ECE571: Advanced Microprocessor Design – Homework 3
Fall 2019

Due: Thursday 26 September 2019, 2:00pm

1. Background

   - For this assignment, log into the Haswell-EP machine just like in HW#1.
   - Create a document that contains the data and answers described in the sections below.

2. Measuring Power using perf/RAPL

   - Run the following 4 benchmarks, gathering the Energy and time values from `perf` for energy-cores, energy-pkg and energy-dram. Use these values to calculate the average power for each. Note: the backslashes are a line-continuation character. Leave them in if you are cutting/pasting in one big chunk, but leave them out if you are hand-typing the commands.
     - sleep
       ```
       perf stat -a -e power/energy-cores/,\
       power/energy-pkg/,power/energy-ram/ sleep 10
       ```
     - stream
       ```
       perf stat -a -e power/energy-cores/,\
       power/energy-pkg/,power/energy-ram/ \
       /opt/ece571/stream-5.10/stream_c
       ```
     - matrix-matrix multiply
       ```
       perf stat -a -e power/energy-cores/,\
       power/energy-pkg/,power/energy-ram/ \
       /opt/ece571/matrix_multiply/matrix_multiply_atlas 100
       ```
     - iozone
       ```
       perf stat -a -e power/energy-cores/,\
       power/energy-pkg/,power/energy-ram/ \
       /opt/ece571/iozone3_417/src/current/iozone -a -g 256k
       ```
   - Make a table with the 3 rows being cores, pkg, ram and the columns being the 4 benchmarks, list the Energy from each.
   - Make another table like above, but instead of Energy use the Energy and time values to report the average Power.
   - Answer the following questions:
     (a) Which benchmark causes the package to use the highest average power?
     (b) Which benchmark causes the RAM to use the highest average power?
     (c) Was there anything weird with the cores result? Why might you see this result?
     (d) Some Intel machines will report GPU power. Why might that not be available on this machine?

3. Calculating Energy-Delay and Energy-Delay-Squared

   - For this problem we will use the `equake_l` version of equake that is from the SPECOMP2001 (SPEC OpenMP) benchmark suite (note: that's a lowercase L, not a 1, in the benchmark name). It is a version of the equake benchmark parallelized with OpenMP directives so that it can take advantage of multiple cores.
   - Run the benchmark with 1, 2, 4, 8, 16, and 32 threads, measuring the energy-cores RAPL counter. The command line for doing this with 1 thread is (all one one line):
     ```
     env OMP_NUM_THREADS=1 perf stat -a -e power/energy-pkg/ \
     /opt/ece571/equake_l.specomp/equake_l \
     < /opt/ece571/equake_l.specomp/inp.in
     ```
     Change the `OMP_NUM_THREADS` value to change the number of threads. You probably want to make sure no one else is logged in and running these tests (use `w` or `top` to check) at the same time you are, otherwise you can affect each other's results. Note: running the benchmark will take a while (up to a few minutes).
   - Create a table that for each thread count (1,2,4,8,16,32) shows the elapsed time, the Energy in Joules, the Energy-Delay value, and the Energy-Delay-Squared value.
   - Answer the following questions:
     (a) Which thread count has the fastest time?
     (b) Which thread count has the lowest energy?
     (c) Which thread count has the lowest energy-delay?
     (d) Which thread count has the lowest energy-delay^2?
     (e) How well does this benchmark scale when adding additional threads? What could explain the scaling behavior you see?
     (f) If you check /proc/cpuinfo you can see Linux sees 32 cores in the system. Why did the 32 thread run not run much faster than the 16 thread run?

4. Submitting your work.

   - Create the document containing the data as well as answers to the questions asked.
   - Please make sure your name appears in the document.
   - e-mail the file to me by the homework deadline.