The 6th International Scientific Conference
"DEFENSE RESOURCES MANAGEMENT IN THE 21st CENTURY"
Braşov, December 02-03, 2011

IMPROVING IT SERVICES THROUGH VIRTUALIZATION

Cpt. OGÎGĂU NEAMŢIU FLORIN

Regional Departament of Defense Resources Management

Abstract:

Even if virtualization was used for the first time in the middle of the '90s its true advantages are being applied today. Virtualization presents an attractive solution for organizations looking to save money and generate value from their IT investments because of its potential to reduce capital expenses and energy costs.

The technology has found important applications over a wide range of areas such as server consolidation, supporting multiple operating systems, kernel debugging and development, system migration, etc, resulting in a widespread usage. Most of them present to the end user a similar operating environment however they tend to vary widely in their levels of abstraction and the underlying architecture.

This article discusses what virtualization is and how organizations can benefit from adopting virtualization into future IT plans.

Key words: virtualization, technology, virtual machine, abstraction layer, hypervisor, server consolidation, load balancing

Virtualization was first developed in the 1960s by IBM to partition large, mainframe hardware for better hardware utilization. Each virtual system used to be an instance of the physical machine that gave users an illusion of accessing the physical machine directly. It was an elegant and transparent way to enable time-sharing and resource-sharing of highly expensive resources. Users could execute, develop, and test applications without having to fear causing a crash to systems used by other users on the same computer. So, in the beginning virtualization was used to reduce the hardware acquisition cost and to improve the productivity by letting more number of users work on it simultaneously.

After a decline of the interest to this technology, in the early 90 there was a great increase in the number of applications which used virtualization concepts across a wide range of areas in computer science. Because of this great range of usage there are a lot of definitions of this technique. However, for this article, the following one is the most appropriate: "Virtualization is a technology that combines or divides computing resources to present one or many operating environments using methodologies like hardware and software partitioning or

aggregation, partial or complete machine simulation, emulation, time-sharing, and many others"[1] .

This means, that virtualization uses techniques to abstract from the real hardware and provides isolated environments. These are capable to support various applications or even a whole operating system.

Even if there are technologies that use virtualization to create certain specific virtual resources (memory virtualization) generally through virtualization one understands the creation of virtual machine software which emulates a whole system.

Virtualization technologies allow multiple virtual machines, with heterogeneous operating systems to run side by side and in isolation on the same physical machine. By emulating a complete hardware system, from processor to network card, each virtual machine can share a common set of hardware unaware that this hardware may also be used by another virtual machine at the same time. The operating system running in the virtual machine sees a consistent, normalized set of hardware regardless of the actual physical hardware components.

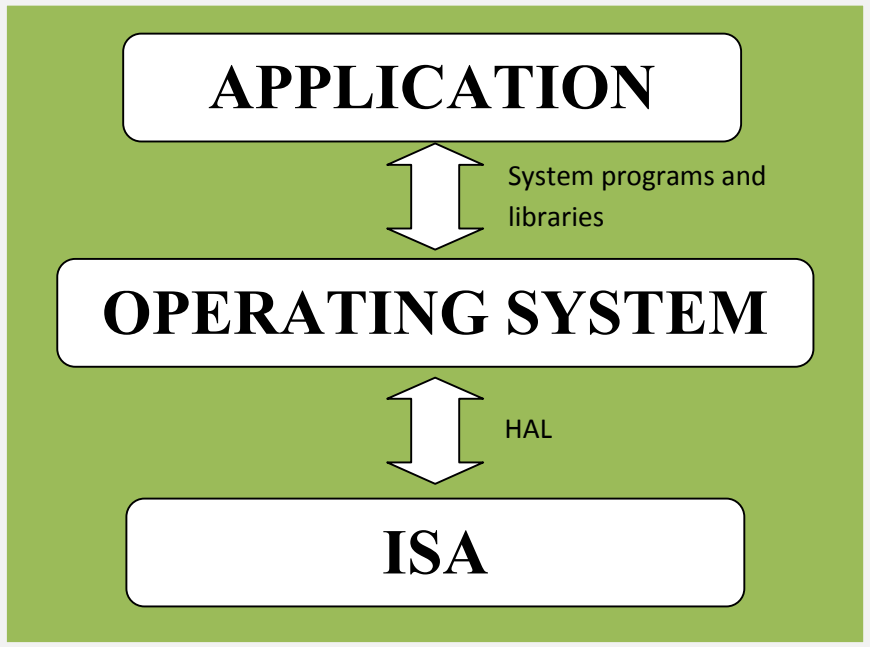

Fig.1 Levels of virtualization

Conceptually a **virtual machine** represents an operating environment for a set of user-level applications, which includes libraries, system call interface/ service, system configuration, system processes, etc. There can be several levels of abstraction where virtualization can take place: instruction set level, hardware abstraction layer (HAL), OS level, or in the application level (Fig.1). Whatever may be the level of abstraction, the general phenomenon still remains the same; it partitions the lower-level resources using some techniques to map it to multiple higher level VMs transparently.

Virtualization at the instruction set architecture level is implemented by emulating an instruction set architecture completely in software. A typical computer consists of processors, memory chips, buses, hard drives, disk controllers, timers, multiple I/O devices, and so on. An

emulator tries to execute instructions issued by the guest machine (the virtual machine that is being emulated) by translating them to a set of native instructions and then executing them on the available hardware. For example, an x86 emulator on Sparc processor can execute any x86 applications, thus giving the illusion to the application as if it is a real x86 processor. To achieve this, however, an emulator would have to be able to translate the guest ISA (x86 here) to the host's ISA (Sparc). For an emulator to successfully emulate a real computer, it has to be able to emulate everything that a real computer does that includes reading ROM chips, rebooting, switching it on, etc.

Virtualization at the HAL exploits the similarity in architectures of the guest and host platforms to cut down the interpretation latency. A hardware abstraction layer (HAL) is an abstraction layer, implemented in software, between the physical hardware of a computer and the software that runs on that computer. Its function is to hide differences in hardware from most of the operating system kernel, so that most of the kernel-mode code does not need to be changed to run on systems with different hardware. On a PC, HAL can basically be considered to be the driver for the motherboard and allows instructions from higher level computer languages to communicate with lower level components, such as directly with hardware.

Most of the today's world's commercial PC emulators on popular x86 platforms (VMware, Virtual PC) use this virtualization technique to make it efficient, viable and practical. Virtualization techniques help the mapping process of the virtual resources to physical resources and the use of native hardware for computations in the virtual machine. When the emulated machine needs to talk to critical physical resources, the simulator takes over and multiplexes appropriately.

The **virtual machine monitor** (VMM) sometimes called hypervisor is a software-abstraction layer that partitions a hardware platform into one or more virtual machines. Each VM provides facilities to an OS or application to believe as if it runs in a normal environment and directly on the hardware. (Fig.2)

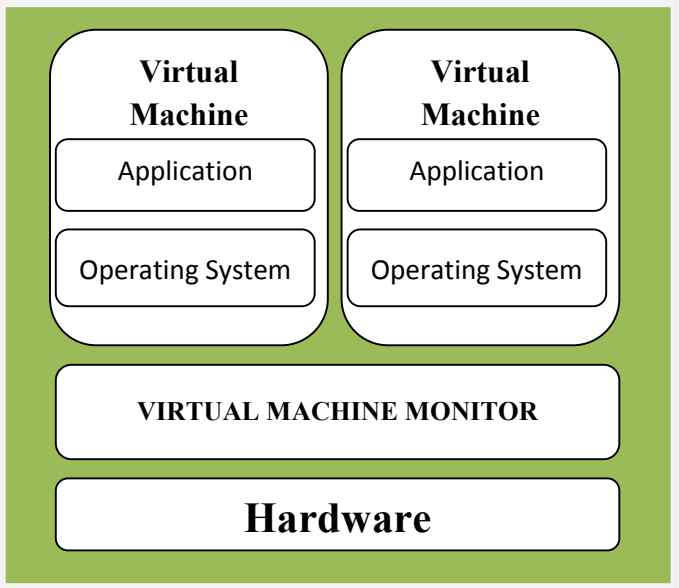

Fig.2 Virtual machine monitor

For such a virtualization technology to work correctly, the VM must be able to trap every privileged instruction execution and pass it to the underlying VMM to be taken care of. This is because, in a VMM environment, multiple VMs may each have an OS running that wants to issue privileged instructions and get the CPU's attention. When a trap occurs during privileged instruction execution, rather than generating an exception and crashing, the instruction is sent to the VMM. This allows the VMM to take complete control of the machine and keep each VM isolated. The VMM then either executes the instruction on the processor, or emulates the results and returns them to the VM. However, the most popular platform, like x86, is not fully-virtualizable, i.e. certain supervisor (privileged) instructions fail silently rather than causing a convenient trap when executed with insufficient privileges. Thus, the virtualization technique must have some workaround to pass control to the VMM when a faulting instruction executes. Most commercial emulators use techniques like code scanning and dynamic instruction rewriting to overcome such issues.

Operating system-level virtualization is based on abstraction of the operating system layer to support multiple, isolated partitions on a single instance host operating system. With a careful partitioning and multiplexing technique, each VM can be able to export a full operating environment and fairly isolated from one another and from the underlying physical machine.

This technique results in very low virtualization overhead and can yield high partition density. However, there are two major drawbacks with this type of solution. The first drawback is the inability to run a heterogeneous operating system mix on a given machine because all partitions share a single operating system kernel. The second big drawback, also caused by the shared kernel model, is the lack of support for running a mixed 32-bit and 64-bit workload. In addition, any operating system kernel patch affects all virtual environments. For these reasons, operating system-level virtualization tends to work best for largely homogeneous workload environments.

Virtualization at the application level is a little different. In this case this technology implements a virtualization layer as an application that eventually creates a virtual machine. The created VM could be as simple as a language interpreter or as complex as Java Virtual Machine.

This technology is geared towards partitioning and isolating client side applications running on the local operating system. Applications are isolated in a virtual environment layered between the operating system and application stack. The virtual environment loads prior to the application, isolates the application from other applications and the operating system, and prevents the application from modifying local resources such as files and registry settings. Applications can read information from the local system registry and files, but writeable versions of these resources are maintained inside the virtual environment. In fact, the application may never be locally installed on the desktop; instead the code bits can be dynamically streamed and cached in the virtual environment as new portions of the application are needed.

There are several benefits provided by application-level virtualization. Of most importance are increased stability of the local desktop, simple application removal without changes to the local environment that could negatively affect other applications, and seamless,

conflict-free, side-by-side execution of multiple instances of an application. Advantages of virtualization

Originally virtualization was used mostly to facilitate server consolidation, but now many other approaches present themselves. Virtualization starts during the design phase of the technical environment, when the design team considers how to support the business processes and identifies assets needed to convert the plan to reality. It is during this phase that enterprises often realize that the life cycle of provisioning the right hardware and other equipment can go on longer than expected.

- **Cost reduction**—By consolidating many instances of (virtualized) servers onto a physical one, enterprises lower their hardware expenditures. In addition to lower capital expenditures, virtualized environments enable enterprises to save on maintenance and energy, often resulting in a reduced total cost of ownership;
- **Dynamic Load Balancing**—Deploying several virtual environments guarantees the good practices of high availability, redundancy and failover since workloads can go where they are more efficient. Thus, virtualization focuses not only on effectiveness (doing the right things) but also efficiency (doing things in a faster, cheaper and more reliable way). The dynamic load balancing creates efficient utilization of server resources.
- **Disaster Recovery**— Disaster recovery is a critical component for IT, as system crashes can create huge losses. Virtualization technology enables a virtual image on a machine to be instantly re-imaged on another server if a machine failure occurs;
- **Responsiveness**—Since the virtual environment has the ability to provision itself to get the best out of available resources, response times are faster and downtimes can be reduced to near zero, improving agility and performance;
- **Segregation** —Processes that once needed to exist within the same physical machine can now be easily separated while still maintaining the robustness and security required. The different virtualized worlds (network, operating system [OS], database, application, etc.) can be decoupled (even distributed in different geographic locations) without threatening integrity within the process;
- **More Operational Flexibility**—The relatively easy creation or preparation of the right environment for the right application enables enterprises to provide flexibility to the infrastructure, not only in the test or preproduction phases but also in the production area. When a new procedure or technical/business requirement arises, virtualization's ability to enable rapid creation of the environment allows the business to implement the environment more rapidly like in the standard way;
- **Agility**—Agility facilitates quick adaptation to business needs, such as when orders peak and additional computing power is needed. An enterprise may even choose to overcome the resources of a physical machine since virtualization facilitates rapid movement of the different resources that "live" in one physical machine to other virtual machines. In this way, virtualization supports alignment with business needs;
- **Simplification**—Even if some resources are virtualized, some of the typical IT difficulties will still be present. However, reducing the number of physical machines significantly reduces the probability of failure and the cost of management and this results in simplification—one of the promises of virtualization;

- **Space utilization**—Server consolidation saves space in the data center and facilitates scalability since many servers exist within one server;
- **Improved System Reliability and Security**—System virtualization helps prevent system crashes due to memory corruption caused by software like device drivers. The virtual software architecture provides methods to better control system devices by defining the architecture for direct memory access and interrupt remapping to ensure improved isolation of Input/Output resources for greater reliability, security, and availability;
- **Automation**—Technology allows some virtualized environments to be provisioned as needed and on the fly, thus facilitating automation of business processes and eliminating the need to continually allocate resources and manage portions of the technical environment that support sporadic business needs. Some virtualization technology facilitates the automatic allocation of a process for its optimal performance within a pool of virtualized environments;
- **Sustainability**—Virtualized environments use less environmental resources. Energy consumption in data centers is often wasted on machines that are consistently underutilized. Since virtualization allows for many virtual machines to run on one physical machine, less energy is needed to power and cool devices.
- **Testing and development**—Use of a VM enables rapid deployment by isolating the application in a known and controlled environment. Unknown factors such as mixed libraries caused by numerous installs can be eliminated. Severe crashes that required hours of reinstallation now take moments by simply copying a virtual image.

Virtualization is one of the more significant technologies to impact computing in the last few years. Even if it hasn't reached the full of its powers virtualization has already affected the way companies run their IT operations. The technology is being used by a growing number of organizations because of its benefits like consolidation of infrastructure, lower costs, ease of management, increased efficiency and utilization, better employee productivity, and more. Even if virtualization has many advantages, companies must also consider the potential security risks and change implications while moving to a virtualized environment. However, virtualization is surely going to make a significant impact on the landscape of computing over the next years.

References:

[1] Susanta Nanda Tzi-cker Chiueh and Stony Brook. A Survey on Virtualization Technologies. RPE Report, pages 1-42, 2005

[2] http://www.intel.com/intelpress/sum_vpio.htm

[3] http://www.vmware.com/virtualization/

[4] http://www.microsoft.com/windows/virtual-pc/http://www.infoq.com/articles/virtualization-intro