Using the Cray perftools-lite Performance Measurement Tool

Helen He
NERSC User Services Group

October 10, 2013
What is Perftools-lite

• A simplified and easy to use version of the CrayPat performance measurement and analysis tool.
• Provides basic performance analysis info automatically with simple steps.
• Users can decide whether to use full perftools version afterwards.
Outputs from Perftools-lite

• In stdout, basic information from the default “sample_profile” option:
  – execution time
  – memory high water mark
  – aggregate FLOPS rate (only on Hopper)
  – top time-consuming user functions
  – MPI information, etc.

• A *.rpt text file with the same info as above
• A *.ap2 file that can be used with:
  – “pat_report” for more detailed information
  – “app2” for graphic visualization

• Possible one or more suggested MPICH_RANK_ORDER_FILE files.
Sample Edison Reports

CrayPat/X: Version 6.1.2 Revision 11877 (xf 11595) 09/27/13 12:00:25
Experiment: lite_sample_profile
Number of PEs (MPI ranks): 4
Numbers of PEs per Node: 4
Numbers of Threads per PE: 1
Number of Cores per Socket: 12
System name and speed: nid01667 2401 MHz

### Table 1: Profile by Function Group and Function (top 1 functions shown)

<table>
<thead>
<tr>
<th>Samp%</th>
<th>Samp</th>
<th>Imb.</th>
<th>Imb.</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PE=HIDE</td>
</tr>
<tr>
<td>100.0%</td>
<td>3584.0</td>
<td>--</td>
<td>--</td>
<td>Total</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>98.8%</td>
<td>3542.8</td>
<td>12.2</td>
<td>0.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>98.8%</td>
<td>3542.8</td>
<td>12.2</td>
<td>0.5%</td>
</tr>
</tbody>
</table>

Observations and suggestions

**MFLOPS not available on Intel Ivy Bridge:**

The document that specifies performance monitoring events for Intel processors does not include events that could be used to compute a count of floating point operations for Ivy Bridge processors: Intel 64 and IA-32 Architectures Software Developer’s Manual, Order Number 253665-046US, March 2013.

End Observations

### Table 2: File Output Stats by Filename (top 10 files shown)

<table>
<thead>
<tr>
<th>Write</th>
<th>Write Rate</th>
<th>Write</th>
<th>Bytes/</th>
<th>File Name[max10]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>MBytes/sec</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Call</td>
</tr>
<tr>
<td>0.000732</td>
<td>0.001267</td>
<td>1.730438</td>
<td>31.0</td>
<td>42.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000732</td>
<td>0.001267</td>
<td>1.730438</td>
<td>31.0</td>
<td>42.87</td>
</tr>
</tbody>
</table>

Program invocation: ./poisson_mpi

For a complete report with expanded tables and notes, run: pat_report /scratch1/scratchdirs/yunhe/perftools-lite/poisson_mpi+1122793-170s.ap2

For help identifying callers of particular functions: pat_report -O callers+src /scratch1/scratchdirs/yunhe/perftools-lite/poisson_mpi+1122793-170s.ap2

To see the entire call tree: pat_report -O calltree+src /scratch1/scratchdirs/yunhe/perftools-lite/poisson_mpi+1122793-170s.ap2

For interactive, graphical performance analysis, run: app2 /scratch1/scratchdirs/yunhe/perftools-lite/poisson_mpi+1122793-170s.ap2
Sample Hopper Reports

CrayPat/X: Version 6.1.2 Revision 11877 (xf 11595) 09/27/13 12:00:25

Experiment: lite sample_profile

Number of PEs (MPI ranks): 240
Numbers of PEs per Node: 24 PEs on each of 10 Nodes
Numbers of Threads per PE: 1
Number of Cores per Socket: 12

Wall Clock Time: 290.822940 secs
High Memory: 243.36 MBytes
MFLOPS (aggregate): 47302.39 M/sec
I/O Read Rate: 0.56 MBytes/Sec
I/O Write Rate: 0.12 MBytes/Sec

Table 1: Profile by Function Group and Function (top 10 functions shown)

<table>
<thead>
<tr>
<th>Function</th>
<th>Time</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>100.0%</td>
<td>28484.6</td>
<td></td>
</tr>
<tr>
<td>61.8%</td>
<td>17598.4</td>
<td>USER</td>
</tr>
<tr>
<td>36.3%</td>
<td>10328.2</td>
<td>58.8</td>
</tr>
<tr>
<td>29.6%</td>
<td>8432.1</td>
<td>4.8%</td>
</tr>
<tr>
<td>9.0%</td>
<td>2571.0</td>
<td>4.8%</td>
</tr>
</tbody>
</table>

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

Table 2: File Input Stats by Filename (top 10 files shown)

Table 3: File Output Stats by Filename (top 10 files shown)
Steps to Use Perftools-lite on Edison

- Can use any PrgEnv
- `module load perftools-lite` (current default is version 6.1.2)
- `module unload darshan` (note: it has conflict with darshan)
- Build as normal
- **In batch script, set CRAY_ROOTFS to DSL**
- For perftools-lite versions 6.1.1 or older, if the job is submitted from a GPFS system, then also in the batch script:
  - set `PAT_RT_EXPFILE_DIR` to a directory in Lustre file system, or,
  - set `PAT_RT_EXPFILE_MAX` to the number of PEs or -1
- Run as normal. Performance data is summarized at the end of the job stdout.
- More detailed info can also be gathered with `pat_report` or `apprentice2` after the run.
Steps to use perftools-lite on Hopper

• To use the newest version perftools/6.1.2, which is built upon cray-mpich/6.x.x:
  – % module unload cray-libsci cray-mpich2
  – % module load cray-libsci/12.1.01
  – % module load cray-mpich/6.1.0
  – % module unload darshan
  – % module load perftools-lite/6.1.2

• To use perftools-lite/6.1.1 or older
  – % module unload darshan
  – % module load perftools-lite/6.1.2

• Follow the rest of steps for Edison
Usage considerations

• Default is sample_profile: a sampling type experiment
• Can also use a tracing type experiment
  – set CRAYPAT_LITE to event_profile
  – relink the application
• It is a good idea to keep *.o files
• IO report starts from perftools-lite/6.1.2 version
• No MFLOPS information on Edison (Intel Ivy-bridge)
• Do follow instructions in the brief report about the pat_report and app2 commands, to see a lot more detailed information
• Do make use of suggested MPICH_RANK_ORDER file
• Increase wall time request when gathering performance data due to the overhead with perftools-lite
  – Not suggested to use for every run
Use apprentice2 to see code details

% app2 cam.perftools-lite.sample+21365190-4717s.ap2
More informations

• % module load training

• See example codes, reports, detailed steps in README at:
  – $EXAMPLES/Edison2013/perftools-lite

• Documentations
  – % man perftools-lite  (when the module is loaded)
  – Using Cray Performance Measurement and Analysis Tools
    http://docs.cray.com/books/S-2376-612/S-2376-612.pdf
National Energy Research Scientific Computing Center