High Performance Computing

# Assignment 1

# High Level basic theory

(This stuff is just to make sure you bother to retain key concepts from the lectures/text)

1. Partially driven by the approaching end of Moore’s law, and partially driven by changes in semiconductor manufacturing, several engineering research papers have attempted to propose alternatives to the semiconductor node sizes measured in nanometres. (E.g. Ryzen 3 were made on a *7nm* node, Intel at the time was using *10 nm*). Discuss why a new measurement system is necessary, and what you think are appropriate alternatives and why, either discuss your own idea or use one of the existing papers as a basis for your opinion. (TL;DR so you know you’re on the right track: it used to be that since transistors made in 2D planes, and the height and width shrunk at the same rate, one number squared corresponded to the inverse of the density, so 7nm was 2x the transistor density of 10nm, 5nm is 2x the density of 7nm, but that underlined bit is not the case anymore and so we need a better measure). Up to 1000 words answer.

1. Based on <https://www.sharcnet.ca/my/systems/show/114> Graham (the main Sharcnet cluster) uses the following for compute nodes:

* Intel E5-2683 v4 (Broadwell) @ 2.1 GHz X2 (so dual socket 16 core CPUs)
* 2 × NVIDIA Pascal P100 GPUs (12GB HBM2) (only some nodes)
* Memory: 128.0 GB (RAM)
* Local storage: 1.2TB.

These are getting old. A reasonable replacement for this would be something using a dual socket AMD Epyc 9000 series CPU. Have a look at <https://www.siliconmechanics.com/systems/gigabyte/serve> and configure 2 different machines one with a single socket, the other with 2 sockets, for part of this consider 2 GPUs per socket and also 0 gpus per socket, make reasonable estimates about RAM per socket (512GG probably) and storage (probably 2-4TB), and go for the ‘best’ you can configure. The fastest CPU would be the Ryzen 9654 (96 core) CPU, that’s about 12000 USD per CPU.

Sharcnet is configured with 800 nodes using no GPUs, and about 150 more with GPUs (and about 100 other machines which we will ignore for now, that are for testing, login, extra memory configurations etc).

Estimate the cost of replacing the 950 nodes (800 nodes without a GPU 150 with a GPU) mentioned with their modern equivalents and attempt to find (from a vendor either siliconmechanics or someone else) something close to the mentioned configuration to justify your answer (Historically Fujistu, IBM, NUDT, Dell EMC, Cray/HPE and Huawei are the biggest vendors in this business, but they all seem to be lagging on new hardware right now).

1. Estimate the power cost for running the most expensive (2 socket + 4 GPU setup) 24/7 for a year. Don’t worry about trying to price out interconnects to other servers or the cost of a rack itself.

Assuming the power supplies in your system draw full power 100% of the time (they wouldn’t in the real world) what is the power cost per week of using electricity rates from:

<https://www.torontohydro.com/for-business/rates>   
  
(you need to do some simple math to figure out the cost for each of the 168 hours in a week and multiply by electricity costs in kilo Watt hours into the same form as power in Watts).

# Programming

1. Numerical method in C++ on Linux/ (If you are using Python and WSL do the same thing, but well, in python).

We’re going to do our own cpu benchmarking program.  
  
This is a Linear Algebra – (solutions to PDE’s), type problem.

Write a C++ program that generates random 10x10 systems of linear equations and solve with with Gauss Jordan elimination. (Note this means that your actual matrix will have 11 columns and 10 rows).

<http://www.cplusplus.com/reference/cstdlib/rand/>

<https://www.geeksforgeeks.org/program-for-gauss-jordan-elimination-method/>  
  
(you probably need it you can find ‘how to generate random matrices in C++ on the web too)

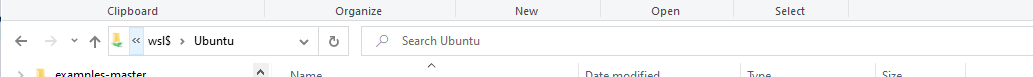
1. Throughput – every 1 second print off how many times it was able to run the solutions, run this for 10 seconds.
2. Capacity computing (job time) – calculate how long it takes for 100, 200, 500, 1000, 2500, 5000, 7500 and 10 000 runs of the calculation. (If that takes absurdly long on your machine pick smaller numbers, I don’t actually care what it is, it seems like 10 000 should take about 5 minutes). Plot the result (feel free to use excel for the plot).
3. Parallelize with OpenMP and see how much (if any) improvement you get to one of the longer calculations in b).

# Linux Commands

1. Demonstrate a basic understanding of Linux Commands based on the following

<https://www.thegeekstuff.com/2010/11/50-linux-commands/?utm_source=feedburner>

(If you’re using WSL and you need to find the files it outputs, make sure you start WSL and then in this box



On the left type [\\wsl$](file:///\\wsl$) which will then show you “Ubuntu” as a network location (this breaks if you’ve got VPN turned on btw, within that, your stuff is likely stored in /home/yourusername)

1. Using the Linux pipe command, output the result of your program in 4 to a file (even if your program does not work you should know how to do this). Paste that file here.
2. While your program in 4 is running doing its thing (or something else is running, doesn’t matter what) use *ncdu* and *top* to see how much memory and what percentage of CPU your program uses. Show your output here. (You can do that with WSL if you need to).
3. Use Grep to find all instances of the word “and” on the syllabus (you’ll need to save the syllabus to .txt first), and print out the lines they are on.
4. Tar up your assignment and submit it (name it for your trent username.tar.gz), on future assignments feel free to use zip or rar like a normal person.