Apache Hive: A Petabyte-Scale Data Warehouse System Over a MapReduce Framework

Test Application: MBV Climate and Ocean Intelligence Africa

Dushime Mudahera Richard

Databases for Big Data Seminar – Prof. Iztok Savnik

January 2026

Agenda & Executive Summary

Outline

1. System Architecture
2. Query Optimization
3. Experimental Results
4. Challenges & Solutions
5. Conclusions

Test Data

- 4.75 million climate records
- 304 MB raw CSV data
- 4 Hive tables (1980–2024)
- 5 African regions, 44 countries

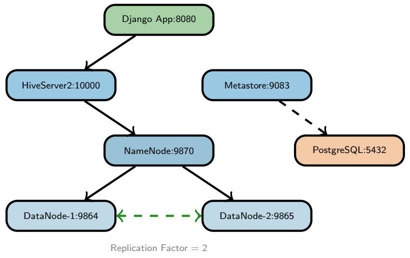
Project Overview

Production-grade **distributed big data ecosystem** simulating climate analytics

- 7-Container Docker Stack – Hadoop 3 + Hive 2.3.2
- Django REST API – Climate dashboard & benchmarking
- 3-Node HDFS Cluster – Distributed storage with replication

Goal: Demonstrate Apache Hive's capability for petabyte-scale analytics on commodity hardware

7-Container Stack Architecture



Container Services

- **Django** – REST API, Web Dashboard
- **HiveServer2** – JDBC/Thrift gateway
- **Metastore** – Schema catalog (decoupled)
- **NameNode** – HDFS namespace manager
- **DataNodes** – Distributed block storage
- **PostgreSQL** – Persistent metadata DB

HDFS Cluster Stats

- Configured Capacity: 416.56 GB
- Live DataNodes: 2 (healthy)
- Under-replicated blocks: 0

Apache Hive & Query Optimization

Hive Architecture

- **HiveServer2** – JDBC/ODBC gateway
- **Metastore** – Centralized schema catalog
- **Execution Engine** – MapReduce backend
- **HDFS Storage** – Distributed file system

Cost-Based Optimizer (CBO)

- Uses Apache Calcite for query planning
- Estimates costs from table statistics
- Selects optimal join algorithms
- Requires `ANALYZE TABLE`

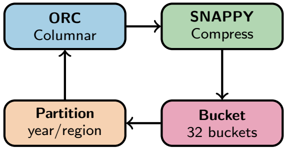
Join Algorithms

- **Shuffle Join** (Reduce-Side)
  - Default, requires data shuffle
  - High network I/O cost
- **Broadcast Join** (Map-Side)
  - Small table broadcast to all mappers
  - **3x faster** for asymmetric joins
- **Sort-Merge-Bucket Join**
  - Most efficient for pre-bucketed data

Key Configuration

```
SET hive.auto.convert.join=true;
```

Storage Optimizations & Performance



Optimization Techniques

- **ORC Format** – Columnar storage, predicate pushdown, reduced I/O
- **SNAPPY** – Fast compression, CPU-efficient
- **Partitioning** – Skip 90%+ data via partition pruning
- **Bucketing** – Enable Map-Side joins, faster GROUP BY

Compression Results

- ORC: **88%** size reduction
- vs. raw CSV (304 MB $\rightarrow$ 36 MB)

Vectorized Execution

- Process 1,024 rows per CPU instruction
- `SET hive.vectorized.execution.enabled=true;`
- Significant speedup for STDDEV, CORR, AVG

Query Performance Benchmarks

Query Execution Times (4.75M rows)

Query Type	**Time (s)**
Simple Regional Aggregation	**9.31**
Complex Monthly Aggregation	14.88
Statistical Analysis	16.13
Yearly Analysis	15.24
Map-Side Join	15.30
Reduce-Side Join	42.70
Data Coverage Query	13.37

Regional Temperature Results

Region	**Avg °C**	
Central	30.49	2.448
West	29.51	2.448
East	26.51	2.448
North	24.51	2.450
South	20.50	2.448

Join Algorithm Comparison

- Map-Side: **15.30s** (2.8x faster)
- Reduce-Side: 42.70s (baseline)

Key Findings

- 10°C gradient Central→South
- Uniform variance (2.45)
- 2-stage MapReduce for ORDER BY
- 28–35 GB HDFS reads per query

Engineering Challenges & Solutions

Challenge 1: ARM Emulation

- Apple Silicon runs x86 images
- 10–20% performance overhead
- 60–120s container startup

Solution

- Extended health check timeouts
- Optimized JVM heap settings
- Rosetta 2 emulation layer

Challenge 2: PostgreSQL Driver

- Hive 2.3.2 JDBC compatibility
- MD5 auth required (PG 9.6)
- Metastore schema init

Solution

- Manual JAR injection
- `postgresql-42.7.2.jar`
- Docker volume mounts

Summary & Key Takeaways

Achievements

- **Infrastructure**
  - 7-container Docker stack
  - Hive 2.3.2 + Hadoop 3
  - 447 GB HDFS capacity
- **Data Processing**
  - 4.75M records (45 years)
  - 44 countries, 5,000 stations
  - Multi-stage MapReduce
- **Performance**
  - 2.8x speedup with Map-Side joins
  - 88% compression with ORC
  - 9–26s query latency

Key Findings

1. ✓ CBO requires fresh statistics
2. ✓ Join algorithm choice is critical
3. ✓ ORC format essential for analytics

Conclusion

Apache Hive provides cost-effective, petabyte-scale analytics on commodity hardware—validated by Netflix, Facebook, Airbnb processing 100+ PB daily.

Thank You!

Questions?

Dushime Mudahera Richard

MBV Climate and Ocean Intelligence Africa

Databases for Big Data Seminar – Prof. Iztok Savnik

University of Primorska — January 2026