CS 280 ASSIGNMENT 1:

Hardware performance counters

NAME: Xiaopeng Xu KAUST ID: 129052 Deadline: 6th April, 2014

1. Task1:

Use PAPI to find out about the Neser cluster system information: processor type, frequency, OS, cache sizes, ...

I write code to gather this information. My results are as follows:

```
Number of CPU's in an SMP Node:     8.
Number of Nodes in the entire system:     1.
The rate of CPU's:      4197563.
Total number of CPU's in the entire system:
Vendor number of CPU:   1.
Vendor string of CPU:   GenuineIntel.
Model number of CPU:    23.
Model string of CPU:    Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz.
Revision of CPU:  10.000000.
Total number of memory levels:     4.
Level 1 cache 1 type:   2.
Level 1 cache 1 size:   32768.
Level 1 cache 1 line size:    64.
Level  1 cache 1 number of lines:   512.
Level  1 cache 1 associativity:     8.
Level 1 cache 2 type:   1.
Level 1 cache 2 size:   32768.
Level 1 cache 2 line size:    64.
Level  1 cache 2 number of lines:   512.
Level  1 cache 2 associativity:     8.
Level 2 cache 1 type:   3.
Level 2 cache 1 size:   6291456.
Level 2 cache 1 line size:    64.
Level  2 cache 1 number of lines:   98304.
Level  2 cache 1 associativity:     24.
Level 2 cache 2 type:   0.
Level 2 cache 2 size:   0.
Level 2 cache 2 line size:    0.
Level  2 cache 2 number of lines:   0.
Level  2 cache 2 associativity:     0.
Level 3 cache 1 type:   0.
Level 3 cache 1 size:   0.
Level 3 cache 1 line size:    0.
Level  3 cache 1 number of lines:   0.
Level  3 cache 1 associativity:     0.
Level 3 cache 2 type:   0.
Level 3 cache 2 size:   0.
Level 3 cache 2 line size:    0.
Level  3 cache 2 number of lines:   0.
Level  3 cache 2 associativity:     0.
Level 4 cache 1 type:   0.
Level 4 cache 1 size:   0.
Level 4 cache 1 line size:    0.
Level  4 cache 1 number of lines:   0.
Level  4 cache 1 associativity:     0.
Level 4 cache 2 type:   0.
Level 4 cache 2 size:   0.
Level 4 cache 2 line size:    0.
Level  4 cache 2 number of lines:   0.
Level  4 cache 2 associativity:       0.
```

Also I can get these info by run papi_avail and papi_mem_info in neser.

But I cannot get the OS info. I didn't find any function that can return OS info. Also I searched the website, and found that PAPI seems not supply such access on user level.

2. Task2:

Download the rtm_kernel code from blackboard and get acquainted to it (make and run). The code solves a wave equation used in the field of Seismic imaging. More details about the application will be presented in a separate lecture in the last weeks of the semester

Instrument the rtm_kernel in order to use hardware performance counters to determine the behavior of the time loop of the code (the timed section of the code)

The hardware performance counters should be based on the PAPI library, and you should monitor the following values:

- Level 1 total cache accesses and misses (include instruction and data caches)
- Level 2 total cache accesses and misses (include instruction and data caches)
- Conditional branch instructions
- Number of floating point operations

Answer:

I counted data cache misses and instruction cache misses separately. For each value, I counted 3 times. Below is the result I get:

L1 Instruction Cache Access	L1 Instruction Cache Miss	L2 Data Cache Access	L2 Data Cache Miss	L2 Instruction Cache Access	L2 Instruction Cache Miss	Conditional Branch Intrunction	Floating Point Operation
1,581,588,679,170	1,012,684	57,787,590,824	14,095,517,975	2,228,978	865,094	40,483,114,304	687,353,282,394
1,580,856,195,708	1,029,563	57,664,483,425	14,095,310,139	2,337,753	866,442	40,483,120,541	687,353,378,870
1,578,854,314,000	1,050,634	57,646,500,543	14,095,512,704	2,322,000	876,007	40,483,115,012	687,352,755,453

3. Task 3:

Edit the makefile to link with PAPI. To link a simple code with PAPI use

gcc example.c –lpapi –o example

Compile your code with different optimization flags

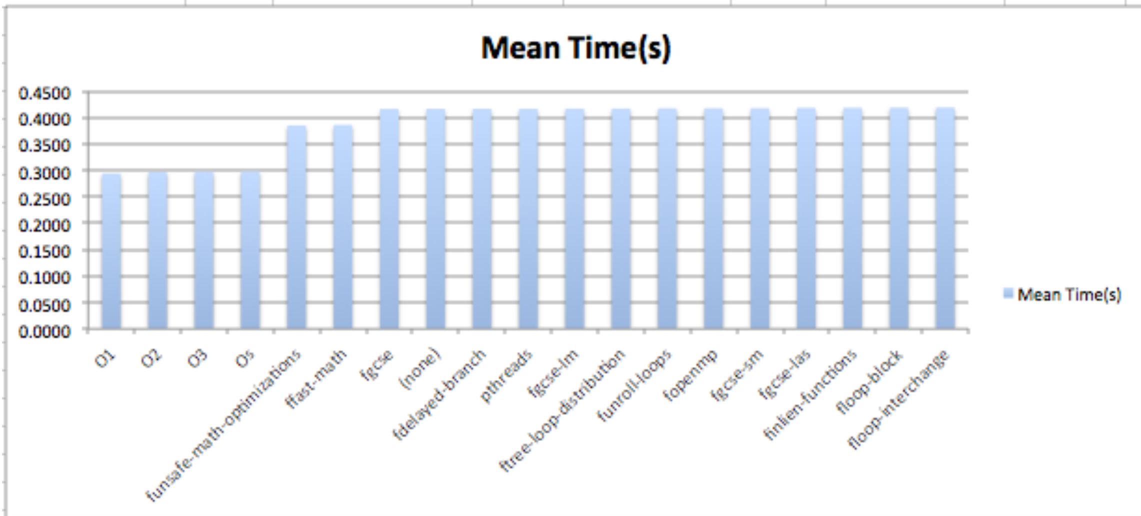
- The predefined "–O's" optimization levels are required
- Among the flags listed in lecture 10, find out at which level are these included. If an optimization is not included in any – O level, it should be investigated.
- Some optimizations (included in the –O's) are parameterized, the effect of modifying such parameters on relevant events is to be investigated

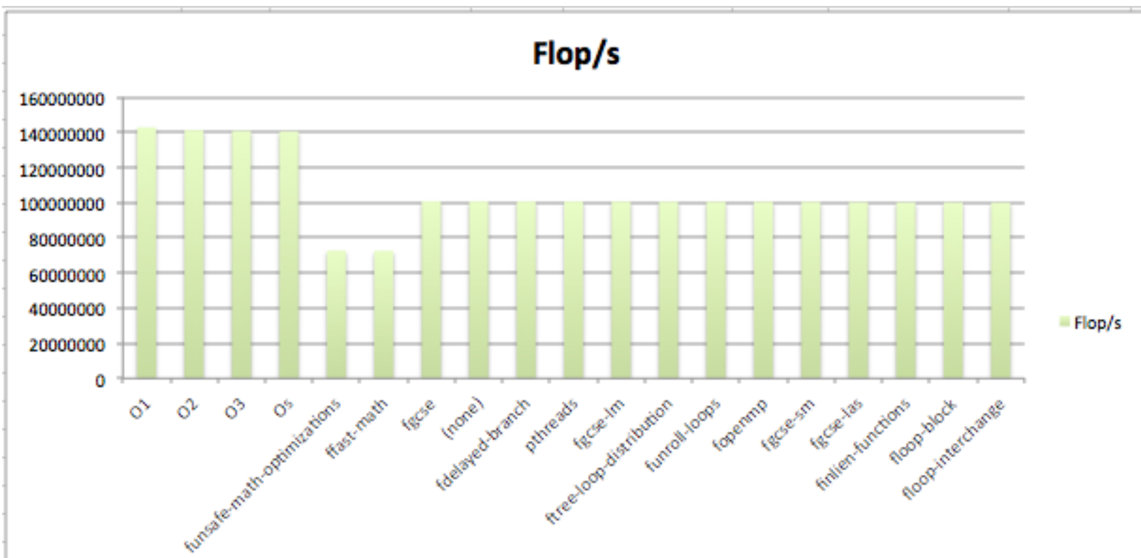
Run the modified code on the Neser cluster and generate graphs for the requested events as a function of the applied optimizations in an incremental way. (get averages of multiple measures)

Compute the actual performance of the code snippet in terms of flop/s (average of multiple measures)

Answer:

I used a simple code snippet to try different optimization flags (run 3 times each). The code is a matrix multiplying with 3. Details of the values I get is in Q2&Q3.xlsx. Here's the charts I got:





4. Task 4:

Write a summary of the presentation given by Peter Ungaro, CEO of Cray inc. on 25/2/2014 about "The Fusion of Supercomputing and Data Analytics To Drive Scientific Discovery".

Half page minimum, one page maximum with regular/standard fonts, font sizes (11-12), line spacing (single to 1.5) and margins (1 inch on all sides)

In the lecture "The Fusion of Supercomputing and Data Analytics To Drive Scientific Discovery", Peter Ungaro introduce Cray's vision and work in the field of supercomputing and data analytics.

First, he introduced big data and the methods and solutions for big data analysis to us. He explained us what's big data and why should we care about big data. Big Data refers to data that is not easily captured, managed and analyzed by traditional tools due to large volume, fast production velocity, and big variety. Then he told us the three challenges in the fusion of supercomputing and big & fast data: advanced analytic appliances, storage & data management, and fast computing by supercomputers. Also data-intensive processing is driving the need for advanced architectures, because it is hard to get data by simple architecture. Big data's methods for analysis like discovering needle in a needle stack, it needs a lot of technological advancements.

Then, he compared supercomputing with large-scale data analysis and told us why "Cray" is necessary for this large-scale data analysis. At first, he talked about different solutions for advanced analytics, like Data Warehouses, NoSQL Databases, Hadoop/MapReduce, and Graph Analytics. Then he introduced the system architecture differences between supercomputing and large-scale data analytics, such as for supercomputing data movement is minimized -- only load the "mesh" into memory, for large-scale data analytics, data movement is maximized – all the data are scanned/sorted/streamed all the time. And applications differences come in several aspects, such as computational requirements far outweighed by capacity, bandwidth or latency requirements of memory hierarchy, applications are either memory-bounded, interconnect-bounded, or I/O-bounded on a given distributed platform, and it is mirror opposite of compute-bound problems -- FLOPS matter less. After that he explained why "cloud" couldn't do this—because convergence is at the high-end. Also there is a trend of big data going to fast data, which required the need for bigger global memory and fast interconnection. Extending adaptive supercomputing to big data workloads needs change from bottom-level change of scalable system infrastructure to up-level apps and programs.

After that, he showed us some of Cray's work on large-scale data analysis. Cray cluster supercomputers for Hadoop, which was built with high performance, great reliability, and easy maintenance, turned the Hadoop in to a "gold" Hadoop. They also add value at the edge of the network to adapt to data-intensive computing.

Finally, he showed us Cray's roadmap for this "Fusion" and their blueprint. Cray's roadmap "Fusion" is to combine workflows & data into a single system, add aggressive local and global memory capabilities, and also tightly integrate software stack (& runtimes) across all 3 capabilities – data ingest, simulation, and analysis. Then he told us Cray's vision – to build a world-class integrated supercomputing environment that enables transformational computing across a broad set of science, engineering and advanced analytics (big data) applications.