**Achieve best performance: Spring-Boot OR Node-JS?**

This article is based on a REAL use case scenario for a database intensive application, with focus on performance. The experimental results I’ve come across can help you choose one or the other.

Quite simply, the same application has been realized twice using different languages: JavaScript (Node-JS) and Java (Spring-Boot) with performance measuring executed in varied environments. All the code including deploy (docker-swarm) and testing environment (JMeter) is available at [link] – free for you to play with and customize.

Index:

1. Application and test description - what are we measuring and how
2. Results and conclusion

**1 – Application and tests description**

We will evaluate the performance of four REST JSON endpoints:

1. /find-all -- gets an increasing number of items from the DB
2. /save -- persists a new item on the DB
3. /get/{id} -- get an item from the DB
4. /cpu/{load} -- CPU intensive operation

To better simulate a real web application scenario, both applications includes:

* JWT validation for each endpoint
* Input DTO validation checks
* Log writing on file system

Load testing with Apache JMeter[link], simulating a discrete application load: 5 users simultaneously looping the four endpoint 20 times, for a total of 100 calls for each endpoint.   
Different tests have been run customizing the following parameters:

1. Percentage of CPU occupation
2. Application’s scaling (using docker swarm increasing replica-set)
3. Memory and CPU reservation for each application

Test metrics: average time response for each method. As a bonus I’ve measured up also container startup time and image build time with this useful script: [link]

Database RDBMS PostgreSQL with a very simple schema [link] <https://gist.github.com/GaetanoPiazzolla/5ee8164d349440cb1f30543940945774>

**2 – Results**