Description of Assignment B

Yixiang Wang

The assignment can be divided into two parts:

1) Generate synthetic transaction record data
2) Calculate the VWAP of synthetic transaction records.

For the first part, I choose to use python with the help of yahoo_finance library to get the current price of each stock. For the second part, I input the data set generated to KDB, calculate and output the VWAP with a small q script.

Please also read readme.txt in the repository.

PART ONE: Generate Synthetic Data

Generating synthetic data requires to make some assumptions of the price series. Since the aim of this assignment is to practice on using in memory database, KDB, with q language, simple assumptions should be preferred (fitting an ARIMA model to the price series of each stock is certainly too much). Yet, I don't want the synthetic date set looks too unrealistic. So the biggest challenge here is too make a tradeoff between efficiency and accuracy.

I made following assumptions:

1) Only the stocks in the Dow Jones 30 index are traded.
2) The initial price of each stock is set to be the price in the day of March, 7th, 2016. This initial price is achieved by a python library called yahoo_finance.
3) The price of each stock will either go up by 2.5%, stay the same, or go down by 2.5%
4) Size of transactions are uniformly distributed from 1 to 1000

Some justifications about the assumptions:

1) The symbols of the stocks are meaningful in real market, so I don't want to use randomly generated symbols (like AAAA), which do not make so much sense. Yet I don't want to incorporate too many symbols to run the risk of reducing efficiency.
2) I choose the price of a trading day to set reasonable initial prices of stocks.
3) The percentage of price movement is simulated as random walk. So the price at time t is calculated conditional on time t-1, which is a more reasonable setting than to assume all the transaction prices are identical independent distributed. Also in this way, we avoid negative prices.

The implementation of this method is quite straightforward. The tickers are put into a hash table mapping from integer to symbol. Each time a random integer from 1 to 30 is generated and the corresponding ticker is chosen. The price movement is simulated in a similar way.

The generated data is saved in data.csv

PART ONE: Calculate the VWAP

It is quiet convenient to calculate the VWAP with KDB and q. I did it with the following steps:

1) Read the csv file to a KDB dictionary.
2) Flip the dictionary to create an unkeyed table.
3) Calculate the date of each transactions and add them as a new column to the table with update function.
4) Group the records by the date and ticker and calculate the VWAP.

The formula used to calculate the VWAP is simply:

$$\frac{\sum NUMERBER\ OF\ SHARES\ BOUGHT\ \times SHARE\ PRICE}{TOTAL\ SHARES\ BOUGHT}$$

The result is output to vwap.csv.

Possible improvement:

The simulated transactions happen on a 24/7 basis, which is very unrealistic. Some clever methods may be developed to take the trading days and hours into consideration in an efficient manner.