Hewlett Packard
Enterprise

APPLICATION PROFILING

Trey White

February 17, 2023

with help from Steve Abbott, James Custer,
Ted Packwood, John Rodgers, and Mark Stock

Hewlett Packard
Enterprise

This presentation has been modified from its original version. It has been formatted to fit this screen and edited for content.

USING HPE PERFORMANCE ANALYSIS TOOLS ON CRUSHER AND FRONTIER

Trey White

January 20, 2023

with help from Steve Abbott, James Custer, Ted Packwood, John Rodgers, and Mark Stock

Hewlett Packard
Enterprise

PART 2: PERFORMANCE PROFILING FOR FRONTIER II

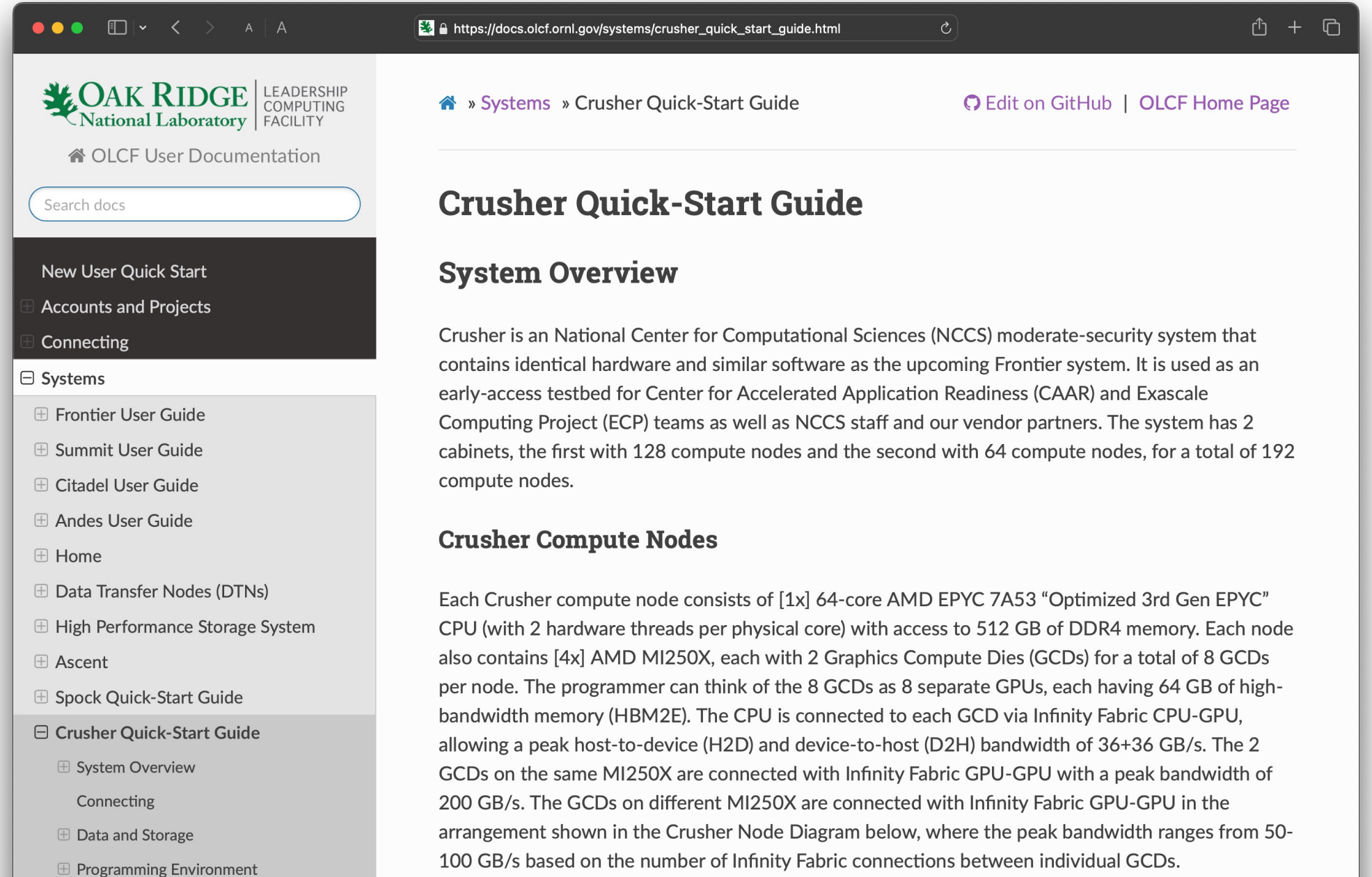
The PATH of ROCM

Trey White

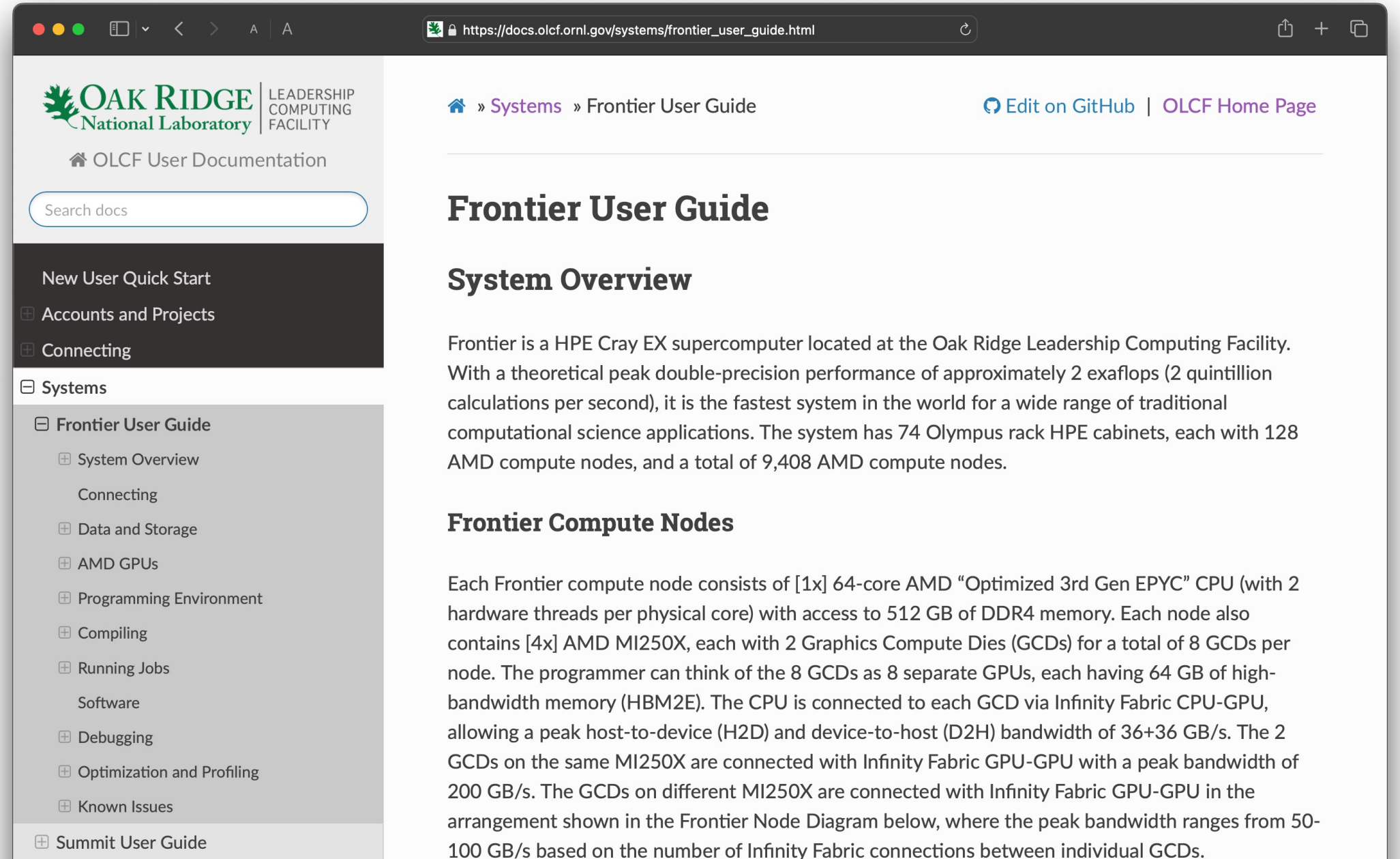
May 6, 2022

https://youtu.be/Bw3CEo2eOUA

THANKS TO OLCF FOR USE OF CRUSHER!



LOOKING FORWARD TO FRONTIER!



WHY USE HPE PERFORMANCE ANALYSIS TOOLS?

- Installed and work on Crusher and Frontier today
- Designed for parallel applications: multiple threads, cores, GPUs, nodes
- Can see all: CPUs, GPUs, MPI, memory, I/O, power
- Relatively low impact on runtime
- Relatively small extra output for default summary tracing
- Relatively efficient extra output for full tracing
- Gui runs locally on your computer to visualize remote performance data on Crusher or Frontier

CAVEATS

- Often don't support the newest Rocm
- Require instrumentation, preferably recompiling and relinking
  - Unlike, say, *rocprof*
- No profiling inside each AMD GPU kernel*
- Currently have very limited support for AMD GPU performance counters*
- Gui is retro*
- Timeline visualization for full traces temporarily broken*
- Local gui only supports MacOS and Windows*

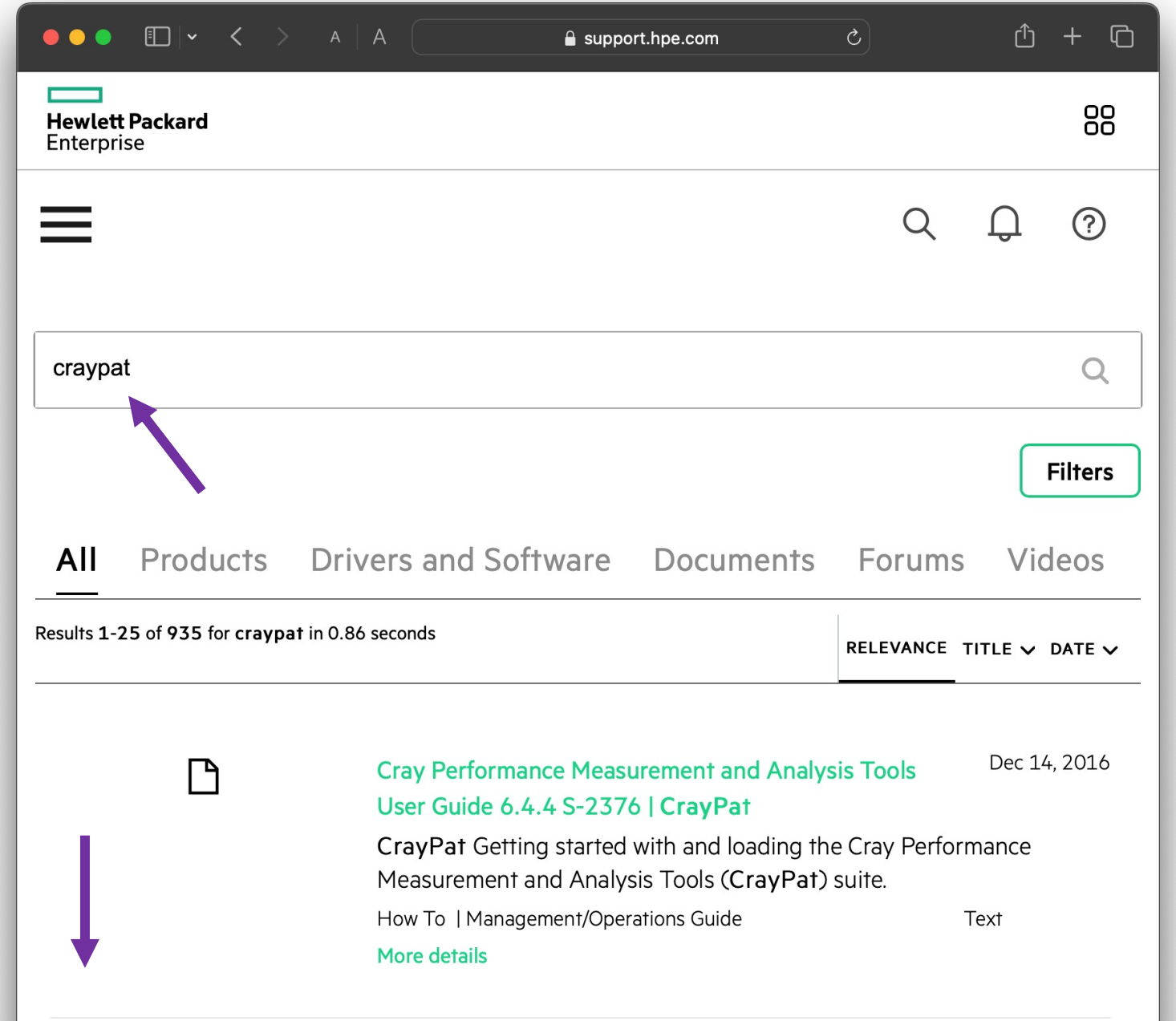
** But we're working on it*

AND WHAT'S IT CALLED, AGAIN?

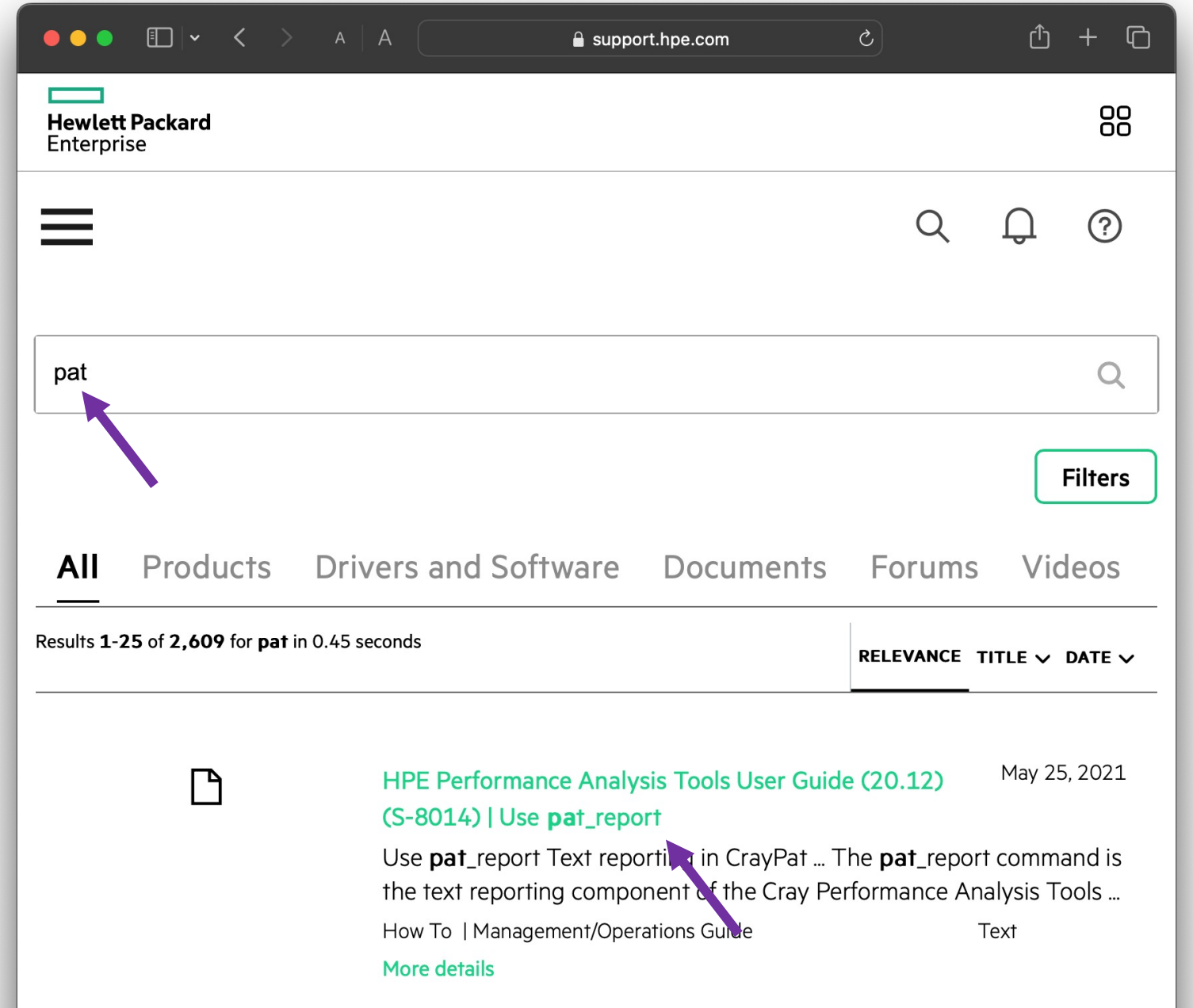
```
man cray_pat
man CrayPat
man intro_craypat
man intro_perftools
man pat
man perftools
```

```
INTRO_CRAYPAT(1)          General Commands Manual          INTRO_CRAYPAT(1)

NAME
       intro_craypat - Introduces CrayPat, the Cray Performance
       Analysis Tool

DESCRIPTION
       CrayPat is an optional performance analysis tool used to
       evaluate program behavior on Cray supercomputer systems.
       CrayPat consists of the following major components.
```

AND WHAT'S IT CALLED, AGAIN?



AND WHAT'S IT CALLED, AGAIN?



BUILD AND RUN → INSTRUMENT AND PROFILE

BUILD FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load rocm
ftn -fopenmp ...
```

BUILD FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load rocm
ftn -fopenmp ...
```

Offload to MI250X

BUILD FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load rocm
ftn -fopenmp ...
```

Set Rocm paths

BUILD FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load rocm
ftn -fopenmp ...
```

Compile for offload

Link to MPI by default

RUN FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load rocm
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>
```

RUN FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load rocm
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>
```

Match build modules

RUN FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load rocm
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>
```

Use module goodies to run

RUN FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load rocm
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>
```

Use GPU buffers for MPI

RUN FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load rocm
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>
```

Align MPI tasks, GPUs, and network interfaces

RUN FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load rocm
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>
```

Slurm reserves every 8th core for OS,
starting with the 1st core (#0),
leaving 7 per MPI task

RUN FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load rocm
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>
```

Make sure to match if using host OpenMP threads

INSTRUMENT FORTRAN WITH OPENMP OFFLOAD?

MATCH ROCM VERSIONS

```
login2:~$ module show cce |& grep -i rocm
- AMD ROCm 5.2 support
- AMD ROCm 5.0 and 5.1 support
  AMD GPU offloading support requires at least ROCm 5.0
login2:~$ module show perftools-base |& grep -i rocm
  o Support for AMD ROCm 5.1
  o As of ROCm version 4.5, SLES 15 support is limited to SP3.
login2:~$ module -t avail perftools-base
/opt/cray/pe/lmod/modulefiles/core:
perftools-base/21.12.0
perftools-base/22.04.0
perftools-base/22.05.0
perftools-base/22.06.0
perftools-base/22.09.0
perftools-base/22.12.0
login2:~$ module show perftools-base/22.12.0 |& grep -i rocm
  o Support for AMD ROCm 5.3.0
  o Support for OpenMP device tracing when using AMD ROCm 5.3.0's OpenMP runtime
login2:~$ module show perftools-base/22.09.0 |& grep -i rocm
  o Support for AMD ROCm 5.2.0
    including Intel icx/ifx, AMD aocc/rocm, and CCE compilers (see the OpenMP section of
  o ROCm 5.2.0 introduced API changes that prevent backwards compatibility with previous
```

MATCH ROCM VERSIONS

See what cce wants

```
login2:~$ module show cce |& grep -i rocm
- AMD ROCm 5.2 support
- AMD ROCm 5.1 and 5.1 support
  AMD GPU offloading support requires at least ROCm 5.0
login2:~$ module show perftools-base |& grep -i rocm
  o Support for AMD ROCm 5.1
  o As of ROCm version 4.5, SLES 15 support is limited to SP3.
login2:~$ module -t avail perftools-base
/opt/cray/pe/lmod/modulefiles/core:
perftools-base/21.12.0
perftools-base/22.04.0
perftools-base/22.05.0
perftools-base/22.06.0
perftools-base/22.09.0
perftools-base/22.12.0
login2:~$ module show perftools-base/22.12.0 |& grep -i rocm
  o Support for AMD ROCm 5.3.0
  o Support for OpenMP device tracing when using AMD ROCm 5.3.0's OpenMP runtime
login2:~$ module show perftools-base/22.09.0 |& grep -i rocm
  o Support for AMD ROCm 5.2.0
    including Intel icx/ifx, AMD aocc/rocm, and CCE compilers (see the OpenMP section of
  o ROCm 5.2.0 introduced API changes that prevent backwards compatibility with previous
```

MATCH ROCM VERSIONS

```
login2:~$ module show cce |& grep -i rocm
- AMD ROCm 5.2 support
- AMD ROCm 5.0 and 5.1 support
  AMD GPU offloading support requires at least ROCm 5.0
login2:~$ module show perftools-base |& grep -i rocm
  o Support for AMD ROCm 5.1
  o As of ROCm version 4.5, SLES 15 support is limited to SP3.
login2:~$ module -t avail perftools-base
/opt/cray/pe/lmod/modulefiles/core:
perftools-base/21.12.0
perftools-base/22.04.0
perftools-base/22.05.0
perftools-base/22.06.0
perftools-base/22.09.0
perftools-base/22.12.0
login2:~$ module show perftools-base/22.12.0 |& grep -i rocm
  o Support for AMD ROCm 5.3.0
  o Support for OpenMP device tracing when using AMD ROCm 5.3.0's OpenMP runtime
login2:~$ module show perftools-base/22.09.0 |& grep -i rocm
  o Support for AMD ROCm 5.2.0
    including Intel icx/ifx, AMD aocc/rocm, and CCE compilers (see the OpenMP section of
  o ROCm 5.2.0 introduced API changes that prevent backwards compatibility with previous
```

Does perftools-base agree?

Maybe, but anything newer?

MATCH ROCM VERSIONS

```
login2:~$ module show cce |& grep -i rocm
- AMD ROCm 5.2 support
- AMD ROCm 5.0 and 5.1 support
  AMD GPU offloading support requires at least ROCm 5.0
login2:~$ module show perftools-base |& grep -i rocm
  o Support for AMD ROCm 5.1
  o As of ROCm version 4.5, SLES 15 support is limited to SP3.
login2:~$ module -t avail perftools-base
/opt/cray/pe/lmod/modulefiles/core:
perftools-base/21.12.0
perftools-base/22.04.0
perftools-base/22.05.0
perftools-base/22.06.0
perftools-base/22.09.0
perftools-base/22.12.0
login2:~$ module show perftools-base/22.12.0 |& grep -i rocm
  o Support for AMD ROCm 5.3.0
  o Support for OpenMP device tracing when using AMD ROCm 5.3.0's OpenMP runtime
login2:~$ module show perftools-base/22.09.0 |& grep -i rocm
  o Support for AMD ROCm 5.2.0
    including Intel icx/ifx, AMD aocc/rocm, and CCE compilers (see the OpenMP section of
  o ROCm 5.2.0 introduced API changes that prevent backwards compatibility with previous
```

What are our options?

MATCH ROCM VERSIONS

```
login2:~$ module show cce |& grep -i rocm
- AMD ROCm 5.2 support
- AMD ROCm 5.0 and 5.1 support
  AMD GPU offloading support requires at least ROCm 5.0
login2:~$ module show perftools-base |& grep -i rocm
  o Support for AMD ROCm 5.1
  o As of ROCm version 4.5, SLES 15 support is limited to SP3.
login2:~$ module -t avail perftools-base
/opt/cray/pe/lmod/modulefiles/core:
perftools-base/21.12.0
perftools-base/22.04.0
perftools-base/22.05.0
perftools-base/22.06.0
perftools-base/22.09.0
perftools-base/22.12.0
login2:~$ module show perftools-base/22.12.0 |& grep -i rocm
  o Support for AMD ROCm 5.3.0
  o Support for OpenMP device tracing when using AMD ROCm 5.3.0's OpenMP runtime
login2:~$ module show perftools-base/22.09.0 |& grep -i rocm
  o Support for AMD ROCm 5.2.0
    including Intel icx/ifx, AMD aocc/rocm, and CCE compilers (see the OpenMP section of
  o ROCm 5.2.0 introduced API changes that prevent backwards compatibility with previous
```

Too new

MATCH ROCM VERSIONS

```
login2:~$ module show cce |& grep -i rocm
- AMD ROCm 5.2 support
- AMD ROCm 5.0 and 5.1 support
  AMD GPU offloading support requires at least ROCm 5.0
login2:~$ module show perftools-base |& grep -i rocm
  o Support for AMD ROCm 5.1
  o As of ROCm version 4.5, SLES 15 support is limited to SP3.
login2:~$ module -t avail perftools-base
/opt/cray/pe/lmod/modulefiles/core:
perftools-base/21.12.0
perftools-base/22.04.0
perftools-base/22.05.0
perftools-base/22.06.0
perftools-base/22.09.0
perftools-base/22.12.0
login2:~$ module show perftools-base/22.12.0 |& grep -i rocm
  o Support for AMD ROCm 5.3.0
  o Support for OpenMP device tracing when using AMD ROCm 5.3.0's OpenMP runtime
login2:~$ module show perftools-base/22.09.0 |& grep -i rocm
  o Support for AMD ROCm 5.2.0
    including Intel icx/ifx, AMD aocc/rocm, and CCE compilers (see the OpenMP section of
  o ROCm 5.2.0 introduced API changes that prevent backwards compatibility with previous
```

Just right

NEWS FLASH!

```
login2:~$ module show cce/15.0.0 |& grep -i rocm
CAST-30478 LAMMPS RIGID package build failure with CCE14+ROCm5.1
AMD GPU offloading support requires at least ROCm 5.0
```

Regression?

No, cce/15 should support newer version of rocm/5.x.x *as they appear*

INSTRUMENT FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load perftools-base/22.09.0
module load rocm/5.2.0
module load perftools
ftn -fopenmp ...
pat_build -g hip,io,mpi,omp -w -f <exe>
```

Switch to matching modules

INSTRUMENT FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load perftools-base/22.09.0
module load rocm/5.2.0
module load perftools        Add perftools module
ftn -fopenmp ...
pat_build -g hip,io,mpi,omp -w -f <exe>
```

INSTRUMENT FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load perftools-base/22.09.0
module load rocm/5.2.0
module load perftools
ftn -fopenmp ...
pat_build -g hip,io,mpi,omp -w -f <exe>
```

Wrappers automatically compile and link for instrumentation

INSTRUMENT FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load perftools-base/22.09.0
module load rocm/5.2.0
module load perftools
ftn -fopenmp ...
pat_build -g hip,io,mpi,omp -w -f <exe>
```

Create instrumented <exe>+pat

INSTRUMENT FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load perftools-base/22.09.0
module load rocm/5.2.0
module load perftools
ftn -fopenmp ...
pat_build -g hip,io,mpi,omp -w -f <exe>
```

Trace Hip, I/O, MPI, and OpenMP functions

INSTRUMENT FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load perftools-base/22.09.0
module load rocm/5.2.0
module load perftools
ftn -fopenmp ...
pat_build -g hip,io,mpi,omp -w -f <exe>
```

Trace user functions

INSTRUMENT FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load perftools-base/22.09.0
module load rocm/5.2.0
module load perftools
ftn -fopenmp ...
pat_build -g hip,io,mpi,omp -w -f <exe>
```

Overwrite <exe>+pat, *if it exists*

PROFILE FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load perftools-base/22.09.0
module load rocm/5.2.0
module load perftools
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>+pat
```

Match build modules

New output:

```
Experiment data directory written:
<path>/<exe>+pat+<unique>t
```

PROFILE FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load perftools-base/22.09.0
module load rocm/5.2.0
module load perftools
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>+pat
```

Particularly important with non-default modules

New output:

```
Experiment data directory written:
<path>/<exe>+pat+<unique>t
```

PROFILE FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load perftools-base/22.09.0
module load rocm/5.2.0
module load perftools
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>+pat
```

Run instrumented executable

New output:

```
Experiment data directory written:
<path>/<exe>+pat+<unique>t
```

PROFILE FORTRAN WITH OPENMP OFFLOAD

```
module load craype-accel-amd-gfx90a
module load perftools-base/22.09.0
module load rocm/5.2.0
module load perftools
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>+pat
```

New output:

Experiment data directory written:
<path>/<exe>+pat+<unique>t

BUILD HIP

```
module load rocm
export CXX=hipcc
export CXXFLAGS="-g -O3 -std=c++17 --offload-arch=gfx90a -Wall
-I${CRAY_MPICH_DIR}/include"
export LD=hipcc
export LDFLAGS="${CXXFLAGS} -Wall -L${CRAY_MPICH_DIR}/lib
${PE_MPICH_GTL_DIR_amd_gfx90a}"
export LIBS="-lmpi ${PE_MPICH_GTL_LIBS_amd_gfx90a}"
make
```

BUILD HIP

Using hipcc *directly, not* PrgEnv[-cray]-amd

```
module load rocm
export CXX=hipcc
export CXXFLAGS="-g -O3 -std=c++17 --offload-arch=gfx90a -Wall
-I${CRAY_MPICH_DIR}/include"
export LD=hipcc
export LDFLAGS="${CXXFLAGS} -L${CRAY_MPICH_DIR}/lib
${PE_MPICH_GTL_DIR_amd_gfx90a}"
export LIBS="-lmpi ${PE_MPICH_GTL_LIBS_amd_gfx90a}"
make
```

BUILD HIP

```
module load rocm
export CXX=hipcc
export CXXFLAGS="-g -O3 -std=c++17 --offload-arch=gfx90a -Wall
-I${CRAY_MPICH_DIR}/include"
export LD=hipcc
export LDFLAGS="${CXXFLAGS} -L${CRAY_MPICH_DIR}/lib
${PE_MPICH_GTL_DIR_amd_gfx90a}"
export LIBS="-lmpi ${PE_MPICH_GTL_LIBS_amd_gfx90a}"
make
```

With HPE Cray MPI

BUILD HIP

```
module load rocm
export CXX=hipcc
export CXXFLAGS="-g -O3 -std=c++17 --offload-arch=gfx90a -Wall
-I${CRAY_MPICH_DIR}/include"
export LD=hipcc
export LDFLAGS="${CXXFLAGS} -L${CRAY_MPICH_DIR}/lib
${PE_MPICH_GTL_DIR_amd_gfx90a}"
export LIBS="-lmpi ${PE_MPICH_GTL_LIBS_amd_gfx90a}"
make
```

GPU aware

RUN HIP

```
module load rocm
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>
```

RUN HIP

Match build modules

```
module load rocm
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>
```

RUN HIP

Use module goodies to run

```
module load rocm
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>
```

RUN HIP

```
module load rocm
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>
```

Use GPU buffers for MPI

RUN HIP

```
module load rocm
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>
```

Align MPI tasks, GPUs, and network interfaces

RUN HIP

```
module load rocm
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>
```

Tasks use multiple threads for GPU signals,
even without OpenMP threading

INSTRUMENT HIP?

MATCH ROCM VERSIONS

```
login2:~$ module -t avail perftools-base
/opt/cray/pe/lmod/modulefiles/core:
perftools-base/21.12.0
perftools-base/22.04.0
perftools-base/22.05.0
perftools-base/22.06.0
perftools-base/22.09.0
perftools-base/22.12.0
login2:~$ module show perftools-base/22.12.0 |& grep -i rocm
  o Support for AMD ROCm 5.3.0
  o Support for OpenMP device tracing when using AMD ROCm 5.3.0's
OpenMP runtime
```

MATCH ROCM VERSIONS

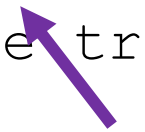
What's the newest perftools-base*?*

```
login2:~$ module -t avail perftools-base
/opt/cray/pe/lmod/modulefiles/core:
perftools-base/21.12.0
perftools-base/22.04.0
perftools-base/22.05.0
perftools-base/22.06.0
perftools-base/22.09.0
perftools-base/22.12.0
login2:~$ module show perftools-base/22.12.0 |& grep -i rocm
  o Support for AMD ROCm 5.3.0
  o Support for OpenMP device tracing when using AMD ROCm 5.3.0's
OpenMP runtime
```

MATCH ROCM VERSIONS

```
login2:~$ module -t avail perftools-base
/opt/cray/pe/lmod/modulefiles/core:
perftools-base/21.12.0
perftools-base/22.04.0
perftools-base/22.05.0
perftools-base/22.06.0
perftools-base/22.09.0
perftools-base/22.12.0
login2:~$ module show perftools-base/22.12.0 |& grep -i rocm
  o Support for AMD ROCm 5.3.0
  o Support for OpenMP device tracing when using AMD ROCm 5.3.0's
OpenMP runtime
```

What rocm *does it support?*



INSTRUMENT HIP

```
module load perftools-base/22.12.0
module load rocm/5.3.0
module load perftools
export CXX=hipcc
export CXXFLAGS="$(pat_opts include hipcc) $(pat_opts pre_compile
hipcc) -g -O3 -std=c++17 --offload-arch=gfx90a -Wall
-I${CRAY_MPICH_DIR}/include $(pat_opts post_compile hipcc)"
export LD=hipcc
export LDFLAGS="$(pat_opts pre_link hipcc) ${CXXFLAGS}
-L${CRAY_MPICH_DIR}/lib ${PE_MPICH_GTL_DIR_amd_gfx90a}"
export LIBS="-lmpi ${PE_MPICH_GTL_LIBS_amd_gfx90a} $(pat_opts
post_link hipcc)"
make
pat_build -g hip,io,mpi,omp -w -f <exe>
```

INSTRUMENT HIP

```
module load perftools-base/22.12.0
module load rocm/5.3.0
module load perftools
export CXX=hipcc
export CXXFLAGS="$(pat_opts include hipcc) $(pat_opts pre_compile
hipcc) -g -O3 -std=c++17 --offload-arch=gfx90a -Wall
-I${CRAY_MPICH_DIR}/include $(pat_opts post_compile hipcc)"
export LD=hipcc
export LDFLAGS="$(pat_opts pre_link hipcc) ${CXXFLAGS}
-L${CRAY_MPICH_DIR}/lib ${PE_MPICH_GTL_DIR_amd_gfx90a}"
export LIBS="-lmpi ${PE_MPICH_GTL_LIBS_amd_gfx90a} $(pat_opts
post_link hipcc)"
make
pat_build -g hip,io,mpi,omp -w -f <exe>
```

Switch to matching modules

INSTRUMENT HIP

```
module load perftools-base/22.12.0
module load rocm/5.3.0
module load perftools
export CXX=hipcc
export CXXFLAGS="$(pat_opts include hipcc) $(pat_opts pre_compile
hipcc) -g -O3 -std=c++17 --offload-arch=gfx90a -Wall
-I${CRAY_MPICH_DIR}/include $(pat_opts post_compile hipcc)"
export LD=hipcc
export LDFLAGS="$(pat_opts pre_link hipcc) ${CXXFLAGS}
-L${CRAY_MPICH_DIR}/lib ${PE_MPICH_GTL_DIR_amd_gfx90a}"
export LIBS="-lmpi ${PE_MPICH_GTL_LIBS_amd_gfx90a} $(pat_opts
post_link hipcc)"
make
pat_build -g hip,io,mpi,omp -w -f <exe>
```

Add perftools *module*

INSTRUMENT HIP

```
module load perftools-base/22.12.0
module load rocm/5.3.0
module load perftools
export CXX=hipcc
export CXXFLAGS="$(pat_opts include hipcc) $(pat_opts pre_compile
hipcc) -g -O3 -std=c++17 --offload-arch=gfx90a -Wall
-I${CRAY_MPICH_DIR}/include $(pat_opts post_compile hipcc)"
export LD=hipcc
export LDFLAGS="$(pat_opts pre_link hipcc) ${CXXFLAGS}
-L${CRAY_MPICH_DIR}/lib ${PE_MPICH_GTL_DIR_amd_gfx90a}"
export LIBS="-lmpi ${PE_MPICH_GTL_LIBS_amd_gfx90a} $(pat_opts
post_link hipcc)"
make
pat_build -g hip,io,mpi,omp -w -f <exe>
```

Add arguments using:
pat_opts <*build phase*> <*compiler*>

INSTRUMENT HIP

Create instrumented <exe>+pat

Trace Hip, I/O, MPI, OpenMP, and user functions

Overwrite <exe>+pat, if it exists

```
module load perftools-base/22.12.0
module load rocm/5.3.0
module load perftools
export CXX=hipcc
export CXXFLAGS="$(pat_opts include hipcc) $(pat_opts pre_compile
hipcc) -g -O3 -std=c++17 --offload-arch=gfx90a -Wall
-I${CRAY_MPICH_DIR}/include $(pat_opts post_compile hipcc)"
export LD=hipcc
export LDFLAGS="$(pat_opts pre_link hipcc) ${CXXFLAGS}
-L${CRAY_MPICH_DIR}/lib ${PE_MPICH_GTL_DIR_amd_gfx90a}"
export LIBS="-lmpi ${PE_MPICH_GTL_LIBS_amd_gfx90a} $(pat_opts
post_link hipcc)"
make
pat_build -g hip,io,mpi,omp -w -f <exe>
```

BUILD HIP WITH CMAKE

Turn on Hip

```
-DENABLE_HIP=ON
-DWITH_HIP=TRUE
```

Target MI250X

```
-DWITH-GPU-ARCH=gfx90a
-DGPU_TARGETS=gfx90a
-DHIP_ARCH=gfx90a
```

Use `hipcc`

```
-DCMAKE_CXX_COMPILER=hipcc
-DMPI_CXX_COMPILER="hipcc -I${MPICH_DIR}/include"
```

Add arguments

```
-DCMAKE_CXX_FLAGS='--offload-arch=gfx90a -munsafe-fp-atomics'
```

INSTRUMENT HIP WITH CMAKE

Turn on Hip

```
-DENABLE_HIP=ON
-DWITH_HIP=TRUE
```

Target MI250X

```
-DWITH-GPU-ARCH=gfx90a
-DGPU_TARGETS=gfx90a
-DHIP_ARCH=gfx90a
```

Use `hipcc`

```
-DCMAKE_CXX_COMPILER=hipcc
-DMPI_CXX_COMPILER="hipcc -I${MPICH_DIR}/include"
```

Double quotes!

Add Perftools arguments

```
-DCMAKE_CXX_FLAGS="$(pat_opts include hipcc) $(pat_opts pre_compile hipcc)
--offload-arch=gfx90a -munsafe-fp-atomics $(pat_opts post_compile hipcc)"
-DCMAKE_EXE_LINKER_FLAGS="$(pat_opts pre_link hipcc) $(pat_opts post_link
hipcc)"
```

```
readelf: Warning: Unrecognized form: 0x23
readelf: Warning: Unrecognized form: 0x22
readelf: Warning: Unrecognized form: 0x22
readelf: Warning: Unrecognized form: 0x22
readelf: Warning: Unrecognized form: 0x22
readelf: Warning: Unrecognized form: 0x22
readelf: Warning: Unrecognized form: 0x22
readelf: Warning: Unrecognized form: 0x23
readelf: Warning: Unrecognized form: 0x22
readelf: Warning: Unrecognized form: 0x22
readelf: Warning: Unrecognized form: 0x23
readelf: Warning: Unrecognized form: 0x22
readelf: Warning: Unrecognized form: 0x22
readelf: Warning: Unrecognized form: 0x23
readelf: Warning: Unrecognized form: 0x22
readelf: Warning: Unrecognized form: 0x23
readelf: Warning: Unrecognized form: 0x22
readelf: Warning: Unrecognized form: 0x23
readelf: Warning: Unrecognized form: 0x22
readelf: Warning: Unrecognized form: 0x22
```

Issue with readelf *from* binutils-2.38 *and before*

Fixed in binutils-2.39

INSTRUMENT HIP WITH CMAKE

Turn on Hip

```
-DENABLE_HIP=ON
-DWITH_HIP=TRUE
```

Target MI250X

```
-DWITH-GPU-ARCH=gfx90a
-DGPU_TARGETS=gfx90a
-DHIP_ARCH=gfx90a
```

Workaround: compile using older debug format

Use `hipcc`

```
-DCMAKE_CXX_COMPILER=hipcc
-DMPI_CXX_COMPILER="hipcc -I${MPICH_DIR}/include"
```

Add Perftools arguments

```
-DCMAKE_CXX_FLAGS="$(pat_opts include hipcc) $(pat_opts pre_compile hipcc)
-gdwarf-4 --offload-arch=gfx90a -munsafe-fp-atomics $(pat_opts
post_compile hipcc)"
-DCMAKE_EXE_LINKER_FLAGS="$(pat_opts pre_link hipcc) $(pat_opts post_link
hipcc)"
```

PROFILE HIP

```
module load perftools-base/22.12.0
module load rocm/5.3.0
module load perftools
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>+pat
```

New output:

```
Experiment data directory written:
<path>/<exe>+pat+<unique>t
```

PROFILE HIP

```
module load perftools-base/22.12.0
module load rocm/5.3.0
module load perftools
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>+pat
```

Match build modules

New output:

```
Experiment data directory written:
<path>/<exe>+pat+<unique>t
```

PROFILE HIP

```
module load perftools-base/22.12.0
module load rocm/5.3.0
module load perftools
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>+pat
```

Run instrumented executable

New output:

```
Experiment data directory written:
<path>/<exe>+pat+<unique>t
```

PROFILE HIP

```
module load perftools-base/22.12.0
module load rocm/5.3.0
module load perftools
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>+pat
```

New output:

```
Experiment data directory written:
<path>/<exe>+pat+<unique>t
```

PROFILE HIP

```
module load perftools-base/22.12.0
module load rocm/5.3.0
module load perftools
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>+pat
```

Or maybe:

```
pat[WARNING][0]: 19543 CID to EID records were dropped. Try
increasing the value of PAT_RT_ACC_CID_TO_EID_BUFFER_SIZE.
Experiment data directory written:
<path>/<exe>+pat+<unique>t
```

PROFILE HIP

```
login2:~$ man pat | grep -A4 PAT_RT_ACC_CID_TO_EID_BUFFER_SIZE
       PAT_RT_ACC_CID_TO_EID_BUFFER_SIZE
           Specify the size in bytes of the buffer used to collect
ROC-tracer callback records. Size is not case-sensitive and can be
specified in kilobytes (KB), megabytes (MB), or gigabytes (GB).

           Default: 1MB
```

PROFILE HIP

```
module load perftools-base/22.12.0
module load rocm/5.3.0
module load perftools
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
export PAT_RT_ACC_CID_TO_EID_BUFFER_SIZE=128MB
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>+pat
```

Your mileage may vary

No more warning output:

```
Experiment data directory written:
<path>/<exe>+pat+<unique>t
```

PAT_REPORT

DEFAULT REPORT

```
$ pat_report <path>/<exe>+pat+<unique>t
```

First time running on this data directory?
Wait for a few seconds to process files

Subsequent *pat_report* calls on this directory should be fast

```
CrayPat/X:  Version 22.12.0 Revision 8379d561d  11/09/22 20:11:43

Number of PEs (MPI ranks):    8

Numbers of PEs per Node:      8

Numbers of Threads per PE:    1

Number of Cores per Socket:  64
...
```

DEFAULT REPORT

Table 1: Profile by Function Group and Function

```
  Time% |       Time |      Imb. |  Imb. |      Calls | Group
        |            |      Time | Time% |            |  Function
        |            |           |       |            |   PE=HIDE

 100.0% | 21.962528 |       -- |    -- | 854,964.6 | Total
|-----------------------------------------------------------------------
|  60.2% | 13.227199 |       -- |    -- | 599,673.0 | MPI
||----------------------------------------------------------------------
||  58.8% | 12.903094 | 2.082692 | 15.9% |  22,200.0 | MPI_Waitall
||   1.4% |  0.310501 | 0.038408 | 12.6% | 288,601.0 | MPI_Isend
||======================================================================
|  32.6% |  7.160514 |       -- |    -- | 254,809.0 | HIP
||----------------------------------------------------------------------
||  10.4% |  2.273137 | 2.487895 | 59.7% |      20.0 | hipStreamCreate
||   7.8% |  1.710117 | 1.106463 | 44.9% |     253.0 | hipMemset
||   7.2% |  1.589945 | 0.085563 |  5.8% |  22,000.0 | hipStreamSynchronize
||   5.7% |  1.260084 | 0.008705 |  0.8% |      20.0 | hipStreamDestroy
||======================================================================
|   5.2% |  1.144598 |       -- |    -- |      25.0 | MPI_SYNC
||----------------------------------------------------------------------
||   5.0% |  1.101117 | 1.100117 | 99.9% |      10.0 | MPI_Barrier(sync)
||======================================================================
```

DEFAULT REPORT

Table 1: Profile by Function Group and Function

Time%	Time	Imb. Time	Imb. Time%	Calls	Group Function PE=HIDE
100.0%	21.962528	--	--	854,964.6	Total
60.2%	13.227199	--	--	599,673.0	MPI
58.8%	12.903094	2.082692	15.9%	22,200.0	MPI_Waitall
1.4%	0.310501	0.038408	12.6%	288,601.0	MPI_Isend
32.6%	7.160514				
10.4%	2.273137				
7.8%	1.710117	1.106463	44.9%	253.0	hipMemset
7.2%	1.589945	0.085563	5.8%	22,000.0	hipStreamSynchronize
5.7%	1.260084	0.008705	0.8%	20.0	hipStreamDestroy
5.2%	1.144598	--	--	25.0	MPI_SYNC
5.0%	1.101117	1.100117	99.9%	10.0	MPI_Barrier(sync)

Hides functions taking less than 1% of host runtime

DEFAULT REPORT

Table 1: Profile by Function Group and Function

Time%	Time	Imb. Time	Imb. Time%	Calls	Group Function PE=HIDE
58.8%	12.903094	2.082692	15.9%	22,200.0	MPI_Waitall
1.4%	0.310501	0.038408	12.6%	288,601.0	MPI_Isend
32.6%	7.160514	--	--	254,809.0	HIP

Imbalance Time = Maximum Time for any Single Task - Average Time across Tasks

*Imbalance Time % = (Imbalance Time / Max Time) * Tasks / (Tasks – 1)*

100% Imbalance Time means only one task spent time in that function

DEFAULT REPORT

Table 1: Profile by Function Group and Function

Time%	Time	Imb. Time	Imb. Time%	Calls	Group Function PE=HIDE
100.0%	21.962528	--	--	854,964.6	Total
60.2%	**13.227199**	**--**	**--**	**599,673.0**	**MPI**
58.8%	**12.903094**	**2.082692**	**15.9%**	**22,200.0**	**MPI_Waitall**
1.4%	**0.310501**	**0.038408**	**12.6%**	**288,601.0**	**MPI_Isend**
32.6%	7.160514	--	--	254,809.0	HIP
10.4%	2.273137	2.487895	59.7%	20.0	hipStreamCreate
7.8%	1.710117	1.106463	44.9%		
7.2%	1.589945	0.085563	5.8%		
5.7%	1.260084	0.008705	0.8%		
5.2%	1.144598	--	--		
5.0%	1.101117	1.100117	99.9%		

MPI time on point to point
(and "active" work of collectives)

DEFAULT REPORT

```
Table 1:  Profile by Function Group and Function

  Time% |       Time |      Imb. |  Imb. |      Calls | Group
        |            |      Time | Time% |            |  Function
        |            |           |       |            |   PE=HIDE

 100.0% | 21.962528 |       -- |    -- | 854,964.6 | Total
|-----------------------------------------------------------------------
|  60.2% | 13.227199 |       -- |    -- | 599,673.0 | MPI
||----------------------------------------------------------------------
||  58.8% | 12.903094 | 2.082692 |
||   1.4% |  0.310501 | 0.038408 |
||======================================
|  32.6% |  7.160514 |       -- |
||--------------------------------
||  10.4% |  2.273137 | 2.487895 |
||   7.8% |  1.710117 | 1.106463 |
||   7.2% |  1.589945 | 0.085563 |   5.8% |  22,000.0 | hipStreamSynchronize
||   5.7% |  1.260084 | 0.008705 |   0.8% |      20.0 | hipStreamDestroy
||======================================================================
|   5.2% |  1.144598 |       -- |    -- |      25.0 | MPI_SYNC
||----------------------------------------------------------------------
||   5.0% |  1.101117 | 1.100117 | 99.9% |      10.0 | MPI_Barrier(sync)
||======================================================================
```

MPI synchronization time
(Perftools adds a barrier before each collective)

DEFAULT REPORT

Table 1: Profile by Function Group and Function

Time%	Time	Imb. Time	Imb. Time%		
100.0%	21.962528	--	--		
60.2%	13.227199	--	--		
58.8%	12.903094	2.082692	15.9%	22,200.0	MPI_Waitall
1.4%	0.310501	0.038408	12.6%	288,601.0	MPI_Isend
32.6%	**7.160514**	**--**	**--**	**254,809.0**	**HIP**
10.4%	**2.273137**	**2.487895**	**59.7%**	**20.0**	**hipStreamCreate**
7.8%	**1.710117**	**1.106463**	**44.9%**	**253.0**	**hipMemset**
7.2%	**1.589945**	**0.085563**	**5.8%**	**22,000.0**	**hipStreamSynchronize**
5.7%	**1.260084**	**0.008705**	**0.8%**	**20.0**	**hipStreamDestroy**
5.2%	1.144598	--	--	25.0	MPI_SYNC
5.0%	1.101117	1.100117	99.9%	10.0	MPI_Barrier(sync)

Host times spent on Hip API calls,
not kernel execution times on accelerator

DEFAULT REPORT

```
Notes for table 2:

  This table shows functions that have the most significant exclusive
    time, taking the maximum time across ranks and threads.
  For further explanation, see the "General table notes" below, or
    use:  pat_report -v -O profile_max ...
```

Table 2: Profile of maximum function times

Time%	Time	Imb. Time	Imb. Time%	Function PE=[max,min]
100.0%	14.985787	2.082692	15.9%	MPI_Waitall
100.0%	14.985787	--	--	pe.7
69.2%	10.365211	--	--	pe.0
31.8%	4.761032	2.487895	59.7%	hipStreamCreate
31.8%	4.761032	--	--	pe.5
1.4%	0.210372	--	--	pe.7

Long poles in the CPU tent

DEFAULT REPORT

```
Notes for table 2:

  This table shows functions that have the most significant exclusive
    time, taking the maximum time across ranks and threads.
  For further explanation, see the "General table notes" below, or
    use:  pat_report -v -O profile_max ...

Table 2:  Profile of maximum function times

  Time% |      Time |     Imb. |  Imb. | Function
        |           |     Time | Time% |  PE=[max,min]
|-----------------------------------------------------------
| 100.0% | 14.985787 | 2.082692 | 15.9% | MPI_Waitall
||----------------------------------------------------------
|| 100.0% | 14.985787 |       -- |    -- | pe.7
||  69.2% | 10.365211 |       -- |    -- | pe.0
||==========================================================
|  31.8% |  4.761032 | 2.487895 | 59.7% | hipStreamCreate
||----------------------------------------------------------
||  31.8% |  4.761032 |       -- |    -- | pe.5
||   1.4% |  0.210372 |       -- |    -- | pe.7
||==========================================================
```

Variability across MPI ranks

DEFAULT REPORT

Made for parallel codes!

Table 3: Load Balance with MPI Message Stats

Time%	Time	MPI Msg Count	MPI Msg Bytes	Avg MPI Msg Size	Group PE=[mmm]
100.0%	21.962528	77,713.0	7,550,930,464.0	97,164.32	Total
60.2%	13.227199	77,713.0	7,550,930,464.0	97,164.32	MPI
69.7%	15.313914	77,713.0	7,550,930,464.0	97,164.32	pe.7
57.8%	12.691260	77,713.0	7,550,930,464.0	97,164.32	pe.2
48.8%	10.716753	77,713.0	7,550,930,464.0	97,164.32	pe.0
32.6%	7.160514	0.0	0.0	--	HIP
48.9%	10.732649	0.0	0.0	--	pe.0
28.6%	6.285409	0.0	0.0	--	pe.4
22.3%	4.899067	0.0	0.0	--	pe.7
5.2%	1.144598	0.0	0.0	--	MPI_SYNC
6.0%	1.317917	0.0	0.0	--	pe.7
6.0%	1.307408	0.0	0.0	--	pe.1
0.2%	0.048479	0.0	0.0	--	pe.0

DEFAULT REPORT

Did I mention "made for parallel codes"?

Table 4: MPI Message Stats by Caller

```
    MPI |   MPI Msg Bytes |  MPI Msg |    MsgSz |    256<= |  64KiB<= | Function
    Msg |                 |    Count |      <16 |    MsgSz |    MsgSz |  Caller
 Bytes% |                 |          |    Count |    <4KiB |    <1MiB |   PE=[mmm]
        |                 |          |          |    Count |    Count |

 100.0% | 7,550,930,464.0 | 77,713.0 | 11,113.0 | 33,300.0 | 33,300.0 | Total
|-----------------------------------------------------------------------
| 100.0% | 7,550,930,400.0 | 77,700.0 | 11,100.0 | 33,300.0 | 33,300.0 | MPI_Isend
|  99.1% | 7,482,904,000.0 | 77,000.0 | 11,000.0 | 33,000.0 | 33,000.0 |  Faces::share
3        |                 |          |          |          |          |   main
||||-------------------------------------------------------------------
4|||  99.1% | 7,482,904,000.0 | 77,000.0 | 11,000.0 | 33,000.0 | 33,000.0 | pe.0
4|||  99.1% | 7,482,904,000.0 | 77,000.0 | 11,000.0 | 33,000.0 | 33,000.0 | pe.4
4|||  99.1% | 7,482,904,000.0 | 77,000.0 | 11,000.0 | 33,000.0 | 33,000.0 | pe.7
|=======================================================================
```

DEFAULT REPORT

Sees all!

Table 5: File Input Stats by Filename

Avg Read Time per Reader Rank	Avg Read MiBytes per Reader Rank	Read Rate MiBytes/sec	Number of Reader Ranks	Avg Reads per Reader Rank	Bytes/ Call	File Name=!x/^/(proc|sys)/ PE=HIDE
0.001096	0.000690	0.629072	8	3.0	241.00	/opt/rocm-5.3.0/bin/.hipVersion
0.000019	0.000097	5.110043	1	102.0	1.00	stdin
0.000006	0.004499	729.125818	1	1.0	4,718.00	/tmp/comgr-d2c04d/input/CompileSource
0.000006	0.004499	802.038400	1	1.0	4,718.00	/tmp/comgr-45ac6b/input/CompileSource
0.000004	0.004499	1,122.893792	1	1.0	4,718.00	/tmp/comgr-65ed8d/input/CompileSource
0.000004	0.004499	1,137.082493	1	1.0	4,718.00	/tmp/comgr-01e20d/input/CompileSource
0.000004	0.000004	1.064072	8	1.0	4.00	/dev/urandom
0.000004	0.004499	1,268.876318	1	1.0	4,718.00	/tmp/comgr-1e375f/input/CompileSource
0.000003	0.004499	1,321.032127	1	1.0	4,718.00	/tmp/comgr-e940d6/input/CompileSource
0.000003	0.004499	1,352.806802	1	1.0	4,718.00	/tmp/comgr-9d0bcc/input/CompileSource
0.000003	0.000216	65.267479	8	1.0	227.00	/etc/os-release
0.000003	0.004499	1,407.833362	1	1.0	4,718.00	/tmp/comgr-d8fa98/input/CompileSource
0.000003	0.008984	2,912.029841	1	1.0	9,420.00	/tmp/comgr-65ed8d/output/CompileSource.bc
0.000003	0.008984	3,308.881054	1	1.0	9,420.00	/tmp/comgr-45ac6b/output/CompileSource.bc
0.000003	0.008984	3,383.658027	1	1.0	9,420.00	/tmp/comgr-d2c04d/output/CompileSource.bc

DEFAULT REPORT

Table 5: File Input Stats by Filename

Avg Read Time per Reader Rank	Avg Read MiBytes per Reader Rank	Read Rate MiBytes/sec	Number of Reader Ranks	Avg Reads per Reader Rank	Bytes/ Call	File Name=!x/^/(proc|sys)/ PE=HIDE
0.001096	**0.000690**	**0.629072**	**8**	**3.0**	**241.00**	**/opt/rocm-5.3.0/bin/.hipVersion**
0.000019	0.000097	5.110043	1	102.0	1.00	stdin
0.000006	0.004499	729.125818	1	1.0	4,718.00	/tmp/comgr-d2c04d/input/CompileSource
0.000006	0.004499	802.				pileSource
0.000004	0.004499	1,122.				pileSource
0.000004	0.004499	1,137.				pileSource
0.000004	0.000004	1.				
0.000004	0.004499	1,268.				pileSource
0.000003	0.004499	1,321.032127	1	1.0	4,718.00	/tmp/comgr-e940d6/input/CompileSource
0.000003	0.004499	1,352.806802	1	1.0	4,718.00	/tmp/comgr-9d0bcc/input/CompileSource
0.000003	0.000216	65.267479	8	1.0	227.00	/etc/os-release
0.000003	0.004499	1,407.833362	1	1.0	4,718.00	/tmp/comgr-d8fa98/input/CompileSource
0.000003	0.008984	2,912.029841	1	1.0	9,420.00	/tmp/comgr-65ed8d/output/CompileSource.bc
0.000003	0.008984	3,308.881054	1	1.0	9,420.00	/tmp/comgr-45ac6b/output/CompileSource.bc
0.000003	0.008984	3,383.658027	1	1.0	9,420.00	/tmp/comgr-d2c04d/output/CompileSource.bc

Each task checking in with Hip

DEFAULT REPORT

Table 5: File Input Stats by Filename

Avg Read Time per Reader Rank	Avg Read MiBytes per Reader Rank	Read Rate MiBytes/sec	Number of Reader Ranks	Avg Reads per Reader Rank	Bytes/ Call	File Name=!x/^/(proc|sys)/ PE=HIDE
0.001096	0.000690	0.629072	8	3.0	241.00	/opt/rocm-5.3.0/bin/.hipVersion
0.000019	**0.000097**	**5.110043**	**1**	**102.0**	**1.00**	**stdin**
0.000006	0.004499	729.125818	1	1.0	4,718.00	/tmp/comgr-d2c04d/input/CompileSource
0.000006	0.004499	802.038400	1	1.0	4,718.00	/tmp/comgr-45ac6b/input/CompileSource
0.000004	0.004499	1,122.893792	1	1.0	4,718.00	/tmp/comgr-65ed8d/input/CompileSource
0.000004	0.004499	1,137				mpileSource
0.000004	0.000004	1				
0.000004	0.004499	1,268				mpileSource
0.000003	0.004499	1,321				mpileSource
0.000003	0.004499	1,352				mpileSource
0.000003	0.000216	65.267479	8	1.0	227.00	/etc/os-release
0.000003	0.004499	1,407.833362	1	1.0	4,718.00	/tmp/comgr-d8fa98/input/CompileSource
0.000003	0.008984	2,912.029841	1	1.0	9,420.00	/tmp/comgr-65ed8d/output/CompileSource.bc
0.000003	0.008984	3,308.881054	1	1.0	9,420.00	/tmp/comgr-45ac6b/output/CompileSource.bc
0.000003	0.008984	3,383.658027	1	1.0	9,420.00	/tmp/comgr-d2c04d/output/CompileSource.bc

Standard input read by first rank

DEFAULT REPORT

Table 5: File Input Stats by Filename

Avg Read Time per Reader Rank	Avg Read MiBytes per Reader Rank	Read Rate MiBytes/sec	Number of Reader Ranks	Avg Reads per Reader Rank	Bytes/ Call	File Name=!x/^/(proc|sys)/ PE=HIDE
0.001096	0.000690	0.629072	8	3.0	241.00	/opt/rocm-5.3.0/bin/.hipVersion
0.000019	0.000097	5.110043	1	102.0	1.00	stdin
0.000006	**0.004499**	**729.125818**	**1**	**1.0**	**4,718.00**	**/tmp/comgr-d2c04d/input/CompileSource**
0.000006	**0.004499**	**802.038400**	**1**	**1.0**	**4,718.00**	**/tmp/comgr-45ac6b/input/CompileSource**
0.000004	**0.004499**	**1,122.893792**	**1**	**1.0**	**4,718.00**	**/tmp/comgr-65ed8d/input/CompileSource**
0.000004	**0.004499**	**1,137.082493**	**1**	**1.0**	**4,718.00**	**/tmp/comgr-01e20d/input/CompileSource**
0.000004	0.000004	1.064072	8	1.0	4.00	/dev/urandom
0.000004	**0.004499**	**1,268.876318**	**1**	**1.0**	**4,718.00**	**/tmp/comgr-1e375f/input/CompileSource**
0.000003	**0.004499**	**1,321.032127**	**1**	**1.0**	**4,718.00**	**/tmp/comgr-e940d6/input/CompileSource**
0.000003	**0.004499**	**1,352.806802**	**1**	**1.0**	**4,718.00**	**/tmp/comgr-9d0bcc/input/CompileSource**
0.000003	0.000216	65.267479	8	1.0	227.00	/etc/os-release
0.000003	**0.004499**	**1,407.833362**	**1**	**1.0**	**4,718.00**	**/tmp/comgr-d8fa98/input/CompileSource**
0.000003	**0.008984**	**2,912.029841**	**1**	**1.0**	**9,420.00**	**/tmp/comgr-65ed8d/output/CompileSource.bc**
0.000003	**0.008984**	**3,308.881054**	**1**	**1.0**	**9,420.00**	**/tmp/comgr-45ac6b/output/CompileSource.bc**
0.000003	**0.008984**	**3,383.658027**	**1**	**1.0**	**9,420.00**	**/tmp/comgr-d2c04d/output/CompileSource.bc**

AMD Rocm Code Object Manager

DEFAULT REPORT

Table 5: File Input Stats by Filename

Avg Read Time per Reader Rank	Avg Read MiBytes per Reader Rank	Read Rate MiBytes/sec	Number of Reader Ranks	Avg Reads per Reader Rank	Bytes/ Call	File Name=!x/^/(proc|sys)/ PE=HIDE
0.001096	0.000690	0.629072	8	3.0	241.00	/opt/rocm-5.3.0/bin/.hipVersion
0.000019	0.000097	5.110043	1	102.0	1.00	stdin
0.000006	0.004499	729.125818	1	1.0	4,718.00	/tmp/comgr-d2c04d/input/CompileSource
0.000006	0.004499	802.038400	1	1.0	4,718.00	/tmp/comgr-45ac6b/input/CompileSource
0.000004	0.004499	1,122.893792	1	1.0	4,718.00	/tmp/comgr-65ed8d/
0.000004	0.004499	1,137.082493	1	1.0	4,718.00	/tmp/comgr-01e20d/
0.000004	**0.000004**	**1.064072**	**8**	**1.0**	**4.00**	**/dev/urandom**
0.000004	0.004499	1,268.876318	1	1.0	4,718.00	/tmp/comgr-1e375f/
0.000003	0.004499	1,321.032127	1	1.0	4,718.00	/tmp/comgr-e940d6/
0.000003	0.004499	1,352.806802	1	1.0	4,718.00	/tmp/comgr-9d0bcc/input/CompileSource
0.000003	0.000216	65.267479	8	1.0	227.00	/etc/os-release
0.000003	0.004499	1,407.833362	1	1.0	4,718.00	/tmp/comgr-d8fa98/input/CompileSource
0.000003	0.008984	2,912.029841	1	1.0	9,420.00	/tmp/comgr-65ed8d/output/CompileSource.bc
0.000003	0.008984	3,308.881054	1	1.0	9,420.00	/tmp/comgr-45ac6b/output/CompileSource.bc
0.000003	0.008984	3,383.658027	1	1.0	9,420.00	/tmp/comgr-d2c04d/output/CompileSource.bc

Some rando?

DEFAULT REPORT

Table 6: File Output Stats by Filename

Avg Write Time per Writer Rank	Avg Write MiBytes per Writer Rank	Write Rate MiBytes/sec	Number of Writer Ranks	Avg Writes per Writer Rank	Bytes/ Call	File Name=!x/^/(proc|sys)/ PE=HIDE
0.000819	2.625626	3,207.497783	1	2.0	1,376,584.00	/tmp/comgr-65ed8d/include/opencl1.2-c.pch
0.000772	2.625626	3,401.109094	1	2.0	1,376,584.00	/tmp/comgr-01e20d/include/opencl1.2-c.pch
0.000752	2.625626	3,491.133451	1	2.0	1,376,584.00	/tmp/comgr-9d0bcc/include/opencl1.2-c.pch
0.000750	2.625626	3,501.492428	1	2.0	1,376,584.00	/tmp/comgr-45ac6b/include/opencl1.2-c.pch
0.000746	2.625626	3,521.488853	1	2.0	1,376,584.00	/tmp/comgr-e940d6/include/opencl1.2-c.pch
0.000743	2.625626	3,532.546999	1	2.0	1,376,584.00	/tmp/comgr-d8fa98/include/opencl1.2-c.pch
0.000726	2.625626	3,614.348795	1	2.0	1,376,584.00	/tmp/comgr-d2c04d/include/opencl1.2-c.pch
0.000686	2.625626	3,829.686799	1	2.0	1,376,584.00	/tmp/comgr-1e375f/include/opencl1.2-c.pch
0.000035	0.000117	3.305461	8	28.6	4.28	stdout
0.000030	0.028086	945.137590	1	8.0	3,681.25	/tmp/comgr-c5aaed/output/linked.bc-39ea54f3.o.tmp
0.000026	0.028086	1,076.122021	1	8.0	3,681.25	/tmp/comgr-2a585c/output/linked.bc-d03ad3cc.o.tmp
0.000024	0.028086	1,149.828405	1	8.0	3,681.25	/tmp/comgr-a4204e/output/linked.bc-d15cf353.o.tmp
0.000023	0.028086	1,196.460280	1	8.0	3,681.25	/tmp/comgr-1c49b2/output/linked.bc-4f751122.o.tmp
0.000022	0.028086	1,270.214310	1	8.0	3,681.25	/tmp/comgr-34b1b2/output/linked.bc-f990c629.o.tmp
0.000021	0.028086	1,312.416291	1	8.0	3,681.25	/tmp/comgr-d935c2/output/linked.bc-30fbf309.o.tmp

DEFAULT REPORT

Table 6: File Output Stats by Filename

Avg Write Time per Writer Rank	Avg Write MiBytes per Writer Rank	Write Rate MiBytes/sec	Number of Writer Ranks	Avg Writes per Writer Rank	Bytes/ Call	File Name=!x/^/(proc|sys)/ PE=HIDE
0.000819	2.625626	3,207.497783	1	2.0	1,376,584.00	/tmp/comgr-65ed8d/include/opencl1.2-c.pch
0.000772	2.625626	3,401.109094	1	2.0	1,376,584.00	/tmp/comgr-01e20d/include/opencl1.2-c.pch
0.000752	2.625626	3,491.133451	1	2.0	1,376,584.00	/tmp/comgr-9d0bcc/include/opencl1.2-c.pch
0.000750	2.625626	3,501.492428	1	2.0	1,376,584.00	/tmp/comgr-45ac6b/include/opencl1.2-c.pch
0.000746	2.625626	3,521.488853	1	2.0	1,376,584.00	/tmp/comgr-e940d6/include/opencl1.2-c.pch
0.000743	2.625626	3,532.546999	1	2.0	1,376,584.00	/tmp/comgr-d8fa98/include/opencl1.2-c.pch
0.000726	2.625626	3,614.348795	1	2.0	1,376,584.00	/tmp/comgr-d2c04d/include/opencl1.2-c.pch
0.000686	2.625626	3,829.686799	1	2.0	1,376,584.00	/tmp/comgr-1e375f/include/opencl1.2-c.pch
0.000035	**0.000117**	**3.305461**	**8**	**28.6**	**4.28**	**stdout**
0.000030	0.028086	945.137590	1	8.0	3,681.25	/tmp/comgr-c5aaed/output/linked.bc-39ea54f3.o.tmp
0.000026	0.028086	1,076.122021	1	8.0	3,681.25	/tmp/comgr-2a585c/output/linked.bc-d03ad3cc.o.tmp
0.000024	0.028086	1,149.828405	1	8.0	3,681.25	/tmp/comgr-a4204e/output/linked.bc-d15cf353.o.tmp
0.000023	0.028086	1,196.4602				output/linked.bc-4f751122.o.tmp
0.000022	0.028086	1,270.2143				output/linked.bc-f990c629.o.tmp
0.000021	0.028086	1,312.4162				output/linked.bc-30fbf309.o.tmp

Standard output from all tasks

DEFAULT REPORT

Table 6: File Output Stats by Filename

Avg Write Time per Writer Rank	Avg Write MiBytes per Writer Rank	Write Rate MiBytes/sec	Number of Writer Ranks	Avg Writes per Writer Rank	Bytes/ Call	File Name=!x/^/(proc|sys)/ PE=HIDE
0.000819	2.625626	3,207.497783	1	2.0	1,376,584.00	/tmp/comgr-65ed8d/include/opencl1.2-c.pch
0.000772	2.625626	3,401.109094	1	2.0	1,376,584.00	/tmp/comgr-01e20d/include/opencl1.2-c.pch
0.000752	2.625626	3,491.133451	1	2.0	1,376,584.00	/tmp/comgr-9d0bcc/include/opencl1.2-c.pch
0.000750	2.625626	3,501.492428	1	2.0	1,376,584.00	/tmp/comgr-45ac6b/include/opencl1.2-c.pch
0.000746	2.625626	3,521.488853	1	2.0	1,376,584.00	/tmp/comgr-e940d6/include/opencl1.2-c.pch
0.000743	2.625626	3,532.546999	1	2.0	1,376,584.00	/tmp/comgr-d8fa98/include/opencl1.2-c.pch
0.000726	2.625626	3,614.348795	1	2.0	1,376,584.00	/tmp/comgr-d2c04d/include/opencl1.2-c.pch
0.000686	2.625626	3,829.686799	1	2.0	1,376,584.00	/tmp/comgr-1e375f/include/opencl1.2-c.pch
0.000035	0.000117	3.305461	8	28.6	4.28	stdout
0.000030	0.028086	945.137590	1	8.0	3,681.25	/tmp/comgr-c5aaed/output/linked.bc-39ea54f3.o.tmp
0.000026	0.028086	1,076.122021	1	8.0	3,681.25	/tmp/comgr-2a585c/output/linked.bc-d03ad3cc.o.tmp
0.000024	0.028086	1,149.828405	1	8.0	3,681.25	/tmp/comgr-a4204e/output/linked.bc-d15cf353.o.tmp
0.000023	0.028086	1,196.460280	1	8.0	3,681.25	/tmp/comgr-1c49b2/output/linked.bc-4f751122.o.tmp
0.000022	0.028086	1,270.214310	1	8.0	3,681.25	/tmp/comgr-34b1b2/output/linked.bc-f990c629.o.tmp
0.000021	0.028086	1,312.416291	1	8.0	3,681.25	/tmp/comgr-d935c2/output/linked.bc-30fbf309.o.tmp

AMD Rocm Code Object Manager

DEFAULT REPORT

GPU stats!

Table 7: Time and Bytes Transferred for Accelerator Regions

Time%	Time	Acc Time%	Acc Time	Acc Copy Out (MiBytes)	Events	Calltree Accelerator ID PE=HIDE
100.0%	21.962528	100.0%	4.68	36.00	122,236	Total
100.0%	21.961887	100.0%	4.68	36.00	122,236	main
68.3%	14.990899	76.9%	3.60	--	121,000	Faces::share
58.7%	12.885117	--	--	--	0	MPI_Waitall acc.0
7.2%	1.589945	--	--	--	22,000	hipStreamSynchronize acc.0
1.4%	0.302991	--	--	--	0	MPI_Isend acc.0
0.6%	0.137602	76.9%	3.60	--	33,000	hipLaunchKernel
0.3%	0.066883	28.5%	1.33	--	22,000	hipKernel.gpuRun3x1<> acc.0
0.2%	0.033095	48.4%	2.27	--	11,000	hipKernel.gpuRun4x1<> acc.0

DEFAULT REPORT

Table 7: Time and Bytes Transferred for Accelerator Regions

Time%	Time	Acc Time%	Acc Time	Acc Copy Out (MiBytes)	Events	Calltree Accelerator ID PE=HIDE
100.0%	21.962528	100.0%	4.68	36.00	122,236	Total
100.0%	21.961887	100.0%	4.68	36.00	122,236	main
68.3%	14.990899	76.9%	3.60	--	121,000	Faces::share
58.7%	12.885117	--	--	--	0	MPI_Waitall
						acc.0
7.2%	1.589945	--	--	--	22,000	hipStreamSynchronize
						acc.0
1.4%	0.302991	--	--	--	0	MPI_Isend
						acc.0
0.6%	0					hipLaunchKernel
0.3%					0	hipKernel.gpuRun3x1<>
						acc.0
0.2%	0.033095	48.4%	2.27	--	11,000	hipKernel.gpuRun4x1<>
						acc.0

But ordered by host runtime…

DEFAULT REPORT

Table 7: Time and Bytes Transferred for Accelerator Regions

Time%	Time	**Acc Time%**	**Acc Time**	Acc Copy Out (MiBytes)	Events	Calltree Accelerator ID PE=HIDE
100.0%	21.962528	100.0%	4.68	36.00	122,236	Total
100.0%	21.961887	100.0%	4.68	36.00	122,236	main
68.3%	14.990899	76.9%	3.60	--	121,000	Faces::share
58.7%	12.885117	--	--	--	0	MPI_Waitall
7.2%	1.589945	--	--	--		ze
1.4%	0.302991	--	--	--		
						acc.0
0.6%	0.137602	**76.9%**	**3.60**	--	**33,000**	**hipLaunchKernel**
0.3%	0.066883	**28.5%**	**1.33**	--	**22,000**	**hipKernel.gpuRun3x1<>**
						acc.0
0.2%	0.033095	**48.4%**	**2.27**	--	**11,000**	**hipKernel.gpuRun4x1<>**
						acc.0

Burying the lead

DEFAULT REPORT

Table 8: Program energy and power usage (from Cray PM)

Node Energy (J)	Node Power (W)	Process Time	PE=HIDE
19,665	888.012	22.144975	Total

Energy bill

DEFAULT REPORT

Table 9: Memory High Water Mark by Numa Node

Process HiMem (MiBytes)	HiMem Numa Node 0 (MiBytes)	HiMem Numa Node 1 (MiBytes)	HiMem Numa Node 2 (MiBytes)	HiMem Numa Node 3 (MiBytes)	Numanode PE=HIDE
321.6	245.4	27.0	24.8	24.5	numanode.0
320.7	23.7	248.3	23.5	25.3	numanode.1
321.4	23.8	25.8	246.3	25.4	numanode.2
320.6	22.6	26.7	24.5	246.9	numanode.3

...

Table 10: Wall Clock Time, Memory High Water Mark

Process Time	Process HiMem (MiBytes)	PE=[mmm]
22.144975	321.1	Total
22.145698	320.6	pe.4
22.145042	322.2	pe.5
22.143826	320.5	pe.6

CPU memory stuff

TELL ME MORE

```
$ pat_report -O -h
pat_report: Help for -O option:
Available option values are in left column, a prefix can be specified:

D1_D2_observation          D1 + D2 cache utilization
D1_D2_util                 Functions with low D1+D2 cache hit ratio
D1_observation             D1 cache utilization
D1_util                    Functions with low D1 cache hit ratio
O_data_movement            Data Movement
O_func_reg_profile         Function/Region Profile
O_load_imb                 Load Imbalance
O_memory_util              Memory Utilization
O_profile                  Profile
TLB_observation            TLB utilization
TLB_util                   Functions with low TLB refs/miss
_group_time_pct            Time% by Function Group
acc_fu                     Time and Bytes Transferred for Accelerator Regions
acc_fu_ht                  Time and Bytes Transferred for Accelerator Regions
acc_kern_stats             Kernel Stats for Accelerator Regions
...
```

ACC_TIME

Call tree ordered by GPU time!

```
$ pat_report -O acc_time <path>/<exe>+pat+<unique>t
...
Table 1:  Time and Bytes Transferred for Accelerator Regions

    Acc |  Acc |      Time |  Acc Copy |  Events | Calltree
  Time% | Time |           |       Out |         |  Accelerator ID
        |      |           | (MiBytes) |         |   PE=HIDE

 100.0% | 4.68 | 21.962528 |     36.00 | 122,236 | Total
|-----------------------------------------------------------------
| 100.0% | 4.68 | 21.961887 |     36.00 | 122,236 | main
||----------------------------------------------------------------
||  76.9% | 3.60 | 14.990899 |        -- | 121,000 | Faces::share
3|  76.9% | 3.60 |  0.137602 |        -- |  33,000 |  hipLaunchKernel
||||------------------------------------------------------------
4|||  48.4% | 2.27 |  0.033095 |        -- |  11,000 | hipKernel.gpuRun4x1<>
5|||       |      |           |           |         |  acc.0
4|||  28.5% | 1.33 |  0.066883 |        -- |  22,000 | hipKernel.gpuRun3x1<>
5|||       |      |           |           |         |  acc.0
```

ACC_TIME

Add line numbers?

```
$ pat_report -O acc_time -s show_ca=fu,so,li <path>/<exe>+pat+<unique>t
...
Table 1:  Time and Bytes Transferred for Accelerator Regions

    Acc |  Acc |      Time |  Acc Copy |  Events | Calltree
  Time% | Time |           |       Out |         |  Accelerator ID
        |      |           | (MiBytes) |         |   PE=HIDE

 100.0% | 4.68 | 21.962682 |     36.00 | 122,236 | Total
|-----------------------------------------------------------------------
|  76.9% | 3.60 | 14.991086 |        -- | 121,000 | main:work/ecpam23/base/main.cpp:line.173
||----------------------------------------------------------------------
||  48.4% | 2.27 |  0.063368 |        -- |  33,000 | Faces::share:ecpam23/base/./gpu.hpp:line.189
3|  48.4% | 2.27 |  0.045324 |        -- |  11,000 |  hipLaunchKernel:==NA==
4|  48.4% | 2.27 |  0.033095 |        -- |  11,000 |   hipKernel.gpuRun4x1<>
5|        |      |           |           |         |    acc.0
||  28.5% | 1.33 |  0.092279 |        -- |  22,000 | Faces::share:ecpam23/base/./gpu.hpp:line.152
3|        |      |           |           |         |  hipLaunchKernel:==NA==
4|  28.5% | 1.33 |  0.066883 |        -- |  22,000 |   hipKernel.gpuRun3x1<>
5|        |      |           |           |         |    acc.0
```

ACC_TIME

```
$ pat_report -O acc_time -s show_ca=fu,so,li <path>/<exe>+pat+<unique>t
...
Table 1:  Time and Bytes Transferred for Accelerator Regions

    Acc |  Acc |      Time |  Acc Copy |  Events | Calltree
  Time% | Time |           |       Out |         |  Accelerator ID
        |      |           | (MiBytes) |         |   PE=HIDE

 100.0% | 4.68 | 21.962682 |     36.00 | 122,236 | Total
|-----------------------------------------------------------------------
|  76.9% | 3.60 | 14.991086 |        -- | 121,000 | main:work/ecpam23/base/main.cpp:line.173
||----------------------------------------------------------------------
||  48.4% | 2.27 |  0.063368 |        -- |  33,000 | Faces::share:ecpam23/base/./gpu.hpp:line.189
3|  48.4% | 2.27 |  0.045324 |        -- |  11,000 |  hipLaunchKernel:==NA==
4|  48.4% | 2.27 |  0.033095 |        -- |  11,000 |   hipKernel.gpuRun4x1<>
5|        |      |           |           |         |    acc.0
||  28.5% | 1.33 |  0.092279 |        -- |  22,000 | Faces::share:ecpam23/base/./gpu.hpp:line.152
3|        |      |           |           |         |  hipLaunchKernel:==NA==
4|  28.5% | 1.33 |  0.066883 |        -- |  22,000 |   hipKernel.gpuRun3x1<>
5|        |      |           |           |         |    acc.0
```

Way up in main

ACC_TIME

```
$ pat_report -O acc_time -s show_ca=fu,so,li <path>/<exe>+pat+<unique>t
...
Table 1:  Time and Bytes Transferred for Accelerator Regions

    Acc |  Acc |      Time |  Acc Copy |  Events | Calltree
  Time% | Time |           |       Out |         |  Accelerator ID
        |      |           | (MiBytes) |         |   PE=HIDE

 100.0% | 4.68 | 21.962682 |     36.00 | 122,236 | Total
|-----------------------------------------------------------------------
|  76.9% | 3.60 | 14.991086 |        -- | 121,000 | main:work/ecpam23/base/main.cpp:line.173
||----------------------------------------------------------------------
||  48.4% | 2.27 |  0.063368 |        -- |  33,000 | Faces::share:ecpam23/base/./gpu.hpp:line.189
3|  48.4% | 2.27 |  0.045324 |        -- |  11,000 |  hipLaunchKernel:==NA==
4|  48.4% | 2.27 |  0.033095 |        -- |  11,000 |   hipKernel.gpuRun4x1<>
5|        |      |           |           |         |    acc.0
||  28.5% | 1.33 |  0.092279 |        -- |  22,000 | Faces::share:ecpam23/base/./gpu.hpp:line.152
3|        |      |           |           |         |  hipLaunchKernel:==NA==
4|  28.5% | 1.33 |  0.066883 |        -- |  22,000 |   hipKernel.gpuRun3x1<>
5|        |      |           |           |         |    acc.0
```

Lines in the "portability layer"

TEMPLATES, TEMPLATES, EVERY WHERE

- C++ codes often launch kernels from template functions that take lambda arguments
- "Real" kernel code is in the user-provided lambdas
- Think "portability layers", Kokkos, Raja, Alpaka, Yakl, *etc.*
- Templates get inlined into user code
- Profiles show line numbers somewhere inside portability layers
  instead of line numbers in user code where lambdas appear

WORKAROUND

- Turn off compiler inlining of top layer of portability-layer templates

```
template <typename F>
#ifdef CRAYPAT
__attribute__((noinline))
#endif
void gpuFor(const std::initializer_list<int> il0, F f,
  const hipStream_t stream = 0)
{
  ...
```

Defined by pat_opts pre_compile

- Minimal impact on runtime
  - Kernel launches are much more expensive than host function calls
- But changes are in portability layer, not user code

ACC_TIME WITH WORKAROUND

```
$ pat_report -O acc_time -s show_ca=fu,so,li <path>/<exe>+pat+<new unique>-<new unique>t
...
Table 1:  Time and Bytes Transferred for Accelerator Regions

    Acc |  Acc |      Time |  Acc Copy |  Events | Calltree
  Time% | Time |           |       Out |         |  Accelerator ID
        |      |           | (MiBytes) |         |   PE=HIDE

 100.0% | 4.75 | 21.796618 |     36.00 | 122,236 | Total
|-------------------------------------------------------------------------------
|  77.4% | 3.68 | 14.931328 |        -- | 121,000 | main:work/ecpam23/base/main.cpp:line.173
||------------------------------------------------------------------------------
||  48.2% | 2.29 |  0.064234 |        -- |  33,000 | Faces::share:work/ecpam23/base/Faces.cpp:line.243
3|        |      |           |           |         |  gpuFor<>:ecpam23/base/./gpu.hpp:line.204
4|  48.2% | 2.29 |  0.045265 |        -- |  11,000 |   hipLaunchKernel:==NA==
5|  48.2% | 2.29 |  0.032908 |        -- |  11,000 |    hipKernel.gpuRun4x1<>
6|        |      |           |           |         |     acc.0
||  24.8% | 1.18 | 12.445056 |        -- |  33,000 | Faces::share:work/ecpam23/base/Faces.cpp:line.326
3|  24.8% | 1.18 |  0.072299 |        -- |  33,000 |  gpuFor<>:ecpam23/base/./gpu.hpp:line.165
4|  24.8% | 1.18 |  0.047754 |        -- |  11,000 |   hipLaunchKernel:==NA==
5|  24.8% | 1.18 |  0.033855 |        -- |  11,000 |    hipKernel.gpuRun3x1<>
6|        |      |           |           |         |     acc.0
||   4.3% | 0.21 |  0.067396 |        -- |  33,000 | Faces::share:work/ecpam23/base/Faces.cpp:line.175
3|   4.3% | 0.21 |  0.067204 |        -- |  33,000 |  gpuFor<>:ecpam23/base/./gpu.hpp:line.165
4|   4.3% | 0.21 |  0.045301 |        -- |  11,000 |   hipLaunchKernel:==NA==
5|   4.3% | 0.21 |  0.032127 |        -- |  11,000 |    hipKernel.gpuRun3x1<>
6|        |      |           |           |         |     acc.0
```

Where the lambdas are!

APPRENTICE2

DOWNLOAD APPRENTICE2 TO YOUR LOCAL COMPUTER

```
$ ls -1 $CRAYPAT_ROOT/share/desktop_installers/App*
/opt/cray/pe/perftools/22.12.0/share/desktop_installers/Apprentice2
Installer-22.12.0-1.dmg
/opt/cray/pe/perftools/22.12.0/share/desktop_installers/Apprentice2
Installer-22.12.0-1.exe
```

RUN PAT_REPORT

- First run *pat_report* on each *<exe>+pat+<unique>t*
  - Adds *.ap2* files
- Otherwise *Apprentice2* will complain "Directory … contains no .ap2 files"
- If a *pat_report* is running long, *control-C* could cause incomplete *.ap2* files
  - Delete incomplete files and run *pat_report* again to completion

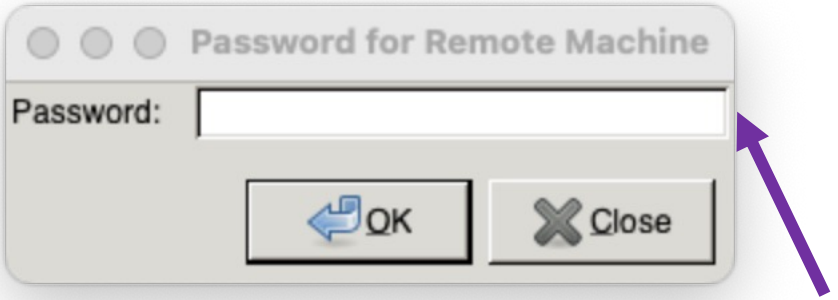
```
rm -rf <exe>+pat+<unique>t/*ap2*
pat_report <exe>+pat+<unique>t
```


About Apprentice2
File Compare View Help
CRAY
APPRENTICE2

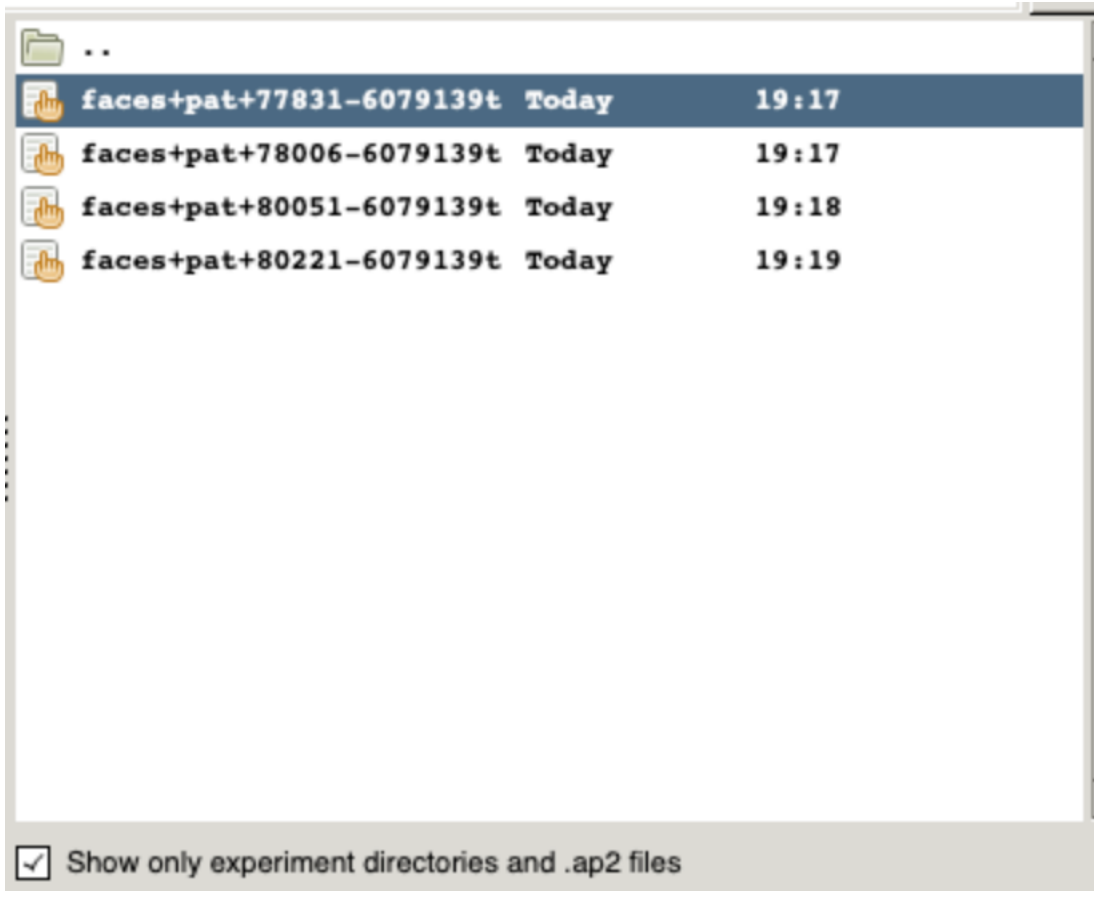
About Apprentice2

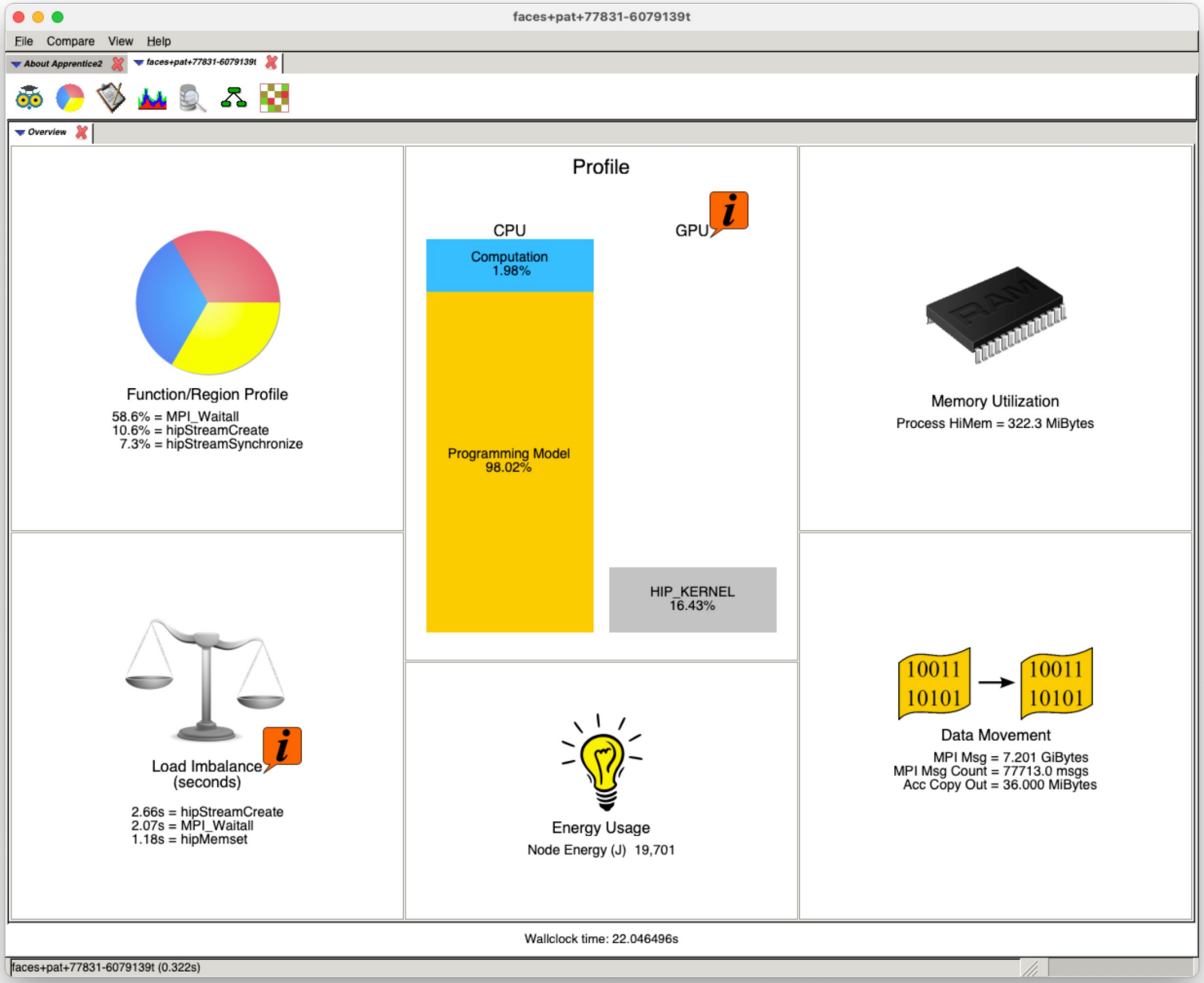
File Compare View Help

Open... ^O
Open Remote... ^R
Screendump...
Quit ^Q



Works with one-time passwords




faces+pat+77831-6079139t
File Compare View Help
About Apprentice2
faces+pat+77831-6079139t
Overview
Profile
CPU
GPU
Computation
1.98%
Programming Model
98.02%
HIP_KERNEL
16.43%
Function/Region Profile
58.6% = MPI_Waitall
10.6% = hipStreamCreate
7.3% = hipStreamSynchronize
Memory Utilization
Process HiMem = 322.3 MiBytes
Load Imbalance
(seconds)
2.66s = hipStreamCreate
2.07s = MPI_Waitall
1.18s = hipMemset
Energy Usage
Node Energy (J) 19,701
10011
10101
10011
10101
Data Movement
MPI Msg = 7.201 GiBytes
MPI Msg Count = 77713.0 msgs
Acc Copy Out = 36.000 MiBytes
Wallclock time: 22.046496s
faces+pat+77831-6079139t (0.322s)

Hewlett Packard Enterprise

Flashback!

See last year's tutorial for a tour of Apprentice2

PART 2: PERFORMANCE PROFILING FOR FRONTIER

Trey White

May 6, 2022

To start at the Apprentice2 *tour:*

https://youtu.be/Bw3CEo2eOUA?t=8490

https://youtu.be/Bw3CEo2eOUA

GPU PERFORMANCE COUNTERS

PAT_HELP

Drop period for interactive mode

```
$ pat_help counters mi200 .
> pat_help counters mi200 .
  Additional topics (case sensitive) that may follow "counters
mi200":

    deriv
    groups
    native
```

PAT_HELP

```
$ pat_help counters mi200 .
> pat_help counters mi200 .
  Additional topics (case sensitive) that may follow "counters
mi200":

    deriv
    groups
    native
```

These will be more useful soon

PAT_HELP

```
$ pat_help counters mi200 groups .
> pat_help counters mi200 groups .

    There are 4 predefined hardware performance counter event groups
    that can be specified by setting PAT_RT_PERFCTR to the group id.

  Additional topics (case sensitive) that may follow "counters mi200
groups":

    _busy_0         _stalled_0
    _mem_0          _write_0
$ pat_help counters mi200 groups _busy_0 .
> pat_help counters mi200 groups _busy_0 .

  rocm:::GPUBusy:device=0
  rocm:::MemUnitBusy:device=0
  rocm:::SALUBusy:device=0
  rocm:::VALUBusy:device=0
```

PAT_HELP

```
$ pat_help counters mi200 groups .
> pat_help counters mi200 groups .

    There are 4 predefined hardware performance counter event groups
    that can be specified by setting PAT_RT_PERFCTR to the group id.

  Additional topics (case sensitive) that may follow "counters mi200
groups":

    _busy_0         _stalled_0
    _mem_0          _write_0
$ pat_help counters mi200 groups _busy_0 .
> pat_help counters mi200 groups _busy_0 .

  rocm:::GPUBusy:device=0
  rocm:::MemUnitBusy:device=0
  rocm:::SALUBusy:device=0
  rocm:::VALUBusy:device=0
```

PAT_HELP

```
$ pat_help counters mi200 groups .
> pat_help counters mi200 groups .

    There are 4 predefined hardware performance counter event groups
    that can be specified by setting PAT_RT_PERFCTR to the group id.

  Additional topics (case sensitive) that may follow "counters mi200
groups":

    _busy_0         _stalled_0
    _mem_0          _write_0
$ pat_help counters mi200 groups _busy_0 .
> pat_help counters mi200 groups _busy_0 .

  rocm:::GPUBusy:device=0
  rocm:::MemUnitBusy:device=0
  rocm:::SALUBusy:device=0
  rocm:::VALUBusy:device=0
```

With --gpu-bind=closest,
you always want device=0

PAPI_NATIVE_AVAIL

Not currently working by default

```
$ papi_native_avail | grep -A1 rocm:::GPUBusy:device=0
$ export ROCP_METRICS="${ROCM_PATH}/lib/rocprofiler/metrics.xml"
$ papi_native_avail | grep -A1 rocm:::GPUBusy:device=0
| rocm:::GPUBusy:device=0                                                    |
|            The percentage of time GPU was busy.                            |
$ papi_native_avail | grep -A4 rocm:::MemUnitBusy:device=0
| rocm:::MemUnitBusy:device=0                                                |
|            The percentage of GPUTime the memory unit is active. The result in|
|            cludes the stall time (MemUnitStalled). This is measured with all |
|            extra fetches and writes and any cache or memory effects taken int|
|            o account. Value range: 0% to 100% (fetch-bound).               |
$ papi_native_avail | grep -A2 rocm:::SALUBusy:device=0
| rocm:::SALUBusy:device=0                                                   |
|            The percentage of GPUTime scalar ALU instructions are processed. V|
|            alue range: 0% (bad) to 100% (optimal).                         |
$ papi_native_avail | grep -A2 rocm:::VALUBusy:device=0
| rocm:::VALUBusy:device=0                                                   |
|            The percentage of GPUTime vector ALU instructions are processed. V|
|            alue range: 0% (bad) to 100% (optimal).                         |
```

PAPI_NATIVE_AVAIL

Temporary workaround

```
$ papi_native_avail | grep -A1 rocm:::GPUBusy:device=0
$ export ROCP_METRICS="${ROCM_PATH}/lib/rocprofiler/metrics.xml"
$ papi_native_avail | grep -A1 rocm:::GPUBusy:device=0
| rocm:::GPUBusy:device=0                                                    |
|            The percentage of time GPU was busy.                            |
$ papi_native_avail | grep -A4 rocm:::MemUnitBusy:device=0
| rocm:::MemUnitBusy:device=0                                                |
|            The percentage of GPUTime the memory unit is active. The result in|
|            cludes the stall time (MemUnitStalled). This is measured with all |
|            extra fetches and writes and any cache or memory effects taken int|
|            o account. Value range: 0% to 100% (fetch-bound).               |
$ papi_native_avail | grep -A2 rocm:::SALUBusy:device=0
| rocm:::SALUBusy:device=0                                                   |
|            The percentage of GPUTime scalar ALU instructions are processed. V|
|            alue range: 0% (bad) to 100% (optimal).                         |
$ papi_native_avail | grep -A2 rocm:::VALUBusy:device=0
| rocm:::VALUBusy:device=0                                                   |
|            The percentage of GPUTime vector ALU instructions are processed. V|
|            alue range: 0% (bad) to 100% (optimal).                         |
```

PAPI_NATIVE_AVAIL

```
$ papi_native_avail | grep -A1 rocm:::GPUBusy:device=0
$ export ROCP_METRICS="${ROCM_PATH}/lib/rocprofiler/metrics.xml"
$ papi_native_avail | grep -A1 rocm:::GPUBusy:device=0
| rocm:::GPUBusy:device=0                                                    |
|            The percentage of time GPU was busy.                            |
$ papi_native_avail | grep -A4 rocm:::MemUnitBusy:device=0
| rocm:::MemUnitBusy:device=0                                                |
|            The percentage of GPUTime the memory unit is active. The result in|
|            cludes the stall time (MemUnitStalled). This is measured with all |
|            extra fetches and writes and any cache or memory effects taken int|
|            o account. Value range: 0% to 100% (fetch-bound).               |
$ papi_native_avail | grep -A2 rocm:::SALUBusy:device=0
| rocm:::SALUBusy:device=0                                                   |
|            The percentage of GPUTime scalar ALU instructions are processed. V|
|            alue range: 0% (bad) to 100% (optimal).                         |
$ papi_native_avail | grep -A2 rocm:::VALUBusy:device=0
| rocm:::VALUBusy:device=0                                                   |
|            The percentage of GPUTime vector ALU instructions are processed. V|
|            alue range: 0% (bad) to 100% (optimal).                         |
```

PROFILE HIP

```
module load perftools-base/22.12.0
module load rocm/5.3.0
module load perftools
export LD_LIBRARY_PATH="${CRAY_LD_LIBRARY_PATH}:${LD_LIBRARY_PATH}"
export MPICH_GPU_SUPPORT_ENABLED=1
export OMP_NUM_THREADS=7
export PAT_RT_ACC_CID_TO_EID_BUFFER_SIZE=8MB
export PAT_RT_PERFCTR=_busy_0
srun ... -c 7 --gpus-per-task=1 --gpu-bind=closest <exe>+pat
```

DEFAULT REPORT

A new table!

Table 8: Profile by Function Group and Function

Group / Function / PE=HIDE

Total			
Time%		100.0%	
Time		166.002745	secs
Imb. Time		--	secs
Imb. Time%		--	
Calls	0.005M/sec	854,964.6	calls
GPUBusy:device=0		--	
MemUnitBusy:device=0		5.3%	
Acc Util		2.5%	
HIP			

DEFAULT REPORT

```
================================================================================
  HIP / hipKernel.gpuRun3x1<>
--------------------------------------------------------------------------------
  Time%                                   19.5%
  Time                                32.414519 secs
  Imb. Time                            7.768107 secs
  Imb. Time%                              22.1%
  Calls              678.708 /sec      22,000.0 calls
  GPUBusy:device=0                        71.2%
  MemUnitBusy:device=0                     5.3%
  Acc Util                                 0.4%
================================================================================
  HIP / hipKernel.gpuRun4x1<>
--------------------------------------------------------------------------------
  Time%                                    9.3%
  Time                                15.365194 secs
  Imb. Time                            1.927023 secs
  Imb. Time%                              12.7%
  Calls              715.904 /sec      11,000.0 calls
  GPUBusy:device=0                        71.2%
  MemUnitBusy:device=0                     5.3%
  Acc Util                                 1.4%
```

Entries for each kernel

DEFAULT REPORT

```
================================================================================
  HIP / hipKernel.gpuRun3x1<>
--------------------------------------------------------------------------------
  Time%                                   19.5%
  Time                                32.414519 secs
  Imb. Time                            7.768107 secs
  Imb. Time%                              22.1%
  Calls              678.708 /sec     22,000.0 calls
  GPUBusy:device=0                        71.2%
  MemUnitBusy:device=0                     5.3%
  Acc Util                                 0.4%
================================================================================
  HIP / hipKernel.gpuRun4x1<>
--------------------------------------------------------------------------------
  Time%                                   9.3%
  Time                                15.365194 secs
  Imb. Time                            1.927023 secs
  Imb. Time%                              12.7%
  Calls              715.904 /sec     11,000.0 calls
  GPUBusy:device=0                        71.2%
  MemUnitBusy:device=0                     5.3%
  Acc Util                                 1.4%
```

Why only 2 of the 4 counters?
(We're working on it.)

ACCPC_CT

```
$ pat_report -O accpc_ct -s show_ca=fu,so,li <exe>+pat+<unique>t
...
Table 1:  ACC Performance Counter Data Calltree

    Acc |   GPUBusy | MemUnitBusy |  Acc | Calltree
  Time% | :device=0 |   :device=0 | Util |  PE=HIDE

 100.0% |        -- |        5.3% | 2.5% | Total
|-------------------------------------------------------------------------
|  73.7% |     71.2% |        5.3% | 1.8% | main:work/ecpam23/base/main.cpp:line.173
||------------------------------------------------------------------------
||  57.7% |     71.2% |        5.3% | 1.4% | Faces::share:work/ecpam23/base/Faces.cpp:line.243
3|        |           |             |      |  gpuFor<>:ecpam23/base/./gpu.hpp:line.204
4|        |           |             |      |   hipLaunchKernel:==NA==
5|        |           |             |      |    hipKernel.gpuRun4x1<>
||  11.4% |     71.2% |        4.9% | 0.3% | Faces::share:work/ecpam23/base/Faces.cpp:line.326
3|        |           |             |      |  gpuFor<>:ecpam23/base/./gpu.hpp:line.165
4|        |           |             |      |   hipLaunchKernel:==NA==
5|        |           |             |      |    hipKernel.gpuRun3x1<>
||   4.6% |     71.2% |        5.3% | 0.1% | Faces::share:work/ecpam23/base/Faces.cpp:line.175
3|        |           |             |      |  gpuFor<>:ecpam23/base/./gpu.hpp:line.165
4|        |           |             |      |   hipLaunchKernel:==NA==
5|        |           |             |      |    hipKernel.gpuRun3x1<>
```

ACCPC_CT

```
$ pat_report -O accpc_ct -s show_ca=fu,so,li <exe>+pat+<unique>t
...
Table 1:  ACC Performance Counter Data Calltree

    Acc |   GPUBusy | MemUnitBusy |  Acc | Calltree
  Time% | :device=0 |   :device=0 | Util |  PE=HIDE

 100.0% |        -- |        5.3% | 2.5% | Total
|-------------------------------------------------------------------------
|  73.7% |     71.2% |        5.3% | 1.8% | main:work/ecpam23/base/main.cpp:line.173
||------------------------------------------------------------------------
||  57.7% |     71.2% |        5.3% | 1.4% | Faces::share:work/ecpam23/base/Faces.cpp:line.243
3|        |           |             |      |  gpuFor<>:ecpam23/base/./gpu.hpp:line.204
4|        |           |             |      |   hipLaunchKernel:==NA==
5|        |           |             |      |    hipKernel.gpuRun4x1<>
||  11.4% |     71.2% |        4.9% | 0.3% | Faces::share:work/ecpam23/base/Faces.cpp:line.326
3|        |           |             |      |  gpuFor<>:ecpam23/base/./gpu.hpp:line.165
4|        |           |             |      |   hipLaunchKernel:==NA==
5|        |           |             |      |    hipKernel.gpuRun3x1<>
||   4.6% |     71.2% |        5.3% | 0.1% | Faces::share:work/ecpam23/base/Faces.cpp:line.175
3|        |           |             |      |  gpuFor<>:ecpam23/base/./gpu.hpp:line.165
4|        |           |             |      |   hipLaunchKernel:==NA==
5|        |           |             |      |    hipKernel.gpuRun3x1<>
```

2 new columns (not yet 4)

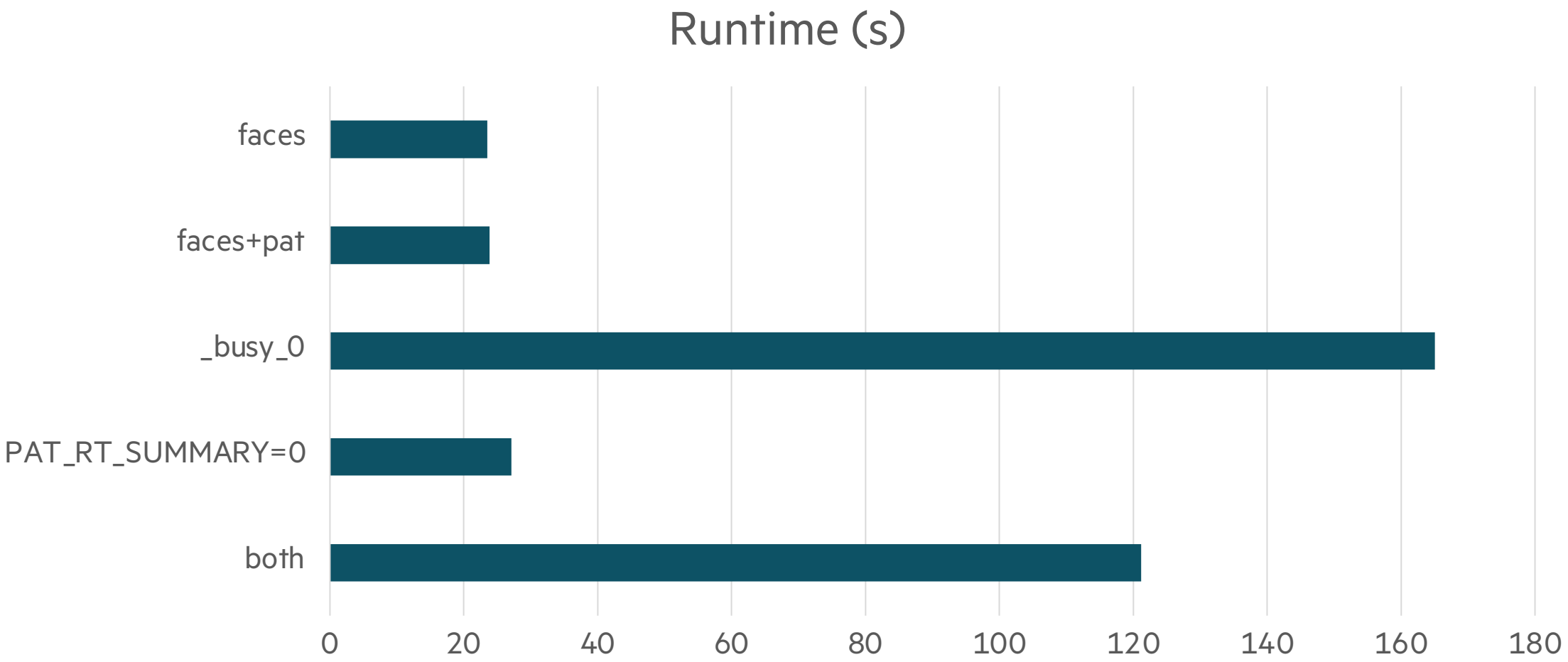
ACCPC_CT

```
$ pat_report -O accpc_ct -s show_ca=fu,so,li <exe>+pat+<unique>t
...
Table 1:  ACC Performance Counter Data Calltree

    Acc |   GPUBusy | MemUnitBusy |  Acc | Calltree
  Time% | :device=0 |   :device=0 | Util |  PE=HIDE

 100.0% |        -- |        5.3% | 2.5% | Total
|-------------------------------------------------------------------------
|  73.7% |     71.2% |        5.3% | 1.8% | main:work/ecpam23/base/main.cpp:line.173
||------------------------------------------------------------------------
||  57.7% |     71.2% |        5.3% | 1.4% | Faces::share:work/ecpam23/base/Faces.cpp:line.243
3|        |           |             |      |  gpuFor<>:ecpam23/base/./gpu.hpp:line.204
4|        |           |             |      |   hipLaunchKernel:==NA==
5|        |           |             |      |    hipKernel.gpuRun4x1<>
||  11.4% |     71.2% |        4.9% | 0.3% | Faces::share:work/ecpam23/base/Faces.cpp:line.326
3|        |           |             |      |  gpuFor<>:ecpam23/base/./gpu.hpp:line.165
4|        |           |             |      |   hipLaunchKernel:==NA==
5|        |           |             |      |    hipKernel.gpuRun3x1<>
||   4.6% |     71.2% |        5.3% | 0.1% | Faces::share:work/ecpam23/base/Faces.cpp:line.175
3|        |           |             |      |  gpuFor<>:ecpam23/base/./gpu.hpp:line.165
4|        |           |             |      |   hipLaunchKernel:==NA==
5|        |           |             |      |    hipKernel.gpuRun3x1<>
```

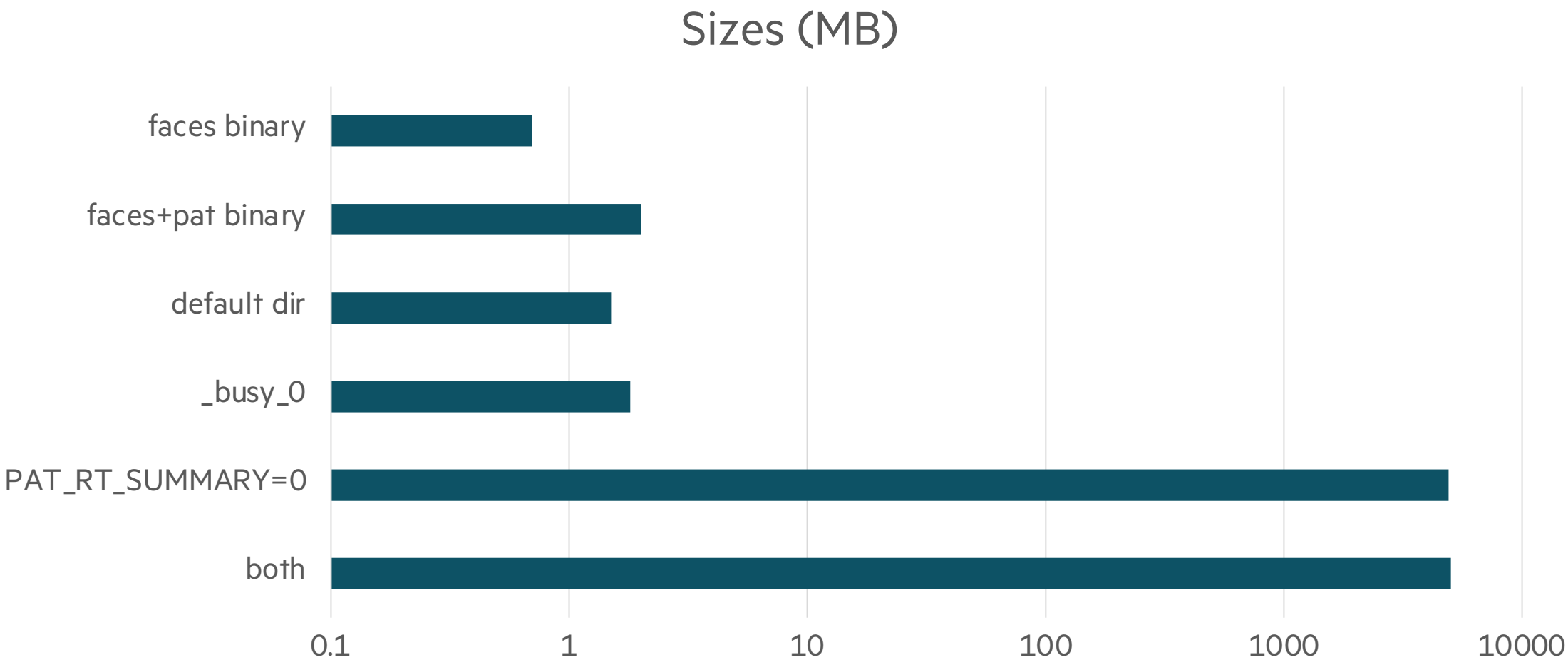
Percent of total time GPU is busy
(This is MPI-dominated code)

ALL THIS CAN BE YOURS FOR THE LOW LOW PRICE OF...

PERFTOOLS COSTS



PERFTOOLS COSTS



WANT TO PROFILE LIMITED SECTIONS OF CODE?

- Add include file

```
#ifdef CRAYPAT
#include "pat_api.h"
#endif
```

- Turn off profiling at the beginning of main

```
int main(int argc, char *argv[])
{
#ifdef CRAYPAT
  PAT_record(PAT_STATE_OFF);
#endif
```

- Turn profiling on just for regions of interest

```
#ifdef CRAYPAT
  PAT_record(PAT_STATE_ON);
#endif
... // Important code
#ifdef CRAYPAT
  PAT_record(PAT_STATE_OFF);
#endif
```

MORE API

```
$ man pat_build
/application program
 APPLICATION PROGRAM INTERFACE (API)
       The CrayPat API calls are functions that can be
       inserted into source code that write special tracing
       records into the experiment data file at runtime.

       Note: After API calls are inserted in the source code,
       the environment variable PAT_RT_TRACE_API can toggle
       data collection on or off. For more information, see
       the intro_craypat(1) man page.

       API calls are supported in both Fortran and C. After
       the perftools-base module is loaded, the include files
       that define the CrayPat API can be found in the
       ...
```

DETECTIVE STORY

The Case of the "Slow MPI"

WHODUNIT?

- Internal timers showed unusually long times for point-to-point communication
- MPI environment variables didn't help
- *pat_report* showed excessive time in *hipMemcpy*
- Changed to GPU-aware MPI
- Overall runtime barely changed
- *pat_report* showed excessive time in *hipDeviceSynchronize*

I ACCUSE...

- *pat_report -O acc_time* showed the real culprit!
- *PAT_RT_PERFCTR=_busy_0* showed the suspect had *MemUnitBusy* 94.1%
- No line profiling, but code inspection showed the "weapon"
- Tuning (reforming?) the culprit improved overall runtime by 2.5x

I ACCUSE...

- *pat_report -O acc_time* showed the real culprit!
- *PAT_RT_PERFCTR=_busy_0* showed the suspect had *MemUnitBusy* 94.1%
- No line profiling, but code inspection showed the "weapon"
- Tuning (reforming?) the culprit improved overall runtime by 2.5x

Kernel Matvec in the level-one cache with the strided loads!

SOLVING YOUR OWN MYSTERIES

DETECTING PERFORMANCE

- Match your *rocm* and *perftools-base* modules
- *module load perftools*
- Use *cce* compilers or *hipcc* with *pat_opts*
- *pat_build -g hip,io,mpi,omp -w -f <exe>*
- Load the same modules when you run
- Set *PAT_RT_ACC_CID_TO_EID_BUFFER_SIZE > 1MB* if necessary
- *pat_report -O acc_time,ct -s show_ca=fu,so,li <exe>+pat+<unique>t*

DETECTING PERFORMANCE

- Match your *rocm* and *perftools-base* modules
- *module load perftools*
- Use *cce* compilers or *hipcc* with *pat_opts*
- *pat_build -g hip,io,mpi,omp -w -f <exe>*
- Load the same modules when you run
- Set *PAT_RT_ACC_CID_TO_EID_BUFFER_SIZE > 1MB* if necessary
- *pat_report -O acc_time,ct -s show_ca=fu,so,li <exe>+pat+<unique>t*

Questions?