# Performance Assessment Scheme

FDM06

# Contents

# 1. Background

The main areas of concern for web application performance are concurrency and latency, through stress testing and load testing, respectively. These metrics provide a great initial insight into how the application performs, both from a user experience (UX) and stability perspective.

In relation to the chosen technical design approach, load testing the architecture will determine the impact of microservices and how the additional network traffic affects wait times for the client. However, the results will not be completely synonymous with a production environment as all services will be hosted on the same machine, meaning all network traffic will occur on a single network interface. On the other hand, this will be a good test for microservice architecture while not taking advantage of its many benefits, such as horizontal scalability and distributed services.

# 2. Benchmarks

## 2.1. Concurrency

Based on the top 20 sites worldwide in 2021 (Clement, 2022), there is a range of around 1.22 to 45.41 billion visits on these sites each month. Considering the scale of these platforms, it would be foolish to assume the architecture will support anything close to these numbers. However, using the lower boundary in this range as an upper benchmark for our test cases should provide a decent start for exploratory testing.

We could then make a naïve assumption that user sessions on our website may last around 30 minutes, and all sessions are spread evenly throughout each 24-hour period. This would indicate the need to support around 850,000 concurrent users (1.22 billion / 30 *days* / 24 *hours* / 2). It is worth noting that sessions do not exist in stateless REST APIs, and therefore sessions in this context will test the server-side asynchrony.

## 2.2. Latency

Benchmarking response time is a difficult problem for web platforms as more complex behavior will naturally result in longer wait times, if left unoptimised. Completely up to interpretation, but 2500ms is a reasonable middle ground for complex interactions. Tracking for outcomes longer than this SLA should be included to determine which calls are taking even longer. It would be wise to re-visit the test schema, and hand pick SLAs for each call based on its technical complexity.

# 3. Test Scheme

The following test cases are exploratory and will be used as a basis to create statistics for subsequent analysis, and as such will not be determined by a pass / failure state. Each test is to be performed on each API endpoint that is exposed to, and utilised by, the client.

## 3.1. Stress Tests

| Test ID | User Count Threshold |
|---|---|
| ST10 | 10 |
| ST100 | 100 |
| ST500 | 500 |
| ST1K | 1,000 |
| ST5K | 5,000 |
| ST10K | 10,000 |
| ST50K | 50,000 |
| ST100K | 100,000 |
| ST250K | 250,000 |
| ST500K | 500,000 |
| ST750K | 750,000 |
| ST1M | 1,000,000 |

## 3.2. Load Tests

The results of load testing the application will be determined as an average from 3 separate calls to each endpoint.

| Test ID | Target Latency (milliseconds) |
|---|---|
| LT1 | 1 |
| LT2 | 2 |
| LT3 | 3 |
| LT4 | 4 |
| LT5 | 5 |
| LT10 | 10 |
| LT15 | 15 |
| LT20 | 20 |
| LT25 | 25 |
| LT50 | 50 |
| LT75 | 75 |
| LT100 | 100 |
| LT250 | 250 |
| LT500 | 500 |
| LT1K | 1,000 |
| LT2K | 2,000 |
| LT3K | 3,000 |
| LT4K | 4,000 |
| LT5K | 5,000 |

## References

Clement, J., 2022. Global top websites by monthly visits 2020 | *Statista*. [online] Statista. Available at: <https://www.statista.com/statistics/1201880/most-visited-websites-worldwide/>.