Term Project Description
CS3339 Fall 2021
Proposal Due: Friday Oct 15 11:59 PM
Full Report Due: Friday Dec 10 11:59 PM

1 Project Proposal

The project proposal is a description of the proposed project. It should be 2-3 pages long at an 11pt font. The proposal must include the following four sections.

1.1 Problem Description

- Describe the problem you want to solve or the question you want to answer. (e.g., *We will conduct a performance evaluation study of Intel and AMD based multicore systems*).
- Make a case for why the work you are doing is interesting or useful (*e.g., Multicore technology has caused a fundamental shift in all computing domains. There are two major manufacturers of high-end multicore chips today. It would be interesting to find out, if systems built by one of the manufactures have significant advantages over the other*).
- Mention expected results and/or specific goals. (*e.g., We expect AMD systems to perform better on FP applications because ...*)

1.2 Methodology

- Outline specific tasks. (*e.g., need to find machines, read up on Intel and AMD architecture specifics, learn to use simulation tool, set up a controlled environment for evaluation etc.*)
- Mention which *specific* benchmark, machine, simulator and performance analysis tool you will use.

1.3 Deliverables

The deliverables will depend on the nature of the project. Each project will have at least two deliverables: (i) an in-class presentation (6-7 mins) and (ii) a written report describing the methodology, experimental results and analysis. If you are developing software then the source code along with appropriate documentation is the main deliverable and the presentation will essentially be a demo of the software. Some projects can have both a report and the code as deliverables. You will receive more details on what to include in your reports after you have submitted your proposal.

1.4 Team

- List the team members and their specific roles. Some overlap in responsibility is OK but complete overlap should be avoided.
- Responsibilities should map to the tasks listed in Section 1.2. Each member should be involved in the presentation. One team member can take the lead in writing the report but all members should contribute.
- Part of the project grade will be based on peer evaluation. So assign roles carefully.

2 Submission

- Project proposals must be submitted as a single PDF. Upload the PDF in the file upload section in the Canvas Assignments titled Project Proposal. Only one member of the team needs to submit.
- The presentations are scheduled for the last week of classes Nov 29 and Dec 1. The actual schedule will be determined later in the semester via lottery.
- Final project reports should also be submitted as a single PDF. Upload the PDF in the file upload section in the Canvas Assignments titled Project Presentation and Final Report. Only one member of the team needs to submit.

- If the final deliverable consists solely of code and documentation upload a text file with the title of the project and a link to the TX State Git repository. Make sure `aq10` has read access to the repo.

3 Grading

The term project is worth 20% of your final grade. The breakdown of the project grade is given below.

Proposal	20%
Presentation	20%
Peer Evaluation	20%
Project Report/Code	40%

The peer evaluation is your opportunity to grade the contributions of your teammates. You will submit the evaluation as an addendum in the final exam (so that they remain anonymous).

4 Project Ideas

A. Programming

1. Implement a simulator for architecture X. Instruction-by-instruction simulation of an assembly program. You can choose either a pipelined (for simpler ISAs) or a non-pipelined (for more complex ISAs) version of the architecture to implement. Detailed guidelines may be available.

2. Implement a disassembler for architecture X. Given a binary file, the disassembler will generate the corresponding assembly instructions.

3. Implement a GUI-based dependence analysis tool. Given an assembly file the tool will highlight all types of data dependencies in the code.

4. Develop a mobile app for processor datapath design.

B. Tools

1. *Simplescalar:* Choose benchmark suite and conduct a cache simulation study using Simplescalar.

2. *PIN:* Use PIN Tools to implement a reuse distance analyzer/branch predictor.

3. *Perf:* Use perf to study the performance of a benchmark suite X. Show effects of Amdahl's law.

C. Performance Evaluation

1. *Dual-core vs Quad-core:* Choose a parallel benchmark suite. Evaluate performance as a function of number of available cores, number of threads etc.

2. *Intel vs AMD:* Choose a benchmark suite and compare performance on Intel and AMD-based platforms.

3. *CISC vs. RISC:* Select benchmark suite and evaluate on MIPS, VAX, x86 etc. Compare performance in terms of instruction count and CPI

4. *CPU vs GPU:* Select a set of CPU applications and develop CUDA-based implementations. Compare the performance of the two.

5. *Energy efficiency:* Select a benchmark suite and evaluate power consumption on a conventional multicore system.

D. Optimization

1. *Graph processing:* Implement a graph processing algorithm on a Raspberry PI cluster. Optimize the code for performance.
2. *Knapsack:* Develop optimization strategies for the knapsack code and evaluate on a conventional multicore system.
3. *Matrix multiply:* Implement matrix multiply for a conventional multicore system. Develop optimization strategies.