Using Virtual Machine Monitors to Overcome the Challenges of Monitoring and Managing Virtualized Cloud Infrastructures

Mervat Adib Bamiah [1], Sarfraz Nawaz Brohi [2] and Suriayati Chuprat [3]
Advanced Informatics School, Universiti Teknologi Malaysia, Kuala Lumpur Malaysia

ABSTRACT

Virtualization is one of the hottest research topics nowadays. Several academic researchers and developers from IT industry are designing approaches for solving security and manageability issues of Virtual Machines (VMs) residing on virtualized cloud infrastructures. Moving the application from a physical to a virtual platform increases the efficiency, flexibility and reduces management cost as well as effort. Cloud computing is adopting the paradigm of virtualization, using this technique, memory, CPU and computational power is provided to clients' VMs by utilizing the underlying physical hardware. Beside these advantages there are few challenges faced by adopting virtualization such as management of VMs and network traffic, unexpected additional cost and resource allocation. Virtual Machine Monitor (VMM) or hypervisor is the tool used by cloud providers to manage the VMs on cloud. There are several heterogeneous hypervisors provided by various vendors that include VMware, Hyper-V, Xen and Kernel Virtual Machine (KVM). Considering the challenge of VM management, this paper describes several techniques to monitor and manage virtualized cloud infrastructures.

Keywords: Virtualization, Hypervisor, Hyper-V, VMware, KVM, Xen.

1. INTRODUCTION

Virtualization is a technique or methodology of logically dividing computer resources (hardware, software) into isolated VMs that execute instructions independent from the underlying Operating System (OS). Virtualization concept was first developed by IBM in 1960s to fully utilize mainframe hardware by logically partitioning them into VMs. These partitions will allow mainframe computers to perform multiple tasks and applications at the same time [1]. Issues such as maintenance, management and infrastructure cost as well as disaster protection lead to the invention of virtualization for x86 platforms. Virtualization dramatically improves efficiency and drives down overall IT cost [6]. Considering the flexibility of virtualization, cloud computing shifted towards virtualization where real resources such as hardware, software, desktop, network components or storages are partitioned into virtual images and offered to clients as on-demand services on pay-per-use pattern. A virtual cloud infrastructure is monitored by a hypervisor that allows multiple VMs to share a single hardware host. Each VM appears to have the host's processor, memory and other resources. However, the hypervisor controls the host processor and resources, allocates what is required to each VM in order to ensure that the guests run smoothly and independent of each other. Virtualization saves the cost for purchasing and maintaining new hardware resources. For-example a physical hard disk or RAM can be divided into several partitions and provided to several VMs. Cloud providers are residing the client applications on VMs that are created from underlying physical hardware. Since applications of each client are placed on a separated VM, it provides the advantage of isolation so if any VMs is malicious or affected with virus, will not affect the other VMs under the same provider whereas in a physical cloud infrastructure when clients' applications are placed on physical hardware there is possibility of various threats that can influence the entire cloud datacenter for-example a malicious client's interaction can affect the providers hardware with virus or security threats, that can result into demolition of complete cloud infrastructure. In order to implement VMMs on cloud infrastructures, normally cloud computing platforms are virtualized based on x86 or x64 architecture.

2. VIRTUALIZATION CHALLENGES

The paradigm of virtualization brought up numerous advantages to a cloud data center such as increased productivity and flexibility with reduced cost. However, it may also result into increased burden on a cloud infrastructure that may causes issues of decreased performance, availability, reliability of components and provided service. The main challenges of a virtualization are described as follows:

2.1. Managing a Virtualized Cloud Infrastructure

In a virtualized cloud infrastructure, each client is allocated with a VM that is running client specific applications. Since OS of cloud provider is running multiple VMs concurrently, it's a difficult task to manage all the VMs to keep the performance alive. Typical virtualization management tools are designed to provide insight only into

Fourth International Conference on Machine Vision (ICMV 2011): Machine Vision, Image Processing,
and Pattern Analysis, edited by Zhu Zeng, Yuting Li, Proc. of SPIE Vol. 8349, 83491M · © 2012 SPIE
CCC code: 0277-786X/12/$18 · doi: 10.1117/12.920880

the virtualized elements of the virtualized environment, not into the data center as a whole. Similarly, existing data center management tools are generally not aware of virtualization components such as VMs and VMMs. As a result, administrators must contend with multiple management tools and incomplete information, which can increase the time, cost, and complexity of managing virtualized infrastructures [3].

2.2. Resource Allocation

Running several VMs on single physical hardware increases server utilization but when number of VMs rapidly keeps on increasing, the computation burden on main OS will increase. For-example when several VMs request for the same I/O such as requesting a network card concurrently. The main OS will respond to one at a time so other VMs need to wait for the requested resource, in certain cases VMs will face starvation due to shared access of hardware. If the main server is switching between the VMs to provide accessibility of network card, it can result into reduced performance of network and may cause extra latency and delay in response time [3].

2.3. Unexpected Additional Cost

Virtualization has proven to be anti-hardware technology, where several images of physical machines are created virtually and used instead of purchasing new hardware that results in minimizing energy consumption and enhancing IT resources without being worried about additional cost. However, in order to solve manageability, performance and availability issues, sometimes providers need to buy additional hardware to keep the VMs alive. For-example when storage requirements of clients rapidly increases provider needs to buy new storage devices because it requires an enormous amount of capacity to create and store Virtual Machine Disk (VMDK) images that can increase from ten to thousand over time, it may result in unexpected additional cost that is not even preplanned [3].

2.4. Network Traffic

Server virtualization platforms offer many advanced capabilities such as live VM migration, virtual software switching and support for virtual LAN segmentation on existing network infrastructures. However, it may not be equipped to support such features. Server virtualization can dramatically increase data storage traffic. For-example, passing large amounts of data from multiple VMs through one host storage network connection can lead to serious storage traffic congestion. Also, moving large VMDK images over WAN connections can be slow and interfere with other traffic. Virtualized environments can offer significant improvements in data center productivity and flexibility if administrators successfully take appropriate steps for creating and maintaining a virtualized cloud infrastructure [3].

3. MACHINE MONITORS FOR MANAGING VIRTUALIZED CLOUD INFRASTRUCTURES

In order to maintain concurrent access of VMs, cloud providers are using hypervisors that are actually tools developed for monitoring and managing VMs such as providing virtual memory, CPU scheduling, network interfaces and maintaining safe as well as efficient operating environment [2]. Hypervisor or VMM is a software-abstraction layer that partitions a hardware platform into one or more virtual machines. Each of these virtual machines was sufficiently similar to the underlying physical machine to run existing software unmodified [4]. Generally hypervisors are divided into two categories i.e. Type 1 and 2. Type-1 hypervisor runs directly upon the hardware with a separated layer from the host OS and Type-2 hypervisor runs together with the host OS [5]. Due to the isolation from the host OS, the security, performance and scalability features in Type-1 are enhanced than Type-2. The architectures of VMM and VM managing as well as monitoring techniques are described as follows:

3.1. Xen Hypervisor

Xen is a Type-1 hypervisor. In order to create a secure operating environment, Xen hypervisor runs guest VMs in isolated environments called as domains i.e. Domain0 (Dom0) and DomainU (DomU) due to the accessibility privileges. When Xen boots, one of the first things it does is to load a dom0 guest kernel. This is typically specified in the boot loader as a module and can be loaded without any file system drivers being available. Dom0 is the first guest to run and has higher privileges [1]. DomU guests have lower privileges and can't access the hardware. Xen does not include any device drivers or user interface by itself, these all are provided by the OS and user space tools running in dom0 guest which is typically a Linux modified kernel. The most obvious task performed by the dom0 guest is to handle I/O operations requested by domU guest as shown in figure-1 [7]. Dom0 and domU communicate by using device drivers. Since dom0 guest runs at a higher level of privilege than domU it can access the hardware directly. For this reason, it is vital that the privileged guest should be properly secured. As I/O requests are passed to dom0, memory management and CPU scheduling are the responsibility of Xen hypervisor. The mechanism used by Xen hypervisor to assign virtual memory to VMs is referred as Memory Overcommit. Using this technique, it is possible to provide virtual memory more than actual physical memory. For-instance if a provider has 6GB of physical RAM and wants to allocate to VMs each of them consuming 1GB simultaneously, maximum it can be assigned to five VMs because 1GB will be reserved for the operations of dom0 guest, but with the use of overcommit technique it's possible to run six or even more than ten VMs using 1GB simultaneously [8]. Xen architecture is designed by considering security of guests VMs, for-example all domains are isolated from each other so if any of the domains is affected with virus or it is malicious it will not affect the other domains. Secondly the privilege levels also play an integral role to keep the operating environment safe and secure, as using Xen service model domU guest is not allowed to directly interact with physical hardware so it keeps the underlying physical hardware safe from the side effects of clients' VMs. Beside service model Xen now uses pass-through model. Under this model domU guests are allowed to interact directly with hardware but still restricted to certain hardware devices according to their privileges [9].

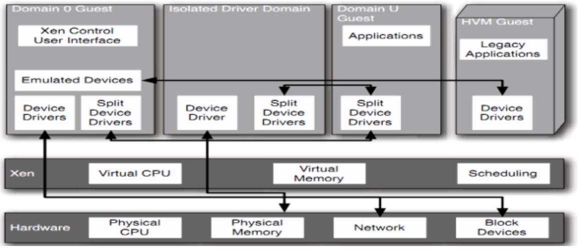

Fig. 1: Xen Hypervisor.

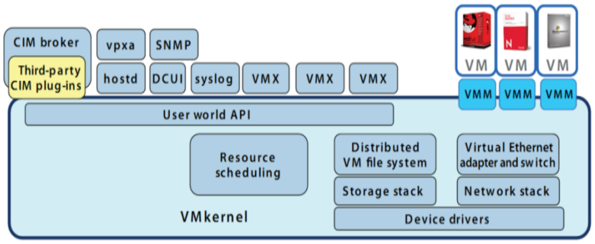

Fig. 2: ESXi Hypervisor.

3.2. VMware ESXi

ESXi is a proprietary hypervisor. The architecture of ESXi consists of an OS called VMkernel and processes running on top of it. VMkernel is an OS developed by VMware to manage and execute all the applications, agents and VMs. VMkernel includes various OS features such as resource scheduling, I/O stack and device drivers. The main components of ESXi hypervisor include Direct Console User Interface (DCUI), VMM and Common Information Module (CIM) as shown in figure-2 [10]. DCUI is the local user interface that is displayed only on the console of an ESXi system. It provides a BIOS-like, menu-driven interface for interacting with the system. Its main purpose is initial configuration and troubleshooting. One of the components in VMkernel is *dcui*, which is used by the DCUI process to identify itself when communicating with other components in the system. The VMM in VMware is the process that provides the execution environment for a VM as well as a helper process known as VMX. The memory management technique used by ESXi is similar to Xen. It also uses memory overcommit technique that provides illusion of providing virtual memory more than physical memory. ESXi is using this technique in such a way that if a domain is idle or nearly so, is probably not using much memory that can be available to be used in another domain or for a newly created domain. This dynamic memory allocation enables the provider to run as many as VMs without being worried about physical memory requirements [11].

VMware ESXi is designed with powerful security techniques to protect the data on a virtual cloud infrastructure. ESXi works by using VMsafe, is a new security technology that helps protect virtualized workloads in a way that were previously not possible with physical machines. VMsafe provides a set of security APIs that enable third-party security products to gain the same visibility as VMware ESX or ESXi into the operation of a VM to identify and eliminate malware, such as viruses, trojans and key-loggers. VMware ESX and ESXi are protected from common attacks and exploits by assuring the integrity of the VMkernel, a core component of the ESXi hypervisor. Disk integrity techniques in ESX and ESXi protect the boot-up of the hypervisor by utilizing the Trusted Platform Module (TPM), a hardware device embedded in servers. ESXi is also using SSL encryption to ensure secure connections [10].

3.3. Kernel Virtual Machine (KVM)

The process of managing the VMs on cloud is similar to management of several simple applications running on a single OS concurrently. Instead of applications, there are VMs that are further running several applications simultaneously. If VMs considered as OS processes, then virtualized cloud infrastructure can be monitored and maintained as a general computing infrastructure by using any OS kernel. KVM actually considers VMs as simple Linux processes so they have modified Linux OS into a hypervisor as shown in figure-3 [12]. Since Linux is an open source OS and it has several OS components such as memory manager, process scheduler, I/O stack device drivers and security manager that are actually required for the implementation of a hypervisor, KVM is designed by turning the Linux kernel into a hypervisors. As Xen has two domains for VMs, similarly Linux OS divides the guest VMs in two modes (user, kernel) according to the access privileges. The user mode is considered as unprivileged and kernel mode is considered as privileged process. By default Linux starts in user mode and it changes to kernel mode when required. KVM is developed by adding one more extra to the existing modes of execution i.e. guest mode as shown in figure-4 [13]. The guest mode itself has two normal modes user and kernel, can be called as guest-user and guest-kernel mode. When a guest process is executing non-I/O guest code, it will run in guest-user mode. In guest-kernel mode, the process handles exits from guest-user mode due to I/O or other special instructions. In user mode, the Linux process performs I/O on behalf of a guest. In the KVM model each guest VM is implemented as a simple Linux process and that process itself is able to run multiple applications concurrently because it is acting as a virtual OS.

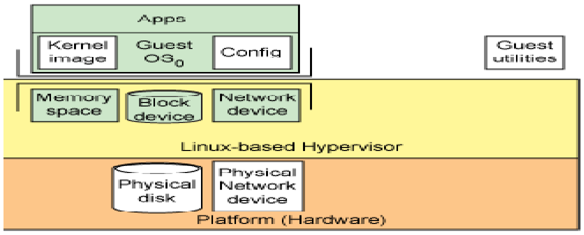

Fig. 3: KVM hypervisor.

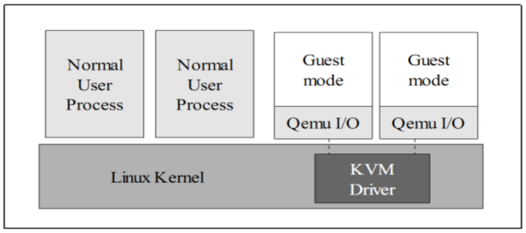

Fig. 4: KVM guest mode.

Each VM is scheduled by standard Linux scheduler. Using KVM, the I/O requests of a guest VM are handled through Quick Emulator (QEMU). It is a platform virtualization solution that allows virtualization of an entire PC environment including (disks, graphic adapters, BIOS, PCI bus, USB and network devices). KVM inherits the powerful memory management and security mechanism of Linux OS. The memory for a VM is stored as memory for any other Linux process and can be swapped, backed by large pages for better performance. Memory management in Linux is supported by Non-Uniform Memory Access (NUMA) technology. It allows VMs to efficiently access large amounts of memory. Since we already discussed, in KVM a VM is implemented as a Linux process, hence the security of VM is based on the standard Linux security policies. The Linux kernel includes Security Enhanced Linux (SELinux) a project developed by the US National Security Agency to add mandatory access controls, multi-level and multi-category security as well as policy enforcement [14].

3.4. Hyper-V

Hyper-V is implemented as a role in Windows Server 2008 and runs multiple OSs on VMs resides on single physical server that maximizes the utilization of server hardware. Hyper-V is free hypervisor-based virtualization technology from Microsoft which is integrated into all Dell supported Windows Server 2008 x64 Editions OSs. Hyper-V enables server consolidation by maintaining the logical OS and application isolation required while sharing physical system resources efficiently without acquiring, maintaining, powering, cooling or administering additional physical hardware [15]. Hyper-V holds the hypervisor smaller (less than 1MB) by using an administrative parent OS that has all of the drivers for better performance

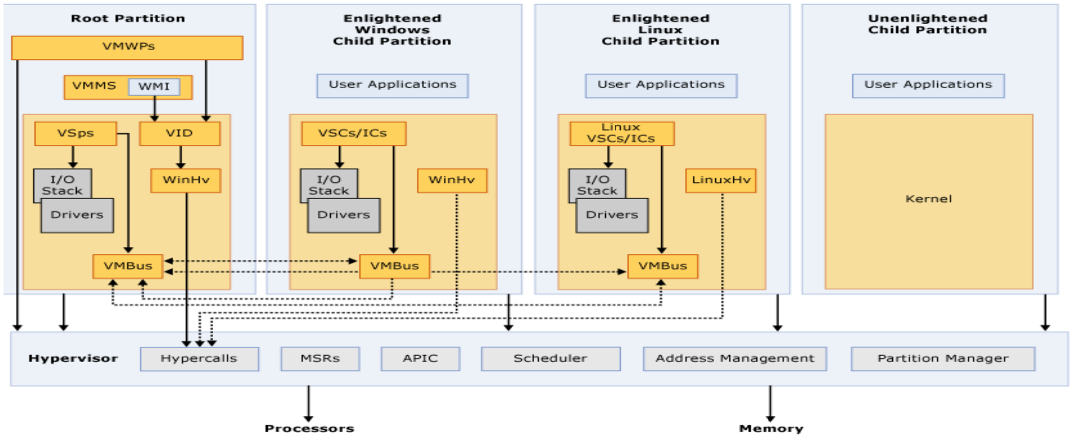

Fig. 5: Hyper-V Hypervisor.

. Hyper-V has a weakness that it does not support live migration that prevents crash (the ability to move a VM from one physical host over to another while powered up and server can be moved over and running full time) which have been enhanced and added in version R2. The hyper-V architecture is based on micro- kernelized hypervisors which depend on a single VMM that is code heavy. The VMM emulates and handles all lower level accesses and therefore is also slower. As shown in figure-5 [15], the guests OSs need to request hardware access from the hypervisor as the device drivers are integrated into the hypervisor. The hypervisor as a micro-kernel provides limited functionality such as managing memory address space, process communication and management while other hardware management tasks are typically managed by processes independent of the kernel. It does not contain device drivers that are present in the OS in the parent or root partition. In hyper-V each VM run in a separate logical partition. A root or parent partition contains the Windows 2008 x64 OS which is responsible for creating the other partitions that contain different guest OSs. This parent partition contains a virtualization stack which is combination of user and kernel level code and interfaces with the hypervisor and the child partitions. The virtualization stack manages memory for child partitions and has direct access to the hardware resources that creates partitions using calls to the hypercall application programming interface. The hypervisor monitors processor interrupts and guest OSs do not have access to the processor. The hypervisor enforces memory, CPU usage and access specifications. The parent partition manages the I/O devices such as keyboard, mouse, printer and other devices connected to the hypervisor. It handles access to devices through virtual devices. Device virtualization is achieved through the use of a provider-consumer model. The child partitions containing guest OSs request the virtual devices through the virtual memory bus. Access to the physical devices is controlled by the hypervisor. Hyper-V improve security by providing the ability to disable the execution bit preventing viruses and worms from executing by avoiding the need to load third-party drivers in the hypervisor as it is micro-kernelized [15].

4. CONCLUSION AND FUTURE WORK

Several VMM techniques are available in industry for managing the virtualized cloud infrastructures. The most efficient and industry required VMM includes Xen, VMware, KVM and Hyper-V. The architecture of every VMM is different from each other. For-example Xen divides the VMs in separate environments called domains such as dom0 and domU whereas KVM divides the VMs in guest-user and guest-kernel modes according to privilege accesses. On the other hand Hyper-V divides the VMs into logical partitions i.e. parent and child and VMware using its own OS VMkernel. It depends on the cloud providers to use any of these hypervisors according to their requirements and suitability of infrastructure. A well managed virtual cloud infrastructure can offer significant improvements in data center such as productivity and flexibility. Considering the current contributions in field of cloud virtualization, it is obvious that there are still tremendous opportunities for industry developers and academic researchers to contribute in this field and bring their innovative and valuable contributions to IT industry.

REFERENCES

[1] Xiantao Zhang and Yaozu Dong, “Optimizing Xen VMM Based on Intel® Virtualization Technology,” in Internet Computing in Science and Engineering, 2008. ICICSE ’08. International Conference on, 2008, pp. 367-374.

[2] Wei Chen, Hongyi Lu, Li Shen, Zhiying Wang, Nong Xiao, and Dan Chen, “A Novel Hardware Assisted Full Virtualization Technique,” in Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for, 2008, pp. 1292-1297.

[3] Dell, "Overcoming 7 Key Challenges to Virtualization", http://www.dell.com/downloads/global/power/ps1q09-50090198-F5.pdf, 2009.

[4] M. Rosenblum and T. Garfinkel, “Virtual machine monitors: current technology and future trends,” Computer, vol. 38, no. 5, pp. 39-47, 2005.

[5] T. Naughton, G. Vallee, S. L. Scott, and F. Aderholdt, “Loadable Hypervisor Modules,” in System Sciences (HICSS), 2010 43rd Hawaii International Conference on, 2010, pp. 1-8.

[6] S. J. Vaughan-Nichols, “New Approach to Virtualization Is a Lightweight,” Computer, vol. 39, no. 11, pp. 12-14, 2006.

[7] David Chisnall, “A definitive guide to Xen hypervisor”, Prentice Hall, 2008.

[8] Dan Magenheimer, "Memory overcommit", http://xen.org/index.php/2008/08/27/xen-33-feature-memory, 2008.

[9] Amit Aneja, “Xen Hypervisor: Designig Embeded Virtualized Intel Architecture”, http://download.intel.com/design/intarch/PAPERS/325258.pdf, 2011.

[10] Charu Chaubal, “Architecture of VMware ESXi”, http://www.vmware.com/files/pdf/vmware_esxi_architecture_wp.pdf, 2008.

[11] I. Ahmad, J. M. Anderson, A. M. Holler, R. Kambo, and V. Makhija, “An analysis of disk performance in VMware ESX server virtual machines,” in Workload Characterization, 2003. WWC-6. 2003 IEEE International Workshop on, 2003, pp. 65-76.

[12] M. Tim Jones, “Anatomy of KVM Hypervisor”, http://www.ibm.com/developerworks/linux/library/l-hypervisor/index.html?ca=dgr-lnxw06Lnx-Hypervisor&S_TACT=105AGX59&S_CMP=grlnxw06, 2009.

[13] Qumranet, “KVM: Kernel-based Virtualization Driver”, http://www.linuxinsight.com/files/kvm_whitepaper.pdf, 2006.

[14] Redhat, “Kernel Based Virtual Machine”, http://www.redhat.com/f/pdf/rhev/DOC-KVM.pdf , 2009.

[15] Naveed Alam, “Survey on Hypervisors”, http://salsahpc.indiana.edu/b534projects/sites, /default/files/public/6_Survey%20On%20Hypervisors_Alam%20Naveed%20Imran.pdf, 2009.