**Mariusz Żytniewski**
**Andrzej Sołtysik**
**Jarosław Kurpanik**

University of Economics in Katowice

# IT SYSTEMS VIRTUALISATION AND THE EXAMPLE OF ITS ACADEMIC USE

## Introduction

The development of information systems, and in particular their hardware layer, causes that the owned by company technical resources often surpass the requirements imposed by the software vendors. In the case of applying IT solutions unused computing power of servers can be optimized by the use of appropriate software to install multiple operating systems on one machine and share its resources across multiple software solutions. This reduces costs of maintenance, equipment, electricity, and facilitates the management processes throughout the environment. Importantly, this approach can be used not only for production environments, but also to test, when the software is part of the training environment and is used in the teaching process. Especially in the case of higher education, when systems used in teaching don't need to be used all the time but only at certain time intervals.

The purpose of this study is to present the concept of virtualization, its development, and to present a typology of several years of experience on the use of virtualization software solutions for teaching purposes, used at the Department of Computer Science, University of Economics in Katowice. The last chapter of this work shows an example of management software infrastructure developed for the purpose of supporting dynamic allocation of server resources.

## 1. Introduction to the theory of resource virtualization

The development of the concept of virtualization reaches the 50's of the last century and the first storage virtualization plans affected computers [www1] and related research in the field of IBM cloning of operating systems [www2]. In

terms of the currently used software solutions mentioned here should be Microsoft Virtual PC, which was a precursor in the Connectix Corp., VMware created at the University of Stanford in California and VirtualBox now available in the Oracle offer. From the point of view of the virtualization solutions the development of technologies is associated with the appearance on the market of CPU brand Intel (year 2005) and AMD (year 2006) to support hardware virtualization. It can be indicated that virtualization is seen as [RuDi07] solution for the separation of software from hardware layer in Figure 1.

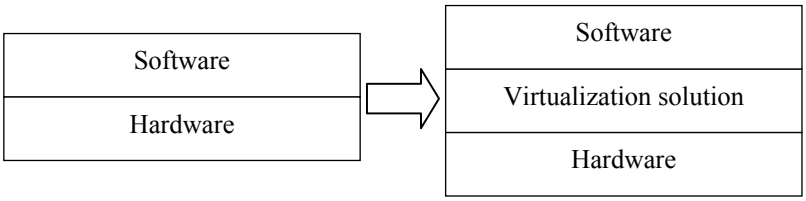


Fig. 1. Hardware virtualization

The result is the ability to share the same hardware between different programs, through the virtualization software on a server or workstation. Virtualization results in greater reliability of the entire solution by providing the ability to configure multiple dedicated servers, instead of a single server installation with a wide functionality. As a result it reduces or even eliminates the resources redundancy and improves the use of IT infrastructure. From the point of view taken in the development of the use of this kind of technology to assist in the education process, in addition to reductions in the cost of hardware, we can distinguish here:

- The ability to install and configure multiple dedicated servers, utilizing the resources according to a specific schedule of working time, may be used during the different classes and at optimum efficiency.
- Providing a testing environment in the form of cloned virtual machines, where exercises within the predefined scenarios can be conducted.
- Providing students with preconfigured software packages that can be run in any hardware environment.
- Easy to perform backup and restore specific versions of the software in the event of problems with its functioning.

To obtain such a broad range of possible virtualization applications we can point out its various types, which will be discussed in the next section.

## 2. Types of resource virtualization

As already mentioned in the previous section, virtualization allows you to manage various system resources and allows the processes and computer systems to share them. Virtualization simulates the physical objects using any of the four methods [Mari13]:

- Division − multiplexing − creating multiple virtual objects with a single instance of a physical object, e.g.:
  - processor (memory, network card) is multiplexed between a number of processed threads.
- Aggregation − the creation of one virtual object of many physical objects, e.g.:
  - RAID disk array is aggregated from a number of independent physical hard disk drives.
- Emulation − constructing a virtual object from another type of the physical object, e.g.:
  - Physical Disk emulates RAM,
  - Emulating Windows OS on an Apple computer.
- Multiplexing and emulation, e.g.:
  - virtual memory with paging multiplexes real memory and disk, and the virtual address emulates the actual address,
  - TCP protocol emulates a reliable stream of bits and multiplexes a physical communication channel and a processor.

Taking into account the possibilities of modern computer hardware, there are many different techniques of virtualization. In the simplest terms, we can assume that in the process of virtualization two kinds of systems are involved:

- The host is the computer or device that is "always powered on" and where we install the other services, for example if we have a computer with Windows 7 or Windows 8, and we want to try Linux or Android applications on it.
- The guest is the other software or "hardware" that we want to try without the need to install it on a hard drive deleting our windows, or if we do not want to change our computer as we have it, or if it can't be installed because the systems require different hardware.

The use of various techniques, however, is dependent on the hardware technology used and system software capabilities [Sing07]. Based on the techniques used types of virtualization are defined. In practice we distinguish four basic types of virtualization:

- Emulation (hardware emulation/software virtualization), involves providing the functionality of target processor completely in software (Figure 2). The main advantage of emulation is that we can emulate almost any processor on

any other type of processor. The virtual environment is able to emulate a hardware architecture which is completely different from the underlying hardware and requires an unmodified guest operating system (e.g. mobile devices). The main disadvantage of emulation is that usually the guest is slower, and require a more powerful hardware to be run the same. Emulators usually do not need administration privileges on the host computer, so we can run it as portable on public computers [www3]. In this kind of virtualization [Bagu12]:

- VM emulates/simulates complete hardware,
- allows for other processors emulation,
- running multiple systems within one,
- unmodified guest operating system for a different PC can be run,
- e.g. a mobile device such as a smart phone emulated on a desktop PC, system environments as: Bochs, VirtualPC for Mac, QEMU.

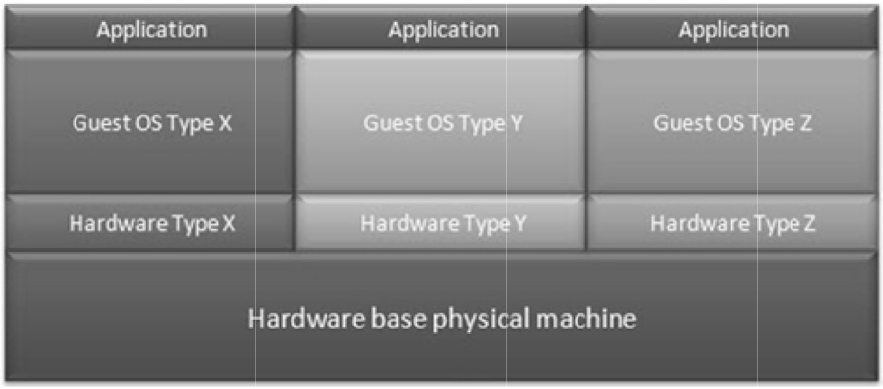


Fig. 2. Emulated-virtualization environment

Source: [www3].

- Full/native virtualization uses an image of a complete unmodified guest operating system, which is run on a virtualized environment (Figure 3). The same hardware architecture can be used to run more than one virtualized guests. All of the guests support the same hardware, which allows them to execute many instructions directly on the hardware, achieving improved performance. The guest OS is not aware it is being virtualized and requires no modification. The hypervisor translates all operating system instructions on the fly and caches the results for future use, while user level instructions run unmodified at native speed [www3]. Full virtualization offers the best isolation and security for virtual machines, and simplifies migration and portability as the same guest OS instance can run virtualized or on native hardware [www5].

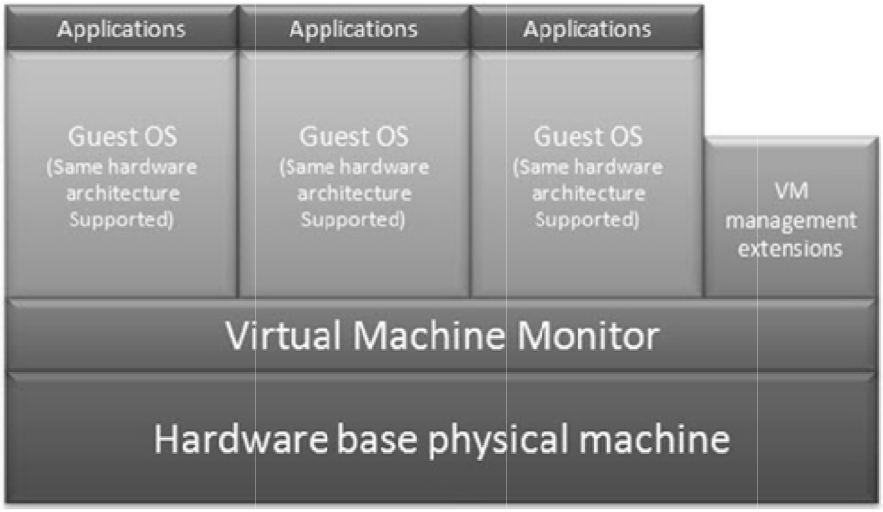


Fig. 3. Full-virtualization environment

Source: [www3].

The main advantage of full-virtualization is that most operating systems can be installed without requiring modifications [www7]. This kind of virtualization gives us the ability to run multiple different operating systems and versions from multiple vendors (e.g. Microsoft Windows Server 2003, Windows Server 2008, Linux, UNIX, Netware). The main disadvantage of virtualization is that virtualized images contains full, complete operating system installations. Very often images are extremely large and needs more resources such as processor, input/output, network, other environments [www3]. In this kind of virtualization [Bagu12]:

- every guest operating system runs in isolation,
- different operating systems mages uses the same hardware CPU,
- e.g. IBM VM family, VMWare Workstation.

- Para-virtualization uses a piece of computer software, firmware or hardware (hypervisor − VMM − virtual machine monitor) to create and run virtual machines. A hypervisor exports a modified copy of the physical hardware (Figure 4). The exported system has the same architecture as the server hardware, but we can do specific modifications that allow the guest operating systems to achieve high near-native performance. Also we need to make modifications of the guest operating systems allowing them to take advantage of these modified tasks [www3]. Not every operating system or application can support being paravirtualized and not every server virtualization vendor supports

running operating systems in a paravirtualized format. The guest OS/program will be aware that it operates in a shared medium at some level, and although it may not be visible to the user, it will be to the system administrator [www6]. The hypervisor also provides hypercall interfaces for other critical kernel operations such as memory management, interrupt handling and time keeping [www5]. Para-virtualized environment provides the same functionality that we would expect from the physical hardware.

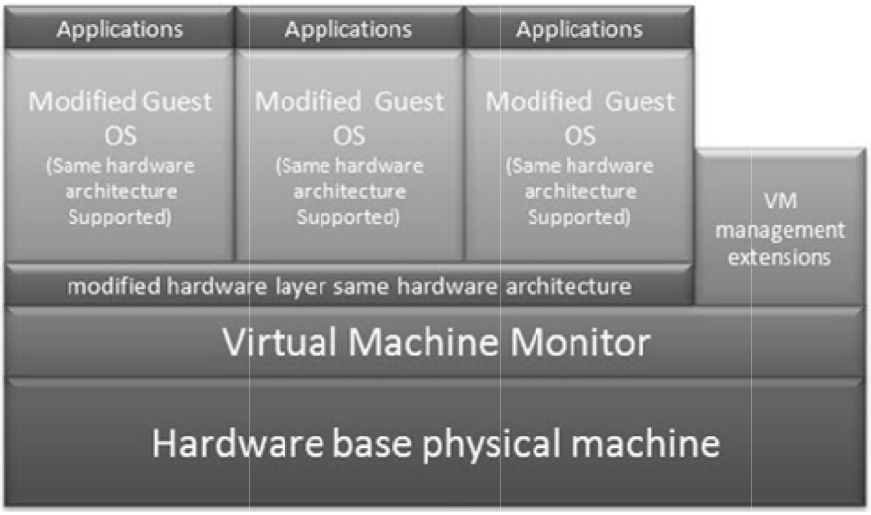


Fig. 4. Para-virtualization environment

Source: [www3].

The main advantage of para-virtualization is that we can achieve resource savings, with unchanged performance. Para-virtualized image sizes are significantly smaller,and can reach near-native performances. This kind of virtualization allows for the virtualization of architectures that would not normally support full virtualization. The main disadvantage of para-virtualization is that guest operating systems require modifications, which allows them to support hypercalls over native functions. In para-virtualization:

- VM does not simulate hardware,
- we need to modify guest operating systems with special API,
- Hypercalls trapped by the Hypervisor and serviced,
- e.g. Xen, VMWare ESX Server.

- Operating system-level virtualization (host-level virtualization). This approach doesn't provide virtual machines. Full virtualization processes is doing completely within a single operating system. A single instance of the host

operating system supports multiple virtual instances, but with the use of the same operating system kernel, the same input-output devices, and addressing the need for virtual servers on the process level [www4]. The guest systems share common features and drivers of the underlying OS, while looking and feeling like completely separate computers. Each guest instance will have its own file system, IP address, and server configuration, and will run completely different applications [McAl07, Yang07].

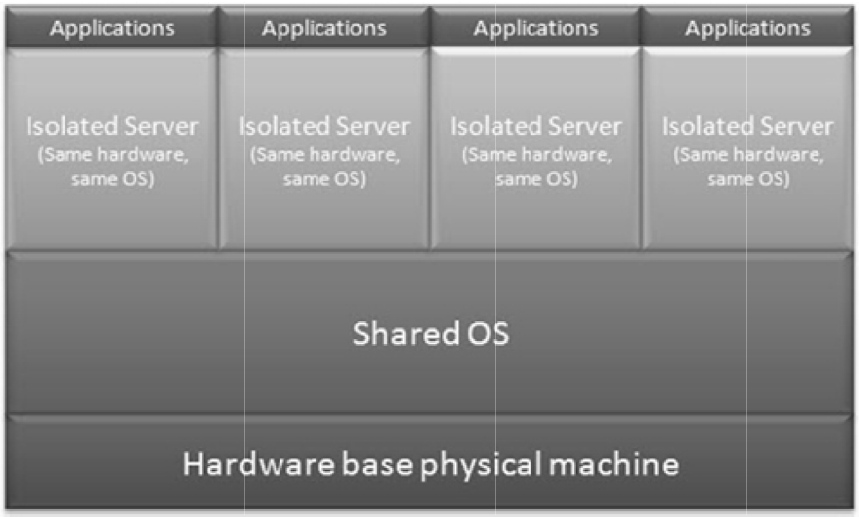


Fig.5. Operating system-virtualization environment

Source: [www3].

Main advantages of host-level virtualization are high speed, lightness, and efficiency. System is able to support a large number of virtual instances at the same time. As disadvantages: isolation of instances and usually needs to special data secure should first be noted. Besides, all virtual instances must support the same operating system. In host-level virtualization:

- operating system allows multiple secure virtual servers to be run,
- guest operating system is the same as the hosts,
- instances are running in isolated environments,
- e.g. Solaris Containers, BSD Jails, Linux Vserver.

- Quite a different approach is application level virtualization. Some of the issues require the use of applications running in a variety of environments. Application level virtualization is designed for applications that are incompatible one to another and gives them the opportunity to be run together side by side [WSMS07]. The virtualization environment intercepts all calls that are

made by the virtualized application to the underlying file systems, redirecting calls to a virtual location. Applications are completely abstracted from their physical platforms and interacts only with the virtualization environment. The main advantage of this kind of virtualization is improving the portability of applications by allowing them to run in different operating environments even not designed for them. Application level virtualization allows incompatible applications to run side by side. Among disadvantages we must recognize that supporting virtual machines can dramatically slow down execution of applications, in both run-time and native environments. Another problem is that application level application is not a complete solution, because some applications can't be virtualized [Varh08]. Application level virtualization:

- allows applications to run in not native environments,
- allows accelerated application deployment through on-demand application streaming,
- applications can be installed, patched, and upgraded once for an entire environment, instead of for each individual computer,
- e.g. JVM, own registry files, global objects.

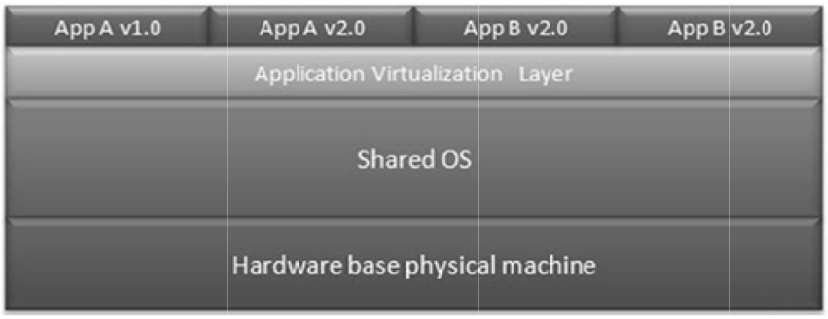


Fig. 6. Application-virtualization environment

Source: [www3].

## 3. Example applications for server virtualization in support of teaching and learning processes

For the purpose of assisting the process of managing the server layer, which is in the possession of the Department of Computer Science, it became necessary to develop a software, which task would be to support the management of virtual machines. Especially because of the fact that the hardware resources available to the Department of Computer Science may not be sufficient for simultaneous launch and operation of all systems used in the classroom teaching at full load, the idea of implementing virtualization software for IT systems and their control came to our attention.

Resources available to the Department of Computer Science is the central unit of IBM Model: IBM eServer x3500 equipped with eight-core Intel Xeon processor X5355 2.66GHz clock frequency, 9GB RAM memory in DDR2 standard and hard drive of 1TB capacity. Software supporting virtualization process which was decided to implement is VMware ESXi version 5.1. Two main factors had influence on the choice of such a solution. First, VMware ESXi is running under the open source license, or at least a major part of the functionality offered by it, and allows the installation of most commercially available operating systems.

To manage the VMware platform used a special VMware vSphere Client was created by the producer. However, the software turned out to be insufficient, as it required each of the tutors to be also a server administrator, or administrators were available during the classes, when the software installed on the server was used, in order to properly manage the server load by specified virtual machines. As a result of the experience gained it was decided that it is necessary to develop a solution, supporting dynamic allocation/reallocation of resources assigned to virtual machines, so that the resources for individual machines were allocated at a given time according to their momentary needs. The second assumed functionality of the software is the possibility to obtain alerts regarding system operation and profiling of individual virtual machines.

To cope with this problem server administrators decided to implement their own software for dynamic resource management, which would work as a plug-in for the already installed VMware ESXiAfter analyzing the availability of libraries for popular programming languages used for making calls and carry out commands on a VMware server it has been decided to choose the Java programming language and a library vijava. Figure 7 shows the elements of the developed solution.

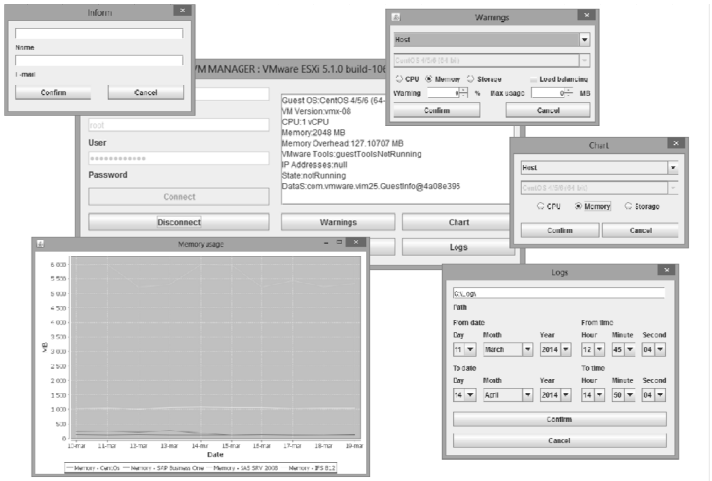


Fig. 7. The development of a Virtual Machine management software

The software is based on historical data, collecting server load logs, which recorded the date, time, type of monitored resource and its load. Resources managed by the application are: CPU power, amount of RAM and the amount of available space on the HDD of the server. Frequency of entry collecting is dependent on the time of day. Higher falls on the time when classes are held, mainly because at this time there are problems with the allocation of resources. Standard retention of storing the alerts is half a year, since the load generated on the Department's server is temporary − semi-annual, semester. During the semester, the load increases and decreases in certain hours, depending on the schedule of classes. The software is adapted for the manual parameterization of the collection of frequencies and retention of logs.

Basing on the gathered information, the application performs the process of requesting and distributes resources according to needs among all the virtual machines running within the host. At the same time, helding a process of continuous monitoring, which involves verification if resources are optimally deployed between the machine, and whether the use of resources allocated in the first phase of the request does not exceed the threshold, exceeding which will restart the process of balancing resources, which reallocates server resources according to the new demand.

Additional functionality of the application is to inform administrators of the server about exceeding threshold values by any of the machines, and the launch of the balancing process by submitting information about the percentage use of resources by the machine that generated the warning. Notifications of virtual machines in addition to the names and numeric values contain a graphical representation of data in graphs.

Below we present an example of the resources balancing implementation process. Initially, all virtual machines running on the server were set to the standard resource values presented in Table 1, assigned by the client program VMware vSphere.

Table 1

Table of resources allocated to individual machines

| Name of virtual machine | RAM Memory [MB] | CPU [GHz] | HDD [GB] |
|---|---|---|---|
| CentOS5 | 2048 | 1 x 2,66 | 40 |
| SAS SRV 2008 | 2048 | 1 x 2,66 | 40 |
| IFS 12B | 2048 | 1 x 2,66 | 40 |
| SAP Business One | 2048 | 1 x 2,66 | 40 |

Resource consumption at server run-time (17 Feb 2014 – beginning of the summer semester 2013/2014) is presented in Figure 8.

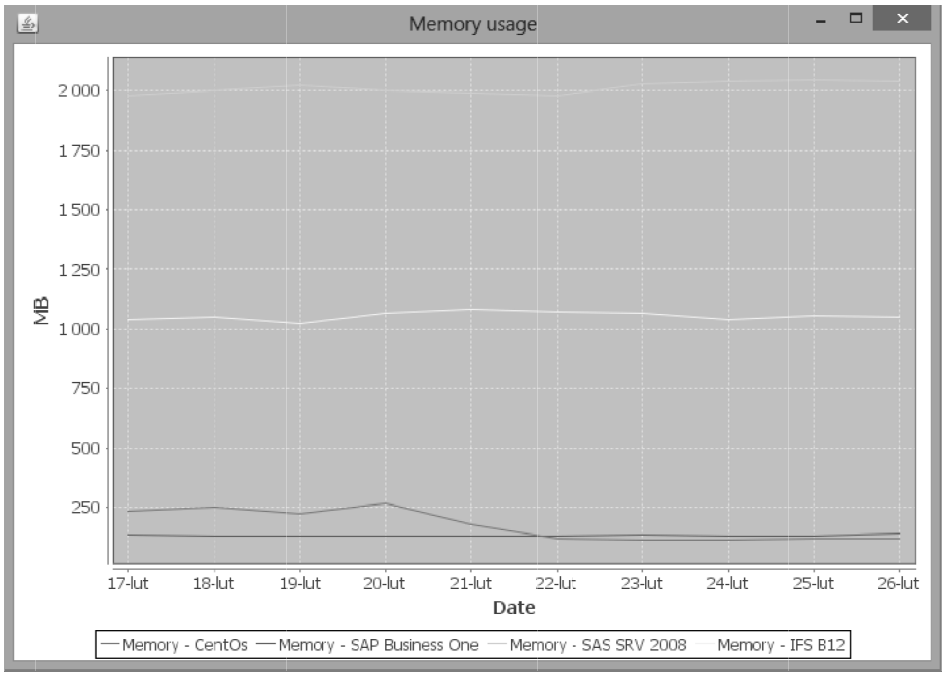


Fig. 8. Memory usage for the host

At the time presented in the diagram (Figure 8) the automatic balancing of resources was off − the application was working only as a tool for monitoring and collecting information on resource usage for a specific machine. As can be seen, the assignment of standard RAM resource values for the three machines: CentOS, SAP and IFS turned out to be more than sufficient. However, the last machine designed for the SAS application showed almost the maximum memory consumption. For this purpose the automatic balancing of resources was enabled, which, based on the data collected several times, relocated the resources leading to the situation shown in Chart 2, while detailed information is contained in Table 2.

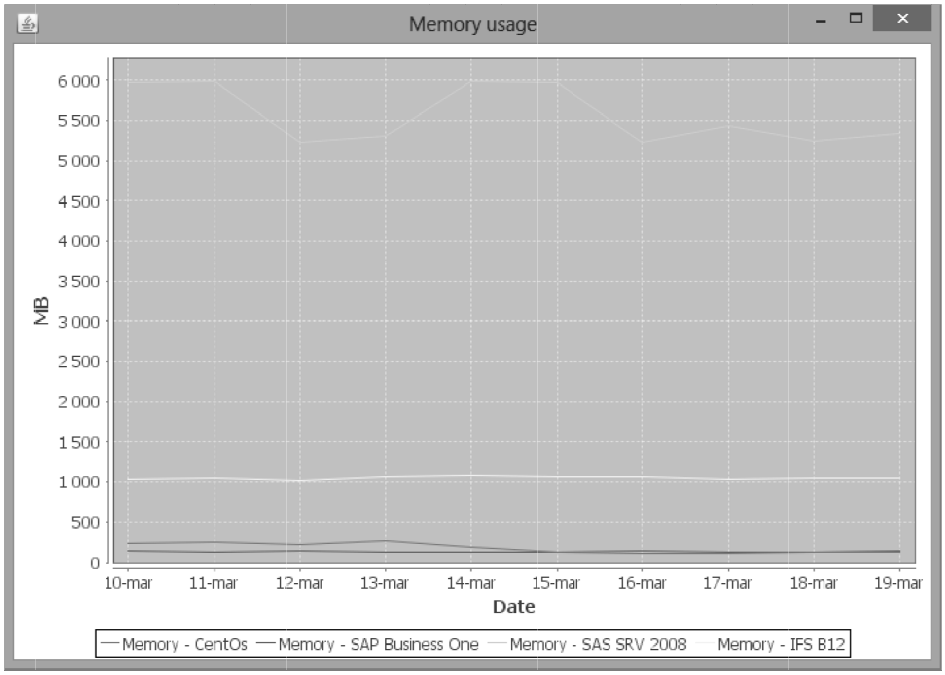


Fig. 9. Memory usage for the host

Table 2

Resources allocated to individual machines in the process of balancing resources

| Name of virtual machine | RAM Memory [MB] | CPU [GHz] | HDD [GB] |
|---|---|---|---|
| CentOS5 | 320 | 1 x 2,66 | 40 |
| SAS SRV 2008 | 7200 | 4 x 2,66 | 40 |
| IFS 12B | 1296 | 1 x 2,66 | 40 |
| SAP Business One | 160 | 1 x 2,66 | 40 |

Resource management algorithm implemented in the application takes into account the maximum noted value and allocates the machine resource value 20% greater than the maximum. If this value is insufficient, as it was in the first step for the SAS machine, the operation is repeated until the desired effect.

The implementation of dynamic process management based on historical data − collection and analysis of program logs, turned out to be a sufficient solution to assist teaching and learning processes, where the dynamics of changes in load on the server by the specified virtual machines is relatively small and has a temporary nature – operation of this type of resource balancing is done mainly at the start of each semester.

## Conclusions

The use of solutions to support virtualization processes, as indicated in the study, may relate to different aspects of this process and can be associated with virtualization hardware and software. The use of dedicated software allows you to efficiently use available resources to optimize hardware and its application, while minimizing the costs of purchase and operation. In order to make best use of the provided configuration of such a system, it is often necessary, however, to use the software solutions that go beyond the currently available solutions on the market. An example of such software is the prototype of an application developed to assist in the process of dynamic allocation of resources, described in this work.

## References

[Bagu12] Bagul A.: What Is Virtualization and Types of Virtualization. http://www.indiangnu.org/, 2012.

[Mari13] Marinescu D.C.: Cloud Computing: Theory and Practice. Morgan Kaufmann, 2013.

[McAl07] McAllister N.: Server Virtualization. InfoWorld, Feb. 12, 2007. Retrieved March 12, 2008. http://www.infoworld.com/article/07/02/12/07FEvirtualserv_1.html.

[RuDi07] Rule D., Dittner R.: The Best Damn Server Virtualization Book Period. Syngress Publishing, Inc. 2007.

[Sing07] Singh A.: An Introduction to Virtualization. Kernelthread.com. Retrieved March 18, 2008. http://www.kernelthread.com/publications/virtualization/.

[Varh08] Varhol P.: Application Virtualization Hits Its Stride. Redmondmag.com, 2007.

[WSMS07] Winslow P., Semple R., Maynard J., Simson D., McGrath B.: Desktop Virtualization Comes of Age. Credit Suisse 2007.

[Yang07] Yang Y.: OS-level Virtualization and Its Applications. ProQuest, 2007.

[www1] Wadowski Ł., Masztafiak P.: Historia wirtualizacji. Cz. 1. www.virtual-it.pl, 2009.

[www2] Lelusz M.: Rys historyczny technologii wirtualizacji. www.blog.inleo.pl, 2009.

[www3] Chantry D.: Mapping Applications to the Cloud. msdn.microsoft.com, 2009.

[www4] Jakubik K.: ABC wirtualizacji – Serwery. www.computerworld.pl, 2008.

[www5] Understanding Full Virtualization, Parawirtualization, and Hardware Assist. VMWare WhitePaper, www.vmware.com.

[www6] Hypervisor vs Paravirtualization − Server Virtualization Overview. www.petri.co.il, 2012.

[www7] Full Server Virtualization versus Para-Virtualization – Which Is Best? http://www.iemc07.org, 2013.

## WIRTUALIZACJA SYSTEMÓW INFORMATYCZNYCH ORAZ JEJ ZASTOSOWANIE NA PRZYKŁADZIE UCZELNI WYŻSZEJ

### Streszczenie

Rozwój systemów informatycznych, a w szczególności ich warstwy sprzętowej powoduje, iż często posiadane przez firmę zasoby techniczne przerastają wymagania narzucane przez dostawców oprogramowania. W przypadku zastosowania rozwiązań informatycznych niewykorzystane moce obliczeniowe serwerów mogą być optymalizowane przez zastosowanie odpowiedniego oprogramowania pozwalającego na instalację wielu systemów operacyjnych na jednej maszynie oraz współdzielenie jej zasobów w obrębie wielu rozwiązań programowych. Powoduje to obniżenie kosztów serwisowania, sprzętu, prądu oraz ułatwia procesy zarządzania całym środowiskiem.

Celem niniejszego opracowania jest ukazanie koncepcji wirtualizacji, jej rozwoju, typologii oraz zaprezentowanie doświadczeń dotyczących kilkuletniego zastosowania oprogramowania wspomagającego wirtualizację dla celów dydaktycznych, stosowanego Katedrze Informatyki Uniwersytetu Ekonomicznego w Katowicach. Ostatnia część ukazuje przykład oprogramowania wspomagającego zarządzanie infrastrukturą informatyczną, opracowanego na potrzeby wspomagania dynamicznej alokacji zasobów serwera.