Qualitative & Quantitative analysis of tiered Architecture of Web-Applications

Muhammad Akram[+] and Sikandar Hayat[++]

[+]*Software Engineer at Europak Web, Stockholm Sweden*
[++]*Dept. of Telecommunication Systems, Blekinge Institute of Technology, Karlskrona Sweden.*
akram.moghal@gmail.com , sikandarhayatraja@hotmail.com

Abstract—In this era web applications are transformed to intricate web applications that work in distributed environment and provide functionality on World Wide Web. It is pragmatic that in most of the cases no specific architecture is followed during their development. This causes high level of error possibility. To develop and maintain a successful high quality well designed web application one has to focus on the application's environment and functional requirement prior to the implementation. Performance, availability and complexity are very important factor in any web application development. In this paper we have perform qualitative (i.e. focus groups) & quantitative (i.e. experiments) analysis on web-application architecture. We have discussed briefly different architectural view points in web applications and structure of tiered architecture. At the end we have mentioned the factors that affect on the choice of architecture and how to overcome performance, availability and complexity challenges in tiered architecture. It is concluded that some specific architecture should be for web applications to attain better performance. It gives whole application development team to focus on single architectural model. Probability of success for an application increases in terms of meeting requirements and needs of the user. Each tier architectures have different level of performance, availability and complexity. The experiment result shows that vertical 3-tier architecture is best among others.

***Index Terms*— N-tier, one tier, Qualitative and Quantitative analysis, tiered architecture, two tier, three tier.**

I. Introduction

Most of the web applications work in distributed environment [2], [7]. These applications may consist of many errors if no specific architecture is followed during their development [9]. This paper describes the importance of different architectures (i.e. one-tier, two-tier, three-tier and n-tier) that an organization can use for the development of better quality web applications. In this paper we have perform qualitative (i.e. by using focus groups) & quantitative (i.e. by using experiments) analysis on web-application architecture. It also explains factors that are necessary to keep in mind during the development. These factors are discussed in detail in rest of paper. We have mentioned the factors that affect on the choice of architecture and how to overcome performance, availability and complexity challenges in tiered architecture. At the end, this paper illustrates a survey that we have conducted to record comments from different web application designers and for quantitative analysis we have perform test. The basic aim was to analyze which architecture is better in terms of performance, availability and complexity.

II. Background

Architecture explains the structure/framework of any software application. In which whole system is divided into sub systems, it tells us the relationship between different components of the application. Due to architecture system becomes easy to understand. [8]

Development of architecture depends upon different factors, like functional requirements (client, user), quality consideration (performance, scalability, reusability), technical aspects (operating system, middleware) etc.

Architecture may change during development if any of the above is changed; requirements gathered at the beginning of project are not clear; they are changed when the system is completed etc.

A. Architecture of Web Applications

Initially centralized, mainframe and time sharing systems were used. In late 1980s client/server architecture was introduced, which improved the usability, flexibility, interoperability and scalability. [10]

In a client/server architecture number of tiers depend on three program layers i.e. presentation layer, application/business logic and database layer. Presentation layer is the top most layers in any architecture that facilitates user to interact with the system. It handles the inputs given by the user and display output on the user screen. Application or business logic layer is the middle layer which comprises of business rules/algorithms. It handles the exchange of information between presentation layer and database layer. Database layer is the bottom layer which gives services (file services, print services, communication services and database services) to upper layers. [11]

Client/server architecture can be categorized as:

- One-tier architecture

- Two-tier architecture
- Three-tier architecture
- N-tier architecture

One tier architecture

It is easy to design and implement one-tier architecture because there is no separation of program layers. All the three program layers are tightly connected with each other. One tier architecture is used when the application is simple i.e. the number of users are less and application requirements are not complex [11].

Two tier architecture

A two-tier architecture is the one in which client directly communicates with the server. It supports integrated development environment to facilitate rapid application development. The upper two layers (presentation layer and business logic) works together but database layer is separate from them. This architecture works well for small environment where users are limited. Application development time of two-tier is less than one-tier. [11]

Two-tier architecture is best in distributed environment where different workgroups are working. But its performance decreases if the number of users exceeds 100. This architecture has limited flexibility, in moving functionality of the program from one server to another. [10]

Three tier architecture

Three-tier architecture is also called three-layer architecture. It was established in 1990s to remove limitations of two tier architecture in terms of scalability; interoperability; system administration, configuration and batch jobs. Third tier introduced in three tier architecture is 'middle tier server' that is sandwiched by user interface and data management component. Function of middle tier is to manage the process by executing business logic and rules. It removed the limitation of two tier architecture by supporting hundreds of users at a time. Three-tier architecture is preferred when high performance is required along with making the user unaware of complexities of distributed client/server environment. [25]
Structure of web application can vary according to the needs but preferably it's a three-tiered application [1], [14].

N-Tier Architecture

The n-tier architecture is developed from client server architecture. Main reason for its evolution is to split an application into parts (tiers) that are maintainable and these parts are placed in a stack one on the other. In this way each tier is isolated from another. Updating an application is easy since changes need to be made in one tier not in whole application. If any changes are made in a tier they does not affect the other tier lying above or below in the stack. Standard object oriented techniques are used to deploy tiers and by connecting tiers directly in the stack. [15]

The fundamental idea of n-tier architecture [18] is to make it possible for a program to call another program, or an application to call another application. In other words there are several parts of an application which are built by different developers but they work together.

III. Key Factors

During development of any software the developers have to make some decisions at the start. Choice of architecture is one of them. Architecture is a key feature which decides that system will meet its performance and other quality objectives or not. On completion of the software, if it is found that software does not meet requirements then it's much costly to renovate it. [12] For choosing architecture that delivers required performance level at very low cost, needs to know the technical and financial constraint of the application.

Factors that affect the choice of architecture are as follows. [13]

- Performance
- Availability
- Complexity
- Cost

Performance refers to number of concurrent users that can be facilitated by the system. Most of the applications restrict limited number of user at a time to access the different services, with limited data. This limited access depends upon the requirements of the organization. While choosing the architecture its must to keep in mind which performance level is necessary.

Availability is system accessibility to the user when it is needed. Application availability timing must be known during architecture selection.

Complexity is the level of difficulty of system in user's and developer's perspective.

Cost is very essential factor during the architecture selection. Selected architecture must be cost effective for an organization. It must give the required results at very low cost. Cost may be divided into "acquisition cost" and "management cost".

IV. Design issues in Tiered Architecture

Current trends are to focus on data rather than code through tiered architecture. Several experiments related to this [19], [20], [21], [22], [18], [23] are carried out for studying design issues in tiered architecture. There are four critical issues during development i.e. loosely coupled, coarse grain communication, asynchronous and temporal.

A. Loosely Coupled

The term loosely coupled corresponds to arrangement, in which implementation change of application A does not affect other applications B and C. They still communicate and their link should not break.

Client-server applications are tightly coupled so client code is broken down when database is changed. Contrary to that three-tier or n-tier architecture are expected to be loosely coupled. To maintain this, designer should avoid platform specific or hard coded interfaces for communication with other applications. It can be done by Simple Object Access Protocol (SOAP) [24] and eXtensible Markup Language (XML) [21], [22]. SOAP model describes the standard for protocols linked with it. While XML based protocol communicates messages from one software component to the other using Hyper Text Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP) or any other standard protocol [22].

The idea behind loosely coupled is basically to decide upon data format and the protocol. In this way application A can communicate with application B without information about implementation and each application is independent.

B. Coarse-grain Communication

Coarse-grain communication implies that if application A requires student record of B, it needs the student name, father's name, personal number and address. Rather than communicating four times with B for each bit of information there is need of coarse-grain communication model. Due to which A gets whole required information once from B. It will reduce cost of communication and increases efficiency of the application.

One among possible designs to implement coarse-grain communication is transferring information as XML message [4], [26].

C. Asynchronous

Suppose that application "A" requests "B" to ship medicines. After sending request A would wait for confirmation from B that the shipment will take place or not. There is need of some agreed time to get message back from B otherwise after the specified interval error would be reported. This is known as asynchronous.

Tiered web applications need to be asynchronous, latency period must be specified by developer during coding. It is actually the acceptable time in which A gets response from B.

D. Temporal Communication

In the real world, there is a lot much conversation between tiered web applications. For example, A requests 10 different items from B through 10 messages. So it is required that this bunch of information should be maintained in some sequence. Accordingly the responses would be generated. This is called temporal communication.

V. Choosing Tiered Architecture

As it is mentioned above that architecture choice depends upon different factors like performance, availability, complexity and cost. Next step in the development of the application is to choose the right architecture for application by keeping in mind all the above mentioned factors. Let us consider the four different types of tier architecture, which are designed to meet the technical requirements of the organization.

- Vertical 2-tier architecture
- Vertical 3-tier architecture
- Horizontal 3-tier architecture
- Diagonal 3-tier architecture

All these architectures provide different level of performance, availability and complexity as shown in table 1. [13] Organization needs to select one which meets their required objectives.

A. Vertical 2-Tier Architecture

In vertical 2-tier architecture, application instances and database instances are working on a single server/domain. There is no separate server for these works. Vertical 2-tier architecture provides very good performance and also complexity in this case is low but availability is marginal as shown in table 1. [13]

B. Vertical 3-Tier Architecture

Application and database instances in vertical 3-tier architecture, works on separate server. If vertical 3-tier architecture is chosen then performance of application will be excellent. Number of user can access the application at a time. Because database instances and application instances works on separate server, so this architecture will not be complex. [13]

C. Horizontal 3-tier Architecture

In horizontal 3-tier architecture database portion is working on vertical server and application instances are working on many small servers i.e. server with 4 or fewer processors. In case of horizontal 3-tier architecture performance of the system will be good, availability will be the excellent. But system complexity is very high as shown in table 1. [13]

D. Diagonal 3-tier Architecture

Same as the horizontal 3-tier architecture, in diagonal 3-tier architecture database works on the vertical server but all the application instances works on the medium servers. If diagonal 3-tier architecture is used in application then performance of the application will be good, availability will be very good and complexity will medium. [13]

Let us consider the Tables 1, which give the summary of the technical analysis of above four architectures.

Table 1: Technical analysis of four architectures

		Architecture			
		Vertical 2-Tier	Vertical 3-Tier	Horizonta l 3-Tier	Diagonal 3-tier
Technical Parameters	Performance	Very Good	Excellent	Good	Good
	Availability	Marginal	Good	Excellent	Very Good
	Complexity	Low	Low	High	Medium

VI. Qualitative approach usgin focus groups

"*Focus group is a tool that can help to investigate user attitude, feelings, and belief on proposed representation with group interview*" [27]. Focus groups can be the good choice in web development field because it is an emerging field in computer science.

We have use the focus group in our study as the qualitative research method. Focus groups are also the best choice for this study because different organizations showed the great interest to find out the best architecture for the web application that overcome the performance, availability and complexity problems. In this way different professionals from the different organizations can share their ideas to enhance the architecture selection.

A. Objectives

"*Focus groups are conducted to obtain specific type of information from a clearly identified set of individuals*". [3]

B. Research Strategy

Focus groups are used as the research strategy because this is an emerging field in web development. Also the basic purpose of this study is that everyone takes advantage.

C. Data Collection

In qualitative research there are six methods for data collection [5]. But in our study, for data collection we have use questionnaire, interviews, observations and reviews of generated documents.

Questionnaire

According to [6] there are six different type of research questions i,e open, closed, reflective, leading, focused and loaded.

In our questionnaire most of the questions are open. Questionnaire that we have design is for every software developer, but especially for the professionals that belongs to the web development field.

Also we have made an online discussion forum and place this questionnaire on it. In response of this many independent observers give the answers of our posted questions. We have stored all the answers in our database for analysis purposes.

Questionnaire**:**

1. What kind of organization you are representing?
2. What is your qualification?
3. Your development experience?
4. How critical your architecture needs are?
5. Which architecture you use in most of the development?
6. Which factors you keep in mind when you chose specific architecture?
7. How effective is your architecture against the different factors like availability, performance, complexity?
8. What is the ratio that your selected architecture is cost effective?

Observations

After this focus groups are assigned duties to observe the different firms. One group has observed which architecture will be best choice with respect to the budget of the project. Another group had made their observation on architecture to get the best performance. Third and last group had made his observation on architecture for getting best availability solution and complexity.

Document Reviews

All the documents i,e focus group summaries, questionnaire, related articles, observation for suitable architecture for specific web related application and interviews are then reviewed and compile to store them in database for the purpose to use them for analysis.

Interviews

"Interviews are most often used to gather detailed, qualitative descriptions of how programs operate and how stakeholders perceive them" [5]. During our research we meet different software developers and take their interview to get their views on different available architecture. Then we have published their interview on our online discussion forum.

D. Data Analysis

After publishing all collected data on the discussion forum, focus group has done the analysis of that data. For better results data is divided into different categories and one focus group is assigned on one category. That focus group was responsible for the to make the summery for that data.

After completing this data is then collected to the central location. Different focus groups then make discussion again to eliminate the irrelevant material. Then this data is again published on the online discussion forum and once again request is made to the software developers to give comments on published results. After getting their comments final results were then published.

E. Validation of Results

In qualitative research reliability and validity can not be defined in same manner as in quantitative research [16]. In qualitative research, validity refers to the trustworthiness of generated research results. In our study all the generated results are validated by interpretation of observations and different comments that were posted on the online discussion forum. All the observations that were made during the study were posted on the online forum in form of interviews, suggestions and answers to questionnaire.

VII. Quantitative analysis using experiments

An experiment is a very good technique to produce data, result on given data, and to check the validity of underlying theory. In application development first step is to select the architecture that will be used during the implementation. To find out the best architecture, we can perform some experiments.

Experiments are best choice because it provides many advantages i.e better control of researcher over test variables, generation of numeric data and possibility of experiment replication. Experiments are divided into definition, planning, operation, analysis and interpretation, presentation and package. [17]

A. Definition

Experiment is done to compare the effectiveness of different available architecture with respect to their performance, availability and complexity.

We have tried to give the answers of the research questions included in this paper, on the basis of data produced in this experiment.

Objects

Objects of this experiment are available architecture, i.e vertical 2-tier architecture, vertical 3-tier architecture, horizontal 3-tier architecture and diagonal 3-tier architecture.

Purpose

Purpose of experiment is to find out which tiered architecture will be the best choice for developing web applications.

Perspective

Experiment is done according to the web project managers and developer's perspective. This experiment will be beneficial for developers and also for companies because architecture selection in any software project is key task. For better performance, a good architecture selection is very important.

B. Planning

Context

Context of this experiment is the comparison of available tiered architectures to find the better choice with best possible performance. In start, a web application was developed by using all available tiered architectures. And then this application with different implementation was tested by different developers.

Hypothesis

Null Hypothesis (H0): As web development is growing, 2-tiered architecture are affecting the efficiency of web application negatively.

Subjects

Subject of this experiment are available tiered architectures.

C. Operation

Operation is the actual execution of experiment.

Preparation

Preparations are done before the actual experiments. As it is stated before, initially a web application was developed by using all available tiered architectures. And then this one application that was developed by using different architecture was delivered to the web developers/designers.

Execution

Next task after delivering the different solution of same problem was to run these implementations and to find out which solution is best in term of availability, performance and complexity. These developers have run different implementations many times. These solutions were also executed on different platforms with different inputs to check their efficiency. After each experiment response of application was noted. At the end developers have given their opinion about vertical 2-tier, vertical 3-tier, horizontal 3-tier and diagonal 3-tier architectures on the basis of their experiments of different underlying solutions. Their results are then analyzed and represented in form of charts.

D. Analysis and Interpretation

Numerical data which we get from the above experiment is then analyzed to check, which architecture will be best in term of performance, availability and complexity. The results of that data are shown in form of the bar charts.

Performance analysis

Performance of vertical 2-tier, vertical 3-tier, horizontal 3-tier and diagonal 3-tier architecture is measured by cumulating the results gathered from designers. They provided their judgment according to their findings related to specific tier architecture in percentage.

Figure1 shows that performance of vertical 2-tier, vertical 3-tier, horizontal 3-tier and diagonal 3-tier architecture is 68.9%, 89.22%, 54.4% and 54.55% respectively. It shows that vertical 3-tier architecture is best among the other tier architectures in performance while horizontal 3-tier architecture has the lowest performance level.

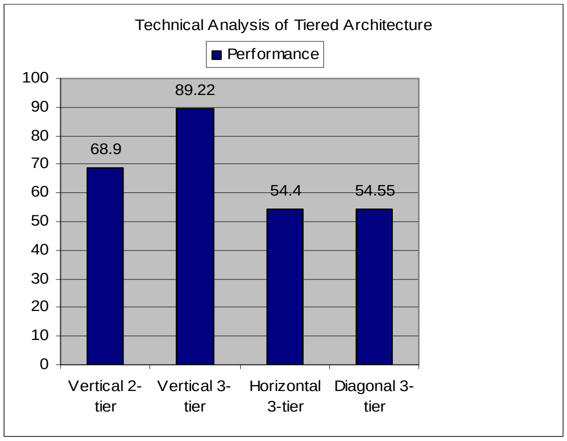

Figure 1: Performance analysis of different types of tier architecture

Availability analysis

According to the results availability of vertical 2-tier, vertical 3-tier, horizontal 3-tier and diagonal 3-tier architecture has 54.7%, 64.38%, 90.02% and 74.37% respectively as shown in figure 2. It depicts that horizontal 3-tier architecture is the best among other architectures in availability perspective, while vertical 2-tier architecture has the lowest availability level.

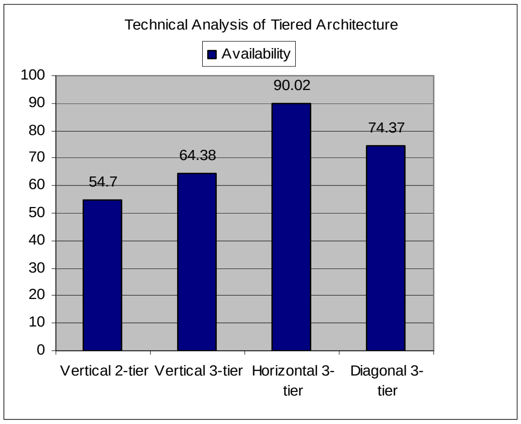

Figure 2: Availability analysis of different types of tier architecture.

Complexity analysis

Complexity analysis is represented in figure 3, which shows the result of vertical 2-tier, vertical 3-tier, horizontal 3-tier and diagonal 3-tier architecture as 53.06%, 54.4%, 89.54% and 69.88% respectively. It shows that horizontal 3-tier architecture is most complex than other tier architectures, but vertical 2-tier architecture has lowest complexity level.

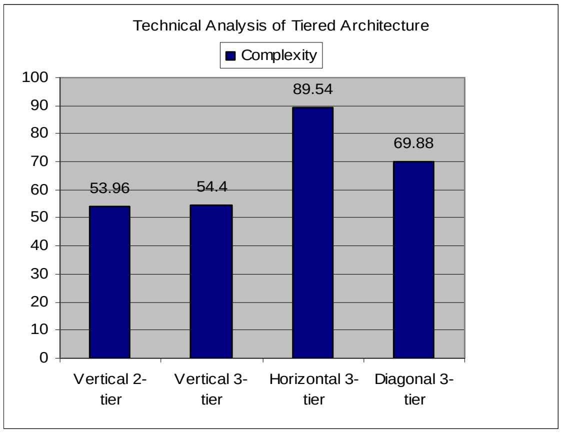

Figure 3: Complexity analysis of different types of tier

VIII. Conclusion

It is concluded that some specific architecture should be for web applications to attain better performance. It gives whole application development team to focus on single architectural model. Probability of success of an application increases in terms of meeting requirements and needs of the user. Each tier architecture has different level of performance, availability and complexity. The experiment result shows that vertical 3-tier architecture is best among others.

References

[1] Xue Liu, Jin Heo and Lui Sha. “Modeling 3-Tiered Web Applications”. *Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems. 2005,* Volume 00, Pages: 307 – 310

[2] Roy T. Fielding and Richard N. Taylor. “Principled design of the modern web architecture”. In *ICSE 2000 Limerick Ireland*, 2000.

[3] Dr. David W Steward, Prem M Shamdasani, Dennis W. Rook, “FOCUS GROUPS, Theory and Practice”, Sage Publications 2007, ISBN: 076192583X.

[4] J. Herman, XML Internet takes off—part 2. *Business Communications Review, Hinsdale* **31** 6 (2001), Pages: 22–26.

[5] Harvard Family Research Project, “Detangling Data Collection: Methods for Gathering Data”, August 2004.

[6] David Leigh, “Group Trainser’s Handbook: Designing and Delivering Training for Groups”, Kogan Page Publisher, August 2006, ISBN: 978-0749447441.

[7] Jeffrey Putz. ‘Application Architecture’ *Maximizing ASP.NET: Real World, Object-Oriented Development,* Addison Wesley Professional, 2005.

[8] Christian Eichinger, *Web Engineering: The Discipline of Systematic Development of Web Applications,* John Wiley & Sons, Ltd., West Sussex, England. ISBN-13:978-0-470-01554-4

[9] Sven Ziemer. *An Architecture for Web Applications.* Essay in DIF 8914 Distributed Information Systems, 2002.

[10] D Sadoski. *Client/Server Software Architectures – An Overview*. Software Engineering Institute, 2005.

[11] Will Fastie. *Understanding Client/Server Computing. PC Magazine*; “Enterprize Computing;” February 9, 1999; page 229-230.

[12] Lloyd G. Williams & Connie U. Smith. ‘PASASM: “a method for the performance assessment of software architectures”, *Proceedings of the 3rd international workshop on Software and performance*, 2002. Pages: 179 – 189.

[13] Sun Microsystems. *How To Guide: Deploying the Right Architecture for the Application Tier [online]*. http://www.sun.com/emrkt/innercircle/newsletter/0704feature_howto.html [Accessed 10 April 2009].

[14] Abhinav Kamra, vishal Misra & Erich Nahum,’ Controling the performance of 3-tier web sites: modling, design and implementation’, *Proceedings of the joint international conference on Measurement and modelling of computer systems*, 2004. Pages: 414 – 415.

[15] Samuel Aslund. “A study of technologies supporting development of software with an N-tier architecture”, 2001.

[16] Nahid Golafshani, “Understanding Reliability and Validity in Qualitative Research”, *the Qualitative Report* Volume 8, December 2003.

[17] Claes Wohlin, Per Runeson, Martin Host, Magnus C. Ohlson, Bjorn Rgnell, and Anders Wesslen,”*Experimentation In Software Engineering: An Introduction*”, Kluwer Academic Publishers, ISBN: 0792386825, 1999.

[18] S.H. Simon, eCommerce Solution for Business and IT Managers, ISBN 0-07-137188-5, McGraw-Hill Publishing, 2001.

[19] A. Brown, eCommerce components for the internet, The Business Forum Journal, 9297 Burton Way, Suite 1, Beverly Hills, California 90210, USA.

[20] T. Chester. High-performance Web sites: ADO versus MSXML. *Dr. Dobb’s Journal; San Mateo* **26** 10 (2001), pages: 81–87.

[21] N. Gunton, SOAP: Simplifying distributed development. *Dr. Dobb’s Journal, San Mateo* **26** 9 (2001), pages: 89–93.

[22] S. Lais , SOAP, other protocols specify security for XML. *Computer world, Framingham* **35** 28 (2001), pages: 37–40.

[23] M. Vizard, IT infrastructures are woefully unprepared for the demands and stresses of e-business. *InfoWorld, Framingham* **22** 2 (2000), pages: 22–27.

[24] T. Modi , Clean up your wire protocol with SOAP, Part 2 Use Apache SOAP to create SOAP-based applications. *JavaWorld* April (2001); (http://www.javaworld.com).

[25] Darleen Sadoski, Santiago Comella-Dorda, 2000. *Three Tier Software Architectures [online]*. http://www.sei.cmu.edu/str/descriptions/threetier_body.html [Accessed 21 March 2009].

[26] J. Fontana , Where Middleware and XML converge Web services. *Network World, Framingham* **18** 39 (2001), pages: 54–57.

[27] Riccardo Mazza, “Evaluating information visualization applications with focus groups: the CourseVis experience”, *BELIV´06: Proceedings of the 2006 AVI workshop on Beyond time and errors: novel evaluation method for information visualization*, May 2006.