CmpE 273 Project

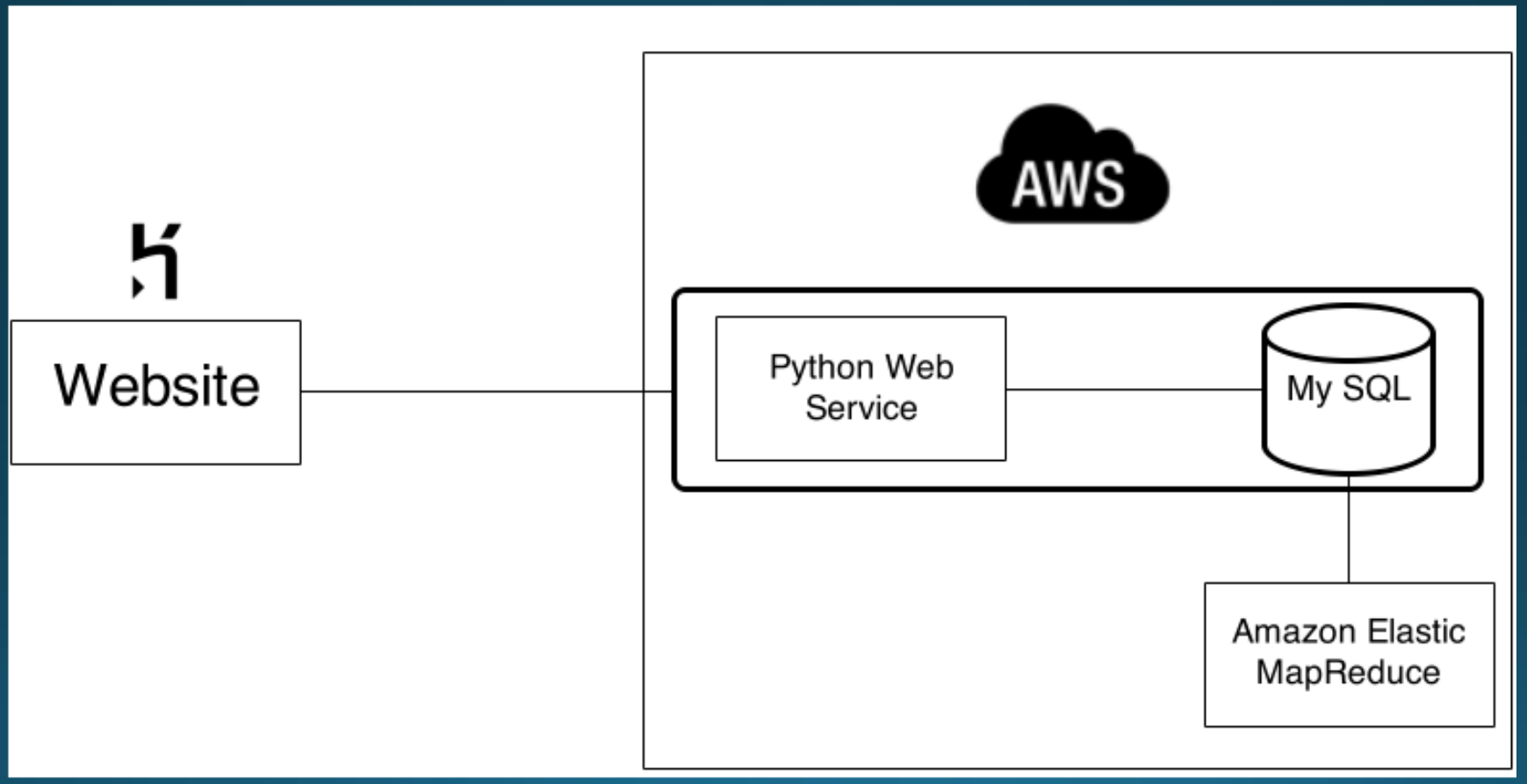
last.fm content dashboards

Bhushan Deo | Gaurav Bhardwaj | Prabhu Siddharth Raveendran | Krishna Chaitanya Dwarapudi

Project Demo

- What is last.fm?
- Last.fm data sets:
  1000 users data set: listening history for each user
  360,000 users data set: top 50 artists for each user
  Source:
  http://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/
- Business case: to create a web application which queries a data set and creates a visually pleasing dashboard.
- http://mysterious-wave-7118.herokuapp.com/
- Code at https://github.com/cmpe273project/lastfmcontentdashboards

Basic Architecture



Web Services

- Built using Python and Bottle Web Framework
- MySQL connector for Python
- Endpoints:

/1K → Track Analysis

/360K → Album Analysis

/usersimilarity → User Similarity

/networkanalysis → Network Analysis

Database Design

Schema:

main_1k : 19150868 tuples

userid | timestamp | artid | artname | traid | traname

profile_1k : 992 tuples

userid | gender | age | country | registered

main_360k : 17559530 tuples

usersha1 | artmbid | artname | plays

profile_360k : 359347 tuples

usersha1 | gender | age | country | registered

Database Design

- Started off in the wrong direction with MongoDB.

main_360k : 17559530 tuples

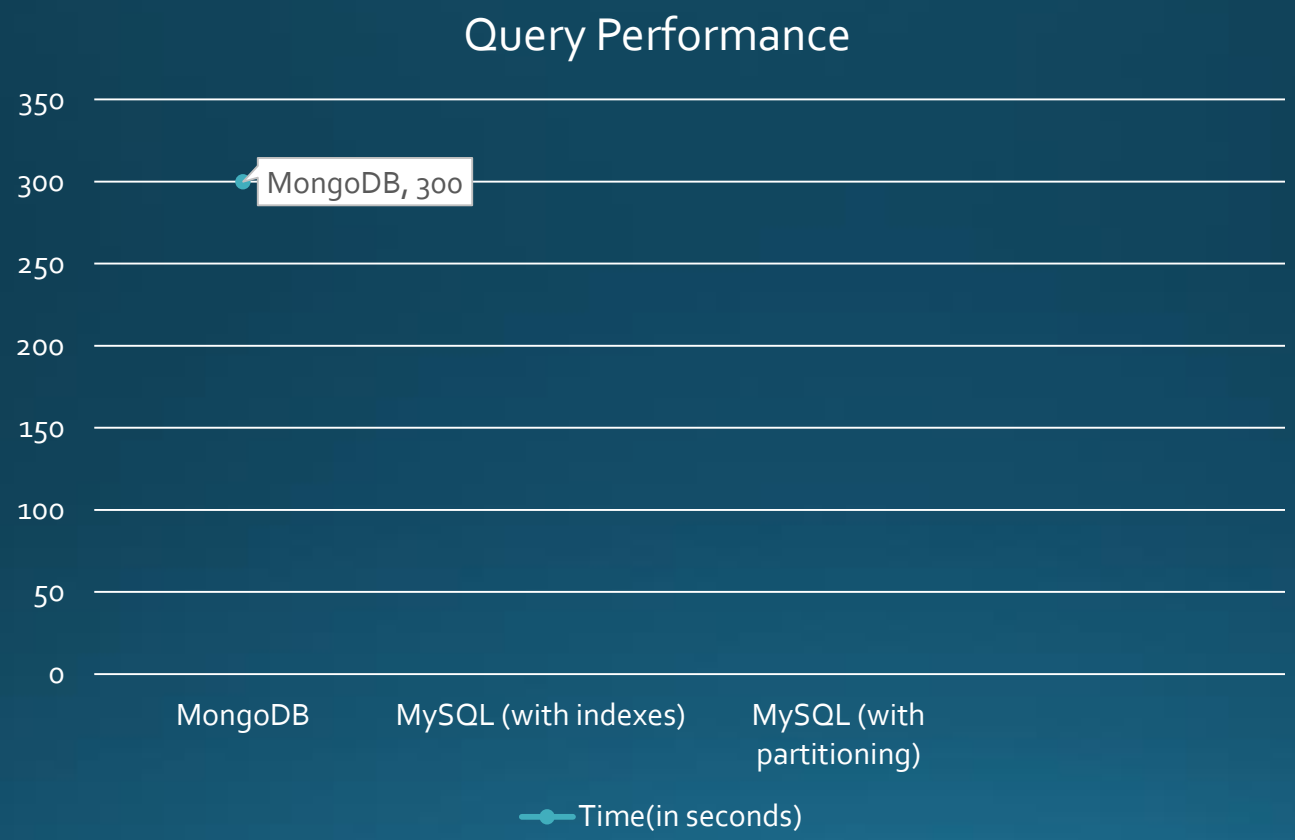
usersha1 | artmbid | artname | plays

profile_360k : 359347 tuples

usersha1 | gender | age | country | registered

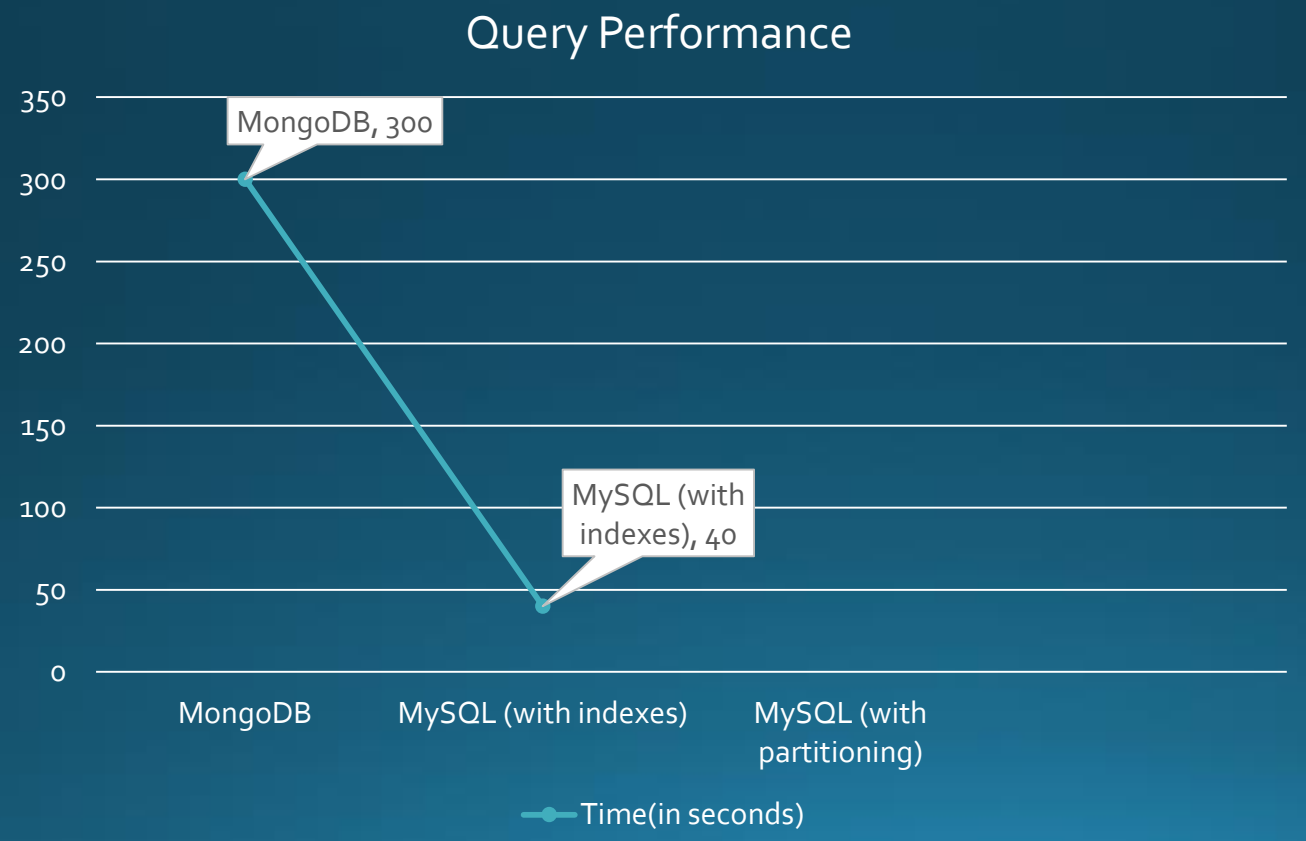
- Thus, due to structure of the data set, we needed joins, but in MongoDB we have to do "joins" programmatically which is inefficient compared to RDBMS.
- Ex: If for an artist "the beatles", we get 40,000 users from the first table, we have to make 40,000 queries to the other table to get the gender, age and country.

Database Design



Database Design

- MySQL: With proper indexes on the fields which we are querying, query time comes down to 30 seconds.



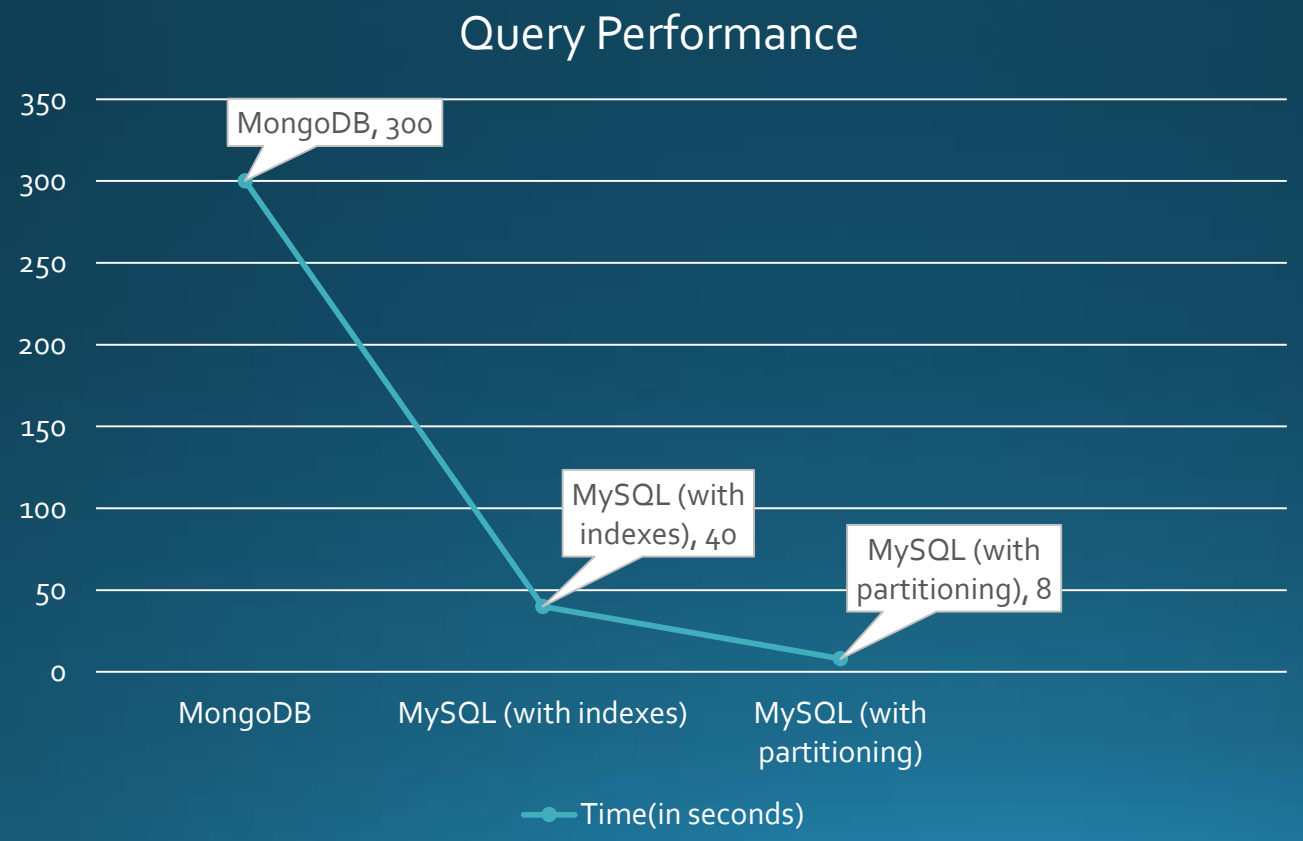
Database Design

- MySQL: To improve performance further, we used partitioning. (based on KEY – MySQL's own internal hashing function.

PARTITION_NAME	TABLE_ROWS
p0	2112318
p1	1571805
p2	1738005
p3	1687585
p4	1776609
p5	1624861
p6	1858028
p7	1684786
p8	1850821
p9	1670825

Database Design

- MySQL: With partitioning, query time comes down to 7-8 seconds in the worst case.



User Similarity

- Music compatibility between two users based on their common artists, and number of plays for those artists

usersha1 | artmbid | artname | plays

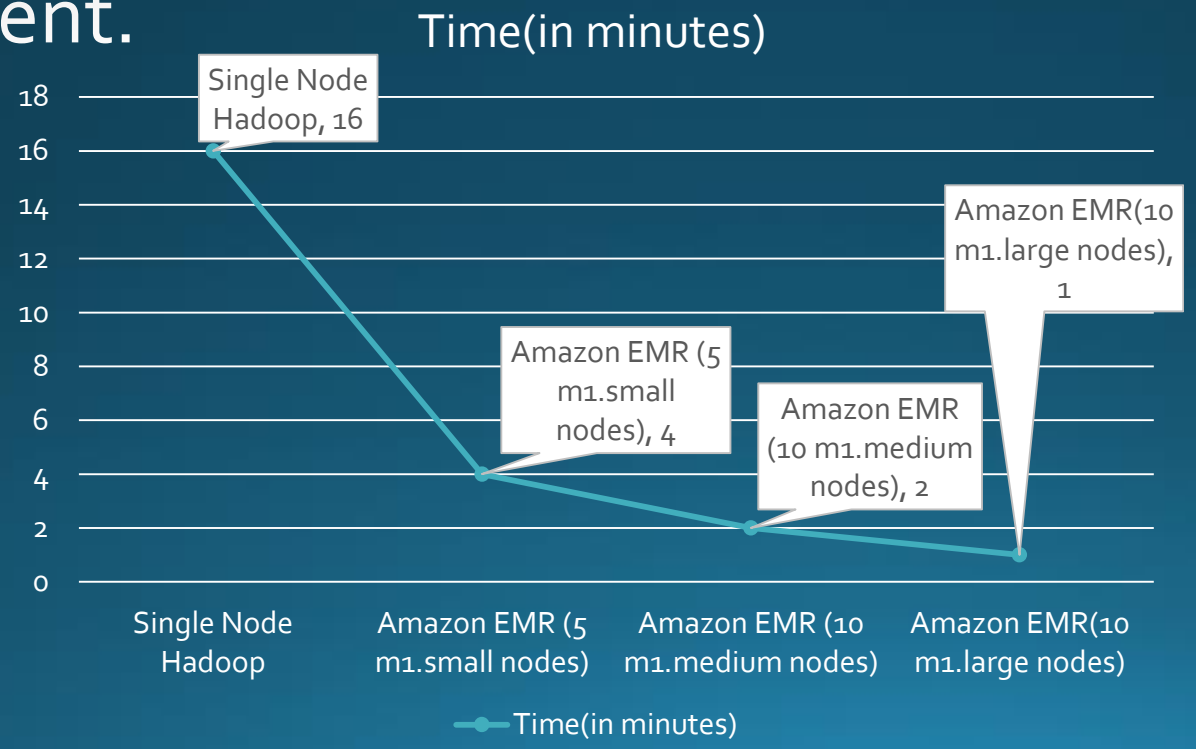
- Cosine Similarity to calculate the music compatibility between users

Entire Network Analysis

- Using Hadoop to come up with trends from the entire network:
  - Top 10 artists in the network
  - Top 10 artists by year
  - Count for usage (scrobbling by users), by hour, day and month.

Entire Network Analysis

- Used Amazon Elastic MapReduce(EMR) minimize the job time.
- Can be used to easily change the number of nodes as per requirement.



Front End

- Website hosted on Heroku platform.
- Foundation front-end framework, JQuery for UI.
- Canvas JS and HERE Maps API for visualizing data.