

Университет ИТМО  
Факультет программной инженерии и компьютерной техники

# Распределённые системы хранения данных. Лабораторная работа №2.

Группа: Р33131  
Студент: Смирнов Виктор Игоревич  
Преподаватель: Афанасьев Дмитрий Борисович  
Вариант: 666

# Ключевые слова

База данных, конфигурация PostgreSQL.

## Содержание

<b>1</b>	<b>Цель работы</b>	<b>1</b>
<b>2</b>	<b>Текст задания</b>	<b>1</b>
2.1	Инициализация кластера БД	1
2.2	Конфигурация и запуск сервера БД	1
2.3	Дополнительные табличные пространства и наполнение базы	2
<b>3</b>	<b>Ход работы</b>	<b>2</b>
3.1	Порядок конфигурации кластера	2
3.2	Подготовка к установке	2
3.3	Инициализация датабазы	3
3.4	Установка конфигурационных файлов	3
3.5	Host-based authentication	3
3.6	Главный конфигурационный файл	3
3.7	Запуск датабазы	4
3.8	Создание табличного пространства и новой датабазы	4
3.9	Заполнение новой датабазы	4
3.10	Вывод табличных пространств кластера и объектов	5
<b>4</b>	<b>Вывод</b>	<b>5</b>

## 1 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.

## 2 Текст задания

### 2.1 Инициализация кластера БД

1. Директория кластера: \$HOME/kor67
2. Кодировка: ANSI251
3. Локаль: русская
4. Параметры инициализации задать через аргументы команды

### 2.2 Конфигурация и запуск сервера БД

1. Способ подключения: сокет TCP/IP, только localhost
2. Номер порта: 9666
3. Остальные способы подключений запретить.
4. Способ аутентификации клиентов: по имени пользователя
5. Настроить следующие параметры сервера БД:
  - (a) max\_connections
  - (b) shared\_buffers

- (c) temp\_buffers
- (d) work\_mem
- (e) checkpoint\_timeout
- (f) effective\_cache\_size
- (g) fsync
- (h) commit\_delay

6. Параметры должны быть подобраны в соответствии со сценарием OLAP:

- 5 одновременных пользователей
- пакетная запись/чтение данных по 128МБ.

7. Директория WAL файлов: \$PGDATA/pg\_wal

8. Формат лог-файлов: .log

9. Уровень сообщений лога: WARNING

10. Дополнительно логировать: попытки подключения и завершение сессий

## 2.3 Дополнительные табличные пространства и наполнение базы

1. Пересоздать шаблон template1 в новом табличном пространстве: \$HOME/yqu90
2. На основе template1 создать новую базу: lazyorangehair
3. Создать новую роль, предоставить необходимые права, разрешить подключение к базе.
4. От имени новой роли (не администратора) произвести наполнение всех созданных баз тестовыми наборами данных. Все табличные пространства должны использоваться по назначению.
5. Вывести список всех табличных пространств кластера и содержащиеся в них объекты.

## 3 Ход работы

### 3.1 Порядок конфигурации кластера

```

1 exit 0
2
3 # login 1: postgres
4
5 cd psql
6 source script/1-initialization.sh
7 source script/2-create.sh
8 source script/3-configuration.sh
9 source script/4-start.sh
10
11 # login 2: root
12
13 cd psql
14 source script/1-initialization.sh
15 # ctrl+d
16 source script/5-recreate-template.sh
17 source script/6-fill-tables.sh
18
19 # login 3: postgres
20 cd psql
21 source script/1-initialization.sh
22 source script/7-print.sh

```

### 3.2 Подготовка к установке

```

1 export POSTGRES_USER=postgres0
2 export POSTGRES_CONFIG_DIRECTORY="./config"
3 export POSTGRES_SUPER_USER_PASSWORD_FILE="$POSTGRES_CONFIG_DIRECTORY/
  postgres-su-password.txt"
4 export POSTGRES_PORT=9666
5 export PGDATA="$HOME/kop67"
6
7 mkdir      $PGDATA                        2> /dev/null
8
9 adduser    $POSTGRES_USER                  2>/dev/null
10 chown      $POSTGRES_USER $PGDATA
11
12 su        $POSTGRES_USER

```

### 3.3 Инициализация базы

```

1 export PG_BIN=/usr/lib/postgresql/14/bin
2
3 "initdb" \
4   --pgdata=$PGDATA \
5   --locale="ru_RU.CP1251" \
6   --encoding=WIN1251 \
7   --pwfile=$POSTGRES_SUPER_USER_PASSWORD_FILE

```

### 3.4 Установка конфигурационных файлов

```

1 cp $POSTGRES_CONFIG_DIRECTORY/pg_hba.conf      $PGDATA/pg_hba.conf
2 cp $POSTGRES_CONFIG_DIRECTORY/postgresql.conf  $PGDATA/postgresql.conf

```

### 3.5 Host-based authentication

```

1 # $PGDATA/pg_hba.conf (Host-based authentication)
2
3 # Note: We wanted to have username based auth, but
4 #       ident server is unavailable at our machine,
5 #       so current solution is password
6
7 # TYPE  DATABASE  USER  ADDRESS        METHOD
8 host    all       all    127.0.0.1/32   scram-sha-256 # ident
9 host    all       all    ::1/128        scram-sha-256 # ident

```

### 3.6 Главный конфигурационный файл

Кроме требуемых параметров попытался подкрутить из интереса. Было бы интересно узнать, насколько адекватные решения я принял.

```

1 # $PGDATA/postgresql.conf (PostgreSQL configuration file)
2
3 # Note: Optimized for OLAP load:
4 # 5 users, packet r/w 128MB
5
6 ## CONNECTIONS
7
8 listen_addresses = '127.0.0.1' # Available only from localhost
9 port             = 9666         # For security
10 unix_socket_directories = ''    # Only TCP/IP
11
12 max_connections   = 6 # 5 users + 1 extra
13 superuser_reserved_connections = 3
14
15
16 ## AUTHENTICATION
17
18 authentication_timeout = 20s      # Type password faster
19 password_encryption    = scram-sha-256 # Strong password hashing
20
21 ## RESOURCE USAGE
22
23 shared_buffers         = 1024MB   # 128MB * (5 + 3) users
24 temp_buffers           = 128MB     # 128MB
25 max_prepared_transactions = 0      # We don't use transactions
26 work_mem               = 256MB     # Expected packet size

```

```

27 hash_mem_multiplier      = 1.5      # Smaller hash tables
28 maintenance_work_mem    = 64MB     # ?
29 autovacuum_work_mem      = -1
30 max_stack_depth          = 4MB      # Be prepared for complex queries
31
32 temp_file_limit          = 4GB      # Something is wrong if we reach this
33
34 ## WRITE-AHEAD LOG
35
36 checkpoint_timeout       = 5min
37 fsync                    = off       # Lost data is not critical, as we can recreate
38 synchronous_commit       = off       # Same
39 wal_compression          = off       # WAL must not be so huge?
40 commit_delay             = 200       # Acceptable to lose 200mc of data
41 effective_cache_size     = 4GB      # OK?
42
43
44 ## REPORTING AND LOGGING
45
46 log_destination          = 'stderr'
47 logging_collector        = off
48 log_directory            = 'log'
49 log_filename             = 'postgresql-%Y-%m-%d_%H%M%S.log'
50 log_min_messages         = warning
51
52 log_connections          = on
53 log_disconnections       = on

```

### 3.7 Запуск базы

```
1 postgres -D $PGDATA
```

### 3.8 Создание табличного пространства и новой базы

```

1 export TABLESPACE_NAME=yqy90
2 export TABLESPACE_LOCATION=$HOME/$TABLESPACE_NAME
3 export NEW_DATABASE_NAME=lazyorangehair
4 export POSTGRES_PORT=9666
5 export NEW_USER=root
6
7 mkdir $TABLESPACE_LOCATION 2>/dev/null
8
9 sql() {
10     psql -h localhost -p $POSTGRES_PORT -c "$1" postgres
11 }
12
13 sql "CREATE TABLESPACE $TABLESPACE_NAME LOCATION '$TABLESPACE_LOCATION';"
14 sql "ALTER DATABASE template1 SET TABLESPACE $TABLESPACE_NAME;"
15 sql "CREATE DATABASE $NEW_DATABASE_NAME TEMPLATE template1;"
16 sql "CREATE ROLE tester;"
17 sql "CREATE USER $NEW_USER WITH LOGIN PASSWORD 'rootik'; GRANT tester TO $NEW_USER;"

```

### 3.9 Заполнение новой базы

```

1 export PRV=postgres
2 export NEW=lazyorangehair
3 export TABLESPACE_NAME=yqy90
4
5 sql() {
6     psql -U $NEW_USER -h localhost -p $POSTGRES_PORT -c "$2" "$1"
7 }
8
9 sql $PRV "CREATE TABLE note_prv (id serial PRIMARY KEY, content text NOT NULL);"
10 sql $NEW "CREATE TABLE note_new (id serial PRIMARY KEY, content text NOT NULL);"
11
12 sql $PRV "INSERT INTO note_prv (content) VALUES ('Note at postgres');"
13 sql $NEW "INSERT INTO note_new (content) VALUES ('Note at lazyorangehair');"
14
15 sql $PRV "SELECT * FROM note_prv;"
16 sql $NEW "SELECT * FROM note_new;"

```

### 3.10 Вывод табличных пространств кластера и объектов

```
1 export PRV=postgres
2 export NEW=lazyorangehair
3
4 sql() {
5     psql -h localhost -p $POSTGRES_PORT -d $PRV -f $1
6 }
7
8 sql "script/tablespace.sql"

1 DROP VIEW IF EXISTS meta_tablespace CASCADE;
2 CREATE VIEW meta_tablespace AS
3     SELECT
4         pg_tablespace.oid           AS id,
5         pg_tablespace.spcname       AS name
6     FROM pg_tablespace;
7
8 DROP VIEW IF EXISTS meta_object CASCADE;
9 CREATE VIEW meta_object AS
10     SELECT
11         pg_class.oid                AS id,
12         pg_class.relname            AS name,
13         (
14             CASE WHEN pg_class.reltablespace = 0
15             THEN (
16                 SELECT id FROM meta_tablespace
17                 WHERE meta_tablespace.name = 'pg_default'
18             )
19             ELSE pg_class.reltablespace END
20         ) AS tablespace_id
21     FROM pg_class
22     WHERE pg_class.relkind IN ('r', 'i', 'S', 'v');
23
24 DO $$
25 DECLARE
26     tablespace record;
27     object      record;
28 BEGIN
29     FOR tablespace IN
30         SELECT * FROM meta_tablespace
31     LOOP
32         RAISE INFO 'TABLESPACE %', tablespace.name;
33         FOR object IN
34             SELECT * FROM meta_object
35             WHERE meta_object.tablespace_id = tablespace.id
36         LOOP
37             RAISE INFO '  TABLE %', object.name;
38         END LOOP;
39     END LOOP;
40 END $$;
```

## 4 Вывод

Данная лабораторная работа помогла мне изучить конфигурацию PostgreSQL.

## Список литературы

- [1] PostgreSQL Documentation, System Catalogs: сайт. - 2024. - URL: <https://www.postgresql.org/docs/14/index.html> (дата обращения: 06.04.2024) - Текст : электронный.