

To Implement an Analyzer for Evaluation of the Performance of Recent Web Development Frameworks

Subhash V. Pingale¹, Vidya G. Phule², B. M. Kore³

^{1,2,3}Department of Computer Science & Engg., University of Solapur
SKN Sinhgad College of Engineering, Korti, Solapur, MS, India

¹sub.pingale83@gmail.com, ²phulevg@gmail.com

Abstract— Everything you see, click, and interact with on a website is the work of front-end web development. Client-side frameworks and scripting languages like JavaScript and various JS libraries like AngularJS, jQuery, Node.js, etc. have made it possible for developers to develop interactive websites with performance improvement. Today the use of web is raised to such an extent that web has evolved from simple text content from one server to complex ecosystem with various types of contents spread across several administrative domains. This content makes the websites more complex and hence affects user experience. Till now efforts has been done to improve the performance at server side by increasing scalability of back-end or making the application more efficient. The performance of client side is not measured or just tested for usability. Some widely used JavaScript benchmark suites for measuring the performance of web browsers. The main focus of such a benchmark is not to measure the performance of a web application itself, but to measure its performance within a specific browser. There is wide variety of literature being available to measure the complexity of web pages and determining the load time. The aim behind our project is that to characterize the complexity of web pages built using different web development technologies like AngularJS, jQuery, AJAX (Client side web development technologies) so as to compare and analyze the usage of proper web development technique. In this paper we have used AngularJS as a case study to evaluate its performance against other frameworks like jQuery and AJAX.

Keywords - AngularJS, Browsers, jQuery, AJAX, MVC, SPA, JSON.

I. INTRODUCTION

Now a day's it's almost impossible to conceive a life without internet. The World Wide Web (WWW) as the largest global information media store through which user can share, read, and writes data through computers connected with internet. As the usage of web is increased to such an extent that web has evolved from simple text content from one server to complex ecosystem. Web site today fetches content not only from servers hosted by its providers, but also from a range of third-party services like advertising agencies, content distribution networks (CDNs), and analytics services [1]. Also, rendering a single Web page today involves fetching several objects from multiple servers under different administrative domains. Because of these factors web pages becomes more complex, hence the user experience gets degraded. With the increasing diversity of client platforms for

accessing the Web, it is important for browser developers to identify the aspects of Web page complexity that impact use experience [1]. They need tools and techniques to evaluate the impact of these services on users.

There is a wide variety of literature being available to measure the complexity of web pages and determining the load time. Here our aim is to provide the Analyzer tool which is capable of profiling web page complexity against certain performance metrics actually impacting web page performance. The benefits of this work will be complexity will be reduced to some extent using proper developmental technique hence load time will get reduced and the user experience will be better.

Now-a-days JavaScript and its libraries provide interactive approach to static web pages [2]. Here we have taken AngularJS, jQuery and AJAX as case studies for comparison as they are widely used for web application development.

I. RECENT WEB DEVELOPMENT FRAMEWORKS OF JAVASCRIPT USED IN PAPER

This paper takes into account latest widely used javascript frameworks viz. AngularJS, jQuery and AJAX. They are introduced in nut-shell as below.

A. AngularJS

AngularJS is a JavaScript framework and specifically designed to create SPAs. It is a library written in JavaScript. AngularJS provides the web application with Model-View Controller (MVC) capability. In contrast to the traditional MVC architecture like SpringMVC, where the website is rendered from the server side, with Angular the view is generated in the browser using its Model which holds all the required data [5]. The controller takes care of the interactions between the HTML page and Model. The upside here is, there is no server side calls involved in these operations and everything is done on the client side with cached data.

AngularJS abstracts the server calls to a separate layer to avoid code redundancy across multiple views for a gateway built with pure HTML5, JavaScript and REST services. AngularJS manipulates the DOM automatically. Because of that the burden of developers gets reduced as view automatically gets updated when data changes. It

makes the view lightweight, by decoupling the view rendering from the server side.

B. Single Page Applications

A single-page application (SPA) is a web application or web site that fits on a single web page with the goal of providing a user experience similar to that of a desktop application. In an SPA is retrieved with a single page load, or the appropriate resources are dynamically loaded and added to the page as necessary, usually in response to user actions. Interaction with the single page application often involves dynamic communication with the web server behind the scenes [6].

In nutshell, how SPA work can be represented as follows.

1. Browser initiates a request to server
2. Server sends a webpage (consists of HTML and JavaScript)
3. Browser loads the entire webpage
4. User clicks on a link (initiates a new request and asks for the information that is only going to need to update the webpage)
5. Server responds with JSON data
6. Browser loads the JSON data into the existing webpage to update it.

This mode of working significantly makes the processing faster and efficient thereby improving the performance of the page.

C. jQuery

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. jQuery is one of the widely used JavaScript libraries today [2].

jQuery, at its core, is a DOM (Document Object Model) manipulation library. The DOM is a tree-structure representation of all the elements of a Web page and jQuery simplifies the syntax for finding, selecting, and manipulating these DOM elements. jQuery also provides a paradigm for event handling that goes beyond basic DOM element selection and manipulation. The event assignment and the event call back function definition are done in a single step in a single location in the code. jQuery also aims to incorporate other highly used JavaScript functionality.

D. Ajax

Ajax (Asynchronous JavaScript and XML) is a set of web development techniques at client side to create asynchronous Web applications. With Ajax, web applications can send data to and retrieve from a server asynchronously (in the background) without interfering with the display and behavior of the existing page [2].

Ajax allows for web pages, and by extension web applications, to change content dynamically without the need to reload the entire page. In practice, modern implementations commonly substitute JSON for XML.

Ajax is not a technology, but a group of technologies. HTML and CSS are used in combination to mark up and style information. The DOM is accessed with JavaScript to dynamically display the information presented.

Now for discussion about the previous work done to analyze the performance of web technologies and their evaluation, not a single application is available at present. Literature is available to measure the complexity of web pages. Browsers built tools are used to measure the loading time by analyzing log files. Also the performance for server side web technologies is measured, while our approach is to measure and evaluate the client side web technologies performance.

II. APPROACH

The process to be followed for the work requires performance metrics to be defined for the comparison of frameworks. For web pages the metrics that affect user experience can be outlined as below.

A. Page Load Time

Loading time of a webpage is the time required to request and render the entire requested web-page in the browser of client.

B. Transition Time

Transition time defines the time required to make a transition from one webpage to another by visiting a link.

C. Throughput

Throughput is characterized by No. of jobs completed per second [3]. Here in our context we'll redefine it as No. of requests processed per second.

D. Response Time

Response time (in case of AJAX)-i.e. RTT i.e. time required for request to travel from client to server and back again.

E. Resource Consumption

Resource consumption is the amount of hardware resources used at a certain point in time.

F. Bandwidth Or Data Consumption

Bandwidth expresses the maximum data flow in and out of a computer system.

G. Complexity Of Webpage - It can be calculated by calculating No. of objects downloaded, Size of the

objects downloaded and No. of distinct non origin servers contacted by the webpage [1].

The empirical results generated will provide the developer with detailed information about performance (execution speed) at client side.

III. MATHEMATICAL MODEL

We will calculate the timing and page complexity metrics as discussed in above section. Let us consider the Loading Time for the module developed in AngularJS, jQuery and AJAX is L_{an} , L_{jq} , L_{ax} , respectively. Also,

Transition Time as T_{an} , T_{jq} , T_{ax}
Throughput as TP_{an} , TP_{jq} , TP_{ax}
Response time as R_{an} , R_{jq} , R_{ax}
Resource consumption as RC_{an} , RC_{jq} , RC_{ax}
Bandwidth as B_{an} , B_{jq} , B_{ax}
Complexity of webpage as C_{an} , C_{jq} , C_{ax}

Now, consider the sets
S1 to represent metrics values for AngularJS
S2 to represent metrics values for jQuery
S3 to represent metrics values for AJAX

So these sets can be represented as,
 $S1 = \{ L_{an}, T_{an}, TP_{an}, R_{an}, RC_{an}, B_{an}, C_{an} \}$
 $S2 = \{ L_{jq}, T_{jq}, TP_{jq}, R_{jq}, RC_{jq}, B_{jq}, C_{jq} \}$
 $S3 = \{ L_{ax}, T_{ax}, TP_{ax}, R_{ax}, RC_{ax}, B_{ax}, C_{ax} \}$

The results will clear that $v(S1)$ is least than others i.e.
 $v(S1) < v(S2) < v(S3)$

IV. CONCLUSION

By implementing an analyzer tool which will compare the performance of recent client side web development technologies on the basis of certain parameters like – loading time, transition time, complexity of code, etc., the results generated will aid in evaluation of the benefits and drawbacks of the web development technologies. It’s concluded from the results that AngularJS gives best performance as compared to jQuery and AJAX. Also it’s specially designed for SPAs and not conventional websites.

REFERENCES

[1] Michael Butkiewicz, Harsha V. Madhyastha, and Vyas Sekar, “Characterizing Web Page Complexity and Its Impact”, *IEEE/ACM TRANSACTIONS ON NETWORKING*, JUNE 2014, VOL. 22, NO. 3.
[2] Rohit Dhand, “Reducing web page post backs through jQuery Ajax calls in a Trust based Framework”, *2012 International Conference on Computing Sciences*, 2012,978-0-7695-4817-3.
[3] Kai Lei, Yining Ma, Zhi Tan, “Performance Comparison and Evaluation of Web Development Technologies in PHP,

Python and Node.js”, *IEEE 17th International Conference on Computational Science and Engineering*, 2014.
[4] Marija Selakovic, Michael Pradel, "Performance Issues and Optimizations in JavaScript: An Empirical Study", *SOLA, October 2015*.
[5] Viknes Balasubramanee, Chathuri Wimalasena, Raminder Singh and Marlon Pierce, “Twitter Bootstrap and AngularJS: Frontend Frameworks to expedite Science Gateway development”, *IEEE Transactions 2013*.
[6] N. Chapman, J. Chapman, "JavaScript on the Server Using Node.js and Express", *MacAvon Media 2014*.
[7] AngularJS Tutorials by Google Inc., [Online] Available: <https://docs.angularjs.org/guide/introduction> - Developer Guide for AngularJS
[8] Single Page Applications from Microsoft website. [Online]. Available:<https://msdn.microsoft.com/en-us/magazine/dn463786.aspx>