Comparative Analysis Of Proxmox VE And Xenserver As Type 1 Open Source Based Hypervisors

Said Ally

Abstract: Advancement use of open source based type 1 hypervisors has transformed the way adopters manage their computing resources, lower IT operational costs and improved performance and flexibility. Use of open source virtualizations provides valuable IT solutions by enhancing high flexibility, scalability, security and performance. In this paper, Proxmox VE and XenServer as most popular type 1 open source based hypervisors are discussed and their functionalities compared. This comparative analysis has focused on hypervisor strengths on virtual resource allocation, virtualization approach supported, server architectures, number of virtual machines, operating system compatibility for host and guest machines and management features. General findings of the study suggest high similarity in many aspects; except that for adopters to attain maximum benefits of virtualized solutions based on open source computing will depend mostly on their adoption, usage and management practices which affect performance and security of virtual machines. However, for best fit and choice of hypervisor, adopter's virtualization requirements, organization size, personnel skills and computing workloads should be considered in advance.

Index Terms: Hypervisor, Open Source Software, Proxmox VE, Virtualization, XenServer

1 INTRODUCTION

SERVER virtualization has become essential component for today IT infrastructure. Virtualization is achieved using special software called hypervisors which provide efficient use of hardware resources, better infrastructure management and low IT costs for power, space and personnel. So, hypervisor is considered as main apparatus for adoption and investment in virtualization technologies. Thus, the capacity, compatibility and performance of hypervisors are very crucial in addressing server management challenges. Adopters need to be assured on hypervisor performance and its compatibility against processor architectures, host and guest operating systems, virtualization methods and their ability to handle critical functional features in management of virtual machines. Most popular and major IT projects are based on cloud computing which uses hypervisors technology [1]. Despite the remarkable price decrease brought by growth of virtualization technology, the cost of IT investment and operations can be reduced further through use of open source virtualization tools [2]. For that reason, this paper presents a comparative study of the two most popular type-1 based open source hypervisors which are Proxmox VE [3] and XenServer [4]. On the other hand, due to the fact that the main imperative and critical components for server performance are the CPU, RAM, HDD, disk I/Os and NICs [5], one of the review criteria used in this paper is the ability of computing resource allocation for each studied hypervisor. Other criteria to assess efficiency of each hypervisor include the virtualization methods, server architectures, number of VMs, compatibility with host and guest OSs and general management features. The reminder of this paper is structured as follows. Section 2 provides related literature review while in section 3 the research methodology is presented. Section 4 presents research findings where the hypervisor evaluation is done, and results are described. Section 5 provides concluding remarks, recommendations and discusses the future work.

2 LITERATURE REVIEW

2.1 Definition

Type 1 hypervisor is considered as a native, embedded, bare metal hypervisor that is incorporated and run directly in hardware to control resources [6]. Basically, type I hypervisor does not have an OS running below it, instead it is fully responsible for scheduling and allocating of systems resources between VMs. It serves as OS. The open source hypervisor as any other kind of open source software must fulfill the definition of open source software as defined by Stallman (1980) that include redistribute the software without restriction, access the source code, modify the source code and distribute the modified version of the software [7]. Generally, the open source hypervisor is available under the open source license GNU/GPL which allows free distribution of the software for use and improvements.

2.2 Review of Open Source Type 1 Hypervisors

Practically, type 1 hypervisors are more advantageous, robust, secure, and more efficient than type 2 due to characteristic of direct access of hardware resources [8]. This argument has also been supported by the IBM 2014 Research report that type 1 hypervisors provide higher performance, availability, and security than type 2 hypervisors [9]. In this paper, two open source (free OSS) based type 1 hypervisors are reviewed. These are XenServer and Proxmox VE.

2.2.1 Proxmox VE

Proxmox VE is an open source based virtualization solution which provides three major functionalities of compute, network and storage in a single package [8]. The Proxmox VE is based on the Debian GNU/Linux distribution [10] with Red Hat (RHEL) kernel. The hypervisor uses underlying technologies of Kernel-based Virtual Machine (KVM) hypervisor which supports full virtualization and LXC containers Open Virtuozzo (OpenVZ) which supports container based virtualization [11]. It uses both GUI and CLI to manage containers and virtual machines [12]. According to [8], the source codes of Proxmox VE are freely offered under the GNU Affero General Public License (AGPL), v3. Being with open

- *Said Ally is ICT lecturer and currently pursuing PhD in Computer Science in Open University of Tanzania, E-mail: said.ally@out.ac.tz*

source codes, Proxmox VE allows software scalability property to integrate any emerging technologies, full access to all functionalities, and high level of reliability and security.

2.2.2 XenServer

The XenServer is a bare-metal described as type 1 hypervisor [13] comes with GNU/GPL open source free license [14] and integrated into multiple cloud platforms [15]. XenServer is used by world's largest clouds and enterprises [4] and has become crucial for mission critical applications [16]. A computer running the Xen hypervisor contains three components: (i) Xen Hypervisor, (ii) Domain 0, the Privileged Domain (Dom0) and (iii) Multiple DomainU, Unprivileged Domain Guests (DomU). The term 'domain' refers to the VM or guest OS [17] with Dom0 for guest management using Linux kernel and DomU for guest virtual machine running in guest OS. According to [17], the Xen hypervisor supports the ARM architecture. The ARM is a short for the Acorn RISC Machine (originally) and Advanced RISC Machine (now). The ARM is a family of Reduced Instruction Set Computing (RISC) architectures i.e. the computer design approach which means processors require fewer transistors than typical Complex Instruction Set Computing (CISC) x86 processors. Also, from the research by [8] the Xen hypervisor works and supports hardware processors with x86, x86-64, Itanium, Power PC, and ARM.

2.3 Review of Server Virtualization Methods

Basically, several virtualization methods exist. The [18] and [19] have identified Full Virtualization (FV), Para Virtualization (PV) and Operating System Level Virtualization (OSV). However, according to [17], the virtualization methods are classified as Full Virtualization (FV), Para Virtualization (PV) and Hardware Assisted Virtualization (HAV).

2.3.1 Full Virtualization (FV)

In FV, also called Hardware Emulation (HE), hypervisors interact directly with the hardware resources [20] where the VMs or guests co-exist on a single physical machine. The guests run independently and are unaware of each other for their existence. In FV, guests can run different OSs [19]. In FV, the hypervisor consumption of processing power and resources is high, so the overall server performance is compromised because the amount of resources allocated to run applications and VMs become limited. In FV, the unmodified guest OS run [21] where the structure of OS and applications are maintained. The FV is achieved using techniques of direct execution and binary translation.

2.3.2 Para-Virtualization (PV)

The PV uses hypervisor like FV, but the guests are aware of the existence of each other. The PV requires less processing power to manage the guest's OSs and the resource demands of each one are known since the whole system works together as a cohesive entity. According to [21], the PV modifies the guest OS with open source characteristics.

2.3.3 OS Level Virtualization (OSV)

The OSV does not require hypervisor at all [19]. Since is possible because the host OS takes responsibility to perform all functions of a fully virtualized hypervisor. One disadvantage of this approach is fact that all guest machines need to run on a homogenous OS.

2.3.4 Hardware Assisted Virtualization (HAV)

The HAV is enabled by AMD-V and Intel technologies commonly in the x86-processor architecture [21]. This method provides way for virtualization of the CPU, memory and I/O resources.

2.4 Review of CPU Architectures

The most common server architectures in virtualization include x86, x64, POWER and SPARC. The x86-architecture is the most popular CPU architecture [21] with tremendous benefits. The report of Gartner's analyst shows that about 80% of x86 server architectures have been virtualized [1]. According to [22], the x86-based CPU represent the Intel family of CPUs which are x86-based systems and Instructional Set Architecture (ISA) for all Intel processors end in '86'. Other types of CPU architectures include x64, POWER and S PARC. On the other hand, the x64 is simply an x86-based 64-bit or x86-64 computer architecture (hardware) for Intel and AMD. The Scalable Performance ARChitecture (SPARC) involves family of RISC CPUs. Unlike the Complex Instruction Set Computing (CISC), the Reduced Instruction Set Computing (RISC) is a CPU design that involves simplified instruction set for higher performance, so the RISC processors use the clock per instruction (CPI). The Performance Optimization With Enhanced RISC CPU (POWER CPU) architecture, is a family of RISC-based CPUs that use the Power Architecture from IBM. The previous studies show that there is high underutilization of traditional x86 standalone server systems with which only 10% to 15% capacity of a server on average is utilized [5] implying that high percentage of server resources are unused.

2.5 Review of Guest OS

The guest OS run inside the VM and it is responsible for running guest applications. The OS uses hardware resources allocated dynamically through hypervisor [25]. The most common used guest OSs are CentOS, UBUNTU, FEDORA, BSD and all versions of MS Windows OS.

3 RESEARCH METHODOLOGY

This study has been conducted using two major approaches. First, an empirical study has been adopted to explore user experience on hypervisor usage, configuration and management. This approach was useful to overcome limitation of acquiring permission to access physical servers. Secondly, an onsite server screening was conducted to evaluate the VMs configurations, management and performance in two hypervisors. The Proxmox VE version 4.8 release of Dec 2016 and XenServer version 7 release of May 2016 were used in this study. The major focus area of the study are the efficiency in number of VMs, supported server architectures, compatibility with host and guest OS, number of virtual CPU and RAM, disks efficiency, virtualization methods and management features. The study was conducted in three public and five private firms using the Proxmox VE and XenServer virtualization platforms in Tanzania. The eight selected organizations are believed to provide good representation of the entire population for research problem. In each firm, server administrators were purposively selected as research respondents [23]. The system administrators are responsible for the day to day server operations in their firms. Considering the nature of this research, a wide range of other research techniques have been applied for the novelty, relevancy and feasibility of results. For instance, a critical analysis of literatures and expert review has been done [24].

4 RESEARCH FINDINGS

Throughout this study, the following results were found.

4.1 Comparative Item – Virtualization Methods

The four common types of virtualization methods used for assessment in this study are the FV, PV, OSV and HAV. When comparing the virtualization methods for the open source based type 1 hypervisors, these VT methods were found to be common. The results show that while Proxmox VE supports FV and OSV, the XenServer supports PV and HAV as indicated in *Table 1* below.

Table 1: *Hypervisor vs. Virtualization Methods*

SN	Hypervisor	Virtualization Method			
		FV	PV	OSV	HAV
1	Proxmox VE	√		√	
2	XenServer		√		√

For the case of XenServer, while the hardware virtualized VMs require x86 CPUs either Intel's VT-x or AMD-V based while operating HAV virtualization, the PV does not require processor capabilities instead they modify guest OS. On the other hand, unlike XenServer, the Proxmox VE supports OS level virtualization which is not common to type 1 hypervisors. Based on the literatures, most of the type 1 based hypervisor support PV and HAV. The performance of FV may not be ideal due to involvement of binary translation during the run-time. The binary translation is time consuming and can lead to extremely performance overhead. An alternative cache usage can be adopted to improve binary translation, but this could add more cost of memory usage. A FV on x86 architecture can provides performance between 80% to 97% of the host machine. The PV overcome performance issue of FV because the OS is able to recognize existence of hypervisor where it sends hypercalls. However, PV requires open source based modified OSs, so all MS Windows cannot be used as guests for XenServer due to virtualization approach of PV.

4.2 Comparative Item – Server Architectures

The hypervisors were compared their compatibility against four CPU architectures which are x86, x64, POWER and SPARC. With regard to the assessed open source based type 1 hypervisors, the results gathered from this study show that only two server architectures which are x86 and x64 are supported by all four hypervisors. Subsequently, all tested hypervisors do support neither POWER nor SPARC architectures. *Table 2* below shows the summary of correspondence between hypervisor and CPU architectures

Table 2: *Hypervisor vs. CPU Architectures*

SN	Hypervisor	x86	x64	POWER	SPARC
1	Proxmox VE	√	√	X	X
2	XenServer	√	√	X	X

The results show that both Proxmox VE and XenServer support equally the CPU architectures, that is they both support x86 and x64 and are incompatible with POWER and SPARC. To run XenServer, an Intel VT or AMD-V 64-bit x86-processor is required. Generally, all servers showed higher performance in 64-bit computing than 32-bit computing despite the fact that other factors such as CPU clock speed, size and architecture of memory and peripheral buses remain constant.

4.3 Comparative Item – Guest and Host OS

In this aspect, the interest was to check the compatibility of Proxmox VE and XenServer against famous used OS. The guest OS used for comparison are the SUSE Linux, RED HAT, UBUNTU, CentOS, FEDORA, FREE BSD and MS Windows. The *Table 3* shows how each hypervisor correspond to the selected guest OS.

Table 3: *Hypervisor vs. Guest Host OS*

SN	Guest OS	Proxmox VE	XenServer
1	UBUNTU	√	√
2	CentOS	√	√
3	FEDORA	√	X
4	FREE BSD	√	X
5	Linux SUSE	√	√
6	MS Windows	√	√
7	REDHAT	√	√
8	RHEL		√

Based on the findings, Proxmox VE is the only open source based type 1 hypervisor which works efficiently with all guests OSs. Unlike Proxmox VE, the XenServer does not support FreeBSD, NetBSD, or any other BSD variants as guest OSs. In Windows OSs, a XenCenter management console can run and connect securely to XenServer using a 256-bit AES SSL encryption. XenServer has a support of MS Windows OS. However, regarding the 32-bit and 64-bit OSs as guests, XenServer support both. It has ability to boot OS on another disk partition as guest.

4.4 Comparative Item – Number of VMs, Processor and Memory Characteristics

Assessing number of VMs supported by each hypervisor, their processing ability and memory characteristic is crucial for resource allocation and server consolidation. These aspects have significant impact on VMs performance. The *Table 4* below summarizes these aspects for Proxmox VE and XenServer hypervisors.

Table 4: *Number of VMs, CPU and RAM Characteristics*

SN	Characteristic	Proxmox VE	XenServer
1	No. of VMs	Varies	500
2	Max. No. of vCPUs per guest	Varies	32
3	No. of virtual CPUs	160	160
4	Max. RAM size on host	Varies	1 TB
5	RAM per VM	2000 GB	128 GB

The maximum number of VMs in XenServer is 500 although for Linux based VMs this amount can grow up to 650. However, number of VMs that can run concurrently with acceptable performance depends on the available resources and the guest workload. This dynamicity applies in Proxmox VE. Regarding the vCPUs allocated to a single VM, for

XenServer, this depends much on the guest OS used. However, XenServer supports up to 32 vCPUs per guest. For 32-bit guest OS, the maximum number of vCPUs that can offer better performance is 8. To avoid VMs starvation due to unequal resource allocation, XenServer uses a fair share balancing algorithm which ensures that the CPU resources split between VMs accordingly. Additionally, XenServer performs dynamic allocation of physical processors to any specific VM. The dynamic allocation allows VM to efficiently utilize the available CPU resources whenever they are available. One good thing to XenServer is its ability to use different types of CPUs in the same resource pool. Although for high efficiency, it is recommended to use the same CPU type (homogeneous resource pool) but it is possible to handle hosts which come with different CPU types (heterogeneous). Normally, XenServer uses technologies of FlexMigration and Extended Migration for Intel and AMD respectively to operate heterogenous environment. For the Proxmox VE, the number of vCPUs that can be supported per host is 160. For the memory allocation, despite the fact that XenServer offers maximum RAM size of 1 TB for host system but the maximum amount of RAM on a single guest is 192 GB. Basically, amount of vRAM varies depending on the type of guest OS. For instance, for the 32-bit Linux guest OS; the maximum amount of RAM on the host is 128 GB. One advantage of XenServer is to optimize the VM memory usage using Dynamic Memory Control (DMC) which automatically adjust memory of running VMs. The DMC keeps the memory values that guarantee VM performance between specified minimum and maximum memory values. For the Proxmox VE, the maximum memory size per host is 2 TB and memory can be allocated dynamically by specifying minimum and maximum values or can be fixed.

4.5 Comparative Item – Virtual NICs

For the virtual Network Interface Cards (NICs), the maximum number of NICs XenServer can allocate per VM is 7. However, there is a variation based on the guest OS. The maximum number of physical NICs supported is 16. With 16 physical NICs, means XenSrerver can support up to 8 logical networks with a limit of 4 NICs per bond. XenServer offers a fair share of the network I/O resources among VMs and control bandwidth through Open vSwitch. The Open vSwitch allows creation of private networks for VMs isolation purpose. Additionally, multiple physical network connections are supported by XenServer regardless they are in homogeneous or heterogeneous environment. This ability goes together with support of VLANs on a physical network interface. XenServer provides isolation property where the virtual networks do not pass network traffic to all VMs because the Open vSwitch acts as a Layer 2 switch which allows a VM to see only traffic designated for that specific VM. This is a very useful aspect in VMs management especially for multi-tenancy services where a high level of isolation and security is required. For the Proxmox VE, the virtual NICs allow connectivity between all VMs, nodes, and shared storage systems using a virtual bridge setup which connects a Proxmox virtual network with a physical network. In Proxmox VE, each node can support up to 4,094 bridges with a common naming format of **vmbrX**, where X represents an integer between 0 and 4,094. One of the strengths of Proxmox VE is to allow live migration without interruption on network connectivity if and only if the same bridge configuration is entered on all nodes, thus allowing the bridge to be used from any nodes in the cluster.

4.6 Comparative Item – Virtual HDDs and Disk I/Os

For the case of virtual disk, I/Os, the maximum number of virtual disk drives XenServer can allocate to a VM is 16, however this may vary depending on the type of guest OS. The disk I/O resources split between the VMs are done using a fair share method based on the set priority levels defined. XenServer uses storage manager which is a built-in support for a file and block based storage types. For the Proxmox VE, the supported storage include directory which cater for local storage, Logical Volume Management (LVM) for local or shared iSCSI targets, Network File Systems (NFS) for (OmniOS, FreeNAS, Ubuntu, etc), Gluster File System (GlusterFS) and RADOS Block Devices (RBD). For the backup purpose, Proxmox uses FreeNAS. Although NFS share is the most commonly used for a VM backup, the full backups can also be taken using local and Ceph FS. The full backups cannot be taken on LVM and Ceph RBD storage.

4.7 Comparative Item – Management Features

Several management features in hypervisors were assessed to compare the two hypervisors. Management features in the hypervisors are very crucial for easy creation and management of VMs. Some of the management features include asset management, configuration snapshots, high availability, live migration, maintenance mode, performance metrics, storage migration, VM cloning, VM migration, capacity planning/management, virtual firewall, VM backup/restore, thin provisioning, configuration mapping, performance reports, auto discovery, failover, and multiple host resource pools. The Table XX below shows how different management features are supported by Proxmox VE and XenServer hypervisors. *Table 5* below shows the mapping between hypervisor management features against Proxmox VE and XenServer.

Table 5: *Management Feature vs. Hypervisors*

SN	Management Feature	Proxmox VE	XenServer
1	Live Migration	√	√
2	High Availability	√	√
3	Storage Migration	√	√
4	VM Cloning	√	√
5	GUI and CLI Features	√	√
6	VM Snapshots	√	√
7	Capacity Planning/Management	√	X
8	Virtual Firewall	√	√
9	VM Backup/Restore	√	
10	Thin Provisioning	√	√
11	Configuration Mapping	X	√
12	Performance Reports	√	√
13	Failover	√	√
14	VM Backup/Restore	√	√
15	Server Consolidation	√	√
16	Development & Testing	X	√
17	Business Critical Apps and Workstation	X	X
	Total (%)	**14/17 ~ 82.4%**	**15/17 ~ 82.2%**

From *Table 5* above, it can be seen that both hypervisors offer the similar management features with average of 82% for each. This implies that there is no significant difference between the two hypervisors. However, there are major differences on how these features work and produce the expected results. For instance, considering the Graphical User Interface (GUI) and Command Line Interface (CLI) editions in XenServer, the result show that all versions of XenServer have CLI named “xe”. The CLI in XenServer allows direct access to the host through a terminal emulator connected using a serial port. The CLI installed in Windows or Linux machines allows XenServer to be accessed remotely. This remote access uses a 256-bit SSL (AES) encryption to secure all communications between the CLI and XenServer host. For the case of Proxmox VE, the supported GUI functionally posses four roles which focus on data center management, nodes, KVM virtual machine and OpenVZ container. Another VM management of interest is the ability of hypervisor in making VM clone. The XenServer creates VM clone and is able to generate VM template useful for auto-creation of additional VMs and for installing most popular guest OS. One way on how this is done is through support of thin cloning of existing VMs on local disks formatted as EXT3 and NFS storage repositories. Regarding the virtual firewall, both hypervisors possess the management feature to enhance security with internal capability of packet filtering and monitoring. From security perspective, XenServer consist of a control domain (Dom0) which is a secure, privileged VM that runs XAPI (the management toolstack) useful for controlling VM lifecycle operations in networking, storage, authentication and computing resource management. Proxmox VE uses a role based user and permissions management to monitor VMs, storage pools and nodes. Also, Proxmox VE applies user authentication using Linux PAM or LDAP. The two hypervisors also provide live migration, High Availability (HA), VM snapshots, thin provisioning and performance reports. They also offer virtualized server isolation, server consolidation, software development, cloud computing services. They also offer live memory allocation. Generally, both hypervisors possess limited features for adaptive analytics useful for reuse of past performance to determine the severity of problems and asset management feature for keeping track of VMs and their resource usage.

4.8 Comparative Item – VM File Formats

Another aspect assessed was to study the simplicity of VM importation, exportation and migration within the same or different hypervisor. Basically, XenServer supports imports and exports of VM in homogeneous and heterogeneous environment. This implies that XenServer supports industry standard Open Virtualization Format (OVF), so it can accept all VMs created from other hypervisors including the proprietary hypervisors such as VMware and Hyper-V. This is mainly achieved using XenServer Conversion Manager. On the other hand, Proxmox VE supports qcow2, raw and vmdk. The storage supported by qcow2 are NFS and directory. The raw format supports LVM, RBD, iSCSI, and directory. The vmdk supports NFS and directory. Due to high I/O overhead and low processing speed, the qcow2 format is not good for data intensive VMs such as database server. However, the qcow2 files are useful when there is a budget constraints and limited storage space. This image type supports KVM live snapshots to preserve VMs states. The raw image type ensures performance since VM has direct pass-through access, so it is faster. The raw file format can only provide a fixed-size or thick-provisioned VM image and so it is a preferred file format for all Proxmox VMs. The virtual drives that can be added in Proxmox are IDE (3), SATA (5), VirtIO (15), SCSI (13). The raw disk image files are always pre-allocated, so there is risk of over provisioning beyond the total available storage space. Another big advantage of this image type is its support on KVM live snapshots. Regarding the vmdk image format, the Proxmox uses it just for ease of VM migration to other hypervisors. Presence of vmdk file format allows Proxmox files to swim across heterogeneous hypervisor environment.

5 CONCLUSIONS

The Proxmox VE and XenServer are similar in many things but the main difference is the XenServer does not allow Linux containers (OS virtualization) while Proxmox VE supports. This feature promotes strength of Proxmox VE compared to XenServer. While Proxmox VE uses unique virtualization API, and take benefits of KVM full virtualization and OpenVZ container based OS virtualization, the XenServer is mostly utilize XAPI facilities. Both are popular hypervisors and require high level of OSS skills for secure management throughout VM lifecycle. Adopters are highly recommended to be well informed on the type 1 OSS based hypervisor best practices for adoption and usage to overcome any kind of malpractices. This goes parallel with being aware of threat sources, attack types and their corresponding solutions. The benefits of using Proxmox VE or XenServer can be realized based on the adopter’s practices towards adoption and usage. Depending on size and requirements of the adopting organization, how resources are allocated and shared among VMs play major role. Therefore, for type-1 OSS based hypervisors, Proxmox VE and XenServer are viable solutions for virtualized server infrastructures. For organizations that aim at cost cutting virtualization solutions with maximum performance, these two hypervisors satisfy. Proxmox VE for instance is an appropriate choice for adopters of all size regardless of resources should be distributed among guest machines. Basically, both two hypervisors can seriously be considered as virtualization platforms and ideal solutions with high compatibility level with hardware virtualization, guest and host OS, and open virtualization formats for easy VMs migration.

ACKNOWLEDGMENT

First, I would like to express my gratitude to Mr. Ajay Rao (IBM expert) currently working at UTL Technologies Ltd in Bangalore, India. He has guided me in setting up server virtualization environment using Proxmox VE. Thereafter I was able to work with XenServer using vast experience of working with Proxmox VE since both software are type-1 and are OSS based hypervisors. Secondly, am very thankful to the Open University of Tanzania for financing part of this work.

REFERENCES

[1] W. Graniszewski and A. Arciszewski, “Performance Analysis of selected hypervisors (Virtual Machine Monitors - -VMMs), International Journal of Electronics and Telecommunications, Vol. 62, No. 3, pp 231-236, 2016.

[2] B. Shrimali and H. B. Patel, “Comparative Study for Selection of Open Source Hypervisors and Cloud Architectures under Variant Requirements”, Indian Journal of Computer Science and Engineering (IJCSE), 2016, ISSN: 0976 – 5166, Vol. 7 No. 2, pp. 28-45, 2016

[3] Proxmox VE, “The Open Source Type 1 Hypervisor”

https://www.proxmox.com/en/proxmox-ve, 2018

[4] XenServer, “The Open Source Type 1 Hypervisor” https://xenserver.org, 2018

[5] D. Nandhagopal, “VMware and Xen Hypervisor Performance Comparisons in Thick and Thin Provisioned Environments”, MSc Dissertation in Interdisciplinary Telecommunications at the University of Colorado, 2012

[6] R. Singh, K. S. Kahlon, and S. Singh, “Comparative Study of Virtual Machine Migration Techniques and Challenges in Post Copy Live Virtual Machine Migration”, International Journal of Science and Research (IJSR), Vol. 5, Issue 3, ISSN: 2319-7064, pp.117–121, 2016

[7] N. Vainio and T. Vaden, “Free Software Philosophy and Open Source”, International Journal of Open Source Software and Processes, 4(4), pp. 56-66, October-December 2012, DOI: 10.4018/ijossp.2012100105

[8] G. C. Obasuyi and A. Sari, “Security Challenges of Virtualization Hypervisors in Virtualized Hardware Environment”, Int. J. Communications, Network and System Sciences, 8, pp. 260-273, 2015, http://dx.doi.org/10.4236/ijcns.2015.87026

[9] W. Felter, A. Ferreira, R. Rajamony and J. Rubio, “An Updated Performance Comparison of Virtual Machines and Linux Containers”, In Performance Analysis of Systems and Software (ISPASS), 2015 IEEE International Symposium On, pp. 171-172. IEEE, 2015

[10] Proxmox VE Server Solutions (2016), https://www.proxmox.com/, software version 4.4, Date Accessed: 03-January-2017

[11] A. Kovári, and P. Dukan, “KVM & OpenVZ virtualization based IaaS open source cloud virtualization platforms: OpenNode, Proxmox VE”, In Intelligent Systems and Informatics (SISY), IEEE 10th Jubilee International Symposium on, pp. 335-339. IEEE, 2012.

[12] R. Goldman, “Learning Proxmox VE”, Packt Publishing, ISBN: 978-1-78398-179-3, 2016

[13] R. P. Goldberg, “Architectural Principles for Virtual Computer Systems”, Doctoral dissertation, Harvard University, 1972

[14] I. Voras, M. Orlic, and B. Mihaljevic, “An Early Comparison of Commercial and Open Cloud Platforms for Scientific Environments”, In KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications, pp. 164-173, Springer, Berlin, Heidelberg , 2012

[15] Xen Hypervisor, “Xen Hypervisor-Software version 4.8”, https://xenproject.org/, Date Accessed: 04-January-2017

[16] XenServer, “XenServer-Software version 7.0”, http://xenserver.org/, Date Accessed: 04-January-2017

[17] Patel A., Daftedar M., Shalan M. and El-Kharashi M. W., “Embedded Hypervisor Xvisor: A comparative analysis”, In Parallel, Distributed and Network-Based Processing (PDP), 23rd Euromicro International Conference on, IEEE, pp. 682-691, 2015

[18] O. Kulkarni, S. Bagul, D. Gawali and P. Swamy, “Virtualization Technology: A leading edge”, International Journal of Computer Application, Issue2, Vol. 2, ISSN: 2250-1797, 2012

[19] J. Strickland, “How server virtualization works”. HowStuffWorks, 2008, http://computer.howstuffworks.com/servervirtualization.htm, Retrieved June 21, 2016

[20] B. A. Yauri and J. Abah “Mitigating Security Threats in Virtualized Environments” International Journal of Computer Science and Network Security, Vol.16 No.1, January 2016

[21] VMware, “A Performance Comparison of Hypervisors”

[22] PC Magazine, http://www.pcmag.com/encyclopedia/term/, 2016

[23] M. N. Saunders, P. Lewis., A. Thornbill, and M. Jenkins, “Research Methods for Business Students” (5th ed.). England: Pearson Education Limited, 2009

[24] R. Molich and R. Jeffries, “Comparative expert reviews”, In CHI ’03 extended abstracts, Ft. Lauderdale, FL, April 2003, 1060–1061

[25] R. P. Padhy, “Virtualization Techniques and Technologies: State-Of-The-Art”, International Journal of Global Research in Computer Science, (UGC Approved Journal), Volume 2, No.12, pp. 29-43, 2012

.