

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320-088X
IMPACT FACTOR: 5.258



IJCSMC, Vol. 5, Issue. 9, September 2016, pg.171 – 178

PERFORMANCE TESTING: IMPROVISED WEB SERVICE TESTING TOOL

Shikha Dhiman

Research Scholar

*Department of Computer Science
Career Point University Hamirpur, India*

Pratibha Sharma

Asstt. Professor

*Department of Computer Science
Career Point University Hamirpur, India*

ABSTRACT: *With today's changing trends and technology the web services technology is one of the most significant web technologies. Due to the rapid popularization of web application i.e. web is evolving and expanding continuously and so are its services, the web application performance testing is gaining wide attention. Users interact with web services by means of Internet and as a result web services are getting more complex but are also easily accessible and usable by user. Hence, it is considered as an important topic in the field of software testing. This has motivated the web experts to propose different methods and different tools to test the new web services efficiently. In this paper we present a tool supported approach to perform efficient performance testing of a web service. This paper investigated open source web service testing tool Httpriider for the lack of parameter i.e. throughput which is then added to it through the modification of the tool. Functional and non functional testing is done with the help of WSDL parsing and performance testing is performed by identifying the changes made thereafter.*

Keywords: *web services, software testing, testing tool, httpriider, performance testing.*

I. Introduction

Web services are very important in the field of software testing. So, different tools have been developed and designed to enhance the use of web services. The benefits and challenges in web services is for those services to be offered in a very useful, flexible, effective and secure way. Web Applications are difficult to test in comparison to traditional applications especially in terms of performance testing such as unpredictable load, response time etc. Therefore different tools have been developed and designed to use different types of services. We can define web service as a service oriented architecture or a software system which designed to provide an interoperable application to application interaction through network, where the systems interact with applications using Simple Object Access Protocol (SOAP). Software testing is a process used to measure the quality of the software. Quality of the product depends upon various parameters such as response time; throughput etc. Response time is the time between the request and first response that is received by the user and throughput is the amount of work that can be performed by a system or component in a given period of time. Throughput depends on transactions/sec or byte/sec [1].

Software testing is one of the crucial phases of the software development life cycle. It is important as it helps the user to measure different parameters of quality like correctness, robustness, usability, integrity and security. Software testing is a set of activities conducted with the intent of finding errors in software. It also verifies and validate whether the program is working correctly without bugs or errors. Testing is the phase where the errors remaining from all the previous phases must be detected. Hence testing performs a very critical role for quality assurance and for ensuring the reliability of software [2]. Common problems faced while conducting performance testing are test environment, testing tool selection, ambiguous requirements, Unavailability of valid data etc. There are a wide variety of performance testing tools available in market. The selection of tool for testing will depend on factors such as, license cost, hardware requirements, platform support etc [3].

Software Testing Objective: The objective of software testing is to eliminate the bugs from the software.

- A superior test case is one that has a prospect of finding an undiscovered error.
- A superior test is not redundant.
- A superior test should neither be too simple nor too complex.
- To check if the system does what it is expected to do.
- To check if the system is “Fit for purpose”.
- To check if the system meets the requirements and be executed successfully in the Intended environment.
- Executing a program with the intent of finding an error.
- Developed test cases for the valid and invalid conditions.
- Apply testing at early stage.
- Testing must be apply by third party for it’s effectiveness.

Software Testing Techniques: In this section we will discuss the software testing techniques. Software testing techniques is categorized in two main areas:

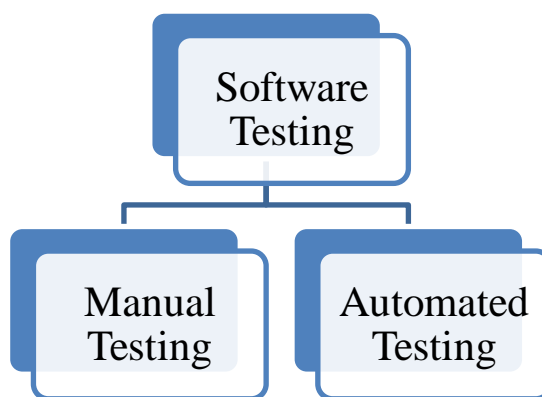


Figure 1: Software Testing Techniques

A. Manual Testing:

“Manual Testing” is also known as static testing. Manual testing will take more time than the automated testing and it will also take huge amount of human resources. It will be perform by the Analyst, Developer and the testing team early at the stage. It will include the different techniques:

- **Walkthrough:** It is the formal review technique. In this technique, the programmer and designer leads the members and interested parties and participants ask questions for intending to finding out the errors, violation of standards and other problems.
- **Informal Review:** Informal reviews may often be unnecessarily expensive (because of time- wasting through lack of focus), and frequently provide a sense of security which is quite unjustified by the relatively small number of real defects found and repaired.
- **Technical Review:** The purpose of a technical review is to arrive at a technically superior version of the work product reviewed, whether by correction of defects or by recommendation or introduction of alternative approaches. While the latter aspect may offer facilities that software inspection lacks, there may be a penalty in time lost to technical discussions or disputes which may be beyond the capacity of some participants.
- **Inspection:** The goal of inspection is to identify the defects in the product. It includes the software requirement specification and test plan. For example, if the inspector is checking the software requirement specification for finding the defects and if it finds than, he disagrees on those defects.

B. Automated Testing:

“Automated Testing” required some special software to control the execution of the test and also compare the actual outcomes with the predicted outcomes. It will include the different techniques:

- **Correctness Testing:** In this technique, the tester may or may not know the internal details of the product. It will tell the behaviour of the product. In this category we can use either black box testing or white box testing.
- **Performance Testing:** It will mainly concern with the performance of the product. The main objective of performance testing is to improve the throughput of the product. The prime goal of performance testing is to focus upon the responsiveness and stability of the system under workload. It will also evaluate the attributes like reliability and scalability. In this category we can perform the load testing, stress testing, soak testing and spike testing [4].

Manual testing is carried out by the tester. Walk through, informal review; technical review and inspection are the techniques of manual testing. But the problems are it is very time consuming process and requires more effort. So, automation testing is used to solve these problems.

To find out the problems the verification, error detection and validation is the main aim of testing. Testing use the words of successful and unsuccessful in the results of test cases. If the test case did not find an error then it's called a successful test run whereas if it discovers an error then it's called an unsuccessful test run [5]. Software testing tools test the process and targeted to specific test environment. The environment may be functional, performance or exceptional testing etc.

Functional testing is also called black box testing. It is based on requirement and functionality. The test cases for functional testing are based on the requirement and specification. Basically it is the method of generating test cases that are independent of software internal structure. Black box testing occur throughout the software development life cycle and software testing life cycle i.e. in regression testing, acceptance testing, unit testing, integration testing and system stage[6]. Increase in the use of software testing tools has lead to new revolution. Varieties of software testing tool available have multiple usage areas. Few of them can be named as test management tools, functional testing tools and load testing tools etc. In the market various functional testing tools are available for testing the web services automatically. These tools are also used to test the application software and web applications such as web sites [7].

In a business environment, where hundreds of web services working together makes testing of web services is essential to achieve the goal of robust and successful web services. A web service is a relatively new concept which can be used to call methods located on a server. It is an interface that describes a collection of operations. The interface hides the implementation details of the service, allowing it to be used independently of the hardware or software platform on which it is implemented and also independently of the programming language in which it is written. Basically web service is a self-contained, self-describing, modular application that can be published, located and invoked by other software systems across the web. This communication is done by using a standardized protocol called Simple Object Access Protocol (SOAP) which is build up by using an Extended Markup Language (XML) schema. Web services work using HTTP and XML which makes the web services technology in heterogeneous in nature because HTTP is transfer independent and XML is data independent. The heterogeneous nature of web services offers various advantages like interoperability, usability, use of standardized communication protocol and deployability [8]. The web service architecture contains four main parts such as XML, SOAP, Web Service Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI) [9].

XML:-The web service architecture is built up using XML. XML is a meta language that has a well defined syntax and semantics. The syntax and semantics "self describing" features of XML make it a simple, powerful mechanism for capturing and exchanging data between different applications.

SOAP:-Soap is a protocol based on XML. It is used for communication between a Web Services provider application and a client application. A SOAP message can be either a response or a request. These SOAP messages are based XML. So it will be easy to communicate with SOAP within a heterogeneous network, where web services can be found at various internet web servers. A Soap message contains four parts envelope, header, body and fault. Figure 2 shows the structure of SOAP message. In the structure the "Envelope" defines the start and the end of the message. The "Header" contains any optional attributes of the message. The "Body" tag is the place where the request and response data are stored. Within the Body tag a "Fault" tag may exist that provides information about errors that occurred while processing the message.

```
<?xml version="1.0"?> <SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://www.w3.org/2001/12/soap-
envelope" SOAP-
ENV:encodingStyle="http://www.w3.org/2001/12/soap-
encoding"> <SOAP-ENV:Header> ... </SOAP-
ENV:Header> <SOAP-ENV:Body> ... <SOAP-
ENV:Fault> ... </SOAP-ENV:Fault> </SOAP-
ENV:Body> </SOAP_ENV:Envelope>
```

Figure 2: A SOAP Message Structure

WSDL: Web service interface is written in WSDL. WSDL document contains the information about the location of the web service and specifications of the method that are present. The method specification includes parameters and the return type of the method. When a web service is published it is actually the WSDL-documents that are published, not the functionality of the web service. It specifies the location of the service, the operation or methods. The WSDL document structure contains the elements such as type, binding, port type, services, message and operations. In document the “types” provides information about any complex data types. The “messages” define abstract description of the data being exchanged, “port types” define abstract collections of operations and “binding” defines concrete protocol and data format specification for a particular port type.

UDDI: - UDDI is a registry where web services are stored or the yellow page of the web services. It is an XML based standard for describing, publishing, and finding web services. Registry and a set of WSDL port type definitions for manipulating and searching that registry are the two main parts of the UDDI.

Web Services Tools:- Web applications can be defined as computer programs consisting of client-server architecture that allows visitors to Submit and retrieve data over the Internet using their web browser and presented to the user within their browser dynamically by web application through server in specific format. Web services are very important in the field of software testing. So, different tools have been developed and designed to enhance the use of web services. Test tool enables the testers to create, manage and execute test for a particular environment and maintained a specific test for a particular application. Various open source web service testing tools are available in the market with different features and functionality. These test tools are free for the users to use. The tools can be downloaded from the internet or can be obtained by the vendor without any charges [2]. The study in this paper is based on the httpriider open source web service testing tool.

The paper can be studied under the following sections: Section I, gives the brief introduction in which software testing, software testing tools, architecture of web services and web service testing tools has been explained. Section II provides the brief overview of httpriider web service testing tool. In Section III, Modification in httpriider according to desired parameter has been given. Section IV gives the result and discussion. Section V concludes the paper along with scope for future work and then references.

II. httpriider: - Brief Overview

Software testing is an important to determine the quality of the software. Testing tool enables the testers to create, manage and execute test in a particular environment, maintained for specific test for a particular application. Testing tools are used to make testing easier and help simulate test environment for a system under testing. The main aim of testing is verification, error detection and validation in order to find the problems and fix them to improve the quality of the software products. Testing Tools automate the process of testing and are targeted to specific test domain. Testing tool automates testing process and make testing easier. Performance testing tools are used for different types of performance testing like load test, stress test, volume test and strength test [10].

A good software testing tools contains various features such as efficiency, maintainability, functionality, usability and adaptability. Though testing is becoming more vast and complex but still accuracy and reliability are the basic needs. There are many web service testing tools are available in software market. So choosing a good software testing tool for any of product will increase the performance of software, improve the communication between the developers, produce the reliable and secure software and it will raise the customer satisfaction [2]. For this research, httpriider, the open source web service testing tool has been used to evaluate a web service.

HttpRider: HttpRider [11] is a lightweight, open source testing tool for performance and stress testing of web sites. It is written in c#. It records http scenarios by creating http requests using Fiddler proxy [12] and can create multiple user threads. It helps user to customize automation parameters like query string and post parameters. It binds them to a customized data source in order to mimic non-identical requests. It records all the requests generated through the browser and then replay it with multiple concurrent users. Also requests can be edited as the convenience of the user. It also enable user to save scenarios to the disk. The study shows that the httpriider is simple and easy to use. So, this research based on the httpriider web service testing tool. The tool can be downloaded from “<http://httpriider.codeplex.com>”. Httpriider is licensed by GPLv2. For this study “http://himachal.nic.in/index.php?lang=1&dpt_id=23” web service has used.

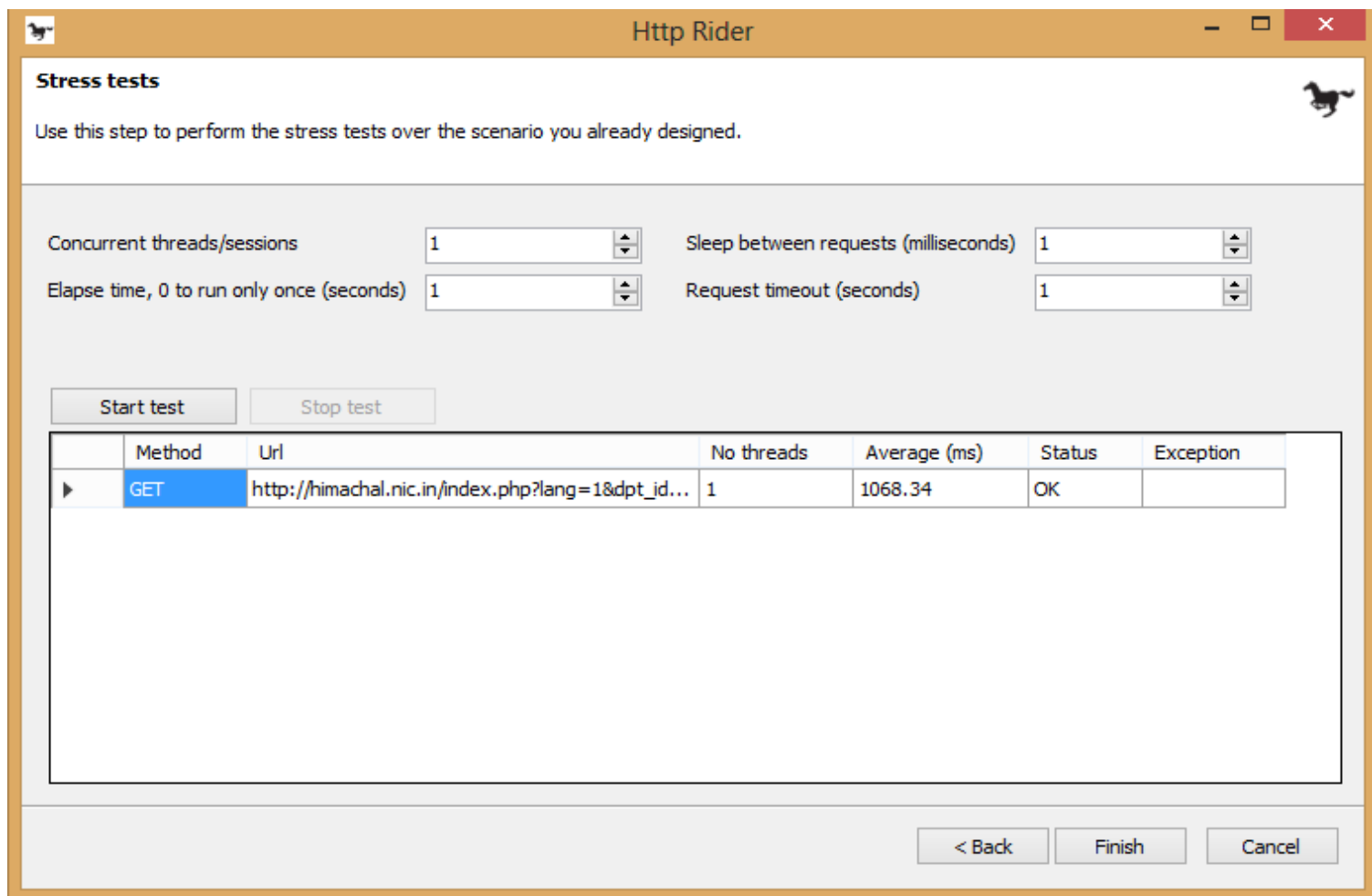


Figure 3: Web Service Result.

Initially the tool was tested for the given web service and the results are shown in Figure 3. It can be seen that the parameter i.e. response time is present while parameters i.e. bytes and throughput which are important for efficient web service testing are absent in http rider tool. Tool provides only the average response time of the web service provided as the input to the tool. So the initiative has been taken to accommodate these important above mentioned parameters into the tool.

III. Modification in http rider

This section represents efforts done to modify the initial tool to accommodate changes.

1) System Requirements

The tool was run on an Intel Core i3 2.40 GHz processor machine with 3GB RAM, Microsoft Windows 8.1, and 1-2 Mbps Internet connection. Testing of the tool requires configuration which in turn includes installation, setting up test environment, data collection, selection of parameters and analytical survey. The sample web service i.e. “http://himachal.nic.in/index.php?lang=1&dpt_id=23” was tested on the selected configure tool.

2) Approach Followed

For this research practical approach has been used. Experiments were conducted based on the desired parameters. Thus observed results were collected and compiled for the analysis and evaluation to determine the efficiency of the tool. “http://himachal.nic.in/index.php?lang=1&dpt_id=23” web service was tested using http rider tool.

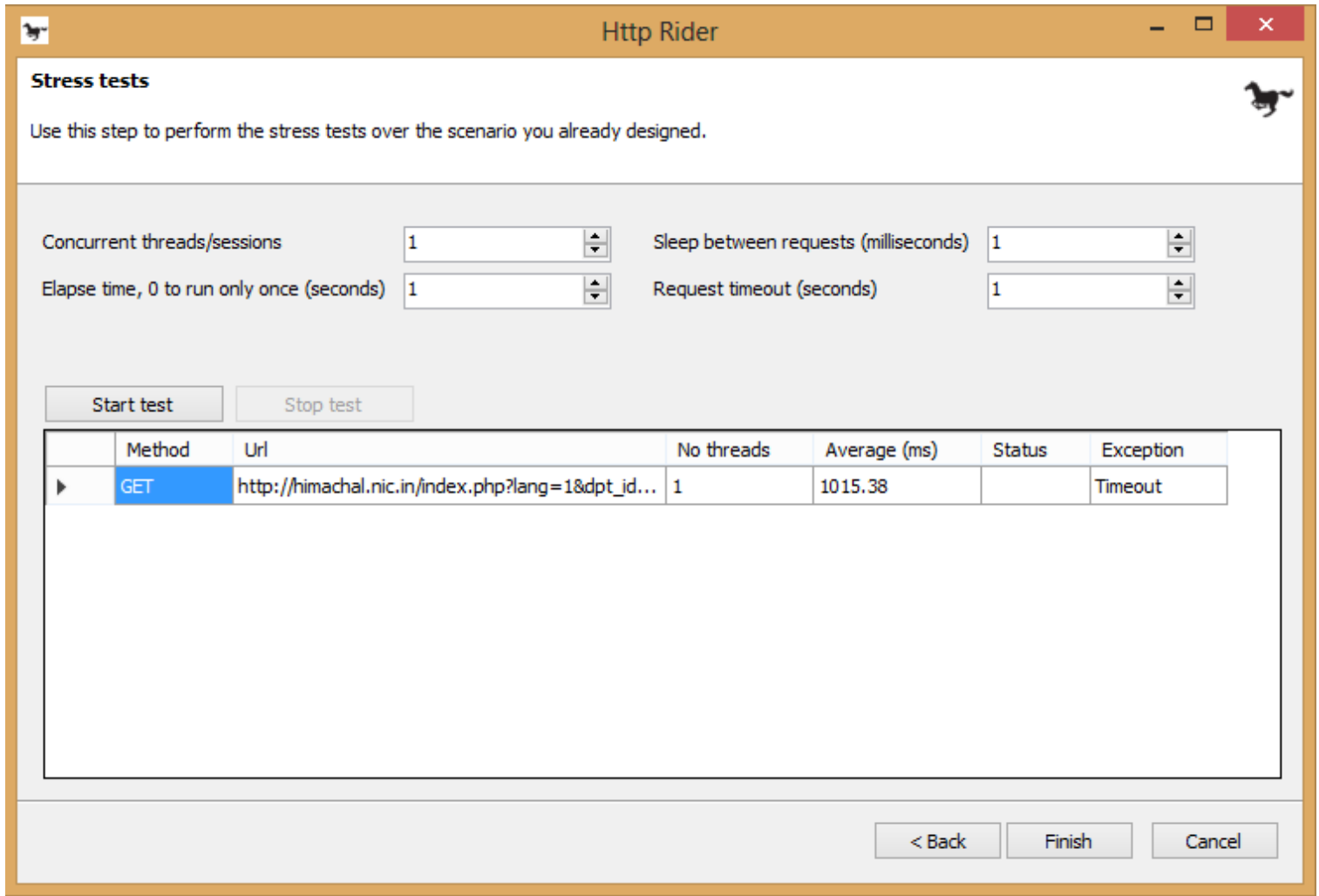


Figure 4:-Results without desired parameters.

Initially http rider tool whose results are shown in Figure 4 does not show the parameter throughput which is considered important for efficient web service testing tools. So the initiative has been taken to accommodate the mentioned parameter into web service testing tool. Respective formulas and coding have been incorporated in the initial source code of http rider tool to calculate the parameter throughput. After the modification in http rider tool and providing the same input i.e. the sample web service, same output and the additional parameter throughput was obtained as shown in Figure 5

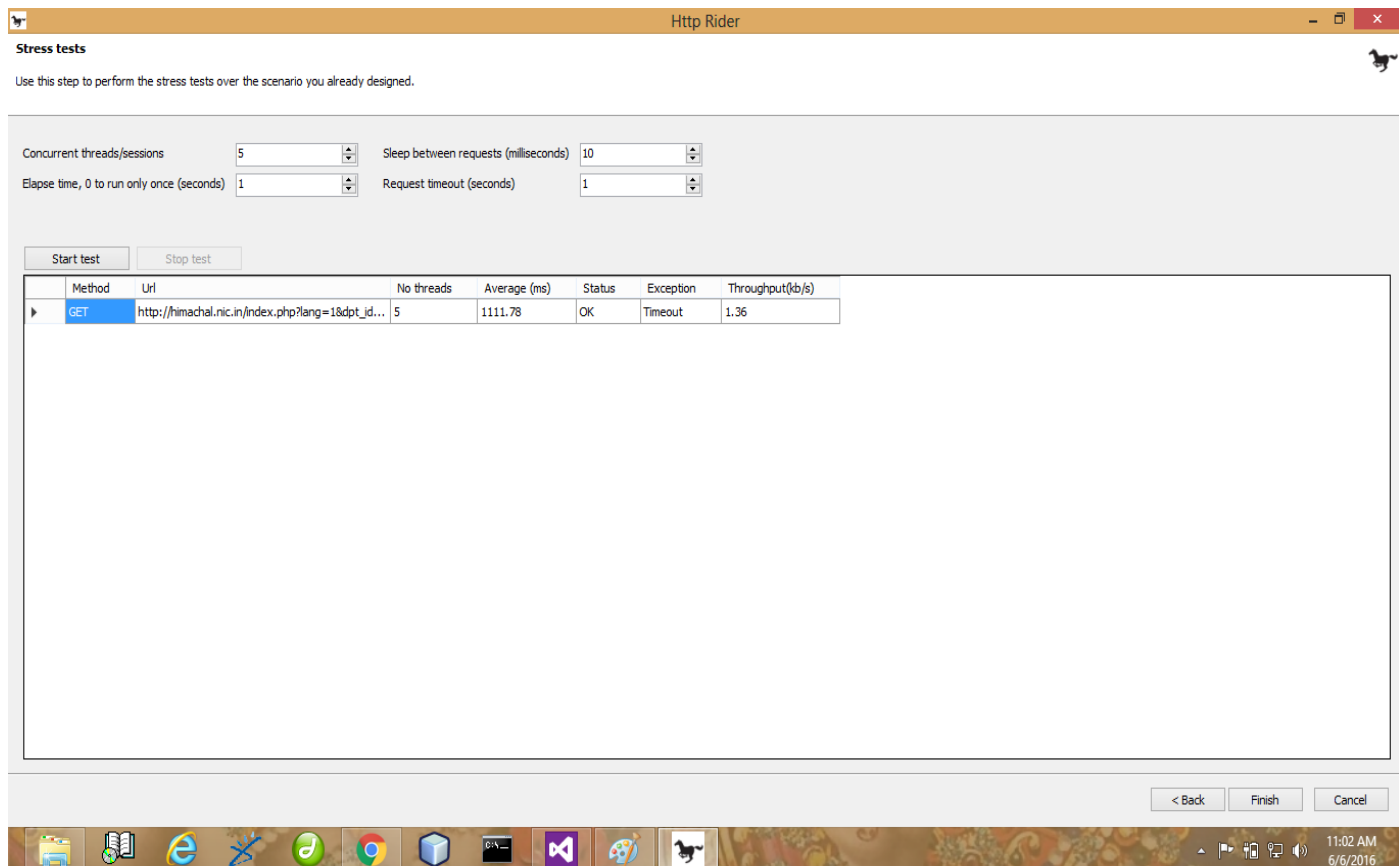


Figure 5:- result for enhanced parameter of web service testing tool

IV. Results and Discussion

The observed results of both the initial and modified version of the http rider has been shown in Table 1 and in Table 2 respectively.

Table 1:- Output without parameter (throughput) in http rider

S. no	Tool name	Web Service	Response time(ms)	Throughput (kb/s)
1	HttpRider	http://himachal.nic.in/index.php?lang=1&dpt_id=23	1015.38	-

Table 2:- Output after integrating parameter (throughput) in http rider

S. no	Tool name	Web Service	Response time(ms)	Throughput (kb/s)
1	HttpRider	http://himachal.nic.in/index.php?lang=1&dpt_id=23	1111.78	1.36

The sample web service i.e. http://himachal.nic.in/index.php?lang=1&dpt_id=23 was given as common input to both the initial and modified version of the http rider. After comparing the two tables the study shows that the initial version of http rider gives output without parameter throughput. Later when the tool was modified and the output was re-evaluated the result included parameter throughput. After parameter integration the tool gives output with response time 1111.78ms and throughput 1.36Kb/s.

The observed result were collected and compiled for the analysis and evaluation to determine the efficiency of the tool.

V. Conclusions and Future Work

Testing of web service is a challenging activity. Testing of a web service includes many characteristics such as runtime discovery, multi organization integration, throughput, calculation of bytes and much more. This paper evaluated the httprider for few of these parameters named as response time and throughput. Effort was made to integrate the calculation of these parameters into the tool and the result was re-evaluated. The difference was found in the prior results which can be seen in the Figure 5. Nevertheless there are still some limitations to current work that can be fruitfully improved and exploited for future work. The modified version of the tool calculated throughput for the single/multiple user which may limit its efficiency and reliability. So, the research work can be extended for the facilitation of throughput calculation for multiple web services testing tool and by comparing the different web services testing tool for the same input, results can be checked and validated. Also we can use more than one web services to compare web services testing tool.

References

- [1] Ahlawat Pooja and Tyagi Sanjay, "A Comparative Analysis of Load Testing Tools Using Optimal Response Time", International Journal of Advanced Research In Computer Science and Software Engineering (IJCSSE), May2013.
- [2] Tanuj Wala, Aman Kumar Sharma , "Improved Software Testing Tool", International Journal of Computer Science and Mobile Computing, Vol.3 Issue.9, September- 2014.
- [3] Dipika Kelkar, Kavita Kandalgaonkar, "Analysis and Comparison of Performance Testing Tools", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 5, May 2015.
- [4] Divya Bhargava, Aditya Veda, "A Different Techniques and Strategies for Software Testing", INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY [Bhargava , 2(12): December, 2013].
- [5] Myers Glenford J., "The art of software testing", New York, 2nd Edition 2004.
- [6] Khan Mohd. Ehmer, "Different Approaches to Black Box Testing Technique for Finding Errors", International Journal of Software Engineering and Applications (IJSEA), October 2011.
- [7] Prasad K.V.K.K.K, "Software Testing Tools", Dreamtech, 1st Edition 2006.
- [8]]Hussain Shariq, Wang Zhaoshun, Toure Ibrahima Kalil and Diop Abdoulaye, "Web Service Testing Tools: A Comparative Study", International Journal of Computer Science (IJCS), January 2013.
- [9] Larsson Andreas and Ekelund Marcus, "Web Services- A Comparative Study", Vaxjo University at School of Mathematics and Systems Engineering (MSI), November 2004.
- [10] Shikha Dhiman, Pratibha Sharma, "Performance Testing: A Comparative Study and Analysis of Web Service Testing Tools", International Journal of Computer Science and Mobile Computing, Vol.5 Issue.6, June- 2016
- [11] www.httprider.codeplex.com,"HTTTPRIDER", Retrieved on Jan 2016.
- [12] <http://www.asjava.com/tools/fiddler-tutorial-how-to-use-fiddler/> Retrieved on May, 2016