Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

Дисциплина: Распределенные системы хранения данных

# Лабораторная работа 3

Вариант: 10

Выполнил:

Гурьянов Кирилл Алексеевич

Группа: P33302

Преподаватель:

Шешуков Дмитрий Михайлович

Санкт-Петербург

2024

# 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.

**1. Резервное копирование**

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

Периодические обособленные (standalone) полные копии.

Полное резервное копирование (pg_basebackup) по расписанию (cron) два раза в сутки. Необходимые файлы WAL должны быть в составе полной копии, отдельно их не архивировать. Срок хранения копий на основной системе - 1 неделя, на резервной - 1 месяц. 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: ~300 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 конфигурационных файлов СУБД 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:

В любой таблице с внешними ключами изменить внешние ключи случайным образом (INSERT, UPDATE)

4.3 Продемонстрировать результат.

4.4 Выполнить восстановление данных указанным способом.

4.5 Продемонстрировать и проанализировать результат.

# Выполнение

## Резервное копирование

### Настройка копирования

Для создании резервной копии по расписанию был создан следующий скрипт, который будет запускаться на основном узле 2 раза в сутки в 6 утра и 18 вечера:

```
#!/usr/local/bin/bash

BACKUPS_DIR=$HOME/backups
DATE=$(date "+%Y-%m-%d-%H:%M:%S")
BACKUP_DIR_NAME=backup_${DATE}
BACKUP_DIR=$BACKUPS_DIR/$BACKUP_DIR_NAME

BACKUPS_TABLESPACES_DIR=$HOME/backups_tablespaces
BACKUP_TABLESPACE_DIR=$BACKUPS_TABLESPACES_DIR/backup_${DATE}

pg_basebackup -p 9512 -h pg198 -U postgres3 -D $BACKUP_DIR
--tablespace-mapping=$HOME/zgf65=$BACKUP_TABLESPACE_DIR -X s

scp -r $BACKUP_DIR
postgres6@pg128:~/korako2_dont_delete_pls/backups/"$BACKUP_DIR_NAME"
scp -r $BACKUP_TABLESPACE_DIR
postgres6@pg128:~/korako2_dont_delete_pls/backups_tablespaces
```

```
find $BACKUPS_DIR -name "*" -type d -mtime +7 -mindepth 1 -maxdepth 1 -exec rm
-rf {} \;
find $BACKUPS_TABLESPACES_DIR -name "*" -type d -mtime +7 -mindepth 1 -maxdepth
1 -exec rm -rf {} \;

ssh postgres6@pg128 "bash ~/korako2_dont_delete_pls/clear.sh"
```

На резервном узле был создал скрипт clear.sh:

```
#!/usr/local/bin/bash

BACKUPS_DIR=$HOME/korako2_dont_delete_pls/backups
BACKUPS_TABLESPACES_DIR=$HOME/korako2_dont_delete_pls/backups_tablespaces

find $BACKUPS_DIR -name "*" -type d -mtime +31 -mindepth 1 -maxdepth 1 -exec rm
-rf {} \;
find $BACKUPS_TABLESPACES_DIR -name "*" -type d -mtime +31 -mindepth 1 -maxdepth
1 -exec rm -rf {} \;
```

Для запуска скрипта по расписанию был создан cron-файл, который описывает правило запуска скрипта:

```
0 6 18 * * * $HOME/backup.sh >> backup.log
```

Проверка работоспособности скрипты

```
bash backup.sh
```

```
[postgres3@pg198 ~]$ ls backups
backup_2024-04-08-22:35:30      backup_2024-04-08-23:18:47
backup_2024-04-08-22:37:28      backup_2024-04-08-23:20:58
backup_2024-04-08-22:40:37      backup_2024-04-08-23:21:57
backup_2024-04-08-22:59:27      backup_2024-04-09-12:20:16
backup_2024-04-08-23:02:18      backup_2024-04-09-12:29:11
backup_2024-04-08-23:02:59      backup_2024-04-09-12:46:23
backup_2024-04-08-23:06:46      backup_2024-04-09-12:47:43
backup_2024-04-08-23:10:18      backup_2024-04-09-13:51:20
backup_2024-04-08-23:11:53      backup_2024-04-09-14:03:07
backup_2024-04-08-23:13:33      backup_2024-04-09-14:22:18
backup_2024-04-08-23:14:47      backup_2024-04-09-14:50:59
[postgres3@pg198 ~]$ ls backups_tablespaces/
backup_2024-04-08-22:35:30      backup_2024-04-08-23:18:47
backup_2024-04-08-22:37:28      backup_2024-04-08-23:20:58
backup_2024-04-08-22:40:37      backup_2024-04-08-23:21:57
backup_2024-04-08-22:59:27      backup_2024-04-09-12:20:16
backup_2024-04-08-23:02:18      backup_2024-04-09-12:29:11
backup_2024-04-08-23:02:59      backup_2024-04-09-12:47:43
backup_2024-04-08-23:06:46      backup_2024-04-09-13:51:20
backup_2024-04-08-23:10:18      backup_2024-04-09-14:03:07
backup_2024-04-08-23:11:53      backup_2024-04-09-14:22:18
backup_2024-04-08-23:13:33      backup_2024-04-09-14:50:59
backup_2024-04-08-23:14:47
```

### Подсчет объема копий

Начальный размер бэкапа составляет примерно $a_1 = 10$ МБ. Размер копий увеличивается каждые сутки на 300 МБ. Будем считать, что добавление данных распределено равномерно в течении дня, таким образом $d = 150$ МБ. Всего за месяц копирования (30 дней) будет создано $n = 60$ копий, т.к. копирование производится 2 раза в сутки. Подсчитаем размер последней копии: $a_{60} = a_1 + (n-1)d = 10 + 59 * 150 = 8860$ МБ.

Подсчитаем суммарный вес всех копий за месяц:

$$S = \frac{a_1+a_n}{2} * n = \frac{10+8860}{2} * 60 = 266100 \text{ МБ} \approx 221 \text{ ГБ}$$

### Анализ результатов

Использование полных копий является эффективным способом резервного копирования в связи с простотой восстановления данных. Однако мы можем видеть, что размер копий растет в арифметической прогрессии с течением времени, из-за чего размер копий базы данных после месяца копирования, достигает существенных размеров. Решением здесь может являться механизм инкрементальных копий, благодаря которым можно сэкономить место на диске.

# Потеря основного узла

Производим копирование файлов кластера из бэкапов в PGDATA.

```
cp -r ~/backups/backup_2024-03-27-01\:52\:44/ ~/jcf50/
cp -r ~/backups_tablespaces/backup_2024-03-27-01\:52\:44/ ~/zgf65/
```

Меняем символическую ссылку на табличные пространства.

```
lrwxr-xr-x    1 postgres6   postgres   47   9 апр.   12:12 16401 ->
/var/db/postgres6/korako2_dont_delete_pls/zgf65
```

Проверяем работоспособность кластера.

```
psql -h pg128 -U postgres3 -d postgres -p 9512

Пароль пользователя postgres3:
```

```
postgres=# \c lazybluearmy

Вы подключены к базе данных "lazybluearmy" как пользователь
"postgres3".

lazybluearmy=# select * from army LIMIT 10;
id |                   country                    |  size
---+----------------------------------------------+---------
 1 | Китай                                        | 2035000
 2 | Индия                                        | 1460350
 3 | Соединённые Штаты Америки                    | 1395350
 4 | Корейская Народно-Демократическая Республика | 1280000
 5 | Россия                                       |  900000
 6 | Пакистан                                     |  651800
 7 | Иран                                         |  610000
 8 | Республика Корея                             |  555000
 9 | Вьетнам                                      |  482000
10 | Египет                                       |  438500
(10 строк)

lazybluearmy=# \db
                    Список табличных пространств
   Имя     | Владелец  |                  Расположение
-----------+-----------+-----------------------------------------
--------
indexspace               |               postgres3               |
/var/db/postgres6/korako2_dont_delete_pls/zgf65
pg_default | postgres3 |
pg_global  | postgres3 |
(3 строки)
```

Кластер восстановлен, никакие данные не были потеряны.

## Повреждение файлов БД

### Симуляция сбоя

Удаляем все конфигурационные файлы из PGDATA.

```
[postgres3@pg198 ~/jcf50]$ rm postgresql.conf postgresql.auto.conf
pg_hba.conf pg_ident.conf
```

Проверяем работоспособность кластера.

```
[postgres3@pg198 ~]$ psql -p 9512 -d postgres -h pg198 -U postgres3
Пароль пользователя postgres3:
psql (14.2)
```

```
Введите "help", чтобы получить справку.

postgres=#
\q
```

Как можем видеть, работа кластера не нарушена, т.к. во время потери конфигурационных файлов он был запущен.

Перезапускаем кластер баз данных.

```
[postgres3@pg198 ~]$ pg_ctl -D /var/db/postgres3/jcf50 stop
ожидание завершения работы сервера....... готово
сервер остановлен
[postgres3@pg198 ~]$ pg_ctl -D /var/db/postgres3/jcf50 start
ожидание запуска сервера....postgres не может открыть файл конфигурации сервера "/var/db/post
gres3/jcf50/postgresql.conf": No such file or directory
прекращение ожидания
pg_ctl: не удалось запустить сервер
Изучите протокол выполнения.
```

Как можем видеть, запустить кластер без конфигурационных файлов не получилось. Приступим к восстановлению.

## Восстановление

```
[postgres6@pg128 ~]$ cd jcf50
```

Копируем с резервного узла конфигурационные файлы на основной узел.

```
[postgres6@pg128 ~]$ scp postgresql.conf postgresql.auto.conf pg_hba.conf pg_ident.conf postgres3@pg198:~/jcf50
postgresql.conf                                               100%
28KB  37.5MB/s   00:00
postgresql.auto.conf                                          100%
88   582.8KB/s   00:00
pg_hba.conf                                                   100%
4863    20.1MB/s   00:00
pg_ident.conf                                                 100%
1636     9.6MB/s   00:00

[postgres3@pg198 ~]$ cd jcf50/
[postgres3@pg198 ~/jcf50]$ ls
base                pg_multixact        PG_VERSION
current_logfiles    pg_notify           pg_wal
global              pg_replslot         pg_xact
```

```
log                                                     pg_serial
postgresql.auto.conf
logfile                 pg_snapshots            postgresql.conf
pg_commit_ts            pg_stat                 postmaster.opts
pg_dynshmem             pg_stat_tmp             postmaster.pid
pg_hba.conf             pg_subtrans             vi.core
pg_ident.conf           pg_tblspc
pg_logical              pg_twophase
```

Копируем с резервного узла табличное прастрансво на основной узел.

```
scp -r zgf65/PG_14_202107181/ postgres3@pg198:~/zgf66/
16435                                                   100%
16KB  25.7MB/s   00:00
```

Меняем символическую ссылку в pg_tblspc на zgf66.

```
[postgres3@pg198 ~]$ ls -l
total 1
lrwxr-xr-x   1 postgres3   postgres   23 27 марта 02:37 16434 ->
/var/db/postgres3/zgf66
```

## Проверка работоспособности и доступности данных

```
[postgres3@pg198 ~]$ pg_ctl -D /var/db/postgres3/jcf50 start
ожидание запуска сервера....2024-03-27 02:37:46.720 MSK [59777]
LOG:  redirecting log output
to logging collector process
2024-03-27 02:37:46.720 MSK [59777] HINT:  Future log output will
appear in directory "log".
готово
сервер запущен
[postgres3@pg198 ~]$ psql -p 9512 -d postgres -h pg198 -U postgres3
Пароль пользователя postgres3:
psql (14.2)
Введите "help", чтобы получить справку.

postgres=# \c lazybluearmy
Вы подключены к базе данных "lazybluearmy" как пользователь
"postgres3".
lazybluearmy=# \dt
         Список отношений
Схема  | Имя  |   Тип   | Владелец
--------+------+---------+-----------
public | army | таблица | postgres3
```

```
(1 строка)

lazybluearmy=# \db
          Список табличных пространств
   Имя      | Владелец  |      Расположение
------------+-----------+------------------------
indexspace | postgres3 | /var/db/postgres3/zgf66
pg_default | postgres3 |
pg_global  | postgres3 |
(3 строки)
```

Анализ результатов

В ходе восстановления на основной узел были возвращены конфигурационные файлы с резервного узла из копии кластера. Также была устранена потеря дополнительных табличных пространств. Директория дополнительных табличных пространств также была скопирована на основной узел. Запуск кластера был произведен успешно. Никаких потерь данных не было замечено, доступность данных была восстановлена.

## Логическое повреждение данных

Внесем изменения в postgresql.conf:
wal_level = replica
archive_mode=on
archive_command = 'scp %p
postgres6@pg128:~/korako2_dont_delete_pls/wal128/%f'

```
lazybluearmy=# select * from armies;
 id | country |  size
----+---------+--------
 1 |       2 | 123123
 2 |       6 |   1234
 3 |       1 |   9999
 4 |       3 |   1312
 5 |       4 |  12312
 6 |       5 |  10000
 7 |       7 |   5555
 8 |       8 |   6666
(8 строк)


lazybluearmy=# select * from country;
 id |      name
```

```
----+----------------
 1 | Германия
 2 | Россия
 3 | Новая Гвинея
 4 | Великобритания
 5 | Никарагуа
 6 | Новая Зеландия
 7 | Грузия
 8 | Норвегия
(8 строк)
```

## Добавление новых строк в таблицы

```
insert into country values (9, 'Казахстан');
INSERT 0 2
lazybluearmy=# insert into armies values (9, 9, 55555);
INSERT 0 2
lazybluearmy=# select * from armies;
id | country |  size
----+---------+--------
 1 |       2 | 123123
 2 |       6 |   1234
 3 |       1 |   9999
 4 |       3 |   1312
 5 |       4 |  12312
 6 |       5 |  10000
 7 |       7 |   5555
 8 |       8 |   6666
 9 |       9 |  55555
(8 строк)

lazybluearmy=# select * from country;
id |      name
----+----------------
 1 | Германия
 2 | Россия
 3 | Новая Гвинея
 4 | Великобритания
 5 | Никорагуа
 6 | Новая Зеландия
 7 | Грузия
 8 | Норвегия
 9 | Казахстан
(8 строк)
```

## Фиксация времени и симуляция ошибки

```
lazybluearmy=# select now();
              now
-------------------------------
2024-04-09 14:51:45.261491+03

(1 строка)

lazybluearmy=# update armies set country = 1 where country = 5;
UPDATE 1
lazybluearmy=# update armies set country = 1 where country = 8;
UPDATE 1
lazybluearmy=# select * from armies;
id | country |  size
----+---------+--------
 1 |       2 | 123123
 2 |       6 |   1234
 3 |       1 |   9999
 4 |       3 |   1312
 5 |       4 |  12312
 7 |       7 |   5555
 6 |       1 |  10000
 8 |       1 |   6666
 9 |       9 |  55555
(8 строк)
```

Остановка кластера и восстановление из WAL файлов

```
[postgres3@pg198 ~]$ pg_ctl stop
ожидание завершения работы сервера.... готово
сервер остановлен
[postgres3@pg198 ~]$ rm -rf pg_wal_back/*
```

Копируем WAL файлы на случай, если там имеются не заархивированные файлы.

```
[postgres3@pg198 ~]$ cp -r jcf50/pg_wal/ pg_wal_back/
```

Удаляем файлы PGDATA и табличных пространств.

```
[postgres3@pg198 ~]$ rm -rf jcf50/* zgf65/*
```

Восстанавливаем кластер из резервной копии.

```
[postgres3@pg198 ~]$ cp -r backups/backup_2024-04-09-14\:50\:59/ jcf50/
[postgres3@pg198 ~]$ cp -r backups_tablespaces/backup_2024-04-09-14\:50\:59/ zgf65
[postgres3@pg198 ~]$ rm -rf ~/jcf50/pg_tblspc/16401
[postgres3@pg198 ~]$ ln -s /var/db/postgres3/zgf65 /var/db/postgres3/jcf50/pg_tblspc/16401
```

Очищаем pg_wal и копируем в него незаархивированные WAL файлы.

```
[postgres3@pg198 ~]$ rm -rf jcf50/pg_wal/*
[postgres3@pg198 ~]$ cp -r pg_wal_back/* jcf50/pg_wal/
[postgres3@pg198                ~/jcf50/pg_wal]$                rm
000000010000000000000002A.00000028.backup
[postgres3@pg198 ~/jcf50/pg_wal]$ rm 000000010000000000000002B
```

Вносим изменения в postgresql.conf:

restore_command = 'scp postgres6@pg128:~/korako2_dont_delete_pls/wal128/%f %p'
recovery_target_time = '2024-04-09 14:51:45.261491+03'
recovery_target_inclusive = off

Создаем файлы recovery.signal и запускаем кластер в режиме восстановления.

```
[postgres3@pg198 ~]$ touch jcf50/recovery.signal

[postgres3@pg198 ~]$ pg_ctl start
ожидание запуска сервера....2024-04-09 14:56:15.319 MSK [11668]
LOG:  redirecting log output
to logging collector process
2024-04-09 14:56:15.319 MSK [11668] HINT:  Future log output will
appear in directory "log".
. готово
сервер запущен
```

Содержимое логов:

```
2024-04-09 14:56:15.525 MSK [11670] LOG:  starting point-in-time recovery to
2024-04-09 14:51
:45.261491+03
2024-04-09 14:56:16.870 MSK [11670] LOG:  restored log file
"000000010000000000000002A" from a
rchive
2024-04-09 14:56:16.972 MSK [11670] LOG:  redo starts at 0/2A000028
2024-04-09 14:56:16.982 MSK [11670] LOG:  consistent recovery state reached at
0/2A000100
2024-04-09 14:56:16.982 MSK [11668] LOG:  database system is ready to accept
read-only connec
tions
2024-04-09 14:56:17.232 MSK [11670] LOG:  restored log file
"000000010000000000000002B" from a
rchive
2024-04-09 14:56:17.272 MSK [11670] LOG:  recovery stopping before commit of
```

```
transaction 747,
time 2024-04-09 14:52:03.468871+03
2024-04-09 1r4:56:17.272 MSK [11670] LOG:  pausing at the end of recovery
2024-04-09 14:56:17.272 MSK [11670] HINT:  Execute pg_wal_replay_resume() to
promote.
```

## Демонстрация результата

**[postgres3@pg198 ~]$ psql -p 9512 -h pg198 -U postgres3 -d lazybluearmy**

Как мы можем видеть, данные, которые были добавлены до точки времени, были восстановлены. Изменения данных, которые производились через update, не были применены.

```
lazybluearmy=# select * from armies;
id | country |  size
----+---------+--------
 1 |       2 | 123123
 2 |       6 |   1234
 3 |       1 |   9999
 4 |       3 |   1312
 5 |       4 |  12312
 6 |       5 |  10000
 7 |       7 |   5555
 8 |       8 |   6666
 9 |       9 |  55555
(9 строк)

lazybluearmy=# select * from country;
id |      name
----+----------------
 1 | Германия
 2 | Россия
 3 | Новая Гвинея
 4 | Великобритания
 5 | Никорагуа
 6 | Новая Зеландия
 7 | Грузия
 8 | Норвегия
 9 | Казахстан
(9 строк)
```

# Вывод

В ходе выполнения лабораторной работы я познакомился с механизмами резервного копирования и репликации. Мною был на практики опробован способ резервирования путем создания полных копий. Также были созданы разные виды сбоев и произведены операции по устранению сбоев и восстановлению доступности баз данных.