

Exercise 2

Kyle Hamilton

Dec 6, 2015

W205-6

Application Use Case

Graphing the counts of celebrities' names, politicians, ideas, etc., to measure and display their popularity over time.

Architecture

Apache Storm ingests Twitter streaming API.

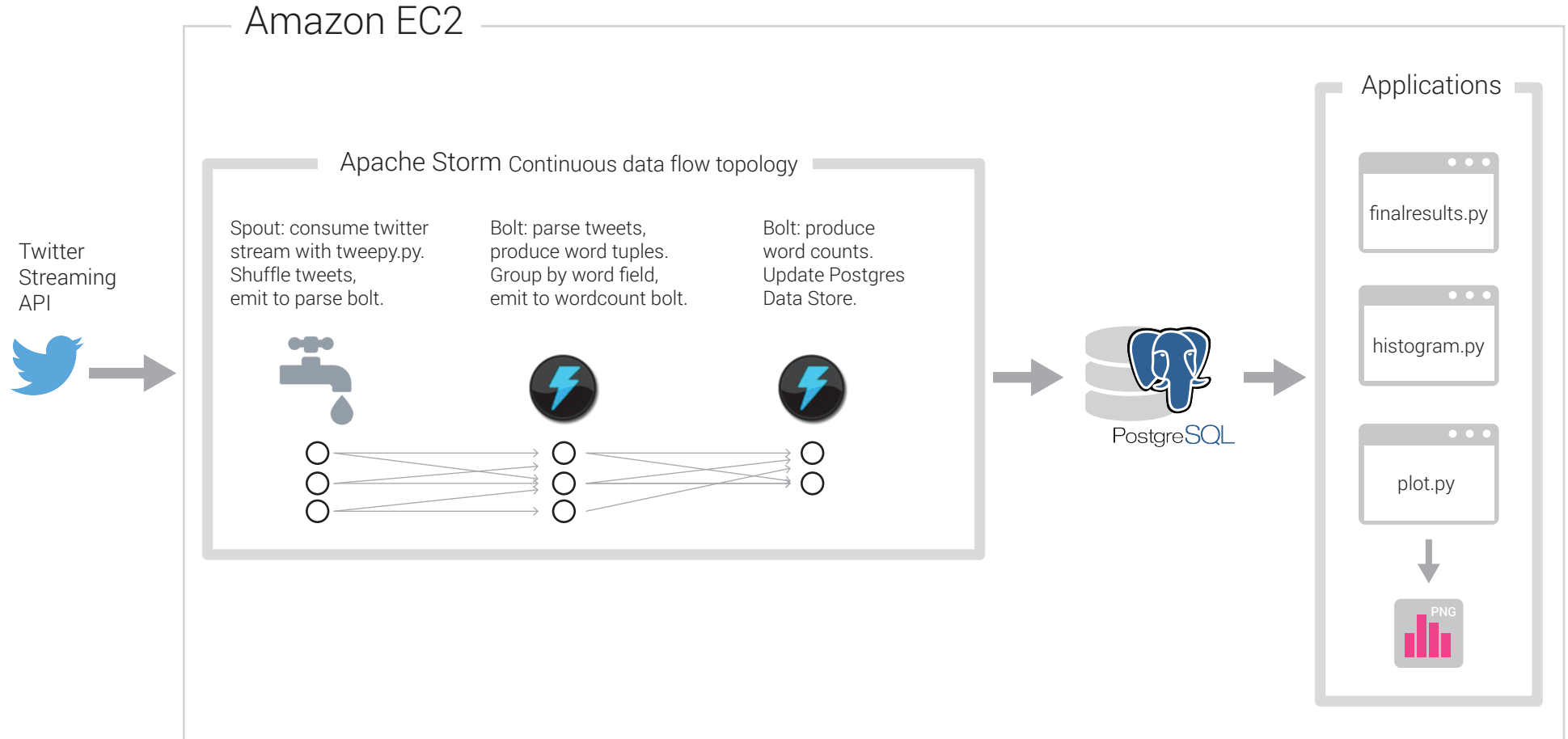
Tweet word counts are appended to a Postgres DB.

See diagram for details.

The following python apps display results:

- **Finalresults.py** – displays count for a word provided as argument, or if no arguments are provided, displays all words and counts in the DB.
- **Histogram.py** – displays words and counts within a given count range
- **Plot.py** – displays a bar graph of the 20 most frequently used words. While all words are stored in the DB, most common words are filtered out, as they are not interesting. This is merely a proof of concept. The common words list is, in reality, a much longer list. They are filtered at this stage, and not before for the sake of possible future applications, where one might actually want to know these words.

Exercise 2 - Architecture Diagram



Running the Application

Requirements:

AMI

The `__ucbw205_complete_plus_postgres_virtual2.7__` AMI is **RECOMMENDED**. It has the python 2.7 virtual environment already set up and running:

`ucbw205_complete_plus_postgres_virtual2.7`
`ami-003f7f6a`

But if you insist... you can use the AMI specified in the exercise. If you use this AMI, python 2.7 will be configured exactly as per the instructions in the exercise. **NOT recommended.**

`ucbw205_complete_plus_postgres_PY2.7`

Volume

`m3.large`

Dependencies:

(These will be installed as needed by following the Steps, see below)

Postgres

Python 2.7

lein

Python modules

matplotlib

streamparse

psycpg2

argparse

numpy

tweepy

Steps to install dependencies and run application

1. Once you have mounted `/data` on your volume:

```
cd /data
git clone git@github.com:kyleiwaniec/ucb205.git
cd ucb205/
git checkout exercise_2
```

2. Install dependencies.

You will be prompted to choose which AMI you are using, and to confirm whether or not Postgres is set up on /data:

```
./data/ucb205/exercise_2/install-dependencies.sh
```

3. Create the Database and table:

```
./data/ucb205/exercise_2/make-db.sh
```

4. Start the application:

```
cd /data/ucb205/exercise_2/EX2Tweetwordcount/  
sparse quickstart EX2Tweetwordcount  
sparse run
```

5. Open a new shell, and run applications:

```
python /data/ucb205/exercise_2/histogram.py  
python /data/ucb205/exercise_2/histogram.py 50 100  
python /data/ucb205/exercise_2/finalresults.py -w mother  
python /data/ucb205/exercise_2/plot.py
```

6. You can use scp to view the generated plot.png bar graph on your local machine:

```
scp -i your_key.pem root@xx.xxx.xx.xx:/data/ucb205/exercise_2/plot.png /path/to/local/dir
```

Complete file structure:

```
├── EX2Tweetwordcount
│   ├── .gitignore
│   ├── README.md
│   ├── config.json
│   ├── fabfile.py
│   ├── project.clj
│   ├── src
│   │   ├── bolts
│   │   │   ├── __init__.py
│   │   │   ├── parse.py
│   │   │   └── wordcount.py
│   │   └── spouts
│   │       ├── __init__.py
│   │       └── tweets.py
│   ├── tasks.py
│   ├── topologies
│   │   └── tweetwordcount.clj
│   └── virtualenvs
│       └── wordcount.txt
├── README.md
├── Twittercredentials.py
├── architecture.pdf
├── finalresults.py
├── histogram.py
├── install-dependencies.sh
├── make-db.sh
├── plot.png
├── plot.py
├── provision.sh
├── screenshots
│   ├── screenshot-finalresults-results-no-arg.png
│   ├── screenshot-finalresults-results.png
│   ├── screenshot-histogram-results-50-100.png
│   ├── screenshot-histogram-results.png
│   ├── screenshot-postgres-results.png
│   ├── screenshot-twitterStream-2.png
│   └── screenshot-twitterStream.png
└── twitter.sql
```