Tau Lab

Login to Ranger with X tunneling turned on (-X or -Y).

% ssh –X –l train## ranger.tacc.utexas.edu

You will now have a “login4 % or login3 % prompt; but for brevity, only “%” will be displayed below.

Make sure you can display back to your (Linux, Windows, Mac) workstation.

% xclock

You should get this clock:

Either close the clock or type ^c (control-c).

Short Lab: Use provided Sample

% tar –xvf ~train00/lab_tau_short.tar

% cd tau_short — READ the Instructions file

% source sourceme.csh — If you are using a C shell (all class accounts)

or

% source sourceme.sh — If you are using a Bourne shell

Long Lab: Build and Submit Example

% tar –xvf ~train00/lab_tau_long.tar

% cd tau READ the Instructions file

% source sourceme.csh If you are using a C shell (all class accounts)

or

% source sourceme.sh If you are using a Bourne shell

% make matmultf create executable for F90 programmers

or

% make matmultc create executable, for C programmers

% qsub job submit job (edit and uncomment ibrun line)

Wait for the output to return, using **`watch`** to view qstat info every 8 seconds:

% watch –n 8 qstat

Exit out of **`watch`** once display shows no queued job, by typing ^c (control-c).

View Profile Information

Look for these directories once the job is finished:
MULTI__GET_TIME_OF_DAY

MULTI__FP_OPS

MULTI__PAPI_L1_DCM

When you launch **`paraprof`**, it will automatically include these directories as “trials”.

% paraprof **(for GUI)** Analyze performance data

Experiment with paraprof options

The Get_Time_of_Day profile
will be displayed automatically

Display the Legend	Windows→Function Legend

(Observe that core 0 is the master and does no mxm work.)

Turn off normalization.	Options→Normalize Bars
Line up bars	Options→Stack Bars Together
Change from usec to seconds	File→preferences… In the **Units** menu select **seconds**, then click apply.

Display Profile for a single function

Show MULTIPLY_MATRICES. Function results.

Click on any blue bar.

Display Statistics

Show the statistics for .
core 1 and core 15.

Within the profile window click
Windows→Threads→Statistics Table.
In new window select n,c,t 15,0,0,
followed by n,c,t 1,0 0.

Determine the message size(s)
of the MPI_Rec function

Within the profile window click
Windows→Threads→
User Event Statistics. (Select any
n, c, t value.) What is the difference in
the number of Send/Receives for task 1
and task 15?

Derive the FLOPS/Cache Miss

Show that the FLOPS/Cache miss ratio is constant for all Cores.	In the manager window click: Options→ Show Derived Metrics Panel Click on the PAPI_L1_DCM trial. (Value appears as Argument 1: box.) Click on PAPI_FP_OPS (Value appears in Argument 1: box, PAPI_L1_DCM is moved to Arg. 2 box) Click on Apply Operation.
Observe the Ratio Profile	Click on “PAPI_FP_OPS + PAPI_L1_DCM” trial.

Other Experiments (Long Lab only!)

Experiment 1 **

Look at the available PAPI counters in the papi_counters_on_nodes file and run the job with a different set of counters (COUNTER2 and COUNTER3):

1.) Edit job (change counters)
2.) Submit job (qsub job)
3.) Run paraprof when job has completed.

Experiment 2 **

Look through the list of TAU_MAKEFILES by executing:

% tauTypes

** Move your old MULT__xxx directories to a subdirectory so that they are not removed and can be viewed for comparison.

Use the callpath makefile (Makefile.tau-callpath-icpc-mpi-pdt) to create an experiment that shows the call tree with these commands:

```
  % setenv TAU_MAKEFILE $MYPKGDIR/x86_64/lib/Makefile.tau-callpath-icpc-mpi-pdt
or
  % source sourceme_callpath.csh
  % make clean
  % make   or  make matmultc
  % qsub job
```

Other Experiments

** Move your old MULT__xxx directories to a subdirectory
so that they are not removed and can be viewed for comparison.

Experiment 3

Edit matmultc.c or matmultf.f and change the broadcast so that B is broadcast as a matrix instead of a set of columns Now recompile, run and compare:
make clean; make matmultc or make matmultf; qsub job; … paraprof.