Cloud Computing Virtualization: A Comprehensive Survey

Amarildo Rista *, Jaumin Ajdari *, Xhemal Zenuni*
* SEE University/Faculty of Contemporary Sciences and Technologies, Tetovo, North Macedonia
ar29102@seeu.edu.mk; j.ajdari@seeu.edu.mk; xh.zenuni@seeu.edu.mk;

Abstract - **Cloud computing technology nowadays is a leading IT technology and introduces the next generation of distributed systems with a great acceptance and impact in the businesses. It influences the businesses on structural and operational processes with potential to reduce total costs and increase revenues. This technology is mainly service oriented and focuses on cost reduction, like hardware reduction, reduction on maintenance and total cost of ownership, and with different pricing models such as pay-per-use. Virtualization is the foundation process and technology in Cloud computing, as it hides the complexity of underlying hardware and software resources. It is the objective of virtualization to achieve an improvement in data security aspects, to provide flexible operations and data transfer, elimination (masking) of system failures and to ensure economic benefits. The aim of this research is through a systematic literature review of the existing work to provide a comprehensive understanding of the role and significance of virtualization within the domain of Cloud Computing**.

Keywords - Virtualization, Cloud Computing, Network Virtualization, Software Virtualization, Server Virtualization, Storage Virtualization

I. INTRODUCTION

Cloud computing technology is considered as a new generation in the field of IT. It is based on a Service Oriented Architecture [1] and Service Level Agreement [2]. Different definitions are given for cloud computing [3], considering it as internet-based computing whereby shared resources, software, storage and information are provided to client on demand. Three service models that are offered by Cloud Computing are: Infrastructure as a Service [4], Platform as a Service [5] and Software as a Service [6]. One of the most important topics in this domain is virtualization [7]. Virtualization is considered as the key technology in providing virtual platform of servers, operating systems and hardware devices such as storage. This platform helps the user by providing multiple machines at the same time. Also, it allows sharing a single physical instance of resource or an application to multiple users. Through cloud virtualizations is managed the workload and make it more scalable, economical and efficient. The environment in which virtualization is built can be private or public [8]. One of the features of virtualization is that it allows sharing of resources to multiple clients and companies. Thus, the client can maximize the resources and reduces the physical systems. Types of virtualization in Cloud Computing [9] are: Network Virtualization that combines the resources of network and its functionality in a single virtual network; Server Virtualization, where a single physical server is separated into multiple servers on demand; Software Virtualization, where a single physical server runs one or more instances of the same software and usually is implemented by use of virtual machines; Storage Virtualization which is considered as a sharing of physical storage into multiple storage devices. In this paper, a systematic literature review of the existing literature is presented with the aim of understanding of the role, the significance and the importance of virtualization in Cloud Computing. The rest of the paper is structured as follows: Section 2 presents a brief discussion on the related works. Section 3 presents the Methodology. Section 4 is focused on the literature review. Section 5 presents the discussion related to the overall results observed in the reviewed literature. And some conclusions are listed in the section 6.

II.RELATED WORK

Virtualization of resources is considered as a key element in Cloud Computing technology. It is the objective of virtualization to achieve an improvement in data security aspects, to provide flexible operations and data transfer, elimination (masking) of system failures and to ensure economic benefits. This section presents different virtualization technologies. One of the efforts in this domain is the network virtualization. The NetVM framework [10] is a tool of network virtualization by enabling high bandwidth network functions to operate at near line speed, while taking advantage of the flexibility and customization of low-cost commodity servers. GLANF (Glasgow Network Functions) framework [11] is another technology tool in network virtualization based on container which creates, deploy sand manages virtual network functions (NFs) to achieve low overhead, fast deployment and high reusability. Many studies are focused on the development of programmable networks through Software Defined Networking (SDN) [12]. OpenStack contains virtual network framework [13] based on SDN architecture in providing and managing the network environment and virtual network infrastructure. The improvement of the SDN is continuously proposed [14], as to minimize the response time, a dynamic controller assignment schema is proposed. In [15], the authors with the use of OpenFlow architecture into SDN have achieved a significant improvement in terms of SFC link throughput and packet loss. The full-stack SDN framework including all protocol layers of the network stack is considered as the next SDN generation, and in the [16] this idea is explained. Andromena is another network virtualization platform designed to improve agility, availability, isolation, and scalability of network [17]. This is a technology developed for Google Cloud Platform (GCP). Recently a new paradigm of Coud Computing, namely the serverless computing, is supported by Network function virtualization (NFV) technology and the research of this is found in papers [18], [19]. To scale cloud resources, it should be forecast the NFV workload, and a method called Wavelet-GMDH-ELM (WGE) for NFV workload forecasting in the paper [20] is proposed and the method predicts and ensembles workload in different time-frequency scales. Another important resource virtualization in Cloud Computing is storage virtualization and basic introduction of this is found in [21]. One way of the storage virtualization, is the so called elastic memory management, described in [22] where this method enables dynamic memory allocation over provisioning and controlling application performance and penalty. This adjustment is beneficial for the resource provider, since the host can take advantage of that memory reduction in order to dedicate it for other processes. In [23], a Heterogeneous Cloud Storage Platform is proposed with a variety of storage technologies and uniform distribution of data stored. This platform is supported by SDN architecture, where the files and parts of file are distributed on different storage servers. In [24] the MDev-NVMe as a new virtualization implementation for NVMe (non-volatile memory express) storage device is proposed and where high throughput, low latency performance and a good device scalability could be achieved. In order to implement huge data storage with high performance, NVMe over remote processor messaging (NVMe-over-RPMsg) is applied to heterogeneous Multi-Core SoCs. The software framework emulates remote storage system as local NVMe device without requiring VMs [25]. The increase of “big data” on cloud computing is another issue [26] and different technologies of storing big data are explained [27]. In [28] the authors have used a network storage device such as a NAS or a SAN to configure it to a large-capacity storage device, in a network environment. NDAS storage systems as well are used to configure a large-capacity storage device in a network environment [29]. In this work, it is shown that the increase of the number of disks into the NDAS system proportionally increases the performance. Virtualization of servers [30] and OS has a very important role in cloud computing. Through server virtualization the resources are more efficiently used, and from user point of view it eliminates the cost of hardware. In Cloud Computing, by use of this virtualization, the workload is balanced and is divided into multiple servers where sometimes virtual servers perform a dedicated task. In [31], the performance analysis of the server virtualization is done and as conclusion the authors have noted that a key element in system performance is the virtualization technology used, notably in case of big data applications. There are two virtualization technologies [32], virtualization technologies based on hypervisor and virtualization technologies based on container. In [33], it is referred that the Docker and LXC virtualization technologies based on container technology introduce lower power consumption compare to KVM and Xen, based on hypervisor virtualization. Also, in the Linux operating system it turns out that Docker is faster than KVM compared by the analysis performed in [34]. Another very important property which needs to be satisfied is the elasticity. Elasticity is one of the main cloud performance and measuring parameter. It measures the ability of a cloud to give fast response to consumer according to their need and demand. In [35], an optimize elasticity framework, based on container technology is proposed. The integration of Docker with OpenStack is another framework that implements a container management service called Yun [36]. Yun improves the container deployment and throughput as well as the system performance by optimizing the message transmission architecture between internal components. Another virtualization framework is the Docker swarm and Kubernetes technology [37]. This technology distributes the work on different nodes by avoid overloaded. In [38], an agent, called Elixir, is proposed. It supports the Docker Swarm to balance the workload among swarm nodes and automatic scaling-up of worker nodes. The issue of the overhead is analyzed in the paper [39] and the authors have proposed an adoption of the OverlayFS to reduce the overhead.

III. METHODOLOGY

This section describes the methodology we follow to design and conduct this research paper. We first describe the research procedure and the research focus. Then we present the metrics on which will be based the analysis and comparison of existing technologies.

I. Research Focus and Procedure

This paper provides a systematic literature review procedure as described by Kitchenham [40] and Brereton [41] on approaches and techniques of resource virtualization in cloud computing. The main goal of this study is to understand the role and significance of resource virtualization in Cloud Computing by considering its benefits. This study introduces the most significant and relevant techniques about resource virtualization that may provide some direction of how the architecture of Cloud environments should be designed. Because of the investigated subject is very extensive, it is impossible to describe all relevant topics. Hence, some related subjects will be briefly mentioned. Based on the searches obtained through Google Scholar, IEEE Xplore, ACM, Springer and Elsevier database we have selected the most relevant and cited articles that describe technologies about virtualization of resources. Referring to keywords, title and content of articles, we have divided them into three section as follows:

A. Different technologies in network virtualization
B. Different technologies in server and OS virtualization
C. Different technologies in storage virtualization

I. Metrics

The metrics on which will be conducted the analysis and comparison of existing technologies, are:
- Performance of the framework
- Flexibility of management operations
- Cost-Effective Strategy
- Smother IT Operations
- Security and system failure

Referring to the advantages and disadvantages of each technology we give some recommendations that which techniques are most relevant to design the architecture of Cloud environments in the future.

IV. VIRTUALIZATION ON CLOUD COMPUTING

A. Different technologies in network virtualization

There are a lot of publications that study network virtualization in cloud computing. One of the most cited publication in this field is "Network Virtualization and Software Defined Networking for Cloud Computing: A Survey" performed by Raj Jain and Subharthi Paul [12]. They have introduced the virtualization of layer 2 and layer 3 of network components by presented a software defined networking (SDN) which makes a special contribution to network virtualization. The most important innovation of the SDN is programmable control plan, which splits the network into several virtual networks with different policies between them and with share of a common hardware infrastructure. As shown in Fig.1, the SDN consists of a centralized control plan with an API for communication with hardware and network applications.

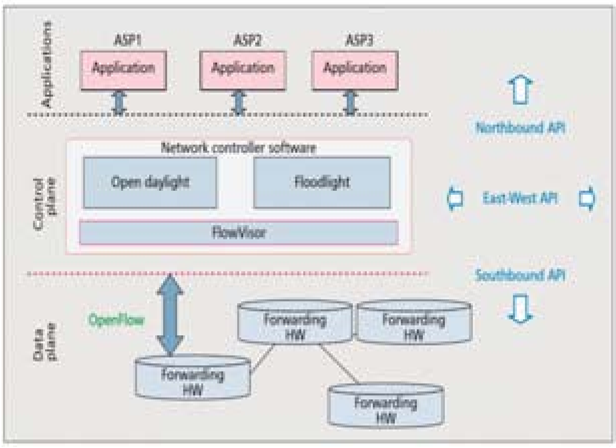

Figure 1. Software defined networking API [12].

Through SDN the networks can be made programmable, partitionable and virtualizable. A critical component for virtualizing SDN networks is hypervisor that abstracts the underlying physical SDN network into multiple logically isolated virtual SDN networks. In [42], a comprehensive understanding of network virtualization hypervisors for SDN is presented. The hypervisor is responsible to ensure that each user is independent from the other users operating a vSDN on the same underlying physical SDN network. In [43] an SDN controller platform called Meridian that supports a service-level model for application networking in clouds is presented. The Meridian platform architecture design is organized in three main logical layers: network model and APIs, network orchestration, and interfaces to underlying network devices as is shown in Fig.2.

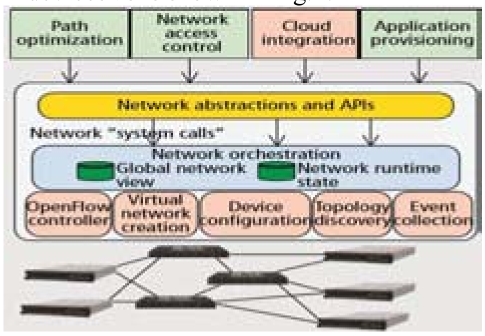

Figure 2. Meridian SDN cloud networking platform architecture [43]

The goal of the API layer is to present applications with a network model and associated APIs that expose only the information needed to interact with the network. The network orchestration layer plays three important roles in the SDN architecture. First, it performs a logical-to-physical translation of commands issued through the abstraction layer above. Second, it provides a set of network-wide services to applications. Third, it provides coordination and arbitration between network requests issued by applications, and mapping of those requests onto the network. And the lowest layer enables the controller to interface with various network technologies or tools. Through this platform can be dynamically update virtual network entities efficiently, and to scalable orchestration of network control as well as configuration tasks. Dmitry Drutskoy , Eric Keller&Jennifer Rexford in [44], introduce a FlowN framework to scalable network virtualization in SDN. This framework is based in two key concepts. First, it lets users to write arbitrary controller software that has full control over the address space and can target an arbitrary virtual topology. It is used a shared controller platform rather than running a separate controller for each user in analogy to container based virtualization. Second, it is used a modern database technology to perform the mapping between the virtual and physical address space. This framework gives each user the own address space, topology, controller, and leverages database technology to efficiently store and manipulate mappings between virtual networks and physical switches. In [45] a SDN/NFV management and orchestration framework is presented. This framework provides a multi-domain network orchestration and virtualization mechanism to offer dynamic and flexible end-to end connectivity and virtual network provisioning services across multi-domain and multi-technology networks. Also, it performs an integrated orchestration of distributed cloud resources and network resources to dynamically deploy virtual machines and VNF instances and provide the required network connectivity between DCs and end-points across heterogeneous multi-domain transport networks. In [46] a SDN integrated with NFV architecture is presented. It consists a control module, forwarding devices and NFV platform at the edge of the network as is shown in Fig.3.

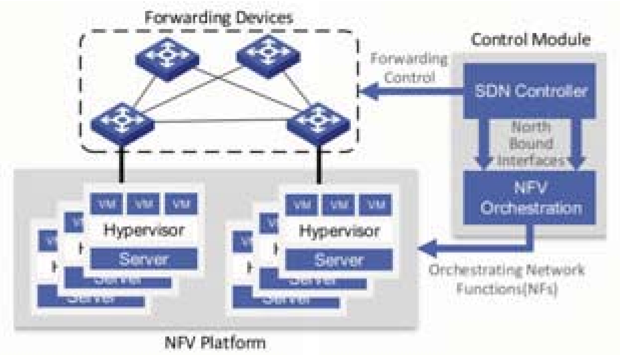

Figure 3. Software-defined NFV system [46]

The packet forwarding is done by the SDN controller according to logical forwarding tables. The Openflow protocol can be used as standardized interfaces in communicating between the SDN controller and distributed forwarding devices. Hypervisors run on the servers to support the VMs that implement the NFs. The SDN controller and the NFV compose the logical control module. The NFV is responsible for virtualized network functions and is controlled by the SDN controller through standard interfaces. While control module computes the optimal function assignments and chooses optimized routing paths. This framework promises agile traffic steering and joint optimization of network functions and resources. In [47], a network virtualization platform (NVP) is proposed in order to provide connectivity on demand. The NVP's architecture is shown in Fig.4.

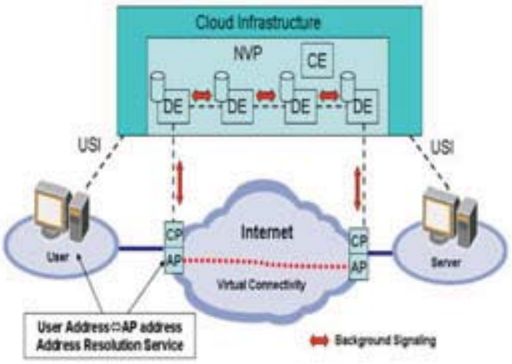

Figure.4. Network Virtualization Platform (NVP) architecture [47].

The NVP’s design is based on requirements and service oriented network architecture model [48, 49]. As shown in Fig. 4, the NVP is composed by a set of distributed entities (DEs) and a centralized entity (CE). Clients request NaaS by sending messages to the NVP via the user - service interface (USI). After this, the DE maps it in the parameters characterizing the CP connectivity. The CE is a database with the client' data and relevant service level agreement (SLA) for the end user authentication and relevant network service authorization. In this way NVP is placed to the cloud virtualization platform and it accepts on-demand NaaS requests from the client through a dedicated interface. Through this platform cloud user can request on-demand connectivity without any knowledge of the complexity and the network technology. A highly cited study in this domain, is conducted by Omar Sefraoui, Mohammed Aissaoui&Mohsine Eleuldj [50]. In this study an OpenStack solution framework is proposed. OpenStack architecture is built based on three main components: OpenStack Compute, Image and Object as is shown in Fig.5.

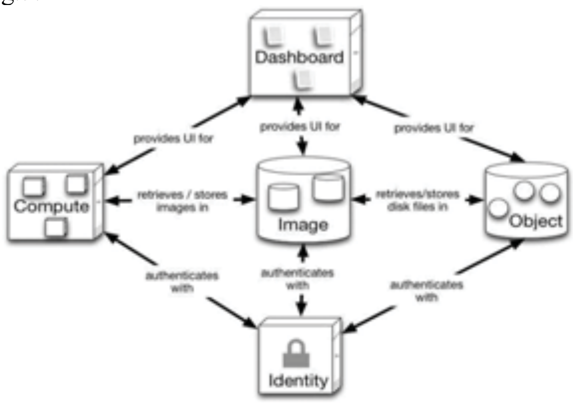

Figure 5. OpenStackArchitecture [50]

OpenStack Compute is a management platform that controls the infrastructure to control IaaS clouds. It allows managing large networks of virtual machines and redundant and scalable architectures. It provides an administrative interface and an API needed for the orchestration of the Cloud. Also, it includes instances management for servers, networks and access control. Imaging Service provides storage services, recording and distributing the images to virtual machine disks. While Object Storage is used to create a storage space redundant and scalable for storing multiple data. This framework promises a scalable, compatible, flexible and open source IaaS. Another study in this domain conducted by Franco Callegati [13]. In this study, a network virtualization architecture based on OpenStack is proposed. The OpenStack [51] provides cloud managers to control a cluster of hosting servers and to manage the virtual network infrastructures. The OpenStack cluster is composed by: a controller node that manages the cloud platform; a network node that host cloud networking services; a number of compute nodes that execute VMs and a number of storage nodes for data and VM images. To manage service networks, a separate component called Neutron which provides a flexible interface for virtual network management is used. An OpenStack user can create a new VM instance specifying the subnet that is assigned via DHCP. In Fig.6 is presented a virtual network infrastructure implemented by OpenStack. As shown in fig.6, a network node and a compute node are running two VMs

Figure 6. Network elements within an OpenStack network node [13].

The network node includes br-data for the data network and br-ex for the external network. While the compute node includes only br-data connected to the data network. Layer 2 virtualization is implemented by VLANs or Virtual eXtensible LAN (VXLAN) that is supported by the programmable features available in OVS through the insertion of appropriate OpenFlow mapping rules in br-int and br-data. When a new VM instance is created, the OpenStack chooses the best compute node that will host it. In [52], a cloud management platform based on OpenStack is proposed to optimize power consumption, host resources and networking. The main issue discussed in this paper is the problem of VM consolidation in cloud scenarios. We summarize briefly the simple use case of a new VM instantiation in order to understand how the different OpenStack services interact. In the first step each compute service periodically sends updates to a SQL server that stores load in formations about provisioned VMs and physical servers. When a new VM is requested the API Service sends a request to the scheduler service which chooses the compute service in which the VM should be launched. After, it sends a new VM instantiation request to the chosen compute node that requests network configuration parameters for the new VM to the network service. Based on experimental results is concluded that VM consolidation based on OpenStack is an extremely feasible solution to reduce power consumption but it has to be carefully guided to prevent excessive performance degradation. Another open source solution for virtual infrastructure management in cloud is OpenNebula [53]. OpenNebula is an open source virtual infrastructure manager that can be used to deploy virtualized services on both a local pool of resources and on external IaaS clouds. To manage a VM, the OpenNebula Core orchestrates three different management areas, namely: image and storage technologies to prepare disk images for VMs; the network fabric to provide the VMs with a virtual network environment; and the underlying hypervisors to create and control VMs. In [54], a comparison between OpenStack and OpenNebula is presented. It is concluded that both frameworks have their advantages and disadvantages depending on the application.

B. Different technologies in server and OS virtualization.

There are three types of server virtualization in cloud computing; para-virtualization, hypervisor and full-virtualization. In [55], is compared and evaluated the system performance of three techniques. The Xen-PV is used for para virtualization, Open VZ for container virtualization and XenServer for full virtualization. Based on various tests conducted by the benchmarking tools, the Xen Server used for full virtualization has comparatively higher system performance in terms of file copy, pipe based context switching, process creation, shell scripts and floating point operations than the Xen-PV and Open VZ. One of the most cited publication in this domain is the paper "Container-based Operating System Virtualization: A Scalable, High-performance Alternative to Hypervisors" [56]. In this paper a Container-based OS virtualization framework is proposed. This framework provides a shared, virtualized OS image consisting of a root file system, a set of system libraries and executables. Resources are assigned to each VM when it is created, but can be dynamically varied at run time.Fig.7 shows the design architecture.

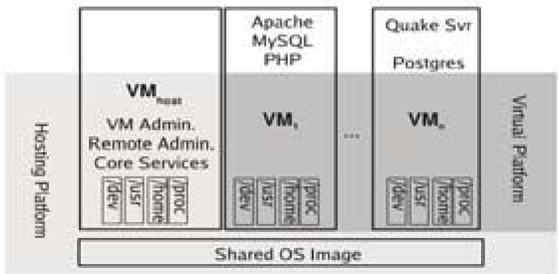

Figure.7 Container –based architecture [56]

As shown in the fig.7, there are three basic components. The hosting platform consists essentially of the shared OS image and a privileged host VM. The virtual platform is part of system seen by the guest VMs. Applications running in the guest VMs work as a corresponding non-container-based OS image. At this point there is little difference between a container and hypervisor based system. However, they differ fundamentally in the techniques they use to implement isolation between VMs. In this study, Linux-VServer technology was considered, comparing it with other technologies based on hypervisor. Based on experiments results container-based systems provide high performance compare to hypervisor based systems for server-type workloads and scale further while preserving performance. Also container based systems such as VServer include more features and compete strongly against hypervisor systems as Xen for various scenarios.

Another highly cited publication in this domain is "An introduction to Docker for reproducible research" [57]. In this study Docker technology is presented which combines several areas such as operating system virtualization, cross-platform portability, modular reusable elements, versioning, and a 'DevOps' philosophy. The Docker architecture [58] consists of four main elements; Docker client and server, Docker Images, registries and Docker containers as is shown in Fig.8.

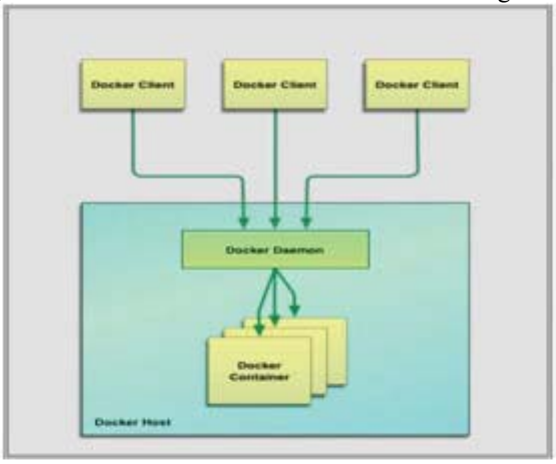

Figure 8. Docker architecture [58]

Docker is as a client-server application. The Docker server (daemon) and client can be run on the same host or local Docker client through a remote daemon run on another host. Images are the building blocks of the Docker. The images can be considered as the "source code" for containers. They are highly portable and can be shared, stored, and updated. Registers serves to store the Docker images. While a Docker container is considered as an image format, a set of standard operations and an execution environment. In [34], four different virtualization technologies are compared: KVM and Xen, which are based on hypervisor virtualization and Docker and LXC which are based on container virtualization. Hypervisor-based virtualization technology operates at the hardware level, while container virtualization operates at the OS level. Fig.9 (a) present the hypervisor-based virtualization architecture and fig.9(b) present the container - based virtualization architecture.

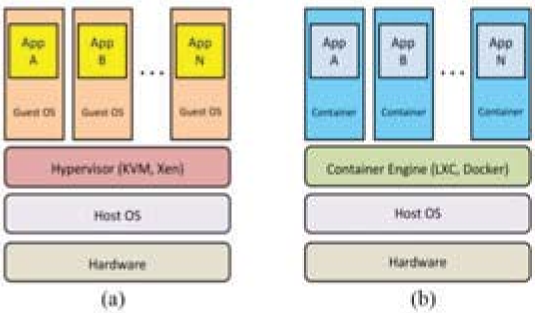

Figure 9. Virtualization Architecture: (a) hypervisor-based; (b) container-based [34].

The container technology allows virtualization of the larger number of instances and the disk image is smaller. Two paradigms of container - based, Docker and LXC are introduced. The performance of each virtualization technology is evaluated in terms of idle state performance, CPU, memory performance and network performance. Based on experiments, the performance of idle state for both technologies, hypervisor and container employ an optimized use of the Linux power saving system. Performance of CPU and Memory show that there is not significant change in power consumption regardless heavy CPU loads and the number of virtual runs. While the network performance shows different results through the different case studies. This is mainly due to the fact that the network packets have to be processed by more layers in a hypervisor environment compare to a container environment. An important study in this direction is presented in [32]. In this paper, the performance of different Apache Spark applications, using both Virtual Machines (VM) and Docker containers comparing for a big data enterprise cloud environment are investigated. Containers and VMs have similar resource isolation benefits but different architecture approach and resource management. Containers perform shared resource management to enable more flexible sharing of resources, while VMs perform distributed resource management where each virtual machine runs with its specific assigned set of resources. The differences between them are summarized in Table 1.

TABLE I. DIFFERENCE BETWEEN VM HYPERVISOR AND DOCKER CONTAINER

Specs	Virtual Machine	Container
Products	VMware, Xen, KVM, etc.	Dockers, rkt, etc.
Guest OS	Included	Not included
virtualization Controller	VM hypervisor	Docker engine
Resource Management	Distributed	Shared
Machine Emulation	Complete (isolated OS kernel)	Partial (shared OS kernel)
Spatial Size (Bytes)	Large	Small
Launch Time	Long	Short
Migration	May require image conversion	Easy

To compare those two technologies, a number of evaluation are conducted. One of the comparison of VM and Docker is by use of the Spark and different implementations run on those, based on different benchmarks. First, it is evaluated the total execution time for both implementations taking into consideration different number of iterations and different number of clusters. And at the end, the utilization of different system resources such as the CPU, disk and memory is investigated. Based on the results, it is concluded that the use of Spark within Docker is better for map and calculation intensive applications. Another highly cited publication in this domain is "Containers and Cloud: From LXC to Docker to Kubernetes" [59]. In this paper is presented an overcome of container-based technology from LXC to Docker to Kubernetes. The innovation of this study is movement around Kubernetes. Kubernetes is considered as the decoupling of application containers from the details of the systems on which they run. This decoupling simplifies application development also simplifies data center operations. In Kubernetes each group of containers deserves its own unique IP address that is reachable from any other group in the cluster, whether they are located on the same physical machine or not. In [60], performance evaluation of leading micro-hosting virtualization approaches are presented with specific focus on Docker and Flockport and their comparison with native platforms. Docker compared to other includes several other capabilities such as NAT, which helps to reduce some of the difficulties of Docker container utilization. On the other hands this directly impact input on throughput quality. As a results Docker can be no quicker compared to Flockport in some cases. In this way the design of one IP address for each Docker container as suggested by the Kubernetes may support quality assurance as well as throughput. These results may provide some direction of how the architecture of cloud environments should be designed.

C. Different technologies in storage virtualization.

About storage virtualization in cloud computing various technologies are proposed. Each of them has its own advantages and disadvantages. In [61], a Software Defined Storage (SDStorage) framework is proposed. SDStorage is designed to handle all the challenges facing traditional storage systems. In large data storage as data centers, that store a huge amount of data it exploits virtualization to deploy the system. The virtualization is used to enhance the utilization of the system resource, also increase the management overhead across the multiple layers produced when the system is virtualized. SDStorage manage huge data in storage systems by isolating the data control layer from the data storage layer. The control layer refers to the software component that manages and controls the storage resources, whereas the data layer refers to the underlying infrastructure of the storage assets. Fig.10 shows the architecture of the SDStorage system.

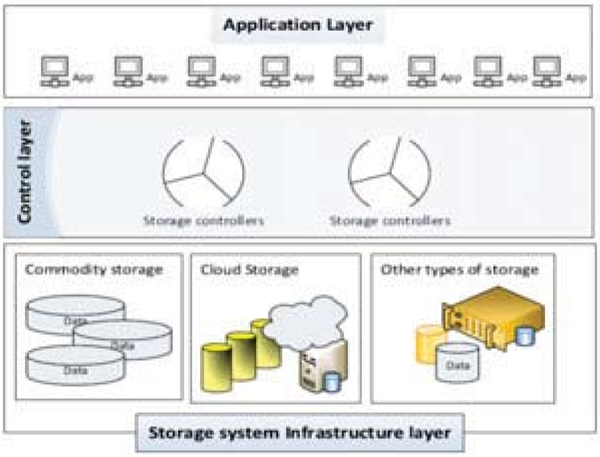

Figure 10. The Software Defined Storage system architecture [61]

The architecture is composed by three layers: infrastructure layer which combines various storage devices storing the raw data, control layer which converts different policies to different instructions inside the system and application layer that allows the end user to interact with storage devices as is shown in fig.10.Another framework in this domain is presented by German Molto [23]. He presents the impact of vertical elasticity for applications with dynamic memory requirements when running on a virtualized environment through the proposal of an architecture that dynamically adapt the memory size of the VM. Vertical elasticity has the ability to rapidly modify the capabilities of the single VMs, in terms of CPU and RAM. Fig.11 shows the architecture of this framework.

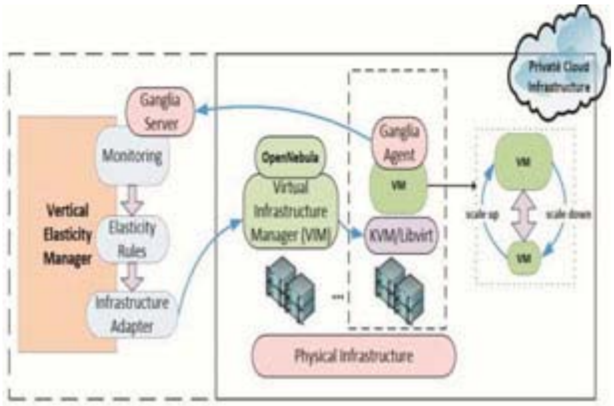

Figure 11. Proposed implementation to enable elastic memory management [23]

This architecture is supported by the KVM [62], which is basically a kernel driver inside the guest OS that behaves as a not-swappable process and can be expand or shrink its memory usage, controlled by the hypervisor on the host OS. If the memory size of a running VM expands the physical memory available in the VM is reduced, that compels the guest OS to reduce the memory of other processes where insufficient free memory is detected. Then, the memory allocated by the memory size of a running VM in the guest OS can be reclaimed by the host OS, which can be possibly used by other VMs. The VMs are deployed via OpenNebula tool [63] on a private cloud infrastructure that are based on a Virtual Machine Image with Ubuntu 12.04 LTS, which kernel supports memory KVM techniques when run as a guest OS. By use of Ganglia Monitoring System [64] the memory usage, in each VM is periodically monitored, in order to report back to the Ganglia' server the periodic state of the VM memory used. The VEM (Vertical Elasticity Manager) includes the implementation of the elasticity rules, to maintaining a user-defined percentage of free memory on the VM. The elasticity rules enable the system to decide when to scale up or scale down, delegating on the Infrastructure Adapter which interacts with the VIM (Virtual Infrastructure Manager) to effectively modify the VM memory size. The VEM can be executed as part of the private Cloud infrastructure or as an external component to the Cloud, which require only inbound connectivity from the Ganglia Agent and outbound connectivity to the VIM. The VEM can simultaneously operate with several VMs to independently scale up and down. To evaluate these techniques, a case study on a private Cloud infrastructure is executed [23]. Based on the results of some tests, in different benchmarks, it is possible to adapt the VM memory size in order to reduce the memory consumption of the VM while maintaining the level of performance of the running application. The vertical elasticity, in the shape of dynamic memory management, enables to mitigate memory over provisioning with controlled application performance penalty. Another study in this domain is presented by Li Zhang & Bing Tang [65]. In this paper, a platform called GrandStore which offers free storage space for individual users, while the free space is quite limited is presented. Through this system each user can manage all their accounts in a unique access interface integrating a plenty of free accounts to get a unified large-scale free personal cloud storage. This platform is designed and implemented based on the principle of OAuth protocol [66] and open API which manage and control the account applied by user where authentication credentials are stored in back-end database of GrandStore, which realizes easily enlarging personal free storage space and managing all storage spaces in a unique access entry. GrandStore is a scalable and open platform that adds dynamically new accounts without disturbing it. Also, it can store a plenty of accounts to obtain large space. Fig.12 shows the architecture of this platform.

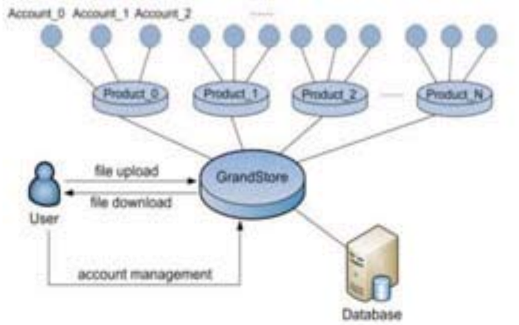

Figure 12. The architecture of GrandStore system [65].

Users firstly open a few accounts, integrate them by GrandStore and then utilize the unique storage space through it. Also, they can manage, integrate, and maintain all their own accounts through GrandStore. This system consists of four components:

- Product. It means free personal cloud storage provider, which provides SDK for developers, such as Google Drive, Dropbox and Kuaipan.
- Account. Each user can register many accounts for one product. Each account has a limited storage space.
- User. User is responsible for adding dynamically new products and new accounts and has the right of all kinds of file operations.
- Database. It is used to store account authentication credential and user’s file information.

This study promises a system with a great practical value to get a unified large scale free personal cloud storage.

Another study presented by Amol Jaikar [29], show two technologies, NAS and SAN that can be used for decentralized storage system. Fig.13 shows the Network Attached Storage (NAS) architecture. NAS is composed by a head computer and storage which is connected to a network. This storage contains one or more hard disks with less featured operating systems such as FreeNAS, FreeBSD and others. To increase the performance, multiple copies of files are stored on multiple locations.

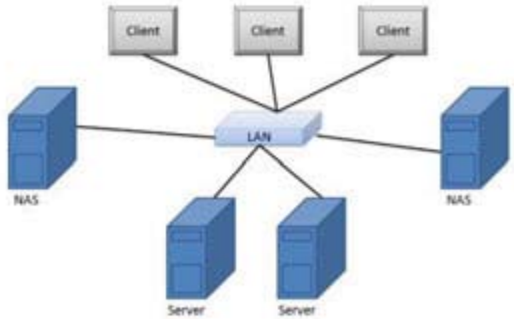

Figure 13. NAS Architecture [29].

In Fig.14 the Storage Area Network (SAN) architecture is presented. The SAN provides storage system with block-level storage. Users also can install file system on the top of SAN to provide file-level access. SAN is normally connected with fiber channel to improve the performance and speed. File sharing in SAN depends on the operating system. The data is accessed by block which is replicated to multiple locations to increase the availability.

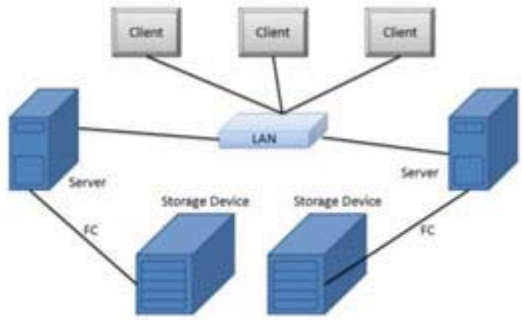

Figure 14. SAN Architecture [29].

Another framework is presented by J. Saravanan & Saravanan. P. in [67]. Direct Attached Storage (DAS) in which storage drives are directly attached to server machine. Fig.15 shows the DAS architecture. One of the benefits of DAS is high performance because of network connectivity and congestion problems that have no immediate effect. Also, DAS is easy to set up and configure as well as has a low price.

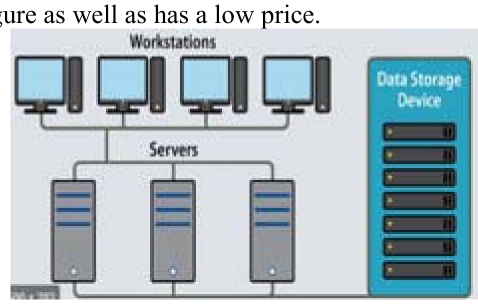

Figure 15. DAS Architecture [67].

V. DISCUSSION

This paper presents a comprehensive understanding about resources virtualization in cloud computing. It is mainly focused on the analysis of the recent technologies about this area. The analysis is focused on the metrics of each technology highlighting the advantages and disadvantages of them. Also, it is analyzed the behavior of different technologies in different applications highlighting the strengths and weaknesses for each application.

A. Different technologies in network virtualization

Network virtualization improves scalability, data transfer rates, security, versatility and reliability in a network. The proposed approach by Fabio Baroncelli [47] presents an abstraction of NaaS that enables cloud users to request on-demand connectivity without any knowledge of network details, just through Network Virtualization

Platform (NVP). Based on the approach, proposed by Raj Jain [12], through SDN the network can be done programmable, partitionable and virtualizable. The network virtualization architecture based on OpenStack [13], provides a cloud manager to control a cluster of hosting servers and to manage the virtual network infrastructures. The performance of network virtualization in this case increases significantly. By analyzing and comparing different technologies about network virtualization we can say that SDN is considered the next generation in this domain. Integration of OpenFlow and OpenStack architecture in SDN and consideration of the full-stack SDN framework that include all protocol layers in the network stack, the network performance can be improved in all aspects. Integrating these frameworks into a single framework are considered the current and future challenge about network virtualization

B. Different technologies in server and OS virtualization

Server virtualization is considered as a separation of physical server into multiple isolated virtual servers. Operating system virtualization is considered as a type of server virtualization. The paper presented by Janki Bhimani & Zhengyu Yang [31] investigates the performance of big data processing applications running on Spark and with different virtualization frameworks. They conclude that the use of Spark under Docker, in case of map and intensive calculation applications, performs better in comparison with Spark under VM. This is due that Docker provides lightweight operation, copy-on-write and intermediate storage drivers that assist Spark application to perform better. Spark used on Docker reaches speed-up of 10 times higher compared to VM. Another study is presented by Roberto Morabito [33], he compares the performance in terms of power consumption of four different virtualization technologies: KVM, Xen, Docker and LXC. Based on the experimental results, is confirmed a similar power consumption, when systems are challenged with a heavy CPU and memory workloads. As far as power consumption is concerned, both container-based solutions introduce lower power consumption. In [59], is presented the Kubernetes framework that is considered as the decoupling of application containers from the details of the systems on which they run in order to simplify application development also simplifies data center operations. By analyzing and comparing different technologies about server virtualization, based on the advantages and disadvantages of them we conclude that Docker is the more efficient technology in server virtualization, and it is considered as the future of them. The Docker improve the performance, reduce overhead, is faster than other technologies by balancing of the workload.

C. Different technologies in storage virtualization

Storage virtualization in Cloud Computing is considered as sharing of physical storage into multiple logical storage devices and such logical device looks like a single physical storage device. A technique for large-scale free personal cloud storage is presented by Li Zhang in his paper [49]. He has proposed a system called GrandStore which offers free storage space for individual users. Users apply for several accounts, and then the account authentication's credentials are stored in the GrandStore back-end database, and the system allows managing of all storage spaces in a unique access entry. Another result is presented by German Molt & Miguel Caballer in their paper [22]. They used an approach of adapting the VM memory size in order to reduce the memory consumption while maintaining the level of running application performances. The vertical elasticity, in the shape of dynamic memory management, enables to mitigate memory over provisioning with controlled application performance penalty. This adjustment is beneficial for the resource provider, since the host can take advantage of that memory reduction in order to dedicate it for other processes. In domain of storage virtualization we found out three types of data storage:

1. The NAS, described in [27] is used for file sharing, backup store memory space, and sharing device among the different machines. To increase the performance, multiple copies of files are stored on multiple locations.
2. The SAN, described into paper [28], provides storage system with block-level storage. The data is accessed by block which is replicated to multiple locations to increase the availability.
3. While in the DAS, described into paper [29] storage drives are directly attached to server machine. One of the benefits of DAS is high performance because of network connectivity and congestion problems have no immediate effect. Regarding the above mentioned, about the storage virtualization we conclude that each technology has its advantages and disadvantages and depends on their use. Integration of the advantages of NAS, SAN, DAC and SDN technologies in a heterogeneous Cloud Storage Platform remains one of the challenges of the future in this domain.

VI. CONCLUSION

In this paper, a systematic literature review of existing work to provide a comprehensive understanding of the role and significance of virtualization within the domain of Cloud Computing is presented. The scope of this paper is targeted towards: network virtualization, server virtualization, OS virtualization and storage virtualization. Network virtualization promises cloud users to request on-demand connectivity without any knowledge of the complexity and the network technology details. In this way, flexibility of management operations is increased, and can be created different virtual networks isolated from each other, allowing the share of a common physical infrastructure among multiple customers and reduce the operational costs. Storage virtualization promise sharing of physical storage into multiple storage devices which further appears to be as a single storage device. Through storage virtualization is achieved an easily data movement, data management improvement, security increase and system failure avoidance. Server virtualization promises a partition of physical servers into multiple isolated virtual servers, where each virtual server runs its own operating system and applications. While OS virtualization is considered as a part of server virtualization which promises to integrate server hardware by moving services on separate servers as well as different computer programs can be run at the same time. In this way, by divided a single server into multiple virtual servers the physical hardware cost and maintenance cost are reduced and at the same time the productivity, security and system failure avoidance, are improved. By analyzing and comparing different technologies about resource virtualization in cloud computing is recommended that integration of OpenFlow and OpenStack architecture in SDN and consideration of the full-stack SDN framework is seen as the best solution in network virtualization and is considered for the next generation of this domain. Regarding to server and operating system virtualization is recommended that the use of Docker technology is seen as the best solution and it is considered for the future use. And regarding to storage virtualization is recommended integration of advantages of NAS, SAN, DAC and SDN technologies in a heterogeneous Cloud Storage Platform and it is considered as one of the big challenges of the future in this domain. However, these technologies remain to be studied in the future by analyzing their performance in different applications and big data as well as in terms of data security.

REFERENCES

[1] Wei-Tek Tsai, Xin Sun, Janaka Balasooriya , "Service-Oriented Cloud Computing Architecture",Seventh International Conference on Information Technology, IEEE 2010.

[2] Pankesh Patel, Ajith Ranabahu, Amit Sheth," Service Level Agreement in Cloud Computing" Digital Commons Network,USA 2009.

[3] Luis M. Vaquero, Luis Rodero-Merino, Juan Caceres, Maik Lindner," A Break in the Clouds: Towards a Cloud Definition" ,ACM Volume 39, Number 1, January 2009.

[4] Sushil Bhardwaj , Leena Jain , Sandeep Jain," CLOUD COMPUTING: A STUDY OF INFRASTRUCTURE AS A SERVICE (IAAS)", 2010 International Journal of Engineering and Information Technology.

[5] Ling Qian,Zhiguro Luo,Yujian Du,Leita Guo," Cloud Computin: An Overview", IEEE & Springer 2009.

[6] Manish Godse ,Shrikant Mulik," An Approach for Selecting Software-as-a-Service (SaaS) Product ",2009 IEEE International Conference on Cloud Computing.

[7] Rakesh Kumar , Shilpi Charu," An Importance of Using Virtualization Technology in Cloud Computing", Global Journal of Computers & Technology Vol. 1, No. 2, February 25, 2015.

[8] Swarnpreet Singh and Tarun Jangwal," Cost breakdown of Public Cloud Computing and Private Cloud Computing and Security Issues", International Journal of Computer Science & Information Technology (IJCSIT) Vol 4, No 2, April 2012.

[9] Aaqib Rashid, Amit Chaturvedi, "Virtualization and its Role in Cloud Computing Environment", International Journal of Computer Sciences and Engineering Vol.7(4), Apr 2019.

[10] Jinho Hwang ,K. K. Ramakrishnan, Timothy Wood, "NetVM: High Performance and Flexible Networking using Virtualization on Commodity Platforms", IEEE Transactions on Network and Service Management (Volume: 12 , Issue: 1 , March 2015).

[11] Richard Cziva, Simon Jouet, Kyle J. S. White and Dimitrios P. Pezaros," Container-based Network Function Virtualization for Software-Defined Networks", 2015 IEEE Symposium on Computers and Communication (ISCC).

[12] Raj Jain and Subharthi Paul, "Network Virtualization and Software Defined Networking for Cloud Computing: A Survey", IEEE Communications Magazine November 2013.

[13] Franco Callegati, Walter Cerroni, Chiara Contoli ,Giuliano Santandrea, " Performance of Network Virtualization in Cloud Computing Infrastructures: The OpenStack Case", 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet).

[14] Samaresh Bera ; Sudip Misra ; Niloy Saha,"Traffic-aware Dynamic Controller Assignment in SDN" ,IEEE 2020

[15] Hiep Binh Nguyen ; Ngoc-Thanh Dinh ; Jaewook Oh ; Younghan Kim "An Openflow-based Scheme for Service Chaining's High Availability in Cloud Network",2019 International Conference on Information and Communication Technology Convergence (ICTC),IEEE.

[16] Gianni Antichi,Gábor Rétvári,"Full-stack SDN: The Next Big Challenge?" ,Proceedings of the Symposium on SDN , Pages 48–54, ACM 2020

[17] Michael Dalton, David Schultz, Jacob Adriaens, Ahsan Arefin, Anshuman Gupta, Brian Fahs, Dima Rubinstein, Enrique Cauich Zermeno, Erik Rubow, James Alexander Docauer, Jesse Alpert, Jing Ai, Jon Olson, Kevin DeCabooter, Marc de Kruijf, Nan Hua, Nathan Lewis, Nikhil Kasinadhuni, Riccardo Crepaldi, Srinivas Krishnan, Subbaiah Venkata, Yossi Richter, Uday Naik, and Amin Vahda,"Andromeda: performance, isolation, and velocity at scale in cloud network virtualization",15th USENIX Conference on Networked Systems Design and Implementation,ACM 2018

[18] Paarijaat Aditya,Istemi Ekin,|Andre Beck,Ruichuann Chen,Volker Hilt,vica Rimac,Klaus Satzkeand and Manuel stein "Will Serverless Computing Revolutionize NFV?", Proceedings of the IEEE (Volume: 107 , Issue: 4 , April 2019

[19] Deze ZengLin GuShengli PanSong Guo "Software Defined Systems" 2020 – Springer

[20] Sima Jeddi,Saeed Sharifia ,"A hybrid wavelet decomposer and GMDH-ELM ensemble model or Network function virtualization workload forecasting in cloud computing ",Elsevier 2019

[21] Wenying Zeng,Yuelong Zhao ,Kairi Ou,Wei Song ,"Research on cloud storage architecture and key technologies", 2009 ACM.

[22] German Molt,Miguel Caballer, Eloy Romero, Carlos de Alfonso ,"Elastic Memory Management of Virtualized Infrastructures for Applications with Dynamic Memory Requirements", International Conference on Computational Science, ICCS 2013,Elsevier.

[23] Chao-Tung Yang ; Shuo-Tsung Chen ; Wei-Hsun Cheng ; Yu-Wei Chan ; Endah Kristiani,"A Heterogeneous Cloud Storage Platform With Uniform Data Distribution by Software-Defined Storage Technologies",11 October 2019 IEEE

[24] Bo Peng,Haozhong Zhang,Jianguo Yao,Yaozu Dong,Yu Xu,Haibing Guan "MDev-NVMe: a NVMe storage virtualization solution with mediated pass-through",Proceedings of the 2018 USENIX Conference on Usenix Annual Technical Conference,July 2018 ,Pages 665–676,ACM

[25] Xiaohao Zhang ; Yunjie Li ; Gang Chen,"NVMe-over-RPMsg: A Virtual Storage Device Model Applied to Heterogeneous Multi-Core SoCs", 2020 10th Annual Computing and Communication Workshop and Conference (CCWC),IEEE

[26] Ibrahim Abaker Targio Hashem , Ibrar Yaqoob , Nor Badrul Anuar,Salimah Mokhtar ,Abdullah Gani, Samee Ullah Khan,"The rise of "big data" on cloud computing: Review and open research issues",Elsevier 2014

[27] Chellammal Surianarayanan,Pethuru Raj Chelliah "Storage Fundamentals and Cloud Storage",Springer 2019

[28] Amol Jaikar, Syed Asif Raza Shah, Seo-Young Noh and Sangwook Bae." Performance Analysis of NAS and SAN Storage for Scientific Workflow", 2016 International Conference on Platform Technology and Service (PlatCon) IEEE.

[29] Jung Kyu Parka,Jaeho Kim "Big data storage configuration and performance evaluation utilizing NDAS storage systems",Elsevier 2018

[30] Malgorzata Steinder , Ian Whalley , David Carrera , Ilona Gaweda,David Chess,Thomas J. Watson, “Server virtualization in autonomic management of heterogeneous workloads”, : 2007 10th IFIP/IEEE International Symposium on Integrated Network Management.

[31] Janki Bhimani, Zhengyu Yang, Miriam Leeser, and Ningfang Mi ,“Accelerating Big Data Applications Using Lightweight Virtualization Framework on Enterprise Cloud”, 2017 IEEE High Performance Extreme Computing Conference (HPEC).

[32] Zheng Li , Maria Kihl , Qinghua Lu and Jens A. Andersson, “Performance Overhead Comparison between Hypervisor and Container based Virtualization”, 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA).

[33] Roberto Morabito, “Power Consumption of Virtualization Technologies: An Empirical Investigation” 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC).

[34] MinSu Chae, HwaMin Lee & Kiyeol Lee "A performance comparison of linux containers and virtual machines using Docker and KVM",Published 20 December 2017 Springer

[35] Noor e Sahir, Muhammad Amir Shahzad, Muhammad Sohaib Aslam,Waseem Sajjad, Muhammad Imran "Optimize Elasticity in Cloud Computing using Container Based Virtualization",International Journal of Innovations in Science & Technology 2020

[36] Sijie Yang, Xiaofeng Wang, Xiaoxue Wang, Lun An & Guizhu Zhang ,"High-performance docker integration scheme based on OpenStack",Springer 2020

[37] Nikhil Marathe ; Ankita Gandhi ; Jaimeel M Shah,"Docker Swarm and Kubernetes in Cloud Computing Environment",2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI),IEEE

[38] Michail S. AlexiouEuripides G. M. Petrakis, "Elixir: An Agent for Supporting Elasticity in Docker Swarm",Springer 2020

[39] Naoki Mizusawa ,Yuya Seki,Jian Tao,Saneyasu Yamaguchi "A Study on I/O Performance in Highly Consolidated Container-Based Virtualized Environment on OverlayFS with Optimized Synchronization",2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM),IEEE

[40] Barbara Kitchenham , O. Pearl Brereton , David Budgen, Mark Turn,er,John Bailey, Stephen Linkman“Systematic literature reviews in software engineering – A systematic literature review” Elsevier 2009.

[41] Pearl Brereton, ,MarkTurner[a] Mohamed Khalil,“ Lessons from applying the systematic literature review process within the software engineering domain”,Elsevier 2007.

[42] Andreas Blenk ; Arsany Basta ; Martin Reisslein ; Wolfgang Kellerer, "Survey on Network Virtualization Hypervisors for Software Defined Networkin",IEEE Communications Surveys & Tutorials (Volume: 18 , Issue: 1 , Firstquarter 2016)

[43] Mohammad Banikazemi ; David Olshefski ; Anees Shaikh ; John Tracey ; Guohui Wang, "Meridian: An SDN Platform for Cloud Network Services", IEEE Communications Magazine (Volume: 51 , Issue: 2 , February 2013)

[44] Dmitry Drutskoy ; Eric Keller ; Jennifer Rexford, "Scalable Network Virtualization in SoftwareDefined Networks",IEEE Internet Computing (Volume: 17 , Issue: 2 , March-April 2013)

[45] Raul Munoz ; Ricard Vilalta ; Ramon Casellas ; Ricardo Martinez ; Thomas Szyrkowiec ; Achim Autenrieth, "Integrated SDN/NFV management and orchestration architecture for dynamic deployment of virtual SDN control instances for virtual tenant networks", IEEE/OSA Journal of Optical Communications and Networking (Volume: 7 , Issue: 11 , November 1 2015)

[46] Yong Li ; Min Chen, "Software-Defined Network Function Virtualization: A Survey", IEEE Access (Volume: 3 ,December 2015)

[47] Fabio Baroncelli & Barbara Martini & Piero Castoldi,” Network virtualization for cloud computing”,Springer 2010.

[48] N. M. Mosharaf Kabir Chowdhury and Raouf Boutaba,” Network Virtualization: State of the Art and Research Challenges”, IEEE Communications Magazine (Volume: 47 , Issue: 7 , July 2009).

[49] Barbara Martini, Valerio Martini, Fabio Baroncelli, Karim Torkman, and Piero Castoldi,” Application-Driven Control of Network Resources in Multiservice Optical Networks”, Journal of Optical Communications and Networking 2009.

[50] Omar SEFRAOUI,Mohammed AISSAOUI, Mohsine ELEULDJ, "OpenStack: Toward an Open-Source Solution for Cloud Computing", International Journal of Computer Applications (0975 - 8887) Volume 55 - No. 03, October 2012

[51] OpenStack project website. Available: http://www.openstack.org,(accessed 12.01.2020)

[52] Antonio Corradi, Mario Fanelli, Luca Foschini, "VM consolidation: A real case based on OpenStack Cloud", Future Generation Computer Systems,Volume 32, March 2014, Pages 118-127 ,Elsevier

[53] Borja Sotomayor, Ruben S. Montero, Ignacio M. Llorente, and Ian Foster, "An Open Source Solution for Virtual Infrastructure Management in Private and Hybrid Clouds", IEEE Internet Computing ,Special Issue on Cloud Computing 2009

[54] Xiaolong Wen ; Genqiang Gu ; Qingchun Li ; Yun Gao ; Xuejie Zhang, "Comparison of open-source cloud management platforms: OpenStack and OpenNebula", 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery,IEEE

[55] Anish Babu S. ; Hareesh M.J. ; John Paul Martin ; Sijo Cherian Rajagiri ; Yedhu Sastri, "System Performance evaluation of Para virtualization, Container virtualization and Full virtualization using Xen, OpenVZ and XenServer", 2014 Fourth International Conference on Advances in Computing and Communications,IEEE

[56] Stephen Soltesz, Herbert Potzl ,Marc E. Fiuczynski, Andy Bavier, and Larry Peterson, "Container-based Operating System Virtualization: A Scalable, High-performance Alternative to Hypervisors", ACM 2007

[57] Carl Boettiger , "An introduction to Docker for reproducible research", ACM SIGOPS Operating Systems ,January 2015

[58] James Turnbull, "The Docker Book: Containerization is the new virtualization", Google book 2014

[59] David Bernstein, "Containers and Cloud: From LXC to Docker to Kubernetes", IEEE Cloud Computing (Volume: 1 , Issue: 3 , Sept. 2014)

[60] Zhanibek Kozhirbayeva ,Richard O.Sinnott, "A performance comparison of container-based technologies for the Cloud",Future Generation Computer Systems Volume 68, March 2017, Pages 175-182,Elsevier

[61] MahmoudAl-Ayyouba, Ala’Darabseha ,Elhadj Benkhelifab, Mladen Voukc, Andy Rindo, "Software defined cloud: Survey, system and evaluation",Future Generation Computer Systems Volume 58, May 2016, Pages 56-74,Elsevier

[62] KVM,KVM,CPU,Hotplug.Available:http://www.linux-kvm.org/page/CPUHotPlug,(accessed 14.01.2020)

[63] Dejan Milojičić Hewlett-Packard Laboratories; Ignacio M. Llorente ; Ruben S. Montero,”OpenNebula: A Cloud Management Tool” , IEEE Internet Computing (Volume: 15 , Issue: 2 , March-April 2011).

[64] Matthew L. Massie , Brent N. Chun , David E. Culler,” The ganglia distributed monitoring system: design, implementation, and experience” ,ELSEVIER Parallel Computing Volume 30, Issue 7, July 2004, Pages 817-840.

[65] Li Zhang, Bing Tang, “GrandStore: Towards Large-Scale Free Personal Cloud Storage”, 2017 International Conference on Computer Network, Electronic and Automation (ICCNEA) & IEEE.

[66] Barry Leiba, “OAuth Web Authorization Protocol”, IEEE Internet Computing (Volume: 16 , Issue: 1 , Jan.-Feb. 2012).

[67] J.Saravanan & Saravanan,” The Storage Process in Virtualization for Cloud Computing”, JAC : A Journal Of Composition Theory, Volume XII Issue XII December 2019.