Measurement of I/O with TAU

Kevin Huck, Allen Malony, Sameer Shende, Aurele Maheo,
Wyatt Spear, Robert Lim, Chad Wood, Jacob Lambert,
Srinivasan Ramesh, Mohammad Alaul Haque Monil

RAPIDS

khuck@cs.uoregon.edu
http://tau.uoregon.edu

U.S. DEPARTMENT OF ENERGY | Office of Science

UNIVERSITY OF OREGON

I/O Library Support in TAU

Measurement Support:

- Instrumentation
  - Source
  - Binary
- **Interposition libraries**
- **Wrappers**
- **Callbacks**
- **Sampling**

Library Support:

- **POSIX**
- **MPI I/O**
- **HDF5**
- **ADIOS**

http://tau.uoregon.edu/tau.tgz

POSIX measurement

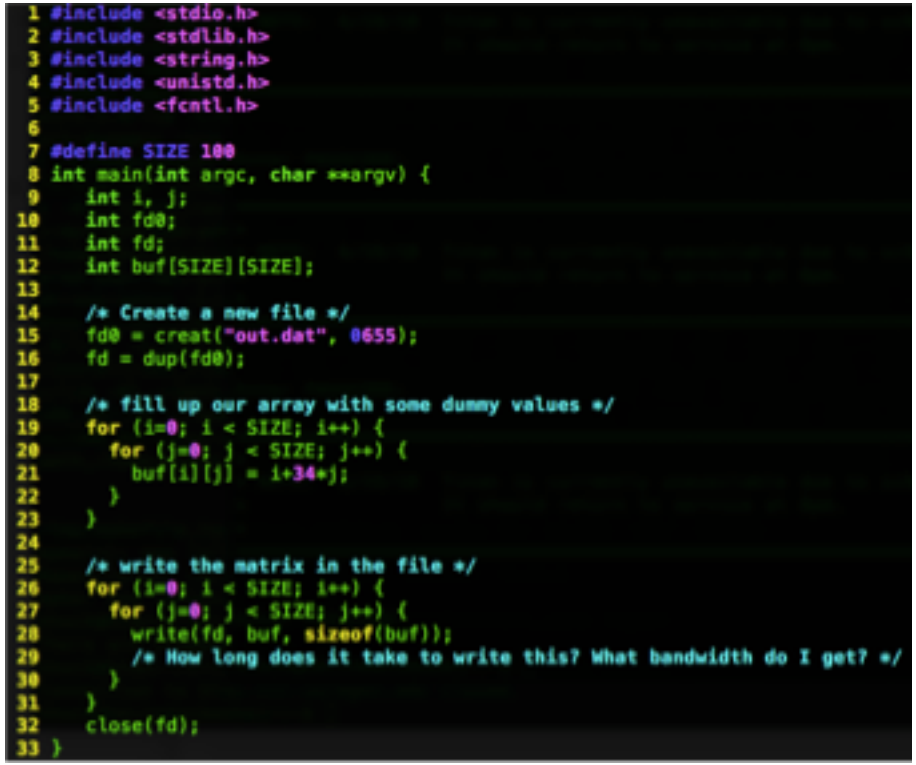
- Built-in support

  `$ ./configure -iowrapper`
- Wraps API calls related to files and sockets
- Linker based instrumentation (static executables)

  `-Wl,-wrap,fopen -Wl,-wrap,fclose ... -lTauPosixWrap`

  Wrapper library implements `__wrap_fopen`, calls `__real_fopen`
- Runtime preloading (dynamic executables)
  - `LD_PRELOAD=libTauPosixWrap.so` (or use `tau_exec -io`)
  - Wrapper library defines `fopen()`, loads *real* fopen using `dlsym()`
- "Characterizing I/O Performance Using the TAU Performance System", Shende et al., *PARCO,* 2011

POSIX static executable example



```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>

#define SIZE 100
int main(int argc, char **argv) {
    int i, j;
    int fd0;
    int fd;
    int buf[SIZE][SIZE];

    /* Create a new file */
    fd0 = creat("out.dat", 0655);
    fd = dup(fd0);

    /* fill up our array with some dummy values */
    for (i=0; i < SIZE; i++) {
      for (j=0; j < SIZE; j++) {
        buf[i][j] = i+34*j;
      }
    }

    /* write the matrix in the file */
    for (i=0; i < SIZE; i++) {
      for (j=0; j < SIZE; j++) {
        write(fd, buf, sizeof(buf));
        /* How long does it take to write this? What bandwidth do I get? */
      }
    }
    close(fd);
}
```

```
# configure TAU with POSIX IO wrapper support
$ ./configure  -iowrapper -bfd=download -unwind=download
# compile TAU
$ make -j install
# add TAU to the path
$ PATH=$PATH:`pwd`/x86_64/bin
# build and run the example
$ cd examples/iowrappers/posixio
$ make
$ ./foo
# view a profile summary
$ pprof
# profile visualizer
$ paraprof
```

Profile Views (pprof, paraprof)

Callpath view in paraprof:

(run with `TAU_CALLPATH=1`, `TAU_CALLPATH_DEPTH=3` or greater)

Name	Exclusive TIME	Inclusive TIME ▽	Calls
.TAU application	11,778	5,460,488	1
main [{/home/users/khuck/src/tau2/examples/iowrappers/posixio/foo.c} {9,0}]	3,049	5,448,710	1
close()	5,001,504	5,001,504	1
write()	265,370	265,370	10,000
creat()	178,771	178,771	1
dup()	16	16	1

Flat profile in paraprof:

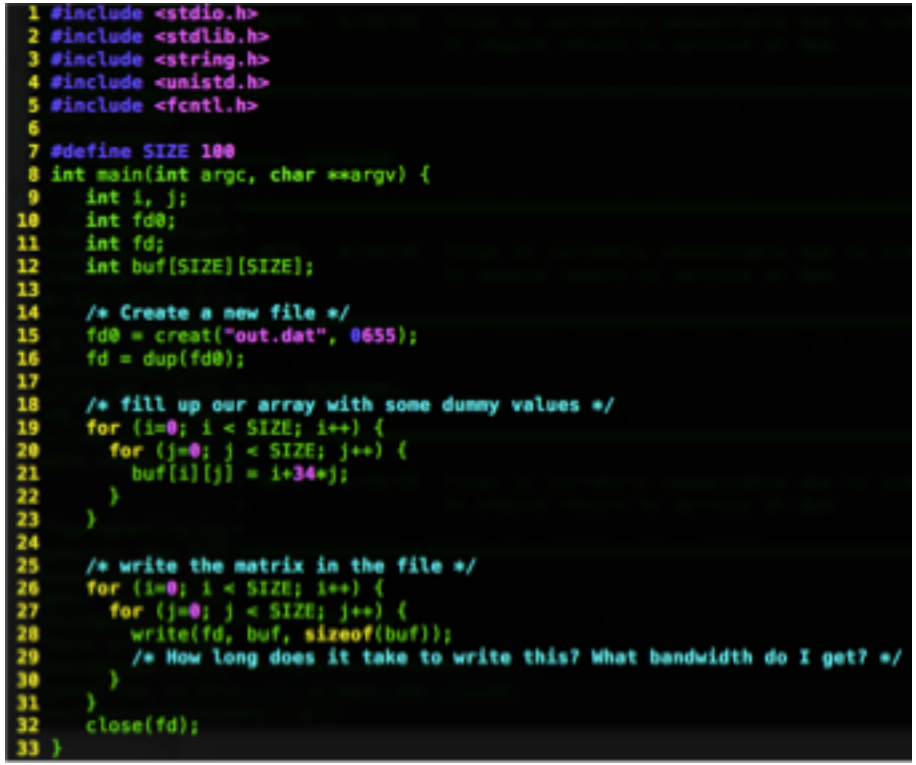
Metric: TIME
Value: Exclusive
Units: seconds

5.428	close()
0.277	write()
0.011	.TAU application
0.003	main [{/home/users/khuck/src/tau2/examples/iowrappers/posixio/foo.c} {9,0}]
0.003	creat()
6.0E-6	dup()

Text output from pprof:

```
/home/users/khuck/src/tau2/examples/iowrappers/posixio
[khuck@delphi posixio]$ pprof -s
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          18        5,720           1           1    5720989 .TAU application
 99.8           2        5,710           1       10003    5710263 main [{/home/users/khuck/src/tau2/examples/
iowrappers/posixio/foo.c} {9,0}]
 94.9       5,427        5,427           1           0    5427647 close()
  4.6         276          276       10000           0         28 write()
  0.0           2            2           1           0       2836 creat()
  0.0       0.006        0.006           1           0          6 dup()
---------------------------------------------------------------------------------------

USER EVENTS Profile :NODE 0, CONTEXT 0, THREAD 0
---------------------------------------------------------------------------------------
NumSamples   MaxValue   MinValue  MeanValue  Std. Dev.  Event Name
---------------------------------------------------------------------------------------
      1E+04      4E+04      4E+04      4E+04          0  Bytes Written
      1E+04      4E+04      4E+04      4E+04          0  Bytes Written : main [{/home/users/khuck/src/tau2/exam
ples/iowrappers/posixio/foo.c} {9,0}] => write()
      1E+04      4E+04      4E+04      4E+04          0  Bytes Written <file=out.dat>
      1E+04      4E+04      4E+04      4E+04          0  Bytes Written <file=out.dat> : main [{/home/users/khuc
k/src/tau2/examples/iowrappers/posixio/foo.c} {9,0}] => write()
      1E+04       2222      555.6       1596      277.2  Write Bandwidth (MB/s)
      1E+04       2222      555.6       1596      277.2  Write Bandwidth (MB/s) : main [{/home/users/khuck/src/
tau2/examples/iowrappers/posixio/foo.c} {9,0}] => write()
      1E+04       2222      555.6       1596      277.2  Write Bandwidth (MB/s) <file=out.dat>
      1E+04       2222      555.6       1596      277.2  Write Bandwidth (MB/s) <file=out.dat> : main [{/home/u
sers/khuck/src/tau2/examples/iowrappers/posixio/foo.c} {9,0}] => write()
---------------------------------------------------------------------------------------
[khuck@delphi posixio]$
```

POSIX dynamic executable example



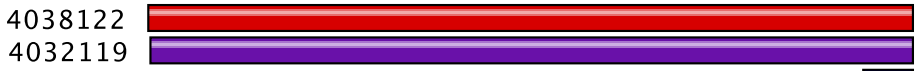
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>

#define SIZE 100
int main(int argc, char **argv) {
    int i, j;
    int fd0;
    int fd;
    int buf[SIZE][SIZE];

    /* Create a new file */
    fd0 = creat("out.dat", 0655);
    fd = dup(fd0);

    /* fill up our array with some dummy values */
    for (i=0; i < SIZE; i++) {
      for (j=0; j < SIZE; j++) {
        buf[i][j] = i+34*j;
      }
    }

    /* write the matrix in the file */
    for (i=0; i < SIZE; i++) {
      for (j=0; j < SIZE; j++) {
        write(fd, buf, sizeof(buf));
        /* How long does it take to write this? What bandwidth do I get? */
      }
    }
    close(fd);
}
```

```
# configure TAU with POSIX IO wrapper support
$ ./configure  -iowrapper -bfd=download -unwind=download
# compile TAU
$ make -j install
# build and run the example
$ cd examples/iowrappers/posixio
$ gcc foo.c -o foo
$ tau_exec -T serial -io -ebs ./foo
# view a profile summary
$ pprof -a
```

Profile Views (pprof, paraprof)

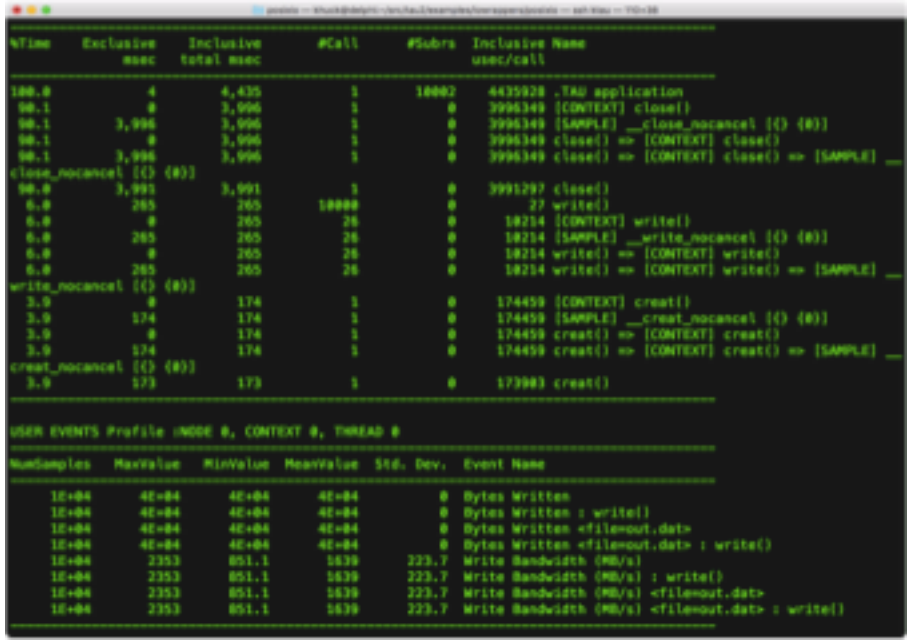
Metric: TIME
Value: Exclusive
Units: microseconds

Rank 0, thread 0:



4038122	[SAMPLE] __close_nocancel [{} {0}]
4032119	close()
266729	write()
266085	[SAMPLE] __write_nocancel [{} {0}]
163925	[SAMPLE] __creat_nocancel [{} {0}]
163248	creat()
5478	.TAU application
0	[CONTEXT] creat()
0	[CONTEXT] write()
0	[CONTEXT] close()

Text output from pprof:



```
%Time    Exclusive    Inclusive       #Call      #Subrs  Inclusive Name
              msec   total msec                          usec/call
---------------------------------------------------------------------------------------
100.0           4        4,435           1       10002    4435928 .TAU application
 90.1           0        3,996           1           0    3996349 [CONTEXT] close()
 90.1       3,996        3,996           1           0    3996349 [SAMPLE] __close_nocancel [{} {0}]
 90.1           0        3,996           1           0    3996349 close() => [CONTEXT] close()
 90.1       3,996        3,996           1           0    3996349 close() => [CONTEXT] close() => [SAMPLE] __
close_nocancel [{} {0}]
 90.0       3,991        3,991           1           0    3991297 close()
  6.0         265          265       10000           0         27 write()
  6.0           0          265          26           0      10214 [CONTEXT] write()
  6.0         265          265          26           0      10214 [SAMPLE] __write_nocancel [{} {0}]
  6.0           0          265          26           0      10214 write() => [CONTEXT] write()
  6.0         265          265          26           0      10214 write() => [CONTEXT] write() => [SAMPLE] __
write_nocancel [{} {0}]
  3.9           0          174           1           0     174459 [CONTEXT] creat()
  3.9         174          174           1           0     174459 [SAMPLE] __creat_nocancel [{} {0}]
  3.9           0          174           1           0     174459 creat() => [CONTEXT] creat()
  3.9         174          174           1           0     174459 creat() => [CONTEXT] creat() => [SAMPLE] __
creat_nocancel [{} {0}]
  3.9         173          173           1           0     173003 creat()
---------------------------------------------------------------------------------------

USER EVENTS Profile :NODE 0, CONTEXT 0, THREAD 0
---------------------------------------------------------------------------------------
NumSamples   MaxValue   MinValue  MeanValue  Std. Dev.  Event Name
---------------------------------------------------------------------------------------
      1E+04      4E+04      4E+04      4E+04          0  Bytes Written
      1E+04      4E+04      4E+04      4E+04          0  Bytes Written : write()
      1E+04      4E+04      4E+04      4E+04          0  Bytes Written <filename.dat>
      1E+04      4E+04      4E+04      4E+04          0  Bytes Written <filename.dat> : write()
      1E+04       2353      851.1       1639      223.7  Write Bandwidth (MB/s)
      1E+04       2353      851.1       1639      223.7  Write Bandwidth (MB/s) : write()
      1E+04       2353      851.1       1639      223.7  Write Bandwidth (MB/s) <filename.dat>
      1E+04       2353      851.1       1639      223.7  Write Bandwidth (MB/s) <filename.dat> : write()
---------------------------------------------------------------------------------------
```

Callpath view:

Name	Exclusive TIME	Inclusive TIME ▽	Calls	Child Calls
.TAU application	5,478	4,467,574	1	10,002
close()	4,032,119	4,032,119	1	0
[CONTEXT] close()	0	4,038,122	1	0
[SAMPLE] __close_nocancel [{} {0}]	4,038,122	4,038,122	1	0
write()	266,729	266,729	10,000	0
[CONTEXT] write()	0	266,085	24	0
[SAMPLE] __write_nocancel [{} {0}]	266,085	266,085	24	0
creat()	163,248	163,248	1	0
[CONTEXT] creat()	0	163,925	1	0
[SAMPLE] __creat_nocancel [{} {0}]	163,925	163,925	1	0

POSIX counters and metadata, too

Broken down by filename

Name △	Total	NumSamples	MaxValue	MinValue	MeanValue	Std. Dev.
Bytes Written	400,000,000	10,000	40,000	40,000	40,000	0
Bytes Written <file=out.dat>	400,000,000	10,000	40,000	40,000	40,000	0
Write Bandwidth (MB/s)		10,000	2,222.222	816.327	1,660.024	208.113
Write Bandwidth (MB/s) <file=out.dat>		10,000	2,222.222	816.327	1,660.024	208.113
write()						
Write Bandwidth (MB/s) <file=out.dat>		10,000	2,222.222	816.327	1,660.024	208.113
Write Bandwidth (MB/s)		10,000	2,222.222	816.327	1,660.024	208.113
Bytes Written <file=out.dat>	400,000,000	10,000	40,000	40,000	40,000	0
Bytes Written	400,000,000	10,000	40,000	40,000	40,000	0

...and by calling context

Metadata for file

Timestamp	1529437165126767
UTC Time	2018-06-19T19:39:25Z
pid	107899
posix open[0]	{"event-type":"output","name":".TAU application","time":"1529437165127306","node-id":"0","thread-id":"0","filename":"out.dat"}
tid	107899
username	khuck

MPI I/O Support

- Built-in support (adding -pthread recommended)

```
$ ./configure -mpi -pthread
```

- Replaces MPI API weak symbols with instrumented versions

`MPI_Send(...) { return PMPI_Send(...); }` is the default implementation

- Linker based instrumentation (static executables)

```
tau_cc.sh -o foo *.o libstuff.a
```

- Runtime preloading (dynamic executables)

```
tau_exec -T mpi,pthread ./my_program
```

MPI dynamic executable example

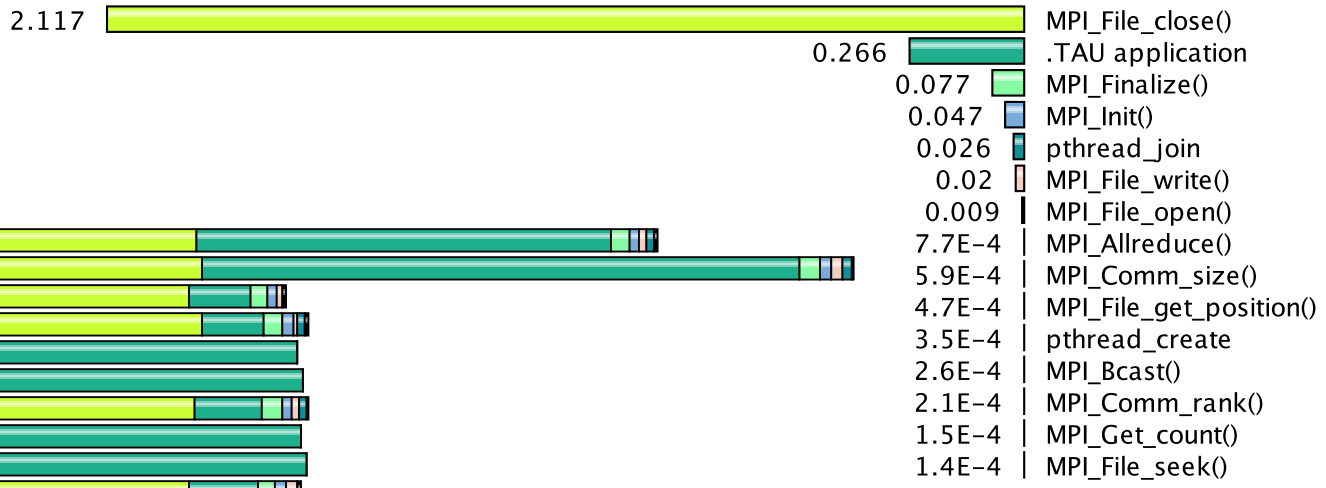
- Example code from https://www.hpc.ntnu.no/display/hpc/Writing+to+MPI+Files

```
# configure TAU with MPI support
$ ./configure  -iowrapper -bfd=download -
unwind=download -mpi -pthread
# compile TAU
$ make -j install
# build and run the example (new example)
$ cd examples/mpi_io
$ mpicc -DDEBUG -g -O3 -o mpiio mpiio.c
$ mpirun -np 4 tau_exec -T mpi,pthread ./mpiio -f
foo -l 10
```

Profile Views

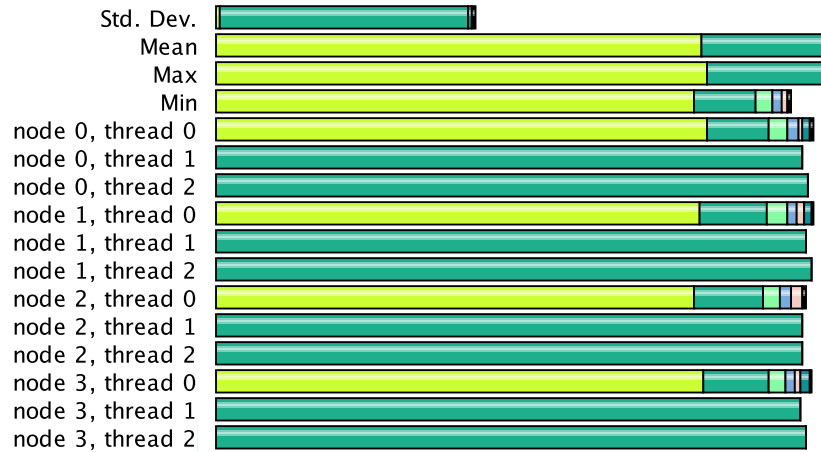
Metric: TIME
Value: Exclusive
Units: seconds

Rank 0, thread 0:



All ranks and threads:

Metric: TIME
Value: Exclusive



MPI and POSIX?

- Just add -io to the tau_exec options
- Can also add -ebs for sampling support

Name	Exclusive TIME	Inclusive TIME ▽	Calls	Child Calls
.TAU application	464,204	2,824,776	1	15
MPI_File_close()	23,043	2,151,751	1	2
close()	2,128,708	2,128,708	2	0
MPI_Finalize()	38,080	98,684	1	41
MPI_Init()	49,709	64,757	1	302
MPI_File_write()	316	22,316	1	2
writev()	21,917	21,917	1	0
lseek()	83	83	1	0
MPI_File_open()	6,493	22,092	1	7
open()	14,817	14,817	4	0
close()	590	590	2	0
write()	192	192	1	0
MPI_Allreduce()	227	227	1	0
MPI_Bcast()	178	178	3	0
MPI_File_get_positio	123	123	2	0
MPI_Comm_size()	115	115	1	0
MPI_Comm_rank()	114	114	1	0
MPI_File_seek()	113	113	1	0
MPI_Get_count()	102	102	1	0

MPI+POSIX

Name	Exclusive TIME	Inclusive TIME ▽	Calls	Child Calls
.TAU application	337,412	7,811,004	1	15
MPI_File_close()	20,324	4,234,121	1	2
close()	4,213,797	4,213,797	2	0
[CONTEXT] close()	0	4,214,726	1	0
[SAMPLE] __close_nocancel [{} {0}]	4,214,726	4,214,726	1	0
[CONTEXT] MPI_File_close()	0	19,850	2	0
[SAMPLE] ompi_mtl_psm2_progress [{/storage/packages	10,000	10,000	1	0
[SAMPLE] UNRESOLVED /usr/lib64/libpsm2.so.2.1	9,850	9,850	1	0
MPI_File_open()	3,013,888	3,021,439	1	7
[CONTEXT] MPI_File_open()	0	3,025,704	1	0
[SAMPLE] __libc_fcntl [{} {0}]	3,025,704	3,025,704	1	0
open()	6,285	6,285	4	0
close()	945	945	2	0
write()	321	321	1	0
[CONTEXT] .TAU application	0	334,270	33	0
[SAMPLE] __random [{} {0}]	284,271	284,271	28	0
[SAMPLE] UNRESOLVED /storage/users/khuck/src/tau2/exa	30,000	30,000	3	0
[SAMPLE] __random_r [{} {0}]	19,999	19,999	2	0
MPI_Finalize()	32,494	144,860	1	41
MPI_Init()	34,862	48,994	1	302
MPI_File_write()	561	22,469	1	2
MPI_Comm_size()	472	472	1	0
MPI_Allreduce()	307	307	1	0
MPI_File_seek()	238	238	1	0
MPI_File_get_position()	236	236	2	0
MPI_Bcast()	172	172	3	0
MPI_Get_count()	163	163	1	0
MPI_Comm_rank()	121	121	1	0

MPI+POSIX+sampling

Counters, too...

Aggregates (broken down by filename):

With calling context:

Name	Total	NumSamples	MaxValue	MinValue	MeanValue	Std. Dev.
Bytes Read	14,059	47	8,192	1	299.128	1,290.66
Bytes Read <file=/packages/openmpi/2.1.3_gcc-7.3/etc/openmpi-mca-params.c	2,818	1	2,818	2,818	2,818	0
Bytes Read <file=/packages/openmpi/2.1.3_gcc-7.3/share/openmpi/mca-btl-op	10,920	2	8,192	2,728	5,460	2,732
Bytes Read <file=/sys/class/infiniband/hfi1_0/device/local_cpus>	6	6	1	1	1	0
Bytes Read <file=/sys/class/infiniband/hfi1_0/node_guid>	20	1	20	20	20	0
Bytes Read <file=/sys/class/infiniband/hfi1_0/node_type>	6	1	6	6	6	0
Bytes Read <file=/sys/class/infiniband/hfi1_0/ports/1/gids/0>	40	1	40	40	40	0
Bytes Read <file=/sys/class/infiniband_verbs/abi_version>	2	1	2	2	2	0
Bytes Read <file=/sys/class/infiniband_verbs/uverbs0/abi_version>	6	3	2	2	2	0
Bytes Read <file=/sys/class/infiniband_verbs/uverbs0/device/device>	70	10	7	7	7	0
Bytes Read <file=/sys/class/infiniband_verbs/uverbs0/device/modalias>	54	1	54	54	54	0
Bytes Read <file=/sys/class/infiniband_verbs/uverbs0/device/vendor>	70	10	7	7	7	0
Bytes Read <file=/sys/class/infiniband_verbs/uverbs0/ibdev>	35	5	7	7	7	0
Bytes Read <file=/sys/class/misc/rdma_cm/abi_version>	2	1	2	2	2	0
Bytes Read <file=/sys/devices/system/cpu/possible>	2	2	1	1	1	0
Bytes Read <file=/tmp/openmpi-sessions-14978@delphi_0/57016/pmix-12680	8	2	4	4	4	0
Bytes Written	41,954,321	107	41,943,040	4	392,096.458	4,035,784.441
Bytes Written <file=/dev/infiniband/rdma_cm>	416	11	144	24	37.818	33.785
Bytes Written <file=/dev/infiniband/uverbs0>	2,152	45	120	12	47.822	41.096
Bytes Written <file=/tmp/OMPIO_foo_-558366719_.sm>	40	1	40	40	40	0
Bytes Written <file=/tmp/openmpi-sessions-14978@delphi_0/57016/1/shared_m	4,136	1	4,136	4,136	4,136	0
Bytes Written <file=/tmp/openmpi-sessions-14978@delphi_0/57016/1/shared_m	4,144	2	4,136	8	2,072	2,064
Bytes Written <file=/tmp/openmpi-sessions-14978@delphi_0/57016/pmix-1268	37	1	37	37	37	0
Bytes Written <file=foo>	41,943,040	1	41,943,040	41,943,040	41,943,040	0
Bytes Written <file=pipe>	4	1	4	4	4	0
Bytes Written <file=unknown>	352	44	8	8	8	0
MPI-IO Bytes Written	41,943,040	1	41,943,040	41,943,040	41,943,040	0
MPI-IO Write Bandwidth (MB/s)		1	1,884.911	1,884.911	1,884.911	0
MPI_File_open()						
write()						
Bytes Written	40	1	40	40	40	0
Bytes Written <file=/tmp/OMPIO_foo_-558366719_.sm>	40	1	40	40	40	0
Write Bandwidth (MB/s)		1	1.818	1.818	1.818	0
Write Bandwidth (MB/s) <file=/tmp/OMPIO_foo_-558366719_.sm>		1	1.818	1.818	1.818	0
MPI_File_write()						

HDF5 support

- Not built-in, but TAU includes an example using the `tau_gen_wrapper` utility
- Uses PDT (http://tau.uoregon.edu/pdt.tgz) to parse a library header, and provides a wrapper library similar to what is available for the POSIX library wrapper
- Linker based instrumentation (static executables)

  i.e. **`tau_cc.sh -o foo *.o -tau_options='-optTauWrapFile=headers/hdf5_wrapper/link_options.tau -optTrackIO' libotherstuff.a`**
- Runtime preloading (dynamic executables)

  i.e. **`tau_exec -T mpi,pthread -loadlib=./libhdf5_wrap.so ./my_program`**

HDF5 tau_gen_wrap example

```
# configure TAU with PDT support
$ ./configure  -iowrapper -bfd=download -unwind=download -mpi -pthread -pdt=/path/to/pdt
# compile TAU
$ make -j install
# build and run the wrapper
$ cd examples/iowrappers/hdf5_wrap
$ mkdir headers ; cd headers ; ln -s /path/to/hdf5/include/* .
$ tau_gen_wrapper hdf5.h /path/to/hdf5/lib/libhdf5.a -f ../select.tau ; cd ..
# build and run the example
$ tau_cc.sh -tau_options='-optTauWrapFile=headers/hdf5_wrapper/link_options.tau -optTrackIO ' -c hyperslab_by_row.c -I/path/to/hdf5/include
$ tau_cc.sh -tau_options='-optTauWrapFile=headers/hdf5_wrapper/link_options.tau -optTrackIO ' -o hyperslab_by_row hyperslab_by_row.o -L/path/to/hdf5/lib -lhdf5_hl -lhdf5 -lz -lsz -ldl -lm -Wl,-rpath,/path/to/hdf5/lib
$ mpirun -np 4 ./hyperslab_by_row
```

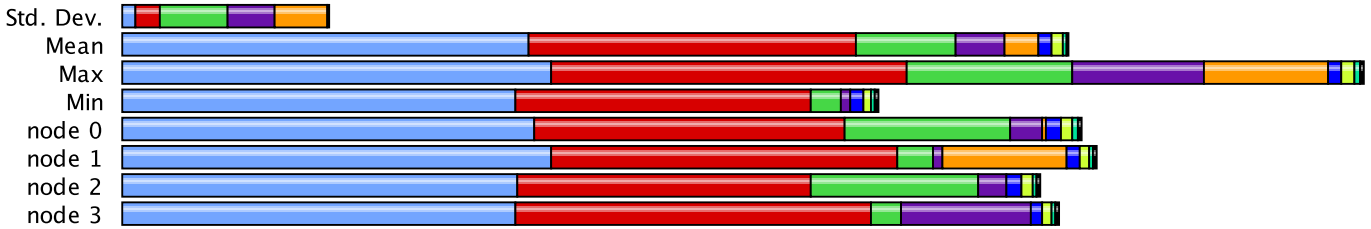
Profile Views

Metric: TIME
Value: Exclusive
Units: microseconds

Rank 0, thread 0:

All ranks and threads:

Metric: TIME
Value: Exclusive



HDF5 tau_wrap example

- Can use a similar process to generate a libhdf5_wrap.so wrapper that can be preloaded with tau_exec:

```
$ tau_gen_wrapper -r ...

$ tau_exec -T mpi,pthread -io –loadlib=./libhdf5_wrap.so
./hyperslab_by_row
```

ADIOS support

- Built-in, uses callback API provided by ADIOS 1.13
- Using the callback API, measurement tools can register with ADIOS during initialization, and be notified when ADIOS events happen (even asynchronous ones)
  - Tool implements `adiost_tool()` function that returns a pointer to an initialization function to register for event notification
- ADIOS is (by default) built as static library on many systems, so only way to register tools is by adding linker options before the ADIOS libraries
- Dynamic linking is easier on other systems, of course (can just use tau_exec)

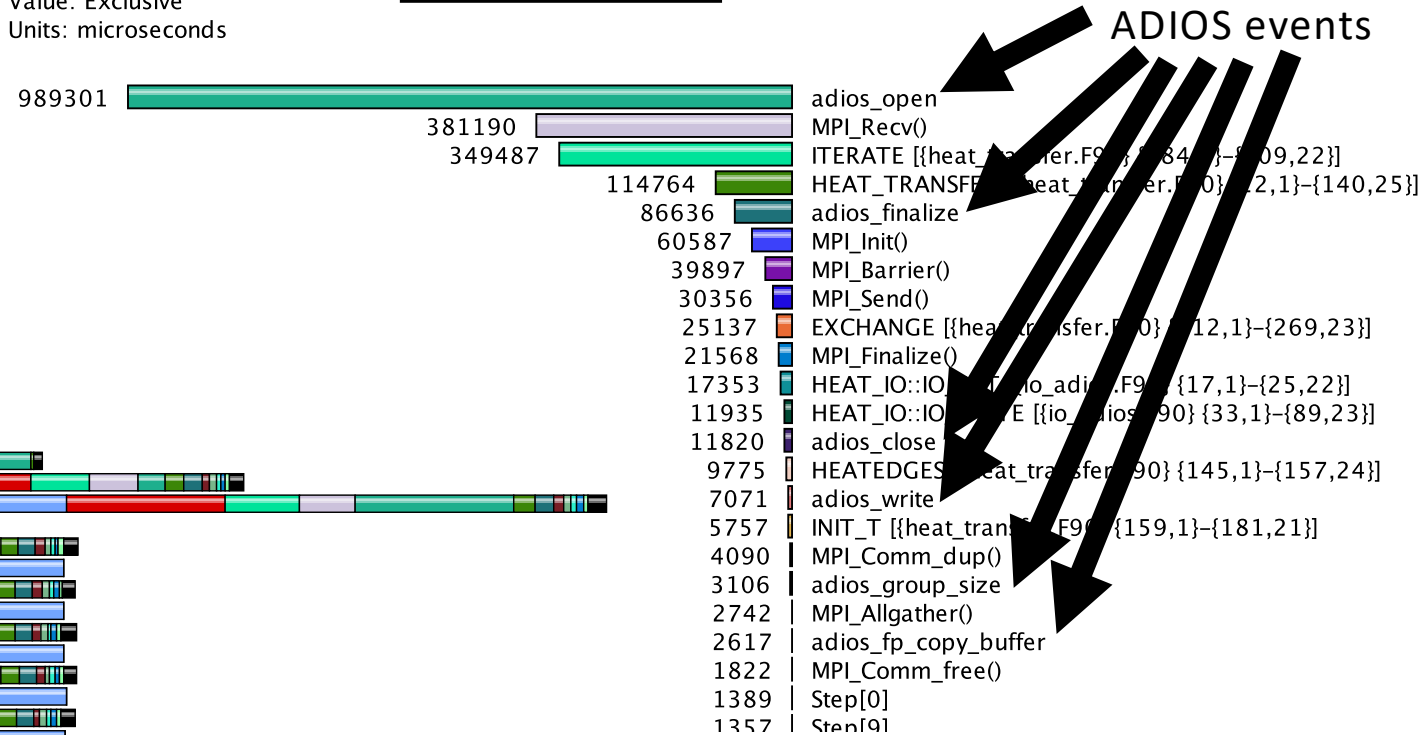
ADIOS example

```
# configure TAU with PDT support
$ ./configure  -iowrapper -bfd=download -unwind=download
-mpi -pthread -pdt=/path/to/pdt -adios=/path/to/adios
# compile TAU
$ make -j install
# build and run the example
$ cd src/Example-Heat_Transfer
# modify Makefile to compile with tau_f90.sh instead of
mpif90:
# FC=tau_f90.sh -optTauSelectFile=select.tau –optShared
$ make
$ mpirun -np 4 ./heat_transfer_adios
```

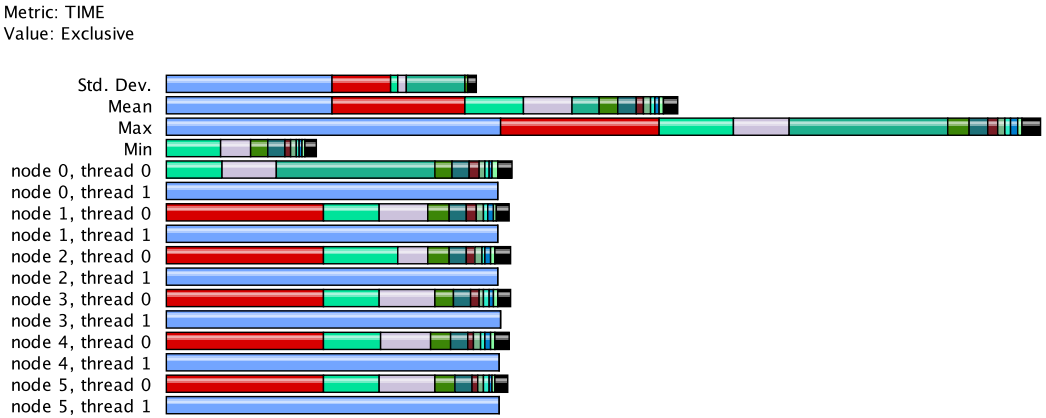
Profile Views

Metric: TIME
Value: Exclusive
Units: microseconds

Rank 0, thread 0:



All ranks and threads:



Profile Views, cont.

Callgraph:

Name	Exclusive TIME	Inclusive TIME ▽	Calls	Child C...
.TAU application	185	2,165,734	1	1
HEAT_TRANSFER [{heat_transfer.F90} {22,1}-{140,25}]	109,402	2,165,549	1	26
Step[0]	1,585	1,051,233	1	1,501
HEAT_IO::IO_WRITE [{io_adios.F90} {33,1}-{89,23}]	1,914	1,001,717	1	18
adios_open	987,126	992,428	1	12
MPI_Barrier()	3,444	3,444	1	0
MPI_Comm_dup()	487	487	2	0
MPI_Gather()	423	423	1	0
adios_define_attribute	241	241	3	0
MPI_Bcast()	239	239	3	0
MPI_Comm_size()	235	235	1	0
MPI_Comm_rank()	233	233	1	0
adios_close	4,585	5,451	1	3
adios_write	870	870	13	0
MPI_Barrier()	822	822	2	0
adios_group_size	232	232	1	0

Counters:

Name △	Total	NumSamples	MaxValue	MinValue	MeanValue	Std. Dev.
ADIOS data size	960,440	10	96,044	96,044	96,044	0
HEAT_TRANSFER [{heat_transfer.F90} {22,1}-{140,25}]						
MPI_Finalize()						
Peak Memory Usage Resident Set Size (VmHWM) (KB)	410,524	12	44,088	16,600	34,210.333	7,661.478
Step[0]						
HEAT_IO::IO_WRITE [{io_adios.F90} {33,1}-{89,23}]						
adios_group_size						
ADIOS data size	96,044	1	96,044	96,044	96,044	0
Step[1]						

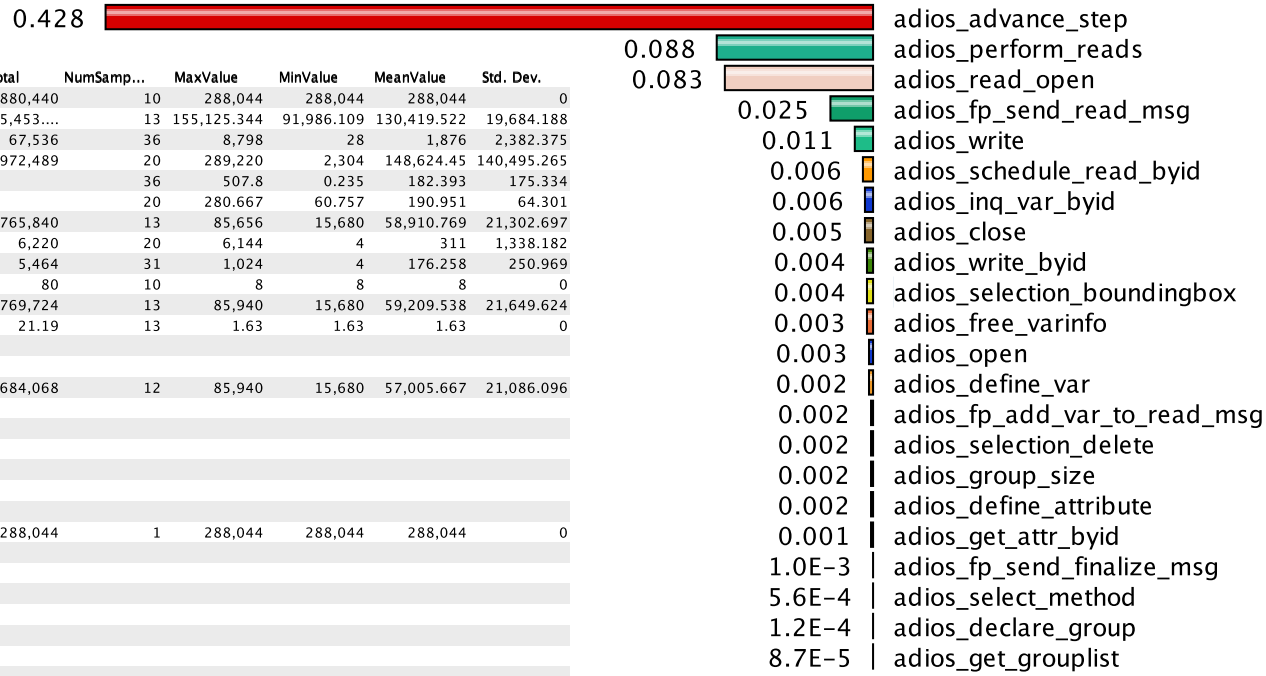
Reader Profile, Counters

Metric: TIME
Value: Exclusive
Units: seconds

Rank 0, thread 0 (only ADIOS events):

Counters:

Name	Total	NumSamp...	MaxValue	MinValue	MeanValue	Std. Dev.
ADIOS data size	2,880,440	10	288,044	288,044	288,044	0
Heap Memory Used (KB)	1,695,453....	13	155,125.344	91,986.109	130,419.522	19,684.188
MPI-IO Bytes Read	67,536	36	8,798	28	1,876	2,382.375
MPI-IO Bytes Written	2,972,489	20	289,220	2,304	148,624.45	140,495.265
MPI-IO Read Bandwidth (MB/s)		36	507.8	0.235	182.393	175.334
MPI-IO Write Bandwidth (MB/s)		20	280.667	60.757	190.951	64.301
Memory Footprint (VmRSS) (KB)	765,840	13	85,656	15,680	58,910.769	21,302.697
Message size for broadcast	6,220	20	6,144	4	311	1,338.182
Message size for gather	5,464	31	1,024	4	176.258	250.969
Message size for scatter	80	10	8	8	8	0
Peak Memory Usage Resident Set Size (VmHWM) (KB)	769,724	13	85,940	15,680	59,209.538	21,649.624
System load	21.19	13	1.63	1.63	1.63	0
int main(int, char **) C [{stage_write.c} {424,1}-{549,1}]						
MPI_Finalize()						
Peak Memory Usage Resident Set Size (VmHWM) (KB)	684,068	12	85,940	15,680	57,005.667	21,086.096
Step[0]						
Step[1]						
Step[2]						
int read_write(int) C [{stage_write.c} {382,1}-{421,1}]						
adios_close						
adios_group_size						
ADIOS data size	288,044	1	288,044	288,044	288,044	0
adios_open						
Step[3]						
Step[4]						
Step[5]						
Step[6]						
Step[7]						
Step[8]						
Step[9]						



UNIVERSITY OF OREGON

RAPIDS Data Management plans

- More ADIOS integration
  - Save TAU data as ADIOS format (and/or with science data)
  - Update to ADIOS 2 support (and other improvements)
- NetCDF support
  - Wrapper library like HDF5?
- Co-location with Darshan
  - Currently have to unload darshan module before using TAU
  - Could benefit from shared measurement
  - MPIT solution?
- Usability improvements (library wrapper)