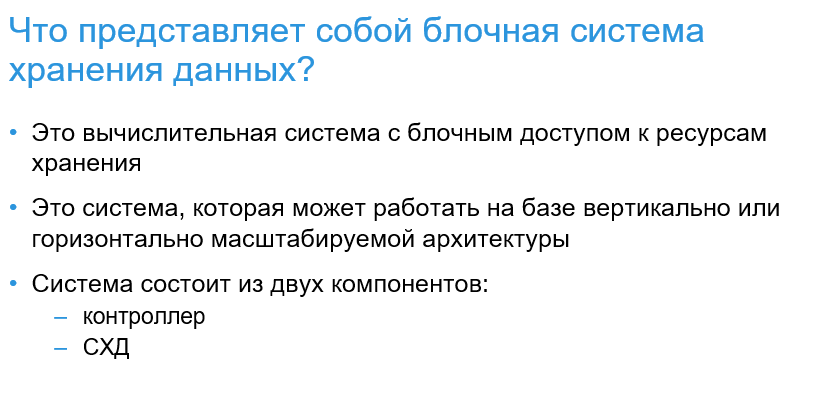
# 1. Компоненты блочной системы хранения (система с

# блочным доступом к ресурсам хранения)?



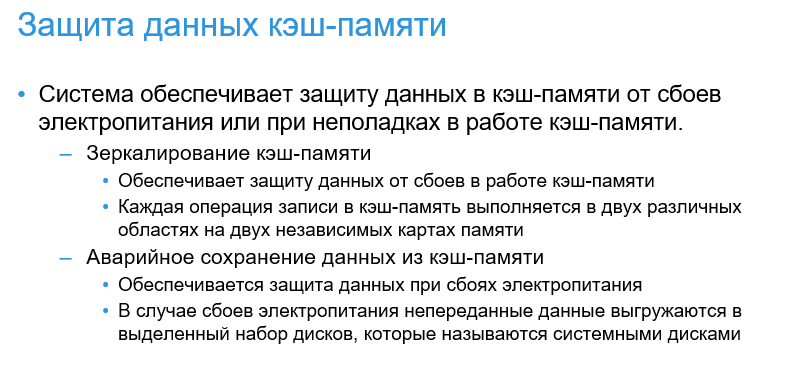
Контроллер блочной системы хранения состоит из трех ключевых компонентов: *клиентский интерфейс*, *кэш-память и серверная часть*. Запрос ввода-вывода передается из вычислительной системы через внешний порт и обрабатывается с помощью кэш-памятии серверной части. Это позволяет хранить данные в СХД и извлекать ихизнее. Запрос чтения может обслуживаться непосредственно из кэш-памяти, если в ней обнаружены запрошенные данные. В современных системах хранения клиентский интерфейс, кэш-память и серверная часть обычно интегрированы в единую панель (которую называют *процессором СХД* или *контроллером системы хранения данных)*.

# 2. Алгоритмы интеллектуального кэширования?

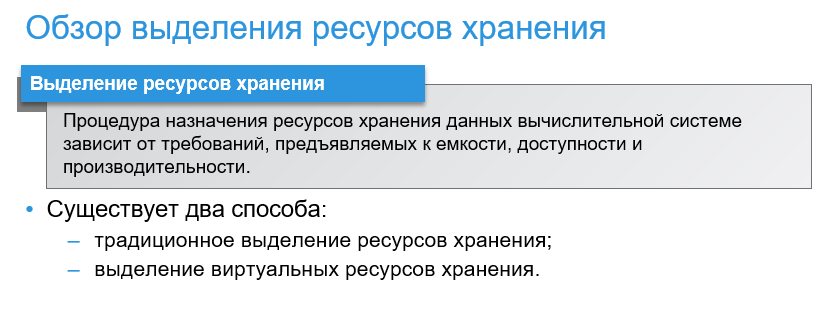
Самые распространенные алгоритмы приведены в следующем списке.

* **LRU (элементы с самой большой давностью использования).**Это алгоритм, который постоянно отслеживает доступ к данным и определяет страницы кэш-памяти, доступ к которым не выполнялся продолжительное время. LRU освобождает эти страницы или помечает их для повторного использования. Этот алгоритм основан на допущении, что данные, доступ к которым не выполнялся в течение некоторого времени, не будут запрашиваться вычислительной системой. Однако если страница содержит записанные данные, которые еще не были переданы в систему хранения, то перед повторным использованием страницы данные сначала записываются в систему хранения.
* **MRU (последние по времени использования элементы).**Этот алгоритм работает по противоположному принципу — страницы, доступ к которым осуществлялся недавно, освобождаются или помечаются для повторного использования. Этот алгоритм основан на допущении, что данные, доступ к которым выполнялся недавно, не будут востребованы в течение некоторого времени.

# 3. Механизм защиты данных кэш-памяти?



# 4. Традиционное и виртуальное выделение ресурсов?



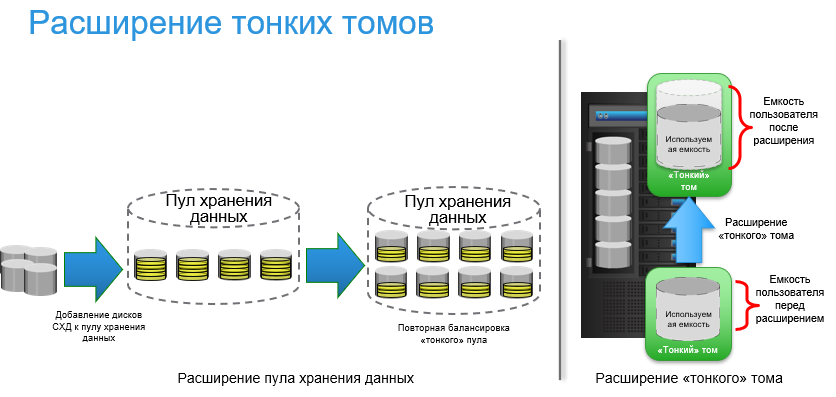
При **традиционном** выделении ресурсов хранения физические диски СХД логически объединяются в группы, использующие необходимый уровень RAID, формируя таким образом набор, называемый набором RAID. Количество дисков в наборе RAID и на уровне RAID определяют доступность, емкость и производительность набора RAID. Настоятельно рекомендуется создавать набор RAID из дисков подобного типа с аналогичными показателями скорости и емкости, чтобы гарантировать максимальную полезную емкость, надежность и согласованную производительность. Например, если в наборе RAID используются диски различной емкости, то на каждом из дисков, входящих в набор, будет использован только объем, соответствующий емкости самого маленького диска. Оставшаяся емкость дисков большего размера не будет использоваться. Аналогичным образом, объединение дисков с более высокой скоростью с дисками с более низкой скоростью снижает общую производительность набора RAID.

**Виртуальное** *выделение ресурсов* позволяет создавать и предоставлять том с емкостью, превышающей выделенную физическую емкость в системе хранения. Том, созданный с помощью виртуального выделения ресурсов, называется *«тонким» томом*, в отличие от традиционных томов.

При создании «тонких» томов и их подключении к вычислительной системе им не требуется выделять физические ресурсы хранения в полном объеме. После этого физические ресурсы хранения выделяются для вычислительной системы из общего пула физической емкости по требованию. *Общий пул* состоит из физических дисков СХД. Общий пул в схеме виртуального выделения ресурсов аналогичен набору RAID, который представляет собой группу дисков, на базе которых создаются тома. Как и набор RAID, общий пул поддерживает один уровень защиты RAID. Однако в отличие от набора RAID, общий пул может содержать большое количество дисков. Общие пулы могут быть однородными (содержать диски одного типа) или гетерогенными (содержать диски разных типов, например твердотельные диски, диски Fibre Channel, SAS и SATA).

Виртуальное выделение ресурсов позволяет более эффективно распределять ресурсы хранения между вычислительными системами. Виртуальное выделение ресурсов также делает возможным избыточное выделение, при котором суммарная емкость, предоставленная вычислительным системам, больше фактически доступной в системе хранения. И общий пул, и «тонкий» том можно расширять без прерывания работы системы по мере роста потребностей вычислительных систем в ресурсах хранения. В системе хранения можно создать несколько общих пулов, а общий пул может совместно использоваться несколькими «тонкими» томами.

# 5. Расширение томов?

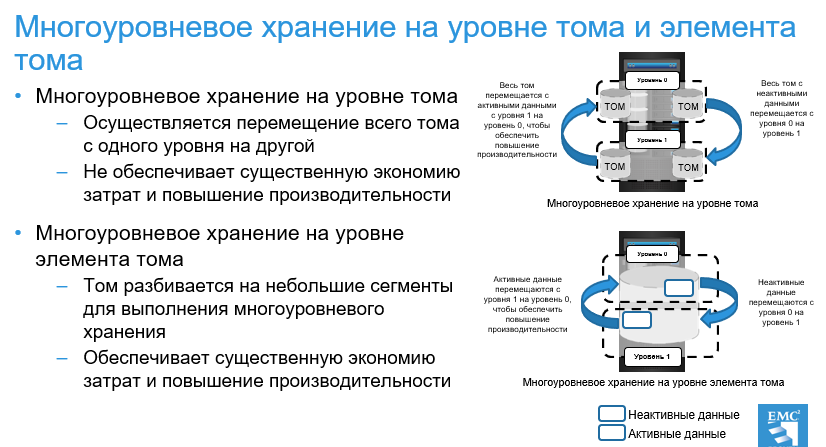


Пул хранения данных включает в себя физические диски, которые предоставляют физические ресурсы хранения, используемые «тонкими» томами. Пул хранения данных создается путем определения набора дисков и типа массивов RAID для этого пула. Затем из этого пула создаются «тонкие» тома (подобно тому, как традиционные тома создаются в наборе RAID). Все «тонкие» тома, созданные из пула, совместно используют ресурсы хранения данных этого пула. Добавление дисков к пулу хранения данных увеличивает емкость, доступную всем «тонким» томам этого пула. Диски можно добавлять в пул хранения данных непосредственно во время работы с инфраструктурой. Пул освобождает выделенную емкость при уничтожении «тонких» томов.

При развертывании пула хранения данных в случае внезапного добавления новых пустых дисков к уже относительно полным дискам может произойти дисбаланс данных. Этот дисбаланс устраняется путем выполнения автоматического однократного перемещения данных, называемого восстановлением баланса. Восстановление баланса пула хранения данных — это технология, позволяющая автоматически перемещать экстенты (минимальный объем физической емкости СХД, который назначается тому из пула) на физические диски СХД через весь пул, когда в пул добавляются новые диски. Восстановление баланса пула хранения данных распределяет данные по всем дискам (существующим и новым) в пуле хранения данных. Это позволяет поровну распределять данные по всем физическим дискам в пуле хранения данных, гарантируя равномерное использование каждого диска в пуле. После увеличения емкости пула хранения данных можно расширить емкость существующих томов

# 6. Многоуровневое хранение на уровне тома и элемента

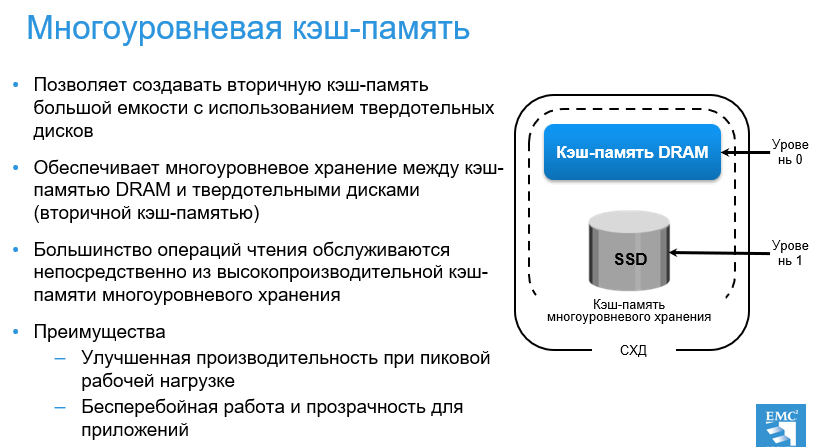
# тома?



Многоуровневое хранение данных — это технология создания иерархии различных типов систем хранения (уровней). Она позволяет хранить нужные данные на необходимом уровне в зависимости от требований к обслуживанию и с минимальными затратами. Каждый уровень имеет различные степени защиты данных, производительности и затрат. Например, высокопроизводительные твердотельные диски (SSD) или диски Fibre Channel можно настроить как систему хранения 1 уровня, чтобы хранить часто используемые данные, а недорогие диски SATA можно использовать как систему хранения 2 уровня, чтобы хранить на них реже используемые данные. Хранение часто используемых данных на твердотельных дисках и дисках Fibre Channel повышает производительность приложений. Перемещение реже используемых данных на диски SATA может высвободить место на высокопроизводительных дисках и сократить затраты на хранение данных. Это перемещение данных регламентируется определенными политиками многоуровневого хранения. Политика многоуровневого хранения может зависеть от частоты доступа и других параметров. Например, если политика предписывает «Переместить данные, которые не использовались в течение последних 30 минут, на нижний уровень», то все файлы, удовлетворяющие этому условию, будут перемещаться на нижний уровень.

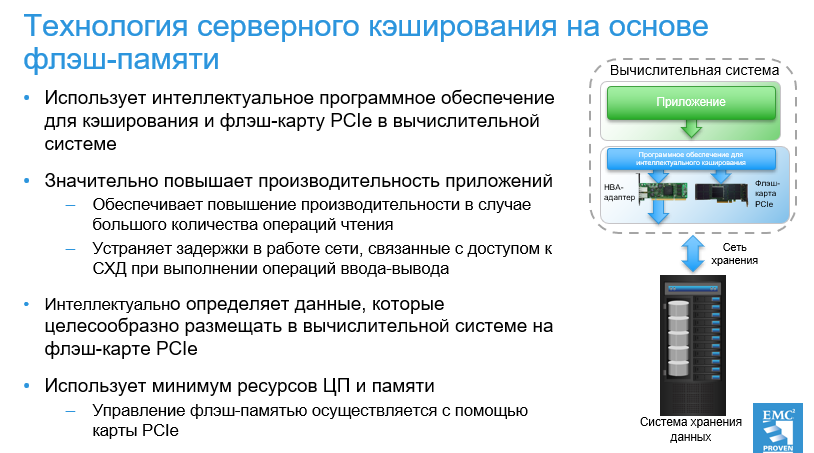
Процесс перемещения данных с уровня одного типа на уровень другого типа обычно автоматизирован. При автоматизированном многоуровневом хранении данных рабочие нагрузки приложений отслеживаются в упреждающем режиме; активные данные автоматически перемещаются на более производительный уровень, а неактивные данные — на уровень с невысокой производительностью, но повышенной емкостью. Перемещение данных между уровнями выполняется без прерывания работы.

Существует три технологии многоуровневого хранения данных, реализованные в блочной системе хранения: многоуровневое хранение на уровне тома и элемента тома, многоуровневая кэш-память и серверное кэширование на основе флэш-памяти.

7. Многоуровневая кэш-память? 

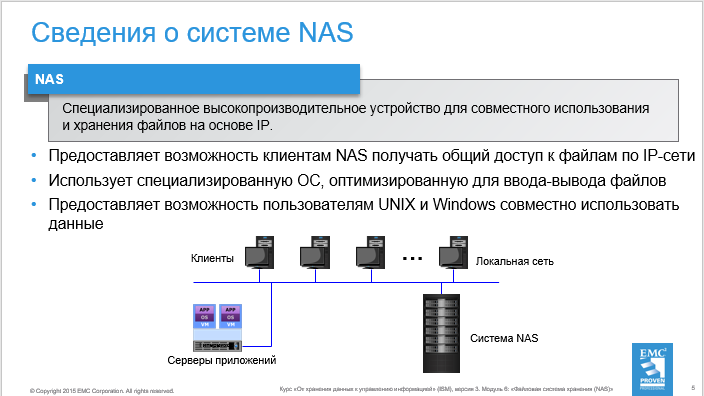
Многоуровневый подход также реализуется на уровне кэш-памяти. Большая кэш-память в системе хранения повышает производительность, сохраняя большой объем часто используемых данных в кэш-памяти. В результате большинство операций чтения выполняется напрямую из кэш-памяти. Однако настройка объемной кэш-памяти в системе хранения влечет за собой большие затраты. Как вариант, можно увеличить размер кэш-памяти, используя твердотельные диски в системе хранения. В многоуровневой кэш-памяти твердотельные диски используются для создания вторичной кэш-памяти большой емкости, что делает возможным многоуровневое хранение данных между DRAM (основная кэш-память) и твердотельными дисками (вторичная кэш-память). Серверный кэш на базе флэш-памяти — это еще один уровень кэш-памяти, в рамках которого на сервере устанавливается флэш-карта кэш-памяти, чтобы еще больше повысить производительность приложений.

# 8. Серверное кэширование на основе флэш-памяти?

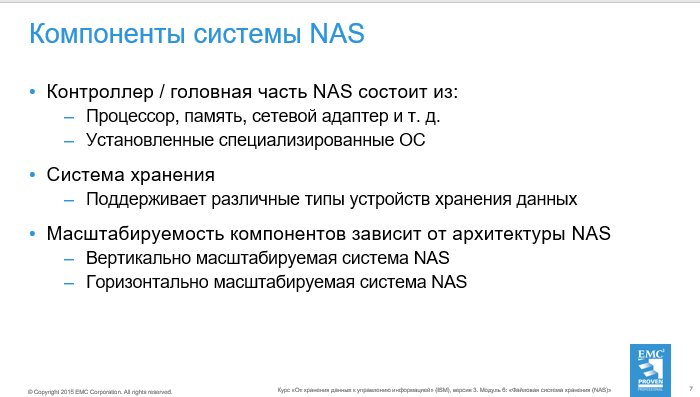


В технологии серверного кэширования на основе флэш-памяти применяется программное обеспечение для интеллектуального кэширования и флэш-карта PCI Express (PCIe), установленные в вычислительной системе. Это существенно повышает производительность приложений за счет сокращения задержки и повышения пропускной способности. Технология серверного кэширования на основе флэш-памяти работает как в физических, так и в виртуальных средах и обеспечивает повышение быстродействия для рабочих нагрузок с интенсивным чтением. В рамках этой технологии ресурсы ЦП и памяти вычислительной системы используются минимально, поскольку функции управления флэш-памятью делегированы карте PCIe.

Технология позволяет с помощью интеллектуальных алгоритмов определять, какие данные целесообразно разместить в вычислительной системе, на флэш-карте PCIe, поближе к приложению. Это позволяет избежать задержек в работе сети, связанных с доступом к системе хранения при выполнении операций ввода-вывода. Таким образом, для доступа к данным, которые чаще всего используются приложением, задействуются ресурсы карты PCIe, а не внутреннего хранилища. Благодаря этому система хранения может выделить большую вычислительную мощность другим приложениям.



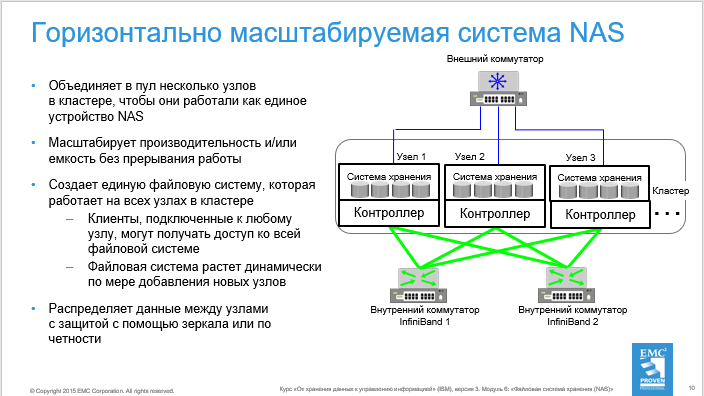
1. Компоненты NAS?



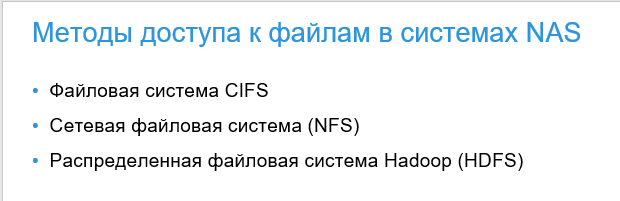
2. Архитектура NAS?

Бывает вертикально масштабируемая и горизонтально масштабируемая



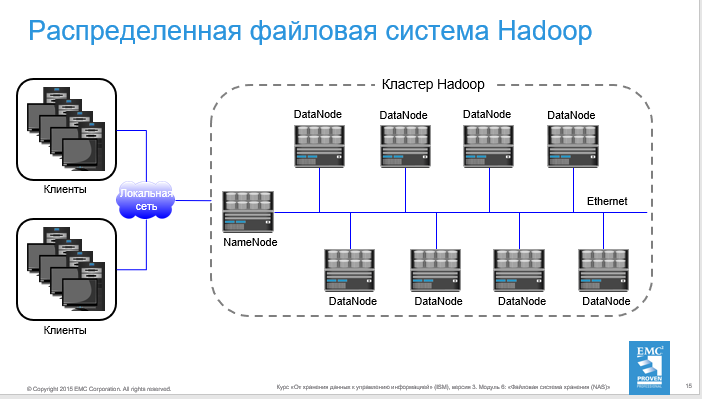


3. Методы доступа к файлам в системе NAS?



Файловая система CIFS — это клиент-серверный протокол приложений, предоставляющий клиентским программам возможность делать запросы файлов или сервисов на удаленных компьютерах по протоколу TCP/IP. Это публичная (или открытая) версия протокола блоков сообщений сервера (SMB). Протокол CIFS предоставляет удаленным клиентам возможность получать доступ к файлам, размещенным на сервере. CIFS обеспечивает общий доступ к файлам совместно с другими клиентами благодаря использованию специальных блокировок. В файловой системе CIFS имена файлов кодируются с помощью символов Unicode.

Network File System (NFS) — это клиент-серверный протокол для организации общего доступа к файлам, который часто используется в системах UNIX. NFS изначально базировался на не требующем соединения протоколе User Datagram Protocol (UDP). Он использует машинно-независимую модель для представления данных пользователя. Кроме того, этот протокол использует удаленный вызов процедур (RPC) в качестве метода межпроцессного взаимодействия между двумя компьютерами.



4. Операции ввода-вывода в системе NAS?

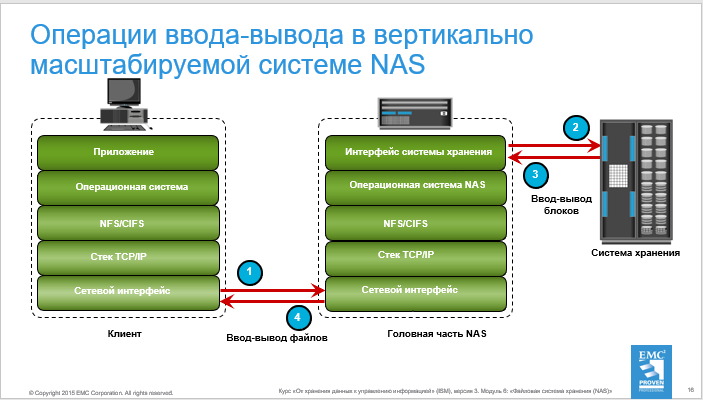
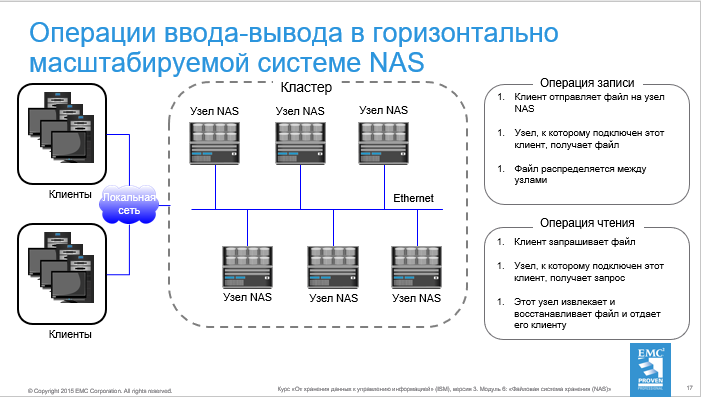


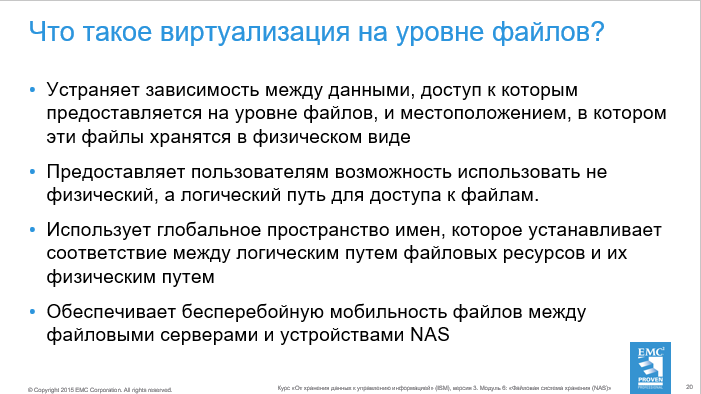
Рисунок на слайде иллюстрирует операцию ввода-вывода в вертикально масштабируемой системе NAS. Процесс обработки операций ввода-вывода в вертикально масштабируемой среде NAS происходит следующим образом:

1. Инициатор запроса (клиент) упаковывает запрос ввода-вывода в пакет TCP/IP и отправляет его через сетевой стек. Система NAS получает этот запрос из сети.
2. Система NAS преобразует этот запрос ввода-вывода в запрос к соответствующей физической системе хранения, который представляет собой операцию ввода-вывода на уровне блоков, и затем выполняет эту операцию в физической системе хранения.
3. Когда система NAS получает данные от системы хранения, она обрабатывает их и повторно упаковывает в ответ соответствующего файлового протокола.
4. Система NAS снова упаковывает этот ответ в пакет TCP/IP и оправляет его по сети клиенту.



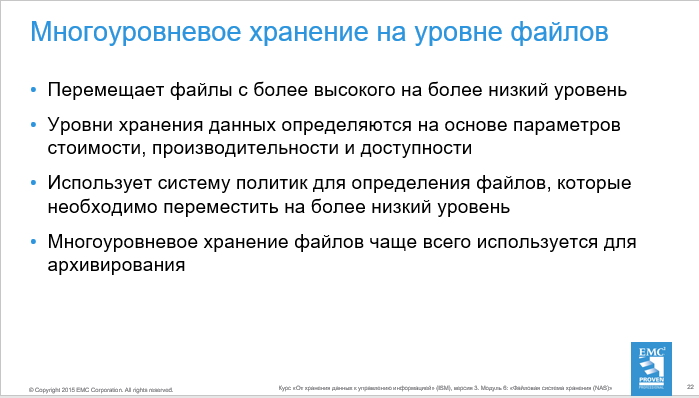
5. Виртуализация на уровне файлов?

Она создает логический пул ресурсов хранения, предоставляя пользователям возможность использовать не физический, а логический путь для доступа к файлам.

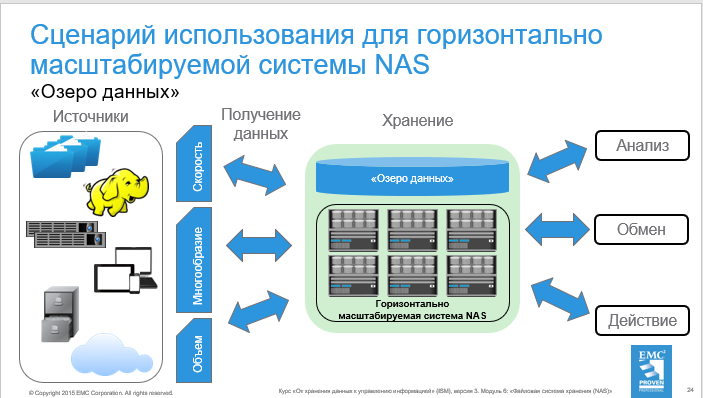


6. Многоуровненое хранение?

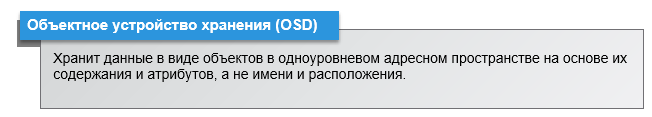
Многоуровневое хранение работает по принципу иерархического управления системой хранения (HSM). По мере роста объемов неструктурированных данных в среде NAS организации развертывают многоуровневую среду хранения данных. Такая среда позволяет оптимизировать основную систему хранения с точки зрения производительности, а вторичную СХД — с точки зрения емкости и стоимости.

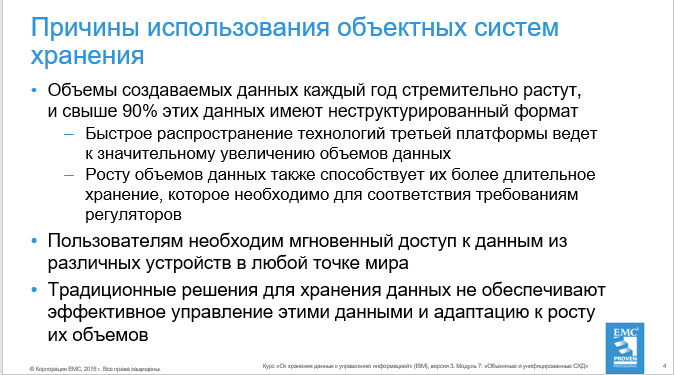


7. Сценарий использования NAS?



8. Причины использования устройств OSD?





9. Сравнение иерархической файловой системы и одноуровневого адресного пространства?



10. Ключевые компоненты OSD?

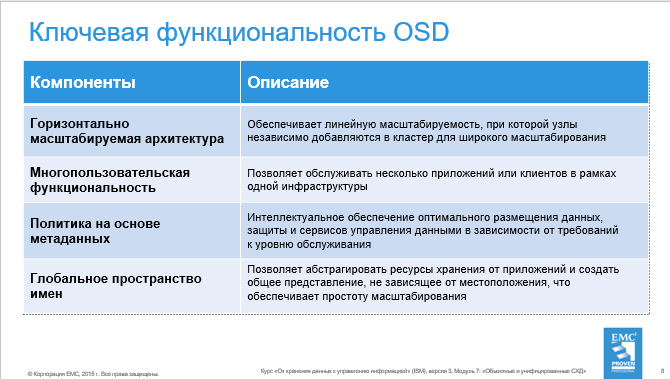
Стандартная система OSD состоит из трех ключевых компонентов: узлов (контроллеров), внутренней сети и СХД.

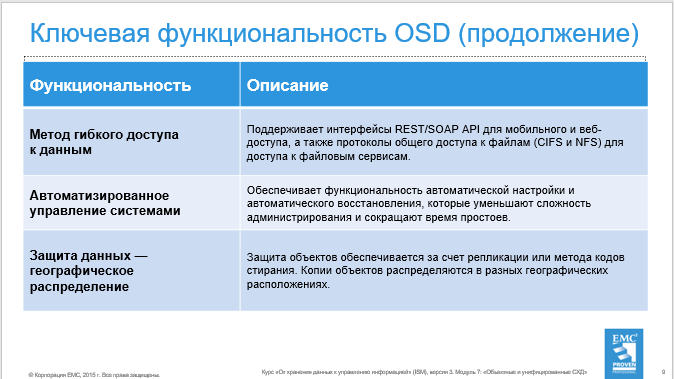
Узел — это сервер, на котором работает операционная среда OSD и который обеспечивает работу сервисов, предназначенных для сохранения и извлечения данных, а также управления ими в системе.

Узлы OSD подключаются к СХД по внутренней сети. Внутренняя сеть обеспечивает подключение по модели «узел-узел» и «узел-СХД».

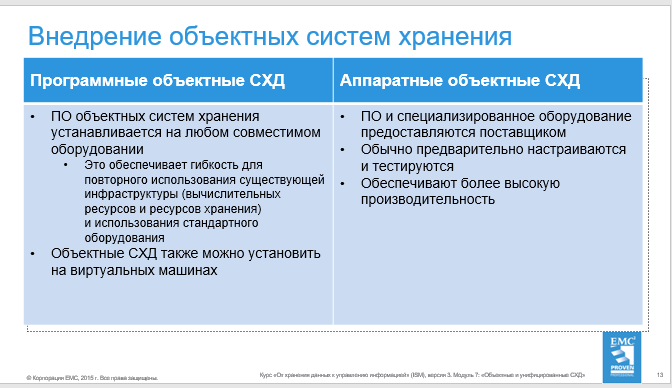
OSD обычно использует для хранения объектов недорогие диски с высокой плотностью записи данных. Если требуется больше дискового пространства, к системе можно добавить дополнительные диски.

11. Ключевая функциональность OSD?





12. Внедрение объектных систем хранения?



13. Процесс сохранения и извлечения данных в объектной системе хранения?





14. Шлюз объектной системы хранения?

Из-за отсутствия стандартизованных интерфейсов API для облачных систем хранения устройство-шлюз стало критически важным компонентом для внедрения облака.

