Scalable Systems for the Cloud

Olivier Roques

2018

Contents

1 Virtualization

1.1 Definitions

VMM: an entity specifically responsible for virtualizing a given architecture, including the instruction set, memory, interrupts, and basic I/O operations.

Hypervisor: combines a kernel with a VMM. The kernel contains a boot loader, stacks for storage and networking, memory and CPU schedulers... To run a VM, the kernel loads the VMM, which encapsulates the details of virtualizing the x86 architecture.

VM: a tightly isolated software container with an operating system (OS) and an application inside. The VM executes directly on top of the VMM, touching the only through the VMM surface area.

Hosted architecture: installs and runs the virtualization layer (hypervisor) as an application on top of an operating system and supports the broadest range of hardware configurations.

Bare-metal architecture: installs the virtualization layer directly on a clean x86-based system.

1.2 CPU Virtualization

Virtualizing the x86 architecture requires placing a virtualization layer under the operating system (which expects to be in the most privileged Ring 0) to create and manage the virtual machines that

deliver shared resources. However some sensitive instructions can't effectively be virtualized as they have different semantics when they are not executed in Ring 0.

Binary Translation. This approach translates kernel code to replace nonvirtualizable instructions with new sequences of instructions that have the intended effect on the virtual hardware. Meanwhile, user level code is directly executed on the processor for high performance virtualization. The guest OS is not aware it is being virtualized and requires no modification.

Paravirtualization. Paravirtualization involves modifying the OS kernel to replace nonvirtualizable instructions with hypercalls that communicate directly with the virtualization layer hypervisor. The hypervisor also provides hypercall interfaces for other critical kernel operations such as memory management, interrupt handling and time keeping.

Hardware Assisted Virtualization. Intel Virtualization Technology (VT-x) and AMD's AMD-V target privileged instructions with a new CPU execution mode feature that allows the VMM to run in a new root mode below ring 0. Privileged and sensitive calls are set to automatically trap to the hypervisor.

1.3 Memory Virtualization

The guest OS controls the mapping of virtual addresses to the guest memory physical addresses, but the guest OS cannot have direct access to the actual machine memory. The VMM is responsible for mapping guest physical memory to the actual machine memory, and it uses shadow page tables to accelerate the mappings. The VMM uses TLB hardware to map the virtual memory directly to the machine memory to avoid the two levels of translation on every access.

1.4 Network Virtualization

Emulation: implements a virtual NIC, vNIC's registers are variables in Hypervisor's memory. Memory is write-protected and hypervisor reacts based on the values being written. Emulation is very slow because of the traps on the many register access.

Paravirtualization: adds virtual NIC driver into guest (front-end) and implements the vNIC in hypervisor (back-end). Everything works just like in the emulation case except the protocol between front-end and back-end which is more efficient: guest does a hypercall and sends start address and length to the hypervisor. Paravirtualization requires guest to have a specific driver.

Direct Device Assignment: pulls NIC out of the host, and plugs it into the guest. The guest is allowed to access NIC registers directly and doesn't need hypervisor intervention. However the host can't access the NIC anymore and one NIC per guest is required.

IOMMU: memory management unit (MMU) that connects a direct-memory-access–capable I/O bus to the main memory. Like a traditional MMU, which translates CPU-visible virtual addresses to physical addresses, the IOMMU maps device-visible virtual addresses (also called device addresses or I/O addresses in this context) to physical addresses. It allows the hypervisor to arrange things such that device use I/O virtual addresses instead of physical addresses for their DMA operations.

SR-IOV: ability of a device to appear to software as multiple devices. The device contains a physical function controlled by the host, used to create virtual functions. Each virtual function is assigned to a guest which uses it to process packets. SR-IOV requires new hardware.

Intel Data Direct I/O: part of the Intel I/O Acceleration Technology suite of hardware features to increase performance. DDIO enables I/O directly in and out of the processor's last level cache. Neither the driver nor the NIC hardware need to do anything, the support is integrated in the platform, invisible to software. DDIO increases bandwidth, reduces power consumption, and reduces latency.

Data Plane Development Kit: sets user-space drivers and libraries for fast packet processing: all traffic bypasses the kernel and thus avoids interrupts. Drawbacks are security, requirement of using a specific API in applications, huge pages.

1.5 Storage Virtualization

Storage is virtualized by emulating multiple logical devices from a single physical device. A VMDK (Virtual Machine Disk) file represents a virtual disk as seen by the virtual machine. Multiple filesystem layers introduces problems: double journaling, reduced performances.

1.6 Containers

Container refers to an operating system feature in which the kernel allows the existence of multiple isolated user-space instances. A computer program running on an ordinary operating system can see all resources (connected devices, files and folders, network shares, CPU power, quantifiable hardware capabilities) of that computer. However, programs running inside a container can only see the container's contents and devices assigned to the container.

chroot. A chroot on Unix operating systems is an operation that changes the apparent root directory for the current running process and its children. A program that is run in such a modified environment cannot name (and therefore normally cannot access) files outside the designated directory tree.

Kernel Namespaces. Namespaces are a feature of the Linux kernel that partitions kernel resources such that one set of processes sees one set of resources while another set of processes sees a different set of resources.

cgroups. Control group is a Linux kernel feature that limits, accounts for, and isolates the resource usage (CPU, memory, disk I/O, network, etc.) of a collection of processes.

1.7 Serverless

Function as a service (FaaS) is a category of cloud computing services that provides a platform allowing customers to develop, run, and manage application functionalities without the complexity of building and maintaining the infrastructure typically associated with developing and launching an app. Building an application following this model is one way of achieving a "serverless" architecture, and is typically used when building microservices applications.