Kings Final Project Proposal

Topic: Mine Venmo Food Transactions with Data Parallelization for the use of Advertisement.

**Introduction**

One of the biggest problems we are having today is the processing of large amounts of data. With 2.5 quintillion bytes of data being created everyday, the way we utilize and manage this data can become largely difficult and sometimes inefficient. Parallel processing can solve this problem by breaking down problems into smaller bits and solved simultaneously in a much faster and efficient way. Parallelism also ensures the effective utilization of hardware resources by allocating data among processing units. Our goal with the project is to achieve parallelism while data mining a particular set of large data for a specific intended audience.

1. **A description of the problem you propose to solve.**

In today’s day many people spend time browsing online through their devices or social media applications. In previous years, most of advertisement has been done through televised ads or paper media ads. However, due to society’s wide usage of technology, advertisements have now begun to take the platform of our devices to target their users.

Venmo is a digital wallet that allows you to make and share payments with friends. A feature on Venmo is that if you have your payments set to public (default setting) you are able to see what payments your contacts and other users are making. It is important to note that the monetary amount is not shown but instead the reason/message for the payment. Since the payments are public, Venmo’s public API (no authentication needed) returns the data for all public payments.

What we would like to solve or in turn make better, is mine the data from these transactions for users who have paid someone for a certain food. Note, that many millennials use emoticons to specify what food category was purchased. With this data, we would have the username of users who have recently made a purchase for food items. This data can now be provided to advertisement agencies who are looking to target users with specific food/beverage brands.

1. **The independent subtasks you want to parallelize.**

In order to achieve targeting specific users with ads we will need to find users that have made food specific transactions. Due to the popular nature of Venmo, millions of public transactions have been made for different payment reasons. Therefore, our dataset has millions of transactions with user data. We will need to parallelize the iteration and food filter for each transaction. This way our end product will only contain the username and food item for a user who payed someone for food. Since our data set is such a large amount, it is appropriate to use a GPU to be able to achieve our goal.

1. **Any applicable software packages you will use (with websites).**

* Venmo Data Transaction - <https://github.com/sa7mon/venmo-data>
  + The link above has already scraped the data from the public API. The data was collected between July/October 2018 and January/February 2019
  + 7,000,000 Data Transactions available
* JavaScript Script - <https://github.com/paolaguitian/Venmo-Data-Minning-CDA4101/blob/master/datahelper.js>
  + Due to the nature of web APIs, the data returned is in a JSON format. A script was needed to convert the JSON into code readable by C/C++ platforms.
  + The code returned from the script has already been cleaned to only return what is needed for this project. Username and Message Description
    - script written by the Kings group members.
* CUDA - <https://developer.nvidia.com/cuda-zone>
  + Software which allows parallel computing, specifically GPU

1. **Some thought as to which technique(s) you will use (SIMD, multicore**

**or GPU).**

We will be using the graphical processing unit (GPU). With the use of CUDA we are to write a program that will distribute the work amongst the cores. The work entails, properly distributing the data to each core, then filtering out the transaction that do not involve food, followed by returning the data needed from the GPU to the CPU.