Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Отчёт по лабораторной работе №2 в рамках дисциплины «Распределенные системы хранения данных»

Вариант 756

Выполнил: студент группы P33131

Бусыгин Дмитрий Алексеевич

Преподаватель:

Афанасьев Дмитрий Борисович

Санкт-Петербург
2024

Задание **3**
Этап 1. Инициализация кластера БД 3
Этап 2. Конфигурация и запуск сервера БД 3
Этап 3. Дополнительные табличные пространства и наполнение базы 3
Выполнение: **4**
Подключение: 4
Инициализация кластера: **4**
Конфигурация и запуск сервера: 4
Задание параметров: **5**
Логирование: **5**
Запуск: **5**
Выключение: **6**
Этап 3. Дополнительные табличные пространства и наполнение базы 6
Дополнительные табличные пространства и наполнение: **6**
Вывод: **6**

Задание

Цель работы - на выделенном узле создать и сконфигурировать новый кластер БД Postgres, саму БД, табличные пространства и новую роль, а также произвести наполнение базы в соответствии с заданием. Отчёт по работе должен содержать все команды по настройке, скрипты, а также измененные строки конфигурационных файлов.
Способ подключения к узлу из сети Интернет через helios:
ssh -J sXXXXXX@helios.cs.ifmo.ru:2222 postgresY@pgZZZ
Способ подключения к узлу из сети факультета:
ssh postgresY@pgZZZ
Номер выделенного узла pgZZZ, а также логин и пароль для подключения Вам выдаст преподаватель.

Этап 1. Инициализация кластера БД

- Директория кластера: $HOME/kmu66
- Кодировка: UTF8
- Локаль: английская
- Параметры инициализации задать через переменные окружения

Этап 2. Конфигурация и запуск сервера БД

- Способы подключения: 1) Unix-domain сокет в режиме peer; 2) сокет TCP/IP, принимать подключения к любому IP-адресу узла
- Номер порта: 9756
- Способ аутентификации TCP/IP клиентов: по паролю SHA-256
- Остальные способы подключений запретить.
- Настроить следующие параметры сервера БД:
    - max_connections
    - shared_buffers
    - temp_buffers
    - work_mem
    - checkpoint_timeout
    - effective_cache_size
    - fsync
    - commit_delay
- Параметры должны быть подобраны в соответствии с аппаратной конфигурацией: оперативная память 24ГБ, хранение на жёстком диске (HDD).
- Директория WAL файлов: $HOME/lrw87
- Формат лог-файлов: .log
- Уровень сообщений лога: WARNING
- Дополнительно логировать: контрольные точки и попытки подключения

Этап 3. Дополнительные табличные пространства и наполнение базы

- Создать новое табличное пространство для индексов: $HOME/vst56
- На основе template0 создать новую базу: bestorangedata
- Создать новую роль, предоставить необходимые права, разрешить подключение к базе.

- От имени новой роли (не администратора) произвести наполнение ВСЕХ созданных баз тестовыми наборами данных. ВСЕ табличные пространства должны использоваться по назначению.
- Вывести список всех табличных пространств кластера и содержащиеся в них объекты.

Выполнение:

Подключение:

```
ssh s335103@se.ifmo.ru -p 2222
ssh postgres1@pg114
```

Инициализация кластера:

Имя узла – pg114

Имя пользователя – postgres1

Директория кластера - $HOME/kmu66

Кодировка, локаль – UTF-8, английская (locale -a)

Инициализация:

```
export LOCALE=en_US.UTF-8
export ENCODING=UTF-8

initdb --locale=$LOCALE --encoding=$ENCODING -D $HOME/kmu66
--username=postgres1 -W
```

Конфигурация и запуск сервера:

Для настройки аутентификации клиентов только по **Unix-domain сокету в режиме peer** или **сокету TCP/IP** и паролем закодированным **SHA-256**, необходимо отредактировать конфигурационный файл pg_hba.conf

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local      all          all                                     peer
# IPv4 local connections:
host       all          all             127.0.0.1/32            scram-sha-256
# IPv6 local connections:
host       all          all             ::1/128                 scram-sha-256
```

А также необходимо добавить конфигурацию хоста и порта в postgresql.conf:

```
port = 9756
listen_addresses = '*'
unix_socket_directories = '/tmp'
```

Задание параметров:

Параметры сервера БД должны быть подобраны в соответствии с аппаратной конфигурацией: оперативная память 24 ГБ, хранение на SSD: max_connections, shared_buffers, temp_buffers, work_mem, checkpoint_timeout, effective_cache_size, fsync, commit_delay.

postgresql.conf:
`max_connections = 100` (оставляю значение по умолчанию, тк мне неизвестно о потенциальном количестве пользователей)

`shared_buffers =GB` (обычно берётся 25% от оперативной памяти, но у нас так было ошибка could not fork process, а работало только с 1 гигабайтом)

`temp_buffers = 8MB` (оставляю по умолчанию)

`work_mem = 4MB` (по умолчанию, зависит от сложности запросов, чего я не знаю, но думаю не будет сложнее, чем базовый)

`checkpoint_timeout = 6min` (по умолчанию = 5, но выставляю больше, чтобы выигрывать в скорости, однако идёт небольшой проигрыш в надёжности)

`effective_cache_size = 4GB` (Значение по умолчанию обычно ставится относительно количество оперативной памяти. По дефолту, как мне кажется, выбрано довольно рациональное значение)

`fsync = on` (так мы теряем в скорости, но при сбое оборудования данные все равно хотелось бы восстанавливать)

`commit_delay = 0` (иначе только если мы хотим тестировать наш сервер и смотреть сколько времени уходит на выполнение транзакций)

Логирование:

Формат лог-файлов - .log, уровень сообщений лога - WARNING, дополнительно логировать - контрольные точки и попытки подключения

```
log_destination = 'stderr'

logging_collector = on

log_min_messages = warning

log_min_error_statement = error

log_connections = on

log_checkpoints = on

log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log'
```

Запуск:

```
pg_ctl -D $HOME/kmu66 -l logfile start
```

Выключение:

```
pg_ctl -D $HOME/kmu66 stop
```

Дополнительные табличные пространства и наполнение:

Создание нового табличного пространства:

```
mkdir -p $HOME/vst56
psql -p 9756 -h localhost postgres, password = chh455

CREATE TABLESPACE vst56 LOCATION '/var/db/postgres1/vst56';
```

Убедимся что табличное пространство создалось:

```
postgres=# select spcname, pg_tablespace_location(oid) from
pg_tablespace;
  spcname   | pg_tablespace_location
------------+-------------------------
 pg_default |
 pg_global  |
 vst56      | /var/db/postgres1/vst56
```

Создание новой базы данных:

```
CREATE DATABASE bestorangedata TEMPLATE template0;
```

Убедимся что база создалась:

```
postgres=# select datname from pg_database;
    datname
---------------
 postgres
 bestorangedata
 template1
 template0
```

Создание роли:

```
CREATE ROLE orange_owner LOGIN PASSWORD '123';

GRANT CONNECT, CREATE ON DATABASE bestorangedata TO orange_owner;
GRANT CREATE ON TABLESPACE vst56 TO orange_owner;
```

Заполнение базы данных:

```
psql -p 9756 -h localhost -U orange_owner bestorangedata;

CREATE TABLE IF NOT EXISTS oranges (
    id SERIAL PRIMARY KEY TABLESPACE vst56,
    color TEXT,
    volume INTEGER,
    harvest_date TIMESTAMP,
    description TEXT
);
```

По умолчанию индексы сохраняются в pg_data, а значит их tablespace пуст.

```
bestorangedata=> select tablename, indexname, tablespace from
pg_indexes where indexname = 'oranges_desc_idx';
 tablename |    indexname     | tablespace
-----------+------------------+------------
 oranges   | oranges_desc_idx |
```

```
ALTER INDEX IF EXISTS oranges_pkey SET TABLESPACE vst56;
CREATE INDEX IF NOT EXISTS oranges_color_idx ON oranges(color)
TABLESPACE vst56;
CREATE INDEX IF NOT EXISTS oranges_harvest_date_idx ON
oranges(harvest_date) TABLESPACE vst56;
```

```
bestorangedata=> SELECT
  c.relname,
  t.spcname
FROM
  pg_class c
    JOIN pg_tablespace t ON c.reltablespace = t.oid
WHERE
  t.spcname = 'vst56';
      relname              | spcname
---------------------------+---------
 oranges_pkey              | vst56
 oranges_color_idx         | vst56
 oranges_harvest_date_idx  | vst56
(3 строки)
```

```
INSERT INTO oranges (color, volume, harvest_date, description)
VALUES ('red', 123, now(), 'the best orange1');
INSERT INTO oranges (color, volume, harvest_date, description)
VALUES ('yellow', 321, now(), 'the best orange2');
INSERT INTO oranges (color, volume, harvest_date, description)
VALUES ('orange', 111, now(), 'the best orange3');
INSERT INTO oranges (color, volume, harvest_date, description)
VALUES ('orange', 222, now(), 'the best orange4');
```

Вывод:

В процессе выполнения лабораторной работы я ознакомился с конфигурацией различных настроек сервера PostgreSQL, попрактиковался в настройках сетевого подключения и логов на уровне сервера, а так же ознакомился с принципами работы табличных пространств