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.

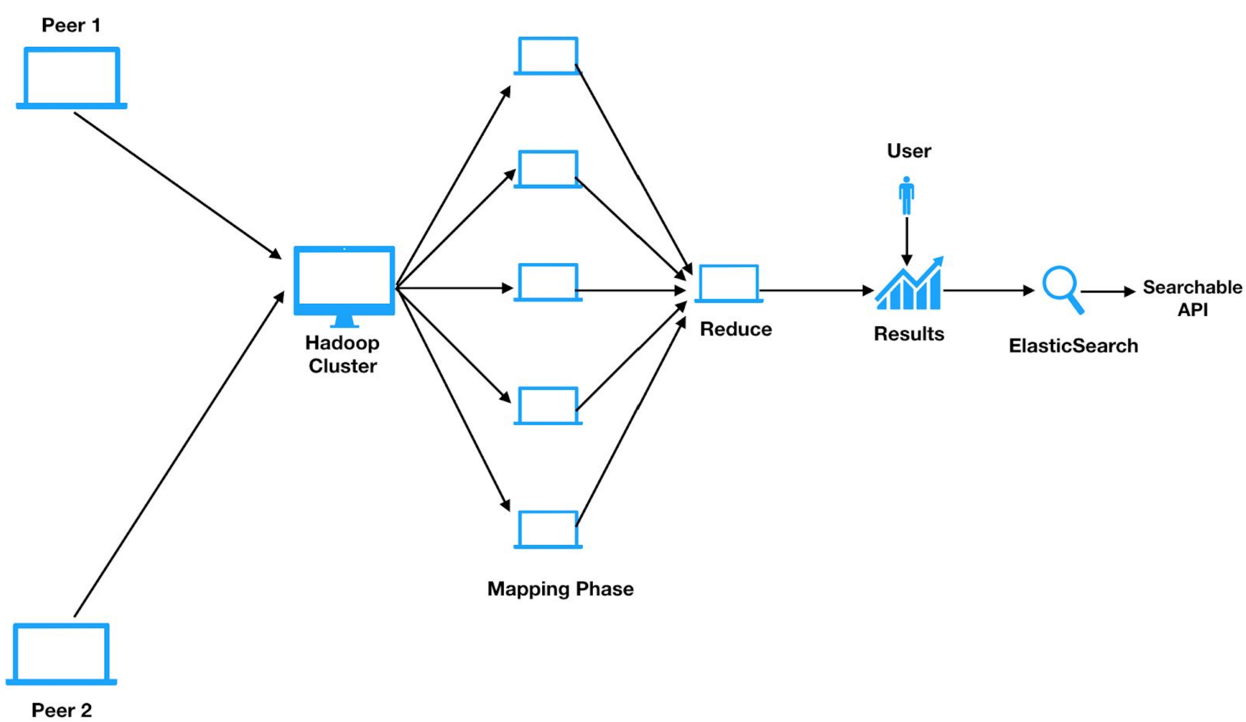
As hard as it is to upload the data, analyzing it becomes a major issue as well. Data is often compressed to a much smaller size than what they originally are. Due to hardware restrictions, often this data that cannot be extracted on local machines it is very difficult for users to envision the structure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

The section above is explained here:
The point above is highlighted again in the two sections below and explained with more context there. Please refer to the Datasets and challenges section.

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
For our test data we plan to use the Twitter monthly archive. It is a compressed dataset of 40GB and uncompressed is almost 400 GB. The fields are unstructured with over 2000+ unique fields over the dataset which are not structured, and a skewed distribution of data over the fields. We feel this dataset will be as challenging as it gets as it will test every aspect of our project.

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.
    a. The datasets that we intend to use is Twitter JSONs. Although this seems like structured data, it is semi-structured in our opinion. There are 2000+ unique fields and analytics on these are not going to be a trivial task. The datasets are also continuously changing due to this semi-structured nature. The aim of the project is for the analytics, summarization rather than parsing.
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

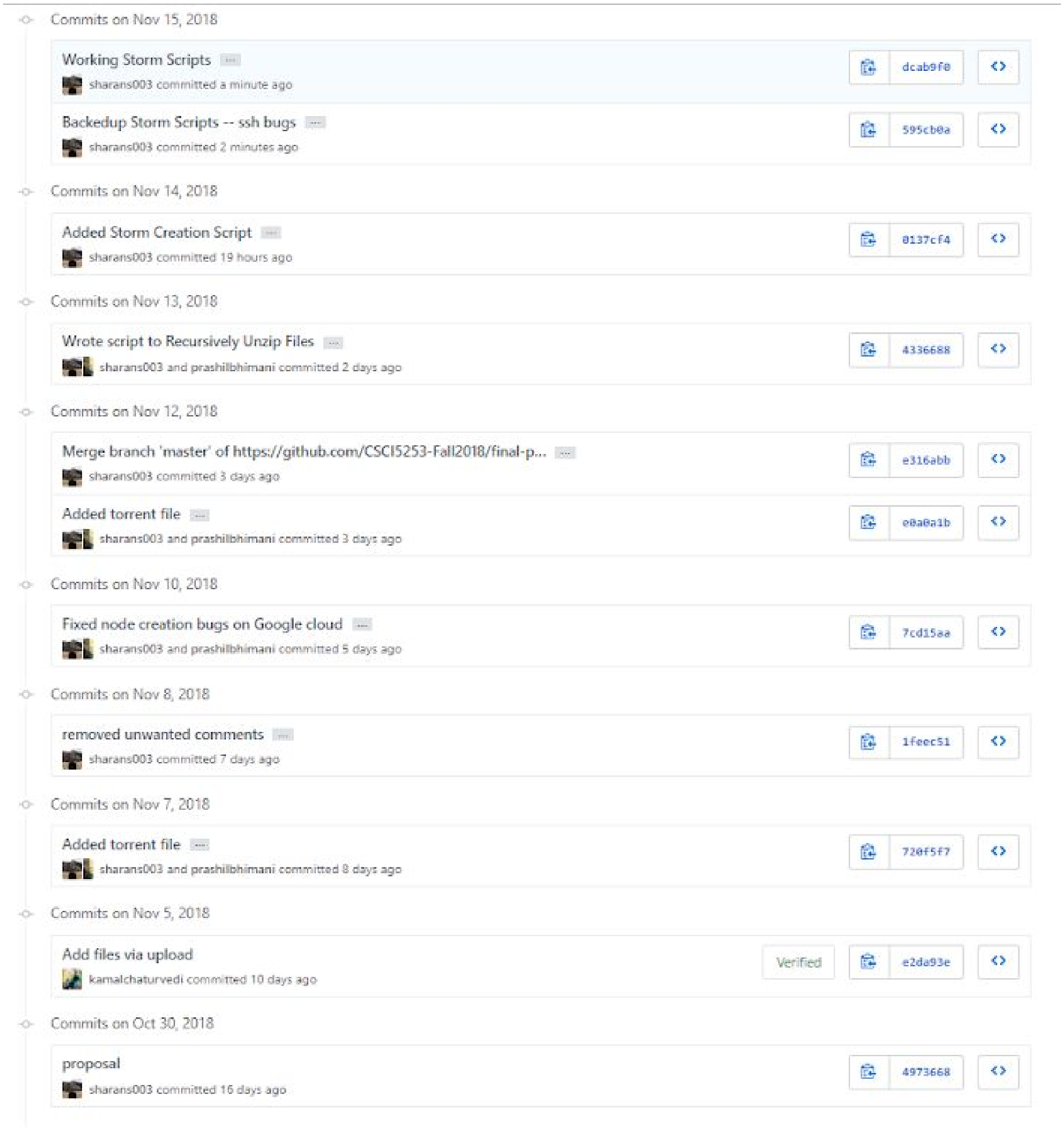
Sl No	Story Title	Story Description	Date
1	Project Proposal	Project report with Description, Timelines, Architecture and Challenges	30/10/2018
2	Set up a Private Peer to Peer network to upload data	Working on Building a tracker file and work on uploading the Compressed data files onto a Computer Node 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 a 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

Checkpoint 1 for Prashil & Sharan

Note : We worked on all stories together and so we have put up one table for this.

Sl No	Story Title	Story Description	Date	Status	Completed on
1	Project Proposal	Project report with Description, Timelines, Architecture and Challenges	30/10/2018	Completed	30/10/2018
2	Upload data using peer to peer networks	Worked on setting up torrent peer architecture on local and Google Cloud machines to download the data using a distributed network	5/11/2018	Completed	8/11/2018
3	Bucket Transfer	Detect when the entire data is uploaded onto the Compute Node, Uncompress the Data and Upload the data into a Blob storage	10/11/2018	Completed	13/11/2018
4	Data Structure	Write MapReduce jobs to determine the structure of the data that is present in the uncompressed file.	15/11/2018	Pending - Worked on scripts to create the distributed computing architecture and tested it with basic topologies	14/11/2018
5	Perform Analytics	Write MapReduce jobs to present Analytics about the Dataset to the user	20/11/2018	Not Started	
6	Selection Framework	Provide a Framework to the User, where the user can choose which fields to select for indexing	25/11/2018	Not started	
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	Not started	
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	Not started	

Github commits for Prashil & Sharan



Note: Due to some issue the coauthored part did not work for newer commit - But each commit message has the tag co-authored

Checkpoint 2 for Prashil & Sharan

There were no comments highlighted from the previous checkpoint. So nothing is highlighted below.

Note: We worked on all the stories together and so we have put up one table for this.

Sl No	Story Title	Story Description	Date	Status	Completed on
1	Project Proposal	Project report with Description, Timelines, Architecture and Challenges	30/10/2018	Completed	30/10/2018
2	Upload data using peer to peer networks	Worked on setting up torrent peer architecture on local and Google Cloud machines to download the data using a distributed network	5/11/2018	Completed	8/11/2018
3	Bucket Transfer	Detect when the entire data is uploaded onto the Compute Node, Uncompress the Data and Upload the data into a Blob storage	10/11/2018	Completed	13/11/2018
4	Data Structure	Write MapReduce jobs to determine the structure of the data that is present in the uncompressed file.	15/11/2018	Completed	20/11/2018
5	Perform Analytics	Write MapReduce jobs to present Analytics about the Dataset to the user	20/11/2018	Completed	30/11/2018
6	Selection Framework	Provide a Framework to the User, where the user can choose which fields to select for indexing	25/11/2018	Completed the terminal based framework for the user. Now working on the react-redux based framework as an incremental update. This work is not prioritized at the moment because we first want to get the whole pipeline working.	30/11/2018

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	Not started	
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	Not started	

Changes to Proposal

We do not have any changes to be made to the project proposal. We intend to implement the original action items.

Team meetings

Since this is a team of two, we regularly met for the major chunk of the project. Even during the break, we met on team viewer and Skype to discuss the action items and work on the implementation to stay true to the timeline.

Costs

The costs have been minimal. The reason for this is because we are using Google Cloud and we pause the instances when not in use.

The Storm development, the Hadoop, Kafka were done mainly on the local machines on much tinier mock datasets rather than consuming cloud resources when not needed. We have used less than 40$ from our Google credits for the project so far.

Dataset Management

The dataset is right now local to our machines in the compressed format (~40GB). These have been uploaded using our P2P logic and the unzip logic and stored in Google Storage for processing.
This section was more for Machine learning projects, but ours is purely a big data infrastructure project.

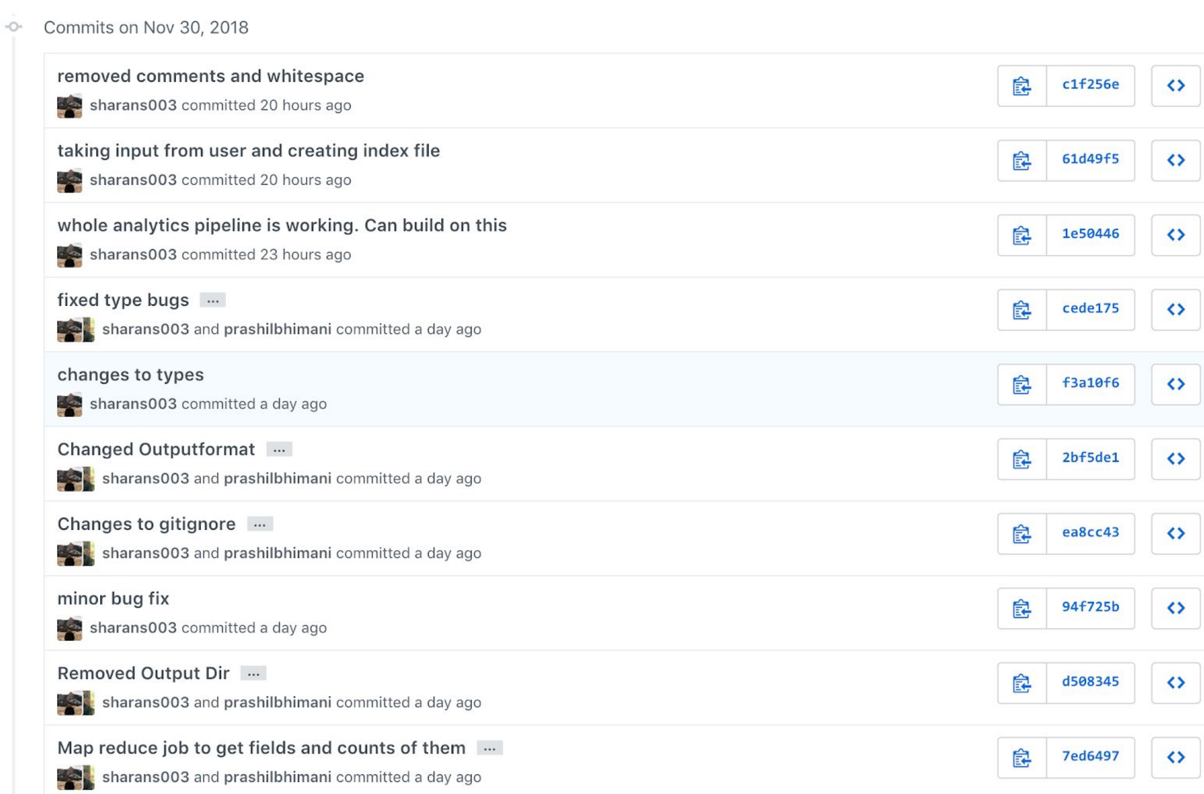
Challenges

We initially wanted to try our hand at Storm for the analytics. Storm was challenging to install and use. We started coding in Storm and realized that this was not the ideal technology for our analytics use case. Storm is used best for Streaming jobs, but since we had static data storm was not ideal and we had to pivot away from Storm to Hadoop. We will try to re-use our storm architecture maybe for the Elasticsearch push. However, this is yet to be decided.

We wanted to move away from Google's EMR for our map reduce jobs and set up a multi-node hadoop cluster on our own. The reason for stepping away was because, in the free tier, Google restricts the total cores in the Hadoop cluster to 8, which would mean that our analysis would take forever to run. Moreover, setting up Hadoop in a multinode cloud environment proved to be a learning experience.

Another challenge was setting up the front end and the server to ping one another. Getting React to work with Async Network requests is a new concept and took some understanding. Further, getting requests to work in Cross-Domain was always a painful concept to debug.

Github Commits For Prashil & Sharan



Commits on Nov 30, 2018

removed comments and whitespace
sharans003 committed 20 hours ago
c1f256e

taking input from user and creating index file
sharans003 committed 20 hours ago
61d49f5

whole analytics pipeline is working. Can build on this
sharans003 committed 23 hours ago
1e50446

fixed type bugs …
sharans003 and prashilbhimani committed a day ago
cede175

changes to types
sharans003 committed a day ago
f3a10f6

Changed Outputformat …
sharans003 and prashilbhimani committed a day ago
2bf5de1

Changes to gitignore …
sharans003 and prashilbhimani committed a day ago
ea8cc43

minor bug fix
sharans003 committed a day ago
94f725b

Removed Output Dir …
sharans003 and prashilbhimani committed a day ago
d508345

Map reduce job to get fields and counts of them …
sharans003 and prashilbhimani committed a day ago
7ed6497

Commits on Nov 23, 2018

Got data displaying
sharans003 committed 8 days ago
7986d8c

moved to a separate function
sharans003 committed 8 days ago
c33dc3f

passed analytics as props to checkbox
sharans003 committed 8 days ago
fc0f205

removed subreddit
sharans003 committed 8 days ago
fd0ad92

baseline. Things are working.
sharans003 committed 8 days ago
372fa3d

Commits on Nov 21, 2018

removed shit part 2
sharans003 committed 10 days ago
a21f3a1

removed shit part 1
sharans003 committed 10 days ago
ecd21fc

react router is also working now
sharans003 committed 10 days ago
6d364fe

got things working.
sharans003 committed 10 days ago
ad86b9e

Got rid of all errors