A Cookbook Approach to TAU

George R Carr Jr, NOAA GSD/CSC
Raghu Reddy, CSC, NOAA NCEP/CSC
Sameer Shende, Univ of Oregon

Outline

- The basics
  - Visualization
  - Instrumentation
- Some advanced items
  - Throttle
  - Selective Instrumentation
- Real Examples (network permitting)
- A bit on installation
- Q & A

What is TAU? (or Why Tau?)

- A set of tools and libraries for instrumenting and analyzing performance of programs
- Capable of analyzing serial, parallel (MPI or OpenMP), hybrid (MPI and OpenMP), and GPU
- Multi-vendor, multi-architecture
- C, C++, Fortran, JAVA, Python, … GPU
- Open Source
- Great community support and training
- Funding: U or Oregon, DOE, NASA, DOD, vendors, others
- Once you know how to use it you can quickly use it again, and again, and again.

What does TAU consist of?

- Instrumentation tools.
- Basic instrumentation stubs indicating how you want your codes instrumented
- Wrapper scripts to make it easy to instrument codes
- Libraries that instrumented code uses
- Post-processing tools for visualizing the data
- A large bag of tools
  - You only need to start out with a small view of the set of tools
  - Think of MPI – large number of routines but you only need to know 6 or 7 calls to do a lot

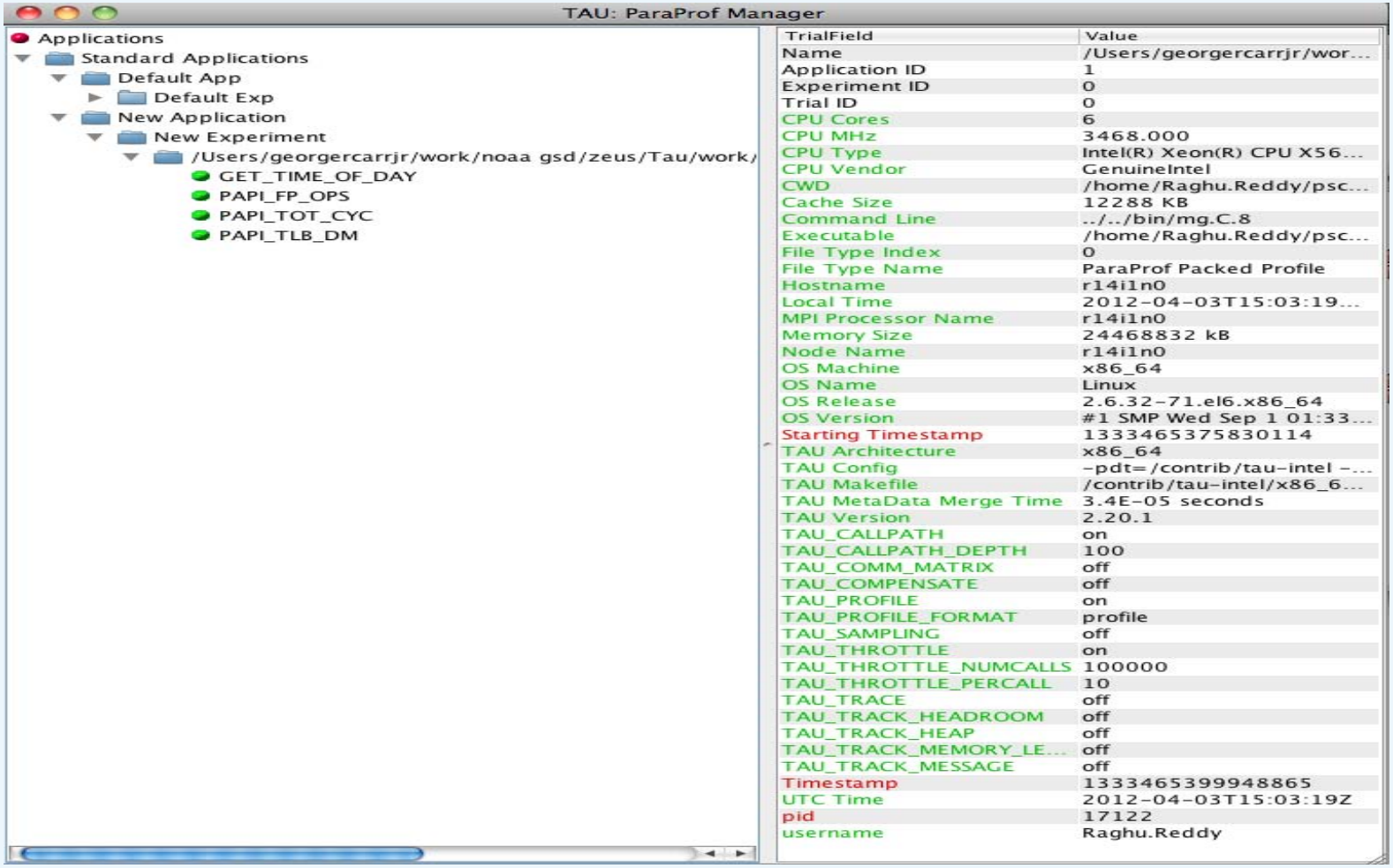
TAU Features

- Very powerful and flexible
- Open source
- Supports automatic profiling
- Support tracing (Vampir, Paraver, JumpShot)
- Supports automatic profiling using either PDT, compiler, and manual instrumentation
  - Automatic function level instrumentation
  - Automatic DO loop instrumentation
  - Selective instrumentation
    - Via hand or parameter settings
- Can be configured to use PAPI where available

Seeing the Results

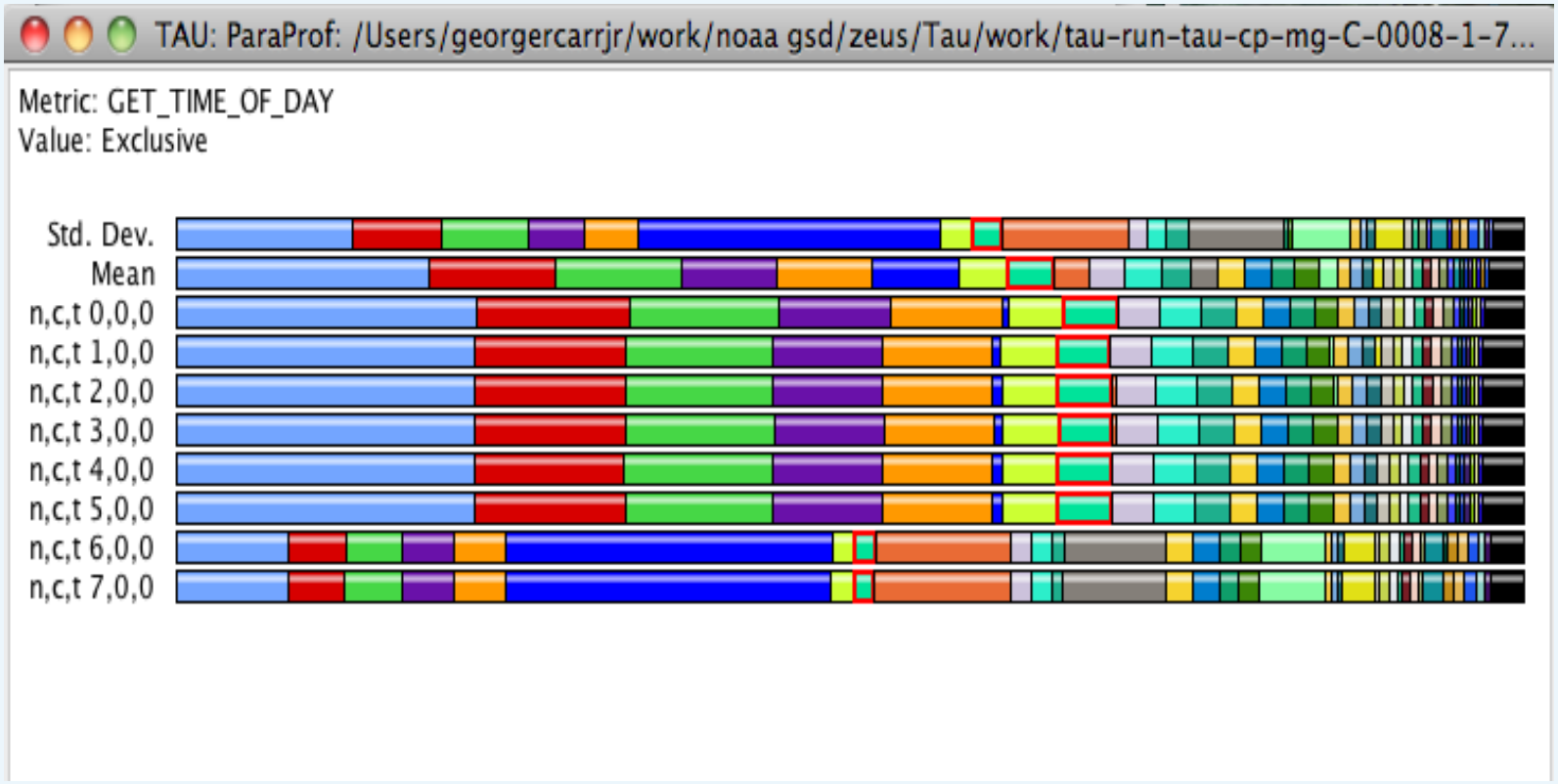
- Text tool pprof
  - Text output
- Graphical tool paraprof
  - 2D and 3D
  - Java app generally run from workstation
  - Call Graph
  - Very light weight, easy to use and understand
  - Very interactive mouse controls

Paraprof Manager

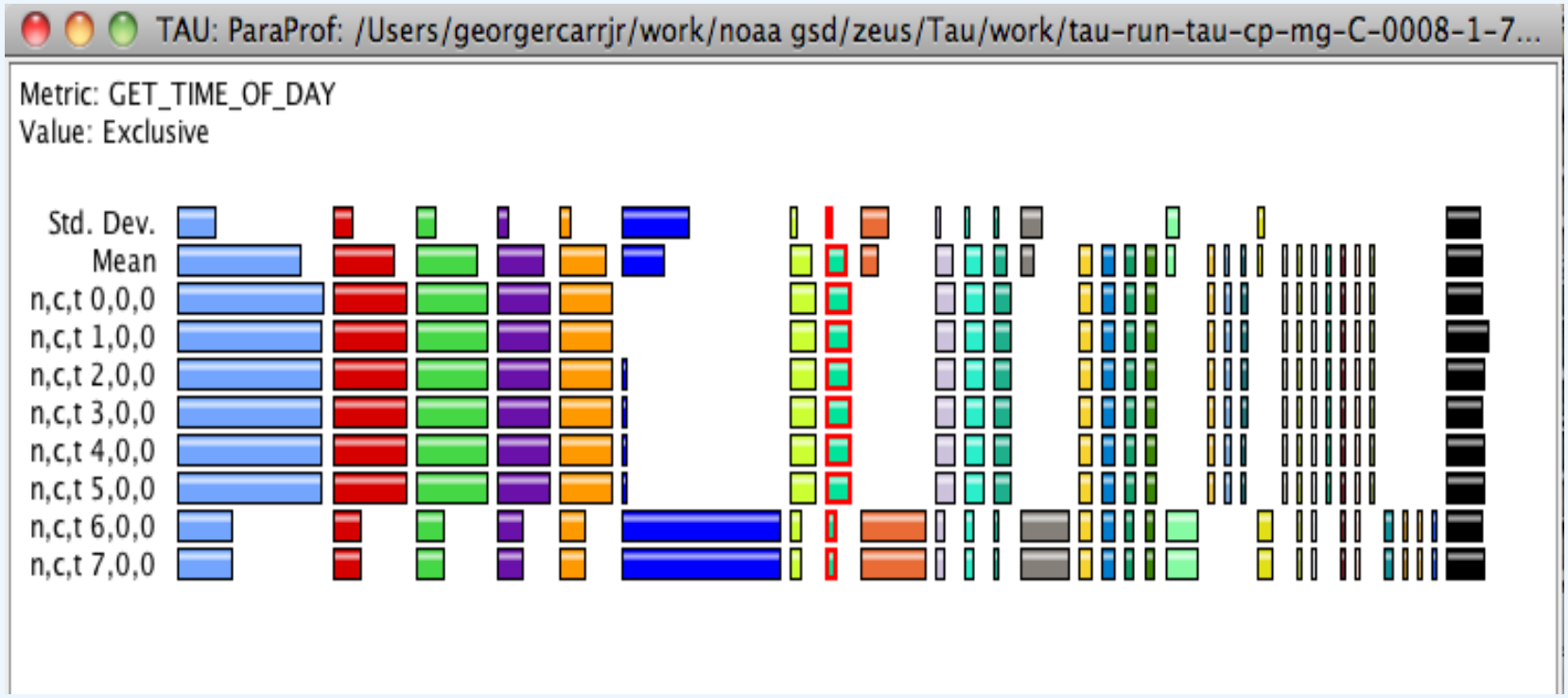


TrialField	Value
Name	/Users/georgercarrjr/wor...
Application ID	1
Experiment ID	0
Trial ID	0
CPU Cores	6
CPU MHz	3468.000
CPU Type	Intel(R) Xeon(R) CPU X56...
CPU Vendor	GenuineIntel
CWD	/home/Raghu.Reddy/psc...
Cache Size	12288 KB
Command Line	../../bin/mg.C.8
Executable	/home/Raghu.Reddy/psc...
File Type Index	0
File Type Name	ParaProf Packed Profile
Hostname	r14i1n0
Local Time	2012-04-03T15:03:19...
MPI Processor Name	r14i1n0
Memory Size	24468832 kB
Node Name	r14i1n0
OS Machine	x86_64
OS Name	Linux
OS Release	2.6.32-71.el6.x86_64
OS Version	#1 SMP Wed Sep 1 01:33...
Starting Timestamp	1333465375830114
TAU Architecture	x86_64
TAU Config	-pdt=/contrib/tau-intel -...
TAU Makefile	/contrib/tau-intel/x86_6...
TAU MetaData Merge Time	3.4E-05 seconds
TAU Version	2.20.1
TAU_CALLPATH	on
TAU_CALLPATH_DEPTH	100
TAU_COMM_MATRIX	off
TAU_COMPENSATE	off
TAU_PROFILE	on
TAU_PROFILE_FORMAT	profile
TAU_SAMPLING	off
TAU_THROTTLE	on
TAU_THROTTLE_NUMCALLS	100000
TAU_THROTTLE_PERCALL	10
TAU_TRACE	off
TAU_TRACK_HEADROOM	off
TAU_TRACK_HEAP	off
TAU_TRACK_MEMORY_LE...	off
TAU_TRACK_MESSAGE	off
Timestamp	1333465399948865
UTC Time	2012-04-03T15:03:19Z
pid	17122
username	Raghu.Reddy

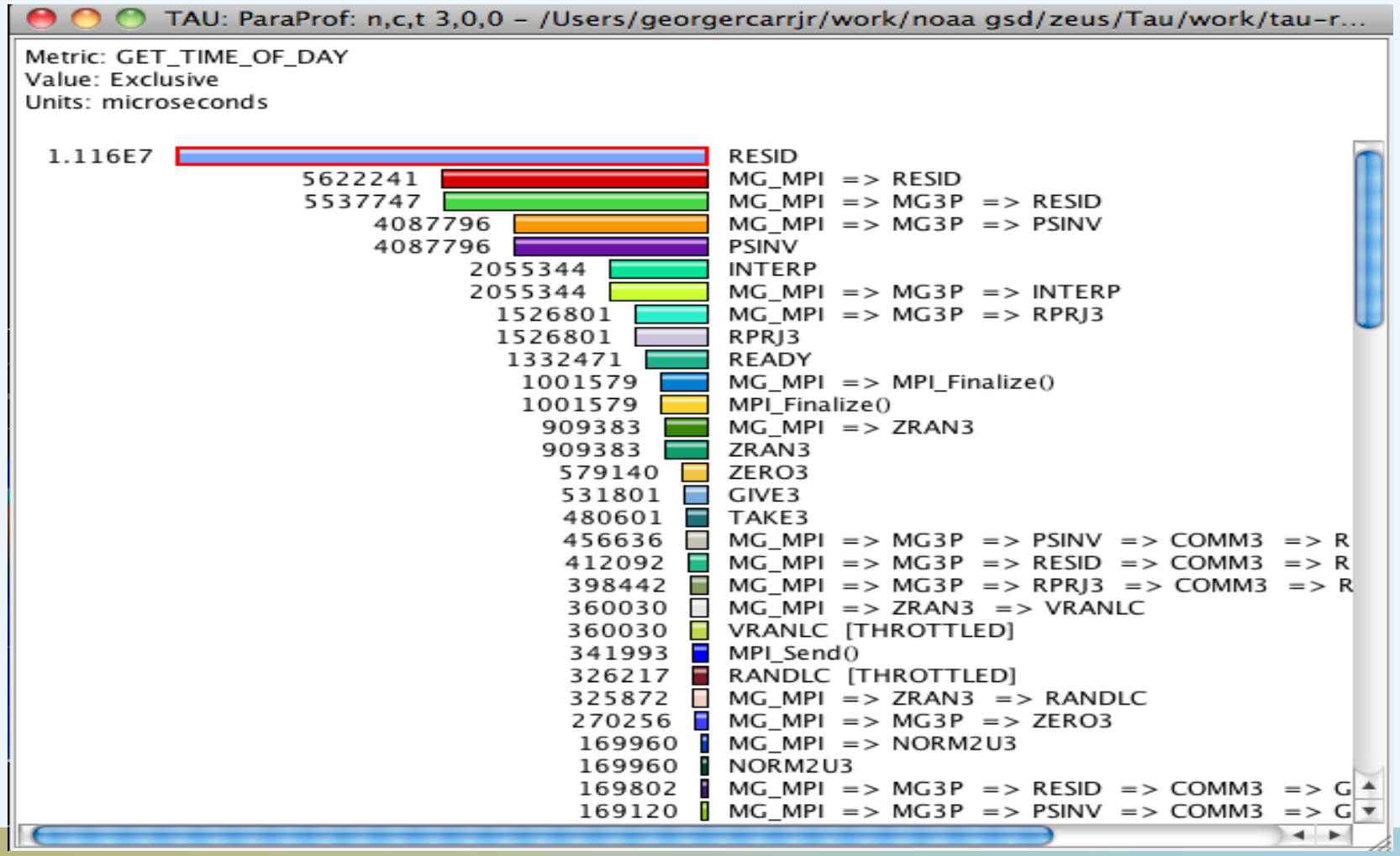
Basic Profile



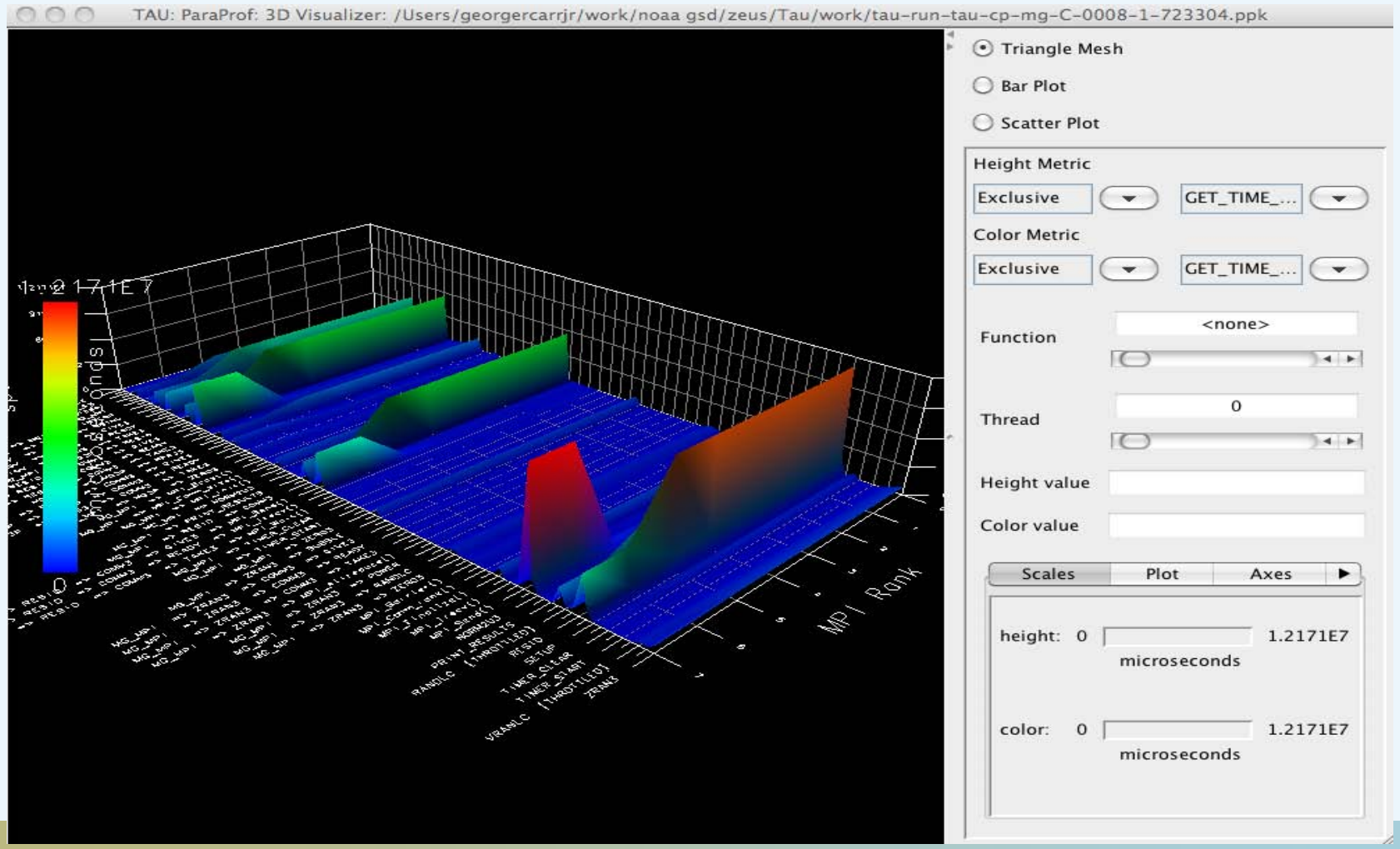
Basic “Stacked” Profile



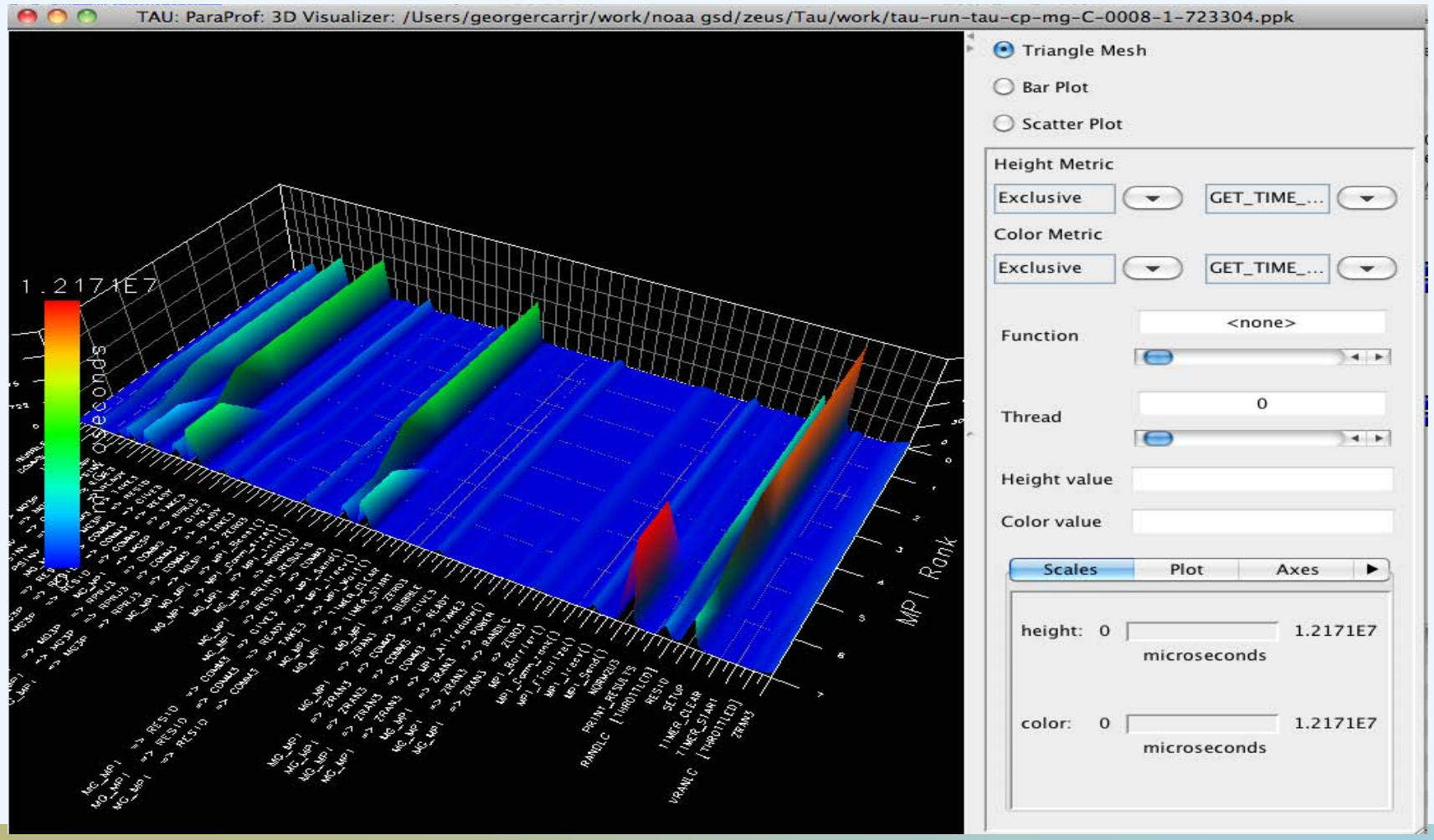
Basic Core Profile



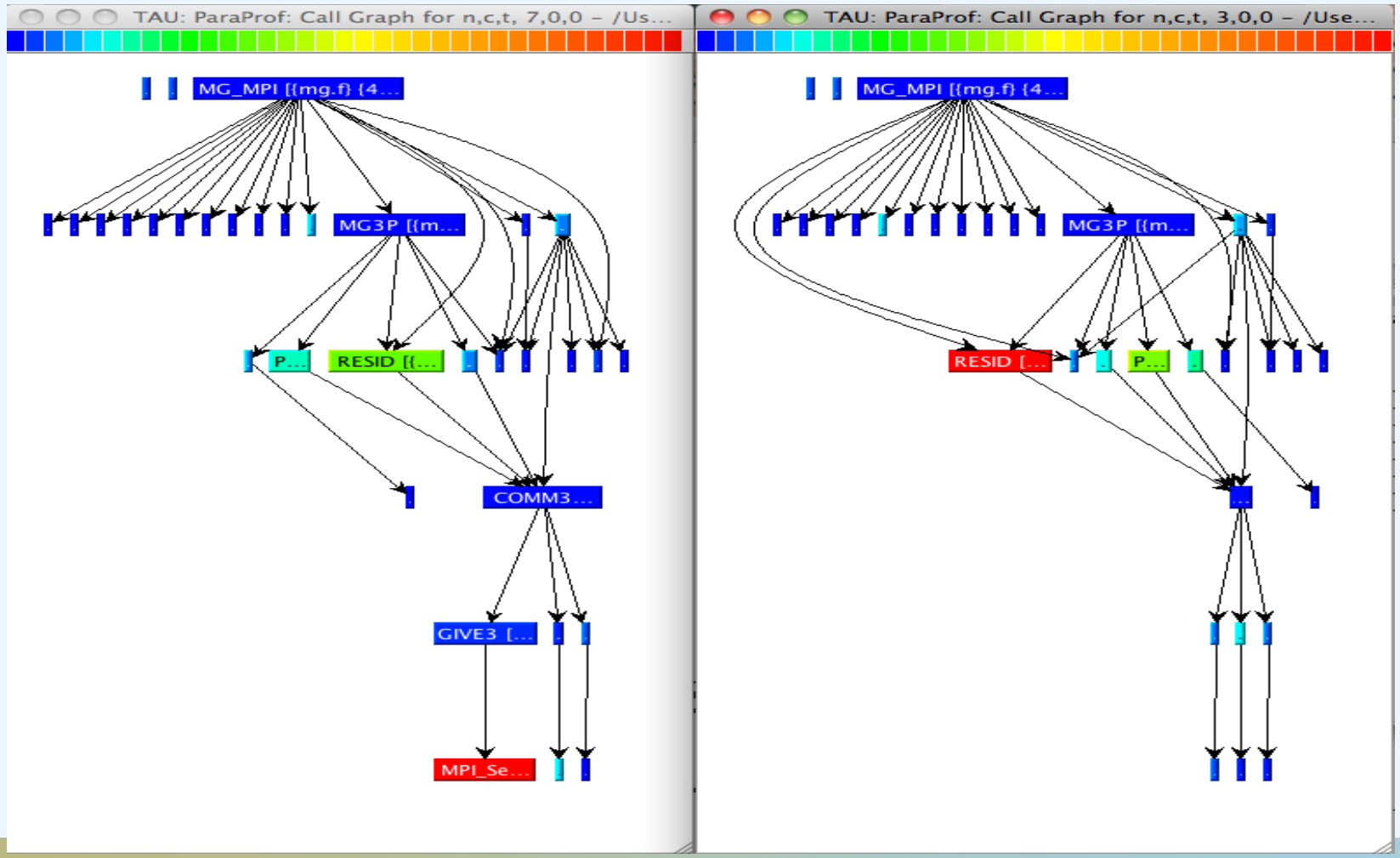
3D View


TAU: ParaProf: 3D Visualizer: /Users/georgercarrjr/work/noaa gsd/zeus/Tau/work/tau-run-tau-cp-mg-C-0008-1-723304.ppk
Triangle Mesh
Bar Plot
Scatter Plot
Height Metric
Exclusive
GET_TIME_...
Color Metric
Exclusive
GET_TIME_...
Function
<none>
Thread
0
Height value
Color value
Scales
Plot
Axes
height: 0
1.2171E7
microseconds
color: 0
1.2171E7
microseconds
MPI Rank

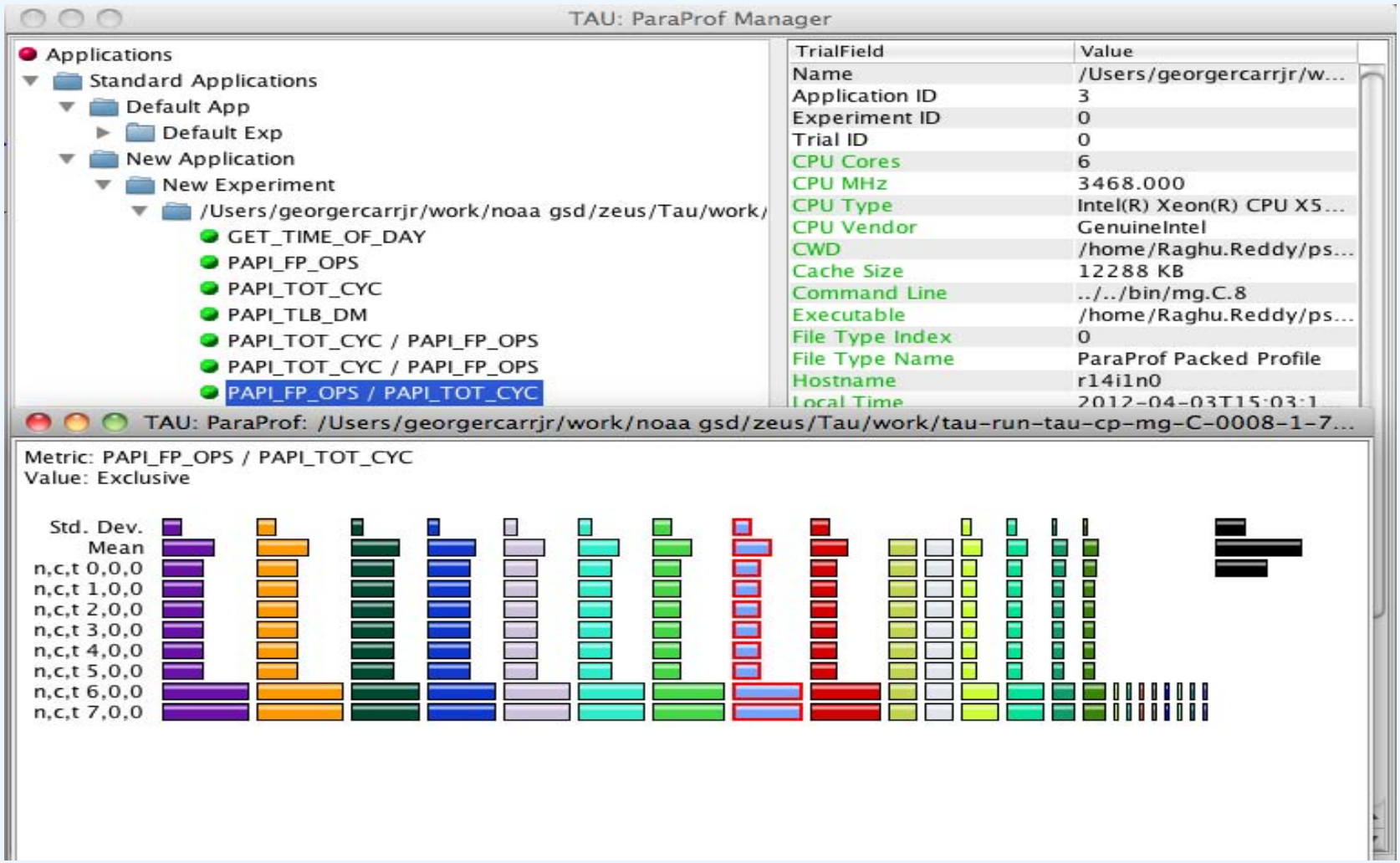
3D Rotation


TAU: ParaProf: 3D Visualizer: /Users/georgercarrjr/work/noaa gsd/zeus/Tau/work/tau-run-tau-cp-mg-C-0008-1-723304.ppk
1.2171E7
Microseconds
MPI Rank
Triangle Mesh
Bar Plot
Scatter Plot
Height Metric
Exclusive
GET_TIME_...
Color Metric
Exclusive
GET_TIME_...
Function
<none>
Thread
0
Height value
Color value
Scales
Plot
Axes
height: 0
1.2171E7
microseconds
color: 0
1.2171E7
microseconds

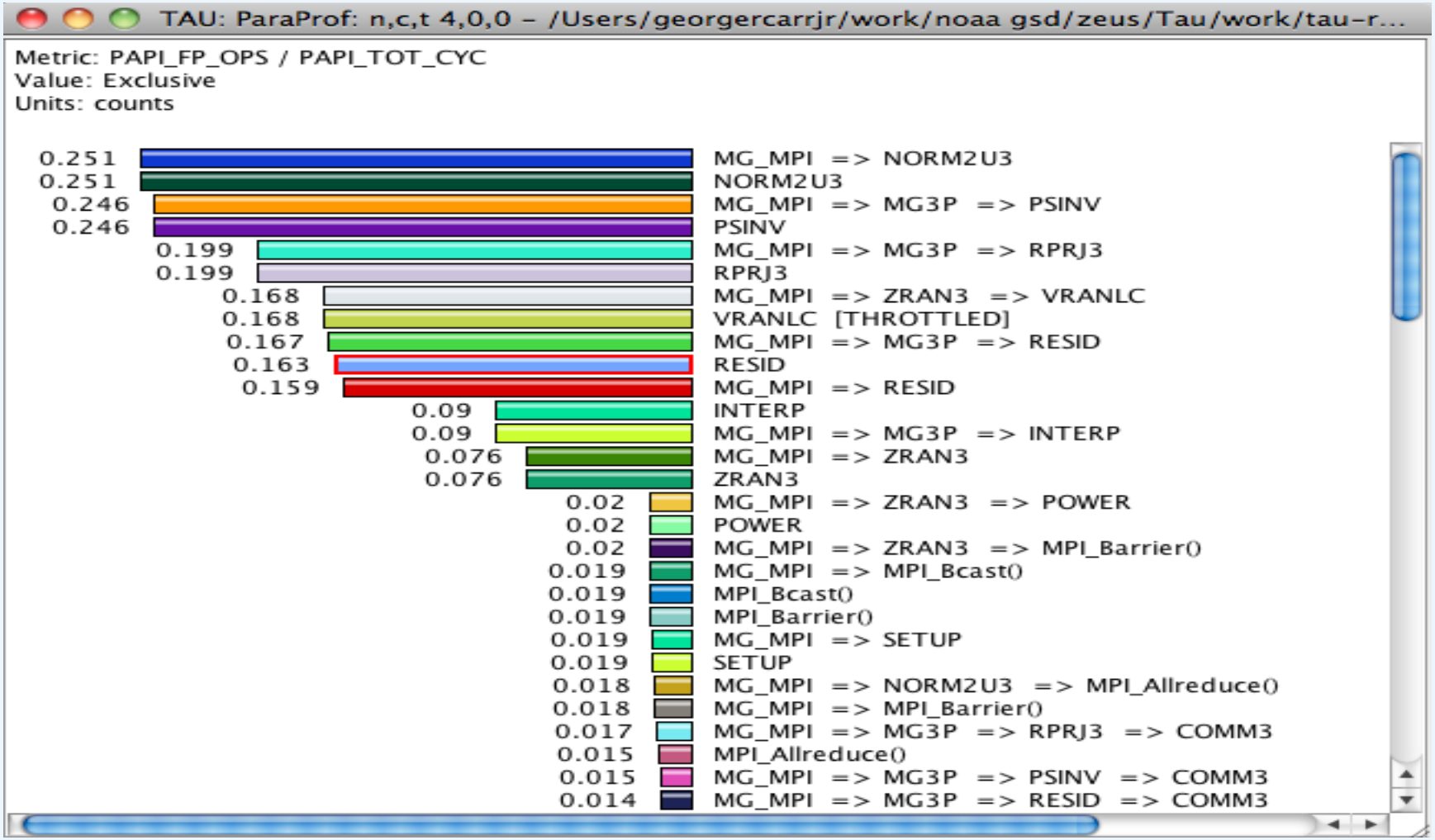
Call Graph


TAU: ParaProf: Call Graph for n,c,t, 7,0,0 – /Us...
MG_MPI [{mg.f} {4...
MG3P [{m...
P...
RESID [{...
COMM3...
GIVE3 [...
MPI_Se...
TAU: ParaProf: Call Graph for n,c,t, 3,0,0 – /Use...
MG_MPI [{mg.f} {4...
MG3P [{m...
RESID [...
P...

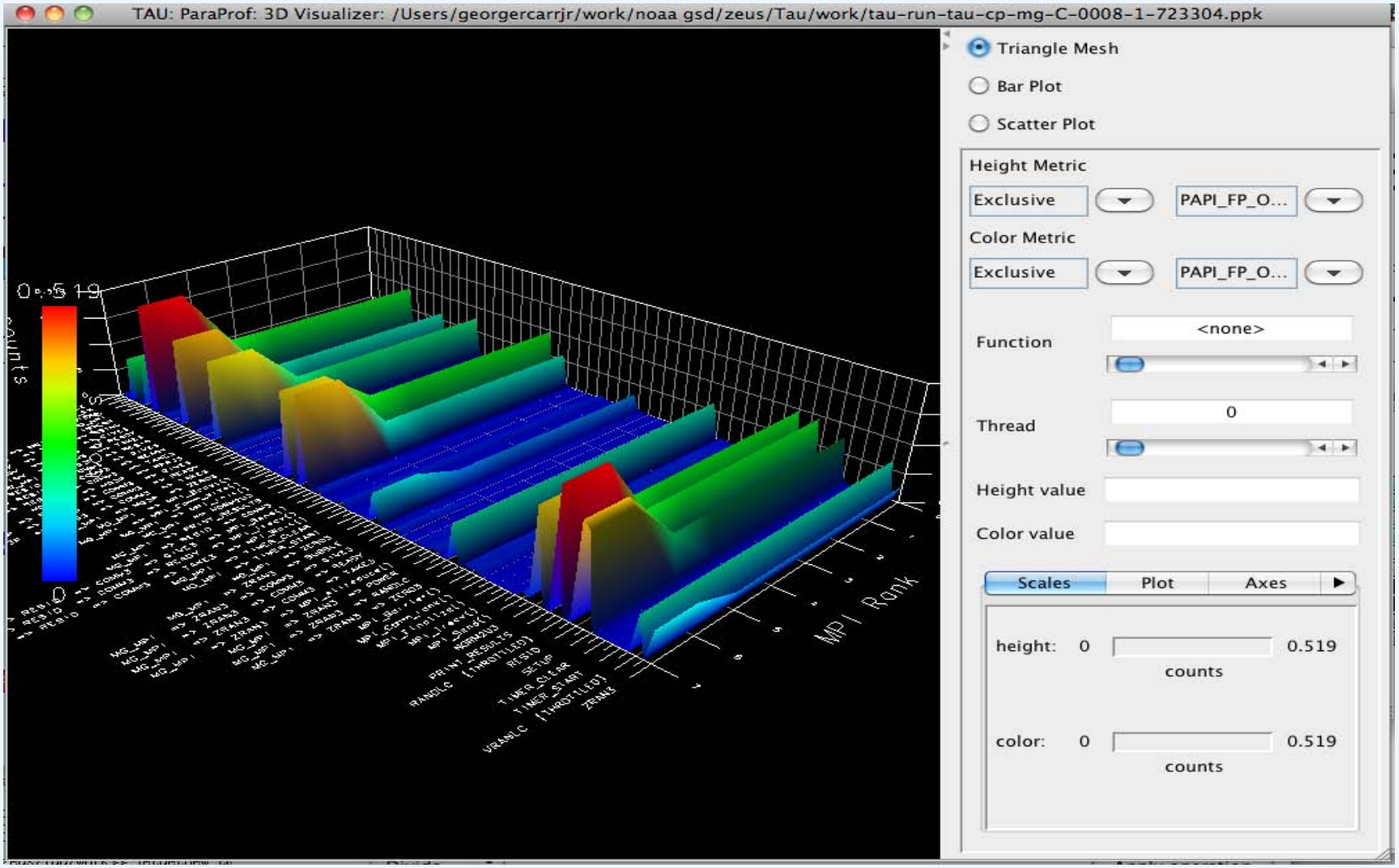
Derived Metrics



Derived Metrics - Core



Derived Metrics – 3D



Output of pprof

- >>pprof
- Reading Profile files in profile.*

- NODE 0;CONTEXT 0;THREAD 0:
- ---
- %Time Exclusive Inclusive #Call #Subrs Inclusive Name
- msec total msec usec/call
- ---
- 100.0 0.727 23,963 1 76 23963403 MG_MPI
- 65.5 0.701 15,703 21 861 747772 MG3P
- 52.7 11,418 12,629 191 191 66123 RESID
- 20.0 3,944 4,798 189 189 25390 PSINV
- 10.7 3 2,569 550 9900 4672 COMM3
- 8.8 1,649 2,112 168 168 12576 RPRJ3

How To Do First Profile

- Must settings
  - Add path to Tau installation directory to search path
  - setenv TAU_MAKEFILE $TAUDIR/tau-2.20.1/x86_64/lib/Makefile.tau-icpc-papi-mpi-pdt
- Optional
  - setenv TAU_OPTIONS "-optVerbose”
  - Many others

My Environment

- TAULIBDIR=/contrib/tau-intel/x86_64/lib
- TAU_ARCH=x86_64
- TAU_INCLUDE_OPTS= -I /contrib/tau-intel/x86_64/include
- TAU_MAKEFILE=/contrib/tau-intel/x86_64/lib/Makefile.tau-icpc-papi-mpi-pdt
- TAU_POST_LINK_OPTS= -L /contrib/tau-intel/x86_64/lib -l TAU
- TAU_ROOT=/contrib/tau-intel/x86_64

Build process transition

- Simple substitution of TAU preprocessor scripts for conventional compiler names
  - ifort –> tau_f70.sh or tau-f90.sh
  - icc -> tau_cc.sh
  - … etc.

Before

```
CXX = CC
F90 = ftn
CFLAGS =
LIBS = -lm
OBJS = f1.o f2.o f3.o … fn.o

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

After

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

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

Example 1: NPB

- qsub -I -X -l procs=12,walltime=0:30:00 –A <account> -q debug

- cd $PBS_O_WORKDIR
- module list
- module use /contrib/modulefiles
- module load tau/2.20.1
- setenv TAU_MAKEFILE $TAULIBDIR/Makefile.tau-icpc-papi-mpi-pdt

- make clean
- make CLASS=C NPROCS=8 mg
- mv bin/mg.C.8 bin/mg-base.C.8

- make clean
- make CLASS=C NPROCS=8 mg MPIF77=tau_f90.sh
- mv bin/mg.C.8 bin/mg-tau.C.8

Example 1: NPB module

```
#%Module#######################################################################
##
##     tau 2.20.1
##
proc ModulesHelp { } {
        puts stderr "Sets up the paths you need to use tau"
}

set sys       [uname sysname]
set base      /contrib/tau-intel
set arch      x86_64
set ver       2.20.1
set path      $base/$ver

   prepend-path PATH $base/$arch/bin
   prepend-path MANPATH $base/man
   prepend-path LD_LIBRARY_PATH $base/$arch/lib
   prepend-path CLASSPATH $base/$arch/lib

   setenv TAU_ROOT       $base/$arch
   setenv TAU_ARCH       $arch
   setenv TAU_INCLUDE_OPTS " -I $base/$arch/include"
   setenv TAU_POST_LINK_OPTS " -L $base/$arch/lib -l TAU"
   setenv TAULIBDIR      $base/$arch/lib
   setenv TAU_MAKEFILE   $base/$arch/lib/Makefile.tau-icpc-papi-mpi-pdt
```

Example 1: NPB execution

- cd $PBS_O_WORKDIR
- module list
- module use /contrib/modulefiles
- module load tau/2.20.1

- mpiexec_mpt -np 8 ../bin/mg-base.C.8
- mpiexec_mpt -np 8 ../bin/mg-tau.C.8

Example 1: NPB Output

- Benchmark completed
- VERIFICATION SUCCESSFUL
- L2 Norm is 0.570673228574 0E-06
- Error is 0.3432373541370E-13

- MG Benchmark Completed.
- Class = C
- Size = 512x 512x 512
- Iterations = 20
- Time in seconds = 20.77
- Total processes = 8
- Compiled procs = 8
- Mop/s total = 7497.60
- Mop/s/process = 937.20
- Operation type = floating point
- Verification = SUCCESSFUL
- Version = 3.3
- Compile date = 10 Aug 2012

Example 1: NPB Tau output

- >> ls
- profile.0.0.0 profile.2.0.0 profile.4.0.0 profile.6.0.0
- profile.1.0.0 profile.3.0.0 profile.5.0.0 profile.7.0.0

- >>pprof
- Reading Profile files in profile.*

- NODE 0;CONTEXT 0;THREAD 0:
- ---
- %Time Exclusive Inclusive #Call #Subrs Inclusive Name
- msec total msec usec/call
- ---
- 100.0 0.727 23,963 1 76 23963403 MG_MPI
- 65.5 0.701 15,703 21 861 747772 MG3P
- 52.7 11,418 12,629 191 191 66123 RESID
- 20.0 3,944 4,798 189 189 25390 PSINV
- 10.7 3 2,569 550 9900 4672 COMM3
- 8.8 1,649 2,112 168 168 12576 RPRJ3

- paraprof --pack ../tau-exmp1.ppk

Example 2: HW Counters

- papi_avail will tell you which papi metrics are available on your machine (in the papi bin directory)
- setenv TAU_METRICS PAPI_FP_OPS:PAPI_TOT_CYC:PAPI_TLB_DM
- No recompilation is needed once code is compiled with TAU instrumentation

- Run it

Output from papi_avail

- Available events and hardware information.
- --------
- PAPI Version : 4.1.2.1
- Vendor string and code : GenuineIntel (1)
- Model string and code : Intel(R) Xeon(R) CPU X5690 @ 3.47GHz (44)
- CPU Revision : 2.000000
- CPUID Info : Family: 6 Model: 44 Stepping: 2
- CPU Megahertz : 3468.000000
- CPU Clock Megahertz : 3468
- Hdw Threads per core : 2
- Cores per Socket : 6
- NUMA Nodes : 2
- CPU's per Node : 12
- Total CPU's : 24
- Number Hardware Counters : 16
- Max Multiplex Counters : 512
- --------
- The following correspond to fields in the PAPI_event_info_t structure.

- Name Code Avail Deriv Description (Note)
- PAPI_L1_DCM 0x80000000 Yes No Level 1 data cache misses
- PAPI_L1_ICM 0x80000001 Yes No Level 1 instruction cache misses
- PAPI_L2_DCM 0x80000002 Yes Yes Level 2 data cache misses
- PAPI_L2_ICM 0x80000003 Yes No Level 2 instruction cache misses
- PAPI_L3_DCM 0x80000004 No No Level 3 data cache misses
- PAPI_L3_ICM 0x80000005 No No Level 3 instruction cache misses
- … and more

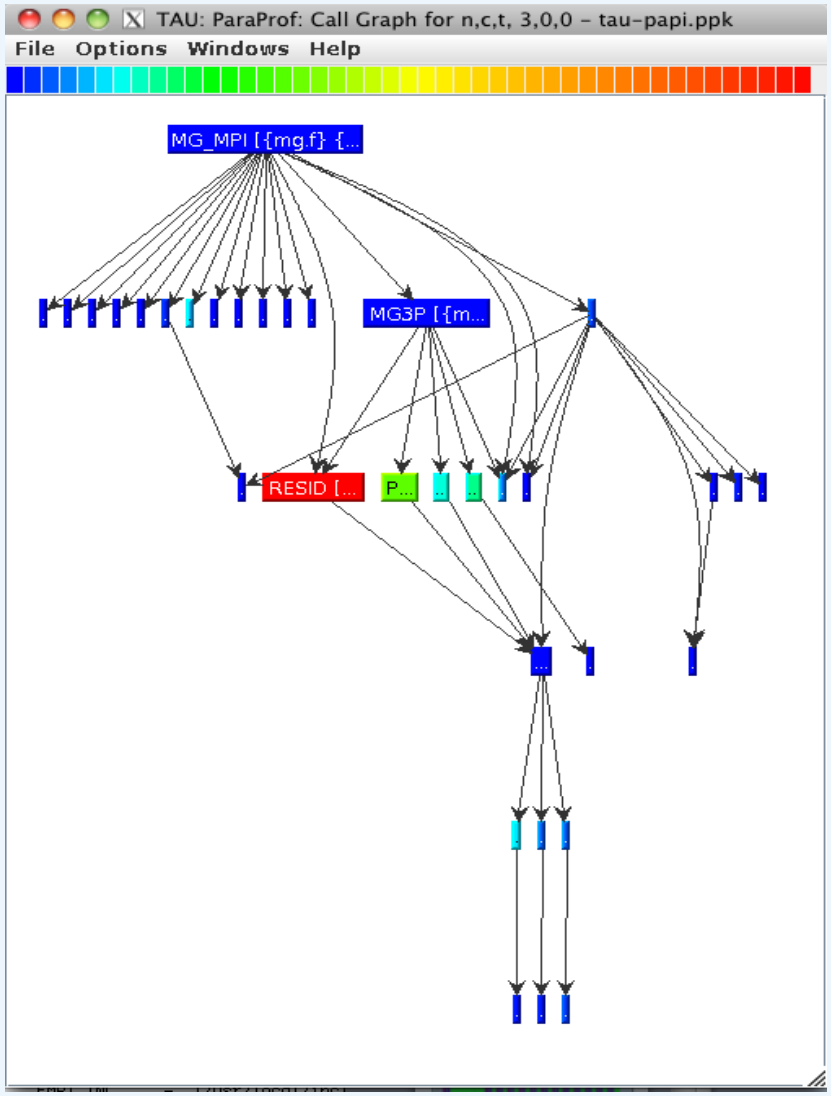
Overhead and Throttle

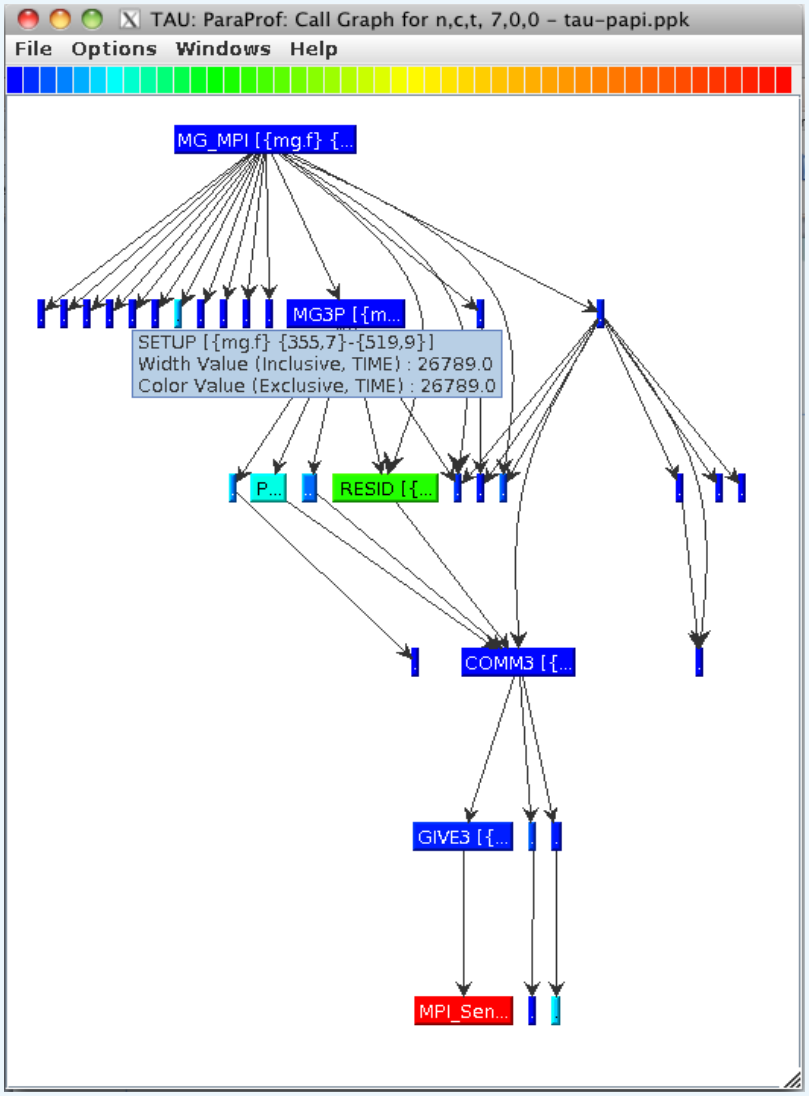
- Runtime environment variables
  - setenv TAU_THROTTLE 1
  - setenv TAU_THROTTLE_NUMCALLS 2000000
  - setenv TAU_THROTTLE_PERCALL 5
- Paraprof -> File _create selective instrumentation file
- Then can rebuild application
  - Throttle routines entered in select.tau
  - setenv TAU_OPTIONS '-optTauSelectFile="select.tau"'

Call Graph

- Runtime parameters
  - setenv TAU_CALLPATH 1
  - setenv TAU_CALLPATH_DEPTH 100
- Use Paraprof to display
  - Windows -> Thread -> Call Graph
  - Choose which thread

Call Graph


TAU: ParaProf: Call Graph for n,c,t, 3,0,0 - tau-papi.ppk
File Options Windows Help
MG_MPI [{mg.f} {...
MG3P [{m...
RESID [...
P...


TAU: ParaProf: Call Graph for n,c,t, 7,0,0 - tau-papi.ppk
File Options Windows Help
MG_MPI [{mg.f} {...
MG3P [{m...
SETUP [{mg.f} {355,7}-{519,9}]
Width Value (Inclusive, TIME) : 26789.0
Color Value (Exclusive, TIME) : 26789.0
P...
RESID [{...
COMM3 [{...
GIVE3 [{...
MPI_Sen...

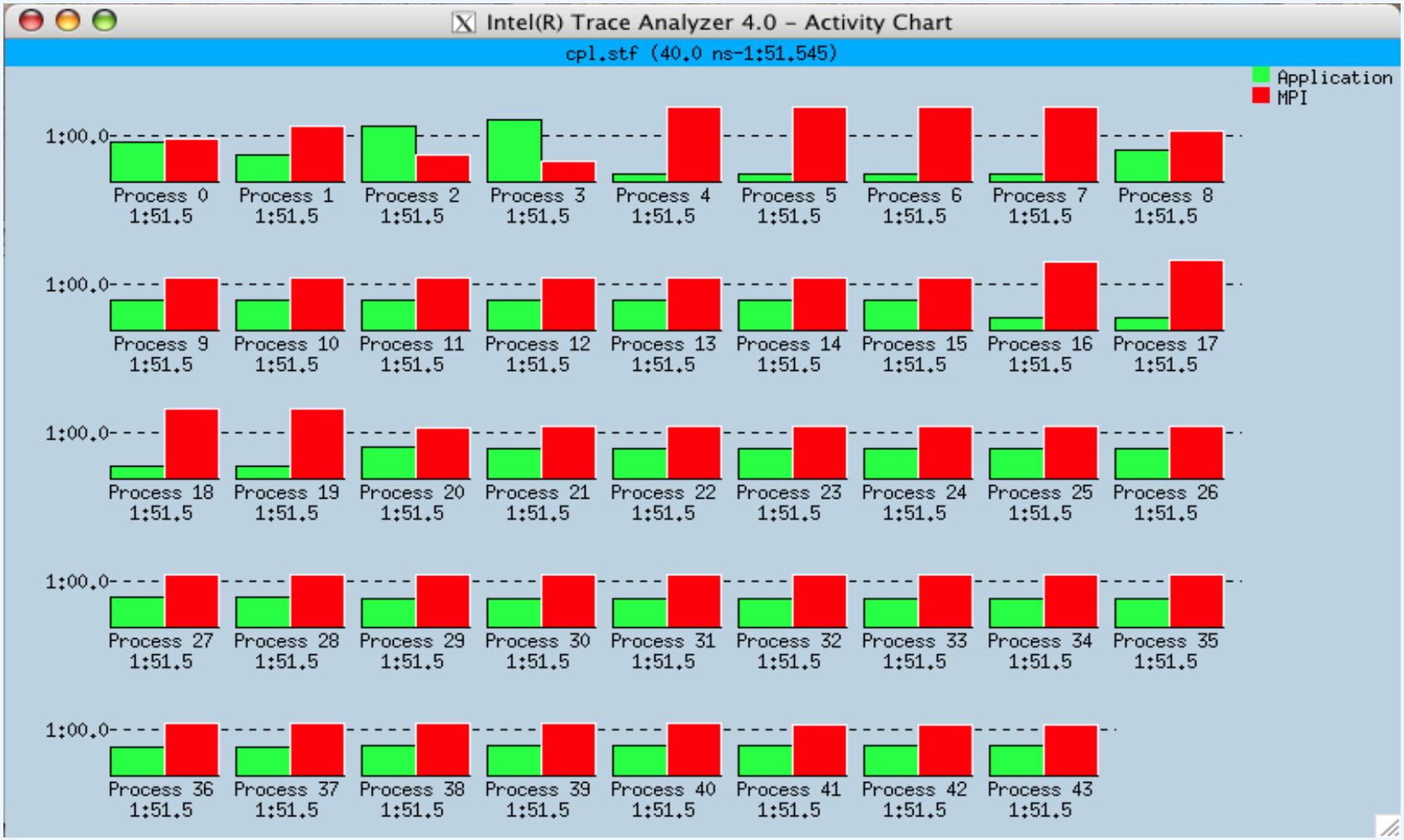
Advanced Topics

- Message passing trace
- Can generate OpenMP statistics
- Compiler instrumentation
- Loop instrumentation by way of command in select.tau (recompile)
  - BEGIN_INSTRUMENT_SECTION
  - loops routine="<routine list>" file="<file list>"
  - END_INSTRUMENT_SECTION
- Phased Profiling
- Memory Profiling
  - memory file="<file list>" routine="<routine list>"
- Much more

Vampir Trace Example

- Show dynamic picture of message passing
- Patterns of application behavior

Vampir Trace Example



Vampir Trace Example

TAU Basic Installation

- Obtain the latest files
  - PAPI (if not already loaded on your machine or even if), PDT, TAU
  - http://www.cs.uoregon.edu/Research/tau/home.php
  - (PAPI) http://icl.cs.utk.edu/papi/software/
- Can be built and installed in user directories or common space for all

A Recent Tau Installation

- Obtain the latest files
  - Used papi-4.1.2.1, pdt-3.16, tau-2.20.1
- Using home file setup
  - Build directories ~/src/tau/
    - Where I put the tar.gz files and where I build
  - Installation directories ~/tau/pdt, ~/tau/tau, ~/tau/papi
    - Where the tools go

Building PAPI

- cd ~/src/tau
- gunzip -c papi-4.1.2.1.tar.gz | tar xvf –
- cd papi-4.1.2.1/src
- ./configure --prefix=/home/George.Carr/tau/papi-4.1.2.1
- make
- make test
- make fulltest
- make install
- make install-man
- make install-tests

Building PDT

- cd ~/src/tau
- gunzip -c pdt-3.16.tar.gz | tar xvf -
- cd pdtoolkit-3.16
- ./configure -ICPC -prefix=/home/George.Carr/tau/pdt-3.16
- make
- make install

Building TAU for MPI

- cd ~/src/tau
- gunzip -c tau_2.20.1.tar.gz | tar xvf -
- cd tau-2.20.1
- ./configure -pdt=/home/George.Carr/tau/pdt-3.16 -papi=/home/George.Carr/tau/papi-4.1.2.1 -MULTIPLECOUNTERS -c++=icpc -cc=icc -fortran=intel -prefix=/home/George.Carr/tau/tau-2.20.1 -mpi
- set path=($path ~/tau/tau-2.20.1) [add to environment]
- make install

- ./configure –help [to get lot's of stuff you don't need to begin with]

Building TAU for MPI and OpenMP

- From previous work area:
  - ./configure -pdt=/home/George.Carr/tau/pdt-3.16 -papi=/home/George.Carr/tau/papi-4.1.2.1 -MULTIPLECOUNTERS -c++=icpc -cc=icc -fortran=intel -prefix=/home/George.Carr/tau/tau-2.20.1 -mpi -openmp -opari
  - set path=($path ~/tau/tau-2.20.1) [add to environment]
  - make install

Some Additional Notes

- Should build for each compiler (Intel, PGI, other)
- Adding additional option configurations as easy as just adding OpenMP
- papi_avail will tell you which papi metrics are available on your machine.
- The paths generally need to be absolute paths
  - Ex ~/tau does not work but /home/George.Carr/tau does
- Add papi bin installation directory to search path
- Tau knows the paths to PAPI and PDT because the configure scripts built them in
- Take a look at tau_f90.sh for a flavor of what is going on

What's it look like?

- /home/George.Carr/tau/ >> ls
- papi-4.1.2.1/ pdt-3.16/ tau-2.20.1/

- /home/user/tau/tau-2.20.1>> ls
- etc/ examples/ include/ man/ tools/ x86_64/

- Look at /home/George.Carr/tau/tau-2.20.1/x86_64/bin/tau_f90.sh
- Look at /home/George.Carr/tau/tau-2.20.1/x86_64/lib/Makefile*

References

- http://www.cs.uoregon.edu/Research/tau/home.php
  - Documentation
  - Presentations
  - Video training
- /home/George.Carr/tau/tau-2.20.1/examples/ >>
  - Many example of codes and ways to instrument them
- George Carr (George.Carr@noaa.gov or georgercarrjr@acm.org)

Thanks
Any questions?