

# CREATING VIRTUAL ENVIRONMENT FOR EDUCATIONAL PURPOSES OF SCHOOLS AND UNIVERSITIES

Bartłomiej Kacprzak, Piotr Goetzen, Alina Marchlewska

IT Institute, Academy of Management, Lodz, Poland  
*bercik1337@gmail.com, (goetzen, amarchlewska)@swspiz.pl*

## Abstract

The objective of this paper is to present how to prepare and setup the virtual machine environment on PC to improve speed, quality and overall feeling of education process with reducing costs of hardware and software. This project is a response to problem arising in most educational agencies and training centers. It is a result of continuously growing amount of data to process in today's world. What is more, the range of subjects to teach widens everyday because of market needing versatile specialist on every level. As a result, the students nowadays have to get to know variety of systems, software and ideas in short period of time.

**Key words:** virtualization, database, VMware, ESXi, Linux, education, reducing costs, saving time

## 1 Introduction

As the years pass by, computer technologies and IT services are common to people. IT market is in a need of good specialists to provide needed service. Educational institutions try to cope with that need implementing constantly growing range of specializations according to market valuation. This requires wide range of hardware (different types of computers: screen sizes, CPU and GPU power etc.), software (application for networking, graphic editing, audio editing, video editing, system administration, software development) and operating system configuration. All tasks performed by students are usually located in different rooms because all mentioned above tasks require different types of system access. If, for example, soon-to-be system administrator re-configures NICs<sup>1</sup> revoking access to Internet access on this station, this would impact all other students using this station. Schools and universities are trying

---

<sup>1</sup> Network Interface Card

to fix this problem by acquiring more workstations with specialized software (e.g. DeepFreeze) with results of rising costs.

Today's world is facing flood of information. Systems process more and more data because of growing population and ease of access to Internet. Increasing number of customers forces service providers to add more servers and storage, build new facilities and lease greater Internet connection. All that results in greater electricity bills, more space to equip and higher maintenance costs. Is it really the only path? Thankfully to virtualization - no. This technology can reduce hardware needed for specific job. Because of system consolidation, servers utilize most of system resources which leads to less hardware needs[3].

So what actually is virtualization? To put it as simple as possible - it's mechanism that separates software (for instance whole operating system) from hardware. Thanks to that "encapsulation" it is possible to run multiple OS's at this same time in one physical machine.

The very concept of virtualization is almost half century old. First working virtual machine was created in early 60's. Even back at that time system were fully functional when ran at the same time. Since that day, personal computers gained great amount of computing power. Thanks to that, anyone can run Virtual Machine (VM) on his PC. Over past few years significant growth of virtualization market can be seen. Reason for that mostly is previously mentioned computing power. On software side of things first steps were made by MIT students. They created system called CTSS (Compatible Time-Sharing System) the ancestor of today's z/VM. PC's virtualization market was pushed forward by VMware company. Started in 1998, their products really popularized virtualization on small and medium sized corporations as well as home oriented solutions[4].

## **2 Education**

Different types of schools nowadays have different types of path for their students with a lot of topics to go through[4,5,6]. Educational system is not coherent. This can be easily observed by looking at computer rooms where computer systems are dedicated usually for one task only. This results in wasting system resources that could be used in better manner. This section of paper will describe differences in educational programs and as a result of that – different specifications of systems required for those tasks.

## **2.1 Primary schools**

During primary schools main topics of IT education are:

- Basics usage of computer
- Personalization of operating system
- File and folder operations
- Text editor usage [6]

## **2.2 High school and middle school**

High schools and later on middle school implemented education of basic programming languages. Those were most commonly PASCAL and BASIC. Each student workstation had to be equipped with programming environment (code editor, compiler, etc). Also basics of HTML are quite popular during this time of education [7]

## **2.3 Universities**

One of the least steps of education tree. There, IT knowledge is widened on multiple specializations. Most common are:

- Computer architecture
- Programming
- Computer networking
- Operating system administration
- Computer graphics [9]

# **3 DESCRIBING THE PROBLEM**

## **3.1 Cases**

As mentioned before, differences in educational programs project on need for specific hardware. The following cases will show some of examples of these issues.

### **Case one:**

For purpose of teaching Linux administration specific tasks, teachers require numerous workstation and server configurations for teaching purposes. Those machines have to be equipped with wide verity of peripherals. For some tasks diskless station with 32MB of RAM will be enough. For others multicore server with gigabytes of RAM is needed.

**Case two:**

In the need of graphic and video editing, students have to use very powerful graphic stations, preferably with multicore CPU, reasonable or even multi-processor GPU and the display of significant size for better editing.

**Case three:**

Programmers during their work spend most of time on reading documentation, analyzing existing code. That requires minimum system resources. Huge CPU and IO utilization is required only to compile the program.

### **3.2 Preparations and tests**

Several test were done. The main aim of tests is to measure the time needed to copy the template of the system onto different locations:

- Test 1 Replication of one copy of the template
- Test 2 Copying the template between the physical disks
- Test 3 Double replication test
- Test 4 Parallel replication
- Test 5 Remote storage template copy
- Test 6 Final benchmark

For purpose of this paper, testing environment was created. It took the following steps to complete:

- a. Hypervisor installation[1]
- b. Hypervisor configuration[3]
- c. Creating teacher's system
- d. Creating student's template
- e. Replication tests
- f. Automation of replication process

VMware ESXi 4.0.0 build 208167 hypervisor software was chosen for this project. The following hardware was used:

- CPU Intel Core 2 Quad Q8200 2,33GHz x 4
- Memory 6GB DDR2
- NIC Intel PRO 1000MT
- HDD Hitachi 250GB SATA2

After completing tasks mentioned in steps A up to D (those steps won't be discussed in this paper) system is ready for replication tests.

In the presented example, we prepared the student's system that takes up 5GB of space on hard drive. When using build in ESXi file transfer tool time needed for data transfer on 100 Mbps network is represented in table 1

**Table 1.** Calculations

$5\text{GB} = 5 * 1024\text{MB} = 5120\text{MB}$
$5120\text{MB} + 20\% = 6144\text{MB}$
$6144\text{MB} / 7\text{MB/s} = 877\text{s}$

Assuming that speed of network link will reach 7MB/s it will take 877 seconds to transfer file to or from host with average encryption overhead of 20% [8]

Using build in tool for replicating images is easy to use and ready out of the box, however it has some major disadvantages:

- In order to copy multiple instances from computer to ESXi server, all images should be already created on computer hard drive and grouped in one folder. It is impossible to issue command multiplying one image n-times.
- During data transfer from/to ESXi host client computer network link has to be up. If the link goes down it will result in transfer brake and will force teacher to start over.
- When the tool is used from remote location with low bandwidth connection, time needed for transfer rises dramatically.

To solve these issues it is best to use SSH connection to ESXi host and make template replications using `/bin/ash` shell. In order to enable this hidden feature one must have physical access to the virtual machine host server. By pressing ALT+F1 keys hidden console will be activated. Within the console the command `unsupported` should be typed in (will not be visible) and after pressing enter key, system will ask for root password. Providing these credentials will result in instant access to shell (Figure 1). From there, `/etc/inetd.conf` file must be edited. Inside the line containing `#ssh` has to be uncommented.

After ESXi reboots, SSH access on default port 22 will be accepting connections.

```
ESXi 4.0 http://www.vmware.com
Copyright (c) 2007-2009 VMware, Inc.

Password:
You have activated Tech Support Mode.
The time and date of this activation have been sent to the system logs.

Tech Support Mode is not supported unless used in consultation
with VMware Tech Support.

VMware offers supported, powerful system administration tools. Please
see www.vmware.com/go/sysadmintools for details.

Tech Support Mode may be disabled by an administrative user.
Disabling requires a reboot of the system. Please consult the ESXi
Configuration Guide for additional important information.

# _
```

Figure 1. Hidden ESXi shell

### 3.2.1 Test 1 Replication of one copy of the template

As calculations show in Table 1, time needed to replicate one copy of template is 877 seconds. The following test will show how much time is needed to copy the same amount of data using SSH within ESXi box to the same physical hard drive.

Table 2. ssh copy

```
~ # cd /vmfs/volumes/datastore1/
/vmfs/volumes/4e4318fb-eaab2e19-f6d9-0002b3ee588a # time cp -r
szablon_sieciowcy/ sieciowcy1/

real 10m 40.19s
user 0m 53.46s
sys 0m 0.00s
```

Table 2 presents result of Test 1. Time needed for operation was significantly reduced. Time savings gained by this solution will multiple n-times for each replication.

### 3.2.2 Test 2 Copying the template between the physical disks

It is possible to connect another physical hard drive and perform copy between physical devices. To do that, after connecting hard drive to ESXi server, HDD has to be initiated and used as Datastore. In panel *Configuration>Storage>Add Storage* newly added device has to be selected.

Datastore is now ready for use. Test 2 will copy template from old *datastore1* to newly created *datastore2*

**Table 3.** Replication to another HDD

```
/vmfs/volumes/4e4318fb-eaab2e19-f6d9-0002b3ee588a # time cp -r
szablon_sieciowcy/ /vmfs/volumes/datastore2/sieciowcy3
real 6m 45.37s
user 0m 51.45s
sys 0m 0.00s
```

Executing Test 2 also reduced time needed for operation. With this setup results are even more vivid. Total time needed for copying the file was reduced almost by half.

### 3.2.3 Replication automation

Automating this environment will provide multiple benefits:

- Reduce time wasted by human activity
- Huge workload can be scheduled for specific time (for instance at night)
- Scripts can be easily modified and tuned to new purposes

Scripts will be created in / path. They will be issued from within */vmfs/volumes/datastore1*. After creation, proper rights to execute have to be granted by issuing *chmod +x /skrypt\**

**Table 4.** Content of skrypt1.sh

```
for x in `seq 1 2`; do cp -r szablon_sieciowcy
/vmfs/volumes/datastore1/sieciowcy$x; done
```

**Table 5.** Content of skrypt2.sh

```
for x in `seq 3 4`; do cp -r szablon_sieciowcy
/vmfs/volumes/datastore2/sieciowcy$x; done
```

### 3.2.4 Test 3 Double replication test

Test 3 will consist of two actions, first, it will replicate twice the template to destination of *datastore1*, and after that replicate the same template to *datastore2* also two times. Commands and times of execution are show in table 6.

**Table 6.** skrypt1.sh and skrypt2.sh execution times

time /skrypt1.sh; time /skrypt2.sh
real 14m 13.91s
user 0m 0.00s
sys 0m 0.00s
real 19m 27.32s
user 0m 0.00s
sys 0m 0.00s

### 3.2.5 Test 4 Parallel replication

It is possible, to perform tasks simultaneously. It can be achieved by sending task to background using ampersand symbol at end of each command.

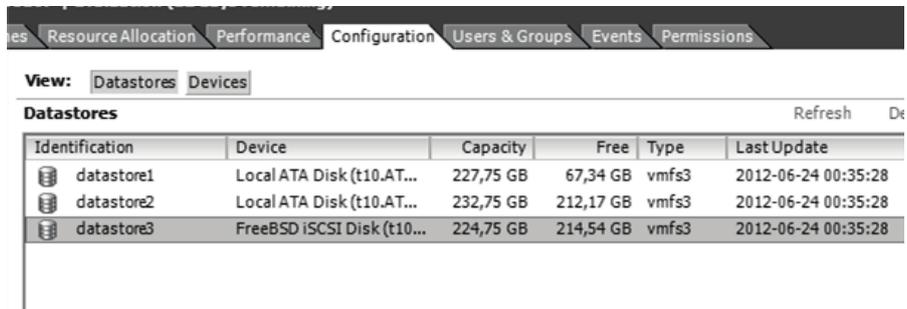
**Table 7.** Forking processes into background

/vmfs/volumes/4e4318fb-eaab2e19-f6d9-0002b3ee588a #
real 24m 32.21s
user 0m 0.00s
sys 0m 0.00s
/vmfs/volumes/4e4318fb-eaab2e19-f6d9-0002b3ee588a #
real 29m 48.87s
user 0m 0.00s
sys 0m 0.00s

As table 7 shows, concurrent replication needed more time to complete. Reason for that is because of cheap hard drive used - performance drops significantly when accessing similar area of data by two different threads. This is a good proof of why those benchmarks should be performed before actually making real production scripts.

### 3.2.6 Test 5 Remote storage template copy

Connecting remote resources such as NFS or iSCSI to ESXi host and using them as a remote datastore is another possibility that VMware ESXi provides. This option is very useful mainly because it skips limit of how many physical devices could be connected to ESXi host. Remote resources can be accessed from both intranet and Internet giving administrators variety options to choose from.



**Figure 2.** Remote iSCSI datastore

In this example remote iSCSI resource is used. It is located physically in the same network. Connection link speed between ESXi and iSCSI hosts is 1000Mbit/s.

One has to create *skrypt3.sh* and tune it to perform the same replication activity as shown in previous tests, except destination which, in this case, is *datastore3*. Execution time is shown in table 8.

**Table 8.** Replication to remote iSCSI datastore

<pre> /vmfs/volumes/4e4318fb-eaab2e19-f6d9-0002b3ee588a # time /skrypt3.sh real 11m 11.94s user 0m 0.00s sys 0m 0.00s </pre>
--

Using remote destination is another time saver, and again the amount of time saved is significant.

All tests above point that the most efficient copy operation can be achieved by queuing tasks one after another.

### 3.2.7 Test 6 Final benchmark

Last test will simulate complete preparation of environment for new group of students. For that following list of copies will be created:

- 9 copies on datastore1
- 9 copies on datastore2
- 19 copies on datastore3

**Table 9.** newclass.sh script

```
for x in `seq 11 20`; do cp -r szablon_sieciowcy
/vmfs/volumes/datastore1/sieciowcy$x; done
for x in `seq 21 30`; do cp -r szablon_sieciowcy
/vmfs/volumes/datastore2/sieciowcy$x; done
for x in `seq 31 50`; do cp -r szablon_sieciowcy
/vmfs/volumes/datastore3/sieciowcy$x; done
```

With all already created images current number of virtual systems will be equal to 50. That number should be enough for most classes and laboratories.

**Table 10.** execution of newclass.sh

```
vmfs/volumes/4e4318fb-eaab2e19-f6d9-0002b3ee588a # time
/nowaklasa.sh
real 4h 48m 24s
user 0m 0.03s
sys 0m 0.00s
```

Table 10 shows execution time of newclass.sh script that replicated 39 additional virtual machines (from 11 to 50). Those VMs are also ready to use right after replication finishes. As mentioned before, number of already crated systems should be enough to cover multiple groups of students for this subject.

Adding the system to the inventory is the last step after system replication. After choosing option Browse Datastore on each DS, newly created folder must be opened. Then, right-click on .vmx file and select *Add to Inventory* (Figure 3). Later, name for imported VM should be chosen. (Figure 4)



**Figure 3.** Adding VM to inventory

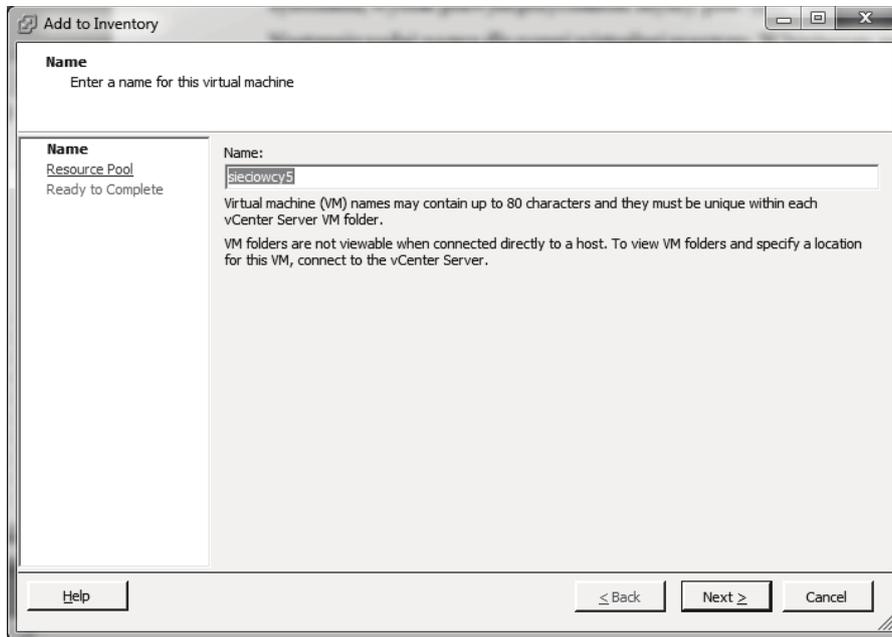


Figure 4. Naming new VM

#### 4 Example usage

After powering on all systems SSH connection can be established. To create new definition of SSH cluster, `.cssh` file must be edited. Portion of file is presented in Table 11

Table 11. `.cssh` file

```
clusters = sieciowcy
sieciowcy = root@192.168.1.110 root@192.168.1.111
root@192.168.1.112 root@192.168.1.113 root@192.168.1.114
root@192.168.1.115 root@192.168.1.116 root@192.168.1.117
root@192.168.1.118 root@192.168.1.119 root@192.168.1.120
root@192.168.1.121 root@192.168.1.122 root@192.168.1.123
root@192.168.1.124 root@192.168.1.125 root@192.168.1.126
root@192.168.1.127 root@192.168.1.128 root@192.168.1.129
root@192.168.1.130 root@192.168.1.131 root@192.168.1.132
root@192.168.1.133 root@192.168.1.134 root@192.168.1.135
root@192.168.1.136 root@192.168.1.137 root@192.168.1.138
root@192.168.1.139 root@192.168.1.140
```

Next step is to physically connect local system to cluster by issuing `cssh sieciowcy --options "-i .ssh/uczen"`. After making connection, system will ask about unknown certificate. It is necessary to accept it by typing "yes". Confirming certificate by typing "yes" is displayed on Figure 5[2]

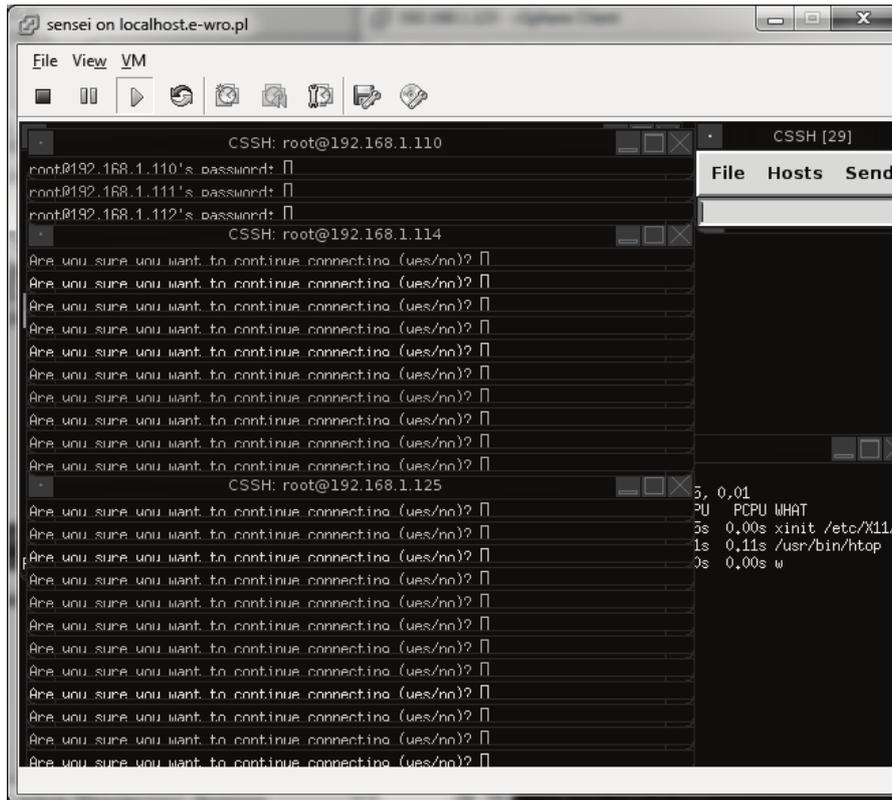


Figure 5. Initiation of SSH connection

## 4 Summary

Tests performed measured time needed for replication of exemplary system. Measurements showed that slowest method of replication is the one where source and destination datastores are the same. Slightly faster solutions is that with different source and destination datastores. The fastest time for task completion was the one that used remote datastores connected via iSCSI protocol. It is worth to point out, that components used in those test were rather low-class, where low cost significantly impacts performance (but low cost is the very important parameter for school budgets). Along with greater budgeted comes more time saving. Also, given that values shown in examples

are rather low, environment fulfills its task. Time needed for copying was shrunk from 15 minutes down to 10 at worst scenario. If one needs to install and configure the system from scratch it takes normally hours. Reducing the installation to 10 minutes is significant. Ease of maintaining the infrastructure will result not only in reducing time for preparing classroom for subject, but also improve , thanks to system centralization , all housekeeping/cleanup tasks such as removing unnecessary VMs. After having implemented this solution, effects are noticed immediately on first use. Infrastructure size is no longer an important factor in this scenario, because this solution benefits when used for either 5 or 500 systems. Moreover, implementation brings more benefits described below.

#### **4.1 Financial benefits**

One of the benefits of this system is reduction of financial costs. They derive directly from reducing used hardware. Instead of constant hardware upgrades in classrooms, which often are equipped with many computers it is possible to upgrade just one central server.

Secondly, moving high load from clients to one centralized server results in no more need for powerful hardware. For remote administration any type of hardware will be enough (both Windows and Linux). What is more, all classroom computers can be sold and superseded with brand new, cheap “Thin Clients”. Another option is using embedded computers like Arduino or Raspberry PI that were meant to be small and cheap minicomputers with super-low power consumption reaching 2W - 5W at high load.

All that transformations lead into money savings. Electricity spending is reduced right after system implementation.

#### **4.2 Time benefits**

Automation of tasks will have impact on time needed for preparing classes environment and also reduce time needed for tasks during lectures and labs. As effect teacher gains more time that could be spent for focusing on subject instead of struggling with hardware problems

#### **4.3 Management**

Centralization of management simplifies control over infrastructure. It can provide monitoring facilities for resources used, and prevent from over or under utilization of equipment. Implementing additional monitoring tools and logging solutions provide teacher ways to ensure the system security.

#### 4.4 Flexibility

Flexibility of system in both hardware and software aspects guarantees wide possibilities of further development. Possibility of main system reconfiguration is essential if system is going to be used for variety of subjects. Physical development of the server will benefit in greater savings mainly because of consolidation of many VMs into one server and at the same time reduce time needed to complete any operation. Figure 6 shows the chart of time needed for copying 5GB virtual machine image related to disk speeds of different sort. With proper budgeted, time needed for copying can be reduced over ten times. Multiplying it by n-times number of disks will result in even less time needed for classroom preparation.

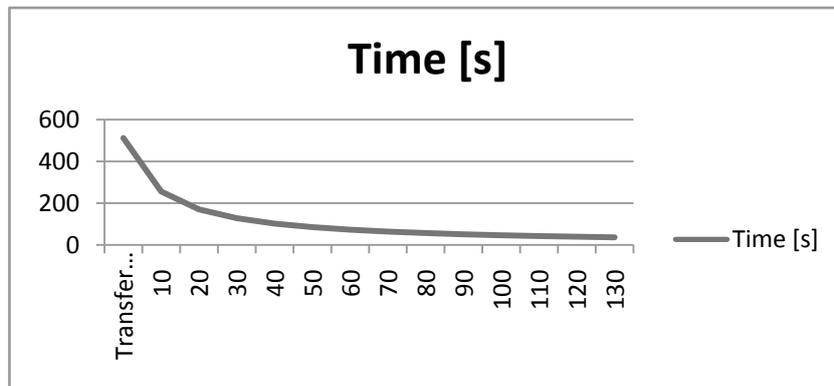


Figure 6. Disk performance

#### 4.5 Meeting the objective

Environment created and presented in this paper meets all goals. Thanks to minimal financial input needed to implement presented solution, ease of management, better hardware utilization and money savings coming from less hardware usage were achieved. Time saved can be spent on subject/topic matter instead of wasting it on classroom preparation. Possibility of immediate logon to student systems gives teacher ability to quickly diagnose and fix problems, or mistakenly configured system. Also snapshot functionality can reduce time needed for systems maintenance and provide safe and quick facility for testing new features. Placing multiple systems onto one server is also useful when devices (classrooms) have to be moved from one place to another. There is no need for moving tens or even hundreds of devices. All that is reduced to moving one server. Any costs regarding hardware failures are also minimalized. Periodic fan, PSU or disk swapping that before used to take

significant amount of money are now independent regardless of how many systems are moved to ESXi host. Possibility of remote access to VMs for both teachers and students at home is another important feature.

#### **4.6 Possibilities of reusing solution**

Improving educational process in any type of educational facilities was one of objectives of this environment. Universality of this solution gives possibility of using it in:

- companies, as a training environment. Useful especially for new employees. Facility similar to real production environment is best solution for teaching newcomers, and yet because of not affecting the production servers any mistakes won't result in downtime or loss of data.
- companies, as testing and development environment. Possibility of creating one to one copy of production environment and testing there new functionality is an great asset. It gives opportunity to completely test new features before implementing them on production servers.
- home, as environment for learning networking and operating systems. Functionality of this solution guarantees possibility of learning different types of tasks. From basics of computer networking, analyzing how it works, up to its reconfiguration using complicated IT solutions, e.g. using HA and load balancing. All these tasks could be achieved before, but for a cost of really expensive architectures.
- courses and presentations, as lab environment. In IT branch some problems have to be shown, or require specific methods of presentation used by speaker (live presentation). In most cases regular slide presentation is not enough especially if problem only occurs during specific scenario/situation. Trying to replicate those situations in regular hotel network is extremely difficult if not impossible. Earlier prepared environment can be easily installed on ESXi host and presented on location, or remotely, in lab environment, via Internet connection.
- applications, for testing them. Great number of systems running on ESXi host and ability to simultaneously control them are very useful when simulating high traffic or specific behaviors on application or website. Thanks to this tool most common mistakes that come from improper resource estimation can be found even before passing software to tester group. As a result software preparation should require less time during creation process and what comes out of it will be resistant to some type of anomalies.

To wrap-up this paper, presented solution brings great amount of benefits, that impact both time and quality of work and education environment. All that can be achieved with almost no cost when using hardware already possessed, or with minimal investment depending on solution purpose. Lack of investment when implementing presented solution won't impact on any possible development of environment. At any time infrastructure can be upgraded to more power full one without the need of VM reconfiguration. It is also possible to completely move all systems to brand new ESXi box without any loss of data.

## References

1. Lowe S., 2010, Wiley, VMware vSphere 4 Administration, , pages 21-58
2. Hertzog R.; Mas R., 2012, *The Debian Administrator's Handbook*, Freexian SARL,
3. VMware Inc, 2011, *ESXi Configuration Guide ESXi 4.0*, VMware, Inc, pages 11-153
4. Kacprzak B, 2010, *Wirtualizacja w systemach X86 X64 i z/9*, SWSPIZ Diploma work, pages 3,5
5. Sysło M.M., *Myślenie komputacyjne. Informatyka dla wszystkich uczniów*, <http://www.up.krakow.pl/ktime/symp2011/referaty2011/syslo.pdf>, page 1
6. Stanecka B., Stanecki C. *Program Nauczania Informatyka W Szkole Podstawowej*, <http://www.stanpol.edu.pl/Pdf/Program-InformatykaSP.pdf>, site 13
7. Filinowicz E., *Program nauczania informatyki w gimnazjum, Informatyka dla Ciebie*, [http://www.nowaera.pl/index.php?option=com\\_docman&task=cat\\_view&gid=100036&Itemid=](http://www.nowaera.pl/index.php?option=com_docman&task=cat_view&gid=100036&Itemid=) site 9
8. Goleń, P., *EFS kontra TrueCrypt, czyli dlaczego szyfruję cały dysk* <http://wampir.mroczna-zaloga.org/archives/377-efs-kontra-truecrypt-czyli-dlaczego-szyfruje-caly-dysk.html>
9. Strona Główna. *Informacje dla wykładowców i Studentów*, <http://wazniak.mimuw.edu.pl/>