BDiF

Big Data in Finance

Baruch MFE “Big Data in Finance” course ~ Assignment B

Document name: “BDiF2016 - Baruch MFE Big Data in Finance Course - Assignment B.pdf”
Date/revision: Sat 5th March 2016 / Rev E.
Author: Andrew Sheppard, © 2016.

Title: "In-memory time-series data analysis with KDB and Q[1]"
When issued: Saturday 5th March 2016
When due: 11:59pm, Saturday 12th March 2016
Contact details: andrew.sheppard@baruch.cuny.edu / 203-779-9343

1. Introduction

Time series analysis is a key skill in financial Big Data. This assignment will help develop those stills in the context of Big Data sets and in-memory database analytics.

2. Problem Description

First, using a synthetic time-series data set you create, you will calculate a VWAP (Volume Weighted Average Price) for trades in a number of symbols (assets, say, equities) over a 10am to 4pm trading day. This work will be done on your laptop or desktop using the free 32-bit version of KDB (https://kx.com/software-download.php for which the maximum in-memory data size is 2GB). You can create this data set externally using any tool of your choice and write the Q code necessary to parse and load it into KDB, or you can generate the synthetic data set within KDB directly by using Q to generate the data.

Second, using a sample (no larger than 1GB) time-series data set of real financial data, you will calculate a VWAP (Volume Weighted Average Price) for trades in a number of symbols over a number of trading days. This work will be done on your laptop or desktop using the free version of KDB. You will have to parse and load the data into KDB and then use the Q language to calculate the VWAP numbers. You will be given the sample data set during Class 6 of the course.

Third, your VWAP code will be run against a data set of real financial data not less that 500GB and not greater than 10TB. Your Q code should assume the data has already been loaded into memory and that all your code must do—in the shortest time possible—is calculate the VWAP numbers for all symbols over all

[1] Usually, the database language for KDB is referred to as simply “q”. However, to make it clear in this document that I am referring to the q language, I use an uppercase “Q”.

days. In this case your code will be run on the SGI UV 300 shared-memory computer (12TB RAM, 384 processors, 64bit version of KDB).

In all three cases, each data record will have at least these three fields:

- Date and time stamp.
- Volume.
- Price.

A good part of this assignment is left unspecified (such as timestamp format) for a good reason. I want you to think about the project, do some research, and make intelligent choices. This is how real-world projects work! Whatever choices you make, just give your rationale in your one-page write up. Don't over think things, but do explain your choices and rationale. If you make reasonable and intelligent choices, your grade will reflect well on those choices.

3. <u>**What This Assignment is Meant to Test**</u>

This assignment is meant to test your skills at generating synthetic data sets.

It also gives you some experience with KDB, Q and in-memory databases.

4. <u>**Problem Specification**</u>

For all three test cases (section 2), write Q code (and any other code needed, though the whole assignment can in fact be written in Q, which you are encouraged to do) to carry out the assigned tasks. Output the results of your program in a way and in a format that makes sense to you; but document and justify your decision in your one-page write up.

At a minimum, your solution must:

- Accept time series of arbitrary length in the format described above.
- Generate output that gives a the VWAP numbers for the symbols and days in the time series data set.
- Describe your program(s) in no more than a single page of text and diagrams, paying particular emphasis on the tradeoffs you have made and anything you think novel in your approach. This need not be typed or drawn on a computer; a handwritten note, if clear and concise, is fine.

Final tests will be carried out on CUNY supercomputing machines. Development can be carried out mostly on your laptop or desktop. All programs and documentation must be submitted via your Github account.

5. <u>**Some Hints to Help You Along**</u>

When creating your synthetic data set make reasonable assumptions. Two symbols (stocks) is probably too few, one thousand too many. One day of data is too short, 30 years of intra-day data too long. On any given day, one trade for each symbol is too few, more than (say) 1000 trades per sec per symbol is too many. Assume that all records are actual trades (quotes have been removed). This is a Big Data course, so generate a data set that at least has the beginnings of a Big Data look and feel.

The goal here is to develop your synthetic data generation skills and get some basic familiarity with KDB & Q (which is a skill sought after in the job market). Don't obsess over the data side of things, concentrate on the programming and learning Q side of things.

As always, use you one-page write up to justify your decisions. If they seem reasonable—even if they simplify things a bit—then you will be graded accordingly. On the other hand, if they seem wildly simplistic or unreasonable, then your grade will suffer.

6. **How This Assignment Will be Graded**

Partial programs that do not meet all the requirements described above will receive credit. You will get marks for showing you were heading in the right direction.

Points will be allocated as follows:

- 20% for meeting the minimum requirements of the program specification. That is generating reasonable VWAP numbers for each of the data sets. You must also output the time(s) it too your program to process the data, accurate to 1ms.
- 30% for programming style. This encompasses not just the clarity of your code, but also such things as efficient data structures, efficient algorithms and use of system APIs and resources.
- 20% for scalability. If your programs run to completion against all input files of various sizes, you will receive 20%; otherwise you will receive some fraction of 20%.
- 30% for overall approach, technique and the one-page write up.

7. **Final Comments**

This is a challenging exercise, so if you find yourself struggling, reach out to me or Andrew (the TA) sooner rather than later. And remember that this is not a pass/fail exercise and that you can still get a good grade with a less than perfect program.