Project Details

Sl No	Field	Value	IdentiKey
1	Team Member 1	Prashil Bhimani	prbh9460
2	Team Member 2	Sharan Srivatsa	shsr7814

Problem Description

Data Analysts and Data Scientists are often tasked with looking at very large datasets, sometimes datasets that do not even fit on their machine. This is a common situation especially for students who cannot afford high-end machines as many Software Companies can. Looking at our own course, during the first assignment our TA faced a similar challenge with the Gutenberg dataset while uploading the same to blob storage.

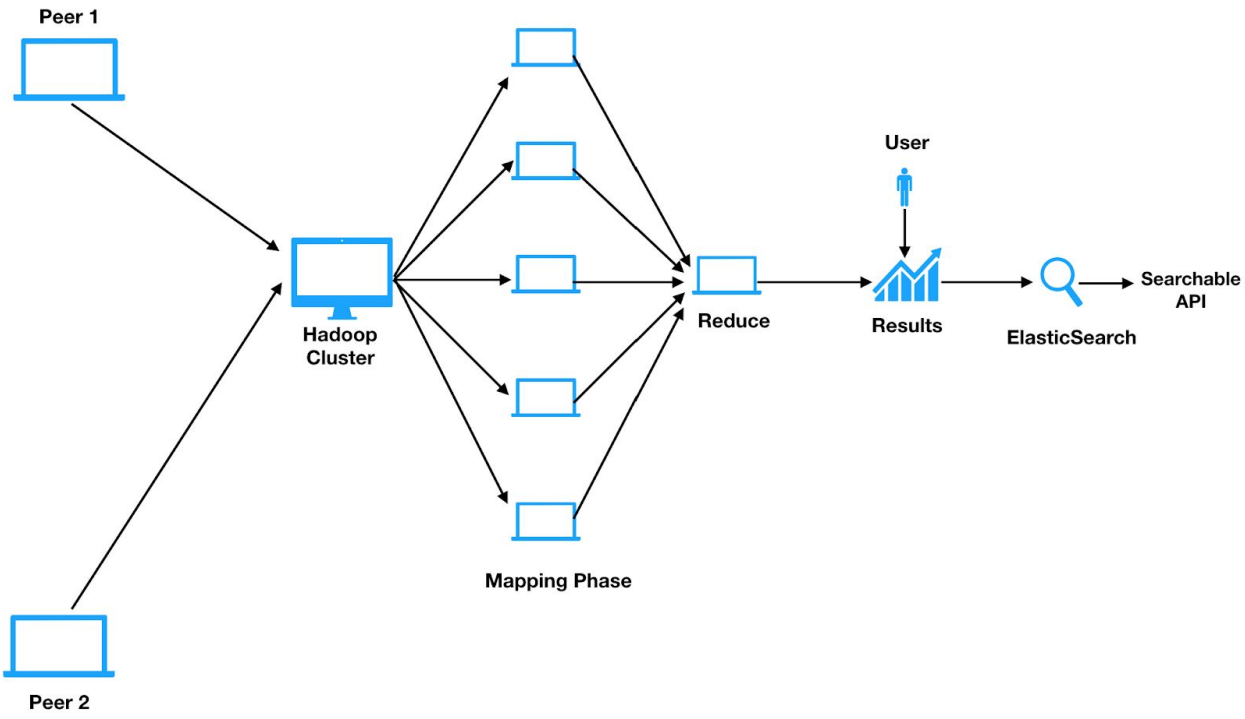
As hard as it is to upload the data, analyzing it becomes a major issue as well. Data is often compressed to a much smaller size than what they originally are. Due to hardware restrictions, often this data that cannot be extracted on local machines it is very difficult for users to envision the structure of the data.

The main advantage that APIs have over static data is that they are searchable, queryable and can be consumed in volume as specified by the user. However, these APIs are often paid and students cannot afford the rates of retrieving this data and need to look for static free alternatives.

Our project tries to solve both these issues. We plan to build a tool that will use peer to peer file transfer (torrents over a private network) to upload the files to an HDFS. Now as we have seen over large data sets that they are compressed with compression over 10 we will let the users upload compressed files and unzip them on the HDFS to reduce the network bandwidth. Once that is done we will

analyze the fields in the dataset to give users an overview of all the fields and also the distribution of the fields in there with map reduce jobs. With the results from that users can select which fields to index in Elasticsearch and hence reduce the overhead there. Therefore proving a searchable API for a static dataset.

High-Level Architecture



The Architecture diagram is as shown above.

The way we intend the architecture to work is as follows:

1. The compressed data files are present on both the peers. The data is uploaded to a compute instance using a peer to peer algorithm similar to that of bit torrent.
2. The data is unzipped here and pushed to the blob storage

3. MapReduce jobs run on the uncompressed data to determine analytics and information about the data that we would like to expose to the users.
    a. The format of the data: The various fields in the data that exist
    b. The count of the different possible values each of these fields can take
    c. The number of times the field is missing or is empty
4. The user has options to choose the fields that he wants to index or select all fields by default
5. The data is then pushed to ElasticSearch and the API is exposed to the user for queries.

Datasets

Since the problem we are not trying to solve is not a Machine Learning problem, we do not need “datasets” for this. However, we do have to validate the performance of the system we will be using various datasets, some of which were used for the assignments during the course.

Challenges

We will face various challenges during the course of our project.

1. The first step involves getting familiar with how peer to peer networks work. Peer to Peer networks involves distributed systems and learning new protocols for getting the system up and running. Since we have not worked with such frameworks before, we could run into hurdles along the way.
2. We intend to write bash scripts to uncompress the data on the cloud. Uncompression will involve writing bash scripts to take care of nested compressed files. Also, files can have various compression algorithms/formats used which need to be handled.
3. Transferring files to blob storage would be an extremely slow process, from what we have seen in the assignments. This needs to be done in a multi-threaded manner.

4. Determining structures in the data by exploring various parsers could be a complex issue as we see more datasets.
5. ElasticSearch is a cluster that is distributed as well, setting that up via a script could be complex.
6. Exposing data analysis to a user in an interactive manner is involves coming up with good UI designs.

Timeline

Sl No	Story Title	Story Description	Date
1	Project Proposal	Project report with Description, Timelines, Architecture and Challenges	30/10/2018
2	Set up a Private Peer to Peer network to upload data	Working on Building a tracker file and work on uploading the Compressed data files onto a Computer Node on AWS/Google Cloud	5/11/2018
3	Bucket Transfer	Detect when the entire data is uploaded onto the Compute Node, Uncompress the Data and Upload the data into an Blob storage	10/11/2018
4	Data Structure	Write MapReduce jobs to determine the structure of the data that is present in the uncompressed file.	15/11/2018
5	Perform Analytics	Write MapReduce jobs to present Analytics about the Dataset to the user	20/11/2018
6	Selection Framework	Provide a Framework to the User, where the user can choose which fields to select for indexing	25/11/2018
7	Spawn ES Cluster	Write a Script to create an ElasticSearch cluster to push data, based on the size of the uncompressed data that we have	30/11/2018
8	Push Data to ES	Write MapReduce jobs to Push Data to ElasticSearch such that the data is searchable through HTTP Protocols	10/12/2018