## Предпосылки и история возникновения облачных вычислений

1. Развитие аппаратных средств:
2. Развитие технологий виртуализации
3. Технологическое развитие
4. Сервисный подход к управлению информационными технологиями
5. Сервисный подход к созданию информационных систем

- Архитектура SOA - это метод разработки программного обеспечения, который использует программные компоненты, называемые сервисами, для создания бизнес-приложений. Каждый сервис предоставляет бизнес-возможности, и сервисы также могут взаимодействовать друг с другом на разных платформах и языках

- Веб-сервисы RestFull – Representational State Transfer – передача состояния представления. Архитектурный стиль взаимодействия компонентов распределенного приложения. Автор Roy Fielding, диссертация «Архитектурные стили и дизайн сетевых программных архитектур» 2000 год

Первые идеи об использовании вычислений как публичной услуги были предложены в 1960-х известным ученым в области ИТ, профессором MIT и изобретателем языка Lisp Джоном Маккарти. Реализация первого реального проекта приписывается компании Salesforce.com, основанной в 1999. Тогда и появилось первое предложение нового вида B2B-прродукта – «Программное обеспечение как сервис» SaaS – Software as a Service

Первое бизнес-решение – это Amazon Web Services (AWS), запущенное в 2005 году компанией Amazon.com. Далее – Google, которая ввела с 2006 года «Google Apps» как систему B2B. Свое предложение анонсировала компания Microsoft, презентовав ее на конференции PDC 2008 под названием «Azure Services Platform», сегодня «Microsoft Azure»

## Обзор моделей (парадигм) распределенных вычислений.

Распределенные вычисления – это модель, в которой ресурсы системы распределяются между несколькими компьютерами для повышения эффективности и производительности

Три важные характеристики распределенных систем:

1. Параллелизм выполнения компонентов
2. Отсутствие глобальной синхронизации часов
3. Возможность независимого отказа отдельных компонентов РИС

Модели распределенных вычислений:

1. P2P одноранговые вычисления – сетевая вычислительная модель для приложений, в которой компьютеры совместно используют ресурсы и сервисы посредством прямого электронного изменения
2. Кластерные вычисления – каждый узел настроен на выполнение один и тех же задач, контролируемых и планируемых ПО
3. Вычисления по запросу – Utility Computing – модель предоставления услуг, в которой поставщик услуг предоставляет клиенту вычислительные ресурсы и управление инфраструктурой в соответствии с потребностями
4. Грид-вычисления
5. Облачные вычисления
6. Туманные вычисления – распределенная вычислительная инфраструктура, в которой некоторые сервисы приложений обрабатываются на интеллектуальных устройствах, а другие – в удаленном центре обработки данных – в облаке
7. Вычисления на гетерогенных системах (Jungle computing) – сочетание разнородных, иерархических и распределенных вычислительных ресурсов

## Особенности облачных вычислений.

Облачные вычисления – вычислительная парадигма, в которой вычисления переносятся с ПК или отдельного сервера приложений на облако компьютеров. Потребители облака должны заботиться только о своих вычислительных потребностях, поскольку все основные детали облачной инфраструктуры скрыты от потребителя

Облако – совокупность ресурсов, которая включает хранилище, серверы, базы данных, сети, программное обеспечение

Особенности:

1. Центральную роль на рынке облачных услуг играют поставщики и потребители этих услуг
2. ИТ-услуги используются в качестве товара на открытом рынке без технических и юридических барьеров
3. Облачные услуги привлекательны для разработчиков ПО без инфраструктуры и которые не могут позволить себе расширение своей
4. Надежная структура, доступна в любое время. Безопасность и надежность продолжает развиваться
5. Вычислительные службы высоконадежные, масштабируемые, доступные вне зависимости от местонахождения пользователя

## Основные характеристики (признаки) облачных вычислений.

Облачные вычисления – модель предоставления по запросу доступа к разделяемому пулу конфигурируемых компьютерных ресурсов (виртуальных машин, систем хранения данных, приложений и сервисов), которые могут быть быстро подготовлены и предоставлены пользователю с минимальными усилиями со стороны поставщика облачных услуг

Характеристики:

1. Самообслуживание по запросу
2. Широкополосный доступ к сети и мобильность
3. Объединение ресурсов в пулы однотипных ресурсов
4. Быстрая эластичность
5. Измеряемое обслуживание и прозрачность. Плата за использование

Участники процесса:

1. Облачный потребитель – лицо или организация, использующая услуги облачного провайдера
2. Облачный провайдер – лицо, организация или сущность, отвечает за доступность облачной услуги для облачного потребителя
3. Облачный аудитор – участник, который может выполняет независимую оценку (assessment) облачных услуг, обслуживания информационных систем, производительности и безопасности реализации облака.
4. Облачный брокер – сущность, управляющая использованием, производительностью и предоставлением облачных услуг, устанавливает отношения Провайдер-Потребитель
5. Облачный оператор связи – посредник, предоставляющий услуги подключения и транспорт доставки облачных услуг от Провайдера к Потребителю

## Модели развертывания облаков: частное облако, публичное облако, гибридное облако, общественное облако.

Приватное облако (частное облако) – инфраструктура, предназначенная для использования облачных вычислений в масштабе одной организации

Публичное облако – инфраструктура, предназначенная для свободного использования облачных вычислений широкой публикой

Гибридное облако – комбинация различных облачных инфраструктур (частных + публичных + общественных), остающихся уникальными объектами, но связанных между собой стандартизированными или частными технологиями, которые обеспечивают возможность обмена данными и приложениями

Облачная инфраструктура является композицией нескольких облаков, остающихся уникальными, но объединенных вместе стандартизированными или частными технологиями, обеспечивающими портируемость данных и приложений между облаками

Облако сообщества – облачная инфраструктура, которая предназначена для исключительного использования облачных вычислений определенным сообществом потребителем от организаций, которые решают общие проблемы

## Модель предоставления услуг облачных вычислений: Software as a Service (SaaS), достоинства и недостатки.

Software as a Service – программное обеспечение как услуга. Основная деятельность (активности) – использует приложения/сервисы для автоматизации бизнес-процессов. Примеры потребителей – бизнес-пользователи, администраторы приложений

SaaS – модель развертывания приложения, которая подразумевает предоставление приложения конечному пользователю как услуги по требованию.

Преимущества:

1. Скорость получения услуги – легкий доступ
2. Провайдеры SaaS отвечают за обновления и поддержку приложений, включая резервное копирование данных, обеспечение безопасности и устранение ошибок
3. Снижение первоначальных затрат, возможность сокращения затрат на весь срок службы используемого приложения
4. Передача некоторых или всех вспомогательных обязательств
5. Устранение лицензионного риска
6. Устранение совместимости версий
7. Сокращение оборудования используемого для работы приложения

Проблемы:

1. Распространение модели безопасности на поставщика
2. Управление биллингом - процессом выставления счетов и сбора платежей за предоставленные товары или услуги
3. Интегрированная поддержка конечных пользователей
4. Масштабируемость

## Модель предоставления услуг облачных вычислений: Platform as a Service (PaaS), достоинства и недостатки.

Platform as a Service – платформа как сервис. В PaaS уровень ПО или среды разработки инкапсулируется и предлагается как услуга, на основе которой могут быть построены другие услуги более высокого уровня.

Преимущества:

1. Оплата по мере использования для среды разработки, тестирования и производства
2. Позволяет разработчикам сосредоточиться на коде приложения
3. Мгновенное устранение зависимости от оборудования и емкости глобальной платформы
4. Присущая масштабируемость, гибкость - разработчики могут легко увеличивать или уменьшать вычислительные ресурсы, объем хранилища и другие компоненты платформы в соответствии с требованиями приложений
5. Упрощенная модель развертывания
6. Ускорение разработки приложений
7. Управление жизненным циклом приложений – PaaS предоставляет инструменты для управления жизненным циклом приложений, включая развертывание, мониторинг, масштабирование, обновление и миграцию

Недостатки:

1. Управление
2. Привязка к поставщику услуг
3. Расширение модели безопасности на провайдера + безопасность данных
4. Связь
5. Опора на стороннюю организацию SLA - это договор между поставщиком услуг и B2B-клиентом

## Модель предоставления услуг облачных вычислений: Infrastructure as a Service (IaaS), достоинства и недостатки.

Infrastructure as a Service – инфраструктура как сервис. IaaS предоставляет базовые вычислительные возможности и хранилище в качестве стандартизированных услуг по сети. Серверы, система хранения, пространство центра обработки данных, сетевое оборудование, объединенные в пул, доступны пользователям

Преимущества:

- Для систем, управляемых на основе SLA, должны означать меньшее количество нарушений договора и обслуживание

- Более высокая рентабельность активов за счет более высокого использования

- Полный контроль и настраиваемость

- Снижение затрат за счет меньшего количества оборудования

- Меньше занимаемой площади за счет меньшего занимаемого оборудования

- Более высокий уровень автоматизации благодаря меньшему количеству администраторов

- Низкое энергопотребление

- Возможность согласовать потребление со спросом

- Доступность и отказоустойчивость

Недостатки:

- Техническая сложность – нужна экспертиза

- Переносимость приложений

- Ответственность за безопасность

- Зависимость от провайдера IaaS

- Ограничения выбора аппаратного обеспечения

- Зрелость инструментов управления системами

- Интеграция через расширение облачных границ моделей внутренний безопасности

## Проблемы облачных вычислений. Этические вопросы применения облачных систем

Проблемы облачных вычислений:

1. Реализация соглашения об уровне обслуживания – SLA
2. Управление данными в облаке. Большие данные
3. Безопасность

- Ботнет – сеть из зараженных зомби-компьютеров (ботов), чаще всего используются хакерами для DDoS-атак и массовой спам-рассылки. DDoS (Distributed Denial of Service) – тип атаки, при которой злоумышленники пытаются перегрузить целевой сервер или сеть, создавая огромное количество запросов с множества компьютеров или ботнета

- Фишинг – мошенничество в Интернете с целью получения незаконного доступа к конфиденциальным данным пользователей

- Потеря данных из-за перехвата, модификации или блокирования данных при передаче по сети, хранении или обработке в облаке

1. Совместимость (интероперабельность) облачных сервисов
2. Управление энегоресурсами – компромисс энергосбережения и производительности
3. Мультарендность ресурсов
4. Консолидация серверов – использовать максимальное число ресурсов при минимальном потреблении энергии
5. Надежность и доступность услуг. Ошибки в ПО
6. Наличие общих облачных стандартов – нужен целостный подход для переносимости

Этические вопросы:

1. Контроль за работой всей или части системы передается сторонним службам
2. Данные хранятся на нескольких сайтах, управляемых несколькими организациями
3. Несколько сервисов разных поставщиков взаимодействуют в сети

## Роль стандартизации API облачных сервисов. Организации, занимающиеся стандартизацией облачных сервисов. Стандарты облачных вычислений.

Роль стандартизации – создание единых наборов правил и спецификаций взаимодействия различных облачных провайдеров и клиентских приложений

Стандарты облачных вычислений:

1. Открытый формат виртуализации OVF
2. P2301 – руководство по профилям облачной переносимости
3. P2302 – стандарт межоблачного взаимодействия и федерации SIIF
4. Открытый интерфейс облачных вычислений OCCI – разработка API для задач управления облаком

Открытые стандарты - организации:

1. OpenStack Foundation
2. Open Grid Forum
3. Open Group

Отраслевые организации стандартизации:

1. Distributed Management Task Force (DMTF)
2. Storage Network Institute Association (SNIA)
3. Рабочая группа по управлению облаком в DMTF
4. Техническая группа по облачному хранению данных в SNIA

Организации поддержки пользователей и SLA:

1. TM Forum
2. Cloud Service Customer Council

Организации стандартизации ИТ:

1. National Institute of Standards and Technology (NIST) – опубликовал определение облачных вычислений
2. OASIS (Organization for the Advancementof Structured Information Standards) – черновые варианты облачных вычислений

## Препятствия для разработки облачных приложений и сервисов.

1. Доступность сервиса – что происходит, когда поставщик услуг не может выполнить поставку?
2. Привязка к поставщику – как только клиент привязан к одному поставщику облачных услуг, трудно перейти к другому поставщику. Усилия по стандартизации NIST пытаются решить это
3. Конфиденциальность данных и возможность аудита – серьёзная проблема
4. Непредсказуемость производительности – последствия совместного использования ресурсов
5. Эластичность, возможность быстрого увеличения и уменьшения – нужны новые алгоритмы управления распределением ресурсов и размещением рабочих нагрузок. Автономные вычисления, основанные на самоорганизации и самоуправлении
6. Узкие места в передачи данных. Решение – высокоскоростные сети с высокой пропускной способностью

Вопросы – безопасность (гарантия, что данные будут защищены в облаке), масштабируемость (облачные системы должны быть способны обрабатывать большие объемы данных и обеспечивать производительность при увеличении нагрузки), интеграция и совместимость с существующими инфраструктурами, зависимость от доступности сети, цена и модель оплаты, сложность разработки

## Понятие, принципы и типы виртуализации. Область эффективной виртуализации. Понятие виртуальной машины.

Виртуализация – логическая абстракция вычислительных ресурсов и предоставление пользователю системы, которая инкапсулирует (скрывает в себе) собственную реализацию. Пользователь работает с удобным для себя представлением объекта, и для него не имеет значения, как устроен объект.

Типы виртуализации:

1. Виртуализация на уровне набора команд – ISA – интерфейс между аппаратными средствами и ПО, который представляет собой набор машинных команд
2. Виртуализация на уровне системных вызовов – эмуляция системных вызовов библиотек функций ОС
3. Виртуализация на уровне библиотечных вызовов – на уровне API приложений – обеспечивает переносимость и совместимость приложений на разных ОС

Виртуализация ЭВМ – размещение нескольких логических машин в рамках физической

Цели:

1. Предоставить каждому пользователю изолированную среду исполнения приложений
2. Повысить гибкость использования ресурсов ЭВМ, исполняемыми на ней приложениями
3. Повысить защищенность приложений друг от друга, исполняемых на одной и той же ЭВМ
4. Повысить эффективность использования аппаратных средств ЭВМ

Область эффективной виртуализации лежит между «производительностью», «гибкостью» и «безопасностью».

Виртуальная машина – это полностью изолированный программный контейнер, который работает с собственной операционной системой и приложениями, подобно физическому компьютеру. Виртуальная машина действует так же, как физический компьютер, и содержит собственные виртуальные (программные) ОЗУ, жесткий диск, сетевой адаптер.

## История развития технологий виртуализации.

1965 – Выражение «Hypervisor» впервые появилось применительно к ПО обработки RPQ на ЭВМ IBM 360/65

1966 – В Кембриджском научном центре эмулятор Cp-40 для S/360-40, CP – Control Program или гипервизор

1967 - CP-67/CMS представляла собой ОС, которая использовала виртуализацию на уровне набора команд (ISA)

В начале 70-х гипервизор СР-67, был переработан в виде OS VM/370 для нового семейства машин System/370, выпущенного на рынок в 1972 г.

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

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 х86. Позднее включились такие компании как: Parallels и Oracle, RedHat и Microsoft

2011 – сформулированы основные модели развертывания и признаки облачных вычислений. В их основе – технологии виртуализации

## Требования к архитектуре ЭВМ для поддержки виртуализации. Проблемы виртуализации платформы Intel x86

Требования виртуализации:

1. Безопасность – у гипервизора должно быть полное управление виртуализированными ресурсами
2. Эквивалентность – поведение программы на ВМ должно быть идентичным поведению этой же программы, запущенной на реальном оборудовании
3. Эффективность – основная часть кода в ВМ должна выполняться без вмешательства гипервизора

Проблемы виртуализации в архитектуре Intel x86: в наборе команд есть команды, способные изменить состояние процессора и исполняемые в пользовательском режиме:

- поведение которых зависит от режима исполнения – служебные инструкции

- которые влияют на управление – привилегированные инструкции

В архитектуре Intel x86 при попытке сделать в пользовательском режиме то, что вы не должны делать в этом режиме, оборудование должно вызвать системное прерывание. Вдобавок к проблемам с инструкциями, которые, выполняясь в пользовательском режиме, вызывали системные прерывания, были еще и инструкции, которые могли считывать конфиденциальное состояние, не вызывая системных прерываний.

До 2005 года на процессорах x86 программа могла определить, в каком режиме она выполняется, пользовательском или в режиме ядра, путем чтения селектора своего кодового сегмента. Операционная система, основываясь на этой информации, могла принимать неправильные решения, если программа предоставляла ложную информацию о своем режиме выполнения.

## Требования к гипервизору, стратегия и основные подходы к обработке невиртуализируемых инструкций процессора.

Требования к гипервизору:

1. Программа, работающая под гипервизором, должна показывать поведение, идентичное тому, что было продемонстрировано при непосредственном запуске на эквивалентной машине
2. Гипервизор должен полностью контролировать виртуализированные ресурсы
3. Статистически значимая часть машинных инструкций должна выполняться без вмешательства гипервизора

Монитор виртуальных машин – гипервизор - 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)

Стратегии обработки невиртуализируемых инструкций:

1. Двоичная трансляция – гипервизор отслеживает работу гостевых ОС, невиртуализированные инструкции гостевой ОС заменяются другими – замена двоичных кодов проблемных команд аналогичными, но допускающими виртуализацию
2. Перехват гипервизором невиртуализируемых инструкций
3. Паравиртуализация – код гостевой ОС модифицируется для использования только инструкций, которые можно виртуализировать
4. Обход или замена невиртуализируемых инструкций
5. Поддержка аппаратной виртуализации

## Виртуализация процессора, памяти, устройств ввода/вывода и сети.

Виртуализация процессора

Виртуализация процессора позволяет запускать несколько виртуальных машин (ВМ) на одном физическом процессоре. ВМ могут работать с различными операционными системами и приложениями, и каждая ВМ будет думать, что она имеет полный контроль над процессором.

Виртуализация памяти

Для виртуализации памяти ВМ необходимо обеспечить несколько уровней трансляции адресов – виртуальной памяти ВМ в физическую память ВМ и физическую память ВМ (находится в виртуальной памяти реальной машины) в физическую память реальной машины

Виртуализация внешней памяти – физические устройства внешней памяти:

1. Жесткие диски HDD
2. Твердотельные диски SDD
3. Подключаемые разделы внешних систем хранения

Виртуальные диски:

1. Файлы образов ВМ и файлы данных ВМ

Виртуализация устройств ввода-вывода

Виртуализация устройств ввода/вывода позволяет нескольким виртуальным машинам использовать общие физические устройства ввода/вывода, такие как сетевые адаптеры, графические карты или дисковые накопители. ВМ обращаются к виртуальным экземплярам устройств, которые перенаправляются на физические устройства.

Виртуализация сети

Виртуализация сети позволяет создавать виртуальные сетевые ресурсы, такие как виртуальные сетевые адаптеры, коммутаторы или маршрутизаторы.

## Два подхода к виртуализации процессора. Полная виртуализация и паравиртуализация.

Полная виртуализация – каждая виртуальная машина работает на точной копии реального оборудования. KVM и VMware ESX - это примеры полной виртуализации. Полная виртуализация полностью имитирует требуемое оборудование для гостевой операционной системы. В итоге получается среда, аналогичная ОС, работающей на отдельном сервере. Использование полной виртуализации позволяет администраторам запускать виртуальные среды любых конфигураций без каких-либо дополнительных аппаратных ухищрений

Паравиртуализация – каждая виртуальная машина работает на слегка измененной копии реального оборудования. Паравиртуализацией называется подход, при котором на ВМ запускается особым образом подготовленная (измененная и/или перекомпилированная) версия гостевой ОС

Предоставляя гостевой ОС доступ к базовому оборудованию, паравиртуализация обеспечивает связь между гостевой ОС и гипервизором, тем самым повышая производительность и эффективность системы. В общем случае паравиртуализация работает быстрее, так как нет накладных расходов на ненужную эмуляцию

Основное различие между паравиртуализацией и полной виртуализацией заключается в возможности внесения изменений в гостевую ОС.

## Два вида реализации виртуальных машин (ВМ процесс и ВМ система). Возможные варианты реализации гипервизора (монитора виртуальных машин).

ВМ-процесс – виртуальная платформа, созданная для отдельного процесса и уничтожаемая после завершения процесса

ВМ-система – виртуальная машина, которая поддерживает ОС вместе со многими пользовательскими процессами. Это виртуализация операционных систем

Возможные варианты реализации монитора виртуальных машин – гипервизора. Гипервизор позволяет нескольким виртуальным машинам совместно использовать систему.

Организация программного стека виртуализации:

1. Естественная (тип 1) – гипервизор реализуется как тонкий программный уровень, работающий непосредственно над аппаратным обеспечением хост-машины.
2. Хостинг (тип 2) – виртуальная машина работает поверх существующей ОС – гипервизор 2го типа
3. Гибридный – гипервизор использует оборудование совместно с существующей ОС

## Два вида виртуализация средствами операционных систем (с помощью VMM и средствами ОС).

2 вида виртуализации ОС:

1. Виртуализация с помощью монитора виртуальных машин VMM (гипервизор)

Гипервизор работает на физической машине (хосте) и управляет созданием и управлением виртуальных машин. Гипервизор создает изолированную среду, в которой каждая ВМ имеет свою собственную операционную систему, библиотеки и приложения. Он контролирует доступ ВМ к аппаратным ресурсам и обеспечивает их изоляцию друг от друга. Примеры гипервизоров включают VMware ESXi, Microsoft Hyper-V, Xen и KVM

1. Виртуализация средствами базовой ОС или контейнеризации (разделение на разделы, partition)

В этом случае виртуализация осуществляется средствами самой операционной системы, без использования отдельного гипервизора. Операционная система предоставляет функциональность, позволяющую создавать и управлять виртуальными средами, называемыми контейнерами или виртуализованными операционными системами. Контейнеры обеспечивают изоляцию и виртуализацию ресурсов на уровне операционной системы, позволяя нескольким приложениям работать в изолированной среде на одном хосте

## Модель защиты памяти архитектуры Intel\_x86. Проблема одновременной работы гипервизора и гостевой ОС в режиме ядра. Режимы защиты памяти при различных типах виртуализации и паравиртуализации.

Защита памяти в архитектуре x86 – иерархическая 4х уровневая система.

* Кольцо 0 – наиболее привилегированный уровень, программы, работающие тут, напрямую взаимодействуют с физическим оборудованием – ЦП и память
* Кольца 1 и 2 – в ОС Linux и Windows не используются
* Кольцо 3 – используется программами в пользовательском режиме

Проблема - VMM требует доступа к памяти, ЦП, устройствам ввода-вывода – это режим ядра. Гостевая ОС также должна иметь доступ ко всем ресурсам, поскольку она не знает об уровне виртуализации, для этого она должна работать в кольце 0, как и VMM. В связи с тем, что в каждый момент времени только одно ядро может работать в кольце 0, то гостевые операционные системы должны работать в другом кольце с меньшими привилегиями (кольцо 1)/должны быть изменены для работы в пользовательском режиме

ИСПОЛЬЗОВАНИЕ КОЛЬЦА 1 ДЛЯ ЗАЩИТЫ ПАМЯТИ: Гипервизор находится в ядре (кольцо 0), приложения работают в пользовательском режиме (кольцо 3), а гостевая операционная система выполняется в кольце 1 с привилегиями, находящимися между ядром и пользовательскими процессами. Ядро имеет более высокие привилегии по сравнению с пользовательскими процессами, попытка доступа к памяти ядра ведет к нарушению прав доступа.

ПРИ ПОЛНОЙ ВИРТУАЛИЗАЦИИ: Гипервизор находится в кольце 0, гостевая ОС – в кольце 1, пользовательские приложения – в кольце 3. Пользовательские приложения делают запрос к аппаратному обеспечению хостовой машины, гостевая ОС обращается к VMM, VMM перехватывает, осуществляет бинарное преобразование команд и обращается к ОС

ПРИ ПАРАВИРТУАЛИЗАЦИИ: Использование паравиртуализации, которая предполагает модификацию гостевых ОС. Пользовательские приложения делают запрос к аппаратному обеспечению хостовой машины, модифицированная гостевая ОС обращается к уровню виртуализации, а тот к аппаратному обеспечению хостовой машины. Обращения к уровню виртуализации заменяются невиртуализированными инструкциями

## Гипервизор первого типа достоинства и недостатки.

Гипервизор 1го типа – на аппаратные средства устанавливается специализированная ОС (гипервизор), предназначенная для виртуализации на уровне набора команд аппаратной платформы. Это «родной монитор виртуальных машин». Он работает непосредственно на аппаратном уровне хостовой машины без привязки к ОС хоста, имеет прямой доступ к аппаратуре

Гипервизор первого типа работает напрямую с аппаратными компонентами хостовой системы, такими как процессоры, память, хранилище и сетевые интерфейсы.

Достоинства:

1. Отсутствие потребности в хозяйской ОС, ВМ устанавливается по факту на голое железо, а аппаратные ресурсы используются более эффективно
2. Изоляция виртуальных машин
3. Производительность – прямой доступ к аппаратным ресурсам

Недостатки:

1. Необходимость поддержки в гипервизоре драйверов внешних устройств, из-за чего возникают накладные расходы на используемые аппаратные ресурсы
2. Отсутствие учета особенностей гостевых ОС, меньшая гибкость в использовании аппаратных средств
3. Сложность настройки
4. Ограничения аппаратной совместимости
5. Отказоустойчивость

## Гипервизор 2-го типа достоинства и недостатки.

Гипервизор 2го типа – монитор виртуальных машин (гипервизор) работает в рамках хозяйской операционной системы (для 1го типа он работает над аппаратным обеспечением без привязки к хостовой ОС)

В процессе своей работы он модифицирует код гостевых ОС с целью обеспечить виртуализацию на уровне команд. При этом код, содержащий «чувствительные» команды, переписывается на лету (VMWare Workstation). Накладные расходы 10-15%

Достоинства:

1. Простота установки – обычное приложение на рабочей ОС
2. Упрощённое управление – удобный интерфейс управления, интегрируемый в ОС хоста
3. Разделение ресурсов между ВМ
4. Поддержка широкого спектра ОС

Недостатки:

1. Производительность – работает поверх ОС хоста, снижение производительности. Операционная система хоста занимает ресурсы, что может оказывать влияние на производительность виртуальных машин
2. Ограничения ОС хоста – гипервизор зависит от ОС хоста и ограничивается ее возможностями
3. Меньшая изоляция
4. Зависимость от ОС хоста – если эта ОС требует обновления, это может повлиять на работу гипервизора

## Паравиртуализация, достоинства и недостатки.

Паравиртуализация – техника виртуализации, при которой гостевые ОС 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

Достоинства:

1. Отсутствие потребности в хостовой ОС – ВМ устанавливаются фактически на голое железо, аппаратные ресурсы используются эффективно

Недостатки:

1. Сложность реализации подхода и необходимости гипервизора – специализированная ОС
2. Гостевые ОС нужно модифицировать

Использование паравиртуализации, которая предполагает модификацию гостевых ОС, таким образом, чтобы нейтрализовать, либо полностью исключить влияние “чувствительных” команд, средствами хозяйской ОС

## Система виртуализации Xen.

Разработка некоммерческого гипервизора Xen началась как исследовательский проект Кембриджского университета. Основатель – Иан Пратт. Изначально самая развитая платформа для паравиртуализации

Citrix XenApp - предназначен для виртуализации и публикации приложений. XenApp имеет огромное количество пользователей по всему миру и во многих компаниях является ключевым компонентом ИТ-инфраструктуры.

Citrix XenServer - платформа для консолидации серверов предприятий среднего масштаба, включающая основные возможности для поддержания виртуальной инфраструктуры

Citrix XenDesktop - решение по виртуализации десктопов предприятия, позволяющее централизованно хранить и доставлять рабочие окружения в виртуальных машинах пользователям.

Бесплатный Xen – Open-source версия обычно в образовательных целях. Версии Xen включаются в дистрибутивы многих Linux-систем, что позволяет их пользователям применять ВМ для изоляции ПО в гостевых ОС с целью его тестирования и изучения проблем безопасности, без необходимости установки платформы виртуализации

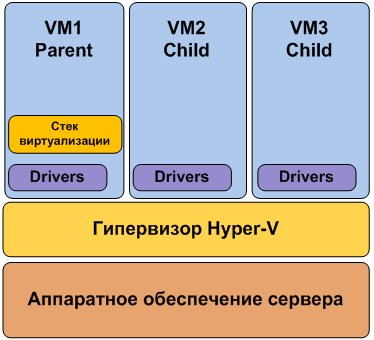
Черты:

1. Открытый исходный код и поддержка сообщества
2. Паравиртуализация и полная виртуализация – 2 режима
3. Поддержка различных операционных систем
4. Ресурсное планирование
5. Живая миграция – перемещение работающих ВМ с одного физического хоста на другой без прерывания работы
6. Безопасность – изоляция ВМ друг от друга и от гипервизора

## Система виртуализации Hyper-V.

Hyper-V представляет собой гипервизор – прослойку между оборудованием и виртуальными машинами уровнем ниже операционной системы. Hyper-V — это технология виртуализации на базе низкоуровневой оболочки (или по-другому "гипервизора") для отдельных 64-разрядных версий Windows. Гипервизор ключевым компонентом технологии виртуализации. Это процессор-зависимая платформа виртуализации, позволяющая нескольким изолированным операционным системам использовать общую аппаратную платформу

Hyper-V поддерживает изоляцию по разделам. Раздел – логическая единица изоляции, поддерживаемая гипервизором, в котором работают ОС.



VMBus — канальный механизм взаимодействия, используемый для взаимодействия между разделами и перечисления устройств в системах с несколькими активными виртуализированными разделами. VMBus устанавливается со службами интеграции Hyper-V

## Система виртуализации VMware.

В 1998 VMware запатентовала свои программные техники виртуализации

Базовой единицей виртуализации VMware является виртуальная машина (ВМ) – программное представление физического компьютера. Операционная система, запущенная в ВМ, называется гостевой ОС. Каждая ВМ включает конфигурационный файл, в котором хранятся параметры ВМ, файл виртуального диска, представляющий собой программную версию жесткого диска, и файл журнала.

Машины настраиваются с помощью клиента vSphere – интерфейса командной строки для управления ВМ. Гипервизор – это программный продукт низкого уровня, взаимодействующий с ресурсами физического компьютера (называется хостом) и распределяющий эти ресурсы между другими ОС, известными как гостевые.

1 типа – Vmware ESXi - обеспечивает уровень виртуализации, который абстрагирует процессор, хранилище, память и сетевые ресурсы физического хоста в несколько ВМ

2 типа - VMware Workstation – работает как приложение в настольной ОС

* VMware Workstation – платформа, ориентированная на Desktop-пользователей и предназначенная для использования разработчиками ПО, а также профессиональными пользователями
* VMware Player – бесплатный проигрыватель виртуальных машин на основе виртуальной машины
* VMware ESX Server – гипервизор, разбивающий физические сервера на множество ВМ. Это основа пакета VMware vSphere, входит во все выпуски
* VMware Fusion – настольный продукт виртуализации на платформе Mac (Apple)
* VMware vSphere – комплекс продуктов, предоставляющий надежную платформу виртуализации ЦОД

## Виртуализация на основе ОС достоинства и недостатки.

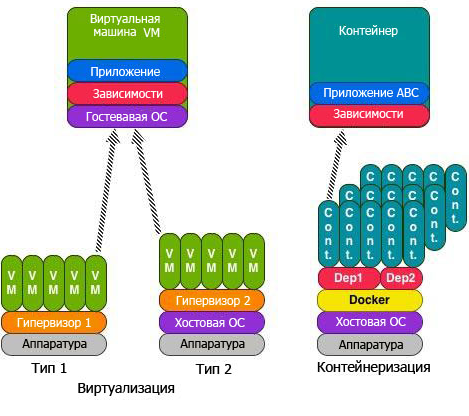
Виртуализация на основе ОС (контейнеризация) виртуализирует физический сервер на уровне операционной системы, позволяя запускать изолированные виртуальные серверы – Виртуальные Частные Серверы (Virtual Private Servers, VPS) или Контейнеры (Container, CTS)

Достоинства:

1. Минимальные накладные расходы
2. Высокая эффективность использования аппаратных ресурсов
3. Самая высокая степень консолидации – объединения нескольких ВМ на одном физическом сервере или хосте. Миграция ВМ, оптимизация ресурсов, устранение фрагментации ресурсов. Снижение количества физических серверов, сокращение затрат на энергию и улучшение использования аппаратных ресурсов
4. Накладные потери производительности контейнеров всего 3%
5. Отличная управляемость
6. Минимальные расходы на приобретение лицензий

Недостатки:

1. Виртуализация на основе ОС не позволяет запускать ОС с ядрами, отличными от ядра базовой ОС
2. Реализация однородных вычислительных сред
3. Степень изоляции контейнеров друг от друга ниже, чем у ВМ, управляемых VMM



## Виртуализация хранилищ данных. Классификация виртуальных хранилищ. Достоинства и недостатки виртуальных хранилищ данных.

Виртуализация хранилищ данных – это представление ресурсов хранения в абстрактном виде, как логические пространства хранения (тома) без привязки к физическим накопителям. Виртуализация хранилища позволяет многим пользователям или приложениям получать доступ к хранилищу, не беспокоясь о том, где и как это хранилище физически расположено или управляется

Достоинства виртуализации хранилищ:

1. Быстрое и надежное хранилище для вычислений и обработки данных
2. Предоставляет администратору системы хранения большую гибкость в управлении хранилищем для конечных пользователей
3. Предоставляет возможности для оптимизации использования хранилища и консолидации серверов, а также для выполнения миграции файлов без прерывания работы
4. Позволяет каждому виртуальному серверу запускать свою собственную операционную систему, и каждый виртуальный сервер также может быть перезагружен независимо друг от друга
5. Снижает стоимость, поскольку требуется меньше оборудования
6. Использует ресурсы для экономии эксплуатационных расходов
7. Уменьшает количество серверов
8. Параллельно можно запускать больше приложений

Недостатки:

1. Объем дискового пространства виртуального хранилища ограничен
2. Миграция данных занимает много времени

Классификация виртуальных хранилищ:

1. Блочная
2. Дисковая
3. Лента (носитель, накопитель, библиотека)
4. Файловая система
5. Виртуализация файлов

## Блочные виртуальные хранилища данных. Достоинства и недостатки.

В системе хранилища на уровне блоков используются «сырые» необработанные тома хранения, не имеющие структуры, каждый блок можно рассматривать как отдельный жесткий диск. Блоки контролируются серверными ОС, они имеют возможность форматировать блоки под размещение любой ФС

Преимущества:

1. Предлагают лучшую производительность, чем системы хранения на уровне файлов
2. Каждый блок или том хранилища может рассматриваться как независимый диск и управляется ОС внешнего сервера
3. Блочные хранилища популярны в сетях хранения данных SAN. Они надежнее и транспортные системы их эффективные. Может поддерживать внешнюю загрузку с них ОС
4. Виртуальное блочное устройство можно форматировать под любую ФС
5. На виртуальное блочное устройство можно установить любую ОС
6. Детальное управление

Недостатки:

1. Привязка к серверу
2. Ограниченность метаданных – их объем гораздо меньше, чем в файловом хранилище
3. Стоимость

Данные на дисках хранятся в блоках. Традиционно их размер 512 байт (1 сектор). Более крупные системы используют блоки по 4 КБ (8 х 512 байт). Блочные системы называются так потому, что хранимые в них данные разбиваются на блоки одинакового размера. Файл данных может быть размещен в конечном числе блоков, и если последний из этих блоков остается незаполненным, то он все равно будет иметь тот же фиксированный размер, что и заполненные блоки. [Сервер](https://itelon.ru/catalog/oborudovanie/servers/) получает доступ к этим блокам через сеть хранения данных SAN (storage attached network).

## Файловые виртуальные хранилища данных. Достоинства и недостатки.

Файловое хранилище организует и предоставляет данные в виде иерархии файлов в папках. [Файловая система хранения](https://itelon.ru/solution/file-server/) больше всего похоже на то, как мы видим информацию в своем компьютере: т.е. в виде файлов во вложенных папках. Путь к файлу в файловом хранилище может быть довольно длинным, сквозь глубокую иерархию вложенных друг в друга папок.

Преимущества:

1. Простота реализации
2. Простота использования
3. Интуитивно понятное. Интерфейс доступен и прост. Иерархичность – знакомая модель
4. Легко масштабируется
5. Совместный доступ не ограничен
6. Невысокая цена

Недостатки:

1. Наличие пределов масштабирования
2. Сложность навигации при росте объемов – увеличивается время доступа. Файлы будут открываться медленнее

* Непрерывное размещение – выделяется набор последовательных блоков
* Цепочное размещение – поблочное размещение, не требуется, чтобы блоки были последовательно по номерам. Каждый блок содержит указатель на следующий блок
* Индексированное размещение - таблица размещения файлов содержит отдельный одноуровневый указатель (индекс) для каждого файла. В индексе последовательно указаны номера блоков, в которых размещен файл.

## Виртуализация сетевых хранилищ данных. Виды сетевых хранилищ (DAS, NAS, SAN).

3 вида сетевых хранилищ:

1. Хранилище с прямым подключением DAS – Direct Attached Storage
2. Сетевое хранилище NAS – Network Area Storage
3. Сеть хранения данных SAN – Storage Area Network

Хранилище с прямым подключением данных DAS – это традиционный метод хранения данных, жесткие диски в нем подключаются к физическому серверу.

Сетевое хранилище NAS – это машина (файловый сервер), которая находится в сети и обеспечивает хранение данных для других машин

Сеть хранения данных SAN – развертывается специальное оборудование (СХД – системы хранения данных) и ПО для преобразования простых дисков в высоконадежные массивы дисков хранения данных

## Виртуализация сети. Методы виртуализации VLAN, VIP, VPN. Достоинства и недостатки виртуализации сетей.

Виртуализация сети – это метод объединения доступных ресурсов в сети. Это возможно за счет разделения доступной полосы пропускания на каналы, каждый из которых независим от других, и каждый может быть назначен (переназначен) определенному серверу или устройству в реальном времени

Достоинства:

1. Экономия средств
2. Снижение общего потребления электроэнергии
3. Возможность быстрого восстановления после аппаратного сбоя
4. Обеспечивает полное аварийное восстановление

Недостатки:

1. Высокая степень сложности, снижение производительности
2. Сетевой администратор и пользователь нужны квалифицированные
3. Требует правильного и вдумчивого планирования

Методы виртуализации сети:

1. VLAN – виртуальная локальная сеть
2. VIP – виртуальный IP-адрес
3. VPN – виртуальная частная сеть

VLAN - на одном физическом интерфейсе создаются виртуальные локальные сети – логическая топология сети никак не будет зависть от физической.

VIP - Виртуальный IP (VIP) — это IP-адрес, который можно назначить сразу нескольким машинами или сделать вторым адресом для сетевого интерфейса. С помощью VIP можно создать отказоустойчивую инфраструктуру. Назначьте его нескольким машинам сразу, и адрес будет переходить от машины к машине

VPN - создает частное сетевое подключение между устройствами с помощью Интернета. Сети VPN используются для безопасной и анонимной передачи данных по публичным сетям. Принцип их работы заключается в маскировании IP-адресов пользователей и шифровании данных. VPN-подключение перенаправляет пакеты данных от вашего компьютера к другому удаленному серверу, перед тем как отправить их куда-то еще