Virtualization Mechanisms and Tools: A Comprehensive Survey

Bakhan Tofiq Ahmed
Department of Information Technology, Technical College of Informatics
Sulaimani Polytechnic University, Sulaimani, Kurdistan Region, Iraq
bakhan.tofiq.a@spu.edu.iq

***Abstract* — Today, virtualization is a critical and widely utilized technology. It is deployed for distributing the abilities of physical computers by separating the resources among operating systems. Furthermore, the virtual machinery can also offer manageable atmospheres for up-to-date computing systems. Virtualization has several advantages such as manageability, flexibility, separation, downtime reduction, cost efficiency, and scalability, etc. In this study, a comprehensive survey about the most potential mechanisms and tools of virtualization has been discussed. The renowned mechanisms of virtualization have been reviewed such as Full Virtualization, Para Virtualization, Desktop Virtualization, and Network Virtualization, etc. Additionally, some of the common virtualization tools like VMware, OpenVZ, Xen, Qemu, Docker, and Kernel-based Virtual Machine 'KVM' had been briefly explained. This study reviewed many up to date works that have been done by other researchers were used various visualization mechanisms and tools. Therefore, a comparison has been done among various tools of the virtualization in terms of some basic factors such as virtualization method, usability, and accessibility. This study will be useful for newer researchers who interests to work in this area.**

Keywords - Virtual Machine Monitor; VMware; Para Virtualization; Full Virtualization; Network Virtualization.

I. INTRODUCTION

The technology is enhanced rapidly day by day. With the expansion of the computer system, the virtual machine has been originated to be the major research topic by researchers. By utilizing the virtual machinery, the computer scheme can combine all forms of data assets or resources, software assets, and hardware assets. Also, virtualization can make these assets to afford facility for diverse tasks. This technology also distributed hardware and software management and afforded valuable features containing performance separation, server unification, and live migration. Additionally, virtualization can also offer transportable environments for up-to-date computer schemes. Consequently, the virtualization machinery has been utilized widely [1]. It is a technology in which had been extensively deployed for sharing the abilities of physical computers by dividing the assets or resources between operating systems 'OS'. The expedition began in 1964 when IBM originated a scheme that formed the Virtual Machines 'VM' term. The original scheme was named 'CP-CMS' scheme and later advanced to the Virtual Machine 'VM'. Generally, the control program 'CP' module was an OS capable to utilize a computing engine to act like several replicas of the engine. Each replica of the virtual machine had organized by its operating system. The scheme that IBM established was low- level virtualization that required extra enhancement to give the full benefit of the produced scheme. Then, in the mid of 1970s, Goldberg and Popek proposed a new scheme that presented a clear conception of the theories and advantages related to the virtualization. Later on, the concentration was diminished until 1999 once the VMware enterprise offered its invention which was the VMware tool for the version x86-32 architecture [2]. The virtualization conception developed from a software-based method to a hardware-based method by memory, processor, and devices virtualization more powerfully. The main advantages of virtualization were to centralize and integrate IT assets. Virtual Machines 'VMs' were utilized for the virtualization of assets such as main memory, CPUs, hard drive, and the input/output 'I/O' devices. In general, virtualization procedures can be classified into these: Full-Virtualization 'FV', Para-Virtualization 'PV', Desktop-Virtualization 'DV', Server-Virtualization 'SV', Storage-Virtualization 'SV', Native-Virtualization 'NV', Operating-System-Level Virtualization, Data-Centers-Virtualization 'DCV', and Application-Virtualization 'AV'. The software layer that provided the virtualization was named Virtual-Machine-Monitor 'VMM' or Hypervisor. This software was interleaved among the hardware and the virtual machines 'VMs'. It permitted the simultaneous execution of several 'VMs', separate them, and schedule them between the existing assets. The Hypervisor could run on both hardware named Bare Metal or Type-I virtualization or on the top of OS host which is named Hosted, or Type-II virtualization. Both 'VMs' and 'VMMs' had been established to provide the best energy resolutions to the virtualization difficulties. Many tools were established by vendors to provide virtualization methods such as OpenVZ, Xen, VSphere, and VMware [3].

The major target of this study is to give a comprehensive review of the virtualization mechanisms and its major tools which are widely utilized by many organizations nowadays.

The arrangement of the other sections will be as follows: in section 2 a systematic review about other researcher's work has been presented. The most common mechanisms of virtualization like Full-Virtualization, Para-Virtualization, and Desktop-Virtualization have been discussed in section 3. Section 4 is about the widely utilized virtualization tools like VMware, OpenVZ, and Xen. A comparison has been done between the virtualization tools is drawn in section 5. Ultimately, section 6 is the paper’s conclusion.

The architecture of virtualization is shown in Figure 1.

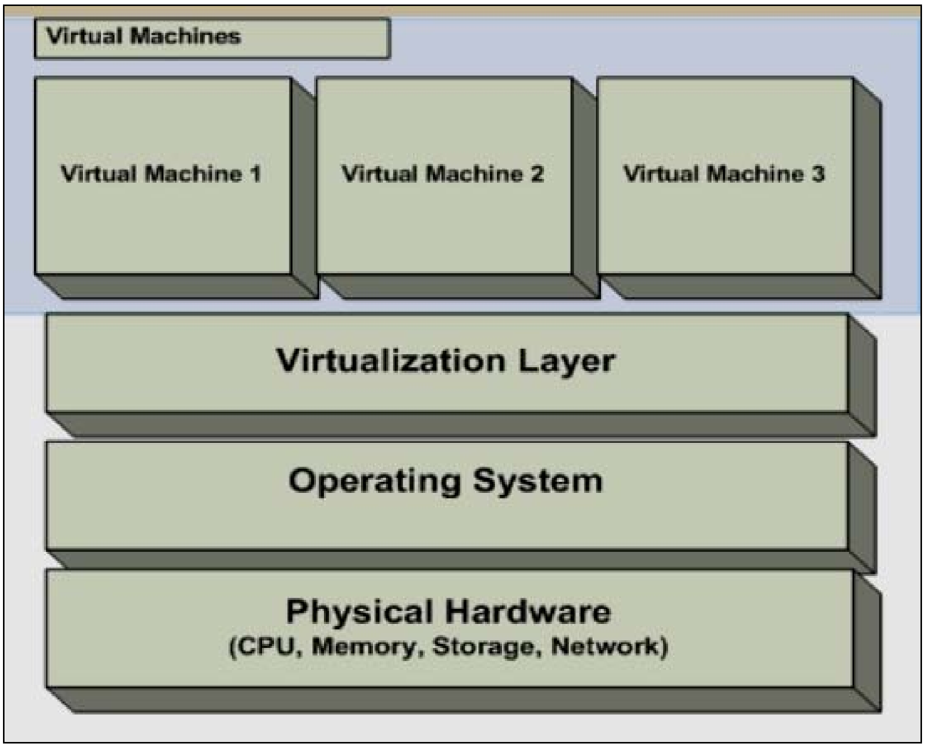

Figure 1. The Architecture of the Virtualization [1].

II. RELATED WORKS

In recent times, many researchers conducted a significant investigation about the concept of visualization, its mechanisms, and tools. In this section, some of the relevant up-to-date studies in this field have been reviewed for this decade.

Lai et al. [4], in this study a Novel Desktop Virtualization ‘NDV’ framework had been proposed to offer secluded virtualization facilities for end-users. In this proposed frame, server assets were incorporated into an influential computing ability to utilize the assets more efficiently. Furthermore, all applications were running on Virtual Machines ‘VMs’ that were mapped from physical assets via Virtual Machine Monitor ‘VMM’ or Hypervisor. Moreover, a Novel Scheduling Algorithm ‘NSA’ was recommended to balance the workload between the entire Virtual Machines ‘VMs’. There were several studies about the Desktop Virtualization which pointed out that it was prominently advanced the Quality of Service ‘QoS’ of end-users and powerfully assigned assets between all clients.

Huang et al. [5], a review had been done about the latest virtualization mechanisms, their source management techniques, and the major attacks that attacked these virtualization mechanisms. Later, the necessities of network Test-Bed and the security effects had been demonstrated whenever the virtualization mechanisms had been presented into a network Test-Bed. Also, the attacks that affected Virtualized Network Test-Bed were revealed. The outcome had shown that the up-to-date virtualization mechanisms were vulnerable to security assaults. Whenever the virtualization mechanisms had been adapted to the network Test-Bed, an invader can get the benefit of the network linking functionality to attack the Test-Bed that could have more endangered over the separation and scalability of the Test-Bed. The improvement of security was extremely important in the network Test-Bed to diminish threats.

Gilles et al. [6], this paper proposed the initial real-time Hypervisor for Multi-Core PC that utilized both Para-Virtualization and Full-Virtualization techniques without depending on any other hardware for virtualization. Proteus guarantees Virtual Machine division and was categorized by a Bare-Metal method and a synchronization technique that did not depend on extra hardware support. The evaluated result had shown that there was a little overhead on both memory and execution time.

Masood et al. [7], this study presented an inclusive review of the foremost methods of virtualization within its significant tools. There were many various mechanisms of the virtualization, for instance, Full-Virtualization 'FV', Native-Virtualization 'NV', Para- Virtualization 'PV', and Operating System-Level Virtualization 'OSLV' revealed in detail. A comparison had been done among the various tools of virtualization like Xen, VM-Ware, Open-VZ, Qemu, VSphere, and Virtual Box 'VB' in this study to aid the newer researcher to have a full understanding about the main aspects of different virtualization tools.

Weiss et al. [8], a new enhanced method of storage virtualization was proposed that relied on the Block Level Address Mapping 'BLAM'. This newly proposed method had a group of interfaces which was assisted the applications to control the storage address mapping. Also, it allowed the hosted system to operate this Address Map with a group of 3 modest actions (Copy, Move, and Delete). Also, a prototype implementation that named Project ANViL had been recommended and demonstrated its effectiveness with a group of case studies. The implementation of these case studies was an ordinary extension of some techniques in Log-Structured data stores like 'FTLs'. As a result, the proposed method had depicted that the interfaces had sufficient flexibility to offer a countless compact of added service to applications. This study discussed that the virtualization was an essential portion of new systems within the beginning of flash it had become significant to study storage virtualization to decrement the overload.

Goel et al. [9], a methodical evaluation was done to detect how virtualization had been implemented efficiently in this decade. Mainly, three fields of IT utilized virtualization techniques and had an efficient impact on these fields particularly in time reduction, cost efficiency, manageability, and flexibility. These areas were Network- virtualization, Storage-Virtualization, and Server- Virtualization. The virtualization conception had the potential impact on improving CPU utilization, high obtainability of the system, reduced power consumption, capability on running platforms like (Windows, Solaris, Linux, and Netware) and applications simultaneously on the similar server, efficiently managing the overhead, flexibility, centralize administrative tasks. In contrast, the limitation of virtualization was the security services and sufficient utility backup.

Plauth et al. [10], in this study an inclusive comparison had been revealed among Container and Unikernels by concentrating on the specific metric that was appropriate to cloud computing. Both Containers and Unikernels were signifying mutual loads in microservice-based applications 'MSBA'. Moreover, the performance of the application had assessed by utilizing HTTP servers and a Key-Value Store 'KVS'. The implementation of microservice-based applications assisted in measuring extra features like startup time, the size of the image, the latency of the network, and memory footprint. The main factor that the comparison based on was throughput. As a consequence, the throughput of Unikernels was better than the Containers. Also, proved that Containers were not preferable for network performance, but the network performance could be obtained by Unikernels. Although the Containers could gain the smallest startup times, whereas the Unikernels were obtained the smallest image sizes and much tinier startup times particularly in circumstances where the image had to be transmitted to the hosted computer. Moreover, the Containers had gained popularity rather than Unikernels. Also, the Unikernels did not surpass a Virtualized Linux Case. Also, a brief comparison had been done among Net-BSD and Rump-Run.

Khedher et al. [11], this thesis presented the novel conceptions of Network Function Virtualization 'NFV' and Software Defined Network 'SDN' that were two significant Network-Virtualization functions to provide a resolution to the client's demands, and increment the performance of the network. Furthermore, a new method is applied to execute optimizations that diminish the number of migration for delivering a task. Then, the virtualization was combined with 'CDNs'. This work presented virtualization in the framework of duplication of Video Content Servers 'VCS'. Therefore, for proposing this work virtualization architecture needed to be designed based on many algorithms that can diminish costs and advance the performance of the system. A precise optimization resolution that relied on the linear programming in which depended on some factors such as the service quality of video clients, the capability of the network links, and the parameters interrelated to Network Function Virtualization 'NFV' and Software Defined Network 'SDN' like the size of the disk, the size of the main memory or RAM, and the capacity of the processor or CPU. Along with these two critical Network-Virtualization functions a new empirical method relied on graph theory and more accurately on Gomory-Hu or 'GH' algorithm for scalability. The major notion behind this resolution was to decrease the network links number. The proposed model relied on open source tools like Open-Stack, Open-Day-Light, and Open-Virtual-Switch. They were combined into an actual network operator.

Steinberg and Ismael [12], a procedure of memory vision had been implemented by utilizing a virtualization layer to execute the conception of virtualization on the location of each network node. The layer of virtualization might contain two modes the first one called a user-mode which had Hyper-Routes and the second one named a kernel-mode which had a Micro-Hypervisor that collaborated to virtualize a Guest-OS kernel with a Virtual Machine 'VM' of each node. The Micro-Hypervisor could further collaborate with the Hyper-Routes as a Guest-Monitor of the virtualization layer to apply one or more memory visions of the Virtual Machine 'VM'. Generally, a memory-vision was a hardware asset (i.e. a group of nested page tables) utilized as a container for many guest processes of the Guest-OS kernel.

Taj et al. [13], in this study, the anomaly-based intrusion detection system 'ABIDS' is proposed to detect and identify illegal activities that are not part of legitimate network traffic in the cloud computing environment by using virtual machines (VMs). The proposed method observes, detects, and identifies the traffic of VMs. It also scans the abnormal behavior in the network traffic and reply restrains the abnormal packets detected in the network traffic.

Alam et al. [14], in this research article, a systematic and comprehensive review of the techniques of virtualization designed for the Internet of Things 'IoT' are presented. The survey is classified into software-defined networks 'SDN' and network function virtualization 'NFV' designed for IoT networks. Also, this work explained the architectural, security, and management solutions for IoT. Moreover, the article highlighted many short-term and long-term challenges and open problems correlated with the implementation of SDN IoT. Therefore, this systematic survey revealed that the SDN and NFV separately cannot accomplish the programmability necessities of modern networks. However, a better result can be attained, if SDN and NFV are combined.

III. THE MECHANISMS OF THE VIRTUALIZATION

The machinery that is widely utilized to partition the abilities of physical computers by separating the assets between Operating Systems 'OSs' are referred to as Virtualization [15]. The layer that managed and organized hardware assets among various Virtual Machines 'VMs' named Hypervisor or Virtual-Machine-Monitor 'VMM'. The operating System 'OS' which is worked with a 'VM' named Guest-OS. This technology attracted the vision of the enterprises because of having several valuable benefits like reducing the cost due to server merging and fewer hardware necessities, Well hardware utilization because several operating systems 'OSs' could be hosted at the same time, Down-Time reduction because of fast recovery, providing swift backup and recovery that saves more time, high-level Error Tolerance which offered superior performance, improved security method, and consistency, and declining overhead because it could be easily administrated [2]. Indeed, there are several virtualization mechanisms but in this section, the most significant and widely utilized mechanisms of virtualization have been briefly discussed as listed below.

A. *Full Virtualization 'FV'*

In this form, the hardware interface delivered to Hypervisor or 'VMM' is nearly similar to the one afforded by the hardware's physical. This was meant that for offering virtualization there was no need to alter the operating systems 'OSs' and applications if they were well-matched with the original hardware. This type also could be further categorized into another sub-forms like Bare-Metal- Virtualization 'BMV' and Hosted-Virtualization 'HV' [16]. Figure 2 shows the full virtualization snapshot.

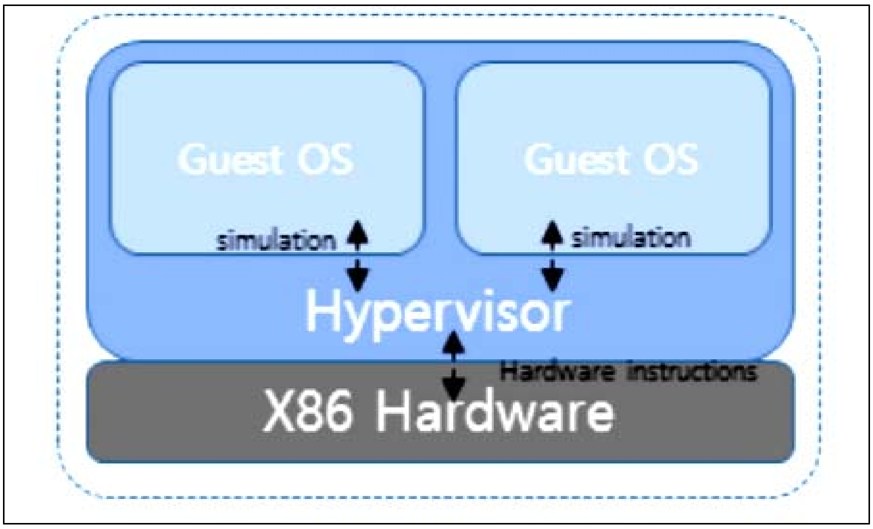

Figure 2. The Snapshot of Full Virtualization [17].

B. Para Virtualization 'PV'

This form is a contrast to the previous type which was Full-virtualization 'FV' because of in the Para-virtualization 'PV' the running Guest-OS was needed to alter. In this method, the Guest-OS engine knew that they were running in a situation that was virtualized. The major benefit of utilizing this machinery was to diminish the virtualization over-heads and offer superior performance. Xen was an example of Para-Virtualization [18]. The snapshot of the para-virtualization is shown in Figure 3.

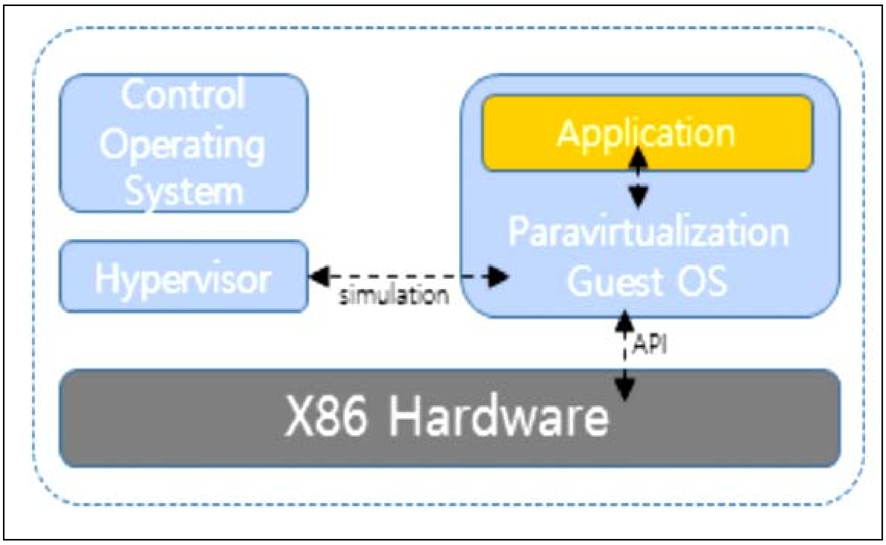

Figure 3. The Snapshot of the Para Virtualization [17].

C. Operating System-Level virtualization 'OSLV'

This kind was known as that the kernel of an OS permitted several separate User-Space instances. These instances track over the top of OS hosted system, worked with a group of libraries that interact with applications, and allowing them to run on a machine devoted to its utilize. This form of virtualization mechanism also referred to as a Container-Based -Virtualization 'CBV' [19]. Figure 4 presents the snapshot of the OSLV.

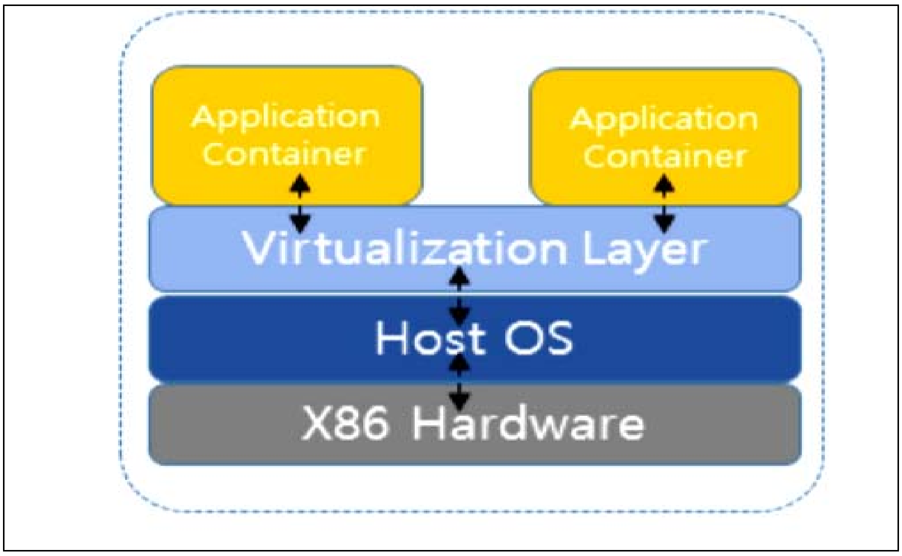

Figure 4. The Snapshot of the OSLV [17].

D. Application Virtualization 'AV'

In this type of virtualization, an end-user is allowed to run an application of the server locally with the assist of native assets without requiring the installation of the complete application on the computer scheme. It also provided a separate virtualization environment to each end-user that acted as a layer among the host and the OS. The most well-known instance of this form was Java Virtual Machine 'JVM' that acted as an intermediate among the OS and the Java- Application-Code 'JAC' [15].

E. Desktop Virtualization 'DV'

It was the conception of splitting the logical desktop from the physical appliance. It was enumerated as hardware virtualization. Virtual Desktop Infrastructure 'VDI' was the main sub-type of this form. Despite the interrelating with a host computer straight through a peripheral computer like keyboard, mouse, and monitor, the end-user interrelated with the host computer utilizing another desktop or a mobile device with the help of the network connection like the Internet. Moreover, the host computer became a server that was able to host several Virtual Machines concurrently for more than one end-users [20].

F. Network Virtualization 'NV'

It had been utilized to unite both hardware 'HW' and software 'SW' assets into a Virtual Network as a distinct group of assets. It assisted in obtaining superior infrastructure utilization in terms of reutilizing a logical or physical asset for several other network assets like hosts, virtual machines 'VMs', and routers, etc. It also assisted in diminishing costs by distributing network assets [21] [14]. In Figure 5 the main difference between virtualized and Non-virtualized systems has been shown.

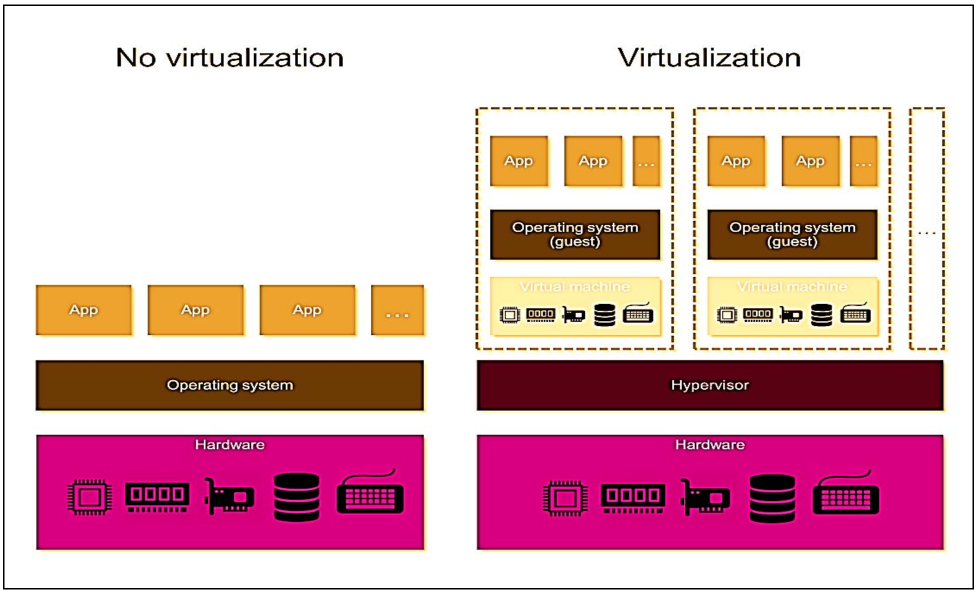

Figure 5. The Virtualized vs. Non-virtualized System [9].

IV. THE TOOLS OF THE VIRTUALIZATION

In this section, the most critical and well-known virtualizations' tools have been illustrated in detail for an instant, VMware, OpenVZ, and Xen, etc.

A. VMware

It was a virtual machine 'VM' that assisted in executing unmodified OS on the Host or User-Level application. The OS that utilized with VMware may get stopped, reinstalled, restarted, or crashed without having any influence on the application that runs on the Hosted CPU. VMware provided the distribution of Guest-OS from the actual Host-OS. As a consequence, if the Guest-OS failed later the physical hardware or the hosted computer did not suffer from the failure. VMware was utilized to create standard illusion hardware on the inner side of the Virtual Machine 'VM'. Hence, the VMware was utilized to execute numerous unmodified OS simultaneously on the distinct hardware engine by executing the OS in the Virtual Machine of a particular OS. Despite that, the code running on the hardware like a simulator, Virtual Machine executed the code straightly on the physical hardware without any software that interprets the code [22].

B. *Xen*

It was the most common virtualization open-source tool that supported both Full-Virtualization 'FV' and Para-Virtualization 'PV'. Xen was an extremely famed virtualization resolution, initially established at the Cambridge University. It was the single Bare-Metal solution that was obtainable as an open-source. It contained several elements that cooperated to supply the virtualization atmosphere comprising Xen Hypervisor 'XH', Domain-0-Guest shortened to Dom-0, and Domain-U-Guest shortened to Dom-U that could be either 'PV'-Guest or 'FV'-Guest. The Xen Hypervisor 'XH' was the layer that resided straightly on the hardware underneath any OS. It was responsible for CPU scheduling and memory segregating of the different 'VMs'. It represented the administration of Domain-U-Guest 'Dom-U' to the Domain-0-Guest 'Dom-0' [23].

C. *Qemu*

This virtualization tool was utilized to execute the virtualization in the OSs such as Linux and Windows. It was counted as the renowned open-source emulator that offered swift emulation with the assist of dynamic translation. It had several valuable commands for managing the Virtual Machine 'VM'. Qemu was the major open-source tool for various hardware architectures. Indeed, It was an example of Native- Virtualization 'NV' [7].

D. *OpenVZ*

It was also an open-source virtualization tool that relied on the control group conceptions. OpenVZ provided Container-Based-Virtualization 'CBV' for the Linux platform. It allowed several distributed execution that named Virtual Environments 'VEs' or Containers with a distinct operating system kernel. It also provided superior performance and scalability when compared with the other virtualization tools [24].

E. *Docker*

Docker is open-source. It is relied on using containers to automatically distribute Linux application. All the necessities like codes, runtime system tools, and system libraries are included in the Docker containers. Docker utilized Linux containers (LXC) library till version 0.9, but after this version, Docker utilizes a lib container for virtualization capabilities provided by a kernel of Linux. It uses to implement an isolated container via a high-level application program interface (API). The operating system (OS) is not required in Docker. The same Linux kernel utilizes by a Docker container but is performed by isolating the user space from the host OS. Docker is only available and compatible with Linux [17] [25].

F. *Kernel-Based Virtual Machine (KVM)*

A KVM is also open-source and is required central processing unit (CPU) technology for (Intel VT or AMD-V). It utilizes the full virtualization 'FV' for Linux x86 and including the extensions of virtualization, the KVM's kernel component is included in Linux, but the KVM's userspace components are included in a quick emulator (QEMU). However, for some devices KVM also supports the para-virtualization 'PV' mechanism. By using KVM end user can turn Linux into a **Hypervisor** that can run multiple and isolated virtual environments called guests. The main limitation of KVM is that it cannot execute emulation. Instead of that, it reveals the KVM interface and it sets up the virtual machine address space and feeds the simulated input/output via QEMU [17]. Figure 6 depicts the KVM's architecture.

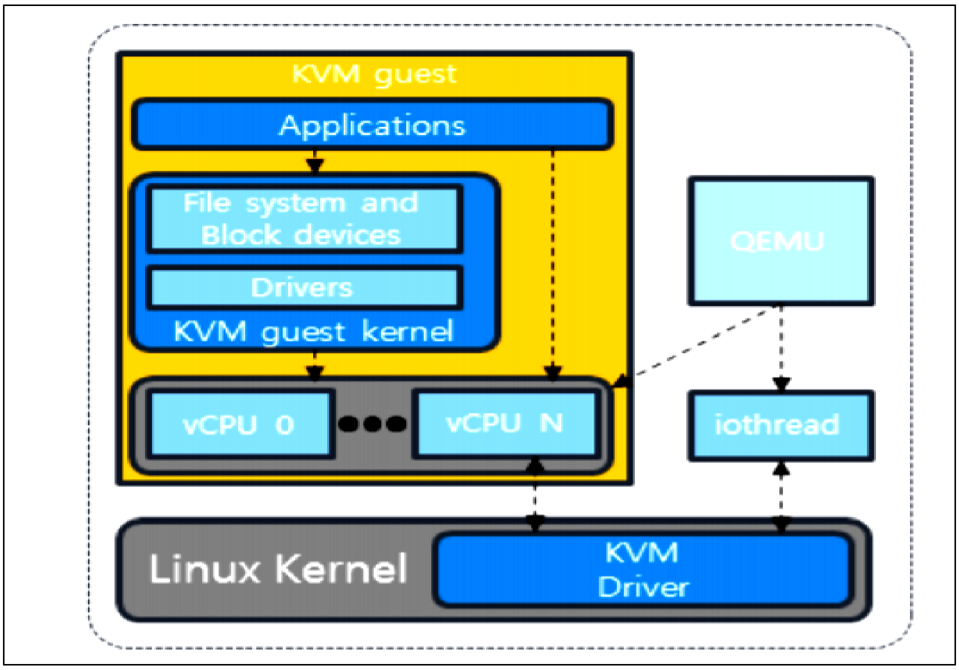

Figure 6. The Architecture of the KVM [17].

The used range of virtualization tools has been revealed in Figure 7. As a consequence, VMware was the most used virtualization tool among others.

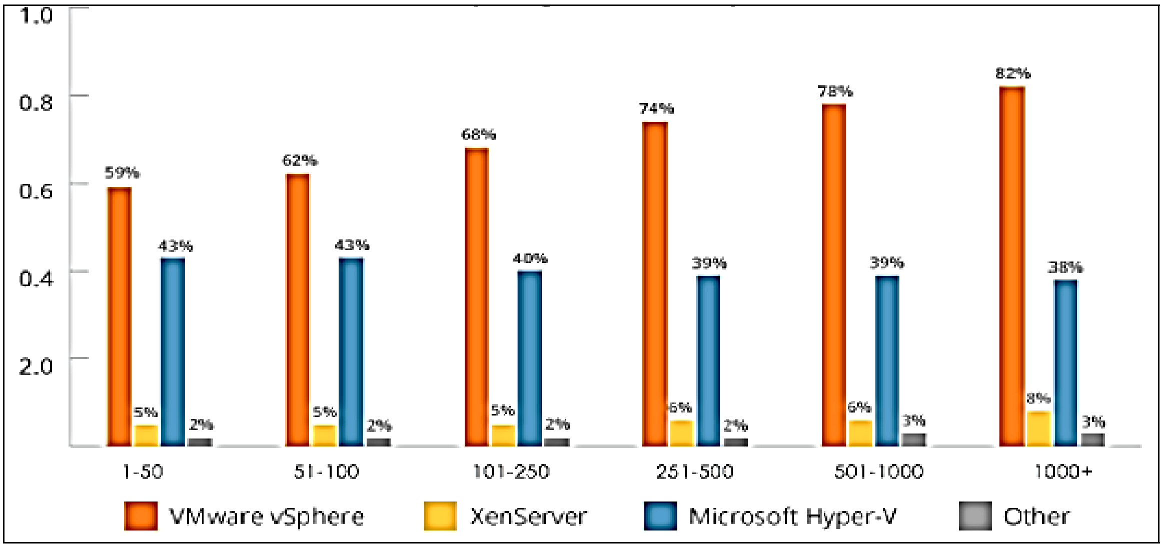

Figure 7. The Range of the Used Virtualization Tools [22].

V. RESULTS AND DISCUSSIONS

In this study, a systematic literature survey has been done about virtualization technology with its mechanisms and tools. As a result, the comparison between various tools of the virtualization in terms of some factors has been done as depicted in Table1. Each virtualization tool supported a different virtualization mechanism, and also they are different in their accessibility and usability. It can be noted that from Figure 7 the VMware was the most widely used tool by the end-user because it is the best one for handling virtual infrastructure.

Table 1. A Comparison among the Various Renowned Virtualization Tools

Ref.	Virtualization Tool	Accessibility	Usability	Virtualization Method
[22]	VMware	Commercial	The best one for handling virtual infrastructure.	Full Virtualization 'FV'
[23]	Xen	Open Source	It is used for the migration purpose of the Virtual Machine 'VM'.	Para Virtualization 'PV'
[7]	Qemu	Open Source	For a diverse range of hardware architectures 'HW-A'. It also can be utilized as an emulator.	Native Virtualization 'NV'
[24]	OpenVZ	Open Source	The main usability was to partition the resources effectively.	Operating System Level Virtualization 'OSLV'
[17] [25]	Docker	Open Source	It uses to implement an isolated container via a high-level application program interface (API).	Operating System-Level Virtualization 'OSLV'
[17]	KVM	Open Source	Allow end-users to turn Linux into a **Hypervisor** that can run multiple and isolated virtual environments called guests.	Full Virtualization and Para Virtualization in some cases

VI. Conclusion

Virtualization is the best evolving machinery of these years in which the expansion is taking place exponentially. Many enterprises are currently aware of the benefits of implementing virtualization. Moreover, the organizations are applying virtualization resolutions to diminish the downtime and make effective power utilization. Virtualization has become a leading means of rendering recently. Nowadays, it obtains a lot of popularity specifically in the IT industry due to having several benefits such as downtime reduction, superior hardware utilization, great obtainability, and best manageability, etc. However, this emerging technology has some drawbacks also such as security problems and the difficulty of the infrastructure that may be affected its enhancement. The major objective of this paper particularly is to concentrate on giving a systematic survey about the major tools and well-known mechanisms of virtualization. Along with reviewing many other researchers' work that worked on different virtualization mechanisms. Few of the renowned virtualization mechanisms with tools are briefly explained which are widely utilized by the end-user. As a result, a comparison has been done among the commonly utilized tools which are VMware, OpenVZ, Qemu, Xen, Docker, and KVM in terms of accessibility, usability, and virtualization method. Therefore, VMware is the mostly virtualization tool used by the end-user because it is the best one for handling virtual infrastructure.

References

[1] Y. Li, W. Li, and C. Jiang, "A survey of virtual machine system: Current technology and future trends," in 2010 Third International Symposium on Electronic Commerce and Security, 2010, pp. 332-336, DOI 10.1109/ISECS.2010.80: IEEE.

[2] F. Rodríguez-Haro et al., "A summary of virtualization techniques," The 2012 Iberoamerican Conference on Electronics Engineering and Computer Science, vol. 3, pp. 267-272, 2012. https://doi.org/10.1016/j.protcy.2012.03.029

[3] S.-H. Li, D. C. Yen, S.-C. Chen, P. S. Chen, W.-H. Lu, and C.-C. Cho, "Effects of virtualization on information security," Computer standards Interfaces, vol. 42, pp. 1-8, 2015. http://dx.doi.org/10.1016/j.csi.2015.03.001

[4] G. Lai, H. Song, and X. Lin, "A service based lightweight desktop virtualization system," in 2010 International Conference on Service Sciences, 2010, pp. 277-282, https://doi.org/10.1109/ICSS.2010.44: IEEE.

[5] Y.-L. Huang, B. Chen, M.-W. Shih, and C.-Y. Lai, "Security impacts of virtualization on a network testbed," in 2012 IEEE Sixth International Conference on Software Security and Reliability, 2012, pp. 71-77, DOI 10.1109/SERE.2012.17: IEEE.

[6] K. Gilles, S. Groesbrink, D. Baldin, and T. Kerstan, "Proteus hypervisor: Full virtualization and paravirtualization for multi-core embedded systems," in International Embedded Systems Symposium, 2013, pp. 293-305, https://link.springer.com/content/pdf/10.1007/978-3-642-38853-8_27.pdf: Springer.

[7] A. Masood, M. Sharif, M. Yasmin, and M. J. N. J. o. S. Raza, "Virtualization tools and techniques: Survey," Nepal Journal of Science Technology, vol. 15, no. 2, pp. 141-150, 2014.

[8] Z. Weiss, S. Subramanian, S. Sundararaman, N. Talagala, A. Arpaci-Dusseau, and R. Arpaci-Dusseau, "ANViL: advanced virtualization for modern non-volatile memory devices," in 13th {USENIX} Conference on File and Storage Technologies ({FAST} 15), 2015, pp. 111-118, https://www.usenix.org/system/files/conference/fast15/fast15-paper-weiss.pdf.

[9] N. Goel, A. Gupta, and S. N. Singh, "A study report on virtualization technique," in 2016 International Conference on Computing, Communication and Automation (ICCCA), 2016, pp. 1250-1255, https://doi.org/10.1109/CCAA.2016.7813908: IEEE.

[10] M. Plauth, L. Feinbube, and A. Polze, "A performance survey of lightweight virtualization techniques," in European Conference on Service-Oriented and Cloud Computing, 2017, pp. 34-48, https://hal.inria.fr/hal-01677609/file/449571_1_En_3_Chapter.pdf: Springer.

[11] H. I. Khedher, "Optimization and virtualization techniques adapted to networking," Institut National des Télécommunications, https://tel.archives-ouvertes.fr/tel-01822403/document, 2018.

[12] U. Steinberg and O. A. Ismael, "Technique for implementing memory views using a layered virtualization architecture," ed. https://patentimages.storage.googleapis.com/a6/28/1f/3b2d8c2769202f/US10191861.pdf: US Patent App. 15/257,704, 2019.

[13] M. S. Taj, S. I. Ullah, A. Salam, and W. U. Khan, "Enhancing Anomaly Based Intrusion Detection Techniques for Virtualization in Cloud Computing Using Machine Learning," International Journal of Computer Science Information Security, vol. 18, no. 5, 2020. https://d1wqtxts1xzle7.cloudfront.net/63506204/10_Paper_01052033_IJCSIS_Camera_Ready_pp68-7820200602-78128-twbz7n.pdf?1591121630=&response-content-disposition=inline%3B+filename%3DEnhancing_Anomaly_Based_Intrusion_Detect.pdf&Expires=1596841133&Signature=cFyMnGyh6W8DnNWcgyw~tbI7hKqJL2uuOvB-Bn2HycP0Ri1J5GVBhrxN~EPIogOG-TdZFxu83kekdYvBGqJNWtgoUycNFly4erFbeTCQX9zxuKLzLfLAKkNR1LLfUURFD~YvdPn-ToTsZpHpqAuzkVO90yVJJ~-pmeOINzzYBTMiekDc38a-8id6Bb1iewUtlp3pNfXEy40aRD88p68xjmCEtbFOvkwExVc7xfasdv0zv8gb8vceBSKP2ND9LMVi7QWU84P2AQjp5fpzcwQFWxZzyckuzPl9F4MwlQP99J0Jbv1SB8NR~CVdz6evjKOGDWMUW6CiKItSZeF4v68CQg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[14] I. Alam et al., "A Survey of Network Virtualization Techniques for Internet of Things Using SDN and NFV," ACM Computing Surveys, vol. 53, no. 2, pp. 1-40, 2020. https://doi.org/10.1145/3379444

[15] M. H. Shirvani, A. M. Rahmani, and A. Sahafi, "A survey study on virtual machine migration and server consolidation techniques in DVFS-enabled cloud datacenter: Taxonomy and challenges," Journal of King Saud University-Computer Information Sciences, vol. 32, no. 3, pp. 267-286, 2020. https://doi.org/10.1016/j.jksuci.2018.07.001

[16] L. Bouali, E. Abd-Elrahman, H. Afifi, S. Bouzefrane, and M. Daoui, "Virtualization Techniques: Challenges and Opportunities," in International Conference on Mobile, Secure, and Programmable Networking, 2016, pp. 49-62, https://www.researchgate.net/deref/http%3A%2F%2Fdx.doi.org%2F10.1007%2F978-3-319-50463-6_5?_sg%5B0%5D=45mwzFfnUGH9BJ7EPtrqDONHk8G0OA1UGBzTXFpw0RtWQTPIlX2Libs_6GCMJ6LLSqxmnhngUYID6CAX2MLpzjvuKg.g2jsoh58qRvT9UTRgwtFMlGS6MvpiTyFisIWpz-DEAwbYkn7W7AdAYxkezc-YVo94WZwbYFqNyl8XgJyYz5sOg: Springer.

[17] M. Chae, H. Lee, and K. Lee, "A performance comparison of linux containers and virtual machines using Docker and KVM," Cluster Computing, vol. 22, no. 1, pp. 1765-1775, 2019. https://doi.org/10.1007/s10586-017-1511-2

[18] G. Aljabari and E. Eren, "Virtualization of wireless LAN infrastructures," in Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems, 2011, vol. 2, pp. 837-841, http://scholar.ppu.edu/bitstream/handle/123456789/98/23.pdf?sequence=1: IEEE.

[19] P. Kedia, R. Nagpal, and T. P. Singh, "A survey on virtualization service providers, security issues, tools and future trends," International Journal of Computer Applications, vol. 69, no. 24, pp. 36-42, 2013. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.403.7219&rep=rep1&type=pdf

[20] S. Kumari and S. Payami, "Virtualization & Scheduling in Real Time OS: A Survey," International Journal of Trend in Scientific Research and Development, vol. 1, no. 4, pp. 434-439, 2017. https://d1wqtxts1xzle7.cloudfront.net/57954178/64_Virtualization___Scheduling_in_Real_Time_OS_A_Survey.pdf?1544346163=&response-content-disposition=inline%3B+filename%3DVirtualization_and_Scheduling_in_Real_Ti.pdf&Expires=1596924635&Signature=fYHmlZbc9T0159rtXvDmT3kuXtFVGJ8SC8z7WamWb3kD5PKh0Hpz9DIJoR-yiIVA7CqvTx-M1uZeiaiTOmLmqlJnnbTdllW7xnmo4vt5w90RvaQ8LiwUn2WEOwKSPR5IJrNtJWBuOZ6wv0hwWfzLV8Nl~9az8OsHNaZoE5KUUMoICnBEsBS1WJhxgzbUnvrHgHgMVv-R44PtJj3VQemvJc8qwEQDO7hpX9icOzB5HZ2w4PYJLwIct6U7RGwF3KPkg~oP0YcH8KLYDHMuEq4mM4dncUUsbTUjG~pnTunWMlGsIPIuz3WQ3~Ok7YoGtDUs3Vq1upCuedF0hBGBYZslZA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[21] T. Gupta, J. Ganatra, and K. Samdani, "A Survey of Emerging Network Virtualization Frameworks and Cloud Computing," in 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2018, pp. 193-198, https://doi.org/10.1109/CONFLUENCE.2018.8442995: IEEE.

[22] J. P. Walters et al., "GPU passthrough performance: A comparison of KVM, Xen, VMWare ESXi, and LXC for CUDA and OpenCL applications," in 2014 IEEE 7th international conference on cloud computing, 2014, pp. 636-643, https://www.isi.edu/sites/default/files/users/jwalters/papers/Cloud_2014.pdf: IEEE.

[23] H. Fayyad-Kazan, L. Perneel, and M. Timmerman, "Full and para-virtualization with Xen: a performance comparison," Journal of Emerging Trends in Computing Information Sciences, vol. 4, no. 9, pp. 719-727, 2013. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.400.645&rep=rep1&type=pdf

[24] Y. Zheng and D. M. Nicol, "A virtual time system for openvz-based network emulations," in Proceedings of the 2011 IEEE Workshop on Principles of Advanced and Distributed Simulation, 2011, pp. 1-10, https://pdfs.semanticscholar.org/4bd2/7813b85c06665d83c478ae71d3ba0e05631a.pdf#page=110: IEEE Computer Society.

[25] A. K. Yadav and M. Garg, "Docker containers versus virtual machine-based virtualization," in Emerging Technologies in Data Mining and Information Securityhttps://doi.org/10.1007/978-981-13-1501-5_12: Springer, 2019, pp. 141-150.