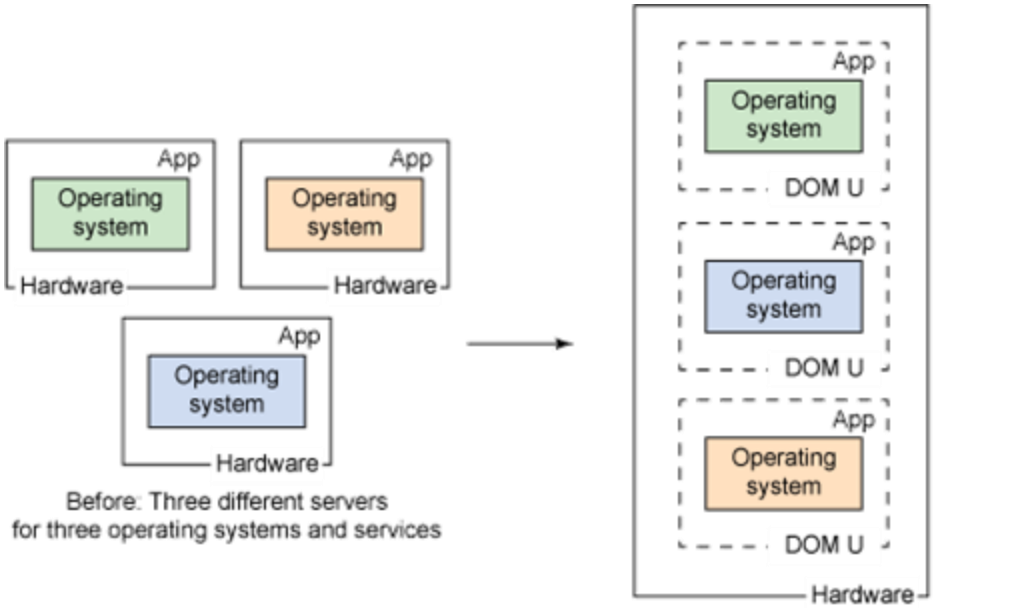
Обзор гипервизоров

author: Fedoseev V.E

От физического к логическому



Технологии виртуализации

- изоляция от аппаратного обеспечения и операционных систем
- оптимального использования системных ресурсов
- надежность и отказоустойчивость
- масштабируемость и гибкость

Who is who?

– – –

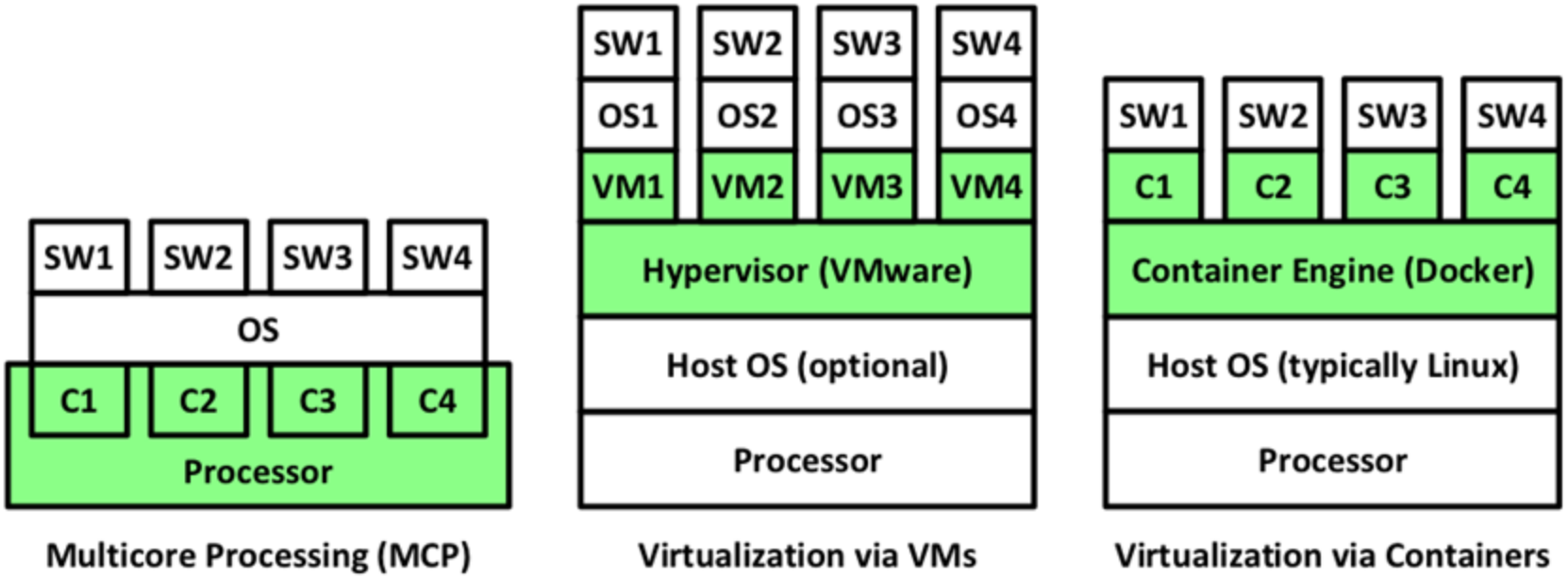
Виртуализация – это набор программных решений, который позволяет приложениям работать с виртуальным оборудованием (виртуализация с помощью виртуальных машин и гипервизора) или в виртуальных операционных системах (виртуализация с помощью контейнеров).

Виртуальная машина – программная эмуляция аппаратного обеспечения, которая предоставляет виртуальную операционную среду для гостевых операционных систем.

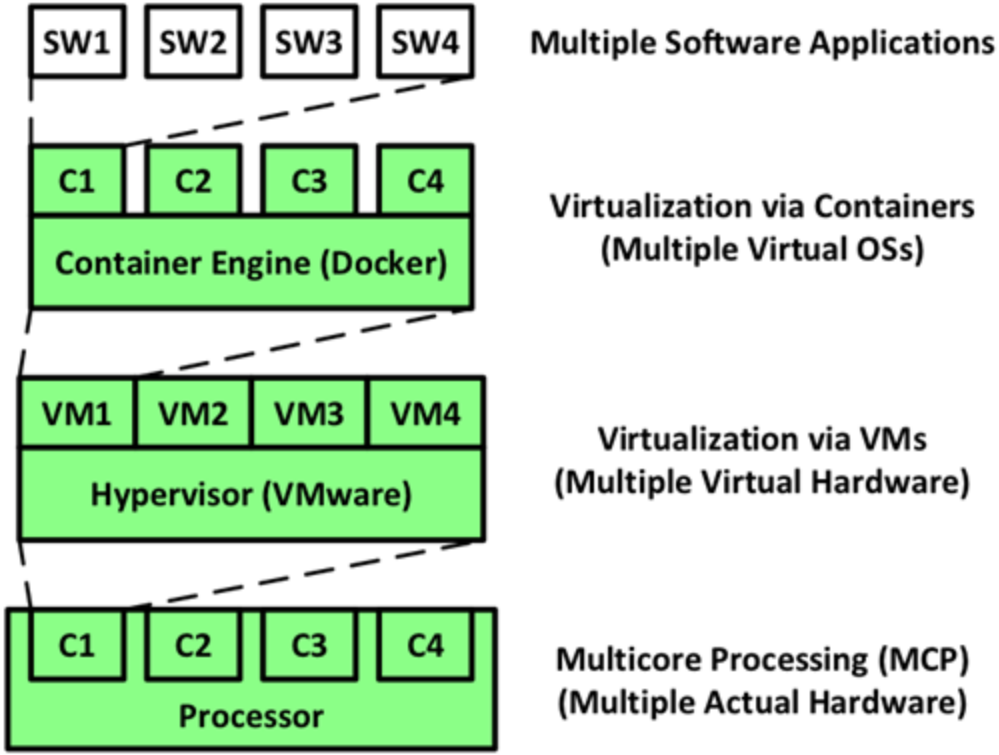
Гипервизор или монитор виртуальных машин – программное обеспечение, которое запускает и контролирует гостевые операционные системы на виртуальных машинах.

Контейнер – виртуальная среда, которая работает на одной ОС без эмуляции аппаратного обеспечения.

Технологии виртуализации



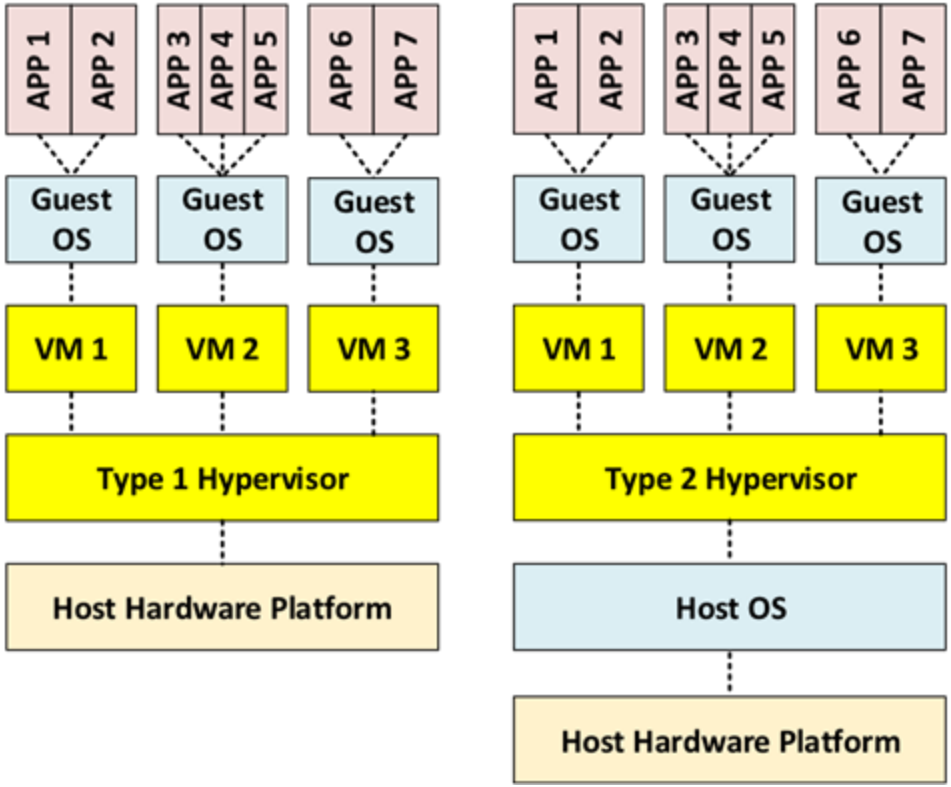
Технологии виртуализации: уровни

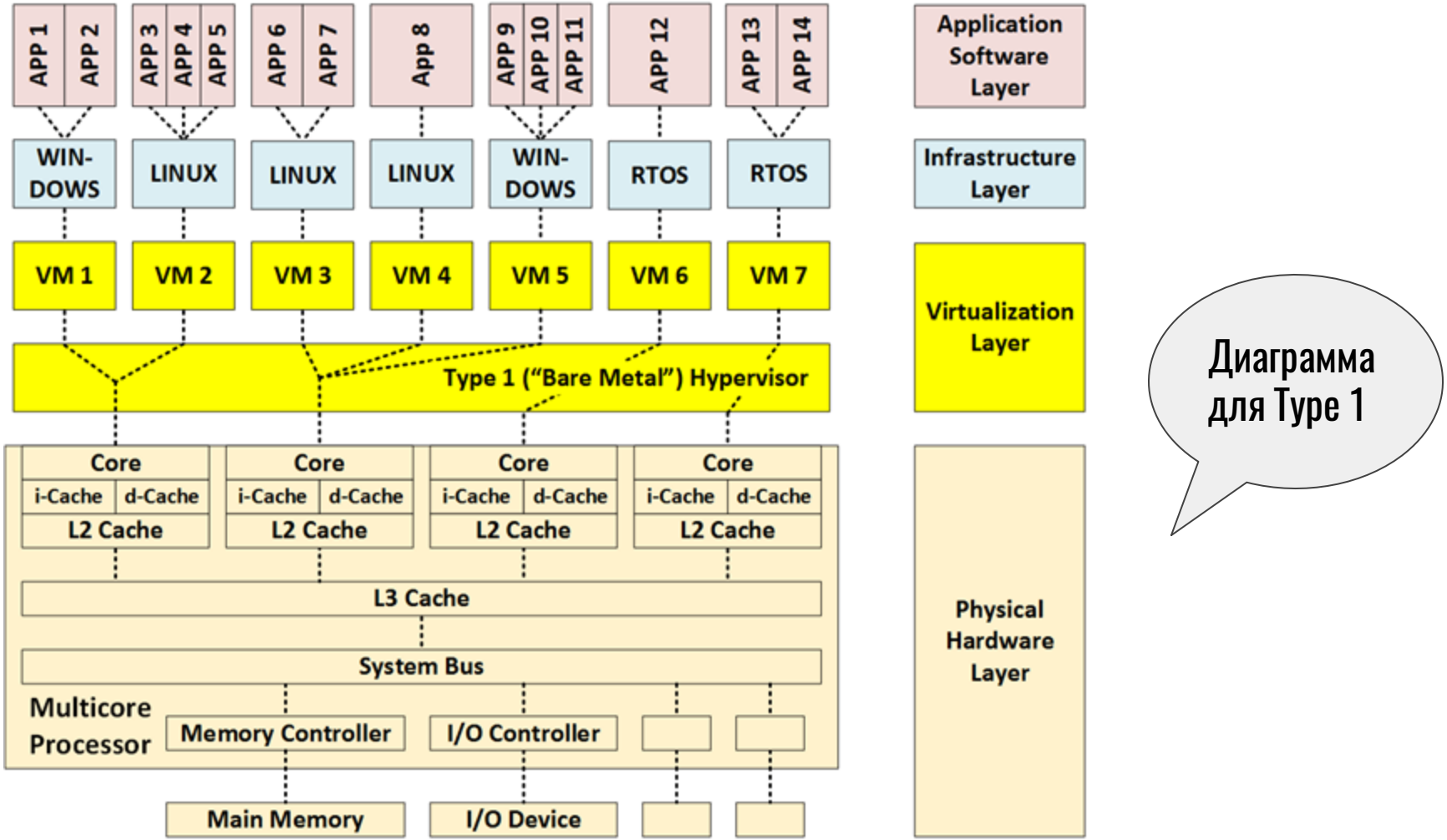


Технологии виртуализации: проблемы

- сложность архитектуры – усложняется анализ и тестирование
- добавляются уровни с разделяемыми ресурсами(кэш, контроллеры памяти и I/O, шины) – единая точка отказа (SPOF) или эффект наложения при ошибках квантования времени
- накладные расходы на виртуализацию
- системы становятся менее предсказуемы
- современные стандарты безопасности могут быть несовместимы с этими технологиями

Типы гипервизоров




APP 1
APP 2
APP 3
APP 4
APP 5
APP 6
APP 7
App 8
APP 9
APP 10
APP 11
APP 12
APP 13
APP 14
Application Software Layer
WIN-DOWS
LINUX
LINUX
LINUX
WIN-DOWS
RTOS
RTOS
Infrastructure Layer
VM 1
VM 2
VM 3
VM 4
VM 5
VM 6
VM 7
Virtualization Layer
Type 1 ("Bare Metal") Hypervisor
Core
i-Cache
d-Cache
L2 Cache
L3 Cache
System Bus
Multicore Processor
Memory Controller
I/O Controller
Main Memory
I/O Device
Physical Hardware Layer
Диаграмма для Type 1

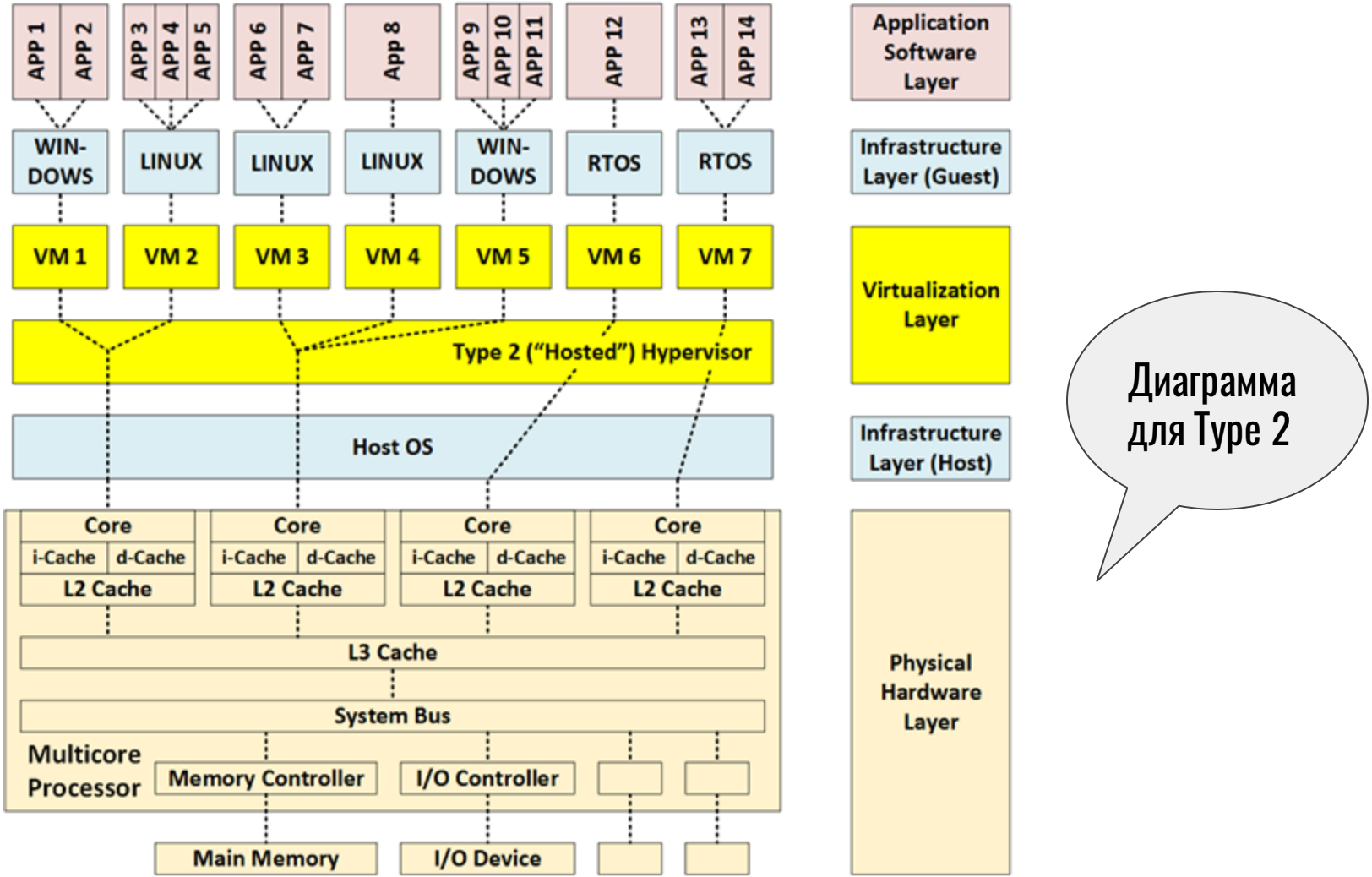


Диаграмма для Type 2

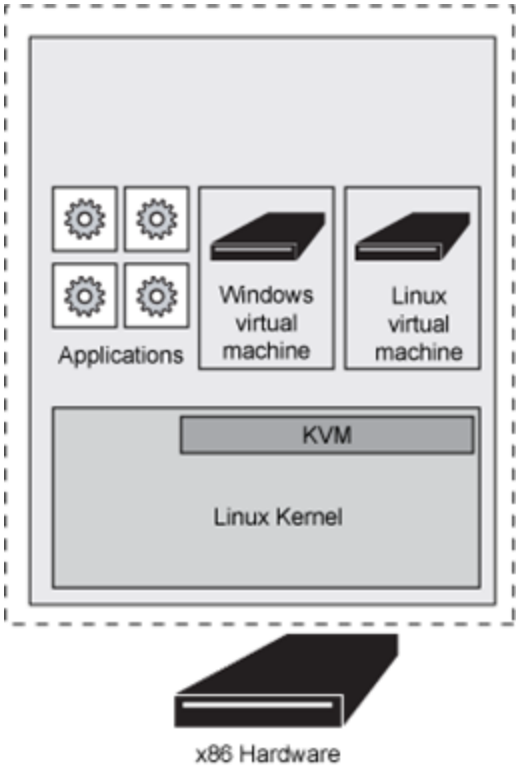
Обзор гипервизоров

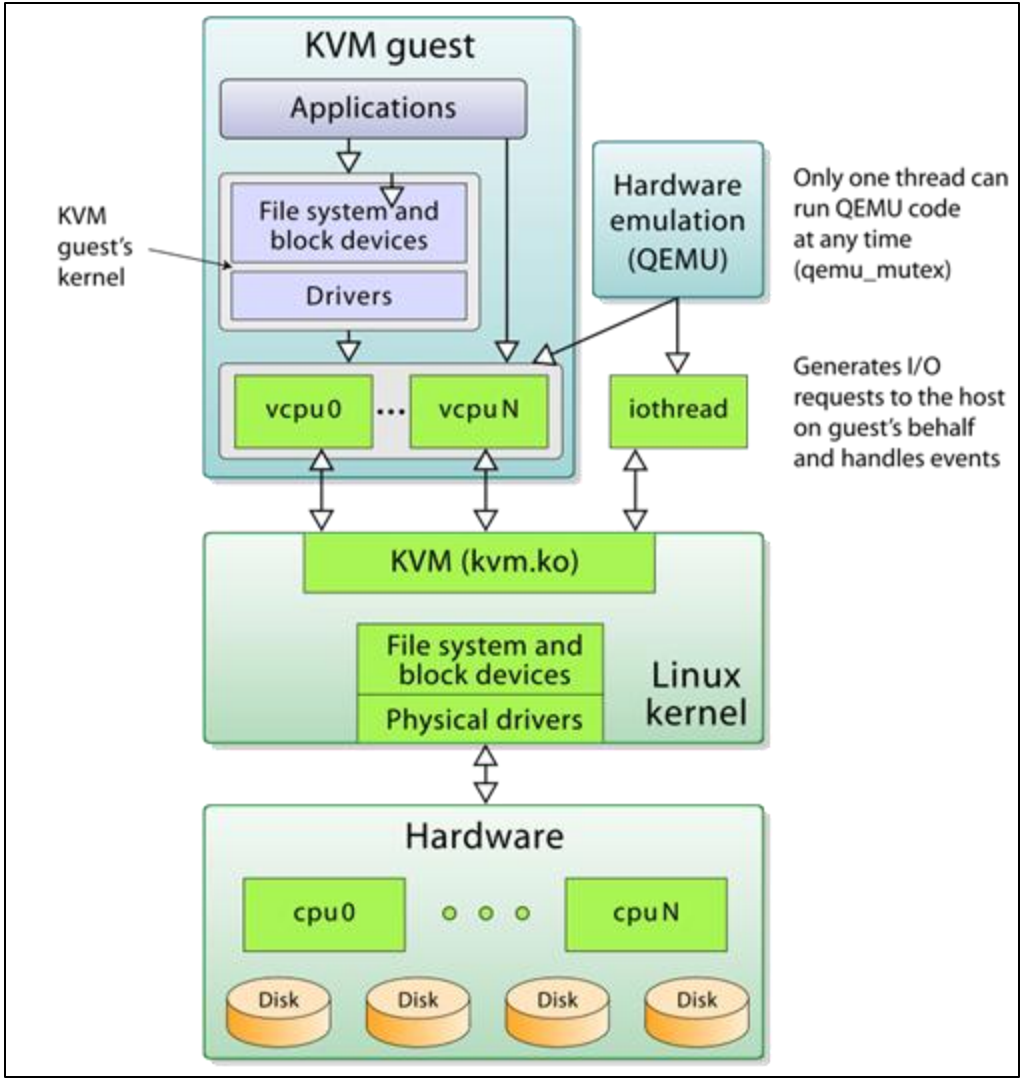
Kernel-based Virtual Machine

KVM

KVM - это программное решение для реализации виртуализации в среде Linux на оборудовании с архитектурой x86, поддерживающей аппаратную виртуализацию (Intel VT или AMD-V).

Программное обеспечение KVM состоит из загружаемого модуля ядра (называемого kvm.ko), предоставляющего базовый сервис виртуализации, процессорно-специфического загружаемого модуля kvm-amd.ko либо kvm-intel.ko, и компонентов пользовательского режима.



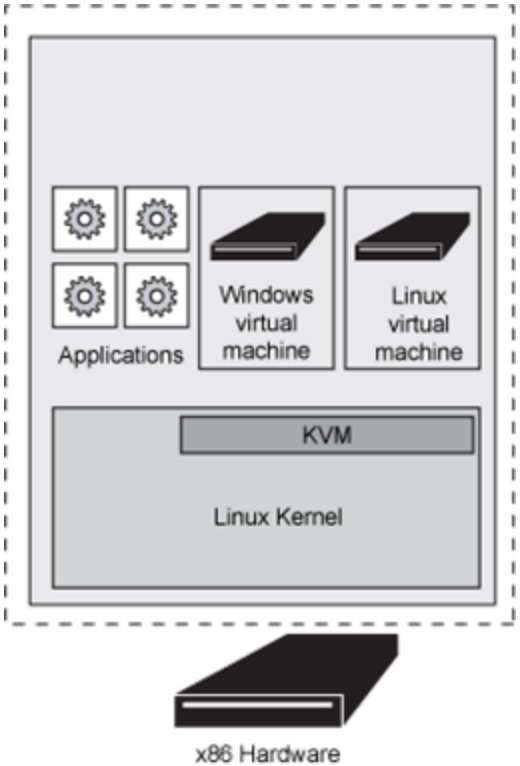


Виртуальная среда KVM

KVM

KVM - это модуль виртуализации в ядре Linux, который позволяет ядру функционировать как гипервизор.

В архитектуре KVM виртуальная машина - это процесс, который управляется стандартным планировщиком Linux. Фактически, каждый виртуальный CPU - это обыкновенный процесс, что позволяет использовать все возможности ядра.

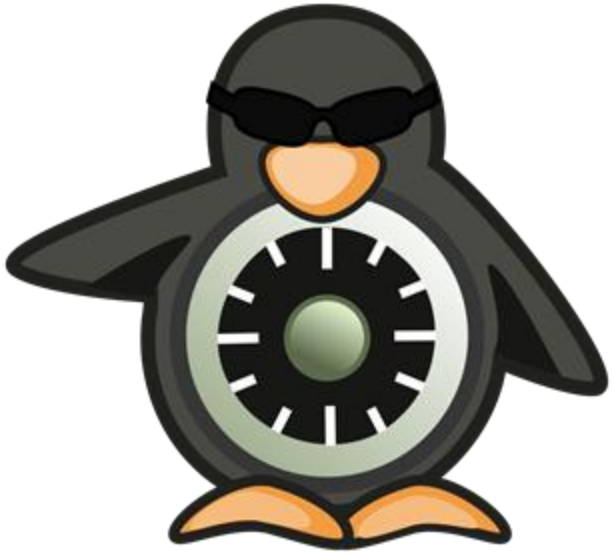


KVM: features

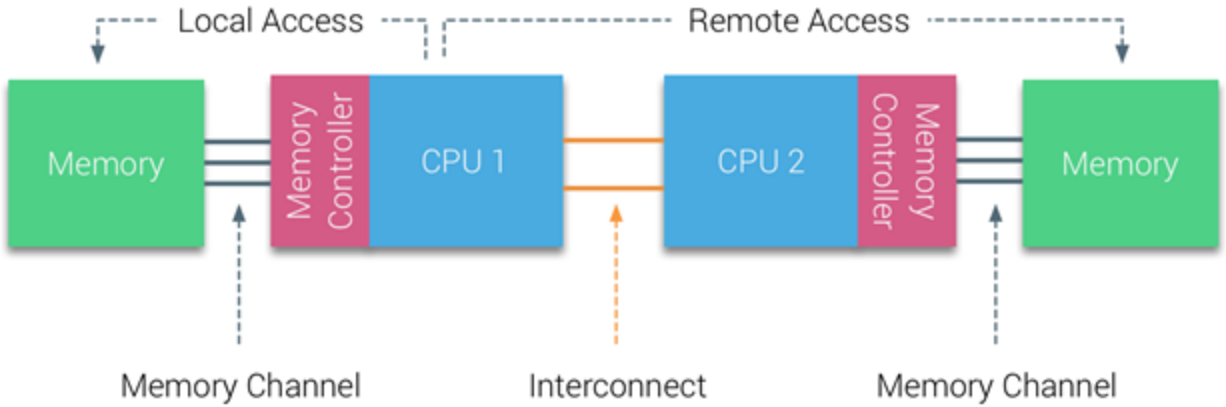
Security

KVM использует SELinux для мандатного управления доступом (MAC).

sVirt - проект интеграции MAC в KVM, основан на SELinux. Позволяет администратору определить политику изоляции виртуальных машин.



KVM: features



Memory management

KVM наследует мощные функции управления памятью от Linux. Память виртуальной машины хранится так же, как и память для любого другого процесса Linux. Поддержка NUMA позволяет виртуальным машинам эффективно обращаться к большим объемам памяти. KVM поддерживает Intel EPT и AMD RVI для снижения загрузки CPU и повышения пропускной способности. Совместное использование страниц памяти поддерживается с помощью функции ядра, называемой Kernel Same-page Merging (KSM).

KVM: features

Storage

KVM может использовать любое хранилище, поддерживаемое Linux, для хранения образов виртуальных машин. KVM также поддерживает образы виртуальных машин в разделяемых файловых системах, таких как GFS2, что позволяет совместно использовать образы виртуальных машин между несколькими хостами или совместно использовать их с помощью логических томов.

KVM: features

Live migration

KVM поддерживает живую миграцию, которая обеспечивает возможность перемещения работающей виртуальной машины между физическими узлами без прерывания обслуживания. Живая миграция прозрачна для пользователя, виртуальная машина остается включенной, сетевые подключения остаются активными, а пользовательские приложения продолжают работать, пока виртуальная машина перемещается на новый физический узел.

В дополнение к живой миграции KVM поддерживает сохранение текущего состояния виртуальной машины на диск, чтобы она могла быть сохранена и возобновлена позже.

KVM: features

Device drivers

KVM поддерживает гибридную виртуализацию, когда в гостевой операционной системе устанавливаются паравиртуализированные драйверы, позволяющие виртуальным машинам использовать оптимизированный интерфейс ввода-вывода, а не эмулируемые устройства для обеспечения высокопроизводительного ввода-вывода для сетевых и блочных устройств.

Гипервизор KVM использует стандарт VirtIO, разработанный IBM и Red Hat совместно с сообществом Linux для паравиртуализированных драйверов; это независимый от гипервизора интерфейс для построения драйверов устройств, позволяющий использовать один и тот же набор драйверов устройств для нескольких гипервизоров, что обеспечивает лучшую совместимость гостя.

KVM: pros

- KVM это легкий модуль, который поставляется с основным ядром Linux, предлагает простую реализацию и постоянную поддержку;
- Гостевые операционные системы взаимодействуют с гипервизором, встроенным в ядро Linux, они могут обращаться к оборудованию напрямую во всех случаях без необходимости модификации виртуализированной операционной системы. Это делает KVM более быстрым решением для виртуальных машин;
- Патчи к KVM совместимы с ядром Linux. KVM реализован в самом ядре Linux; следовательно, это облегчает управление процессами виртуализации.

KVM: cons

- предназначен для опытных пользователей;
- ограниченное кол-во поддерживаемых процессоров;
- KVM необходимо улучшить поддержку виртуальных сетей, поддержку виртуальных хранилищ, повышенную безопасность, высокую доступность, отказоустойчивость, управление питанием, поддержку HPC, масштабируемость виртуальных ЦП, совместимость с разными поставщиками, переносимость виртуальных машин и создание облачных сервисов

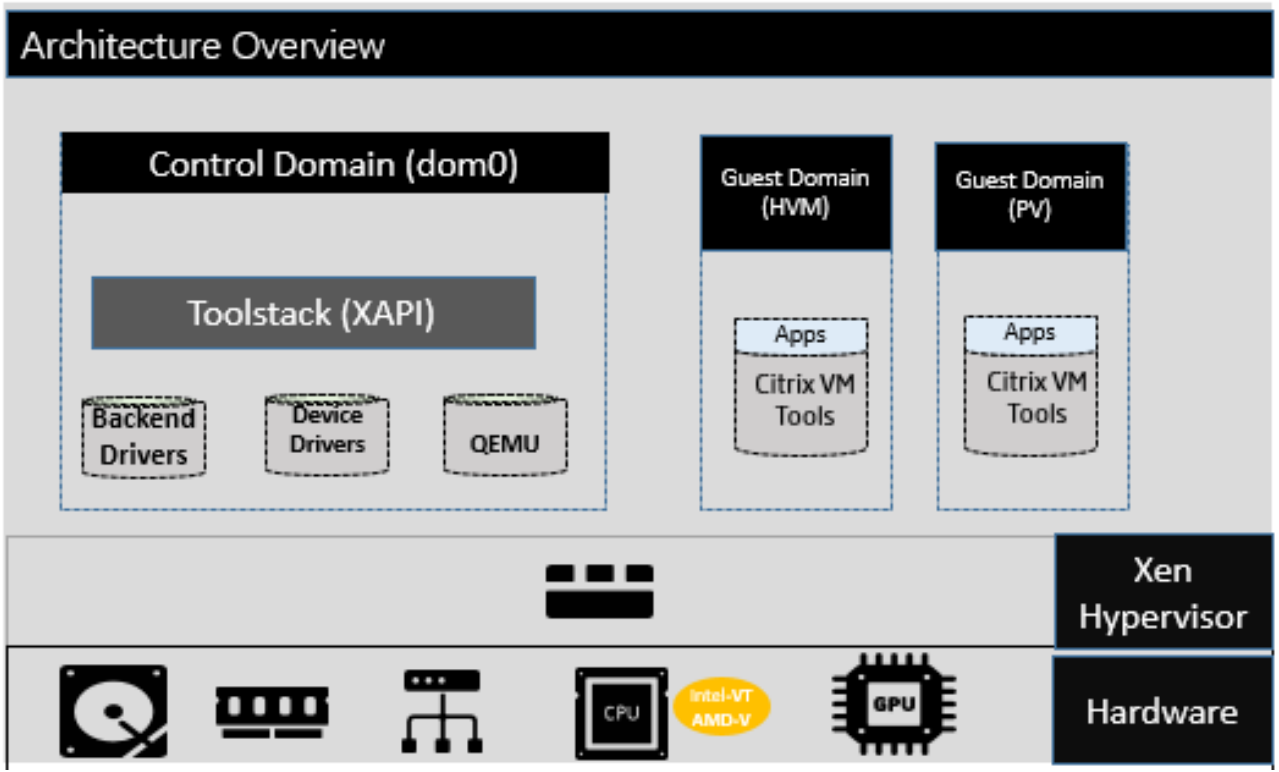
KVM: summary

Name	Creator	Host CPU	Guest CPU	Host OS	Guest OS	SMP
KVM	Red Hat	x86. x86-64, IA-64, with x86 virtualization. s390, PowerPC, ARM	Same as host	Linux. FreeBSD, illumos	FreeBSD, Linux. Solaris. Windows	Yes

Run arbitrary OS	Supported Guest OS drivers	Method of operation	License	Typical use	Speed relative to host OS
Yes	Yes	Hardware virtualization. Paravirtualization	GPL2	Virtualized server isolation. server/desktop consolidation, software development. cloud computing, other purposes	Up to near native

Xen

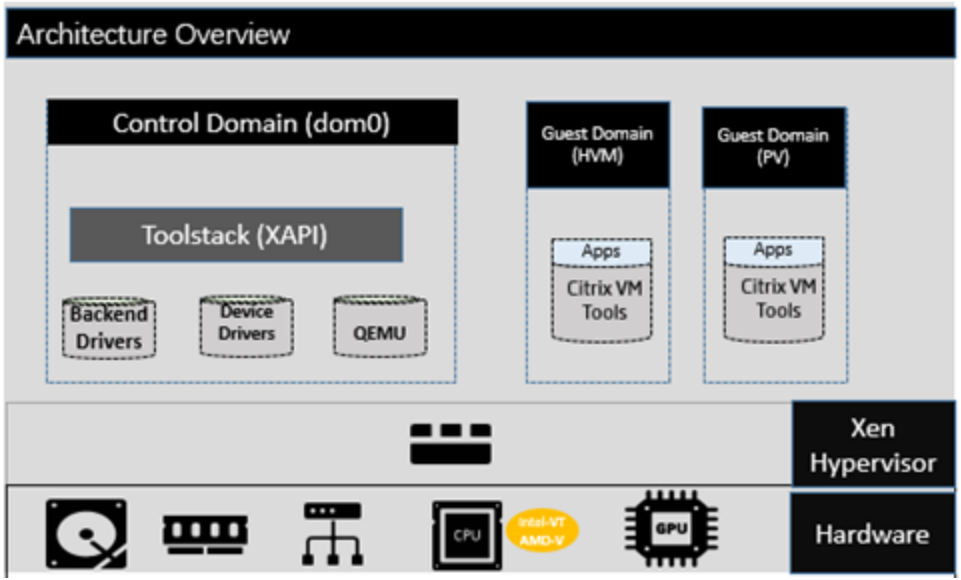
Xen


Architecture Overview
Control Domain (dom0)
Toolstack (XAPI)
Backend Drivers
Device Drivers
QEMU
Guest Domain (HVM)
Apps
Citrix VM Tools
Guest Domain (PV)
Apps
Citrix VM Tools
Xen Hypervisor
CPU
Intel-VT AMD-V
GPU
Hardware

Xen

Гипервизор Xen Project - это гипервизор типа 1 или «bare-metal».

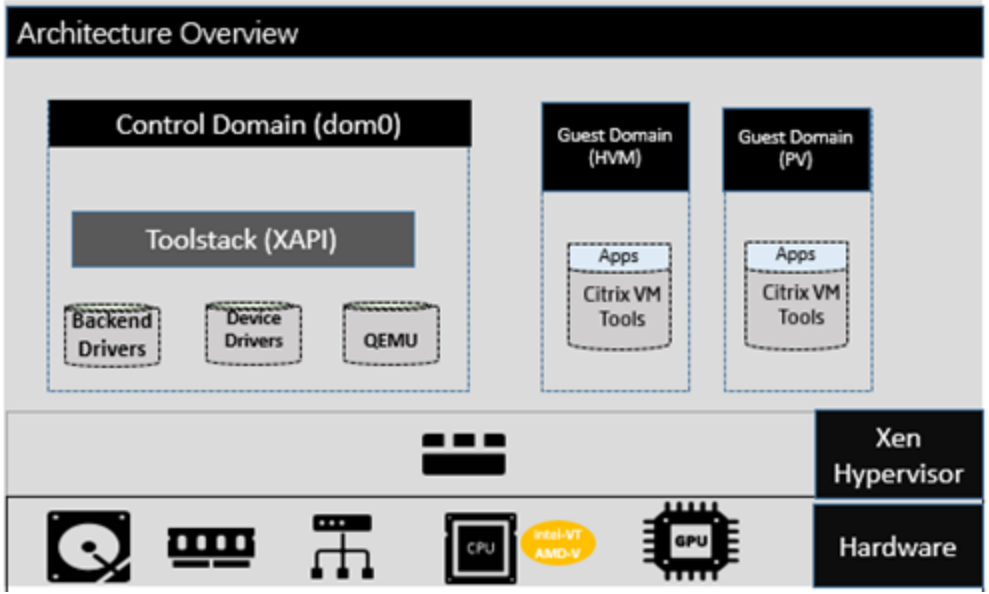
Xen



Концепция доменов

Доменом называется запущенная копия виртуальной машины. Если виртуальная машина перезагружается, то её домен завершается (в момент перезагрузки) и появляется новый домен. Более того, даже при миграции содержимое копируется из одного домена в другой домен. Таким образом, за время своей жизни практически все виртуальные машины оказываются по очереди в разных доменах. Xen оперирует только понятием домена, а понятие «виртуальной машины» появляется на уровне администрирования.

Xen



Control domain

Домены бывают нескольких типов. Самые известные **dom0** и **domU**. **dom0** — первый запущенный Xen домен, обычно он автоматически создается и загружается сразу после загрузки и инициализации гипервизора. Этот домен имеет особые права на управление гипервизором и по умолчанию всё аппаратное обеспечение компьютера доступно из **dom0**. Фактически, **dom0** — это место жизни ПО, управляющего Xen . **dom0** всегда один.

Xen

domU — рядовой домен (сокращение от User domain), содержащий в себе домен выполняющихся виртуальных машин. Обычно не имеет доступа к реальному оборудованию и является «полезной нагрузкой» системы виртуализации. В отличие от **dom0**, **domU** может быть множество (обычно несколько десятков).

domain builder (конструктор доменов) — программа, которая создает **domU** (загружает в него нужный код и сообщает гипервизору о необходимости запуска). Помимо конструирования домена, обычно занимается подключением и конфигурированием виртуальных устройств, доступных для виртуальной машины. Она же отвечает за процесс миграции виртуальной машины с хоста на хост.

Xen

Features

- Paravirtualization Mode (PV) + Hardware Virtualization Mode (HVM) = Full virtualization
- Transfer VM memory images (VM migration)
- Mirror disks (storage migration)
- Minimizing Hypervisor Functions

Xen: pros

- Сочетание паравиртуализации и аппаратной виртуализации с ОС позволяет разработать более простой гипервизор, который обеспечивает хорошую производительность.
- Xen обеспечивает сложную балансировку рабочей нагрузки; он предлагает два режима оптимизации: один для производительности и другой для плотности.
- Xen использует уникальную функцию интеграции хранилища, которая называется Citrix Storage Link.
- Включает в себя поддержку многоядерных процессоров, динамическую миграцию, инструменты преобразования физических серверов в виртуальные машины (P2V) и преобразования виртуальных машин в виртуальные (V2V), централизованное управление серверами, мониторинг производительности в реальном времени.

Xen: cons

- Xen опирается на Linux в dom0;
- Xen полагается на сторонние решения для драйверов устройств, хранилища, резервного копирования и восстановления, а также отказоустойчивости;
- Неравномерное использование ресурсов между доменами;
- Проблемы с интеграцией.

Xen: summary

Name	Creator	Host CPU	Guest CPU	Host OS	Guest OS	SMP
Xen	Citrix Systems	x86. x86-64, ARM	Same as host	GNU/Linux, Unix-like	GNU/Linux, FreeBSD, MiniOS. NetBSD, Solaris. Windows	Yes. v4.0.0: up to 128 VCPUs per VM

Run arbitrary OS	Supported Guest OS drivers	Method of operation	License	Typical use	Speed relative to host OS
Yes	Yes	Paravirtualization and porting or hardware virtualization	GPL2	Virtualized server isolation, server/desktop consolidation, software development, cloud computing, other purposes. Xen powers most public cloud services and many hosting services, such as Amazon Web Services, Rackspace Hosting and Linode.	Up to native

VMware ESXi

ESXi

ESXi является гипервизором первого типа «bare-metal».

Предъявляет определенный набор требований к аппаратному обеспечению — в частности, является обязательным наличие поддержки виртуализации со стороны процессора и материнской платы.

ESXi: vmkernel

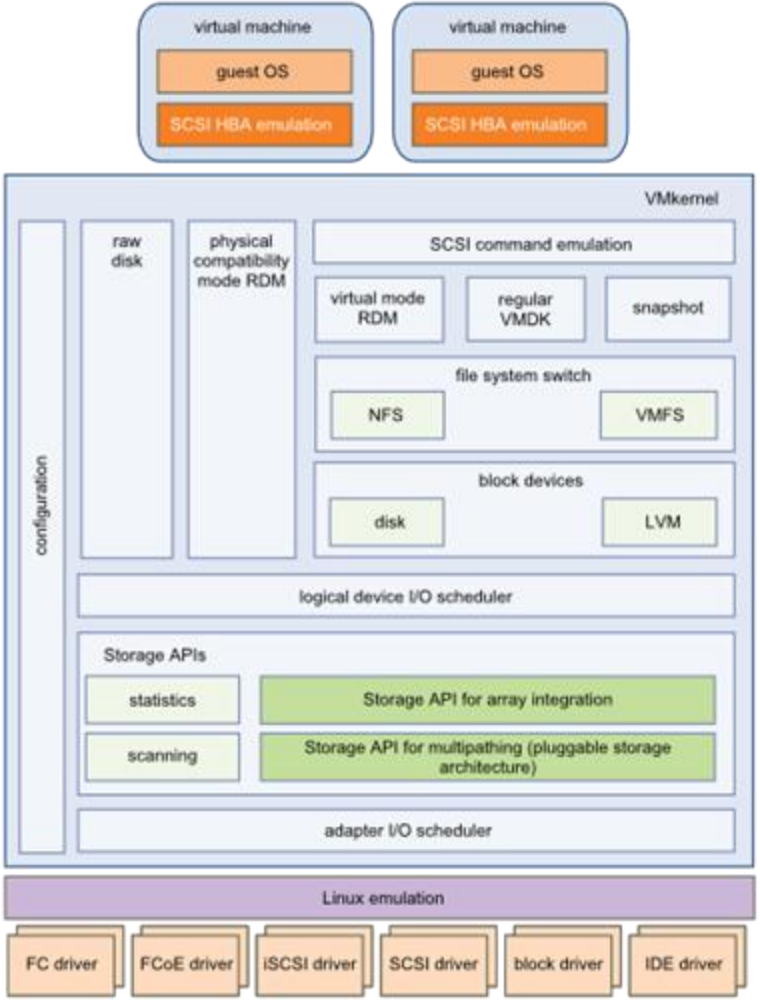
Уникальность ESXi в том, что ядро гипервизора было написано с нуля.

Предыдущие версии ESX гипервизора были построены на ядре Linux.

Но VMware прекратила разработку ESX в версии 4.1 и теперь развивает новую ветку ESXi, которая основана на собственном микроядре vmkernel.

ESXi: vmkernel

vmkernel - это высокопроизводительная операционная система, которая работает непосредственно на хосте ESXi. VMkernel управляет большинством физических ресурсов на оборудовании, включая память, физические процессоры, системы хранения и сетевые контроллеры.



ESXi

Hardware Monitoring: CIM

Common Information Model (CIM) предоставляют функции управления, такие как создание отчетов для мониторинга работоспособности hardware или обновление прошивки драйвера.

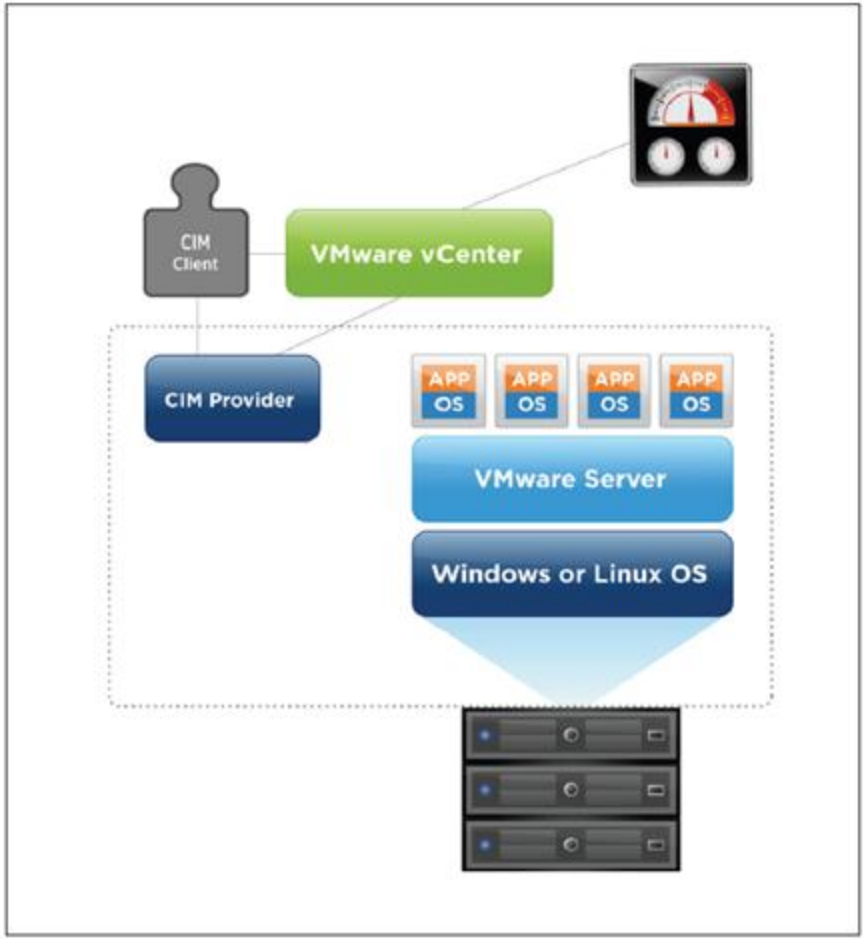
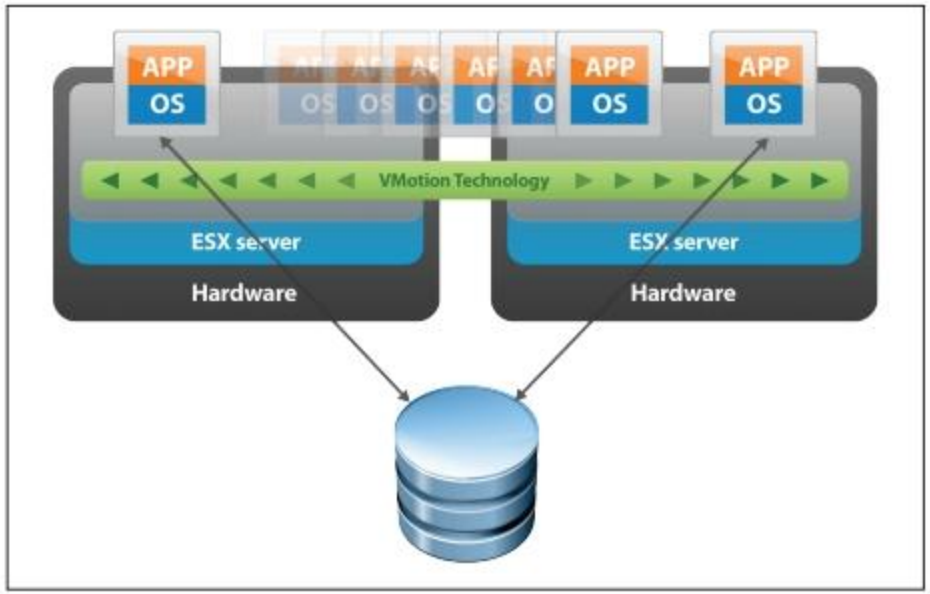


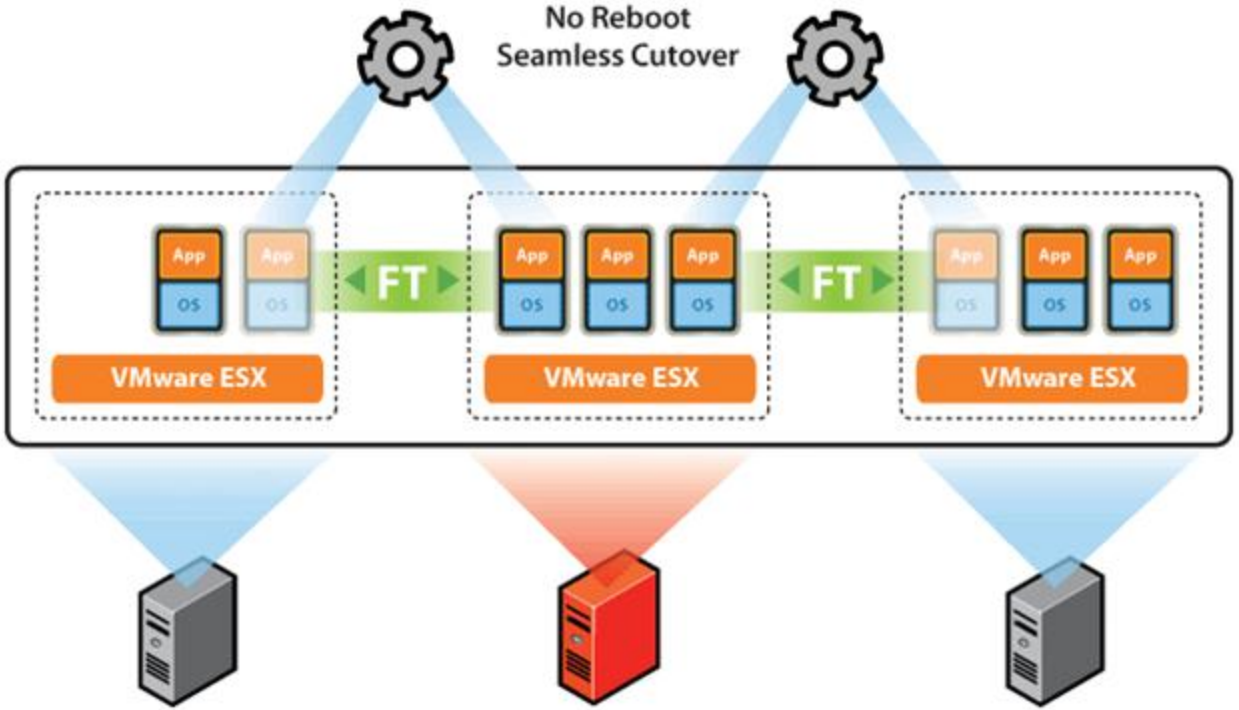
Figure 1. CIM Solution Architecture

ESXi: migration

vMotion — очень удобная функция, которая дает возможность переносить работающую виртуальную машину с сервера на сервер без простоя.



ESXi: fault tolerance



ESXi: pros

- Компактность;
- Масштабируемость инфраструктуры; каждая ESXi поддерживает до 256 включенных виртуальных машин;
- Система хранения добавляет и расширяет виртуальные диски без перерыва на работающей виртуальной машине.
- Отказоустойчивость и доступность;
- VMware обеспечивает центральную точку управления (vCenter) для управления виртуализацией.

ESXi: cons

- VMware требует больше патчей и обновлений;
- VMware vCenter требует сторонней базы данных для хранения информации и управления конфигурациями хост-системы.
- В VMware есть некоторые дыры в безопасности (например, проблемы с раздуванием памяти).

ESXi: summary

Name	Creator	Host CPU	Guest CPU	Host OS	Guest OS	SMP
ESXi	VM	x86, x86-64	x86, x86-64	No host OS	Windows. Linux. Solaris, FreeBSD	Yes. add-on, up to 64-way

Run arbitrary OS	Supported Guest OS drivers	Method of operation	License	Typical use	Speed relative to host OS
No	Yes	Virtualization	Proprietary	Server consolidation, service continuity, dev/test, cloud computing, business critical applications, Infrastructure as a Service IaaS	Up to near native

Источники

1. https://insights.sei.cmu.edu/sei_blog/multicore-processing-and-virtualization/
2. https://developer.ibm.com/articles/cl-hypervisorcompare-kvm/
3. https://developer.ibm.com/articles/cl-hypervisorcompare-xen/
4. https://developer.ibm.com/articles/cl-hypervisorcompare-vmwareesx/
5. https://wiki.xenproject.org/wiki/Category:Overview
6. https://docs.vmware.com/en
7. https://www.linux-kvm.org/page/Main_Page
8. https://en.wikipedia.org/wiki/Main_Page