Movie Recommendation System

CS9223D Cloud Computing
Professor Sambit Sahu
Fall 2016

Anirudh (abm491), Anubhav (ass518), Chiquita (crp380), Shashank (sg4437), Parteek (psk287), Aditya (ak5656), Ankit (ab6530), Aayush (aa4960), Ganapathy (gsa277), Karthik (vst216), Kashish (kd1651), Ravi (rar555)

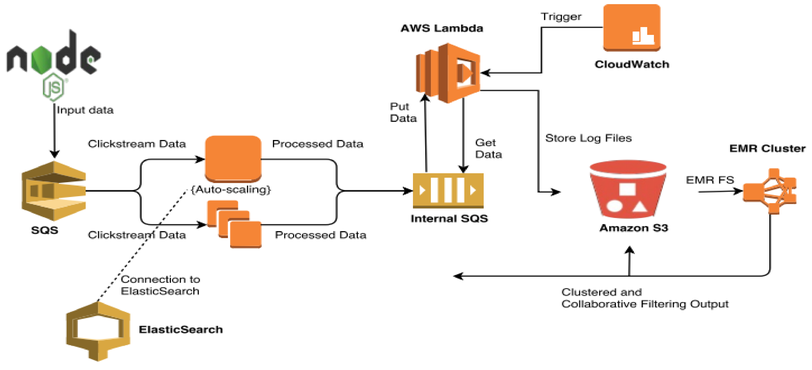
Goal

A scene based movie recommendation system based on movies from TMDB that uses clickstream data processing for effective collaborative filtering and deep learning.

Workflow

- The project is split into 4 main categories.
- App Server
- Audience tagging/Clustering service
- Content Tagging service
- Recommender service

Audience tagging and clickstream data processing



Clickstream data

- Designed for a fast read-lazy write system.
- Processes all clickstream data including debouncing & dedouping.
- Stores the processed data in another internal SQS for speed.

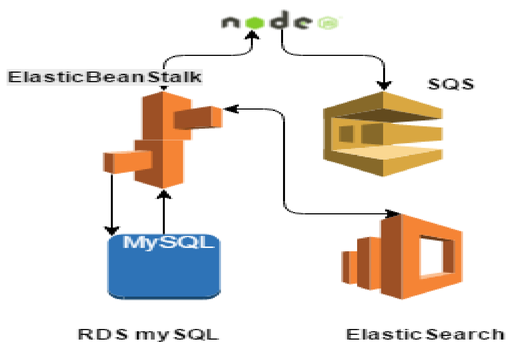
Logging data

- An independent lambda service deques the internal sqs to write logs to S3 bucket.
- This log serves as input to Clustering users for collaborative filtering recommender service.

Clustering in spark

- A spark job that runs in EMR cluster environment filters all logs for movie clicks.
- Uses Kmeans clustering for 27 genres and 27 clusters.
- The resultant rdd is passed for collaborative filtering.

App Server architecture



App Server - tasks

UI works

- Users can login/signup to this web application and stream movies.
- Automatic movie recommendations are provided to users.
- Users can also browse movies based on genres/keywords.

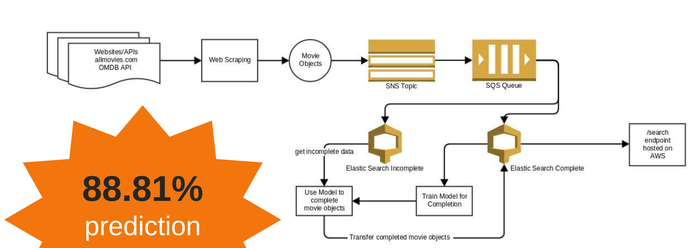
User meta data

- The main meta data is the users clickstream data catagorised into events. Each of these events are published to a SQS queue as json.

Dependent on services

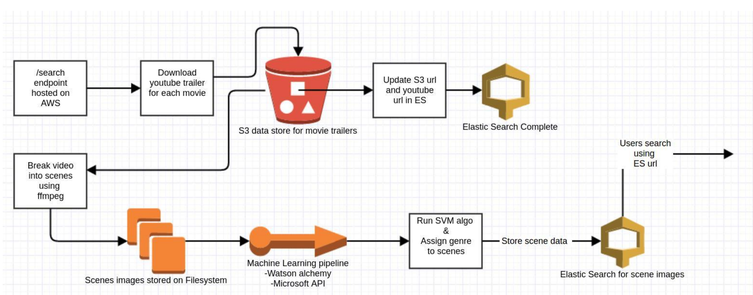
- The recommendation of movies depends on the call issued to recommender service that spontaneously gives a set of movies.
- Uses the tags generated by content tagging service to browse movies.

Content tagging phase I

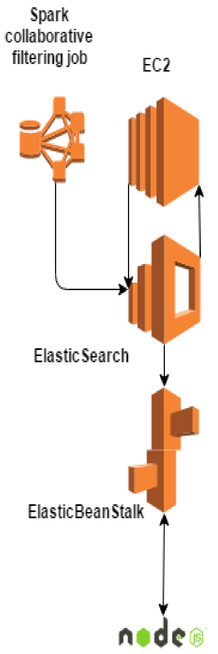


88.81% prediction accuracy

Content tagging phase II



Recommender service



- Uses the clustered data to perform collaborative filtering ALS algorithm on each cluster and update the dependencies in ElasticSearch. Using clustered data reduces computational time without much loss of accuracy.
- Also uses the graph algorithm to find similar movies . Uses the genres and other keywords provided for each movie to create a graph, connecting all movies. The degree of connection/ the number of links between each movie tells the similarity between movies.
- Communicates with frontnd to provide spontaneous responses for recommendation requests received.

- **Algorithms used** -
- Collaborative Filtering
- Connected components in a graph

Different use cases for recommendation

- When new user / old user signs in
- When new user / old user clicks a movie
- When new user / old user clicks a genre

Conclusion

- In this project, a prototype system for a scene based movie streaming and recommendation system was built on a sample set of data. The project covers a wide area of web development, machine learning and real time big data processing which helped us to learn new real world nuances and contribute a working model. We thank professor Sambit Sahu for giving us this opportunity to learn and work parallel in a real worl system.