International Burch University
Faculty of Engineering and Natural Sciences
Department of Information Technologies

CEN 261 - Computer Organization

Virtual Machines

Sarajevo, 2022 **Muhamed Hamzić 20002331**

Table of Contents

1. Abstract

This research project will introduce the user to the inner workings of a virtual machine, by firstly discussing the inception of virtual machines; the context which called for the need of virtual machines, namely the double state architecture. As a result of this architecture a number of problems were introduced which hindered the process of computing development. The virtual machine concept was developed in order to combat these issues. Furthermore, the research project will provide a brief overview of the first virtual machines and their broad and useful utilization techniques in the past as well as the reason for the decline of their usage in the 1980s and the early 1990s. A spark which led to the reemergence of virtual machines will be discussed and the concrete solution which fueled this fire of change. In addition a number of today's usage of virtual machines will be named. Lastly the topic of importance of virtual machines for an individual will be stated. At the end the conclusion will summarize the covered topics and provide the reader with future trends in the space of virtual machine solutions.

2. Inception of Virtual Machines

In the early 1960s two major changes were made when it came to computer architecture. The first one being the emergence of I/O processors. Which in turn resulted that computer systems became multiprocessors where nonidentical processors could have access to the common main memory module. The second change was multiprogramming, this technique was introduced in order to enable better resource utilization and overall performance of the system. As a result multiple computational processors were sharing a common pool of computing resources, this sharing was implemented with the use of time multiplexing. [1]

Even though these improvements resulted in higher overall performance of a computing system, it introduced serious potential problems in respect to the system's integrity. Meaning that an incorrect I/O program could alter the areas of main memory it should not have access to, likewise an incorrect computational process faced the same problem of unprivileged memory rewrites.[2] An approach was chosen to combat this issue, namely the double state architecture; the following section will delve deeper into this concept.

2.1. Double state architecture

This architecture works by effectively creating two different classes of system operations: the one which is privileged and the other which is not. The privileged class was a relatively small piece of software which can be assumed to be mostly correct, this class was responsible for dealing with all functionality which could cause undesirable inferences with multiple non-privileged processes. Operations which were in the privileged domain included: channel program initiation, modification of address mapping mechanisms, direct monitoring of external interrupts, etc. This approach showed to be effective if the privileged software is compact, stable and requires minor changes and improvements over a long period of time and is maintained by skilled professional programmers. Although this architecture proved to be very efficient and improved overall performance of multiple I/O processors and multiprogramming problems it was faced with problems which were the reason for the emergence of virtual machine monitors and virtual machines in general.[3] The dual state architecture is illustrated in Figure 1.

[1] Paraphrased from [1]
[2] Paraphrased from [1]
[3] Paraphrased from [1]

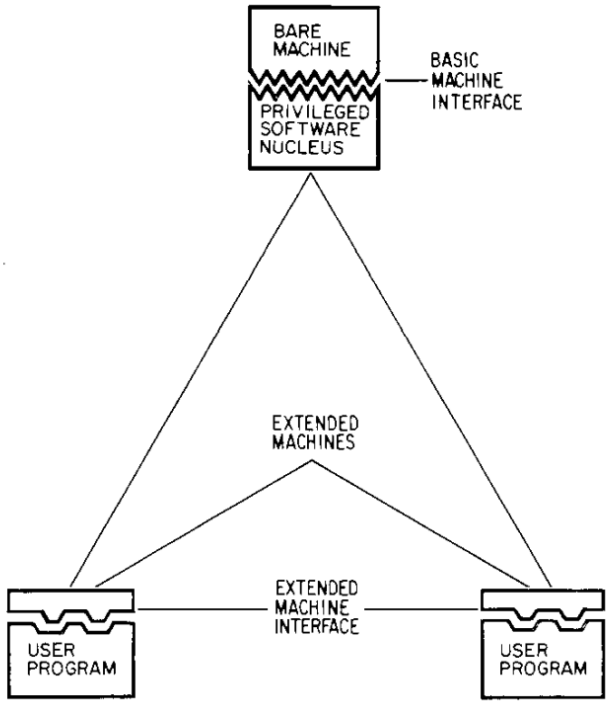

Figure 1. Illustrating the dual state architecture - Source: [2]

2.2. Problems Virtual Machines strived to solve

The shortcomings of the dual state architecture were, but not limited to the following two issues: transportability and the development of the software nucleus.

When it comes to transportability issues, since user programs were written for the extended machine interface which was specific for the software nucleus which was in turn written for the target machine, program transportability was virtually impossible without modifications to the user program. A user who wished to run a program from a different extended machine had to convert the program to his installation of the extended machine or to let the desired program run on the "foreign" machine.
On the other hand the development of the software nucleus was challenging, since only one software nucleus could be run on the physical machine, this fact resulted in a hindered process of software nucleus testing and improvement. As a result programmers who were developing the nucleus had to work odd hours in order to have the machine available to them for installing and testing new software nucleus versions.[4]

[4] Paraphrased from [1]

3. The Virtual Machine Concept

The concept of virtual machines were necessary to solve the problems previously stated and many others. The main cog that made the virtual machine organization possible, was the virtual machine monitor (VMM, used interchangeably with - hypervisor). VMM transforms the single machine interface into the illusion of many, each of the interfaces that the VMM provides is a replica of the original computer system. This enables running of multiple operating systems on top of one machine. Figure 2. illustrates the organization of the virtual machine concept.

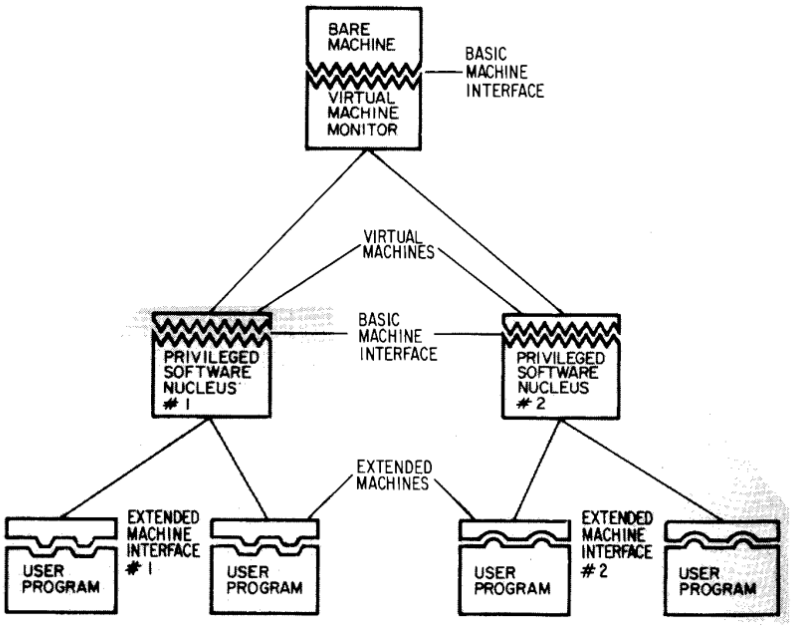

Figure 2. Organization of the virtual machines concept, Source: [2]

Now, the VMM can take multiprogramming and I/O processors into its advantage, enabling an efficient resource utilization of multiple software nucleuses. Nevertheless, the VMM still had to deal with privileged instructions that multiple software nucleus demanded. In order to provide isolation of parallelly run operating systems, instructions that did not interfere with other virtual machines or were not privileged were allowed to run directly to the bare machine without the intervention of the VMM, all other instructions were trapped in the VMM for further handling.

The trapped instructions were simulated in software using the virtual mode bit to determine the appropriate action in each case. The virtual mode bit indicated the state in which the machine would be if the instruction was to be run on the bare machine.[5]

[5] Paraphrased from [2]

3.1. Early Virtual Machines

Nelson published an internal research report at IBM that outlined plans for an experimental machine based on the IBM 7044, called the M44. The work introduced an early form of virtual memory, they called this memory a particular mapped virtual memory for a particular process - virtual machine. 44X stood for the virtual machines/virtual memory which was running on top of the M44 host machine. Even though M44/44X lacked all of the features we now consider integral to virtual machines, Denning reflected that M44/44X was central to significant theoretical advances in memory research around paging, segmentation, and virtual memory in the 1960s.[6]

The Virtual Machine Facility/370 or VM/370 was a convenient name for three different operating systems: the Control Program (CP), the Conversational Monitor System (CMS), and the Remote Spooling and Communications Subsystem (RSCS). CP was an operating system that used the computer to simulate multiple copies of the machine itself, the copies here referred to our today's view of virtual machines. Supporting and enabling the interactive use of a computer machine by one person was the responsibility of the CMS operating system. While the RSCS was the operating system responsible for handling and providing communication between virtual machines.[7]

3.2. Utilization of early virtual machines

In this section we will introduce the reader to some of the uses of virtual machines in early computing. Virtual machines proved to be a valuable concept which has a wide range of usages.

When it came to installation management, virtual machines allowed for scheduling flexibility by permitting privileged software development, test and diagnostic functions, and multiple operating system execution parallely. Virtual machines were also used to alleviate the new release "trauma" by permitting the new, not yet tested operating system to be run and tested before the final release. This also allowed for a milder transition between OS versions, since the users had time to adapt and test their programs on a new version before it was released. Figure 3. illustrates how the percent of people using the new release changes through time.[8]

[6] Paraphrased from [5]
[7] Paraphrased from [4]
[8] Example taken from [2]

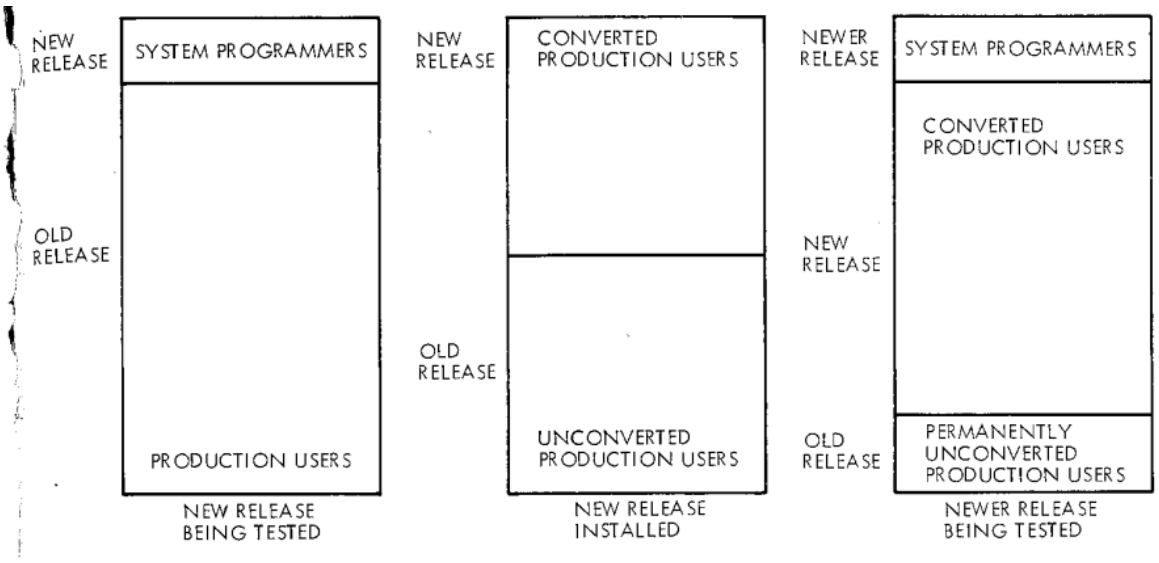

Figure 3. Virtual Machines supporting multiple releases, taken from [2]

It can be noticed in the leftmost section of the figure that system programmers could develop and test new releases of the software nucleus, while the production users were able to continue to work uninterrupted on the old release. The middle section of the figure shows the way virtual machines relieved the "trauma" of switching to a newer release. And the rightmost section gives insight to another useful feature, if it was impossible for a user program to be rewritten for the new release, or if it was not economically viable since the program was not used that often a small portion of system resources could be reserved for these programs.

Another useful feature that was enabled by virtual machines was retrofitting[9] an old operating system with new features through virtual devices. An older system, which might not support new external devices out of the box, would use virtualization of devices to bridge the gap between the old system and the new device. The operating system would use a VMM to virtualize the new, not supported real device into a virtual device which the system and the operating system support. Doing it this way usually required modification of the VMM but it proved as a much less complex task then to make the old machine support the new device natively. Figure 4. illustrates with a diagram how this virtualization process took place. [10]

[9] The process of adding new technology to older systems
[10] Example taken from [2]

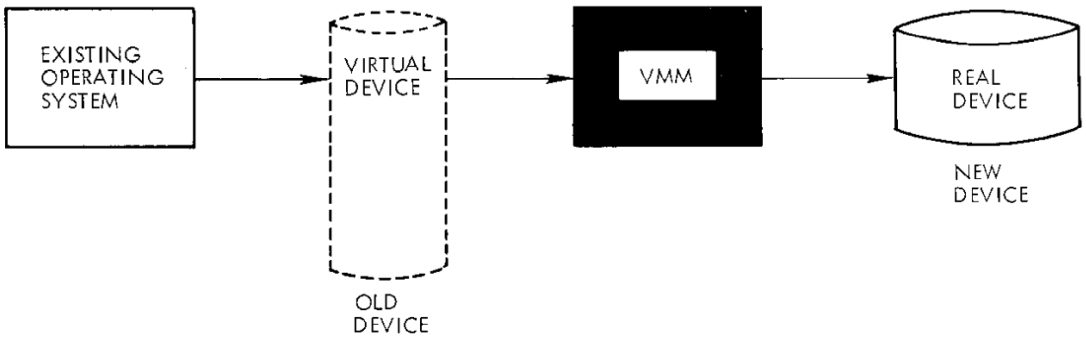

Figure 4. Virtual Devices, taken from [2]

There are a number of early uses of virtual machines left, but we will briefly go over some of them. Virtual machines were also used in development of computer network software, where VMs were used to simulate multiple computers and communication between them. VMs also enabled powerful debugging and performance monitoring tools, this greatly improved the development of software and operating systems. Virtualization also opened new doors when it came to education, enabling students to construct sample student operating systems.[11]

3.3. Decline of utilization

As smaller hardware was introduced to the market in the 1970s from companies such as DEC, Honeywell, HP, Intel, and Xerox it did not include hardware support for virtual memory and the ability to trap all sensitive instructions. This resulted in a challenging implementation of strong isolation using virtual machine techniques. As personal computers were penetrating the market, it decreased interests in early forms of virtual machines.[12] However Creasy "recognized the potential for virtual machines to serve "the future's network of personal computers"[13]

Virtual Machines continued to exist in the 1980s and the 1990s but only with a minimum of activity and interest. "IBM's line of VM products, descended from VM/370 continued to have a small but loyal following."[14] Perhaps the best example of the decay of the idea of virtual machines, was the programming language Java which re-purposed the term virtual machine - to refer to an abstraction layer of a language runtime and not to the software replication of real hardware architecture.[15]

[11] Examples taken from [2]
[12] Paraphrased from [5]
[13] Cited from [5]
[14] Cited from [5]
[15] Paraphrased from [5]

4. Modern Virtual Machines

The late 1990s started bringing the technology of virtual machines back on track, a starting point for the renaissance of virtual machines was a research project at Stanford University named Disco. This research project done in 1997, found that VMs and VMMs enabled multiple extended operating systems to run efficiently on large-scale shared memory multiprocessors.[16]
Conclusions from the research are perfectly contained in the following citation : *"...we show that many of the problems of traditional virtual machines are no longer significant. Our experiments show that the overheads imposed by the virtualization are modest both in terms of processing time and memory footprint…".* [17]
A year later, researchers behind Disco founded VMware to continue their work, the first workstation product was released in the 1999 named Workstation 1.0 allowing users to run multiple operating systems on a single computer[18]

The Denali VMM solution stood out from the rest of VMM solutions by introducing para-virtualization techniques.[19]"*Para-virtualization entails selectively modifying the virtual architecture to enhance scalability, performance, and simplicity."*[20]

Furthermore, the Xen project from the University of Cambridge also used paravirtualization techniques and specially developed guest operating systems systems, but emphasized that the guest applications should run unmodified. Additionally this project introduced the notion of individually billing tenants sharing physical machines, which was thought to be a radical idea at the time. This directly led to the starting of Amazon's Elastic Compute Cloud (EC2) a few years down the line.[21]

It is important to note that previously mentioned virtualization solutions all faced a problem when put up against x86 architectures, since this architecture did not have built in support for virtualization. Due to the increased demand for virtual machines and the challenges of implementing them on x86 hardware, Intel and AMD added hardware support for virtualization.[22]

[16] Paraphrased from [6]
[17] Cited from [6]
[18] Paraphrased from [7]
[19] Paraphrased from [8]
[20] Cited from [8]
[21] Paraphrased from [5]
[22] Paraphrased from [5]

4.1. Current usage of Virtual Machines

There exists a number of usages of virtual machines in today's times, but we will name some of them, starting with video game preservation. Preserving video games is more than a mere individual's will to feel nostalgic, it is about containing and saving a piece of history, and the ideas and expressions of the developers at the time. Fortunately, virtual machines exist and play a crucial role in running old operating systems which in turn can run conserved games. As we are currently amazed by the art of the old masters, it can be theorized that in the not so distant future, new generations will be awed by the glory of current and previous games that came.

Cloud computing (CC) provides computing resources on a pay-per-use basis by orchestrating the current resource demand with the availability of large data centers. This is enabled by exploiting the benefits of virtualization. User requests for the computing resources of provisioned virtual machines vary by CPU speeds, memory size, different storage capacity, etc. It is the task of the cloud computing provider to map requests on physical machines. This introduces the problem of assigning the proper amount of resources and avoiding under- and over-provisioning. Over-provisioning of resources refers to provisioning more resources than needed and under-provisioning to provisioning less resources than needed. A number of algorithms were developed to tackle this problem by introducing a range of measurements to make the CC business as profitable as possible while keeping its customers satisfied.[23] Cloud computing continues to be one of the biggest users of the virtual machine concept.

In order to help their mapping process, cloud computing vendors provide a number of differently configured virtual machines for their users to choose from. We will briefly look into the offerings of Google’s Cloud. Their repertoire of virtual machines is divided into four main categories: general purpose, compute-optimized, memory-optimized and accelerator-optimized.

The first category of general purpose VMs balance price and performance and are suitable for most workloads including databases, development and testing environments, web applications, and mobile gaming.

The second - compute-optimized VMs provide the highest performance per core on Compute Engine and are optimized for compute-intensive workloads. Another optimized category is optimized in terms of memory, these VMs offer up to 12 TB for a single instance, and can offer up to 416 vCPU. And the last category the accelerator-optimized VMs are based on the NVIDIA Ampere A100 Tensor Core GPU, these VMs are suitable for demanding workloads like machine learning and training.[24]

[23] Paraphrased from [3]
[24] Paraphrased from [12]

An example of services which can be built with the power of cloud computing is gaming on demand. During previous years a number of cloud services providing powerful virtual machines on demand for game enthusiasts has increased. It became a lucrative business with big players competing in the market: Nvidia's GeForce Now, Google's Stadia, Amazon's Luna and Microsoft's Xbox Cloud Gaming service. [25] Cloud gaming seems like a solid short and long-term solution to the ever increasing crisis of chip deficiency in the world.

5. Discussion

In this section we will deal with a concrete example of utilizing virtual machines in education in network penetration and security. Setting up a safe hacking environment consists of choosing the operating system from which we will do the penetration, in our case it was Kali Linux since it comes preloaded with a number of features we will use, as well as choosing a vulnerable by design web server to attack, we choose MrRobot - inspired by the popular Netflix show. After we have downloaded the needed ISO images and imported them in Virtual Box we started isolating these virtual machines. This is important because we do not want to accidentally harm other systems on the internet by a simple mistype of an IP address. Additionally, because we will be running a vulnerable by design web server we do not want to expose this system to potential black-hat hackers.

Setting up an internal network was simple as Virtual Box provides us with all the necessary options in their GUI. We attached our Kali and MrRobot virtual machines on the same internal network named "SafeNet". Because our virtual machines are now on their own private network they haven't been assigned their IP addresses. This problem can be solved by installing the VirtualBox Extension pack which has a set of handy functions, including a function that enables us to set up a virtual DHCP on our internal network "SafeNet". Finally, our virtual machines can communicate with each other and not be reached by devices outside their network. This is shown in the following Figure 5. [26]

[25] Paraphrased from [9]
[26] Paraphrased from [10]

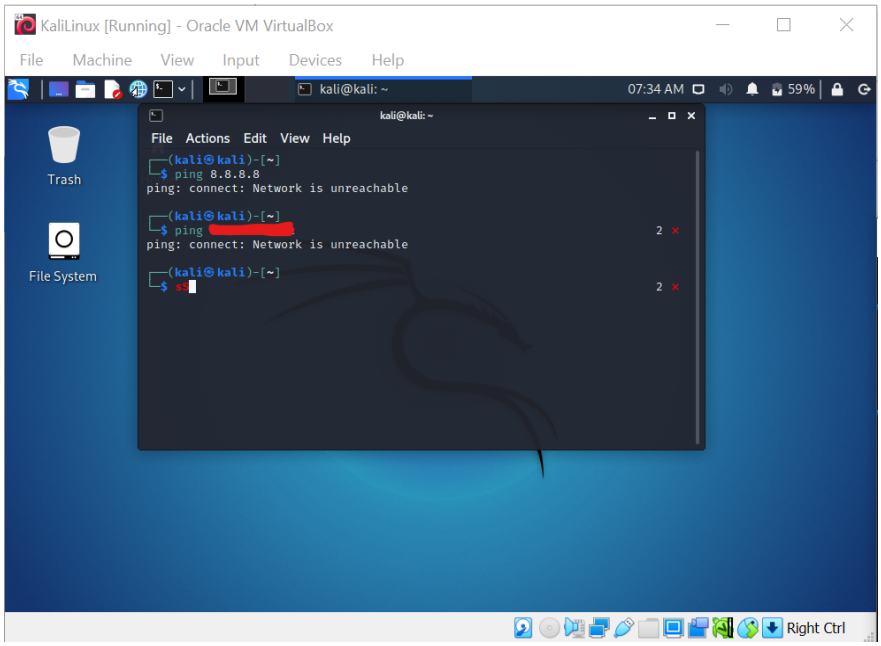

Figure 5. Screen capture of a Kali VM - Source [10]

Here we can see upon trying to reach Google's DNS server with a ping command we are unable to. It is also not possible for us to reach the host system running the virtual machine. The following Figure 6. shows an unsuccessful attempt to ping the Kali Linux virtual machine.

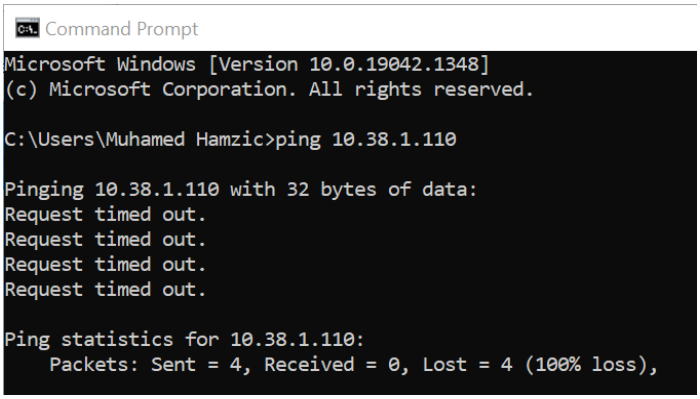

Figure 6. Screen capture of the host system - Source [10]

This adds to the argument that VMM implementations have matured greatly over the years and provide users with an easy to navigate interface, for a wide range of functions. Furthermore, it enables new opportunities for all people who aspire to learn more about cybersecurity, by enabling them to create their own hacking laboratories in which they can safely and most importantly legally learn and improve their knowledge.

In previous sections we talked about a number of complex usages of virtual machines and a day-to-day computer user might find them too ambiguous and not useful for them. Fortunately, there are a number of ways a traditional computer user can use a simple virtual machine setup and gain valuable benefits. For example, running software which does not come from trusted sources under a virtual machine for testing, protects the user's host system from malicious code. Another useful feature derived from virtualization is increased privacy, by rotating and changing virtual machine available resources one can protect itself from cross site cookies and make their digital fingerprint harder to create.

What makes virtual machines so great is that they can solve specific problems, which usually do not come to our mind. One example is reported on the VirtualBox official forums where a user going by the handle Trippp asks the community for help in solving a peculiar problem. His mother who is 89, a professional caterer has her cookbook program which contains 45 years of her recipes saved on a computer running Windows XP. Recently she got herself a new Windows 10 HP computer, which unfortunately does not support the cookbook program written for Windows XP. The Virtual Box community was swift in answering the question and providing guidance to Tripp in order to get the system up and running and enable his mom to continue using her cookbook program.

6. Conclusion

The concept of virtual machines has been present and used since the 1960s with some fluctuations in the frequency and scale of usage. Nevertheless, after the introduction of specialized commercial software and native hardware support from processor manufacturers, virtual machines have been booming with the usage in business as well as with individuals. Virtual machines continue to be one of the leading solutions when it comes to cloud computing and dynamically allocating needed resources. Virtual machines are here to stay, and this research project strives to provide an entry level knowledge and a stepping stone for further research into the concept of virtual machines.

7. References

[1] - The evolution of virtual machines architecture by J.P. Buzen and U.O. Gagliardi, Honeywell Information Systems Inc. Billerica, Massachusetts and Harvard University Cambridge Massachusetts

[2] - Survey of Virtual Machine Research by Robert P. Goldberg, Honeywell Information Systems and Harvard University.

[3] - Mapping Virtual Machines onto Physical Machines in Cloud Computing : A Survey, by Ilia Pietri and Rizos Sakellariou, University of Manchester

[4] - IBM Journal of Research and Development Volume 25. No. 5, September 1981 R.J. Creasy

[5] - The ideal Versus the Real: Revisiting the History of Virtual Machines and Containers, Alison Randal, University of Cambridge

[6] - Disco: Running Commodity Operating Systems on Scalable Multiprocessors, Edouard Bugnion, Scott Devine, and Mendel Rosenblum, Computer Systems Laboratory, Stanford University

[7] - Official VMWare website - https://www.vmware.com/timeline.html

[8] - Denali: Lightweight Virtual Machines for Distributed and Networked Applications, Andrew Whitaker, Marianne Shaw, and Steven D. Gribble, The University of Washington

[9] - TechCrunch - Nvidia expands its GeForce Now game streaming ecosystem | TechCrunch

[10] - Computer Networks Research Project - "Introduction to network penetration" - Muhamed Hamzić, International Burch University

[11] - Virtual Box End User Forum - virtualbox.org • View topic - [Solved] Hiding VirtualBox

[12] - Google Cloud - Choose the right Google Compute Engine machine type for you