CSE 237A Winter 2018: Individual Project

Part 1 of 2: Environment setup and workload characterization
Assignment Overview

The goal of this project is to develop an energy-efficient scheduler, which handles multiple workloads controlling sensors running on Raspberry PI 3 (RPi3). During the course of the project, you will develop a program which characterizes workloads using performance counters (part 1), and an energy-efficient user-level LIST scheduler program which manages the sensors (part 2). For more detailed information see the instructions document in the project folder.

Individual Project Part 1 Assignment

During the first part of the project, you will setup the development environment for Raspberry Pi 3 (RPi3), and then measure performance of sample code running on RPi3.

Complete the following steps:

1. Cross-compile/install the provided Linux kernel and the sample kernel module
    - Linux kernel source code is available via git:
      `http://seelab.ucsd.edu/~shepherd/cse237a_wi18_kernel.git`
    - sample kernel code is in "sample_module" directory
2. Implement a kernel module to access the performance measurement unit (PMU) using the provided skeleton code
    - Skeleton code is in "pmuon" directory.
    - Monitor six PMU events.
3. Study how the provided workload characteristics changes over frequency settings. Modify the provided skeleton code to capture performance counter readings for the workloads (../workload_analysis.tar.gz).
4. Write a report that provides detailed performance analysis using the implemented kernel module (pmuon) and the user-space program (workload_analysis)
    - Analyze the program performance over different workload sizes and CPU frequencies, with plots that describe your analysis results

Submission:

1. Demo on Jan 23, 2017 (in CSE 3219) your Raspberry Pi 3 ~~that uses a single core~~ running correctly compiled sample kernel module, "hello.ko". Demonstrate successful execution of "check_kernel", provided on the class website.
2. Submit these three files into TED (Due: 23:59:59 PST, Jan 30, 2017)
    a. A kernel module source code for the event counter selection: `pmuon.c`
    b. A user-space program which measures the performance counters: `pmu_reader.c`
    c. A report on the performance analysis including the discussion of required plots (PDF)