Performance Analysis and Comparison of Node.Js and Java Spring Boot in Implementation of Restful Applications

Shyam Mohan J S[1] and Krishanu Goswami[2]

[1]GITAM - Bengaluru Campus
[2]Liverpool John Moores University

October 6, 2023

Abstract

The response time of a website has an impact on the overall user experience. Slow-loading or unresponsive applications can quickly frustrate and alienate users, leading to dissatisfaction and potentially driving them away. Conversely, fast and efficient applications can greatly enhance user engagement, satisfaction, and ultimately contribute to the success of an online business. The main objective of this work is to check and compare the performance between Node.js and Java Spring Boot, in the context of implementing RESTful applications. REST (Representational State Transfer) has emerged as a dominant architectural style for building web services due to its simplicity, scalability, and compatibility with the stateless nature of the web. To achieve this objective, both implementations were created with an identical business use case, aiming to provide the fundamental CRUD operations (Create, Read, Update, and Delete). To conduct the performance analysis, Apache JMeter tool is used that enables the creation of a diverse set of concurrent virtual users, simulating real-world scenarios to load both applications at regular intervals. By subjecting the applications to varying user loads, it can be assessed how each platform performs under different workloads and identify potential bottlenecks and areas for improvement. The results obtained suggest that Node.js excels in faster response times for HTTP GET, ideal for read operations. However, Spring Boot outperforms Node.js for complex operations like POST, PUT, and DELETE under higher user loads. Both faced challenges at 64000 virtual users, with error rates from 10% to 75%, but Spring showed better error handling. Spring required fewer lines of code, benefiting from its environment and dependencies. Resource utilization in Node.js proved to be efficient with CPU load below 40% even under heavy loads. Spring on the other hand provided better response times, exhibited higher CPU usage at times, suggesting more resource-intensive optimization. The comparison emphasizes the importance of technology stack choice in development, speed and performance. The findings serve as a basis for developers to make informed decisions when selecting the appropriate technology stack based on specific project requirements, workload considerations, and performance expectations.

Hosted file

Wiley_SPE_2023.doc available at https://authorea.com/users/671723/articles/671117-performance-analysis-and-comparison-of-node-js-and-java-spring-boot-in-implementation-of-restful-applications