Report

Team Members:
1. Joe Khater
2. Aasim Malik

Use Case:

The project implements a movie recommendation system that utilizes three types of NoSQL databases: MongoDB, Neo4j, and Redis. The system is designed to recommend movies to users based on genres of movies they like and their relationships with other users.

Database Usage:

1. MongoDB:

MongoDB is used to store detailed movie information, including titles, descriptions, genres, cast, release dates, and reviews. Its document-based model allows flexible and scalable storage of structured data. Sample data includes:
- Movie details with genres and reviews by users.
- Queries to retrieve movies by genre or other attributes.

2. Neo4j:

Neo4j is used to model relationships between users and movies. The graph structure is ideal for traversing relationships to recommend movies based on user preferences and social connections. Key relationships include:
- `LIKES` relationships between users and movies.
- Genre-based recommendations to suggest movies with similar genres to those a user likes.

3. Redis:

Redis is employed as a caching layer to store frequently accessed data, such as user preferences and precomputed recommendations. This enhances performance by reducing database queries for repeated requests. Examples include:
- Cached favorite genres for each user.
- Precomputed movie recommendations for quick access.

Example Workflow:

1) A user (e.g., User 104) likes a movie with specific genres (e.g., Action, Drama).
2) Using Neo4j, movies with similar genres that the user hasn’t liked are recommended.
3) MongoDB retrieves detailed information about the recommended movies.
4) Redis caches the recommendations for User 104 for faster future retrieval.

This integration showcases how different NoSQL databases can complement each other to deliver a robust recommendation system.