

Taller 2

Daniel Guillermo Crespo Duarte – Código: 40573

UNIVERSIDAD ECCI

**SEMINARIO BIG DATA Y GERENCIA DE DATOS
2024-1**

**PROFESOR : ELIAS BUITRAGO BOLIVAR
BOGOTÁ, D.C.**

2024

Resultados de Pandas

Importar archivos:

Archivos

🔍

📁

📄

📷

🔗

{x}

drive

sample_data

+ Código + Texto

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

Playing with pandas

import pandas as pd

#flights_file1 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2018.parquet"

#flights_file2 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2019.parquet"

#flights_file3 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2020.parquet"

#flights_file4 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2021.parquet"

#flights_file5 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2022.parquet"

df1 = pd.read_parquet(flights_file1)

df2 = pd.read_parquet(flights_file2)

df3 = pd.read_parquet(flights_file3)

df4 = pd.read_parquet(flights_file4)

df5 = pd.read_parquet(flights_file5)

[] # df = pd.concat([df3, df5])

df = df2

[] # %%timeit

df_agg = df.groupby(["Airline", "Year"])[["DepDelayMinutes", "ArrDelayMinutes"]].agg(["mean", "sum", "max"])

df_agg = df_agg.reset_index()

df_agg.to_parquet("temp_pandas.parquet")

[] !ls -l temp_pandas.parquet

12K -rw-r--r-- 1 root 9.1K Jun 19 20:06 temp_pandas.parquet

[] pd.read_parquet("temp_pandas.parquet")

Airline	Year	DepDelayMinutes	ArrDelayMinutes
		mean	sum
		max	mean
			sum
			max

Realizar pruebas con los archivos 3 y 5:

Archivos

🔍

📁

📄

📷

🔗

{x}

drive

sample_data

+ Código + Texto

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

Playing with pandas

[] import pandas as pd

#flights_file1 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2018.parquet"

#flights_file2 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2019.parquet"

#flights_file3 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2020.parquet"

#flights_file4 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2021.parquet"

#flights_file5 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2022.parquet"

df1 = pd.read_parquet(flights_file1)

df2 = pd.read_parquet(flights_file2)

df3 = pd.read_parquet(flights_file3)

df4 = pd.read_parquet(flights_file4)

df5 = pd.read_parquet(flights_file5)

[] df = pd.concat([df3, df5])

df = df2

[] # %%timeit

df_agg = df.groupby(["Airline", "Year"])[["DepDelayMinutes", "ArrDelayMinutes"]].agg(["mean", "sum", "max"])

df_agg = df_agg.reset_index()

df_agg.to_parquet("temp_pandas.parquet")

[] !ls -l temp_pandas.parquet

12K -rw-r--r-- 1 root 9.1K Jun 19 20:06 temp_pandas.parquet

[] pd.read_parquet("temp_pandas.parquet")

Airline	Year	DepDelayMinutes	ArrDelayMinutes
		mean	sum
		max	mean
			sum
			max

Obtener información de los archivos:

Archivos

🔍

📁

📄

📷

🔗

{x}

drive

sample_data

+ Código + Texto

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

Playing with pandas

[1] import pandas as pd

#flights_file1 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2018.parquet"

#flights_file2 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2019.parquet"

#flights_file3 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2020.parquet"

#flights_file4 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2021.parquet"

#flights_file5 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2022.parquet"

df1 = pd.read_parquet(flights_file1)

df2 = pd.read_parquet(flights_file2)

df3 = pd.read_parquet(flights_file3)

df4 = pd.read_parquet(flights_file4)

df5 = pd.read_parquet(flights_file5)

[2] df = pd.concat([df3, df5])

df = df2

[3] %%timeit

df_agg = df.groupby(["Airline", "Year"])[["DepDelayMinutes", "ArrDelayMinutes"]].agg(["mean", "sum", "max"])

df_agg = df_agg.reset_index()

df_agg.to_parquet("temp_pandas.parquet")

1.63 s ± 305 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)

[] !ls -l temp_pandas.parquet

Resultados del modelo:

Archivos

🔍

📁

🔄

🖨

🔊

{x}

📁

drive

📁

sample_data

📁

temp_pandas.parquet

+ Código + Texto

✓ RAM Disco

+ Gemini

1.63 s ± 385 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)

[5] [ls -GFlash temp_pandas.parquet

12K -rw-r--r-- 1 root 9.9K Jul 3 00:21 temp_pandas.parquet

[5] pd.read_parquet("temp_pandas.parquet")

14	Compass Airlines	2020	0.410000	1.200100	1431.0	0.041490	1.200930	1414.0
15	Delta Air Lines Inc.	2020	5.581694	3083283.0	1195.0	6.209070	3424414.0	1193.0
16	Delta Air Lines Inc.	2022	13.842472	6948367.0	1287.0	13.111550	6565084.0	1285.0
17	Empire Airlines Inc.	2020	6.861561	32613.0	274.0	7.028136	33222.0	272.0
18	Endeavor Air Inc.	2020	5.653603	1156405.0	2579.0	5.877866	1200707.0	2560.0
19	Endeavor Air Inc.	2022	13.284602	1819220.0	1973.0	14.184149	1935952.0	1968.0
20	Envoy Air	2020	6.685444	1339382.0	2248.0	8.417206	1682490.0	2238.0
21	Envoy Air	2022	10.196954	1498942.0	5327.0	10.942435	1603023.0	5324.0
22	ExpressJet Airlines Inc.	2020	6.396004	308249.0	1357.0	7.443095	357797.0	1338.0
23	Frontier Airlines Inc.	2020	7.398021	640062.0	645.0	7.330406	633479.0	638.0
24	Frontier Airlines Inc.	2022	23.560713	1982257.0	1288.0	23.980928	2011856.0	1311.0
25	GoJet Airlines, LLC db/a United Express	2020	8.309550	304786.0	1408.0	8.336411	305196.0	1382.0
26	GoJet Airlines, LLC db/a United Express	2022	20.259624	656817.0	1220.0	21.749690	702254.0	1199.0
27	Hawaiian Airlines Inc.	2020	3.592000	137491.0	1484.0	4.092661	156487.0	1481.0
28	Hawaiian Airlines Inc.	2022	8.890739	372193.0	1847.0	9.069343	379289.0	1805.0
29	Horizon Air	2020	4.849894	445133.0	775.0	5.351232	489932.0	758.0
30	Horizon Air	2022	8.893327	492717.0	803.0	9.589591	529729.0	803.0
31	JetBlue Airways	2020	8.213825	1118838.0	1135.0	8.503231	1155334.0	1153.0

<>

Archivos

🔍

📁

🔄

🖨

🔊

{x}

📁

drive

📁

sample_data

📁

temp_pandas.parquet

+ Código + Texto

✓ RAM Disco

+ Gemini

[5] pd.read_parquet("temp_pandas.parquet").info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46 entries, 0 to 45
Data columns (total 9 columns):
Column Non-Null Count Dtype
0 (Airline,) 46 non-null object
1 (Year,) 46 non-null int64
2 (DepDelayMinutes, mean) 46 non-null float64
3 (DepDelayMinutes, sum) 46 non-null float64
4 (DepDelayMinutes, max) 46 non-null float64
5 (ArrDelayMinutes, mean) 46 non-null float64
6 (ArrDelayMinutes, sum) 46 non-null float64
7 (ArrDelayMinutes, max) 46 non-null float64
dtypes: float64(6), int64(1), object(1)
memory usage: 3.0+ KB

<>

Playing with Polars

[] 4 celdas ocultas

Resultados de Polars

Archivos

🔍

📁

🔄

🖨

🔊

{x}

📁

drive

📁

sample_data

📁

temp_pandas.parquet

+ Código + Texto

✓ RAM Disco

+ Gemini

Playing with Polars

[1] import polars as pl

[3] flights_file1 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2018.parquet"
flights_file2 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2019.parquet"
flights_file3 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2020.parquet"
flights_file4 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2021.parquet"
flights_file5 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2022.parquet"
df1 = pl.scan_parquet(flights_file1)
df2 = pl.scan_parquet(flights_file2)
df3 = pl.scan_parquet(flights_file3)
df4 = pl.scan_parquet(flights_file4)
df5 = pl.scan_parquet(flights_file5)

[4] %%timeit
df_polars = (
pl.concat([df1, df2, df3, df4, df5])
.groupby(["Airline", "Year"])
.agg(
pl.col("DepDelayMinutes").mean().alias("avg_dep_delay"),
pl.col("DepDelayMinutes").sum().alias("sum_dep_delay"),
pl.col("DepDelayMinutes").max().alias("max_dep_delay"),
pl.col("ArrDelayMinutes").mean().alias("avg_arr_delay"),
pl.col("ArrDelayMinutes").sum().alias("sum_arr_delay"),
pl.col("ArrDelayMinutes").max().alias("max_arr_delay"),
)
)
df_polars.write_parquet("temp_polars.parquet")

%magic-timeit>3: DeprecationWarning: 'groupby' is deprecated. It has been renamed to 'group_by'.
10.2 s ± 1.13 s per loop (mean ± std. dev. of 7 runs, 1 loop each)

[5] [ls -GFlash temp_polars.parquet

12K -rw-r--r-- 1 root 8.1K Jul 3 00:29 temp_polars.parquet

<>

Disco

79.23 GB de espacio disponible

0 s completado a las 19:29

✕

Resultados de Pyspark

Archivos

drive

sample_data

temp_spark.parquet

RAM

Disco

Gemini

[4] spark = SparkSession.builder.master("local[1]").appName("airline-example").getOrCreate()

[5] flights_file1 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2018.parquet"
flights_file2 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2019.parquet"
flights_file3 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2020.parquet"
flights_file4 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2021.parquet"
flights_file5 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2022.parquet"

[6] df_spark1 = spark.read.parquet(flights_file1)
df_spark2 = spark.read.parquet(flights_file2)
df_spark3 = spark.read.parquet(flights_file3)
df_spark4 = spark.read.parquet(flights_file4)
df_spark5 = spark.read.parquet(flights_file5)

[7] df_spark = df_spark1.union(df_spark2)
df_spark = df_spark.union(df_spark3)
df_spark = df_spark.union(df_spark4)
df_spark = df_spark.union(df_spark5)

[8] %%timeit

df_spark_agg = df_spark.groupby("Airline", "Year").agg(
 avg("ArrDelayMinutes").alias('avg_arr_delay'),
 sum("ArrDelayMinutes").alias('sum_arr_delay'),
 max("ArrDelayMinutes").alias('max_arr_delay'),
 avg("DepDelayMinutes").alias('avg_dep_delay'),
 sum("DepDelayMinutes").alias('sum_dep_delay'),
 max("DepDelayMinutes").alias('max_dep_delay'),
)
df_spark_agg.write.mode('overwrite').parquet("temp_spark.parquet")

9.79 s ± 716 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)

ls -l temp_spark.parquet

ls: cannot access 'temp_spark.parquet': No such file or directory

Plavinn with dask

0 s completado a las 18:27

Resultados de Dask

Archivos

drive

sample_data

temp_spark.parquet

temp_dask.parquet

RAM

Disco

Gemini

Playing with dask

[3] import pandas as pd
import dask.dataframe as dd
flights_file1 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2018.parquet"
flights_file2 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2019.parquet"
flights_file3 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2020.parquet"
flights_file4 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2021.parquet"
flights_file5 = "/content/drive/MyDrive/SEMINARIO/ACTIVIDAD1/Combined_Flights_2022.parquet"
df1 = dd.read_parquet(flights_file1)
df2 = dd.read_parquet(flights_file2)
df3 = dd.read_parquet(flights_file3)
df4 = dd.read_parquet(flights_file4)
df5 = dd.read_parquet(flights_file5)

[4] df = dd.concat([df3, df5])

[5] print(df.compute())

FlightDate Airline Origin Dest Cancelled Diverted \
0 2020-09-01 Comair Inc. PHL DAV False False
1 2020-09-02 Comair Inc. PHL DAV False False
2 2020-09-03 Comair Inc. PHL DAV False False
3 2020-09-04 Comair Inc. PHL DAV False False
4 2020-09-05 Comair Inc. PHL DAV False False
... ..
590537 2022-03-31 Republic Airlines MSY EWR False True
590538 2022-03-17 Republic Airlines CLT EWR True False
590539 2022-03-08 Republic Airlines ALB ORD False False
590540 2022-03-25 Republic Airlines EWR PIT False True
590541 2022-03-07 Republic Airlines EWR RDU False True

CRSDepTime DepTime DepDelayMinutes DepDelay ... WheelsOff \
0 1905 1858.0 0.0 -7.0 ... 1914.0
1 1905 1855.0 0.0 -10.0 ... 2000.0
2 1905 1857.0 0.0 -8.0 ... 1910.0
3 1905 1856.0 0.0 -9.0 ... 1910.0
4
590537 1940 2014.0 25.0 25.0 ... 2031.0
590538 1733 1817.0 44.0 44.0 ... NaN
590539 1700 2118.0 378.0 378.0 ... 2137.0

0 s completado a las 19:11

Comparación

- Pandas no puede analizar 5 datos a la vez, solo puede procesar 2 y solo permite agrupar de 2 por lo cual no toma tanto tiempo su ejecución.
- Polaris sí puede analizar 5 datos a la vez, y puede agrupar 5 datos pero su tiempo de ejecución es más extenso.
- PySpark si puede analizar 5 datos a la vez, pero no puede agrupar datos, sólo puede analizarlos en paralelo y tiene un rendimiento similar a Polaris.
- Dask si puede analizar 5 datos a la vez, pero no puede agrupar datos, sólo puede analizarlos en paralelo.