

Факультет Программной Инженерии и Компьютерной техники

Распределенные системы хранения данных
Лабораторная работа №3
Вариант 696

Выполнили:
Калабухов М. А.
Коновалов А. А.
Группа Р33131
Преподаватель:
Афанасьев Д.Б.

Санкт-Петербург
2024

Оглавление

Оглавление	2
Задание	3
Этап 1. Резервное копирование.....	3
Этап 2. Потеря основного узла.....	3
Этап 3. Повреждение файлов БД.....	3
Этап 4. Логическое повреждение данных.....	4
Ход работы	5
Этап 1. Резервное копирование:.....	5
На основном узле был создан юзер admin с правами суперпользователя.....	5
Скрипт на основном узле:.....	5
Расчеты:.....	7
Этап 3. Сценарий: Повреждение данных DB	8
Этап 4. Сценарий: Логическое повреждение данных.....	10
Вывод	11

Задание

Этап 1. Резервное копирование

- Настроить резервное копирование с основного узла на резервный следующим образом:
Периодические холодные полные копии.
Полная копия (rsync) по расписанию (cron) раз в сутки. СУБД на время копирования должна останавливаться. На резервном узле хранить 14 копий, после успешного создания пятнадцатой копии, самую старую автоматически уничтожать.
- Подсчитать, каков будет объем резервных копий спустя месяц работы системы, исходя из следующих условий:
 - Средний объем новых данных в БД за сутки: 550МБ.
 - Средний объем измененных данных за сутки: 500МБ.
- Проанализировать результаты.

Этап 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.

Этап 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:

- Симулировать сбой:
 - удалить с диска директорию любой таблицы со всем содержимым.
- Проверить работу СУБД, доступность данных, перезапустить СУБД, проанализировать результаты.
- Выполнить восстановление данных из резервной копии, учитывая следующее условие:
 - исходное расположение директории PGDATA недоступно - разместить данные в другой директории и скорректировать конфигурацию.
- Запустить СУБД, проверить работу и доступность данных, проанализировать результаты.

Этап 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:

- В каждую таблицу базы добавить 2-3 новые строки, зафиксировать результат.
- Зафиксировать время и симулировать ошибку:
 - перезаписать строки любой таблицы "мусором" (INSERT, UPDATE)
- Продемонстрировать результат.
- Выполнить восстановление данных указанным способом.
- Продемонстрировать и проанализировать результат.

Ход работы

Данные для подключения:

ssh -p 2222 s335094@se.ifmo.ru (pass:*****)

Узел 1 - ssh postgres1@pg108 (pass:*****)

Узел 2 - ssh postgres0@pg116 (pass:*****)

Этап 1. Резервное копирование:

На основном узле был создан юзер admin с правами суперпользователя.

Скрипт на основном узле:

```
#!/bin/bash
CURRENT_DATE=$(date "+%Y-%m-%d-%H:%M:%S")
BACKUP_DIR_NAME="backup_${CURRENT_DATE}"
pg_ctl stop -D $HOME/qkl81
rsync -avv $HOME $HOME/qkl81/pg_wal
postgres1@pg108:~/backups/$BACKUP_DIR_NAME
postgres -D $HOME/qkl81 >~/logfile 2>&1 &
ssh postgres1@pg108 "bash /var/db/postgres1/remove_script.sh"
echo "$(date): Backup $BACKUP_DIR_NAME was successfully created in directory
~/backups"
```

Скрипт на резервном узле:

```
#!/bin/bash
BACKUP_CNT=`ls -l /var/db/postgres1/backups | grep ^d | grep -c backup`
BACKUP_OLDEST=`ls -l /var/db/postgres1/backups | grep ^d | grep backup | awk
'{print $9}' | sort | head -1`
MAX_BACKUP_CNT=14

if (($BACKUP_CNT > $MAX_BACKUP_CNT)); then
```

```
    echo "$(date) : Backup count is $BACKUP_CNT. Remove the oldest one  
$BACKUP_OLDEST" >> back_log.log
```

```
    rm -rf /var/db/postgres1/backups/$BACKUP_OLDEST
```

```
else
```

```
    echo "$(date) : Backup count is $BACKUP_CNT" >> back_log.log
```

```
fi
```

Вывод при ручном использовании скрипта командой

```
bash backup_script.bash >> log_backup.log
```

```
[postgres0@pg116 ~]$ cat log_backup.log
```

ожидание завершения работы сервера.... готово

сервер остановлен

opening connection using: ssh -l postgres1 pg108 rsync --server -vvlogDtpre.iLsfxCIvu .

"~/backups/backup_2024--14-18:05:50" (9 args)

sending incremental file list

[sender] expand file_list pointer array to 1024 bytes, did move

[sender] expand file_list pointer array to 4096 bytes, did move

[sender] expand file_list pointer array to 1024 bytes, did move

[sender] expand file_list pointer array to 4096 bytes, did move

[sender] expand file_list pointer array to 1024 bytes, did move

[sender] expand file_list pointer array to 4096 bytes, did move

[sender] expand file_list pointer array to 1024 bytes, did move

[sender] expand file_list pointer array to 4096 bytes, did move

created directory /var/db/postgres1/backups/backup_2024--14-18:05:50

delta-transmission enabled

pg_wal/

pg_wal/00000000100000000000000000000001

pg_wal/archive_status/

postgres0/

postgres0/.bash_history

postgres0/.lesshst

postgres0/.psql_history

postgres0/backup_script.sh

postgres0/fill.sql

postgres0/log_backup.log

postgres0/logfile

postgres0/pg_hba.conf

postgres0/postgresql.conf

postgres0/script.sql

postgres0/.ssh/

postgres0/.ssh/id_rsa

postgres0/.ssh/id_rsa.pub

postgres0/.ssh/known_hosts

postgres0/kdu94/

postgres0/kdu94/Pg_14_202107181/

[sender] expand file_list pointer array to 1024 bytes, did move

[sender] expand file_list pointer array to 4096 bytes, did move

postgres0/kdu94/Pg_14_202107181/16396/

postgres0/kdu94/Pg_14_202107181/16396/112

postgres0/kdu94/PG_14_202107181/16396/113

...

postgres0/ygl69/PG_14_202107181/16397/pg_filenode.map

postgres0/ygl69/PG_14_202107181/16397/pg_internal.init

total: matches=0 hash_hits=0 false_alarms=0 data=96646364

sent 96.798.883 bytes received 42.142 bytes 64.560.683,33 bytes/sec

total size is 96.645.638 speedup is 1,00

вторник, 14 мая 2024 г. 18:05:52 (MSK): Backup backup_2024--14-18:05:50 was successfully created in directory ~/backups

Вывод на резервном узле:

[postgres1@pg108 ~]\$ cat back_log.log

вторник, 14 мая 2024 г. 17:42:34 (MSK) : Backup count is 1

вторник, 14 мая 2024 г. 17:43:08 (MSK) : Backup count is 2

вторник, 14 мая 2024 г. 17:43:26 (MSK) : Backup count is 3

вторник, 14 мая 2024 г. 17:43:31 (MSK) : Backup count is 4

вторник, 14 мая 2024 г. 17:43:49 (MSK) : Backup count is 5

вторник, 14 мая 2024 г. 17:44:18 (MSK) : Backup count is 6

вторник, 14 мая 2024 г. 17:44:22 (MSK) : Backup count is 7

вторник, 14 мая 2024 г. 17:44:27 (MSK) : Backup count is 8

вторник, 14 мая 2024 г. 17:44:37 (MSK) : Backup count is 9

вторник, 14 мая 2024 г. 17:44:42 (MSK) : Backup count is 10

вторник, 14 мая 2024 г. 17:46:48 (MSK) : Backup count is 11

вторник, 14 мая 2024 г. 17:46:53 (MSK) : Backup count is 12

вторник, 14 мая 2024 г. 17:47:06 (MSK) : Backup count is 13

вторник, 14 мая 2024 г. 17:47:11 (MSK) : Backup count is 14

вторник, 14 мая 2024 г. 17:47:15 (MSK) : Backup count is 15. Remove the oldest one backup_2024--14-17:42:30

вторник, 14 мая 2024 г. 17:47:24 (MSK) : Backup count is 15. Remove the oldest one backup_2024--14-17:43:06

вторник, 14 мая 2024 г. 18:05:52 (MSK) : Backup count is 15. Remove the oldest one backup_2024--14-17:43:23

вторник, 14 мая 2024 г. 18:09:07 (MSK) : Backup count is 15. Remove the oldest one backup_2024--14-17:43:28

Расчеты:

Этап 2. Сценарий 1. Потеря основного узла.

Представим, что наше физическое оборудование основного узла просто вынесли и в итоге у нас нет к нему доступа.

Заходим на наш резервный узел, и восстанавливаем последний бэкап нашей базы.

Т.к. у нас один узел postgres1, а второй postgres0 => нужно пофиксить симлинки, которые не совпадают. Делаем это следующими командами:

```
ln -s /var/db/postgres0/qkl81 ~/qkl81/pg_tblspc/*
```

-самый новый бэкап-

```
cp -r ./backups/backup_2024--14-17\46\49 ./
```

```
pg_ctl start -D qkl81/postgres0/qkl81
```

Вывод:

```
[postgres1@pg108 ~]$ pg_ctl start -D qkl81/postgres0[postgres1@pg108
~/backups]$ du -hs backup_2024--14-17:43:47
23M backup_2024--14-17:43:47
```

Итак, размер нашего бэкапа 23 МБ.

С помощью арифметической прогрессии вычислим дальнейшие вычисления:

$a_1 = 12 \text{ МБ}$

$d = 500 \text{ МБ}$

$n = 30 \text{ дней}$

S - частная формула для суммы

$S_n = (2 \cdot a_1 + d \cdot (n-1)) / 2 \cdot n$

$S_{30} = (2 \cdot 23 + 500 \cdot 29) / 2 \cdot 30 = 239940 \text{ МБ} \approx 234 \text{ GB}$

Также плодятся новые wal файлы по 16 МБ каждый. Т.к. изменений в сутки на 500 МБ => будет создано 32 wal файла.

За месяц таким образом мы получим: 238080 МБ wal файлов.

Итого: Собщ = 239940 МБ + 238080 МБ \approx 466 ГБ

/qkl81

ожидание запуска сервера....2024-05-14 19:45:57.271 MSK [42255] LOG: redirecting log output to logging collector process

2024-05-14 19:45:57.271 MSK [42255] HINT: Future log output will appear in directory "log".

готово

сервер запущен

Этап 3. Сценарий: Повреждение данных ~~DB~~.

```
[postgres0@pg116 ~/qkl81]$ psql -d lazybluelake1 -U admin -h pg116 -p 9696
```

Пароль пользователя admin:

psql (14.2)

Введите "help", чтобы получить справку.

lazybluelake1=# \d

Список отношений

Схема	Имя	Тип	Владелец
public	lake	таблица	s335094
public	lake_id_seq	последовательность	s335094

(2 строки)

lazybluelake1=# drop table lake;

DROP TABLE

lazybluelake1=#

Подтягиваем бэкап:


```
[postgres1@pg108 ~/backups]$ rsync -avv
$HOME/backups/backup_2024--14-19:48:20/postgres0/*
postgres0@pg116:~/
```

opening connection using: ssh -l postgres0 pg116 rsync --server -vvlogDtpre.iLsfxCIvu . "~/\" (9 args)

Password for postgres0@pg116.cs.ifmo.ru:

sending incremental file list

[sender] expand file_list pointer array to 1024 bytes, did move

[sender] expand file_list pointer array to 4096 bytes, did move

[sender] expand file_list pointer array to 1024 bytes, did move

[sender] expand file_list pointer array to 4096 bytes, did move

[sender] expand file_list pointer array to 1024 bytes, did move

[sender] expand file_list pointer array to 4096 bytes, did move

[sender] expand file_list pointer array to 1024 bytes, did move

[sender] expand file_list pointer array to 4096 bytes, did move

delta-transmission enabled

backup_script.sh is uptodate

fill.sql is uptodate

log_backup.log is uptodate

pg_hba.conf is uptodate

postgres.conf is uptodate

script.sql is uptodate

logfile

kdu94/Pg_14_202107181/16396/112 is uptodate

kdu94/Pg_14_202107181/16396/113 is uptodate

kdu94/Pg_14_202107181/16396/1247 is uptodate

kdu94/Pg_14_202107181/16396/1247_fsm is uptodate

...

qkl81/pg_stat/global.stat

qkl81/pg_stat_tmp/

total: matches=24 hash_hits=125 false_alarms=0 data=44386

sent 91.546 bytes received 101.537 bytes 77.233,20 bytes/sec

total size is 79.399.219 speedup is 411,22

Запуск сервера после восстановления:

```
[postgres0@pg116 ~]$ pg_ctl start -D qkl81
```

ожидание запуска сервера....2024-05-14 20:12:57.117 MSK [46250] LOG: redirecting log output to logging collector process

2024-05-14 20:12:57.117 MSK [46250] HINT: Future log output will appear in directory "log".

готово

сервер запущен

```
[postgres0@pg116 ~]$ psql -d lazybluelake1 -U admin -h pg116 -p 9696
```

Пароль пользователя admin:

psql (14.2)

Введите "help", чтобы получить справку.

```
lazybluelake1=# \dt
```

Список отношений

```
Схема | Имя | Тип | Владелец
```

```
-----+-----+-----+-----
```

```
public | lake | таблица | s335094
```

(1 строка)

```
lazybluelake1=#
```

Как можно увидеть, удаленная таблица была восстановлена.

Этап 4. Сценарий: Логическое повреждение данных.

заполним тестовыми данными таблицу.

```
lazybluelake1=# insert into lake (name, square) values ('bebra', 11), ('bobra', 12), ('dobre', 13);
```

```
INSERT 0 3
```

```
lazybluelake1=# select * from lake;
```

```
lazybluelake1=# select * from lake;
```

```
id | name | square
```

```
-----+-----+-----
```

```
1 | 1 | 1
```

```
2 | 2 | 2
```

```
3 | 3 | 3
```

```
4 | bebra | 11
```

```
5 | bobra | 12
```

```
6 | dobre | 13
```

(6 строк)

фиксируем ошибку и время ошибки

```
lazybluelake1=# select now();
```

```
now
```

```
-----
```

```
2024-05-15 12:15:22.101149+03
```

```
lazybluelake1=# delete from lake where (id%2)= 1;
```

```
DELETE 3
```

```
lazybluelake1=# select pg_switch_wal();
```

```
lazybluelake1=# exit
```

```
[postgres0@pg116 ~]$ pg_ctl stop -D qkl81
```

Перед этим вводим некоторые изменения в postgresql.conf:

```
archive_mode = on
```

```
archive_command = 'scp %p postgres1@pg108:~/wal/%f'
```

```
wal_level = replica
```

```
restore_command = 'scp postgres1@pg108:~/wal/%f %p'
```

Создаем recovery.signal и в postgresql.conf добавляем следующую строку:

```
recovery_target_time = '2024-05-15 13:27:00'
```

Сносим pg_wal, чтобы убедиться, что не рекаверим какую-то шляпу:

```
rm -rf qkl81/pg_wal/*
```

Запускаем сервер в recovery mode и в логах видим следующее:

```
2024-05-15 13:44:46.559 MSK [23704] LOG: starting PostgreSQL 14.2 on amd64-portbld-freebsd13.0,
compiled by FreeBSD clang version 11.0.1 (git@github.com:llvm/llvm-project.git llvmorg-11.0.1-0-g43ff75f2c3fe),
64-bit
2024-05-15 13:44:46.559 MSK [23704] LOG: listening on IPv6 address "::", port 9696
2024-05-15 13:44:46.559 MSK [23704] LOG: listening on IPv4 address "0.0.0.0", port 9696
2024-05-15 13:44:46.577 MSK [23706] LOG: database system was interrupted while in recovery at log time
2024-05-15 13:32:49 MSK
scp: /var/db/postgres1/wal/00000002.history: No such file or directory
2024-05-15 13:44:46.766 MSK [23706] LOG: starting point-in-time recovery to 2024-05-15 13:27:00+03
2024-05-15 13:44:47.058 MSK [23706] LOG: restored log file "00000001000000000000000004" from archive
```

Результат:

```
lazybluelake1=# select * from lake;
```

id	name	square
1	1	1
2	2	2
3	3	3
4	bebra	11
5	bobra	12
6	dobre	13

(6 строк)

Вывод

На протяжении 8 часов мы учились делать бекапы базы данных и восстанавливать её в различных ситуациях.