На правах рукописи

Лукьянов Николай Михайлович

Разработка архитектуры и методов организации слабосвязанных архивных систем для автоматизации проектирования

05.13.12 – Системы автоматизации проектирования
(приборостроение)

Автореферат
диссертации на соискание учёной степени
кандидата технических наук

Санкт-Петербург
2011

Работа выполнена в Санкт-Петербургском государственном университете информационных технологий, механики и оптики.

Научный руководитель: Кандидат технических наук, доцент
Тимченко Борис Дмитриевич

Официальные оппоненты: Доктор технических наук, профессор
Водяхо Александр Иванович
Санкт-Петербургский государственный электротехнический университет.

Доктор технических наук, профессор
Арустамов Сергей Аркадьевич
Санкт-Петербургский государственный университет информационных технологий, механики и оптики.

Ведущая организация: Санкт-Петербургский институт информатики и автоматизации РАН

Защита состоится 31 мая 2011 г. в 15-50 минут на заседании диссертационного совета Д 212.227.05 Санкт-Петербургского по адресу: 197101, Санкт-Петербург, Кронверкский пр., д. 49.

С диссертацией можно ознакомиться в библиотеке Санкт-Петербургского государственного университета информационных технологий, механики и оптики (национальный исследовательский университет).

Автореферат разослан 29 апреля 2011 г.

Учёный секретарь диссертационного
совета Д.212.227.05 В.И. Поляков

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы.

В современных ИТ системах постоянно растет значимость ресурсов хранения, архивных систем. Объективные предпосылки этого многочисленны: государственное регулирование в части электронного архивирования, мультимедийные файловые форматы во всех сферах деятельности, специальные системы обмена контентом и много другое. Можно считать общепринятой оценку ожидаемого удвоения годового объема сохраняемых данных, три четверти которого составляет неструктурированные файловые данные. Поэтому в последнее время ключевым моментом в развитии ИТ инфраструктур становятся не серверы и средства передачи данных, а средства их хранения, обеспечивающие также надежный доступ к ним.

Ведущие мировые производители систем хранения, такие как EMC, IBM, HP и другие ведут активные разработки в этой области. В центре внимание ведущих компаний находятся высокопроизводительные системы среднего и высокого класса, которые имеют высокоскоростные каналы связи с серверами и архитектурно функционируют в масштабе локальной сети. Сфера их применения – дата центры.

Другим типом систем хранения являются слабосвязанные системы, которые территориально распределены, то есть существуют в масштабе “широких” сетей, строятся из гетерогенных элементах и функционируют в условия ограниченной пропускной способности каналов передачи данных. Системы этого класса известны гораздо меньше. Их архитектура нуждается в исследовании и дальнейшей разработки в связи с растущими требованиями к любым видам информационного обслуживания. Мы считаем, что важнейшим качеством таких систем является их технико-экономическая доступность.

Исследованию и программной реализации именно такого класса систем, как слабосвязанные архивные системы хранения и доставки данных и посвящена данная диссертация.

Целью работы является разработка архитектуры и программная реализация географически распределенной системы архивного хранения и доставки файловых данных из гетерогенных узлов с узкими каналами связи.

Для достижения поставленной цели в работе решались следующие задачи:

1. Разработать архитектуру распределенной системы архивного хранения и доставки данных построенную на слабосвязанных гетерогенных узлах с ограниченными каналами связи.
2. Предложить и обосновать разделение программной и аппаратной составляющих системы.

3. Разработать метод адаптации системы в части управления, как размещением, так и доставкой контента.

Научная новизна работы.

Разработана архитектура распределенной системы архивного хранения и доставки неструктурированных данных с адаптивными свойствами, способная, в отличие от большинства существующих систем, работать в условиях ограниченных каналов связи.

Разработан метод обеспечения естественной масштабируемости распределенной системы хранения и доставки данных программно-аппаратного типа с использованием гетерогенных узлов (серверов) начального уровня, который позволяет осуществлять горизонтальное масштабирование системы без остановки или изменения конфигурации.

Разработан программный метод резервирования данных путем их динамического размещения по узлам системы, в которых не используются аппаратные средства резервирования данных.

На защиту выносятся следующие основные результаты работы:

- Реализация архитектуры распределенной системы архивного хранения данных, позволяющей избежать использования аппаратных средств резервирования и распределения данных в системе;
- Реализация программного метода резервирования данных путем их динамического размещения по узлам системы;
- Авторский метод восстановления данных с отказавших узлов системы, использующий принцип зеркального копирования узла без использования в своей работе сервера метаданных;

Использование результатов исследования. Результаты данной работы применяются в информационных системах компании ООО «ТВ КУПОЛ» (Телеканал 100ТВ), о чём имеется справка о внедрении.

Достоверность положений, выводов и практических рекомендаций подтверждается экспериментами на работающем прототипе системы, построенном на предложенной архитектуре и алгоритмах.

Методы исследования. Результаты диссертационной работы получены на основе использования методов объектно-ориентированного проектирования и программирования. Использованы модели и методы теории массового обслуживания, математической статистики и теории вероятностей.

Практическая значимость работы заключается в применимости полученных научно-технических результатов при проектировании, разработке и эксплуатации программно-аппаратных решений для Интернет и Интранет систем, реализованных в виде приложений картографии, астрономии, электронных библиотеках, средств автоматизации документирования и безбумажного документооборота.

Апробация работы. Основные результаты диссертации докладывались на следующих семинарах и конференциях:

- XXXVII научной и учебно-методической конференции Санкт-Петербургского государственного университета информационных технологий, механики и оптики (Санкт-Петербург, 29 января – 1 февраля 2008 г.), доклад: «Методы построения распределенных хранилищ данных»;
- V Всероссийской межвузовской конференции молодых ученых (Санкт-Петербург, 15-18 апреля 2008 г.), секция "Информационные технологии", доклад «Анализ факторов, влияющих на качественные и количественные показатели функционирования систем распределенного хранения данных»;
- Получен грант на исследование в конкурсе грантов для студентов и аспрантов вузов и академических институтов Санкт-Петербурга Российского государственного гидрометеорологического университета (Санкт-Петербург, 4 мая 2008г.);
- XXXVIII научной и учебно-методической конференции Санкт-Петербургского государственного университета информационных технологий, механики и оптики (Санкт-Петербург, 3-6 февраля 2009 г.), доклад: «Структура и программные средства распределенной системы хранения данных»;
- VI Всероссийской межвузовской конференции молодых ученых (Санкт-Петербург, 14-17 апреля 2009 г.), секция "Информационно-телекоммуникационные системы", доклад «Алгоритмы обработки информационных поток в распределенной системе хранения данных»;
- XXXIX научной и учебно-методической конференции Санкт-Петербургского государственного университета информационных технологий, механики и оптики (Санкт-Петербург, 2-5 февраля 2010 г.), доклад: «Организация сетевого взаимодействия узлов распределенной системы хранения данных»;
- VII Всероссийской межвузовской конференции молодых ученых (Санкт-Петербург, 20-23 апреля 2010 г.), секция "Информационные технологии", доклад «Принципы организации многоточечного доступа к распределенной системе хранения данных»;
- Многие из предложенных в данной работе решений уже нашли применение в прикладных информационных системах компании ООО «ТВ КУПОЛ» (Телеканал 100ТВ) и ЗАО “Концерн Струйные технологии”;

Публикации. Результаты диссертации опубликованы в научных работах, список которых приведён в конце автореферата. В том числе, имеются 5 публикации в научных периодических изданиях, из них 4 – в соавторстве. В совместных работах автору принадлежат основные результаты.

Структура и объём диссертации. Диссертация состоит из введения, трёх основных глав, заключения, списка литературы из 53 наименований. Общий объём диссертации составляет 85 страниц машинописного текста.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении представляется состояние вопроса и приводится краткое описание работы.

Первая глава носит аналитический характер и посвящена рассмотрению и оценке различных архитектурных решений.

В общем, все архитектуры распределенных систем хранения данных подразделяются на жесткосвязанные и слабосвязанные системы. Жесткосвязанность систем хранения обуславливается специфическими архитектурами, интерфейсами и техническими средствами связи среды хранения с серверами. Это системы LAN уровня с высокой пропускной способностью каналов.

Именно такие системы выпускают ведущие производители: EMC, предоставляющая полную линейку дорогостоящих жесткосвязанных систем хранения, IBM, HP и некоторые другие. Это наиболее распространенные системы, их выпускают промышленно и используют для задач хранения в высокопроизводительных кластерных системах, грид архитектурах, в бизнес приложениях, базах данных, где распределенность системы понимается, как взаимодействие по сети передачи данных в пределах одного ЦОД.

Особенность слабосвязанных систем состоит в том, что они предполагают географически распределенные узлы хранения, связанные низкоскоростными каналами. К слабосвязанным системам относят те, которые ориентированы на использование за границами ЦОДа, в частности в Веб сервисах. Фактически это не только системы хранения, но и доставки данных. Примерами слабосвязанных систем являются: EMC Atmos, Oracle Lustre, Apache Hadoop и др.

Далее проводится анализ перечисленных выше слабосвязанных систем. Рассматриваются основные сетевые протоколы работы распределенных систем хранения – CIFS, NFS и FTP. На основе анализа и сопоставления этих протоколов обосновывается целесообразность использования самого распространенного протокола для географически распределенных систем – HTTP. Хотя HTTP не является файл-ориентированным протоколом, нами обосновывается целесообразность его использования по соображениям распространенности и реализации в любых гетерогенных системах.

Рассматриваются основные типы архитектуры построения систем хранения: DAS, NAS, SAN, а также приводится их сравнительная характеристика (Таб. 1).

Обсуждается доступность систем, способы резервирования данных, таких как репликация и использование аппаратных средств архитектуры избыточных массивов жестких магнитных дисков (RAID). Приводятся оценки вероятности безвозвратной потери данных в результате выхода из строя одного или нескольких накопителей.

Табл. 1.Сравнительная характеристика типов подключения систем хранения

Характеристика	SAS (DAS)	NAS	SAN
Протоколы передачи данных	SCSI, SSA	CIFS, HTTP, NFS, FTP	FC, iSCSI
Скорость передачи	100-300 МБ/с	1 Гб/с на порт	4 Гб/с на порт
Сетевые протоколы	SCSI, нет сетевого интерфейса	TCP/IP через Ethernet, 1/10 Gigabit Ethernet	Fibre Channel, 1/10 Gigabit Ethernet
Масштабирование	Ограничено, один сервер	Среднее, необходим центр координации	Высокое, по SAN сети
Миграция данных	Снижает производительность сервера	Программное резервирование/ восстановление	Дублирование данных на аппаратном уровне
Экономическая эффективность	Высокая, (на старте)	Средняя	Высокая, (при развитии)

Согласно данным производителей, а также статистики Google (рис. 1) в системе из 1000 накопителей в среднем будет выходить из строя один накопитель в день. Вероятность выхода из строя второго накопителя в RAID массиве из 10-ти накопителей будет равна 1/100, т.е. в течение 100 дней может выйти из строя одновременно два накопителя в массиве, что приведет к полной потере данных.

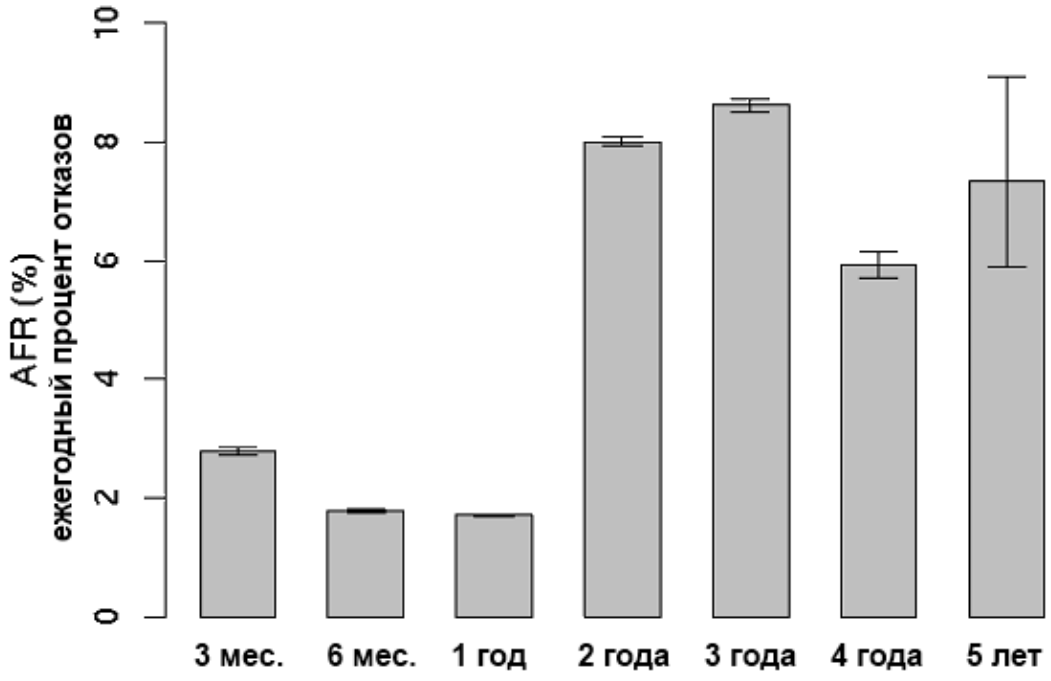

Рис. 1. Ежегодный процент отказов накопителей (статистика Google)

Проведенный анализ показывает, что актуально создание базовой архитектуры слабосвязанных систем хранения, для работы с неструктурированными данными в условиях ограниченной полосы пропускания. Необходимо создание алгоритмов доступа

к данным, учитывающих архитектурные особенности и способы резервирования данных.

В качестве единого протокола взаимодействия, как между внутренними узлами предлагаемой архитектуры, так и на клиентском уровне обоснованно использование протокола HTTP, который, не смотря на его архитектурные ограничения, обеспечивает важнейшее качество в части совместимости архитектуры хранения с клиентскими системами.

Во второй главе разрабатывается архитектура слабосвязанной распределенной системы хранения и доставки данных с учетом выработанных в первой главе требований, описываются ее основные аппаратные и программные компоненты, представляется алгоритм обработки информационных потоков.

На рис. 2 укрупненно представлена предлагаемая архитектура системы хранения и доставки данных LCDS2, названная по аналогии с терминологией, используемой в англоязычной литературе: Loosely Coupled Data Storage and Delivery System (LCDS2). В отличие известных и распространенных аппаратно-программных систем, предлагаемая архитектура в большей мере является программно-аппаратной.

Программная реализация должна обеспечивать не только плавное масштабирование, но и облегчать использование в гетерогенном окружении, то есть с разнородными программными и аппаратными компонентами. Один из факторов этого – использование для передачи данных общедоступного протокола HTTP. Основными элементами LCDS2 являются серверы управления доставкой, серверы метаданных и узлы хранения.

По обсуждавшейся выше классификации предлагаемая архитектура является гибридной и использует составляющие двух разновидностей распределенных систем хранения: параллельных и гибридных с API доступом.

Взаимодействие пользователя с системой происходит через сервер управления доставкой, производящий прием и обработку данных для хранения, а также непосредственно с узлами хранения, предоставляющими данные пользователю. Подчеркивается, что обмен с внешней средой происходит исключительно по протоколу HTTP.

Серверы метаданных содержат базы данных о местоположении и других параметрах файлов. Серверы управления доставкой контента выполняют работу по взаимодействию пользователя с системой хранения при загрузке данных в систему. Узлы хранения, содержащие в себе сами файлы, могут быть совершенно разными гетерогенными системами. Это значительное отличие предлагаемой архитектуры даже от близкой к ней архитектуры системы EMC Atmos.

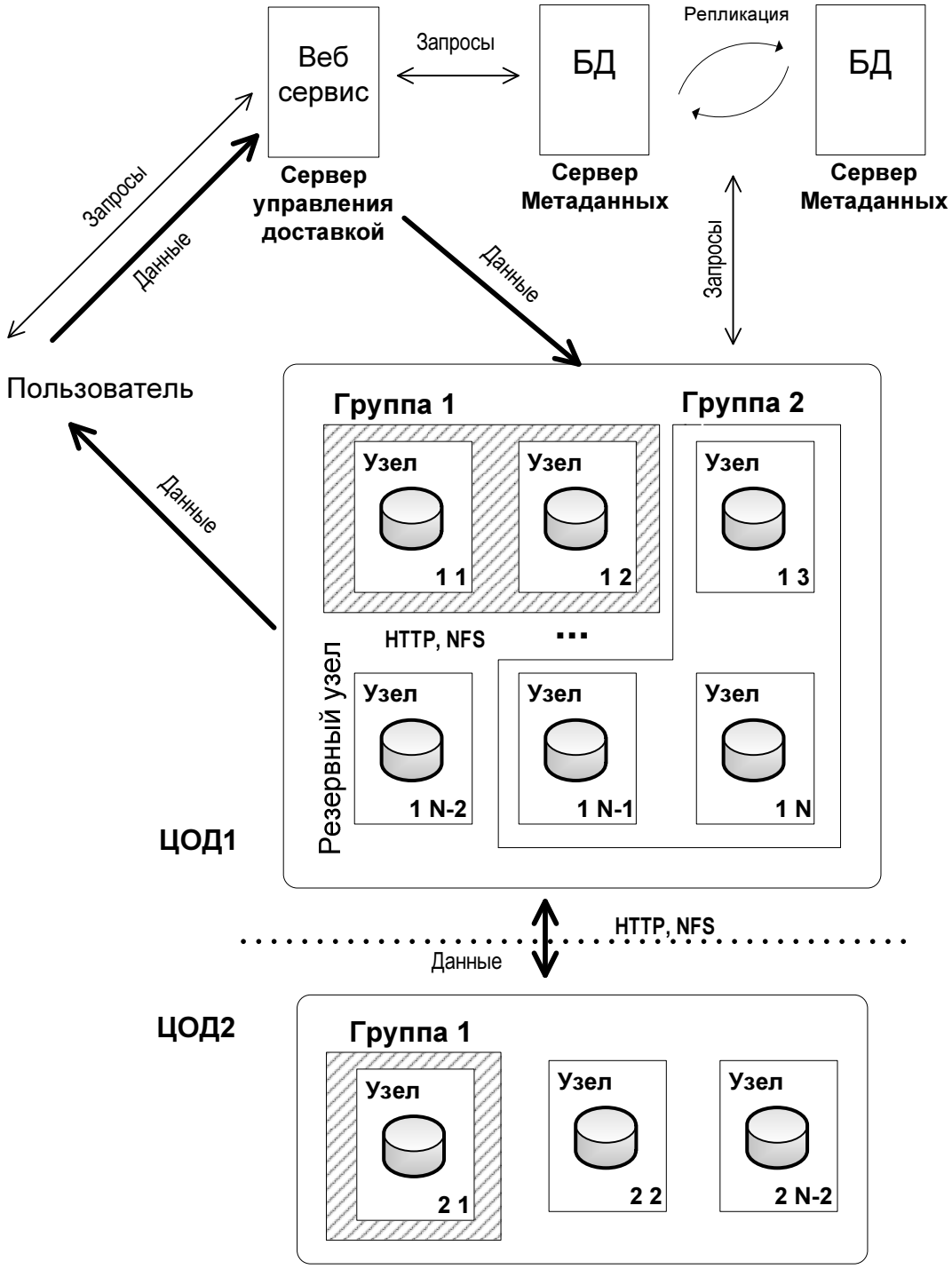

Рис. 2. Архитектура слабосвязанной системы хранения данных

Поскольку на узлах хранения не используется аппаратных средств резервирования вопрос дублирования и резервирования данных решается путем репликации файлов на нескольких узлах хранения в системе.

Узлы хранения, содержащие одинаковые файлы, объединяются в группы. При этом группа может содержать в себе узлы, находящиеся в разных ЦОД. Такой принцип резервирования данных будем называть сетевым зеркалированием узлов.

Комплекс – это объединение узлов, соединенных высокоскоростным магистральными каналами (не менее 1 Гбит/с), которые используется в связи с нахождением узлов в пределах одного ЦОД. Группы представляют собой объединение

узлов хранения, которые содержат идентичные данные, однако могут находиться в разных комплексах системы.

В предлагаемом решении метаданные связываются с единицей хранения – файлом. Отказ от блочного хранения файлов и распределения фрагментов файла по узлам системы обеспечивает прямое взаимодействие пользователя с узлами хранения систем. Это исключает дополнительную нагрузку по составлению файла из фрагментов, как это делается в таких системах, как Hadoop и GoogleFS.

Последовательность работы с данными, применяемые в представляемой распределенной системе хранения данных схематично изображены на рис. 3 в виде модулей. Для обеспечения совместимости работы в различных программных окружениях предлагается реализовать их в виде исполняемых веб сервером набором PHP процедур и функций.

.

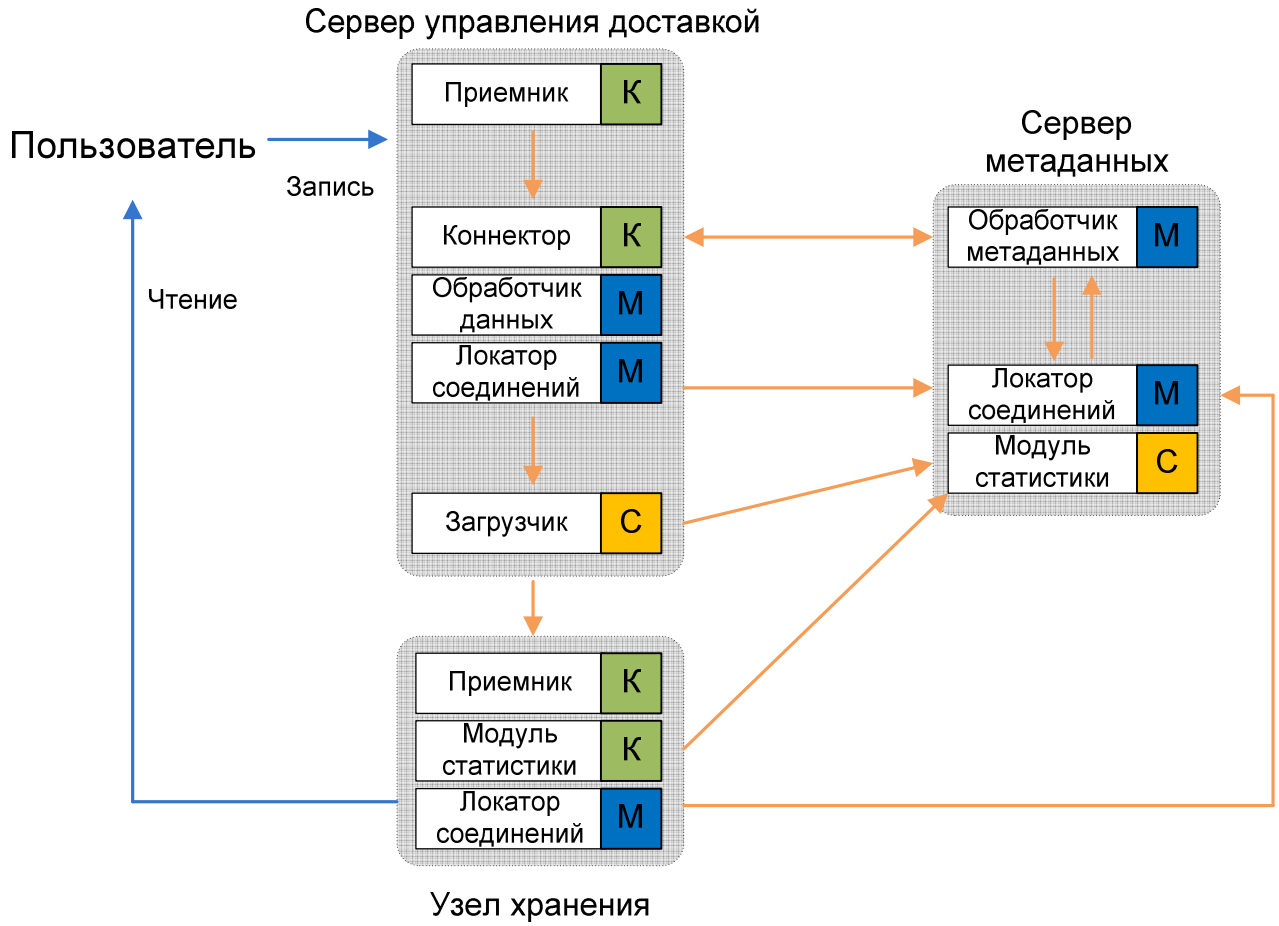

Рис. 3. Программные компоненты распределенной слабосвязанной системы хранения и доставки данных LCDS2

Модули подразделяются на следующие типы:

1. Клиентские модули (К) являются сборщиками и обработчиками информации, которая в дальнейшем направляется на сервер метаданных.
2. Серверные модули (С) для своей работы используют данные, полученные от клиентских модулей. В случае сервера управления загрузкой, модуль

“Загрузчик” выполняет реплицирование файла пользователя на группу узлов хранения. В случае сервера метаданных, “Модуль статистики” производит ранжирование узлов хранения согласно их загруженности и доступности в системе.

Обособленные модули (М) для своей работы не требуют клиентской части

Далее рассматривается вопрос репликации данных, а также обсуждаются разработанные алгоритмы и методы распределения данных в системе. Под репликацией в нашем контексте понимается процесс создания нескольких копий (реплик) одного и того же файла, каждая из которых хранится на отдельном узле хранения. Управление репликами централизовано и происходит через сервер метаданных, что гарантирует целость и непротиворечивость данных.

По данным Symantec репликация является наиболее эффективным решением для обеспечения доступности данных. Это объясняется тем, что при выходе из строя одного устройства хранения, нужные данные можно получить, обратившись к другому устройству, на котором имеется копия требуемого файла.

Таким образом, система оперирует несколькими репликами файлов, причем каждая копия находится на отдельном файловом сервере. Имеется несколько причин для предоставления этого сервиса, главными из которых являются:

- Увеличение доступности за счет наличия независимых копий каждого файла на разных файловых серверах. При отказе одного из них файл остается доступным.
- Распределение нагрузки между несколькими серверами.

При увеличении количества реплик файла, растет доступность и скорость доступа к ним. С другой стороны увеличение числа копий приводит к росту затрат памяти. Так возникает задача управления репликами, решение которой должно снижать необоснованные затраты.

Рекомендуемое число реплик N_{rec} предлагается рассчитывать, как сумму минимального числа реплик, принимаемого из соображений обеспечения принципа мажоритарности N_{min} и дополнительных реплик.

При расчете дополнительного числа реплик будем исходить из:

- популярности файла P_i, то есть количества обращений к файлу за определенный период времени;
- степени важности файла I_I;
- количества свободного места на носителях F_i;
- типа файла T_i.

Соответственно, каждому типу файла присвоен балл исходя из среднего размера для каждого типа, например: видео – 0, аудио – 1, картинки – 2, тестовые данные – 3. Важность каждого из параметров оценивается в баллах W_k (экспертная оценка) для учета

их значимости при вычислении дополнительного количества реплик. Затем находятся весовые коэффициенты исходя из бальных оценок.

Табл. 2. Баллы и весовые коэффициенты для параметров выбора дополнительного количества реплик

Параметр	Балл W_k	Весовой коэффициент w_k
Популярность файла	4	0.4
Степень важности	2	0.2
Тип файла	2	0.2
Свободное место на дисках	2	0.2

На основе вектора весовых коэффициентов и значений параметров, можно вычислить рекомендуемое число реплик для i-ого файла:

$$N_i = N_{\min} + N_{add} * (w_{pop} \frac{P_i}{P_{\max}} + w_{imp} \frac{I_i}{I_{\max}} + w_{free} \frac{F_i}{F_{\max}} + w_{type} \frac{T_i}{T_{\max}}),$$

где N_{add} – наибольшее возможное число дополнительных реплик, P_{max}, I_{max}, F_{max}, T_{max} – максимальные значения параметров, а w_{pop}, w_{imp}, w_{free}, w_{type} – их весовые коэффициенты.

Далее в главе предлагается алгоритм выбора узлов хранения для загрузки реплик файла, а также выбор предпочтительного узла для получения файла. Ключевыми в работе Интернет систем, кроме доступности, являются параметры масштабируемости и скорости. Для уменьшения задержек желательно стремиться к равномерной загрузки узлов.

Для распределения нагрузки между серверами обычно используются ресурсы системы DNS, которая работает по принципу циклической выборкой (round-robin). Она предусматривает возможность круговой передачи IP-адреса любого сервера, любому клиенту, что в конечном итоге равномерно распределяет нагрузку на серверы.

В предлагаемой архитектуре LCDS2 этот механизм нельзя считать достаточно эффективным, поскольку возможности аппаратных компонентов отдельных узлов системы неравнозначны. При этом запросы не могут быть распределены равномерно на все группы узлов системы, так как они не являются взаимозаменяемыми и хранят свой уникальный набор данных.

В процессе эксплуатации все узлы хранения периодически сообщают серверу метаданных о своих показателях работы. Модуль статистики этих узлов каждые три минуты передает следующие характеристики:

- количество обработанных запросов, $q_{зап}$
- среднее время обработки запроса, $t_{ср}$

- загрузку на центральный процессор с момента последнего опроса, $\alpha_{проц}$
- загрузку канала передачи данных, $\alpha_{кан}$
- емкость свободного места на дисках, $\alpha_{диск}$

Большая часть обработки статистических данных производиться модулем статистики на сервере метаданных. Обрабатываются показатели серверов за определенный период времени, например, последние 48 часов. Им присваиваются весовые коэффициенты, с учётом которых далее рассчитывается общий индекс. Весовые коэффициенты, применяемые в настоящей методике, подбирались экспертным методом и приведены в табл. 3.

Табл. 3. Весовые коэффициенты параметров быстродействия

Параметр	Весовой коэффициент
Загрузка процессора	0,15
Загрузка канала	0,75
Заполнение диска	0,1

Таким образом, общий рейтинг узла определяется следующим образом:

$$r_i = \alpha_{проц} * w_{проц} + \alpha_{кан} * w_{кан} + \alpha_{диск} * w_{диск}$$

По рейтингу составляется упорядоченный список узлов, готовых к приему и предоставлению данных пользователю. Используя эту таблицу, сервер метаданных сообщает модулю загрузки на сервере управления доставкой доступные для загрузки группы узлов хранения. Таким образом, чем меньше рейтинг, тем менее загруженным считается узел.

В третьей главе представляется программная реализация LCDS2, а также программный интерфейс DDPI (от англ. Data Delivery Program Interface), на базе которого узлы системы обмениваются данными.

Для построения прототипа системы была выбрана связка из нескольких программных компонент, обеспечивающих работу как POSIX, так и Windows совместимых систем в предлагаемой архитектуре. Прототип системы используется в области архивного хранения и предоставления видео данных и изображений. На серверах метаданных, серверах управления доставкой и большинства узлов хранения прототипа используется Linux Red Hat 5.0 и PostgreSQL 8.4. Все компоненты прототипа написаны на PHP и исполняются на веб сервере Apache 2.0. Для выполнения специфических внутрисистемных задач используются shell скрипты (восстановление данных узлов в группе по протоколу NFS).

Для обеспечения корректного взаимодействия узлов системы в гетерогенных программных окружениях предлагается использовать программный интерфейс DDPI на базе стандартных методов протокола HTTP.

С помощью этого интерфейса в системе производится:

- создание, удаление и изменение объектов (файлов)
- передача служебных данных (статистики работы) с узлов хранения на сервер метаданных
- управление узлами хранения сервером метаданных (например, инициирование процесса восстановления данных)

Для доступа к файлам и передачи служебной информации существуют две модели – модель пространства имен и модель объектов соответственно.

Модель пространства имен – наиболее распространенная в области разработки файловых систем. Она представляется как плоский адрес URI с указанием каталогов и подкаталогов, а также именем файла, например:

*node34.domain.name/upi/files/*folder1/subfolder2/0df3axvkmsdfks54.doc

В этом примере URI состоит из:

1. DNS имени одного из узлов хранения в группе, которое может меняться в зависимости от загруженности узлов. Это динамически меняющаяся переменная. Для каждой реплики файла, хранящейся в системе, существует еще несколько URI с другими DNS именами узлов, например node11.domain.name и node32.domain.name.
2. конструкции upi/files, которая указывает на то, что обращение к узлу хранения происходить к файловому пространству.
3. пути к файлу в файловой системе хранения данных на этом узле
4. уникального мета имени файла, которое отличается от его оригинального имени.

Модель объектов используется для передачи команд и служебных данных между узлами. Примерами URI может служить обращение к серверу метаданных одного из узлов хранения, чтобы загрузить данные статистики своей работы:

*mds01.domain.name/upi/cmd/*node/putinfo/
fsnode_fgklszersdpkqo3zcmbqpckvwskce_0401

В этом примере URI состоит из:

1. DNS имени сервера метаданных, на который происходит обращение
2. конструкции upi/cmd, которая указывает на то, что обращение будет происходить к некоторому объекту
3. конструкции node/putinfo, которая указывает на то, что вызывается команда загрузки статистических данных узла хранения
4. уникального идентификатора узла хранения

В данном случае при обращении по DDPI с использованием модели объектов важно использовать уникальные идентификаторы узлов, а не их DNS имена, поскольку в процессе динамических изменений конфигурации системы хранения, добавления и удаления узлов хранения DNS имена не могут однозначно определить уникальность узлов.

Во второй части главы обсуждается поведение системы в случае потери работоспособности ее узлов, например, выхода из строя дисковых накопителей или потери сетевого соединения.

Группа узлов может находиться в одном из состояний, которое определяется состоянием узлов внутри самой группы и правилами, регулирующими количество реплик:

- нормальное – в группе присутствует необходимое количество зеркалированных узлов;
- деградированное – в группе недостаточное количество зеркалированных узлов;
- избыточное – количество узлов в группе больше минимального, необходимого для обеспечения принципа мажоритарности.

Вопросам переведения узла из нормального состояния в избыточное занимается модуль статистики и обработчик метаданных на сервере метаданных. Такая операция может понадобиться в случае необходимости увеличения числа реплик файла для повышения его доступности и распределения нагрузки за счет увеличения числа узлов в группе.

Работа группы в деградированном состоянии – нештатная ситуация. Контролем процесса восстановления данных узлов занимается модуль статистики, обработчик метаданных, а также менеджер копий на сервере метаданных.

Процесс восстановления данных одного из узлов хранения показан на рис. 4. Группа X представлена тремя узлами: node1_2.domain.name, node1_8.domain.name и node1_6.domain.name. Эти узлы расположены в одном ЦОД, имеют высокоскоростное внутреннее соединение (1 Гбит/с) и находятся в состоянии нормального функционирования в группе, то есть на этих узлах возможны операции чтения, записи и удаления файлов.

В тот момент, когда один из узлов группы (node1_6) прекращает сообщать данные о параметрах своей работы на сервер метаданных (пункт 1 на рис. 4), обработчик метаданных исключает узел node1_6 из таблицы активных узлов (пункт 2). Группа X начинает функционировать в деградированном состоянии, а узлы хранения переводятся

в режим ограниченного функционирования, когда разрешено чтение и удаление, но запрещена запись на узлы.

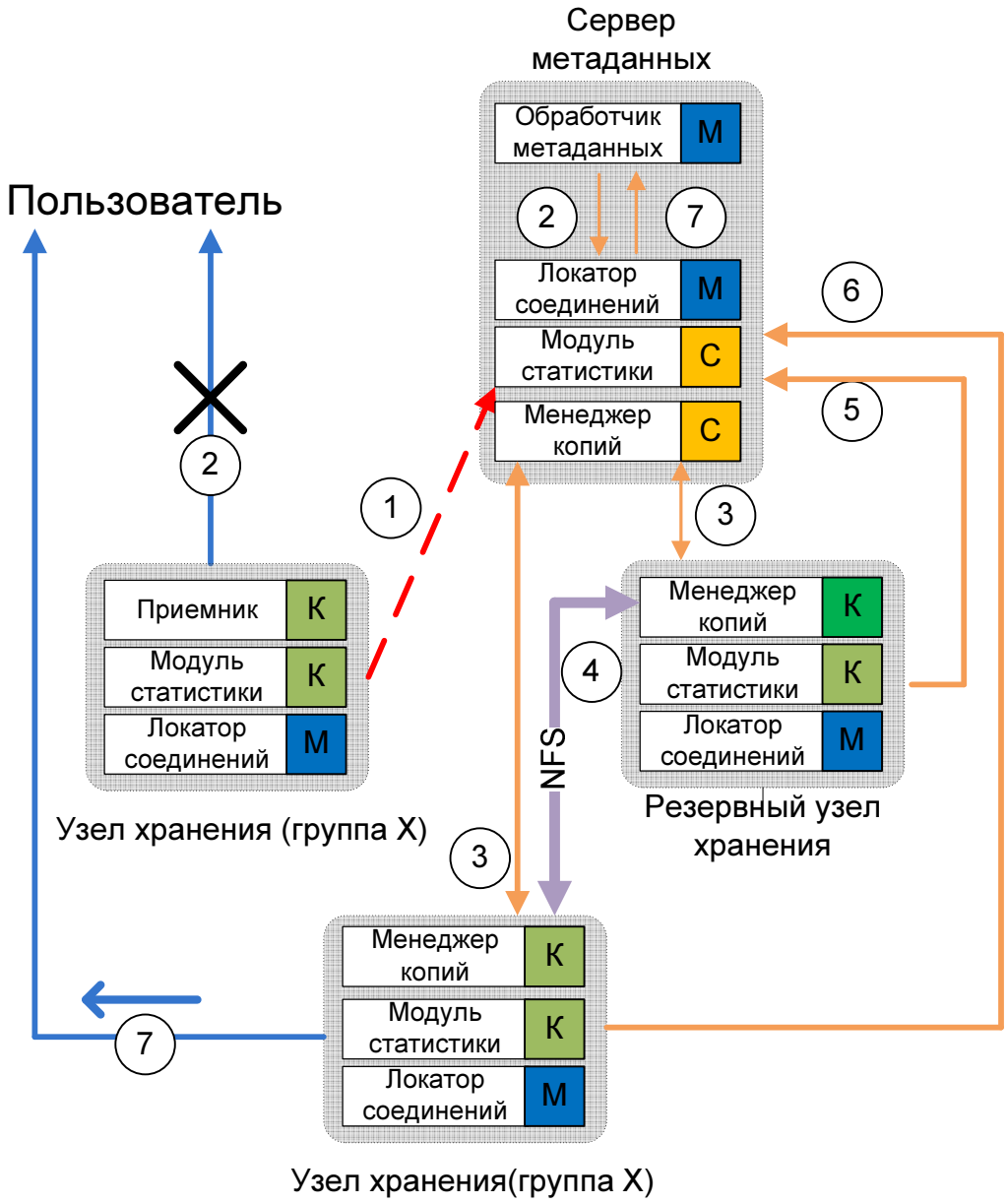

Рис. 4. Процесс работы системы по восстановлению данных узла хранения

В случае отсутствия данных от узла более чем 60 минут, из горячего резерва в группу вводится новый, запасной узел хранения node1_7 (пункт 3). Затем инициируется автономный процесс восстановления данных на узел node1_7 путем прямого копирования данных с работоспособного узла группы node1_8 по протоколу NFS (пункт 4).

За счет архитектурного решения восстановление узла происходит без использования сервера метаданных. Происходит лишь начальное извлечение данных об узлах, но не о файлах, располагающихся на этих узлах. Копирование происходит по внутреннему высокоскоростному соединению, поскольку узлы находятся в пределах одного ЦОД.

После завершения копирования данных новый узел node1_7 сообщает серверу метаданных о своем нормальном функционировании (пункт 5,6). Сервер метаданных модифицирует DNS запись резервного узла node1_7 на DNS запись потерянного узла node1_6.

Таким образом, весь процесс восстановления данных узлов происходит с минимальным изменением метаданных, то есть в этом процессе в минимальном объеме используются ресурсы сервера метаданных. Это в свою очередь положительно сказывается на производительности всей системы в целом.

Табл. 4. Время восстановления данных систем резервирования

Тип	**Объем дисков, МБ**	**Кол-во дисков, шт.**	**Уровень RAID**	**Пропускная способность NAS, МБ/с**	**Нормированное время, ч**
SAS	300	4	5	109	7,8
SATA	1500	4	5	94	10,5
Узел хранения	1500	1	нет	81	20,8
	1500	2	0	97	18,2

Время восстановления данных на узлах системы с помощью авторского алгоритма было проверено экспериментально. В табл. 4 полученное время сравнивается со временем восстановления данных в традиционных аппаратных RAID системах.

Результаты показывают, что восстановление данных по предлагаемому нами методу составляет 18 часов, что примерно в 2 раза больше, чем в традиционных системах.

Мы считаем это приемлемой величиной, поскольку, увеличивая время восстановления, мы исключаем возможность катастрофического отказа и полную потерю данных, что может произойти в первых двух случаях в традиционных RAID системах.

Возможность выхода из строя RAID системы крайне мала, однако при увеличении размеров системы до уровня глобальной распределенной системы, возможность полной потери данных становится более вероятной. В этом случае использование репликации данных, по предложенному алгоритму, безусловно, замедлит время восстановления, но в тоже время уменьшит вероятность потери данных.

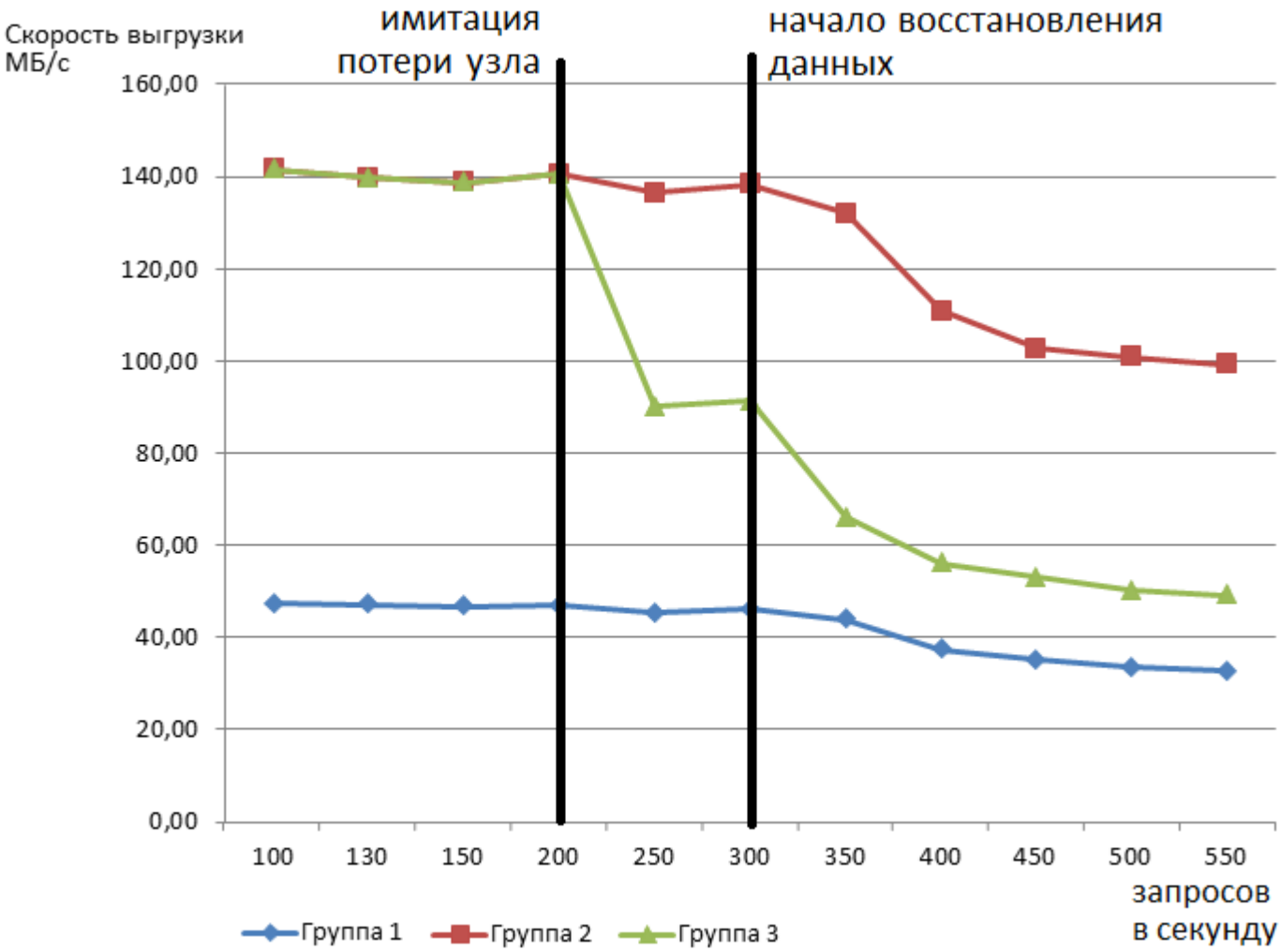

Рис. 5. Скорость выгрузки данных группы

На графике (рис. 5) вы можете наблюдать скорость выгрузки данных с узлов хранения в зависимости от количества запросов в секунду. Группа 1 состоит из одного узла, группа 2 и 3 из трех узлов с одинаковым, предварительно загруженным набором данных. Клиенты производят синхронные запросы на чтение с группы узлов, суммарно запрашивая 150 файлов по 100 Кб. В момент времени t1 происходит изъятие одного из узлов группы 3, тем самым имитируется его выход из строя. В результате этого уменьшается скорость выгрузки данных из группы.

В следующий момент времени t2 в группу 3 вводиться новый узел и начинается копирование на него данных, что еще больше уменьшает скорость выгрузки данных из группы. Однако при этом до сих пор соблюдается принцип мажоритарности, поскольку данные все еще зеркалируются.

Заключение содержит краткое описание полученных результатов.

В библиографическом разделе указаны все встреченные автором упоминания источников по тематике диссертации, как в печатном, так и в электронном виде.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

В настоящей диссертационной работе предложена и исследована архитектура и программная реализация географически распределенной системы архивного хранения и доставки файловых данных, которая состоит из гетерогенных улов с узкими каналами связи.

Основные результаты состоят в следующем:

1. Разработана архитектура распределенной системы архивного хранения данных удовлетворяющая сформулированным требованиям и реализующая практически неограниченное, плавное масштабирование и возможность использования гетерогенных, экономически доступных аппаратных компонент (серверов нижнего уровня или достаточно мощных рабочих станций).
2. На работающем прототипе подтверждена работоспособность предложенного разделения программных и аппаратных компонент.
3. Разработан программный метод адаптации системы в части управления, как размещением, так и доставкой контента.
4. Предложен авторский алгоритм управления количеством реплик, обеспечивающий достижение требуемой доступности хранимых данных.
5. Разработан программный метод восстановления данных с отказавших узлов системы с помощью сетевого реплицирования данных.
6. Реализована и эксплуатируется распределенная система хранения файлового контента в области архивного хранения и представления видеоданных.

СПИСОК ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Лукьянов Н.М., Дергачев А.М. Организация сетевого взаимодействия узлов распределенной системы хранения данных // Научно-технический вестник СПбГУ ИТМО №02(72). – 2011. – 137-141 с.
2. Лукьянов Н.М., Кириллов В.В. Анализ факторов, влияющих на качественные и количественные показатели функционирования систем распределенного хранения данных // Научно-технический вестник СПбГУ ИТМО №56. – 2008. – 9-17 с.
3. Лукьянов Н.М., Дергачев А.М. Ложные вычислительные системы для исследования и отвлечения атак // Научно-технический вестник СПбГУ ИТМО №45. – 2007. – 32-39 с.
4. Лукьянов Н.М. Принципы организации многоточечного доступа к распределенной системе хранения данных // Сборник тезисов докладов конференции молодых ученых, Выпуск 1.Труды молодых ученых СПбГУ ИТМО – 2010. – 17-18 с.
5. Лукьянов Н.М., Дергачев А.А. “Алгоритмы обработки информационных потоков в распределенной системе хранения данных” // Сборник тезисов докладов конференции молодых ученых, Выпуск 4.Труды молодых ученых СПбГУ ИТМО – 2009. – 217-223 с.