

KRZYSZTOF RZECKI  
MICHAŁ NIEDŹWIECKI  
TOMASZ SOŚNICKI  
ANDRZEJ MARTYNA

## EXPERIMENTAL VERIFICATION OF HYPER-V PERFORMANCE ISOLATION QUALITY LEVEL

### Abstract

*The need for cost optimization in a broad sense constitutes the basis of operation of every enterprise. In the case of IT structure (which is present in almost every field of activity these days), one of the most commonly applied technologies leading to good cost-to-profit adjustment is virtualization. This allows the location of several operational systems with IT systems on a single server. In order for such an optimization to be carried out correctly, it has to be strictly controlled by means of allocating access to resources (which is known as performance isolation). Modern virtualizers allow us to set up this allocation in quantitative terms (the number of processors, size of RAM, or disc space). It appears, however, that in qualitative terms (processor time, RAM, or hard disc bandwidth), the actual allocation of resources does not always correspond with this configuration. This paper provides an experimental presentation of the achievable quality level of performance isolation of the Hyper-V virtualizer.*

### Keywords

virtualization, performance isolation, benchmark, Hyper-V

## 1. Introduction

The first computers in the beginning of the mid-20th Century were used to solve individual tasks of data processing. Due to the development of science and technology, the number and diversity of these tasks grew. In order to reduce the cost of buying and maintaining a computer (in other words, optimizing its use), hardware and software technologies such as multi-tasking and multi-threading[3] were developed. They allowed many tasks to run simultaneously on a single device. However, none of these technologies has the functionality of separating IT systems, including the operating systems. The scope of such a separation is to achieve free OS configuration and increased safety.

It is only virtualization that isolates the whole OS environment through the creation of conditions for independent installation and functioning. Briefly speaking, virtualization makes simultaneous operations of several virtual machines (VMs) possible on a single physical machine, known as the host system. A virtual machine is called the guest system, and it represents a hardware computer system. Virtualization is carried out by a virtual machine monitor (VMM) whose functionality is limited to the minimum; thus, high stability (high failure isolation quality level) is expected. Virtualization allows virtual machines to perform such functionalities as pause (freeze) and starting it back in the same place it was stopped, migration, and back up (which has a considerable impact upon the economy of an enterprise, reducing the costs of hardware purchase and maintenance as well as human labor). It is also important for environmental protection, since it helps to reduce the consumption of power as well as wear and tear on the hardware. Due to the presented advantages, the technology of virtualization has been broadly used in private data centers as well as public computing clouds. This gave rise to such concepts as IaaS (Infrastructure as a Service) and PaaS (Platform as a Service)[4], which translated into a reduction of costs and an increase in the quality of those services.

This model is very well absorbed by companies; however, there are a number of barriers to the use of such services. For this reason, these companies are still interested in optimizing the use of virtualization within their own infrastructure.

From the beginning, unfortunately, virtualization technology has been plagued by the problem of performance isolation. Performance isolation is a notion related to the influence of the operation of one guest OS on another one within the same host system. This paper presents the results of experiments over the performance isolation quality level of the Hyper-V virtual machine monitor in quantitative terms. Measurements have been taken of such resources as the processor, memory, and hard disc. Of particular interest were the differences between the declared and actual performance isolation quality levels. In order to not restrict the scope of the research, no admissible margins were defined.

This paper describes the scope of research, a review of surveys carried out so far in this field, the project of the measurements (including results), and conclusions based on the obtained results.

## 2. Motivation

The basic functionality of virtualization technology is a possibility of locating more than one OS (i.e., several VMs) on a single host machine. Unfortunately, many VMs share the same physical computer resources at the same time, so the software that manages the VMs must provide appropriate access so one VM does not reduce the task performance capability of another. This issue is known as the performance isolation of a given virtualization technology, and it constitutes the scope of the investigation which is described in this paper.

Virtualization technology was used to fulfill functionality of the project described in the acknowledgments. In operations such as self-operating creation of environments upon request and their utilization, it is necessary to control the performance isolation parameters. It is indispensable to measure the level of isolation and to properly select the allocated resources for environments.

Each VMM that is available on the market offers functionality related to the performance isolation configuration between VMs; unfortunately, there are no available publications at the moment of writing this paper that would present the confrontation of these settings with actual performance in objective terms. That is why the problem has been described and the results of VM performance tests for various Hyper-V virtualizer settings have been presented. In this way, the authors decided to verify (in the course of experiments) the efficiency of performance isolation provided by Hyper-V.

## 3. State of the art

For each OS, virtualization requires that the CPU execute additional instructions related to virtualization handling. The producers of VMMs as well as independent researchers carry out various virtualization performance comparisons. In [7], four of the most-common VMMs have been compared; namely, Hyper-V, Xen, VMWare, and KVM. The comparison consisted of performing various benchmarks on a VM using the 1 VCPU and 4 VCPU configurations. Then, the results obtained with various VMMs were compared. Similar tests are described in [6]. However, the focus there was on VMM overhead, in terms of percentage, depending on the number of simultaneously-launched VMs. In [15], a particular program was tested in relation to the number of users as well as the number of simultaneously-launched VMs.

In [18] and [19], the frameworks for VM performance tests were presented. The scope of investigation in [13] was network bandwidth. Five different VMMs and a native system were compared. The method of performance testing that we used in our work is similar to that described in [14]. Namely, we applied the concept of two VMs: we launched a benchmark on one of them while simultaneously starting a program that generated a load of a specific type on the other.

In this paper, the authors present the configuration options for Microsoft Hyper-V performance isolation [11] as well as the results of performance isolation tests for various settings.

Synthetic tests are used for single-system resource performance measurement; e.g., performance of the CPU for floating operations. The tests presented in this paper were performed using synthetic benchmarks. The CPU performance, RAM and mass storage bandwidth tests were carried out. CPU performance was measured separately for both fixed-point (with the dhrystone test [16]) and floating-point operations (the whetstone test [5]). RAM-bandwidth performance was verified with the RAMSpeed software [12], while the mass storage bandwidth was established with help of Iozone [17]. These tools were selected because of their efficiency, automatic test option, and the fact that they were used in other works [7, 19].

## 4. Types of the resources

A resource allocation is basically divided into categories in terms of:

- size – e.g., of disc space, RAM, etc.,
- bandwidth – e.g., CPU time, network bandwidth, etc.

The size is related to sub-assemblies that are responsible for storing data, while the bandwidth is responsible for the speed of data transfer and processing. In both cases, resource allocation can be either fixed or dynamic. However, in the case of size allocation, the use of allocated resources (in terms of percentage) changes relatively slowly over time (e.g., the use of allocated disc space) and the situations where the percentage of use is 0% or 100% are exceptional. This type of resource is known as type A. In the case of bandwidth allocation, the use of this resource in a unit of time very often ranges from 0% to 100%, but it rarely stays close to 100% for an extended period of time. This type of resource is known as type B.

Table 1 presents an overview of resources, including type description and expected (i.e., theory-based) performance isolation.

**Table 1**  
Overview of resources

Resource.	Type	Performance isolation
CPU time.	B	<i>High</i>
Number of cores.	A	<i>High</i>
RAM size.	A	<i>High</i>
System bus bandwidth.	B	<i>Medium</i>
Mass storage size.	A	<i>High</i>
Mass storage access bandwidth.	B	<i>Low</i>
Network interface bandwidth.	B	<i>High</i>

If the allocation is fixed, the sum of the resources allocated to the simultaneously-launched VMs cannot be higher than 100% of the resources offered by the physical (host) machine. In the case of B-type resource allocation, it limits the use of the actual virtual machine. That is why dynamic allocation is applied here, which means that

the VMM allocates a given amount of resources at a given time, which constitutes a compromise between what can be allocated and what a VM demands at the moment.

Due to the essence of the problem as well as the simplicity of A-type resource allocation, only B-type resources (excluding the network interface) were tested. We excluded network isolation as a resource that is not server dependent only.

## 5. Research objectives, tools and procedures

The type and power of hardware in medium and large companies varies very much – from PCs to professional servers. Only the largest companies have their own data centers. For this reason, the study took into account only basic equipment that is commonly available on the market.

The aim of the research was to verify the level of performance isolation of Hyper-V technology. The basis for the research was the creation of arbitrary conditions to conduct the tests and to define the measurements, not so sensitive to interfering factors and, as much as possible, unambiguous in their interpretation. Preparations for the tests involved: a computing platform, the Hyper-V parameterization, the benchmark set, and the used platform profiles.

### 5.1. Computing platform

The test was performed using two computing platforms. Their technical details are:

- Computer 1: Intel i3 530 2.93GHz, 2-core with disabled SpeedStep™ and HyperThreading™ [9], RAM: 4GB DDR3, Dual Channel, 9-9-9-24, PC3-10700 (667 MHz), HDD: WDC WD5000AADS-00S9B0, 500GB 7200 RPM.
- Computer 2: CPU: AMD Phenom II X4 945 [2], with disabled Cool'n'Quiet, RAM: 4GB DDR3, Dual Channel, 9-9-9-24, PC3-10700 (667 MHz), HDD: SAM-SUNG HD754JJ 750GB 7200 RPM.

In order to take objective base measurements, the decision was made to disable all hardware technologies that might influence the results of the experiments. For Computer 1, the HyperThreading™ technology was disabled [10]; however, the tests were performed with the SpeedStep™ energy-saving mode enabled and disabled [8]. In the case of Computer 2, we did not manage to efficiently disable the Cool'n'Quiet technology implemented within [1].

The intention of the experiments was to test isolation quality on basic systems used by medium-sized companies with self-supported servers. Such companies use virtualization mainly for better hardware utilization when providing services to internal and external clients. Additionally, software companies use virtualization in all stages of software production processes. The hardware used in such cases is based on the architectures we used in experiments as the elementary computer systems. Tests using basic systems are not sufficient to issue conclusions that will fit the needs of huge data centers, where the systems are much more powerful and based on different architectures. The last one will be used in future experiments.

We used a small-sized laboratory with a small number of VMs, because having answers for questions about performance isolation within such a small environment will make it easier to issue proper conclusions in future experiments using larger environments. However, we are aware that the problems may not be scalable.

HyperThreading technology is based on the fact that a single physical CPU core (implementing HyperThreading) is visible to the operating system as two logical CPU cores. As a result, the core is capable of processing two threads simultaneously. However, processing performance of these two threads is lower than if these threads were performed by separate physical cores.

While logical cores may be assigned for the process in the Windows Task Manager 2012, there is no similar option in the Hyper-V VMM. The results of preliminary tests using two VMs (with vCPUs assigned) suggested that the logical cores are assigned at random. The results of performance measurement attempts had a large scatter. We concluded that the tests were once carried out when the two threads run within one physical core, and other times performed on two separate cores. This problem disappeared when HT was disabled, so we decided not to use HT.

## 5.2. Hyper-V parameterization

Parameterization of a single virtual machine, which is allowed by Microsoft Hyper-V in the context of performance isolation includes:

- CPU limit – VMM planner reduces the use of a CPU,
- CPU reservation – VMM planner guarantees CPU accessibility,
- weights – VMM planner allocates the CPU according to weights,
- allocation of the number of cores to each VM.

The unit of CPU performance limit and CPU reservation is a number expressed in percents. The weight unit is a weight-point which, by default, is 100 for each VM. In the case of the RAM and mass storage, Hyper-V allows us to allocate the amount of that resource only (the amount allocated statically or dynamically), and there is no possibility of reserving a specified bandwidth.

Measurements were made under Windows 2012 Server with the Hyper-V version 3.0. The term max vCPUs means that the VM has received as many cores as many were available in the host operating system. So for the 2-core Intel processor with HT disabled, it was 2 cores, and for the 4-core AMD CPU, it was 4 cores.

## 5.3. Benchmark set

In order to test CPU performance, two test programs were prepared: dhrystone (fixed point operations) and whetstone (floating point operations). Both programs adopt the number of testing loop iterations as the only parameter. Thus, the execution time of the whole benchmark procedure is directly proportional to this number. Due to interferences in measurements arising from processor preemption, large numbers were selected as the testing procedure parameter.

Although the first implementations of the selected benchmark algorithms (whetstone [5] – '72, dhrystone [16] – '84) were issued a long time ago relative to computing history, the contemporary implementations exist and are in use by many researchers who describe results of their work in scientific journals. For example, over the last 5 years, there were over 300 new articles published by Springer and over 600 by Elsevier where dhrystone or whetstone were called (found in Springer and Elsevier search engines). Those benchmark algorithms are used by many popular benchmark tools, like SiSoft Sandra, and are used by organizations represented by well-known Internet portals who issue performance tests of new hardware available on the market, such as Tom's Hardware, Guru3D, and CPU-World. The whetstone and dhrystone implementations usually use open-source licensing, so researchers are able to investigate what kind of benchmark is in fact performed. Finally, the adequate simplicity of both benchmarks makes them usable enough in our experiments.

RAM was verified by use of the RAMSpeed software. RAMSpeed delivers the tests that involve write/read of fixed point and floating point numbers as well as write/read by use of the MMX/SSE instruction set, including or excluding cache. Moreover, RAMSpeed allows the definition of data package size and the amount of tested data. In order to maximize data bus saturation and to avoid the impact of buffering in cache memory, the tests were performed using the data block of the arbitrarily-selected size of 64MB. Write and read operations (excluding cache) were also selected.

Hard drive performance was measured using Iozone program when reading sequential data (the disc head does not have to jump often to other locations on the hard disc), when reading random data (the disc head often jumps to other location on the hard disc), when writing data (sequential or random write), and when it writes data with forced buffer emptying (sequential or random write). In order to avoid the influence of data buffering, a 4GB chunk of data was read from the disc (equal to accessible RAM) prior to launching the proper testing procedure, and each test was carried out on a separate data file.

Benchmark programs were used to test the performance as well as to generate the load.

#### 5.4. Platform profiles

The research was divided into test-platform profiles representing the configuration of hardware, Hyper-V software, and the used benchmarks. The full platform profile is composed of performance measurement results for all of the possible configuration combinations. The outcome is a 4-dimension board of numbers which constitute a single result:

- Level 1 – benchmark runtime environment settings (host or a VM with appropriate performance isolation settings).
- Level 2 – the type of benchmark (CPU, RAM, mass storage).
- Level 3 – the type of load (lack, CPU, RAM, mass storage).
- Level 4 – CPU sharing (1 – core, all cores).

Since the testing procedures for mass storage can run for several minutes, testing all of the configurations would last too long. Due to this, some groups of configurations were omitted. However, these result were of small significance.

## 5.5. Performance isolation measures

Measurement results are provided in absolute and relative values. The absolute value result is a result returned by the benchmark procedure in its own units. On the other hand, the relative result is a benchmark result obtained for a given configuration in relation to its reference result for the platform. The benchmark reference result for a platform is a result of the same benchmark for the configuration [host, benchmark, lack, number of cores].

In addition, we calculate the so-called efficiency measurement of VMM described by the equation:

$$q = \frac{v}{v_{ref}s} \quad (1)$$

Efficiency measure  $q$  of VMM is the relation of the measured performance  $v$  of VM to the expected performance of this VM based on  $s$  ratio and  $v_{ref}$  reference performance. The value of  $v$  is a benchmark result and can use units like MIPS. Ratio  $s$  is a value set up in VMM settings for particular VM and express the expected performance by percentages. The value of  $v_{ref}$  represents native benchmark results with units like MIPS. For instance, if the native benchmark result is 100% and the set CPU efficiency limit and reservation for a VM are set to 50%, then the result should be 50% and then efficiency will equal 100%. But if limit and reservation are set to 50% and the result is 25%, then efficiency will equal 50%.

## 6. Results

In the tests, two VMs were used (Tested VM and Loading VM). Their weights were set to the same value. CPU limit and reservation were set on the Tested VM, while on the Loading VM, the CPU limit was set to 100% and the CPU reservation to 0%.

Performance measurement was taken from the Tested VM at various loads generated by the Loading VM.

The following were tested:

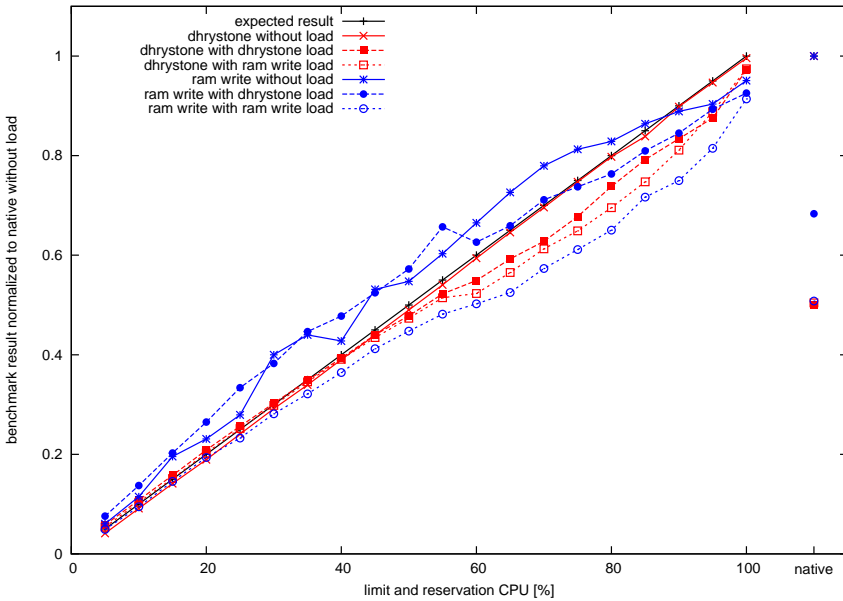
1. Provision of limit and reservation – CPU limit and reservation for the Tested VM were set to the same value.
2. The limit and reservation – as in point 1, but here the settings were performed on the Loading VM.

The tests were carried out for the 1 VCPU and max VCPU configurations, where the maximum available number of cores in the system was: Intel – 2, AMD – 4, accordingly.



### 6.1. CPU

In the test concerning the provision of CPU efficiency limit and reservation on the Intel processor, the achieved efficiency (i.e., the relative distance of the diagram line from the black line in Fig. 1) was above 96% when there was no load (dhrystone without load) and above 90% when the Loading VM was fully loaded for the CPU resource (dhrystone with dhrystone load).



**Figure 1.** Provision of limit and reservation on the Intel device (Computer 1) with disabled energy saving mode.

On the other hand, the result of the test that concerned the effects of setting a limit and reservation shows that Hyper-V does not allocate CPU reservation in a hard setting. If a machine with a reserved number of CPUs is not loaded, then unused power is provided for other VMs (Fig. 2). And when load is present, we can see that the power that should be allocated to the Tested VM happens to be, in some locations, too high. For instance, when the Loading VM has the CPU limit and reservation set to 95%, then there should be 5% left for the tested VM, while the measurement shows the result of 8%.

Hyper-V allocates CPU limits on the basis of CPU time use in percentage terms, and not its full performance. This is proven by measurements taken with the energy-saving option disabled. This is particularly clear for the AMD processor, where speed dropped 4 times in the energy saving mode (on the Intel processor, it dropped twofold). It is presented as a diagram in Figure 3, where the efficiency of providing limit and reservation ranges from 25% to 28% for settings lower than 70% of CPU allocation.

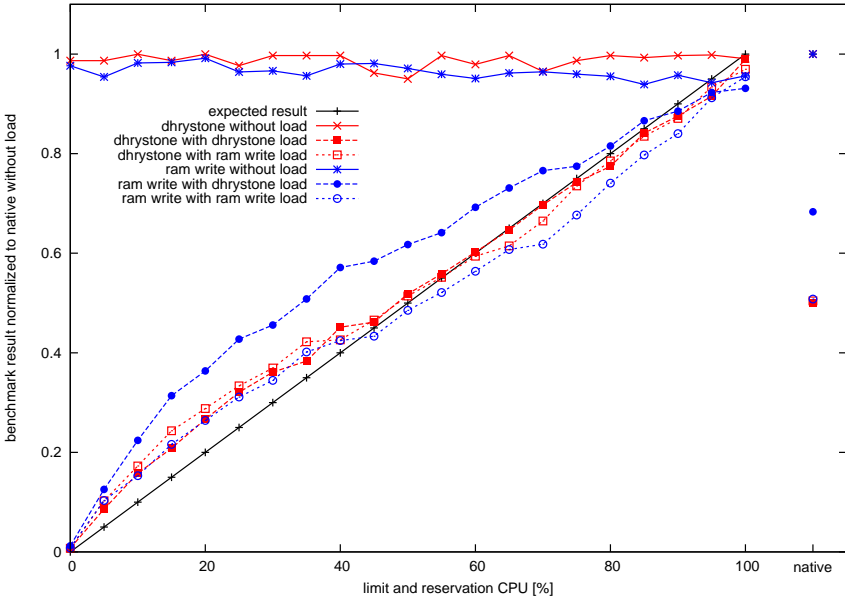


Figure 2. The result of providing processor reservation and limit for the Loading VM.

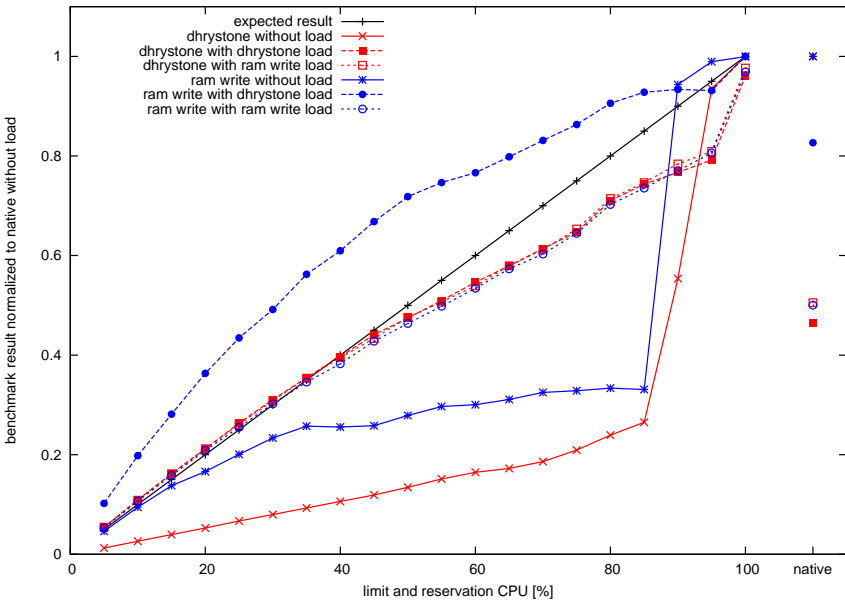


Figure 3. Provision of CPU limit and reservation on the AMD computer (Computer 2) with enabled energy saving mode.

Then, as the CPU allocation grew in terms of percentage, efficiency increased to reach 61%, 98%, and 100% for 90%, 95%, and 100% allocations, accordingly. When the CPU resource was loaded (dhrystone with dhrystone load) or when RAM bandwidth was loaded (dhrystone with ram write load), the results related to efficiency were closer to those obtained for the Intel processor with the disabled energy-saving option.

## **6.2. RAM**

RAM throughput tests, where RAM is a bottleneck, have slightly different characteristics. Hyper-V only manages CPU performance limits and reservations. This implies that, in the case the CPU, performance reservation and limit values are low and the efficiency of RAM bandwidth is too high (even 130%). And at higher limits and reservation, and when other VMs are loaded with the write to RAM operation, a decrease in the tested efficiency to 80% appeared, which means it was twice as high as in the case of a CPU (where performance dropped to 90%). This is the result of the bottleneck, the source of which is RAM.

In tests aimed at verifying the results of providing limit and reservation for the Loading VM (presented in Fig. 2), we can see that the efficiency of RAM bandwidth is considerably higher than it was in the previous test. Thus, the method of CPU allocation in Hyper-V results in a machine with set limits and reservation obtaining bandwidth that is too small, while the other machine obtains bandwidth that is too great as compared with what was expected on the basis of limit and reservation settings.

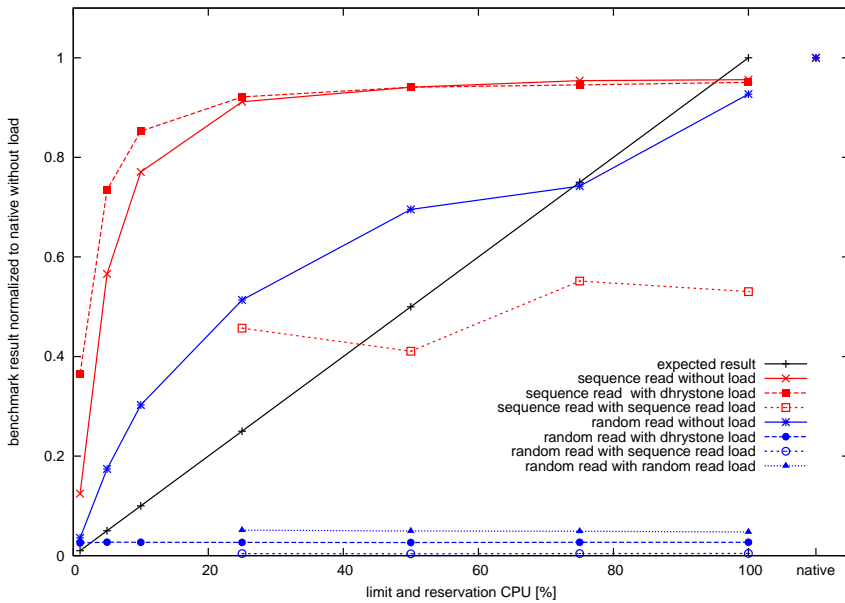
The results obtained for the AMD processor with the energy saving option enabled are even more interesting (Fig. 3). What we see here is very low efficiency at no load. This is the result of decreasing the processor multiplier with no FSB or RAM timing.

## **6.3. HDD**

Mass storage tests revealed that Hyper-V does not provide any performance isolation. As it has been shown (Fig. 4), mass storage performance strictly depends on other simultaneous operations that are run on it. The result of setting a CPU limit is that the benchmark procedure is slower at sending the write order or new data download order to the mass storage controller.

## **7. Conclusions**

The performed tests showed that Microsoft Hyper-V can provide CPU performance isolation pretty well. The situation is a bit worse when we speak of RAM bandwidth performance isolation; and as for mass storage performance isolation, Hyper-V cannot provide it at all. Moreover, in establishing CPU limits and reservations, Hyper-V only allows for the percentage of processor time in a time unit. This functions well after the energy-saving mode is disabled. However, when this mode is enabled, then CPU



**Figure 4.** Provision of limit and reservation on the Intel device (Computer 1) with disabled energy saving mode; tests performed for mass storage.

allocation is based on its current speed. Thus, when the limit is 40% and in the energy-saving mode, the processor only has  $\frac{1}{4}$  of the power, and what we actually obtain is a mere 10%. Since the processor is not loaded in at least 90%, it does not switch to a faster mode.

RAM access performance isolation looks even worse when RAM is the bottleneck.

The results of tests, however, showed that Hyper-V does not try to maintain performance isolation at all (in the case of access to mass storage). This creates an obvious threat from a malicious applications installed on one of the VMs, which would use the hard disc in an intensive way. Such an application might efficiently slow down the operation of applications on other VMs.

From a business point of view, this study provide essential information for IT professionals in commercial companies. This study provides reliable information about the characteristics of the behavior of individual types of resources. It also allows us to get a broader perspective of thinking in the categories of the concept of “performance isolation”. Such knowledge reduces the time of deployments of virtualization, improves the quality of these deployments, and may affect purchases or modernization of the relevant infrastructure components.

Engineers receive clear information about CPU and RAM resource characteristics in virtual environments. In contrast, a lack of performance isolation of mass storage is bad news. This means that there is a major difficulty in predictability planning for

infrastructure in this regard. If the expected performance isolation quality is high, there will be required:

- Significant investments in high-speed hard disk drives (especially SSD) and high-speed interfaces (SAS, Fibre Channel).
- Continuous measurements of the actual use of mass storage to be able to analyze situations of low performance isolation.

The fact that mass storage is a major problem within virtualization is consistent with the intuition and experience of engineers. However, the results of this study indicate exactly what engineers can expect in relation to particular types of resources and how they can more effectively plan and monitor virtualized environments.

## Acknowledgements

*The research described in this paper was carried out as part of project No. POIG.01.04.00-12-075/11 – “Development of an innovative utility and maintenance platform”, co-financed by the European Regional Development Fund, the Innovative Economy Operational Programme 2007-2013.*

## References

- [1] AMD: *AMD Cool'n'Quiet Technology*. Visited 2013.09.19.
- [2] AMD: *AMD Phenom II 945 processor specification*. Visited 2013.09.19.
- [3] Butenhof D. R.: *Programming with POSIX Threads*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997. ISBN 0-201-63392-2.
- [4] Demchenko Y., Makkes M., Strijkers R., De Laat C.: Intercloud Architecture for interoperability and integration. In: *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, pp. 666–674. 2012. <http://dx.doi.org/10.1109/CloudCom.2012.6427607>.
- [5] Harbaugh S., Forakis J. A.: Timing Studies Using a Synthetic Whetstone Benchmark. *Ada Lett.*, IV(2): 23–34, 1984. ISSN 1094-3641. <http://dx.doi.org/10.1145/998395.998396>.
- [6] Huang X., Bai X., Lee R.: An Empirical Study of VMM Overhead, Configuration Performance and Scalability. In: *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on*, pp. 359–366. 2013. <http://dx.doi.org/10.1109/SOSE.2013.72>.
- [7] Hwang J., Zeng S., Wu F., Wood T.: A component-based performance comparison of four hypervisors. In: *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pp. 269–276. 2013.
- [8] Intel: *Enhanced Intel SpeedStep Technology – How To Document*. Visited 2013.09.19.
- [9] Intel: *Intel Core i3-530 Processor (4M Cache, 2.93 GHz) specification*. Visited 2013.09.19.
- [10] Intel: *Intel Hyper-Threading Technology*. Visited 2013.09.19.

- [11] Microsoft: *Windows Server 2012 R2*. Visited 2013.09.19.
- [12] Rhett M. Hollander P.V.B.: *RAMspeed, a cache and memory benchmarking tool*. Visited 2013.09.19.
- [13] Schlosser D., Duelli M., Goll S.: Performance Comparison of Hardware Virtualization Platforms. In: *Proceedings of the 10th International IFIP TC 6 Conference on Networking - Volume Part I*, NETWORKING'11, pp. 393–405. Springer-Verlag, Berlin, Heidelberg, 2011. ISBN 978-3-642-20756-3.
- [14] Somani G., Chaudhary S.: Application Performance Isolation in Virtualization. In: *Cloud Computing, 2009. CLOUD '09. IEEE International Conference on*, pp. 41–48. 2009. <http://dx.doi.org/10.1109/CLOUD.2009.78>.
- [15] Tope I., Zavarsky P., Ruhl R., Lindsog D.: Performance and Scalability Evaluation of Oracle VM Server Software Virtualization in a 64 Bit Linux Environment. In: *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pp. 1119–1124. 2011. <http://dx.doi.org/10.1109/PASSAT/SocialCom.2011.126>.
- [16] Weicker R. P.: Dhrystone: A Synthetic Systems Programming Benchmark. *Commun. ACM*, 27(10): 1013–1030, 1984. ISSN 0001-0782. <http://dx.doi.org/10.1145/358274.358283>.
- [17] William D. Norcott D. C.: *Iozone Filesystem Benchmark*. Visited 2013.09.19.
- [18] Ye K., Che J., Jiang X., Chen J., Li X.: vTestkit: A Performance Benchmarking Framework for Virtualization Environments. In: *ChinaGrid Conference (China-Grid), 2010 Fifth Annual*, pp. 130–136. 2010. <http://dx.doi.org/10.1109/ChinaGrid.2010.12>.
- [19] Yuan P., Ding C., Cheng L., Li S., Jin H., Cao W.: *VITS Test Suit: A Microbenchmark for Evaluating Performance Isolation of Virtualization Systems*. In: *e-Business Engineering (ICEBE), 2010 IEEE 7th International Conference on*, pp. 132–139. 2010. <http://dx.doi.org/10.1109/ICEBE.2010.71>.

## Affiliations

### Krzysztof Rzecki

Cracow University of Technology, Krakow, Poland , [krz@pk.edu.pl](mailto:krz@pk.edu.pl)

### Michał Niedźwiecki

Cracow University of Technology, Krakow, Poland , [nkg@pk.edu.pl](mailto:nkg@pk.edu.pl),  
AGH University of Science and Technology, Krakow, Poland, [nkg@agh.edu.pl](mailto:nkg@agh.edu.pl)

### Tomasz Sośnicki

Cracow University of Technology, Krakow, Poland , [tom.sosnicki@gmail.com](mailto:tom.sosnicki@gmail.com)

### Andrzej Martyna

VSoft Spółka Akcyjna, ul. Juliusza Lea 112, 30-133 Krakow, [andrzej.martyna@vsoft.pl](mailto:andrzej.martyna@vsoft.pl)

**Received:** 20.09.2013

**Revised:** 13.02.2014

**Accepted:** 20.02.2014