Processing Different File Format :

legacy: csv, json, xml,.....
new: parquet,
delta table
apache iceberg

What is a file format?

standardized way of organizing and storing data in a file

we have multiple file format available like csv, txt, xml, json.

Introducing Parquet: The Next-Gen File Format.

Parquet is a modern columnar storage format designed for efficient analytics.

Why Move to Parquet-Based Formats Like Delta & Apache Iceberg?

Parquet is efficient, modern **Lake House formats** like **Delta Lake** and **Apache Iceberg** provide:

Pros:
ACID Transactions – Ensures data consistency (unlike raw Parquet files)
Schema Evolution – Handle changes without breaking queries
Time Travel – Query past versions of data
Partitioning & Indexing – Boosts query performance

Where we are store data ?

- local system
- cloud storage like AWS →s3, Google Cloud → google cloud storage, Microsoft Azure → Azure Blob Storage

- here I am using AWS S3

Why we need to migrate from legacy file format to new era file format ?

Details of Dataset Used for This Analysis

Source Stored in : S3

Initially stored as CSV, later converted to Parquet and Iceberg

CSV files size: 25.1 GB

Total rows: `520950475`

Note: simple integer and text data in this csv file , no long text data exist in this data set.

Infra Used

Method 1 : DataBricks Community version

Community Version provides us → Active cores : 2 , Memory : 15.5GB , Spark Version 3.3.2

Method 2 : Local System: Using Mac M4 Pro with Ram 24 GB , Spark 3.5.5, PySpark 3.5.5 [pyspark configure will show in later slide]

Method 3 : Databricks Community Version – Processing Data in Delta Format

Method 4: Dremio Setup on Local Machine (Mac M4) – Processing Delta & Apache Iceberg Tables

Method 1: Processing Data Using Databricks Community Edition

Read s3 folder [csv files] using `df_csv = spark.read.csv("s3a://S3-BUCKETNAME/FolderName/", header=True, inferSchema=True)`

– it took around ~24-30 minutes to process ~25 GB data, then we can run SQL like count and aggregate query, for faster processing we can cache dataframe [using df_csv.cache()]

Method 2: **Processing Data Using Local Spark Setup (macOS M4)**

Tried using pyspark --master "local[3]" --driver-memory 8G – It took more than 3++ hours then I cancelled – no point to wait further

Tried : pyspark --master "local[6]" --driver-memory 16G – I waited ~30-40 minutes then cancelled the job – again no point to wait further

Tried: pyspark --master "local[6]" --driver-memory 16G --executor-memory 8G --executor-cores 4 --conf spark.executor.instances=2

Tried other optimal config, but it will be always more than ~30 minutes ++ , not bothering too much on configuration as I am running simple query on this dataset
`[df csv = spark.read.csv("s3a://S3-BUCKETNAME/FolderName/", header=True, inferSchema=True) ] – same using in Method 1.`

Method 3: **Processing Data using Databricks Community edition [Delta Format]**

believe me or not it took less than minutes , same data set only difference is we converted it on delta format

delta conversion using same databricks community cluster , it took around ~40 min , but it does not matter we are converting from csv just to check processing time on delta table.

```
df_csv.write.format("delta").mode("overwrite").save("s3a://S3-BUCKETNAME/delta_TABLENAME/")
#creating df
 →
```

Method 3: **Processing using Databricks community edition [Delta Format]** – continue

```
df_delta =
spark.read.format("delta").load("s3a:///S3-BUCKETNAME/delta_TA
BLENAME/")
df_delta.show()
print(df_delta.count())
#520950475

# group by and other aggregate query is also much optimized.
```

Method 4: **Processing Data Local Dremio**

Iceberg Table created from same s3 CSV folder using Dremio – as DATABRICKS Community version not supporting apache iceberg it is available on PAID Version.

Read Delta Format and Iceberg Format ,using Dremio believe me both table is much much more performant way to query this data set.

it took almost ~1 minutes on both format.

I tried

Method 4: **Processing Data Local Dremio** -Continue

```
#DELTA
select  customer_id,count(*) as cc FROM
"SOURCE"."SCHEMA"."DELTA_TABLENAME" where EXTRACT(YEAR FROM
date_inserted) in (2025,2024) group by customer_id order by cc
desc  ;
```

```
# ICEBERG
select  uid,count(*) as cc FROM
"SOURCE"."SCHEMA".."ICEBERG_TABLENAME" where EXTRACT(YEAR FROM
date_inserted) in (2025,2024) group by customer_id order by cc
desc  ;
```

Data Size on S3

All Folder / Format is available on AWS S3:

CSV : ~25.1 GB , 10420 Files

DELTA : ~6.6 GB , 527 Files , Data File format : parquet with other metadata

Apache Iceberg : ~ 4.5 GB , 56 Files ,Data File format : parquet with other metadata

Note: DELTA and Apache Iceberg has less files as we use CSV folder to convert data into delta and iceberg format.

CSV has lots of small files as it is dumped in batches .

Findings

This is just for demonstrate if we use Delta / apache Iceberg format , believe me analysis is much easier and much faster than traditional format csv.

even I did not use partitioning here on Delta/ apache Iceberg , I simple created data using same CSV bucket and running SQL .

Spark On local required configuration and resource too

Databricks – required python/Pyspark knowledge.

Dremio – if we use Dremio , we just need to know SQL

Conclusion

If we use new format Delta / Apache Iceberg definitely we can quickly analyze huge data set

Dremio , no need to know Python,spark just SQL knowledge is enough

Dremio setup is quite simple and easy on Local setup & we also have option Dremio Cloud .

Upcoming ..

Dremio setup and further process huge data set.

Thanks You