

УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Распределенные системы хранения данных»

Лабораторная работа №2

Вариант 54

Студент

Кузнецов М. А.

P33131

Преподаватель

Шешуков Д. М.

Санкт-Петербург, 2023 г.

Описание задания

На выделенном узле создать и сконфигурировать новый кластер БД, саму БД, табличные пространства и новую роль в соответствии с заданием. Произвести наполнение базы.

Отчёт должен содержать все команды по настройке, а также измененные строки конфигурационных файлов.

Подключение к узлу через helios:

1) ssh s666666@se.ifmo.ru -p 2222

2) ssh пользователь@узел

Персональный пароль для работы с узлом выдается преподавателем.

Обратите внимание, что домашняя директория пользователя

/var/postgres/\$LOGNAME

Этапы выполнения работы:

Инициализация кластера БД

- Имя узла — pg110.
- Имя пользователя — postgres2.
- Директория кластера БД — \$HOME/u06/dir6.
- Кодировка, локаль — ANSI1251, русская
- Перечисленные параметры задать через переменные окружения.

Конфигурация и запуск сервера БД

- Способ подключения к БД — TCP/IP socket, номер порта 9054.
- Остальные способы подключений запретить.
- Способ аутентификации клиентов — по паролю MD5.
- 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 8 ГБ, хранение на SSD;
- Директория WAL файлов — поддиректория в PGDATA.
- Формат лог-файлов — csv.
- Уровень сообщений лога — WARNING.
- Дополнительно логировать — попытки подключения и продолжительность выполнения команд.

Дополнительные табличные пространства и наполнение

- На основе шаблона template0 пересоздать базу postgres в новом табличном пространстве:
? \$HOME/u06/dir7.
 - На основе template1 создать новую базу — newtestdb4.
 - От имени новой роли (не администратора) произвести наполнение существующих баз тестовыми наборами данных. Предоставить права по необходимости. Табличные пространства должны использоваться по назначению.
 - Вывести список всех табличных пространств кластера и содержащиеся в них объекты.
- Пароль для подключения к узлу: <запросить>
- Пожалуйста, не выкладывайте пароль в общий доступ.
- Не забывайте останавливать свой экземпляр БД, когда прекращаете с ним работать.

Выполнение

Инициализируем кластер базы данных с помощью переменных окружения

```
PGDATA=$HOME/u06/dir6
PGENCODING=WIN1251
PGLOCALE=ru_RU.CP1251
PGSUSERSNAME=postgres2
```

```
export PGDATA PGLOCALE PGENCODING PGSUSERSNAME
```

```
mkdir -p $PGDATA
```

```
initdb --encoding=$PGENCODING --locale=$PGLOCALE --
username=$PGSUSERSNAME
```

```
[postgres2@pg110 ~]$ initdb --locale=$PGLOCALE --encoding=$PGENCODING --username=$PGSUSERSNAME
Файлы, относящиеся к этой СУБД, будут принадлежать пользователю "postgres2".
От его имени также будет запускаться процесс сервера.
```

```
Кластер баз данных будет инициализирован с локалью "ru_RU.CP1251".
Выбрана конфигурация текстового поиска по умолчанию "russian".
```

```
Контроль целостности страниц данных отключён.
```

```
создание каталога /var/db/postgres2/u06/dir6... ок
создание подкаталогов... ок
выбирается реализация динамической разделяемой памяти... posix
выбирается значение max_connections по умолчанию... 100
выбирается значение shared_buffers по умолчанию... 128MB
выбирается часовой пояс по умолчанию... W-SU
создание конфигурационных файлов... ок
выполняется подготовительный скрипт... ок
выполняется заключительная инициализация... ок
сохранение данных на диске... ок
```

```
initdb: предупреждение: включение метода аутентификации "trust" для локальных подключений
Другой метод можно выбрать, отредактировав pg_hba.conf или используя ключи -A,
--auth-local или --auth-host при следующем выполнении initdb.
```

```
Готово. Теперь вы можете запустить сервер баз данных:
```

```
pg_ctl -D /var/db/postgres2/u06/dir6 -l файл_журнала start
```

```
[postgres2@pg110 ~]$ █
```

Чтобы отредактировать файлы конфигурации, скачаем их предварительно к себе и изменим:

```
scp postgres2@pg110:u06/dir6/postgresql.conf /home/studs/s311694
```

```
scp postgres2@pg110:u06/dir6/pg_hba.conf /home/studs/s311694
```

pg_hba.conf

```
# TYPE DATABASE USER ADDRESS METHOD
# connection config for lab2
host all all all md5
# "local" is for Unix domain socket connections only
local all all reject
# IPv4 local connections:
host all all 127.0.0.1/32 reject
# IPv6 local connections:
host all all ::1/128 reject
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all reject
host replication all 127.0.0.1/32 reject
host replication all ::1/128 reject
```

postgresql.conf

Меняем порт для подключения, значение max_connections заданное по умолчанию.

```
# (change requires restart)
port = 9054 # (change requires restart)
max_connections = 100 # (change requires restart)
```

Меняем кодировку паролей на MD5

```
#authentication_timeout = 1min # 1s-600s
password_encryption = md5 # scram-sha-256 or md5
#db_user_namespace = off
```

Значение shared_buffers принято брать как четвертую часть от доступной памяти. В данном случае 8/4=2 ГБ. Выделение большего размера приведет к недостатку памяти -> к снижению производительности -> частый swapping-а страниц.

```
shared_buffers = 2048MB # min 128kB
# (change requires restart)
```

Значение temp_buffers для хранения временных таблиц нельзя делать слишком большим, чтобы избежать неэффективного использования памяти.

```
temp_buffers = 16MB # min 800kB
#max_prepared_transactions = 0 # zero
```

Параметр work_mem для операций чтения и сортировки выставяем так же, исходя из конфигурации.

```
work_mem = 16MB # min 64kB
```

Флаг fsync можно отключить если у нас read-only бд, в ином случае можем включить для повышения надежности.

```

# (change requires restart)
fsync = on          # flush data to disk for crash safety
# (turning this off can cause

```

Изменение задержки перед сохранением WAL имеет смысл только в том случае, если есть возможность протестировать его влияние на общую производительность. В нашем случае оставляем 0 по умолчанию.

Ставить задержки на сохранение WAL можно только если мы хотим протестировать это влияние на общую производительность системы.

А так как у нас такой цели не стоит, то ее можно убрать вовсе.

```

commit_delay = 0          # range 0-100000, in microseconds

```

Значение паузы между точками восстановления по умолчанию не изменяем.

```

checkpoint_timeout = 5min  # range 30s-1d

```

Параметр `effective_cache_size` влияет на эффективность планирования выполнения запросов.

ВАЖНО! Параметр не может быть меньше `shared_buffers`

```

effective_cache_size = 4GB

```

Логируем попытки подключения и длительность выполнения команд

```

#log_checkpoints = off
log_connections = on
#log_disconnections = off
log_duration = on
#log_error_verbosity = default

```

Уровень сообщений логирования

```

log_min_messages = warning  # values in order of decreasing detail:
#   debug5
#   debug4
#   debug3

```

Формат log - файлов

```

log_destination = 'csvlog'
#log_destination = 'stderr'      # Valid values are combinations of
#                                # stderr, csvlog, syslog, and eventlog,
#                                # depending on platform.  csvlog
#                                # requires logging_collector to be on.

# This is used when logging to stderr:
logging_collector = on # Enable capturing of stderr and csvlog
# into log files. Required to be on for
# csvlogs.
# (change requires restart)

# These are only used if logging_collector is on:
log_directory = 'log'          # directory where log files are written,
# can be absolute or relative to PGDATA

```

Запись WAL-файлов в поддиректорию PGDATA

```

archive_mode = on      # enables archiving; off, on, or always
# (change requires restart)
archive_command = 'cp %p $HOME/u06/dir6/archives/%f' # c
# placeholders: %p = path of file to archive

```

Теперь выгружаем наши измененные файлы обратно:

```

scp /home/studs/s311694/pg_hba.conf postgres2@pg110:u06/dir6/
scp /home/studs/s311694/postgresql.conf postgres2@pg110:u06/dir6/

```

Запускаем сервер PostgreSQL, создаем БД и роли

```

pg_ctl start -l log
psql -p 9054 -d template1 -h pg110

```

Задание: на основе шаблона template0 пересоздать базу postgres в новом табличном пространстве: \$HOME/u06/dir7.

```

CREATE TABLESPACE newtablespace LOCATION
'/var/db/postgres2/u06/dir7';

DROP DATABASE postgres;

CREATE DATABASE postgres WITH TEMPLATE = template0 TABLESPACE =
newtablespace;

```

Задание: на основе template1 создать новую базу — newtestdb4.

```

CREATE DATABASE newtestdb4 WITH TEMPLATE = template1;

```

Задание: от имени новой роли (не администратора) произвести наполнение существующих баз тестовыми наборами данных. Предоставить права по необходимости. Табличные пространства должны использоваться по назначению.

```

CREATE ROLE s311694 WITH LOGIN PASSWORD '1234';

```

```
psql -h pg110 -p 9054 -d newtestdb4 -f data_storage.sql
```

```
psql -h pg110 -p 9054 -d newtestdb4 -f insert.sql
```

```
[s311694@helios ~]$ psql -h pg110 -p 9054 -d newtestdb4 -f data_storage.sql
psql:data_storage.sql:1: ЗАМЕЧАНИЕ:  таблица "users" не существует, пропускается
psql:data_storage.sql:1: ЗАМЕЧАНИЕ:  таблица "payments" не существует, пропускается
psql:data_storage.sql:1: ЗАМЕЧАНИЕ:  таблица "orders" не существует, пропускается
psql:data_storage.sql:1: ЗАМЕЧАНИЕ:  таблица "user_payments" не существует, пропускается
DROP TABLE
psql:data_storage.sql:2: ЗАМЕЧАНИЕ:  тип "paymenttypes" не существует, пропускается
DROP TYPE
CREATE TYPE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
[s311694@helios ~]$ psql -h pg110 -p 9054 -d newtestdb4 -f insert.sql
INSERT 0 10
INSERT 0 10
INSERT 0 10
INSERT 0 10
[s311694@helios ~]$
```

Задание: вывести список всех табличных пространств кластера и содержащиеся в них объекты.

```
SELECT * FROM pg_tablespace;
```

```
postgres=# select * from pg_tablespace;
 oid |      spcname      | spcname | spcname | spcname |
-----+-----+-----+-----+-----+
 1663 | pg_default        |         |         |         |
 1664 | pg_global         |         |         |         |
16387 | newtablespace     |         |         |         |
(3 строки)
```

```
SELECT c.relname, t.spcname FROM pg_class c JOIN pg_tablespace t
ON c.reltablespace = t.oid;
```

```
postgres=# SELECT c.relname, t.spcname FROM pg_class c JOIN pg_tablespace t ON c.reltablespace = t.oid;
```

relname	spcname
pg_toast_1262	pg_global
pg_toast_1262_index	pg_global
pg_toast_2964	pg_global
pg_toast_2964_index	pg_global
pg_toast_1213	pg_global
pg_toast_1213_index	pg_global
pg_toast_1260	pg_global
pg_toast_1260_index	pg_global
pg_toast_2396	pg_global
pg_toast_2396_index	pg_global
pg_toast_6000	pg_global
pg_toast_6000_index	pg_global
pg_toast_3592	pg_global
pg_toast_3592_index	pg_global
pg_toast_6100	pg_global
pg_toast_6100_index	pg_global
pg_database_datname_index	pg_global
pg_database_oid_index	pg_global
pg_db_role_setting_databaseid_rol_index	pg_global
pg_tablespace_oid_index	pg_global
pg_tablespace_spcname_index	pg_global
pg_authid_rolname_index	pg_global
pg_authid_oid_index	pg_global
pg_auth_members_role_member_index	pg_global
pg_auth_members_member_role_index	pg_global
pg_shdepend_depender_index	pg_global
pg_shdepend_reference_index	pg_global
pg_shdescription_o_c_index	pg_global
pg_replication_origin_roident_index	pg_global
pg_replication_origin_roname_index	pg_global
pg_shseclabel_object_index	pg_global
pg_subscription_oid_index	pg_global
pg_subscription_subname_index	pg_global
pg_authid	pg_global
pg_subscription	pg_global
pg_database	pg_global
pg_db_role_setting	pg_global
pg_tablespace	pg_global
pg_auth_members	pg_global
pg_shdepend	pg_global
pg_shdescription	pg_global
pg_replication_origin	pg_global
pg_shseclabel	pg_global

```
SELECT d.datname, t.spcname FROM pg_tablespace t JOIN pg_database d ON d.dattablespace = t.oid;
```

```
postgres=# SELECT d.datname, t.spcname FROM pg_tablespace t JOIN pg_database d ON d.dattablespace = t.oid;
```

datname	spcname
postgres	newtablespace
template1	pg_default
template0	pg_default
newtestdb4	pg_default

(4 строки)

Вывод

Во время выполнения лабораторной работы я:

- научился создавать и конфигурировать кластер БД PostgreSQL.
- познакомился с созданием и работой табличных пространств и ролей.