COMP7705 Project

Detailed Project Proposal

Project Title: **Real-time Cryptocurrency Analysis System**

Supervisor: Dr.T.W.Chim

Student 1 (Leader) Liang, Zhihao, 3035561651

Student 2 Wang,Xue, 3035562289

Student 3

Student 4

Student 5

Aim

Objectives:

1. Achieve a cryptocurrency price prediction system which has a large scalability and fault tolerance in real-time
2. Build a Machine Learning Model which can support large volume of Natural Language Processing queries in real-time

Since the creation of Bitcoin, cryptocurrencies are attracting significant attentions from researchers. They have been proposing many solutions for analysing the price trend. One dimension of these researches is to analyse the sentiment trend in social media like Twitter and Reddit. Some of these solutions even implement near real-time processing on Spark framework. However Spark is a framework dedicated for batch processing, which suffers from high latency. To minimize latency, Spark has implemented streaming API by applying micro-batch processing. But its performance in iterative or interactive applications is still unsatisfactory. In the area of capital market, the price fluctuation is very fast. Analytics and stakeholders are demanding a timely system that can assist their decision making. In this background, the demand for a truely real-time crypotocurrency analysation platform is rising rapidly. **In this paper, we propose a Flink-based cryptocurrency analysation system that can handle massive amount of data in real-time. Streaming data is evaluated continuously and the result is updated in seconds, not days or months.**

Brief Literature Review

Cryptocurrency

Cryptocurrency is a kind of digital asset that's decentralized and secured by strong cryptography algorithms. Satoshi Nakamoto created the first generation of cryptocurrency, Bitcoin in 2009. The validity of Bitcoin is provided by blockchain technology. A blockchain is a continuously growing list of records which is linked by hash function. Hash function ensures that none of the records can be modified without being caught by other.

The Efficient Market Hypothesis

The Efficient Market Hypothesis states that current stock prices have reflected all the available information. And price variation is largely driven by incoming information. These new information broadcasts on social media like twitter and reddit rapidly. Researchers have devoted to find the correlation between public mood and stock price. One approach is to do sentiment analysis on tweets by applying machine learning algorithms.

Traditional ETL and its Limits

For many years, ETL (Extract, Transform and Load) is the mainstrem procedure for business intelligence and data analysis. The objective of ETL is to extract data from source system, apply some transformation, and finally load into target data store. However traditional ETL systems are limited by their scalability and fault tolerent ability. According to a report presented in 2017 by IDChttps://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdfthe global data volume will grow expronentially from 33 zettabytes in 2018 to 175 zettabytes by 2025. IDC also forecasts that we will have 150 billions devices connected globally by 2025. And real-time data will account for around 30 percents of the global data. We demand for a system that's able to distribute computations to thousands of machines and runs parallely.

MapReduce

https://dl.gi.de/bitstream/handle/20.500.12116/20456/327.pdf?sequence=1https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

MapReduce is a programming model that is able to process vast amounts of datasets in parallel. It's inspired by the map and reduce operation in functional languages like Lisp. MapReduce is compose of three core operations: map, shuffle and reduce. A job is usually splited into multiple independent subtasks and run parallely on the map stage. Then the outputed data from map stage is shuffled by its key, such that data with the same key occurence on the same workder node. Finally, reducers start processing each group of data in parellel.

Hadoop

Hadoop is consist of Hadoop Distributed File System(HDFS) and Hadoop MapReduce framework. It's inspired by GFS and MapReduce. **Note: we will supplement more literature review and background information later**

Kappa architecture and Lambda Architecture

To accomodate the need for both high throughput and low latency, (N. Marz and J. Warren. Big data: principles and best practices of scalable realtime data systems. Manning, 2013.) proposed a mixed architecture: **lambda architecture**. Lambda architecture is a data

processing paradigm that is capable of dealing with massive amount of data. It mixes both batch and stream processing methods. Lambda architecture is compose of batch layer and speed layer. The batch layer is focus on increasing the accuracy by taking account into all available data. The result produced by batch layer is equivalent to equation "query result = f(all data)". Where f is the processing logic for the data. The speed layer is focus on providing immediate view to the new incoming data. Query from clients are answered through the serving layer, which merges result from both batch layer and speed layer.

Kappa architecture is a simplified architecture with batch processing system removed. It enable analytics to do data processing with a single technology stack.

Spark

Note: we will supplement more literature review and background information about spark and compare it with flink later

Flink

Apache flink is a distributed stateful stream processing framework.
Building blocks of flink: https://flink.apache.org/flink-applications.html

- **Stream**: Bounded, unbounded
- **State**:
- **Time**: Event-time, Ingestion time, Processing time

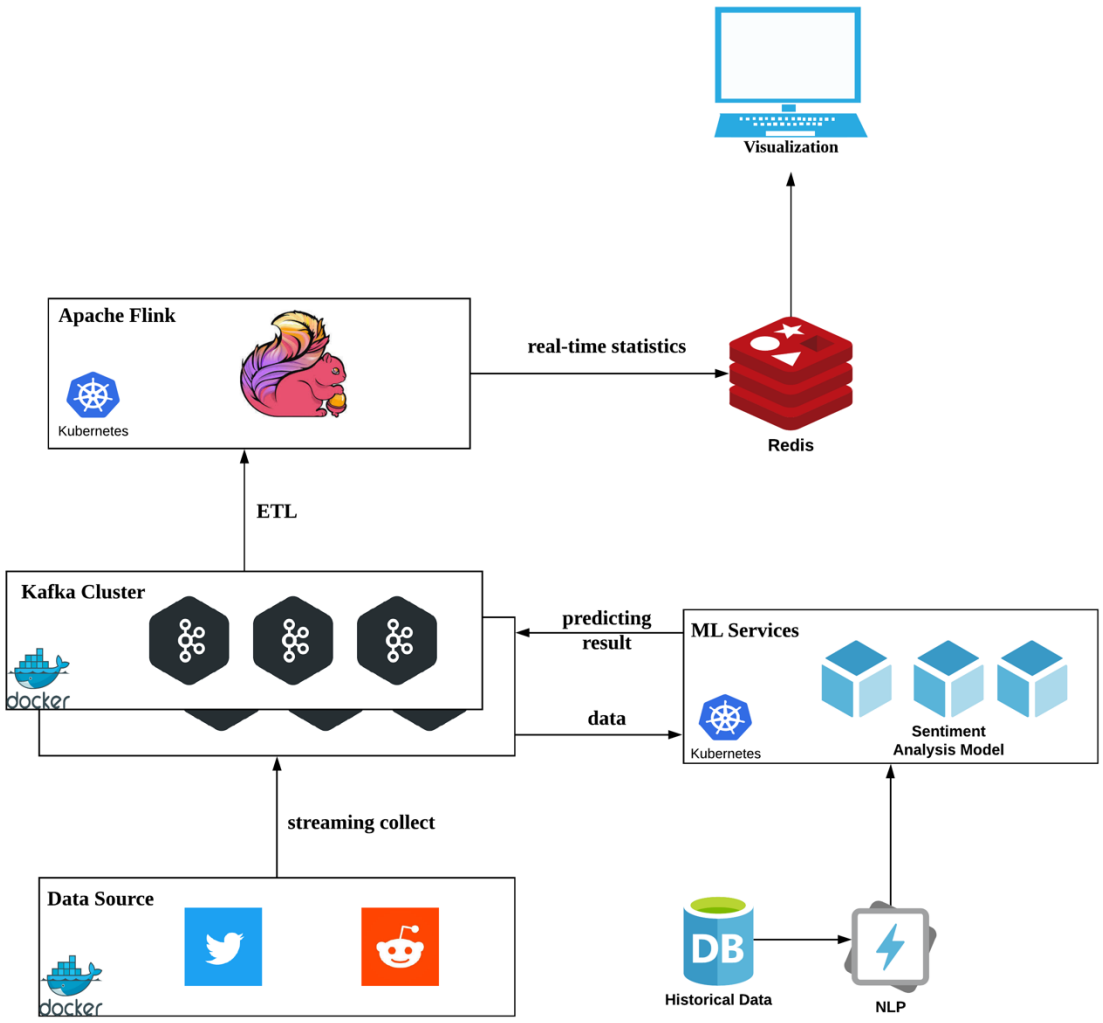
Characteristics of flink

- High throughput
- Low latency
- Exactly once semantcs
- Event Processing
- State management
- Time sementics
- Fault tolerent

Flink guarantees exactly-once state consistency in case of failures by periodically and asynchronously checkpointing the local state to durable storage.
Note: we will supplement more literature review and background information later

Proposed Methodology

Here is our system architecture.



Technologies that we will use include Twitter/Reddit Streaming API, Kafka, Flink, Nginx, Docker and Redis. Due to implementation complexity, we might not take the advantage of Kubernetes at this time.

Milestones

	Tasks	***Estimated completion time***	***Estimated number of learning hours***
1	Project Webpage	1st June, 2020	15
2	Streaming data source module development	1st June, 2020	30
3	Sentiment Analysis Module Development	15th June, 2020	70
4	Kafka Cluster Setup	1st July, 2020	30
5	Apache Flink Cluster Setup	1st July, 2020	30
6	Real-time Data Aggregation with Flink	1st July, 2020	60
7	Web UI and Visualization	15th July, 2020	30
8	Poster	15th July, 2020	20
9	Final Report	1st August, 2020	60
10	Revised Final Report	15th August, 2020	10
			Total: 345

Deliverables

Items	
1	Project Webpage
2	Streaming data source module
3	Sentiment Analysis Module
4	Building Kafka Cluster
5	Buiding Apache Flink Cluster
6	Real-time Data Aggregation with Flink
7	Web UI and Visualization
8	Poster
9	Final Report submission
10	Revised Final Report submission