Acta Universitatis Sapientiae
Electrical and Mechanical Engineering, 8 (2016) 42-61
DOI: 10.1515/auseme-2017-0004

Evaluation of Different Thin-Client Based Virtual Classroom Architectures

Örs DARABONT, Konrád József KISS, József DOMOKOS

[1] Department of Electrical Engineering, Faculty of Technical and Human Sciences,
Sapientia Hungarian University of Transylvania, Tg. Mureş,
e-mail: darabont_ors@yahoo.com, konrad@ms.sapientia.ro, domi@ms.sapientia.ro

Manuscript received November 20, 2016; revised February 26, 2017

Abstract: This paper presents an evaluation of different methods used to deliver virtual machines capable of being accessed remotely by thin-clients. The objective of the research was to provide a recommendation for building a cost-effective computer infrastructure for use in two scenarios: as a programming lab, and as an office infrastructure.

We have found that different thin-client solutions based on single board computers are reliable solutions for commercially available thin client replacement, because they can run free Linux-based operating systems, can handle Remote Desktop Protocol, have lower acquisition costs, lower power consumption and offer almost the same computing performance.

For providing remote desktops, there are several methods and virtualization platforms available. We benchmarked some of these platforms in order to choose the one best-suited for implementation. Our conclusion is that Microsoft Remote Desktop Services outperforms the virtualization based solutions, but it entails high license fees. Of the virtualization solutions tested, the VMW are ESXi based one is the most reliable choice.

Keywords: platform virtualization, virtual machine, virtual classroom, remote desktop, thin-client/server architecture.

1. Introduction

The fast spread of computer networks with broadband Internet access, along with the rapid development of different operating systems, virtualization technologies and platforms, makes it possible to use different computing solutions in many domains of interest [1]. The release and spread of virtualization platforms has allowed the development of cost-effective information systems capable of providing dynamic resource management and simplified system administration [2], [3]. Thus, the educational process in the fields of informatics, computer science, information technologies and communications or other related domains can be optimized to minimize the cost of educational lab resources [4], [5]. Universities often use virtualization technologies to reduce the total cost of IT equipment and management, and also to simplify management of resources and technologies [6], [7]. These observations apply also in the case of the office infrastructure.

An infrastructure introduced by VMware (http://www.vmware.com/) for supporting desktop virtualization for research and educational user communities is presented in paper [8]. The software infrastructure is called VMLab. These software resources were used to develop also the Virtual Desktop benchmarking toolkit for thin-clients called VDBench [9].This toolkit can perform CPU usage, memory usage and network bandwidth measurements.

Paper [4] presents an original educational infrastructure using virtualization technologies implemented in a training process at the Kaunas University of Technology. This infrastructure allows university students and staff to access virtual desktops and applications online, both in computer classes at the University, and for self-working at home.

In paper [5] the authors underline how recent developments in network and hardware technologies have made thin-client/server architecture more efficient regarding the total cost of ownership, administration, maintenance, security and power consumption, and introduce such architecture in a Turkish Public College.

The novelty of this paper lies in the description and evaluation of different virtualization techniques and the delivery of virtual machines (VMs) or virtual desktops (VDs) that can be accessed remotely by using thin clients. The aim of our research was to find an optimal solution; one that helps reduce energy consumption and which allows us to create a cost-effective network infrastructure in terms of the acquisition cost of hardware and software licenses as well as long-term operating and maintenance costs. The objective of the project was also to develop a cost-effective computer infrastructure considering desktop virtualization technologies and thin clients for use in two scenarios: as programming labs and as office infrastructure if possible [1], [6], [7].

In the first stage of the project we studied different thin client solutions, from commercially available dedicated thin clients to alternative thin client solutions. The findings of this stage of the project are briefly presented in Section 2 and also in paper [6].

For providing remote desktops (RDs)or VMs, there are several virtualization software tools available [10], [12], [14] such as VMW are ESXi, Citrix Xen Server, Oracle VM VirtualBox or Microsoft Hyper-V, and other alternatives such as Microsoft Remote Desktop Services [7]. Our aim was to benchmark these solutions in order to choose the one best-suited for the implementation of a virtual classroom and an office. Section 3 of the paper introduces the benchmarked system architectures and describes the test environments.

Section 4 contains the performed evaluation tests with detailed discussion on the results obtained.

Section 5 summarizes the main conclusions and the paper concludes with acknowledgements and references.

2. Thin-client solutions

Thin-clients are typically small form factor computing devices which are able to act as interfaces between the user and a server computer [15], [16], [17], [18]. These thin-clients include peripherals such as keyboard and mouse connectors (PS/2 or USB) for registering input signals from the user, video output (HDMI, VGA, DVI, etc.) and network interfaces for communication with the server computer [6], [10].

Because the thin-client acts only as an interface – the actual computing being performed at the server side – it requires no powerful internal components, and hence requires low electrical power consumption. This also reduces the cost of manufacturing and thus the acquisition cost of a thin-client. Additionally, as most thin-clients do not require active cooling, noise pollution is reduced.

In a complex computing infrastructure it is common practice to have most of the servers in one or more datacenters, the users being located in different buildings or different geographical locations. In this way the servers are easier to maintain and service [2].

From a functional perspective three categories of thin-client device can be distinguished [7], [10], [15].

The **zero clients** (ultra-thin-clients) are very small devices that don’t feature a standalone operating system. These devices are based on proprietary PC-over-IP protocols (PCoIP) and have hardware decoding (through dedicated chips) that don’t even require a local CPU to function.

Typical thin-clients are embedded computer systems that feature most of the components of a PC. They have a CPU, RAM, persistent storage (flash memory or HDD), peripheral connections (USB) and they run a fully functional operating system (generally Linux-based). The user can also run applications locally if necessary.

Table 1: Comparison of tested thin-clients

Thin-client device	CPU	Memory	Video output	Audio Input-Outputs	USB	Operating system	Power	Price
HP T5545	VIA Eden x86, 1 GHz, single core	512 MB	VGA, DVI	Microphone input, headphones output	6 x USB 2.0	Windows CE 6, HP ThinOS (Linux-based)	50W	$299
NComputing N500	Numo 3 SoC ARM Cortex-A9, 1 GHz, dual core	1 GB	DVI-I	Microphone input, headphones output	4 x USB 2.0	None	5W	$200
Raspberry Pi Model B	ARMv6, 700 MHz, single-core	512 MB	HDMI	Headphones output	2 x USB 2.0	Raspbian (and other Linux distributions)	3.5 W	$35
Raspberry Pi Model B+	ARMv6, 700 MHz, single-core	512 MB	HDMI	Headphones output	4 x USB 2.0	Raspbian	3 W	$35
ODroid U3	ARMv7, 1.7 GHz, quad-core	2 GB	uHDMI	Microphone input, headphones output	3 x USB 2.0	XUbuntu (and other Linux distributions), Android	5W	$65
Cubie board 2	ARM Cortex-A7, 1 GHz, dual-core	1 GB	HDMI	Line input and output	2 x USB 2.0	Ubuntu, Android 4.2, other Linux distributions	3W	$65
Asus Google Nexus 7	ARM Cortex-A9, 1.2 GHz, quad-core	1 GB	7” touch screen	Built in microphone and speaker	1 x uUSB	Android 4.4.4 Kitkat	2.25W	$230
HTC Desire C	ARM Cortex-A5, 600 MHz, single-core	512 MB	3.5” touch screen	Built in microphone and speaker	1 x uUSB	Android 4.0.3	~1W	$15

The third category of thin-client includes **Personal Computers** (PCs). PCs can be used as standalone computing devices and also as thin-clients. From the perspective of functionality in thin-client mode they act like a dedicated thin-client, possibly featuring higher performance than dedicated thin-clients.

From the hardware perspective we can differentiate two categories of thin-client [6], [7]. The first category consists of purpose-built, commercially available thin-clients (these include zero clients and thin-clients). These are products that were built to function specifically as thin-clients. The second category features multi-purpose embedded computers, single board computers and development boards that can be adapted and configured to function as thin-clients. While commercially available thin-clients tend to have a high acquisition price, single board computers are becoming faster and cheaper at a rapid pace [7]. See table 1 for acquisition cost details.

Because of the great potential of single board computers regarding performance and their low acquisition cost, we conducted experiments to find out if these devices are a viable alternative to commercial thin-clients or not. We also wanted to pinpoint any technical limitations and address them.

Table 1 presents a comparison between the tested thin-client equipment. We studied different thin-client solutions: commercially available dedicated thin-clients and alternative thin-client solutions such as Android based smart devices and single board computers such as the Raspberry Pi, Odroid U3 and Cubie Board. We found that alternative thin-clients are a reliable solution for commercially available thin-client replacement. These single board computers are flexible, they can run Linux-based operating systems, can handle Remote Desktop Protocol (RDP) using the freeware FreeRDP software application [13] for connecting to a server, the acquisition costs are an order of magnitude cheaper than that of commercially available thin-clients and they also have lower power consumption [7]. Furthermore, all the operating systems, software applications and tools used at the thin-client side are freely available, therefore in cases where a large number of client stations are required, we can achieve large savings in license cost.

3. Benchmarked architectures

The main purpose of this paper is to present a comparative performance study of the different virtualization techniques and delivery of VMs and VDs. For remote desktop virtualization there are several virtualization software tools including VMWare ESXi, Citrix Xen Server, Oracle VM VirtualBox or Microsoft Hyper-V, and other alternatives such as Microsoft Remote Desktop Services. We have tried to evaluate some of these solutions. The benchmarked systems and test infrastructure are presented here.

All measurements were made on the same hardware configuration, an HP ProLiant DL380p Gen 8 server station with an Intel C600 Series chipset, equipped with one Intel Xeon E5-2609 v2 @ 2.5 GHz processor having 4 cores, 8×4GB DDR3 memory, Smart Array P420i/ZM storage controller and 3×500 GB 7200 RPM SAS HDD, connected to the network through a 1Gb 4 port 331FLR Ethernet adapter.

With the complete test environment in place, performance tests were carried out to evaluate memory, CPU and storage management using the built-in benchmark tools.

3.1. Microsoft Hyper-V and Remote Desktop Services

The first test environment included the two Microsoft tools for Remote Desktop Virtualization: Hyper-V hypervisor and Remote Desktop Services.

Fig. 1 presents the test environment for both Remote Desktop Services (RDS) and Hyper-V based Desktop Virtualization scenarios. We used the Windows 2012 Server R2 operating system with both Hyper-V role and RDS installed and configured on an HP ProLiant DL380p Gen 8 server station. On the left part of the figure are the RDS components, marked in green, and on the right the Hyper-V with the guest VMs and their clients marked in red. Hyper-V Manager was used to create and manage VMs.

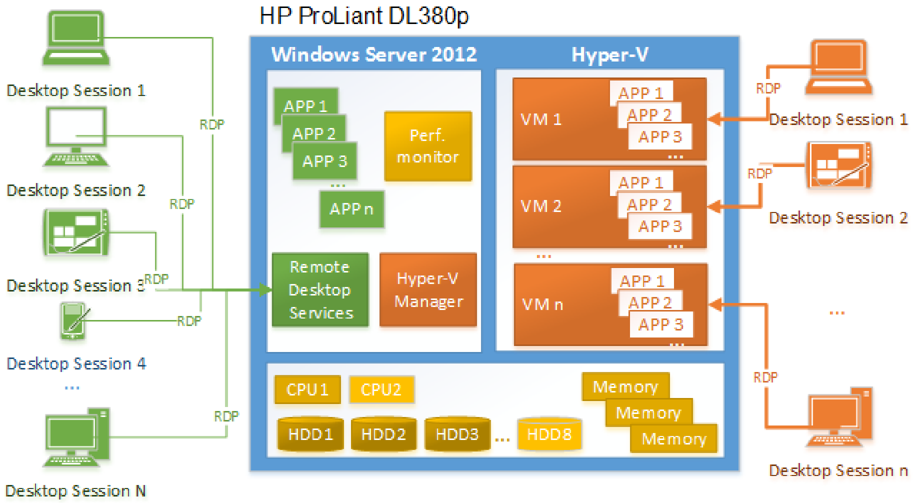


Figure 1: Test environment for Microsoft Hyper-V and Remote Desktop Services

A. Microsoft Hyper-V

Hyper-V is a proprietary virtualization environment created by Microsoft and included with the Windows Server edition operating systems. It is a Type 1 hypervisor that runs directly on the system hardware offering efficient virtualization. The Hyper-V hypervisor is responsible for server hardware resource sharing between the created VMs. Several types of client presented in Section 2 can be connected to the VMs or directly to RDS such as personal computers, laptops, dedicated thin-clients or thin-clients based on single board computers, and even Android based smartphones or tablets. The connections are made using Remote Desktop Protocol (RDP).

In order to test the performance of both our Hyper-V and Remote Desktop Services implementations we used the built-in Performance Monitor tool, marked in yellow in Fig. 1.

For our tests in the Hyper-V scenario, we created 12 second-generation virtual machines, each with one quad-core virtual processor, 2 GB of RAM, IDE attached VHDx virtual disks and a virtual network adapter. The virtual machines (guests) were running Windows 7 Enterprise Edition operating systems. Fig. 2 shows the configuration of the virtual machines in the Hyper-V Manager application used for creating and managing the virtual machines. In order to improve storage I/O throughput, we disabled the page file and the System Restore service in the guest operating systems.

B. Remote Desktop Services

Microsoft Remote Desktop Services is an alternative means of offering virtual desktops to users. RDS is also included in Windows Server editions (2012 and up), but has a separate licensing strategy. The test environment is presented on the left side of Fig. 1. RDS makes multi-user connections to a Windows server station possible. Connected clients use RDP to transfer commands to the server and to receive the graphical user interface of the remote desktop. Connected users are granted access to all the software applications installed on the server station. In this case the received remote desktop is a Windows Server 2012 R2 desktop environment.

3.2. VMWare ESXi

The test environment used was based on an ESXi vSphere hypervisor core and the vCenter Server software management package. The two modules were deployed on two different machines: the vSphere on the HP ProLiant DL380p Gen 8 server station and the vCenter Server on a management server station. Reaching the management server was possible through HTTP protocol using the vCenterServer Web Client application.

The standalone management server improves system performance and capacity because the management software application doesn’t require an extra load for the server running VMs. However, this makes the configuration and administration of the whole system a little bit more difficult.

When the ESXi hypervisor is freeware, the vCenter Server management application can be purchased only with a license.

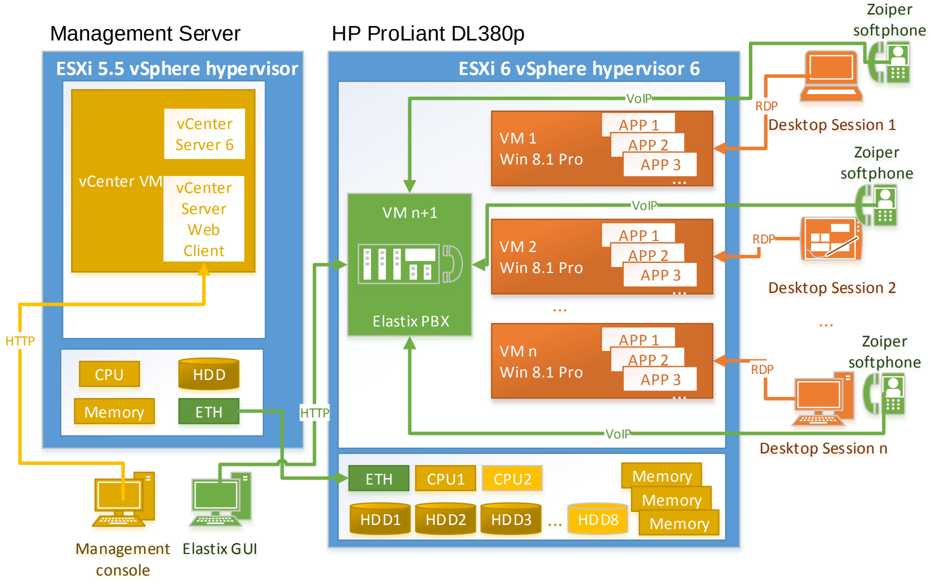


Figure 2: Test environment built using VMWare ESXi

3.3. XenServer

The XenServer is an open source virtualization platform previously developed by Citrix which is freely available to be used for any educational, research or business applications.

The platform contains a hypervisor core, namely the XenServer and a management application. The management application is built into the XenServer and can be reached via HTTP protocol using the Citrix XenCenter application.

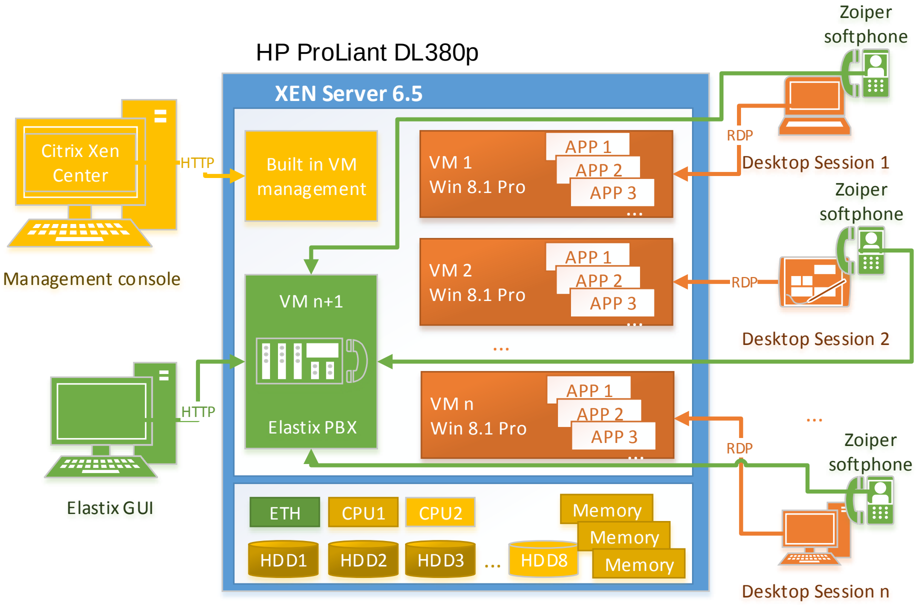


Figure 3: Test environment built using XenServer

In all virtualization test scenarios we used a virtual machine configured as a VoIP PBX. The software PBX used was Elastix.The PBX can be reached and configured through HTTP protocol using the Elastix GUI application. On the client side we installed Zoiper software IP phones. The advantage of this softphone, besides incurring no additional costs, is that it runs on different operating systems (Windows, Linux and Android), and its low resource requirement means it can be easily run on thin-client devices.

4. Measurements and test results

For the performance tests we configured 12 second-generation virtual machines, each featuring 4 CPU cores, 1 GB of RAM, SCSI virtual hard drive controllers and virtual network interface cards. The virtual machines ran Windows 8.1 Pro operating systems.

For testing the Microsoft Hyper-V and Remote Desktop Services scenarios we used the built in Performance Monitor application; an application included in the Windows Server 2012 operating system. Performance Monitor runs on the hypervisor host machine and is able to monitor the parameters of a local or remote computer in real time. It can handle more than 350 counters. The monitored data can be saved in files or can be graphically displayed.

For test measurements we initialized a data collector set which included the following counters:

– Hyper-V Hypervisor Logical Processor – % Total Run Time, - representing the total processor usage including the CPU load of the host operating system and also the CPU load of VMs,
– Memory – Committed Bytes – representing the total amount of memory used both by the host operating system and the VMs, in bytes,
– LogicalDisk – Average Disk Queue Length – representing the HDD queue in units.

These parameters were saved in the test measurement log files with a period of 1 second.

The same parameters were monitored in the VMW are ESXi vSphere scenario, but here they have different names:

– *CPU utilization (%),*
– *RAM allocation (bytes),*
– *Logical disk latency (ms) – instead of disk queue length.*

The XenServer offers a limited possibility of monitoring in comparison to Microsoft Hyper-V and VMWare vSphere. In this scenario we could monitor only 2 parameters, the:

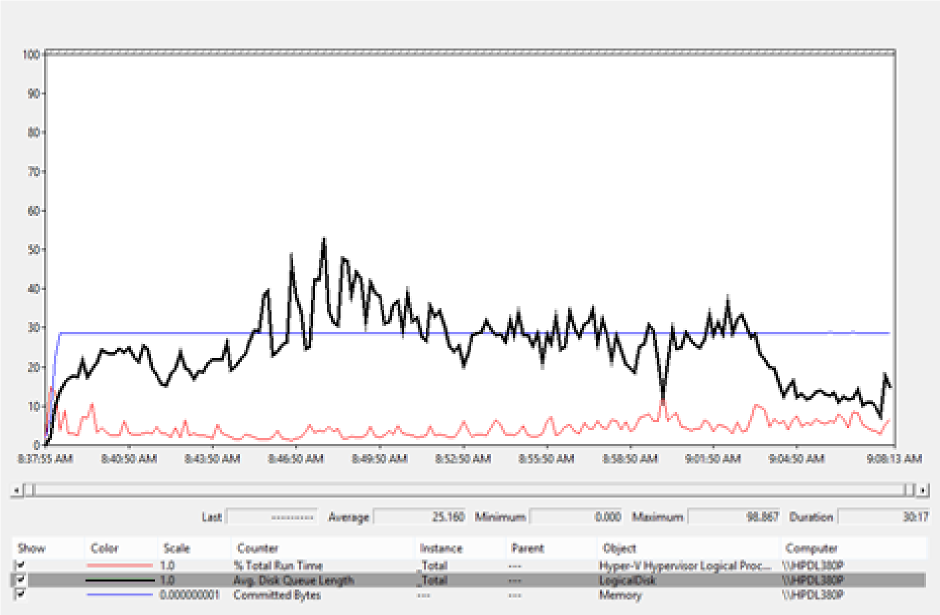
- Average CPU *(%)*
- Used memory *(bytes)*.

CPU utilization test measurements using the Microsoft products are reported in conference paper [1]. The load for these tests was produced artificially on each VM using the CPU Stress application [11].

CPU utilization tests with realistic data are presented here. Two realistic CPU load tests were performed: virtual machine boot test and Matlab application start test. The tests were repeated in all test environments: Hyper-V, ESXi and XenServer based virtualization and also RDS.

4.1. Microsoft hyper-V and RDS platform test results

Fig. 5 shows the results of the Virtual Machine Boot test in the Hyper-V based scenario. The test began with the startup of the virtual machines (VMs) and ended when all of them had finished booting and remote desktop connections were engaged. We started all the VMs at once. The server had 3 hard drives installed, so we made a RAID 0 array for best performance. All 3 parameters logged are presented: CPU usage marked in red, RAM allocation in blue and disk queue highlighted in black.



Last	----------	Average	25.160	Minimum	0.000	Maximum	98.867	Duration	30:17

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
☑	🟥	1.0	% Total Run Time	_Total	---	Hyper-V Hypervisor Logical Proc...	\\HPDL380P
☑	■	1.0	Avg. Disk Queue Length	_Total	---	LogicalDisk	\\HPDL380P
☑	🟦	0.000000001	Committed Bytes	---	---	Memory	\\HPDL380P

Figure 4: CPU usage, RAM allocation and disk queue variations measured during Virtual Machine Boot test in the Hyper-V scenario

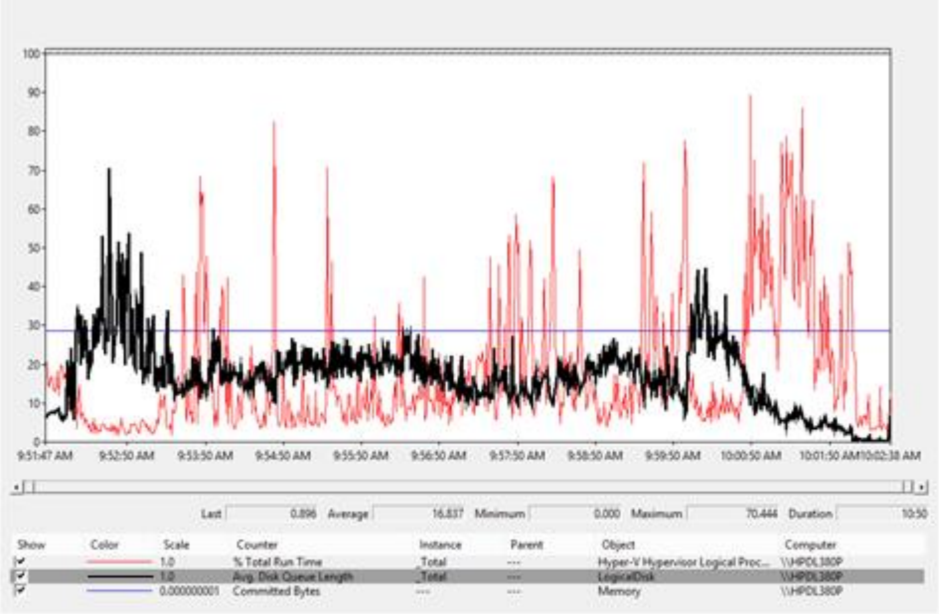
The most important parameter that we can observe on the above chart is the Average Disk Queue Length (highlighted in black). This parameter shows how many disk commands were waiting in queue, to be executed. The average value is around 25-30. The maximum value should be the number of hard drives in the server, which was 3 in our case. This means that during the test, there was a severe performance bottleneck which slowed down the booting of the virtual machines. The performance bottleneck arose from the poor performance of the storage subsystem. The resulting performance bottleneck made the whole infrastructure almost unusable.

The committed bytes (RAM) performance counter(blue) shows an abrupt increase shortly after the start of the test, resulting from the allocation of RAM for every virtual machine at startup. The virtual machines featured static RAM allocation.

The CPU utilization performance counter shows a low average value.

This also confirms the storage bottleneck, because the CPU has to wait until the storage subsystem returns the desired information.

In the second test we measured the time taken for the Matlab software platform to start. We have also monitored the performance counters mentioned above. Measured parameters are presented in Fig. 6. The test started with the first virtual machine giving Matlab the start command, followed by the same thing replicated on all 12 VMs. The test finished when all the Matlab programs were running on all 12 VMs and the “Ready” message was displayed in the status bar.



Last 0.896 Average 16.837 Minimum 0.000 Maximum 70.444 Duration 10:50

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
☑	🟥	1.0	% Total Run Time	_Total	---	Hyper-V Hypervisor Logical Proc...	\\HPDL380P
☑	⬛	1.0	Avg. Disk Queue Length	_Total	---	LogicalDisk	\\HPDL380P
☑	🟦	0.000000001	Committed Bytes	---	---	Memory	\\HPDL380P

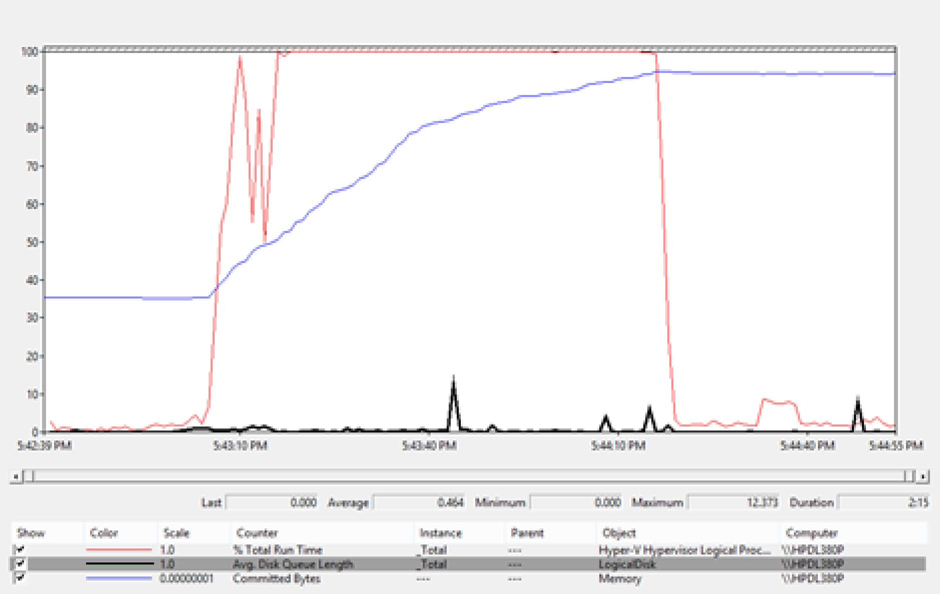
Figure 5: CPU usage, RAM allocation and disk queue variations measured during the Matlab startup test in Hyper-V scenario

Because the Matlab platform start time is shorter than the boot time of the VMs, we can observe more aggressive changes on the chart above. The Average Disk Queue Length counter (highlighted in black)still returned high values, while CPU utilization remained relatively low, indicating a severe performance bottleneck.

The same tests, repeated in the RDS scenario, reveal better RDS performances than those of the virtualization-based WM delivery solution.

The main difference is that there was no more virtualization and just one operating system running on the server. Only the applications started by the remote desktop users are multiplied. That is why there is a significantly lower memory amount in use. In comparison to the almost 20 GB RAM used for the host operating system and the 12 virtual machines in the hyper-V scenario, we see just around 10 GB RAM used in the RDS based test.

Fig. 6 shows much lower disk queue length values: the average is 0.464 < 3. Additionally, much more processor occupancy is reached because the logic drive bottleneck doesn't appear. High CPU utilization and low Average Disk Queue Length numbers indicate no logical disk performance bottleneck.



Last 0.000 Average 0.464 Minimum 0.000 Maximum 12.373 Duration 2:15

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
☑	🟥	1.0	% Total Run Time	_Total	---	Hyper-V Hypervisor Logical Proc...	\\HPDL380P
☑	⬛	1.0	Avg. Disk Queue Length	_Total	---	LogicalDisk	\\HPDL380P
☑	🟦	0.00000001	Committed Bytes	---	---	Memory	\\HPDL380P

Figure 6: CPU usage, RAM allocation and disk queue values during Matlab startup test in RDS based scenario

4.2. VMWare ESXi platform test results

During testing of the ESXi virtualization environment, there were no performance counters monitoring the server, because Windows Server 2012 R2 wasn't installed. Instead, the virtual machines ran directly in the virtualization environment so only the built-in ESXi performance metrics were available.

The Virtual Machine Boot test had the same implementation as in Hyper-V.

The test began with all the VMs started at once, and finished when all were responsive and engaged in RDP connections.

Surprisingly, while using Hyper-V, the boot time of the VMs was around 40 minutes: in the ESXi implementation it only took 3 minutes to fully boot up.

The ESXi management VCenter Server was running on a different server station. In this way the management server was not an extra load for the server running the VMs.

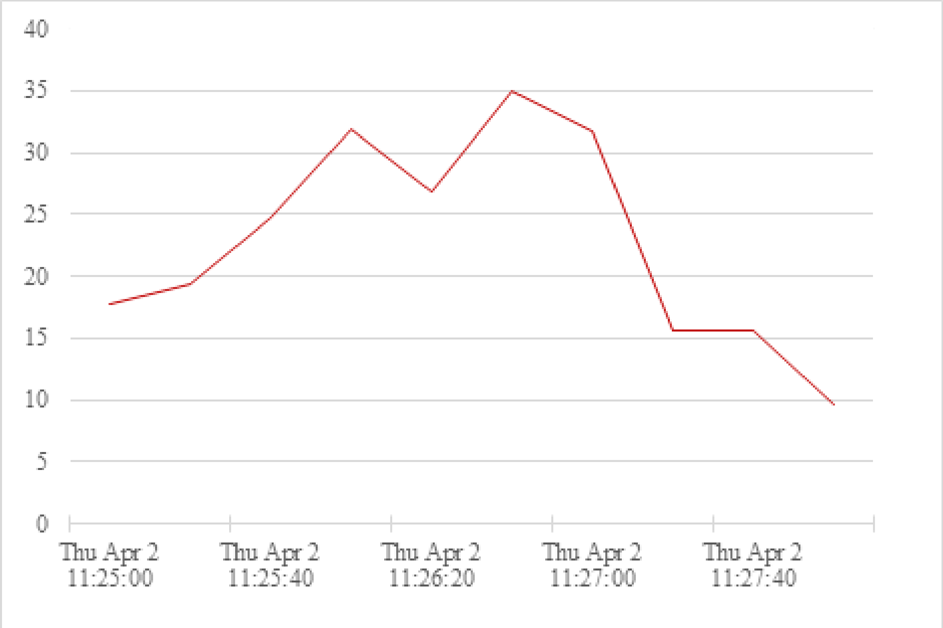


Figure 7: ESXi 6.0 Virtual machine boot test - CPU utilization (%)

Fig. 8 shows memory allocation test results and Fig. 9 shows logical disk latency test results for the ESXi VM boot test. In both tests ESXi slightly outperforms hyper-V.

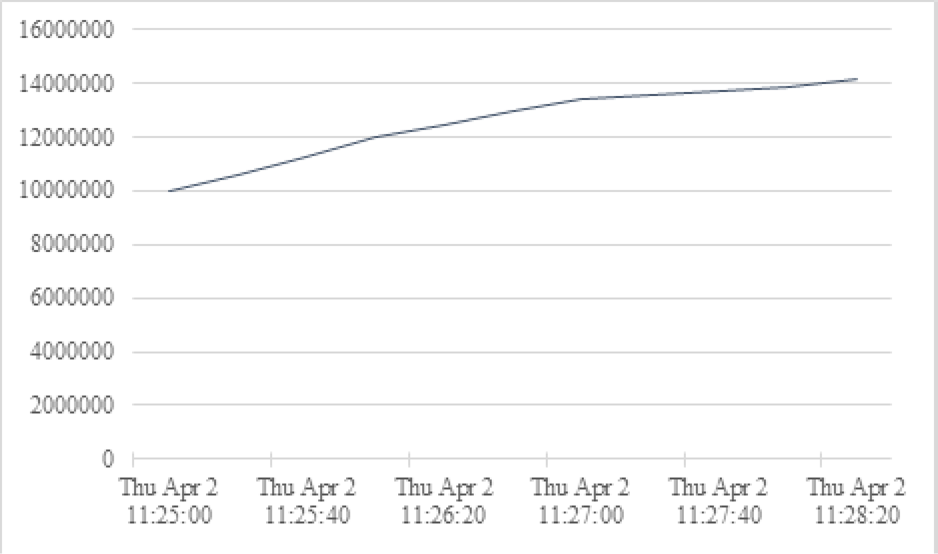


Figure 8: ESXi 6.0 Virtual machine boot test – RAM allocation (bytes)

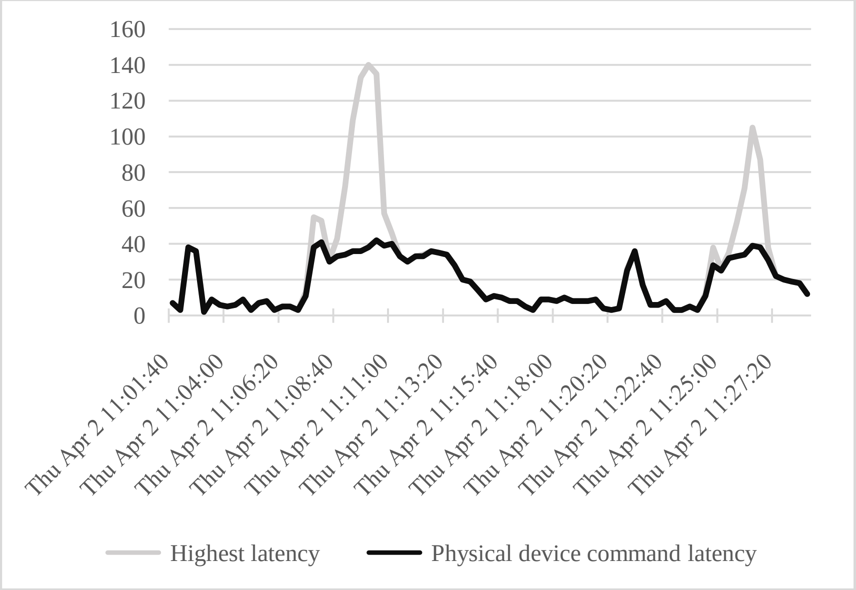


Figure 9: ESXi 6.0 Virtual machine boot test- Logical disk latency (ms)

In the following three figures: Fig. 10, Fig.11, and Fig.12are presented the CPU utilization, memory allocation and logical disk latency test results for the ESXi Matlab platform startup test.

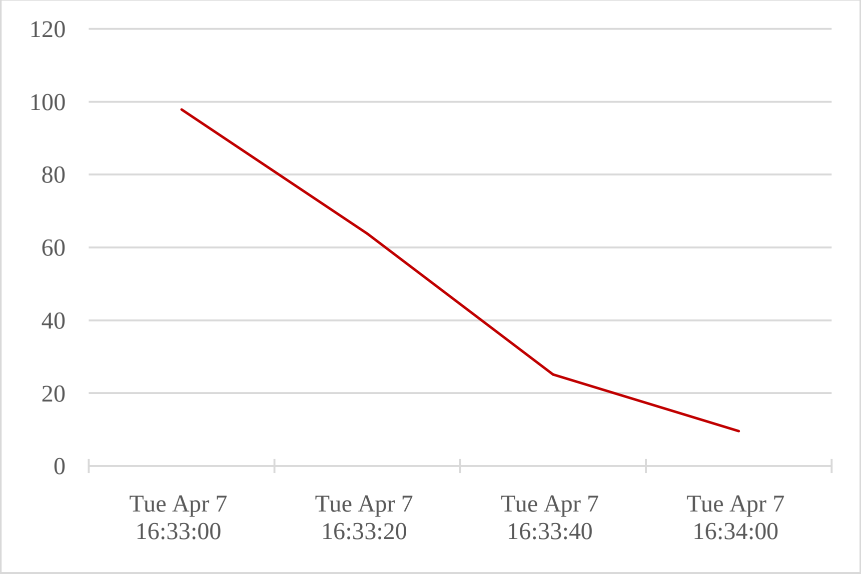


Figure 10: ESXi 6.0 Matlab startup test - CPU utilization (%)

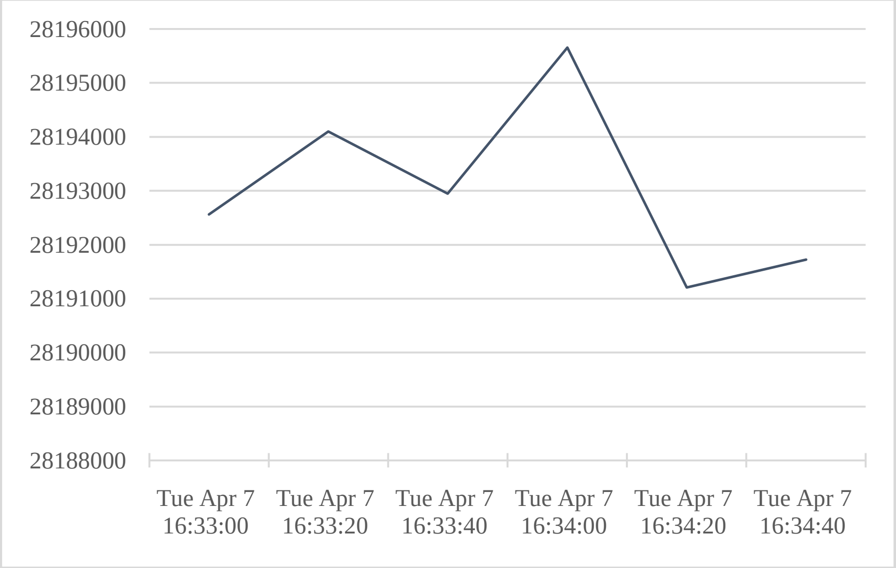


Figure 11: ESXi 6.0 Matlab startup test - RAM allocation (bytes)

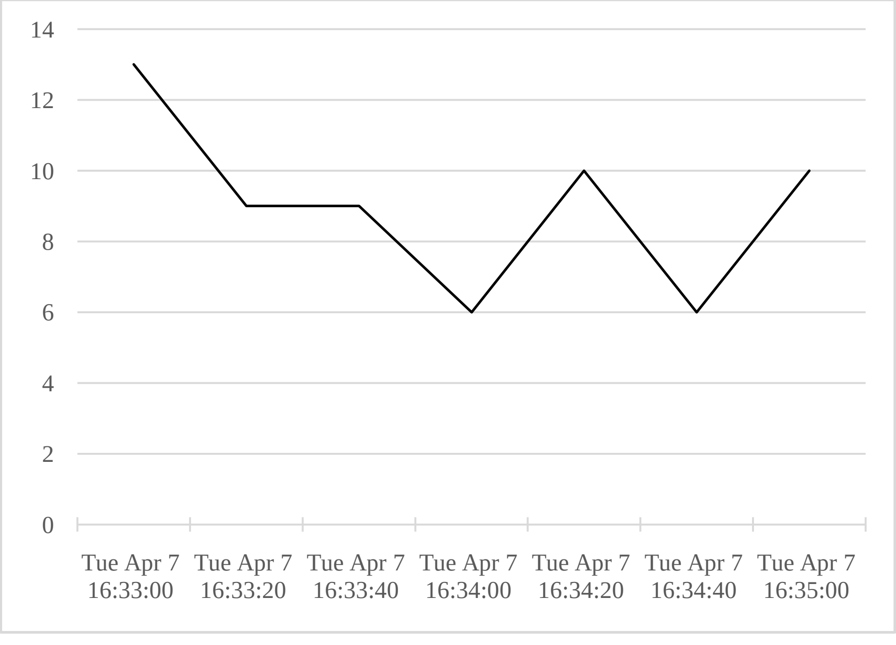


Figure 12: ESXi 6.0 Matlab startup test-Logical disk latency (ms)

4.3. XenServer platform test results

For the XenServer scenario, the procedure was then repeated the same way as for the Hyper-V and ESXi tests. The total run time for XenServer VM boot test was 4 minutes and 35 seconds. There were no disk latency or queue metrics provided in XenServer to measure logical disk performance. However, looking at the CPU utilization chart and the runtime of the test, we can deduce that a severe performance bottleneck, such as that seen with Hyper-V, is not present.

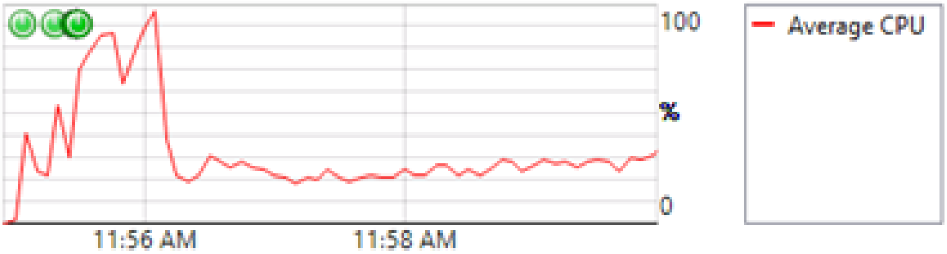


Figure 13: XenServer 6.5 Virtual Machine boot test - CPU utilization (%)

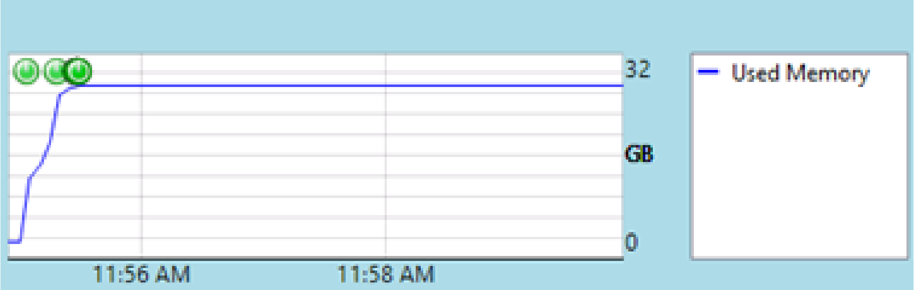


Figure 14: XenServer 6.5 Virtual Machine boot test - RAM allocation(GB)

The Matlab platform startup test was conducted identically to that of the Hyper-V and ESXi. Total runtime of the test was 2 minutes, 7 seconds. Test results are presented in Fig. 15 and Fig. 16.

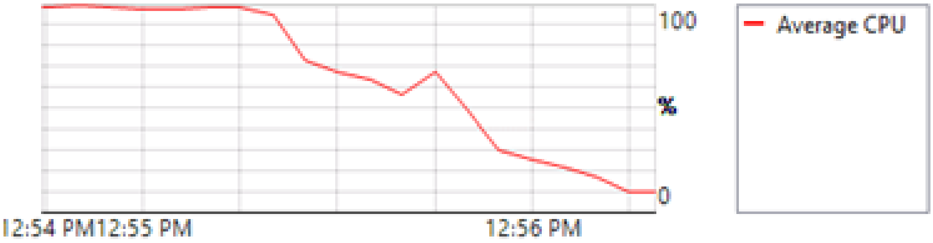


Figure 15: XenServer 6.5 Matlab startup test - CPU utilization (%)

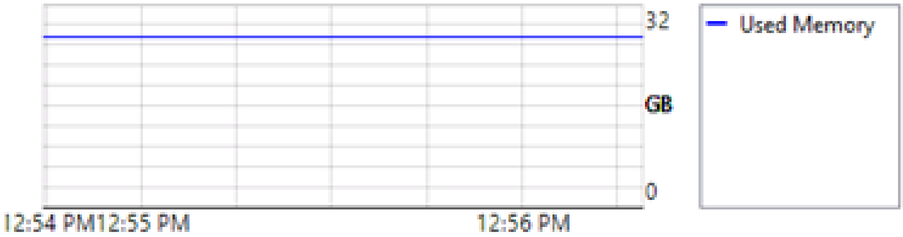


Figure 16: XenServer 6.5 Matlab startup test - RAM allocation (GB)

As static memory allocation was used for the VMs, the entire memory for VMs was allocated at startup.

5. Conclusions

In this paper we have presented experiments conducted in order to compare platforms that can be used for the implementation of a virtual classroom and an office IT infrastructure, and to identify the best solution among them.

We benchmarked different methods for delivering virtual machines capable of being accessed remotely by thin-clients and we performed tests with different thin-clients such as dedicated thin-clients, single board computers, smart phones and tablets. We have found that alternative thin-clients are a very good solution for commercially available thin-client replacement. These single board computers are flexible, they can run Linux-based operating systems, can handle Remote Desktop Protocol (RDP) for connecting to the server, the acquisition costs are an order of magnitude cheaper than those of the commercially available thin-clients and they also have lower power consumption.

There are several virtualization platforms available for providing remote desktops. We have benchmarked some of these platforms to see how they actually perform and have made a comparison between the acquisition and maintenance costs as well as relative performance.

We built three test environments for testing four scenarios. The first one was based on Microsoft products and tested hyper-V hypervisor and Remote Desktop Services, the second was based on VMW are ESXi and the third one was based on the Xen Server platform. For each platform we measured CPU usage, memory allocation and disk management parameters to evaluate the whole system performance. Tests were performed in a real network environment and using a real load.

We also studied the total cost of acquisition and the license fees for each technical solution.

Our main conclusion is that Microsoft Remote Desktop Services out-performs the virtualization based solutions, but incurs high license fees. From the virtualization solutions tested, the VMW are ESXi based one seems to be the most reliable choice.

We highlight two problems encountered during our tests. The first is the high-level of memory usage in the case of virtualization based solutions compared to that of RDS. The second aspect is the bottleneck identified in the logical disk management. High-level disk queue was detected in the case of the hyper-V hypervisor based scenario. Both problems lead to a limitation of the number of delivered VMs.

Disk queue problems were not identified in the tests performed with the other two virtualization platforms. This can be explained by the fact that the management servers were running on a different server station, not on the one which delivers the VMs.

Acknowledgements

This research and its publication have been supported by the European Union and Hungary and co-financed by the European Social Fund through the project TAMOP- 4.2.2.C-11/1/KONV-2012-0004 - National Research Center for Development and Market Introduction of Advanced Information and Communication Technologies.

References

[1] Darabont Ö., Kiss K. J., Domokos J., “Performance Analysis of Remote Desktop Virtualization based on Hyper-V versus Remote Desktop Services”, in *Proceedings of the 5th International Conference on Recent Acievements in Mechatronics, Automation, Computer Science and Robotics (MACRo 2015), Tîrgu Mureș, Romania*, 2015, pp. 125-135.

[2] Bari F., Boutaba R., Esteves R., Granville L. Z., Podlesny M., Rabbani G., Zhang Q., Zhani M. F., “Data Center Network Virtualization: A Survey”, in *IEEE Communications Surveys & Tutorials*, Vol. 15, No. 2, pp. 909-918, 2013.

[3] Rodríguez-Haro F., Freitag F., Navarro L., Hernández-Sánchez E., Farías-Mendoza N., Guerrero-Ibanez J. A., González-Potes A., “A summary of virtualization techniques”, in *Proceedings of the 2012 Iberoamerican Conference on Electronics Engineering and Computer Science*, Procedia Technology 3, 2012, pp. 267 – 272.

[4] Miseviciene R., Ambraziene D., Tuminauskas R., Pažereckas N., “Educational Infrastructure Using Virtualization Technologies: Experience at Kaunas University of Technology”, in *Informatics in Education, Vol. 11, No. 2,* pp. 227-237, 2012.

[5] Cimen C., Kavurucu Y., Aydin H., “Usage Of Thin-client / Server Architecture In Computer Aided Education”, *TOJET: The Turkish Online Journal of Educational Technology*, volume 13, issue 2, pp. 181-185, 2014.

[6] Darabont Ö, Kiss K. J., Domokos J., “Cost effective computer infrastructure” in *Proc. of the XXIVnd International Conference on Computers and Education (SZÁMOKT 2014), Székelyudvarhely, Romania,*, 2014, pp. 144-147.

[7] Domokos J., Kiss K. J., Darabont Ö., “Economic and technological analysis of the development of a virtual machine room”, in *Proc. of the Annual Scientific Conference of*

the Hungarian National Coordinating Center for Infocommunications ASCONIKK, Veszprém, Hungary, 2014.

[8] Calyam P., Berryman A., Lai A.M., Honigford M., “VMLab: Infrastructure to Support Desktop Virtualization Experiments for Research and Education”, in *VM-ware Technical Journal*, 2012.

[9] Berryman A., Calyam P., Honigford M., Lai A. M., “VDBench: A Benchmarking Toolkit for Thin-client based Virtual Desktop Environments”, in *Proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Science*, 2010, pp. 480-487.

[10] Kaur D., Bhathal G. S., “Thin Client Cloud Computing: An Overview”, *International Journal for Multi Disciplinary Engineering and Business Management*, Volume-2, Issue-1, pp. 8-10, 2014.

[11] MSDN blog Tools to simulate CPU / Memory / Disk Load, Windows sysinternals CPU Stress application: *http://blogs.msdn.com/b/vijaysk/archive/2012/10/27/tools-to-simulate-cpu-memory-diskload.aspx.*

[12] Pék G.,Buttyán L., Bencsáth B.,“A Survey of Security Issues in Hardware Virtualization”, *ACM Computing Surveys (CSUR)*, Volume 45 Issue 3, pp 40:1-40:34, 2013.

[13] The FreeRDP software webpage: http://www.freerdp.com/

[14] Sahoo J., Mohapatra S., Lath R., “Virtualization: A Survey On Concepts, Taxonomy And Associated Security Issues”, *in Second International Conference on Computer and Network Technology*, 2010, pp. 222–226.

[15] Nieh J., Yang S. J., Novik N.,“A Comparison of Thin-Client Computing Architectures”, *Technical Report CUCS-022-00, Network Computing Laboratory, Columbia University,* 2000.

[16] Lai A. M., Nieh J., “On the Performance of Wide-Area Thin-Client Computing”, *ACM Transactions on Computer Systems,* Vol. 24, No. 2, pp. 175–209, 2006.

[17] Baratto R. A., Kim L. N., Nieh J., “THINC: A Virtual Display Architecture for Thin Client Computing”, *in Proceedings of the twentieth ACM symposium on Operating systems principles*, 2005, pp. 277-290.

[18] Jae Yang S., Nieh J., Naomi N., “Measuring Thin-Client Performance Using Slow-Motion Benchmarking”*, in ACM Transactions on Computer Systems*, Volume 21, Issue 1, pp. 87-115, 2003.