

PostgreSQL

控制文件作用与管理

Objectives



PolarDB



PostgreSQL

- 控制文件的结构
- 如何维护控制文件
- 无备份控制文件损坏恢复

控制文件的位置



PolarDB



- 控制文件位置与大小

逻辑位置：存放在pg_global表空间中

物理位置与名字：\$PGDATA/global/pg_control

控制文件的物理大小是8K

控制文件内容尽量保持小于512个字节

控制文件的内容

- initdb时生成的静态信息

```
pg_control version number:      1201
Catalog version number:        201909212
Database system identifier:     6744130188358323871
Maximum data alignment:        8
Database block size:           8192
Blocks per segment of large relation: 131072
WAL block size:                16384
Bytes per WAL segment:         16777216
Maximum length of identifiers:  64
Maximum columns in an index:    32
Maximum size of a TOAST chunk:  1996
Date/time type storage:        64-bit integers
Float4 argument passing:       by value
Float8 argument passing:       by value
```

控制文件的内容

- postgresql.conf中的配置信息

```
wal_level setting:          replica
wal_log_hints setting:      on
max_connections setting:    1000
max_worker_processes setting: 8
max_wal_senders setting:    10
max_prepared_xacts setting: 0
max_locks_per_xact setting: 64
track_commit_timestamp setting: off
```

控制文件的内容

- write-ahead logging以及checkpoint的动态信息

```
Latest checkpoint location:      2E/CD32D270
Latest checkpoint's REDO location: 2E/CD32D270
Latest checkpoint's REDO WAL file: 000000110000002E000000CD
Latest checkpoint's TimeLineID:   17
Latest checkpoint's PrevTimeLineID: 17
Latest checkpoint's full_page_writes: on
Latest checkpoint's NextXID:      0:538902
Latest checkpoint's NextOID:      385511
Latest checkpoint's NextMultiXactId: 5
.....
Time of latest checkpoint:        Sat 18 Jul 2020 10:11:14 PM EDT
Fake LSN counter for unlogged rels: 0/3E8
Backup start location:            0/0
Backup end location:              0/0
End-of-backup record required:    no
```

控制文件的维护

- 控制文件维护

- 1、固定部分
初始化数据库时产生，固定不变
- 2、有些信息随时更新
如果发生检查点、备份、日志切换等操作，则自动更新
- 3、`postgres.conf`相关参数被更新
如果配置文件中重要的相关参数被修改，则也会自动更新
- 4、数据库备份时会一起备份
- 5、不能手动修改该文件
- 6、启动和恢复数据库时需要，当前没有避免发生单一故障点而设计的保护策略

控制文件备份

- 备份方式（与数据库备份一起备份，一起恢复，不能单独备份）

1、tar命令

```
tar -jcv -f baseline.tar.bz2 $PGDATA
```

2、pg_basebackup命令

```
pg_basebackup -D bk1 -Ft -z -P
```

3、pg_rman命令

```
pg_rman backup --backup-mode=full -B /home/postgres/pg_rman_bk/ -C -P
```


控制文件恢复

- 重建控制文件

- Pg 10版本以前版本使用pg_resetxlog工具
- Pg10及以后版本使用pg_resetwal工具

pg_resetwal工具需要下面四个参数：

-x XID set next transaction ID

-m MXID set next and oldest multitransaction ID

-O OFFSET set next multitransaction offset

-I XLOGFILE force minimum WAL starting location for new transaction log

控制文件恢复

- 重建控制文件

-l XLOGFILE force minimum WAL starting location for new transaction log
(XLOGFILE强制新事务日志的最小WAL起始位置)

定位办法: pg_wal下面创建的最大的日志文件, 编号+1

```
cd /usr/local/pg12.2/data/pg_wal/
```

```
ll
```

```
-rw----- 1 postgres postgres    270 Mar 21 09:40 00000010.history
-rw----- 1 postgres postgres 16777216 Jul 19 06:19 0000001100000038000000C4
-rw----- 1 postgres postgres 16777216 Jul 19 06:16 0000001100000038000000C5
-rw----- 1 postgres postgres 16777216 Jul 19 06:17 0000001100000038000000C6
-rw----- 1 postgres postgres 16777216 Jul 19 06:17 0000001100000038000000C7
```

确定-l 的参数值为:

```
-l 0000001100000038000000C8
```

控制文件恢复

- 重建控制文件

-O OFFSET set next multitransaction offset

在pg_multixact/members下面，取最大值加1然后乘以65536.转换成16进制为100000,然后末尾添加4个0

-O=0x1000000000

步骤:

1、进入相应目录

```
cd $PGDATA/pg_multixact/members/
```

2、查看文件

```
ll
```

```
-rwxr-x--- 1 postgres postgres 8192 Jul 18 22:11 0000
```

3、计算结果

-O=0x1000000000

控制文件恢复

- 重建控制文件

- m =MXID,MXID set next and oldest multitransaction ID

在pg_multixact/offsets下面，找到最大的文件编号，+1 后面跟上4个0
如：0000

-m =0x00010000,0x00010000

步骤：

1、进入相应目录

```
cd $PGDATA/pg_multixact/offsets
```

2、查看文件

```
-rwxr-x--- 1 postgres postgres 8192 Jul 19 05:57 0000
```

3、计算结果

```
-m=0x00010000,0x00010000
```

控制文件恢复

- 重建控制文件

- x XID set next transaction ID

在pg_xact下面，找到最大的文件编号，+1 后面跟上5个0
如：0000

-x = 0x000100000

步骤：

1、进入相应目录

```
cd $PGDATA/pg_xact/
```

2、查看当前文件

```
ll
```

```
total 140
```

```
-rwxr-x--- 1 postgres postgres 139264 Jul 18 23:21 0000
```

3、计算-x值

-x=0x000100000

控制文件恢复

- 重建控制文件

- 执行重建控制文件语句:

```
cd $PGDATA/global
```

```
touch pg_control
```

```
pg_resetwal -l 0000001100000038000000CC -O 0x1000000000 \  
-m 0x00010000,0x00010000 -x 0x000100000 -f $PGDATA
```

```
pg_controldata
```

- 启动数据库:

```
pg_ctl start
```

总结



PolarDB



PostgreSQL

- 控制文件的结构
- 如何维护控制文件
- 无备份控制文件损坏恢复

练习

- 1、在数据库中创建一些表, 写入一些数据库, 执行一些更新操作. 记录下所有的记录值.
- 2、执行检查点
- 3、查询数据库的控制文件内容
- 4、删除控制文件
- 5、关闭数据库实例
- 6、启动数据库实例, 观察报什么错
- 7、使用pg_resetwal重建控制文件, 应该设置哪些值? 这些值如何计算?
- 8、启动实例, 观察控制文件的内容与之前有哪些变化?
- 9、观察数据是否还在数据库实例中? 有没有丢失?

