ESXI, Microsoft Hyper-V, XenServer: FileSystem Performance Comparasion for Type-1 Hypervisors

Borislav Đorđević, *Member, IEEE*, Valentina Timčenko, *Member, IEEE*, Stefan Kovačević

***Abstract*—This author guide gives information and a template for preparing camera-ready manuscripts for the IcETRAN Conference. You may use the built-in styles from this document if you are using Microsoft *Word* XP or later. Otherwise, use this document as an instruction set. Paper titles should be written in uppercase and lowercase letters, not all uppercase. Avoid using subscripts, superscripts or special symbols in the title and in the abstract. Full names of authors are required in the author field, not only the initials of given names. Define all abbreviations in the abstract when first using them. Do not cite references in the abstract. Do not delete the 2pt blank line immediately above the abstract; it sets the footnote at the bottom of the left column on the first page containing author affiliations and email addresses.**

***Index Terms*—Enter key words or phrases, separated by semicolons.**

I. Introduction

The application of the virtualization technologies can provide enterprises high advantages in terms of saving resources and providing additional security. It achieves optimal hardware utilization, stronger reliability and security improvements, and simpler administration. A hypervisor is a software abstraction layer that lies between hardware and operating system [1].

There are two types of hypervisors (Figure 1): type-1 hypervisor that is executed directly on hardware and manages guest operating systems (Hyper-V, ESXi, Xen); and type-2 hypervisor that is executed on the host operating system (VirtualBox, VMware Workstation).

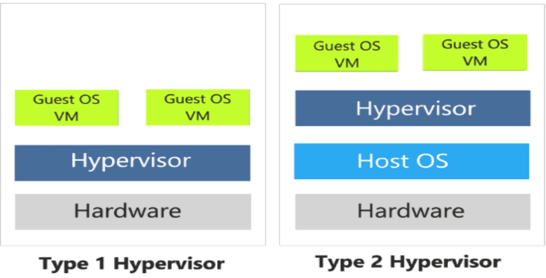

Figure 1. Hypervisor types and differences.

Borislav Đorđević - Institute Mihajlo Pupin, Volgina 15, 11000 Belgrade, Serbia, (borislav.djordjevic@pupin.rs)

Valentina Timčenko - Institute Mihajlo Pupin, Volgina 15, 11000 Belgrade, Serbia (valentina.timcenko@pupin.rs)

Stefan Kovačević - VISER, School of Electrical and Computer Enginering of Applied Studies, Belgrade, Serbia (kovacevic.stefan@hotmail.com)

As the type-1 hypervisor has direct access to hardware, while type-2 hypervisor accesses hardware through host operating system, we assume that type-1 hypervisor provides more scalability, reliability, and better performance [2].

Type-1 hypervisors are grouped into two subcategories: Monolithic and Micro-kernelized, where the main difference is reflected through device drivers characteristics (i.e. the drivers location). For the needs of this research we have evaluated three type-1 hypervisor representatives: Microsoft Hyper-V (hereinafter referred to as Hyper-V), VMware ESXi (hereinafter reffered as ESXi), and Xen with the Windows Server 2012 R2 guest operating system.

II. Related work, objective and motivation

In this paper, the research is focused to the performance comparison of three type-1 hypervisors. Numerous studies refer to the research results related to Hyper-V, ESXi, Xen, and KVM, mainly basing the results on the use of proven benchmarks: FileBench, Bonnie, HDTunePro, ATTO [12], [13].

Since the development in this area is still on the run, it is not a surprise to find large number of papers refering to the comparison and performance evaluation of different virtualization approaches. The main focus is on the speed of data writing and reading (I/O), which is particularly important for applications running in cloud environment. Additionaly, many studies explore the solutions for virtualization management, I/O speed, security, capabilities, etc [14].

The purpose of this work is to compare the performance of write and read operations when using Hyper-V, ESXi, and Xen on identical hardware, the same virtual machine parameters and the same guest operating system (Windows Server 2012 R2). Analized hypervisors are representatives of type-1, and the fundamental difference is reflected in the fact that Hyper-V uses paravirtualization, while ESXi and Xen apply the full hardware virtualization. ATTO Benchmark 4.01. software is used to determine input and read performance on virtual machines. A particular set of hypotheses and mathematical model are defined, based on which the performance is measured and interpreted [4], [13].

III. Hypotheses of the expected behavior

All hypervisors are type-1, and work directly on the hardware. They are very thin and realized in microkernel architecture. The total processing time for each workload T_W can be calcoulated according to the following equation:

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

where T_{RR} and T_{SR} represent random and sequential read components, and T_{RW} and T_{SW} stand for the random and sequential write components. For each specific workload we have calculated the expected access time for a filesystem (FS) that includes five components, as shown below:

$$T_{WL} = f(T_D + T_M + T_{FL} + T_{FB} + T_J + T_{HK}) \quad (2)$$

where T_{WL} represents the total time to complete all operations for the defined workload. T_D, T_M, T_{FL}, T_{FB}, T_J, T_{HK} represent the time required to complete all operations related to the directory, metadata, free lists, file blocks, journaling and house-keeping operations in the FS.

In the context of virtual environment, there are three components that impact the T_W:

$$T_W = f(gFS\text{-}proc, H\text{-}proc, hFS\text{-}proc) \quad (3)$$

1) *gFS-proc:* guest FileSystem processing component represent the benchmark interaction with guest filesystem. For all the explored hypervisors, as the test environment relies on the use of the identical benchmark, VMs, and guest filesystem (NTFS), it is expected that this component provides an identical effect on T_w.

2) *(H-proc):* hypervisor processing component is different for explored ESXi, Hyper-V and Xen and MS Windows OS based guests hypervisors. It depends on the type of virtualization and hypervisor delays.
When considering the context of virtualization type:

a) ESXi: For most guest operating system, Xen employs only the full-hardware virtualization. It is based on the CPU assisted full-hardware WMware virtualization.

b) Xen: exhibits strong paravirtualization effects for Linux PV guests. In the case of the MS Windows guests, Xen can only use the full-hardware virtualization. It is QEMU full-hardware virtualization with CPU hardware assisted virtualisation (HVM guests).

c) Hyper-V: For MS Windows based guests, it manifests strong paravirtualization effects, relying on VM bus (RAM) and complete Hyper-V vitrualization stack componets.

We expect remarkable advantage for Hyper-V, because paravirtualization should be solidly faster than full-hardware virtualization. Also Paravirtualization is less portable and compatible compared to full-hardware virtualization.

In the context of the Hypervisor-processing type: different delays depanding on the used hypervisor: ESXi, Xen and Hyper-V. Delay represents the time required for the hypervisor to receive the requests from virtual hardware of guest OS and forward them to the host OS drivers. The FS requests from the guest filesystem are forwarded to the FS host filesystem. In this case, we expected remarkable advantage for Hyper-V, mostly due to paravirtualization and smaller number of context switches.

3) *hFS-proc:* it is expected that the Host FileSystem processing component generates big difference between analyzed hypervisors. ESXi and Xen rely on ext4 as host OS filesystem, while Hyper-V uses NTFS. Both filesystem are modern and 64bit, but with performance differences.

As the tests are focused on the performance of MS Windows guest, it is expected the dominant influence of the 2nd and 3rd component from the formula (3), especially 2nd component which depends on the virtualization type.

This research is focused to the use of the same gostOS. It will be used as a native system and excuted as full hardware emulation on ESXi and Xen, while Hyper-V will paravirtualize the same guestOS. Based on the defined hypothesis and using practical experience, Hyper-V is expected to produce the best performance.

IV. Microsoft Hyper-v, VMware Esxi and XenServer

Microsoft Hyper-V (Figure 2) is a native hypervisor that runs directly on the hardware, just below the operating system. The virtualization software runs in the "Parent partition" and has direct access to the hardware devices. For the needs of freeing up the space for the guest operating system, the "Parent partition" creates new "child partitions" that do not have direct access to hardware, but represent virtual devices/resources. VM Bus provides communication between partitions. The parent partition starts the Virtualization Service Provider (VSP), which connect to the VMBus and processes the requests of the child partition to access devices. Virtual devices assigned to a child partition run a (VSC) , which redirects requests to VSPs in the parent partition via VMBUS. VCSs are drivers in virtual machines, which together with other component provide advanced performance and features for a virtual machine [3], [15].

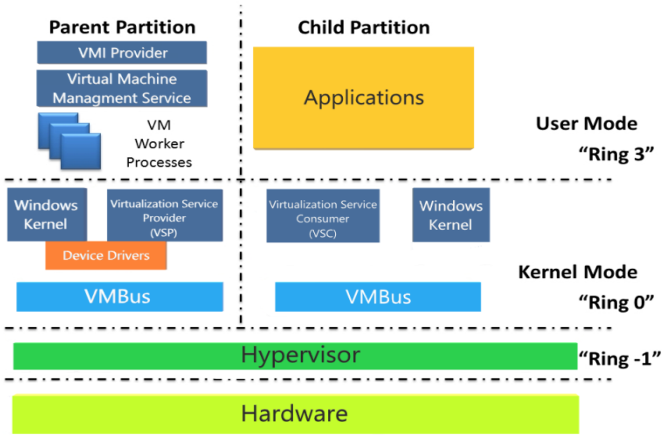

Figure 2. Microsoft Hyper-V VM Bus.

For each guest OS, Hyper-V creates a single virtual hard disk format as .vhdx file (older format .vhd). It uses VT-x or AMD-v hardware acceleration. Given that Hyper-V is the native hypervisor, as long as it is installed, other software cannot use Vt-x or AMD-v [4].

Hyper-V is a micro-core hypervisor that uses para-virtualization and full-virtualization [5], while hardware drivers do not belong to the hypervisor layer. Para-virtualization is more secure than the full-virtualization.

ESXi (Figure 3) is a native hypervisor. It is not an application software installed on the operating system, but a virtualization software that runs the kernel. Monolithic hypervisors use hardware emulation (full virtualization) [6]. The monolithic hypervisor manages hardware access to every virtual machine. It contains all the hardware drivers that virtual machines need for proper functioning (storage devices, network). The advantage of this design is that it does not require a host operating system but the hypervisor acts as an operating system platform.

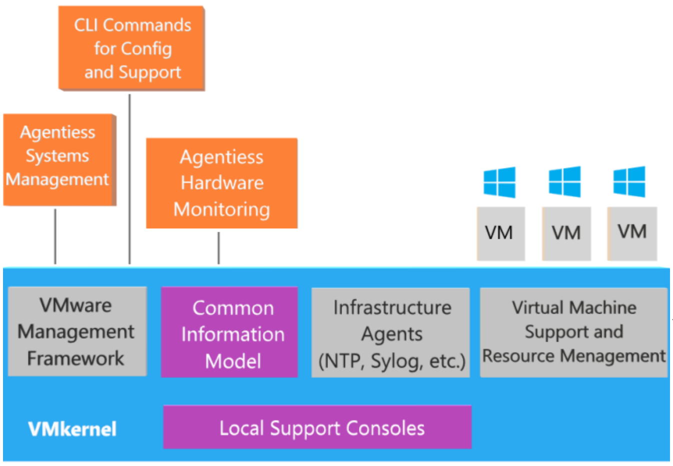

Figure 3. ESXi architecture.

VMkernel is responsible for manages virtual machines and also manages access to basic physical hardware.

The main processes that run at the top of the kernel are: [7]

- **DCUI -** Interface for the low-level configuration and management, primarily used for the initial basic configuration
- **VMM** - each running machine has its own VMM and VMX process, which provides an executable environment for the virtual machine.
- **CIM** - allows you to manage hardware from remote applications via API.

Xen (Figure 4) is the only native hypervisor that is available as open source [8]. Xen hypervisor is a software layer that runs direcly on the hardware. It is also responsible for CPU scheduling, as well as memory allocation for virtual machines. When running Xen, the hypervisor takes control of the system and then loads the first Dom0 guest operating system. Dom0 is a modified Linux kernel that has special access rights to physical I/O resources, as well as the right to interact with other virtual machines (Dom) [9].

The Dom, unlike Dom0, does not have direct access to the hardware. DomU PVS are modified Linux, Solaris, and FreeBSD operating systems, while DomU HVM guests run the standard Windows operating system or any other immutable operating system [10].

In Hyper-V, hardware drivers should not be part of the hypervisor layer.

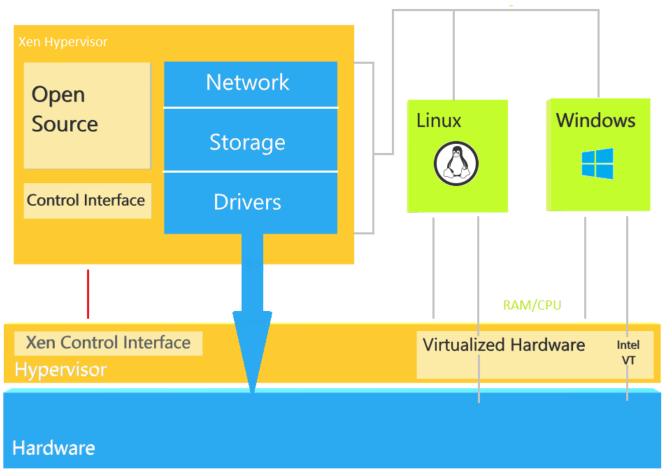

Figure 4. Xen architecture

V. TESTING

The presented graphs are based on the average values from three conducted tests run for each virtual machine. In order to achieve equal testing for all participants, the same hardware configuration was used. Table 1 shows the hardware configuration with the components used for the tests.

TABLE I
SERVER TEST ENVIRONMEENT

PHE ProLiand DL360 G5	
Component	***Characteristic***
CPU	2 x Intel Xeon E5420 QuadCore 2.5GHz
RAM	48GB DDR2
Storage Controllers	Smart Array p400i 256MB
SSD	Geil Zenith R3 GZ25R3 128GB
Network	2 x 1Gb/s

The parameters of virtual machines are shown in Table 2. All used virtual machines have identical characteristics.

TABLE II
VIRTUAL MACHINE PARAMETERS

Component	***Characteristic***
vCPU	4
RAM	4GB
Disk	20GB + 5GB
OS	Windows Server 2012 R2

Atto Disk Benchmark 4.0 is used for test procedures. It is designed to measure the performance of storage systems with different data transmission sizes and lengths of read and write tests. It supports 512B to 64MB data transfer size and 64KB to 32GB transfer lengths.

Testing is performed on the same hardware, with each virtual machine having an identical environment. The impact of other system operations is minimized during testing. First, we have installed Windows Server with Hyper-V, applied the configuration for virtual machine and installed the Windows Server 2012 as a guest. After running the testing procedure, the disk was formatted. Then, the VMware ESXi is installed, the virtual machine configured, and the corresponding guest OS installed. The tests were performed three times for each test to achieve data reliability. Next, the ESXi disk is reformatted and XenServer installed. We have configured VM, installed the appropriate operating system and run the performance measurement tests. Figure 5 and Table 3 show the write speeds with configuration based on the use of one virtual machine.

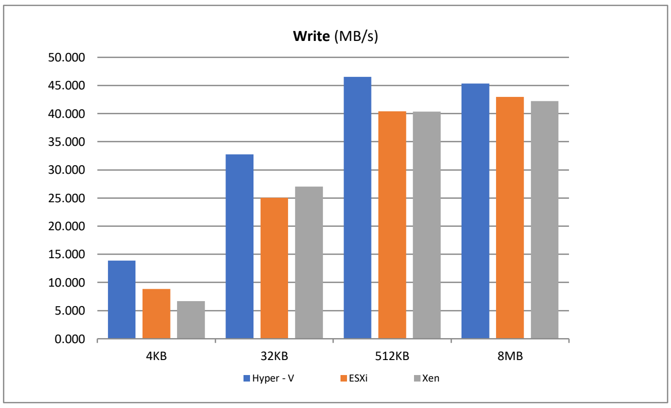

Figure 5. Write performance for a single virtual machine. (in MB/s)

TABLE III
WRITE PERFORMANCE FOR A SINGLE VM

Atto Benchmark	4KB	32KB	512KB	8MB
Hyper-V	13.89 MB/s	32.73 MB/s	46.54 MB/s	45.31 MB/s
ESXi	8.84 MB/s	25.05 MB/s	40.38 MB/s	42.97 MB/s
XenServer	6.66 MB/s	27.05 MB/s	40.33 MB/s	42.24 MB/s

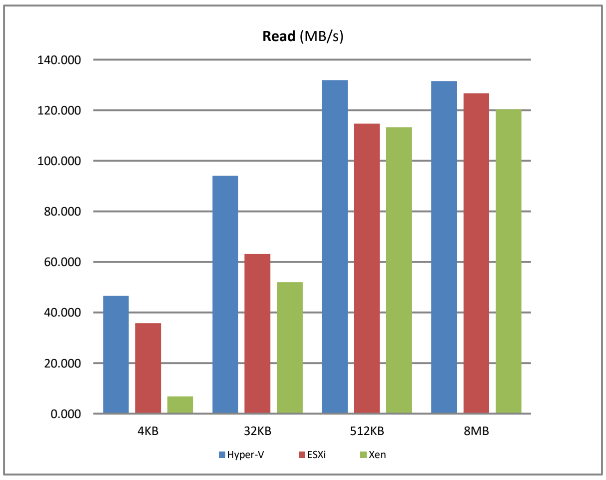

Figure 6. Read performance for a single virtual machine. (in MB/s)

Figure 6 and Table 4 show read speeds for single virtual machine case.

TABLE IV
READ PERFORMANCE FOR FOR A SINGLE VM

Atto Benchmark	4KB	32KB	512KB	8MB
Hyper-V	46.6 MB/s	94.12 MB/s	131.91 MB/s	131.59 MB/s
ESXi	35.92 MB/s	63.20 MB/s	114.66 MB/s	126.74 MB/s
XenServer	6.92 MB/s	52.05 MB/s	113.29 MB/s	120.37 MB/s

In the case for writing operation, Hyper-V shows the best performance for all block sizes. ESXi shows better writing performance than Xen with 4KB and 8MB block sizes, while Xen shows better writing performances than ESXi when operating with 32KB block size. In the case of the 512KB block size, ESXi and Xen show similar performance. For the case of read operation, Hyper-V continues to be dominantly the best. ESXi shows better read performance when compared to Xen (for all block sizes), while Xen has the strongest performance deviation for the case of writing 4KB (small block sizes).

In the second part of the test, two virtual machines were run simultaneously. The results of the write speed are presented in figure 7 and Table 5.

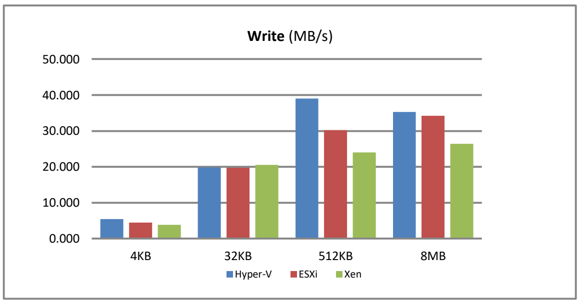

Figure 7. Write performance for two virtual machine. (in MB/)

TABLE V
WRITE PERFORMANCE FOR FOR TWO VM

Atto Benchmark	4KB	32KB	512KB	8MB
Hyper-V	5.42 MB/s	19.81 MB/s	39.01 MB/s	35.31 MB/s
ESXi	4.47 MB/s	19.77 MB/s	30.19 MB/s	34.55 MB/s
XenServer	3.80 MB/s	20.51 MB/s	24.01 MB/s	26.44 MB/s

The read speed results are shown in Figure 8 and Table 6.

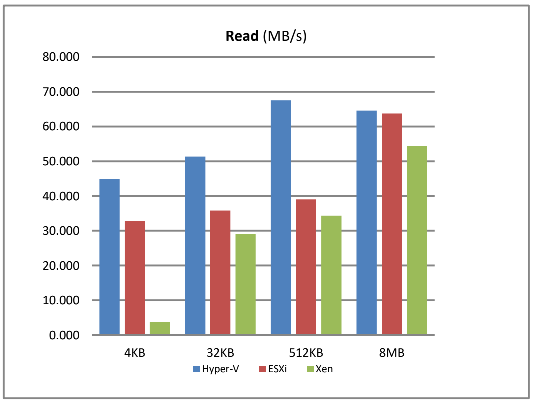

Figure 8. Read performance for two virtual machine. (in MB/s)

TABLE VI
READ PERFORMANCE FOR FOR TWO VM

Atto Benchmark	4KB	32KB	512KB	8MB
Hyper-V	44.82 MB/s	51.32 MB/s	67.51 MB/s	64.58 MB/s
ESXi	32.89 MB/s	35.89 MB/s	39.03 MB/s	63.73 MB/s
XenServer	3.82 MB/s	29.07 MB/s	34.32 MB/s	54.35 MB/s

Again, with 4KB, 512KB, and 8MB blocks size, Hyper-V provided the best results, while Xen was the best option when operating with 32KB blocks size. For write operation, in the case of 4KB, 512KB and 8MB block sizes, ESXi provides better performance when compared to Xen. In general, Hyper-V shows the best performance while Xen again has the highest deviation of write performances when opetaing with small blocks (4KB).

The third test relied on running 3 virtual machines at the same time, and it is repeated three times. The average results are shown in figure 9 and table 7.

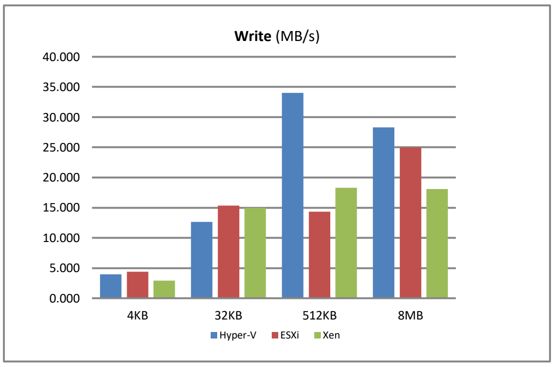

Figure 9. Write performance for three virtual machine. (in MB/s)

TABLE VII
WRITE PERFORMANCE FOR FOR THREE VM

Atto Benchmark	4KB	32KB	512KB	8MB
Hyper-V	3.98 MB/s	12.65 MB/s	34.01 MB/s	28.28 MB/s
ESXi	4.39 MB/s	15.32 MB/s	14.33 MB/s	24.98 MB/s
XenServer	2.94 MB/s	14.95 MB/s	18.32 MB/s	18.11 MB/s

The read speed results are shown in Figure 10 and Table 6.

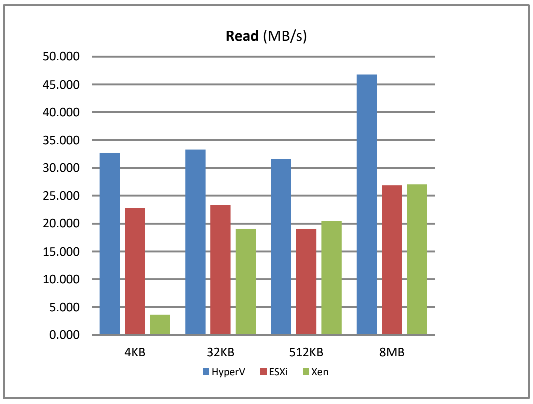

Figure 10. Read performance for three virtual machine. (in MB/s)

TABLE VIII
READ PERFORMANCE FOR FOR THREE VM

Atto Benchmark	4KB	32KB	512KB	8MB
Hyper-V	32.73 MB/s	33.31 MB/s	31.61 MB/s	46.78 MB/s
ESXi	22.78 MB/s	23.37 MB/s	19.06 MB/s	26.88 MB/s
XenServer	3.65 MB/s	19.09 MB/s	20.49 MB/s	27.05 MB/s

Unlike the previous two tests, in this test ESXi shows slightly better results for write operation for small block sizes (4KB and 32KB), while Hyper-V still dominates for block sizes of 512KB and 8MB. That Hyper-V dominance is poved for read operation as well.

In general, for all three tests routines Hyper-V with guest OS Windows Server 2012 has showed the best write and read results (Table IX). This is expected, mostly due to the paravirtualization of the guest OS, and is reflected through formula 3 and its 2^{nd} and 3^{rd} components. Hyper-V is in most of the cases much better than the other two hypervisors, mostly due to the effects of paravirtualization. It is mostly obvious in the cases of VM BUS.

TABLE IX
HYPERVISOR PERFORMANCE RESULTS

	1VM Write	1VM Read	2VM Write	2VM Read	3VM Write	3VM Read
Hyper-V	4KB, 32KB, 512KB, 8MB	4KB, 32KB, 512KB, 8MB	512KB, 8MB	4KB, 32KB, 512KB	512KB, 8MB	4KB, 32KB, 512KB, 8MB
ESXi			4KB	8MB	4KB, 32KB	
Xen			32KB			

Hyper –V is dominantly the best for:

- 1VM Write test - block sizes 4KB, 32KB, 512KB and 8MB
- 1VM Read test - block sizes 4KB, 32KB, 512KB and 8MB
- 2VM Write test - block sizes 512KB and 8MB
- 2VM Read test - block sizes 4KB, 32KB and 512KB
- 3VM Write test - block sizes 512KB and 8MB

- 3VM Read test - block sizes 4KB, 32KB, 512KB and 8MB.

Xen and ESXi results are relatively similar. ESXi was better for:

- 1VM Write test - block sizes 4KB and 8MB
- 1VM Read test - block sizes 4KB, 32KB, 512KB and 8MB
- 2VM Write test - block sizes 4KB, 512KB and 8MB
- 2VM Read test - block sizes 4KB, 32KB, 512KB and 8MB
- 3VM Write test - block sizes 4KB, 32KB and 8MB
- 3VM Read test - block sizes 4KB, 32KB, and 8MB.

Xen was better in the case of:

- 1VM Write test - block sizes 32KB
- 2VM Write test - block size 32KB
- 3VM Write test - block size 512KB
- 3VM Read test - block sizes 512KB and 8MB.

VI. CONCLUSION

In this paper we compared three powerful hypervisors of type-1: Hyper-V, ESXi, Xen, based on the use of Windows Server 2012 R2 as the guest operating system. We have set up mathematical model, measured the performances, presented and interpreted the obtained results based on the defined mathematical model. In majority of the tested cases Hyper-V proved to be significantly better than other two tested hypervisors, basicaly due to the use of paravirtualization. ESXi and Xen proved to be similar, with some variations depending on the analzyed case. The reason for this lies in the solution implemented by these two hypervisors, enforcing the full virtualization. ESXi uses its own solution, while Xen implements an open source QEMU solution. Still, in some cases these two hypervisors showed better performance than the favored Hyper-V. Future work will include a comparison of these three hypervisors with Linux-based guests.

ACKNOWLEDGMENT

The work presented in this paper has partially been funded by the Ministry of Education, Science and Technological Development of the Republic of Serbia: V. Timcenko by grants TR-32037, TR -32025, and B. Djordjevic by grant III-43002.

REFERENCES

[1] Fatma Bazargan, Chan Yeob Yeun, Mohamed Jamal Zemerly, State-of-the-Art of Virtualization, its Security Threats and Deployment Models, international Journal for Information Security Research (IJISR),Volume 2, Issues 3/4, September/December 2012.

[2] VMWare,“Understanding Full virtualization, Para virtualization and hardware Assist“ 2007.

[3] Microsoft,“Hyper-V Architecture“,online:https://msdn.microsoft.com/en-us/library/cc768520(v=bts.10).aspx, 2017.

[4] Yrewicze, Andry,“VMware vs Hyper-V: Architectural Differences", 2018

[5] A. Finn and P. Lownds, Mastering Hyper-V Deployment, Wiley Publishing Inc.

[6] John Savill, What’s the difference between monolithic and microkernelized hypervisors?, http://windowsitpro.com/virtualization/q-what-s-difference-between-monolithic-and-microkernelized-hypervisors, 2008

[7] VMware, "The Architecture of VMware ESXi", 2007.

[8] Linux Foundation, “The Xen Project, the powerful open source industry standard for virtualization”

[9] Linux Foundation, “How Xen Works”

[10] Hwanju K., Heeseung J., and Joonwon L. XHive: Efficient Cooperative Caching for Virtual Machines , IEEE Transactions on Computers, 2011.

[11] Atto Disk Benchmark, onlin:https://www.atto.com/disk-benchmark/

[12] Borislav Đorđević, Nemanja Maček, Valentina Timčenko, Performance Issues in Cloud Computing:KVM Hypervisor’s Cache Modes Evaluation, Vol. 12, No. 4, 2015, pp 147-165, http://uni-obuda.hu/journal/Dordevic_Macek_Timcenko_60.pdf,

[13] Varun Kumar Manik, Deepak Arora, "Performance comparison of commercial VMM: ESXi, Xen, Hyper-V & KVM", 2016 https://ieeexplore.ieee.org/abstract/document/7724572

[14] D. Armstrong, K. Djemame: Performance Issues in Clouds: An Evaluation of Virtual Image Propagation and I/O Paravirtualization. The Computer Journal, Vol. 54, No. 6, 2011, pp. 836-849.

[15] Leandro Carvalho, Windows Server 2012 Hyper-V Cookbook, Packt Publising, 2012.