УДК 519.688
doi:10.31799/1684-8853-2019-2-52-56

Научные статьи
Articles

Структура каталога как фильтр поиска записей службы каталогов OpenLDAP

А. В. Гордеев[a], *доктор техн. наук, профессор, orcid.org/0000-0001-5390-3498, ff2avg@mail.ru*
А. В. Андреев[a], *аспирант, orcid.org/0000-0001-5743-0229*
[a]*Санкт-Петербургский государственный университет аэрокосмического приборостроения, Б. Морская ул., 67, Санкт-Петербург, 190000, РФ*

Введение: *количество публикаций по исследованию структур каталогов ограничено видимой простотой закрытых, проприетарных, реализаций. Эта закрытость накладывает ограничения на возможности исследования служб каталогов, особенно вопросов, касающихся изменения структуры каталога. Открытые решения, такие как OpenLDAP, получили популярность относительно недавно, и большинство исследований сводятся к вопросам интеграции открытых решений в существующую инфраструктуру.* ***Цель:*** *определение влияния структуры хранения данных каталога при различных параметрах индексирования на время отклика на запрос к каталогу.* ***Результаты:*** *основными параметрами, влияющими на эффективность использования индексирования при работе со службами каталогов, определены количество записей каталога и типы индексов, использующиеся в фильтре поиска. Предложен новый подход формирования структуры службы каталогов с учетом параметров поиска информации. Выявлено, что при наличии индексируемых атрибутов у большого количества записей каталога наименее эффективными являются индексы типов присутствия и подстроки. Дополнительные ветки каталога, созданные по определенным критериям записей, были заданы как исходные ветки поиска записей и использовались как дополнительный фильтр поиска. Данная модификация структуры каталога позволила дополнительно уменьшить общее число записей при изначальной выборке. Такое решение существенно улучшило время отклика на запрос к каталогу при использовании индексов присутствия и подстроки в фильтре поиска, а также сократило фильтр поиска.* ***Практическая значимость:*** *формирование структуры каталога с учетом параметров записей и способов поиска параметров записей позволяет уменьшить время отклика каталога на запрос.*

Ключевые слова *— служба каталогов, структура каталога, OpenLDAP, индексирование, тип индекса.*

Для цитирования: Гордеев А. В., Андреев А. В. Структура каталога как фильтр поиска записей службы каталогов OpenLDAP. *Информационно-управляющие системы*, 2019, № 2, с. 52–56. doi:10.31799/1684-8853-2019-2-52-56
For citation: Gordeyev A. V., Andreyev A. V. OpenLDAP directory service structure as a search filter. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2019, no. 2, pp. 52–56 (In Russian). doi:10.31799/1684-8853-2019-2-52-56

Введение

Служба каталогов (Directory Service) — средство иерархического представления различных ресурсов сети и хранения информации об этих ресурсах. В качестве ресурсов могут выступать персонал, различные информационно-вычислительные ресурсы, сетевые ресурсы [1] и т. д. Информация об определенном объекте (ресурсе) хранится в виде набора значений атрибутов этого объекта. Самые распространенные службы каталогов — Active Directory, OpenLDAP, 389-DS и др. [2]. Различные службы каталогов широко используются малыми, средними и крупными предприятиями, где служба каталогов не только применяется для хранения учетных записей и аутентификации пользователей, но и выступает в роли механизма авторизованного доступа к ресурсам сети [3]. Многие продукты, например почтовый сервер Zimbra, в основе своей архитектуры используют службу каталогов [4]. Также существуют облачные реализации служб каталогов, например, JumpCloud предлагает облачный продукт Directory-as-a-Service, использующий OpenLDAP [5].

Количество исследований и, соответственно, публикаций по данной тематике не так велико и лимитировано популярностью, а также видимой простотой закрытых, проприетарных, реализаций. Закрытость таких реализаций накладывает ограничения на возможности исследования служб каталогов, особенно вопросов, касающихся изменения структуры каталога. Компании, реализующие закрытые решения, не раскрывают архитектуру построения служб каталогов и являются основным источником информации. Например, компания Microsoft является разработчиком популярной закрытой реализации службы каталогов Active Directory. Актуальную информацию об Active Directory компания распространяет через свой веб-сайт. Открытые решения, такие как OpenLDAP, получили популярность относительно недавно, и большинство исследований сводится к вопросам интеграции открытых решений в существующую инфраструктуру.

Службы каталогов используют структуру хранения данных B-дерево и работают по протоколу LDAP [6].

Одной из основных характеристик работы службы каталогов является время отклика на запрос поиска записи или записей.

Производительность каталога зависит от множества факторов, например:

— аппаратных характеристик серверов и сетевой инфраструктуры;

— видов записей;

— количества записей в базе данных службы каталога.

Ранее проводимые исследования рассматривают различные аспекты внедрения службы каталогов в инфраструктуру вычислительной сети, где структура каталога формируется по организационной принадлежности объекта [7]. С увеличением общего количества сервисов, поддерживающих работу по протоколу LDAP, увеличивается и количество применяемых атрибутов объектов, что накладывает ограничения на подходы в целях оптимизации служб каталогов.

Один из таких подходов — использование индексирования [8].

Параметры индексирования записей в службе каталога

Индексирование атрибутов записей служб каталогов — распространенная практика оптимизации поиска записей. Индексируемый атрибут используется в качестве фильтра поиска. Самый распространенный фильтр поиска — (uid=%u). Различные службы сети могут создать разнообразные комбинации фильтров, оперируя логическими операциями И, ИЛИ, НЕ [9]. Например, служба файлового сервера, реализуемая средствами samba, пользуется по умолчанию фильтром поиска (&(uid=%u)(objectClass=sambaAccount), но рекомендуется индексировать следующий набор атрибутов для оптимизации запросов: uid, cn, sambaSID, gidNumber, uidNumber, displayName [10]. Помимо файлового сервера, один каталог могут использовать разные службы: почтовый сервер, VPN-сервер, прокси-сервер, сервер репликации данных, web-серверы, серверы обмена сообщениями и различные внутренние службы сети. Все службы имеют свои наборы фильтров поиска и атрибутов хранения данных.

Индексирование имеет ряд ограничений, которые не позволяют индексировать все необходимые атрибуты для всех сервисов сети сразу:

1. Индексы занимают дисковое пространство. Для хранения данных каталога OpenLDAP по умолчанию используется бэкенд hdb, где данные хранятся в двух файлах: dn2id.bdb и id2entry.bdb. Для каждого индекса создается отдельный файл. Например, при индексировании атрибута “objectClass” будет создан файл objectclass.bdb. Размер файла зависит от выбранных типов индексов для атрибута.

2. Каждое добавление новой записи с индексированными атрибутами наравне с добавлением записи добавляет новые индексы, что влияет на скорость добавления или обновления записей [11].

3. Рекомендуется индексировать атрибуты, которые часто используются, но в сетях со множеством различных служб, одни службы используются чаще других, а другие — реже. В таком случае индексирование атрибутов редко используемых служб является напрасной тратой аппаратных ресурсов [12].

4. Использование неиндексированных атрибутов вместе с индексированными в одном фильтре поиска отрицательно сказывается на производительности такого поиска. Также количество и уникальность атрибутов фильтра влияют на скорость поиска соответствующей записи [13].

Помимо перечисленных ограничений, также существуют ограничения, накладываемые типом индекса. Следует внимательно выбирать атрибуты с типами индексирования присутствия (presence) и подстроки (substring) [14]. Если некий атрибут существует у большинства записей, то все эти записи будут в конечной выборке, потому что они все подходят под фильтр поиска. Скорость поиска будет в данном случае зависеть от количества записей. Например, если большинство пользователей организации используют почтовый сервер, применение индекса присутствия для атрибутов почтового сервиса, содержащихся в фильтре поиска, мало влияет на скорость поиска записей, так как все записи будут в конечной выборке [15].

Структура каталога

В случае неэффективности индексирования из-за ограничений, накладываемых типами фильтров, предлагается использовать структуру каталога как фильтр поиска, где каждая ветка каталога содержит только записи, относящиеся к конкретной службе или выбранному критерию.

Рассмотрим каталог, где каждая запись-пользователь имеет набор общих атрибутов “objectclass”, а также один уникальный атрибут “objectclass”, который позволяет сгруппировать эти записи по заданному критерию. Все записи хранятся в одной ветке, что является распространенной практикой построения каталога [16]. Каталог содержит 100 000 записей: 50 131 запись вида “posixAccount”, 25 053 записи вида “shadowAccount”, 24 816 записей вида “account”.

Записи “posixAccount”:

```
dn: uid=User9999,ou=People,dc=example,dc=com
ou: PosixAccount
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
```

```
objectClass: posixAccount
uid: User9999
cn: User9999_
sn: User9999--
userPassword:: c2VjcmV0
mail: User9999@example.com
uidNumber: 19999
gidNumber: 19999
homeDirectory: /home/User9999
loginShell: /bin/bash
```

Записи “shadowAccount”:

```
dn: uid=User10000,ou=People,dc=example,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: shadowAccount
ou: People
uid: User10000
cn: User10000_
sn: User10000--
userPassword:: c2VjcmV0
description: Line for ShadowAccount
```

Записи “account”:

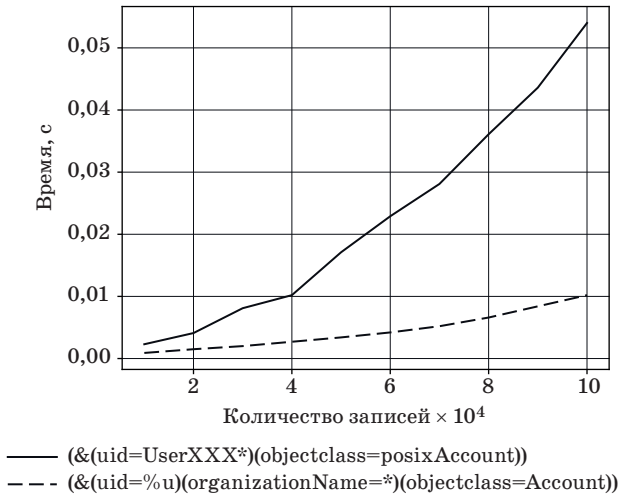
```
dn: uid=User9997,ou=People,dc=example,dc=com
ou: Account
objectClass: top
objectClass: account
uid: User9997
organizationName: User9997
```

Рассмотрим два фильтра поиска (&(uid=UserXXX*)(objectclass=posixAccount)) и (&(uid=%u)(organizationName=*) (objectclass=Account)), где XXX — сгенерированное случайное трехзначное значение от 1 до 999, %u — заданные uid, например User9997.

Первый фильтр использует индекс подстроки и осуществляет поиск заданного набора записей вида “posixAccount”. Второй фильтр использует индекс присутствия и осуществляет поиск заданной записи. Операция поиска повторяется 100 раз после добавления 10 000 записей, а по результатам высчитывается среднее значение. Результаты поиска записей для двух фильтров с использованием индексирования представлены на рис. 1.

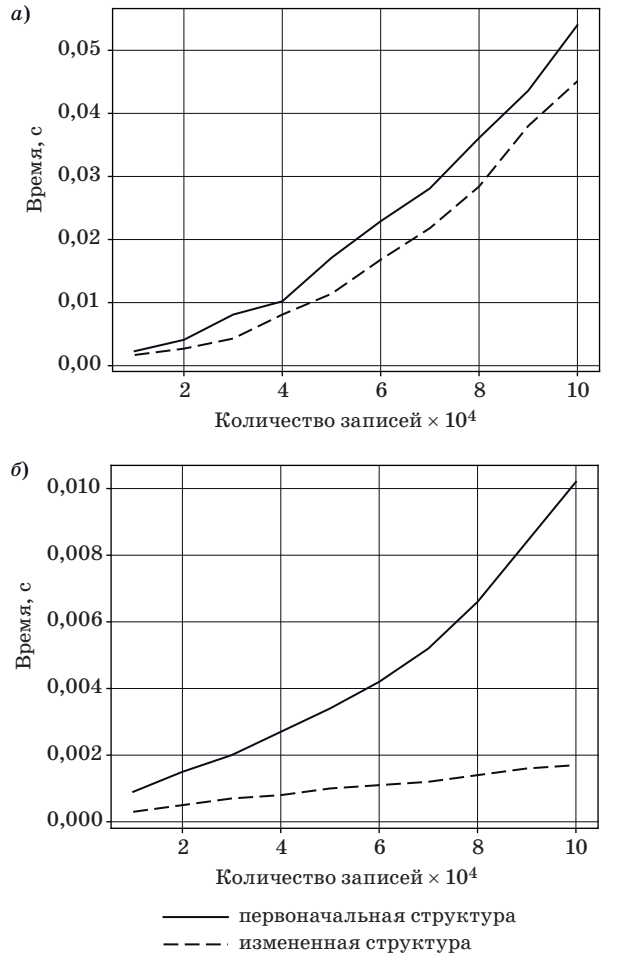
В данном случае структура каталога может быть изменена по условному критерию — типу записи. Записи “posixAccount” будут храниться в ou=PosixAccount, ou=People, dc=example, dc=com, записи “shadowAccount” — в ou=People, dc=example, dc=com, а записи “account” — в ou=Account, ou=People, dc=example, dc=com.

Такое перераспределение записей позволяет уменьшить общее количество записей поиска, а также использовать ветку поиска как фильтр поиска, сократив тем общее количество атрибутов в фильтре поиска до (uid=UserXXX*) и (&(uid=%u)(organizationName=*)). При этом сохраняется возможность поиска всех записей при



■ ***Рис. 1.*** Время поиска записей индексированного каталога

■ ***Fig. 1.*** The search time of records for the indexed directory



■ ***Рис. 2.*** Время поиска записей вида “posixAccount” (*а*) и “account” (*б*)

■ ***Fig. 2.*** The search time of records “posixAccount” (*а*) and “account” (*б*)

использовании поиска типа subtree и ветки поиска ou=People,dc=example,dc=com.

Результаты поиска записей индексированного каталога с первоначальной структурой и измененной для первого фильтра поиска изображены на рис. 2, *а*, для второго — на рис. 2, *б*.

Как видно по полученным данным, количество записей в ветке поиска, количество атрибутов в фильтре поиска и тип индекса влияют на скорость поиска записей.

Заключение

Производительность каталога зависит от множества факторов. Индексирование — хорошее решение для оптимизации службы каталогов, но данный подход имеет ряд ограничений, зависящих от того, как используется каталог и как используются данные, хранящиеся в данном каталоге.

Одними из таких ограничений являются тип фильтра и параметры его использования. Тип фильтра влияет на эффективность индексирования, например, индексы присутствия и подстроки. Одним из решений оптимизации такого каталога может быть новый подход формирования структуры службы каталогов, где ветка поиска заменяет основной фильтр поиска и уменьшает общее количество записей для выборки. Основной особенностью данного подхода является перенос заранее известного значения элемента логического выражения фильтра поиска в область, задающую начальный объект поиска записей службы каталогов. Данное изменение можно реализовать с помощью записей-ссылок, не изменяя основную структуру каталога, используя сетевую структуру хранения данных [17].

Проведенные эксперименты показали эффективность данного подхода, но конечный результат зависит от конкретного исследуемого каталога.

Литература

1. **Kretschmer F., von Arnim C., Lechler A., Verl A.** Persistent data backend for OPC UA namespaces in IT infrastructures. *Procedia CIRP*, 2018, vol. 72, pp. 174–178. doi:10.1016/j.procir.2018.03.233
2. *List of LDAP Software*. https://en.wikipedia.org/wiki/List_of_LDAP_software (дата обращения: 25.12.2018).
3. *Microsoft Active Directory*. https://discovery.hgdata.com/product/microsoft-active-directory (дата обращения: 25.12.2018).
4. *Zimbra Directory Service* (*LDAP*). https://wiki.zimbra.com/wiki/Zimbra_Directory_Service_(LDAP) (дата обращения: 25.12.2018).
5. *Top 5 Challenges with OpenLDAP*. https://jumpcloud.com/blog/top-5-challenges-with-openldap/ (дата обращения: 25.12.2018).
6. *Introduction to OpenLDAP Directory Services*. https://www.openldap.org/doc/admin24/intro.html (дата обращения: 25.12.2018).
7. **Andjarwirawan J., Palit H. N., Salim J. C.** Linux PAM to LDAP authentication migration. *2017 Intern. Conf. on Soft Computing, Intelligent System and Information Technology (ICSIIT)*, Denpasar, 2017, pp. 155–159. doi:10.1109/ICSIIT.2017.66
8. *Tuning*. https://www.openldap.org/doc/admin24/tuning.html (дата обращения: 03.01.2019).
9. *Lightweight Directory Access Protocol* (*LDAP*)*: String Representation of Search Filters*. https://tools.ietf.org/search/rfc4515#page-2 (дата обращения: 03.01.2019).
10. **Eckstein R., Carter G., Ts J.** *Using Samba*. O'Reilly Media, 2007. 600 p.
11. *Indexing Attributes in the Directory*. https://docs.oracle.com/cd/E15217_01/doc.1014/e12490/perform.htm#CFHICJIE (дата обращения: 09.01.2019).
12. *Managing Indexes*. https://access.redhat.com/documentation/en-US/Red_Hat_Directory_Server/8.2/html/Administration_Guide/Managing_Indexes.html (дата обращения: 08.01.2019).
13. **Lowe-Norris A., Desmond B., Richards J., Allen R.** *Active Directory*. O'Reilly Media, 2008. 866 p.
14. *LDAP Setup and Configuration Guide*. https://docs.oracle.com/cd/E19455-01/806-5580/6jej518pd/index.html (дата обращения: 20.11.2018).
15. *LDAP Indexes*. https://ldapwiki.com/wiki/LDAP%20Indexes (дата обращения: 08.01.2019).
16. **Butcher M.** *Mastering OpenLDAP*. Packt Publishing, 2007. 484 p.
17. **Андреев А. В.** Сетевая модель данных службы каталогов. *Тр. МФТИ*, 2014, т. 6, № 3, с. 27–36.

UDC 519.688
doi:10.31799/1684-8853-2019-2-52-56

OpenLDAP directory service structure as a search filter

A. V. Gordeyev[a], Dr. Sc., Tech., Professor, orcid.org/0000-0001-5390-3498, ff2avg@mail.ru
A. V. Andreyev[a], Post-Graduate Student, orcid.org/0000-0001-5743-0229
[a]Saint-Petersburg State University of Aerospace Instrumentation, 67, B. Morskaia St., 190000, Saint-Petersburg, Russian Federation

Introduction: The number of directory structure studies is limited by the seeming simplicity of closed proprietary implementations. This closedness complicates studying directory services, especially the issues related to changing the directory structure. Open source solutions, such as OpenLDAP, gained popularity relatively recently, and most research is about integrating them into the existing infrastructure. **Purpose:** Determining the influence of directory data-storing structure on the directory query response time, under various indexing parameters. **Results:** The main parameters affecting the indexing efficiency are the number of records in the directory and the types of indexes used in the search filters. A new approach to building up a directory service structure is proposed, taking into account the information search parameters. It has been found out that when a large number of records have indexable attributes, the most efficient indices are those of presence and substring types. Additional subtrees created according to certain criteria of the records were specified as the starting points for the search and used as an additional search filter. This approach can additionally reduce the total amount of records for the initial search and the number of attributes in the main search filter. Besides, it can significantly improve the response time. **Practical relevance:** A directory structure which takes into account the record parameters and the ways of search for them can provide a shorter response time.

Keywords — directory service, directory structure, OpenLDAP, indexing, index type.

For citation: Gordeyev A. V., Andreyev A. V. OpenLDAP directory service structure as a search filter. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2019, no. 2, pp. 52–56 (In Russian). doi:10.31799/1684-8853-2019-2-52–56

References

1. Kretschmer F., von Arnim C., Lechler A., Verl A. Persistent data backend for OPC UA namespaces in IT infrastructures. *Procedia CIRP*, 2018, vol. 72, pp. 174–178. doi:10.1016/j.procir.2018.03.233
2. *List of LDAP Software*. Available at: https://en.wikipedia.org/wiki/List_of_LDAP_software (accessed 25 December 2018).
3. *Microsoft Active Directory*. Available at: https://discovery.hgdata.com/product/microsoft-active-directory (accessed 25 December 2018).
4. *Zimbra Directory Service* (*LDAP*). Available at: https://wiki.zimbra.com/wiki/Zimbra_Directory_Service_(LDAP) (accessed 25 December 2018).
5. *Top 5 Challenges with OpenLDAP*. Available at: https://jumpcloud.com/blog/top-5-challenges-with-openldap/ (accessed 25 December 2018).
6. *Introduction to OpenLDAP Directory Services*. Available at: https://www.openldap.org/doc/admin24/intro.html (accessed 25 December 2018).
7. Andjarwirawan J., Palit H. N., Salim J. C. Linux PAM to LDAP authentication migration. *2017 Intern. Conf. on Soft Computing, Intelligent System and Information Technology (ICSIIT)*, Denpasar, 2017, pp. 155–159. doi:10.1109/ICSIIT.2017.66
8. *Tuning*. Available at: https://www.openldap.org/doc/admin24/tuning.html (accessed 3 January 2019).
9. *Lightweight Directory Access Protocol (LDAP): String Representation of Search Filters*. Available at: https://tools.ietf.org/search/rfc4515#page-2 (accessed 3 January 2019).
10. Eckstein R., Carter G., Ts J. *Using Samba*. O'Reilly Media, 2007. 600 p.
11. *Indexing Attributes in the Directory*. https://docs.oracle.com/cd/E15217_01/doc.1014/e12490/perform.htm#CFHICJIE (accessed 9 January 2019).
12. *Managing Indexes*. Available at: https://access.redhat.com/documentation/en-US/Red_Hat_Directory_Server/8.2/html/Administration_Guide/Managing_Indexes.html (accessed 8 January 2019).
13. Lowe-Norris A., Desmond B., Richards J., Allen R. *Active Directory*. O'Reilly Media, 2008. 866 p.
14. *LDAP Setup and Configuration Guide*. Available at: https://docs.oracle.com/cd/E19455-01/806-5580/6jej518pd/index.html (accessed 20 November 2018).
15. *LDAP Indexes*. Available at: https://ldapwiki.com/wiki/LDAP%20Indexes (accessed 8 January 2019).
16. Butcher M. *Mastering OpenLDAP*. Packt Publishing, 2007. 484 p.
17. Andreyev A. V. Network data model of directory services. *Trudy Moskovskogo fiziko-tekhnicheskogo instituta* [Proc. of MIPT], 2014, vol. 6, no. 2, pp. 27–36 (In Russian).