e-Business Performance Issues, Quality Metrics and Development Frameworks

Vijaya Shetty S
Department of CSE, NMIT
P.B. NO.6429, Govindapura,
Bangalore – 560064

H. Sarojadevi, PhD.
Department of CSE, NMIT
P.B. NO.6429, Govindapura,
Bangalore - 560064

ABSTRACT

e-Business systems have emerged as new powerful promising web service providers to perform business on the Internet. e-Business applications are complex to develop and deploy despite the availability of many software application development tools, frameworks, patterns and methodologies. Many e-Business projects failed to give agreed performance due to unaddressed challenging issues of these complex applications. The paper explores some of the performance issues that need to be considered in the development of e-Business applications. A framework of quality metrics for performance analysis of these systems is also presented in the paper. The work presented gives a broader perspective of performance requirements of e-Business systems to developers, test engineers, and users of these applications. The paper also contributes to the choice of suitable development technology for developing flexible e-Business applications. This research work also compares and contrasts the features of two popular development technologies; J2EE and Microsoft's .NET framework pertaining to e-Business application development.

General Terms

e-Business application performance, e-Business as Web Service, Internet and distributed computer systems, Semantic Web Technologies and Social Semantic Web.

Keywords

e-Business performance issues, Web Services, Quality Metrics Framework, e-Business application development.

1. INTRODUCTION

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network [1]. They have become a promising technology to design and build powerful e-Business applications. e-Business applications promise some of the greatest advances in information technology. e-Business applications are gaining importance day by day as they work like a virtual shop [2]. However the powerful and flexible e-Business systems also pose significant new risks and challenges [3]. Recent distributed e-Business applications typically depend upon various technologies in their realization including the web, scripting languages, server-side processing and a persistent database. The system designed by integrating these technologies requires attention to the performance issues of individual components and the system as a whole. e-Business applications are of the following different types.

- B2B – Business to Business.
- B2C – Business to Consumer
- B2E – Business to Employee
- C2C – Consumer to Consumer
- C2B – Consumer to Business
- G2G – Government to Government

To be specific, e-Business is buying and selling of goods and services and transfer of funds through digital communications. e-Business also refers to online stock, bond transactions, buying and downloading software without going to a physical store [4].Today e-Business means interoperable e-Commerce, e-Collaboration, e-Government, e-Procurement etc. [5]. Figure 1 shows the structure of a 3-tier e-Business site.

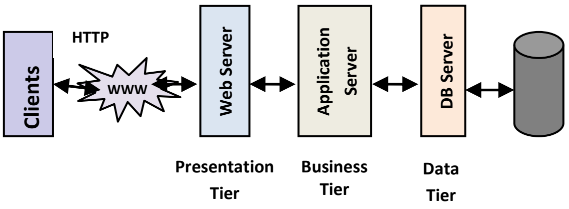

Fig 1: Structure of an e-business site.

An e-Business site is usually structured into three logical tiers: Presentation tier, Business tier and Data tier [6]. Each tier is responsible for different activities. The presentation tier is the web server which receives the client requests and passes them to the next tier if business-related processing is needed. The business tier is the application server which responds to requests from the presentation tier. Application server hosts different threads to perform necessary computations using business logic and enterprise data. The data required by the computation is fetched from the database through data tier if not available in the local cache. Any update of the data is committed to database by means of a transaction.

However, with the dynamic and unpredictable nature of the e-Business infrastructure, providing the agreed QoS is really challenging to the business provider. QoS in e-Business plays a vital role in giving business value to the business provider. In the interest of e-Business performance improvement, the major performance threats to e-Business systems have been identified and presented in section 2 of the paper. The framework of quality metrics pertaining to performance of e-Business is presented in section 3 of the paper. Section 4 explores the features of two major development technologies; J2EE and .NET framework in the context of multi-tier e-Business application development.

2. ISSUES IN E-BUSINES SYSTEMS

There are a number of issues that influence the performance of complex e-Business systems. Some of these issues are outlined below.

2.1 Inadequate Demand Estimation

It is important to determine well in advance the workload and scalability needs of the system and the system must be designed to meet these demands. Otherwise the system may not cope with massive increase in user demands. e-Business applications use database server as backend in a multitier configuration. Such systems when scaled up to support growing internet population with unpredictable usage patterns face several design challenges. Middle tier database caching is one solution to this problem as it improves the attributes such as scalability, flexibility, availability and performance [7]. Several research works have addressed the scalability issue of these systems in literature. One such proposal [8] suggests the use of general-purpose industrial-strength DBMS for middle-tier database caching in multi-tier e-Business infrastructure. Examples of such commercial products include the Database Cache of Oracle 9i Internet Application Server [9] and TimesTen's Front-Tier [10]. In these systems scalability is improved by distributing query workload from backend to several cheap front-end systems. Flexibility is increased by Quality Of Service (QoS) control where each cache hosts different parts of the backend data. Availability of service is achieved for applications that use only cached tables when the backend server is unavailable and performance is improved by responding to locality patterns in the workload.

2.2 Poor Performance Specifications

Poor specification of performance requirements may hide its significance from system designers. This can easily lead to a system design not achieving its performance requirements.

2.3 Unbalanced Architecture

Unbalanced application architecture sometimes leads to over- or under-utilization of some of its components. If some system component is severely under-utilized, it means either it is not really required, or the system is not designed for its proper usage. Under-utilized components should be prevented from the design.

2.4 Inefficient Request Scheduling

Request scheduling refers to the order in which concurrent requests are served. If the request is not scheduled properly or not prioritized correctly then the system may work, but with degraded performance. This is particularly important for significantly high load, where low-priority requests need to be suspended. Usually, servers consider this as an aspect of the operating system. However, some research proposals suggest the use of nontraditional request scheduling policies. The common idea in these proposals is to distinguish classes of requests and schedule these request classes in a specific order, thereby providing different QoS to different class of requests. These research proposals implement the concept of service differentiation by means of request scheduling. For instance, several research works [11–14] have used techniques based on the shortest remaining processing time first scheduling (SRPT) to give priority to the service of short static web content requests against long requests. These research works provide results that demonstrate the efficiency and effectiveness of SRPT scheduling in providing significantly better response time to short requests at relatively low cost.

2.5 Unmonitored System

A system should be monitored continuously to achieve its target performance. Monitoring helps administrator to tune the performance of the system as per user demands. Performance monitoring and tuning tools can be used for this purpose.

2.6 Single Threaded Application Code

This represents a potential performance bottleneck in many complex software systems. Multi-threaded applications can be built for parallel execution using multiple processing elements. A few research works in the literature have demonstrated performance improvement of multi-threaded e-Business applications. One such thesis [15] contributes to performance improvement of multi-threaded Java application execution on multiprocessor systems. The thesis focuses on resource provisioning for multi-threaded Java application’s work area. A proposal of mechanisms to support the cooperation between the applications and the execution environment to improve the performance of Java applications is also presented in the thesis. The thesis suggested modifications that are required in the Java execution environment to allow this cooperation. The cooperation is achieved by establishing a bi-directional communication path between the applications and the underlying system. On one direction, the applications can request the execution environment, the number of processors they need. On the other direction, the applications can request the execution environment at any time to know about their processor assignments. With this resource allocation information, the self-adaptive Java applications can adapt their behavior to the amount of resources allocated to them.

2.7 Poor Testing

Poor testing strategy may not find performance bottlenecks of the design and may even give incorrect opinions about the performance capabilities of the system. Proper testing strategies and tools help improve the application performance.

2.8 Trusting Vendor's Benchmarks

A vendor's benchmark is only an indicative of system performance. The vendor's benchmark may not represent the actual usage scenario of the system or the actual system configuration.

2.9 Insecure Communication

Any distributed system is subject to security threats such as eavesdropping, masquerading, corruption, denial of service, replaying, and repudiation. These security threats may degrade the overall performance of e-Business system. In e-Business systems web clients may employ different technologies such as applets, html, xml, and scripts. The middle-tier contains business logic using technologies such as WebObjects, ASPs, Java J2EE, servlets and JSP. The combination of different technologies at each tier presents special challenges to security of the overall application. The purpose of security in e-Business is to meet the security expectations of users and providers. Web security refers to client-side security, server-side security and secure transmission of information. e-Business security is often designed as a communication security problem using Cryptography. Encryption algorithms and digital signatures are considered to be the basic building blocks of e-Business security. Protocols like SSL (Secure Socket Layer) constitute the next layer of mechanisms that in turn support applications like secure E-mail, electronic payment schemes, a protocol for payment-card transactions etc.

2.10 Server Hardware

The likelihood of hardware failure is more when it is older. Hardware failure may be total failure or partial. The easiest way to prevent such issues is to keep server hardware fresh. Many systems provide hardware fault detection methods where the operating system reboots when hardware failure is detected. Such systems disable the faulty hardware upon reboot. This is a viable solution if the system can work with limited resources and tolerate this downtime. One way to prevent such problems is to use redundant hardware and monitor the system regularly. Throwing more hardware may improve overall performance of the system if the application is compute intensive. However, it is always good practice to upgrade server after addressing the following issues:

2.10.1 CPU Bottleneck

CPU load is a common source of performance bottlenecks. CPUs can handle millions of calculations and instructions, but performance degrades when the number of these operations exceeds the capacity or if the instruction scheduling is inefficient. A high percentage of processor time and a high processor queue length may lead to CPU bottleneck. CPU bottleneck can be minimized using techniques such as:

- If observed rate of context switching is high, reduce the thread count of the process before increasing the number of processors.
- Tune the application that is leading to high CPU utilization.
- Analyze the log file generated by the application and isolate the subsystem that takes the maximum amount of execution time.
- For multi-threaded applications add multiple processing cores.

2.10.2 Memory

The memory system is often a performance bottleneck in terms of bandwidth or latency. When performance problem points to memory, the general idea to solve this problem is to add more memory. However, this practice is effective only in short term. Performance bottlenecks that point to memory are sometimes the result of poorly designed software (memory leaks) or other system flaws that result in memory problems. However, if an application is to perform well, then the developers' need to understand which transactions are likely to conflict and to adapt their application to minimize this. The key to solve memory performance problems is to find the root cause of the problem before adding more memory.

2.10.3 Storage Technology

Disk I/O is another common source of performance bottleneck. Disk I/O is an issue related to disk speed, storage type, RAID (Redundant Arrays of Independent Disks) type and controller technology. Best practices should be used while combining and separating workloads on disks. Directly Attached Storage (DAS) are sometimes faster in some applications than the fastest Storage Area Network (SAN). Large scale enterprise SANs should meet the performance requirements of most applications as they have improved sharing of the storage resources, scalability, manageability, and availability. For instance, a large database can benefit from the backup solutions based on SAN. Similarly, the servers in a data center can use the SAN level replication technologies for application disaster recovery in mass. For these applications DAS cannot provide this kind of scalability. However, there exist practical and physical limits to performance even in the pace of best contemporary disk technology. The key aspects to choose storage technology include reliability, capacity, performance, complexity and cost.

3. QUALITY METRICS OF e-BUSINESS SYSTEMS

Complex deployments and integrations frequently leave e-Business users and enterprises confused. This section focuses on outlining the most important metrics to monitor and analyze within e-Business application as it relates to its performance. By focusing on these metrics, application can be more proactive to improve the effectiveness of e-Business performance.

3.1 Performance

Performance is a critical aspect of any web based system. e-Business systems are expected to give a fast response even when scaled up to run many transactions per time unit. The standard measures of performance are the TPC benchmarks, which compare transaction processing systems based on their maximum transaction rate and price per transaction for a standardized application workload. Performance of the system determines the rate at which requests are serviced. It can be measured in terms of throughput, response time, latency, turnaround time, efficiency, nominal capacity, usable capacity and so on.

- Throughput: the amount of work done per unit time. Examples: Jobs per second, Requests per second, Transactions Per Second (TPS).
- Response time: the time a system takes to react to a given input between sending a request and receiving the response.
- Latency: the round-trip delay between sending a request and receiving the response.
- Turnaround time: the time elapsed between the submission of a job and the completion of its output.
- Efficiency: Ratio of usable capacity to nominal capacity. Or, the ratio of the performance of an n-processor system to that of one-processor system is its efficiency.
- Nominal Capacity: maximum achievable throughput under ideal workload conditions. E.g., bandwidth in bits per second.
- Usable capacity: maximum throughput achievable without exceeding a pre-specified response-time limit.

3.2 Availability

An e-Business system is often critical to proper functioning of the enterprise that uses it. Therefore, another important property of these systems is availability. Availability is the ability of the system to provide continuous service to its users. Availability of the system is determined by the rate of system failures and recovery from failures. Availability measures the readiness of the system. It is the probability that the system is available for service and is related to reliability. Time-to-Repair is one of the metrics that represents availability of the system.

3.3 Scalability

Scalability is the ability of the system to expand in terms of requests, clients or transactions without degrading performance. It is the ability of the system to adapt to growth. It is important for the e-Business model to be scalable, since there is much potential for growth in the online market. The development of scalable technological infrastructure is an important aspect of growing e-Business. Lack of scalability could be harmful to an e-business, disrupting its regular operations, which can result in loss of revenue. Improving an IT infrastructure or the process of upgrading the system could prove to be more costly than designing a scalable system.

3.4 Security

Security metrics for e-Business measures different security aspects like non-repudiation, confidentiality, authentication, authorization, encryption and access control. e-Business provider may apply different levels of security policy depending on the service request. There is a growing concern about security of e-Business as the service is delivered over public Internet.

3.5 Integrity

Data plays an integral role in the operations of an e-Business. Integrity of data has gained importance with the advent of e-Business. Preserving the integrity of data is crucial for businesses. Integrity is the ability of the system to prevent unauthorized access to or modification of data [16]. Integrity is the measure of consistency of the data before and after the execution of the requests. To be specific integrity means accuracy and completeness. Data integrity can be violated by system crashes, errors made by the individuals, viruses and hackers.

3.6 Interoperability

Interoperability measures the degree to which an e-Business service can interoperate with clients and servers implemented in different languages and/or platforms. Interoperable e-Business services can be hosted on any operating system. An e-Business service can be made interoperable by using a set of implementation standards.

3.7 Fault tolerance

Fault tolerance is the ability of the system to operate in the face of partial failure. For transactional workloads, it is the ability of the system to recover from a failure without losing any data or updates from recently committed transactions.

3.8 Concurrency

Concurrency is the ability of the system to support several independent concurrent users. Concurrency issues are critical to performance of the application and are difficult to isolate. Concurrency is the measure of system's capability to withstand different types of requests under varying loading conditions.

3.9 Reliability

Reliability is the ability of the system to sustain continued load that has unpredictable pattern, with an acceptable level of performance degradation. Reliability of the system depends on various factors such as memory consumption, CPU utilization, performance measured by response time and throughput, and cycle time for each class of workflow. Performance is considered as the end user responsiveness of the system under various loading conditions. Reliability is the overall measure of the system to maintain its service quality and is related to the number of failures per unit time.

3.10 Robustness

Robustness measures the degree to which an e-Business service can still perform correctly in the presence of erroneous inputs.

4. E-BUSINESS FRAMEWORKS

E-Business has been popular since the emergence of Internet and World-Wide-Web. E-Business applications are based on N-tier architecture to improve efficiency, availability and performance. These applications are developed using development platforms/frameworks. A development framework includes programming languages, run-time environment, and reusable class libraries [17]. There are two popular technologies in enterprise-level application development. They are Microsoft's .NET framework and Sun Microsystems' J2EE. In addition, there are alternative technologies like WebObjects, Coldfusion, PHP (Hypertext Preprocessor) etc. This section of the paper describes compares and contrasts J2EE and .NET from the perspective of e-Business application development and its performance.

J2EE is a good platform to build e-Business systems for a server site in Unix (Linux) or Windows operating systems. The applications of J2EE are normally built in Windows operating system and can be published on any operating system. Microsoft's ASP (Active Server Pages) can also be used to build e-Business applications. ASP applications are developed and published in Windows operating systems only. Most of the initial banking and other e-Business applications were built in J2EE instead of ASP because Unix systems are considered to be more stable than windows based system. To control e-Business market, Microsoft presented .NET framework to the market in 2000. .NET is both platform independent and programming language independent.

4.1 Java 2 Enterprise Edition (J2EE)

Java platform consists of Java Application Programming Interfaces (APIs) and Java Virtual Machine (JVM). Java interpreter interprets Java program to Java byte codes that can be run on JVM. JVM translates these byte codes to machine code of target machine for execution. Thus, Java byte codes are executable on any operating system if JVM is installed. Java APIs are byte codes available in built-in libraries. APIs of J2EE are extended from J2SE (Java 2 platform Standard Edition). J2EE Components support modularization to simplify the development cycle of the application. Developer can concentrate more on the business logic as there is no need of designing expensive applications for transaction security, database management, naming service etc. J2EE extends the platform independency and object-oriented language features of Java 2 platform Standard Edition. J2EE supports e-Business systems providing JDBC API for database access, Enterprise JavaBeans (EJB), Java Servlets API, Java Server Pages (JSP) API, XML, Java Mail API, Java Messaging API etc. In order to execute J2EE codes, both J2EE application server and JVM are needed. A number of application servers are available in the market such as BEA WebLogic, IBM WebSphere, ATG Dynamo, Sun One Application server etc. To connect databases, JDBC driver is needed for each type of database. JDBC driver is normally provided by the vendor. Basic J2EE multi-tier application architecture is shown in figure 2.

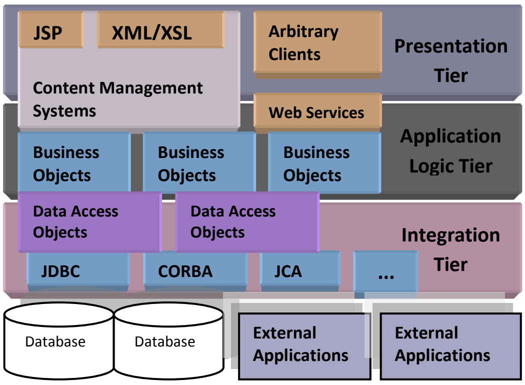

Fig 2: J2EE multi-tier application architecture

J2EE enterprise applications are generally considered as multi-tiered applications that majorly consist of three tiers: Presentation (client) tier, Application logic tier and Integration tier. Arbitrary client represents a Web browser, a Java application, an applet and so on. The clients make requests to the web server .Web server serves the request either by returning static content, if it is present in the web server or by forwarding the request to either servlet or JSP in the application server for either static or dynamic content. Presentation tier consists of the presentation logic required to serve clients. A servlet or JSP in the presentation tier processes clients' requests, manages logons, sessions, accesses the business services and constructs a response which is delivered to client. Application /Business logic tier offers the business services. This tier contains the business logic and the entire business logic is bundled into this tier. The benefit of this bundling is that the same business logic can support different types of clients including browser and other applications. Integration tier takes the responsibility of communication with external resources such as databases, legacy systems, ERP systems, messaging systems etc. The components in this tier use JDBC, JMS, J2EE Connector Architecture (JCA) and some proprietary middleware to access the resources such as a database, ERP system etc. These external systems are responsible for data storage. This tier is also known as Data Tier or EIS (Enterprise Information System) Tier.

4.2 .NET framework

.NET framework supports multiple languages such as Visual Basic, C++, C#, and J#. Source code written in one of these languages is compiled to a MSIL (Microsoft Intermediate Language) code. CLR (Common Language Runtime) of .NET framework interoperates the MSIL codes so that MSIL codes in any language can communicate with each other. CLR translates the MSIL codes to the machine codes. Even though it runs only on Microsoft Windows system, Microsoft provides CLS (Common Language Specification) to achieve platform independency.

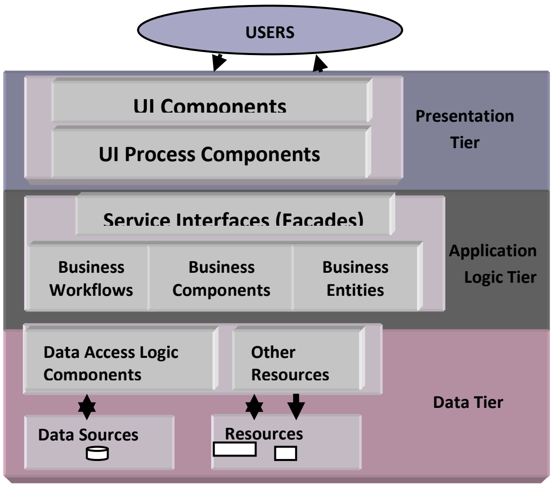

Fig 3: .NET Framework application architecture.

The basic multitier architecture of an application in .NET framework is shown in figure 3.The architecture has three main tiers: Presentation tier, Business tier and Data (or Resource) tier [18]. The Presentation tier is responsible for user interaction with the application. Different types of applications including web applications such as ASP.NET and desktop applications such as windows forms can be designed using components of this tier. The presentation tier mainly consists of two types of components; User interface (UI) and User interface process components. The Business tier consists of the components to implement the business logic for presentation tier requests. Components of this tier can exchange data with the data tier or any external services. Business tier contains the elements that are required to execute the business logic. A Business tier can have elements such as business components, business workflows, business entities and service interfaces. Business components implement the functionality of a particular use case scenario. The Data Access Logic tier consists of the elements necessary to access data from data sources and returns it to the business tier. The Data Access Logic tier has data access logic components, data access logic service façade and databases. Data access logic components provide simple access to functionalities such as queries and data operations. These components can return both simple and complex data structures as results. The data access logic service facade handles requests from the business tier and sends them to the data access logic components of the data access logic tier.

4.3 Comparing and Contrasting

In table 1, J2EE and .NET technologies are compared and contrasted based on the criteria unique for performance of e-Business application.

5. CONCLUSIONS

e-Business is an emerging web service technology that promises a service-oriented, component-based application architecture that is based on an open Internet based infrastructure and protocols. They are rapidly growing as popular standards for sharing data and applications over the web. Many enterprises are investing on web services for integrating their business solutions to e-Business applications. This has led to the need of developing flexible, powerful e-Business applications with QoS. To achieve this, an extensive survey of the literature has been carried out, which resulted in a set of QoS issues pertinent to performance and a design of quality metrics framework to quantify various performance attributes of e-Business. A critical comparison of two popular development technologies - J2EE and .NET framework for multi-tier e-Business applications is also presented in the paper. This paper gives a broader perspective of performance requirements of e-Business to developers of these applications and their performance tools. The contribution of this paper to e-Business paradigm would be useful to developers as it is difficult to predetermine the performance issues, quality metrics and suitable development technologies for dynamic e-Business infrastructure that are unpredictable in nature.

6. ACKNOWLEDGMENTS

We thank Nitte Meenakshi Institute of Technology for providing the support for publishing this paper. Our special thanks to Prof. N.R.Shetty, Director NMIT, Dr. H.C.Nagaraj, Principal NMIT and Dr.Nalini N., HOD department of CSE NMIT for their invaluable support.

7. REFERENCES

[1] Venu Datla, Goseva-Popstojanova, K., "Measurement-based Performance Analysis of E-commerce Applications with Web Services Components", e-Business Engineering, PP. 305 – 314, 2005.

[2] Pradnya B. Rane, B. B. Meshram, "Application- Level and Database Security for E-Commerce Application", IJCA, Volume 41 - Number 18, 2012.

[3] Dr. John Murphy, "Assuring Performance in E-Commerce Systems",http://www.eeng.dcu.ie/~pel.

[4] Ijaj Md. Laisuzzaman, Nahid Imran, Abdullah Al Nahid, Md. Ziaul Amin, Md. Abdul Alim, "The framework for implementing ecommerce: the role of bank and telecom in bangladesh", Journal Of Telecommunications, Volume 1, Issue 1, February 2010.

[5] Frank-Dieter Dorloff, Ejub Kajan, "Handbook of Research on E-Business Standards and Protocols: Documents, Data and Advanced Web Technologies", ISBN13: 9781466601468.

[6] Giovanna Ferrari, Paul Ezhilchelvan, Isi Mitrani, "Performance Modeling and Evaluation of E-Business Systems", ACM ANSS '06, PP. 135-142, 2006.

[7] C.Mohan, "Caching Technologies for Web Applications." ACM VLDB '01,PP. 726,2001.

[8] Qiong Luo, Sailesh Krishnamurthy, C. Mohan, Hamid Piraheshd ,Honguk Wooq , Bruce G. Lindsayd , Jeffrey F. Naughton, "Middle-Tier Database Caching for e-Business", ACM SIGMOD '02,PP. 600 – 611,2002.

[9] "Oracle9i Application Server Overview Guide", Release 1.0.2,http://docs.oracle.com/cd/A97336_01/ias.102/a87353/componen.htm#1017116

[10] "Oracle TimesTen In-Memory Database", http://www.timesten.com/products/fronttier/index.html

[11] Mark E. Crovella, Robert Frangioso, and Mor Harchol-Balter, "Connection scheduling in web servers", USITS' 99, 1999.

[12] Harchol-Balter M, Schroeder B, Bansal N, Agrawal M. "Size-based scheduling to improve web performance", ACM Transactions on Computer Systems (TOCS) , 21(2):207–233,2003.

[13] Schroeder B, Harchol-Balter M. "Web servers under overload: How scheduling can help", ACM Transactions on Internet Technology, 6(1):20–52, 2006.

[14] Rawat M, Kshemkayani A. "SWIFT: Scheduling in web servers for fast response time", Second IEEE International Symposium on Network Computing and Applications (NCA 2003), Cambridge, MA, U.S.A., PP.51–58, 2003.

[15] Jordi Guitart Fernández, "Performance improvement of multithreaded java applications execution on multiprocessor systems", A dissertation submitted in partial fulfilment of the requirements for the degree of doctor per la university at politècnica de catalunya, 2005.

[16] Mohamad Ibrahim Ladan, "Web Services Metrics: A Survey and A Classification", International Association of Computer Science & Information Technology (IACSIT), 2012

[17] Tejinder Singh, "Structural, Technically and Performance Aspects in Enterprise Applications or Projects", International Journal of Scientific and Research Publications, Volume 2, Issue 4, April 2012.

[18] "Implementing Interoperability Design Elements", http://msdn.microsoft.com/en-us/library/ff649508.aspx

Table 1: Comparing and contrasting J2EE and .NET framework

Criteria	J2EE	.NET	Performance Remarks
Programming Language supported	Java with enterprise APIs such as JDBC (Java Database Connectivity),JNDI (Java Naming and Directory Interface), Servlet, JSP (Java Server Page), RMI (Remote Method Invocation), EJB (Enterprise JavaBean), and JMS (Java Messaging Service).	Multiple languages such as C++, Visual Basic, ASP, C# and J#. Applications written in these languages can interoperate with each other.	J2EE has better application portability, and developer friendliness, while .NET framework is programming language independent and is more extensible than J2EE.
Platform Independency	Platform independent language with Java Virtual Machine (JVM) provided by Sun Microsystems.	Platform dependent. Works on windows environment. .NET framework may achieve platform independency using Platform Adaption Layer such as SSCLI (Shared Source Common Language Implementation) working on Microsoft Windows XP, the FreeBSD OS, and Mac OS X 10.2	.NET is basically designed to work on Microsoft Windows. J2EE platform independency supports development of applications that are more stable and secure.
Component Model	Provides component model named EJB (Enterprise JavaBeans). It runs on an EJB application server	COM (Component Object Model), DCOM (Distributed Component Object Model) and COM+	.NET has better flexibility and portability of components.
Database Connection	JDBC (Java Database Connectivity) – technology. An entity bean of EJB has database connection Interfaces.	OLE (Object Linking and Embedding) DB. ADO (ActiveX Data Object) .NET is on top of OLE DB which is a database object model that is composed of many standard classes to refer to data from any database.	.NET on OLEDB is more complex, hence may incur more performance overhead.
Application Server	Popular application servers in the market are BEA WebLogic, IBM WebSphere, Macromedia JRun, ATG Dynamo, and Oracle application server Tomcat , JBoss etc.	Microsoft IIS web Server. IIS server handles both static and dynamic web pages and hence can be called as application server.	J2EE developer has the flexibility to choose one of the many application servers in the market. Performance of J2EE application depends on inherent performance of the application server used.
Standard	Liberal standards	Proprietary and rigid standard	.NET has better amenability for Quality standards