This Python file is to run in the background to log the CPU usage (%) during my activities on my laptop every 10 seconds.

This helps me practice performance load testing.

I used psutil library to get the CPU information and used pandas library to convert it to Excel data frame, then open the Excel file using openpyxl and log information. Also, threading library was used to run the program in the background. A one-million-loop calculation was done to calculate a relative CPU speed in comparison with the data from psutil.

Run this by cloning this repo, or following this bash:

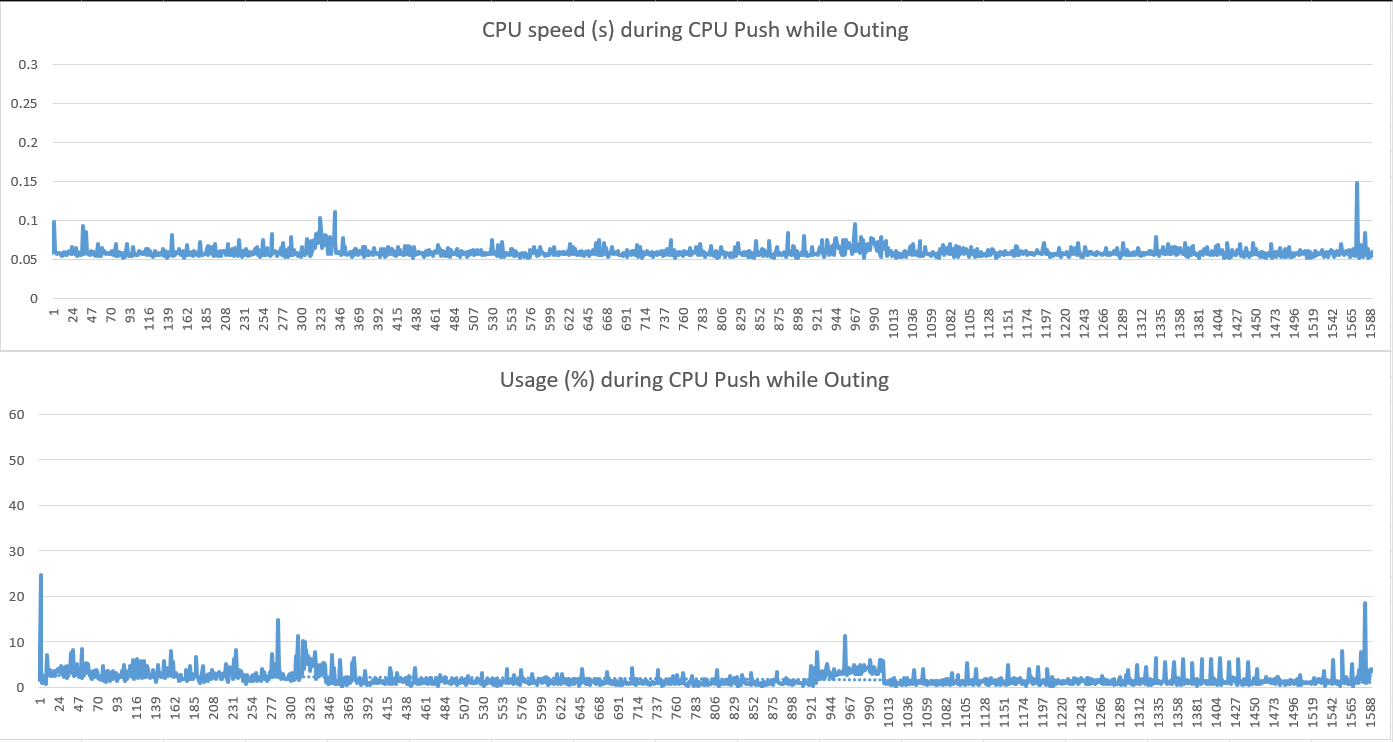
Git clone

Cd “<LOCATION\_OF\_THE\_FILE”

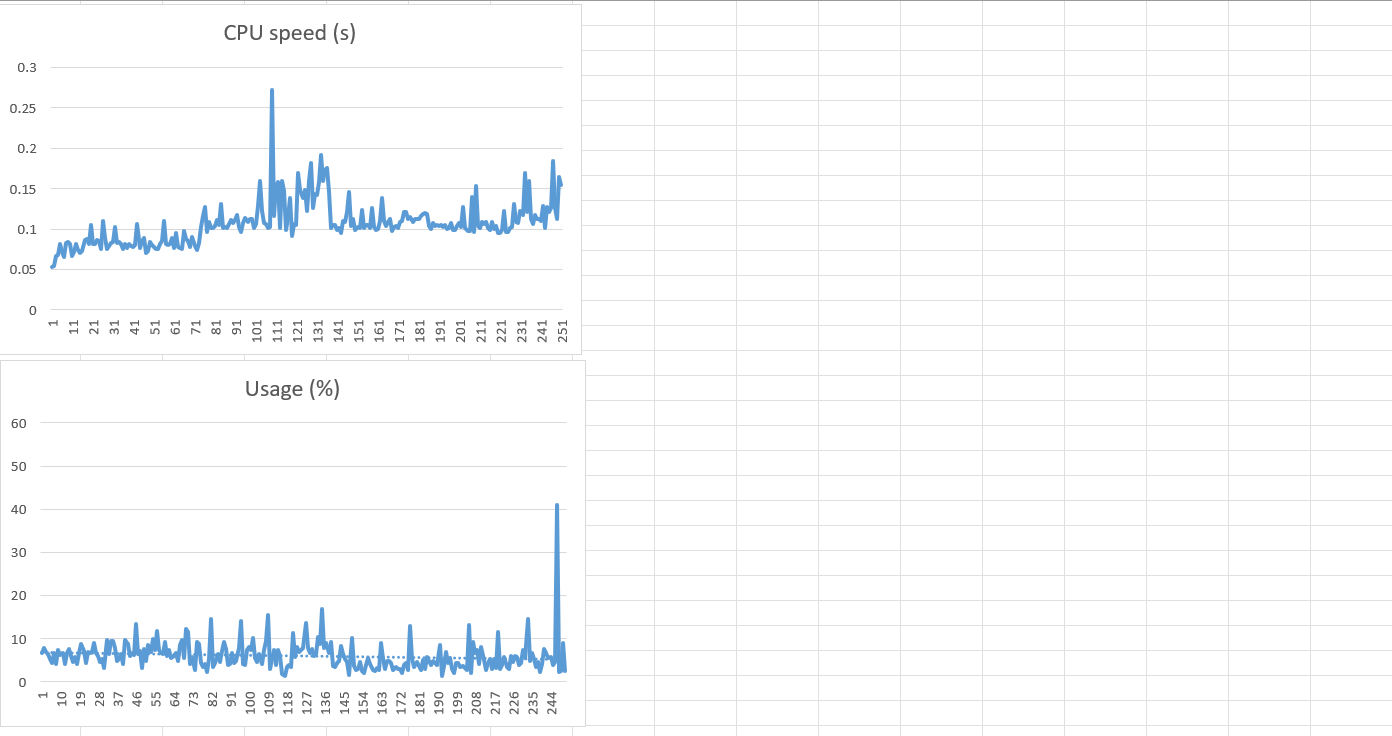
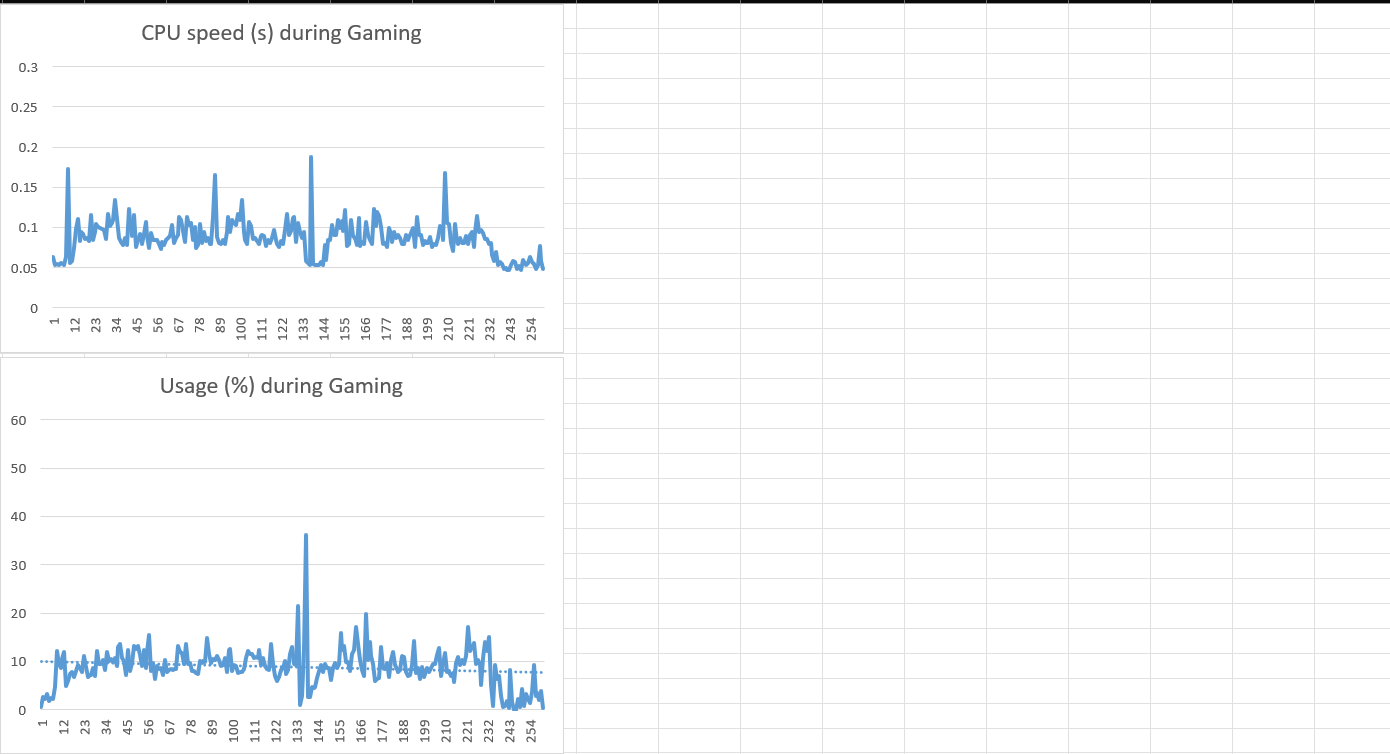
Python CPU\_speed\_during\_ROS2\_dev\_test.py

**By doing this, I have gathered the following data.**

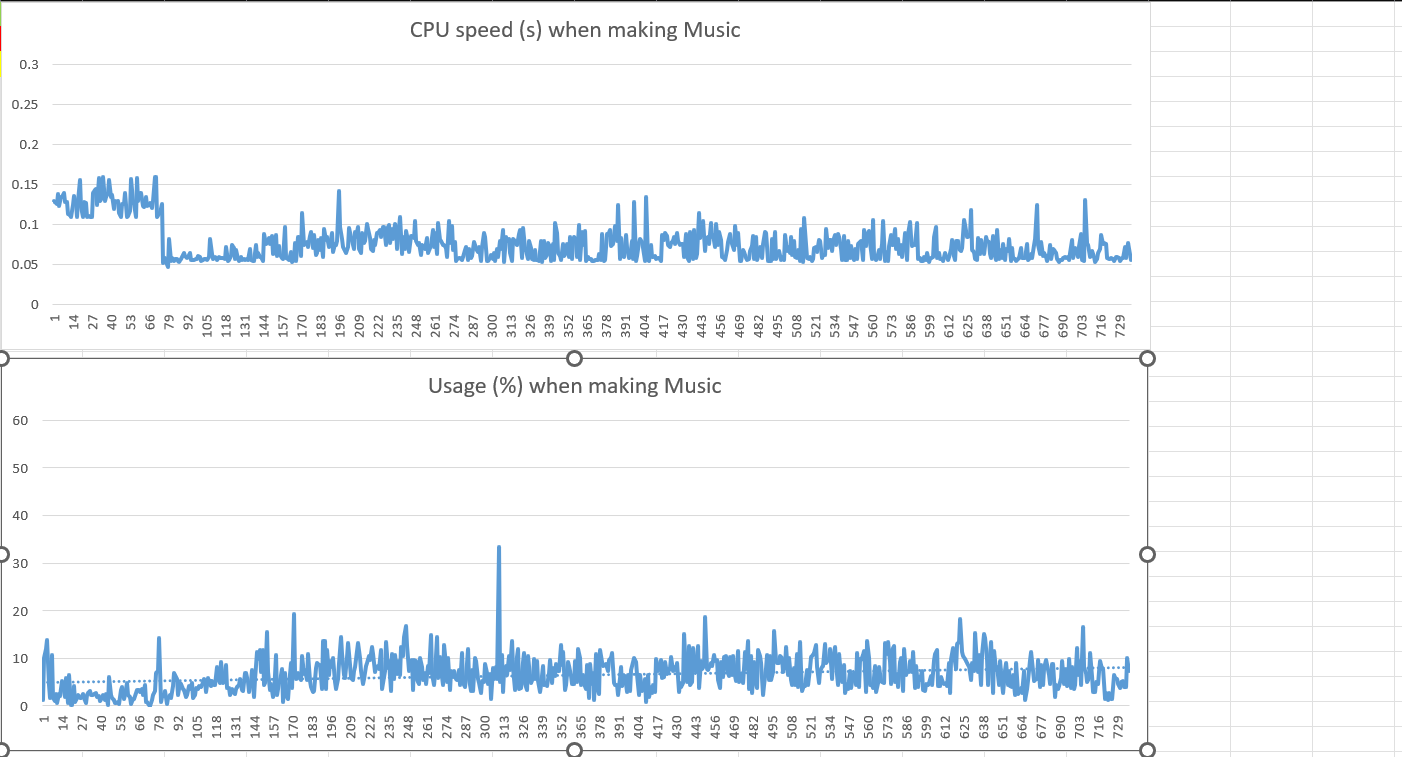
The information below let me know that for a project as complex as ROS2, using Docker is very favourable, as it saves half of the time, and uses much less CPU. Also, LOL gaming and music making would be my activities that take up the most CPU.

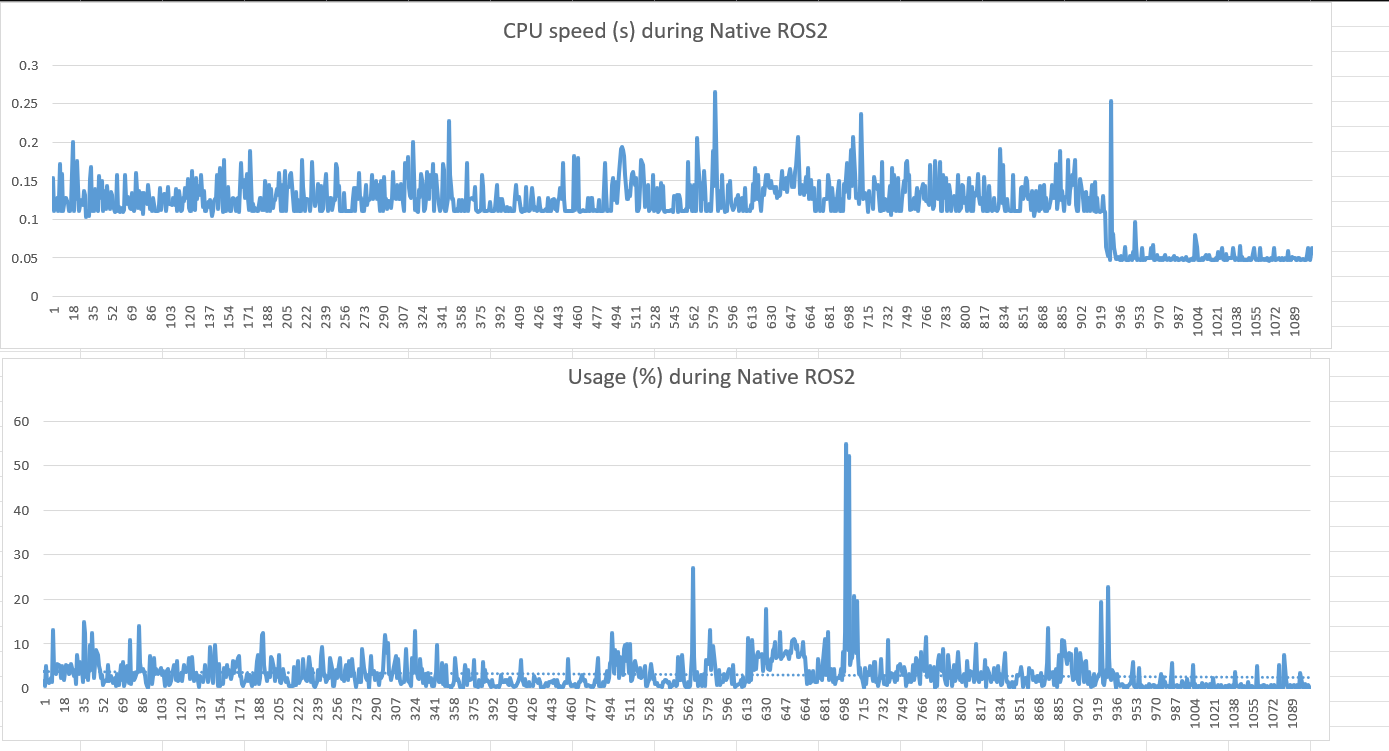


When I left home to hang out for 5 - 6 hours and left the laptop running with 20 active web browsers running, an average of 1.85% CPU was running with the peak of only 24% when I first turned on the 20 browser tabs.

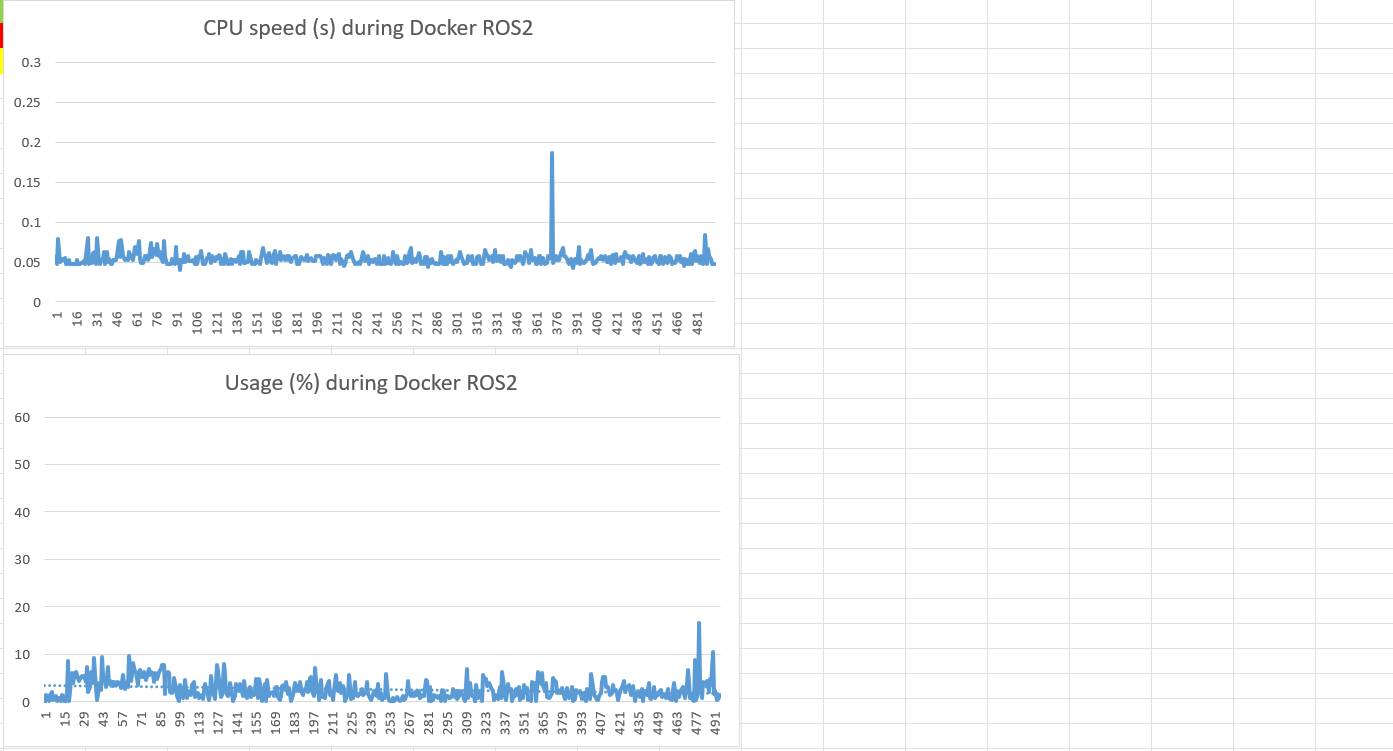


For each 1 hour of League of Legends gaming, CPU average 6-10%, with the peak of up to 36 - 45% once a match finishes. This probably relates to how the game has to load post-match data.

During my 2 hours of making music, Davinci Resolve uses 6.59% CPU on average, with a peak of 33%, occurring only once.



During my 3 hours of setting up dependencies and coding ROS2 natively, CPU used average 3.14%, but has exceeded 20% multiple time, with the peak of up to 55% CPU



With the same amount of tasks, coding ROS2 using Docker takes much less time to set up, average only 2,62% CPU with a peak of only 16.6%. This means it is the lightest activity.