

НИУ ИТМО

Факультет программной инженерии и компьютерных технологий

Отчет по лабораторной работе №2

по дисциплине Распределенные системы хранения данных

Студент группы № Р33151

Шипулин Павел Андреевич

Преподаватель

Перцев Тимофей Сергеевич

Санкт-Петербург

2024

# Задание

## Этап 1. Инициализация кластера БД

- Директория кластера: \$HOME/yqi56
- Кодировка: ANSI1251
- Локаль: русская
- Параметры инициализации задать через аргументы команды

## Этап 2. Конфигурация и запуск сервера БД

- Способы подключения: 1) Unix-domain сокет в режиме peer; 2) сокет TCP/IP, только localhost
- Номер порта: 9142
- Способ аутентификации TCP/IP клиентов: по паролю MD5
- 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 со сценарием OLAP: 5 одновременных пользователей, пакетная запись/чтение данных по 192МБ.

- Директория WAL файлов: \$PGDATA/pg\_wal
- Формат лог-файлов: .log

- Уровень сообщений лога: WARNING
- Дополнительно логировать: попытки подключения и завершение сессий

### **Этап 3. Дополнительные табличные пространства и наполнение базы**

- Создать новые табличные пространства для различных таблиц: \$HOME/mwd84, \$HOME/orw97, \$HOME/uzb16
- На основе template0 создать новую базу: sickorangepcity
- 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

## Этап 1. Инициализация кластера БД

```
PGDATA=$HOME/yqi56
PGLOCALE=ru_RU.CP1251
PGENCODING=WIN1251
PGUSERNAME=postgres6
PGHOST=pg100
export PGDATA PGLOCALE PGENCODING PGUSERNAME PGHOST

mkdir $PGDATA

initdb --locale=$PGLOCALE --encoding=$PGENCODING --username=$PGUSERNAME

pg_ctl -D /var/db/postgres6/yqi56 -l logfile start
```

Замечание преподавателя: нет смысла указывать файл лога в команде запуска, так как по заданию далее идет настройка файла лога в postgresql.conf.

## Этап 2. Конфигурация и запуск сервера БД

### 1. Настройка способов подключения

Изменения в файле 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	md5
# IPv6 local connections:					
host	all		all	:::1/128	md5
# 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

Изменения в файле postgresql.conf:

```
64 port = 9142
```

```
96 password_encryption = md5
```

## 2. Параметры сервера в соответствии со сценарием OLAP

- `max_connections` — отвечает за максимальное количество одновременных подключений.

Значение: 5

Замечание преподавателя: хотя бы одно соединение должно быть всегда доступно администратору, поэтому значение параметра должно быть хотя бы 6.

- `shared_buffers` — определяет количество памяти, которое сервер использует для буферов общей памяти (доступной различным процессам). Рекомендуется устанавливать примерно 25% от доступной памяти ОЗУ (устанавливать много больше нет смысла так как postgres активно использует кэш). Изменение (более 25% ОЗУ) требует соответствующего изменения `max_wal_size`.

Значение: 2GB.

- `temp_buffers` — определяет максимальное количество памяти, которое сервер использует для буферов отдельных сессий при доступе к временным таблицам.

Значение: 8MB (по-умолчанию).

- `work_mem` — устанавливает количество памяти, которое используется в каждой отдельной операции запроса (например, `order by` / `merge join` в запросе может быть несколько) перед записью во временные файлы диска.

Значение: 4MB (по-умолчанию).

Замечание преподавателя: так как по условию объемы чтения/записи велики, а количество пользователей ограничено малым числом — этот параметр можно значительно увеличить.

- `checkpoint_timeout` – максимальное время между автоматической установкой контрольных точек для WAL файлов. Чем меньше – тем чаще создаются контрольные точки, тем больше нагрузки.

Значение: 5min (по-умолчанию).

- `effective_cache_size` – устанавливает предположение планировщика об эффективном размере дискового кэша, доступного для одного запроса. Чем больше – тем более вероятней, что планировщик будет использовать сканирования индекса таблицы.

Значение: 2 GB.

- `fsync` – если включен, то сервер будет пытаться проверять, записаны ли изменения данных из основной памяти на диск.

Значение: on.

- `commit_delay` – устанавливает задержку перед записью WAL файлов на диск. Чем больше – тем больше (успевших стать готовыми) транзакций будут записаны одной операцией WAL flush.

Значение: 1s.

Замечание преподавателя: задержка перед записью такие коммитов не имеет особого смысла в данной системе, так как упор идет на большой объем коммитов при их малом количестве.

### 3. Настройка WAL

```

242 # - Archiving -
243
244 archive_mode = on           # enables archiving; off, on, or always
245                             # (change requires restart)
246
247 archive_command = 'cp %p $HOME/yqi56/pg_wal/%f'
```

Замечание преподавателя: архивация по заданию не нужна.

### 4. Настройка логов

Изменения в файле `postgresql.conf`:

```

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

```
478 log_min_messages = warning
538 log_connections = on
539 log_disconnections = on
```

## Этап 3. Дополнительные табличные пространства и наполнение базы

### 1. Создание табличных пространств

```
create tablespace ts_mwd84 location '/var/postgres/postgres6/mwd84';
create tablespace ts_orw97 location '/var/postgres/postgres6/orw97';
create tablespace ts_uzb16 location '/var/postgres/postgres6/uzb16';
```

### 2. Создание новой базы данных

```
create database sickorangecity with template = template0;

create table sickorange (
    id serial primary key,
    name text,
    birthdate date
)
tablespace ts_mwd84;

create table building (
    id serial primary key,
    street_id int,
    foreign key street_id references street(id),
    name text
)
tablespace ts_orw97;

create table street (
    id serial primary key,
    name text
)
tablespace ts_uzb16;
```

### 3. Создание и настройка роли

```
create role s311817 login password '123';
grant select, insert, update, delete
on table sickorange, building, street
to s311817;
```

#### 4. Наполнение базы данных

```
insert into sickorange values (default, 'SickOrange3812536', '01/01/2000');
insert into sickorange values (default, 'SickOrange812738', '01/02/2001');
insert into sickorange values (default, 'SickOrange1234512', '02/01/2000');
insert into sickorange values (default, 'SickOrange94721', '03/03/2003');
insert into sickorange values (default, 'SickOrange712637', '11/11/2000');
insert into sickorange values (default, 'SickOrange761273', '09/11/2006');
insert into sickorange values (default, 'SickOrange9812378', '10/28/2000');
insert into sickorange values (default, 'SickOrange761237', '07/17/2004');
```

```
insert into street values (default, 'Main');
insert into street values (default, 'Branch13231');
insert into street values (default, 'Branch3211');
insert into street values (default, 'Branch98392');
```

```
insert into building values (default, 1, 'Building132763172');
insert into building values (default, 1, 'Building18237');
insert into building values (default, 1, 'Building9812376');
insert into building values (default, 1, 'Building1236712');
insert into building values (default, 2, 'Building99999');
insert into building values (default, 2, 'Building182620');
insert into building values (default, 2, 'Building87123671');
insert into building values (default, 3, 'Building11111111');
insert into building values (default, 3, 'Building331823782');
insert into building values (default, 3, 'Building323232323');
insert into building values (default, 3, 'Building8718273');
insert into building values (default, 3, 'Building9919237');
insert into building values (default, 3, 'Building3267666');
insert into building values (default, 4, 'Building8129362');
insert into building values (default, 4, 'Building993827');
insert into building values (default, 4, 'Building38382173');
```

#### 5. Список табличных пространств и объектов в кластере

```
sickorangecity=# select * from pg_tablespace;
 oid | spcname | spcowner | spcacl | spcoptions
-----+-----+-----+-----+-----
 1663 | pg_default |      10 |        |
 1664 | pg_global |      10 |        |
16388 | ts_uzbl6 |      10 |        |
16389 | ts_mwd84 |      10 |        |
16390 | ts_orw97 |      10 |        |
(5 строк)
```



```
sickorangecity=# select relname, spcname from
pg_class pgc inner join pg_tablespace pgts
on pgc.reltablespace = pgts.oid
where spcname in ('ts_uzb16', 'ts_orw97', 'ts_mwd84');
```

relname	spcname
pg_toast_16445	ts_uzb16
pg_toast_16445_index	ts_uzb16
pg_toast_16454	ts_orw97
pg_toast_16454_index	ts_orw97
pg_toast_16427	ts_mwd84
pg_toast_16427_index	ts_mwd84
sickorange	ts_mwd84
building	ts_orw97
street	ts_uzb16

(9 строк)

```
sickorangecity=# select * from building;
```

id	street_id	name
1	1	Building132763172
2	1	Building18237
3	1	Building9812376
4	1	Building1236712
5	2	Building99999
6	2	Building182620
7	2	Building87123671
8	3	Building11111111
9	3	Building331823782
10	3	Building323232323
11	3	Building8718273
12	3	Building9919237
13	3	Building3267666
14	4	Building8129362
15	4	Building993827
16	4	Building38382173

(16 строк)

```
sickorangecity=# select * from sickorange;
```

id	name	birthdate
1	SickOrange3812536	2000-01-01
2	SickOrange812738	2001-02-01
3	SickOrange1234512	2000-01-02
4	SickOrange94721	2003-03-03
5	SickOrange712637	2000-11-11
6	SickOrange761273	2006-11-09
7	SickOrange9812378	2000-08-28
8	SickOrange761237	2004-07-17

(8 строк)

```
sickorangecity=# select * from street;
```

id	name
1	Main
2	Branch13231
3	Branch3211
4	Branch98392

(4 строки)

## Вывод

Научился инициализировать и настраивать кластеры БД postgresql. Создал новую базу данных на основе шаблона template0 и наполнил её тестовыми объектами. Научился создавать роли пользователей и устанавливать им права.