CS533 Programming Project Assignment
Spring 2018 - Karavanic

Project Goal: our goal is to extend the state of the art in regards to measuring and characterizing the performance of the operating system, in relation to application, hypervisor, hardware, and interrupts.

Project Groups: projects will be done in groups of 3-5 students.

Project Meetups: project groups should meet weekly

Key Dates:

May 9, 16, 23, 30	Weekly Reports due electronically
May 30	Version 1 Complete
Mon June 11	(final exam time slot) Project Presentations
Wed June 13 (11:59pm)	Individual Emails Due
Wed June 13 (11:59pm)	Project Reports due (Version 2 complete)

Project Deliverables:

- First meeting: outline the approach, a list of tasks each person is working on, a plan for group meetings, and a description of the platform and software that will be used. Outline what will be included in Version 1, and what will be included in Version 2.
- Weekly reports describe the progress, any issues/questions, and a plan for the remaining work.
- Final Report is a technical paper describing the work done, demonstrating the results, and a discussion section. The report follows the same format as a technical paper, including proper reference formatting. There is one final report per group.
- Individual email is a short email from each individual listing what group you are in and what particular part of the work you did.

Project Grading:

Each person will receive an individual grade.

Project Choices CS 533 Spring 2018

A: Virtualized Server Performance

A1 External Interference
This group will conduct an External interference study (measuring application performance effects caused by other VMs). This includes designing and setting up the study, including selecting the workload application(s) and installing them. You will deliberately drive up the resource consumption of a VM so that it interferes with the VM you are measuring. The goal is to allow an application running in a VM to be informed of external interference.

A2 Studying the overhead of system calls
This group will conduct a comparative performance study between Xen and a native Linux system, to measure and compare the overhead of system calls. This includes designing the study, including selecting the workloads and possibly installing needed software.

B. Power and the tickless kernel

Linux includes an option to run a "tickless" kernel, that does not perform the regular clock tick in order to allow the hardware to remain in a sleep state and save power. This group will study the performance and power of benchmark codes that run in tickless or regular power mode.

C. Power and data movement

In this class I have mentioned that data movement is a current focus of performance concerns, particularly power consumption. This group will write benchmark codes that target specific configurations expected to be power efficient or power inefficient. Then they will conduct a study using the benchmarks to actually measure the power consumption.

D. Scheduling Simulator

This project has two steps: first, find out how the default CPU scheduler in Linux works. Then, write a simulator to compare it to two other, simpler scheduling algorithms. You will need to generate simulator inputs to conduct a comparison study.