Intel Performance Counter Monitor

Introduction

Intel PCM (Performance Counter Monitor) is a tool to monitor performance hardware counters on Intel>® processors, similar to PAPI. The difference between PCM and PAPI is that PCM supports only Intel hardware, but PCM can monitor also uncore metrics, like memory controllers and >QuickPath Interconnect links.

Installed version

Currently installed version 2.6. To load the module, issue :

```
$ module load intelpcm
```

Command line tools

PCM provides a set of tools to monitor system/or application.

pcm-memory

Measures memory bandwidth of your application or the whole system. Usage:

```
$ pcm-memory.x <delay>|[external_program parameters]
```

Specify either a delay of updates in seconds or an external program to monitor.
If you get an error about PMU in use, respond “y” and relaunch the program.

Sample output:

```
---------------------------------------||---------------------------------------
--          Socket 0           --||--          Socket 1           --
---------------------------------------||---------------------------------------
---------------------------------------||---------------------------------------
---------------------------------------||---------------------------------------
--  Memory Performance Monitoring  --||--  Memory Performance Monitoring  --
---------------------------------------||---------------------------------------
-- Mem Ch 0: Reads (MB/s):   2.44 --||-- Mem Ch 0: Reads (MB/s):   0.26 --
--        Writes(MB/s):   2.16 --||--        Writes(MB/s):   0.08 --
-- Mem Ch 1: Reads (MB/s):   0.35 --||-- Mem Ch 1: Reads (MB/s):   0.78 --
--        Writes(MB/s):   0.13 --||--        Writes(MB/s):   0.65 --
-- Mem Ch 2: Reads (MB/s):   0.32 --||-- Mem Ch 2: Reads (MB/s):   0.21 --
--        Writes(MB/s):   0.12 --||--        Writes(MB/s):   0.07 --
-- Mem Ch 3: Reads (MB/s):   0.36 --||-- Mem Ch 3: Reads (MB/s):   0.20 --
```

```
--         Writes(MB/s):   0.13  --||--         Writes(MB/s):   0.07  --
-- NODE0 Mem Read (MB/s):     3.47  --||-- NODE1 Mem Read (MB/s):     1.45  --
-- NODE0 Mem Write (MB/s):    2.55  --||-- NODE1 Mem Write (MB/s):    0.88  --
-- NODE0 P. Write (T/s) :    31506  --||-- NODE1 P. Write (T/s):     9099  --
-- NODE0 Memory (MB/s):       6.02  --||-- NODE1 Memory (MB/s):      2.33  --
---------------------------------------||---------------------------------------
--              System Read Throughput(MB/s):     4.93             --
--             System Write Throughput(MB/s):     3.43             --
--            System Memory Throughput(MB/s):     8.35             --
---------------------------------------||---------------------------------------
```

pcm-msr

Command pcm-msr.x can be used to read/write model specific registers of the CPU.

pcm-numa

NUMA monitoring utility does not work on Anselm.

pcm-pcie

Can be used to monitor PCI Express bandwith. Usage: pcm-pcie.x <delay>

pcm-power

Displays energy usage and thermal headroom for CPU and DRAM sockets. Usage: > pcm-power.x <delay> | <external program>

pcm

This command provides an overview of performance counters and memory usage. >Usage: > pcm.x <delay> | <external program>

Sample output :

```
$ pcm.x ./matrix

 Intel(r) Performance Counter Monitor V2.6 (2013-11-04 13:43:31 +0100 ID=db05e43)

 Copyright (c) 2009-2013 Intel Corporation

Number of physical cores: 16
```

```
Number of logical cores: 16
Threads (logical cores) per physical core: 1
Num sockets: 2
Core PMU (perfmon) version: 3
Number of core PMU generic (programmable) counters: 8
Width of generic (programmable) counters: 48 bits
Number of core PMU fixed counters: 3
Width of fixed counters: 48 bits
Nominal core frequency: 2400000000 Hz
Package thermal spec power: 115 Watt; Package minimum power: 51 Watt; Package maximum power: 180
Socket 0: 1 memory controllers detected with total number of 4 channels. 2 QPI ports detected.
Socket 1: 1 memory controllers detected with total number of 4 channels. 2 QPI ports detected.
Number of PCM instances: 2
Max QPI link speed: 16.0 GBytes/second (8.0 GT/second)

Detected Intel(R) Xeon(R) CPU E5-2665 0 @ 2.40GHz "Intel(r) microarchitecture codename Sandy Bri

 Executing "./matrix" command:

Exit code: 0

 EXEC  : instructions per nominal CPU cycle
 IPC   : instructions per CPU cycle
 FREQ  : relation to nominal CPU frequency='unhalted clock ticks'/'invariant timer ticks' (inclu
 AFREQ : relation to nominal CPU frequency while in active state (not in power-saving C state)='u
 L3MISS: L3 cache misses
 L2MISS: L2 cache misses (including other core's L2 cache *hits*)
 L3HIT : L3 cache hit ratio (0.00-1.00)
 L2HIT : L2 cache hit ratio (0.00-1.00)
 L3CLK : ratio of CPU cycles lost due to L3 cache misses (0.00-1.00), in some cases could be >1.0
 L2CLK : ratio of CPU cycles lost due to missing L2 cache but still hitting L3 cache (0.00-1.00)
 READ  : bytes read from memory controller (in GBytes)
 WRITE : bytes written to memory controller (in GBytes)
 TEMP  : Temperature reading in 1 degree Celsius relative to the TjMax temperature (thermal headr

 Core (SKT) | EXEC | IPC  | FREQ  | AFREQ | L3MISS | L2MISS | L3HIT | L2HIT | L3CLK | L2CLK  | READ

  0   0    0.00  0.64  0.01   0.80   5592     11 K   0.49   0.13   0.32   0.06    N/A    N/A    6
  1   0    0.00  0.18  0.00   0.69   3086    5552    0.44   0.07   0.48   0.08    N/A    N/A    6
  2   0    0.00  0.23  0.00   0.81    300     562    0.47   0.06   0.43   0.08    N/A    N/A    67
  3   0    0.00  0.21  0.00   0.99    437     862    0.49   0.06   0.44   0.09    N/A    N/A    73
  4   0    0.00  0.23  0.00   0.93    293     559    0.48   0.07   0.42   0.09    N/A    N/A    73
  5   0    0.00  0.21  0.00   1.00    423     849    0.50   0.06   0.43   0.10    N/A    N/A    69
  6   0    0.00  0.23  0.00   0.94    285     558    0.49   0.06   0.41   0.09    N/A    N/A    71
  7   0    0.00  0.18  0.00   0.81    674    1130    0.40   0.05   0.53   0.08    N/A    N/A    6
  8   1    0.00  0.47  0.01   1.26   6371     13 K   0.51   0.35   0.31   0.07    N/A    N/A    6
```

```
 9   1    2.30   1.80   1.28    1.29    179 K    15 M    0.99    0.59    0.04    0.71    N/A     N/A     6
10   1    0.00   0.22   0.00    1.26    315      570     0.45    0.06    0.43    0.08    N/A     N/A     6
11   1    0.00   0.23   0.00    0.74    321      579     0.45    0.05    0.45    0.07    N/A     N/A     6
12   1    0.00   0.22   0.00    1.25    305      570     0.46    0.05    0.42    0.07    N/A     N/A     6
13   1    0.00   0.22   0.00    1.26    336      581     0.42    0.04    0.44    0.06    N/A     N/A     6
14   1    0.00   0.22   0.00    1.25    314      565     0.44    0.06    0.43    0.07    N/A     N/A     6
15   1    0.00   0.29   0.00    1.19   2815     6926     0.59    0.39    0.29    0.08    N/A     N/A     6
-------------------------------------------------------------------------------------------------------------
SKT   0    0.00   0.46   0.00    0.79     11 K    21 M    0.47    0.10    0.38    0.07    0.00    0.00
SKT   1    0.29   1.79   0.16    1.29    190 K    15 M    0.99    0.59    0.05    0.70    0.01    0.01
-------------------------------------------------------------------------------------------------------------
TOTAL *    0.14   1.78   0.08    1.28    201 K    15 M    0.99    0.59    0.05    0.70    0.01    0.01

Instructions retired: 1345 M ; Active cycles:  755 M ; Time (TSC):  582 Mticks ; C0 (active,non-h

C1 core residency: 0.14 %; C3 core residency: 0.20 %; C6 core residency: 0.00 %; C7 core residenc
C2 package residency: 48.81 %; C3 package residency: 0.00 %; C6 package residency: 0.00 %; C7 pa

PHYSICAL CORE IPC                 : 1.78 => corresponds to 44.50 % utilization for cores in active st
Instructions per nominal CPU cycle: 0.14 => corresponds to 3.60 % core utilization over time int

Intel(r) QPI data traffic estimation in bytes (data traffic coming to CPU/socket through QPI lin

               QPI0     QPI1     |  QPI0   QPI1
-------------------------------------------------------------------------------------------------------------
SKT    0         0        0      |    0%     0%
SKT    1         0        0      |    0%     0%
-------------------------------------------------------------------------------------------------------------
Total QPI incoming data traffic:   0     QPI data traffic/Memory controller traffic: 0.00

Intel(r) QPI traffic estimation in bytes (data and non-data traffic outgoing from CPU/socket thr

               QPI0     QPI1     |  QPI0   QPI1
-------------------------------------------------------------------------------------------------------------
SKT    0         0        0      |    0%     0%
SKT    1         0        0      |    0%     0%
-------------------------------------------------------------------------------------------------------------
Total QPI outgoing data and non-data traffic:    0

-------------------------------------------------------------------------------------------------------------
SKT    0 package consumed 4.06 Joules
SKT    1 package consumed 9.40 Joules
-------------------------------------------------------------------------------------------------------------
TOTAL:                    13.46 Joules

-------------------------------------------------------------------------------------------------------------
```

```
 SKT    0 DIMMs consumed 4.18 Joules
 SKT    1 DIMMs consumed 4.28 Joules
---------------------------------------------------------------------------------------------
 TOTAL:                  8.47 Joules
Cleaning up
```

pcm-sensor

Can be used as a sensor for ksysguard GUI, which is currently not installed on Anselm.

API

In a similar fashion to PAPI, PCM provides a C++ API to access the performance counter from within your application. Refer to the doxygen documentation for details of the API.

Due to security limitations, using PCM API to monitor your applications is currently not possible on Anselm. (The application must be run as root user)

Sample program using the API :

```
#include <stdlib.h>
#include <stdio.h>
#include "cpucounters.h"

#define SIZE 1000

using namespace std;

int main(int argc, char **argv) {
 float matrixa[SIZE][SIZE], matrixb[SIZE][SIZE], mresult[SIZE][SIZE];
  float real_time, proc_time, mflops;
  long long flpins;
  int retval;
  int i,j,k;

  PCM * m = PCM::getInstance();

  if (m->program() != PCM::Success) return 1;

  SystemCounterState before_sstate = getSystemCounterState();

  /* Initialize the Matrix arrays */
```

```
  for ( i=0; i<SIZE*SIZE; i++ ){
    mresult[0][i] = 0.0;
    matrixa[0][i] = matrixb[0][i] = rand()*(float)1.1; }

  /* A naive Matrix-Matrix multiplication */
  for (i=0;i<SIZE;i++)
    for(j=0;j<SIZE;j++)
      for(k=0;k<SIZE;k++)
        mresult[i][j]=mresult[i][j] + matrixa[i][k]*matrixb[k][j];

  SystemCounterState after_sstate = getSystemCounterState();

 cout << "Instructions per clock:" << getIPC(before_sstate,after_sstate)
 << "L3 cache hit ratio:" << getL3CacheHitRatio(before_sstate,after_sstate)
 << "Bytes read:" << getBytesReadFromMC(before_sstate,after_sstate);

  for (i=0; i<SIZE;i++)
    for (j=0; j<SIZE; j++)
       if (mresult[i][j] == -1) printf("x");

  return 0;
}
```

Compile it with :

```
$ icc matrix.cpp -o matrix -lpthread -lpcm
```

Sample output :

```
$ ./matrix
Number of physical cores: 16
Number of logical cores: 16
Threads (logical cores) per physical core: 1
Num sockets: 2
Core PMU (perfmon) version: 3
Number of core PMU generic (programmable) counters: 8
Width of generic (programmable) counters: 48 bits
Number of core PMU fixed counters: 3
Width of fixed counters: 48 bits
Nominal core frequency: 2400000000 Hz
Package thermal spec power: 115 Watt; Package minimum power: 51 Watt; Package maximum power: 180
Socket 0: 1 memory controllers detected with total number of 4 channels. 2 QPI ports detected.
Socket 1: 1 memory controllers detected with total number of 4 channels. 2 QPI ports detected.
Number of PCM instances: 2
Max QPI link speed: 16.0 GBytes/second (8.0 GT/second)
Instructions per clock:1.7
L3 cache hit ratio:1.0
Bytes read:12513408
```

References

1. https://software.intel.com/en-us/articles/intel-performance-counter-monitor-a-better-way-to-measure-c
2. https://software.intel.com/sites/default/files/m/3/2/2/xeon-e5-2600-uncore-guide.pdf Intel® Xeon® Processor E5-2600 Product Family Uncore Performance Monitoring Guide.
3. http://intel-pcm-api-documentation.github.io/classPCM.html API Documentation