Unstack the Big Stack

1. Problem Definition

Stack Overflow is a very widely used question and answer forum for professional and enthusiastic programmers. A post can be a question, an answer to a question or comments on other posts. Posts contain various attributes like tags, upvotes, downvotes, views, etc. Users of Stack Overflow are encouraged to participate in the website to post quality questions and answers and are awarded reputation scores and badges. Such features help employers identify potential developers on the site for a particular technology.
Started in 2008, this popular site has huge amounts of data that need to be handled and processed at a continuous rate.
So, the objective of the project is to engineer this real-time data with streaming and do some analysis to visualize some of the interesting trends.

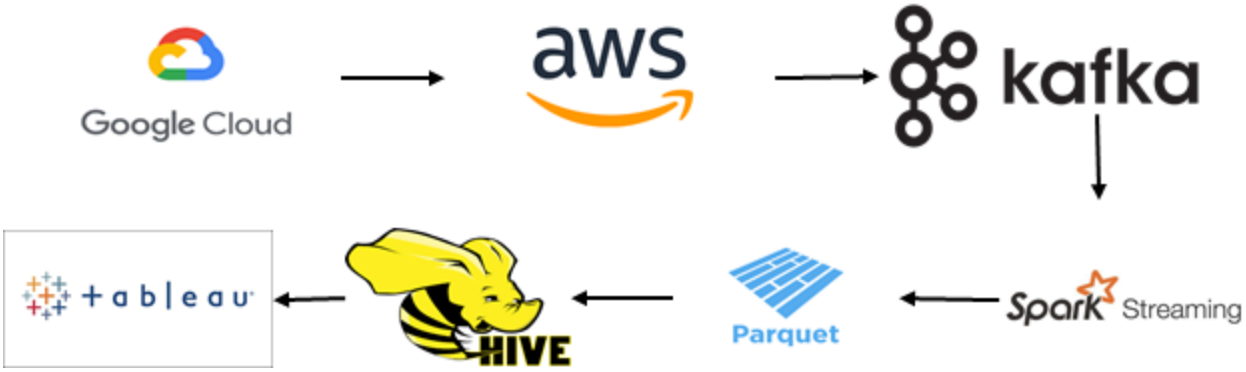
1.1 Challenges

Stack Overflow is a dynamically changing website, it is more meaningful to perform real-time analysis on its data. In performing the same, we need to ensure data consistency across the pipeline. The aggregated data after validation and cleaning should be made visible to the end-user to see results and perform actionable insights.
To summarize:

- Stack Overflow has a treasure trove of data and thus processing it in real-time is a challenging problem.
- Constructing a big data pipeline that ensures data consistency and availability.
- Preprocessing this data to remove unimportant data that do not give any useful analysis along with null values that create overhead.
- To build visualizations to get meaningful insights. We have to make sure that the interface is smooth and there is no performance overhead.

2. Methodology

To solve the above problems and overcome the challenges, we implemented the below pipeline:



2.1 Tech Stack Overview

BigQuery is Google's fully managed, petabyte-scale, low-cost analytics data warehouse.

AWS EMR - Amazon EMR is the industry-leading cloud-native big data platform for processing vast amounts of data quickly and cost-effectively at scale. Using open-source tools such as Apache Spark, Apache Hive, Apache HBase and Presto, coupled with the dynamic scalability of Amazon EC2 and scalable storage of Amazon S3, EMR gives analytical teams the engines and elasticity to run Petabyte-scale analysis for a fraction of the cost of traditional on-premise clusters.

AWS S3 - It is a public cloud storage resource available in Amazon Web Services' (AWS) Simple Storage Service (S3), an object storage offering. Amazon S3 buckets, which are similar to file folders, store objects, which consist of data and its descriptive metadata.

Pandas - It is an open-source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

Kafka - Apache Kafka is a distributed streaming platform. A streaming platform has three key capabilities. It is used for publishing and subscribing to streams of records, similar to a message queue or enterprise messaging system. Kafka is run as a cluster on one or more servers that can span multiple data centers. The Kafka cluster stores the stream of records in categories called topics.

Boto3 - It is the Amazon Web Services (AWS) SDK for Python. It enables Python developers to create, configure, and manage AWS services, such as EC2 and S3. Boto provides an easy to use, object-oriented API, as well as low-level access to AWS services.

Spark Streaming - It is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams. Data can be ingested from many sources like Kafka, Flume, Kinesis, or TCP sockets, and can be processed using complex algorithms expressed with high-level functions like map, reduce, join and window.

Parquet- It is a columnar storage format available to any project in the Hadoop ecosystem, regardless of the choice of data processing framework, data model or programming language.

Hive - It is data warehouse software that facilitates reading, writing, and managing large datasets residing in distributed storage using SQL.

Tableau - Tableau is a powerful and fastest-growing data visualization tool used in Business Intelligence. Data analysis is very fast with Tableau and the visualizations created are in the form of dashboards and worksheets.

2.2 Stages

Data Collection

Google cloud platform (GCP) hosts the StackOverflow data in form of tables which is available publicly. Using Google's BigQuery API, the data can be analyzed and can be extracted to do further analysis on top of it. Data fetched from tables in GCP was preprocessed and enriched data was pushed to the AWS S3 bucket in the form of a pipe separated format. Once the data was present in the bucket, the Boto3 package was then used to interact with data present in the S3 bucket.

Tables

- Posts_questions
- Post_history
- Users
- Tags
- Badges

ETL

We removed NULL values encountered in the data while loading it from GCP using the Big Query API. Columns containing fillers and non-ASCII values were cleaned.

Stream processing

Pandas Python library was used to read the pipe separated data and its feature of converting the big sized file to smaller chunks was utilized to convert the batch data to a streaming format. Once the data was available in streaming, Kafka producer published the data in the form of streams which was received in the form of Direct Stream using Consumer KafkaUtils.

Batch processing

We performed batch processing on all tables using AWS new feature called S3 SelectCSV.
S3 Select allows applications to retrieve only a subset of data from an object. For Amazon EMR, the computational work of filtering large data sets for processing is "pushed down" from the cluster to Amazon S3, which can improve performance in some applications and reduces the amount of data transferred between Amazon EMR and Amazon S3.

Storage

External Hive table was created on top of the Parquet file stored in the S3 bucket. This can store up to 5TB of data per object. A directory is created for different tables and streamed parquet files are stored in these directories.

- **Visualization and Analysis**
  We set up an input connection to Amazon EMR Hive in Tableau and load tables from the database named 'stackoverflow'. We performed joins and calculations in Tableau and our analysis is as follows:

- **User Analysis**
  This dashboard gives most active users in different locations across the world. We also show how many users joined Stack Overflow over the years starting from 2008.

- **Tags and Post Analysis**
  This dashboard depicts the most active regions across the world in terms of posts created in the form of questions, answers or comments. The dashboard has the capability wherein a user can select a particular tag and the most scored and most viewed question will appear. This will help users if they want to know the question related to a particular tag.

- **Questions Analysis**
  This dashboard gives analysis of top N questions where N is input from user. This results will be displayed as the top viewed, top scored and top answered questions in StackOverflow.

- **Badges Awarded**
  StackOverflow awards reputations points to the users if they helpful to the developers community in some way. . We found the top 3 users with the highest reputation and the badges they have received. Also, our dashboard is able to fetch the most popular badges and the class to which they belong. Class is categorized as Gold, Silver and Bronze.

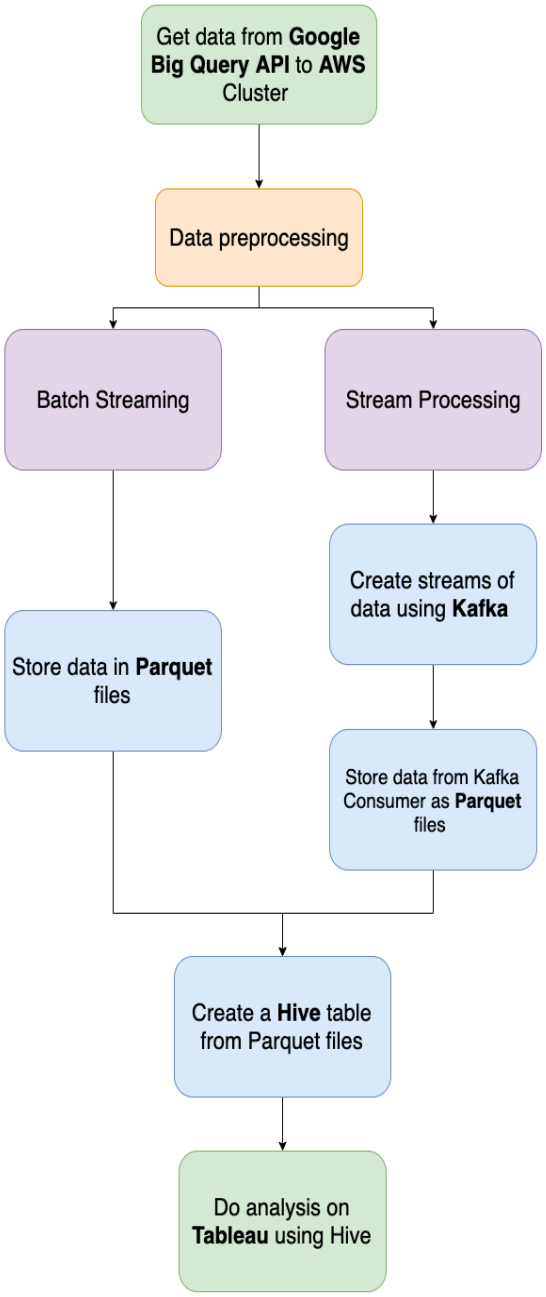
The visualization dashboard can be seen by clicking the link:
https://public.tableau.com/views/StackOverflow_Analysis/UserAnalysis?:display_count=y&publish=yes&:origin=viz_share_link

2.3 Optimizations employed in pipeline

- Parquet file format was used to store the data coming in as streams and batch. It provides columnar storage format. As the data increases cost for processing and storage increases. Parquet is the choice of Big data as it serves both needs, efficient and performance in both storage and processing.
- S3Select CSV: S3 Select allows applications to retrieve only a subset of data from an object. For Amazon EMR, the computational work of filtering large data sets for processing is "pushed

down" from the cluster to Amazon S3, which improves performance in applications and reduces the amount of data transferred between Amazon EMR and Amazon S3.

- Tableau Dashboard Optimization: We enabled extract refresh option in Tableau for faster processing of queries when user interacts with Tableau and this gives results within 1-2 seconds.



The StackOverflow data was present in the Google BigQuery platform. **Google API** was used to fetch data from Google Big Query to **AWS** S3 bucket in CSV format.

The data contained non-alphanumeric in multiple columns. So, filters were applied to obtain relevant data for analysis.

In order to simulate a real-time scenario, this batch data was then converted to streams.

Pandas chunks were used to send data in streams using **Kafka Streaming**.

The data sent by Kafka Producer was received in the form of direct streams using **KafkaUtils.**

The initial proposal of using HBase was discarded due to constraints in Python libraries supporting HBase. So **Hive** was chosen as the database having **Parquet** as the underlying file format to optimize query performance.

The analysis of the data is done using **Tableau** software.

3. Problems encountered and their Solutions

- **Huge data**
  One of the initial approaches to ingest the data was to use XML files on a local or SFU's gateway cluster. Reading huge XML file in the memory of the local system would have made the process really slow. Thus, the data was loaded using Google Big Query API into the AWS S3 bucket.
- **AWS limited credits**
  Due to limited AWS credits, the EMR cluster instance had to be terminated when not in use and recreated. So, a bootstrap script was created which installed the required libraries in the required EMR cluster on startup.
- **Choice of database**
  For storing the streamed data in the database, HBase was chosen. Due to the limited functionality provided in Python Hbase API, Hive which provides the functionality of a data Warehouse. Hive tables were created on top of S3 with Parquet as the underlying file format.
- **Spark Kafka Python API issues on EMR cluster**
  Kafka Spark Consumer had an issue when ran on AWS EMR cluster. So, Kafka Utils was used to stream the data published from Kafka Producer.

4. Results

Outcomes

- An end to end implementation and deployment of real time pipeline for stack overflow data.
- Processed and validated data flowing across the pipeline.
- Interactive dashboards for end user to view the analysis.

Learnings from data analysis

- India, Canada and the UK are the most active locations of users on Stack Overflow.
- We are able to fetch the most popular question and most scored question for a particular tag or a technology.
- We can identify most reputed users and what is the technology they are working on, to reach out to them easily later on. This can be useful for employers to find skilled developers from this site and recruit them.

Learnings from implementation

- We learnt how to effectively stream data using big data technologies like Kafka and Spark Streaming.
- Creating AWS EMR Cluster and deploying big data pipeline on top of it.
- We learnt to bootstrap the EMR cluster. Bootstrapping will install the required libraries and make the cluster ready to run required big data technologies.
- We learnt how to solve a real time streaming problem and ensure data availability and consistency at various stages in Big Data pipeline.

5. References

https://aws.amazon.com/
https://cloud.google.com/bigquery/public-data/
https://www.kaggle.com/stackoverflow/stackoverflow
https://meta.stackexchange.com/questions/2677/database-schema-documentation-for-the-public-data-dump-and-sede

6. Project Summary

Getting the data	2
ETL	2
Problem	3
Algorithmic work	0
Bigness/parallelization	4
UI	1
Visualization	3
Technologies	5
Total	20