ECE574: Cluster Computing – Homework 2

Due: Thursday 17 September 2014, 3:30PM

1. Background

   - For this assignment, log into my Haswell machine as described on the account slip that I handed out in class.
     On Linux or OSX you will do the following (replace username with the one on the slip):
     `ssh -p 2131 username@vincent-weaver-2.umelst.maine.edu`
     On a Windows machine you'll want to get a program such as `putty`, some directions can be found here, be sure you connect to port 2131:
     `http://web.eece.maine.edu/~vweaver/classes/ece571_2013s/using_ssh.html`
   - We will be using the HPL Linpack benchmark for this test.
   - Create a document that contains the data described in the Analysis sections below. A .pdf or .txt file is preferred but I can accept MS Office format if necessary.

2. Aggregate measurements

   - `time` tool
     - First copy xhpl and HPL.dat to your local directory:
       `cp /opt/ece574/hpl/* .`
     - That command should also have copied a file called `time_hpl.sh`. We will be using the "slurm" job submission tool in this class (more discussion of this will happen in class on Tuesday). The Haswell machine is a shared machine, and the job submission tool makes sure that there are never more jobs running than cores are available.
     - To submit a job, you create a shell script (such as the provided `time_hpl.sh`) and submit it with a command such as:
       `sbatch ./time_hpl.sh`
       The job will be queued up (you can check the queue with the `squeue` command.
       Once it runs, the output will be created in the current directory with a filename such as `slurm.haswell.X.out` which contains the output, where X is the job number.
     - The time tool actually prints its output to stderr, which can be found in `slurm.haswell.X.err`
     - Try submitting the job and looking at the output just to make sure it works.
     - You can change the number of threads used by this version of HPL by setting the `OMP_NUM_THREADS` environment variable. You can do this by putting a line like this: `export OMP_NUM_THREADS=1` in the shell script just before the line that runs xhpl.
     - Analysis:
       (a) Run xhpl as described above with 1, 2, 4, and 8 threads.
           Record the values in your writeup.
       (b) List the speedup of 2, 4, and 8 threads (versus 1 thread).
       (c) List the parallel efficiency of 2, 4 and 8 threads.
       (d) Does this benchmark show strong scaling? Why or why not?

(e) Do the results gathered contain enough info to say if the benchmark exhibits weak scaling?

- Profiling with perf
  - Create or modify a script that runs
    `perf record`
    on HPL while measuring 4 threads.
  - Use `perf annotate` and answer the following questions.
  - Analysis
    (a) What instruction is reported as taking the most time?
    (b) Is it believable that that instruction would be a hotspot in the code?

3. 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.