International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Performance Analysis of J2SE and J2EE in Enterprise Level Applications

Hareesh Kumar Rapolu

hareeshkumar.rapolu@gmail.com

Abstract

The following research project has exemplified the performance analysis of J2SE and J2EE in the context of enterprise level applications. It has measured the performance levels which has catered to explore the handling of both the frameworks to manage huge sets of data. J2SE provides limited scalability and security features while J2EE has nurtured advanced security and scalability characteristics. This has served to be crucial in the segment of lifecycle management for determining successful enterprise applications. Therefore, the relevant findings have promoted making curated decisions. This has been intriguing for selecting the best platform that is fully baked on application requirements.

Keywords: Java, J2SE, J2EE, Performance Analysis, Enterprise Level Applications, Scalability, Security, Lifecycle Management

I. INTRODUCTION

The research project will describe the performance analysis of J2SE and J2EE in the context of enterprise level applications. It will be used to manage large datasets and concurrent user requests. At the same time, J2SE and J2EE are two renowned platforms for the development of such applications. The research project will help to analyse the performance of J2SE and J2EE in enterprise level applications undermining their potential. However, this will shed light on providing a vivid explanation of the scalability and security of both the J2SE and J2EE in the segment of enterprise level applications. Furthermore, the research project will discuss lifecycle management to achieve sustainable results thereby benefiting the enterprises. This will allow it to observe third-party services and libraries thereby analysing performance analysis concerning J2SE and J2EE.

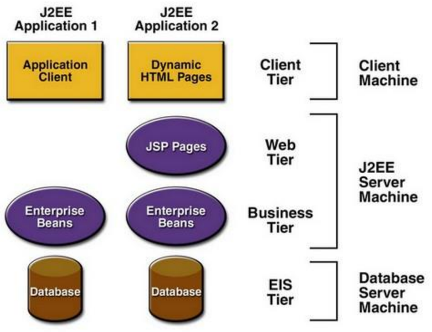

Figure 1: Understanding J2SE and J2EE Platforms

II. PERFORMANCE ANALYSIS OF J2SE AND J2EE IN ENTERPRISE LEVEL APPLICATIONS

The following section defines the performance analysis of J2SE and J2EE in enterprise level applications to develop standalone applications. The analysis measures the CPU utilisation by mastering the usage of memory and determining the startup time. On one hand, J2SE uses Java Virtual Machine configuration which helps to benchmark in handling users with proper scalability[1]. On the other hand, J2EE implements EJB also known as Enterprise Java Beans that is used to measure the startup time and memory usage.

Test	J2SE	J2EE
Startup Time (in ms)	100 - 300	300 - 700
Utilisation Of CPU (%)	10 - 20	10 - 30
Usage of Memory (MB)	50- 100	150- 250
Handling Concurrent User (req / sec)	100 - 500	1000- 5000

Table 1: Demonstrating Performance Analysis of J2SE and J2EE

III. CONSTRUCTING J2SE AND J2EE IN ENTERPRISE LEVEL APPLICATIONS

This section explains the construction of J2SE and J2EE within enterprise level applications and portrays its functioning. Thecase of J2SE illustrates that the construction of an easy inventory management system aids in determining effective applications. It sheds itslight on the main object-oriented principles like encapsulation and manipulation[2]. As a result, this nurtures with complete simplicity and ease of use efficiently for desktop applications. The pseudocode for J2SE is mentioned below.

```
// J2SE Application: Simple Inventory System
class Inventory
{
  list (Item) items;
function addItem(Item item)
{
iems.add(item);
}
function displayItems ()
{
    for each item in items
{
          print(item.details ( ) );
```

```
        }
    }
}
```

The J2EE portrays that fostering with a stateless session bean structured for balancing online order processing within the distributed system. It helps to correspond to annotations for the management of the transactions and persistence. This results in consistency in performance and dependability[3]. Therefore, this makes J2EE facilitate optimum scalability and integration with enterprise level services thereby making it appropriate for large applications. Thus, the pseudocode for J2EE is mentioned below.

```
// J2EE Application: Online Order Management System
@Stateless
Public class OrderService
{
    @PersistContext
EntityManagerem;
     public void processOrder (Order order)
{
em.persist(order);
sendNotification(order);
}
Private void sendNotification(Order order)
{
// Code to send email notification
        }
}
```

IV. UNDERSTANDING SCALABILITY AND SECURITY OF J2SE AND J2EE IN ENTERPRISE LEVEL APPLICATIONS

Scalability and Security of J2SE: It is evident that J2SE primarily resonates on the core Java functionalities which indicates lagging features such as scattered computing and load balancing. It poses an advantage for scaling up the applications[4]. However, this intrigues in managing large amounts of user bases and high traffic volumes.In terms of security, J2SE contains basic security features which indulge in access controls and authentication. It does not support cohesive security mechanisms that are essential for enterprise level applications like role-based access controls (RBAC) and secure communication protocols.

Scalability and Security of J2EE: It is obvious that J2EE is recognised for its improved scalability. The components containing characteristics such as stateless session beans and distributed processing grants applications to manage high loads[5]. A suitable example states that e-commerce that is made on J2EE has the probability to accommodate numerous users in a streamlined fashion. This is done by the augmentation of load balancing and clustering. Similarly, in terms of security, J2EE serves with congruent security measures consisting of JAAS also abbreviated as “Java Authentication and

Authorization Service". Moreover, this is used to promote protected user authentication and role-based access controls. As a result, this served to be necessary for rendering sensitive enterprise data management[6]. Therefore, this categorises J2EE to be identified as a superior choice within high-security environments.

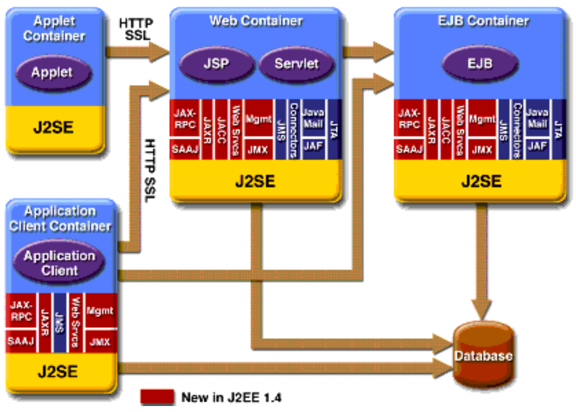

Figure 2: Demonstrating the Architecture of J2SE and J2EE

V. MAINTAINING THE LIFECYCLE MANAGEMENT OF J2SE AND J2EE APPLICATIONS

The following section discusses the lifecycle management of J2SE and J2EE applications in a synchronised form. On one hand, for J2SE, its lifecycle management directly depends on the involvement of continuous monitoring and controlling the upliftment of the states. It contains elements like exceptional designs with operation and retirement. At the same time, this also consists of version controls followed by updates daily and checking the performance to detect stringent security and compatibility[7]. On the other hand, observing the lifecycle management of a J2EE application involves constant amalgamation and rigorous checking. It includes elements such as application server management with version controls and security assessments. Thus, using a framework such as JPA for data management and supplementing with automated testing results in complete dependability and scalability throughout the lifecycle process.

VI. CONCLUSION

This research project has highlighted the performance analysis of J2SE and J2EE in the segment of enterprise level applications. It has been observed that J2SE stands to be suitable for easier applications containing manageable datasets. It has limited security and scalability features. On the contrary, J2EE excels in having robust security and scalability characteristics and has the potential to manage high transactions. Moreover, the research project has analysed that both frameworks have supported sustainable lifecycle management with an elevation in operational efficiency. Furthermore, focusing on performance elements like startup time, CPU utilisation, and memory usage has provided optimum results by getting aligned with specific application needs and preferences.

Abbreviations and Acronyms

- J2SE- Java 2 Platform Standard Edition
- J2EE- Java 2 Platform, Enterprise Edition
- JVM- Java Virtual Machine
- RAM- Random Access Memory
- CPU- Central Processing Unit
- EJB- Enterprise Java Beans
- JAAS-JavaAuthenticationandAuthorization Service

Units

- Startup time is measured in milliseconds
- CPU utilisation is measured in percentage
- Memory usage is calculated in megabytes

Equations

- CPU Utilisation (%) = [(Active CPU Time / Total Time) X 100]
- Total Memory Used (MB) = [Allocated Memory - Memory Free]

REFERENCES

[1]M.Schönefeld, “Pentesting J2EE,” Jan. 2006.Available:https://blackhat.com/presentations/bh-federal-06/BH-Fed-06Schoenefeld-up.pdf

[2] N. Joncheere, “Aspect-Oriented Software Development for the Java 2 Platform, Enterprise Edition,” Jun. 2004.Available: https://soft.joncheere.be/publications/MSc.pdf

[3] P. Schill, “Eclipse as client container for J2EE applications,” Feb. 2005. Available: https://hdms.bszbw.de/files/500/Eclipse_as_Client_Container_for_J2EE_applications.pdf

[4]R.Johnson,“J2EEDevelopmentFrameworks,”Ieee.org,Jan.2005.https://ieeexplore.ieee.org/iel5/2/30112/01381270.pdf

[5] S. Kelly and K. O 'dubhchair, “The Application of Architectural and Design Patterns in Enterprise Systems,” Sep. 2005. Available:https://core.ac.uk/download/pdf/51065374.pdf

[6] V. SumioTasaka and J. Ginbayashi, “Latest Progress and Trends in Java/EJB Technologies,”*FUJITSUSci.Tech.J*,vol.40,pp.85–93,Feb.2004.Available:https://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol40-1/paper11.pdf

[7] X. Yang, weifeng Yin, J. Li, and D. Yu, “Study on Application of Advanced J2EE in StocksExchange,”Jan.2009.Available:http://www.wseas.us/elibrary/transactions/information/2009/31-878.pdf