

БГУИР

Кафедра ЭВМ

Контрольная работа
Тема: «Виртуальные машины»

Выполнила:
студент группы 355841 Деркач А.В.

Проверил:
Селезнёв И.Л.
к.т.н., доцент _____

Минск
2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 ЦЕЛЬ РАБОТЫ И ПРОРАБАТЫВАЕМЫЕ ВОПРОСЫ.....	4
1.1 Цель работы	4
1.2 Прорабатываемые вопросы в рамках работы.....	4
2 ВИРТУАЛИЗАЦИЯ И УРОВНИ ЕЁ РЕАЛИЗАЦИИ.....	5
2.1 Определение виртуализации.....	5
2.2 Преимущества виртуализации.....	6
2.3 Уровни виртуализации и их реализация.....	7
3 ПРОГРАММНАЯ И АППАРАТНАЯ ВИРТУАЛИЗАЦИЯ	9
3.1 Программная виртуализация	9
3.2 Аппаратная виртуализация	14
4 СОВРЕМЕННЫЕ СРЕДЫ ВИРТУАЛИЗАЦИИ	18
4.1 Среда виртуализации VirtualBox	18
4.2 Среда виртуализации Hyper-V	19
4.3 Среда виртуализации VMware Workstation	21
ЗАКЛЮЧЕНИЕ	24
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	25

ВВЕДЕНИЕ

В современном информационном обществе, где технологический прогресс стремительно продвигает границы возможностей, виртуализация и виртуальные машины становятся ключевыми элементами цифровой революции. Эти технологии преобразуют традиционные подходы к работе с компьютерами и серверами, предоставляя уникальные возможности для оптимизации ресурсов, повышения эффективности и улучшения общей производительности информационных систем.

Основной принцип виртуализации заключается в абстрагировании программного обеспечения от аппаратных ресурсов, что упрощает управление и повышает надежность системы. Виртуальные машины могут быть легко масштабируемыми и перемещаемыми между различными физическими серверами, что обеспечивает высокую доступность и отказоустойчивость.

Технологии виртуализации, следуя требованиям современного бизнеса, претерпели в промежуток с 2005 года по настоящее время существенные изменения.

- повышается роль технологий виртуализации в формировании ИТ инфраструктуры предприятия за счет потенциального сокращения серверов и оптимизации нагрузки на имеющееся оборудование;

- рост числа компактных вычислительных устройств и беспроводного доступа к интернет порождают необходимость образования центров обработки данных (ЦОД).

Актуальность темы виртуализации и виртуальных машин подчеркивается не только в корпоративной сфере, но и в области разработки программного обеспечения, тестирования, обучения и даже в исследованиях. Эти технологии открывают новые перспективы для инноваций и улучшений в различных областях, делая их неотъемлемой частью современного цифрового общества. В данной контрольной работе будет рассмотрена суть виртуализации, принципы ее работы, а также основные преимущества, которые она приносит в различные сферы человеческой деятельности.

1 ЦЕЛЬ РАБОТЫ И ПРОРАБАТЫВАЕМЫЕ ВОПРОСЫ

1.1 Цель работы

Целью контрольной работы является изучение современных подходов и методов практической реализации технологий виртуализации вычислительных систем.

1.2 Прорабатываемые вопросы в рамках работы

Вопросы, прорабатываемые в рамках контрольной работы:

- виртуализация и уровни её реализации;
- программная и аппаратная виртуализация;
- характеристики работы современных сред виртуализации;
- установка виртуальных машин VMWare Workstation, VirtualBox и Virtual PC (Hyper-V).

2 ВИРТУАЛИЗАЦИЯ И УРОВНИ ЕЁ РЕАЛИЗАЦИИ

2.1 Определение виртуализации

Виртуализация – это процесс, который позволяет компьютеру совместно использовать свои аппаратные ресурсы с несколькими средами, разделенными цифровым способом. Каждая виртуализированная среда работает в пределах выделенных ресурсов, таких как память, вычислительная мощность и хранилище (см. рисунок 2.1). Это технология, которую можно использовать для создания виртуальных представлений серверов, хранилища, сетей и других физических устройств. Виртуальное программное обеспечение подражает функциям физического оборудования для одновременной работы виртуальных машин на одной физической машине. Предприятия применяют виртуализацию, чтобы эффективно использовать аппаратные ресурсы и обеспечивать работу облачных вычислительных сервисов [1].

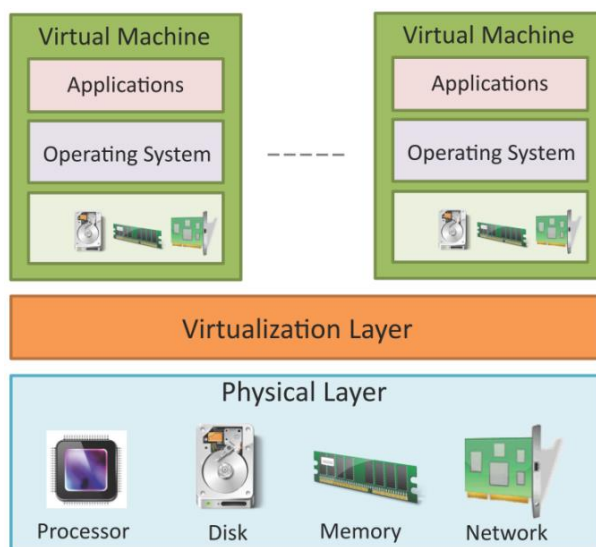


Рисунок 2.1 – Общая схема понятия виртуализации

Виртуальная машина – это программно-определяемый компьютер, работающий на физическом компьютере с отдельной операционной системой и вычислительными ресурсами. Физический компьютер называется хост-машиной, а виртуальные машины – гостевыми машинами. На одной физической машине может работать несколько виртуальных машин. Виртуальные машины абстрагируются от аппаратного обеспечения компьютера с помощью гипервизора.

Гипервизор – это программный компонент, управляющий несколькими виртуальными машинами на компьютере. Он гарантирует, что каждая виртуальная машина получает выделенные ресурсы и не мешает работе других виртуальных машин. Существует два типа гипервизоров [2]:

1. Гипервизоры первого типа, также известные как гипервизоры без операционной системы, – это программы-гипервизоры, установленные непосредственно на аппаратном обеспечении компьютера, а не на операционной системе. Поэтому гипервизоры первого типа обладают более высокой производительностью и обычно используются в корпоративных приложениях. Общая схема гипервизора первого типа представлена на рисунке 2.2.

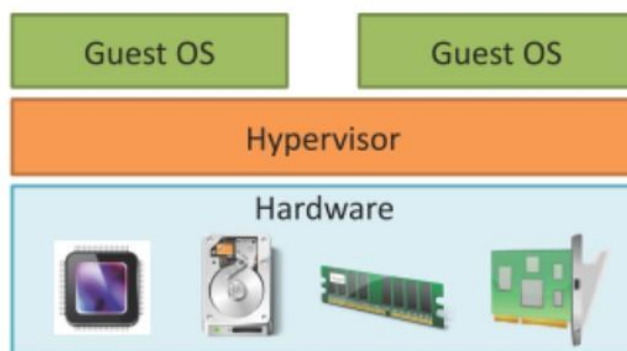


Рисунок 2.2 – Общая схема гипервизора первого типа

2. Гипервизоры второго типа, также известные как размещенные гипервизоры, устанавливаются в операционной системе. Гипервизоры второго типа подходят для вычислительных возможностей конечных пользователей. Общая схема гипервизора второго типа представлена на рисунке 2.3.

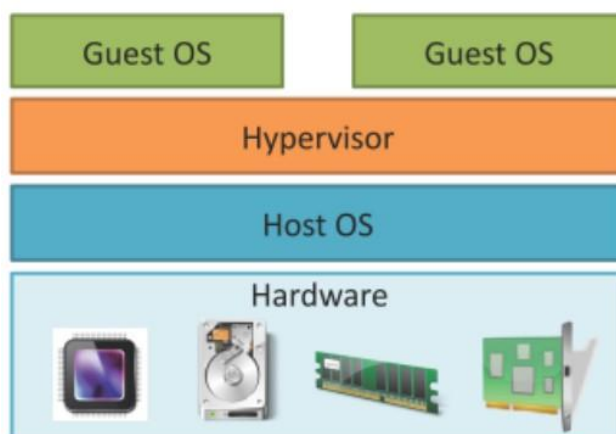


Рисунок 2.3 – Общая схема гипервизора второго типа

2.2 Преимущества виртуализации

Виртуализация предоставляет ряд преимуществ, основные из них [3]:

1. Эффективное использование ресурсов. Использование виртуализации позволяет на одном физическом сервере устанавливать на виртуальных

машинах разные операционные системы без необходимости приобретать отдельные физические серверы для различных задач.

2. Улучшенное управление ресурсами. Виртуализация предоставляет средства для более эффективного управления ресурсами, такими как процессорное время, память и хранилище данных. Администраторы могут динамически изменять выделенные ресурсы виртуальным машинам в соответствии с текущими потребностями, обеспечивая оптимальную производительность и экономию ресурсов.

3. Повышение отказоустойчивости. Виртуальные машины изолированы друг от друга. Проблемы с одной виртуальной машиной или приложением на ней не затрагивают все остальные (в отличие от случаев, когда без применения виртуализации на физический сервер устанавливалось множество служб и приложений). Можно включать, выключать и перезагружать виртуальные серверы независимо друг от друга.

4. Легкость резервного копирования и восстановления. Виртуализация обеспечивает удобные средства для создания резервных копий виртуальных машин и их восстановления в случае сбоев. Это значительно упрощает процессы обеспечения безопасности данных и повышает уровень готовности к аварийным ситуациям.

5. Быстрое развертывание и масштабирование. Виртуализация упрощает процессы создания и управления виртуальными машинами. Новые виртуальные серверы могут быть развернуты значительно быстрее, чем при использовании физических серверов. Кроме того, масштабирование системы, добавление новых ресурсов или изменение конфигурации, становятся более простыми и гибкими операциями.

2.3 Уровни виртуализации и их реализация

Понятие виртуализации чрезвычайно емкое и включает технологии абстрагирования самых разных уровней. Уровни виртуализации различаются по степени абстракции и уровню взаимодействия с программами, операционными системами и аппаратными средствами. Ниже приведены наиболее популярные из них [4].

Уровни аппаратной виртуализации:

– Виртуализация уровня аппаратного обеспечения – включает в себя использование гипервизора для создания виртуальных машин, которые могут выполнять несколько операционных систем на одном физическом сервере. Гипервизор обеспечивает виртуализацию аппаратных ресурсов, таких как процессоры, память и устройства ввода-вывода (I/O). Примеры реализаций: Intel Virtualization Technology (VT-x), AMD Virtualization (AMD-V), Hyper-V, KVM.

– Виртуализация уровня инструкций – позволяет гостевым виртуальным машинам выполнять инструкции, предназначенные для физического процессора. Это достигается переводом или эмуляцией инструкций, которые

могут вызвать конфликты в виртуальной среде. Примеры реализаций: QEMU (Quick Emulator), Bochs.

- Виртуализация памяти – обеспечивает разделение физической памяти на виртуальные блоки, предоставляя каждой виртуальной машине свою собственную виртуальную память. Гипервизор контролирует доступ и распределение физической памяти между виртуальными машинами. Примеры реализаций: VMware Memory Overcommit, Microsoft Hyper-V Dynamic Memory.

- Виртуализация уровня устройств ввода-вывода – обеспечивает виртуальным машинам доступ к физическим устройствам ввода-вывода (например, сетевым картам, дискам) через абстракцию, предоставляемую гипервизором. Примеры реализаций: Intel Virtualization Technology for Directed I/O (VT-d), AMD I/O Virtualization Technology (IOMMU).

Уровни программной виртуализации:

- Серверная виртуализация (иногда употребляют термин «полная виртуализация») – представляет собой полную эмуляцию оборудования на уровне программного обеспечения. Примеры реализаций: VMware vSphere/ESXi, Microsoft Hyper-V, KVM (Kernel-based Virtual Machine).

- Паравиртуализация – техника виртуализации, при которой гостевые операционные системы подготавливаются для исполнения в виртуализированной среде, для чего их ядро незначительно модифицируется. Примеры реализаций: Xen, Citrix Hypervisor (ранее XenServer), Oracle VM Server for x86.

- Виртуализация уровня операционной системы (также имеющая название виртуализации высокого уровня или совместной виртуализации ОС) – виртуализирует сервера на уровне ядра операционной системы. Примеры реализаций: Docker, Kubernetes, Podman.

- Виртуализация ресурсов – позволяет концентрировать, абстрагировать и упрощать управление группами ресурсов, таких как сети, хранилища данных и пространства имен. Примеры реализаций: Apache Mesos, OpenStack, VMware vCloud.

- Виртуализация прикладных приложений – подразумевает, что приложения разрабатываются и упаковываются с их зависимостями, чтобы они могли работать в изолированном окружении и быть переносимыми между различными средами выполнения. Примеры реализаций: Java Virtual Machine (JVM) для Java-приложений, .NET Framework для приложений, использующих технологии Microsoft .NET.

3 ПРОГРАММНАЯ И АППАРАТНАЯ ВИРТУАЛИЗАЦИЯ

3.1 Программная виртуализация

Как было описано в пункте 2.3, программная виртуализация разделяется на серверную виртуализацию, паравиртуализацию, виртуализацию уровня операционной системы, виртуализацию ресурсов и виртуализацию прикладных приложений.

Общая схема серверной виртуализации представлена на рисунке 3.1.



Рисунок 3.1 – Общая схема серверной виртуализации

Данный вид виртуализации основан на применении гипервизора. Примером такой реализации является виртуальная машина – окружение, которое представляется для «гостевой» операционной системы, как аппаратное. Однако на самом деле это программное окружение, которое эмулируется программным обеспечением Host-системы.

Операционная система, управляющая реальным оборудованием и предоставляющая функции для доступа к нему, называется Host-системой, или основной, управляющей операционной системой. Основная операционная система загружается самостоятельно и не требует виртуальной машины для своей работы. Операционные системы, работающие в виртуальных машинах, называются гостевыми операционными системами. На одном физическом компьютере может быть запущена одна основная и много гостевых операционных систем.

Общая системная архитектура виртуальной машины построена на взаимодействии трех основных компонентов:

- приложение виртуальной машины;
- драйвер виртуальных машин;
- монитор виртуальной машины.

Приложение виртуальной машины – это обычное приложение, выполняющееся под управлением основной операционной системы. Приложение виртуальной машины имеет графический интерфейс и позволяет пользователю взаимодействовать с виртуальной машиной и гостевой операционной системой. Приложение является переносимым компонентом

виртуальной машины, поскольку разрабатывается для конкретной управляющей операционной системы и использует ее функции для отображения графического интерфейса и доступа к внешним устройствам. Как правило, для портирования виртуальной машины под другую основную операционную систему, необходимо полностью переписать приложение.

Приложение виртуальной машины построено по многопоточной технологии и поддерживает три следующих основных потока:

- поток виртуализации для передачи управления монитору и обмена информационными сообщениями с ним;
- графический поток для отображения видеобuffers гостевой операционной системы;
- поток GUI для работы пользовательского интерфейса и передачи событий от «мыши» и клавиатуры гостевой операционной системе.

Для каждой виртуальной машины запускается своя копия приложения виртуальной машины. Приложение виртуальной машины выполняет следующие основные функции:

- создание, удаление и конфигурирование виртуальных машин;
- включение, выключение и управление работой виртуальных машин;
- обеспечение интерфейса пользователя с гостевой операционной системой, ввод с клавиатуры/мыши и отображение экрана гостевой операционной системы;
- выделение памяти для виртуальной машины и загрузка монитора виртуальной машины;
- взаимодействие с физическими ресурсами компьютера через функции управляющей операционной системы (работа с жесткими и гибкими дисками, видеокартой, последовательными и параллельными портами и т.д.)

Драйвер виртуальных машин – это системный драйвер, работающий на уровне привилегий ядра управляющей операционной системы. Драйвер является шлюзом между приложением и монитором виртуальной машины, позволяющий им передавать управление и обмениваться информационными сообщениями между собой. Кроме того, драйвер выполняет функции взаимодействия с управляющей операционной системой, такие как выделение и закрепление страниц памяти по физическим адресам. Драйвер виртуальной машины является переносимым компонентом виртуальной машины. Для портирования виртуальной машины под другую основную операционную систему необходимо полностью переписать драйвер, используя средства этой операционной системы. Все виртуальные машины пользуются одной копией драйвера виртуальных машин.

Монитор виртуальной машины или гипервизор – это управляющий компонент виртуальной машины. Благодаря аппаратной поддержке виртуализации стала возможной эффективная работа гипервизора. Монитор не зависит от конкретной управляющей операционной системы и отвечает за создание виртуальной среды для исполнения гостевой операционной системы. Логическая схема монитора виртуальных машин представлена на рисунке 3.2.

Монитор работает на уровне привилегий ядра управляющей операционной системы и реализует выбранную технологию виртуализации. Монитор включает в себя блок эмуляции процессора и внешних устройств. Он вынужден обращаться к приложению виртуальной машины за доступом к реальным внешним устройствам. Для каждой виртуальной машины запускается своя копия монитора виртуальной машины.

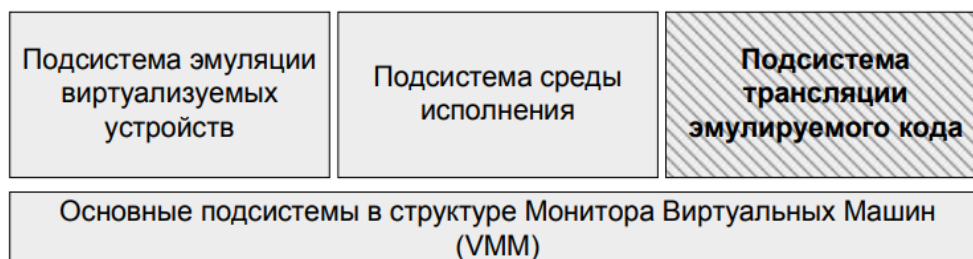


Рисунок 3.2 – Логическая схема монитора виртуальных машин

Монитор может взаимодействовать с приложением двумя способами:

- синхронно при помощи обмена информационными сообщениями через драйвер виртуальных машин;
- асинхронно при помощи разделяемых системных структур и участков памяти.

Монитор работает в изолированном от управляющей операционной системы контексте и поддерживает свои собственные системные таблицы GDT, LDT, IDT и т.д. При переключении контекста между монитором и управляющей операционной системой выполняется операция сохранения одного контекста и загрузка другого. Переключение контекста напоминает процедуру переключения задач операционной системы, но включает в себя дополнительный набор данных. Также, монитор должен выявлять и перенаправлять управляющей операционной системе все прерывания от реальных внешних устройств.

Подход к виртуализации, основанный на применении монитора виртуальных машин, предоставляет целый ряд преимуществ, по сравнению с другими технологиями виртуализации [5].

1. Монитор предоставляют более цельную эмулируемую среду, позволяющую запускать «любое» программное обеспечение, в том числе и системного уровня, такого как операционные системы и т.п.
2. Позволяет запускать любое число операционных систем одновременно на одном оборудовании.
3. Предоставляет независимость гостевой операционной системы от оборудования, на котором он исполняется. Поскольку всё оборудование виртуализуется, вся система может быть динамически перенесена на другое оборудование, без потери работоспособности системы. На пример, для

проведения технического обслуживания система может быть перенесена с одного мэйнфрейма на другой.

4. Добавляет уровень изоляции для оборудования – не позволяя непроверенному коду исполняться на реальном оборудовании.

Рассмотренный вид виртуализации позволяет создавать изолированные, масштабируемые гостевые виртуальные среды, поведение которых близко к поведению реальных серверов. Например, виртуальные машины фирм Parallels и VMware с закрытым кодом и Virtual Box с открытым кодом. Именно аппаратная виртуализация чувствительна к наличию аппаратной поддержки.

Недостатками серверной виртуализации являются существенные накладные расходы, которые зависят от числа эмулируемых устройств и от степени их похожести на физические. Технология плохо масштабируется – нет возможности эффективно и совместно использовать все доступные аппаратные ресурсы, а оперативная память физической машины фактически должна быть жестко разделена для того, чтобы предоставить каждой запущенной виртуальной свою собственную неразделяемую память. Кроме того, существенная доля памяти должна быть зарезервирована для накладных расходов самой системы виртуализации. Фактически на обычных серверах невозможно запустить более 10-15 виртуальных машин. Еще одним недостатком является жесткое разделение дискового пространства между виртуальными машинами. Обычно, файловая система для виртуальной машины расположена внутри файла управляющей ОС (или раздела диска) и уникальна для каждого экземпляра виртуальной машины.

Общая схема паравиртуализации представлена на рисунке 3.3.

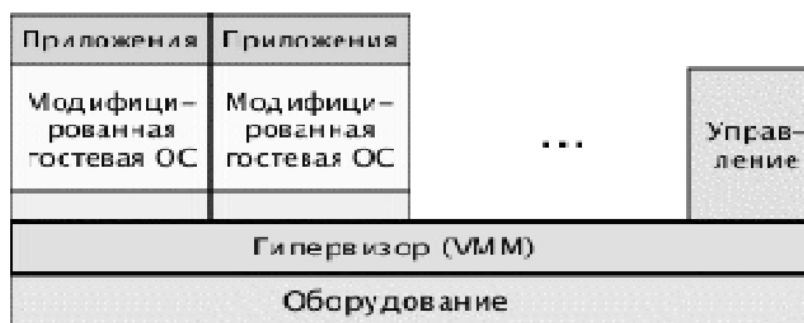


Рисунок 3.3 – Общая схема паравиртуализации

Схема паравиртуализации почти идентична схеме серверной виртуализации, за исключением изменения ядра гостевой операционной системы. Цель изменения заключается в сокращении доли времени выполнения операций в гостевой операционной системе. Это существенный недостаток, так как подобное изменение возможно лишь в случае, если гостевые системы имеют открытые исходные коды, которые можно модифицировать согласно лицензии. Преимуществом паравиртуализации является производительность близкая к производительности реальной, не

виртуализированной системы. Как и при полной виртуализации, одновременно могут поддерживаться многочисленные различные операционные системы.

Общая схема виртуализации уровня операционной системы представлена на рисунке 3.4.

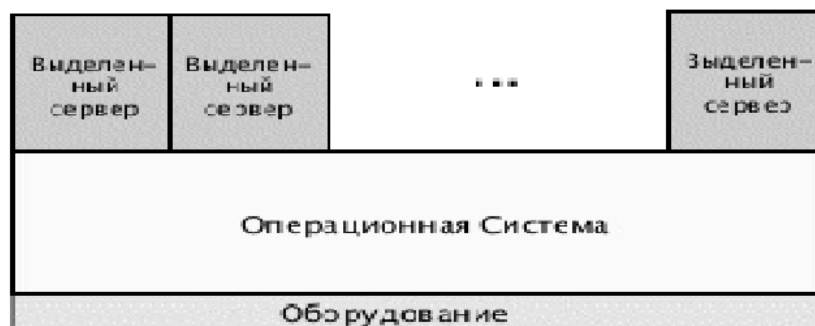


Рисунок 3.4 – Общая схема виртуализации уровня операционной системы

Виртуализация уровня ОС рационализирована для достижения лучшей производительности, управляемости и эффективности гостевой операционной системы. В основе лежит стандартная основная операционная система – это может быть Windows или Linux-based система. Далее идет слой виртуализации с собственной файловой системой и слой абстракции службы ядра, который обеспечивает изоляцию и безопасность ресурсов различных контейнеров.

Контейнер является виртуальной средой исполнения – «высокоуровневой виртуализацией», которая обеспечивает каждой виртуальной машине свое собственное уникальное изолированное окружение: файлы и системные ресурсы, сервисы, системные способы связи с управляющей системой. Слой виртуализации позволяет каждому контейнеру функционировать как самостоятельный сервер. Наконец, в самом контейнере размещается приложение или рабочий процесс.

По сравнению с серверной виртуализацией, виртуализации уровня операционной системы дает большие преимущества: гибкое управление памятью и файловым пространством, низкие затраты на переключение между виртуальными средами, возможность мгновенного получения всех ресурсов компьютера.

Есть еще некоторые существенные отличия рассмотренных видов виртуализации. Например, в контейнере в любой момент возможен прямой административный доступ к файлам из управляющей ОС, тогда как в типовых реализациях виртуальной машины он возможен только по сети, если это сделал администратор соответствующей виртуальной машины. В виртуальной машине имеется возможность сделать «снимок» состояния системы для дальнейшего возвращения к этому состоянию, а в контейнере это делается иначе – на уровне группы процессов, а не всего ядра.

3.2 Аппаратная виртуализация

Под аппаратной виртуализацией понимают аппаратную поддержку, обеспеченную самим процессором. Аппаратная виртуализация – это общее название для двух независимых, но очень похожих технологий Intel и AMD, которые направлены на улучшение производительности процессора для общих задач виртуализации. Некоторые BIOS позволяют включать/выключать поддержку аппаратной виртуализации.

В отличие от программной виртуализации, с помощью данной техники возможно использование изолированных гостевых систем, управляемых гипервизором напрямую. Гостевая система не зависит от архитектуры хостовой платформы и реализации платформы виртуализации. Аппаратная виртуализация обеспечивает производительность, сравнимую с производительностью неvirtуализованной машины, что дает виртуализации возможность практического использования и влечет её широкое распространение. В качестве примера можно привести технологии Intel VT, VT-D и AMD-V. Используя такие технологии, монитор виртуальных машин может передавать управление физическими устройствами непосредственной виртуальной машине. Очевидно, что без аппаратной поддержки, использование виртуализации бессмысленно.

– В Intel VT (Intel Virtualization Technology) реализована виртуализация режима реальной адресации (режим совместимости с 8086). Соответствующая аппаратная виртуализация ввода-вывода – VT-D (кодовое название – Vanderpool). Часто обозначается аббревиатурой VMX (Virtual Machine eXtension).

– AMD-V часто обозначается аббревиатурой SVM (Secure Virtual Machines). Кодовое название – Pacifica. Соответствующая технология виртуализации ввода-вывода – IOMMU. AMD-V проще и эффективнее, чем Intel VT. Поддержка AMD-V появилась в Xen 3.3.

Первые реализации аппаратной поддержки виртуализации были сосредоточены на перехвате инструкций, то есть нацеливались, прежде всего, на виртуализацию процессора. Однако со временем производители добавили средства работы с памятью и с устройствами, предоставив возможность создавать гипервизоры, способные контролировать практически все активности системы. Любая аппаратная виртуализация должна обеспечивать следующие функции для стабильной работы:

1. Виртуализация процессоров. Все технологии и инструкции, которые доступны физическому процессору, становятся доступны и виртуальным.

2. Графические представления. Обеспечивается изолированный доступ гостевых ОС к графическому адаптеру.

3. Ввод/вывод (I/O). Гипервизор разделяет доступ к устройствам ввода/вывода и другой периферией между виртуальными системами без конфликтов и зависаний.

Идея аппаратной виртуализации не нова: впервые она была воплощена в 386-х процессорах и носила название V86 mode. Этот режим работы 8086-го процессора позволял запускать параллельно несколько DOS-приложений. Теперь аппаратная виртуализация позволяет запускать несколько независимых виртуальных машин в соответствующих разделах аппаратного пространства компьютера [6]. Аппаратная виртуализация является логическим продолжением эволюции уровней абстрагирования программных платформ – от многозадачности до уровня виртуализации.

Многозадачность является первым уровнем абстракции приложений. Каждое приложение разделяет ресурсы физического процессора в режиме разделения исполнения кода по времени.

Технология гиперпоточности в широком смысле также представляет собой аппаратную технологию виртуализации, поскольку при ее использовании в рамках одного физического процессора происходит симуляция двух виртуальных процессоров в рамках одного физического с помощью техники Symmetric Multi Processing (SMP) (см. рисунок 3.5).

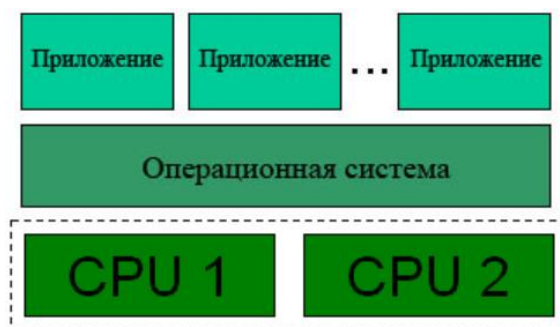


Рисунок 3.5 – Общая схема гиперпоточности

Виртуализация представляет собой эмуляцию нескольких виртуальных процессоров для каждой из гостевых операционных систем. При этом технология виртуального SMP позволяет представлять несколько виртуальных процессоров в гостевой ОС при наличии технологии гиперпоточности или нескольких ядер в физическом процессоре (см. рисунок 3.6).

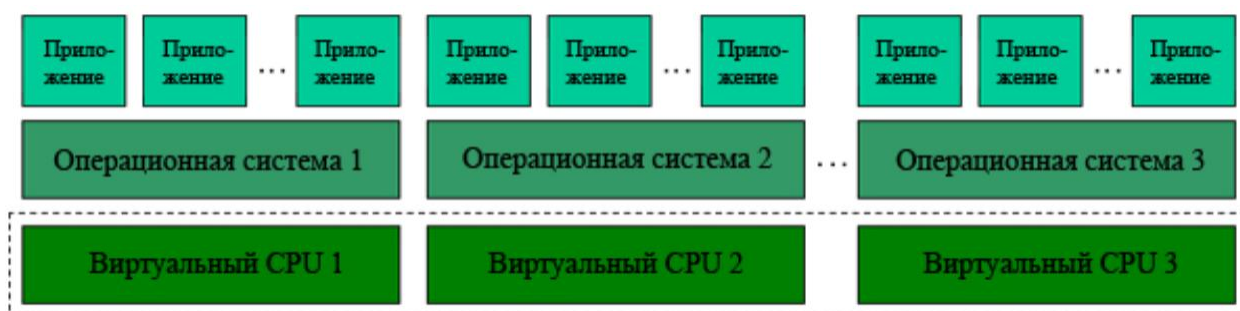


Рисунок 3.6 – Общая схема аппаратной виртуализации

Программная виртуализация в данный момент превалирует над аппаратной на рынке технологий виртуализации ввиду того, что долгое время производители процессоров не могли должным образом реализовать поддержку виртуализации. Процесс внедрения новой технологии в процессоры требовал серьезного изменения их архитектуры, введения дополнительных инструкций и режимов работы процессоров. Это рождало проблемы обеспечения совместимости и стабильности работы, которые были полностью решены в 2005-2006 годах в новых моделях процессоров. Несмотря на то, что программные платформы весьма продвинулись в отношении быстродействия и предоставления средств управления виртуальными машинами, технология аппаратной виртуализации имеет некоторые неоспоримые преимущества перед программной:

1. Упрощение разработки платформ виртуализации за счет предоставления аппаратных интерфейсов управления и поддержки виртуальных гостевых систем. Это способствует появлению и развитию новых платформ виртуализации и средств управления, в связи с уменьшением трудоемкости и времени их разработки.

2. Возможность увеличения быстродействия платформ виртуализации. Поскольку управление виртуальными гостевыми системами производится с помощью небольшого промежуточного слоя программного обеспечения (гипервизора) напрямую, в перспективе ожидается увеличение быстродействия платформ виртуализации на основе аппаратных техник.

3. Возможность независимого запуска нескольких виртуальных платформ с возможностью переключения между ними на аппаратном уровне. Несколько виртуальных машин могут работать независимо, каждая в своем пространстве аппаратных ресурсов, что позволит устранить потери быстродействия на поддержание хостовой платформы, а также увеличить защищенность виртуальных машин за счет их полной изоляции.

4. Абстрагирование гостевой системы от архитектуры хостовой платформы и реализации платформы виртуализации. С помощью технологий аппаратной виртуализации возможен запуск 64-битных гостевых систем из 32-битных хостовых системах, с запущенными в них 32-битными средами виртуализации.

Необходимость поддержки аппаратной виртуализации заставила производителей процессоров несколько изменить их архитектуру за счет введения дополнительных инструкций для предоставления прямого доступа к ресурсам процессора из гостевых систем. Этот набор дополнительных инструкций носит название Virtual Machine Extensions (VMX). VMX предоставляет следующие инструкции: VMPTLDR, VMPTLST, VMCLEAR, VMREAD, VMWRITE, VMCALL, VMLAUNCH, VMRESUME, VMXON и VMXOFF [6].

Процессор с поддержкой виртуализации может работать в двух режимах: root operation и non-root operation. В режиме root operation работает специальное программное обеспечение, являющееся «легковесной»

прослойкой между гостевыми операционными системами и оборудованием — монитор виртуальных машин (Virtual Machine Monitor, VMM), носящий также название гипервизор. Интерфейс монитора виртуальных машин полностью (либо частично) повторяет интерфейс виртуализуемой системы, благодаря чему, программное обеспечение не может определить факт присутствия гипервизора. В режиме non-root operation работает виртуализированная гостевая система.

Чтобы перевести процессор в режим виртуализации, платформа виртуализации должна вызвать инструкцию VMXON и передать управление гипервизору, который запускает виртуальную гостевую систему инструкцией VMLAUNCH и VMRESUME (точки входа в виртуальную машину). Virtual Machine Monitor может выйти из режима виртуализации процессора, вызвав инструкцию VMXOFF (см. рисунок 3.7).

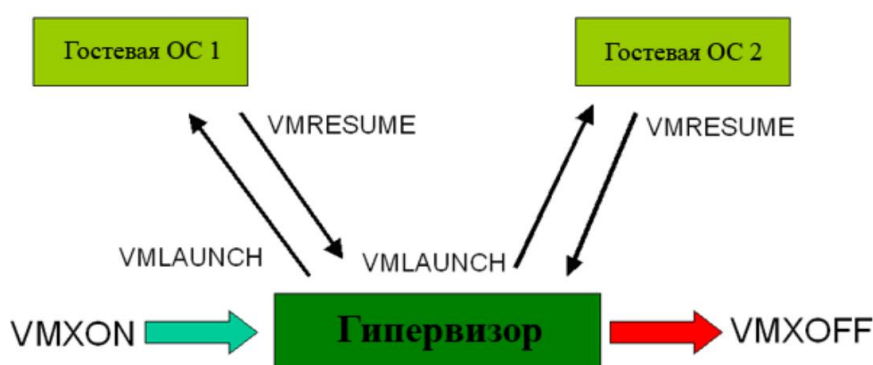


Рисунок 3.7 – Процедура запуска виртуальных машин при аппаратной виртуализации

Каждая из гостевых операционных систем запускается и работает независимо от других и является изолированной с точки зрения аппаратных ресурсов и безопасности.

Стоит также отметить, что аппаратная виртуализация потенциально несет в себе не только положительные моменты. Возможность управления гостевыми системами посредством гипервизора и простота написания платформы виртуализации с использованием аппаратных техник дают возможность разрабатывать вредоносное программное обеспечение, которое после получения контроля на хостовой операционной системой, виртуализует ее и осуществляет все действия за ее пределами.

4 СОВРЕМЕННЫЕ СРЕДЫ ВИРТУАЛИЗАЦИИ

4.1 Среда виртуализации VirtualBox

Oracle Virtual Box – бесплатный программный продукт с открытым исходным кодом, который распространяется под Универсальной Общественной Лицензией GNU, 2-ой версии (GPLv2). Последняя версия Virtual Box, которая в настоящее время доступна, – это Virtual Box 5.2.22.

Oracle Virtual Box является гипервизором 2-ого типа и должен быть установлен на операционной системе хоста в качестве программного приложения. Продукт поддерживает аппаратную виртуализацию, а также виртуализацию программного обеспечения. Это означает, что есть возможность запускать виртуальные машины с Virtual Box даже на старом оборудовании, на котором центральный процессор не поддерживает аппаратные функции виртуализации.

На виртуальных машинах, работающих на Virtual Box, могут быть установлены следующие гостевые операционные системы: Linux, Windows, Solaris, FreeBSD, macOS.

Графический интерфейс Oracle Virtual Box представлен на рисунке 4.1.

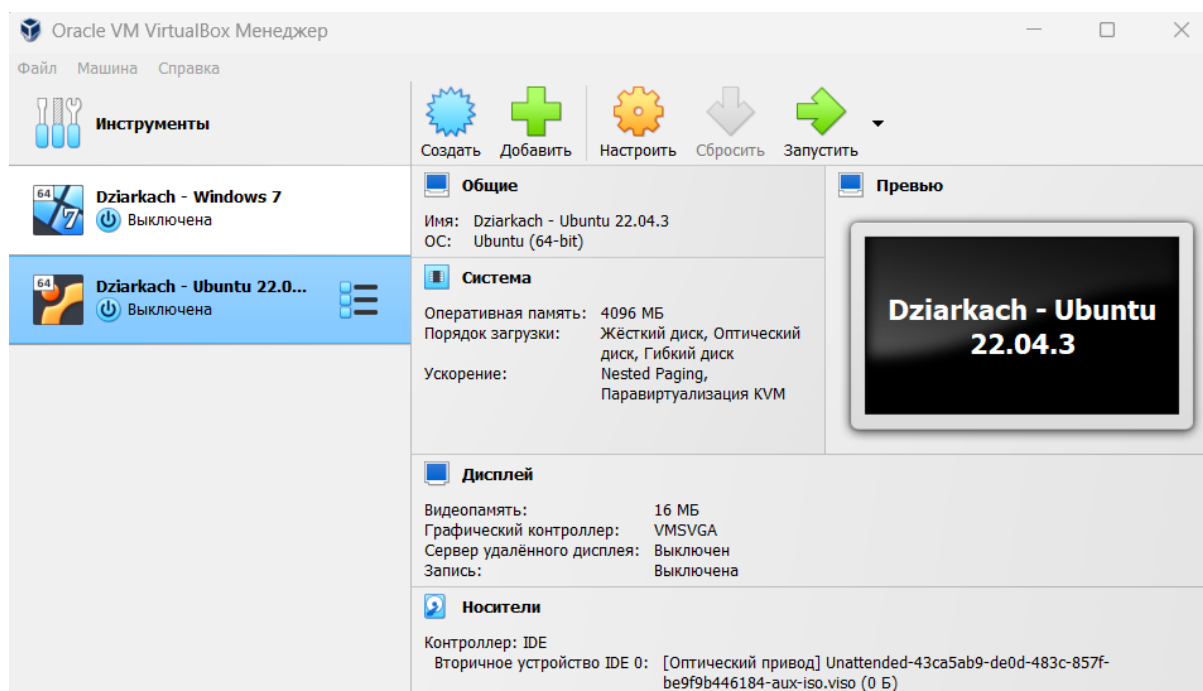


Рисунок 4.1 – Графический интерфейс среды виртуализации VirtualBox

Требования к компьютеру при использовании Oracle Virtual Box зависят от цели использования виртуальной машины. Для запуска устаревших ОС подойдет ПК или ноутбук с двухъядерным процессором и 2 Гб оперативной памяти.

Если же требуется запуск 64-разрядной системы для полноценной работы с ресурсоемкими приложениями, то потребуются компьютер с большим объемом ОЗУ и процессором с тактовой частотой не менее 2.9 ГГц. Чем мощнее реальное оборудование, тем лучше будет работать виртуальная машина без влияния на хостевую ОС.

Для создания виртуальной машины Oracle Virtual Box [7] нужно убедиться в том, что компьютер обладает достаточной мощностью и скачать образ соответствующей гостевой системы. Затем можно запустить VirtualBox и нажать на кнопку создания новой машины. Следует дать ей понятное имя и выбрать правильный тип.

Далее указывается объем оперативной памяти для виртуальной машины. Он должен быть не менее 900 Мбайт и не более половины от размера реального ОЗУ, установленного внутри ПК. Для пробного запуска допустимо установить значение в 1 гигабайт.

Затем нужно создать виртуальный носитель, на который будет установлена гостевая операционная система. Для этого выбирается физический жесткий диск, на котором присутствует свободное место, достаточное для гостевой ОС. В открывшемся окне формат VDI оставляется без изменений. Вслед за нажатием кнопки «далее» выбирается тип виртуального носителя. Если создается полноценная ВМ для постоянной работы, используется динамический диск. Его размер будет увеличиваться при необходимости. Если создается виртуальной машины для проверки или непостоянной работы, следует создать фиксированный виртуальный диск. Он обладает большим быстродействием. На следующем шаге нужно задать объем носителя.

Создание диска займет некоторое время. Эту операцию можно отменить, чтобы, например, создать носитель в другом месте. После этого появится главное окно диспетчера VMBox, в котором будет отображена созданная виртуальная машина.

4.2 Среда виртуализации Hyper-V

Hyper-V – система аппаратной виртуализации для x64-систем на основе гипервизора 1-го типа, разработанная компанией Microsoft [8]. Она позволяет создавать и управлять виртуальными машинами на компьютере под управлением Windows. Бета-версия Hyper-V была включена в x64-версии Windows Server 2008, а законченная версия была выпущена 26 июня 2008. Ранее была известна как виртуализация Windows Server.

Hyper-V предоставляет возможность выполнять виртуализацию оборудования. Это означает, что каждая виртуальная машина работает на виртуальном оборудовании. Hyper-V позволяет создавать виртуальные жесткие диски, виртуальные коммутаторы и ряд других виртуальных устройств, каждое из которых можно добавить в виртуальную машину.

Технология Hyper-V доступна в 64-разрядной версии Windows. Для Hyper-V требуется функция преобразования адресов второго уровня (SLAT). Она есть в текущем поколении 64-разрядных процессоров Intel и AMD.

На узле, имеющем 4 ГБ оперативной памяти, можно запустить тричетыре базовые виртуальные машины, однако для большего числа виртуальных машин потребуется больше ресурсов. Кроме того, можно создать мощные виртуальные машины с 32 процессорами и 512 ГБ ОЗУ в зависимости от оборудования.

Графический интерфейс диспетчера Hyper-V представлен на рисунке 4.2.

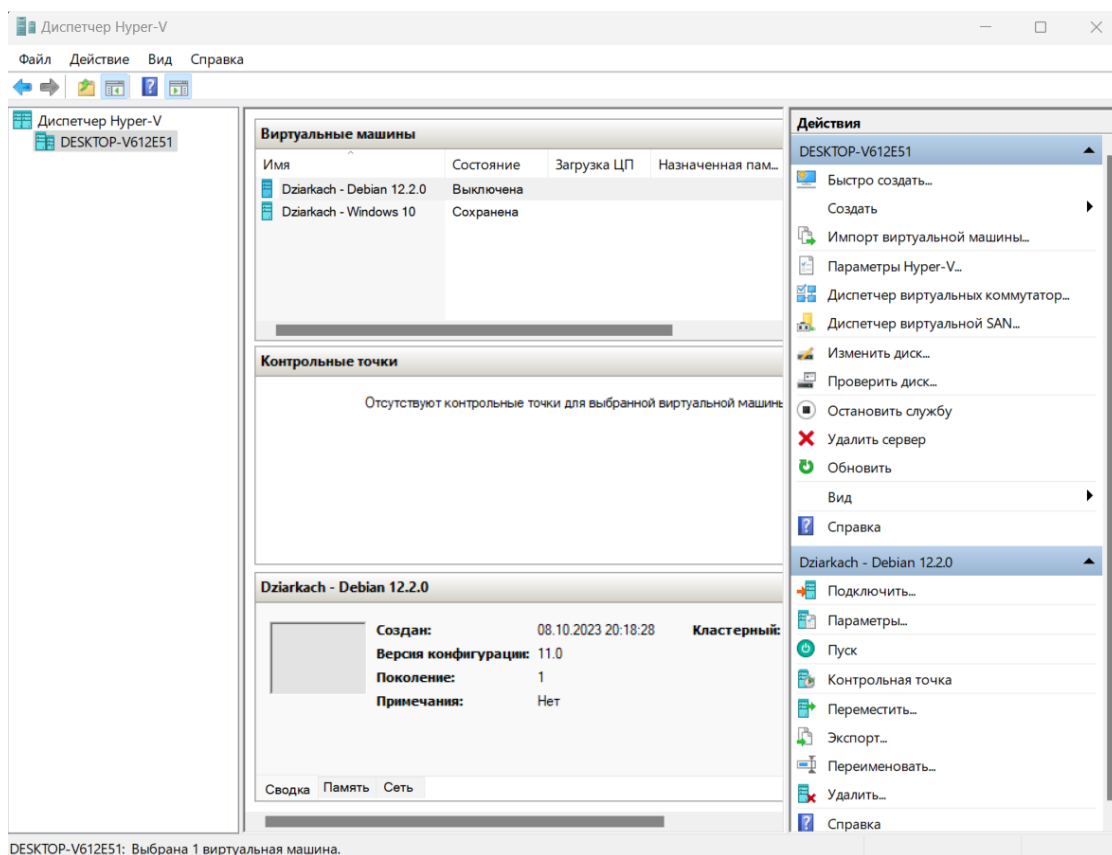


Рисунок 4.2 – Графический интерфейс диспетчера Hyper-V

Архитектура ввода-вывода для Hyper-V состоит из поставщиков служб виртуализации (VSPs) в корневой секции и клиентах служб виртуализации (VSCs) в дочерней секции (см. рисунок 4.3). Каждая служба предоставляется как устройство через шину виртуальной машины, которая выступает в качестве шины ввода-вывода и обеспечивает высокопроизводительное взаимодействие между виртуальными машинами, использующими такие механизмы, как общая память. Диспетчер самонастраивающийся гостевой операционной системы перечисляет эти устройства, включая шину виртуальной машины, и загружает соответствующие драйверы устройств, клиенты виртуальной службы. Службы, отличные от операций ввода-вывода, также предоставляются с помощью этой архитектуры.

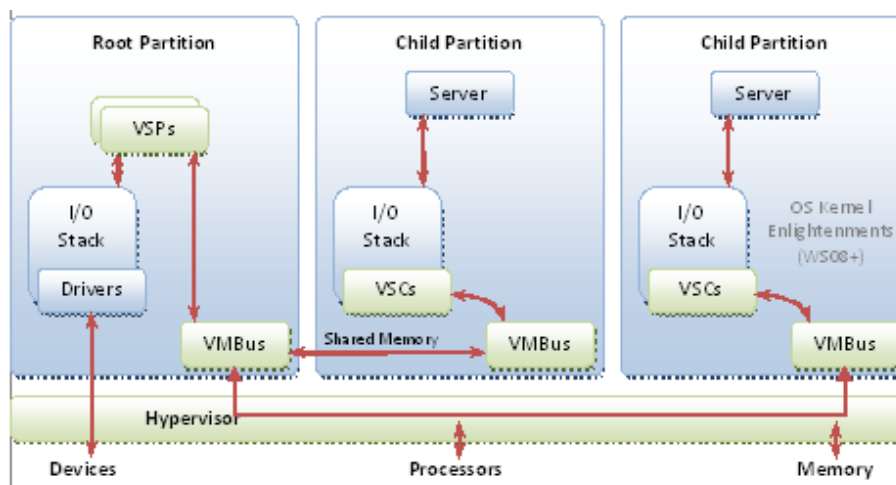


Рисунок 4.3 – Архитектура ввода-вывода для Hyper-V

Hyper-V доступен только в 64-разрядной Windows не ниже профессионального издания. Является единственной на данный момент технологией виртуализации Microsoft, поддерживающей 64-разрядные гостевые системы. В клиентских системах поддерживает режимы сна и гибернации, в отличие от серверного гипервизора.

Для работы Hyper-V у процессора должна быть в наличии и включена в BIOS аппаратная виртуализация и трансляция адресов второго уровня (SLAT).

Hyper-V представляет собой компонент операционной системы, который изначально отключен. Опытные пользователи могут включить его одной командой PowerShell: `Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All`.

Также компонент Hyper-V можно включить через графический интерфейс. Для этого необходимо нажать комбинацию клавиш Win + R и ввести `OptionalFeatures`, после чего нажать Enter. В открывшемся окне необходимо установить флаг рядом с Hyper-V. Так или иначе, компонент станет доступен после перезагрузки системы.

Мастер создания виртуальной машины предельно прост. Стандартное расположение для виртуальных машин – папка `ProgramData`, но его можно изменить. Если уже есть виртуальный диск в формате VHD, можно подключить его.

4.3 Среда виртуализации VMware Workstation

VMware Workstation – программное обеспечение виртуализации, предназначенное для компьютеров x86-64 операционных систем Microsoft Windows и Linux. Является гипервизором 2-ого типа и поддерживает аппаратную виртуализацию.

Позволяет пользователю установить одну или более виртуальных машин на один физический компьютер и запускать их параллельно с ним.

Каждая виртуальная машина может выполнять свою операционную систему, включая Microsoft Windows, Linux, BSD, и MS-DOS. VMware Workstation разработана и продается компанией VMware, подразделением EMC Corporation.

VMware Workstation поддерживает мосты с сетевым адаптером реального компьютера, а также создание общих папок с виртуальной машиной. Программа может монтировать реальные CD или DVD диски или ISO образы в виртуальные оптические приводы, при этом виртуальная машина будет считать, что приводы настоящие. Виртуальные жесткие диски хранятся в файлах .vmdk.

VMware Workstation в любой момент может сохранить текущее состояние виртуальной машины (снимок). Данные снимки позже могут быть восстановлены, что возвращает виртуальную машину в сохраненное состояние.

VMware Workstation включает в себя возможность объединять несколько виртуальных машин в группу, которую можно включать, выключать, приостанавливать или возобновлять как единый объект, что является полезным для тестирования технологий клиент-сервер.

Продукты VMware поддерживают установку следующих гостевых операционных систем: Linux, Windows, Solaris, FreeBSD, macOS. Стоит иметь в виду, что macOS может быть официально установлена только в качестве гостевой операционной системы на VMware Fusion/Fusion Pro.

Графический интерфейс VMware Workstation представлен на рисунке 4.4.

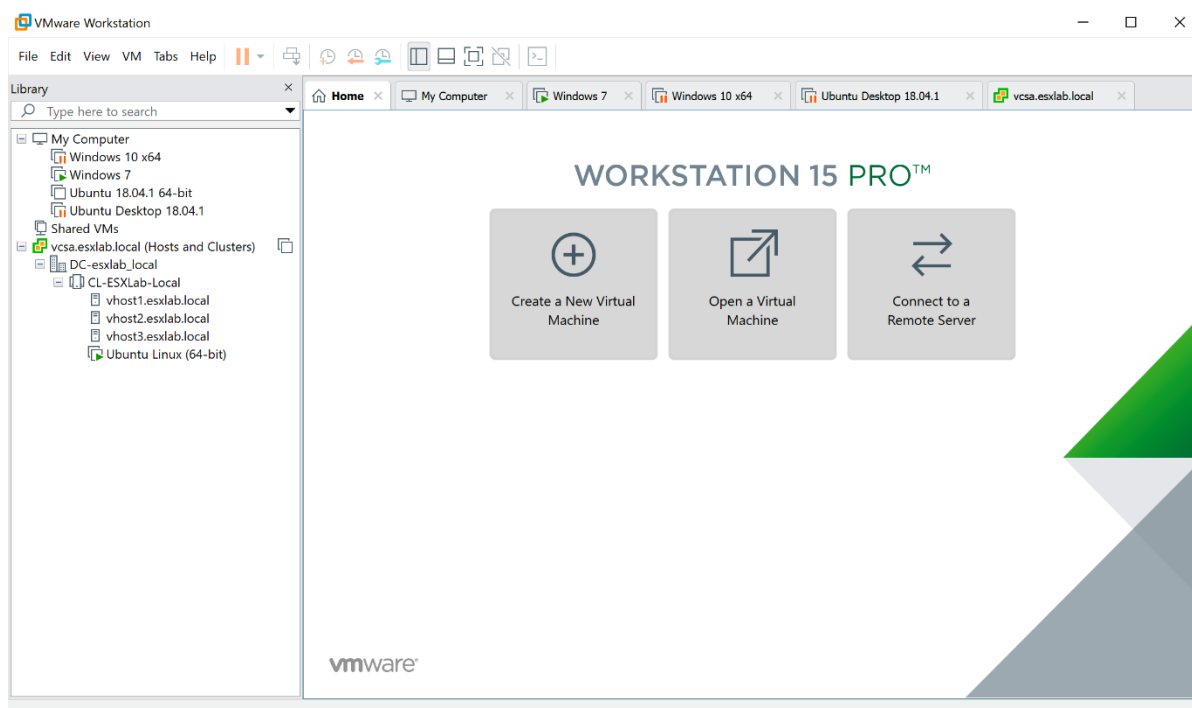


Рисунок 4.4 – Графический интерфейс среды виртуализации VMware Workstation

При работе VMware Workstation создаются полностью изолированные друг от друга виртуальные машины, которые напрямую используют ресурсы физической рабочей станции, где они установлены (см. рисунок 4.5). Таким образом, каждая виртуальная машина имеет свою ОС и набор софта, выделенный ресурс ЦП и оперативной памяти, дисковое пространство и устройства ввода-вывода, имитируя полноценный x86 или x64 компьютер, который работает полностью обособленно от других виртуальных машин. Благодаря доступу к гостевой ОС виртуальные машины получают поддержку практически любого программного и аппаратного обеспечения, что позволяет запускать даже специализированные программы без проблем и без ущерба производительности.

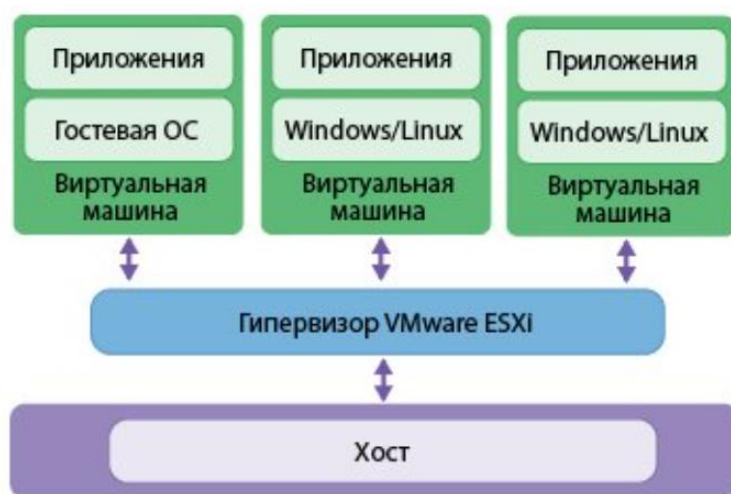


Рисунок 4.5 – Общая структура виртуализации VMware Workstation

Для создания виртуальной машины в VMware Workstation необходимо на домашней странице программы выбрать функцию создания новой виртуальной машины. Где будет предложено два типа создания ВМ:

1. Обычный – упрощённый вариант с большей частью заданных самой программой параметров;
2. Выборочный – пошаговый мастер с возможностью выбора многих значимых параметров.

Все шаги по созданию виртуальной машины интуитивно понятны. В процессе настройки необходимо выбрать образ устанавливаемой гостевой системы, а также число ядер процессора и выделяемую память. Дополнительно можно указать и другие параметры, такие как: тип прошивки BIOS, сетевое подключение, тип контроллеров ввода-вывода и д.р.

ЗАКЛЮЧЕНИЕ

В 2006 году технологии виртуализации переживали заметный взлет: новые участники вступили на рынок, вышло множество платформ и инструментов управления, и было заключено значительное количество партнерских соглашений и альянсов. Все эти факторы свидетельствовали о том, что виртуализация обречена стать ключевой технологией в будущем. Рынок средств виртуализации находился на заключительном этапе формирования, множество производителей аппаратного обеспечения заявили о поддержке технологий виртуализации.

Сегодня виртуализация стала ещё ближе к конечным пользователям: интерфейсы для работы с виртуальными машинами стали более интуитивными, появились неофициальные соглашения об использовании различных средств и техник, а миграция между виртуальными платформами упростилась. Виртуализация неизбежно займет своё место в списке обязательных технологий при проектировании ИТ-инфраструктуры предприятий. Даже обычные пользователи найдут применение виртуальным машинам, особенно с ростом производительности настольных компьютеров, позволяя поддерживать несколько пользовательских окружений на одной машине.

Производители аппаратного обеспечения также активно развиваются: ожидается появление аппаратных систем, полностью интегрирующих виртуализацию и предоставляющих удобные интерфейсы для программного обеспечения. Это содействует разработке надежных и эффективных платформ виртуализации. Возможно, в будущем операционные системы будут автоматически виртуализироваться, а специализированное низкоуровневое программное обеспечение, используя аппаратные функции, обеспечит переключение между запущенными операционными системами без потери производительности.

Суть, заложенная в технологиях виртуализации, открывает широкие перспективы для их применения. Эти изменения, в конечном итоге, направлены на улучшение пользовательского опыта и упрощение использования привычных технологий. Каким образом это сможет существенно повлиять на экономию ресурсов, покажет время.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1]. Что такое виртуализация? [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/what-is/virtualization/> – Дата доступа: 16.10.2023.
- [2]. Типы гипервизоров – сравнительный анализ [Электронный ресурс]. – Режим доступа: <https://www.cloud4u.ru/blog/typy-gipervizorov-sravnitelnyy-analiz/> – Дата доступа: 18.10.2023.
- [3]. Подробно о виртуализации: типы, преимущества и решения [Электронный ресурс]. – Режим доступа: <https://onbiz.biz/ru/about-virtualization/> – Дата доступа: 21.10.2023.
- [4]. Уровни виртуализации [Электронный ресурс]. – Режим доступа: <https://mega-obzor.ru/urovni-virtualizacii.html> – Дата доступа: 22.10.2023.
- [5]. Виртуализация в высокопроизводительных вычислительных системах [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/virtualizatsiya-v-vysokoproizvoditelnyh-vychislitelnyh-sistemah> – Дата доступа: 28.10.2023.
- [6]. Технологии аппаратной виртуализации [Электронный ресурс]. – Режим доступа: <https://www.ixbt.com/cm/virtualization-h.shtml> – Дата доступа: 31.10.2023.
- [7]. Download VirtualBox [Электронный ресурс]. – Режим доступа: <https://www.virtualbox.org/wiki/Downloads> – Дата доступа: 03.11.2023.
- [8]. Обзор технологии Hyper-V [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/windows-server/virtualization/hyper-v/hyper-v-technology-overview> – Дата доступа: 04.11.2023.
- [9]. VMware Workstation Download Center [Электронный ресурс]. – Режим доступа: https://customerconnect.vmware.com/en/downloads/info/slug/desktop_end_user_computing/vmware_workstation_pro/15_0 – Дата доступа: 05.11.2023.