Scalable Atmospheric Data Pipeline for Global Tropical Cyclone Intensity Forecasting

Bryan Julius
CS 4265: Big Data Analytics
Milestone 1: Project Proposal
https://github.com/Bryan-Julius/Big-Data-Processing-Project

I. Project Overview

A. Domain

This project falls within the domain of **Meteorology and Climate Science**, specifically focusing on the computational challenges of high-resolution atmospheric modeling.

B. Problem Statement

Accurate hurricane intensity forecasting requires processing petabytes of high-resolution satellite imagery and atmospheric sensor data. Current bottlenecks in forecasting arise from the inability to ingest, clean, and synchronize disparate data sources (SST, wind shear, moisture) at a scale that allows for modern Transformer-based training without significant latency. This system aims to solve the scalability gap between raw atmospheric data and model-ready tensors.

C. Scope

The scope includes building a distributed ETL pipeline for satellite data using HDFS and Apache Spark. The project will implement a wide column data store using HBase for efficient range scan queries of storm windows. The scope is limited to data engineering and infrastructure; hyperparameter tuning of the forecasting model is excluded.

II. System Description

A. Data Sources and Characteristics

- **Sources:** NOAA GOES-R satellite imagery (via S3) and HURDAT2 historical tracks.
- **Volume:** Terabytes of historical imagery requiring distributed storage.
- **Variety:** Unstructured NetCDF imagery and structured sensor CSVs.
- **Velocity:** Batch processing of historical data with a simulated streaming layer for real-time inference.

B. Stack Layers and Course Concepts

The system engages three primary layers of the Big Data stack:

1) **Storage (HDFS):** Utilizing HDFS blocks and a replication factor of 3 to ensure fault tolerance and parallel read throughput for large image files.
2) **Processing (Spark):** Utilizing Spark DataFrames for distributed normalization and Map/Reduce phases to aggregate atmospheric features by storm ID.
3) **Data Stores (HBase):** Implementing a wide-column model where regions are partitioned by storm timestamps for fast retrieval.

III. Implementation Approach

A. Technology Choices

The stack will utilize HDFS for storage, Parquet for encoding, Spark for processing, and HBase for the data store.

B. Scalability and Metrics

Horizontal scaling will be achieved by adding DataNodes to the HDFS cluster. I will measure throughput (images processed per second) and latency (time from raw ingestion to model-ready query).

IV. Literature Review

The design is informed by the following foundational concepts:

- **GFS/HDFS Architecture:** Understanding distributed file systems.
- **Spark RDD Abstraction:** Leveraging lazy evaluation for iterative feature engineering.
- **Wide-Column Design:** Utilizing HBase for structured atmospheric data.
- **MapReduce Model:** For initial parallel data cleaning.
- **The Big Data Textbook:** Guidance on schema validation and data models.

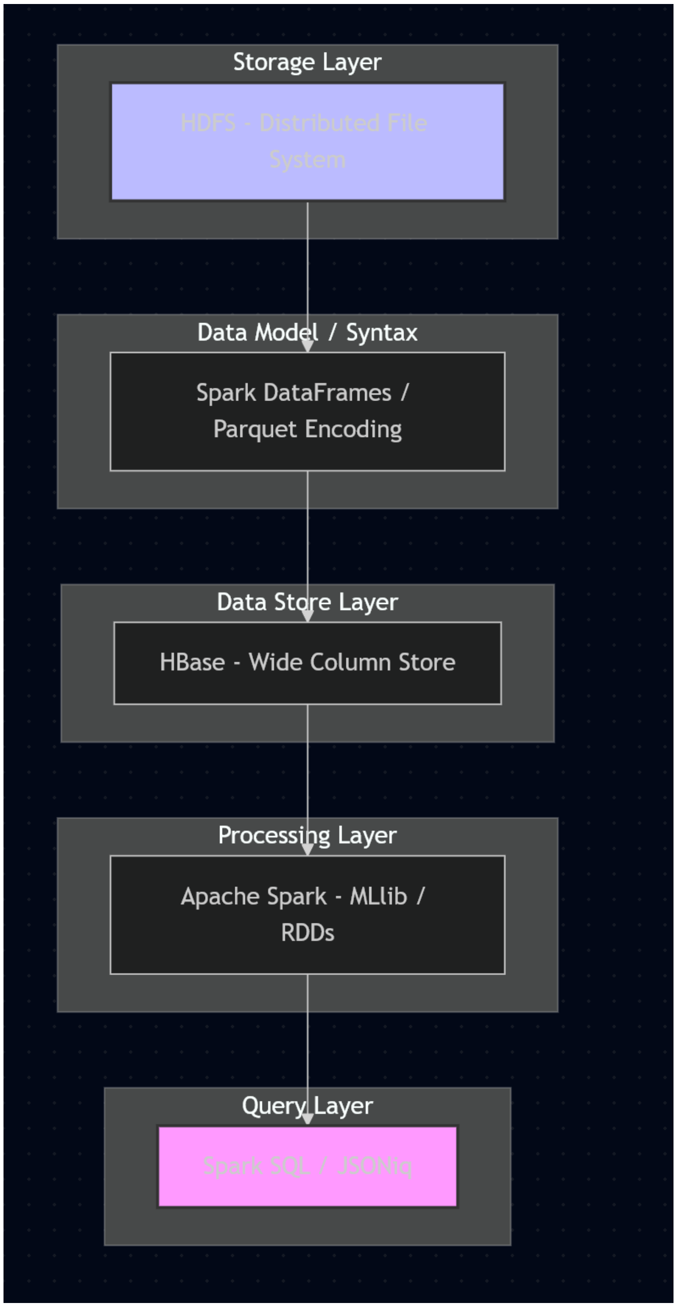

Fig. 1. Stack Architecture Diagram

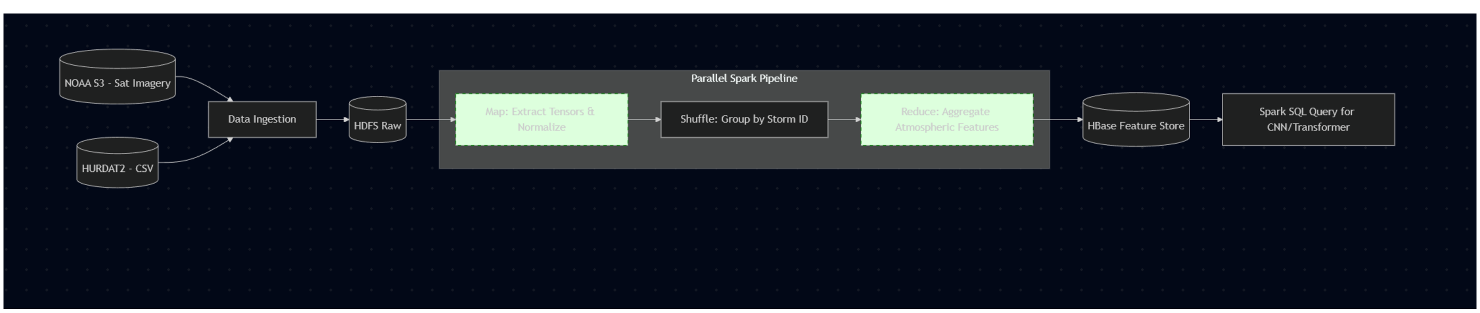

Fig. 2. Data Flow Diagram