# **Уровни инфраструктуры ЦОД?**

Пять уровней:

физическую инфраструктуру,

виртуальную инфраструктуру,

программно-определяемую инфраструктуру,

уровень оркестрации

уровень услуг.

**ИНФРАСТРУКТУРА ЦЕНТРА ОБРАБОТКИ ДАННЫХ**

**БЕЗОПАСНОСТЬ**

**НЕПРЕРЫВНОСТЬ БИЗНЕСА**

**УПРАВЛЕНИЕ**

**ФИЗИЧЕСКАЯ ИНФРАСТРУКТУРА (ЛУЧШАЯ В СВОЕМ КЛАССЕ / КОНВЕРГЕНТНАЯ)**

**Вычислительные ресурсы**

**СХД**

**ВИРТУАЛЬНАЯ ИНФРАСТРУКТУРА**

**Виртуальная сеть**

**Виртуальные ресурсы**

**Виртуальное хранилище**

**ОРКЕСТРАЦИЯ**

**УСЛУГИ**

**Каталог услуг**

**ПРОГРАММНО-ОПРЕДЕЛЯЕМАЯ ИНФРАСТРУКТУРА**

**Сеть**

**Программно-определяемые вычислительные ресурсы**

**Программно-определяемая система хранения**

**Программно-определяемая сеть**

**Механизмы отказоустойчивости**

**Резервное копирование и архивирование**

**Репликация**

**Механизмы обеспечения безопасности**

**Стратегическое управление, снижение рисков и обеспечение соответствия требованиям регуляторов**

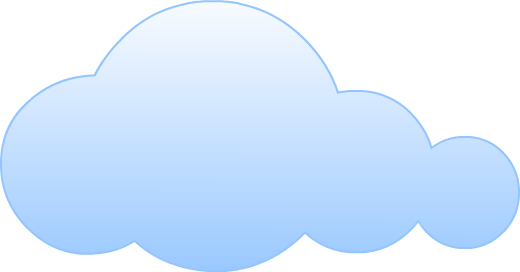
**ПРИЛОЖЕНИЯ**

**Бизнес-приложения**

**Внутренние приложения**

**Приложения для   
3-й платформы**

**Управление работой системы хранения**



**Расширяемость  
облачной среды**

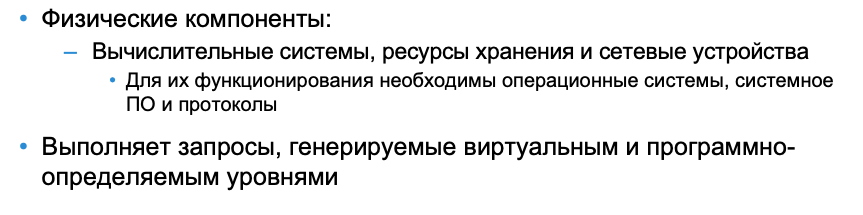
**ОБЛАКО**

**ПО для оркестрации**

**Портал самообслуживания**

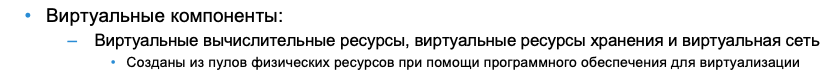
# **Компоненты и функции каждого уровня ЦОД?**

Физическая инфраструктура:



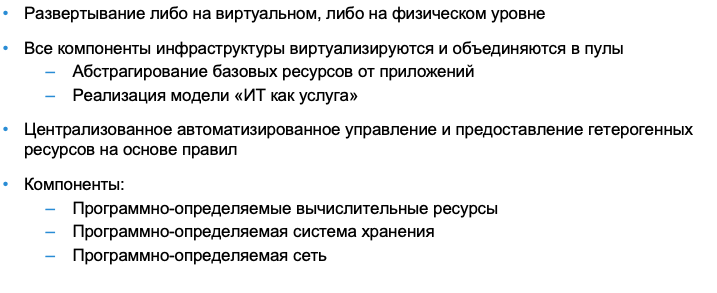
Ключевая функция физической инфраструктуры заключается в исполнении запросов, генерируемых виртуальной и программно-определяемой инфраструктурами. Эти запросы могут включать сохранение данных в устройствах хранения данных, выполнение обмена данными между двумя вычислительными системами, выполнение программ в вычислительных системах и создание резервных копий данных.

Виртуальная инфраструктура:

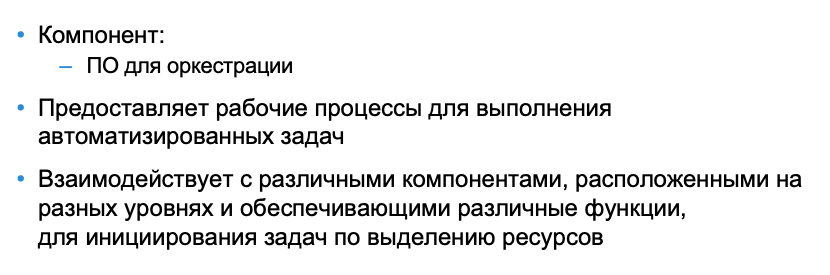


*Виртуализация —* это процесс абстрагирования физических ресурсов,например, вычислительных ресурсов, ресурсов хранения и сетевых ресурсов, и создания из них виртуальных ресурсов.

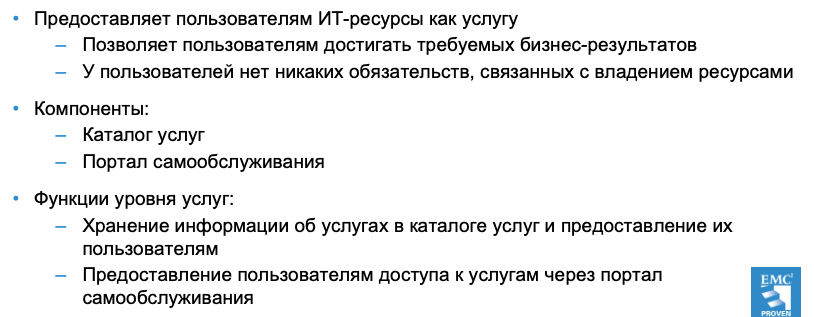
Программно-определяемая инфраструктура:



Орекстрация:



Услуги:

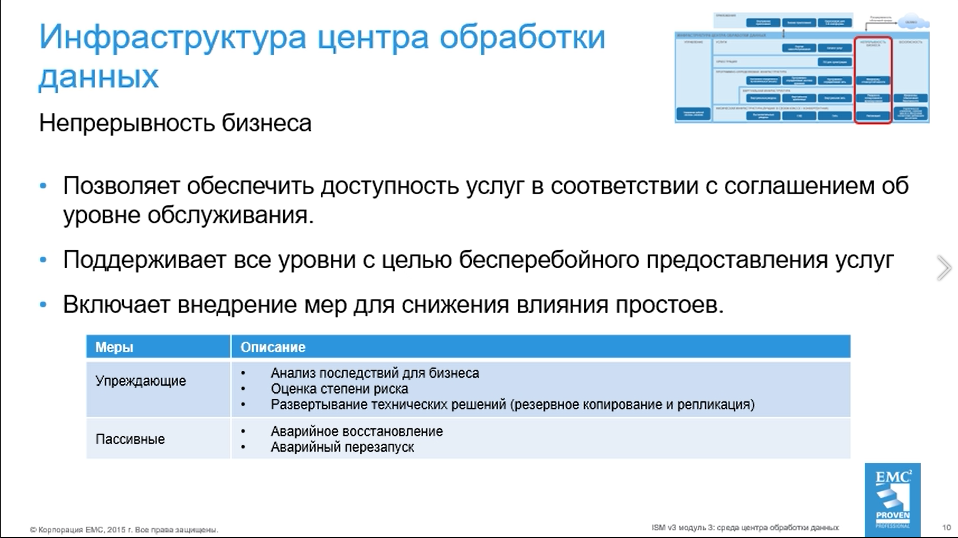


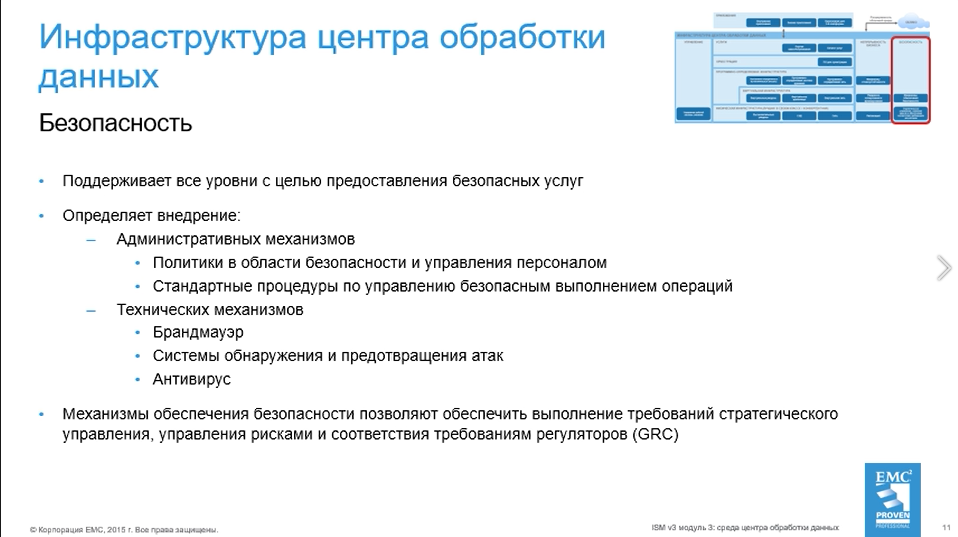
# **Межуровневые функции в ЦОД?**

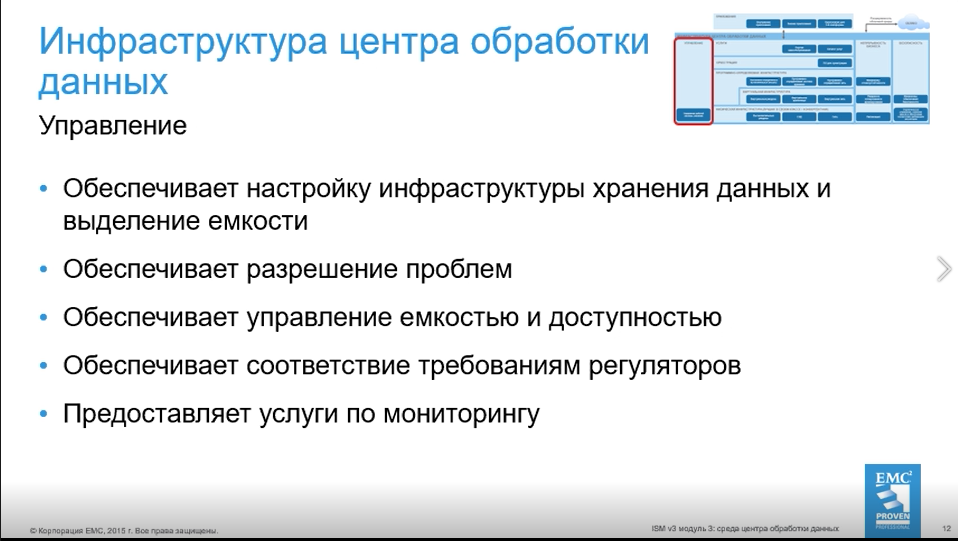
Три межуровневые функции — это непрерывность бизнеса, безопасность и управление.

Функции непрерывности бизнеса и безопасности включают в себя механизмы и процессы, необходимые для обеспечения надежного и безопасного доступа к приложениям, информации и услугам.

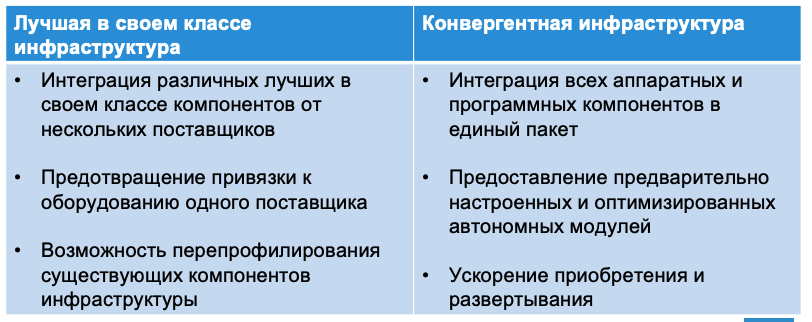
Функция управления включает в себя различные процессы, обеспечивающие эффективное администрирование центра обработки данных, а также услуги, предназначенные для удовлетворения бизнес-требований.





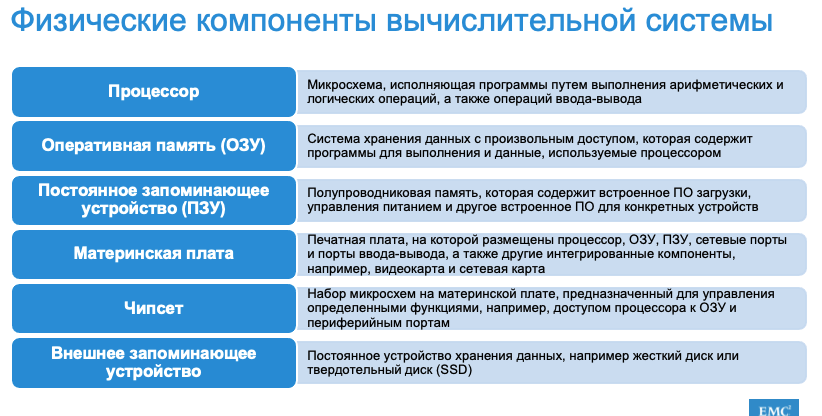


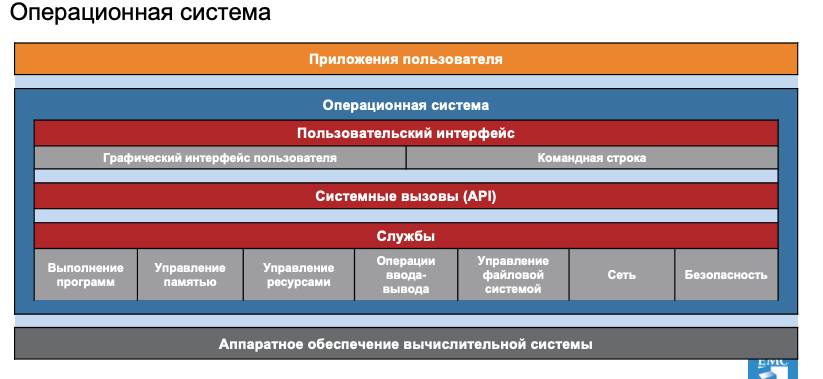
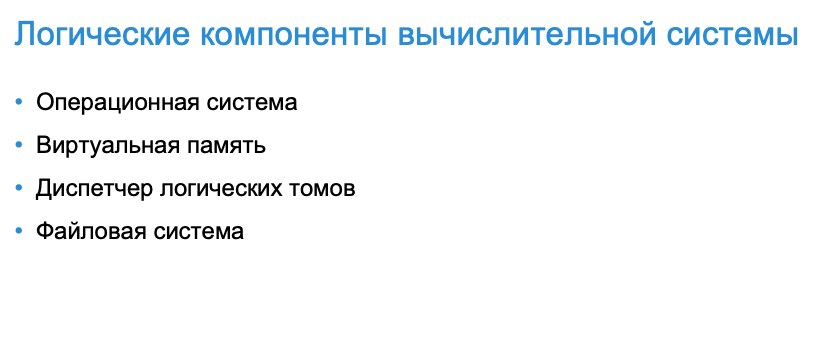
# **Отличия лучшей в своем классе инфраструктуры и конвергированной инфраструктуры?**

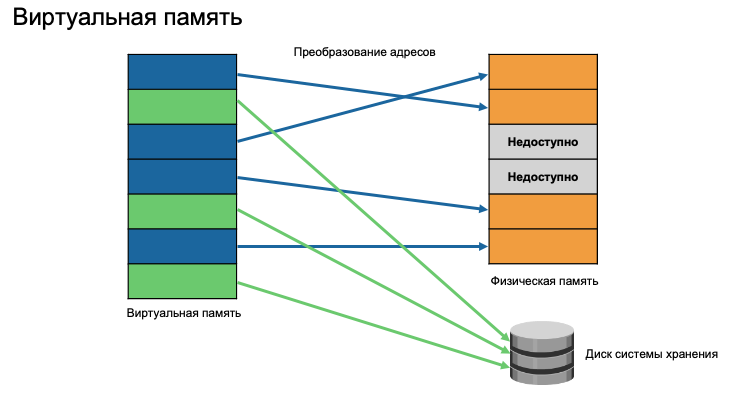
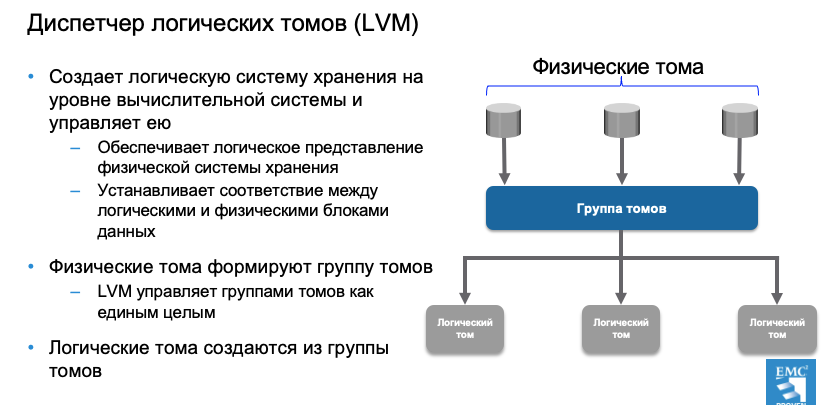
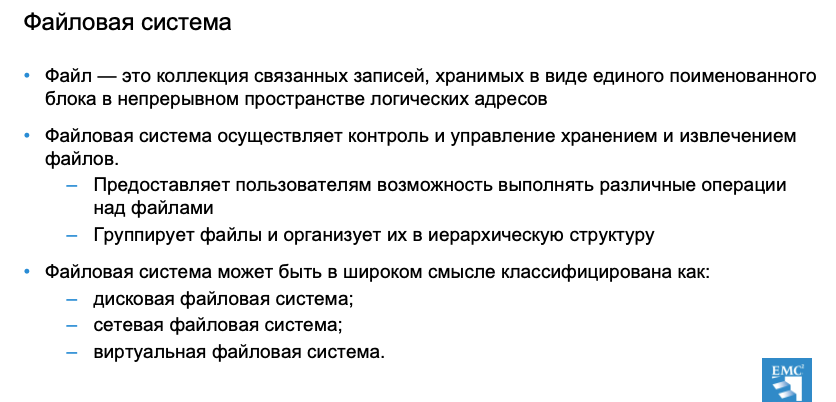
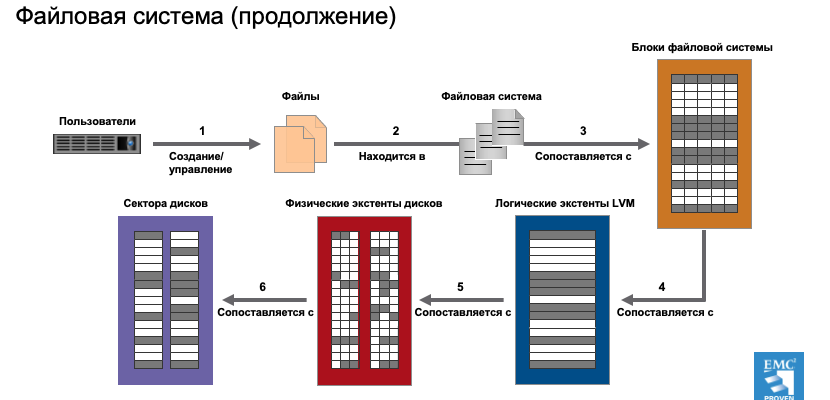


# **Физические и логические компоненты вычислительной системы?**

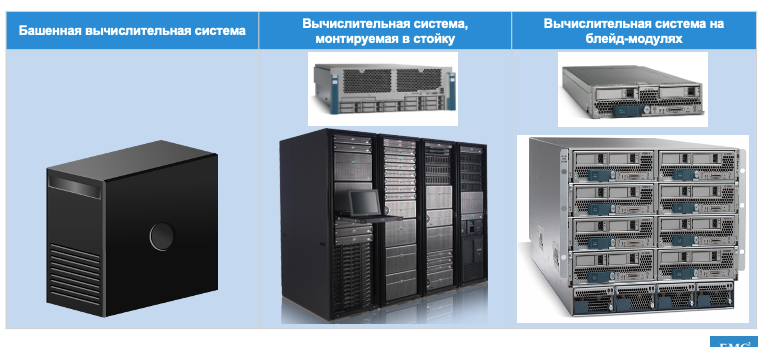






# **Типы вычислительных систем?**



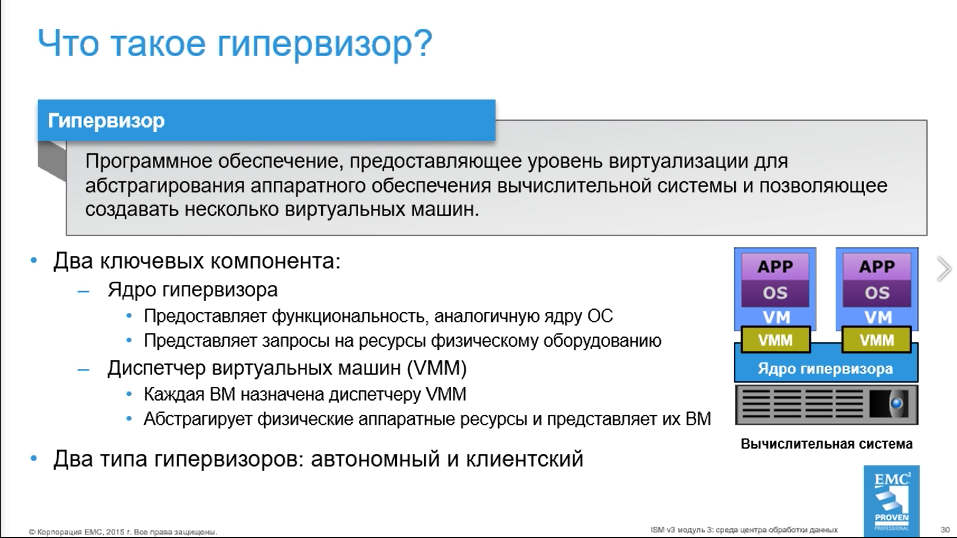
Башенная вычислительная система, известная также как башенный сервер — это вычислительная система, собранная в вертикальном автономном корпусе, называемом «башней», который напоминает по внешнему виду корпус настольного ПК.

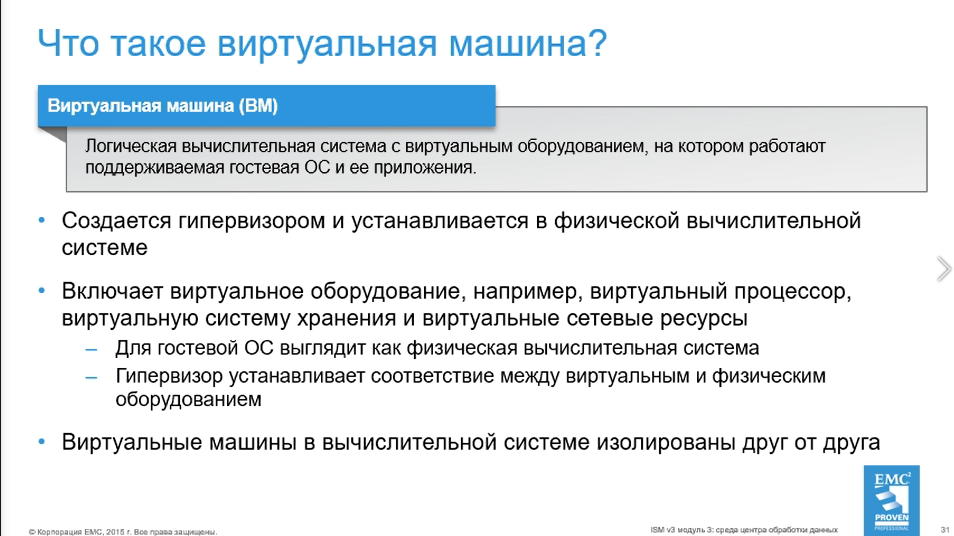
Вычислительная система, монтируемая в стойку, известна также как стоечный сервер и предназначена для закрепления в раме, которая называется «стойкой».

Вычислительная система на блейд-модуле, называемая также блейд-сервером, представляет собой электронную печатную плату, на которой присутствуют только базовые компоненты обработки, такие как процессор(ы), память, интегрированные сетевые контроллеры, диск системы хранения, а также необходимые карты и порты ввода-вывода.

# **Виртуализация вычислительных ресурсов, гипервизор и виртуальная машина?**







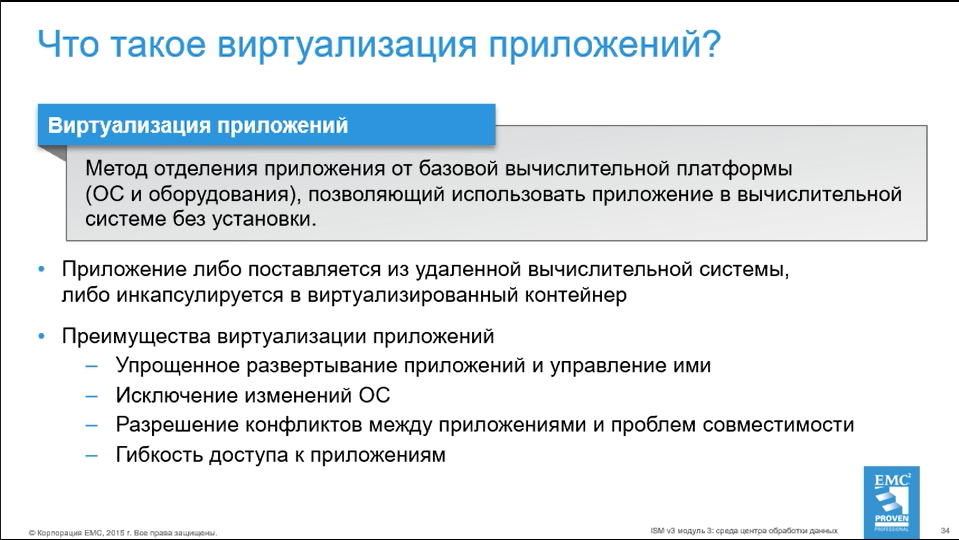
Виртуализация вычислительных ресурсов — это метод абстрагирования физического оборудования вычислительной системы от операционной системы (ОС) и приложений.

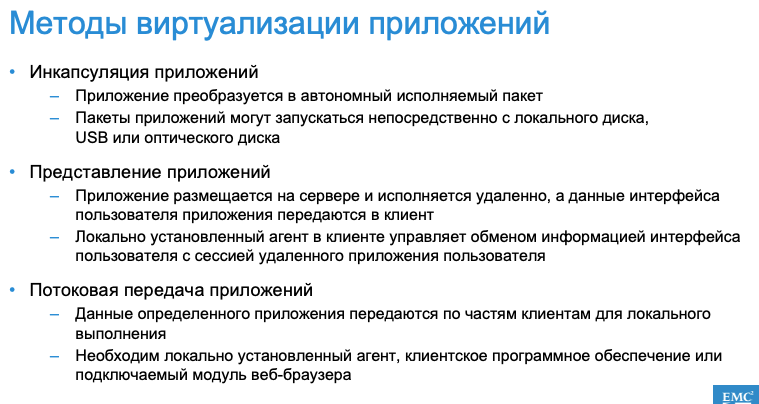
Отделение физического оборудования от ОС и приложений обеспечивает возможность параллельной работы нескольких операционных систем в одиночной или кластерной физической вычислительной системе. Виртуализация вычислительных ресурсов позволяет создавать виртуальные вычислительные системы, называемые виртуальными машинами (ВМ). На каждой ВМ запущены операционная система и приложения, и такая ВМ изолирована от других ВМ в той же вычислительной системе.

Виртуализация вычислительных ресурсов достигается за счет использования гипервизора, который представляет собой программное обеспечение для виртуализации, устанавливаемое в физической вычислительной системе. Гипервизор предоставляет всем ВМ виртуальные аппаратные ресурсы: ЦП, память, ресурсы хранения и сетевые ресурсы. В зависимости от возможностей оборудования в одной физической вычислительной системе можно создать большое количество ВМ.

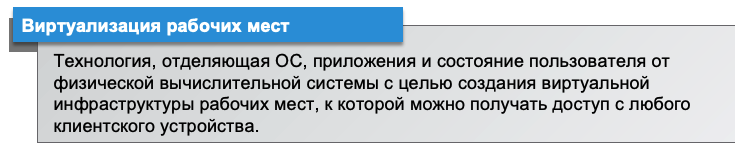
ВМ — это логический объект, но для ОС, запущенной на этой ВМ, она выглядит как физическая вычислительная система со своим процессором, памятью, сетевым контроллером и дисками. Тем не менее, все ВМ совместно используют то же базовое физическое оборудование вычислительной системы, а гипервизор динамически выделяет каждой ВМ аппаратные ресурсы вычислительной системы. С точки зрения гипервизора каждая ВМ представляет собой отдельный набор файлов. Этот вопрос будет рассмотрен позже в этом уроке.

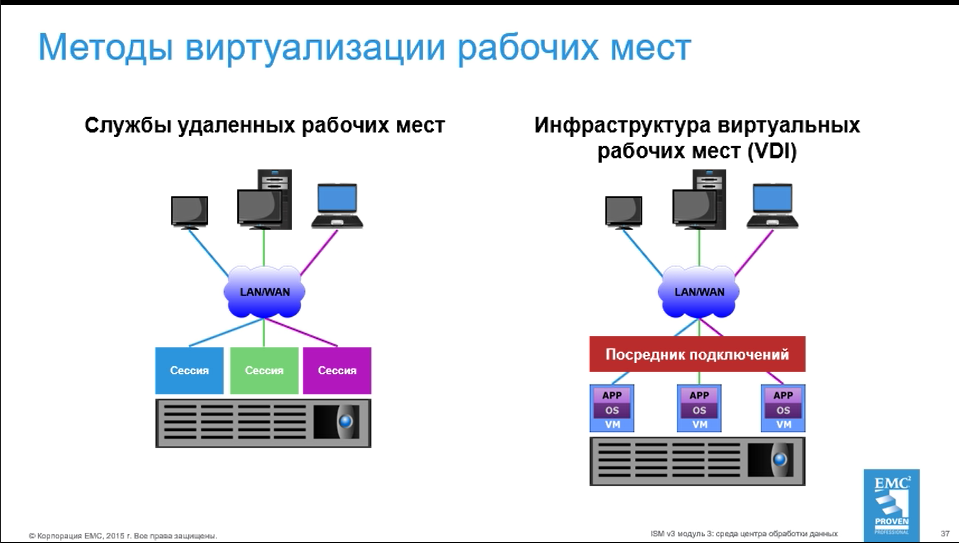
# **Виртуализация приложений и используемые для этого методы?**



Виртуализация приложений — это метод отделения приложения от базовой вычислительной платформы (ОС и оборудования), позволяющий использовать приложение в вычислительной системе без установки. 

# **Виртуализация рабочих мест используемые для этого методы?**





Существует два метода виртуализации рабочих мест: службы удаленных рабочих мест (RDS) и инфраструктура виртуальных рабочих мест (VDI).

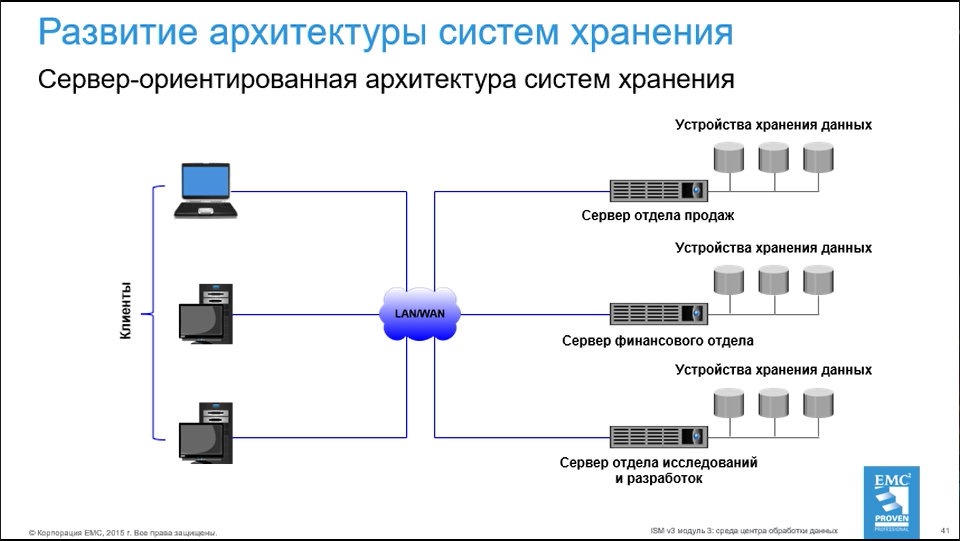
При использовании служб удаленных рабочих мест ОС и приложения размещаются в удаленной вычислительной системе и совместно используются несколькими пользователями. Каждый пользователь использует отдельную сессию виртуального рабочего места, в рамках которой выполняются приложения.

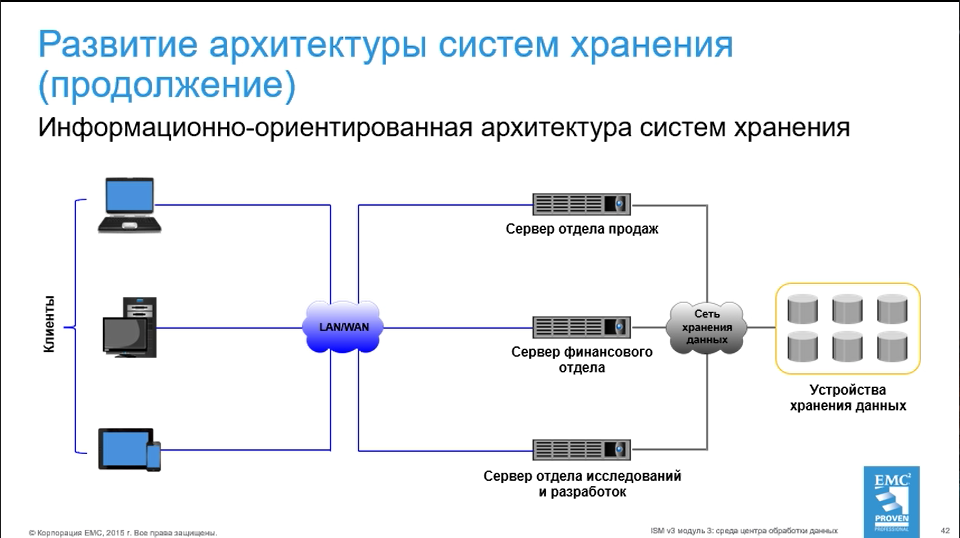
*В инфраструктуре виртуальных рабочих мест* ОС и приложения размещаются на виртуальных машинах, запущенных в удаленной вычислительной системе. ВМ виртуальных рабочих мест обычно размещаются на автономном гипервизоре. Каждому пользователю предоставляется отдельная ВМ рабочего места с собственной ОС и приложениями. Интерфейс пользователя ОС и приложений в ВМ рабочего места передается на конечное устройство по протоколу удаленного отображения, обеспечивающему возможность взаимодействия пользователя с виртуальным рабочим местом.

# **Развитие архитектуры систем хранения данных?**

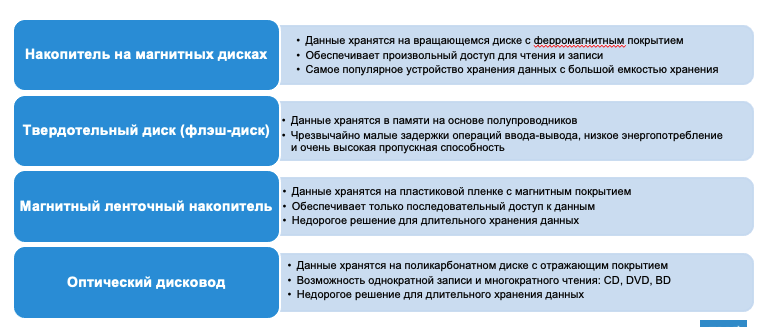
*сервер-ориентированной*. В этой архитектуре каждый сервер имеет ограниченное число устройств хранения данных, а каждое устройство хранения данных существует только в привязке к серверу, к которому оно подключено. Количество устройств хранения данных, которые можно подключить к одному серверу, ограничено, и емкость системы невозможно масштабировать.

В информационно-ориентированной архитектуре устройства хранения данных существуют совершенно независимо от серверов. Они управляются централизованно и совместно используются несколькими вычислительными системами. Устройства хранения данных, установленные в системах хранения, образуют пулы хранения данных, и несколько вычислительных систем получают доступ к одному пулу по специализированной высокоскоростной сети хранения данных (SAN). Сеть SAN используется для обмена информацией между вычислительными системами и системами хранения, а также для подключения систем хранения.





# **Типы устройств хранения данных?**



# **Возможности подключения вычислительной системы к вычислительной системе и к системе хранения?**



Для подключения вычислительной системы к вычислительной системе обычно используют протоколы на основе Internet Protocol (IP). Каждая физическая вычислительная система подключена к сети через одно или несколько устройств интерфейса хоста, называемых *сетевыми картами* (NIC). Для подключения обычно используют физические коммутаторы и маршрутизаторы. Коммутатор дает возможность различным вычислительным системам в сети связываться друг с другом. Маршрутизатор — это OSI-устройство 3-го уровня, которое позволяет различным сетям связываться друг с другом. Обычно в качестве сетевых кабелей используются медные и волоконно-оптические кабели. На приведенном на слайде рисунке показана сеть (локальная или глобальная), которая обеспечивает подключения между физическими вычислительными системами. Чтобы обеспечить требуемую производительность сети, необходимо убедиться в доступности соответствующих коммутаторов и маршрутизаторов с достаточной полосой пропускания и нужными портами.

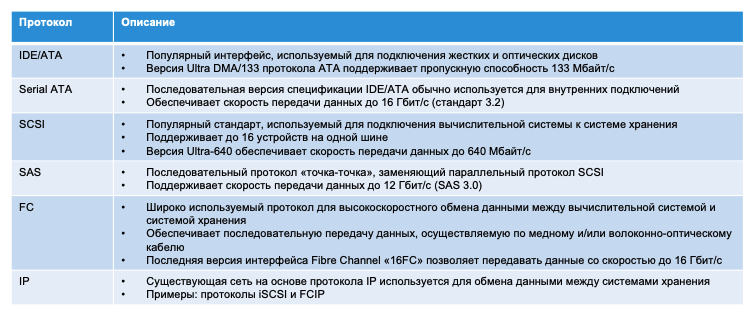
Возможность подключения и обмен данными между вычислительной системой и системой хранения обеспечиваются при посредстве физических компонентов и протоколов обмена данными. Физические компоненты, которые подключают вычислительную систему к системе хранения — это устройство интерфейса хоста, порт и кабель.

**HBA-адаптер.** *HBA-адаптер* — это устройство интерфейса хоста, которое подключает вычислительную систему к системе хранения или к сети хранения данных. Это плата со специализированной интегральной микросхемой (ASIC), которая выполняет функции интерфейса ввода-вывода между вычислительной системой и системой хранения, освобождая процессор от дополнительной рабочей нагрузки по обработке ввода-вывода. Обычно вычислительная система имеет несколько HBA-адаптеров.

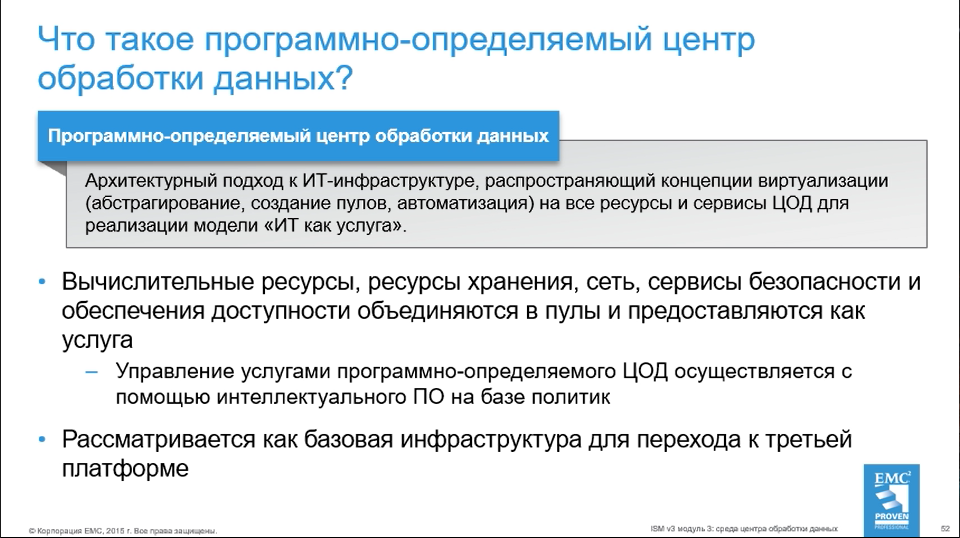
**Порт.** Порт — это специализированный разъем, обеспечивающий возможность подключения между вычислительной системой и системой хранения. HBA-адаптер может иметь один или несколько портов для подключения вычислительной системы к системе хранения. Кабели подключают вычислительные системы к внутренним или внешним устройствам при помощи медного провода или оптоволокна.

**Протокол.** Протокол обеспечивает обмен данными между вычислительной системой и системой хранения. Протоколы реализуются при помощи устройств интерфейса (или контроллеров), имеющихся в устройстве-источнике и устройстве назначения. Для обмена данными между вычислительной системой и системой хранения чаще всего используют протоколы Integrated Device Electronics/Advanced Technology Attachment (IDE/ATA), Small Computer System Interface (SCSI), Fibre Channel (FC) и Internet Protocol (IP).

# **Протоколы подключения систем хранения?**



# **Архитектура программно-определяемого ЦОД?**





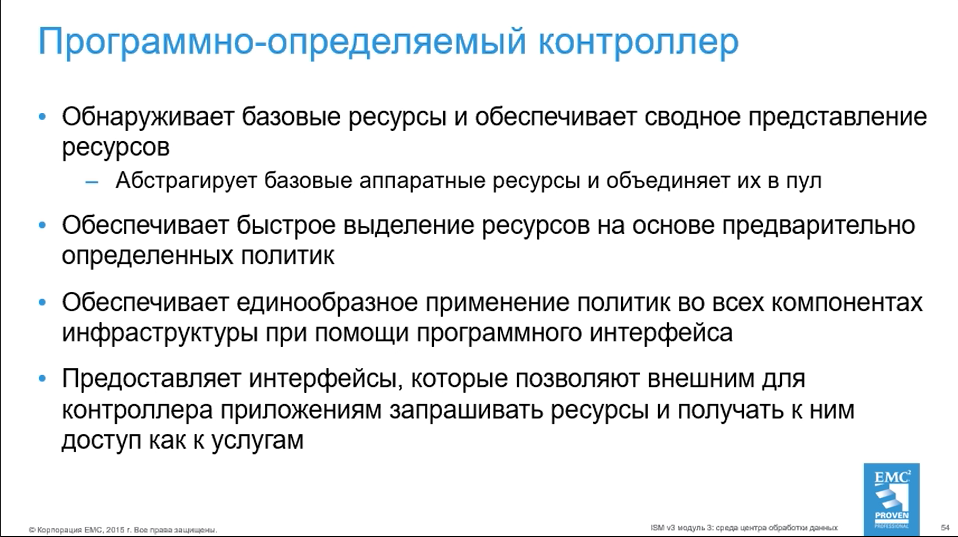
Программно-определяемый подход отделяет функции контроля или управления от базовых компонентов и предоставляет их внешнему программному обеспечению. Внешнее программное обеспечение берет на себя операции управления и обеспечивает возможность централизованного управления компонентами инфраструктуры от различных производителей.

В общем случае, компонент физической инфраструктуры (вычислительный, сетевой или устройство хранения) имеет путь управления и путь передачи данных. *Путь управления* устанавливает политики для ресурсов и управляет ими, а *путь передачи данных* выполняет действительную передачу данных. Программно-определяемый подход отделяет путь управления от пути передачи данных. Благодаря абстрагированию пути управления функция управления ресурсами действует на уровне управления. Это дает возможность разбивать пулы ресурсов и управлять ими на основе уникальных политик. Такое разделение пути управления и пути передачи данных обеспечивает централизацию задач предоставления данных и управления ими при помощи программного обеспечения, которое является внешним для компонентов инфраструктуры. Это программное обеспечение работает в централизованной вычислительной системе или на автономном устройстве, называемом *программно-определяемым контроллером*. Рисунок на слайде служит иллюстрацией программно-определяемой архитектуры, в которой функция управления абстрагирована при помощи контроллера от компонентов базовой инфраструктуры.

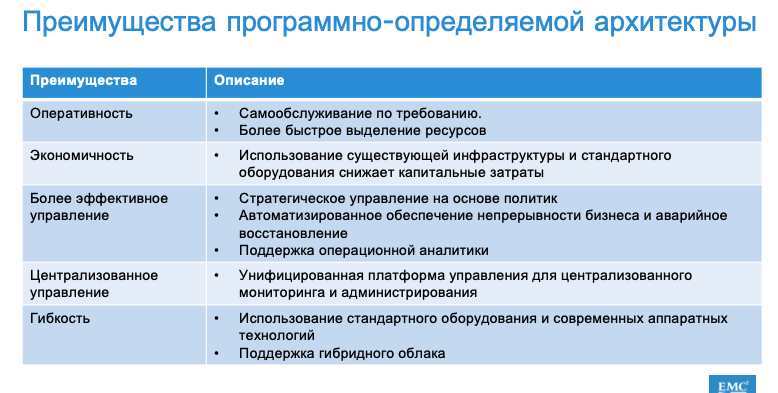
# **Программно-определяемый контроллер?**

*Программно-определяемый контроллер* — это программное обеспечение со встроенными интеллектуальными функциями, которое автоматизирует выделение ресурсов и настройку на основе определенных политик. Оно предоставляет организациям возможность динамически и единообразно изменять свою инфраструктуру и упрощает управление ею. Контроллер обнаруживает доступные базовые ресурсы и обеспечивает сводное представление ресурсов. Он абстрагирует базовые аппаратные ресурсы (вычислительные, сетевые и ресурсы хранения) и объединяет их в пул. Это обеспечивает быстрое выделение ресурсов из пула на основе предварительно определенных политик, которые соответствуют соглашениям об уровне обслуживания для различных потребителей.

Контроллер предоставляет единую точку управления для всей инфраструктуры, обеспечивая возможность управления инфраструктурой на основе политик. Контроллер позволяет администратору использовать программный интерфейс для управления ресурсами, возможностями подключения узлов и потоком трафика; контролировать поведение базовых компонентов; единообразно применять политики ко всем компонентам инфраструктуры; обеспечивать безопасность. Кроме того, контроллер предоставляет интерфейсы, которые позволяют внешним для контроллера приложениям запрашивать ресурсы и получать к ним доступ как к услугам.



# **Преимущества программно-определяемой архитектуры?**



1. Компоненты интеллектуальной системы хранения



Интеллектуальная система хранения состоит из двух ключевых компонентов: контроллера и системы хранения. Контроллер — это вычислительная система, которая запускает специально созданную ОС, ответственную за выполнение ряда ключевых функций для системы хранения. Примеры таких функций: обслуживание операций ввода-вывода с серверов приложений, управление хранением, защита RAID, локальная и удаленная репликация, выделение ресурсов хранения, автоматизированное многоуровневое хранение, сжатие, шифрование данных и интеллектуальное управление кэш-памятью.

Интеллектуальная система хранения обычно имеет несколько контроллеров для резервирования. Каждый контроллер состоит из одного или нескольких процессоров и определенного объема кэш-памяти для обработки большого количества запросов ввода-вывода. Эти контроллеры подключены к вычислительной системе напрямую или через сеть хранения. Контроллеры получают запросы ввода-вывода от вычислительных систем, которые контроллер читает или записывает в систему хранения. В зависимости от типа метода доступа к данным (рассматривается далее в этом модуле), который используется для системы хранения, контроллер может классифицироваться как блочный, файловый, объектный или унифицированный.

Интеллектуальная система хранения может состоять только из жестких дисков, только из твердотельных дисков, либо сочетать эти диски.

2. Компоненты, адресация и производительность жёстких дисков (HDD)

Жесткий диск — это постоянное устройство хранения, где данные хранятся и извлекаются с помощью быстро вращающихся дисков (пластин), покрытых магнитным материалом. Ключевые компоненты жесткого диска (HDD) — это пластина, шпиндель, головка чтения и записи, блок приводного рычага и плата контроллера. Операции ввода-вывода на жестком диске выполняются путем быстрого перемещения рычага между вращающимися плоскими пластинами, покрытыми магнитным материалом. Данные передаются между дисковым контроллером и магнитными пластинами через головку чтения и записи, подключенную к рычагу. Данные могут быть записаны и стерты с магнитных пластин любое количество раз.

* **Пластина.** Обычный жесткий диск состоит из одного или нескольких плоских круглых дисков, именуемых пластинами. Данные записываются на эти пластины в виде бинарных кодов (ноли и единицы). Набор вращающихся пластин запаян в корпус, который называется блок дисков с головками (HDA). Пластина — это жесткий круглый диск, покрытый магнитным материалом с обеих сторон (внешней и внутренней). Данные шифруются путем поляризации магнитной области или секторов поверхности диска. Данные можно записывать и считывать с обеих поверхностей пластины. Количество пластин и емкость хранения каждой пластины определяют общую емкость диска.
* **Шпиндель.** Шпиндель соединяет все пластины и подключается к приводу. Привод шпинделя вращается с постоянной скоростью. Пластина диска вращается со скоростью несколько тысяч оборотов в минуту (об/мин). Шпиндель может вращаться со скоростью 5400 об/мин, 7200 об/мин, 10 000 об/мин и 15 000 об/мин. Скорость пластины увеличивается с усовершенствованием технологии, хотя улучшение имеет свои границы.
* **Головка чтения/записи:** Головки чтения/записи, чтение и запись данных с/на пластины. Дисководы оснащены двумя головками чтения/записи на каждой пластине, по одной с каждой стороны пластины. Головка чтения/записи изменяет магнитную поляризацию на поверхности пластины во время записи данных. Во время чтения данных головка определяет магнитную поляризацию на поверхности пластины. При чтении и записи головка чтения/записи распознает магнитную поляризацию и никогда не касается поверхности пластины. Во время вращения шпинделя между головками чтения/записи и пластинами появляется микроскопический воздушный промежуток, известный как высота зависания головки. Воздушный промежуток исчезает когда шпиндель перестает вращаться, а головка чтения/записи находится в состоянии покоя в специальной зоне на пластине около шпинделя. Эта зона называется зоной парковки. Зона парковки покрыта смазкой для уменьшения трения между головкой и пластиной. Логика дискового накопителя обеспечивает перемещение головок в зону парковки до того, как они коснутся поверхности. Если диск неисправен и головка чтения/записи случайно касается поверхности пластины за пределами зоны парковки, происходит разрушение головки. В случае разрушения головки магнитное покрытие пластины царапается и может повредить головку чтения/записи. Обычно разрушение головки приводит к потере данных.
* **Блок приводного рычага.** Головки чтения/записи монтируются на блок приводного рычага, который позиционирует головку чтения/записи на пластине в том месте, где нужно записать или прочитать данные. Головки чтения/записи для всех пластин дисковода подключены к одному блоку приводного рычага и перемещаются между пластинами одновременно.
* **Плата контроллера диска.** Контроллер — это печатная плата, смонтированная в нижней части дискового накопителя. Он состоит из микропроцессора, внутренней памяти, микросхем и встроенного ПО. Встроенное ПО контролирует питание, которое подается на привод шпинделя, а также контролирует скорость двигателя. Контроллер управляет связью между диском и вычислительной системой. Кроме того, он контролирует операции чтения/записи, перемещая приводной рычаг и переключая головки чтения/записи, а также оптимизирует доступ к данным.

**Адресация**

Данные записываются на диск в виде *дорожек* — концентрических колец на пластине вокруг шпинделя. Дорожки пронумерованы с наружного края пластины (начиная с нуля). Количество *дорожек на дюйм* (TPI) пластины (или *плотность дорожек*) измеряет, насколько плотно дорожки расположены на пластине.

Каждая дорожка разделена на более мелкие единицы — *сектора*. Сектор — это наименьшая единица системы хранения с индивидуальной адресацией. Структура дорожки и сектора записывается на пластину производителем диска с использованием операции форматирования низкого уровня. Количество секторов на дорожке может быть разным в зависимости от типа диска. В зависимости от физических размеров и плотности записи, на пластине могут быть тысячи дорожек.

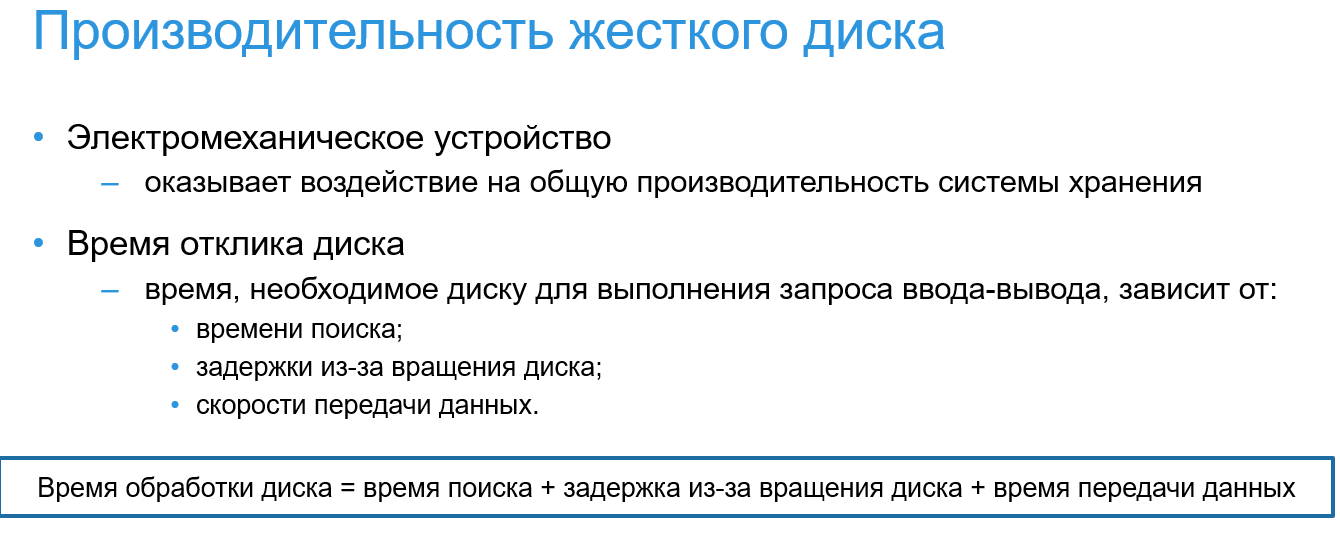
Обычно сектор содержит 512 байт пользовательских данных, хотя некоторые диски могут быть отформатированы с большими размерами сектора. Помимо пользовательских данных в секторе также хранится другая информация: номер сектора, номер головки или пластины, а также номер дорожки. С помощью этой информации контроллер размещает данные на диске.

Цилиндр — это набор идентичных дорожек на обеих поверхностях каждой пластины диска. Размещение головок чтения/записи указывается в номере цилиндра, а не дорожки.

Раньше диски использовали физические адреса, состоящие из номера *цилиндра, головки и сектора (CH*S*)* для ссылки на конкретные местоположения на диске, а ОС нужно было уведомлять о геометрии каждого используемого диска. *Адресация логического блока* (LBA) упростила адресацию использованием линейного адреса для доступа к физическим блокам данных. Дисковый контроллер переводит адрес LBA в адрес CHS, а вычислительной системе нужно знать только размер дискового накопителя относительно количества блоков. Установлено соответствие логических блоков с физическими секторами в отношении 1:1.

На этом слайде диск имеет восемь секторов на каждой дорожке, шесть головок и четыре цилиндра. Таким образом, общее количество блоков: 8 × 6 × 4 = 192, а номер блока от 0 до 191. Каждый блок имеет свой уникальный адрес.

При условии, что сектор содержит 512 байт, диск на 500 Гбайт с форматированной емкостью 465,7 Гбайт имеет 976 000 000 избыточных блоков.



*Время поиска* (также именуемое *время доступа*) описывает время, необходимое для позиционирования головок чтения/записи на пластине с радиальным движением (перемещение по радиусу пластины). Другими словами, это время, необходимое для позиционирования и установки рычага и головки на правильной дорожке. Таким образом, чем меньше времени занимает поиск, тем быстрей проходит операция ввода-вывода. Производители дисков указывают такие характеристики времени поиска:

* **Полный оборот.** Это время, необходимое для перемещения головки чтения/записи по всей ширине диска, от внутренней дорожки до внешней.
* **Среднее время поиска.** Это среднее время, необходимое головке чтения/записи для перемещения с одной произвольной дорожки на другую. Обычно оно указывается как время для совершения трети полного оборота.
* **Переход с дорожки на дорожку.** Это время, необходимое головке чтения/записи для перемещения между соседними дорожками.

Каждая из этих характеристик измеряется в миллисекундах (мс). Время поиска диска обычно указывается его производителем. Среднее время поиска для современного диска обычно составляет от 3 до 15 мс. Время поиска влияет на операции ввода-вывода произвольных дорожек больше, чем соседних дорожек. Чтобы сократить время поиска, данные можно записывать только в подраздел доступных цилиндров. В результате полезная емкость будет меньше фактической емкости диска. Например, дисковый накопитель на 500 Гбайт установлен на использование только первых 40 процентов цилиндров и в результате рассматривается как диск на 200 Гбайт. Это известно как *«короткий ход»* диска.

Для доступа к даннымприводной рычаг перемещает головку чтения/записи над пластиной к конкретной дорожке, в то время как пластина вращается для позиционирования необходимого сектора под головкой чтения/записи. Время, необходимое пластине для вращения и позиционирования данных в головке чтения/записи, называется *задержка из-за вращения диска*. Эта задержка зависит от скорости вращения шпинделя и измеряется в миллисекундах. Средняя задержка из-за вращения диска равна половине времени, необходимого для полного вращения. Аналогично времени поиска, задержка из-за вращения диска больше влияет на чтение/запись произвольных секторов диска, а не тех же операций в соседних секторах.

Средняя задержка из-за вращения диска приблизительно составляет 5,5 мс для диска на 5400 об/мин и около 2 мс для диска на 15 000 об/мин, как показано ниже.

Средняя задержка из-за вращения диска для диска на 15 000 об/мин (30 0000/15 000) = 2 30 000/15 000 = 2 мс

3. Компоненты, адресация и производительность твердотельных дисков

Твердотельные диски — это устройства хранения данных, содержащие энергонезависимую флэш-память. Твердотельные диски превосходят механические жесткие диски с точки зрения производительности, энергопотребления и доступности. Эти диски особенно хорошо подходят для приложений с малой задержкой, требующих стабильного, быстрого (менее 1 мс) времени отклика при чтении/записи.

Внутренняя архитектура аппаратного обеспечения твердотельного диска состоит из интерфейса ввода-вывода, контроллера и накопителя.

Интерфейс ввода-вывода позволяет подключить разъемы для питания и линии передачи данных к твердотельным дискам. Твердотельные диски обычно поддерживают стандартные разъемы, такие как SATA, SAS или FC.

Контроллер включает в себя контроллер дисков, ОЗУ и энергонезависимую память (NVRAM). Контроллер дисков управляет всеми функциями диска. Твердотельные диски поддерживают множество функций, таких как шифрование и объединение операций записи. Энергонезависимая память (NVRAM) используется для хранения оперативного ПО и данных твердотельного диска. Не во всех твердотельных дисках есть отдельная память NVRAM. В некоторых моделях программы и данные хранятся в общей емкости накопителя. ОЗУ используется для управления считываемыми и записываемыми данными на твердотельном диске как кэш-памяти и для операционных программ и данных твердотельного диска. Часть ОЗУ диска, используемая для кэш-памяти контроллера улучшает общую производительность твердотельного диска. Накопитель, состоящий из наборов флэш-памяти, выполняет операции чтения быстрее, чем операции записи. ОЗУ диска используется для уменьшения количества операций записи и улучшения времени отклика диска.

# Адресация

Микросхемы твердотельной памяти могут иметь разную емкость, например, 32 или 4 Гбайт в каждой микросхеме. Однако все микросхемы памяти располагаются в одной и той же логической структуре — страницы и блоки.

На самом низком уровне твердотельный диск сохраняет биты. Восемь бит составляют байт. В то время, как обычные для механического жесткого диска 512 байт составляют сектор, твердотельные диски не имеют секторов. Твердотельные диски имеют подобный физический объект данных под названием страница. Как и в секторе механического жесткого диска, страница — наименьший объект, который можно прочитать или записать на твердотельный диск. В отличии от механических жестких дисков, у страниц нет стандартной емкости. Емкость страницы зависит от архитектуры микросхемы твердотельной памяти. Типичные значения емкости страницы — 4 Кбайт, 8 Кбайт и 16 Кбайт.

Блок твердотельного диска состоит из страниц. Один блок может содержать 32, 64 или 128 страниц. Стандартный размер блока — 32 страницы. Общая емкость блока зависит от размера страницы микросхемы твердотельного диска. Только целые блоки можно записать или стереть в микросхеме твердотельной памяти. Отдельные страницы могут быть читаемыми или недействительными(логическая функция). Для записи блока страницы собираются в полные блоки в кэш-памяти твердотельного диска, после чего записываются в объект блочной системы хранения.

Производительность

Твердотельные диски — это полупроводниковые устройства с произвольным доступом. Они имеют очень короткое время отклика по сравнению с жесткими дисками. В сочетании с несколькими параллельными каналами ввода-вывода во внутреннем хранилище, это обеспечивает твердотельным дискам лучшие характеристики производительности, чем у жестких дисков.

Производительность твердотельного диска зависит от типа доступа, состояния диска и продолжительности рабочей нагрузки. Твердотельный диск лучше всего выполняет произвольные операции чтения. Благодаря тщательно настроенным многопоточным произвольным рабочим нагрузкам ввода-вывода в средах хранения данных небольшими блоками твердотельные диски могут доставить данные за меньшее время с большей пропускной способностью, чем жесткие диски. Это связано с тем, что операции ввода-вывода при произвольном чтении обычно не могут обслуживаться алгоритмами упреждающего чтения на жестком диске или кэш-памятью чтения в системе хранения. Задержка при произвольном чтении напрямую связана с временем поиска жесткого диска. Для жестких дисков это — физическое перемещение головки чтения/записи диска для доступа к необходимой зоне. Поскольку твердотельные диски — устройства с произвольным доступом, они не обращают внимание на извлечение операций ввода-вывода, которые находятся в нескольких местах. В результате их время отклика на порядок меньше, чем время отклика жестких дисков.

Для операций ввода-вывода в больших блоках твердотельные диски используют все внутренние каналы ввода-вывода параллельно. Поскольку единичные последовательные потоки ввода-вывода на жестких дисках FC не сталкиваются с задержками при поиске и вращении (благодаря кэш-памяти системы хранения), единичные последовательные потоки ввода-вывода в больших блоках не демонстрируют значительных улучшений производительности в твердотельных дисках по сравнению с жесткими дисками FC. Однако в связи с повышенной параллельной работой приложения (поскольку добавляется больше потоков), нагрузка начинает напоминать рабочую нагрузку произвольных операций с большими блоками. В данном случае появляются задержки поиска и вращения, снижающие эффективность работы жесткого диска FC, но не влияющие на эффективность работы твердотельного диска.

Новый твердотельный диск или твердотельный диск с большим объемом неиспользованного пространства имеет лучшую производительность. Переполненным дискам понадобится больше времени для выполнения цикла чтение-изменение-запись. Твердотельные диски лучше всего подходят для рабочих нагрузок с кратковременными всплесками активности.

4. Описание методов реализации массивов RAID

RAID — это метод, в котором несколько дисковых накопителей соединяются в логический элемент или набор RAID, а данные записываются в блоки на дисках в наборе RAID. Массивы RAID защищают от потери данных в случае выхода из строя диска путем использования резервных дисков и контроля четности. Массивы RAID также помогают улучшать производительность системы хранения, поскольку операции чтения и записи обрабатываются одновременно из нескольких дисковых накопителей.

Массивы RAID обычно внедряются путем использования специального контроллера оборудования, находящегося в вычислительной системе или в системе хранения. Ключевые функции контроллера RAID — управление и контроль агрегирований дисков, перевод запросов ввода-вывода между логическими и физическими дисками и восстановление данных в случае сбоя диска.

Существует два метода реализации массивов RAID: аппаратный и программный. Оба метода имеют свои преимущества и недостатки.

***Программные RAID*** используют ПО на базе вычислительной системы для предоставления RAID функций и внедряются на уровне ОС. Программные реализации RAIDимеют преимущества в цене и простоте использования по сравнению с аппаратными RAID. Тем не менее, они имеют следующие ограничения:

* **Производительность.** Программные RAID оказывают воздействие на производительность всей системы. Это связано с дополнительными циклами ЦП, которые необходимы для выполнения вычислений RAID.
* **Поддерживаемая функциональность.** Программные RAID не поддерживают все уровни RAID.
* **Совместимость с ОС.** Программные RAID привязаны к ОС. Таким образом, обновления программных RAID или ОС необходимо проверять на совместимость. Это приводит к недостаточной гибкости в среде обработки данных.

**Аппаратные реализации RAID** обычно внедряются путем использования специального контроллера оборудования, находящегося в вычислительной системе или в системе хранения. Контроллерная карта RAID — это аппаратная реализация RAID на базе вычислительной системы, где в вычислительной системе устанавливается специальный контроллер RAID, а к нему подключаются диски. Производители также интегрируют контроллеры RAID в материнские платы. Контроллер RAID на базе вычислительной системы — неэффективное решение для среды центра обработки данных с большим количеством вычислительных систем. Внешний контроллер RAID — это оборудование RAID на базе системы хранения. Он работает как интерфейс между вычислительной системой и дисками. Он представляет тома хранения данных вычислительной системе, а вычислительная система управляет этими томами как физическими дисками. Ключевые функции контроллеров RAID:

* управление и контроль агрегирования дисков;
* перевод запросов ввода-вывода между логическими дисками и физическими дисками;
* восстановление данных в случае сбоя диска.

5. Описание трех методов RAID?

Три разных метода RAID, формирующие основу для определения различных уровней RAID — это распределение, зеркалирование и контроль четности. Эти методы определяют доступность данных и производительность набора RAID, а также относительную стоимость развертывания уровня RAID.

**Распределение** — это метод распространения данных между несколькими дисками (более одного) для их параллельного использования. Все головки чтения/записи работают одновременно, позволяя обрабатывать большее количество данных за меньший промежуток времени и увеличивая производительность по сравнению с чтением и записью с одного диска. В каждом диске из набора RAID определенное количество блоков дисков с непрерывной адресацией определяются как элемент *Размер элемента распределения* (также именуемый *глубина полосы*) описывает количество блоков в *элементе распределения* (представлены как «A1, A2, A3 и A4»). Это — максимальное количество данных, которые можно записать или прочитать с одного диска из набора при условии, что используемые данные начинаются в начале элемента распределения. Все элементы распределения в полосе имеют одинаковое количество блоков. Наличие элемента распределения меньшего размера означает, что данные разбиваются на более мелкие части при распределении по дискам. Размер полосы (представлен как А) — это величина, кратная размеру элемента распределения по количеству дисков данных в наборе RAID. Например, в распределенном наборе RAID из четырех дисков с размером элемента распределения 64 Кбайт, размер полосы будет 256 Кбайт(64 Кбайт x 4). Другими словами, A = A1 +A2 + A3 + A4. *Ширина полосы* обозначает количество элементов распределения данных в полосе. Распределенный RAID не предоставляет защиту данных, если не используется контроль четности или зеркалирование.

**Зеркалирование** — это метод, в котором те же данные хранятся на двух разных дисках, предоставляя две копии данных. В случае сбоя одного диска данные остаются на другом уцелевшем диске, а контроллер продолжит обслуживать запросы данных вычислительной системы с уцелевшего диска зеркалированной пары. При замене сбойного диска новым контроллер копирует данные из уцелевшего диска зеркалируемой пары.

***Контроль четности*** — это метод защиты распределенных данных от сбоя дискового накопителя без затрат на зеркалирование. Для сохранения четности (математической конструкции, позволяющей повторно создавать недостающие данные) добавляется дополнительный диск. Контроль четности — это метод резервирования, обеспечивающий защиту данных без сохранения целого набора продублированных данных. Вычисление четности — это функция контроллера RAID. Контрольная информация может храниться на отдельных выделенных дисках, либо распределяться по всем дискам в наборе RAID. Первые три диска на рисунке, отмеченные как *D1 — D3,* содержат данные. На четвертом диске, отмеченном как *P*, хранится контрольная информация, в данном случае — сумма элементов в каждой строке. Если происходит сбой данных на дисках, недостающее значение можно вычислить путем вычитания суммы остальных элементов из значения четности. Для большей простоты, вычисление четности представлено как арифметическая сумма данных. Однако вычисление четности — это *побитовая операция XOR*.

По сравнению с зеркалированием, внедрение четности значительно уменьшает затраты, связанные с защитой данных. Рассмотрим пример настройки контроля четности RAID с четырьмя дисками, где три диска содержат данные, а в четвертом находится контрольная информация. В этом примере контроля четности требуется только 33 процента дополнительного места на диске, в то время как для зеркалирования нужно 100 процентов дополнительного места. Однако есть некоторые недостатки использования контроля четности. Контрольная информация генерируется из данных на диске данных. Таким образом, четность повторно вычисляется каждый раз при внесении изменений в данные. Это повторное вычисление отнимает много времени и снижает производительность массива RAID.

В массивах RAID с контролем четности вычисление размера полосы не включает в себя элемент распределения четности. Например, в наборе RAID с контролем четности из четырех дисков с размером элемента распределения 64 Кбайт размер полосы будет 192 Кбайт (64 Кбайт x 3).

6. Описание часто используемых уровней RAID

Конфигурация **RAID 0** использует методыраспределения данных, где данные распределяются между всеми дисками в наборе RAID. Таким образом используется вся емкость хранения набора RAID. Для чтения данных контроллер собирает все элементы распределения. При увеличении количества дисков в наборе RAID улучшается производительность, поскольку одновременно можно записать или прочитать больше данных. RAID 0 — это хороший вариант для приложений, которым необходима большая пропускная способность операций ввода-вывода. Однако если этим приложениям необходима высокая доступность во время сбоев дисков, RAID 0 не сможет предоставить защиту и доступность данных.

**RAID 1** основывается на методе зеркалирования. В этой конфигурации RAID данные зеркалируются, чтобы обеспечить отказоустойчивость. Набор RAID 1 состоит из двух дисковых накопителей. Каждая запись выполняется на оба диска. Зеркалирование происходит незаметно для вычислительной системы. При сбое дисков воздействие на восстановление данных в RAID 1 самое малое среди всех реализаций RAID. Это связано с тем, что контроллер RAID использует зеркальный диск для восстановления данных. RAID 1 подходит для приложений, где требуется высокая доступность, а стоимость не ограничена.

Большинству центров обработки данных требуются резервирование данных и производительность массивов RAID. **RAID 1+0** сочетает преимущества производительности RAID 0 и резервирования RAID 1. Здесь используются методы зеркалирования и распределения и сочетаются их преимущества. Для этого типа массивов RAID требуется четное количество дисков (максимум 4).

RAID 1+0 также известен как RAID 10 (десять) или RAID 1/0. RAID 1+0 еще называют распределенным зеркалом. Основной элемент RAID 1+0 — это зеркалированная пара. Это означает, что данные сначала зеркалируются, а после обе копии данных распределяются по нескольким парам дисков в наборе RAID. При замене сбойного диска восстанавливается только зеркало. Иными словами, контроллер системы хранения использует уцелевший диск в зеркалированной паре для восстановления данных и непрерывной работы. Данные из уцелевшего диска копируются на диск для замены.

**RAID 3** распределяет данные для производительности и использует контроль четности для отказоустойчивости. Контрольная информация хранится на выделенном диске, чтобы данные можно было воссоздать в случае сбоя диска в наборе RAID. Например, в наборе из пяти дисков четыре используются для данных и один для контроля четности. Таким образом, необходимая общая емкость диска в 1,25 раз больше размера дисков данных. RAID 3всегдачитает и записывает целые полосы данных на все диски, поскольку диски работают параллельно. Выполнять запись частично, обновляя один из многих элементов в полосе, невозможно.

**RAID 5** — это универсальная реализация RAID. Она похожа на RAID 4, поскольку использует распределение. Диски (элементы распределения) также доступны по отдельности. Разница между RAID 4 и RAID 5 состоит в расположении четности. В RAID 4 контроль четности записывается на выделенный диск, создавая узкое место при записи на диск четности. В RAID 5 контроль четности распределяется по всем дискам во избежание возникновения узкого места при записи выделенного диска четности.

**RAID 6** работает таким же образом, как и RAID 5, за исключением того, что RAID 6 содержит второй элемент четности для сохранения данных в случае сбоя обоих дисков в наборе RAID. Таким образом, для реализации RAID 6 требуется как минимум четыре диска. RAID 6 распределяет контроль четности по всем дискам. Издержка записи (рассматривается далее в этом модуле) в RAID 6 больше, чем в RAID 5. Следовательно, выполнение записи в RAID 5 происходит лучше, чем в RAID 6. Операция восстановления в RAID 6 может занять больше времени, чем в RAID 5 в связи с наличием двух наборов четности.

7. Описание воздействия массивов RAID на производительность?

При выборе типа массивов RAID важно учесть воздействие на производительность диска и IOPS приложения. В зеркалированной и в четной конфигурации RAID каждая операция записи приводит к большему количеству служебных данных, возникающих при выполнении операций ввода-вывода для дисков. Это явление называется *издержки записи*. В реализации RAID 1 каждая операция записи должна выполняться на двух дисках, сконфигурированных как зеркалированная пара. В то же время в реализации RAID 5 операция записи может оформляться как четыре операции ввода-вывода. При выполнении операций ввода-вывода на диск, сконфигурированный с RAID 5, контроллеру необходимо читать, повторно вычислять и записывать сегмент четности для каждой операции записи данных.

На этом слайде показана одна операция записи на RAID 5, где содержится группа из пяти дисков. Четность (P) на контроллере вычисляется следующим образом:

**Cp = C1 + C2 + C3 + C4 (операции XOR)**

Каждый раз при выполнении контроллером записи операции ввода-вывода необходимо вычислить контроль четности путем чтения старого контроля четности (Cp старый) и старых данных (C4 старый) с диска, что означает две операции ввода-вывода при чтении. После этого новый контроль четности (Cp новый) вычисляется следующим образом:

**Cp новый = Cp старый – C4 старый + C4 новый (операции XOR)**

После вычисления нового контроля четности контроллер завершает операции ввода-вывода при чтении путем записи новых данных и нового контроля четности на диски, что составляет две операции ввода-вывода при записи. Таким образом, контроллер выполняет две операции чтения диска и две операции записи диска для каждой операции записи, а издержки записи равны 4.

В RAID 6 с двойным контролем четности для записи диска необходимо выполнить три операции чтения: две для контроля четности и одну для данных. После вычисления обоих новых процессов контроля четности контроллер выполняет три операции записи: две для контроля четности и одну для операций ввода-вывода. Следовательно, при реализации RAID 6 контроллер выполняет шесть операций ввода-вывода для каждого ввода-вывода при записи, а издержки записи равны 6.

8. Сравнение уровней RAID исходя из стоимости, производительности и

защиты?



9. Методы доступа к данным?

Доступ и хранение данных выполняется приложениями с использованием базовой инфраструктуры. Ключевые компоненты этой инфраструктуры — это ОС (или файловая система), возможность подключения и система хранения. Контроллерная карта вычислительной системы получает доступ к устройствам хранения данных, используя стандартные протоколы (например, IDE/ATA, SCSI или Fibre Channel (FC)). IDE/ATA и SCSI часто используются в небольших и персональных вычислительных средах для доступа к внутренней системе хранения. Протоколы FC и iSCSI используются для доступа к данным из внешних устройств хранения данных (или подсистем). Внешние устройства хранения можно подключить к вычислительной системе напрямую или через сеть хранения. При подключении системы хранения напрямую к вычислительной системе, получается система хранения данных, подсоединенная непосредственно к серверу (DAS).

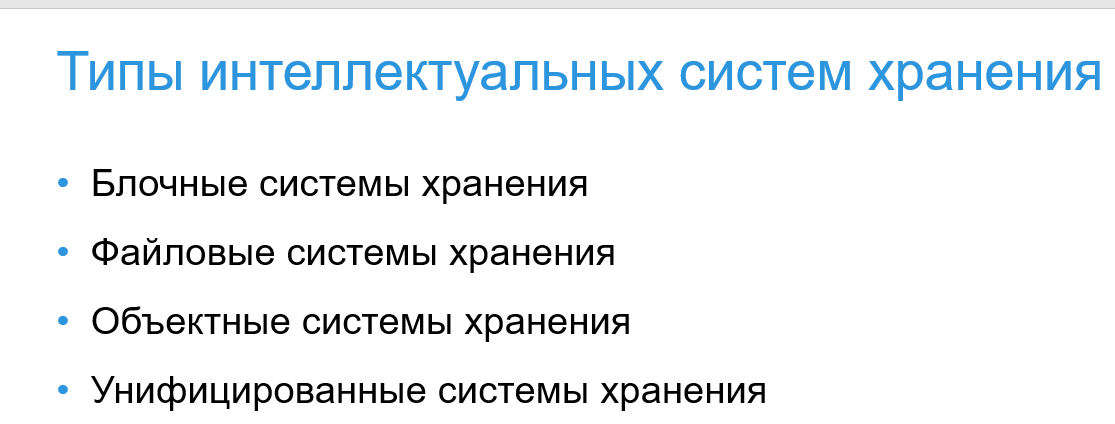
Можно получать доступ к данным через сеть одним из следующих способов: на уровне блоков, на уровне файлов или на уровне объектов. Обычно приложение запрашивает данные из файловой системы (или ОС), указывая имя и местоположение файла. Файловая система имеет два компонента: пользовательский компонент и компонент хранения данных. Пользовательский компонент файловой системы выполняет такие функции, как управление иерархией, присвоение имен и контроль доступа пользователей. Компонент хранения данных устанавливает соответствия файлов к физическому расположению на устройстве хранения данных. Файловая система устанавливает соответствия для атрибутов файлов к адресу LBA данных и отправляет запрос устройству хранения данных. Устройство хранения данных конвертирует адрес LBA в адрес CHS и выбирает данные.

При доступе на уровне блоков файловая система создается на базе вычислительных систем, а доступ к данным осуществляется через сеть на уровне блоков. В этом случае для вычислительной системы назначаются необработанные диски или логические тома, чтобы создать файловую систему.

При доступе на уровне файлов файловая система создается на отдельном файловом сервере или на стороне СХД, а запрос на уровне файлов отправляется через сеть. Поскольку доступ к данным происходит на уровне файлов, этот метод имеет большие издержки по сравнению с доступом к данным на уровне блоков.

Доступ на уровне объектов — это интеллектуальная эволюция, где доступ к данным происходит через сеть с помощью автономных объектов с уникальными идентификаторами. При этом типе доступа пользовательский компонент файловой системы находится в вычислительной системе, а компонент хранения данных — в системе хранения.

10. Типы интеллектуальных систем хранения?



11. Вертикально и горизонтально масштабируемые архитектуры?

Интеллектуальная система хранения может создаваться на основе вертикально или горизонтально масштабируемой архитектуры.

Вертикально масштабируемая архитектура хранения предоставляет возможность масштабировать емкость и производительность одной системы хранения на основе требований. Вертикальное масштабирование системы хранения предполагает обновление или добавление контроллеров и системы хранения. Эти системы имеют фиксированную емкость, что ограничивает их масштабируемость, а производительность также начинает снижаться при достижении предела емкости.

Горизонтально масштабируемая архитектура предоставляет возможность увеличения емкости путем простого добавления узлов в кластер. Узлы можно быстро добавлять в кластер, если необходимо получить большую производительность и емкость, при этом не вызывая простой. Это предоставляет гибкие возможности по использованию множества узлов со средней производительностью и доступностью для создания системы с улучшенными совокупными показателями производительности и доступности. Горизонтально масштабируемая архитектура объединяет ресурсы в кластер и распределяет рабочую нагрузку между всеми узлами. Таким образом обеспечивается линейное улучшение производительности, поскольку в кластер добавляется больше узлов.