Performance Evaluation in Web Architectures

RUPESH NAGENDRA

Department of Information Technology, Shri Vaishnav Institute of Technology and Science, Indore, India.

(Received: March 14, 2014; Accepted: March 28, 2014)

ABSTRACT

There are several web architectures present in World Wide Web (WWW) such as Client-server web architecture (CWA), distributed web architecture (DWA) and service-oriented architecture (SOA). DWA and SOA is the most popular distributed technology in whole WWW. The use of SOA or more specifically the web services technology is an important architecture decision. Marketing application also used only distributed technology, because distributed technology is provided several quality attributes modifiability, performance, availability, security, interoperability and scalability. CWA doesn’t provide quality attribute as much DWA and SOA. Today distributed web architecture has become the most popular approach for developing web architecture. In this paper we have evaluated the performance quality attributes for CWA, DWA and SOA. These performance evaluations have observed in terms of time.

Key words: CWA, DWA, Quality attributes, SOA, WWW.

INTRODUCTION

Now a day’s for web development mainly used CWA, DWA and SOA. However business stakeholders believe in DWA and SOA for his business application. Distributed web-applications play major role to improve web application through quality attributes. Along with the great demand of web applications in the era of pervasive and ubiquitous computing, a number of issues related to quality of web-applications have also gained a great attention in web-application development process. It supports quality attributes like reliability, usability, security, availability, scalability, interoperability, performance and maintainability.

As far as quality attributes are concerned, distributed web architecture (DWA and SOA) provides much better quality attributes over the CWA. Web architecture has different types of tier (N-tier) architecture.

In which CWA is considered as 1-tier or 2-tier architecture, DWA 3-tier or N-tier architecture and SOA is considered as 3-tier or N-tier architecture. In this paper, we have evaluated CWA provides high performance over the DWA, SOA. The reason behind it, CWA is 1-tier or 2-tier architecture. In this paper, we have also described how to improve performance in DWA and SOA such as CWA.
Related work

A framework for Component-Based Distributed Applications Design known as Component Oriented Distributed Environment. This paper describes distributed architecture without considering quality attributes.

In Distributed web architecture, it has described about Remote Management and Configuration, On-line Configuration, Multi-platform runtime.

An Architecture for Acceleration of Large Scale Distributed Web Applications, in this paper it has described about proxy server and middleware.

Service-Oriented Architecture and its Implications for Software Maintenance and Evolution [12], Service-Oriented Architecture (SOA) is having a substantial impact on the way software systems are developed. According to a 2007 Gartner Group report, 50% of new mission-critical operational applications and business processes were designed in 2007 around SOA, and that number will be more than 80% by 2010. This means that as service-oriented systems are deployed and legacy systems are migrated to this environment, a main concern is now their maintenance and evolution. This paper provides an overview of SOA concepts, best practices for implementation of service-oriented systems, the implications that SOA environments have on software maintenance and evolution activities, and finally some research challenges for the software maintenance and evolution research community.

Quality Attributes for Service-Oriented Architectures, the use of SOA or more specifically the Web services technology is an important architecture decision. An architect should understand how different quality attributes for a system are impacted by that decision. While there are significant benefits with respect to interoperability and modifiability, other qualities such as performance, security and testability are concerns. This paper discusses how the different quality attributes of a system can be positively or negatively affected by the use of such technology.

Overview of web architecture

The CWA supports client-server architecture. Client sends the request to server through browser. Then server sends the response in form of HTML. It does not support many quality attributes.

Modern web architecture should be distributed web architecture, because it provides some quality attributes such as reliability, usability, security, availability, scalability, interoperability, performance and maintainability. It also provides some other advanced technology which is required now a day’s such as fault-tolerance, high computational power, middleware supports etc. A good system architect is a heavily strategic thinker. The architect needs to understand these tactical issues and determine the best high-level solution. It also provides good support for requirements of stakeholders.

DWA architecture provides a model for developers to create a flexible and reusable application. By breaking up an application into tiers, developers only have to modify or add a specific
layer, rather than have to rewrite the entire application over. There should be a presentation tier, a business or data access tier, and a data tier.

Service-Oriented Architecture (SOA) is a way of designing, developing, deploying and managing systems characterized by coarse-grained services that represent reusable business functionality and service consumers that compose applications or systems using the functionality provided by these services through standard interfaces. Benefits commonly associated to SOA adoption are cost-efficiency, business agility, adaptability and leverage of legacy systems. There are many successful case studies of SOA adoption, mainly in commercial enterprises, where the main goal for SOA implementation is internal integration and business process improvement. There are also multiple studies, such as the InfoWorld SOA Trend Survey, that state that many SOA adoption barriers are lack of governance and finding people with the right skills.

At a high level, SOA is a way of designing, developing, deploying and managing systems, in which:
(a) Services provide reusable business functionality.
(b) Applications or other service consumers are built using functionality from available services.
(c) A SOA infrastructure enables discovery, composition, and invocation of services.
(d) Protocols are predominantly, but not exclusively, message-based document exchanges.

Quality attributes
There are several types of quality attributes present in software architecture to improve the applications, which are given as below:

Reliability
Reliability is the ability of a system to keep operating over time without failure. Many researchers are reluctant to admit it; most software currently produced does not need to be highly reliable. Many businesses’ commercial success depends on Web software; however, if the software does not work reliably, the businesses will not succeed. The reliability of a system is typically measured as its mean time to failure (MTTF), the expected life of the system. So, reliability should be maintained in distributed web architecture.

Usability
Traditional software applications have users, but web sites have customers. As said above, most web software applications have a broad customer base. These customers have grown to expect software to be as simple to learn as shopping at a store. Although a lot of knowledge is available for how to develop usable software and web sites many web sites still do not meet the usability requirements that most of us expect. This, coupled with the fact that customers exhibit little site loyalty, means that web sites that are not usable will not be used: customers will quickly switch to more usable web sites as soon as they are put online.

Security
We have all heard about Web sites being cracked and private customer information distributed or held for ransom. This is only one example of the many potential security flaws in Web software applications. Today, however, the breach of a company’s Web site can cause significant revenue losses, large repair costs, legal consequences, and loss of credibility with customers.

Availability
Availability is fraction of mean time to failure to addition of mean time to failure and mean time to repair.

\[
\text{Availability} = \frac{MTF}{MTF + MTR}
\]

Here, MTF = mean time to failure
MTR = mean time to repair

Scalability
Scalability is the ability of the system to be changed in size or in volume in order to meet users’ needs. The major issue in Distributed web application scalability is the ability of the site where the services are located to accommodate an increasing number of service users without performance degradation. The Web application
technology does not offer any inherent scalability feature. To respond to scalability requirements, the architect has to rely on the mechanisms provided by the platform vendors. Strategies to improve scalability on the server provider side include:

**Horizontal scaling**
- add load-balanced servers and Vertical scaling: increase the capacity of a server.

**Maintainability**
In engineering, maintainability is the ease with which a product can be maintained in order to:
- (a) isolate defects or their cause
- (b) correct defects or their cause
- (c) meet new requirements
- (d) make future maintenance easier, or
- (e) cope with a changed environment

In some cases, maintainability involves a system of continuous improvement - learning from the past in order to improve the ability to maintain systems, or improve reliability of systems based on maintenance experience.

**Performance**
If a web site is too slow, users will lose patience and go away. Users will lose concentration if they have to wait more than a few seconds and will leave your web site never to return after a delay of 30 seconds. Thus performance is important, but it is more important to note that performance is dominated by the internet traffic, often at the user's end. While software can impact the overall performance, it is really hard to write software that poorly. So performance should be improved, for improving the performance load balancing memory management concept can be employed.

**Reusability**
In computer science and software engineering, reusability is the likelihood a segment of source code that can be used again to add new functionalities with slight or no modification. Reusable modules and classes reduce implementation time, increase the likelihood that prior testing and use has eliminated bugs and localizes code modifications when a change in implementation is required.

**Interoperability**
Interoperability refers to the ability of a collection of communicating entities to share specific information and operate on it according to an agreed-upon operational semantics. Interoperability provides business to business Communication. Distributed web application doesn't provide interoperability, to achieve that need any connector which should be able to communicate with two different technologies like Java and .Net.

<table>
<thead>
<tr>
<th>Quality Attributes relationship in web architecture</th>
<th>CWA</th>
<th>DWA</th>
<th>SOA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web architecture</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modifiability</td>
<td>Below average</td>
<td>Average</td>
<td>Strong</td>
</tr>
<tr>
<td>Reusability</td>
<td>Average</td>
<td>Very strong</td>
<td>Very strong</td>
</tr>
<tr>
<td>Scalability</td>
<td>Poor</td>
<td>Strong</td>
<td>Strong</td>
</tr>
<tr>
<td>Inter-operability</td>
<td>Average</td>
<td>Strong</td>
<td>Very strong</td>
</tr>
<tr>
<td>Portability</td>
<td>Below average</td>
<td>Strong</td>
<td>Strong</td>
</tr>
<tr>
<td>Security</td>
<td>Average</td>
<td>Average</td>
<td>Average</td>
</tr>
<tr>
<td>Performance</td>
<td>Strong</td>
<td>Strong</td>
<td>Strong</td>
</tr>
<tr>
<td>Availability</td>
<td>Average</td>
<td>Average</td>
<td>Average</td>
</tr>
<tr>
<td>Reliability</td>
<td>Below average</td>
<td>Strong</td>
<td>Very strong</td>
</tr>
<tr>
<td>Usability</td>
<td>Below average</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Key's relationship for Table 1 in ascending order has poor, below average, average, strong, very strong
comparision between TWA, DWA and SOA

The comparison between CWA, DWA and SOA is shown in the tabular form. This table (TABLE I) has been created with help of literature survey, software engineering websites, analysis, design, and implementation strategies.

In scalability distributed web architecture have strong relationship because of distributed have web server, and middleware. Fault tolerance, reusability, and portability are strong part of distributed web architecture providing good loosely coupled environment.

DWA and SOA is poor in some quality attributes over the CWA. For performance attribute, in distributed web application, it gives the bad performance over the traditional web architecture. Availability and reliability quality attribute gives the poor relationship in DWA over CWA. We have to improve the particular quality attributes.

Finally from table I, it is found that distributed web architecture has good capability to provide the quality attributes over the tradition web architecture. This is the reason why the use of distributed web architecture is increasing day by day. Further, distributed web architecture fulfills all types business stakeholder’s requirements.

Implementation

For implementation, we have designed and implemented a small application both in CWA and distributed web architecture. Then non functional requirement mainly performance was observed. This performance is better in CWA than in distributed web architecture.

There are six figures illustrated here. The first two figures show the implementation of a very small application for CWA while next two figures considers a distributed web architecture. We developed this application to search employee detail using employee id in the created database for TWA and DWA.

For CWA we used localhost machine in which database is present in single machine, which has been shown in the Figure 1 and Figure 2.

For DWA database is kept in second machine, and the client program is kept first machine which requests for database which is present in second machine which has been shown in figure 3 and figure 4.

To develop this application in SOA, we create the wsdl file for person details, where create a web-service method for person detail with license id parameter and output comes from database, which has been shown in figure 5 and figure 6.

In Figure 1 it is simple web page for searching employee detail through employee id. In Figure 2, all detail of employee is fetched from localhost database.

In Figure 3 it is simple web page for searching employee detail through employee id. In Figure 4, all details of employees are coming from second system where database is present.

However employee details display the time elapsed. This time elapsed in displaying the employee details for CWA, DWA and SOA is to be noted for checking performance.

In Figure 7 we calculated the time for CWA, DWA and SOA for one request through Figure 1, 2, 3, 4, 5 and 6. Figure 7 shows that how much time is taken to complete the job and display the employee details. It also shows average time. These times can be noted for further performance evaluation.

Fig. 1: Web page of search detail through employee in CWA
Fig. 2: Web page of employee detail through employee id in CWA

Fig. 3: Web page of search detail through employee id in DWA

Fig. 4: Web page of employee detail through employee id in DWA

Fig. 5: Web page of search detail through employee in SOA

Fig. 6: Web page of employee detail through employee id in SOA

Fig. 7: Performance evaluation in web architecture
From Figure 7 it is illustrated, CWA application has taken minimum time. CWA, DWA and SOA have taken 1.87 seconds, 2.03 seconds and 2.24 seconds to complete the job. According to data, the sequence order of performance is CWA, DWA then SOA.

It is observed as per evaluation that performance of CWA, DWA and SOA are not similar but in fact it does not happen in WWW where a lot of company uses DWA and SOA because its quality attributes are poor with respect to CWA. So company architects have given extra efforts to improve the distributed architecture similar to CWA for some particular quality attributes.

In WWW, DWA gives response quickly, because it uses memory management, server management, middleware, load balancing concept. But while we use important quality attributes in distributed web architecture, its performance degrades which requires some efforts to improve services.

CONCLUSION

Choosing an architecture that satisfies the functional and the quality attribute requirements is vital to the success of a system. Nonetheless, the requirements that will shape the architecture design are the quality attribute requirements. DWA and SOA are able to solve the problems. Several types of quality attributes provided by distributed web application can be helpful for architect and end users. Here we analysis the performance quality attribute for CWA, DWA and SOA. CWA has provided good performance over the DWA and SOA, but developers can achieve the performance in DWA and SOA like CWA with some new technology such as load balancing, memory management, sever management, middleware etc. We observed SOA is more popular web architecture because it provides the more quality attributes with strong relationship. Some quality attributes is not much better in SOA like availability, reliability and performance but developers can develop good architecture design for improve those quality attributes.

REFERENCES


