UDC 621

ARTEMII VASILYEVICH KROPACHEV
Bell Integrator USA Automation Solution Department Manager, USA, Colorado
DENIS OLEGOVICH ZUEV
Independent Consultant Lead Arcitect, Network and Cloud USA, Colorado

MAIN ASPECTS OF THE MODERN INFORMATION SYSTEMS HARDWARE RESOURCES VIRTUALIZATION METHODOLOGY

Main aspects of peculiarities of the modern information systems hardware resources virtualization were analyzed. It was shown that virtual machines concept provides great opportunity for parallel computing while virtualization technology enables sharing of hardware resources by multiplexing virtual machines on the same server's farm. Analysis has demonstrated that virtualization can be implemented at different operational levels: instruction set architecture level, hardware abstraction level; operating system level; user level API and application level. Depending on that it could be defined classes of virtual machine architecture: hypervisor architecture, host-based virtualization, paravirtualization. It was demonstrated that there are two types of hypervisors: micro-kernel architecture and monolithic hypervisor architecture groups which are proves to be effective and flexible but requires a lot of resources. Host-based virtualization class advantages were shown as installation without modifying the host operation system and various host machine configurations which could be adopted. It was noticed that performance of this architecture is rather low so it usually cannot be adopted. It was also demonstrated that paravirtualization method implies modifying the guest operation system and development special APIs set so virtualization layer can be inserted at different positions in server software. It was analyzed development hardware-assisted virtualization technology and analysis demonstrated virtualization algorithm have to include further virtualization techniques and tools: virtualization technology for directed input/output, virtualization technology for connectivity, interrupt remapping as software capability for rerouting signals sent from peripheral devices, memory management unit and translation lookaside buffer.

Keywords: *virtualization level, virtual machine architectures, virtual networking, virtual cluster construction, cloud computing, virtual machine monitor.*

АРТЕМИЙ ВАСИЛЬЕВИЧ КРОПАЧЕВ
руководитель департемента решений автоматизации Bell Integrator USA, Колорадо,США
ДЕНИС ОЛЕГОВИЧ ЗУЕВ
независимый консультант, ведущий архитектор сетей и облачных вычислений, Колорадо США

ОСНОВНЫЕ АСПЕКТЫ МЕТОДОЛОГИИ ВИРТУАЛИЗАЦИИ АППАРАТНЫХ РЕСУРСОВ СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Проанализированы основные аспекты особенностей виртуализации аппаратных ресурсов информационных систем. Показано, что концепция виртуализации предоставляет большие возможности для параллельных вычислений, позволяя совместно использовать аппаратные ресурсы путем мультиплексирования виртуальных машин на базе серверного парка. Анализ продемонстрировал, что виртуализация может быть реализована на разных операционных уровнях: уровень архитектуры набора команд, уровень абстракции аппаратных средств; уровень операционной системы; API уровень пользователя и уровень приложения. В зависимости от этого могут быть определены классы архитектуры виртуальной машины: архитектура гипервизора, виртуализация на основе хоста, паравиртуализация. Было продемонстрировано, что существуют два типа гипервизоров, которые отличаются параметрами эффективности, адаптивности и ресурсоемкости. Преимущества класса хост-виртуализации включают в себя инсталляцию без изменения операционной системы хоста и адаптивность к различным конфигурациям хост-машины. Было отмечено, что производительность этой архитектуры довольно низкая, поэтому зачастую не рассматривают при разработке виртуальной машины. Было также продемонстрировано, что метод паравиртуализации подразумевает модификацию гостевой операционной системы и специальных API-интерфейсов разработки, поэтому уровень виртуализации может быть внедрен на разных уровнях серверного программного обеспечения. Было проанализировано развитие технологии аппаратной виртуализации, анализ показал, что алгоритм виртуализации должен включать дополнительные технологии и средства виртуализации: технология виртуализации связанные с работой процессора, оперативного запоминающего устройства (физические и виртуальные адреса: GVA, GPA, HPA) и устройств ввода-вывода (функциональные узлы контроля памяти MMU и оптимизации TLB).

Ключевые слова: *уровень виртуализации, архитектура виртуальных машин, виртуальные сети, виртуальные кластеры, облачные вычисления, монитор виртуальной машины.*

1. Introduction

Virtual machines (VMs) concept provides a great opportunity for parallel and distributed computing. Virtualization paradigm technology enables sharing of hardware and software resources by multiplexing VMs on the same servers farm of hardware hosts. A traditional server farm uses host operating system which should be used in accordance for its hardware architecture, but after virtualization procedure it is became to run different user applications managed by their own operating systems (OS) on the same server farm. Most simple model of virtualization implies additional software implementation which for virtualization layer, known as virtual machine monitor (VMM) or hypervisor [1, 2]. VM uses virtualized hardware resources (CPU, RAM, cash-memory, data storage, graphics card and input-output components). Thereby, software layer virtualizes the physical hardware of a server farm into virtual resources of the VMs which can be implemented at different operational levels [1-4] such as:

instruction set architecture (ISA) level;

- hardware abstraction level;

- operating system level;
- user level API (library level);
- application level.

At the instruction set architecture level, virtualization could be performed by emulation of ISA. It brings possibility to run larger amount of program code for various processors types on limited host hardware resources. The emulation method has to be based on code interpretation. The algorithm should interpret the source instructions to get target instructions and for each source instruction it could several target instructions to perform its function. So the main disadvantage of this operational level is complex algorithm which requires a lot of time. To solve this problem dynamic binary translation was developed. This algorithm translates basic blocks of dynamic source instructions to target instructions. According to the method basic blocks can also be extended to group of blocks in order to increase translation efficiency. It should be mentioned that instruction set emulation requires further development of binary translation and optimization.

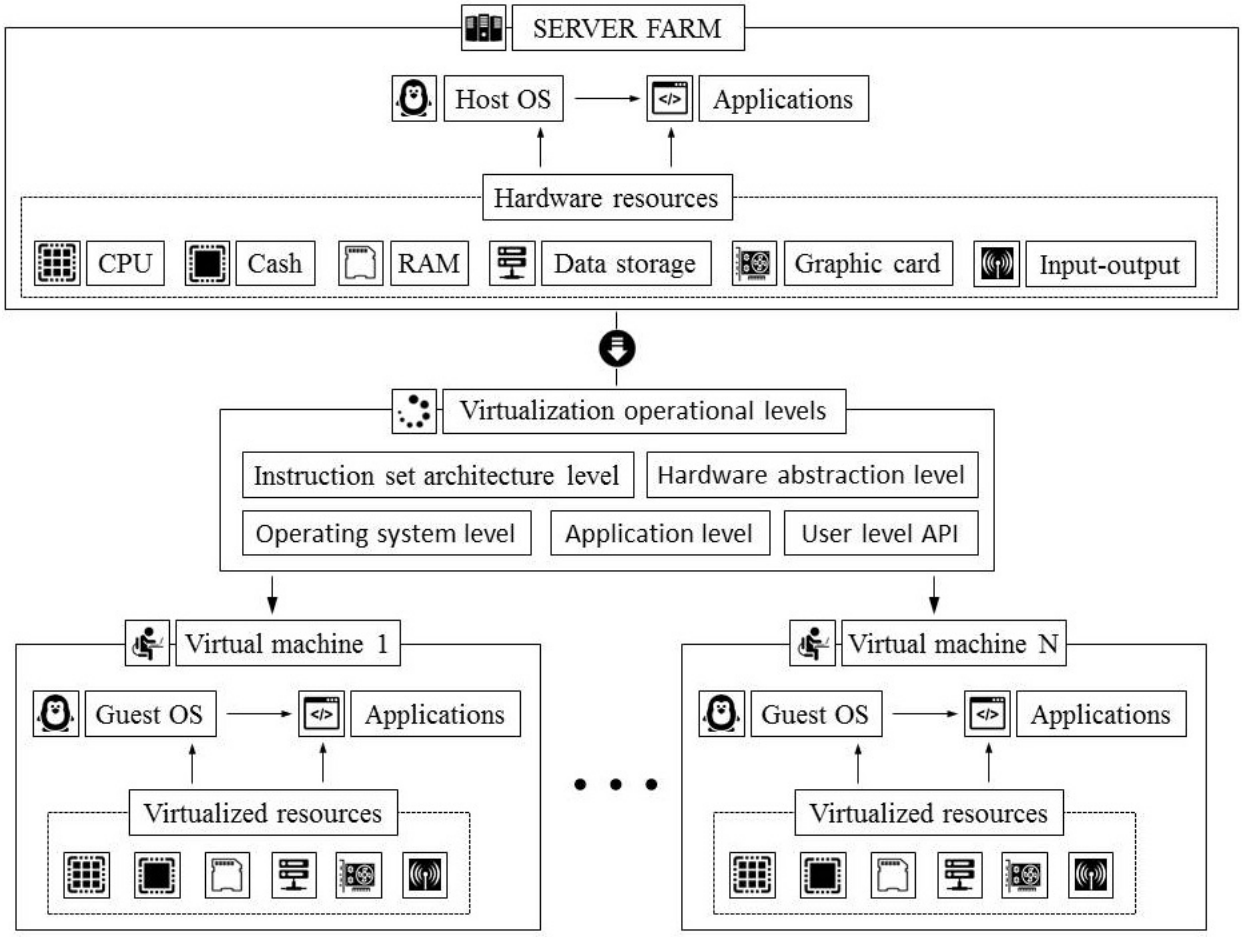

Fig. 1. Basic algorithm of information system hardware resources virtualization process

Hardware abstraction level virtualization is performed to the hardware resources of server farm. This method allows to generate a virtual hardware environment for a VM and to manage hardware resource through virtualization. Up to it the goal is to upgrade the hardware utilization rate by multiple users concurrently. Operating system level refers to an abstraction layer which lay between traditional OS and user applications. It creates isolated platform on physical servers that operates like real servers. Operating system level virtualization is often used in creating virtual hosting environments to divide hardware resources among a big number of users. Library support (API) level was designed in order to use APIs exported libraries rather than using system calls by the OS. Nowadays most systems provide APIs which are documented good enough so this operational level becomes a popular one. Library interface virtualization is possible by controlling the communication link between applications and the system through API hooks. User application level virtualization naturally implies virtualization of application as a VM. Application usually works as a process so application level virtualization is could be called process level virtualization. According to this method virtualization layer work with application program and operating system, thus layer exports an abstraction of a VM that can run programs compiled to a abstract machine definition. Main benefit of the conception implementation is a simplification of application distribution and removal from user workstations.

2. Virtualization tools and mechanisms

As it was mentioned above there are five levels of virtualization which could be used in development of classes of VM architecture. It should be noticed that before virtualization OS manages the hardware, while after virtualization layer should be inserted between the hardware and the OS. Operational layer of virtualization converts hardware resources into virtual hardware, thereby different user OS can run on the same physical platform, simultaneously. Depending on the virtualization layer, there are must be defined further classes of VM architectures:

- hypervisor architecture;
- host-based virtualization;
- paravirtualization.

The hypervisor supports hardware-level virtualization and functions directly between the physical hardware and its OS, providing hypercalls for the guest OS and applications (Fig. 2). Hypervisors are usually to be divided into micro-kernel architecture and monolithic hypervisor architecture groups, depending on its functionality. A micro-kernel hypervisor includes basic functions (memory management and processor scheduling) while monolithic hypervisor architecture type implies also work with changeable components like devices drivers. Therefore, micro-kernel hypervisor code is smaller but monolithic hypervisor proves to be more effective and flexible.

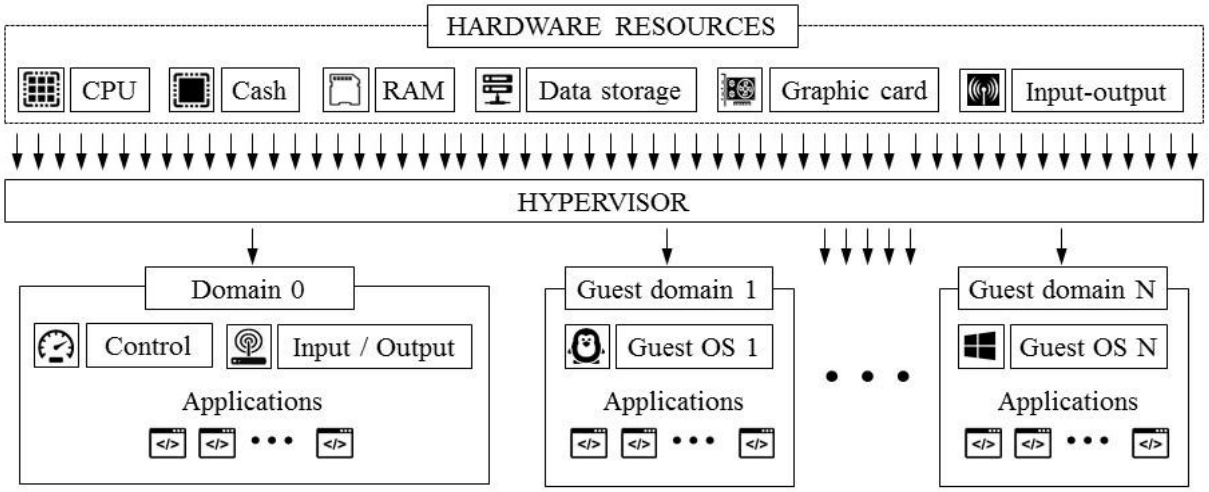

Fig. 2. Hypervisor-type virtual machine architecture

Host-based virtualization implies installation of virtualization layer on top of the host OS which is responsible for managing the hardware while guest OS should be installed on top of the virtualization layer and user applications will run on the VMs (Fig. 3). This architecture class has further advantages:

- installation this VM architecture without modifying the host OS, it simplify VM design and its deployment;
- host-based method appeals to various host machine configurations.

But it should be noticed that host-based while architecture has high flexibility, its performance is too low to be widely used.

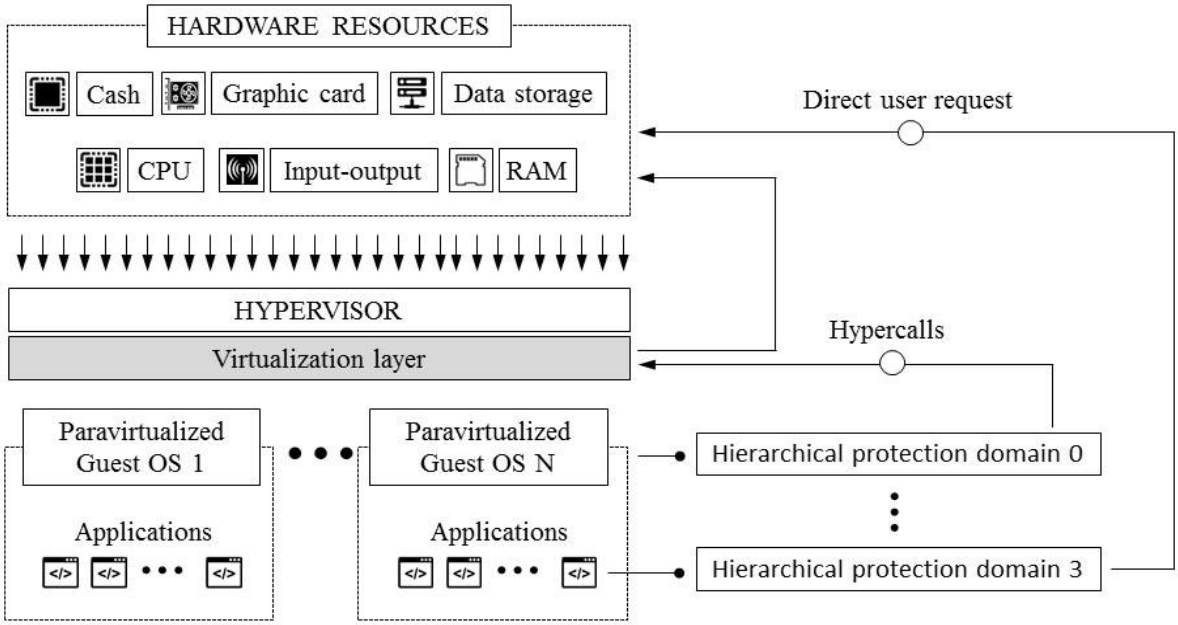

Fig. 3. Paravirtualized VM architecture, which implies modification of the guest OS

Paravirtualization method implies modifying the guest OS while it provides special APIs. Thereby performance decrease is a main problem of paravirtualization. The virtualization layer can be inserted at different positions in server software set. Performance could be improved by modifying only the guest OS. Figure 3 shows paravirtualized VM architecture where the guest OS are paravirtualized. This process must be assisted by compiler which replaces the OS instructions that cannot be virtualized by hypercalls. It should be noticed that lower the hierarchical protection domain (protection ring) number has to be associated with higher privilege of instruction to be executed. The OS manages the hardware and the instructions at domain 0, while user-level applications run at domain 3. It's important to mention that virtualized OS cannot on the hardware directly.

3. Hardware support for virtualization paradigm

To implement servers' hardware virtualization paradigm were developed hardware-assisted virtualization technology which included special running mode and instructions for x86 class Intel and AMD CPU. Hypervisor platform and guest OS should use different modes which are switchable on hardware level. This approach allowed to run multiple processes simultaneously. For system protection from crash all processors uses at least two modes (user mode and supervisor mode) which has to ensure controlled access of critical hardware. Instructions of supervisor mode are privileged ones, while other instructions are unprivileged ones [8, 9].

Figure 4 shows scheme of full virtualization techniques. For processor virtualization, there were used VT-x or VT-i techniques.

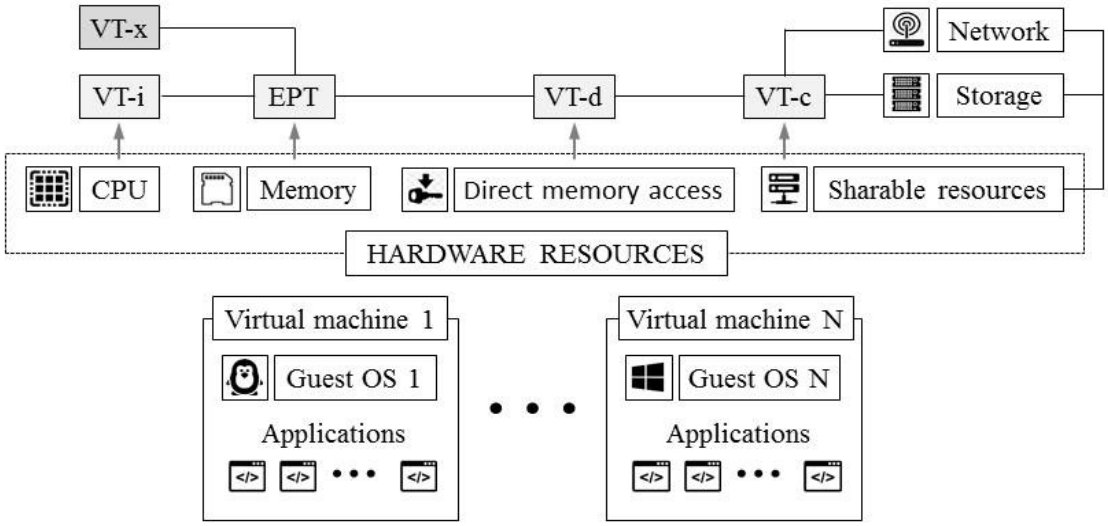

Fig. 4. Model of Intel hardware support for platform virtualization

VT-x adds a privileged mode and special processors' instructions. It was proposed for memory virtualization to use extended page tables (EPT), which allows to translate the virtual address to the machine's physical addresses for higher performance. For I/O virtualization was implemented virtualization technology for directed input/output VT-d, virtualization technology for connectivity VT-c and interrupt remapping as software capability for rerouting signals sent from peripheral devices (Fig. 5).

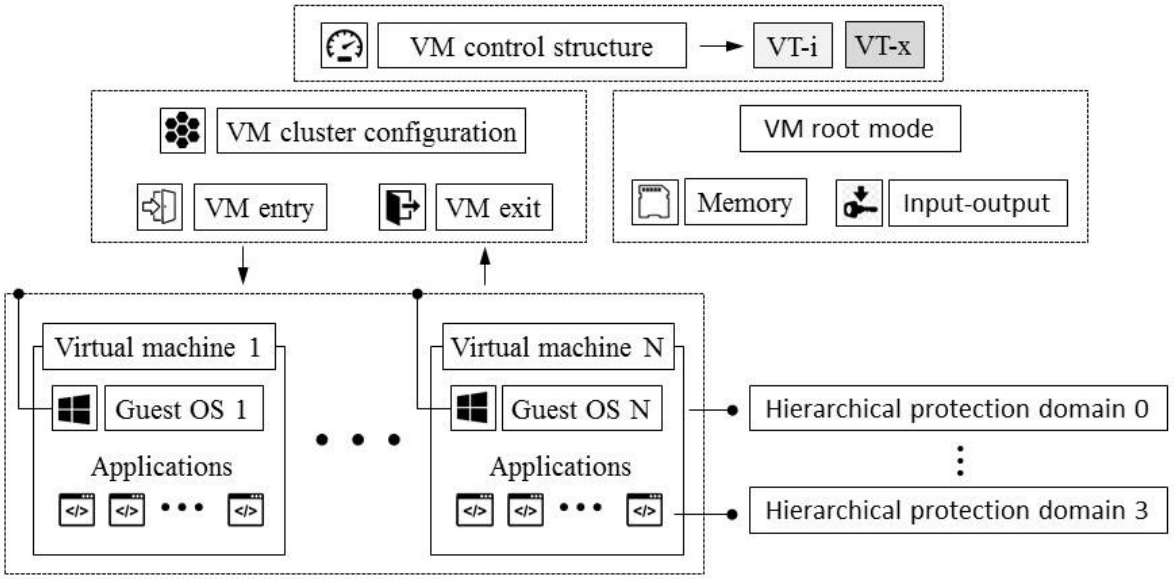

Fig. 5. Model of Intel hardware-assisted CPU virtualization

It's well known that x86 processors are not virtualizable primarily but Intel and AMD took great effort is taken to virtualize them. VT-x technology demonstrates hardware assisted virtualization class. Start and stop of a VM lifecycle and allocation of memory page is maintained by implementation of additional set. Hardware assisted virtualization model shows high efficiency but main problem to solve is still problem of binary translation performing. Paravirtualization systems often use a hybrid approach, so some tasks are loaded to the hardware, while other ones should be done in software environment.

Memory virtualization algorithm also includes special virtualization technique and to observe it we should define further terms:

- guest virtual addresses (GVA) as a virtual memory address of a process in guest OS;
- guest physical addresses (GPA) as a physical memory address in guest OS;
- host physical address (HPA) as a physical memory address of the host machine;
- memory management unit (MMU);
- translation lookaside buffer (TLB) as a tool used to optimize virtual memory performance.

Traditionally OS maintains mappings of virtual memory to machine memory by page tables (so called one-stage mapping). Modern x86 CPUs operates with MMU and TLB to optimize virtual memory performance. But virtual memory virtualization also involves sharing the physical system memory to allocate it to the physical memory of the VM components. In this case two-stage mapping process is to be used:

1. virtual memory to physical memory sharing;
2. physical memory to machine memory sharing.

MMU virtualization must be also supported as transparent to the guest OS process. The guest OS has to control the mapping of virtual addresses to the physical memory of VM but has no access the actual machine memory.

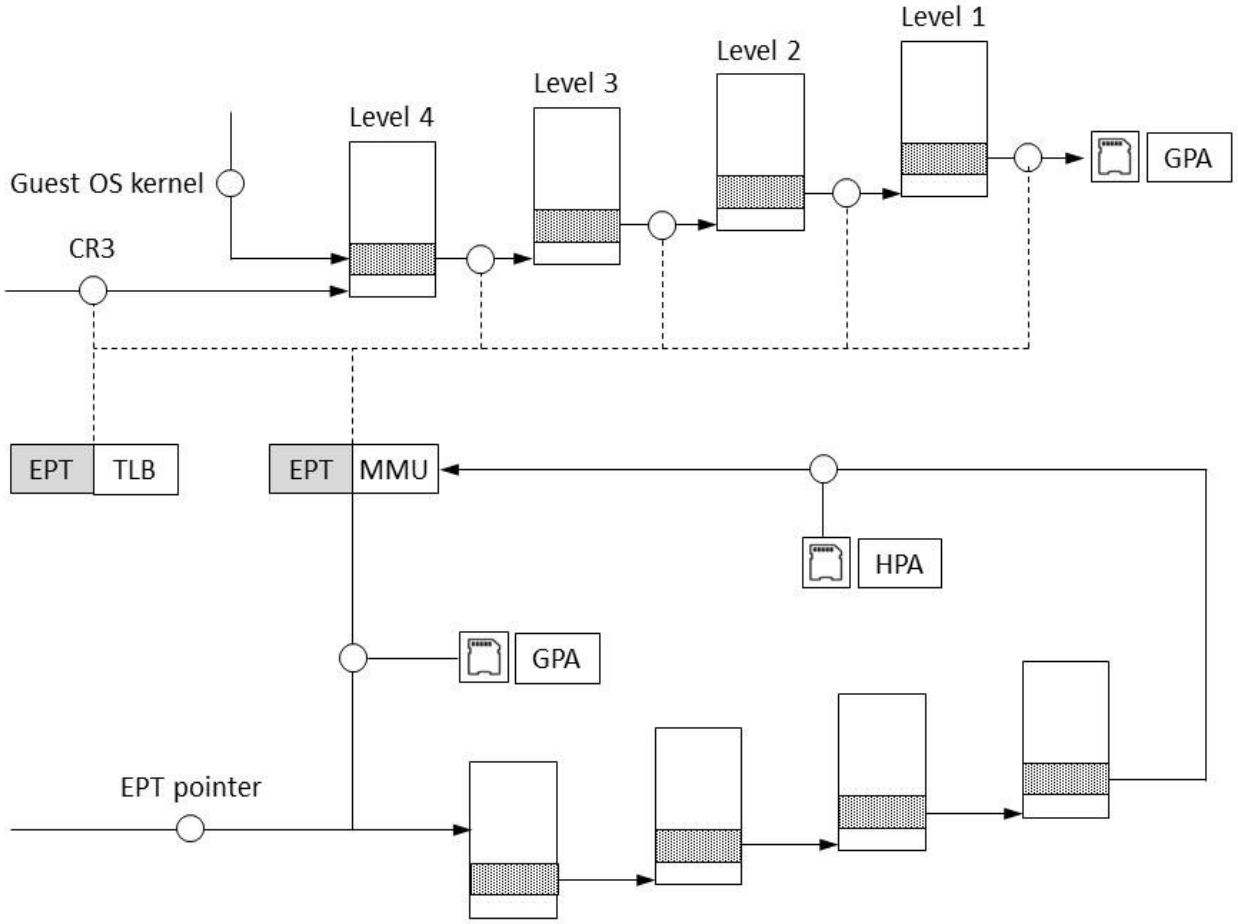

Fig. 6. Model of memory virtualization scheme using EPT

Hardware virtualization based EPT technique is shown at Figure 6. Up to the scheme page tables of the guest OS and EPT should has four-levels. When a virtual address has to be translated, the CPU will start from level 4 page table pointed to by Guest CR3 register and converts the Guest CR3 GPA to the HPA. CPU checks the EPT TLB to find translation and if there is no translation CPU will look for it in the EPT. If the there are no translation in the EPT, an EPT violation exception will run. At the next stage CPU should calculate the GPA of the level 3 page table by using the GVA and the content of the level 4 page table. If it will find a page fault, the CPU should generate a

page fault interrupt and then guest OS kernel will handle the interrupt. When the PGA of the level 3 page table is obtained, the CPU should look for the EPT to find the HPA of the level 3 page table and continue the procedure in same way.

Conclusions

Peculiarities of the modern information systems hardware resources virtualization were analyzed. It was shown that virtual machines concept provides opportunity for parallel computing, virtualization technology enables sharing of hardware resources by multiplexing virtual machines on the same server's farm. Software layer virtualizes the information system hardware into virtual resources which can be implemented at different operational levels: instruction set architecture level, hardware abstraction level; operating system level; user level API and application level. Depending on the virtualization layer, there are could be defined classes of virtual machine architecture: hypervisor architecture, host-based virtualization, paravirtualization. It was demonstrated that hypervisors should be divided into micro-kernel architecture and monolithic hypervisor architecture groups which are proves to be effective and flexible but requires a lot of resources. Host-based virtualization class advantages were shown, there are: installation without modifying the host operation system and various host machine configurations which could be adopted. But it was noticed that performance of this architecture is rather low. It was also demonstrated that paravirtualization method implies modifying the guest operation system and development special APIs set so virtualization layer can be inserted at different positions in server software. It was analyzed development hardware-assisted virtualization technology in order to implement servers' hardware virtualization paradigm. Analysis demonstrated virtualization algorithm includes special virtualization techniques and tools such as virtualization technology for directed input/output, virtualization technology for connectivity, interrupt remapping as software capability for rerouting signals sent from peripheral devices, memory management unit and translation lookaside buffer.

References

1. Meyer, S., Martini, N., & Witt, N. (2014) Virtualization: Analysis of different virtual machine solutions 128 p.
2. Kusnetzky, D. (2011). Virtualization: A managers guide. Sebastopol, CA: OReilly.
3. Papazoglou, M. (2012). Web services & SOA: Principles and technology. Essex, England: Pearson Education.
4. Cherkaoui, O., & Menon, R. (2014). Virtualization, Cloud, SDN, and SDDC in Data Centers. Data Center Handbook, 389-400. doi:10.1002/9781118937563.ch20
5. Nikolskiy, A. V., & Vasil'Ev, Y. S. (2015). Formal model of cyber attacks on virtualization tools and a measure of hypervisor vulnerability. Automatic Control and Computer Sciences, 49(8), 751-757. doi:10.3103/s014641161508012x
6. Elaffendi, M. A., & Alamudy, A. L. (2017). Could Virtualization be the Ultimate Solution for IoT Resource Constrained Devices Problem? A Multilevel Security Framework Based on Device Virtualization. 2017 International Conference on Computer and Applications (ICCA). doi:10.1109/comapp.2017.8079750
7. Wang, X., Sun, Y., Luo, Y., Wang, Z., Li, Y., Zhang, B., . . . Li, X. (2010). Dynamic memory paravirtualization transparent to guest OS. Science China Information Sciences, 53(1), 77-88. doi:10.1007/s11432-010-0008-x
8. Index. (2016). Intel Xeon Phi Processor High Performance Programming, 623-632. doi:10.1016/b978-0-12-809194-4.09985-3
9. Hamburger, V. (2016). Building VMware Software-Defined Data Centers. Birmingham: Packt Publishing.

Рецензія/Peer review : 11.1.2018 р. Надрукована/Printed :9.4.2018 р.
Рецензент :