SUTD 2021 50.003 Problem Set 4

James Raphael Tiovalen

Cohort Exercise 1

Most parts of our project are sequential since we need user data to perform the backend computation. However, since one of our project's implicit non-functional requirements is to return the result in a short amount of time, we need all the performance gain that we can obtain to improve the speed of the algorithmic localization computation.

1. One part that can be parallelized would be the localization algorithm itself. Once an algorithm has been chosen by the user, as the mapping process is being done, the dataset can already be passed on to the backend to serve as the training dataset for the machine learning model (in an incremental method). This allows faster turnover rate since the user does not need to wait for a longer amount of time once they have done mapping.
2. Another part that can be parallelized would be the Wi-Fi AP BSSID-RSSI data collection. Instead of collecting the RSSI values for each Wi-Fi AP BSSID in a for loop, we can implement a multi-threaded approach, whereby each thread will take care of collecting the RSSI values of each Wi-Fi AP BSSID. This is because the hardcoded Wi-Fi AP BSSID values can be quite numerous. By spawning the same number oof threads as the number of Wi-Fi AP BSSIDs, we can shorten the amount of time required for the mapping process.
3. The last part that can be parallelized would be the interaction between the user-facing Android UI and the back-side computation. The user interaction can remain to stay interactive while the backend is processing and training the collected dataset. This is also a good way to incentivize the user to keep the app running and not let the app sleep in the background or kill the app's process.

Cohort Exercise 2-4

Solutions to these Cohort Exercises are included with this document in these respective code files: `FactorThreadNoInterrupt.java`, `FactorThread.java` and `DiningPhilDemo.java`.