

# Load Test Report

## v1.0

### Revision History

| Date            | Version | Description    | Author            |
|-----------------|---------|----------------|-------------------|
| Mon 15 Apr 2019 | v1.0    | Report version | Andrej Skeledzija |
|                 |         |                |                   |
|                 |         |                |                   |
|                 |         |                |                   |
|                 |         |                |                   |

## **Contents**

- 1. Introduction**
- 2. Test Case Specification**
- 3. Test Data and Requirements**
- 4. Results**

## 1. Introduction

Application Performance requirements could manifest themselves in a number of ways, however, these will boil down to three main points:

**Response time:** is the elapsed time from the moment when a given request is sent to the server until the moment when the last bit of information has returned to the client.

**Throughput:** is the number of requests per unit of time (seconds, minutes, hours) that are sent to server during the test.

**Concurrent Users:** The number of users the application can service while under load.

All of these artifacts are related to each other, for example, the Throughput should rise as the number of concurrent users increases (more users demanding more activity), however Throughput can also decrease with an increase in the number of users if the system becomes overloaded. Response time should remain constant but if Throughput and / or the number of concurrent users reaches a point at where the load overloads the system then response time will start to degrade. Mapping response time to the number of concurrent users is a very efficient way to evaluate whether the system is handling the load well or not.

## 2. Test case specification

**Project Name:** Module 3 - Performance test - Web App

**Test Design ID:** TD10

**Test Case ID:** TC01

**Test Designed by:** Andrej Skeledzija

**Test Priority (Low/Medium/High):** High

**Test Designed date:** 04/11/2019

**Module Name:** Server Performance: Response time - time from when a user enters a request until the first character of the response is received.

**Test Executed by:** Andrej Skeledzija

**Test Title:** Retrieve Home Page By 1000 Users and measure server response time

**Test Execution date:** 15/11/2019

**Description:** Load test needs to simulate 1000 users who will visit the homepage in a period of 15s. It is necessary to measure the response time before and during the load.

**Automated:** YES

**Pre-conditions:** Web application is completely stable and fully operational.

**Dependencies:**

**Reference:** TestDesignSpecification\_v1.0

## 3. Test Data and Requirements

- **Web App:** <http://newtours.demoaut.com/mercurywelcome.php>
- **Threads (user) count:** 1000
- **RampUp:** 1
- **Duration:** 15 (sec)
- **HTTP Request method:** GET
- **JMeter:**
  - Property file Reader - Environment
  - Property file Reader - Module
  - Thread Group
  - Sampler - HTTP Request
  - View Result Tree
  - Graph Results
  - Aggregate Report
  - Response times vs. Threads
  - Response times distribution

#### 4. Results:

**Response time before test run: 589 ms**

| Label        | # Samples | Average | Median | 95% Line | Min | Max | Error % | Throughput | Received KB/sec | Sent KB/sec |
|--------------|-----------|---------|--------|----------|-----|-----|---------|------------|-----------------|-------------|
| HTTP Request | 1         | 589     | 589    | 589      | 589 | 589 | 0%      | 1,69       | 26,99           | 0,23        |

#### During Test Run:

| Label        | # Samples | Average | Median | 95% Line | Min | Max  | Error % | Throughput | Received KB/sec | Sent KB/sec |
|--------------|-----------|---------|--------|----------|-----|------|---------|------------|-----------------|-------------|
| HTTP Request | 1000      | 5428    | 5118   | 9282     | 614 | 9553 | 10.200% | 99.81036   | 1449.54         | 12.08       |

**Throughput = 5988.622 / minute** means that **~99,8** Requests per Second are sent to the server.

**Average = 5428 ms:** This is the Average (Arithmetic mean  $\mu = 1/n * \sum_{i=1}^n x_i$ ) Response time of total samples.

**Deviation = 2526:** The **standard deviation ( $\sigma$ )** measures the mean distance of the values to their average ( $\mu$ ). It gives idea of the **dispersion or variability** of the measures to their mean value.

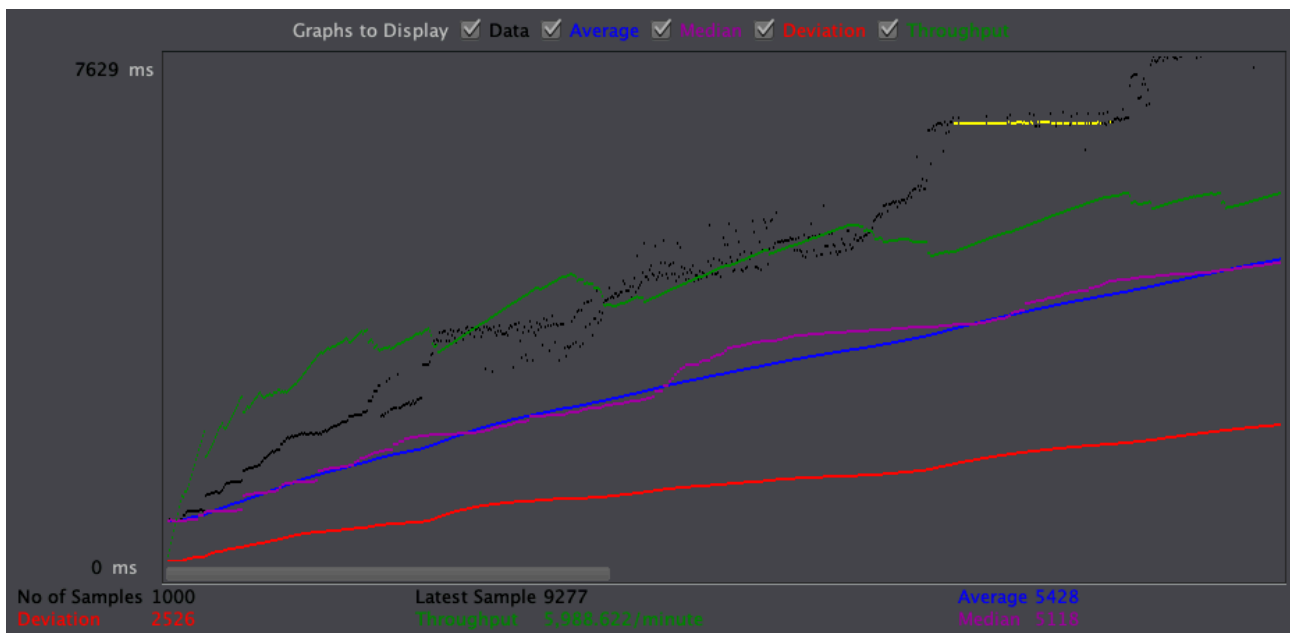
**Latest Sample = 9277ms:** This is the Latest samples **Response time (9277 ms)!!!**

**No of samples = 1000:** Total number of samples during the Test!

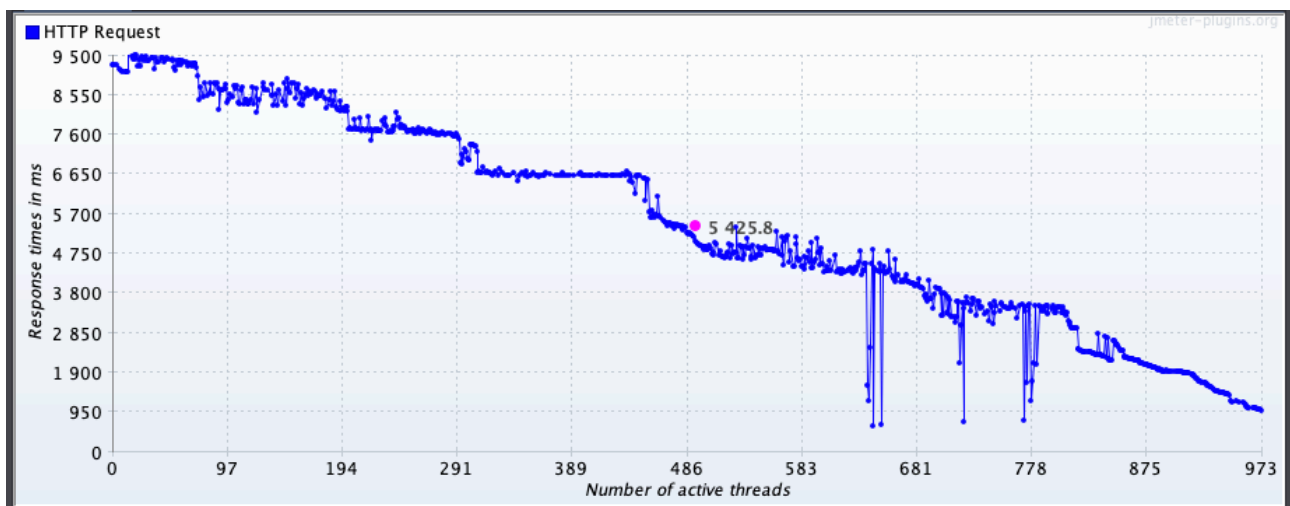
**Error:** 10% of samples was failed.

Graphical representation of results:

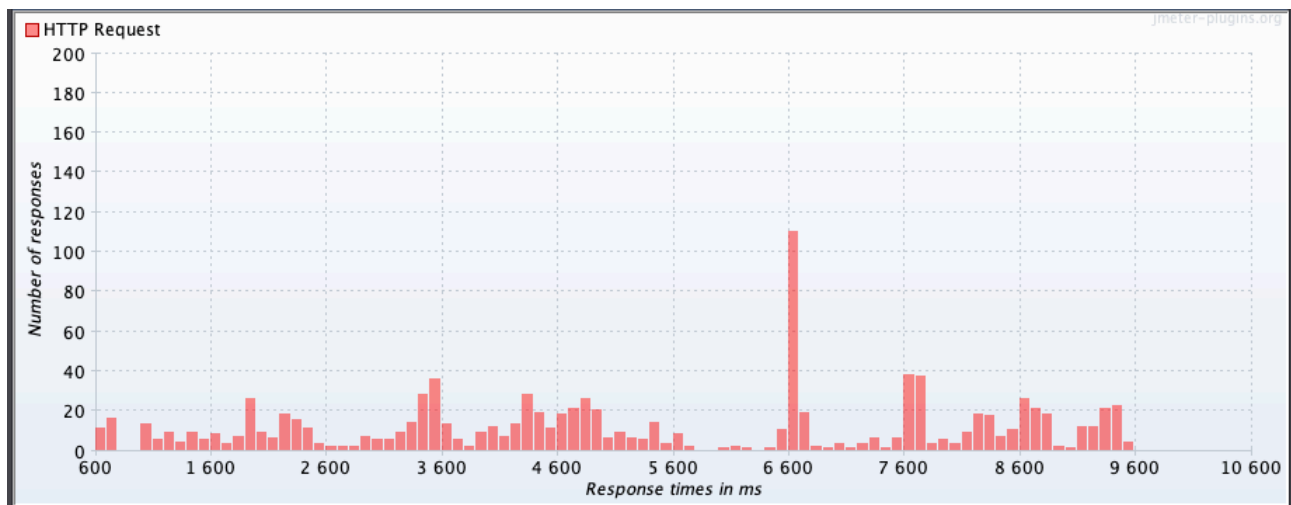
Throughput and Deviation:



Response Times vs Threads



## Response Times Distribution



### Summary:

Response Times and Latency can adversely affect negatively a user's experience, so it's important to try to achieve as optimal a performance as possible. Poor response time or high latency can result in






















- drop in product use
- poor product perception

The following are suggested best practices for acceptable response times and latency that all combine to deliver acceptable if not good user experience.

- **0.1 second** is about the limit for having the user feel that the system is reacting instantaneously, meaning that no special feedback is necessary except to display the result.
- **1.0 second** is about the limit for the user's flow of thought to stay uninterrupted, even though the user will notice the delay. Normally, no special feedback is necessary during delays of more than 0.1 but less than 1.0 second, but the user does lose the feeling of operating directly on the data.
- **10 seconds** is about the limit for keeping the user's attention focused on the dialogue. For longer delays, users will want to perform other tasks while waiting for the computer to finish, so they should be given feedback indicating when the computer expects to be done. Feedback during the delay is especially important if the response time is likely to be highly variable, since users will then not know what to expect.

Our Load test has shown that had an impact on web application response time, but still had a response time within acceptable limits, while at times not overloaded it had a response time less than 1 sec.

### Response analysis:

| Name   | × | Headers                     | Preview | Response | Timing    | » |
|--|---|-----------------------------|---------|----------|-----------|---|
|  mercurywel...      |   | Queued at 0                 |         |          |           |   |
|  html.gif           |   | Started at 10.20 ms         |         |          |           |   |
|  boxad1.gif         |   |                             |         |          |           |   |
|  banner2.gif        |   | Resource Scheduling         |         |          | TIME      |   |
|  featured_de...     |   | Queueing                    |         |          | 10.20 ms  |   |
|  hdr_special...     |   | Connection Start            |         |          | TIME      |   |
|  spacer.gif         |   | Stalled                     |         |          | 173.07 ms |   |
|  hdr_tips.gif      |   | DNS Lookup                  |         |          | 0         |   |
|  tip93.gif        |   | Initial connection          |         |          | 172.96 ms |   |
|  hdr_findflig...  |   |                             |         |          |           |   |
|  hdr_right.gif    |   | Request/Response            |         |          | TIME      |   |
|  btn_signin.gif   |   | Request sent                |         |          | 48 µs     |   |
|  hdr_destina...   |   | Waiting (TTFB)              |         |          | 284.92 ms |   |
|  icn_destinat...  |   | Content Download            |         |          | 176.17 ms |   |
|  hdr_vacatio...   |   |                             |         |          |           |   |
|  icn_vacatio...   |   | <a href="#">Explanation</a> |         |          | 644.41 ms |   |
|  hdr_register...  |   |                             |         |          |           |   |
|  icn_register.gif |   |                             |         |          |           |   |
|  hdr_links.gif    |   |                             |         |          |           |   |
|  logo.gif         |   |                             |         |          |           |   |
|  black            |   |                             |         |          |           |   |

Response GET <http://newtours.demoaut.com/mercurywelcome.php>



**Queueing.** The browser queues requests when:

- There are higher priority requests.
- There are already six TCP connections open for this origin, which is the limit. Applies to HTTP/1.0 and HTTP/1.1 only.
- The browser is briefly allocating space in the disk cache

**Stalled.** The request could be stalled for any of the reasons described in Queueing.

**DNS Lookup.** The browser is resolving the request's IP address.

**Proxy negotiation.** The browser is negotiating the request with a proxy server.

**Request sent.** The request is being sent.

**ServiceWorker Preparation.** The browser is starting up the service worker.

**Request to ServiceWorker.** The request is being sent to the service worker.

**Waiting (TTFB).** The browser is waiting for the first byte of a response. TTFB stands for

**Time To First Byte.** This timing includes 1 round trip of latency and the time the server took to prepare the response.

**Content Download.** The browser is receiving the response.

**Receiving Push.** The browser is receiving data for this response via HTTP/2 Server Push.

**Reading Push.** The browser is reading the local data previously received.