ECE574: Cluster Computing – Final Project

Due: Friday, 3 May 2019 (Last day of Classes)

Overview:

- Design a project that involves cluster or parallel computing.

Guidelines:

- You may work either alone or in a group. If you work in a group your end project will have higher expectations.
- You may use any of the computers used in this class, or any other computer you may have access too. If you might need access to more exotic hardware let me know and I can see what I can do.
- You may use any programming language or parallel toolkit that you like, however if you use something other than the ones used in class I might not be able to provide as much help. So feel free to use Matlab, C++, Fortran, Python, etc.
- The project can be related to your thesis research or senior project. However it cannot be just a stripped down subset of your research, it has to be something new that has been developed for this class.

Part 1: Topic Selection (due 28 March 2019) (5pts)

Each group should send a brief e-mail describing your project topic and listing group members.

Part 2: Progress Report / Related Work (due 18 April 2019) (10pts)

A brief status update detailing progress your group has made. This is primarily to make sure your project is on track to be finished in time; if things are not going well the topic can be adjusted.

Also provide some "related work" for your project. Do a literature search and find examples of other people who have done similar research.

I'd prefer if the references you find are books or academic papers, but (especially for programming projects) blog or website references are OK.

Send this report by e-mail. Only one submission is needed per group.

1. State in one sentence a summary of your project.
2. Describe the hardware and software that you will be using.
3. Have you made any progress on running code in such a setup?
4. The related work. Two references are fine for a single-person project, at least four if you have more than one person in your group.
5. Will you be willing to volunteer to present early (Thursday the 25th or Tuesday the 30th)?
6. Submit the status update by e-mail.

An short example of roughly what I expect:

Our project is a supercomputer made out of 30 year old Apple IIe computers. We have the computers hooked up via Ethernet and we have started implementing the benchmark code in BASIC.

We are willing to present on Tuesday.

Related Work:

Our research involves making a high-performance Apple II cluster.

Michael Mahon[1] constructed a cluster of 17 Apple II machines that use the joystick interface for networking. Our cluster will use Ethernet instead.

J.R. Vokey[2] wrote some advanced matrix multiplication libraries for Apple II basic. We plan to do something similar.

C. Marck[3] used an Apple IIe to do DNA analysis. We plan to do physics simulations, not biological ones.

[1] M. Mahon. **Apple Crate II**. `http://michaeljmahon.com/NadaNet.html` `http://michaeljmahon.com/AppleCrateII.html`

[2] J.R. Vokey, **MATRICKS: Matrix algebra for Applesoft BASIC**. Behavior Research Methods, Instruments, & Computers (1986) 18: 409. doi:10.3758/BF03204431.

[3] C. Marck, **Fast analysis of DNA and protein sequence on Apple IIe: restriction sites search, alignment of short sequence and dot matrix analysis**, Nucleic Acids Res (1986) 14 (1): 583-590.

Part 3: In-Class Presentation 30 April & 2 May 2019 (40pts)

- You will have 15 minutes to present. Plan for 10 minutes of presenting plus 5 minutes for questions.
- You may present slides using the projector if you want, but that's not strictly necessary.
- You should cover the following things (but feel free to include more):
  - An overview of exactly what you were doing.
  - The hardware that you ran on (CPU type, if it is a cluster, number of cores, RAM, etc)
  - The software that you used (programming language, parallel libraries such as OpenMP, MPI, cuda, etc), and math libraries (like OpenBLAS) or performance libraries (like PAPI)
  - Related work: list any similar projects, and how your project differs.
  - Results of your experiments. Tables or graphs are nice. Ideally also include performance comparison (time) but also energy/power if possible.

- Future work. If you had more time, what else would you do.

Part 4: Project Writeup, Officially due 3 May 2019 (45pts)

This will be a short paper (at least 6 pages, but you can include pictures, diagrams, etc.) that must contain all of the following:

1. Introduction: What your project is and what the goal was.
2. Related work. List and properly cite at least three related projects that did similar work (academic papers would be best, but you can also cite websites and such). Describe how your work differs from the previous work.
3. Hardware
   (a) Describe the computing hardware that you run on. Say if it is shared memory, a distributed system, or something else. List the CPU (architecture, type, speed), RAM, and network.
4. Software
   (a) Programming Language: Which one did you use? Why?
   (b) Any reliability issues? What would happen in face of a hardware/software error?
   (c) Also list the versions for the operating system and other software. This is useful if anyone tries to reproduce this work down the road.
5. Project Overview and Results
   (a) Write about what you did.
   (b) Write about any results you obtained.
   (c) If your project involves performance, give time and power comparisons if possible.
   (d) Pictures and graphs are useful, if applicable.
   (e) Feel free to write about things that *didn't* work. That's still of interest too.
6. Conclusion
   (a) If you worked in a group: List who worked on what part.
   (b) Summarize what you did.
   (c) Future Work: List any improvements you might make if you had more time and resources to work on the project.
7. Appendix
   (a) I would like a copy of any relevant source code if possible. (this can be submitted as a separate file, does not have to be included in the report).

You can e-mail your final report to me. pdf or word document is fine, the code should be attached too.

Potential Project Ideas:

- Take some code that you are interested in and make it parallel.
  - Image Processing
  - Fractals
  - Raytracing
- Take some existing parallel code and make it run faster (by using performance measuring tools and finding bottlenecks).
- Take some code of interest and make it parallel using different languages and compare the tradeoffs (for example, MATLAB vs C or C vs Python).
- Take some existing parallel code (for example the sobel code in class) and hand-optimize using SIMD assembly instructions (either SSE/AVX on x86 or NEON on ARM).
- Build a cluster out of some machines that you have and run parallel jobs on them.
- Take some code and make it run on a GPU (CUDA, OpenCL)
- Graphical MPI visualization tool for the Pi cluster (to demo to visiting students the power of parallel computers)

Projects from Previous years:

- Raspberry Pi Cluster
- Older hardware cluster
- Parallel FFT
- Pi Docker swarm
- Hybrid MPI/GPU cluster
- Parallel fractal
- Parallel ocean vegetation modeling
- Parallel Octave/Matlab calculations