International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 3, May-June-2026, ISSN (Online): 2395-566X

A Comparative Study of Performance and Scalability in Java vs. ASP.NET Enterprise Web Application Frameworks

Assistant Professor Deepa Barethiya, Sakshi Jibhkate, Samiksha Daronde
Department of Master in Computer Application, Nagpur, India

Abstract — This paper compares Java-based frameworks and ASP.NET Core for web applications used by companies. It looks at how they work and how well they handle a large number of users. The study checks things like how long it takes for the application to respond, how much work it can handle, how much of the computer's brain it uses and how much memory it needs when a lot of people are using it at the same time. They ran tests to see what would happen if many people used the application. The results show that ASP.NET Core is really good at responding and using resources wisely. Java-based frameworks are good at handling a lot of users and working with computers at the same time. This study tells us what is good and what is not so good about Java-based frameworks and ASP.NET Core. It helps people choose the tools to build big web applications for companies.

Keywords— Java, ASP.NET Core, performance, scalability, enterprise applications, web frameworks..

I. INTRODUCTION

Enterprise web applications are the backbone of many digital ecosystems, including e-commerce applications, bank management systems, healthcare applications, and enterprise resource planning solutions. These applications need to efficiently handle a large number of concurrent users, and need to be scalable and reliable at all times. So, choosing the right development environment is a critical part of building enterprise-grade applications [1], [2].

Java-based frameworks, especially Spring Boot and ASP.NET Core are the very popular frameworks for developing enterprise applications. Java was the most popular technology in the industry for a long time because of its independence across different platforms. Frameworks such as Spring Boot make it easier for developers to build better, more efficient applications with features such as simple dependency injection, embedded servers and simple cloud integration [1], [7].

However, ASP.NET Core is a framework by Microsoft, which is known for its fast execution speed, small size and efficient resource management techniques. ASP.NET Core is a cross-platform .NET framework for building high-performance web applications. ASP.NET Core is one of the most demanding web development technologies in today's world [9].

Table 1 presents a comparison of ASP.NET Core and Spring Boot frameworks according to important development features of development. As we can see in the table above, there are differences in programming language, controller syntax, view engine, dependency injection, and routing mechanisms. ASP.NET Core uses C# and has built-in dependency injection, whereas Spring Boot is written in Java and uses Spring's DI container. Both frameworks provide efficient support for developing enterprise web applications [1], [10].

Table 1. Feature Comparison between ASP.NET Core and Spring Boot

Feature	ASP.NET Core	Spring Boot
Language	C#	Java
Controller Syntax	[Controller], [ApiController]	@Controller, @RestController
View Engine	Razor (.cshtml)	Thymeleaf, JSP
Dependency Injection	Built-in	Spring's DI container
Routing	Attribute or convention-based	Annotation-based (@GetMapping)

II. PROBLEM STATEMENT

Enterprise applications require a solid framework that can handle heavy use while maintaining good performance and scalability [1]. Two of the most popular frameworks for enterprise-level applications are Spring Boot and ASP.NET Core. It can be challenging and subjective to compare speed, resource consumption, and scalability between the two approaches [2].

While both frameworks can be used to create modern enterprise applications, ASP.NET Core is considered to be faster, more efficient and Spring Boot to be more flexible and better suited to building distributed systems [3], [10]. Since there is no study

that compares these frameworks, developers and organizations are still unsure which one to choose.

The goal of this study is to find the best framework for an enterprise-grade web application development and compare the Spring Boot and ASP.NET Core frameworks in terms of response time, throughput, CPU performance, memory consumption and scalability [1], [6].

Objectives

The aim of this study is to explore the performance and scalability of the Spring Boot and ASP.NET Core frameworks for the enterprise web application development. This paper reviews the performance of the frameworks under different workloads and user traffic conditions [1], [2].

In this work the efficiency of each framework is explored by analyzing the performance metrics such as response time, throughput, CPU utilization and memory utilization. It also looks at the scalability properties of both technologies as the number of concurrent users grows.

The purpose of this research is to analyze the pros and cons of Spring Boot and ASP.NET Core in terms of design, resource management and enterprise application support. It is also talking about what is the better framework to build reliable, scalable and high performing applications [5], [8].

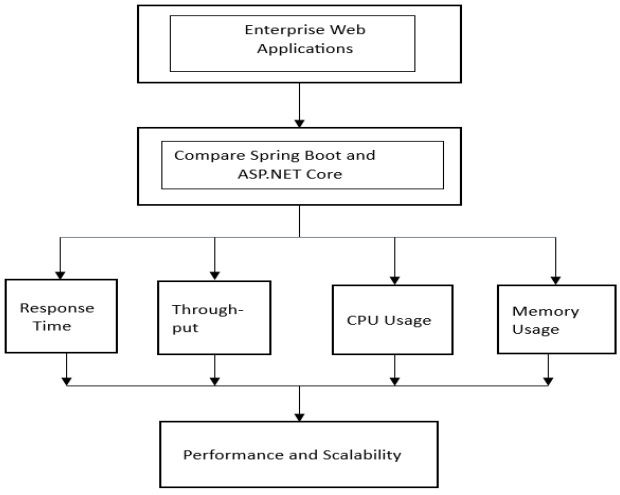

Fig 1. Objectives of the Proposed Comparative Study

Fig. 1 shows the main objectives of the research. The study compares Spring Boot and ASP.NET Core frameworks using performance parameters such as response time, throughput, CPU usage, and memory consumption to evaluate their scalability and efficiency in enterprise web applications [1], [6].

III. METHODOLOGY

Experimental Setup The experimental environment is prepared to ensure a fair and consistent comparison between the Java (Spring Boot) and ASP.NET Core frameworks. The methodology consists of system setup, performance metrics, testing tools, and execution procedure [1], [2].

1. System Setup

The experimental environment is prepared to ensure a fair and consistent comparison between the two frameworks. The applications are deployed in the same hardware and software environment.

- Java Framework: Spring Boot
- ASP.NET Framework: ASP.NET Core
- Database: MySQL
- Operating System: Windows/Linux
- Hardware Specifications:
- Processor: Intel Core i5
- RAM: 8 GB

The methodology consists of system setup, performance metrics, testing tools, and execution procedure. The experimental environment was set up to ensure a fair and consistent comparison of the Java (Spring Boot) and ASP.NET Core frameworks. Methodology includes system setup, metrics for performance, tools for testing and execution procedure [1].

2. System Architecture

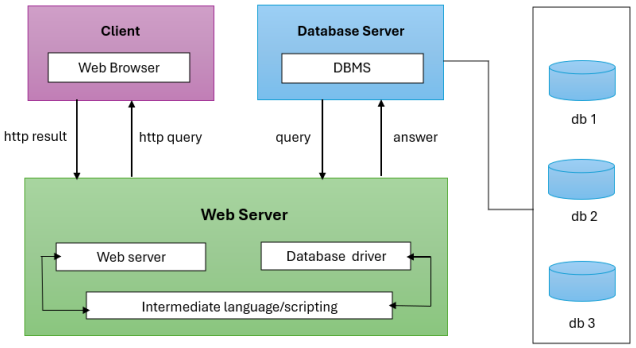

Fig.: illustrates the three-tier architecture used for implementing and evaluating the performance of Java and ASP.NET Core frameworks

The architecture based on the characteristics of enterprise web applications, this study employs the three-tier system

architecture which widely used, including a good scalability, modularity and maintainability.This is the presentation layer or client layer. In this, it is the presentation layer or the client layer. The user interface where users interact with the application is this layer. It is meant to be used to send HTTP requests to the server and show the results it receives. The client may be a web browser, or any front end application.

It is the top layer of the application where incoming requests are received and processed, business logic is implemented and the interface with the database layer is effected. It serves as a middle-man between client and database.It is used for data storage and management. MySQL is used as database system in this study. Application layer coordinates with the database to perform operations such as data retrieval, data insert, and database update [3].

The way in which the architecture functions. How the design works. The system's overall flow is given as follows: The client makes an HTTP request to the server.Client sends an HTTP request to a server. The request is passed to the application layer (Java / ASP.).NET) Application interacts with Database to get or store data.

This is a three-tier architecture because of the following reasons:

Clearly separating concerns.Clearly separates concerns.

- It enhances scalability and performance
- It enables independent development and deployment of components. It can be implemented in enterprise level applications.

IV. LITERATURE REVIEW

Several researches were carried out to determine the performance and scalability of different web application framework platforms. Spring Boot and ASP.NET Core are some of the technologies that are widely used in the current environment. These platforms have been evaluated by scholars on the basis of various performance measures [1].

The research literature suggests that ASP.NET Core is relatively more advanced when it comes to offering fast response time because of its lightweight framework architecture and runtime environment. This technology is therefore suitable for applications that require fast processing and little delay [3], [6].

On the other hand, Spring Boot is flexible and platform-independent and has great potential to be integrated with distributed and microservices architectures. Also, there are various research findings suggesting that frameworks developed in Java are effective in managing large enterprise systems due to the wide range of libraries and active developer community.

The previous studies have mostly focused on performance measures like speed of execution and memory usage. Nevertheless, not much research has been done on an in-depth comparison of the two frameworks based on different performance measures like response time, throughput, CPU utilization, memory usage, and scalability [4], [5], [10].

V. RESULT AND ANALYSIS

To evaluate the performance of the frameworks on Enterprise Web Applications under equivalent workload conditions we measured the performance of each framework using throughput, response time, CPU Usage and Memory Usage.

ASP.NET Core yielded faster responses to requests with increased user levels than that of Spring Boot; the response time for ASP.NET Core when processing requests decreased very steadily as the number of users connecting increased. The CPU usage and memory consumption of ASP.NET Core have also been considerably lower than that of Spring Boot thus producing greater efficiency for the operating system [1].

Spring Boot provides consistent high levels of throughput and ability to process multiple users concurrently, without any variation regardless of the number of concurrent users. This results in a scalable solution and a flexible solution. However, as Spring Boot operates on Java Virtual Machine (JVM), it experiences higher memory requirements across all tested workloads.

Throughput testing indicated that ASP.NET Core has a much larger request processing capability than that of Spring Boot; this would suggest that ASP.NET Core is preferred for applications where low-latency and high-throughput processing is required.

Spring Boot is well suited to support both Distributed and/or Microservices applications; Spring Boot's library support and framework support is very comprehensive [10].

Table 2. Comparative Analysis of Spring Boot and ASP.NET Core

Parameter	Spring Boot	ASP.NET Core
Response Time	Moderate	Fast
Throughput	Good	Higher
CPU Usage	Higher	Lower
Memory Usage	Higher	Lower
Scalability	Strong	Strong
Platform Support	Cross-platform	Cross-platform

Comparison of Performance Between Spring Boot Framework and ASP.NET Core Framework According to Various System Parameters is Presented in the Table Above. In terms of response time, ASP.NET Core framework exhibits better results due to its high efficiency and relatively low CPU and memory usage. The Spring Boot framework also exhibits excellent performance and has good scalability options for enterprise solutions. Cross-platform capabilities and modern-day web application development support are provided by both the frameworks mentioned above. Thus, ASP.NET Core framework is more efficient concerning speed, while Spring Boot has more flexibility options.

VI. CONCLUSION

This paper presents a comparative analysis of the performance and scalability characteristics of Java and ASP.NET frameworks in the context of enterprise web application development. While each framework offers distinct advantages and limitations, the ultimate success of implementation depends primarily on organizational requirements, existing infrastructure, and the development environment Java frameworks demonstrate exceptional portability, platform independence, and scalability, particularly within the JVM ecosystem through robust multithreading capabilities and distributed system support. These characteristics make Java frameworks particularly suitable for large-scale enterprise applications that require cross-platform compatibility, flexibility, and long-term maintainability. Java benefits from a mature ecosystem and extensive open source community support, establishing it as a dependable choice for large-scale, high-traffic systems [7].

Acknowledgment

I am very thankful to all those who helped me and guided me to successfully complete my research paper entitled “A Comparative Study of Performance and Scalability in Java vs. ASP.NET Enterprise Web Application Frameworks."

I am very thankful to my project guide for valuable guidance, continuous support & encouragement during research work. Their smart recommendations and experienced help helped me to understand the principles and to do my research properly.

I would also like to thank the faculty members of the Department for providing the required resources and academic environment for this project. They were the catalyst for the creation of this initiative.

REFERENCES

1. Kronis, K., & Uhanova, M. (2018). Performance Comparison of Java EE and ASP.NET Core Technologies for Web API Development. Applied Computer Systems, 23(1), 37–44.
2. Hamed, O., & Kafri, N. (2009). Performance Prediction of Web-Based Application Architectures: Case Study of .NET vs. Java EE. International Journal of Web Applications, 1(3), 146–156.
3. Dalbard, A., & Isacson, J. (2021). Comparative study on performance between ASP.NET and Node.js Express for web-based calculation tools.
4. Huynh, T. (2024). Web Technologies Comparison Between Nuxt.js and ASP.NET.
5. Poudel, A. (2018). A comparative study of project management system web applications built on ASP.NET Core and Laravel MVC frameworks.
6. Adebukola, O. M., & Kazeem, O. B. (2014). Performance Comparison of Dynamic Web Scripting Language: A Case Study of PHP and ASP.NET. International Journal of Scientific & Engineering Research, 5(7), 78–89.
7. Maddox, A., & Roberts, T. Distributed Web Application Development: A Comparison of. Net and J2EE. Publication of the Manukau Institute of Technology, Auckland.
8. Malavasi, A. (2025). Modern Full-Stack Web Development with ASP. NET Core: A project-based guide to building web applications with ASP. NET Core 9 and JavaScript frameworks. Packt Publishing Ltd.
9. Aroraa, G., & Dash, T. (2018). Building RESTful Web Services with. NET Core: Developing Distributed Web Services to Improve Scalability with. NET Core 2.0 and ASP. NET Core 2.0. Packt Publishing Ltd.
10. Kamiński, F., Kłonica, R., & Pańczyk, B. (2025). Comparative Performance Analysis of RabbitMQ and Kafka Message Queue Systems in Spring Boot and ASP. NET Environments. Journal of Computer Sciences Institute, 37, 457-462.