# Profiling and Tracing TAU and pgprof

Timothy H. Kaiser, Ph.D.
tkaiser@mines.edu

# Tracing and Profiling - TAU

- A portable profiling and tracing toolkit for performance analysis of parallel programs written in Fortran, C, C++, Java, Python
- Capable of gathering performance information through instrumentation of functions, methods, basic blocks, and statements
- Profile visualization tool, paraprof, provides graphical displays of all the performance analysis results
- http://www.cs.uoregon.edu/research/tau/home.php
- http://petra.mines.edu/tau/

# TAU - a few details

- Currently available for use with MPI versions:
  - /opt/ra5_openmpi_intel/1.4.1
  - “Standard” version

- Setup for bash (put in your .bashrc file)

```
export JAVA_HOME=/lustre/home/apps/java/jdk1.6.0_13
export LD_LIBRARY_PATH=/lustre/home/apps/papi/lib:$LD_LIBRARY_PATH
export PATH=/lustre/home/apps/java/jdk1.6.0_13/bin:$PATH
export PATH=/lustre/home/apps/TAU/2.19.2/ra5_openmpi_intel/1.4.1/x86_64/bin:$PATH
export PATH=/lustre/home/apps/TAU/pdtoolkit-3.16/x86_64/bin:$PATH
export TAU_MAKEFILE=/lustre/home/apps/TAU/2.19.2/ra5_openmpi_intel/1.4.1/x86_64/lib/\
Makefile.tau-depthlimit-icpc-papi-mpi-pthread-compensate-python-pdt-openmp-profile-trace-mpitrace
```

or

```
source /lustre/home/apps/TAU/2.19.2/ra5_openmpi_intel/1.4.1/setup
```

# TAU - a few more details

- Need to use Tau "compilers" which are preprocessing scripts
- Takes original code and inserts instrumentation
- Calls the real compiler
- Does not like if(mytype%a .and. act%b)then

# Stommel makefile changes

```
[tkaiser@ra5 stommel]$ diff -c makefile makefile.tau
*** makefile  2008-09-30 12:30:14.000000000 -0600
--- makefile.tau  2010-07-22 14:11:33.000000000 -0600
***************
*** 4,13 ****
--- 4,16 ----
  c:  $(ALLC)
  f:  $(ALLF)

+ TAUROOTDIR      = /lustre/home/apps/TAU/2.19.2/ra5_openmpi_intel/1.4.1
+ include $(TAUROOTDIR)/include/Makefile


  SFC=ifort
  FC=mpif77
+ FC    = $(TAUROOTDIR)/$(TAU_ARCH)/bin/tau_f90.sh
  FFLAGS=-O3

  SCC=icc
[tkaiser@ra5 stommel]$
```

# Stommel pbs script changes

```
#!/bin/bash
#PBS -l nodes=1:ppn=8
#PBS -l walltime=02:00:00
#PBS -N testIO
##PBS -o out.$PBS_JOBID
##PBS -e err.$PBS_JOBID
#PBS -o stdout
#PBS -e stderr
#PBS -r n
#PBS -V
#-----------------------------------------------

cd $PBS_O_WORKDIR

export TAU_TRACE=yes
export TAU_PROFILE=yes

#save a nicely sorted list of nodes
sort  -u $PBS_NODEFILE > mynodes.$PBS_JOBID

mpiexec -np 8 ./stf_03 < st.in
```

If you elect to do a **trace**, that is a **time history**, you will have two sets of files generated during your run, events.*.edf and tautrace.*.trc. There will be one of each of these files for each process.

If you elect to do a **profile**, that is a **summary of where time is spent**, you will have a profile.* file for each process.

# Extra files created at run time

```
[rawork0100@ra stommel]$ ls -l events* profile* tautrace*
-rw-rw-r--  1 rawork0100 rawork0100   3586 Aug 17 15:54 events.0.edf
-rw-rw-r--  1 rawork0100 rawork0100   3366 Aug 17 15:54 events.1.edf
-rw-rw-r--  1 rawork0100 rawork0100   3366 Aug 17 15:54 events.2.edf
-rw-rw-r--  1 rawork0100 rawork0100   3366 Aug 17 15:54 events.3.edf
-rw-rw-r--  1 rawork0100 rawork0100   3366 Aug 17 15:54 events.4.edf
-rw-rw-r--  1 rawork0100 rawork0100   3366 Aug 17 15:54 events.5.edf
-rw-rw-r--  1 rawork0100 rawork0100   3366 Aug 17 15:54 events.6.edf
-rw-rw-r--  1 rawork0100 rawork0100   3366 Aug 17 15:54 events.7.edf

-rw-------  1 rawork0100 rawork0100 820272 Aug 17 15:54 tautrace.0.0.0.trc
-rw-------  1 rawork0100 rawork0100 906576 Aug 17 15:54 tautrace.1.0.0.trc
-rw-------  1 rawork0100 rawork0100 906576 Aug 17 15:54 tautrace.2.0.0.trc
-rw-------  1 rawork0100 rawork0100 762384 Aug 17 15:54 tautrace.3.0.0.trc
-rw-------  1 rawork0100 rawork0100 762384 Aug 17 15:54 tautrace.4.0.0.trc
-rw-------  1 rawork0100 rawork0100 906576 Aug 17 15:54 tautrace.5.0.0.trc
-rw-------  1 rawork0100 rawork0100 906576 Aug 17 15:54 tautrace.6.0.0.trc
-rw-------  1 rawork0100 rawork0100 762384 Aug 17 15:54 tautrace.7.0.0.trc

-rw-rw-r--  1 rawork0100 rawork0100   7302 Aug 17 15:54 profile.0.0.0
-rw-rw-r--  1 rawork0100 rawork0100   7078 Aug 17 15:54 profile.1.0.0
-rw-rw-r--  1 rawork0100 rawork0100   7080 Aug 17 15:54 profile.2.0.0
-rw-rw-r--  1 rawork0100 rawork0100   7052 Aug 17 15:54 profile.3.0.0
-rw-rw-r--  1 rawork0100 rawork0100   7058 Aug 17 15:54 profile.4.0.0
-rw-rw-r--  1 rawork0100 rawork0100   7082 Aug 17 15:54 profile.5.0.0
-rw-rw-r--  1 rawork0100 rawork0100   7082 Aug 17 15:54 profile.6.0.0
-rw-rw-r--  1 rawork0100 rawork0100   7056 Aug 17 15:54 profile.7.0.0
```

# Post Processing and viewing

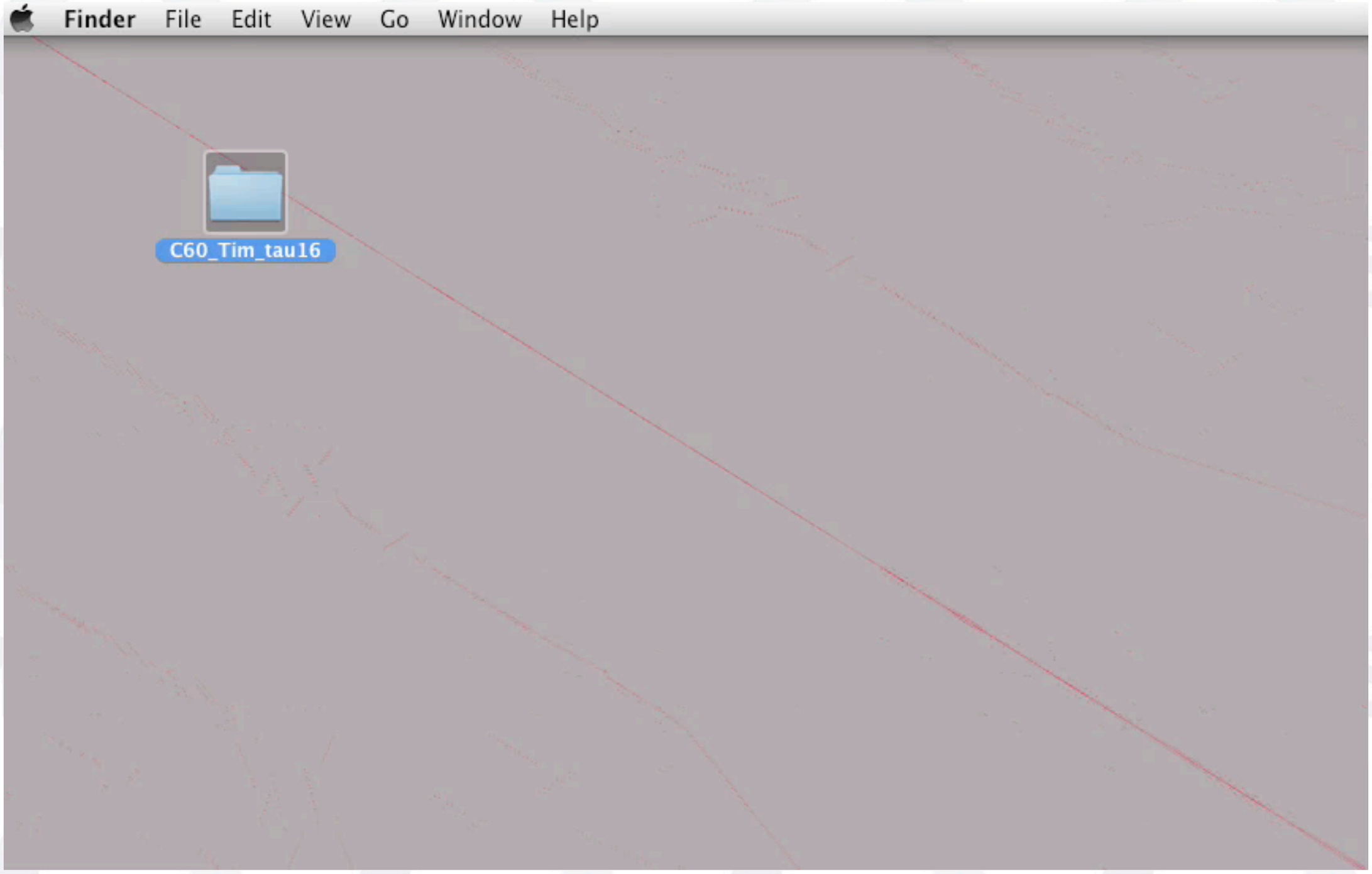
- The trace and profile files are not in a from that can be read by the analysis tools. To convert them run the commands:
  - `tau_treemerge.pl`
  - `tau2slog2 tau.trc tau.edf -o tau.slog2`
  - `pprof -a > tau.profile`
- The first two commands are for traces and the third is for profiles.

# Post Processing and viewing

- Our primary profile and trace visualization tools for use with TAU are paraprof and jumpshot.
  - These tools are available on RA but it is often better to download the TAU outputs and visualize locally.
  - The files from the TAU web page
    - `tau-2.18.2.dmg`
    - `tau-2.18.2p1.exe`
- Contain copies of these tools for OSX and Windows.

# Example @ http://petra.mines.edu/tau/

- The local TAU web page contains a collection of run PARSEC examples
- PARSEC (http://parsec.ices.utexas.edu/) is a real-space density-functional theory code for electronic structure calculations.
- 0501_112246C60_Tim_tau8.tgz ( 107 Mbytes)
  - Parsec C60 8-core run with profiles
- 0501_115133C60_Tim_tau16.tgz ( 107 Mbytes)
  - Parsec C60 16-core run with profiles
- 0501_131002C60_Tim_tau8.tgz( 242 Mbytes)
  - Parsec C60 8-core run with profiles and trace files
- paraprof.mov
  - A movie of paraprof used to visualize the 16 core data set


Finder
File
Edit
View
Go
Window
Help
C60_Tim_tau16

# Local success stories

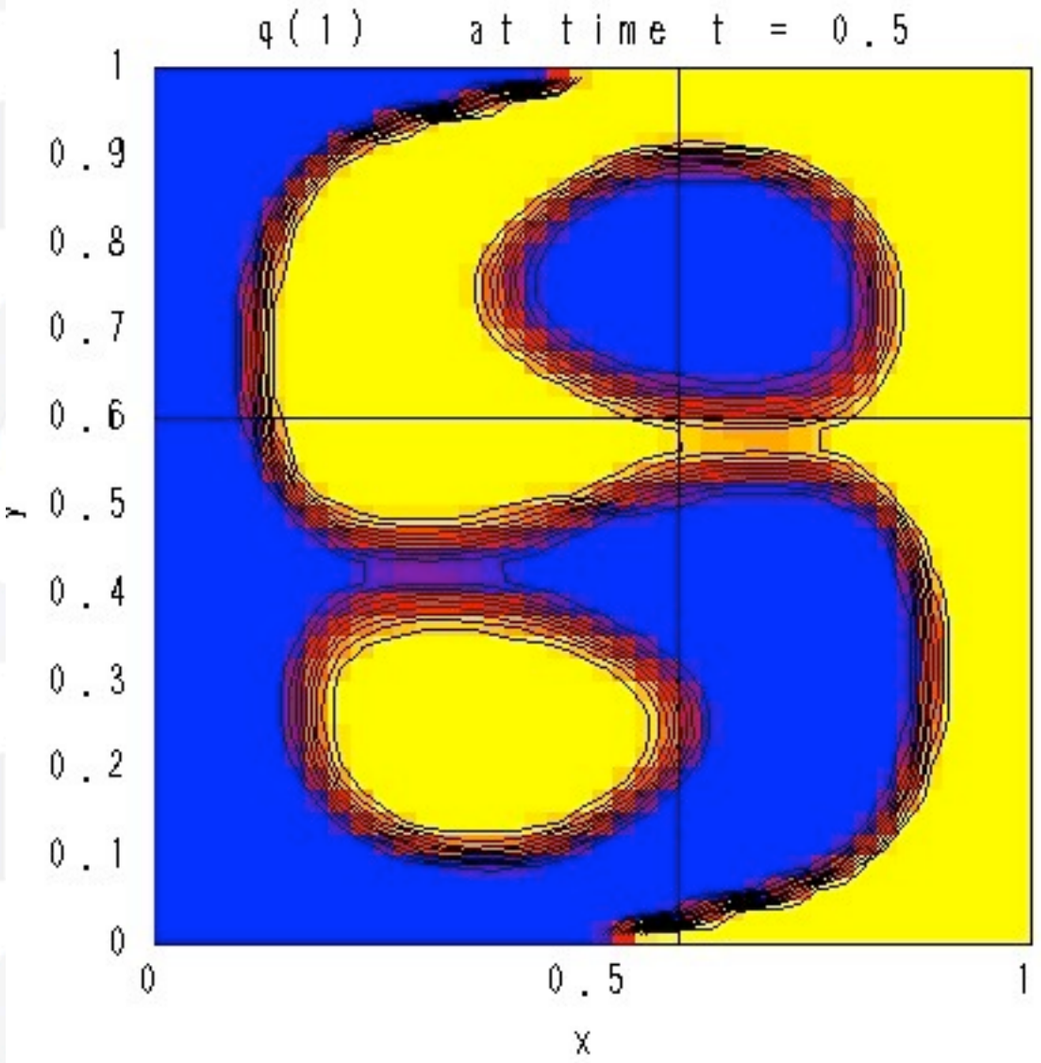
- Parallel GA - found time critical routine cut runtime down by 45%
- Parsec - found reason why it was not scaling well

# Another use:

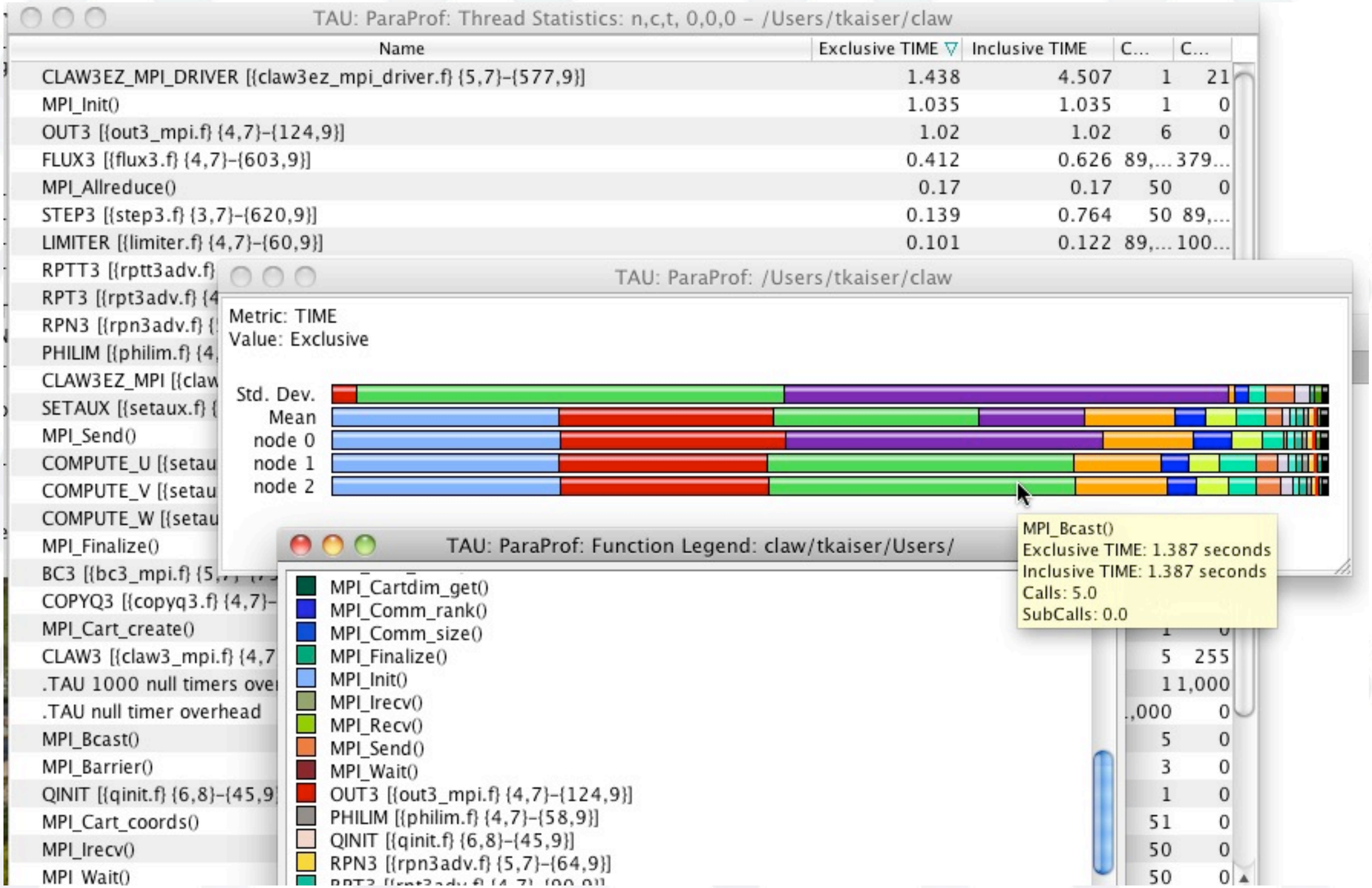
## Learn a program that you did not write

- xclawmpi
  - 3D advection equation
  - CLAWPACK is a software package designed to compute numerical solutions to hyperbolic partial differential equations using a wave propagation approach.
  - http://www.amath.washington.edu/~claw/
  - LeVeque, Randall (2002), Finite Volume Methods for Hyperbolic Problems, Cambridge University Press.

$$\frac{\partial q}{\partial t} + U\frac{\partial x}{\partial t} + V\frac{\partial y}{\partial t} + W\frac{\partial z}{\partial t} = 0$$




**Solution in x-y plane at t = 0.5, z = 0.425 using view(zSlice):**


TAU: ParaProf: Thread Statistics: n,c,t, 0,0,0 - /Users/tkaiser/claw
Name
Exclusive TIME
Inclusive TIME
CLAW3EZ_MPI_DRIVER [{claw3ez_mpi_driver.f} {5,7}-{577,9}] 1.438 4.507 1 21
MPI_Init() 1.035 1.035 1 0
OUT3 [{out3_mpi.f} {4,7}-{124,9}] 1.02 1.02 6 0
FLUX3 [{flux3.f} {4,7}-{603,9}] 0.412 0.626 89,... 379...
MPI_Allreduce() 0.17 0.17 50 0
STEP3 [{step3.f} {3,7}-{620,9}] 0.139 0.764 50 89,...
LIMITER [{limiter.f} {4,7}-{60,9}] 0.101 0.122 89,... 100...
RPTT3 [{rptt3adv.f}
RPT3 [{rpt3adv.f} {4
RPN3 [{rpn3adv.f} {
PHILIM [{philim.f} {4,
CLAW3EZ_MPI [{claw
SETAUX [{setaux.f} {
MPI_Send()
COMPUTE_U [{setau
COMPUTE_V [{setau
COMPUTE_W [{setau
MPI_Finalize()
BC3 [{bc3_mpi.f} {5,
COPYQ3 [{copyq3.f} {4,7}-
MPI_Cart_create()
CLAW3 [{claw3_mpi.f} {4,7
.TAU 1000 null timers over
.TAU null timer overhead
MPI_Bcast()
MPI_Barrier()
QINIT [{qinit.f} {6,8}-{45,9}
MPI_Cart_coords()
MPI_Irecv()
MPI_Wait()
TAU: ParaProf: /Users/tkaiser/claw
Metric: TIME
Value: Exclusive
Std. Dev.
Mean
node 0
node 1
node 2
MPI_Bcast()
Exclusive TIME: 1.387 seconds
Inclusive TIME: 1.387 seconds
Calls: 5.0
SubCalls: 0.0
TAU: ParaProf: Function Legend: claw/tkaiser/Users/
MPI_Cartdim_get()
MPI_Comm_rank()
MPI_Comm_size()
MPI_Finalize()
MPI_Init()
MPI_Irecv()
MPI_Recv()
MPI_Send()
MPI_Wait()
OUT3 [{out3_mpi.f} {4,7}-{124,9}]
PHILIM [{philim.f} {4,7}-{58,9}]
QINIT [{qinit.f} {6,8}-{45,9}]
RPN3 [{rpn3adv.f} {5,7}-{64,9}]

# Portland Group Profiler

- pgprof
- Easy to use
- Currently only with OpenMPI
- Documentation
  - http://geco.mines.edu/compilerDocs/pg/index.htm
  - http://inside.mines.edu/mio/doc/pg/index.htm

# pgprof - three build options

```
[tkaiser@mio stommel]$ mpif90 -Mprof=time  stf_03.f90 -o stf_03.tprof
[tkaiser@mio stommel]$ mpif90 -Mprof=lines stf_03.f90 -o stf_03.lprof
[tkaiser@mio stommel]$ mpif90 -Mprof=func  stf_03.f90 -o stf_03.fprof
[tkaiser@mio stommel]$ ls -l stf_03.*
-rw-rw-r-- 1 tkaiser tkaiser  17693 Mar  8 14:38 stf_03.f90
-rwxrwxr-x 1 tkaiser tkaiser 564317 Aug  2 16:15 stf_03.fprof
-rwxrwxr-x 1 tkaiser tkaiser 574301 Aug  2 16:14 stf_03.lprof (SLOW)
-rwxrwxr-x 1 tkaiser tkaiser 570288 Aug  2 16:14 stf_03.tprof
[tkaiser@mio stommel]$
```

# Normal run command

```
-bash-3.2$ mpiexec stf_03.fprof < st.in
...
...
-bash-3.2$ ls -lt pg*
-rw------- 1 tkaiser tkaiser 574 Aug  2 16:28 pgprof1.out
-rw------- 1 tkaiser tkaiser 575 Aug  2 16:28 pgprof2.out
-rw------- 1 tkaiser tkaiser 577 Aug  2 16:28 pgprof3.out
-rw------- 1 tkaiser tkaiser 579 Aug  2 16:28 pgprof4.out
-rw------- 1 tkaiser tkaiser 577 Aug  2 16:28 pgprof5.out
-rw------- 1 tkaiser tkaiser 579 Aug  2 16:28 pgprof6.out
-rw------- 1 tkaiser tkaiser 574 Aug  2 16:28 pgprof7.out
-rw------- 1 tkaiser tkaiser 731 Aug  2 16:28 pgprof.out
-bash-3.2$
```

Creates profile files

# pgprof viewer

- Forward X-window connection
  - ssh -Y mio.mines.edu
  - ssh -Y ra.mines.edu
- Go to your run directory and

# pgprof -exe stf_03.fprof

Name of your program

pgprof

File Settings Processes View Sort Search Help

Find: HotSpot: Time

pgprof.out

| | Function | Process | Time | Count | Messages | Messages recvd | Byte |
|---|---|---|---|---|---|---|---|
| i | do_jacobi | (Max) | 3.90544 = 57% | 75,000 = 25% | | | |
| i | stommel | (Max) | 1.75952 = 26% | 1 = 0% | 150,016 = 20% | 75,008 = 20% | |
| i | do_transfer | (Max) | 0.904142 = 13% | 75,001 = 25% | 600,008 = 80% | 300,004 = 80% | 36 |
| i | write_grid | (Max) | 0.236458 = 3% | 1 = 0% | 808 = 0% | 404 = 0% | |
| i | even | (Max) | 0.006063 = 0% | 150,002 = 49% | | | |
| i | force | (Max) | 0.000248 = 0% | 5,000 = 2% | | | |
| i | do_force | (Max) | 0.000175 = 0% | 1 = 0% | | | |
| i | unique | (Max) | 0.000023 = 0% | 1 = 0% | | | |
| i | bc | (Max) | 0.000019 = 0% | 1 = 0% | | | |

Sort By Time

Display Processes Thread Processes.Threads

| Processes | Time | Count | Messages | Messages r... | Bytes | Bytes sent | Bytes recvd |
|---|---|---|---|---|---|---|---|
| 2 (do_jacobi) | 3.90544 = 13% | 75,000 = 13% | | | | | |
| 6 (do_jacobi) | 3.86119 = 13% | 75,000 = 13% | | | | | |
| 5 (do_jacobi) | 3.85742 = 13% | 75,000 = 13% | | | | | |
| 0 (do_jacobi) | 3.85148 = 12% | 75,000 = 13% | | | | | |
| 1 (do_jacobi) | 3.85089 = 12% | 75,000 = 13% | | | | | |
| 4 (do_jacobi) | 3.84679 = 12% | 75,000 = 13% | | | | | |
| 3 (do_jacobi) | 3.84362 = 12% | 75,000 = 13% | | | | | |
| 7 (do_jacobi) | 3.81166 = 12% | 75,000 = 13% | | | | | |

Sort By Time

Parallelism Histogram Compiler Feedback System Configuration Accelerator Performance

Profiled: ./stf_03.fprof on Mon Aug 02 16:18:40 MDT 2010 for 6.297215 seconds with 8 processes

Profile: ./pgprof.out

# Summary

- Tools are relatively easy to use
- Profiling can change timing results
- Can help you understand "your" program
- Profiling can produce significant performance gains