openGauss 3.0.0

管理员指南

文档版本 01

发布日期 2022-03-31





版权所有 © 华为技术有限公司 2022。 保留一切权利。

非经本公司书面许可,任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部,并不得以任何形式传播。

商标声明



nuawe和其他华为商标均为华为技术有限公司的商标。 本文档提及的其他所有商标或注册商标,由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束,本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定,华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因,本文档内容会不定期进行更新。除非另有约定,本文档仅作为使用指导,本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址: 深圳市龙岗区坂田华为总部办公楼 邮编: 518129

网址: https://www.huawei.com

客户服务邮箱: support@huawei.com

客户服务电话: 4008302118

目录

1 启停 openGauss	1
2 状态查询	3
3 实例主备切换	6
4 生成配置文件	9
5 例行维护	11
5.1 日维护检查项	11
5.2 检查操作系统参数	13
5.3 检查 openGauss 健康状态	15
5.4 检查数据库性能	24
5.5 检查和清理日志	26
5.6 检查时间一致性	28
5.7 检查应用连接数	28
5.8 例行维护表	30
5.9 例行重建索引	31
5.10 数据安全维护建议	32
5.11 慢 SQL 诊断	33
6 备份与恢复	34
6.1 概述	34
6.2 物理备份恢复	38
6.2.1 gs_basebackup	38
6.2.2 PITR 恢复	42
6.2.3 gs_probackup	44
6.3 逻辑备份恢复	53
6.3.1 gs_dump	54
6.3.2 gs_dumpall	64
6.3.3 gs_restore	69
6.3.4 gs_backup	75
6.4 闪回恢复	77
6.4.1 闪回查询	77
6.4.2 闪回表	78
6.4.3 闪回 DROP/TRUNCATE	78

openGauss	
管理员指南	

7 高危操作	乍一览表	81
8 日志参	¥	83

1 启停 openGauss

启动 openGauss

步骤1 以操作系统用户omm登录数据库主节点。

步骤2 使用以下命令启动openGauss。

gs_om -t start

山 说明

双机启动必须以双机模式启动, 若中间过程以单机模式启动, 则必须修复才能恢复双机关系, 用gs_ctl build进行修复,gs_ctl的使用方法请参见《工具参考》。

----结束

停止 openGauss

步骤1 以操作系统用户omm登录数据库主节点。

步骤2 使用以下命令停止openGauss。

gs_om -t stop

山 说明

启停节点及AZ的操作请参见《工具参考》中"服务端工具 > gs_om"章节。

----结束

示例

启动openGauss:

gs_om -t start Starting cluster.

Successfully started.

停止openGauss:

gs_om -t stop Stopping cluster.

Successfully stopped cluster.

End stop cluster.

错误排查

如果启动openGauss或者停止openGauss服务失败,请根据日志文件中的日志信息排查错误,参见**8 日志参考**。

如果是超时导致启动失败,可以执行如下命令,设置启动超时时间,默认超时时间为300s。

gs_om -t start --time-out=300

2 状态查询

背景信息

openGauss支持查看整个openGauss的状态,通过查询结果确认openGauss或者单个主机的运行状态是否正常。

前提条件

openGauss已经启动。

操作步骤

步骤1 以操作系统用户omm登录数据库主节点。

步骤2 使用如下命令查询openGauss状态:

gs_om -t status --detail

openGauss状态显示结果的参数说明请参见表2-1。

若要查询某主机上的实例状态,请在命令中增加"-h"项。示例如下:

gs_om -t status -h plat2

其中,plat2为待查询主机的名称。

----结束

参数说明

表 2-1 节点角色说明

字段	字段含义	字段值
cluster_state	openGauss状态。 显示整个 openGauss是否运 行正常。	Normal:表示openGauss可用,且数据有 冗余备份。所有进程都在运行,主备关系 正常。
		● Unavailable:表示openGauss不可用。
		Degraded:表示openGauss可用,但存在 故障的数据库节点、数据库主节点实例。

字段	字段含义	字段值
node	主机名称	表示该实例所在的主机名称。多AZ时会显示 AZ编号。
node_ip	主机IP	表示该实例所在的主机IP。
instance	实例ID	表示该实例的ID。
state	实例角色	• Normal: 表示单主机实例
		● Primary:表示实例为主实例。
		● Standby:表示实例为备实例。
		● Cascade Standby: 表示实例为级联备实例。
		● Pending:表示该实例在仲裁阶段。
		● Unknown:表示实例状态未知。
		● Down:表示实例处于宕机状态。
		● Abnormal:表示节点处于异常状态。
		● Manually stopped:表示节点已经被手动 停止。

每个角色也存在不同的状态,例如启动,连接等,其各个状态说明如下

表 2-2 节点状态说明

状态	字段含义
Normal	表示节点启动正常
Need repair	当前节点需要修复
Starting	节点正在启动中
Wait promoting	节点正等待升级中, 例如备机向主机发送升级请求后, 正在等待主机 回应时的状态
Promoting	备节点正在升级为主节点的状态
Demoting	节点正在降级中, 如主机正在降为备机中
Building	备机启动失败, 需要重建,
Catchup	备节点正在追赶主节点
Coredump	节点程序崩溃
Unknown	节点状态未知

当节点出现Need repair状态时,可能需要对该节点进行重建使其恢复正常,通常情况下,节点重建原因说明如下:

表3 节点重建原因说明

状态	字段含义
Normal	表示节点启动正常
WAL segment removed	主机日志WAL日志不存在,备机日志比主机日志新
Disconnect	备机不能连接主机
Version not matched	主备二进制版本不一致
Mode not matched	主备角色不匹配, 例如两个备机互联
System id not matched	主备数据库系统id不一致, 主备双机要求System ID必须一致
Timeline not matched	日志时间线不一致
Unknown	其他原因

示例

查看openGauss详细状态信息,含实例状态信息。

3 实例主备切换

操作场景

openGauss在运行过程中,数据库管理员可能需要手工对数据库节点做主备切换。例如发现数据库节点主备failover后需要恢复原有的主备角色,或怀疑硬件故障需要手动进行主备切换。级联备机不能直接转换为主机,只能先通过switchover或failover成为备机,然后再成为主机。

□ 说明

- 主备切换为维护操作,确保openGauss状态正常,所有业务结束后,再进行切换操作。
- 在开启极致rto时,不支持级联备机。级联备机因为极致rto开启情况下备机不支持连接而无法同步数据。

操作步骤

步骤1 以操作系统用户omm登录数据库任意节点,执行如下命令,查看主备情况。 gs om -t status --detail

gs_om t status detait

步骤2 以操作系统用户omm登录准备切换为主节点的备节点,执行如下命令。

gs_ctl switchover -D /home/omm/cluster/dn1/

/home/omm/cluster/dn1/为备数据库节点的数据目录。

须知

对于同一数据库,上一次主备切换未完成,不能执行下一次切换。对于业务正在操作时,发起switchover,可能主机的线程无法停止导致switchover显示超时,实际后台仍然在运行,等主机线程停止后,switchover即可完成。比如在主机删除一个大的分区表时,可能无法响应switchover发起的信号。

主机故障时,可以在备机执行如下命令。

gs ctl failover -D /home/omm/cluster/dn1/

步骤3 switchover成功后,执行如下命令记录当前主备机器信息。

gs_om -t refreshconf

----结束

示例

将数据库节点备实例切换为主实例。

1.查询数据库状态。

```
gs_om -t status --detail
[ Cluster State ]
cluster_state : Normal
redistributing: No
current_az : AZ_ALL
[ Datanode State ]
node
                       node_ip
                                   port
                                                     instance
                                                                              state
1 pekpopgsci00235 10.244.62.204
                                 5432
                                          6001 /home/wuqw/cluster/dn1/ P Primary Normal
2 pekpopgsci00238 10.244.61.81
                                 5432
                                          6002 /home/wuqw/cluster/dn1/ S Standby Normal
```

2.登录备节点,进行主备切换。另外,switchover级联备机后,级联备机成为备机,备机降为级联备。

```
gs_ctl switchover -D /home/wuqw/cluster/dn1/
[2020-06-17 14:28:01.730] [24438] [] [gs_ctl]: gs_ctl switchover ,datadir is -D "/home/wuqw/cluster/dn1"
[2020-06-17 14:28:01.730] [24438] [] [gs_ctl]: switchover term (1)
[2020-06-17 14:28:01.768] [24438] [] [gs_ctl]: waiting for server to switchover.....................
[2020-06-17 14:28:11.175] [24438] [] [gs_ctl]: done
[2020-06-17 14:28:11.175] [24438] [] [gs_ctl]: switchover completed (/home/wuqw/cluster/dn1)
```

3.保存数据库主备机器信息。

```
gs_om -t refreshconf
Generating dynamic configuration file for all nodes.
Successfully generated dynamic configuration file.
```

错误排查

如果switchover过程中出现失败,请根据日志文件中的日志信息排查错误,参见8日志参考。

异常处理

异常判断标准如下:

- 业务压力下,主备实例切换时间长,这种情况不需要处理。
- 其他备机正在build的情况下,主机需要发送日志到备机后,才能降备,导致主备 切换时间长。这种情况不需要处理,但应尽量避免build过程中进行主备切换。
- 切换过程中,因网络故障、磁盘满等原因造成主备实例连接断开,出现双主现象时,此时请参考如下步骤修复。

▲ 警告

出现双主状态后,请按如下步骤恢复成正常的主备状态。否则可能会造成数据丢 失。

1. 执行以下命令查询数据库当前的实例状态。

gs_om -t status --detail

若查询结果显示两个实例的状态都为Primary,这种状态为异常状态。

2. 确定降为备机的节点,在节点上执行如下命令关闭服务。

gs_ctl stop -D /home/omm/cluster/dn1/

- 3. 执行以下命令,以standby模式启动备节点。 gs_ctl start -D /home/omm/cluster/dn1/ -M standby
- 4. 保存数据库主备机器信息。 gs_om -t refreshconf
- 5. 查看数据库状态,确认实例状态恢复。

4 生成配置文件

背景信息

在openGauss使用过程中,如果静态配置文件无意损坏后,会影响openGauss感知openGauss拓扑结构和主备关系gs_om工具生成静态配置文件,替换已经损坏的配置文件,保证openGauss的正常运行。

前置条件

无。

操作步骤

步骤1 以操作系统用户omm登录数据库主节点。

步骤2 执行如下命令会在本服务器指定目录下生成配置文件。

gs_om -t generateconf -X /opt/software/openGauss/clusterconfig.xml --distribute

/opt/software/openGauss/clusterconfig.xml为openGauss安装时XML配置文件的存放目录。

□ 说明

- 1. 执行命令后,日志信息中会有新文件的存放的目录。以一主两备环境为例,打开新文件存放 目录,会出现3个以主机名命名的配置文件,需要用这3个文件分别替换对应主机的配置文 件。
- 2. 若不使用--distribute参数,需执行**步骤3**将静态配置文件分配到对应节点;若使用--distribute参数,则会将生成的静态配置文件自动分配到对应节点,无需执行**步骤3**。

步骤3 (可选)分别替换3台主机的/opt/gaussdb/app/bin目录下损坏的静态配置文件。

这里以其中一台主机为例进行介绍。

mv /opt/huawei/wisequery/script/static_config_files/cluster_static_config_SIA1000056771 /opt/gaussdb/app/bin/cluster_static_config

----结束

示例

在openGauss中的任意主机上执行如下命令,生成配置文件:

gs_om -t generateconf -X /opt/software/openGauss/clusterconfig.xml --distribute Generating static configuration files for all nodes.

Creating temp directory to store static configuration files.

Successfully created the temp directory.

Generating static configuration files.

Successfully generated static configuration files.

Static configuration files for all nodes are saved in /opt/huawei/Bigdata/mppdb/wisequery/script/static_config_files.

Distributing static configuration files to all nodes.

Successfully distributed static configuration files.

打开生成的配置文件目录,会看到新生成的3个文件:

cd /opt/huawei/Bigdata/mppdb/wisequery/script/static_config_files

total 456

-rwxr-xr-x 1 omm dbgrp 155648 2016-07-13 15:51 cluster_static_config_plat1

-rwxr-xr-x 1 omm dbgrp 155648 2016-07-13 15:51 cluster_static_config_plat2

-rwxr-xr-x 1 omm dbgrp 155648 2016-07-13 15:51 cluster_static_config_plat3

5 例行维护

- 5.1 日维护检查项
- 5.2 检查操作系统参数
- 5.3 检查openGauss健康状态
- 5.4 检查数据库性能
- 5.5 检查和清理日志
- 5.6 检查时间一致性
- 5.7 检查应用连接数
- 5.8 例行维护表
- 5.9 例行重建索引

5.10 数据安全维护建议

为保证openGauss数据库中的数据安全,避免丢失数据,非法访问数据等事故发生,请仔细阅读以下内容。

5.11 慢SQL诊断

5.1 日维护检查项

检查 openGauss 状态

通过openGauss提供的工具查询数据库和实例状态,确认数据库和实例都处于正常的运行状态,可以对外提供数据服务。

- 检查实例状态
 - gs_check -U omm -i CheckClusterState
- 检查参数 openGauss=# **SHOW** *parameter_name*;
- 修改参数 gs_guc reload -D /gaussdb/data/dbnode -c "paraname=value"

检查锁信息

锁机制是数据库保证数据一致性的重要手段,检查相关信息可以检查数据库的事务和 运行状况。

- 查询数据库中的锁信息。 openGauss=# **SELECT** * **FROM pg_locks**;
- 查询等待锁的线程状态信息。 openGauss=# **SELECT** * **FROM pg_thread_wait_status WHERE wait_status** = '**acquire lock**';
- 结束系统进程

查找正在运行的系统进程,然后使用kill命令结束此进程。

ps ux kill -9 *pid*

统计事件数据

SQL语句长时间运行会占用大量系统资源,用户可以通过查看事件发生的时间,占用内存大小来了解现在数据库运行状态。

• 查询事件的时间

查询事件的线程启动时间、事务启动时间、SQL启动时间以及状态变更时间。openGauss=# SELECT backend_start,xact_start,query_start,state_change FROM pg_stat_activity;

- 查询当前服务器的会话计数信息
 openGauss=# SELECT count(*) FROM pg_stat_activity;
- 查询系统级统计信息

查询当前使用内存最多的会话信息。

openGauss=# SELECT * FROM pv_session_memory_detail() ORDER BY usedsize desc limit 10;

对象检查

表、索引、分区、约束等是数据库的核心存储对象,其核心信息和对象维护是DBA重要的日常工作。

- 查看表的详细信息 openGauss=# \d+ table_name
- 查询表统计信息 openGauss=# **SELECT * FROM pg_statistic**;
- 查看索引的详细信息 openGauss=# \d+ index_name
- 查询分区表信息
 openGauss=# SELECT * FROM pg_partition;
- 收集统计信息 使用ANALYZE语句收集数据库相关的统计信息。使用VACUUM语句可以回收空间并更新统计信息。
- 查询约束信息 openGauss=# **SELECT** * **FROM pg_constraint**;

SQL 报告检查

使用EXPLAIN语句查看执行计划。

备份

数据备份重于一切,日常应检查备份执行情况,并检查备份有效性,确保备份能够保 障数据安全,备份安全加密也应兼顾。

- 指定用户导出数据库:
 - gs_dump dbname -p port -f out.sql -U user_name -W password
- 导出schema:
 - gs_dump dbname -p port -n schema_name -f out.sql
- 导出table:
 - gs_dump dbname -p port -t table_name -f out.sql

基本信息检查

基本信息包括版本、组件、补丁集等信息,定期检查数据库信息并登记在案是数据库 生命周期管理的重要内容之一。

- 版本信息 openGauss=# SELECT version();
- 容量检查

openGauss=# SELECT pg_table_size('table_name'); openGauss=# SELECT pg_database_size('database_name');

5.2 检查操作系统参数

检查办法

通过openGauss提供的gs_checkos工具可以完成操作系统状态检查。

前提条件

- 当前的硬件和网络环境正常。
- 各主机间root互信状态正常。
- 只能使用root用户执行gs_checkos命令。

操作步骤

步骤1 以root用户身份登录任意一台服务器。

步骤2 执行如下命令对openGauss节点服务器的OS参数进行检查。

gs checkos -i A

检查节点服务器的OS参数的目的是为了保证openGauss正常通过预安装,并且在安装 成功后可以安全高效的运行。详细的检查项目请参见《工具参考》中"服务端工具> gs_checkos"章节。

----结束

示例

执行gs_checkos前需要先使用gs_preinstall工具执行前置脚本,准备环境。以参数 "A"为例。

gs_checkos -i A Checking items: A1. [OS version status]

: Normal

```
A2. [ Kernel version status ]
                                                  : Normal
  A3. [ Unicode status ]
                                                 : Normal
  A4. [ Time zone status ]
                                                  : Normal
  A5. [ Swap memory status ]
                                                     : Normal
                                                       : Normal
  A6. [ System control parameters status ]
  A7. [ File system configuration status ]
                                                      : Normal
  A8. [ Disk configuration status ]
                                                    : Normal
  A9. [ Pre-read block size status ]
                                                    : Normal
  A10.[ IO scheduler status ]
                                                   : Normal
  A11.[ Network card configuration status ]
                                                        : Normal
  A12.[ Time consistency status ]
                                                     : Warning
  A13.[ Firewall service status ]
                                                   · Normal
  A14.[ THP service status ]
                                                   : Normal
Total numbers:14. Abnormal numbers:0. Warning number:1.
```

以参数"B"为例。

```
gs_checkos -i B
Setting items:
  B1. [ Set system control parameters ]
                                                       : Normal
  B2. [ Set file system configuration value ]
                                                       : Normal
  B3. [ Set pre-read block size value ]
                                                     : Normal
  B4. [ Set IO scheduler value ]
                                                    : Normal
  B5. [ Set network card configuration value ]
                                                         : Normal
  B6. [ Set THP service ]
                                                  : Normal
  B7. [ Set RemoveIPC value ]
                                                     : Normal
  B8. [ Set Session Process ]
                                                   : Normal
Total numbers:6. Abnormal numbers:0. Warning number:0.
```

异常处理

使用gs_checkos检查openGauss状态,可以使用如下命令查看详细的错误信息。gs_checkos -i A --detail

其中,Abnormal为必须处理项,影响openGauss安装。Warning可以不处理,不会影响openGauss安装。

- 如果操作系统版本(A1)检查项检查结果为Abnormal,需要将不属于混编范围的操作系统版本替换为混编范围内的操作系统版本。
- 如果内核版本(A2)检查项检查结果为Warning,则表示openGauss内平台的内核版本不一致。
- 如果Unicode状态(A3)检查项检查结果为Abnormal,需要将各主机的字符集设置为相同的字符集,可以在/etc/profile文件中添加"export LANG=XXX"(XXX为Unicode编码)。

vim /etc/profile

- 如果时区状态(A4)检查项检查结果为Abnormal,需要将各主机的时区设置为相同时区,可以将/usr/share/zoneinfo/目录下的时区文件拷贝为/etc/localtime文件。
 - cp /usr/share/zoneinfo/\$主时区/\$次时区/etc/localtime
- 如果交换内存状态(A5)检查项检查结果为Abnormal,可能是因为swap空间大于mem空间,可减小Swap解决或者增大Mem空间解决。
- 如果系统控制参数(A6)检查项检查结果为Abnormal,可以使用以下两种方法进行设置。
 - 可以使用如下命令进行设置。
 gs_checkos -i B1
 - 根据错误提示信息,在/etc/sysctl.conf文件中进行设置。然后执行sysctl -p使 其生效。

vim /etc/sysctl.conf

如果文件系统配置状态(A7)检查项检查结果为Abnormal,可以使用如下命令进行设置。

gs_checkos -i B2

● 如果磁盘配置状态(A8)检查项检查结果为Abnormal,需修改磁盘挂载格式为: "rw,noatime,inode64,allocsize=16m"。

使用linux的man mount命令挂载XFS选项:

rw,noatime,inode64,allocsize=16m

也可以在/etc/fstab文件中设定XFS选项。如下示例:

/dev/data /data xfs rw,noatime,inode64,allocsize=16m 0 0

如果预读块大小(A9)检查项检查结果为Abnormal,可以使用如下命令进行设置。

gs_checkos -i B3

如果IO调度状态(A10)检查项检查结果为Abnormal,可以使用如下命令进行设置。

gs_checkos -i B4

如果网卡配置状态(A11)检查项检查结果为Warning,可以使用如下命令进行设置。

gs_checkos -i B5

- 如果时间一致性(A12)检查项检查结果为Abnormal,需检查是否安装ntp服务,以及ntp服务是否启动;并与ntp时钟源同步。
- 如果防火墙状态(A13)检查项检查结果为Abnormal,需关闭防火墙服务。使用如下命令进行设置。
 - Centos7/openEuler:
 systemctl stop firewalld
 systemctl disable firewalld
- 如果THP服务(A14)检查项检查结果为Abnormal,可以使用如下命令进行设置。

gs_checkos -i B6

5.3 检查 openGauss 健康状态

检查办法

通过openGauss提供的gs_check工具可以开展openGauss健康状态检查。

注意事项

- 扩容新节点检查只能在root用户下执行,其他场景都必须在omm用户下执行。
- 必须指定-i或-e参数,-i会检查指定的单项,-e会检查对应场景配置中的多项。
- 如果-i参数中不包含root类检查项或-e场景配置列表中没有root类检查项,则不需要交互输入root权限的用户及其密码。
- 可使用--skip-root-items跳过检查项中包含的root类检查,以免需要输入root权限用户及密码。
- 检查扩容新节点与现有节点之间的一致性,在现有节点执行gs_check命令指定-hosts参数进行检查,其中hosts文件中需要写入新节点IP。

操作步骤

方式1:

步骤1 以操作系统用户omm登录数据库主节点。

步骤2 执行如下命令对openGauss数据库状态进行检查。

gs_check -i CheckClusterState

其中,-i指定检查项,注意区分大小写。格式: -i CheckClusterState、-i CheckCPU或-i CheckClusterState,CheckCPU。

取值范围为所有支持的检查项名称,详细列表请参见《工具参考》中"服务端工具 > gs_checkos > openGauss状态检查表",用户可以根据需求自己编写新检查项。

----结束

方式2:

步骤1 以操作系统用户omm登录数据库主节点。

步骤2 执行如下命令对openGauss数据库进行健康检查。

gs_check -e inspect

其中,-e指定场景名,注意区分大小写。格式: -e inspect或-e upgrade。

取值范围为所有支持的巡检场景名称,默认场景包括: inspect(例行巡检)、upgrade(升级前巡检)、binary_upgrade(就地升级前巡检)、health(健康检查巡检)、slow_node(节点)、longtime(耗时长巡检)、install(安装),用户可以根据需求自己编写场景。

----结束

openGauss巡检的主要作用是在openGauss运行过程中,检查整个openGauss状态是否正常,或者重大操作前(升级、扩容),确保openGauss满足操作所需的环境条件和状态条件。详细的巡检项目和场景请参见《工具参考》中"服务端工具 > gs_checkos > openGauss状态检查表"。

示例

执行单项检查结果:

perfadm@lfgp000700749:/opt/huawei/perfadm/tool/script> gs_check -i CheckCPU Parsing the check items config file successfully Distribute the context file to remote hosts successfully Start to health check for the cluster. Total Items:1 Nodes:3

Checking... [========] 1/1

Start to analysis the check result

CheckCPU.....OK

The item run on 3 nodes. success: 3

Analysis the check result successfully

Success. All check items run completed. Total:1 Success:1 Failed:0

For more information please refer to /opt/huawei/wisequery/script/gspylib/inspection/output/ CheckReport_201902193704661604.tar.gz

本地执行结果:

perfadm@lfgp000700749:/opt/huawei/perfadm/tool/script> gs_check -i CheckCPU -L

2017-12-29 17:09:29 [NAM] CheckCPU

2017-12-29 17:09:29 [STD] 检查主机CPU占用率,如果idle 大于30%并且iowait 小于 30%.则检查项通过,否则检查项不通过

2017-12-29 17:09:29 [RST] OK

2017-12-29 17:09:29 [RAW]

Linux 4.4.21-69-default (lfgp000700749) 12/29/17 _x86_64_

17:09:24	CPU	%user	%ni	ce %sy	stem 9	%iowait	%steal	%idle
17:09:25	all	0.25	0.00	0.25	0.00	0.00	99.50	
17:09:26	all	0.25	0.00	0.13	0.00	0.00	99.62	
17:09:27	all	0.25	0.00	0.25	0.13	0.00	99.37	
17:09:28	all	0.38	0.00	0.25	0.00	0.13	99.25	
17:09:29	all	1.00	0.00	0.88	0.00	0.00	98.12	
Average:	all	0.43	0.00	0.35	0.03	0.03	99.17	

执行场景检查结果: [perfadm@SIA1000131072 Check]\$ qs_check -e inspect Parsing the check items config file successfully The below items require root privileges to execute: [CheckBlockdev CheckIOrequestqueue CheckIOConfigure CheckCheckMultiQueue CheckFirewall CheckSshdService CheckSshdConfig CheckCrondService CheckBootItems CheckFilehandle CheckNICModel CheckDropCache] Please enter root privileges user[root]:root Please enter password for user[root]: Please enter password for user[root] on the node[10.244.57.240]: Check root password connection successfully Distribute the context file to remote hosts successfully Start to health check for the cluster. Total Items:59 Nodes:2 Checking...] 21/57 [======] 59/57 Checking... Start to analysis the check result CheckClusterState.....OK The item run on 2 nodes. success: 2 CheckDBParams.....OK The item run on 1 nodes. success: 1 CheckDebugSwitch.....OK The item run on 2 nodes. success: 2 CheckDirPermissions.....OK
The item run on 2 nodes. success: 2 CheckReadonlyMode.....OK The item run on 1 nodes. success: 1 CheckEnvProfile.....OK The item run on 2 nodes. success: 2 (consistent) The success on all nodes value: GAUSSHOME /usr1/gaussdb/app LD_LIBRARY_PATH /usr1/gaussdb/app/lib /usr1/gaussdb/app/bin CheckBlockdev.....OK The item run on 2 nodes. success: 2 CheckCurConnCount.....OK The item run on 1 nodes. success: 1 CheckCursorNum.....OK The item run on 1 nodes. success: 1 CheckPgxcgroup.....OK The item run on 1 nodes. success: 1 CheckDiskFormat.....OK The item run on 2 nodes. success: 2 CheckSpaceUsage.....OK The item run on 2 nodes. success: 2 CheckInodeUsage.....OK The item run on 2 nodes. success: 2

CheckSwapMemory.....OK

The item run on 2 nodes. success: 2
CheckLogicalBlockOK The item run on 2 nodes. success: 2
ChecklOrequestqueueWARNING The item run on 2 nodes. warning: 2 The warning[host240,host157] value: On device (vdb) 'IO Request' RealValue '256' ExpectedValue '32768' On device (vda) 'IO Request' RealValue '256' ExpectedValue '32768'
CheckMaxAsylOrequestsOK The item run on 2 nodes. success: 2
ChecklOConfigureOK The item run on 2 nodes. success: 2
CheckMTUOK The item run on 2 nodes. success: 2 (consistent) The success on all nodes value: 1500
CheckPingOK The item run on 2 nodes. success: 2
CheckRXTXNG The item run on 2 nodes. ng: 2 The ng[host240,host157] value: NetWork[eth0] RX: 256 TX: 256
CheckNetWorkDropOK The item run on 2 nodes. success: 2
CheckMultiQueueOK The item run on 2 nodes. success: 2
CheckEncodingOK The item run on 2 nodes. success: 2 (consistent) The success on all nodes value: LANG=en_US.UTF-8
CheckFirewallOK The item run on 2 nodes. success: 2
CheckKernelVerOK The item run on 2 nodes. success: 2 (consistent) The success on all nodes value: 3.10.0-957.el7.x86_64
CheckMaxHandleOK The item run on 2 nodes. success: 2
CheckNTPDOK host240: NTPD service is running, 2020-06-02 17:00:28 host157: NTPD service is running, 2020-06-02 17:00:06
CheckOSVerOK host240: The current OS is centos 7.6 64bit. host157: The current OS is centos 7.6 64bit.
CheckSysParamsWARNING The item run on 2 nodes. warning: 2 The warning[host240,host157] value: Warning reason: variable 'net.ipv4.tcp_retries1' RealValue '3' ExpectedValue '5'.

Warning reason: variable 'net.ipv4.tcp_syn_retries' RealValue '6' ExpectedValue '5'.
CheckTHPOK The item run on 2 nodes. success: 2
CheckTimeZoneOK The item run on 2 nodes. success: 2 (consistent) The success on all nodes value: +0800
CheckCPUOK The item run on 2 nodes. success: 2
CheckSshdServiceOK The item run on 2 nodes. success: 2
Warning reason: UseDNS parameter is not set; expected: no
CheckCrondServiceOK The item run on 2 nodes. success: 2
CheckStackOK The item run on 2 nodes. success: 2 (consistent) The success on all nodes value: 8192
CheckSysPortRangeOK The item run on 2 nodes. success: 2
CheckMemInfoOK The item run on 2 nodes. success: 2 (consistent) The success on all nodes value: totalMem: 31.260929107666016G
CheckHyperThreadOK The item run on 2 nodes. success: 2
CheckTableSpaceOK The item run on 1 nodes. success: 1
CheckSysadminUserOK The item run on 1 nodes. success: 1
CheckGUCConsistentOK All DN instance guc value is consistent.
CheckMaxProcMemoryOK The item run on 1 nodes. success: 1
CheckBootItemsOK The item run on 2 nodes. success: 2
CheckHashIndexOK The item run on 1 nodes. success: 1
CheckPgxcRedistbOK The item run on 1 nodes. success: 1
CheckNodeGroupNameOK The item run on 1 nodes. success: 1
CheckTDDateOK The item run on 1 nodes. success: 1
CheckDilateSysTabOK The item run on 1 nodes. success: 1

CheckKeyProAdj.....OK The item run on 2 nodes. success: 2 CheckProStartTime.....WARNING host157 STARTED COMMAND Tue Jun 2 16:57:18 2020 /usr1/dmuser/dmserver/metricdb1/server/bin/gaussdb --single_node -D /usr1/ dmuser/dmb1/data -p 22204 Mon Jun 1 16:15:15 2020 /usr1/gaussdb/app/bin/gaussdb -D /usr1/gaussdb/data/dn1 -M standby CheckFilehandle.....OK The item run on 2 nodes. success: 2 CheckRouting.....OK The item run on 2 nodes. success: 2 CheckNICModel.....OK
The item run on 2 nodes. success: 2 (consistent) The success on all nodes value: version: 1.0.0 model: Red Hat, Inc. Virtio network device CheckDropCache.....WARNING
The item run on 2 nodes. warning: 2 The warning[host240,host157] value: No DropCache process is running CheckMpprcFile.....NG The item run on 2 nodes. ng: 2 The ng[host240,host157] value: There is no mpprc file Analysis the check result successfully Failed. All check items run completed. Total:57 Success:50 Warning:5 NG:2 For more information please refer to /usr1/gaussdb/tool/script/gspylib/inspection/output/ CheckReport_inspect611.tar.gz

异常处理

如果发现检查结果异常,可以根据以下内容进行修复。

表 5-1 检查 openGauss 运行状态

检查项	异常状态	处理方法
CheckClusterS tate(检查 openGauss状	openGauss未启 动或openGauss 实例未启动	使用以下命令启动openGauss及实例。 gs_om -t start
态)	openGauss状态 异常或 openGauss实例 异常	检查各主机、实例状态,根据状态信息进行排 查。 gs_check -i CheckClusterState
CheckDBPara ms(检查数据 库参数)	数据库参数错误	通过gs_guc工具修改数据库参数为指定值。
CheckDebugS witch(检查调 试日志)	日志级别不正确	使用gs_guc工具将log_min_messages改为指定内容。

检查项	异常状态	处理方法
CheckDirPermi ssions(检查目 录权限)	路径权限错误	修改对应目录权限为指定数值(750/700)。 chmod 750 DIR
CheckReadonl yMode(检查 只读模式)	只读模式被打开	确认数据库节点所在磁盘使用率未超阈值(默认60%)目未在执行其他运维操作。gs_check -i CheckDataDiskUsage ps ux 使用gs_guc工具关闭openGauss只读模式。gs_guc reload -N all -I all -c 'default_transaction_read_only = off' gs_guc reload -N all -I all -c 'default_transaction_read_only = off'
CheckEnvProfil e(检查环境变 量)	环境变量不一致	重新执行前置更新环境变量信息。
CheckBlockdev (检查磁盘预 读块)	磁盘预读块大小 不为16384	使用gs_checkos设置预读块大小为16384KB, 并写入自启动文件。 gs_checkos -i B3
CheckCursorN um(检查游标 数)	检查游标数失败	检查数据库能否正常连接,openGauss状态是 否正常。
CheckPgxcgro up(检查重分 布状态)	有未完成重分布 的pgxc_group表	继续完成扩容或缩容的数据重分布操作。 gs_expand、gs_shrink
CheckDiskFor mat(检查磁 盘配置)	各节点磁盘配置 不一致	将各节点的磁盘规格改为相同。
CheckSpaceUs age(检查磁盘 空间使用率)	磁盘可用空间不 足	清理或扩展对应目录所在的磁盘。
CheckInodeUs age(检查磁盘 索引使用率)	磁盘可用索引不 足	清理或扩展对应目录所在的磁盘。
CheckSwapMe mory(检查交 换内存)	交换内存大于物 理内存	将交换内存调小或关闭。
CheckLogicalB lock(检查磁 盘逻辑块)	磁盘逻辑块大小 不为512	使用gs_checkos修改磁盘逻辑块大小为 512KB,并写入开机自启动文件。 gs_checkos -i B4
ChecklOreque stqueue(检查 IO请求)	IO请求值不为 32768	使用gs_checkos设置IO请求值为32768,并写入开机自启动文件。 gs_checkos -i B4

检查项	异常状态	处理方法
CheckCurConn Count(检查当 前连接数)	当前连接数超过 最大连接数的 90%	断开未使用的数据库主节点连接。
CheckMaxAsyl Orequests(检 查最大异步请 求)	最大异步请求值 小于104857600 或当前节点数据 库实例数) *1048576	使用gs_checkos设置最大异步请求值为 104857600和当前节点数据库实例数*1048576 中的最大值。 gs_checkos -i B4
CheckMTU (检查MTU 值)	MTU值不一致	设置各节点的MTU一致为1500或8192。 ifconfig eth* MTU 1500
CheckIOConfig ure(检查IO配 置)	IO配置不是 deadline	使用gs_checkos设置IO配置为deadline,并写 入开机自启动文件。 gs_checkos -i B4
CheckRXTX (检查RXTX 值)	网卡RX/TX值不 是4096	使用checkos设置openGauss使用的物理网卡 RX/TX值为4096 gs_checkos -i B5
CheckPing(检 查网络通畅)	存在openGauss IP无法ping通	检查异常ip间网络设置和状态、防火墙状态。
CheckNetWor kDrop(检查网 络丢包率)	网络通信丢包率 高于1%	检查对应IP间网络负载、状态。
CheckMultiQu eue(检查网卡 多队列)	未开启网卡多队 列并未将网卡中 断绑定到不同 CPU核心	开启网卡多队列并将网卡队列中断绑定到不同的CPU核心。
CheckEncodin g(检查编码格 式)	各节点编码格式 不一致	在/etc/profile中写入一致的编码信息。 echo "export LANG=XXX" >> /etc/profile
CheckFirewall (检查防火 墙)	防火墙未关闭	关闭防火墙服务。 redHat(CentOS)7.x: systemctl status firewalld.service redHat(CentOS)6.x: service iptables down SuSE: SuSEfirewall2 down
CheckKernelVe r(检查内核版 本)	内核版本不一致	使用gs_replace替换不一致的节点。 gs_replace
CheckMaxHan dle(检查最大 文件句柄数)	最大文件句柄数 小于1000000	设置91-nofile.conf/90-nofile.conf最大文件句 柄数软硬限制为1000000。 gs_checkos -i B2
CheckNTPD (检查时间同 步服务)	NTPD服务未开 启或时间误差超 过一分钟	开启NTPD服务并设置时钟一致。

检查项	异常状态	处理方法
CheckOSVer (检查操作系 统版本)	操作系统不被支持或不在同一混搭列表	使用gs_replace将异常节点替换为受支持系统或在同一混搭列表内的系统的节点。gs_replace
CheckSysPara ms(检查操作 系统参数)	操作系统参数设 置不满足要求	使用gs_checkos进行参数设置或手动设置。 gs_checkos -i B1 vim /etc/sysctl.conf
CheckTHP(检 查THP服务)	THP服务未开启	使用gs_checkos设置THP服务。 gs_checkos -i B6
CheckTimeZon e(检查时区)	时区不一致	设置各节点为同一时区。 cp /usr/share/zoneinfo/ <i>s主时区/s次时区</i> /etc/localtime
CheckCPU(检 查CPU)	CPU占用过高或 IO等待过高	进行CPU配置升级或磁盘性能升级。
CheckSshdSer vice(检查 SSHD服务)	未开启SSHD服 务	启动SSHD服务并写入开机自启动文件。 server sshd start echo "server sshd start" >> initFile
CheckSshdConf ig(检查SSHD 配置)	SSHD服务配置 错误	设置SSHD服务。 PasswordAuthentication=no; MaxStartups=1000; UseDNS=yes; ClientAliveInterval=10800/ClientAliveInterval=0 并重启服务: server sshd start
CheckCrondSe rvice(检查 Crond服务)	Crond服务未启 动	安装Crond服务并启用。
CheckStack (检查堆栈大 小)	堆栈大小小于 3072	使用gs_checkos设置为3072并重启堆栈值过小 进程。 gs_checkos -i B2
CheckSysPortR ange(检查系 统端口设置)	系统ip端口不在 预期范围内或 openGauss端口 在系统ip端口内	设置系统ip端口范围参数到26000-65535之中;设置openGauss端口在系统IP端口范围外。 vim /etc/sysctl.conf
CheckMemInf o(检查内存信 息)	各节点内存大小 不一致	使用相同规格的物理内存。
CheckHyperTh read(检查超 线程)	未开启CPU超线 程	开启CPU超线程。
CheckTableSp ace(检查表空 间)	表空间路径和 openGauss路径 存在嵌套或表空 间路径相互存在 嵌套	将表空间数据迁移到路径合法的表空间中。

5.4 检查数据库性能

检查办法

通过openGauss提供的性能统计工具gs_checkperf可以对硬件性能进行检查。

前提条件

- openGauss运行状态正常。
- 运行在数据库之上的业务运行正常。

操作步骤

步骤1 以操作系统用户omm登录数据库主节点。

步骤2 执行如下命令对openGauss数据库进行性能检查。

gs_checkperf

----结束

具体的性能统计项目请参见《工具参考》中"服务端工具 > gs_checkperf > 性能检查项"。

示例

以简要格式在屏幕上显示性能统计结果。

```
gs_checkperf -i pmk -U omm
Cluster statistics information:
  Host CPU busy time ratio
                                    : 1.43
  MPPDB CPU time % in busy time
                                         : 1.88
                                                   %
  Shared Buffer Hit ratio
                                   : 99.96
                                              %
  In-memory sort ratio
                                   : 100.00
                                 : 4
: 25
  Physical Reads
  Physical Writes
  DB size
                                 70
                                          MB
                                  : 25
  Total Physical writes
  Active SQL count
                                  : 2
                                    3
  Session count
```

异常处理

使用gs_checkperf工具检查openGauss性能状态后,如果发现检查结果发现异常,可以根据以下内容进行修复。

表 5-2 检查 openGauss 级别性能状态

异常状态	处理方法
主机CPU占有率高	1、更换和增加高性能的CPU。 2、使用top命令查看系统哪些进程的CPU占有率高,然后使用 kill命令关闭没有使用的进程。 top

异常状态	处理方法
openGauss CPU占有率高	1、更换和增加高性能的CPU。 2、使用top命令查看数据库哪些进程的CPU占有率高,然后使用kill命令关闭没有使用的进程。 top 3、使用gs_expand工具扩容,增加新的主机均衡CPU占有率。
共享内存命中率低	1、扩大内存。 2、使用如下命令查看操作系统配置文件/etc/sysctl.conf,调 大共享内存kernel.shmmax值。 vim /etc/sysctl.conf
内存中排序比率低	扩大内存。
I/O、磁盘使用率高	1、更换高性能的磁盘。 2、调整数据布局,尽量将I/O请求较合理的分配到所有物理磁盘中。 3、全库进行VACUUM FULL操作。 vacuum fult; 4、进行磁盘整理。 5、降低并发数。
事务统计	查询pg_stat_activity系统表,将不必要的连接断开。(登录 数据库后查询:openGauss=# \d+ pg_stat_activity;)

表 5-3 检查节点级别性能状态

异常状态	处理方法
CPU占有率高	1、更换和增加高性能的CPU。 2、使用top命令查看系统哪些进程的CPU占有率高,然后使用 kill命令关闭没有使用的进程。 top
内存使用率过高情 况	扩大或清理内存。
I/O使用率过高情况	1、更换高性能的磁盘。 2、进行磁盘清理。 3、尽可能用内存的读写代替直接磁盘I/O,使频繁访问的文件 或数据放入内存中进行操作处理。

表 5-4 会话/进程级别性能状态

异常状态	处理方法
CPU、内存、I/O使 用率过高情况	查看哪个进程占用CPU/内存高或I/O使用率高,若是无用的进程,则kill掉,否则排查具体原因。例如SQL执行占用内存大,查看是否SQL语句需要优化。

表 5-5 SSD 性能状态

异常状态	处理方法
SSD读写性能故障	使用以下命令查看SSD是否有故障,排查具体故障原因。 gs_checkperf -i SSD -U omm

5.5 检查和清理日志

日志是检查系统运行及故障定位的关键手段。建议按月度例行查看操作系统日志及数据库的运行日志。同时,随着时间的推移,日志的增加会占用较多的磁盘空间。建议 按月度清理数据库的运行日志。

检查操作系统日志

建议按月检查操作系统日志,排除操作系统运行异常隐患。

执行如下命令查看操作系统日志文件。

vim /var/log/messages

关注其中近一个月出现的kernel、error、fatal等字样,根据系统报警信息进行处理。

检查 openGauss 运行日志

数据库运行时,某些操作在执行过程中可能会出现错误,数据库依然能够运行。但是此时数据库中的数据可能已经发生不一致的情况。建议按月检查openGauss运行日志,及时发现隐患。

前提条件

- 收集日志的主机网络通畅且未宕机,数据库安装用户互信正常。
- 日志收集工具依赖操作系统工具如gstack,如果未安装该工具,则提示错误后, 跳过该收集项。

操作步骤

步骤1 以操作系统用户omm登录数据库主节点。

步骤2 执行如下命令收集数据库日志。

gs_collector --begin-time="20160616 01:01" --end-time="20160616 23:59"

20160616 01:01为日志的开始时间, 20160616 23:59为日志的结束时间。

步骤3 根据**步骤2**的界面输出提示,进入相应的日志收集目录,解压收集的日志,并检查数据库日志。

以下以日志收集路径"/opt/gaussdb/tmp/gaussdba_mppdb/collector 20160726 105158.tar.gz"为例进行操作。

tar -xvzf /opt/gaussdb/tmp/gaussdba_mppdb/collector_20160726_105158.tar.gz cd /opt/gaussdb/tmp/gaussdba_mppdb/collector_20160726_105158

----结束

示例

• 以--begin-time与--end-time为参数执行gs_collector命令。 gs_collector --begin-time="20160616 01:01" --end-time="20160616 23:59"

当显示类似如下信息表示日志已经归档。

Successfully collected files

All results are stored in /tmp/gaussdba_mppdb/collector_20160616_175615.tar.gz.

以--begin-time, --end-time与-h为参数执行gs_collector命令。
 gs_collector --begin-time="20160616 01:01" --end-time="20160616 23:59" -h plat2
 当显示类似如下信息表示日志已经归档。

Successfully collected files

All results are stored in /tmp/gaussdba_mppdb/collector_20160616_190225.tar.gz.

以--begin-time, --end-time与-f为参数执行gs_collector命令。
 gs_collector --begin-time="20160616 01:01" --end-time="20160616 23:59" -f /opt/software/gaussdb/output

当显示类似如下信息表示日志已经归档。

Successfully collected files

All results are stored in /opt/software/gaussdb/output/collector_20160616_190511.tar.gz.

 以--begin-time, --end-time与--keyword为参数执行gs_collector命令。 gs_collector --begin-time="20160616 01:01" --end-time="20160616 23:59" --keyword="os" 当显示类似如下信息表示日志已经归档。

Successfully collected files.

All results are stored in /tmp/gaussdba_mppdb/collector_20160616_190836.tar.gz.

以--begin-time, --end-time与-o为参数执行gs_collector命令。
 gs_collector --begin-time="20160616 01:01" --end-time="20160616 23:59" -o /opt/software/gaussdb/output

当显示类似如下信息表示日志已经归档。

Successfully collected files.

All results are stored in /opt/software/gaussdb/output/collector_20160726_113711.tar.gz.

 以--begin-time, --end-time与-l为参数(文件名必须以.log为后缀)执行 gs collector命令。

gs_collector --begin-time="20160616 01:01" --end-time="20160616 23:59" -l /opt/software/gaussdb/logfile.log

当显示类似如下信息表示日志已经归档。

Successfully collected files.

All results are stored in /opt/software/gaussdb/output/collector_20160726_113711.tar.gz.

清理运行日志

数据库运行过程中会产生大量运行日志,占用大量的磁盘空间,建议清理过期日志文件,只保留一个月的日志。

操作步骤

步骤1 以操作系统用户omm登录数据库主节点。

步骤2 清理日志。

- 1. 将超过1个月的日志备份到其他磁盘。
- 2. 进入日志存放目录。

cd \$GAUSSLOG

3. 进入相应的子目录,使用如下方式删除1个月之前产生的日志。 rm *日志名称*

日志文件的命名格式为"postgresgl-年-月-日_HHMMSS"。

----结束

5.6 检查时间一致性

数据库事务一致性通过逻辑时钟保证,与操作系统时间无关,但是系统时间不一致会导致诸多潜在问题,主要是后台运维和监控功能异常,因此在月度检查时建议检查各个节点的时间一致性。

操作步骤

步骤1 以操作系统用户omm登录数据库主节点。

步骤2 创建记录openGauss各节点的配置文件(*mpphosts文件目录*用户可随意指定,建议放在/tmp下)。

vim /tmp/mpphosts

增加各节点的主机名称。

plat1 plat2 plat3

步骤3 保存配置文件。

:wq!

步骤4 执行如下命令,输出各节点上的时间到"/tmp/sys_ctl-os1.log"文件中。

for ihost in `cat /tmp/mpphosts`; do ssh -n -q \$ihost "hostname;date"; done > /tmp/sys_ctl-os1.log

步骤5 根据输出确认各个节点的时间一致性,节点之间时间差异不能超过30秒。

cat /tmp/sys_ctl-os1.log plat1 Thu Feb 9 16:46:38 CST 2017 plat2 Thu Feb 9 16:46:49 CST 2017 plat3 Thu Feb 9 16:46:14 CST 2017

----结束

5.7 检查应用连接数

如果应用程序与数据库的连接数超过最大值,则新的连接无法建立。建议每天检查连接数,及时释放空闲的连接或者增加最大连接数。

操作步骤

步骤1 以操作系统用户omm登录数据库主节点。

步骤2 使用如下命令连接数据库。

gsql -d postgres -p 8000

postgres为需要连接的数据库名称,8000为数据库主节点的端口号。

连接成功后,系统显示类似如下信息:

gsql((openGauss x.x.x build f521c606) compiled at 2021-09-16 14:55:22 commit 2935 last mr 6385 release) Non-SSL connection (SSL connection is recommended when requiring high-security) Type "help" for help.

openGauss=#

步骤3 执行如下SQL语句查看连接数。

openGauss=# SELECT count(*) FROM (SELECT pg_stat_get_backend_idset() AS backendid) AS s;

显示类似如下的信息,其中2表示当前有两个应用连接到数据库。

```
count
------
2
(1 row)
```

步骤4 查看现有最大连接数。

openGauss=# SHOW max connections;

显示信息如下,其中200为现在的最大连接数。

```
max_connections
------
200
(1 row)
```

----结束

异常处理

如果显示的连接数接近数据库的最大连接数max_connections,则需要考虑清理现有连接数或者增加新的连接数。

步骤1 执行如下SQL语句,查看state字段等于idle,且state_change字段长时间没有更新过的连接信息。

openGauss=# SELECT * FROM pg_stat_activity where state='idle' order by state_change;

显示类似如下的信息:

```
datid | datname |
                    pid
                           | usesysid | usename | application_name | client_addr
| client_hostname | client_port |
                                   backend_start
                                                     | xact_start |
y_start
                  state_change
                                    | waiting | enqueue | state | resource_pool
           query
                  13626 | postgres | 140390162233104 |
                                                             | 2016-07-15 14
:09:04.496769+08 | 2016-07-15 14:09:04.496975+08 | f
                                                            | idle | default_pool
| select count(group_name) from pgxc_group;
| 13626 | postgres | 140390132872976 | 10 | gaussdba | cn_5002
                                                                     | 10.180.123.163
                48614 | 2016-07-15 14:11:16.014871+08 | | 2016-07-15 14
:21:17.346045+08 | 2016-07-15 14:21:17.346095+08 | f
                                                             | idle | default_pool
| SET SESSION AUTHORIZATION DEFAULT; RESET ALL;
```

步骤2 释放空闲的连接数。

查看每个连接,并与此连接的使用者确认是否可以断开连接,或执行如下SQL语句释放连接。其中,pid为上一步查询中空闲连接所对应的pid字段值。

openGauss=# SELECT pg_terminate_backend(140390132872976);

显示类似如下的信息:

```
openGauss=# SELECT pg_terminate_backend(140390132872976);
pg_terminate_backend
------
t
(1 row)
```

如果没有可释放的连接,请执行下一步。

步骤3 增加最大连接数。

gs_guc set -D /gaussdb/data/dbnode -c "max_connections= 800"

其中800为新修改的连接数。

步骤4 重启数据库服务使新的设置生效。

□□说明

重启openGauss操作会导致用户执行操作中断,请在操作之前规划好合适的执行窗口。

gs_om -t stop && gs_om -t start

----结束

5.8 例行维护表

为了保证数据库的有效运行,数据库必须在插入/删除操作后,基于客户场景,定期做 VACUUM FULL和ANALYZE,更新统计信息,以便获得更优的性能。

相关概念

使用VACUUM、VACUUM FULL和ANALYZE命令定期对每个表进行维护,主要有以下原因:

- VACUUM FULL可回收已更新或已删除的数据所占据的磁盘空间,同时将小数据 文件合并。
- VACUUM对每个表维护了一个可视化映射来跟踪包含对别的活动事务可见的数组的页。一个普通的索引扫描首先通过可视化映射来获取对应的数组,来检查是否对当前事务可见。若无法获取,再通过堆数组抓取的方式来检查。因此更新表的可视化映射,可加速唯一索引扫描。
- VACUUM可避免执行的事务数超过数据库阈值时,事务ID重叠造成的原有数据丢失。
- ANALYZE可收集与数据库中表内容相关的统计信息。统计结果存储在系统表 PG_STATISTIC中。查询优化器会使用这些统计数据,生成最有效的执行计划。

操作步骤

步骤1 使用VACUUM或VACUUM FULL命令,进行磁盘空间回收。

VACUUM:

对表执行VACUUM操作 openGauss=# VACUUM customer;

VACUUM

可以与数据库操作命令并行运行。(执行期间,可正常使用的语句:SELECT、INSERT、UPDATE和DELETE。不可正常使用的语句:ALTER TABLE)。

对表分区执行VACUUM操作

openGauss=# **VACUUM** *customer_par* **PARTITION** (*P1*); VACUUM

VACUUM FULL:

openGauss=# VACUUM FULL customer; VACUUM

需要向正在执行的表增加排他锁,且需要停止其他所有数据库操作。

步骤2 使用ANALYZE语句更新统计信息。

openGauss=# **ANALYZE** customer; ANALYZE

使用ANALYZE VERBOSE语句更新统计信息,并输出表的相关信息。

openGauss=# **ANALYZE VERBOSE** *customer*; ANALYZE

也可以同时执行VACUUM ANALYZE命令进行查询优化。

openGauss=# **VACUUM ANALYZE** *customer*; VACUUM

□ 说明

VACUUM和ANALYZE会导致I/O流量的大幅增加,这可能会影响其他活动会话的性能。因此,建议通过"vacuum_cost_delay"参数设置《开发者指南》中"GUC参数说明 > 资源消耗 > 基于开销的清理延迟"。

步骤3 删除表

openGauss=# DROP TABLE customer; openGauss=# DROP TABLE customer_par; openGauss=# DROP TABLE part;

当结果显示为如下信息,则表示删除成功。

DROP TABLE

----结束

维护建议

- 定期对部分大表做VACUUM FULL,在性能下降后为全库做VACUUM FULL,目前 暂定每月做一次VACUUM FULL。
- 定期对系统表做VACUUM FULL,主要是PG_ATTRIBUTE。
- 启用系统自动清理线程(AUTOVACUUM)自动执行VACUUM和ANALYZE,回收被标识为删除状态的记录空间,并更新表的统计数据。

5.9 例行重建索引

背景信息

数据库经过多次删除操作后,索引页面上的索引键将被删除,造成索引膨胀。例行重建索引,可有效的提高查询效率。

数据库支持的索引类型为B-tree索引,例行重建索引可有效的提高查询效率。

- 如果数据发生大量删除后,索引页面上的索引键将被删除,导致索引页面数量的 减少,造成索引膨胀。重建索引可回收浪费的空间。
- 新建的索引中逻辑结构相邻的页面,通常在物理结构中也是相邻的,所以一个新 建的索引比更新了多次的索引访问速度要快。

重建索引

重建索引有以下两种方式:

- 先运行DROP INDEX语句删除索引,再运行CREATE INDEX语句创建索引。 在删除索引过程中,会在父表上增加一个临时排他锁,阻止相关读写操作。在创 建索引过程中,会锁住写操作但是不会锁住读操作,此时读操作只能使用顺序扫 描。
- 使用REINDEX语句重建索引。
 - 使用REINDEX TABLE语句重建索引,会在重建过程中增加排他锁,阻止相关 读写操作。
 - 使用REINDEX INTERNAL TABLE语句重建desc表(包括列存表的cudesc表) 的索引,会在重建过程中增加排他锁,阻止相关读写操作。

操作步骤

假定在导入表 "areaS"上的 "area_id"字段上存在普通索引 "areaS_idx"。重建索 引有以下两种方式:

- 先运行DROP INDEX语句删除索引,再运行CREATE INDEX语句创建索引。
 - 删除索引。

openGauss=# DROP INDEX areaS idx; DROP INDEX

创建索引。

openGauss=# CREATE INDEX areaS idx ON areaS (area_id); CREATE INDEX

- 使用REINDEX重建索引。
 - 使用REINDEX TABLE语句重建索引。 openGauss=# REINDEX TABLE areaS;

REINDEX

使用REINDEX INTERNAL TABLE重建desc表(包括列存表的cudesc表)的索

openGauss=# REINDEX INTERNAL TABLE areaS; REINDEX

□□ 说明

在重建索引前,用户可以通过临时增大maintenance_work_mem和psort_work_mem的取值来 加快索引的重建。

5.10 数据安全维护建议

为保证openGauss数据库中的数据安全,避免丢失数据,非法访问数据等事故发生, 请仔细阅读以下内容。

避免数据被丢失

建议用户规划周期性的物理备份,且对备份文件进行可靠的保存。在系统发生严重错误的情况下,可以利用备份文件,将系统恢复到备份前的状态。

避免数据被非法访问

- 建议对数据库用户进行权限分级管理。数据库管理员根据业务需要,建立用户并 赋予权限,保证各用户对数据库的合理访问。
- 对于openGauss的服务端和客户端(或基于客户端库开发的应用程序),最好也部署在可信任的内网中。如果服务端和客户端一定要部署在非信任的网络中,需要在服务启动前,打开SSL加密,保证数据在非信任网络上的传输安全。需要注意的是,打开SSL加密会降低数据库的性能。

避免系统日志泄露个人数据

将调试日志发给他人进行分析前,请删除个人数据。

□ 说明

因为日志级别(log_min_messages) 设置为DEBUGx(x为DEBUG级别,取值范围为1~5) 时,调试日志中记录的信息可能包含用户的个人数据。

- 将系统日志发给其他人进行分析前,请删除个人数据。因为在默认配置下,当
 SQL语句执行错误时,日志中会记录出错的SQL语句,而这些SQL语句中可能包含用户个人数据。
- 将log_min_error_statement参数的值设置为PANIC,可以避免将出错的SQL语句记录在系统日志中。若禁用该功能,当出现故障时,很难定位故障原因。

5.11 慢 SQL 诊断

背景信息

在SQL语句执行性能不符合预期时,可以查看SQL语句执行信息,便于事后分析SQL语句执行时的行为,从而诊断SQL语句执行出现的相关问题。

前提条件

- 数据库实例运行正常。
- 查询SQL语句信息,需要正确设置GUC参数track stmt stat level。
- 只能用系统管理员和监控管理员权限进行操作。

执行命令查看数据库实例中SQL语句执行信息

gsql> select * from dbe_perf.get_global_full_sql_by_timestamp(start_timestamp, end_timestamp); 执行命令查看数据库实例中慢SQL语句执行信息

gsql> select * from dbe_perf.get_global_slow_sql_by_timestamp(start_timestamp, end_timestamp); 查看当前节点SQL语句执行信息

gsql> select * from statement_history;

6 备份与恢复

- 6.1 概述
- 6.2 物理备份恢复
- 6.3 逻辑备份恢复
- 6.4 闪回恢复

6.1 概述

数据备份是保护数据安全的重要手段之一,为了更好的保护数据安全,openGauss数据库支持三种备份恢复类型、多种备份恢复方案,备份和恢复过程中提供数据的可靠性保障机制。

备份与恢复类型可分为逻辑备份与恢复、物理备份与恢复、闪回恢复。

- 逻辑备份与恢复:通过逻辑导出对数据进行备份,逻辑备份只能基于备份时刻进行数据转储,所以恢复时也只能恢复到备份时保存的数据。对于故障点和备份点之间的数据,逻辑备份无能为力,逻辑备份适合备份那些很少变化的数据,当这些数据因误操作被损坏时,可以通过逻辑备份进行快速恢复。如果通过逻辑备份进行全库恢复,通常需要重建数据库,导入备份数据来完成,对于可用性要求很高的数据库,这种恢复时间太长,通常不被采用。由于逻辑备份具有平台无关性,所以更为常见的是,逻辑备份被作为一个数据迁移及移动的主要手段。
- 物理备份与恢复:通过物理文件拷贝的方式对数据库进行备份,以磁盘块为基本单位将数据从主机复制到备机。通过备份的数据文件及归档日志等文件,数据库可以进行完全恢复。物理备份速度快,一般被用作对数据进行备份和恢复,用于全量备份的场景。通过合理规划,可以低成本进行备份与恢复。
- 闪回恢复:利用回收站的闪回恢复删除的表。数据库的回收站功能类似于 windows系统的回收站,将删除的表信息保存到回收站中。利用MVCC机制闪回恢 复到指定时间点或者CSN点。

以下为openGauss支持的三类数据备份恢复方案,备份方案也决定了当异常发生时该如何恢复。

表 6-1 三种备份恢复类型对比

备份类 型	应用场景	支持的介质	工具 名称	恢复时间	优缺点
逻辑备份与复	适量景 可表单和dashase 于的	磁盘 SS D	gs_du mp	纯式复长。格恢中 生物,也是一个, 生物,也是一个, 生物,也是一个, 生物,也是一个, 生物,也是一个, 生物,也是一个, 生物,也是一个, 生物,也是一个, 是一个, 生物, 生物, 生物, 生物, 生物, 生物, 生物, 生物, 生物, 生物	导出数据库相关信息的工具,用户可以自定义导出。一个数据库相关信息的引力,则据库或其中的等)。 支持导出据库或现度可以,是以为据库。 以是是,以是自己的人。 这种,是是是一个数据库,是是一个数据库,是是一个数据库。 是是一个数据库,是是一个数据库,是是一个数据库。 是是一个数据库,是是一个数据,是一个数据,是一种数据,是一种数据,是一种数数据,是一种数数据,是一种数,是一种数数,是一种数数,是一种数数,是一种数数,是一种数,是一种数数,是一种数数,是一种数数数,是一种数数,是一种数数,是一种数数,是一种数数数,是一种数数数,是一种数数数,是一种数数数,是一种数数数,是一种数数数数数数数数数数
	长时间。		gs_du mpall	数据恢复时间长。	导出所有数据库相关信息 工具,它可以导出 openGauss数据库的所有 数据,包括默认数据库 postgres的数据、自定义 数据库的数据、以及 openGauss所有数据库公 共的全局对象。 只能导出纯文本格式的数 据,导出的数据只能通过 gsql进行恢复,恢复时间 较长。
物理备份复	适量景于备也数W志进分的要数场要数复整中档行备。 据,全份可据以和进行数据,有的日日		gs_ba ckup	数据量小 数据恢复 时间快。	导出数据库相关信息的 OM工具,可以导出数据 库参数文件和二进制文 件。帮助openGausss备 份、恢复重要数据、显。 帮助信息和版本号信息。 在进行备份的类型,在进行 备份内容的类型,在进行 还原目录中存在备份 还原目录中存在备份文 件。在集群恢复时,通过 静态配置文件中的集群 息进行恢复的实效。

备份类 型	应用场景	支持的介质	工具 名称	恢复时间	优缺点
			gs_ba sebac kup	恢以贝有件直份启库时复直替的,接的动,接的动,间的,接的动,间下的,所以的,则则有,则则是是一个,是是一个,是是一个,是是一个,是是一个,是是一个,是是一个,是是	对服务器数据库文件的二 进制进行全量拷贝,只能 对数据库某一个时间点的 时间作备份。结合PITR恢 复,可恢复全量备份时间 点后的某一时间点。
			gs_pr oback up	恢以复备在库数恢快复直到份备上据复。可恢个,的动,间	gs_probackup是一个用于管理openGauss数是一个用音份据对的更加恢复的工具。它行识别是一个库理的更加恢复的工具。它们是一个原生的工具。它们是一个原生的工具。它们是一个原生的工具。这个人们是一个人们,这个人们是一个人们,这个人们的人们的人们,这个人们的人们的人们,这个人们的人们的人们,这个人们的人们的人们,这个人们的人们的人们,这个人们的人们的人们的人们的人们的人们的人们的人们的人们的人们的人们的人们的人们的人

备份类 型	应用场景	支持的介质	工具 名称	恢复时间	优缺点
闪复 恢	适用于: 1) 以别称		无	可的复时者构的恢快以状到间是删状复。将态指点表除态时表恢定或结前,间	闪高的复前PIT据数闪的,时。 回 基据除据通间或复区 基回误TT复站恢及的技术的,采能手修钟技据需和 持 MVCC适新和旧行令的 实际的过机则数影。,RT联大体,在是一个人的一个人们过找,至后修砂据,从用通路的人的人们,大 复 版于误复本应查间 www.正,表配行以见好的人们,这是不是一个人的人,这级库 恢 多用、恢版相,时以及了,令、正通并,以及以上,这是一个人的人,这级库 恢 多用、恢版相,时以及了,令、正通并,以及以上,一个人,以为回路,以为回路,以为回路,以为回路,以为回路,以为回路,以为回路,以为回路

当需要进行备份恢复操作时,主要从以下四个方面考虑数据备份方案。

- 备份对业务的影响在可接受范围。
- 数据库恢复效率。
 - 为尽量减小数据库故障的影响,要使恢复时间减到最少,从而使恢复的效率 达到最高。
- 数据可恢复程度。当数据库失效后,要尽量减少数据损失。
- 数据库恢复成本。

在现网选择备份策略时参考的因素比较多,如备份对象、数据大小、网络配置等,表6-2列出了可用的备份策略和每个备份策略的适用场景。

表 6-2 备份策略典型场景

备份策 略	关键性能因素	典型数据 量	性能规格
数据库 实例备	数据大小网络配置	数据: PB 级 对象: 约 100万个	备份: 每个主机80 Mbit/s(NBU/EISOO +磁盘) 约90%磁盘I/O速率(SSD/HDD)
表备份	表所在模式网络配置 (NBU)	数据: 10 TB级	备份:基于查询性能速度+I/O速度 说明 多表备份时,备份耗时计算方式: 总时间 = 表数量 x 起步时间 + 数据总量 / 数据备份速度 其中: 磁盘起步时间为5s左右,NBU起步时间比DISK长(取决于NBU部署方案)。 数据备份速度为单节点50MB/s左右(基于1GB大小的表,物理机备份到本地磁盘得出此速率)。 表越小,备份性能更低。

6.2 物理备份恢复

6.2.1 gs_basebackup

背景信息

openGauss部署成功后,在数据库运行的过程中,会遇到各种问题及异常状态。openGauss提供了gs_basebackup工具做基础的物理备份。gs_basebackup的实现目标是对服务器数据库文件的二进制进行拷贝,其实现原理使用了复制协议。远程执行gs_basebackup时,需要使用系统管理员账户。gs_basebackup当前支持热备份模式和压缩格式备份。

□ 说明

- gs_basebackup仅支持主机和备机的全量备份,不支持增量。
- gs_basebackup当前支持热备份模式和压缩格式备份。
- gs_basebackup在备份包含绝对路径的表空间时,如果在同一台机器上进行备份,可以通过 tablespace-mapping重定向表空间路径,或使用归档模式进行备份。
- 若打开增量检测点功能且打开双写,gs_basebackup也会备份双写文件。
- 若pg_xlog目录为软链接,备份时将不会建立软链接,会直接将数据备份到目的路径的 pg_xlog目录下。
- 备份过程中收回用户备份权限,可能导致备份失败或者备份数据不可用。
- 如果因为网络临时故障等原因导致Server端无应答,gs_basebackup将在最长等待120秒后退出。

前提条件

- 可以正常连接openGauss数据库。
- 备份过程中用户权限没有被回收。
- pg_hba.conf中需要配置允许复制链接,且该连接必须由一个系统管理员建立。
- 如果xlog传输模式为stream模式,需要配置max_wal_senders的数量, 至少有一个可用。
- 如果xlog传输模式为fetch模式,有必要把wal_keep_segments参数设置得足够高,这样在备份末尾之前日志不会被移除。
- 在进行还原时,需要保证各节点备份目录中存在备份文件,若备份文件丢失,则 需要从其他节点进行拷贝。

语法

- 显示帮助信息 gs_basebackup -? | --help
- 显示版本号信息 gs_basebackup -V | --version

参数说明

gs_basebackup参数可以分为如下几类:

- -D directory备份文件输出的目录,必选项。
- 常用参数:
 - -c,--checkpoint=fast|spread 设置检查点模式为fast或者spread(默认)。
 - -l, --label=LABEL 为备份设置标签。
 - -P, --progress启用进展报告。
 - -v, --verbose 启用冗长模式。

– -V, --version 打印版本后退出。

- -?, --help

显示gs_basebackup命令行参数。

- -T, -tablespace-mapping=olddir=newdir

在备份期间将目录olddir中的表空间重定位到newdir中。为使之有效,olddir必须正好匹配表空间所在的路径(但如果备份中没有包含olddir中的表空间也不是错误)。olddir和newdir必须是绝对路径。如果一个路径凑巧包含了一个=符号,可用反斜线对它转义。对于多个表空间可以多次使用这个选项。

- -F, -format=plain|tar

设置输出格式为plain(默认)或者tar。没有设置该参数的情况下,默认-format=plain。plain格式把输出写成平面文件,使用和当前数据目录和表空间相同的布局。当集簇没有额外表空间时,整个数据库将被放在目标目录中。如果集簇包含额外的表空间,主数据目录将被放置在目标目录中,但是所有其他表空间将被放在它们位于服务器上的相同的绝对路径中。tar模式将输出写成目标目录中的 tar 文件。主数据目录将被写入到一个名为base.tar的文件中,并且其他表空间将被以其 OID 命名。生成的tar包,需要用gs_tar命令解压。

- -X, -xlog-method=fetch|stream

设置xlog传输方式。没有设置该参数的情况下,默认-xlog-method=stream。在备份中包括所需的预写式日志文件(WAL文件)。这包括所有在备份期间产生的预写式日志。fetch方式在备份末尾收集预写式日志文件。因此,有必要把wal_keep_segments参数设置得足够高,这样在备份末尾之前日志不会被移除。如果在要传输日志时它已经被轮转,备份将失败并且是不可用的。stream方式在备份被创建时流传送预写式日志。这将开启一个到服务器的第二连接并且在运行备份时并行开始流传输预写式日志。因此,它将使用最多两个由max_wal_senders参数配置的连接。只要客户端能保持接收预写式日志,使用这种模式不需要在主控机上保存额外的预写式日志。。

- -x,
 - -xlog 使用这个选项等效于和方法fetch一起使用-X。
- Z –compress=level

启用对 tar 文件输出的 gzip 压缩,并且制定压缩级别(0 到 9,0 是不压缩,9 是最佳压缩)。只有使用 tar 格式时压缩才可用,并且会在所有tar文件名后面自动加上后缀.gz。

- -7

启用对 tar 文件输出的 gzip 压缩,使用默认的压缩级别。只有使用 tar 格式时压缩才可用,并且会在所有tar文件名后面自动加上后缀.gz。

- -t, -rw-timeout

设置备份期间checkpoint的时间限制,默认限制时间为120s。当数据库全量 checkpoint耗时较长时,可以适当增大rw-timeout限制时间。

● 连接参数

- -h, --host=HOSTNAME指定正在运行服务器的主机名或者Unix域套接字的路径。
- -p,--port=PORT 指定数据库服务器的端口号。

可以通过port参数修改默认端口号。

- -U,--username=USERNAME 指定连接数据库的用户。
- -s, --status-interval=INTERVAL发送到服务器的状态包的时间(以秒为单位)。
- -w,--no-password不出现输入密码提示。
- -W, --password当使用-U参数连接本地数据库或者连接远端数据库时,可通过指定该选项出现输入密码提示。

示例

gs_basebackup -D /home/test/trunk/install/data/backup -h 127.0.0.1 -p 21233 INFO: The starting position of the xlog copy of the full build is: 0/1B800000. The slot minimum LSN is: 0/1B800000.

从备份文件恢复数据

当数据库发生故障时需要从备份文件进行恢复。因为gs_basebackup是对数据库按二进制进行备份,因此恢复时可以直接拷贝替换原有的文件,或者直接在备份的库上启动数据库。

□ 说明

- 若当前数据库实例正在运行,直接从备份文件启动数据库可能会存在端口冲突,这时需要修 改配置文件的port参数,或者在启动数据库时指定一下端口。
- 若当前备份文件为主备数据库,可能需要修改一下主备之间的复制连接。即配置文件中的 postgre.conf中的replconninfo1 、 replconninfo2等 。
- 若配置文件postgresql.conf的参数data_directory打开且有配置,当使用备份目录启动数据库 时候,data_directory和备份目录不同会导致启动失败。可以修改data_directory的值为新的 数据目录,或者注释掉该参数。

若要在原库的地方恢复数据库,参考步骤如下:

- 1. 停止数据库服务器, 具体操作请参见《管理员指南》。
- 2. 将原数据库和所有表空间复制到另外一个位置,以备后面需要。
- 3. 清理原库中的所有或部分文件。
- 4. 使用数据库系统用户权限从备份中还原需要的数据库文件。
- 5. 若数据库中存在链接文件,需要修改使其链接到正确的文件。
- 6. 重启数据库服务器,并检查数据库内容,确保数据库已经恢复到所需的状态。

□ 说明

- 暂不支持备份文件增量恢复。
- 恢复后需要检查数据库中的链接文件是否链接到正确的文件。

6.2.2 PITR 恢复

背景信息

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。

□ 说明

- PITR仅支持恢复到物理备份数据之后的某一时间点。
- 仅主节点可以进行PITR恢复,备机需要进行全量build达成与主机数据同步。

前提条件

- 基于经过物理备份的全量数据文件。
- 基于已归档的WAL日志文件。

PITR 恢复流程

步骤1 将物理备份的文件替换目标数据库目录。

步骤2 删除数据库目录下pg_xlog/中的所有文件。

步骤3 将归档的WAL日志文件复制到pg_xlog文件中(此步骤可以省略,通过配置 recovery.conf恢复命令文件中的restore_command项替代)。

步骤4 在数据库目录下创建恢复命令文件recovery.conf,指定数据库恢复的程度。

步骤5 启动数据库。

步骤6 连接数据库,查看是否恢复到希望预期的状态。

步骤7 若已经恢复到预期状态,通过pq_xloq_replay_resume()指令使主节点对外提供服务。

----结束

recovery.conf 文件配置

归档恢复配置

restore_command = string

这个SHELL命令是获取WAL文件系列中已归档的WAL文件。字符串中的任何一个%f是用归档检索中的文件名替换,并且%p是用服务器上的复制目的地的路径名替换。任意一个%r是用包含最新可用重启点的文件名替换。

示例:

restore_command = 'cp /mnt/server/archivedir/%f %p'

archive_cleanup_command = string

这个选项参数声明一个shell命令。在每次重启时会执行这个shell命令。 archive_cleanup_command为清理备库不需要的归档WAL文件提供一个机制。 任何一个%r由包含最新可用重启点的文件名代替。这是最早的文件, 因此必须*保留*以允许恢复能够重新启动,因此所有早于%r的文件可以安全的移除。

示例:

archive_cleanup_command = 'pg_archivecleanup /mnt/server/archivedir %r'

需要注意的是,如果多个备服务器从相同的归档路径恢复时, 需要确保在任何一个备服务器在需要之前,不能删除WAL文件。

recovery_end_command = string

这个参数是可选的,用于声明一个只在恢复完成时执行的SHELL命令。 recovery_end_command是为以后的复制或恢复提供一个清理机制。

恢复目标设置

recovery_target_name = string

此参数声明命名还原到一个使用pg_create_restore_point()创建的还原点。

示例:

recovery_target_name = 'restore_point_1'

recovery_target_time = timestamp

此参数声明命名还原到一个指定时间戳。

示例:

recovery_target_time = '2020-01-01 12:00:00'

recovery_target_xid = string

这个参数声明还原到一个事务ID。

示例:

recovery_target_xid = '3000'

recovery_target_lsn = string

这个参数声明还原到日志的指定LSN点。

示例:

recovery_target_lsn = '0/0FFFFFF'

recovery_target_inclusive = boolean

声明是否在指定恢复目标(true)之后停止,或在这(false)之前停止。该声明仅支持恢复目标为recovery_target_time,recovery_target_xid和recovery_target_lsn的配置。

示例:

recovery_target_inclusive = true

山 说明

- recovery_target_name, recovery_target_time, recovery_target_xid, recovery_target_lsn 这四个配置项仅同时支持一项。
- 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。

6.2.3 gs_probackup

背景信息

gs_probackup是一个用于管理openGauss数据库备份和恢复的工具。它对openGauss实例进行定期备份,以便在数据库出现故障时能够恢复服务器。

- 可用于备份单机数据库,也可对主机或者主节点数据库备机进行备份,为物理备份。
- 可备份外部目录的内容,如脚本文件、配置文件、日志文件、dump文件等。
- 支持增量备份、定期备份和远程备份。
- 可设置备份的留存策略。

前提条件

- 可以正常连接openGauss数据库。
- 若要使用PTRACK增量备份,需在postgresql.conf中手动添加参数 "enable_cbm_tracking = on"。
- 为了防止xlog在传输结束前被清理,请适当调高postgresql.conf文件中wal_keep_segments的值。

限制说明

- 备份必须由运行数据库服务器的用户执行。
- 备份和恢复的数据库服务器的主版本号必须相同。
- 如果要通过ssh在远程模式下备份数据库,需要在本地和远程主机安装相同主版本的数据库,并通过ssh-copy-id remote_user@remote_host命令设置本地主机备份用户和远程主机数据库用户的无密码ssh连接。
- 远程模式下只能执行add-instance、backup、restore子命令。
- 使用restore子命令前,应先停止gaussdb进程。
- 当存在用户自定义表空间时,备份的时候要加上 --external-dirs 参数,否则,该表空间不会被备份。
- 当备份的规模比较大时,为了防止备份过程中timeout发生,请适当调整 postgresql.conf文件的参数 session_timeout、wal_sender_timeout。并且在备份 的命令行参数中适当调整参数--rw-timeout的值。
- 恢复时,使用-T选项把备份中的外部目录重定向到新目录时,请同时指定参数-external-mapping。
- 增量备份恢复后,之前创建的逻辑复制槽不可用,需删除重建。
- 当使用远程备份时,请确保远程机器和备份机器的时钟同步,以防止使用--recovery-target-time恢复的场合,启动gaussdb时有可能会失败。
- 当远程备份有效时(remote-proto=ssh),请确保-h和--remote-host指定的是同一台机器。当远程备份无效时,如果指定了-h选项,请确保-h指定的是本机地址或本机主机名。
- 当前暂不支持备份逻辑复制槽。

命令说明

• 打印gs_probackup版本。

gs_probackup -V|--version gs_probackup version

● 显示gs_probackup命令的摘要信息。如果指定了gs_probackup的子命令,则显示可用于此子命令的参数的详细信息。

gs_probackup -?|--help gs_probackup help [command]

初始化备份路径 backup-path中的备份目录,该目录将存储已备份的内容。如果备份路径 backup-path已存在,则 backup-path必须为空目录。
 gs_probackup init -B backup-path [--help]

● 在备份路径 backup-path内初始化一个新的备份实例,并生成pg_probackup.conf 配置文件,该文件保存了指定数据目录 pgdata-path的gs_probackup设置。

gs_probackup add-instance -B backup-path -D pgdata-path --instance=instance_name [-E external-directories-paths] [remote_options] [--help]

● 在备份路径 backup-path内删除指定实例相关的备份内容。

gs_probackup del-instance -B *backup-path* --instance=*instance_name* [--help]

 将指定的连接、压缩、日志等相关设置添加到pg_probackup.conf配置文件中,或 修改已设置的值。不推荐手动编辑pg_probackup.conf配置文件。

修改已设置的值。不推荐手动编辑pg_probackup.conf配置文件。 gs_probackup set-config -B *backup-path* --instance=*instance_name* [-D *pgdata-path*] [-E *external-directories-paths*] [--archive-timeout=*timeout*]

[--retention-redundancy=*retention-redundancy*] [--retention-window=*retention-window*] [--waldepth=*wal-depth*]

[--compress-algorithm=compress-algorithm] [--compress-level=compress-level] [-d dbname] [-h hostname] [-p port] [-U username]

[logging_options] [remote_options]

[--help]

● 将备份相关设置添加到backup.control配置文件中,或修改已设置的值。

gs_probackup set-backup -B *backup-path* --instance=*instance_name* -i *backup-id* [--note=*text*] [pinning_options] [--help]

● 显示位于备份目录中的pg_probackup.conf配置文件的内容。可以通过指定-format=json选项,以json格式显示。默认情况下,显示为纯文本格式。

gs_probackup show-config -B *backup-path* --instance=*instance_name* [--format=plain|json] [--help]

 显示备份目录的内容。如果指定了instance_name和backup_id,则显示该备份的 详细信息。可以通过指定--format=json选项,以json格式显示。默认情况下,备 份目录的内容显示为纯文本格式。

gs_probackup show -B *backup-path*[--instance=*instance_name* [-i *backup-id*]] [--archive] [--format=plain|json]
[--help]

创建指定实例的备份。

gs_probackup backup -B backup-path --instance=instance_name -b backup-mode
[-D pgdata-path] [-C] [-S slot-name] [--temp-slot] [--backup-pg-log] [-j threads_num] [--progress]
[--no-validate] [--skip-block-validation] [-E external-directories-paths] [--no-sync] [--note=text]
[--archive-timeout=timeout] [-t rwtimeout]
[logging_options] [retention_options] [compression_options]
[connection_options] [remote_options] [pinning_options]
[--help]

从备份目录 backup-path中的备份副本恢复指定实例。如果指定了恢复目标选项,gs_probackup将查找最近的备份并将其还原到指定的恢复目标。否则,使用最近一次备份。

gs_probackup restore -B backup-path --instance=instance_name
[-D pqdata-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_options] [remote_options] [logging_options]
[--help]

● 将指定的增量备份与其父完全备份之间的所有增量备份合并到父完全备份。父完 全备份将接收所有合并的数据,而已合并的增量备份将作为冗余被删除。

gs_probackup merge -B *backup-path* --instance=*instance_name* -i *backup_id* [-j *threads_num*] [--progress] [logging_options] [--help]

• 删除指定备份,或删除不满足当前保留策略的备份。

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]
[logging_options]
[--help]

 验证恢复数据库所需的所有文件是否存在且未损坏。如果未指定 instance_name, gs_probackup将验证备份目录中的所有可用备份。如果指定 instance_name而不指定任何附加选项,gs_probackup将验证此备份实例的所有 可用备份。如果指定了instance_name并且指定backup-id或恢复目标相关选项, gs_probackup将检查是否可以使用这些选项恢复数据库。

gs_probackup validate -B backup-path
[--instance=instance_name] [-i backup-id]
[-j threads_num] [--progress] [--skip-block-validation]
[--recovery-target-time=time | --recovery-target-xid=xid | --recovery-target-lsn=lsn | --recovery-target-name=target-name]
[--recovery-target-inclusive=boolean]
[logging_options]
[--help]

参数说明

通用参数

command

gs_probackup除version和help以外的子命令: init、add-instance、del-instance、set-config、set-backup、show-config、show、backup、restore、merge、delete、validate。

• -?, --help

显示gs_probackup命令行参数的帮助信息,然后退出。 子命令中只能使用--help,不能使用--?。

-V, --version

打印qs_probackup版本,然后退出。

B backup-path, --backup-path=backup-path
 备份的路径。

系统环境变量: \$BACKUP PATH

• -D *pgdata-path*, --pgdata=*pgdata-path*

数据目录的路径。

系统环境变量: \$PGDATA

--instance=*instance_name*实例名。

- -i backup-id, --backup-id=backup-id
 备份的唯一标识。
- --format=format

指定显示备份信息的格式,支持plain和json格式。

默认值: plain

• --status=*backup status*

删除指定状态的所有备份,包含以下状态:

- OK: 备份已完成且有效。
- DONE:备份已完成但未经过验证。
- RUNNING:备份正在进行中。
- MERGING: 备份正在合并中。
- DELETING: 备份正在删除中。
- CORRUPT: 部分备份文件已损坏。
- ERROR:由于意外错误,备份失败。
- ORPHAN:由于其父备份之一已损坏或丢失,备份无效。
- -j threads_num, --threads=threads_num
 设置备份、还原、合并进程的并行线程数。
- --archive

显示WAL归档信息。

- --progress显示讲度。
- --note=*text*给备份添加note。

备份相关参数

• -b *backup-mode*, --backup-mode=*backup-mode*

指定备份模式,支持FULL和PTRACK。

FULL: 创建全量备份,全量备份包含所有数据文件。

PTRACK: 创建PTRACK增量备份。

-C, --smooth-checkpoint

将检查点在一段时间内展开。默认情况下,gs_probackup会尝试尽快完成检查点。

- -S slot-name, --slot=slot-name 指定WAL流处理的复制slot。
- --temp-slot

在备份的实例中为WAL流处理创建一个临时物理复制slot,它确保在备份过程中, 所有所需的WAL段仍然是可用的。

默认的slot名为pg_probackup_slot,可通过选项--slot/-S更改。

--backup-pg-log

将日志目录包含到备份中。此目录通常包含日志消息。默认情况下包含日志目录,但不包含日志文件。如果修改了默认的日志路径,需要备份日志文件时可使用-E参数进行备份,使用方法见下文。

• -E external-directories-paths, --external-dirs=external-directories-paths 将指定的目录包含到备份中。此选项对于备份位于数据目录外部的脚本、sql转储和配置文件很有用。如果要备份多个外部目录,请在Unix上用冒号分隔它们的路径。

例如: -E /tmp/dir1:/tmp/dir2

--skip-block-validation关闭块级校验,加快备份速度。

--no-validate在完成备份后跳过自动验证。

--no-sync

不将备份文件同步到磁盘。

--archive-timeout=timeout以秒为单位设置流式处理的超时时间。

默认值: 300

• -t rwtimeout

以秒为单位的连接的超时时间。

默认值: 120

恢复相关参数

-l, --incremental-mode=none|checksum|lsn若PGDATA中可用的有效页没有修改,则重新使用它们。

默认值: none

--external-mapping=OLDDIR=NEWDIR

在恢复时,将包含在备份中的外部目录从*OLDDIR*重新定位到*NEWDIR*目录。 *OLDDIR*和*NEWDIR*都必须是绝对路径。如果路径中包含"=",则使用反斜杠转 义。此选项可为多个目录多次指定。

- -T OLDDIR=NEWDIR, --tablespace-mapping=OLDDIR=NEWDIR
 在恢复时,将表空间从OLDDIR重新定位到NEWDIR目录。OLDDIR和NEWDIR必须都是绝对路径。如果路径中包含"=",则使用反斜杠转义。多个表空间可以多次指定此选项。此选项必须和--external-mapping一起使用。
- --skip-external-dirs

跳过备份中包含的使用--external-dirs选项指定的外部目录。这些目录的内容将不会被恢复。

--skip-block-validation

跳过块级校验,以加快验证速度。在恢复之前的自动验证期间,将仅做文件级别的校验。

--no-validate跳过备份验证。

--force

允许忽略备份的无效状态。如果出于某种原因需要从损坏的或无效的备份中恢复 数据,可以使用此标志。请谨慎使用。

恢复目标相关参数(recovery_options)

□ 说明

当前不支持配置连续的WAL归档的PITR,因而使用这些参数会有一定限制,具体如下描述。如果需要使用持续归档的WAL日志进行PITR恢复,请按照下面描述的步骤:

- 1. 将物理备份的文件替换目标数据库目录。
- 2. 删除数据库目录下pg_xlog/中的所有文件。
- 3. 将归档的WAL日志文件复制到pg_xlog文件中(此步骤可以省略,通过配置recovery.conf恢复命令文件中的restore_command项替代)。
- 4. 在数据库目录下创建恢复命令文件recovery.conf,指定数据库恢复的程度。
- 5. 启动数据库。
- 6. 连接数据库,查看是否恢复到希望预期的状态。若已经恢复到预期状态,通过 pg_xlog_replay_resume()指令使主节点对外提供服务。
- --recovery-target-lsn=*lsn*

指定要恢复到的lsn, 当前只能指定备份的stop lsn。

--recovery-target-name=target-name
 指定要将数据恢复到的已命名的保存点,保存点可以通过查看备份中recovery-name字段得到。

- --recovery-target-time=time指定要恢复到的时间,当前只能指定备份中的recovery-time。
- --recovery-target-xid=xid指定要恢复到的事务ID,当前只能指定备份中的recovery-xid。
- --recovery-target-inclusive=boolean

当该参数指定为true时,恢复目标将包括指定的内容。

当该参数指定为false时,恢复目标将不包括指定的内容。

该参数必须和--recovery-target-name、--recovery-target-time、--recovery-target-lsn或--recovery-target-xid一起使用。

留存相关参数(retention_options)

□ 说明

可以和backup和delete命令一起使用这些参数。

--retention-redundancy=retention-redundancy
 指定在数据目录中留存的完整备份数。必须为正整数。0表示禁用此设置。
 默认值: 0

• --retention-window=retention-window

指定留存的天数。必须为正整数。0表示禁用此设置。

默认值: 0

--wal-depth=wal-depth

每个时间轴上必须留存的执行PITR能力的最新有效备份数。必须为正整数。0表示禁用此设置。

默认值: 0

• --delete-wal

从任何现有的备份中删除不需要的WAL文件。

--delete-expired

删除不符合pg_probackup.conf配置文件中定义的留存策略的备份。

--merge-expired

将满足留存策略要求的最旧的增量备份与其已过期的父备份合并。

--dry-run

显示所有可用备份的当前状态,不删除或合并过期备份。

固定备份相关参数(pinning_options)

□ 说明

如果要将某些备份从已建立的留存策略中排除,可以和backup和set-backup命令一起使用这些 参数 。

• --ttl=*interval*

指定从恢复时间开始计算,备份要固定的时间量。必须为正整数。0表示取消备份 固定。

支持的单位: ms, s, min, h, d(默认为s)。

例如: --ttl=30d。

• --expire-time=*time*

指定备份固定失效的时间戳。必须是ISO-8601标准的时间戳。

例如: --expire-time='2020-01-01 00:00:00+03'

日志相关参数(logging_options)

日志级别: verbose、log、info、warning、error和off。

• --log-level-console=*log-level-console*

设置要发送到控制台的日志级别。每个级别都包含其后的所有级别。级别越高,发送的消息越少。指定off级别表示禁用控制台日志记录。

默认值: info

• --log-level-file=*log-level-file*

设置要发送到日志文件的日志级别。每个级别都包含其后的所有级别。级别越高,发送的消息越少。指定off级别表示禁用日志文件记录。

默认值: off

• --log-filename=*log-filename*

指定要创建的日志文件的文件名。文件名可以使用strftime模式,因此可以使用%-escapes指定随时间变化的文件名。

例如,如果指定了"pg_probackup-%u.log"模式,则pg_probackup为每周的每一天生成单独的日志文件,其中%u替换为相应的十进制数字,即pg_probackup-1.log表示星期一;pg_probackup-2.log表示星期二,以此类推。如果指定了--log-level-file参数启用日志文件记录,则该参数有效。

默认值: "pg_probackup.log"

- --error-log-filename=error-log-filename
 指定仅用于error日志的日志文件名。指定方式与--log-filename参数相同。
 此参数用于故障排除和监视。
- --log-directory=log-directory
 指定创建日志文件的目录。必须是绝对路径。此目录会在写入第一条日志时创建。

默认值: \$BACKUP_PATH/log

• --log-rotation-size=*log-rotation-size*

指定单个日志文件的最大大小。如果达到此值,则启动gs_probackup命令后,日志文件将循环,但help和version命令除外。0表示禁用基于文件大小的循环。

支持的单位: KB、MB、GB、TB(默认为KB)。

默认值: 0

• --log-rotation-age=*log-rotation-age*

单个日志文件的最大生命周期。如果达到此值,则启动gs_probackup命令后,日志文件将循环,但help和version命令除外。\$BACKUP_PATH/log/log_rotation目录下保存最后一次创建日志文件的时间。0表示禁用基于时间的循环。

支持的单位: ms, s, min, h, d (默认为min) 。

默认值: 0

连接相关参数(connection options)

山 说明

可以和backup命令一起使用这些参数。

• -d *dbname*, --pgdatabase=*dbname*

指定要连接的数据库名称。该连接仅用于管理备份进程,因此您可以连接到任何现有的数据库。如果命令行、PGDATABASE环境变量或pg_probackup.conf配置文件中没有指定此参数,则gs_probackup会尝试从PGUSER环境变量中获取该值。如果未设置PGUSER变量,则从当前用户名获取。

系统环境变量: \$PGDATABASE

• -h *hostname*, --pghost=*hostname*

指定运行服务器的系统的主机名。如果该值以斜杠开头,则被用作到Unix域套接字的路径。

系统环境变量: \$PGHOST

默认值: local socket

• -p *port*, --pgport=*port*

指定服务器正在侦听连接的TCP端口或本地Unix域套接字文件扩展名。

系统环境变量: \$PGPORT

默认值: 5432

• -U *username*, --pguser=*username*

指定所连接主机的用户名。

系统环境变量: \$PGUSER

-w, --no-password

不出现输入密码提示。如果主机要求密码认证并且密码没有通过其它形式给出, 则连接尝试将会失败。 该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

• -W password, --password=password

指定用户连接的密码。如果主机的认证策略是trust,则不会对系统管理员进行密码验证,即无需输入-W选项;如果没有-W选项,并且不是系统管理员,则会提示用户输入密码。

压缩相关参数(compression_options)

□ 说明

可以和backup命令一起使用这些参数。

• --compress-algorithm=compress-algorithm

指定用于压缩数据文件的算法。

取值包括zlib、pglz和none。如果设置为zlib或pglz,此选项将启用压缩。默认情况下,压缩功能处于关闭状态。

默认值: none

• --compress-level=*compress-level*

指定压缩级别。取值范围: 0~9

- 0表示无压缩。
- 1表示压缩比最小,处理速度最快。
- 9表示压缩比最大,处理速度最慢。
- 可与--compress-algorithm选项一起使用。

默认值: 1

--compress

以--compress-algorithm=zlib和--compress-level=1进行压缩。

远程模式相关参数(remote_options)

□ 说明

通过SSH远程运行gs_probackup操作的相关参数。可以和add-instance、set-config、backup、restore命令一起使用这些参数。

• --remote-proto=*protocol*

指定用于远程操作的协议。目前只支持SSH协议。取值包括:

ssh: 通过SSH启用远程备份模式。这是默认值。

none: 显式禁用远程模式。

如果指定了--remote-host参数,可以省略此参数。

--remote-host=destination

指定要连接的远程主机的IP地址或主机名。

--remote-port=port

指定要连接的远程主机的端口号。

默认值: 22

--remote-user=username

指定SSH连接的远程主机用户。如果省略此参数,则使用当前发起SSH连接的用户。

默认值: 当前用户

• --remote-path=*path*

指定gs_probackup在远程系统的安装目录。

默认值: 当前路径

• --remote-libpath=libpath

指定gs_probackup在远程系统安装的lib库目录。

--ssh-options=ssh_options 指定SSH命令行参数的字符串。

例如: --ssh-options='-c cipher_spec -F configfile'

□ 说明

- 如果因为网络临时故障等原因导致server端无应答,gs_probackup将在等待archive-timeout(默认300秒)后退出。
- 如果备机lsn与主机有差别时,数据库会不停地刷以下loq信息,此时应重新build备机。

LOG: walsender thread shut down

LOG: walsender thread started

LOG: received wal replication command: IDENTIFY_VERSION LOG: received wal replication command: IDENTIFY_MODE LOG: received wal replication command: IDENTIFY_SYSTEM

LOG: received wal replication command: IDENTIFY_CONSISTENCE 0/D0002D8

LOG: remote request lsn/crc: [xxxxx] local max lsn/crc: [xxxxx]

备份流程

步骤1 初始化备份目录。在指定的目录下创建backups/和wal/子目录,分别用于存放备份文件和WAL文件。

gs_probackup init -B backup_dir

步骤2 添加一个新的备份实例。gs_probackup可以在同一个备份目录下存放多个数据库实例的备份。

gs_probackup add-instance -B backup_dir -D data_dir --instance instance_name

步骤3 创建指定实例的备份。在进行增量备份之前,必须至少创建一次全量备份。gs_probackup backup_backup_dir --instance_instance_name -b backup_mode

步骤4 从指定实例的备份中恢复数据。

gs_probackup restore -B backup_dir --instance instance_name -D pgdata-path -i backup_id

----结束

故障处理

问题描述	原因和解决方案
ERROR: query failed: ERROR: canceling statement due to conflict with recovery (错误:查询失败:由于与恢复操作冲突,正在取消语句命令)	原因:在备机上执行的操作正在访问存储行,主机上更改或者删除了对应的行,并将xlog在备机上重放,迫使备机上操作取消。解决方案: 1. 适当增加如下配置参数的值max_standby_archive_delaymax_standby_streaming_delay 2. 增加如下配置hot_standby_feedback = on

6.3 逻辑备份恢复

6.3.1 gs_dump

背景信息

gs_dump是openGauss用于导出数据库相关信息的工具,用户可以自定义导出一个数据库或其中的对象(模式、表、视图等),回收站对象除外。支持导出的数据库可以是默认数据库postgres,也可以是自定义数据库。

gs_dump工具由操作系统用户omm执行。

gs_dump工具在进行数据导出时,其他用户可以访问openGauss数据库(读或写)。

gs_dump工具支持导出完整一致的数据。例如,T1时刻启动gs_dump导出A数据库,那么导出数据结果将会是T1时刻A数据库的数据状态,T1时刻之后对A数据库的修改不会被导出。

qs_dump时生成列不会被转储。

qs_dump支持导出兼容v1版本数据库的文本格式文件。

gs_dump支持将数据库信息导出至纯文本格式的SQL脚本文件或其他归档文件中。

- 纯文本格式的SQL脚本文件:包含将数据库恢复为其保存时的状态所需的SQL语句。通过gsql运行该SQL脚本文件,可以恢复数据库。即使在其他主机和其他数据库产品上,只要对SQL脚本文件稍作修改,也可以用来重建数据库。
- 归档格式文件:包含将数据库恢复为其保存时的状态所需的数据,可以是tar格式、目录归档格式或自定义归档格式,详见表6-3。该导出结果必须与gs_restore配合使用来恢复数据库,gs_restore工具在导入时,系统允许用户选择需要导入的内容,甚至可以在导入之前对等待导入的内容进行排序。

主要功能

gs_dump可以创建四种不同的导出文件格式,通过[-F或者--format=]选项指定,具体如表6-3所示。

表 6-3 导出文件格式

格式名称	-F的 参数 值	说明	建议	对应导入工具
纯文本 格式	р	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	С	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。

格式名称	-F的 参数 值	说明	建议	对应导入工具
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。	-	

山 说明

可以使用gs_dump程序将文件压缩为目录归档或自定义归档导出文件,减少导出文件的大小。 生成目录归档或自定义归档导出文件时,默认进行中等级别的压缩。gs_dump程序无法压缩已 归档导出文件。

注意事项

- 禁止修改导出的文件和内容,否则可能无法恢复成功。
- 为了保证数据一致性和完整性,gs_dump会对需要转储的表设置共享锁。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。如果无法在指定时间内锁定某个表,转储会失败。用户可以通过指定--lock-wait-timeout选项,自定义等待锁超时时间。
- 不支持加密导出存储过程和函数。

语法

gs_dump [OPTION]... [DBNAME]

□ 说明

"DBNAME"前面不需要加短或长选项。"DBNAME"指定要连接的数据库。

例如:

不需要-d,直接指定"DBNAME"。

gs_dump -p port_number postgres -f dump1.sql

或者

export PGDATABASE=postgres

gs_dump -p *port_number* -f dump1.sql

环境变量: PGDATABASE

参数说明

通用参数:

• -f, --file=FILENAME

将输出发送至指定文件或目录。如果省略该参数,则使用标准输出。如果输出格式为(-F c/-F d/-F t)时,必须指定-f参数。如果-f的参数值含有目录,要求当前用户对该目录具有读写权限,并且不能指定已有目录。

-F, --format=c|d|t|p

选择输出格式。格式如下:

- p|plain:输出一个文本SQL脚本文件(默认)。
- c|custom:输出一个自定义格式的归档,并且以目录形式输出,作为 gs_restore输入信息。该格式是最灵活的输出格式,因为能手动选择,而且能 在恢复过程中将归档项重新排序。该格式默认状态下会被压缩。
- d|directory: 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。
- t|tar:输出一个tar格式的归档形式,作为gs_restore输入信息。tar格式与目录格式兼容;tar格式归档形式在提取过程中会生成一个有效的目录格式归档形式。但是,tar格式不支持压缩且对于单独表有8GB的大小限制。此外,表数据项的相应排序在恢复过程中不能更改。

输出一个tar格式的归档形式,也可以作为gsql输入信息。

-v, --verbose

指定verbose模式。该选项将导致gs_dump向转储文件输出详细的对象注解和启动/停止次数,向标准错误流输出处理信息。

-V, --version

打印gs dump版本, 然后退出。

• -Z, --compress=0-9

指定使用的压缩比级别。

取值范围: 0~9

- 0表示无压缩。
- 1表示压缩比最小,处理速度最快。
- 9表示压缩比最大,处理速度最慢。

针对自定义归档格式,该选项指定单个表数据片段的压缩,默认方式是以中等级 别进行压缩。tar归档格式和纯文本格式目前不支持压缩。

--lock-wait-timeout=TIMEOUT

请勿在转储刚开始时一直等待以获取共享表锁。如果无法在指定时间内锁定某个表,就选择失败。可以以任何符合SET statement_timeout的格式指定超时时间。

• -?, --help

显示qs_dump命令行参数帮助,然后退出。

转储参数:

-a, --data-only

只输出数据,不输出模式(数据定义)。转储表数据、大对象和序列值。

• -b, --blobs

该参数为扩展预留接口,不建议使用。

• -c, --clean

在将创建数据库对象的指令输出到备份文件之前,先将清理(删除)数据库对象的指令输出到备份文件中。(如果目标数据库中没有任何对象,gs_restore工具可能会输出一些提示性的错误信息)

该选项只对文本格式有意义。针对归档格式,可以调用gs_restore时指定选项。

-C, --create

备份文件以创建数据库和连接到创建的数据库的命令开始。(如果命令脚本是这种方式执行,可以先指定任意数据库用于执行创建数据库的命令,数据不会恢复到指定的数据库中,而是恢复到创建的数据库中。)

该选项只对文本格式有意义。针对归档格式,可以在调用gs_restore时指定选项。

-E, --encoding=ENCODING

以指定的字符集编码创建转储。默认情况下,以数据库编码创建转储。(得到相同结果的另一个办法是将环境变量"PGCLIENTENCODING"设置为所需的转储编码。)

-n, --schema=SCHEMA

只转储与模式名称匹配的模式,此选项包括模式本身和所有它包含的对象。如果该选项没有指定,所有在目标数据库中的非系统模式将会被转储。写入多个-n选项来选择多个模式。此外,根据gsql的\d命令所使用的相同规则,模式参数可被理解成一个pattern,所以多个模式也可以通过在该pattern中写入通配符来选择。使用通配符时,注意给pattern打引号,防止shell扩展通配符。

□ 说明

- 当-n已指定时,gs_dump不会转储已选模式所附着的任何其他数据库对象。因此,无法保证某个指定模式的转储结果能够自行成功地储存到一个空数据库中。
- 当-n指定时,非模式对象不会被转储。

转储支持多个模式的转储。多次输入-n schemaname转储多个模式。

例如:

gs_dump -h *host_name* -p *port_number* postgres -f *backup/bkp_shl2.sql* -n *sch1* -n *sch2*

在上面这个例子中, sch1和sch2会被转储。

-N, --exclude-schema=SCHEMA

不转储任何与模式pattern匹配的模式。pattern将参照针对-n的相同规则来理解。可以通过输入多次-N,不转储与任何pattern匹配的模式。

当同时输入-n和-N时,会转储与至少一个-n选项匹配、与-N选项不匹配的模式。 如果有-N没有-n,则不转储常规转储中与-N匹配的模式。

转储过程支持排除多个模式。

在转储过程中,输入-N exclude schema name排除多个模式。

例如:

gs dump -h host name -p port number postgres -f backup/bkp shl2.sql -N sch1 -N sch2

在上面这个例子中,sch1和sch2在转储过程中会被排除。

• -o. --oids

转储每个表的对象标识符(OIDs),作为表的一部分数据。该选项用于应用以某种方式参照了OID列的情况。如果不是以上这种情况,请勿使用该选项。

-O, --no-owner

不输出设置对象的归属这样的命令,以匹配原始数据库。默认情况下,gs_dump会发出ALTER OWNER或SET SESSION AUTHORIZATION语句设置所创建的数据库对象的归属。如果脚本正在运行,该语句不会执行成功,除非是由系统管理员触发(或是拥有脚本中所有对象的同一个用户)。通过指定-O,编写一个任何用户都能存储的脚本,且该脚本会授予该用户拥有所有对象的权限。

该选项只对文本格式有意义。针对归档格式,可以在调用gs_restore时指定选项。

- -s, --schema-only只转储对象定义(模式), 而非数据。
- -S, --sysadmin=NAME 该参数为扩展预留接口,不建议使用。
- -t, --table=TABLE

指定转储的表(或视图、或序列、或外表)对象列表,可以使用多个-t选项来选择 多个表,也可以使用通配符指定多个表对象。

当使用通配符指定多个表对象时,注意给pattern打引号,防止shell扩展通配符。 当使用-t时,-n和-N没有任何效应,这是因为由-t选择的表的转储不受那些选项的 影响。

□ 说明

- -t参数选项个数必须小于等于100。
- 如果-t参数选项个数大于100,建议使用参数--include-table-file来替换。
- 当-t已指定时,gs_dump不会转储已选表所附着的任何其他数据库对象。因此,无法保证某个指定表的转储结果能够自行成功地储存到一个空数据库中。
- -t tablename只转储在默认搜索路径中可见的表。-t '*.tablename'转储数据库下所有模式下的tablename表。-t schema.table转储特定模式中的表。
- -t tablename不会导出表上的触发器信息。

例如:

gs_dump -h host_name -p port_number postgres -f backup/bkp_shl2.sql -t schema1.table1 -t schema2.table2

在上面这个例子中,schema1.table1和schema2.table2会被转储。

- --include-table-file=FILENAME
 指定需要dump的表文件。
 - -T, --exclude-table=TABLE

不转储的表(或视图、或序列、或外表)对象列表,可以使用多个-T选项来选择 多个表,也可以使用通配符指定多个表对象。

当同时输入-t和-T时,会转储在-t列表中,而不在-T列表中的表对象。

例如:

gs_dump -h *host_name* -p *port_number* postgres -f *backup/bkp_shl2.sql* -T *table1* -T *table2* 在上面这个例子中,table1和table2在转储过程中会被排除。

 --exclude-table-file=FILENAME 指定不需要dump的表文件。

□ 说明

同--include-table-file,其内容格式如下: schema1.table1 schema2.table2

- -x, --no-privileges|--no-acl防止转储访问权限(授权/撤销命令)。
- -q, --target

指定导出兼容其他版本数据库的文本文件,目前支持v1和v5参数。v1参数用于导出v5数据库的数据为兼容v1的文本文件。v5参数用于导出v5数据库的数据为v5格式的文本文件,减少了导入v5时的可能的报错情况。

在使用v1参数时,建议和--exclude-guc="enable_cluster_resize",--exclude-function,--exclude-with等选项共用,否则导入到v1时可能报错。

--exclude-guc

导出的文本文件中,不包括相关guc参数的set命令,目前只支持enable cluster resize。

--exclude-function

不导出函数和存储过程。

--exclude-with

导出的表定义,末尾不添加WITH(orientation=row,compression=on) 这样的描述。

--binary-upgrade

该参数为扩展预留接口,不建议使用。

--binary-upgrade-usermap="USER1=USER2" 该参数为扩展预留接口,不建议使用。

• --column-inserts|--attribute-inserts

以INSERT命令带列名(INSERT INTO表(列、···)值···)方式导出数据。这会导致恢复缓慢。但是由于该选项会针对每行生成一个独立分开的命令,所以在重新加载某行时出现的错误只会导致那行丢失,而非整个表内容。

--disable-dollar-quoting

该选项将禁止在函数体前使用美元符号\$,并强制使用SQL标准字符串语法对其进行引用。

--disable-triggers

该参数为扩展预留接口,不建议使用。

--exclude-table-data=TABLE

指定不转储任何匹配表pattern的表这方面的数据。依照针对-t的相同规则理解该pattern。

可多次输入--exclude-table-data来排除匹配任何pattern的表。当用户需要特定表的定义但不需要其中的数据时,这个选项很有帮助。

排除数据库中所有表的数据,参见--schema-only。

--inserts

发出INSERT命令(而非COPY命令)转储数据。这会导致恢复缓慢。

但是由于该选项会针对每行生成一个独立分开的命令,所以在重新加载某行时出现的错误只会导致那行丢失,而非整个表内容。注意:如果重排列顺序,可能会导致整个恢复失败。列顺序改变时,--column-inserts选项不受影响,虽然会更慢。

--no-publications

不转储发布。

--no-security-labels该参数为扩展预留接口,不建议使用。

--no-subscriptions

不转储订阅。

--no-tablespaces

不输出选择表空间的命令。使用该选项,无论默认表空间是哪个,在恢复过程中 所有对象都会被创建。

该选项只对文本格式有意义。针对归档格式,可以在调用gs_restore时指定选项。

• --no-unlogged-table-data

该参数为扩展预留接口,不建议使用。

--non-lock-table

该参数为扩展预留接口,不建议使用。

--include-alter-table

转储表删除列。该选项会记录列的删除。

--quote-all-identifiers

强制对所有标识符加引号。为了向后续版本迁移,且其中可能涉及引入额外关键词,在转储相应数据库时该选项会有帮助。

--section=SECTION

指定已转储的名称区段(pre-data、data和post-data)。

• --serializable-deferrable

转储过程中使用可串行化事务,以确保所使用的快照与之后的数据库状态一致;要实现该操作需要在无异常状况的事务流中等待某个点,因为这样才能保证转储成功,避免引起其他事务出现serialization_failure要重新再做。

但是该选项对于灾难恢复没有益处。对于在原始数据库进行升级的时候,加载一个数据库的拷贝作为报告或其他只读加载共享的转储是有帮助的。没有这个选项,转储会反映一个与任何事务最终提交的序列化执行不一致的状态。

如果当gs_dump启动时,读写事务仍处于非活动状态,即便使用该选项也不会对 其产生影响。如果读写事务处于活动状态,转储的开始时间可能会延迟一段不确 定的时间。

--use-set-session-authorization

输出符合SQL标准的SET SESSION AUTHORIZATION命令而不是ALTER OWNER 命令来确定对象所有权。这样令转储更加符合标准,但是如果转储文件中的对象的历史有些问题,那么可能不能正确恢复。并且,使用SET SESSION AUTHORIZATION的转储需要数据库系统管理员的权限才能转储成功,而ALTER OWNER需要的权限则低得多。

--with-encryption=AES128
 指定转储数据需用AES128进行加密。

--with-key=KEY

AES128密钥规则如下:

- 密钥长度为8~16个字符。
- 至少包含大写字母(A-Z)、小写字母(a-z)、数字(0-9)、非字母数字字符(限定为~!@#\$%^&*()-_=+\[{{}}];;,<.>/?)四类字符中的三类字符。

□ 说明

- 使用gs_dump工具进行加密导出时,仅支持plain格式导出。通过-F plain导出的数据,需要通过gsql工具进行导入,且如果以加密方式导入,在通过gsql导入时,需要指定--with-key参数。
- 不支持加密导出存储过程和函数。
- --with-salt=RANDVALUES
 qs_dumpall使用此参数传递随机值。
- --include-extensions
 - 在转储中包含扩展。
- --include-depend-objs

备份结果包含依赖于指定对象的对象信息。该参数需要同-t/--include-table-file参数关联使用才会生效。

--exclude-self

备份结果不包含指定对象自身的信息。该参数需要同-t/--include-table-file参数关 联使用才会生效。

--pipeline

使用管道传输密码,禁止在终端使用。

--dont-overwrite-file

文本、tar以及自定义格式情况下会重写现有文件。这对目录格式不适用。

例如:

设想这样一种情景,即当前目录下backup.sql已存在。如果在输入命令中输入-f backup.sql选项时,当前目录恰好也生成backup.sql,文件就会被重写。

如果备份文件已存在,且输入--dont-overwrite-file选项,则会报告附带'转储文件已经存在'信息的错误。

gs_dump -p *port_number* postgres -f *backup.sql* -F *plain* --dont-overwrite-file

山 说明

- -s/--schema-only和-a/--data-only不能同时使用。
- -c/--clean和-a/--data-only不能同时使用。
- --inserts/--column-inserts和-o/--oids不能同时使用,因为INSERT命令不能设置OIDS。
- --role和--rolepassword必须一起使用。
- --binary-upgrade-usermap和--binary-upgrade必须一起使用。
- --include-depend-objs/--exclude-self需要同-t/--include-table-file参数关联使用才会生效
- --exclude-self必须同--include-depend-objs一起使用。

连接参数:

-h, --host=HOSTNAME

指定主机名称。如果数值以斜杠开头,则被用作到Unix域套接字的路径。缺省从PGHOST环境变量中获取(如果已设置),否则,尝试一个Unix域套接字连接。该参数只针对openGauss外,对openGauss内本机只能用127.0.0.1。

例如: 主机名

环境变量: PGHOST

-p, --port=PORT

指定主机端口。在开启线程池情况下,建议使用 pooler port,即主机端口+1。 环境变量:PGPORT

-U, --username=NAME

指定所连接主机的用户名。

不指定连接主机的用户名时,用户默认系统管理员。

环境变量: PGUSER

-w, --no-password

不出现输入密码提示。如果主机要求密码认证并且密码没有通过其它形式给出,则连接尝试将会失败。 该选项在批量工作和不存在用户输入密码的脚本中很有帮助。

-W, --password=PASSWORD

指定用户连接的密码。如果主机的认证策略是trust,则不会对系统管理员进行密码验证,即无需输入-W选项;如果没有-W选项,并且不是系统管理员,"Dump Restore工具"会提示用户输入密码。

--role=ROLENAME

指定创建转储使用的角色名。选择该选项,会使gs_dump连接数据库后,发起一个SET ROLE角色名命令。当所授权用户(由-U指定)没有gs_dump要求的权限时,该选项会起到作用,即切换到具备相应权限的角色。某些安装操作规定不允许直接以超系统管理员身份登录,而使用该选项能够在不违反该规定的情况下完成转储。

 --rolepassword=ROLEPASSWORD 指定角色名的密码。

说明

如果openGauss有任何本地数据要添加到template1数据库,请谨慎将gs_dump的输出恢复到一个真正的空数据库中,否则可能会因为被添加对象的定义被复制,出现错误。要创建一个无本地添加的空数据库,需从template0而非template1复制,例如:

CREATE DATABASE foo WITH TEMPLATE template0;

tar归档形式的文件大小不得超过8GB(tar文件格式的固有限制)。tar文档整体大小和任何其他输出格式没有限制,操作系统可能对此有要求。

由gs_dump生成的转储文件不包含优化程序用来做执行计划决定的统计数据。因此,最好从某转储文件恢复之后运行ANALYZE以确保最佳效果。转储文件不包含任何ALTER DATABASE···SET命令,这些设置由gs_dumpall转储,还有数据库用户和其他完成安装设置。

示例

使用gs_dump转储数据库为SQL文本文件或其它格式的操作,如下所示。

示例中"Bigdata@123"表示数据库用户密码;"backup/MPPDB_backup.sql"表示导出的文件,其中backup表示相对于当前目录的相对目录;"37300"表示数据库服务器端口;"postgres"表示要访问的数据库名。

□ 说明

导出操作时,请确保该目录存在并且当前的操作系统用户对其具有读写权限。

示例1: 执行gs_dump,导出postgres数据库全量信息,导出的MPPDB_backup.sql文件格式为纯文本格式。

```
gs_dump -U omm -W Bigdata@123 -f backup/MPPDB_backup.sql -p 37300 postgres -F p gs_dump[port='37300'][postgres][2018-06-27 09:49:17]: The total objects number is 356. gs_dump[port='37300'][postgres][2018-06-27 09:49:17]: [100.00%] 356 objects have been dumped. gs_dump[port='37300'][postgres][2018-06-27 09:49:17]: dump database postgres successfully gs_dump[port='37300'][postgres][2018-06-27 09:49:17]: total time: 1274 ms
```

使用gsgl程序从纯文本导出文件中导入数据。

示例2:执行gs_dump,导出postgres数据库全量信息,导出的MPPDB_backup.tar文件格式为tar格式。

```
gs_dump -U omm -W Bigdata@123 -f backup/MPPDB_backup.tar -p 37300 postgres -F t gs_dump[port='37300'][postgres][2018-06-27 10:02:24]: The total objects number is 1369. gs_dump[port='37300'][postgres][2018-06-27 10:02:53]: [100.00%] 1369 objects have been dumped. gs_dump[port='37300'][postgres][2018-06-27 10:02:53]: dump database postgres successfully gs_dump[port='37300'][postgres][2018-06-27 10:02:53]: total time: 50086 ms
```

示例3: 执行gs_dump,导出postgres数据库全量信息,导出的MPPDB_backup.dmp 文件格式为自定义归档格式。

```
gs_dump -U omm -W Bigdata@123 -f backup/MPPDB_backup.dmp -p 37300 postgres -F c gs_dump[port='37300'][postgres][2018-06-27 10:05:40]: The total objects number is 1369. gs_dump[port='37300'][postgres][2018-06-27 10:06:03]: [100.00%] 1369 objects have been dumped. gs_dump[port='37300'][postgres][2018-06-27 10:06:03]: dump database postgres successfully gs_dump[port='37300'][postgres][2018-06-27 10:06:03]: total time: 36620 ms
```

示例4:执行gs_dump,导出postgres数据库全量信息,导出的MPPDB_backup文件格式为目录格式。

```
gs_dump -U omm -W Bigdata@123 -f backup/MPPDB_backup -p 37300 postgres -F d gs_dump[port='37300'][postgres][2018-06-27 10:16:04]: The total objects number is 1369. gs_dump[port='37300'][postgres][2018-06-27 10:16:23]: [100.00%] 1369 objects have been dumped. gs_dump[port='37300'][postgres][2018-06-27 10:16:23]: dump database postgres successfully gs_dump[port='37300'][postgres][2018-06-27 10:16:23]: total time: 33977 ms
```

示例5: 执行gs_dump,导出postgres数据库信息,但不导出/home/MPPDB_temp.sql 中指定的表信息。导出的MPPDB backup.sql文件格式为纯文本格式。

```
gs_dump -U omm -W Bigdata@123 -p 37300 postgres --exclude-table-file=/home/MPPDB_temp.sql -f backup/MPPDB_backup.sql gs_dump[port='37300'][postgres][2018-06-27 10:37:01]: The total objects number is 1367. gs_dump[port='37300'][postgres][2018-06-27 10:37:22]: [100.00%] 1367 objects have been dumped. gs_dump[port='37300'][postgres][2018-06-27 10:37:22]: dump database postgres successfully gs_dump[port='37300'][postgres][2018-06-27 10:37:22]: total time: 37017 ms
```

示例6: 执行gs_dump,仅导出依赖于指定表testtable的视图信息。然后创建新的testtable表,再恢复依赖其上的视图。

● 备份仅依赖于testtable的视图。

```
gs_dump -s -p 37300 postgres -t PUBLIC.testtable --include-depend-objs --exclude-self -f backup/MPPDB_backup.sql -F p
gs_dump[port='37300'][postgres][2018-06-15 14:12:54]: The total objects number is 331.
gs_dump[port='37300'][postgres][2018-06-15 14:12:54]: [100.00%] 331 objects have been dumped.
gs_dump[port='37300'][postgres][2018-06-15 14:12:54]: dump database postgres successfully
gs_dump[port='37300'][postgres][2018-06-15 14:12:54]: total time: 327 ms
```

- 修改testtable名称。
 - gsql -p 37300 postgres -r -c "ALTER TABLE PUBLIC.testtable RENAME TO testtable_bak;"
- 创建新的testtable表。
 CREATE TABLE PUBLIC.testtable(a int, b int, c int);
- 还原依赖于testtable的视图。
 gsql -p 37300 postgres -r -f backup/MPPDB_backup.sql

相关命令

6.3.2 gs_dumpall, 6.3.3 gs_restore

6.3.2 gs_dumpall

背景信息

gs_dumpall是openGauss用于导出所有数据库相关信息工具,它可以导出openGauss数据库的所有数据,包括默认数据库postgres的数据、自定义数据库的数据以及openGauss所有数据库公共的全局对象。

gs_dumpall工具由操作系统用户omm执行。

gs_dumpall工具在进行数据导出时,其他用户可以访问openGauss数据库(读或写)。

gs_dumpall工具支持导出完整一致的数据。例如,T1时刻启动gs_dumpall导出openGauss数据库,那么导出数据结果将会是T1时刻该openGauss数据库的数据状态,T1时刻之后对openGauss的修改不会被导出。

gs_dumpall时生成列不会被转储。

gs_dumpall在导出openGauss所有数据库时分为两部分:

- gs_dumpall自身对所有数据库公共的全局对象进行导出,包括有关数据库用户和组、表空间以及属性(例如,适用于数据库整体的访问权限)信息。
- gs_dumpall通过调用gs_dump来完成openGauss中各数据库的SQL脚本文件导出,该脚本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。

以上两部分导出的结果为纯文本格式的SQL脚本文件,使用gsql运行该脚本文件可以恢复openGauss数据库。

注意事项

- 禁止修改导出的文件和内容,否则可能无法恢复成功。
- 为了保证数据一致性和完整性,gs_dumpall会对需要转储的表设置共享锁。如果 某张表在别的事务中设置了共享锁,gs_dumpall会等待此表的锁释放后锁定此 表。如果无法在指定时间内锁定某张表,转储会失败。用户可以通过指定--lock-wait-timeout选项,自定义等待锁超时时间。
- 由于gs_dumpall读取所有数据库中的表,因此必须以openGauss管理员身份进行 连接,才能导出完整文件。在使用gsql执行脚本文件导入时,同样需要管理员权 限,以便添加用户和组以及创建数据库。

语法

gs_dumpall [OPTION]...

参数说明

通用参数:

-f, --filename=FILENAME将输出发送至指定文件。如果这里省略,则使用标准输出。

-v, --verbose

指定verbose模式。该选项将导致gs_dumpall向转储文件输出详细的对象注解和启动/停止次数,向标准错误流输出处理信息。

• -V, --version

打印gs_dumpall版本,然后退出。

--lock-wait-timeout=TIMEOUT

请勿在转储刚开始时一直等待以获取共享表锁。如果无法在指定时间内锁定某个表,就选择失败。可以以任何符合SET statement timeout的格式指定超时时间。

• -?, --help

显示gs_dumpall命令行参数帮助,然后退出。

转储参数:

-a, --data-only只转储数据,不转储模式(数据定义)。

• -c, --clean

在重新创建数据库之前,执行SQL语句清理(删除)这些数据库。针对角色和表空间的转储命令已添加。

-g, --globals-only只转储全局对象(角色和表空间),无数据库。

• -o. --oids

转储每个表的对象标识符(OIDs),作为表的一部分数据。该选项用于应用以某种方式参照了OID列的情况。如果不是以上这种情况,请勿使用该选项。

-O, --no-owner

不输出设置对象的归属这样的命令,以匹配原始数据库。默认情况下,gs_dumpall会发出ALTER OWNER或SET SESSION AUTHORIZATION语句设置所创建的模式元素的所属。如果脚本正在运行,该语句不会执行成功,除非是由系统管理员触发(或是拥有脚本中所有对象的同一个用户)。通过指定-O,编写一个任何用户都能存储的脚本,且该脚本会授予该用户拥有所有对象的权限。

-r, --roles-only

只转储角色,不转储数据库或表空间。

-s, --schema-only

只转储对象定义(模式),而非数据。

• -S, --sysadmin=NAME

在转储过程中使用的系统管理员名称。

• -t, --tablespaces-only

只转储表空间,不转储数据库或角色。

• -x, --no-privileges

防止转储访问权限(授权/撤销命令)。

• --column-inserts|--attribute-inserts

以INSERT命令带列名(INSERT INTO表(列、···)值···)方式导出数据。这会导致恢复缓慢。但是由于该选项会针对每行生成一个独立分开的命令,所以在重新加载某行时出现的错误只会导致那行丢失,而非整个表内容。

--disable-dollar-quoting

该选项将禁止在函数体前使用美元符号\$,并强制使用SQL标准字符串语法对其进行引用。

--disable-triggers

该参数为扩展预留接口,不建议使用。

--inserts

发出INSERT命令(而非COPY命令)转储数据。这会导致恢复缓慢。注意:如果 重排列顺序,可能会导致恢复整个失败。--column-inserts选项更加安全,虽然可 能更慢些。

--no-publications

不转储发布。

--no-security-labels

该参数为扩展预留接口,不建议使用。

--no-subscriptions

不转储订阅。

--no-tablespaces

请勿输出创建表空间的命令,也请勿针对对象选择表空间。使用该选项,无论默 认表空间是哪个,在恢复过程中所有对象都会被创建。

--no-unlogged-table-data

该参数为扩展预留接口,不建议使用。

--include-alter-table

导出表中已删除的列信息。

--quote-all-identifiers

强制对所有标识符加引号。为了向后续版本迁移,且其中可能涉及引入额外关键词,在转储相应数据库时该选项会有帮助。

• --dont-overwrite-file

不重写当前文件。

--use-set-session-authorization

输出符合SQL标准的SET SESSION AUTHORIZATION命令而不是ALTER OWNER 命令来确定对象所有权。这样令转储更加符合标准,但是如果转储文件中的对象的历史有些问题,那么可能不能正确恢复。并且,使用SET SESSION AUTHORIZATION的转储需要数据库系统管理员的权限才能转储成功,而ALTER OWNER需要的权限则低得多。

--with-encryption=AES128

指定转储数据需用AES128进行加密。

--with-key=KEY

AES128密钥规则如下:

- 密钥长度为8~16个字符。
- 至少包含大写字母(A-Z),小写字母(a-z),数字(0-9),非字母数字字符(限定为~!@#\$%^&*()-_=+\[{{}}];;,<.>/?)四类字符中的三类字符。
- --include-extensions

如果include-extensions参数被设置,将备份所有的CREATE EXTENSION语句。

- --include-templatedb 转储过程中包含模板库。
- --binary-upgrade该参数为扩展预留接口,不建议使用。
- --binary-upgrade-usermap="USER1=USER2" 该参数为扩展预留接口,不建议使用。
- --non-lock-table该参数仅供OM工具使用。
- --tablespaces-postfix 该参数为扩展预留接口,不建议使用。
- --parallel-jobs 指定备份进程并发数,取值范围为1~1000。
- --pipeline 使用管道传输密码,禁止在终端使用。

山 说明

- -g/--globals-only和-r/--roles-only不能同时使用。
- -g/--globals-only和-t/--tablespaces-only不能同时使用。
- -r/--roles-only和-t/--tablespaces-only不能同时使用。
- -s/--schema-only和-a/--data-only不能同时使用。
- -r/--roles-only和-a/--data-only不能同时使用。
- -t/--tablespaces-only和-a/--data-only不能同时使用。
- -g/--globals-only和-a/--data-only不能同时使用。
- --tablespaces-postfix和--binary-upgrade必须一起使用。
- --binary-upgrade-usermap和--binary-upgrade必须一起使用。
- --parallel-jobs和-f/--file必须一起使用。

连接参数:

-h. --host

指定主机的名称。如果取值是以斜线开头,它将用作Unix域套接字的目录。默认值取自PGHOST环境变量;如果没有设置,将启动某个Unix域套接字建立连接。该参数只针对openGauss外,对openGauss内本机只能用127.0.0.1。

环境变量: PGHOST

-l, --database

指定所连接的转储全局对象的数据库名称,并去寻找还有其他哪些数据库需要被转储。如果没有指定,会使用postgres数据库,如果postgres数据库不存在,会使用template1。

-p, --port

指定服务器所侦听的TCP端口或本地Unix域套接字后缀,以确保连接。默认值设置为PGPORT环境变量。

在开启线程池情况下,建议使用 pooler port,即侦听端口+1。

环境变量: PGPORT

• -U, --username

所连接的用户名。

环境变量: PGUSER

-w, --no-password

不出现输入密码提示。如果服务器要求密码认证并且密码没有通过其它形式给 出,则连接尝试将会失败。 该选项在批量工作和不存在用户输入密码的脚本中很 有帮助。

-W, --password

指定用户连接的密码。如果主机的认证策略是trust,则不会对系统管理员进行密码验证,即无需输入-W选项;如果没有-W选项,并且不是系统管理员,"Dump Restore工具"会提示用户输入密码。

--role

指定创建转储使用的角色名。选择该选项,会使gs_dumpall连接数据库后,发起一个SET ROLE角色名命令。当所授权用户(由-U指定)没有gs_dumpall要求的权限时,该选项会起到作用,即切换到具备相应权限的角色。某些安装操作规定不允许直接以系统管理员身份登录,而使用该选项能够在不违反该规定的情况下完成转储。

--rolepassword 指定具体角色用户的角色密码。

说明

由于gs_dumpall内部调用gs_dump,所以一些诊断信息参见6.3.1 gs_dump。

一旦恢复,最好在每个数据库上运行ANALYZE,优化程序提供有用的统计数据。

gs_dumpall恢复前需要所有必要的表空间目录才能退出;否则,对于处在非默认位置的数据库,数据库创建会失败。

示例

使用gs_dumpall一次导出openGauss的所有数据库。

□ 说明

gs_dumpall仅支持纯文本格式导出。所以只能使用gsql恢复gs_dumpall导出的转储内容。

```
gs_dumpall -f backup/bkp2.sql -p 37300
gs_dump[port='37300'][dbname='postgres'][2018-06-27 09:55:09]: The total objects number is 2371.
gs_dump[port='37300'][dbname='postgres'][2018-06-27 09:55:35]: [100.00%] 2371 objects have been dumped.
gs_dump[port='37300'][dbname='postgres'][2018-06-27 09:55:46]: dump database dbname='postgres' successfully
gs_dump[port='37300'][dbname='postgres'][2018-06-27 09:55:46]: total time: 55567 ms
gs_dumpall[port='37300'][2018-06-27 09:55:46]: dumpall operation successful
qs_dumpall[port='37300'][2018-06-27 09:55:46]: total time: 56088 ms
```

相关命令

6.3.1 gs_dump, 6.3.3 gs_restore

6.3.3 gs_restore

背景信息

gs_restore是openGauss提供的针对gs_dump导出数据的导入工具。通过此工具可由 gs_dump生成的导出文件进行导入。

qs_restore工具由操作系统用户omm执行。

主要功能包含:

• 导入到数据库

如果连接参数中指定了数据库,则数据将被导入到指定的数据库中。其中,并行导入必须指定连接的密码。导入时生成列会自动更新,并像普通列一样保存。

● 导入到脚本文件

如果未指定导入数据库,则创建包含重建数据库所必须的SQL语句脚本并写入到 文件或者标准输出。等效于直接使用gs_dump导出为纯文本格式。

命令格式

gs_restore [OPTION]... FILE

山 说明

- FILE没有短选项或长选项。用来指定归档文件所处的位置。
- 作为前提条件,需输入dbname或-l选项。不允许用户同时输入dbname和-l选项。
- 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)已存在于将要还原的数据库中的数据库对象。
- 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。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。

参数说明

通用参数:

- -d, --dbname=NAME连接数据库dbname并直接导入到该数据库中。
- -f, --file=FILENAME

指定生成脚本的输出文件,或使用-l时列表的输出文件。 默认是标准输出。

□ 说明

- -f不能同-d一起使用。
- -F, --format=c|d|t

指定归档格式。由于gs_restore会自动决定格式,因此不需要指定格式。 取值范围:

- c/custom: 该归档形式为6.3.1 gs_dump的自定义格式。
- d/directory:该归档形式是一个目录归档形式。
- t/tar:该归档形式是一个tar归档形式。

-l, --list

列出归档形式内容。这一操作的输出可用作-L选项的输入。注意如果像-n或-t的过滤选项与-l使用,过滤选项将会限制列举的项目(即归档形式内容)。

-v, --verbose

指定verbose模式。

-V, --version

打印qs restore版本,然后退出。

• -?, --help

显示gs_restore命令行参数帮助,然后退出。

导入参数:

-a, --data-only

只导入数据,不导入模式(数据定义)。gs_restore的导入是以追加方式进行的。

• -c, --clean

在重新创建数据库对象前,清理(删除)已存在于将要还原的数据库中的数据库 对象。

• -C, --create

导入数据库之前会先使用CREATE DATABASE创建数据库。(指定该选项后,-d指定的数据库仅用以执行CREATE DATABASE命令,所有数据依然会导入到创建的数据库中。)

• -e, --exit-on-error

当发送SQL语句到数据库时如果出现错误,请退出。默认状态下会继续,且在导入后会显示一系列错误信息。

-I, --index=NAME

只导入已列举的index的定义。允许导入多个index。如果多次输入-l index导入多个index。

例如:

gs_restore -h *host_name* -p *port_number* -d postgres -l *Index1* -l *Index2 backup/MPPDB_backup.tar*在上面这个例子中,Index1和Index2会被导入。

-j, --jobs=NUM

运行gs_restore最耗时的部分(如加载数据、创建index或创建约束)使用并发任务。该选项能大幅缩短导入时间,即将一个大型数据库导入到某一多处理器的服务器上。

每个任务可能是一个进程或一个线程,这由操作系统决定;每个任务与服务器进 行单独连接。

该选项的最优值取决于服务器的硬件设置、客户端以及网络。还包括这些因素,如CPU核数量、硬盘设置。建议是从增加服务器上的CPU核数量入手,更大的值(服务器上CPU核数量)在很多情况下也能导致数据文件更快的被导入。当然,过高的值会由于超负荷反而导致性能降低。

该选项只支持自定义归档格式。输入文件必须是常规文件(不能是像pipe的文件)。如果是通过脚本文件,而非直接连接数据库服务器,该选项可忽略。而且,多任务不能与--single-transaction选项一起使用。

-L. --use-list=FILENAME

只导入列举在list-file中的那些归档形式元素,导入顺序以它们在文件中的顺序为准。注意如果像-n或-t的过滤选项与-L使用,它们将会进一步限制导入的项目。

一般情况下,list-file是通过编辑前面提到的某个-l参数的输出创建的。文件行的位置可更改或直接删除,也可使用分号(;)在行的开始注出。

-n, --schema=NAME

只导入已列举的模式中的对象。

该选项可与-t选项一起用以导入某个指定的表。

多次输入-n schemaname可以导入多个模式。

例如:

gs_restore -h *host_name* -p *port_number* -d postgres -n *sch1* -n *sch2 backup/MPPDB_backup.tar* 在上面这个例子中,sch1和sch2会被导入。

-O, --no-owner

不输出设置对象的归属这样的命令,以匹配原始数据库。默认情况下,gs_restore 会发出ALTER OWNER或SET SESSION AUTHORIZATION语句设置所创建的模式元素的所属。除非是由系统管理员(或是拥有脚本中所有对象的同一个用户)进行数据库首次连接的操作,否则语句会失败。使用-O选项,任何用户名都可用于首次连接,且该用户拥有所有已创建的对象。

-P, --function=NAME(args)

只导入已列举的函数。请按照函数所在转储文件中的目录,准确拼写函数名称和 参数。

当-P单独使用时,表示导入文件中所有'function-name(args)'函数;当-P同-n一起使用时,表示导入指定模式下的'function-name(args)'函数;多次输入-P,而仅指定一次-n,表示所有导入的函数默认都是位于-n模式下的。

可以多次输入-n schema-name -P 'function-name(args)'同时导入多个指定模式下的函数。

例如:

gs_restore -h *host_name* -p *port_number* -d postgres -n *test1* -P '*Func1(integer)*' -n *test2* -P '*Func2(integer)*' *backup/MPPDB_backup.tar*

在上面这个例子中,test1模式下的函数Func1(i integer)和test2模式下的函数Func2(j integer)会被一起导入。

-s, --schema-only

只导入模式(数据定义),不导入数据(表内容)。当前的序列值也不会导入。

-S, --sysadmin=NAME

该参数为扩展预留接口,不建议使用。

-t, --table=NAME

只导入已列举的表定义、数据或定义和数据。该选项与-n选项同时使用时,用来指定某个模式下的表对象。-n参数不输入时,默认为PUBLIC模式。多次输入-n <schemaname> -t <tablename>可以导入指定模式下的多个表。

例如:

导入PUBLIC模式下的table1

gs_restore -h *host_name* -p *port_number* -d postgres -t *table1 backup/MPPDB_backup.tar*

导入test1模式下的test1和test2模式下test2

gs_restore -h *host_name* -p *port_number* -d postgres -n test1 -t test1 -*n test2* -t test2 *backup/MPPDB_backup.tar*

导入PUBLIC模式下的table1和test1模式下test1

gs_restore -h host_name -p port_number -d postgres -n PUBLIC -t table1 -n test1 -t table1 backup/ MPPDB backup.tar

□说明

-t不支持schema_name.table_name的输入格式。

-T, --trigger=NAME该参数为扩展预留接口。

-x, --no-privileges/--no-acl防止导入访问权限(GRANT/REVOKE命令)。

• -1, --single-transaction

执行导入作为一个单独事务(即把命令包围在BEGIN/COMMIT中)。 该选项确保要么所有命令成功完成,要么没有改变应用。该选项意为--exit-onerror。

--disable-triggers该参数为扩展预留接口,不建议使用。

--no-data-for-failed-tables

默认状态下,即使创建表的命令失败(如表已经存在),表数据仍会被导入。使用该选项,像这种表的数据会被跳过。如果目标数据库已包含想要的表内容,这种行为会有帮助。

该选项只有在直接导入到某数据库中时有效,不针对生成SQL脚本文件输出。

--no-publications不导入发布。

--no-security-labels该参数为扩展预留接口,不建议使用。

--no-subscriptions不导入订阅。

--no-tablespaces

不输出选择表空间的命令。使用该选项,无论默认表空间是哪个,在导入过程中所有对象都会被创建。

--section=SECTION

导入已列举的区段(如pre-data、data或post-data)。

--use-set-session-authorization

该选项用来进行文本格式的备份。

输出SET SESSION AUTHORIZATION命令,而非ALTER OWNER命令,用以决定对象归属。该选项使转储更加兼容标准,但通过参考转储中对象的记录,导入过程可能会有问题。使用SET SESSION AUTHORIZATION的转储要求必须是系统管理员,同时在导入前还需参考"SET SESSION AUTHORIZATION",手工对导出文件的密码进行修改验证,只有这样才能进行正确的导入操作,相比之下,ALTER OWNER对权限要求较低。

--pipeline

使用管道传输密码,禁止在终端使用。

须知

如果安装过程中有任何本地数据要添加到template1数据库,请谨慎将gs_restore的输出载入到一个真正的空数据库中;否则可能会因为被添加对象的定义被复制,而出现错误。要创建一个无本地添加的空数据库,需从template0而非template1复制,例如:

CREATE DATABASE foo WITH TEMPLATE template0;

 gs_restore不能选择性地导入大对象;例如只能导入那些指定表的对象。如果某个 归档形式包含大对象,那所有大对象都会被导入或一个都不会被导入,如果它们通过-L、-t或其他选项被排除。

山 说明

- 1.-d/--dbname 和 -f/--file 不能同时使用;
- 2. -s/--schema-only 和 -a/--data-only不能同时使用;
- 3. -c/--clean 和 -a/--data-only不能同时使用;
- 4. 使用--single-transaction时, -j/--jobs必须为单任务;
- 5. --role 和 --rolepassword必须一起使用。

连接参数:

-h, --host=HOSTNAME

指定的主机名称。如果取值是以斜线开头,他将用作Unix域套接字的目录。默认值取自PGHOST环境变量;如果没有设置,将启动某个Unix域套接字建立连接。该参数只针对openGauss外,对openGauss内本机只能用127.0.0.1。

-p, --port=PORT

指定服务器所侦听的TCP端口或本地Unix域套接字后缀,以确保连接。默认值设置为PGPORT环境变量。

在开启线程池情况下,建议使用 pooler port,即侦听端口+1。

- -U, --username=NAME 所连接的用户名。
- -w, --no-password

不出现输入密码提示。如果服务器要求密码认证并且密码没有通过其它形式给 出,则连接尝试将会失败。 该选项在批量工作和不存在用户输入密码的脚本中很 有帮助。

-W, --password=PASSWORD

指定用户连接的密码。如果主机的认证策略是trust,则不会对系统管理员进行密码验证,即无需输入-W参数;如果没有-W参数,并且不是系统管理员, "gs_restore"会提示用户输入密码。

--role=ROLENAME

指定导入操作使用的角色名。选择该参数,会使gs_restore连接数据库后,发起一个SET ROLE角色名命令。当所授权用户(由-U指定)没有gs_restore要求的权限时,该参数会起到作用,即切换到具备相应权限的角色。某些安装操作规定不允许直接以初始用户身份登录,而使用该参数能够在不违反该规定的情况下完成导入。

--rolepassword=ROLEPASSWORD 指定具体角色用户的角色密码。

示例

特例:执行gsql程序,使用如下选项导入由gs_dump/gs_dumpall生成导出文件夹(纯文本格式)的MPPDB_backup.sql文件到postgres数据库。

```
gsql -d postgres -p 15400 -W Bigdata@123 -f /home/omm/test/MPPDB_backup.sql
SET
SET
SET
SET
SET
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
CREATE INDEX
CREATE INDEX
CREATE INDEX
SET
CREATE INDEX
REVOKE
REVOKE
GRANT
GRANT
total time: 30476 ms
```

gs restore用来导入由gs dump生成的导出文件。

示例1:执行gs_restore,将导出的MPPDB_backup.dmp文件(自定义归档格式)导入到postgres数据库。

```
gs_restore -W Bigdata@123 backup/MPPDB_backup.dmp -p 15400 -d postgres
gs_restore: restore operation successful
gs_restore: total time: 13053 ms
```

示例2: 执行gs_restore,将导出的MPPDB_backup.tar文件(tar格式)导入到 postgres数据库。

```
gs_restore backup/MPPDB_backup.tar -p 15400 -d postgres
gs_restore[2017-07-21 19:16:26]: restore operation successful
gs_restore[2017-07-21 19:16:26]: total time: 21203 ms
```

示例3:执行gs_restore,将导出的MPPDB_backup文件(目录格式)导入到postgres数据库。

```
gs_restore backup/MPPDB_backup -p 15400 -d postgres
gs_restore[2017-07-21 19:16:26]: restore operation successful
gs_restore[2017-07-21 19:16:26]: total time: 21003 ms
```

示例4:执行gs_restore,使用自定义归档格式的MPPDB_backup.dmp文件来进行如下导入操作。导入PUBLIC模式下所有对象的定义和数据。在导入时会先删除已经存在的对象,如果原对象存在跨模式的依赖则需手工强制干预。

```
gs_restore backup/MPPDB_backup.dmp -p 15400 -d postgres -e -c -n PUBLIC
gs_restore: [archiver (db)] Error while PROCESSING TOC:
gs_restore: [archiver (db)] Error from TOC entry 313; 1259 337399 TABLE table1 gaussdba
gs_restore: [archiver (db)] could not execute query: ERROR: cannot drop table table1 because other objects
depend on it
DETAIL: view t1.v1 depends on table table1
HINT: Use DROP ... CASCADE to drop the dependent objects too.
Command was: DROP TABLE public.table1;
```

手工删除依赖,导入完成后再重新创建。

```
gs_restore backup/MPPDB_backup.dmp -p 15400 -d postgres -e -c -n PUBLIC
gs_restore[2017-07-21 19:16:26]: restore operation successful
gs_restore[2017-07-21 19:16:26]: total time: 2203 ms
```

示例5:执行gs_restore,使用自定义归档格式的MPPDB_backup.dmp文件来进行如下 导入操作。只导入PUBLIC模式下表table1的定义。

gs_restore backup/MPPDB_backup.dmp -p 15400 -d postgres -e -c -s -n PUBLIC -t table1 gs_restore[2017-07-21 19:16:26]: restore operation successful gs_restore[2017-07-21 19:16:26]: total time: 21000 ms

示例6:执行gs_restore,使用自定义归档格式的MPPDB_backup.dmp文件来进行如下 导入操作。只导入PUBLIC模式下表table1的数据。

gs_restore backup/MPPDB_backup.dmp -p 15400 -d postgres -e -a -n PUBLIC -t table1 gs_restore[2017-07-21 19:16:26]: restore operation successful gs_restore[2017-07-21 19:16:26]: total time: 20203 ms

相关命令

6.3.1 gs_dump, 6.3.2 gs_dumpall

6.3.4 qs backup

背景信息

openGauss部署成功后,在数据库运行的过程中,会遇到各种问题及异常状态。 openGauss提供了gs backup工具帮助openGauss备份、恢复重要数据、显示帮助信息 和版本号信息。

前提条件

- 可以正常连接openGauss数据库。
- 在进行还原时,需要保证各节点备份目录中存在备份文件。
- 需以操作系统用户omm执行gs_backup命令。

语法

- 备份数据库主机
 - gs_backup -t backup --backup-dir=BACKUPDIR [-h HOSTNAME] [--parameter] [--binary] [--all] [-l LOGFILE]
- 恢复数据库主机
 - gs_backup -t restore --backup-dir=BACKUPDIR [-h HOSTNAME] [--parameter] [--binary] [--all] [-l LOGFILE [--force]
- 显示帮助信息 gs_backup -? | --help
- 显示版本号信息
- gs_backup -V | --version

参数说明

gs_backup参数可以分为如下几类:

- 备份数据库主机参数:
 - -h

指定存储备份文件的主机名称。

取值范围:主机名称。如果不指定主机名称,则备份当前集群。

--backup-dir=BACKUPDIR

备份文件保存路径。

--parameter

备份参数文件,不指定--parameter、--binary、--all参数时默认只备份参数文件。

--binary

备份app目录下的二进制文件。

--all

备份app目录下的二进制文件、pg_hba.conf和postgsql.conf文件。

- -l

指定日志文件及存放路径。

默认值: \$GAUSSLOG/om/gs_backup-YYYY-MM-DD_hhmmss.log

- 恢复数据库主机参数:
 - -h

指定需要恢复主机的名称。

取值范围: 主机名称。如果不指定主机,则恢复对应的备份节点。

- --backup-dir=BACKUPDIR 恢复文件提取路径。

--parameter

恢复参数文件,不指定--parameter、--binary、--all参数时默认只恢复参数文件。

--binary

恢复二进制文件。

– --all

恢复二进制和参数文件。

– -l

指定日志文件及存放路径。

默认值: \$GAUSSLOG/om/gs_backup-YYYY-MM-DD_hhmmss.log

--force

节点的静态文件丢失后强行restore,仅限--all 或者--binary一起使用时才生效。

• 其他参数:

- -?, --help

显示帮助信息。

- -V, --version

显示版本号信息。

- -t

指定操作类型。

取值范围: backup 或者 restore。

示例

● 使用gs_backup脚本备份数据库主机。

gs_backup -t backup --backup-dir=/opt/software/gaussdb/backup_dir -h plat1 --parameter Backing up openGauss.
Parsing configuration files.
Successfully parsed the configuration file.
Performing remote backup.
Remote backup succeeded.
Successfully backed up openGauss.

● 使用gs_backup脚本恢复数据库主机。

gs_backup -t restore --backup-dir=/pt/software/gaussdb/backup_dir -h plat1 --parameter Restoring openGauss.
Parsing the configuration file.
Successfully parsed configuration files.
Performing remote restoration.
Remote restoration succeeded.
Successfully restored openGauss.

6.4 闪回恢复

闪回恢复功能是数据库恢复技术的一环,可以有选择性的撤销一个已提交事务的影响,将数据从人为不正确的操作中进行恢复。在采用闪回技术之前,只能通过备份恢复、PITR等手段找回已提交的数据库修改,恢复时长需要数分钟甚至数小时。采用闪回技术后,恢复已提交的数据库修改前的数据,只需要秒级,而且恢复时间和数据库大小无关。

□ 说明

ASTORE引擎暂不支持闪回功能。

6.4.1 闪回查询

背景信息

闪回查询可以查询过去某个时间点表的某个snapshot数据,这一特性可用于查看和逻辑重建意外删除或更改的受损数据。闪回查询基于MVCC多版本机制,通过检索查询旧版本,获取指定老版本数据。

前提条件

• undo retention time参数用于设置undo旧版本的保留时间。

语法

```
{[ ONLY ] table_name [ * ] [ partition_clause ] [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
[ TABLESAMPLE sampling_method ( argument [, ...] ) [ REPEATABLE ( seed ) ] ]
[TIMECAPSULE { TIMESTAMP | CSN } expression ]
[( select ) [ AS ] alias [ ( column_alias [, ...] ) ]
[with_query_name [ [ AS ] alias [ ( column_alias [, ...] ) ] ]
[function_name ( [ argument [, ...] ] ) [ AS ] alias [ ( column_alias [, ...] | column_definition [, ...] ) ]
[function_name ( [ argument [, ...] ] ) AS ( column_definition [, ...] )
[from_item [ NATURAL ] join_type from_item [ ON join_condition | USING ( join_column [, ...] ) ]
```

语法树中"TIMECAPSULE {TIMESTAMP | CSN} expression"为闪回功能新增表达方式,其中TIMECAPSULE表示使用闪回功能,TIMESTAMP以及CSN表示闪回功能使用具体时间点信息或使用CSN(commit sequence number)信息。

参数说明

TIMESTAMP

- 指要查询某个表在TIMESTAMP这个时间点上的数据,TIMESTAMP指一个具体的历史时间。

CSN

 指要查询整个数据库逻辑提交序下某个CSN点的数据,CSN指一个具体逻辑 提交时间点,数据库中的CSN为写一致性点,每个CSN代表整个数据库的一 个一致性点,查询某个CSN下的数据代表SQL查询数据库在该一致性点的相关 数据。

使用示例

● 示例1:

SELECT * FROM t1 TIMECAPSULE TIMESTAMP to_timestamp ('2020-02-11 10:13:22.724718', 'YYYY-MM-DD HH24:MI:SS.FF');

● 示例2:

SELECT * FROM t1 TIMECAPSULE CSN 9617;

• 示例3:

SELECT * FROM t1 AS t TIMECAPSULE TIMESTAMP to_timestamp ('2020-02-11 10:13:22.724718', 'YYYY-MM-DD HH24:MI:SS.FF');

● 示例4:

SELECT * FROM t1 AS t TIMECAPSULE CSN 9617;

6.4.2 闪回表

背景信息

闪回表可以将表恢复至特定时间点,当逻辑损坏仅限于一个或一组表,而不是整个数据库时,此特性可以快速恢复表的数据。闪回表基于MVCC多版本机制,通过删除指定时间点和该时间点之后的增量数据,并找回指定时间点和当前时间点删除的数据,实现表级数据还原。

前提条件

• undo retention time参数用于设置undo旧版本的保留时间。

语法

TIMECAPSULE TABLE table_name TO { TIMESTAMP | CSN } expression

使用示例

TIMECAPSULE TABLE t1 TO TIMESTAMP to_timestamp ('2020-02-11 10:13:22.724718', 'YYYY-MM-DD HH24:MI:SS.FF');

TIMECAPSULE TABLE t1 TO CSN 9617;

6.4.3 闪回 DROP/TRUNCATE

背景信息

- 闪回DROP:可以恢复意外删除的表,从回收站(recyclebin)中恢复被删除的表及其附属结构如索引、表约束等。闪回drop是基于回收站机制,通过还原回收站中记录的表的物理文件,实现已drop表的恢复。
- 闪回TRUNCATE:可以恢复误操作或意外被进行truncate的表,从回收站中恢复 被truncate的表及索引的物理数据。闪回truncate基于回收站机制,通过还原回收 站中记录的表的物理文件,实现已truncate表的恢复。

前提条件

- 开启enable_recyclebin参数,启用回收站。
- recyclebin_retention_time参数用于设置回收站对象保留时间,超过该时间的回收站对象将被自动清理。

相关语法

● 删除表

DROP TABLE table_name [PURGE]

• 清理回收站对象

```
PURGE { TABLE { table_name } 
| INDEX { index_name } 
| RECYCLEBIN 
}
```

闪回被删除的表

TIMECAPSULE TABLE { table_name } TO BEFORE DROP [RENAME TO new_tablename]

截断表

TRUNCATE TABLE { table_name } [PURGE]

• 闪回截断的表

TIMECAPSULE TABLE { table_name } TO BEFORE TRUNCATE

参数说明

- DROP/TRUNCATE TABLE table_name PURGE
 - 默认将表数据放入回收站中,PURGE直接清理。
- PURGE RECYCLEBIN
 - 表示清理回收站对象。
- TO BEFORE DROP

使用这个子句检索回收站中已删除的表及其子对象。

可以指定原始用户指定的表的名称,或对象删除时数据库分配的系统生成名称。

- 回收站中系统生成的对象名称是唯一的。因此,如果指定系统生成名称,那么数据库检索指定的对象。使用"select * from gs_recyclebin;"语句查看回收站中的内容。
- 如果指定了用户指定的名称,且如果回收站中包含多个该名称的对象,然后数据库检索回收站中最近移动的对象。如果想要检索更早版本的表,你可以这样做:
 - 指定你想要检索的表的系统生成名称。
 - 执行TIMECAPSULE TABLE ... TO BEFORE DROP语句,直到你要检索的表。
- 恢复DROP表时,只恢复基表名,其他子对象名均保持回收站对象名。用户可根据需要,执行DDL命令手工调整子对象名。
- 回收站对象不支持DML、DCL、DDL等写操作,不支持DQL查询操作(后续支持)。
- 闪回点和当前点之间,执行过修改表结构或影响物理结构的语句,闪回失败。执行过DDL的表进行闪回操作报错: "ERROR: The table definition of %s has been changed."。涉及namespace、表名改变等操作的DDL执行闪回操作报错: ERROR: recycle object %s desired does not exis;

RENAME TO

为从回收站中检索的表指定一个新名称。

• TO BEFORE TRUNCATE

闪回到TRUNCATE之前。

语法示例

```
PURGE TABLE t1;

PURGE TABLE t1;

PURGE TABLE "BIN$04LhcpndanfgMAAAAAANPw==$0";

PURGE INDEX i1;

PURGE INDEX "BIN$04LhcpndanfgMAAAAAANPw==$0";

PURGE RECYCLEBIN;

TIMECAPSULE TABLE t1 TO BEFORE DROP;

TIMECAPSULE TABLE t1 TO BEFORE DROP RENAME TO new_t1;

TIMECAPSULE TABLE "BIN$04LhcpndanfgMAAAAAANPw==$0" TO BEFORE DROP;

TIMECAPSULE TABLE "BIN$04LhcpndanfgMAAAAAANPw==$0" TO BEFORE DROP;

TIMECAPSULE TABLE "BIN$04LhcpndanfgMAAAAAANPw==$0" TO BEFORE DROP RENAME TO new_t1;
```

了 高危操作一览表

各项操作请严格遵守指导书操作,同时避免执行如下高危操作。

表7-1中描述在产品的操作与维护阶段,进行日常操作时应注意的严禁操作。

表 7-1 禁用操作

操作名称	操作风险
严禁修改数据目录下文件名,权限,内 容不能修改,不能删除内容。	导致数据库节点实例出现严重错误,并 且无法修复。
严禁删除数据库系统表或系统表数据。	删除系统表将导致无法正常进行业务操 作。

表7-2中描述在产品的操作与维护阶段,进行日常操作时应注意的高危操作。

表 7-2 高危操作

操作 分类	操作名称	操作风险	风险 等级	规避措施	重大操 作观察 项目
数据库	直接在配置文件中手动修改端口号。	导致数据 库启动不 了或者连 接不上。		尽量使用工具修改, 不要手动操作。	无。
	修改pg_hba.conf配置文件中的内容。	导致客户 端连接不 上。		严格根据产品手册操 作。	无。
	手动修改pg_xlog的内容。	导致数据 库无法启 动,数据 不一致。		尽量使用工具修改, 不要手动操作。	无。

操作分类	操作名称	操作风险	风险 等级	规避措施	重大操 作观察 项目
作业	使用kill -9终止作业 进程。	导致作业 占用的系 统资源无 法释放。	A A	尽量登录数据库使用 pg_terminate_backe nd, pg_cancel_backend 操作终止作业,或使 用Ctrl+C终止作业进 程。	观察资 源使用 情况。

8 日志参考

日志类型简介

在数据库运行过程中,会出现大量日志,既有保证数据库安全可靠的WAL日志(预写式日志,也称为Xlog),也有用于数据库日常维护的运行和操作日志等。在数据库发生故障时,可以参考这些日志进行问题定位和数据库恢复的操作。

日志类型

日志类型的详细说明请参见下表。

表 8-1 日志类型

类型	说明
系统日志	数据库系统进程运行时产生的日志,记录系统进程的异常信息。
操作日志	通过客户端工具(例如gs_guc)操作数据库时产生的日志。
Trace日志	通过gstrace工具指定启动trace功能的DN实例后,会记录大量的Trace 日志。这些日志可以用来分析数据库的异常信息。
黑匣子日 志	数据库系统崩溃的时候,通过故障现场堆、栈信息可以分析出故障发生时的进程上下文,方便故障定位。黑匣子具有在系统崩溃时,dump出进程和线程的堆、栈、寄存器信息的功能。
审计日志	开启数据库审计功能后,将数据库用户的某些操作记录在日志中,这 些日志称为审计日志。
WAL日志	又称为REDO日志,在数据库异常损坏时,可以利用WAL日志进行恢 复。由于WAL日志的重要性,所以需要经常备份这些日志。
性能日志	数据库系统在运行时检测物理资源的运行状态的日志,在对外部资源 进行访问时的性能检测,包括磁盘、Hadoop openGauss等外部资源 的访问检测信息。

系统日志

openGauss运行时数据库节点以及openGauss安装部署时产生的日志统称为系统日志。如果openGauss在运行时发生故障,可以通过这些系统日志及时定位故障发生的原因,根据日志内容制定恢复openGauss的方法。

日志文件存储路径

数据库节点的运行日志放在"/var/log/gaussdb/用户名/pg_log"中各自对应的目录下。

OM openGauss安装卸载时产生的日志放在"/var/log/gaussdb/用户名/om"目录下。

日志文件命名格式

数据库节点运行日志的命名规则:

postgresgl-创建时间.log

默认情况下,每日0点或者日志文件大于16MB或者数据库实例(数据库节点)重新启动后,会生成新的日志文件。

- CM的运行日志的命名规则:
 - cm_agent的日志: cm_agent-创建时间.log、cm_agent-创建时间current.log、system_call-创建时间.log、system_call-创建时间current.log。
 - cm_server的日志: cm_server-创建时间.log、cm_server-创建时间-current.log; key_event-创建时间.log、key_event-创建时间-current.log。
 - om_monitor的日志: om_monitor-创建时间.log、om_monitor-创建时间current.log。

其中,不带current标识符的文件是历史日志文件,带current标识符的文件是当前日志文件。最初调用进程时,进程会先创建一个带current标识符的日志文件,当该日志文件的大小超过16MB时,会将当前日志文件重命名为历史日志文件,并以当前时间生成新的当前日志文件。

日志内容说明

- 数据库节点每一行日志内容的默认格式: 日期+时间+时区+用户名称+数据库名称+会话ID+日志级别+日志内容
- cm_agent、cm_server、om_monitor每一行日志内容的默认格式:

时间+时区+会话ID+日志内容

SYSTEM_CALL系统调用日志记录了CM_AGENT在运行过程中调用工具命令的情况。

key_event每一行日志内容的默认格式: 时间+线程号+线程名: 关键事件类型+仲裁对象实例ID+仲裁细节

操作日志

操作日志是指数据库管理员使用工具操作数据库时以及工具被openGauss调用时产生的日志。如果openGauss发生故障,可以通过这些日志信息跟踪用户对数据库进行了哪些操作,重现故障场景。

日志文件存储路径

默认在"\$GAUSSLOG/bin"目录下,如果环境变量\$GAUSSLOG不存在或者变量值为空,则工具日志信息不会记录到对应的工具日志文件中,日志信息只会打印到屏幕上。

其中\$GAUSSLOG默认为"/var/log/gaussdb/用户名"。

□ 说明

如果使用om脚本部署时,则日志路径为 "/var/log/gaussdb/用户名"。

日志文件命名格式

日志文件命名格式为:

- 工具名-日志创建时间.log
- 工具名-日志创建时间-current.log

其中,"工具名-日志创建时间.log"是历史日志文件,"工具名-日志创建时间-current.log"是当前日志文件。

如果日志大小超过16MB,在下一次调用该工具时,会重命名当前日志文件为历史日志文件,并以当前时间生成新的当前日志文件。

例如将"gs_guc-2015-01-16_183728-current.log"重命名为 "gs_guc-2015-01-16_183728.log",然后重新生成"gs_guc-2015-01-17_142216current.log"。

维护建议

建议定时对过期的日志文件进行转储,以避免大量日志占用太多的磁盘空间和避免重要日志丢失。

审计日志

审计功能开启时会不断产生大量的审计日志,占用磁盘空间。用户可以根据磁盘空间的大小设置审计日志维护策略。

关于如何设置审计日志维护策略请参见《开发者指南》中"管理数据库安全 > 设置数据库审计 > 维护审计日志"章节。

WAL 日志

预写式日志WAL(Write Ahead Log,也称为Xlog)是指如果要修改数据文件,必须是在这些修改操作已经记录到日志文件之后才能进行修改,即在描述这些变化的日志记录刷新到永久存储器之后。在系统崩溃时,可以使用WAL日志对openGauss进行恢复操作。

日志文件存储路径

以一个数据库节点为例,默认在"/gaussdb/data/data dn/pg xlog"目录下。

其中"/gaussdb/data/data_dn"代表openGauss节点的数据目录。

日志文件命名格式

日志文件以段文件的形式存储的,每个段为16MB,并分割成若干页,每页8KB。对WAL日志的命名说明如下:一个段文件的名称由24个十六进制组成,分为三个部分,每个部分由8个十六进制字符组成。第一部分表示时间线,第二部分表示日志文件标

号,第三部分表示日志文件的段标号。时间线由1开始,日志文件标号和日志文件的段标号由0开始。

□ 说明

这些数字一般情况下是顺序增长使用的(要把所有可用数字都用光也需要非常长的时间),但也 存在循环使用的情况。

日志内容说明

WAL日志的内容取决于记录事务的类型,在系统崩溃时可以利用WAL日志进行恢复。 默认配置下,openGauss每次启动时会先读取WAL日志进行恢复。

维护建议

WAL日志对数据库异常恢复有重要的作用,建议定期对WAL日志进行备份。

性能日志

性能日志主要关注外部资源的访问性能问题。性能日志指的是数据库系统在运行时检测物理资源的运行状态的日志,在对外部资源进行访问时的性能检测,包括磁盘、Hadoop openGauss等外部资源的访问检测信息。在出现性能问题时,可以借助性能日志及时的定位问题发生的原因,能极大地提升问题解决效率。

日志文件存储路径

数据库节点的性能日志目录在"\$GAUSSLOG/gs_profile"中各自对应的目录下。

日志文件命名格式

数据库节点的性能日志的命名规则:

postgresql-创建时间.prf

默认情况下,每日0点或者日志文件大于20MB或者数据库实例(数据库节点)重新启动后,会生成新的日志文件。

日志内容说明

数据库节点每一行日志内容的默认格式:

主机名称+日期+时间+实例名称+线程号+日志内容。