**CSP-554-BIG DATA TECHNOLOGIES**

**------------------------------------------------------------------------------**

**PROJECT PROPOSAL**

**------------------------------------------------------------------------------**

**BIG DATA PROCESSING PIPELINE**

**MARCH 25, 2021**

|  |  |
| --- | --- |
| **YASH PATEL** | **A20451170** |
| **HARSH VORA** | **A20445400** |
| **VISHNU BHARATH** | **A20465596** |
| **VARUN VEERLA** | **A20458191** |

**ILLINOIS INSTITUTE OF TECHNOLOGY**

**PROF. JOSEPH ROSEN**

**INTRODUCTION**

**PROBLEM STATEMENT**

The stream API on Twitter allows you to receive approximately 50 tweets per second. However, this figure must be even higher. Handling, processing, and analyzing this massive volume of real-time data upon its arrival in order to gain information without exceeding the time allotted for decision making or an analytical procedure.

**PROPOSED SOLUTION**

A big data processing pipeline is proposed as a workaround. To collect real-time data, also known as event streaming, we will use Apache Kafka as the first portion of the pipeline, which offers a coherent, high-throughput, and low-latency solution. The performance of Apache Kafka will be absorbed as the middle portion of the pipeline for real-time stream data processing into the Apache Spark distributed processing system, which provides data parallelism and fault tolerance. To store vast volumes of processed real-time data, we can use Google Firebase Realtime Database as the last portion of the pipeline, which is a NoSQL database that allows you to store, sync, and query data between users in real-time. We will stream this real-time data to the HTML web-page client for visualization using the firebase kit in the Node.js server.

**PROJECT GOALS:**

* Ingest data using Twitter’s streaming API.
* Capture data using Apache Kafka.
* Process streaming data using Apache Spark.
* Store these processed data using Google Firebase.
* Visualize these processed data using Node.js server and HTML web-page client.

**BIG DATA TECHNOLOGIES:** Kafka, Spark, Firebase

**OTHER TECHNOLOGIES:** Node.js, HTML

**REFERENCES**

The list of sources below is a recommended reading list. The knowledge collection mechanism may determine whether or not a reference is eventually relevant to this program, and to what capacity/extent. As a result, all references listed below may or may not be referenced. Additional sources may also be applied during the review process.

|  |  |
| --- | --- |
| **[1]** | [**https://developer.twitter.com/en/docs/twitter-api/tweets/filtered-stream/introduction**](https://developer.twitter.com/en/docs/twitter-api/tweets/filtered-stream/introduction) |
| **[2]** | [**https://developer.twitter.com/en/docs/twitter-api/tweets/sampled-stream/introduction**](https://developer.twitter.com/en/docs/twitter-api/tweets/sampled-stream/introduction) |
| **[3]** | [**https://dzone.com/articles/running-apache-kafka-on-windows-os**](https://dzone.com/articles/running-apache-kafka-on-windows-os) |
| **[4]** | [**https://phoenixnap.com/kb/install-spark-on-windows-10**](https://phoenixnap.com/kb/install-spark-on-windows-10) |
| **[5]** | [**https://firebase.google.com/docs/database**](https://firebase.google.com/docs/database) |
| **[6]** | [**https://pypi.org/project/firebase/**](https://pypi.org/project/firebase/) |
| **[7]** | [**https://www.npmjs.com/package/firebase**](https://www.npmjs.com/package/firebase) |