Course Project - Graduate Operating Systems - Fall 2021

Overview

A key component of Graduate Operating Systems will be the creation and / or modification of a system artifact for the purposes of instrumentation and analysis. The project itself will involve an exploration of the prevailing literature, identification of key challenges, development or improvement of a system, and a fully analysis of said system proving successful achievement of your system design goals. Alternatively stated, your goal is to execute a successful systems pilot project that could plausibly turn into a fully-fledged systems research paper.

The course project constitutes 50% of your course grade.

Projects

For your project, you must select from one of the following topics as an initial starting point for the research:

- *Discrete event simulation:* Pick a particular focus topic from the course (storage, scheduling, virtual memory, virtualization) or something related to your research and create a discrete event simulator to conduct a "bake off" between various strategies.
    - Twists: Explore variations beyond the course involving discrete events, e.g. communications between drones, infectious disease modeling
- *Open-source software acceleration:* Take an existing software package that is used by researchers at Notre Dame and explore key bottlenecks or performance constraints in system. Explore how one might effectively distribute or compute across a reasonable set of computation nodes and measure / understand how such computation can scale.
    - Variation: Explore how one might transition a particular piece of software onto a GPU
- *Graph analysis or machine learning:* Work with an existing open-source graph analysis package or machine learning toolkit to explore performance properties
    - How fast could one execute WalkTrap or other algorithms on graphs of various sizes and complexities? What / where are the key performance impact factors?
    - Optimize graphlet extraction and identification across large-scale graphs. Capture either the system or algorithmic key metrics for performance.
    - Where are the particular bottlenecks in the system? I/O, computation?
    - How do implementations vary in terms of energy efficiency and performance across various languages (e.g. C / C++ vs. Python)?

- *High performance spectrum capture*: The RTL2832U is an intriguing chip for capturing small snippets of the electromagnetic spectrum. Explore the various bottlenecks with regards to performance of the capture chip which is a USB peripheral.
    - What is the maximum sample rate achievable with the RTL2832U? How consistent is the sampling rate at maximum speeds? What is the maximum speed for consistent performance?
    - What is the energy consumption of the USB peripheral? How long could a Rasperry Pi and accompanying battery last while capturing?
- *Docker:* Docker is an approach or creating a virtual machine in a box but yet without all of the components necessary for a virtual machine. Explore various performance aspects of Docker in a variety of scenarios (database, web server, simulation engine, etc.).
    - What is the overhead for running certain operations in Docker versus running those same operations natively?
    - Is the overhead consistent between various systems (ex. Macbook Pro vs. Windows desktop)?
- *Ultra-low energy proximity extraction:* Bluetooth LE and iBeacons offer interesting mechanisms by which proximity between users can be extracted. The RSSI or signal strength of the iBeacon can be quite informative when determining distance. Construct a prototype that uses multiple Raspberry Pis with battery backup and then records instances of proximity between two or more individual, each carrying iBeacon key fobs. Optimize your approach to minimize battery consumption.
    - How low can you drive the battery consumption? What is the overall lifetime with a reasonably sized battery?
    - How can you coordinate time between the Raspberry Pi units? Is it necessary?
- *Operating system fundamentals:* One of the more difficult concepts to teach is that of threaded / multi-programming due to the degree of difficulty with respect to debugging. Create an explore various environments that might assist in debugging or understanding multi-threaded programming.

While it is possible to propose an alternative project, the project must be well-conceived and discussed with Prof. Striegel prior to the concept stage. Customized projects outside of the above list are not recommended.

Team

Your project may be completed in a group of up to three students. While you may work alone, it is highly recommended that you work with at least one partner for the purposes of this project. In line with neither conferences nor journals making special accommodations for group size, there will be no expectation of quality or quantity differences between groups of differing sizes. Hence, it is likely in your own interest to work in as large of a group size as you can plausibly manage.

However, please note that groups are fixed by the start of fall break (Friday, October 15th @ 10 PM). It will also be in your interest to ensure that group members are adequately contributing to ensure the success of the group.

Deliverables

The deliverables for the course project are as follows:

Concept	5%	A brief three slides set conveying the focus of your project and group members. Presented orally (15 minute meeting) and submitted as a PDF.
Annotated Bibliography	5%	Identify ten references related to your work as well as a brief 2-3 sentence summary of the relevance of each work.
Proposal	10%	A three to four page overview of your proposed course project taking special care to identify the key components, resources, and timeline to successfully achieve your work. The proposal should also properly contextualize the importance of your project with regards to the overall body of literature in systems research. A key portion is a statement of your research hypothesis(es) and how you intend to test said hypotheses.
Interim Presentation	10%	You will be asked to make a group presentation for your project in a talk of 10 minutes or less providing a status update / presentation to the whole class. Your score will be determined by grades from the instructor as well as your peers.
Draft Paper	10%	A draft paper will be due the just before the week of Thanksgiving. The interim

		draft will be an approaching final version of your project.
Final Presentation	30%	You will be responsible for giving a final presentation during finals week for your project. The presentation may be up to 20 minutes long.
Final Paper	30%	Final version of paper

For each item, a detailed rubric for grading will be posted.

Due Dates

Concept	Friday, September 17th	10 PM
Bibliography	Friday, September 24th	10 PM
Proposal	Friday, October 8th	10 PM
Int. Presentation	Week of Nov 1st	
Draft Paper	Friday, November 19th	10 PM
Final Presentation	Week of December 13-16	
Final Paper	Monday, December 13th	10 PM

Submission of materials will be done via Sakai with all submission items allowing for unlimited resubmissions before the due date.