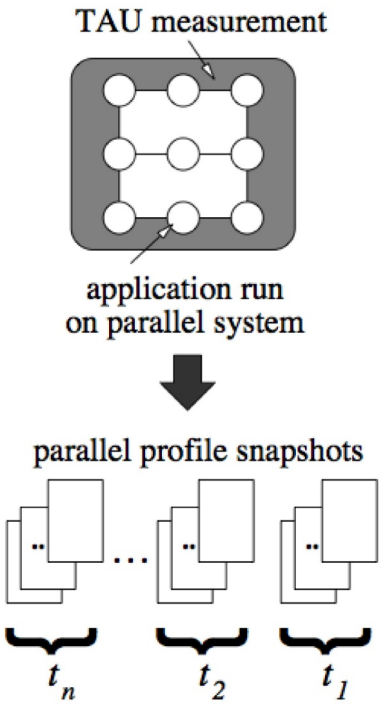
```
% source ~sshende/pkgs/tau.caf_openmpi.bashrc
% cd workshop/caf
% make clean
% make
% getnode -A <project_id>  (e.g., -A sore0001)
% cafrun -n 4 ./pi
% cafrun -n 4 tau_exec -T openmpi -ebs ./pi
% pprof -a
% paraprof  (Right click on node -> Show thread statistics
Table -> expand node .TAU Application -> right click ->Show
source code. )

Using compiler-based instrumentation:
% make clean
% make F90=tau_caf.sh
% cafrun -n 4 ./pi
% pprof -a
```

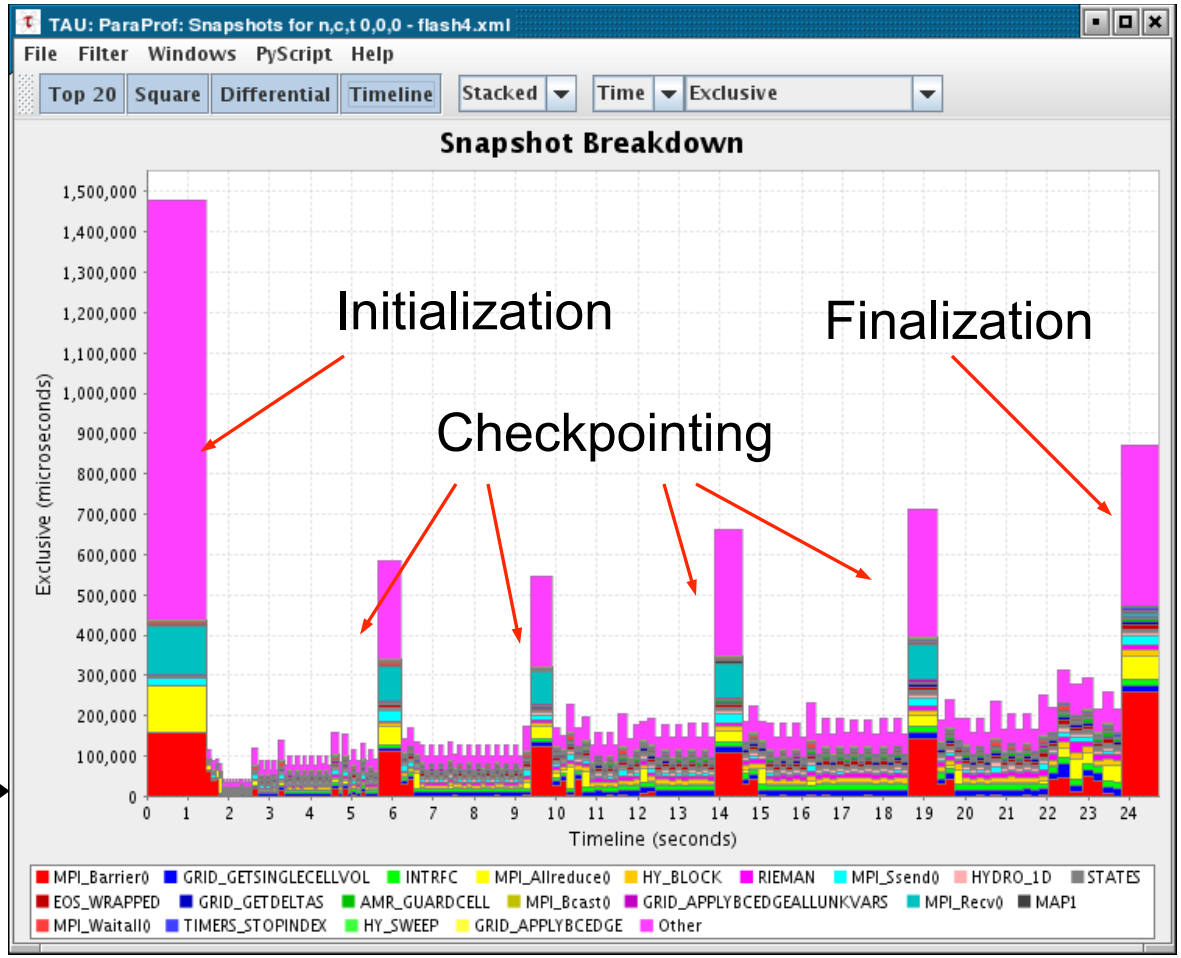
ParaProf Profile Analysis Framework



Profile Snapshots in ParaProf

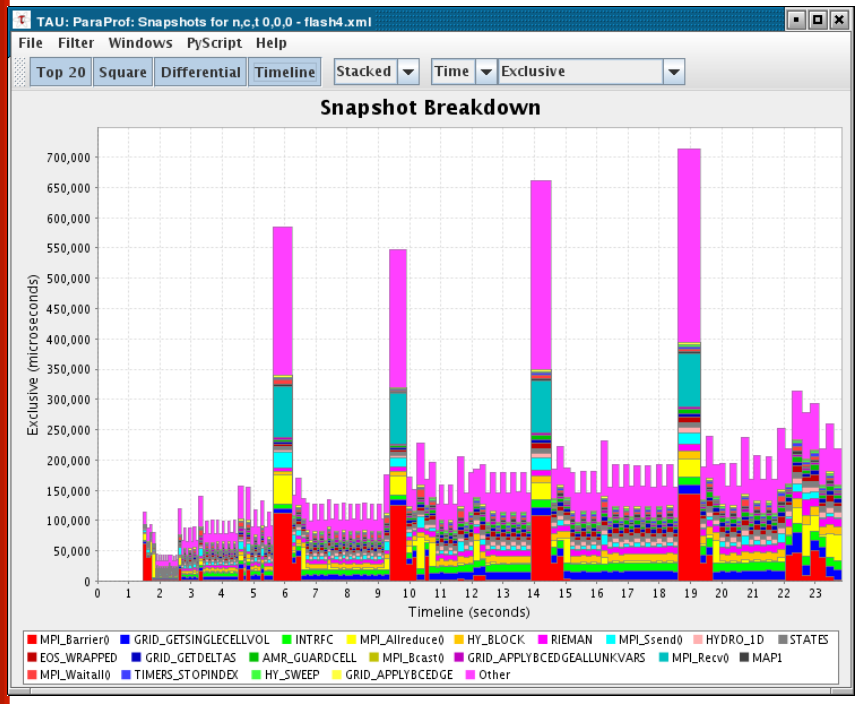
- Profile snapshots are profiles recorded at runtime
- Shows performance profile dynamics (all types allowed)



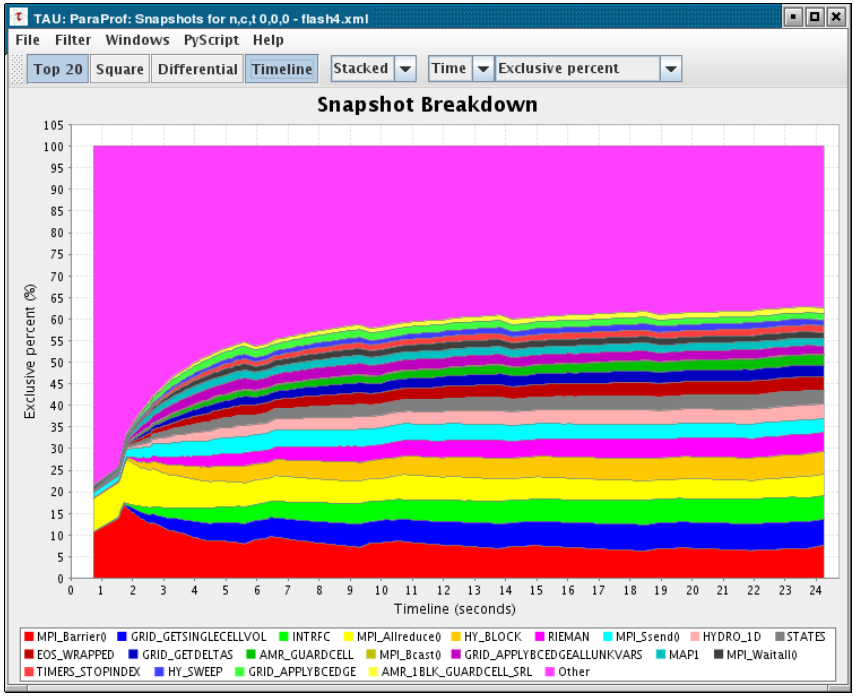


Profile Snapshot Views

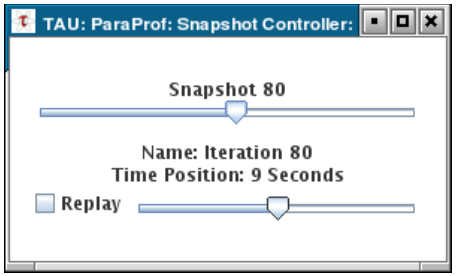
Percentage breakdown



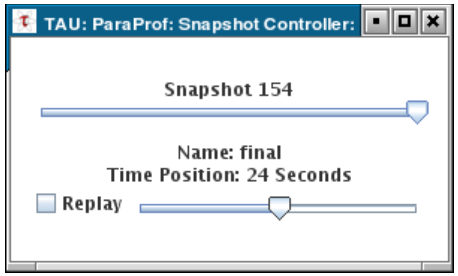
Only show main loop

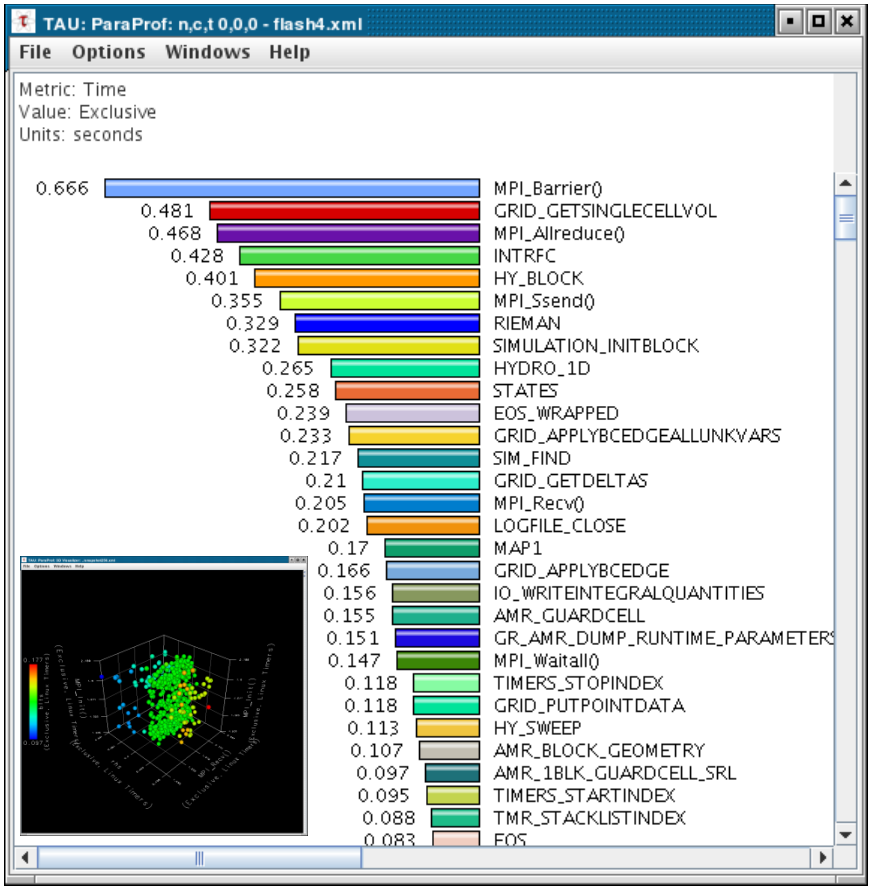


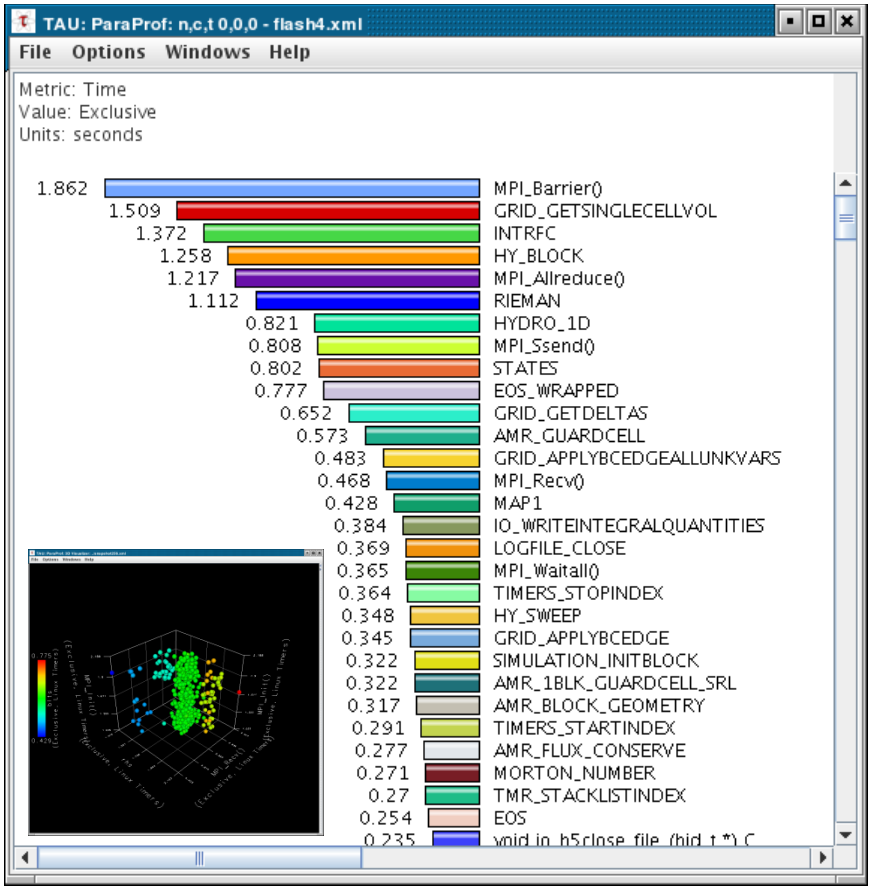
Snapshot Replay in ParaProf



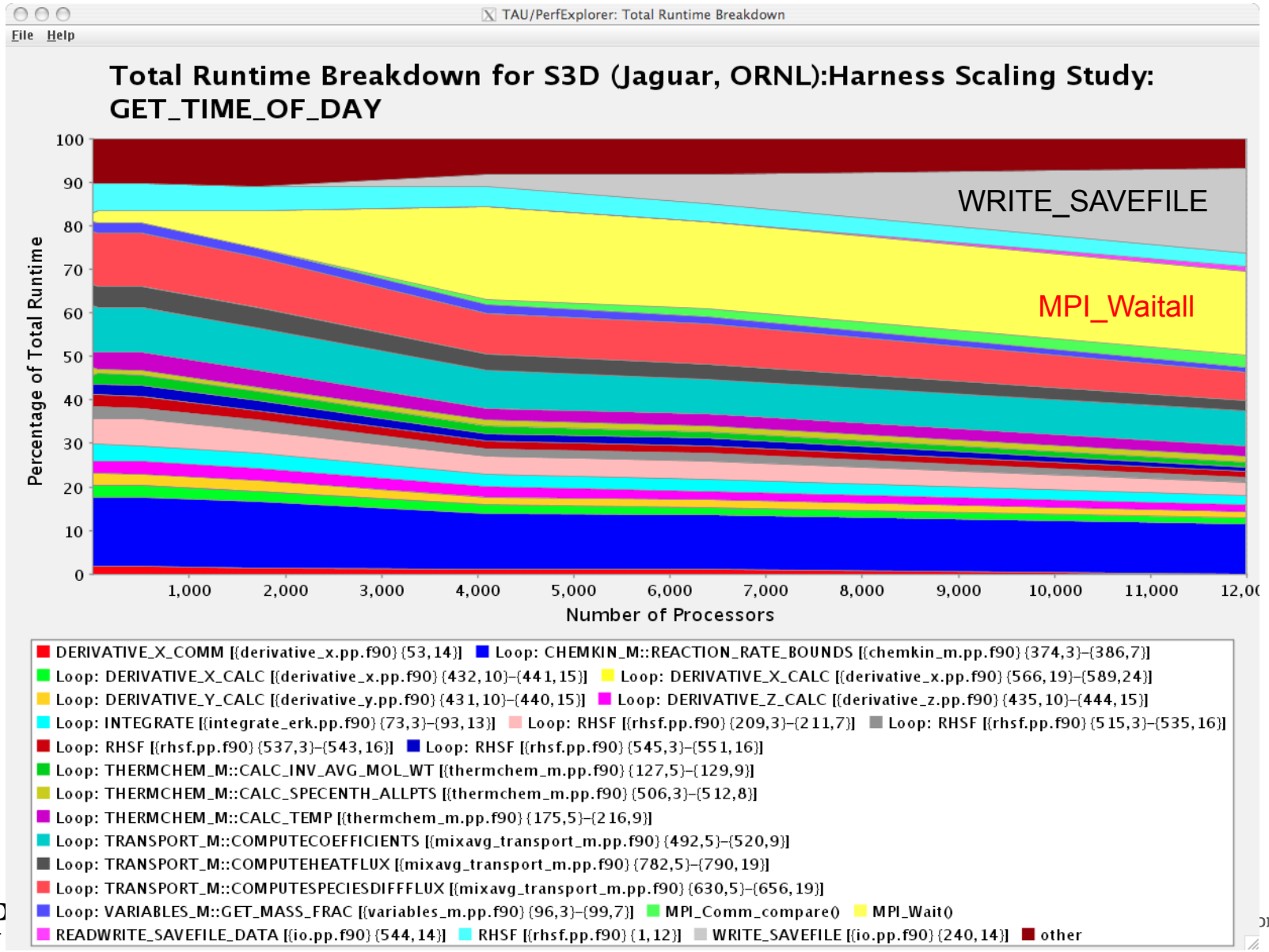
All windows dynamically update



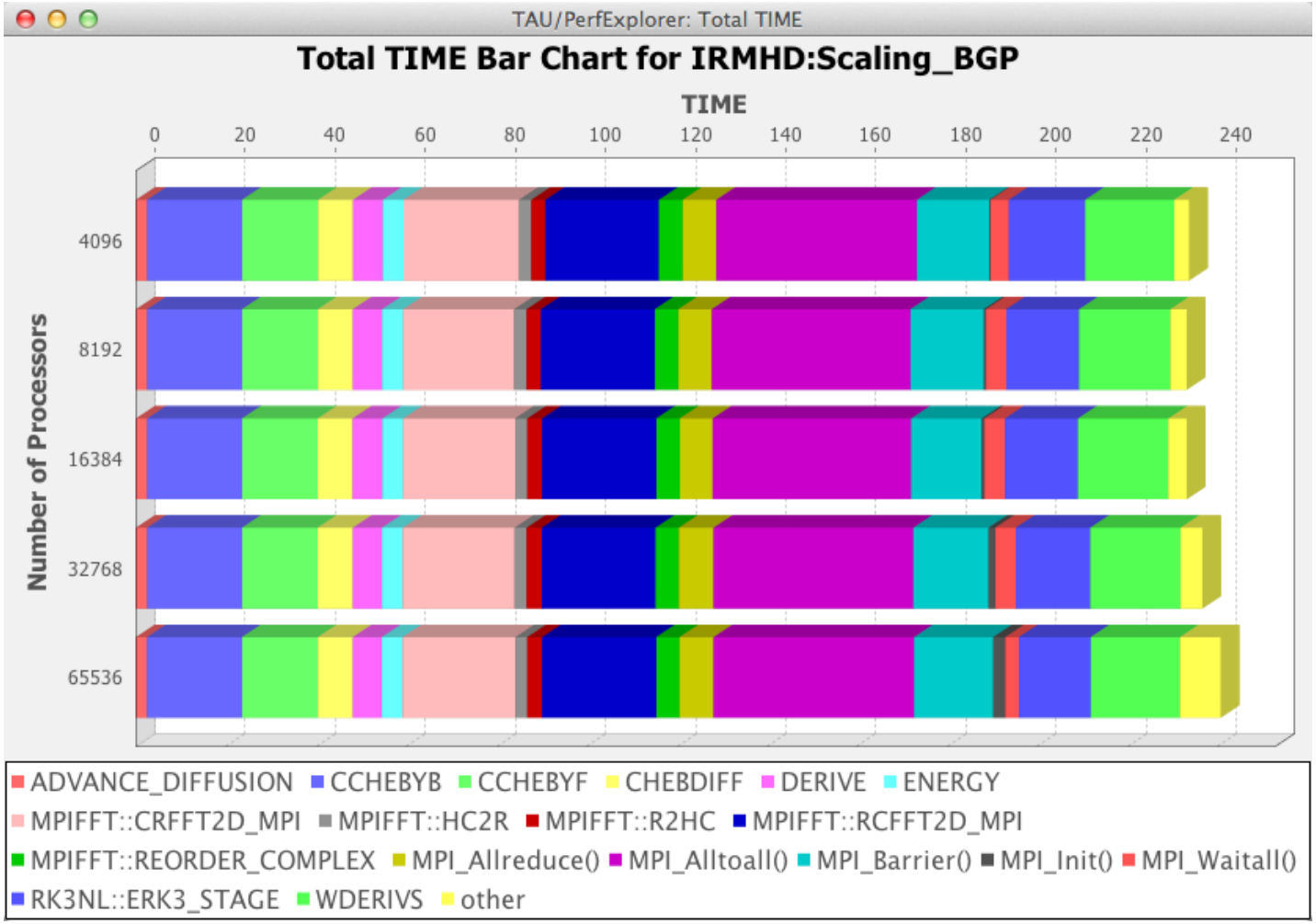




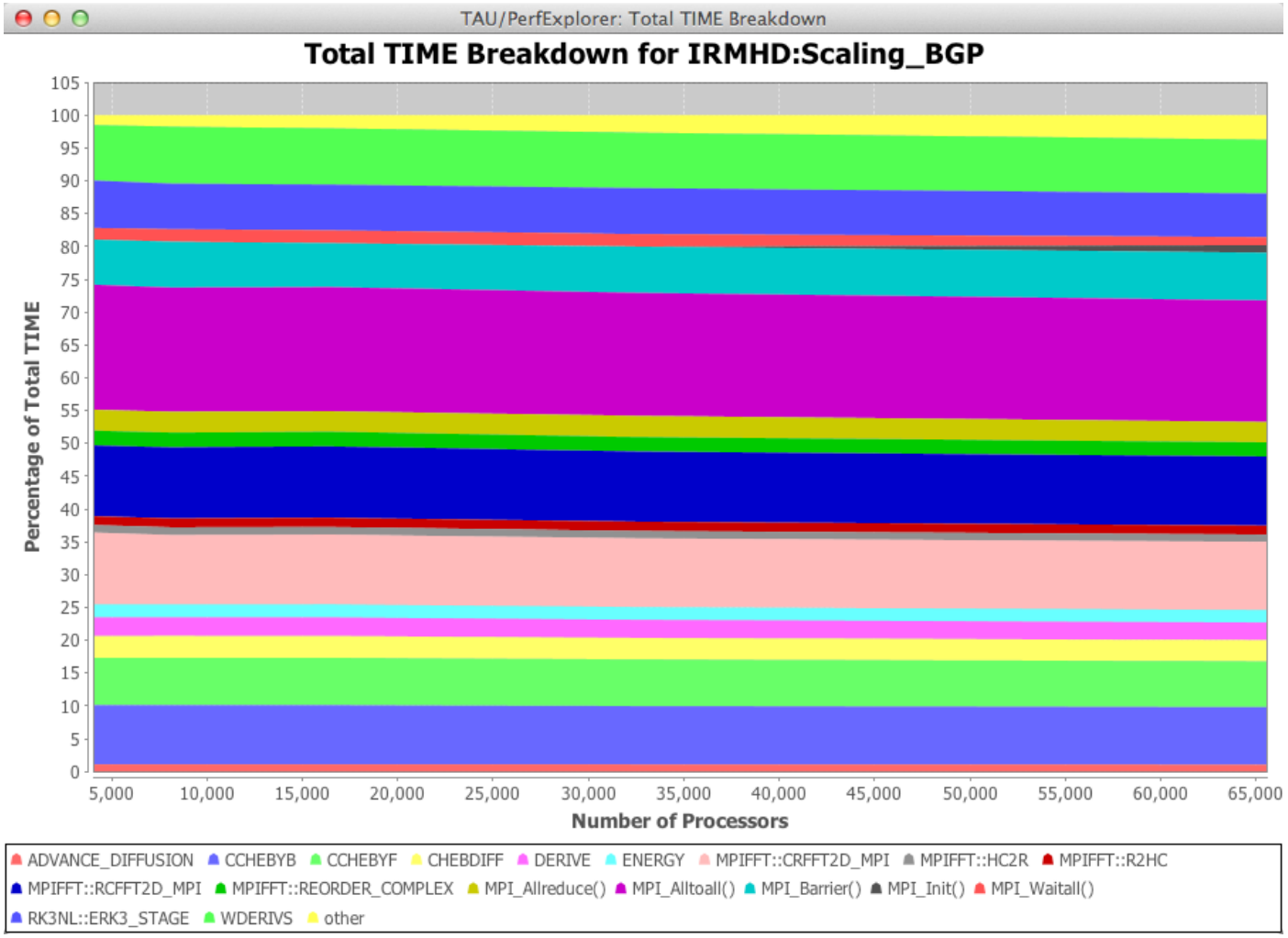
PerfExplorer – Runtime Breakdown



Evaluate Scalability

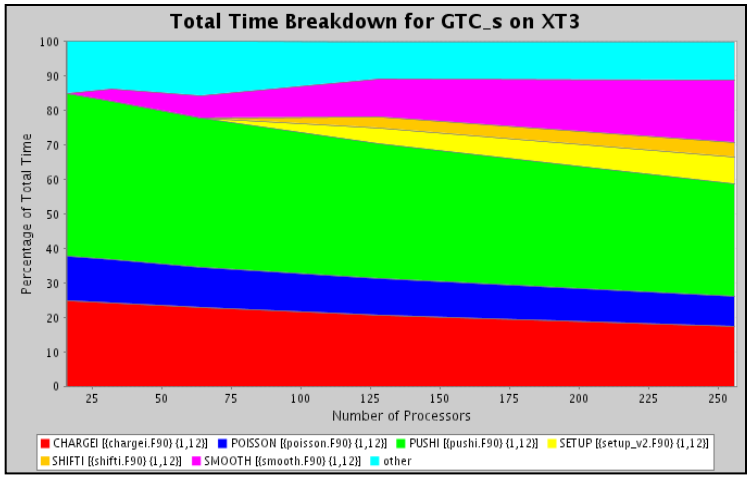


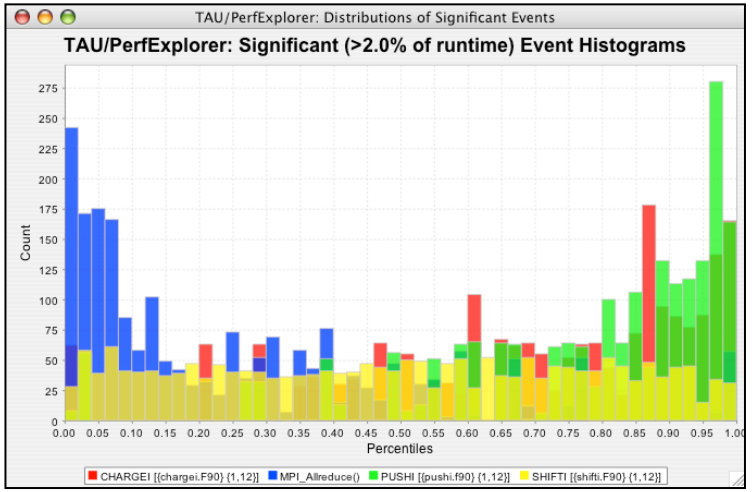
Runtime Breakdown



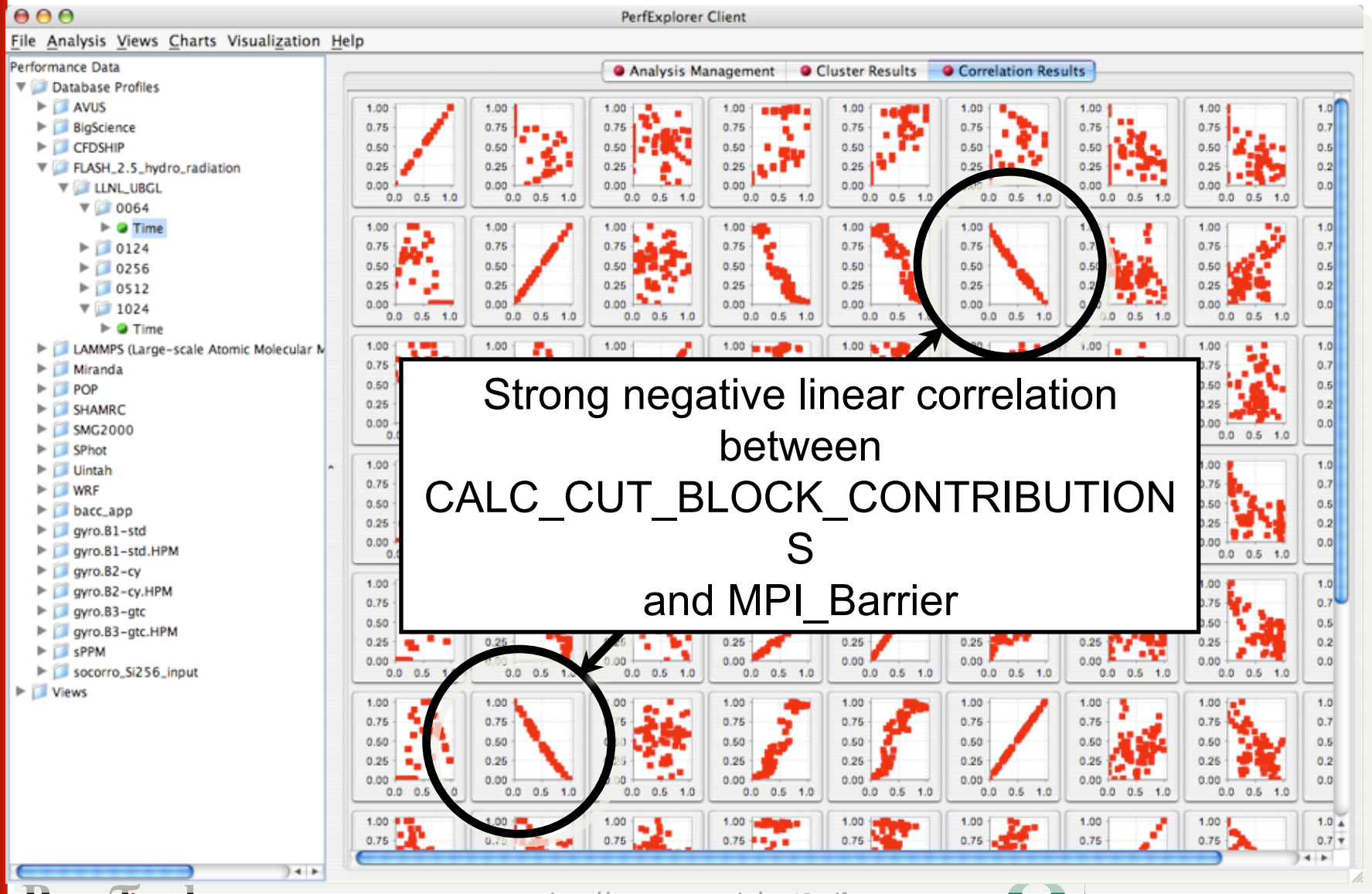
PerfExplorer – Relative Comparisons

- Total execution time
- Timesteps per second
- Relative efficiency
- Relative efficiency per event
- Relative speedup
- Relative speedup per event
- Group fraction of total
- Runtime breakdown
- Correlate events with total runtime
- Relative efficiency per phase
- Relative speedup per phase
- Distribution visualizations



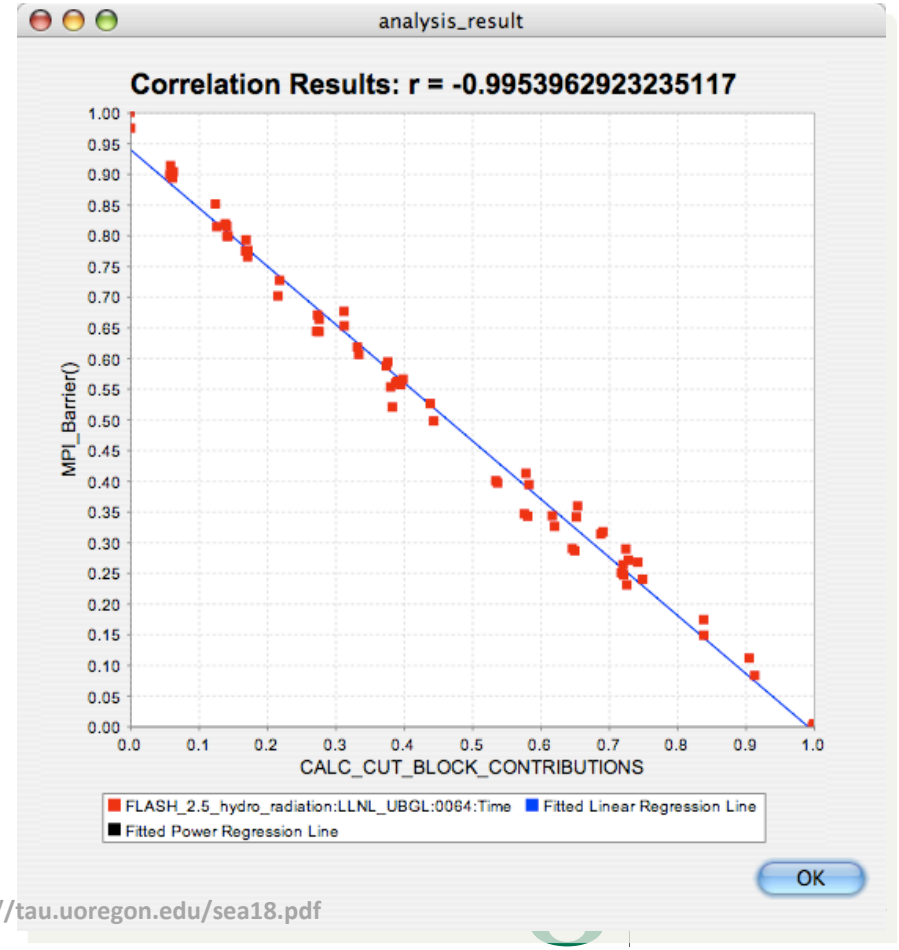


PerfExplorer – Correlation Analysis

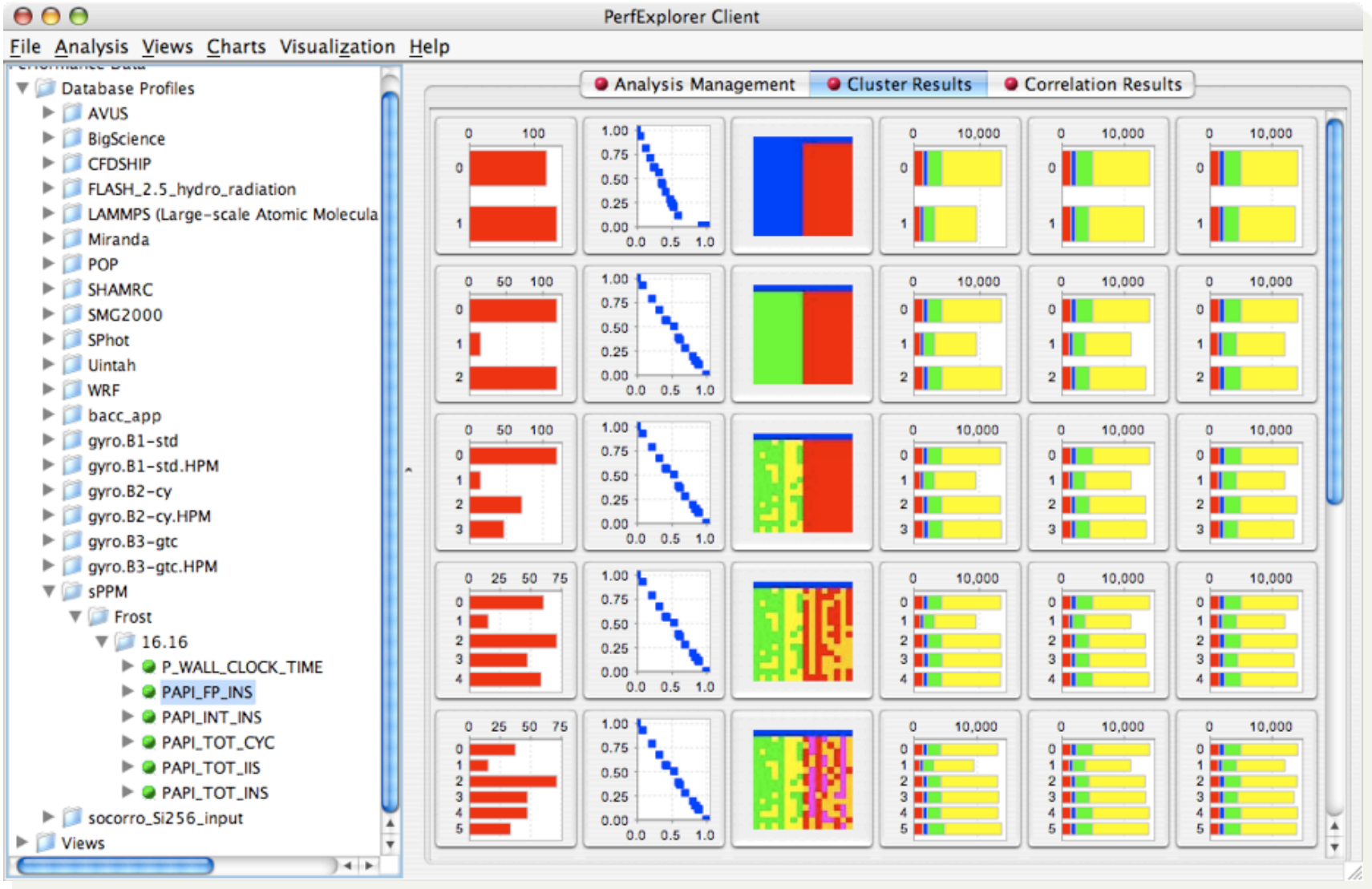


PerfExplorer – Correlation Analysis

-0.995 indicates strong, negative relationship. As CALC_CUT_BLOCK_CONTRIBUTIONS() increases in execution time, MPI_Barrier() decreases



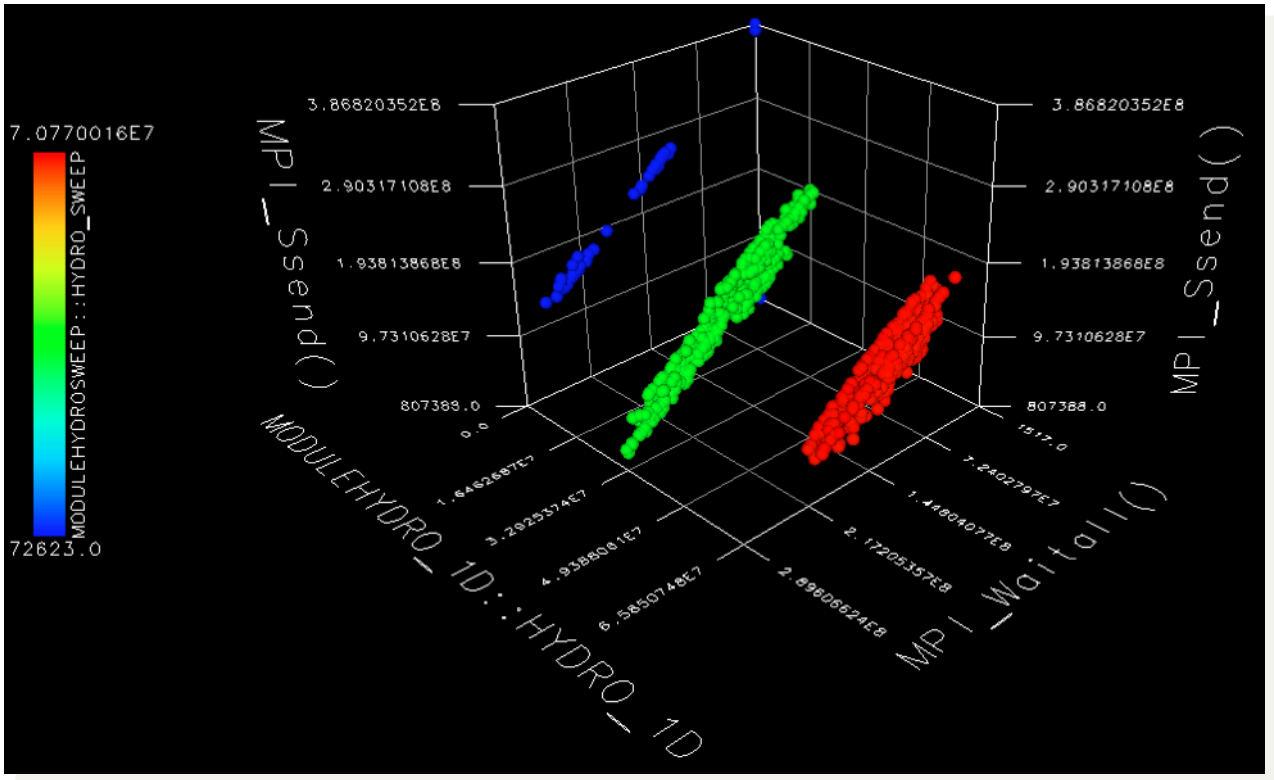
PerfExplorer – Cluster Analysis



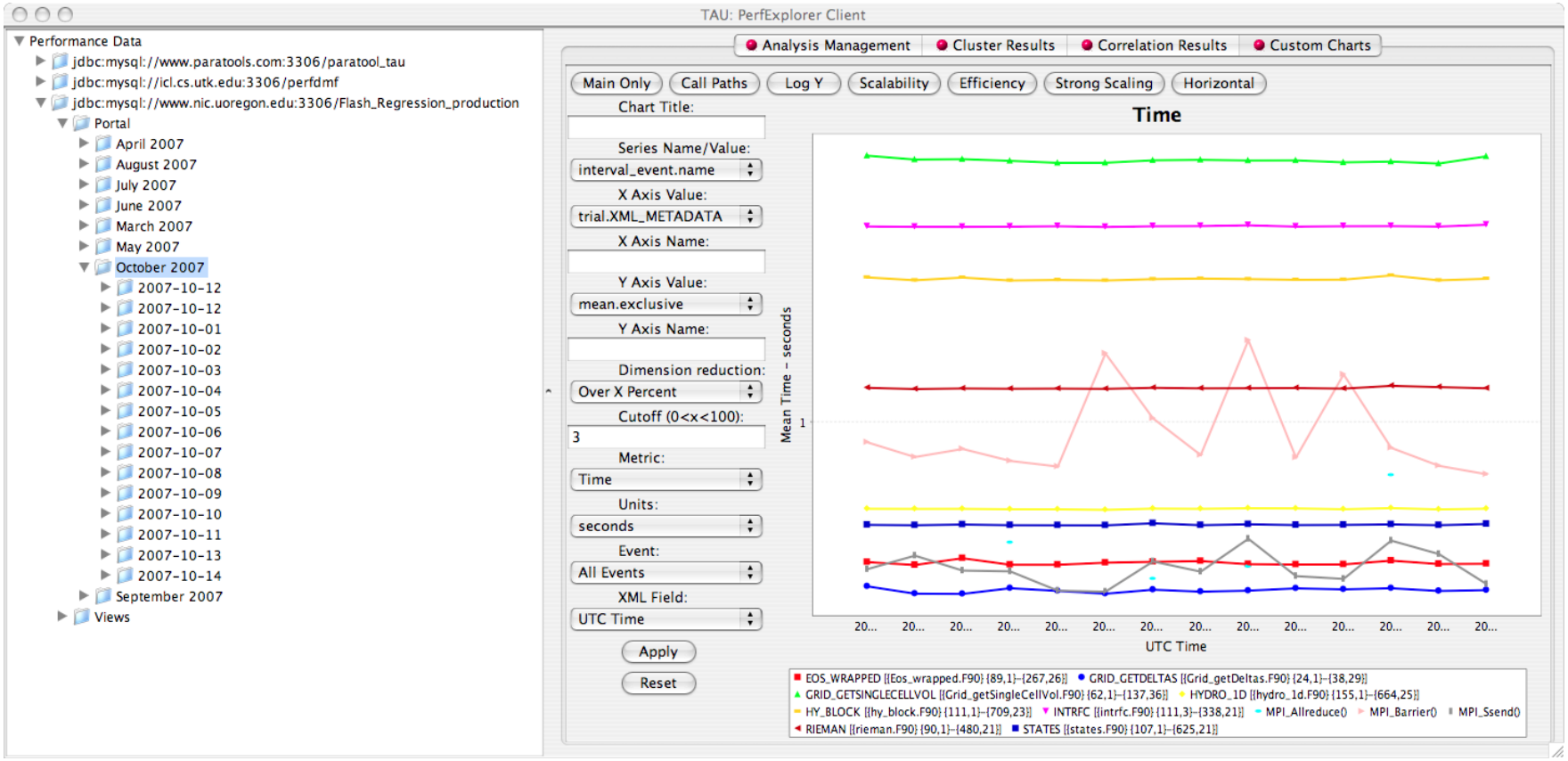
PerfExplorer – Cluster Analysis

Four significant events automatically selected

Clusters and correlations are visible



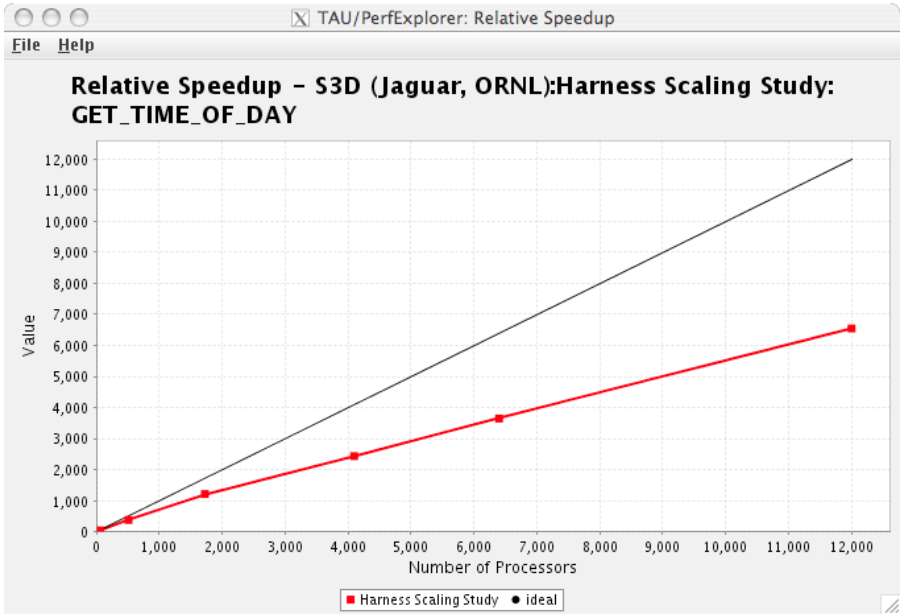
PerfExplorer – Performance Regression

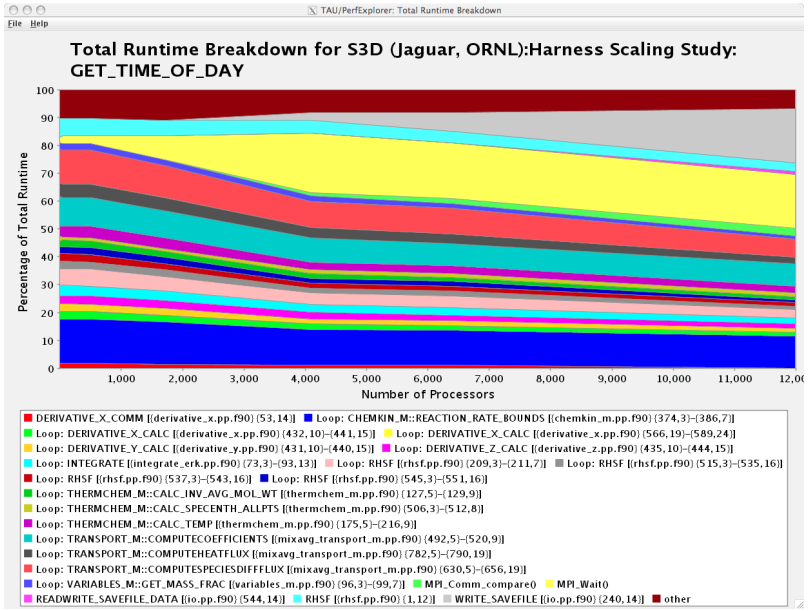


Evaluate Scalability

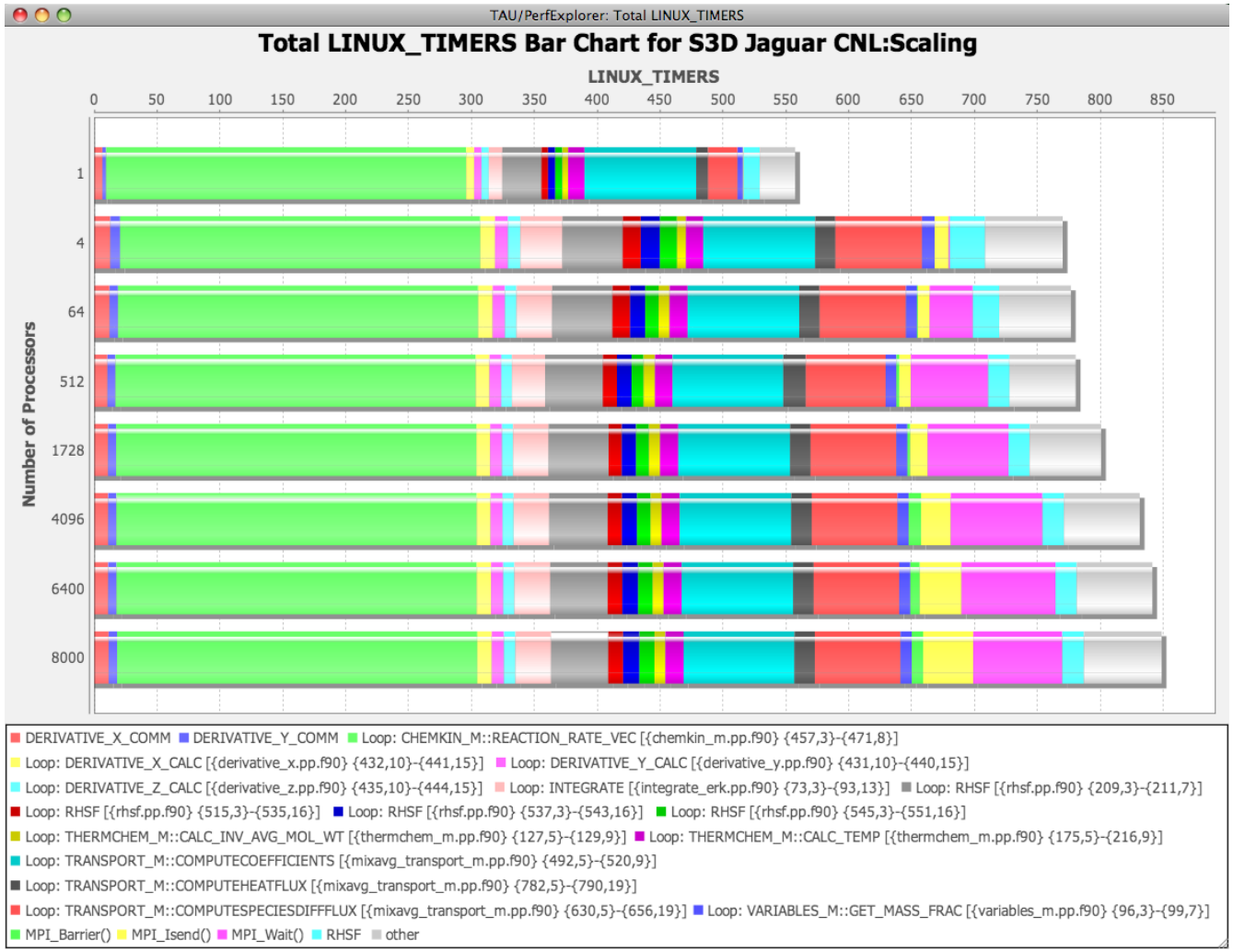
Goal: How does my application scale? What bottlenecks at what CPU counts?

Load profiles in PerfDMF database and examine with PerfExplorer

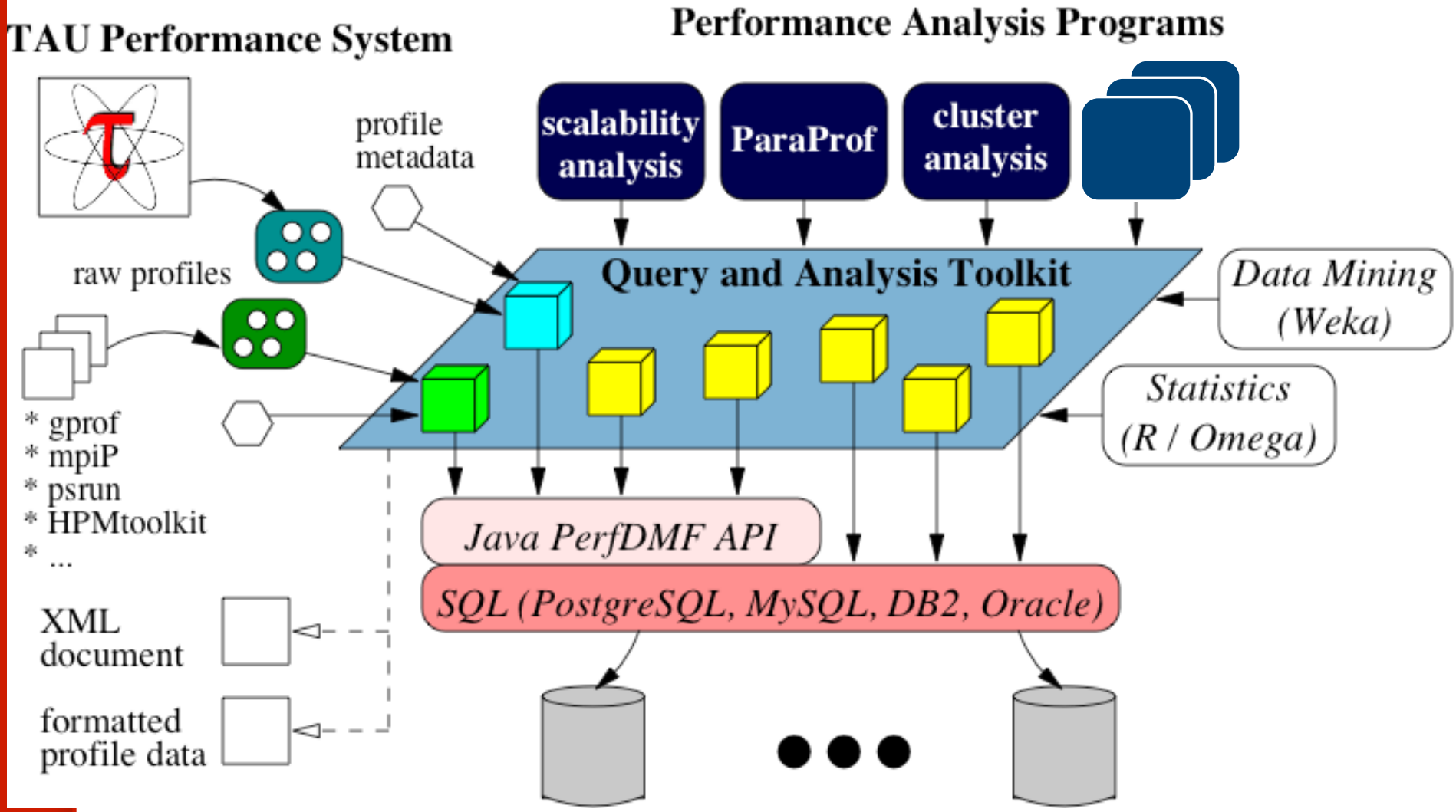




Usage Scenarios: Evaluate Scalability



TAUdb: Framework for Managing Performance Data



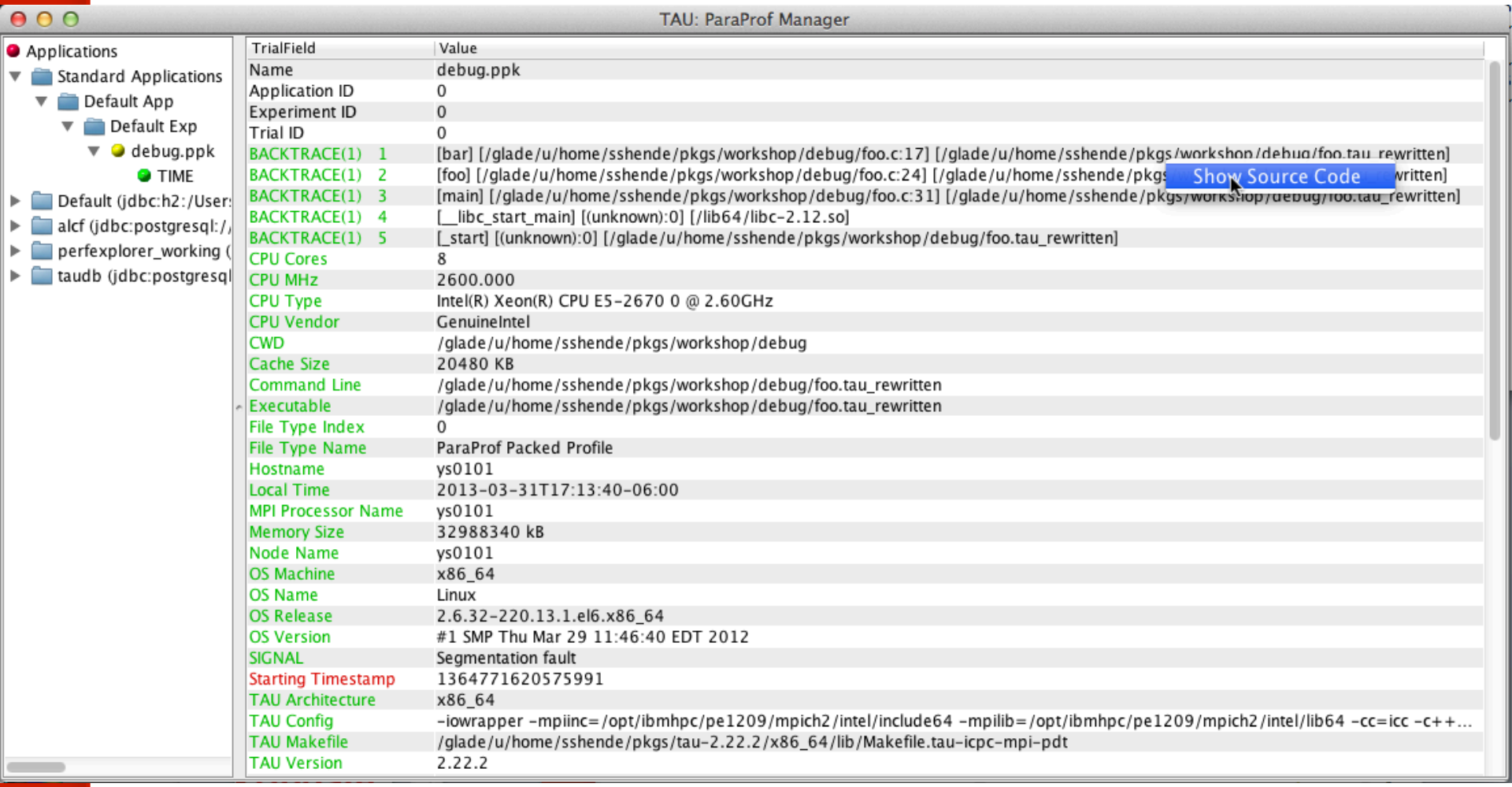
Evaluate Scalability using PerfExplorer Charts

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

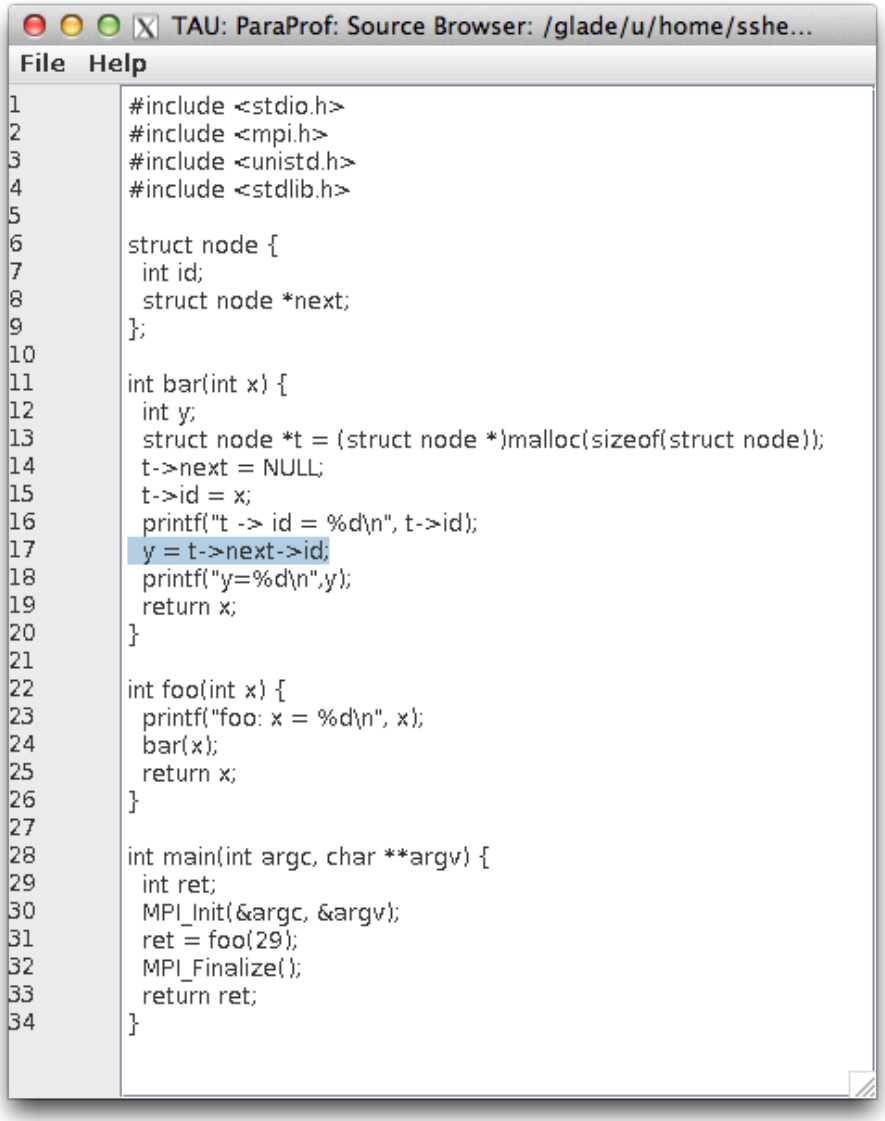
Multi-language Application Debugging

```
% export TAU_MAKEFILE=$TAU/Makefile.tau-mpt-icpc-papi-mpi-pdt
% export TAU_OPTIONS='-optMemDbg -optVerbose'
% make F90=tau_f90.sh CC=tau_cc.sh CXX=tau_cxx.sh
% mxterm 1 16 120
% export TAU_MEMDBG_PROTECT_ABOVE=1
% export TAU_MEMDBG_PROTECT_BELOW=1
% export TAU_MEMDBG_PROTECT_FREE=1
% mpirun -np 4  ./matmult
% paraprof
```

Multi-language Application Debugging



Location of segmentation violation



Memory Leak Detection

```
% export TAU_MAKEFILE=$TAU/Makefile.tau-icpc-papi-mpi-pdt
% export TAU_OPTIONS='-optMemDbg -optVerbose'
% make F90=tau_f90.sh CC=tau_cc.sh CXX=tau_cxx.sh
% mxterm 1 16 120

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

Multi-language Memory Leak Detection

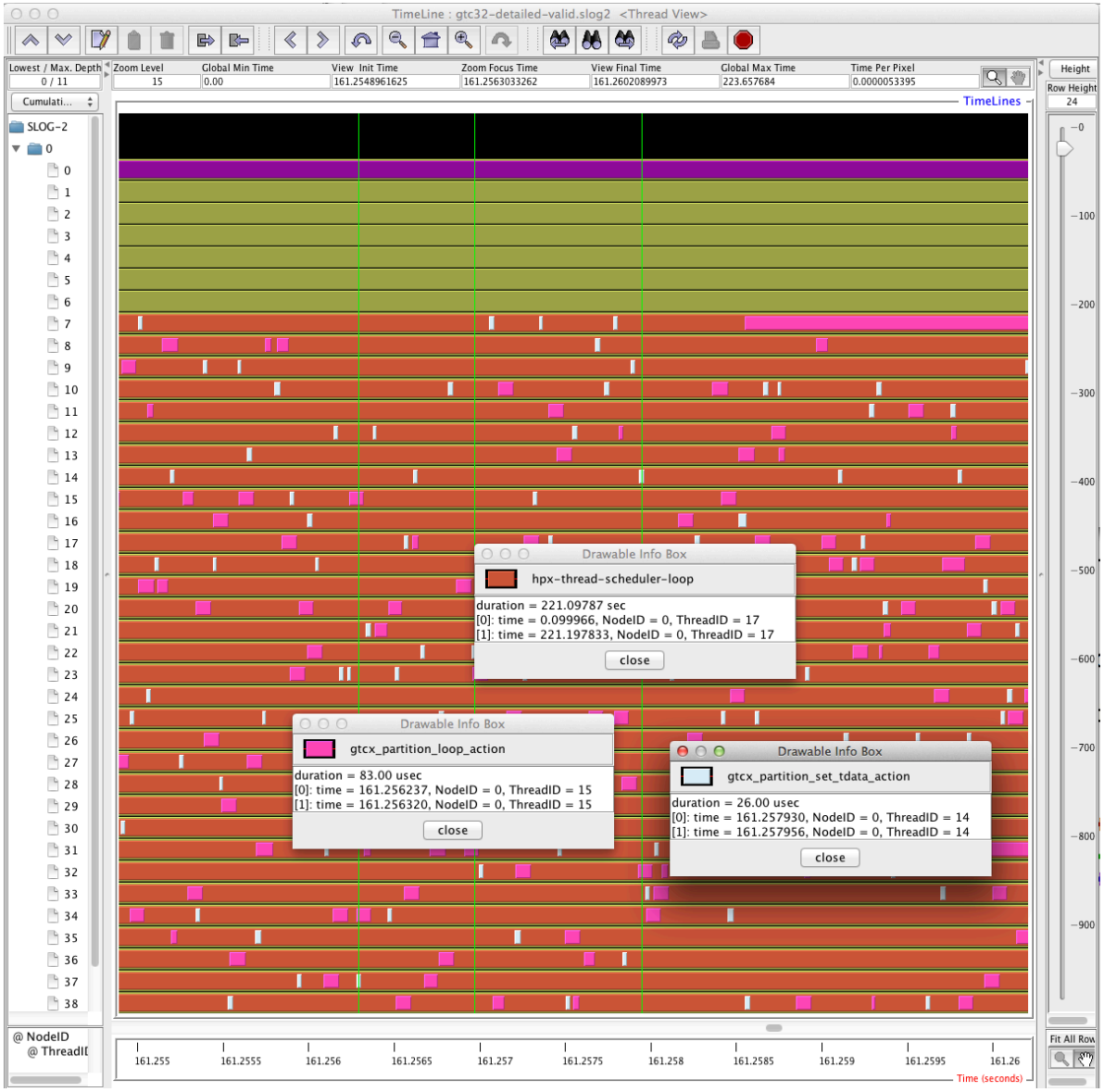
TAU: ParaProf: Context Events for: node 0 – memleak.ppk

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

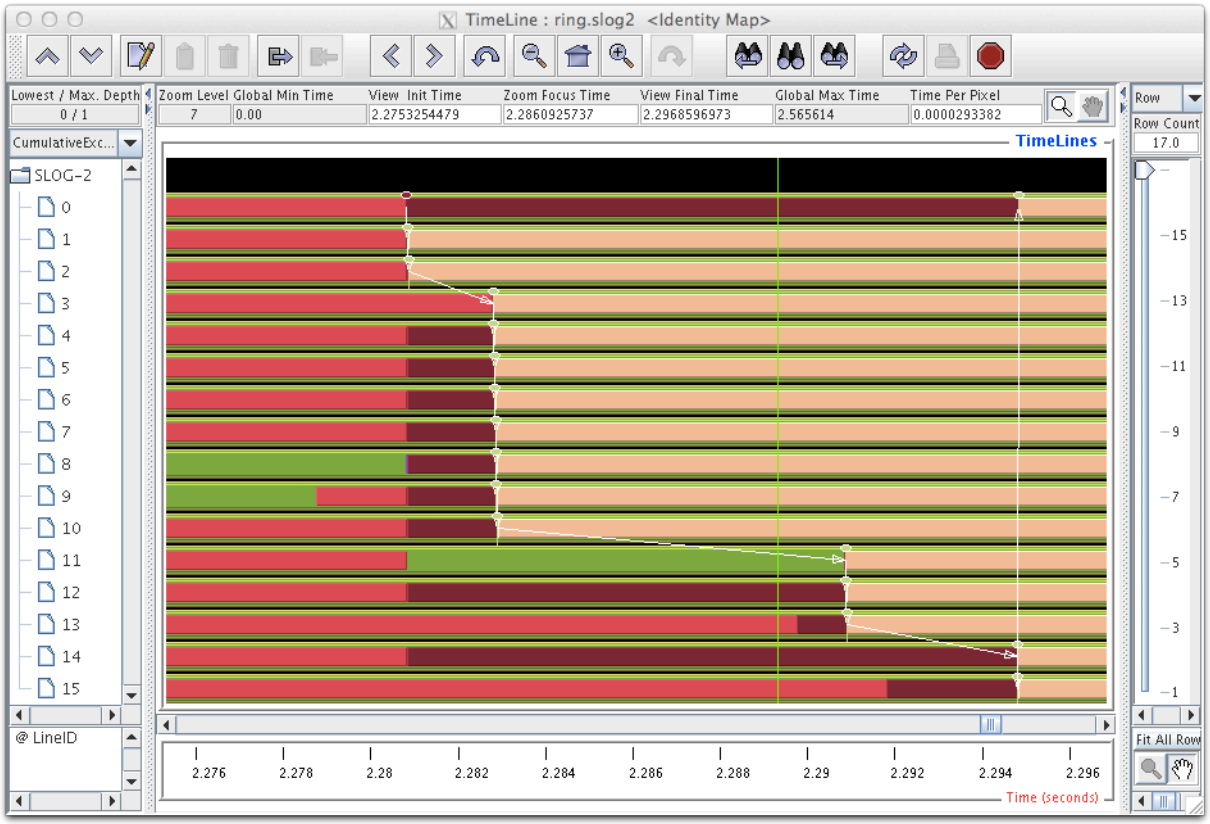
Trace Visualization in TAU

```
Jumpshot (ships with TAU, free trace visualizer)
% source ~sshende/pkgs/tau.bashrc; cd NPB3.1; make clean; make
% cd bin
% getnode -A <ID>
% export TAU_CALLSITE=1;
% export TAU_TRACE=1

Vampir (Commercial trace visualizer)
% export TAU_TRACE=1
% export TAU_TRACE_FORMAT=otf2
% mpirun -np 8 tau_exec ./lu.W.8
% vampir traces.otf2 &        (No need to merge/convert traces!)

% mpirun -np 8 tau_exec ./lu.W.8
% tau_treemerge.pl; module load java ;
% tau2slog2 tau.trc tau.edf -o app.slog2
% jumpshot app.slog2 &
```

Jumpshot Trace Visualizer in TAU

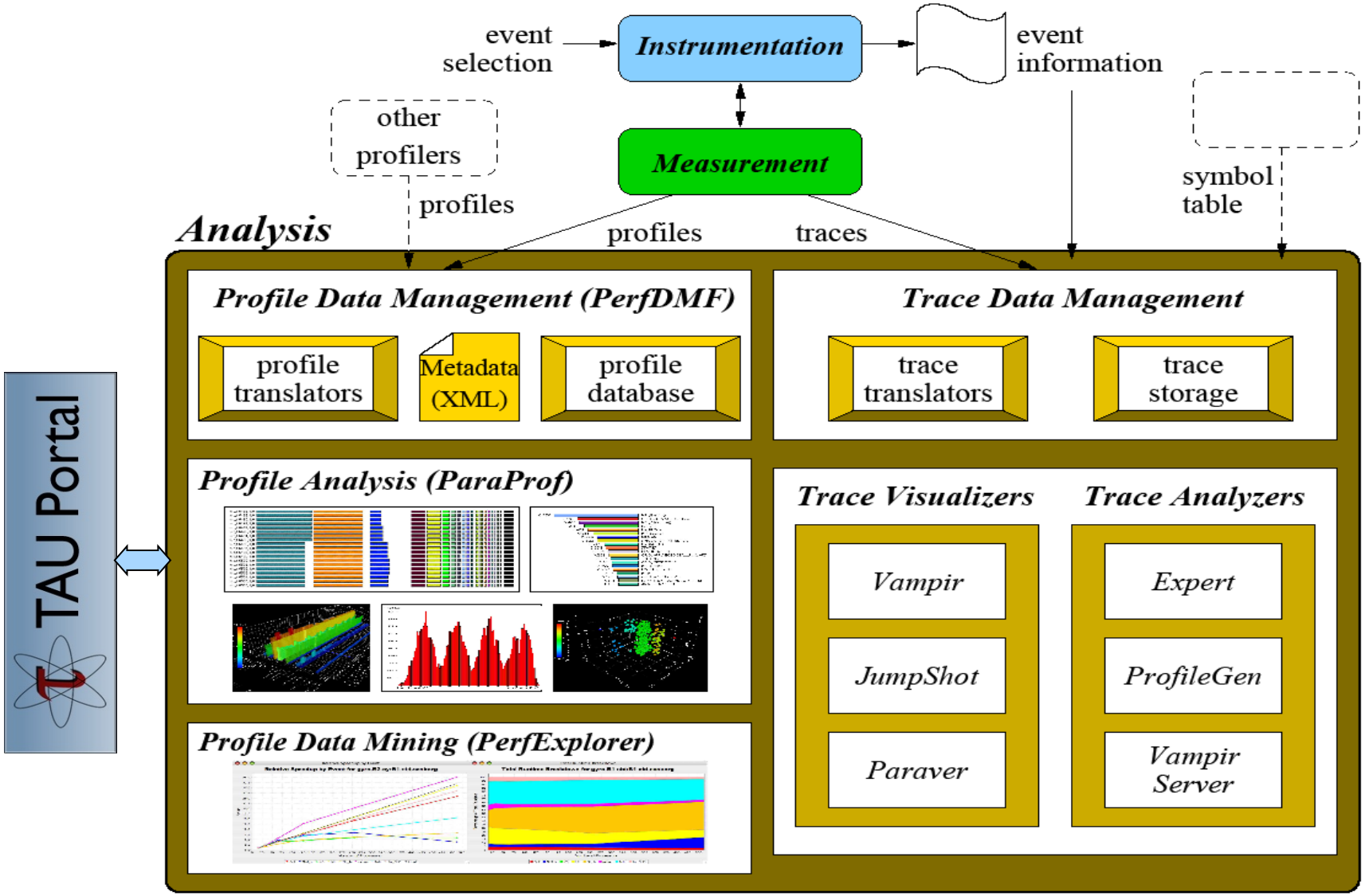


Tracing Communication in Jumpshot



% tau init

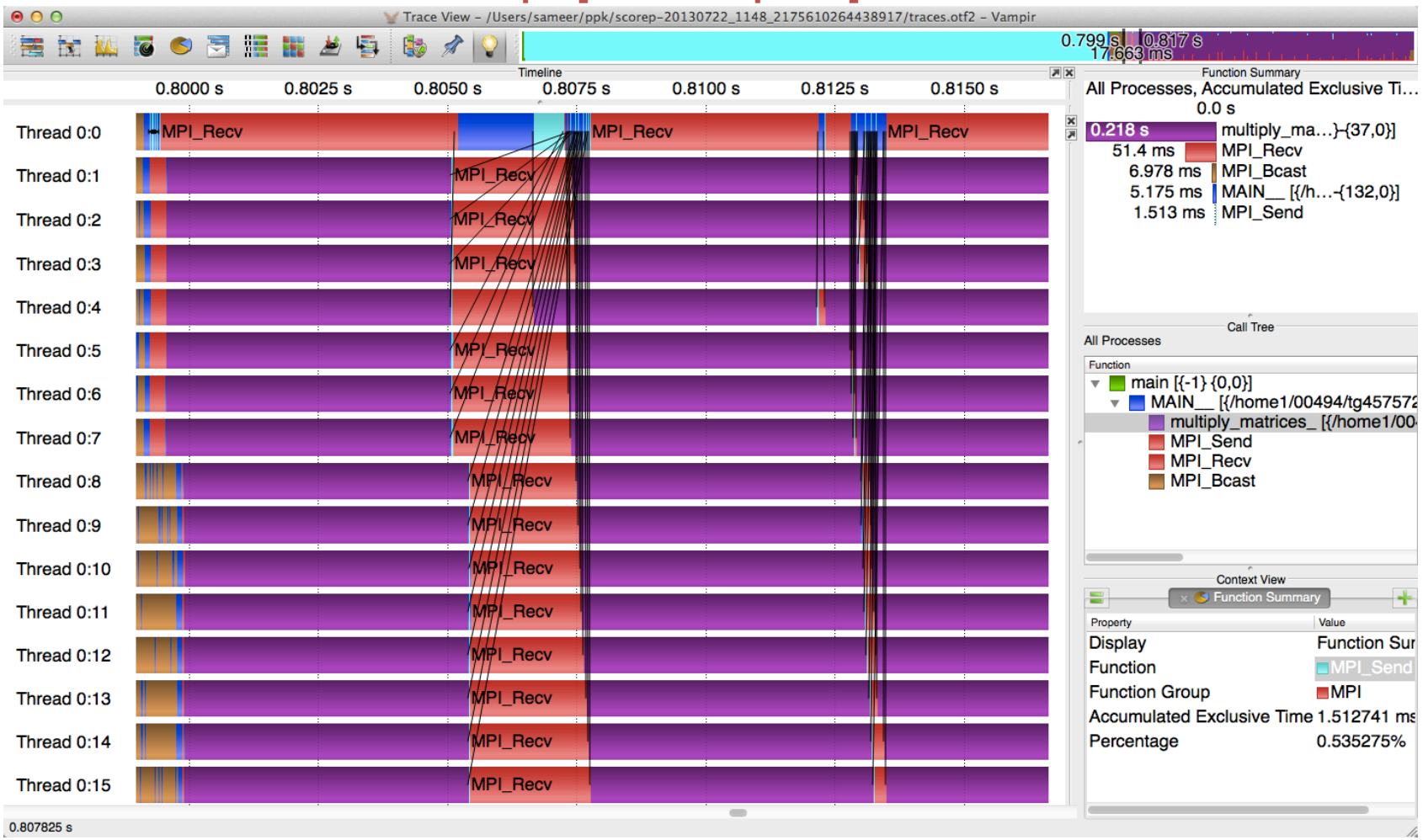
Performance Analysis



Visualizing Event Traces with Vampir

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

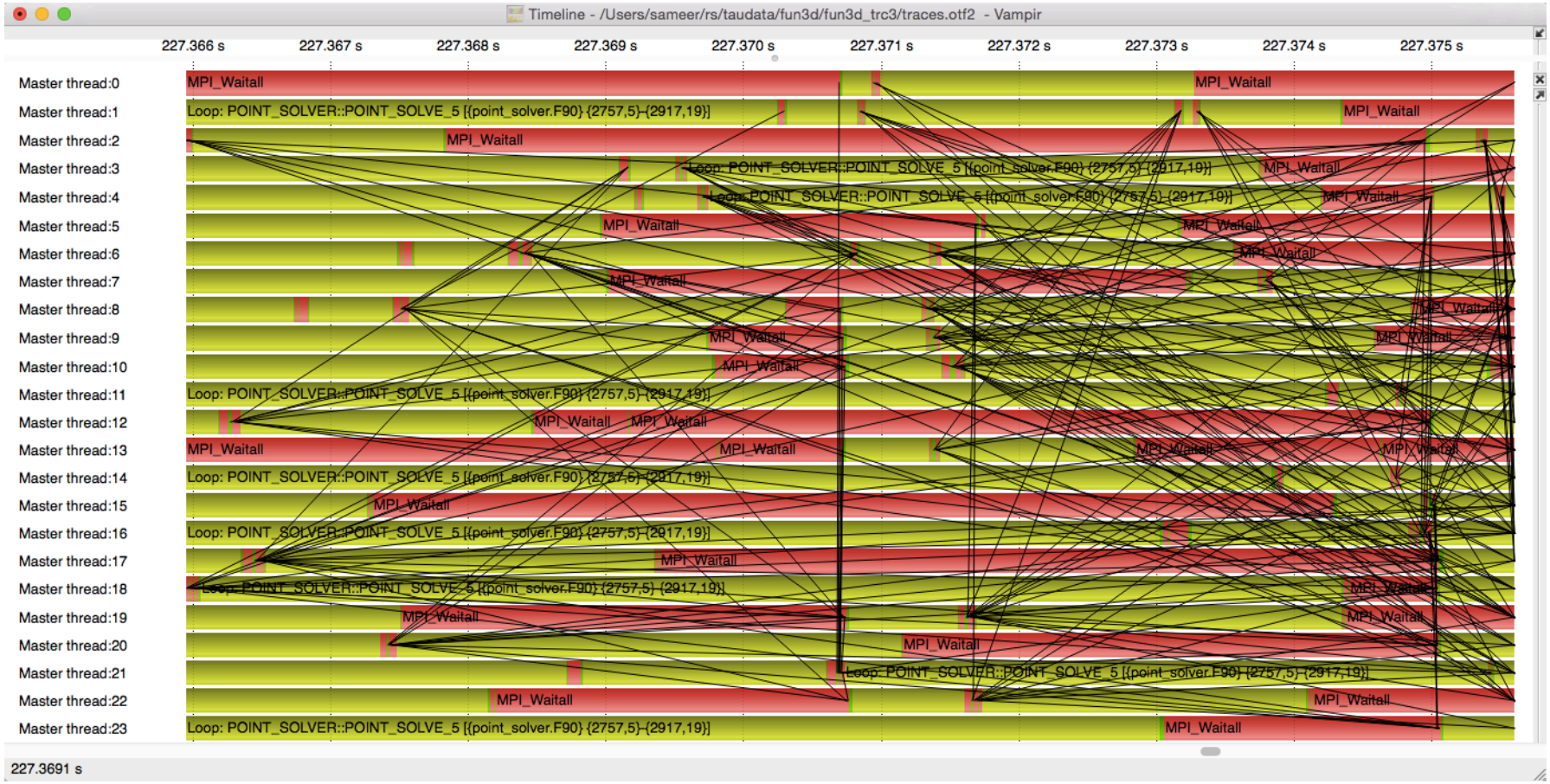
Event trace visualized in Vampir [www.vampir.eu]



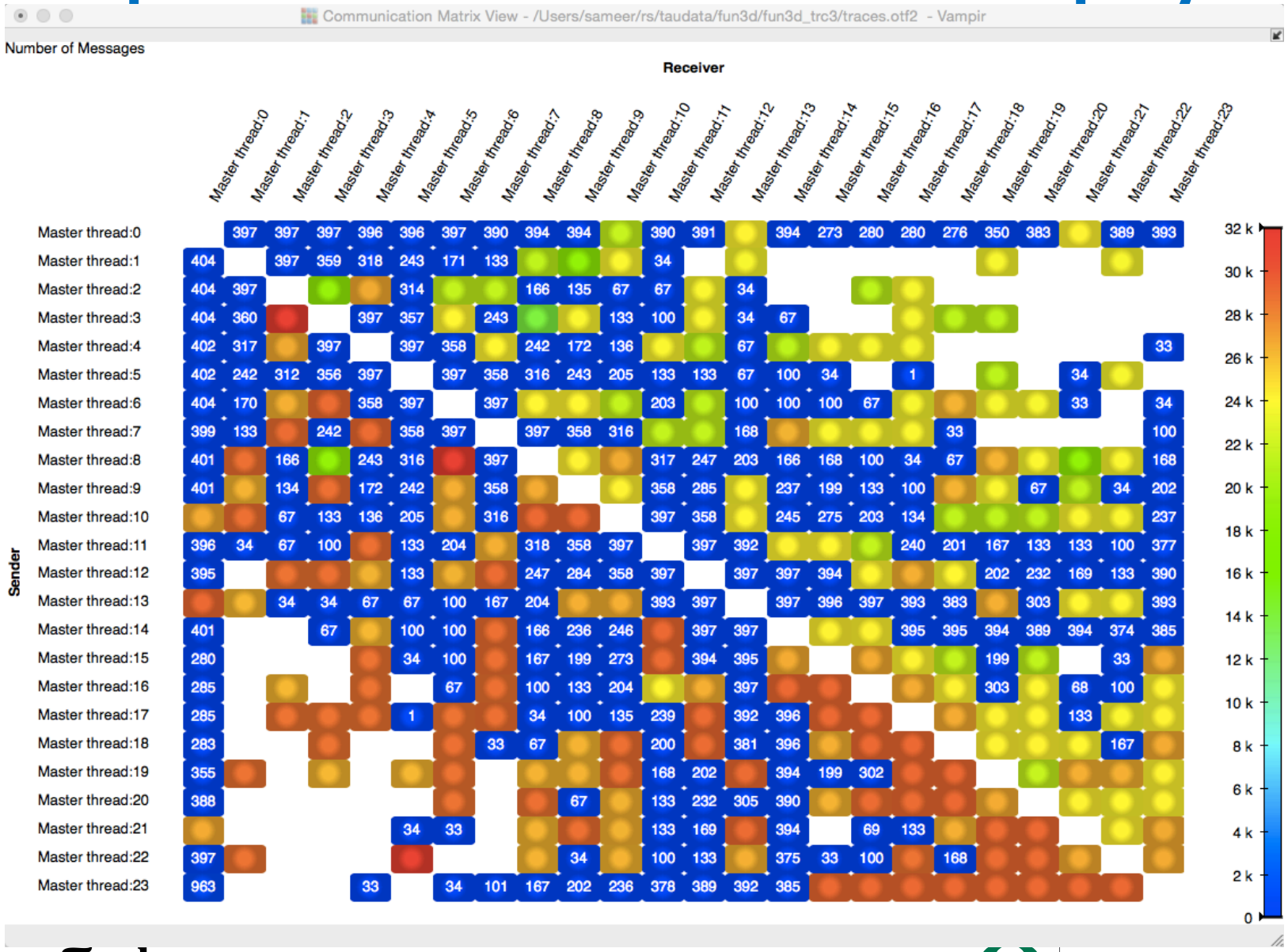
Vampir [T.U. Dresden] Timeline Display

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

Vampir Timeline Display



Vampir Communication Matrix Display



Tools: TAU Commander

TAU Commander

Universal tool or integrated toolkit

Unbiased, accurate measurements

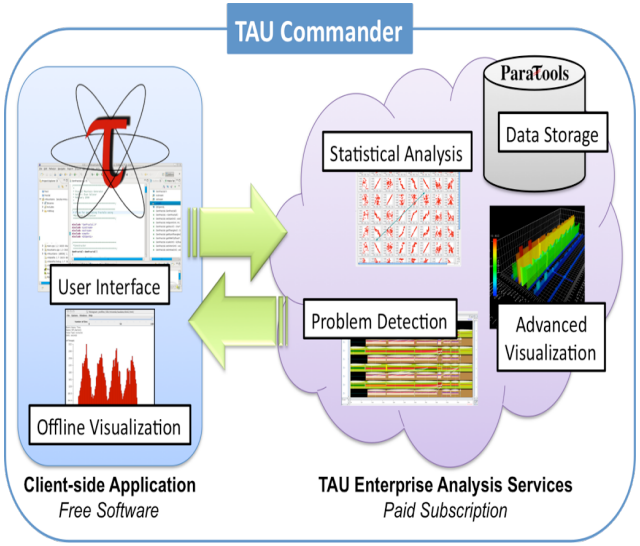
- File I/O: serial and parallel
- Communication: inter- and intra-node
- Memory: allocation and access
- CPU: vectorization, cache utilization, etc.

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***

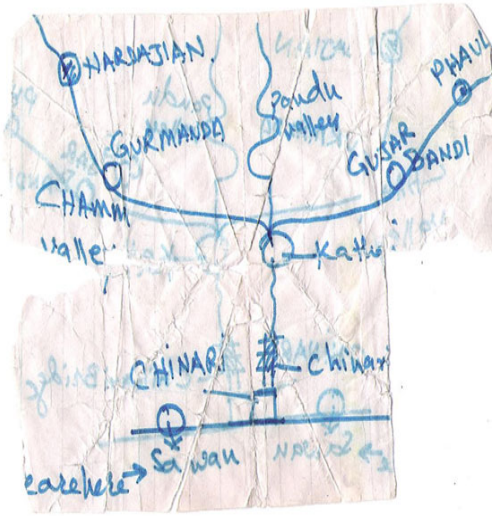


TAU Commander's Approach

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

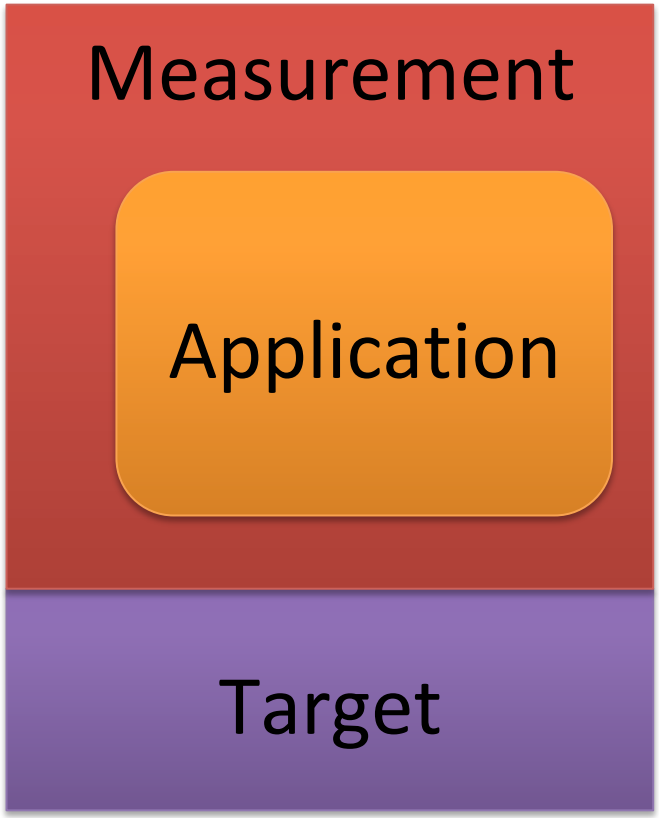
- Installed software
- Available compilers
- Host architecture/ OS

Application

- MPI, OpenMP, CUDA, OpenACC, etc.

Measurement

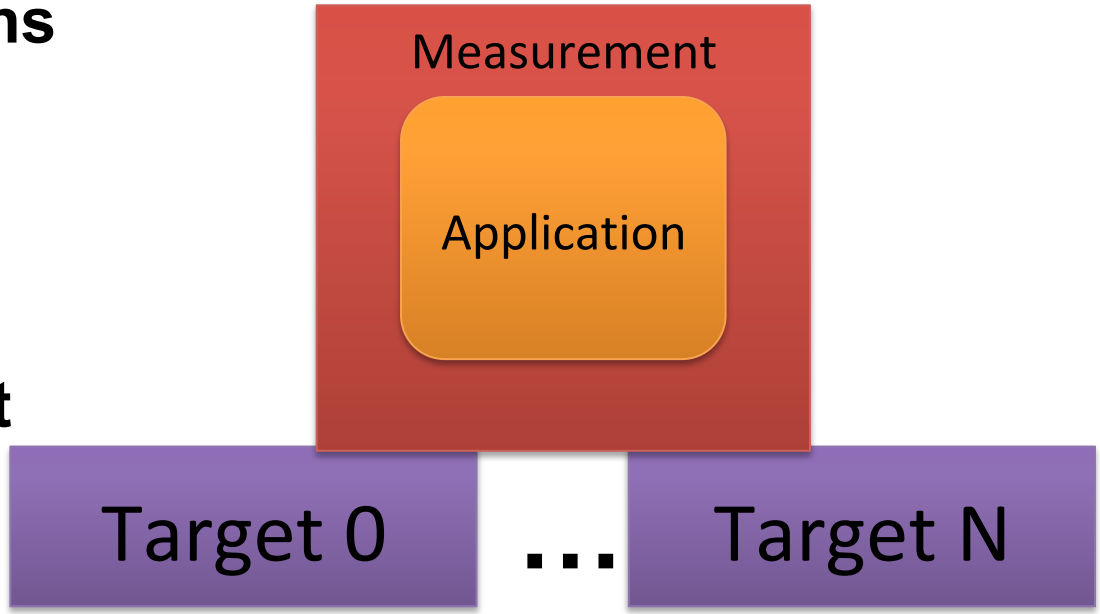
- Profile, trace, or both
- Sample, source inst…



Experiment = (Target, Application, Measurement)

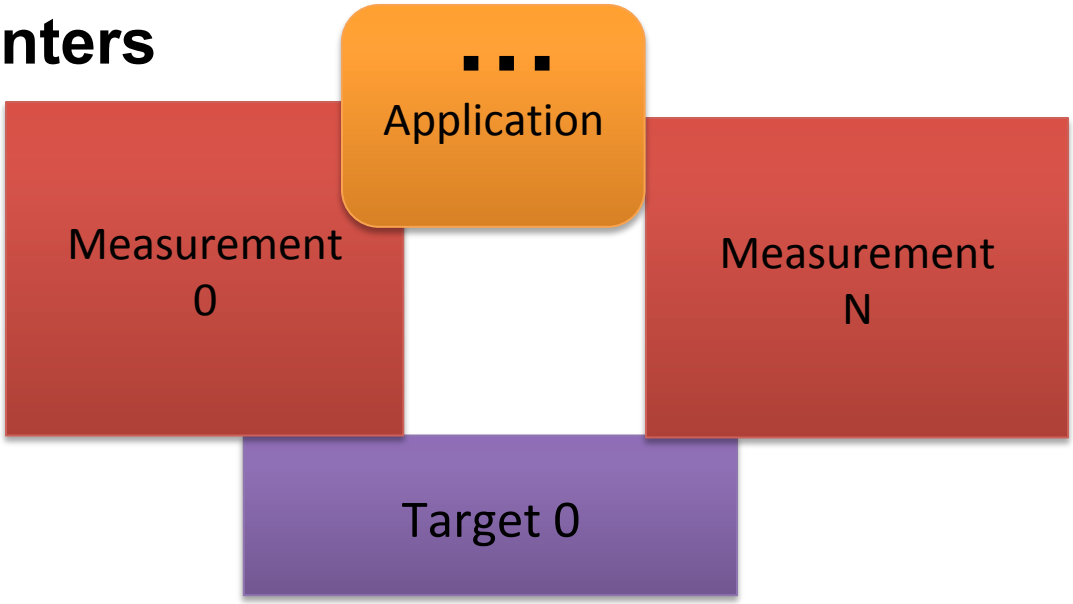
Which platform is best for my application?

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



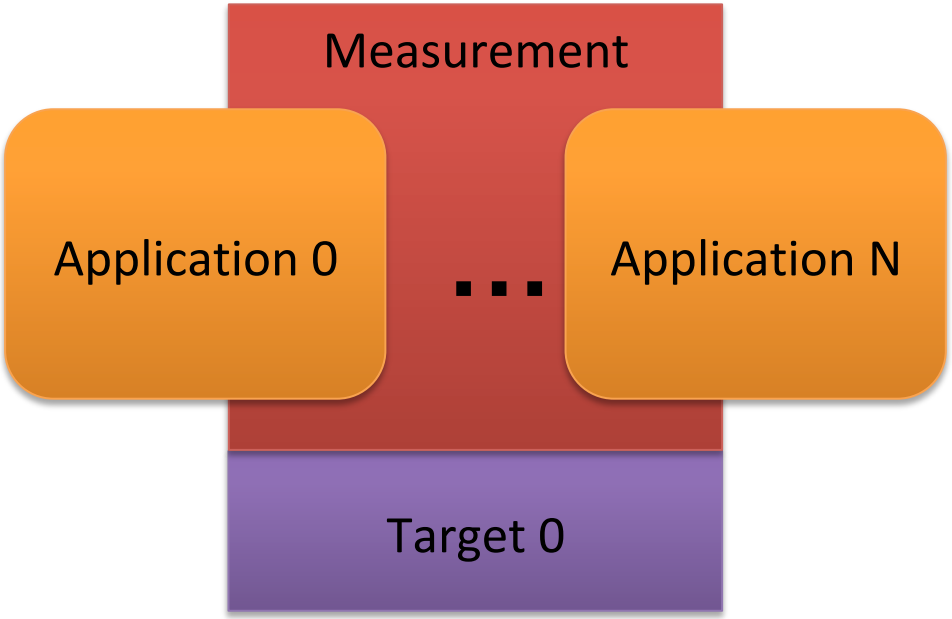
What are the performance characteristics of my application?

- 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 init --mpi
2. tau mpif90 *.f90 -o foo
3. tau mpirun -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 --mpi
```

- 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

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

```
jlinford@cori09 ~/workspace/openshmem17/applications/ISx $ tau initialize --shmem
[TAU] Cray C++ compiler '/opt/cray/pe/craype/2.5.7/bin/CC' wraps
[TAU]     '/opt/intel/compilers_and_libraries_2017.2.174/linux/bin/intel64/icpc'
[TAU] Cray Fortran compiler '/opt/cray/pe/craype/2.5.7/bin/ftn' wraps
[TAU]     '/opt/intel/compilers_and_libraries_2017.2.174/linux/bin/intel64/ifort'
[TAU] Cray C compiler '/opt/cray/pe/craype/2.5.7/bin/cc' wraps
[TAU]     '/opt/intel/compilers_and_libraries_2017.2.174/linux/bin/intel64/icc'
[TAU] Cray MPI C compiler '/opt/cray/pe/craype/2.5.7/bin/cc' wraps
[TAU]     '/opt/intel/compilers_and_libraries_2017.2.174/linux/bin/intel64/icc'
[TAU] Cray MPI C++ compiler '/opt/cray/pe/craype/2.5.7/bin/CC' wraps
[TAU]     '/opt/intel/compilers_and_libraries_2017.2.174/linux/bin/intel64/icpc'
[TAU] Cray MPI Fortran compiler '/opt/cray/pe/craype/2.5.7/bin/ftn' wraps
[TAU]     '/opt/intel/compilers_and_libraries_2017.2.174/linux/bin/intel64/ifort'
[TAU] Cray SHMEM C compiler '/opt/cray/pe/craype/2.5.7/bin/cc' wraps
[TAU]     '/opt/intel/compilers_and_libraries_2017.2.174/linux/bin/intel64/icc'
[TAU] Cray SHMEM C++ compiler '/opt/cray/pe/craype/2.5.7/bin/CC' wraps
[TAU]     '/opt/intel/compilers_and_libraries_2017.2.174/linux/bin/intel64/icpc'
[TAU] Cray SHMEM Fortran compiler '/opt/cray/pe/craype/2.5.7/bin/ftn' wraps
[TAU]     '/opt/intel/compilers_and_libraries_2017.2.174/linux/bin/intel64/ifort'
[TAU] Created a new project named 'ISx'.
[TAU] Added application 'ISx' to project configuration 'ISx'.
[TAU] Added target 'cori09' to project configuration 'ISx'.
[TAU] Added measurement 'sample' to project configuration 'ISx'.
[TAU] Added measurement 'profile' to project configuration 'ISx'.
[TAU] Added measurement 'trace' to project configuration 'ISx'.
[TAU] Created a new experiment 'cori09-ISx-sample'
[TAU] Installing PDT to '/global/project/projectdirs/m88/jlinford/taucmdr-test/system/pdt/77f947dd'
[TAU] Using PDT source archive '/global/project/projectdirs/m88/jlinford/taucmdr-test/system/src/pdt.tgz'
[TAU] Checking contents of '/global/project/projectdirs/m88/jlinford/taucmdr-test/system/src/pdt.tgz'
[TAU] Completed in 8.276 seconds
[TAU] Extracting '/global/project/projectdirs/m88/jlinford/taucmdr-test/system/src/pdt.tgz' to create
[TAU]     '/dev/shm/tmpQl6qTD/./pdtoolkit-3.24'
[TAU] Completed in 5.216 seconds
[TAU] Configuring PDT...
[TAU] Completed in 17.439 seconds
[TAU] Compiling PDT...
[TAU] Completed in 6.394 seconds
[TAU] Installing PDT...
[TAU] Completed in 0.115 seconds
[TAU] Checking installed files...
[TAU] Completed in 0.115 seconds
[TAU] Setting file permissions...
[TAU] Completed in 0.136 seconds
[TAU] Verifying PDT installation...
[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 26.358 seconds
[TAU] Compiling and installing TAU...
[TAU] 5.0 seconds [CPU: 32.4
```

Compiler detection

Project initialization

Download and install PDT

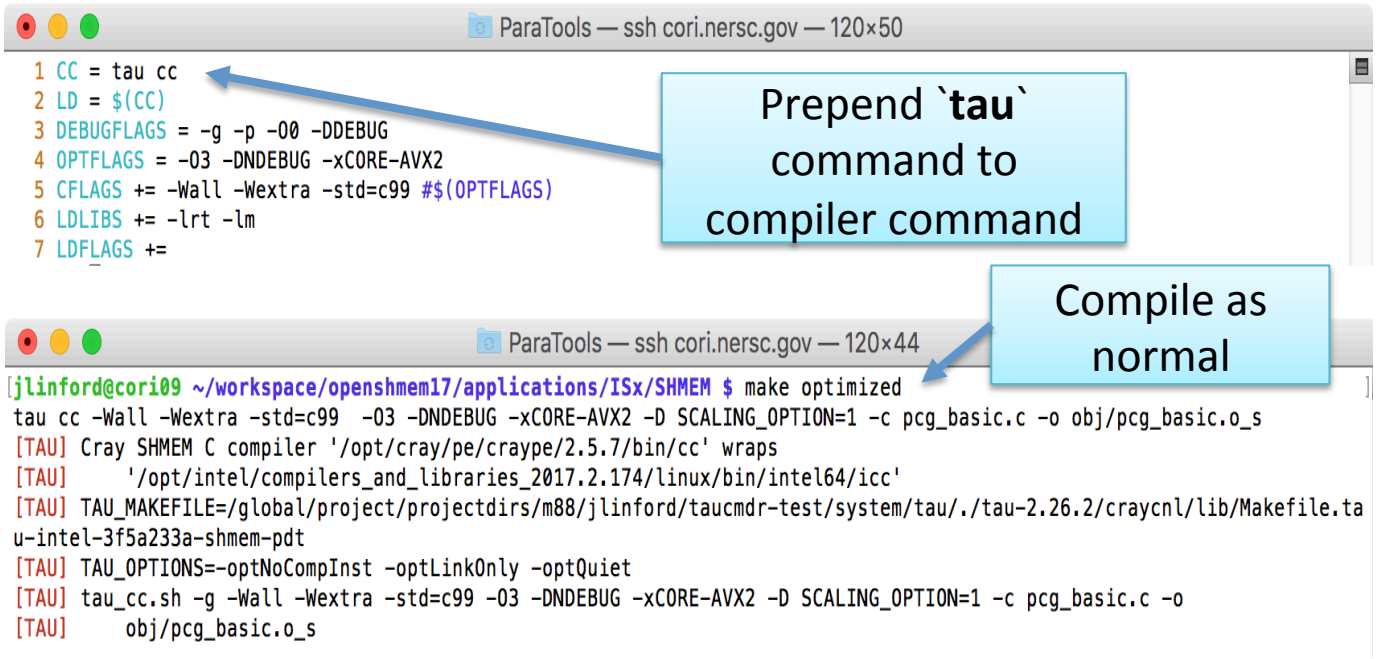
TAU installation progress

Project Dashboard (`tau dashboard`)

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

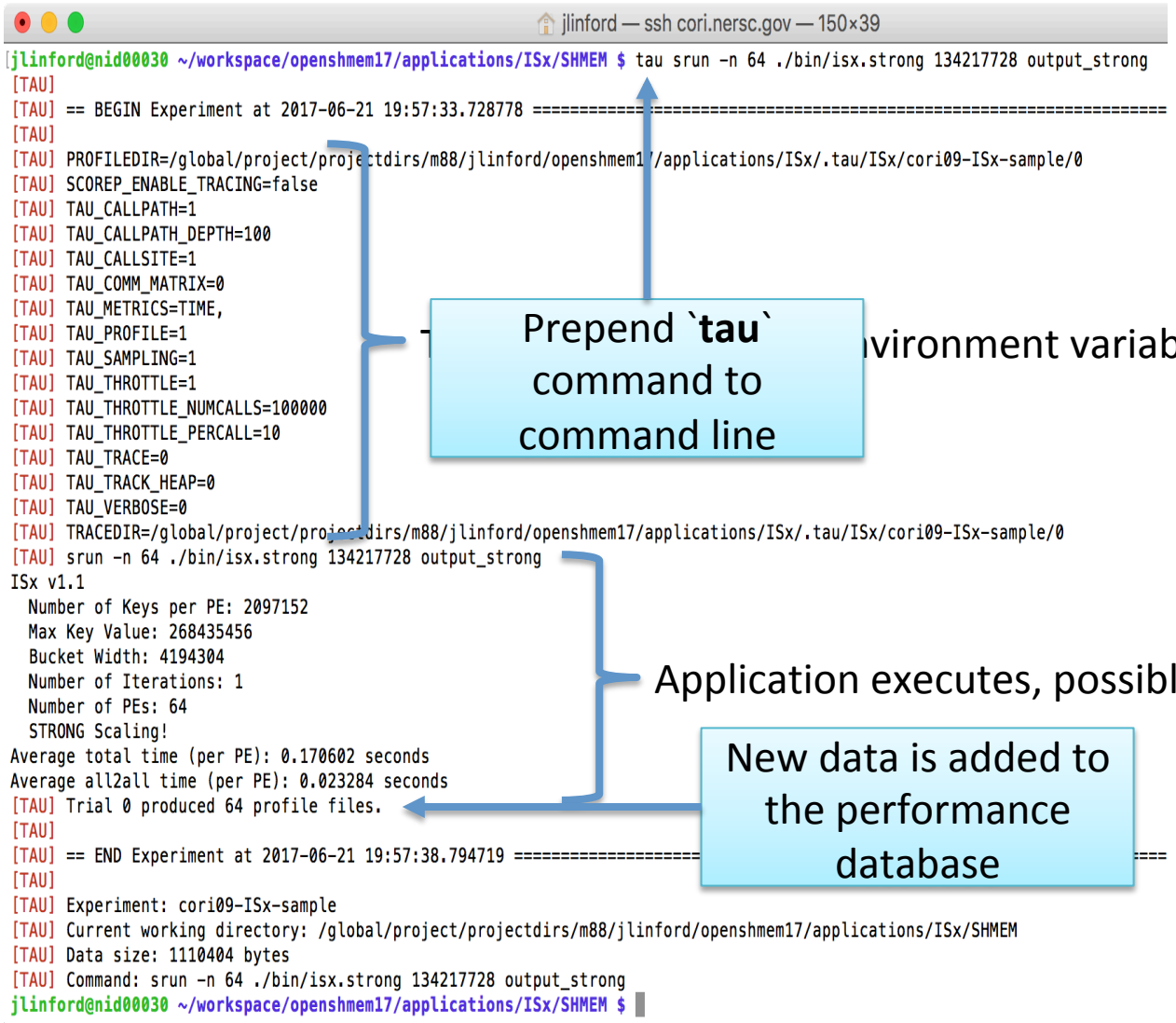
```
== Project Configuration (/global/project/projectdirs/m88/jlinford/openshmem17/applications/ISx/.tau/project.json) ==

+------+---------+--------------+-----------------------+---------------+
| 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



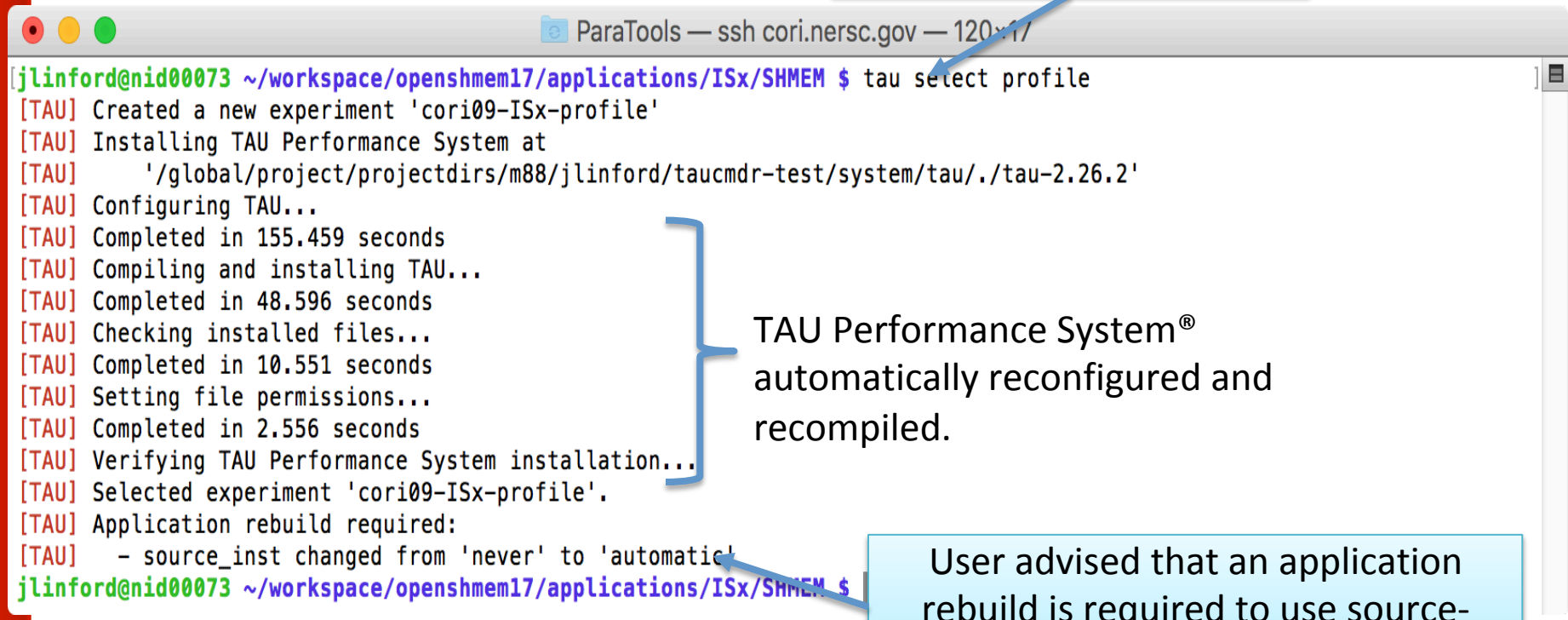
Step 4: Use `tau` to view data (`tau show`)

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



TAU Performance System® automatically reconfigured and recompiled.

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

TAU Commander at NCAR

```
On Cheyenne
% source ~sshende/pkgs/tau.bashrc
% which tau
% cd workshop/matmult
% tau init --mpi
% make clean;
% make F90='tau mpif90'
% tau mpirun -np 64 ./matmult
% tau show
And try the examples. Try:
% tau --help
% tau meas edit --help
```

TAU and PDT for Source Instrumentation

```
On Cheyenne
% source ~sshende/pkgs/tau.bashrc
% which tau
% cd workshop/matmult
% tau init -mpi
% tau dash
% tau select profile
% make clean
% make F90='tau mpif90'
% tau mpirun -np 8 ./matmult
% module load java
% 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)
```

Use Compiler-Based Instrumentation

```
% tau init -mpi
% tau dash
% tau meas edit profile --compiler-inst always
% tau select profile
% make CC=`tau mpicc'
% tau mpirun -np 16 ./a.out
% tau show
% tau show -help
% tau show -profile-tools pprof | more
```

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.
```

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

CEA, France

Partners:

University of Oregon

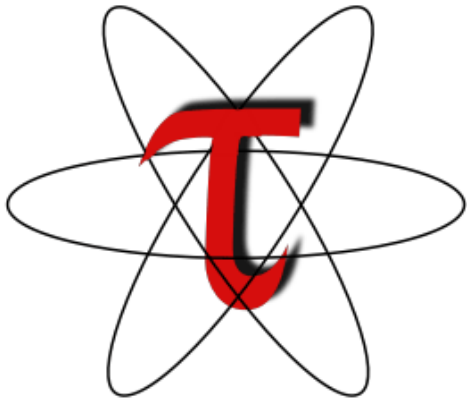
ParaTools, Inc.

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

Runtime Environment Variables

Environment Variable	Default	Description
TAU_TRACE	0	Setting to 1 turns on tracing
TAU_CALLPATH	0	Setting to 1 turns on callpath profiling
TAU_TRACK_MEMORY_FOOTPRINT	0	Setting to 1 turns on tracking memory usage by sampling periodically the resident set size and high water mark of memory usage
TAU_TRACK_POWER	0	Tracks power usage by sampling periodically.
TAU_CALLPATH_DEPTH	2	Specifies depth of callpath. Setting to 0 generates no callpath or routine information, setting to 1 generates flat profile and context events have just parent information (e.g., Heap Entry: foo)
TAU_SAMPLING	1	Setting to 1 enables event-based sampling.
TAU_TRACK_SIGNALS	0	Setting to 1 generate debugging callstack info when a program crashes
TAU_COMM_MATRIX	0	Setting to 1 generates communication matrix display using context events
TAU_THROTTLE	1	Setting to 0 turns off throttling. Throttles instrumentation in lightweight routines that are called frequently
TAU_THROTTLE_NUMCALLS	100000	Specifies the number of calls before testing for throttling
TAU_THROTTLE_PERCALL	10	Specifies value in microseconds. Throttle a routine if it is called over 100000 times and takes less than 10 usec of inclusive time per call
TAU_CALLSITE	0	Setting to 1 enables callsite profiling that shows where an instrumented function was called. Also compatible with tracing.
TAU_PROFILE_FORMAT	Profile	Setting to “merged” generates a single file. “snapshot” generates xml format
TAU_METRICS	TIME	Setting to a comma separated list generates other metrics. (e.g., ENERGY,TIME,P_VIRTUAL_TIME,PAPI_FP_INS,PAPI_NATIVE_<event>:<subevent>)
TAU_TRACE	0	Setting to 1 turns on tracing
TAU_TRACE_FORMAT	Default	Setting to “otf2” turns on TAU’s native OTF2 trace generation (configure with –otf=download)
TAU_EBS_UNWIND	0	Setting to 1 turns on unwinding the callstack during sampling (use with tau_exec –ebs or TAU_SAMPLING=1)
TAU_TRACK_LOAD	0	Setting to 1 tracks system load on the node
TAU_SELECT_FILE	Default	Setting to a file name, enables selective instrumentation based on exclude/include lists specified in the file.

Runtime Environment Variables (contd.

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