

Университет ИТМО
Факультет ФПИ и КТ
Р33131

Отчет
по лабораторной работе №4
«Распределенные системы хранения
данных»
Вариант 20

Студент:

Ляо Ихун

Гр.Р33131

Предподаватель:

Шешуков Дмитрий Михайлович

Задание:

Лабораторная работа включает настройку резервного копирования данных с основного узла на резервный, а также несколько сценариев восстановления. Узел из предыдущей лабораторной работы используется в качестве основного; новый узел используется в качестве резервного. В сценариях восстановления необходимо использовать копию данных, полученную на первом этапе данной лабораторной работы.

Требования к отчёту

Отчет должен быть самостоятельным документом (без ссылок на внешние ресурсы), содержать всю последовательность команд, содержимое скриптов по каждому пункту задания. Для демонстрации результатов приводить команду вместе с выводом (самой наглядной частью вывода, при необходимости).

1. Резервное копирование

1.1 Настроить резервное копирование с основного узла на резервный следующим образом:

Периодические полные копии с помощью SQL Dump

По расписанию (cron) раз в сутки, методом SQL Dump с сжатием. Созданные архивы должны сразу перемещаться на резервных хост, они не должны храниться на основной системе. Срок хранения архивов на резервной системе - 4 недели. По истечении срока хранения, старые архивы должны автоматически уничтожаться.

1.2 Подсчитать, каков будет объем резервных копий спустя месяц работы системы, исходя из следующих условий:

Средний объем измененных данных за сутки: ~750 МБ.

1.3 Проанализировать результаты.

2. Потеря основного узла

Этот сценарий подразумевает полную недоступность основного узла. Необходимо восстановить работу СУБД на резервном узле, продемонстрировать успешный запуск СУБД и доступность данных.

3. Повреждение файлов БД

Этот сценарий подразумевает потерю данных (например, в результате сбоя диска или файловой системы) при сохранении доступности основного узла. Необходимо выполнить полное восстановление данных из резервной копии и перезапустить СУБД на основном узле.

Ход работы:

3.1 Симулировать сбой:

удалить с диска директорию любого табличного пространства со всем содержимым.

3.2 Проверить работу СУБД, доступность данных, перезапустить СУБД, проанализировать результаты.

3.3 Выполнить восстановление данных из резервной копии, учитывая следующее условие:

Исходное расположение дополнительных табличных пространств недоступно - разместить в другой директории и скорректировать конфигурацию.

3.4 Запустить СУБД, проверить работу и доступность данных, проанализировать результаты.

4. Логическое повреждение данных

Этот сценарий подразумевает частичную потерю данных (в результате нежелательной или ошибочной операции) при сохранении доступности основного узла. Необходимо выполнить восстановление данных на основном узле следующим способом:

Восстановление с использованием архивных WAL файлов. (СУБД должна работать в режиме архивирования WAL, потребуется задать параметры восстановления).

Ход работы:

4.1 В каждую таблицу базы добавить 2-3 новые строки, зафиксировать результат.

4.2 Зафиксировать время и симулировать ошибку:

Перезаписать строки любой таблицы "мусором" (INSERT, UPDATE)

4.3 Продемонстрировать результат.

4.4 Выполнить восстановление данных указанным способом.

4.5 Продемонстрировать и проанализировать результат.

Резервное копирование

Средний объем измененных данных за сутки: ~750 МБ. Считаю это изменение файла sql.

объем резервных копий спустя месяц работы системы:

$700\text{МБ} \times 30 + \text{исходной размер} = 21\text{ГБ} + \text{исходной размер}$

Здесь сохраняем его архив, поэтому объем меньше выше.

Создание копии

```
[postgres0@pg112 ~/backup]$ pg_dump -U postgres -h localhost -p 9048 -Fc postgres | gzip > $HOME/backup/mydatabase_backup.gz
П а р о л ь:
[postgres0@pg112 ~/backup]$ ls
logfile      mydatabase_backup.gz
[postgres0@pg112 ~/backup]$ rsync -avz $HOME/backup/mydatabase_backup.gz postgres3@pg124:$HOME/backup/
Password for postgres3@pg124.cs.ifmo.ru:
sending incremental file list
rsync: [Receiver] ERROR: cannot stat destination "/var/db/postgres0/backup/": Permission denied (13)
rsync error: errors selecting input/output files, dirs (code 3) at main.c(770) [Receiver=3.2.7]
[postgres0@pg112 ~/backup]$ rsync -avz $HOME/backup/mydatabase_backup.gz postgres3@pg124:backup/
Password for postgres3@pg124.cs.ifmo.ru:
sending incremental file list
mydatabase_backup.gz
sent 1.293 bytes  received 35 bytes  241.45 bytes/sec
total size is 1.161  speedup is 0.87
[postgres0@pg112 ~/backup]$ ls
logfile      mydatabase_backup.gz
[postgres0@pg112 ~/backup]$
```

```
[postgres3@pg124 ~]$ ls
mydatabase_backup.sql.gz
[postgres3@pg124 ~]$
```

Создание скрипта выполнения создания копии и её отправления

```
[postgres0@pg112 ~/lab3]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/var/db/postgres0/.ssh/id_rsa): ./final-ssh
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ./final-ssh.
Your public key has been saved in ./final-ssh.pub.
The key's fingerprint is:
SHA256:VoEvVM/gGeHhgtPg8H4BEuFlnGJ9JlhevKhrhUq8j/w postgres0@pg112.cs.ifmo.ru
The key's randomart image is:
+----[RSA 4096]-----+
|
|.o+..ooo
|.o=.o.o.
|..o.o+...
|o.o.+=.
|.=ooS=
|. =+o*
|.o..+ *
|o.+ o o
|.=+E
+----[SHA256]-----+
[postgres0@pg112 ~/lab3]$ ls
final-ssh      final-ssh.pub  lab.sh
[postgres0@pg112 ~/lab3]$ vi lab.sh
pg_dump -U postgres0 -h localhost -p 9048 -Fc postgres | gzip > $HOME/backup/mydatabase.sql.gz

rsync -avz -e "ssh -i ./final-ssh" $HOME/backup/mydatabase.sql.gz postgres3@pg124:backup/

rm $HOME/backup/mydatabase.sql.gz
```

```
[postgres0@pg112 ~/lab3]$ touch ~/.pgpass
[postgres0@pg112 ~/lab3]$ vi ~/.pgpass
```

```
total size is 1.185 speedup is 6.07
[postgres0@pg112 ~/lab3]$ cat ~/.pgpass
localhost:9048:postgres:postgres0:liao
```

```
[postgres0@pg112 ~/lab3]$ ./lab.sh
sending incremental file list
mydatabase.sql.gz

sent 1.288 bytes  received 47 bytes  2.670,00 bytes/sec
total size is 1.159  speedup is 0,87
[postgres0@pg112 ~/lab3]$
```

Создание crontab

```
[postgres0@pg112 ~]$ crontab -e
crontab: no crontab for postgres0 - using an empty one
```

```
0 0 * * * $HOME/lab.sh
```

Создание crontab и скрипта для удаления просрочной копии

```
[postgres3@pg124 ~]$ cat lab.sh
find $HOME/backup -name "*.sql.gz" -type f -mtime +28 -exec rm {} \;
```

```
[postgres3@pg124 ~]$ crontab -e
```

```
0 0 * * * $HOME/lab.sh
```

Потеря основного узла

Имитация полной потери узлов

Мы можем делаем восстановление базы данных на резервном узле чтобы делать имитацию полной потери узлов.

```
[postgres3@pg124 ~]$ pg_restore -U postgres -d postgres -p 5432 -C -v $HOME/backup/mydatabase.sql
pg_restore: подключение к базе данных для восстановления
pg_restore: создается DATABASE "postgres"
pg_restore: при обработке оглавления:
pg_restore: из записи оглавления 3684; 1262 14115 DATABASE postgres postgres0
pg_restore: ошибка: could not execute query: [ ] "postgres"
Выполнялась команда: CREATE DATABASE postgres WITH TEMPLATE = template0 ENCODING = 'ISO_8859_5' LOCALE = 'ru_RU.ISO8859-5';

pg_restore: ошибка: could not execute query: [ ] "postgres0"
Выполнялась команда: ALTER DATABASE postgres OWNER TO postgres0;

pg_restore: подключение к новой базе данных "postgres"
pg_restore: создается COMMENT "DATABASE postgres"
pg_restore: создается TABLE "public.test1"
pg_restore: из записи оглавления 212; 1259 16398 TABLE test1 newrole
pg_restore: ошибка: не удалось задать для default_tablespace значение space0: [ ] "default_tablespace": "space0"
ПОДРОБНОСТИ:
pg_restore: ошибка: could not execute query: [ ] "newrole"
Выполнялась команда: ALTER TABLE public.test1 OWNER TO newrole;

pg_restore: создается SEQUENCE "public.test1_id_seq"
pg_restore: из записи оглавления 211; 1259 16397 SEQUENCE test1_id_seq newrole
pg_restore: ошибка: could not execute query: [ ] "newrole"
Выполнялась команда: ALTER TABLE public.test1_id_seq OWNER TO newrole;

pg_restore: создается SEQUENCE OWNED BY "public.test1_id_seq"
pg_restore: создается TABLE "public.test2"
pg_restore: из записи оглавления 210; 1259 16391 TABLE test2 newrole
pg_restore: ошибка: could not execute query: [ ] "newrole"
Выполнялась команда: ALTER TABLE public.test2 OWNER TO newrole;

pg_restore: создается SEQUENCE "public.test2_id_seq"
pg_restore: из записи оглавления 209; 1259 16390 SEQUENCE test2_id_seq newrole
pg_restore: ошибка: could not execute query: [ ] "newrole"
Выполнялась команда: ALTER TABLE public.test2_id_seq OWNER TO newrole;

pg_restore: создается SEQUENCE OWNED BY "public.test2_id_seq"
pg_restore: создается DEFAULT "public.test1_id"
pg_restore: создается DEFAULT "public.test2_id"
pg_restore: обрабатываются данные таблицы "public.test1"
pg_restore: обрабатываются данные таблицы "public.test2"
pg_restore: выполняется SEQUENCE SET test1_id_seq
pg_restore: выполняется SEQUENCE SET test2_id_seq
pg_restore: создается CONSTRAINT "public.test1_test1_pkey"
pg_restore: создается CONSTRAINT "public.test2_test2_pkey"
```

```
postgres=# \d
          Список отношений
 С х е м а | И м я | Т и п | В л а д е л е ц
-----+-----+-----+-----
 public   | test1 | та б л и ц а | postgres3
 public   | test1_id_seq | п о с л е д о в а т е л ь н о с т ь | postgres3
 public   | test2 | та б л и ц а | postgres3
 public   | test2_id_seq | п о с л е д о в а т е л ь н о с т ь | postgres3
(4 строки)
```

```
[postgres3@pg124 ~]$ psql -p 5432 -U postgres3 -d postgres -h localhost
psql (14.2)
```

Видимо что там все информации о содержимом и структуре данных сохраняются. Но информации о tablespaces и роли нет.

Это из-за того что мы использовали pg_dump. pg_dumpall

решает эту проблему и мы будем его использовать в следующих этапах.

Повреждение файлов БД

Dump_all

1. `pg_dumpall -h localhost -p 9048 -U postgres0 | gzip >`

`$HOME/backupall/all.sql.gz`

2. `rsync -avz -e "ssh -i ./final-ssh"`

`$HOME/backupAll/mydatabase_backup.sql.gz`

`postgres3@pg124:backupall/`

3. Scp [postgres3@pg124:backupall/all.sql.gz](#)

`$HOME/backall/all.sql.gz`

Добавляем те команды в `lab.sh` и получаем новую копию, которая имеет информацию о `tablespace`.

```
Password for postgres3@pg124.cs.ifmo.ru:
all.sql 100% 1571 169.9KB/s 00:00
[postgres0@pg112 ~/backupall]$ ls
all.sql
```


Перед удалением

```
postgres=# \db+
      Имя      | Владелец | Список табличных пространств |
      +-----+-----+-----+-----+-----+-----+
      pg_default | postgres0 | /var/db/postgres0/u03/tp1 |
      pg_global  | postgres0 | /var/db/postgres0/u100 |
      space0     | postgres0 |
      space114   | postgres0 |
      (4 строки)

postgres=# select * from test1;
 id | val
----+----
(0 строк)

postgres=# \q
postgres0@pg112 ~]$ rm -rf ./u03/tp1
postgres0@pg112 ~]$ psql -p 9048 -U postgres -d postgres -h localhost
psql (14.2)
Введите "help", чтобы получить справку.

postgres=# \db+
      Имя      | Владелец | Список табличных пространств |
      +-----+-----+-----+-----+-----+-----+
      pg_default | postgres0 | /var/db/postgres0/u03/tp1 |
      pg_global  | postgres0 | /var/db/postgres0/u100 |
      space0     | postgres0 |
      space114   | postgres0 |
      (4 строки)
```

После удаления

```
postgres=# create table test3(id serial primary key, val int) tablespace space0;
ОШИБКА: не удалось создать каталог "pg_tblspc/16408/PG_14_202107181/14115": No such file or directory
postgres=# create table test3(id serial primary key, val int) tablespace space0;
ОШИБКА: не удалось создать каталог "pg_tblspc/16408/PG_14_202107181/14115": No such file or directory
postgres=#

postgres=# select tablename, tablespace from pg_tables where tablespace is not null;
  tablename  | tablespace
-----+-----
 test2       | space0
 test1       | space0
 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
(12 строк)

postgres=# \db+
      Имя      | Владелец | Список табличных пространств |
      +-----+-----+-----+-----+-----+-----+
      pg_default | postgres0 | /var/db/postgres0/u03/tp1 |
      pg_global  | postgres0 | /var/db/postgres0/u100 |
      space0     | postgres0 |
      space1     | postgres0 |
      space114   | postgres0 |
      (5 строки)

postgres=# select * from test1;
postgres=# ;
ОШИБКА: не удалось открыть файл "pg_tblspc/16387/PG_14_202107181/14115/16398": No such file or directory
postgres=# select * from test1;
postgres=# ;
ОШИБКА: не удалось открыть файл "pg_tblspc/16387/PG_14_202107181/14115/16398": No such file or directory
postgres=#
```

Восстановление

```
[postgres@pg112 ~]$ cat backupall/*.sql | grep create
[postgres@pg112 ~]$ cat backupall/*.sql | grep "space0"
CREATE TABLESPACE space0 OWNER postgres0 LOCATION '/var/db/postgres0/u03/tp1';
GRANT ALL ON TABLESPACE space0 TO newrole;
SET default_tablespace = space0;
-- Name: test1; Type: TABLE; Schema: public; Owner: newrole; Tablespace: space0
-- Name: test2; Type: TABLE; Schema: public; Owner: newrole; Tablespace: space0
[postgres@pg112 ~]$

postgres=# drop tablespace space0;
ПРЕДУПРЕЖДЕНИЕ: не удалось открыть каталог "pg_tblspc/16408/PG_14_202107181": No such
DROP TABLESPACE
postgres=# drop tablespace space0;
ОШИБКА: табличное пространство "space0" не существует
postgres=# drop tablespace space0;
ОШИБКА: табличное пространство "space0" не существует
postgres=# \q
[postgres@pg112 ~/u01/dir1]$ psql -d postgres -U postgres0 -p 9048 -h localhost -f $HOME/backupall/all.sql
SET
SET
SET
psql:/var/db/postgres0/backupall/all.sql:14: ☐ "newrole"
ALTER ROLE
psql:/var/db/postgres0/backupall/all.sql:16: ☐ "postgres0"
ALTER ROLE
psql:/var/db/postgres0/backupall/all.sql:26: ☐ "space114"
CREATE TABLESPACE
Пароль пользователя postgres0:
Вы подключены к базе данных "template1" как пользователь "postgres0".

[postgres@pg112 ~/u03/tp1]$ psql -d postgres -U postgres0 -p 9048 -h localhost -f $HOME/backupall/all.sql
SET
SET
SET
psql:/var/db/postgres0/backupall/all.sql:14: ☐ "newrole"
ALTER ROLE
psql:/var/db/postgres0/backupall/all.sql:16: ☐ "postgres0"
ALTER ROLE
psql:/var/db/postgres0/backupall/all.sql:26: ☐ "space114"
psql:/var/db/postgres0/backupall/all.sql:27: ☐ "space0"
Пароль пользователя postgres0:
Вы подключены к базе данных "template1" как пользователь "postgres0".

SET
ALTER TABLE
[postgres@pg112 ~/u01/dir1]$ psql -p 9048 -U postgres0 -d postgres -h localhost
psql (14.2)
Введите "help", чтобы получить справку.

postgres=# select * from test1;
 id | val
----+----
(0 строк)
```

Важно что нужно предварительно создать все катологии, где сохраняются все пространства таблицы, и вручную удалять все таблицы и пространства в БД.

Логическое повреждение данных

```
#wal_level = replica
```

```
# - Archiving -
archive_mode = on                # enables archiving; off, on, or always
                                # (change requires restart)
archive_command = 'cp %p $HOME/pg/archive/%f' # command to use to archive a logfile segment
                                                # placeholders: %p = path of file to archive
                                                # %f = file name only
                                                # e.g. 'test ! -f /mnt/server/archivedir/%f && cp %p /mnt/server/archivedir/%f'
archive_timeout = 10             # force a logfile segment switch after this
                                # number of seconds; 0 disables
```

```
postgres=# select current_timestamp;
          current_timestamp
-----
2023-05-29 22:06:26.251315+03
(1 строка)

postgres=# select * from test;
 id | val
----+----
(0 строка)

postgres=# insert into test(val) values (1), (2);
INSERT 0 2
postgres=# select * from test;
 id | val
----+----
  1 |   1
  2 |   2
(2 строки)

postgres=# select current_timestamp;
          current_timestamp
-----
2023-05-29 22:06:45.342688+03
(1 строка)
```

pg_ctl -D \$HOME/u01/dir1 -l \$HOME/logfile stop

```
archive_mode = off                # enables archiving; off, on, or always
                                # (change requires restart)
archive_command = 'cp %p $HOME/u01/archive/%f' # command to use to archive a logfile segment
                                                # placeholders: %p = path of file to archive
                                                # %f = file name only
                                                # e.g. 'test ! -f /mnt/server/archivedir/%f && cp %p /mnt/server/archivedir/%f'
archive_timeout = 10             # force a logfile segment switch after this
                                # number of seconds; 0 disables

# - Archive Recovery -
# These are only used in recovery mode.

restore_command = 'cp $HOME/u01/archive/%f %p' # command to use to restore an archived logfile segment
                                                # placeholders: %p = path of file to restore
                                                # %f = file name only
```

touch ./recovery.signal

rm -rf pg_wal/*

pg_ctl -D \$HOME/u01/dir1 -l \$HOME/logfile start

```
[postgres0@pg112 ~/u01]$ psql -p 5432 -U postgres0 -d postgres -h localhost
psql (14.2)
Введите "help", чтобы получить справку.

postgres=# select * from test;
 id | val
----+-----
(0 строк)
```

Вывод

В ходе выполнении работы изучал как создать резервное копирование при использовании sql dump. Успешно создал скрипт, который автоматически периодически выполняется каждый день, при помощи ssh. Успешно восстановил бд при использовании wal файл.