HPC Lab Week 3

AI/Machine Learning services available on Cloud uses HPC for accommodating frequent calls from millions of users. Here are some questions for applications of HPC in Machine Learning. Scenario here is you are providing cloud HPC (similar to Azure/AWS/GCP) for training and testing of machine learning models on large datasets. In this lab we are going to implement task parallelism for training different machine learning models on the large-scale dataset "BitcoinHeistRansomwareAddressDataset Data Set" containing 3M samples of Bitcoin ransomware attack data available at

https://archive.ics.uci.edu/ml/datasets/BitcoinHeistRansomwareAddressDataset

Training and testing ML/DL models on large scale datasets serially will be taking a lot of time. Hence we go parallel.

Preferred platform: Google Colab

1. **Dataset processing:** Download the dataset to your colab or local jupyter platform. Apply preprocessing and one hot encoding wherever required. Divide the data into 50% for training and 50% for testing. The objective is to label each sample with the name of the ransomware family (e.g., Cryptxxx, cryptolocker etc) or white (i.e., not known to be ransomware).
2. **AI/Machine Learning/Deep Learning model training:** Training at least 2 (as colab has 2 cores but you can train more models) machine learning/deep learning models (such as Decision tree, random forest, KNN, Neural Network) using concurrently running multiple threads/processes using same training dataset having 1.5 Million samples. If you want to train neural networks, please implement using Sklearn which will train on CPU (as implementing in keras will be training the model on GPU in colab and CPU in your PC). Save the trained model to the disk as pkl file. You can experiment with more models and perform concurrent training on 1.5M samples if you have more free cores in your PC.
3. **AI/Machine Learning/Deep Learning model testing:** Load at least 2 trained models in concurrently running threads/processes and run the model on testing dataset with 1.5 Million samples. After getting results from the different models, perform average voting or max voting to combine results of the ML models. Note: for max voting we need 3 models, so you can do average voting. The maximum voting or average voting of prediction from different classifiers can be performed only when all the threads finish their execution. The predictions can be further compared with original labels to compute the accuracy.