

Case Study: Cost-Effective Performance Testing for a Healthcare Mobile App (Android & iOS) Using RedLine13 and JMeter at QualiLabs

Client Overview

A leading healthcare provider engaged QualiLabs to perform performance testing on their mobile app for both Android and iOS platforms. The app, backed by REST and GraphQL APIs, facilitated key services such as telemedicine, medical record access, and appointment scheduling. The client sought a cost-effective solution to ensure their app could handle growing user traffic and maintain optimal performance during peak times.

Challenges

- The app needed to support thousands of concurrent users efficiently.
- REST and GraphQL APIs required extensive performance testing to ensure optimal speed and reliability.
- The client requested a **cost-effective solution** that minimized cloud infrastructure expenses while ensuring comprehensive testing.

Objectives

- Simulate heavy user loads to evaluate the app's performance.
- Identify bottlenecks in the app and backend APIs.
- Provide detailed performance metrics, ensuring the solution remains budget-friendly and scalable.

Solution

Solution by QualiTlabs: Performance Testing with RedLine13 and JMeter

1. Test Planning and Setup

At QualiTlabs, we utilized **RedLine13** and **JMeter** to conduct large-scale performance testing. RedLine13 integrates seamlessly with **our own AWS infrastructure**, allowing us to scale the tests in the cloud while ensuring cost-efficiency. By leveraging AWS, we could simulate thousands of concurrent users in a flexible, secure, and scalable environment, minimizing the need for additional infrastructure.

2. Key Metrics Captured

- **Response Time:** Measured for both REST and GraphQL APIs, including average, minimum, and maximum response times.
- **90th, 95th, and 99th Percentile Response Times:** Provided insight into the majority of users' experience under high traffic.
- **Throughput:** Evaluated the number of requests processed per second, helping assess the system's capacity.
- **Top 10 API Endpoints with the Highest Response Time:** Identified the most time-consuming API endpoints for targeted optimizations.
- **Top 10 API Endpoints with the Highest Error Rates:** Pinpointed problematic endpoints with frequent errors.
- **Error Count and Types:** Recorded detailed information on errors for root cause analysis.
- **CPU and Memory Usage:** Monitored resource utilization to ensure efficient use of cloud resources.
- **Data Transfer Rate:** Captured the amount of data transferred, ensuring bandwidth efficiency.
- **Latency:** Measured the time data requests travel between the app and backend servers.
- **Root Cause Analysis:** Detailed analysis to identify and resolve any underlying performance issues.

3. Cost-Effectiveness with RedLine13 and AWS Integration

RedLine13's integration with **AWS** allowed us to perform cloud-based load testing efficiently and cost-effectively. This solution provided significant savings compared to traditional infrastructure-heavy methods. Here's how our approach ensured cost-effectiveness:

- **Cloud Scalability:** We ran tests on demand through AWS, eliminating the need for expensive hardware investments. This allowed the client to benefit from high scalability without long-term infrastructure costs.
- **Pay-As-You-Go Model:** The AWS integration allowed us to scale our load tests dynamically and only pay for the resources consumed, ensuring maximum cost efficiency.
- **Open-Source JMeter:** Leveraging JMeter, an open-source tool, reduced licensing costs while providing a powerful platform for comprehensive testing.

4. Execution and Findings

Our performance testing simulated various user load scenarios, delivering key insights:

- **API Performance:** REST APIs showed slower performance for large datasets, while GraphQL queries performed well, though certain complex queries needed optimization.
- **Bottlenecks:** The slowest API endpoints were linked to database queries, leading us to recommend optimization strategies.
- **Error Handling:** High-error endpoints were identified, allowing the team to address specific vulnerabilities.

5. Recommendations and Optimizations

Based on the findings, we recommended:

- **Optimizing database queries** to enhance performance for large dataset retrieval.
- **Implementing caching mechanisms** to improve API response times under heavy loads.
- **Refining complex GraphQL queries** to ensure faster and more reliable data retrieval.

6. Post-Execution Support

Upon completing the performance testing, we provided:

- **Daily Reports:** Tracking key metrics such as response times, throughput, and errors.
- **Detailed Final Report:** Summarizing findings, root cause analysis, and actionable recommendations for system improvement.

Conclusion

Through QualiTlabs' integration of RedLine13 with AWS and the use of JMeter, we provided the client with a scalable and cost-effective performance-testing solution.

This approach allowed the healthcare provider to:

- Identify performance bottlenecks and optimize their system for scalability.
- Ensure their app can handle high user traffic during peak times.
- Minimize costs through cloud-based infrastructure and pay-as-you-go pricing without compromising the quality or depth of the analysis.