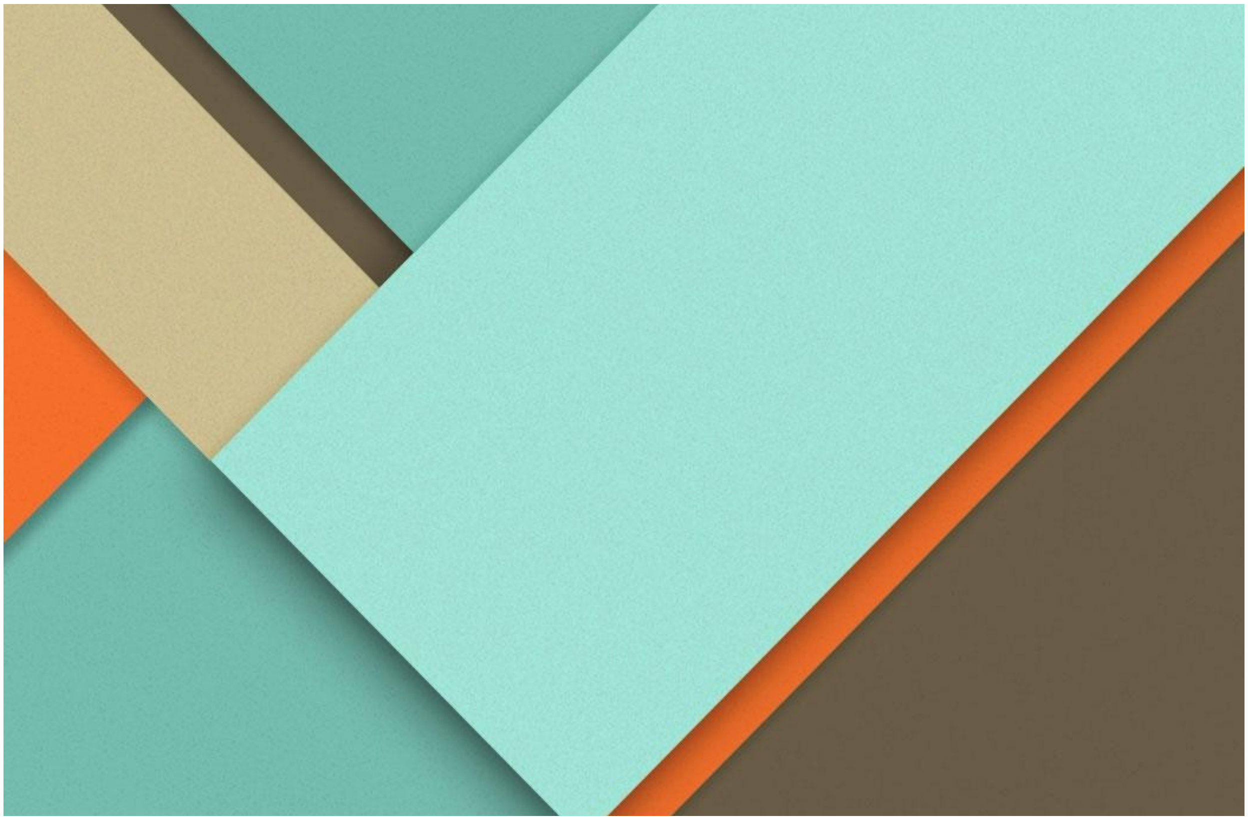

Data Warehousing Solution

Team Id: 03

Ankur Shrivastava (201405551)

Tushan Jain (201405560)

Pulkit Agrawal(201201188)

Anurag Tyagi(201303012)

Abstract

In computing, a **Data Warehouse** (**DW** or **DWH**), also known as an **Enterprise Data Warehouse** (**EDW**), is a system used for reporting and data analysis. DWs are central repositories of integrated data from one or more disparate sources. They store current and historical data and are used for creating analytical reports for knowledge workers throughout the enterprise. Examples of reports could range from annual and quarterly comparisons and trends to detailed daily sales analyses.

Background Analysis

Data vendors provide industry classification for each company, and it helps a lot in industries like retail (Wal-Mart is good comparable to Costco), energy (Chevron and Exxon Mobil) but it stumbles with many other companies. People tend to compare Amazon with Google as a two major players in it business, but data vendors tend to put Amazon in retail industry with Wal-Mart/Costco as comparables.

We came up with an idea that if companies are often mentioned in news articles and tweets together, it’s probably a sign that people think about them as comparable companies. In this post I’ll show how we built proof of concept for this idea with Spark, Spark Streaming and Cassandra. We use only Twitter live stream data for now, accessing high quality news data is a bit more complicated problem.

From this tweet we can derive 2 mentions for 2 companies. For Facebook it will be Twitter and vice-versa. If we collect tweets for all companies over some period of time, and take a ratio of joint appearance in same tweet as a measure of “similarity”, we can build comparable company recommendations based on this measure.

Usecase

Consider a tweet

“Trying to spot the next #FB or #TWTR? These 10 startups are worth keeping an eye on http://cnn.it/1CHur0d”

From this tweet we can derive 2 mentions for 2 companies. For Facebook it will be Twitter and vice-versa. If we collect tweets for all companies over some period of time, and take a ratio of joint appearance in same tweet as a measure of “similarity”, we can build comparable company recommendations based on this measure.

Approach

1. We will use Spark Streaming Twitter integration to subscribe for real-time twitter updates.
2. Then we will extract company mentions and put them to Cassandra.
3. Count mentions for each pair of tickers.
4. After aggregates computed, we sort them globally and then group them by key (Ticker). After all aggregates grouped we produce recommendation in single traverse distributed for each key.

Few libraries which might be of use are and their description:

Spark SQL :- Spark SQL provides the capability to expose the Spark datasets over JDBC API and allow running the SQL like queries on Spark data using traditional BI and visualization tools. Spark SQL allows the users to ETL their data from different formats it’s currently in (like JSON, Parquet, a Database), transform it, and expose it for ad-hoc querying.

MLLIB :- Machine Learning library in spark.:MLLIb is Spark’s scalable machine learning library consisting of common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, dimensionality reduction, as well as underlying optimization primitives.

GRAPHX :- Graph parallel computation in spark.:GraphX is the new (alpha) Spark API for graphs and graph-parallel computation. At a high level, GraphX extends the Spark RDD by introducing the Resilient Distributed Property Graph: a directed multi-graph with properties attached to each vertex and edge. To support graph computation, GraphX exposes a set of fundamental operators (e.g., subgraph, join Vertices, and aggregate Messages) as well as an optimized variant of the Pregel API. In addition, GraphX includes a growing collection of graph algorithms and builders to simplify graph analytics tasks.

Spark Streaming :- It can be used for processing the real-time streaming data. This is based on micro batch style of computing and processing. It uses the DStream which is basically a series of RDDs, to process the real-time data.

BlinkDB :- BlinkDB is an approximate query engine and can be used for running interactive SQL queries on large volumes of data. It allows users to trade-off query accuracy for response time. It works on large data sets by running queries on data samples and presenting results annotated with meaningful error bars.

Tachyonis :- Tachyonis a memory-centric distributed file system enabling reliable file sharing at memory-speed across cluster frameworks, such as Spark and MapReduce. It cache working set files in memory, thereby avoiding going to disk to load datasets that are frequently read. This enables different jobs/queries and frameworks to access cached files at memory speed.

Why Apache Spark ?

- Spark comes with GraphX, a distributed graph system.
- Spark can run on Hadoop alongside other tools in the Hadoop ecosystem including Hive and Pig.
- Faster batch processing than MapReduce.

- Spark executes batch-processing jobs 10 to 100 times faster than MapReduce.
- Spark is ideal for iterative processing, interactive processing and event stream processing.
- Combine Sql, streaming and complex analytics

End Result Expected

We will get a system that can analysis the trending data from twitter and can give comparable company recommendations build from Twitter stream as mentioned in the use case.