CS 488/588: Cloud & Cluster Data Management

Freeway Data in MongoDB

Portland State University, Fall 2020

The DJS

Dominique
Moore

Jane
Seigman

Santiago
Tobon

TABLE OF CONTENTS

01

DATA

The Freeway Dataset, our MongoDB Data Model,
and our ETL Process

Freeway Dataset

- Relational data in .csv files
- Full of empty NULL values
- Querying in this format would require many joins
- Large freeway_loopdata file

detectorid	starttime	volume	speed	occupancy	status	dqflags
1345	9/15/2011 0:00:00	0		0	0	0
1345	9/15/2011 0:00:20				0	0
1345	9/15/2011 0:00:40	0		0	0	0
1345	9/15/2011 0:01:00	0		0	0	0
1345	9/15/2011 0:01:20	0		0	0	0
1345	9/15/2011 0:01:40	1	47	0	3	0
1345	9/15/2011 0:02:00	0		0	0	0
1345	9/15/2011 0:02:20	0		0	0	0

Table	A <->	Collection
Rows	of <->	Documents
Primary Key	identified by a <->	_id:ObjectId()
Columns	and contains <->	Fields

```
{
    _id: ObjectId(123),
    field: "value"
}
{
    _id: ObjectId(124),
    field: "value"
}
```

A NoSQL, distributed, document-oriented database that uses collections of JSON-type documents which support embedded fields for the storage of related data.

MongoDB

OUR DATA MODEL

Detector Collection:


```
{
        _id: ObjectId("5fad9e5bfb3c57497c725fa5")
        detectorid:1345
        highway: {
                highwayid:3
                shortdirection:"N"
                direction:"NORTH"
                highwayname:"I-205"
        }
        milepost:14.32
        locationtext:"Sunnyside NB"
        detectorclass:1
        lanenumber:1
        station: {
                stationid:1045
                upstream:0
                downstream:1046
                stationclass:1
                numberlanes:4
                latlon:"45.43324,-122.565775"
                length:0.94
        }
}
```


Freeway Loop Data Collection:


```
{
        _id: ObjectId("5fad9f01fb3c57497c725faf")
        detectorid:1345
        starttime:"2011-09-15 00:02:40-07"
        volume:1
        speed:66
}
```

EXTRACT TRANSFORM LOAD

```bash
#!/bin/bash
FNAME=freeway_loopdata.csv
HEADER=$(head -1 $FNAME)
split -b 300m $FNAME sections
n=1
for f in sections*
do
    if [ $n -gt 1 ]; then
        echo $HEADER > part${n}.csv
    fi
    cat $f >> part${n}.csv
    rm $f
    ((n++))
done
```

- The raw freeway_loopdata.csv was too large to load into our database
- We used this split.sh script to section it into workable files
- We cleaned each file of NULL/0 values.
- This allowed us to load all data

- The detectors, stations, and highways were in separate .csv files
- We wrote a Python script to format the data into a detector collection fitting our data model.
- This was saved to a JSON file and loaded to our database

```python
import csv
import json
import pandas as pd

# read in csv files into dataframes
df_de = pd.read_csv ("data/freeway_detectors.csv")
df_st = pd.read_csv ("data/freeway_stations.csv")
df_hw = pd.read_csv ("data/highways.csv")


#Save detector list to json file
with open("detectors.json", "w") as detectors_file:
    json.dump(detectors, detectors_file, indent=4)
```

02

Queries

How we answered the freeway dataset questions

Connecting with Python

```python
from pymongo import MongoClient
from pprint import pprint
import getpass as gp
pw = gp.getpass()
username = "DJs"
password = pw
dbname = "djs-freeway"
uri = "mongodb+srv://" + username + ":" + password + \
    "@ccdm-project.f4c6t.mongodb.net/" + dbname + "?retryWrites=true&w=majority"
#client = MongoClient()
#client = MongoClient(uri)
#db = client.test
try:
    client = MongoClient(uri)
    db = client.test
    print("Connected Successfully!!!")
except:
    print("Could not connect to db :( ")


mydb = client[dbname]

de_collection = mydb["freeway_detectors"]
lp_collection = mydb["freeway_loopdata"]

```

Freeway Dataset Questions

Q1

Count low speeds and high speeds: Find the number of speeds < 5 mph and > 80 mph in the data set.

Q2

Volume: Find the total volume for the station Foster NB for Sept 15, 2011.

Q3

Single-Day Station Travel Times: Find travel time for station Foster NB for 5-minute intervals for Sept 15, 2011. Report travel time in seconds.

Q4

Peak Period Travel Times: Find the average travel time for 7-9AM and 4-6PM on September 22, 2011 for the I-205 NB freeway. Report travel time in minutes.

Q5

Route Finding: Find a route from Johnson Creek to Columbia Blvd on I-205 NB using the upstream and downstream fields.

Q6

Update: Change the milepost of the Foster NB station to 22.6.

Q1

Count low speeds and high speeds: Find the number of speeds < 5 mph and > 80 mph in the data set.

```
# Query 1: count the number of speeds < 5 mph and > 80 mph
result1 = lp_collection.count_documents(
    {"speed": {"$lt": 5}})
print("Number of speeds < 5:", result1)
result2 = lp_collection.count_documents(
    {"speed": {"$gt": 80}})
print("Number of speeds > 80:", result2)
```

```
Number of speeds < 5: 1269204
Number of speeds > 80: 62203
```

Q6

Update: Change the milepost of the Foster NB station to 22.6.

```
#Query 6 update milepost at "Foster NB" from 18.1 -> 22.6
qry6 = de_collection.update_many({"locationtext": {"$eq":'Foster NB'}}, {"$set": {"milepost": 22.6}})

cursor = de_collection.find({"locationtext": {"$eq": 'Foster NB'}})
for record in cursor:
  print(record)
```

```
{'_id': ObjectId('5fc420ddce0db46a9c85b88d'), 'milepost': 18.1, 'locationtext': 'Foster NB'}
{'_id': ObjectId('5fc420ddce0db46a9c85b88e'), 'milepost': 18.1, 'locationtext': 'Foster NB'}
{'_id': ObjectId('5fc420ddce0db46a9c85b88f'), 'milepost': 18.1, 'locationtext': 'Foster NB'}
Matched:3
Updated:3
{'_id': ObjectId('5fc420ddce0db46a9c85b88d'), 'milepost': 22.6, 'locationtext': 'Foster NB'}
{'_id': ObjectId('5fc420ddce0db46a9c85b88e'), 'milepost': 22.6, 'locationtext': 'Foster NB'}
{'_id': ObjectId('5fc420ddce0db46a9c85b88f'), 'milepost': 22.6, 'locationtext': 'Foster NB'}
```

Clean up freeway loop data for the rest of the queries

```python
# Remove unneeded loopdata after running Q1
# Q2 & 3 only need loopdata from Sept 15
# Q4 only needs loopdata from Sept 22
# Q5 onyl needs detector collection
filter={
    '$nor': [
        {
            'starttime': {
                '$regex': '2011-09-22',
                '$options': 'i'
            }
        }, {
            'starttime': {
                '$regex': '2011-09-15',
                '$options': 'i'
            }
        }
    ]
}

result = lp_collection.delete_many(
  filter=filter
)
print(f"Deleted:{result.deleted_count}")
```

03

SELF-CRITIQUE

Challenges we faced and how we would change our approach in the future.

	QUERY PLANS	DATASET	ETL METHODS
DID CHANGE	• First, used MongoDB Compass UI aggregation tool • Switched to using a Python connection script • After running Q1, we truncated freeway_loopdata due to "max space used (512mb) error stopping us from performing operations.	StackOverflow (public dataset) • Exported data from BigQuery to GCS bucket files • Mounted GCS bucket to VM so we could access it like a local file system • AND THEN 1 out of 1000s of files took up all 512MB of space in MongoDB Switched to Freeway Dataset	• At first, we loaded in all four datasets as is, but realized we had to combine some of the datasets. • Edited the data in the freeway_loopdata to contain the only the fields we needed for the queries.
WOULD CHANGE	• Start figuring out queries earlier with smaller freeway_loopdata subset • Start with Python connection script, then use Compass UI as needed for assistance	• Do more research on dataset's size before trying to use it • Find a subset that would work for StackOverflow dataset	• Design data model FIRST • Write script to clean the freeway_loopdata of NULL/0 values • Add another index, other than the default one if possible

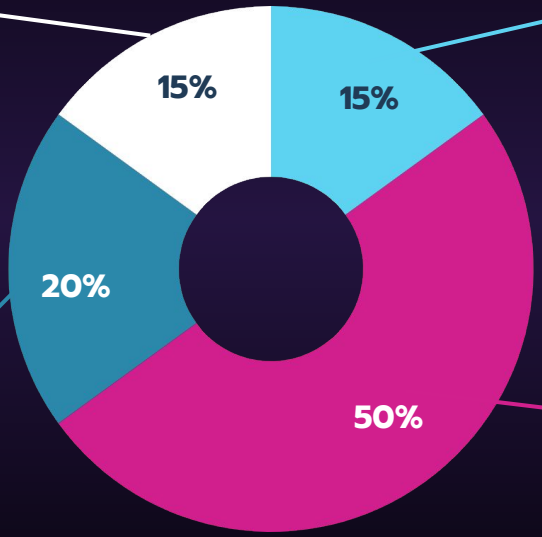
TIME INVESTMENT

ADMIN

- Setting up MongoDB went pretty smoothly on our own boxes and we also set up a VM on GCP. We created a Github for scripts and a shared Google drive for cleaned and transformed data.

ANALYZE

This has the potential to be a bit more than 15% by the time we complete them, but so far we have invested a disproportionate time doing ETL activities.



ARCHITECTURE

We spent a good amount of time thinking about the best data model with the goal of fitting it to the queries we needed to make.

ETL

Choosing a dataset, cleaning and transforming the data, and loading it to our data model took a lot of time.

04

LESSONS LEARNED

MongoDB and Cloud Data Management

OUR TAKEAWAYS

- ETL takes time and planning
- MongoDB is a user-friendly database
- Dataset sizes can be very misleading
- Taking the time to write a script is better than trying to do the task manually
- MongoDB gives you a default index if you don't specify one
- Denormalized data > Normalized data

OUR ADVICE

- Don't try to do ETL without deciding on a data model
- Some of the functions in the manual are deprecated, so check which versions supports the functions you're using.
- DB compass is great as a guide for writing some queries ...
- ... But just default to using a good connection script in a language you know

QUESTIONS?

05

APPENDIX

Dataset Size: *Full Freeway Dataset (approximately 660MB)*
Number of Queries : 4

Q1

Count low speeds and high speeds: Find the number of speeds < 5 mph and > 80 mph in the data set.

```
# Query 1: count the number of speeds < 5 mph and > 80 mph
result1 = lp_collection.count_documents(
    {"speed": {"$lt": 5}})
print("Number of speeds < 5:", result1)
result2 = lp_collection.count_documents(
    {"speed": {"$gt": 80}})
print("Number of speeds > 80:", result2)
```

```
Number of speeds < 5: 1269204
Number of speeds > 80: 62203
```

Q2

Volume: Find the total volume for the station Foster NB for Sept 15, 2011.

```
#Query 2 Find the total volume for the station Foster NB for Sept 15, 2011
qry2 = [{'$lookup': {'from': 'freeway_detectors',
            'localField': 'detectorid',
            'foreignField': 'detectorid',
            'as': 'detectors'}},
            {'$match': {'detectors.locationtext': 'Foster NB'}},
            {'$match': {'starttime': {'$regex': '2011-09-15'}}},
            {'$group': {'_id': 'None', 'TotalVolume': {
                '$sum': '$volume'}}}]
cursor = lp_collection.aggregate(qry2)
result = list(cursor)
print("Query 2 results: ",result)
```

```
Query 2 results:  [{'_id': 'None', 'TotalVolume': 49891}]
```

Q3

Single-Day Station Travel Times: Find travel time for station Foster NB for 5-minute intervals for Sept 15, 2011. Report travel time in seconds.

Q4

Peak Period Travel Times: Find the average travel time for 7-9AM and 4-6PM on September 22, 2011 for the I-205 NB freeway. Report travel time in minutes.

Q5

Route Finding: Find a route from Johnson Creek to Columbia Blvd on I-205 NB using the upstream and downstream fields.

```
#Query 5 Find the path from Johnson Creek to I-205 NB at Columbia
i = 0
text = 'Johnson Cr NB'
print(i,":",text)
while text != 'I-205 NB at Columbia':
  qry5 = [ {'$match': {'locationtext': text}},
         {'$lookup': {
            'from': 'freeway_detectors',
            'let': {'down': '$station.downstream',
                'lanenum': '$lanenumber'},
            'pipeline': [{'$match': {'$expr': {
                            '$eq': ['$station.stationid', '$$down']}}},
                            {'$match': {'$expr': {'$eq': ['$lanenumber', '$$lanenum']}}}
            ],
            'as': 'downstation'}},
            {'$limit': 1},
        {'$unwind': {'path': '$downstation'}},
        {'$project': {'downstation.locationtext': 1}}]
  cursor = de_collection.aggregate(qry5)
  result = list(cursor)
  i += 1
  for doc in result:
    text2 = doc["downstation"]
    text = text2["locationtext"]
    print(i,":",text)
```

```
0 : Johnson Cr NB
1 : Foster NB
2 : Powell to I-205 NB
3 : Division NB
4 : I-205 NB at Glisan
5 : I-205 NB at Columbia
```

Q6

Update: Change the milepost of the Foster NB station to 22.6.

```
#Query 6 update milepost at "Foster NB" from 18.1 -> 22.6
qry6 = de_collection.update_many({"locationtext": {"$eq":'Foster NB'}}, {"$set": {"milepost": 22.6}})

cursor = de_collection.find({"locationtext": {"$eq": 'Foster NB'}})
for record in cursor:
  print(record)
```

```
{'_id': ObjectId('5fc420ddce0db46a9c85b88d'), 'milepost': 18.1, 'locationtext': 'Foster NB'}
{'_id': ObjectId('5fc420ddce0db46a9c85b88e'), 'milepost': 18.1, 'locationtext': 'Foster NB'}
{'_id': ObjectId('5fc420ddce0db46a9c85b88f'), 'milepost': 18.1, 'locationtext': 'Foster NB'}
Matched:3
Updated:3
{'_id': ObjectId('5fc420ddce0db46a9c85b88d'), 'milepost': 22.6, 'locationtext': 'Foster NB'}
{'_id': ObjectId('5fc420ddce0db46a9c85b88e'), 'milepost': 22.6, 'locationtext': 'Foster NB'}
{'_id': ObjectId('5fc420ddce0db46a9c85b88f'), 'milepost': 22.6, 'locationtext': 'Foster NB'}
```

Source code available on Github

https://github.com/santitobon9/Cloud-Cluster-Freeway-Project.git