Систем за обработка на финансиски податоци во реално време

1. Вовед

Системот за обработка на финансиски податоци во реално време е дизајниран за собирање, обработка и визуелизација на податоци од берзата. Овој систем овозможува следење на цените на акциите, технички индикатори, сентимент анализа и предвидување на идните движења на цените. Главните функционалности на системот вклучуваат:

- Собирање на податоци за цените на акциите во реално време
- Обработка на податоците со помош на *Apache Spark*
- Пресметка на технички индикатори (подвижни проседи, *RSI, MACD, Bollinger Bands*)
- Сентимент анализа на пазарот
- Предвидување на идните цени на акциите
- Интерактивен веб интерфејс за визуелизација на податоците

2. Архитектура на системот

Системот е изграден со микросервисна архитектура и се состои од следните компоненти:

2.1. Компоненти за собирање на податоци

- ***Yahoo Finance API* интеграција**: Собира податоци за цените на акциите од *S&P 500* индексот
- ***Kafka Producer***: Ги праќа собраните податоци во *Kafka* теми

2.2. Компоненти за обработка на податоци

- ***Apache Kafka***: Служи како централен систем за стриминг на податоци
- ***Apache Zookeeper**:* Обезбедува координација за *Kafka* кластерот
- ***Apache Spark***: Врши обработка на податоците во реално време
- ***Spark Streaming***: Чита податоци од *Kafka* и врши анализа

2.3. Компоненти за визуелизација

- ***Dash***: Рамка за создавање на интерактивни веб апликации

- ***Plotly***: Библиотека за создавање на интерактивни графикони
- ***Flask***: Веб сервер за хостирање на *Dash* апликацијата

2.4. Инфраструктура

- ***Docker***: Контејнеризација на сите компоненти
- ***Docker Compose***: Оркестрација на контејнерите

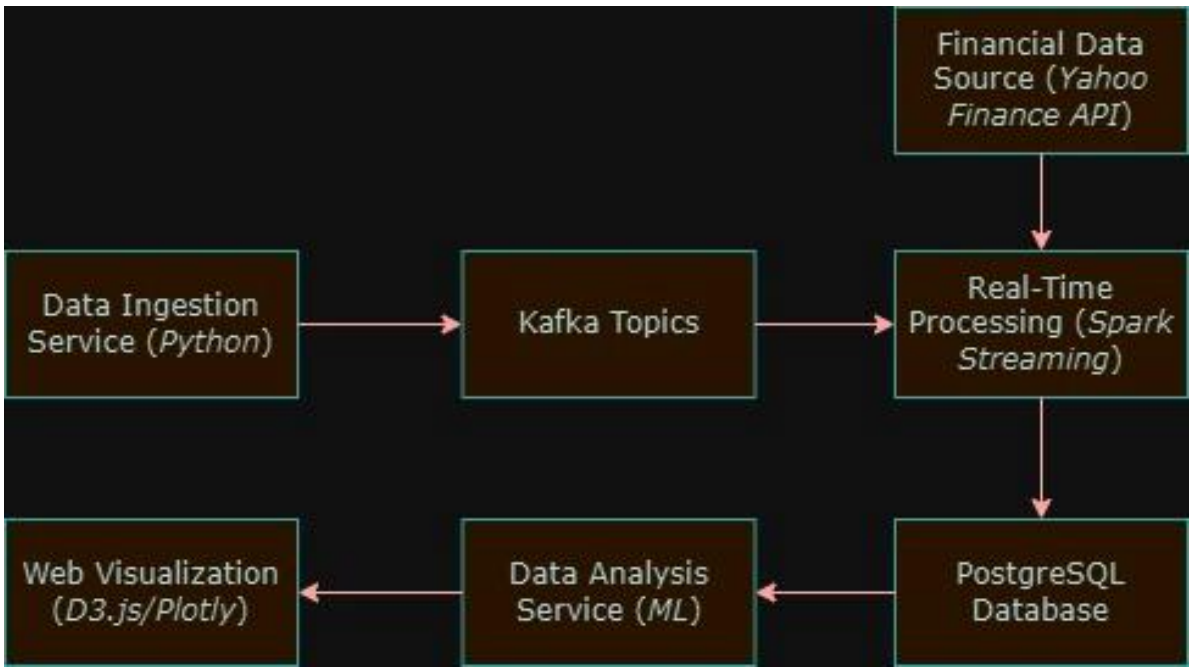
3. Технички детали

3.1. Собирање на податоци

Системот користи *yfinance* библиотека за собирање на податоци од *Yahoo Finance API*. Податоците се собираат за сите компании од *S&P 500* индексот и вклучуваат:

- Цени на отворање, затворање, највисока и најниска цена
- Волумен на тргување
- Датум и време

Кодот за собирање на податоци се наоѓа во модулот *data_ingestion*.



3.2. Обработка на податоци

Собраните податоци се праќаат во *Kafka* тема stock_data. *Spark Streaming* ги чита овие податоци и врши следните операции:

- Пресметка на технички индикатори
- Подвижни проседи (5-дневен и 20-дневен)
- Релативен индекс на сила (*RSI*)
- *MACD* (Moving Average Convergence Divergence)
- Bollinger Bands
- Сентимент анализа
- Предвидување на идните цени

Обработените податоци се зачувуваат во соодветни структури за понатамошна визуелизација.

3.3. Визуелизација

Веб интерфејсот е изграден со *Dash* и *Plotly* и овозможува:

- Избор на компанија од *S&P 500* индексот
- Избор на временски период (1 недела, 1 месец, 3 месеци, 6 месеци, 1 година)
- Приказ на графикони за цената на акцијата
- Приказ на технички индикатори
- Приказ на сентимент анализа
- Приказ на предвидување на идните цени

Интерфејсот е дизајниран да биде интуитивен и лесен за користење, со респонзивен дизајн што работи на различни уреди.

3.4. Докеризација

Системот е целосно докеризиран, што овозможува лесно распоредување во различни околини. *Docker Compose* конфигурацијата ги дефинира следните сервиси:

- ***zookeeper***: *Apache Zookeeper* за координација на *Kafka*
- ***kafka***: *Apache Kafka* за стриминг на податоци
- ***spark-master***: Главен јазол на *Spark* кластерот
- ***spark-worker***: Работен јазол на *Spark* кластерот
- ***financial-app***: Главната апликација што ги содржи сите *Python* сервиси

4. Инсталација и користење

4.1. Предуслови

- *Docker* и *Docker Compose*
- *Git* (опционално)

4.2. Инсталација

```
git clone https://github.com/korisnik/financial-data-system.git
    cd financial-data-system
    docker-compose build
    docker-compose up
```

4.3. Користење

1. Отворете го веб интерфејсот на адреса http://localhost:5000
2. Изберете компанија од паѓачкото мени
3. Изберете временски период
4. Кликнете на "Fetch Data" за да ги вчитате податоците
5. Истражувајте ги различните графикони и анализи

5. Структура на проектот

```
financial-data-system/
├── data_ingestion/
│   └── ... (модули за собирање на податоци)
├── real_time_processing/
│   ├── kafka_consumer.py
├── data_visualization/
│   ├── app.py
├── Dockerfile
├── docker-compose.yml
└── start_system.py
```

6. Технички предизвици и решенија

6.1. Интеграција на Spark со Kafka на Windows

Еден од главните предизвици беше интеграцијата на *Spark* со *Kafka* на *Windows* околина. Решението вклучуваше:

- Инсталација на *Hadoop* библиотеки за *Windows*
- Поставување на соодветни променливи на околината (*HADOOP_HOME*)
- Конфигурација на *log4j* за соодветно логирање

6.2. Справување со грешки при вчитување на податоци

Системот имплементира механизам за справување со грешки при вчитување на податоци од *Yahoo Finance API*:

- Обид за вчитување на податоци од *API*
- Логирање на грешките за понатамошна анализа

6.3. Оптимизација на перформансите

За да се обезбедат добри перформанси, системот:

- Користи кеширање на податоци каде што е соодветно
- Оптимизира *Spark* задачите за ефикасна обработка
- Имплементира асинхроно вчитување на податоци во веб интерфејсот

7. Идни подобрувања

Планирани идни подобрувања на системот вклучуваат:

- Додавање на повеќе извори на податоци (*Bloomberg, Alpha Vantage*, итн.)
- Имплементација на понапредни алгоритми за машинско учење
- Додавање на функционалност за известување (е-пошта, мобилни пораки)
- Подобрување на скалабилноста на системот
- Имплементација на автентикација и авторизација
- Додавање на функционалност за персонализирани портфолија

8. Заклучок

Системот за обработка на финансиски податоци во реално време претставува моќна алатка за анализа на пазарот на акции. Со комбинирање на модерни технологии како *Apache Kafka, Apache Spark* и *Dash*, системот овозможува собирање, обработка и визуелизација на финансиски податоци на интуитивен и ефикасен начин. Докеризацијата на системот обезбедува лесно распоредување и скалирање, додека модуларната архитектура овозможува лесно проширување со нови функционалности во иднина.