Issues and Strategies of Reliability in Web Application

Misbah Perveen
Bahria University Karachi
Pakistan

Usman Waheed
Senior Assistant Professor
C & SE Department

M. Waleed Kamran
Bahria University Karachi
Pakistan

ABSTRACT

In web application Reliability is the main concern. It basically have the factors which makes web application strong, usable, mature and effective. It deals with the rate of failure in web application.

Reliability deals with many issues. In this research performance and scalability issue, availability issue, failure cause, application fault and application error are discussed briefly. The key issues to make web application reliably is the performance and scalability, and availability of the web application or server.

To come over these issues there are many techniques like software/ code refactoring, software clustering, replication and software fault tolerance but this research says that applying software/ code refactoring techniques and software fault tolerance to reduce the chances of availability issue, failure cause, application fault and error, and performance and scalability issue in web application, would be easy and better option among all.

Reliability is the mandatory element of every web application. In future testers conduct reliability testing to reduce the chances of performance and scalability, and availability issues and to ensure web application success.

Keywords

Performance Issue, Scalability Issue, Downtime, Availability Issue, Application Fault, Replication, Software/ Code Refactoring, Software Clustering, Fault Tolerance and N-versioning.

1. INTRODUCTION

One of the most important aspect of software quality is Software reliability. Customer expectations are very high in web applications. Reliability testing concept is inspected in terms of overall, in collective and sometimes specific factors between the objects and from the sample to sample constancy of factors. Mainly, it is defined as the probability of computer software's process completion in an identified surroundings for an identified time [1-3]. However, the Web environment presents many new challenges and requires adapted or newly developed techniques based on the characterization of the Web, its usage, and related issues [4].

Reliability is treated as vital concern, as it is known that the web applications are up to date, they are updated as time passes. Accurate results constantly and no or less downtime are those term which are also treated as reliability of the data produced from software Web Applications.

2. ISSUES

Reliability of a web application is decreased due to downtime of server or sometimes due to application bugs. Downtime is unavailability of the server it occurs due to low scalability and effect the performance of the web application. The other issue is failure cause which is the divergence of the output of web application and expected output by the user. Another is the issue of application failure arise due to the bad programming which will lead the web application fail and rejected by the user. At last, application error caused by the wrong user input which will increase the performance issues of any web application. The detail of these issues are discussed below:

2.1. Performance and Scalability Issues

Performance and scalability of any web application is highly granted. If application is taking more time then acceptable or expected time by increasing number of users or requests, it is said to be non-performing, non-scalable or degraded [5]. To prevent application unreliability, the safety measures should be adapted to develop a product which have good performance and scalability level. [6]

For example in Online Transection Process like online shopping, when the number of online customers are increased it effect the performance of the web application. It shows that the web is not as scalable as it must be. When it provide the unwanted performance it will be rejected by the user and it lead the application to the failure.

2.2. Availability Issue

When the server is not available for transections, or the functionalities which must be provided by web application are not achieved then it will cause unreliable and unavailable web application. [7]

For example in Online Ticket Reservation, it is expected that Online Ticket Reservation system provide 24/7 hours service so it will be available any time, whenever user want to reserve a seat. But, the issue has been seen at the peak times, the server is very slow, sometimes their reservation is not done because of unavailability of the server. In fact, a Web application can sometimes be available to some users and not to others. A customer in Tokyo might have problems accessing your Web site while a user in New York might have no trouble at all. This will annoy users to do their desired task and it lead the application to failure.

2.3. Failure Cause

Web application are not being used effectively when problems occurred in normal working. It might be routine maintenance or software upgrading. Sometime it is unwillingness of users to use the application as they are not skilled enough and lack technical knowledge. [8, 9]

Let's consider the issue discussed in performance and scalability section. If the scalability is not controlled and the performance level is less than expected then it will cause the failure of the application.

2.4. Application Fault

The application fault are due to the wrong programming of the application, coding problem in business logic, bad user interface problem and page navigation problem. The reliability is badly effected when the programming fault occurs. The whole application would be rejected by the users as they noticed a fault. They either want to post a complaint or they don't want to come back on this web application again. [10]

Now, consider the example discussed in the availability section. When the Online Ticketing or Ordering is done it shows the calculated bill. If there is a logical issue in programming of application and it shows wrong amount of bill, the reliability factor of these type of application will be lost and the high severity level will also shows that uses are not liking to use this application again. [11]

2.5. Application Error

Errors of the application may be due to the wrong attempt by the users and the application is not effectively dealing with the errors occurred due to the wrong attempts. It is necessary for any application to be error free. So that, it will be a reliable web application.

For example in web application if the validations checks are not attached with the input fields which are being filled by the user side. Then, when user enters the wrong input or select incorrect option, it will cause an error and there is no any check programmed to cope up with that error it will be frustrating for the users to deal with it and it is also shown that those web application which do not give response to the users will produces errors.

3. STRATIGIES

There are many strategies which can be used to solve or reduce the reliability issues in web applications like improving the nonfunctional requirements by Software/ Code Refactoring, maintaining backups by Replication and Clustering or resolving fault issues by Software Fault Tolerance and many others. These strategies are briefly discussed below:

3.1. Software/ Code Refactoring

To improve the nonfunctional attribute of any web application the code refactoring techniques are used. It is use to change the structure of code but it does not affect the functionality of the web application. [13, 18]

Code refactoring can refactor the code, like removing the extra check which have less priority level. Create super class by moving most frequently used method (Move up Method) in main class so that the declaration of same method won't be repeated again and again, or split large class in different subclasses (Extract Method) according to the situation. Accordingly move field to the super class or move field from super class to the sub class to reduce the chances of multiple declaration of same object (Split Temporary Variable). The other solution is to marge to small class in one large class (Inline Class) to reduce different object creation or marge to small classes to create one Extract class. These techniques are also used to reduce the chances of failure in web application, application fault and application error.

3.1.1. Solution to Performance and Scalability

Instead of using or maintaining different servers these techniques are more effective to increase the performance and scalability of the web application.

3.1.2.Solution to Application Fault

When there are issues like spaghetti code which reduce the reliability of web application. Code refactoring techniques can format code in a way it produces good results and structure full programming of web application help us to identify the piece of code which will cause error.

3.1.3.Solution to Application Error

Errors in working application is not easy to solve and very difficult to identify but the easiest solution is code refactoring technique. These technique reduce the extra code and formulate the whole business layer, it help us to check and/ or test the code.

3.2. Software Clustering

Software clustering is a function which link different server in one cluster [12]. Like different functionality in one particular system. Software clusters are designed for applications that have long-running in-memory state or frequently updated data. Typical uses for server clusters include file servers, print servers, database servers, and messaging servers. [14]

3.2.1.Solution to Performance and Scalability

Software clustering should be opt for any application so that whenever there is any issue occurred in a single module is will not affect the whole application down and it will balance the load by clusters to sustain large number of users.

3.2.2.Solution to Availability

It will help to easily replace, modify or add new module or server to increase the availability and scalability of application which is also known as parallel programming.

3.3. Replication

Replacement of server from one to another is done by the replication strategies in case of unexpected downtime or even scheduled downtime. [6, 15]

To reduce the downtime there must be any backup system like secondary server which is updated as the standby server (primary server) is updated on each and every transaction.

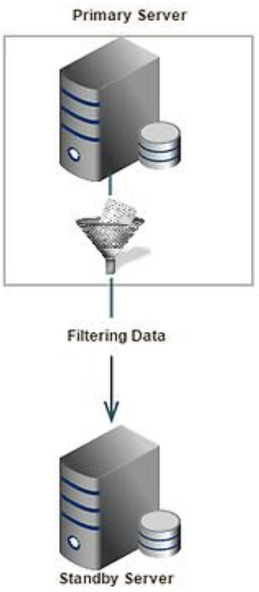

Figure 1 Server Data Filtration in Replication Strategies

In Figure 1, it is shown that Online Ticket Reservation system must incorporate standby server and update filtered data correspondingly to reduce the chances of unavailability of the server.

3.3.1.Solution to Performance and Scalability

Whenever downtime is noticed immediately secondary server is installed at primary server place to maintain the performance of the web application.

3.3.2.Solution to Availability

Secondary server is monitored and updated time to time with filtered data because when first server is down or some time it is needed to maintain, the secondary server is attached with main application so that, user used application any time.

3.4. Software Fault-Tolerance

Technique which is used to make web application or any software system fault less and help to be protected from high level software failures is software fault tolerance. [16, 17]

3.4.1.Failure Cause

To make any application failure free the sub technique fault prevention is used. It can be tacker by formal verification, conduct test cases, avoiding incomplete specification, avoid ambiguous specification, and maintain sequence between levels especially where one level is dependent upon previous level.

3.4.2.Application Failure

N version programing is the sub technique of software fault tolerance. Making different version of any application will constantly reduce the chances of faults of the application. If it is being noticed that there is a fault in web application the developer must update like add new module to manage the application fault and release the new version which is fault free.

3.4.3.Application Error

By using N versioning, the error of the applications are removed but, by reducing the chances of error occurrence it is necessary to use some controls and actions to limit user so that user won’t enter incorrect data. Appling validation constrains are the best solution for this issue.

4. BENCHMARK

The issues which are discussed previously, occurred due to absence of the reliability factor. Which can be resolved by some previously discussed strategies. Table 1 relate issues with the strategy to be opt to reduce the chances of that particular issue.

The most significant strategy is using software/ code refactoring and software fault techniques. It is easy to make web application scalable, effective, and fault free before launching and also, it is not more expensive then maintaining different backups and clusters.

Table 1: Shows reliability issues and strategies to resolve them

Issues	Strategies	How
Performance and Scalability	Software Code Refactoring	To use move up method, extract method, inline class or by extract class code refactoring techniques instead of maintaining different servers these techniques are more effective. [13, 18]
	Software Clustering	Whenever there is any issue occurred in a single module is will not affect the whole application down and it will balance the load by clusters to sustain large number of users. [14]
	Replication	Whenever downtime is noticed immediately secondary server is installed at primary server place to maintain the performance of the web application. [6, 15
Availability	Software Clustering	It will help to easily replace, modify or add new module or server to increase the

		availability and scalability of application which is also known as parallel programming. [12]
	Replication	Secondary server is monitored and updated time to time with filtered data because when first server is down or some time it is needed to maintain, the secondary server is attached with main application so that, user used application any time. [6, 15]
Software Failure	Software Fault Tolerance	It can be tacker by formal verification, conduct test cases, avoiding incomplete specification, avoid ambiguous specification, and maintain sequence between levels especially where one level is dependent upon previous level. [16, 17]
Application Fault	Software Code Refactoring	When there are issues in code of web application. Code refactoring techniques can format code in a way it produces good results and structure full programming which help us to identify the piece of code which will cause error. [13]
	Software Fault Tolerance	Making different version of any application will constantly reduce the chances of faults of the application. If it is being noticed that there is a fault in web application the developer must update like add new module to manage the application fault and release the new version which is fault free. [16]
Application Error	Software Code Refactoring	Software code refactoring technique reduce the extra code and formulate the whole business layer, it help us to check and/ or test the code. [13]
	Software Fault Tolerance	By using N versioning, the error of the applications are removed but, by applying validation constrains this issue will be resolved easily. [16, 17]

5. CONCLUSION

This paper discussed detailed about reliability issues and strategies in web applications. And also map the best strategies to the issues mentioned and also provide a bird view of implementation of strategies according to the issues. This paper also shows that the software/ code refactoring and software fault tolerance are the best strategies to reduce the reliability issues from web application. It is shown that in future the reliability testing would be mandatory element of every web application development process to make it reliable. It is being plan to develop tools to measure the reliability of web application. Every hard work should lead us to a more effective and operational strategy to achieve and maintain high reliability for Web applications.

6. REFERENCES

[1] Musa, Jhon D., "*Software reliability engineering: more reliable software, faster and cheaper.*" Tata McGraw-Hill Education, 2004.

[2] Nixon, B.A. (2000), *"Management of performance requirements for information systems"*, IEEE Transactions on Software Engineering, Vol. 26, pp. 1122-46.

[3] Wen, R.B. (2001), *"E-business reliability with web performance management"*, Proceedings of the IEEE International Symposium on Software Reliability Engineering (ISSRE), Hong Kong, 27-30 November.

[4] R.Manjula, P., Sriram, E.A, *"Article:reliability evaluation of web applications from click-stream data."* International Journal of Computer Applications 9(5), 23–29 (November 2010), published By Foundation of Computer Science

[5] Thongpapanl, Narongsak, and Abdul Rehman Ashraf. *"Enhancing online performance through website content and personalization."* Journal of computer information systems 52.1 (2011): 3.

[6] Rabinovich, Michael, and Oliver Spatscheck. *"Web caching and replication."* Boston, USA: Addison-Wesley, 2002.

[7] Zeng, Liangzhao, et al. *"Qos-aware middleware for web services composition."* Software Engineering, IEEE Transactions on 30.5 (2004): 311-327.

[8] Isermann, Rolf. *"Supervision, fault-detection and fault-diagnosis methods—an introduction."* Control engineering practice 5.5 (1997): 639-652.

[9] Guo, Y. and Sampath, S., 2008, October. *"Web application fault classification-an exploratory study."* In Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement (pp. 303-305). ACM

[10] Dobolyi, K. and Weimer, W., 2010, February. *"Addressing high severity faults in web application testing."* In Proceedings of the IASTED International Conference (Vol. 677, No. 075, p. 240)

[11] Guo, Yuepu, and Sreedevi Sampath. *"Web application fault classification-an exploratory study."* Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement. ACM, 2008.

[12] Doval, D., Mancoridis, S. and Mitchell, B.S., 1999. *"Automatic clustering of software systems using a genetic algorithm."* In Software Technology and Engineering Practice, 1999. STEP'99. Proceedings (pp. 73-81). IEEE.

[13] Binkley, David, et al. *"Tool-supported refactoring of existing object-oriented code into aspects."* Software Engineering, IEEE Transactions on 32.9 (2006): 698-717.

[14] Barroso, Luiz André, Jeffrey Dean, and Urs Holzle. *"Web search for a planet: The Google cluster architecture."* Micro, Ieee 23.2 (2003): 22-28.

[15] Plattner, Christian, and Gustavo Alonso. *"Ganymed: Scalable replication for transactional web applications."* Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware. Springer-Verlag New York, Inc., 2004.

[16] Avizienis, Algirdas. *"The N-version approach to fault-tolerant software."* IEEE Trans. Software Eng. 11.12 (1985): 1491-1501.

[17] Conallen, Jim. "Modeling Web application architectures with UML."Communications of the ACM 42.10 (1999): 63-70.

[18] Fowler, Martin. *"Refactoring: improving the design of existing code".* Pearson Education India, 2009.