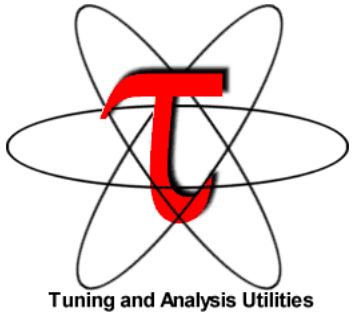
TAU Performance System®
Hands on session



Sameer Shende
sameer@cs.uoregon.edu
University of Oregon
http://tau.uoregon.edu

Copy the workshop tarball

- Setup preferred program environment compilers
  - Default set Intel Compilers with Intel MPI

```
% source /home/hpc/a2c06/lu23vox/load_tau.sh
% tar xf ~lu23vox/workshop.tgz; cd workshop; cat README; cd NPB3.1;
# If you have previous performance data from Score-P, you may view it with TAU's paraprof
% paraprof profile.cubex &
```

Copy the workshop tarball

- Setup preferred program environment compilers
  - Default set Intel Compilers with Intel MPI

```
% source /home/hpc/a2c06/lu23vox/load_tau.sh
% tar xf ~lu23vox/workshop.tgz; cd workshop; cat README; cd NPB3.1;
# If you have previous performance data from Score-P, you may view it with TAU's paraprof
% paraprof profile.cubex &
```

NPB-MPI Suite

- The NAS Parallel Benchmark suite (MPI+OpenMP version)
  - Available from:

    http://www.nas.nasa.gov/Software/NPB
  - 3 benchmarks in Fortran77
  - Configurable for various sizes & classes
- Move into the NPB3.1 root directory

```
% ls
bin/    common/  jobscript/  Makefile  README.install   SP/
BT/  config/  LU/      README    README.tutorial  sys/
```

- Subdirectories contain source code for each benchmark
  - plus additional configuration and common code
- The provided distribution has already been configured for the tutorial, such that it's ready to “make” one or more of the benchmarks and install them into a (tool-specific) “bin” subdirectory

NPB-MPI / BT: config/make.def

```
#                SITE- AND/OR PLATFORM-SPECIFIC DEFINITIONS.
#
#---------------------------------------------------------------------

#---------------------------------------------------------------------
# Configured for generic MPI with GCC compiler
#---------------------------------------------------------------------
#OPENMP  = -fopenmp      # GCC compiler
OPENMP =           # Intel compiler

...
#---------------------------------------------------------------------
# The Fortran compiler used for MPI programs
#---------------------------------------------------------------------
# MPIF77 = mpif77 # OpenMPI with Intel compiler
MPIF77 = mpif77
# Alternative variant to perform instrumentation
# MPIF77 = tau_f90.sh
# PREP is a generic preposition macro for instrumentation preparation
#MPIF77 = $(PREP) mpif77 -f77=ifort
#MPIF77 = scorep …
...
```

Default (no instrumentation)

NPB-MPI Benchmark

```
% make
    ========================================
    =      NAS Parallel Benchmarks 3.1       =
    =      MPI/F77/C                         =
    ========================================
  To make a NAS benchmark type

         make <benchmark-name> CLASS=<class> NPROCS=<nprocs>

  where <benchmark-name> is "bt", "lu", or "sp"
        <class>          is "S", "W", "A" through "F"
        <nprocs>         is number of processes

   [...]

 ***********************************************************
 * Custom build configuration is specified in config/make.def  *
 * Suggested tutorial exercise configuration for HPC systems:  *
 *       make bt CLASS=B NPROCS=64                             *
 ***********************************************************
```

- Type "make" for instructions

Building an NPB-MPI Benchmark

```
% make -j
make[1]: Entering directory `BT'
make[2]: Entering directory `sys'
cc  -o setparams setparams.c -lm
make[2]: Leaving directory `sys'
../sys/setparams bt 64 B
mpiifort -c  -O3   -g bt.f
mpiifort -c  -O3   -g make_set.f
mpiifort -c  -O3   -g initialize.f
mpiifort -c  -O3   -g exact_solution.f
mpiifort -c  -O3   -g exact_rhs.f
…
mpiifort  -o ../bin/bt.B.64 bt.o make_set.o initialize.o exact_solution.o
 exact_rhs.o set_constants.o adi.o define.o copy_faces.o rhs.o x_solve.o
 y_solve.o z_solve.o add.o error.o verify.o setup_mpi.o
../common/print_results.o ../common/timers.o ../common/randi8.o btio.o
make[2]: Leaving directory `BT'
Built executable ../bin/bt_32.8
make[1]: Leaving directory `BT'
```

- Specify the benchmark configuration
  - benchmark name: **bt**, lu, sp
  - the number of MPI processes: NPROCS=**64**
  - the benchmark class (S, W, A, B, C, D, E): CLASS=**B**

```
% make suite
```

Copy the workshop tarball

- Setup preferred program environment compilers
  - Default set Intel Compilers with Intel MPI

```
% source /home/hpc/a2c06/lu23vox/load_tau.sh
% tar xf ~lu23vox/workshop.tgz; cd workshop; cat README; cd NPB3.1;
# If you have previous performance data from Score-P, you may view it with TAU's paraprof
% paraprof profile.cubex &
```

Using TAU with MAQAO: tau_rewrite

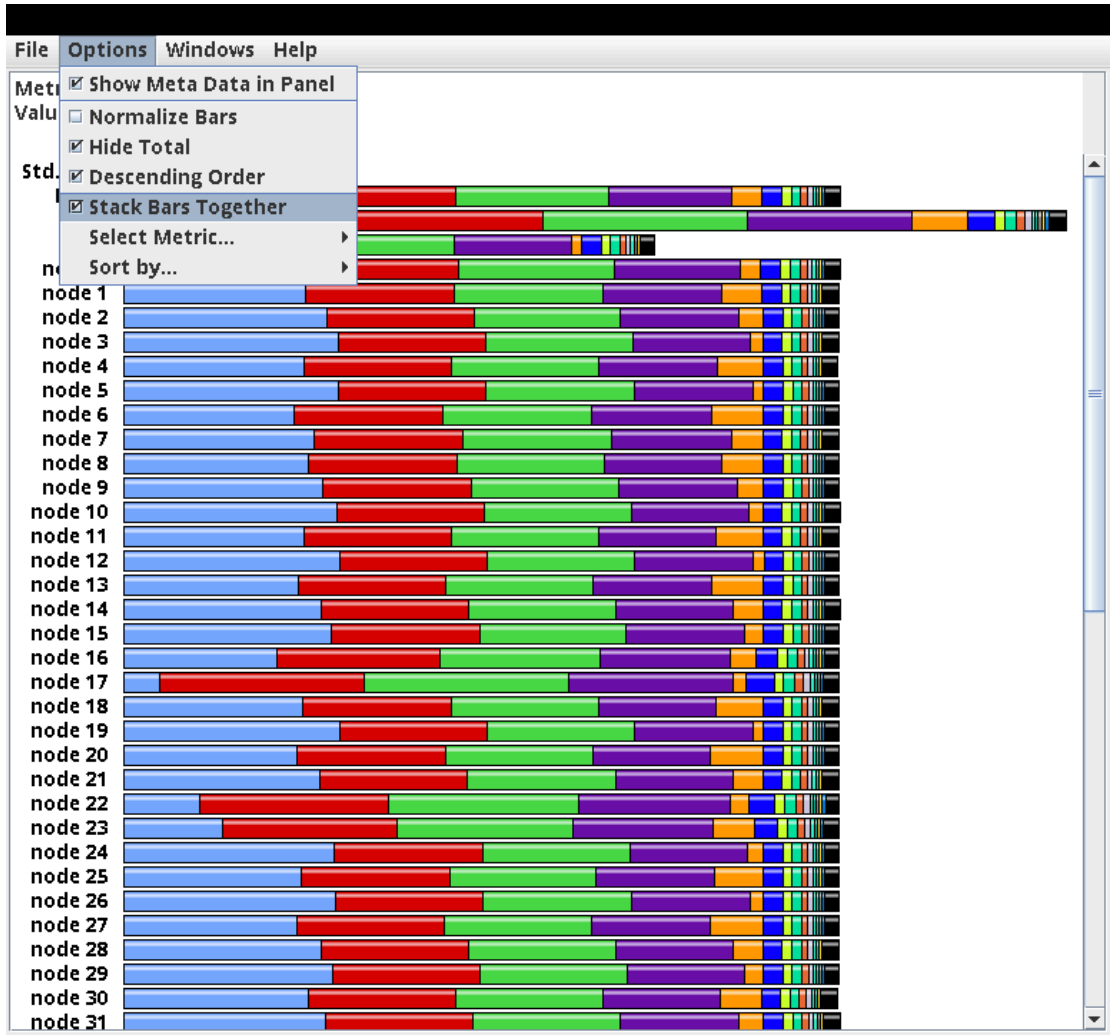
- Setup preferred program environment compilers
  - Default set Intel Compilers with Intel MPI

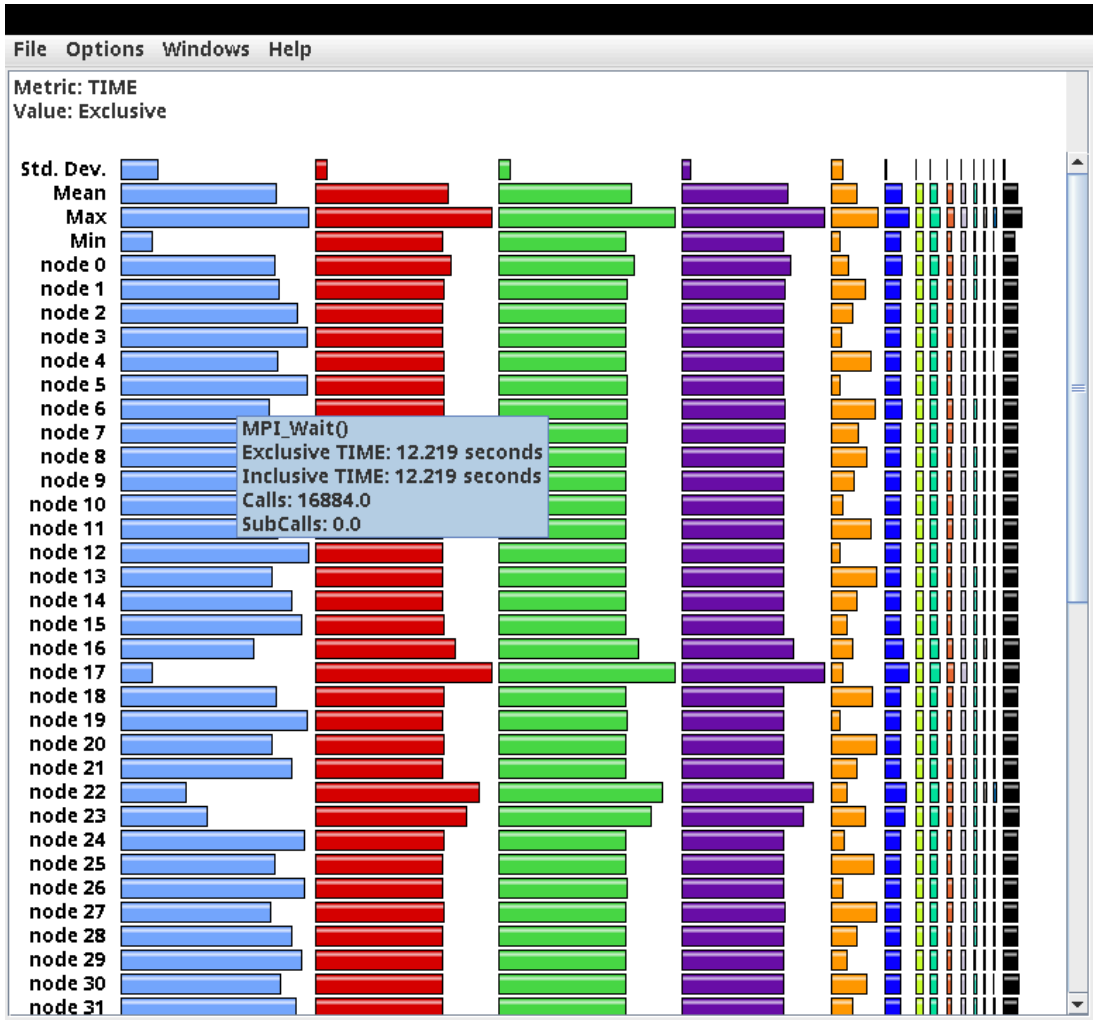
```
% source /home/hpc/a2c06/lu23vox/load_tau.sh
% tar xf ~lu23vox/workshop.tgz; cd workshop; cat README; cd NPB3.1;
% make; cd bin
% tau_rewrite bt.B.64 bt.i


# Get an interactive node
% salloc  --nodes=1  --time=02:00:00 --constraint=cache,quad --reservation=TuningWorkshop \
    --partition=mpp3_batch
% source /home/hpc/a2c06/lu23vox/load_tau.sh
% cd workshop/NPB3.1/bin
% mpirun -np 64 ./bt.B.64  (Uninstrumented)
% mpirun -np 64 ./bt.i
% paraprof &
```

ParaProf Profile Browser

```
% paraprof
```

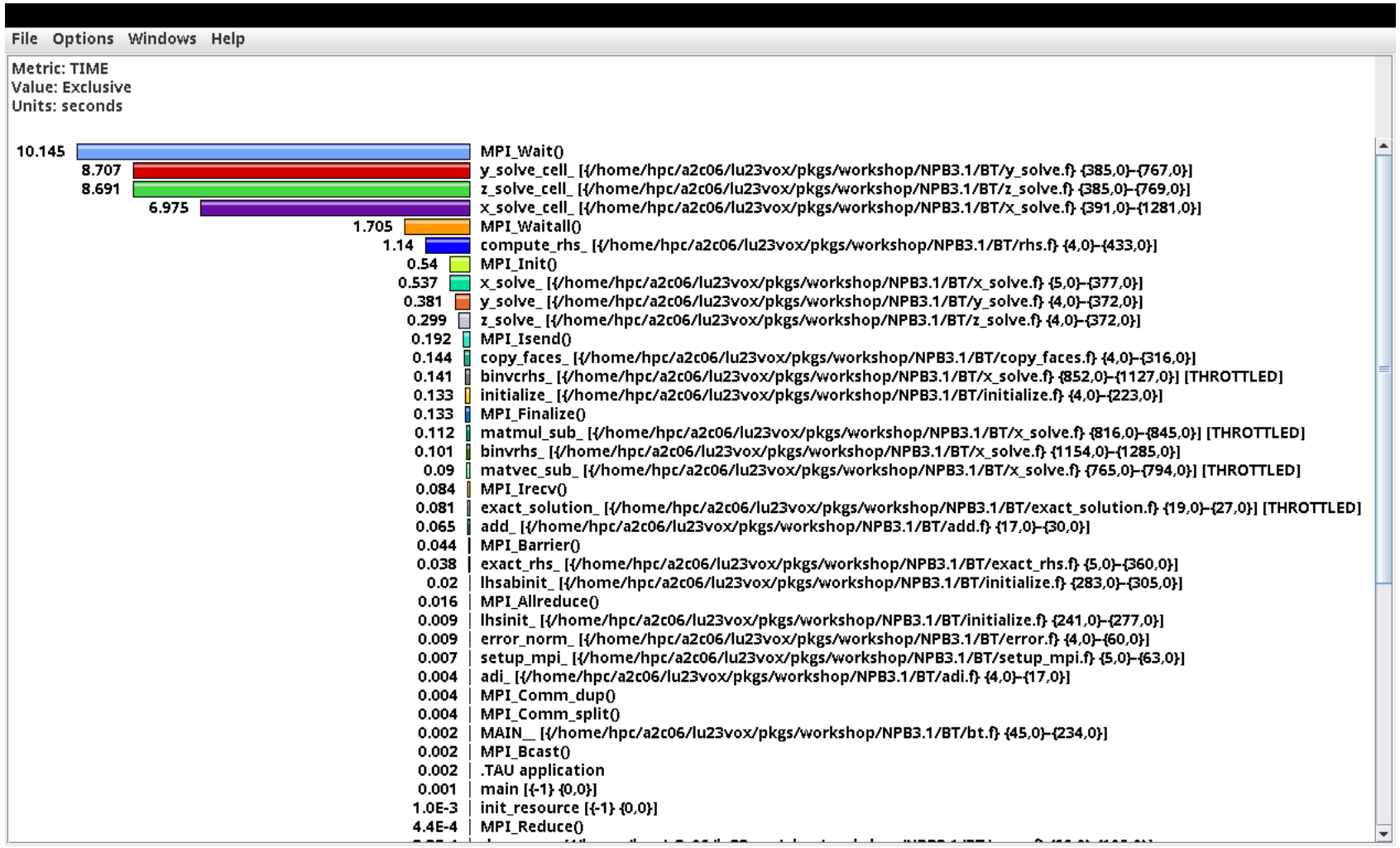
ParaProf Profile Browser





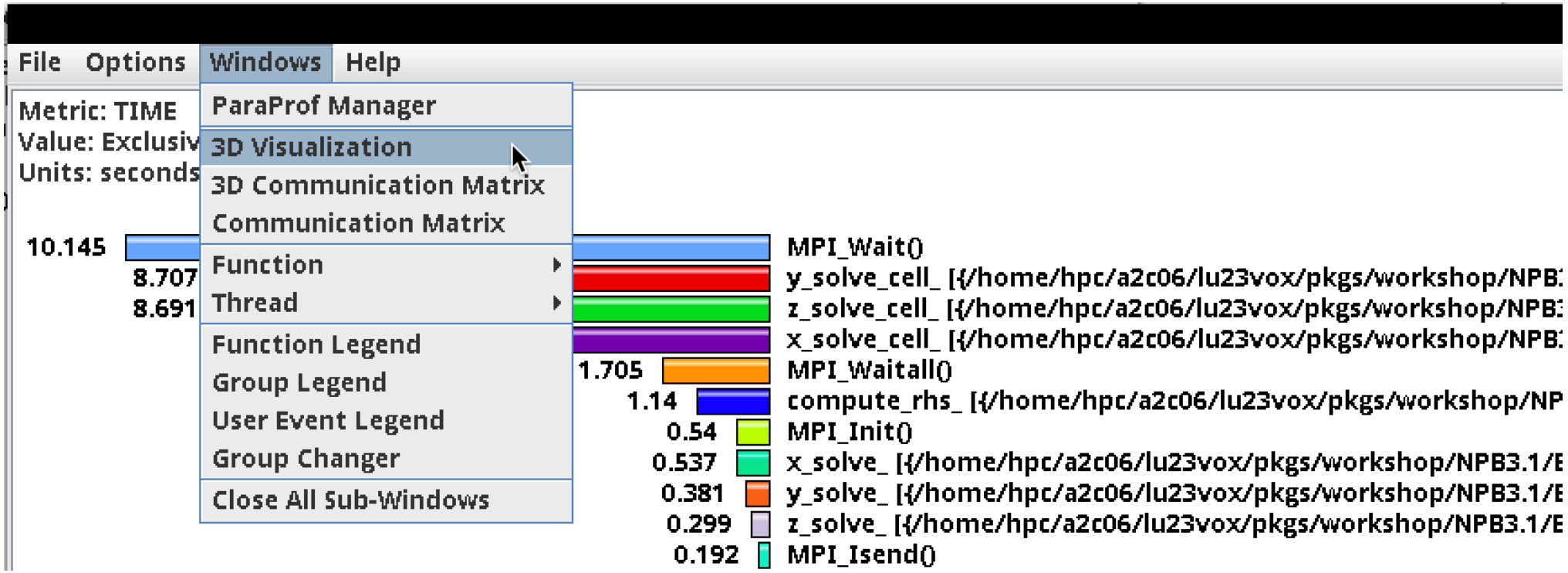
ParaProf Profile Browser

Click on node 0



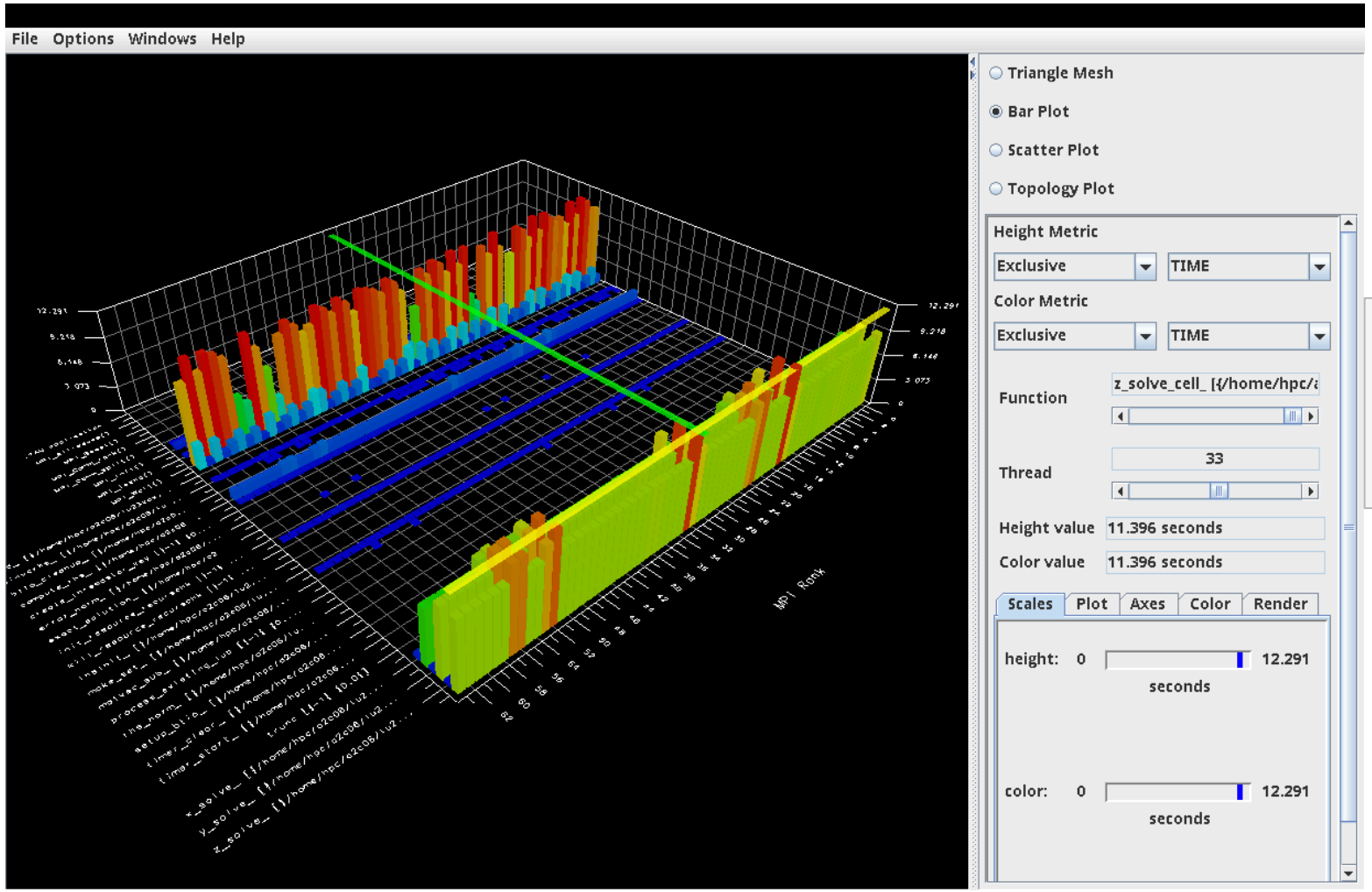
THROTTLED !

3D Visualization Window



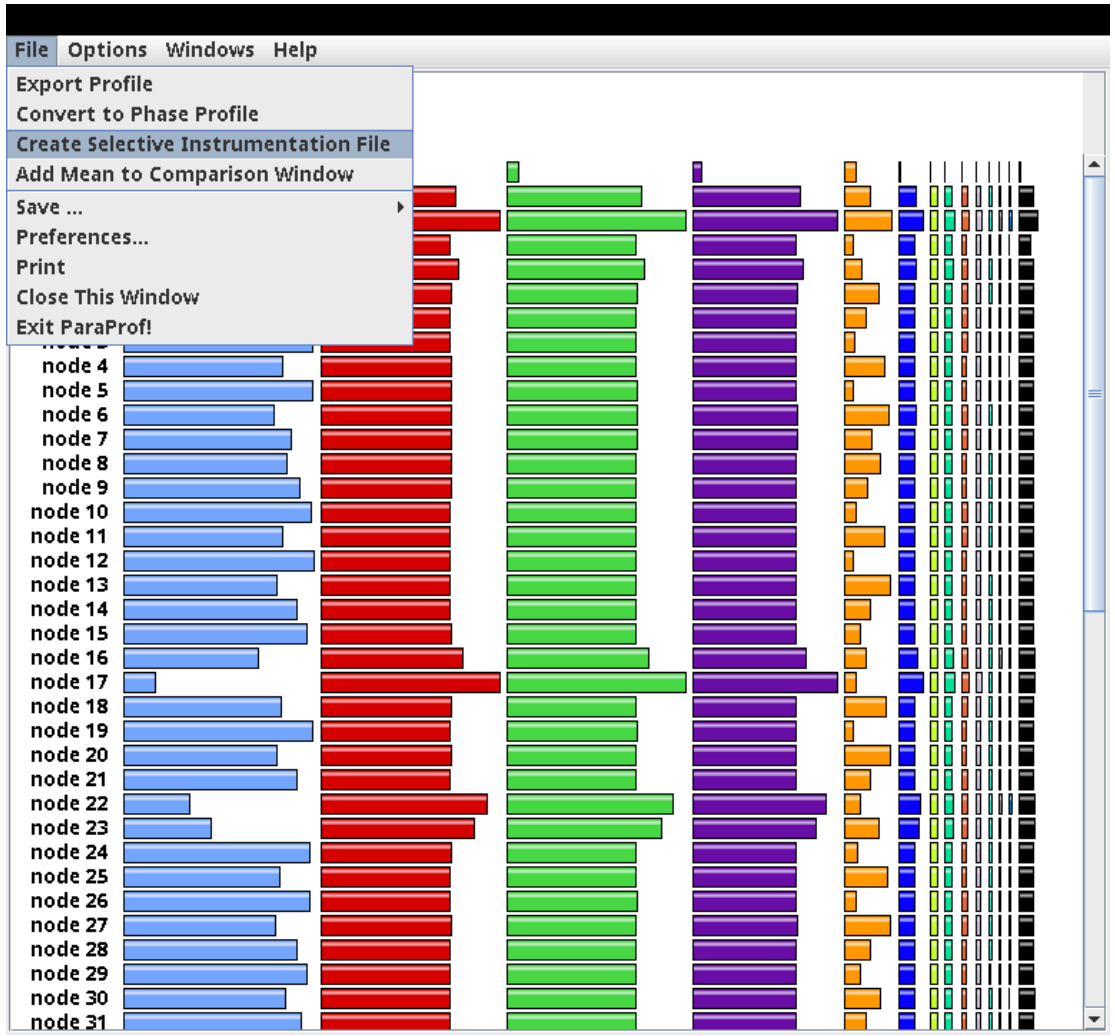
Choose Windows -> 3D Visualization

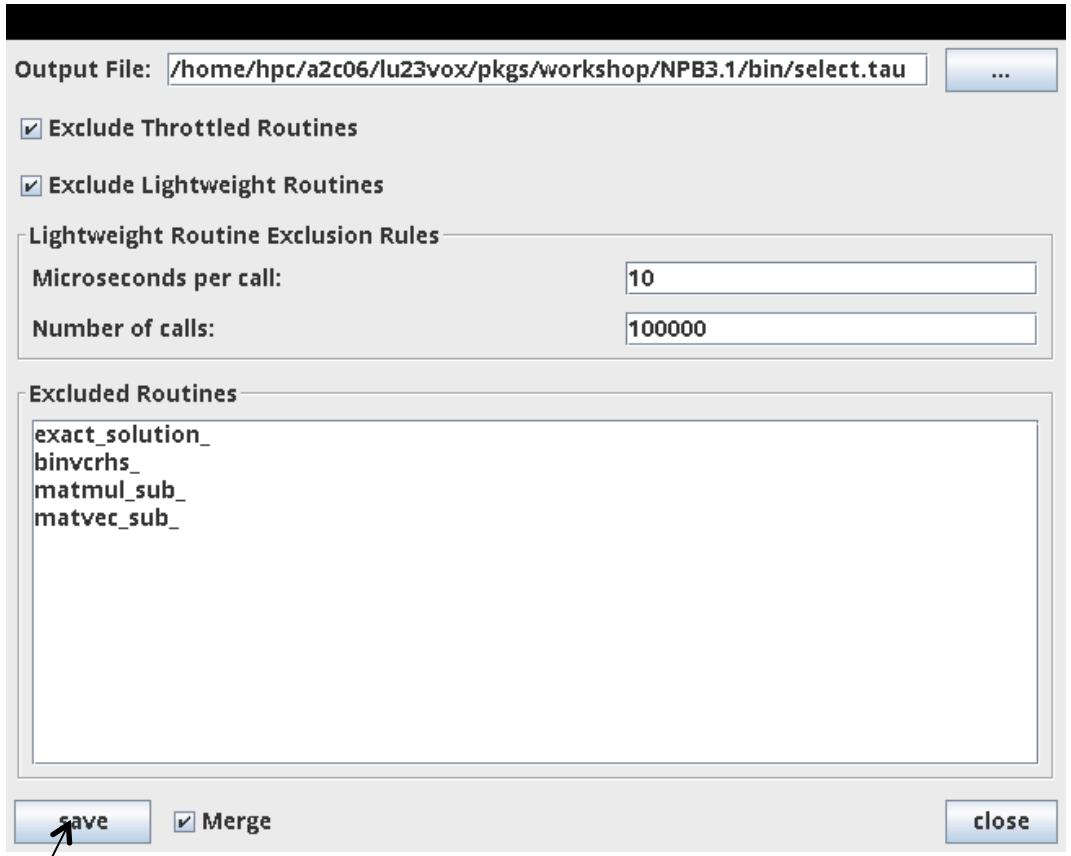
ParaProf 3D Visualization Window



1st mouse button: rotate
2nd mouse button: translate
Scroll wheel (+/-) : Zoom in/out
Choose function/thread slider

Create a filter/selective instrumentation file from the main window





Click save

Creates select.tau

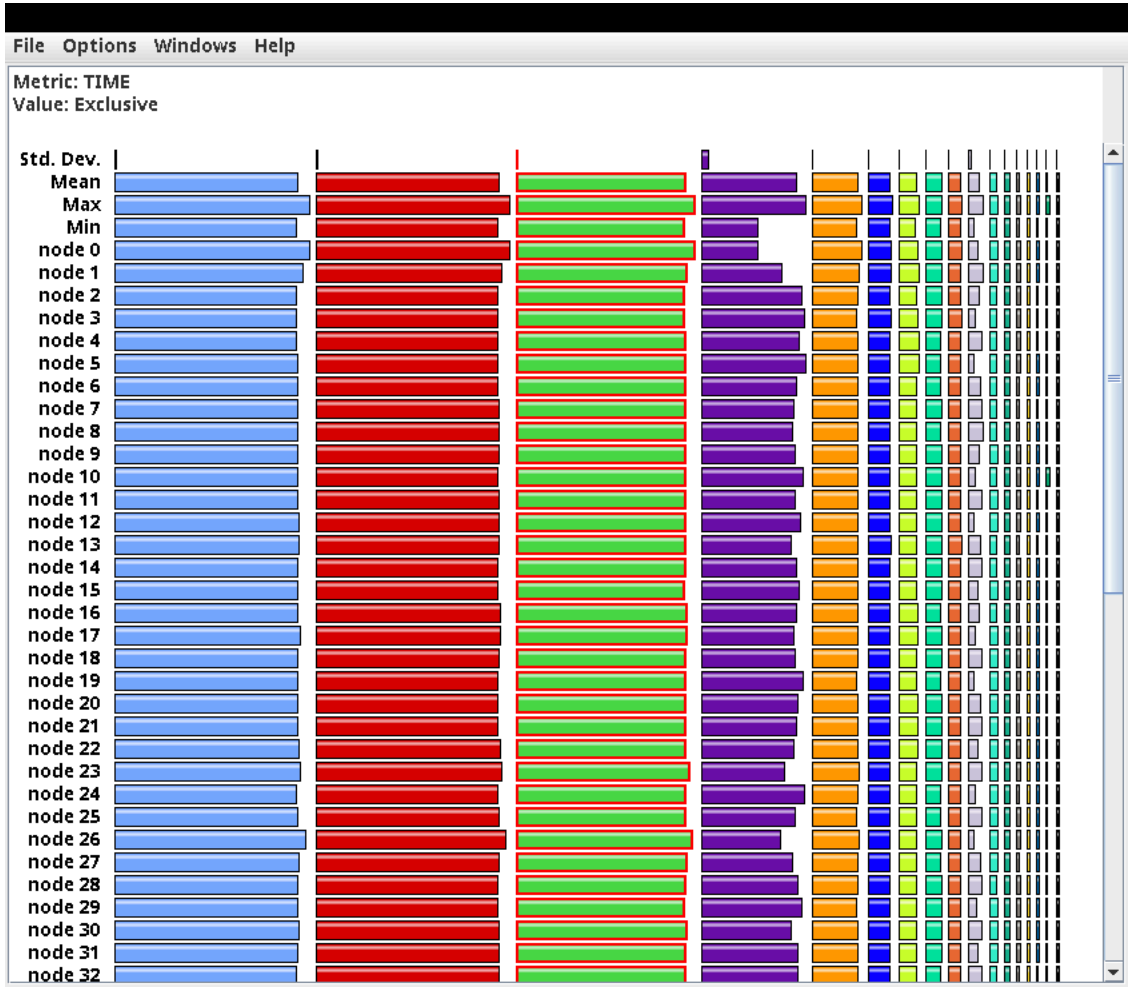
Re-instrument BT benchmark using selective instrumentation file

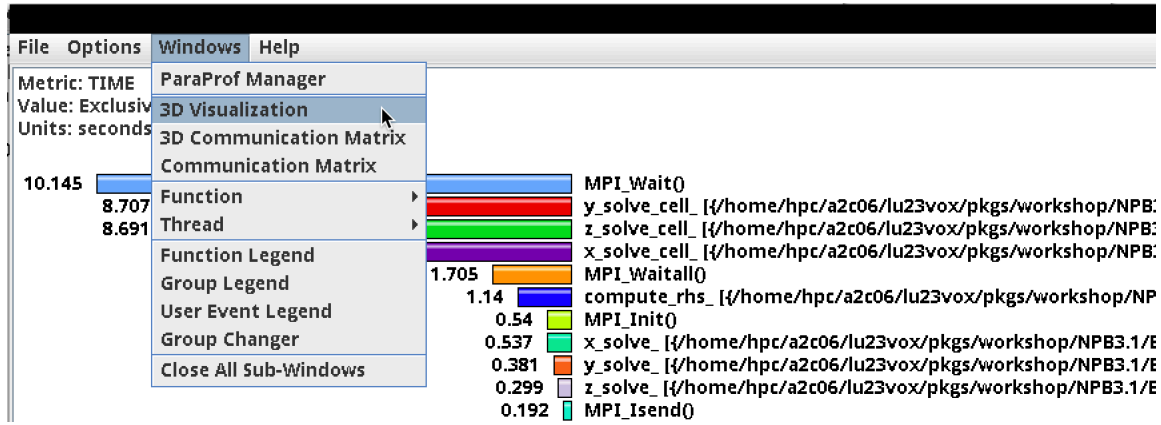
Terminal — ssh lrz — 126×22

```
lu23vox@mpp3-login8:~/pkgs/workshop/NPB3.1/bin> cat select.tau
BEGIN_EXCLUDE_LIST
exact_solution_
binvcrhs_
matmul_sub_
matvec_sub_
END_EXCLUDE_LIST

lu23vox@mpp3-login8:~/pkgs/workshop/NPB3.1/bin> tau_rewrite -f select.tau bt.B.64 bt.i
tau_rewrite: Using maqao binary from MAQAO_BINARY environment variable: /home/hpc/a2c06/lu23vof/MAQAO//bin/maqao
tau_rewrite: Binary instrumentation done through MAQAO Multi-Architecture Disassembler, Rewriter and ASsembler technology

lu23vox@mpp3-login8:~/pkgs/workshop/NPB3.1/bin> mpirun -np 64 ./bt.i
```

```
On login node:
% paraprof
```

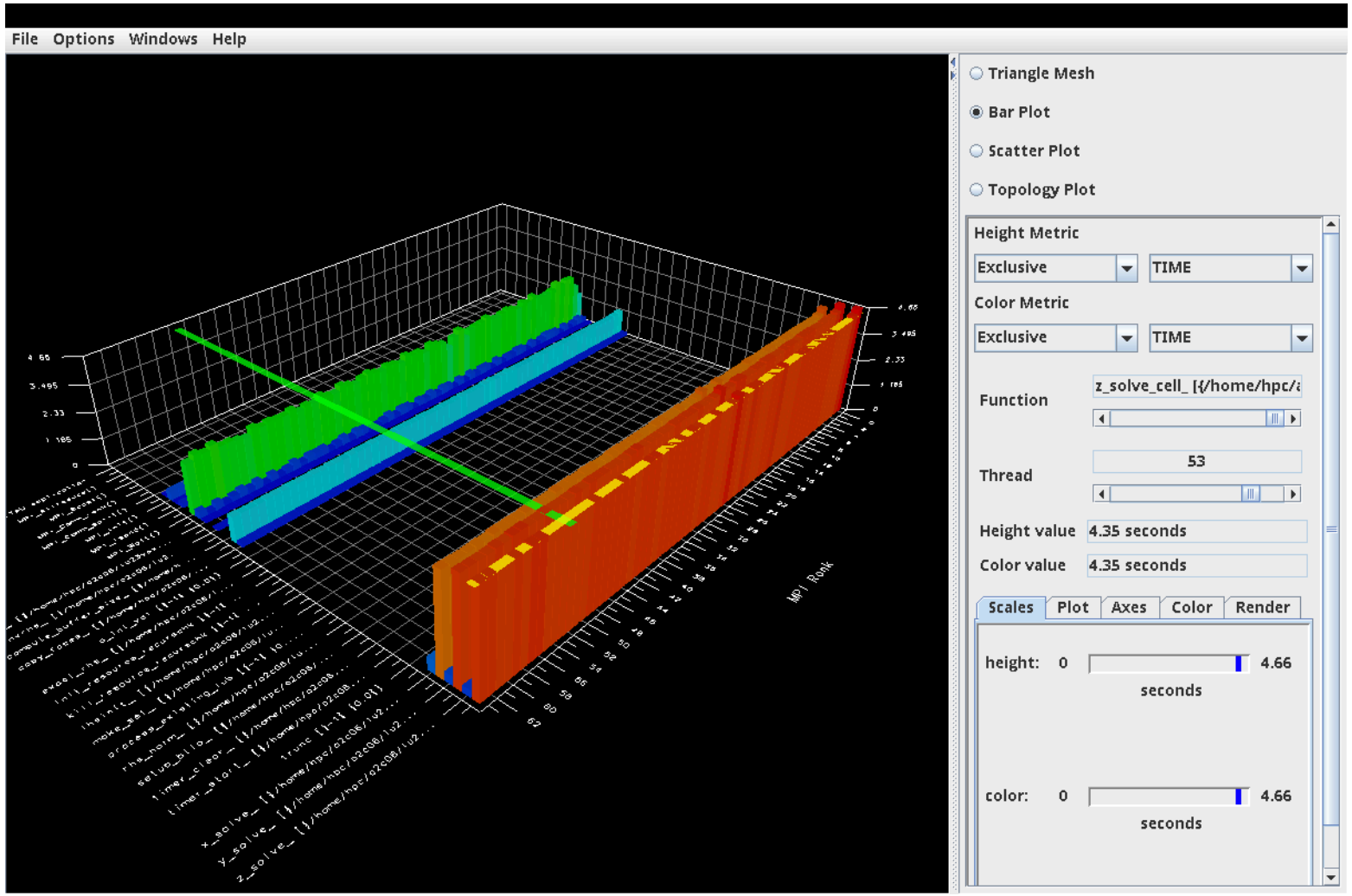
After optimizing instrumentation with TAU and MAQAO






```
Choose Windows -> 3D Visualization
```

ParaProf Profile Browser



Optimized instrumentation!

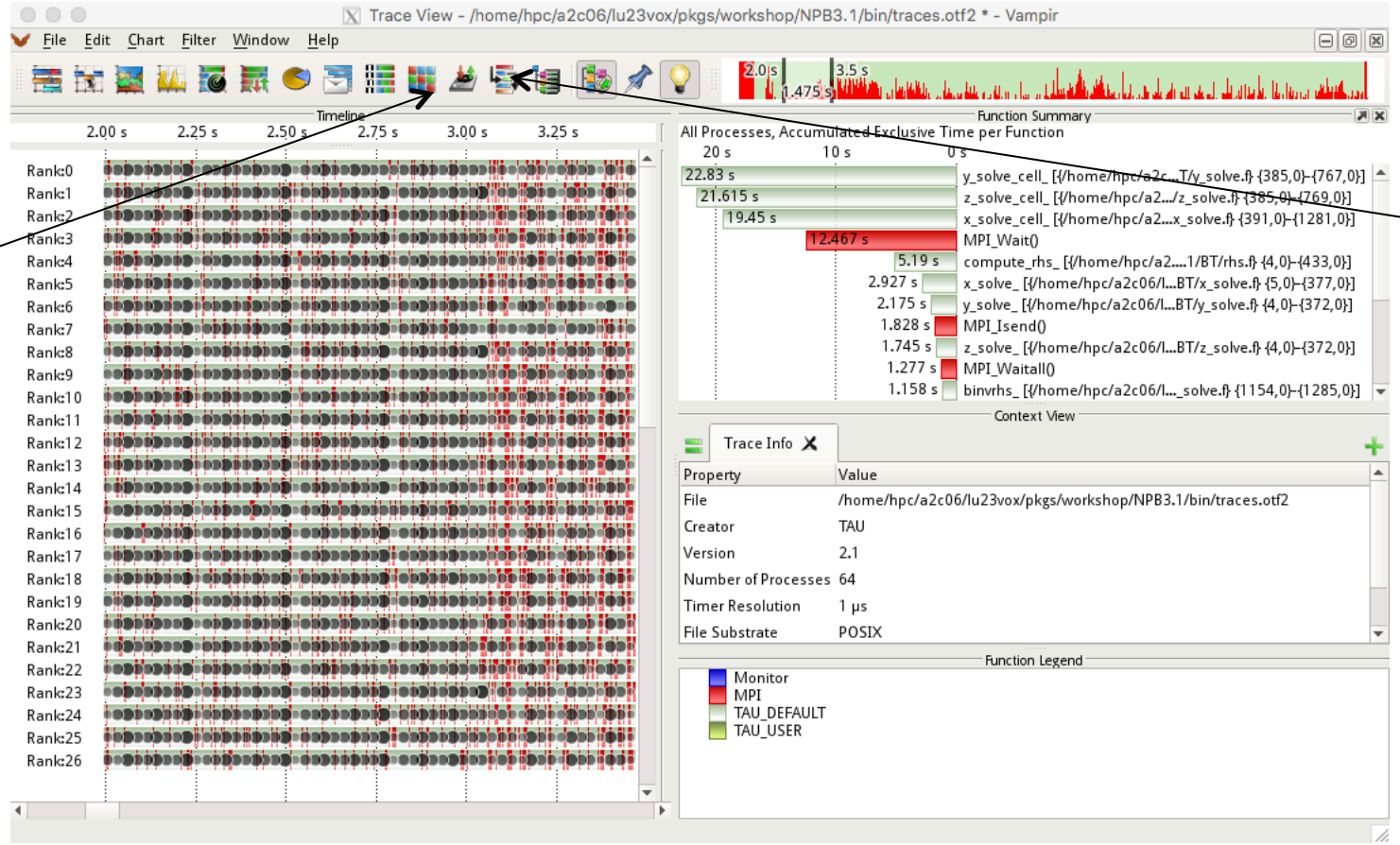
Create a Score-P tracefile

- Reinstrument the BT binary to use TAU's Score-P configuration
  - Run and then launch Vampir trace visualzer

```
% tau_rewrite -f select.tau -T scorep ./bt.B.64 bt.i

% export SCOREP_ENABLE_TRACING=1
% mpirun -np 64 ./bt.i

# On login node:
% cd scorep-<dir>; vampir traces.otf2 &
```

Vampir Trace Visualizer [TU Dresden]

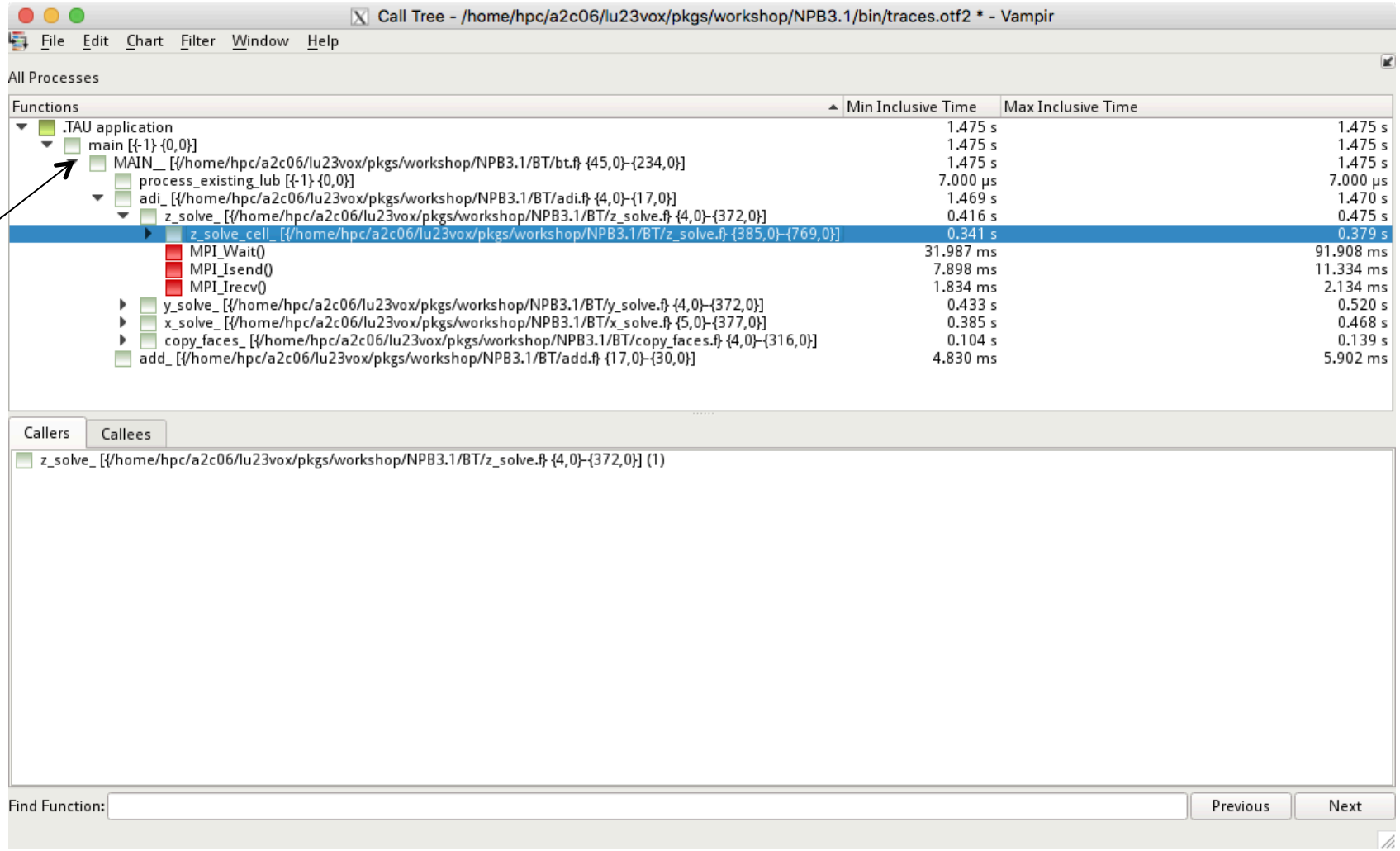
Click on Communication matrix display icon



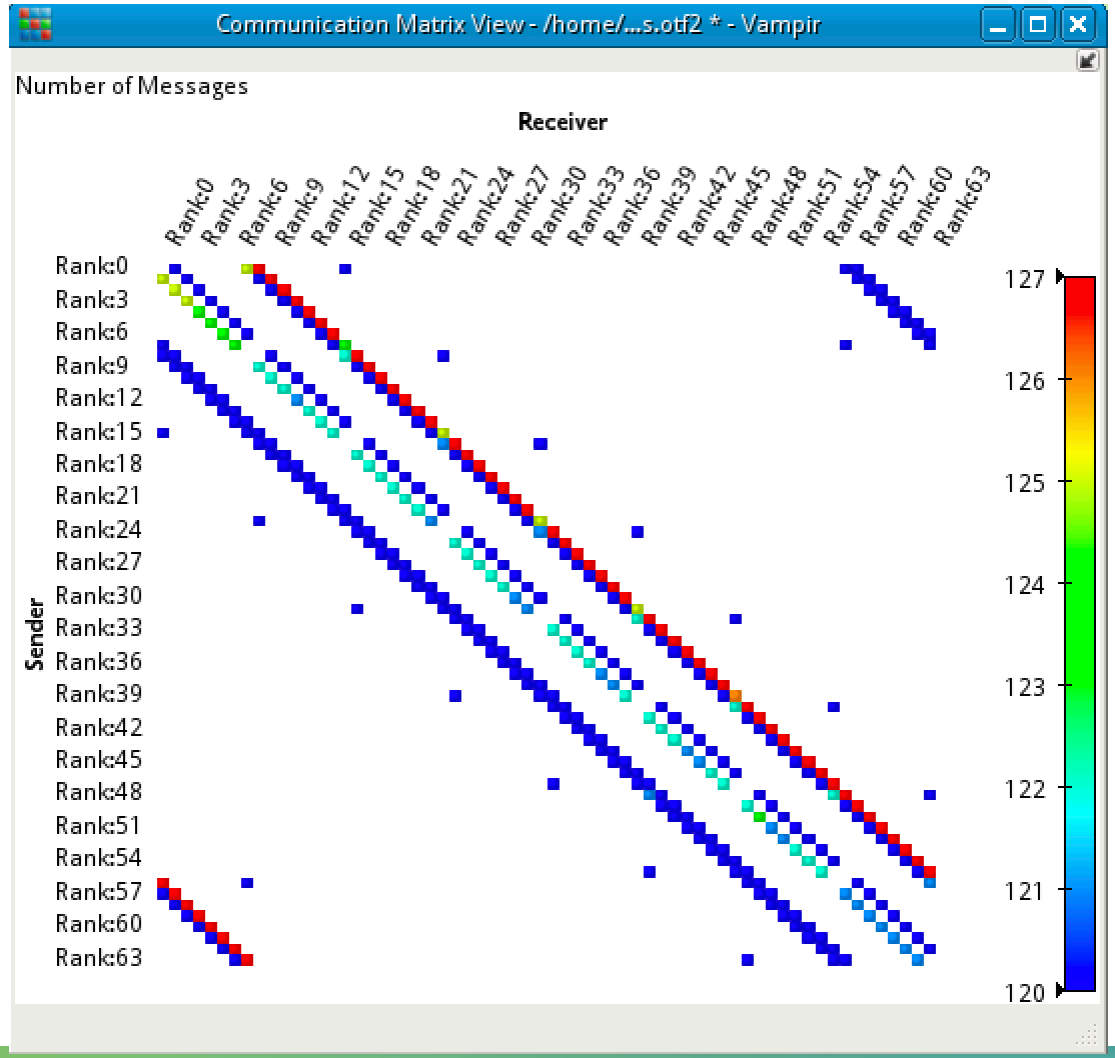
Click calltree icon

Vampir Calltree window

Expand nodes

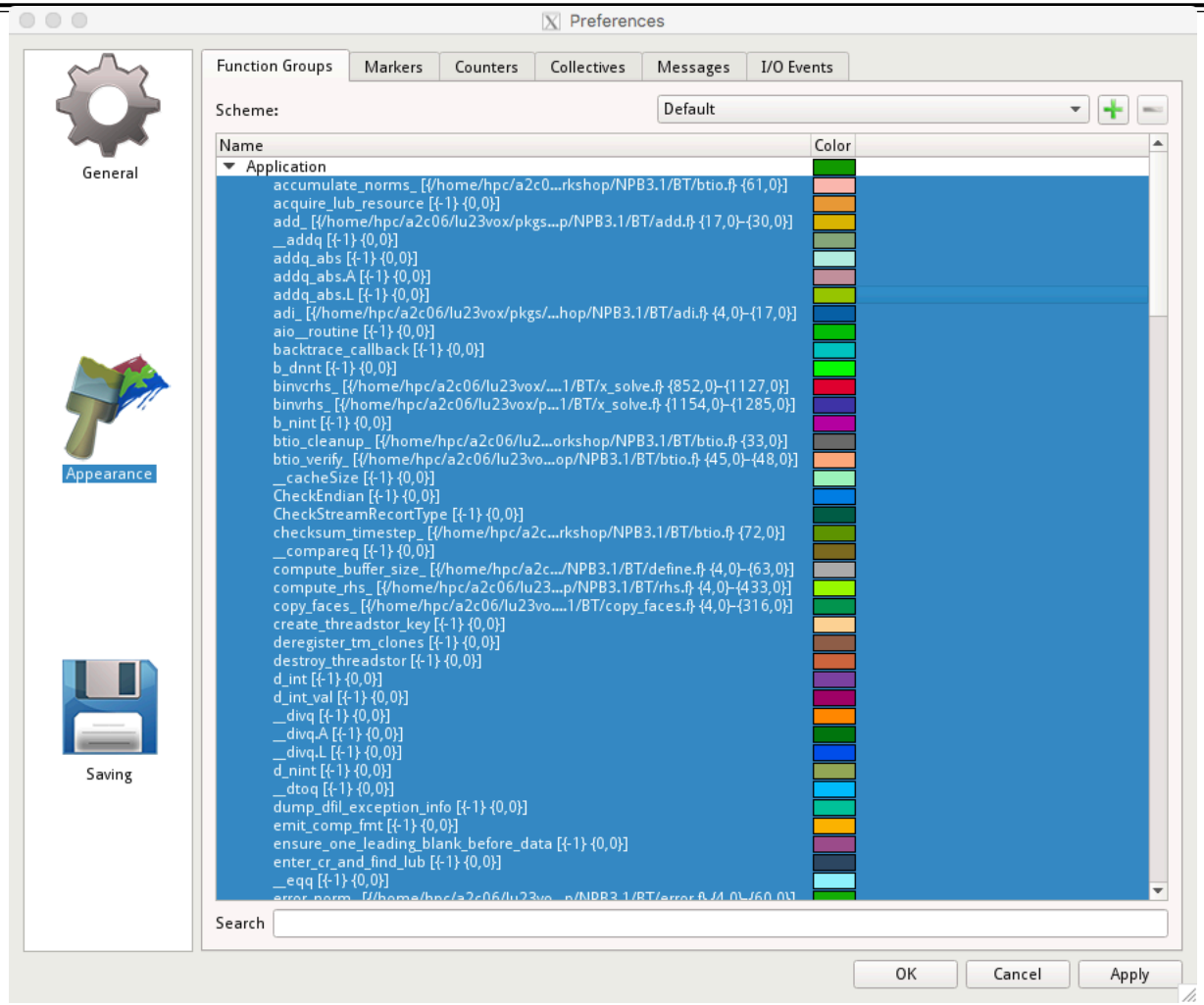


Communication Matrix Display

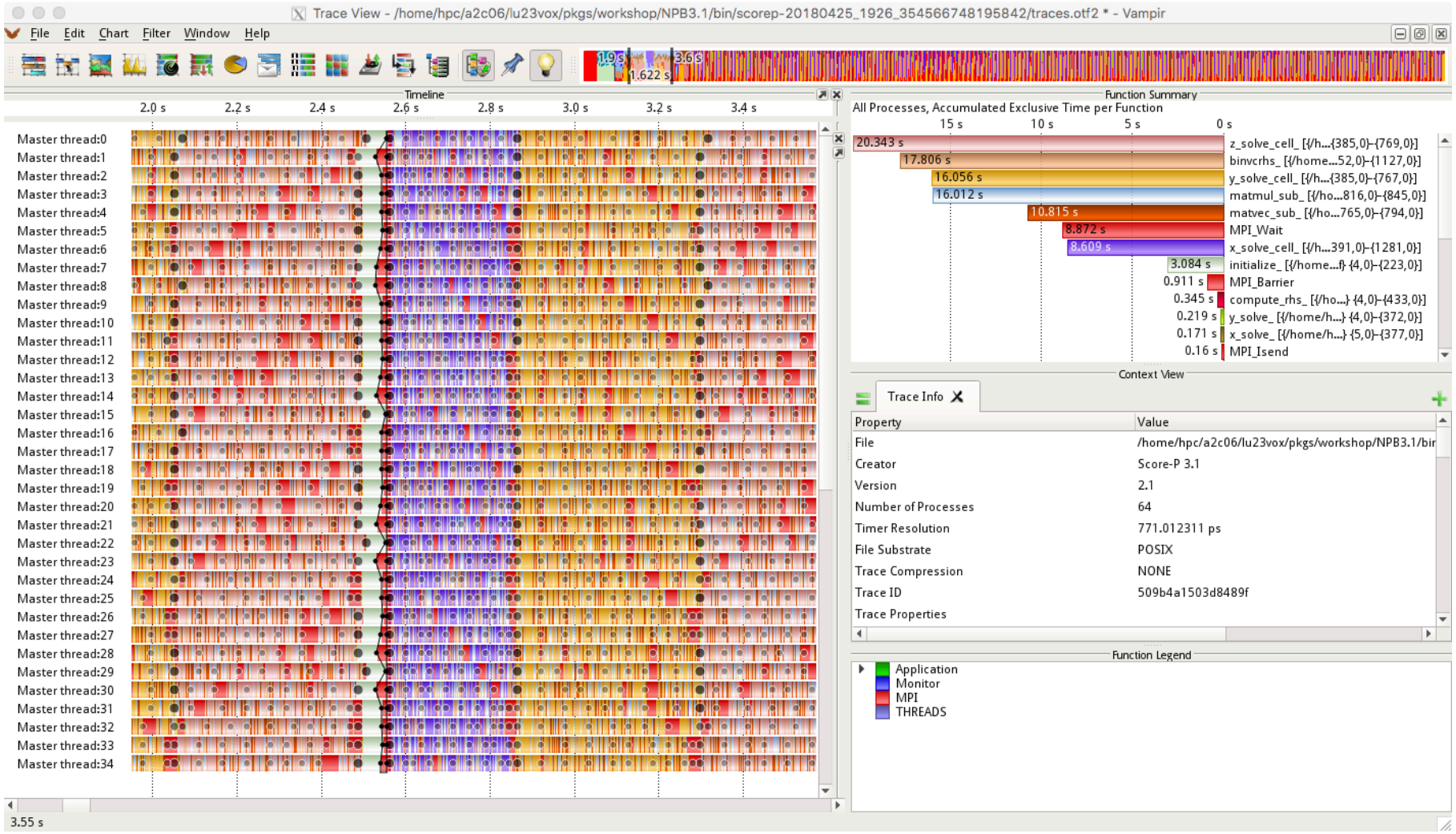


File Preferences Window: Appearances

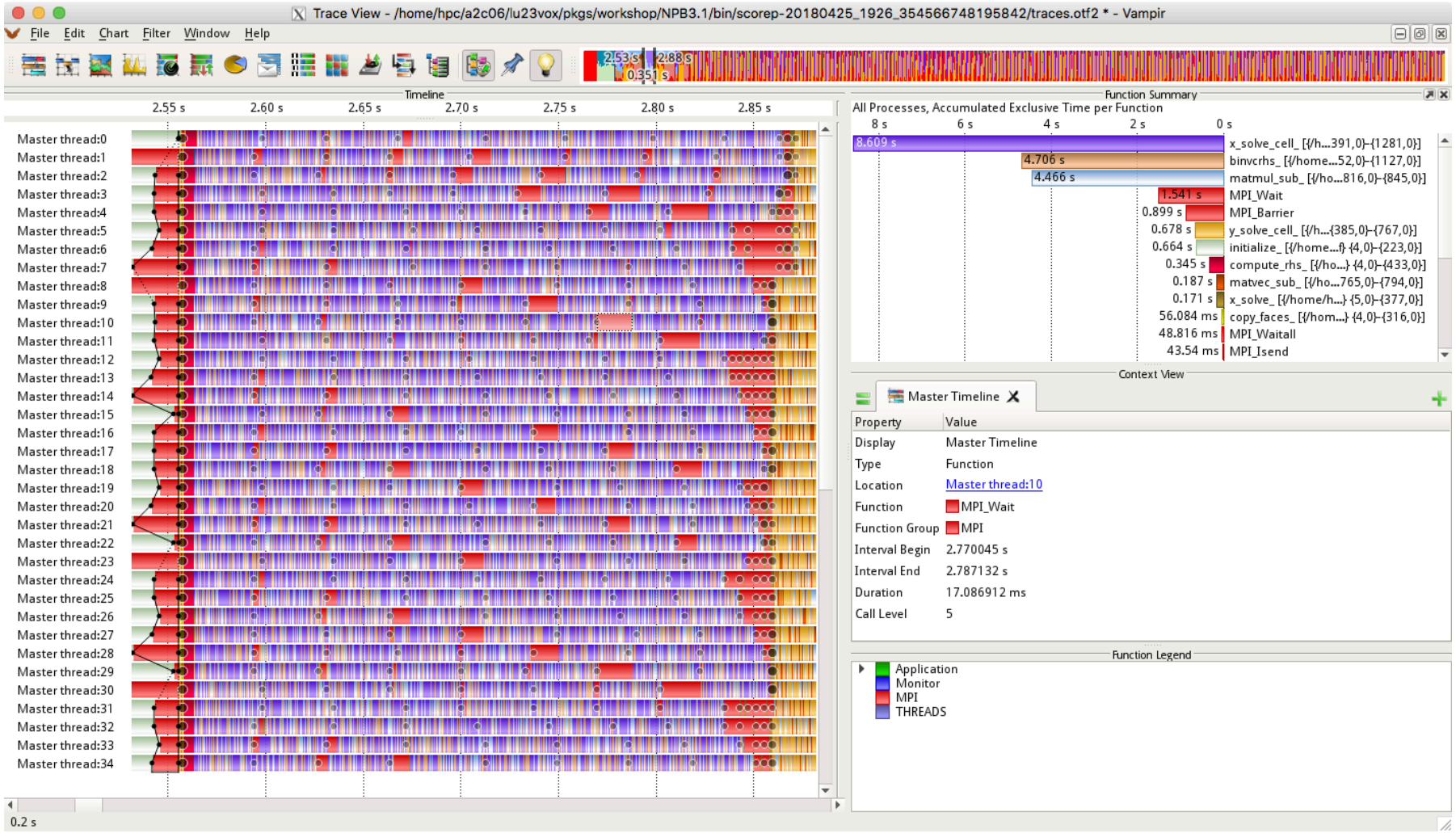
```
File -> preferences
Multi select and
Right click ->
Set random colors
```



Vampir Timeline Window

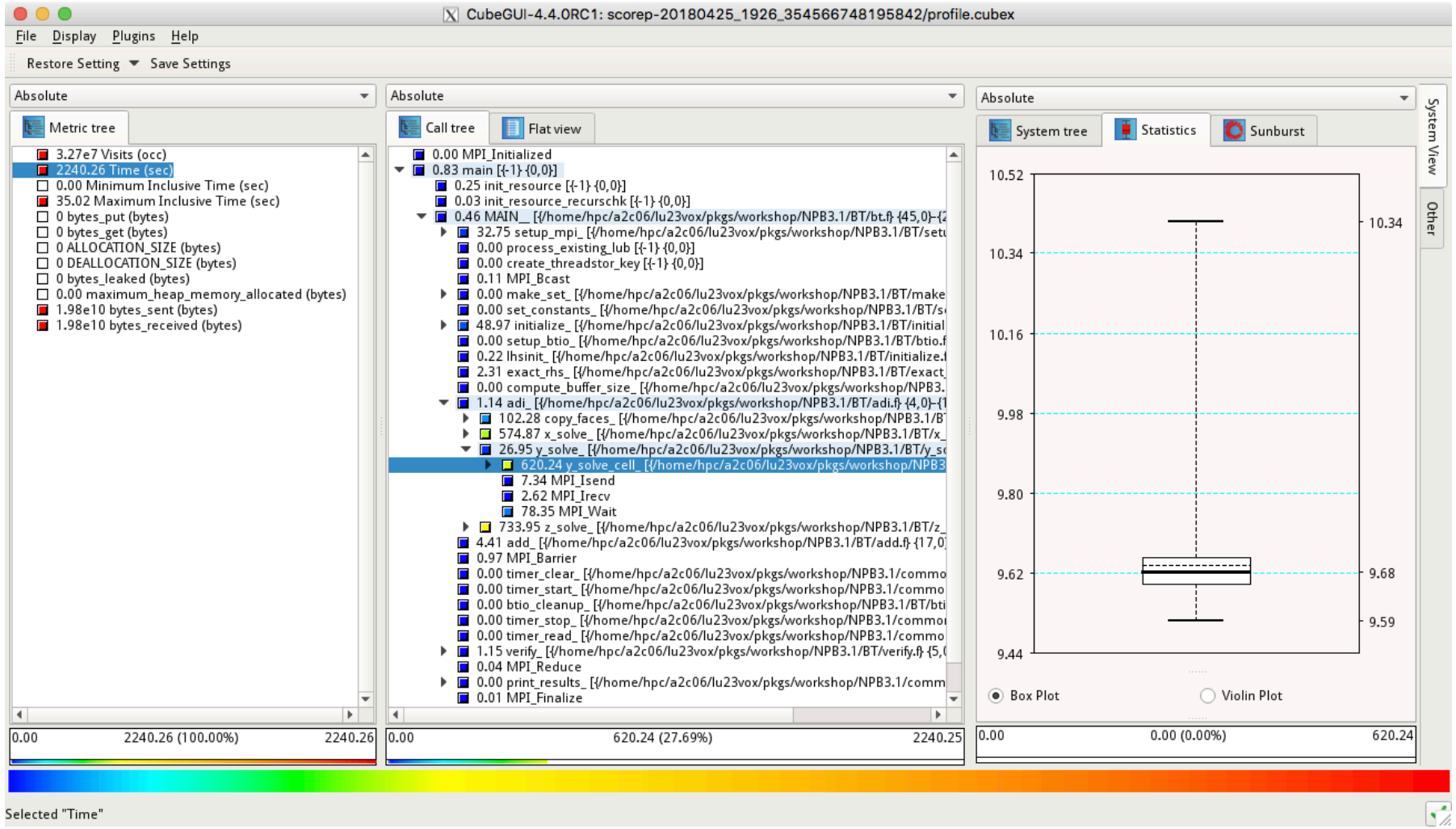


Vampir Timeline Window



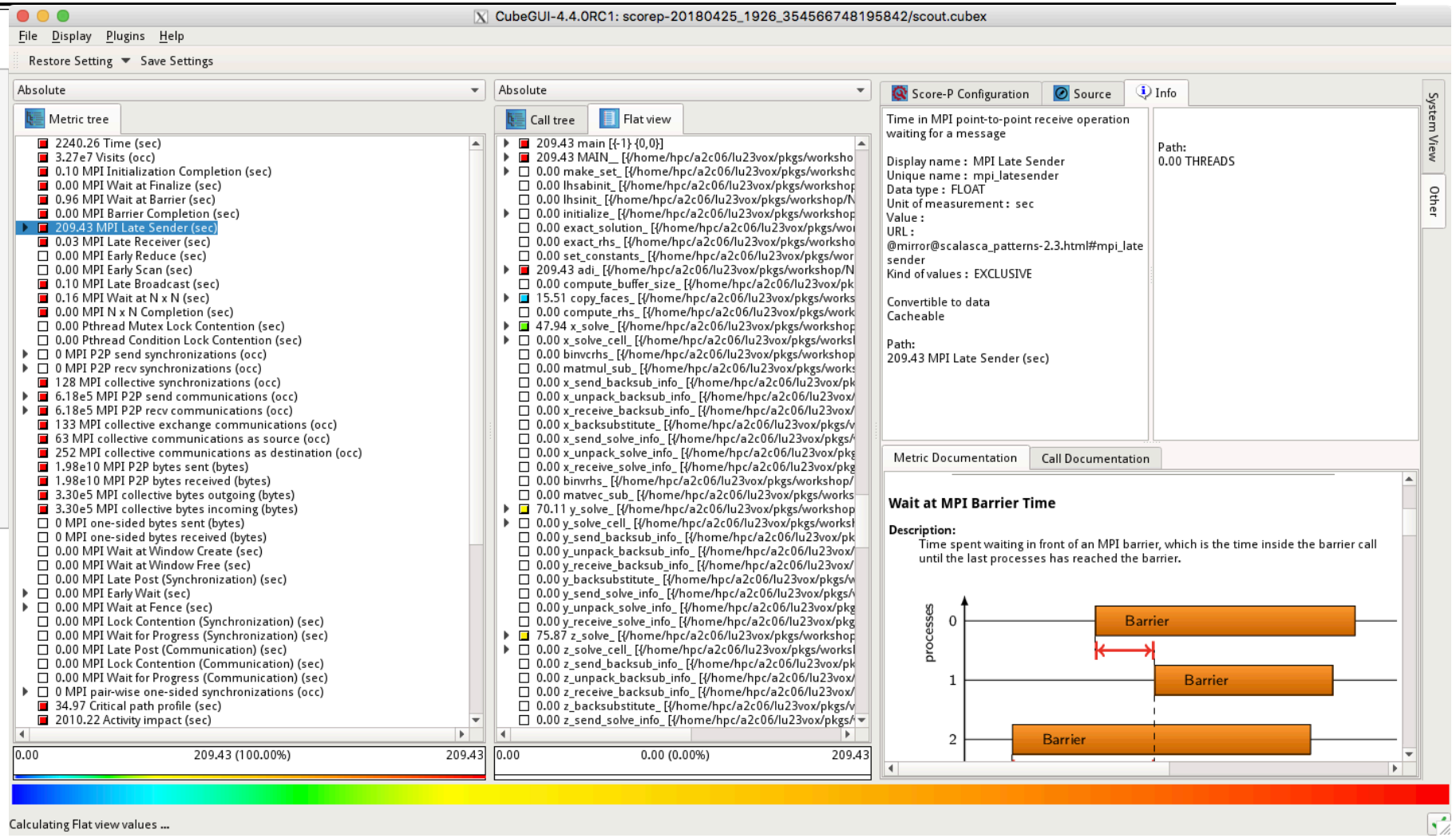
Examine CUBE files generated

```
% cube profile.cubex
```



Scalasca: Trace analysis on OTF2 traces generated traces

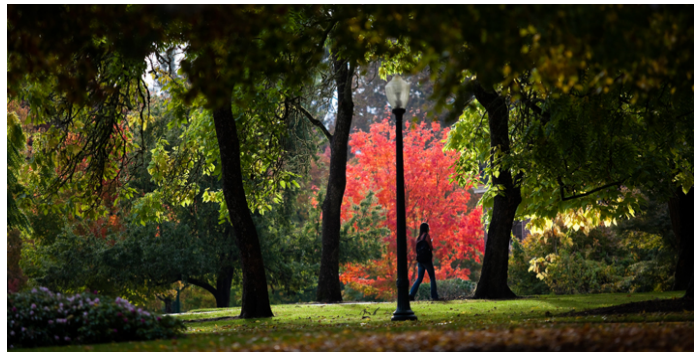
```
% cd scorep-<dir>
% mpirun -np 64 scout.mpi
# or use scan -t while
# launching job

# On login node:
% cube scout.cubex


# Also try:
% paraprof scout.cubex
```



What VI-HPS tools did we use?

- MAQAO for binary rewriting with tau_rewrite tool
- TAU's measurement library for generating profile files
- TAU's ParaProf profile browser to view profiles and create a filter/selective instrumentation file
- MAQAO to re-instrument the binary using TAU's Score-P configuration
- Score-P measurement library to generate CUBEX profiles and OTF2 traces natively
- Vampir to visualize the trace files
- CUBE to visualize profile files
- Scalasca's scout to search for performance properties (bottlenecks) in OTF2 traces
- CUBE to visualize the profile data generated by Scalasca
- TAU's ParaProf to visualize the performance bottlenecks
- **Many tools, but demonstrated good integration and interoperability of tools!**
- **No changes to the binary! No need to recompile or relink!**

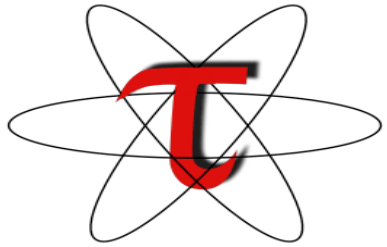
Performance Research Lab, University of Oregon, Eugene, USA



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 ECP contract
- Department of Defense (DoD)
  - PETTT, HPCMP
- National Science Foundation (NSF)
  - Glassbox, SI-2
- CEA, France
- NASA
- Partners:
  - University of Oregon
  - ParaTools, Inc., ParaTools, SAS
  - The Ohio State University
  - University of Tennessee, Knoxville
  - T.U. Dresden, GWT
  - Juelich Supercomputing Center

Download TAU from U. Oregon



http://tau.uoregon.edu

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

Free download, open source, BSD license