

## Assignment 2: Data Storage

### Batch Analytics Pipeline on HDFS & Hive

#### 1. Introduction

MediaCo collects large daily logs of user activity from a streaming platform. The objective of this project is to design a batch analytics pipeline using HDFS for data storage and Hive for querying. The pipeline performs data ingestion, transformation, and analysis to extract meaningful insights.

#### 2. Data Description

The dataset includes:

- **User Logs:** Contains user interactions such as plays, pauses, and skips.
  - Format: CSV or JSON
  - Columns: user\_id, content\_id, action, timestamp, device, region, session\_id
  - Stored in /raw/logs/YYYY/MM/DD/
- **Content Metadata:** Static reference data about content.
  - Columns: content\_id, title, category, length, artist
  - Stored in /raw/metadata/

#### 3. Data Ingestion

The ingestion process involves:

1. Running generate\_data.py to generate synthetic logs and metadata.
2. Executing ingest\_logs.sh to move data to HDFS.
3. Using move\_metadata.sh to relocate metadata files.

#### Shell Script: ingest\_logs.sh

This script:

- Accepts a date parameter.
- Parses year, month, and day.
- Moves logs to /raw/logs/YYYY/MM/DD/ in HDFS.
- Moves metadata to /raw/metadata/.

./ingest\_logs.sh YYYY-MM-DD

## 4. Hive Schema

### Raw Tables

```
CREATE EXTERNAL TABLE raw_logs (  
    user_id INT,  
    content_id INT,  
    action STRING,  
    timestamp STRING,  
    device STRING,  
    region STRING,  
    session_id STRING  
) PARTITIONED BY (year INT, month INT, day INT)  
STORED AS TEXTFILE LOCATION '/raw/logs';
```

### Star Schema

- **Fact Table:** fact\_user\_actions (stores user interactions, partitioned by date).
- **Dimension Table:** dim\_content (stores content metadata).

```
CREATE TABLE fact_user_actions (  
    user_id INT,  
    content_id INT,  
    action STRING,  
    timestamp TIMESTAMP,  
    device STRING,  
    region STRING,  
    session_id STRING  
) PARTITIONED BY (year INT, month INT, day INT)  
STORED AS PARQUET;
```

## 5. Data Transformation

Data is moved from raw tables to the star schema using Hive SQL:

```
INSERT OVERWRITE TABLE fact_user_actions PARTITION (year, month, day)
```

```
SELECT user_id, content_id, action,
```

```
CAST(timestamp AS TIMESTAMP), device, region, session_id,
```

```
year(timestamp), month(timestamp), day(timestamp)
```

```
FROM raw_logs;
```

## 6. Analytical Queries

### Query 1: Monthly Active Users by Region

```
SELECT year, month, region, COUNT(DISTINCT user_id) AS active_users
```

```
FROM fact_user_actions
```

```
GROUP BY year, month, region
```

```
hanzi@DESKTOP-FC02M5Q: ~/hive/conf
hive> SELECT
  year,
  mon
  > th,
  > region,
  COUNT
  > (DISTINCT user_id) AS active_users
FROM fact_user_actions
G
> ROUN BY ye
> ar, month, region
ORDER BY
  > year DESC, month DESC;
Query ID = hanzi_20250306173404_6db59c0f-a474-4776-9d75-c281e55fc455
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reducers=<number>
Starting Job = job_1741254545702_0017, Tracking URL = http://DESKTOP-FC02M5Q.localdomain:8088/proxy/application_1741254545702_0017/
Kill Command = /home/hanzi/hadoop/bin/mapred job -kill job_1741254545702_0017
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2025-03-06 17:34:15,970 Stage-1 map = 0%, reduce = 0%
2025-03-06 17:34:23,307 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.87 sec
2025-03-06 17:34:29,578 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.49 sec
MapReduce Total cumulative CPU time: 5 seconds 490 msec
Ended Job = job_1741254545702_0017
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reducers=<number>
Starting Job = job_1741254545702_0018, Tracking URL = http://DESKTOP-FC02M5Q.localdomain:8088/proxy/application_1741254545702_0018/
Kill Command = /home/hanzi/hadoop/bin/mapred job -kill job_1741254545702_0018
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2025-03-06 17:34:46,535 Stage-2 map = 0%, reduce = 0%
2025-03-06 17:34:53,931 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 2.83 sec
2025-03-06 17:35:01,262 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 6.41 sec
MapReduce Total cumulative CPU time: 6 seconds 410 msec
Ended Job = job_1741254545702_0018
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 5.49 sec HDFS Read: 11859 HDFS Write: 173 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 6.41 sec HDFS Read: 8837 HDFS Write: 161 SUCCESS
Total MapReduce CPU Time Spent: 11 seconds 900 msec
OK
2025 3 US 9
2025 3 EU 9
2025 3 APAC 7
Time taken: 57.872 seconds, Fetched: 3 row(s)
```

### Query 2: Top Content Categories by Play Count

```
SELECT c.category, COUNT(*) AS play_count
```

```

FROM fact_user_actions f

JOIN dim_content c ON f.content_id = c.content_id

WHERE action = 'play'

GROUP BY c.category

ORDER BY play_count DESC;

```

```

Select hanzi@DESKTOP-FC02M5Q: ~/hive/conf
hive> SELECT
  > c.category,
  COU > NT(*) AS play_count
FROM fact_user > _actions f
JOIN dim > _content c
ON f.cont > ent_id = c.content_id
  > WHERE f.action = 'play'
GR > OUP BY c.category
  > ORDER BY play_count DESC
LIMIT 5;
  > Query ID = hanzi_20250306173625_e41bd462-e51b-457c-8fe6-ec7eaafe18f4
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1741254545702_0019, Tracking URL = http://DESKTOP-FC02M5Q.localdomain:8088/proxy/application_1741254545702_0019/
Kill Command = /home/hanzi/hadoop/bin/mapred job -kill job_1741254545702_0019
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2025-03-06 17:36:37,008 Stage-1 map = 0%, reduce = 0%
2025-03-06 17:36:47,874 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 13.59 sec
2025-03-06 17:36:55,340 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 16.76 sec
MapReduce Total cumulative CPU time: 16 seconds 760 msec
Ended Job = job_1741254545702_0019
Launching Job 2 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1741254545702_0020, Tracking URL = http://DESKTOP-FC02M5Q.localdomain:8088/proxy/application_1741254545702_0020/
Kill Command = /home/hanzi/hadoop/bin/mapred job -kill job_1741254545702_0020
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2025-03-06 17:37:12,386 Stage-2 map = 0%, reduce = 0%
2025-03-06 17:37:19,717 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 2.73 sec
2025-03-06 17:37:25,909 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 5.63 sec
MapReduce Total cumulative CPU time: 5 seconds 630 msec
Ended Job = job_1741254545702_0020
Launching Job 3 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1741254545702_0021, Tracking URL = http://DESKTOP-FC02M5Q.localdomain:8088/proxy/application_1741254545702_0021/
Kill Command = /home/hanzi/hadoop/bin/mapred job -kill job_1741254545702_0021
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1
2025-03-06 17:37:43,830 Stage-3 map = 0%, reduce = 0%
2025-03-06 17:37:51,182 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 2.61 sec
2025-03-06 17:37:58,554 Stage-3 map = 100%, reduce = 100%, Cumulative CPU 5.77 sec
MapReduce Total cumulative CPU time: 5 seconds 770 msec
Ended Job = job_1741254545702_0021
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2 Reduce: 1 Cumulative CPU: 16.76 sec HDFS Read: 21605 HDFS Write: 165 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 5.63 sec HDFS Read: 6809 HDFS Write: 165 SUCCESS
Stage-Stage-3: Map: 1 Reduce: 1 Cumulative CPU: 5.77 sec HDFS Read: 7763 HDFS Write: 144 SUCCESS
Total MapReduce CPU Time Spent: 28 seconds 160 msec
OK
Jazz 4
Rock 3
News 1
Time taken: 93.847 seconds, Fetched: 3 row(s)

```

### Query 3: Average Session Length per Week

```

SELECT year, weekofyear(timestamp) AS week, AVG(length) AS avg_session_length

FROM fact_user_actions

JOIN dim_content ON fact_user_actions.content_id = dim_content.content_id

```

GROUP BY year, weekofyear(timestamp);

```
Select hanzi@DESKTOP-FC02M5Q: ~/hive/conf
hive> SELECT
  > year,
  weekofyear(
    > event_time) AS week,
  AVG(1
    > length) AS avg_session_length
FROM fact_user a
  > ctions f
JOIN dim_c
  > ontent c
ON f.content
  > _id = c.content_id
GROUP BY yea
  > r, weekofyear(event_time)
ORDER BY yea
  > r DESC, week DESC;
Query ID = hanzi_20250306172454_c4ef1745-9ea7-489d-83a1-326254997ad4
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1741254545702_0014, Tracking URL = http://DESKTOP-FC02M5Q.localdomain:8088/proxy/application_1741254545702_0014/
Kill Command = /home/hanzi/hadoop/bin/mapred job -kill job_1741254545702_0014
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2025-03-06 17:25:05,396 Stage-1 map = 0%, reduce = 0%
2025-03-06 17:25:15,114 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 9.38 sec
2025-03-06 17:25:22,482 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 13.66 sec
MapReduce Total cumulative CPU time: 13 seconds 660 msec
Ended Job = job_1741254545702_0014
Launching Job 2 out of 3
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1741254545702_0015, Tracking URL = http://DESKTOP-FC02M5Q.localdomain:8088/proxy/application_1741254545702_0015/
Kill Command = /home/hanzi/hadoop/bin/mapred job -kill job_1741254545702_0015
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2025-03-06 17:25:39,188 Stage-2 map = 0%, reduce = 0%
2025-03-06 17:25:46,607 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 2.51 sec
2025-03-06 17:25:53,925 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 6.8 sec
MapReduce Total cumulative CPU time: 6 seconds 800 msec
Ended Job = job_1741254545702_0015
Launching Job 3 out of 3
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1741254545702_0016, Tracking URL = http://DESKTOP-FC02M5Q.localdomain:8088/proxy/application_1741254545702_0016/
Kill Command = /home/hanzi/hadoop/bin/mapred job -kill job_1741254545702_0016
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1
2025-03-06 17:26:11,303 Stage-3 map = 0%, reduce = 0%
2025-03-06 17:26:17,585 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 2.72 sec
2025-03-06 17:26:24,970 Stage-3 map = 100%, reduce = 100%, Cumulative CPU 5.77 sec
MapReduce Total cumulative CPU time: 5 seconds 770 msec
Ended Job = job_1741254545702_0016
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2 Reduce: 1 Cumulative CPU: 13.66 sec HDFS Read: 22268 HDFS Write: 121 SUCCESS
Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 6.8 sec HDFS Read: 8476 HDFS Write: 125 SUCCESS
Stage-Stage-3: Map: 1 Reduce: 1 Cumulative CPU: 5.77 sec HDFS Read: 7815 HDFS Write: 126 SUCCESS
Total MapReduce CPU Time Spent: 26 seconds 230 msec
OK
2025 10 454.84615384615387
Time taken: 91.504 seconds, Fetched: 1 row(s)
```

## 7. Performance Considerations

- **Partitioning:** Data is partitioned by year, month, and day for efficient querying.
- **Columnar Storage:** Using Parquet reduces storage and improves query performance.
- **Optimized Queries:** Queries are structured to minimize data scans.

## 8. Execution Times

- **Pipeline Execution Time:** ~1 minute
- **Query Execution Times:**
  - Monthly Active Users: ~57 seconds
  - Top Content Categories: ~93 seconds

- Average Session Length: ~97 seconds

## 9. Conclusion

This project successfully implements a batch analytics pipeline, enabling MediaCo to analyze user interactions efficiently. The design choices optimize performance through proper partitioning, columnar storage, and structured queries.

## 10. References

- [Apache Hive Documentation](#)
- [HDFS Architecture](#)