

# Performance analysis of XML\_HTML Converter

Preetika  
Student, Computer Science  
DPGITM  
Gurgaon, India  
preetikakaushik@gmail.com

Neetika  
HOD, Computer Science  
DPGITM  
Gurgaon, India  
nitikasagi@gmail.com

Manish Kumar  
IT  
Cargoflash Infotech Pvt. Ltd.  
Gurgaon, India  
Kumar.kumarmanish@gmail.com

**Abstract**— as one knows that websites are generated in HTML and XML is used to transport and store data. And most of configuration files are generated in XML. One generally use asp.net web server controls for creating web pages. But life cycle of an asp.net control is too big and asp.net server controls take much more to get rendered. In the case of Web requests, the actual rendering is performed by a client's Web browser or other viewing device. The task of the ASP.NET page framework is to send HTML (or text in another markup language such as XML or WML) in response to a Web request. It takes a lot of time to convert an asp.net server control to HTML. In this paper I am presenting a solution to this problem by creating a converter which will convert XML files with predefined tags into HTML. The HTML controls will be generated from XML files. And will measure the performance by testing two sites one with pages having asp.net server controls and other with controls generated using XML files.

**Index Terms**- Purpose, Testing, Response Time Graph, Aggregate graph.

## 1. INTRODUCTION (XML\_HTML CONVERTER)

XML\_HTML Converter is a tool which takes XML files with predefined tags as input and generates equivalent HTML file. Although .NET allows one to insert an XML file and create asp.net page from it. But it is basically used to display data. By using this converter one can generate HTML from predefined XML tags. One can Create textbox, label, dropdown, calendar controls etc. They will take less time to get rendered as there is not such a big life cycle as there is in asp.net server controls. All properties and validations of all controls are directly invoked using this single file. If there is any modification this can be applied to this single file.

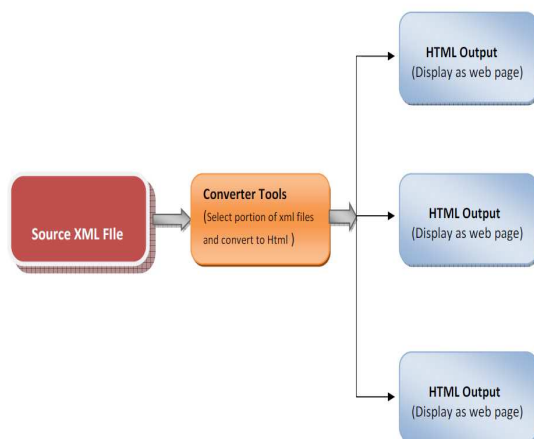


Figure 1 Creating HTML Output from XML file

## 2. ASP.NET PAGE CYCLE

ASP.NET page has a life cycle in which it goes through a number of steps. When one creates a custom control, one must have knowledge of page life cycle in order to correctly initialize controls with their properties. The life cycle of a control is based on life cycle of page, events raised for a control by page is more than the events available of page alone.

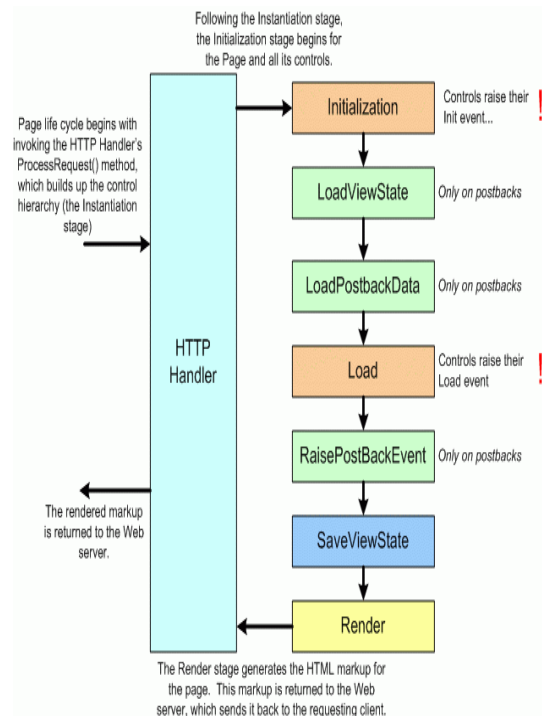


Figure 2 ASP.NET Page Cycle

### **3. PURPOSE**

Whenever client requests a page server respond to it by sending an HTML or (XML files).The response time of a web page containing asp.net server control is large. As asp.net controls are too heavy and they take lot of time to get rendered, one needed a mechanism by which the time taken by server for responding to a web page becomes small .The purpose of HTML\_XML converter is to minimize the response time of any requested web page, leading to better performance, by minimizing the steps followed to convert any requested page to equivalent HTML file.

### **4. TESTING**

In this research paper, I will show the results obtained by testing two websites one having asp.net server controls and other having controls generated using XML files. The test will be conducted using open source tools JMeter and the Web Performance and load test tools of visual studio 2013.

Steps to be followed during testing:

1. Firstly select two websites one which was created using XML files and the converter name it as Website1 and second created using pure asp.net server controls name it as Website2.

2. We will use Aggregate Graphs and Response Time Graphs to measure the response time of any requested web page and aggregate graphs shows the aggregate time used.

#### **Response Time Graph**

The Response Time Graph draws a line chart showing the evolution of response time during the test, for each labeled request. If many samples exist for the same timestamp, the mean value is displayed.

### Response Time Graph

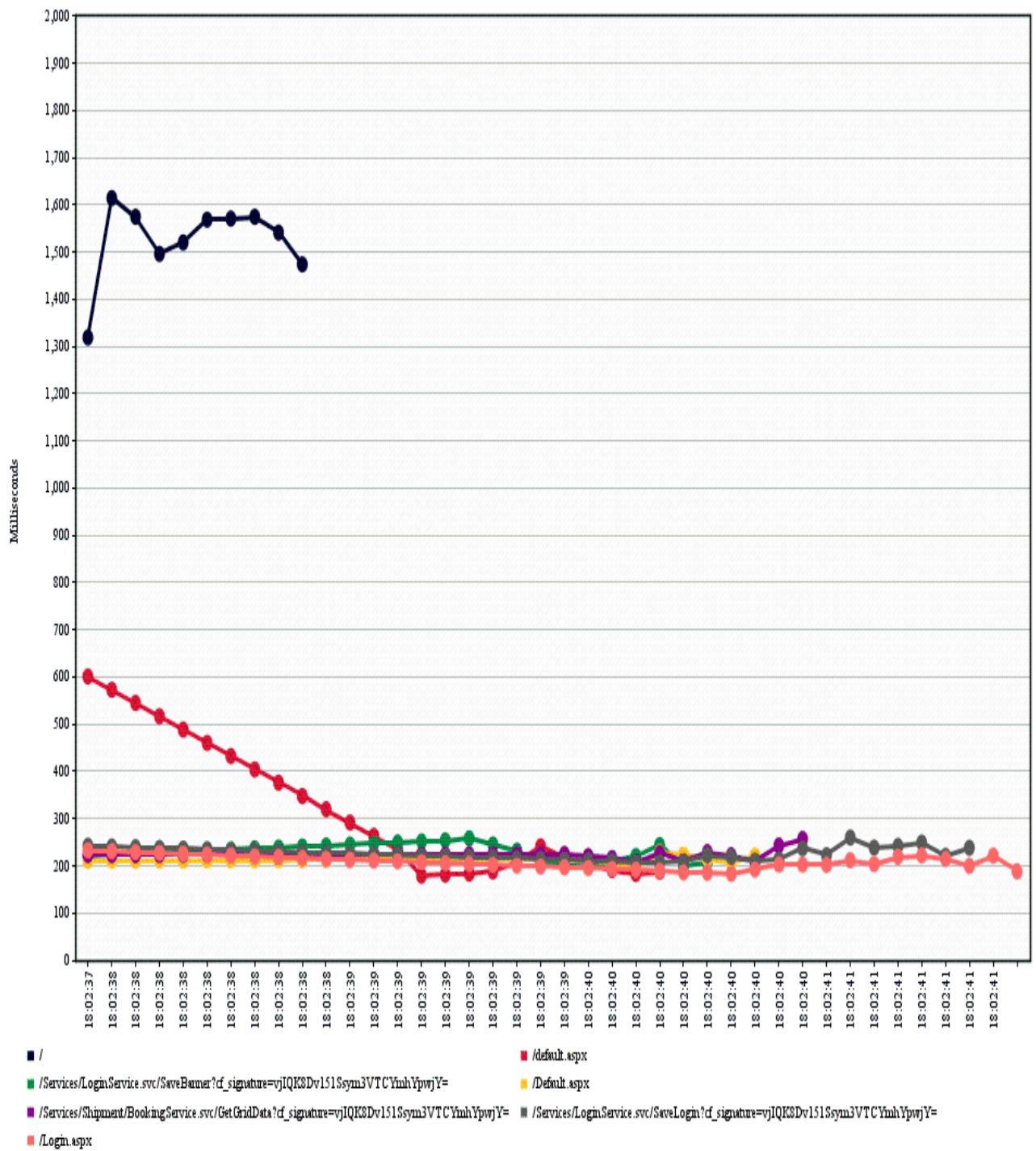


Figure 3 Response time graph of Website1

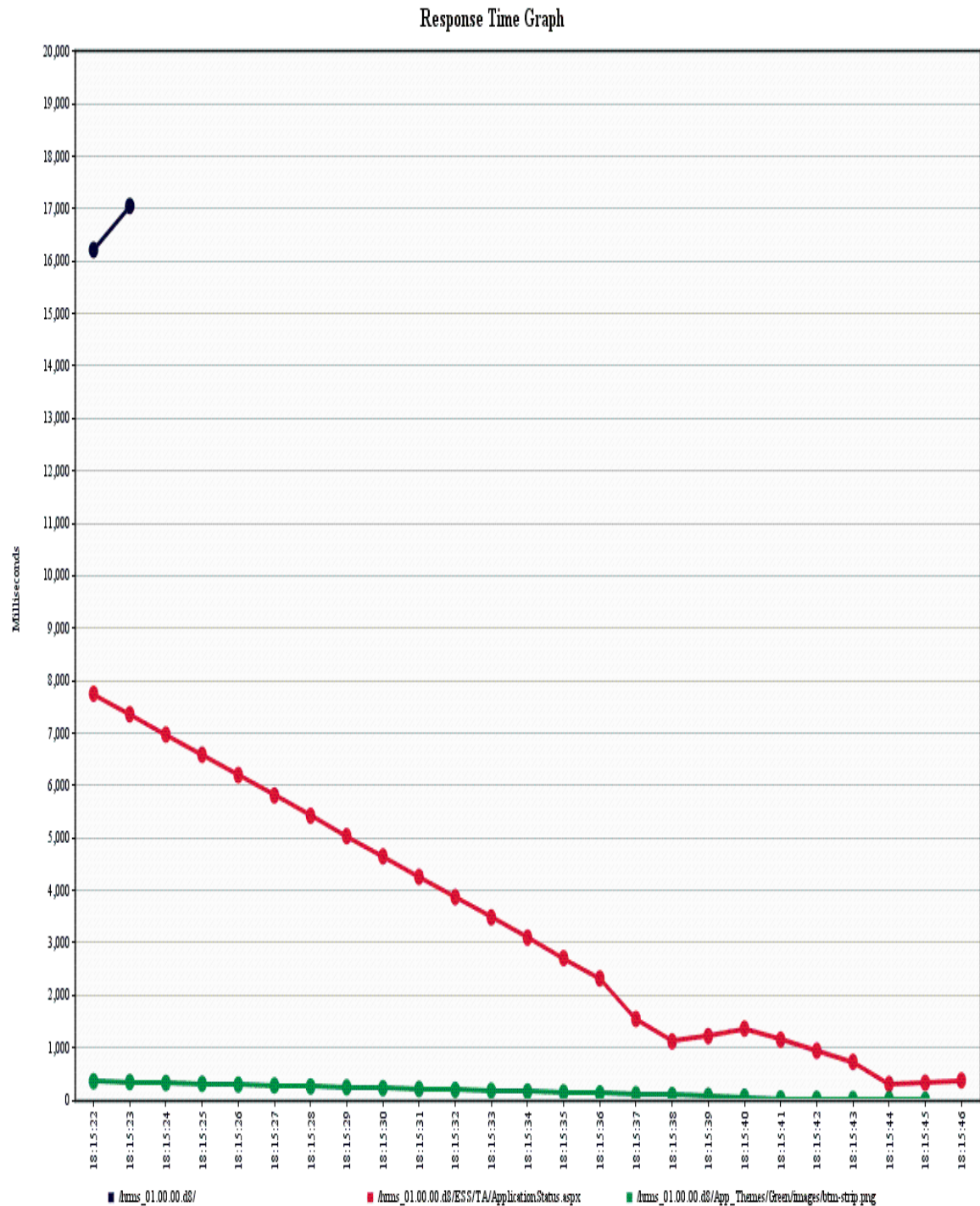


Figure 4 Response Time Graph of Website2

- (1) X Axis: It contains date and time.
- (2) Y Axis: Define a custom maximum value for Y Axis in milli-seconds. Define the increment for the scale (in ms) Show or not the number grouping in Y Axis labels.

Website2 as one can see from graphs. The max response time of Website1 is 1600 ms and Website2 is 17000 ms respectively.

Response time of Website1 generated using XML file is much better than the response time of

Aggregate Graph

For each request, it totals the response information and provides request count, min, max, average, error rate, approximate throughput (request/second) and Kilobytes per second throughput. Once the test is done, the throughput is the actual through for the duration of the entire test. Aggregate graph is an easy way to generate bar graphs and save the graph as a PNG file.

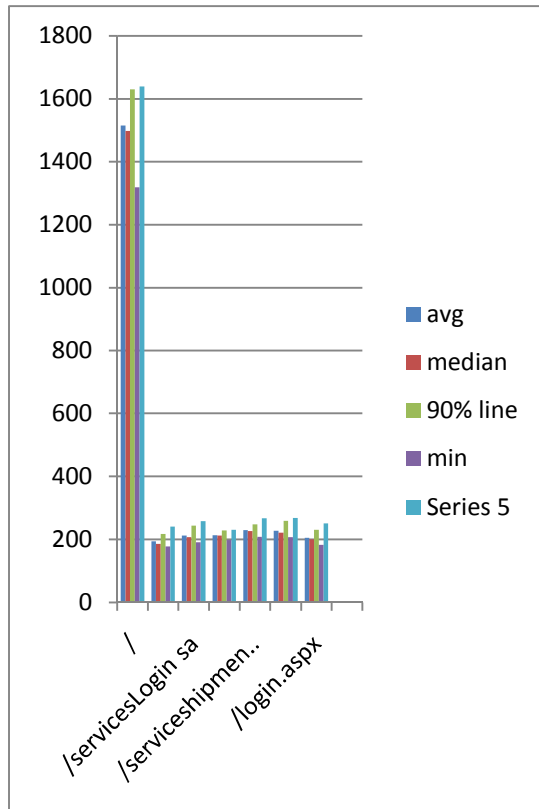


Figure 5 Aggregate graph for Website1

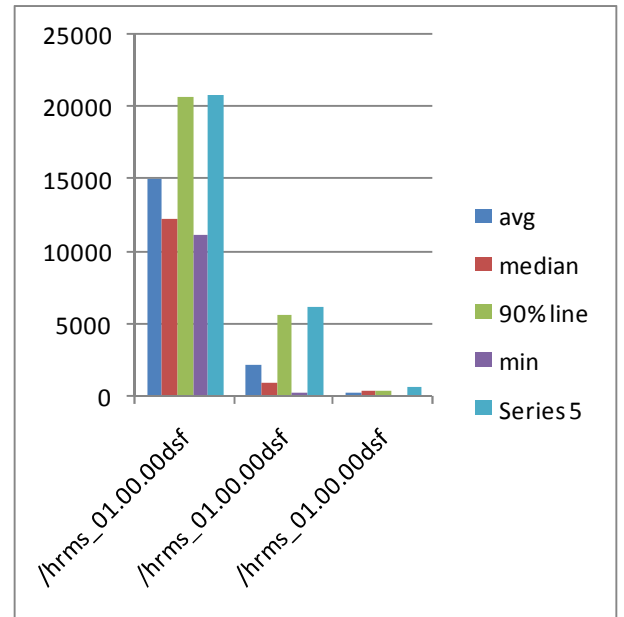


Figure 6 Aggregate graph for Website2

- Average - The average time of a set of results
- Median - The median is the time in the middle of a set of results. 50% of the samples took no more than this time; the remainder took at least as long.
- 90% Line - 90% of the samples took no more than this time.
- Min - The shortest time for the samples with the same label
- Max - The longest time for the samples with the same label

Aggregate graph of Website1 generated using XML file is much better than the response time of Website2 as one can see from graphs. The max time of Website1 is 1700 ms and Website2 is 21000 ms respectively.

## 5. CONCLUSION

It can be concluded from test results that the response time and aggregate graph time of Website1 is much smaller than the Website2 and so does the performance of the website. It saves lots of time of client and saves lots of resources as time and cost. Making changes is easy as it is once implemented in a single file can be propagated to all.

## **REFERENCES**

- [1] <http://www.w3schools.com/aspnet/>
- [2] <http://www.w3schools.com/xml/>.
- [3] Programming in C# A PRIMER by E. Balagurusamy.
- [4] <http://www.tutorialspoint.com/>
- [5] [http://jmeter.apache.org/usermanual/component\\_reference.html#Aggregate\\_Graph](http://jmeter.apache.org/usermanual/component_reference.html#Aggregate_Graph)
- [6] <http://www.codeproject.com/Articles>
- [7] [http://link.springer.com/chapter/10.1007%2F978-3-642-25989-0\\_52](http://link.springer.com/chapter/10.1007%2F978-3-642-25989-0_52)
- [8] <http://codeverge.com/asp.net.getting-started/for-thesis-asp.net-as-a-programming-l/683473>
- [9] <http://publications.lib.chalmers.se/records/fulltext/118608.pdf>
- [10] <http://tomasp.net/academic/>
- [11] <http://msdn.microsoft.com/en-us/library>