Comparison of file system performance in full virtualization with MS Hyper-V and KVM hypervisors

Borislav Đorđević, *Member IEEE,* Miloš Piljić, Nenad Kraljević and Valentina Timčenko, *Member IEEE*

***Abstract* - This paper presents a comparison of the performance of native hypervisors on the example of MS Hyper-V and QEMU/KVM virtual platforms. Their quality was examined through aspects of file system performance. Filebench program was used for testing procedure, which is an application that guarantees equality and independence from the impact of hardware environment. CentOS 7, an operating system from the Linux distribution family, was used as the guest operating system. The tests were performed for one, two and finally three virtual machines that are running simultaneously. The results were further validated based on the defined hypotheses related to the expected behavior of the hypervisors.**

***Index Terms* - MS Hyper-V; QEMU/KVM; CentOS; virtual machines; performance.**

I. INTRODUCTION

In the area of information technology, virtualization is a way of creating a virtual version of computer resources. Virtualization is a simulation of the hardware or software that other software, such as various operating system, is running. Virtualization is initially applied by IBM in 1960s as a method for the logical division of mainframe computer system resources between different applications. The need to manage the "one server-one application" model has been eliminated, opening the possibility of running multiple operating systems on the same hardware platform. The advantages and savings that are obtained by using such a system are more than obvious: hardware, CPU, memory resources, administration staff. All this is a plus for virtualization in the reliability segment. The virtualization solutions allow easiness in adding new servers, as well as in data migration from one server to another. This is an additional advantage of this technology in the field of scalability.

Borislav Đorđević - Mihailo Pupin Institute, Volgina 15 Belgrade, Serbia, (borislav.djordjevic@pupin.rs)
Miloš Piljić - VISER, School of Electrical and Computer Engineering of Applied Studies, Belgrade, Serbia, (milosrin4520@gs.viser.edu.rs)
Nenad Kraljević - VISER, School of Electrical and Computer Engineering of Applied Studies, Belgrade, Serbia, (nenadk@gs.viser.edu.rs)
Valentina Timčenko - Institute Mihailo Pupin, Volgina 15 Belgrade, Serbia, (valentina.timcenko@pupin.rs)

Hardware virtualization, which is the topic of this paper, is the most popular and widespread type of virtualization [1]. The software that controls virtualization is called a Virtual Machine Monitor (VMM). According to the most common form of use in a professional IT environment, the process of creating and managing virtual machines is also called server virtualization. There are two categories of hypervisor: type-1 (native) and type-2 (hosted). In this paper, type-1 hypervisors were tested for the case of MS Hyper-V and QEMU/KVM virtual platforms (Figure 1) [2].

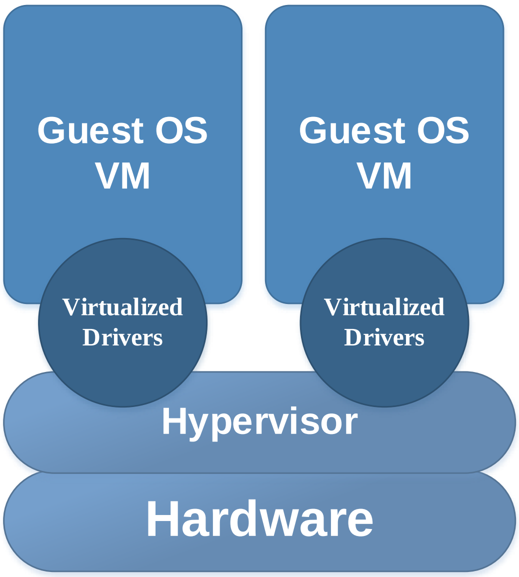

Figure 1. Native hypervisors

II. RESEARCH WORK, GOAL AND MOTIVATION

The literature related to this field is mostly focused on comparative analysis of hypervisor performance, using by different test methodology and benchmark tools. For this purpose, some proven benchmark tools are usually used, which is one of the cornerstones for obtaining quality level results. We recommend the Filebench, as open source solution, because it is a versatile, powerful, multithread and it simulates the real application workloads. We recommend the Fio tool, similar benchmark as Filebench, and some synthetic benchmarks such are Bonnie++, Postmark etc. The main contribution of this paper is the mathematical modeling

 ISBN 978-86-7466-930-3

of hypervisor-based virtualization in the context of the file system performance and applying the model on a performance case study for the interpretation of benchmark results. Because the complex virtual environment includes large number of factors, model expects there is no single winner hypervisor and depends on the case study i.e. the workload characteristics. In relation to competition, we are forcing a mathematical model and a number of case studies based on model with practical performance tests. The server variant of the virtualization stands for a great solution, primarily due to the introduction of the infrastructure costs and hardware reduction, followed by the easier administration. Still, there is a lot of room and opened questions for the improvement in this area. This paper contribution is the validation and comparison of two hypervisors, namely MS Hyper-V and QEMU/KVM, for which we have tested the quality and performances in identical conditions. Both hypervisors use full virtualization, while MS Hyper-V is also suitable for the use of the paravirtualization. As the guest operating system we have used CentOS 7, popular distribution from the Linux OS family, while for testing needs we have applied Filebench benchmark program with 4 different workloads. After defining the hypotheses, a mathematical model was set up, and validated by the obtained results [3], [4].

III. MS Hyper-V and QEMU/KVM

MS Hyper-V is an efficient hypervisor, developed by Microsoft, which enables virtualization of operating systems in a server environment (Figure 2). With the release of Windows Server 2008 R2 version, Microsoft has included a Hyper-V virtualization solution in the operating system itself. MS Hyper-V is a role that allows administrators to create multiple virtual machine, and supports isolation of partitions in which guest operating systems will run [5], [6].

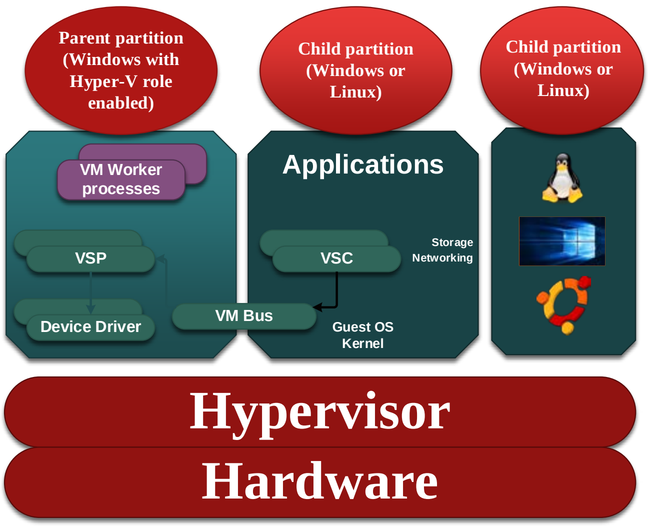

Figure 2. MS Hyper-V architecture

When it comes to virtualization under the Linux operating system, KVM (Kernel-based Virtual Machine) is almost indispensable technology. It is originally created as the Red Hat sponsored project. KVM is implemented in the form of a kernel module and is an integral part of the Linux kernel from version 2.6.20. For the KVM it cannot be said that it is a type-1 or type-2 hypervisor. On the one hand, KVM extends the Linux kernel and adds virtualization capabilities to it, allowing Linux itself to be treated as a native hypervisor (Figure 3). On the other hand, Linux is a standalone OS on which KVM functionality relies orthogonally, so it can be said that KVM runs above the main OS (hosted hypervisor), using already implemented system functions in the absence of its own (QEMU) [7-9].

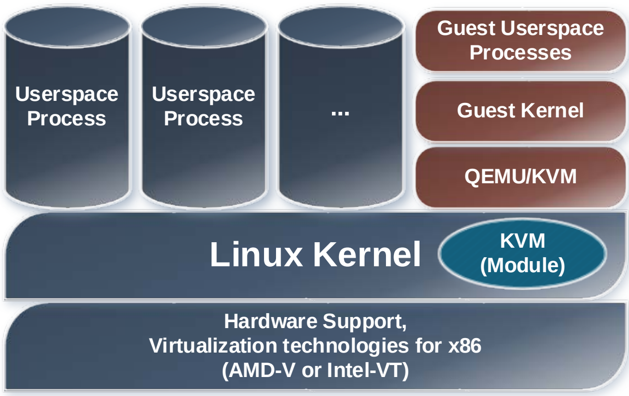

Figure 3. QEMU/KVM architecture

IV. Hypotheses about Expected Behavior

Both hypervisors are native, they work directly on the hardware and are realized in the microkernel architecture. The total processing time for each load *Tw* (Time workload) can be calculated as follows (eq. 1):

$$T_W = T_{RW} + T_{SW} + T_{RR} + T_{SR} \quad (1)$$

where T_{RW} and T_{SW} represent random and sequential data entry times, while T_{RR} and T_{SR} represent random and sequential read times. For each of these workloads, there is an expected access time for a file system that includes five components (eq. 2):

$$T_{WL} = T_{FB} + T_{FL} + T_J + T_{HK} + T_{DIR} + T_{META} \quad (2)$$

where T_{WL} represents the total time for the implementation of all operations for a defined workload, and the elements from equation (2) represent the time required for the implementation all operations related to file blocks, file lists, journaling, house-keeping, metadata and directory in the file system. There are 5 components that have an impact on the workload time T_W (eq. 3):

$$T_W = f\,(\text{Bn, gOS-FS, Hp-proc, VH-proc, hOS-FS}) \quad (3)$$

 ISBN 978-86-7466-930-3

The first and second components Bn (Benchmark) and gOS- FS (guest file system) are identical for KVM and Hyper-V. Since an identical benchmark and the same virtual machines with their ext4 guest file system are used in testing, we can assume that these components will have the same impact on the third component, Hp-proc (hypervisor processing) which represents a typical delay of hypervisor (KVM-delay, Hyper-V-delay). This represents the time that takes the hypervisor to receive a request from the virtual hardware and forward it to the host drivers. The fourth component, VH-proc (virtual hardware processing) for KVM is QEMU full virtualization, and for Hyper-V MS full virtualization. Although these are full hardware emulations, both hypervisors have their own solutions that will certainly differ in performance. The fifth component is hOS-FS (host file system). KVM uses ext4 and the Hyper-V NTFS file system, and this component is expected to cause different processing time for hypervisors. Since the tests are focused on the performance of native virtualized guests, the dominant influence of the third, fourth and fifth components of formula (3) is expected.

V. Test configuration and benchmark application

In order for testing to be adequate and high quality, it is necessary to use the same hardware configuration, the same guest operating system, choose a quality benchmark test program and the same performance measurement methodology. The tests were performed on an IBM server, whose characteristics can be seen in Table I, and the characteristics of the hard disk on which the tests were performed can be seen in Table II. CentOS 7 was used as a guest OS [10].

TABLE I - Server/Test Environment

	IBM 7945J2G - System x3650 M3
Processor	Intel® Xeon E5620 2.4GHz
Memory	32GB DDR3
Cache	12MB L3
Hard Disk	8 x Kingston 240GB SSD Now V300 SATA 3 2.5 (SV300S37A / 240G)
Network	2 x 1Gb / s

Virtual Platforms MS Hyper-V and QEMU/KVM are installed on hard drives converted into RAID 10, size 960GB (4x240GB SSD), while the other (RAID10/960GB) served as a repository on which virtual machines were created.

TABLE II - Hard Disk/Test Environment

	Kingston 240GB SSD Now V300
Model Number	SV300S37A/240GB
Model Name	SSD Now V300
Capacity	240GB
Interface	SATA 3.0 (6Gb/s)
Connectivity Technology	SATA
Hard Disk Form Factor	2.5 Inches
Read / Write Speed	450MB/s
Cache Size	240GB

All tests were done using Filebench, a benchmark program version 1.4.9.1-3. This program is designed to measure the performance of file systems and storage space and is capable of generating a large number of workloads. In this paper, 4 different workloads are used simulating environments when using services: web, mail and file server [11].

VI. Testing and results

This paper presents a comparison of the performance of virtual platforms for server use. Disk performance and data throughput were tested. In order to make testing meaningful, all virtual machines were created with identical characteristics (Table III).

TABLE III - Virtual Machine Parameters

Components	**Characteristics**
Virtual Processor	1
Memory	8GB
Virtual Hard Disk	200GB

For mail, file and web server test needs, we have modified the base code files for analyzed workloads: *webserver.f*, *varmail.f*, *fileserver.f* and *randomfileaccess.f*. First, Hyper-V was tested, which was activated as a role on Windows Server 2016, by creating one virtual machine that was tested. The same procedure is repeated for testing the environment with two and three virtual machines. Each test lasted 120 seconds and was repeated ten times. The final result represents the average value of the obtained test results. Before testing the KVM virtual platform (using CentOS 7 with the KVM option checked), the Windows server with its virtual machines was uninstalled in order to clean the environment. An identical installation and testing procedure were then conducted with the KVM virtual platform. In this way, fair-play conditions were acquired for

 ISBN 978-86-7466-930-3

both virtual platforms. The results of Varmail workload testing can be seen in Table IV and Figure 3.

TABLE IV - VARMAIL BENCHMARK RESULTS

Varmail	1VM (MB/s)	2VM (MB/s)	3VM (MB/s)	Native (MB/s)
MS Hyper- V	25.21	19.82	13.11	
QEMU/KVM	13.04	12.79	12.22	
Native OS				68.77

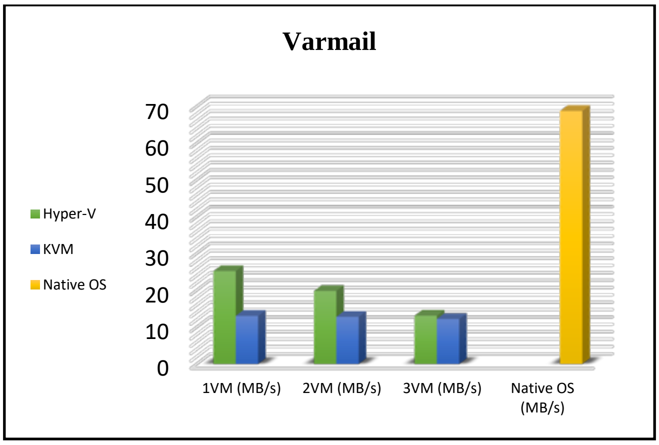

Figure 3. Varmail test results

For the "*Varmail*" workload, we notice that Hyper-V is solidly better than KVM. In this workload, besides the random read components these are synchronous random write components too for which the impact of the FS caching is very small. In this case, Hyper-V is better, primarily due to the fifth component of formula (3), where NTFS for this workload performed better in FS pair (ext4 on NTFS compared to ext4 on ext4).

The results of testing other workloads can be seen in the graphs (Figures 4.5 and 6), as well as in Tables V, VI and VII.

TABLE V - FILESERVER BENCHMARK RESULTS

Fileserver	1VM (MB/s)	2VM (MB/s)	3VM (MB/s)	Native (MB/s)
MS Hyper-V	146.04	83.75	47.43	
QEMU/KVM	155.44	138.84	115.46	
Native OS				555.63

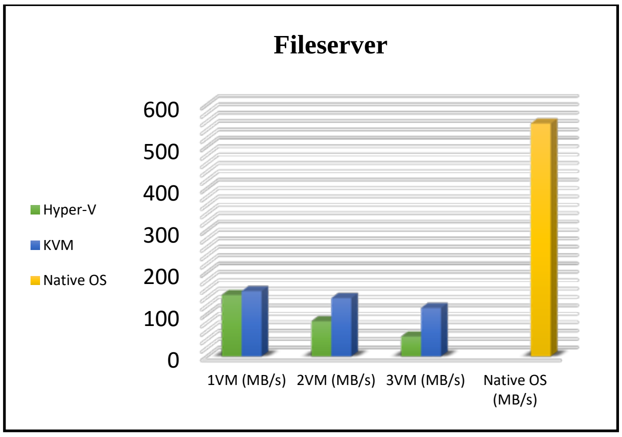

Figure 4. Fileserver test results

For the "*Fileserver*" workload, we notice that KVM is better than Hyper-V. In a complex workload such as Fileserver in which there are random and sequential write components, the FS cache effect on the guest and host OS is significant, so KVM wins primarily because of the third and fourth components of formula (3). We believe that KVM has better virtual hardware processing and less hypervisor latency.

TABLE VI - WEBSERVER BENCHMARK RESULTS

Webserver	1VM (MB/s)	2VM (MB/s)	3VM (MB/s)	Native (MB/s)
MS Hyper-V	53.94	47.73	43.28	
QEMU/KVM	39.62	37.99	36.44	
Native OS				115.26

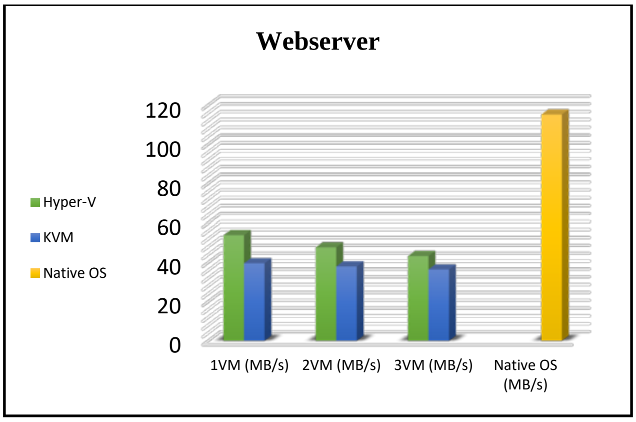

Figure 5. Webserver test results

For the "*Webserver*" workload, we can see that the Hyper-V is again solidly better than the KVM. In the Webserver workload, which has random read components and very few random write components, there is less influence of FS caching, so Hyper-V manages better, primarily due to the fifth component of formula (3), or FS pair (ext4 on NTFS in relative to ext4 to ext4) and the combined effect of FS caching.

 ISBN 978-86-7466-930-3

TABLE VII - RANDOMFILEACCESS BENCHMARK RESULTS

Random fileaccess	1VM (MB/s)	2VM (MB/s)	3VM (MB/s)	Native (MB/s)
MS Hyper-V	3153.46	2588.48	2056.96	
QEMU/ KVM	2121, 15	2007.26	1890.35	
Native OS				13780.5 2

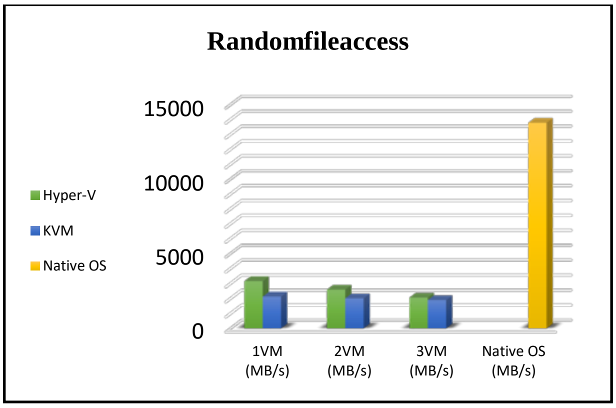

Figure 6. Randomfileaccess test results

For the "*Randomfileaccess*" workload, we again notice that Hyper-V is solidly better than KVM. In this workload, which has a lot of asynchronous random write components as well as random read components, there is a solid impact of FS caching, especially for random write, and for that reason Hyper-V performed better than KVM. This is primarily the effect of the fifth component of formula (3), NTFS, i.e. FS pair (ext4 on NTFS versus ext4 on ext4) and solid cache effect in random write.

VII. CONCLUSION

Virtualization has already proven itself in the field of information technology and has found an adequate place. In addition to all the benefits that this technology brings, it is necessary to emphasize that it's large share in the preservation of the human environment, and we can emphasize that it can be successfully used in the domain of green technologies. For the research presented in this paper, Hyper-V outperformed KVM in 3 out of 4 workloads, while in the most complex workload (*Fileserver*), KVM was dominant. For this kind of hardware and experiment, the crucial role in the differences in performance was brought by the difference in the file system of the host OS, the difference in the FS pair (ext4 on NTFS vs. ext4 on ext4). There are also differences in virtual hardware processing and hypervisor processing, which have proven to be the most complex workload (*Fileserver*). Future work in this area may focus on testing different types of servers, as well as other commonly used virtual platforms.

ACKNOWLEDGMENT

The work presented in this paper has been partially funded by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

LITERATURE:

[1] Christopher Strachey, "*Time Sharing in Large Fast Computers*". Online: https://archive.org/details/large-fast-computers/page/n5/mode/2up, 2021.

[2] Virtualization. Online: https://www.redhat.com/en/topics /virtualization /what-is-virtualization, 2021.

[3] Full Virtualization. Online: https://www.sciencedirect.com/topic s/computer-science/full-virtualization, 2021.

[4] Matthew Portnoy: "*Virtualization Essentials*", 2016.

[5] Jadran Torbić, Ivan Stanković, Borislav S. Đorđević, Valentina Timčenko: "*Hyper-V and ESXi hypervisors comparison in Windows Server 12 virtual environment*", 17th International Symposium INFOTEH-JAHORINA, 2018.

[6] Microsoft, Online: https://www.microsoft.com/, 2021.

[7] B. Đorđević, V. Timčenko, N. Kraljević, N. Maček: "*File System Performance Comparison in Full Hardware Virtualization with ESXi, KVM, Hyper-V and Xen Hypervisors*", Advances in Electrical and Computer Engineering, vol.21, iss.1, 2021.

[8] Linux KVM, Online: https://www.linux-kvm.org/, 2021.

[9] Qemu process emulator, https://www.qemu-project.org/, 2021.

[10] M. Polenov, V. Guzik, V. Lukyanov: "*Hypervisors comparison and their performance*", Computer Science Online Conference, 2018.

[11] Filebench, Online: https://github.com/filebench/, 2021.

 ISBN 978-86-7466-930-3