PostgreSQL 之 pg_rewind使用详解

[日期：2019-04-17] 来源：Linux社区 作者：fflixiang [字体：大 中 小]

pg_rewind

是postgresql主从数据库之同步数据目录的工具。需要目标服务器在postgresql.conf 中允许wal_log_hints，或者在 initdb初始化集群时允许 checksums ，full_page_writes也必须为on

pg_rewind只复制表数据文件中更改的块;所有其他文件都被完整复制，包括配置文件。pg_rewind相对于使用pg_basebackup备份或rsync等工具的优势在于，pg_rewind不需要读取数据库中未更改的块。这使得在数据库很大且之间只有一小部分块不同的情况下，速度会快得多。

pg_rewind [option...] { -D | --target-pgdata } directory { --source-pgdata=directory | --source-server=connstr

参数：

-D directory --target-pgdata=directory

此选项指定与源同步的目标数据目录。在运行pg_rewind之前，必须干净关闭目标服务器

--source-pgdata=directory

指定要与之同步的源服务器的数据目录的文件系统路径。此选项要求干净关闭源服务器

--source-server=connstr

指定要连接到源PostgreSQL服务器的libpq连接字符串。连接必须是具有超级用户访问权限的正常(非复制)连接。此选项要求源服务器正在运行，而不是处于恢复模式。

-n --dry-run

除了实际修改目标目录之外，执行所有操作。

-P --progress

使进展报告。

实验使用两台主机，都安装postgresql-10.7，已配置流复制

- 主：192.168.56.5 m1
- 从：192.168.56.25 m7

m1(主)：创建测试表和数据

```
postgres=# create table test (id int,e_name varchar(100),e_mail varchar(100),d_id int);
CREATE TABLE
postgres=# \d
List of relations
Schema | Name | Type | Owner
--------+------+-------+----------
public | test | table | postgres
(1 row)
postgres=# insert into test values(1,'zbs','123@126.com',10);
INSERT 0 1
postgres=# select * from test;
id | e_name | e_mail | d_id
----+--------+-------------+------
1 | zbs | 123@126.com | 10
(1 row)
```

m7 (从)：查询数据复制成功

```
[postgres@z_leader ~]$ psql postgres
psql (10.7)
Type "help" for help.
postgres=# \d
List of relations
Schema | Name | Type | Owner
```

```
---------+------+-------+----------
public | test | table | postgres
(1 row)
postgres=# select * from test;
id | e_name | e_mail | d_id
----+--------+-------------+------
1 | zbs | 123@126.com | 10
(1 row)
```

提升从库为新主库

```
[postgres@z_leader data]$ pg_ctl promote -D /usr/local/pg/data
waiting for server to promote.... done
server promoted
[postgres@z_leader data]$ psql postgres
psql (10.7)
Type "help" for help.
postgres=# select pg_is_in_recovery();
pg_is_in_recovery
-------------------
f
(1 row)
```

m1(原主库)插入一条记录，模拟原主库上的数据没有复制到原从库上

```
postgres=# insert into test values(2,'zbs1','124@126.com',10);
INSERT 0 1
postgres=# select * from test;
id | e_name | e_mail | d_id
----+--------+-------------+------
1 | zbs | 123@126.com | 10
2 | zbs1 | 124@126.com | 10
(2 rows)
```

m7：在原从库上(已提升为主库)插入一条记录并查看结果

```
postgres=# insert into test values(3,'zbs2','124@126.com',10);
INSERT 0 1
postgres=# select * from test;
id | e_name | e_mail | d_id
----+--------+-------------+------
1 | zbs | 123@126.com | 10
3 | zbs2 | 124@126.com | 10
(2 rows)
```

m1 将原主库变为新主库的从库

```
[postgres@localhost ~]$ kill -INT `head -1 /usr/local/pg/data/postmaster.pid`
```

--配置流复制文件和参数

```
[postgres@localhost data]$ mv recovery.done recovery.conf
[postgres@localhost data]$ cat recovery.conf
standby_mode = 'on'
restore_command = 'cp /usr/local/pg/arch/%f'
primary_conninfo = 'host=192.168.56.25 port=5432 user=rep'
recovery_target_timeline = 'latest'
[postgres@localhost data]$
```

--启动数据库

```
[postgres@localhost ~]$ /usr/local/pg/bin/pg_ctl -D /usr/local/pg/data -l logfile start
waiting for server to start.... done
server started
[postgres@localhost data]$ psql postgres
psql (10.7)
Type "help" for help.
postgres=# select pg_is_in_recovery();
pg_is_in_recovery
-------------------
t
(1 row)
postgres=# select * from test;
id | e_name | e_mail | d_id
----+--------+-------------+------
```

```
1 | zbs | 123@126.com | 10
2 | zbs1 | 124@126.com | 10
(2 rows)

--在m7上插入的记录未能复制过来

---日志信息

2019-03-02 09:15:17.415 CST [2492] LOG: consistent recovery state reached at 0/D000098
2019-03-02 09:15:17.415 CST [2492] LOG: invalid record length at 0/D000098: wanted 24, got 0
2019-03-02 09:15:17.415 CST [2490] LOG: database system is ready to accept read only connections
2019-03-02 09:15:17.429 CST [2500] LOG: fetching timeline history file for timeline 6 from primary server
2019-03-02 09:15:17.460 CST [2500] FATAL: could not start WAL streaming: ERROR: requested starting point 0/D000000 on timeline 5 is not in this
server's history
DETAIL: This server's history forked from timeline 5 at 0/C003168.
cp: missing destination file operand after `/usr/local/pg/arch/00000006.history'
Try `cp --help' for more information.
cp: missing destination file operand after `/usr/local/pg/arch/00000007.history'
Try `cp --help' for more information.
cp: missing destination file operand after `/usr/local/pg/arch/00000006.history'
Try `cp --help' for more information.
2019-03-02 09:15:17.469 CST [2492] LOG: new timeline 6 forked off current database system timeline 5 before current recovery point 0/D000098
cp: missing destination file operand after `/usr/local/pg/arch/0000000500000000000000D

[postgres@localhost ~]$ kill -INT `head -1 /usr/local/pg/data/postmaster.pid`

---使得pg_rewind 同步数据库时间线

[[postgres@localhost ~]$ pg_rewind --target-pgdata /usr/local/pg/data --source-server='host=192.168.56.25 port=5432 user=postgres dbname=postgres' -P
connected to server
servers diverged at WAL location 0/C003168 on timeline 5
rewinding from last common checkpoint at 0/C003010 on timeline 5
reading source file list
reading target file list
reading WAL in target
need to copy 100 MB (total source directory size is 118 MB)
102599/102599 kB (100%) copied
creating backup label and updating control file
syncing target data directory
Done!

--pg_rewind后此文件需要重新配置

[postgres@localhost data]$ cat recovery.conf
standby_mode = 'on'
restore_command = 'cp /usr/local/pg/arch/%f'
primary_conninfo = 'host=192.168.56.25 port=5432 user=rep'
recovery_target_timeline = 'latest'

[postgres@localhost ~]$ /usr/local/pg/bin/pg_ctl -D /usr/local/pg/data -l logfile start
waiting for server to start.... done
server started
[postgres@localhost ~]$ psql postgres
psql (10.7)
Type "help" for help.
postgres=# select * from test;
id | e_name | e_mail | d_id
----+--------+-------------+------
1 | zbs | 123@126.com | 10
3 | zbs2 | 124@126.com | 10
(2 rows)
postgres=# select pg_is_in_recovery();
pg_is_in_recovery
-------------------
t
(1 row)

--原主库没有复制到从库的记录消失，在新主库上插入的记录已同步

m7(新主库)
[postgres@z_leader ~]$ psql postgres
psql (10.7)
Type "help" for help.
postgres=# insert into test values(4,'zbs2','124@126.com',10);
INSERT 0 1
```

```
postgres=# select * from test;
id | e_name | e_mail | d_id
----+--------+-------------+------
1 | zbs | 123@126.com | 10
3 | zbs2 | 124@126.com | 10
4 | zbs2 | 124@126.com | 10
(3 rows)
```

m1(新丛库)

```
postgres=# select * from test;
id | e_name | e_mail | d_id
----+--------+-------------+------
1 | zbs | 123@126.com | 10
3 | zbs2 | 124@126.com | 10
4 | zbs2 | 124@126.com | 10
(3 rows)
```


监听故障TNS-12537: TNS:connection closed 解决对策 关于Oracle Job定时任务配置讲解

相关资讯 pg_rewind pg_rewind使用

本文评论 查看全部评论 (0)

表情： 姓名： 匿名 匿名 字数 0

同意评论声明 请登录

评论声明

- 尊重网上道德，遵守中华人民共和国的各项有关法律法规
- 承担一切因您的行为而直接或间接导致的民事或刑事法律责任
- 本站管理人员有权保留或删除其管辖留言中的任意内容
- 本站有权在网站内转载或引用您的评论
- 参与本评论即表明您已经阅读并接受上述条款