15th International Research/Expert Conference
"Trends in the Development of Machinery and Associated Technology"
TMT 2011, Prague, Czech Republic, 12-18 September 2011

IMPROVING PERFORMANCE OF A WEB BASED SOFTWARE APPLICATION

Sabedin A. Meha
University of Prishtina, Department of Mechatronics
Bregu i Diellit pn, Prishtina, Kosova

Agni H. Dika, Isak R. Shabani, Hysen N. Sadiku
Faculty of Electrical and Computer Engineering
Bregu i Diellit pn, Prishtina, Kosova

Avni R. Hajdini
Faculty of Contemporary Sciences and Technologies
Ilindenska bb Tetovo, Macedonia

ABSTRACT
In this paper the performance of a web based software application at courts all over Kosovo is presented. Due to the large number of users writing and reading at the same time in database the server failed to respond. As a result of such overload the reports were unable to be generated and the work of application in all institutions has been stopped. To overcome the problem, it was necessary to find what caused the problem, to find a quick temporary solution for the application to work and to find the best solution that will also improve the performance by reducing server load.
Keywords: Web Based Software Application, Database, Server Load

1. INTRODUCTION
Web applications are among the fastest growing classes of software systems in use today. These applications are being used to support a wide range of important activities: business functions such as product sale and distribution, scientific activities such as information sharing and proposal review, and medical activities such as expert-system based diagnoses. Given the importance of such applications, faulty web applications can have far-ranging consequences on businesses, economies, scientific progress, and health [1].
After the war in Kosovo, new institutions took place trying to offer services for citizens and for business community. With years passing, the need for software to do specific services became more and more necessary. Kosovo has around 60 courts located in major cities, all of them supervised from the Judicial Council located in Prishtina. The supervision is not so easy, especially when dealing with courts revenues. So a need for software that will register all court revenues to monitor court performance online became very important task. For this reason Kosovo Judicial Council decided to choose a company that will build a web based software application that will generate reports about all the types of revenues. The chosen company developed the Software for Court Revenues, which now has around 150 users all writing and reading in one single central database. When first started to be used, the server failed to respond generating overloading errors.
This paper presents this problem and also the solution that now makes the Software for Court Revenues one of the most used in public institutions of Kosovo. Authors of this paper were closely involved in developing and implementing this software.

2. ASP.NET APPLICATIONS

ASP.NET applications always work in conjunction with a web server – a specialized piece of software that accepts requests over HTTP (Hypertext Transport Protocol) and serves content. When you're running your web application in Visual Studio, you use the test web server that's built in. When you deploy your website to a broader audience, you need a real web server, such as IIS.

Web servers run special software to support mail exchange, FTP and HTTP access, and everything else clients need in order to access web content.

The easiest job a web server has is to provide ordinary HTML pages. When you request such a file, the web server simply reads it off the hard drive (or retrieves it from an in-memory cache) and sends the complete document to the browser, which displays it. In this case, the web server is just a glorified file server that waits for network requests and dishes out the corresponding documents.

When you use a web server in conjunction with dynamic content such as an ASP.NET page, something more interesting takes place. On its own, the web server has no idea how to process ASP.NET tags or run C# code.

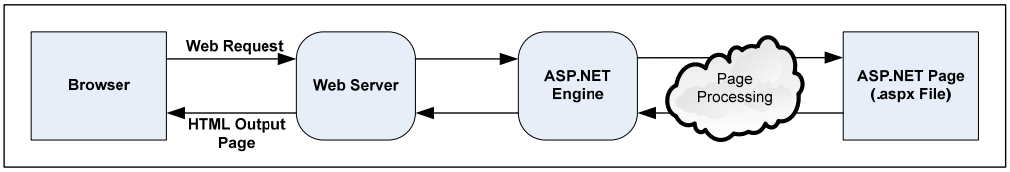

Figure 1. How IIS handles an ASP file request

However, it's able to enlist the help of the ASP.NET engine to perform all the heavy lifting. Figure 1 diagrams how this process works for ASP.NET pages. For example, when you request the page Default.aspx, the web server sends the request over to the ASP.NET engine (which starts automatically if needed). The ASP.NET engine loads the requested page, runs the code it contains, and then creates the final HTML document, which it passes back to IIS. IIS then sends the HTML document to the client [2].

For developing the software, we used ASP.NET and SQL Server 2008. For designing reports we used Crystal Reports for .NET component that comes integrated with Visual Studio .NET pack.

2.1. Problem with reports

Software for Court Revenues started its application in all courts in the beginning of this year. We had not enough time for testing the software and we had to take the risk to release the software without testing it in the real environment. However some basic testing procedures took place in the company and it seemed that the software is working fine. Figure two explains some basic steps how user interacts with application hosted in web server.

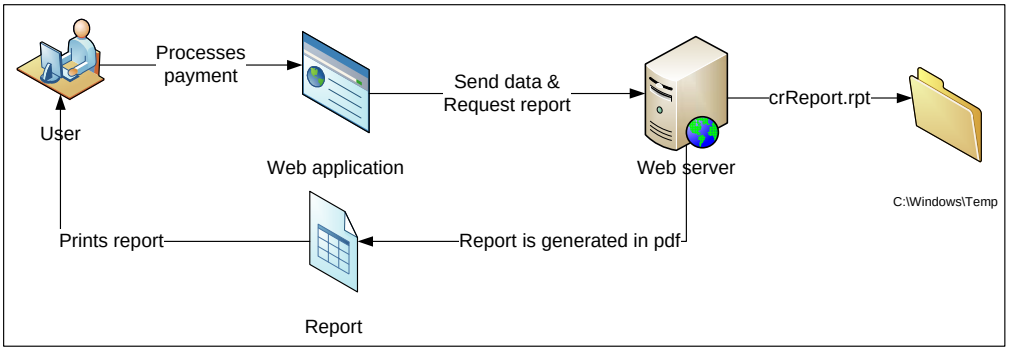

Figure 2. User Interaction with web server

The user first logs on the system by entering his/her valid credentials. After that, the main thing that user has to do with the software is to process a payment that citizens has to do. After filling appropriate fields in the web form, the software sends data to the database and also a request for server to generate a report in pdf format that then will be printed out from the officer that uses the software, and used by the citizen. When generating a report in pdf format, the server sends in **C:\Windows\Temp** folder a crystal report file that is needed for filling pdf file with fields and with the appropriate design. These files had size of no more than 208KB.
In the first day of software application, users started to work with new software and the work was going pretty well; the software was fast enough, the user interface was good and the reports were generated properly. Then after an hour of proper functioning, suddenly the software through this error:

```
Error: Sys.WebForms.PageRequestManagerServerErrorExpection: Load
Report failed.
```

After carefully searching for the reason why this happened, we realized that the problem was with the Windows Server Temporary Folder. When the total amount of the crystal report files reached 8MB, the server could no more load and convert crystal reports files to the portable digital format (pdf).
The users were annoyed with this problem and they started to work manually like they did before implementing software.

3. TEMPORARY SOLUTION

It was no easy task to deal with more than 50 software users when they experienced software failure. They all started to call and to ask what is going on and how to proceed since the citizens started to get frustrated.
We started to search for the solution and tried following approaches:

- Increasing bandwidth size,
- Increasing temporary folder size,
- Redesigning code for generating reports,
- Stopping other applications from consuming server resources, etc.

All these things didn't solve the problem and the only way to start the software to function again was to clear the Temp Folder, i.e. to delete temporary crystal report files. It took approximately one hour of good functioning and then again the error stopped the software.
While searching for other solutions, we had to create a .bat file to do following things:

- Stop IIS
- Delete content from Temp Folder
- Restart IIS

The .bat file was created by using simple text editor like notepad with following commands on it:

```
iisreset /stop
DEL /S /Q "C:\WINDOWS\Temp\"
iisreset /start
```

The good thing with this approach was that the users could work with software but one bad thing was that when stopping and restarting IIS all users lost their sessions and so had to log into the software again, so there was a possibility to lose data if they were working in the moment when the IIS was stopped.

4. SOLUTION

The solution that permanently solved the problem was the usage ReportViewer control that is a part of great amount of controls which comes with Visual Studio and the avoidance of Crystal Reports. This approach required complete report reconstruction. In this case, the server converted directly the content of the ReportViewer control to the pdf format, without writing anything to the server memory. Basically, this should be used from the beginning of the project, but since Crystal Reports offered a lot

of possibilities in design and functionality, we chose this approach without fully considering risks of overloading.
In the Figure 3 it is presented the basic flowchart for all users that use the Software for Court Revenues in Kosovo.

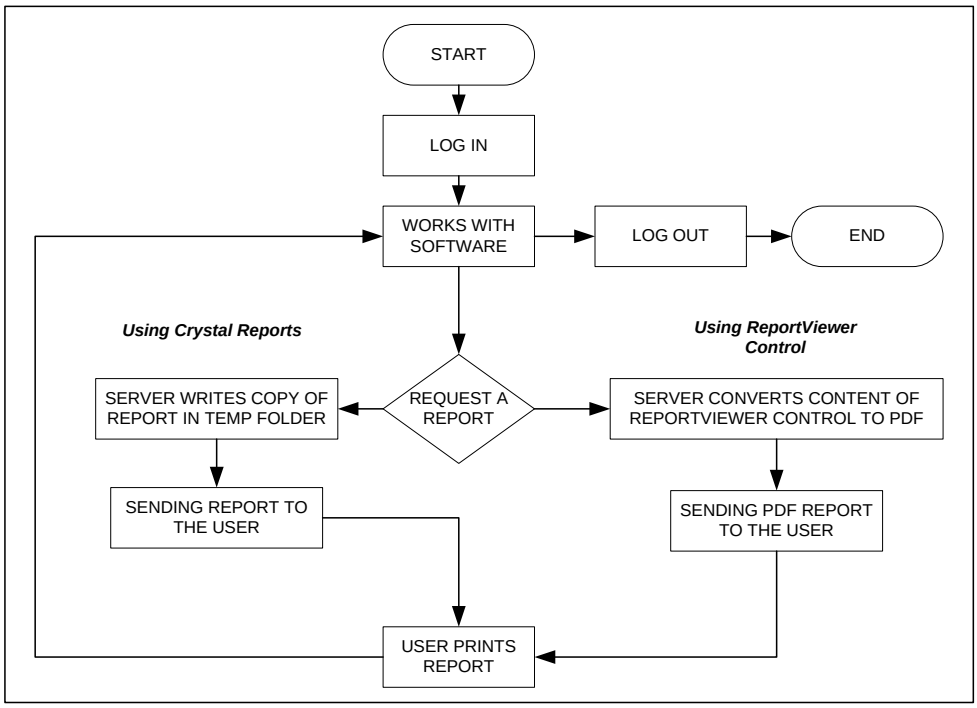

Figure 3. User basic flowchart in two approaches

The difference between two approaches is the way of generating the reports. Generating reports using ReportViewer control reduces server memory usage, since there is no need to write and read data to system temporary folder. After implementing this approach, the users are having much better experience working with software.

5. CONCLUSION

Risk analysis is not difficult to perform properly, and the benefits in project planning are great. However, if it is not done properly, the results can be disastrous as the analysis can severely understate risk and lead to unsatisfactory conclusions about project viability [3]. Software designers and developers need to consider all aspects when designing and developing software, especially how the software will react when it faces the real environment. Testing software in company environment is very important but you need also to take into consideration the test in real environment before releasing software for usage.

6. REFERENCES

[1] S. Elbaum, S. Karre, G. Rothermel: Improving Web Application Testing with User Session Data,
[2] M. MacDonald: Beginning ASP.NET 3.5 in C# 2008 from novice to professional, Second Edition, Apress 2007
[3] S. Meha, L.Gashi, B. Fejzullahu.: E-Governemnt – Wealth Management System Risk Identification, 14th International Research/Expert Conference 2010.