Тема 10.
Виртуализация

Понятие виртуализации

- В компьютерных технологиях под термином "виртуализация" обычно понимается абстракция вычислительных ресурсов и предоставление пользователю системы, которая "инкапсулирует" (скрывает в себе) собственную реализацию.
- Проще говоря, пользователь работает с удобным для себя представлением объекта, и для него не имеет значения, как объект устроен в действительности.

Преимущества виртуализации:

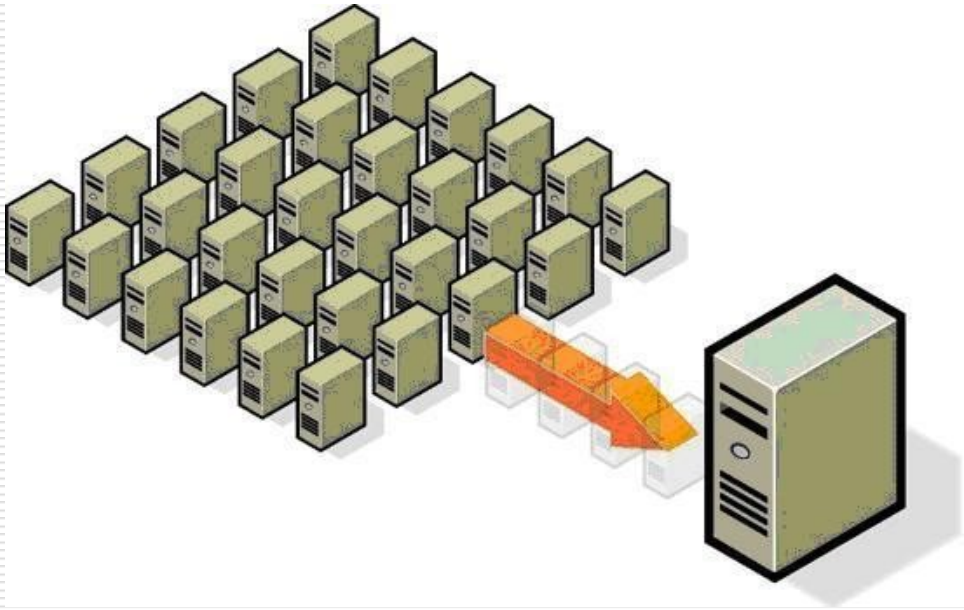
1. Эффективное использование вычислительных ресурсов
2. Сокращение расходов на инфраструктуру
3. Снижение затрат на программное обеспечение.
4. Повышение гибкости и скорости реагирования системы.
5. Несовместимые приложения могут работать на одном компьютере
6. Повышение доступности приложений и обеспечение непрерывности работы предприятия
7. Возможности легкой архивации
8. Повышение управляемости инфраструктуры

Виртуализация ЭВМ

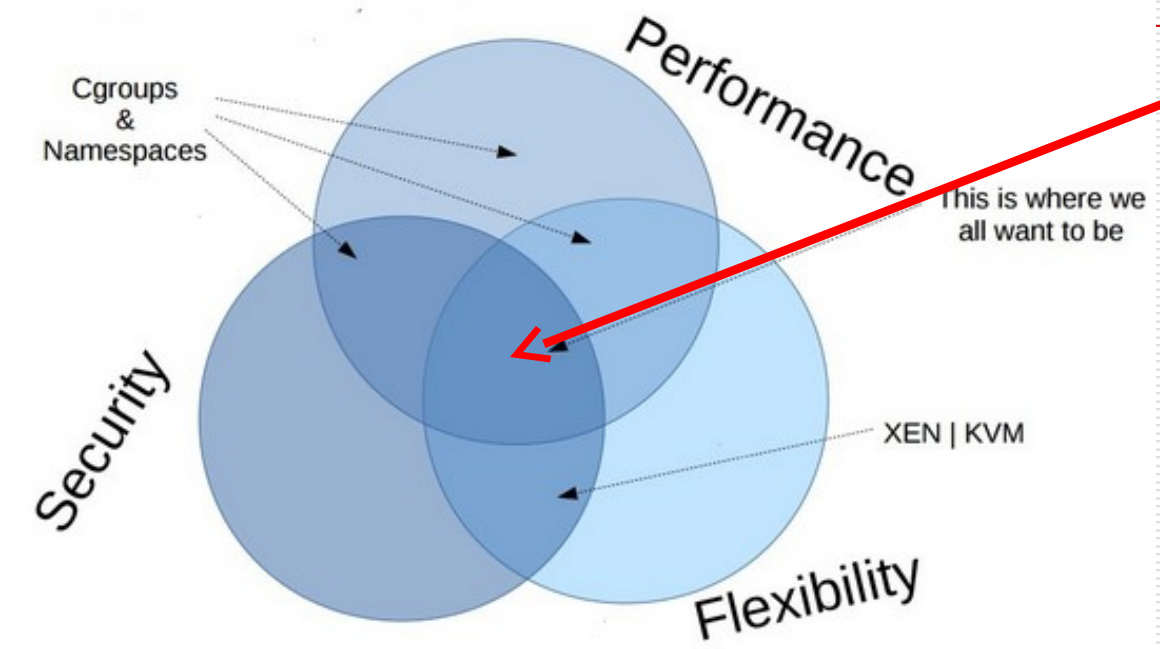
- Виртуализация серверов — размещение нескольких логических серверов в рамках одного физического.
- Цели виртуализации:
  - Предоставить каждому пользователю изолированную среду исполнения приложений.
  - Повысить гибкость использования ресурсов ЭВМ исполняемыми на ней приложениями.
  - Повысить защищенность приложений друг от друга исполняемых на одной и той же ЭВМ.
  - Повысить эффективность использования аппаратных средств ЭВМ.

Виртуализация ресурсов физического сервера

- Виртуализация ресурсов физического сервера позволяет:
  - гибко распределять их между приложениями, каждое из которых при этом "видит" только предназначенные ему ресурсы и "считает", что ему выделен отдельный сервер, т. е. в данном случае реализуется подход "один сервер — несколько приложений"



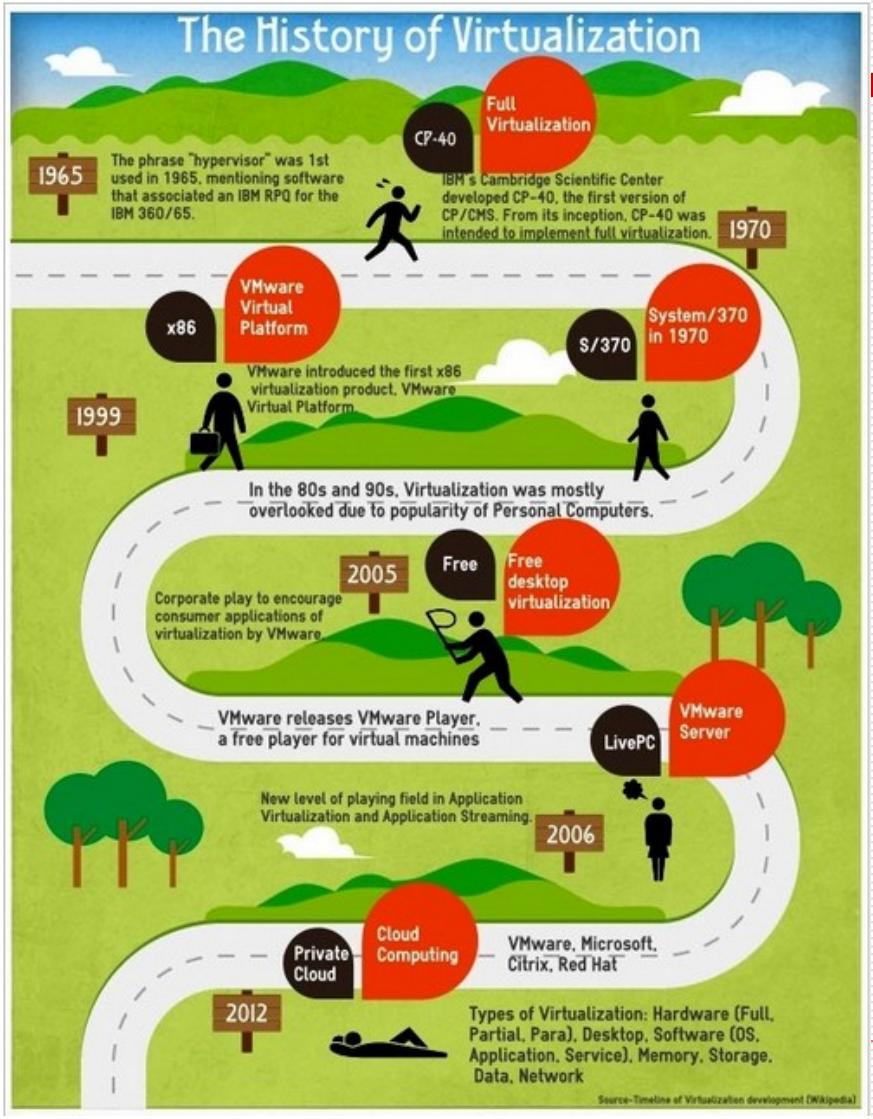
Цели виртуализации



Область эффективной виртуализации

- Первые версии гипервизоров отличались относительной медлительностью и действительно приводили к серьезному снижению производительности по сравнению с тем, как операционные системы и запущенные в них приложения работали на «реальном железе»
- Сегодня появилась модель предоставления виртуального хостинга на базе контейнеров ОС (или «Легких ВМ») – максимум эффективности при минимальной нагрузке на серверы.

История виртуализации (1)



- 1965. Выражение “Hypervisor” впервые появилось применительно к ПО обработки RPQ на ЭВМ IBM 360/65.

- Примерно 1966. В кембриджском научном центре разработан эмулятор CP-40 для S/360-40, явившийся первой попыткой реализации полной программной виртуализации физической ЭВМ.
- 1967. На основе этой разработки была создана CP[-67]/CMS – virtual machine/virtual memory time-sharing operating system для S/360-67

История виртуализации (2) Первый гипервизор VM/370

- В начале 70-х гипервизор CP-67, был переработан в виде OS VM/370 для нового семейства машин System/370, выпущенного на рынок в 1972 г.
- Линейка машин System/370 в 1990-х годах была заменена компанией IBM линейкой System/390. Виртуализация в ОС MVS 390 была сохранена.
- В 2000 году IBM выпустила машины z-серии, поддерживающие 64-разрядное виртуальное адресное пространство при сохранении обратной совместимости с System/360 и поддержке виртуализации.
- Все эти системы фирмы IBM поддерживали виртуализацию на десятилетия раньше того момента, когда она приобрела популярность на машинах семейства x86.

Машина IBM System/ 370.

Экран терминала с заставкой VM/370

История виртуализации (3)

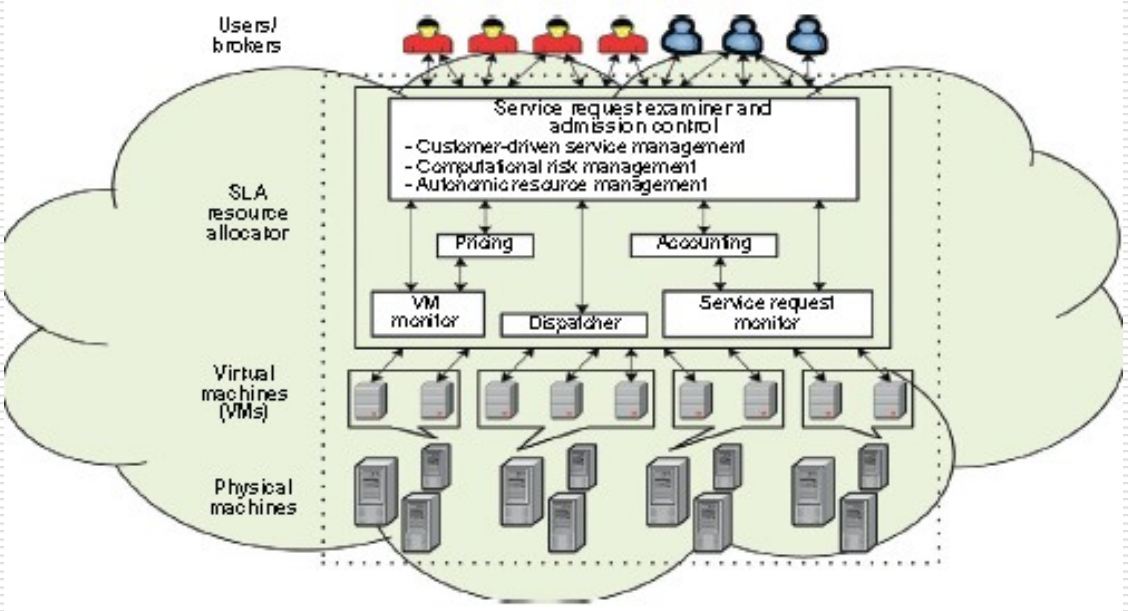
- 1980-90 г.г. В это время основные работы в области виртуализации велись в направлении адаптации этой технологии для персональных ЭВМ и прежде всего для архитектуры Intel x86.
- В 1999 г. компания VMware представила технологию виртуализации систем на базе x86 получившую название VMware Virtual Platform. Первым продуктом реализующим новую технологию было ПО VMware WorkStation (гирервизор на основе хозяйской ЭВМ).
- 2005 г. VMware выпустила первое бесплатное ПО виртуализации десктопов - VMware Player.
- 2006 г. Выпустила ПО VMware ESX Server – первый гипервизор полной виртуализации серверов для архитектуры x86 (native hypervisor – “родной” гипервизор x86)

История виртуализации (3)

- Позднее в "битву“включились такие компании как:
  - Parallels (ранее SWsoft), продукты Parallels Workstation, **Parallels Desktop для Mac** (2006), **Parallels Virtuozzo Containers** (технология виртуализации средствами ОС);
  - Oracle (Sun Microsystems), **VirtualBox** (2008) была приобретена у компании Innotek в 2007;
  - Citrix Systems (**XenSourse**), Xen разработан в кембриджеском университете (там же где и СР-40), первый публичный релиз Xen выпущен в 2003. Гипервизор Xen использует технологию паравиртуализации. В октябре 2007 Citrix купила XenSource и осуществила переименование продуктов Xen;
  - RedHat. Программное обеспечение **KVM** было создано, разрабатывается и поддерживается фирмой Qumranet, которая была куплена Red Hat за $107 млн 4 сентября 2008 года. После сделки KVM (наряду с системой управления виртуализацией oVirt) вошла в состав платформы виртуализации RHEV;
  - Корпорация Microsoft вышла на рынок средств виртуализации в 2003 г. с приобретением компании Connectix, выпустив свой первый продукт **Virtual PC** для настольных ПК. Microsoft ***Hyper-V*** (кодовое имя Viridian), — система аппаратной виртуализации для x64-систем на основе гипервизора. Бета-версия *Hyper-V* была включена в x64-версии Windows Server 2008, а законченная версия (автоматически, через Windows Update) была выпущена 26 июня 2008.

История виртуализации (4)

- 2011г. Сформулированы основные модели развертывания и признаки облачных вычислений.
- В основе облачных вычислений лежат технологии виртуализации.



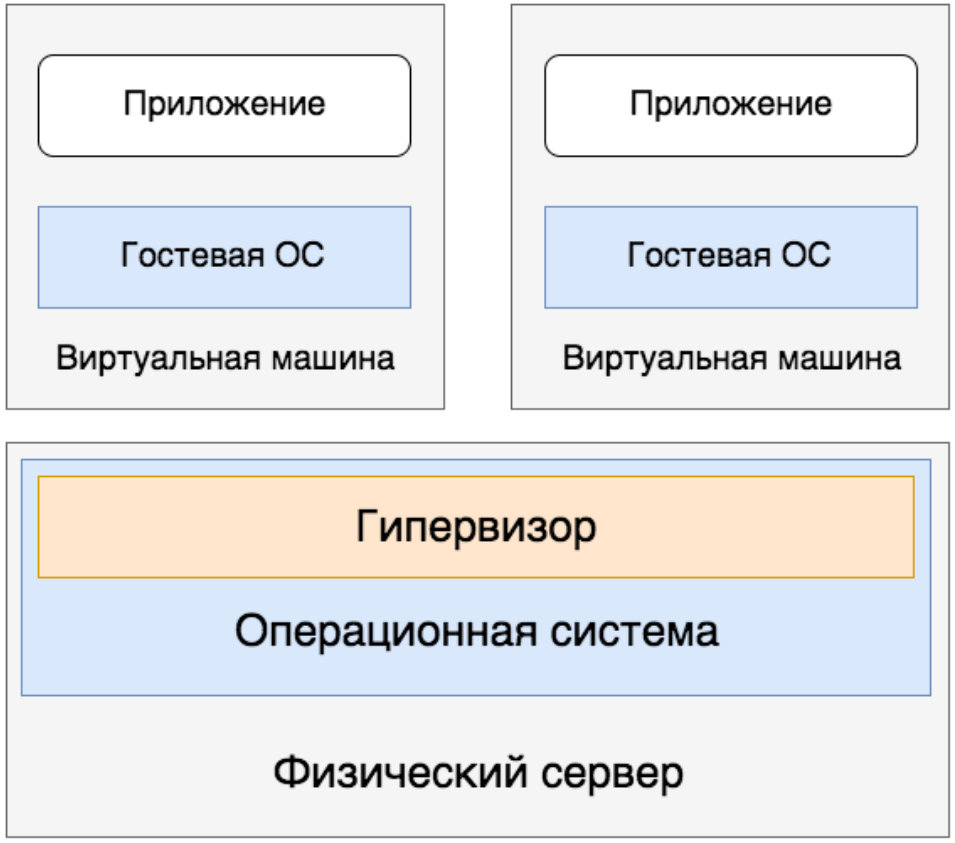
Виртуальная машина

Достоинства:

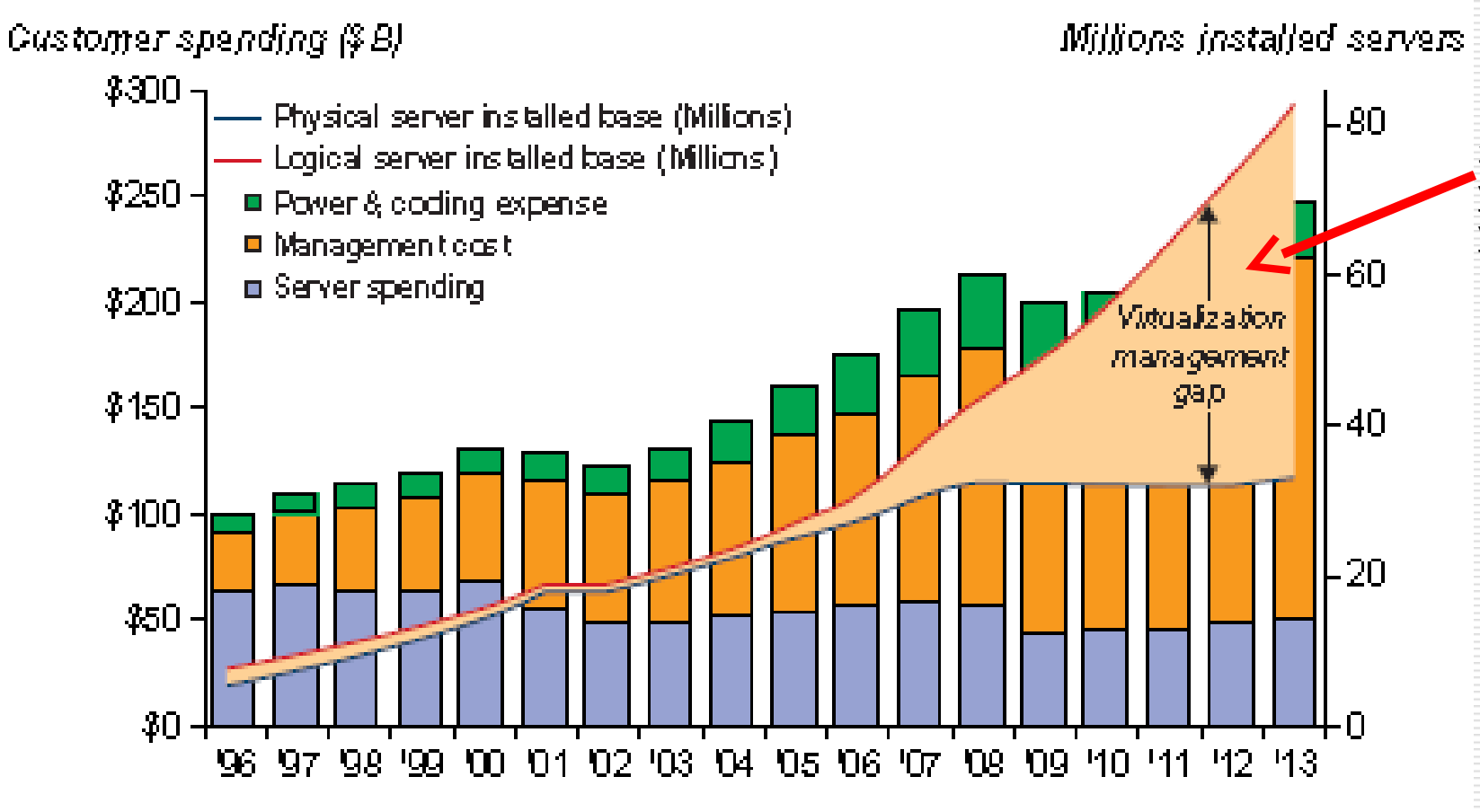
- Эффективность использования ресурсов
- Масштабируемость
- Простые резервное копирование и миграция
- Гибкость

Недостатки:

- Проблема в распределении ресурсов при высокой загрузке
- Vendor lockin
- Сложная настройка

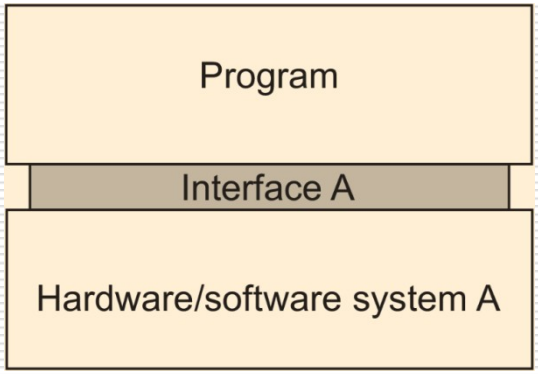


Рост стоимости и числа виртуализированных ЦОД в период 1996-2013



Рост доли затрат на управление управления виртуализа цией.

Принципы виртуализации

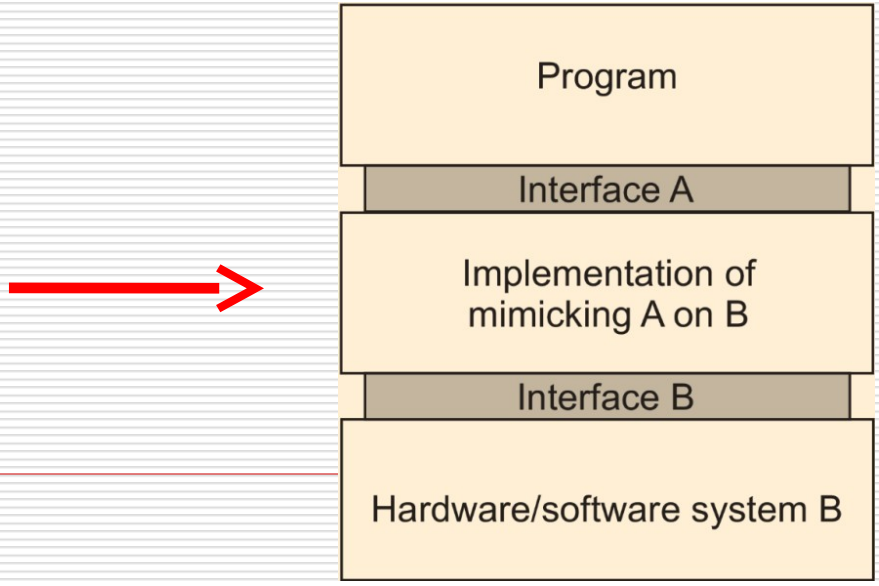


- В реальной программной системе имеется ряд интерфейсов, начиная с базового набора команд ЦПУ и кончая коллекцией API поставляемых с различным ПО современных систем с промежуточным слоем.
- Виртуализация – это по сути замена существующих интерфейсов на интерфейсы имитирующие поведение других систем.

Виртуализация и рапределенные системы

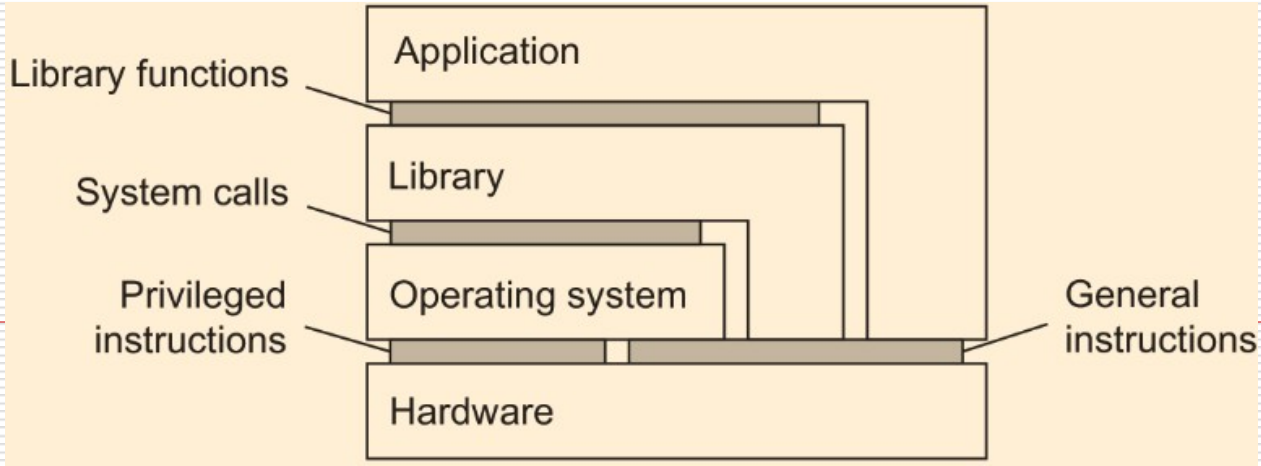
- Рассмотрим случай, когда необдходимо, обеспечить исполнение программы разработанной для аппаратной платформы A, на платформе.
- Для подобной виртуализации (исполнения) программы, созданной для платформы A, для новой платформе B, необходимо обеспечить преобразование интерфейса A (между программой и аппаратной платформой A) в интерфейс для платформы B.

- В распределенных системах, такое преобразование реализуется средствами ПО промежуточного слоя. В том числе и средствами ОС.

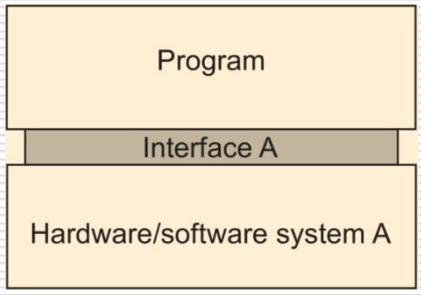


Типы виртуализации.

- Имеется несколько способов реализации виртуализации в зависимости от типа интерфейса компьютерной системы, используемого для этого:
  - Виртуализация на уровне набора команд (ISA). Используется интерфейс между аппаратными средствами и ПО, который представляет собой набор машинных команд;
  - Виртуализация на уровне системных вызовов (эмуляция системных вызовов библиотек функций ОС);
  - Виртуализация на уровне библиотечных вызовов (на уровне API) приложений.

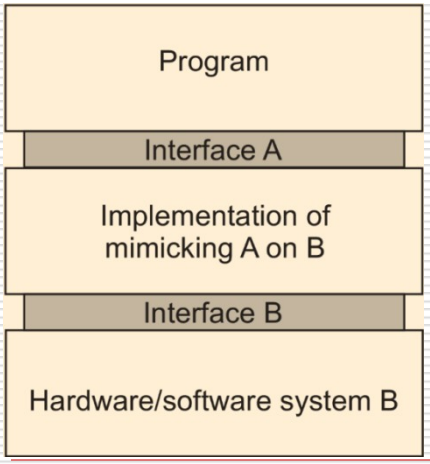


Принципы виртуализации



В компьютерной системе имеется ряд интерфейсов, начиная с базового набора команд ЦПУ и кончая коллекцией API поставляемых с различным ПО современных систем с промежуточным слоем.

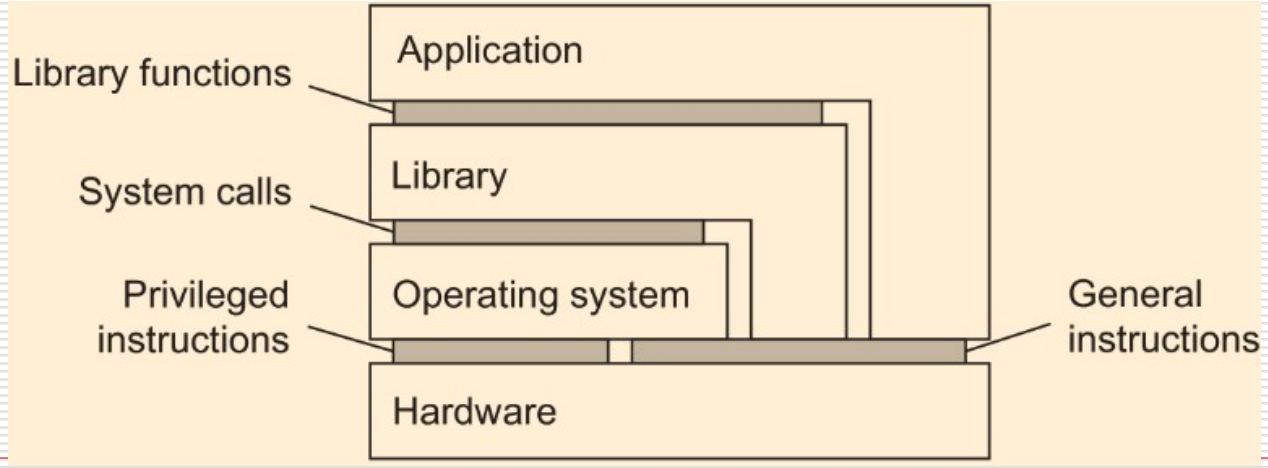
- **Виртуализация – это по сути замена существующих интерфейсов на интерфейсы имитирующие поведение других систем.**



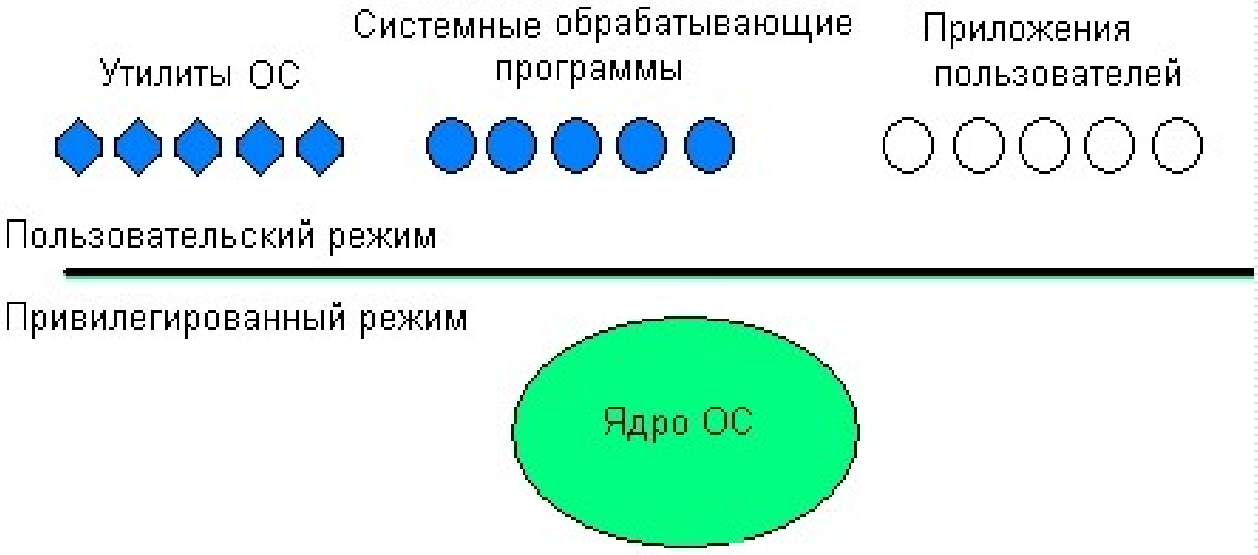
Например, для исполнения под ОС Unix приложений Windows, необходимо использовать эмулятор обеспечивающий преобразование системных вызовов Windows в системные вызовы UNIX.

Виды интерфейсов в компьютерной системе

- В компьютерной системе в общем случае выделяют 4 слоя, связанные тремя типам интерфейсов:
  - Интерфейс между аппаратными средствами и операционной системой (ISA – Instruction Set Archecture).
  - Интерфейс системных вызовов ОС (.
  - Интерфейс вызова библиотечных функций (Library function) (API).



Архитектура операционной системы с ядром в привилегированном режиме

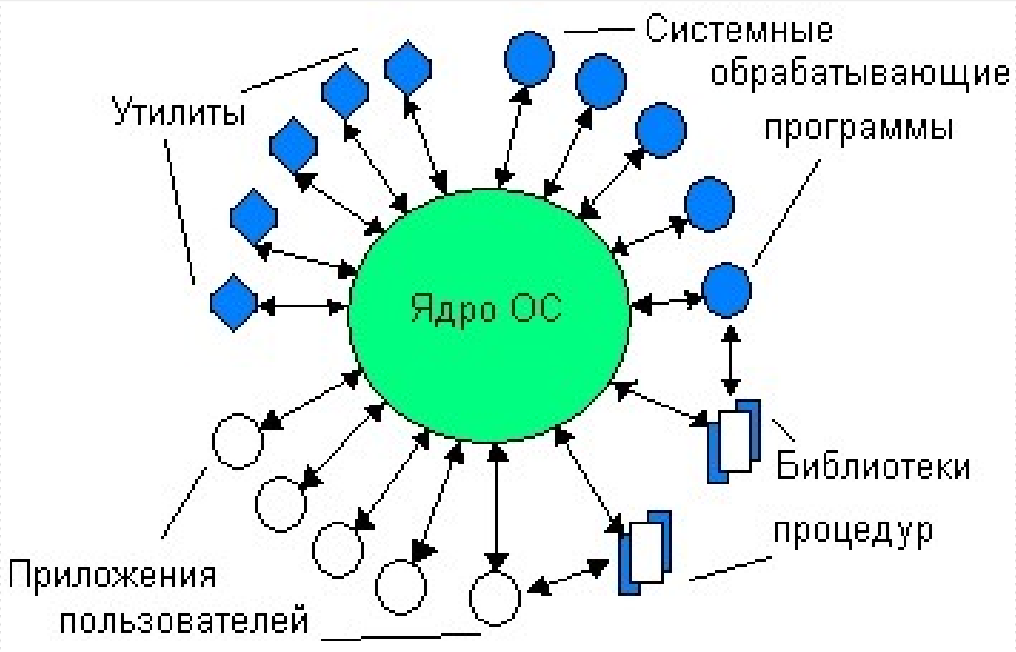


- **Обеспечить привилегии операционной системе невозможно без специальных средств аппаратной поддержки. Аппаратура компьютера должна поддерживать как минимум два режима работы — пользовательский режим (user mode) и привилегированный режим, который также называют режимом ядра (kernel mode), или режимом супервизора (supervisor mode). Подразумевается, что операционная система или некоторые ее части работают в привилегированном режиме, а приложения — в пользовательском режиме.**
- **Так как ядро выполняет все основные функции ОС, то чаще всего именно ядро становится той частью ОС, которая работает в привилегированном режиме**

Смена режимов при выполнении системного вызова к привилегированному ядру

- **Архитектура ОС, основанная на привилегированном ядре и приложениях пользовательского режима, стала, по существу, классической.**
- **Ее используют многие популярные операционные системы, в том числе многочисленные версии UNIX, VAX VMS, IBM OS/390, OS/2, и с определенными модификациями — Windows NT/2000/2003/XP.**

Способы реализации прикладных программных сред



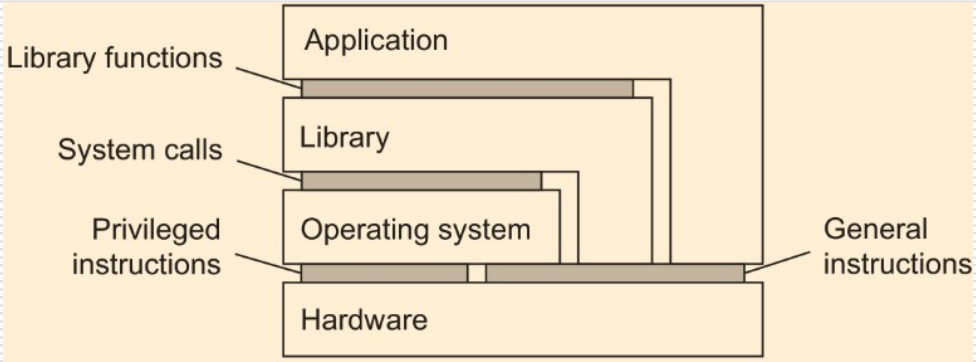
Создание полноценной прикладной среды, полностью совместимой со средой другой операционной системы, является сложной задачей, тесно связанной со структурой операционной системы.

Существуют различные варианты построения множественных прикладных сред:

- в виде обычного приложения (UNIX).
- в виде серверов пользовательского режима (Windows NT/2000/2003/XP);
- средства организации прикладных сред встроены глубоко в операционную систему (OS/2)

Типы виртуализации

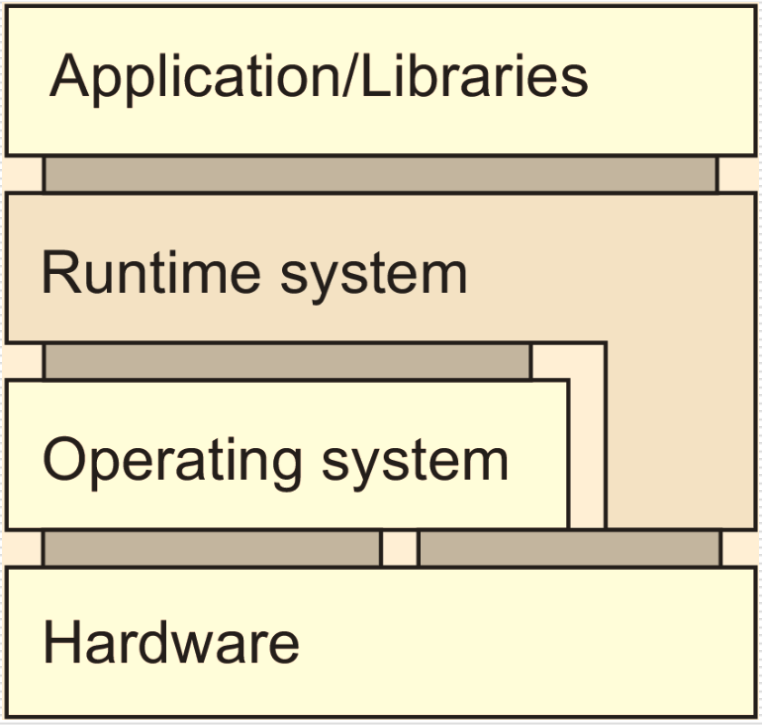
- Тип виртуализации определяются типом интерфейса на котором она выполняется:
- Виртуализация на уровне набора команд (интерфейс ISA). При этом надо учитывать, что в состав набора команд входят привилегированные и непривилегерованные команды;
- Виртуализация на уровне системных вызовов (интерфейс системных вызовов);
- Виртуализация на уровне библиотечных вызовов (на интерфейсе вызова библиотечных функций или API).



Два способа реализации виртуализации

- Суть виртуализации – имитация поведения интерфейса компьютерной системы. Существуют два способа реализации такой имитации:
  - Средствами среды исполнения приложения.
  - Средствами экранирования аппаратных средств машины с помощью ПО преобразования в набор команд той же или другой архитектуры, используемого в качестве интерфейса. Это ПО получило название **VMM – Virtual Machine Monitor** (Монитор виртуальных машин).

Виртуализация на уровне среды исполнения прикладной программы. Виртуальная машина одного процесса.



- В этом случае среда исполнения обеспечивает либо:
  - интерпретацию кода приложения построенного на основе некоторого абстрактного набора команд. Пример: Java VM, реализующая интерпретацию байт кода в машинные команды аппаратной платформы.
  - эмуляцию кода приложения, написанного для одной ОС в код приложения предназначенного для другой ОС. Пример: ПО Wine для ОС Linux, позволяющее исполнять программы для Windows.
- Оба способа исполнения работают только с непривилегированными командами процессора, поэтому они не требуют специальных условий для испонения кода.
- Такая виртуализация получила название **виртуальной машины одного процесса**.

Требования к архитектуре ЭВМ, для поддержки виртуализации

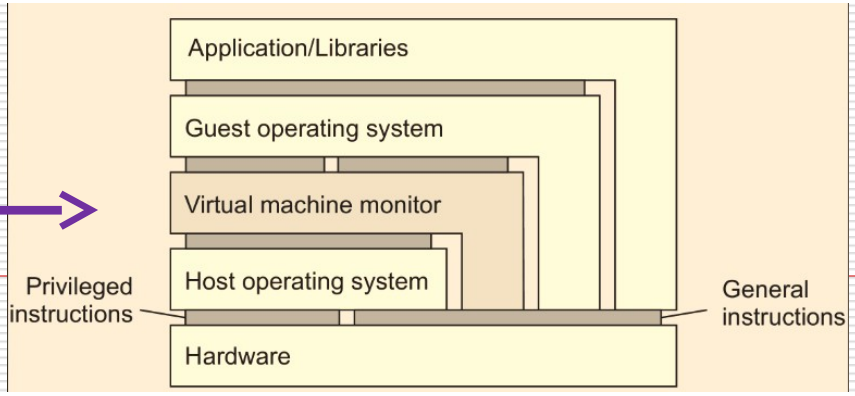
- В 1974 году двое ученых из Калифорнийского университета (Лос-Анджелес), работающих в компьютерной сфере, Геральд Попек (Gerald Popek) и Роберт Голдберг (Robert Goldberg), опубликовали основополагающую статью («Formal Requirements for Virtualizable Third Generation Architectures»), в которой дали точный перечень тех условий, которым должна отвечать компьютерная архитектура, чтобы иметь возможность эффективно поддерживать виртуализацию.
- Такими требованиями являются:
  - 1. **Безопасность** — у гипервизора должно быть полное управление виртуализированными ресурсами.
  - 2. **Эквивалентность** — поведение программы на виртуальной машине должно быть идентичным поведению этой же программы, запущенной на реальном оборудовании.
  - 3. **Эффективность** — основная часть кода в виртуальной машине должна выполняться без вмешательства гипервизора.

Проблемы виртуализации в архитектуре Intel x86

- В наборе команд Intel x86 (включая и x64) имеются команды способные изменить состояние процессора исполняемые в пользовательском режиме. Различают команды:
  - поведение которых зависит от режима исполнения (behavior sensitive);
  - которые влияют на управление (control senitive)
- Например команда **POPF**, которая устанавливает флаг разрешения прерывания, только когда работает в режиме ядра, но относится к общим командам.
- Имеется еще **17** команд являющихся не привилегированными и не перехватываемых операционной системой, но чувствительных к режиму исполнения.
- Для решения проблемы виртуализации архитектуры Intel используется дополнительное ПО, получившее название **монитор виртуальных машин**.

Монитор виртуальных машин

- Идея монитора виртуальных машин не нова. Она была предложена еще в 1974 г. в работе Г. Popek, P. Goldberg, «Formal Requirements for Virtualizable Third Generation Architectures».
- Монитор виртуальных машин обеспечивает интерфейс между гостевой ОС и хозяйской ОС. При этом подавляющая часть кода приложения, гостевой ОС, а также хозяйской ОС выполняется напрямую на аппаратных средствах физической машины (все общие команды и часть привилегированных).
- И лишь небольшая часть кода, которую составляют некоторые привилегированные команды перехватываются монитором виртуальных машин, модифицируются им, и затем передаются на исполнение аппаратным средствам.

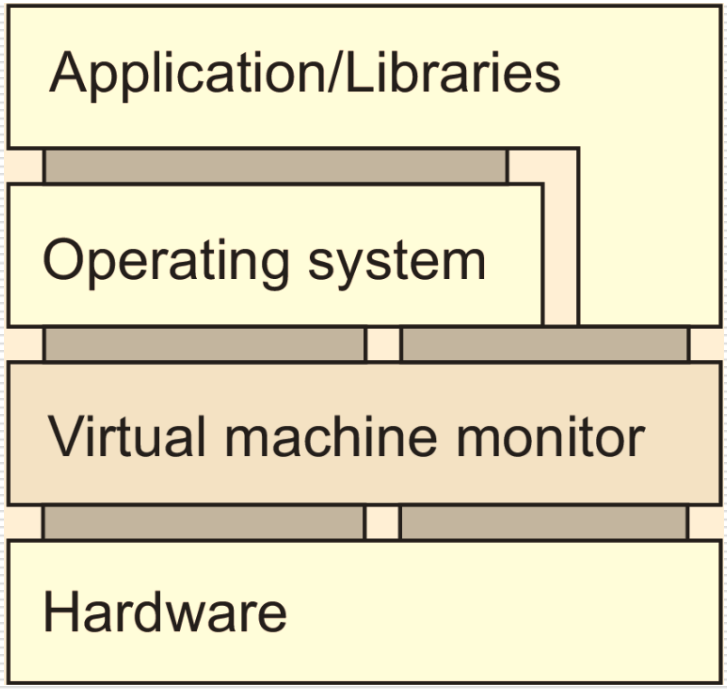


Монитор виртуальных машин получил название Гипервизор.
Различают:
гипервизоры I и II типов.

Два подхода к реализации монитора виртуальных машин

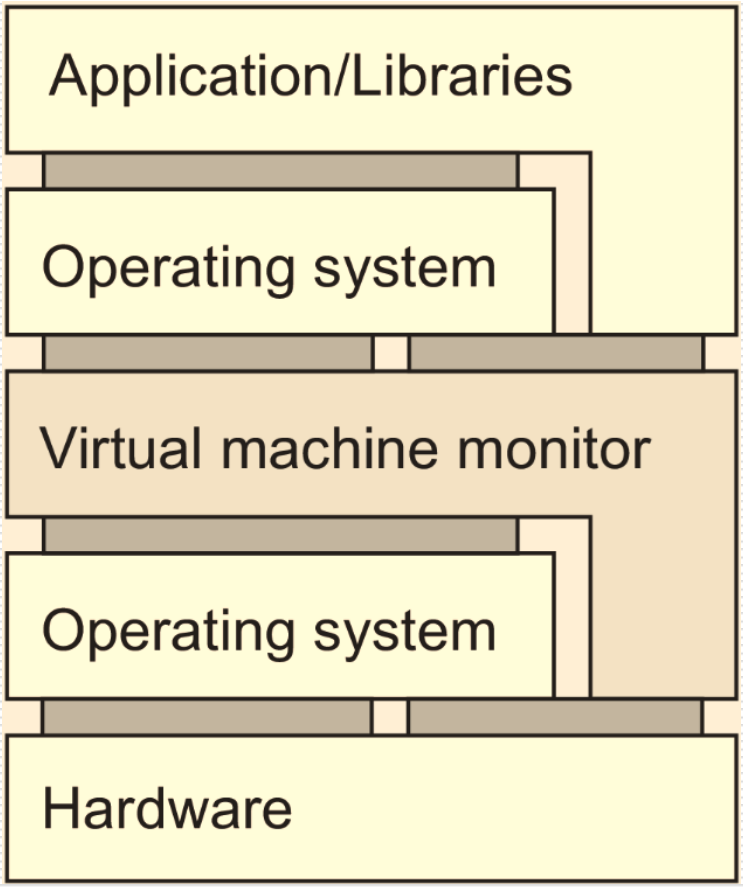
- Имеются два подхода к построению мониторов виртуальных машин:
  - **Эмуляция** всех инструкций, что влечет за собой резкое падение производительности. Этот подход в измененном виде используется современными **мониторами виртуальных машин**.
  - Использование **паравиртуализации**, которая предполагает модификацию гостевых ОС, таким образом, чтобы нейтрализовать, либо полностью исключить влияние “чувствительных” команд, средствами хозяйской ОС.

Монитор виртуальных машин исполняемый как отдельная ОС (Гипервизор первого типа)



- На аппаратные средства устанавливается специализированная ОС (гипервизор) предназначенная для виртуализации на уровне набора команд аппаратной платформы (Vmware ESX).
- Такой монитор получил название “родного” (naitive) монитора виртуальных машин.
- Достоинства данной технологии заключаются в:
  - отсутствии потребности в хостовой ОС – ВМ, устанавливаются фактически на "голое железо", а аппаратные ресурсы используются более эффективно.
- Недостатки:
  - необходимость поддержки в гипервизоре собственных драйверов внешних устройств, из-за чего возникают высокие дополнительные накладные расходы на используемые аппаратные ресурсы,
  - отсутствие учета особенностей гостевых ОС, меньшая, чем нужно, гибкость в использовании аппаратных средств

Монитор виртуальных машин исполняемый в среде хозяйской ОС (Гипервизор второго типа)



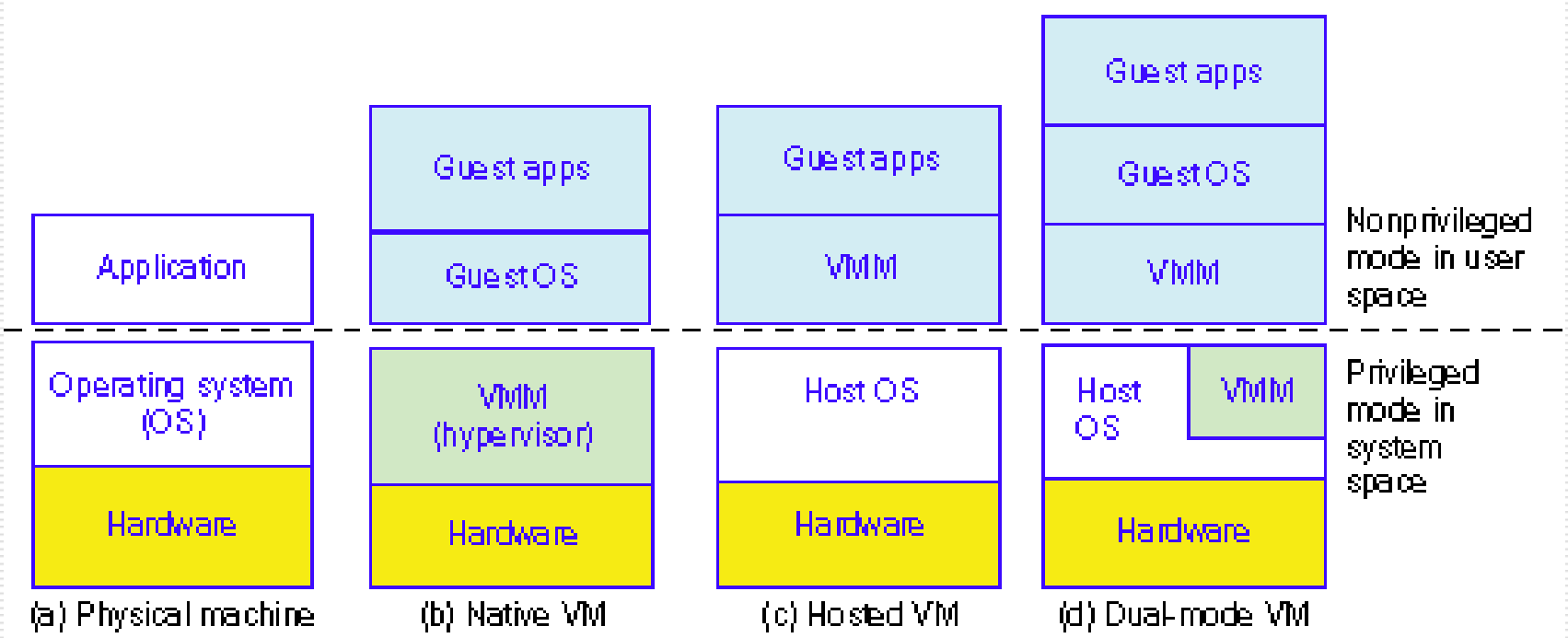
- Монитор виртуальных машин работает в рамках хозяйской ОС.
- В процессе своей работы он модифицирует код гостевых ОС с целью обеспечения их виртуализации на уровне набора команд.
- При этом код содержащий “чувствительные” команды переписывается на лету (Vmware WorkStation).

Паравиртуализация

- техника виртуализации, при которой гостевые операционные системы подготавливаются для исполнения в виртуализированной среде, для чего их ядро незначительно модифицируется.
- Операционная система взаимодействует с программой гипервизора, который предоставляет ей гостевой API, вместо использования напрямую таких ресурсов, как таблица страниц памяти.
- Этот подход не только поддерживает высокую производительность, но и позволяет формировать гетерогенную среду, в которой работает несколько гостевых операционных систем.
- Накладные расходы составляют 5-10%

Архитектура физической машины (a) и три архитектуры виртуализации (b-d)

- Монитор виртуальных машин (VMM – Virtual Machine Monitor)



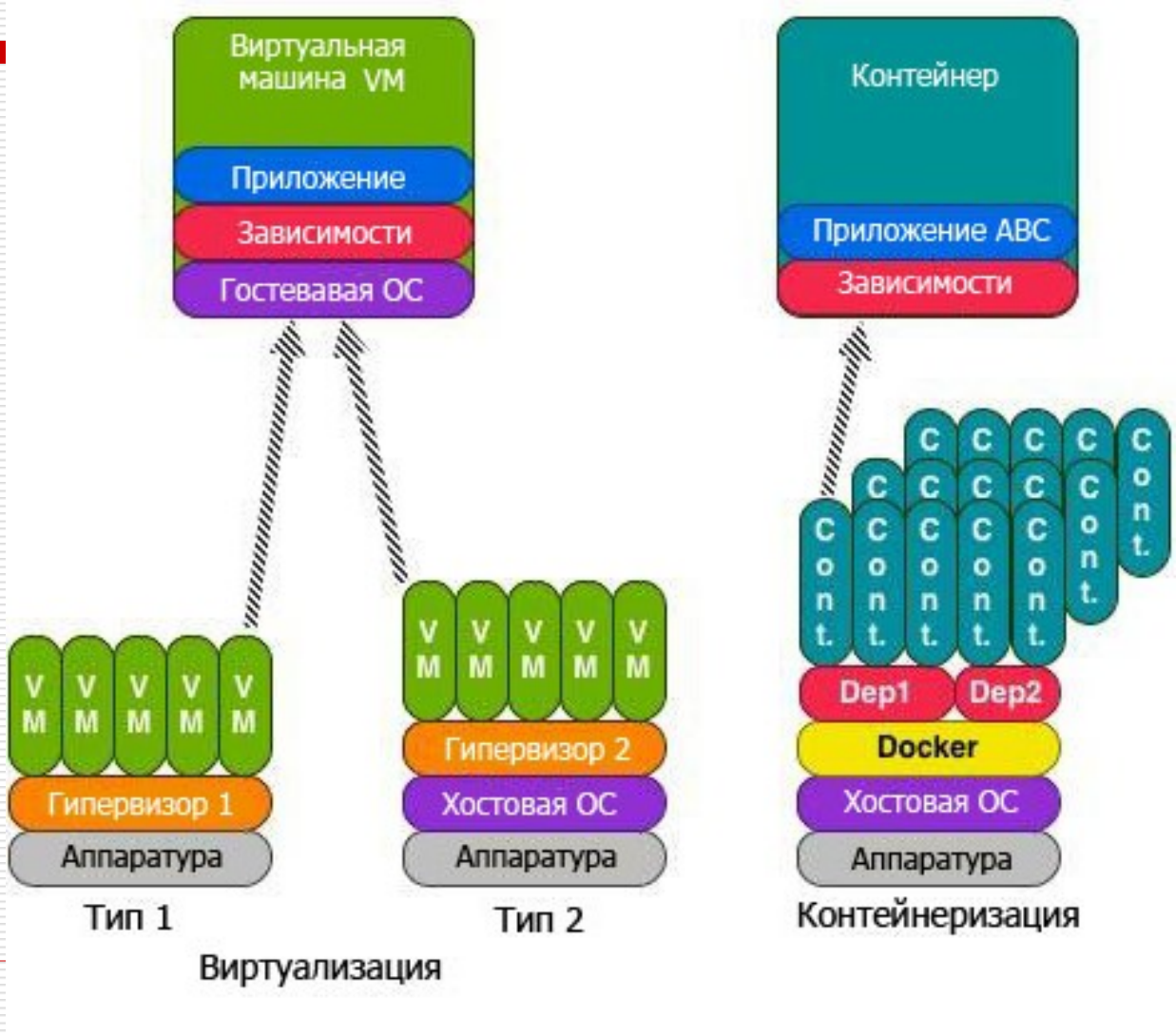
Виртуализация на уровне операционной системы

- — виртуализирует физический сервер на уровне ОС, позволяя запускать изолированные виртуальные серверы называемые Виртуальные Частные Серверы (Virtual Pivate Servers, VPS) или Контейнеры (Container, CT).
- Виртуализация на уровне операционной системы имеет минимальные накладные расходы и обеспечивает самую высокую степень консолидации, однако эта технология не позволяет запускать ОС с ядрами, отличными от ядра базовой ОС.
- **Накладные потери производительности контейнеров составляют 3%**

Контейнерная виртуализация в Linux

- Экземпляры пространств пользователя (часто называемые контейнерами или зонами) с точки зрения пользователя полностью идентичны реальному серверу, но они в своей работе используют один экземпляр ядра операционной системы.
- Для linux-систем, эта технология может рассматриваться как улучшенная реализация механизма chroot.
- Ядро обеспечивает полную изолированность контейнеров, поэтому программы из разных контейнеров не могут воздействовать друг на друга.
- **Наиболее распространены сейчас OpenVZ, LXC, FreeBSD jail и Solaris Containers**

Виртуализация против контейнеризации



Виртуализация и распределенные системы

- Одной из важнейшей причиной, но не единственной, появления технологии виртуализации, явилась необходимость выполнения унаследованного ПО.
- По мере того как стоимость аппаратных средств снижалась, потребность в виртуализации снижалась и не была актуальной до 90-х г.г. XX века.
- Однако в конце 90-х необходимость в виртуализации вновь появилась, в связи с ростом темпов обновления архитектур процессоров, при том, что темпы сменяемости ПО не изменились.
- Виртуализация может обеспечить практически немедленный перенос унаследованного ПО на новые аппаратные платформы. При этом каждое приложение (сервис), включая требуемую ОС и все необходимые ему библиотеки может работать на отдельной виртуальной машине. А группа виртуальных машин может быть развернута на единой аппаратной платформе (физическом сервере).
- При виртуализации обеспечивается высокая степень переносимости ПО между физическими аппаратными платформами, что является важнейшей причиной почему виртуализация играет ключевую роль во многих распределенных системах.

Спасибо за внимание !