UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

BIG DATA:
Lab 07

Name:
Felix Blanco
Andres Riveros

May 11, 2023

Repositorio de Github: LAB_07

DuskDB

Para el siguiente laboratorio se trataron de ejecutar todas las consultas de la carpeta de *benchmark_queries* del repositorio de Github adjunto, como por ejemplo:

Listing 0.1 SQL *query* 06

```
SELECT
    SUM(l_extendedprice * l_discount) AS revenue
FROM
    lineitem
WHERE
    l_shipdate >= date '1994-01-01'
    AND l_shipdate < date '1994-01-01' + interval '1' year
    AND l_discount BETWEEN .06 - 0.01 AND .06 + 0.01
    AND l_quantity < 24;
```

Donde cada query del 1 al 23 se ejecutaron 5 veces para obtener un promedio de los tiempos de ejecución, los tiempos de ejecución se pueden encontrar en el archivo *data/duckdb_times.csv*. Por otro lado, para la ejecución de cada consulta se procedió de la siguiente manera:

Listing 0.2 DuckDB python files

```
def read_parquet(name):
  return pd.read_parquet(path.join(parquet_path, name))

def read_benchmark(name):
  with open(path.join(benchmark_path, name)) as f:
    return f.read()

def make_query(query, name):
  start = time.time()
  output = db.sql(query)
  end = time.time()
  with open(duckdb_times, 'a') as f:
    f.write(f'{name},{(end - start) * 1000}\n')
  # Results to CSV
  column_names = [desc[0] for desc in output.description]
  df = pd.DataFrame(output.fetchall(), columns=column_names)
  df.to_csv(path.join(results_path, f'{name:02d}.csv'), index=False)
```

Como se puede ver en el código *python* se leen las tablas de los archivos *.parquet*, asi como también se leen las consultas. Para finalmente hacer las *queries* respectivas con el commando *sql* de *duckdb*. Cada uno de los tiempos de ejecución se guardan en el archivo *duckdb_times.csv* como también los resultados de la query en su respectivo archivo *.csv*, todo esto para hacer las comparaciones respectivas en los siguientes experimentos.

Para más detalles se pueden revisar los siguiente archivos en el repositorio:

- **Carpeta benchmark_queries:** Para ver las consultas ejecutadas
- **Carpeta data:** Para ver los resultados de los tiempos de ejecución
- **Carpeta expected_results:** Para ver los resultados de las consultas
- **Archivo LAB_06.py:** Para ver el código *python* de *DuckDB*

todo esto de manera más detallada en el repositorio *Github* en el siguiente link: LAB 07

Dask

A diferencia de *DuckDB*, con *Dask* se tiene que cargar las tablas al contexto de ejecución. Por lo cual, luego de cargar las tablas del formato *.parquet* se ejecutó el siguiente código:

Listing 0.3 Dask SQL python files

```
# Dask temporary tables
c.create_table('customer', customer)
c.create_table('lineitem', lineitem)
c.create_table('nation', nation)
c.create_table('orders', orders)
c.create_table('part', part)
c.create_table('partsupp', partsupp)
c.create_table('region', region)
c.create_table('supplier', supplier)
```

Donde cada query del 1 al 23 se ejecutaron 5 veces para obtener un promedio de los tiempos de ejecución, los tiempos de ejecución se pueden encontrar en el archivo *data/dask_times.csv*. Por otro lado, para la ejecución de cada consulta se procedió de la siguiente manera:

Listing 0.4 Dask SQL python files

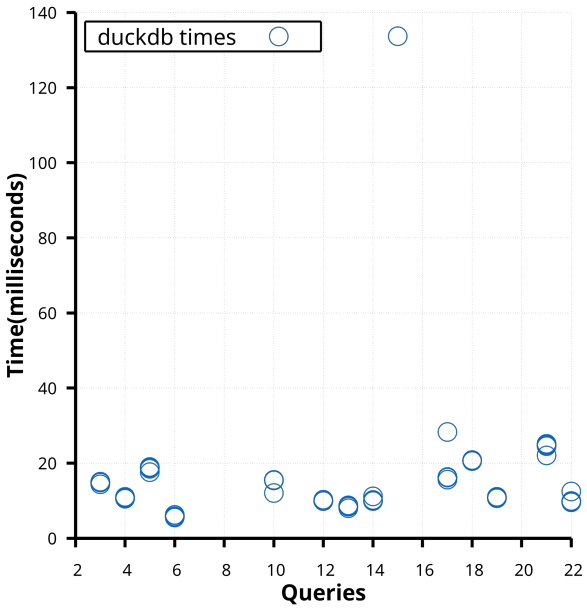
```
def read_parquet(name):
  return pd.read_parquet(path.join(parquet_path, name))

def read_benchmark(name):
  with open(path.join(benchmark_path, name)) as f:
    return f.read()

def make_query(query, name):
  start = time.time()
  output = c.sql(query).compute()
  end = time.time()
  with open(dask_times, 'a') as f:
    f.write(f'{name},{(end - start) * 1000}\n')
  # Results to CSV
  df.to_csv(path.join(results_path, f'{name:02d}.csv'), index=False)
```

Como se puede ver en el código *python* se leen las tablas de los archivos *.parquet*, asi como también se leen las consultas. Para finalmente hacer las *queries* respectivas con el commando *sql* de *dask*; sin embargo, debido a la naturaleza de *Dask* la consulta será ejecutada al llamar al commando*compute*. Cada uno de los tiempos de ejecución se guardan en el archivo *dask_times.csv* como también los resultados de la query en su respectivo archivo *.csv*, todo esto para hacer las comparaciones respectivas en los siguientes experimentos.

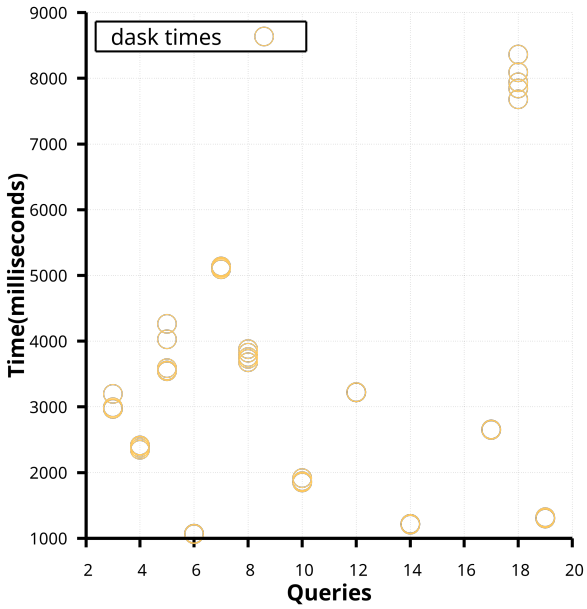
Comparación DuckDB vs Dask SQL

Con los tiempos obtenidos de las consultas de *DuckDB* y *Dask SQL* se procedió a realizar una comparación de los tiempos de ejecución de cada una de las consultas, obteniendo los siguientes resultados:

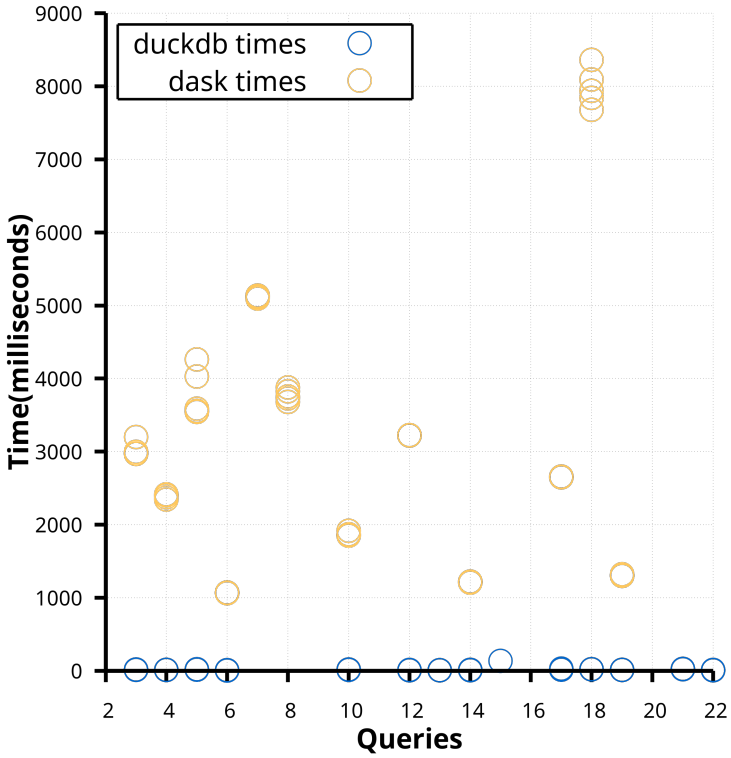
Como se puede ver en la figura 1a y 1b se puede ver que los tiempos de ejecución de *DuckDB* son menores que los tiempos de ejecución de *Dask SQL*, esto se debe quizá al tamaño del dataset propuesto.



(a) DuckDB execution times



(b) Dask SQL execution times



(c) Comparing DuckDB and Dask SQL execution times