Kasane Utsumi – 205 Assignment 2 (Revised)

What was revised in this version:

1. I improved the Architecture Design document (see below).
2. I improved the resiliency of the program by adding try..catch statement in critical flows that are most likely to fail (for example, twitter api call, S3 connection/call and opening a file). Also I added interrupt handling as well.
3. I added filtering for URLs & usernames before running analysis.
4. I added header and short comment to each python files to improve the code legibility. I also added comments to any part of the code that was unclear.

   Bucket Location: https://moonlightbucket.s3.amazonaws.com/ (Please ignore files under db_streamT and db_tweets folders. Those are for assignment 3)

How to retrieve tweets with the keywords and store on S3

1. Command format: python retrieve.py “startdate” “enddate”

2. In the beginning, I had one script to get tweets for the entire week. However, my Linux VM had numerous issues, including 1)KILLED message showing up 2)running out of disc space. Thus, I turned “start” and “end” date into command line arguments so I can query a day’s worth of tweets each time. This way, if retrieval failed, I only have to rerun the retrieval for the specified day. The downside of this is that I had to run the command manually 7 times:

   ```
   python “2015-02-07” “2015-02-08”

   python “2015-02-08” “2015-02-09”

   python “2015-02-09” “2015-02-10”

   python “2015-02-10” “2015-02-11”

   python “2015-02-11” “2015-02-12”

   python “2015-02-12” “2015-02-13”

   python “2015-02-13” “2015-02-14”
   ```

3. These are the python files used to run this part of the assignment:

    a. retreive.py - Retrieves tweets with hashtag #Microsoft or #Mojang via twitter api and stores them in a file (500 tweet in json format per file) and uploads each file to S3 bucket. It takes two command arguments: Start date to start twitter collection and stop collection BEFORE the specified end date.

    b. tweetserializer.py - A utlity class that takes care of keeping count of tweets per file, and uploads file to S3 when the quota has been met.

How to analyze tweets on S3 and create histogram

1. Run “python analyze.py”
2. First I tried to create histogram from entire data. However, my VM kept giving me a “KILLED” error (http://stackoverflow.com/questions/19189522/what-does-killed-mean-in-python). Therefore, I decided to make a histogram only for words with more 1000 counts. Please see “histogramForOver1000Counts.png”.
3. These are the python files used to run this part of the assignment:
    a. analyze.py - Retrieves a dictionary of frequent word and displays in a histogram.
    b. tweetanalyzer.py - A utility class that is used to retrieve twitter files from S3 and create a dictionary of frequent words.