CS 500, Spring 2021
Systems Evaluation Lab
QuickShift

The goal of this project is to allow you appreciate the complexities and challenges of measuring power dissipation, energy consumption, and heat dissipation of programs running with different software/hardware system configurations. It also will help you to understand the possible power/energy/thermal tradeoffs of different hardware platforms and execution modes, i.e., sequential and parallel.

Through experiments, you will explore different forms of parallelism - multi-threaded parallelism using OpenMP and "vector" parallelism on GPUs using NVIDIA's CUDA language, and their tradeoffs in terms of:

Execution Time vs. Energy vs. Power vs. Heat

The experiments will be based on a single image processing application, namely the image smoothing code (QuickShift). The project consists of an evaluation of QuickShift running on a single CPU, multiple cores, and the GPU of NVIDIA's single board, heterogeneous Jetson TX-1 system. The deliverable of the project is a written report that describes your experiments, and presents and discusses your measurement results. A deep understanding of the quickshift application code is not required. Given the input image, the application reports the execution time in milliseconds. A working knowledge of Linux, C++ and OpenMP is desirable, latter if you want to change the number of cores used. Important metrics to report are:

Execution Time: reported by the application (milliseconds)
Power: voltage * current (reported by the powerstrip)
Energy: power * execution time; you may assume that the power is constant; in general, it is the integral under the power curve.
Heat: reported in degrees centigrate

1. Jetson TX-1 board

A detailed description of the board that you will be using can be found online.
NVIDIA's TX-1 board: http://elinux.org/Jetson_TX1

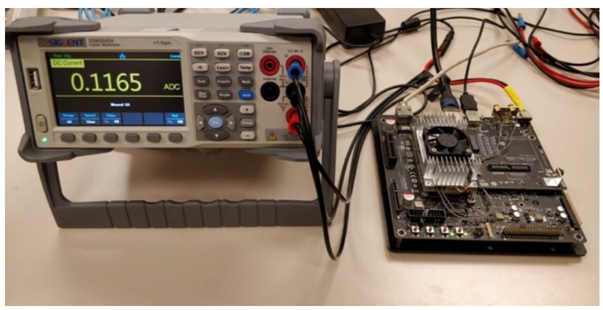
Please feel free to just perform a google search to find more information about this system. All of you have been assigned a TX-1 board for code development and testing purposes, so please use only your assigned machine. You can remotely log into these systems from an iLab machine and run your application code. You need to be on a machine at Rutgers, so directly logging into the TX-1 boards from outside is not possible. **You are sharing the development machine with other students, so please be**

considerate. You should have received an email from me regarding your TX-1 development board and its password. There is only a single user on these boards, called “csuser”. Do not look into other student’s directories.

Each of the three measurement stations in Hill 254 will have their own TX-1 board. Instructions of how to log into these boards and access the multimeter are posted at each measurement station. These boards are called jetson-1.cs.rutgers.edu, jetson-2.cs.rutgers.edu, and jetson-3.cs.rutgers.edu.

2. Siglent SDM3045X Digital Multimeter (voltage and current)

Hill 254 (the CAVE) will have three measurement stations equipped with a TX-1 board and a multimeter. In addition, there is a networked Raspberry PI controlling RTD heat sensors on the processor heat sinks for “external” heat measurements.



The multimeter is set up to either take voltage or current measurements. The multimeter is not able to measure both metrics at the same time. You are able to start your measurements, and the multimeter will record measurements for 30 seconds time intervals. The measurement resolution is 40 milliseconds (25 measurements per second), with a total of 750 measurements. After 30 seconds, you can download the measurements. You have to follow a specific procedure, i.e., sequence of commands to do so.

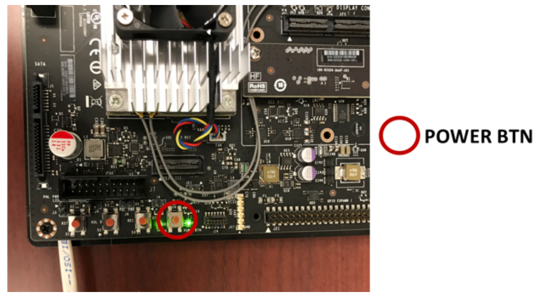
The initial procedure is listed below and is subject to change. You must use ilab machines ilabu1.cs.rutgers.edu, ilabu2.cs.rutgers.edu, iabu3.cs.rutgers.edu, or ilabu4.cs.rutgers.edu. “mmX.cs.rutgers.edu” will be the network names of your particular station multimeter. The Jetson TX-1 boards and their multimeters can be accessed using “jetson-X.cs.rutgers.edu” and “mmX.cs.rutgers.edu”, with “X” either 1, 2, or 3, reflecting the particular lab station you are using. Please watch the 11 minutes video “LabStationTutorial” available on sakai/Resources/videos, or here:
https://www.cs.rutgers.edu/courses/500/classes/spring_2021_kremer/videos/LabStationTutorial.mp4

You also will need file “setup.scpi” to initialize your multimeter, which must have the content:

```
CONF:CURR:DC 6A
SENS:CURR:DC:NPLC 1
TRIG:BUS
TRIG:COUN 750
TRIG:DEL +2.33333333E-03
```

A copy of this file is available on our course website. Setup.scpi is used in step 10 as described below.

Please follow the instructions below carefully. If for some reason the equipment does not work properly, you may reboot the system, including the multimeter and board (green = ilabu, red = TX-1):



1. From your laptop, ssh into ilabuX.cs.rutgers.edu, where X can be between 1 and 4. These numbers are not related to your lab station, so any of the four machines will do. Make sure that file setup.scpi is available in your project directory.
2. Turn multimeter on, wait for it to boot (around 10 seconds)
3. Put multimeter in DC current mode: press <Shift> then <DCV>
4. Set current range to 6A: <Range> then <+> <+> <+> (push <+> button 3x)
5. Turn on Jetson TX-1 board (tap "POWER BTN" twice); green lights will come on
6. Wait for 20 seconds until Jetson TX board is up.
7. From ilabuX.cs.rutgers.edu, ssh into jetson-?.cs.rutgers.edu, i.e., the specific TX board on your station. The password will be posted at the station. '?' is either 1, 2, or 3, depending on your particular station.
8. Select DC voltage mode: push <DCV>
9. Record operating voltage (e.g.: 19.275 – needed for power/energy computation)
10. Load data acquisition setup

```
lxi scpi -a mm?.cs.rutgers.edu -s setup.scpi 'ABOR'
```

This will change the configuration of your multimeter to record current (Amps) readings
11. Start acquisition when ready

```
lxi scpi -a mm?.cs.rutgers.edu 'INIT'
```

This erases previous values and start measurements.
12. Run the executable program that you want to measure (e.g. quickshift) on your TX board
13. Wait 30 seconds before you can retrieve data. Multimeter will stop blinking (top left corner of screen) when all 750 measurements have been acquired.
14. Retrieve stored data

```
lxi scpi -a mm?.cs.rugers.edu 'FETC?'
```

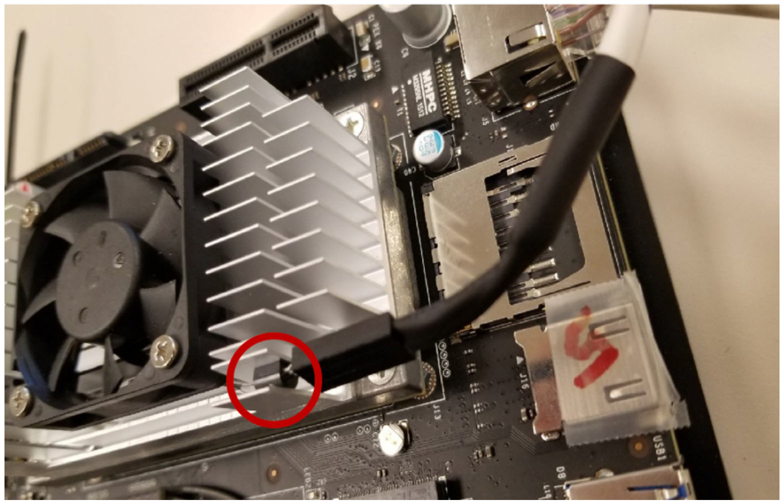
capture by redirection; numbers separated by commas
15. Repeat steps **11-14** as many times as desired
16. Soft shutdown Jetson TX-1 board.

```
sudo shutdown -P now
```

If a soft shutdown is not possible, you may use a hard shutdown as your last resort by holding down the power button "POWER BTN" for 5 seconds, or until the green lights go off.
17. Turn off multimeter

3. Raspberry PI with “external” thermal sensors, and TX-1 “internal” sensors

Temperature measurements will not use the multimeter, but have their own measurement infrastructure consisting of a Raspberry Pi and RTD (Resistance Temperature Detector) sensors mounted on the heat sink of the board as shown below, and a set of internal thermal sensors of the TX-1 architecture.



The head sink temperature can be access using the following commands from your ilabu machine.

Station1:
curl http://temp-pi.cs.rutgers.edu/json/28.7A11E00C0000/temperature/

Station 2:
curl http://temp-pi.cs.rutgers.edu/json/28.A099DF0C0000/temperature/

Station 3:
curl http://temp-pi.cs.rutgers.edu/json/28.3368DF0C0000/temperature/

Temperature readings update no more than once per second. The temperature will be expressed in degrees centigrade. The sensors have a resolution of 0.0625°C. The accuracy is ±0.5°C from -10°C to +85°C and ±2°C accuracy from -55°C to +125°C. You may write your own script (e.g.: Python script) to pull temperature readings every second.

In addition, the TX-1 board has 8 thermal zones each with an associated sensor. For example, to get the “domain” of thermal_zone1, execute the following commands on the board:

```
cat /sys/devices/virtual/thermal/thermal_zone1/type
```

To get the actual readings, use command

```
cat /sys/devices/virtual/thermal/thermal_zone1/temp
```

Readings are in degrees centigrade (Celsius).

For this project, thermal_zone1 (CPU-therm) and thermal_zone2 (GPU-therm) will be most relevant. Part of the project will require you to measure temperature behavior over time when a program is running, showing how the "internal" temperatures relate to the temperatures measured at the heat sink. The measurement resolution should be less than 2 seconds. You may use any program(s) for your measurements, including versions of quickshift or your stresser code. Make sure that you have at least a measurement time window of 30 seconds to show the correlations.

4. QuickShift application

There are three versions of this code, one sequential (cpu), one using multi-threading with OpenMP (omp), and one using the 256 core GPU (gpu). The application has two parameters, tau and sigma, that can influence the quality of the program outcome. The application takes a single image as input, for example "flowers2.pnm" shown on the left:

The application computes pixels that have similar color values. Significantly different color values indicate object boundaries. The image within such boundaries is then smoothened, i.e., made identical. The notion of "identical" and the size of the regions that can be smoothened can be adjusted, resulting in different output image qualities. The code is written in C++ / CUDA. The outcome for tau=6 and sigma=10 (default values) is the right picture shown above.

You may experiment with different values of tau and sigma, but this is not really required.

5. How to get started and run the experiments

Log into your assigned development TX-1 system. Copy cs500-project.tar into your personal directory named with your Net ID. All students have the same user name "csuser". To log into your assigned TX-1 board, execute "ssh -X csuser@tx?.cs.rutgers.edu", where ? is your assigned system number. Untar the project file "tar -xf cs500-project.tar". Go to the quickshift directory and say "make", and the executable "quickshift" is generated. To clean up, say "make clean". There is a single picture, flowers2.pnm, in this directory. Additional pictures can be found in the pictures subdirectory. You can change the number of cores used in OpenMP through routine "omp_set_num_threads" in file "quickshift_cpu.cpp". The default setting is 4 threads. If you want to change this, you will need to remake the executable.

You can call quickshift on a file using different computing resources as follows:

```
./quickshift  --file  <input.pnm> --mode cpu
./quickshift  --file  <input.pnm> --mode omp
./quickshift  --file  <input.pnm> --mode gpu
```

If you just say ./quickshift, it gives you the possible command line options.

Values for sigma and tau can be provided using "—sigma" and "—tau" command line options. The default is 6 and 10, respectively. After you executed quickshift, the resulting image can be found in
<input>-gpu.pnm, <input>-cpu.pnm, and <input>-omp.pnm

The basic image viewer on Ubuntu is called "eog". You can visually inspect the output to see whether they are all identical. You may also be able to use "display".

Once you are ready to run your codes on the measurement platforms in the Hill254 lab, sign up for lab times using the posted google doc on sakai/Announcements. Do not share this link with anyone.

6. Signing-Up for Lab Slots

Lab times are crucial resources for everyone. Slot durations are subject to change, and are currently 2 hours long. You MUST sign-up for a slot in advance through our shared google doc sign-up sheet. Please see the appropriate post on sakai/Announcements. Here are some important rules:

- COVID-19: You must have submitted your COVID test at least 7 days before you can come to the lab.
- You need to be current with your weekly testing while coming to the lab.
- Use the My Campus Pass symptom checker self-screening app when entering the Hill building.
- You can have only at most two active entries in the sign-up sheet at any point in time. Active means that it is a future slot.
- You may **cancel a slot up to 2 hours before the slot starts**. You will fail the class if you have more than 1 "no-show".
- You may **sign up for a slot until 1 ½ hours in advance**. You cannot just "walk in".
- Be fair and considerate. Allow everyone to get their share of access to the lab space.

7. Safety Protocol

While going to the lab, you have to be weekly tested for COVID. Your initial visit to the lab can only be 7 days after you have had your COVID test. You have to wear masks and gloves while being in the lab. Gloves and "back-up" masks are provided. Gloves are needed since you are touching the multimeter buttons and the TX-1 booting switch, and these surfaces are hard to clean. Please bring your own laptop. Station 3 has an associated workstation which would not require you to bring a laptop. Bringing a laptop is just a safer option.

At any given time, no more than 3 students will be in the room, and sometimes the ilab assistant. Please stay socially distant. The ilab assistant will come to the lab on a regular basis to check whether everything is fine. Please follow the ilab assistant's instructions.

Ten minutes before the end of the session, you should shut down the TX-1 board and the multimeter. Then please use the provided wipes to clean the desk surface and armrests of your chair. Let's keep everyone safe!

8. Grading

Grading is based on a written report (three to five pages). No code needs to be submitted.
You may "play" with different tau and sigma values for QuickShift, or use different sets of images that you may get from the internet. However, keep in mind that you have a window of 30 seconds for your measurements. You should run quickshift on different numbers of OpenMP cores (2, 3, or 4, which is the default). Show what you did in a form that makes it accessible to a reader not familiar with the issues. Go for quality rather than quantity. Graphs are good tools to convey a trend and/or tradeoffs. Is there a lesson to be learned here?

9. Project Questions

All project related questions should be posted on piazza. Thanks!