# SCALABILITY REPORT

Bakery App Starter for FW8 and Spring

## Introduction

The purpose of this document is to show how well Bakery App Starter web application scales out. We have performed scalability tests on it and the results are discussed in the document. We also provide recommendations for production setup of the web application.

## Test setup

Scalability tests were run on Mid 2015 MacBook Pro, 4 core (8 logical processors by hyper threading) 2.5 GHz i7 and 16GB of RAM. We used embedded Spring boot Apache Tomcat/ 8.5.15 server. On the same machine, we run MySQL 5.7.10 too. For optimal performance, we recommend running the database server on a separate server and using native libraries and performance tuning on the application server (see Summary for more information).

## Scalability on the CPU level

We measured the CPU usage and requests' response times by running the provided Gatling test (Barista.scala) on the same test machine where the Tomcat and MySQL were run. Performance was measured by running increasing number (1000, 2000, 3000) of virtual concurrent users with Gatling tool. We limited the maximum number of virtual users to 3000, since after at that point the CPU load of the MySQL and Gatling started limiting the available CPU resources for Tomcat. Due to the length of the recorded test case, the truly concurrent users count is lower than the virtual user count of the test. For example, the test with 2000 concurrent users, corresponds about 680 truly concurrent users using the application simultaneously.

Below is the CPU load of Tomcat server for 2000 and 3000 virtual users (Figure 1). Charts' x-axes represent elapsed time from the beginning of the test and y-axes represent CPU usage. The CPU percentage of 100% means that we are fully utilising one logical processor, when the (theoretical) maximum would be 800% when utilising all available logical processors.

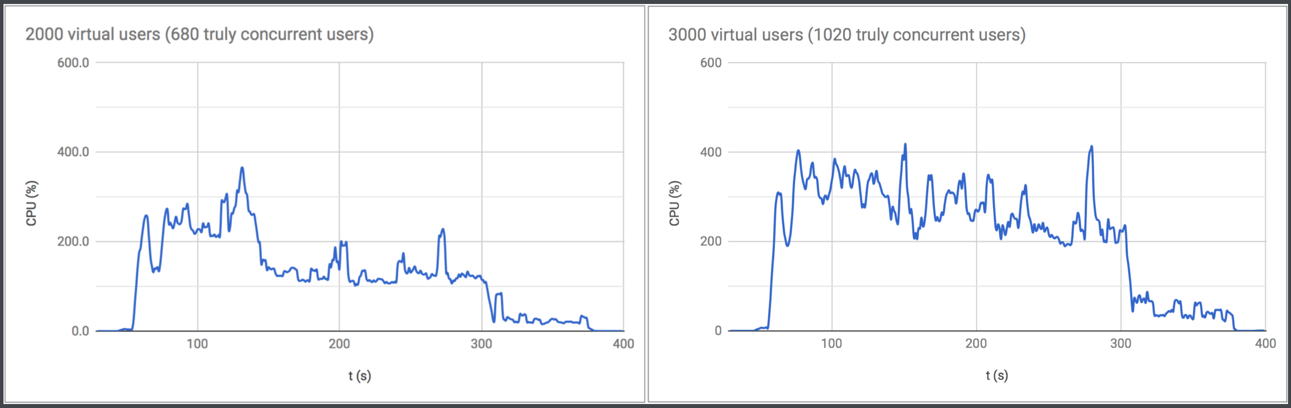

**Figure 1** The CPU usage during tests for 2000 (left) and 3000 virtual users (right).

We can observed that 1000 virtual users produced about 80-90% CPU usage. And similarly, 2000 virtual users consumes about two times as much CPU. Higher peaks in the graphs represents typically garbage collection events. As a rough estimate, we suggest to have at least one logical CPU core per every 1000 virtual users (340 truly concurrent users). For example, if you expect that your web application in the production, is constantly used by 2000 truly concurrent users, you should reserve at least 2000/340 = 5.89 ~ 6 logical (at least i7 level) cores for it. Another alternative is of course distributing the load by clustering (see Summary).

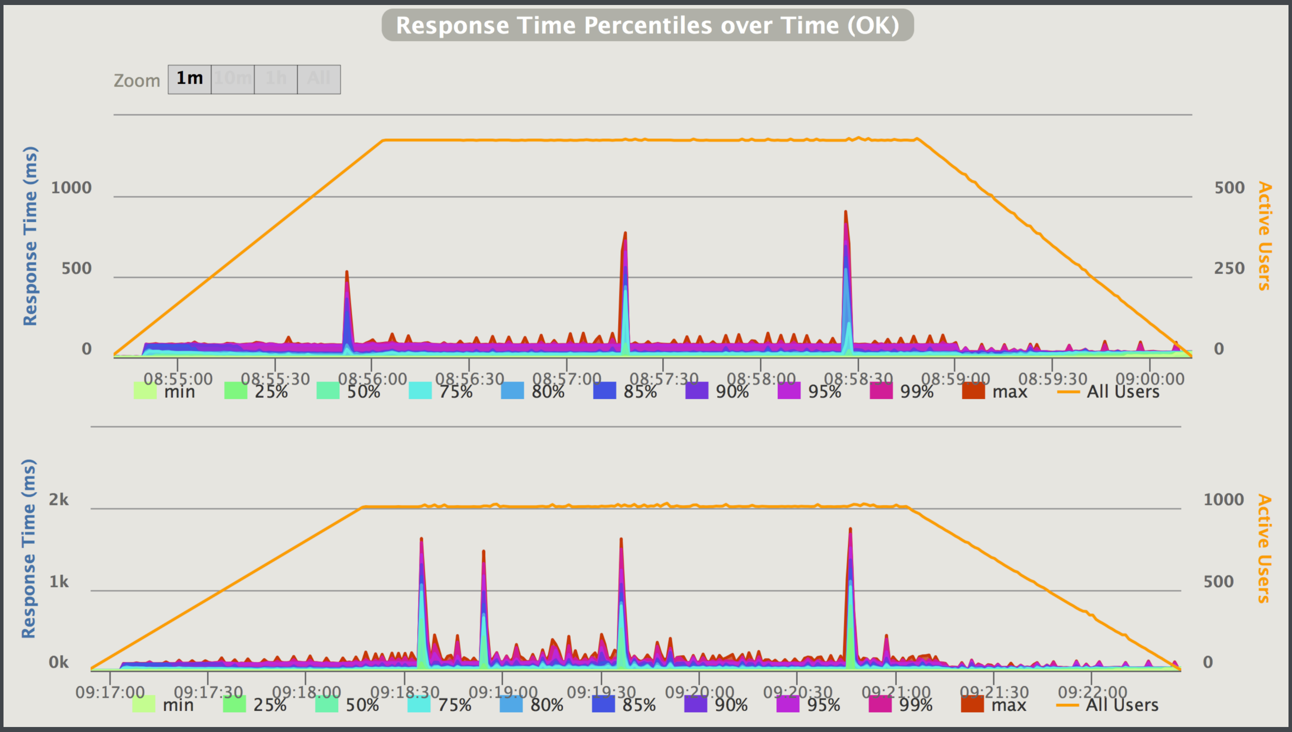

Figure 2 Response time percentiles over time for 2000 (above) and, 3000 virtual users (below).

In the Figure 2 (above) the response time percentiles over the duration of the tests for 2000 and 3000 virtual users are presented. We see that request's response times were low during tests except short moments during the expected garbage collect breaks. We recommend to use optimised garbage collector (such as G1 collector) for your production application server to minimise those peaks in the response times and CPU loads.

## The size of a session

Bakery App Starter consumes about 50 MB of memory on an idle use. The size of a session of Orders web application is about 210 kB. The size of a session can be used to estimate the memory need for a production server. For example if you want to serve 2000 concurrent sessions / server, you should reserve at minimum 0.21 MB * 2000 + 50 MB = 470 MB memory for it.

Though, to be on a safe side and enable optimal performance (without frequent garbage collection), we recommend reserving at least 1 - 1.5 GB of memory for every 1000 concurrent users. In addition, you should measure the session size every now and then when further developing the web application and making sure that it will not increase too much or adjust memory need of your production servers.

## Summary

Having a hyper threaded 4 core i7 server with 16GB of memory for application and database server, you should be able to serve constantly about 1000 concurrent users. Moving the database into separate server should double the maximum concurrent user count to about 2000.

You should be able to further increase the concurrent user count by  using optimised garbage collection, native application server libraries and by serving static resources from basic http server (e.g. Nginx or Apache2) instead of application server such as Tomcat.

On the other hand, we recommend clustering your application server (with sticky sessions) at latest if you are expecting to have more than 1000 concurrent users on it. By clustering you should be able to scale out Bakery App Starter to theoretically any amount of concurrent users. Clustering on early phase also adds capacity for possible rush usage even though the expected average load would be low.