RESEARCH ARTICLE OPEN ACCESS

Automating the Software Application Installation Process for Distributed Systems

Haripriya Regula[1], Mohammed Shiyabudeen Qureshi[2], Dr.Vaidehi vijayakumar[3]

P.G. Student, Department of Computer Engineering, Vellore Institute of Technology, Chennai, India[1]

Sub System Architect, Ericsson India Global Services Private Limited, Chennai, India[2]

Senior Professor, Department of Computer Engineering,Vellore Institute of Technology, Chennai, India [3]

Abstract:

To install System/Application software, patch, update, system-replacement in high network traffic controlled domain with prior operating system (OS) installation following the traditional way is a long time taking and continuous manual intervention procedure. Today Multi-National Companies are working over high complicated distributed systems whose software should compute in real-time. The demand for continuous service is also increasing for these applications and it is unacceptable to restart the system during software upgrade. This paper describes how to automate the entire process of software installation and maintenance irrespective of the hardware type (Bare metal/Virtual) using the command line interface (CLI) which facilitates the reconfiguration management of the orchestration tool System Orchestration Framework (SOF). The integration of hardware health check which is considered as a pre-installation phase, application installation over the Linux operating system which is considered into the installation phase and the validation of kernel values and all the application services which is considered as the post-installation phase using the continuous integration tool Jenkins thereby drastically reduce time and human effort.

Keywords **—Bare metal, Virtualization, Automation, SOF.**

1. INTRODUCTION

Software applications designed to serve in mission-critical distributed real-time and embedded systems (DRE) [1] are installed on either single host node or a cluster node based on the workflow and the disruptive operations that it is going to perform. Distributed System consists of multiple ruggedized servers, computing boards and workstations linked by computer network which are equipped with application and system software. For these mission- and safety-critical applications [2], it is unacceptable to shut down and restart the system during software upgrade, since monetary loss, interruption of service, and damage can be caused with a traditional installation process. The high availability criteria in the network communications industry require the services to be provided 24 hours a day, 7 days a week, with a near 99.99% uptime. Irrespective of the hardware type, the software application can be installed on both bare metal hardware [3] and virtual hardware [3]. Based on the customer requirement the application is installed on the specified hardware. The automation works for both bare metal hardware and virtual hardware.

Before Applications installation with prior OS installation, few checkslike System Board Generation, CPU speed, Memory size, Network Capacity, Controller Firmware version, ILO Firmware version,Basic Input Output System (BIOS) revision firmware version etc., has to be done on the hardware so as to ensure the hardware health whether it can support the software application or not. During application installation, all the network related details like internet protocol (IP) address, subnet, and gateway of the operation and maintenance network channel and all the other available network channels are to be provide, user login credentials while creating the user are to be provided, https service details are to be provided and infrastructure details like keyboard and time zone values of the server are to be provided, Domain Name System (DNS) servers details, Network Time Protocol (NTP) and Domain name are to be provided. In the post-installation phase, all the kernel values of a single host node or for all the nodes of a cluster are to be validated, control status and control version details of all the third-party applications, https enable status, business logic status, pace-maker status in case of high-availability and other software application related tests cases are tested.

SOF supports Graphical User Interface (GUI) as well as Command Line Interface (CLI). In either of the way the job can be triggered. And the job performs the application installation along with the OS installation on the chosen hardware type. Not only the maiden (first time installation) but also the update, system replacement, node expansion and node removal things can be made possible by choosing the appropriate installation type. The detailed inputs to be provided for SOF in creating the job successfully are explained further in this paper.

Automation can be defined as the technology by which a process or procedure is performed without human assistance. In other words, Automation, in various level of charging system is operating systeminstallation, 3PP software installation and 3PP software configuration. To do the same process in manual method required large amount of time and highly technical human resource are required.

In this paper we will have a brief description of SOF, SOF CLI, and Jenkins [14] integration and the difference in human's effort in the installation procedure with and without automation.

The remainder of the paper is organized as follows. Section 4 presents the overview of orchestration tool (SOF) and its configurations, the list of CLI commands to operate SOF tasks and a complete explanation of SOF usage through GUI. Section 5 presents the generation of SOF job input files (questionnaire, network, playlist, and package) and importing them automatically, the executing of pre and post installation test cases automatically and the integration of individual modules with the continuous integration tool Jenkins.The results are discussed in the Section 6 and finally Section 7 is presented with the conclusion and scope for future work.

2. LITERATURE SURVEY

Mission- and safety-critical software applications such as Internet infrastructure, aerospace, telecommunication, military defence and medical applications relied on multiple humane resources for deployment and maintaining. Furthermore, they also struggled to keep control over expenses from Resource. In support of the Automation, more than 95% of them were automated and the average response time for deployment was reduced to 50%, allowing for a significant boost in customer service offerings.

The various COTS tools [1][2] exist in the market but each has the limited capabilities. As RHEL satellite provides remote operating system installation solution limited to its own OS only. IBM's Urban Code [11] provides deployment features but no feature catering to OS installation. Similarly NINITE [10] does not provide missing file detection. The features Of SOF are compared with other tools in TABLE 1.

Table 1Feature comparison of SOF with other similar products

S.NO	Features	SOF	NINITE	URBAN CODE	FAI [12]
1.	Remote OS installation	YES	NO	YES	YES
2.	Remote Deployment of Applications/ Libraries/XMLs etc.	YES	YES	YES	YES
3.	Version Mismatch Detection	YES	NO	YES	NO
4.	Missing Files Detection	YES	NO	YES	NO
5.	Patch Installation	YES	YES	YES	YES
6.	Factory Restore	YES	NO	NO	NO

3. COMPARITIVE STUDY

In legacy approach operating system installation, application/system software installation and patch management are manual activities.

In Traditional method only one application can be installed at a time and usually with one man resource and one installation media, an average of 1 hour is required to make a node to be active [2]. Consider an example of a distributed system with 9 nodesand the system has to be made functional from scratch it takes 9 hours to make all the nodes operationalize. Using the orchestration tool SOF, the multiple nodes in the cluster can be installed in parallel there by reducing the waiting time that it takes for a sequential installation to one-ninth of its installation in parallel installation. Thus it drastically reduces human effort and computational resources and the entire manual process can be automated.

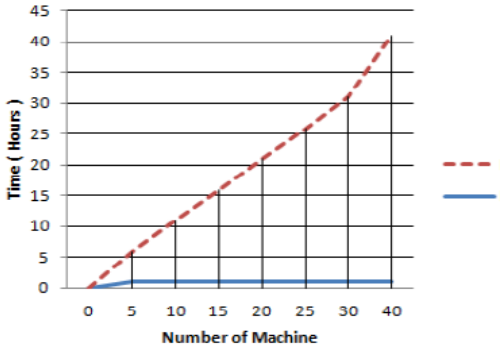

Figure 1Time graph of Legacy Approach vs. SOF

4. SYSTEM ORCHESTRATION FRAMEWORK OVERVIEW

SOF is a framework as well as a tool for performing remote upgrades andinstallations of nodes in a centralized way. Using its web interface, a user cantrigger and monitor upgrades on one or more nodes at the same time.

4.1. Workflow:

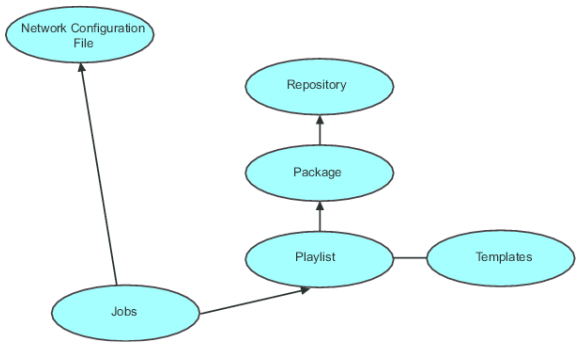

Figure 2 SOF Workflow

As shown in Figure 2 a job requires a playlist, which is located in a package which in turn is stored in the repository. The playlist may use a template file that holds some pre-populated values which can be changed during job creation if needed. The templates are also stored inside the package. Additionally, the job also required to know on which host/hosts it must be executed. Thus, it also has a connection to the applicable network list

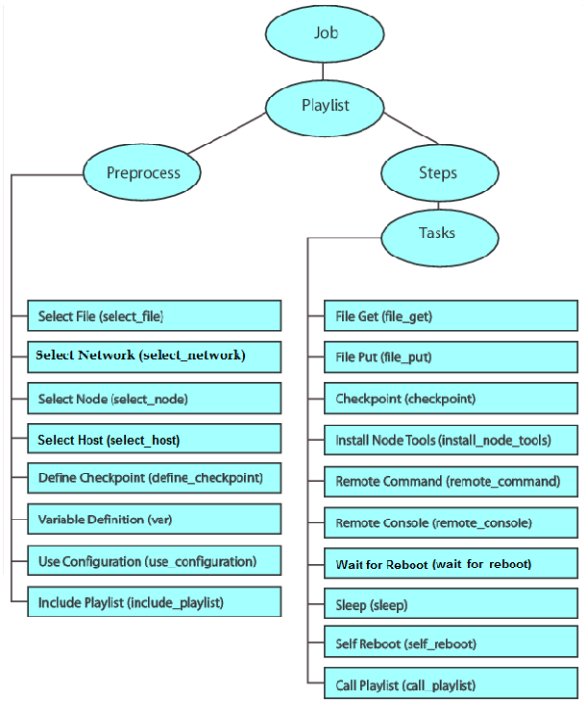

Figure 3 SOF Job Relationship

In a job, the relationships are as shown in Figure 3. A job consists of a number of steps. Each step contains a number of tasks. A task is the actual operation that the playlist writer wants the application to perform within a specific step. Before any jobs can be run, at least one network list must be configured in the application. This is done by importing the network list directly into the application GUI. If no network lists are available, the application will not know which hosts are available in the network, as this is specified in the network lists. The user can verify the available network lists directly under the network tab in the application GUI. It is very important to verify the availability of the required network lists.

4.2. SOF Network Configuration

The network configuration in SOF is used to store information about theSystem networks on which SOF operates.

The Network Model:

The network model is a logical model of the network and describes a networkas a hierarchical structure of elements and their attributes.Elements are entities, logical as well as physical, that constitute the network like nodes, hosts and users.An element has one or more standard (predefined) attributes associated with it,for example node or host name, node type, a host's address and so on. Anelement can also have one or more user defined attributes associated with it.An element can have zero or more child elements which appear below it inthe network hierarchy.An element has exactly one parent element which appears above it in thenetwork hierarchy.Elements that are at the same level in the hierarchy and share the same parentelement are called sibling elements.Figure 4illustrates the hierarchical structureof the network model and theelement relationships.

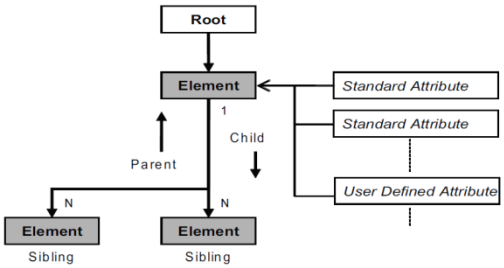

Figure 4 Hierarchical Structure of the Network Model

4.3. SOF Job Creation through GUI

Even though SOF is orchestrated, certain inputs must be given. Inputs like an XML based network file, an XML based playlist and excel file (Node Details) are validated and converted into informal file format. The excel file called questionnaire that contains internet protocol (IP) Address(Traffic Signalling-1 and Signalling-2 – IP addresses, Subnet – mask IP addresses and Gateway IP

addresses), system Node Type, hardware type(Bare Metal, Virtualization), operator country, update business logic product number, DNS servers, Domain and NTP details of a single node or for all the node in a cluster are filled up in separate sheets. Each node requires two sheets (HOST1 and HOST1-CONFIG). These two sheets are configured with the node specific OS related details for OS installation. CLUSTER-CONFIG sheet is for application configuration details and infrastructure related details in the INFRASTRUCTURE sheet. A single questionnaire should be created for all nodesof a single or a cluster node. The naming style and worksheet organization of a questionnaire file for a single system and for a cluster system has to be defined as shown in Figure 5 and in Figure 6 respectively.

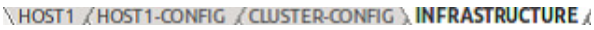

Figure 5worksheet naming style for single system

Figure 6worksheet naming style for cluster system

Following are the steps for a creating a job:

1. Playlist, Network/Node/Host & File Selection
2. Variable & Configuration Selection
3. Step & Task Selection
4. Review Summary

After making the input files (SOF Network File, Questionnaire File, SOF Package) ready, they have to be placed in the specified directory and the SOF network file and SOF package have to be imported into the SOF GUI. Then based on the installation type (Maiden, Update by patching, Fall back, System Replacement, migration) the respective playlist has to be chosen and respective questionnaire file has to be chosen in the first step of Job creation. In the second step, the user login credentials of a particular network are provided. These details are used in creating a user (say ROOT and GRUB). In the third step all the installation steps and tasks are defined with checkboxes. Required tasks or required steps of a particular task can be checked. In the fourth step, the summary of all the three steps are defined. Figure 7illustrates the architectural diagram of a SOF job. For creating a SOF job, the network file and the playlist to the respective cluster has to be available. Then have to follow the steps of SOF job creation. Based on the node type, either sequential or parallel installation takes place.

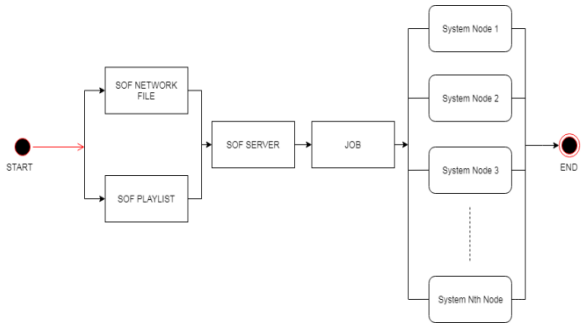

Figure 7 Application Installations through SOF

4.5. SOF Job creation through Command Line Interface (CLI)

SOF command line interface is sof. All the operations that can be done through user friendly interface can also be performed with commands. Rather than filling all the details through SOF GUI every time in creating a job, running a single script to perform all the necessary steps reduces the resource person's time and effort. Following are few basic commands used in the automation script to create a SOF job after satisfying all the pre-requisites.

Command to login to the SOF GUI: sof[-h <host>] [-po<port>] [-u <user_name>] [-p <password>];
Command to list the available network file in the SOF server: sofnetwork-list [-m]
Command to import a network file: sof network-import –f <network_conf_file>
Command to delete a network file: sofnetwork-delete [-f <filename.xml>]
Command to import a package: sof package-import –n <package_name> –f <package_file> (–nt<node_type> | -lib)
Command to delete a package: sof package-delete –n <package_name> (–nt<node_type> | -lib)
Command to create SOF job template:sof job-createtemplate –nt<node_type> -pkg<package> -pl<playlist> –f <file>
Command to create a SOF Job:sof job-create –n <job_name> -f <response_file>

5. PROPOSED AUTOMATION AND CONTINUOUS INTEGRATION

5.1. Integrating the scripts of pre-installation, installation and post-installation through Jenkins

The above mentioned manual work in providing the inputs in the format of Excel file, Network file and Playlist for a single host node or for a cluster node in both bare metal and virtualized hardware is automated. The process of pre-installation, creating a job in SOF and post-installation checks are automated. This pre-installation, installation and post-installation processes are integrated through a continuous integration tool called Jenkins.

Figure 8 is the architectural diagram of Jenkins job automation flow. The detailed description of the integrated modules and its workflow is as follows:
The parameters provided for Jenkins job creation are node names of a single host or a cluster, installation type, SOF server name, Open Virtual Framework (OVF) file path (for VM). These limited details makes the Jenkins job run and it don't even take a minute in triggering the job. After the Jenkins job got triggered, it will make a Secure Shell (SSH) connection with SOF server. Run the respective scripts in checking the hardware health status. Then starts running the script for questionnaire file generation with all the essential details filled appropriately in all the sheets of excel workbook, makes an SSH connection with SOF in order to place the generated questionnaire file in the

specified directory. In the next step it will trigger scripts to generate network file and makes an SSH connection with SOF server and checks its existence in the list of network files in SOF. If so delete and upload the latest generated network file using SOF CLI commands. Now it is time to trigger the SOF job importing the package file and run the scripts that makes the four steps SOF job to start running. Then after the successful completion of the SOF job, the post-installation scripts are triggered. All the test-case for the successful software application installation of all the nodes of a cluster is scripted to check automatically in single host node or in all the nodes of a cluster.

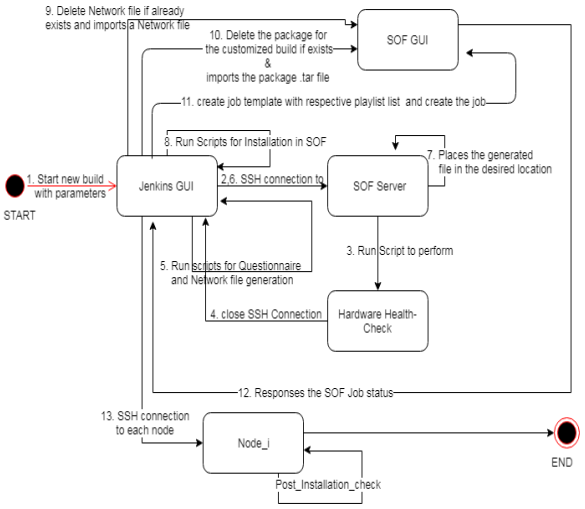

Figure 8 Automation Architecture Diagram

6. RESULTS

6.1. Auto-filled Questionnaire file generation

Figure 9 is the console output of the Jenkins job in generating the questionnaire file automatically.For a single node, a maximum of four worksheets of data has to be filled. For a cluster node, based on the number of nodes involved 2n+2 worksheets of data has to be filled. The efforts involved and time taken in filling up the data in multiple sheets of an excel workbook are made free by just providing the server names.

```
IF YOU ARE NOT AN AUTHORIZED USER, PLEASE EXIT IMMEDIATELY
root@aa.bb.cc.dd's password:
/usr/lib/python2.7/site-packages/openpyxl/xml/__init__.py:15: UserWarning: The installed version of
lxml is too old to be used with openpyxl
  warnings.warn("The installed version of lxml is too old to be used with openpyxl")
Questionnaire File Generated and placed under the directory /var/lib/sofstorage/
```

Figure 9Questionnaire File Generation

6.2. Auto-filled Network file generation

```
[2018/Feb/07 12:12:53] Checking if network UNI_BM_3829 is available in suf
[2018/Feb/07 12:12:56] Creating a network UNI_BM_3829 in suf
```

Figure 10Network File Generation

From figure 10 clearly shows that the Jenkins job checks for the existence of the network file in SOF. If not exists it will import the network file else deletes the existing one and uploads the generated one.

6.3. Automatic SOF Job creation

Figure 11 shows the result of a console output of a Jenkins job. It triggers the SOF job creation and will wait till the job gets completed and displays the job status.The four step SOF job creation is automated with the help of CLI commands which were used in the automation script. Once the pre-installation task is successfully completed, the SOF job creation task gets started. The progress of the SOF job can be seen in SOF GUI which can be identified with the SOF job id from the list of running SOF jobs.

```
Installation applicable

mdn

[2018/Feb/07 12:13:11] Automation Jenkins: Transferring deliverables to SUF
[2018/Feb/07 12:13:11] Automation Jenkins: Starting transfering sw to SUF

[2018/Feb/07 12:13:25] Automation Jenkins: SUF job creation
mdn
[2018/Feb/07 12:13:25] Automation Jenkins: Creating for suf rc file
[2018/Feb/07 12:13:25] Automation Jenkins: Checking if node OCC is available in suf
[2018/Feb/07 12:13:25] Automation Jenkins: Checking if network UNI_BM_3829 is available in suf

[2018/Feb/07 12:13:28] Automation Jenkins: Check if package is available

[2018/Feb/07 12:13:31] Automation Jenkins: Deleting package

[2018/Feb/07 12:13:39] Automation Jenkins: Check if package is available

[2018/Feb/07 12:13:44] Automation Jenkins: Create job template

[2018/Feb/07 12:13:49] Automation Jenkins: Executing the start job

[2018/Feb/07 12:14:09] Automation Jenkins: Running job with jobID - 6035

[2018/Feb/07 13:35:03] Automation Jenkins: Job Completed
```

Figure 11SOF Job Creation through Jenkins

6.4. Post-Installation Test-Cases auto execution

```
Collecting File list from config file for this file
=============================================================================================
S.NO   Test                          PASS    FAIL    TOTAL   STATUS
=============================================================================================
   1   sysctl                        19       0       19     PASSED
   2   os_version                    1        0        1     PASSED
   3   kernelcheck                   1        0        1     PASSED
   4   le_rpmcheck                   11       0       11     PASSED
   5   os_rpmcheck                   31       0       31     PASSED
   6   Ehardening_check              12       0       12     PASSED
   7   logfile_check                 9        0        9     PASSED
   8   PCS_monitor                   9        0        9     PASSED
   9   3PPControl_status             9        0        9     PASSED
  10   3PPControl_version            11       0       11     PASSED
  11   OnlineControl_status          1        1        2     FAILED
  12   OnlineControl_BL_status       0        10       10    FAILED
  13   https                         3        0        3     PASSED
  14   logcheck                      4        0        4     PASSED
  15   hardencheck                   2        0        2     PASSED
  16   ansiblecheck                  1        0        1     PASSED
  17   ntpcheck                      1        0        1     PASSED
  18   postvalidation                3        2        5     FAILED
  19   backupconfiguration_check     2        0        2     PASSED
  20   tcpwrappers_check             1        0        1     PASSED
  21   3ppcontrol_check              4        0        4     PASSED
  22   sshd                          14       0       14     PASSED
  23   pwquality                     5        0        5     PASSED
  24   hosts_allow                   1        0        1     PASSED
  25   hosts_deny                    1        0        1     PASSED
  26   resolv                        2        0        2     PASSED
=============================================================================================
**************** Result log -> /var/tmp/postcheck/result.log **************
=============================================================================================
```

Figure 12 Auto execution of post-installation task results

Figure 12 is the result of PASS/FAIL status of the post-installation test cases. It shows the name of the test case, total number of parameters, the count of PASSED parameters, the count of FAILED parameters and its test-case status. Of all the defined parameters to be checked under a test case, even if one parameter value is mismatched the test case status will be FAILED. The post installation check is made on all the servers defined in a cluster as a part of the automation and the node respective result log file of the post-installation will be placed under the temporary directory with the file name result.log in the respective node.

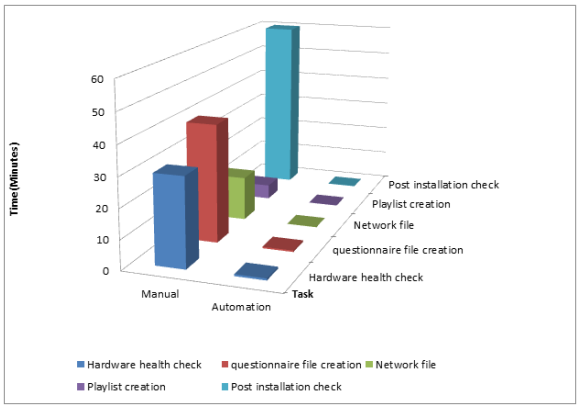

Figure 13Time graph of Manual tasks Vs Automation tasks

Figure 13 shows the efforts made on different tasks with respect to time to do the manual job and automatic job. As all the individual automated tasks are integrated in Jenkins, it takes a minute to trigger the job in Jenkins. By this the goal of automation is achieved and it is clear to state that the automation aids in managing the resource persons time and energy efficiently.

7. CONCLUSION& FUTURE WORK

Without automation, system installation and its implementations are observed to be hectic wherein the work results are based on human experience. An experienced person who is familiar with the system, dealt with usual and unusual scenarios of the system is considered to be efficient and knows the in and out of a system. Comparing this to a learning resource that is new to the system, this wouldn't be the case. The resource might not be familiar with the complete system behaviour and its dependencies. This brings mismatch to the knowledge among the resources though they work on a same system. Errors or mistakes caused by new learner might consume a lot of time to analyse the root cause of any issue which in return causes hardware crunch. Modifying the configuration data without complete domain knowledge will lead to unexpected system behaviour. A new learner might not follow the specification document, but automation is done only after the complete study of specification document ensuring no feature is misused. The gap in knowledge base and system learning is bridged by this automation technology. Root cause analysis might need efficient study of issue and requires time and resource.

Avoiding such dependency in analysis, automation provides quick fix to the issue by referring to help manuals/system docs which are verified. Tedious rework of certain steps which are time consuming can be executed very easily using common scripts.

A procedure of system installation/any system activities which might take hours to give output will now give them in very less time. This in return forwards the dependent task's time by giving tremendous efficiency hike in system performance.

As per the requirement, there are advantages and disadvantages with both bare metal hardware and virtual hardware. In future cloud infrastructure [4][5][6] plays a major part compared to virtual hardware. Organizations find more advantages with respect to cloud when compared to virtual hardware. Automation for cloud infrastructure is expected in future.

8. REFERENCES

[1] Sumit Jain, Manjeet Gupta, Rakesh, Aditya Baunthiyal, Central Research Laboratory Bharat Electronics Limited, Ghaziabad (India) – "IHMU: Remote Installation of OS/ System software/Application software/ Patch installation/ Version anomaly detection and System health monitoring in distributed system", 3rd IEEE Interna- tional Conference on "Computational Intelligence and Communication Technology" (IEEE-CICT 2017).

[2] Lizhou Yu, Anand Srinivasan, Department of Computer Science, University of Victoria, Canada – "A Framework for Live Software Upgrade"

[3] CharalamposGavriilKominos, Department of Computer Science, Uppsala Univer-sity, Uppsala, Sweden, Nicolas Seyvet and Konstantinos Vandikas, Management and Operations of Complex Systems, Ericsson Research, Kista, Sweden – "Bare-metal, Virtual Machines and Containers in OpenStack".

[4] Nancy Jain, SakshiChoudhary, "Overview of Virtualization in Cloud Computing", 2016 Symposium on Colossal Data Analysis and Networking (CDAN)

[5] Durairaj. M, Kannan.P, " A Study On Virtualization Techniques And Challenges In Cloud Computing", IINTERNATIONAL JOURNAL OF SCIENTIFIC & TECH- NOLOGY RESEARCH VOLUME 3, ISSUE 11, NOVEMBER 2014

[6] MeryemeALOUANE ,Hanan EL BAKKALI, "Virtualization in Cloud Computing: NoHype vs HyperWall", 2nd International Conference on Electrical and Information Technologies ICEIT'2016.

[7] Hypervisor,

https://searchservervirtualization.techtarget.com/definition/hypervisor
[8] Introduction to VMware vSphere, https://pubs.vmware.com
[9] Industrializing Network Functions Virtualization with Software-Defined Infrastructure https://www.ericsson.com
[10] NINITE, https://ninite.com/help/how-ninite-works/
[11] URBANCODE, https://developer.ibm.com/urbancode/products/urbancode-deploy/
[12] FAI, https://developer.ibm.com/urbancode/products/urbancode-deploy/
[13] Config File, https://docs.python.org/3.4/library/configparser.html
[14] Jenkins, https://jenkins.io/