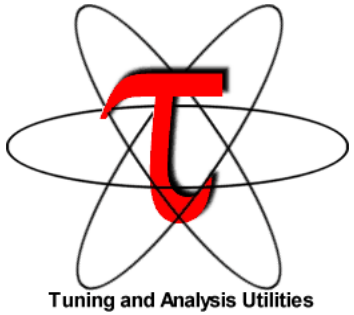
TAU Performance System®
Hands on session



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

```
% cd $WORK
% tar zxf ~hpclab11/tutorial/NPB3.3-MZ-MPI.tar.gz
% module load UNITE tau
```

NPB-MZ-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 NPB3.3-MZ-MPI root directory

```
% ls
bin/     common/  jobscript/  Makefile  README.install   SP-MZ/
BT-MZ/   config/  LU-MZ/      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-MZ-MPI / BT: config/make.def

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

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

...
#---------------------------------------------------------------------------
# The Fortran compiler used for MPI programs
#---------------------------------------------------------------------------
# MPIF77 = mpiifort # Intel compiler

# Alternative variant to perform instrumentation
MPIF77 = tau_f77.sh

# PREP is a generic preposition macro for instrumentation preparation
#MPIF77 = $(PREP) mpif77 -f77=ifort
#MPIF77 = scorep …
...
```

Default (no instrumentation)

Uncomment TAU's compiler wrapper to do source instrumentation with TAU
Comment out Score-P wrapper

Building an NPB-MZ-MPI Benchmark

```
% make
  ============================================
  =      NAS PARALLEL BENCHMARKS 3.3         =
  =      MPI+OpenMP Multi-Zone Versions      =
  =      F77                                 =
  ============================================

  To make a NAS multi-zone benchmark type

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

  where <benchmark-name> is "bt-mz", "lu-mz", or "sp-mz"
        <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-mz CLASS=C NPROCS=8                           *
 ***************************************************************
```

- Type "make" for instructions

Building an NPB-MZ-MPI Benchmark

```
% make suite
make[1]: Entering directory `BT-MZ'
make[2]: Entering directory `sys'
cc  -o setparams setparams.c -lm
make[2]: Leaving directory `sys'
../sys/setparams bt-mz 8 C
make[2]: Entering directory `../BT-MZ'
tau_f77.sh -c  -O3 -g -openmp       bt.f
[…]
tau_f77.sh -c  -O3 -g -openmp       mpi_setup.f
cd ../common;  mpiifort -c  -O3 -g -qopenmp        print_results.f
cd ../common;  mpiifort -c  -O3 -g -qopenmp        timers.f
tau_f77.sh -O3 -g -openmp   -o ../bin.tau/bt-mz_C.8 bt.o
 initialize.o exact_solution.o exact_rhs.o set_constants.o adi.o
 rhs.o zone_setup.o x_solve.o y_solve.o  exch_qbc.o solve_subs.o
 z_solve.o add.o error.o verify.o mpi_setup.o ../common/print_results.o
 ../common/timers.o
make[2]: Leaving directory `BT-MZ'
Built executable ../bin.tau/bt-mz_C.8
make[1]: Leaving directory `BT-MZ'
```

- Specify the benchmark configuration
  - benchmark name: **bt-mz**, lu-mz, sp-mz
  - the number of MPI processes: NPROCS=**8**
  - the benchmark class (S, W, A, B, C, D, E): CLASS=**C**

Shortcut: `% make suite`

NPB-MZ-MPI / BT (Block Tridiagonal Solver)

- What does it do?
  - Solves a discretized version of the unsteady, compressible Navier-Stokes equations in three spatial dimensions
  - Performs 200 time-steps on a regular 3-dimensional grid
- Implemented in 20 or so Fortran77 source modules

- Uses MPI & OpenMP in combination
  - 8 processes each with 6 threads should be reasonable

  - bt-mz_C.8 should take around 45 seconds

tau_exec

```
$ tau_exec

Usage: tau_exec [options] [--] <exe> <exe options>

Options:
        -v              Verbose mode
        -s              Show what will be done but don't actually do anything (dryrun)
        -qsub           Use qsub mode (BG/P only, see below)
        -io             Track I/O
        -memory         Track memory allocation/deallocation
        -memory_debug Enable memory debugger
        -cuda           Track GPU events via CUDA
        -cupti          Track GPU events via CUPTI (Also see env. variable TAU_CUPTI_API)
        -opencl         Track GPU events via OpenCL
        -openacc        Track GPU events via OpenACC (currently PGI only)
        -ompt           Track OpenMP events via OMPT interface
        -armci          Track ARMCI events via PARMCI
        -ebs            Enable event-based sampling
        -ebs_period=<count> Sampling period (default 1000)
        -ebs_source=<counter> Counter (default itimer)
        -um           Enable Unified Memory events via CUPTI
        -T <DISABLE,GNU,ICPC,MPI,OMPT,OPENMP,PAPI,PDT,PROFILE,PTHREAD,SCOREP,SERIAL> : Specify TAU tags
        -loadlib=<file.so>    : Specify additional load library
        -XrunTAUsh-<options> : Specify TAU library directly
        -gdb            Run program in the gdb debugger

Notes:
           Defaults if unspecified: -T MPI
           MPI is assumed unless SERIAL is specified
```

- Tau_exec preloads the TAU wrapper libraries and performs measurements.

No need to recompile the application!

tau_exec Example (continued)

```
Example:
    mpirun -np 2 tau_exec -T icpc,ompt,mpi  -ompt ./a.out
    mpirun -np 2 tau_exec -io ./a.out
Example - event-based sampling with samples taken every 1,000,000 FP instructions
    mpirun -np 8 tau_exec -ebs -ebs_period=1000000 -ebs_source=PAPI_FP_INS ./ring
Examples - GPU:
    tau_exec -T serial,cupti -cupti ./matmult (Preferred for CUDA 4.1 or later)
    tau_exec -openacc ./a.out
    tau_exec -T serial -opencl ./a.out (OPENCL)
    mpirun -np 2 tau_exec -T mpi,cupti,papi -cupti -um ./a.out (Unified Virtual Memory in CUDA 6.0+)

qsub mode (IBM BG/Q only):
    Original:
      qsub -n 1 --mode smp -t 10 ./a.out
    With TAU:
      tau_exec -qsub -io -memory -- qsub -n 1 … -t 10 ./a.out

Memory Debugging:
    -memory option:
      Tracks heap allocation/deallocation and memory leaks.
    -memory_debug option:
      Detects memory leaks, checks for invalid alignment, and checks for
      array overflow.  This is exactly like setting TAU_TRACK_MEMORY_LEAKS=1
      and TAU_MEMDBG_PROTECT_ABOVE=1 and running with -memory
```

- tau_exec can enable event based sampling while launching the executable using env TAU_SAMPLING=1 or tau_exec -ebs

NPB-MZ-MPI / BT with TAU uses OMPT for OpenMP instrumentation

```
% cd bin
% cp ../jobscript/claix/tau_exec.lsf .
% bsub < tau_exec.lsf
% cat mzmpibt.o<job_id>
 NAS Parallel Benchmarks (NPB3.3-MZ-MPI) - BT-MZ MPI+OpenMP Benchmark
 Number of zones:   16 x 16
 Iterations:  200   dt:   0.000300
 Number of active processes:     8
 Total number of threads:       48 (  6.0 threads/process)

 Time step    1
 Time step   20
  [...]
 Time step  180
 Time step  200
 Verification Successful

 BT-MZ Benchmark Completed.
 Time in seconds = 45.88
% paraprof &
% paraprof --pack bt.ppk
<Copy file over to desktop using scp>
% paraprof bt.ppk  &
```

- Copy jobscript and launch as a hybrid MPI+OpenMP application

Hint: save the benchmark output (or note the run time) to be able to refer to it later

Using Just MPI without any OpenMP flags or TAU

- Edit config/make.def to get rid of -qopenmp flag!

```
#COMPFLAGS = -qopenmp -extend-source # intel
COMPFLAGS = -extend-source # intel

MPIF77=mpif77
```

- OR just copy /tmp/tau/config.def to your config directory

```
% cp /tmp/tau/config.def NPB3.3-MZ-MPI/config
```

Compile BT-MZ Class A (for a smaller problem) WITHOUT TAU

- Build Class A benchmark

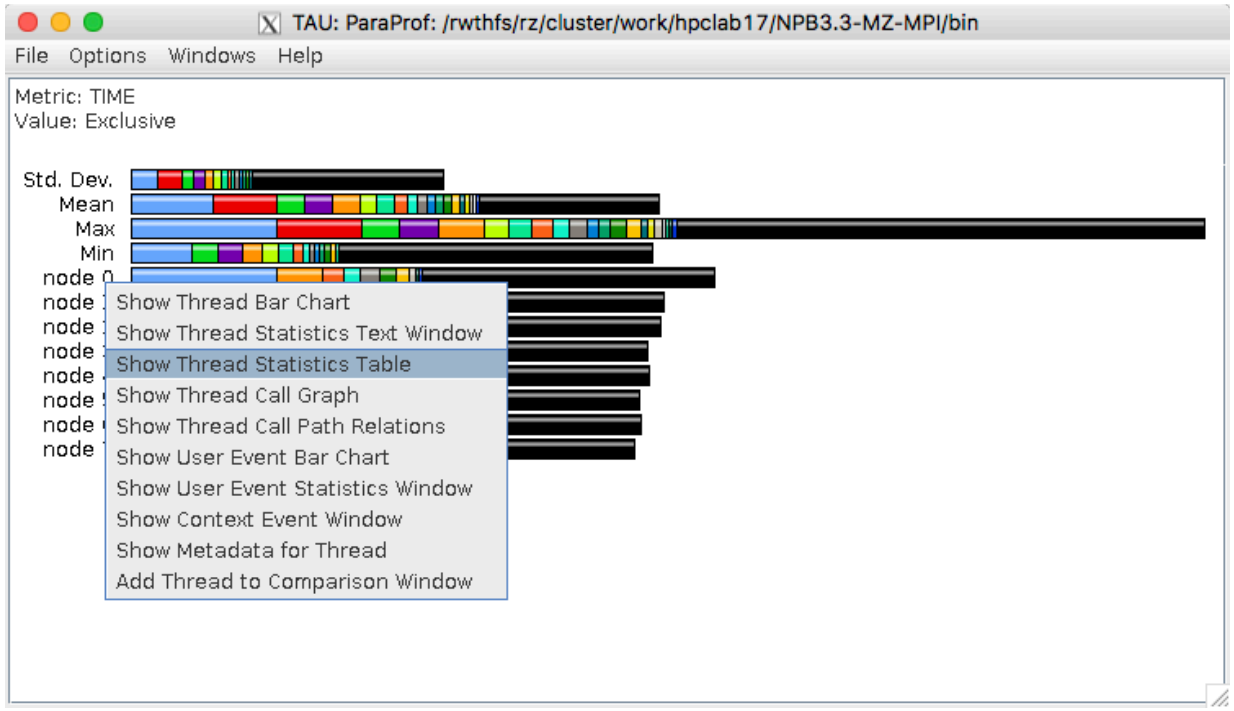
```
% cd NPB3.3-MZ-MPI
% make clean;
% make bt-mz CLASS=A NPROCS=8
% cd bin
% cp /tmp/tau/*.lsf .
% bsub < orig.lsf
% cat mzmpibt.*
```

```
BT-MZ Benchmark Completed.
Class           =                        A
Size            =              128x  128x 16
Iterations      =                      200
Time in seconds =                     8.92
Total processes =                        8
Total threads   =                        8
Mop/s total     =                 16389.32
Mop/s/thread    =                  2048.66
Operation type  =            floating point
Verification    =                SUCCESSFUL
Version         =                    3.3.1
Compile date    =              30 Mar 2017
```

Event Based Sampling with TAU

- Launch paraprof

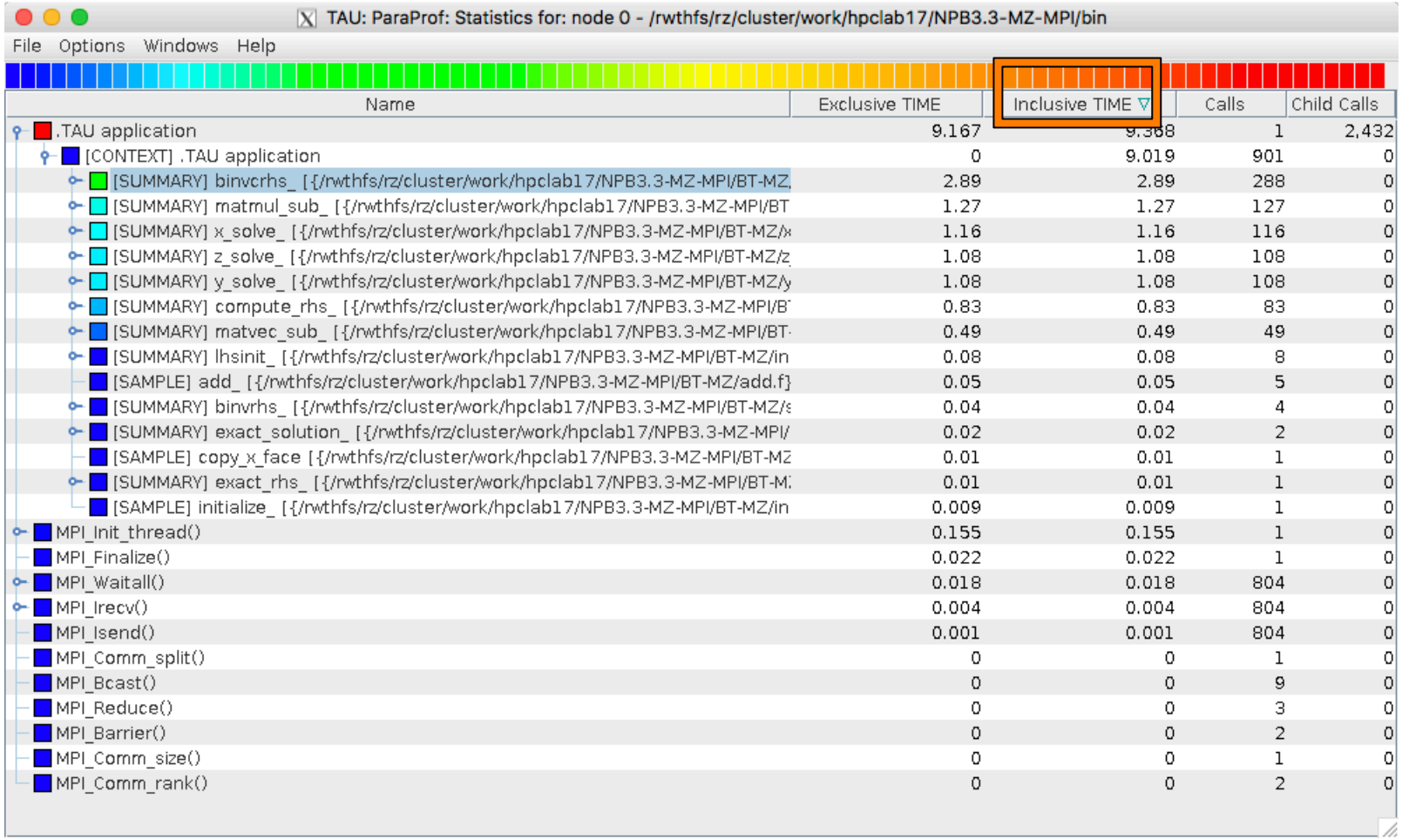
```
% cd NPB3.3-MZ-MPI
% make clean;
% make bt-mz CLASS=A NPROCS=8
% cd bin
% bsub < tau_1.lsf
% module load UNITE tau
% paraprof
```



- Right Click on Node 0 and choose Show Thread Statistics Table

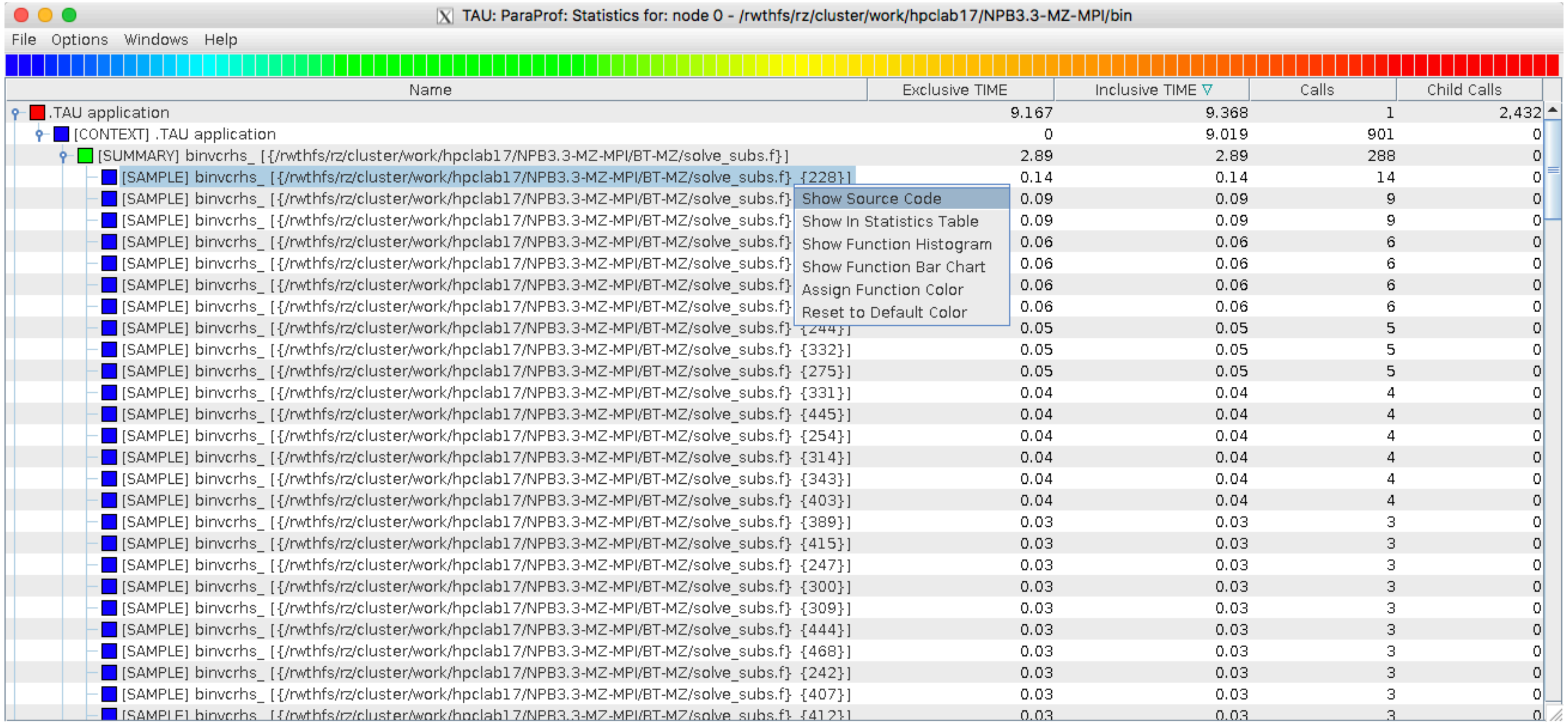
ParaProf

- Click on Columns: to sort by incl time
- Open binvcrhs
- Click on Sample



Name	Exclusive TIME	Inclusive TIME ▽	Calls	Child Calls
.TAU application	9.167	9.368	1	2,432
[CONTEXT] .TAU application	0	9.019	901	0
[SUMMARY] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ	2.89	2.89	288	0
[SUMMARY] matmul_sub_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT	1.27	1.27	127	0
[SUMMARY] x_solve_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/x	1.16	1.16	116	0
[SUMMARY] z_solve_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/z	1.08	1.08	108	0
[SUMMARY] y_solve_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/y	1.08	1.08	108	0
[SUMMARY] compute_rhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/B	0.83	0.83	83	0
[SUMMARY] matvec_sub_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT	0.49	0.49	49	0
[SUMMARY] lhsinit_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/in	0.08	0.08	8	0
[SAMPLE] add_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/add.f}	0.05	0.05	5	0
[SUMMARY] binvrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/s	0.04	0.04	4	0
[SUMMARY] exact_solution_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/	0.02	0.02	2	0
[SAMPLE] copy_x_face [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ	0.01	0.01	1	0
[SUMMARY] exact_rhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-M	0.01	0.01	1	0
[SAMPLE] initialize_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/in	0.009	0.009	1	0
MPI_Init_thread()	0.155	0.155	1	0
MPI_Finalize()	0.022	0.022	1	0
MPI_Waitall()	0.018	0.018	804	0
MPI_Irecv()	0.004	0.004	804	0
MPI_Isend()	0.001	0.001	804	0
MPI_Comm_split()	0	0	1	0
MPI_Bcast()	0	0	9	0
MPI_Reduce()	0	0	3	0
MPI_Barrier()	0	0	2	0
MPI_Comm_size()	0	0	1	0
MPI_Comm_rank()	0	0	2	0

ParaProf



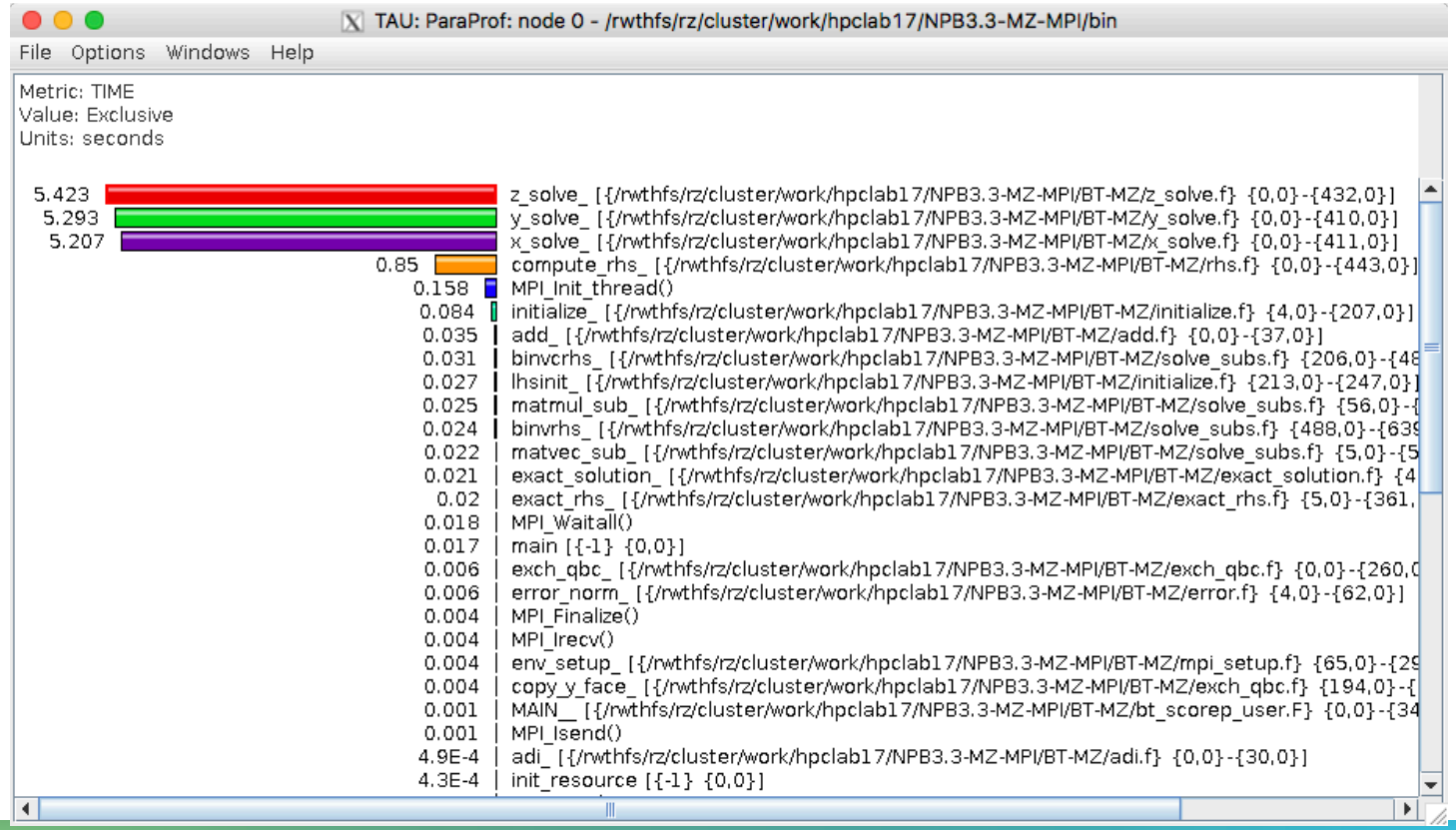
Name	Exclusive TIME	Inclusive TIME ▽	Calls	Child Calls
.TAU application	9.167	9.368	1	2,432
[CONTEXT] .TAU application	0	9.019	901	0
[SUMMARY] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f}]	2.89	2.89	288	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {228}]	0.14	0.14	14	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f}	0.09	0.09	9	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f}	0.09	0.09	9	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f}	0.06	0.06	6	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f}	0.06	0.06	6	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f}	0.06	0.06	6	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f}	0.06	0.06	6	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {244}]	0.05	0.05	5	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {332}]	0.05	0.05	5	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {275}]	0.05	0.05	5	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {331}]	0.04	0.04	4	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {445}]	0.04	0.04	4	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {254}]	0.04	0.04	4	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {314}]	0.04	0.04	4	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {343}]	0.04	0.04	4	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {403}]	0.04	0.04	4	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {389}]	0.03	0.03	3	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {415}]	0.03	0.03	3	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {247}]	0.03	0.03	3	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {300}]	0.03	0.03	3	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {309}]	0.03	0.03	3	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {444}]	0.03	0.03	3	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {468}]	0.03	0.03	3	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {242}]	0.03	0.03	3	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {407}]	0.03	0.03	3	0
[SAMPLE] binvcrhs_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/solve_subs.f} {412}]	0.03	0.03	3	0

Instrument binary using MAQAO: tau_rewrite

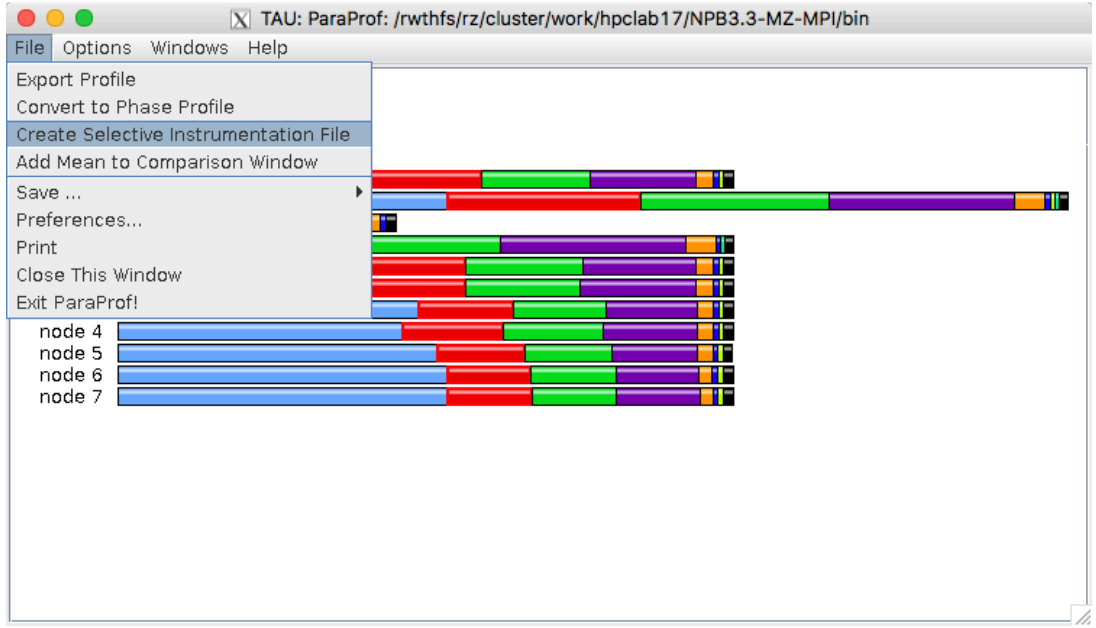
- We can read an uninstrumented binary and generate an instrumented binary
- All functions are being instrumented (higher overhead). TAU throttles.

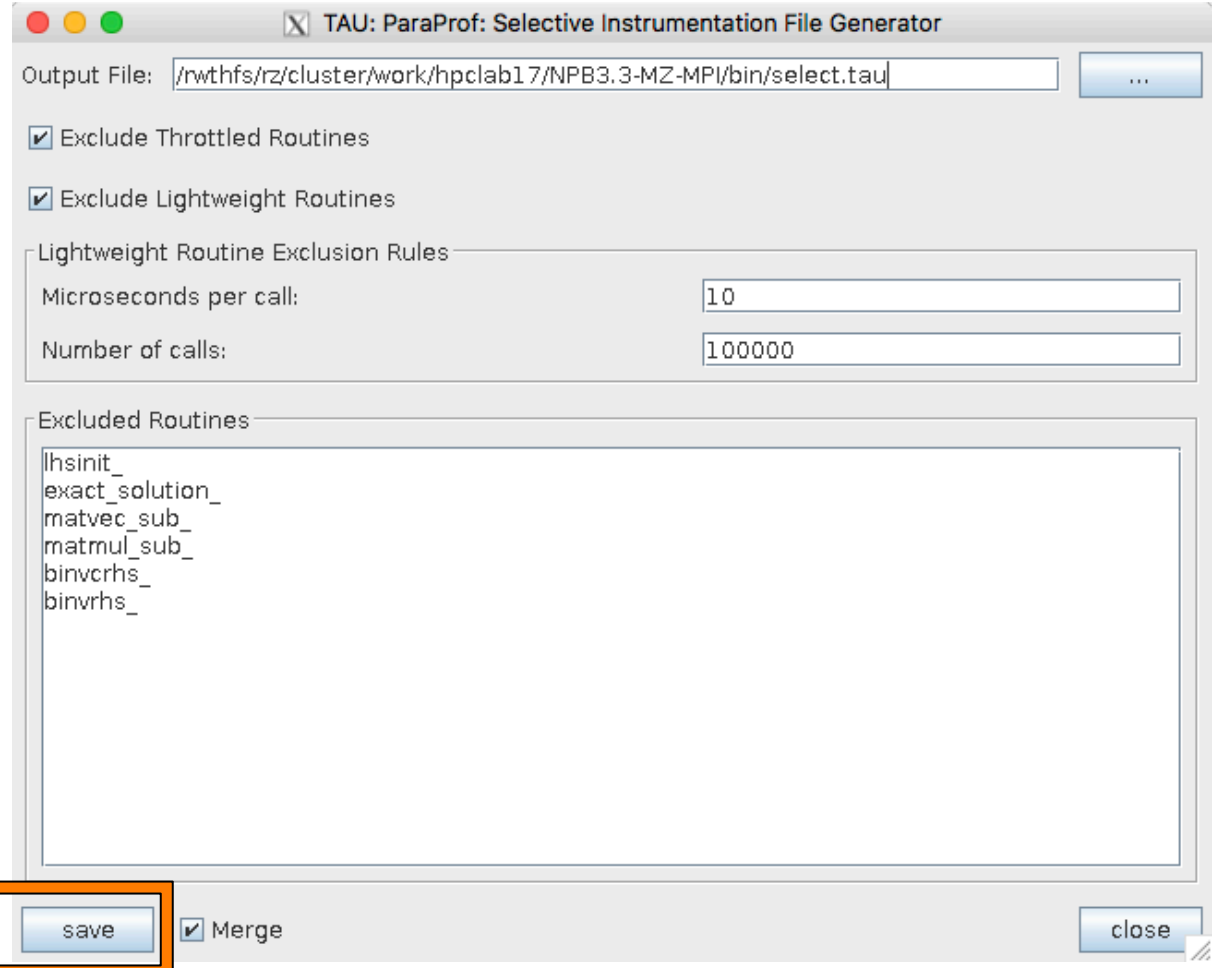
```
% tau_rewrite bt-mz_A.8 bt.i
% bsub < tau_2.lsf
% cat mzmpibt.<jobid>
```

```
BT-MZ Benchmark Completed.
Class           =                        A
Size            =            128x  128x 16
Iterations      =                      200
Time in seconds =                    16.79
Total processes =                        8
Total threads   =                        8
Mop/s total     =                  8708.93
Mop/s/thread    =                  1088.62
Operation type  =            floating point
Verification    =                SUCCESSFUL
Version         =                    3.3.1
Compile date    =              30 Mar 2017
```



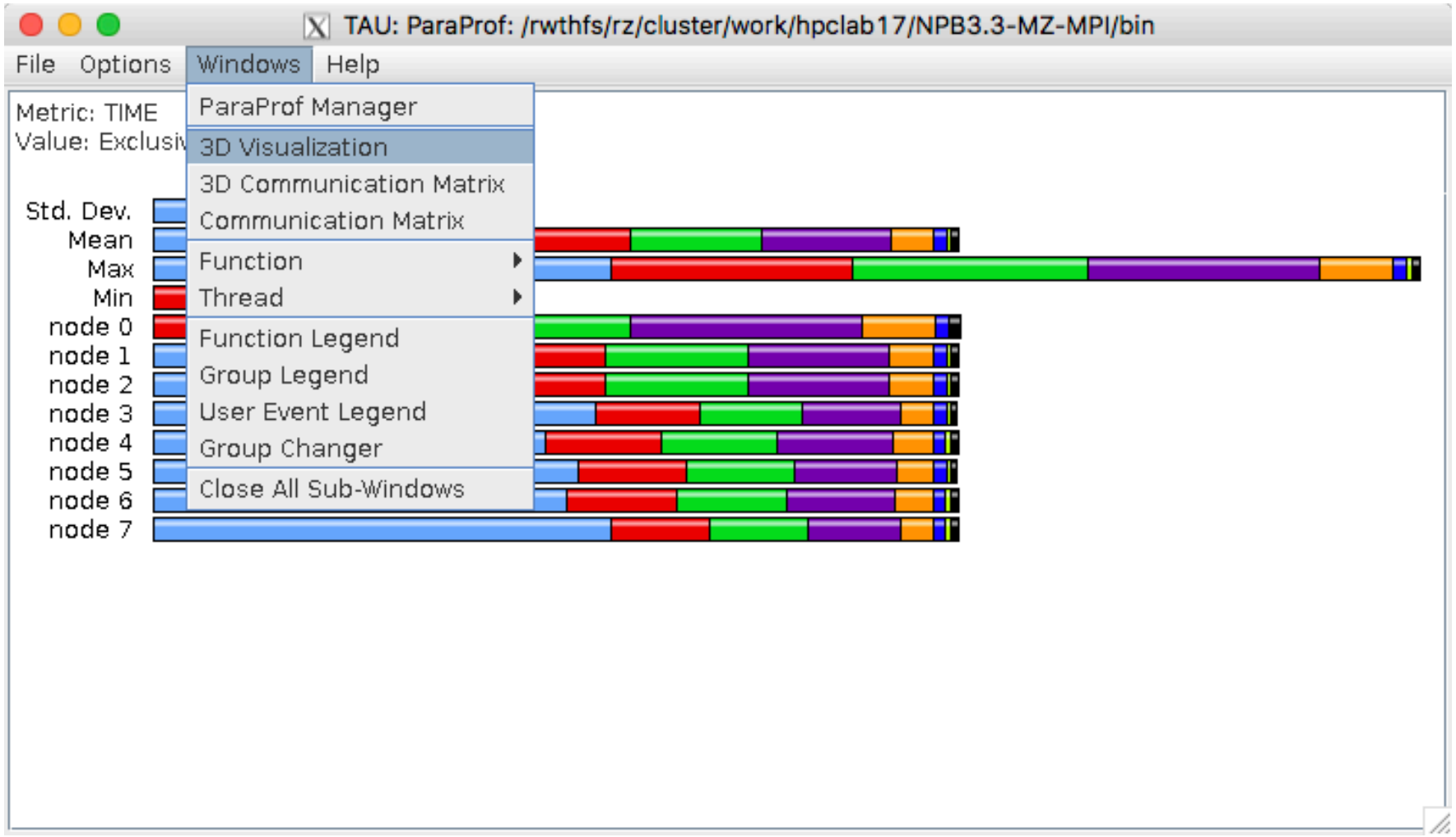
Create a Selective Instrumentation File, Re-instrument, Re-run



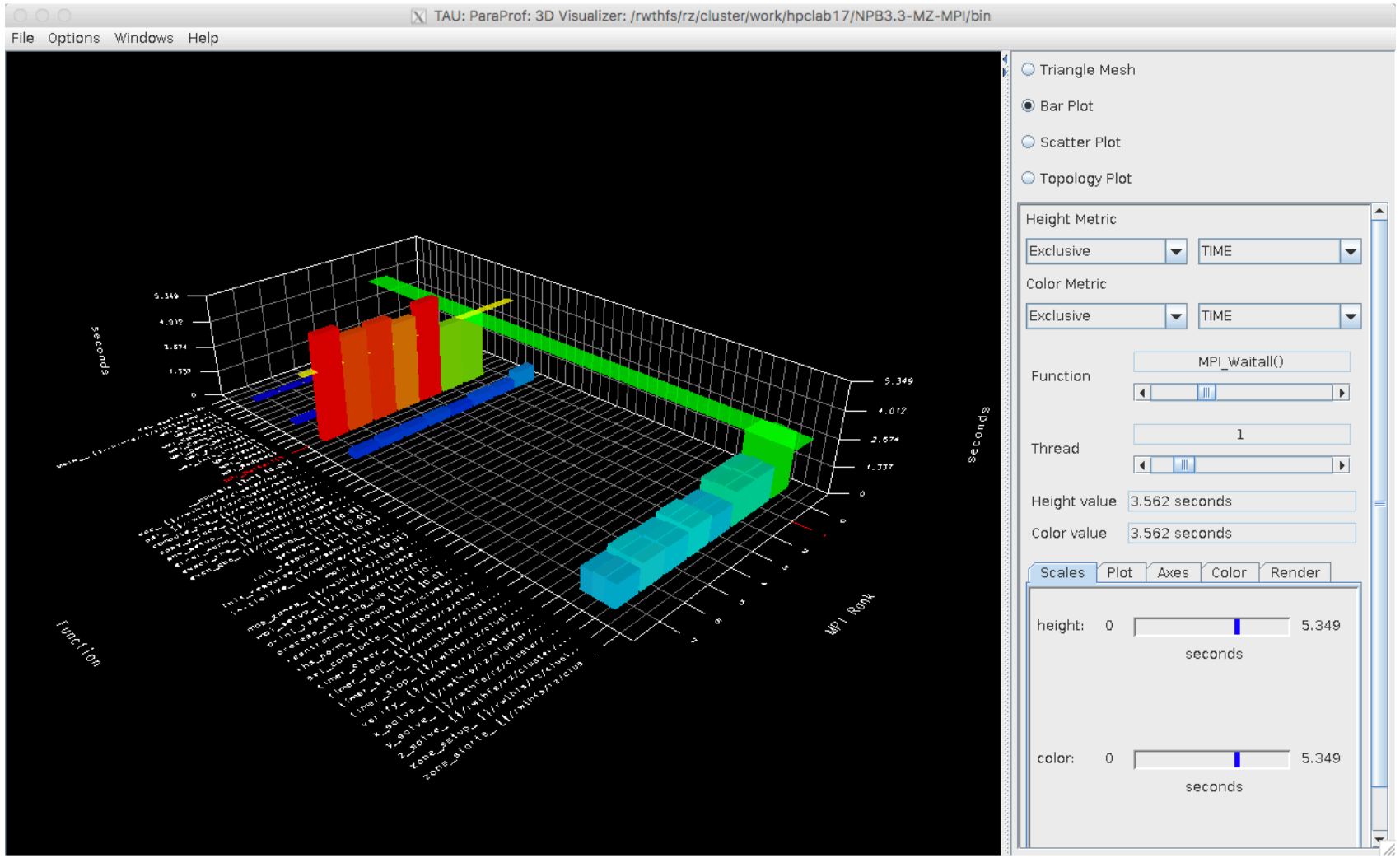



```
% tau_rewrite -f select.tau bt-mz_A.8 bt.i
% bsub < tau_3.lsf
% cat mzmpibt.<jobid>
 Iterations      =                    200
 Time in seconds =                   9.13
% paraprof &
```

ParaProf with Optimized Instrumentation

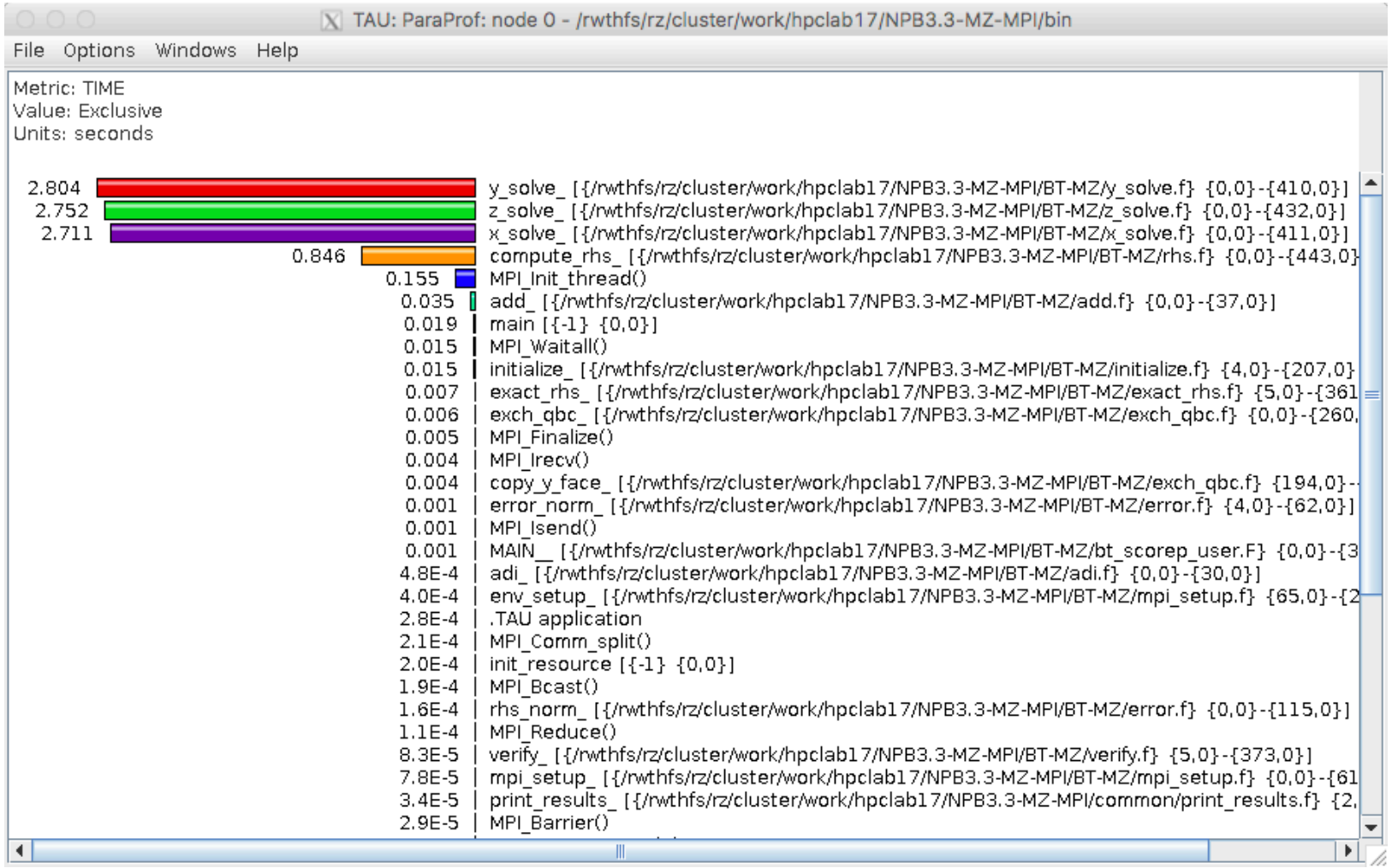


3D Visualization with ParaProf



ParaProf: Node 0

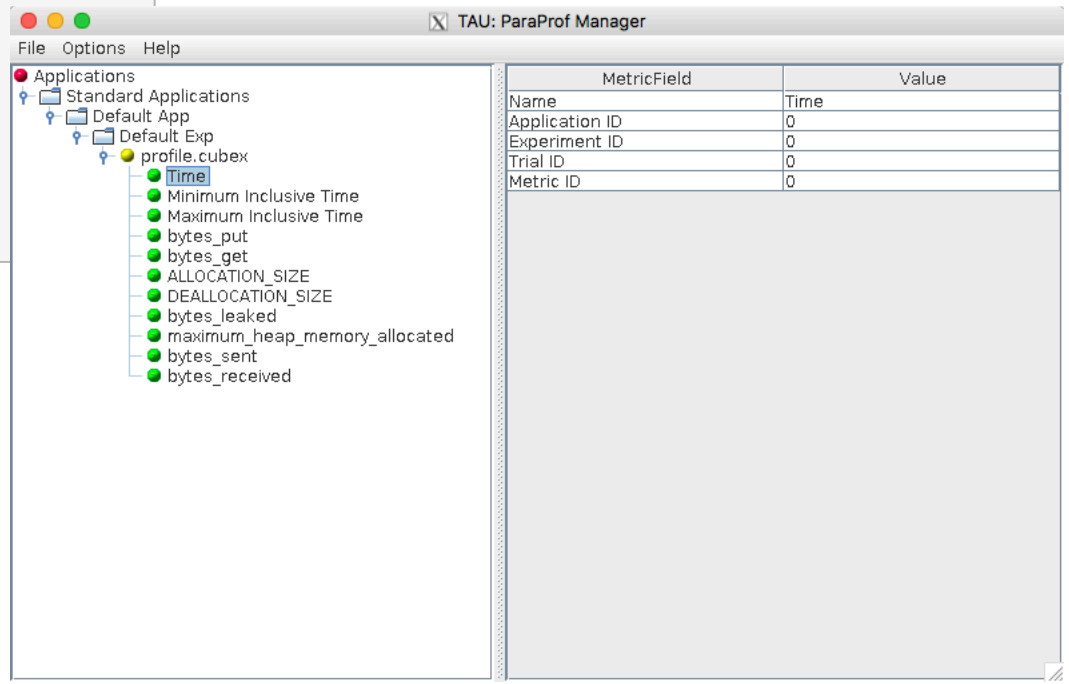
- Optimized instrumentation!



Using Score-P with TAU and MAQAO

- Making profile measurements with Score-P using TAU

```
% tau_rewrite -f select.tau -T scorep ./bt-mz_A.8 bt.i
% bsub < tau_4.lsf
export SCOREP_EXPERIMENT_DIRECTORY=scorep_bt-mz_sum
$MPIEXEC $FLAGS_MPI_BATCH  ./bt.i
% cd scorep_bt-mz_sum
% module load UNITE cube
% cube profile.cubex &
% paraprof profile.cubex &
```



- TAU's paraprof can read CUBEX format!
- Choose metric by double-clicking on it

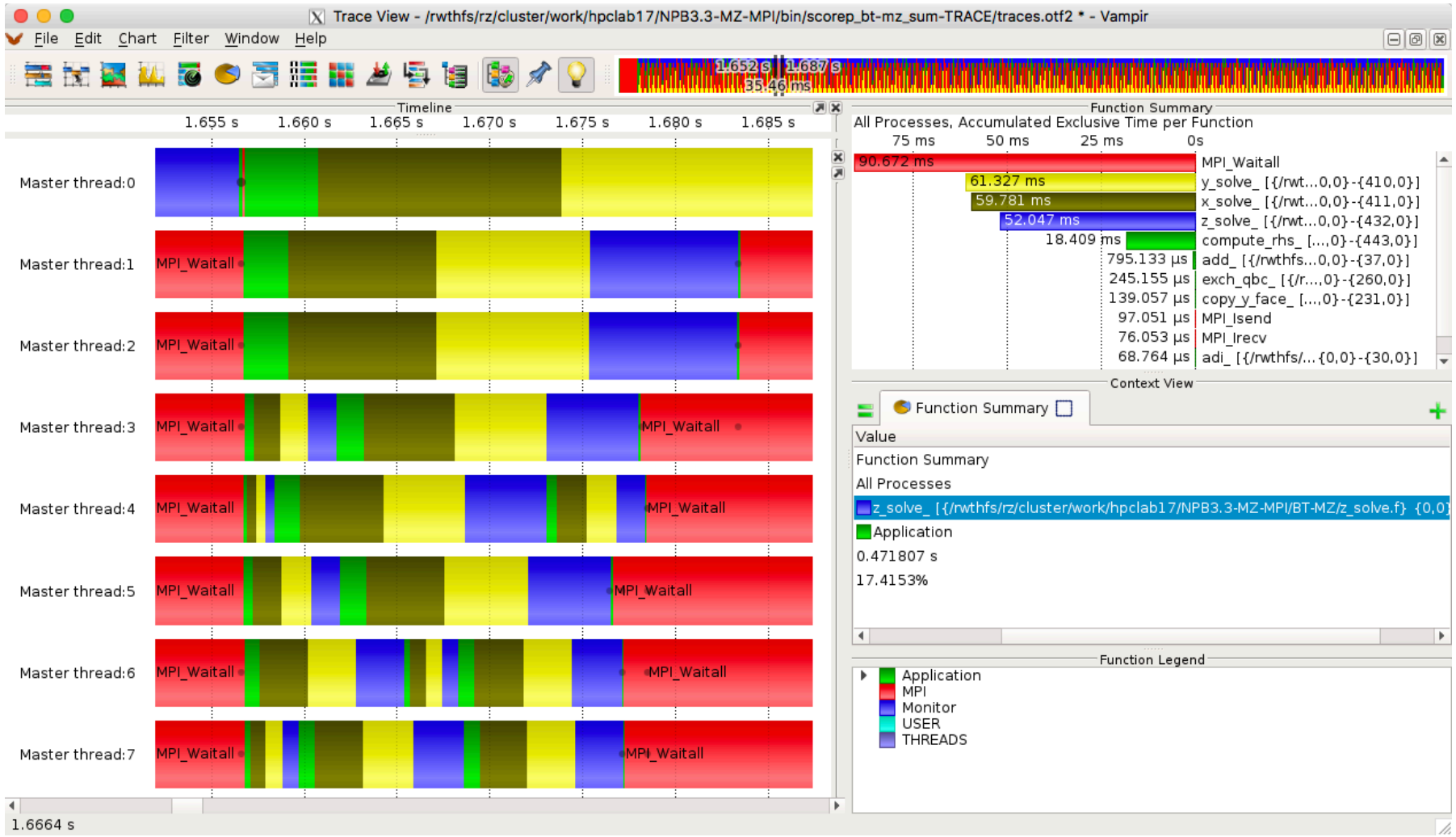
Generating traces for Vampir and Scalasca!

- Bringing it all together: TAU+MAQAO+Score-P+CUBE+Vampir!

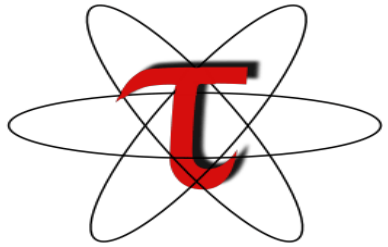
```
% tau_rewrite -f select.tau -T scorep ./bt-mz_A.8 bt.i
% cat tau_5.lsf
…
export SCOREP_EXPERIMENT_DIRECTORY=scorep_bt-mz_sum-TRACE
export SCOREP_ENABLE_TRACING=true

$MPIEXEC $FLAGS_MPI_BATCH  ./bt.i
% bsub < tau_5.lsf
% cd scorep_bt-mz_sum-TRACE
% module load UNITE vampir
% vampir traces.otf2 &
```

VI-HPS: A suite of integrated performance tools!

Vampir


Trace View - /rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/bin/scorep_bt-mz_sum-TRACE/traces.otf2 * - Vampir
File Edit Chart Filter Window Help
Timeline
1.655 s 1.660 s 1.665 s 1.670 s 1.675 s 1.680 s 1.685 s
Master thread:0
Master thread:1
Master thread:2
Master thread:3
Master thread:4
Master thread:5
Master thread:6
Master thread:7
MPI_Waitall
1.6664 s
Function Summary
All Processes, Accumulated Exclusive Time per Function
75 ms 50 ms 25 ms 0s
90.672 ms MPI_Waitall
61.327 ms y_solve_ [{/rwt...0,0}-{410,0}]
59.781 ms x_solve_ [{/rwt...0,0}-{411,0}]
52.047 ms z_solve_ [{/rwt...0,0}-{432,0}]
18.409 ms compute_rhs_ [...,0}-{443,0}]
795.133 µs add_ [{/rwthfs...0,0}-{37,0}]
245.155 µs exch_qbc_ [{/r...,0}-{260,0}]
139.057 µs copy_y_face_ [...,0}-{231,0}]
97.051 µs MPI_Isend
76.053 µs MPI_Irecv
68.764 µs adi_ [{/rwthfs/...{0,0}-{30,0}]
Context View
Function Summary
Value
Function Summary
All Processes
z_solve_ [{/rwthfs/rz/cluster/work/hpclab17/NPB3.3-MZ-MPI/BT-MZ/z_solve.f} {0,0}
Application
0.471807 s
17.4153%
Function Legend
Application
MPI
Monitor
USER
THREADS

Download TAU from U. Oregon



http://tau.uoregon.edu

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

Free download, open source, BSD license