ECE-568 Software Engineering Web Applications

Final Project-Phase 1

Data Collection Module

Group Member & NetID

Sifan Yuan sy609, Dazhi Li dl939

Haocong Wang mw814, Mingming Pei mp1636

GitHub URL

https://github.com/DazhiLi-hub/SoftEng_WebAPP

1 Project Description & Requirement

To analyze the stock data and do prediction, we need to get the complete dataset of the stock information both real time data and historical data.

The project we do in phase 1 is to develop an application that runs continuously as a background process and periodically retrieves stock information, parses the received responses, and stores the extracted parameters into a local relational database. There are 10 stocks information for us to collect. For each stock, we need to store at least one-day real time data and one-year historical data. The real time data contain the real time price, the time stamp and the volume, and we define the time slice between two real-time points to 1 minute. The historical data contain time, open, high, low, close and volume.

To make the data fetching work easier, we collect the data from 2 different website: ***Yahoo!Finance*** and ***CNBC***. The database we used is ***MongoDB***. MongoDB is a none-relation database, which is very suitable for our application. What's more, MongoDB is very fast to operate the data, and the ***.json*** file create by MongoDB performs pretty well for text type file reading and writing.

At the data fetching part, in order to get the real time data, we use python to write a crawler algorithm to get the data from the website. In this algorithm, we use regular expression operation to locate the data we want to get.

But, unfortunately, we can't find historical data from the website, which means that there's no way to use crawler algorithm to get it. To solve this, we use the python package called ***Pandas Data Reader***. It can directly connect to yahoo's API to get the historical data.

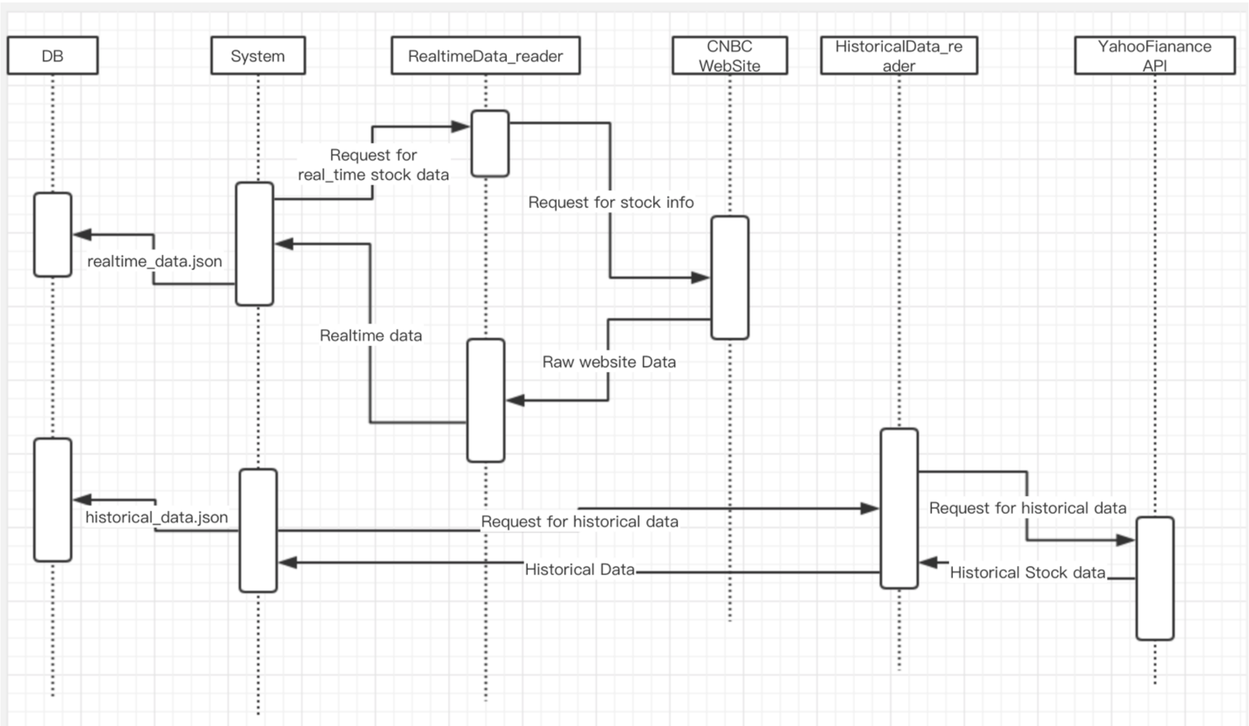
At the database writing process, we use a library called ***pymongo***. With this library, we can operate the MongoDB database in python environment.

For each company, we create 2 tables for them, one is for the real-time data, one is for the historical data. Through this way, we can easily organize the collected data, also, pretty easy to get the data from database. There are 20 tables in total, to avoid the collision, for each table, we define "time" as the primary key.

Note:

You can find the source code from both Sakai.Assignment and GitHub
url: https://github.com/DazhiLi-hub/SoftEng_WebAPP

2 System Design Diagram



From this UML diagram, you can easily get the system design of our application.

We use two method to collect different data from different website. Then, create json file and operate the database.

The detail description of our application can be found in the first part of this report.

3 Target

- Collecting data from Yahoo Finance
- Stocks included:GOOG(Google), MSFT(Microsoft), AAPL(Apple), NVDA(NVDIA), BTC-USD(Bitcoin), AMZN(Amazon), OVTZ(Oclus Vison Tech.), IBM(IBM), AMD(AMD), INTC(Intel)
- Utilized language: Python
- Database: MongoDB(Non-relation)
- Data transfered datatype: .json
- Data are divided into 2 parts: History data (2017-1-1 to Now), Realtime data (One day stock price&volome per minute)
- Runing Final_collector.py to enter, continuously fetching real-time data until keyborad interrupt (Ctrl + C)

4 Requirement

- apscheduler
- pandas
- pandas-datareader
- datetime
- requests
- re
- json

5 References

https://blog.csdn.net/Hellolijunshy/article/details/82527643
https://blog.quantinsti.com/stock-market-data-analysis-python/
https://blog.csdn.net/huanbia/article/details/72674832
https://blog.csdn.net/wyongqing/article/details/46738405
https://www.cnblogs.com/bigberg/p/6430095.html