VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computing Science and Engineering

LAB - 6 Exercises

Course Code	**:**	**CSE3025 – Large Scale Data Processing**	**Date**	**:**	**04/09/2019**
Lab Experiment	**:**	**MapReduce Programming Exercise Currency Exchange rates**	**Slots**	**:**	**L15+L16**
Instructors	**:**	**Dr. Bharadwaja Kumar and Prof. Ramesh Ragala**			

Objective:

1. To understand the detailed processing of MapReduce Framework

Problem- 1:

Consider currencies being traded on different stock markets, and you're looking for arbitrage opportunities to make money. We have hourly data for all currency pairs for all currency markets across the world, which could account to many terabytes of data. That's probably a bit too much for Excel to handle.

The fields in dataset as follows

CurrencyMarketSymbol = Currency Market

Date = Date of Order

LocalTime = Local Time the Order was Placed

IsBuy = Is the order a buy or sell order

Rate = What is the change rate being asked for the order

Amount = The total amount of currency to be exchanged

IsCompleted = Has the order been completed?

Example columns: (CurrencyMarketSymbol, Date, LocalTime, Instrument, IsBuy, Rate, Amount, IsCompleted)

We would attack this problem in several map and reduce steps. The first would be to run map over every Date and LocalTime, to see if we can find an open buy order (IsBuy=True) for the same Instrument where IsCompleted = False. We can then assign a UUID to each record we have so we can match it to another record.

UUIDs have an astounding uniqueness rate, where generating 1 billion UUIDs every second for 100 years, gives us a probability of 50% of creating just one duplicate. (CurrencyMarketSymbol, Date, LocalTime, Instrument, IsBuy=True, Rate, Amount, IsCompleted=False, UUID)

Let's say we wanted to look at simple arbitrage opportunity between CAD, USD, and EUR currencies.

Assume these fake currency exchange rates of the EUR/CAD, USD/EUR, CAD/USD pairs are 0.664 (1 CAD gets you .0.664 EUR), 1.234, and 1.398, respectively.

Because we're looking for IsBuy=True all of the actors we're interacting with want to buy currency from us .We buy EUR$6,640 for CAD$10,000. We can then buy USD$8,193.76 with our EUR$6,640 for (6,640*1.234). Finally, we can buy CAD$11,454.87 from our USD$8193.76, making us a profit of CAD$1,453.87.

This is what you call triangular arbitrage, where the if the product of all the exchange rates (0.664*1.234*1.398 = 1.14) is greater than 1, there is room to make money. We won't go into the more technical applications of finance here.

Now that we know what we're looking for we can start use MapReduce into three reduced sections, and then compute the opportunity for the different Rate returned. And because we have the UUIDs on every open transaction, we can start to compare all potential transactions to see we can make a profit.

Section 1 - Find EUR to Buy with CAD

(CurrencyMarketSymbol, Date, LocalTime, Instrument=EUR/CAD, IsBuy=TRUE, Rate, Amount, IsCompleted=False, UUID)# Section 2 - Find USD to BUY with EUR

(CurrencyMarketSymbol, Date, LocalTime, Instrument=USD/EUR, IsBuy=TRUE, Rate, Amount, IsCompleted=False, UUID)# Section 2 - Find CAD to BUY with USD

(CurrencyMarketSymbol, Date, LocalTime, Instrument=CAD/USD, IsBuy=TRUE, Rate, Amount, IsCompleted=False, UUID)# Profit formula sudo code

```
HashMap<UUID, Double> r1 = new HashMap<UUID, Rate_EUR/CAD>();

HashMap<UUID, Double> r2 = new HashMap<UUID, Rate_USD/EUR>();

HashMap<UUID, Double> r3 = new HashMap<UUID, Rate_CAD/USD>();for value_1 in r1:

    for value_2 in r2:

        for value_3 in r3:

          if value_1*value_2*value_3> 1:

              return key_1, key_2, key_3
```

You are not suppose creating an arbitrage program, but you have to implement a MapReduce application to see how currency FX change day to day as a percentage. Use python and Java language to solve this problem.

Dataset name : dataset.csv