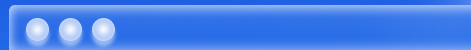





МойОфис



# Графовая БД ArangoDB для предоставления совместного доступа к документам



**Анна Рукавицына**  
Senior AQA Engineer

 @anna\_rukavitsyna96

**Владимир Ревякин**  
Senior Developer

 @revarev96



# Содержание



Для вопросов  
и комментариев

# Содержание

- Структура хранения данных на основе графовой БД ArangoDB



МойОфис



Для вопросов  
и комментариев

# Содержание

- Структура хранения данных на основе графовой БД ArangoDB
- Разработка и тестирование предоставления совместного доступа к объектам



Для вопросов  
и комментариев

# Содержание

- Структура хранения данных на основе графовой БД ArangoDB
- Разработка и тестирование предоставления совместного доступа к объектам
- Примеры багов и фиксов



МойОфис



Для вопросов  
и комментариев

# Содержание

- Структура хранения данных на основе графовой БД ArangoDB
- Разработка и тестирование предоставления совместного доступа к объектам
- Примеры багов и фиксов
- Плюсы и минусы использования ArangoDB для Dev и QA



МойОфис



Для вопросов  
и комментариев

# Содержание

- Структура хранения данных на основе графовой БД ArangoDB
- Разработка и тестирование предоставления совместного доступа к объектам
- Примеры багов и фиксов
- Плюсы и минусы использования ArangoDB для Dev и QA
- Сравнение реализации функционала в ArangoDB и PostgreSQL



МойОфис



Для вопросов  
и комментариев

# Содержание

- Структура хранения данных на основе графовой БД ArangoDB
- Разработка и тестирование предоставления совместного доступа к объектам
- Примеры багов и фиксов
- Плюсы и минусы использования ArangoDB для Dev и QA
- Сравнение реализации функционала в ArangoDB и PostgreSQL
- Технические особенности построения приложения на базе ArangoDB



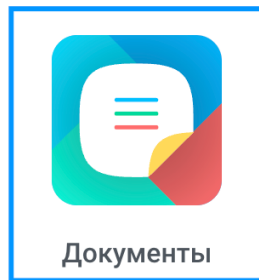
МойОфис



Для вопросов  
и комментариев



# Экосистема продуктов МойОфис



Документы



Почта



Контакты



Календарь



Профиль



Squadus



МойОфис

# **Высоконадежная, масштабируемая и отказоустойчивая СХД**

# Высоконадежная, масштабируемая и отказоустойчивая СХД

- Хранение данных

# Высоконадежная, масштабируемая и отказоустойчивая СХД

- Хранение данных
- Аутентификация

# Высоконадежная, масштабируемая и отказоустойчивая СХД

- Хранение данных
- Аутентификация
- Авторизация

# Высоконадежная, масштабируемая и отказоустойчивая СХД

- Хранение данных
- Аутентификация
- Авторизация
- Разграничение прав доступа

# Высоконадежная, масштабируемая и отказоустойчивая СХД

- Хранение данных
- Аутентификация
- Авторизация
- Разграничение прав доступа
- Файловые операции

# Высоконадежная, масштабируемая и отказоустойчивая СХД

- Хранение данных
- Аутентификация
- Авторизация
- Разграничение прав доступа
- Файловые операции
- Функции администрирования



# Высоконадежная, масштабируемая и отказоустойчивая СХД

- Хранение данных
- Аутентификация
- Авторизация
- Разграничение прав доступа
- Файловые операции
- Функции администрирования

Администрирование

Системное название

default

Отображаемое название ⓘ

new\_organisation

Поддержка

Email

Телефон

Настройки исходящих системных уведомлений

Сервер исходящей почты (SMTP)

Email

Логин

Пароль

Порт

465

SSL

Язык

Русский

Сохранить

# Высоконадежная, масштабируемая и отказоустойчивая СХД

- Хранение данных
- Аутентификация
- Авторизация
- Разграничение прав доступа
- Файловые операции
- Функции администрирования

Администрирование

Организация

- Основные настройки
- Домены
- Ключи доступа
- Безопасность
- Настройки загрузки

Пользователи

Группы

Публичные ссылки

Общие папки

Удаленные объекты

+ Новый пользователь

Поиск

Всего пользователей: 8

Максимальное количество: 1000

Выберите фильтр

Скачать список

Имя ↑	Отчество ↓	Фамилия ↓	E-mail ↓	Действия
TT Test		Test	pgs_3653@rukavitsyna.m...	✎ 🔒 🗑
TU Test		User	test_user@rukavitsyna.m...	✎ 🔒 🗑
MS Matt		Smith	newuser@rukavitsyna.my...	✎ 🔒 🗑
II Ivan		Ivanov	nextone@rukavitsyna.my...	✎ 🔒 🗑



МойОфис

Документы ▾

+ Добавить

Мои документы

- PGS
- Test Folder**
- Доступные мне
- Общие папки
- Отмеченные
- Недавние
- Корзина

> Test Folder

Имя ↑      Размер      Дата изменения      🚩

📁	My folder	--	Сегодня, 09:36	
📄	Presentation Document.xodp	17.23 КБ	Сегодня, 09:36	
📄	Table Document.xods	10.95 КБ	Сегодня, 09:35	
📄	Text Document.xodt	15.95 КБ	Сегодня, 09:35	

Общий доступ к «Table Document.xods» [Поделиться ссылкой...](#)

Владимир Ревякин

Уведомить письмом  Уведомить в Squadus

Добавить сообщение

Уведомление можно отправить только отдельным пользователям, а не группам

Добавьте пользователей

Редактирова...

- Управление**  
Можно редактировать объект и управлять доступом других пользователей к нему.
- Редактирование**  
Можно вносить изменения, создавать новые версии и добавлять комментарии.
- Просмотр**  
Можно просматривать, скачивать и копировать объект.
- Доступ запрещен**  
Нельзя просматривать объект или совершать иные действия с ним.

Закреть

# Предоставление совместного доступа на примере использования ArangoDB



# Файловая система

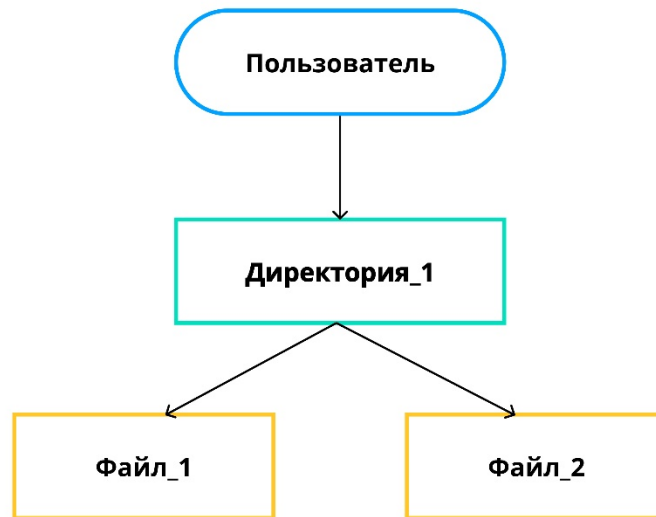
# Файловая система

**ArangoDB** — NoSQL база данных,  
реализующая графовую модель данных.



# Файловая система

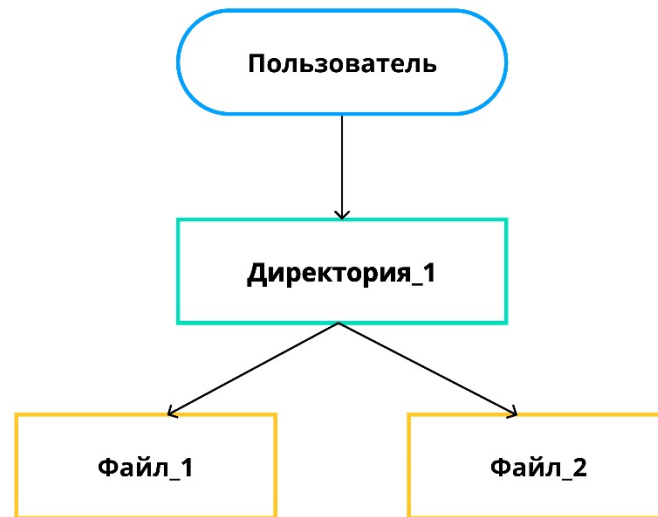
**ArangoDB** — NoSQL база данных, реализующая графовую модель данных.



# Файловая система

**ArangoDB** — NoSQL база данных, реализующая графовую модель данных.

Существуют два основных типа коллекций документов:

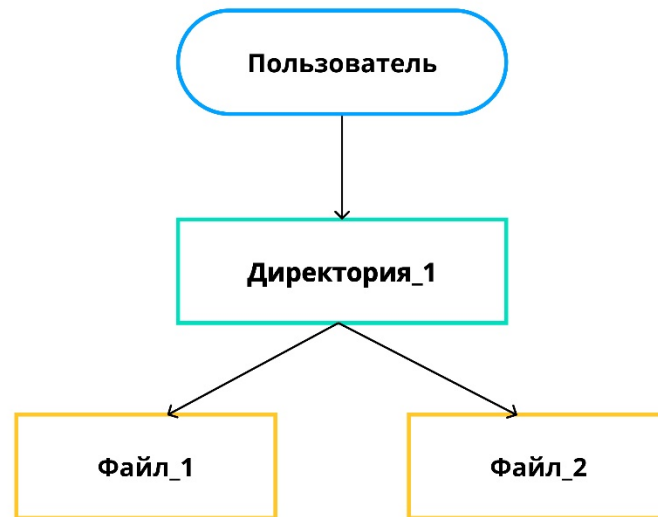


# Файловая система

**ArangoDB** — NoSQL база данных, реализующая графовую модель данных.

Существуют два основных типа коллекций документов:

- **вершины**

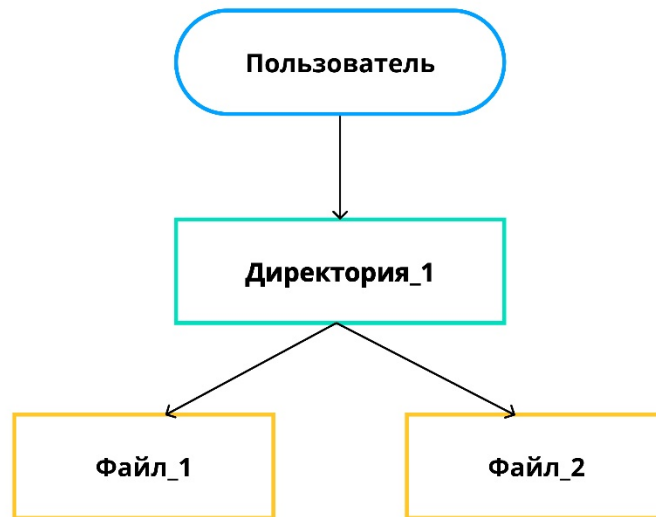


# Файловая система

**ArangoDB** — NoSQL база данных, реализующая графовую модель данных.

Существуют два основных типа коллекций документов:

- **вершины**
- **ребра** — документы, соединяющие вершины в однонаправленные графы.



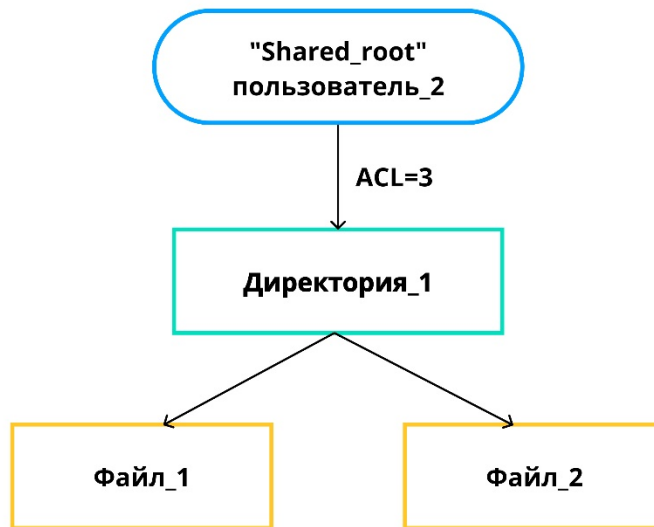
# «Шаринг» объектов

# «Шаринг» объектов

Реализация предоставления  
**совместного доступа** к документам  
на примере графовой БД ArangoDB

# «Шаринг» объектов

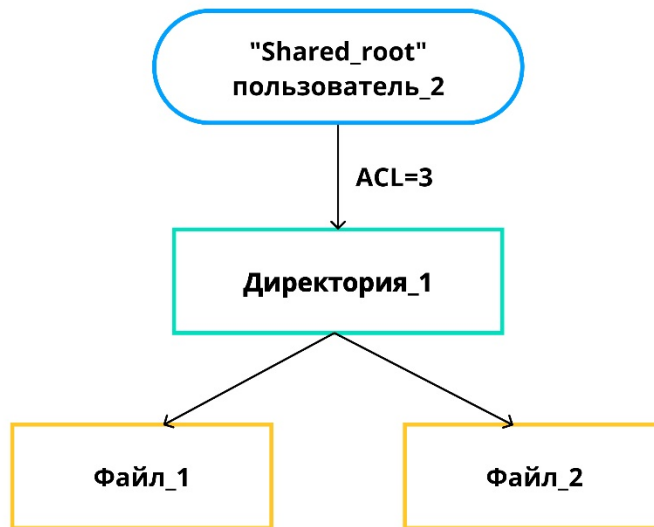
Реализация предоставления  
**совместного доступа** к документам  
на примере графовой БД ArangoDB



# «Шаринг» объектов

Реализация предоставления  
**совместного доступа** к документам  
на примере графовой БД ArangoDB

**Объект ACL (Access Control List)** —  
ребро «шаринга»  
с необходимыми правами доступа





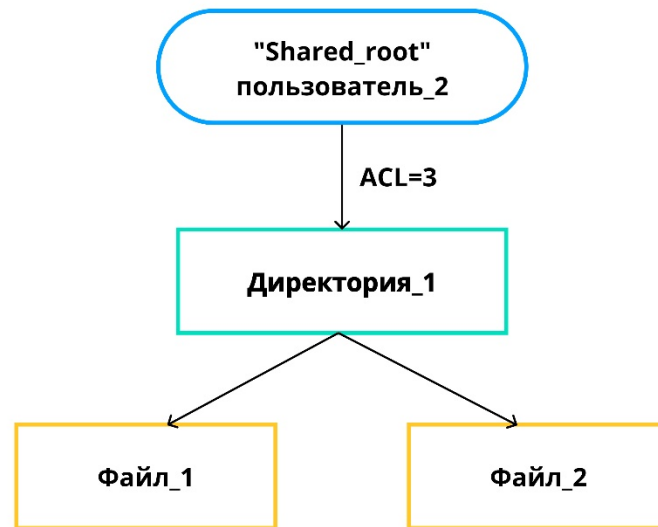
# Разрешение конфликтов

# Разрешение конфликтов

Пример возникновения конфликтной ситуации при наличии нескольких прав доступа к одному объекту

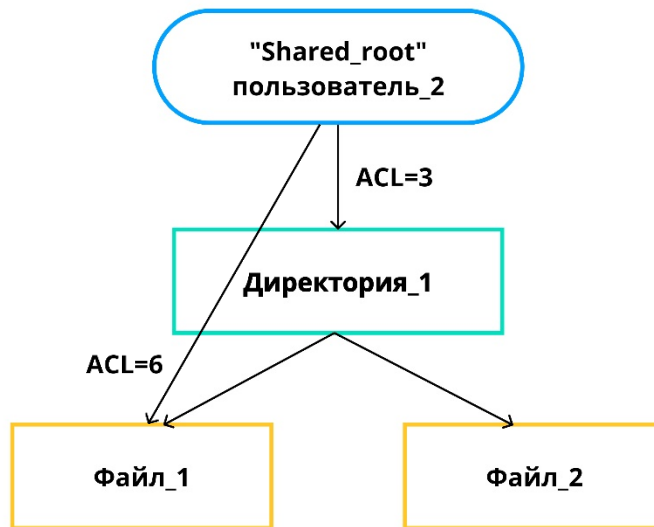
# Разрешение конфликтов

Пример возникновения конфликтной ситуации при наличии нескольких прав доступа к одному объекту



# Разрешение конфликтов

Пример возникновения конфликтной ситуации при наличии нескольких прав доступа к одному объекту

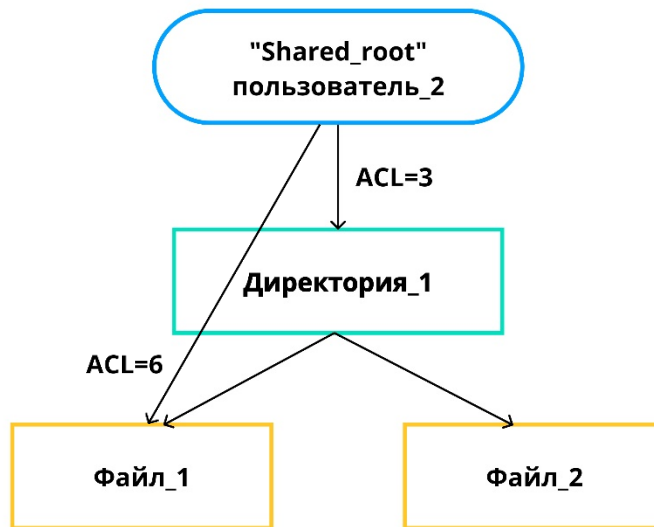


# Разрешение конфликтов

Пример возникновения конфликтной ситуации при наличии нескольких прав доступа к одному объекту

ACL=6 — **прямой** «шаринг»

ACL=3 — **унаследованный** «шаринг»



# Тезисы:

# Тезисы:

- **прямой** «шаринг» всегда преобладает над унаследованным

## Тезисы:

- **прямой** «шаринг» всегда преобладает над унаследованным
- эмпирическое правило – **более короткая длина** пути обладает большим приоритетом



# Тезисы:

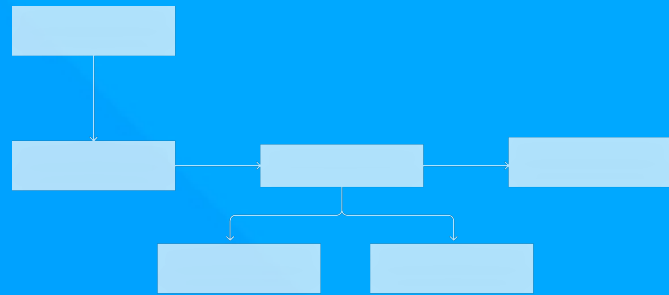
- **прямой** «шаринг» всегда преобладает над унаследованным
- эмпирическое правило – **более короткая длина** пути обладает большим приоритетом
- в случае одинаковой длины «шаринга» преобладает шаринг с **большим приоритетом ACL**

## Тезисы:

- **прямой** «шаринг» всегда преобладает над унаследованным
- эмпирическое правило – **более короткая длина** пути обладает большим приоритетом
- в случае одинаковой длины «шаринга» преобладает шаринг с **большим приоритетом ACL**

acl	Права
0	Доступ запрещен
3	Чтение
6	Запись
7	Чтение + запись
8	Управление
9	Владение

# Специфика тест-дизайна функциональности предоставления совместного доступа



# Множественный групповой «шаринг» файла

# Множественный групповой «шаринг» файла

Типы тестирования:

# Множественный групповой «шаринг» файла

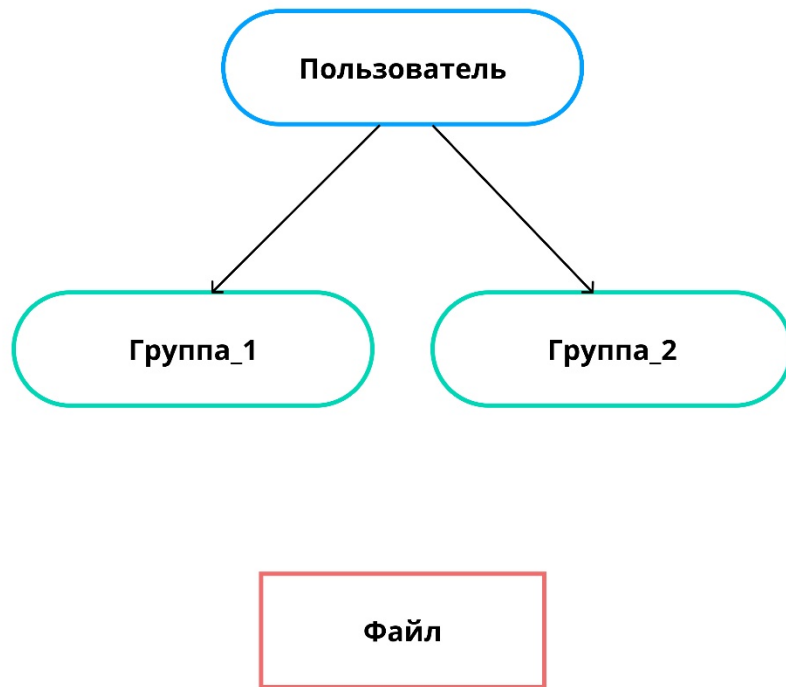
Типы тестирования:

- Конструктивное

# Множественный групповой «шаринг» файла

Типы тестирования:

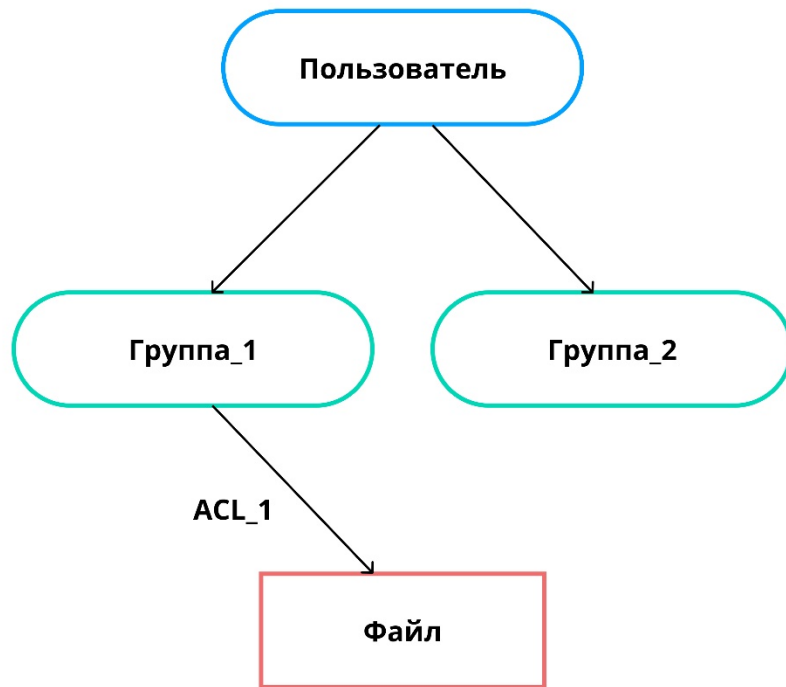
- Конструктивное



# Множественный групповой «шаринг» файла

Типы тестирования:

- Конструктивное

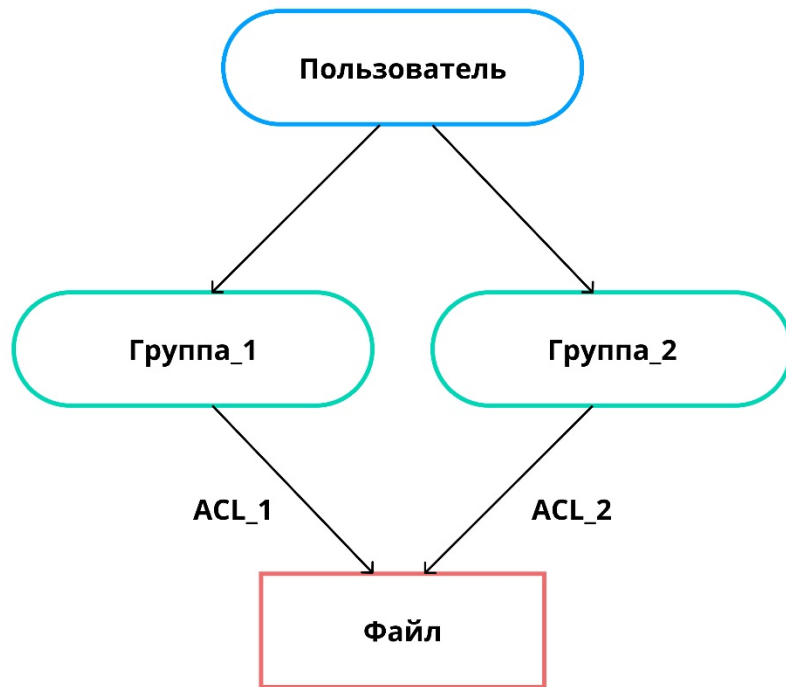




# Множественный групповой «шаринг» файла

Типы тестирования:

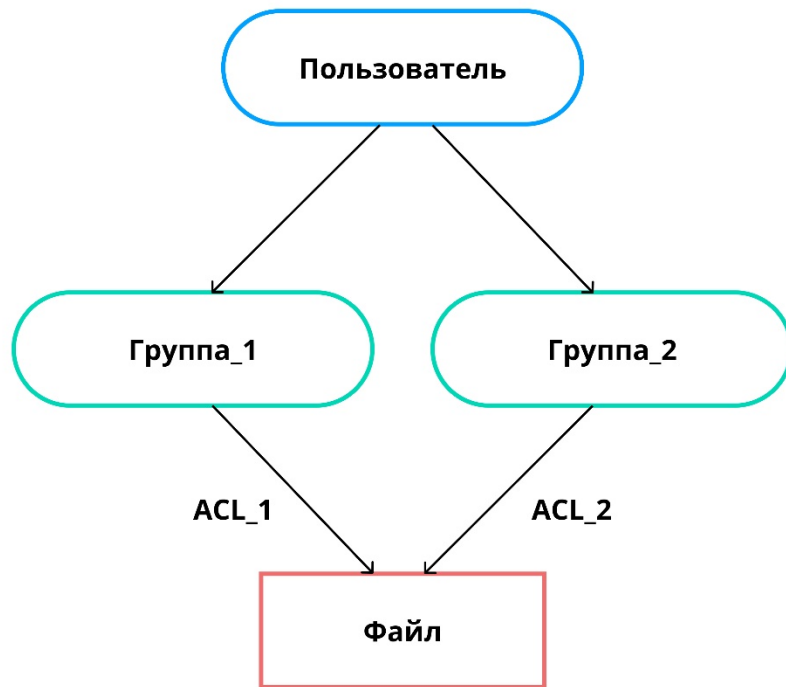
- Конструктивное



# Множественный групповой «шаринг» файла

Типы тестирования:

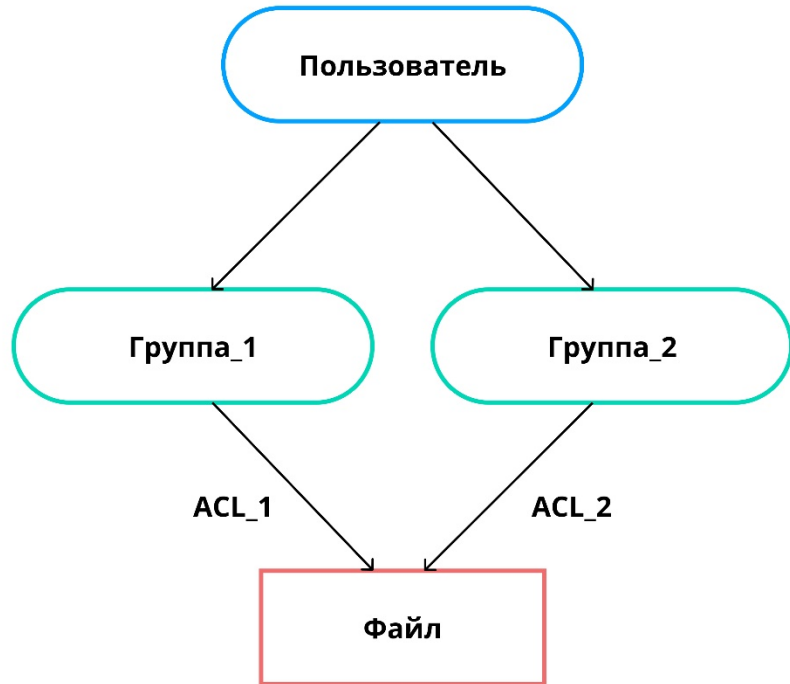
- Конструктивное
- Модифицирующее



# Множественный групповой «шаринг» файла

Типы тестирования:

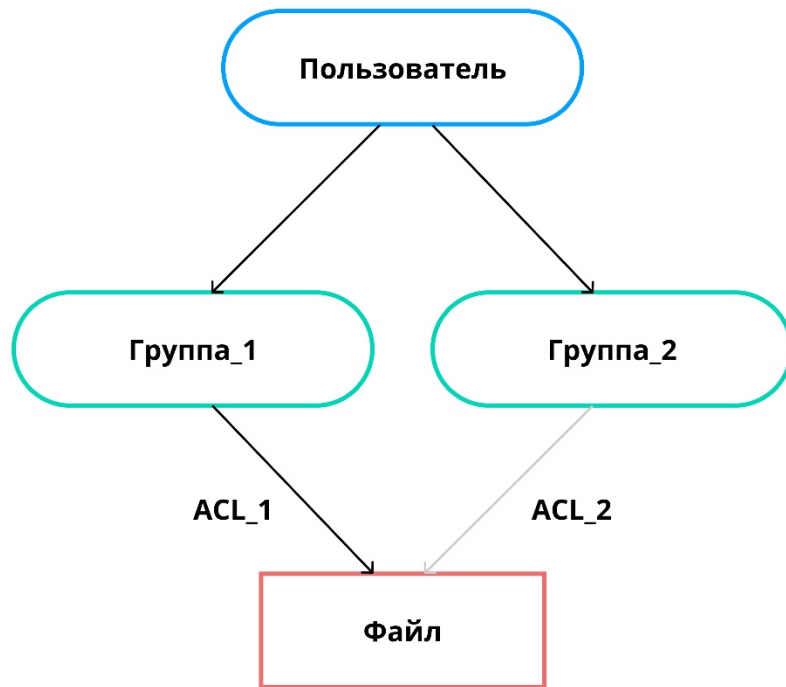
- Конструктивное
- Модифицирующее
- Реконструктивное



# Множественный групповой «шаринг» файла

Типы тестирования:

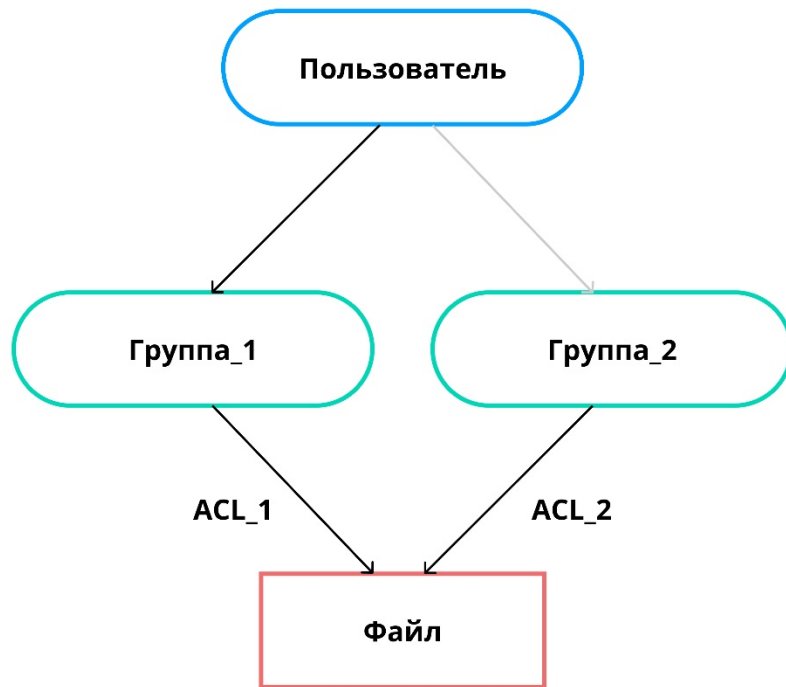
- Конструктивное
- Модифицирующее
- Реконструктивное



# Множественный групповой «шаринг» файла

Типы тестирования:

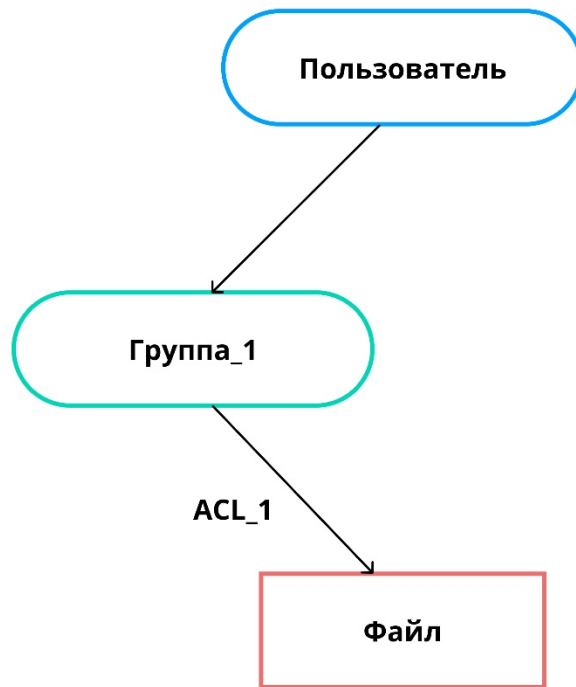
- Конструктивное
- Модифицирующее
- Реконструктивное



# Множественный групповой «шаринг» файла

Типы тестирования:

- Конструктивное
- Модифицирующее
- Реконструктивное



# Функциональное тестирование

# Функциональное тестирование

- Robot Framework



# Функциональное тестирование

- Robot Framework

```
139 ▶ Script_1 - 1_Create group_1 and add user to group_1
    [Documentation] Creating group_1 and adding user into group_1
140
141
142     ${group_name}= Generate Random String    length=${GROUP_NAME_LEN}    chars=[LOWER][NUMBERS]
143
144     ${response}= Create New Group    ${euclid_token}    ${tenant_name}    ${group_name}
145     IF    ${response.status_code} != ${OK_STATUS}
146         Fail    msg=${response.json()}
147     END
148     ${group_1_id}= Set Variable    ${response.json()['id']}
149
150     ${response}= Add User To Group    ${euclid_token}    ${tenant_name}    ${group_1_id}    ${user_id}
151     IF    ${response.status_code} != ${OK_STATUS}
152         Fail    msg=${response.json()}
153     END
154
155     Set Suite Variable    ${group_1_id}
156
157 ▶ Script_1 - 2_Share dir_1 for group_1 with acl=${ACL_MANAGER}
    [Documentation] Sharing dir_1 for group_1 with acl=${ACL_MANAGER}
158
159
160     ${response}= Share Object With Group_id    ${aristotel_token}    ${dir_1_id}    ${ACL_MANAGER}    ${group_1_id}
161     IF    ${response.status_code} != ${OK_STATUS}
162         Fail    msg=${response.json()}
163     END
```



# Функциональное тестирование

- Robot Framework

```
139 ▶ Script_1 - 1_Create group_1 and add user to group_1
    [Documentation] Creating group_1 and adding user into group_1
141
142     ${group_name}= Generate Random String length=${GROUP_NAME_LEN} chars=[LOWER][NUMBERS]
143
144     ${response}= Create New Group ${euclid_token} ${tenant_name} ${group_name}
145     IF ${response.status_code} != ${OK_STATUS}
146         Fail msg=${response.json()}
147     END
148     ${group_1_id}= Set Variable ${response.json()['id']}
149
150     ${response}= Add User To Group ${euclid_token} ${tenant_name} ${group_1_id} ${user_id}
151     IF ${response.status_code} != ${OK_STATUS}
152         Fail msg=${response.json()}
153     END
154
155     Set Suite Variable ${group_1_id}
156
157 ▶ Script_1 - 2_Share dir_1 for group_1 with acl=${ACL_MANAGER}
    [Documentation] Sharing dir_1 for group_1 with acl=${ACL_MANAGER}
159
160     ${response}= Share Object With Group_id ${aristotel_token} ${dir_1_id} ${ACL_MANAGER} ${group_1_id}
161     IF ${response.status_code} != ${OK_STATUS}
162         Fail msg=${response.json()}
163     END
```

# Функциональное тестирование

- Robot Framework

```
139 ▶ Script_1 - 1_Create group_1 and add user to group_1
    [Documentation] Creating group_1 and adding user into group_1
141
142     ${group_name}= Generate Random String length=${GROUP_NAME_LEN} chars=[LOWER][NUMBERS]
143
144     ${response}= Create New Group ${euclid_token} ${tenant_name} ${group_name}
145     IF ${response.status_code} != ${OK_STATUS}
146         Fail msg=${response.json()}
147     END
148     ${group_1_id}= Set Variable ${response.json()['id']}
149
150     ${response}= Add User To Group ${euclid_token} ${tenant_name} ${group_1_id} ${user_id}
151     IF ${response.status_code} != ${OK_STATUS}
152         Fail msg=${response.json()}
153     END
154
155     Set Suite Variable ${group_1_id}
156
157 ▶ Script_1 - 2_Share dir_1 for group_1 with acl=${ACL_MANAGER}
    [Documentation] Sharing dir_1 for group_1 with acl=${ACL_MANAGER}
159
160     ${response}= Share Object With Group_id ${aristotel_token} ${dir_1_id} ${ACL_MANAGER} ${group_1_id}
161     IF ${response.status_code} != ${OK_STATUS}
162         Fail msg=${response.json()}
163     END
```

# Функциональное тестирование

- Robot Framework

```
139 ▶ Script_1 - 1_Create group_1 and add user to group_1
    [Documentation] Creating group_1 and adding user into group_1
141
142     ${group_name}= Generate Random String length=${GROUP_NAME_LEN} chars=[LOWER][NUMBERS]
143
144     ${response}= Create New Group ${euclid_token} ${tenant_name} ${group_name}
145     IF ${response.status_code} != ${OK_STATUS}
146         Fail msg=${response.json()}
147     END
148     ${group_1_id}= Set Variable ${response.json()['id']}
149
150     ${response}= Add User To Group ${euclid_token} ${tenant_name} ${group_1_id} ${user_id}
151     IF ${response.status_code} != ${OK_STATUS}
152         Fail msg=${response.json()}
153     END
154
155     Set Suite Variable ${group_1_id}
156
157 ▶ Script_1 - 2_Share dir_1 for group_1 with acl=${ACL_MANAGER}
    [Documentation] Sharing dir_1 for group_1 with acl=${ACL_MANAGER}
159
160     ${response}= Share Object With Group_id ${aristotel_token} ${dir_1_id} ${ACL_MANAGER} ${group_1_id}
161     IF ${response.status_code} != ${OK_STATUS}
162         Fail msg=${response.json()}
163     END
```

# Функциональное тестирование

- Robot Framework
- CI/CD Integration

```
139 ▶ Script_1 - 1_Create group_1 and add user to group_1
    [Documentation] Creating group_1 and adding user into group_1
141
142     ${group_name}= Generate Random String length=${GROUP_NAME_LEN} chars=[LOWER][NUMBERS]
143
144     ${response}= Create New Group ${euclid_token} ${tenant_name} ${group_name}
145     IF ${response.status_code} != ${OK_STATUS}
146         Fail msg=${response.json()}
147     END
148     ${group_1_id}= Set Variable ${response.json()['id']}
149
150     ${response}= Add User To Group ${euclid_token} ${tenant_name} ${group_1_id} ${user_id}
151     IF ${response.status_code} != ${OK_STATUS}
152         Fail msg=${response.json()}
153     END
154
155     Set Suite Variable ${group_1_id}
156
157 ▶ Script_1 - 2_Share dir_1 for group_1 with acl=${ACL_MANAGER}
    [Documentation] Sharing dir_1 for group_1 with acl=${ACL_MANAGER}
159
160     ${response}= Share Object With Group_id ${aristotel_token} ${dir_1_id} ${ACL_MANAGER} ${group_1_id}
161     IF ${response.status_code} != ${OK_STATUS}
162         Fail msg=${response.json()}
163     END
```

# Функциональное тестирование

- Robot Framework
- CI/CD Integration
- Arango Functions

```
7 from typing import Any
8 from arango import ArangoClient
9 from arango.client import StandardDatabase
10
11
12 class QueryTestModel:
13     ROBOT_LIBRARY_SCOPE = 'GLOBAL'
14
15     def __init__(self, **kwargs):
16         self.client: ArangoClient
17         self.db: StandardDatabase
18         self.client, self.db = self.create_connection(**kwargs)
19
20     @staticmethod
21     def create_connection(**kwargs) -> tuple[ArangoClient, Any]:
22         arango_host = kwargs["arango_host"]
23         arango_username = kwargs["arango_username"]
24         arango_password = kwargs["arango_password"]
25
26         client = ArangoClient(hosts='http://' + arango_host)
27         db = client.db('pgs', username=arango_username, password=arango_password)
28         return client, db
29
30     def remove_creator(self, file_id):
31         query = """
32         with CreatedByUser, StoredObjects
33         let edge_to_remove = (for v, e in 1..1
34             inbound 'StoredObjects/{file_id}' CreatedByUser
35             return e._key)
36         remove edge_to_remove[0] in CreatedByUser
37         """
38         self.db.aql.execute(query.format({'file_id': file_id}))
39         return query
40
```



# Функциональное тестирование

- Robot Framework
- CI/CD Integration
- Arango Functions

```
7 from typing import Any
8 from arango import ArangoClient
9 from arango.client import StandardDatabase
10
11
12 class QueryTestModel:
13     ROBOT_LIBRARY_SCOPE = 'GLOBAL'
14
15     def __init__(self, **kwargs):
16         self.client: ArangoClient
17         self.db: StandardDatabase
18         self.client, self.db = self.create_connection(**kwargs)
19
20     @staticmethod
21     def create_connection(**kwargs) -> tuple[ArangoClient, Any]:
22         arango_host = kwargs["arango_host"]
23         arango_username = kwargs["arango_username"]
24         arango_password = kwargs["arango_password"]
25
26         client = ArangoClient(hosts='http://' + arango_host)
27         db = client.db('pgs', username=arango_username, password=arango_password)
28         return client, db
29
30     def remove_creator(self, file_id):
31         query = """
32         with CreatedByUser, StoredObjects
33         let edge_to_remove = (for v, e in 1..1
34             inbound 'StoredObjects/{file_id}' CreatedByUser
35             return e._key)
36         remove edge_to_remove[0] in CreatedByUser
37         """
38         self.db.aql.execute(query.format({'file_id': file_id}))
39         return query
40
```

# Функциональное тестирование

- Robot Framework
- CI/CD Integration
- Arango Functions

```
7 from typing import Any
8 from arango import ArangoClient
9 from arango.client import StandardDatabase
10
11
12 class QueryTestModel:
13     ROBOT_LIBRARY_SCOPE = 'GLOBAL'
14
15     def __init__(self, **kwargs):
16         self.client: ArangoClient
17         self.db: StandardDatabase
18         self.client, self.db = self.create_connection(**kwargs)
19
20     @staticmethod
21     def create_connection(**kwargs) -> tuple[ArangoClient, Any]:
22         arango_host = kwargs["arango_host"]
23         arango_username = kwargs["arango_username"]
24         arango_password = kwargs["arango_password"]
25
26         client = ArangoClient(hosts='http://' + arango_host)
27         db = client.db('pgs', username=arango_username, password=arango_password)
28         return client, db
29
30     def remove_creator(self, file_id):
31         query = """
32         with CreatedByUser, StoredObjects
33         let edge_to_remove = (for v, e in 1..1
34             inbound 'StoredObjects/{file_id}' CreatedByUser
35             return e._key)
36         remove edge_to_remove[0] in CreatedByUser
37         """
38         self.db.aql.execute(query.format({'file_id': file_id}))
39         return query
40
```



# Нагрузочное тестирование

# Нагрузочное тестирование

- Gatling

# Нагрузочное тестирование

- Gatling

```
23  setUp(  
24      scenarios.scenarioForOnlyFsLogin.inject(  
25          rampUsers(AccountParams.accountsSize) during (FsTestParams.loginTimeSec seconds)  
26      ).andThen(  
27          scenarios.fsListShares.inject(  
28              constantUsersPerSec(FsTestParams.rpsListShares) during (FsTestParams.staySec seconds)  
29          )  
30      ),  
31  ).protocols(httpConf)  
32  }  
33
```

# Нагрузочное тестирование

- Gatling

```
23  setUp(  
24      scenarios.scenarioForOnlyFsLogin.inject(  
25          rampUsers(AccountParams.accountsSize) during (FsTestParams.loginTimeSec seconds)  
26      ).andThen(  
27          scenarios.fsListShares.inject(  
28              constantUsersPerSec(FsTestParams.rpsListShares) during (FsTestParams.staySec seconds)  
29          )  
30      ),  
31  ).protocols(httpConf)  
32  }  
33
```

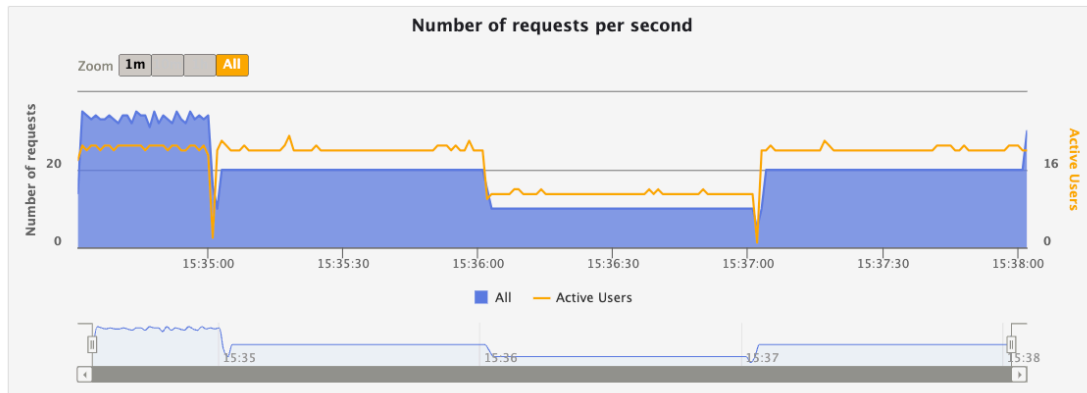
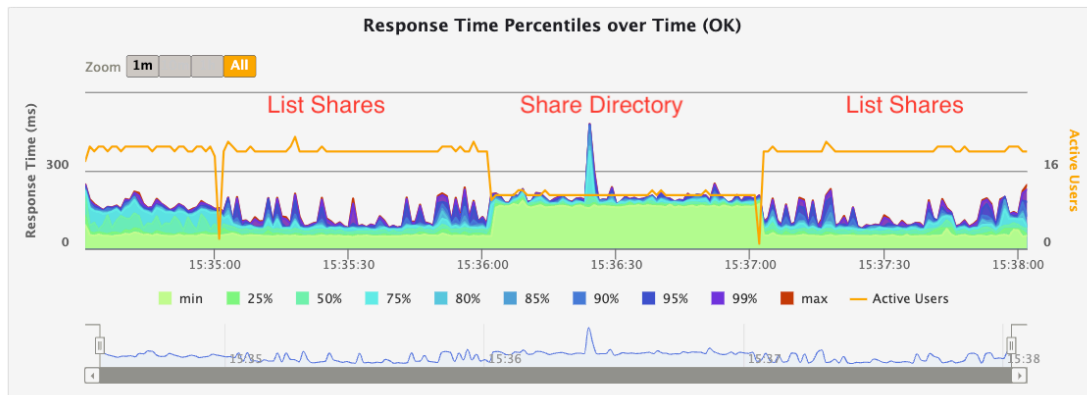
# Нагрузочное тестирование

- Gatling

```
23  setUp(  
24      scenarios.scenarioForOnlyFsLogin.inject(  
25          rampUsers(AccountParams.accountsSize) during (FsTestParams.loginTimeSec seconds)  
26      ).andThen(  
27          scenarios.fsListShares.inject(  
28              constantUsersPerSec(FsTestParams.rpsListShares) during (FsTestParams.staySec seconds)  
29          )  
30      ),  
31  ).protocols(httpConf)  
32  }  
33
```

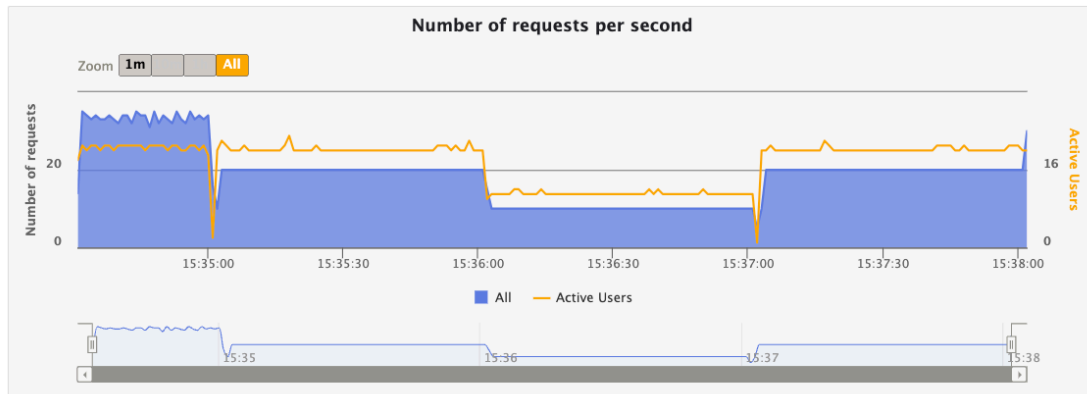
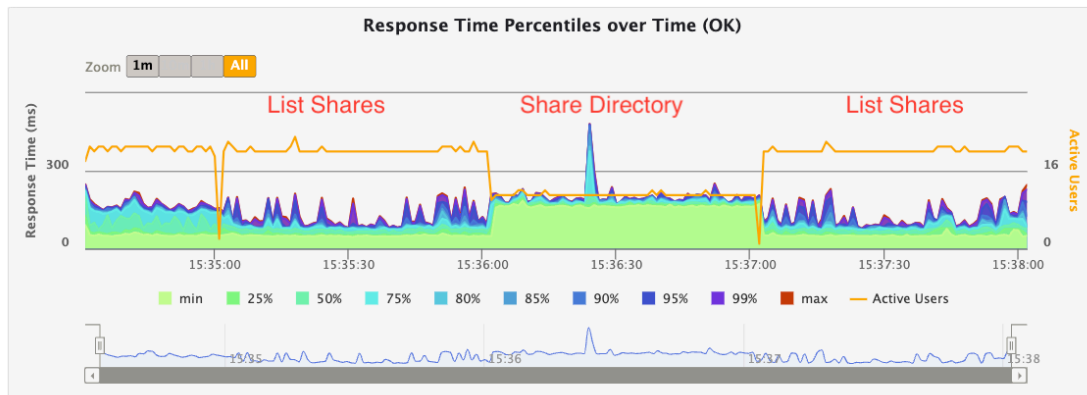
# Нагрузочное тестирование

- Gatling
- (Не)Инвазивное тестирование



# Нагрузочное тестирование

- Gatling
- (Не)Инвазивное тестирование
- Разносторонняя параметризация



# Нагрузочное тестирование

- Gatling
- (Не)Инвазивное тестирование
- Разносторонняя параметризация
- Детальный мониторинг





# Примеры найденных багов функционала «шаринга»



# Менеджер выдает право на владение

# Менеджер выдает право на владение

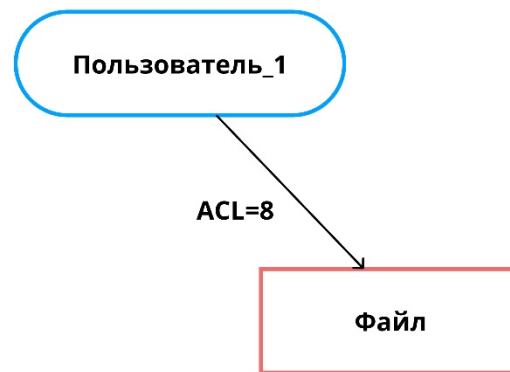
- Есть файл (с владельцем)

# Менеджер выдает право на владение

- Есть файл (с владельцем)
- Пользователю\_1 предоставлено право на управление файлом

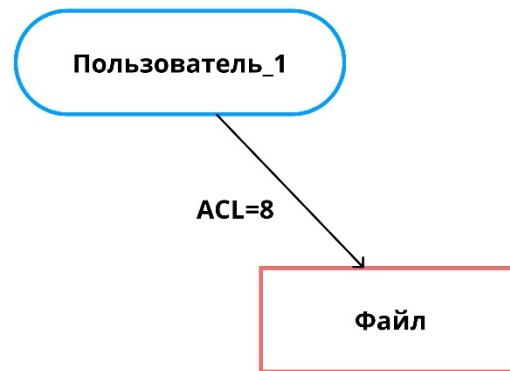
# Менеджер выдает право на владение

- Есть файл (с владельцем)
- Пользователю\_1 предоставлено право на управление файлом



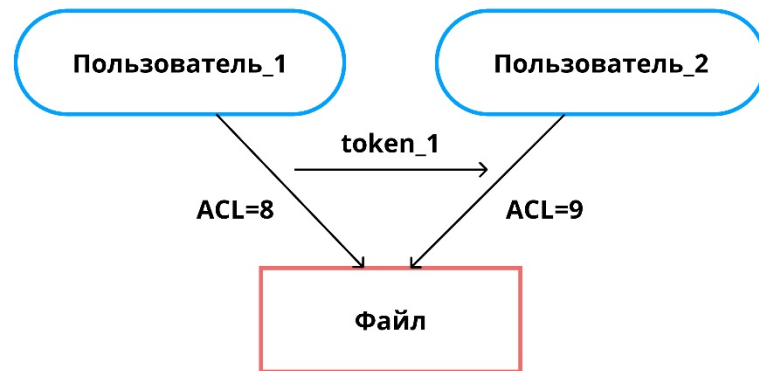
# Менеджер выдает право на владение

- Есть файл (с владельцем)
- Пользователю\_1 предоставлено право на управление файлом
- Пользователь\_1 выдает пользователю\_2 право на владение файлом



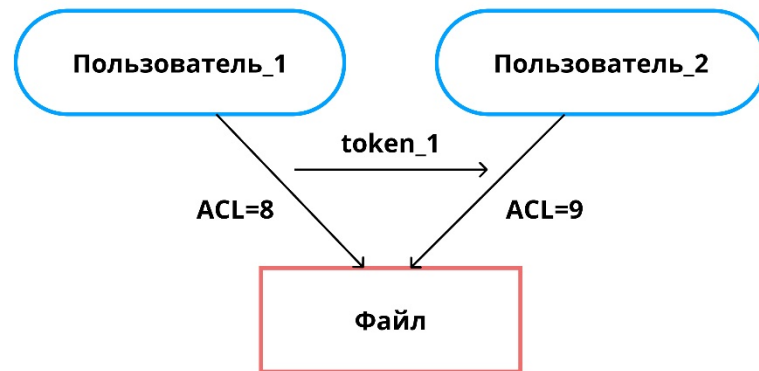
# Менеджер выдает право на владение

- Есть файл (с владельцем)
- Пользователю\_1 предоставлено право на управление файлом
- Пользователь\_1 выдает пользователю\_2 право на владение файлом



# Менеджер выдает право на владение

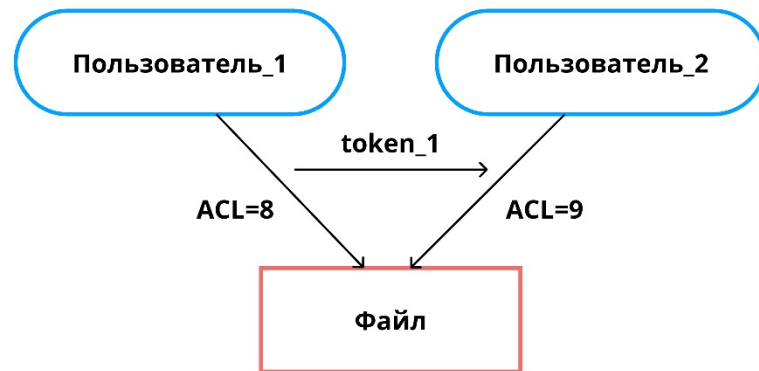
- Есть файл (с владельцем)
- Пользователю\_1 предоставлено право на управление файлом
- Пользователь\_1 выдает пользователю\_2 право на владение файлом
- Предоставляем право большее, чем имеем





# Менеджер выдает право на владение

- Есть файл (с владельцем)
- Пользователю\_1 предоставлено право на управление файлом
- Пользователь\_1 выдает пользователю\_2 право на владение файлом
- Предоставляем право большее, чем имеем
- У файла — несколько владельцев



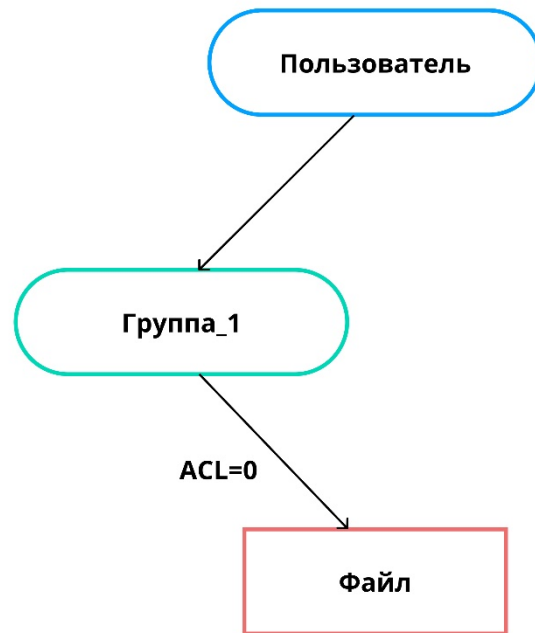
# Отмена запрета на доступ к файлу

# Отмена запрета на доступ к файлу

- Запрет доступа к файлу для группы

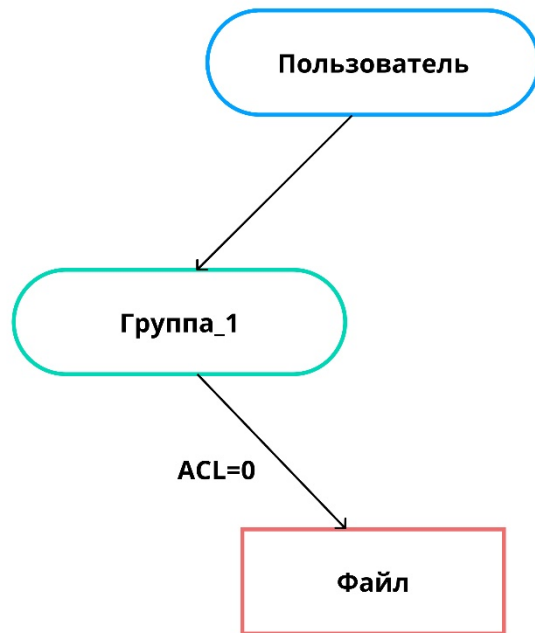
# Отмена запрета на доступ к файлу

- Запрет доступа к файлу для группы



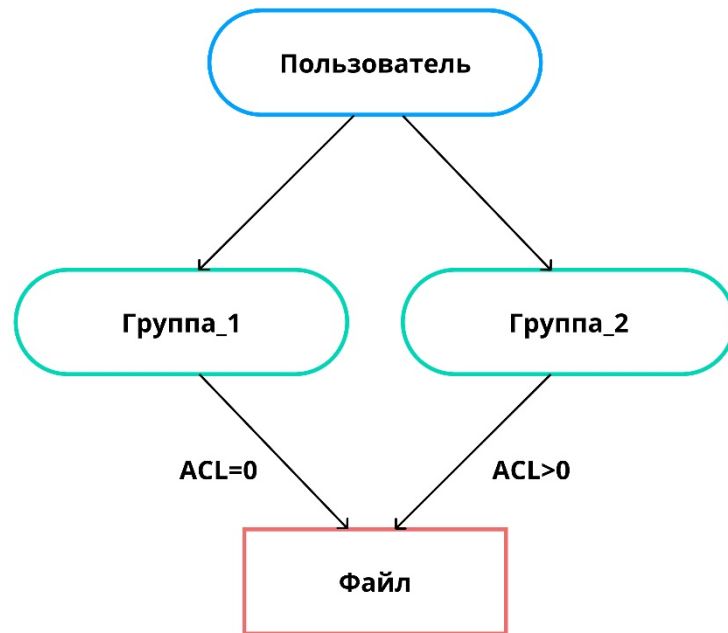
# Отмена запрета на доступ к файлу

- Запрет доступа к файлу для группы
- Предоставление доступа к файлу для группы (acl > 0 )



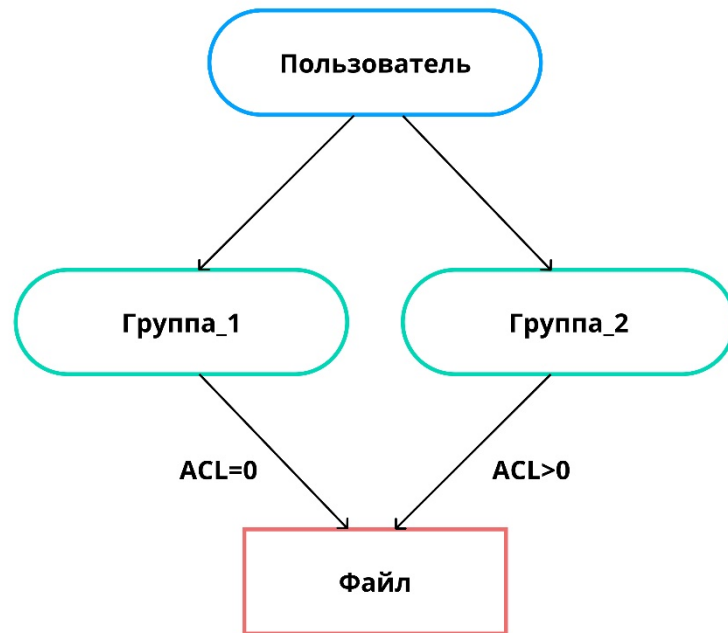
# Отмена запрета на доступ к файлу

- Запрет доступа к файлу для группы
- Предоставление доступа к файлу для группы (acl >0 )



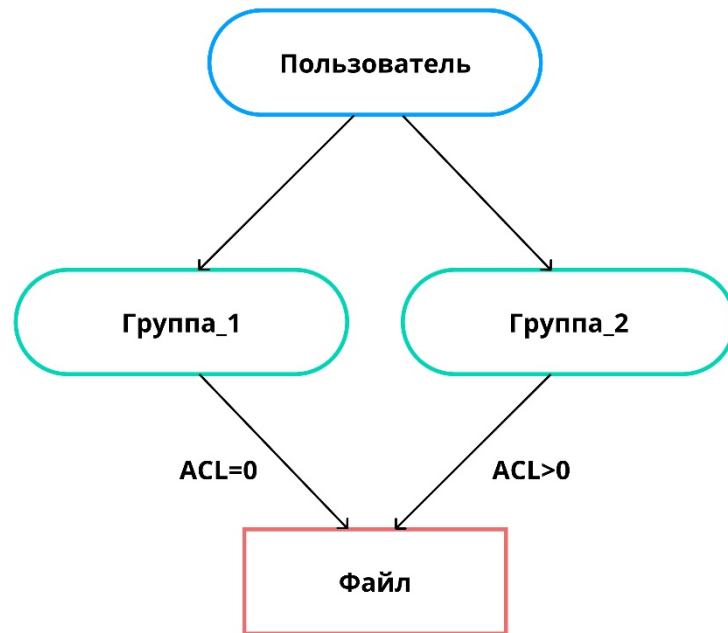
# Отмена запрета на доступ к файлу

- Запрет доступа к файлу для группы
- Предоставление доступа к файлу для группы (acl >0 )
- Право доступа перекрывает запрет



# Отмена запрета на доступ к файлу

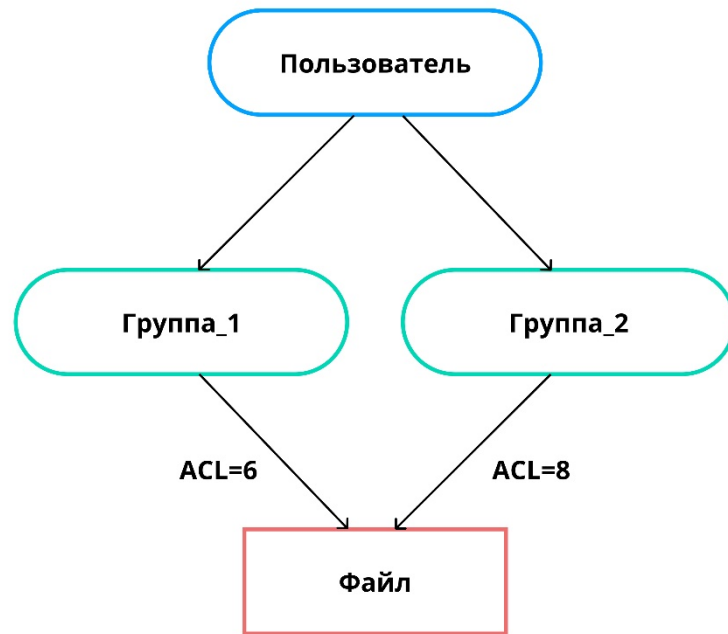
- Запрет доступа к файлу для группы
- Предоставление доступа к файлу для группы (acl >0 )
- Право доступа перекрывает запрет
- Пользователь получает доступ к запрещенным файлам





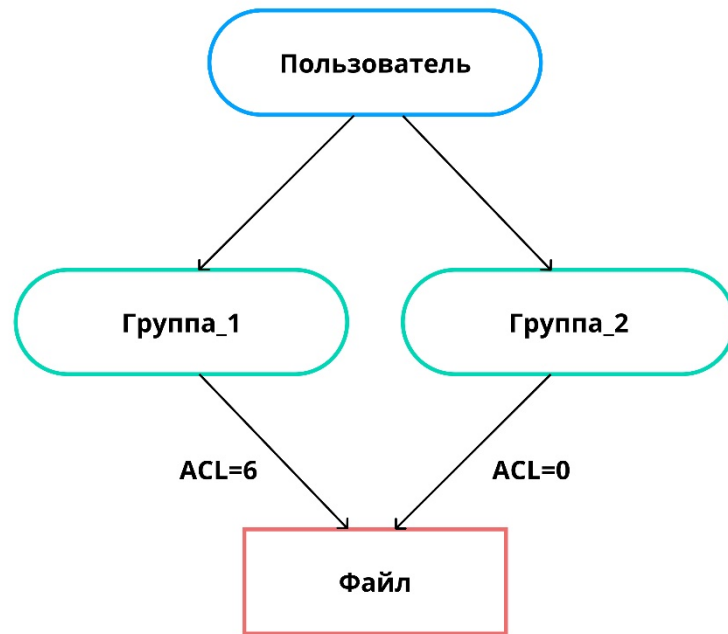
# Отмена запрета на доступ к файлу

- Запрет доступа к файлу для группы
- Предоставление доступа к файлу для группы (acl > 0 )
- Право доступа перекрывает запрет
- Пользователь получает доступ к запрещенным файлам



# Отмена запрета на доступ к файлу

- Запрет доступа к файлу для группы
- Предоставление доступа к файлу для группы (acl > 0 )
- Право доступа перекрывает запрет
- Пользователь получает доступ к запрещенным файлам



# Плюсы использования ArangoDB для Dev



# Документация

# Документация

The screenshot shows the ArangoDB documentation page for 'What is ArangoDB?'. The page title is '3.11.8 > About ArangoDB'. The main heading is 'What is ArangoDB?'. Below the heading, the text reads: 'ArangoDB is a scalable graph database system to drive value from connected data, faster'. To the right of the main content, there is a 'CONTRIBUTE' link and a section titled 'On this page' with links to 'What are Graphs?' and 'Beyond Graphs'. The left sidebar contains a navigation menu with items: 'About ArangoDB', 'Use Cases', 'Features', 'Get Started', 'Concepts', 'ArangoGraph', 'AQL', 'Graphs', 'Data Science', 'Index and Search', 'Components', 'Deploy', 'Operations', 'Develop', and 'Release Notes'. The main content area features a diagram illustrating the ArangoDB architecture. The diagram is a circular flow starting from a user icon on the left, pointing to a central 'Scalable Graph Technology' hub. This hub is surrounded by six data models: 'Document Data Model', 'Key Value Data Model', 'Graph Data Model', 'Full-Text ArangoSearch', 'Iterative Graph Processing Pregel', and 'ArangoML'. The hub is also connected to 'ArangoGraph Graph data and analytics platform' and 'Kube-Arango Kubernetes integration'. To the right of the diagram, a box titled 'Unified Engine and Queries' lists three benefits: 'Single Source of Truth', 'Developer Knowledge', and 'Less Maintenance', and 'Faster Performance'. An 'AQL' arrow points from the user icon to the central hub.

ArangoDB

3.11.8 > About ArangoDB

## What is ArangoDB?

ArangoDB is a scalable graph database system to drive value from connected data, faster

CONTRIBUTE

On this page

- What are Graphs?
- Beyond Graphs

Unified Engine and Queries

- Single Source of Truth
- Developer Knowledge
- Less Maintenance
- Faster Performance



# Синтаксис

# Синтаксис

Editor Running Queries Slow Query History

☆ Queries New Save as 1000 results

```
1 WITH Dirs, Subdirs
2 let dir = (FOR d IN Dirs
3   FILTER d.name == @name
4   return d)[0]
5 FOR v IN 1..1000
6   OUTBOUND dir Subdirs
7   return v.name
```

Key	Value	JSON
name	Anna	

Create Debug Package Profile Explain Execute



# Движок RocksDB

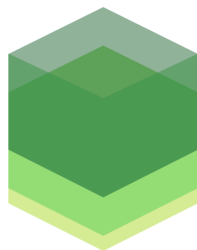




# Движок RocksDB



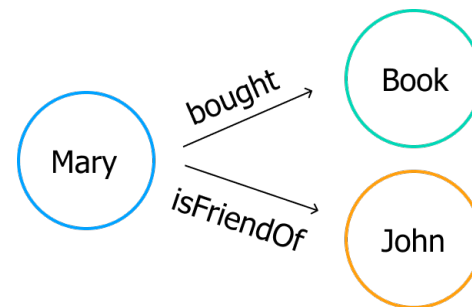
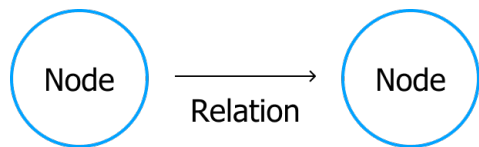
# Движок RocksDB



**LEVELDB**

# Графовая структура

# Графовая структура



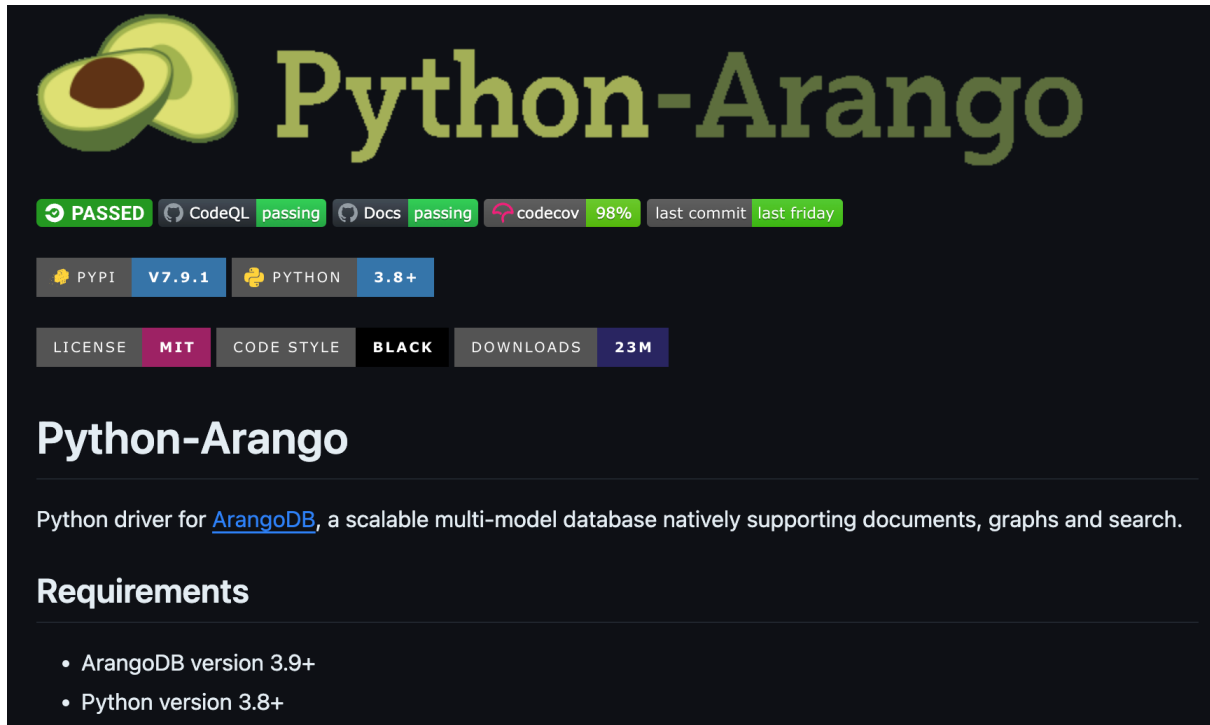
# Минусы использования ArangoDB для Dev



# Библиотеки



# Библиотеки



The image shows a dark-themed banner for the Python-Arango project. At the top left is the logo, which consists of three slices of an avocado. To the right of the logo, the text "Python-Arango" is written in a large, light green, sans-serif font. Below the logo and title, there is a row of status indicators: a green circle with a checkmark and the word "PASSED", a CodeQL icon with the word "passing", a Docs icon with the word "passing", a Codecov icon with "98%", and "last commit" with "last friday". Below this row are two buttons: one with a PYPI icon and the text "v7.9.1", and another with a PYTHON icon and the text "3.8+". Below these are four more buttons: "LICENSE" with "MIT" in a pink box, "CODE STYLE" with "BLACK" in a black box, "DOWNLOADS" with "23M" in a blue box, and a partially visible button. Below the buttons, the text "Python-Arango" is written in a white, sans-serif font. Below this, a paragraph of text reads: "Python driver for [ArangoDB](#), a scalable multi-model database natively supporting documents, graphs and search." Below the paragraph, the text "Requirements" is written in a white, sans-serif font. Below "Requirements" is a bulleted list with two items: "ArangoDB version 3.9+" and "Python version 3.8+".

## Python-Arango

Python driver for [ArangoDB](#), a scalable multi-model database natively supporting documents, graphs and search.

### Requirements

- ArangoDB version 3.9+
- Python version 3.8+



# Слабое русскоязычное сообщество



# Слабое русскоязычное сообщество

Хабр



КАК СТАТЬ АВТОРОМ

Моя лента

Все потоки

Разработка

Администрирование

Дизайн

Менеджмент

Маркетинг

Научпоп

Arango



ПУБЛИКАЦИИ

ХАБЫ

КОМПАНИИ

ПОЛЬЗОВАТЕЛИ

КОММЕНТАРИИ

по релевантности



найдено 13 публикаций



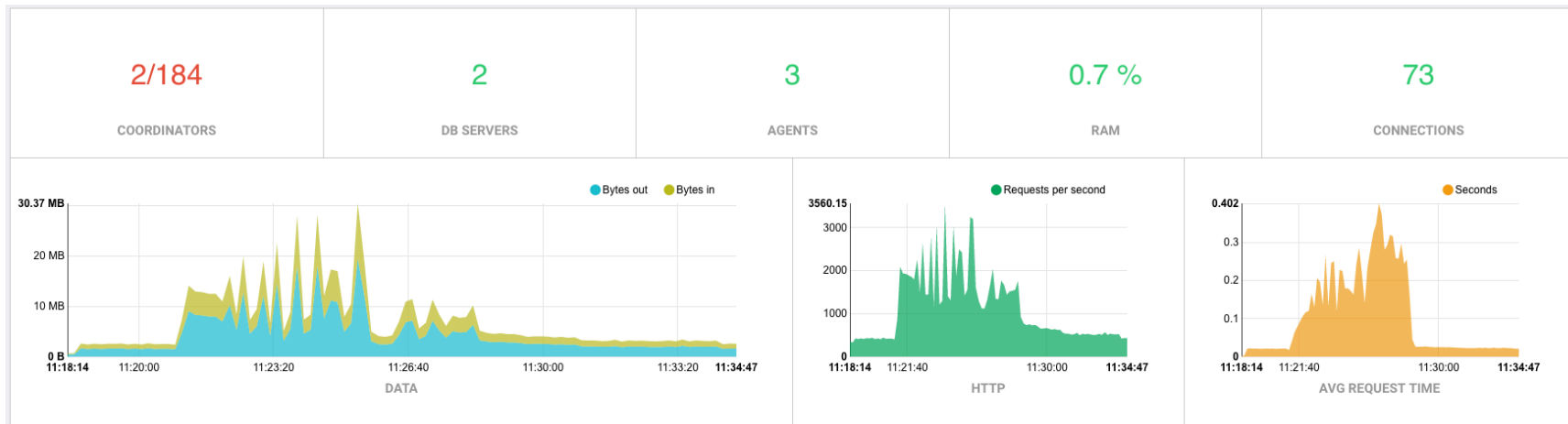
МойОфис

# Плюсы использования ArangoDB для QA

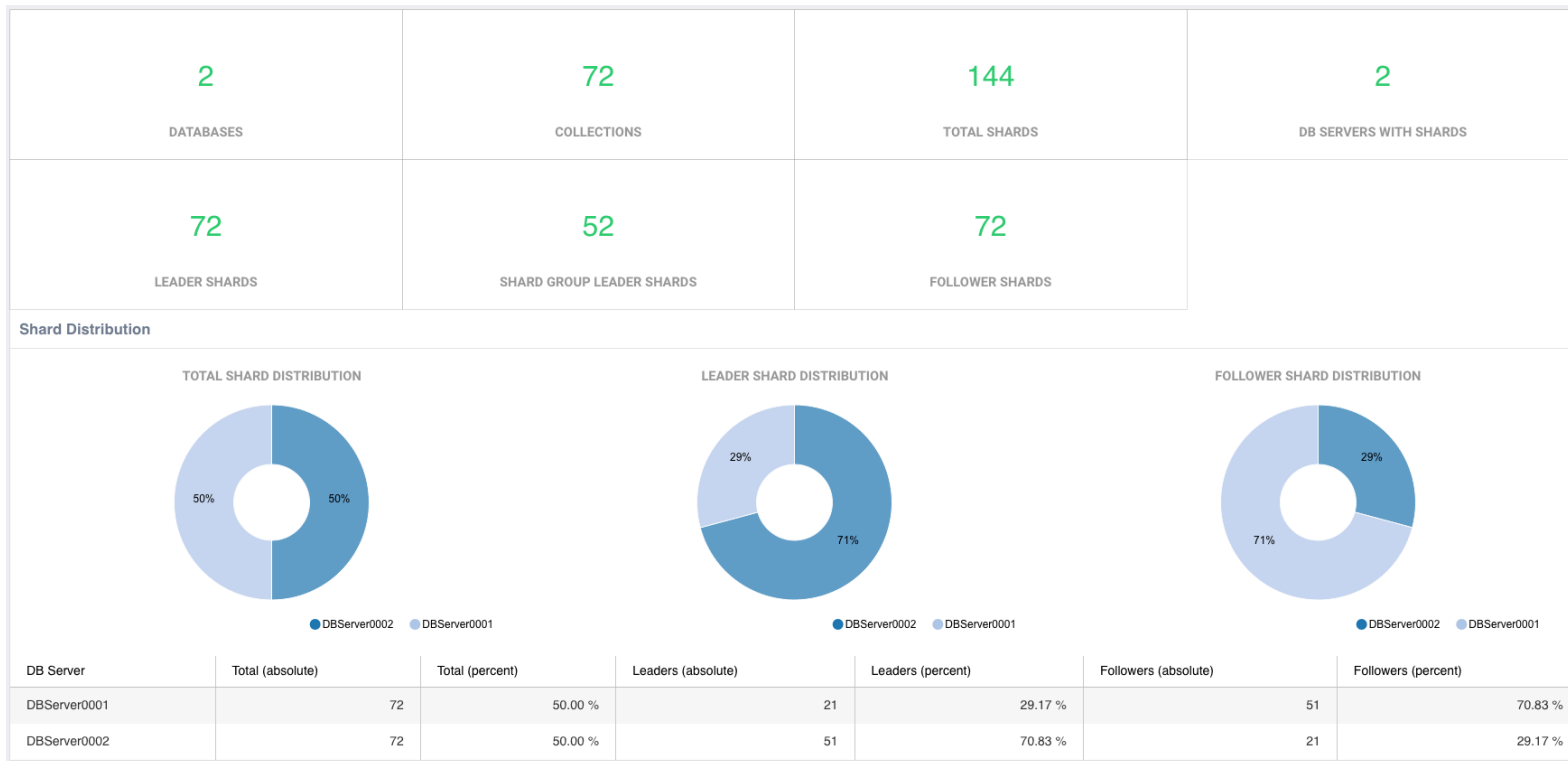


# Web-интерфейс ArangoDB

# Web-интерфейс ArangoDB



# Web-интерфейс ArangoDB



# Web-интерфейс ArangoDB

The screenshot displays the ArangoDB web interface. On the left is a dark sidebar with navigation options: COLLECTIONS, VIEWS, QUERIES, GRAPHS, SERVICES, LOGS, SUPPORT, and HELP US. The main area shows a grid of collection cards. Each card has an icon (a share symbol or a document symbol) and a label with a 'loaded' status indicator. A search bar is located in the top right corner of the main area.

Collection Name	Status
+ Add Collection	
Acl	loaded
Avatars	loaded
CreatedByUser	loaded
DomainInfo	loaded
EventHistory	loaded
Flagged	loaded
GroupedCtypes	loaded
GroupMember	loaded
Groups	loaded
IndexOrPrerender	loaded
NotificationRead	loaded
Notifications	loaded
NotificationsInfo	loaded
ProhibitionCtypes	loaded
ProhibitionOrException	loaded
PromoCodes	loaded
Promocodes	loaded
PublicLink	loaded
RemovedObjects	loaded
StoredObjects	loaded
Subdir	loaded
Subscriptions	loaded
TenantInfo	loaded
TouchedByUser	loaded
Version	loaded

# Web-интерфейс ArangoDB

The screenshot displays the ArangoDB web interface's 'Services Store'. On the left is a dark sidebar with navigation options: DASHBOARD, COLLECTIONS, VIEWS, QUERIES, GRAPHS, SERVICES (highlighted), and USERS. The main content area has a top navigation bar with 'Store', 'Upload', 'New', 'GitHub', and 'Remote'. Below this, seven service cards are arranged in a grid. Each card features an icon, a name, a category, and an 'Install' button. The services shown are:

- demo-geo-s2** (geo,service)
- demo-graphql** (demo,service,graphql)
- grafana-connector** (connector,service)
- itzpapatl** (service,demo)
- prometheus-exporter** (prometheus,monitoring)
- qlik-connector** (connector,service)
- tableau-connector** (connector,service)

# AQL запросы



# AQL запросы

```
1 FOR doc IN StoredObjects
2   COLLECT WITH COUNT INTO length
3   RETURN length
```

Query

1 elements

32.934 ms

1 ▾ [

2 250186

3 ]]



# AQL запросы

```
1 FOR doc IN StoredObjects
2   COLLECT WITH COUNT INTO length
3   RETURN length
```

Query 1 elements 32.934 ms

```
1 [
2   250186
3 ]
```

```
1 RETURN LENGTH(Acl)
2
```

Query 1 elements 0.537 ms

```
1 [
2   32
3 ]
```

# AQL запросы

Explain

```
1 Query String (77 chars, cacheable: true):
2 FOR doc IN StoredObjects
3   COLLECT WITH COUNT INTO length
4   RETURN length
5
6 Execution plan:
7 Id  NodeType  Site  Est.  Comment
8 1  SingletonNode  DBS    1    * ROOT
9 9  IndexNode    DBS  1516628  - FOR doc IN StoredObjects /* primary index scan, scan only, 1 shard(s) */
10 7  RemoteNode   COOR 1516628  - REMOTE
11 8  GatherNode   COOR 1516628  - GATHER /* unsorted */
12 3  CollectNode  COOR    1    - COLLECT AGGREGATE length = LENGTH() /* count */
13 4  ReturnNode   COOR    1    - RETURN length
14
15 Indexes used:
16 By  Name  Type  Collection  Unique  Sparse  Selectivity  Fields  Ranges
17 9   primary  primary  StoredObjects  true   false   100.00 %  [ `'_key`' ]  *
18
19 Optimization rules applied:
20 Id  RuleName
21 1   scatter-in-cluster
22 2   remove-unnecessary-remote-scatter
23 3   reduce-extraction-to-projection
24
25 Optimization rules with highest execution times:
26 RuleName  Duration [s]
27 scatter-in-cluster  0.00001
28 reduce-extraction-to-projection  0.00001
29 restrict-to-single-shard  0.00001
30 remove-unnecessary-remote-scatter  0.00001
31 use-indexes  0.00001
32
33 52 rule(s) executed, 1 plan(s) created
34
35
```

Query 1 elements 309.110 ms

```
1 - [
2   1516628
3 ]
```



# Инструмент тестирования — arangobench

# Инструмент тестирования — arangobench

## arangobench

`arangobench` is a benchmark and test tool that can be used to issue test requests to the database system for performance and server function testing

*arangobench* is a client tool which makes network connections to an ArangoDB server in about the same way as a client application would do via an ArangoDB client driver. It thus often provides good enough throughput and performance estimates. It provides different test cases that can be executed, that reflect a broader set of use cases. It is useful to pick and run the test cases that most closely resemble typical or expected workloads. It supports parallel querying and batch requests.

# Инструмент тестирования — arangobench

## arangobench

`arangobench` is a benchmark and test tool that can be used to issue test requests to the database system for performance and server function testing

*arangobench* is a client tool which makes network connections to an ArangoDB server in about the same way as a client application would do via an ArangoDB client driver. It thus often provides good enough throughput and performance estimates. It provides different test cases that can be executed, that reflect a broader set of use cases. It is useful to pick and run the test cases that most closely resemble typical or expected workloads. It supports parallel querying and batch requests.

Run the `version` test case with 1000 requests, without threads:

```
arangobench --test-case version --requests 1000 --threads 1
```

# Инструмент тестирования — arangobench

Run the `document` test case with 2000 requests, with threads 2, with async requests:

```
arangobench --test-case document --requests 1000 --threads 2 --async true
```

Run the `document` test case with 2000 requests, with threads 2, using batch requests:

```
arangobench --test-case document --requests 1000 --threads 2 --batch-size  
10
```

# Инструмент тестирования — arangobench

```
/ # arangobench --server.username root --server.password RjirbyfKfgf --test-case document --complexity 2 --async --requests 5000
```

```
starting threads...
```

```
executing tests...
```

```
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 250
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 500
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 750
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 1000
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 1250
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 1500
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 1750
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 2000
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 2250
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 2500
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 2750
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 3000
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 3250
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 3500
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 3750
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 4000
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 4250
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 4500
2024-02-27T13:17:22Z [308] INFO [c3604] {general} number of operations: 4750
```

```
Total number of operations: 5000, runs: 1, keep alive: yes, async: yes, batch size: 0, replication factor: 1, number of shards: 1, wait for sync: false, concurrency level (threads): 1
```

```
Test case: document, complexity: 2, database: '_system', collection: 'ArangoBenchmark'
```

```
Total request/response duration (sum of all threads): 0.295421 s
```

```
Request/response duration (per thread): 0.295421 s
```

```
Time needed per operation: 0.000061 s
```

```
Time needed per operation per thread: 0.000061 s
```

```
Operations per second rate: 16361.796531
```

```
Elapsed time since start: 0.305590 s
```

```
Min Request time: 0.037432ms
```

```
Avg Request time: 0.059072ms
```

```
Max Request time: 1.204967ms
```

```
Interval/Percentile:      50.00%      80.00%      85.00%      90.00%      95.00%      99.00%
0.00449700ms      0.04950000ms  0.06300000ms  0.07640000ms  0.10340000ms  0.12590000ms  0.16190000ms
```



# Минусы использования ArangoDB для QA



# Зависимость производительности от размера БД

# Зависимость производительности от размера БД

- Тесты могут добавлять новые объекты в БД

# Зависимость производительности от размера БД

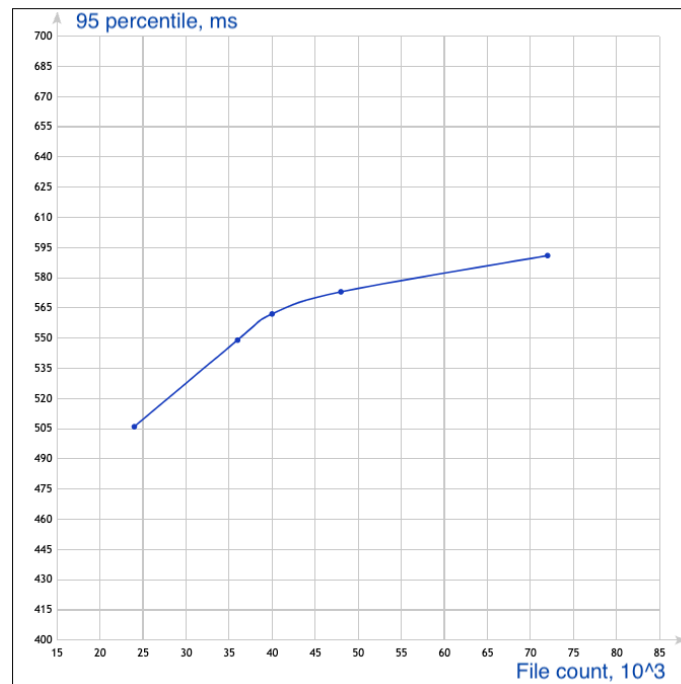
- Тесты могут добавлять новые объекты в БД
- Тесты надо разделять на инвазивные и неинвазивные

# Зависимость производительности от размера БД

- Тесты могут добавлять новые объекты в БД
- Тесты надо разделять на инвазивные и неинвазивные
- Метрики и методики разных тестов различны

# Зависимость производительности от размера БД

- Тесты могут добавлять новые объекты в БД
- Тесты надо разделять на инвазивные и неинвазивные
- Метрики и методики разных тестов различны



# Бэкапирование и восстановление из дампов

# Бэкапирование и восстановление из дампов

- Восстановление из дампа — возвращаем исходное состояние БД

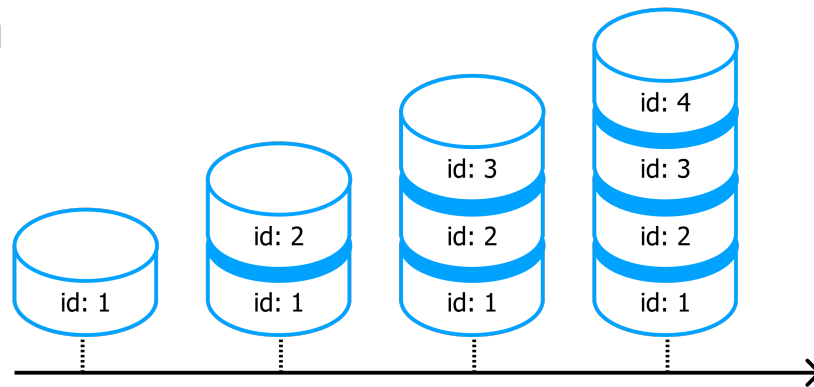


# Бэкапирование и восстановление из дампов

- Восстановление из дампа — возвращаем исходное состояние БД
- Снимаем дампы на различных уровнях наполненности БД

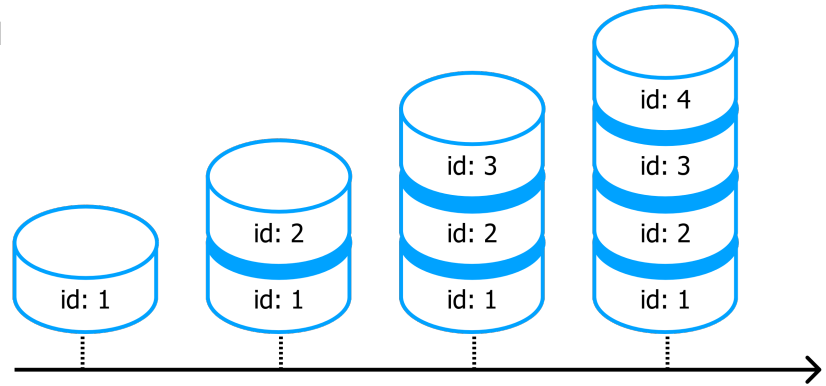
# Бэкапирование и восстановление из дампов

- Восстановление из дампа — возвращаем исходное состояние БД
- Снимаем дампы на различных уровнях наполненности БД



# Бэкапирование и восстановление из дампов

- Восстановление из дампа — возвращаем исходное состояние БД
- Снимаем дампы на различных уровнях наполненности БД
- Скрипты должны быть автоматизированными

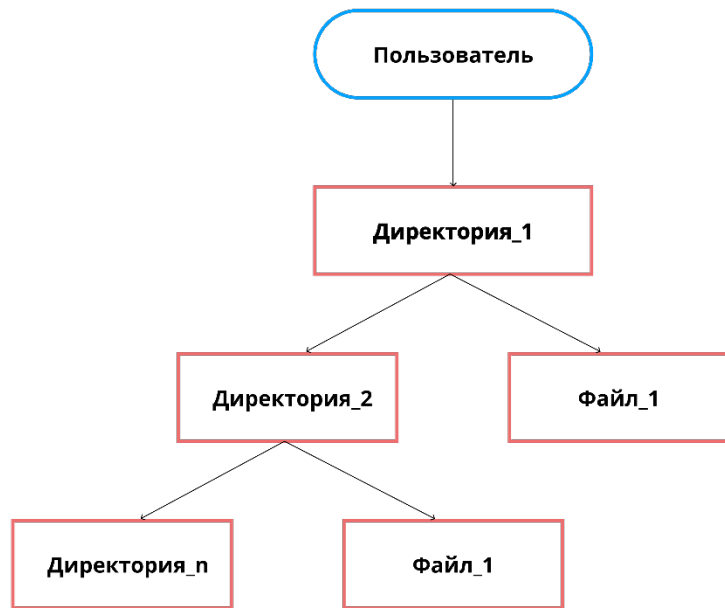


# Пример реализации графовой структуры в реляционных БД

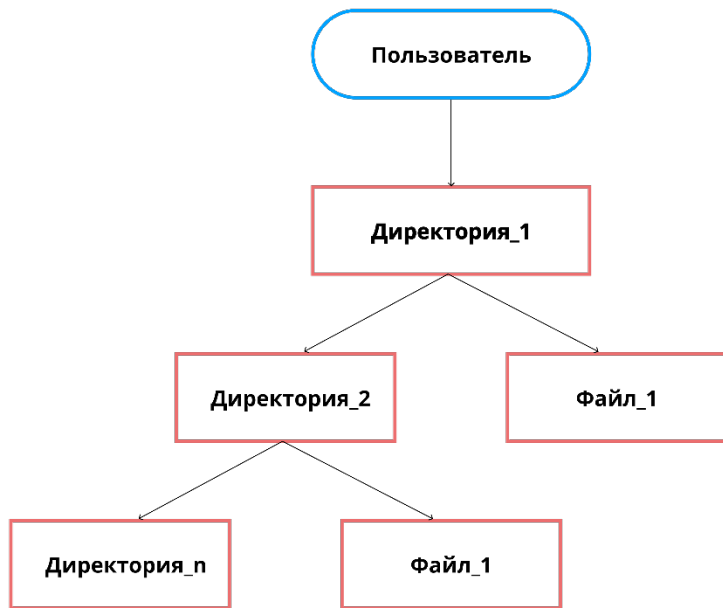


# Аранго запрос на обход графов

# Arango запрос на обход графов

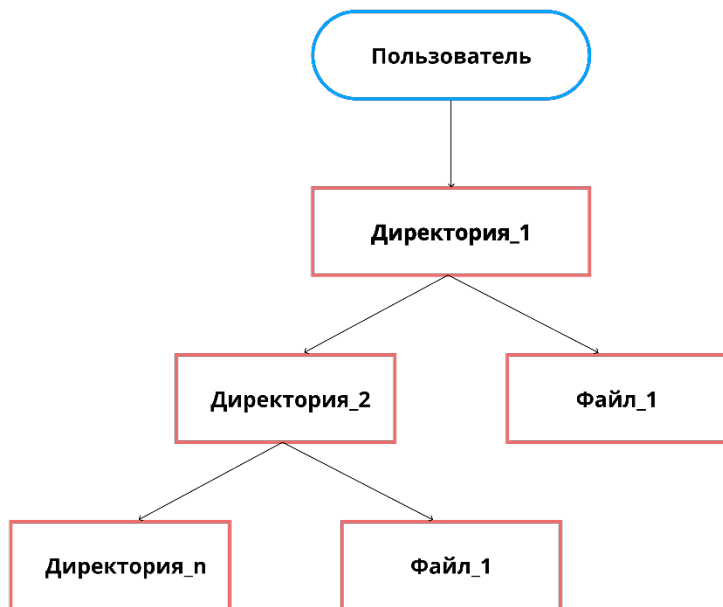


# Arango запрос на обход графов



```
WITH Dirs, Subdirs
let dir = (FOR d IN Dirs
FILTER d.name == @name
return d)[0]
FOR v IN 1..1000
OUTBOUND dir Subdirs
return v.name
```

# Arango запрос на обход графов

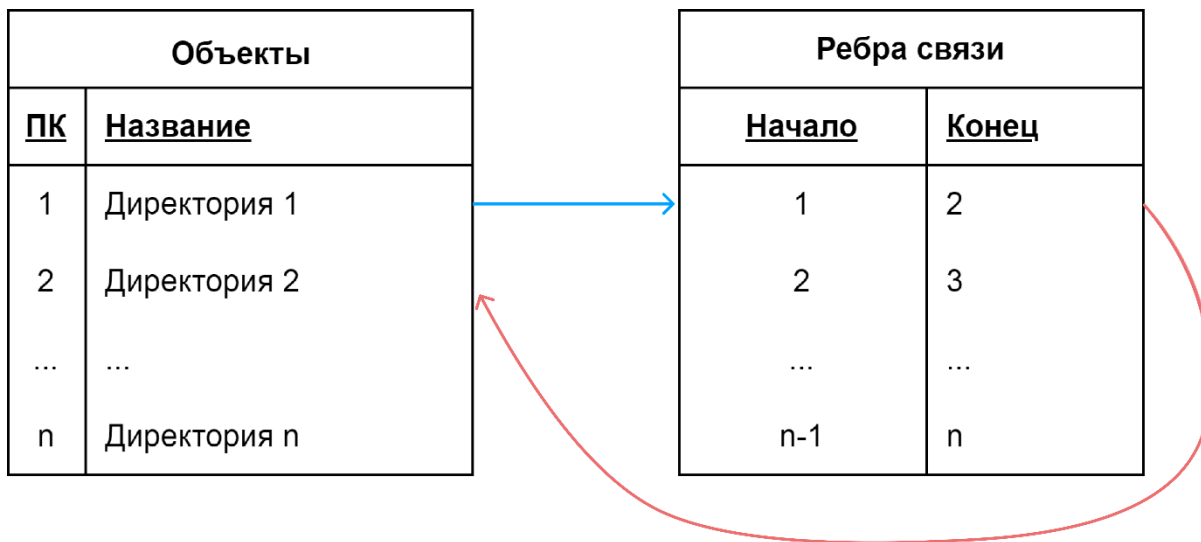


```
WITH Dirs, Subdirs
let dir = (FOR d IN Dirs
FILTER d.name == @name
return d)[0]
FOR v IN 1..1000
OUTBOUND dir Subdirs
return v.name
```



# Пример реализации графовой связи в реляционной БД

# Пример реализации графовой связи в реляционной БД



# Пример рекурсивного запроса в PostgreSQL

# Пример рекурсивного запроса в PostgreSQL

```
with recursive
  in_dirs(id, depth) as (
    select id, 1
    from dirs where name = %s
    union
    select e.tail_id, in_dirs.depth + 1 from edges_dirs as e
    join in_dirs
    on e.head_id = in_dirs.id
    where in_dirs.depth < %s
  )
select dirs.name, in_dirs.depth from in_dirs
join dirs
on in_dirs.id = dirs.id
```

# Пример рекурсивного запроса в PostgreSQL

```
with recursive
  in_dirs(id, depth) as (
    select id, 1
    from dirs where name = %s
    union
    select e.tail_id, in_dirs.depth + 1 from edges_dirs as e
    join in_dirs
    on e.head_id = in_dirs.id
    where in_dirs.depth < %s
  )
select dirs.name, in_dirs.depth from in_dirs
join dirs
on in_dirs.id = dirs.id
```

# Пример рекурсивного запроса в PostgreSQL

```
with recursive
  in_dirs(id, depth) as (
    select id, 1
    from dirs where name = %s
    union
    select e.tail_id, in_dirs.depth + 1 from edges_dirs as e
    join in_dirs
    on e.head_id = in_dirs.id
    where in_dirs.depth < %s
  )
select dirs.name, in_dirs.depth from in_dirs
join dirs
on in_dirs.id = dirs.id
```

# Пример рекурсивного запроса в PostgreSQL



```
with recursive
  in_dirs(id, depth) as (
    select id, 1
    from dirs where name = %s
    union
    select e.tail_id, in_dirs.depth + 1 from edges_dirs as e
    join in_dirs
    on e.head_id = in_dirs.id
    where in_dirs.depth < %s
  )
select dirs.name, in_dirs.depth from in_dirs
join dirs
on in_dirs.id = dirs.id
```

# Особенности работы с ArangoDB для QA





# Различные алгоритмы поиска внутри БД

# Различные алгоритмы поиска внутри БД

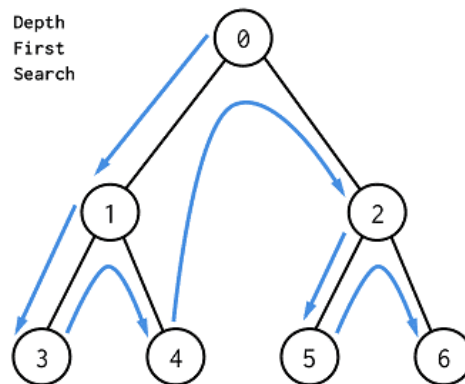
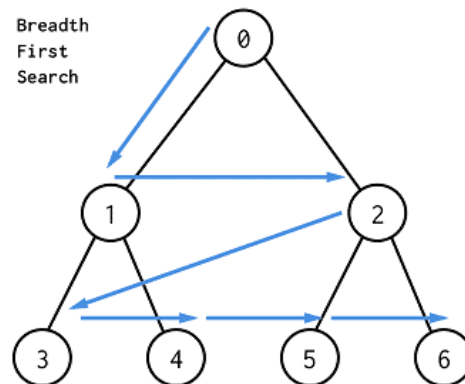
- Breadth-first search — поуровневое исследование графа

# Различные алгоритмы поиска внутри БД

- Breadth-first search — поуровневое исследование графа
- Depth-first search — «идем» вглубь графа, насколько возможно

# Различные алгоритмы поиска внутри БД

- Breadth-first search — поуровневое исследование графа
- Depth-first search — «идем» вглубь графа, насколько возможно



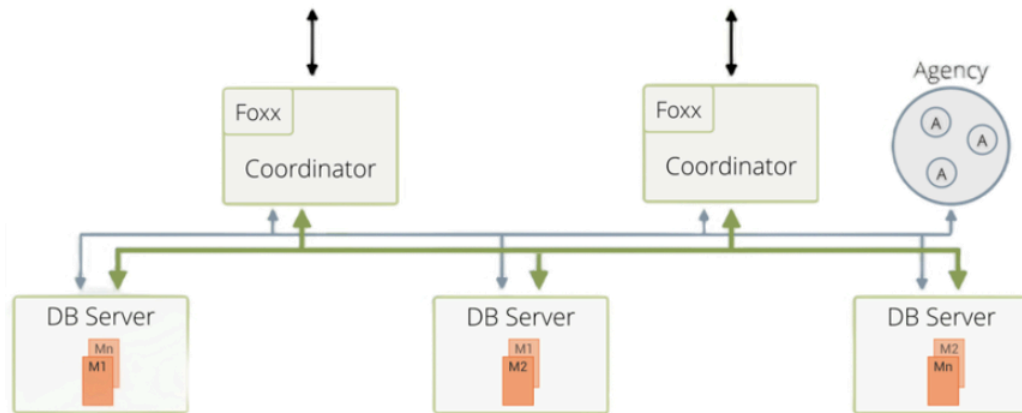
# Сингл и кластерный варианты развертывания

# Сингл и кластерный варианты развертывания

- Свойства различны
- Выбор зависит от целевого использования приложения
- Требуют разных ресурсов
- Настройки и фичи могут применяться не одинаково

# Сингл и кластерный варианты развертывания

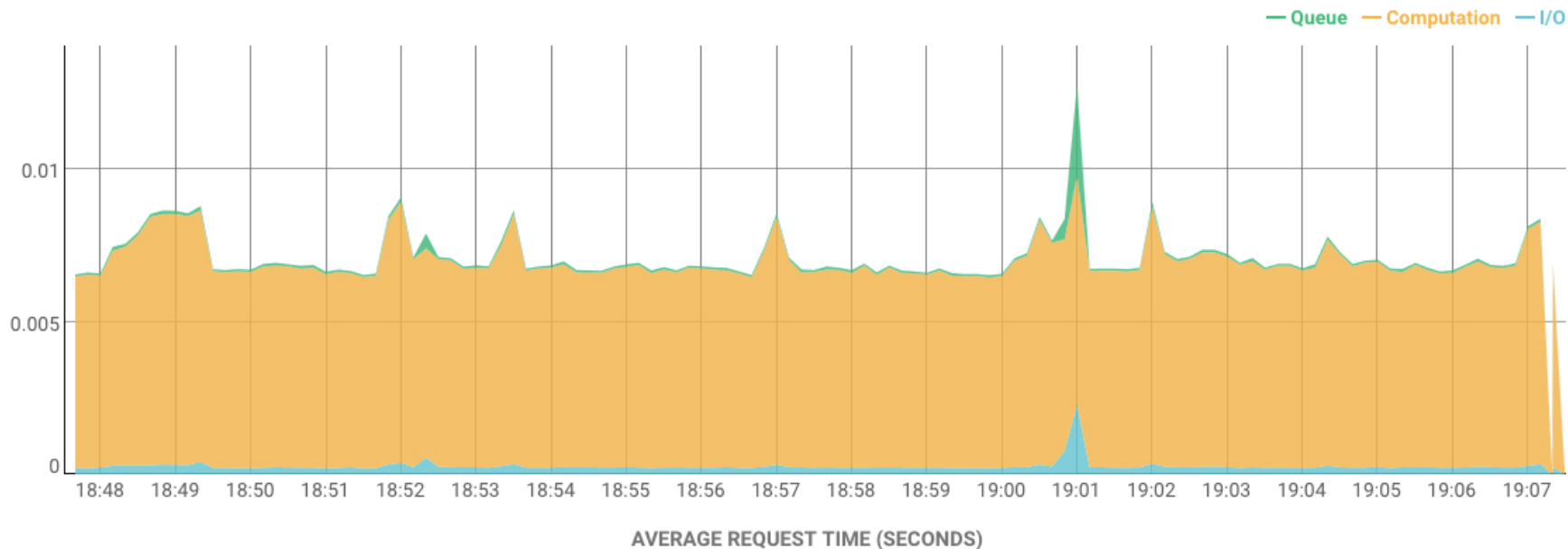
- Свойства различны
- Выбор зависит от целевого использования приложения
- Требуют разных ресурсов
- Настройки и фичи могут применяться не одинаково



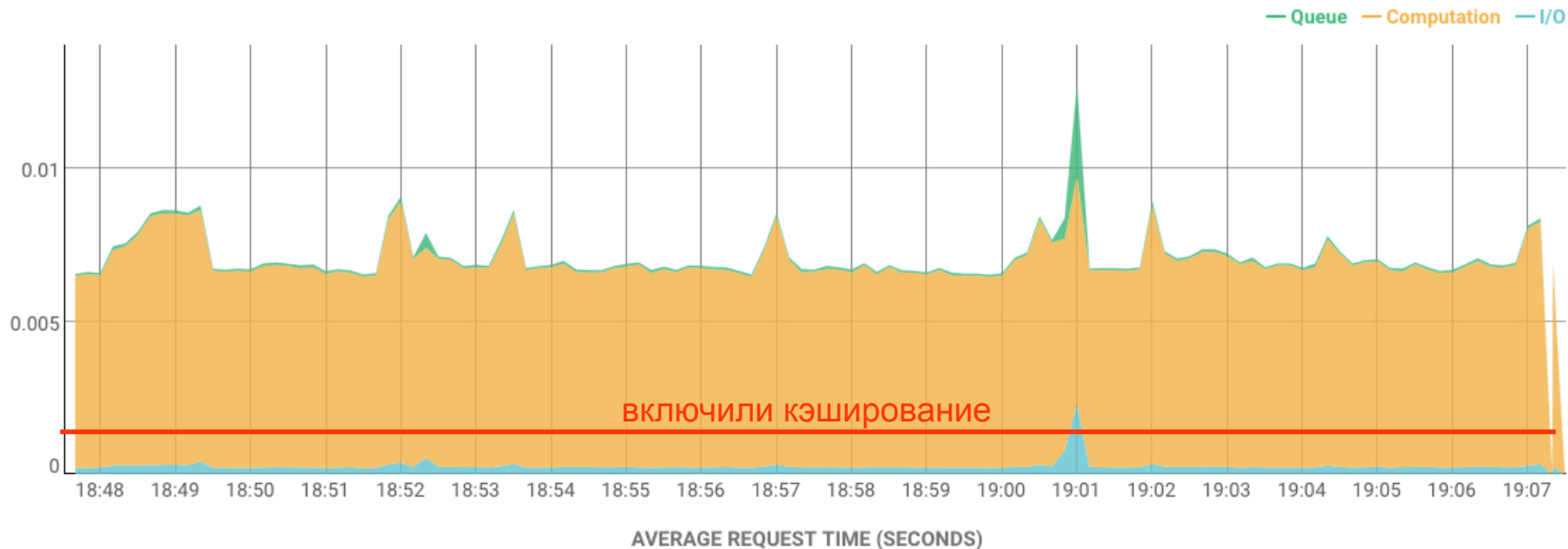
# Возможность кэширования запросов ArangoDB



# Возможность кэширования запросов ArangoDB



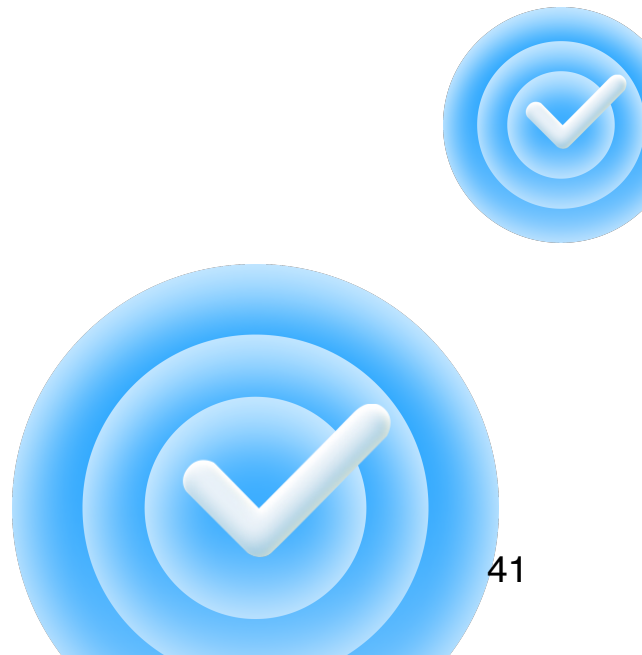
# Возможность кэширования запросов ArangoDB



# Советы:

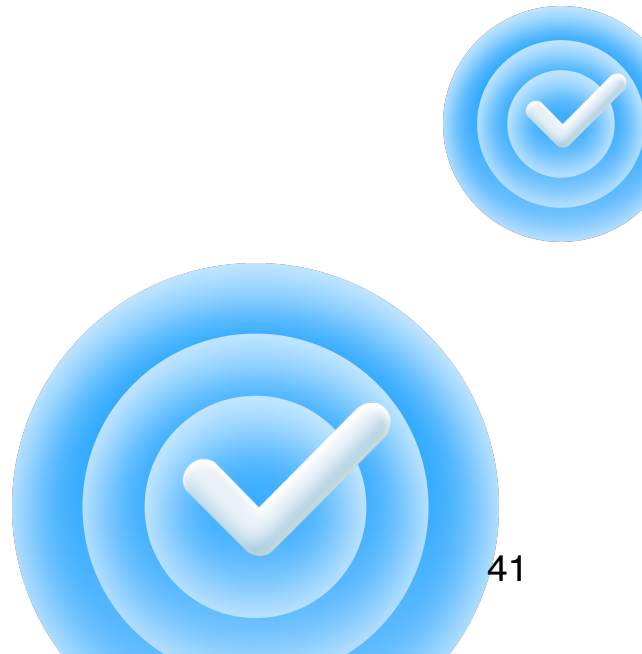


МойОфис



## Советы:

- Использование ArangoDB требует определенной архитектуры приложения



## Советы:

- Использование ArangoDB требует определенной архитектуры приложения
- Графовая модель требует определенных типов взаимосвязи данных

## Советы:

- Использование ArangoDB требует определенной архитектуры приложения
- Графовая модель требует определенных типов взаимосвязи данных
- Используете графовую БД? Мыслите конструктивно!



## Советы:

- Использование ArangoDB требует определенной архитектуры приложения
- Графовая модель требует определенных типов взаимосвязи данных
- Используете графовую БД? Мыслите конструктивно!
- Нагрузочное тестирование - обязательно, с акцентом на движок ArangoDB



## Советы:

- Использование ArangoDB требует определенной архитектуры приложения
- Графовая модель требует определенных типов взаимосвязи данных
- Используете графовую БД? Мыслите конструктивно!
- Нагрузочное тестирование - обязательно, с акцентом на движок ArangoDB
- Не забывайте проверить работу приложения с другим типом инсталляции/настроек графовой БД.






# Спасибо за внимание!



[Форма для вопросов](#)



[Тестовый репозиторий](#)

 @anna\_rukavitsyna96

 @revareva96

