# GreenPlum 使用指南

连接数据库

psql -d postgres

## 集群部署

### 1.安装依赖库

所有节点执行

# yum -y install rsync coreutils glib2 lrzsz sysstat e4fsprogs xfsprogs ntp readline-devel zlib zlib-devel openssl openssl-devel pam-devel libxml2-devel libxslt-devel python-devel tcl-devel gcc make smartmontools flex bison perl perl-devel perl-ExtUtils\* OpenIPMI-tools openldap openldap-devel logrotate python-py gcc-c++ libevent-devel apr-devel libcurl-devel bzip2-devel libyaml-devel

easy\_install pip

pip install paramiko

pip install psutil

pip install lockfile

### 2.修改内核参数

# vi /etc/sysctl.conf

#net.core.netdev\_max\_backlog = 2500->10000

#add by langke install greenplum

kernel.shmmni = 4096

kernel.sem = 50100 64128000 50100 1280

kernel.msgmni = 2048

net.ipv4.conf.all.arp\_filter = 1

net.ipv4.netfilter.ip\_conntrack\_max = 655360

net.ipv4.tcp\_keepalive\_probes = 9

net.ipv4.tcp\_keepalive\_intvl = 7

vm.overcommit\_memory = 2

fs.file-max = 7672460

fs.aio-max-nr = 1048576

上面这个内核参数设置之后影响elasticsearch进程（进程自动退出，报OOM：Native memory allocation (mmap) failed to map 11314593792 bytes for committing reserved memory,可能是vm.overcommit\_memory，参数影响）

kernel.shmmni = 4096

kernel.sem =250 32000 32 128

kernel.msgmni = 32768

net.ipv4.conf.all.arp\_filter = 0

net.ipv4.tcp\_keepalive\_probes = 9

net.ipv4.tcp\_keepalive\_intvl = 75

vm.overcommit\_memory = 0

fs.file-max = 7672460

fs.aio-max-nr = 1048576

# sysctl -p

受内核参数影响kafka java客户端推送数据也受影响，soa\_logs队列推送很慢 ，内核参数一直调整无效，最终迁移kafka节点：所以内核参数最好还是不要修改

# rm -f /etc/security/limits.d/90-nproc.conf

# vi /etc/security/limits.conf

\* soft nproc 204800

\* hard nproc 204800

\* soft memlock unlimited

\* hard memlock unlimited

设置块设备预读大小

/sbin/blockdev --setra 16384 /dev/sda1

/sbin/blockdev --setra 16384 /dev/sdb1

/sbin/blockdev --setra 16384 /dev/sdc1

/sbin/blockdev --setra 16384 /dev/sdd1

### 3.初始化安装环境

#切换到hadoop用户

su hadoop

master节点:

mkdir -p /data/disk1/gp/master

chown -R hadoop /data/disk1/gp/master

chmod 700 /data/disk1/gp/master

所有 segment 节点:

mkdir -p /data/disk1/gp/data

mkdir -p /data/disk2/gp/data

mkdir -p /data/disk3/gp/data

mkdir -p /data/disk4/gp/data

mkdir -p /data/disk1/gp/mirror

mkdir -p /data/disk2/gp/mirror

mkdir -p /data/disk3/gp/mirror

mkdir -p /data/disk4/gp/mirror

chown -R hadoop /data/disk1/gp/data

chmod -R 700 /data/disk1/gp/data

### 4.安装：

./greenplum-db-4.3.9.1-build-1-rhel5-x86\_64.bin

安装到/opt/17173\_install/greenplum-db-4.3.9.1

安装完成

cd greenplum-db/bin

./pg\_config

cd ..

cat greenplum\_path.sh

添加到用户的环境变量

cat greenplum\_path.sh >> ~/.bash\_profile

. ~/.bash\_profile

创建主机文件，包括所有节点以及主节点本身

vi host

sea2

sea3

sea4

sea5

sea6

交换KEY，master 访问所有的segment不需要输入密码，master pub拷贝到所有的segment authorized\_keys,安装过hadoop可以略过

gpssh-exkeys -f ./host

安装软件到segment hosts

gpseginstall -f ./host -u hadoop

### 5.初始化数据库

配置文件

cp docs/cli\_help/gpconfigs/gpinitsystem\_config ./

chmod 644 ./gpinitsystem\_config

vi ./gpinitsystem\_config

declare -a DATA\_DIRECTORY=(/data/disk1/gp/data /data/disk2/gp/data /data/disk3/gp/data /data/disk4/gp/data)

MASTER\_HOSTNAME=sea6

MASTER\_DIRECTORY=/data/disk1/gp/master

MIRROR\_PORT\_BASE=50000

REPLICATION\_PORT\_BASE=41000

MIRROR\_REPLICATION\_PORT\_BASE=51000

declare -a MIRROR\_DATA\_DIRECTORY=(/data/disk1/gp/mirror /data/disk2/gp/mirror /data/disk3/gp/mirror /data/disk4/gp/mirror)

DATABASE\_NAME=sea

MACHINE\_LIST\_FILE=/opt/17173\_install/greenplum-db/host\_segment

编辑主机文件，不要包含master, standby，除非master,standby节点也需要当segment node使用.

vi host\_segment

sea2

sea3

sea4

sea5

sea6

初始化数据库

gpinitsystem -c ./gpinitsystem\_config

### 6.增加standby

一个gp集群只有一个master肯定会让人不放心，还好有备用，当master宕掉后，会自动启用standby作为master，下面来看一下standby怎么添加

在standby服务器上执行,sea5作为standby

mkdir /data/disk1/gp/master

chown hadoop /data/disk1/gp/master

在master服务器上执行

gpinitstandby -s sea5

中间输入一次Y

### 7.增加mirror

mirror就是镜像，也叫数据备份。mirror对于数据存储来说很重要，因为我们的服务器指不定什么时候出毛病，有mirror就好很多了，因为两台存储节点同时宕掉的几率还是很小的。如果前面在GP初始化文件里忘记配置mirror了，请按照下面的方法添加

.bashrc和.bash\_profile最后都添加下面两行

source /opt/17173\_install/greenplum-db/greenplum\_path.sh

export MASTER\_DATA\_DIRECTORY=/data/disk1/gp/master/gpseg-1

设置完后记得source一下使其立即生效

gpaddmirrors -p 1000

运行过程中需要输入两次mirror路径：/data/disk1/gp/mirror

出现异常：

-gpaddmirrors failed. (Reason='FATAL: System was started in master-only utility mode - only utility mode connections are allowed

vi env\_gp.sh

export MASTER\_DATA\_DIRECTORY=/data/disk1/master/gpseg-1

export PGHOST=127.0.0.1

export PGPORT=5432

export PGUSER=hadoop

export PGDATABASE=sea

. ./env\_gp.sh

### 8.尝试连接到postgres数据库

psql -d postgres

postgres=# select datname,datdba,encoding,datacl from pg\_database;

创建数据库，创建表

createdb sea -E utf-8

psql -d sea

select version();

create table test01(id int primary key,col1 varchar(50));

insert into test01 select 1,'sea' ;

select \* from test01;

### 9.创建用户、修改用户、删除用户

CREATE ROLE user01 WITH LOGIN;

ALTER ROLE user01 WITH PASSWORD '111111';

GRANT ALL ON DATABASE sea TO user01;

取消授权：

REVOKE ALL ON DATABASE sea from user01;

CREATE USER user01 WITH PASSWORD '111111' NOSUPERUSER;

\du

CREATE ROLE users;

GRANT users TO user01;

GRANT ALL PRIVILEGES ON DATABASE sea TO user01;

psql: FATAL: no pg\_hba.conf entry for host

修改：/data1/gpdata/master/gpseg-1/pg\_hba.conf

host all all 10.5.15.222/32 trust

重新加载配置：select pg\_reload\_conf();

### 10.数据库启动关闭

gpstart

gpstop

gpstate

也可以用gpstat来查看当前的状态，此命令也是排错时的必备工具

gpstat -e #查看mirror的状态，我们这次的案例没有安装mirror

gpstat -f #查看standby master的状态

gpstat -s #查看整个GP群集的状态

gpstat -i #查看GP的版本

gpstat --help #帮助文档，可以查看gpstat更多用法，

### 二、GreenPlum安装性能监控工具GPCCC（Greenplum Command Center Console）

1.performance monitor安装

使用gpperfmon\_install命令，Greenplum安装完成后已经包含该命令,安装后会建立名为gpperfmon的数据库，默认使用gpmon用户。

gpperfmon\_install --enable --password gpmon --port 5432

然后重启数据库，-r的含义是restart

gpstop -r

2.确认监控采集进程是否已经启动

ps -ef|grep gpmmon|grep -v grep

#确认Performance Monitor数据库写入数据是否正常，检查是否有记录写入

psql -d gpperfmon -c 'select \* from system\_now'

如果配置了master的standby,拷贝Master主机拷贝配置文件到Standby Master的相应目录.

gpscp -h sea5 /data/disk1/gp/master/gpseg-1/pg\_hba.conf =:$MASTER\_DATA\_DIRECTORY/

gpscp -h sea5 ~/.pgpass =:~/

3.安装Greenplum Command Center Console

下载地址 <https://network.pivotal.io/products/pivotal-gpdb>

3.1安装gpccc

./greenplum-cc-web-1.3.0.0-build-91-RHEL5-x86\_64.bin

输入performance monitor的安装目录如 /opt/17173\_install/greenplum-cc-web-2.4.0

chown -R hadoop greenplum-cc-web-2.4.0

chown -R gpadmin:gpadmin greenplum-cc-web

gpccc和gpdb一样，都会创建一个软链接，分别是greenplum-db和greenplum-cc-web.

3.2设置环境变量

source greenplum-cc-web/gpcc\_path.sh

或者vi ~/.bashrc ~/.bash\_profile

source /opt/17173\_install/greenplum-db/greenplum\_path.sh

source /opt/17173\_install/greenplum-cc-web/gpcc\_path.sh

source ~/.bashrc

3.3 在所有主机安装GPCCC

这里的all\_host文件就采用前面安装GPDB的时候的所有节点的文件

gpccinstall -f /opt/17173\_install/greenplum-db/host

#注意在所有机器上配置环境变量，或者使用gpscp 将一台机器的文件copy到其他机器

vi .bashrc

source /opt/17173\_install/greenplum-cc-web/gpcc\_path.sh

3.4 配置安装详细项

cd /opt/17173\_install/greenplum-cc-web-2.4.0

gpcmdr --setup

Please enter a new instance name:输入gpcc

Is the master host for the Greenplum Database remote? Yy|Nn (default=N):n

What would you like to use for the display name for this instance:gpcc

What port does the Greenplum Database use? (default=5432):回车

will you install workload manaager (default=N):Y

What port would you like the web server to use for this instance?回车

Do you want to enable SSL for the Web API Yy|Nn (default=N):n

Do you want to copy the instance to a standby master host Yy|Nn (default=Y): 没有standby n

What is the hostname of the standby master host? [sea6]:回车

3.5 启动及相关操作

启动实例：

gpcmdr --start gpcc

查看端口状态： lsof -i :28080

发现是lighttpd

浏览：<http://sea6:28080/>

4.相关注意事项

4.1重启应用的时候，如果有其他链接，可以强制重启：gpstop -M immediate

4.2 错误error：

[1.no](http://1.no/) pg\_hba.conf entry for host “::1”, user “gpmon”, database “gpperfmon”, SSL off

解决：

vi pg\_hba.conf增加：

host gpperfmon gpmon ::1/128 trust (此处的trust应该为md5，否则后面会报错)

重新加载配置：select pg\_reload\_conf();

ERROR: relation "gp\_toolkit.\_\_gp\_log\_master\_ext" does not exist

报错为ERROR: relation "gp\_toolkit.\_\_gp\_log\_master\_ext" does not exist，数据库gppfermon中gp\_toolkit schema中无内容，可能是template0,1中的gp\_toolkit内容丢失；没办法，手动补：

psql -f $GPHOME/share/postgresql/gp\_toolkit.sql gpperfmon

3.登陆web提示：trust login is disabled.trust user gpmon is not allowed to login Command Center

描述：

用psql -d gpperfmon -U gpmon -W 输入密码可以正常登陆

查看日志：/home/greenplum-4.3-cc/instances/gpcc2/logs/gpmonws.log

修改pg\_hba.conf将其中的trust修改为md5

source /usr/local/greenplum-db/greenplum\_path.sh 不要写到profile或者./bashrc文件中

因为该句会导致python环境变量的改变，致使yum不能正常使用，出现no module named yum问题

解决办法：使用是直接运行不添加，或者新建立用户中，在新用户家目录的bashrc文件中添加该环境变量，当然该用户不能使用yum命令。

root用户使用su -登入不带环境变量

## segment数量定义

1. 和CPU核数，块设备数量，RAID卡通道数量，网卡数量都有关系。  
   要做到均衡配比，资源的利用才能最合理。  
   你可以这样来分配，例如CPU核数乘以0.8作为segment数量，然后再均匀分配SEG对应的块设备，网卡，RAID卡，
2. 1.考虑服务器是几路服务器：x86服务器是2路甚至4路的话，建议1路（即一个NUMA节点）至少对应一个，并且1个segment不要跨NUMA  
   2.考虑服务器具体芯片的处理能力（不仅仅看core数），好点的CPU还是有好处的，好的CPU一个NUMA节点可以对应几个segment  
   一定要注意RAID卡、网卡，多配几块有好处的。