Exercise 2

Kyle Hamilton
Dec 6, 2015
W205-6

Application Use Case

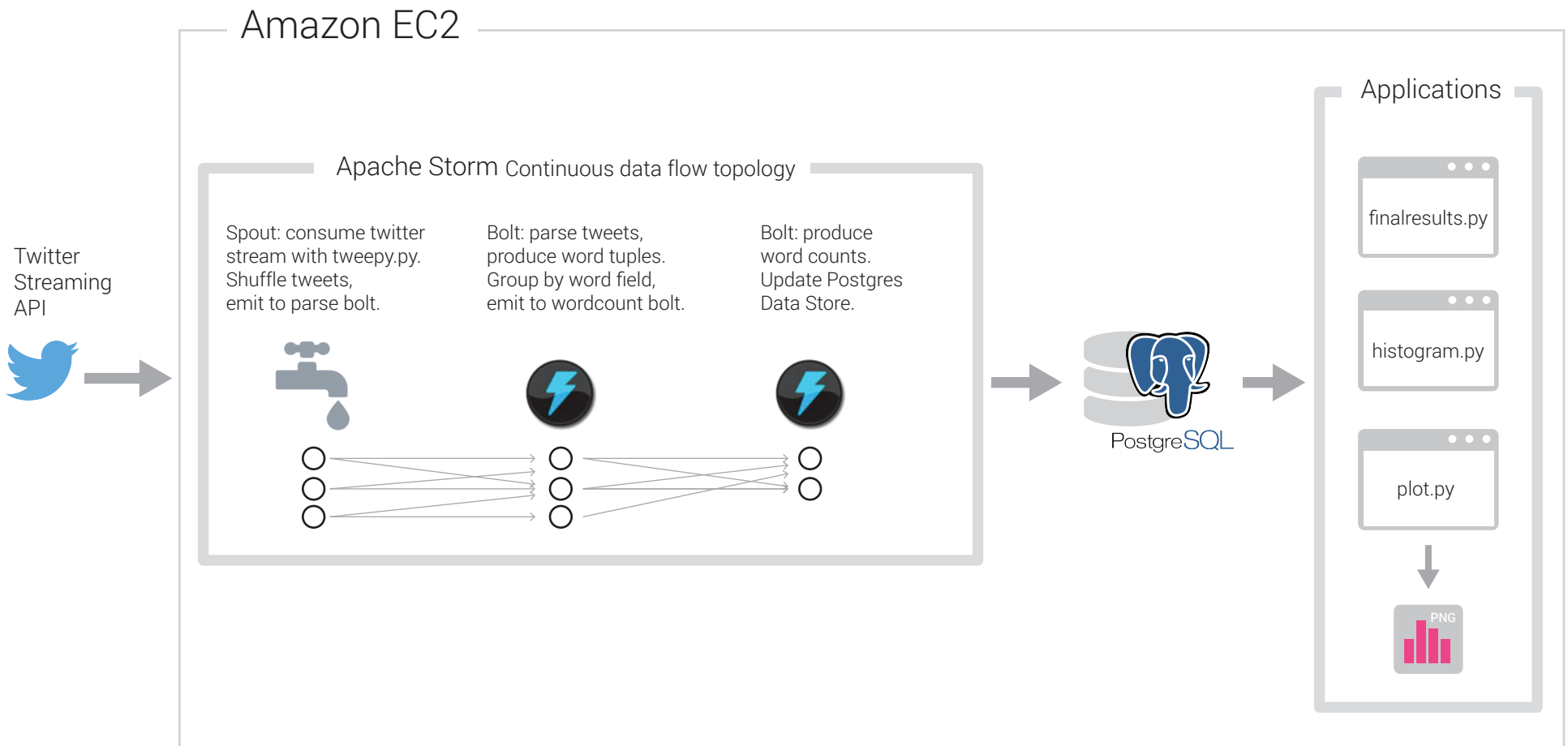
Graphing the counts of celebrities' names, politicians, ideas, etc., to measure and display their popularity over time.

Architecture

1. Apache Storm processes Twitter streaming API.
2. Tweet word counts are appended to a Postgres DB.
   (See diagram for details)

3. 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 from the DB.
   - **Historgram.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. Another way would be to simply specify an OFFSET and LIMIT in the query, but then there is a possibility (albeit small) that we miss something.

Exercise 2 - Architecture Diagram



Running the Application

Requirements:

AMI
ucbw205_complete_plus_postgres_PY2.7
IMPORTANT: python 2.7 will be run in a virtualenv. Lein will also be installed in the appropriate location, namely /usr/local/bin/lein. This is important! Do not use the python 2.7 installation instructions in the exercise, as that method breaks the necessary components to install some dependencies.

Alas, what if you've already got python 2.7 and lein running according to the exercise instructions. In order to install matplotlib, you have to jump through a couple of hoops first. Do:

```
mkdir /usr/local/lib/tmp_h/
mv /usr/local/lib/libpython2.7.a /usr/local/lib/tmp_h/
pip install --no-cache-dir matplotlib
mv /usr/local/lib/tmp_h/libpython2.7.a /usr/local/lib/
```

I really hope it doesn't come to that ^^

Volume
A brand new **m3.large**
(if you are using a pre-existing volume and you already have postgres running, you will get the option to skip the postgres setup)

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

Postgres
Python 2.7
lein

Python modules
matplotlib
streamparse
psycopg2
argparse
numpy
tweepy

Steps to install dependencies and run application

1. **Once you have mounted /data on your volume, and set your github keys:**

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

2. **Install dependencies.**

You will be prompted to confirm whether or not postgres is set up on /data/pgsql (if you attached a brand new volume, the answer is no). You will also be prompted to enter your twitter credentials.

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

3. **Start the application:**

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

4. **Open a new shell, and run applications:**

   ```
   source  ~/27env/bin/activate

   python /data/ucb205/exercise_2/histogram.py [min] [max]
   python /data/ucb205/exercise_2/finalresults.py -w [word]
   python /data/ucb205/exercise_2/plot.py
   ```

5. **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
├── 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
├── set-twitter-keys.sh
└── twitter.sql
```