International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

Virtual Machine Performance on AMD E2-7110 Quad Core Machines

Wari Denyefa Cletus
Department of Computer Science,
School of Applied Sciences
Federal Polytechnic Ekowe
Bayelsa State, Nigeria

Seiyaboh Zidiegha
Institute of Research in Applicable Computing,
Research Graduate School
University of Bedfordshire LU1 3JU
United Kingdom

***Abstract*— Virtualization technology is a fundamental research area in the field of computing. Researches show that this innovation has become popular in both industry and academia due to various features designed to ease system management and administration. Its emergence is seen as a solution to the underutilization of servers and have offered data centres to speedily move towards it use. However, there is an increase demand for better measurements so that critical business applications running on this virtualized IT infrastructure can function without downtime. This paper Investigate three different virtualization software tools (VMware, VirtualBox and QEMU) on AMD E2-7110 QuadCore Machine and uses an existing benchmark to measure them. The performance results indicate that VMware showed better performance compared to other virtualization software tools. Graphs were created which helped to highlight the differences in the results and provide meaningful analysis.**

Keywords—AMD; virtual machine; virtualization software; data centre; server

I. INTRODUCTION

Virtualization technology is one of the fundamental research areas in the field of computing. This technology was introduced and developed during the 1960s by IBM Corporation. In that period, IBM created a few virtual machines which can run on a single physical mainframe. Virtualization technology is the creation of a virtual rendition (not genuine) of a specific thing, which can be a computer hard-ware platform, storage devices, operating systems and so on. In the general sense, it is viewed as a strategy of logically apportioning system resources provided by a mainframe computer into different applications. This technology provides services same as the real machine services which are at variance with the traditional system of one computer to one operating system (OS). The only difference here is that one computer hardware can run dissimilar types of operating systems (OS).

A virtual illustration of the underlying hardware is achieved employing virtualization tools also known as virtualization software tools (VSTs). The underlying hardware contains dynamic memory, static memory and other resources which are similar to the real hardware machine. The guest OS can be installed utilizing these underlying features. It performs the same function just as the physical machine and the guest OS also like the real OS.

Typically, virtualization alludes to the creation of a virtual machine that can virtualize the entirety of the hardware resource, which may include processors, memory, storage, and system network. With virtualization, physical hardware resource can be shared by at least one virtual machines (Ali and Meghanathan, 2011).

There are three viewpoints to satisfy virtualization. Firstly, the virtualization ought to give an equivalent environment to run a program compared to a traditional system. If by chance the program shows an alternative behaviour under the virtualization, it may not be qualified as a virtualized environment. Secondly, the virtualization needs to give secure control of virtualized resources. Having full control of resources is imperative to secure information and assets on each virtual environment from any dangers or performance obstruction in sharing physical resources. Thirdly, virtualization frequently expects performance degradation because of the extra virtualization tasks, however good performance ought to be accomplished with software or hardware support in taking care of special instructions. With these prerequisites, proficient virtualization is ensured. (Hyungro, L., 2010).

It is important to know what hypervisor is all about to appreciate the functionality of a virtual machine. The virtual machine is controlled and managed by a hypervisor also called a virtual machine monitor (VMM). It is a program that creates a virtual environment, manages the guest OS and their storage areas, memory and other resources which are being shared among the operating systems. It also manages incoming interrupt requests (Chen and Noble 2001). Resource like processors must have virtualization capabilities to support virtual replication of hardware because not all processors can. Processors which support this type of hardware are typically found in mid to high rnd microprocessors. Newer processors with more than one core should be able to run virtualization software. Examples are Core 2 Duo, Core 2 Quad, AMD Athlon X2, AMD Athlon X4 and so on.

A further classification of hypervisors which exist are type 1 and type 2 hypervisors. A type 1 hypervisor runs directly on hardware and allocates resources as requested by guests. While a type 2 hypervisor runs within OS, which allocates resources to the hypervisor and then it allocates them to VMs.

There are several virtualization approaches depending on the hardware capabilities of the host and guest operating system (OS). They include:

i. Full virtualization
ii. Paravirtualization
iii. Operating system-level virtualization

Full virtualization allows diverse operating systems like Windows and Linux. Microsoft virtual server and VMware

ESX Server are two instances of full virtualization. Even though one potential shortcoming of full virtualization is its performance as application typically run fairly slower on the virtualized system. This is because the computing power of a physical server and related resources are reserved for the virtual machine manager that needs data processing (Ondrej, 2011)

Another virtualization technique is paravirtualization. It refers to a technique where each virtual machine is provided with a similar abstraction of the hardware but not the same as the underlying physical hardware. Therefore, paravirtualization technique requires modifications to the guest operating systems to work with the commands of the virtual machine (Sampathkumar, 2013). As a result, the guest operating systems are aware that they are executing on virtual machines. This provides several benefits such as less complex virtualization layer and more opportunities for optimizing as OS is aware of its environment. However, one potential downside of this technology is that the modified guest operating system cannot be migrated back to run on physical hardware (Singh and Singh, 2018).

One of the best-known systems which implement paravirtualization is the Xen (Praveen, 2011). The potential disadvantage of paravirtualization is erased with the new virtualization innovation from Intel and AMD permitting the guest operating system to run without altering it. A special privilege called the root mode is introduced in this level. The root mode allows one to install an unmodified guest operating system.

Another virtualization approach is the Operating system (OS) level virtualization, likewise called lightweight virtualization or shared operating system virtualization. This technique partitions the physical machine's resources at the operating system level. This partitioning creates several isolated virtual machines (VM), also called containers, that are isolated from one another however sharing a common operating system kernel. These virtual machines offer a virtual implementation environment that can be pronged instantly from the root operating environment (Arif et al., 2018).

The operating-system-level virtualization architecture offers some advantages where the entire containers are installed on top of a single OS. Such advantages include low overhead that assists to maximize efficient use of server resources that are available to the applications running in the containers. It is profitable and convenient as patches or modifications can easily be made to the host server that could be instantly applied to all the containers. However, this approach typically restricts the operating system choice as every guest operating system must be identical or similar to the host in terms of version number and patch level. Example of implementation of virtualization on operating system level includes OpenVZ, FreeBSD Jails (Neda, 2017).

Today, data centres and other businesses employ virtualization innovation to enhance their workload, productivity and resource management because it is shown to significantly improve the use of network assets, increase network scalability and create a durable network that can be man-aged easily. It can reduce the cost of managing a network in different ways for instance, deploying fewer machines means less power and lower cost. The traditional way of using one physical machine to one operating system alone has become obsolete and inefficient compared to a virtualized infrastructure which is inexpensive to deploy and cost-effective to maintain. Another noticeable advantage of a virtual machine is that the capital cost involved is reduced.

This study aims at evaluating the performance of virtual machines on AMD E2-7110 QuadCore Machines. Its objectives are to evaluate the performance of major virtualization tools on Linux Fedora Operating System and determine the pros and cons of virtualization software tools.

The remainder of the paper discusses the following: is as follows. Section 2 discusses related literature concerning virtualization. Section 3 provides a detailed description of the experimental process and platform. Section 4 describes the observations and Section 5 talks about the directions for future work and conclusions.

II. LITERATURE SURVEY

The performance of the guest operating system and real operating system have been appraised using Windows XP, Windows Vista, and Windows 7 implemented on Windows Vista guest operating system. This experiment analyzed the performance of the three OS with the same guest OS virtual operating system. It showed that the best performance is obtained with Windows 7 (Koh et al., 2007).

A comparative analysis of different virtual machine of Ubuntu and Windows XP was carried out to determine performance in the CPU running time, Memory usage, per cent CPU usage in the KVM hypervisor virtualization. The analysis showed how the different virtual machine of a single host affects the performance and make an informed decision about the choice of the best virtual machine assigned to the cloud guest (Patidar, 2017).

Perera and Keppitiyagama experimented to determine the performance of 32bit Debian 6.0 Virtual Machine running on Xen and VMware Esxi. The experiment aimed at measuring the virtual machines' performance based on network activity, file system I/O, CPU and memory. Benchmarks were performed on two servers with similar hardware. The benchmark results, according to Perera and Keppitiyagama, which is based on memory operations showed that on both hypervisors memory operations performance was roughly equal. Both Xen and VMware Esxi were able to perform equally while guest Oss were using high memory bandwidth. Benchmark results regarding network activities showed that both hypervisors performed closely. However, VMware Esxi performed slightly better than Xen. Benchmark results regarding file system based activities showed that VMware Esxi performed better than Xen. The performance of Xen with regards to writing to file system was poor. However, results regarding CPU intensive applications showed that both Xen and VMware Esxi performed equally but Xen's performance was slightly better than VMware Esxi. A lack of performance in the Xen's platform was because Xen was designed to host para-virtualized guest operating system with less support to fully-virtualized guests and VMware Esxi was designed to host fully virtualized guest operating system (Perera et al., 2011).

Findings from the literature survey demonstrated that the use of virtualization technologies in any information technology infrastructure is essential and is needed to improve performance and save cost. It is believed that many different businesses and institutions which deploys virtualization technologies benefit immensely from it. Various virtualization programs such as VMware Workstation, VirtualBox and Virtual PC were reviewed and explained.

III. METHODOLOGY

This study adopts a mixed research methodology. It is called a mixed research methodology because qualitative and quantitative research methodologies are combined and used (Johnson and Onwuegbuzie, 2004). In this research, a qualitative method was used to gather information and perform a literature review, this process helped to deepen the researcher's knowledge regarding virtualization and the research problem. Also, the quantitative method was used to analyze results which were obtained through the experiments. This process helped the researcher to evaluate the results, document dissimilarities and form meaningful conclusions based on the findings.

A. Data Collection Process

The methods used to gather data for this research include the following steps:

Step 1: Methods which were used in this research to gather the required data were based on experimental results. The basis of this research was the literature review which helped the researcher to increase his knowledge about the research problem by studying related studies. The main sources for literature were academic papers, articles, journals, reports, conference papers, books and reliable technology-related websites.

Step 2: Several experiments were conducted to evaluate the virtual machine performance. The results collected from the experiments were documented. The experiment process involved setting up the experimental environment on Linux Fedora 30, conducting the experiments to analyze the Drive/HDD Write Speed on Host Machine and Virtual Machine and finally, analyze the RAM Transfer Speed on Host Machine and Virtual Machine, time to unload a compressed file and installation time.

Step 3: The data collected from the experiments were converted into graphs for better evaluation. MATLAB R2016a was used to create the graphs. The graphs created helped to highlight the differences in the results and provide meaningful analysis.

Step 4: Conclusions were made based on the results of the analysis.

IV. RESULTS ANALYSIS

A. Drive/HDD write speed on the host and guest operating systems

Fig. 1 presents the results for the Drive/HDD Write Speed. This experiment focused on the drive write speed of the host and the guest operating systems. To measure the drive write speed on the guest operating systems, Linux Fedora 30 was used as the guest operating system and was installed on the three different virtualization platforms, VMware Workstation, VirtualBox and QEMU. Table I shows the drive write speed of the host and guest operating system and their average results were noted and presented in MB/Second respectively.

TABLE I. DRIVE/HDD WRITE SPEED

System	Fedora 30 1st	Fedora 30 2nd	Fedora 30 3rd	Mean
Real Host	69.60 MB/Second	68.77 MB/Second	70 MB/Second	69.46 MB/Second
QEMU	53.16 MB/Second	41.43 MB/Second	37 MB/Second	43.86 MB/Second
Virtual Box	45.72 MB/Second	44.05 MB/Second	44.65 MB/Second	44.81 MB/Second
VMware Workstation	64 MB/Second	59.05 MB/Second	55.10 MB/Second	59.38 MB/Second

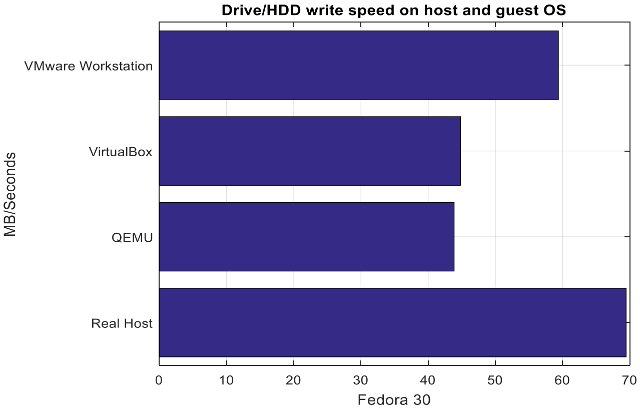

Fig. 1. Drive/HDD write speed on the host and guest operating systems

A comparison among the three virtual platforms showed that VMware workstation performed better than the other two virtualization platforms, VirtualBox and QEMU. The results show that VMware Workstation performed better than VirtualBox and QEMU by 15.77 MB/Second and 14.51 MB/Second respectively. VirtualBox performed better than QEMU by 0.95 MB/Second.

B. RAM Transfer Speed

Table II presents the results for the RAM Transfer Speed on the guest operating system (Linux Fedora 30). Linux Fedora 30 was installed on the three different virtualization platforms; VMware Workstation, VirtualBox and QEMU. Fig. 2 shows the RAM transfer speed on the guest operating system and the average results were noted and presented in MB/Second.

TABLE II. RAM TRANSFER SPEED

System	Fedora 30 1st	Fedora 30 2nd	Fedora 30 3rd	Mean
Real Host	4086 MB/s	4093 MB/s	4054 MB/s	4077.67 MB/s
VirtualBox	3092 MB/s	3109 MB/s	3059 MB/s	3086.67 MB/s
VMware Workstation	3001 MB/s	2984 MB/s	2912 MB/s	2965.67 MB/s
QEMU	1892 MB/s	2103 MB/s	1753 MB/s	1916.00 MB/s

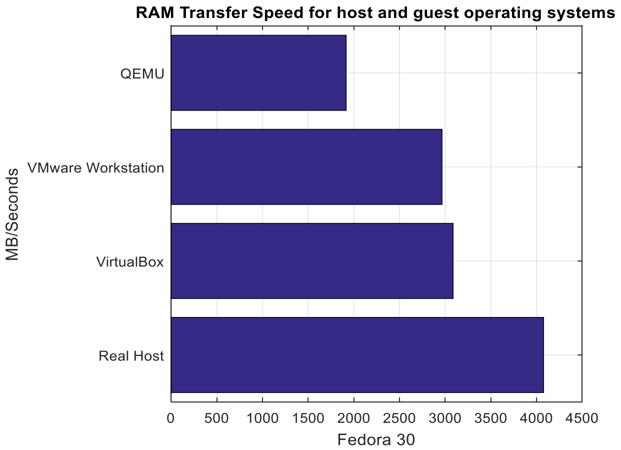

Fig. 2. RAM transfer speed on the guest operating system.

The highest RAM transfer speed was observed on VirtualBox at approximately 3086.67 MB/s, while the second-highest was observed on VMware Workstation at approximately 2965.67 MB/s. The lowest RAM transfer speed was observed on QEMU at approximately 1916 MB/s.

A comparison among the three virtualization platforms showed that VirtualBox performed better than the other two virtualization platforms. The results show that VirtualBox performed better than VMware workstation and QEMU by 121 MB/s and 1170.67 MB/s respectively. VMware workstation performed better than QEMU by 1049.67 MB/s.

C. Unpacking a compressed file

Table III shows the average results which focused on discovering the time it takes to unload a compressed file based on the host and the different virtualization software tools; VMware Workstation, VirtualBox and QEMU. To test unloading time, a compressed file was used, both on the host and guest operating systems. Fig. 3 shows the time the unloading procedure took on each virtualization software tool and their average results were obtained and presented in seconds.

TABLE III. UNPACKING A COMPRESSED FILE

System	Fedora 30 1st	Fedora 30 2nd	Fedora 30 3rd	Mean
Real Host	37 seconds	35 seconds	35 seconds	35.67 seconds
QEMU	62 seconds	50 seconds	57 seconds	56.33 seconds
VMware Workstation	49 seconds	51 seconds	50 seconds	50.00 seconds
VirtualBox	50 seconds	47 seconds	49 seconds	48.67 seconds

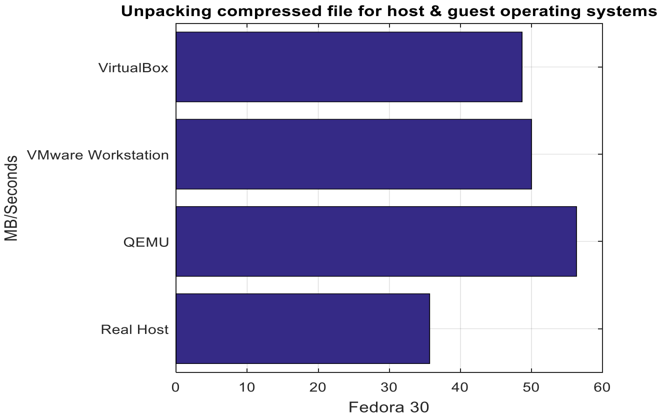

Fig. 3. Compressed file unpacking time

An evaluation of VMware Workstation, VirtualBox and QEMU indicated that the shortest amount of time was observed on VirtualBox was at approximately 48.67 seconds, which is followed by VMware workstation at 50.00 seconds, while the highest amount of time was observed on QEMU was at approximately 56.33 seconds.

D. Installation time

Table IV presents the results for the installation time for real and guest operating system. In this experiment, the real machine and three different virtual platforms (VMware Workstation, VirtualBox and QEMU) were tested based on Linux Fedora 30. Emphasis is on the time each operating system took to install which is displayed on the graph.

TABLE IV. INSTALLATION TIME

System	Fedora 30
Real Host	**479 Seconds**
VMware Workstation	**491 Seconds**
VirtualBox	**523 Seconds**
QEMU	**599 Seconds**

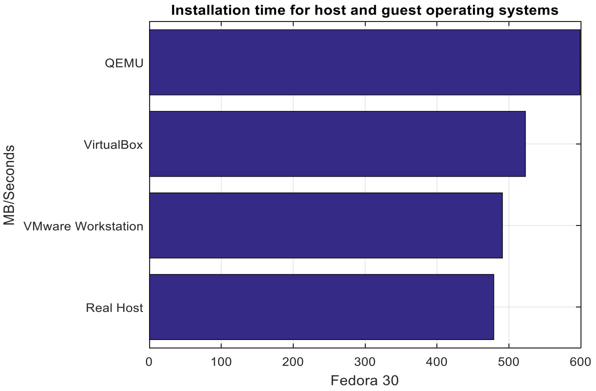

Fig. 4. Installation time for the host and guest operating systems

Based on fig. 4, the highest time to install the Linux Fedora 30 guest OS was taken by QEMU at approximately 599 seconds, whereas the lowest amount of time to install the Linux Fedora 30 guest OS was VMware workstation at approximately 491 seconds.

V. CONCLUSION AND FUTURE WORK

Through the conducting of various experiments on the virtualization software, a deeper understanding of the performance of various virtualization software such as VMware Workstation, VirtualBox and QEMU has been obtained. The preliminary results gave a clear indication of which virtualization software performed better on Linux Fedora 30. Based on the results it was observed that VMware Workstation and VirtualBox performed close to each other and better than QEMU. The experiment indicated the performance differences between VMware Workstation, VirtualBox and QEMU. The findings demonstrated that QEMU lacks both features and performance, but it is a very good product for home users who may like to use different versions of Linux OS or Windows OS together.

The future work is to perform network tests on these virtualization software tools and analyzing their performance using different network bandwidth.

REFERENCES

[1] Ali, I., and Meghanathan, N. (2011). Virtual Machines and Networks – Installation, Performance, Study, Advantages and Virtualization Options. International Journal of Network Security and Its Applications (IJNSA), 3(1), 1–15.

[2] Arif, M. M., Shang, W., and Shihab, E. (2018). An empirical study on the discrepancy between performance testing results from virtual and physical environments. Empirical Software Engineering, 23(3), 1490–1518. https://doi.org/10.1007/s10664-017-9553-x

[3] Chen, P. M., and Noble, B. D. (2001). When virtual is better than real. In Hot Topics in Operating Systems, 2001. Proceedings of the Eighth Workshop on. IEEE, 3, 133-. https://doi.org/10.1109/HOTOS.2001.990073\

[4] Hyungro Lee. (2010). Virtualization Basics: Understanding Techniques and Fundamentals, 5. Retrieved from http://dsc.soic.indiana.edu/publications/virtualization.pdf

[5] Johnson, R. B. and Onwuegbuzie, A. J. (2004). "Mixed Methods Research: A Research Paradigm Whose Time Has Come " Educational Researcher, 33, 7(14-26).

[6] Koh, Y., Knauerhase, R., Brett, P., Bowman, M., Wen, Z., and Pu, C. (2007). An analysis of performance interference effects in virtual environments. ISPASS 2007: IEEE International Symposium on Performance Analysis of Systems and Software, 200–209. https://doi.org/10.1109/ISPASS.2007.363750

[7] Neda, S. (2017). ISLAMIC AZAD UNIVERSITY SCIENCE AND RESEARCH BRACH Faculty of Engineering -Department of computer engineering Thesis for receiving « M. Sc » degree in IT engineering Subject : Improving Energy Consumption In Virtual Machines Migration by Using Threshold.

[8] Ondrej, F. (2011). Virtual machine management software.

[9] Patidar, M. (2017). Performance Analysis of Various Guest OS in Ubuntu 14. 04, 6(3), 25–30.

[10] Perera, M. P. and Keppitiyagama, C. (2011). A Performance Comparison of Hypervisors. Advances in ICT for Emerging Regions (ICT) 2011 International Conference, 120-120

[11] Praveen, G. (2011). Analysis of Performance in the Virtual Machines Environment. International Journal of Advanced Science & Technology, 32(July), 53–64. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=iih&AN=63536576&lang=es&site=ehost-live

[12] Sampathkumar, A. (2013). Virtualizing Intelligent River ®: A Comparative Study of Alternative Virtualization Technologies.

[13] Singh, B., and Singh, G. (2018). a Study on Virtualization and Hypervisor in Cloud Computing. International Journal of Computer Science and Mobile Applications, 6(1), 17–22.