

На правах рукописи



Андреев Андрей Васильевич

**МЕТОДИКА ПЕРЕПРОЕКТИРОВАНИЯ СТРУКТУРЫ ХРАНЕНИЯ  
ДАнных СЛУЖБЫ КАТАЛОГОВ**

Специальность 05.13.11 –  
«Математическое и программное обеспечение вычислительных машин,  
комплексов и компьютерных сетей»

Автореферат  
диссертации на соискание учёной степени  
кандидата технических наук

Санкт-Петербург – 2020

Работа выполнена на кафедре вычислительных систем и сетей «Санкт-Петербургского государственного университета аэрокосмического приборостроения» (СПбГУАП).

**Научный руководитель:**

**Гордеев Александр Владимирович**, доктор технических наук, профессор, профессор кафедры вычислительных систем и сетей СПбГУАП.

**Официальные оппоненты:**

**Новиков Борис Асенович**, доктор физико-математических наук, профессор, профессор департамента информатики национального исследовательского университета «Высшая школа экономики» (г. Санкт-Петербург);

**Тюгашев Андрей Александрович**, доктор технических наук, доцент, заведующий кафедрой «Прикладная математика, информатика и информационные системы» федерального государственного бюджетного образовательного учреждения высшего образования «Самарский государственный университет путей сообщения» (г. Самара).

**Ведущая организация:**

федеральное государственное бюджетное образовательное учреждение высшего образования «Санкт-Петербургский государственный университет» (г. Санкт-Петербург).

Защита диссертации состоится «09» декабря 2020 г. в 15:30 часов на заседании совета по защите докторских и кандидатских диссертаций Д 212.238.01 Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» им. В.И. Ульянова (Ленина) по адресу: 197376, СанктПетербург, ул. Профессора Попова, д. 5.

С диссертацией можно ознакомиться в библиотеке Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» им. В.И. Ульянова (Ленина) и на сайте университета [www.etu.ru](http://www.etu.ru).

Автореферат разослан «08» октября 2020 г.

Учёный секретарь  
диссертационного совета  
кандидат технических наук, доцент



Пазников А.А.

## Общая характеристика работы

**Актуальность работы.** Служба каталогов - база данных, используемая как централизованное средство иерархического представления различных ресурсов и хранения информации об этих ресурсах. В качестве ресурсов выступают материальные ресурсы, персонал, информационно-вычислительные ресурсы, сетевые ресурсы и так далее.

Служба каталогов - программная система, позволяющая осуществлять создание, поиск и редактирование (добавление, удаление, исправление) информации об информационно-вычислительных ресурсах, хранящейся в виде каталога, т.е. списка записей с определенным набором атрибутов. Помимо перечисленного, в каталоге могут быть опубликованы различные информационно-вычислительные ресурсы для организации быстрого и простого их поиска. Кроме того, каталог часто используется для управления доступом к ресурсам и управления самой сетью - групповые политики, программное обеспечение и т.д.. Диссертационная работа посвящена усовершенствованию работы службы каталогов, строящейся на базе LDAP (Lightweight Directory Access Protocol). Самые распространённые службы каталогов - Active Directory, OpenLDAP, 389-DS и другие. Различные службы каталогов широко используются малыми, средними и крупными организациями и предприятиями, где служба каталогов выступает не только в роли централизованного инструмента хранения учётных записей и аутентификации пользователей, но и в роли механизма авторизованного доступа к ресурсам сети. Многие продукты, например почтовый сервер Zimbra или Microsoft Active Directory Domain Services, в основе своей архитектуры используют службу каталогов. Также существуют облачные реализации служб каталогов, например, JumpCloud предлагает облачный продукт Directory-as-a-Service, использующий OpenLDAP.

Для любого каталога одним из основных параметров эффективности является скорость поиска информации и её передачи для последующей обработки, будь то телефонный справочник или сервис доменных имён. Чем быстрее информация будет найдена, тем быстрее она будет передана для последующей обработки. Актуальность проблемы уменьшения времени отклика службы каталогов на запрос очевидна и обусловлена растущими объёмами информации, необходимыми для хранения, и требованиями к скорости дальнейшей обработки этой информации.

Высоконагруженная вычислительная сеть - это интенсивно работающая информационно-вычислительная сеть, как правило имеющая весьма развитую инфраструктуру. В её состав обычно входят информационные системы (Интернет сайты, системы информационного оповещения и связи), системы электронного документооборота, файловые хранилища, и так далее. Целью ИВС является централизация всех информационных процессов предприятия. Так, например, для доступа в сеть и работы с её ресурсами, как правило, используется единая система идентификации пользователей (служба каталогов): при входе в сеть пользователь представляется системе (проходит процедуру аутентификации) и может использовать любые её сервисы без повторной аутентификации. Такая система не только облегчает работу пользователя, но и позволяет более эффективно организовывать работу других сервисов ИВС, например отправку и получение пользователем электронных сообщений, хранение служебной информации пользователя и распределение прав доступа к ней, предоставление пользователю определённых полномочий и так далее. Ещё одним примером централизации является организация единого адресного пространства для всех служб ИВС. Процесс аутентификации всех пользователей такой сети может занимать длительное время, даже с применением всех оптимизационных методов, предлагаемых разработчиками реализация служб каталогов, что приводит к простоям и задержкам всех последующих сервисов сети. Таким образом можно судить, что служба каталогов выступает в роли программного инструмента организации взаимодействия программ и программных систем ИВС, где время отклика на запрос к службе каталогов имеет приоритетную значимость.

**Степень разработанности темы.** Начальным этапом исследования стали технические спецификации и стандарты RFC (Request for Comments) и ISO (International Organization for

Standardization), описывающие протоколы X.500 и LDAP, являющимися стандартами построения служб каталогов.

Значительный вклад в изучение вопросов, связанных с хранением и обработкой данных службой каталогов, внесли такие учёные, как Т. Хоус, М. Роуз, М. Смит.

Существующие подходы улучшения эффективности работы службы каталогов направлены на уменьшение времени отклика на запрос к данным путём использования различных способов кэширования дисковых подсистем, баз данных, используемых в качестве хранилища данных, а также создания индексов, по строго определённого минимальному набору атрибутов. Данные способы описаны в различных публикациях. Например, в работах Т. Хоуса, а далее в документации таких компаний, как Red Hat, Microsoft и IBM. Полезность данных способов очевидна до определённого количества записей каталога, так как с увеличением общего количества записей увеличивается и количество индексов, требуемых для обработки и хранения, а также размера кэша и периодичность его обновления. Современные исследования на тему служб каталогов рассматривают вопросы интеграции различных служб каталогов в текущую инфраструктуру ИВС.

В диссертационной работе рассматриваются вопросы организации хранения данных службы каталогов для того, чтобы ускорить обработку запросов на поиск искомых данных и повысить эффективность работы всей службы каталогов.

**Цель** диссертационной работы - разработать методику такого перестроения структуры хранения данных в службе каталогов, которое приводит к уменьшению времени отклика сервера службы каталогов на запросы клиентов к данным.

**Объектом исследования** в диссертационной работе является служба каталога, структуры данных в ней и время поиска данных.

**Предметом исследования** являются структуры хранения данных в службе каталогов и методика перепроектирования структуры хранения данных.

Для достижения поставленной цели были сформулированы следующие **задачи**:

1. Определить рабочее окружение существующих служб каталогов, основные параметры работы служб каталогов и способы хранения данных, сформулировать условия работы служб каталогов.

2. Предложить структуру хранения данных, позволяющую уменьшить время отклика на запрос за счёт изменения структуры хранения данных с целью уменьшения количества записей для поиска.

3. Разработать метод разбиения множества параметров каталога, хранящихся в службе каталогов, на подмножества с учётом приоритета их использования, что в дальнейшем позволит сократить перебор данных и ускорить поиск.

4. Разработать метод перехода от иерархической структуры хранения данных (без учёта приоритетов использования параметров) к предлагаемой структуре, которая учитывает приоритеты использования параметров.

5. Разработать и протестировать методику перепроектирования структуры хранения данных службы каталогов для уменьшения времени отклика сервера службы каталогов на запросы клиентов к данным.

**Основные положения, выносимые на защиту:**

1. Методика перепроектирования структуры хранения данных службы каталогов.

2. Метод разбиения множества параметров каталога, хранящихся в службе каталогов, на подмножества с учётом приоритета их использования с целью уменьшения времени отклика на запрос.

3. Метод перехода от иерархической структуры хранения данных без учёта приоритетов использования параметров к структуре с учётом приоритетов использования параметров на основе анализа основных параметров работы службы каталогов и способов хранения данных.

**Методология и методы исследования.** Для решения поставленных задач были использованы аппарат теории графов, теории множеств, методы формализации и идеализации.

**Научная новизна** работы состоит в:

1. Предложена методика перепроектирования структуры хранения данных службы каталогов, в которой определена новая сетевая структура хранения данных службы каталогов с учетом приоритетов данных вместо стандартной иерархической без учета приоритетов данных.
2. Предложен метод разбиения множества параметров каталога, хранящихся в службе каталогов, на подмножества с учетом приоритета их использования, в котором впервые учитываются условия группирования записей каталога на основании принадлежности и приоритетности к сервисам информационной вычислительной сети.
3. Предложен новый метод перестроения структуры хранения данных службы каталогов с учетом приоритетов данных и новых условий группирования записей.

**Теоретическая значимость** результатов исследования заключается в:

- проведении анализа основных факторов, влияющих на временную характеристику служб каталогов;
- определении условий эффективности работы служб каталогов;
- разработке методика перепроектирования структуры хранения данных служб каталогов для уменьшения времени отклика сервера службы каталогов на запрос клиентов к данным;
- разработке метода разбиения множества параметров каталога, хранящихся в службе каталогов, на подмножества с учётом приоритета их использования;
- разработке метода перехода от иерархической структуры хранения данных без учёта приоритетов использования параметров к структуре с учётом приоритетов использования параметров на основе анализа основных параметров работы служб каталогов и способов хранения данных.

**Практическая значимость** диссертационной работы определяется в применении основных положений, позволяющих, помимо стандартных методов оптимизации работы служб каталогов, повысить эффективность работы служб каталогов с помощью изменения структуры хранения данных, проектировать и конфигурировать службы каталогов с учётом приоритетов обрабатываемых данных, а также производить построение адаптивных служб каталогов в зависимости от требований к времени доступа к данным.

**Достоверность результатов.** Достоверность изложенных в работе результатов обеспечивается корректным использованием математического аппарата теории множеств, теории графов. Для проверки полученных результатов была смоделирована и протестирована служба каталогов по сформулированным в работе условиям.

**Апробация работы.** Основные научные положения и результаты диссертационной работы докладывались, обсуждались и получили одобрение на научно-методических семинарах кафедры «Вычислительные системы и сети» ГУАП и докладывались на 72-й научной сессии ГУАП (апрель 2019, г. Санкт-Петербург).

**Внедрение** результатов диссертационной работы. Результаты работы применялись при проектировании и построении тестовых окружений сервисов служб каталогов выпусков Oracle Linux ООО «Оракл Девелопмент СПб» и при проектировании сетевых решений на базе LW\DSA, разработанных ЗАО «ВизардСофт.Ру».

На специальное программное обеспечение для формирования сетевой структуры хранения данных службы каталогов получено свидетельство о государственной регистрации программ для ЭВМ №2019614753.

**Личный вклад автора** диссертационной работы заключается в:

- разработке методика перепроектирования структуры хранения данных службы каталогов для уменьшения времени отклика при запросе к данным;
- определению рабочего окружения существующих служб каталогов, основных параметров работы служб каталогов и способов хранения данных, формулированию условий работы служб каталогов;

- проведении анализа временной характеристики службы каталогов на основе встроенных инструментов и методов поиска, а также анализа основных факторов, влияющих на временные характеристики службы каталогов;
- предложении сетевой структуры хранения данных, позволяющей уменьшить время отклика на запрос за счёт изменения структуры хранения данных с целью уменьшения количества записей для поиска.
- предложении и разработке метода разбиения множества параметров каталога, хранящихся в службе каталогов, на подмножества с учётом приоритета их использования с целью уменьшения времени отклика на запрос.
- предложении и разработке метода перехода от иерархической структуры хранения данных (без учёта приоритетов использования параметров) к структуре с учётом приоритетов использования параметров;
- разработке программного обеспечения для формирования сетевой структуры службы каталогов по разработанным методам и методике.

**Публикации по теме диссертации.** Материалы, отражающие основное содержание и результаты диссертационной работы, опубликованы в 7 печатных работах. Из них 3 работы опубликованы в рецензируемых научных журналах, внесённых в перечень ВАК, и 1 работа опубликована в издании, индексируемом в Scopus.

**Объём и структура работы.** Диссертация состоит из введения, четырёх разделов, заключения. Полный объём диссертации составляет 138 страниц с 24 рисунками и 10 таблицами и приложений, включающих акты внедрения, свидетельство о государственной регистрации программы для ЭВМ, примеры используемого исходного кода. Список используемых источников содержит 60 наименований.

## **Содержание работы**

**Во введении** обосновывается актуальность исследований, проводимых в рамках данной диссертационной работы, приводится обзор научной литературы по изучаемой проблеме, формулируется цель, ставятся задачи работы, сформулированы научная новизна, теоретическая и практическая значимости представляемой работы.

**Первый раздел** посвящён LDAP, как сетевому протоколу доступа к службам каталогов и стандарту построения архитектуры каталога. Определены общие понятия и различия служб каталогов и служб каталогов LDAP, где основное отличительное свойство - архитектура построения каталога, а именно иерархическая структура, где структура каталога организована по физическому принципу расположения объекта, например, записи сотрудников организации структурированы по принадлежности к отделу организации. В работе рассматриваются службы каталогов LDAP, а под термином служба каталогов понимается служба каталогов LDAP.

Рассмотрена открытая реализация службы каталогов - OpenLDAP. OpenLDAP используется как основная служба каталогов для рассмотрения в данной работе в силу своей открытости и строгого следования стандарту LDAP.

Дано описание рабочего окружения служб каталогов. Основным составляющим рабочего окружения являются сервисы, использующие службу каталогов, так как они формируют тип записей и содержимое записей, а также как часто и каким способом данные, хранящиеся в службе каталогов, будут использованы. Также отмечена роль человека, как конечного и основного пользователя услуг службы каталогов, работающего напрямую со службой каталогов или через соответствующие сервисы сети, работающие со службой каталогов.

Различные сервисы сети используют службу каталогов для аутентификации, авторизации или хранения данных. Для доступа в сеть и работы с сетевыми ресурсами служба каталогов используется как единая система идентификации пользователей: при входе в сеть пользователь представляется системе, проходит процедуры аутентификации и авторизации и может использовать любые её сервисы без повторной аутентификации. Такая система не только облегчает работу пользователя, но и позволяет более эффективно организовывать работу других

сервисов сети, например отправку пользователем электронных сообщений, хранение служебной информации пользователя и распределение прав доступа к ней, предоставление пользователю определённых полномочий. Рабочее окружение службы каталогов изображено на рисунке 1.



Рисунок 1 - Рабочее окружение службы каталогов.

Процесс аутентификации и авторизации всех пользователей сети или поиска информации об объекте каталога может занимать длительное время, даже с применением всех оптимизационных шагов, что приводит к простоям и задержкам всех последующих сервисов сети, использующих службу каталогов. Скорость получения данных от службы каталогов влияет на время последующей обработки данных следующим сервисом и вывода результатов пользователю. Таким образом, важной характеристикой эффективности работы службы каталогов является временная характеристика, а именно, время отклика на запрос (system response time).

Основной вывод раздела - время отклика на запрос - одна из важнейших характеристик службы каталогов и основная характеристика, рассматриваемая в данной работе, а задача уменьшения времени отклика на запрос к разнообразным данным представляет значительный интерес так, как данное улучшение эффективности работы службы каталогов позволяет ускорить последующую обработку запрашиваемых данных и вывода результата запроса пользователю.

**Второй раздел** посвящён анализу временной характеристики службы каталогов, как основной. Определены основные факторы, влияющие на производительность службы каталогов:

- аппаратные характеристики сервера, обслуживающего службу каталогов;
- пропускная способность сети;
- дополнительные роли сервера, обслуживающего службу каталогов;
- сервисы, использующие службу каталогов;
- конфигурация службы каталогов.

Сформулирован набор рассматриваемых условий работы службы каталогов, используемый для анализа временной характеристики:

- на сервере выполняется только служба каталогов;
- служба каталогов использует для хранения записей модифицированную нереляционную базу данных Berkeley DB - HDB;
- опции кэширования отключены для службы каталогов, базы данных HDB;
- индексирование не используется;
- все записи расположены в одной ветви.

В данном разделе описана используемая для исследования конфигурация службы каталогов с учётом рабочего окружения определённого в первом разделе.

Проведён анализ стандартных средств и алгоритмов поиска записей каталога. Анализ указывает на то, что при поиске записей каталога используется обход дерева в прямом порядке, что позволило сформулировать способ расчёта времени отклика на запрос, как:

$$t = \frac{E * N * 1024}{\left( (iops * N_h * \%read) + ((iops * N_h * \%write) * RAIDPenalty) \right) * KB} + \left( \frac{utime_0 - utime_1 + stime_0 - stime_1}{HZ} \right), \quad (1)$$

где  $E$  - размер записи в мегабайтах,  $N$  - количество записей в заданной ветви поиска каталога,  $iops$  - количество операций ввода/вывода дисковой подсистемы в секунду заданных производителем,  $N_h$  - количество носителей,  $\%read$  - процент операций чтения,  $\%write$  - процент операций записи,  $RAIDPenalty$  - значение штрафа для RAID,  $KB$  - размер блока в Berkeley DB в килобайтах,  $utime_0$  - изначальное время выполнения процесса в режиме задачи в секундах,  $utime_1$  - текущее время выполнения процесса в режиме задачи в секундах,  $stime_0$  - изначальное время выполнения процесса в режиме ядра в секундах,  $stime_1$  - текущее время выполнения процесса в режиме ядра в секундах,  $HZ$  - количество тиков за секунду.

Данный способ расчёта показал, что, помимо аппаратных характеристик сервера службы каталогов, время отклика на запрос зависит от размера записей и их количества в заданной ветви поиска. Дальнейшее рассмотрение влияния таких не аппаратных факторов, как размер записи, количество записей и индексирование, на время отклика на запрос показало, что время отклика на запрос зависит от общего количества записей в заданной ветви поиска каталога и размера записей, а индексирование - хорошее решение улучшения работы службы каталогов, но данный подход имеет ряд ограничений, зависящих от того, как используется каталог и как используются данные, хранящиеся в данном каталоге.

Одним из таких ограничений, например, являются тип фильтра и параметры его использования. Тип фильтра влияет на эффективность индексирования, например, индексы присутствия и подстроки. Одним из решений улучшения работы такого каталога может быть использование изменения структуры каталога по определённому заранее критерию, где ветвь поиска будет заменять основной фильтр поиска и уменьшать общее количество записей для выборки. Данное изменение можно реализовать с помощью записей-ссылок, не изменяя основную структуру каталога, используя сетевую структуру хранения данных.

В данном разделе проведен анализ современных исследований приемлемого значения времени отклика на запрос для человеко-машинного взаимодействия, на основании которого сформулированы условия эффективности работы службы каталогов, как:

$$\sqrt{\frac{1}{n-1} \sum_{i=1}^n \left( t_i - \frac{1}{n} \sum_{i=1}^n t_i \right)^2} \leq Tv, \quad (3)$$

$$\frac{1}{n} \sum_{i=1}^n t_i \leq T, \quad (4)$$

где  $T$  - ожидаемое время поиска записи в условиях сложности задачи в секундах,  $t_i$  - фактическое время поиска записи каталога для  $n$  повторений в секундах,  $Tv$  - допустимая величина вариации результатов времени отклика на запрос в секундах.

При невыполнении условий (3) и (4) следует рассматривать применение возможных способов улучшения работы службы каталогов, таких как:

- улучшение аппаратных характеристик,



- индексирование часто используемых атрибутов,
- кэширование данных,
- изменение структуры хранения данных.

Основной вывод второго раздела это - изменение структуры каталога по определенным условиям и критериям приоритетности условий следует рассматривать как дополнительный способ улучшения эффективности работы службы каталогов, позволяющий существенно уменьшить общее количество записей для поиска, а соответственно и время отклика на запрос, при определённых выше условиях. Поиск записи, соответствующей определенному условию и критерию приоритетности условия, осуществляется в дополнительно созданной ветви каталога, которая содержит только записи соответствующие выбранному условию. За счет этого, поиск записи осуществляется обходом в прямом порядке дополнительной ветви каталога, соответствующей определенному условию, где изначальное количество записей для выборки существенно меньше, чем изначальное количество записей в исходной ветви каталога. Данное изменение задаёт новый способ построения структуры хранения данных службы каталогов, которая организована по условиям приоритетности данных, задаваемыми приложениями, использующими службу каталогов, а не физическому или организационному признакам принадлежности записей, как это задается стандартом LDAP.

**Третий раздел** посвящен рассмотрению структурных вопросов хранения данных службы каталогов, моделей данных, описывающих структуры, и методике перепроектирования исходной структуры хранения данных службы каталогов, а также условиям применения методики. Исходя из предположения, что время отклика на запрос зависит от общего количества записей в заданной ветви поиска, можно сделать следующее предположение, что изменяя структуру каталога путём создания дополнительных вершин для обхода части дерева, можно добиться уменьшения времени отклика на запрос. Однако, одни и те же записи могут удовлетворять условиям принадлежности сразу к нескольким сервисам, что противоречит иерархической модели данных, описывающей иерархическую структуру хранения данных, так как в данном случае вершины записей каталога должны иметь несколько вершин-родителей. Это можно рассматривать как основной недостаток иерархической модели данных в данном контексте. Иерархическая структура каталога изображена на рисунке 1.

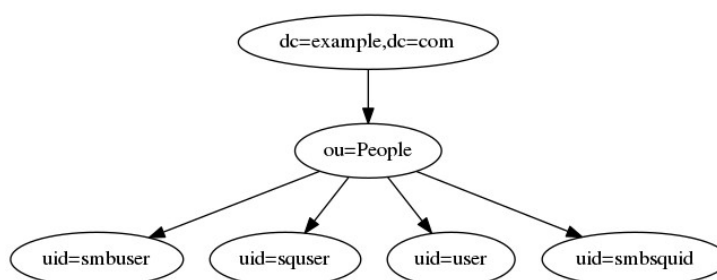


Рисунок 2 - Иерархическая структура хранения данных службы каталогов

Структура каталога изображенная на рисунке 2 является стандартной иерархической структурой, а также связным ориентированным ациклическим графом  $G=(V, E)$  древовидной структуры, объединяющий сегменты, где  $V(G)$  - множество вершин записей каталога, а  $E(G)$  - множество ориентированных ребер, дуг.

$$\begin{aligned}
 G &= (V, E) \\
 V(G) &= \{vd, vp, v_0, \dots, v_i\} \\
 E(G) &= \{(vd, vp), (vp, v_0), \dots, (vp, v_n)\}
 \end{aligned}
 \tag{5}$$

где  $vd$  - корень дерева,  $vp$  - вершина-родитель для группы записей ( $ou=People$ ),  $v_i$  - вершина-потомок для записи в каталоге, например  $uid=smbuser$ ,  $i$  - количество записей,  $i=1, \dots, |V|$ .

Предлагается использовать сетевую структуру хранения данных, вместо стандартной иерархической, где вершины записей каталога могут иметь несколько вершин-родителей.

Сетевая модель данных позволяет масштабировать дерево каталога в ширь, позволяя спроектировать сетевую структуру хранения данных службы каталогов. Разница между иерархической моделью данных и сетевой состоит в том, что в иерархических структурах запись-потомок должна иметь в точности одного предка, а в сетевой структуре у потомка может иметься любое число предков.

Для текущей редакции технической спецификации RFC 4512 от июня 2006 года дуга между вершинами графа, формирующего иерархическую структуру каталога, является обозначением некой физической взаимосвязи объектов, отражённых в каталоге. Для такой структуры характерно такое построение дерева каталога, где записи сотрудников организации или предприятия хоть и группируются по принадлежности к подразделению, но хранятся в рамках одной общей ветви каталога обозначающей запись сотрудников. При таком построение структуры каталога службы сети, использующие службу каталогов, настраиваются на обработку одной общей ветви каталога с глубиной поиска SUP, то есть с поиском на всех подуровнях начиная с заданной начальной вершины. В рамках больших организаций или каталогов с большим количеством хранящихся данных, такой способ построения каталога влияет на время отклика при запросе к данным, так как изначальная выборка включает в себя избыточное множество объектов каталога.

Вместо обозначения связи между объектами используемого в RFC 4512, под связью родитель-потомок между записями каталога предлагается использовать отношение выполнения некоторого условия принадлежности с:

- приложением, использующим службу каталогов;
- фильтром поиска;
- другими условиями.

Техническая спецификация RFC 4511 не запрещает создавать и работать с одним объектом каталога в разных ветвях каталога через записи-ссылки, а клиентам - работать с записями-ссылками. Таким образом, группируя записи с учетом новых предлагаемых условий, можно сформировать более упорядоченную структуру хранения данных службы каталогов на основе сетевой модели данных. Использование записей-ссылок добавляет возможность создания адаптивных каталогов, меняющим структуру согласно требованиям к каталогу в определенные промежутки времени.

Таким образом, иерархическую структуры хранения данных изображённую на рисунке 2, можно представить в виде сетевой структуры хранения данных изображённой на рисунке 3, где записи каталога перераспределены по сервисам, использующих их.

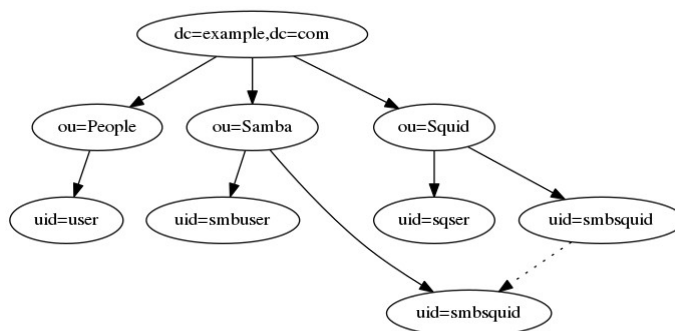


Рисунок 3 - Сетевая структура хранения данных службы каталогов

Рассмотрим структуру изображённую на рисунке 3, данная структура - сетевая структура хранения данных службы каталогов с корневым узлом «dc=example, dc=com», в которой записи сгруппированы согласно сервисам использующем их, где записи файлового сервера содержатся в ветви файлового сервера, а записи прокси-сервера в ветви прокси-

сервера. Ветвь прокси-сервера содержит запись-ссылку «uid=smb squid» на запись в ветви файлового сервера для записи, соответствующей обоим сервисам. При изменении данных записи «uid=smb squid», соответствующей обоим сервисам, необходимые изменения вносятся только для записи в ветви файлового сервера «ou=Samba», так как запись-ссылка в ветви «ou=Squid» содержит только указатель на запись в ветви сервиса с большим приоритетом.

Предлагаемая структура каталога изображенная на рисунке 3 является сетевой структурой, а также ориентированной сетью, которая задается графом  $G' = (V', E')$ , где  $V'(G')$  - множество вершин записей каталога, а  $E'(G')$  - множество ориентированных ребер, дуг.

$$G' = (V', E') \quad (6)$$

$$V'(G') = \{vd, vp, v_0, \dots, v_i, vo_0, \dots, vo_k, vs_{00}, \dots, vs_{0j}, \dots, vs_{k0}, \dots, vs_{kj}, vl_0, \dots, vl_m\}$$

$$E'(G') = \{(vd, vp), (vd, vo_0), \dots, (vd, vo_k), (vp, v_0), \dots, (vp, v_i), (vo_0, vs_{00}), \dots, (vo_0, vs_{0j}), \dots, (vo_k, vs_{k0}), \dots, (vo_k, vs_{kj})\} \cup \{(v, u) | \forall v, u: v \in V_{S_k} \wedge u \in V_{S_{k+1}} \wedge v = u\}$$

где  $vd$  - корень дерева иерархической структуры,  $vp$  - вершина-родитель для группы записей (ou=People),  $v_i$  - вершина-потомок для записи каталога ветви ou=people,  $vl_m$  - вершины-ссылки, они же alias,  $vo_k$  — вершины сервисов,  $vs_{jk}$  - вершина-потомок ветви сервиса с индексом группирования  $k$ ,  $Vp$  - множество вершин в исходной ветви, ou=People,  $V_{S_k}$  - множество вершин в ветви сервиса с индексом группирования  $k$ ,  $Vl$  - множество вершин-ссылок,  $k = 1, \dots, s$ ;  $s$  - количество сервисов,  $i = 1, \dots, |Vp|$ ,  $m = 1, \dots, |Vl|$ ,  $j = 1, \dots, |V_{S_k}|$ .

Для проектирования и построения предлагаемой сетевой структуры хранения данных службы каталогов предлагается использовать соответствующий способ теоретического исследования и проектирования сетевой структуры каталога, исследуемой службы каталогов, и практического осуществления перехода от иерархической структуры к сетевой - методика перепроектирования структуры хранения данных службы каталогов, которая сформулирована следующей последовательностью из 9 шагов.

1. Определить структуру и типы записей исследуемого каталога.

Следует определить расположение записей в ветвях каталога. Различные реализации служб каталогов имеет различные преконфигурированные ветви каталога для универсальных типов записей, например записи пользователей в одних реализациях могут храниться в ветви cn=Users, dc=example, dc=com, а в других в ou=People, dc=example, dc=com, где dc=example, dc=com - корневая вершина структуры каталога.

Первоначальные условия группирования записей могут быть определены уже на первом шаге методики, поэтому после определения исходной ветви каталога следует определить отображение объекта каталога в реальном мире, например, записи в исходной ветви каталога могут быть отображением сотрудников организации или рабочими станциями.

2. Обозначить набор сервисов сети, работающих со службой каталогов.

Определение набора сервисов, работающих со службой каталогов, позволяет далее определить условия группирования записей. Сервисы сети могут быть определены по ролями серверов сети или различными приложениям, использующих службу каталогов. Например файловый сервис соответствует серверу с ролью файлового сервера.

Среди наиболее часто встречающихся ролей серверов выделяются: почтовый сервер, прокси-сервер, файловый сервер, сервер аутентификации.

Данный шаг позволяет точно определить атрибуты записей, соответствующие выбранному набору сервисов.

3. Измерить частоту использования службы каталогов сервисами.

Если сервис из определенного набора сервисов на предыдущем шаге редко обращается к службе каталогов за данными, его не следует выбирать как условие группирования записей. Частота использования сервисом службы каталогов может быть определена с использованием журналов работы службы каталогов.

4. Зафиксировать типы запросов сервисов к службе каталогов.

Предпочтение при выборе сервиса, использующего службу каталогов, в качестве условия группирования записей, следует отдавать сервису с преобладающими запросами чтения данных над запросами добавления или изменения. Типы запросов могут быть определены с использованием журналов работы службы каталогов в режиме отладки (debug).

5. Оценить эффективность работы службы каталогов с сервисами.

Для оценки эффективности работы службы каталогов предлагается использовать условия сформулированные во втором разделе работы на основании проведенного исследования приемлемого значения времени отклика на запрос для человеко-машинного взаимодействия.

Если условия (3) и (4) выполняются, то служба каталогов работает эффективно с заданным сервисом, а дальнейшее использование сервиса в качестве условия группирования нецелесообразно.

6. Установить условия группирования записей.

На основании результатов полученных при выполнении шагов 1-5 данной методики следует определить окончательный набор условий группирования записей для выбранной ветви каталога.

7. Определить атрибуты записей, соответствующие выбранному набору условий.

Сервисы, работающие со службами каталогов, используют определённый набор атрибутов, идентифицирующий запись или набор записей. Некоторые сервисы используют атрибуты, предоставляемые разработчиками реализацией службы каталогов, а некоторые, на пример, samba, используют специфичные набор атрибутов. Перед определением множеств записей каталога для переноса с учётом приоритета их использования, следует определить множества необходимых атрибутов записей  $B_i$  для каждого сервиса. Список используемых атрибутов формируется по открытым данным используемых схем и типам запросов сервиса к каталогу. Например, служба samba, выступающая в роли сервиса файлового сервера, использует файл-схему samba, которая содержит перечень всех возможных (MAY) и необходимых (MUST) атрибутов, требующихся для работы данного сервиса. Используя только необходимые атрибуты классов, определяются множества соответствующих атрибутов  $B_i$  для каждого сервиса.

Множества необходимых сервисам атрибутов  $B_i$  определяются анализом журналов службы каталогов в режиме отладки или документации разработчика сервиса (программного обеспечения).

8. Сгруппировать множества записей каталога для переноса с учетом приоритета их использования.

Для перехода от иерархической структуры каталога к предлагаемой сетевой требуется сгруппировать записи каталога согласно выбранным условиям группирования и приоритетности условий. Так как одна и та же запись каталога может удовлетворять нескольким условиям группирования, исходную запись следует располагать в ветви сервиса определяющего условие с наибольшим приоритетом.

Для формирования подмножеств записей каталога из исходного множества всех записей в заданной ветви каталога с учетом приоритетности условий группирования предлагается использовать соответствующий разработанные метод разбиения множества параметров каталога, хранящихся в службе каталогов, на подмножества с учётом приоритета их использования, который будет описан далее.

9. Перестроить структуру хранения данных службы каталогов с учетом приоритетов использования параметров службы каталогов. Определив на предыдущем шаге принадлежность и приоритетность параметров каталога, требующих переноса в ветви сервисов, следует перестроить структуру каталога с использованием предлагаемого метода перехода от иерархической структуры хранения данных без учёта приоритетов использования параметров к структуре с учётом приоритетов использования параметров, который будет описан далее.

Для построения предлагаемой сетевой структуры каталога следует учитывать условия принадлежности записи к определённому сервису, а также приоритетность таких условия. Одна и та же запись каталога может соответствовать сразу несколькими условиям группирования

(сервисам). Это следует учитывать и располагать такие записи в ветви сервиса с наибольшим приоритетом, а в ветвях сервисов с меньшим приоритетом использовать записи-ссылки.

Структура хранения записей, атрибуты записей влияют на временные характеристики служб каталогов, как это было определено ранее, поэтому в данном методе записи каталога рассматриваются как параметры каталога.

Процесс определения параметров, требующих перемещения в дереве каталога, можно представить в виде предлагаемого ниже метода разбиения множества параметров каталога, хранящихся в службе каталогов, на подмножества с учётом приоритета их использования с целью уменьшения времени отклика на запрос.

**Входные данные:**

- количество сервисов  $n$ ;
- множества атрибутов и классов для каждого сервиса  $B_i$ , где  $i=1, \dots, n$ ;
- критерии приоритетности сервисов  $k$ , где 1 — наивысший приоритет,  $k=1, \dots, n$ ;
- множество параметров каталога  $E$ , где каждый элемент есть множество атрибутов параметра каталога  $e_j$ , такое что  $e_{j0}$  содержит атрибут определяющий уникальное имя параметра — DN (distinguished name),  $j=1, \dots, |E|$ .

**Выходные данные:**

- множества параметров  $V_i$  для каждого заданного сервиса, где  $i=1, \dots, n$

**Метод разбиения множества параметров каталога, хранящихся в службе каталогов, на подмножества с учетом приоритета их использования:**

1. Определение уникального множества используемых атрибутов  $C$  для всех заданных сервисов. По множеству атрибутов и классов для каждого сервиса  $B_i$  определяется уникальное множество используемых атрибутов  $C$  для текущей конфигурации каталога;

$$C = ((B_i \setminus B_{i+1}) \cup \dots \cup (B_{n-1} \setminus B_n)) \cup (B_n \setminus ((B_i \setminus B_{i+1}) \cup \dots \cup (B_{n-1} \setminus B_n))) \quad (7)$$

где  $i=1..n$ ,  $n$  - количество сервисов,  $B_i$  - множество атрибутов для определённого сервиса,  $C$  - уникальное множество атрибутов.

2. Формирование множеств параметров  $V_k$  для заданных сервисов.  $V_k$  - есть множество элементов  $e_{j0}$  множества  $e_j$ , для которого хотя бы один элемент содержится в множестве  $C$ , и все элементы множества  $B_i$  содержатся в  $e_j$ , где  $j = 1, \dots, |E|$ ,  $i = k$ ,  $k = 1, \dots, n$ ;

$$V_k = \{e_{j0} | \exists c \in C \wedge c \in e_j \wedge B_i \subset e_j\} \quad (8)$$

3. Присваивание начальных значений переменным.  $k=1$ .

4. Определение множеств параметров  $V_k$  с учётом приоритетов заданных сервисов. При условии  $k \leq n-1$ , если параметр сервиса с приоритетом равным  $k$  содержится в множестве параметров сервисов с меньшим приоритетом, то параметр сохраняется в множестве сервиса с приоритетом  $k$ , а в множествах с меньшим приоритетом параметр заменяется параметром, содержащим запись-ссылку на параметр с наибольшим приоритетом.

$$V_{k+1} = (V_{k+1} \setminus v) \cup vl \quad (9)$$

В противном случае, перейти к шагу 5. При невыполнении условия  $k \leq n-1$ , перейти к шагу 6.

5. Обновление значения переменной  $k$ . Увеличить значение переменной  $k$  на 1 и перейти к шагу 3.

6. Конец метода. Результатом выполнения метода будут множества параметров  $V_i$ , отсортированные с учетом приоритета, для каждого заданного сервиса.

Определив все параметры каталога, требующие переноса в заданные ветви сервисов, метод перехода от иерархической структуры хранения данных без учёта приоритетов

использования параметров к структуре с учётом приоритетов использования параметров на основе анализа основных параметров работы службы каталогов и способов хранения данных можно сформулировать следующим образом.

**Входные данные:**

- исходный граф  $G(V, E)$ , описывающий стандартную иерархическую структуру;
- множества параметров  $V_i$  для заданного набора сервисов.

**Выходные данные:**

- граф  $G'(V', E')$ , описывающий сетевую структуру каталогов.

**Метод перехода от иерархической структуры хранения данных без учёта приоритетов использования параметров к структуре с учётом приоритетов использования параметров:**

1. Добавление дополнительных вершин и дуг сервисов к исходному графу  $G$ . Для того чтобы перенести параметры каталога из исходной ветви в ветвь соответствующего сервиса, к исходному графу  $G$  добавляются вершины  $vo_j$  и дуги  $\{vd, voj\}$  сервисов. Результатом данного шага будет новый граф  $G' = (V', E')$  с соответствующими множествами вершин  $V'(G')$  и дуг  $E'(G')$ .

$$\begin{aligned} V'(G') &= V'(G) \cup vo_j, \\ E'(G') &= E'(G) \cup \{vd, voj\}, \end{aligned} \quad (10)$$

где  $vo_j$  - вершины сервисов,  $j=1, \dots, n$ ,  $n$  - количество сервисов,  $vd$  - начальная вершина построения структуры;

2. Присваивание начальных значений переменным.  $k=1$ ;
3. Проверка условия  $k \leq n$ . Если условие верно, перейти к шагу 4, иначе перейти к шагу 10.
4. Присваивание начальных значений переменным.  $j=1$ ;
5. Добавление вершин и дуг параметров в заданной ветви сервиса. Пока  $j \leq |V_k|$ , для параметра  $v_j$  множества параметров  $V_k$  сервиса с приоритетом равным  $k$ , если  $v_j$  не является параметром-ссылкой, к графу полученному на предыдущем шаге, следует добавить вершину соответствующую параметру  $v_j$  с дугой  $(vo_k, v_j)$  и перейти к шагу 6.

$$\begin{aligned} V'(G') &= V'(G') \cup v_j, \\ E'(G') &= E'(G') \cup \{vo_k, v_j\}, \end{aligned} \quad (11)$$

Если  $v_j$  является параметром-ссылкой, к графу, полученному на предыдущем шаге, следует добавить вершину-ссылку  $vl_j$ , соответствующую параметру  $v_j$  с дугами  $(\{vo_k, vl_j\}, \{vl_j, v_j\})$ , и перейти к шагу 7;

$$\begin{aligned} V'(G') &= V'(G') \cup vl_c, \\ E'(G') &= E'(G') \cup \{\{vo_k, vl_j\}, \{vl_j, v_j\}\} \end{aligned} \quad (12)$$

6. Удаление исходных вершин параметров. Для исключения дублирования информации необходимо удалить исходную вершину параметра  $v_j$ , и инцидентную вершине  $v_j$  дугу  $\{vp, v_j\}$  в исходной ветви каталога, где  $vp$  - вершина-родитель для вершины  $v_j$  в исходной ветви;

$$\begin{aligned} V'(G) &= V'(G') \setminus v_j, \\ E'(G) &= E'(G') \setminus \{vp, v_j\} \end{aligned} \quad (13)$$

7. [Опционально] Создание вершин-ссылок в исходной ветви параметра. С целью совместимости конфигураций прочих сервисов с новой структурой для созданной вершины  $v_j$  в ветви сервиса создаётся соответствующая вершина-ссылка  $vnl_j$  в исходной ветви и дополнительные дуги ( $\{v_p, vnl_j\}, \{vnl_j, v_j\}$ ).

$$\begin{aligned} V'(G') &= V(G') \cup vnl_j, \\ E'(G') &= E(G') \cup \{\{v_p, vnl_j\}, \{vnl_j, v_j\}\} \end{aligned} \quad (14)$$

8. Обновление значения переменной  $j$ . Увеличить значение переменной  $j$  на 1 и перейти к шагу 5;

9. Обновление значения переменной  $k$ . Увеличить значение переменной  $k$  на 1 и перейти к шагу 3;

10. Конец метода. Результатом выполнения метода будет новый граф  $G'$  описывающий сетевую структуру каталога.

Предлагаемые выше методы были реализованы в виде программы для ЭВМ «Система формирования сетевой структуры каталога», которая предназначена для формирования сетевой структуры хранения данных службы каталогов с учетом приоритетов обрабатываемых данных из иерархической без учета приоритетов данных.

Формирование сетевой структуры каталога осуществляется по заданным параметрам сервисов, использующих службу каталогов:

- имя сервиса;
- критерий приоритетности данных сервиса;
- набор атрибутов, определяющий принадлежность записи к сервису.

Для каждого заданного сервиса создаётся отдельная ветвь каталога. В зависимости от критерия приоритетности сервиса, записи, определяемые как соответствующие этому сервису, переносятся в ветку сервиса или создаются записи-ссылки на записи в ветви сервиса с большим приоритетом.

Программа написана на высокоуровневом языке программирования общего назначения Python 3 с использованием следующих модулей расширения:

- logging (модуль протоколирования);
- ldap (модуль протокола LDAP);
- wx (модуль работы с графическим интерфейсом).

Данный выбор языка программирования и набора модулей расширений обеспечивают кроссплатформенность запуска и исполнения программы в рабочих окружениях различных операционных систем. Программа обладает графическим интерфейсом со встроенными средствами помощи.

Основные результаты данного раздела можно сформулировать следующим образом:

1. Предложена сетевая структура хранения данных.
2. Разработана методика перепроектирования структуры хранения данных службы каталогов для уменьшения времени отклика при запросе к данным.
3. Сформулированы условия применения методики к исследуемому каталогу.
4. Разработан метод разбиения множества параметров каталога, хранящихся в службе каталогов, на подмножества с учётом приоритета их использования с целью уменьшения времени отклика на запрос.
5. Разработан метод перехода от иерархической структуры хранения данных без учёта приоритетов использования параметров к структуре с учётом приоритетов использования параметров на основе анализа основных параметров работы службы каталогов и способов хранения данных.

6. Разработана программа для ЭВМ «Система формирования сетевой структуры каталога» № 2019614753, дата гос.рег. 11.04.2019.

**В четвёртом разделе** проведено тестирование разработанных методики и методов. Для этого смоделирована служба каталогов и проведён сравнительный анализ времени отклика на

запрос поиска значения атрибута записи для обеих структур смоделированного каталога по следующим параметрам:

- время отклика на запрос;
- общее количество записей каталога;
- занимаемое дисковое пространство;
- кратчайший путь до записи.

Для анализа времени отклика на запрос поиска значения атрибута записи, общего количества записей каталога, занимаемого дискового пространства по условиям сформулированным в разделе 2 был смоделирован каталог, работающий со следующими ролями сервера:

- файловый сервер;
- прокси-сервер.

Таким образом, служба каталогов используется двумя сервисами с соответствующими видами записей и наборами атрибутов.

Рассматриваются записи трёх видов:

- записи, содержащие атрибуты только файлового сервиса;
- записи, содержащие атрибуты только прокси сервиса;
- записи, содержащие атрибуты обоих сервисов.

Для службы каталогов с иерархической структурой хранения данных изначальная база данных записей каталога формировалась случайным образом. По добавлении 10000 записей, для каждого сервиса создавались запросы поиска соответствующего сервису атрибута записи, где фильтр поиска был сгенерирован случайным образом атрибутом `uid` и атрибутом объектного класса сервиса.

Используемые фильтры поиска:

- `(&(uid=%u)(objectClass=sambaSamAccount))`
- `(&(uid=%u)(objectClass=proxyAccount))`

Данные фильтры позволяют строго определить искомую запись сервиса. Атрибут объектного класса со значением «`sambaSamAccount`» определяет конечную выборки записей файлового сервера из общего количества записей, а атрибут объектного класса «`proxyAccount`» определяет конечную выборки записей прокси-сервера из общего количества записей.

Запрос поиска повторялся 1000 раз с разными сгенерированными значениями атрибута `uid`. Далее был произведён расчёт среднего значения времени отклика на запрос, значения медианы и среднего квадратического отклонения. Общее количество сгенерированных случайным образом записей составляло 150000.

Полученное количество записей по принадлежности к виду:

- записи, содержащие атрибуту только файлового сервиса - 77171;
- записи, содержащие атрибуту только прокси сервиса - 37458;
- записи, содержащие атрибуты обоих сервисов — 37315.

Зависимость времени отклика на запрос поиска значения атрибута записи файлового сервера от общего количества записей каталога для обеих структур изображена на рисунке 5, а аналогичная зависимость для прокси-сервера изображена на рисунке 6. Для построения графиков использовались средние значения, а экспериментальные данные изображены красными точками.



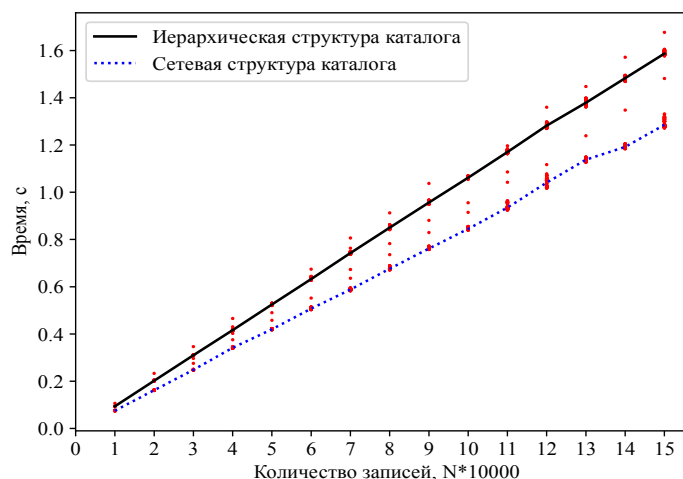


Рисунок 5 - Зависимость времени отклика на запрос поиска значения атрибута записи прокси-сервера от общего количества записей каталога.

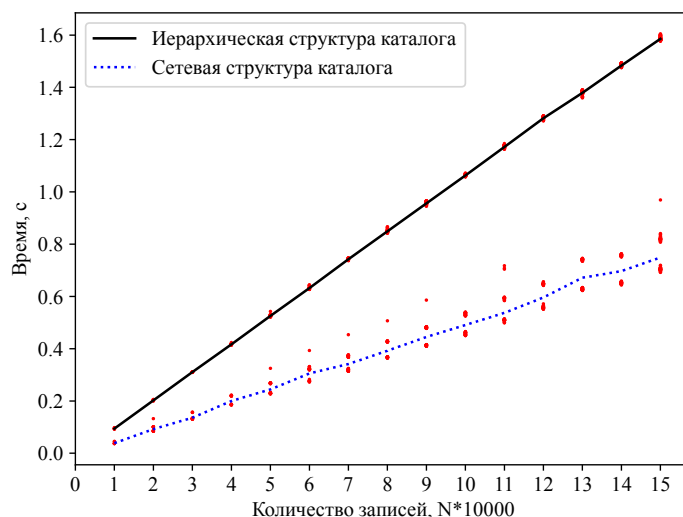


Рисунок 6 - Зависимость времени отклика на запрос поиска значения атрибута записи прокси-сервера от общего количества записей каталога.

Время отклика на запрос поиска значения атрибута записи при сетевой структуре каталога с учётом приоритетов использования параметров меньше для обоих фильтров поиска, чем при стандартной иерархической структуре без учёта приоритетов. Наибольший приоритет был отдан роли файлового сервера, однако среднее значение уменьшения времени отклика на запрос для файлового сервера составило 16.1%, что меньше чем, для прокси-сервера, где среднее значение улучшения составило 54.7%. Это обусловлено такими факторами как:

- количество записей файлового сервера больше, чем прокси-сервера;
- записи-ссылки и записи прокси-сервера меньше, чем записи файлового сервера.

Оба фактора рассматривались ранее во втором разделе, а полученные результаты соответствуют полученным выводам раздела. Однако, такое построение каталога, позволяет применить разные методы оптимизации, инструменты хранения и обработки данных или даже другое физическое расположение ветви каталога для разных веток каталога.

Результаты времени отклика на запрос поиска значения атрибута записи в исходной ветви каталога с использованием сетевой структуры ожидаемы и обусловлены увеличением количества записей для изначальной выборки более чем в два раза, так как одна запись исходной структуры может удовлетворять сразу обоим условиям группирования, иметь сразу две записи-ссылки: в исходной ветви поиска и в одной из веток сервиса. Также на полученные

результаты влияют размер записей-ссылок и их распределение между ветвями сервисов. По причине возможного увеличения времени отклика на запрос поиска в исходной ветви каталога предлагаемый метод перехода от иерархической структуры хранения данных к сетевой включает именно опциональный шаг создания вершины-ссылки в исходной ветви параметра.

Как было определено ранее в разделе 2, индексирование является действенным способом улучшения эффективности работы службы каталогов, зависящим от типов индексов, фильтров поиска и общего количества записей каталога.

С целью установления степени воздействия перепроектирования структуры хранения данных моделируемой службы каталогов на эффективность применения индексирования, атрибуты, используемые в запросах поиска записей сервисов (`uid`, `objectclass`), были проиндексированы с использованием трех рассматриваемых ранее типов индексов:

- `eq` (индекс эквивалентности значения);
- `pres` (индекс присутствия);
- `sub` (индекс подстроки).

Для каждого сервиса были собраны статистические данные времени отклика на запрос поиска значения соответствующего сервису атрибута записи идентичному запросам произведенным ранее для иерархической и сетевой структур каталога, идентификатор которой был сгенерирован случайным образом из всех возможных значений, хранящихся в заданный момент времени поиска.

Зависимости времени отклика на запрос поиска значения атрибута записи файлового сервера и прокси-сервера для обеих структур с индексами эквивалентности, присутствия и подстроки изображены на рисунке 7. Для построения графиков использовались средние значения, а экспериментальные данные изображены красными точками.

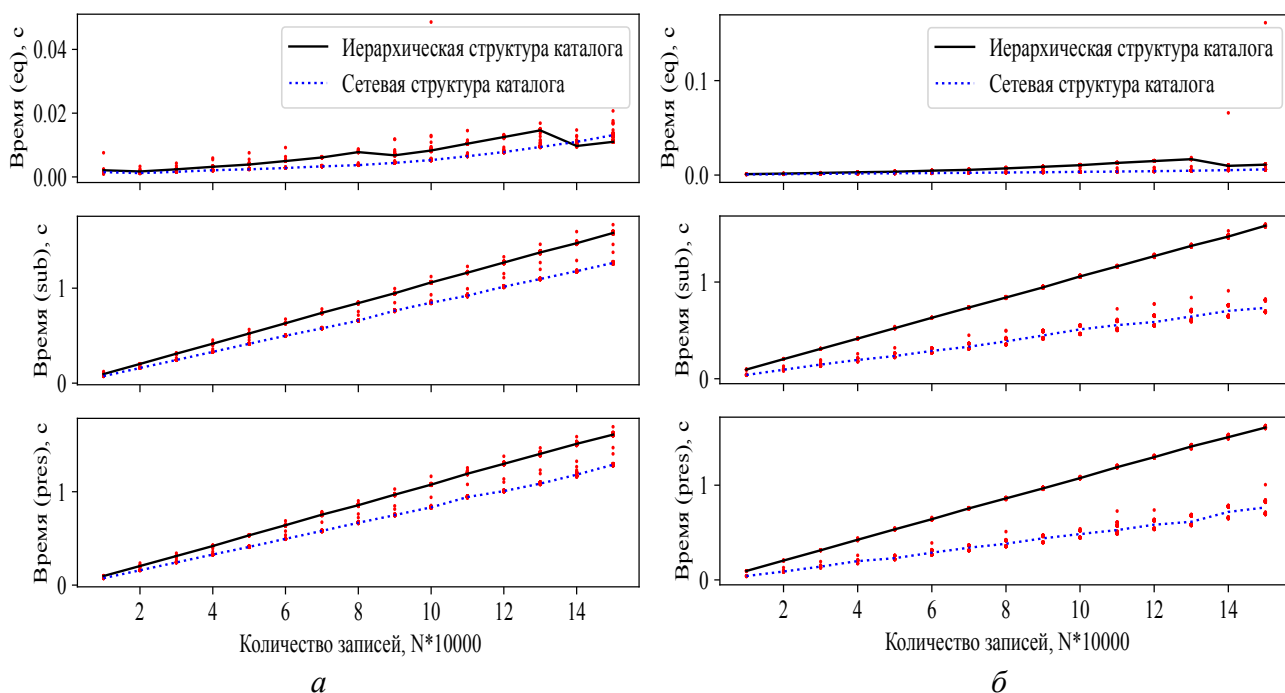


Рисунок 7 - Зависимости времени отклика на запрос поиска значения атрибута записи для обеих структур с индексами эквивалентности, присутствия и подстроки: *а* - файловый сервер; *б* - прокси-сервер

Полученные результаты времени отклика на запрос поиска значения атрибута записи для соответствующей ветви сервиса соответствуют результатам полученным в разделе 2, а именно изменение структуры каталога по критерию принадлежности к сервису, использующему запись, и условию приоритетности файлового сервера позволило существенно уменьшать изначальное количество записей для выборки, а соответственно и время отклика на запрос, в случае

применения индексов подстроки и присутствия. Результаты для исходной ветви каталога с применением индексов подстроки и присутствия соответствуют результатам, полученным ранее в данном разделе. В случае индексов эквивалентности результаты не имеет большой разницы, что демонстрирует отсутствие значительного ухудшения или улучшения использования исходной ветви каталога после изменения структуры в случае применения индексов эквивалентности.

Общее количество записей каталога увеличилось за счёт дополнительных записей-ссылок, однако, количество записей в сформированных ветвях сервисов меньше, чем общее количество записей при иерархической структуре каталога. Уменьшение изначального количества записей для запросов сервисов уменьшает общее количество вершин для перебора и путь до искомой вершины должен уменьшиться, поэтому для сравнения полученной сетевой структуры хранения данных с иерархической, стандартной, используемой по умолчанию, предлагается использовать алгоритм поиска кратчайшего пути Дейкстры, где поиск кратчайшего пути осуществляется на одном уровне заданной ветви каталога.

Применение алгоритма поиска кратчайшего пути Дейкстры показало, что путь до искомой вершины при сетевой структуре каталога короче, чем при стандартной иерархической, а общее время поиска значения атрибута записи, соответственно, должно осуществляться быстрее, что подтверждается результатами полученными эмпирическим путём.

Сравнительный анализ показал, что применение сетевой структуры хранения данных каталога имеет свои достоинства и недостатки в сравнении с иерархической структурой каталога.

К достоинствам сетевой структуры каталога в сравнении с иерархической относится следующее:

- при одинаковых условиях запроса, время отклика на запрос при сетевой структуре меньше, чем при иерархической;
- возможность применения различных дополнительных методов оптимизации работы службы каталогов для записей в разных ветвях каталога, например индексирование.

К недостаткам сетевой структуры каталога в сравнении с иерархической относится следующее:

- увеличение общего количества записей каталога и незначительное увеличение дискового пространства требуемого для хранения данных;
- сложность структуры.

К основным результатам четвертого раздела следует отнести следующее:

- изменение структуры каталога по заданным критериям позволяет существенно уменьшить время отклика на запрос;
- количество записей каталога и распределение записей между сервисами являются основными факторами влияющими на время отклика на запрос.

В **заключении** приведены основные результаты работы, в соответствии с целью и задачами сформулированными во введении.

## **Заключение**

В диссертационной работе решена актуальная задача разработки методики перепроектирования структуры хранения данных службы каталогов для уменьшения времени отклика сервера каталога на запросы клиентов к данным в высоконагруженных вычислительных сетях.

1. Определены рабочее окружение существующих служб каталогов, основные параметры работы служб каталогов и способы хранения данных. Сформулированы условия работы служб каталогов LDAP.

2. Предложена сетевая структура хранения данных службы каталогов, позволяющая уменьшить время отклика на запрос за счёт учета приоритета используемых параметров, полученных на этапе анализа и перестроения структуры данных.

3. Предложен метод разбиения множества параметров каталога, хранящихся в службе каталогов, на подмножества с учётом приоритета их использования.

4. Предложен метод перехода от иерархической структуры хранения данных (без учёта приоритетов использования параметров) к сетевой структуре с учётом приоритетов использования параметров.

5. Разработана методика перепроектирования структуры хранения данных службы каталогов для уменьшения времени отклика сервера службы каталогов на запросы клиентов к данным.

В результате применения предложенных решений, методики и методов основная цель была достигнута, что демонстрируют теоретические и эмпирические результаты. Приведенные акты о внедрении результатов работы подтверждают эффективность разработанной методики перепроектирования структуры хранения данных.

На основе решённых в диссертации задач можно определить следующие **направления дальнейших исследований**:

1. Совершенствование метода формирования множеств параметров каталога, требующих изменения структуры хранения данных с целью построения служб каталогов на базе графовых баз данных.

2. Разработка метода построения службы каталогов с использованием графовых баз данных.

#### **Публикации в изданиях, рекомендованных ВАК РФ**

1. Андреев, А.В. Методика оптимизации службы каталогов / А.В. Андреев // Информационно-управляющие системы. - 2014. - № 2. - С. 40

2. Андреев, А.В. Математическая модель службы каталогов / А.В. Андреев // Информационно-управляющие системы. - 2014. - № 4. - С. 85.

3. Андреев, А.В. Сетевая модель данных службы каталогов / А.В. Андреев // Труды МФТИ. - 2014. - Т6. - С. 27

#### **Публикации в изданиях, индексируемых в Scopus**

4. Андреев, А.В. Структура каталога как фильтр поиска записей службы каталогов OpenLDAP. / А.В. Гордеев, А.В. Андреев // Информационно-управляющие системы. - 2019 - № 2. С. 52-56. <https://doi.org/10.31799/1684-8853-2019-2-52-56>

#### **Публикации в других изданиях**

5. Андреев, А.В. OpenLDAP как замена Active Directory / А.В. Андреев // LinuxFormat. - 2011. - № 3. - С. 40-43.

6. Андреев, А.В. Службы с LDAP-аутентификацией / А.В. Андреев // LinuxFormat. - 2011. - № 5. - С. 44-47.

7. Андреев, А.В. Метод проектирования структуры хранения данных службы каталогов для уменьшения времени отклика при запросе к большим объемам данных / А.В. Андреев // Интеллектуальные технологии на транспорте. - 2019. - Вып. 19. - С. 5-10.

#### **Свидетельства о государственной регистрации программы для ЭВМ**

8. Свидетельство 2019614753 РФ. Система формирования сетевой структуры каталога: свидетельство об официальной регистрации программы для ЭВМ / А.В. Андреев; заявитель и патентообладатель СПбГУАП; дата гос. рег. 11.04.2019.