Architecture Selection for API: Rest vs GraphQL

Team members: **Arsalan Imran (002344722), Guan Wang (002341590), Dan Luo (002352806)**

**Introduction**

the choice of API architecture can significantly impact the performance, scalability, and overall user experience of an application. The decision between Rest and GraphQL often poses a significant challenge while REST offers a well-established approach based on resource-oriented principles, GraphQL introduces a more dynamic and efficient method for fetching data. This project aims to provide empirical insights into the performance of REST and GraphQL APIs by evaluating key metrics such as error rates, throughput, response times, and load. We offer developers a comprehensive understanding of how each architecture performs in real-world scenarios.

**Objective**

The objectives of this project can be outlined as follows:

1. Performance Evaluation: performance evaluations of REST and GraphQL APIs under different scenarios.

2. Metric Analysis: analyze key performance metrics such as error rates, throughput, response times, and system scalability for both REST and GraphQL architectures.

3. Identify Use Cases: Identify specific use cases where REST or GraphQL architecture excels or faces challenges, providing developers with practical guidance on architectural selection.

4. Scalability Assessment: Assessing scalability of REST and GraphQL APIs by analyzing ability to handle increasing numbers of concurrent users and data requests.

Methodology:

Dataset Creation: We compile a dataset with comprehensive attribute including Api type, catagory,labels/name of samples, average response time, category, samples, errors, median, min response time, max response time, 20th percentile, 90%, 95%, 99% Line (Percentiles), Throughput, KB/sec (Kilobytes per second).

Type: Api type either Rest or GraphQL.

**Analysis:** We will conduct a comprehensive analysis after we compare the performance of apis using the dataset and statistical analysis. we plan to do three data anaylses

* Multivariate Regression with User Count:

(a)Understand how user count affects API performance(e.g., error rate, average response time, throughput).

(b)Determine which API handles varying user loads more effectively.

* Comparative EDA Analysis of Two APIs' Categories

(a) Understand the distribution and relative importance of categories for each API in the dataset.

(b) Reveal similarities or differences between the categories of the two APIs, guiding strategic decisions and market positioning.

* Significance Testing for Error Rate(e.g., t-tests, Wilcoxon signed-rank tests):

(a)Identify if one API has significantly lower error rates.

(b)Choose the more reliable API with fewer operational issues.

* Significance Testing for Throughput (e.g., t-tests, Wilcoxon signed-rank tests):

(a) Ascertain if there are significant differences in throughput data between the two APIs

(b) Determine which API performs better in handling workload or processing capacity

* Significance Testing for Average Response Time (e.g., t-tests, Wilcoxon signed-rank tests):

(a)Assess if one API consistently responds faster.

(b)Select the API that provides better user experience and engagement.

Based on the results of the correlation test (which we haven't done yet due to lack of data), for the last three tests, we'll probably only select part of them do finish.