Тема 2
Создание, хранение и исполнение виртуальных машин

Что такое виртуальная машина?

- **Виртуальная машина (ВМ)** - программная или аппаратная среда, эмулирующая аппаратное обеспечение некоторой платформы, также спецификация некоторой вычислительной среды (например: «виртуальная машина языка программирования Си»).

- ВМ исполняет некоторый машинно-независимый код (например, байт-код, шитый код, р-код или машинный код реального процессора);
- ВМ может имитировать работу отдельных компонентов аппаратного обеспечения и/или целый реальный компьютер(BIOS, оперативная память, жесткий диск и другие периферийные устройства);
- В ВМ и на реальный компьютер можно устанавливать ОС. На одном компьютере может функционировать несколько виртуальных машин.

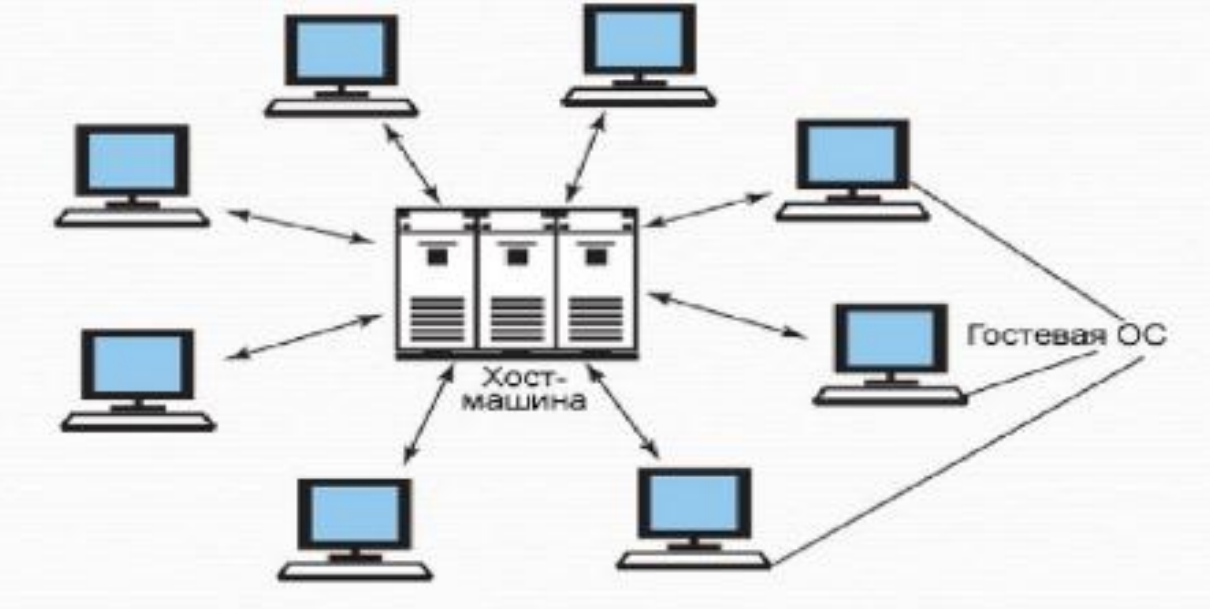
Архитектура виртуальной машины

- Хостовая ОС и монитор ВМ разделяют между собой права на управление аппаратными компонентами компьютера;
- Монитор ВМ контролирует распределение ресурсов между запущенными ВМ, создавая для них иллюзию непосредственного доступа к аппаратному уровню(этот механизм называют виртуализацией);
- Гостевые ОС в пределах выделенных им ресурсов управляют работой запущенных под их правлением приложений.

Архитектура виртуальной машины

Гостевая ОС - операционная система, работающая в виртуальной машине;

На одном реальном компьютере может быть запущена одна хостовая и одна или множестве гостевых ОС.



Как реализуется изоляция ВМ друг от друга и гипервизора

Обычно приложения работают в изолированном адресном пространстве и взаимодействуют с оборудованием при помощи интерфейса API, предоставляемого ОС.

- Если две ОС совместимы по своим интерфейсам API(например Windows 98 и Windows ME), то приложения, разработанные для одной из них, будут работать и под управлением другой.
- Если же две ОС несовместимы, необходимо обеспечить перехват обращений приложения к API гостевой ОС и имитировать ее поведение средствами хостовой ОС.

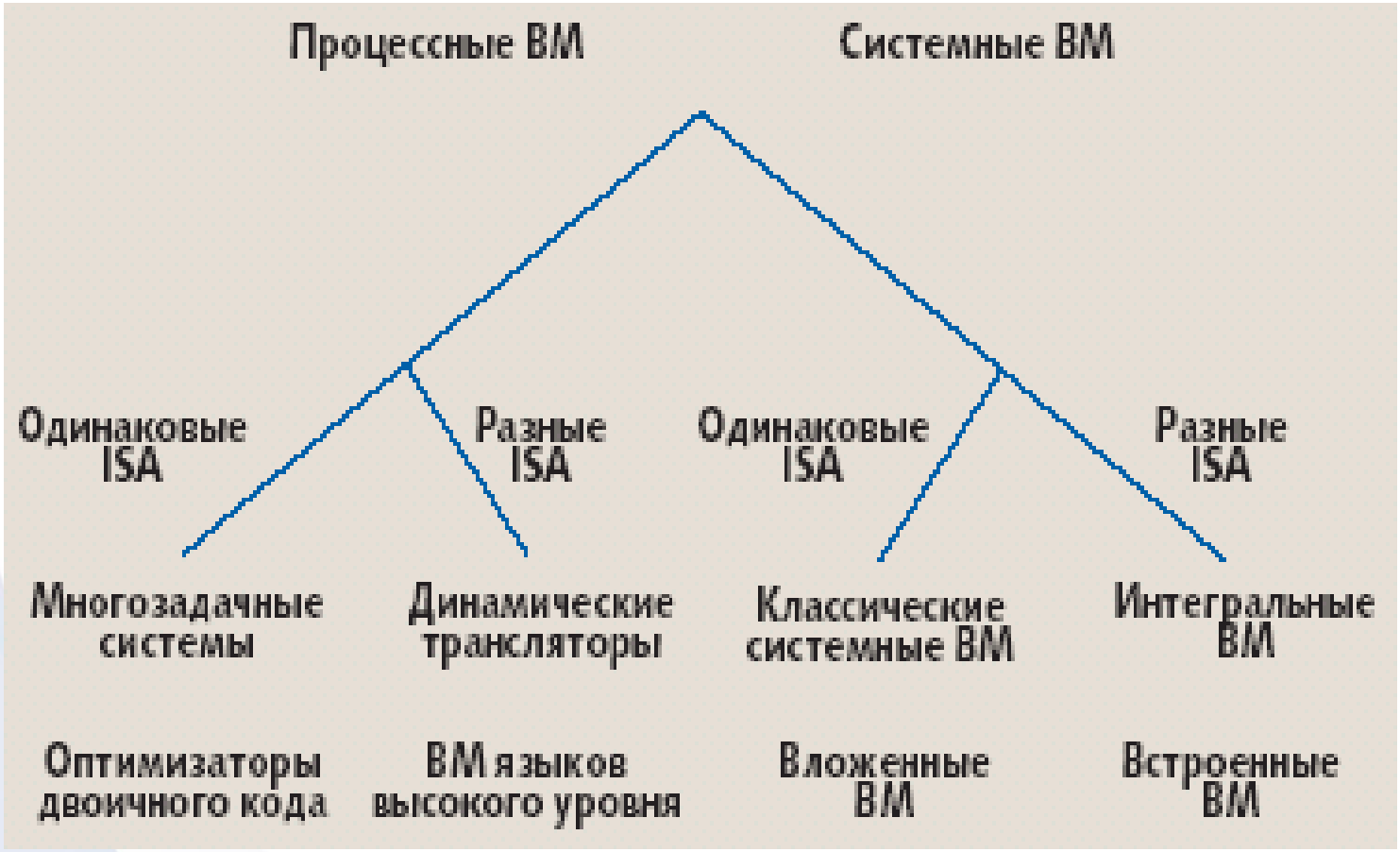
При таком подходе можно устанавливать одну операционную систему и работать одновременно как с ее приложениями, так и с приложениями другой ОС.

Как реализуется изоляция ВМ друг от друга и гипервизора

Гипервизор (монитор виртуальных машин) — программа или аппаратная схема, обеспечивающая или позволяющая одновременное, параллельное выполнение нескольких ОС на одном и том же хост-компьютере.

- обеспечивает изоляцию операционных систем друг от друга, защиту и безопасность, разделение ресурсов между различными запущенными ОС и управление ресурсами.
- предоставляет работающим под его управлением на одном хост-компьютере ОС средства связи и взаимодействия между собой так, как если бы эти ОС выполнялись на разных физических компьютерах.
- Гипервизор сам по себе в некотором роде является минимальной операционной системой (микроядром или наноядром).

Классификация виртуальных машин



Процессные виртуальные машины

Процессные ВМ создают виртуальные среды ABI или API для пользовательских приложений.

1) **Многозадачные системы.** Самая распространенная процессная ВМ. Большинство ОС могут работать в многозадачном режиме, поддерживая несколько пользовательских процессов одновременно. Каждый процесс имеет собственные адресное пространство, регистры и файловую структуру. ОС организует работу базового оборудования в режиме разделения времени;
2) **Оптимизаторы двоичного кода.** Для повышения производительности динамические трансляторы иногда оптимизируют двоичный код. Эта возможность приводит к идее создания виртуальной машины, в которой гость и хост используют одну и ту же систему команд. Единственным назначением такой ВМ становится оптимизация двоичного кода. Оптимизаторы используют информацию из профиля, собранную при интерпретации или трансляции, чтобы оптимизировать двоичный код «на лету»;

Процессные виртуальные машины

3) **Интерпретаторы и динамические трансляторы двоичного кода.**

Более сложной проблемой для процессных ВМ является поддержка программ в двоичном коде, которые скомпилированы для системы команд, отличающейся от системы команд хоста. Проще всего эмулировать команды путем интерпретации.

Программа-интерпретатор одну за другой выбирает из памяти, декодирует и выполняет команды гостевой программы. Этот процесс может быть довольно медленным и требовать десятков команд хост-машины для интерпретации каждой исходной команды. Более высокая производительность достигается путем динамической трансляции двоичного кода, в ходе которой команды гостевой программы поблочно преобразуются в команды хоста и сохраняются для повторного использования в кэш-памяти команд.

Таким образом, в случае циклического выполнения транслированных команд удается амортизировать относительно высокие затраты на трансляцию.

Процессные виртуальные машины

4) **ВМ для языков высокого уровня.** Основное назначение процессных ВМ — обеспечить переносимость программного обеспечения. При этом эмуляция одной распространенной системы команд с помощью другой требует значительных усилий программистов, но обеспечивает кроссплатформенную совместимость лишь в некоторых случаях.

Полной переносимости проще достичь путем встраивания процессной ВМ в среду разработки приложений на языках высокого уровня. ВМ языка высокого уровня не связана напрямую ни с одной реальной платформой; скорее, она предназначена для упрощения переноса программ и реализации функций одного или нескольких языков высокого уровня.

Преимуществом ВМ языка высокого уровня является простота переноса прикладных программ при условии, что ВМ и библиотеки уже реализованы на базовой платформе. Хотя реализация ВМ требует некоторых усилий, это намного проще, чем разработка полнофункционального транслятора и перенос каждого приложения путем перекомпиляции.

Системные виртуальные машины

Обеспечивает полнофункциональную среду, в которой могут сосуществовать ОС и несколько процессов, относящихся к разным пользователям. С помощью системных ВМ одна аппаратная платформа способна одновременно поддерживать несколько изолированных гостевых ОС.Одним из важнейших применений технологии системных ВМ является изоляция систем, одновременно работающих на общей аппаратной платформе. В таких случаях отказ в работе или нарушение безопасности одной из гостевых систем не влияет на программное обеспечение, выполняющееся на других гостевых системах.

Большинство системных ВМ обеспечивают примерно одинаковые функциональные возможности, но различаются деталями реализации.

Системные виртуальные машины

1) **Классические системные ВМ.** При классическом подходе монитор VMM устанавливается на «голом железе», а ВМ располагается поверх него. VMM выполняется в привилегированном режиме, в то время как гостевые операционные системы лишены почти всех привилегий. Это необходимо, чтобы VMM мог перехватывать и эмулировать те их действия, которые в обычных условиях связаны с управлением жизненно важными аппаратными ресурсами.
2) **Вложенные ВМ.** В альтернативной реализации системной ВМ программное обеспечение виртуализации располагается поверх существующей хост-системы. Такая ВМ называется вложенной. Преимущество вложенных ВМ состоит в том, что пользователь устанавливает их точно так же, как любую прикладную программу. Чтобы получить доступ к драйверам устройств и другим низкоуровневым услугам, программное обеспечение виртуализации обращается к хост-системе, а не к VMM.

Системные виртуальные машины

3) **Интегральные ВМ.** В обычных системных ВМ хост-система, все гостевые операционные системы и прикладные программы используют архитектуру ISA базового оборудования. Однако в некоторых ситуациях гость и хост не имеют общей ISA. Интегральные ВМ справляются с такой ситуацией, виртуализируя все ПО, в том числе ОС и приложения. Из-за различий в системах команд ВМ должна эмулировать и код приложений, и код операционной системы.

Системные виртуальные машины

4) **Встроенные ВМ.** Реализуют новые частные архитектуры, нацеленные на повышение производительности и эффективности работы. ISA хост-машины может быть полностью новой или расширять возможности существующей. Для встроенной ВМ нет готовых приложений, а монитор VMM выглядит как составная часть оборудования, единственным назначением которого является эмуляция гостевой ISA. Для поддержания этой иллюзии VMM располагается в области памяти, скрытой от обычного программного обеспечения. В его состав входит транслятор двоичного кода, который преобразует гостевые команды в оптимизированные последовательности базовых команд и помещает их в скрытую кэш-память.

Системные виртуальные машины

5) **Многопроцессорная виртуализация.** Особый интерес представляет виртуализация большого многопроцессорного комплекса с разделяемой памятью. Важно разделить крупную систему на несколько меньших многопроцессорных систем, распределив аппаратные ресурсы базовой системы.

С помощью физического разбиения на разделы аппаратные ресурсы, используемые одной виртуальной системой, отделяются от ресурсов других виртуальных систем. Физическое разделение обеспечивает высокую степень изоляции, поэтому ни проблемы ПО, ни ошибки аппаратных средств в одном разделе не затрагивают программы в других разделах.

Системные виртуальные машины

При логическом разбиении на разделы разделы сервера совместно используют базовые аппаратные ресурсы в режиме временного мультиплексирования, что позволяет повысить коэффициент загрузки оборудования. Однако при этом теряются некоторые из преимуществ изоляции аппаратных средств. При реализации обоих методов обычно используются дополнительное программное обеспечение или микропрограммы для базового оборудования, специально доработанного для обеспечения поддержки логических разделов.

Преимущества виртуальных машин

Основные преимущества:

1) В рамках ВМ можно работать с устаревшими программными решениями и ОС;
2) Возможность создать защищенные пользовательские окружения для работы с сетью, в этом случае вирусные атаки могут нанести вред ОС, а не ВМ;
3) Несколько ВМ, развернутых на физических ресурсах одного компьютера, изолированы друг от друга, таким образом, сбой одной из виртуальных машин не повлияет на доступность и работоспособность сервисов и приложений других;

Преимущества виртуальных машин

4) ВМ идеально подходят для процессов обучения и переподготовки, поскольку позволяют развернуть требуемую платформу вне зависимости от параметров и ПО хоста (физического компьютера, на котором функционирует ВМ);
5) В рамках одной гостевой операционной системы может быть развернуто несколько ВМ, объединенных в сеть и взаимодействующих между собой;
6) Виртуальные машины могут создавать представления устройств, которых физически нет (эмуляция устройств).

Недостатки виртуальных машин

1) Обеспечение единовременной работы нескольких виртуальных машин потребует достаточного количества аппаратных мощностей;
2) В зависимости от используемого решения, операционная система ВМ может работать медленнее, чем на “чистом” аналогичном аппаратном обеспечении;
3) Различные платформы виртуализации не поддерживают виртуализацию всего аппаратного обеспечения и интерфейсов.

Формат хранения образов

- **raw**
  Представляет собой набор данных, записанных в том же формате, как и на обычном блочном устройстве. Файл размера 5G будет занимать это место целиком, независимо от содержания полезных данных.
- **qcow**
  Отличается от формата raw тем, что может быть инкрементным, т.е. постепенно увеличивается размер файла, поддерживает AES шифрование и сжатие zlib. Файл нельзя смонтировать на машину непосредственно на которой он находится (по сети или эмуляцию).
- **qcow2**
  Представляет собой обновленную версию формата qcow. Основное отличие заключается в том, что он поддерживает снапшоты.

Формат хранения образов

- **vmdk**
  формат виртуальных дисков, используемый продуктами VMware. Формат который позволяет файлу постепенно увеличиваться. Позволяет иметь файл виртуального диска, разделенный на несколько мелких файлы с размерам по 2 ГБ. Высокая скорость репликации и синхронизации.
- **vdi**
  формат виртуального диска, используемый системой виртуализации Oracle Virtualbox.
- **vhdx**
  формат виртуального диска, используемый системой виртуализации Microsoft Hyper-V.

Формат хранения образов

- **qed**
  Это инкрементный формат COW виртуального диска, который создает наименьшую нагрузку на хост. Он не поддерживает никакого сжатия и использует две параллельные таблицы для адресации блоков данных.
- **ovf**
  описывает открытый, переносимый, расширяемый формат для распространения образов виртуальных машин. Не привязан к какой-либо реализации гипервизора или аппаратной архитектуре.
- **vpc**
  формат виртуального диска используемый системой виртуализации Microsoft VirtualPC.

Хранение виртуальных машин

Создатели придумали достаточно разумный ход использования мощностей компьютера: жесткий диск созданной ВМ занимает ровно столько места из резервов реального компьютера, сколько ему необходимо в данный момент времени.

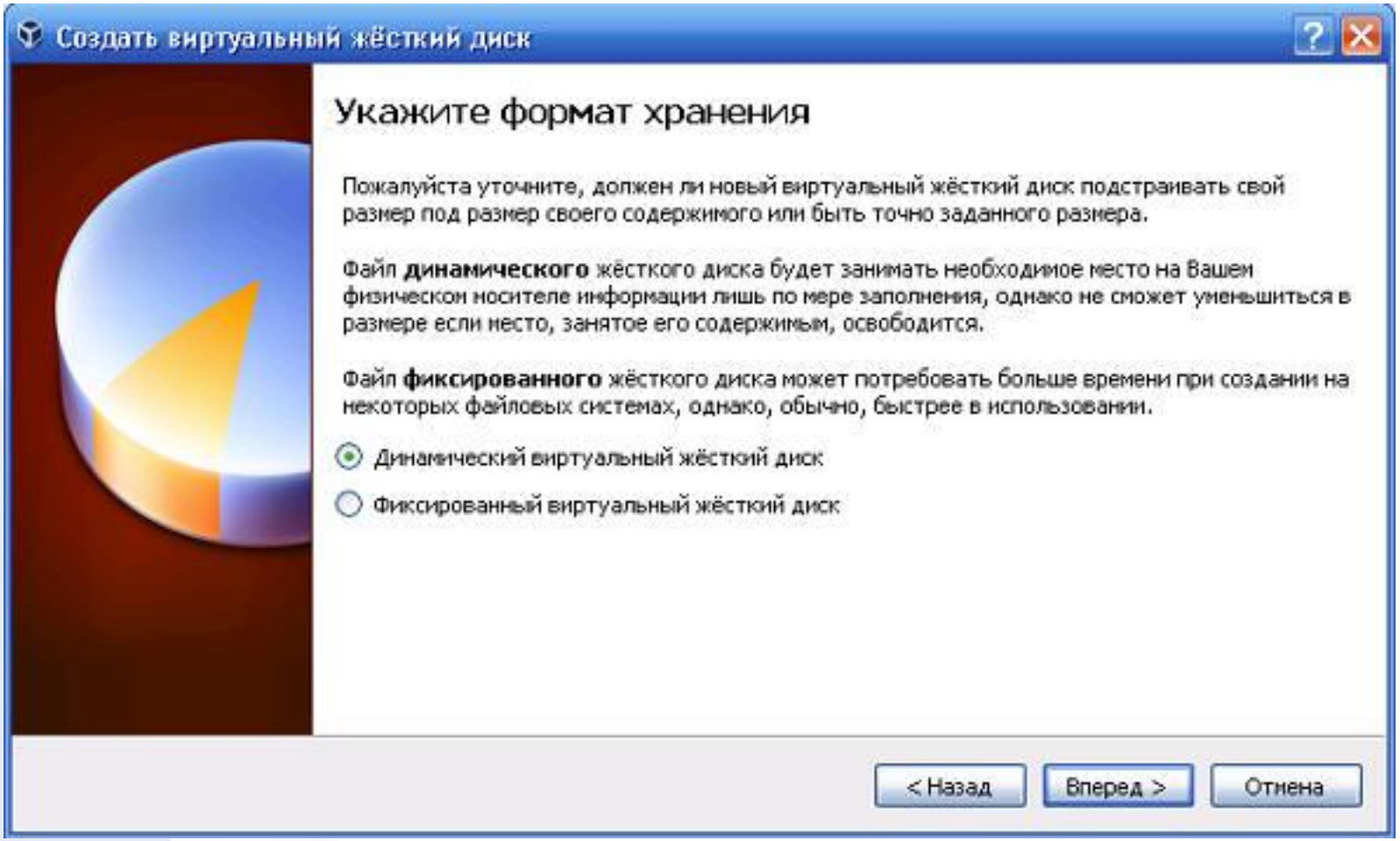
В целом, ВМ является программой, состоящей из двух файлов, каждый из которых записывается на жесткий диск реального компьютера.

Хранение виртуальных машин

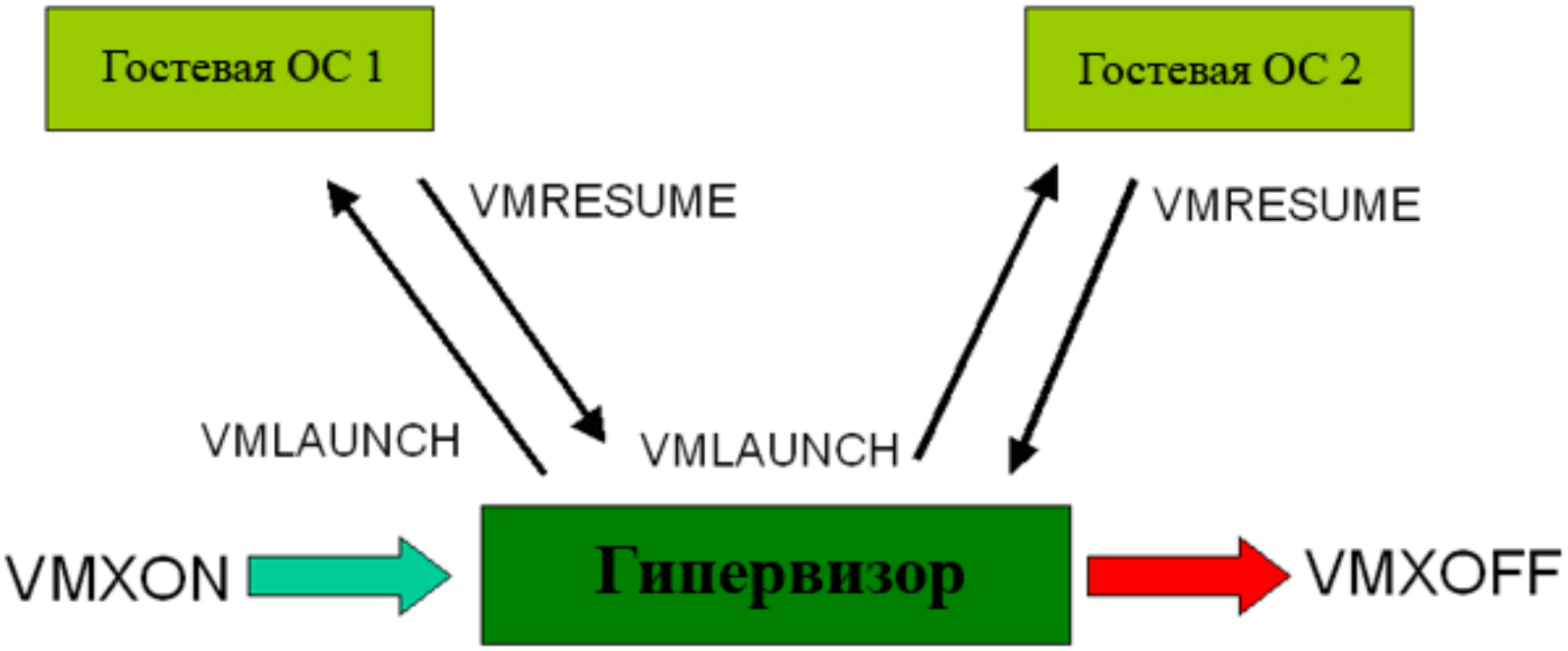
Информация в файлах разделена следующим образом:

- файл меньшего размера содержит описание ВМ и сведения о состоянии оперативной памяти, объеме жесткого диска, разрешении экрана;
- второй файл имеет больший размер и является непосредственным местом хранения жесткого диска виртуального компьютера.

Бывают ситуации, когда возникает необходимость использования большего места, чем это позволяет размер виртуального жесткого диска. Тогда недостающее виртуальное пространство восполняется резервами реального компьютера.


Создать виртуальный жёсткий диск
Укажите формат хранения
Пожалуйста уточните, должен ли новый виртуальный жёсткий диск подстраивать свой размер под размер своего содержимого или быть точно заданного размера.
Файл динамического жёсткого диска будет занимать необходимое место на вашем физическом носителе информации лишь по мере заполнения, однако не сможет уменьшиться в размере если место, занятое его содержимым, освободится.
Файл фиксированного жёсткого диска может потребовать больше времени при создании на некоторых файловых системах, однако, обычно, быстрее в использовании.
Динамический виртуальный жёсткий диск
Фиксированный виртуальный жёсткий диск
< Назад
Вперед >
Отмена

Процедура запуска виртуальных машин



VMX предоставляет следующие инструкции: VMPTRLD, VMPTRST, VMCLEAR, VMREAD, VMREAD, VMWRITE, VMCALL, VMLAUNCH, VMRESUME, VMXON и VMXOFF.

Исполнение виртуальных машин

- **Предварительное планирование.**

Особое внимание следует уделить размеру будущей инфраструктуры. В облачной инфраструктуре не всегда работает правило «больше, значит лучше», ресурсы необходимо выделять, проанализировав данные мониторинга. Полученные сведения позволят определить объем ресурсов, которые потребуются для будущих ВМ в облаке провайдера.

Перенос приложений из локальной инфраструктуры в облачную требует обязательного контроля корректности работы ПО. Работа приложения внутри ВМ может отличатся от локальной, прежде чем его запускать его в «продакшен» необходимо выполнить тестирование. Это позволит грамотно рассчитать нагрузку на хост арендуемой виртуальной инфраструктур и не допустить падения производительности приложений в рабочем режиме.

Исполнение виртуальных машин

- **Особенности правильной настройки виртуальных машин.**

Большинство облачных провайдеров используют платформу *VMware* для предоставления услуги виртуальной инфраструктуры IaaS. Настраивая ВМ в этой среде, важно обратить внимание на число виртуальных процессоров. Рекомендуем не использовать больше 8 vCPU. Ограничения обусловлены использованием архитектуры неравномерного доступа к памяти — **NUMA**.

Если ВМ назначено больше ядер и ОП, чем физически приходится на один процессор, она начнет обращение к другому процессору, это процесс медленный и снижает производительность ВМ. В NUMA-архитектуре для каждого процессора выделяется собственная локальная память, совокупность CPU и RAM объединяется в NUMA-узел. Если ВМ не хватает ресурсов своего узла, она обращается к памяти другого, скорость такой транзакции медленнее чем при работе с локальной RAM. Идеально, когда ВМ работает с одним физическим NUMA-узлом.

Исполнение виртуальных машин

- **Как добавлять ресурсы ВМ на лету.**

Если в процессе работы ВМ ее производительности становится недостаточно, платформа WMware позволяет добавить вычислительные ресурсы без остановки машины — на лету. Для этого необходимо воспользоваться опцией Hot Add, с помощью которой можно увеличить ресурсы процессора и объем назначенной оперативной памяти.

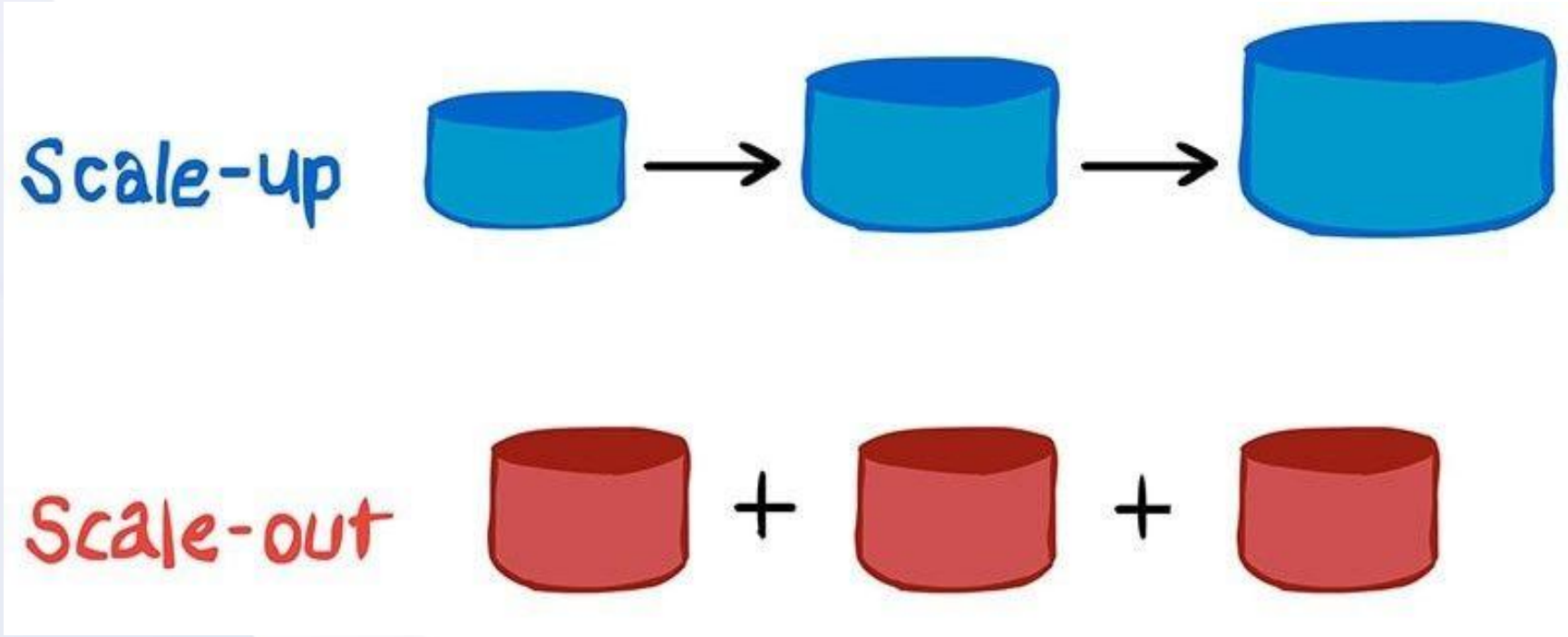
Однако, Hot Add для процессора (в настройках ВМ — Hot Plug) отключает vNUMA.

Исполнение виртуальных машин

В результате, операционная система ВМ будет расценивать всю свободную память как доступную без разделения по узлам NUMA, что может привести к падению производительности, так как доступ к памяти другого процессора более медленный. Прежде чем включить Hot Add для процессора необходимо проанализировать как это скажется на конкретном приложении.

Исполнение виртуальных машин

- **Разница в подходах масштабирования Scale UP и Scale OUT.**



Исполнение виртуальных машин

Вариант *вертикального масштабирования Scale UP* выбирают, как наиболее простой и понятный. Все приложения и базы данных с одного мощного локального хоста переносят на одну виртуальную машину, а в случае, когда ее ресурсов начинает не хватать, администратор добавляет еще оперативной памяти и число процессоров или ядер. Подробнее о таком подходе можно прочитать в статье Виртуальный ЦОД.

В свою очередь при использовании *горизонтального масштабирования — Scale OUT*, мощность локального хоста распределяется между несколькими виртуальными машинам в составе одного кластера. К преимуществам такого похода можно отнести большую гибкость по управлению производительностью каждого из приложений, а также большую отказоустойчивость, так как в случае выхода из строя одной вычислительной единицы (виртуального сервера), остальная инфраструктура продолжает работать.

Исполнение виртуальных машин

- **Настройка использования оперативной памяти.**

В процессе распределения оперативной памяти между виртуальными машинами следует помнить про обслуживание физического хоста. Для его нормального функционирования требуется зарезервировать несколько гигабайт RAM, который не будет задействован при создании ВМ. Заранее спланируйте как будет распределяться доступная физическая память между созданными виртуальными машинами.

Вопросы для контроля

1. Что называется виртуальной машиной?
2. Какие преимущества у виртуальной машины? Какие недостатки?
3. Что такое гипервизор?
4. Классификация ВМ(перечислить)?
5. Процессные ВМ.
6. Системные ВМ.
7. Перечислите форматы хранения образов.
8. Разница в подходах масштабирования Scale UP и Scale OUT.