Integrantes:
Paula Rios -91105
Oscar Velasquez -91273

Laboratorio 2

Introducción

En este laboratorio se realizarán pruebas con las siguientes librerías: Pandas, Polars, Spark y Dask; en el cual se evidenciará cuales librerías son más eficientes en cuanto al manejo de grandes volúmenes de datos y también cuales son las menos eficientes.

Pruebas

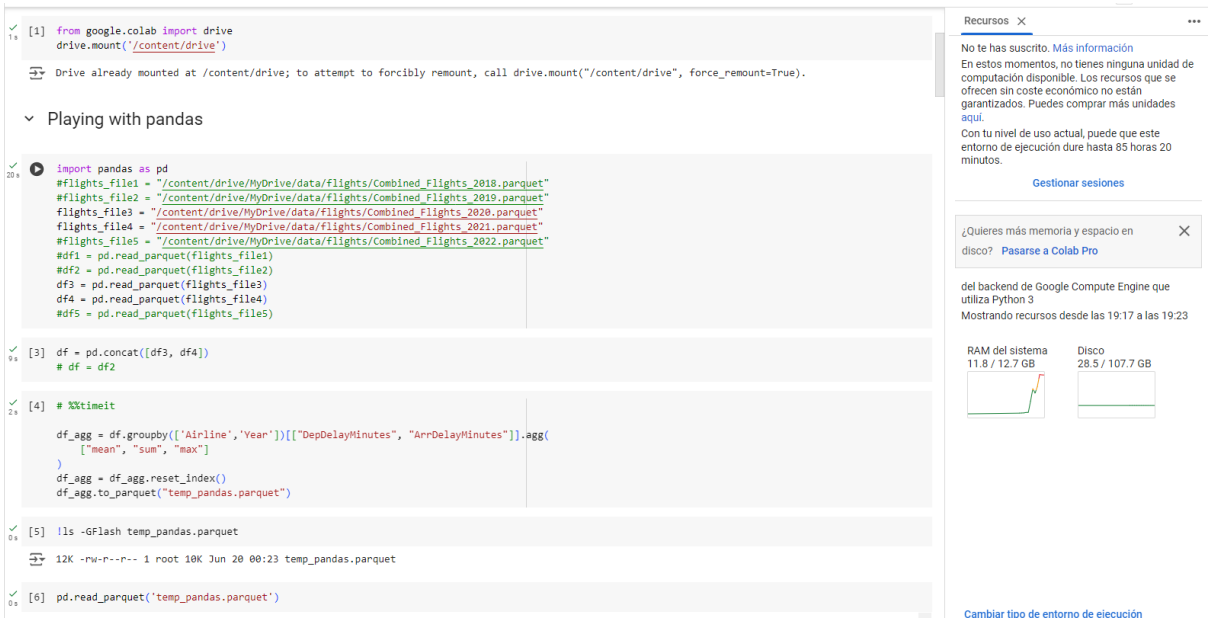
A continuación, se realizará una tabla con los distintos intentos de pruebas en las librerías mencionadas anteriormente y este nos mostrará datos como los archivos usados, La Ram consumida por intento y el tiempo que tardo en su ejecución:

Pandas	intento 1	Archivos del año 2018, 2019, 2020	Tiempo: 38 seg Ram: 12.7 GB **Fallido concatenación**
	intento 2	Archivos del año 2021, 2019, 2020	Tiempo: 40 seg Ram: 12.7 GB **Fallido carga**
	intento 3	Archivos del año 2021, 2020	Tiempo: 36 seg Ram: 11.8 GB **Exitoso**
	intento 4	Archivos del año 2020, 2022, 2018	Tiempo: 45 seg Ram: 12.7 GB **Fallido carga**
	intento 5	Archivos del año 2021, 2019, 2020	Tiempo: 39 seg Ram: 12.7 GB **Fallido concatenación**
Polars	intento 1	Archivos del año 2021, 2019, 2020, 2022, 2018	Tiempo: 1 min 28 seg Ram: 2.7 GB **Exitoso**
Spark	intento 1	Archivos del año 2021, 2019, 2020, 2022, 2018	Tiempo: 1 min 36 seg Ram: 2.1 GB **Exitoso**

Dask	intento 1	Archivos del año 2018, 2019, 2020	Tiempo: 38 seg Ram: 12.7 GB **Fallido print**
	intento 2	Archivos del año 2021, 2019, 2020	Tiempo: 26 seg Ram: 12.7 GB **Fallido print**
	intento 3	Archivos del año 2021, 2020	Tiempo: 1 min Ram: 11.5 GB **Exitoso**
	intento 4	Archivos del año 2020, 2022, 2018	Tiempo: 40 seg Ram: 12.7 GB **Fallido print**
	intento 5	Archivos del año 2021, 2019, 2020	Tiempo: 26 seg Ram: 12.7 GB **Fallido print**

Evidencias

Pandas



```
[1] from google.colab import drive
    drive.mount('/content/drive')

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/data/flights/Combined_Flights_2018.parquet"
#flights_file2 = "/content/drive/MyDrive/data/flights/Combined_Flights_2019.parquet"
flights_file3 = "/content/drive/MyDrive/data/flights/Combined_Flights_2020.parquet"
flights_file4 = "/content/drive/MyDrive/data/flights/Combined_Flights_2021.parquet"
#flights_file5 = "/content/drive/MyDrive/data/flights/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)

[3] df = pd.concat([df3, df4])
    # df = df2

[4] # %%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")

[5] !ls -GFlash temp_pandas.parquet

12K -rw-r--r-- 1 root 10K Jun 20 00:23 temp_pandas.parquet

[6] pd.read_parquet('temp_pandas.parquet')
```

Recursos

No te has suscrito. Más información

En estos momentos, no tienes ninguna unidad de computación disponible. Los recursos que se ofrecen sin coste económico no están garantizados. Puedes comprar más unidades aquí.

Con tu nivel de uso actual, puede que este entorno de ejecución dure hasta 85 horas 20 minutos.

Gestionar sesiones

¿Quieres más memoria y espacio en disco? Pasarse a Colab Pro

del backend de Google Compute Engine que utiliza Python 3

Mostrando recursos desde las 19:17 a las 19:23

RAM del sistema 11.8 / 12.7 GB

Disco 28.5 / 107.7 GB

Cambiar tipo de entorno de ejecución

Polars

```
%%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"),
    ])
).collect()

df_polars.write_parquet('temp_polars.parquet')

<magic-timeit>:3: DeprecationWarning: `groupby` is deprecated. It has been renamed to `group_by`.
```

Mostrando recursos desde las 19:24 a las 19:29

RAM del sistema 2.7 / 12.7 GB

Disco 28.5 / 107.7 GB

Spark

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

[12] flights_file1 = "/content/drive/MyDrive/data/flights/Combined_Flights_2018.parquet"
     flights_file2 = "/content/drive/MyDrive/data/flights/Combined_Flights_2019.parquet"
     flights_file3 = "/content/drive/MyDrive/data/flights/Combined_Flights_2020.parquet"
     flights_file4 = "/content/drive/MyDrive/data/flights/Combined_Flights_2021.parquet"
     flights_file5 = "/content/drive/MyDrive/data/flights/Combined_Flights_2022.parquet"

[13] 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)

[14] 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)

[15] %%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')

8.66 s ± 1.14 s per loop (mean ± std. dev. of 7 runs, 1 loop each)

!ls -GFlash temp_pyspark.parquet

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

Recursos

No te has suscrito. Más información

En estos momentos, no tienes ninguna unidad de computación disponible. Los recursos que se ofrecen sin coste económico no están garantizados. Puedes comprar más unidades aquí.

Con tu nivel de uso actual, puede que este entorno de ejecución dure hasta 85 horas.

Gestionar sesiones

¿Quieres más memoria y espacio en disco? Pasarse a Colab Pro

del backend de Google Compute Engine que utiliza Python 3

Mostrando recursos desde las 19:24 a las 19:37

RAM del sistema 1.9 / 12.7 GB

Disco 29.4 / 107.7 GB

Dask

```
import pandas as pd
import dask.dataframe as dd
#flights_file1 = "/content/drive/MyDrive/data/flights/Combined_Flights_2018.parquet"
#flights_file2 = "/content/drive/MyDrive/data/flights/Combined_Flights_2019.parquet"
flights_file3 = "/content/drive/MyDrive/data/flights/Combined_Flights_2020.parquet"
flights_file4 = "/content/drive/MyDrive/data/flights/Combined_Flights_2021.parquet"
#flights_file5 = "/content/drive/MyDrive/data/flights/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)

[2] df = dd.concat([df3,df4])

[3] print(df.compute())

       FlightDate               Airline Origin Dest  Cancelled  Diverted  \
0      2020-09-01          Comair Inc.    PHL  DAY      False     False
1      2020-09-02          Comair Inc.    PHL  DAY      False     False
2      2020-09-03          Comair Inc.    PHL  DAY      False     False
3      2020-09-04          Comair Inc.    PHL  DAY      False     False
4      2020-09-05          Comair Inc.    PHL  DAY      False     False
...           ...                  ...    ...  ...        ...       ...
573774 2021-06-01  Southwest Airlines Co.    BNA  MDW      False     False
573775 2021-06-01  Southwest Airlines Co.    BNA  MDW      False     False
573776 2021-06-01  Southwest Airlines Co.    BNA  MIA      False     False
573777 2021-06-01  Southwest Airlines Co.    BNA  MIA      False     False
573778 2021-06-01  Southwest Airlines Co.    BNA  MKE      False     False

        CRSDepTime  DepTime  DepDelayMinutes  DepDelay  ...  WheelsOff  \
0             1905   1858.0              0.0      -7.0  ...     1914.0
1             1905   1858.0              0.0      -7.0  ...     1914.0
2             1905   1855.0              0.0     -10.0  ...     2000.0
3             1905   1857.0              0.0      -8.0  ...     1910.0
4             1905   1856.0              0.0      -9.0  ...     1910.0
...            ...      ...              ...       ...  ...        ...
573774        1255   1301.0              6.0       6.0  ...     1310.0
573775         730    727.0              0.0      -3.0  ...      740.0
573776         800    757.0              0.0      -3.0  ...      811.0
573777        1300   1252.0              0.0      -8.0  ...     1300.0
```

Recursos

No te has suscrito. Más información

En estos momentos, no tienes ninguna unidad de computación disponible. Los recursos que se ofrecen sin coste económico no están garantizados. Puedes comprar más unidades aquí.

Con tu nivel de uso actual, puede que este entorno de ejecución dure hasta 84 horas 50 minutos.

Gestionar sesiones

¿Quieres más memoria y espacio en disco? Pasarse a Colab Pro

del backend de Google Compute Engine que utiliza Python 3

Mostrando recursos desde las 19:51 a las 19:55

RAM del sistema 7.1 / 12.7 GB

Disco 29.4 / 107.7 GB

Cambiar tipo de entorno de ejecución

Conclusiones

- En la librería Pandas se puede observar que es la que toma más tiempo en procesar el código y más recursos usa.
- La librería Polars es la más veloz y también una de las que menos recursos usa siendo la más efectiva de las 4 librerías.
- Al comparar la librería Polars y Spark, realizando el cargue de 5 archivos correspondientes de los años 2018 - 2022, se logra evidenciar que Polars necesita menos tiempo para realizar el proceso, sin embargo, Spark es la librería que menos Ram consume.
- Las librerías Pandas y Dask son las menos eficientes, ya que, mientras Polars y Spark pueden realizar el proceso con los 5 archivos y con 12.5 GB de Ram, estas 2 tan solo pueden realizar el proceso con 2 archivos.