Meetup Data ETL Progress Update
Karin Brood, Chandler McCann, Natarajan Shankar, Dan Watson

Project Idea

Our goal is explore how people come together. With the availability of data from Meetup.com, we seek to explore what activities are most popular throughout different regions of the country. By categorizing and quantifying the data from different Meetups, we hope to identify the most popular activities that bring people together in different regions of the country.

After finding out the most popular activities in different regions, we'll look for other interesting similarities and differences in the country. We want to explore if similar activities are popular throughout the entire country, or if activities vary by region. Hopefully we'll be able to discover dissimilar regions with similar activity profiles and be able to identify factors that lead to the same outcome in meetup activity.

The project has multiple use cases. From helping people move to areas where their interests are most popular to helping marketers gain a better understanding of what make certain communities tick. We also see positive societal outcomes in being able to provide services that communities demand, but are currently unavailable.

Changes in Project Architecture

Since our initial proposal, our team has decided to make changes in the project architecture to better answer the research question: What activities are communities throughout the United States excited about?

Rather than analyzing streaming data, our research question is better served by batch processing large quantities of data for multiple regions. The general process flow is:

- Connect to Meetup's streaming API
- Capture JSON data from the API and store in MongoDB
- Integrate Pyspark to process the data
- Connect data source to Tableau for visualization

Below, we cover a more in-depth discussion of the project organization strategy and tools used to complete the analysis.

Project Framework:

This ETL infrastructure is driven by data from Meetup.com and consumes, extracts, transforms, modifies and summarizes meeting topics and context.

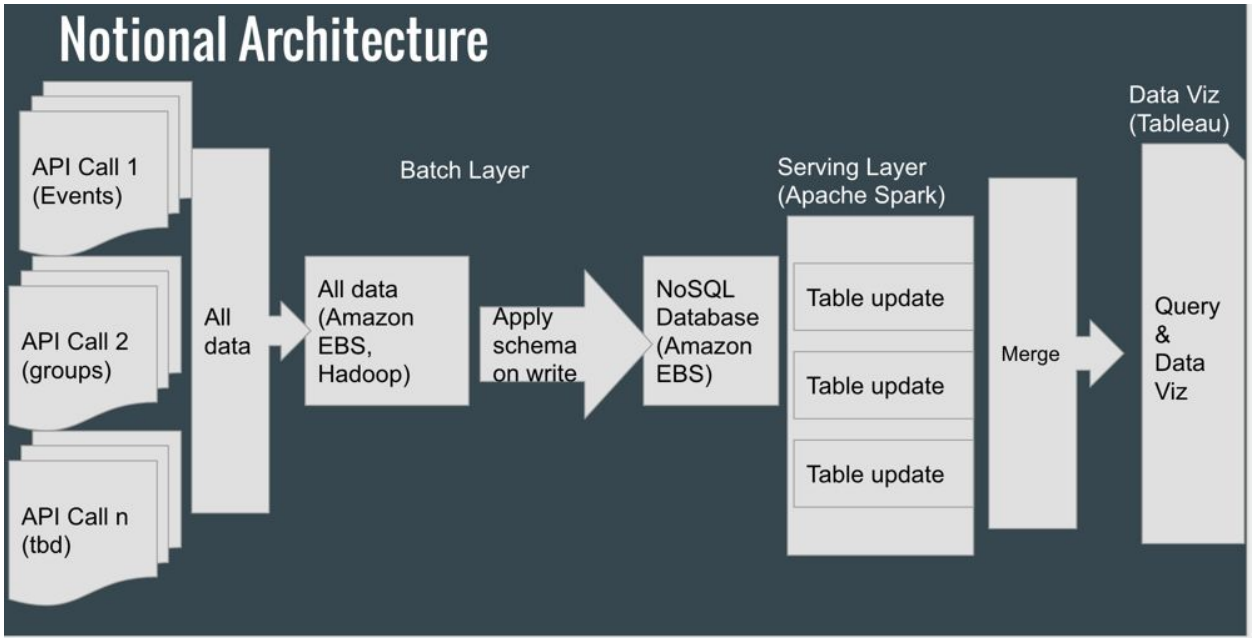

Figure 1: Notional architecture for Meet-Up data ingest, transform, process and query

Meetup.com interface

- A data spout has been created via a GET Categories API call into the api.meetup.com host.
- To throttle the spout in this implementation stage, the GET Categories has been restricted to a list of 5 prominent US cities. Although restrictive, this keeps the data pipe full.
- Meetup.com APIs do not provide an extensive filtration capability. The GET Categories API does not offer a data filter and meetup events are reported by City with an associated epoch that cannot be filtered up at the source.
- The data stream from Meetup.com is then fed into a batch store, a Mongo DB instance.
    - To accomplish this interface:
        - A Mongo client is being implemented
        - This client will open a TCP session with the Mongo DB Server
        - A data filter will filter the streams data prior to writing it to the Server

MongoDB

MongoDB will be the the storage solution for our data as it comes in from the Meetup API. A mongo client will receive the Meetup JSON from our API interface, and send to data to be stored in a document collection on our Mongo server, utilizing the pymongo package to accomplish this. There will be only minimal processing of the Meetup data before it is stored, with the processing consisting of deserializing the raw JSON before writing the data.

Our MongoDB setup is currently a single server, which we expect should be sufficient for the purposes of this assignment. However, our architecture does provide the option to scale out through sharding should be become necessary.

Data Analysis

To perform analysis we will integrate Spark and MongoDB. Mongo-hadoop now comes with a package called pymongo-spark, which allows PySpark to interact with PyMongo, the MongoDB Python driver. Our data is structured according to the schema defined by the Meet-Up API, and we will be using Spark to build the tables we need to answer our research question. Specifically, we'll be using PySpark to:

- Find the total number of groups by city.
- For each city, find: 1) the number of members for each category type, like "tech;" 2) the growth by year; and 3) the popularity of each category group.

Mongo can support basic aggregations, however if we have the bandwidth to explore more in-depth statistical/ machine learning analysis, PySpark provides that functionality. We will be creating data frames as Spark RDDs ,directly aggregating on the cities and categories. From there, we will then store these as persistent tables to connect to Tableau Server via the Spark ThriftServer.

Visualization

Our back-end layer for visualization will be Tableau, which easily connects to SparkSql through Spark ThriftServer. This will allow for interesting graphical displays showing growth of communities over time, as well as basic graphical analysis for our research question. Tableau easily scales as we add more tables, and allows for the merging of tables within Tableau as new tables from Spark are generated.

Current State of Project

At this point in time we have made the key decisions regarding what questions we wish to answer and what data we will require from the raw meetup data to answer those questions as already detailed. Substantial progress has been made on our data pipeline with acquisition of of the data from Meetup APIs and storage of that data into MongoDB, as well as the standing up of our MongoDB server. Only some work is left to connect the stages to have a complete pipeline from the Meetup API to Mongo. We have made the determination to use Tableau for our visualization/dashboard which will be able to connect to a Spark ThriftServer to query RDDs created as part of our data analysis. Work on our spark jobs to build the required RDDs is still ongoing but a basic proof of concept is in place.