轻松上手openGauss

[高校课堂]

——openGauss备份恢复

贾军锋

2021.12.13

Gauss松鼠会 openGauss

目录

01 备份类型介绍

02 逻辑备份恢复

03 物理备份恢复

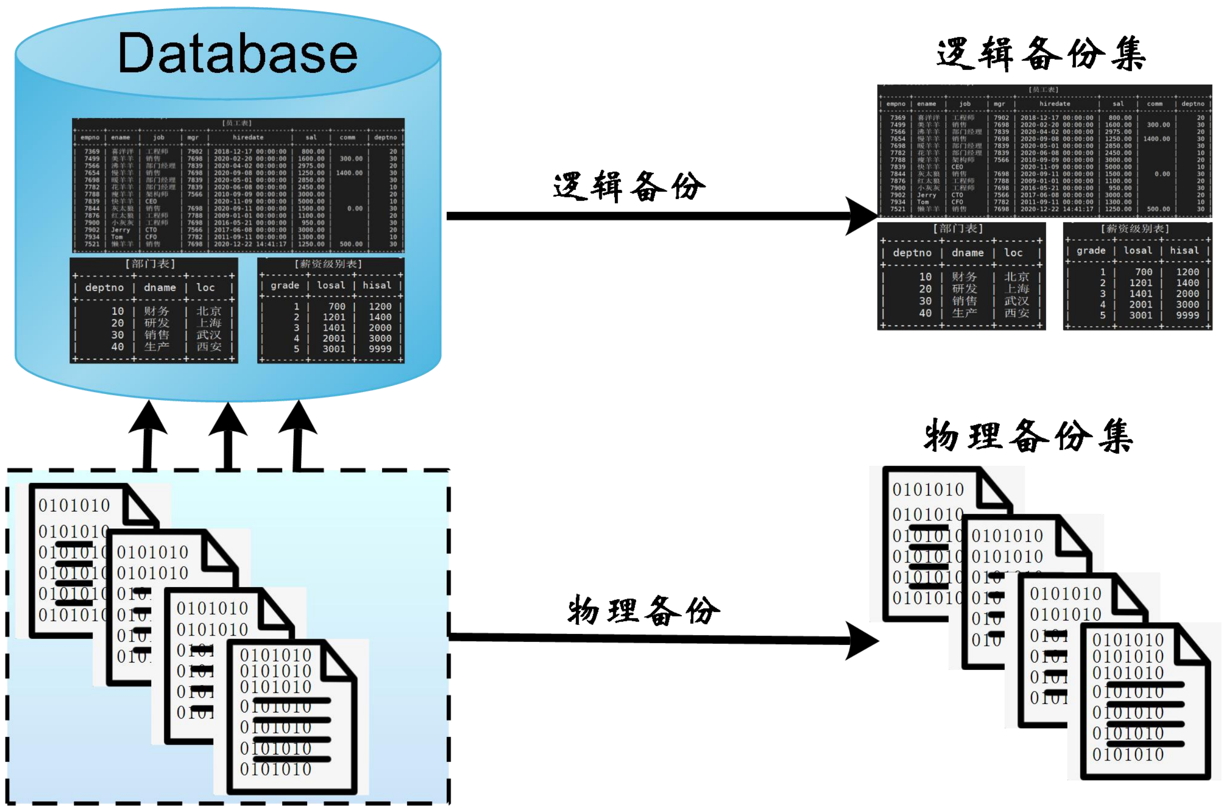
04 二进制程序备份恢复

PART ONE

备份类型介绍

备份类型介绍

数据备份是保证数据安全的重要手段之一，为了更好的保证数据安全，openGauss数据库支持**逻辑备份**和**物理备份**两种备份类型。



```
                                    [员工表]
+-------+--------+----------+------+---------------------+---------+---------+--------+
| empno | ename  |   job    | mgr  |      hiredate       |   sal   |  comm   | deptno |
+-------+--------+----------+------+---------------------+---------+---------+--------+
|  7369 | 喜洋洋 | 工程师   | 7902 | 2018-12-17 00:00:00 |  800.00 |         |     20 |
|  7499 | 美羊羊 | 销售     | 7698 | 2020-02-20 00:00:00 | 1600.00 |  300.00 |     30 |
|  7566 | 沸羊羊 | 部门经理 | 7839 | 2020-04-02 00:00:00 | 2975.00 |         |     20 |
|  7654 | 慢羊羊 | 销售     | 7698 | 2020-09-08 00:00:00 | 1250.00 | 1400.00 |     30 |
|  7698 | 暖羊羊 | 部门经理 | 7839 | 2020-05-01 00:00:00 | 2850.00 |         |     30 |
|  7782 | 花羊羊 | 部门经理 | 7839 | 2020-06-08 00:00:00 | 2450.00 |         |     10 |
|  7788 | 瘦羊羊 | 架构师   | 7566 | 2010-09-09 00:00:00 | 3000.00 |         |     20 |
|  7839 | 快羊羊 | CEO      |      | 2020-11-09 00:00:00 | 5000.00 |         |     10 |
|  7844 | 灰太狼 | 销售     | 7698 | 2020-09-11 00:00:00 | 1500.00 |    0.00 |     30 |
|  7876 | 红太狼 | 工程师   | 7788 | 2009-01-01 00:00:00 | 1100.00 |         |     20 |
|  7900 | 小灰灰 | 工程师   | 7698 | 2016-05-21 00:00:00 |  950.00 |         |     30 |
|  7902 | Jerry  | CTO      | 7566 | 2017-06-08 00:00:00 | 3000.00 |         |     20 |
|  7934 | Tom    | CFO      | 7782 | 2011-09-11 00:00:00 | 1300.00 |         |     10 |
|  7521 | 懒羊羊 | 销售     | 7698 | 2020-12-22 14:41:17 | 1250.00 |  500.00 |     30 |
+-------+--------+----------+------+---------------------+---------+---------+--------+

         [部门表]                         [薪资级别表]
+--------+-------+------+        +-------+-------+-------+
| deptno | dname | loc  |        | grade | losal | hisal |
+--------+-------+------+        +-------+-------+-------+
|     10 | 财务  | 北京 |        |     1 |   700 |  1200 |
|     20 | 研发  | 上海 |        |     2 |  1201 |  1400 |
|     30 | 销售  | 武汉 |        |     3 |  1401 |  2000 |
|     40 | 生产  | 西安 |        |     4 |  2001 |  3000 |
+--------+-------+------+        |     5 |  3001 |  9999 |
                                 +-------+-------+-------+
```

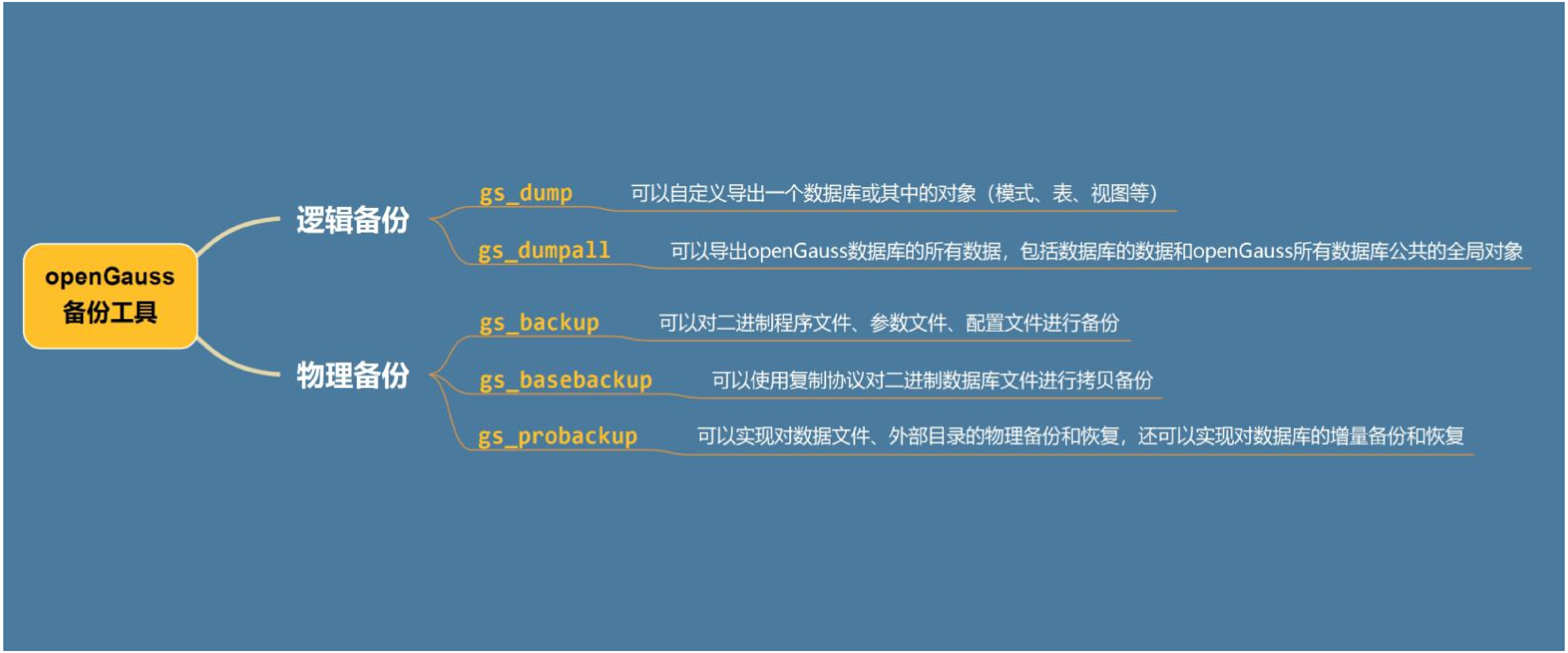
备份方案考虑要素：

- 备份对业务的影响在可接受范围内
- 数据库恢复效率
- 数据可恢复程度
- 数据库备份成本

➢两种备份恢复类型对比

备份类型	应用场景	支持的介质	优缺点
逻辑备份与恢复	适合于数据量小的场景。 目前用于表备份恢复，可以备份恢复单表和多表。	• 机械磁盘 • SSD	可按用户需要进行指定对象的备份和恢复，灵活度高。 当数据量大时，备份效率低。
物理备份与恢复	适用于数据量大的场景，主要用于全量数据备份恢复，也可对整个数据库中的WAL归档日志和运行日志进行备份恢复。		数据量大时，备份效率高。

openGauss备份工具概览



openGauss 备份工具

逻辑备份

gs_dump 可以自定义导出一个数据库或其中的对象（模式、表、视图等）

gs_dumpall 可以导出openGauss数据库的所有数据，包括数据库的数据和openGauss所有数据库公共的全局对象

物理备份

gs_backup 可以对二进制程序文件、参数文件、配置文件进行备份

gs_basebackup 可以使用复制协议对二进制数据库文件进行拷贝备份

gs_probackup 可以实现对数据文件、外部目录的物理备份和恢复，还可以实现对数据库的增量备份和恢复

PART TWO

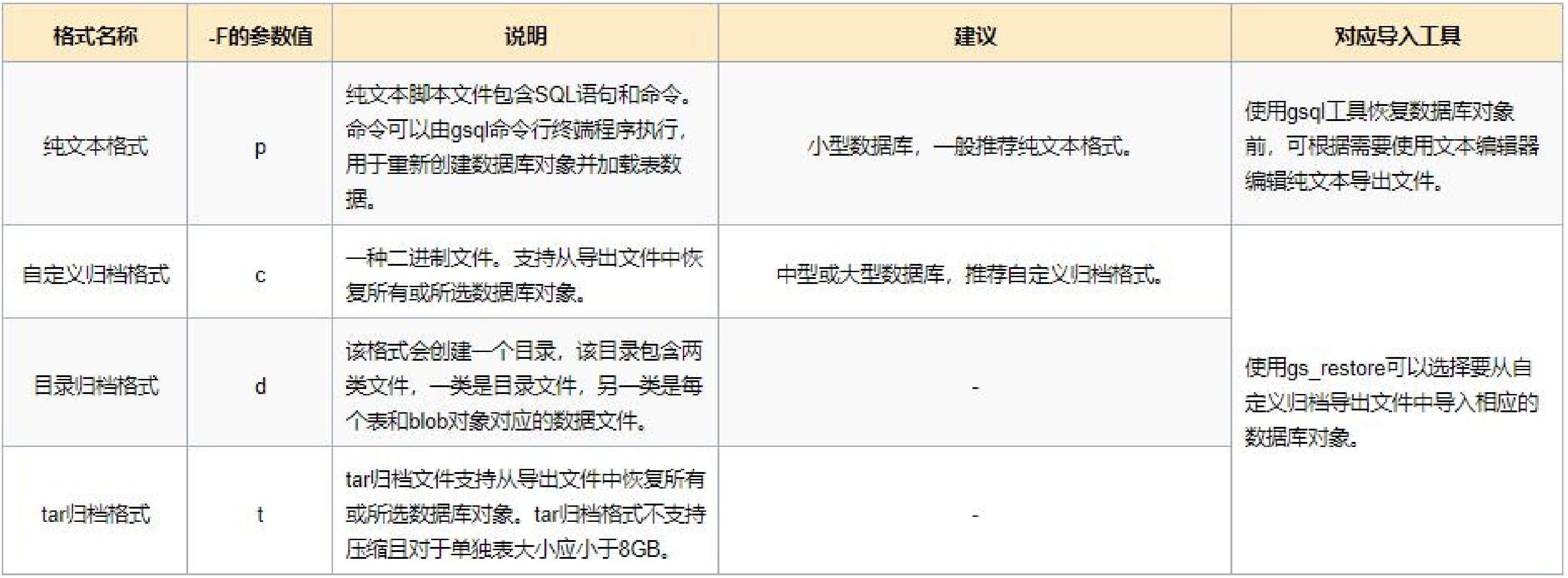
逻辑备份恢复

gs_dump概述

gs_dump工具可以在线导出数据库的数据，这些数据包括整个数据库或数据库中指定的对象(如：模式、表、视图等)。并且支持导出完整一致的数据。

备份格式

-F

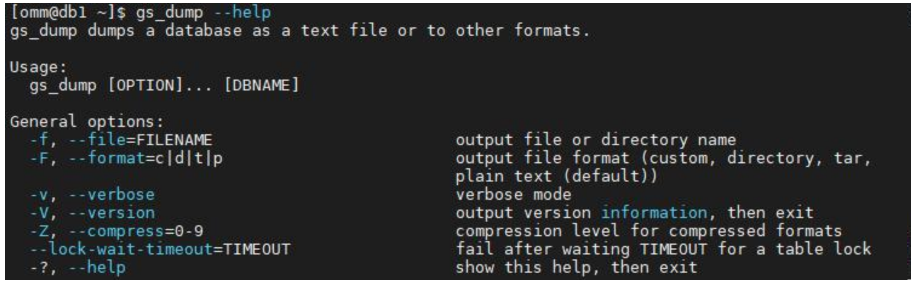


格式名称	-F的参数值	说明	建议	对应导入工具
纯文本格式	p	纯文本脚本文件包含SQL语句和命令。命令可以由gsql命令行终端程序执行，用于重新创建数据库对象并加载表数据。	小型数据库，一般推荐纯文本格式。	使用gsql工具恢复数据库对象前，可根据需要使用文本编辑器编辑纯文本导出文件。
自定义归档格式	c	一种二进制文件。支持从导出文件中恢复所有或所选数据库对象。	中型或大型数据库，推荐自定义归档格式。	使用gs_restore可以选择要从自定义归档导出文件中导入相应的数据库对象。
目录归档格式	d	该格式会创建一个目录，该目录包含两类文件，一类是目录文件，另一类是每个表和blob对象对应的数据文件。	-	
tar归档格式	t	tar归档文件支持从导出文件中恢复所有或所选数据库对象。tar归档格式不支持压缩且对于单独表大小应小于8GB。	-	

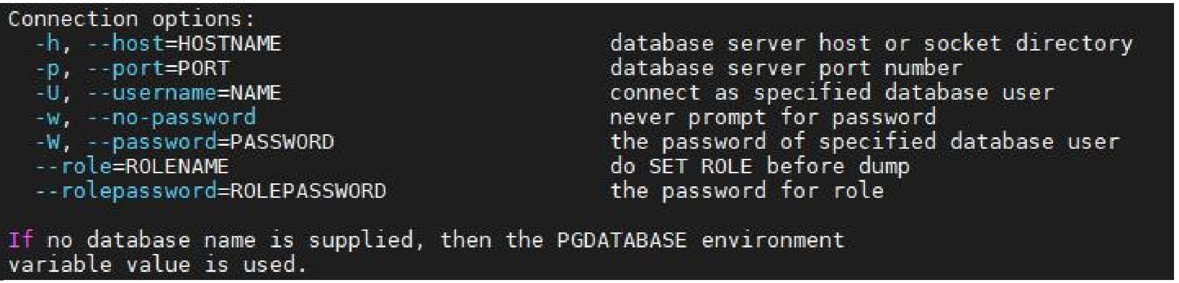
➢**注意事项:**

1. 当数据库的对象数量较多时，可以适当增加参数max_prepared_transactions和max_locks_per_transaction的值，以提升导出效率；
2. gs_dump生成的转储文件不包含统计数据。因此建议从某转储文件恢复之后运行ANALYZE以确保最佳效果；
3. gs_dump导出时会对需要转储的表设置共享锁，以确保数据的一致性和完整性。如果表在别的事务中设置了共享锁，gs_dump会等待锁释放后锁定表。

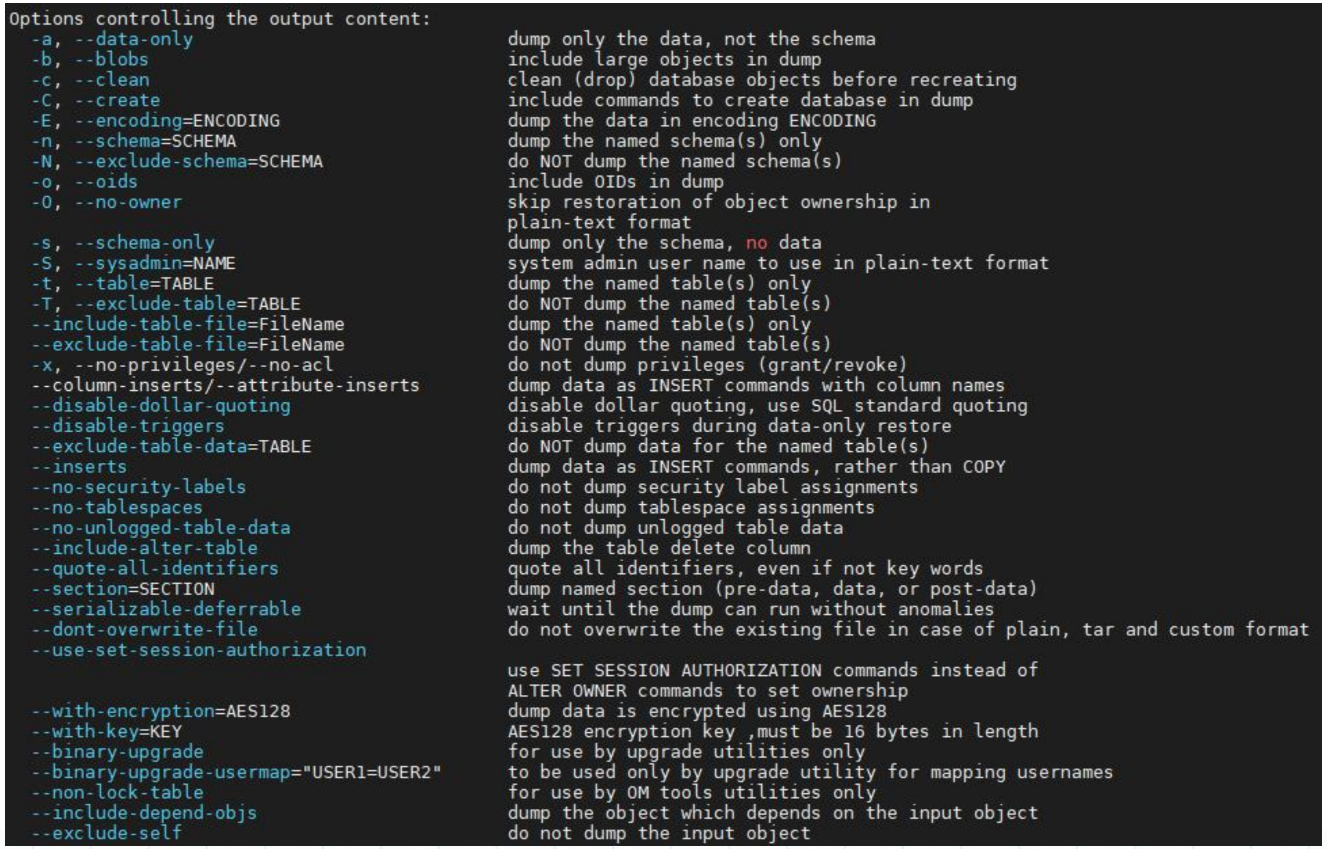
gs_dump参数简介



```
[omm@db1 ~]$ gs_dump --help
gs_dump dumps a database as a text file or to other formats.

Usage:
  gs_dump [OPTION]... [DBNAME]

General options:
  -f, --file=FILENAME                          output file or directory name
  -F, --format=c|d|t|p                         output file format (custom, directory, tar,
                                               plain text (default))
  -v, --verbose                                verbose mode
  -V, --version                                output version information, then exit
  -Z, --compress=0-9                           compression level for compressed formats
  --lock-wait-timeout=TIMEOUT                  fail after waiting TIMEOUT for a table lock
  -?, --help                                   show this help, then exit
```



```
Connection options:
  -h, --host=HOSTNAME                        database server host or socket directory
  -p, --port=PORT                            database server port number
  -U, --username=NAME                        connect as specified database user
  -w, --no-password                          never prompt for password
  -W, --password=PASSWORD                    the password of specified database user
  --role=ROLENAME                            do SET ROLE before dump
  --rolepassword=ROLEPASSWORD                the password for role

If no database name is supplied, then the PGDATABASE environment
variable value is used.
```



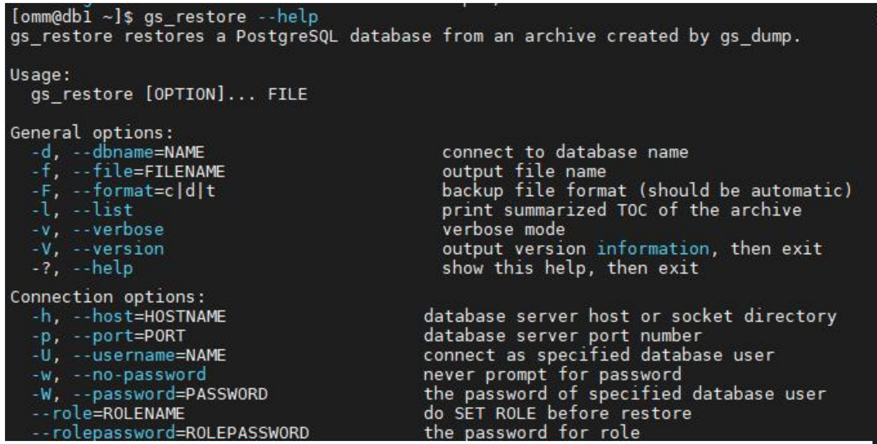
```
Options controlling the output content:
  -a, --data-only                              dump only the data, not the schema
  -b, --blobs                                  include large objects in dump
  -c, --clean                                  clean (drop) database objects before recreating
  -C, --create                                 include commands to create database in dump
  -E, --encoding=ENCODING                      dump the data in encoding ENCODING
  -n, --schema=SCHEMA                          dump the named schema(s) only
  -N, --exclude-schema=SCHEMA                  do NOT dump the named schema(s)
  -o, --oids                                   include OIDs in dump
  -O, --no-owner                               skip restoration of object ownership in
                                               plain-text format
  -s, --schema-only                            dump only the schema, no data
  -S, --sysadmin=NAME                          system admin user name to use in plain-text format
  -t, --table=TABLE                            dump the named table(s) only
  -T, --exclude-table=TABLE                    do NOT dump the named table(s)
  --include-table-file=FileName                dump the named table(s) only
  --exclude-table-file=FileName                do NOT dump the named table(s)
  -x, --no-privileges/--no-acl                 do not dump privileges (grant/revoke)
  --column-inserts/--attribute-inserts         dump data as INSERT commands with column names
  --disable-dollar-quoting                     disable dollar quoting, use SQL standard quoting
  --disable-triggers                           disable triggers during data-only restore
  --exclude-table-data=TABLE                   do NOT dump data for the named table(s)
  --inserts                                    dump data as INSERT commands, rather than COPY
  --no-security-labels                         do not dump security label assignments
  --no-tablespaces                             do not dump tablespace assignments
  --no-unlogged-table-data                     do not dump unlogged table data
  --include-alter-table                        dump the table delete column
  --quote-all-identifiers                      quote all identifiers, even if not key words
  --section=SECTION                            dump named section (pre-data, data, or post-data)
  --serializable-deferrable                    wait until the dump can run without anomalies
  --dont-overwrite-file                        do not overwrite the existing file in case of plain, tar and custom format
  --use-set-session-authorization
                                               use SET SESSION AUTHORIZATION commands instead of
                                               ALTER OWNER commands to set ownership
  --with-encryption=AES128                     dump data is encrypted using AES128
  --with-key=KEY                               AES128 encryption key ,must be 16 bytes in length
  --binary-upgrade                             for use by upgrade utilities only
  --binary-upgrade-usermap="USER1=USER2"       to be used only by upgrade utility for mapping usernames
  --non-lock-table                             for use by OM tools utilities only
  --include-depend-objs                        dump the object which depends on the input object
  --exclude-self                               do not dump the input object
```

gs_restore介绍

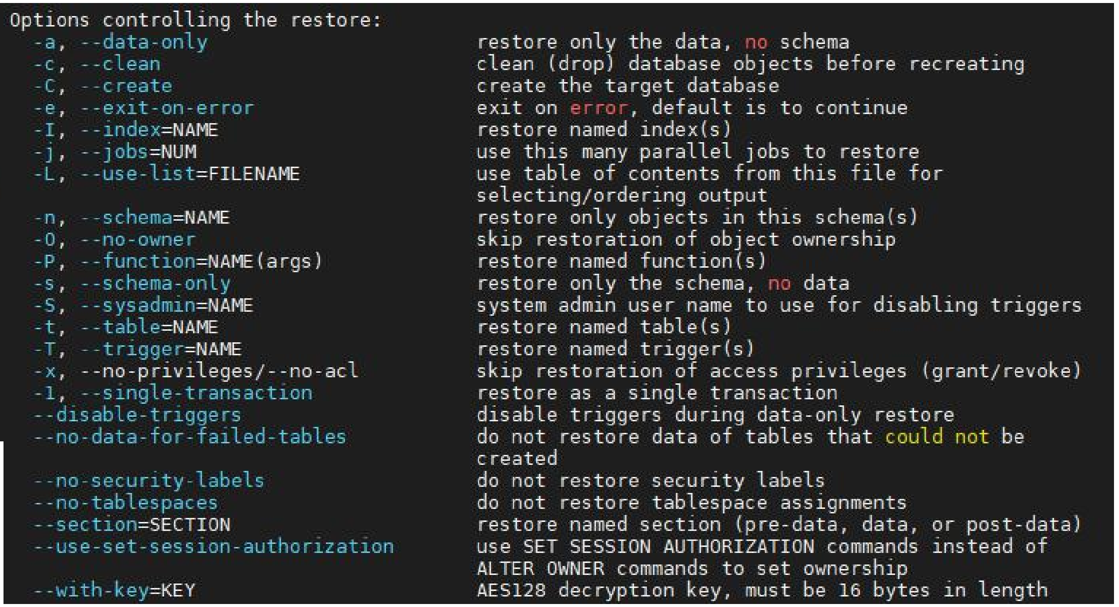
用户可以使用gs_restore工具将gs_dump导出数据备份导入到数据库或指定文件中(等效于直接使用gs_dump导出为纯文本格式)。

➢**注意事项:**

1. gs_restore默认是以追加的方式进行数据导入。为避免多次导入造成数据异常，在进行导入时，建议使用”-c” 参数，在重新创建数据库对象前，清理(删除)已存在的目标数据库对象。
2. 日志打印无开关，若需隐藏日志，请将日志重定向到日志文件。
3. 若恢复表数据时，数据量很大，会分批恢复，因此会多次出现“表数据已完成导入”的日志。



```
[omm@db1 ~]$ gs_restore --help
gs_restore restores a PostgreSQL database from an archive created by gs_dump.

Usage:
  gs_restore [OPTION]... FILE

General options:
  -d, --dbname=NAME                        connect to database name
  -f, --file=FILENAME                      output file name
  -F, --format=c|d|t                       backup file format (should be automatic)
  -l, --list                               print summarized TOC of the archive
  -v, --verbose                            verbose mode
  -V, --version                            output version information, then exit
  -?, --help                               show this help, then exit

Connection options:
  -h, --host=HOSTNAME                    database server host or socket directory
  -p, --port=PORT                        database server port number
  -U, --username=NAME                    connect as specified database user
  -w, --no-password                      never prompt for password
  -W, --password=PASSWORD                the password of specified database user
  --role=ROLENAME                        do SET ROLE before restore
  --rolepassword=ROLEPASSWORD            the password for role
```



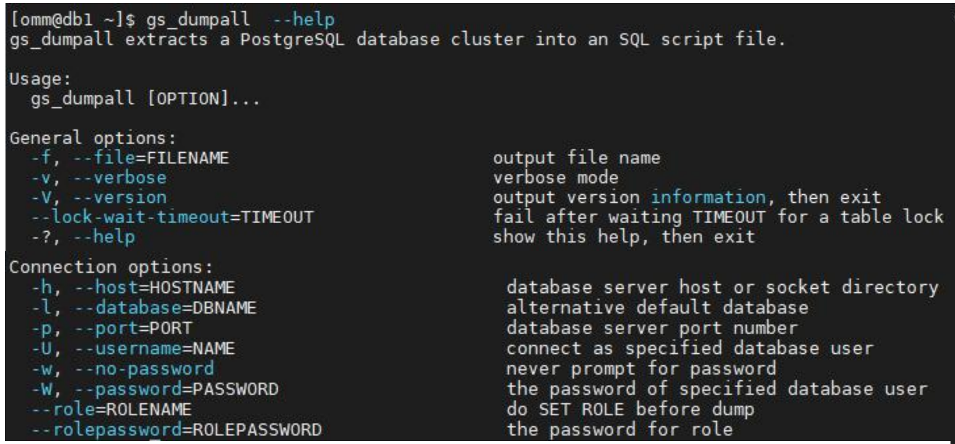
```
Options controlling the restore:
  -a, --data-only                      restore only the data, no schema
  -c, --clean                          clean (drop) database objects before recreating
  -C, --create                         create the target database
  -e, --exit-on-error                  exit on error, default is to continue
  -I, --index=NAME                     restore named index(s)
  -j, --jobs=NUM                       use this many parallel jobs to restore
  -L, --use-list=FILENAME              use table of contents from this file for
                                       selecting/ordering output
  -n, --schema=NAME                    restore only objects in this schema(s)
  -O, --no-owner                       skip restoration of object ownership
  -P, --function=NAME(args)            restore named function(s)
  -s, --schema-only                    restore only the schema, no data
  -S, --sysadmin=NAME                  system admin user name to use for disabling triggers
  -t, --table=NAME                     restore named table(s)
  -T, --trigger=NAME                   restore named trigger(s)
  -x, --no-privileges/--no-acl         skip restoration of access privileges (grant/revoke)
  -1, --single-transaction             restore as a single transaction
  --disable-triggers                   disable triggers during data-only restore
  --no-data-for-failed-tables          do not restore data of tables that could not be
                                       created
  --no-security-labels                 do not restore security labels
  --no-tablespaces                     do not restore tablespace assignments
  --section=SECTION                    restore named section (pre-data, data, or post-data)
  --use-set-session-authorization      use SET SESSION AUTHORIZATION commands instead of
                                       ALTER OWNER commands to set ownership
  --with-key=KEY                       AES128 decryption key, must be 16 bytes in length
```

gs_dumpall介绍

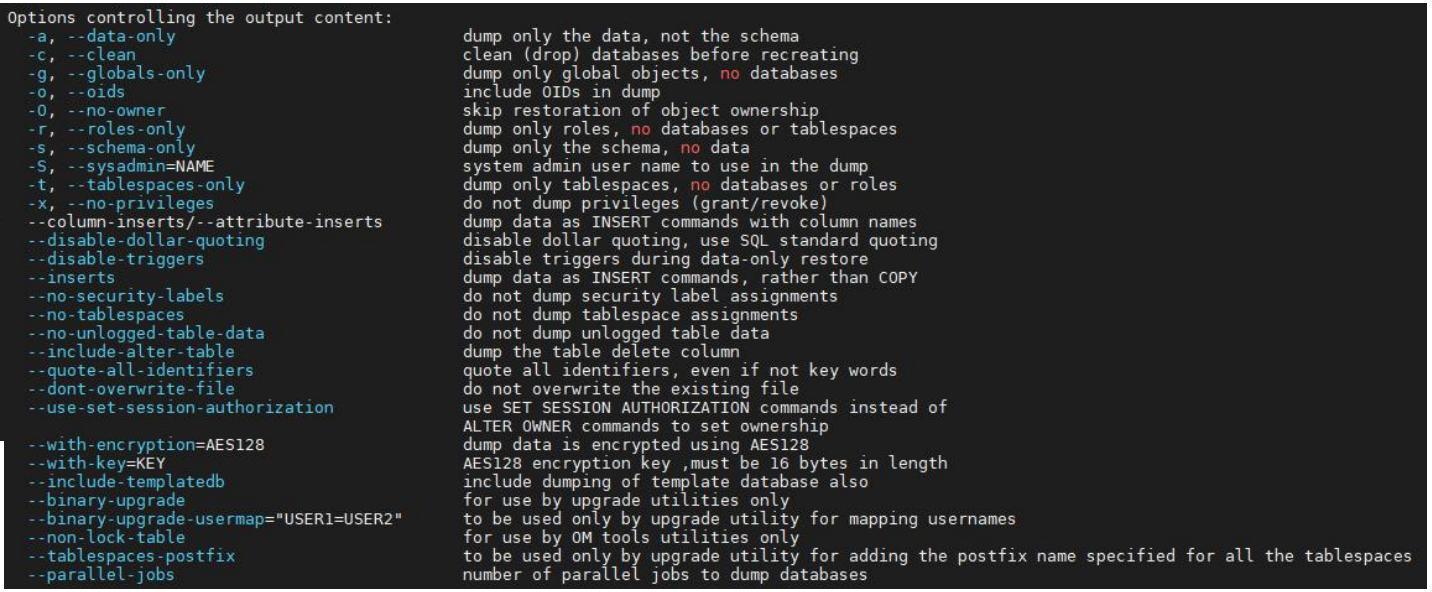
gs_dumpall导出内容分为两部分：

① 公共的全局对象导出，包括有关数据库用户和组，表空间以及属性信息。

② 针对各数据库的SQL文件，该文件包含将数据库恢复为其保存时的状态所需要的全部SQL语句。



```
[omm@db1 ~]$ gs_dumpall  --help
gs_dumpall extracts a PostgreSQL database cluster into an SQL script file.

Usage:
  gs_dumpall [OPTION]...

General options:
  -f, --file=FILENAME                         output file name
  -v, --verbose                               verbose mode
  -V, --version                               output version information, then exit
  --lock-wait-timeout=TIMEOUT                 fail after waiting TIMEOUT for a table lock
  -?, --help                                  show this help, then exit

Connection options:
  -h, --host=HOSTNAME                          database server host or socket directory
  -l, --database=DBNAME                        alternative default database
  -p, --port=PORT                              database server port number
  -U, --username=NAME                          connect as specified database user
  -w, --no-password                            never prompt for password
  -W, --password=PASSWORD                      the password of specified database user
  --role=ROLENAME                              do SET ROLE before dump
  --rolepassword=ROLEPASSWORD                  the password for role
```



```
Options controlling the output content:
  -a, --data-only                              dump only the data, not the schema
  -c, --clean                                  clean (drop) databases before recreating
  -g, --globals-only                           dump only global objects, no databases
  -o, --oids                                   include OIDs in dump
  -O, --no-owner                               skip restoration of object ownership
  -r, --roles-only                             dump only roles, no databases or tablespaces
  -s, --schema-only                            dump only the schema, no data
  -S, --sysadmin=NAME                          system admin user name to use in the dump
  -t, --tablespaces-only                       dump only tablespaces, no databases or roles
  -x, --no-privileges                          do not dump privileges (grant/revoke)
  --column-inserts/--attribute-inserts         dump data as INSERT commands with column names
  --disable-dollar-quoting                     disable dollar quoting, use SQL standard quoting
  --disable-triggers                           disable triggers during data-only restore
  --inserts                                    dump data as INSERT commands, rather than COPY
  --no-security-labels                         do not dump security label assignments
  --no-tablespaces                             do not dump tablespace assignments
  --no-unlogged-table-data                     do not dump unlogged table data
  --include-alter-table                        dump the table delete column
  --quote-all-identifiers                      quote all identifiers, even if not key words
  --dont-overwrite-file                        do not overwrite the existing file
  --use-set-session-authorization              use SET SESSION AUTHORIZATION commands instead of
                                               ALTER OWNER commands to set ownership
  --with-encryption=AES128                     dump data is encrypted using AES128
  --with-key=KEY                               AES128 encryption key ,must be 16 bytes in length
  --include-templatedb                         include dumping of template database also
  --binary-upgrade                             for use by upgrade utilities only
  --binary-upgrade-usermap="USER1=USER2"       to be used only by upgrade utility for mapping usernames
  --non-lock-table                             for use by OM tools utilities only
  --tablespaces-postfix                        to be used only by upgrade utility for adding the postfix name specified for all the tablespaces
  --parallel-jobs                              number of parallel jobs to dump databases
```

gs_dump备份示例

➢ **创建备份用户**

```
gsql -d mydb -p 26000 -c "create user rep1 with sysadmin identified by 'gauss@123';"
gsql -d mydb -p 26000 -c "alter user rep1 with replication;"
```

➢ **修改hba.conf**

```
sed -i '/192.168.0.99/d' /gauss/data/db1/pg_hba.conf
gs_guc reload -N all -I all -h "host replication rep1 192.168.0.99/24 sha256"
gs_guc reload -N all -I all -h "host all rep1 192.168.0.99/24 sha256"
```

➢ **备份数据库**

```
gs_dump -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 mydb -f /home/omm/gs_dump/db_backup.sql -F p     -- 导出纯文档格式
gs_dump -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 mydb -f /home/omm/gs_dump/db_backup.tar -F t     -- 导出tar格式
gs_dump -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 mydb -f /home/omm/gs_dump/db_backup.dmp -F c     -- 导出自定义归档格式
gs_dump -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 mydb -f /home/omm/gs_dump/db_backup     -F d     -- 导出目录格式
gs_dump -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 mydb -f /home/omm/gs_dump/db_define.sql -s -F p  -- 仅备份定义
gs_dump -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 mydb -f /home/omm/gs_dump/data_only.sql -a -F p  -- 仅备份数据
```

➢ **备份schema**

```
gs_dump -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 mydb -n user1 -n user2 -Z 9 -f /home/omm/gs_dump/schema_bak.tar.gz -F p -- 压缩备份schema(user1和user2)
gs_dump -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 mydb -N user2 -f /home/omm/gs_dump/schema_bak2.sql -F p                 -- 备份数据库mydb并排除schema(user2)
```

➢ **备份Table**

```
gs_dump -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 mydb -t public.emp -f /home/omm/gs_dump/emp_bak.sql -F p                -- 备份指定表
```

-- 仅备份user1.*表的依赖对象

```
gs_dump -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 mydb -t user1.* --include-depend-objs --exclude-self -f /home/omm/gs_dump/user1_emp_dep.sql -F p
```

-- 加密备份表user1.*的定义

```
gs_dump -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 mydb -t user1.* -s -F p -f /home/omm/gs_dump/user1_emp_def_encrypt.sql --with-encryption=AES128 --with-
key=1234567812345678
```

-- 备份user1.*和public.*的表(排除public.products表、排除public.newproducts的数据)

```
gs_dump -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 mydb -t public.* -T public.products -t user1.* --exclude-table-data public.newproducts -F p -f
/home/omm/gs_dump/table_bak2.sql
```

gs_restore恢复示例

- **恢复普通格式的备份**

```
gsql -d mydb -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 -f /home/omm/gs_dump/db_define.sql  -- 加密备份需使用-k参数指定秘钥口令
```

- **恢复其他格式的备份**

```
-- 恢复数据库
gs_restore -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 -d postgres -c -C -v -F c /home/omm/gs_dump/db_backup.dmp

-- 恢复schema
gs_restore -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 -d mydb -n public -F c /home/omm/gs_dump/db_backup.dmp

-- 恢复表(-t不支持schema_name.table_name的输入格式)   ## 前提是必须有对应的schema
gs_restore -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 -d mydb -n public -t emp -F c /home/omm/gs_dump/db_backup.dmp

-- 恢复表定义
gs_restore -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 -d mydb  -e -c -s -n public -t emp -F c /home/omm/gs_dump/db_backup.dmp

-- 恢复表数据
gs_restore -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 -d mydb  -e -a -n public -t emp  -F c /home/omm/gs_dump/db_backup.dmp

-- 恢复索引
gs_restore -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 -d mydb -I t1_id_indx -F c /home/omm/gs_dump/db_backup.dmp

-- 恢复函数get_id
gs_restore -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 -d mydb -n public -P 'get_id()' -F t /home/omm/gs_dump/db_backup.tar
```

gs_dumpall示例

- **备份示例**

```
-- 备份所有数据库
gs_dumpall -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 -f /home/omm/gs_dumpall/gs_all.sql

-- 备份全局用户和表空间
gs_dumpall -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 -f /home/omm/gs_dumpall/gs_all1.sql -g

-- 备份全局用户信息
gs_dumpall -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 -f /home/omm/gs_dumpall/gs_all2.sql -r

-- 仅备份数据库定义
gs_dumpall -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 -f /home/omm/gs_dumpall/gs_all3.sql -s

-- 仅加密备份数据
gs_dumpall -U rep1 -W gauss@123 -h 192.168.0.99 -p 26000 -f /home/omm/gs_dumpall/gs_all4.sql   -a --with-encryption=AES128
  --with-key=1234567812345678
```

- **恢复示例**

由于gs_dumpall仅支持纯文本格式导出，所以可以使用gsql客户端读取备份文件，以恢复gs_dumpall导出的数据。
例如：

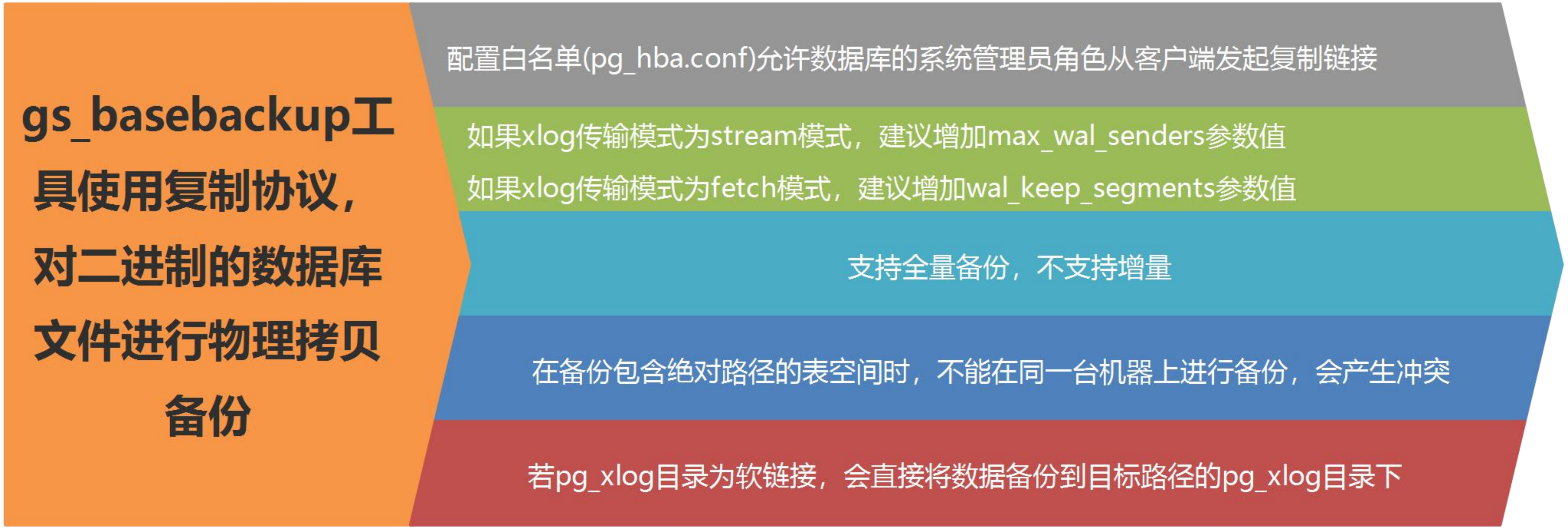
```
gsql -d postgres -p 26000 -f /home/omm/gs_all.bak
```

注意：恢复数据时，由于postgres数据库不进行recreate，所以postgres数据库中已存在的表并没有删除，脚本执行create失败，insert数据时可能造成数据重复的问题。

PART THREE

物理备份恢复

gs_basebackup概述



gs_basebackup工具使用复制协议，对二进制的数据库文件进行物理拷贝备份

配置白名单(pg_hba.conf)允许数据库的系统管理员角色从客户端发起复制链接

如果xlog传输模式为stream模式，建议增加max_wal_senders参数值

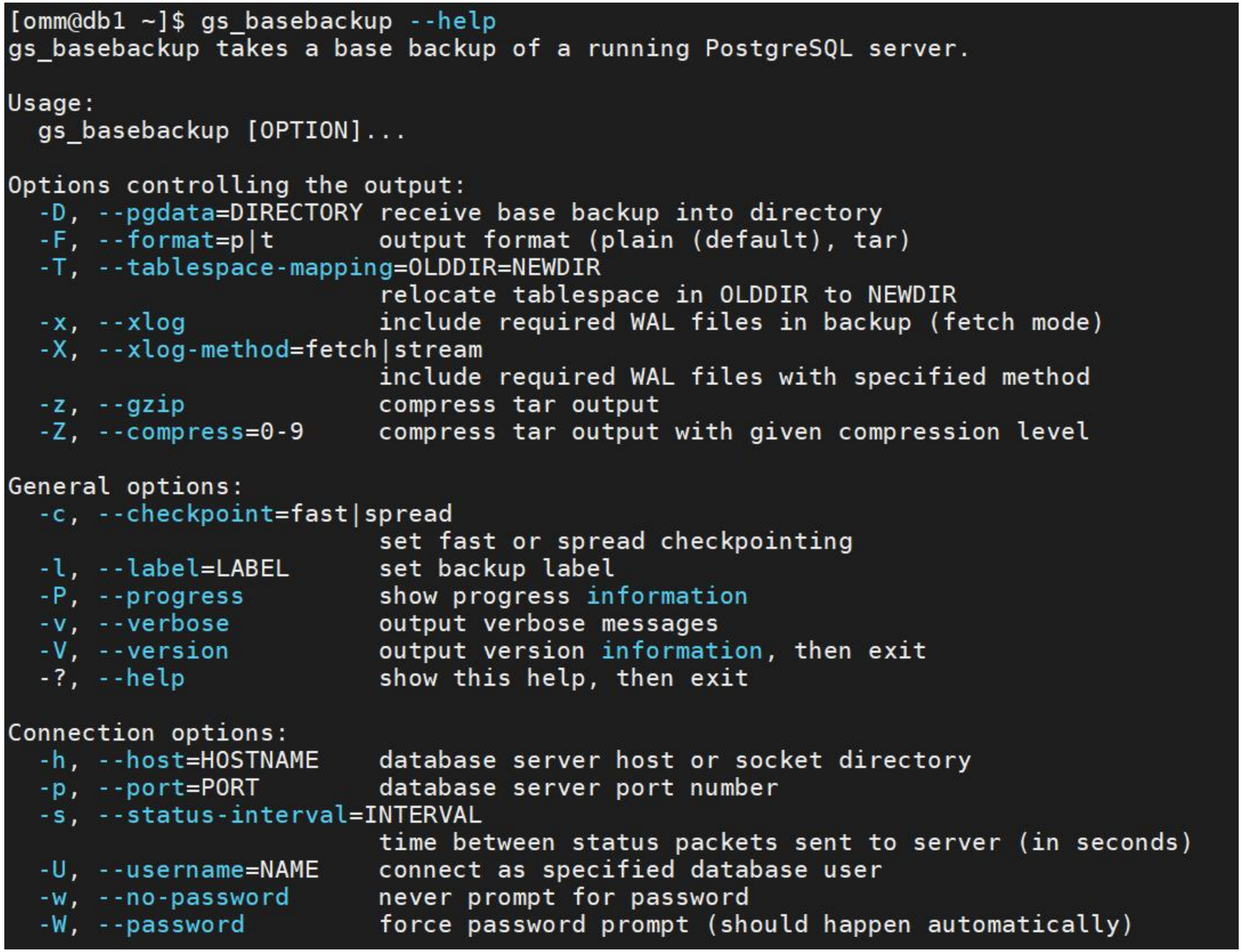
如果xlog传输模式为fetch模式，建议增加wal_keep_segments参数值

支持全量备份，不支持增量

在备份包含绝对路径的表空间时，不能在同一台机器上进行备份，会产生冲突

若pg_xlog目录为软链接，会直接将数据备份到目标路径的pg_xlog目录下

gs_basebackup备份参数介绍



```
[omm@db1 ~]$ gs_basebackup --help
gs_basebackup takes a base backup of a running PostgreSQL server.

Usage:
  gs_basebackup [OPTION]...

Options controlling the output:
  -D, --pgdata=DIRECTORY receive base backup into directory
  -F, --format=p|t       output format (plain (default), tar)
  -T, --tablespace-mapping=OLDDIR=NEWDIR
                         relocate tablespace in OLDDIR to NEWDIR
  -x, --xlog             include required WAL files in backup (fetch mode)
  -X, --xlog-method=fetch|stream
                         include required WAL files with specified method
  -z, --gzip             compress tar output
  -Z, --compress=0-9     compress tar output with given compression level

General options:
  -c, --checkpoint=fast|spread
                         set fast or spread checkpointing
  -l, --label=LABEL      set backup label
  -P, --progress         show progress information
  -v, --verbose          output verbose messages
  -V, --version          output version information, then exit
  -?, --help             show this help, then exit

Connection options:
  -h, --host=HOSTNAME    database server host or socket directory
  -p, --port=PORT        database server port number
  -s, --status-interval=INTERVAL
                         time between status packets sent to server (in seconds)
  -U, --username=NAME    connect as specified database user
  -w, --no-password      never prompt for password
  -W, --password         force password prompt (should happen automatically)
```

gs_basebackup备份示例

➢ **目标机配置操作**

```
-- 查询wal_sender信息
postgres=# select * from pg_stat_get_wal_senders();
postgres=# show max_wal_senders;
 max_wal_senders
-----------------
 8
默认WAL日志使用stream方式复制，该方式最多占用2个walsender线程，需要确保该线程配置足够大。

-- 创建备份用户并放开权限(远程执行gs_basebackup时，需要使用系统管理员账户)
postgres=# create user rep1 with sysadmin identified by 'gauss@123';

$ vi pg_hba.conf
-------------------------------------------------------------------
添加：
host     replication    rep1       192.168.0.0/24          sha256
-------------------------------------------------------------------

-- 创建测试数据
postgres=# create tablespace tbs1 location '/gauss/data/tbs1';   ## 创建绝对路径的表空间
postgres=# create table bak_test(name varchar(20)) tablespace tbs1;
postgres=# insert into bak_test values('This is a test');
postgres=# select * from bak_test;
      name
----------------
 This is a test
```

gs_basebackup备份示例

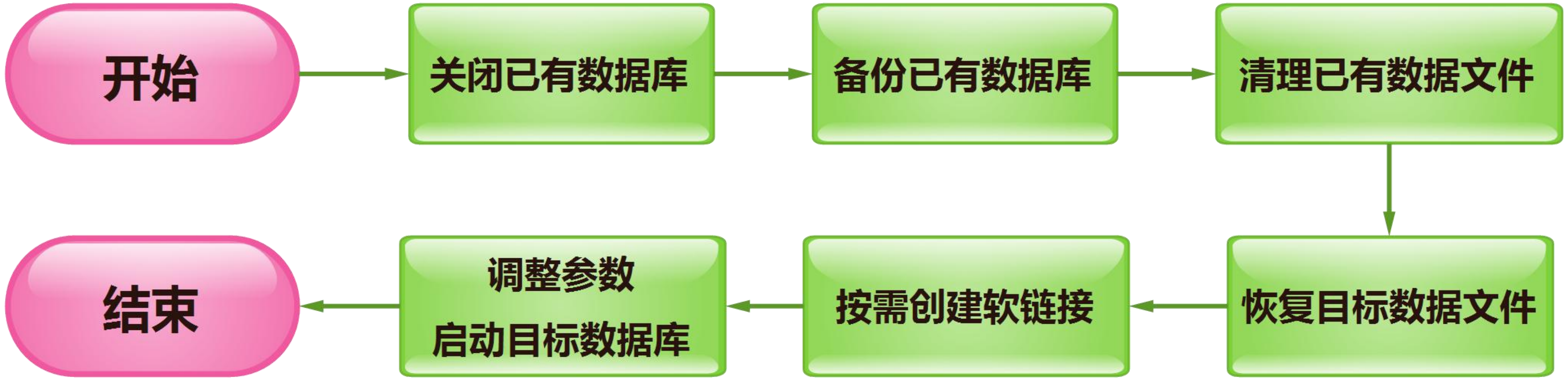
➢ **客户机备份操作**

```
-- 普通备份操作示例:
[omm@client ~]$ gs_basebackup -D /home/omm/gs_bak -h 192.168.0.225 -p 26000 -U rep1 -W
Password:
INFO:  The starting position of the xlog copy of the full build is: 0/37000028. The slot minimum LSN is: 0/0.
begin build tablespace list
finish build tablespace list
begin get xlog by xlogstream
 check identify system success
 send START_REPLICATION 0/37000000 success
 keepalive message is received
 keepalive message is received
 keepalive message is received

-- 使用tar格式压缩备份时，xlog模式不能使用stream，生成的tar包需要用gs_tar命令解压
[omm@client ~]$ gs_basebackup -D /home/omm/gs_bak -X fetch -F t -z -h 192.168.0.225 -p 26000 -U rep1 -W
Password:
INFO:  The starting position of the xlog copy of the full build is: 0/41000028. The slot minimum LSN is: 0/0.
begin build tablespace list
finish build tablespace list
[omm@client gs_bak]$ ls
17161.tar.gz  base.tar.gz

-- 当有绝对路径表空间时，备份操作建议重新定位表空间，或者在远程客户端操作，否则有冲突
[omm@client ~]$ gs_basebackup -D /home/omm/gs_bak -T /gauss/data/tbs1=/home/omm/gs_bak/tbs1 -h 192.168.0.225 -p 26000 -U rep1 -W

-- 检查备份文件
[omm@client ~]$ du -sh gs_bak/
94M     gs_bak/
[omm@client ~]$ ll /gauss/data/tbs1/PG_9.2_201611171_dn_6001/14858/        ## 绝对路径的表空间自动创建成功
total 8
-rw------- 1 omm dbgrp 8192 Nov  6 15:44 17177
```

gs_basebackup恢复概述

gs_basebackup备份的是数据库的二进制文件，因此在恢复时可以直接拷贝替换原有的文件，或者直接在备份目录启动数据库。但需要注意的是，必要时需要在实例启动前先修改配置参数(如：服务端口，主备复制配置等信息)

若要在原库的地方恢复数据库，建议操作如下：



gs_basebackup恢复示例

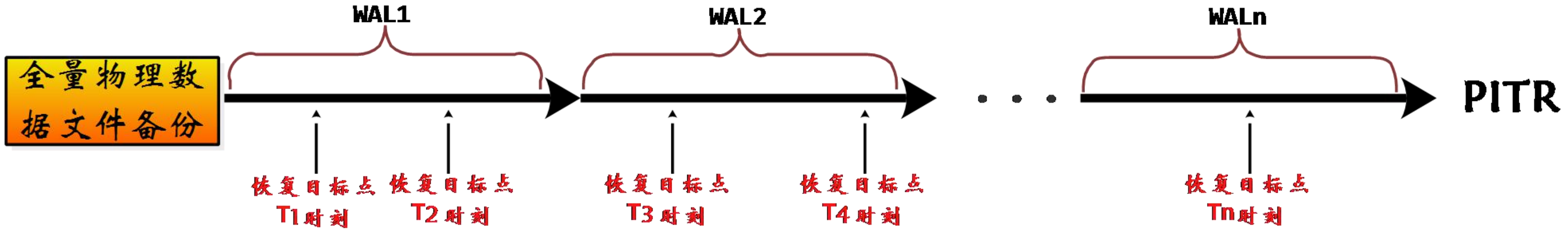
➢ **客户机恢复操作**

```
-- 备份原数据库目录
cd /gauss
mv data data_bak
mkdir -p data/db1


-- 恢复base.tar至/gauss/data/db1
cd /home/omm/bak/
gunzip *.gz
gs_tar -D /gauss/data/db1 -F base.tar  ## tar包需要用gs_tar
  命令解压备份至指定目录


-- 检查表空间映射信息
[omm@db2 db1]$ cd /gauss/data/db1
[omm@db2 db1]$ cat tablespace_map
16434 /gauss/data/tbs2
16386 /gauss/data/db1/pg_location/tablespace/tbs1

-- 解压表空间备份至指定目录
mkdir -p /gauss/data/tbs2
mkdir -p /gauss/data/db1/pg_location/tablespace/tbs1
cd /home/omm/bak/
gs_tar -D /gauss/data/tbs2 -F 16434.tar
gs_tar -D /gauss/data/db1/pg_location/tablespace/tbs1 -F
  16386.tar
```

```
-- 修改postgres.conf文件
[omm@client ~]$ cd /gauss/data/db1/
[omm@client db1]$ vi postgresql.conf
-------------------------------------------
## 修改:
listen_addresses = '192.168.0.226'
local_bind_address = '192.168.0.226'
port = 27000
## 修改或删除复制链接
## replconninfo1 = 'localhost=192.168.100.11
  localport=26001 localheartbeatport=26005
  localservice=26004 remotehost=192.168.100.12
  remoteport=26001 remoteheartbeatport=26005
  remoteservice=26004'
--------------------------------------------


-- 启动备份数据库
[omm@client db1]$ gs_ctl  start -D /gauss/data/db1/


-- 检查恢复后的数据库状态
[omm@db2 db1]$ gsql -d mydb -p 27000 -r
mydb=# select * from bak_test;
      name
-----------------
 This is a test.
```

PITR恢复概述

当数据库崩溃或希望回退到数据库之前的某一状态时，opengauss的即时恢复功能(Point-In-Time Recovery，简称PITR)可以支持恢复到备份归档数据之后的任意时间点。



全量物理数据文件备份

WAL1

WAL2

WALn

・・・

PITR

恢复目标点 T1时刻

恢复目标点 T2时刻

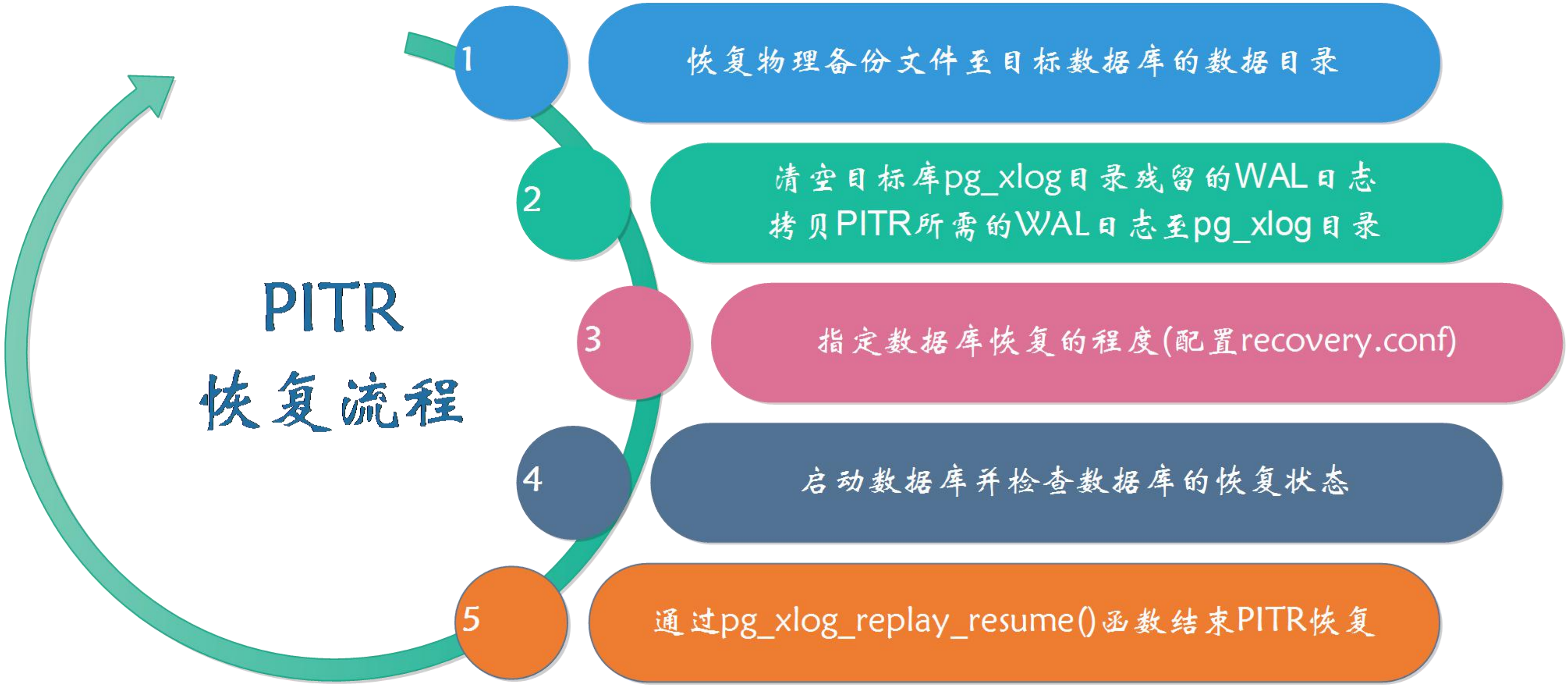
恢复目标点 T3时刻

恢复目标点 T4时刻

恢复目标点 Tn时刻

```
➢ recovery.conf文件配置
-----------------------------------------
## 归档恢复配置
restore_command = 'cp /mnt/server/archivedir/%f %p' ## 获取所需的WAL文件。%f即归档检索中的文件名，%p即复制目的路径

## 恢复目标设置(四选一)
recovery_target_name = 'restore_point_1'       ## 还原到一个使用pg_create_restore_point()创建的还原点
recovery_target_time = '2020-01-01 12:00:00'   ## 还原到一个指定时间戳
recovery_target_xid = '3000'                   ## 还原到一个事务ID。
recovery_target_lsn = '0/0FFFFFF'              ## 还原到日志的指定LSN点。
recovery_target_inclusive = true               ## 指定恢复目标之后停止(true) 或 之前停止(false)
## Tips：如果不配置任何恢复目标 或 配置目标不存在，则默认恢复到最新的WAL日志点。
-------------------------------------------------
```

PITR恢复流程



1 恢复物理备份文件至目标数据库的数据目录

2 清空目标库pg_xlog目录残留的WAL日志
拷贝PITR所需的WAL日志至pg_xlog目录

3 指定数据库恢复的程度(配置recovery.conf)

4 启动数据库并检查数据库的恢复状态

5 通过pg_xlog_replay_resume()函数结束PITR恢复

openGauss

Gauss松鼠会

PITR恢复示例(准备)

➢ 创建测试数据并全备数据库

```
postgres=# create table t1 (id int,tm timestamp,LSN varchar(20));
postgres=# insert into t1 values(1,now(),'Started');
postgres=# select * from t1;
postgres=# select * from pg_current_xlog_location();
 pg_current_xlog_location
--------------------------
 9/A002860

-- 全库备份
$ mkdir /home/omm/gs_bak
$ gs_basebackup -D /home/omm/gs_bak -p 26000
```

➢ 第1次插入数据

```
postgres=# insert into t1 values(2,now(),'First Insert');
postgres=# select * from t1;
 id |             tm             |      lsn
----+----------------------------+--------------
  1 | 2021-07-28 11:11:19.538926 | Started
  2 | 2021-07-28 11:13:12.063549 | First Insert

postgres=# select * from pg_current_xlog_location();
 pg_current_xlog_location
--------------------------
 9/C000348
```

➢ 第2次插入数据

```
postgres=# insert into t1 values(3,now(),'Second Insert');
postgres=# select * from t1;
 id |             tm             |      lsn
----+----------------------------+--------------
  1 | 2021-07-28 11:11:19.538926 | Started
  2 | 2021-07-28 11:13:12.063549 | First Insert
  3 | 2021-07-28 11:14:24.783596 | Second Insert

postgres=# select * from pg_current_xlog_location();
 pg_current_xlog_location
--------------------------
 9/C000550
```

➢ 增量WAL日志拷贝

```
postgres=# select pg_switch_xlog();
postgres=# select pg_switch_xlog();
 pg_switch_xlog
----------------
 9/D000160

cp /gauss/data/db1/pg_xlog/*{A,B,C}  /home/omm/gs_bak/pg_xlog/
```

PITR恢复示例(恢复)

➢ 全量恢复数据库

```
$ gs_om -t stop
$ rm -fr /gauss/data/db1
$ mkdir /gauss/data/db1
$ cp -fr /home/omm/gs_bak/*  /gauss/data/db1/
```

➢ 停止PITR恢复

```
postgres=# select pg_is_in_recovery();
 pg_is_in_recovery
-------------------
 t

postgres=# select pg_xlog_replay_resume();
 pg_xlog_replay_resume
-----------------------
```

```
postgres=# select pg_is_in_recovery();
 pg_is_in_recovery
-------------------
 f

postgres=# select pg_last_xact_replay_timestamp();
 pg_last_xact_replay_timestamp
-------------------------------
 2021-07-28 11:14:24.783853+08
```

➢ 第1次PITR恢复

-- 配置recovery.conf文件

```
$ vi /gauss/data/db1/recovery.conf
---------------------------------------
restore_command = 'cp /home/omm/gs_bak/pg_xlog/%f %p'
recovery_target_lsn = '9/A002860'
recovery_target_inclusive = false
---------------------------------------
```

-- 启动数据库，检查恢复情况

```
$ gs_om -t start
$ gsql -d postgres -p 26000 -r
postgres=# select * from t1;
 id |             tm             |   lsn
----+----------------------------+---------
  1 | 2021-07-28 11:11:19.538926 | Started
```

➢ 第2次PITR恢复

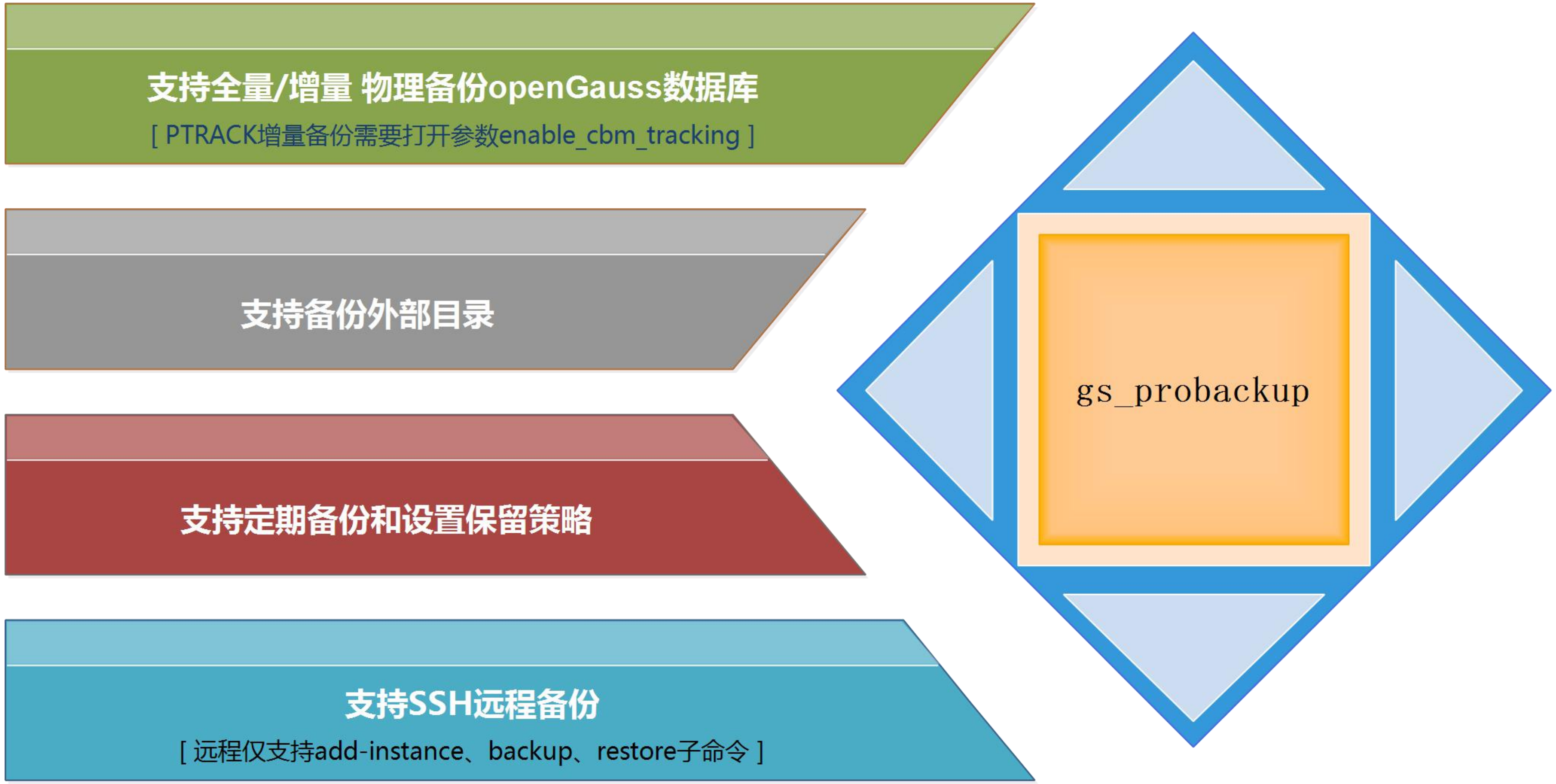
```
$ gs_om -t stop
$ vi /gauss/data/db1/recovery.conf
---------------------------------------
restore_command = 'cp /home/omm/gs_bak/pg_xlog/%f %p'
recovery_target_lsn = '9/C000348'
recovery_target_inclusive = false
---------------------------------------
$ gs_om -t start

-- 检查第一次PITR恢复情况
$ gsql -d postgres -p 26000 -r
postgres=# select * from t1;
 id |             tm             |     lsn
----+----------------------------+--------------
  1 | 2021-07-28 11:11:19.538926 | Started
  2 | 2021-07-28 11:13:12.063549 | First Insert
```

➢ 第3次PITR恢复

```
$ gs_om -t stop
$ vi /gauss/data/db1/recovery.conf
---------------------------------------
restore_command = 'cp /home/omm/gs_bak/pg_xlog/%f %p'
recovery_target_lsn = '9/C000550'
recovery_target_inclusive = false
---------------------------------------
$ gs_om -t start

-- 检查第二次PITR恢复情况
$ gsql -d postgres -p 26000 -r
postgres=# select * from t1;
 id |             tm             |      lsn
----+----------------------------+---------------
  1 | 2021-07-28 11:11:19.538926 | Started
  2 | 2021-07-28 11:13:12.063549 | First Insert
  3 | 2021-07-28 11:14:24.783596 | Second Insert
```

注意：此时openGauss处于recover状态，只读。

gs_probackup简介



支持全量/增量 物理备份openGauss数据库

[PTRACK增量备份需要打开参数enable_cbm_tracking]

支持备份外部目录

支持定期备份和设置保留策略

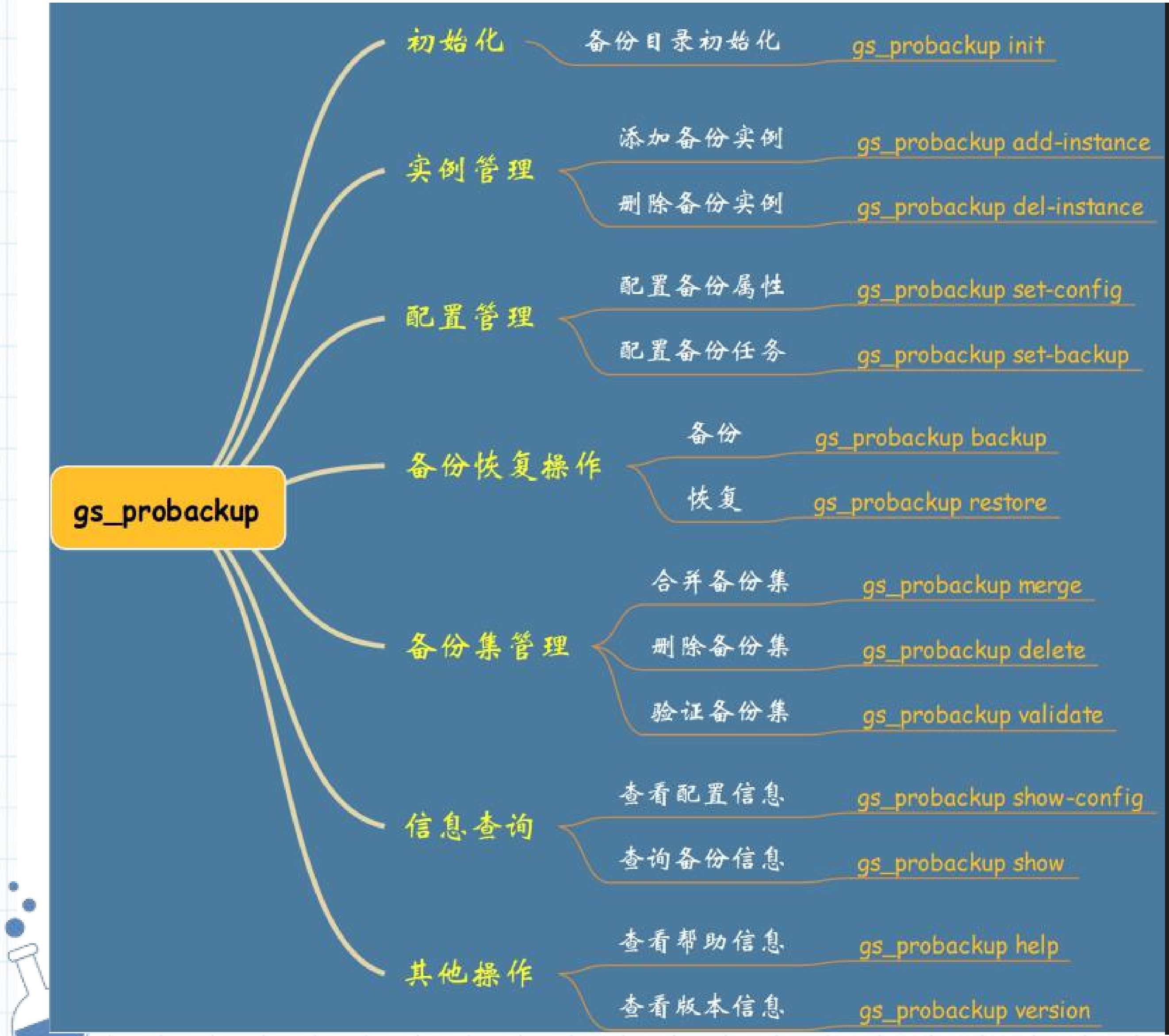
支持SSH远程备份

[远程仅支持add-instance、backup、restore子命令]

注意事项

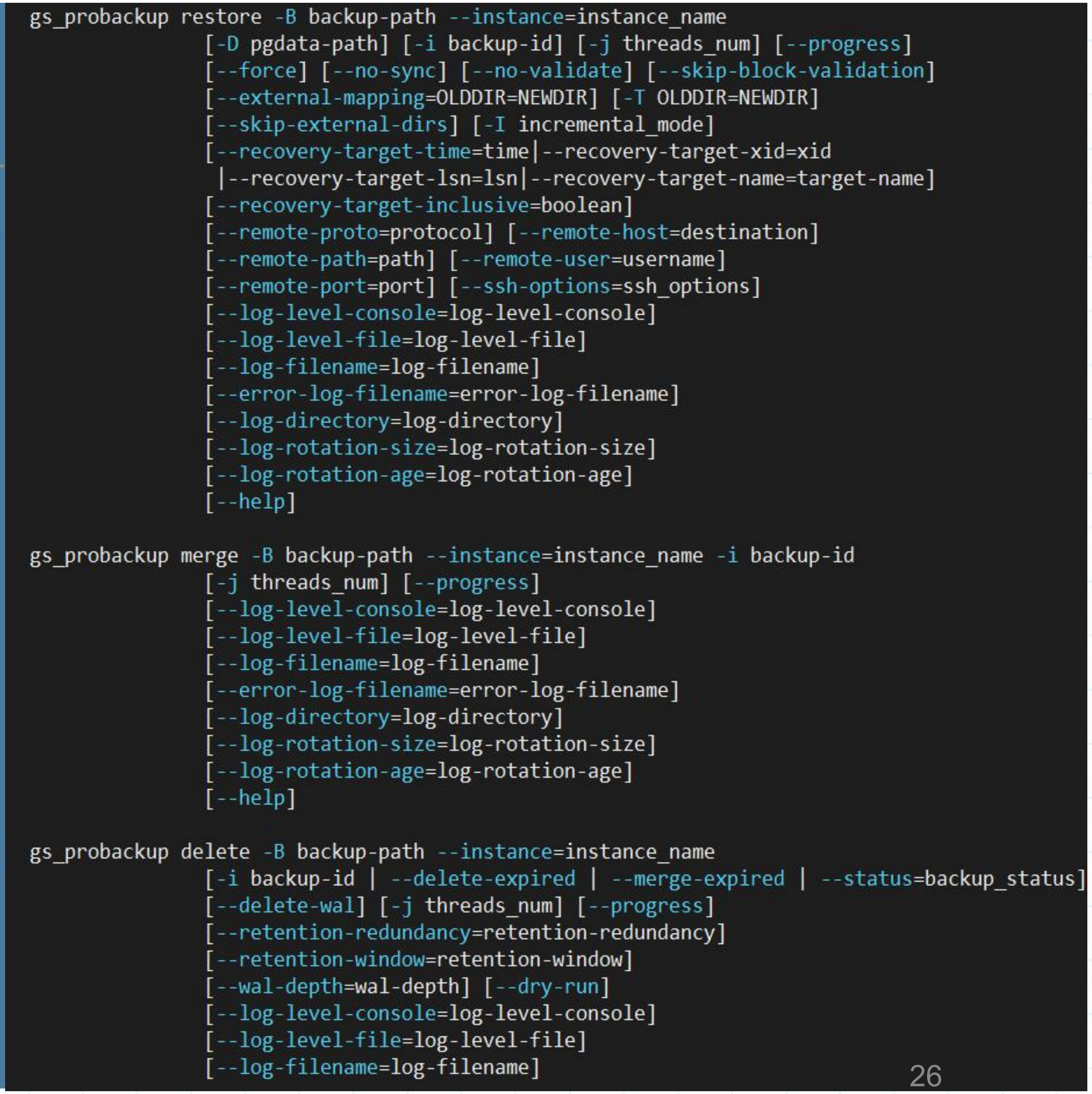
- 备份操作必须由运行数据库的用户执行。
- 服务端、备份端 和 恢复端的数据库主版本号必须一致。

gs_probackup功能概览



gs_probackup

- 初始化
  - 备份目录初始化　gs_probackup init
- 实例管理
  - 添加备份实例　gs_probackup add-instance
  - 删除备份实例　gs_probackup del-instance
- 配置管理
  - 配置备份属性　gs_probackup set-config
  - 配置备份任务　gs_probackup set-backup
- 备份恢复操作
  - 备份　gs_probackup backup
  - 恢复　gs_probackup restore
- 备份集管理
  - 合并备份集　gs_probackup merge
  - 删除备份集　gs_probackup delete
  - 验证备份集　gs_probackup validate
- 信息查询
  - 查看配置信息　gs_probackup show-config
  - 查询备份信息　gs_probackup show
- 其他操作
  - 查看帮助信息　gs_probackup help
  - 查看版本信息　gs_probackup version



```
gs_probackup restore -B backup-path --instance=instance_name
               [-D pgdata-path] [-i backup-id] [-j threads_num] [--progress]
               [--force] [--no-sync] [--no-validate] [--skip-block-validation]
               [--external-mapping=OLDDIR=NEWDIR] [-T OLDDIR=NEWDIR]
               [--skip-external-dirs] [-I incremental_mode]
               [--recovery-target-time=time|--recovery-target-xid=xid
                |--recovery-target-lsn=lsn|--recovery-target-name=target-name]
               [--recovery-target-inclusive=boolean]
               [--remote-proto=protocol] [--remote-host=destination]
               [--remote-path=path] [--remote-user=username]
               [--remote-port=port] [--ssh-options=ssh_options]
               [--log-level-console=log-level-console]
               [--log-level-file=log-level-file]
               [--log-filename=log-filename]
               [--error-log-filename=error-log-filename]
               [--log-directory=log-directory]
               [--log-rotation-size=log-rotation-size]
               [--log-rotation-age=log-rotation-age]
               [--help]

gs_probackup merge -B backup-path --instance=instance_name -i backup-id
               [-j threads_num] [--progress]
               [--log-level-console=log-level-console]
               [--log-level-file=log-level-file]
               [--log-filename=log-filename]
               [--error-log-filename=error-log-filename]
               [--log-directory=log-directory]
               [--log-rotation-size=log-rotation-size]
               [--log-rotation-age=log-rotation-age]
               [--help]

gs_probackup delete -B backup-path --instance=instance_name
               [-i backup-id | --delete-expired | --merge-expired | --status=backup_status]
               [--delete-wal] [-j threads_num] [--progress]
               [--retention-redundancy=retention-redundancy]
               [--retention-window=retention-window]
               [--wal-depth=wal-depth] [--dry-run]
               [--log-level-console=log-level-console]
               [--log-level-file=log-level-file]
               [--log-filename=log-filename]
```

openGauss

Gauss松鼠会

gs_probackup备份示例1(初始化)

1. 打开参数enable_cbm_tracking,跟踪数据页的变化

```
mydb=# alter system set enable_cbm_tracking=on;
mydb=# show enable_cbm_tracking;
 enable_cbm_tracking
---------------------
 on
```

2. 初始化备份目录(/home/omm/gs_bak2021)

```
[omm@prod ~]$ gs_probackup init -B /home/omm/gs_bak2021/
INFO: Backup catalog '/home/omm/gs_bak2021' successfully inited
```

3. 添加备份实例

```
[omm@prod ~]$ gs_probackup add-instance -B /home/omm/gs_bak2021 -D /gauss/data/db1 --instance gs_bak2021_inst
INFO: Instance 'gs_bak2021_inst' successfully inited

[omm@prod ~]# tree -L 3 /home/omm/gs_bak2021/
/home/omm/gs_bak2021/
|-- backups
|   `-- gs_2021_inst
|       `-- pg_probackup.conf
`-- wal
    `-- gs_2021_inst
[omm@prod ~]$ gs_probackup show -B /home/omm/gs_bak2021/
BACKUP INSTANCE 'gs_bak2021_inst'
=================================================================================================================
 Instance  Version  ID  Recovery Time  Mode  WAL Mode  TLI  Time  Data  WAL  Zratio  Start LSN  Stop LSN  Status
=================================================================================================================
```

gs_probackup备份示例2(全量备份)

4. 执行一次全量备份

```
[omm@prod ~]$ gs_probackup backup -B /home/omm/gs_bak2021 --instance gs_bak2021_inst -b full -D /gauss/data/db1 -d
mydb -p 26000 \
> --log-directory=/home/omm/gs_bak2021/log --log-filename=full_20210111.log --log-rotation-size=10GB --log-rotation-
age=30d  --log-level-file=info \
> --retention-redundancy=2 \
> --compress \
> --progress --note='This is full backup set.'


[omm@prod ~]$ gs_probackup show -B /home/omm/gs_bak2021/
BACKUP INSTANCE 'gs_bak2021_inst'
=====================================================================================================================
===================
 Instance         Version  ID      Recovery Time           Mode  WAL Mode  TLI  Time   Data   WAL  Zratio  Start LSN
Stop LSN   Status
=====================================================================================================================
===================
 gs_bak2021_inst  9.2      QMR0ZE  2021-01-11 10:45:17+08  FULL  STREAM    1/0    9s  551MB  16MB    1.05  0/F000028
0/F0001E0  OK
```

gs_probackup备份示例3(增量备份)

5. 执行第一次增量备份

```
[omm@prod ~]$ gs_probackup backup -B /home/omm/gs_bak2021 --instance gs_bak2021_inst -b PTRACK -D /gauss/data/db1 -d
mydb -p 26000 --progress \
> --log-directory=/home/omm/gs_bak2021/log  --log-rotation-size=10GB --log-rotation-age=30d  --log-level-file=info
--log-filename=incr1_20210111.log \
> --delete-expired --delete-wal \
> --retention-redundancy=2 \
> --compress  \
> --note='This is the first incremental backup set.'
```

6. 执行第二次增量备份

```
[omm@prod ~]$ gs_probackup backup -B /home/omm/gs_bak2021 --instance gs_bak2021_inst -b PTRACK -D /gauss/data/db1 -d
mydb -p 26000 --progress \
> --log-directory=/home/omm/gs_bak2021/log  --log-rotation-size=10GB --log-rotation-age=30d  --log-level-file=info
--log-filename=incr2_20210111.log \
> --delete-expired --delete-wal \
> --retention-redundancy=2 \
> --compress  \
> --note='This is the second incremental backup set.'
```

gs_probackup备份示例4(备份信息查询)

7. 查看备份清单

```
[omm@prod ~]$ gs_probackup show -B /home/omm/gs_bak2021/

BACKUP INSTANCE 'gs_bak2021_inst'
======================================================================================================================
======================
 Instance         Version  ID      Recovery Time           Mode    WAL Mode  TLI  Time   Data   WAL  Zratio  Start LSN
Stop LSN    Status
======================================================================================================================
======================
 gs_bak2021_inst  9.2      QMR1J4  2021-01-11 10:57:05+08  PTRACK  STREAM    1/1    5s  273MB  16MB    0.94
0/15000028  0/150002F8  OK
 gs_bak2021_inst  9.2      QMR19V  2021-01-11 10:51:31+08  PTRACK  STREAM    1/1    5s  273MB  16MB    0.94
0/11000028  0/110002F8  OK
 gs_bak2021_inst  9.2      QMR0ZE  2021-01-11 10:45:17+08  FULL    STREAM    1/0    9s  551MB  16MB    1.05  0/F000028
0/F0001E0   OK
```

gs_probackup恢复示例1(全量恢复)

```
1. 全量恢复
[omm@prod ~]$ gs_probackup restore -B /home/omm/gs_bak2021/ -D /gauss/data/db1 --instance=gs_bak2021_inst -i QMR0ZE --progress -j 4    ##  -i指定备份文件ID，QMR0ZE即为全备的ID

-- 验证全备数据(全备时刻)
[omm@prod ~]$ gs_ctl start -D /gauss/data/db1
[omm@prod ~]$ gsql -d mydb -p 26000 -r
gsql ((openGauss 1.1.0 build 392c0438) compiled at 2020-12-31 20:07:42 commit 0 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

mydb=# \d
                                List of relations
 Schema |   Name   |        Type        | Owner |              Storage
--------+----------+-------------------+-------+----------------------------------
 public | dept     | table             | omm   | {orientation=row,compression=no}
 public | emp      | table             | omm   | {orientation=row,compression=no}
 public | mv_emp   | materialized view | omm   |
 public | salgrade | table             | omm   | {orientation=row,compression=no}
 public | seq1     | sequence          | omm   |
 public | v_emp    | view              | omm   |
```

gs_probackup恢复示例2(第一次增备恢复)

```
2. 第一次增量恢复
[omm@prod ~]$ gs_probackup restore -B /home/omm/gs_bak2021/ --instance=gs_bak2021_inst -D /gauss/data/db1 -i QMR19V --
progress  ##  -i指定备份文件ID, QMR19V即第一次增备的ID

-- 验证数据
[omm@prod ~]$ gs_ctl start -D /gauss/data/db1/
[omm@prod ~]$ gsql -d mydb -p 26000 -r

mydb=# \d
                                 List of relations
 Schema |   Name     |        Type        | Owner |             Storage
--------+------------+--------------------+-------+----------------------------------
 public | dept       | table              | omm   | {orientation=row,compression=no}
 public | emp        | table              | omm   | {orientation=row,compression=no}
 public | incr_bak1  | table              | omm   | {orientation=row,compression=no}
 public | mv_emp     | materialized view  | omm   |
 public | salgrade   | table              | omm   | {orientation=row,compression=no}
 public | seq1       | sequence           | omm   |
 public | v_emp      | view               | omm   |

mydb=# select * from incr_bak1;
           name
---------------------------
 This is the first change.
```

gs_probackup恢复示例3(第二次增备恢复)

```
3. 第二次增量恢复
[omm@prod ~]$ gs_probackup restore -B /home/omm/gs_bak2021/ --instance=gs_bak2021_inst -D /gauss/data/db1 -i QMR1J4
--progress   ##  -i指定备份文件ID，QMR1J4即第二次增备的ID

-- 验证数据
[omm@prod ~]$ gs_ctl start -D /gauss/data/db1/
[omm@prod ~]$ gsql -d mydb -p 26000 -r

mydb=# \d
                                 List of relations
 Schema |    Name    |        Type        | Owner |               Storage
--------+------------+--------------------+-------+----------------------------------
 public | dept       | table              | omm   | {orientation=row,compression=no}
 public | emp        | table              | omm   | {orientation=row,compression=no}
 public | incr_bak1  | table              | omm   | {orientation=row,compression=no}
 public | incr_bak2  | table              | omm   | {orientation=row,compression=no}
 public | mv_emp     | materialized view  | omm   |
 public | salgrade   | table              | omm   | {orientation=row,compression=no}
 public | seq1       | sequence           | omm   |
 public | v_emp      | view               | omm   |

mydb=# select * from incr_bak1,incr_bak2;
          name             |             name
---------------------------+-----------------------------
 This is the first change. | This is the second change.
```

gs_probackup小结

注意事项

1 默认连接和当前用户同名的数据库

使用gs_probackup时，需要通过-d指定备份时连接的数据库，否则默认会使用和当前用户同名的数据库，这样会因为这个数据库不存在而导致报错

2 实际上备份的是整个数据库集簇

从备份文件看，gs_probackup工具属于物理备份，虽然备份时需要连接特定的数据库，但是实际上备份的是整个database cluster

3 增备恢复时仅需指定恢复目标

增量恢复时，gs_probackup会验证所有所需的全备和增备文件，然后根据全备+增备文件顺序恢复，用户仅需要指定恢复目标(如：整个备份集、指定的[lsn/xid/time/save point]等recovery_target)

4 备份错误需删除错误备份集

gs_probackup备份任务发生错误时，并不会完全干净地退出，导致无法继续执行备份操作，此时需要手动删除ERROR的备份集

PART FOUR

二进制程序备份恢复

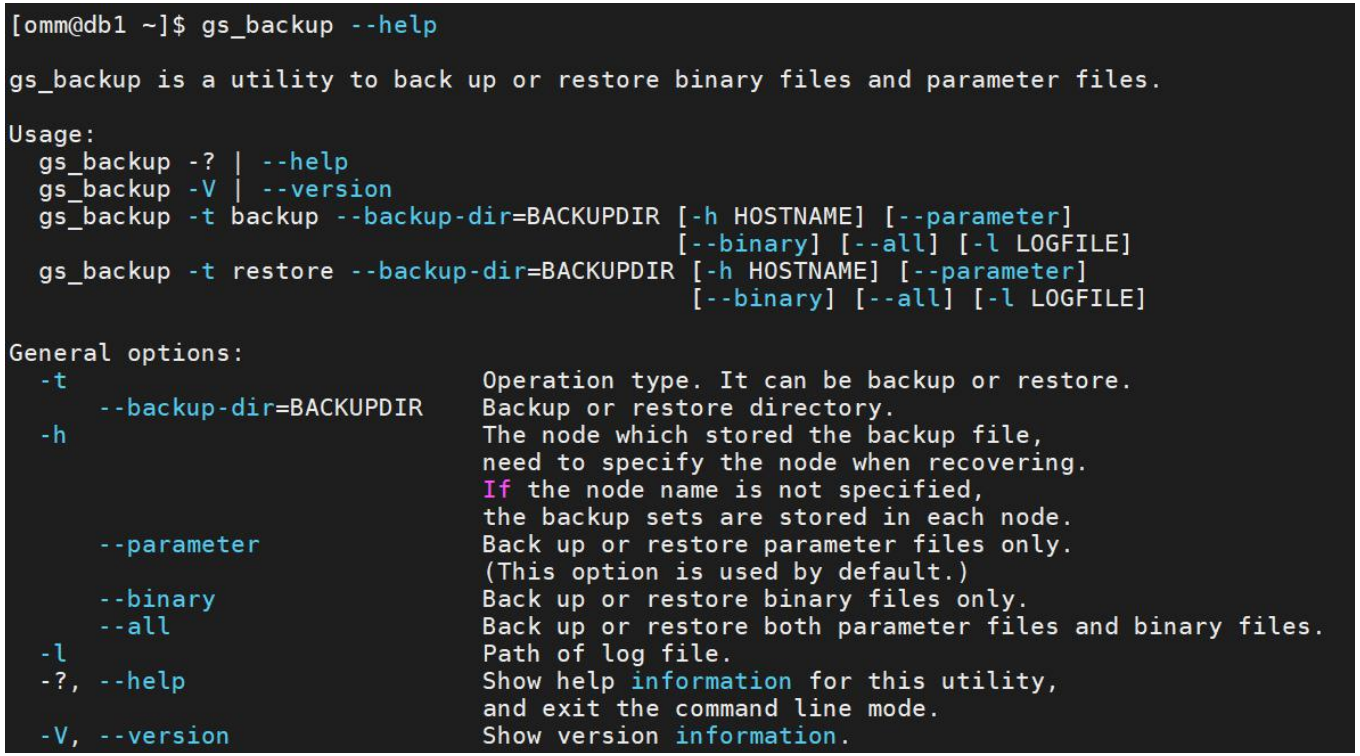
gs_backup概述

gs_backup工具可以帮助用户备份和恢复openGauss数据库二进制程序和参数文件等。

➢ **前提条件**

1. gs_backup命令备份的是openGauss数据库二进制程序和参数文件，并非备份数据；
2. 如果没有使用-h参数指定节点，则备份时会备份到所有节点的备份目录中，恢复时也会从所有节点的备份目录中读取备份文件。

➢ **语法**



```
[omm@db1 ~]$ gs_backup --help

gs_backup is a utility to back up or restore binary files and parameter files.

Usage:
  gs_backup -? | --help
  gs_backup -V | --version
  gs_backup -t backup --backup-dir=BACKUPDIR [-h HOSTNAME] [--parameter]
                                             [--binary] [--all] [-l LOGFILE]
  gs_backup -t restore --backup-dir=BACKUPDIR [-h HOSTNAME] [--parameter]
                                              [--binary] [--all] [-l LOGFILE]

General options:
  -t                                Operation type. It can be backup or restore.
      --backup-dir=BACKUPDIR        Backup or restore directory.
  -h                                The node which stored the backup file,
                                    need to specify the node when recovering.
                                    If the node name is not specified,
                                    the backup sets are stored in each node.
      --parameter                   Back up or restore parameter files only.
                                    (This option is used by default.)
      --binary                      Back up or restore binary files only.
      --all                         Back up or restore both parameter files and binary files.
  -l                                Path of log file.
  -?, --help                        Show help information for this utility,
                                    and exit the command line mode.
  -V, --version                     Show version information.
```

gs_backup示例

- **备份：使用gs_backup脚本备份openGauss主机的二进制程序和参数文件**

```
[omm@db1 ~]$ gs_backup -t backup --backup-dir=/home/omm/backup --all -l /home/omm/backup/log20200818.log
Parsing configuration files.
Successfully parsed the configuration file.
Performing remote backup.
Remote backup succeeded.
Successfully backed up cluster files.

[omm@db1 ~]$ cd /home/omm/backup/
[omm@db1 backup]$ ls
binary.tar  gs_local-2020-08-18_215240.log  log20200818-2020-08-18_215239.log  parameter.tar
[root@db1 ~]# tree -L 2 /home/omm/backup/
/home/omm/backup/
├── app_0bd0ce80
│   ├── bin
│   ├── etc
│   ├── include
│   ├── lib
│   └── share
├── binary_db1.opengauss.com.tar
├── binary.tar
├── gs_local-2020-08-18_215240.log
├── log20200818-2020-08-18_215239.log
├── parameter_db1.opengauss.com
│   ├── 6001_pg_hba.conf
│   ├── 6001_postgresql.conf
│   └── HOSTNAME
├── parameter_db1.opengauss.com.tar
├── parameter_db2.opengauss.com
│   ├── 6002_pg_hba.conf
│   ├── 6002_postgresql.conf
│   └── HOSTNAME
├── parameter_db2.opengauss.com.tar
└── parameter.tar

8 directories, 13 files
```

- **恢复：使用gs_backup脚本恢复主/备节点openGauss的二进制程序和参数文件**

```
## 恢复前请先确保恢复目标目录存在cluster_static_config文件
[omm@db1 gauss]$ mkdir -p /gauss/app_0bd0ce80/bin/
[omm@db1 gauss]$ cp /home/omm/backup/app_0bd0ce80/bin/cluster_static_config
/gauss/app_0bd0ce80/bin/
[omm@db1 gauss]$ ln -s app_0bd0ce80 app

[omm@db1 gauss]$ gs_backup -t restore --backup-dir=/home/omm/backup/   --all
Parsing configuration files.
Successfully parsed the configuration file.
Performing remote restoration.
Successfully restored cluster files.
```

openGauss

Gauss松鼠会

感谢聆听

Gauss松鼠会公众号

Gauss松鼠会小助手