ECE575: Cluster Computing – Homework 8
GPGPU and Energy

Due: Friday 11 December 2015, 5:00pm

1. **GPGPU/CUDA (3pts)**

   - For this first part, you will need to log into the "Quadro" machine in my lab, which is a Haswell machine with a Quadro NVidia card.
     To get to this machine, first log into the same Haswell machine we used in previous homeworks. As a reminder, use the username handed out in class and ssh in like this
     `ssh -p 2131 username@vincent-weaver-2.umelst.maine.edu`
   - Next you will need to ssh one more time, through the reverse-ssh tunnel. Note that this is similar to logging into the pi-cluster, but in this case the port number is 19999 not 19998.
     `ssh -p 19999 localhost`
     Your password should be the same as it was on the Haswell machine.
   - Download the code template from the webpage. You can do this directly via
     `wget http://web.eece.maine.edu/~vweaver/classes/ece574_2015f/ece574_hw8_code.tar.gz`
     to avoid the hassle of copying it back and forth.
   - Decompress the code
     `tar -xzvf ece574_hw8_code.tar.gz`
   - Run `make` to compile the code.
   - The code in question is a SAXPY code, that is it does a single-precision (32-bit floating point) vector multiply plus add, i.e. $y[i] = a * x[i] + y[i]$
   - The code provided is both a C version and a CUDA version. Take a look at the CUDA version to review that you remember how CUDA code works.
   - By default a vector size of 1 million is being used. Both the C and CUDA version take a command line argument that specifies how many repetitions of SAXPY to run.
   - Use the time command to measure the wall-clock time needed for 1,2,4,8,16,32,64,128,256,512,1024, and 2048 iterations for both the C and CUDA versions.
   - You should see results similar to this:
     1 repetition C=0.033s, cuda=0.141s
     2048 repetitions C=11.65s cuda= 3.184s

   (a) Why is the C version faster with only 1 repetition?

   (b) Why is the Cuda version faster for 2048 repetitions?

   (c) What is the crossover point where Cuda is faster than C?

   (d) How could you improve the performance of the C version?

2. **Power/Energy (3 points)**

   - Re-run the SAXPY results for 2048 repetitions, but measuring energy usage. To measure the CPU energy being used, you might use a command line like this:
     `perf stat -a -e cycles,power/energy-pkg/ ./saxpy 2048`
     You should see results similar to this:

Type	Energy	Time
C	337.69J	11.605s
cuda	84.29J	3.184s

   - You can measure the power used by the GPU with the `nvidia-smi` tool. Try running it to look at the results. You can get the power results specifically with the following command:
     `nvidia-smi --query-gpu=utilization.gpu,power.draw --format=csv -lms 100`
     If you run that in one window while running SAXPY in another you will see that the CUDA SAXPY code uses 17.5W.

   (a) How much total CPU+GPU energy is consumed by the C implementation?

   (b) How much total CPU+GPU energy is consumed by the GPU implementation?

   (c) Are these results what you would expect?

3. **Big Data / Hadoop (2 points)**

   (a) What two major operations are used by Hadoop?

   (b) What language is used when writing Hadoop code?

   (c) Name one benefit of a distributed filesystem (such as the hadoop HDFS filesystem) over a centralized filesystem such as NFS.

4. **Reliability / Checkpointing (2 point)**

   (a) Why might a cluster located at an observatory at the top of Mauna Kea in Hawaii have a higher failure rate than an identical cluster located at UMaine?

   (b) List a benefit to using application-level checkpointing in your code.

   (c) List a downside to using application-level checkpointing in your code.

5. Submitting your work.

   - Send me a document (pdf, txt, docx) including the data asked for and answers to the questions by the deadline. Please e-mail your document to me.