

**EECE5640**  
**High Performance Computing**  
**Homework 1**

**\*Submit your work on Turnitin on Blackboard**

1. (30) In this problem, you are to select a set of 2 single-threaded benchmark programs. Try to provide some diversity in the set of workloads you have chosen (e.g., floating point, integer, memory intensive, sparse). Then complete the following question using these benchmarks. Make sure to provide as many details as possible about the systems you are using, and where you obtained the source code for the benchmarks.
  - a. Compile and run these 2 benchmarks on any Linux-based system of your choosing (you can use either the COE systems or the Discovery systems). Provide detailed information about the platform you choose, including the model and frequency of the CPU, number of cores, the memory size, and the operating system version. You should record the execution time, averaged over 10 runs of the program. Is any run of the program faster than another? If so, comment on any differences you observe in your write-up, providing some justification for the differences.
  - b. Next, explore the compiler optimizations available with the compiler on your system (e.g., gcc), and report on the performance improvements found for the 2 workloads. Describe the optimization you applied, and provide insight why each of your benchmarks benefitted from the specific compiler optimization applied.
  - c. Assume that you were going to rewrite these applications using pthreads. Describe how you would use pthreads to obtain additional speedup by running your benchmark on multiple cores.

\*Answers to this question should be included in your homework 1 write-up in pdf format, and submitted through Turnitin.
2. (30) In this problem, you will start by selecting a sorting algorithm to sort 10,000 random integers with values 1-10,000. We will use pthreads in the problem to generate a parallel version of your sorting algorithm. If you select code from the web (and you are encouraged to do so), make sure to cite the source of where you obtained the code. You are also welcome to write your own code from scratch. You will also need to generate the 10,000 random integers (do not time this part of your program). Make sure to print out the final results of the sorted integers to demonstrate that your code works (do not include a printout of the 10,000 numbers). Provide your source code on the link for Code-1.2 in Turnitin.

- a. Run your program with 1, 2, 4 and 8 threads on any system of your choosing. The system should have at least 4 cores for you to run your program on. Report the performance of your program.
- b. Describe some of the challenges faces when performing sorting with multiple threads.
- c. Evaluate both the weak scaling and strong scaling properties of your sorting implementation.

\*You should include your C/C++ program for this problem using the Code-1.2 Turnitin link. Also include your written answers to questions 2 (a-c) in your homework 1 write-up.

3. (10) Select a node on the Discovery Cluster and report the following information about that node: (please do not ask the system admins for this information)
  - a. The CPU model
  - b. The cache memory hierarchy, including sizes and associativity of each cache.
  - c. The bandwidth and latency of the network interconnect for this node.
  - d. The version of Linux installed.

\*Answers to this question should be included in your homework 1 write-up in pdf format, and submitted through Turnitin.

4. (30) Find the current listing for the Top500 list for 2017. Review the system architecture for each of the top 10 systems on this list. Provide a summary of trends that you detect, and discuss what kind of system you would design in the future in order to be included in this impressive list. Make sure to provide a diagram of the architecture you would develop, and include details of the CPUs, memory and interconnect used.

\*Answers to this question should be included in your homework 1 write-up in pdf format, and submitted through Turnitin.