CSC591-002(DIC) Project Overview

Project

Team number: BetterCallSaul
Project title: Text Search Engine augmented with a Real-Time classifier.
Project description:
A scalable search engine on text documents with constant query latency. It will handle a large number of user queries and return the relevant documents. The application also uses a Machine Learning model to classify incoming news articles into relevant categories to improve search results.
Team members:
Amit Kanwar, Ashis Kumar Sahoo, Satvik Shetty
{akanwar2, aksahoo, smshetty}@ncsu.edu

Deliverables

Application Deliverables
1. A web-application that accepts user queries and returns text articles matching the user query in constant time.
2. A machine learning model that classifies incoming news articles in real-time and also updates the learning model.
3. Displaying trending search topics.

Performance Guarantees
1. **Volume:** The application adds all new incoming articles in constant time. ElasticSearch scales to other nodes in the cluster.
2. **Velocity:** It handles all incoming queries in constant time. Query latency to user remains constant.
3. **Reliability:** The application continues to run if a node fails.

Dependencies

1. **Platform Requirements**
- AWS EMR - To train classifier on text articles
- AWS Elastic Search – Database to store/index text/articles
- AWS Kibana – Reporting query statistics
- AWS Kinesis – Queuing user queries
- AWS Elastic Beanstalk – Hosting the web application
  OR
- At least 4 VCL clusters with above software stack (ElasticSearch, Kibana, Kafka, Local App hosting)

2. **Data Requirements**
- LexisNexis Academic Database
- Guardian News Article Corpus

Issues

AWS free tier limitations:
1. At max 2 nodes of t2.micro are allowed per instance
2. RAM is limited to 1GB and Storage is limited to 8GB per node

We need at least t2.large/t2.xlarge to train our model but that may drain the available credits quickly.