

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем
(код и наименование направления подготовки, специальности)

WEB-дизайн и мультимедиа
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Алгоритм подбора оборудования для систем хранения данных»

Обучающийся

Н.В. Шимон
(Инициалы Фамилия)

(личная подпись)

Руководитель

к.т.н., доцент, О.В. Аникина

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

М.М. Бажутина

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2022

АННОТАЦИЯ

Темой бакалаврской работы является «Алгоритм подбора оборудования для систем хранения данных».

Выпускная работа состоит из введения, трёх глав, заключения, таблиц, списка литературы, включая зарубежные источники, и прототипа алгоритма для подбора оборудования систем хранения данных.

Ключевой актуальностью выпускной квалификационной работы является необходимость разработки актуального решения для хранения большого количества данных. После создания такого решения оно будет внедрено в работу сотрудников фармацевтической компании «Озон». В дальнейшем это значительно упростит их работу по подбору оборудования.

Целью работы является создание алгоритма для подбора оборудования систем хранения данных для фармацевтической компании «Озон».

Бакалаврская работа может быть разделена на следующие логически взаимосвязанные части: классификация существующих систем хранения данных и их программная часть; разбор инфраструктуры доступа, аварийного восстановления и видов RAID; анализ видов систем хранения данных; разбор аппаратных комплектующих и назначений систем хранения данных.

В результате исследования мы представляем готовый алгоритм по подбору комплектующих для создания необходимой системы хранения данных для фармацевтической компании «Озон». Фармацевтическая компания «Озон» и ООО «Управляющая компания» являются частью холдинга. Разработка велась в ИТ-подразделении ООО «Управляющая компания».

Подводя итоги, мы бы хотели подчеркнуть, что данная работа актуальна не только в решении проблемы сложности эффективного хранения большого массива данных и доступа к нему, но и в вопросе упрощения процесса отбора компонентов при обновлении существующих систем хранения данных в фармацевтической компании «Озон».

ABSTRACT

The title of the bachelor's thesis is "Algorithm for selecting equipment for data storage systems."

The bachelor's thesis consists of a scheme, three main ones, detection, tables, a list of references, including foreign sources, and a prototype algorithm for selecting equipment for data storage systems.

The key relevance of the bachelor's thesis is the need to develop an up-to-date solution for storing large amounts of data. After creating such a solution, it will be implemented in the work of the employees of the pharmaceutical company Ozon. In the future, this will greatly simplify their work on the selection of equipment.

The aim of the work is to create an algorithm for selecting equipment for data storage systems for the pharmaceutical company "Ozon".

The bachelor's thesis can be divided into the following logically interconnected parts: classification of existing data storage systems and their software part; analysis of access infrastructure, disaster recovery and types of RAID; analysis of types of data storage systems; analysis of hardware components and purposes of data storage systems.

As a result of the study, we present a ready-made algorithm for selecting components to create the necessary data storage system for the Ozone pharmaceutical company. Pharmaceutical company "Ozon" and LLC "Management Company" are part of the holding. The development was carried out in the IT department of Management Company LLC.

Summing up, we would like to emphasize that this work is relevant not only in solving the problem of the complexity of efficient storage of a large amount of data and access to it, but also in the issue of simplifying the process of selecting components when updating existing data storage systems in the pharmaceutical company Ozon.

Оглавление

Введение.....	5
Глава 1 Классификация систем хранения данных и их программная часть.....	8
1.1 Инфраструктура доступа.....	9
1.1.1 Системы прямого подключения (DAS)	10
1.1.2 Устройства хранения данных, подключаемые к NAS	13
1.1.3 SAN.....	16
1.2 Аварийное восстановление (Disaster Recovery).....	18
1.3 Программный и аппаратный RAID	20
Глава 2 Анализ видов систем хранения данных	28
2.1 Блочные системы хранения	28
2.2 Файловая система хранения.....	31
2.3 Объектные системы хранения данных	35
Глава 3 Разработка алгоритма подбора оборудования для систем хранения данных	41
3.1 Спецификация задачи.....	41
3.2 Аппаратные комплектующие системы хранения данных	41
3.2.1 Устройства хранения в СХД.....	42
3.2.2 Контроллер диска и задняя панель	44
3.2.3 Контроллер системы хранения.....	45
3.2.4 Сетевые адаптеры	48
3.3 Типы назначений систем хранения данных	49
Заключение	56
Список используемой литературы	57
Приложение А Листинг программного кода.....	61
Приложение Б Основные характеристики SDS решений	67

Введение

«С каждым годом объем информации, который создается в мире, растет в десятки и сотни раз. Согласно данным компании IDC, на скорости этого роста не сказывается даже глобальный мировой кризис и геополитические ситуации. Более того, в ближайшие несколько лет эксперты прогнозируют сильное увеличение потребности в объемах таких хранилищ на 50% в год. Таким образом, через пару лет электронный баланс может составить более 85 единиц» [8] [6].

В условиях увеличения количества компаний, которые собирают и хранят огромное количество информации в цифровом виде, особенно остро встает проблема передачи, хранения, защиты и управления этими данными. Для оптимизации ресурсов хранения, а также для более эффективной работы с данными, как правило, используются специализированные решения в отношении хранения и структуризации данных. Для этих целей создаются организованные системы хранения данных (СХД). Любая из таких систем должна включать в себя специальные устройства хранения информации, подходящее программное обеспечение и систему управления для корректного доступа и изменения всего хранилища данных.

Во многих крупных компаниях обострена проблема хранения данных. Данные могут храниться на ненадежных ресурсах (облачные хранилища и одиночные съемные диски). Таким хранилищам тяжело обеспечить сохранность и безопасность данных, к ним могут получить доступ третьи лица, а поиск информации в этой базе данных займет очень много времени, не говоря уже о том, что сам поиск может быть неудобным и неэффективным. Эту проблему можно устранить, обеспечив качественную систему сохранения информации, что подойдет конкретной кампании для её нужд.

Объектом исследования выпускной квалификационной работы является системы хранения данных.

Предметом исследования является алгоритм подбора оборудования систем хранения данных.

Целью выпускной квалификационной работы является создание алгоритма для подбора оборудования систем хранения данных для фармацевтической компании «Озон».

Актуальность решаемой в выпускной квалификационной работы задачи обусловлена необходимостью разработки актуального и нового решения хранения большого количества данных. После создания такого решения оно будет внедрено в работу сотрудников фармацевтической компании, что впоследствии значительно упростит их работу по подбору оборудования.

Для достижения данной цели необходимо:

- провести анализ вариантов создания локальных систем хранения данных;
- изучить виды носителей и составные части систем хранения данных;
- описать варианты решения поставленной задачи на основе полученных и изученных данных;
- разработать алгоритм для подбора оборудования систем хранения данных для фармацевтической компании «Озон» на основе полученных в исследовании выводов.

Практическая значимость выпускной квалификационной работы является создание алгоритма для подбора оборудование систем хранения данных.

Данная работа состоит из введения, трёх глав, заключения и списка используемой литературы.

В первой главе описываются технические характеристики различных систем хранения данных (СДХ), их виды, варианты использования, различия и преимущества.

Вторая глава посвящена способу хранения информации в системах хранения данных.

В третьей главе описан процесс создания алгоритма подбора системы хранения данных, его кастомная сборка и сравнение этой сборки с вариантами, представленными на рынке.

В заключении описываются результаты исследования и итог выпускной квалификационной работы.

Бакалаврская работа состоит из 60 страниц текста и 21 источника.

Глава 1 Классификация систем хранения данных и их программная часть

«Система Хранения Данных (СХД) – это комплексное программно-аппаратное решение по организации надежного хранения информационных ресурсов и предоставления гарантированного доступа к ним» [14].

«Для любой организации, даже если это несколько десятков компьютеров и несколько территориально разнесенных офисов, характерны типичные проблемы, связанные с растущими объемами информации» [8].

- «Децентрализация информации – если раньше все данные могли храниться буквально на одном жестком диске, то теперь любая функциональная система нуждается в отдельном хранилище – к примеру, серверах электронной почты, СУБД, домене и пр. Ситуация усложняется в случае наличия территориально-распределённых офисов (филиалов)» [8].

- «Лавинообразный рост информации – зачастую количество жестких дисков, которые можно установить в конкретный сервер, не может покрыть необходимую системе емкость. И, как следствие, – невозможность полноценно защитить хранимые данные, недостаточная скорость обработки информации, сложность резервного копирования (архивирования)» [8].

- «Сложно или невозможно предусмотреть требуемый объем дискового пространства при развертывании компьютерной системы, вследствие чего возникают проблемы расширения дисковых емкостей и неэффективной утилизации ресурсов» [8].

- «Низкая степень конфиденциальности распределенных данных – невозможно проконтролировать и ограничить доступ в соответствии с политикой безопасности предприятия. Это касается как доступа к данным по существующим для этого каналам (локальная сеть), так и физического доступа к носителям» [8].

- «Сложность управления распределенными потоками информации – любые действия, направленные на изменение данных в каждом филиале,

содержащем часть распределенных данных, создает определенные проблемы, начиная от сложности синхронизации различных баз данных, версий файлов разработчиков и заканчивая ненужным дублированием информации» [8].

- «Низкий экономический эффект внедрения «классических» решений – по мере роста информационной сети, больших объемов данных и все более распределенной структуры предприятия финансовые вложения оказываются не столь эффективны и зачастую не способствуют решению возникающих проблем» [8].

- «Высокие затраты используемых ресурсов для поддержания работоспособности всей информационной системы предприятия – начиная с необходимости содержать большой штат квалифицированного персонала и заканчивая многочисленными дорогостоящими аппаратными решениями относительно проблемы объемов и скоростей доступа к информации вкупе с надежностью хранения и защитой от сбоев» [8].

Исходя из этих проблем, рано или поздно возникающих у любой динамично развивающейся компании, можно сформулировать требования к современным системам хранения данных.

1.1 Инфраструктура доступа

Развитие сетевых технологий привело к появлению двух сетевых решений для СХД – сетей хранения Storage Area Network (SAN) для обмена данными на уровне блоков, поддерживаемых клиентскими файловыми системами, и серверов для хранения данных на файловом уровне Network Attached Storage (NAS). Чтобы отличать традиционные СХД от сетевых был предложен еще один термин – Direct Attached Storage (DAS).

Появлявшиеся на рынке последовательно DAS, SAN и NAS отражают эволюционирующие цепочки связей между приложениями, использующими данные, и байтами на носителе, содержащим эти данные. Когда-то сами программы-приложения читали и писали блоки, затем появились драйверы

как часть операционной системы. В современных DAS, SAN и NAS цепочка состоит из трех звеньев: первое звено – создание RAID-массивов, второе – обработка метаданных, позволяющих интерпретировать двоичные данные в виде файлов и записей, и третье – сервисы по предоставлению данных приложению. Они различаются по тому, где и как реализованы эти звенья. В случае с DAS СХД является «голой», она только лишь предоставляет возможность хранения и доступа к данным, а все остальное делается на стороне сервера, начиная с интерфейсов и драйвера. С появлением SAN обеспечение RAID переносится на сторону СХД, все остальное остается так же, как в случае с DAS. А NAS отличается тем, что в СХД переносятся к тому же и метаданные для обеспечения файлового доступа, здесь клиенту остается только лишь поддерживать сервисы данных.

1.1.1 Системы прямого подключения (DAS)

DAS — это архитектура, в которой хранилище подключается напрямую к хостам. Внутренний диск хоста и напрямую подключенный внешний массив хранения являются некоторыми примерами DAS. Хотя внедрение сетевых технологий хранения данных становится все более популярным, DAS по-прежнему подходит для локализованного доступа к данным в небольших средах, таких как персональные компьютеры и рабочие группы. DAS классифицируется как внутренний или внешний в зависимости от расположения устройства хранения относительно хоста. Во внутренних архитектурах DAS устройство хранения внутренне подключено к хосту через последовательную или параллельную шину. На физических шинах ограничения по расстоянию, они поддерживаются только на более коротком расстоянии для высокоскоростных подключений. Помимо этого некоторые внутренние шины поддерживают лишь небольшое количество устройств, но занимают большое пространство внутри хоста и мешают обслуживанию других компонентом.

На внешней архитектуре DAS доступ к информации осуществляется на уровне блоков, а сервер подключается напрямую к внешнему устройству

хранения. SCSI или FC используют для связи между хостом и устройством хранения. Благодаря этому внешнее хранилище может преодолеть ограничение по расстоянию и количеству устройств, а также обеспечить централизованного управления устройствами хранения.

При этом DAS требует гораздо меньших инвестиций, чем сетевые архитектуры хранения данных. Простая и удобная конфигурация DAS позволяет ее быстро развернуть и использовать в работе. Управляемость осуществляется с использованием инструмента для управления хранилищем на основе хоста, например, ОС хоста, что упрощает задачи управления хранилищем в небольших средах.

Как следствие, DAS имеет простую архитектуру и для ее настройки необходимо меньше функций управления, аппаратных элементов. Несмотря на это, DAS не может быть масштабируемым. У массива хранения есть ограниченное количество портов, что ограничивает число хостов для непосредственного подключения к хранилищам.

DAS не может обеспечить оптимальное использование ресурсов из-за ограниченного количества возможностей совместного использования интерфейсных портов. В средах DAS неиспользуемые ресурсы не могут быть легко перераспределены, что приводит к островкам чрезмерно используемых и недоиспользуемых пулов хранения.

Преимущества и недостатки DAS перечислены в таблице 1.

Таблица 1- Преимущества и недостатки сети прямого подключения DAS

	Преимущества	Недостатки
	Низкая стоимость	Слабая управляемость
	Простота развертывания и администрирования	Низкая степень консолидации ресурсов – вся ёмкость доступна одному или двум серверам, что снижает гибкость распределения данных между серверами. В результате необходимо закупать либо больше внутренних жестких дисков, либо ставить дополнительные дисковые полки для других серверных систем
	Системы хорошо подходят для хранения потоковых мультимедиа данных из-за высокой скорости обмена с дисками, а также возможности построения емких систем	Не оптимальная утилизация ресурсов, поскольку каждая DAS система требует подключения выделенного сервера и позволяет подключить максимум 2 сервера к дисковой полке в определенной конфигурации.
DAS	Высокая скорость обмена данными между системой хранения и сервером	Низкая надежность. При выходе из строя сервера, к которому подключено данное хранилище, данные перестают быть доступными
		Ограничены небольшим расстоянием подключения сервера к системам хранения данных
		Решения DAS, также, не позволяют нескольким серверам совместно использовать файлы данных
		Защита информации ограничена возможностями ОС: это разграничение прав доступа к объектам и специализированное ПО (антивирусные системы, криптографические инструменты)

Один из главных недостатков технологии DAS в сегодняшней ИТ-инфраструктуре — доступность дисковой системы, ассоциированной с

данным сервером, только через это устройство (DAS), что приводит к созданию отдельных «островов» данных. Кроме того, файлы не могут совместно использоваться несколькими серверами, а для добавления дисковых накопителей к сети серверы должны быть переведены в автономный режим. Чтобы разрешить эти проблемы и обеспечить избыточность и высокую готовность данных, компаниям приходилось копировать данные на несколько серверов и использовать различные RAID-конфигурации.

Системы DAS следует использовать при необходимости увеличения дискового пространства одного сервера и вынесения его за корпус. Также DAS-системы можно рекомендовать к применению для рабочих станций, обрабатывающих большие объемы информации.

На смену DAS-системам постепенно приходят либо универсальные решения с возможностью плавной миграции с NAS-системам, либо системы, предусматривающие возможность их использования как в качестве DAS-, так и NAS- и даже SAN-систем.

1.1.2 Устройства хранения данных, подключаемые к NAS

NAS – это специализированное высокопроизводительное устройство, которое предназначено для хранения и обмена файлами. На сегодняшний день, благодаря NAS, клиенты могут обмениваться файлами по IP-сети. NAS обеспечивает удобство в консолидации серверов, устраняя необходимость в нескольких файловых серверов. Он объединяет хранилище, используемое клиентами в единую систему хранения. Это упрощает управление хранилищем. Его также можно использовать для управления складом и хранением.

На данный момент NAS использует сетевые протокол и протоколы обмена данными для обеспечения доступа к данным файла. TCP/IP используется для передачи данных, а также общую файловую систему Интенсификации (CIFS) и сетевую файловую систему (NFS). Эти протоколы включают в себя общие файловые системы CIFS и NFS для сетевой файловой службы. NAS позволяет пользователям UNIX и Microsoft Windows

безошибочно обмениваться одними и теми же данными. NAS предоставляет пользователям UNIX и Microsoft Windows возможность обмена одними и теми же данными.

Устройство NAS использует свою собственную операционную систему и интегрированные аппаратно-программный и программный компоненты для удовлетворения конкретных требований файловых служб. Информационные службы имеют собственные функции, которые обеспечивают удовлетворительные требования к файлам. У него есть операционная система, которая оптимизирована для файлового ввод-выпуска и поэтому выполняет файловый ввод-вывод лучше, чем сервер общего назначения.

В результате устройство NAS может обслуживать больше клиентов, чем серверы общего назначения, и обеспечивает преимущества консолидации серверов. Сетевая среда обмена файлами состоит из нескольких файловых серверов или устройств NAS. Может потребоваться переместить файлы с одного устройства на другое по таким причинам, как стоимость или производительность. Виртуализация на уровне файлов, реализованная в среде обмена файлами, обеспечивает простое и бесперебойное решение для обеспечения мобильности файлов. Что позволяет перемещать файлы между NAS устройствами, даже если к файлам исполняется доступ. В главе описываются компоненты NAS, различные типы реализации NAS и протоколы обмена файлами, используемые в реализациях NAS. В главе также объясняются факторы, влияющие на производительность NAS, и виртуализация на уровне файлов.

NAS предлагает следующие преимущества:

- Доступ к информации: гарантирует действенный обмен файлами и поддерживает конфигурации «все к одному» и «один ко всем». Взаиморасположение «все к одному» позволяет обслуживать множество клиентов устройству NAS одновременно. Взаиморасположение «один ко всем» позволяет подключаться ко многим устройствам NAS, одному клиенту одновременно.

- Повышенная эффективность: NAS обеспечивает более высокую производительность по сравнению с файловым сервером общего назначения, поскольку NAS использует операционную систему, специализированную для обслуживания файлов.

- Улучшенная гибкость: совместимость с клиентами на платформах UNIX и Windows, использующих стандартные отраслевые протоколы. NAS является гибким и может обслуживать запросы от разных типов клиентов из одного и того же источника.

- Централизованное хранилище: централизует хранилище данных, чтобы свести к минимуму дублирование данных на клиентских рабочих станциях и обеспечить лучшую защиту данных.

- Упрощенное управление: централизованная консоль, позволяющая эффективно управлять файловыми системами.

- Масштабируемость: хорошо масштабируется с различными профилями использования и типами бизнес-приложений благодаря высокой производительности и низкой задержке.

- Высокая доступность: предлагает эффективные варианты репликации и восстановления, позволяя высокой доступностью данных. NAS использует резервные компоненты, которые обеспечивают максимальные возможности подключения. Устройство NAS поддерживает технологию кластеризации для аварийного переключения.

- Защищенность: обеспечивает надежность, пользовательскую идентификацию и блокировку файлов с помощью обычных методик безопасности.

- Невысокая стоимость: NAS использует компоненты Ethernet, только те что дешевые и доступные.

- Несложность развертывания: простейшая установка на клиенте, поскольку клиенты имеют встроенное программное обеспечение для подключения к NAS.

Также можно выделить ряд недостатков.

- Доступ к информации через протоколы “сетевых файловых систем” зачастую медленнее, чем как к локальному диску.

- Большинство недорогих NAS-серверов не позволяют обеспечить скоростной и гибкий метод доступа к данным на уровне блоков, присущих SAN системам, а не на уровне файлов.

Главный недостаток NAS заключается в повышенной нагрузке на сеть. Для соединения с сетью обычно используется несколько гигабитных Ethernet-контроллеров, работающих в дуплексном режиме, и хотя таким образом достигается должная скорость обмена данными по сетевой среде, нагрузка на сеть возрастает. Поэтому предварительно необходимо рассчитать расположение и количество NAS-устройств.

NAS распространены намного меньше, чем устройства SAN, но доля таких систем постоянно растёт за счёт вытеснения DAS.

«Network Attached Storage (NAS) – сетевая система хранения данных, сетевое хранилище. По сути это компьютер с некоторым дисковым массивом, подключенный к сети (обычно локальной) и поддерживающий работу по принятым в ней протоколам. Часто диски NAS объединены в RAID-массив. Несколько таких компьютеров могут быть объединены в одну систему» [2].

Подключенный накопитель к сетевым настройкам, обеспечивающий файловый доступ к информации для сред LAN/WAN. Главные преимущества – высокая скорость развертывания, отлично организованные операции взаимодействия с контейнерами и возможность использования в узконаправленных секторах.

1.1.3 SAN

SAN — это высокоскоростная выделенная сеть серверов и общих устройств хранения. SAN обеспечивает консолидацию хранилищ и облегчает централизованное управление данными. Он эффективно удовлетворяет потребности в хранении с большей экономией за счет масштаба, а также обеспечивает эффективное обслуживание и защиту данных. Виртуализированная SAN и виртуализация блочных хранилищ обеспечивают

улучшенное использование и совместную работу распределенных ресурсов хранения. Внедрение виртуализации в SAN обеспечивает повышенную производительность, использование ресурсов и управляемость. Распространенными развертываниями SAN являются SAN Fibre Channel (FC) и IP SAN. Fibre Channel SAN использует протокол Fibre Channel для передачи данных, команд и информации о состоянии между серверами (или хостами) и устройствами хранения. IP SAN использует для связи протоколы на основе IP. В этой главе подробно рассказывается о технологии FC, на которой развернута сеть FC SAN. Он также охватывает компоненты FC SAN, топологии и виртуализацию блочных хранилищ.

Достоинства и недостатки SAN указаны в таблице 2.

SAN применяется в типах систем хранения данных Software-Defened storage (SDS), список готовых решений и их характеристики указаны в таблице Б1 приложения Б.

Таблица 2- Преимущества и недостатки SAN

	Преимущества	Недостатки
SAN	Высокая надёжность доступа к данным, находящимся на внешних системах хранения. Независимость топологии SAN от используемых СХД и серверов	Все недостатки такого рода решений, объединены только высокой стоимостью. На стоимость очень сильно влияет дефицит и высокая стоимость скоростных каналов связи.
	Централизованное хранение данных (надёжность, безопасность).	
	Удобное централизованное управление коммутацией и данными	
	Перенос интенсивного трафика ввода-вывода в отдельную сеть – разгрузка LAN	
	Высокое быстродействие и низкая латентность	
	Масштабируемость и гибкость логической структуры SAN	
	Географические размеры SAN, в отличие от классических DAS, практически не ограничены	
	Возможность оперативно распределять ресурсы между серверами	

Продолжение таблицы 2

Возможность строить отказоустойчивые кластерные решения без дополнительных затрат на базе имеющейся SAN	
Простая схема резервного копирования – все данные находятся в одном месте	
Наличие дополнительных возможностей и сервисов (снапшоты, удаленная репликация).	
Высокая степень безопасности SAN	

Ещё одним преимуществом является возможность прямо из сети хранения данных выгружать сервера. При таком способе можно очень быстро заменить упавший сервер на сервер-замену, настроив ... так, что он будет загружаться с SAN упавшего сервера. На эту операцию уходит примерно тридцать минут. Идея новая, но уже используется в новейших дата-центрах.

1.2 Аварийное восстановление (Disaster Recovery)

Аварийное восстановление – это метод восстановления работоспособности ИТ-инфраструктуры и доступа к ней после стихийного бедствия или сбоя, возникшего по вине человека, например, после кибератаки или отказа оборудования. Частью плана аварийного восстановления могут быть самые разные методы. Аварийное восстановление — это один из аспектов непрерывности бизнеса [9] [12].

Это не резервное копирование, а технология для сокращения простоя ИТ-систем.

Есть два основных понятия в технологии DR.

RPO (Recovery point objective, «целевая точка восстановления») — максимальный период, за который могут быть потеряны данные. Время восстановления файлов из резервного хранилища не должно превышать показателя RPO. Если, например, RPO равен 90 минутам, будут потеряны данные, наработанные не более, чем за последние полтора часа. Соответственно, снапшот должен проводиться не реже, чем раз в 90 минут. Резервироваться могут не только файлы на дисках, но и настройки

приложений, серверных ОС, а также состояние процессов в оперативной памяти.

От выбора минимальной частоты резервного копирования зависит выбор технологий аварийного восстановления: физическая или облачная платформа, тип СХД, алгоритм репликации.

RPO устанавливается с учетом того, какой объем данных может потерять пользователь (клиент) в случае инцидента. Определиться с этим помогает расчет точки безубыточности, когда стоимость потери данных равна стоимости обеспечения их доступности. Затраты на восстановление в идеале не должны быть избыточными или недостаточными.

RTO (Recovery time objective, «целевое время восстановления») — период, в течение которого система может оставаться недоступной в случае аварии или сбоя. Другими словами, — минимальное время, за которое система должна восстановиться в рабочее состояние из любого уровня инцидента. Под «системой» понимается приложение, компьютер, сервер, сеть и т. д. Ключевой критерий для расчета RTO — стоимость простоя системы.

Пример Disaster Recovery в действии показан на рисунке 1.

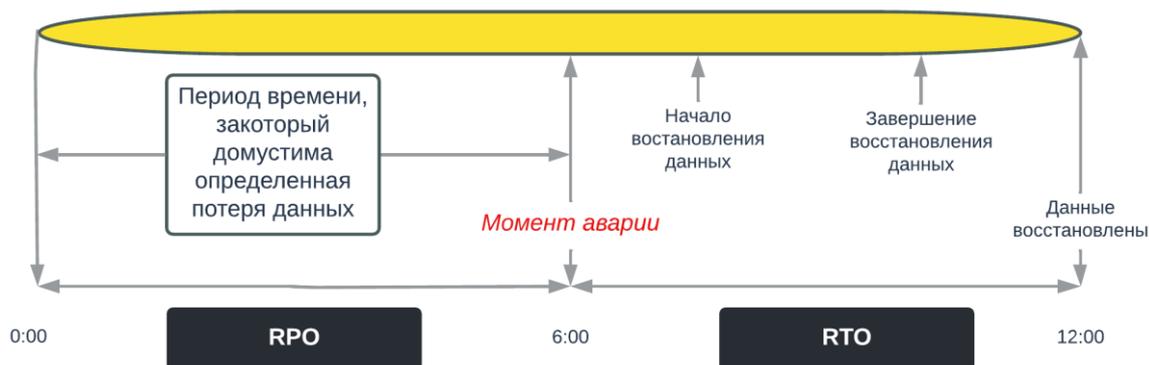


Рисунок 1- Демонстрация работы DR в действии

RTO может составлять от нескольких секунд — например, при репликации в облаке, до нескольких дней, когда бэкап пишется на физические

носители — например, на магнитную ленту. Выбор остается за заказчиком: если это клиентский портал банка, время простоя не должно превышать нескольких минут, если персональный сайт — он может «полежать» и сутки.

1.3 Программный и аппаратный RAID

Предсказать выход диска из строя со 100%-ой точностью невозможно, поэтому следует предусмотреть вариант, при котором это не станет проблемой или причиной остановки сервисов. Использование RAID-массивов решает эту задачу. Рассмотрим два основных подхода, применяющихся для этой задачи:

Программный RAID - использует программное обеспечение на основе хоста для обеспечения функций RAID. Он реализован на уровне операционной системы и не использует выделенный аппаратный контроллер для управления массивом RAID. Программные реализации RAID предлагают преимущества в цене и простоте по сравнению с аппаратными RAID. Однако они имеют следующие ограничения:

- Производительность: программный RAID влияет на общую производительность системы. Это связано с дополнительными циклами ЦП, необходимыми для выполнения вычислений RAID.
- Поддерживаемые функции: программный RAID поддерживает не все уровни RAID.
- Совместимость с операционной системой: обновления программного RAID или операционной системы должны быть проверены на совместимость. Это приводит к нестабильности в среде обработки данных.

Аппаратный RAID - В аппаратных реализациях RAID специализированный аппаратный контроллер реализуется либо на хосте, либо на массиве.

«Карта контроллера RAID — это аппаратная реализация RAID на базе хоста, в которой на хосте установлен специализированный RAID-контроллер,

к которому подключены дисковые накопители. Производители также интегрируют RAID-контроллеры в материнские платы.» [21]

Контроллер RAID на основе хоста не является эффективным решением в среде центра обработки данных с большим количеством хостов. Внешний RAID-контроллер представляет собой аппаратный RAID-массив. Он действует как посредник между сервером и дисковыми массивами. Он представляет серверу тома хранилища, а сервер управляет этими томами как физическими дисками. Основные функции RAID-контроллеров следующие:

- Управление и контроль дисковых агрегатов
- Трансляция запросов ввода-вывода между логическими дисками и физическими дисками
- Регенерация данных в случае сбоя диска

Производители СХД с программным RAID используют всю мощь нового поколения аппаратных комплектующих и на один-два года опережают производителей аппаратных RAID-массивов по срокам выпуска новых моделей. В то время как производителям аппаратного RAID необходимо модернизировать производственный процесс, для СХД с программным RAID достаточно протестировать новую корзину или процессор, - и новая модель готова к поставке.

Различают следующие уровни RAID-массивов:

- RAID 0, чередующийся набор без отказоустойчивости;
- RAID 1, отзеркалирование дисков;
- RAID 3, чередующийся набор с параллельным доступом и выделенным диском четности;
- RAID 5, чередующийся набор с независимым доступом к диску и распределенной четностью;
- RAID 6, чередующийся набор с независимым доступом к диску и двойной распределенной четностью.

Рассмотрим подробнее каждый уровень.

В конфигурации RAID 0 используются методы чередования данных, при которых данные распределяются по всем дискам в наборе RAID. Поэтому он использует полную емкость хранилища RAID-набора. Для чтения данных все полосы снова соединяются контроллером.

На рисунке 2 показан RAID 0 в массиве, в котором данные чередуются на двух дисках. Когда количество дисков в наборе RAID увеличивается, производительность повышается, поскольку одновременно можно считывать или записывать больше данных. RAID 0 — хороший вариант для приложений, которым требуется высокая пропускная способность ввода-вывода. Однако если этим приложениям требуется высокая доступность во время сбоев дисков, RAID 0 не обеспечивает защиту и доступность данных.

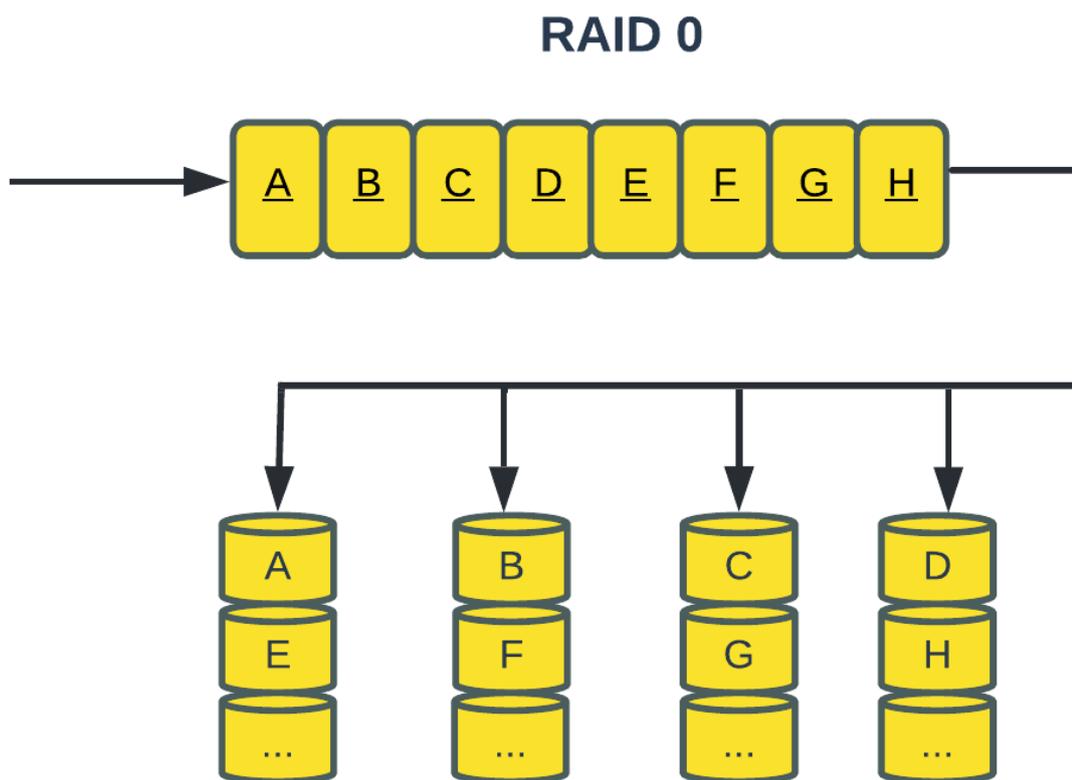


Рисунок 2- Принцип выполнения RAID 0 (разделение)

RAID 1 основан на технологии зеркалирования. В этой конфигурации RAID данные зеркалируются для обеспечения отказоустойчивости (см.

рисунок 3). Набор RAID 1 записывает одну запись на два диска параллельно. Зеркалирования прозрачно для сервера. В случае сбоя диска влияние на восстановление данных в RAID 1 является наименьшим среди всех реализаций RAID. Это связано с тем, что контроллер RAID использует зеркальный диск для восстановления данных. RAID 1 подходит для приложений, которым требуется высокая доступность, а стоимость не является ограничением [20].

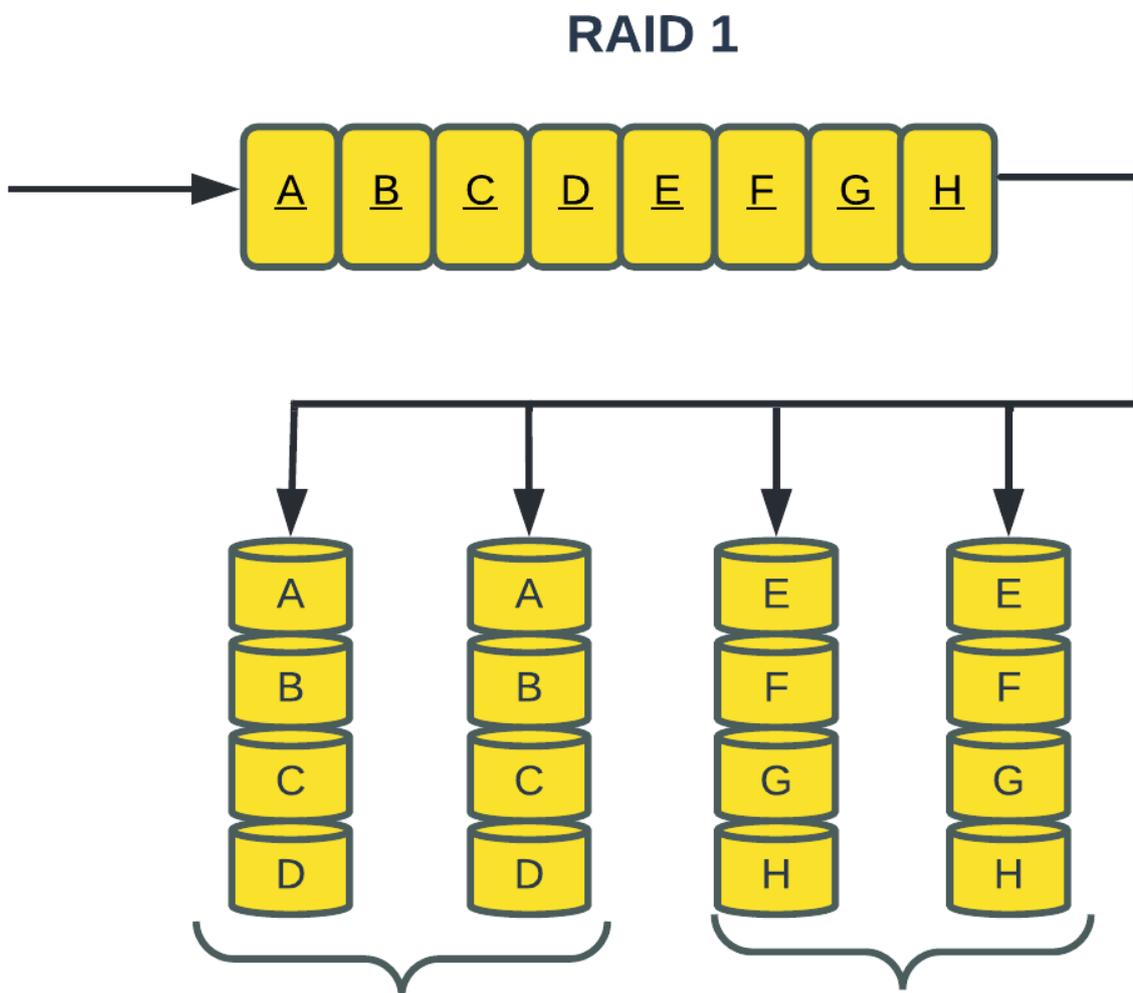


Рисунок 3- Выполнение RAID 1 (дублирование)

RAID 3 чередует данные для повышения производительности и использует контроль четности для обеспечения отказоустойчивости. Информация о четности хранится на выделенном диске, чтобы данные можно было восстановить в случае сбоя диска в массиве RAID. Например, в наборе

из пяти дисков четыре используются для данных и один для контроля четности. Общее требуемое дисковое пространство увеличивается в 1,25 раза от исходного значения. Диски работают параллельно, поэтому RAID 3 считывает и записывает полные полосы данных. Отсутствуют частичные записи, которые обновляют лишь часть данных в полосе. Рисунок 4 иллюстрирует реализацию RAID 3.

RAID 3 обеспечивает хорошую производительность для приложений, требующих последовательного доступа к большим объемам данных, таких как резервное копирование данных или потоковое видео.

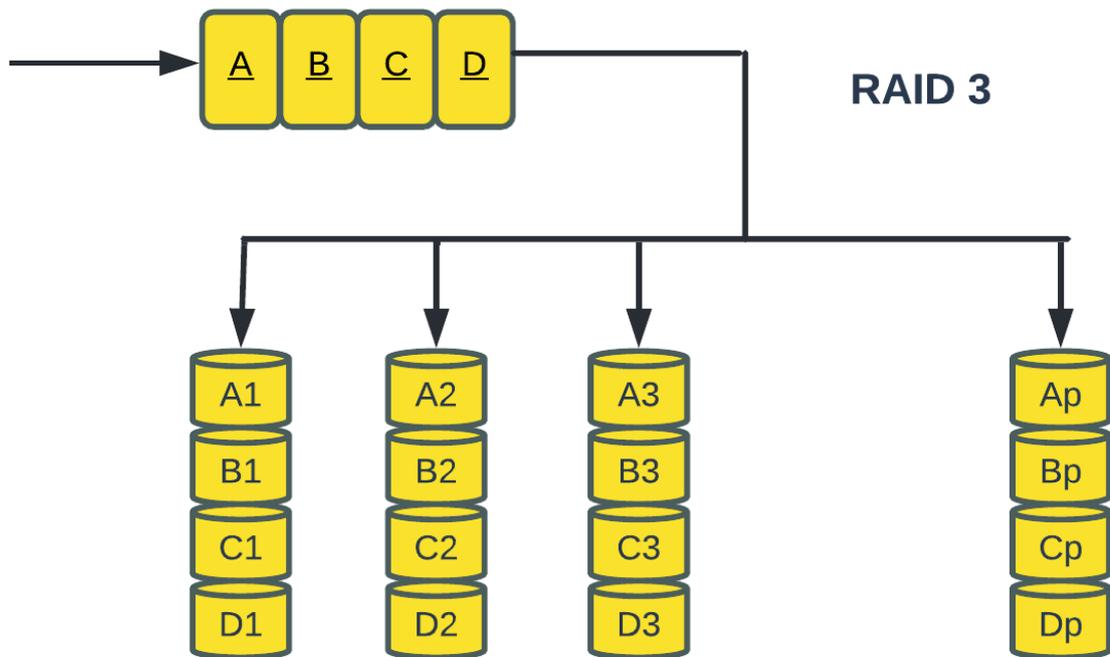


Рисунок 4 - Выполнение RAID 3

RAID 5 — это универсальная реализация RAID. Он похож на RAID 4, поскольку использует чередование. Диски (планки) также доступны независимо. Разница между RAID 4 и RAID 5 заключается в местоположении четности. В RAID 4 контроль четности записывается на выделенный диск, что создает узкое место при записи для диска контроля четности. В RAID 5

контроль четности распределяется по всем дискам, чтобы преодолеть узкое место записи, связанное с выделенным диском контроля четности. Рисунок 5 иллюстрирует реализацию RAID 5.

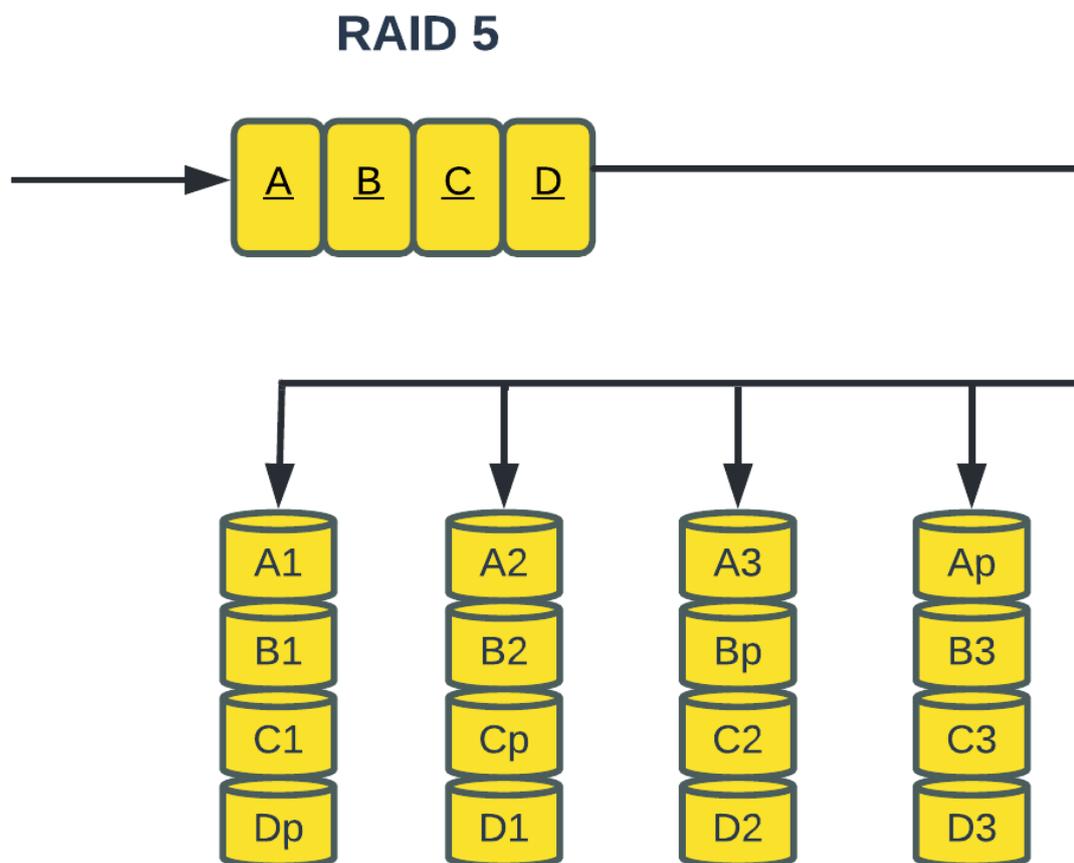


Рисунок 5- Выполнение RAID 5

RAID6 аналогичен RAID 5, за исключением того факта, что RAID6 имеет второй элемент контроля четности, который обеспечивает работоспособности в случае отказа двух дисков одновременно в наборе RAID (см. рисунок 6). Для реализации RAID6 нужно минимум 4 диска. RAID6 распределяет четность по всем дискам поочередно. Штраф за запись в RAID6 больше, чем в RAID 5. Поэтому запись в RAID 5 выполняется быстрее, чем в RAID6. Операция восстановления в RAID6 может занять больше времени, чем в RAID 5, из-за наличия двух наборов четности.

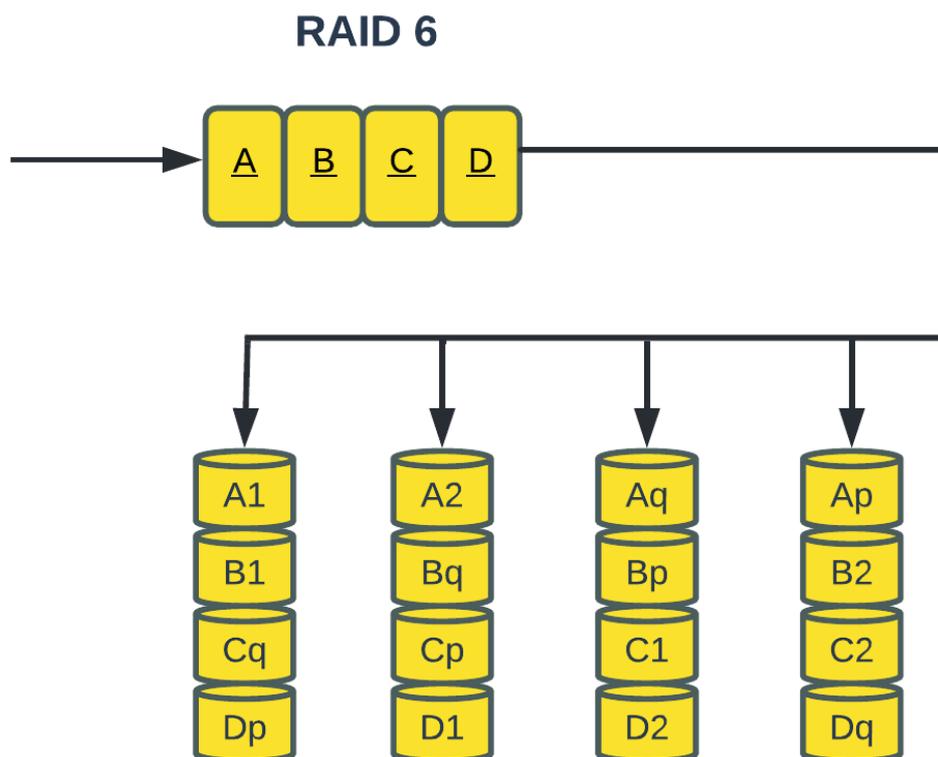


Рисунок 6- Выполнение RAID 6

Вышеприведенные схемы хорошо иллюстрируют принцип действия RAID. Основные уровни RAID и их характеристики указаны в таблице 3.

Таблица 3- Основные характеристики видов RAID

Уровень RAID	RAID0	RAID1	RAID5	RAID6	RAID10
Устойчивость к ошибкам	Низкая	Высокая	Средняя	Наивысшая	Высокая
Избыточность	Никакая	Дублирование	Паритет	Паритет	Дублирование
Доступная емкость	100%	50%	$(N-1)/N$	$(N-2)/N$	50%
Производительность	Наивысшая	Низкая	Средняя	Средняя	Высокая

«При определении кол-ва дисков, необходимых для приложения, нужно учитывать влияние RAID на IOPS, генерируемого приложением. Нагрузка на диск должна рассчитываться с учетом типа конфигурации RAID и соотношения операций чтения и записи с хоста.» [21]

Следующий пример иллюстрирует метод вычисления нагрузки на диск в различных типах RAID.

Рассмотрим случай, например, приложение которое генерирует 5200 операций ввода-вывода в секунду, причем 60 процентов из них приходится на чтение. Нагрузка на диск в RAID 5 рассчитывается следующим образом:

Нагрузка на диск RAID 5 (чтение + запись) = $0,6 * 5200 + 4 * (0,4 * 5200)$
[поскольку штраф за запись для RAID 5 равен 4].

$$= 3120 + 4 * 2080$$

$$= 3120 + 8320$$

$$= 11\,440 \text{ операций ввода-вывода в секунду}$$

Нагрузка на диск в RAID 1 рассчитывается следующим образом:

Нагрузка на диск RAID 1 = $0,6 * 5200 + 2 * (0,4 * 5200)$ [поскольку каждая запись проявляется как две операции записи на диски]

$$= 3120 + 2 * 2080$$

$$= 3120 + 4160$$

$$= 7280 \text{ операций ввода-вывода в секунду}$$

«Расчетная нагрузка на диск определяет количество дисков, необходимых для приложения. Если в этом примере необходимо использовать дисковый накопитель с максимальным значением 180 IOPS, количество дисков, необходимое для соответствия рабочей нагрузке конфигурации RAID.» [21]

Получим следующие:

$$\text{RAID5: } 11440/180 = 64 \text{ диска}$$

$$\text{RAID1: } 7280/180 = 42 \text{ диска (округляется всегда в большую).}$$

Глава 2 Анализ видов систем хранения данных

2.1 Блочные системы хранения

Данные, хранящиеся в блочных системах, делятся на блоки одинакового размера. Блок данных - это не файл, а недостающий объект, и лишь некоторые данные в фрагменте фиксированного размера (английский - chunk). Файл данных может быть помещен в ограниченное количество блоков, а если последний из этих блоков не будет заполнен, он все равно будет иметь тот же фиксированный размер, что и заполненные блоки. Доступ к блокам сервер получает через СХД SAN (это сеть, которая подключена к хранилищу) [8].

Блок-хранилище может использоваться любой операционной системой в качестве дискового блока этой системы. Каждому серверу может быть предоставлен практически неограниченный размер диска. Пользователь видит этот диск в интерфейсе операционной системы сервера. Серверная операционная система может подключаться к блокам данных через высокоскоростные интерфейсы FC (Fiber Channel) или iSCSI. Поэтому главным преимуществом блочных систем является высокая производительность и скорость [5].

Операционная система сервера может быть, как физической (хост), так и виртуальной (виртуальной машиной). Однако следует отметить, что физическим серверам необходимы дополнительные контроллеры для доступа к блокам ресурсов.

Величина агрегатов зависит от производителя оборудования. Однако архитектор, создающий дизайн корпоративной ИТ-системы, может определить размер блока данных в зависимости от требований к конфигурации и сценария реализации. Размер блока может зависеть от типа физических дисков в системе хранения (жестких дисков или твердотельных накопителей) и типа данных клиента. Во многих случаях правильный выбор

размера диска поможет оптимизировать работу корпоративной ИТ-системы в целом.

Программное обеспечение сервера назначает любому блоку информации несложный определитель местоположения (идентификатор местоположения), который можно быстро найти в SAN, поэтому хранение блоков имеет высокую производительность.

Например, в системах хранения объектов используются метаданные (тип файла, имя файла, дата и т. д.). Обратим внимание, что он используется как такой идентификатор. Хотя системы хранения объектов получают достаточно преимуществ для других ситуаций, очевидно, что при таком механизме увеличивается время нахождения объекта в системе.

Обычно при разработке ИТ-системы IOPS выбирает пары критических параметров, таких как длительность введение и выведения информации (операции ввода-вывода в секунду) и задержка доступа к информации благодаря SAN (время задержки). Как вариант, это могут быть серверы баз информации или критически важные для бизнеса приложения, требующие высокой производительности и высокой надежности.

Блок-системы используются в базах данных, таких как Oracle или SAP Hana. Виртуальные машины VMware по умолчанию также используют блочное хранилище, так как можно быстро создать многофункциональную виртуальную инфраструктуру. Блочное хранилище также может использоваться практически во всех архитектурах центров обработки данных, будь то один сервер или высоконадежные кластеризованные серверные решения [8].

В настоящее время блочные системы хранения данных являются абсолютной долей в хранении корпоративных данных как в малом бизнесе, так и в ИТ-системах крупных компаний.

Наиболее распространенные применения хранения блоков:

- Бизнес-приложения и корпоративные системы CRM/ ERP, такие как СУБД (сервер баз данных SQL), 1C, SAP, Navision и другие.

- Компании интернет-торговли, различные государственные учреждения и так далее. Хранение данных.
- Различные системы виртуализации: рабочие места, сетевые функции, контейнерные технологии и т. д.
- Хранение "необработанных" данных из систем Интернета вещей (IoT).

Рассмотрим преимущества и недостатки хранения блоков.

В настоящее время в центрах обработки данных хранение блоков очень распространено из-за следующих преимуществ:

Производительность: Производительность хранения блоков. Например, для программ, которым необходимы большие скорости ввода и вывода данных (IOPS), таких как серверы баз данных, лучше всего использовать блочное хранилище. Производительность в сочетании с низкой задержкой доступа является основным преимуществом хранения блоков.

Гибкость: По мере роста организации можно быстро передавать, расширять и добавлять единицы с блоками данных, а также быстро передавать, расширять и добавлять, не влияя на производительность. Кроме того, вы можете легко перемещать хранилище блоков с первого сервера на второй, просто организовав конечный пункт маршрута.

Простота изменения файлов: В хранилище блоков при модификации данных требуется изменять только те блоки, на которые повлияли эти изменения. В хранилищах файлов и объектов для этого придется перезаписать весь файл или объект.

Вид загрузаемости ОС: Операционная система может загружаться напрямую из хранилища блоков через SAN. Для этого необходимо, чтобы BIOS сервера (физический или виртуальный) мог работать с SAN.

Работа с разрешениями: простота контроля доступа к данным.

Есть и недостатки в хранении блоков. К ним может быть связано следующее:

Привязка к серверу: блок-хранилище всегда фиксируется на одном сервере, и в этот момент доступ со второго сервера в системе не может быть разрешен. Как мы знаем, есть разные решения с программами, которые могут устранить этот недостаток, но для этого требуется большая нагрузка (перегрузка) информационной (файловой) системы.

Ограниченность объема метаданных: Размер метаданных (показывающих, что именно в базах) намного меньше, чем при хранении файлов или объектов. Как правило это влияет на работоспособность некоторых приложений, имеющих решающее значение для использования метаданных.

Стоимость: Стоимость на блочных системах довольно высока. Хотя для клиентов сервиса облачная служба хранения блоков вполне выгодна по стоимости, для провайдеров облачных хранилищ и ИТ-систем для бизнеса это очень дорого, потому что системы САН высоки по стоимости и часто нуждаются в сотрудниках высокой квалификации для их содержания. Более того, клиентам облачных сервисов всегда необходимо оплачивать объем блочного хранилища полностью, даже если весь объем не используется.

2.2 Файловая система хранения

Система хранения файлов выглядит знакомо - она похожа на то, как мы привыкли воспринимать информацию на нашем компьютере: например, файлы в подпапках. Путь к файлу в хранилище файлов может быть довольно длинным по глубокой иерархии вложенных папок.

Доступ к данным из хранилища файлов можно получить через идентификатор файла, содержащий имя сервера в общедоступной сети NAS (Network Attached Storage), путь к каталогу (папке) и имя искомого файла (server name + directory path + filename). С помощью идентификатора файла сервер хранения находит данные на диске в Nas.

Ограничения на хранение и доступ к файлам через NAS, ограничения на масштабирование базовой файловой системы и невозможность развертывания рабочей нагрузки на нескольких файловых серверах. Другими словами, масштабирование системы обычно включает в себя создание ресурсов файлового сервера и не загрузку одного или нескольких из них.

Файлы в файловой системе хранятся в каталогах (каталогах, «папках»). В каталоге хранятся сведения о свойствах, местоположении и владельце файлов. Большая часть этой информации, особенно большая часть информации, непосредственно связанной с хранением, управляется файловой системой. Сам каталог также является служебным файлом, к которому можно получить доступ с помощью различных административных процедур.

Следующая информация может храниться в каталоге:

- Название файла;
- Вид файла;
- Где он находится (путь к файлу);
- Настоящая длина файла;
- Максимально возможная длина файла;
- дата и время крайнего доступа к файлу;
- дата и время крайнего изменения файла;
- Определитель владельца файла (User ID, Owner ID);
- Данные о безопасности файла.

Такие операции могут происходить в каталогах:

- найти файл;
- создать файл и назвать его;
- удалить файл;
- составить перечень файлов в директории;
- изменить имя файла;
- переместить в другую систему файлов.

Рассмотрим уровни индекса (директорий).

Одноуровневые индексы (директории)

В каталогах с одним уровнем файлы открыты для всех юзеров.

Однако в таких каталогах пользователи не могут иметь одинаковое имя для разных файлов.

Двухуровневые директории

Двухуровневая модель файловой системы позволяет разным пользователям индивидуально получать доступ к каталогам. Различные юзеры не смогут иметь доступ и видеть файлы других пользователей.

В таком подходе у разных юзеров могут быть документы с идентичным названием. Поиск информации в этой директории более удачен, чем одноуровневых файлов.

Древообразная модель директорий

В такой модели каталоги могут быть организованы в виде древа.

Так же существует несколько методов размещения файлов.

Непрерывное размещение

Когда вы постоянно размещаете файл (Continuous Allocation), в момент его создания выделяется ряд последовательных блоков. В таблице распределения файлов (file allocation table) необходимо указать только начальную порядковую цифру блока и размер документа. Такой метод наиболее подходит для последовательного размещения отдельных файлов в пустом хранилище.

Преимущества: это размещение сразу нескольких блоков для чтения, что позволяет осуществлять высокоскоростной ввод-вывод. Доступ к отдельным подразделениям также прост. Например, если файл начинается в блоке b и нам нужен блок i , то его местоположение в репозитории просто вычисляется как $b + i - 1$.

Недочеты: документы могут быть разделены на части, если размер и длина их превышает количество доступных последовательных блоков. В то же время необходимо выполнить дополнительные действия, чтобы найти фрагменты и создать из них целостный файл. октября. Поэтому используются

специальные алгоритмы, которые помещают файлы на склад – например, как селедку в бочке на рыбном заводе – без перерыва. Таким образом, мы уменьшаем необходимый объем хранения.

Однако выполнение таких ненужных задач с точки зрения основной работы снижает общую производительность системы и приводит к необоснованному потреблению ресурсов. Кроме того, при постоянном размещении необходимо сообщить длину файла в момент его создания.

Цепочное размещение

В цепном размещении (Linked Allocation) это также называется Non-contiguous allocation, поселения выполняются в блоках. В данном случае это не требует, чтобы блоки в файле соответствовали числам. Каждый блок может содержать указатель на последующий блок цепочки, в которой продолжается файл.

Преимущества: Для таблицы размещения файлов требуется только одна запись для каждого файла, в которой указывается номер начального блока и общая длина файла. Блоки могут быть непоследовательными. Увеличение длины файла осуществляется только путем добавления свободных блоков. При таком способе размещения фрагментация файлов не имеет большого значения.

Недостатки: последние блоки файлов, возможно, не были полностью заняты. Для направления указателя на следующий блок в каждом предыдущем блоке требуется дополнительная информация (оверхед). Если указатель потерян, файл может разорваться и стать недоступным.

Индексированные размещения

Индексированное размещение (Indexed Allocation) предназначено для решения проблем непрерывного и цепного размещения. Если это так, таблица размещения файлов содержит отдельный одноранговый указатель (каталог) для каждого файла. В каталоге последовательно перечислены номера блоков, в которые был помещен файл. Блоки в этом случае могут быть как одинакового размера, так и разного размера. При этом, устраняя

фрагментацию файла, размещение с блоками разного размера исключает пустые очереди в последнем блоке каждого файла, если файл не вписывается в определенное количество блоков. Этот метод размещения файлов обеспечивает как последовательный, так и прямой доступ к файлам и в настоящее время является самым популярным.

Таким образом, были выявлены достоинства и недостатки файловой системы хранения данных.

Главное преимущество систем хранения файлов в том, что они интуитивно понятны: интерфейс похож на файловый менеджер на любом компьютере, а принцип иерархичности (вложенности) папок с файлами также взят из повседневной жизни.

Система довольно легко масштабируется (до определенных пределов). Совместное использование пользователей в системном масштабе практически ничем не ограничивается. Преимущества также можно отнести к относительно низкой цене.

Если файлы в файловой системе не масштабируются, их нужно искать достаточно быстро. Система имен файлов позволяет разным владельцам иметь файлы с одинаковым именем - даже если содержимое файлов с разными идентификаторами владельцев одинаковое, с точки зрения системы это будут разные файлы.

Удобно группировать файлы по жанрам: например, все программы, написанные на языке Java, все игровые файлы и так далее.

К недостаткам системы хранения файлов следует отнести наличие определенных ограничений масштабирования. По мере увеличения объема системы навигация становится более сложной, а время доступа к файлам увеличивается. Значит, файлы будут открываться медленнее.

2.3 Объектные системы хранения данных

Рассмотрим традиционный подход к системам хранения данных.

Традиционный подход к системам хранения данных SAN (Storage Area Network) или NAS (Network Attached Storage) - это, например, внешняя дисковая стойка, подключенная непосредственно к RAID-контроллеру сервера, учитывая недостаточно простую систему DAS (Direct Attached Storage).

Этот метод подходит для хранения в относительно небольших количествах. По мере роста дискового пространства возникают проблемы с файловой системой. Традиционные файловые системы обычно делят каждый файл на небольшие блоки на 4 килобайта и хранят местоположение каждого блока в таблицах просмотра файловой системы. Для небольших объемов данных это хорошо, но когда вы расширите свою систему хранения до петабайт или более, таблицы будут очень большими. Это значительно замедляет поиск нужного блока и увеличивает вероятность ошибок.

Из-за этого пользователи вынуждены каким-то образом распределять свои наборы данных с логическими числами на большое количество логических узлов, чтобы поддерживать производительность на приемлемом уровне. Однако сложность и стоимость управления и обслуживания ИТ-системы значительно возрастают, в то время как проблемы с производительностью, потеря данных и сбои в работе системы влияют на бизнес-процессы.

Распределенная файловая система

Для решения этой проблемы стали использоваться горизонтально масштабируемые (масштабируемые) файловые системы, такие как HDFS (Hadoop Distributed File System). Это распределенная файловая система, основанная на Hadoop, свободно распространяемом наборе утилит для создания распределенных систем, работающих на кластерах из сотен и тысяч узлов. В данном случае решается проблема масштабирования, но поддержка таких систем тоже довольно трудоемкая. Они структурно сложны и требуют постоянного ухода. Кроме того, они часто используют механизм репликации данных, то есть используются для хранения копий одних и тех же данных в

разных частях октября. В стандартной комплектации хранятся три копии каждого файла. Излишне говорить, что это увеличивает необходимый объем диска до 200%! Хотя цены на диски всегда снижаются, объемы данных, которые необходимо хранить, растут еще быстрее. Это полностью съедает сбережения на дешевых дисках. Такой подход недопустим для большого объема информации.

Облачное хранение

Чтобы свести к минимуму эти проблемы, многие начали прибегать к облачному хранилищу. Модель оплаты по потреблению (платите, как идете) работает нормально, но все же работает для относительно небольших объемов данных и редко используется. При постоянном масштабировании объемов данных и интенсивной работе с ними такой подход становится очень дорогостоящим, если не дешевле HDFS. Дело в том, что многие поставщики облачных вычислений взимают плату не только за объем сохраненных данных, но и за полученный/записанный трафик данных, а также за количество транзакций, совершаемых в хранилище. Таким образом, в то время как большой анализ данных имеет дело с большой передачей данных, хранение в общедоступном облаке, вероятно, является самым дорогим подходом. Кроме того, если в октябре многие другие пользователи интенсивно используют ресурсы этого облака, могут возникнуть проблемы с конфиденциальностью данных и производительностью системы.

В таком случае выход может быть объектно-ориентированной системой хранения данных (object storage), использующей те же технологии, что и в общедоступном облаке (HTTP, API). Хранилища объектов можно легко масштабировать до объемов в петабайтах в одном домене без снижения производительности. Кроме того, хранилища объектов имеют функции управления данными, которых нет в традиционных системах: создание версий, настройка метаданных в октябре и встроенная аналитика.

Эти характеристики достигаются путем абстрагирования системных слоев - общего подхода, который в настоящее время используется практически

во всех ИТ- и телекоммуникационных системах, а не только в системах хранения данных. Каждый диск на нижнем уровне отформатирован простой локальной файловой системой, такой как ЕХТ4. На самом высоком уровне, абстрагированном снизу, размещены элементы управления, позволяющие интегрировать все элементы в единое целое. Различные типы файлов хранятся как «объекты», а не файлы в файловой системе. Поскольку управление блоками низкого уровня передается в локальную файловую систему, хранилище объектов знает только функции управления верхнего уровня, которые управляют нижним уровнем через стандартный интерфейс прикладного программирования (Application Programming Interface).

Принцип хранения объектов сопоставим с политикой парковки, когда вы покидаете машину (объект) для размещения на парковке и получаете карту, которую можете вернуть. На карту можно добавить «метаданные»: ваше имя, номер и марку машины. Вас не интересует, где именно припарковать машину (абстракция), и вам не нужно передвигаться по стоянке в поисках свободного места.

Такой подход позволяет хранить таблицы представлений файловой системы каждого узла внизу в управляемом размере. Это позволяет масштабировать систему до сотен петабайт без заметного снижения производительности.

Структурированные и неструктурированные данные

Понятие «структурированных» и «неструктурированных» данных довольно относительное. Все файлы данных имеют ту или иную структуру, тип файла. Поэтому с этой точки зрения все файлы настроены. Когда говорят, что данные неструктурированы, значит, данные не хранятся в одной базе данных и содержат разные типы данных. Это набор файлов, не отличающихся друг от друга, созданных в разных приложениях и полученных из разных источников. Если вы откроете папку «Мои документы» на своем компьютере, это будет примерно там.

Хранилище объектов в первую очередь предназначено для работы с неструктурированными данными. Можно пометить неструктурированные объекты данных метаданными, определяющими их содержимое, и помочь быстро извлечь нужный объект из хранилища. В таком случае сами метаданные будут настроены, то есть они будут иметь стандартную форму, определенную в API. Это позволяет отслеживать и индексировать объекты без необходимости реализации внешних программ или баз данных. Использование метаданных открывает новые возможности для аналитики. Файлы (объекты) можно индексировать и искать в хранилище объектов, не зная структуры их содержимого или в какой программе они были созданы.

Защита данных

Нужно ли безопасно хранить репликацию данных в объектной системе? Да, это необходимо, но утроить дисковое пространство, как в блочной, не выйдет. Используем технологию Erasure Coding (AB), чтобы максимально использовать доступное дисковое пространство и сохранить данные. Проще говоря, можно назвать это новым поколением известных методов защиты данных RAID, которым требуются двойные или тройные резервные копии.

В методе EC объектные файлы разбиваются на куски (фрагменты). Для некоторых копии избыточности создаются в формате N+M. Например, если копии созданы в шести из десяти штук, то это в формате 10+6. Например, данные, необходимые для N дисков, резервные копии распределяются на N+M дисков (в данном случае 16). Если вы потеряете шесть дисков, оставшихся десяти достаточно для восстановления исходных данных. Таким образом, избыточное хранение не так велико, как RAID, и может выдерживать множественные сбои диска без риска потери данных. Подразделения EC могут выдерживать больше сбоев диска, чем массивы RAID-дисков. Однако масштабирование системы в петабайтах не стоит так дорого, как в файловых системах при покупке дисков.

Хранилище объектов обычно выбирается для данных WORM, которые записываются один раз, но читаются много раз (Write Once Read Many). Он не

подходит для всех объемов данных и сценариев использования, но, безусловно, имеет много применений.

Системы хранения объектов рекомендуется использовать в следующих случаях:

Длительное хранение статических данных, таких как различные нормативные документы (документы).

Резервные копии - дампы баз данных, файлы журналов и резервные копии существующего программного обеспечения.

Среда разработки DevOps - это единое общее пространство имен, к которому можно легко получить доступ с помощью простых запросов для управления различными объектами: большими базами данных, таблицами, изображениями, аудио- и видеофайлами и т. д. D.

Неструктурированные данные - медиафайлы, документы, изображения, аудио- и видеофайлы

Отрасли с большим объемом хранения данных, такие как:

Здравоохранение - электронные медицинские записи, цифровые изображения, данные системы ПГС;

Финансовые и торговые площадки, электронные письма, мессенджеры, оцифрованные архивы музеев, конструкторские бюро и др.

Статический веб-контент - большое количество HTML-файлов и изображений для сайтов.

Основываясь на этом, мы видим, что объектно-ориентированные системы хранения данных хорошо подходят для хранения больших неструктурированных данных и отвечают требованиям быстрого роста объемов данных, которые необходимо хранить, обрабатывать и анализировать в различных отраслях. Поэтому объемы систем хранения объектов растут значительно быстрее, чем в файловых системах.

Глава 3 Разработка алгоритма подбора оборудования для систем хранения данных

3.1 Спецификация задачи

Чтобы создать правильно работающий алгоритм выбора хранилища, необходимо собрать наиболее полную информацию о доступных типах оборудования, назначениях и типах соединений.

В данной ВКР мы создаем алгоритм подбора оборудования для IT отдела «Управляющей компании» Фармацевтической компании ОЗОН.

ОЗОН — один из лидеров отечественного фармацевтического рынка по производству генериков — аналогов патентованных готовых лекарственных средств (ГЛС). Стандарты GMP (Good Manufacturing Practice) позволяют компании производить препараты высокого качества, доступные широким слоям населения.

В настоящее время производство компании ОЗОН составляет около 500 млн. упаковок в год, а в продуктивном портфеле более 300 наименований GLS. С каждым годом растет количество выпускаемых лекарств, растут объемы производства. Поэтому необходимо частичное или полное обновление оборудования системы хранения данных.

3.2 Аппаратные комплектующие системы хранения данных

Мы начинаем создавать алгоритм из составных частей оборудования для систем хранения данных.

Как правило, система хранения включает в себя такие компоненты:

- Устройства хранения данных (дисковые массивы, ленточные библиотеки);
- Контроллер диска и задняя панель;
- Контроллер СХД;

- Сетевые адаптеры.

Рассмотрим подробнее каждый компонент.

3.2.1 Устройства хранения в СХД

Хранилище может работать с различными носителями, такими как магнитная лента, жесткие диски (жесткие диски) и твердотельные накопители (твердотельные накопители/nvme). Мы рассмотрим последние два, поскольку они распространены в большинстве систем как универсальные носители, а затем проанализируем типы дисковых массивов и ленточных библиотек.

Следует понимать, что драйверы в хранилищах устанавливают ограничение на производительность оборудования: система не может работать быстрее, чем диски ее. Медленнее — может быть.

Диски имеют множество важных параметров и функций, которые следует учитывать при создании хранилища, но основными функциями можно назвать тип интерфейса и форм-фактор.

Интерфейсы современных HDD и SSD

Интерфейс - это протокол связи между декомпозицией и вычислительными мощностями системы. Интерфейс является важным фактором, влияющим на параметры привода: от него зависит емкость, латентность, расширяемость, горячая взаимозаменяемость и, конечно, стоимость.

Изначально именно на жестких дисках появлялись интерфейсы SATA и SAS, но позже они стали стандартными и для твердотельных накопителей. Однако они не могут полностью реализовать максимальную возможность работоспособности жесткого диска, в связи с этим чтобы подсоединить твердое хранилище теперь чаще всего применяется внешний вид PCIe и соглашение NVMe. Стоит также отметить диски NL-SAS, которые по существу являются гибридом интерфейса SAS-интерфейса и SATA производительности [1], [3], [13].

Сравнение производительности жестких дисков и накопителей показано в таблице 4.

Таблица 4- Сравнение производительности жестких дисков и SSD-накопителей

Класс	HDD			SSD		
	SATA	SAS		SATA	SAS	PCIe
Накопитель	SATA	NL-SAS	SAS	SATA	SAS	NVMe
Надежность	Маленькая	Средняя	Высокая	Средняя	Высокая	Высокая
Оптимизация	Маленькая	Маленькая	Средняя	Высокая	Высокая	Высшая
Цена	Маленькая	Маленькая	Средняя	Средняя	Высокая	Высшая

Форм-фактор.

Все жесткие диски имеют аналогичную конструкцию подвижного элемента, поэтому их внешний корпус представляет собой прямоугольный корпус вида SFF (Small Form Factor, 2.5") или LFF (Large Form Factor, 3.5"). Каждый из этих типов корпусов может иметь разные интерфейсные разъемы.

Флеш-диски не имеют движущихся частей и поэтому реализуются в более разнообразных формах. Дополнительным стимулом для различных форм-факторов SSD стала разработка интерфейса PCIe, который добавлял параметры хранения непосредственно на серверную платформу.

Зависимость интерфейса от форм-фактора показано в таблице 5.

Таблица 5- Форм-фактор носителей информации

Форм-фактор	HDD	SSD
3.5" (LFF)	SATA, NL-SAS, SAS	—
2.5" (SFF)	SATA, NL-SAS, SAS	SATA, SAS, NVMe
M.2*	—	SATA, NVMe
Add-In Card (AIC)	—	NVMe
EDSFF	—	NVMe

Форм-фактор - очень динамичный параметр, который постоянно меняется и развивается как из-за изменений в интерфейсах, так и из-за изменений в подходах к хранению.

В современных хранилищах диски можно размещать как в главном корпусе, так и в корзинах для дисков (JBOD). Физически такие корзины представляют собой стеллаж, заполненный складскими помещениями. JBOF (Just a Bunch Of Flash) для дисков NVMe в настоящее время активно используется со специальными корзинами флешек. Например, OpenFlex Data24 от Western Digital.

Типы систем хранения данных (СХД)

Дисковые хранилища

Их выбирают для быстрой работы с данными и для создания промежуточных резервных копий, во избежание потерь данных.

Существуют следующие типы дисковых хранилищ:

-Хранилища для бизнес-данных (оборудование высокой производительности);

- Системы для резервного копирования (дисковые библиотеки);

- Системы длительного хранения архивов (системы CAS).

Ленточные системы хранения

Они используются в основном для того чтобы создавать резервные копии и большие архивы данных.

Рассмотрим самые распространенные типы ленточных хранилищ:

- отдельные диски;

- автозагрузчики (один диск и несколько разъемов для лент);

- ленточные библиотеки (более одного диска, множество слотов для лент).

3.2.2 Контроллер диска и задняя панель

Дисковый контроллер (HBA)

В системах хранения данных контроллер диска - это устройство, на котором сопоставленные диски передаются на вычислительные мощности

системы. Физически контроллер диска обычно представлен отдельной картой расширения, но также может быть в виде чипа, встроенного в материнскую плату. Проще говоря, контроллер диска позволяет просматривать все последующие уровни хранения и работать с дисками.

НВА (Host Bus Adapter, адаптер главной шины) — это своего рода контроллер диска, который позволяет системе видеть подключенные диски отдельно.

Контроллер, работающий с тремя протоколами (Triple-mode)

Существуют дисковые контроллеры, сочетающие возможность одновременной работы с тремя протоколами (SATA, SAS, NVMe). Такой подход полезен для обеспечения гибкости в проектировании систем хранения данных, поскольку он может подключаться как к традиционным дискам SATA, так и к сверхбыстрым дискам NVMe.

Бэмплейн и экспандер

В большинстве случаев диски в хранилище подключаются непосредственно через бэмплейн (backplane) - выделенную плату в дисковой стойке или на сервере для подключения к SAS, SATA, NVMe накопителей. Подключается к контроллеру диска. Сам контроллер диска обычно поддерживает прямое подключение ограниченного количества дисков. Используют экстендер, чтобы увеличить количество подключаемых дисков. В большинстве случаев это чип, установленный на бэмплейн.

3.2.3 Контроллер системы хранения

Контроллер системы

Это главный компонент управления хранилищем. Он представляет уровень обработки данных, отвечающий за создание дисковых массивов, расчет контрольных сумм, управление доступом и выполнение других вспомогательных операций.

Контроллер хранения является функциональным блоком, поэтому физически он может быть представлен совершенно другим набором оборудования.

RAID-контроллер

Теоретически понятие RAID-контроллера и контроллера хранилища часто представлено в виде дескрипторов. Это связано с тем, что оба выполняют свои основные функции, такие как создание и управление дисковым массивом. На практике контроллер RAID обычно относится к адаптеру, который подключается к серверу и создает набор сопоставленных дисков. Контроллер RAID в этом случае создает подсистему хранения для конкретного локального клиента (инициализатора), но не является встроенным хранилищем значений.

Контроллер хранилища как выделенный компьютер

Если мы говорим о системе хранения как о автономной инфраструктуре с сетевыми клиентами, то в этом случае контроллер хранения - это специализированный компьютер с программным обеспечением для управления. Он оснащен обычной материнской платой, центральным процессором и модулями ОЗУ. Такой контроллер можно разместить как в специальном корпусе, так и в обычном корпусе для монтажа, в стойку.

В некоторых решениях программное обеспечение для управления может выполнять функцию создания RAID-массива с дисков, устраняя необходимость в аппаратном RAID-контроллере. Этот механизм обычно используется в программно-определяемом хранилище, к которому системный контроллер может применяться на основе стандартных серверных платформ.

Двойной режим управления

Системы хранения могут быть одним контроллером, двумя контроллерами и многими контроллерами. Последние два варианта используются для повышения производительности и большей отказоустойчивости: в случае аппаратного сбоя активного контроллера второй узел "получает" работу всей системы без остановки.

В двух- и много - конфигурациях контроллера узлы соединяются по каналам связи. Эти каналы отличаются, отличия зависят от архитектуры и

функциональных особенностей системы, в которой они размещены. Например, в классической двойной системе управления на основе RAIDIX это:

- Соединение между узлами (Heartbeat) - соединение между узлами Ethernet для проверки работоспособности контроллеров. Декомпрессионное соединение между узлами.
- Синхронизация кеша (Cache sync) - соединение для синхронизации кеша на основе протоколов IB-SRP, iSCSI или SAS.

Но стоит заметить, что наличие двух контроллеров не значит использование двух отдельных дисковых носителей. Чаще всего дублирующие материнские платы контроллера размещены в разных отсеках внутри одной и той же платформы.

Вычислительные ресурсы

По мере увеличения количества выполняемых функций и увеличения количества используемых дисков системе требуется больше вычислительных ресурсов.

Например, для оптимальной производительности в одном хранилище контроллеров RAIDIX требуются два восьмиядерных процессора с частотой менее 2,1 ГГц и 48 ГБ оперативной памяти на 60 хранилищ, а для 600-дисковой системы - два восьмиядерных процессора объемом менее 3.5 ГГц и более 256 ГБ оперативной памяти.

Сейчас используются в качестве чипов Intel Xeon x86 и, реже, процессоров хранения EPYC AMD. Есть решения, собранные на других процессорах (на отечественном Эльбрусе, к примеру), но они не популярны на рынке [11] [4].

Сравнительные характеристики процессоров указаны в таблице 6.

Таблица 6- Характеристики специализированных процессоров Intel и AMD

	Intel Xeon Scalable				AMD EPYC	
	Bronze	Silver	Gold	Platinum	1- Socket	2- Socket
Кол-во сокетов	От 1 до 2	От 1 до 2	От 2 до 4	От 2 до 8	1	2
Кол-во ядер	От 6 до 8	От 4 до 12	От 4 до 22	От 4 до 28	От 16 до 32	От 16 до 32
Частота процессора	1.7 ГГц	От 1.8 до 2.6 ГГц	От 1.9 до 3.5 ГГц	От 2 до 3.6 ГГц	От 2 до 2.4 ГГц	От 2 до 2.4 ГГц
Тип поддерживаемой памяти	DDR4- 2133	DDR4- 2400	DDR4- 2666	DDR4- 2666	DDR4- 2666	DDR4- 2666
Кол-во линий PCIe	48				128	

На объем вычислительных мощностей сильно влияют бизнес-требования к инфраструктуре хранения, характер и интенсивность нагрузок, а также доступная сетевая среда.

3.2.4 Сетевые адаптеры

Сетевые адаптеры (адаптеры хоста) являются последним звеном в цепочке клиентских коммуникаций. Это устройство позволяет системе хранения подключаться к серверам, рабочим станциям и другим компонентам сетевой инфраструктуры “к внешнему миру”. Сетевой адаптер - это карта с интерфейсными портами (Ethernet, FC, IB, SAS), которые используют разъем на материнской плате или непосредственно входят в нее [5].

Некоторые адаптеры хоста могут одновременно иметь несколько интерфейсов. Например, у Mellanox есть адаптер, работающий с интерфейсами Infiniband и Ethernet. [14] [15].

Современные адаптеры хоста и их характеристики указаны в таблице 7.

Таблица 7- Особенности современных адаптеров хоста в СХД

Тип интерфейса	Блочные протоколы	Файловые протоколы	Скорость	Популярные бренды
FC	Fibre Channel FC-NVMe	—	8Gb/16Gb/32Gb	Marvell QLogic Broadcom ATTO
Ethernet	iSCSI, iSER, FCoE, NVMe-oF	SMB, NFS, AFP, FTP	10Gb/25Gb/40Gb/50Gb/100Gb/200Gb	Broadcom Mellanox
Infiniband	IB-SRP	—	20Gb/40Gb/56Gb/100Gb	Mellanox
SAS	SAS	—	12Gb	Broadcom Adaptec

Выбор адаптера зависит от сетевой инфраструктуры, запланированных задач и конфигурации оборудования для хранения.

3.3 Типы назначений систем хранения данных

Согласно общепринятым стандартам, по назначению существуют следующие виды хранения:

- High-end midrange storage – высокопроизводительные системы хранения используются для критически важных задач в средних и малых предприятиях и центрах обработки данных. Используется с использованием технологий SAN Storage, SAS HDD, SSD.

– Entry storage – блоки хранения входного и среднего уровня используются для хранения файлов и блоков в компаниях малого и среднего уровня.

– Backup/Recovery, Archive storage – дисковые системы хранения данных для сбора резервных копий изображений и данных, а также ленточные системы для длительного хранения резервных копий большого объема и архивных резервных копий. Используется для архивов файлов, видеонаблюдения и архивов резервных данных. Используется в сочетании с технологией NAS, DAS / LAN, LTU tape media и utoloder, external drives, libraries.

– Software-Defened storage – в частном или частном публичном облаке с виртуальным хранилищем на сервере и распределением уровня доступа. Он используется в качестве виртуального хранилища для управления неиспользуемым дисковым пространством на сервере и использования совместного использования данных, когда физическое хранилище не используется, и при необходимости для гибкого масштабирования.

Сравнение систем хранения данных по назначениям перечислены в таблице 8.

Таблица 8- Сравнительная таблица назначений систем хранения

Назначение СХД	Тип схд	Инфраструктура доступа	Интерфейс	Протокол
High-end midrange storage	Дисковый массив для рабочих станций	NAS	SAS	SAS
Entry storage	Дисковый массив для рабочих станций	DAS NAS	FC Ethernet	Fibre Channel, iSCSI, iSER, FCoE
Backup/Recovery , Archive storage	Дисковая библиотека для резервных копий	NAS+DAS	FC	Fiber Channel, iSCSI
Software-Defened storage	Виртуальное хранилище в облаке	SAN		iSCSI

Окончательный выбор оборудования зависит от задач и особенностей компании.

По выше перечисленным данным собирается следующий алгоритм подбора оборудования для систем хранения данных:

- В первую очередь выбирается инфраструктура доступа или тип системы хранения данных.
 - Зная тип СХД подбирается оптимальный RAID.
 - Далее в зависимости от поставленной задачи получаем примерный IOPS (количество операций ввода-вывода в секунду).
 - Теперь определяемся с объемом хранилища данных (с запасом на 5-10 лет, а также оно должно удовлетворять условию IOPS).
 - Далее выбирается наличие карты дискового контроля.
 - Контроллер системы подбирается под конкретную задачу.
 - В зависимости от мощности выбираемой системы выбирается встроенный или аппаратный RAID.
 - Выбирается количество контроллеров управления.
 - Под задачу выбирается, центральный процессор и объем ОЗУ.
 - В заключение выбирается (-ются) сетевой (-ые) адаптер (-ы).
- Алгоритм подбора оборудования в схеме представлен на рисунке 7.

По разработанному алгоритму была реализована простейшая программа для подбора оборудования.

Программа написана на языке программирования C#. Во время разработки использовались следующие библиотеки, функции и методы [2] [16] [17]:

- System;
- System.Collections.Generic;
- System.ComponentModel;
- System.Data;
- System. Drawing;
- System.Linq;
- System. Text;
- System.Threading.Tasks;
- System.Windows.Forms.

На вход программа принимает три аргумента:

-Тип системы хранения данных (см рисунок 8)

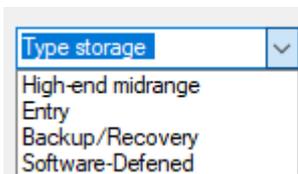


Рисунок 8 - Варианты выбора типа СХД

- Вариацию RAID (см рисунок 9)



Рисунок 9 - Выбор RAID в программе

- Предполагаемую нагрузку (см рисунок 10)

- 10 000 IOPS
- 20 000 IOPS
- 30 000 IOPS

Рисунок 10- Выбор нагрузки в программе

Предполагаемая задача, подобрать оборудование для компании, которая будет добавлять новых клиентов и хранить из персональные данные [19].

Решение показано на рисунке 11.

Инфраструктура доступа : NAS+DAS

Интерфейс : FC

Количество контроллеров : 1

Центральный процессор : Intel Xeon Scalable Silver

Оперативная память : 24 Gb,DDR4-2400 (4x 8 Gb,DDR4-2400)

Жесткие диски (без учета RAID) : 24x HDD SAS 2,5', 900Gb, 12 Gb/s, 15k RPM, 256 cache

Жесткие диски (с учетом RAID) : На чтение IOPS= 10 100, на запись IOPS= 7 080, рабочий объем 10,1 Тб

Backup/Recovery

RAID 1

10 000 IOPS

20 000 IOPS

30 000 IOPS

Подобрать

Рисунок 11- Подбор оборудования для Recovery, Archive storage

Для хранения персональных данных лучше всего подойдет СХД с назначением Архивации и долговременного хранения данных. RAID 1 был выбран по причине безопасности хранения информации, можно было выбрать RAID 5 или RAID 6, но это бы замедлило систему. Но если компания не большая или не требуется добавлять одновременно много клиентов, то лучше выбрать RAID 5.

Другая предполагаемая задача, подобрать оборудование для фирмы, которой важна скорость записи, перезаписи и удаление данных в СХД.

Решение показано на рисунке 12.

Инфраструктура доступа : NAS

Интерфейс : SAS

Количество контроллеров : 2+

Центральный процессор : Intel Xeon Scalable Gold

Оперативная память : 128 Gb,DDR4-2666 (4x 32 Gb,DDR4-2666)

Жесткие диски (без учета RAID) : 80x HDD SAS 2,5', 900Gb, 12 Gb/s, 15k RPM, 256 cache

Жесткие диски (с учетом RAID) : На чтение IOPS= 34 800, на запись IOPS= 32 400, рабочий объем 57,6 Tb

High-end midrange

RAID 3

10 000 IOPS

20 000 IOPS

30 000 IOPS

Подобрать

Рисунок 12- Подбор оборудования High-end midrange storage

Для скорости записи и чтения лучше всего подойдет СХД с назначением Высокого класса поскольку считается самой скоростной из типов. RAID 3 был выбран по причине скорости и рабочего объема, можно было выбрать RAID 0 с максимальной скоростью, но это бы он имеет наименьшую безопасность и при выходе из строя диска мы потеряем все данные что на нем хранились. RAID 5 или RAID 6 мы не будем использовать из-за очень низкого IOPS.

На данный момент готова бета-версия программы, которая корректно подбирает минимальный набор оборудования исходя из заданных параметров. В последствии алгоритм будет дорабатываться. На данный момент, он не учитывает контроль хранилища, контроллер системы и протокол. В будущих версиях всё это будет добавлено и учтено. Текущая версия работает без сбоев, оборудование подбирает корректно [18].

Заключение

Современное оборудование для хранения данных можно рассматривать как цепочку слоев, в которых данные передаются и обрабатываются между местом их хранения и клиентом декомпозиции. Этот внешний вид процесса дает хорошее представление о том, что производительность может превратиться в «горлышко бутылки», какой компонент оборудования обладает не нужными в данной ситуации свойствами и может снизить общую стоимость решения.

На основании проведенного исследования, мы выяснили как классифицируются системы хранения данных, а также как делится их программная часть. На основании анализа данных о видах систем хранения данных мы смогли составить основной список необходимого оборудования для хранения данных. При решении задачи по подбору систем хранения данных была разработана демонстрационная версия ПО для подбора комплектующих для создания СХД.

В заключении отметим, что использование этой информации и демонстрационного ПО поможет нам ориентироваться в выборе оборудования для хранения данных и упростит процесс отбора компонентов при обновлении и обновлении существующих систем в фармацевтической компании «ОЗОН».

Список используемой литературы

1. Анализ подходов, делающих СХД на "Эльбрусе" конкурентоспособными / Я. А. Брыш, М. А. Копосов, Г. Н. Панов, Г. Н. Сергеев // Наноиндустрия. – 2020. – Т. 13. – № S5-1(102). – С. 102-104. – DOI 10.22184/1993-8578.2020.13.5s.102.104. – EDN VXAXLM.

2. Бурдин, И. М. Выбор рациональной структуры для хранения данных и разработка web-приложения / И. М. Бурдин // Международный молодёжный конкурс научных проектов «Стираем границы» : сборник материалов Международного молодёжного конкурса научных проектов, Москва, 20–21 октября 2021 года. – Москва: Федеральное государственное бюджетное образовательное учреждение высшего образования "Российский государственный университет имени А.Н. Косыгина (Технологии. Дизайн. Искусство)", 2021. – С. 47-50. – EDN QYTLTC.

3. Бережной, А. Стандарт SCSI как основа СХД / А. Бережной // Системный администратор. – 2017. – № 5(174). – С. 16-20. – EDN YODCIN

4. Гантимуров, А. П. Способ обеспечения отказоустойчивости и определения периодичности создания точек восстановления данных СХД / А. П. Гантимуров // Автоматизация в промышленности. – 2020. – № 6. – С. 55-59. – DOI 10.25728/avtprom.2020.06.12. – EDN GGXDMZ.

5. Дубровин, А. А. Многоступенчатая архитектура контроллера твердотельного накопителя с повышенной пропускной способностью / А. А. Дубровин // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). – 2020. – № 2. – С. 158-164. – DOI 10.31114/2078-7707-2020-2-158-164. – EDN TTYUYUS

6. Завгородний, В. И. Выбор системы хранения данных / В. И. Завгородний // Современная математика и концепции инновационного математического образования. – 2020. – Т. 7. – № 1. – С. 302-310. – EDN RZFBZH.

7. Логиновский, О. В. Построение современных корпоративных информационных систем / О. В. Логиновский, А. Л. Шестаков, А. А. Шинкарев // Управление большими системами: сборник трудов. – 2019. – № 81. – С. 113-146. – DOI 10.25728/ubs.2019.81.5. – EDN TSGJZM.

8. Лисецкий, Ю. М. Корпоративные системы хранения данных. Построение / Ю. М. Лисецкий // Управляющие системы и машины. – 2013. – № 6(248). – С. 68-71. – EDN ТВНМЕV. URL: https://www.elibrary.ru/download/elibrary_22633439_55359475.pdf.

9. Митник, Б. Использование твердотельных накопителей SSD для повышения производительности СХД / Б. Митник // Журнал сетевых решений LAN. – 2015. – № 4. – С. 51-54. – EDN ТРКVСН.

10. Майер, С. Ф. Языки программирования (С#) : Учебно-методическое пособие / С. Ф. Майер. – Ростов-на-Дону : Южный федеральный университет, 2020. – 184 с. – EDN DXSAIA.

11. Пойманова, Е. Д. Архитектура многоуровневой системы хранения данных / Е. Д. Пойманова, П. С. Летуновская, Б. С. Шром // Обработка, передача и защита информации в компьютерных системах '22 : Сборник докладов Второй Международной научной конференции, Санкт-Петербург, 11–15 апреля 2022 года. – Санкт-Петербург: Санкт-Петербургский государственный университет аэрокосмического приборостроения, 2022. – С. 282-286. – DOI 10.31799/978-5-8088-1701-2-2022-2-282-286. – EDN ТКZQEW.

12. Свидетельство о государственной регистрации программы для ЭВМ № 2019661945 Российская Федерация. Программа прогноза наращивания ёмкости систем хранения данных : № 2019619010 : заявл. 22.07.2019 : опубл. 12.09.2019 / Е. Д. Пойманова, Т. М. Татарникова, Н. В. Яготинцева. – EDN EFWWYU.

13. Свидетельство о государственной регистрации программы для ЭВМ № 2020610585 Российская Федерация. Блочная СХД для систем виртуализации : № 2019667084 : заявл. 20.12.2019 : опубл. 16.01.2020 / С. Б.

Полуновский ; заявитель Общество с ограниченной ответственностью «НПК «Антей». – EDN PQUKNK.

14. [Электронный ресурс] URL:<http://eadgene.org/index.php/term/30-knizhnaya-biblioteka,2347-sistemy-hraneniya-dannyh.xhtml>.

15. Успенский, М. Б. Обнаружение неисправностей в системах хранения данных для предотвращения повреждения или потери информации / М. Б. Успенский // Состояние и перспективы развития современной науки по направлению "Информационная безопасность" : сборник статей II Всероссийской научно-технической конференции, Анапа, 19–20 марта 2020 года / Федеральное государственное автономное учреждение "Военный инновационный технополис "ЭРА". – Анапа: Федеральное государственное автономное учреждение "Военный инновационный технополис "ЭРА", 2020. – С. 384-391. – EDN PRYQSO.

16. Entwicklung eines Softwareprodukts zur Bestimmung des Druckes über die Länge der Gasleitung in der Programmiersprache C# / В. А. Sokolov, N. E. Bessarabov, T. A. Mirnoshnichenko, A. M. Podkolzin // Проблемы научной мысли. – 2021. – Vol. 9. – No 2. – P. 73-76. – EDN VETNGI.

17. Čižmárik, A. SharpDetect: Dynamic Analysis Framework for C#.NET Programs / A. Čižmárik, P. Parížek // Lecture Notes in Computer Science. – 2020. – Vol. 12399 LNCS. – P. 298-309. – DOI 10.1007/978-3-030-60508-7_16. – EDN UXVGRW.

18. Belyaev, M. V. Modeling of library functions in an industrial static code analyzer / M. V. Belyaev, E. S. Romanenkov, V. N. Ignatyev // Proceedings of the Institute for System Programming of the RAS. – 2020. – Vol. 32. – No 3. – P. 21-32. – DOI 10.15514/ISPRAS-2020-32(3)-2. – EDN CCCCWL.

19. Introduction of rapid prototyping in solving applied problems in production / V. A. Brykin, A. P. Voroshilin, P. A. Uhov, A. V. Ripetskiy // Periodico Tche Quimica. – 2020. – Vol. 17. – No 35. – P. 354-366. – EDN PMQIOG.

20. Bogatyrev, V. A. Reliability of Computer Systems during Physical and Informational Recovery of Duplicated Memory / V. A. Bogatyrev, S. V. Bogatyrev,

A. V. Bogatyrev // CEUR Workshop Proceedings, Yalta, Crimea, 20–22 сентября 2021 года. – Yalta, Crimea, 2021. – P. 274-279. – EDN XLQRVX.

21. От хранения данных к управлению информацией : перевод с английского / ред. : Г. Сомасундарам, Алок Шривастава. – Москва [и др.] : Питер, 2010. – 522 с. – ISBN 978-5-4237-0008-9. – EDN QMVRAD.

Приложение А

Листинг программного кода

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace diplom
{
    public partial class Form1 : Form
    {
        private string a;
        private string Infr;
        private string inter;
        private string conter;
        private string proc;
        private string ozu;
        private string disk;
        private object gg;
        private string raid;

        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
        }
        private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
        {
        }
        private void label1_Click(object sender, EventArgs e)
        {
            string g = comboBox1.Text;
            string gg = comboBox2.Text;
            if (g == "Entry")
            {
                Infr = "Инфраструктура доступа : DAS";
                inter = "Интерфейс : Ethernet";
                conter = "Количество контроллеров : 1";
                proc = "Центральный процессор : Intel Xeon Scalable Silver";
                if (checkBox3.Checked)
                {
                    ozu = "Оперативная память : 64 Gb,DDR4-2400 (2x 32 Gb,DDR4-2400)";
                    disk = "80x HDD SAS 2,5' , 900Gb, 12 Gb/s, 15k RPM, 256 cache";
                    if (gg == "RAID 0")
                    {
                        raid = "На чтение IOPS= 34 800, на запись IOPS = 43 200, рабочий объем 72 Tb ";
                    }
                    if (gg == "RAID 1")
                    {
                        raid = "На чтение IOPS= 34 800, на запись IOPS= 21 600, рабочий объем 36 Tb";
                    }
                    if (gg == "RAID 3")
                    {
                        raid = "На чтение IOPS= 34 800, на запись IOPS= 32 400, рабочий объем 57,6 Tb";
                    }
                    if (gg == "RAID 5")
                    {
                        raid = "На чтение IOPS= 34 800, на запись IOPS= 395, рабочий объем 68,5 Tb";
                    }
                    if (gg == "RAID 6")
                    {
                        raid = "На чтение IOPS= 34 800, на запись IOPS= 420, рабочий объем 65,6 Tb";
                    }
                }
            }
            else
            {
                if (checkBox2.Checked)
                {
                    ozu = "Оперативная память : 48 Gb,DDR4-2400 (3x 16 Gb,DDR4-2400)";
                    disk = "50x HDD SAS 2,5' , 900Gb, 12 Gb/s, 15k RPM, 256 cache";
                    if (gg == "RAID 0")
                }
            }
        }
    }
}
```

Продолжение Приложения А

```
{
  raid = "На чтение IOPS= 20 800, на запись IOPS = 29 500, рабочий объем 45 Tb ";
}
if (gg == "RAID 1")
{
  raid = "На чтение IOPS= 20 800, на запись IOPS= 14 750, рабочий объем 22,5 Tb";
}
if (gg == "RAID 3")
{
  raid = "На чтение IOPS= 20 800, на запись IOPS= 22 125, рабочий объем 36 Tb";
}
if (gg == "RAID 5")
{
  raid = "На чтение IOPS= 20 800, на запись IOPS= 375, рабочий объем 42,8 Tb";
}
if (gg == "RAID 6")
{
  raid = "На чтение IOPS= 20 800, на запись IOPS= 410, рабочий объем 41 Tb";
}
}
else
{
  if (checkBox1.Checked)
  {
    ozu = "Оперативная память : 24 Gb,DDR4-2400 (4x 8 Gb,DDR4-2400)";
    disk = "24x HDD SAS 2,5' , 900Gb, 12 Gb/s, 15k RPM, 256 cache";
    if (gg == "RAID 0")
    {
      raid = "На чтение IOPS= 10 100, на запись IOPS = 14 100, рабочий объем 20,1 Tb ";
    }
    if (gg == "RAID 1")
    {
      raid = "На чтение IOPS= 10 100, на запись IOPS= 7 080, рабочий объем 10,1 Tb";
    }
    if (gg == "RAID 3")
    {
      raid = "На чтение IOPS= 10 100, на запись IOPS= 10 575, рабочий объем 16,8 Tb";
    }
    if (gg == "RAID 5")
    {
      raid = "На чтение IOPS= 10 100, на запись IOPS= 490, рабочий объем 19,3 Tb";
    }
    if (gg == "RAID 6")
    {
      raid = "На чтение IOPS= 10 100, на запись IOPS= 500, рабочий объем 18,4 Tb";
    }
  }
  else
  {
    ozu = " ? IOPS ?";
    disk = " ? IOPS ?";
  }
}
}
}
if (g == "High-end midrange")
{
  Infr = "Инфраструктура доступа : NAS";
  inter = "Интерфейс : SAS";
  conter = "Количество контроллеров : 2+";
  proc = "Центральный процессор : Intel Xeon Scalable Gold";
  if (checkBox3.Checked)
  {
    ozu = "Оперативная память : 128 Gb,DDR4-2666 (4x 32 Gb,DDR4-2666)";
    disk = "80x HDD SAS 2,5' , 900Gb, 12 Gb/s, 15k RPM, 256 cache";
    if (gg == "RAID 0")
    {
      raid = "На чтение IOPS= 34 800, на запись IOPS = 43 200, рабочий объем 72 Tb ";
    }
    if (gg == "RAID 1")
    {
      raid = "На чтение IOPS= 34 800, на запись IOPS= 21 600, рабочий объем 36 Tb";
    }
    if (gg == "RAID 3")
    {
      raid = "На чтение IOPS= 34 800, на запись IOPS= 32 400, рабочий объем 57,6 Tb";
    }
  }
}
```

Продолжение Приложения А

```
}
if (gg == "RAID 5")
{
raid = "На чтение IOPS= 34 800, на запись IOPS= 395, рабочий объем 68,5 Tb";
}
if (gg == "RAID 6")
{
raid = "На чтение IOPS= 34 800, на запись IOPS= 420, рабочий объем 65,6 Tb";
}
}
else
{
if (checkBox2.Checked)
{
ozu = "Оперативная память : 96 Gb,DDR4-2666 (3x 32 Gb,DDR4-2666)";
disk = "50x HDD SAS 2,5' , 900Gb, 12 Gb/s, 15k RPM, 256 cache";
if (gg == "RAID 0")
{
raid = "На чтение IOPS= 20 800, на запись IOPS = 29 500, рабочий объем 45 Tb ";
}
if (gg == "RAID 1")
{
raid = "На чтение IOPS= 20 800, на запись IOPS= 14 750, рабочий объем 22,5 Tb";
}
if (gg == "RAID 3")
{
raid = "На чтение IOPS= 20 800, на запись IOPS= 22 125, рабочий объем 36 Tb";
}
if (gg == "RAID 5")
{
raid = "На чтение IOPS= 20 800, на запись IOPS= 375, рабочий объем 42,8 Tb";
}
if (gg == "RAID 6")
{
raid = "На чтение IOPS= 20 800, на запись IOPS= 410, рабочий объем 41 Tb";
}
}
}
else
{
if (checkBox1.Checked)
{
ozu = "Оперативная память : 64 Gb,DDR4-2666 (4x 16 Gb,DDR4-2666)";
disk = "24x HDD SAS 2,5' , 900Gb, 12 Gb/s, 15k RPM, 256 cache";
if (gg == "RAID 0")
{
raid = "На чтение IOPS= 10 100, на запись IOPS = 14 100, рабочий объем 20,1 Tb ";
}
if (gg == "RAID 1")
{
raid = "На чтение IOPS= 10 100, на запись IOPS= 7 080, рабочий объем 10,1 Tb";
}
if (gg == "RAID 3")
{
raid = "На чтение IOPS= 10 100, на запись IOPS= 10 575, рабочий объем 16,8 Tb";
}
if (gg == "RAID 5")
{
raid = "На чтение IOPS= 10 100, на запись IOPS= 490, рабочий объем 19,3 Tb";
}
if (gg == "RAID 6")
{
raid = "На чтение IOPS= 10 100, на запись IOPS= 500, рабочий объем 18,4 Tb";
}
}
}
else
{
ozu = " ? IOPS ?";
disk = " ? IOPS ?";
}
}
}
}
if (g == "Backup/Recovery")
{
Infr = "Инфраструктура доступа : NAS+DAS";
inter = "Интерфейс : FC";
}
```

Продолжение Приложения А

```
conter = "Количество контроллеров : 1";
proc = "Центральный процессор : Intel Xeon Scalable Silver";
if (checkBox3.Checked)
{
    ozu = "Оперативная память : 64 Gb,DDR4-2400 (2x 32 Gb,DDR4-2400)";
    disk = "80x HDD SAS 2,5' , 900Gb, 12 Gb/s, 15k RPM, 256 cache";
    if (gg == "RAID 0")
    {
        raid = "На чтение IOPS= 34 800, на запись IOPS = 43 200, рабочий объем 72 Tb ";
    }
    if (gg == "RAID 1")
    {
        raid = "На чтение IOPS= 34 800, на запись IOPS= 21 600, рабочий объем 36 Tb";
    }
    if (gg == "RAID 3")
    {
        raid = "На чтение IOPS= 34 800, на запись IOPS= 32 400, рабочий объем 57,6 Tb";
    }
    if (gg == "RAID 5")
    {
        raid = "На чтение IOPS= 34 800, на запись IOPS= 395, рабочий объем 68,5 Tb";
    }
    if (gg == "RAID 6")
    {
        raid = "На чтение IOPS= 34 800, на запись IOPS= 420, рабочий объем 65,6 Tb";
    }
}
else
{
    if (checkBox2.Checked)
    {
        ozu = "Оперативная память : 48 Gb,DDR4-2400 (3x 16 Gb,DDR4-2400)";
        disk = "50x HDD SAS 2,5' , 900Gb, 12 Gb/s, 15k RPM, 256 cache";
        if (gg == "RAID 0")
        {
            raid = "На чтение IOPS= 20 800, на запись IOPS = 29 500, рабочий объем 45 Tb ";
        }
        if (gg == "RAID 1")
        {
            raid = "На чтение IOPS= 20 800, на запись IOPS= 14 750, рабочий объем 22,5 Tb";
        }
        if (gg == "RAID 3")
        {
            raid = "На чтение IOPS= 20 800, на запись IOPS= 22 125, рабочий объем 36 Tb";
        }
        if (gg == "RAID 5")
        {
            raid = "На чтение IOPS= 20 800, на запись IOPS= 375, рабочий объем 42,8 Tb";
        }
        if (gg == "RAID 6")
        {
            raid = "На чтение IOPS= 20 800, на запись IOPS= 410, рабочий объем 41 Tb";
        }
    }
    else
    {
        if (checkBox1.Checked)
        {
            ozu = "Оперативная память : 24 Gb,DDR4-2400 (4x 8 Gb,DDR4-2400)";
            disk = "24x HDD SAS 2,5' , 900Gb, 12 Gb/s, 15k RPM, 256 cache";
            if (gg == "RAID 0")
            {
                raid = "На чтение IOPS= 10 100, на запись IOPS = 14 100, рабочий объем 20,1 Tb ";
            }
            if (gg == "RAID 1")
            {
                raid = "На чтение IOPS= 10 100, на запись IOPS= 7 080, рабочий объем 10,1 Tb";
            }
            if (gg == "RAID 3")
            {
                raid = "На чтение IOPS= 10 100, на запись IOPS= 10 575, рабочий объем 16,8 Tb";
            }
            if (gg == "RAID 5")
            {
                raid = "На чтение IOPS= 10 100, на запись IOPS= 490, рабочий объем 19,3 Tb";
            }
            if (gg == "RAID 6")
            {
            }
        }
    }
}
```

Продолжение Приложения А

```
        {
            raid = "На чтение IOPS= 10 100, на запись IOPS= 500, рабочий объем 18,4 Тб";
        }
    }
    else
    {
        ozu = " ? IOPS ?";
        disk = " ? IOPS ?";
    }
}
}
}
if (g == "Software-Defened ")
{
    label6.Text = " ";
    label7.Text = " ";
    Infr = "Наиболее известные решения: VMware vSAN, HPE StoreVirtual VSA, NetApp ONTAP
Select, ";
    label8.Text = "EMC ScaleIO, Huawei Fusion Storage, StarWind Virtual SAN, Datacore SANsymphony";
    inter = "Для любого из решений нужно: Три сервера и три локальных устройств хранения,
минимум";
    conter = "Гипервизор (для каждого решения свой)";
    proc = "Вся подробная информация содержится в таблице 6, приложения 2";
}
if (g != "Software-Defened ")
{
    label8.Text = " ";
    label6.Text = "Жесткие диски (без учета RAID) : " + disk;
    label7.Text = "Жесткие диски (с учетом RAID) : " + raid;
}
label1.Text = Infr;
label2.Text = inter;
label3.Text = ozu;
label4.Text = conter;
label5.Text = proc;
}
private void checkedListBox1_SelectedIndexChanged(object sender, EventArgs e)
{
}
private void checkedListBox1_SelectedIndexChanged_1(object sender, EventArgs e)
{
}
private void domainUpDown1_SelectedItemChanged(object sender, EventArgs e)
{
}
private void label4_Click(object sender, EventArgs e)
{
}
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
}
}
internal struct NewStruct
{
    public string V;
    public string V1;

    public NewStruct(string v, string v1)
    {
        V = v;
        V1 = v1;
    }

    public override bool Equals(object obj)
    {
        return obj is NewStruct other &&
            V == other.V &&
            V1 == other.V1;
    }

    public override int GetHashCode()
    {
        int hashCode = -678510503;
        hashCode = hashCode * -1521134295 + EqualityComparer<string>.Default.GetHashCode(V);
        hashCode = hashCode * -1521134295 + EqualityComparer<string>.Default.GetHashCode(V1);
        return hashCode;
    }
}
```

Продолжение Приложения А

```
public void Deconstruct(out string v, out string v1)
{
    v = V;
    v1 = V1;
}

public static implicit operator (string V, string V1)(NewStruct value)
{
    return (value.V, value.V1);
}
public static implicit operator NewStruct((string V, string V1) value)
{
    return new NewStruct(value.V, value.V1);
}
public static implicit operator Array(NewStruct v)
{
    throw new NotImplementedException();
}
}
```

Приложение Б

Основные характеристики SDS решений

Таблица Б1- Основные характеристики SDS решений

	VMware vSAN	HPE StoreVirtual VSA	NetApp Data ONTAP Select	EMC ScaleIO	Huawei Fusion Storage	StarWind Virtual SAN	Datacore SANsymphony
Поддерживаемые гипервизоры	ESXI	ESXI, Hyper-V, KVM	ESXI	ESXI, Hyper-V, KVM, Xen	ESXi, Hyper-V, KVM, Xen, FusionCompute	ESXi, Hyper-V, KVM, Xen	ESXi, Hyper-V, KVM, XEN и др. (Листа совместимости нет)
Удаленная асинхронная репликация	Да, через vSphere Replication с RPO от 5 минут	Да, несколько площадок, RPO 15 минут	Да	Да	Нет	Да	Да (Двухнаправленная, интеграция с VMware)
Поддержка снапшотов	Да	Да	Да	Да	Да	Да	Да
Работа со снапшотами	Чтение/Запись	Чтение/Запись	Чтение/Запись	Чтение/Запись	-	Чтение/Запись	Чтение/Запись
Шифрование данных	Да. На уровне vSphere (с 6.5), SED диски RPQ, на уровне vSAN public beta	Нет	Нет	Да	Есть	Нет	На уровне блочном не шифрует. При файловом доступе и асинхронной репликации - да.
Дедупликация	Только на All Flash (4K, near-line)	Нет	Да	Нет	Только для объектного доступа	Да	Да
Сжатие	Только на All Flash (4K, LZ4, near-line)	Нет	Да	Нет	Нет	Нет	Да
Flash-Cache	Да, обязательное требование	Да, на RAID-контроллере	Да	Да	Да	Да	Да (RAM)
Tiering	Нет	Да, вкл/выкл на уровне каждого тома	Нет	Нет	Да	Нет	Да (15 уровней, онлайн)

Продолжение Приложения Б

Продолжение таблицы Б1

Автоматическая балансировка нагрузки между узлами	Да	Да, все узлы Active-Active, Multi Pathing	Нет	Нет.	Да	Да	Да
QoS (Quality of Service)	Да	Да	Нет	Да	Да	Да	Да
Файловый доступ	Нет	Нет	CIFS, NFS	NFS	NFS, CIFS, FTP, HDFS	SMB3, NFS	SMB3, CIFS, NFS
Схема лицензирования	За CPU на сервере или конкурентные подключения (для VDI) или по VM (ROBO, <25VM на площадку)	За узел. Ограничение по объему на узел	За узел	По объему сырого пространства	По объему сырого пространства	По объёму/узлам	За узел + ТВ полезной емкости
Особенности		Одновременная синхронная и асинхронная репликация томов. Поддержка VMware Metro Storage Cluster			Есть поддержка объектного доступа по протоколам S3 и SWIFT, Интеграция с OpenStack	Virtual Tape Library (VTL)	Виртуализация физических СХД (SAN, NAS, DAS); презентация виртуальных дисков любым внешним хостам/гипервизорам; RWA – преобразование случайной записи в последовательную; Parallel IO параллельная обработка