W205-1 Spring 2016

Alfred W. Arsenault

# Exercise 2 – Architecture

This paper summarizes the architecture of my Exercise 2 implementation.

## Streamparse:

The streamparse implementation consists of a topology file, tweetwordcount.clj, and the source files for the spouts and bolts.

There is one spout, tweets.py, which monitors my Twitter application (see below) and retrieves tweets. It passes these tweets to the parse bolt.

There are two bolts in the system, parse.py and wordcounts.py. Parse.py takes the tweets from the spout and splits them up into words. Undesired "words" such as RT (for "retweet") are discarded. The cleaned words are then passed to the wordcounts bolt.

The wordcounts bolt takes words in from the parse bolt, and updates a database of wordcounts. If this is the first time that this word has been seen, the word is inserted into the database with a count of "1". If this word has previously been seen, its count entry in the database is incremented by 1.

When the program completes, there is a database of words that have appeared in tweets, along with a count of how many times each word has appeared.

## Twitter app:

The Twitter application reads tweets from selected people and allows them to be captured by the streamparse implementation.

## Results:

The two programs, finalresults.py and histogram.py, can be used to show the results of the streamparse implementation.

## Database creation:

The TCount database is created from the command line, using the command

$*createdb –U postgres -e -h localhost –p 5432 TCount*