# Mpp数据库工具指南





# MPP数据库工具手册

本文档介绍命令行工具和客户端程序的参考信息。

- **管理工具参考** 本章描述了MPP数据库提供的命令行管理工具相关内容
- **客户端工具参考** 描述MPP数据库随附的命令行客户端工具。
- 附加程序 本节介绍MPP数据库安装后的可用附加程序

# 管理工具参考

本章描述了MPP数据库提供的命令行管理工具相关内容

MPP数据库使用标准的PostgreSQL客户端和服务器程序,并且额外提供了用于管理分布式MPP数据库的管理工具。MPP数据库管理工具位于\$GPHOME/bin目录下。

Note: 在gpfdist的URL中引用IPv6地址或者在任何管理工具中使用数字形式的IP地址而不是主机名时,请把IP地址放在方括号中。在命令提示符使用时,最好的方式是将方括号转义或者把它们放在引号内。 例如:\[2620:0:170:610::11\]或'[2620:0:170:610::11]'。

下列是MPP数据库的管理工具。

analyzedb	gplogfilter
-----------	-------------

gpactivatestandby	gpmapreduce
gpaddmirrors	gpperfmon_install
gpbackup	gppkg
gpcgroup	gprebuildsystem (已弃用)
gpcheck (已弃用)	gprecoverseg
gpcheckcat	gpreload
gpchecknet (已弃用)	gprestore
gpcheckos (已弃用)	gpsizecalc (已弃用)
gpcheckperf	gpscp
gpconfig	gpskew (已弃用)
gpdeletesystem	gpseginstall
gpdetective (已弃用)	gpshrink
gpexpand	gpssh
gpfdist	gpssh-exkeys
gpinitstandby	gpstart
gpinitsystem	gpstate
gpload	gpstop
	gpsys1

iolimit
pgbouncer
pgbouncer.ini
pgbouncer-admin
pxf
pxf cluster

# 后端服务器程序

MPP数据库提供了下列标准的PostgreSQL数据库管理程序,它们位于\$GPHOME/bin中。为了处理MPP数据库系统的并行性和分布性,这些程序都经过了不同程度的修改。用户只能通过MPP数据库管理工具来访问这些程序。

Table 1. MPP数据库后端服务程序

程序名	描述	替代程序
initdb	在初始化一个MPP数据库集群时,这个程序会被gpinitsystem调用。它被用来创建集群内部每一个segment实例和master实例。	gpinitsystem
ipcclean	MPP数据库中没有使用	N/A

程序名	描述	替代程序
pg_basebacku p	该程序用来对某个单独的数据 库实例创建一个二进制拷贝。 MPP数据库 会在创建备用mast er实例或创建一个segment镜 像的完整拷贝时使用。不要使 用该工具备份MPP 数据库seg ment实例,因为它达不到MPP 一致性备份的要求。	gpinitstandby, gprecover seg
pg_controldata	MPP数据库中没有使用	gpstate
pg_ctl	在启动或者停止MPP数据库阵列时,这个程序会被gpstart和gpstop调用。在集群内部,它被用来以正确的选项并行地停止和启动segment实例和 mast er实例。	gpstart, gpstop
pg_resetxlog	不要使用 警告: 这个程序可能会导致数据丢失或者导致数据变得不可用。如果使用了这个程序,集群必须被重新初始化然后回复数据。	N/A
postgres	postgres可执行程序是实际处 理查询的服务器进程。	主postgres进程(postma ster)按照需要创建其他 的 postgres子进程以及po stgres会话来处理客户端 连接。

程序名	描述	替代程序
postmaster	postmaster开启接受客户端连接的postgres 数据库服务器监听进程。在MPP数据库中,MPP的master实例和每一个segment实例上都会运行一个postgres数据库监听进程。	在MPP数据库中,用户可以使用gpstart和 gpstop在系统中以正确的顺序和正确的选项一次性开启和关闭所有的postmaster进程(postgres进程)。

# analyzedb

一个提供对表的递增和并发ANALYZE操作的工具。对追加优化表来说, analyzedb只在统计数据不是最新的时候才更新统计信息。

### 概要

```
analyzedb -d dbname
     { -s schema |
     { -t schema.table
     [ -i col1[, col2, ...] |
     -x col1[, col2, ...]]}|
     { -f | --file} config-file }
     [-I | --list]
     [--gen profile only]
     [ -p parallel-level ]
     [ --full ]
     [ -v | --verbose ]
     [ --debug ]
     [-a]
      analyzedb { --clean last | --clean all }
      analyzedb --version
     analyzedb { -? | -h | --help }
```

### 描述

analyzedb工具递增并发地更新MPP数据库中指定表的表数据的统计信息。

在执行ANALYZE操作时,analyzedb创建了表元数据的快照并将它存储在Master主机的磁盘上。只有在表格被修改的情况下才执行ANALYZE操作。如果表或者分区自从上次分区依赖没有被修改过,则analyzedb会自动跳过表或分区,因为它已经包含了最新的统计信息。

- 对追加优化表来说,analyzedb如果统计数据数据不是最新的,则会逐渐分析统计数据。例如,如果表数据在表的统计数据收集之后改变了。如果没有该表的统计数据,则收集统计数据。
- 对于堆表,统计信息总是被更新。

即使表格统计信息是新的,也可以指定--full选项来更新追加优化表的统计信息。

默认情况下,analyzedb最多创建5个并发会话来并行分析表。 对于每个会话,analyzedb发出一个ANALYZE命令到 数据库并指定不同的表名。-p选项控制最大并发会话数。

#### 分区追加优化表

对于一个分区的追加优化表analyzedb检查分区表的根分区和叶子分区。 如果需要,该工具将更新非当前分区和根分区的统计信息。

GPORCA需要根分区统计信息。默认情况下,如果统计信息不存在,那么analyzedb 工具将收集分区表的根分区的统计信息。如果任何叶子节点有陈旧的统计信息,那么,analyzedb 也会刷新根分区的统计信息。刷新根节点的统计信息的成本和分析一个叶子分区相当。用户可以指定 --skip\_root\_stats来禁用分区表的根分区的统计信息的收集。

### 注意

如果表已经由DML或DDL命令(包括INSERT, DELETE, UPDATE, CREATE TABLE, ALTER TABLE 和 TRUNCATE)修改,则analyzedb工具会更新追加优化表的统计信息。该工具通过比较表的元数据和之前analyzedb操作中所保存的快照目录元数据来确定表是否已经被修改。表元数据的快照作为状态文件存储在MPP数据库主数据目录中的db\_analyze/<db\_name>/<timestamp>目录中。

该工具不会自动删除旧的快照信息。时间久了,这些快照会占用大量的磁盘空间。为了回收磁盘空间,您可以保留最近的快照信息,将旧的快照删掉。用户可以指定--clean\_last或者--clean\_all选项来删除由 analyzedb生成的状态文件。

如果用户未指定表,一组表或模式,那么analyzedb工具会根据需要收集所有系统。 统目录表和用户定义的表中的统计信息。

外部表不受analyzedb的影响。

包含空格的表名是不支持的。

在表上运行ANALYZE命令,(不使用analyzedb工具),不会更新表的元数据, analyzedb工具通常使用这些元数据通常来确定表的统计信息是否是最新的表的元数据。

### 选项

#### --clean last

删除由上一次analyzedb操作生成的状态文件。所有的其他选项,除了-d都会被忽略。

#### --clean all

删除由analyzedb生成的所有状态文件。 所有其他操作除了-d都会被忽略。

#### -d dbname

指定包含要分析的表的数据库的名称。如果未指定此选项,则从环境变量PGDATABASE中读取数据库名称。如果未设置PGDATABASE,则使用连接指定的用户名。

#### --debug

如果指定,则将日志级别设置为调试。在执行命令期间,调试级别信息被写入日志文件和命令行。 这些信息包括公用程序执行的命令和每个ANALYZE操作的持续时间。

#### -f config-file | --file config-file

包含要分析的表的列表的文本文件。可以指定当前目录的相对文件路径。该文件每行列出一个表。表名必须使用模式名称进行限定。或者,可以使用·i或 -x来指定列名。文件中国不允许其他选项。其他诸如--full 的选项必须在命令行中指定。

只有一个选项可用于指定要分析的文件:-f或--file,-t,或-s。

当在多个表上执行ANALYZE操作时候,analyzedb 会创建并行会话来并行分析表。该-p选项控制并发会话的最大数目。

在下面的示例中,第一行施加了一个ANALYZE操作对表public.nation,第二行仅对public.lineitem表的l\_shipdate和 l\_receiptdate列施加了一

#### 个ANALYZE操作。

public.nation

public.lineitem -i I shipdate, I receiptdate

#### --full

对所有指定的表执行ANALYZE操作。即使统计数据是最新的,也会执行 该操作。

#### --gen\_profile\_only

不执行ANALYZE操作,但是更新analyzedb收集的表统计信息快照。 如果有其他选项制定了表或者模式名,该工具仅更新指定表的快照信息。 如果ANALYZE命令在数据库表上运行并且您想更新analyzedb该表的快照信息,请指定该选项。

#### -i col1, col2, ...

可选的,必须用-t选项指定。对于使用-t选项指定的表,仅收集指定列的统计信息。

只能指定-i,或-x。这两个选项不能同时指定。

#### -l | --list

列出将用指定选项进行分析的表格。ANALYZE操作不执行。

#### -p parallel-level

并行分析的表的数量。*并行级别*可以是1到10之间的整数,包含首尾值。 默认值是5.

#### -s schema

指定要分析的模式。模式中的所有表将被分析。只能在命令行中指定一个模式名称。

只有一个选项可用于指定要分析的文件:-f或--file,-t,或-s。

#### -t schema.table

只收集*schema.table*的统计信息。 表名称必须使用模式名称进行限定。在命令行上只能指定一个表名。用户可以指定-f 选项来指定文件中的多个表,或者指定-s选项来指定模式中的所有表。

只有其中一个选项可用于指定要分析的文件:-f或--file, -t, 或-s。

#### -x col1, col2, ...

可选的,必须用-t选项来指定。对于使用-t选项指定的表,请排除指定列上

的收集的统计信息。统计信息只收集在未列出的列上。 只能指定-i, 或-x。这两个选项不能同时指定。

-a

静默模式,不提示用户确认。

- -h | -? | --help 展示在线帮助。
- -v | --verbose

如果指定,则将日志记录级别设置为verbose,以在命令执行过程中向日志文件和命令行写入附加信息。 这些信息包括所有要分析的表的列表(包括分区表的子叶子分区)。输出还包括每个ANALYZE 操作操作的持续时间。

--version

显示此工具的版本。

### 示例

仅收集一组表格列的统计信息的示例。在数据库mytest中,收集表 public.orders上的shipdate和receiptdate 列的统计信息:

analyzedb -d mytest -t public.orders -i shipdate, receiptdate

一个在表上收集统计信息并排除一组列的示例。在数据库mytest中,收集表public.foo的统计数据,并且不收集bar和test2列的统计信息。

analyzedb -d mytest -t public.foo -x bar, test2

指定包含表列表的文件的示例。此命令收集名为mytest的数据库中文件 analyze-tables列出的表的统计信息。

analyzedb -d mytest -f analyze-tables

如果用户未指定表,一组表或模式,则analyzedb实用程序会根据需要收集所有目录表和用户定义的表上的统计信息。此命令刷新数据库mytest中的系统目录表和用户定义的表上的表统计信息。

```
analyzedb -d mytest
```

用户可以创建一个PL/Python函数来运行analyzedb程序作为MPP的数据库函数。 这个例子CREATE FUNCTION命令创建一个用户定义的PL/ Python函数,运行 analyzedb工具并在命令行上显示输出。指定analyzedb选项作为函数参数。

```
CREATE OR REPLACE FUNCTION analyzedb(params TEXT)

RETURNS VOID AS

$BODY$

import subprocess

cmd = ['analyzedb', '-a'] + params.split()

p = subprocess.Popen(cmd, stdout=subprocess.PIPE, stderr=subprocess.STDOUT)

# verbose output of process

for line in iter(p.stdout.readline, "):

plpy.info(line);

p.wait()

$BODY$

LANGUAGE plpythonu VOLATILE;
```

当这个SELECT命令由gpadmin用户运行时,该analyzedb工具对数据库mytest中的public.mytable表执行一个分析操作。

```
SELECT analyzedb('-d mytest -t public.mytable');
```

Note: 注意:要创建一个PL/Python函数,该PL/Python程序语言必须在数据库中注册。例如,以管理员运行的 CREATE LANGUAGE命令将吧PL/Python注册成不受信任的语言:

CREATE LANGUAGE plpythonu;

### 另见

**ANALYZE** 

# gpactivatestandby

激活备机并使其成为MPP数据库系统的活动主机。

### 概要

gpactivatestandby -v

 $gpactivates tand by \ \hbox{-} ? \ | \ \hbox{-} h \ | \ \hbox{--help}$ 

### 描述

gpactivatestandby工具激活了一台备份主机上的后备Master实例 ,并将其作为MPP数据库系统的活动Master实例运行。被激活的后备Master节点有效 地成为MPP数据库的Master来接受端口上的客户端连接。

在初始化后备Master时,默认情况下使用与活动Master相同的端口。有关后备Master端口的信息,请参考gpinitstandby。

用户必须从正在激活的Master主机上运行此工具,而不是在被停用的故障Master主机上运行。 运行此工具时候假设用户为系统配置了一台后备Master主机。 (参见gpinitstandby)。

#### 该工具执行以下步骤:

- 停止后备Master上的同步进程 (walreceiver)
- 使用日志更新后备Master的系统目录表
- 激活后备Master成为系统中新的活动Master
- 重启带有新Master主机的MPP数据库系统

在MPP的主Master主机变得不可操作时,备份的后备Master主机充当"温备份"的角色。 后备Master通过事务日志复制进程(walsender和walreceiver) 保持最新状态,这两个进程一个运行在主Master,一个运行在后备Master,并使得主Master和 后备Master之间保持数据同步。

如果主Master出现故障,日志复制进程将关闭,后备Master可以通过使用gpactivatestandby 工具在其位置激活。一旦后备Master激活,复制的日志将用于重建最后一次成功提交事务之后 MPP中Master主机的状态。

为了使用gpactivatestandby来激活一个新的主Master主机,之前充当主Master的Master主机则不能再运行。该工具会检查已被禁用的Master主机的数据目录中的postmaster.pid文件,如果发现该文件,则会假定旧的Master机仍处于活动状态。在某些情况下,在运行gpactivatestandby(例如,如果被禁用的Master主机进程被意外终止)之前,用户可能需要从被禁用的Master主机的数据目录中删除 postmaster.pid。

激活后备Master之后,运行ANALYZE来更新数据库查询统计信息。例如:

psql dbname -c 'ANALYZE;'

将后备的Master激活成为主Master之后,该MPP数据库系统将不再配置有后备 Master。用户可能想要用 gpinitstandby工具指定另外一台主机作为新的后备。

### 选项

-a (不要提示)

不要提示用户确认。

-d standby\_master\_datadir

用户正在激活的Master主机的数据目录的绝对路径。

如果未指定此选项,gpactivatestandby将使用用户正在激活的Master主机上的 MASTER\_DATA\_DIRECTORY环境变量设置。如果指定了这一选项,它会覆盖 MASTER\_DATA\_DIRECTORY的设置。

如果无法确定目录,则该工具将返回错误。

-f (强制激活)

使用此选项强制激活备份的Master主机。只有在确定后备Master主机和主 Master主机一致的情况下,才能使用此选项。

-l logfile\_directory

写入日志文件的目录。默认为~/gpAdminLogs。

-q (没有屏幕输出)

以静默模式运行。命令输出不会显示在屏幕上,但是仍然会写入日志文件。

-v (显示工具的版本)

显示这一工具的版本、状态、上次更新的日期和校验和。

-?|-h|--help (帮助)

显示在线帮助。

### 示例

激活后备Master主机并使其成为MPP数据库系统的活动Master实例(从正在激活的备份Master主机运行):

gpactivatestandby -d /gpdata

### 另见

gpinitsystem, gpinitstandby

# gpaddmirrors

将镜像Segment添加到最初没有配置镜像的MPP数据库系统。

### 概要

### 描述

gpaddmirrors工具为初始仅配置了主Segment实例的现有MPP数据库系统配置 镜像 Segment实例。该工具创建镜像实例并开始主Segment实例和镜像 Segment实例之间的在线复制进程。一旦所有 的镜像与其主Segment同步好, 用户的MPP数据库系统就建立了完全的数据冗余。

Important: 在在线复制进程中,MPP数据库应处于静止状态,负载和其他查询不应该运行。

默认情况下,该工具将提示用户输入将创建镜像Segment数据目录的文件系统位置。如果用户不想被提示,可以使用-m选项传递包含文件系统位置的文件。

镜像位置和端口必须与用户的主Segment数据位置和端口不同。如果用户创建了额外的文件空间,则还将提示用户为每个文件空间提供镜像位置。

该工具使用预定义的命名习惯在指定位置中为每个镜像Segment实例创建唯一的数据目录。必须为镜像Segment 实例声明与主Segment实例相同数量的文件系统位置。如果用户希望在同一位置创建镜像数据目录,可以多次指定同样的目录名称,或者可以为每个镜像输入不同的数据位置。对于文件系统位置,请输入绝对路径。例如:

Enter mirror segment data directory location 1 of 2 > /gpdb/mirror Enter mirror segment data directory location 2 of 2 > /gpdb/mirror

或

Enter mirror segment data directory location 1 of 2 > /gpdb/m1

Enter mirror segment data directory location 2 of 2 > /gpdb/m2

还有,用户可以运行gpaddmirrors工具,并使用-i选项提供详细的 配置文件。如果用户希望镜像Segment位于完全不同于主Segment的主机集合上,这非常有用。镜像配置文件的 格式是:

mirror<row id>=<contentID>:<address>:<port>:<data dir>

此处row\_id是文件中的行号,*contentID*是segment实例的content\_id, *address*是segment所在的主机名或IP地址,*port*是交互端口, data\_dir是segment实例数据目录。

例如:

mirror0=0:sdw1-1:60000:/gpdata/mir1/gp0 mirror1=1:sdw1-1:60001:/gpdata/mir2/gp1

gp\_segment\_configuration系统目录表可以帮助用户确定当前的主Segment配置,以便用户可以规划镜像Segment配置。例如,运行以下的查询:

=# SELECT dbid, content, address as host\_address, port, datadir FROM gp\_segment\_configuration ORDER BY dbid;

如果在后补的镜像主机上创建镜像,则新的镜像Segment主机必须预先安装MPP数据库软件,并且被配置为 与现有的主Segment主机完全相同。

用户必须确保运行gpaddmirrors的用户(gpadmin用户)有权在指定的数据目录位置写入。用户可能想要在Segment主机上创建这些目录并且在运行gpaddmirrors之前将这些目录的拥有者改为(chown)适当的用户。

Note: 该工具在系统内部采用SSH连接执行各项操作任务。在大型MPP集群、云部署或每台主机部署了大量的 segment实例时,可能会遇到超过主机最大授权连接数限制的情况。此时需要考虑更新SSH配置参数MaxStartups 以提高该限制。更多关于SSH配置的选项,请参考您的Linux分发版的SSH文档。

### 选项

-a (不提示)

以静默模式运行 - 不提示输入信息。如果使用这一选项,必须使用-m或-i提供一个配置文件。

-B parallel\_processes

并行启动的镜像设置进程的数量。如果未指定,则该工具将启动最多10个并行进程,具体取决于需要设置多少个镜像Segment实例。

-d master\_data\_directory

主数据目录。如果没有指定,将使用 为\$MASTER\_DATA\_DIRECTORY设置的值。

-i mirror\_config\_file

一个配置文件,每个要创建的镜像Segment对应一行。用户必须为系统中的每个主Segment都在该文件中列出一个镜像Segment。该文件格式如下(如gp segment configuration 目录表中的每个属性):

mirror<row\_id>=<contentID>:<address>:<port>:<data\_dir>

此处row\_id是文件中的行号,*contentID*是segment实例的content\_id, *address*是segment所在的主机名或IP地址,*port*是交互端口, data\_dir是 segment实例数据目录。

#### -l logfile\_directory

用来写日志的目录,默认为~/gpAdminLogs。

#### -m datadir config file

包含将创建镜像数据目录的文件系统位置列表的配置文件。如果未提供,该工具将提示用户输入位置。 文件中的每一行都指定一个镜像数据目录位置。例如:

#### /gpdata/m1

/gpdata/m2 /gpdata/m3 /gpdata/m4

#### -o output\_sample\_mirror\_config

如果用户不确定如何布置-i选项使用的镜像配置文件,用户使用这一选项运行 gpaddmirrors来生成一个基于主Segment配置的镜像配置文件示例。该工具 将提示用户输入用户的镜像Segment数据目录位置(除非用户使用-m在一个文件 中提供了这些位置)。然后,用户可以编辑此文件根据需要将主机名更改为后补的镜像主机。

#### -p port offset

可选。这个数字被用于计算用于镜像Segment的数据库端口和复制端口。 默认偏移量为1000。 镜像端口分配计算如下:

主机端口 + 偏移量 = 镜像数据库端口

例如,如果主Segment的端口为50001,则默认情况下,其镜像将使用数据库端口51001。

#### -s (散布镜像)

在可用主机上散布镜像Segment。默认情况下,将把一组镜像Segment集合在一起放在不同于其主Segment集合的后补主机上。镜像散布将把每个镜像放置在MPP数据库阵列中的不同主机上。只有在阵列中有足够数量的主机(主机数大于每个主机的Segment实例数)时,才允许使用散布镜像方式。

#### -v (详细模式)

将日志记录输出设置为详细。

#### --version (显示工具版本)

显示该工具的版本。

#### -? (help)

显示在线帮助。

### 示例

使用和主要数据相同的一组主机向现有的MPP数据库系统添加镜像。通过在当前主Segment端口号上加上 100来计算镜像数据库端口和复制端口:

\$ gpaddmirrors -p 100

使用和主要数据不同的一组主机向现有的MPP数据库系统添加镜像:

\$ gpaddmirrors -i mirror\_config\_file

其中mirror\_config\_file看起来像这样:

mirror0=0:sdw1-1:52001:/gpdata/mir1/gp0

mirror1=1:sdw1-2:52002:/gpdata/mir2/gp1 mirror2=2:sdw2-1:52001:/gpdata/mir1/gp2 mirror3=3:sdw2-2:52002:/gpdata/mir2/gp3

使用gpaddmirrors -o输出一个要使用的镜像配置文件样例:

\$ gpaddmirrors -o /home/gpadmin/sample\_mirror\_config

### 另见

gpinitsystem, gpinitstandby, gpactivatestandby

# gpbackup

创建一个可以给gprestore工具使用的MPP数据库备份。

### 概要

```
gpbackup --dbname database name
  [--backup-dir directory]
  [--compression-level |evel|
  [--data-only]
  [--debug]
  [--exclude-schema schema name]
  [--exclude-table schema.table]
  [--exclude-table-file file name]
  [--include-schema schema_name]
  [--include-table schema.table]
  [--include-table-file file name]
  [--incremental [--from-timestamp backup-timestamp]]
  [--jobs int]
  [--leaf-partition-data]
  [--metadata-only]
  [--no-compression]
  [--plugin-config config file location]
  [--quiet]
  [--single-data-file]
  [--verbose]
  [--version]
  [--with-stats]
gpbackup --help
```

### 描述

gpbackup工具用来备份数据库的内容到元数据文件集合和数据文件集合,这些文件可以被 gprestore工具用来恢复数据库。当备份数据库时,可以通过指定表级别和模式级别选项来备份某些特定的表。例如,可以通过组合模式级别和表级别选项来备份模式下的所有表,并排除某一张单独的表。

默认情况下,gpbackup备份指定数据库的对象和MPP数据库系统全局对象。可以通过 gprestore工具指定可选参数--with-globals来恢复全局对象。更多 信息请见备份或还原中包含的对象。

gpbackup默认将MPP数据库备份对象元数据文件和DDL文件存储在Master数据目录下。 MPP Segment使用COPY ... ON SEGMENT命令将数据备份为压缩 CSV数据文件,并存储在每个Segment的数据目录下。更多信息请见理解备份文件。

可以通过指定--backup-dir选项将Master和Segment主机上的数据备份到一个绝对路径下。可以通过指定其他选项来过滤备份集合来排除或包含指定的表。

可以通过指定 --incremental选项来启动增量备份。增量备份在追加优化表或 表分区上的变化数据小于未发生变化的数据时有效。有关增量备份的详细信息,请见使用gpbackup和gprestore创建增量备份。

指定--jobs选项(1 job),MPP数据库Master主机上的每个 gpbackup操作都会启动一个单独的事务。COPY ... ON SEGMENT命令在每个Segment主机上并行执行备份任务。备份进程会在备份的每张表 上唤起ACCESS SHARE锁。在表锁处理过程中,数据库处于静止状态。

当备份操作完成后,gpbackup会返回状态码。详情请见返回码。

gpexpand正在初始化新segments时,gpbackup工具不能运行。集群扩展完成后,在扩展之前创建的备份不能被gprestore使用。

gpbackup可以在备份操作完成后发送Email状态通知。客户可以在配置文件中创建工具发送和接收服务器信息。详情请见配置邮件通知。

Note: 该工具在系统内部采用SSH连接执行各项操作任务。在大型MPP集群、云部署或每台主机部署了大量的 segment实例时,可能会遇到超过主机最大授权连接数限制的情况。此时需要考虑更新SSH配置参数MaxStartups 以提高该限制。更多关于SSH配置的选项,请参考您的Linux分发版的SSH文档。

### 选项

#### --dbname database name

必须。指定要备份的数据库。

#### --backup-dir directory

可选。复制所有备份文件(元数据文件和数据文件)到指定路径。必须指定 directory 为绝对路径(不能是相对路径)。如果不提供该选项,元数据文件会保存在MPP Master主机的

\$MASTER\_DATA\_DIRECTORY/backups/YYYYMMDD/YYYYMMDDhhn路径下。Segment主机的CSV数据文件保存

在<seg\_dir>/backups/YYYYMMDD/YYYYMMDDhhmmss/路径下。当指定该选项时,文件会被复制到备份目录的子路径下。

该选项不能和--plugin-config一起使用。

#### --compression-level level

可选。指定用来压缩数据文件的gzip压缩级别(从1到9)。默认为1。注意,gpbackup 默认使用压缩。

#### --data-only

可选。仅将表数据备份到CSV文件,不备份用来重建表和其他数据库对象的元数据文件。

#### --debug

可选。显示操作期间的调试信息。

#### --exclude-schema schema name

可选。指定要被排除备份的模式名。该参数可以指定多次以排除多个模式。该选项不能与--include-schema一起使用,也不能与表过滤选项例如:--include-table一起使用。更多信息请见过滤备份或恢复的内容。

#### --exclude-table schema.table

可选。指定要排除在备份中的表。被排除的表必须为格式<schemaname>.<table-name>。如果表或模式名使用了任何非小写字母、数字或下划线,必须用双引号包裹名字。该选项可以指定多次。 该选项不能与-exclude-schema或--include-table同时使用。

该选项不能和--leaf-partition-data组合使用。虽然您可以指定子分区名字,但是 gpbackup会忽略分区名称。

更多信息详见过滤备份或恢复的内容。

#### --exclude-table-file file\_name

可选。指定一个包含需要排除在备份之外的列表文件。每行必须指定一个

单独的表,格式为 <schema-name>.<table-name>。文件不能包含多余的行。 如果表或模式名使用了任何非小写字母、数字或下划线,必须用双引号包裹名字。该选项可以指定多次。 该选项不能与--exclude-schema或--include-table同时使用。

该选项不能和--leaf-partition-data组合使用。虽然您可以在--exclude-table-file 中指定子分区名字,但是gpbackup会忽略分区名称。

更多信息请见过滤备份或恢复的内容。

#### --include-schema schema name

可选。指定要包含在备份中的数据库模式名。可以多次指定该参数来包含多个备份模式。如果指定该选项,任何不包含在--include-schema中的模式都不会放在备份集中。该选项不能与选项 --exclude-schema、--include-table或 --include-table-file—起使用。更多详情请见过滤备份或恢复的内容。

#### --include-table schema.table

可选。指定一个要包含在备份中的表。该表的格式必须为<schemaname>.<table-name>。 如果表或模式名使用了任何非小写字母、数字或下划线,必须用双引号包裹名字。有关模式和表名中支持的字符信息详情,请见模式和表名中的模式和表名部分。

该选项可以指定多次。该选项不能和模式筛选选项--include-schema或--exclude-table-file等一起使用。

客户也可以指定序列或视图的名字。

如果指定该选项,该工具不会自动备份依赖对象。客户必须指定相关依赖对象。例如,如果备份了视图 ,客户必须备份试图所用的表。如果备份表用到了一个序列,客户必须也备份相关的序列。

可以在表名处通过指定--leaf-partition-data选项来指定表子分区名字, 仅备份指定的分区。当指定子分区备份时,子分区数据和分区表的元数据信息都会被备份。

更多信息请见过滤备份或恢复的内容。

#### --include-table-file file\_name

可选。指定一个备份时需要的表名列表文件。文件中的每行必须定义为单独的表名,格式为: <schema-name>.<table-name>。该文件不能包含多余的列。 有关模式和表名中支持的 字符信息详情,请见模式和表名中的模式和表名部分。

任何没有列在文件中的表都会被排除在备份集以外。该选项不能与模式过滤选项--include-schema 或--exclude-table-file等一起使用。

该选项也可以指定序列和视图名称。

如果指定该选项,该工具不会自动备份依赖对象。客户必须指定相关依赖对象。例如,如果备份了视图 ,客户必须备份试图所用的表。如果备份表用到了一个序列,客户必须也备份相关的序列。

可以在表名处通过指定--leaf-partition-data选项来指定表子分区名字, 仅备份指定的分区。当指定子分区备份时,子分区数据和分区表的元数据信息都会被备份。

更多信息请见过滤备份或恢复的内容。

#### --incremental

指定该选项可以增加一个增量备份到增量备份集中。一个备份集包括一个全量备份和一个或多个增量备份。 该备份集合必须是连续性的,以保证在恢复时是可用的。

默认情况下,gpbackup会找出最近的备份。如果该备份是全备份,工具会创建一个备份集合。如果备份为增量备份,工具会将备份增加到已经存在的备份集。增量备份被增加到备份集的最后。可以通过指定--from-timestamp来覆盖默认的行为。

#### --from-timestamp backup-timestamp

可选。指定备份时间戳。指定的备份必须已经有增量备份存在。如果指定的备份只有全量备份,该工具 会创建一个备份集。如果指定的备份是增量备份,工具会将增量备份直接备份到该备份集合。

该选项必须与--leaf-partition-data选项一起使用。不能与--data-only 或--metadata-only一起使用。

如果备份操作不能增加备份到已经存在的增量备份集或者不能使用备份来创建一个备份集,工具将不能创建备份 并且工具会返回错误。

有关创建和使用增量备份的详细信息,请见使用gpbackup和gprestore创建增量备份。

#### --jobs int

可选。指定进行表备份的并行任务数量。默认gpbackup采用1个任务(数据库连接)。增加该参数的值可以提高数据备份的速度。多运行多个任务时,每个任务会在一个单独的事务中备份表。例如,如果指定--jobs 2,

工具会创建2个进程,每个进程启动一个单独的事务 ,工具会使用这两个进程并行的备份表。

Important: 如果指定超过1的值,数据库在工具从表上获取备份用的锁时处于静默状态。 如果数据库操作正在那些被备份的表上执行,此时进行备份的话这些表在不同事务中的锁处理和一致性 问题不能被保证。

该选项不能和--metadata-only、--single-data-file或 --plugin-config一起使用。

#### --leaf-partition-data

可选。对于分区表,该选项定义后会为每一个子分区创建一个单独的数据文件,而不是为整个表创建一个完成的文件(默认)。使用该选项要求客户通过--include-table-file选项指定包含在备份中的单独子节点分区。该选项不能与--exclude-table-file或 --exclude-table选项同时使用。

#### --metadata-only

可选。仅创建可用来重建数据库的元数据文件(DDL),不备份表的实际数据。

#### --no-compression

可选。不对表数据CSV文件进行压缩。

#### --plugin-config config-file\_location

指定gpbackup插件配置文件的位置,该文件是一个YAML格式的文本文件。 该文件包含gpbackup在备份操作期间使用的配置信息。

如果在备份数据库时指定了--plugin-config选项,那么在从备份 恢复数据库时也需要同样指定对应的配置信息。

该选项不能与--backup-dir一起使用。

有关使用存储插件应用的详细信息,请见使用gpbackup存储插件部分。

#### --quiet

可选。不显示所有非告警、非错误日志信息。

#### --single-data-file

可选。在每个Segment主机上为所有备份的表创建一个单独的数据文件。 默认情况下,每个gpbackup 会为每张表创建一个压缩CSV文件。

Note: 如果选用--single-data-file选项来将每个Segment上的数据备份为1个文件,就不能在使用gprestore时指定--jobs选项为超过1的值来执行并行恢复。

#### --verbose

可选。打印详细日志信息。

#### --version

可选。打印工具版本号并退出。

#### --with-stats

可选。在备份集中包含查询计划统计信息。

#### --help

显示在线帮助信息。

### 返回码

gpbackup完成后会返回如下返回码之一。

- **0** 成功完成备份。
- 1 备份完成,没有严重错误。具体信息见日志文件。
- 2-备份失败,有严重错误。具体信息见日志文件。

### 模式和表名

当指定表过滤项--include-table或--include-table-file 为列表文件时,gpbackup支持备份模式或表名包含以下特殊字符时。

如果在命令行指定--include-table选项时,名字包含大写字母或特殊字符时,名字必须被 单引号包裹。

gpbackup --dbname test --include-table 'my#1schema'.'my\_\$42\_Table'

当使用--include-table-file选项,表名被放在文件中时,不需要使用单引号。例如,使用--include-table-file选项备份两个表。

my#1schema.my\_\$42\_Table my#1schema.my \$590 Table

Note: --include-table和--include-table-file选项不支持模式名或表名包含双引号(")、点号(.)、换行符(\n)或空格符()。

### 示例

备份"demo"数据库中的所有模式和表,包括MPP数据库全局系统对象:

\$ gpbackup --dbname demo

备份"demo"数据库中除了"twitter"模式以外的所有模式和表:

\$ gpbackup --dbname demo --exclude-schema twitter

仅备份"demo"数据库中的"twitter"模式:

\$ gpbackup --dbname demo --include-schema twitter

备份"demo"数据库中的所有模式和表,包括MPP数据库全局系统对象,将所有备份文件复制到 /home/gpadmin/backup路径下。

\$ gpbackup --dbname demo --with-stats --backup-dir /home/gpadmin/backup

该示例使用--include-schema和--exclude-table 备份指定模式,除了单独的一张表。

\$ gpbackup --dbname demo --include-schema mydata --exclude-table mydata.addresses

--exclude-schema选项不能与表过滤选项一起使用,例如--include-table。

### 另见

gprestore、使用gpbackup和gprestore并行备份和使用带有gpbackup和gprestore的S3存储插件

### gpcgroup

资源管理器。

### 概要

```
gpcgroup { init | set | destroy | list }
    [-f host_file | -h hostname [-h hostname ...]] [-S]
    [-G group { [-R read_speed] [-W write_speed] [-N send_speed] }]
gpcgroup help
```

### 先决条件

- 分布式MPP的所有节点主机支持cgroup,并且blkio子系统和net\_cls子系统已经挂载。
- 当前操作用户有sudo权限。

### 描述

gpcgroup工具用于管理资源限制所需的控制组(cgroup)。用户可以通过 gpcgroup创建和删除控制组,也可以设置每个控制组的资源配额。使用之前建 议用户先了解一下Linux的cgroup。

在使用gpcgroup之前,用户必须确保分布式MPP已经安装并且环境变量 GPHOME已经设置,涉及到的主机之间已经建立可信的主机设置。

Note:该工具在系统内部采用SSH连接执行各项操作任务。在大型MPP集群、云部署或每台主机部署了大量的segment实例时,可能会遇到超过主机最大授权连接数限制的情况。此时需要考虑更新SSH配置参数MaxStartups以提高该限制。更多关于SSH配置的选项,请参考您的Linux分发版的SSH文档。

操作系统重启之后所有gpcgroup的操作会丢失,需要重新初始化控制组,设置资源配额等。

### 选项

init (初始化)

初始化资源管理,即创建所有支持的控制组,目前支持expand和basebackup控制组,此操作需要sudo权限。init可以重复执行,如果控制组已经存在则会忽略。

#### destroy (销毁)

销毁资源管理,即删除所有在初始化时创建的控制组,此操作需要sudo权限。

#### set (设置)

设置资源配额,可以设置的资源有磁盘读写速度、网络速度,支持动态调整,即时生效。

#### list (查看)

#### 查看控制组及资源配置。

help (帮助) 查看帮助信息。

#### -f host\_file

可选,指定包含将参与操作的主机列表的文件的名称。主机文件的语法是每行一个主机。 如果-f和-h都没有指定,将通过MASTER\_DATA\_DIRECTORY变量读取并连接数据库,从数据库系统包获取所有节点的主机名。

#### -h hostname

可选,指定将参与操作的单个主机名。用户可以多次使用-h 选项来指定多个主机名。

#### -G group

必选,只对set操作有效。指定要设置资源配额的控制组,目前支持的控制组有epxand和basebackup。

#### -R read\_speed

可选,设置控制组的磁盘读速度,单位是MB/s,范围是0到1024,其中0表示无限制。 所有的磁盘的读速度分别被设置为read\_speed指定的值,如果使用该控制组范围内的进程涉及到了多个磁盘的读操作,每个磁盘的读速度上限都不能超过read\_speed。 注意,如果要读的文件在操作系统缓存,该限制无效。

#### -W write\_speed

可选,只对set操作有效。设置控制组的磁盘写速度,单位是MB/s,范围是0到1024,其中0表示无限制。 所有的磁盘的写速度分别被设置为write\_speed指定的值,如果使用该控制组范围内的进程涉及到了多个磁盘的写操作, 每个磁盘的读速度上限都不能超过write\_speed。注意,只对不使用操作系统的写操作有效。

#### -N send\_speed

可选,只对set操作有效。设置控制组的网络发送速度,单位是MB/s,范围是0到1024,其中0表示无限制。 所有的网卡的发送速度分别被设置为send\_speed

指定的值,如果使用该控制组范围内的进程涉及到了多个网卡的发送,每个网卡的发送速度上限都不能超过send\_speed。 注意,限制带宽使用了tc工具,如果当前用户没有tc权限,将执行sudo tc。

#### -S (特殊权限)

可选,只对init和destroy操作有效。给tc工具添加或删除"s"特殊权限。初始化时指定了-S选项,那么设置网络带宽限速时不需要再使用sudo权限。

### 示例

初始化host file文件指定主机的资源管理器:

\$ gpcgroup init -f host\_file

设置expand控制组的读速度为10MB/s,网络发送速度为5MB/s:

\$ gpcgroup set -G expand -R 10 -N 5

设置h112主机的basebackukp控制组的写速度为5.5MB/s:

\$ gpcgroup set -h h112 -G basebackup -W 5.5

#### 查看配置:

\$ gpcgroup list

## gpcheckcat

gpcheckcatgpcheckcat工具测试MPP数据库目录表的不一致性。

该工具在\$GPHOME/bin/lib中。

### 概要

```
gpcheckcat [ options] [ dbname]

Options:
    -g dir
    -p port
    -P password
    -U user_name
    -S {none | only}
    -O
    -R test_name
    -C catalog_name
    -B parallel_processes
    -v
    -A

gpcheckcat -I

gpcheckcat -?
```

### 描述

gpcheckcat工具运行多种测试检查数据库目录的不一致。 某些测试不能与其他负载语句同时运行, 否则结果将不可用。运行gpcheckcat时,要以受限模式重

新启动数据库,否则,gpcheckcat 可能会由于正在进行的数据库操作而报告不一致,这会与实际的不一致数量不同。如果用户在不停止数据库活动的情况下运行 gpcheckcat,请使用-O选项运行它。

Note: 每当用户运行该工具时,它都会检查并删除指定数据库中的孤立的临时数据库方案(没有会话ID的临时方案)。该工具在命令行上显示孤立临时模式检查的结果,并在日志中记录该结果。

目录不一致是MPP数据库系统表之间发生的不一致。一般来说,有三种不一致:

- Segment级的系统表不一致。例如,包含表数据的系统表与包含列数据的系统表之间的不一致。另一个例子是一个系统表在本该唯一的列中包含重复。
- Segment之间相同系统表之间的不一致。例如,一个系统表在一个Segment 上缺少一行,但其他Segment具有这一行。作为另一个例子,特定的行列数 据的值在不同的Segment之间不同,例如表所有者或表访问特权。
- 持久化系统表对象状态和文件系统对象间的不一致。例如,一个文件在数据库目录中存在,但数据库系统表中不存在相应的对象。

## 选项

-A

在MPP数据库安装的所有数据库上运行gpcheckcat。

-B parallel processes

并行运行的进程数量。

gpcheckcat工具尝试确定要同时使用的进程数(批尺寸)。该工具假定它可以为每个进程使用最小为20MB的缓冲区。并行进程的最大数量是MPP数据库Segment实例的数量。 当该工具开始检查目录时,它会显示所使用的并行进程的数量。

Note: 如果返回的错误数量超过缓冲区大小,则该工具可能会耗尽内存。如果发生内存不足错误,可以使用 -B选项降低批尺寸。例如,如果该工具显示批尺寸为936并且内存不足,则可以指定 -B 468并行运行468个进程。

-C catalog\_table

对指定的目录表运行交叉一致性、外键和ACL测试。

-g data\_directory

生成SQL脚本来修复目录不一致。脚本被放置在data\_directory中。

-|

列出gpcheckcat测试。

-O

只运行可以在线 (不受限) 模式运行的gpcheckcat测试。

-p port

这一选项指定MPP数据库使用的端口。

-P password

连接到MPP数据库的用户的密码。

-R test name

指定要运行的测试。某些测试只有当MPP数据库处于受限模式时才能运行。

这些是可以执行的测试:

acl - 对访问控制特权的交叉一致性检查

duplicate - 检查重复项

foreign\_key - 检查外键

inconsistent - 对Master和Segment不一致性的交叉一致性检查

missing extraneous - 对缺少的或无关的项的交叉一致性检查

owner - 检查表的拥有关系是否与Master数据库不一致

orphaned\_toast\_tables -检查孤立的TOAST表

Note: 有很多种情况都会导致TOAST表变成孤立表,此时不会生成修复脚本,要求进行手工元数据维护。 其中一种情况是*pg\_class*中的reltoastrelid指向一个不正确的TOAST表(不能匹配到TOAST表)。另外一种情况可能是*pg\_class*表的reltoastrelid和pg\_depend丢失(双孤立

TOAST表)。如果需要手工修复元数据,gpcheckcat会显示详细的元数据更新步骤。

part\_integrity - 检查*pg\_partition*分支的完整性、带OID的分区、 分区分布 策略

part\_constraint - 检查分区表上的约束

unique\_index\_violation - 检查有唯一索引约束的列的表中是否有重复项

dependency - 检查不存在对象的依赖关系 (仅限于受限模式)

distribution\_policy - 检查随机分布表上的约束 (仅限于受限模式)

namespace - 检查缺少方案定义的方案 (仅限于受限模式)

pgclass - 检查没有任何对应的*pg\_attribute*项的(仅限于受限模式) *pg\_class*项。

-S {none | only}

指定这一选项以控制对MPP数据库安装中所有数据库(如*pg\_database*)上共享的目录表的测试。

值none禁用对共享目录表的测试。值only仅测试共享目录表。

-U user\_name

连接到MPP数据库的用户。

-? (帮助)

显示在线帮助。

-v (详细模式)

显示所执行测试的详细信息。

# 沖解

该工具可识别缺少属性的表,并将其以非标准格式显示在输出中的各个位置。在

显示输出信息之后,该工具还会以格式 *database.schema.table.segment\_id* 显示缺少属性的表的摘要列表。

如果gpcheckcat检测到不一致的OID(对象ID)信息,它将生成一个或者多个包含SQL查询的验证文件。 用户可以运行SQL查询来查看有关OID不一致的详细信息,并调查不一致之处。这些文件在gpcheckcat 被调用的目录中生成。

#### 这是该文件的格式:

gpcheckcat.verify.dbname.catalog\_table\_name.test\_name.TIMESTAMP.sql

这是当gpcheckcat检测到数据库mydb中的目录表 $pg_type$ 中的不一致OID(对象ID)信息时,它创建的验证文件名示例:

gpcheckcat.verify.mydb.pg\_type.missing\_extraneous.20150420102715.sql

#### 这是验证文件中查询的一个示例:

```
SELECT *
FROM (
SELECT relname, oid FROM pg_class WHERE reltype
IN (1305822,1301043,1301069,1301095)
UNION ALL
SELECT relname, oid FROM gp_dist_random('pg_class') WHERE reltype
IN (1305822,1301043,1301069,1301095)
) alltyprelids
GROUP BY relname, oid ORDER BY count(*) desc;
```

# gpcheckpert

验证指定主机的基准硬件性能。

### 概要

```
gpcheckperf -d test_directory [-d test_directory ...]
     {-f hostfile_gpcheckperf | - h hostname [-h hostname ...]}
     [-r ds] [-B block_size] [-S file_size] [-D] [-v|-V]

gpcheckperf -d temp_directory
     {-f hostfile_gpchecknet | - h hostname [-h hostname ...]}
     [ -r n|N|M [--duration time] [--netperf] ] [-D] [-v | -V]

gpcheckperf -?

gpcheckperf --version
```

# 描述

gpcheckperf工具在指定的主机上启动会话并运行以下性能测试:

- 磁盘I/O测试(dd测试)—要测试逻辑磁盘或文件系统的顺序吞吐性能,该工具使用dd命令,该命令是一个标准的UNIX工具。它记录花费多长时间在磁盘上读写一个大文件,并以兆字节(MB)每秒为单位计算磁盘I/O性能。默认情况下,用于测试的文件尺寸按照主机上的总随机访问内存(RAM)的两倍计算。这确保了测试是真正地测试磁盘I/O而不是使用内存缓存。
- 内存带宽测试(流) 为了测试内存带宽,该工具使用STREAM基准程序来测量可持续的内存带宽(以MB/s为单位)。这测试用户的系统在不涉及CPU计算性能情况下是否受系统内存带宽的限制。在数据集较大的应用程序中(如在MPP数据库中),低内存带宽是一个主要的性能问题。如果内存带

宽明显低于CPU的理论带宽,则会导致CPU花费大量的时间等待数据从系统内存到达。

● 网络性能测试(gpnetbench\*) — 为了测试网络性能(以及MPP 数据库 Interconnect的性能),该工具运行一种网络基准测试程序,该程序当前主机 发送5秒钟的数据流到测试中包含的每台远程主机。数据被并行传输到每台远程主机,并以兆字节(MB)每秒报告最小、最大、 平均和中位网络传输速率。如果汇总的传输速率比预期慢(小于100MB/s),则可以使用-r n 选项串行地运行该网络测试以获取每台主机的结果。要运行全矩阵带宽测试,用户可以指定-r M ,这将导致每台主机都发送和接收来自指定的每台其他主机的数据。该测试最适用于验证交换结构是否可以承受全矩阵负载。

为了指定要测试的主机,请使用-f选项指定包含主机名列表的文件,或使用-h选项在命令行上指名单个主机名。如果运行网络性能测试,主机文件中的所有项必须是同一子网内的网络接口。如果用户的Segment主机具有在不同子网上配置的多个网络接口,请为每个子网运行一次网络测试。

用户还必须指定至少一个测试目录(使用-d)。运行gpcheckperf 的用户必须具有对所有远程主机上指定测试目录的写入权限。对于磁盘I/O测试,测试目录应与用户的 Segment数据目录(主Segment和/或镜像Segment)相对应。对于内存带宽和网络测试,测试程序文件需要临时目录。

在使用gpcheckperf之前,用户必须在涉及性能测试的主机之间建立可信的主机设置。 用户可以使用gpssh-exkeys工具更新已知主机文件并在主机之间交换公钥 (如果尚未这样做的话)。请注意,gpcheckperf调用gpssh 和gpscp,这些MPP工具也必须在\$PATH中。

### 选项

-B block\_size

指定用于磁盘I/O测试的块大小(以KB或MB为单位)。缺省值是32KB,与MPP数据库页面大小相同。最大块大小是1 MB。

-d test\_directory

对于磁盘I/O测试,指定要测试的文件系统目录位置。用户必须具有对性能测试中涉及的所有主机上测试目录的写入权限。用户可以多次使用-d选项指定多个测试目录 (例如,测试主数据目录和镜像数据目录的磁盘I/O)。

#### -d temp directory

对于网络和流测试,指定单个目录,测试程序文件在测试期间将被复制到该目录。用户必须具有 对测试中涉及的所有主机上该目录的写入权限。

#### -D (显示每台主机的结果)

报告每个主机的磁盘I/O测试的性能结果。缺省情况下,仅报告具有最低和最高性能的主机的结果,以及所有主机的总体和平均性能。

#### --duration *time*

以秒 (s)、分钟 (m)、小时 (h) 或天数 (d) 指定网络测试的持续时间。默认值是15秒。

### -f hostfile\_gpcheckperf

对于磁盘I/O和流测试,请指定一个包含将参与性能测试的主机名的文件名称。主机名是必需的,用户可以选择指定每个主机的后补用户名和/或SSH端口号。主机文件的语法是每行一台主机 ,如下所示:

[username@]hostname[:ssh\_port]

### -f hostfile\_gpchecknet

对于网络性能测试,主机文件中的所有项都必须是同一子网内的主机地址。如果用户的Segment 主机在不同子网上配置有多个网络接口,请为每个子网运行一次网络测试。例如(包含互连子网1 的Segment主机地址名的主机文件):

sdw1-1

sdw2-1

sdw3-1

#### -h hostname

指定将参与性能测试的单个主机名(或主机地址)。用户可以多次使用-h 选项来指定多个主机名。

### --netperf

指定应该用netperf二进制文件来执行网络测试,而不是MPP网络测试。 要使用此选项,用户必须从http://www.netperf.org下载 netperf并且安装到所有MPP主机(Master和Segment)的 \$GPHOME/bin/lib目录中。

### $-r ds\{n|N|M\}$

指定要运行的性能测试,默认是dsn:

- 磁盘I/O测试 (d)
- 流测试 (s)
- 网络性能测试,串行(n)、并行(N)或全矩阵(M)模式。可选的--duration选项指定了运行网络测试的时间(以秒为单位)。要使用并行(N)模式,用户必须在偶数台主机上运行测试。如果用户宁愿使用netperfnetperf(http://www.netperf.org)而不是MPP网络测试,用户必须下载它并安装到所有MPP主机(Master和)

网络测试,用户必须下载它并安装到所有MPP主机 (Master和 Segment) 的\$GPHOME/bin/lib目录中。 然后,用户可以指定可选的--netperf选项来使用 netperf二进制文件而不是默认的gpnetbench\*工具。

### -S file\_size

指定用于-d所指定的所有目录的磁盘I/O测试的总文件尺寸。 *file\_size*应该等于主机上总RAM的两倍。如果未指定,则默认值是 在执行gpcheckperf的主机上的总RAM的两倍,这确保了测试是真正地 测试磁

盘I/O而不是使用内存缓存。用户可以以KB、MB或GB为单位指定尺寸。

-v (详细模式) |-V (非常详细模式)

详细 (Verbose) 模式显示性能测试运行时的进度和状态信息。非常详细 (Very Verbose) 模式显示该工具生成的所有输出消息。

--version

显示该工具的版本。

-? (帮助)

显示在线帮助。

## 示例

使用/data1和/data2作为测试目录在文件 host file中的所有主机上运行磁盘I/O和

### 内存带宽测试:

\$ gpcheckperf -f hostfile\_gpcheckperf -d /data1 -d /data2 -r ds

在名为*sdw1*和*sdw2* 的主机上只使用测试目录 /data1运行磁盘I/O测试。显示单个主机结果并以详细模式运行:

\$ gpcheckperf -h sdw1 -h sdw2 -d /data1 -r d -D -v

使用测试目录/tmp,运行并行网络测试,其中hostfile\_gpcheck\_ic\*指定同一Interconnect子网内的所有网络接口的主机地址名称:

\$ gpcheckperf -f hostfile\_gpchecknet\_ic1 -r N -d /tmp
\$ gpcheckperf -f hostfile\_gpchecknet\_ic2 -r N -d /tmp

运行与上面相同的测试,但使用netperf而不是MPP网络测试 (注意, netperf必须安装在所有MPP主机上的 \$GPHOME/bin/lib中) :

\$ gpcheckperf -f hostfile\_gpchecknet\_ic1 -r N --netperf -d /tmp
\$ gpcheckperf -f hostfile\_gpchecknet\_ic2 -r N --netperf -d /tmp

### 另见

gpssh, gpscp, gpcheck

# gpconfig

在MPP数据库系统中所有的Segment上设置服务器配置参数。

### 概要

### 描述

gpconfig工具允许用户在MPP数据库系统中所有实例(Master、Segment和镜像)的postgresql.conf文件中设置、 复原或查看配置参数。设置参数时,如果需要,还可以为Master指定一个不同的值。 例如,诸如max\_connections之类的参数要求Master的设置 不同于Segment的设置。如果要设置或复原全局参数或仅可对Master设置的参数, 请使用--masteronly选项。

gpconfig只能用来管理某些参数。例如,用户不能使用它来设置 port等参数,这些参数对每个Segment实例都不同。使用-I (list) 选项查看gpconfig支持的配置参数的完整列表。

当gpconfig在Segment的postgresql.conf 文件中设置配置参数时,新的参数设置将总是显示在该文件的底部。当用户使用 gpconfig移除配置参数时,gpconfig会在所有 Segment的postgresql.conf文件中把该参数注释掉,从而恢复系统 默认

设置。例如,如果使用gpconfig 删除(注释掉)一个参数,并且 稍后把它添加回来(设置新值),则该参数会有两个实例,一个被注释掉,另一个被启用并添加到postgresql.conf文件的底部。

设置参数之后,用户必须重新启动其MPP数据库系统,或者重新加载postgresql.conf 文件以使得更改生效。是否需要重新启动或者加载取决于被设置的参数。

有关服务器配置参数的更多信息,请参阅MPP数据库参考手册。

要显示系统中当前参数的设置值,请使用-s选项。

gpconfig使用以下环境变量连接到MPP数据库的 Master实例并获取系统配置信息:

- PGHOST
- PGPORT
- PGUSER
- PGPASSWORD
- PGDATABASE

### 选项

-c | --change param\_name

通过在postgresql.conf文件的底部添加新的设置来改变配置参数的设置。

-v | --value value

用于由-c选项指定的配置参数的值。默认情况下,此值将应用于所有 Segment及其镜像、Master和后备Master。

非单个字符或数字的参数值必须用单引号包裹(')。例如,包括空格或特殊字符的字符串必须用单引号包裹。如果要在字符串参数重嵌入单引号,需要用2个单引号或反斜杠进行转移(\')。

工具会在将值写入postgresql.conf时带着单引号。

-m | --mastervalue master\_value

用于由-c选项指定的配置参数的Master值。如果指定,则该值仅适用于Master和后备Master。该选项只能与-v一起使用。

### --masteronly

当被指定时,gpconfig将仅编辑Master的postgresql.conf文件。

-r | --remove param name

通过注释掉postgresql.conf文件中的项删除配置参数。

-l | --list

列出所有被gpconfig工具支持的配置参数。

-s | --show param\_name

显示在MPP数据库系统中所有实例(Master和Segment)上使用的配置参数的值。如果实例中参数值存在差异,则工具将显示错误消息。使用-s选项运行 gpconfig将直接从数据库中读取参数值,而不是从postgresql.conf文件中读取。如果用户使用gpconfig在所有Segment中设置配置参数,然后运行gpconfig -s来验证更改,用户仍可能会看到以前的(旧)值。用户必须重新加载配置文件(gpstop -u)或重新启动系统(gpstop -r)以使更改生效。

--file

对于配置参数,显示在MPP数据库系统中的所有Segment (Master和 Segment) 上的postgresql.conf 文件中的值。如果实例中的参数值存在差异,则工具会显示一个消息。必须与-s选项一起指定。

例如,使用ALTER ROLE为用户设置配置参数statement\_mem为64MB,而postgresql.conf文件中的值为128MB。运行命令gpconfig -s statement\_mem --file显示 128MB。用户运行的命令gpconfig -s statement\_mem显示64MB。

与--file-compare选项一起指定时无效。

#### --file-compare

对于配置参数,将当前MPP数据库值与主机(Master和Segment) 上postgresql.conf 文件中的值进行比较。postgresql.conf files文件中的值表示重新启动MPP数据库时的值。

如果值不一样,该工具显示来自所有主机的值,如果所有主机的值一样,则该程序显示摘要报告。

与--file选项一起指定时无效。

#### --skipvalidation

覆盖gpconfig的系统验证检查,并允许用户对任何服务器配置参数进行操作,包括隐藏参数和gpconfig无法更改的受限参数。当与-l选项(列表)一起使用时,它显示受限参数的列表。

Warning: 使用此选项设置配置参数时要格外小心。

#### --verbose

在gpconfig命令执行期间显示额外的日志信息。

--debug

设置日志输出级别为调试级别。

-? | -h | --help

显示在线帮助。

### 示例

设置所有Segment上的max connections为100,而Master上为10。

```
gpconfig -c max_connections -v 100 -m 10
```

该示例展示bash shell字符串处理的语法要求。

```
gpconfig -c search_path -v "'\$user",public'
    gpconfig -c dynamic library path -v '\$libdir'
```

配置参数增加到postgresql.conf文件的格式如下。

```
search_path="'$user",public'
dynamic_library_path='$libdir'
```

注释掉default\_statistics\_target配置参数的所有实例,并恢复系统默认值:

gpconfig -r default\_statistics\_target

列出所有gpconfig支持的配置参数:

gpconfig -l

显示系统中一个特定配置参数的值:

gpconfig -s max\_connections

## 另见

gpstop

# gpdeletesystem

删除一个使用gpinitsystem初始化的MPP数据库系统gpinitsystem。

### 概要

gpdeletesystem [-d master\_data\_directory] [-B parallel\_processes]
 [-f] [-l logfile\_directory] [-D]

gpdeletesystem -?

gpdeletesystem -v

### 描述

gpdeletesystem工具执行以下两种操作:

- 停止所有postgres进程 (Segment实例和Master实例)。
- 删除所有的数据目录。

在运行gpdeletesystem之前:

- 将所有备份文件移出Master数据目录和Segment数据目录。
- 确保MPP数据库在运行。
- I如果用户当前位于Segment数据目录中,请将目录更改为另一个位置。从 Segment数据目录中运行时,该工具会失败,并显示错误。

该工具不会卸载MPP数据库软件。

### 选项

-d master data directory

指定Master主机数据目录。如果未指定此选项,则使用环境变量MASTER\_DATA\_DIRECTORY的设置。 如果指定了此选项,则会覆盖MASTER\_DATA\_DIRECTORY的任何设置。如果无法确定 master\_data\_directory,则该工具返回错误。

-B parallel\_processes

并行删除的Segment数。如果未指定,则该工具将根据需要删除多少个Segment实例启动 最多60个并行进程。

-f (force)

即使在数据目录中找到备份文件,也强制删除。如果备份文件存在,默认不删除MPP 数据库实例。

-l logfile\_directory

写入日志文件的目录。默认为~/gpAdminLogs。

-D (调试)

设置日志级别为debug。

-? (帮助)

显示在线帮助。

-v (显示工具版本)

显示该工具的版本、状态、上次更新的日期和校验和。

### 示例

删除一个MPP数据库系统:

gpdeletesystem -d /gpdata/gp-1

删除一个MPP数据库系统,即使备份文件存在:

gpdeletesystem -d /gpdata/gp-1 -f

### 另见

gpinitsystem

# gpexpand

在阵列中的新主机上扩展现有的MPP数据库。

### 概要

## 先决条件

• 用户作为MPP数据库超级用户 (gpadmin) 登录。

- 新的Segment主机已被根据现有的Segment主机安装和配置。这包括:
  - 。 配置硬件和操作系统
  - 。安装MPP软件
  - 。 创建gpadmin用户帐户
  - 。交换SSH密钥
- 用户的Segment主机上有足够的磁盘空间来临时保存最大表的副本。
- 重新分布数据时,MPP数据库必须以生产模式运行。MPP数据库不能是 受限模式或 Master模式。不能指定gpstart的选项-R或者-m 启动MPP数据 库。

Note: 当gpexpand正在执行segment初始化时,以下工具不能被运行。

- gpbackup
- gpcheckcat
- gpconfig
- gppkg
- gprestore

有关扩展MPP系统的更多信息,请见*MPP数据库管理员手册*的扩展MPP数据库系统。

### 描述

gpexpand工具分两个阶段执行系统扩展:Segment初始化和表重新分布。

在初始化阶段,gpexpand用一个输入文件运行,该文件指定新Segment的数据目录、 *dbid*值和其他特征。用户可以手动创建输入文件,也可以在交互式对话中 按照提示进行操作。

如果用户选择使用交互式对话创建输入文件,则可以选择指定包含扩展主

机列表的文件。在提示输入信息时,如果用户的平台或命令shell限制可键入的主机名列表的长度,则可能不得不用-f指定主机。

除了初始化Segment,初始化阶段还执行这些操作:

• 创建扩展模式gpexpand以存储扩展操作的状态,包括表的详细状态。

在表数据重分布阶段,gpexpand会重分布表的数据,使数据在新旧 segment 实例之间平衡。

#### Note:

- 1. 如果启动扩容追加写功能,则重分布之前需要将guc参数 gp\_autostats\_mode设置为none,防止在扩容期间触发analyze影响性能,扩容结束后需要将guc参数gp\_autostats\_mode再设置为default。
- 2. 数据重新分布应该在低峰 (相对高峰期) 时段进行。
- 3. 重新分布可以在很长时间内分批次进行。
- 4. 追加优化表的存储模型是为批量数据装载而优化。不推荐对追加优化表的单行 INSERT 语句。对于追加优化表,如果启动扩容追加写功能,则数据库在扩容数据重分布期间支持并发INSERT事务插入到一个追加优化表的最多并发数将少于127个。

gpexpand的数据重分布阶段支持IO资源限制,有两种资源限制模式:cgroup模式和throttle模式。cgroup模式支持限制磁盘IO和网络IO,throttle模式支持限制磁盘IO。

Note:cgroup只支持限制直接的磁盘IO,操作系统缓存和数据库buffer不受影响,throttle是对数据库读写buffer的速度做限制。 throttle的限制对象是实例级别,cgroup是操作系统级别。另外,cgroup支持动态调整资源并立即生效。

要开始重分布阶段,可以通过运行gpexpand并指定-d(运行时间周期)或-e(结束时间)选项,或者不指定任何选项。如果客户指定了结束时间或运行周期,工具会在扩展模式下重分布表,直到达到设定的结束时间或执行周期。如果没指定任何选项,工具会继续处理直到扩展模式的表全部完成重分布。每张表都会通过ALTER TABLE命令来在所有的节点包括新增加的

segment实例 上进行重分布,并设置表的分布策略为其原始策略。如果gpexpand完成所有表的重分布,它会 显示成功信息并退出。

在数据重分布以前,可以通过设置GUC参数来控制是否允许在数据重分布期间查询和导入数据。相关的GUC参数和函数说明如下:

- GUC参数gpexpand\_rw.enable:扩容数据重分布期间是否允许查询和导入数据。默认为off。如果设置为off,则不允许对扩容时正在重分布数据的表进行查询或者导入数据等操作;否则,扩容时允许对正在执行数据重分布的表进行查询(SELECT)操作,如果正在进行数据重分布的表是追加表(APPENDONLY=TRUE),则还支持导入(INSERT、COPY)数据到该表。NOTE:其中分区表不支持,同时不推荐在gpexpand\_rw.enable为on并且重分布数据时进行update/delete操作,由于数据库的封锁机制,这时候执行update/delete会阻塞之后的insert和copy操作。
- GUC参数gpexpand\_rw.enable\_deadlock\_process:扩容数据重分布期间死锁时取消哪个进程。默认为on。当gpexpand\_rw.enable为on时,重分布数据可能会导致死锁。如果gpexpand\_rw.enable\_deadlock\_process为on,则死锁时取消非扩容进程;否则可能会取消扩容进程。应该根据实际业务设置gpexpand\_rw.enable\_deadlock\_process参数,以保证想要保护的进程在数据重分布期间可以不被死锁检查机制意外取消。
- GUC参数gpexpand\_rw.enable\_optimizer:是否启用增量数据获取流程优化。默认为on。gpexpand\_rw.enable为on时该GUC参数生效。

Note: 如果是以升级的方式增加扩容期间允许查询和导入数据功能,则需要在升级数据库后修改所有实例数据目录下 postgresql.conf 文件中的 shared\_preload\_libraries 选项,在该选项最后添加 gpexpand\_rw 并重启数据库(在数据库启动的状态下可以使用gpconfig工具进行设置),最后还需要在数据库中执行 create extension gpexpand\_rw; 语句来添加 gpexpand\_rw扩展。例如:

\$ gpconfig -c shared\_preload\_libraries -v pg\_stat\_statements,passwordcheck,security\_eridentity\_auth,gpexpand\_rw

- \$ gpstop -r
- \$ psql -c 'show gpexpand\_rw.enable;' #此时可以看到GUC参数已经加载
- \$ pql -c 'create extension gpexpand\_rw'
- \$ psql -c '\dx' #此时可以看到gpexpand rw扩展

Note: 该工具在系统内部采用SSH连接执行各项操作任务。在大型MPP集群、云部署或每台主机部署了大量的 segment实例时,可能会遇到超过主机最大授权连接数限制的情况。此时需要考虑更新SSH配置参数MaxStartups 以提高该限制。更多关于SSH配置的选项,请参考您的Linux分发版的SSH文档。

## 选项

-a | --analyze

在扩展后运行ANALYZE更新表的统计信息,默认是不运行ANALYZE。

-B batch size

在暂停一秒钟之前发送给给定主机的远程命令的批量大小。默认值 是16,有效值是1-128。

gpexpand工具会发出许多设置命令,这些命令可能会超出主机的已验证 连接的最大阈值(由SSH守护进程配置中的MaxStartups定义)。该一秒钟 的暂停允许在gpexpand发出更多命令之前完成认证。默认值通常不需要改变。但是,如果gpexpand由于连接错误(例如'ssh\_exchange\_identification: Connection closed by remote host.') 而失败,则可能需要减少命令的最大数量。

-c | --clean

删除扩展模式。

- -d | --duration *hh:mm:ss* 扩展会话的持续时间。
- -e | --end 'YYYY-MM-DD hh:mm:ss'

扩展会话的结束日期及时间。

#### -f | --hosts-file filename

指定包含用于系统扩展的新主机列表的文件的名称。文件的每一行都必须包含一个主机名。

该文件可以包含指定或不指定网络接口的主机名。gpexpand工具处理 这两种情况,如果原始节点配置了多个网络接口,则将接口号添加到 主机名的末尾。

Note: MPP数据库Segment主机的命名习惯是sdwN,其中sdw是前缀并且N是数字。例如,sdw1、sdw2等等。 对于具有多个接口的主机,约定是在主机名后面添加破折号 (-) 和数字。例如sdw1-1和sdw1-2是主机sdw1的两个接口名称。

#### -i | --input input\_file

指定扩展配置文件的名称,其中为每个要添加的Segment包含一行, 格式为:

hostname:address:port:datadir:dbid:content:preferred\_role

-L

以cgroup资源限制模式运行数据重分布,使用之前请确保iolimit扩展已经加载,并且expand控制组的资源配额已经配置(使用gpcgroup工具初始化控制组和设置expand控制组的资源配额)。数据重分布的数据库后台进程将受到expand控制组的资源限制。

#### -n parallel\_processes

要同时重新分布的表的数量。有效值是1-96。

每个表重新分布过程都需要两个数据库连接:一个用于更改表,另一个用于在扩展方案中更新表的状态。 在增加-n之前,检查服务器配置参数max\_connections的当前值,并确保不超过最大连接限制。

### -r | --rollback

回滚失败的扩展设置操作。

#### -R max\_rate

以throttle资源限制模式运行数据重分布,使用之前请确保iolimit扩展已经加载。 max\_reate的单位是KB/s,范围从32 (32KB/s) 到 1048576 (1GB/s)。

### -s | --silent

以静默模式运行。在警告时,不提示确认就可继续。

#### -S | --simple-progress

如果指定,gpexpand工具仅在MPP数据库表 *gpexpand.expansion\_progress*中记录最少的进度信息。该工具不在表 *gpexpand.status\_detail*中记录关系大小信息和状态信息。指定此选项可通过减少写入*gpexpand*表的进度信息量来提高性能。

### [-t | --tardir] *directory*

Segment主机上一个目录的完全限定 *directory*,gpexpand 工具会在其中拷贝一个临时的tar文件。该文件包含用于创建 Segment实例的 MPP数据库文件。默认目录是用户主目录。

#### -v | --verbose

详细调试输出。使用此选项,该工具将输出用于扩展数据库的所有DDL和DML。

#### --version

显示工具的版本号并退出。

### -? | -h | --help

显示在线帮助。

### 示例

使用输入文件运行gpexpand以初始化新Segment,并在postgres数据库中创建扩展模式:

\$ gpexpand -i input\_file

运行gpexpand最长持续60小时,以将表重新分布给新的Segment:

\$ gpexpand -d 60:00:00

以cgroup资源限制模式运行gpexpand最长持续60小时,以将表重新分布给新的Segment,限制每个节点的重分布读速度为20MB/s、网络传输速度

#### 10MB/s, 1分钟后调整为读速度 10MB/s, 不限制网络传输速度:

- \$ gpconfig -c session preload libraries -v iolimit
- \$ gpstop -u
- \$ gpcgroup set -f all hosts -G expand -R 20 -N 10
- \$ sleep 60 && gpcgroup set -f all hosts -G expand -R 10 -N 0
- \$ gpexpand -d 60:00:00 -L

以throttle资源限制模式运行gpexpand最长持续60小时,以将表重新分布给新的Segment,限制每个segment的重分布速度为4MB/s:

- \$ gpconfig -c session\_preload\_libraries -v iolimit
- \$ gpstop -u
- \$ gpexpand -d 60:00:00 -R 4096

扩容数据重分布阶段开始前修改gpexpand\_rw.enable参数以支持数据重分布期间读及AO表追加写:

- \$ gpconfig -c gpexpand rw.enable -v on
- \$ gpconfig -c gp\_autostats\_mode -v none #设置guc参数防止扩容期间触发analyze
- \$ gpstop -u
- \$ gpexpand -d 01:00:00 #此时,可以对正在进行数据重分布的表进行读和对 AO表进行追力
- \$ gpconfig -c gp\_autostats\_mode -v default #扩容结束后需要将参数设置回来
- \$ gpstop -u

### 另见

MPP数据库管理员手册中的gpssh-exkeys扩展MPP系统。

# gpfdist

将数据文件载入MPP数据库Segment或从其中写出数据文件到文件系统。

### 概要

```
gpfdist [-d directory] [-p http_port] [-P last_http_port] [-l log_file]
    [-t timeout] [-S] [-w time] [-v | -V] [-s] [-m max_length]
    [--ssl certificate_path [--sslclean wait_time] ]
    [-c config.yml]

gpfdist -? | --help

gpfdist --version
```

### 描述

gpfdist是MPP数据库并行文件分发程序。它可以被外部表和gpload 用来并行地将外部表文件提供给所有的MPP数据库Segment。它由可写外部表使用,并行接受来自MPP数据库 Segment的输出流,并将它们写出到文件中。

Note: gpfdist和gpload是在MPP的主版本级别有效的。 例如,MPP 4.x版本的gpfdist不能用于MPP 5.x或6.x版本。

为了使外部表使用gpfdist,外部表定义的LOCATION子句必须使用gpfdist://协议(参见MPP数据库命令CREATE EXTERNAL TABLE)。

Note: 如果--ssl选项被指定来启用SSL安全性,请使用gpfdists://协议创建外部表。

使用gpfdist的好处是在读取或写入外部表时可以保证最大的并行性,从而提供最佳的性能,并且更容易管理外部表。

对于只读外部表,当用户在外部表中SELECT时,gpfdist 将数据文件均匀地分析并提供给MPP数据库系统的所有Segment实例。对于可写的外部表,gpfdist在用户INSERT外部表时接受来自Segment的并行输出流,并写入输出文件。

对于可读外部表,如果被加载的文件使用gzip或bzip2 (具有.gz或.bz2的文件扩展名),gpfdist 会在装载之前自动解压文件,前提是gunzip或bunzip2在用户的可执行文件路径中。

Note: 目前,可读外部表不支持在Windows平台上的压缩,可写外部表不支持任何平台上的压缩。

当使用gpfdist或gpfdists协议读写数据时,MPP数据库在HTTP请求头部中包含X-GP-PROTO,以指示该请求来自MPP数据库。该工具拒绝请求头部中不包含X-GP-PROTO的HTTP请求。

大多数情况下,用户很可能希望在ETL机器而不是安装MPP数据库的主机上运行gpfdist。要在其他主机上安装gpfdist,只需简单的将该程序复制到该主机上,然后将gpfdist添加到用户的\$PATH路径中。

Note: 使用IPv6时,请始终将数字IP地址包裹在括号内。

### 选项

### -d directory

指定一个目录,gpfdist将从该目录中为可读外部表提供文件,或为可写外部表创建输出文件。如果没有指定,默认为当前目录。

#### -l log\_file

要记录标准输出消息的完全限定路径和日志文件名称。

#### -p http\_port

gpfdist提供文件要使用的HTTP端口。默认为8080。

### -P last\_http\_port

gpfdist将会提供文件服务的端口号范围 (http\_port到last\_http\_port包含岂止号码) 中最后一个HTTP端口号号码。gpfdist会 以端口号设定范围内第一个成功绑定的端口号作为服务端口。

#### -t timeout

设置MPP数据库建立与gpfdist进程的连接所允许的时间。默认值是5秒。 允许的值是2到7200秒(2小时)。在网络流量大的系统上可能要增加。

#### -m max\_length

设置以字节为单位的最大数据行长度。默认值是32768。当用户数据包含非常宽的行时(或者当line too long错误消息发生时)应该使用,否则不应该使用,因为它会增加资源分配。有效范围是32K到256MB(Windows系统上限为1MB)。

Note: 如果用户指定较大的最大行长度并运行大量的gpfdist并发连接,则可能会发生内存问题。例如,使用96个并行gpfdist进程需要大约 24GB的内存((96 + 1) x 246MB)。

-S

启用简化的日志记录。指定此选项时,只有具有WARN级别或者更高级别的消息才会写入gpfdist日志文件。INFO级别的消息不写入日志文件。 如果未指定这一选项,则所有gpfdist消息都写入日志文件。

用户可以指定此选项以减少写入日志文件的信息。

#### -S (use O SYNC)

使用O\_SYNC标志打开同步I/O的文件。任何对结果文件描述的写都会阻塞gpfdist,直到数据被物理地写到底层硬件。

#### -w time

设置关闭目标文件(如命名管道)之前MPP数据库延迟的秒数。默认值是0,没有延迟。最大值是7200秒(2小时)。

对于具有多个Segment的MPP数据库,在将不同Segment中的数据写入文件时,Segment之间可能会有延迟。 用户可以指定MPP数据库关闭文件之前要等待的时间,以确保所有数据都被写入文件。

#### --ssl certificate\_path

将SSL加密添加到使用gpfdist传输的数据。使用--ssl certificate\_path选项执行gpfdist之后,从此文件服务器加载数据的唯一方法是使用gpfdist://协议。有关gpfdist://协议的信息,请参阅*MPP*数据库管理员手册中的"装载和卸载数据"部分。

在 certificate\_path中指定的位置必须包含以下文件:

- 服务器证书文件, server.crt
- 服务器私钥文件, server.key
- 可信证书机构, root.crt

根目录 (/) 不能指定为 certificate\_path。

#### --sslclean wait time

使用--ssl选项运行该工具时,设置该工具完成向MPP数据库Segment 读写数据 后关闭SSL会话和清除SSL资源之前要延迟的秒数。默认值是0,没有延迟。最大值是500秒。如果延迟增加,传输速度降低。在某些情况下,复制大量数据时可能会发生此错误:gpfdist server closed connection。为了避免这类错误,用户可以增加一个延迟,例如--sslclean 5。

### -c config.yaml

指定gpfdist在装载或抽取数据时,用来选择要应用的转换的规则。gpfdist配置文件是一个YAML1.1文档。

有关文件格式的信息,请参阅*MPP数据库管理员手册*中的配置文件格式配置件格式。 有关用gpfdist配置数据转换的信息,请参阅*MPP数据库管理员手册*中的使用gpfdist和gpload转换外部数据转换XML数据。该选项在windows平台中不可用。

### -v (详细模式)

显示进度和状态信息的详细模式。

### -V (非常详细模式)

显示由该工具生成的所有输出信息的详细模式。

### -? (帮助)

显示在线帮助。

#### --version

显示该工具的版本。

### 注解

服务器配置文件verify\_gpfdists\_cert用来控制MPP数据库与gpfdist工具沟通进行读写数据时是否启用SSL授权。当在进行MPP数据库外部表与gpfdist工具测试时,客户可以设置该参数值为false来禁用授权。如果参数设置为false,以下SSL异常会被忽略:

- gpfdist使用的自有签名SSL证书授权不被MPP信任。
- SSL证书中包含的主机名与gpfdist主机名不匹配。

Warning: 禁用SSL证书授权会暴露安全风险,因为gpfdist SSL不会再进行验证。

如果gpfdist工具在读写操作中hang住,客户可以在下次hang住时生成一个core dump文件来帮助调试问题原因。在gpfdist强制退出之前,设置环境变量GPFDIST\_WATCHDOG\_TIMER为异常时间段的秒数。 当客户设置了该环境变量,并且gpfdist hang住时,工具会在特定秒数后停止并创建 core dump文件,发送相关信息到日志文件。

例子中设置的Linux系统环境变量为gpfdist在无活动操作后300秒(5分钟)退出。

export GPFDIST WATCHDOG TIMER=300

# 示例

使用端口8081从指定目录提供文件(并在后台启动gpfdist):

gpfdist -d /var/load\_files -p 8081 &

在后台启动gpfdist并将输出和错误重定向到日志文件:

gpfdist -d /var/load\_files -p 8081 -l /home/gpadmin/log &

停止在后台运行的gpfdist:

--首先找到它的进程ID:

ps ax | grep gpfdist

--然后杀死该进程,例如:

kill 3456

### 另见

gpload、 MPP数据库管理员手册中的CREATE EXTERNAL TABLE

# gpinitstandby

为MPP数据库系统添加或初始化后备Master主机。

## 概要

gpinitstandby -v

gpinitstandby -?

## 描述

gpinitstandby工具为用户的MPP数据库系统添加一个备份的后备Master主机。如果用户的系统已经配置了后备Master主机,请在添加新的后备Master主机之前使用-r 选项将其删除。

在运行此工具之前,请确保MPP数据库软件已安装在后备Master主机上,并且用户已在主机之间交换了 SSH密钥。推荐在Master主机和后备Master主机上将主端口设置为相同的端口号。

此工具应该在当前活动的*主*Master主机上运行。方法请参见*MPP数据库安 装手册*。

### 该工具执行以下的步骤:

- 更新该MPP数据库系统目录来删除现有的后备Master主机信息 (如果指定了-r)
- 更新该MPP数据库系统目录来添加新的后备Master主机信息

- 编辑MPP数据库Master的pg\_hba.conf来允许从新添加的后备Master主机进行访问。
- 在后补Master主机上设置后备Master实例。
- 开始同步进程。

如果主Master主机变为不可操作,后备Master主机将作为"热备用"。后备Master通过事务日志 复制进程(walsender和walreceiver)保持最新 状态,这些进程在主Master主机和后备Master主机上运行,并使两者的数据保持同步。如果主 Master失效,日志复制过程将关闭,后备Master可以使用gpactivatestandby 工具在其位置激活。一旦后备Master激活,复制日志将用于重建最后一次成功提交事务时Master 主机的状态。

激活的后备Master实际上成为MPP数据库的Master,在主端口上接受客户端连接并执行正常的 Master操作,例如SQL命令处理和负载管理。

Important: 如果gpinitstandby工具之前初始化后备Master失败,则必须显示删除后备Master数据目录中的文件,然后再次运行gpinitstandby。 后备Master数据目录在初始化失败后不会被清理,因为它包含可帮助确定失败原因的日志文件。

如果发生初始化失败,则会在后备主机目录/tmp中生成摘要报告文件。 该报告列出了后备主机上需要清理的目录。

## 选项

- -a (不提示)
  - 不要提示用户确认。
- -D (调试)
  - 设置日志级别为debug。
- -l logfile\_directory
  - 写入日志的目录,默认为~/gpAdminLogs。
- -n (重启被主机)

指定此选项以启动已配置但由于某种原因已停止的MPP数据库的后备 Master。

#### -P port

此选项指定MPP数据库的后备Master使用的端口。默认值是MPP数据库的活动 Master使用的端口。

如果MPP数据库的后备Master和活动Master在同一台主机上,则端口必须不同。 如果活动Master和后备Master的端口相同,并且所在的主机也相同,则该工具将返回错误。

-q (无屏幕输出)

以静默模式运行。命令输出不显示在屏幕上,但是仍然写入日志文 件。

-r (移除后备Master)

从MPP数据库系统中删除当前已配置的后备Master。

-s standby\_hostname

后备Master的主机名。

-S standby\_data\_directory

新备用Master主机使用的数据目录。默认为和活动Master相同的路径。

如果备用Master和活动Master在同一台主机上,必须使用该选项定义 一个不同的路径。

-v (显示工具版本)

显示该工具的版本、状态、上次更新的日期和校验和。

-? (帮助)

显示在线帮助。

### 示例

添加一个后备Master到用户的MPP数据库系统,并重启同步进程:

gpinitstandby -s host09

开启一个存在的后备Master主机,并且和当前的主Master主机同步:

gpinitstandby -n

Note: 不要在同一命令指定-n和-s选项。

指定一个不同的端口向MPP数据库系统添加一台后备Master主机:

gpinitstandby -s myhost -P 2222

如果指定的主机名与MPP数据库的活动Master相同,则用作后备Master的MPP数据库软件必须用-P参数指定与MPP数据库的活动Master的软件位于不同位置。 另外,后备Master使用的文件空间位置必须与 MPP数据库的活动Master不同。

从MPP系统配置中删除现有的后备Master:

gpinitstandby -r

### 另见

gpinitsystem, gpaddmirrors, gpactivatestandby

# gpinitsystem

使用gpinitsystem\_config文件中指定的配置参数初始化一个MPP数据库系统。

### 概要

```
gpinitsystem -c cluster_configuration_file
        [-h hostfile_gpinitsystem]
       [-B parallel_processes]
        [-p postgresql_conf_param_file]
        [-s standby_master_host
          [-P standby_master_port]
          [-S standby_master_datadir | --standby_datadir=standby_master_datadir]]
        [-m number | --max_connections=number]
        [-b size | --shared buffers=size]
        [-n locale | --locale=locale] [--lc-collate=locale]
        [--lc-ctype=locale] [--lc-messages=locale]
        [--lc-monetary=locale] [--lc-numeric=locale]
        [--lc-time=locale] [-e password | --su_password=password]
        [--mirror-mode={group|spread}] [-a] [-q] [-l logfile_directory] [-D]
        [-l input_configuration_file]
        [-O output_configuration_file]
gpinitsystem -v | --version
gpinitsystem -? | --help
```

### 描述

gpinitsystem工具将使用配置文件中定义的值和客户提供的命令行选项 创建一个MPP数据库实例。 有关配置文件的更多信息,请参见初始配置文件格式。在 运行此工具之前,请确保已经在阵列中所有主机上安装了MPP数据库软件。

带有-O output\_configuration\_file选项时,gpinitsystem不会创建一个新数据库,而是把所有的配置都写入一个配置文件。该文件使用QD\_PRIMARY\_ARRAY和 PRIMARY\_ARRAY参数来定义每个成员相关的主机名、端口号、数据目录、Segment前缀、Segment ID和Content ID。集群配置的详细信息可以根据需要修改,以匹配MPP数据库备份中的可用值,活着可以简单的被用来重建相同集群配置。使用QD\_PRIMARY\_ARRAY和 PRIMARY\_ARRAY的配置文件必须被传递给gpinitsystem -I input\_configuration\_file。详细信息请见初始配置文件格式。

在MPP数据库DBMS中,必须在系统中的所有主机上初始化每个数据库实例 (Master和所有的Segment) ,以便它们可以作为统一的DBMS一起使用。gpinitsystem工具负责初始化MPP的 Master和每个Segment实例,并作为一个整体配置系统。

在运行gpinitsystem之前,用户必须设置\$GPHOME环境变量以 指向 Master 上的MPP数据库安装位置,并且使用gpssh-exkeys在阵列中的 所有主机地 址之间交换SSH密钥。

### 这个工具执行以下任务:

- 验证配置文件中参数的正确。
- 确保可以建立到每个主机地址的连接。如果主机地址无法到达,该工具 将退出。
- 验证区域设置。
- 显示将要使用的配置并提示用户进行确认。
- 初始化Master实例。
- 初始化后备Master实例(如果指定)。
- 初始化主Segment实例。
- 初始化镜像Segment实例(如果配置)。
- 配置MPP数据库系统并检查错误。

• 启动MPP数据库系统。

Note: 该工具在系统内部采用SSH连接执行各项操作任务。在大型MPP集群、云部署或每台主机部署了大量的 segment实例时,可能会遇到超过主机最大授权连接数限制的情况。此时需要考虑更新SSH配置参数MaxStartups 以提高该限制。更多关于SSH配置的选项,请参考您的Linux分发版的SSH文档。

Note: 凝思系统默认只存在zh\_CN.UTF-8编码,不存在en\_US.UTF-8编码。 因此初始化的时候需要做以下两件事之一,否则集群初始化会失败:

- 1、gpinitsystem指定数据库编码为zh CN.UTF-8;
- 2、在/etc/locale.gen文件中添加en\_US.UTF-8 UTF-8编码,然后用 sudo权限执行local-gen命令添加en\_US编码。

# 选项

-a

不提示用户进行确认。

-B parallel\_processes

要并行创建的Segment数。如果未指定,该工具一次最多启动4个并行进程。

-c cluster\_configuration\_file

必需。配置文件的完整路径和文件名称,其中包括所有已经定义的参数,用于配置和初始化新的 MPP数据库系统。有关此文件的说明,请参见初始配置文件格式初始化配置文件格式。gpinitsystem必须指定 - c cluster\_configuration\_file选项或 - l input\_configuration\_file选项。

-D

设置日志输出等级为debug。

-h hostfile\_gpinitsystem

可选。包含Segment主机地址的文件的完整路径和文件名。如果未在命令行中指定,则可以使用 *gpinitsystem\_config*文件中的MACHINE\_LIST\_FILE 参数指定主机文件。

## -l input\_configuration\_file

配置文件的全路径及文件名,该文件使用QD\_PRIMARY\_ARRAY和PRIMARY\_ARRAY参数定义MPP数据库成员和Segment实例。该配置文件通常用gpinitsystem -O output\_configuration\_file生成。gpinitsystem必须指定 -c cluster\_configuration\_file选项或 -I input\_configuration\_file 选项。

## -n locale | --locale=locale

设置MPP数据库使用的默认区域。如果未指定,则Master主机的LC\_ALL、LC\_COLLATE或 LANG环境变量决定区域。如果这些没有设置,则默认的区域是C(POSIX)。区域标识符由语言标识符、地区标识符和可选的字符集编码组成。例如,sv\_SE是瑞典语,en\_US是美国英语,fr\_CA是加拿大法语。如果不止一个字符集可以用于一个区域,则规范如下所示:en\_US.UTF-8(区域规范和字符集编码)。在大多数系统中,命令locale 将显示区域环境设置,locale -a将会显示所有可用的区域的列表。

### --lc-collate=locale

类似于--locale,但是设置用于排序规则(排序数据)的区域。 MPP数据库初始化后无法更改排序顺序,因此有必要选择与用户计划用于数据的 字符集编码兼容的排序规则区域。 C或POSIX 有一个特殊的排序规则名称(字节顺序排序而不是字典顺序排序)。 C 排序规则可以用于任何字符编码。

# --lc-ctype=locale

类似于--locale,但设置用于字符分类的语言环境 (哪些字符序列是有效的,以及它们如何被解释)。在MPP数据库初始化之后,这是不能更改的,因此有必要选择一个与用户计划存储在MPP数据库中的数据兼容的字符分类区域。

## --lc-messages=locale

类似于--locale,但设置用于MPP数据库输出消息的语言环境。 当前版本的MPP数据库不支持输出消息的多种区域(所有消息均为英文),所以更改此 设置不会有任何效果。

# --lc-monetary=locale

类似--locale,但是设置用于格式化货币金额的区域。

## --lc-numeric=*locale*

类似--locale, 但是设置用于格式化数字的区域。

### --lc-time=locale

类似--locale,但设置用于格式化日期和时间的区域。

## -l logfile\_directory

写入日志文件的目录,默认为~/gpAdminLogs。

-m number | --max\_connections=number

设置Master允许的最大客户端连接数。默认值是250。

## -O output configuration file

当使用-O选项时,gpinitsystem不会创建一个新的MPP集群,而是会将提供的集群配置信息写入指定的 When used with the option, does not create a new MPP Database cluster but instead writes the supplied cluster configuration information to the specified output\_configuration\_file文件。该文件使

用QD\_PRIMARY\_ARRAY和 PRIMARY\_ARRAY参数来定义每个成员相关的主机名、端口号、数据目录、Segment前缀、 Segment ID和 Content ID。集群配置的详细信息可以根据需要修改,以匹配MPP数据库备份中的可用值,活着可以简单的被用来重建相同集群配置。使用QD\_PRIMARY\_ARRAY、

PRIMARY\_ARRAY和MIRROR\_ARRAY的配置文件必须被传递给gpinitsystem -I *input\_configuration\_file*以初始化集群。

# -p postgresql\_conf\_param\_file

可选。包含用户想要为MPP数据库设置的postgresql.conf参数 设置的文件名称。这些设置将在初始化单个Master和Segment实例时使用。用户也可以在 初始化后使用gpconfig工具来设置参数。

-q 以静默模式运行。命令行输出不显示在屏幕上,但仍然写入日志文 件。

# -b size | --shared\_buffers=size

设置MPP服务器实例用于共享内存缓冲区的内存量。用户可以指定以 千字节(KB)、兆字节(MB)或千兆字节(GB)为单位的大小。默 认值是125MB。

# -s standby\_master\_host

可选。如果用户希望配置备份Master主机,请使用此选项指定主机名

称。 MPP数据库 软件必须已经在该主机上安装和配置。

# -P standby\_master\_port

如果使用-s配置了备用Master实例,可以用该选项定义端口号。 默认端口号与Master端口号相同。为了在同一主机上运行备用和主Master,必须使用该选项 指定一个不同的端口号。MPP数据库软件必须已经在备用主机上安装和配置好。

- -S standby\_master\_datadir | --standby\_dir=standby\_master\_datadir 如果使用-s配置备用Master主机,可以使用该选项指定数据目录。 如果在同一台主机上配置备用Master和主Master实例,那必须定义该选项以指定不同的目录。

## 推荐的最佳安全实践:

- 不要生产环境中使用默认密码选项。
- 安装之后立即更改密码。

# --mirror\_mode={group|spread}

使用该选项指定镜像Segment实例所在的镜像主机。默认采用group模式,会将一台主机上的所有主Segment实例的镜像放到另外一台主机。 spread会将一台主机上的所有实例散布在另外的主机上。 spread仅在集群主机数量大于每台主机上的实例数量时可用。 有关MPP数据库镜像策略的详细信息,请见Segment镜像概述。

# -v | --version

显示gpinitsystem的版本。

# -? | --help

显示gpinitsystem命令行参数,然后退出。

# 初始配置文件格式

gpinitsystem需要定义有以下参数的配置文件。示例初始化配置文件可以在\$GPHOME/docs/cli\_help/gpconfigs/gpinitsystem\_config中找到。

为了避免MPP数据库和其他应用程序之间的端口冲突,MPP数据库端口号不应该在操作系统参数net.ipv4.ip\_local\_port\_range指定的范围之内。例如,如果 net.ipv4.ip\_local\_port\_range = 10000 65535,则可以将MPP数据库基本端口号设置为以下这些值。

PORT\_BASE = 6000 MIRROR PORT BASE = 7000

### **ARRAY NAME**

必需。用户正在配置的阵列的名称。用户可以使用任何用户喜欢的名字。 如果名称包含空格,请将名称放在引号中。

# MACHINE\_LIST\_FILE

可选。可以用来替代-h选项。这指定包含构成MPP数据库系统的 Segment主机地址名称列表的文件。Master主机被假定为运行该工具的主机,并且不应该被包含在此文件中。如果用户的Segment主机有多个网络接口,则该文件将包含该主机的所有地址。给出该文件的绝对路径。

# **SEG\_PREFIX**

必需。这指定了一个前缀,用于命名Master和Segment实例上的数据目录。 MPP数据库系统中数据目录的命名约定

是 SEG\_PREFIXnumber, 其中 number对 Segment实例从0开始 (Master始终为-1)。因此,如果用户选择前缀gpseg,则用户的 Master实例数据目录将会被命名为 gpseg-1,并且 Segment实例将被命名为gpseg0、 gpseg1、gpseg2、gpseg3等等。

# PORT\_BASE

必需。这指定计算主Segment端口用到的基础数字。主机上的第一个主Segment端口被设置为 PORT\_BASE, 然后对该主机上的每个额外主Segment都加一,有效值为1-65535。

# DATA\_DIRECTORY

必需。这指定工具将创建主Segment数据目录的数据存储位置。列表

中的位置数量决定了每台物理主机将创建的主Segment的数量(如果主机文件中列出了主机的多个地址,Segment的数量 将被均匀分布在指定的接口地址间)。如果用户希望在同一位置创建数据目录,则可以多次列出相同的数据存储区域。运行gpinitsystem的用户(例如,gpadmin用户)必须具有写入这些目录的权限。例如,这将为每台主机创建六个主Segment:

declare -a DATA\_DIRECTORY=(/data1/primary /data1/primary /data1/primary /data2/primary /data2/primary)

## MASTER HOSTNAME

必需。 Master实例的主机名。这个主机名必须与机器上配置的主机名完全匹配 (运行hostname命令以确定正确的主机名)。

# MASTER\_DIRECTORY

必需。这指定在Master主机上创建数据目录的位置。用户必须确保运行gpinitsystem的用户(例如gpadmin用户)有权写入此目录。

## **MASTER PORT**

必需。Master实例的端口号。这是访问MPP数据库系统时用户和客户端连接将使用的端口号。

# TRUSTED\_SHELL

必需。gpinitsystem工具用来在远程主机上执行命令的shell。 允许的值为ssh。用户必须在运行gpinitsystem工具之前设置用户的可信主机环境(用户可以使用gpssh-exkeys来做这件事)。

# **CHECK POINT SEGMENTS**

必需。日志文件段(每个段通常为16兆字节)中自动预写式日志 (WAL) 检查点之间的最大距离。 这将在MPP数据库系统的每个 Segment实例的postgresql.conf文件中 设置checkpoint\_segments参数。

### **ENCODING**

必需。要使用的字符集编码。该字符集必须与所使用的--locale设置兼容,尤其是--lc-collate和--lc-ctype。MPP数据库 支持和PostgreSQL相同的字符集。

## DATABASE NAME

可选。系统初始化之后要创建的MPP数据库的名称。用户可以随后使用 CREATE DATABASE命令或createdb工具创建数据库。

## MIRROR PORT BASE

可选。这指定计算镜像Segment端口号用到的基数。主机上的第一个镜像Segment端口设置为 MIRROR\_PORT\_BASE, 然后对该主机上每个额外的镜像Segment加一。有效值范围 从1到65535, 不能与PORT\_BASE计算的端口冲突。

## MIRROR DATA DIRECTORY

可选。指定工具将创建镜像Segment数据目录的数据存储位置。必须为镜像Segment 实例声明与主Segment实例相同数量的数据目录(请参阅DATA\_DIRECTORY参数)。运行gpinitsystem的用户(例如,gpadmin用户)必须具有写入这些目录的权限。例如:

declare -a MIRROR\_DATA\_DIRECTORY=(/data1/mirror /data1/mirror /data2/mirror /data2/mirror /data2/mirror)

QD\_PRIMARY\_ARRAY, PRIMARY\_ARRAY, MIRROR\_ARRAY

这些参数只能通过配置文件提供给gpinitsystem -l input\_configuration\_file。

QD\_PRIMARY\_ARRAY、PRIMARY\_ARRAY和 MIRROR\_ARRAY定义MPP数据库Master主机、主实例和 镜像实例,格式如下:

host~port~data\_directory/seg\_prefix<segment\_id>~dbid~content\_id

MPP数据库Master会一直使用-1作为Segment ID和Content ID。例如:

客户可以使用gpinitsystem -O output\_configuration\_file来生成QD\_PRIMARY\_ARRAY、PRIMARY\_ARRAY、MIRROR\_ARRAY参数,包含主机、数据目录、Segment前缀和端口号信息。出于恢复的目的,您可以编辑Segment和Content ID的值以匹配现存MPP 数据库备份的值。

## **HEAP CHECKSUM**

可选。该参数指定堆表的数据是否启用checksum。当启用时,所有数据库上的堆存储进行checksum,MPP数据库可以预防I/O系统损坏导致数据损坏。该选项只有在系统初始化时进行设置并且以后不能改变。

HEAP\_CHECKSUM选项默认启用,强烈不建议将该选项设置为禁用。 如果禁用该选项,存储的数据损坏不会被检测到,数据恢复会相当困难。

要检查是否MPP数据库启用了堆表checksum,可以通过gpconfig 管理工具查询data\_checksums参数:

\$ gpconfig -s data\_checksums

# HBA\_HOSTNAMES

可选。该参数控制gpinitsystem工具在pg\_hba.conf 中开放访问IP地址或者主机名的信息。默认值为0,该工具会 使用IP地址更新hba文件。当初始化MPP数据库系统时,指定 HBA\_HOSTNAMES=1来让工具使用主机名更新 pg\_hba.conf文件。

更多有关MPP数据库解决pg\_hba.conf文件中主机名的信息,请见配

## 置客户端认证。

# 示例

通过提供配置文件和Segment主机地址文件来初始化MPP数据库阵列,并设置一个散布镜像 (--mirror-mode=spread) 配置:

\$ gpinitsystem -c gpinitsystem\_config -h hostfile\_gpinitsystem --mirror-mode=spread

初始化MPP数据库阵列并设置超级用户远程口令:

\$ gpinitsystem -c gpinitsystem\_config -h hostfile\_gpinitsystem --su-password=mypassword

初始化带有可选后备Master主机的MPP数据库阵列:

\$ gpinitsystem -c gpinitsystem\_config -h hostfile\_gpinitsystem -s host09

代替初始化一个MPP数据库集群,将提供的配置文件信息写入到输出文件。该输出文件采用QD\_PRIMARY\_ARRAY和 PRIMARY\_ARRAY参数定义Master和Segment主机:

\$ gpinitsystem -c gpinitsystem\_config -h hostfile\_gpinitsystem --mirror-mode=spread -O

采用输入带有QD\_PRIMARY\_ARRAY和PRIMARY\_ARRAY参数的配置文件(该文件定义MPP数据库集群)初始化MPP数据库:

\$ gpinitsystem -I cluster\_init.config

# 另见

gpssh-exkeys, gpdeletesystem

# gpload

按照一个YAML格式的控制文件的定义运行一个装载作业。

# 概要

```
gpload -f control_file [-l log_file] [-h hostname] [-p port]
  [-U username] [-d database] [-W] [--gpfdist_timeout seconds]
  [--no_auto_trans] [[-v | -V] [-q]] [-D]
```

gpload -?

gpload --version

# 先决条件

要执行gpload命令的客户机必须具有下列要求:

Python 2.6.2或更新版本,装有pygresql (Python的PostgreSQL接口包),和pyyaml。注意Python及所需的Python库被包含在MPP数据库安装

包中,因此如果在gpload运行的机器上安装有MPP数据库,用户就不需要单独安装Python。

Note: MPP数据库的Windows装载客户端仅支持Python 2.5(您可以从https://www.python.org获取)

0

- gpfdist并行文件分发程序被安装在\$PATH中。这个程序位于MPP数据库的\$GPHOME/bin目录下。
- gpload客户机可以访问(被访问) MPP数据库集群 (Master和 Segments) 中所有主机。
- gpload客户及可以访问(被访问)所有可能用来装载数据的主机(ETL 服务器)。

# 描述

gpload是一个数据装载工具,它扮演着MPP数据库外部表并行装载 特性的接口的角色。通过一个用YAML格式控制文件定义的装载说明,gpload 调用MPP数据库的并行文件服务器 (gpfdist)执行 文件装载,基于源数据的定义创建一个外部表定义,并且指定INSERT、 UPDATE或MERGE操作把源数据装载到数据库中的目标表中。

Note: gpfdist和gpload是在MPP的主版本级别有效的。 例如,MPP 4.x版本的gpfdist不能用于MPP 5.x或6.x版本。

Note: 如果目标表的列名为保留关键字、有大写字母或包含任何双引号,那么MERGE和UPDATE操作不被支持。

在目标表上指定多个同时的装载操作时,操作包括在YAML控制文件(控制文件格式见控制文件格式)的SQL集合中指定的任何SQL命令会在单个事务中执行以防止数据不一致。

# 选项

### -f control file

必选项。包含装载说明详情的YAML文件。请见控制文件格式。

## --gpfdist\_timeout seconds

为gpfdist并行文件分发程序发送响应设置超时时间。 输入一个 从0到30秒的值 (输入"0"会禁用超时)。 注意在高流量网络上可能需要增加这个值。

## -l log\_file

指定在哪里写日志文件。默认

是~/gpAdminLogs/gpload\_*YYYYMMDD*。 有关日志文件的更多信息请见日志文件格式。

## --no auto trans

如果用户在目标表上执行单个装载操作,可指定--no\_auto\_trans 禁用把装载操作作为单个事务处理的特性。

默认情况下,在一个目标表上执行多个同时的操作时,gpload把每个 装载操作处理为 单个事务以防止不一致的数据。

## -q (无屏幕输出)

运行在静默模式中。命令输出不会被显示在屏幕上,但仍将被写入到日志文件。

# -D (调试模式)

检查错误情况,但是不执行装载。

# -v (详细模式)

在装载步骤被执行时,显示它们的详细输出。

# -V (非常详细模式)

显示非常详细的输出。

# -? (显示帮助)

显示帮助,然后退出。

#### --version

显示这个工具的版本,然后退出。

# 连接选项

### -d database

要装载到的数据库。如果没有指定,则从装载控制文件、环境变量\$PGDATABASE读取或者默认为当前系统用户名。

#### -h hostname

指定MPP的Master数据库服务器在其上运行的机器的主机名。如果没有指定,会从装载控制文件、环境变量\$PGHOST读取或者默认为localhost。

### -p port

指定MPP的Master数据库服务器在其上监听连接的TCP端口。如果没有指定,会从装载控制文件、环境变量\$PGPORT读取或者默认为5432。

### -U username

要用其进行连接的数据库角色名。如果没有指定,会从装载控制文件、环境变量\$PGUSER读取或者默认为当前系统用户名。

## -W (强制口令提示)

强制口令提示。如果没有指定,会从环境变量\$PGPASSWORD、 \$PGPASSFILE指定的口令文件或~/.pgpass 中的口令文件中读取口 令。如果这些都没有设置,即使没有提供-W,gpload也将提示要求一 个口令。

# 控制文件格式

gpload控制文件使用 YAML 1.1 文档格式,然后为定义MPP数据库装载操作的多个步骤实现了其自身的模式。 该控制文件必须是一个有效的YAML文档。

gpload程序按顺序处理控制文件文档并且使用缩进(空格)来判断文档层次以及各个部分之间的关系。空格的使用是有意义的。不能把空格简单地用于格式化目的,并且不能使用制表符。

## 一个装载控制文件的基础结构是:

**VERSION: 1.0.0.1** 

DATABASE: db\_name

```
USEN: UD_USemanne
HOST: master_hostname
PORT: master_port
GPLOAD:
 INPUT:
  - SOURCE:
     LOCAL HOSTNAME:
      - hostname or ip
     PORT: http_port
    | PORT_RANGE: [start_port_range, end_port_range]
     FILE:
      - /path/to/input_file
     SSL: true | false
     CERTIFICATES PATH: /path/to/certificates
  - FULLY QUALIFIED DOMAIN NAME: true | false
  - COLUMNS:
      - field name: data type
  - TRANSFORM: 'transformation'
  - TRANSFORM CONFIG: 'configuration-file-path'
  - MAX LINE LENGTH: integer
  - FORMAT: text | csv
  - DELIMITER: 'delimiter character'
  - ESCAPE: 'escape_character' | 'OFF'
  - NULL AS: 'null_string'
  - FORCE NOT NULL: true | false
  - QUOTE: 'csv_quote_character'
  - HEADER: true | false
  - ENCODING: database_encoding
  - ERROR LIMIT: integer
  - LOG ERRORS: true | false
 EXTERNAL:
   - SCHEMA: schema | '%'
 OUTPUT:
  - TABLE: schema.table name
  - MODE: insert | update | merge
  - MATCH COLUMNS:
      - target column name
  - UPDATE COLUMNS:
      - target column name
  - UPDATE CONDITION: 'boolean_condition'
  - MAPPING:
        target_column_name: source_column_name | 'expression'
 PRELOAD:
```

- TRUNCATE: true | false
- REUSE TABLES: true | false
- STAGING\_TABLE: external\_table\_name
- FAST MATCH: true | false

#### SQL:

BEFORE: "sql\_command"AFTER: "sql\_command"

### **VERSION**

可选。gpload控制文件模式的版本。当前版本是1.0.0.1。

### **DATABASE**

可选。指定MPP数据库系统要连接到哪个数据库。如果没有指定则默认为\$PGDATABASE。如果\$PGDATABASE也没有设置,则默认为当前系统用户名。用户还可以在命令行上用-d选项指定数据库。

### **USER**

可选。指定用于连接的数据库角色。如果没有指定,默认为当前用户或者\$PGUSER(如果设置)。用户还可以在命令行上用-U选项指定数据库角色。

如果运行gpload的用户不是MPP数据库的超级用户,那么必须为该用户授予适当的权限。更多信息请见*MPP数据库参考手册*。

#### **HOST**

可选。指定MPP数据库的Master主机名。如果没有指定,默认为localhost或者 \$PGHOST (如果设置)。用户还可以在命令行上用-h选项指定Master主机名。

#### PORT

可选。指定MPP数据库的Master端口。如果没有指定,默认为5432或者 \$PGPORT (如果设置)。用户还可以在命令行上用-p 选项指定 Master端口。

### **GPLOAD**

必需。开始装载说明部分。GPLOAD说明必须定义有一个INPUT 小节和一个OUTPUT小节。

#### INPUT

必需。定义要装载的输入数据的位置和格式。gpload 将在当前 主机上启动gpfdist文件分布程序的一个或者更多 实例并且在 MPP数据库中创建指向源数据的外部表定义。注意在其上运行gpload的主机必须对所有的MPP数据库主机(Master和Segment)通过网络可访问。

### SOURCE

必需。INPUT说明的SOURCE块定义源文件的位置。一个INPUT小节可以定义多个SOURCE块。每个定义的SOURCE块对应于将在本地机器上启动的一个gpfdist文件分布程序的实例。每个定义的SOURCE块必须有一个FILE说明。

更多关于使用gpfdist并行文件服务器和单个以及多个 gpfdist实例的信息,请见*MPP数据库管理员手册*中的"装载 和卸载数据"部分。

### LOCAL HOSTNAME

可选。指定gpload运行其上的本地机器的主机名或者IP地址。如果这个机器被配置有多个网络接口卡(NICs),用户可以指定每块NIC的主机名或者IP,以便允许网络流量同时使用所有的NIC。默认是仅使用本地机器的主要主机名或者IP。

#### PORT

可选。指定gpfdist 文件分布程序应该使用的特定端口号。用户还可以提供一个PORT\_RANGE 来从指定的范围中选择可用的端口。如果PORT和PORT\_RANGE同时被定义,那么PORT优先。 如果PORT和PORT\_RANGE都没有定义,默认为在8000和9000之间选择一个可用端口。

如果在LOCAL\_HOSTNAME中声明多个主机名,这个端口号被用于所有主机。如果用户想要使用所有的NICs装载一个给定目录位置的同一个文件或者文件集合,这种配置就是用户想要的。

### **PORT RANGE**

可选。可被用来代替PORT提供一个端口号范围, gpload可以从其中为这个gpfdist 文件分布程序实例选 择一个可用的端口。

#### FILE

必需。指定本地文件系统上的一个文件位置、命名管 道或者目录位置, 其中包含要被装载的数据。用户可 以声明多于一个文件, 只要所有指定文件中数据的格 式相同。

如果这些文件被使用gzip或者bzip2 (有.gz或者.bz2文件扩展名) 压缩,这些文件将被自动解压缩 (在用户路径中有gunzip或者 bunzip2)。

在指定要装载哪些源文件时,用户可以使用通配符 (\*) 或其他C风格的模式匹配来指示多个文件。被指 定的文件假定在相对于gpload 被执行的当前目录的位 置(或者用户可以声明绝对路径)。

#### SSL

可选。指定SSL加密的使用。 如果SSL被设置为 true, gpload 用--ssl启动gpfdist 服务器并且使用gpfdists://协议。

## **CERTIFICATES PATH**

当SSL为true时必需;当SSL为false或者没有指定时不能指定这个参数。CERTIFICATES\_PATH中指定的位置必须包含下列文件:

- 服务器证书文件 server.crt
- 服务器私钥文件 server.key
- 可信证书授权 root.crt

根目录(/)不能被指定为 CERTIFICATES\_PATH。

## **FULLY QUALIFIED DOMAIN NAME**

可选。指定gpload是否把主机名解析成完全限定的域名 (FQDN) 或者本地主机名。如果值被设置为true,名称会被解析到FQDN。如果该值被设置为false,则解析到本地主机名。默认是false。

在某些情况下可能要求一个完全限定的域名。例如,如果 MPP 数据库系统在与ETL应用不同的域且该域能够 被gpload访问。

### **COLUMNS**

可选。以*field\_name:data\_type* 这样的格式指定源数据文件的模式。源文件中的DELIMITER 字符是分隔两个数据值域(列)的东西。一行由一个换行字符(0x0a决定)。

如果输入COLUMNS没有指定,则使用输出TABLE的模式,意味着源数据必须与目标表具有相同的列序、列数以及数据格式。

默认的source-to-target映射基于这一节定义的列名与目标TABLE中列名之间的匹配。默认映射可以使用MAPPING小节覆盖。

## **TRANSFORM**

可选。指定传递给gpload的输入转换的名字。 有关XML转换的信息,请见*MPP数据库管理员手册*中的"装载和卸载数据"。

# TRANSFORM\_CONFIG

当TRANSFORM被指定时,这个元素是必需的。 指定在上面TRANSFORM参数中指定的转换的 配置文件位置。

## MAX LINE LENGTH

可选。一个整数,指定传递给gpload 的XML转换数据中一行的最大长度。

#### **FORMAT**

可选。指定源数据文件的格式:纯文本(TEXT)格式,逗号分隔值(CSV)格式。如果没有指定,这个默认为TEXT。更多有关源数据格式的信息,请见 MPP数据库管理员手册中的"装载和卸载数据"。

### **DELIMITER**

可选。指定在每行数据内分隔列的单个ASCII字符。在TEXT模式中默认是一个制表符,在CSV模式中默认是一个逗号。用户还可以指定一个非可打印ASCII字符或者非可打印Unicode字符,例如:"\x1B"或者 "\u001B"。对于非可打印字符也支持转义字符串语法E'character-code'。 ASCII或Unicode字符必须被封闭在单引号中。例如:E'\x1B'或者E'\u001B'。

### **ESCAPE**

指定用于C转义序列(例如\n、\t、\100等等)以及转义可能被当作行列定界符的数据字符的单个字符。确保选择一个在实际列数据中任何地方都没有使用的转义字符。文本格式文件的默认转义字符是一个\(反斜线),csv格式文件的默认转义字符是一个'(双引号)。不过可以指定另一个字符来表示转义。还可以在文本格式文件中通过指定'OFF'值作为转义值来禁用转义。这对于其中嵌有很多不准备作为转义字符的反斜线的文本格式的Web 日志数据非常有用。

# NULL\_AS

可选。指定表示空值的字符串。TEXT模式中默认是\N (反斜线-N), CSV模式中默认是没有引用的空的值。即便在TEXT模式中,对于想要把空值与空字符串区分开来的情况,用户也可以使用空字符串。任何匹配这个字符串的源数据项将被认为是一个空值。

# FORCE\_NOT\_NULL

可选。在CSV模式中,处理每个被指定的列,仿佛它被引用并且因此不是一个NULL值。对于CSV模式中的默认空值字符串(两个定界符之间什么都没有),这导致缺失的值被计算为长度为零的字符串。

### QUOTE

当FORMAT是CSV时,这个元素是必需的。为CSV模式指定引用字符。默认是双引号(")。

### **HEADER**

可选。指定数据文件中的第一行是一个头部行(包含列名)并且不 应被包括在要被装载的数据中。如果使用多个数据源文件,所有的 文件必须有一个头部行。默认是假定输入文件没有头部行。

### **ENCODING**

可选。源数据的字符集编码。可指定一个字符串常量(例如'SQL\_ASCII')、一个整数编码编号,或者指定'DEFAULT'以使用默认客户端编码。如果没有指定,默认的客户端编码会被使用。有关支持的字符集的信息,请见*MPP数据库参考手册*。

# **ERROR\_LIMIT**

可选。为这个装载操作启用单行错误隔离模式。当被启用时,在输入被处理期间只要没有达到错误限制计数,任何MPP数据库 Segment会抛弃有格式错误的输入行。如果错误限制没有达到,所有好的行将会被装载并且任何错误行都将被抛弃或者被捕捉在错误日志信息中。默认是在遇到第一个错误时中止装载操作。注意单行错误隔离 只适用于有格式错误的数据行,例如有额外或者缺失的属性、有错误数据类型的属性或者有无效的客户端编码序列。如果遇到约束错误(例如主键约束)仍将导致装载操作中止。有关处理装载错误的信息,请见MPP数据库管理员手册中的"装载和卸载数据"。

## **LOG ERRORS**

当ERROR\_LIMIT被声明时,这个元素是可选的。 值可以是true或者false。默认值是false。如果值是true,当运行在单行错误隔离模式中时,格式错误的行会被内部记录下来。 用户可以用MPP数据库的内建SQL函

数gp\_read\_error\_log('*table\_name*') 检查格式错误。如果在装载数据时检测到格式错误,gpload 会用包含错误信息的

表的名字生成一个警告消息,看起来类似于这个消息。

timestamp|WARN|1 bad row, please use GPDB built-in function gp\_re to access the detailed error row

如果LOG\_ERRORS: true被指定,必须指定REUSE\_TABLES: true以便在MPP数据库的错误日志中保留格式错误。如果没有指定REUSE\_TABLES: true,错误信息会在gpload操作后被删除。只有关于格式错误的总结信息会被返回。用户可以用MPP数据库的函数gp\_truncate\_error\_log()从错误日志中删除格式错误。更多有关处理装载错误的信息,请见MPP数据库管理员手册中的"装载和卸载数据"。有关gp\_read\_error\_log()函数的信息,请见MPP数据库参考手册中的CREATEEXTERNAL TABLE命令。

### **EXTERNAL**

可选。定义gpload创建的外部表数据库对象所属的方案。 默认是使用MPP数据库的search\_path。

### **SCHEMA**

当EXTERNAL被声明时,这个元素是必需的。 外部表所在的方案的名称。如果该方案不存在,会返回一个错误。

如果%(百分号字符)被指定,会使用OUTPUT小节中TABLE 指定的表名的方案。如果这个表名没有指定一个方案,则会使用默认方案。

### **OUTPUT**

必需。定义要被装载到数据库中的目标表和最终数据列值。

#### **TABLE**

必需。要装载到其中的目标表名。

#### MODE

可选。如果没有指定,则默认为INSERT。 有三种可用的装

## 载模式:

INSERT - 使用下列方法装载数据到目标表中:

INSERT INTO target\_table SELECT \* FROM input\_data;

UPDATE - 更新目标表中MATCH\_COLUMNS 属性值等于输入数据并且UPDATE\_CONDITION为true (可选条件) 的行的UPDATE\_COLUMNS。如果目标表的列名为保留关键字、有大写字母或包含双引号 ("") ,那么不支持UPDATE。

MERGE - 插入新行并且更新FOOBAR 属性值等于相应输入数据而且MATCH\_COLUMNS为true (可选条件)的已有行的UPDATE\_COLUMNS。当源数据中

的MATCH\_COLUMNS值在目标表数据中没有相应值时会被标识成新行。在那种情况下,源文件中的整个行会被插入,而不仅仅是MATCH和UPDATE列。如果有多个相等的新MATCH\_COLUMNS值,只有其中一个新行将被插入。使用UPDATE\_CONDITION可过滤掉要抛弃的行。 如果目标表的列名为保留关键字、 有大写字母或包含双引号(""),那么不支持MERGE。

# MATCH\_COLUMNS

如果MODE为UPDATE或者 MERGE,则这个元素是必需的。指定被用作更新的连接条件的列。对于要在目标表中更新的行,指定目标列中的属性值必须等于相应的源数据列值。

## **UPDATE COLUMNS**

如果MODE为UPDATE 或者MERGE,则这个元素是必需的。指定对符合MATCH\_COLUMNS条件和可选UPDATE\_CONDITION的行要更新的列。

## **UPDATE CONDITION**

可选。指定目标表中要被更新的行(在MERGE 情况下是要被插入的行)必须满足的一个布尔条件(类似于在WHERE子句中声明的那样)。

### **MAPPING**

可选。如果指定一个映射,它会覆盖默认的source-to-target列映射。默认的source-to-target映射基于源COLUMNS小节定义的列名与目标TABLE中列名之间的匹配。映射可以被指定为:

target\_column\_name: source\_column\_name 或者

target\_column\_name: 'expression'

其中*expression*是在查询的SELECT 列表中指定的任意表达式,例如常量值、列引用、操作符调用、函数调用等等。

### **PRELOAD**

可选。指定在装载操作之前运行的操作。目前唯一的预装载操作是TRUNCATE。

### **TRUNCATE**

可选。如果设置为true,gpload将在装载目标表之前 移除其中所有的行。

# **REUSE TABLES**

可选。如果设置为true,gpload将不会删除它创建的外部表对象和阶段性对象。这些对象将被重用于未来使用同一装载说明的装载操作。这会提高小型装载的性能(正在进行的到同一目标表的小型装载)。

如果LOG\_ERRORS: true被指定,REUSE\_TABLES: true 必须被指定以保留MPP数据库错误日志中的格式错误。如果REUSE\_TABLES: true 没有被指定,格式错误信息会在gpload操作之后被删除。

如果*external\_table\_name*存在,工具会使用存在的表。 如果OUTPUT表结构与数据库中表结构不吻合,工具会返回错误。

# STAGING TABLE

可选。指定gpload操作过程中创建的临时外部表的名称。 该外部表会被gpfdist使用。 REUSE\_TABLES: true必须被指定。如果 REUSE\_TABLES为false或没有指定, STAGING\_TABLE会

被忽略。默认情况下, gpload会采用随机名称创建一个临时外 部表。

如果*external\_table\_name*包含点号(.),gpload会返回错误。如果表已经存在,工具会使用该表。如果表的结构与OUTPUT表定义的结构不匹配,工具会返回错误。

工具会使用EXTERNAL部分定义的SCHEMA值作为 external\_table\_name。如果 SCHEMA的值为%, external\_table\_name的名字会和目标表相同, TABLE的表结构与OUTPUT部分定义的相同。

如果没有设置SCHEMA,工具会在数据库中搜索表 (在search\_path中定义的模式下),如果找不到该表 ,*external\_table\_name*表会在PUBLIC 模式下被创建。

## **FAST MATCH**

可选项。如果设置为true,在重用外部表时,gpload仅 搜索匹配外部表对象的数据库。工具不会从pg\_attribute 检查外部表的列名和列类型。设置该值可以在重用外部表的前提下提高,gpload性能。但是如果实际的列不匹配,该工具会在实际执行时返回错误

默认值为false,工具会去预先检查外部表定义的列名和列类型。REUSE\_TABLES: true也必须被定义。如果 REUSE\_TABLES为false或者没定义,并且 FAST\_MATCH: true被指定, gpload会返回告警信息。

#### SQL

可选。定义在装载操作之前或者之后要运行的SQL命令。用户可以指定多个BEFORE或者AFTER命令。按照想要的执行顺序列出命令。

### **BEFORE**

可选。在装载操作开始之前要运行的一个SQL命令。将命令封闭 在引号中。

### **AFTER**

可选。在装载操作完成之后要运行的一个SQL命令。将命令封闭在引号中。

# 日志文件格式

gpload输出的日志文件具有下面的格式:

timestamp|level|message

其中 *timestamp*的形式是:YYYY-MM-DD HH:MM:SS,*level*是 DEBUG、LOG、INFO、ERROR中间的一个,而 *message*是普通文本消息。

日志文件中可能让人感兴趣的一些INFO消息是(其中#对应于实际的秒数、数据的单位或者失败的行):

INFO|running time: #.## seconds

INFO transferred #.# kB of #.# kB.

INFO|gpload succeeded

INFO|gpload succeeded with warnings

INFO gpload failed

INFO|1 bad row

INFO # bad rows

# 注解

如果用户的数据库对象名使用双引号标识符(定界的标识符)创建,用户必须在gpload控制文件中用单引号指定定界的名称。例如,如果用户这样创建一个表:

CREATE TABLE "MyTable" ("MyColumn" text);

用户的YAML格式的gpload控制文件应该按如下方式引用上述表和列名:

- COLUMNS:

- "MyColumn": text

**OUTPUT:** 

- TABLE: public."'MyTable"

如果YAML控制文件包含MPP数据库4.3.x的ERROR\_TABLE元素,gpload会提示一个告警,显示表明ERROR\_TABLE 是不支持的,如果LOG\_ERRORS和REUSE\_TABLE 被设置为true,加载错误信息会被正常处理。当运行在单行事务处理模式时,格式错误的行会被记录到内部数据库日志。

# 示例

按my\_load.yml中的定义运行一个装载作业:

gpload -f my\_load.yml

装载控制文件的例子:

```
VERSION: 1.0.0.1
DATABASE: ops
USER: gpadmin
HOST: mdw-1
PORT: 5432
GPLOAD:
 INPUT:
  - SOURCE:
     LOCAL HOSTNAME:
      - etl1-1
      - etl1-2
      - etl1-3
      - etl1-4
     PORT: 8081
     FILE:
      - /var/load/data/*
  - COLUMNS:
      - name: text
      - amount: float4
      - category: text
      - descr: text
      - date: date
  - FORMAT: text
  - DELIMITER: '|'
  - ERROR LIMIT: 25
  - LOG ERRORS: true
 OUTPUT:
  - TABLE: payables.expenses
  - MODE: INSERT
 PRELOAD:
  - REUSE TABLES: true
 SQL:
 - BEFORE: "INSERT INTO audit VALUES('start', current timestamp)"
 - AFTER: "INSERT INTO audit VALUES('end', current_timestamp)"
```

# 另见

gpfdist, MPP数据库参考手册中的CREATE EXTERNAL TABLE。

# gplogfilter

在MPP数据库日志文件中搜索指定的项。

# 概要

```
gplogfilter [timestamp_options] [pattern_options]
  [output_options] [input_options] [input_file]
```

gplogfilter --help

gplogfilter --version

# 描述

gplogfilter工具可以被用来在一个MPP数据库日志文件中搜索匹配指定条件的项。如果没有提供输入文件,那么gplogfilter将使用环境变量\$MASTER\_DATA\_DIRECTORY 在标准日志位置定位MPP的Master日志文件。要从标准输入读取,可用一个破折号(-)作为输入文件名。输入文件可以被使用gzip压缩。在输入文件中,日志项由其YYYY-MM-DD[hh:mm[:ss]]格式的时间戳标识。

如果通过gpssh工具运行 gplogfilter,用户还可以用它来一次搜索所有的 Segment日志文件。例如,要显示每个Segment 日志文件中的最后三行:

gpssh -f seg\_host\_file

- => source /usr/local/mpp-db/mpp path.sh
- => gplogfilter -n 3 /gpdata/\*/pg\_log/gpdb\*.csv

gplogfilter的输出默认被发送到标准输出。可以使用-o选项把输出发送到一个文件或者一个目录。如果用户提供了一个以.gz结束的输出文件名,该输出文件默认将被以最大压缩率压缩。如果输出目标是一个目录,输出文件将被命名为输入文件的名字。

# 选项

时间戳选项

# -b datetime | --begin=datetime

以格式YYYY-MM-DD [hh:mm[:ss]]指定要搜索的日志项的开始日期和时间。

如果指定一个时间,日期和时间必须被封闭在单引号或者双引号中。 这个例子将日期和时间封闭在单引号中:

gplogfilter -b '2013-05-23 14:33'

## -e datetime | --end=datetime

指定一个YYYY-MM-DD [hh:mm[:ss]]格式的结束日期和时间来停止搜索日志信息。

## -d time | --duration=time

以[hh][:mm[:ss]]格式指定要搜索的日志项所在的时间长度。如果指定这个选项 但没有指定-b或-e选项,将使用当前时间作为基础。

# 模式匹配选项

-c i [gnore] | r [espect] | --case=i [gnore] | r [espect]

除非前面放上--case=ignore选项,字母表符号的匹配默认是大小写敏感的。

-C 'string' | --columns='string'

从日志文件中选择特定的列。将想要的列指定为由逗号定界的列编号串,列编号从1开始,从左到右依次是1、2、3等等。关于日志文件格式的详情以及可用列的列表及其相关编号请见*MPP数据库管理员手册*中的"查看数据库服务器日志文件"部分。

-f 'string' | --find='string'

查找包含指定字符串的日志项。

-F 'string' | --nofind='string'

拒绝包含指定字符串的日志项。

-m regex | --match=regex

查找匹配指定的Python正则表达式的日志项。Python正则表达式语法请参考 https://docs.python.org/library/re.html

-M regex | --nomatch=regex

拒绝匹配指定Python正则表达式的日志项。Python正则表达式语法请参考 https://docs.python.org/library/re.html

-t | --trouble

只查找在第一行有ERROR:、FATAL:或者 PANIC:的日志项。

## 输出选项

-n integer | --tail=integer

限制输出为找到的符合条件日志项中的最后integer项。

-s offset [limit] | --slice=offset [limit]

从符合条件的日志项列表中,从offset项处开始返回limit个项,其中为零(0)的offset指示结果集中的第一项并且任何不超过零的offset表示从结果集的末尾开始数。

-o output\_file | --out=output\_file

将输出写到指定的文件或者目录位置而不是STDOUT。

-z 0-9 | --zip=0-9

使用gzip把输出文件压缩到指定的压缩级别,其中0是不压缩而9是最大压缩。如果用户提供一个以.gz结束的输出文件名,该输出文件默认

将被使用最大压缩级别压缩。

## -a | --append

如果输出文件已经存在,会追加到该文件而不是覆盖它。

## 输入选项

## input\_file

要在其中搜索的输入日志文件的名称。如果没有提供输入文件,gplogfilter将使用环境变量\$MASTER\_DATA\_DIRECTORY来定位MPP数据库的Master日志文件。要从标准输入读取,可使用一个破折号(-)作为输入文件名。

## -u | --unzip

使用gunzip解压输入文件。如果输入文件名以.gz结束,它默认将被解压。

## --help

显示在线帮助。

### --version

显示这个工具的版本。

# 示例

显示Master日志文件中的最后三个错误消息:

gplogfilter -t -n 3

显示Master日志文件中时间戳位于最后10分钟内的所有日志消息:

gplogfilter -d:10

显示Master日志文件中包含字符串 | con6 cmd11 | 的日志消息:

gplogfilter -f '|con6 cmd11|'

使用gpssh 在Segment主机上运行gplogfilter并且在Segment日志文件中搜索含有字符串 con6的日志消息,把输出保存到一个文件。

gpssh -f seg\_hosts\_file -e 'source
/usr/local/mpp-db/mpp\_path.sh ; gplogfilter -f
con6 /gpdata/\*/pg\_log/gpdb\*.csv' > seglog.out

# 另见

gpssh, gpscp

# gpmapreduce

按照YAML规范文档中的定义运行MPP的MapReduce作业。

# 概要

```
gpmapreduce -f yaml_file [dbname [username]]
    [-k name=value | --key name=value]
    [-h hostname | --host hostname] [-p port| --port port]
    [-U username | --username username] [-W] [-v]

gpmapreduce -x | --explain

gpmapreduce -X | --explain-analyze

gpmapreduce -V | --version

gpmapreduce -h | --help
```

# 先决条件

在运行此程序之前,需要执行以下操作:

- 用户必须在YAML文件中定义用户的MapReduce作业。 有关MPP MapReduce规范的信息,请参阅 MPP数据库参考手册。
- 用户必须是MPP数据库超级用户才能运行使用不可信Perl或Python编写的MapReduce作业。
- 用户必须是MPP数据库超级用户才能运行带有EXEC和FILE输入的 MapReduce作业。
- 用户必须是MPP数据库超级用户才能运行带有GPFDIST输入的 MapReduce作业,除非用户具有适当的权限。有关更多信息,请参 阅MPP数据库参考手册。

# 描述

MapReduce是由Google开发的用于在商用服务器阵列上处理和生成大型数

据集的编程模型。 MPP MapReduce允许熟悉MapReduce范式的编程人员编写map和reduce函数,并将它们提交给 MPP数据库并行引擎处理。

为了使MPP能够处理MapReduce函数,需要在YAML文档中定义函数,然后将其传递到MPP MapReduce程序gpmapreduce,以供MPP数据库并行引擎执行。MPP 系统负责分布输入数据、在一组机器上执行程序、处理机器故障以及管理所需的机器间通信。

# 选项

## -f yaml\_file

必须。包含MPP MapReduce作业定义的YAML文件。请参阅 MPP数据库参考手册。

-? | --help

显示帮助,然后退出。

-V | --version

显示版本信息,然后退出。

-v | --verbose

显示详细输出。

-x | --explain

不运行MapReduce作业,而是生成解释计划。

-X | --explain-analyze

运行MapReduce作业并生成解释 - 分析计划。

-k | --keyname=value

设置一个YAML变量。需要一个数值。如果未指定变量名称,则默认为"key"。

## 连接选项

## -h host | --host host

指定运行MPP的Master数据库服务器的机器的主机名。 如果未指定,则从环境变量 PGHOST读取或默认为localhost。

# -p port | --port port

指定MPP的Master数据库服务器正在侦听连接的TCP端口。 如果未指定,则从环境 变量PGPORT读取或默认为5432。

-U username | --username username

T要连接的数据库角色名称。如果未指定,则从环境变量PGUSER 读取或默认为当前系统用户名。

-W | --password

强制口令提示。

# 示例

运行my\_yaml.txt中定义的MapReduce作业,并连接到数据库mydatabase:

gpmapreduce -f my yaml.txt mydatabase

# 另见

MPP数据库参考手册中的MPP MapReduce规范。

# gpmovemirrors

移动镜像实例到新位置

# 概要

```
gpmovemirrors -i move_config_file [-d master_data_directory]
     [-l logfile_directory]
     [-B parallel_processes] [-v]
```

gpmovemirrors -?

gpmovemirrors --version

# 描述

gpmovemirrors工具将镜像实例移动到新位置。客户可能会想移动镜像到新位置以优化分布和数据存储

在移动Segment之前,该工具会验证Mirror是否存在,他们对应的主节点是 否在线和目前同步状态在同步还是 异步模式。

默认情况下,工具会提示您输入所有镜像要移动到的文件系统路径。

客户必须保证运行gpmovemirrors的用户(gpadmin用户) 拥有该位置的访问权限。所以最好在运行gpmovemirrors工具之前在各个Segment主机上创建好对应的路径,并使用chown命令付给指定用户相应的权限。

# 选项

-B parallel\_processes

要并行移动的镜像Segment数量。如果不指定,工具默认启动4个并行

进程,具体数量与要移动多少镜像实例数量有关。

-d master data directory

主机数据目录。如果不指定,默认使用\$MASTER DATA DIRECTORY的值。

-i move\_config\_file

包含哪些镜像要被移动,被移动到哪里等信息的配置文件。

每一个主Segment必须有一个对应的镜像Segment列出。配置文件中的每行都用以下格式 (就像gp\_segment\_configuration元数据表中的信息一样):

address|port|data\_dir new\_address|port|data\_dir

contentID是Segment实例的content ID, address是Segment主机的主机名或IP地址, port是沟通端口,data\_dir是Segment 实例的数据目录。

-l logfile\_directory

日志写入路径。默认为~/gpAdminLogs。

-v (详细日志)

设置日志输出为详细级别。

--version (显示工具版本)

显示工具的版本信息。

-? (帮助)

显示在线帮助信息。

### 示例

将MPP数据库系统移动到一个不同的主机集合:

\$ gpmovemirrors -i move\_config\_file

此处move\_config\_file的配置信息如下:

sdw2|50001|/data2/mirror/gpseg1 sdw3|50001|/data/mirror/gpseg1 sdw2|50001|/data2/mirror/gpseg2 sdw4|50001|/data/mirror/gpseg2 sdw3|50001|/data2/mirror/gpseg3 sdw1|50001|/data/mirror/gpseg3

# gpperfmon\_install

安装MPP Command Center使用的gpperfmon数据库,并可选择启用数据收集代理。

## 概要

```
gpperfmon_install --port gpdb_port

[--enable --password gpmon_password [--pgpass path_to_file]]

[--verbose]

gpperfmon_install --help | -h | -?
```

## 描述

gpperfmon\_install工具自动执行启用数据收集代理程序所需的步骤。用户必须是MPP数据库系统用户 (gpadmin) 才能运行此工具。 --port选项是必须的。使用--enable选项时,还需要--password选项。使用--port选项来 提供MPP数据库的Master实例的端口。如果使用--enable选项,必须在工具完

成后重新启动MPP数据库。

在没有--enable选项时,此工具只创建gpperfmon数据库(用于存储由数据收集代理收集的系统度量的数据库)。在使用--enable选项运行时,该工具还会运行启用性能监视器数据收集代理所需的以下附加任务:

- 1. 在MPP数据库中创建gpmon超级用户角色。数据收集代理需要连接到数据库并写入数据。gpmon超级用户角色默认使用MD5加密口令认证。使用--password选项来设置gpmon超级用户的口令。
- 2. 更新\$MASTER\_DATA\_DIRECTORY/pg\_hba.conf文件。 该工具将这些行添加到基于主机的认证文件 (pg\_hba.conf) 中:

local gpperfmon gpmon md5
host all gpmon 127.0.0.1/28 md5
host all gpmon ::1/128 md5

第二和第三行的host项,让gpmon能访问 所有MPP数据库。

Note: 可能有必要在运行gpperfmon\_install工具以限制 gpmon角色访问数据库或更改身份验证方法之后再编辑 pg\_hba.conf文件中的行。运行gpstop -u重新加载 MPP数据库的pg\_hba.conf文件。

 要将gpmon的访问限制为仅访问gpperfmon数据库,编 辑pg\_hba.conf文件中host项。对于 gpmon用户,将第二个字段 从all更改为 gpperfmon:

local gpperfmon gpmon md5
host gpperfmon gpmon 127.0.0.1/28 md5
host gpperfmon gpmon ::1/128 md5

。 gpperfmon\_install工具采用默认的MD5认证方法。MPP数据库可以选择性地配置为使用SHA-256哈希算法来计算保存在系统目录中的口令哈希值。这与MD5认证方法不兼容,MD5认证方法在系统目录中需要MD5哈希或明文口令。 因此,如果在数据库中启用了SHA-256哈希算法,则必须在运行gpperfmon\_install工具后编辑pg\_hba.conf文件。

对于host项,将 gpmon角色的身份认证从md5改为 password:

localgpperfmongpmonmd5hostallgpmon127.0.0.1/28passwordhostallgpmon::1/128password

password认证方法提交用户的明文口令进行验证,不应在不可信的网络上使用。请参阅*MPP数据库管理员手册*中的"保护MPP数据库中的密码"。

3. 更新口令文件(.pgpass)。为了允许数据收集代理作为 gpmon角色进行连接而无需口令提示,用户必须拥有一个包含gpmon 用户项的口令文件。该工具将以下项添加到用户的口令文件(如果该文件不存在,该工具将创建它):

\*:5432:gpperfmon:gpmon:gpmon\_password

如果用户的口令文件不在默认位置(~/.pgpass),使用--pgpass 选项指定文件位置。

- 4. 设置MPP Command Center的服务器配置参数。数据收集代理必须启用以下参数才能开始收集数据。该工具在MPP数据库的postgresql.conf配置文件中设置以下参数:
  - gp\_enable\_gpperfmon=on (在集群中所有的postgresql.conf 文件中)
  - o gpperfmon\_port=8888 (在集群中所有的postgresql.conf文件中)
  - gp\_external\_enable\_exec=on (在Master的 postgresql.conf文件中)

数据收集代理可以通过在gpperfmon.conf配置文件中设置参数进行配置。参阅数据收集代理配置获取更多详细信息。

## 选项

#### --enable

除了创建gpperfmon数据库外,还执行启用数据收集代理所需的附加步骤。指定--enable时,工具还将创建并配置gpmon超级用户帐户,并在postgresql.conf文件中设置Command Center服务器配置参数。

### --password gpmon\_password

如果必须指定了--enable。设置gpmon超级用户的口令。 如果未指定--enable则不允许使用。

### --port gpdb\_port

必需。指定MPP数据库Master的连接端口。

### --pgpass path\_to\_file

在--enable选项被指定时可选。如果口令文件不在~/.pgpass的默认位置,则这个选项指定口令文件的位置。

#### --verbose

将日志记录级别设置为详细。

### --help | -h | -?

显示在线帮助。

### 数据收集代理配置

\$MASTER\_DATA\_DIRECTORY/gpperfmon/conf/gpperfmon.conf文件存储数据收集代理的配置参数。要使这些选项的配置更改生效,必须保存gpperfmon.conf然后重新启动MPP数据库 服务器 (gpstop -r)。

gpperfmon.conf文件包含以下配置参数。

参数	描述
log _lo cat ion	指定gpperfmon日志文件的目录位置。默认是\$MASTER_DATA_DIRECTORY/gpperfmon/logs。

mi n\_ qu ery \_ti me

指定统计信息收集的最短查询运行时间(秒)。所有运行时间超过此值的查询都将记录在queries\_history 表中。对于运行时间较短的查询,不收集历史数据。默认为20秒。

如果用户知道要为所有查询收集数据,则可以将此参数设置为较低值。但是,如果将最小查询运行时间设置为零,即使对于MPP Command Center运行的大量查询也会收集数据,从而创建大量可能无用的数据。

ma
x\_l
og
\_si
ze

这个参数不包含在gpperfmon.conf中,但是它可能被添加到这个文件中。

为防止日志文件增长过大,可以将max\_log\_size参数添加到gpperf mon.conf 中。该参数的值以字节为单位。例如:

max log size = 10485760

使用此设置,日志文件将在系统转到新的日志文件之前增长到10M B。

par titi on

\_a

ge

gperfmon统计数据将被保留的月数。它的默认值是0,这意味着我们不会丢失任何数据。

qu ant um 指定所有Segment上的数据收集代理更新之间的时间(以秒为单位)。有效值为10、15、20、30和60。缺省值为15秒。

如果用户偏好较低粒度的性能视图,或者想要收集和分析系统度量的最小量,请选择较高的quantum。要更频繁地收集数据,请选择一个较低的值。

har ve st\_ int erv

数据捕获时间段,以秒为单位。一次数据捕获会将gpperfmon外部表 (\_tail) 的近期数据移动到对应的历史文件。默认为120。最小值为30。

ign ore \_q

ex

ec

р

al

(弃用)。设置为true时,数据收集代理不会收集gpperfmon数据库的 queries\_\*表中的性能数据: rows\_out、cpu\_elapsed、cpu\_currpct、 skew\_cpu和skew\_rows 缺省设置true可以减少gpmmon进程所消耗的内存量。如果用户需要这些额外的性能数据,请将此参数设置为false。

ac ket

此参数允许用户为后备Master指定额外的主机名。例如,如果后备Master有两个NIC,则可以输入:

sm dw \_ali as es

smdw\_aliases=smdw-1,smdw-2

如果MPP Command Center失去与后备Master的连接,则这个可选的容错参数非常有用。它不会持续重试连接到主机smdw,而是尝试连接到基于NIC的别名smdw-1或 smdw-2。这确保了Command Center控制台可以持续调查和监控后备Master。

## 注解

gpperfmon数据库和MPP Command Center需要gpmon角色。 在创建gpperfmon数据库和gpmon角色后,用户可以更改 gpmon角色的口令并更新MPP Command Center信息用来连接到gpperfmon 数据库:

1. 以超级用户身份登录到MPP数据库,并使用ALTER ROLE命令更改gpmon的口令。

# ALTER ROLE gpmon WITH PASSWORD 'new\_password';

2. 更新MPP Command Center使用的.pgpass文件中的口令。默认文件位置是gpadmin 的主目录(~/.pgpass)。.pgpass文件包含一行gpmon口令。

\*:5432:gpperfmon:gpmon:new\_password

3. 使用Command Center的gpcmdr工具重新启动MPP Command Center。

\$ gpcmdr --restart

gpperfmon监控系统在启动后需要一些初始化。监控信息在几分钟后出现,而不是在安装和启动gpperfmon系统之后立即出现。

### 示例

只创建gpperfmon数据库:

\$ su - gpadmin \$ gpperfmon install --port 5432

### 创建gpperfmon数据库,创建gpmon超级用户,并启用数据收集代理:

\$ su - gpadmin \$ gpperfmon\_install --enable --password changeme --port 5432 \$ gpstop -r

## 另见

gpstop

# gppkg

在整个集群中安装.gppkg格式的MPP数据库扩展 (例如PL/Java、PL/R和MADlib) 及其依赖项。

## 概要

### 描述

MPP软件包管理器 (gppkg) 工具在集群中的所有主机上安装 MPP数据库扩展以及任何依赖项。在系统扩展和Segment恢复的情况下,它也将自动在新主机上安装扩展。

Note: 在MPP数据库重要升级之后,用户必须再次下载并安装所有 gppkg扩展。

### 选项

-a (不要提示) 不要提示用户确认。

-c | --clean

协调集群的包状态以匹配Master主机的状态。在失败或部分的安装/卸载后运行此选项可确保软件包安装状态在集群间保持一致。

-d master\_data\_directory

如果未指定,则使用为\$MASTER\_DATA\_DIRECTORY设置的值。

-i package | --install=package

安装给定的包。这包括任何前/后安装步骤以及任何依赖关系的安装。

#### --migrate GPHOME\_old GPHOME\_new

从单独的\$GPHOME迁移软件包。将包从一个版本的MPP数据库运输到另一个版本。

例如:gppkg --migrate /usr/local/mpp-db-<old-version> /usr/local/mpp-db-<new-version>

当迁移包时,要求满足以下条件。

- 至少目标数据库集群的Master实例要启动(实例安装 在GPHOME\_new)。在运行gppkg命令之前执行gpstart -m命令启 动 Master实例。
- 运行 GPHOME\_new 安装目录下的gppkg工具。

### -q | --query query\_option

提供有关已安装软件包的query\_option指定的信息。一次只能指定一个query\_option。 下表列出了query\_option的可能值。<package\_file>是一个包的名字。

Table 1. o	gppkg的查询选项	Į
------------	------------	---

query_option	返回
<package_file></package_file>	是否安装了指定的软件包。
info <package_file></package_file>	关于指定软件包的名称、版本和其 他信息。
list <package_file></package_file>	指定包的文件内容。
all	列出所有已安装的包。

### -r name-version | --remove=name-version

删除指定的包。

### -u package | --update=package

更新给定的包。

Warning: 更新程序包的过程包括删除与程序包相关的系统对象的所有 先前版本。 例如,以前版本的共享库被删除。更新过程之后,如果函 数引用已被删除的包文件, 则数据库函数在调用时将失败。

- --version (显示工具版本) 显示此工具的版本。
- -v | --verbose 将日志记录级别设置为详细。
- -?|-h|--help 显示在线帮助。

## gprecoverseg

恢复已标记为down的主Segment实例或镜像Segment实例(如果启用了镜像)。

## 概要

```
gprecoverseg [-p new_recover_host[,...]] | -i recover_config_file] [-d master_data_director
    [-B parallel_processes] [-F [ [-L|-R max_rate]] [-a] [-q]
    [--no-progress] [-l logfile_directory]

gprecoverseg -r

gprecoverseg -o output_recover_config_file
    [-p new_recover_host[,...]]

gprecoverseg -? | --help

gprecoverseg --version
```

## 描述

在启用了镜像的系统中,gprecoverseg工具会重新激活故障的Segment实例,并识别需要重新同步的已更改的数据库文件。一旦gprecoverseg完成这个过程,系统进入resyncronizing模式,直到被恢复的Segment更新为最新为止。在重新同步期间系统处于在线状态并且能被正常操作。

在增量恢复期间(未指定-F选项),如果gprecoverseg 在启用了镜像的系统中检测到禁用镜像的Segment实例,则工具将报告该Segment禁用了镜像,不会尝试恢复该Segment实例,并继续恢复过程。

Segment实例可能由于多种原因故障,如主机故障、网络故障或磁盘故障。当一个Segment实例故障时,其状态在MPP数据库系统目录中被标记为 down并且在 change tracking模式下激活其镜像。为了使发生故障的Segment实例重新运行,首先必须纠正使其故障的问题,然后使用gprecoverseg在MPP数据库中恢复Segment实例。

使用gprecoverseg进行Segment恢复需要有一个活动镜像来从其中恢复。对于 没有启用镜像的系统,或者在发生双重故障的情况下(主Segment和镜像Segment同时故障),必须采取 手动步骤恢复出现故障的Segment实例,然后执行系统重新启动让Segment重新联机。例如,该命令重新 启动系统。

#### gpstop -r

缺省情况下,将恢复出现故障的Segment,这意味着系统将该Segment重新联机到与最初配置的主机和数据目录位置相同的位置。在这种情况下,请使用以下格式的恢复配置文件(使用-i)。

<failed\_host\_address>:<port>:<data\_directory>

在某些情况下,这可能是不可能的(例如,如果主机物理损坏,无法恢

复)。gprecoverseg 允许用户将失败的Segment恢复成全新的主机(使用-p),在剩余的活动Segment主机上的 备用数据目录位置(使用-s)或通过以下面的格式提供恢复配置文件(使用-i)。 SPACE关键字表示所需空间的位置。不要添加额外的空间。

<failed\_host\_address>:<port>:<data\_directory>SPACE 
<recovery\_host\_address>:<port>:<data\_directory>

有关恢复配置文件的详细信息和示例,请参阅下面的-i选项。

gp\_segment\_configuration系统目录表可以帮助用户确定当前的Segment 配置,以便用户可以规划镜像恢复配置。例如,运行以下查询:

=# SELECT dbid, content, address, port, datadir FROM gp\_segment\_configuration ORDER BY dbid;

新恢复的Segment主机必须预先安装MPP数据库软件,并且配置与现有 Segment主机完全相同。所有 当前配置的Segment主机上必须存在备用数 据目录位置,并且有足够的磁盘空间来容纳故障的Segment。

恢复过程会在MPP数据库系统目录中再次标记该Segment,然后启动重新同步过程,以使Segment的事务状态处于最新状态。在重新同步期间系统在线并且可用。要检查重新同步进程运行的状态:

gpstate -m

gprecoverseg的全量恢复支持IO资源限制,有两种资源限制模式:cgroup模式和throttle模式。 cgroup模式支持限制磁盘IO和网络IO, throttle模式支持限制磁盘IO。

Note:cgroup只支持限制直接的磁盘IO,操作系统缓存和数据库buffer不受

影响, throttle是对传输文件的速度做限制。throttle的限制对象是实例级别, cgroup是操作系统级别。另外cgroup支持动态调整资源并立即生效。

资源限制是通过内核扩展模块iolimit实现的,所有的限制都是作用于主节点。对于gprecoverseg的全量恢复,主节点几乎没有写操作,所以只限制写没什么效果。

## 选项

### -a (不提示)

不要提示用户确认。

#### -B parallel\_processes

并行恢复的Segment数。如果未指定,则实用程序将启动最多16个并行进程,具体取决于需要恢复多少个Segment实例。

### -d master\_data\_directory

可选。Master主机的数据目录。如果未指定,则使用为\$MASTER DATA DIRECTORY设置的值。

### -F (完全恢复)

可选。执行活动Segment实例的完整副本以恢复出现故障的 Segment。 默认情况下,仅复制Segment关闭 时发生的增量更改。

Warning: 全量恢复会在从活动(当前主)Segment实例复制数据目录到故障Segment之前将 故障Segment的数据目录清空。在执行全量恢复前,确保Segment故障不会引起数据损坏,并且任何Segment 磁盘故障问题均被修复。

全量恢复会持续很长时间,所以gprecoverseg会显示每个Segment的运行处理过程。每个Segment的处理过程每秒更新一次,采用ANSI逃逸符更新每一个Segment行,如果您将gprecoverseg工具执行日志输出到文件或您的终端不支持ANSI逃逸符,可以采用-s选项停用ANSI逃逸符。该操作会使每个Segment每秒输出一个新行。带有--no-progress时会禁用处理报告。

### -i recover\_config\_file

指定文件的名称以及有关故障Segment要恢复的详细信息。文件中的

每一行都是以下格式。SPACE 关键字表示所需空间的位置。不要添加额外的空间。

#### 注释

以#开始的行被视为注释并被忽略。

### 要恢复的Segments

第一行之后的每一行指定要恢复的Segment。这一行可以有两种格式之一。在就地恢复的情况下,在该行中输入一组冒号分隔的字段。例如:

failedAddress:failedPort:failedDataDirectory

要恢复到新位置,请在行中输入由空格分隔的两组字段。SPACE表示不要添加额外的空间。

failedAddress:failedPort:failedDataDirectorySPACEnewAddress: newPort:newDataDirectory

示例

单个镜像的就地恢复

sdw1-1:50001:/data1/mirror/gpseg16

将单个镜像恢复到新主机

sdw1-1:50001:/data1/mirror/gpseg16SPACEsdw4-1:50001:/data1/recover1/gpseg16

#### 获取示例文件

用户可以使用-o选项输出一个恢复配置文件的样例以便从它开始。

#### -l logfile\_directory

写入日志文件的目录。默认为~/gpAdminLogs。

-L

以cgroup资源限制模式运行数据重分布,使用之前请确保iolimit扩展已经加载,并且expand控制组的资源配额已经配置(使用gpcgroup工具初始化控制组和设置expand控制组的资源配额)。数据重分布的数据库后台进程将受到basebackup控制组的资源限制。

#### -o output\_recover\_config\_file

指定文件名称和位置以输出示例恢复配置文件。输出文件以-i选项所需的格式列出当前故障的Segment及其默认恢复位置。 与-p选项一起使用可输出用于在不同主机上恢复的示例文件。 如果需要,可以编辑此文件以提供备用恢复位置。

### -p new\_recover\_host[,...]

在当前配置的MPP数据库阵列之外指定一个备用主机,用于恢复故障的Segment。在多个故障的Segment 主机的情况下,用户可以指定一个逗号分隔的列表。备用主机必须安装和配置MPP数据库软件,并具有与当前 网段主机相同的硬件和操作系统配置(相同的OS版本、语言环境、gpadmin用户帐户、创建的 数据目录位置、交换的ssh密钥、网络接口数量、网络接口命名约定等)。

### -q (无屏幕输出)

以静默模式运行。命令输出不显示在屏幕上,但仍然写入日志文件。

### -r (重新平衡Segment)

在Segment恢复之后,Segment实例可能并未回到系统初始化时为它给定的优先角色。这可能会让系统陷入一种潜在的非平衡状态,因为某些Segment主机拥有的活动Segment数量可能超过最高系统性能时的数量。这一选项通过将主Segment和镜像Segment返回到其优先角

色来重新平衡主Segment和镜像Segment。在运行gprecoverseg -r 之前,所有的Segment都必须有效且同步。如果有任何正在进行的查询,它们将被取消并回滚。

#### -R max rate

以throttle资源限制模式运行数据重分布,使用之前请确保iolimit扩展已经加载。 max\_reate的单位是KB/s,范围从32(32KB/s)到 1048576(1GB/s)。

-S

展示pg\_basebackup顺序处理过程而不是当前情况。此选项在需要写入文件或tty不支持转义字符时有用。默认是展示当前处理过程。

#### --no-progress

禁用产生于pg\_basebackup的处理过程报告。默认配置为显示基础备份的处理过程。

-v (详细模式)

将日志记录输出设置为verbose。

--version (版本)

显示此工具的版本。

-? (帮助)

显示在线帮助。

## 示例

恢复所有故障的Segment实例:

#### \$ gprecoverseg

恢复后,重新平衡用户的MPP数据库系统,将所有Segment重置为其首选 角色。 首先检查所有Segment已启动并同步。 \$ gpstate -m \$ gprecoverseg -r

将任何故障的Segment实例恢复到新配置的空闲Segment主机:

\$ gprecoverseg -i recover\_config\_file

输出默认恢复配置文件:

\$ gprecoverseg -o /home/gpadmin/recover\_config\_file

以cgroup资源限制模式全量恢复所有故障的Segment示例,限制每个节点的全量恢复读速度为20MB/s、网络传输速度10MB/s,1分钟后取消资源限制:

- \$ gpconfig -c session preload libraries -v iolimit
- \$ gpstop -u
- \$ gpcgroup set -f all\_hosts -G basebackup -R 20 -N 10
- \$ sleep 60 && gpcgroup set -f all\_hosts -G basebackup -R 0 -N 0
- \$ gprecoverseg -F -a -L

以throttle资源限制模式全量恢复所有故障的Segment示例,限制每个 segment的全量恢复速度为4MB/s:

- \$ gpconfig -c session\_preload\_libraries -v iolimit
- \$ gpstop -u
- \$ gprecoverseg -F -a -R 4096

gpstart, gpstop

# gpreload

重新装载MPP数据库表数据,根据指定的列对数据进行排序。

## 概要

gpreload -d database [-p port] {-t | --table-file} path\_to\_file [-a]

gpreload -h

gpreload --version

## 描述

gpreload工具重新装载列数据被排序过的表数据。对于使用表存储选项 appendoptimized=TRUE并启用压缩创建的表,使用排序数据重新装载数据 可以改善表压缩。用户在一个文本文件中指定要重新装载的表要排序的表 列列表。

当列中的数据的不同值与行总数相比来说相对较小时,压缩可以通过对数 据进行排序来改进。 对于正在重新装载的表,要排序的列顺序可能会影响压缩。具有最少不同值的列应该首先列出。 例如,列出州和城市一般会获得比列出城市和州更好的压缩。

public.cust\_table: state, city
 public.cust\_table: city, state

有关gpreload使用的文件格式的信息,请参阅--table-file选项。

## 注解

为了提高重载性能,在重载数据之前,应该删除正在重载的表上的索引。

重新装载表数据后运行ANALYZE命令可能会因为重新装载数据的数据分布发生更改而查询性能。

对所有的表,该工具都会将数据复制到临时表,然后将实际表清空,再从 临时表按照指定顺序插入到实际表。 每个表的重载操作都在一个单一的事 务中处理。

对于分区表来说,可以单独重载子分区的数据。但是无论如何,数据都会从根分区表插入,这会在整个表上产生 ROW EXCLUSIVE锁。

### 选项

-a (不提示)

可选。如果指定,则gpreload工具不会提示用户进行确认。

-d database

包含要重新装载的表的数据库。gpreload工具以运行该工具的用户的

身份连接到数据库。

#### -p *port*

MPP数据库的Master端口。如果未指定,则使用PGPORT环境变量的值。 如果该值不可用,则返回错误。

### {-t | --table-file } path\_to\_file

包含要重新装载的方案限定的表名列表的文件位置和名称,该文件中还有要从MPP数据库重新排序的列名。 仅支持用户定义的表。视图或系统目录表不受支持。

如果在文件中列出的表上定义有索引,gpreload会提示继续。 每一行都指定一个表名和要排序的列的列表。这是文件中每一行的格 式:

schema.table\_name: column [desc] [, column2 [desc] ... ]

表名后跟一个冒号(:),然后至少一个列名。如果指定多个列,请使用逗号分隔列名称。列按升序排序。 在列名后面指定关键字desc以降序对列进行排序。

通配符不受支持。

如果文件中有错误,gpreload会报告第一个错误并退出。

以下例子展示了重载三个表:

public.clients: region, state, rep\_id desc public.merchants: region, state test.lineitem: group, assy, whse

在第一个表public.clients中,rep\_id列中的数据 被按照降序排序。其他列中的数据被按照升序排序。

### --version (显示工具版本)

列出工具版本。

### -? (帮助)

提供在线帮助。

### 示例

此示例命令将重新装载文件data-tables.txt中列出的数据库 mytest中的表。

gpreload -d mytest --table-file data-tables.txt

## 另见

MPP数据库参考手册中的CREATE TABLE。

# gprestore

恢复由gpbackup创建的MPP数据库备份。默认gprestore 使用位于MPP Master主机数据目录下的元数据文件和DDL文件,使用存储于Segment主机上的表数据CSV文件。

## 概要

```
gprestore --timestamp YYYYMMDDHHMMSS
  [--backup-dir directory]
  [--create-db]
  [--debug]
  [--exclude-schema schema_name]
  [--exclude-table schema.table]
  [--exclude-table-file file_name]
  [--include-schema schema name]
  [--include-table schema.table]
  [--include-table-file file_name]
  [--data-only | --metadata-only]
  [--jobs int]
  [--on-error-continue]
  [--plugin-config config_file_location]
  [--quiet]
  [--redirect-db database_name]
  [--verbose]
  [--version]
  [--with-globals]
  [--with-stats]
```

## 描述

gprestore --help

使用gprestore从备份集恢复时,必须包含--timestamp选项来指定 确切一个来自备份集的恢复时间值(YYYYMMDDHHMMSS)。如果在备份时指定了--backup-dir来获取备份文件。

如果指定的是增量备份,那么还需要一个完全备份的文件集合(一个全备和所有需要的增量备份)。 gprestore会在开始恢复之前验证备份集合是可用的。

Important: 对于增量备份集,备份必须在单独设备上。例如,备份集的所有 文件必须全部在 Data Domain系统。 有关增量备份的详细信息,请见使用gpbackup和gprestore创建增量备份。

当从备份集恢复时,gprestore会将数据恢复到创建备份时指定的数据库。如果目标数据库存在,并且数据库中存在该表,恢复操作会失败。如果目标数据库不存在,可以通过指定 --create-db选项来创建数据库。也可以通过指定--redirect-db 选项来将数据恢复到不同的数据库。

当恢复分区表的子分区时,分区表和数据会一同被恢复。例如,使用gpbackup的 --include-table-file选项创建了一个备份。恢复数据时会恢复创建分区表并 只将数据恢复到列表文件中指定的分区。

MPP数据库系统对象会自动包含在gpbackup备份集中,但是只有在用gprestore进行恢复时指定--with-globals选项才会进行恢复。相似的情况还有,如果备份时使用--with-stats选项备份了查询计划统计信息,那么在使用gprestore进行恢复时必须指定--with-stats才能对这部分数据进行恢复。默认情况下,只有备份集中的数据库对象会被恢复。

恢复操作的性能可以通过提高创建多个并行连接的方式来同时恢复多个表和元数据。默认gprestore 使用1个连接,但是我们可以通过--jobs选项来为大的恢复过程增加处理进程数。

当一个操作完成后,gprestore会返回一个状态码。详见 Return Codes。

gprestore在恢复操作完成后可以发送邮件通知。可以指定发送邮件和接收邮件的配置文件。详见配置邮件通知。

Note:该工具在系统内部采用SSH连接执行各项操作任务。在大型MPP集群、云部署或每台主机部署了大量的 segment实例时,可能会遇到超过主机最大授权连接数限制的情况。此时需要考虑更新SSH配置参数MaxStartups 以提高该限制。更多关于SSH配置的选项,请参考您的Linux分发版的SSH文档。

#### 火火火

#### --timestamp YYYYMMDDHHMMSS

必须。指定一个用来恢复gpbackup备份集的时间戳。默认gprestore 会尝试从Master主机

的\$MASTER\_DATA\_DIRECTORY/backups/YYYYMMDD/YYYYMMDD 路径下定位时间戳对应的元数据文件,从每个Segment主机

的<seg\_dir>/backups/YYYYMMDD/YYYYMMDDhhmmss/路径下定位CSV数据文件。

### --backup-dir directory

可选。从指定的路径下找所有备份文件(元数据文件和数据文件)。 必须指定*directory* 为一个绝对路径(不能是相对路径)。如果没有指 定该选项,gprestore会尝试从Master主机的

\$MASTER\_DATA\_DIRECTORY/backups/YYYYMMDD/YYYYMMDDhr 路径下定位时间戳对应的元数据文件,从每个Segment主机

的<seg\_dir>/backups/YYYYMMDD/YYYYMMDDhhmmss/路径下定位 CSV数据文件。如果gpbackup操作指定了该选项,那么恢复时请同时 指定该选项。

该选项不能和--plugin-config选项一起使用。

#### --create-db

可选。在恢复数据库对象元数据之前先创建数据库。

通过复制空的系统标准数据库template0来创建指定的数据库。

### --data-only

可选。仅恢复gpbackup创建的表数据,不恢复创建数据库表的操作。 该选项假定数据库中已经存在对应的表。如果要恢复指定的集合,可 以通过指定选项来包含表/模式或排除表/模式。 指定--with-stats选项来 从备份集恢复表的统计信息。

备份集必须包含要被恢复的表数据。例如,gpbackup采用选项 -- metadata-only备份的备份集不包含表数据,不能用来恢复数据。如果要仅恢复数据库表,不恢复表里的数据,参见选项 --metadata-only。

### --debug

可选。显示操作期间的详细信息和调试日志。

#### --exclude-schema schema\_name

可选。指定恢复操作期间要排除的数据库模式。可以多次指定以排除多个模式。该选项不能和 --include-schema或表过滤选项 (例如: --include-table) 一起使用。

#### --exclude-table schema.table

可选。指定恢复操作期间要排除的表。指定的格式必须为<schemaname>.<table-name>。如果表名或模式名使用了非小写字母、数字或下划线,那改名字可以用双引号包裹。客户可以多次指定该选项。 如果表不在备份集中,恢复操作会失败。不能指定分区表的子分区。该选项不能与--exclude-schema或表过滤选项(例如:--include-table)一起使用。

#### --exclude-table-file file\_name

可选。指定恢复期间要排除的表的列表文件。文件为text格式,并且每行都必须定义一个单独的表,格式为 <schema-name>.<table-name>。文件不能包含多余的行。如果表名或模式名使用了非小写字母、数字或下划线,那改名字可以用双引号包裹。客户可以多次指定该选项。如果表不在备份集中,恢复操作会失败。不能指定分区表的子分区。

该选项不能与--exclude-schema或表过滤选项(例如:--include-table) 一起使用。

### --include-schema schema name

可选。指定要恢复的数据库模式。可以多次指定该选项以包含多个模式。如果指定了该选项,那定义的任何模式都必须在备份集中。没在-include-schema选项中指定的所有模式都会被忽略。

如果指定的模式在目标数据库中存在,该工具会产生一个错误然后继续操作。如果被恢复的表在数据库中存在,该工具会失败。

如果备份集在多个模式下存在依赖关系,那不能使用该选项。

更多信息请见过滤备份和恢复的内容。

#### --include-table schema.table

可选。指定一个要恢复的表。格式必须为<schema-name>.<table-name>。 如果任何表名或模式名使用非小写字母、数字或下划线,那么可以用双引号包裹名字。该选项可以多次指定。 不能指定分区表的子分区。

也可以指定一个有效的模式或视图。

如果指定了该选项,工具不会自动恢复依赖对象。必须指定需要的依赖对象。例如,如果恢复一个视图,那么必须也恢复它对应的表。如果恢复的表使用到了一个序列,那么也必须恢复该序列。这些对应的对象 也必须都在备份集中存在。

不能将该选项与--include-schema或表过滤选项 (例如--exclude-table-file) 一起使用。

#### --include-table-file file name

可选。指定一个包含恢复表名的列表文件。每行代表一张单独的表,格式必须为 <schema-name>.<table-name>。该文件不能包含多余的行。如果表名或模式名使用了非小写字母、数字或下划线,那改名字可以用双引号包裹。客户可以多次指定该选项。如果表不在备份集中,恢复操作会失败。不能指定分区表的子分区。

也可以指定一个有效的模式或视图。

如果指定了该选项,工具不会自动恢复依赖对象。必须指定需要的依赖对象。例如,如果恢复一个视图,那么必须也恢复它对应的表。如果恢复的表使用到了一个序列,那么也必须恢复该序列。这些对应的对象 也必须都在备份集中存在。

如果使用--include-table-file选项,gprestore 不会创建表的角色或用户集合。工具会恢复表的索引和规则。触发器也会被恢复,但是MPP目前不支持触发器。

更多信息请见过滤备份或恢复的内容。

### --jobs int

可选。指定恢复表数据和元数据的并行连接的数量。默认gprestore使用1个连接。增加该参数会提高数据恢复的速度。

Note: 如果使用gpbackup --single-data-file选项合并表备份为每个 Segment 节点1个单独的文件,那么不能在恢复过程中指定--jobs超过 1来进行并行恢复。

#### --metadata-only

可选。从一个gpbackup创建的备份集中创建数据库表,不恢复数据。 该选项 假设目标数据库中不存在这些表。要从备份集创建指定的表集 合,可以通过指定对应的选项来选择或 排除相应的表或模式。指定选 项--with-globals来恢复MPP数据库 系统对象。

备份集必须包含要恢复的表的DDL数据。例如,gpbackup选项--data-

only生成的备份集就不包含表的DDL信息。

创建完数据库表后再恢复表数据,请见选项 --data-only。

#### --on-error-continue

可选。指定该选项以在创建数据库元数据(例如表、对象或函数)或恢复数据过程中出现SQL错误时,忽略错误继续执行。如果有其他类型错误出现,工具会退出。工具会显示错误信息汇总并将错误信息写入到 gprestore日志文件,并继续恢复操作。

默认为第一次出现错误即退出。

### --plugin-config config-file\_location

指定gpbackup插件配置文件位置,它是一个YAML格式的文本文件。 该文件包含 gprestore在恢复操作期间使用的插件的配置信息。

如果备份数据库时指定了--plugin-config选项,那么在从该备份集恢复数据时也必须指定该配置文件。

该选项不能与--backup-dir同时使用。

有关存储插件应用的详细信息,请见 使用gpbackup存储插件。

#### --quiet

可选。禁止任何非告警、非错误日志输出。

### --redirect-db database\_name

可选。恢复到指定的*database\_name*,而不是备份时的默认数据库名。

#### --verbose

可选。显示恢复操作期间的详细日志信息。

#### --version

可选。打印版本号并退出。

### --with-globals

可选。恢复备份集合中的MPP数据库系统对象,和其余数据库对象。 详见备份或还原中包含的对象。

#### --with-stats

可选。从备份集合恢复查询计划统计信息。

#### --help

显示在线帮助信息。

### **Return Codes**

gprestore完成后会返回如下返回码之一

- 0-成功完成恢复。
- 1-恢复完成,没有严重错误。具体信息见日志文件。
- 2-恢复失败,有严重错误。具体信息见日志文件。

## 示例

创建demo数据库并恢复备份集合中指定时间的所有表名和模式名:

- \$ dropdb demo
- \$ gprestore --timestamp 20171103152558 --create-db

恢复备份集到"demo2"数据库而不是默认备份的"demo"数据库:

- \$ createdb demo2
- \$ gprestore --timestamp 20171103152558 --redirect-db demo2

恢复数据库全局对象和查询计划统计信息,而不是默认的全部数据:

\$ gprestore --timestamp 20171103152558 --create-db --with-globals --with-stats

使用创建在/home/gpadmin/backup下的文件恢复数据,创建8个并行连接:

\$ gprestore --backup-dir /home/gpadmin/backups/ --timestamp 20171103153156 --create-c

仅恢复备份集中的"wikipedia"模式:

- \$ dropdb demo
- \$ gprestore --include-schema wikipedia --backup-dir /home/gpadmin/backups/ --timestan

如果从增量备份集中恢复,所有需要的备份文件都必须对gprestore可用。例如,以下时间戳指定增量备份集。20170514054532时增量备份集合的全备份。

20170514054532 (full backup)

20170714095512

20170914081205

20171114064330

20180114051246

以下gprestore命令指定时间戳20121114064330。时间戳为20120714095512和20120914081205的增量备份和全量备份必须都能被恢复程序访问。

gprestore --timestamp 20121114064330 --redirect-db mystest --create-db

## 另见

gpbackup、使用gpbackup和gprestore并行备份和使用带有gpbackup和gprestore的S3存储插件

## gpscp

一次在多个主机之间复制文件。

### 概要

```
gpscp { -f hostfile_gpssh | -h hostname [-h hostname ...] }
    [-J character] [-v] [[user@]hostname:]file_to_copy [...]
    [[user@]hostname:]copy_to_path

    gpscp --?

gpscp --version
```

## 描述

gpscp工具允许用户使用SCP (安全复制) 在一个命令中将一个或多个文件 从指定的主机 复制到其他指定的主机。例如,用户可以同时将文件从MPP 数据库的Master主机复制到所有的 Segment主机。

要指定SCP会话中涉及的主机,请使用-f选项指定包含主机名列表的文件,或使用-h选项在命令行上命名单个主机名。至少需要一个主机名(-h)或主机文件(-f)。-J选项允许用户指定单个字符来替代 copy from和copy to的目标字符串中的*hostname*。如果未指定-J,则默认的替代字符是等号(=)。例如,以下命令将从本地主机将.bashrc复制到hostfile\_gpssh中指定的 所有主机上的/home/gpadmin:

```
gpscp -f hostfile_gpssh .bashrc =:/home/gpadmin
```

如果未在主机列表中指定用户名,或者在文件路径中指定了user@,

则gpscp将以当前登录用户的身份复制文件。要确定当前登录的用户,请执行 whoami命令。默认情况下,登录后gpscp将转到远程 主机上会话用户的\$HOME。为确保将文件复制到远程主机上的正确位置, 建议使用绝对路径。

在使用gpscp之前,用户必须在涉及SCP会话的主机之间建立可信的主机设置。 用户可以使用工具gpssh-exkeys更新已知的主机文件并在主机之间交换公钥 (如果尚未这样做的话)。

## 选项

### -f hostfile\_gpssh

指定包含将参与此SCP会话的主机列表的文件的名称。主机文件的语法是每行一台主机 ,如下所示:

<hostname>

#### -h hostname

指定将参与此SCP会话的单个主机名。用户可以多次使用-h选项来指 定多个主机名。

#### -J character

-J选项允许用户指定单个字符来替代copy from和copy to目标字符串中的*hostname*。 如果未指定-J,则默认的替代字符是等号(=)。

### -v (详细模式)

可选。除SCP命令输出外还报告其他消息。

#### file\_to\_copy

必需。要复制到其他主机(或文件位置)的文件的文件名(或绝对路径)。这可以是本地主机上的文件或另一台被提及主机上的文件。

#### copy\_to\_path

必需。用户希望将文件复制到指定主机上的路径。如果没有使用绝对路径,文件将被复制到相对于会话用户的\$HOME的位置。用户也可以使用等号'='(或使用-J选项)指定的另一个字符代替 hostname。这将

随后在所提供 的主机文件 (-f) 或-h选项中指定的每个主机名中进行替换。

### -? (帮助)

显示在线帮助。

#### --version

显示此工具的版本。

## 示例

将名为installer.tar的文件复制到文件hostfile\_gpssh 中的所有主机上的/路径下。

gpscp -f hostfile\_gpssh installer.tar =:/

将名为myfuncs.so的文件复制到名为sdw1和 sdw2的主机上的指定位置:

gpscp -h sdw1 -h sdw2 myfuncs.so =:/usr/local/mpp-db/lib

## 另见

gpssh, gpssh-exkeys

### ypstymsian

在Segment主机上安装MPP数据库。

## 概要

gpseginstall -f hostfile [-u gpdb\_admin\_user] [-p password]

[-c u|p|c|s|E|e|I|v]

gpseginstall --help

## 描述

gpseginstall工具提供了一种在主机列表文件中指定的Segment主机上快速安装 MPP数据库的简单方法。该工具不会在Master主机上安装或更新MPP数据库。用户可以以 root身份或以非root用户身份运行gpseginstall。gpseginstall不执行数据库初始化。有关初始化数据库的更多信息,请参阅gpinitsystem。

以root身份运行时,gpseginstall默认操作是添加一个系统用户(默认是gpadmin),创建口令(默认是changeme),并在部分主机上部署和安装MPP数据库。为此,gpseginstall从当前 用户的环境变量(\$GPHOME)中的安装路径下查找Master上的当前MPP数据库二进制文件。它将MPP数据库软件压缩成tar.gz文件并执行MD5 校验和以验证文件完整性。

然后,它将MPP数据库拷贝到Segment主机,安装(解压缩)MPP数据库,并将 MPP数据库安装的所有权更改为用户使用-u选项指定的系统用户。 最后,它将作为根用户以及用-u选项指定的系统用户在所有MPP数据库主机之间进行密钥交换。 gpseginstall还会执行用户限制检查并验证所有Segment上的MPP数据库的版本号。

如果以非root用户的身份运行gpseginstall,则gpseginstall仅会在Segment 主机上压缩、复制并 安装MPP数据库。它还可以为当前系统用户在MPP数据库主机之间交换密钥,并验证所有 Segment上的MPP数据库的版本号。

## 选项

#### -c | --commands option\_list

可选。这使用户可以自定义gpseginstall操作。请注意,如果不在gpseginstall语法中指定-c选项,则默认 执行这些命令选项。

- u:添加系统用户(仅限root使用)。
- p: 为系统用户更改口令(仅限root使用)。
- s:将MPP数据库压缩、复制、解压 (安装到) 所有Segment。
- c: 更改Segment主机上的MPP数据库安装 目录的所有权(仅限root使用)。
- E: 为root用户在MPP数据库的Master主机和Segment 主机之间进行密钥交换(仅限root使用)。
- e: 为非root系统用户在MPP数据库的Master主机和 Segment主机之间进行密钥交换。
- I: (仅适用于Linux) 将新用户添加到Segment主机时,检查并修改用户限制配置文件(/etc/security/limits.conf) (仅限root使用)。
- v:验证在所有Segment上运行的MPP数据库的版本。
   gpseginstall检查由\$GPHOME环境变量引用的MPP数据库安装的版本号以及到安装目录的符号链接。如果存在版本号不匹配或者无法找到MPP数据库安装目录,则会发生错误。

### -f | --file hostfile

必需。指定包含要安装MPP数据库的Segment主机的文件。 主机列表文件的每行必须包含一个主机名,并包含MPP系统中每个主 机的主机名。确保没有空行或多余的空格。 如果主机有多个配置的主机名,则每个主机只能使用一个主机名。例如:

sdw1-1

sdw2-1

sdw3-1

sdw4-1

如果可用,用户可以使用在MPP数据库主机间交换密钥时使用的同一个 gpssh-exkeys主机列表。

### -p | --password password

可选。使用-u选项设置用户指定用户的口令。默认口令是 changeme。此选项仅在以root用户身份运行gpsetinstall 时使用。 推荐的安全最佳实践:

- 一直使用口令。
- 不要使用默认口令。
- 安装后立即更改默认口令。

### -u | --user *user*

可选。指定系统用户。此用户也是MPP数据库管理用户。此用户拥有MPP数据库安装 并且管理数据库。这也是MPP数据库启动/初始化的用户。此选项仅在以root 身份运行gpseginstall时可用。默认是gpadmin。

### --help (帮助)

显示在线帮助。

## 示例

以root身份,在所有Segment上安装MPP数据库,将系统用户保留为 默认值 (gpadmin) 并将gpadmin的口令设置为 secret123:

# gpseginstall -f my\_host\_list\_file -p secret123

作为非root用户,将MPP数据库二进制文件压缩并复制到所有Segment (作为gpadmin):

\$ gpseginstall -f host\_file

以root身份,添加用户(gpadmin2),为该用户设置口令(secret1234),作为新用户在主机之间交换密钥,检查用户限制,验证版本号,但不要更改MPP二进制文件的所有权,压缩/复制/在Segment上安装MPP数据库,或者以root身份交换密钥。

\$ gpseginstall -f host\_file -u gpadmin2 -p secret1234 -c upelv

### 另见

gpinitsystem, gpssh-exkeys

## gpshrink

对现有的MPP数据库进行在线缩容。

### 概要

## 先决条件

- 用户作为MPP数据库超级用户(gpadmin)登录。
- 用户的Segment主机上有足够的磁盘空间来临时保存最大表的副本。
- 服务端gp\_shrink插件已加载 (默认加载)。
- 重新分布数据时,MPP数据库必须以生产模式运行。MPP数据库不能是 受限模式或 Master模式。不能指定gpstart的选项-R或者-m 启动MPP数据 库。

Note: 在gpshrink缩容已经开始并且未彻底完成之前gpexpand不能执行。 在gpshrink运行期间,以下工具不能被执行:

- gpbackup
- gprestore

当gpshrink进入表数据重分布阶段并且未完成时(无论gpshrink是否正在运行),以下工具不能被执行:

### • gpconfig

当gpshrink进入缩容清理阶段并且未完成时(无论gpshrink是否正在运行),以下工具不能被执行:

- gpbackup
- gpcheckcat
- gppkg
- gprestore

## 描述

gpshrink工具分三个阶段执行系统缩容:缩容标记、表重新分布、缩容清理。

在缩容标记阶段,gpshrink用一个输入文件运行,该文件指定新Segment 的数据目录、 *dbid*值和其他特征。用户可以手动创建输入文件,也可以在 交互式对话中 按照提示进行操作。

如果用户选择使用交互式对话创建输入文件,则可以选择指定包含缩容主机列表的文件。在提示输入信息时,如果用户的平台或命令shell限制可键入的主机名列表的长度,则可能不得不用-f指定主机。

除了标记Segment,缩容标记阶段还执行这些操作:

• 创建缩容模式*gpshrink*以存储缩容操作的状态,包括表的详细状态。

Note: 目前只支持从最大的content号开始的连续实例进行缩容(也就是从最晚加入集群的实例开始缩容),并且要保证剩余的实例在节点之间是平衡的。

在表数据重分布阶段,gpshrink会重分布表的数据到剩余的segment,使数据在剩余的segment实例之间平衡。

Note: 数据重新分布应该在低峰 (相对高峰期) 时段进行。重新分布可以在很长时间内分批次进行。

要开始重分布阶段,可以通过运行gpshrink并指定-d(运行时间周期)或-e(结束时间)选项,或者不指定任何选项。如果客户指定了结束时间或运行周期,工具会在缩容模式下重分布表,直到达到设定的结束时间或执行周期。如果没指定任何选项,工具会继续处理直到缩容模式的表全部完成重分布。每张表都会通过ALTER TABLE命令来在所有的节点包括新增加的segment实例上进行重分布,并设置表的分布策略为其原始策略。如果gpshrink完成所有表的重分布,它会显示成功信息并退出。

在缩容清理阶段,gpshrink会更新节点元信息、删除缩容实例、删除缩容模式。缩容清理在交互模式下运行,用户可以选择删除或保留缩容实例,可以选择备份元数据等。如果用户选择了删除缩容实例,gpshrink会停止缩容实例并删除相应的数据目录,否则保留缩容实例,但是缩容实例已经不属于数据库集群系统。

Note:该工具在系统内部采用SSH连接执行各项操作任务。在大型MPP集群、云部署或每台主机部署了大量的 segment实例时,可能会遇到超过主机最大授权连接数限制的情况。此时需要考虑更新SSH配置参数MaxStartups 以提高该限制。更多关于SSH配置的选项,请参考您的Linux分发版的SSH文档。

## 选项

### -a | --analyze

在缩容后运行ANALYZE更新表的统计信息,默认是不运行ANALYZE。

### -B batch\_size

在暂停一秒钟之前发送给给定主机的远程命令的批量大小。默认值 是16,有效值是1-128。

gpshrink工具会发出许多设置命令,这些命令可能会超出主机的已验

证 连接的最大阈值(由SSH守护进程配置中的MaxStartups定义)。该一秒钟 的暂停允许在gpshrink发出更多命令之前完成认证。默认值通常不需要改变。但是,如果gpshrink由于连接错误(例如'ssh\_exchange\_identification: Connection closed by remote host.') 而失败,则可能需要减少命令的最大数量。

#### -c | --clean

删除缩容模式。

### -d | --duration hh:mm:ss

缩容会话的持续时间。

### -e | --end 'YYYY-MM-DD hh:mm:ss'

缩容会话的结束日期及时间。

### -f | --hosts-file filename

指定包含用于系统缩容的新主机列表的文件的名称。文件的每一行都必须包含一个主机名。

该文件可以包含指定或不指定网络接口的主机名。gpshrink工具处理 这两种情况,如果原始节点配置了多个网络接口,则将接口号添加到 主机名的末尾。

Note: MPP数据库Segment主机的命名习惯是sdwN,其中sdw是前缀并且N是数字。例如,sdw1、sdw2等等。 对于具有多个接口的主机,约定是在主机名后面添加破折号 (-) 和数字。例如sdw1-1和sdw1-2是主机sdw1的两个接口名称。

### -i | --input input\_file

指定缩容配置文件的名称,其中为每个要添加的Segment包含一行, 格式为:

hostname:address:port:datadir:dbid:content:preferred\_role

### -n parallel\_processes

要同时重新分布的表的数量。有效值是1-96。

每个表重新分布过程都需要两个数据库连接:一个用于更改表,另一个用于在缩容方案中更新表的状态。 在增加-n之前,检查服务器配置参数max\_connections的当前值,并确保不超过最大连接限制。

### -r | --rollback

回滚失败的缩容设置操作。

### -s | --silent

以静默模式运行。在警告时,不提示确认就可继续。

### -S | --simple-progress

如果指定,gpshrink工具仅在MPP数据库表 gpshrink.shrinkage\_progress中记录最少的进度信息。该工具不在表 gpshrink.status\_detail中记录关系大小信息和状态信息。

指定此选项可通过减少写入gpshrink表的进度信息量来提高性能。

### -v | --verbose

详细调试输出。使用此选项,该工具将输出用于缩容数据库的所有DDL和DML。

#### --version

显示工具的版本号并退出。

-? | -h | --help

显示在线帮助。

## 示例

使用输入文件运行gpshrink以初始化新Segment,并在postgres数据库中创建缩容模式:

\$ gpshrink -i input\_file

运行gpshrink最长持续60小时,以将表重新分布给新的Segment:

\$ gpshrink -d 60:00:00

执行缩容清理:

\$ gpshrink -c

## gpssh

一次提供对多台主机的SSH访问。

## 概要

## 描述

gpssh工具允许用户使用SSH (安全shell) 一次在多台主机上运行bash shell命令。用户可以通过在命令行上指定一个命令来执行单个命令,也可以省略该命令以进入交互式命令行会话。

要指定参与SSH会话的主机,请使用-f选项指定包含主机名列表的文件,或使用-h选项在命令行上指明单个主机名。至少需要一个主机名(-h)或主

机文件 (-f)。请注意,当前主机默认*不*包含在会话中——要 包含本地主机,用户必须在会话中涉及的主机列表中明确声明它。

在使用gpssh之前,用户必须在涉及SSH会话的主机之间建立可信的主机设置。 用户可以使用工具gpssh-exkeys更新已知的主机文件并在主机之间交换 公钥(如果尚未这样做的话)。

如果用户没有在命令行上指定命令,gpssh将进入交互模式。在gpssh命令提示符(=>)处,用户可以像在常规bash终端命令行中那样输入命令,并且该命令将在会话涉及的所有主机上执行。要结束交互式会话,请按键盘上的CTRL+D或键入exit或quit。

如果主机文件中没有指定用户名,则gpssh将以当前登录的用户身份执行命令。要确定当前登录的用户,请执行whoami命令。默认情况下,登录后gpssh将转到远程主机上的会话用户的\$HOME。为确保所有远程主机上的命令都能正确执行,应始终输入绝对路径。

如果在使用gpssh时遇到网络超时问题,可以使用-d和 -t选项或者 在gpssh.conf文件中设置参数来控制 gpssh在验证初始ssh连接时使用的时 间。有关配置 文件的信息,请参阅gpssh配置文件gpssh配置文件。

## 选项

### bash\_command

在此会话中涉及的所有主机上执行的bash shell命令(可选择封闭在引号中)。如果未指定,则gpssh启动交互式会话。

### -d (延迟) seconds

可选。指定用ssh开始gpssh交互开始时等待的时间(以秒为单位)。默认值是0.05。此选项将覆盖gpssh.conf配置文件中指定的delaybeforesend的值。

增加此值可能会导致在gpssh启动过程中等待很长时间。

### -e (回显)

可选。以非交互模式运行时,回显传递给每个主机的命令及其结果输出。

### -f hostfile\_gpssh

指定包含将参与此SSH会话的主机列表的文件的名称。主机文件的语 法是每行一个主机。

#### -h hostname

指定将参与此SSH会话的单个主机名。用户可以多次使用-h 选项来指 定多个主机名。

-S

可选。如果指定,则在执行目标主机上的任何命令之前,gpssh将在 \$GPHOME环境变量指定的目录中引用文件mpp\_path.sh。

该选项对交互模式和单命令模式都有效。

### -t multiplier

可选。大于0的十进制数,它是gpssh在验证ssh 提示时使用的超时的倍数。默认值为1。此选项将覆盖gpssh.conf 配置文件中的prompt\_validation\_timeout值。

增加此值对gpssh启动过程影响不大。

### -v (详细模式)

可选。在非交互模式下运行时,除了输出命令外,还会报告其他消息。

#### --version

显示此工具的版本。

### -? (帮助)

显示在线帮助。

## gpssh配置文件

gpssh.conf文件包含的参数可让用户调整gpssh在验证初始 ssh连接时使用的时间。gpssh会话使用ssh 执行命令之前,这些参数会影响网络连接。该文件的位置由环境MASTER\_DATA\_DIRECTORY 指定。如果未定义环境变量或gpssh.conf文件不存在,gpssh将 使用默认值或使用-d和-t选项设置的

值。有关环境变量的信息,请参考MPP数据库参考手册。

gpssh.conf文件是由[gpssh]部分和参数组成的文本文件。#(井号)代表注释的开始。这是一个gpssh.conf文件的例子。

#### [gpssh]

delaybeforesend = 0.05
prompt\_validation\_timeout = 1.0
sync\_retries = 5

这些是gpssh.conf参数。

### delaybeforesend = seconds

指定用ssh启动gpssh交互时等待的时间(以秒为单位)。默认是 0.05。增加此值可能会导致gpssh启动过程中等待很长时间。-d选项 覆盖此参数。

### prompt\_validation\_timeout = multiplier

大于0的十进制数,它是gpssh在验证ssh 提示时使用的超时的倍数。增加这个值对gpssh启动过程影响不大。 默认值是1。-t选项覆盖此参数。

### sync\_retries = attempts

一个非负整数,指定gpssh尝试连接远程MPP数据库主机的最大次数。 缺省值为3。如果值为0,则如果初始连接尝试失败,gpssh将返回错误。 增加尝试次数也增加了重试次数之间的时间。此参数不能使用命令行选项进行配置。

-t选项也影响重试尝试之间的时间。

增加此值可以弥补网络性能下降或部分主机性能问题,例如繁重的 CPU或I/O负载。但是, 当无法建立连接时,增加的值也会增加错误被 返回时的延迟。

## 示例

与文件hostfile\_gpssh中列出的所有主机启动交互式SSH会话组:

```
$ gpssh -f hostfile_gpssh
```

在gpssh交互式命令提示符处,在此会话中涉及的所有主机上运行shell命令。

```
=> Is -a /data/primary/*
```

### 退出交互式会话:

```
=> exit
```

=> quit

使用名为sdw1和sdw2的主机启动非交互式SSH会话组,并将包含多个名为command\_file的命令的文件传递给gpssh:

```
$ gpssh -h sdw1 -h sdw2 -v -e < command_file
```

在主机sdw2和localhost上以非交互模式执行单个命令:

```
$ gpssh -h sdw2 -h localhost -v -e 'ls -a /data/primary/*'
$ gpssh -h sdw2 -h localhost -v -e 'echo $GPHOME'
$ gpssh -h sdw2 -h localhost -v -e 'ls -1 | wc -l'
```

### 另见

gpssh-exkeys, gpscp

## gpssh-exkeys

在主机之间交换SSH公钥。

## 概要

```
gpssh-exkeys -f hostfile_exkeys | - h hostname [-h hostname ...]

gpssh-exkeys -e hostfile_exkeys -x hostfile_gpexpand

gpssh-exkeys -?
```

gpssh-exkeys --version

## 描述

gpssh-exkeys工具在指定的主机名(或主机地址)之间交换SSH密钥。这允许 MPP主机和网络接口之间的SSH连接,而不需要口令提示。该工具用于最初将MPP数据库系统准备好用于无口令的SSH访问,以及在扩展MPP数据库系统时添加额外的ssh密钥。

要指定参与初始SSH密钥交换的主机,请使用-f选项指定包含主机名列表的

文件 (推荐) ,或使用-h选项在命令行上指出单个主机名。至少需要一个主机名 (-h) 或主机文件。请注意,本地主机默认包含在密钥交换中。

指定要添加到现有MPP数据库系统的新扩展主机,请使用-e和-x 选项。-e选项指定一个文件,其中包含系统中已有SSH密钥的现有主机列表。-x选项指定一个文件,其中包含需要参与SSH密钥交换的新主机列表。

密钥要作为当前登录的用户交换。用户应该执行两次密钥交换过程:一次作为root用户,一次作为gpadmin用户(拥有MPP数据库安装的用户)。MPP数据库管理要求在MPP数据库系统的所有主机上创建相同的非root用户,并且这些工具必须能够以该用户的身份连接到所有主机而无需口令。

### gpssh-exkeys工具使用以下步骤执行密钥交换:

- 为当前用户创建一个RSA标识密钥对(如果尚不存在)。该密钥对中的公钥被添加到当前用户的authorized\_keys文件中。
- 使用-h、-f、-e 以及-x选项指定的每台主机的主机密钥更新当前用户的 known\_hosts文件。
- 使用ssh连接到每个主机,并获取authorized\_keys、known\_hosts和id\_rsa.pub文件以设置无口令访问。
- 将从每个主机获取的id\_rsa.pub文件中的密钥添加到当前用户的authorized\_keys文件中。
- 使用新的主机信息(如果有)更新所有主机上的authorized\_keys、 known\_hosts和id\_rsa.pub文件。

### 选项

### -e hostfile\_exkeys

进行系统扩展时,这个选项指定包含*当前*MPP系统中每台主机 (Master、后备Master和Segment)的所有已配置主机名和主机地址 (接口名称)的文件的名称和位置,每行一个名称,其中没有空行或 额外的空格。这个文件中指定的主机不能出现在 -x使用的主机文件中。

### -f hostfile\_exkeys

指定包含MPP系统中每台主机(Master、后备Master和Segment)的 所有已配置主机名和主机地址(接口名称)的文件的名称和位置,每 行一个名称,其中没有空行或额外的空格。

#### -h hostname

指定将参与SSH密钥交换的单个主机名(或主机地址)。用户可以多次使用-h 选项来指定多个主机名和主机地址。

#### --version

显示此工具的版本。

### -x hostfile\_gpexpand

进行系统扩展时,这是一个文件的名称和位置,该文件包含要添加到 MPP系统的 每台新 Segment 主机的所有已配置主机名和主机地址(接口名称),每行一个名称,其中没有空行或额外的空格。这个文件中指定的主机不能出现在 -e使用的主机文件中。

### -? (帮助)

显示在线帮助。

### 示例

在文件hostfile\_exkeys中列出的所有主机名和地址之间交换SSH密钥:

\$ gpssh-exkeys -f hostfile\_exkeys

在主机sdw1、sdw2和sdw3之间交换SSH密钥:

\$ gpssh-exkeys -h sdw1 -h sdw2 -h sdw3

在现有主机sdw1、sdw2和sdw3,新主机sdw4和sdw5作为系统扩展操作的一部分:

```
$ cat hostfile_exkeys
         mdw
         mdw-1
         mdw-2
         smdw
         smdw-1
         smdw-2
         sdw1
         sdw1-1
         sdw1-2
         sdw2
         sdw2-1
         sdw2-2
         sdw3
         sdw3-1
         sdw3-2
         $ cat hostfile_gpexpand
         sdw4
         sdw4-1
         sdw4-2
         sdw5
         sdw5-1
         sdw5-2
         $ gpssh-exkeys -e hostfile_exkeys -x hostfile_gpexpand
```

## 另见

gpssh, gpscp

## gpstart

启动一个MPP数据库系统。

### 概要

## 描述

gpstart工具用于启动MPP数据库服务器进程。当用户启动一个MPP数据库系统时,用户实际上是同时启动了几个postgres数据库服务器监听器进程(Master和所有的Segment实例)。gpstart工具处理各个实例的启动。每个实例都是并行启动的。

管理员第一次运行gpstart时,该工具将在用户的主目录中创建一个名为 .gphostcache的主缓存文件。随后,该工具使用此主机列表更有效地启动 系统。 如果将新主机添加到系统中,则必须手动从gpadmin用户的主目录中删除此文件。 该工具将在下次启动时创建一个新的主机缓存文件。

作为启动过程的一部分,该工具会检查堆表checksum设置是否在集群上启用。如果堆表checksum启用情况在各个实例之间不同,MPP数据库不会启动,会返回一个错误信息。该验证选项可以通过指定选项 --skip-heap-checksum-validation来停用。更多关于堆表checksum的情况,请见MPP

数据库管理员手册中的启用高可用和数据一致性特性部分。

Note: 在启动MPP数据库系统之前,用户必须首先使用gpinitsystem初始化系统。 堆表checksum特性在系统初始化时配置启用和禁用,系统初始化后不能修改。

### 选项

-a

不要提示用户确认。

-B parallel\_processes

并行启动的Segment数。如果未指定,则该工具将启动最多64个并行进程,具体取决于需要启动多少个Segment实例。

-d master\_data\_directory

可选。Master主机的数据目录。如果未指定,则使用为\$MASTER\_DATA\_DIRECTORY设置的值。

-l logfile\_directory

写入日志文件的目录。默认为~/gpAdminLogs。

-m

-R

可选。仅启动Master实例,这可能对维护任务有用。该模式只允许连接到utility模式下的Master。例如:

PGOPTIONS='-c gp\_session\_role=utility' psql

此模式下不检查Master和Segment实例的堆表checksum设置一致性。

-q 以静默模式运行。命令输出不显示在屏幕上,但仍然写入日志文件。

以受限模式启动MPP数据库(只允许数据库超级用户连接)。

--skip-heap-checksum-validation

启动时指定该选项,该工具不检查Master和Segment之间的堆表 checksum一致性。默认情况下该设置在所有实例上是一样的,或者 启用或者禁用。

Warning: 不进行该项验证启动MPP数据库会导致数据丢失。 只有在必须要忽略堆表checksum验证错误以恢复数据或进行错误调试时,才能 使用该选项。

### -t timeout\_seconds

指定等待Segment实例启动的超时时间(秒)。如果某个Segment实例异常关闭(例如由于电源故障或终止其postgres数据库监听器进程),由于数据库恢复和验证过程,启动可能需要较长的时间。如果未指定,则默认超时时间为60秒。

**-v** 显示工具输出的详细状态,进度和错误消息。

-y 可选。不启动后备Master主机。默认是启动后备Master主机和同步过程。

- -?|-h|--help 显示在线帮助。
- --version 显示工具的版本。

### 示例

启动MPP数据库系统:

gpstart

以受限模式启动MPP数据库系统(仅允许超级用户连接):

gpstart -R

### 仅启动MPP主实例并以utility模式连接:

gpstart -m PGOPTIONS='-c gp\_session\_role=utility' psql

## 另见

gpstop, gpinitsystem

## gpstate

显示正在运行的MPP数据库系统的状态。

## 概要

```
gpstate [-d master_data_directory] [-B parallel_processes]
[-s | -b | -Q | -e] [-m | -c] [-p] [-i] [-f] [-v | -q] | -x
[-l log_directory]
```

gpstate -? | -h | --help

## 描述

gpstate工具显示有关正在运行的MPP数据库实例的信息。由于MPP数据库系统由跨多台机器的多个PostgreSQL数据库实例(Segment)组成,因此用户可能需要了解关于MPP数据库系统的额外信息。gpstate工具为MPP数据库系统提供了额外的状态信息,例如:

- 哪台Segment主机已被关闭?
- Master和Segment配置信息 (主机、数据目录等)。
- 系统使用的端口。
- 主Segment到其相应的镜像Segment的映射。

## 选项

-b (简要状态)

可选。显示MPP数据库系统状态的简要概述。这是默认选项。

-B parallel\_processes

并行检查的Segment数。如果未指定,则工具将根据需要检查多少个Segment实例启动最多60个并行进程。

- -c (显示主Segment到镜像Segment的映射) 可选。显示主Segment到镜像Segment的映射。
- -d master\_data\_directory

可选。Master的数据目录。如果未指定,则使用为\$MASTER\_DATA\_DIRECTORY设置的值。

-e (显示镜像状态出问题的Segment)

显示具有潜在问题的主要/镜像Master对的详细信息,例如1)活动 Segment正在Change Tracking模式下运行,表示Segment处于关闭 状态;2)活动Segment处于重新同步模式,这意味着它正在赶上对镜像的更改;3)一个Segment不是它的首选角色,例如在系统初始化时作为一个主Segment的现在作为一个镜像Segment,这意味着用户可能有一台或多台Segment主机的处理负载不平衡。

-f (显示后备Master的详情)

显示后备Master主机的详细信息(如果配置)。

-i (显示MPP数据库的版本)

显示每个实例的MPP数据库软件版本信息。

-l logfile\_directory

写入日志文件的目录。默认为~/gpAdminLogs。

-m (列出镜像)

可选。列出系统中的镜像Segment实例及其当前角色和同步状态。

-p(显示端口)

列出整个MPP数据库系统使用的端口号。

-q (没有屏幕输出)

可选。以静默模式运行。除了警告信息之外,屏幕上不显示命令输出。但是,这些信息仍然写入到日志文件中。

-Q (快速状态)

可选。在Master主机上的系统目录中检查Segment的状态。不直接向 Segment征询状态。

-s (详细状态)

可选。显示MPP数据库系统的详细状态信息.

-v (详细输出)

可选。显示错误消息并输出详细的状态和进度信息。

-x (expand)

Optional. Displays detailed information about the progress and state of a MPP system expansion.

-? | -h | --help (help)

显示在线帮助。

## 输出字段定义

以下输出字段由主机的gpstate -s报告:

Table 1. gpstate为Master给出的输出数据

输出数据	描述
Master host	Master的主机名
Master postgres process ID	主数据库侦听进程的PID
Master data directory	主数据目录的文件系统位置
Master port	Master上postgres数据库监听器进程的 端口
Master current role	dispatch = 常规操作模式
	utility = 维护模式
MPP array configuration type	Standard = 每台主机一个NIC
	Multi-Home = 每台主机多个NIC
MPP initsystem version	系统第一次初始化时的MPP数据库版本
MPP current version	MPP数据库当前的版本
Postgres version	MPP数据库基于的PostgreSQL版本
MPP mirroring status	物理镜像或无镜像
Master standby	后备Master的主机名
Standby master state	后备Master的状态:活跃或被动

### 以下输出字段由gpstate -s为每个Segment报告:

Table 2. gpstate为Segment给出的输出数据

输出数据	描述
Hostname	系统配置的主机名

输出数据	描述
Address	网络地址主机名 (NIC名称)
Datadir	Segment数据目录的文件系统位置
Port	Segment的postgres数据库监听器进程 的端口号
Current Role	Segment的当前角色: <i>Mirror</i> 或 <i>Primary</i>
Preferred Role	系统初始化时的角色:Mirror或 Primary
Mirror Status	主/镜像Segment对的状态:
	Synchronized = 两者上的数据都是最新
	Resynchronization = 数据当前正被从 一个拷贝到另一个
	Change Tracking = Segment停止并且 活动Segment正在记录更改
Change tracking data size	当处于 <i>Change Tracking</i> 模式中时,更改日志文件的尺寸(如果应用压缩,尺寸可能增长或者收缩)
Estimated total data to synchr onize	当处于 <i>Resynchronization</i> 模式中时, 剩下要同步的数据的估计尺寸
Data synchronized	当处于 <i>Resynchronization</i> 模式中时, 已经被同步的数据的估计尺寸
Estimated resync progress wi th mirror	当处于 <i>Resynchronization</i> 模式中时, 完成百分比的估计值
Estimated resync end time	当处于 <i>Resynchronization</i> 模式中时, 完成时间的估计

输出数据	描述
File postmaster.pid	postmaster.pid锁文件的状态: <i>Found</i> 或 <i>Missing</i>
PID from postmaster.pid file	postmaster.pid文件中找到的PID
Lock files in /tmp	为Segment的postgres会在 /tmp中创建 Segment端口锁文件(当Segment关闭 时会移除该文件)
Active PID	Segment的活动进程ID
Master reports status as	系统目录中报告的Segment状态: <i>Up</i> 或 <i>Down</i>
Database status	MPP数据库对进入请求的状态: <i>Up、Down</i> 或 <i>Suspended。 Suspended</i> 状态 意味着当 Segment正从一种状态转移到另一种状态时,数据库活动被临时暂停。

以下输出字段由gpstate -f为每个后备Master复制状态报告:

Table 3. gpstate对Master复制给出的输出数据

输出数据	描述
Standby address	后备Master的主机名
Standby data dir	后备Master的数据目录的文件系统位置
Standby port	后备Master的postgres数据库监听器进程的端口
Standby PID	后备Master的进程ID
Standby status	后备Master的状态: <i>Standby host pass ive</i>

输出数据	描述
WAL Sender State	预写式日志(WAL)流状态: streamin g、 startup、backup、catchup
Sync state	WAL发送者同步状态: <i>sync</i>
Sent Location	WAL发送者事务日志 (xlog) 记录发送 位置
Flush Location	WAL接收者xlog记录刷入位置
Replay Location	后备上xlog记录重放位置

## 示例

显示MPP数据库系统的详细信息:

gpstate -s

在Master主机系统目录中快速检查下游Segment:

gpstate -Q

显示关于镜像Segment实例的信息:

gpstate -m

显示关于后备Master配置的信息:

gpstate -f

显示MPP软件版本信息:

gpstate -i

## 另见

gpstart, gpexpandgplogfilter

## gpstop

停止或者重启MPP数据库系统。

概要

```
gpstop [-d master_data_directory] [-B parallel_processes]
        [-M smart | fast | immediate] [-t timeout_seconds] [-r] [-y] [-a]
        [-l logfile_directory] [-v | -q]

gpstop -m [-d master_data_directory] [-y] [-l logfile_directory] [-v | -q]

gpstop -u [-d master_data_directory] [-l logfile_directory] [-v | -q]

gpstop --host host_name [-d master_data_directory] [-l logfile_directory]
        [-t timeout_seconds] [-a] [-v | -q]

gpstop --version

gpstop -? | -h | --help
```

## 描述

gpstop工具用于停止构成MPP数据库系统的数据库服务器。当用户停止一个 MPP数据库系统时,用户实际上是一次停止几个postgres数据库服务器进程(Master和所有的Segment实例)。gpstop工具处理个别实例的关闭。每个实例 被并行地关闭。

默认情况下,如果有任何客户端连接到数据库,则不允许关闭MPP数据库。使用-M fast 选项来回滚所有正在进行的事务,并在关闭之前终止所有连接。如果有任何事务正在进行,则默认行为是 在关闭之前等待它们提交。

使用-u选项时,该工具会上传对主pg\_hba.conf文件所做的更改,或者在postgresql.conf文件中将*运行时*配置参数上载到服务中。请注意,任何活动的会话在重新连接到数据库之前都不会获取更改。

## 选项

-a

不要提示用户确认。

### -B parallel\_processes

并行停止的Segment数。如果未指定,则工具将启动最多64个并行进程,具体取决于需要停止多少个Segment实例。

### -d master\_data\_directory

可选。Master主机的数据目录。如果未指定,则使用为\$MASTER DATA DIRECTORY设置的值。

### --host host\_name

The utility shuts down the MPP Database segment instances on the specified host to allow maintenance on the host. Each primary segment instance on the host is shut down and the associated mirror segment instance is promoted to a primary segment if the mirror segment is on another host. Mirror segment instances on the host are shut down.

The segment instances are not shut down and the utility returns an error in these cases:

- Segment mirroring is not enabled for the system.
- The master or standby master is on the host.
- Both a primary segment instance and its mirror are on the host.

This option cannot be specified with the -m, -r, -u, or -y options. Note: The gprecoverseg utility restores segment instances. Run gprecoverseg commands to start the segments as mirrors and then to return the segments to their preferred role (primary segments).

### -l logfile\_directory

写入日志文件的目录。默认为~/gpAdminLogs。

-m

可选。关闭在维护模式下启动的MPP主实例。

#### -M fast

快速关闭。任何正在进行的事务都会中断并回滚。

#### -M immediate

立即关闭。任何正在进行的事务都会中止。

该模式杀死所有postgres进程,而不允许数据库服务器完成事务处理 或清理任何临时或进程内工作文件。

#### -M smart

智能关闭。如果存在活动连接,则此命令将失败并显示警告。这是默认的关闭模式。

-q

以静默模式运行。命令输出不显示在屏幕上,但仍然写入日志文件。

-r

关机完成后重新启动。

### -t timeout\_seconds

指定等待Segment实例关闭的超时阈值(以秒为单位)。如果 Segment实例没有在指定的秒数内关闭, gpstop将显示一条消息,指 示一个或多个Segment仍处于关闭过程中,并且 直到Segment实例停 止后才能重新启动MPP数据库。这个选项在gpstop 被执行且有非常大 的事务需要回滚的情况下非常有用。这些大型事务可能需要一分钟才 能回滚,并超过 600秒的默认超时时间。

-u

此选项将重新加载Master和Segment的pg\_hba.conf文件以及 postgresql.conf文件的运行时参数,但不会关闭数据库阵列。 编辑postgresql.conf或pg\_hba.conf 之后,使用此选项可使新的配置设置处于活动状态。请注意,这仅适用于设计为*运行时* 的配置参数。

-V

显示工具输出的详细状态,进度和错误消息。

**-y** 

不要停止后备Master进程。默认是停止后备Master。

-? | -h | --help

显示在线帮助。

--version

显示工具的版本。

## 示例

在智能模式下关闭MPP数据库系统:

gpstop

在快速模式下关闭MPP数据库系统:

gpstop -M fast

停止所有的Segment实例,然后重新启动系统:

gpstop -r

停止在维护模式下启动的Master实例:

gpstop -m

在进行配置更改后重新加载postgresql.conf和pg\_hba.conf 文件,但不要关闭MPP数据库阵列:

gpstop -u

## 另见

**gpstart** 

## gpsys1

显示有关操作系统的信息。

## 概要

```
gpsys1 [ -a | -m | -p ]
```

gpsys1 -?

gpsys1 --version

## 描述

gpsys1显示当前主机的平台和安装的内存(以字节为单位)。例如:

linux 1073741824

## 选项

-a (显示所有)

显示当前主机的平台和内存信息。这是默认选项。

-m (仅显示内存)

以字节为单位显示安装的系统内存。

-p (仅显示平台)

显示操作系统平台。可以是linux、darwin或sunos5。

-? (帮助)

显示在线帮助。

--version

显示此工具的版本。

## 示例

显示有关当前主机操作系统的信息:

gpsys1

## 另见

gpcheckperf

## iolimit

内核扩展,支持资源限制。

## 概要

iolimit扩展使内核支持cgroup模式和throttle模式的资源限制。cgroup模式

基于linux的cgroup实现,原理是把当前进程加入到控制组实现资源隔离。 throttle模式基于内部限速,原理是限制当前进程读写buffer的速度。

iolimit支持限制数据读写和日志读写速度,也可以单独限制数据或日志。 iolimit支持限制读速度、写速度或读写速度。通过GUC参数配置。

不建议用户直接操作iolimit, 一般是集成到工具内使用。

## GUC参数

### iolimit.cgroup

cgroup模式使用。设置控制组,把当前进程加入到iolimit.cgroup指定的控制组。如果iolimit.cgroup设置为空不执行任何操作。

#### iolimit.throttle

throtle模式使用。功能总开关,默认关闭。

### iolimit.throttle\_data

throtle模式使用。数据限流开关,默认打开。

### iolimit.throttle xlog

throtle模式使用。日志限流开关,默认打开。

### iolimit.throttle mode

throtle模式使用。限流模式,默认是rw(读和写),支持r、w和rw。

### iolimit.throttle\_bps

throtle模式使用。速度设置,默认是10MB/s。

## 示例

### cgroup模式,把当前进程加入到expand:

```
$ set iolimit.cgroup to 'gpdb_ext/expand';
```

以throttle模式限制当前进程的日志写速度为2MB/s:

```
$ set iolimit.throttle_bps to '2MB/s';
$ set iolimit.throttle_mode to 'w';
$ set iolimit.throttle_data to off;
$ set iolimit.throttle_xlog to on;
$ set iolimit.throttle to on;
```

throttle模式,限制当前进程的日志和数据整体的读写速度是15MB/s:

```
$ set iolimit.throttle_bps to '15MB/s';
$ set iolimit.throttle_mode to 'rw';
$ set iolimit.throttle_data to on;
$ set iolimit.throttle_xlog to on;
$ set iolimit.throttle to on;
```

throttle模式,关闭当前进程的限制:

\$ set iolimit.throttle to off;

# 客户端工具参考

描述MPP数据库随附的命令行客户端工具。

MPP数据库使用标准的PostgreSQL客户端程序,并提供其他客户端工具来管理分布式MPP数据库DBMS。 MPP数据库客户端工具位于\$GPHOME/bin.中。

以下是MPP数据库客户端工具。

- clusterdb
- createdb
- createlang
- createuser
- dropdb
- droplang
- dropuser
- pg\_config
- pg\_dump
- pg\_dumpall
- pg\_restore
- psql
- reindexdb
- vacuumdb
- 客户端工具摘要

Parent topic: MPP数据库工具手册

# 客户端工具摘要

# clusterdb

重新群集之前用CLUSTER群集过的表。

```
clusterdb [connection-option ...] [--verbose | -v] [--table | -t table] [[--dbname | -d] dbname clusterdb [connection-option ...] [--all | -a] [--verbose | -v] clusterdb -? | --help clusterdb -V | --version
```

更多信息请参考clusterdb。

# createdb

创建一个新的数据库。

```
createdb [connection-option ...] [option ...] [dbname ['description']]

createdb -? | --help

createdb -V | --version
```

更多信息请参考createdb。

# createlang

为数据库定义新的过程语言。

```
createlang [connection_option ...] [-e] langname [[-d] dbname]

createlang [connection-option ...] -I dbname

createlang -? | --help

createlang -V | --version
```

更多信息请参考createlang。

# createuser

创建一个新的数据库角色。

```
createuser [connection-option ...] [role_attribute ...] [-e] role_name

createuser -? | --help

createuser -V | --version
```

更多信息请参考createuser。

# dropdb

删除数据库。

```
dropdb [connection-option ...] [-e] [-i] dbname

dropdb -? | --help

dropdb -V | --version
```

更多信息请参考dropdb。

# droplang

删除过程语言。

```
droplang [connection-option ...] [-e] langname [[-d] dbname]

droplang [connection-option ...] [-e] -l dbname

droplang -? | --help

droplang -V | --version
```

更多信息请参考droplang。

# dropuser

删除数据库角色。

```
dropuser [connection-option ...] [-e] [-i] role_name

dropuser -? | --help

dropuser -V | --version
```

更多信息请参考dropuser。

# pg\_config

检索有关MPP数据库已安装版本的信息。

```
pg_config [option ...]

pg_config -? | --help

pg_config --version
```

更多信息请参考pg\_config。

# pg\_dump

Extracts a database into a single script file or other archive file.

```
pg_dump [connection-option ...] [dump_option ...] [dbname]

pg_dump -? | --help

pg_dump -V | --version
```

更多信息请参考pg\_dump。

# pg\_dumpall

将MPP数据库系统中的所有数据库提取到单个脚本文件或其他归档文件

中。

```
pg_dumpall [connection-option ...] [dump_option ...]

pg_dumpall -? | --help

pg_dumpall -V | --version
```

更多信息请参考pg\_dumpall。

# pg\_restore

从pg\_dump创建的存档文件中恢复数据库。

```
pg_restore [connection-option ...] [restore_option ...] filename

pg_restore -? | --help

pg_restore -V | --version
```

更多信息请参考pg\_restore。

# psql

MPP数据库的交互式命令行接口

```
psql [option ...] [dbname [username]]
```

更多信息请参考psql。

# reindexdb

重建数据库中的索引。

```
reindexdb [connection-option ...] [--table | -t table ]
        [--index | -i index ] [dbname]

reindexdb [connection-option ...] --all | -a

reindexdb [connection-option ...] --system | -s [dbname]

reindexdb -? | --help

reindexdb -V | --version
```

更多信息请参考reindexdb。

# vacuumdb

垃圾收集并分析数据库。

```
vacuumdb [connection-option...] [--full | -f] [--freeze | -F] [--verbose | -v]
        [--analyze | -z] [--analyze-only | -Z] [--table | -t table [( column [,...] )] ] [dbname]

vacuumdb [connection-option...] [--all | -a] [--full | -f] [-F]
        [--verbose | -v] [--analyze | -z]
        [--analyze-only | -Z]

vacuumdb -? | --help

vacuumdb -V | --version
```

更多信息请参考vacuumdb。

# clusterdb

重新群集之前用CLUSTER群集过的表。

# 概要

```
clusterdb [connection-option ...] [--verbose | -v] [--table | -t table] [[--dbname | -d] dbname clusterdb [connection-option ...] [--all | -a] [--verbose | -v] clusterdb -? | --help clusterdb -V | --version
```

# 描述

对表进行群集意味着根据索引对磁盘上的表进行物理重新排序,以便索引扫描操作可以按某种顺序访问磁盘上的数据,从而提高使用该索引的查询的索引查找性能。

clusterdb工具将在数据库中找到以前已使用CLUSTER SQL命令进行群集的任何表,并将它们再次群集在最后使用的同一索引上。 从未群集过的表

#### 不受影响。

clusterdb是SQL命令CLUSTER的包装。 尽管MPP数据库支持以这种方式对表进行群集,但是不建议这样做,因为CLUSTER操作本身非常慢。

如果确实需要以这种方式对表进行排序以提高查询性能,请使用CREATE TABLE AS语句对磁盘上的表进行重新排序,而不要使用CLUSTER。 如果您以这种方式"群集"表,那么clusterdb将不相关。

# 选项

-a | --all

群集所有数据库。

[-d] *dbname* | [--dbname=]*dbname* 

指定要群集的数据库的名称。 如果未指定,则从环境变量PGDATABASE读取数据库名称。 如果未设置,则使用为连接指定的用户名。

-e | --echo

输出clusterdb生成并发送到服务器的命令。

-q | --quiet

不显示响应。

-t table | --table=table

仅群集命名表。可以通过编写多个-t开关来群集多个表。

-v | --verbose

在外理过程中打印详细信息。

-V | --version

打印clusterdb版本并退出。

-? | --help

显示有关clusterdb命令行参数的帮助,然后退出。

### 连接选项

#### -h host | --host=host

运行MPP master数据库服务器的计算机的主机名。 如果未指定,则从环境变量PGHOST读取或默认为localhost。

# -p port | --port=port

MPP master数据库服务器正在侦听连接的TCP端口。 如果未指定,则从环境变量PGPORT读取或默认为5432。

### -U username | --username=username

要用作连接的数据库角色名称。 如果未指定,则从环境变量 PGUSER读取或默认为当前系统角色名称。

## -w | --no-password

不发出密码提示。如果服务器要求密码验证,而其他方式(例如.pgpass文件)无法使用密码,则连接尝试将失败。此选项在没有用户输入密码的批处理作业和脚本中很有用。

### -W | --password

强制输入密码提示。

#### --maintenance-db=dbname

指定要连接以发现还应群集哪些其他数据库的数据库的名称。 如果未指定,将使用postgres数据库,如果不存在,将使用template1。

# 示例

#### 群集数据库test:

clusterdb test

在名为xyzzy的数据库中群集单个表foo:

clusterdb --table foo xyzzyb

# 另见

#### **CLUSTER**

# createdb

创建一个新的数据库。

# 概要

```
createdb [connection-option ...] [option ...] [dbname ['description']]
createdb -? | --help
```

createdb -V | --version

# 描述

createdb在MPP数据库系统中创建一个新数据库。

通常,执行此命令的数据库用户将成为新数据库的所有者。 但是,如果执行用户具有适当的特权,则可以通过-O选项指定其他所有者。

createdb是SQL命令CREATE DATABASE的包装。

# 选项

#### dbname

要创建的数据库的名称。 该名称在MPP系统中的所有其他数据库中必须唯一。 如果未指定,则从环境变量PGDATABASE中读取,然后从PGUSER中读取,或者默认为当前系统用户。

### description

与新创建的数据库关联的注释。 包含空格的说明必须用引号引起来。

-D tablespace | --tablespace=tablespace

指定数据库的默认表空间。(此名称作为双引号标识符处理。)

-e echo

回显createdb生成并发送到服务器的命令。

-E encoding | --encoding encoding

在新数据库中使用的字符集编码。指定字符串常量(例如'UTF8'),整数编码数字或DEFAULT以使用默认编码。 有关支持的字符集的信息,请参阅《MPP数据库参考手册》。

-l locale | --locale locale

指定此数据库中要使用的语言环境。 这等效于同时指定--lc-collate和--lc-ctype。

--lc-collate locale

指定要在此数据库中使用的LC\_COLLATE设置。

--lc-ctype locale

指定要在此数据库中使用的LC\_CTYPE设置。

--maintenance-db=dbname

指定在创建新数据库时要连接的数据库的名称。 如果未指定,将使用postgres数据库。 如果不存在(或者它是正在创建的新数据库的名称),将使用template1。

-O owner | --owner=owner

拥有新数据库的数据库用户的名称。 默认为执行此命令的用户。 (此名称作为双引号标识符处理。)

-T template | --template=template

从中创建新数据库的模板的名称。默认为template1。 (此名称作为双引号标识符处理。)

#### -V | --version

打印createdb版本并退出。

### -? | --help

显示有关createdb命令行参数的帮助,然后退出。

选项-D,-I,-E,-O和-T对应于基础SQL命令CREATE DATABASE的选项。 有关它们的更多信息,请参见CREATE DATABASE。 for more information about them.

#### 连接选项

-h host | --host=host

运行MPP master数据库服务器的计算机的主机名。 如果未指定,则从环境变量PGHOST读取或默认为localhost。

-p port | --port=port

MPP master数据库服务器正在侦听连接的TCP端口。 如果未指定,则从环境变量PGPORT读取或默认为5432。

-U username | --username=username

要用作连接的数据库角色名称。 如果未指定,则从环境变量 PGUSER读取或默认为当前系统角色名称。

-w | --no-password

不发出密码提示。如果服务器要求密码验证,而其他方式(例如.pgpass文件)无法使用密码,则连接尝试将失败。此选项在没有用户输入密码的批处理作业和脚本中很有用。

-W | --password

强制输入密码提示。

# 示例

使用默认选项创建数据库test:

#### createdb test

使用主机gpmaster上的MPP master (端口54321) 并使用LATIN1编码方案来创建数据库demo:

createdb -p 54321 -h gpmaster -E LATIN1 demo

# 另见

**CREATE DATABASE**, dropdb

# createlang

为数据库定义新的过程语言。

# 概要

```
createlang [connection_option ...] [-e] langname [[-d] dbname]
createlang [connection-option ...] -I dbname
createlang -? | --help
createlang -V | --version
```

# 描述

createlang工具将新的过程语言添加到数据库。 createlang是SQL命令CREATE EXTENSION的包装。

Note: createlang已弃用,将来的发行版中可能会将其删除。 建议直接使用CREATE EXTENSION命令。

标准MPP数据库分发中包含的过程语言包是:

- PL/pgSQL
- PL/Perl
- PL/Python

默认情况下,PL/pgSQL语言已在所有数据库中注册。

MPP数据库还具有用于PL/Java和PL/R的语言处理程序,但是MPP数据库尚未预安装这些语言。有关更多信息,请参阅文档中的MPP PL/Java语言扩展和MPP PL/R语言扩展部分。

# 选项

### langname

指定要安装的过程语言的名称。(此名称为小写。)

### [-d] dbname | [--dbname=]dbname

指定应向其中添加语言的数据库。 默认设置是使用PGDATABASE环境变量设置,或与当前系统用户相同的名称。

#### -e | --echo

回显createlang生成并发送到服务器的命令。

### -I dbname | --list dbname

显示目标数据库中已经安装的语言的列表。

#### -V | --version

打印createlang版本并退出。

### -? | --help

显示有关createlang命令行参数的帮助,然后退出。

#### 连接选项

### -h host | --host=host

运行MPP master数据库服务器的计算机的主机名。 如果未指定,则从环境变量PGHOST读取或默认为localhost。

### -p port | --port=port

MPP主数据库服务器正在侦听连接的TCP端口。 如果未指定,则从环境变量PGPORT读取或默认为5432。

### -U username | --username=username

要用作连接的数据库角色名称。 如果未指定,则从环境变量 PGUSER读取或默认为当前系统角色名称。

### -w | --no-password

不发出密码提示。如果服务器要求密码验证,而其他方式(例如.pgpass文件)无法使用密码,则连接尝试将失败。 此选项在没有用户输入密码的批处理作业和脚本中很有用。

### -W | --password

强制输入密码提示。

# 示例

## 要将语言plperl安装到数据库mytestdb中:

createlang plperl mytestdb

# 另见

droplang, CREATE EXTENSION, CREATE LANGUAGE, DROP LANGUAGE

# createuser

创建一个新的数据库角色。

# 概要

```
createuser [connection-option ...] [role_attribute ...] [-e] role_name

createuser -? | --help

createuser -V | --version
```

# 描述

createuser创建一个新的MPP数据库角色。 您必须是超级用户或具有CREATEROLE权限才能创建新角色。 您必须以超级用户身份连接到数据库才能创建新的超级用户。

超级用户可以绕过数据库中的所有访问权限检查,因此不应轻易授予超级用户特权。

createuser是SQL命令CREATE ROLE的包装。

# 选项

#### role\_name

要创建的角色的名称。 此名称必须不同于该MPP数据库安装中的所有现有角色。

-c *number* | --connection-limit=*number* 

设置新角色的最大连接数。默认设置为无限制。

-d | --createdb

新角色将被允许创建数据库。

-D | --no-createdb

不允许新角色创建数据库。这是默认值。

-e | --echo

回显createuser生成并发送到服务器的命令。

-E | --encrypted

加密存储在数据库中的角色密码。 如果未指定,则使用默认密码行为。

-i | --inherit

新角色将自动继承其成员角色的特权。这是默认值。

-I | --no-inherit

新角色将不会自动继承其成员角色的特权。

--interactive

如果在命令行上未指定用户名,则提示输入用户名,并提示在命令行中未指定选项-d/-D,-r/-R,-s/-S中的任何一个。

#### -l | --login

新角色将被允许登录MPP数据库。这是默认值。

#### -L | --no-login

新角色将不允许登录 (组级别角色)。

### -N | --unencrypted

不加密存储在数据库中的角色密码。如果未指定,则使用默认密码行为。

### -P | --pwprompt

如果提供,则createuser将提示输入新角色的密码。 如果您不打算使用密码身份验证,则没有必要。

#### -r | --createrole

新角色将被允许创建新角色(CREATEROLE特权)。

#### -R | --no-createrole

新角色将不允许创建新角色。这是默认值。

#### -s | --superuser

新角色将是超级用户。

### -S | --no-superuser

新角色将不是超级用户。这是默认值。

### -V | --version

打印createuser版本并退出。

### --replication

新用户将具有REPLICATION特权,有关CREATE ROLE的文档中对此进行了更详细的描述。

### --no-replication

新用户将没有REPLICATION特权,有关CREATE ROLE的文档中对此进行了更详细的描述。

### -? | --help

显示有关createuser命令行参数的帮助,然后退出。

### 连接选项

### -h host | --host=host

运行MPP master数据库服务器的计算机的主机名。 如果未指定,则从环境变量PGHOST读取或默认为localhost。

### -p port | --port=port

MPP master数据库服务器正在侦听连接的TCP端口。 如果未指定,则从环境变量PGPORT读取或默认为5432。

#### -U username | --username=username

要用作连接的数据库角色名称。 如果未指定,则从环境变量 PGUSER读取或默认为当前系统角色名称。

### -w | --no-password

不发出密码提示。如果服务器要求密码验证,而其他方式(例如.pgpass文件)无法使用密码,则连接尝试将失败。此选项在没有用户输入密码的批处理作业和脚本中很有用。

### -W | --password

强制输入密码提示。

# 示例

要在默认数据库服务器上创建角色joe:

\$ createuser joe

要在默认数据库服务器上创建角色joe并提示一些其他属性:

\$ createuser --interactive joe

Shall the new role be a superuser? (y/n) n

Shall the new role be allowed to create databases? (y/n) n

Shall the new role be allowed to create more new roles? (y/n) n

**CREATE ROLE** 

要使用连接选项来创建相同的角色joe,并显式指定属性,然后查看基本命令:

createuser -h masterhost -p 54321 -S -D -R -e joe CREATE ROLE joe NOSUPERUSER NOCREATEDB NOCREATEROLE INHERIT LOGIN; CREATE ROLE

要将角色joe创建为超级用户,并立即分配密码admin123:

createuser -P -s -e joe

Enter password for new role: admin123

Enter it again: admin123

CREATE ROLE joe PASSWORD 'admin123' SUPERUSER CREATEDB

**CREATEROLE INHERIT LOGIN;** 

**CREATE ROLE** 

在上面的示例中,键入时实际上并未回显新密码,但是为了清楚起见,我们显示了键入的内容。 但是,密码将显示在echoed命令中,如使用-e选项所示。

# 另见

**CREATE ROLE**, dropuser

# dropdb

删除数据库。

# 概要

```
dropdb [connection-option ...] [-e] [-i] dbname
```

dropdb -? | --help

dropdb -V | --version

# 描述

dropdb销毁现有数据库。 执行此命令的用户必须是超级用户或要删除的数据库的所有者。

dropdb是SQL命令DROP DATABASE的包装。 有关DROP DATABASE的信息,请参阅《MPP数据库参考手册》。

# 选项

dbname

要删除的数据库的名称。

-e | --echo

回显dropdb生成并发送到服务器的命令。

-i | --interactive

在执行破坏性操作之前发出验证提示。

-V | --version

打印dropdb版本并退出。

--if-exists

如果数据库不存在,请不要报错。在这种情况下发出通知。

-? | --help

显示有关dropdb命令行参数的帮助,然后退出。

#### 连接选项

### -h host | --host=host

运行MPP master数据库服务器的计算机的主机名。 如果未指定,则从环境变量PGHOST读取或默认为localhost。

#### -p port | --port=port

MPP master数据库服务器正在侦听连接的TCP端口。 如果未指定,则从环境变量PGPORT读取或默认为5432。

### -U username | --username=username

要用作连接的数据库角色名称。 如果未指定,则从环境变量 PGUSER读取或默认为当前系统角色名称。

#### -w | --no-password

不发出密码提示。如果服务器要求密码验证,而其他方式(例如.pgpass文件)无法使用密码,则连接尝试将失败。此选项在没有用户输入密码的批处理作业和脚本中很有用。

### -W | --password

强制输入密码提示。

#### --maintenance-db=dbname

指定要删除目标数据库要连接的数据库的名称。 如果未指定,将使用postgres数据库。 如果不存在(或者它是要删除的数据库的名称),则将使用template1。

# 示例

要使用默认连接参数销毁名为demo的数据库:

#### dropdb demo

要使用连接选项(带有验证)并查看基础命令来删除名为demo的数据库: To destroy the database named demo using connection options, with verification, and a peek at the underlying command: dropdb -p 54321 -h masterhost -i -e demo Database "demo" will be permanently deleted. Are you sure? (y/n) y DROP DATABASE "demo" DROP DATABASE

# 另见

createdb, DROP DATABASE

# droplang

删除过程语言。

# 概要

```
droplang [connection-option ...] [-e] langname [[-d] dbname]
droplang [connection-option ...] [-e] -l dbname
droplang -? | --help
droplang -V | --version
```

# 描述

droplang从数据库中删除现有的过程语言。

droplang是SQL命令DROP EXTENSION的包装。

# 选项

#### langname

指定要删除的过程语言的名称。(此名称为小写。)

### [-d] dbname | [--dbname=]dbname

指定应从哪个数据库中删除语言。 默认设置是使用PGDATABASE环境变量设置,或与当前系统用户相同的名称。

-e | --echo

回显droplang生成并发送到服务器的命令。

-l | --list

显示目标数据库中已经安装的语言的列表。

-V | --version

打印droplang版本并退出。

-? | --help

显示有关droplang命令行参数的帮助,然后退出。

### 连接选项

-h host | --host=host

运行MPP master数据库服务器的计算机的主机名。 如果未指定,则从环境变量PGHOST读取或默认为localhost。

-p port | --port=port

MPP主数据库服务器正在侦听连接的TCP端口。 如果未指定,则从环境变量PGPORT读取或默认为5432。

-U username | --username=username

要用作连接的数据库角色名称。 如果未指定,则从环境变

量PGUSER读取或默认为当前系统角色名称。

### -w | --no-password

不发出密码提示。如果服务器要求密码验证,而其他方式(例如.pgpass文件)无法使用密码,则连接尝试将失败。 此选项在没有用户输入密码的批处理作业和脚本中很有用。

### -W | --password

强制输入密码提示。

# 示例

要从mydatabase数据库中删除语言pltcl:

droplang pltcl mydatabase

# 另见

createlang, DROP EXTENSION, DROP LANGUAGE

# dropuser

删除数据库角色。

# 概要

dropuser [connection-option ...] [-e] [-i] role\_name

dropuser -? | --help

dropuser -V | --version

# 描述

dropuser从MPP数据库中删除现有角色。 只有超级用户和具有CREATEROLE特权的用户才能删除角色。 要删除超级用户角色,您必须自己是超级用户。

dropuser是SQL命令DROP ROLE的包装。

# 选项

#### role\_name

要删除的角色的名称。 如果未在命令行上指定,则将提示您输入名称,并且使用了-i/--interactive选项。

#### -e | --echo

回显dropuser生成并发送到服务器的命令。

## -i | --interactive

在实际删除角色之前提示确认,如果在命令行上未指定角色名称,则提示输入角色名称。

#### --if-exists

如果用户不存在,请不要报错。在这种情况下发出通知。

# -V | --version

打印dropuser版本并退出。

#### -? | --help

显示有关dropuser命令行参数的帮助,然后退出。

#### 连接选项

#### -h host | --host=host

运行MPP master数据库服务器的计算机的主机名。 如果未指定,则从环境变量PGHOST读取或默认为localhost。

### -p port | --port=port

MPP master数据库服务器正在侦听连接的TCP端口。 如果未指定,则从环境变量PGPORT读取或默认为5432。

-U username | --username=username

要用作连接的数据库角色名称。 如果未指定,则从环境变量 PGUSER读取或默认为当前系统角色名称。

### -w | --no-password

不发出密码提示。如果服务器要求密码验证,而其他方式(例如.pgpass文件)无法使用密码,则连接尝试将失败。此选项在没有用户输入密码的批处理作业和脚本中很有用。

### -W | --password

强制输入密码提示。

# 示例

要使用默认连接选项删除角色joe:

dropuser joe DROP ROLE

要使用连接选项 (带有验证) 并查看基本命令来删除角色joe:

dropuser -p 54321 -h masterhost -i -e joe Role "joe" will be permanently removed. Are you sure? (y/n) y DROP ROLE "joe" DROP ROLE

# 另见

createuser, DROP ROLE

# pg\_config

检索有关MPP数据库已安装版本的信息。

# 概要

```
pg_config [option ...]

pg_config -? | --help

pg_config --version
```

# 描述

pg\_config工具显示当前安装的MPP数据库版本的配置参数。 例如,它打算由希望与MPP数据库进行接口连接的软件包使用,以便于查找所需的头文件和库。 请注意,由pg\_config打印出的信息仅适用于MPP数据库master。

如果给出多个选项,则按该顺序打印信息,每行打印一项。 如果未给出任何选项,则会打印所有可用信息以及标签。

# 选项

#### --bindir

打印用户可执行文件的位置。 例如,使用它来查找psql程序。 通常这也是pg\_config程序所在的位置。

#### --docdir

打印文档文件的位置。

#### --includedir

打印客户端接口的C头文件的位置。

### --pkgincludedir

打印其他C头文件的位置。

#### --includedir-server

打印C头文件的位置以进行服务器编程。

#### --libdir

打印对象代码库的位置。

#### --pkglibdir

打印动态可加载模块的位置,或服务器在其中搜索它们的位置。 (其他与体系结构相关的数据文件也可以安装在此目录中。)

#### --localedir

打印语言环境支持文件的位置。

#### --mandir

打印手册页的位置。

#### --sharedir

打印与体系结构无关的支持文件的位置。

#### --sysconfdir

打印系统范围的配置文件的位置。

#### --pgxs

打印扩展makefile的位置。

### --configure

打印为MPP数据库配置构建时为配置脚本提供的选项。

--CC

打印用于构建MPP数据库的CC变量的值。 这显示了使用的C编译器。

### --cppflags

打印用于构建MPP数据库的CPPFLAGS变量的值。 这显示了预处理时需要的C编译器开关。

#### --cflags

打印用于构建MPP数据库的CFLAGS变量的值。 这显示了C编译器开关。

### --cflags\_sl

打印用于构建MPP数据库的CFLAGS\_SL变量的值。 这显示了用于构建共享库的其他C编译器开关。

#### --Idflags

打印用于构建MPP数据库的LDFLAGS变量的值。 这显示了链接器开关。

#### --Idflags ex

打印用于构建MPP数据库的LDFLAGS\_EX变量的值。 这显示了仅用于生成可执行文件的链接器开关。

#### --Idflags sl

打印用于构建MPP数据库的LDFLAGS\_SL变量的值。 这显示了仅用于构建共享库的链接器开关。

#### --libs

打印用于构建MPP数据库的LIBS变量的值。 它通常包含-I开关,用于链接到MPP数据库的外部库。

#### --version

打印MPP数据库的版本。

# 示例

要重现当前MPP数据库安装的构建配置,请运行以下命令:

eval ./configure 'pg\_config --configure'

pg\_config --configure的输出包含shell引号,因此带有空格的参数可以正确表示。 因此,需要使用eval才能获得正确的结果。

# psql

MPP数据库的交互式命令行接口

# 概要

psql [option ...] [dbname [username]]

# 描述

psql是MPP数据库的基于终端的前端。它使您能够以交互方式键入查询,将其发布到MPP数据库,并查看查询结果。或者,输入可以来自文件。此外,它提供了许多元命令和各种类似于shell的功能,以帮助编写脚本和自

#### 动化各种任务。

# 选项

#### -a | --echo-all

读取时将所有非空输入行打印到标准输出。 (这不适用于以交互方式读取的行。) 这等效于将变量 *ECHO*设置为all。

### -A | --no-align

切换到不对齐输出模式。 (默认输出模式已对齐。)

#### -c 'command' | --command='command'

指定psql将执行指定的命令字符串,然后退出。 这在shell脚本中很有用。 command必须是服务器可以完全解析的命令字符串,也可以是单个反斜杠命令。 因此,您不能将SQL和psql元命令与此选项混合使用。 为此,您可以将字符串通过管道传递到psql中,如下所示:

echo '\x \\ SELECT \* FROM foo;' | psql

(\是分隔符元命令。)

如果命令字符串包含多个SQL命令,则它们将在单个事务中进行处理,除非字符串中包含明确的BEGIN/COMMIT命令以将其分为多个事务。 这与将相同的字符串提供给psql的标准输入时的行为不同。 此外,仅返回最后一个SQL命令的结果。

#### -d dbname | --dbname=dbname

指定要连接的数据库的名称。这等效于在命令行上将dbname指定为第一个非选项参数。

如果此参数包含=符号或以有效的URI前缀

(postgresql://或postgres://) 开头,则将其视为conninfo字符串。 有关更多信息,请参见PostgreSQL文档中的连接字符串。

### -e | --echo-queries

将所有发送到服务器的SQL命令复制到标准输出。

### -E | --echo-hidden

回显由\d和其他反斜杠命令生成的实际查询。 您可以使用它来研究psql的内部操作。 这等效于将变量 *ECHO\_HIDDEN*设置为on。

### -f filename | --file=filename

使用*filename*作为命令源,而不是交互读取命令。 处理完文件后,psql终止。 在许多方面,这等效于元命令\i。

如果文件名是-(连字符),则将读取标准输入,直到EOF指示或\q元命令为止。但是请注意,在这种情况下不使用Readline(就像已指定-n一样)。

使用此选项与编写psql < filename稍有不同。通常,两者都可以达到您的期望,但是使用-f可以启用一些不错的功能,例如带有行号的错误消息。使用此选项还可能会减少启动开销。另一方面,使用shell的输入重定向的变体在理论上可以保证产生与您手工输入的所有内容完全相同的输出。

### -F separator | --field-separator=separator

使用指定的分隔符作为未对齐输出的字段分隔符。

#### -H | --html

打开HTML表格输出。

### -l | --list

列出所有可用的数据库,然后退出。其他非连接选项将被忽略。

### -L filename | --log-file=filename

除正常输出目标外,还将所有查询输出写入指定的日志文件。

### -n | --no-readline

不要将Readline用于行编辑,也不要使用命令历史记录。 剪切和粘贴时,这对于关闭选项卡扩展很有用。

### -o filename | --output=filename

将所有查询输出放入指定的文件。

## -P assignment | --pset=assignment

允许您在命令行上以\pset样式指定打印选项。 请注意,这里必须用等号(而不是空格)分隔名称和值。 因此,要将输出格式设置为LaTeX,可以编写-P format=latex。

### -q | --quiet

指定psql应该安静地工作。默认情况下,它打印欢迎消息和各种信息输出。如果使用此选项,则不会发生任何情况。 这对于-c选项很有

用。这等效于将变量 QUIET设置为on。

-R separator | --record-separator=separator

使用separator作为未对齐输出的记录分隔符。

#### -s | --single-step

以单步模式运行。 这意味着在将每个命令发送到服务器之前都会提示 用户,并且还可以选择取消执行。 使用它来调试脚本。

### -S | --single-line

在单行模式下运行,其中新行像分号一样终止SQL命令。

#### -t | --tuples-only

关闭列名和结果行计数页脚等的打印。 此命令等效于\pset tuples\_only,并且为方便起见而提供。

-T table\_options | --table-attr=table\_options

允许您指定要放置在HTML表格标记内的选项。 有关详细信息,请参见\pset。

-v assignment | --set=assignment | --variable=assignment

执行变量分配,例如\set meta命令。请注意,必须在命令行上用等号分隔名称和值(如果有)。要取消设置变量,请取消等号。要将变量设置为空值,请使用等号,但不要使用该值。这些分配是在启动的非常早期阶段完成的,因此保留给内部使用的变量可能会在以后被覆盖。

### -V | --version

打印psql版本并退出。

# -x | --expanded

打开扩展表格式模式。

### -X | --no-psqlrc

不要读取启动文件(系统范围内的psqlrc文件或用户的~/.psqlrc文件都不能读取)。

### -z | --field-separator-zero

将未对齐输出的字段分隔符设置为零字节。

#### -0 | --record-separator-zero

将未对齐输出的记录分隔符设置为零字节。 这对于例如与xargs -0配合非常有用。

## -1 | --single-transaction

当psql执行脚本时,添加此选项会将BEGIN/COMMIT包装在脚本周围,以将其作为单个事务执行。 这样可以确保所有命令都成功完成,或者不应用任何更改。

如果脚本本身使用BEGIN,COMMIT或ROLLBACK,则此选项将不会达到预期的效果。 另外,如果脚本包含无法在事务块内执行的任何命令,则指定此选项将导致该命令(并因此导致整个事务)失败。

#### -? | --help

显示有关psql命令行参数的帮助,然后退出。

#### 连接选项

### -h host | --host=host

运行MPP数据库master数据库服务器的计算机的主机名。 如果未指定,则从环境变量PGHOST读取或默认为localhost。

在master主机上启动psql时,如果*host*值以斜杠开头,则它将用作UNIX域套接字的目录。

## -p port | --port=port

MPP master数据库服务器正在侦听连接的TCP端口。 如果未指定,则从环境变量PGPORT读取或默认为5432。

#### -U username | --username=username

要用作连接的数据库角色名称。 如果未指定,则从环境变量PGUSER读取或默认为当前系统角色名称。

### -W | --password

强制输入密码提示。每当服务器请求密码验证时,psql都会自动提示输入密码。但是,当前密码请求检测并不完全可靠,因此此选项会强制提示。如果未发出密码提示,并且服务器要求密码验证,则连接尝试将失败。

#### -w --no-password

不发出密码提示。如果服务器要求密码验证,而其他方式(例如.pgpass文件)无法使用密码,则连接尝试将失败。此选项在没有用户输入密码的批处理作业和脚本中很有用。

注意:此选项在整个会话中保持设置状态,因此会影响元命令\connect的使用以及初始连接尝试。

# 退出状态

如果psql正常完成,它将返回0到shell程序;如果发生自身的致命错误(内存不足,找不到文件),则返回1;如果与服务器的连接断开并且会话不是交互式的,则返回2;如果脚本中发生错误,并设置了变量ON\_ERROR\_STOP,则返回3。

# 用法

连接到数据库

psql是MPP数据库的客户端应用程序。为了连接到数据库,您需要知道目标数据库的名称,MPP主服务器的主机名和端口号以及要用作连接的数据库用户名。可以通过命令行选项(分别为-d,-h,-p和-U)告知psql这些参数。如果找到的参数不属于任何选项,它将被解释为数据库名称(或用户名,如果已经给出数据库名称)。并非所有这些选项都是必需的;有一些可用的默认值。如果省略主机名,则psql将通过UNIX域套接字连接到本地主机上的master服务器,或者通过TCP/IP连接到没有UNIX域套接字的计算机上的localhost。默认的master端口号是5432。如果为master使用其他端口,则必须指定端口。默认数据库用户名是您的操作系统用户名,默认数据库名也是如此。请注意,您不能仅以任何用户名连接到任何数据库。您的数据库管理员应已将您的访问权限告知您。

当默认值不合适时,可以通过将任何或所有环境变量PGAPPNAME, PGDATABASE,PGHOST,PGPORT和PGUSER设置为适当的值来节省键 入时间。

拥有~/.pgpass文件也很方便,可以避免定期输入密码。 该文件应位于您的主目录中,并包含以下格式的行:

hostname:port:database:username:password

.pgpass的权限必须禁止任何其他用户或组的访问(例如:chmod 0600 ~/.pgpass)。 如果权限不严格于此权限,则文件将被忽略。 (但是,当前未在Microsoft Windows客户端上检查文件权限。)

指定连接参数的另一种方法是在conninfo字符串或URI中,而不是使用数据库名称。 这种机制使您可以非常广泛地控制连接。 例如:

\$ psql "service=myservice sslmode=require"

\$ psql postgresql://gpmaster:5433/mydb?sslmode=require

这样,您还可以按照PostgreSQL文档中的LDAP查找连接参数中所述使用LDAP进行连接参数查找。 有关所有可用连接选项的更多信息,请参见PostgreSQL文档中的参数关键字。

如果由于某种原因(权限不足,服务器未运行等)而无法建立连接,则psql将返回错误并终止。

如果标准输入或标准输出中至少有一个是终端,则psql会将客户端编码设置为auto,它将从语言环境设置(Unix系统上的LC\_CTYPE环境变量)中检测适当的客户端编码。如果无法按预期进行,则可以使用环境变量PGCLIENTENCODING覆盖客户端编码。

## 输入SQL命令

在正常操作中,psql会提示您输入当前已连接psql的数据库的名称,后跟字符串=>(对于常规用户)或=#(对于超级用户)。例如:

testdb=> testdb=#

在提示符下,用户可以键入SQL命令。 通常,当到达命令终止分号时,会将输入行发送到服务器。 行尾不会终止命令。 因此,为了清楚起见,命令可以分布在几行上。 如果命令已正确发送和执行,则命令的结果将显示在

### 屏幕上。

如果不受信任的用户可以访问未采用安全模式使用模式的数据库,请通过从 search\_path中删除可公开写入的模式来开始会话。 您可以在连接字符串中添加options=-csearch\_path=或在其他SQL命令之前发出SELECT pg\_catalog.set\_config('search\_path', '', false)。 此注意事项并非特定于psql;它适用于执行任意SQL命令的每个接口。

## 元命令

在psql中输入的任何以不带引号反斜杠开头的内容都是psql元命令,该命令由psql本身处理。 这些命令有助于使psql对管理或脚本化更加有用。 元命令通常被称为斜杠或反斜杠命令。

psql命令的格式为反斜杠,后跟命令动词,然后是参数。参数与命令动词 之间用任意数量的空格字符分隔。

要将空格包含在参数中,可以用单引号将其引起来。要将单引号包含在这样的参数中,请在单引号文本中写两个单引号。此外,单引号中包含的所有内容都会用类似C的方式替换\n(换行),\t(制表符),\b(退格键),\r(回车),\f(换页),\digits(八进制)和\xdigits(十六进制)。在单引号包围的文本中,反斜杠前导任何其他字符,都表示是该单个字符,不管它是什么。

在一个参数中,加上反勾号(`)的文本被看做是一个命令行,传递给shell。 该命令的输出(删除结尾的新行)替换了加上反勾号的文本。

如果一个未加引号的冒号(:)后面跟着一个psql变量名 ,出现在一个参数 中,那么它会被变量的值取代,就像SQL代换中描述的那样。

有些命令以一个SQL标识的名称(比如一个表名)为参数。 这些参数遵循SQL语法关于双引号的规则:不带双引号的标识强制成小写,而双引号(")保护

字母不进行大小写转换,并且允许在标识符中使用空白。 在双引号中,成对的双引号在结果名字中分析成一个双引号。 比如,FOO"BAR"BAZ解析成fooBARbaz;而"A weird" name weird" name.

对参数的分析在行的末尾停止,或者在找到另一个不带引号的反斜杠时停止。一个不带引号的反斜杠会认为是一个新的元命令的开始。 特殊的双反斜杠序列\\标识参数的结尾并将继续分析后面的SQL命令(如果存在的话)。 这样SQL和psql命令可以自由的在一行里面混合。 但是在任何情况下,一条元命令的参数不能延续超过行尾。

## 下面是已定义的元命令:

\a

如果目前的表输出格式是不对齐的,则切换成对齐的。 如果是对齐的,则切换成不对齐。 这条命令是为了向后兼容。 参阅\pset获取一个更通用的解决方法。

## \c | \connect [dbname [username] [host] [port]] | conninfo

与一个PostgreSQL服务器建立一个新的连接。 要使用的连接参数通过位置语法或者使用conninfo连接字符串 (在libpq连接字符串描述)指定。

如果该命令省略了数据库名称,用户,主机或端口,则新连接可以重用先前连接中的值。 默认情况下,除了处理conninfo字符串时,都将重用先前连接的值。 传递-reuse-previous=on或-reuse-

previous=off的第一个参数将覆盖该默认值。 当命令既未指定也不重用特定参数时,将使用libpq缺省值。

将 dbname, username, host或 port中的任何一个指定为-等同于省略该参数。

如果成功制作了新连接,那么关闭以前的连接。如果连接失败,那么 仅当psql处于交互模式时将保留前面的连接。如果运行的是非交互的 脚本,处理会马上停止并返回一个错误。 设置这样的区别一方面是为 用户使用方便考虑,另一方面也为了保证脚本不会碰巧操作了错误的 数据库的安全机制考虑。

## 示例:

- => \c mydb myuser host.dom 6432
- => \c service=foo
- => \c "host=localhost port=5432 dbname=mydb connect timeout=10 sslmode=dis
- => \c postgresql://tom@localhost/mydb?application\_name=myapp

## \C [title]

把正在打印的表的标题设置为一个查询的结果或者取消这样的设置。 这条命令等效于\pset title。

## \cd [directory]

把当前工作目录改变到directory。 没有参数则是改变到当前用户的家目录。 使用\!pwd打印出当前工作目录。

#### \conninfo

显示有关当前连接的信息,包括数据库名称,用户名,连接类型 (UNIX域套接字,TCP/IP等),主机和端口。

'copy {table [(column\_list)] | (query)} {from | to} {'filename' | program
'command' | stdin | stdout | pstdin | pstdout} [with] (option [, ...]) ]

执行前端(客户端)拷贝。这是一个运行SQL COPY命令的操作,不同的是服务器在读写指明的文件,而psql读写文件并作为本地的文件系统和服务器之间的跳板取出或写入数据。这意味着文件访问性和权限都是本地用户的,而不是服务器的,因此不需要SQL超级用户权限。当指定了program时,command是通过psql执行的,并且来自或到达command的数据在服务器和客户端之间传送。再次,文件访问性和权限都是本地用户的,而不是服务器的,因此不需要SQL超级用户权限。

\copy ... from stdin | to stdout分别基于命令输入和输出进行读/写。从发出命令的同一源读取所有行,直到\.为止或流到达EOF。输出被发送到与命令输出相同的位置。要从psql的标准输入或输出读取/写入,请使用pstdin或pstdout。此选项对于在SQL脚本文件中内联填充表很有用。

该命令的语法与SQL COPY命令的语法相似,并且 option必须指示 SQL COPY命令的选项之一。请注意,因此,特殊的解析规则适用 于\copy命令。特别是,变量替换规则和反斜杠转义不适用。 此操作不如 SQL COPY命令有效,因为所有数据都必须通过客户端/服

务器连接传递。

## \copyright

显示MPP数据库所基于的PostgreSQL的版权和发行条款。

## \d [relation\_pattern] | \d+ [relation\_pattern] | \dS [relation\_pattern]

对于每个匹配pattern的关系 (表、视图、索引、序列或外部表)或复合类型,显示所有列、它们的类型、表空间(如果不是缺省的)和任何特殊属性(诸如NOT NULL或缺省等)。 相关的索引、约束、规则、触发器也同样显示出来。 对于外部表,也显示相关的外部服务器。

- 对于一些关系类型, \d为每个字段显示了额外的信息: 序列的字段值,索引的索引表达式和外部表的外部数据封装器选项。
- 命令形式\d+相同,除了显示更多信息:显示与表的列关联的所有注释,以及表中是否存在OID,如果关系是视图,则为视图定义。对于分区表,用根分区表或子分区表指定的命令\d或\d+显示有关该表的信息,包括分区表当前级别上的分区键。命令\d+还显示表的直接子分区以及该子分区是外部表还是常规表。对于追加优化的表和面向列的表,\d+显示表的存储选项。对于追加优化的表,将显示该表的选项。对于面向列的表,将为每列显示存储选项。
- 缺省的,只显示用户创建的对象;应用一个模式或S修饰符包含系 统对象。

Note: 如果不带任何pattern参数调用\d,那么等效于\dtvsE,将显示一个所有可见表,视图,序列和外部表的列表。

## \da[S] [aggregate\_pattern]

列出聚合函数,以及它们的返回类型和操作的数据类型。 如果声明了 pattern,那么只显示匹配模式的聚合函数。 缺省的,只显示用户创建的对象;应用一个模式或S修饰符包含系统对象。

## \db[+] [tablespace\_pattern]

列出所有可用的表空间及其对应的路径。 如果指定了模式,则仅显示名称与模式匹配的表空间。 如果在命令名称后附加+,则会列出每个对象及其关联的权限。

## \dc[S+] [conversion\_pattern]

列出字符集编码之间的转换。 如果指定了模式,则仅列出名称与该模

式匹配的转换。默认情况下,仅显示用户创建的对象。 提供一个模式或S修饰符以包含系统对象。 如果在命令名称后附加+,则会列出每个对象及其相关描述。

## \dC[+] [pattern]

列出类型转换。 如果指定了模式,则仅列出其源或目标类型与该模式 匹配的转换。 如果在命令名称后附加+,则会列出每个对象及其相关 描述。

## \dd[S] [pattern]

显示类型constraint, operator class, operator

family,rule和trigger的对象的描述。 所有其他注释都可以通过相应的 反斜杠命令查看那些对象类型。

\dd显示与模式匹配的对象的描述,或者如果没有给出参数,则显示适当类型的可见对象的描述。 但无论哪种情况,仅列出具有描述的对象。 默认情况下,仅显示用户创建的对象。 提供一个模式或S修饰符以包含系统对象。

可以使用COMMENT SQL命令创建对象的描述。

## \ddp [pattern]

列出默认的访问权限设置。 将为每个角色 (和模式,如果适用) 显示一个条目,其默认权限设置已从内置默认值更改。 如果指定

了 pattern,则仅列出角色名称或模式名称与模式匹配的条目。

ALTER DEFAULT PRIVILEGES命令用于设置默认访问权限。 特权显示的含义在GRANT下进行了说明。

## \dD[S+] [domain\_pattern]

列出域。如果指定了模式,则仅显示名称与该模式匹配的域。 默认情况下,仅显示用户创建的对象。 提供一个模式或S修饰符以包含系统对象。 如果在命令名称后附加+,则会列出每个对象及其关联的权限和描述。

\dEistPv[S+] [external\_table | index | sequence | table | parent table | view] 这不是实际的命令名称:字母E,i,s,t,P和v分别代表外部表,索引,序列,表,父表和视图。 您可以按任何顺序指定任何或所有这些字母,以获得这些类型的对象的列表。 例如,\dit列出索引和表。 如果在命令名称后附加+,则会列出每个对象及其在磁盘上的物理大小及其相关的描述(如果有)。 如果指定了模式,则仅列出名称与该模式

匹配的对象。 默认情况下,仅显示用户创建的对象。 提供一个模式或 S修饰符以包含系统对象。

## \des[+] [foreign\_server\_pattern]

列出外部服务器。如果指定了模式,则仅列出名称与该模式匹配的服务器。如果使用\des+形式,则会显示每个服务器的完整描述,包括服务器的ACL,类型,版本,选项和描述。

## \det[+] [foreign\_table\_pattern]

列出所有外部表。 如果指定了模式,则仅列出其表名或模式名称与该模式匹配的条目。 如果使用\det+形式,则还将显示通用选项和外部表描述。

## \deu[+] [user\_mapping\_pattern]

列出用户映射。 如果指定了模式,则仅列出其用户名与模式匹配的那些映射。 如果使用\deu+形式,则会显示有关每个映射的其他信息。 Warning: \deu+可能也显示远程用户的用户名和密码,所以要小心不要透漏它们。

## \dew[+] [foreign\_data\_wrapper\_pattern]

列出外部数据包装器。如果指定了模式,则仅列出名称与该模式匹配的数据包装器。如果使用\dew+形式,则还将显示ACL,选项和外部数据包装程序的描述。

## \df[antwS+] [function\_pattern]

列出函数以及它们的参数,返回类型和函数类型,这些函数分为"agg"(聚合),"normal","trigger"或"window"。要仅显示特定类型的函数,请在命令中添加相应的字母a,n,t或w。如果指定了模式,则仅显示名称与该模式匹配的函数。如果使用\df+形式,则会显示有关每个函数的其他信息,包括安全性,易变性,语言,源代码和描述。默认情况下,仅显示用户创建的对象。提供一个模式或S修饰符以包含系统对象。

## \dF[+] [pattern]

列出文本搜索配置。 如果指定了模式,则仅显示名称与该模式匹配的配置。 如果使用\dF+形式,则会显示每个配置的完整描述,包括基础文本搜索解析器和每种解析器令牌类型的字典列表。

## \dFd[+] [pattern]

列出文本搜索词典。 如果指定了模式,则仅显示名称与该模式匹配的

词典。 如果使用\dFd+形式,则会显示有关每个所选词典的其他信息,包括基础的文本搜索模板和选项值。

## \dFp[+] [pattern]

列出文本搜索解析器。 如果指定了模式,则仅显示名称与模式匹配的解析器。 如果使用\dFp+形式,则会显示每个解析器的完整说明,包括基础函数和已识别令牌类型的列表。

## \dFt[+] [pattern]

列出文本搜索模板。 如果指定了模式,则仅显示名称与该模式匹配的模板。 如果使用\dFt+形式,则会显示有关每个模板的其他信息,包括基础函数名称。

## \dg[+] [role\_pattern]

列出数据库角色。 (由于将"用户"和"组"的概念统一为"角色",因此该命令现在等效于\du。) 如果指定了模式,则仅列出名称与模式匹配的角色。 如果使用\dg+形式,则将显示有关每个角色的其他信息;否则,将显示其他信息。 目前,这会为每个角色添加注释。

\dl

这是\lo\_list的别名,其中显示了大对象列表。

Note: MPP数据库不支持PostgreSQL 大对象工具来流存储在大对象结构中的用户数据。

## \dL[S+] [pattern]

列出过程语言。如果指定了模式,则仅列出名称与该模式匹配的语言。默认情况下,仅显示用户创建的语言。 提供S修饰符以包括系统对象。 如果在命令名称后附加+,则将列出每种语言及其调用处理程序,验证程序,访问特权以及是否为系统对象。

## \dn[S+] [schema\_pattern]

列出所有可用的模式(名称空间)。如果指定了模式,则仅列出名称与模式匹配的模式。默认情况下,仅显示用户创建的对象。提供一个模式或S修饰符以包含系统对象。如果在命令名后附加+,则列出每个对象及其相关的权限和描述(如果有)。

## \do[S] [operator\_pattern]

列出可用的运算符及其操作数和返回类型。 如果指定了模式,则仅列出名称与模式匹配的运算符。 默认情况下,仅显示用户创建的对象。 提供一个模式或S修饰符以包含系统对象。

## \dO[S+] [pattern]

列出排序规则。如果指定了模式,则仅列出名称与模式匹配的排序规则。默认情况下,仅显示用户创建的对象。提供一个模式或S修饰符以包含系统对象。如果在命令名称后附加+,则将列出每个排序规则及其关联的描述(如果有)。请注意,仅显示可与当前数据库的编码一起使用的排序规则,因此在同一安装的不同数据库中,结果可能会有所不同。

## \dp [relation\_pattern\_to\_show\_privileges]

列出具有相关访问权限的表,视图和序列。 如果指定了模式,则仅列出名称与该模式匹配的表,视图和序列。 GRANT和REVOKE命令用于设置访问权限。 特权显示的含义在GRANT下进行了说明。

## \drds [role-pattern [database-pattern]]

列出定义的配置设置。这些设置可以是角色特定的,数据库特定的或两者。 *role-pattern*和 *database-pattern*分别用于选择特定角色和要列出的数据库。 如果省略,或者指定\*,则列出所有设置,包括分别不是角色特定的或数据库特定的设置。

ALTER ROLE和ALTER DATABASE命令用于定义每个角色和每个数据库的角色配置设置。

## \dT[S+] [datatype\_pattern]

列出数据类型。如果指定了模式,则仅列出名称与模式匹配的类型。如果在命令名称后附加+,则将列出每种类型及其内部名称和大小,如果是枚举类型则允许的值及其关联的权限。默认情况下,仅显示用户创建的对象。提供一个模式或S修饰符以包含系统对象。

## \du[+] [role\_pattern]

列出数据库角色。 (由于将"用户"和"组"的概念统一为"角色",因此此命令现在等效于\dg。) 如果指定了模式,则仅列出名称与模式匹配的角色。 如果使用\du+形式,则会显示有关每个角色的其他信息;目前,这会为每个角色添加注释。

## \dx[+] [extension\_pattern]

列出已安装的扩展。 如果指定了模式,则仅列出名称与该模式匹配的 扩展名。 如果使用\dx+形式,则列出属于每个匹配扩展的所有对象。

## \dy[+] [pattern]

列出事件触发器。 如果指定了模式,则仅列出其名称与该模式匹配的

触发器。如果在命令名称后附加+,则会列出每个对象及其相关描述。

Note: MPP数据库不支持用户定义的触发器。

## \e | \edit [filename] [line\_number]

如果指定filename,则文件被编辑;编辑器退出后,其内容将复制回查询缓冲区。如果未提供filename,则将当前查询缓冲区复制到一个临时文件,然后以相同的方式对其进行编辑。

然后根据psql的常规规则重新解析新的查询缓冲区,其中整个缓冲区被视为一行。(因此,您不能以这种方式制作脚本。为此请使用\i。

) 这也意味着,如果查询以分号结尾(或包含分号),则会立即执行 该查询。 在其他情况下,它将仅在查询缓冲区中等待;输入分号 或\g发送,或\r取消。

如果指定了行号,则psql将光标定位在文件或查询缓冲区的指定行上。 请注意,如果给出单个全数字参数,则psql假定它是行号,而不是文件名。

有关配置和自定义编辑器的信息,请参见环境变量。

### \echo text[...]

将参数输出到标准输出,中间用一个空格分隔,后跟换行符。 这对于在脚本输出中散布信息很有用。 如果第一个参数是未加引号的-n,则不写结尾的换行符。

Note: 如果使用\o命令重定向查询输出,则可能希望使用\qecho代替此命令。

## \ef [function\_description [line\_number]]

该命令以CREATE OR REPLACE FUNCTION命令的形式获取并编辑命名函数的定义。编辑的方式与\edit相同。编辑器退出后,更新的命令在查询缓冲区中等待;输入分号或\g发送,或\r取消。

目标函数可以仅通过名称指定,也可以通过名称和参数指定,例如foo(integer, text)。 如果有多个具有相同名称的函数,则必须给出参数类型。

如果未指定任何函数,则会显示空白的CREATE FUNCTION模板以供编辑。

如果指定了行号,则psql将光标定位在函数主体的指定行上。 (请注意,函数主体通常不在文件的第一行开始。)

有关配置和自定义编辑器的信息,请参见环境变量。

## \encoding [encoding]

设置客户端字符集编码。如果不带参数,此命令将显示当前编码。

## \f [field\_separator\_string]

为未对齐的查询输出设置字段分隔符。 默认值为竖线 (|) 。 另请参阅\pset以获取设置输出选项的通用方法。

## \g [filename]

## \g [ | command ]

将当前查询输入缓冲区发送到服务器,并有选择地将查询的输出存储到 filename中,或将输出通过管道传递给 shell命令 command。 仅当查询成功返回零个或多个元组时,才写入文件或命令,如果查询失败或是不返回数据的 SQL命令,则不写入文件或命令。

裸\g本质上等效于分号。 带参数的\g是\o命令的一次性替代方案。

## \gset [prefix]

将当前查询输入缓冲区发送到服务器,并将查询的输出存储到psql变量中。要执行的查询必须恰好返回一行。该行的每一列都存储在一个单独的变量中,该变量与该列相同。例如:

=> SELECT 'hello' AS var1, 10 AS var2;

-> \gset

=> \echo :var1 :var2

hello 10

如果指定*prefix*,则在查询的列名称之前添加该字符串以创建要使用的变量名称:

=> SELECT 'hello' AS var1, 10 AS var2;

-> \gset result\_

=> \echo :result\_var1 :result\_var2

hello 10

如果列结果为NULL,那么对应的变量是未设置的。 如果查询失败或不返回一行,则不会更改任何变量。

## \h | \help [sql\_command]

提供有关指定SQL命令的语法帮助。如果未指定命令,则psql将列出所有可使用语法帮助的命令。如果 command 是星号(\*),则会显示所有SQL命令的语法帮助。为了简化键入,不必引用由几个单词组成的命令。

## \H | \html

打开HTML查询输出格式。如果HTML格式已打开,则将其切换回默认的对齐文本格式。此命令是为了兼容性和方便起见,但有关设置其他输出选项的信息,请参见\pset。

## \i | \include filename

从文件filename中读取输入,并像在键盘上键入文件一样执行它。如果filename是-(连字符),则将读取标准输入,直到EOF指示或\q元命令为止。这可用于将交互式输入与文件输入进行穿插。请注意,只有在最外层处于活动状态时,才会使用Readline行为。如果要在阅读时看到屏幕上的行,必须将变量ECHO设置为all。

## \ir | \include\_relative filename

\ir命令类似于\i,但是以不同的方式解析相对文件名。 在交互模式下执行时,两个命令的行为相同。 但是,从脚本调用时,\ir会相对于脚本所在目录而不是当前工作目录来解释文件名。

## \[+] | \list[+] [pattern]

列出服务器中的数据库,并显示其名称,所有者,字符集编码和访问特权。 如果指定了模式,则仅列出名称与该模式匹配的数据库。 如果在命令名称后附加+,则还会显示数据库大小,默认表空间和描述。 (大小信息仅适用于当前用户可以连接到的数据库。)

## \lo export loid filename

从数据库读取OID *loid*的大对象,并将其写入*filename*。 请注意,这与服务器函数lo\_export稍有不同,后者以数据库服务器运行的用户权限和服务器文件系统上的用户权限运行。 使用\lo\_list找出大对象的OID。

Note: MPP数据库不支持PostgreSQL大对象工具来流存储在大对象结构中的用户数据。

## \lo\_import large\_object\_filename [comment]

将文件存储到大对象中。 (可选) 它将给定的注释与对象相关联。 例: mydb=> \lo\_import '/home/gpadmin/pictures/photo.xcf' 'a picture of me' lo import 152801

该响应表明大对象接收到对象ID 152801,如果要再次访问该对象,应该记住该对象。 因此,建议始终将可读的注释与每个对象相关联。 然后可以使用\lo\_list命令查看这些内容。 请注意,此命令与服务器端lo\_import略有不同,因为它充当本地文件系统上的本地用户,而不是服务器的用户和文件系统。

Note: MPP数据库不支持PostgreSQL大对象工具来流存储在大对象结构中的用户数据。

## \lo\_list

显示当前存储在数据库中的所有大型对象的列表,以及为其提供的任何注释。

Note: MPP数据库不支持PostgreSQL大对象工具来流存储在大对象结构中的用户数据。

## \lo\_unlink largeobject\_oid

从数据库中删除指定OID的大对象。使用\lo\_list找出大对象的OID。

Note: MPP数据库不支持PostgreSQL大对象工具来流存储在大对象结构中的用户数据。

## \o | \out [ filename ]

## \o | \out [ | command ]

将将来的查询结果保存到filename,或将将来的结果通过管道传递给shell命令command。如果未指定任何参数,则查询输出将重置为标准输出。查询结果包括从数据库服务器获得的所有表,命令响应和通知,以及查询数据库的各种反斜杠命令的输出(例如\d),但不包括错误消息。要在查询结果之间插入文本输出,请使用\qecho。

/p

将当前查询缓冲区打印到标准输出。

## \password [username]

更改指定用户(默认情况下为当前用户)的密码。该命令提示输入新密码,对其进行加密,然后将其作为ALTER ROLE命令发送到服务器。这样可以确保新密码不会在命令历史记录,服务器日志或其他地

方以明文形式出现。

## \prompt [ text ] name

提示用户提供文本,该文本已分配给变量*name*。 可以指定一个可选的提示字符串,*text*。 (对于多字提示,请用单引号将文本引起来。)

默认情况下,\prompt使用终端进行输入和输出。 但是,如果使用-f命令行开关,\prompt将使用标准输入和标准输出。

## \pset [print\_option [value]]

此命令设置影响查询结果表输出的选项。 *print\_option*描述了要设置的选项。 *value*的语义取决于所选选项。 对于某些选项,省略 *value*会导致该选项被切换或取消设置,如特定选项所述。 如果未提及此类行为,则省略 *value*只会导致显示当前设置。

没有任何参数的\pset将显示所有打印选项的当前状态。

## 可调打印选项包括:

- border *value*必须是一个数字。 通常,数字越大,表就有越宽的 边界和越多的线,但是这个参数取决于实际的格式。 在HTML格式中,这个参数会直接翻译成border=...属性, 在其它的格式中,只有值0(无边界)、1(内部分隔线)、2(表框架)有意义。 latex和latex-longtable也支持border值为3,在每行之间添加一个分隔线。
- columns 设置wrapped格式的目标宽度,以及用于确定输出是否足够宽以要求分页或在扩展自动模式下切换到垂直显示的宽度限制。默认值为零。零导致目标宽度由环境变量COLUMNS控制,如果未设置COLUMNS,则检测到的屏幕宽度。另外,如果columns为零,则wrapped格式仅影响屏幕输出。如果columns不为零,则文件和管道输出也将被包装为该宽度。设置目标宽度后,请使用\pset format wrapped命令启用包装格式。
- expanded | x 如果指定了 value,则必须是on或off,这将启用或禁用扩展模式或auto。如果省略 value,该命令将在on和off设置之间切换。启用扩展模式后,查询结果将显示在两列中,列名在左侧,数据在右侧。如果在正常的"水平"模式下数据无法显示在屏幕上,则此模式很有用。在auto设置中,只要查询输出比屏幕宽,就使用扩展模式,否则使用常规模式。 auto设置仅在对齐和包装的格式中

有效。 在其他格式下,它始终表现为关闭扩展模式。

- fieldsep 指定在未对齐输出模式下使用的字段分隔符。这样,就可以创建制表符或逗号分隔的输出,而其他程序可能更喜欢这种输出。要将选项卡设置为字段分隔符,请键入\pset fieldsep '\t'。默认字段分隔符为'|'(竖线)。
- fieldsep\_zero 设置非对齐输出格式中使用的域分隔符为0字节。
- footer 如果指定了 *value*,则必须将其打开或关闭,这将启用或禁用表脚的显示((*n*行) 计数)。如果省略 *value*,则该命令将打开或关闭页脚显示。
- format 将输出格式设置为unaligned, aligned, html, latex (使用tabular), latex-longtable, troff-ms或wrapped之一。允许使用唯一的缩写,包括一个字母。

unaligned格式将一行的所有列写在一行上,由当前活动的字段分隔符分隔。这对于创建可能打算由其他程序读取的输出(例如,制表符分隔或逗号分隔的格式)很有用。

aligned格式是标准的,人类可读的,格式良好的文本输出;这是默 认值。

html,latex,latex-longtable和troff-ms模式把表输出为可用于文档里的对应标记语言。它们还不是完整的文档!(可能对于HTML变化还不是太大,但是在LaTeX中,您必须具有完整的文档包装器。latex-longtable也需要LaTeX longtable和booktabs软件包。)wrapped格式类似于aligned,但是在行之间包装宽数据值以使输出适合目标列宽。目标宽度是根据columns选项下所述确定的。请注意,psql不会尝试包装列标题。如果列标题所需的总宽度超过目标,则wrapped格式与aligned的行为相同。

linestyle [unicode | ascii | old-ascii] – 将边界线绘制样式设置为unicode, ascii或old-ascii中的一种。 三种样式允许使用唯一的缩写,包括一个字母。默认设置为ascii。 此选项仅影响aligned和wrapped的输出格式。

ascii – 使用纯ASCII字符。数据中的换行符在右边距中使用+符号显示。当换行格式将数据从一行换行到下一行而没有换行符时,在第

一行的右边距中显示一个点(.),在下一行的左边距中再次显示一个点(.)。

old-ascii – 样式使用纯ASCII字符,并使用PostgreSQL 8.4及更早版本中使用的格式样式。数据中的换行符用:符号代替左侧的列分隔符。当数据从一行到另一行的换行而没有换行符时,;符号用于代替左侧的列分隔符。

unicode – 样式使用Unicode框画字符。数据中的换行符在右边缘使用回车符显示。当数据从一行换行到下一行而没有换行符时,在第一行的右边距中显示省略号,并在下一行的左边距中显示省略号。

当border设置大于零时,此选项还确定绘制边框线所使用的字符。 纯ASCII字符在任何地方都可以使用,但是Unicode字符在可以识别 它们的显示器上看起来更好。

- null 'string' 第二个参数是一列为空时要打印的字符串。 默认设置为不打印任何内容,这很容易被误认为是空字符串。 例如,可能更喜欢\pset null '(null)'。
- numericlocale 如果指定了 value,则必须将其打开或关闭,这将 启用或禁用显示特定于语言的字符以将小数点标记左侧的数字组分 开。如果省略 value,则该命令在常规和特定于区域的数字输出之 间切换。
- pager 控制用于查询和psql帮助输出的分页器程序的使用。如果设置了环境变量PAGER,则将输出通过管道传递到指定的程序。否则,将使用依赖于平台的默认值(例如more)。禁用时,不使用分页器程序。启用时,仅在适当的时候使用分页器,即当输出到终端且不匹配屏幕时。也可以将分页器设置为always,这将使该分页器用于所有终端输出,而不管其是否适合屏幕显示。\pset pager (不带 value) 可打开和关闭分页器的使用。
- recordsep 声明在非对齐模式时的记录分隔符。缺省是换行符。
- recordsep\_zero 设置在非对齐输出格式中使用的记录分隔符为0字 节。
- tableattr | T [text] 在HTML格式下,这指定了要放置在table标签中

的属性。例如cellpadding或bgcolor。请注意,你可能不需要在这里声明border,因为已经在\pset border里用过了。如果没有给出 *value*,那么表的属性是未设置的。

在latex-longtable格式中,这控制每个包含左对齐数据类型的字段的比例宽度。 它声明为空白分隔的值列表,比如'0.2 0.2 0.6'。 未指定的输出列使用最后指定的值。

- title [text] 为任何随后打印的表设置标题。 这个参数可以用于给你的输出一个描述性标记。 如果没有给出 value,则标题未设置。
- tuples\_only | t [novalue | on | off] 如果指定了 value,那么必须是on或off,这将启用或禁用仅元组模式。如果省略了 value,那么该命令在普通和仅元组输出间切换。普通输出包括额外的信息,比如列头、标题、各种脚注等。在仅元组模式下,只显示实际的表数据。\t命令等效于\psettuples\_only,并且为方便起见而提供。

## Tip:

有很多用于\pset的快速命令。 参阅\a, \C, \f, \H, \t, \T和\x。

## $\q \mid \q$

退出psql程序。在脚本文件中,仅终止该脚本的执行。

## \qecho text[...]

此命令与\echo相同,除了输出将被写入\o设置的查询输出通道外。

## \r | \reset

重置 (清除) 查询缓冲区。

## \s [filename]

将psql的命令行历史记录打印到filename。如果省略filename,则将历史记录写入标准输出(如果合适,使用分页器)。如果psql是在没有Readline支持的情况下构建的,则此命令不可用。

## \set [name [value [ ... ]]]

将psql变量 *name*设置为 *value*,或者如果给定多个值,则将其全部串联。如果仅给出一个参数,则该变量将设置为空值。 要取消设置变量,请使用\unset命令。

没有任何参数的\set将显示所有当前设置的psql变量的名称和值。 有效的变量名称可以包含字符,数字和下划线。 请参阅高级特性中的"变量"。 变量名称区分大小写。 尽管你可以设置任何变量为任意值,psql对一些变量特殊对待。它们在关于变量的小节里面有文档。

该命令与SQL命令SET无关。

## \setenv name [ value ]

将环境变量 *name*设置为 *value*,或者如果未提供该 *value*,则取消设置环境变量。例:

testdb=> \setenv PAGER less testdb=> \setenv LESS -imx4F

## \sf[+] function\_description

该命令以CREATE OR REPLACE FUNCTION命令的形式获取并显示命名函数的定义。 定义将打印到\o设置的当前查询输出通道。

目标函数可以仅通过名称指定,也可以通过名称和参数指定,例如foo(integer, text)。 如果有多个相同名称的函数,则必须提供参数类型。

如果在命令名称后附加+,则对输出行进行编号,函数主体的第一行为 行1。

## \t [novalue | on | off]

\t命令本身可切换输出列名称标题和行计数页脚的显示。 无论当前设置如何,打开和关闭的值都会设置元组显示。 此命令等效于\pset tuples\_only,并且为方便起见而提供。

## \T table options

声明HTML输出格式中放在table标记里的属性。 此命令等效于\pset tableattr *table options* 

## \timing [novalue | on | off]

如果没有参数,则以毫秒为单位切换显示每个SQL语句花费的时间。 无论当前设置如何,打开和关闭的值都会设置时间显示。

#### \unset name

取消设置 (删除) psql变量 name。

#### \w | \write filename

## \w | \write | command

将当前查询缓冲区输出到filename或将其通过管道传递给shell命

#### **♦ command**。

## \watch [seconds]

重复执行当前查询缓冲区(如\g),直到被中断或查询失败。在两次执行之间等待指定的秒数(默认为2)。

## \x [ on | off | auto ]

设置或切换扩展表格式设置模式。 因此,它等效于\pset expanded。

## \z [pattern]

列出具有相关访问权限的表,视图和序列。 如果指定了模式,则仅列出名称与该模式匹配的表,视图和序列。 这是\dp的别名。

## \! [command]

转义到单独的shell或执行shell命令 command。参数不会被进一步解释,shell 将看到全部参数。特别是,变量替换规则和反斜杠转义不适用。

\?

显示有关psql反斜杠命令的帮助信息。

## 模式

各种\d命令都接受一个pattern参数,声明要显示的对象名字。 最简单的情况下pattern正好等于对象的名字。 pattern中的字符通常会被自动转换成小写,就像SQL名字一样。 例如\dt FOO将显示名为foo的表。 与在SQL名字中相同的是双引号界定的pattern将保持原样(不做大小写转换)。 如果需要在双引号界定的pattern中使用双引号字符,你可以写两个并列的双引号,这与SQL的引号规则相同。 例如,\dt "FOO""BAR"将会显示名为FOO"BAR的表,但是不会显示foo"bar。 与一般的SQL名字规则不同的是,你可以仅用双引号界定名字的一部分,例如\dt FOO"FOO"BAR将显示名为fooFOObar的表。

在模式中,\*匹配任何字符序列(包括无字符)和?匹配任何单个字符。 (此表示法可与UNIX Shell文件名模式相提并论。) 例如,\dt int\*显示名称 以int开头的所有表。 但是在双引号内,\*和?失去了这些特殊含义,只是字 面上的匹配。 包含点(.)的模式被解释为模式名称模式,后跟对象名称模式。例如,\dt foo\*.bar\*将显示其表名以bar开头的所有表,这些表的模式名以foo开头。如果没有点出现,则该模式仅匹配在当前模式搜索路径中可见的对象。同样,双引号中的点失去其特殊含义,并且在字面上匹配。

高级用户可以使用正则表达式。除了.如上所述被当做分隔符,\*将被理解成.\*,?将被理解成.,所有正则表达式特殊字符均按PostgreSQL正则表达式文档的规定工作。这样一来,你就可以用?代替.、用(R+|)代替R\*、用(R|)代替R?。请记住,模式必须匹配全名,这与对正则表达式的通常解释不同;如果您不希望锚定模式,请在开头和/或结尾处输入\*。请注意,在双引号中,所有正则表达式特殊字符都会失去其特殊含义,并且会在字面上进行匹配。同样,正则表达式特殊字符在运算符名称模式(例如\do的参数)中按字面值进行匹配。

每当完全省略pattern参数时,\d命令都会显示在当前模式搜索路径中可见的所有对象 - 这等效于使用模式\*。要查看数据库中的所有对象,请使用模式\*.\*。

## 高级特性

## 变量

psq提供类似通常Unix命令shell那样的变量替换特性。 变量只是简单的名称/值对,这里的值可以是任何长度的任何值。 名字必须由字母(包括非拉丁字母)、数字和下划线组成。

要设置一个变量,使用psql元命令\set。例如:

testdb=> \set foo bar

把变量foo的值设置为bar。要检索变量的内容,在变量名前面放上冒号,例如:

testdb=> \echo :foo bar

这在SQL命令和元命令中都能运行;更详细信息在下面的SQL代换中给出。

如果你不带第二个参数调用\set,那么设置这个变量,带有一个空字符串作为 value。 要重置(也就是删除)一个变量,使用\unset命令。 要显示所有变量的值,不带有任何参数调用\set。

Note: \set的参数服从和其它命令一样的替换规则。 因此你可以构造有趣的引用,像\set:foo 'something'这样,获得分别像Perl或PHP那样有名的''软连接''或''变量变量''。 不幸的是,用这些构造不能做任何有用的事情。 另一方面,\set bar:foo是一个非常有效的拷贝变量的方法。

有一些常用变量被psql特殊相待。它们代表特定的选项设置,这些选项在运行时可以通过改变变量的值而改变,或者在某些情况下代表psql的可变状态。尽管你可以把这些变量用于其它用途,但是不鼓励这么做,因为程序的行为可能会变得非常奇怪。通常,所有特殊对待的变量名都是由大写ASCII字母组成(可能还有数字和下划线)。为了保证和未来的最大限度的兼容性,请避免使用这样的变量。下面是一个所有特殊对待的变量列表。

#### **AUTOCOMMIT**

启用时(默认),成功完成后将自动提交每个SQL命令。要以这种方式推迟提交,必须输入BEGIN或START TRANSACTION SQL命令。禁用或未设置时,除非明确发出COMMIT或END,否则不会提交SQL命令。自动提交模式的工作方式是:在任何尚未在事务块中且本身不是BEGIN或其他事务控制命令的命令或无法在事务块内部执行的命令之前(例如VACUUM),为您发出隐式BEGIN。

在自动提交模式下,必须通过输入ABORT或ROLLBACK明确放弃任何失败的事务。 另外请记住,如果您退出会话而不提交,则您的工作将会丢失。

自动提交模式是PostgreSQL的传统行为,但是自动提交模式更接近SQL规范。如果您喜欢关闭自动提交,在~/.psqlrc文件中进行设置。

## **COMP KEYWORD CASE**

在完成SQL关键字时决定哪个字母使用大小写。 如果设置为lower或upper,则完成的单词将分别为小写或大写。 如果设置为preserve-lower或preserve-upper (缺省) ,那么完成的单词将是输入时的情况, 但是单词在没有任何输入的情况下完成,将分别是小写或大写的情况。

## **DBNAME**

正在连接着的数据库名称。每次与一个数据库联结都会设置这个值(包括程序启动),但是可以删除。

#### **ECHO**

如果设为all,那么所有非空输入行在读取时都回显到标准输出。(这不适用于交互式读取的行。)使用-a选项声明在程序启动时就默认这样做。如果设置为queries,那么psql只是在查询发送给服务器时打印到标准输出。这个开关是-e。

## **ECHO\_HIDDEN**

当此变量设置为on并且反斜杠命令查询数据库时,将首先显示该查询。此特性可帮助您研究MPP数据库的内部结构,并在您自己的程序中提供类似的功能。(要在程序启动时选择此行为,请使用开关-E。)加里格亦量设置为值noevec、则仅显示查询、但实际上并未将其发

) 如果将变量设置为值noexec,则仅显示查询,但实际上并未将其发送到服务器并执行查询。

#### **ENCODING**

当前的客户端字符集编码。

## **FETCH COUNT**

如果将此变量设置为大于0的整数值,则将提取SELECT查询的结果并将其显示在这么多行的组中,而不是在显示之前收集整个结果集的默认行为。因此,无论结果集的大小如何,仅使用有限的内存量。 启用此功能时,通常使用100到1000的设置。 请记住,使用此功能时,在显示了一些行后查询可能会失败。

尽管可以使用此特性使用任何输出格式,但是默认的对齐格式看起来很糟糕,因为每组FETCH\_COUNT行将分别设置格式,从而导致各行组的列宽变化。不过对于其他格式这个特性工作的很好。

#### HISTCONTROL

如果将这个变量设为ignorespace,那么以空格开始的行将不会进入历

史列表。如果设置为ignoredups,那么与以前历史记录里匹配的行也不会进入历史记录。值ignoreboth是上面两个的结合。如果删除此变量或者其值为任何与上面的值不同的东西,所有交互模式读入的行都被保存入历史列表。

#### HISTFILE

此文件将用于存储历史列表。 默认值是~/.psql\_history。 例如,在~/.psqlrc里使用:

\set HISTFILE ~/.psql history-:DBNAME

将使得psql为每个数据库维护一个独立的历史。

#### **HISTSIZE**

保存在命令历史里的命令的个数。缺省值是500。

#### **HOST**

当前你正连接的数据库服务器主机。 这是在每次你与数据库连接时(包括程序启动)设置的,但是可以删除。

#### **IGNOREEOF**

如果删除此变量,向一个交互的psql会话发送一个EOF(通常是CTRL+D)将终止应用。 如果设置为一个数字值,那么在应用终止前该数值的EOF字符将被忽略。 如果设置了此变量但是没有数字值,缺省是10。

#### **LASTOID**

最后影响的OID值,即为从一条INSERT或lo\_import命令返回的值。 此变量只保证在下一条SQL命令的结果显示之前有效。

## ON\_ERROR\_ROLLBACK

当设置为on时,如果一个事务块里的语句产生错误,这个错误将被忽略而事务将继续。 当设置为interactive时,这样的错误只是在交互的会话里忽略,而不是在从读取脚本文件的时候。 如果未设置或者设置为off,事务块里一个语句生成的错误将会中止整个事务。 错误回滚的模式是通过在一个事务块的每个命令前为你隐含地发出一

个SAVEPOINT的方式工作的,在命令错误的时候回滚到该保存点。

## ON\_ERROR\_STOP

缺省时,遇到错误后命令处理继续进行。当这个变量设置为on,处理

会立即停止。在交互模式下,psql将返回到命令提示符;否则,psql将退出,并返回错误代码3,以示这个情况与致命错误条件的区别,致命错误条件的错误代码为1。不管在哪种情况下,任何当前运行的脚本(顶级脚本,如果有,和任何它调用的其他脚本)都将立即终止。如果顶级命令字符串包含多个SQL命令,处理将在当前命令停止。

#### PORT

当前你正在连接的数据库服务器的端口。这是在每次你与数据库连接时(包括程序启动)设置的,但是可以取消设置。

#### PROMPT1

#### PROMPT2

#### PROMPT3

这些指明psql显示的提示符如何显示。参阅下面的提示符。

#### QUIET

设置这个变量为on,等效于命令行选项-q。可能在交互模式下没有什么用。

#### **SINGLELINE**

此变量等效于命令行选项-S。

#### **SINGLESTEP**

将此变量设置为on等效于命令行选项-s。

#### **USER**

当前连接的数据库用户。每次您连接到数据库(包括程序启动)时都会进行设置,但是可以取消设置。

#### **VERBOSITY**

可以将此变量设置为default, verbose或terse值,以控制错误报告的详细程度。

## SQL代换

psql变量的关键特性是你可以把它们替换成正规的SQL语句,也可以是元命令的参数。 另外,psql提供工具确保变量值用作SQL文本并且正确的引用标识符。 不用引用替换一个值的语法是在变量名前面加一个冒号(:)。例如:

testdb=> \set foo 'my\_table'
testdb=> SELECT \* FROM :foo;

将会查询my\_table表。请注意,这可能是不安全的:变量的值是逐字拷贝的,所以它甚至可以包含不对称的引号或反斜杠命令。 你必须保证你输入的东西是有意义的。

当一个值被用作SQL文本或标识符时,将其引用是最安全的。要引用一个变量的值作为SQL文本,在单引号中的变量名后面写一个冒号。要引用值作为SQL标识符,在双引号中的变量名后面写一个冒号。这些构造正确的处理引号和嵌入在变量值中的其他特殊字符。上面的例子这样写更安全:

testdb=> \set foo 'my\_table'
testdb=> SELECT \* FROM :"foo";

变量替换将不在引用的SQL文本和标识符中执行。 因此,一个构造比如':foo'并不从变量的值中产生一个引用的文本 (如果它确实发生了,那么它是不安全的,因为它不会正确的处理嵌入在值中的引号。)

使用这个机制的一个例子是拷贝一个文件的内容到一个表字段中。 首先加载文件到一个变量,然后替换变量的值为引用的字符串:

testdb=> \set content `cat my\_file.txt`
testdb=> INSERT INTO my\_table VALUES (:'content');

(注意,如果my\_file.txt包含NUL字节,则仍然无法使用。 psql在变量值中不支持嵌入的NUL字节。)

由于冒号可以合法地出现在SQL命令中,因此除非当前设置了命名变量,否则不会替换明显的插值尝试(即:name,:'name'或:''name'')。 无论如何,您都可以使用反斜杠转义冒号,以防止其被替换。

变量的冒号语法符合诸如ECPG之类的嵌入式查询语言的SQL标准。 数组切

片和类型转换的冒号语法是MPP数据库扩展,有时可能与标准用法冲突。 将变量值转义为SQL文字或标识符的冒号引用语法是psql扩展。

## 提示符

可以根据您的喜好自定义psql提示符。三个变

量PROMPT1, PROMPT2和PROMPT3包含字符串和特殊的转义序列,用于描述提示符的外观。PROMPT1是psql请求新命令时显示的普通提示。当在命令输入期间需要更多输入时(例如,因为命令未使用分号终止或未关闭引号),将显示PROMPT2。当您运行SQL COPY FROM STDIN命令并且您需要在终端上输入行值时,将显示PROMPT3。

所选提示变量的值按字面意义打印,除非遇到百分号(%)。 根据下一个字符,某些其他文本将被替换。 定义的替换是:

#### %M

数据库服务器的完整主机名(带有域名),如果连接是通过UNIX域套接字的,则为[local];如果UNIX域套接字的默认设置为未编译的默认位置,则为[local:/dir/name]。

%m

数据库服务器的主机名,在第一个点处被截断,如果连接是通过UNIX 域套接字的,则为[local]。

%>

数据库服务器正在侦听的端口号。

%n

数据库会话用户名。 (此值的扩展可能在数据库会话期间由于命令SET SESSION AUTHORIZATION的结果而改变。)

%/

当前数据库的名称。

%~

与%/类似,但是如果数据库是您的默认数据库,则输出为~(波浪号)。

%#

如果会话用户是数据库超级用户,则为#,否则为>。 (此值的扩展可

能在数据库会话期间由于命令SET SESSION AUTHORIZATION的结果而改变。)

%R

在PROMPT1中通常为=,但在单行模式下为^或!如果会话与数据库断开连接(如果\connect失败,则可能发生)。在PROMPT2中,%R被一个字符替换,该字符取决于psql期望更多输入的原因:-如果命令尚未终止;\*如果有未完成的/\*...\*/注释;',如果有一个未完成的带引号的标识符;\$,如果有一个未完成的带引号的标识符;\$,如果有一个未完成的带美元引号的字符串;或者(,如果有一个不匹配的左括号。在PROMPT3中,%R不产生任何东西。

%X

事务状态:不在事务块中时为空字符串;在事务块中时为\*;或!当事务失败时;或?当事务状态不确定时(例如,因为没有连接)。

## %digits

指示的八进制代码的字符被替换。

#### %:name:

psql变量name的值。 参阅高级特性中变量小节获取更多信息。

### %`command`

command的输出,类似于通常的反引号替换。

## %[...%]

提示可能包含终端控制字符,例如,这些字符会更改提示文本的颜色,背景或样式,或更改终端窗口的标题。 为了使行编辑正常工作,必须将这些非打印控制字符用%[和%]括起来以指定为不可见。 提示中可能会出现多个这种配对。例如,

testdb=> \set PROMPT1 '%[%033[1;33;40m%]%n@%/%R%[%033[0m%]%#'

会在兼容VT100的有颜色的终端上显示黑底黄字(33;40)的粗体(1;)。要将百分号插入提示中,请输入%%。对于提示1和2,默认提示为'%/%R%#',对于提示3,默认提示为'>>'。

## 命令行编辑

psql支持NetBSD libedit库,用于方便的行编辑和检索。 当psql退出时,命

令历史记录将自动保存,而在psql启动时,将重新加载命令历史记录。 尽管补全逻辑不声称是SQL解析器,但也支持制表符补全。 制表符补全生成的查询也可能会干扰其他SQL命令,例如SET TRANSACTION ISOLATION LEVEL。 如果由于某种原因您不喜欢制表符补全,可以通过将其放在主目录中名为.inputrc的文件中来将其关闭:

\$if psql set disable-completion on \$endif

## 环境变量

#### **COLUMNS**

如果\pset columns为零 ,则控制换行格式的宽度和以确定宽度输出是需要分页器还是应在扩展自动模式下切换为垂直格式的宽度。

#### **PAGER**

如果查询结果无法显示在屏幕上,则通过此命令通过管道传递查询结果。 典型值为more或less。默认值取决于平台。 可以通过将 *PAGER*设置为空或使用\pset命令中与分页器相关的选项来禁用该分页器的使用。

**PGDATABASE** 

**PGHOST** 

**PGPORT** 

**PGUSER** 

默认连接参数。

**PSQL EDITOR** 

**EDITOR** 

**VISUAL** 

\e和\ef命令使用的编辑器。 按照列出的顺序检查变量;使用设置的第一个。

内置的默认编辑器在Unix系统上为vi,在Windows系统上

为notepad.exe。

## **PSQL EDITOR LINENUMBER ARG**

当\e或\ef与行号参数一起使用时,此变量指定用于将起始行号传递给用户的编辑器的命令行参数。 对于Emacs或vi等编辑器,这是一个加号。 如果选项名称和行号之间需要有空格,请在变量的值中包含尾随空格。例子:

PSQL\_EDITOR\_LINENUMBER\_ARG='+'
PSQL\_EDITOR\_LINENUMBER\_ARG='--line'

在Unix系统上,默认值为+(与默认编辑器vi对应,对许多其他常用编辑器很有用);但Windows系统上没有默认设置。

#### **PSQL HISTORY**

命令历史记录文件的备用位置。进行波浪(~)扩展。

#### **PSQLRC**

用户的.psqlrc文件的备用位置。进行波浪(~)扩展。

#### SHELL

命令由\!命令执行。

#### **TMPDIR**

用于存储临时文件的目录。默认值为/tmp。

## 文件

## psqlrc和~/.psqlrc

除非传递了-X或-c选项,否则psql在连接数据库之后接受正常的命令之前,尝试从系统范围的启动文件(psqlrc),然后用户的启动文件(~/.psqlrc)读取并执行命令。

系统范围的启动文件名为psqlrc,可在安装的"系统配置"目录中查找,该目录可以通过运行pg\_config --sysconfdir来准确获取。 默认情况下,相对于包含MPP数据库可执行文件的目录,该目录为 ../etc/。可以通过PGSYSCONFDIR环境变量显式设置此目录的名称。

用户的个人启动文件名为.psqlrc,在用户的主目录中查找。 在缺少这

种概念的Windows上,个人启动文件名

为%APPDATA%\postgresql\psqlrc.conf。 可以通过PSQLRC环境变量显式设置用户启动文件的位置。

通过在文件名后附加破折号和基础的PostgreSQL主要或次要发行版号 (例如~/.psqlrc-9.4) ,可以使系统范围内的启动文件和用户的个人 启动文件都特定于psql版本。 与非版本特定的文件相比,将优先读取最特定的版本匹配文件。

## .psql\_history

命令行历史存储在~/.psql\_history文件中,或者在Windows里是%APPDATA%\postgresql\psql\_history文件。

历史文件的位置可以通过PSQL\_HISTORY环境变量来明确设置。

## 注解

psql与相同或较旧的主要版本的服务器一起使用时效果最佳。如果服务器的版本比psql本身新,则反斜杠命令很可能会失败。但是,\d系列的反斜杠命令可以与较旧的服务器版本一起使用,尽管不一定与psql本身更新的服务器一起使用。 运行SQL命令和显示查询结果的常规功能也应与较新的主要版本的服务器一起使用,但这不能在所有情况下都得到保证。

如果要使用psql连接到多个具有不同主要版本的服务器,则建议使用最新版本的psql。 另外,您可以保留每个主要版本的psql副本,并确保使用与各自服务器匹配的版本。 但是在实践中,这种额外的复杂性应该没有必要。

## Windows用户注解

psql被构建为控制台应用程序。由于Windows控制台窗口使用与系统其余部分不同的编码,因此在psql中使用8位字符时必须格外小心。如果psql检测到有问题的控制台代码页,它将在启动时警告您。要更改控制台代码页,需要做两件事:

## 通过输入以下内容来设置代码页:

```
cmd.exe /c chcp 1252
```

1252是拉丁字母的字符编码,Microsoft Windows将其用于英语和某些其他西方语言。 如果使用的是Cygwin,则可以将此命令放在/etc/profile中。

将控制台字体设置为Lucida Console,因为栅格字体不适用于ANSI代码页。

## 示例

以交互方式启动psql:

```
psql -p 54321 -U sally mydatabase
```

在psql交互方式下,将命令分布在多行输入上。注意更改提示:

```
testdb=> CREATE TABLE my_table (
testdb(> first integer not null default 0,
testdb(> second text)
testdb-> ;
CREATE TABLE
```

## 查看表定义:

## 通过传入包含SQL命令的文件,以非交互方式运行psql:

psql -f /home/gpadmin/test/myscript.sql

## reindexdb

重建数据库中的索引。

## 概要

```
reindexdb [connection-option ...] [--table | -t table ]
        [--index | -i index ] [dbname]

reindexdb [connection-option ...] --all | -a

reindexdb [connection-option ...] --system | -s [dbname]

reindexdb -? | --help

reindexdb -V | --version
```

## 描述

reindexdb是用于在MPP数据库中重建索引的工具。

reindexdb是SQL命令REINDEX的包装。 通过此工具和通过其他访问服务器的方法为数据库重新索引之间没有明显的区别。

## 选项

-a | --all

重新索引所有数据库。

[-d] dbname | [--dbname=]dbname

指定要重新建立索引的数据库的名称。 如果未指定并且未使用-all,则从环境变量PGDATABASE读取数据库名称。 如果未设置,则使用为连接指定的用户名。

-e | --echo

回显reindexdb生成并发送到服务器的命令。

-i *index* | --index=*index* 

仅重新创建索引。

-q | --quiet

不显示响应。

-s | --system

重新索引系统catalog。

-t table | --table=table

仅重新索引表。可以通过编写多个-t开关为多个表重新索引。

-V | --version

打印reindexdb版本并退出。

-? | --help

显示有关reindexdb命令行参数的帮助,然后退出。

## 连接选项

-h host | --host=host

运行MPP master数据库服务器的计算机的主机名。 如果未指定,则从环境变量PGHOST读取或默认为localhost。

-p port | --port=port

MPP master数据库服务器正在侦听连接的TCP端口。 如果未指定,则从环境变量PGPORT读取或默认为5432。

## -U username | --username=username

要用作连接的数据库角色名称。 如果未指定,则从环境变量 PGUSER读取或默认为当前系统角色名称。

## -w | --no-password

不发出密码提示。如果服务器要求密码验证,而其他方式(例如.pgpass文件)无法使用密码,则连接尝试将失败。此选项在没有用户输入密码的批处理作业和脚本中很有用。

## -W | --password

强制输入密码提示。

#### --maintenance-db=dbname

指定要连接以发现应重新索引哪些其他数据库的数据库的名称。 如果未指定,将使用postgres数据库,如果不存在,将使用template1。

## 注解

reindexdb可能需要连接几次master服务器,每次都要求输入密码。 在这种情况下,使用~/.pgpass文件很方便。

## 示例

要重新索引数据库mydb:

reindexdb mydb

要在名为abcd的数据库中为表foo和索引bar重建索引:

reindexdb --table foo --index bar abcd

## 另见

### **REINDEX**

## vacuumdb

垃圾收集并分析数据库。

## 概要

```
vacuumdb [connection-option...] [--full | -f] [--freeze | -F] [--verbose | -v]
    [--analyze | -z] [--analyze-only | -Z] [--table | -t table [( column [,...] )] ] [dbname]

vacuumdb [connection-option...] [--all | -a] [--full | -f] [-F]
    [--verbose | -v] [--analyze | -z]
    [--analyze-only | -Z]

vacuumdb -? | --help

vacuumdb -V | --version
```

## 描述

vacuumdb是用于清理MPP数据库的工具。 vacuumdb还将生成MPP数据库

查询优化器使用的内部统计信息。

vacuumdb是SQL命令VACUUM的包装。 通过此工具和通过其他访问服务器的方法来清理数据库之间没有明显的区别。

## 选项

#### -a | --all

清理所有数据库。

## [-d] dbname | [--dbname=]dbname

要清理的数据库的名称。 如果未指定此选项,并且未使用-a(或--all),则从环境变量PGDATABASE读取数据库名称。 如果未设置,则使用为连接指定的用户名。

### -e | --echo

回显vacuumdb生成并发送到服务器的命令。

#### -f | --full

选择完全清理,这可以回收更多空间,但是需要更长的时间并排他锁定表。

Warning: 在MPP数据库中不建议使用VACUUM FULL。

## -F | --freeze

冻结行事务信息。

## -q | --quiet

不显示响应。

## -t table [(column)] | --table=table [(column)]

仅清理或分析该表。 只能与--analyze或--analyze-all选项一起指定列名。 可以通过写入多个-t开关来清理多个表。 如果指定列,则可能必须从shell转义括号。

## -v | --verbose

在处理过程中打印详细信息。

## -z | --analyze

收集统计信息以供查询优化器使用。

## -Z | --analyze-only

仅计算统计信息以供查询优化器使用(不清理)。

#### -V | --version

打印vacuumdb版本并退出。

#### -? | --help

显示有关vacuumdb命令行参数的帮助,然后退出。

## 连接选项

## -h host | --host=host

运行MPP master数据库服务器的计算机的主机名。 如果未指定,则从环境变量PGHOST读取或默认为localhost。

## -p port | --port=port

MPP master数据库服务器正在侦听连接的TCP端口。 如果未指定,则从环境变量PGPORT读取或默认为5432。

## -U username | --username=username

要用作连接的数据库角色名称。 如果未指定,则从环境变量PGUSER读取或默认为当前系统角色名称。

## -w | --no-password

不发出密码提示。如果服务器要求密码验证,而其他方式(例如.pgpass文件)无法使用密码,则连接尝试将失败。 此选项在没有用户输入密码的批处理作业和脚本中很有用。

## -W | --password

强制输入密码提示。

## --maintenance-db=dbname

指定要连接以发现应清理哪些其他数据库的数据库名称。 如果未指定,将使用postgres数据库,如果不存在,将使用template1。

## 注解

vacuumdb可能需要连接几次master服务器,每次都要求输入密码。 在这种情况下,使用~/.pgpass文件很方便。

## 示例

清理数据库test:

vacuumdb test

清理分析数据库bigdb:

vacuumdb --analyze bigdb

要清理名为mydb的数据库中的单个表foo,并分析表的单个列bar。 请注意表名和列名两边的引号,从shell转义括号:

vacuumdb --analyze --verbose --table 'foo(bar)' mydb

## 另见

**VACUUM, ANALYZE** 

# 附加程序

本节介绍MPP数据库安装后的可用附加程序

以下PostgreSQL contrib服务器工具程序被安装:

- pg\_upgrade 用来升级MPP数据库服务器实例的服务端程序。
   Note: pg\_upgrade目前不会被MPP 6使用,但是在未来版本会被MPP升级工具使用。
- pg\_upgrade\_support pg\_upgrade支持库。
- pg\_xlogdump 显示MPP数据库 WAL日志为人类可读形式的服务器端程序。

Parent topic: MPP数据库工具手册