Test Target App

Version 1.3

Contents

[Functional Requirements 1](#_Toc112837632)

[Non-Functional Requirements (NFR) 1](#_Toc112837633)

[Performance Efficiency 1](#_Toc112837634)

[Test Design 2](#_Toc112837635)

[Solution 1 2](#_Toc112837636)

[Solution 2 2](#_Toc112837637)

[Test Report 3](#_Toc112837638)

[Test 1 - Locust 3](#_Toc112837639)

[Test 2 - JMeter 4](#_Toc112837640)

# Functional Requirements

TBD

# Non-Functional Requirements (NFR)

## Performance Efficiency

Typical usage of test-target by users according business model

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Operation | Services | Expected response time, s | Peak response time, s | Expected load, per hour | Peak load, per hour | Notes |
| Math Stuff |  | 3 | 6 | 15 000 | 30 000 |  |
|  | /fibonacci |  |  |  |  | Range [20, 25] |
|  | /sort |  |  |  |  | Range [15k, 20k] |
| Wait |  | 5 | 5 | 5 000 | 5 000 |  |
|  | /wait |  |  |  |  | 5 sec |
| Demo |  | 2 | 4 | 10 000 | 10 000 |  |
|  | / |  |  |  |  |  |
|  | /unstable |  |  |  |  |  |
|  | /sort |  |  |  |  | 10k |
| Main |  | 2 | 4 | 10 000 | 10 000 |  |
|  | / |  |  |  |  |  |

## Test Design

Notes

* Fortunately, operations do not have end-to-end data to be passed through endpoints
* Load need to be calculated in RPS
* Same endpoints in different operation can be combined

### Solution 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Services | Expected response time, s | Peak response time, s | Expected load, rps | Peak load, rps | Notes |
| /fibonacci | 1.5 | 3 | 4 | 8 | Range [20, 25] |
| /sort | 1.5 | 3 | 4 | 8 | Range [15k, 20k] |
| /wait | 5 | 5 | 1 | 1 | 5 sec |
| /unstable | 1 | 2 | 3 | 3 |  |
| /sort | 1 | 2 | 3 | 3 | 10k |
| / | 1 | 2 | 6 | 6 | Demo + Main |
|  |  |  |  |  |  |
| Total | - | - | 21 | 29 |  |

### Solution 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Operation | Services | Expected response time, s | Peak response time, s | Expected load, rps | Expected load, % | Peak load, rps | Notes |
| Math Stuff |  | 3 | 6 | 4 |  | 8 |  |
|  | /fibonacci |  |  | 4 | 19.05 | 8 | Range [20, 25] |
|  | /sort |  |  | 4 | 19.05 | 8 | Range [15k, 20k] |
| Wait |  | 5 | 5 | 1 |  | 1 |  |
|  | /wait |  |  | 1 | 4.76 | 1 | 5 sec |
| Demo |  | 2 | 4 | 3 |  | 3 |  |
|  | / |  |  | 3 | 14.29 | 3 |  |
|  | /unstable |  |  | 3 | 14.29 | 3 |  |
|  | /sort |  |  | 3 | 14.29 | 3 | 10k |
| Main |  | 2 | 4 | 3 |  | 3 |  |
|  | / |  |  |  | 14.29 |  |  |
| Total |  |  |  | 21 |  | 29 |  |

## Test Report

Environment metrics:

* Local network: wi-fi 5, speed (receive/transmit): 351/260 (Mbps)
* Network latency: 1ms
* Server:
  + Intel Celeron J1900 2.00GHz
  + 8 Gb RAM
  + Docker
* Connection: http

### Test 1 - Locust

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Operation | Services | Expected response time, s | Actual response time, s | Expected load, rps | Actual throughput, rps |
| Math Stuff |  | 3 | 0.4 | 4 |  |
|  | /fibonacci |  | 0.24 | 4 | 3.7 |
|  | /sort |  | 0.14 | 4 | 3.9 |
| Wait |  | 5 | 5.2 | 1 |  |
|  | /wait |  | 5.2 | 1 | 0.9 |
| Demo |  | 2 | 0.33 | 3 |  |
|  | / |  | 0.094 | 3 | 5.6\* |
|  | /unstable |  | 0.11 | 3 | 2.9 |
|  | /sort |  | 0.12 | 3 | 3.1 |
| Main |  | 2 | 0.094 | 3 |  |
|  | / |  | 0.094 |  | 5.6\* |
| Total |  |  |  | 21 | 20.1 |

* Test duration: 5 min
* CPU %: 20
* RAM usage: 0

Saturation point:

* 29 RPS
* Response time 1.1 sec

Chart, histogram

Description automatically generated

### Test 2 - JMeter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Operation | Services | Expected response time, s | Actual response time, s | Expected load, rps | Actual throughput, rps |
| Math Stuff |  | 3 | 0.27 | 4 |  |
|  | /fibonacci |  | 0.16 | 4 | 4.2 |
|  | /sort |  | 0.11 | 4 | 4.2 |
| Wait |  | 5 | 5.12 | 1 |  |
|  | /wait |  | 5.12 | 1 | 1 |
| Demo |  | 2 | 0.25 | 3 |  |
|  | / |  | 0.08 | 3 | 6.2\* |
|  | /unstable |  | 0.08 | 3 | 3 |
|  | /sort |  | 0.09 | 3 | 3.1 |
| Main |  | 2 | 0.08 | 3 |  |
|  | / |  | 0.08 |  | 6.2\* |
| Total |  |  |  | 21 | 20.8 |