Argonne NATIONAL LABORATORY

Cray Performance Tools

Colleen Bertoni
Performance Engineering group

www.anl.gov

Outline

- Quick introduction to terminology
- Overview of Cray Performance Tools
- Introduction to CrayPat-lite
    - Usage and output
    - Learning more without rerunning
- Experiments to try

Quick intro to profiling terminology

- Goal of profiling is to determine where a program is spending time and help tune code to the target system

Profiling by sampling

- At given intervals, samples the program counters to see what function the code is in
- Statistical, might miss some functions with small runtime
- Usually lower overhead than other methods

Tracing by instrumentation

- Logs timeline information on enter and exit of function (traces function calls)
- More accurate about the program execution, but more overhead since each function is instrumented

Overview of Cray Performance Tools

- CrayPat: Cray’s **P**erformance **A**nalysis **T**ool
- Used to analyze program performance on Cray Supercomputers
- Both sampling and tracing modes
- Supports a wide range of programming models (including MPI, OpenMP, and CUDA)
- Can be used to identify hotspots, load imbalances, MPI rank communication information, I/O, and memory usage
- Supports Cray, Intel, and GCC compilers

Cray Performance Tools Components

- **CrayPat-lite and CrayPat**
  - Can be used to identify hotspots, load imbalances, MPI rank communication information, I/O, and memory usage, etc.
- **Cray Apprentice2**
  - Visualization of load imbalance, communication
- **Reveal**
  - Tool to show Cray compiler feedback about where to vectorize, and advice on modifying code to add OpenMP
  - module PrgEnv-cray should be used
- **PAPI**
  - Standard API for performance counters
  - Can set CrayPat environment variables to monitor and display hardware counters

CrayPat-lite

- Very easy-to-use version of CrayPat
- Simple interface
- Default is sampling-based profiling
- Automatically does data processing and generation of text report on the compute nodes at the end of a job

Using CrayPat-lite: Overview

1. Environment setup

```
user@thetalogin6:~> module unload darshan
user@thetalogin6:~> module load perftools-lite
```

2. Compiling the code to use CrayPat-lite

```
user@thetalogin6:~> make
```

3. Running the code

```
user@thetalogin6:~> qsub ./jobscript
```

4. Analyzing the output

```
Condensed report to stdout
```

```
pat_report -O lite-samples data_directory
```

Using CrayPat-lite: Overview

1. Environment setup

```
user@thetalogin6:~> module unload darshan
user@thetalogin6:~> module load perftools-lite
```

2. Compiling the code to use CrayPat-lite

```
user@thetalogin6:~> make
```

3. Running the code

```
user@thetalogin6:~> qsub ./jobscript
```

4. Analyzing the output

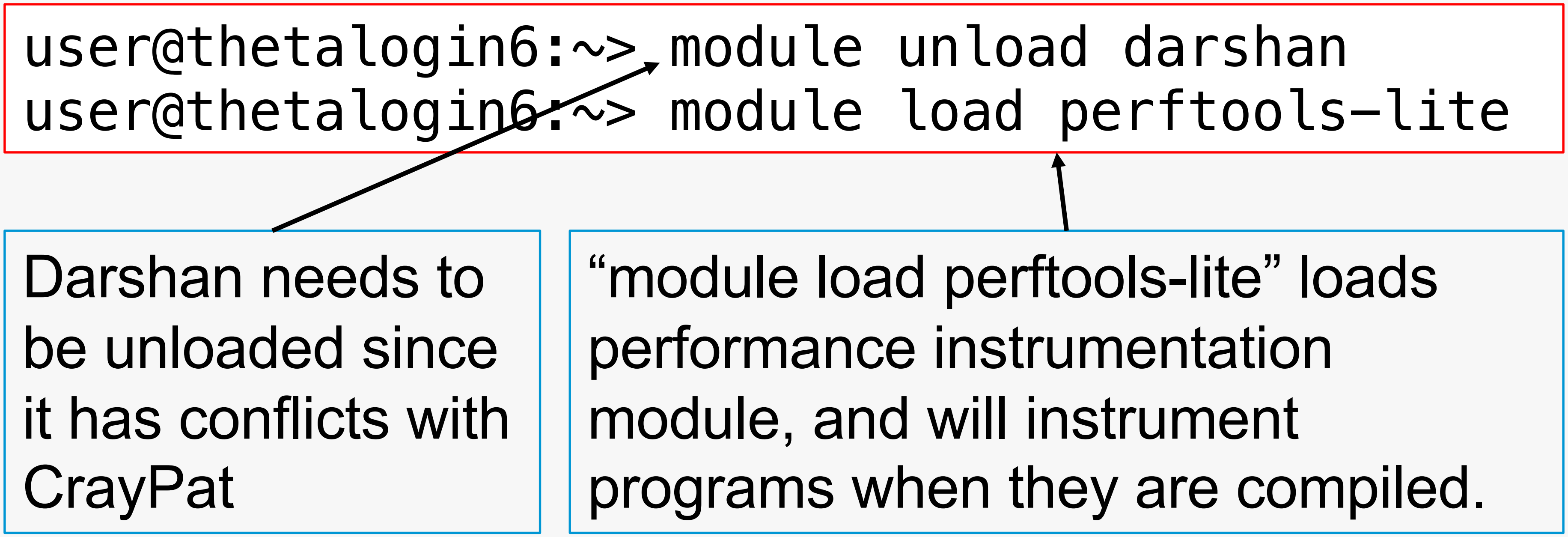
```
Condensed report to stdout
```

More information with pat_report or Apprentice2 without re-running

Document results, make changes, repeat

Using CrayPat-lite

1. Environment Setup



```
user@thetalogin6:~> module unload darshan
user@thetalogin6:~> module load perftools-lite
```

Darshan needs to be unloaded since it has conflicts with CrayPat

“module load perftools-lite” loads performance instrumentation module, and will instrument programs when they are compiled.

perftools-base module should already be listed under “module list” If not, load perftools base with “module load perftools-base”. This provides access to man pages and help system, and does not instrument code. (Should be loaded by default.)

Using CrayPat-lite

1. Environment Setup

```
user@thetalogin6:~> module unload darshan
user@thetalogin6:~> module load perftools-lite
```

```
user@thetalogin6:~> module list
Currently Loaded Modulefiles:
  1) modules/3.2.11.1                       13) dvs/2.7_2.2.118-6.0.7.1_10.1
  2) intel/18.0.0.128                       14) alps/6.6.43-6.0.7.1_5.45.ari
  3) craype-network-aries                   15) rca/2.2.18-6.0.7.1_5.47.ari
  4) craype/2.5.15                          16) atp/2.1.3
  5) cray-libsci/18.07.1                    17) perftools-base/7.0.4
  6) udreg/2.3.2-6.0.7.1_5.13               18) PrgEnv-intel/6.0.4
  7) ugni/6.0.14.0-6.0.7.1_3.13             19) craype-mic-knl
  8) pmi/5.0.14                             20) cray-mpich/7.7.3
  9) dmapp/7.1.1-6.0.7.1_5.45__             21) nompirun/nompirun
 10) gni-headers/5.0.12.0-6.0.7.1_3.11      22) trackdeps
 11) xpmem/2.2.15-6.0.7.1_5.11__            23) xalt
 12) job/2.2.3-6.0.7.1_5.43__               24) perftools-lite
```

Using CrayPat-lite

2. Compiling the code to use CrayPat-lite

(Any build script, not just make)

```
user@thetalogin6:~> make clean
user@thetalogin6:~> make
```

Rebuild code as usual

```
user@thetalogin6:~> make
ftn -O3 -qopt-report=5  -align array64byte -c test.f90 -o test.o
...
...
INFO: creating the CrayPat-instrumented executable 'test' (lite-
samples) ...OK
```

Using CrayPat-lite

3. Running the code

```
user@thetalogin6:~> qsub –A proj –n 128 ./jobscript.sh
Job routed to queue "default".
229865
```

Run code as usual

```
user@thetalogin6:~> cat jobscript.sh
#!/bin/bash –x
#COBALT --attrs mcdram=cache:numa=quad
#COBALT –t 30
#COBALT –n 128
#COBALT –A proj

echo "Starting Cobalt job script"
aprun –n 8192 –N 64 test
```

Using CrayPat-lite

4. Analyzing the output

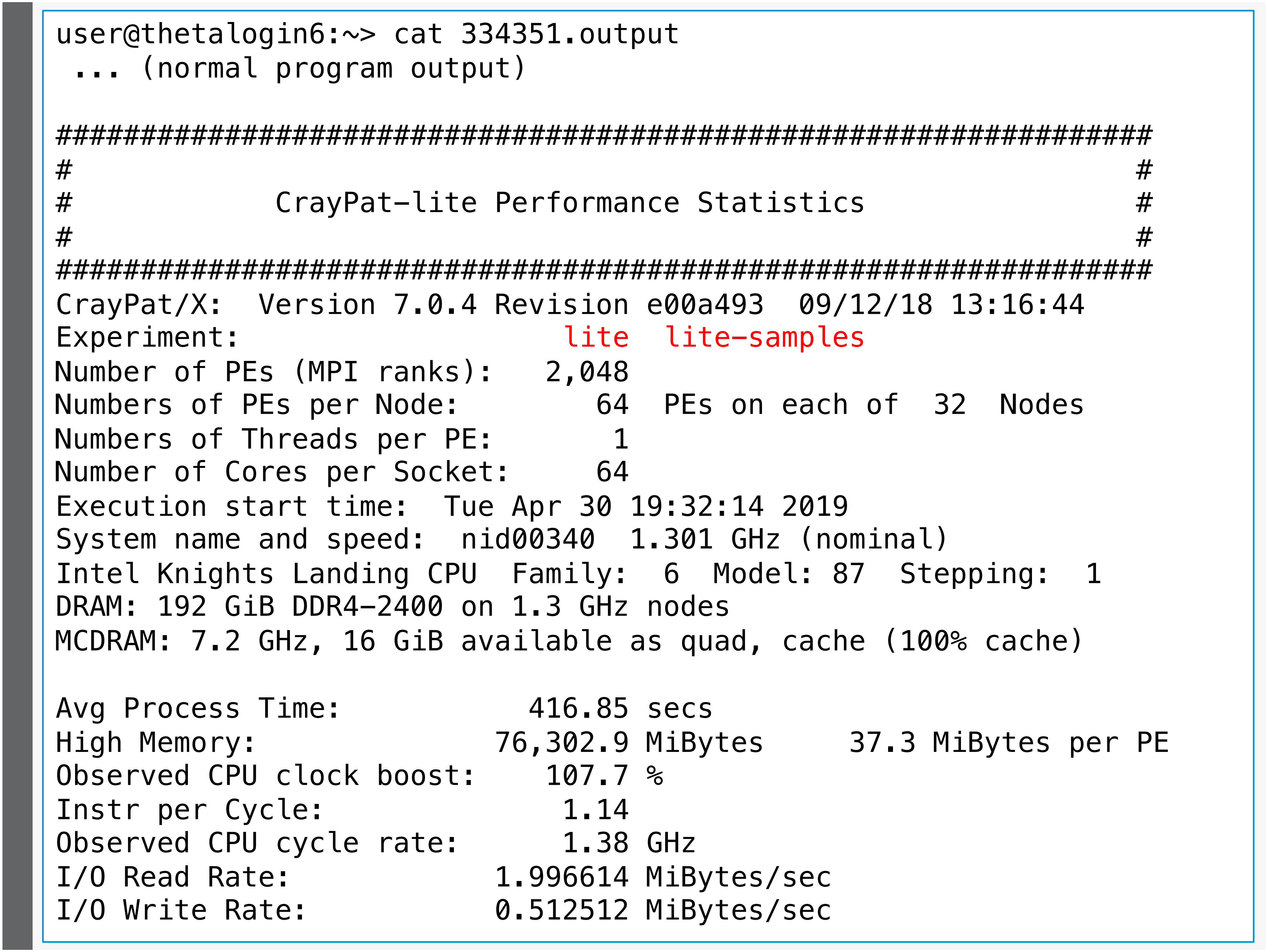
Condensed report to stdout (likely at the end of the Cobalt .output file), and more information in *.ap2, *.xf files and possibly a MPICH_RANK_ORDER file in a created subdirectory

```
user@thetalogin6:~> ls -ltr
total 230912
-rw-r--r--  1 user users  60192869 Apr 30 18:19 test
-rwxr-xr-x  1 user users  54547880 Apr 30 18:19 test+orig
-rw-r--r--  1 user users       303 Apr 30 19:32 334351.error
-rw-r--r--  1 user users     12826 Apr 30 19:43 334351.output
-rw-r--r--  1 user cobalt     2188 Apr 30 19:43 334351.cobaltlog
drwxr-x---  5 user users       512 Apr 30 22:25 test+42377-340s
```

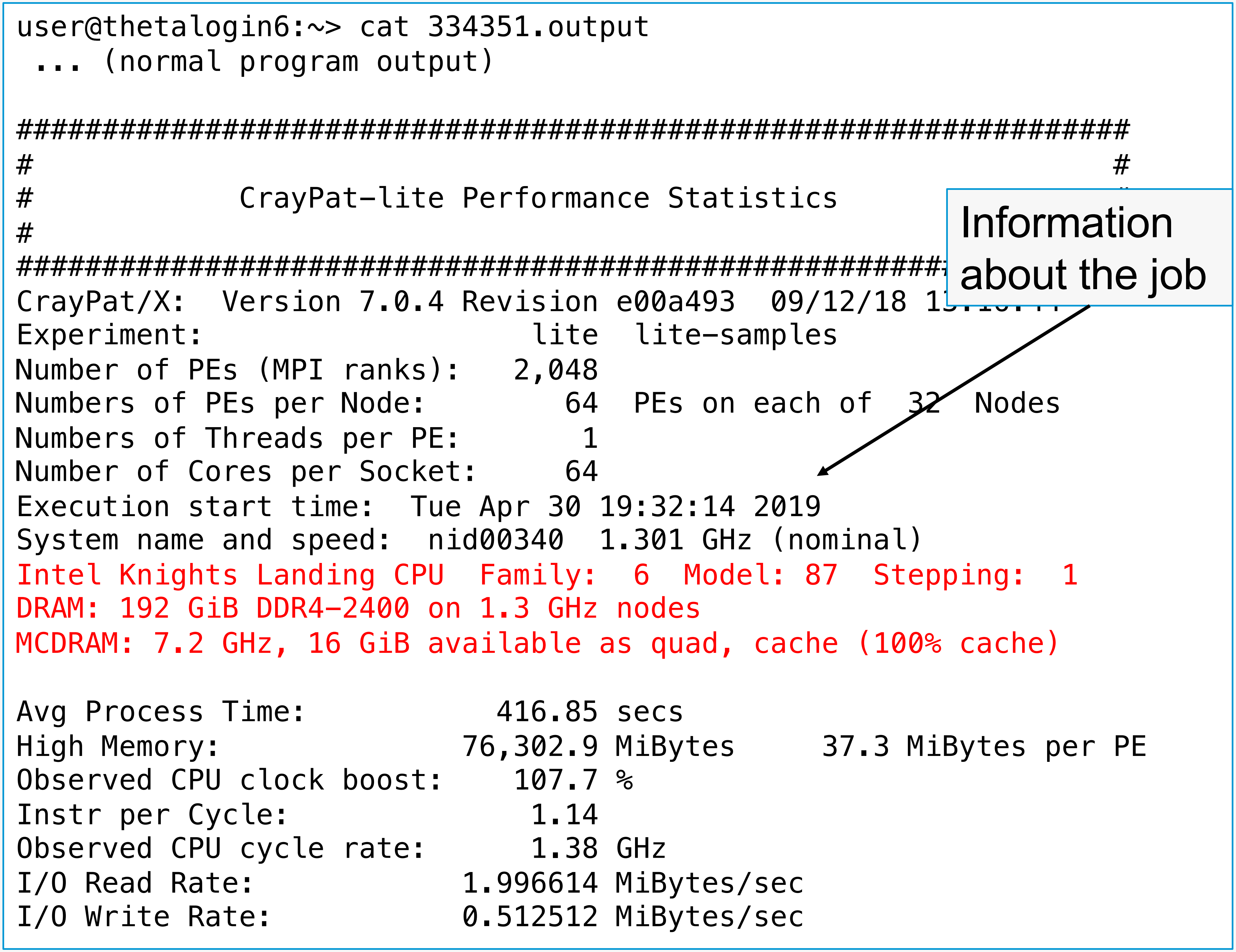
Condensed text report in .output file

Directory with additional files for analysis

```
user@thetalogin6:~> cat 334351.output
 ... (normal program output)

#################################################################
#                                                               #
#            CrayPat-lite Performance Statistics                #
#                                                               #
#################################################################
CrayPat/X:  Version 7.0.4 Revision e00a493  09/12/18 13:16:44
Experiment:                   lite  lite-samples
Number of PEs (MPI ranks):   2,048
Numbers of PEs per Node:        64  PEs on each of  32  Nodes
Numbers of Threads per PE:       1
Number of Cores per Socket:     64
Execution start time:  Tue Apr 30 19:32:14 2019
System name and speed:  nid00340  1.301 GHz (nominal)
Intel Knights Landing CPU  Family:  6  Model: 87  Stepping:  1
DRAM: 192 GiB DDR4-2400 on 1.3 GHz nodes
MCDRAM: 7.2 GHz, 16 GiB available as quad, cache (100% cache)

Avg Process Time:           416.85 secs
High Memory:              76,302.9 MiBytes     37.3 MiBytes per PE
Observed CPU clock boost:    107.7 %
Instr per Cycle:              1.14
Observed CPU cycle rate:      1.38 GHz
I/O Read Rate:            1.996614 MiBytes/sec
I/O Write Rate:           0.512512 MiBytes/sec
```

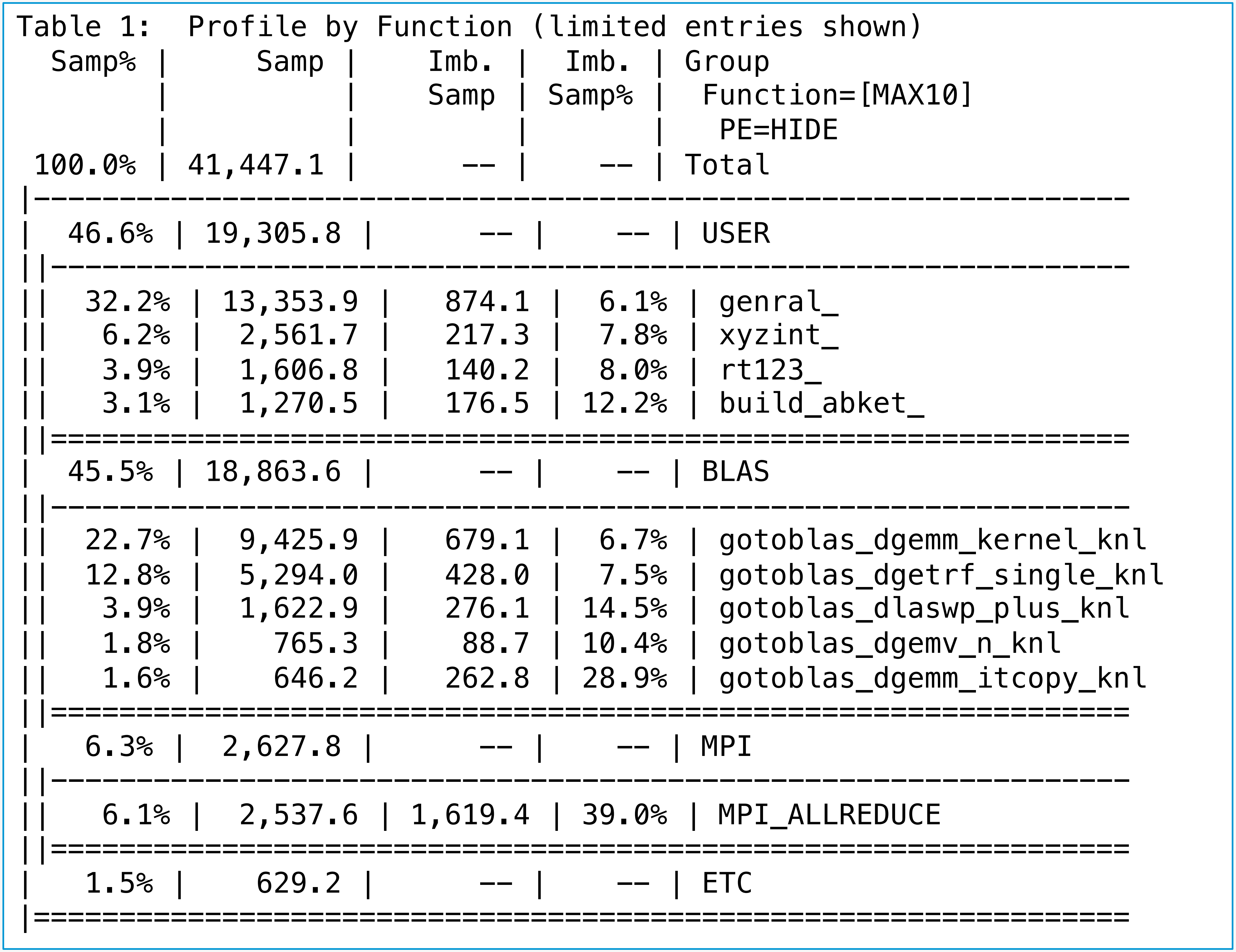


```
user@thetalogin6:~> cat 334351.output
 ... (normal program output)

#################################################################
#                                                               #
#            CrayPat-lite Performance Statistics                #
#                                                               #
#################################################################
CrayPat/X:  Version 7.0.4 Revision e00a493  09/12/18 13:16:44
Experiment:                   lite  lite-samples
Number of PEs (MPI ranks):   2,048
Numbers of PEs per Node:        64  PEs on each of  32  Nodes
Numbers of Threads per PE:       1
Number of Cores per Socket:     64
Execution start time:  Tue Apr 30 19:32:14 2019
System name and speed:  nid00340  1.301 GHz (nominal)
Intel Knights Landing CPU  Family:  6  Model: 87  Stepping:  1
DRAM: 192 GiB DDR4-2400 on 1.3 GHz nodes
MCDRAM: 7.2 GHz, 16 GiB available as quad, cache (100% cache)

Avg Process Time:           416.85 secs
High Memory:              76,302.9 MiBytes     37.3 MiBytes per PE
Observed CPU clock boost:    107.7 %
Instr per Cycle:              1.14
Observed CPU cycle rate:      1.38 GHz
I/O Read Rate:            1.996614 MiBytes/sec
I/O Write Rate:           0.512512 MiBytes/sec
```

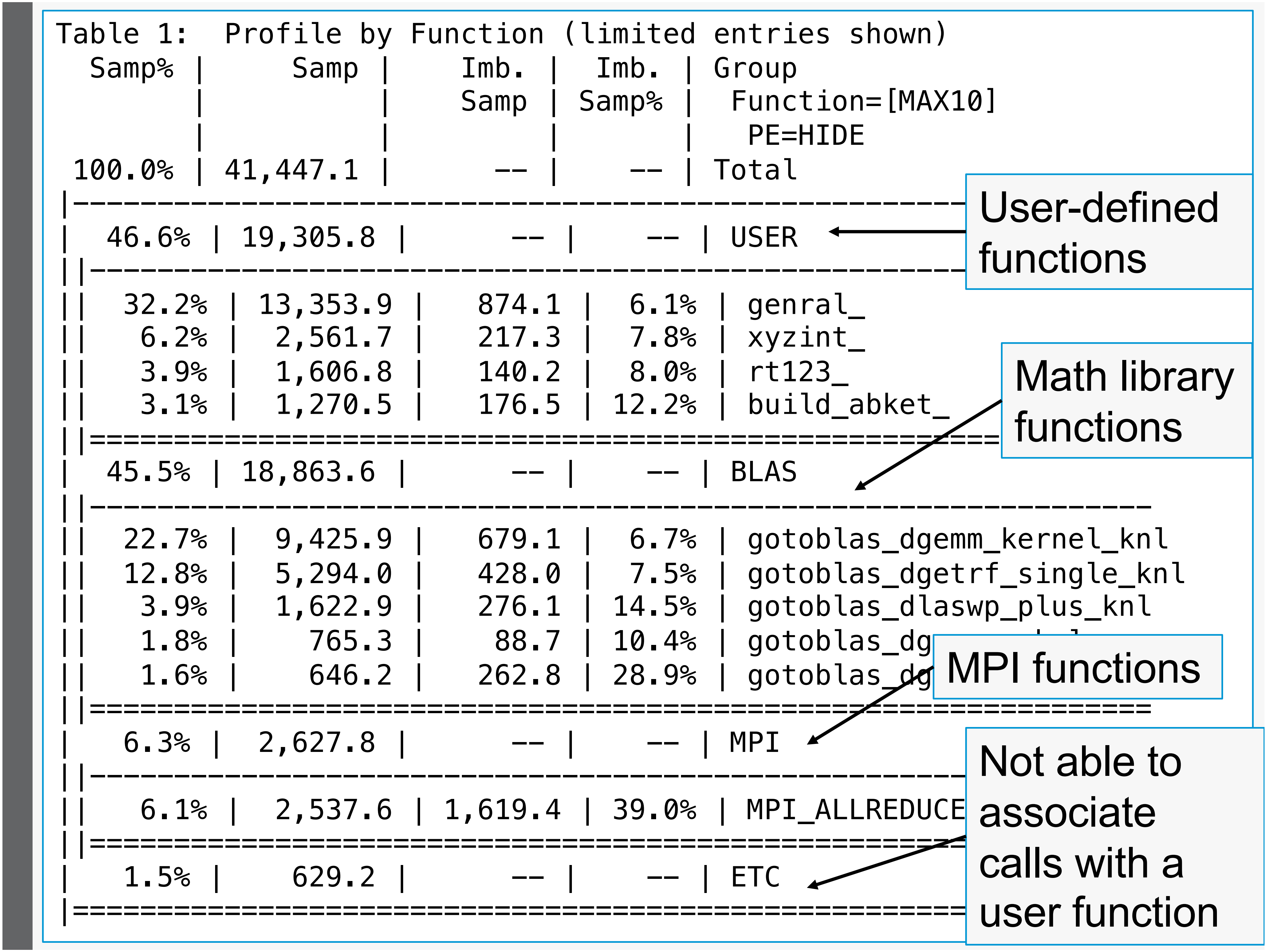


```
user@thetalogin6:~> cat 334351.output
 ... (normal program output)

################################################################
#                                                              #
#            CrayPat-lite Performance Statistics               #
#                                                              #
################################################################
CrayPat/X:  Version 7.0.4 Revision e00a493  09/12/18 13:16:44
Experiment:                     lite  lite-samples
Number of PEs (MPI ranks):   2,048
Numbers of PEs per Node:        64  PEs on each of  32  Nodes
Numbers of Threads per PE:       1
Number of Cores per Socket:     64
Execution start time:  Tue Apr 30 19:32:14 2019
System name and speed:  nid00340  1.301 GHz (nominal)
Intel Knights Landing CPU  Family:  6  Model: 87  Stepping:  1
DRAM: 192 GiB DDR4-2400 on 1.3 GHz nodes
MCDRAM: 7.2 GHz, 16 GiB available as quad, cache (100% cache)

Avg Process Time:             416.85 secs
High Memory:                76,302.9 MiBytes     37.3 MiBytes per PE
Observed CPU clock boost:      107.7 %
Instr per Cycle:                1.14
Observed CPU cycle rate:        1.38 GHz
I/O Read Rate:              1.996614 MiBytes/sec
I/O Write Rate:             0.512512 MiBytes/sec
```

Information about the job



```
Table 1:  Profile by Function (limited entries shown)
  Samp% |     Samp |    Imb. |  Imb. | Group
        |          |    Samp | Samp% |  Function=[MAX10]
        |          |         |       |   PE=HIDE
 100.0% | 41,447.1 |      -- |    -- | Total
|----------------------------------------------------------------
|  46.6% | 19,305.8 |      -- |    -- | USER
||---------------------------------------------------------------
||  32.2% | 13,353.9 |   874.1 |  6.1% | genral_
||   6.2% |  2,561.7 |   217.3 |  7.8% | xyzint_
||   3.9% |  1,606.8 |   140.2 |  8.0% | rt123_
||   3.1% |  1,270.5 |   176.5 | 12.2% | build_abket_
||===============================================================
|  45.5% | 18,863.6 |      -- |    -- | BLAS
||---------------------------------------------------------------
||  22.7% |  9,425.9 |   679.1 |  6.7% | gotoblas_dgemm_kernel_knl
||  12.8% |  5,294.0 |   428.0 |  7.5% | gotoblas_dgetrf_single_knl
||   3.9% |  1,622.9 |   276.1 | 14.5% | gotoblas_dlaswp_plus_knl
||   1.8% |    765.3 |    88.7 | 10.4% | gotoblas_dgemv_n_knl
||   1.6% |    646.2 |   262.8 | 28.9% | gotoblas_dgemm_itcopy_knl
||===============================================================
|   6.3% |  2,627.8 |      -- |    -- | MPI
||---------------------------------------------------------------
||   6.1% |  2,537.6 | 1,619.4 | 39.0% | MPI_ALLREDUCE
||===============================================================
|   1.5% |    629.2 |      -- |    -- | ETC
|================================================================
```



```
Table 1:  Profile by Function (limited entries shown)
  Samp% |      Samp |     Imb. |  Imb. | Group
        |           |     Samp | Samp% |  Function=[MAX10]
        |           |          |       |   PE=HIDE
 100.0% | 41,447.1 |       -- |    -- | Total
|-----------------------------------------------------
|  46.6% | 19,305.8 |       -- |    -- | USER
||----------------------------------------------------
||  32.2% | 13,353.9 |   874.1 |  6.1% | genral_
||   6.2% |  2,561.7 |   217.3 |  7.8% | xyzint_
||   3.9% |  1,606.8 |   140.2 |  8.0% | rt123_
||   3.1% |  1,270.5 |   176.5 | 12.2% | build_abket_
||====================================================
|  45.5% | 18,863.6 |       -- |    -- | BLAS
||-------------------------------------------------------------
||  22.7% |  9,425.9 |   679.1 |  6.7% | gotoblas_dgemm_kernel_knl
||  12.8% |  5,294.0 |   428.0 |  7.5% | gotoblas_dgetrf_single_knl
||   3.9% |  1,622.9 |   276.1 | 14.5% | gotoblas_dlaswp_plus_knl
||   1.8% |    765.3 |    88.7 | 10.4% | gotoblas_dg
||   1.6% |    646.2 |   262.8 | 28.9% | gotoblas_dg
||=============================================================
|   6.3% |  2,627.8 |       -- |    -- | MPI
||----------------------------------------------------
||   6.1% |  2,537.6 | 1,619.4 | 39.0% | MPI_ALLREDUCE
||====================================================
|   1.5% |    629.2 |       -- |    -- | ETC
|=====================================================
```

User-defined functions

Math library functions

MPI functions

Not able to associate calls with a user function

```
Table 1:  Profile by Function (limited entries shown)

  Samp% |     Samp |    Imb. |  Imb. | Group
        |          |    Samp | Samp% |  Function=[MAX10]
        |          |         |       |   PE=HIDE

 100.0% | 41,447.1 |      -- |    -- | Total
|-----------------------------------------------------------------
|  46.6% | 19,305.8 |      -- |    -- | USER
||----------------------------------------------------------------
||  32.2% | 13,353.9 |   874.1 |  6.1% | genral_
||   6.2% |  2,561.7 |   217.3 |  7.8% | xyzint_
||   3.9% |  1,606.8 |   140.2 |  8.0% | rt123_
||   3.1% |  1,270.5 |   176.5 | 12.2% | build_abket_
||================================================================
|  45.5% | 18,863.6 |      -- |    -- | BLAS
||----------------------------------------------------------------
||  22.7% |  9,425.9 |   679.1 |  6.7% | gotoblas_dgemm_kernel_knl
||  12.8% |  5,294.0 |   428.0 |  7.5% | gotoblas_dgetrf_single_knl
||   3.9% |  1,622.9 |   276.1 | 14.5% | gotoblas_dlaswp_plus_knl
||   1.8% |    765.3 |    88.7 | 10.4% | gotoblas_dgemv_n_knl
||   1.6% |    646.2 |   262.8 | 28.9% | gotoblas_dgemm_itcopy_knl
||================================================================
|   6.3% |  2,627.8 |      -- |    -- | MPI
||----------------------------------------------------------------
||   6.1% |  2,537.6 | 1,619.4 | 39.0% | MPI_ALLREDUCE
||================================================================
|   1.5% |    629.2 |      -- |    -- | ETC
|=================================================================
```

By default, sampling experiment with 100 samples per second

```
Table 1:  Profile by Function (limited entries shown)
  Samp% |     Samp |    Imb. |  Imb. | Group
        |          |    Samp | Samp% |  Function=[MAX10]
        |          |         |       |   PE=HIDE
 100.0% | 41,447.1 |      -- |    -- | Total
|----------------------------------------------
|  46.6% | 19,305.8 |      -- |    -- | US
||---------------------------------------------
||  32.2% | 13,353.9 |   874.1 |  6.1% | g
||   6.2% |  2,561.7 |   217.3 |  7.8% | x
||   3.9% |  1,606.8 |   140.2 |  8.0% | r
||   3.1% |  1,270.5 |   176.5 | 12.2% | b
||=============================================
|  45.5% | 18,863.6 |      -- |    -- | BL
||---------------------------------------------
||  22.7% |  9,425.9 |   679.1 |  6.7% | gotoblas_dgemm_kernel_knl
||  12.8% |  5,294.0 |   428.0 |  7.5% | gotoblas_dgetrf_single_knl
||   3.9% |  1,622.9 |   276.1 | 14.5% | gotoblas_dlaswp_plus_knl
||   1.8% |    765.3 |    88.7 | 10.4% | gotoblas_dgemv_n_knl
||   1.6% |    646.2 |   262.8 | 28.9% | gotoblas_dgemm_itcopy_knl
||===============================================================
|   6.3% |  2,627.8 |      -- |    -- | MPI
||---------------------------------------------------------------
||   6.1% |  2,537.6 | 1,619.4 | 39.0% | MPI_ALLREDUCE
||===============================================================
|   1.5% |    629.2 |      -- |    -- | ETC
|================================================================
```

Samp % is the percent of total samples taken which occurred in the given routine, averaged over all processes.

```
Table 1:  Profile by Function (limited entries shown)
  Samp% |     Samp |    Imb. |  Imb. | Group
        |          |    Samp | Samp% |  Function=[MAX10]
        |          |         |       |   PE=HIDE
 100.0% | 41,447.1 |      -- |    -- | Total
|------------------------------------
|  46.6% | 19,305.8 |      -- |    -- |
||-----------------------------------
||  32.2% | 13,353.9 |   874.1 |  6.1%
||   6.2% |  2,561.7 |   217.3 |  7.8%
||   3.9% |  1,606.8 |   140.2 |  8.0%
||   3.1% |  1,270.5 |   176.5 | 12.2%
||===================================
|  45.5% | 18,863.6 |      -- |    -- | BLAS
||--------------------------------------------------------------
||  22.7% |  9,425.9 |   679.1 |  6.7% | gotoblas_dgemm_kernel_knl
||  12.8% |  5,294.0 |   428.0 |  7.5% | gotoblas_dgetrf_single_knl
||   3.9% |  1,622.9 |   276.1 | 14.5% | gotoblas_dlaswp_plus_knl
||   1.8% |    765.3 |    88.7 | 10.4% | gotoblas_dgemv_n_knl
||   1.6% |    646.2 |   262.8 | 28.9% | gotoblas_dgemm_itcopy_knl
||==============================================================
|   6.3% |  2,627.8 |      -- |    -- | MPI
||--------------------------------------------------------------
||   6.1% |  2,537.6 | 1,619.4 | 39.0% | MPI_ALLREDUCE
||==============================================================
|   1.5% |    629.2 |      -- |    -- | ETC
|===============================================================
```

Samp is the number of samples which occurred in the given routine, averaged over all processes.

```
Table 1:  Profile by Function (limited entries shown)
  Samp% |      Samp |    Imb. |  Imb. | Group
        |           |    Samp | Samp% |  Function=[MAX10]
        |           |         |       |   PE=HIDE
 100.0% | 41,447.1 |       -- |    -- | Total
|-------------------------------------
|  46.6% | 19,305.8 |       -- |    --
||------------------------------------
||  32.2% | 13,353.9 |    874.1 |  6.1%
||   6.2% |  2,561.7 |    217.3 |  7.8%
||   3.9% |  1,606.8 |    140.2 |  8.0%
||   3.1% |  1,270.5 |    176.5 | 12.2%
||====================================
|  45.5% | 18,863.6 |       -- |    --
||------------------------------------
||  22.7% |  9,425.9 |    679.1 |  6.7% | gotoblas_dgemm_kernel_knl
||  12.8% |  5,294.0 |    428.0 |  7.5% | gotoblas_dgetrf_single_knl
||   3.9% |  1,622.9 |    276.1 | 14.5% | gotoblas_dlaswp_plus_knl
||   1.8% |    765.3 |     88.7 | 10.4% | gotoblas_dgemv_n_knl
||   1.6% |    646.2 |    262.8 | 28.9% | gotoblas_dgemm_itcopy_knl
||================================================================
|   6.3% |  2,627.8 |       -- |    -- | MPI
||----------------------------------------------------------------
||   6.1% |  2,537.6 | 1,619.4 | 39.0% | MPI_ALLREDUCE
||================================================================
|   1.5% |    629.2 |       -- |    -- | ETC
|=================================================================
```

Imb. Samp is ((maximum number of samples taken in given routine by one process) – (average number of samples taken in that routine)).

```
Table 1:  Profile by Function (limited entries shown)
  Samp% |      Samp |    Imb. |  Imb. | Group
        |           |    Samp | Samp% |  Function=[MAX10]
        |           |         |       |   PE=HIDE
 100.0% | 41,447.1 |      -- |    -- | Total
|-----------------------------------------------------------------
|  46.6% | 19,305.8 |      -- |    -- | USER
||----------------------------------------------------------------
||  32.2% | 13,353.9 |   874.1 |  6.1% | genral_
||   6.2% |  2,561.7 |   217.3 |  7.8% | xyzint_
||   3.9% |  1,606.8 |   140.2 |  8.0% | rt123_
||   3.1% |  1,270.5 |   176.5 | 12.2% | build_abket_
||================================================================
|  45.5% | 18,863.6 |      -- |    -- | BLAS
||--------
||  22.7%
||  12.8%
||   3.9%
||   1.8%
||   1.6% |     646.2 |   262.8 | 28.9% | gotoblas_dgemm_itcopy_knl
||================================================================
|   6.3% |  2,627.8 |      -- |    -- | MPI
||----------------------------------------------------------------
||   6.1% |  2,537.6 | 1,619.4 | 39.0% | MPI_ALLREDUCE
||================================================================
|   1.5% |    629.2 |      -- |    -- | ETC
|=================================================================
```

$$\text{Imb. Samp\%} = \frac{(\text{Imb. Samp})}{(\text{maximum samples by one process})} * \frac{(\text{number of processes})}{(\text{number of processes} - 1)} * 100\%$$

```
Table 2:  Profile by Group, Function, and Line (limited entries shown)

  Samp% |      Samp |     Imb. |  Imb. | Group
        |           |     Samp | Samp% |  Function=[MAX10]
        |           |          |       |   Source
        |           |          |       |    Line
        |           |          |       |     PE=HIDE


 100.0% | 41,447.1 |       -- |    -- | Total
|----------------------------------------------------------------------
|  46.6% | 19,305.8 |       -- |    -- | USER
||---------------------------------------------------------------------
||  32.2% | 13,353.9 |       -- |    -- | genral_
3|        |          |          |       |  vsvb.f90
||||-------------------------------------------------------------------
4|||   1.6% |    645.6 |    88.4 | 12.0% | line.3729
4|||   1.3% |    550.5 |    90.5 | 14.1% | line.3818
4|||   1.1% |    457.2 |   100.8 | 18.1% | line.3829
4|||   2.2% |    929.3 |   145.7 | 13.6% | line.3840
4|||   1.2% |    498.7 |    79.3 | 13.7% | line.3862
4|||   2.2% |    908.9 |   155.1 | 14.6% | line.3867
```

```
================  Observations and suggestions  ======================

MPI Grid Detection:

    There appears to be point-to-point MPI communication in a 35 X 60
    grid pattern. The 20.3% of the total execution time spent in MPI
    functions might be reduced with a rank order that maximizes
    communication between ranks on the same node. The effect of
    several rank orders is estimated below.

    A file named MPICH_RANK_ORDER.Grid was generated along with this
    report and contains usage instructions and the Custom rank order
    from the following table.


         Rank    On-Node     On-Node   MPICH_RANK_REORDER_METHOD
        Order   Bytes/PE   Bytes/PE%
                            of Total
                            Bytes/PE



        Custom  4.050e+09     34.77%  3
           SMP  2.847e+09     24.45%  1
          Fold  1.025e+08      0.88%  2
    RoundRobin  6.098e+01      0.00%  0
```

MPI rank reordering

- Ideally, one would maximize on-node communication between MPI ranks and minimize inter-node communication
- “Observations” in output helps detect point-to-point MPI communication and suggests ways to reorder MPI ranks to reduce inter-node communication
- In addition to other files, a MPICH_RANK_ORDER is produced in the subdirectory
- If CrayPat-lite decides work is well balanced across the nodes, it will not be produced

```
user@thetalogin6:~> pat_help balance mpi_rank_order
user@thetalogin6:~> pat_help balance mpi_rank_order examples
```

```
================  Observations and suggestions  ======================

MPI Grid Detection:

    There appears to be point-to-point MPI communication in a 35 X 60
    grid pattern. The 20.3% of the total execution time spent in MPI
    functions might be reduced with a rank order that maximizes
    communication between ranks on the same node. The effect of
    several rank orders is estimated below.

    A file named MPICH_RANK_ORDER.Grid was generated along with this
    report and contains usage instructions and the Custom rank order
    from the following table.


        Rank     On-Node      On-Node   MPICH_RANK_REORDER_METHOD
       Order    Bytes/PE    Bytes/PE%
                             of Total
                             Bytes/PE



       Custom  4.050e+09      34.77%  3
          SMP  2.847e+09      24.45%  1
         Fold  1.025e+08       0.88%  2
   RoundRobin  6.098e+01       0.00%  0
```

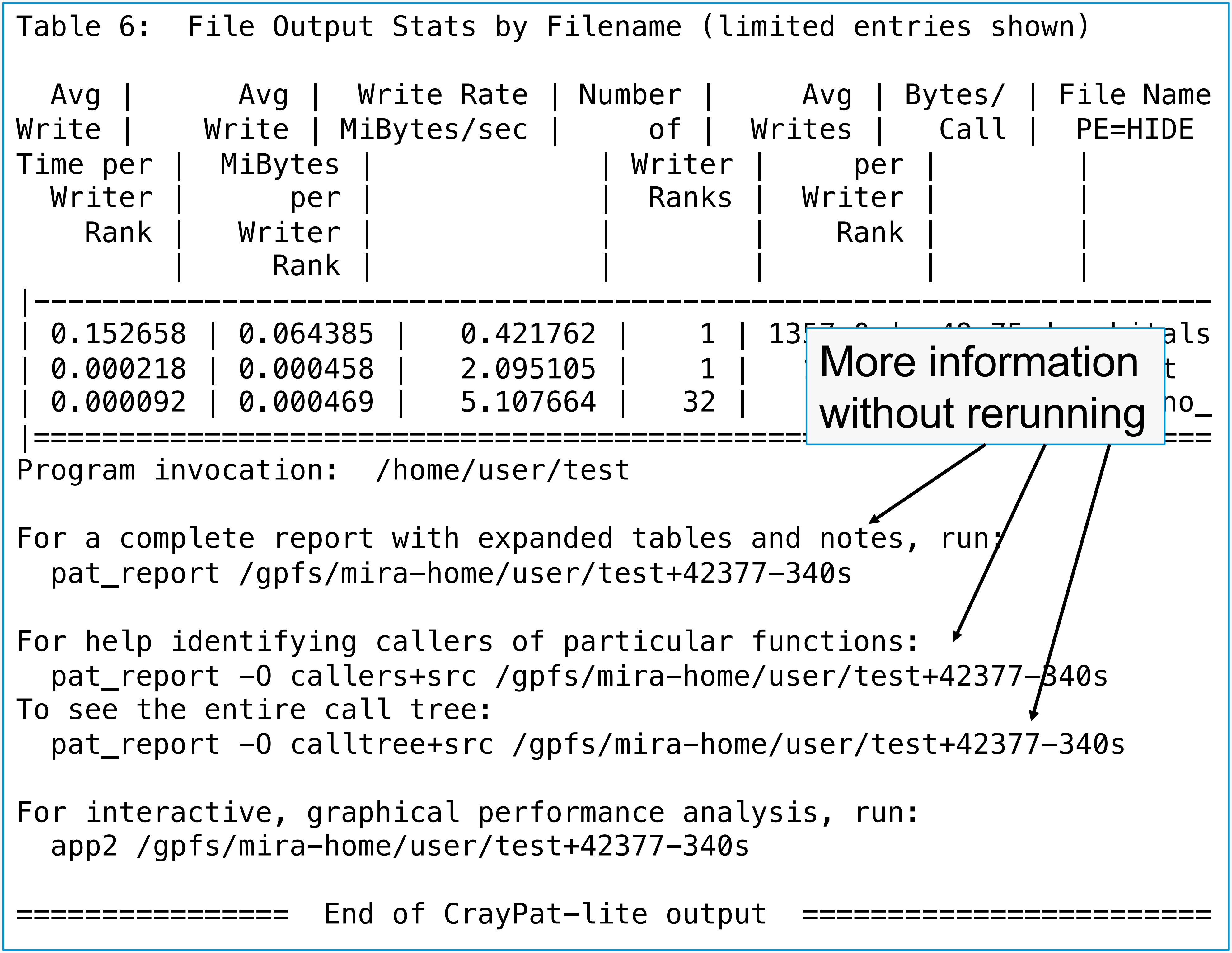
```
================  Observations and suggestions  ======================

MPI Grid Detection:

    There appears to be point-to-point MPI communication in a 35 X 60
    grid pattern. The 20.3% of the total execution time spent in MPI
    functions might be reduced with a rank order that maximizes
    communication between ranks on the same node. The effect of
    several rank orders is estimated below.

    A file named MPICH_RANK_ORDER.Grid was generated along with this
    report and contains usage instructions and the Custom rank order
    from the following table.


        Rank     On-Node      On-Node   MPICH_
       Order    Bytes/PE    Bytes/PE%
                             of Total
                             Bytes/PE


        Custom  4.050e+09      34.77%  3
           SMP  2.847e+09      24.45%  1
          Fold  1.025e+08       0.88%  2
    RoundRobin  6.098e+01       0.00%  0
```

In the subdirectory test+65086-1481s. Note that the instructions for using each MPICH_RANK_ORDER file are included within that file

```
================  Observations and suggestions  ========================

MPI Grid Detection:

    There appears to be point-to-point MP                         60
    grid pattern. The 20.3% of the total                          MPI
    functions might be reduced with a ran
    communication between ranks on the sa
    several rank orders is estimated belo

    A file named MPICH_RANK_ORDER.Grid wa                         his
    report and contains usage instructions and the Custom rank order
    from the following table.


        Rank     On-Node       On-Node   MPICH_RANK_REORDER_METHOD
       Order    Bytes/PE    Bytes/PE%
                             of Total
                             Bytes/PE


       Custom  4.050e+09       34.77%  3
          SMP  2.847e+09       24.45%  1
         Fold  1.025e+08        0.88%  2
   RoundRobin  6.098e+01        0.00%  0
```

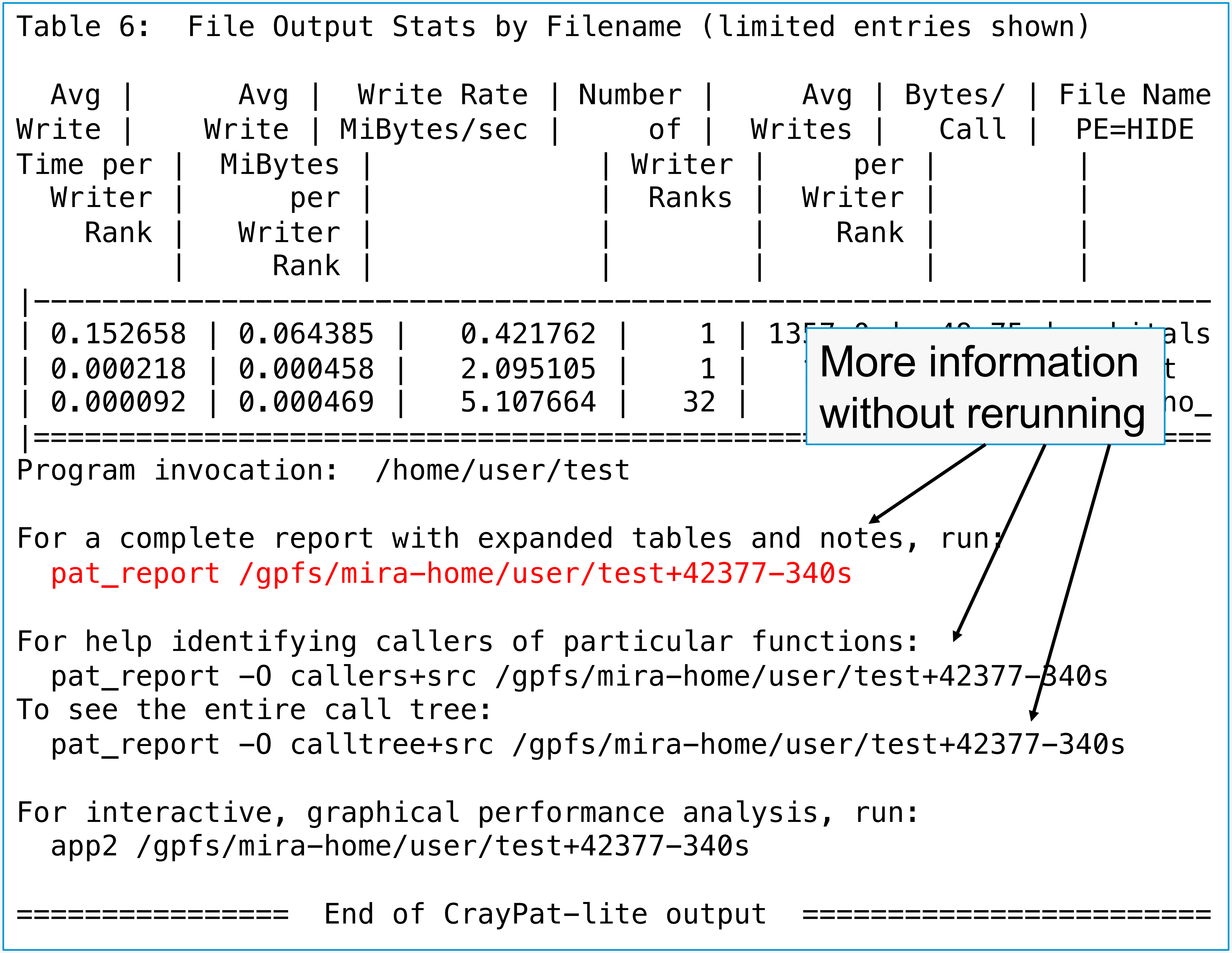
Percentage of total message bytes sent per PE stayed within each local compute node (the higher the percentage the better). In this case, the Custom order was a little better than the default SMP order.

```
Table 3:  Memory Bandwidth by Numanode (limited entries shown)

  Memory |     DDR |  MCDRAM |    Thread |  Memory | Numanode
 Traffic |  Memory |  Memory |      Time |Traffic  |Node Id=[max3,min3]
  GBytes | Traffic | Traffic |           |  GBytes |   PE=HIDE
         |  GBytes |  GBytes |           |   / Sec |
 |-------------------------------------------------------------------
 |  33,445 |  153.02 |  33,292 | 417.182412 |   80.17 | numanode.0
 ||------------------------------------------------------------------
 ||  33,306 |   14.33 |  33,292 | 417.140768 |   79.84 | nid.4022
 ||  33,292 |    0.16 |  33,292 | 417.120838 |   79.81 | nid.345
 ||  33,285 |   26.95 |  33,258 | 417.128666 |   79.80 | nid.346
 ||  32,867 |    0.19 |  32,867 | 417.100249 |   78.80 | nid.343
 ||  32,811 |   14.82 |  32,797 | 417.133453 |   78.66 | nid.3734
 |===================================================================

Table 5:  File Input Stats by Filename (limited entries shown)
 Avg Read | Avg Read |   Read Rate | Number|  Avg | Bytes/ | File Name
 Time per |  MiBytes | MiBytes/sec |   of |    Reads |   Call
 |  PE=HIDE
   Reader |      per |             |Reader|      per |        |
     Rank |   Reader |             |Ranks |   Reader |        |
          |     Rank |             |      |     Rank |        |
 |-------------------------------------------------------------------
 | 0.405698 | 0.079402 |    0.195717 |  1 | 83,259.0 |   1.00 | stdin
 | 0.000023 | 0.000023 |    1.001237 | 32 |      3.1 |   8.00 | _Unkno_
 |===================================================================
Table 6:  File Output Stats by Filename (limited entries shown)

  Avg |      Avg |  Write Rate | Number |     Avg | Bytes/ | File Name
Write |    Write | MiBytes/sec |     of |  Writes |   Call |  PE=HIDE
Time per |  MiBytes |             | Writer |     per |        |
  Writer |      per |             |  Ranks |  Writer |        |
    Rank |   Writer |             |        |    Rank |        |
         |     Rank |             |        |         |        |
|---------------------------------------------------------------------
| 0.152658 | 0.064385 |   0.421762 |    1 | 1357.0 |  49.75 | orbitals
| 0.000218 | 0.000458 |   2.095105 |    1 |   10.0 |  48.00 | stdout
| 0.000092 | 0.000469 |   5.107664 |   32 |    15.4 |  32.00 | _Unkno_
|=====================================================================
Program invocation:  /home/user/test

For a complete report with expanded tables and notes, run:
  pat_report /gpfs/mira-home/user/test+42377-340s

For help identifying callers of particular functions:
  pat_report -O callers+src /gpfs/mira-home/user/test+42377-340s
To see the entire call tree:
  pat_report -O calltree+src /gpfs/mira-home/user/test+42377-340s

For interactive, graphical performance analysis, run:
  app2 /gpfs/mira-home/user/test+42377-340s

================  End of CrayPat-lite output  ========================
```



```
Table 6:  File Output Stats by Filename (limited entries shown)

  Avg |      Avg |  Write Rate | Number |     Avg | Bytes/ | File Name
Write |    Write | MiBytes/sec |     of |  Writes |   Call |  PE=HIDE
Time per |  MiBytes |             | Writer |     per |        |
  Writer |      per |             |  Ranks |  Writer |        |
    Rank |   Writer |             |        |    Rank |        |
         |     Rank |             |        |         |        |
|-----------------------------------------------------------------------
| 0.152658 | 0.064385 |   0.421762 |    1 | 1357.0 |   40.75 |   initials
| 0.000218 | 0.000458 |   2.095105 |    1 |                           t
| 0.000092 | 0.000469 |   5.107664 |   32 |                          ho_
|=======================================================================
Program invocation:  /home/user/test

For a complete report with expanded tables and notes, run:
  pat_report /gpfs/mira-home/user/test+42377-340s

For help identifying callers of particular functions:
  pat_report -O callers+src /gpfs/mira-home/user/test+42377-340s
To see the entire call tree:
  pat_report -O calltree+src /gpfs/mira-home/user/test+42377-340s

For interactive, graphical performance analysis, run:
  app2 /gpfs/mira-home/user/test+42377-340s

================  End of CrayPat-lite output  =========================
```

More information without rerunning



```
Table 6:  File Output Stats by Filename (limited entries shown)

  Avg |      Avg |  Write Rate | Number |     Avg | Bytes/ | File Name
Write |    Write | MiBytes/sec |     of |  Writes |   Call |  PE=HIDE
Time per |  MiBytes |             | Writer |     per |        |
  Writer |      per |             |  Ranks |  Writer |        |
    Rank |   Writer |             |        |    Rank |        |
         |     Rank |             |        |         |        |
|-----------------------------------------------------------------------
| 0.152658 | 0.064385 |   0.421762 |     1 | 135                     als
| 0.000218 | 0.000458 |   2.095105 |     1 |                         t
| 0.000092 | 0.000469 |   5.107664 |    32 |                        no_
|=======================================================================
Program invocation:  /home/user/test

For a complete report with expanded tables and notes, run:
  pat_report /gpfs/mira-home/user/test+42377-340s

For help identifying callers of particular functions:
  pat_report -O callers+src /gpfs/mira-home/user/test+42377-340s
To see the entire call tree:
  pat_report -O calltree+src /gpfs/mira-home/user/test+42377-340s

For interactive, graphical performance analysis, run:
  app2 /gpfs/mira-home/user/test+42377-340s

================  End of CrayPat-lite output  ==========================
```

More information without rerunning

Using CrayPat-lite: More information without re-running

```
user@thetalogin6:~> pat_report -v /gpfs/mira-home/user/test+42377-340s
```

More details than the default report, including notes for each table

```
Notes for table 1:
  Table option:
    -O samp_profile
  Options implied by table option:
    -d sa%@0.95,sa,imb_sa,imb_sa% -b gr,fu,pe=HIDE

  Options for related tables:
    -O samp_profile+src        -O profile_max

  The Total value for Samp is the sum of the Group values.
  The Group value for Samp is the sum of the Function values.
  The Function value for Samp is the avg of the PE values.
  (To specify different aggregations, see: pat_help report options s1)
...

Table 1:  Profile by Function

  Samp% |     Samp |    Imb. |  Imb. | Group
        |          |    Samp | Samp% |  Function
        |          |         |       |   PE=HIDE

 100.0% | 41,447.1 |      -- |    -- | Total
|-------------------------------------------------------------------
|  46.6% | 19,305.8 |      -- |    -- | USER
||------------------------------------------------------------------
||  32.2% | 13,353.9 |   874.1 |  6.1% | genral_
```

```
Notes for table 1:
  Table option:
    -O samp_profile
  Options implied by table option:
    -d sa%@0.95,sa,imb_sa,imb_sa% -b gr,fu,pe=HIDE

  Options for related tables:
    -O samp_profile+src        -O profile_max

  The Total value for Samp is the sum of the Group values.
  The Group value for Samp is the sum of the Function values.
  The Function value for Samp is the avg of the PE values.
  (To specify different aggregations, see: pat_help report options s1)
...

Table 1:  Profile by Function

  Samp% |      Samp |    Imb. |
        |           |    Samp |
        |           |         |        |   PE=HIDE

 100.0% | 41,447.1 |      -- |     -- | Total
|-----------------------------------------------------------------
|  46.6% | 19,305.8 |      -- |     -- | USER
||----------------------------------------------------------------
||  32.2% | 13,353.9 |   874.1 |   6.1% | genral_
```

Explanations of details of the tables can be found by running “man pat_report”

Using CrayPat-lite: More information without re-running

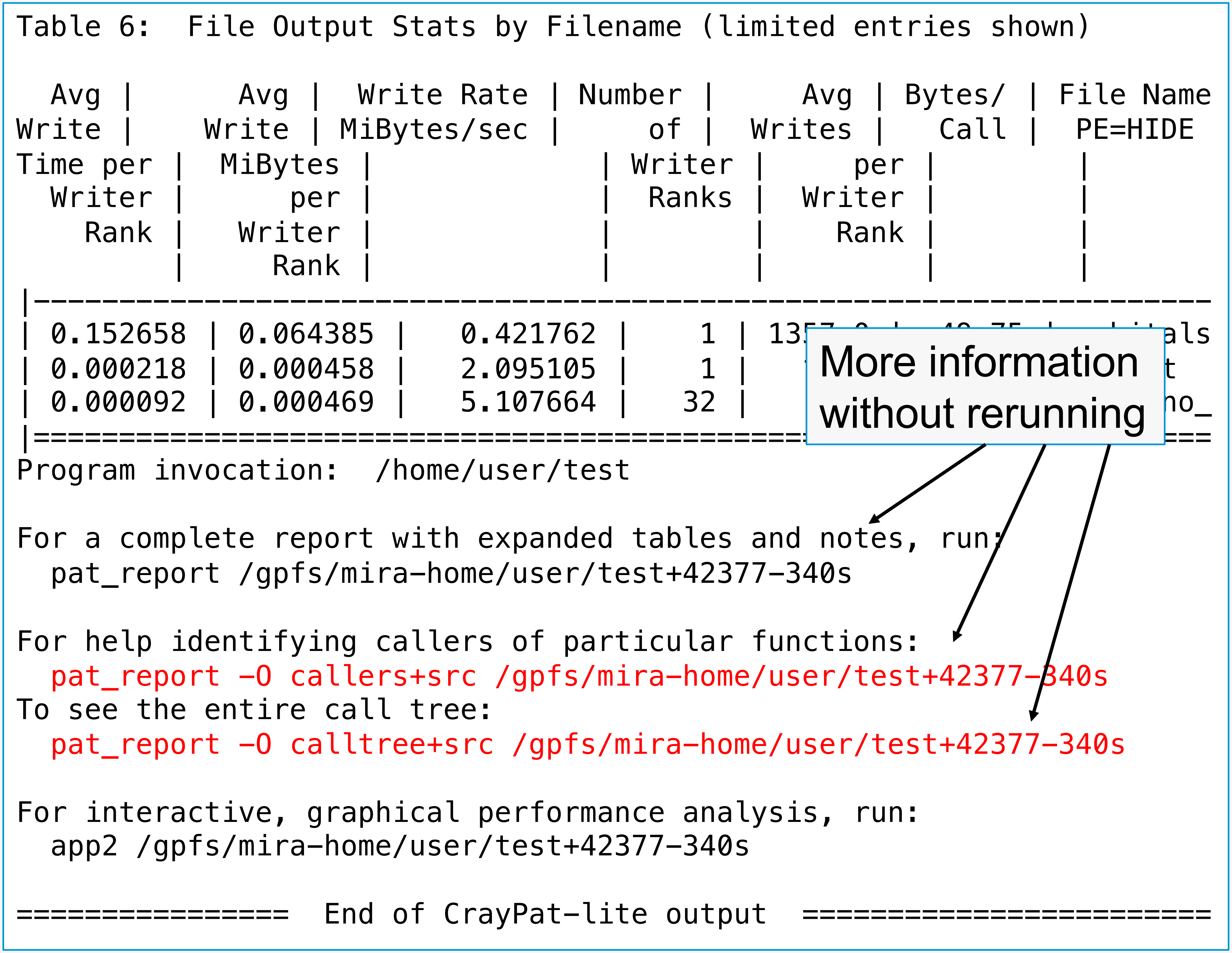
```
user@thetalogin6:~> pat_report -v /gpfs/mira-home/user/test+42377-340s
```

- Includes a load imbalance table organized by maximum samplings
- Shows the maximum and minimum samplings in a given function

```
Table 2:  Profile of maximum function times (limited entries shown)
  Samp% |     Samp |    Imb. |  Imb. | Function
        |          |    Samp | Samp% |  PE=[max,min]
|-----------------------------------------------------------------
| 100.0% | 14,228.0 |   874.1 |  6.1% | genral_
||----------------------------------------------------------------
|| 100.0% | 14,228.0 |      -- |    -- | pe.1216
||  89.6% | 12,745.0 |      -- |    -- | pe.1507
||================================================================
|  71.0% | 10,105.0 |   679.1 |  6.7% | gotoblas_dgemm_kernel_knl
||----------------------------------------------------------------
||  71.0% | 10,105.0 |      -- |    -- | pe.960
||  63.1% |  8,980.0 |      -- |    -- | pe.1252
||================================================================
|  40.2% |  5,722.0 |   428.0 |  7.5% | gotoblas_dgetrf_single_knl
||----------------------------------------------------------------
||  40.2% |  5,722.0 |      -- |    -- | pe.128
||  35.4% |  5,034.0 |      -- |    -- | pe.1841
||================================================================
|  29.2% |  4,157.0 | 1,619.4 | 39.0% | MPI_ALLREDUCE
||----------------------------------------------------------------
||  29.2% |  4,157.0 |      -- |    -- | pe.1252
||   0.3% |     38.0 |      -- |    -- | pe.960
||================================================================
|  19.5% |  2,779.0 |   217.3 |  7.8% | xyzint_
||----------------------------------------------------------------
||  19.5% |  2,779.0 |      -- |    -- | pe.1023
```

```
Table 2:  Profile of maximum function times (limited entries shown)
  Samp% |     Samp |    Imb. |  Imb. | Function
        |          |    Samp | Samp% |  PE=[max,min]
|-----------------------------------------------------------------
| 100.0% | 14,228.0 |   874.1 |  6.1% | genral_
||----------------------------------------------------------------
|| 100.0% | 14,228.0 |       -- |    -- | pe.1216
||  89.6% | 12,745.0 |       -- |    -- | pe.1507
||===================================================
|  71.0% | 10,105.0 |   679.1 |  6.7% | gotoblas_dgem
||---------------------------------------------------
||  71.0% | 10,105.0 |       -- |    -- | pe.960
||  63.1% |  8,980.0 |       -- |    -- | pe.1252
||===================================================
|  40.2% |  5,722.0 |   428.0 |  7.5% | gotoblas_dget
||---------------------------------------------------
||  40.2% |  5,722.0 |       -- |    -- | pe.128
||  35.4% |  5,034.0 |       -- |    -- | pe.1841
||===================================================
|  29.2% |  4,157.0 | 1,619.4 | 39.0% | MPI_ALLREDUCE
||----------------------------------------------------------------
||  29.2% |  4,157.0 |       -- |    -- | pe.1252
||   0.3% |     38.0 |       -- |    -- | pe.960
||================================================================
|  19.5% |  2,779.0 |   217.3 |  7.8% | xyzint_
||----------------------------------------------------------------
||  19.5% |  2,779.0 |       -- |    -- | pe.1023
```

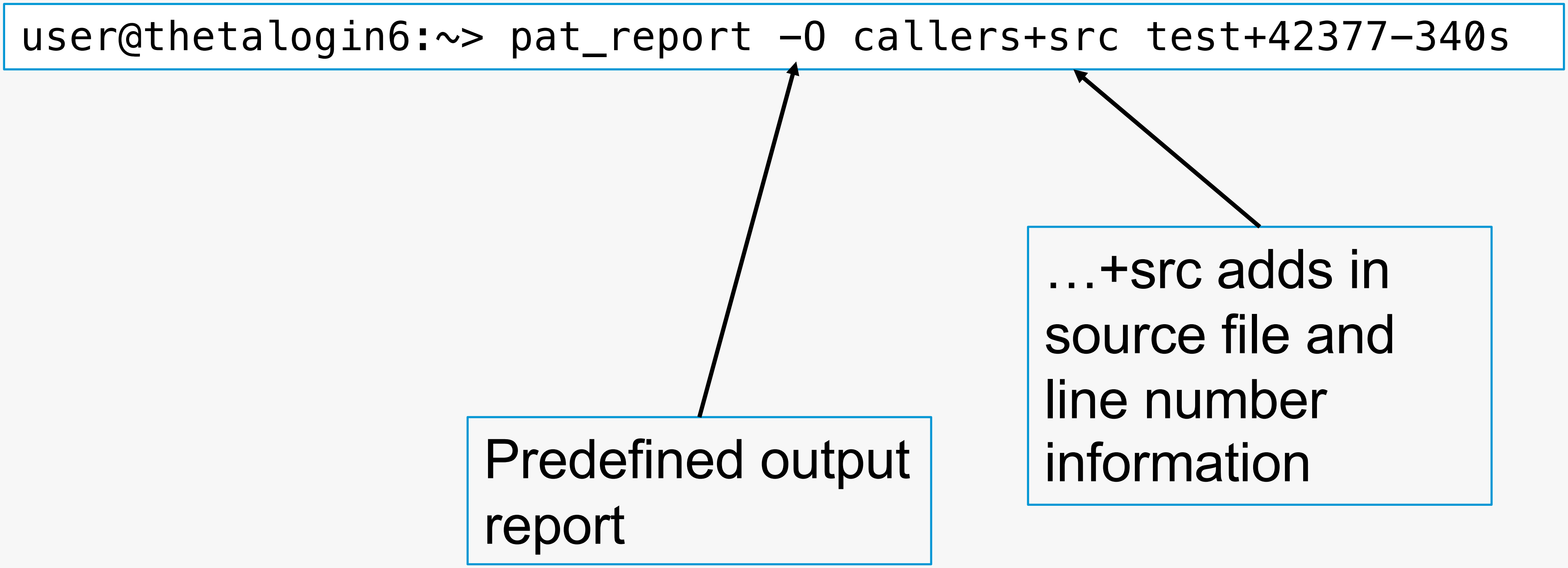
Samp is the maximum number of samplings in a given function by a PE



```
Table 6:  File Output Stats by Filename (limited entries shown)

  Avg |      Avg |  Write Rate | Number |      Avg | Bytes/ | File Name
Write |    Write | MiBytes/sec |     of |  Writes |   Call |  PE=HIDE
Time per |  MiBytes |             | Writer |      per |        |
  Writer |      per |             |  Ranks |  Writer |        |
    Rank |   Writer |             |        |     Rank |        |
         |     Rank |             |        |          |        |
|-----------------------------------------------------------------------
| 0.152658 | 0.064385 |   0.421762 |     1 | 135                    als
| 0.000218 | 0.000458 |   2.095105 |     1 |                        t
| 0.000092 | 0.000469 |   5.107664 |    32 |                        no_
|==========================================                        ====
Program invocation:  /home/user/test

For a complete report with expanded tables and notes, run:
  pat_report /gpfs/mira-home/user/test+42377-340s

For help identifying callers of particular functions:
  pat_report -O callers+src /gpfs/mira-home/user/test+42377-340s
To see the entire call tree:
  pat_report -O calltree+src /gpfs/mira-home/user/test+42377-340s

For interactive, graphical performance analysis, run:
  app2 /gpfs/mira-home/user/test+42377-340s

================  End of CrayPat-lite output  ==========================
```

More information without rerunning

Using CrayPat-lite: More information without re-running



```
user@thetalogin6:~> pat_report –O callers+src test+42377–340s
```

Predefined output report

…+src adds in source file and line number information

Using CrayPat-lite: More information without re-running

```
user@thetalogin6:~> pat_report -O callers+src test+42377-340s
```

```
Table 1:
  Profile by Function and Callers, with Line Numbers
  Samp% |     Samp | Group
        |          |  Function
        |          |   Caller
        |          |    PE=HIDE
 100.0% | 41,447.1 | Total
|---------------------------------------------------
|  46.6% | 19,305.8 | USER
||--------------------------------------------------
||  32.2% | 13,353.9 | genral_
3|  32.2% | 13,351.5 |  int2e_:vsvb.f90:line.3487
||||------------------------------------------------
4|||  21.1% |  8,760.5 | vsvb_energy_:vsvb.f90:line.1482
5|||        |          |  MAIN__:vsvb.f90:line.654
6|||        |          |   main
4|||  11.1% |  4,585.0 | vsvb_energy_:vsvb.f90:line.1490
5|||        |          |  MAIN__:vsvb.f90:line.654
6|||        |          |   main
```

Functions which call top functions

Using CrayPat-lite: More information without re-running

```
user@thetalogin6:~> pat_report -O calltree+src test+42377-340s
```

```
Table 1:
  Calltree View with Callsite Line Numbers

  Samp% |     Samp | Calltree
        |          |  PE=HIDE
 100.0% | 41,447.1 | Total
|----------------------------------------------------
|  54.5% | 22,586.3 | main
||---------------------------------------------------
||  48.2% | 19,985.1 | MAIN__:vsvb.f90:line.654
|||--------------------------------------------------
3||  29.2% | 12,109.4 | vsvb_energy_:vsvb.f90:line.1482
4||  28.8% | 11,952.4 |  int2e_:vsvb.f90:line.3487
|||||-----------------------------------------------
5||||   5.6% |  2,328.2 | genral_:vsvb.f90:line.3840
5||||   3.4% |  1,424.9 | genral_:vsvb.f90:line.3818
6||||   1.6% |    652.3 |  rt123_:util.f90:line.581
5||||   1.5% |    615.6 | genral_:vsvb.f90:line.3867
```

Functions called by top functions

Using CrayPat-lite: More information without re-running

```
user@thetalogin6:~> pat_report -s pe=ALL test+42377-340s
```

“show”

- Seeing per-rank or per-thread data
- Fine-grained imbalance information

```
Table 1:  Profile by Function

  Samp% |      Samp |     Imb. |  Imb. | Group
        |           |     Samp | Samp% |  Function
        |           |          |       |   PE

 100.0% | 41,447.1 |       -- |     -- | Total
|-----------------------------------------------------------------
|  46.6% | 19,305.8 |       -- |     -- | USER
||----------------------------------------------------------------
||  32.2% | 13,353.9 |   874.1 |  6.1% | genral_
|||---------------------------------------------------------------
3||  34.3% | 14,228.0 |       -- |     -- | pe.1216
3||  34.2% | 14,176.0 |       -- |     -- | pe.1344
3||  34.2% | 14,171.0 |       -- |     -- | pe.64
3||  34.1% | 14,152.0 |       -- |     -- | pe.1280
3||  34.1% | 14,124.0 |       -- |     -- | pe.128
3||  34.0% | 14,094.0 |       -- |     -- | pe.448
3||  34.0% | 14,092.0 |       -- |     -- | pe.192
3||  34.0% | 14,088.0 |       -- |     -- | pe.704
3||  33.9% | 14,065.0 |       -- |     -- | pe.1024
3||  33.9% | 14,059.0 |       -- |     -- | pe.1664
3||  33.9% | 14,057.0 |       -- |     -- | pe.832
3||  33.9% | 14,050.0 |       -- |     -- | pe.1984
```

Samplings in every MPI rank

Using CrayPat-lite: More information without re-running

Possible issue: Missing a function you expected

1. Samples in function were attributed to a caller function (associating lower level library routines with callers)
   - Disable this adjustment

```
user@thetalogin6:~> pat_report –P test+42377-340s
```

2. Function was below sampling threshold (0.95%)
   - Turn off thresholding:

```
user@thetalogin6:~> pat_report –T test+42377-340s
```

Using CrayPat-lite: Overview

1. Environment setup

```
user@thetalogin6:~> module unload darshan
user@thetalogin6:~> module load perftools-lite
```

2. Compiling the code to use CrayPat-lite

```
user@thetalogin6:~> make
```

3. Running the code

```
user@thetalogin6:~> qsub ./jobscript
```

4. Analyzing the output

```
Condensed report to stdout
```

```
pat_report -O lite-samples test+42377-340s
```

Helpful experiments to try

```
module avail perftools
```

- Identify time-consuming areas
  - perftools-lite
- Identify time-consuming loops (needs Cray compiler)
  - perftools-lite-loops

```
man perftools-lite
```

- Identify MPI communication issues
  - perftools-lite
  - perftools (pat_build –g mpi) to collect more MPI-specific information (analyzing size of MPI messages)
- Analyze hardware counters
  - perftools
  - papi_avail on compute nodes to see available counters
  - Set environment variable PAT_RT_PERFCTR

```
man hwpc
```

Using CrayPat: MPI communication

1. Environment setup

```
user@thetalogin6:~> module unload darshan
user@thetalogin6:~> module load perftools
```

2. Compiling the code to use CrayPat

```
user@thetalogin6:~> make
user@thetalogin6:~> pat_build –g mpi program
```

3. Running the code (inside a submission script)

```
user@thetalogin6:~> qsub –A proj –n 8 ./jobscript.sh
user@thetalogin6:~> cat jobscript.sh
#!/bin/bash –x
aprun –n 512 –N 64 program+pat
```

4. Analyzing the output

```
pat_report program+pat+41757–3827t/
```

Using CrayPat: MPI communication

1. Environment setup

```
user@thetalogin6:~> module unload
user@thetalogin6:~> module load p
```

Produces an instrumented copy of the original executable, program+pat

2. Compiling the code to use CrayPat

```
user@thetalogin6:~> make
user@thetalogin6:~> pat_build –g mpi program
```

3. Running the code (inside a submission script)

```
user@thetalogin6:~> qsub –A proj –n 8 ./jobscript.sh
user@thetalogin6:~> cat jobscript.sh
#!/bin/bash –x
aprun –n 512 –N 64 program+pat
```

4. Analyzing the output

https://pubs.cray.com

```
pat_report program+pat+41757–3827t/
```

References

- User guide “Cray Performance Measurement and Analysis Tools User Guide” available at http://pubs.cray.com
- pat_help (after module perftools-base is loaded)

```
user@thetalogin6:~> pat_help
```

- Man pages

```
# basic usage for craypat-lite
user@thetalogin6:~> man perftools-lite
# output report information
user@thetalogin6:~> man pat_report
# basic usage and environment variables info
user@thetalogin6:~> man intro_craypat
```

Summary

- CrayPat-lite
  - Easy-to-use, simple interface
  - Lets you run on many nodes, look at performance when running at scale
- CrayPat
  - More control over functions traced, more MPI communication output
  - Lets you run on many nodes, look at performance when running at scale
- Try it out!

Acknowledgements and Thanks

- Previous CrayPat tutorials, Scott Parker, Ray Loy

Using CrayPat-lite: OpenMP

- OpenMP information (overhead from entering and leaving OpenMP regions, per-thread timings, thread load imbalances)
  - Collected by default
  - Most detail from using the Cray compiler

```
function.REGION@li.49
function.LOOP@li.53
```

```
user@thetalogin6:~> module swap PrgEnv-intel PrgEnv-cray
```

```
================  Observations and suggestions  ========================

MPI Grid Detection:

    There appears to be point-to-point MPI communication in a 35 X 60
    grid pattern. The 20.3% of the total execution time spent in MPI
    functions might be reduced with a rank order that maximizes
    communication between ranks on th
    several rank orders is estimated

    A file named MPICH_RANK_ORDER.Gri                               is
    report and contains usage instruc                               er
    from the following table.


        Rank    On-Node      On-Node   MPICH_RANK_REORDER_METHOD
       Order   Bytes/PE    Bytes/PE%
                            of Total
                            Bytes/PE


       Custom  4.050e+09      34.77%  3
          SMP  2.847e+09      24.45%  1
         Fold  1.025e+08       0.88%  2
    RoundRobin  6.098e+01       0.00%  0
```

This is the grid that pat_report identified by studying MPI message traffic. It can be changed by the user via the -s rank_grid_dim option.

```
================  Observations and suggestions  ======================

MPI Grid Detection:

    There appears to be point-to-point MPI communication in a 35 X 60
    grid pattern. The 20.3% of the total execution time spent in MPI
    functions might be reduced with a rank order that maximizes
    communication between ranks on the same node. The effect of
                           s is estimated below.

                           _RANK_ORDER.Grid was generated along with this
                           s usage instructions and the Custom rank order
                             table.



                           de      On-Node   MPICH_RANK_REORDER_METHOD
        Order    Bytes/PE    Bytes/PE%
                              of Total
                              Bytes/PE



        Custom   4.050e+09      34.77%   3
           SMP   2.847e+09      24.45%   1
          Fold   1.025e+08       0.88%   2
    RoundRobin   6.098e+01       0.00%   0
```

This MPI-based rank order is calculated only if this application shows that significant (>10%) time is spent doing MPI-related work.

```
================  Observations and suggestions  ========================

MPI Grid Detection:

    There appears to be point-to-point MPI communication in a 35 X 60
    grid pattern. The 20.3% of the total execution time spent in MPI
    functions might be reduced with a rank order that maximizes
    communication between ranks on the same node. The effect of
    several rank orders is estimated below.

    A file named MPICH_RANK_ORDER.Grid was generated along with this
    report and contains usage instructions and the Custom rank order
    from the following table.


         Rank    On-Node      On-Node   MPICH_
        Order   Bytes/PE    Bytes/PE%
                             of Total
                             Bytes/PE


        Custom  4.050e+09      34.77%  3
           SMP  2.847e+09      24.45%  1
          Fold  1.025e+08       0.88%  2
    RoundRobin  6.098e+01       0.00%  0
```

In the subdirectory test+65086-1481s. Note that the instructions for using each MPICH_RANK_ORDER file are included within that file

```
================  Observations and suggestions  ======================

MPI Grid Detection:

    There appears to be point-to-point MP                              X 60
    grid pattern. The 20.3% of the total                              MPI
    functions might be reduced with a ran
    communication between ranks on the sa
    several rank orders is estimated belo

    A file named MPICH_RANK_ORDER.Grid wa                             his
    report and contains usage instructior                             der
    from the following table.


        Rank    On-Node      On-Node   MPICH_RANK_REORDER_METHOD
       Order   Bytes/PE    Bytes/PE%
                            of Total
                            Bytes/PE


      Custom  4.050e+09       34.77%  3
         SMP  2.847e+09       24.45%  1
        Fold  1.025e+08        0.88%  2
  RoundRobin  6.098e+01        0.00%  0
```

Custom rank order was able to arrange the ranks such that 34% of the total MPI message bytes sent per PE stayed within each local compute node (the higher the percentage the better). In this case, the Custom order was a little better than the default SMP order.