федеральное государственное автономное образовательное учреждение
высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

ОТЧЕТ

по лабораторной работе №2

по дисциплине «**Распределенные системы хранения данных**»

Вариант 35

Автор: Кулаков Н. В.

Факультет: ПИиКТ

Группа: P33312

Преподаватель: Шешуков Д. М.

Санкт-Петербург 2023

Постановка задачи и исходные данные

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.

Этапы выполнения работы:

Инициализация кластера БД

- Имя узла — pg107.
- Имя пользователя — postgres2.
- Директория кластера БД — $HOME/u05/dir5.
- Кодировка, локаль — ISO_8859_5, русская
- Перечисленные параметры задать через переменные окружения.

Конфигурация и запуск сервера БД

- Способ подключения к БД — TCP/IP socket, номер порта 9035.
- Остальные способы подключений запретить.
- Способ аутентификации клиентов — по паролю в открытом виде.
- 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 16 ГБ, хранение на SSD;
- Директория WAL файлов — $HOME/u05/dir6.
- Формат лог-файлов — log.

- Уровень сообщений лога — ERROR.
- Дополнительно логировать — контрольные точки и попытки подключения. Дополнительные табличные пространства и наполнение
- Создать новые табличные пространства для временных объектов:
    - $HOME/u03/dir7;
    - $HOME/u04/dir8.
- На основе template1 создать новую базу — crazyprog5.
- 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.

Выполнение

Подключение к узлу

```
ssh -p 2222 <isu>@se.ifmo.ru
ssh postgres2@pg107
```

Инициализация кластера БД

PGDATA

```
export PGDATA="$HOME/u05/dir5"
```

ENCODING, LOCALE

Для установки локали и кодировки анализируются переменные среды в приведённом ниже порядке до тех пор, пока одна из них не окажется заданной: `LC_ALL`, `LC_COLLATE` (или переменная, относящаяся к соответствующей категории), `LANG`.

```
export LANG="ru_RU.ISO8859-5"
```

После было все изменено:

```
[postgres2@pg107 ~]$ locale
LANG=ru_RU.ISO8859-5
LC_CTYPE="ru_RU.ISO8859-5"
LC_COLLATE="ru_RU.ISO8859-5"
LC_TIME="ru_RU.ISO8859-5"
LC_NUMERIC="ru_RU.ISO8859-5"
LC_MONETARY="ru_RU.ISO8859-5"
LC_MESSAGES="ru_RU.ISO8859-5"
LC_ALL=
```

Имя пользователя

Совпадает с текущем пользователем системы

```
uid=772(postgres2) gid=770(postgres) groups=770(postgres)
```

Команда

```
initdb
```

Конфигурация и запуск сервера БД

Передача конфигурационных файлов через scp

На узле нет vim'a, а работать в vi некомфортно. Поэтому передает на гелиос конфигурационные файлы:

```
scp postgres2@pg107:u05/dir5/pg_hba.conf .
scp postgres2@pg107:u05/dir5/postgresql.conf .
```

Обратно отправляем на узел:

```
scp pg_hba.conf postgres2@pg107:u05/dir5/pg_hba.conf
scp postgresql.conf postgres2@pg107:u05/dir5/postgresql.conf
```

pg_hba.conf (auth)

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only (without hostname)
local   all             all                                     password # plain
password
```

```
host    all             all             all                     password # plain
password
# 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

порт и порты слушания

```
listen_addresses = '*'                  # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' for
all
                                        # (change requires restart)
port = 9035                             # (change requires restart)
```

max_connections

Если мы увеличиваем max_connections, то, во-первых, для этого должны быть причины (у нас их нет), во-вторых, для этого соответственно необходимо увеличивать параметры `shared_buffers` и `kernel.shmmax`, поэтому мы менять ничего не будем.

shared_buffers

Резонно увеличивать до значения, равного ¼ от общего числа оперативной памяти.

```
shared_buffers = 4096 MB
```

temp_buffers

В настоящее время используется только для хранения временных таблиц в памяти. Если ваше приложение требует интенсивного использования временных таблиц, то вы можете значительно увеличить этот параметр. Надо

быть осторожным, поскольку это не разделяемая оперативная память, она выделяется на каждую сессию.

```
temp_buffers = 32MB                     # min 800kB
```

work_mem

Его размер применяется к каждой сортировке, выполняемой каждым пользователем, и сложные запросы могут использовать несколько буферов сортировки рабочей памяти. Установив его на 50 МБ, при наличии 30 пользователей, отправляющих запросы, и будем использовать 1,5 ГБ реальной памяти.

Были выставлены на субъективный взляд оптимальные значения. Если потребуется увеличить сильнее, то следует это сделать на уровне сессии.

```
work_mem = 32MB                         # min 64kB
```

checkpoint_time

Оставлен по умолчанию.

```
#checkpoint_timeout = 5min              # range 30s-1d
```

effective_cache_size

Этот параметр дает оценку общего объема памяти, доступной для дискового кэширования. Это не точный размер кеша, а ориентировочный. `effective_cache_size` не выделяет память, вместо этого он информирует оптимизатора об уровне кэша, доступного в ядре. Если это значение установлено низким, то планировщик запросов будет автоматически воздерживаться от использования определенных индексов, а иногда эти индексы бывают полезны.

По тому же эмпирическому правилу, по которому `shared_b uffers` занимают 25% системной памяти, `effective_cache_size` должен составлять от 50 до 75% оперативной памяти.

```
effective_cache_size = 8GB
```

fsync

Если этот параметр установлен, сервер PostgreSQL старается добиться, чтобы изменения были записаны на диск физически, выполняя системные вызовы fsync() или другими подобными методами. Это даёт гарантию, что кластер баз данных сможет вернуться в согласованное состояние после сбоя оборудования или операционной системы.

Хотя отключение fsync часто даёт выигрыш в скорости, это может привести к неисправимой порче данных в случае отключения питания или сбоя системы. Поэтому отключать fsync рекомендуется, только если вы легко сможете восстановить всю базу из внешнего источника, поэтому оставляем его включенным.

```
#fsync = on                             # flush data to disk for crash safety
                                        # (turning this off can cause
                                        # unrecoverable data corruption)
```

commit_delay

Параметр commit_delay добавляет паузу (в микросекундах) перед собственно выполнением сохранения WAL. Эта задержка может увеличить быстродействие при фиксировании множества транзакций, позволяя зафиксировать большее число транзакций за одну операцию сохранения WAL, если система нагружена достаточно сильно и за заданное время успевают зафиксироваться другие транзакции. Поэтому данный параметр может быть выставлен только когда будет ясен характер использования бд.

```
#commit_delay = 0                       # range 0-100000, in microseconds
```

log file format

```
log_destination = 'syslog'
```

log level

```
log_min_messages = error                # values in order of decreasing detail:
```

log connections and checkpoints

```
log_checkpoints = on
```

```
log_connections = on
#log_disconnections = off
#log_duration = off
#log_error_verbosity = default          # terse, default, or verbose messages
#log_hostname = off
```

Изменение расположения wal файлов:

1) Stop the running cluster

```
pg_ctl stop -mf # stop fast
```

2) Create a new directory

```
mkdir -p $HOME/u05/dir6
```

3) Copy existing files and directory

```
cp -rf $HOME/u05/dir5/pg_wal/*  $HOME/u05/dir6
```

4) Create symbolic link to new directory by renaming existing directory.

```
# move to temporary directory existing files
mkdir -p $HOME/u05/bck
mv $HOME/u05/dir5/pg_wal $HOME/u05/bck/pg_wal
# create a link
ln -s $HOME/u05/dir6 $HOME/u05/dir5/pg_wal
```

Result:

```
lrwxr-xr-x   1 postgres2  postgres     26 26  10:55 pg_wal ->
/var/db/postgres2/u05/dir6
```

Дополнительные табличные пространства и наполнение

Запуск БД и подключение

```
pg_ctl start -l $HOME/u05/pg.log
```

Сначала установить `trust` in `pg_hba.conf` и обновить дефолтный пароль для postgresql2. Затем мы можем подключаться (зависит от того где мы это делаем):

```
psql -p 9035 -h pg107 -d template1 -U postgres2
# or
psql -p 9035 -d template1 -U postgres2
```

Создание tablespaces для временных объектов

Создаем предварительно директории:

```
mkdir -p $HOME/u03/dir7
mkdir -p $HOME/u04/dir8
```

Создание табличных пространств от администратора:

```
template1=# create tablespace tbstmp1 location '/var/db/postgres2/u03/dir7';
CREATE TABLESPACE
template1=# create tablespace tbstmp2 location '/var/db/postgres2/u04/dir8';
CREATE TABLESPACE
```

```
template1=# select * from pg_catalog.pg_tablespace ;
  oid  |  spcname   | spcowner | spcacl | spcoptions
-------+------------+----------+--------+------------
  1663 | pg_default |       10 |        |
  1664 | pg_global  |       10 |        |
 16385 | tbstmp1    |       10 |        |
 16386 | tbstmp2    |       10 |        |
(4 строки)
```

Добавление табличных пространств в temp_tablespaces

Через alter system:

```
template1=# alter system set temp_tablespaces = 'tbstmp1,tbstmp2';
ALTER SYSTEM
template1=# select pg_reload_conf();
```

Также можно изменять через параметр в postgresql.conf

```
template1=# create temporary table tmp4 as select * from generate_series(1,1000000);
SELECT 1000000
template1=# create temporary table tmp5 as select * from generate_series(1,1000000);
SELECT 1000000
template1=# \db+
                                    Список табличных пространств
    Имя     | Владелец |        Расположение         | Права доступа | Параметры | Размер | Описание
------------+----------+-----------------------------+---------------+-----------+--------+----------
↳ -
 pg_default | postgres2 |                             |               |           | 33 MB  |
 pg_global  | postgres2 |                             |               |           | 560 kB |
 tbstmp1    | postgres2 | /var/db/postgres2/u03/dir7  |               |           | 35 MB  |
 tbstmp2    | postgres2 | /var/db/postgres2/u04/dir8  |               |           | 35 MB  |
(4 строки)

template1=# \dt+
                                    Список отношений
   Схема   | Имя  |   Тип   | Владелец | Хранение | Метод доступа | Размер | Описание
-----------+------+---------+----------+----------+---------------+--------+----------
 pg_temp_3 | tmp4 | таблица | postgres2 | временное | heap          | 35 MB  |
 pg_temp_3 | tmp5 | таблица | postgres2 | временное | heap          | 35 MB  |
(2 строки)
```

Создание базы данных и пользователя

```
template1=# create database crazyprog5 with template = template1;
CREATE DATABASE
template1=# create role s312563 login password 'pass';
CREATE ROLE
```

```
\du+
 Имя роли  |                                Атрибуты                                 | Член ролей | Описание
-----------+-------------------------------------------------------------------------+------------+----------
 postgres2 | Суперпользователь, Создаёт роли, Создаёт БД, Репликация, Пропускать RLS | {}         |
 s312563   |
```

Также добавляем привилегии, чтобы пользователь мог использовать данную бд:

```
grant create on tablespace tbstmp1, tbstmp2 to s312563;
```

Вставка значений в бд

```
psql -p 9035 -h pg107 -d crazyprog5 -W

create temp table tmp1 as
    (select
        i as first,
        i::text as second,
        i::text as third
    from generate_series(1,1000000) i);

create temp table tmp2 as
    (select
        i as first,
        i::text as second,
        now() as third
    from generate_series(1,10000) i);

insert into tmp2 (first,second,third) select i,i::text,now() from
generate_series(1,1000000) i;
```

Исполним команды выше от суперпользователя и обычного пользователя для демонстрации работоспособности.

От админа (postgres2)

```
                                  Список табличных пространств
    Имя     | Владелец |        Расположение         | Права доступа | Параметры | Размер | Описание
------------+----------+-----------------------------+---------------+-----------+--------+----------
 pg_default | postgres2 |                             |               |           | 34 MB  |
 pg_global  | postgres2 |                             |               |           | 560 kB |
 tbstmp1    | postgres2 | /var/db/postgres2/u03/dir7  |               |           | 49 MB  |
 tbstmp2    | postgres2 | /var/db/postgres2/u04/dir8  |               |           | 50 MB  |
(4 строки)

crazyprog5=# \dt+
                                      Список отношений
   Схема   | Имя  |   Тип   | Владелец  | Хранение  | Метод доступа | Размер | Описание
-----------+------+---------+-----------+-----------+---------------+--------+----------
 pg_temp_3 | tmp1 | таблица | postgres2 | временное | heap          | 49 MB  |
 pg_temp_3 | tmp2 | таблица | postgres2 | временное | heap          | 50 MB  |
(2 строки)
```

От обычного пользователя

```
crazyprog5=⇒ \dt+
                                      Список отношений
   Схема   | Имя  |   Тип   | Владелец | Хранение  | Метод доступа | Размер | Описание
-----------+------+---------+----------+-----------+---------------+--------+----------
 pg_temp_3 | tmp1 | таблица | s312563  | временное | heap          | 49 MB  |
 pg_temp_3 | tmp2 | таблица | s312563  | временное | heap          | 50 MB  |
(2 строки)
```

Access tablespaces from admin (because user doesn't have access).

```
crazyprog5=# \db+
                                  Список табличных пространств
    Имя     | Владелец |        Расположение         |      Права доступа      | Параметры | Размер | Описание
------------+----------+-----------------------------+-------------------------+-----------+--------+----------
 pg_default | postgres2 |                             |                         |           | 34 MB  |
 pg_global  | postgres2 |                             |                         |           | 560 kB |
 tbstmp1    | postgres2 | /var/db/postgres2/u03/dir7  | postgres2=C/postgres2+  |           | 49 MB  |
            |           |                             | s312563=C/postgres2     |           |        |
 tbstmp2    | postgres2 | /var/db/postgres2/u04/dir8  | postgres2=C/postgres2+  |           | 50 MB  |
            |           |                             | s312563=C/postgres2     |           |        |
(4 строки)
```

Список табличных пространств и объектов в них

`select relname, spcname from pg_class join pg_tablespace on pg_class.reltablespace = pg_tablespace.oid;`

Воспользуемся командой выше для демонстрации результатов от разных пользователей.

При вставке от суперпользователя (postgres2)

```
crazyprog5=# select relname, spcname from pg_class jo
                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_roiident_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_toast_16484                         | tbstmp1
 pg_toast_16484_index                   | tbstmp1
 tmp1                                   | tbstmp1
 pg_toast_16489                         | tbstmp2
 pg_toast_16489_index                   | tbstmp2
 tmp2                                   | tbstmp2
 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
(49 строк)
```

При вставке от обычного пользователя

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_roiident_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
tmp1	tbstmp1
pg_toast_16548	tbstmp1
pg_toast_16548_index	tbstmp1
pg_toast_16553	tbstmp2
pg_toast_16553_index	tbstmp2
tmp2	tbstmp2
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

Вывод

В ходе выполнения работы был сконфигурирован кластер баз данных под требуемые характеристики аппаратного обеспечения. Были изучены способы подключения и настройки инстанса PostgreSQL: права доступа пользователей, создание пользователей, настройка параметров в postgresql.conf, изучены способы подключений с параметром хоста и без. Кроме того, были созданы различные объекты базы данных: временные объекты и табличные пространства, базы данных.

Во время выполнения задания возникли трудности с использованием табличных пространств под временные объекты, а именно с их правами пользования для юзеров. Также с настройкой параметров через alter system + pg_reload_conf(), система отказывалась синхронизироваться с postgresql.conf() и вносить в него изменения, но может быть это фишка, или автор невнимательный.