Result Management System

Parth Sarthi Singh

February 28, 2025

Abstract

This report describes the implementation of a Result Management System using Apache Spark and Google Drive as a storage alternative to HDFS. The system generates and analyzes marks for 10,000 students across six subjects, stores the data in Google Drive, and presents statistical insights using Python and Matplotlib.

1 Introduction

With the increasing number of students in universities, managing and analyzing student results efficiently has become crucial. This project leverages Apache Spark for distributed processing and Google Drive as a storage solution in Google Colab.

2 Technology Stack

The following technologies are used:

- Apache Spark (PySpark) for distributed data processing
- Google Colab for cloud-based execution
- Google Drive for data storage
- Python (Pandas, Matplotlib) for visualization

3 Implementation

3.1 Student Data Generation

A dataset of 10,000 students is created with random ages and marks for six subjects:

- Electronics
- Programming

- Database
- Data Science
- Mathematics
- Data Structures and Algorithms (DSA)

Each student is assigned a unique ID and name.

3.2 Data Storage in Google Drive

Since HDFS is not available in Google Colab, Google Drive is used to store student data and analysis results. The dataset is saved in CSV format in Google Drive, ensuring persistent storage across sessions.

3.3 Data Processing with PySpark

The stored CSV data is loaded back into a Spark DataFrame for statistical analysis. The following metrics are computed:

- Average marks for each subject
- Maximum and minimum marks
- Standard deviation for performance variability

3.4 Visualization

Matplotlib is used to visualize the computed statistics as bar charts, providing insights into student performance distribution.

4 Code Implementation

```
# Install dependencies
!pip install pyspark

# Import libraries
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, mean, max, min, stddev
import random
import matplotlib.pyplot as plt
import pandas as pd
import os

# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Define paths
path = "/content/drive/MyDrive/ResultManagement"
```

```
os.makedirs(path, exist_ok=True)
students_path = f"{path}/students.csv"
results_path = f"{path}/results.csv"

# Initialize Spark
spark = SparkSession.builder.appName("ResultManagement").getOrCreate()

# Generate student data
num_students = 10000
subjects = ["Electronics", "Programming", "Database", "DataScience", "Mathematics", "DSA"]
students = [(i, f"Student_{i}", random.randint(18, 25), *(random.randint(40, 100) for _ in sub
columns = ["StudentID", "Name", "Age"] + subjects
df = spark.createDataFrame(students, columns)

# Save and reload data
df.toPandas().to_csv(students_path, index=False)
df_drive = spark.read.csv(students_path, header=True, inferSchema=True)

# Compute statistics
stats = df_drive.agg(*[mean(col(s)).alias(f"Avg_{s}") for s in subjects],
                     *[max(col(s)).alias(f"Max_{s}") for s in subjects],
                     *[min(col(s)).alias(f"Min_{s}") for s in subjects],
                     *[stddev(col(s)).alias(f"Std_{s}") for s in subjects])

# Save results
stats.toPandas().to_csv(results_path, index=False)

# Visualization
pdf = stats.toPandas().T
pdf.columns = ["Value"]
plt.figure(figsize=(12,6))
plt.bar(pdf.index, pdf["Value"], color='steelblue', edgecolor='black')
plt.xticks(rotation=45, ha='right')
plt.title("Subject-wise␣Statistics")
plt.ylabel("Values")
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.show()

# Stop Spark
spark.stop()
```

5 Results and Observations

The computed statistics provide insights into student performance. The visualization helps in understanding subject-wise trends, identifying challenging subjects, and planning interventions.

6 Conclusion

This project demonstrates how Apache Spark can efficiently process large student datasets while leveraging Google Drive for storage in Colab. Future im-

provements can include advanced predictive analytics and integrating a web-based dashboard.

7 References

- Apache Spark Documentation: `https://spark.apache.org/docs/latest/`
- Google Colab Documentation: `https://colab.research.google.com/`