Shruti Droupadkar, et. al., International Journal of Engineering Research and Applications
www.ijera.com
ISSN: 2248-9622, Vol. 15, Issue 3, March 2025, pp 196-199

RESEARCH ARTICLE OPEN ACCESS

Load testing of static content & live streaming by using software automation

Damodara Rao Tammineni, Shivashenkreppa A Deginal, Bommalapati Malleswari, Mohan Kumar S, Bhagyalaxmi, T S Swamy
Centre for Development of Telematics, Electronic City, Phase-1
Hosur Road, Bengaluru India, 560100,
Email: damodhart@cdot.in, shivu@cdot.in, malli@cdot.in, mohans@cdot.in, bhagya@cdot.in, tss@cdot.in

Abstract
In this paper, we proposed Automation of Load testing on static (audio/video/text) and live content (TV/FM Radio channels). Our Automation aims to include a wide range of metrics such as user creation, number of users supported and measures the performance of Converged Content Player (CCP) in peak load conditions. Automation of load testing also provides active users concurrently and measures performance bottlenecks. It is important to identify and address the problems associated like buffering issues with respect to the Converged Content Player using the load testing automation.

Keywords:
CCP: Converged Content Player
ADB: Android Debug Bridge
%CPU: Represents the CPU usage.
%MEM: Shows the Memory usage of task.
HD channel: High Definition channel
SD channel: Standard Definition channel

Date of Submission: 15-03-2025 Date of acceptance: 31-03-2025

I. INTRODUCTION

In the Converged Content Player, Digital Terrestrial/Satellite/Cable content is received, decoded, and selectively streamed over the WLAN/Wi-Fi. CCP contains Local Content Server and Web Server. Local Content Server can be hosted for various offline services like static content (Video, Audio & books and many more). The content is consumed by end-users through accessing Web Server using browsers on smartphones, tablets and laptops by the WLAN/Wi-Fi. The user shall be able to play the content directly in the browser by selecting. The CCP system is capable of supporting Free Channels/FTA TV program channels, the capabilities to support 200 to 1000 active connections simultaneously. To get the peak performance of the CCP system we need to put load on web server and local content server by accessing the live or static content by different number of users, so that we can check performance issues [9] as well as threshold of the system. An important part of running load tests is to know the web server capacity for live streaming while users watching the video and audio simultaneously.

Traditionally, load testing was manual, posing challenges in producing repeatable results and measurable stress levels. Manual failure diagnosis can be very time consuming and error-prone [7].

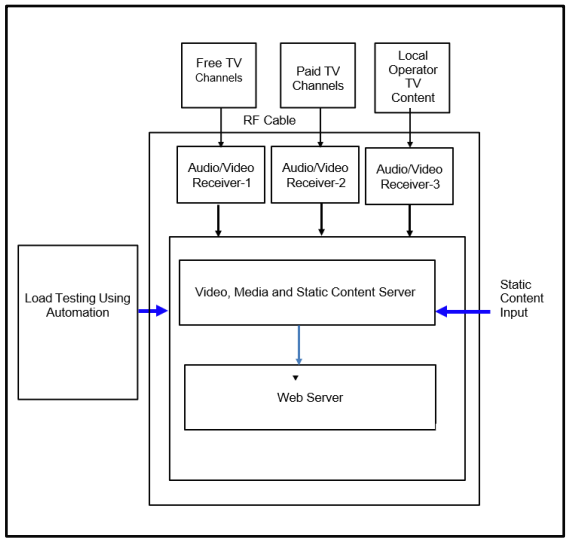


Figure 1: CCP Test setup

II. AUTOMATION OF LOAD TESTING

We used ROBOT Framework, ADB to simulate the users by invoking different types of browsers and making them as interactive as possible for the different number of users.

PRE-REQUISITES:

1. Software: Robot framework, ADB
2. Browsers: Any browser (Chrome, Firefox, etc.)
3. OS: Linux, Windows

LOAD SIMULATION:

As every portable device have browsers and no need of installing apps for load testing [2]. To create the load on webserver we require the unique users with credentials. Users with valid credentials only can access the content of CCP. So we created users (with unique names) using automation script. Steps involved in user creation is open browser enter the web server address, clicks on signup button, then enter the username, password and clicks on create user button to create the User. This process is repeated until given numbers of users are created. Now we start our load test with created users on webserver. Steps involved in Load test, Open any browser, enter the URL of web server, login with user credentials, and Select the any content to watch, at this point of time we can interact with browser manually. Repeat the same procedure for different users. While doing this process within the pre-defined time, we are monitoring the CCP system performance as mentioned earlier we can interact with the browsers manually for each user. After pre-defined time we are stopping the automation process. All the parameters (Pre-defined time, content type (live or static), channel number, system monitoring time) can be modified to re-run the script. This test analyses adding 1 user takes 5 seconds until reaching 100 users. The entire stepping process takes 500 seconds. After reaching 1,000 users, all of them will continue running and hitting the server together for 8.3 minutes. By using this script, we can generate 500 users simultaneously using 5 systems in 8.3 minutes.

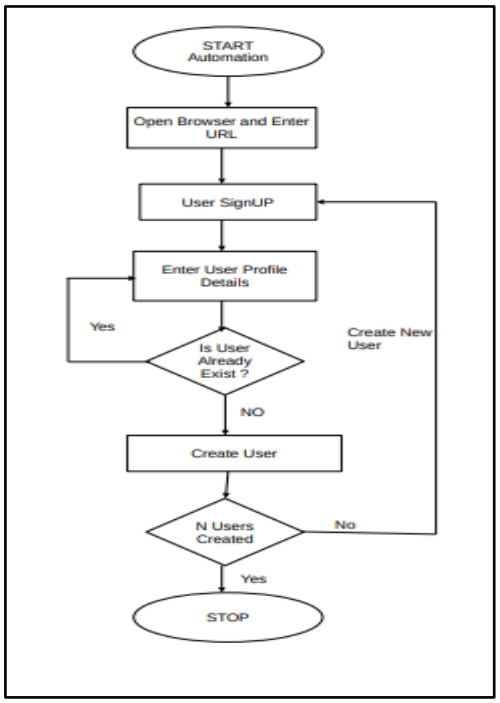


Figure 2: User creation flow diagram

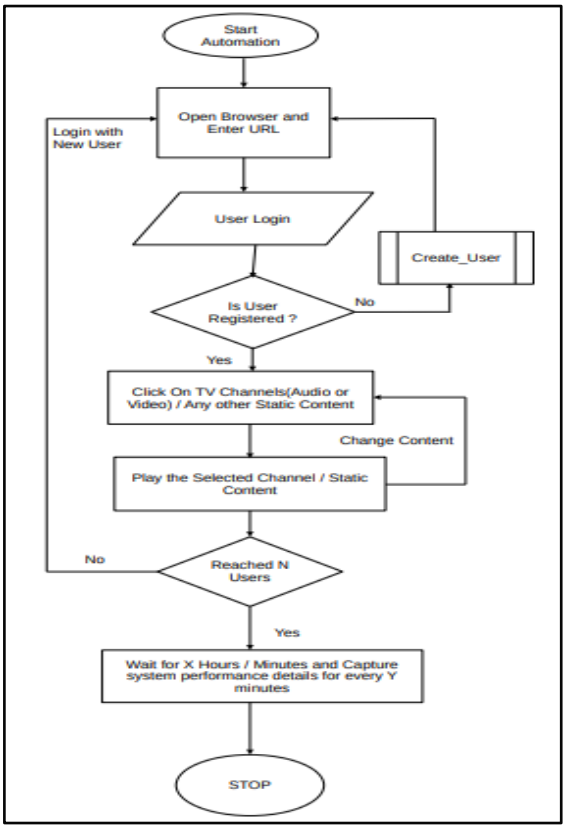
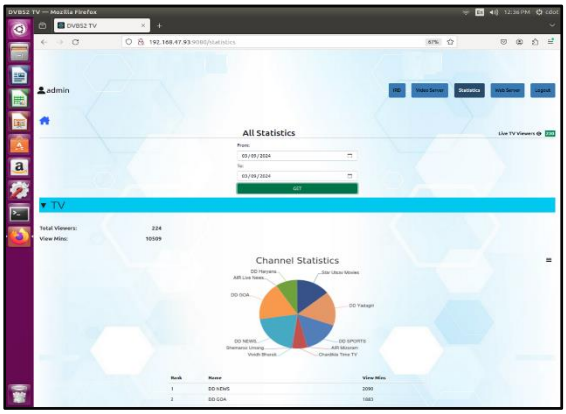


Figure 3: Automation flow diagram

III. ANALYSIS OF THE RESULTS

Our automation of load testing [9], [10] will give the number of users supported, system performance [1] issues (buffering, Contents played will be blurred, No Signal), response time, down time, automatically comes up or reboot required. We can also see across browser compatibility (Browsers used [2]). We can use the Viewership Statistics provided by Web Server of CCP system which will display of TV Live Viewer count (number of users).



OPTIMIZE AND RETEST:

We are able to show that by monitoring a customer-affecting metric and frequently comparing its degradation to the performance objective by increasing the no. of users. With this solution, we are ensuring the CCP system is stable to provide the services to the no. of users/customers satisfactorily and enables to implement the load testing at very low cost [3]. The paper outlines the concept of regression benchmarking as a variant of regression testing focused at detecting performance regressions [4],[5]. Fix the performance issues and retest the application until it meets the performance requirements [6]. For viewing HD channels system takes more memory compare to SD channels.

PERFORMANCE MONITORING TABLE

S. No	*No. of Users*	*No. of systems used to simulate users*	*%CPU*	*%MEM*
1	0	0	49.3	40.21
2	10	Laptop-1	58.9	40.33
3	54	Laptop-2 Desktop-1	58.9	40.35
4	107	Laptop-2 Desktop-2	60.5	40.46
5	203	Laptop-1 Desktop-5	65.9	40.47
6	251	Laptop-1 Desktop-6	66.7	43.65

IV. KEY FEATURES OF AUTOMATION OF LOAD TESTING

1. Creates number of users (N)
2. Takes less time for user creation (Approximately takes half an hour for 500 users).
3. Load run time can be X hours: X can be any hours.
4. Capture system performance details for every Y minutes.
5. Eliminates Performance Issues: Identifies bottlenecks [8] for the number of users supported, accessing the web server simultaneously using web browser ensuring robustness.
6. Scalability: Helps to prepare for peak loads by testing varying number of users specific to its capacity and number of systems (Laptop/desktop) required.
7. Common script for Satellite, cable & Terrestrial broadcasting applications
8. Dynamic change of channels (By Name or Channel ID).
9. Dynamic change of number of users.
10. It works in Automatic and manual interactive mode as we are using web browsers.

V. CONCLUSIONS

Load testing is a crucial process that requires preparation and design in order to ensure success. It involves a series of steps, including planning, creating scripts, scaling tests, and analysing results. Proper load testing can prevent downtime, improve user experience, and ultimately lead to greater user satisfaction. We can identify potential performance bottlenecks. It can provide with the actual picture of how systems handle under high load and browser compatibility.

REFERENCES

[1]. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5306331: Zhen Ming Jiang, Ahmed E. Hassan, Gilbert Hamann and Parminder Flora: Automated Performance Analysis of Load Tests

[2]. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8821989: Shahnaz M. Shariff∗, Heng Li∗, Cor-Paul Bezemer†, Ahmed E. Hassan∗, Thanh H.D. Nguyen‡, Parminder Flora‡: Improving the Testing Efficiency of Selenium-based Load Tests

[3]. A. Avritzer, A. Bondi, and E. J. Weyuker. Ensuring stable performance for systems that degrade. In the 5th international workshop on Software and performance, 2005.

[4]. L. Bulej, T. Kalibera, and P. T˚uma. Repeated results analysis for middleware regression

benchmarking. Perform. Eval., 60(1-4):345–358, 2005.

[5]. L. Cherkasova, K. Ozonat, N. Mi, J. Symons, and E. Smirni. Anomaly? application change? or workload change? towards automated detection of application performance anomaly and change. In IEEE International Conference on Dependable Systems and Networks, 2008.

[6]. Z. M. Jiang, A. E. Hassan, G. Hamann, and P. Flora. Automatic identification of load testing problems. In In Proceedings of the 24th IEEE International Conference on Software Maintenance (ICSM), 2008.

[7]. N. Mi, L. Cherkasova, K. M. Ozonat, J. Symons, and E. Smirni. Analysis of application performance and its change via representative application signatures. In Network Operations and Management Symposium, 2008.

[8]. E. J. Weyuker and F. I. Vokolos. Experience with performance testing of software systems: Issues, an approach, and case study. IEEE Trans. Softw. Eng., 26(12), 2000.

[9]. Neha Thakur, Dr. K.L. Bansal. Load Testing on Web Application using Automated Testing Tool: Load Complete.IJIRCCE: Vol. 3, Issue 10, October 2015

[10]. Xingen Wang, Bo Zhou, Wei Li, International Symposium on Parallel and Distributed Processing with Applications.