Big Data Solutions for Financial Market Data

Paul Adams, *paula@smu.edu*, Rikel Djoko, *rdjoko@smu.edu*, and Stuart Miller, *stuart@smu.edu*

I. Research Problem

We want to build a large-scale data framework that will enable us to perform rapid querying of financial market data and supporting data to drive future predictions for investment.

We are going to install a data lake ecosystem on the cloud to house structured and unstructured data that we will then extract, translate, and load (ETL) into the data warehouse. To accomplish this, we will use Apache Hadoop for the lake, Amazon AWS for the cloud, Apache NiFi for ETL, and Apache Hive for the data warehouse. Through this approach we will optimize the data warehouse to support relatively unlimited parallel processing that will enable future business modeling for predicting stock prices.

II. Research Methodology

As a group, we would like to gather data to make predictions within the financial markets for directing investment and business development. This not only includes investment opportunity, but potentially business development consulting based on market intelligence. The data feeds for these sources will be generated as described in section III. Table I provides an example of the types of data that will be gathered from the sources shown. For this, we need a database that will allow the storage of many data types. Within the scope of this project, however, we will focus largely on unstructured natural language data as well as structured data gathered from the web. Consequently, we need a database framework that supports the collection and normalization of both structured and unstructured data as we will find the most use for the end-state of the data existing in a structured design.

In conventional databases, a database is installed on a server where only that server can be used for the database. With Apache's Hadoop Distributed File System (HDFS), data can be distributed from one database across multiple servers. This allows data to process much more quickly, in turn allowing storage of many data types. We are interested in using Hadoop as a data lake for its data diversity; it supports storage of multiple file types of both structured and unstructured form. This enables us to be relatively boundless in the types of data we can gather. Currently, we seek to gather a combination of both natural language and structured data, but would like the ability to store things such as images and emoji data if this becomes opportunistic. Furthermore, HDFS distributes data via a programming model called MapReduce. MapReduce functions on a split-apply-combine strategy that enables parallel, distributed processing across clusters of server nodes. While we will initially use only one server, this ability supports future data expansion.

To structure the data via the process called Extracting, Transforming, and Loading (ETL), we are interested in utilizing the Apache NiFi software, which supports the use of a Structured Query Language (SQL)-like language called Hive Query Language (HQL). HQL is abstracted with underlying Java code, which supports Application Programming Interfaces (APIs) through which custom functionality can be delivered to support many tasks, such as connecting to various databases and other data sources. This will be outside the scope of this project, but is important to understand since it present the opportunity to further advance our business model in the future.

Finally, once this data has been structured, we will load it into a data warehouse. Because HQL has a familiar syntax with SQL, which performs very well for storing and retrieving structured data and implementing such processes into automation not limited to, but including web user interfaces (web UIs), we will use Apache Hive as the data warehouse. Hive pairs very well with Hadoop and NiFi and will allow us to access our data with simplicity, providing downstream processing a normalized format with which to analyze data.

Because this data will need to be accessed by the multiple users in our team and will need to scale well – meaning, for example, in the event we need more than one server for storage – we will store our data on the cloud. The cloud we will use is Amazon Web Services (AWS). Amazon maintains a large network of servers in multiple locations in order to minimize latency in data access speeds and maximize data security, such as during national disaster. AWS supports MapReduce in its Elastic Map Reduce (EMR) cluster platform that supports processing within the Hadoop ecosystem. Among this ecosystem are tools such as Apache Spark, which supports the programming of business logic that can be applied using a number of programming languages, including Java, R, and Python, among many others. This supports a large breadth of analytical tools, including machine learning, deep learning, and neural networks.

III. Previous Related Work

This section relates to Big Data warehouse systems that will be used for housing the data and data requirements for natural language processing.

A. Data Warehousing

This project focuses on housing structured and unstructured data so that it can be accessed quickly for analysis. Traditionally, relational databases have been the bedrock of data warehousing systems. However, traditional database systems were designed to handle *smaller sets of structured* data [1]. Big Data tools were developed to address the gap between

TABLE I
Data Types in Warehouse

Data Type	Examples of Data Variables
Stock Symbol Data (Yahoo)	Date; Open Price, High Price, Low Price, Close Price, Volume.
Stock Symbol Financials (Yahoo)	Revenue, profit, and expense values for previous 4 years.
Conversations (Yahoo)	Raw text, user names, date and time of post, reactions to post
Social media posts (Twitter, StockTwits)	Raw text, emojis, hashtags, tweet reactions
News (NYT, CNN, etc)	Raw text, publication date

Types of data and variables expected to be stored in the database.

traditional database systems and the growing volume of unstructured data. Currently, Apache Hadoop is the de facto system for large scale data warehousing. Apache Hadoop runs on a distributed file system called MapReduce. MapReduce abstracts the complex details of compute cluster management away from the developer [2]. Amazon Elastic MapReduce (EMR) is a managed Hadoop framework, which provides scalable, cost-effective solution for processing large amounts of data [1]. Amazon EMR can be scaled dynamically across Amazon Elastic Compute 2 (EC2) instances as the data size increases.

B. Data Requirements for Natural Language

Before natural language (NL) data can be used in a machine learning system, it must undergo some linguistic preprocessing. Tokenization and Parts of Speech tagging are examples of typical preprocessing steps required for NL applications [3]. Either raw language data must be stored in a form suitable for these types of transformations or the transformed versions of the raw data should be stored themselves. R. Agerri, X. Artola, Z. Beloki, G. Rigau, and A. Soroa presented a solution for preprocessing NL data using Apache Storm where the raw NL data was preprocessed through IXA pipelines in parallel on virtual machines (VM) [5, 6]. Newer natural language data available on the web, especially data from Twitter, contains special indicators called hashtags and special characters called emojis, which can be used to annotate and group tweets (or related documents) [4].

IV. Research Plan and Schedule

1. Business Requirements Development – Week One
   - This is a one week process that involves understanding the data we need to accomplish our goals. Furthermore, it involves vetting the data to ensure it is an appropriate fit.
2. Data Gathering – Weeks Two and Three
   - This phase involves wrangling our data. During this process, we will begin accessing and storing data from our identified sources. We will clean as much as needed, which includes parsing out unnecessary data, such as stop-words, and removing special characters in preparation for ingestion into Hadoop.
3. Data Structuring: Weeks Three and Four
   - During this time, we begin – in conjunction with the gathering phase – structuring the data into files that we will load into Hadoop. These files must exist as entities within Hadoop so during this phase, we structure them accordingly, where like-data will be associated with like-data.
4. Installing Data Lake, NiFi, Data Warehouse on Amazon AWS: Week Five
   - This phase involves using Amazon EMR to install the configuration on the server we need to process our data.
5. Develop Automation Workflow: Weeks Six and Seven
   - This phase will be labor-intensive and involve writing HQL to process the data from the lake-side into the warehouse according to the format required. Focus involves importing, formatting, and testing. We will also run queries in Hive to ensure proper functionality.
6. Project Report Development, Go-Live: Weeks Eight and Nine
   - During this phase, we complete our project. We will develop the final report and demonstration template for presentation of our development to the class. This period will also enable us to ensure there are no bugs in our system and if there are, that we have enough time to debug.

V. Required Resources

The following items are required for completion of the project.

- Amazon EMR
- Amazon EC2 Instances
- Historical stock prices
- Relevant text data
  - News
  - Twitter
  - StockTwits

References

[1] R. Kune, P. Konugurthi, A. Agarwal, R. Chillarige, R. Buyya, "The Anatomy of Big Data Computing," *Software: Practice and Experience*, Vol.46 no. 1, pp.79-105, Jan. 2016.

[2] "Extract, Transform, and Load Big Data with Apache Hadoop," Intel, USA, 2013. Available: https://software.intel.com/sites/default/files/article/402274/etl-big-data-with-hadoop.pdf

[3] S. Bird, E. Klein, and E. Loper, "Categorizing and Tagging Words," in *Natural Language Processing with Python*, 1st ed. Sebastopol, CA: O'Reilly Media, 2009, pp. 179-213.

[4] B. Guthier, K. Ho, and A. El Saddik, "Language-independent data set annotation for machine learning-based sentiment analysis" in *Proc. IEEE SMC*, 2017, pp. 2105-2110.

[5] R. Agerri, X. Artola, Z. Beloki, G. Rigau, and A. Soroa, "Big Data for Natural Language Processing: A streaming approach," *Knowledge Based Systems*, Vol. 79, pp. 36-42, 2015.

[6] R. Agerri, J. Bermudez and G. Rigau, "IXA pipeline: Efficient and Ready to Use Multilingual NLP tools", in *Proc. of the 9th LREC*, Reykjavik, Iceland, 2014, pp. 26-31