International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-7, Issue-3, May-2019
DOI: 10.21276/ijircst.2019.7.3.9

CMRepo End-to-End Automation

Shravya G, Prof. Smitha G R

***Abstract*— Performance Automation is a technique which helps in determining the performance of the system, by determining various system parameters under different workloads. This project aims at developing the framework for determining the performance. It mainly captures system level CPU and memory consumption during any of the operations being performed. Whenever any operation is done in CMRepo, the communication takes place between client and CMRepo server. This server hence communicates with relay, and relay to proxy. Therefore, this framework also helps in capturing the timestamps for each of the messages that are communicated between client, server, relay and proxy. CPU and Memory consumption, along with total time taken to complete the whole operation help in determining the performance of CMRepo. This project is done using Python scripting and Unix commands. This paper describes the designed framework that helps in determining the overall performance of CMRepo. If the determined performance is not matching with the threshold values, it means that the operation is not successful or if there's any issue or bug within the product. The data is captured at the idle condition of the system as well as when the operation is performed. This helps in giving clear picture of how the system is performing at various conditions.**

***Index Terms:* IMS, CMRepo, CSCF, CPU, HSS, IMS, Memory Performance**

I. INTRODUCTION

IMS is also known as IP Multimedia Core Network Subsystem. IMS is described as the network framework that is used for providing IP multimedia services. This is known for providing packet-switched network. The IP based services are Voice over IP (VoIP). IMS was introduced as Multimedia Domain (MMD) in the fifth release under Third Generation Partnership Project (3GPP). Later, it evolved in later releases as a huge network for IP based services and became IP Multimedia Services [1].

Manuscript received May 17, 2019

Shravya G, M.Tech., Software Engineering, RV College of Engineering®, Bengaluru -59. (email: shravyag.sse17@rvce.edu.in)

Prof. Smitha G R, Assistant Professor, RV College of Engineering®, Bengaluru-59,

The protocol used by IMS was IETF Protocol, example Session Initiation Protocol (SIP). As per 3GPP, IMS is used for providing access to multimedia and voice applications with either wireline or wireless terminals [2]. This is possible because of isolation between access network and service layer with the help of horizontal control layer. Overview of IMS are: Core-control architectural framework for multiple service networks, Fixed broadband deployments with SIP-based VoIP, Transition phase from 2G/3G towards LTE by voice call services, Complete standardization by 3GPP towards LTE core-control architecture [1]

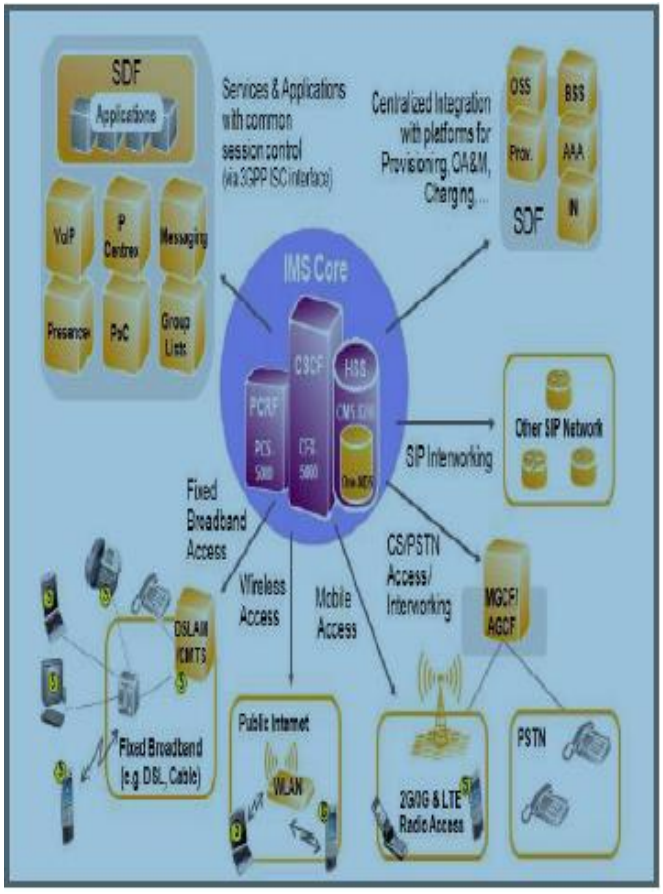

Fig. 1: IMS overview

CMRepo stands for Configuration Management Repository that acts as a very important pre-requisite for the bulk roll-out of Network Elements. These NEs include CSCF and HSS. It takes care of parameters belonging to NEs which are required for the functioning of IMS. CMRepo server provides the central Configuration Management (CM) to manage multiple NEs. CMRepo is also used for supporting NetAct, Open Northbound Interface (NBI) and Element Management System (EMS) belonging to Nokia as central CM system for maintaining the online parameters that are

changeable. Therefore, CMRepo is used for the management of these parameters.

II. RELATED WORK

The brief architecture of the IMS and its working is explained. This paper explains the various components of the IMS and their functionalities. It also covers the benefits of using the IMS. This paper can be used as the base paper for any future enhancements done in the IMS [1]. In [2], it is predicted that IP multimedia subsystem is the network framework introduced by the 3rd Generation Partnership Project, as an independent core network. IMS provides services for the multi-media domain and multi-session applications. Paper deals with the integration of IMS network with the packet switched domain from the mobile network perspective. Explanation of the IMS layered architecture and session establishment for the any user to get connected with the network by making use of the IMS core components such as CSCF (Call Session Control Function), HSS (Home Subscriber Server), and different gateways. The session management protocols are different from the general network protocols and it explains about the SIP (session Initiation Protocols). In [3], it explains that, IMS provides the services for many of the NGN (Next generation Networks) such as 4G-5G. As it also plays a very crucial role in all the mobile network application. The IMS has its independent layered architecture which contains three layers. Connectivity layer, Control layer, application layer. For the efficient working of the IMS, CSCF components must be equal efficient. The capacity of the controllers is studied and measured for the better applications.

III. PROBLEM STATEMENT

There will be number of updates or releases or new products being developed at Nokia. For each of these cases, it is mandatory to know how CMRepo behaves at different situations. These updates can either happen frequently or even rarely but for any new update testing the performance is compulsory.

Performance testing can be done whenever required. If manual process is involved, the accuracy of the report obtained will not be up to the mark. The CPU consumption for each second has to be noted clearly without a mistake and the average has to be calculated. It also takes more time to generate the whole report and becomes of no use when the report is need immediately or in a very short duration.

This project deals with problems, where the user need not have to perform any of the operations manually. It consists of few scripts that require to be copied to the respective nodes at the beginning. Once the scripts are copied, the user has to trigger one script on the CSCF node and wait for all the 150 operations to get over. Final report will be generated on the same node and the same can be used for the analysis.

IV. OBJECTIVES

The objectives of this project are: To know CPU and memory consumption at idle state of the system that includes both CSCF and Server**,** to know CPU and memory consumption before the operation begins and ends at CSCF and Server nodes, calculating the average of CPU and memory at all conditions that gives the overall consumption on both CSCF and Server nodes, capturing Relay, Proxy and Server time from CSCF node after the operation is completed, obtaining the data recorded in a particular format so that it is in readable form.

V. SYSTEM ARCHITECTURE

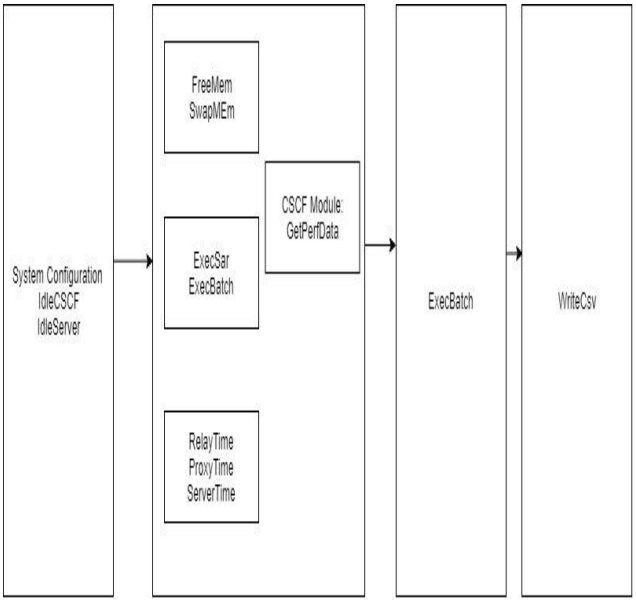

Fig. 2: System Architecture

The existing work at Nokia involved manual work. It involved a complex process, where CPU and memory details are obtained by first connecting to CSCF node. Then it requires connecting the server manually to execute the batch. During execution, another server node has to be connected to know how much CPU and memory are being consumed. There's a need to to always keep an eye on values for CPU and memory. Periodic checking has to be done to check whether the operation is completed or not. If the operation is successful, CSCF node has to be connected again manually to get the same details from the CSCF node as well as to record the Relay, Proxy and Server time. All the data obtained beginning from idle machine till the captured time, everything has to be updated into a file. This is done for a single batch operation. But there will be 150 batch operations to be done. Repeating the above steps for all the 150 operations manually, become time consuming and results in data entered wrong or any missing data as well.

Figure 2 shows the system architecture for CMRepo End-to-End Automation. It handles everything where it first executes script to capture CPU and memory details for idle condition of server and CSCF nodes, executes batch operation on server, capturing the same details from the same. It then executes script on CSCF to get the details of the same. It even captures relay, proxy and server time. All these are done without any manual work and the final .csv is obtained that contains all the data that is captured for all the operations performed.

VI. OVERVIEW OF METHODOLOGY

The script is first triggered on CSCF node. The system state is idle and involves no operation performed on CSCF. Therefore, CPU and memory consumption is calculated at the beginning from CSCF. Once the data is captured, the next step involves creating a .csv file on CSCF and writing the same details in the file [4].

The script from CSCF node executes another script on Server node. The system state is idle and involves no operation performed on Server. Therefore, CPU and memory consumption is calculated at the beginning from Server. Once the data is captured, the next step involves creating a .csv file on Server and writing the same details in the file [13].

After the details are captured from both idle CSCF and Server nodes, the commands for CPU and Memory on both the nodes are executed and it starts running. The next step involves connecting to Server and executing the batch operation. This operation involves Insertion, Deletion and Modification of rows in the tables. This takes time depending on the operation is being performed [5]. So once the operation is completed, the commands executing for CPU and Memory are stopped. Then CPU and Memory consumption is captured from the Server node and the same are appended to the .csv file created during idle condition of Server. This .csv file is then copied [6] to the CSCF node. The next step involves exiting from the Server node and connecting back to the CSCF node. Here as well, the commands for CPU and Memory are stopped and average is calculated to know the consumption. The captured details are written to the .csv created during idle condition of CSCF [11]. After capturing details from CSCF, the relay, proxy and server time are checked from logs and appended to the same .csv on CSCF.

To obtain the final report, First condition is to check in CSCF node, whether there exists a .csv file that consists of Server related data which is copied from Server node. Next condition is to check in CSCF node [12], whether there exists a .csv file that consists of CSCF related data created on the same node [10]. The next step involves merging these two .csv files on CSCF nodes that creates another .csv. This file gives end-to-end information for each of the operations performed.

Figure 3 gives the flowchart of how the operations are performed. This first step in that indicates calculation of CPU and Memory consumption for idle state of CSCF and Server and writing it to csv [7].

The next step indicates starting CPU and Memory on CSCF, connect to CMRepo Server start CPU and Memory, execute batch. Then calculate CPU and Memory consumption on Server and get back to CSCF node and calculate the average of CPU and Memory consumption. The data obtained so far is then written to csv file [9]. Then logs are checked on CSCF node to get the time for relay, proxy and server. The final result of the operation depends on the data captured so far [8].

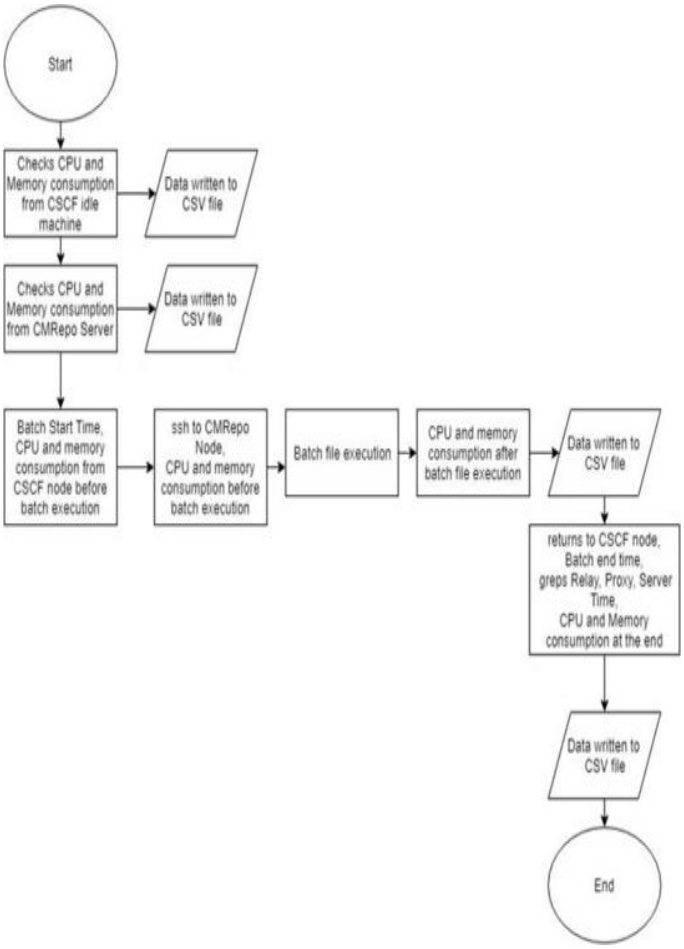

Fig. 3: Flowchart

VII. TOOLS AND TECHNIQUES

It is a computer program that decides whether the system is performing well under various workloads. This script makes use of certain unix commands to record the system activity and the final result is obtained in .csv file.

Different commands are used to get the CPU and memory consumption while performing the operation. They are SAR and free commands. SAR stands for System Activity Report. It is used for collecting, reporting and saving CPU usage, Memory usage as well as I/O usage in OS like system. In this project, CPU consumption is calculated for every second for a period of time for only User CPU and System CPU and average is calculated. Free command is used to check the memory consumption and how much memory is free. This project calculates both total available memory and swap memory. The average is calculated at the end.

Relay, proxy and server time are calculated from the logs after the operation is done. This time is taken from log with the help of grep command.

Python csv module is used for creating, writing and appending data that is recorded to the csv file. SSH protocol is used to connect to the two nodes which establishes secure connection and performs the required operations and exits from the connected node. The different tools used are WinSCP, XShell:

WinSCP is used for transferring the scripts from the local system to either CSCF node or Server nodes securely as shown in Figure 4. Any time the script can be modified and transferred to the nodes and can be executed. XShell is used as the terminal emulator, shown in Figure 5 that can be used for either professional purpose or personal use. It is a network program that exists for emulating the virtual terminal. Multiple languages are supported by XShell. Therefore, the levels of performing automation tasks are extremely superior with the help of XSHhell It helps in connecting to multiple nodes using ssh and executing any of the commands or scripts. Hence it helps in performing multiple tasks. It does not provide or support any unauthorized access.

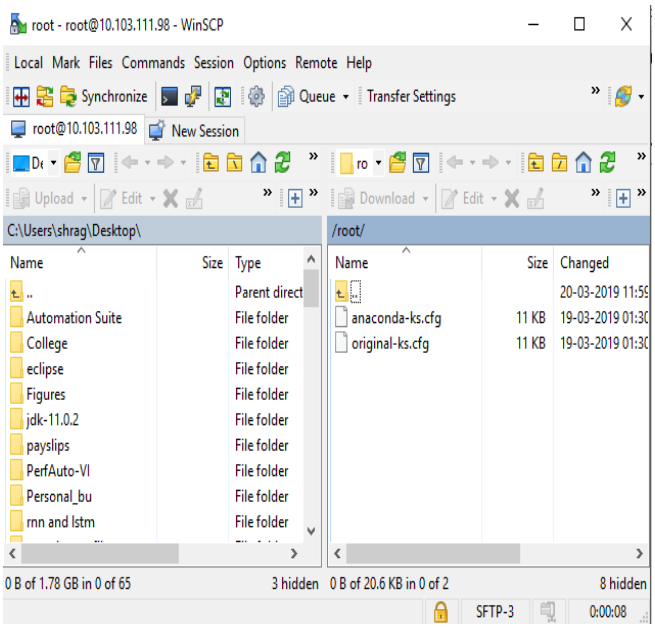

Fig. 4: Screenshot showing WinSCP

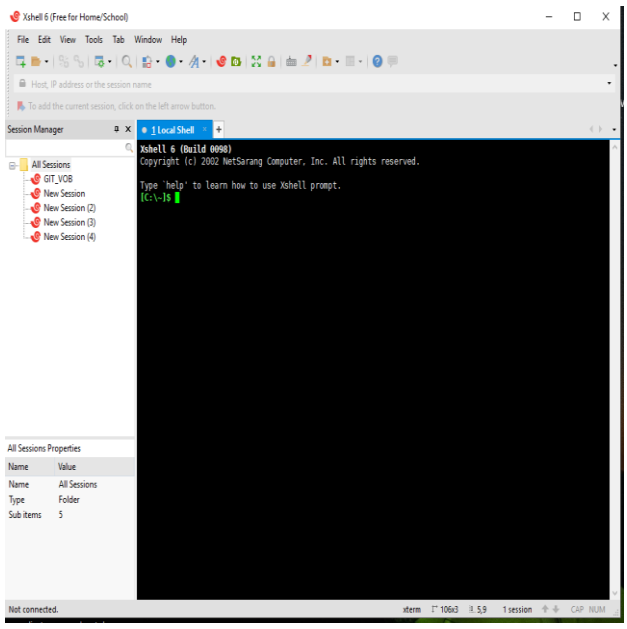

Fig. 5: Screenshot showing XShell interface

VIII. RESULTS AND DISCUSSION

CMRepo End-to-End Automation performs all the operations without any manual work by the user. The result of getting final csv is obtained at the end. Before the final csv is genertated, initially csv files will be created in the individual nodes, CSCF and Server. Once the operation going on Server gets completed, the resulting csv is copied to CSCF. CSCF contains one csv file with the data obtained from the same node.

At the end, both the csv files present on CSCF are merged together to obtain the final csv. This is done for 15 different operations, 11 insert operations, 3 delete operations and 1 modify operation for different batch files in the database. Once the script is triggered in the beginning, the work involved by the user is to keep checking the csv file to know the result. If the total time taken to perform the operation is taking more than 5 minutes, the result for that particular operation is considered to be failed else pass.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Table I	Ope	Server Tin	Relay Tim	Proxy Tim	Total Time	CSCF I	CSCF	CSCF I	CSC	SERV	SER	SERVER	SER\	Result
Idle co	-	-	-	-	-	0.365	0.4	2844	0	0.1	0.1	55417	0	-
POOL	INS	0:05:00	0:00:02	0:01:00	0:06:02	0.36	0.3	17013	0	0.1	0.1	55415	0	Fail
POOL	DEL	0:00:15	0:00:00	0:00:40	0:00:55	0.34	0.6	2806	0	0.1	0.1	55409	0	Pass

Fig. 6: Final csv result

Figure 6 shows the screenshot of the final csv result obtained. The different columns are, the first one is the name of the table or batch file to which operation is performed in the database, second is the operation performed, 3rd, 4th, 5th and 6th columns are Server time, Relay time, Proxy time and Total time taken. The next four columns are the data captured from CSCF. They include CSCF: User CPU (in %), System CPU (in %), Free memory (inMB) and Swap memory (in MB).

The next four columns contain the data that is obtained from the server. They include CSCF: User CPU (in %), System CPU (in %), Free memory (inMB) and Swap memory (in MB). The last column is the final result of the operation that is either Fail or Pass.

IX. CONCLUSION

This project, thus reduces the manual work of the user for calculating the different data on different nodes. It avoids any wrong information being collected as a result. It captures the correct data and calculates the average at the end without any mistake.

Multiple conditions are checked while capturing CPU and memory consumption. This project thus helps in carrying out these operations without any risk. It then takes different times to know if the right messages are being communicated by server, relay and proxy.

Therefore, this project helps in determining the overall performance of the system under various workloads and also determines if the operation performed is successful or not.

X. FUTURE SCOPE

Currently, CPU consumption is calculated for only user CPU and system CPU. This project can be enhanced to capture other data as well such as I/O wait, Idle CPU. Memory consumption is done for free memory and swap memory. This can also be enhanced to capture heap memory as well.

ACKNOWLEDGMENT

This project work, carried out at Nokia Solutions and Networks, is supported as part of major project for M.Tech, Software Engineering Post Graduate Program in the R.V.College of Engineering®, Bengaluru-560059.

REFERENCES

[1] V. Koukoulidis and M. Shah, "The IP multimedia domain in wireless networks: concepts, architecture, protocols and applications," IEEE Sixth International Symposium on Multimedia Software Engineering, Miami, FL, USA, 2004, pp. 484-490.

[2] W. A. Aziz, S. H. Elramly and M. M. Ibrahim, "Capacity Measurement of IP–Multimedia Subsystem (IMS) Controllers," 2010 Second International Conference on Computational Intelligence, Modelling and Simulation, Tuban, 2010, pp. 541-545.

[3] B. Dhindsa, A. Kaur and S. Ahuja, "LTE interfaces and protocols," 2015 International Conference on Advances in Computer Engineering and P. Tirana and D. Medhi, "Congestion avoidance in S-CSCF selection in an IMS network," 2010 22nd International Teletraffic Congress (lTC 22), Amsterdam, 2010, pp. 1-8.

[4] P. Tirana and D. Medhi, "Congestion avoidance in S-CSCF selection in an IMS network," 2010 22nd International Teletraffic Congress (lTC 22), Amsterdam, 2010, pp. 1-8.

[5] O. I. Romanov, M. M. Nesterenko, L. A. Veres and Y. S. Hordashnyk, "IMS: Model and calculation method of telecommunication network's capacity," 2017 International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo), Odessa, 2017, pp. 1-5.

[6] S. Zoric, J. Barakovic and H. Hodzic, "QoS architecture in IP multimedia subsystem of UMTS," 2008 50th International Symposium ELMAR, Zadar, 2008, pp. 253-256.

[7] X. Chen, W. Yan, Z. Wang, Y. Jin and G. Shi, "Optimization and simulation of satellite IMS access gateway and signal's propagation dela," 2017 IEEE International Conference on Real-time Computing and Robotics (RCAR), Okinawa, 2017, pp. 366-370.

[8] M. Pirhadi, S. M. S. Hemami and A. I. Tabrizipoor, "Call set-up time modeling for SIP-based stateless and stateful calls in next generation networks," 2009 11th International Conference on Advanced Communication Technology, Phoenix Park, 2009, pp. 1299-1304.

[9] J. Niemöller, E. Freiter, K. Vandikas, R. Quinet, R. Levenshteyn and I. Fikouras, "Multi- Technology Service Composition for the Telecommunication Domain - Concepts and Experiences," 2010 Fourth International Conference on Next Generation Mobile Applications, Services and Technologies, Amman, 2010, pp. 34-41.

[10] H. Pranoto and A. Ulvan, "Retransmission issue of SIP session over UDP transport protocol in IP Multimedia Subsystem - IMS," 2013 3rd International Conference on Instrumentation, Communications, Information Technology and Biomedical Engineering (ICICI-BME), Bandung, 2013, pp. 273-277.

[11] S. S. Wagle, N. Doulamis, M. Ade and M. G. Ullah, "Performance analysis of IP Multimedia Subsystem (IMS) signaling in heterogeneous access networks," 2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), Chennai, 2011, pp. 1-5.

[12] S. Zoric, J. Barakovic and H. Hodzic, "QoS architecture in IP multimedia subsystem of UMTS," 2008 50th International Symposium ELMAR, Zadar, 2008, pp. 253-256.

[13] Y. Cao, L. Zhang, Q. Qi and Y. Wu, "A novel paralleling session setup mechanism in IMS," 2011 4th IEEE International Conference on Broadband Network and Multimedia Technology, Shenzhen, 2011, pp. 250-254.

