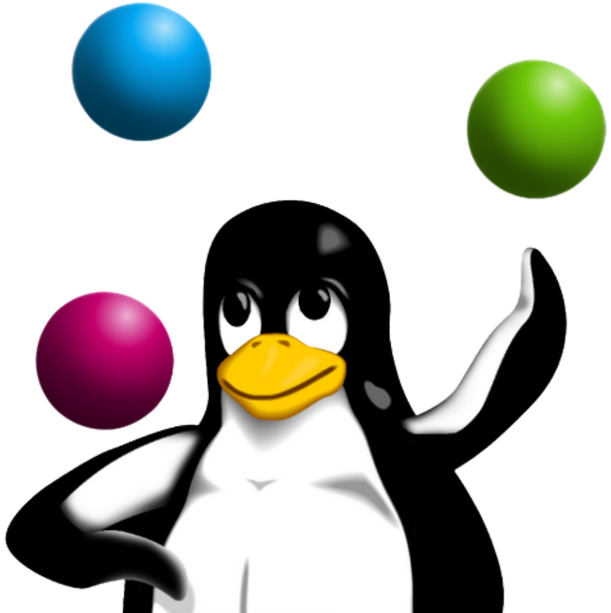
KVM

Kernel-based Virtual Machine



Team Members

Onkar Gavali

Vinayak Gaikwad

Akanksha Chaudhari

Anushka Joshi

Aashita Gupta

Prathamesh Mehare

Amaan Shaikh

Shreyash Malu

Aditya Sarnobat

Vaishnavi Mahajan

Devang Kamble

Sayali Desai

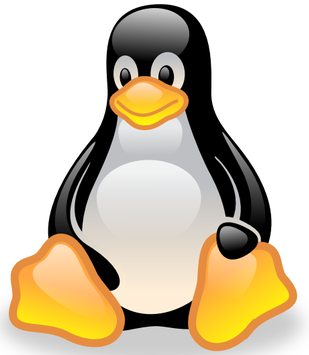
Suyash Chavan

Pratik Raut

Today's Agenda

1. Introduction and History
2. Architecture
3. Working
4. Implementation
5. Features
6. Management
7. KVM V/S XEN
8. Conclusion

Introduction of KVM



1. Kernel-based Virtual Machine (KVM) is an open source virtualization technology built into Linux

2. KVM turns a Linux host machine into a hypervisor

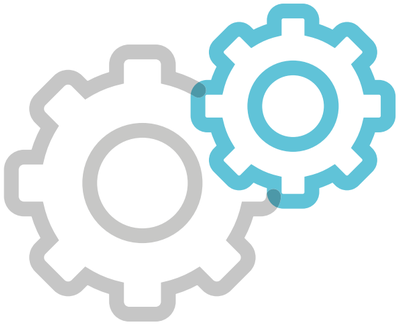
3. Hypervisor allows a host machine to run multiple, isolated virtual environments called guests or virtual

4. KVM is part of existing Linux code, it immediately benefits from every new Linux features, fix, and advancement without additional engineering.

Need

- When the user wants to modify the host systems processor so that it could be easy to communicate with the hypervisor and thus support the virtualization.
- Used to remove the need for binary translation and emulation.
- Used when one goes with full virtualization rather than para-virtualization.
- Used when modifying the host operating system so that it could support the virtualization. It results in native hypervisors rather than hosted ones.

Histroy of KVM



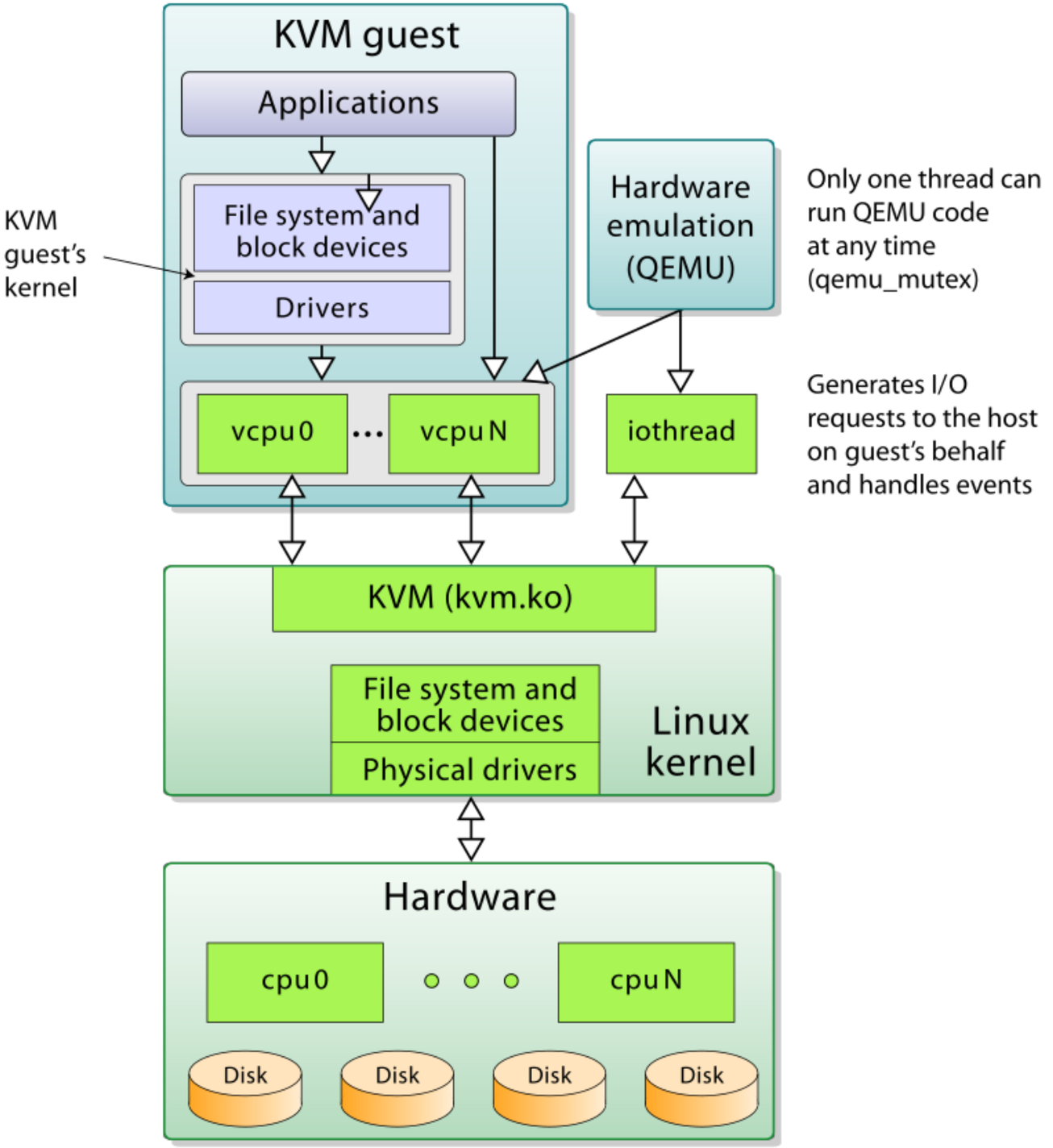
1. Qumranet began the development of KVM in mid-2006.

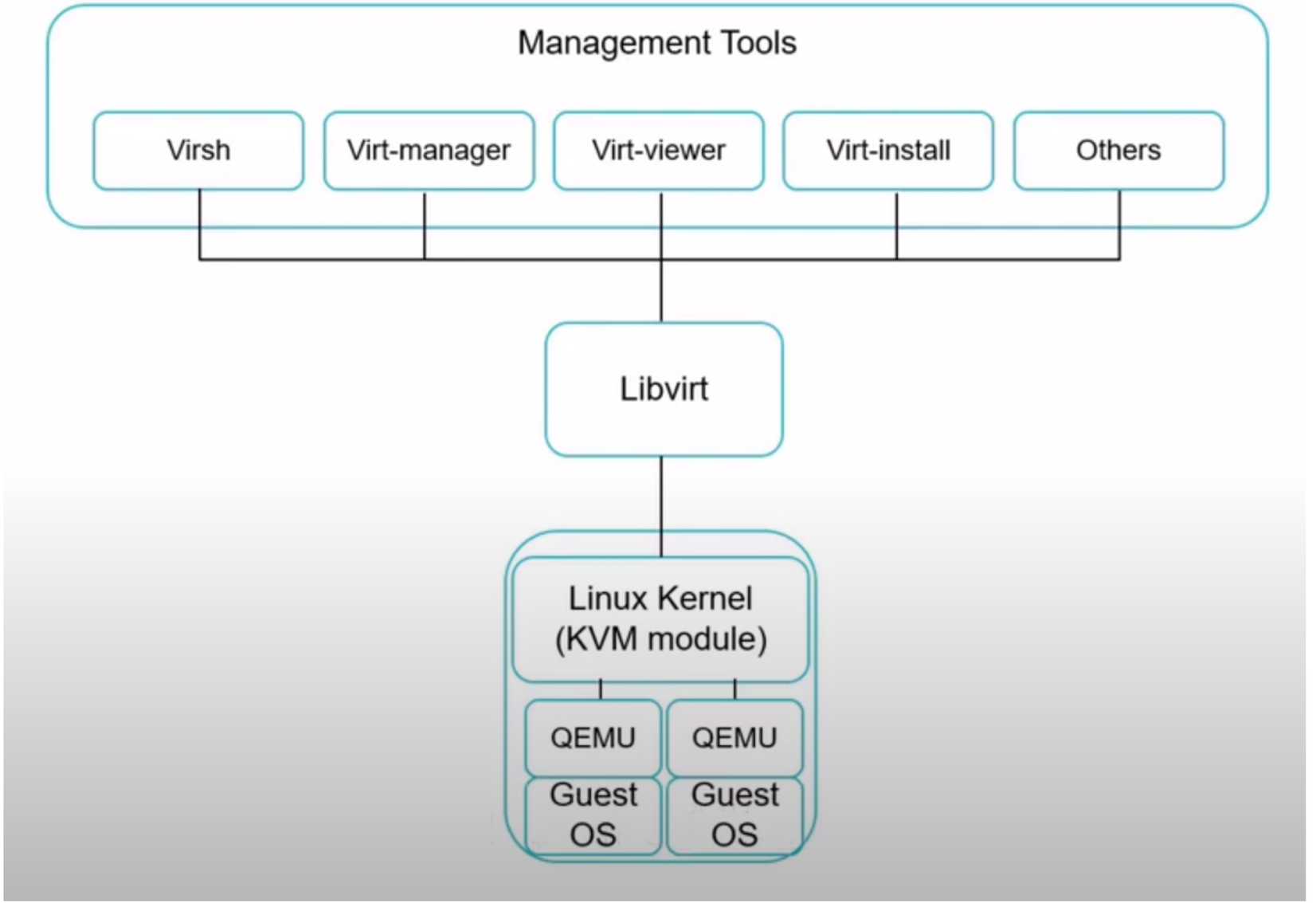
2. KVM surfaced in October 2006 and was merged into the Linux kernel mainline in kernel version 2.6.20, which was released on 5 February 2007.

3. It was acquired by Red Hat in 2008.

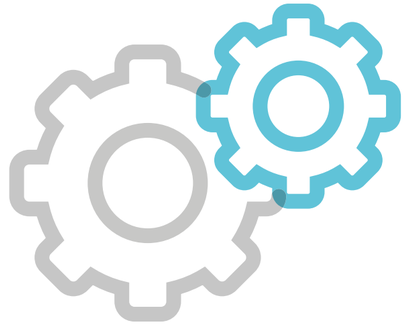
4. KVM is maintained by Paolo Bonzini.

Architecture


KVM guest
Applications
File system and block devices
Drivers
KVM guest's kernel
vcpu 0
vcpu N
Hardware emulation (QEMU)
Only one thread can run QEMU code at any time (qemu_mutex)
iothread
Generates I/O requests to the host on guest's behalf and handles events
KVM (kvm.ko)
File system and block devices
Physical drivers
Linux kernel
Hardware
cpu 0
cpu N
Disk
Disk
Disk
Disk


Management Tools
Virsh
Virt-manager
Virt-viewer
Virt-install
Others
Libvirt
Linux Kernel
(KVM module)
QEMU
QEMU
Guest
OS
Guest
OS

How KVM Works



1. KVM converts Linux into a type-1 (bare-metal) hypervisor

2. All hypervisors need some operating system-level components. KVM has all these components because it’s part of the Linux kernel

3. Every VM is implemented as a regular Linux process, scheduled by the standard Linux scheduler, with dedicated virtual hardware like a network card, graphics adapter, CPU(s), memory, and disks.

Implementation

- Requirements:
    a. Machine: X86 Hardware with Virtualization capabilities
    b. Modules: Host kernel module and processor-specific module

- Implementing KVM on a supported Linux distribution expands KVM's capabilities, letting you swap resources among guests, share common libraries, and optimize system performance.

XEN V/S KVM

• VMM implementation of its own hypervisor	• Kernel Module
• Kernel as I/O dispatcher and management domain	• Uses Kernel as VMM
• Maintained and supported as a patch to mainline kernel	• Maintained and supported as a patch to the upstream kernel
• Supports fully virtualized and paravirtualized VMs	• Supports only fully virtualized VMs

Features

Security

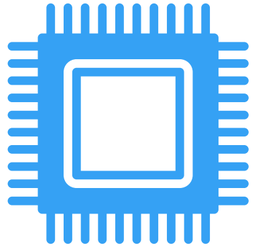
- Uses SELinux + sVirt for enhanced VM security and isolation.
- SELinux (security-enhanced Linux) establishes security boundaries around VMs.
- sVirt (secure virtualization) extends SELinux’s capabilities.
- Mandatory Access Control (MAC) security is applied to guest VMs and prevents manual labeling errors.

Features

Storage

- Able to use any storage supported by Linux
- Use of Multipath I/O to improve storage and provide redundancy.
- KVM also supports shared file systems.
- Disk images support thin provisioning, allocating storage on demand.

Features



Hardware Support

- KVM can use a wide variety of certified Linux-supported hardware platforms.
- Hardware vendors regularly contribute to kernel development.
- The latest hardware features are often rapidly adopted in the Linux kernel.

Features

Memory management

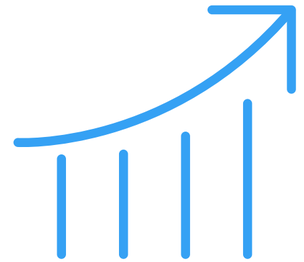
- Inherits the memory management features of Linux, including non-uniform memory access and kernel same-page merging.
- The memory of a VM can be swapped, backed by large volumes for better performance, and shared or backed by a disk file.

Features

Live migration

- It is the ability to move a running VM between physical hosts with no service interruption.
- The VM remains powered on, network connections remain active, and applications continue to run while the VM is relocated.
- KVM also saves a VM's current state so it can be stored and resumed later.

Features



Performance and scalability

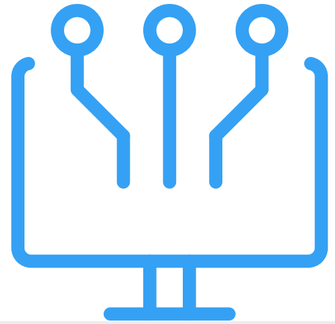
- KVM inherits the performance of Linux, scaling to match demand load if the number of guest machines and requests increases.
- It is the basis for many enterprise virtualization setups, such as datacenters and private clouds (via OpenStack®).

Features

Scheduling and resource control

- A VM is a Linux process, scheduled and managed by the kernel.
- The Linux scheduler allows fine-grained control of the resources allocated to a Linux process and guarantees a quality of service for a particular process.
- In KVM, this includes the completely fair scheduler, control groups, network namespaces, and real-time extensions.

Features



Lower latency and higher prioritization

- The Linux kernel features real-time extensions that allow VM-based apps to run at lower latency with better prioritization (compared to bare metal).
- The kernel also divides processes that require long computing times into smaller components, which are then scheduled and processed accordingly.

Pros & Cons

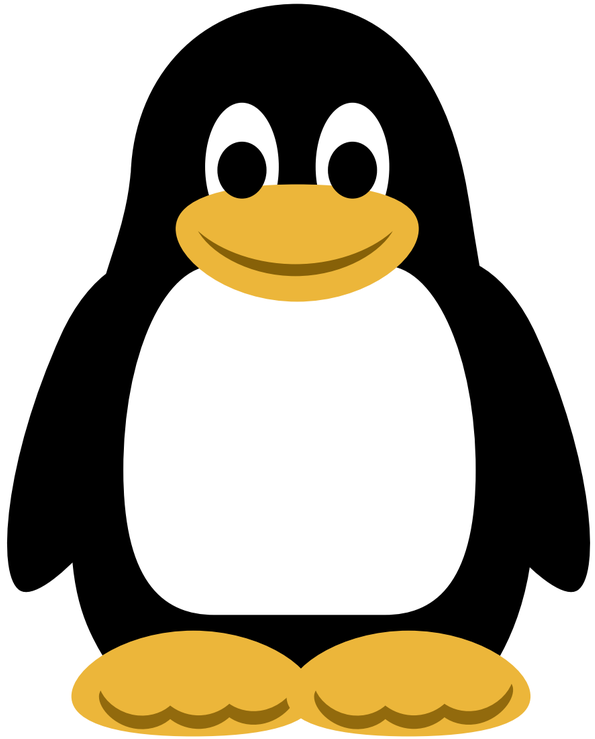
Pros

- KVM comes in-built in most of Linux distributions.
- KVM is open source and hence free to use.
- KVM ensures performance and stability.
- Libvirt makes administration automated.

Cons

- KVM is available only for Linux distributions.
- The host computer needs to be powerful enough to take the load of multiple VMs.
- Since the hardware is centralized, hence the risk of losing data in case of system failure increases exponentially.

To Summarise...



Thank you!