

Correlation for Dynamic Values Using WDiff Performance Testing Technique

MuniDhanalakshmi Kumbakonam

Research Scholar, Department of Computer Science Bharathiar University, Coimbatore, Tamilnadu, India.

Mahammad Shafi RajaSaheb

Professor, Department of MCA, Sree Vidyanikethan Engineering College, Tirupati, Andrapradesh, India.

Abstract— Correlation is the elementary formation and the primary test involved by the performance testers while scripting the business progression for load testing. Correlation is used to capture the dynamic values return by the server. Dynamic values such as Cookies, Checksums, and Session Identification's vary every moment we utilize an application. At time of Script Rerun, the traced dynamic values might not be equal the real dynamic values and our script not succeed. To overcome that failure, we should do the correlation for the dynamic values and replace the user session values. This concept called as Correlation. In this Paper for this correlation we are used Load Runner Performance Testing Tool Techniques. With the help of Virtual User Generator (VuGen) we will do the Automatic and Manual correlation. So that our script/transaction will get success. WDiff Utility is one of the Comparison Technique in Loadrunner.

Keywords— Automatic Correlation, Manual Correlation, WDiff, Dynamic values, VuGen, Session ID

I. INTRODUCTION

Correlation is one of the important mechanism in LR For Dynamic value or session value discovery purpose Correlation is Very useful. Client responsibility is to send the request accordingly server will give the response. Server will give response for any type of request. In this stipulation Correlation comes to the picture in LR. There are two types of Correlations in LR. Those are Manual Correlation and Automated Correlation. In this Manual correlation we should replace the dynamic value with parameter name in the Application script. These all are done by manually. Before replace we should recognize the dynamic value .Immediately confine that value from server response of earlier request.

Automated Correlation depends on various rules, which are associate with Tool. One more advantage or responsibility of Correlation is Identification of various records, which consist of data and also come back by Server. In this situation various users also will be come. Server will also returns the data for various users. The data consist of various records. Correlation responsibility is to identify the those data records and bring those two script actions with the help of various clients. Now we should compare these scripts with Tool[4].

Necessary of the correlation ? : Realize the Correlation with the subsequent picture. Do the script cassette with Load Runner. Here Client-Server mechanism shows in Fig 1.

- Throughout Recording, the consumers throw a appeal to the server to begin a process of session. The Server will accept that appeal and assign a fresh session Identification DEF. consumer provisions the session Identification DEF in the play. Consumer begins the session process with ID DEF and throw a appeal to a server to permit it to trace. The Server identify the Identification DEF and allocate the consumer to trace the play[1].

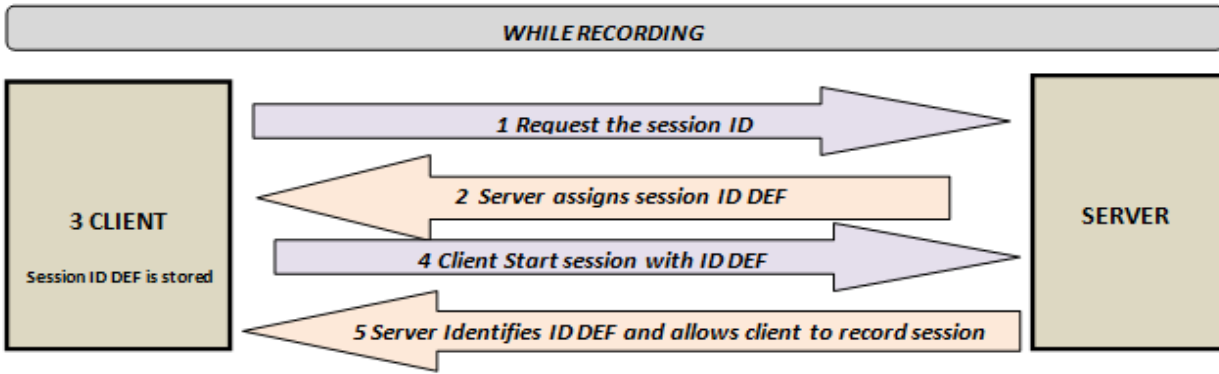


Fig 1: While Recording the Process

Currently, let us look at the client-server mechanism throughout rerun/rerecord. Discussed in Fig 2.

- Throughout Rerun, a consumer throw a appeal to server to begin a meeting. A Server accepts the appeal ,accordingly assign a fresh process ID ZYX. A Consumer begin the process session with earlier stored session Identification DEF and throw appeal to server to permit it to trace. A Server is not capable to recognize the Identification DEF , because expect ID ZYX that it have owed and sessionprocess not pass[12].

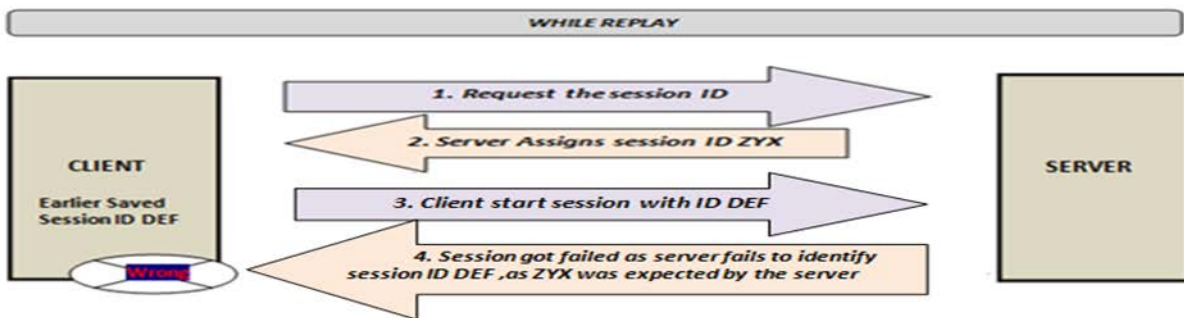


Fig 2: While Replay the Script

We require a result, which to confine parse the changing value sent at run-time and go back this value to server. Here is how the client-server mechanism will look like with this result , discussed in Fig 3.

- Throughout Rerun, a customer throw a appeal to the server to begin a sessionprocess. A Server accept the appeal, and assign a fresh process ID YYY. A Client throws this new session Identification YYY from the requeststate. Consumer propel a appeal to begin session process with id YYY. Server identify the Identification YYY and permit consumer to rerun/rerun the actions[12].

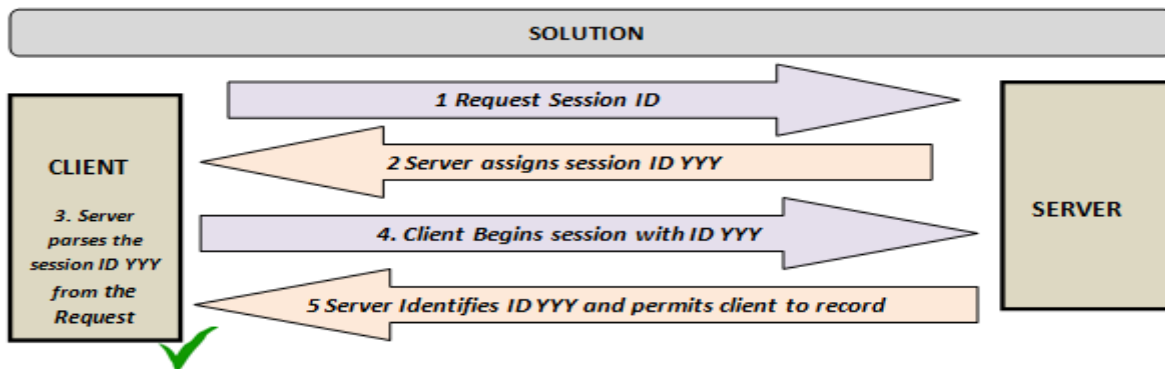


Fig 3: Solution with Correlation

This is nothing but correlation.

Correlation happen in to paths: Manual mechanism is Manual correlation, Automatic process is Automatic correlation.

Manual correlation can be done as: Manual Regulations, Snapshot Evaluation, Virtual user Evaluation (suggested)[12].

II. LITERATURE REVIEW

Confine the dynamic value at server response is recognized as correlation mechanism. Correlation is wherever the actions/process is tailored so that some of the fix standards in the script process are no longer fixed. Instead of transfer the novel assessment to the server, we can propel diverse standards. For Instance, the novel stored actions might be incorporated the server transfer the consumer a session Identification, a little to recognize the consumer throughout that exacting session. This assembly Identification was fixed addicted to the script throughout playing. throughout rerun, the server shall sends the consumer a fresh process Identification. If we confine this assessment, integrate it addicted to the actions so that we can throw it reverse to the server to suitably identify our self for this fresh process. If we run off the process unchanged, we shall send the older fixed Session Identification to the server[2]. The server should observe it and believe it worthless or unidentified, and so should not propel the sheets that has been demanded. Rerunning a action with previous process Identification might not be profitably trap the server into deem it is a client. Correlation is the confine of dynamic values approved from the server to the consumer and reverse. we keep these imprison value into a constraint, called parameter and after that utilize this parameterization assessment in the play in lay of the novel value. throughout rerun, the rerun steam shall take note to what the server drive to it, and when it create requirements of the server, propel this novel, suitable value reverse to the server; thus fooling the server into consider it is talking to a real client[1].

The primary reason for correlating statement is

- To make simpler or optimize our code
- For Dynamic Data[3]
- To accommodate unique records.

You can also enhance the scripts by

- Insert transactions into a Virtual user script.
- Insert Rendezvous point into a Virtual user script
- Inserting comments into the script.
- Sending messages output.
- Handling errors in Vuser Scripts[11].

What errors mean I have to correlate?, According to this question we have following survey. That is, at hand no exact mistakes that are linked by correlation, other than here are mistakes that might happen as a value has not be correlated. For instance, if a improper process Identification is throw to a Web server, how that server reacts based on the accomplishment of that server. It might send a page exclusively stating the Session Identification is invalid and ask you to log in another time. It may perhaps sends an HTTP 404 Page, indicates not found error as the demanding consumer did not have authentications intended for the particular sheets. consequently the server might not place the sheet. In universal, a few mistake communication go back from the server after the action makes a appeal that protests about permissions can tip to a fixed value that required to be correlated [3].

WDiff default Utility in Load runner to Compare the Script. To use WDiff effectively, you record the identical operation twice, and compare the scripts (or data files for Tuxedo, WinSock, and Jolt). WDiff shows differences in yellow. Reminder that not all differences designate a

value to correlate. For instance, certain receive buffers that indicate the time of carrying out do not involve correlation. Only Server Responses values need to Correlate. Path is Tools->Compare in Load Runner [12].

III METHODOLOGY

Correlation Research Methodology:

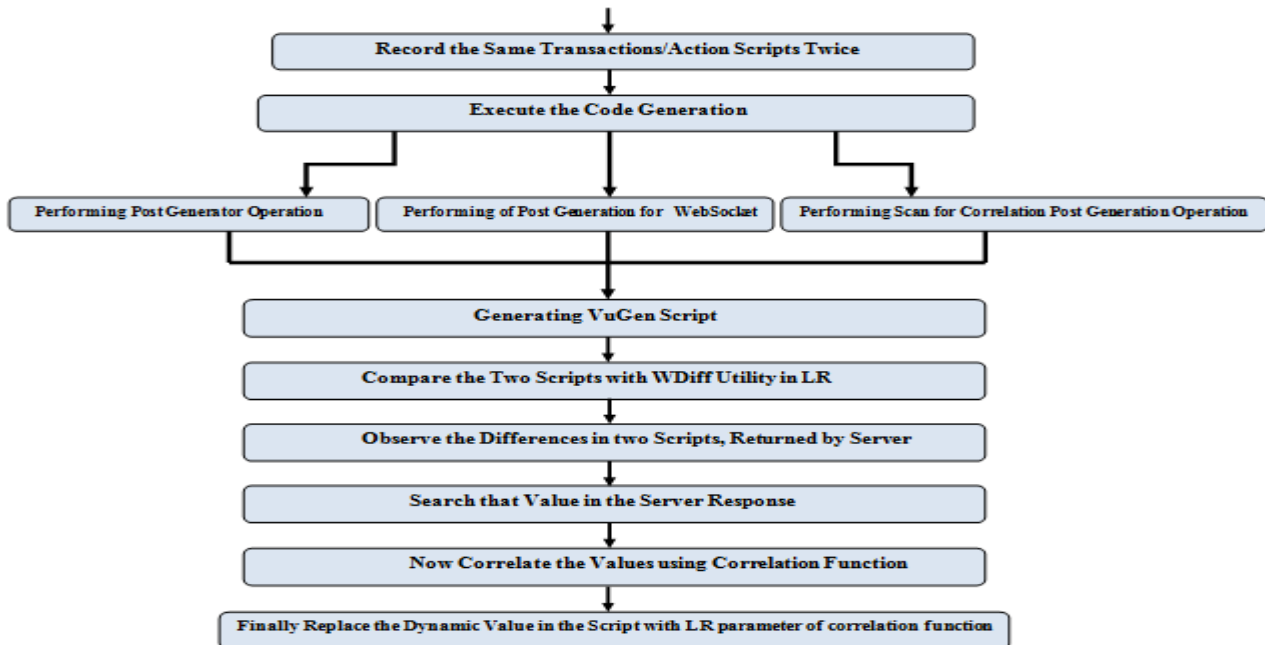


Fig 4: Research Methodology of Correlation

Above Diagrams Indicates the flow of Correlation in Load Runner. According to this flow will get the correlation for dynamic values[7]

IV MANUAL CORRELATION PROCESS

1. **Identify the Dynamic values to be corrected:** record the same script twice and compare both the recording to identify the values.

Script 1 (Left)	Script 2 (Right)
<pre> web_url("WebTours", "URL=http://127.0.0.1:1000/WebTours/", "TargetFrame=", "Resource=0", "RecContentType=text/html", "Referer=", "Snapshot-t1.inf", "Mode=HTML", LAST); lr_think_time(6); web_submit_data("login.pl", "Action=http://127.0.0.1:1000/WebTours/login.pl", "Method=POST", "TargetFrame=", "RecContentType=text/html", "Referer=http://127.0.0.1:1000/WebTours/nav.pl?in=home", "Snapshot-t2.inf", "Mode=HTML", IFERROR("Name=sessionId", "Value=109968.0(CorrelationParameter 1)", "Name=username", "Value=NaveenKumar", ENDIF, "Name=password", "Value=naveen", ENDIF, "Name=JSFormSubmit", "Value=on", ENDIF, "Name=login.x", "Value=5", ENDIF, "Name=login.y", "Value=9", ENDIF, LAST); </pre>	<pre> web_url("WebTours", "URL=http://127.0.0.1:1000/WebTours/", "TargetFrame=", "Resource=0", "RecContentType=text/html", "Referer=", "Snapshot-t1.inf", "Mode=HTML", LAST); lr_think_time(6); web_submit_data("login.pl", "Action=http://127.0.0.1:1000/WebTours/login.pl", "Method=POST", "TargetFrame=", "RecContentType=text/html", "Referer=http://127.0.0.1:1000/WebTours/nav.pl?in=home", "Snapshot-t2.inf", "Mode=HTML", IFERROR("Name=sessionId", "Value=109968.1813689751zrztlipbcQ0zzd", "Name=username", "Value=NaveenKumar", ENDIF, "Name=password", "Value=naveen", ENDIF, "Name=JSFormSubmit", "Value=on", ENDIF, "Name=login.x", "Value=6", ENDIF, "Name=login.y", "Value=2", ENDIF, LAST); lr_think_time(5); web_url("SignOFF Button", "URL=http://127.0.0.1:1000/WebTours/welcome.pl?signOff=1", "TargetFrame=body", "Resource=0", "RecContentType=text/html", "Referer=http://127.0.0.1:1000/WebTours/nav.pl?page=menu&i", "Snapshot-t3.inf", "Mode=HTML", LAST); </pre>

Fig 5: Comparison of Script with WDiff Utility and Identify the Dynamic values

Above Fig 1 Shows the Script for Web Tour Application. In this we can record any number of Transaction as per our Application. Above Script Consist of Launch, Login, Transaction Process, Logout. So record these steps into a particular action file called action 1 .Now same script with same transaction we try to the rerecord it. But this time we try to store it in another action called action2[5]. Now we compare these two actions Action1 and Action2 with the help of VuGen Compare tool. It will give us the recommendations as how these two scripts are different. It will be marked as yellow as how Action1 is different from Action2. So little bit of analysis we will be able to figure out how to identify which one used to correlate. Few Server response values are will get different in this Comparison. This different values we have to identify and that values to be correlated[4].

2. Search that value in the server response : once, the value is identified, check for that value in the server response to identify left boundary and right boundary.

```

web_url("welcome.pl",
  "URL=http://127.0.0.1:1080/cgi-bin/welcome.pl?signOff=true",
  "Resource=0",
  "RecContentType=text/html",
  "Referer=http://127.0.0.1:1080/WebTours/index.htm",
  "Snapshot=t58.inf",
  "Mode=HTML",
  LAST);

/* *****login***** */

lr_save_string(lr_decrypt("594a705219e984de"), "PasswordParameter");

lr_think_time(16);

web_submit_data("login.pl",
  "Action=http://127.0.0.1:1080/cgi-bin/login.pl",
  "Method=POST",
  "RecContentType=text/html",
  "Referer=http://127.0.0.1:1080/cgi-bin/nav.pl?in=home",
  "Snapshot=t59.inf",
  "Mode=HTML",
  ITEMDATA,
  "Name=userSession", "Value=121348.773349534zctVi0HpfCQVzzzH0HtfVpckfVHf", ENDITEM,
  "Name=username", "Value=jojo", ENDITEM,
  "Name=password", "Value={PasswordParameter}", ENDITEM,
  "Name=login.x", "Value=19", ENDITEM,
  "Name=login.y", "Value=6", ENDITEM,
  "Name=login", "Value=Login", ENDITEM,
  "Name=JSFormSubmit", "Value=off", ENDITEM,
  LAST);
  
```



Fig 6: Identify the Correlated Value and Search that value in the Server Response

Search for this value in the server response of the previous request let's say there are two requests one is request 1 and second is request 2 and you have identify the values to be correlated on the request 2. now we have to go to the server response of request one and search for this value in the server response of request one once you find a value in the server response of request one what we have to do is we have to pick up the left boundary and the right boundary from there and once we have picked up the left boundary and right boundary then we have to use a correlation function, web_reg_save_param_ex, using LB and RB to correlate it and finally once the correlation is done we have to change the value, the hard coded value which is there in request 2 we have to change it with the parameter name. Once you have done this our manual correlation is completed[6].

3. Correlate: Correlate the value using Web_Reg_Save_param_ex using Left Boundary (LB) and Right Boundary (RB)

Go to the response of this request (i.e., welcome.pl). For this click on previous request and click on snapshot pane.

Snapshot pane will be displayed. In the snapshot pane for this request (previous request pane) there are so many HTTP requests are available accordingly we search the our session could be there in any of this HTTP requests. Now Start searching in the response of the first HTTP request click on the first http request and see below raw data tab, which contains request & response data information. If we click on second Http request we will find their request & response. do this

procedure until our user session value could be there in the response of any one of the HTTP Request[8].

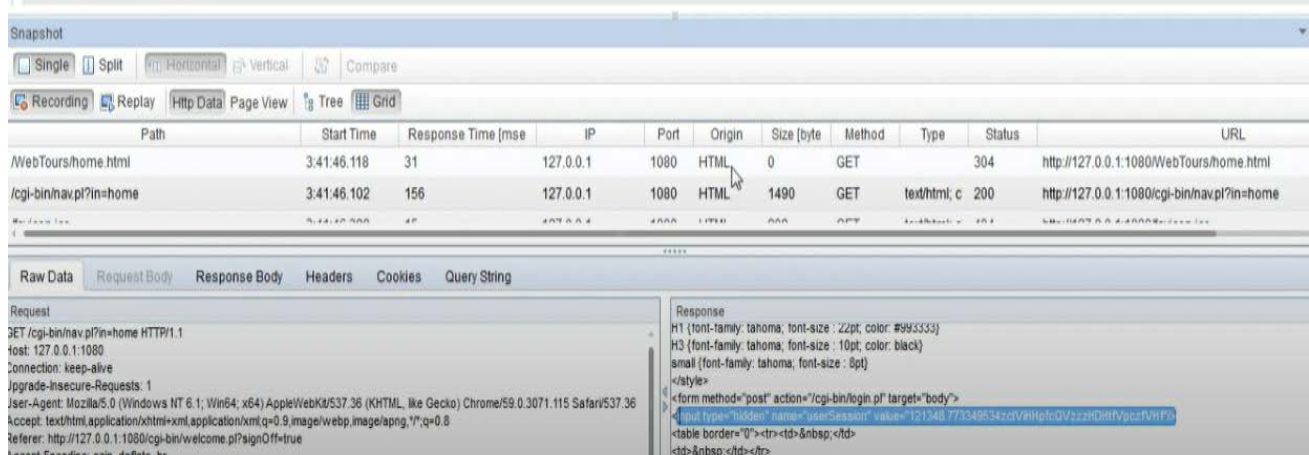


Fig 7: Searching Response of HTTP Request

In the response the text which is left to it is the left boundary and the text which is right to it is the right boundary. So copy the whole text in the response area .So this is the way we found it in the response of this request.

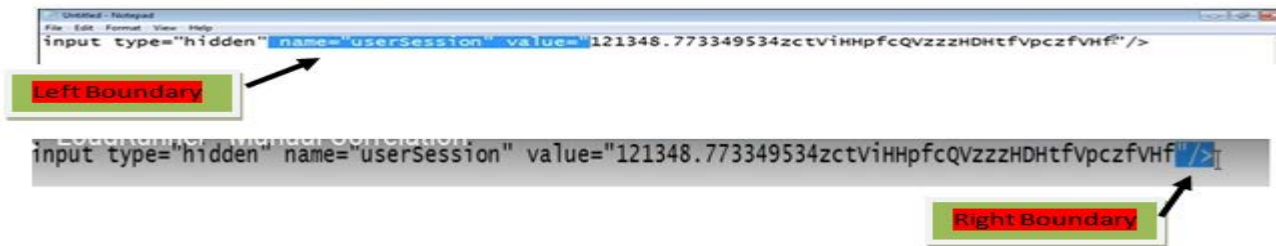


Fig 8: Identify the Left Boundary and Right Boundary

Above Figure Shows the Left Boundary and Right Boundary. After getting the server response of previous request in the snap shot area.we should indentify the Boundary values[7] .

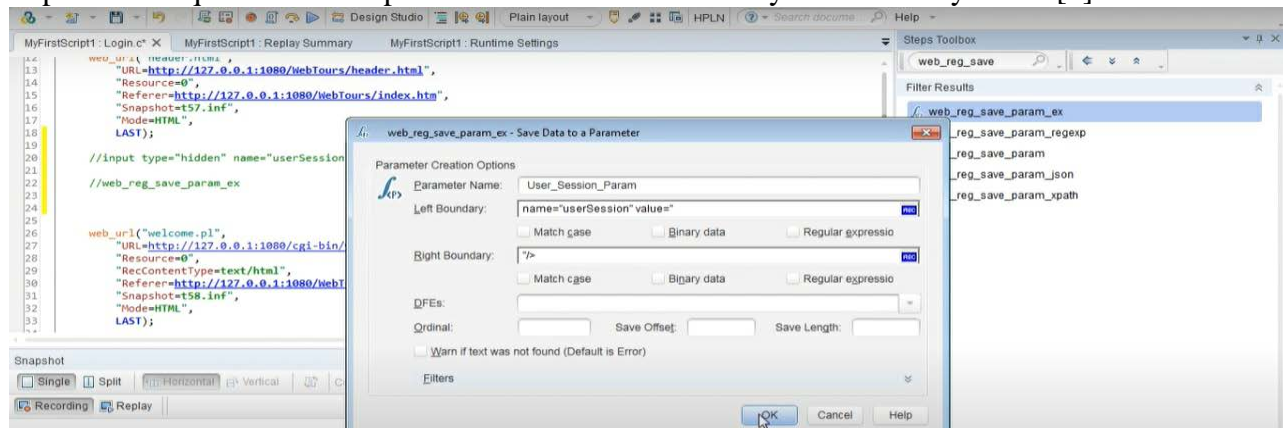


Fig 9: Apply the Correlation Function

Above Figure Indicates manual correlation functioning in LR.For this we have specified function.i.e Web_reg_save_param_ex. Process of Function: Insert->web_reg_save_param_ex (using steps toolbox)->double Click->prompt will come-> fill parameter name, left boundary and right boundary data -> click ok .Once click ok . Function script will be built on top of welcome.pl (which is response of previous request) [9]

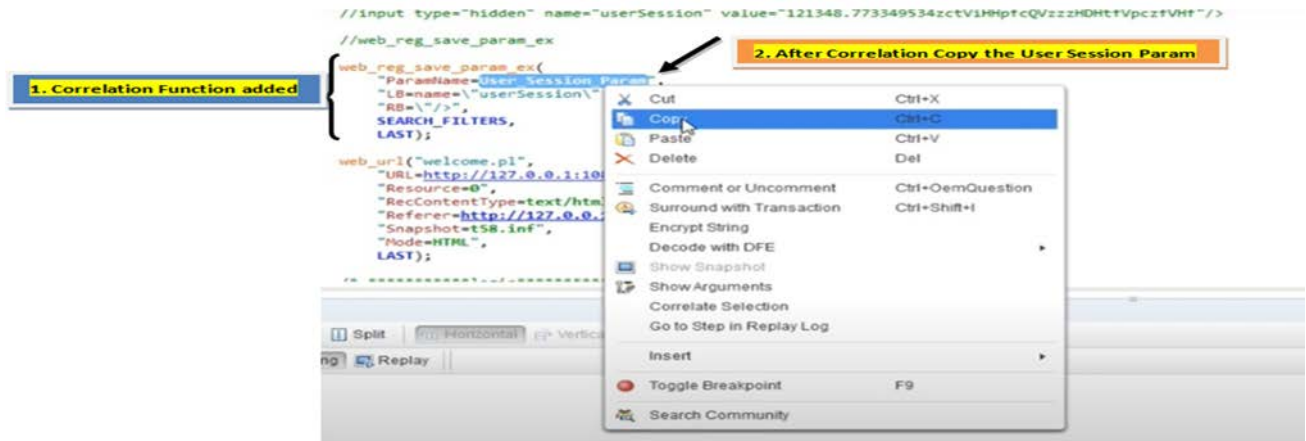


Fig 10: Correlation Function

4. Replace: Replace the dynamic value present in the script with the LR parameter of correlation function.



Fig 11: Replace the dynamic value present in the script with the LR parameter of correlation function

Once Correlated Function is added in the Server response of previous request then copy that User Session Param and replace the dynamic value present in the script. Now Successfully Manual Correlation will be completed[6].

V AUTO CORRELATION

It Consist of 1. Play again the action script. 2. Recognize the changed values to correlate. 3. choose the ideals and tick the auto-correlate key. 4. Prove the script. Classification Of Automatic Correlation Contains, 1. Automatic detection Correlation: perceive and compare active data of function servers carry by HP LoadRunner. 2. Regulation Based Correlation: It is worn as operational with a unsupported application server for which the framework is known. 3. Correlating the whole statements: help out for correlate the complete changed data[9].

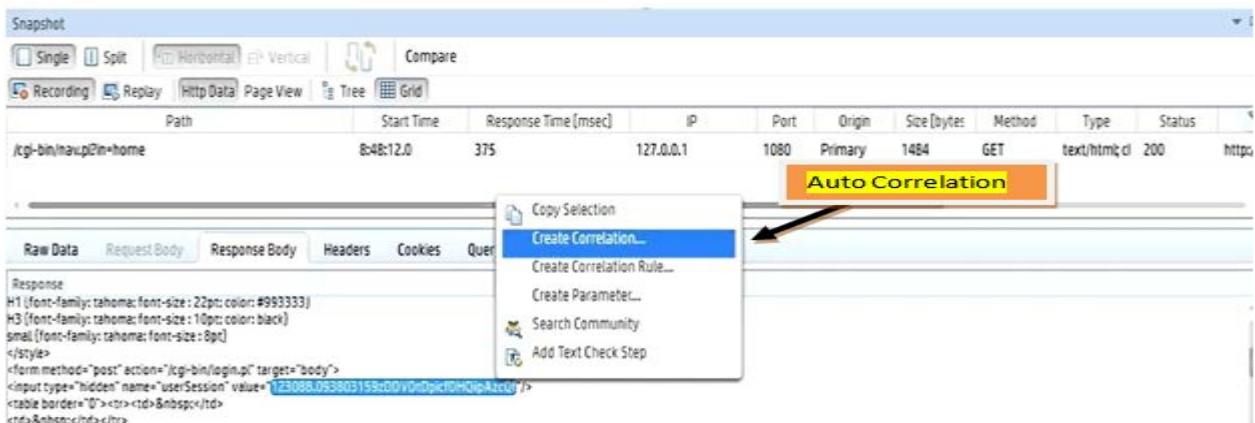


Fig 12: Automatic Correlation Using 'Create Correlation' Method

Once find the server response of previous request in snap shot pane select the User session value and right click on that value, accordingly select the ‘Create Correlation option’ ,then click on that option, automatically Auto correlation script will be generated[9].



```
web_reg_save_param_ex(
    "ParamName=userSession",
    "LB=name=\"userSession\" value=\"",
    "RB=\"",
    SEARCH_FILTERS,
    "Scorp-Badu"
```

Fig 13: Auto Correlation Script using Create Correlation option in snap shot pane

VI CONCLUSION AND FUTURE WORK

In this Research work, applications are intended in regulate to have accurate statistics from every customer, so simulating consumers needs to be done in the same way as the application expects which means that each user desires an individual ID. In LoadRunner, correlation will provide high-quality solution for these issues [10].

VII REFERENCES

- [1] Bhatti Sandeep, Raj Kumari, “Comparative Study of Load Testing Tool”, International Journal of Innovative Research in Computer and Communication Engineering (IJRCCE), Volume 3, Issue 3, March 2015.
- [2] Sharma Jagdish, Monika Sharma, “A Comprehensive review of web-based Automation Testing Tools”, International Journal of Innovative Research in Computer and Communication Engineering (IJRCCE), Volume 3, issue 10, October 2015.
- [3] Angmo Rigzin, Monika Sharma, " Performance Evaluation of Web Based Automation Testing Tools" IEEE 2014.
- [4] Bhatia Reenu, Anita Ganpati, “In Depth Analysis of Web Performance Testing Tools”, IRACST Engineering Science and Technology: An International Journal (ESTIJ), Vol.6, no. 5, Sept-Oct 2016.
- [5] Ms. Sharmila S., E Ramadevi., “Analysis of performance testing on web applications”, International Journal of Advanced Research in Computer and Communication Engineering, Vol 3, March 2014.
- [6] Vikas Nandal and Dr. Kamna Solanki “LITERATURE REVIEW OF PERFORMANCE TESTING ON WEB-BASED APPLICATION USING LOADRUNNER”, JETIR (ISSN-2349-5162), February 2018, Volume 5, Issue 2.
- [7] Shalini, Jawahar Thakur, “Web Performance Testing Tools”, 2017.
- [8] Ramadev et. al. “Analysis of Performance Testing on Web Applications” 2014
- [9] Harsh Verma, Arun Krishnan and Sivasubramanian, “Analytics Performance Load Test”, International Journal of Computer Applications, Vol. 176, No. 2, 2017
- [10] Neha Bhateja , “A Study on Various Software Automation Testing Tools” ,(IJACSA), Volume 5, Issue 6, June 2015 .

- [11] Zhi Quan Zhou, Shaowen Xiang, and Tsong Yueh Chen “Metamorphic Testing for Software Quality Assessment: A Study of Search Engines”, 2015 IEEE
[12] Santoso Wibowo and Srimannarayana Grandhi “A Performance-Based Approach for Assessing the Quality of Group Buying Websites”, 2014 IEEE.



MuniDhanalakshmi K, Pursuing PhD, from BharathiarUniversity, Coimbatore, India, received the MTech degree from Dr. MGR University, Chennai, India and MCA. degree from S.V University, Tirupathi, India. She is currently an Assistant professor with the Department of Computer Science and Engineering at Bharat Institute of Engineering and Technology, Hyderabad, India. Her research interests include Software Quality Assurance, Search engine and machine learning. She has co- authored a number of research articles in various journals, conferences.



Dr. Mahammad Shafi RajaSaheb is currently working as Professor with the Department of Master of Computer Applications at Sree Vidyanikethan Engineering College, Tirupati, India. He received PhD degree in the year 2010 and M.Tech degree in the year 2005. He has published number of research articles in various journals and conferences. He has extensive skill in Software Quality Assurance and Testing, Cloud Computing and Big Data Analytics.