Lab 1 : Software Energy Measurements

1 Measuring software energy with a powermeter

In this exercise, we will compare the CPU energy consumption of internet browsers using a powermeter and PowerJoular.

Software Requirements

We will be using a powermeter and compare three different internet browsers : Mozilla Firefox, Google Chrome and Gnome Web. You can also use other internet browsers in addition to these three.

Measuring the Idle Power Consumption

We will first measure the idle power consumption of the computer in order to have a measurement baseline. Restart your computer, then close all running user programs if any are open (email client, browser, etc.).

Use PowerSpyCli to capture power data from PowerSpy2 : `https://github.com/joular/powerspycli`, and start logging for 60 seconds.

Stop the logging (by holding `CTRL+c` in the terminal), and calculate the average power consumption from the generated CSV log file, and its standard deviation. We will consider this value as the idle power consumption.

Monitoring Power Consumption of Internet Browsers

Open one browser, and open the following video : `https://cdn.kde.org/promo/Announcements/Plasma/5.20/Video.webm`. Don't start the video (rewind to the beginning if it autostarted). Start logging power using the powermeter and name the csv file with the browser's name (such as *firefox-1.csv*), then run the video in the browser. When the video ends, stop the logging (`CTRL+c` in the terminal).

Close the browser, and repeat the same experiment at least 2 more times (for a minimum total of three separate files the browser).

Repeat the same experiment for each of the remaining browsers.

Analyzing Power Data

Collect the different CSV log files, and extract the power consumption evolution for each browser. For each line of the collected data, subtract from it the idle power consumption. Finally, compare the power consumption and total energy for all browsers (produce charts of power evolution, energy comparison, and summary analysis).

Compare all three browsers. Which one is the most energy-efficient in this experiment ?

2 Generating power models for computing and electric devices

In this exercise, we will generate power models for various computing and electric devices, using a regression-based approach.

2.1 Raspberry Pi 4

We will generate a power model for a Raspberry Pi 4 device. To do so, we will use the power benchmark suite called CPU Power Benchmark which can be downloaded from : `https://github.com/joular/cpupowerbench`.

Configure and run the benchmark on the Raspberry Pi device, while collecting its power consumption using the PowerSpy2 powermeter. Use PowerSpyCli to capture power data from PowerSpy2 : `https://github.com/joular/powerspycli`.

From the collected data, generate a power model using regression algorithms (linear, polynomial, etc.). Calculate the error rate of the model on the collected data.

Repeat the benchmark and calculations at least 3 times to calculate an average, and compare the results. If the 3 results are within an acceptable margin, then merge the data from all experiments to build a big data set and use it to generate a new power model, and calculate its error rate.

2.2 Electric and electronic devices

Now we aim to generate a power model for non-computing devices, such as smart objects or electronic devices.

Following the same approach as with the Raspberry Pi, we will generate a power model for a computer monitor.

We will need to create our own benchmark, or to benchmark the device manually. In particular, identify a characteristic to model (such as the screen brightness), then manually (if automating it is not possible) change its value while measuring its energy consumption for at least 60 seconds.

With the data collected, and the experiment repeated 3 times, generate a power model for the computer monitor and calculate its error rate.