Next-gen profiling-infrastructure for supercomputers based on hybrid nodes

Juan Gonzalez, Eun Kyung Lee, I-Hsin Chung
IBM Research - High Performance Systems Software
Motivation and objectives

- **Understand interactions among hardware subsystems** from the performance perspective
  - System usage
  - Application characterization

- IBM integrated profiling for current hybrid architectures (Firestone, Garrison, etc.)

- Integrate existing profiling techniques for CPU + GPU + MPI + …

- Understand interactions among hardware subsystems
  - **To explore optimization opportunities** for applications to achieve greater performance
  - **To analyze hardware performance** to verify/assist architectural design on hybrid nodes
IBM next-gen profiling: modules & interactions
Use Case: LULESH CUDA

- The LULESH benchmark represents workloads found in Lagrangian hydrodynamics codes
  - Approximates the hydrodynamics equations discretely by partitioning the problem domain into a collection of volumetric elements defined by a mesh.
  - Designed to support an unstructured mesh with hexahedral elements
Use Case: LULESH CUDA experimental setup

- 5 Hardware counters groups (22 counters)
- Execution with 8 MPI ranks, 1 CPU per rank, 1 GPU for all ranks.
- Problem size = 100, Iterations = 100
Use Case: LULESH CUDA hpm_job_summary

Hardware counter summary for job 4746, counter group 13.
Number of MPI ranks in the reporting group = 2.
Max number of MPI ranks per node = 8.

--- Counter values for processes in this reporting group ---

<table>
<thead>
<tr>
<th>label</th>
<th>min-value</th>
<th>min-rank</th>
<th>max-value</th>
<th>max-rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>(PM_CYC) Processor cycles [shared core]</td>
<td>8.224742e+10</td>
<td>5</td>
<td>9.253404e+10</td>
<td>0</td>
</tr>
<tr>
<td>(PM_CMPLUSTALL) Stalled cycles</td>
<td>4.321196e+10</td>
<td>5</td>
<td>5.295399e+10</td>
<td>0</td>
</tr>
<tr>
<td>(PM_CMPLUSTALL_THRD) Thread Blocked</td>
<td>7.247977e+09</td>
<td>0</td>
<td>8.297821e+09</td>
<td>5</td>
</tr>
<tr>
<td>(PM_RUNINSTCMPL) Instructions completed</td>
<td>9.169238e+10</td>
<td>5</td>
<td>9.464965e+10</td>
<td>0</td>
</tr>
<tr>
<td>(PM_CMPLUSTALL_BRUCRU) Stall due to BR or CR</td>
<td>4.007753e+08</td>
<td>5</td>
<td>8.585453e+08</td>
<td>0</td>
</tr>
<tr>
<td>(PM_RUNCYC) Run cycles</td>
<td>8.224742e+10</td>
<td>5</td>
<td>9.253404e+10</td>
<td>0</td>
</tr>
<tr>
<td>MPI p2p communication count</td>
<td>1.048000e+03</td>
<td>5</td>
<td>1.148000e+03</td>
<td>0</td>
</tr>
<tr>
<td>MPI p2p communication data size</td>
<td>8.863595e+07</td>
<td>0</td>
<td>8.863595e+07</td>
<td>0</td>
</tr>
<tr>
<td>MPI p2p communication time (us)</td>
<td>7.061505e+06</td>
<td>0</td>
<td>8.900850e+06</td>
<td>5</td>
</tr>
<tr>
<td>MPI collective communication count</td>
<td>2.100000e+01</td>
<td>0</td>
<td>2.100000e+01</td>
<td>0</td>
</tr>
<tr>
<td>MPI collective communication data size</td>
<td>1.600000e+02</td>
<td>0</td>
<td>1.600000e+02</td>
<td>0</td>
</tr>
<tr>
<td>MPI collective communication time (us)</td>
<td>2.328000e+04</td>
<td>0</td>
<td>3.993870e+05</td>
<td>5</td>
</tr>
<tr>
<td>MPI I/O communication count</td>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>0</td>
</tr>
<tr>
<td>MPI I/O communication data size</td>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>0</td>
</tr>
<tr>
<td>MPI I/O communication time (us)</td>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>0</td>
</tr>
<tr>
<td>CUDA total calls to runtime</td>
<td>4.956000e+03</td>
<td>5</td>
<td>5.064000e+03</td>
<td>0</td>
</tr>
<tr>
<td>CUDA total time in runtime (us)</td>
<td>3.487636e+07</td>
<td>5</td>
<td>3.543401e+07</td>
<td>0</td>
</tr>
<tr>
<td>CUDA total kernels executed</td>
<td>1.887000e+03</td>
<td>5</td>
<td>2.047000e+03</td>
<td>0</td>
</tr>
<tr>
<td>CUDA total time in kernels (us)</td>
<td>3.476472e+07</td>
<td>5</td>
<td>3.534324e+07</td>
<td>0</td>
</tr>
<tr>
<td>CUDA Host-to-Device bytes transferred</td>
<td>2.086202e+08</td>
<td>5</td>
<td>2.284816e+08</td>
<td>0</td>
</tr>
<tr>
<td>CUDA Device-to-Host bytes transferred</td>
<td>5.725858e+08</td>
<td>5</td>
<td>6.228501e+07</td>
<td>0</td>
</tr>
<tr>
<td>inst_executed</td>
<td>5.496137e+07</td>
<td>0</td>
<td>7.013558e+07</td>
<td>5</td>
</tr>
<tr>
<td>active_cycles</td>
<td>8.775782e+07</td>
<td>0</td>
<td>1.092330e+08</td>
<td>5</td>
</tr>
<tr>
<td>warps launched</td>
<td>1.597370e+05</td>
<td>0</td>
<td>1.877000e+05</td>
<td>5</td>
</tr>
</tbody>
</table>
Use Case: LULESH CUDA hpm_job_summary

CalcKinematicsAndMonotonicQGradient
call count = 20, avg time (us) = 321498, max time (us) = 367254 :

--- Counter values for processes in this reporting group ---

<table>
<thead>
<tr>
<th>min-value</th>
<th>min-rank</th>
<th>max-value</th>
<th>max-rank</th>
<th>avg-value</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.247995e+06</td>
<td>5</td>
<td>4.269374e+06</td>
<td>0</td>
<td>4.258684e+06</td>
<td>(PM_CYC) Processor cycles [shared core]</td>
</tr>
<tr>
<td>2.137681e+06</td>
<td>0</td>
<td>2.170562e+06</td>
<td>5</td>
<td>2.154122e+06</td>
<td>(PM_CMPLUSTALL) Stalled cycles</td>
</tr>
<tr>
<td>4.096230e+05</td>
<td>0</td>
<td>4.116920e+05</td>
<td>5</td>
<td>4.105705e+05</td>
<td>(PM_CMPLUSTALL_THRD) Thread Blocked</td>
</tr>
<tr>
<td>2.903952e+06</td>
<td>5</td>
<td>2.904712e+06</td>
<td>0</td>
<td>2.904332e+06</td>
<td>(PM_RUNINST_CMP) Instructions completed</td>
</tr>
<tr>
<td>1.057470e+05</td>
<td>0</td>
<td>1.061930e+05</td>
<td>5</td>
<td>1.059700e+05</td>
<td>(PM_CMPLUSTALL_BRU_CRU) Stall due to BR or CR</td>
</tr>
<tr>
<td>4.247995e+06</td>
<td>5</td>
<td>4.269374e+06</td>
<td>0</td>
<td>4.258684e+06</td>
<td>(PM_RUN_CYC) Run cycles</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>5</td>
<td>0.000000e+00</td>
<td>MPI p2p communication count</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>MPI p2p communication data size</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>MPI p2p communication time (us)</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>5</td>
<td>0.000000e+00</td>
<td>MPI collective communication count</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>MPI collective communication data size</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>MPI collective communication time (us)</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>5</td>
<td>0.000000e+00</td>
<td>MPI I/O communication count</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>MPI I/O communication data size</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>MPI I/O communication time (us)</td>
</tr>
<tr>
<td>4.000000e+01</td>
<td>0</td>
<td>4.000000e+01</td>
<td>0</td>
<td>4.000000e+01</td>
<td>CUDA total calls to runtime</td>
</tr>
<tr>
<td>2.751730e+05</td>
<td>0</td>
<td>3.667060e+05</td>
<td>5</td>
<td>3.209260e+05</td>
<td>CUDA total time in runtime (us)</td>
</tr>
<tr>
<td>2.000000e+01</td>
<td>0</td>
<td>2.000000e+01</td>
<td>0</td>
<td>2.000000e+01</td>
<td>CUDA total kernels executed</td>
</tr>
<tr>
<td>2.751630e+05</td>
<td>0</td>
<td>3.666890e+05</td>
<td>5</td>
<td>3.209395e+05</td>
<td>CUDA total time in kernels (us)</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>5</td>
<td>0.000000e+00</td>
<td>CUDA Host-to-Device bytes transferred</td>
</tr>
<tr>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>0</td>
<td>0.000000e+00</td>
<td>CUDA Device-to-Host bytes transferred</td>
</tr>
<tr>
<td>7.962863e+06</td>
<td>0</td>
<td>1.982396e+07</td>
<td>5</td>
<td>1.389341e+07</td>
<td>inst_executed</td>
</tr>
<tr>
<td>2.349379e+07</td>
<td>0</td>
<td>5.445842e+07</td>
<td>5</td>
<td>3.897611e+07</td>
<td>active_cycles</td>
</tr>
<tr>
<td>5.290000e+03</td>
<td>0</td>
<td>1.308400e+04</td>
<td>5</td>
<td>9.187000e+03</td>
<td>warps_launched</td>
</tr>
</tbody>
</table>
Use Case: LULESH CUDA hpm_process_summary

======================================================================
Hardware counter data, group = 13, rank = 0.
Number of MPI ranks per node = 8.
======================================================================

**mpiAll** call count = 1, elapsed time = 68142052 (us)
92534042775 : (PM_CYC) Processor cycles [shared core]
52953986794 : (PM_CMPLUSTALL) Stalled cycles
7247976531 : (PM_CMPLUSTALL_THRD) Thread Blocked
94649648153 : (PM_RUN_INST_CMP) Instructions completed
885845339 : (PM_CMPLUSTALL_BRU_CRU) Stall due to BR or CR
92534042775 : (PM_RUN_CYC) Run cycles
1148 : MPI p2p communication count
88635952 : MPI p2p communication data size
7661505 : MPI p2p communication time (us)
21 : MPI collective communication count
160 : MPI collective communication data size
23280 : MPI collective communication time (us)
0 : MPI I/O communication count
0 : MPI I/O communication data size
0 : MPI I/O communication time (us)
5064 : CUDA total calls to runtime
35430111 : CUDA total time in runtime (us)
2047 : CUDA total kernels executed
35432444 : CUDA total time in kernels (us)
228481640 : CUDA Host-to-Device bytes transferred
62285008 : CUDA Device-to-Host bytes transferred
54961368 : inst_executed
87757815 : active_cycles
159737 : warps_launched
Use Case: LULESH CUDA hpm_process_summary

<table>
<thead>
<tr>
<th>Function</th>
<th>Count</th>
<th>Elapsed Time (us)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CalcKinematicsAndMonotonicQGradient</td>
<td>20</td>
<td>275742</td>
</tr>
</tbody>
</table>

- Processor cycles [shared core]: 4269374
- Stalled cycles: 2137681
- Thread blocked: 409623
- Instructions completed: 2904712
- Stall due to BR or CR: 105747
- Run cycles: 4269374

- MPI p2p communication count: 0
- MPI p2p communication data size: 0
- MPI p2p communication time (us): 0

- MPI collective communication count: 0
- MPI collective communication data size: 0
- MPI collective communication time (us): 0

- MPI I/O communication count: 0
- MPI I/O communication data size: 0
- MPI I/O communication time (us): 0

- CUDA total calls to runtime: 40
- CUDA total time in runtime: 275173
- CUDA total kernels executed: 20
- CUDA total time in kernels: 275163

-Host-to-Device bytes transferred: 0
-Device-to-Host bytes transferred: 0

- Instructions executed: 7962863
- Active cycles: 23493788
- Warps launched: 5290
Use Case: LULESH CUDA mpi_profile

Data for MPI rank 0 of 8:
Times and statistics from MPI_Init() to MPI_Finalize().

<table>
<thead>
<tr>
<th>MPI Routine</th>
<th>#calls</th>
<th>avg. bytes</th>
<th>time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPI_Comm_rank</td>
<td>185</td>
<td>0.0</td>
<td>0.000</td>
</tr>
<tr>
<td>MPI_Comm_size</td>
<td>1</td>
<td>0.0</td>
<td>0.000</td>
</tr>
<tr>
<td>MPI_Isend</td>
<td>207</td>
<td>142428.1</td>
<td>0.011</td>
</tr>
<tr>
<td>MPI_Irecv</td>
<td>347</td>
<td>170470.7</td>
<td>0.001</td>
</tr>
<tr>
<td>MPI_Wait</td>
<td>347</td>
<td>0.0</td>
<td>5.283</td>
</tr>
<tr>
<td>MPI_Waitall</td>
<td>61</td>
<td>0.0</td>
<td>2.367</td>
</tr>
<tr>
<td>MPI_Barrier</td>
<td>1</td>
<td>0.0</td>
<td>0.009</td>
</tr>
<tr>
<td>MPI_Reduce</td>
<td>1</td>
<td>8.0</td>
<td>0.001</td>
</tr>
<tr>
<td>MPI_Allreduce</td>
<td>19</td>
<td>8.0</td>
<td>0.014</td>
</tr>
</tbody>
</table>

MPI task 0 of 8 had the median communication time.
total communication time = 7.685 seconds.
total elapsed time = 69.496 seconds.
user cpu time = 28.499 seconds.
system time = 5.604 seconds.
max resident set size = 412.500 MBytes.

Rank 3 reported the largest memory utilization : 413.69 MBytes
Rank 0 reported the largest elapsed time : 69.50 sec
Use Case: LULESH CUDA mpi_profile

Message size distributions:

<table>
<thead>
<tr>
<th>Function</th>
<th>#calls</th>
<th>avg. bytes</th>
<th>time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MPI_Isend</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>8.0</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>24.0</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>808.0</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>2424.0</td>
<td>0.003</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>81608.0</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>242412.0</td>
<td>0.007</td>
</tr>
<tr>
<td><strong>MPI_Irecv</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>8.0</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>24.0</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>48.0</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>808.0</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>2424.0</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>4848.0</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>81608.0</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>242412.0</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>489648.0</td>
<td>0.000</td>
</tr>
<tr>
<td><strong>MPI_Reduce</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>8.0</td>
<td>0.001</td>
</tr>
<tr>
<td><strong>MPI_Allreduce</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>8.0</td>
<td>0.014</td>
</tr>
</tbody>
</table>
Use Case: LULESH CUDA mpi_profile

Communication summary for all tasks:

minimum communication time = 4.098 sec for task 2
median communication time = 7.685 sec for task 0
maximum communication time = 11.219 sec for task 7

MPI timing summary for all ranks:

<table>
<thead>
<tr>
<th>taskid</th>
<th>comm(s)</th>
<th>elapsed(s)</th>
<th>user(s)</th>
<th>system(s)</th>
<th>size(MB)</th>
<th>switches</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7.68</td>
<td>69.50</td>
<td>28.50</td>
<td>5.60</td>
<td>412.50</td>
<td>51102</td>
</tr>
<tr>
<td>1</td>
<td>8.33</td>
<td>66.03</td>
<td>24.45</td>
<td>5.41</td>
<td>411.12</td>
<td>50155</td>
</tr>
<tr>
<td>2</td>
<td>4.10</td>
<td>66.03</td>
<td>23.58</td>
<td>5.62</td>
<td>411.25</td>
<td>50107</td>
</tr>
<tr>
<td>3</td>
<td>7.48</td>
<td>66.03</td>
<td>23.54</td>
<td>5.53</td>
<td>413.69</td>
<td>50604</td>
</tr>
<tr>
<td>4</td>
<td>7.94</td>
<td>66.03</td>
<td>24.71</td>
<td>5.44</td>
<td>411.31</td>
<td>48061</td>
</tr>
<tr>
<td>5</td>
<td>9.34</td>
<td>66.01</td>
<td>25.35</td>
<td>5.18</td>
<td>412.06</td>
<td>48064</td>
</tr>
<tr>
<td>6</td>
<td>6.60</td>
<td>66.02</td>
<td>25.44</td>
<td>5.51</td>
<td>413.00</td>
<td>47436</td>
</tr>
<tr>
<td>7</td>
<td>11.22</td>
<td>66.00</td>
<td>27.07</td>
<td>5.36</td>
<td>413.25</td>
<td>47294v</td>
</tr>
</tbody>
</table>
Use Case: LULESH CUDA Using the DB

- Structured performance data + standard SQL data access
  - Views to ease data access + GUI (MySQL Workbench)
Work in Progress

- Work with individual applications
- Support different compilers/ instrumentation methods
- Add more performance data sources (I/O, OpenMP, network, energy)
- Explore modeling with PAPI/native metrics
Thank you!
CORAL system profiling

- Sub-routine hooks (TRACE)
  - GCC ‘-finstrument-function’

- HPM (CPU)
  - PAPI

- MPI (Comms.)
  - PMPI

- CUDA (GPU)
  - CUPTI (Callback + Event APIs)

- HPM+MPI+X
IBM next-gen profiling: modules & interactions

- **TRACE** (Sub-routines)
- Sub-routine Entry/Exit Hooks + App Init/Fini
  - GCC’s –finstrument-function
  - Manual instrumentation API
  - Based on IBM BG/Q performance repository
IBM next-gen profiling: modules & interactions

- TRACE (Sub-routines)

- HPM (CPU)
  - CPU HW counter measurements
  - PAPI
  - Runtime SW counters
  - Based on HPCT HPM Code
IBM next-gen profiling: modules & interactions

- MPI Entry/Exit
- PMPI
- Runtime counters to HPM
- Based on HPCT mpi_profiler

TRACE (Sub-routines)

MPI (Messaging) → HPM (CPU)
IBM next-gen profiling: modules & interactions

**TRACE (Sub-routines)**

- CUDA Entry/Exit
  - CUPTI Callback API
- GPU HW counters
  - CUPTI Event API
- Runtime counters to HPM
- 100% New 😊

**MPI (Messaging)**

**HPM (CPU)**

**CUDA (GPU)**
IBM next-gen profiling: module details

- **MPI (Messaging)**
  - mpi_profile(s)

- **HPM (CPU)**
  - hpm_job_summary
  - hpm_profile(s)

- **CUDA (GPU)**

**TRACE** (Sub-routines)
Usage flow chart

```
foo();

void foo()
{
    // stuff that matters
    MPI_AlltoAll(...);
    // more stuff that matters
    myKernel<...,...>;
    // even more!
}

entry_measurements(char* function)
{
    // Entry CPU counters
}
```
Usage flow chart

```c
void foo()
{
    // stuff that matters
    MPI_AlltoAll(...);
    // more stuff that matters
    myKernel<..., ...>;
    // even more!
}

MPI_AlltoAll_Interposition(...)
{
    // MPI entry measurements
    PMPI_AlltoAll(...)
    // MPI exit measurements
}
```
Usage flow chart

```c
foo();
void foo()
{
  // stuff that matters
  MPI_AlltoAll(...);
  // more stuff that matters
  myKernel<...,...>
  // even more!
}

kernel_entry_callback(...)
{
  // CUDA entry measurements
}

kernel_exit_callback(...)
{
  // CUDA exit measurements
}
```
Usage flow chart

```c
foo();

void foo()
{
    // stuff that matters
    MPI_AlltoAll(…);
    // more stuff that matters
    myKernel<<…,…>>
    // even more!
}

exit_measurements(char* function)
{
    // Entry CPU counters
}
```
CUPTI (Profiling GPU)

- **Measurements**
  - Number of instructions executed
  - Number of warps launched
  - Number of threads launched
  - Memory usage
  - ....

- **Limitation**
  - Number of events in a single run
  - Performance degradation while measuring too many events

- **Plan:** Need to get feedback from the user to collect necessary events