**Achieve best performance: SpringBoot OR NodeJs?**

The experimental results I’ve come across can help you choose one or the other.

This article is based on a real use case scenario for a database intensive application. All the code of the two applications [based on Java(SpringBoot) and NodeJs(Express)] including dockerized deploy, postman collections and testing environment 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 compare the performance of four endpoints:

1. /findall: gets a lot 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 app scenario, I’ve included:

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

Tests have been runs using Apache JMeter[link], simulating a discrete application load: 5 users simultaneously 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.

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

2 - RESULTS