**Twitter Analytics Web Service Project Report**

* ***Front end (Tianwei Li)***

Front end submission requirements:

* 1. The design of the front end system
  2. The code to build front end system
  3. The type of instances used and justification
  4. The number of instances used and justification
  5. Other configuration parameters used
  6. The cost per hour for the front end system
  7. The total development cost of the front end system
  8. The throughput of your front end for the given workload of q1 queries
* ***Back end (Tao Yu)***

* The design of the back end system

sudo apt-get install libmysqlclient-dev mysql-client

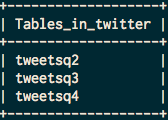
sudo apt-get install mysql-server

MySQL

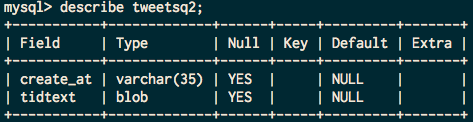
Innodb performance optimization

Warm up the database

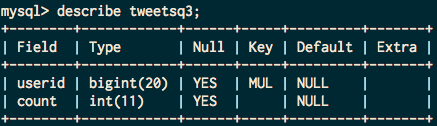
* The table structure of the database, justify your design decisions
* 3 tables for each query



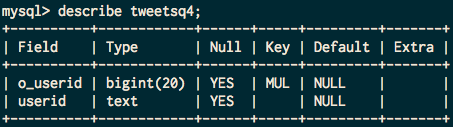
* Table tweetsq2 for the query 2 (Text of tweets)



* Table tweetsq3 for the query 3 (Number of tweets)



* Table tweetsq4 for the query 4 (Who retweeted a tweets)

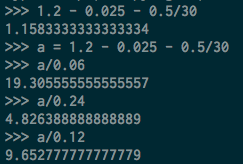


* 1. The type of instances used and justification
  2. The number of instances used and justification
  3. The cost per hour for the back end system
* $0.025 per Elastic Load Balancer-hour (or partial hour)

|  |  |
| --- | --- |
| m1.small | $0.060 per Hour |
| m1.medium | $0.120 per Hour |
| m1.large | $0.240 per Hour |

### Amazon EBS Standard volumes

* $0.10 per GB-month of provisioned storage
* $0.10 per 1 million I/O requests



* 1. The spot cost for all instances used
  2. The total development cost of the back end system
* ***ETL(Wendi Zhang)***
* ETL submission requirements:
  1. The code for the ETL job

s3://zwd199032/extra/\*.java

* 1. The programming model used for the ETL job and justification

We choose to use Amazon EMR tool to do the ETL job. The job flow is shown as below:



First we remove duplicate tweets in the sample tweets files, and use the result as input of data extraction process and extract data for different queries in the format which can easily load in database.

* 1. The type of instances used and justification

We choose to launch m1.large instance because at spot price, there's not huge differences between m1.large and m1.medium, since using m1.large can save much time, it's saving our money to some aspects.

* 1. The number of instances used and justification

We choose to launch 6 m1.large instances at a time because we found at this point each map-reduce job can be done within one hour, thus the cheapest approach.

* 1. The spot cost for all instances used

6 m1.large \* (0.026+0.06)$/h \* 4h = 2.064$

* 1. The execution time for the entire ETL process

The whole ETL process can be done within 4 hours. However, for some reasons problems like duplication of tweets , the format problem of query2 and definition of original user show up, we have to redo the ETL process again and again, so it might take about 20 hours for entire ETL process.

* 1. The overall cost of the ETL process

6 m1.large \* (0.026+0.06)$/h \* 20h = 10.32$

* 1. The number of incomplete ETL runs before your final run

About 50 times, if compile errors count.

* 1. Discuss difficulties encountered

The difficulty is in the design process, when we have to decide which kind of interface I'm going to provide to for further use, the rest part is the simple coding job. Also there's one thing to notice is that if we use json object in java, it will automatically convert encoding characters in text field to utf-8 characters, which is not we want. So in ETL of q2 we have to extract the text field by matching certain pattern.

* 1. The size of the resulting database and reasoning
  2. The time required to backup the database on S3

The size of backup file is about 5GB, so assuming the average uploading speed is 20MB/S, it will take 5\*1024/20 = 256 seconds

* 1. The size of S3 backup

5GB.