Greenplum 集群扩容总结

1 概述

Greenplum 集群扩容一般分为在原始集群上加集群或者重新搭建新的集群并把数据重新导入到新的集群中。在原始集群扩容时一般扩容的 segment 是原始集群 segment 机器的倍数关系，例如源集群的规模是 1 台 master1 台 stadby 3 台 segment 节点，那么需要准备 3 台 segment 或者 3 的倍数的机器。而使用新的集群则需要关注新的机器的配置即可。

2 扩容前准备

在集群扩容前需要先检查集群没有任何的链接，如果在扩容时有应用链接集群，可能会出现元数据的丢失或损坏，导致集群无法启动或损坏集群。

3 扩容方案对比

3.1 使用 gpexpand 进行数据库扩容

优点

1、官方给出的扩容组件，操作相对简单
2、扩容效率高，不需要关注元数据迁移问题
3、只有 segment 初始化和表重分布两个阶段执行系统扩展

缺点

1、需要与源 segment 相同配置的机器，并且新的机器的数量是源 segment 机器个数的倍数关系
2、对集群环境要求苛刻，如果有不符合扩容条件直接报错
3、可以查看数据平衡的进度，但无法干预数据平衡的时间
4、不能解决系统表倾斜的问题
5、集群操作期间无法提供服务

3.2 新建 Greenplum 集群，重新导入数据

优点

1、使用同步工具，可以把数据同步到新的集群中，可以使用批量方式同步，加快同步速度
2、同步的元数据信息和数据都可以手动控制，可以过滤掉不需要的元数据信息和表
3、由于表和索引是在新的集群上重新创建，可以解决表和索引的膨胀问题
4、如果在新的集群上改变表的分布键可以解决表的倾斜的情况

缺点

1、新的集群需要满足旧集群的所有用户自定义的配置
2、如果新的集群升级新的版本需要注意兼容性的问题
3、数据导入新的集群效率低，受 master 节点的硬件影响
4、由于是人工干预操作，数据量大，耗时比较长
5、集群的硬件设备会影响数据同步的性能

4 查看集群的基本信息

4.1 查看集群的版本

```
select     version();
```

```
PostgreSQL 9.4.24 (Greenplum Database 6.7.0 build
commit:2fbc274bc15a19b5de3c6e44ad5073464cd4f47b) on x86_64-unknown-linux-gnu,
compiled by gcc (GCC) 6.4.0, 64-bit compiled on Apr 16 2020 02:24:06
```

4.2 查看 master 和 stadby 信息

使用 select * from gp_segment_configuration where content = -1 命令查看 master 配置信息，
使用 gpstate -f 命令查看 standby 配置信息

4.3 查看 segment 信息

gpstate -m
在输出的中 Type = Group 是 mirror 的数据分配方式

4.4 当前集群链接检查

select * from pg_stat_activity where state != 'idle';

如果以上返回结果，应该关闭应用端的所有请求，避免在扩容的过程中有异常发生。

4.5 查看集群的运行状态信息

-- 检查 segment 的 down 的节点
select * from gp_segment_configuration where status='d';

-- 检查 segment 正在做日志和重新同步的节点
select * from gp_segment_configuration where mode= 'c' or mode = 'r';

4.6 使用 gpstate 命令查看集群信息

使用以下的命令查看集群所有的配置信息
gpstate -s

在 master 节点上中的 catalog 快速查看 down 掉的 segment 节点
gpstate -Q

查看所有节点的软件版本信息
gpstate -i

5 配置新的 segment 节点

5.1 操作系统配置

5.1.1 关闭防火墙

关闭防火墙
service iptables stop
chkconfig iptables off
setenforce 0

vi /etc/selinux/config
SELINUX=disabled

查看防火墙的状态
service iptables status
sestatus

5.1.2 修改/etc/sysctl.conf 文件

在以下文件末尾追加以下配置
vim /etc/sysctl.conf
****

kernel.shmmax = 34000000000
kernel.shmmni = 8192
kernel.shmall = 34000000000
kernel.sem = 1000 8192000 400 8192
kernel.sysrq = 1
kernel.core_uses_pid = 1
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.msgmni = 2048
net.ipv4.tcp_syncookies = 1
net.ipv4.ip_forward = 0
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.tcp_tw_recycle = 1

net.ipv4.tcp_max_syn_backlog = 4096
net.ipv4.conf.all.arp_filter = 1
net.ipv4.ip_local_port_range = 1025 65535
net.core.netdev_max_backlog = 10000
net.core.rmem_max = 2097152
net.core.wmem_max = 2097152
vm.overcommit_memory = 2
vm.swappiness = 1
kernel.pid_max = 655350

5.1.3 文件修改/etc/security/limits.conf

在文件/etc/security/limits.conf 追加以下配置

****
End of file

* soft nofile 65536
* hard nofile 65536
* soft nproc 131072
* hard nproc 131072

5.1.4 文件修改/etc/security/limits.d/90-nproc.conf（针对RedHat6.x 系统）

在文件/etc/security/limits.d/90-nproc.conf 追加以下配置
****

End of file
* soft nofile 1048576
* hard nofile 1048576
* soft nproc 1048576
* hard nproc 1048576

5.1.5 文件修改/etc/hosts

在/etc/hosts 文件追加需要添加集群的名字

192.168.***.** gpmaster
192.168.***.** gpsdw1
192.168.***.** gpsdw2
192.168.***.** gpsdw3

追加需要添加的集群的名字

5.1.6 修改主机名字

vim /etc/sysconfig/network
HOSTNAME=hostname

Hostname： 当前机器的名字

5.1.7 禁用内存大页

vim /etc/rc.local #追加，禁用大页

```
if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi
if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
```

5.1.8 服务器时间同步

备份原始文件
cp /etc/ntp.conf /etc/ntp.conf.bak

添加需要参考的主机集群时间

vim /etc/ntp.conf
server mdw

mdw : 参考时间机器的名字

启动服务
chkconfig ntpd on
service ntpd restart

查看时间
date -R

5.2 软件需要环境配置

以下操作全部在 master 节点操作

5.2.1 创建主机映射文件

创建 all_hosts 文件，包含所有机器的 IP 与名字映射的文件

vim all_hosts
192.168.***.**
192.168.***.**
192.168.***.**
192.168.***.**
192.168.***.**
192.168.***.**
192.168.***.**
192.168.***.**

创建 new_nodes 文件，包含新的机器的 IP 与名字映射的文件
192.168.***.**
192.168.***.**
192.168.***.**

创建 exist_nodes 文件，包含新的机器的 IP 与名字映射的文件
192.168.***.**
192.168.***.**
192.168.***.**

192.168.***.**
192.168.***.**

5.2.2 把所有的机器进行相互免密

在 master 节点上进行新机器的相互免密操作

su - gpadmin

gpssh-exkeys -e exist_nodes -x new_nodes

5.2.3 集群之间同步时钟

gpssh -f all_host_file -v -e 'ntpd'

5.2.4 在新的 segment 节点上创建用户与文件夹

在新的 segment 机器上创建 gpadmin 用户，并且修改密码为 gpadmin

```
gpssh -f new_nodes -v -e "groupadd -g 530 gpadmin";
gpssh -f new_nodes -v -e "useradd -g 530 -u 530 -m -d /home/gpadmin -s /bin/bash gpadmin";
gpssh -f new_nodes -v -e "chown -R gpadmin:gpadmin /home/gpadmin";
gpssh -f new_nodes -v -e "echo "gpadmin" | passwd --stdin gpadmin";
```

在新的 segment 机器上创建目录并付给权限

```
gpssh -f new_nodes -v -e “mkdir -p /data1/primary”;
gpssh -f new_nodes -v -e “mkdir -p /data1/mirror”;

gpssh -f new_nodes -v -e “mkdir -p /data2/primary”;
gpssh -f new_nodes -v -e “mkdir -p /data2/mirror”;

gpssh -f new_nodes -v -e “chown -R gpadmin:gpadmin /data1”;
gpssh -f new_nodes -v -e “chown -R gpadmin:gpadmin /data2”;
```

5.2.5 所有的机器进行时间同步

gpssh -f all_host_file -v -e 'ntpd'

ntpd 有一个自我保护设置：如果本机与上源时间相差太大, ntpd 不运行．所以新设置的时间服务器一定要先 ntpdate 从上源取得时间初值，然后启动 ntpd 服务。ntpd 服务运行后，先是每 64 秒与配置服务器同步一次，根据每次同步时测得的误差值经复杂计算逐步调整自己的时间，随着误差减小，逐步增加同步的间隔．每次跳动，都会重复这个调整的过程。

检测集群时间
gpssh -f all_hosts -v -e "date -R "

5.2.6 重启新的 segment 机器

reboot

6 新的 segment 节点软件安装与集群统计

6.1 在新的 segment 机器上安装软件包

现在 master 节点上找到之前安装过的 rpm 或 zip 包，如果找不到到/usr/local 下打包。

打包安装文件
zip -r greenplum-db-$version.zip greenplum-db-$version

把打包的文件传送给新的 segment 机器上
scp -r greenplum-db-$version.zip root@IPADDRESS://usr/local

登录到新的 segment 机器上解压文件并创建软连接
unzip greenplum-db-$version.zip
ln -s greenplum-db-$version greenplum-db

6.2 检测新的 segment 机器的硬件性能

6.2.1 测试新的 segment 节点对的 I/O 与内存大写情况

$ gpcheckperf -f new_nodes -d /data1 -d /data2 -r ds

6.2.2 测试新的 segment 节点的网络情况

$ gpcheckperf -f new_nodes -r N -d /tmp

6.3 备份主要的数据库中数据

备份常用的表和 function 等主要的数据，避免升级失败无法回滚数据。

6.4 集群中常用的统计

6.4.1 数据库的大小

select pg_size_pretty(pg_database_size('databasename'));

databasename : 数据库的名字

6.4.2 查看 schema 的大小

查看每个 schema 的大小
select pg_size_pretty(cast(sum(pg_relation_size(schemaname || '.' || tablename)) as bigint)), schemaname from pg_tables t inner join pg_namespace d on t.schemaname=d.nspname group by schemaname;

查看指定 schema 的大小
select pg_size_pretty(cast(sum(pg_relation_size(schemaname || '.' || tablename)) as bigint)), schemaname from pg_tables t inner join pg_namespace d on t.schemaname=d.nspname and schemaname = 'schemaname' group by schemaname;

schemaname : 需要查询的 schema 的名字

6.4.3 查看内表与外表的数量

查看 schema 下的内表的数量
select count(*) from pg_catalog.pg_class c,pg_catalog.pg_namespace n where n.oid = c.relnamespace and n.nspname = ‘schemaname ;

schemaname : 需要查询的 schema 的名字

查询 schema 下外表的数量
select count(*) from pg_class t1, pg_namespace t2
where t1.relnamespace=t2.oid and relstorage in ('x') and relkind='r' and t1.relname = ‘schemaname ;

schemaname : 需要查询的 schema 的名字

6.4.4 查询指定 schema 的函数的数量

select c.relname from pg_catalog.pg_class c, pg_catalog.pg_namespace n where n.oid = c.relnamespace and n.nspname=’schemaname’

schemaname : 需要查询的 schema 的名字

6.4.5 查看数据库中膨胀率超过 10 的表

select * from (select t2.nspname, t1.relname, (gp_toolkit.__gp_aovisimap_compaction_info(t1.oid)).* from pg_class t1, pg_namespace t2 where t1.relnamespace=t2.oid and relstorage in ('c', 'a')) t where t.percent_hidden > 10;

6.4.6 查看需要执行 analyze

select * from gp_toolkit.gp_stats_missing where smisize='f' and smischema in ('riskbell','main');

6.5 查看集群的基本信息

查看请参考标题 4

7 升级执行过程

7.1 生成扩展文件

一下命令会在当前文件夹下生成一个带日期的文件
gpexpand -f new_nodes -D databasename

databasename： 数据库的名字

7.2 查看生成的扩展文件

cat gpexpand_inputfile_日期

主要核对一下新的机器的数据分布方式是否与原集群分布一样以及分布的节点配置信息。

7.3 执行开始扩展

gpexpand -i gpexpand_inputfile_日期 -D databasename -v -n 16 -t /tmp

databasename： 数据库的名字
-n 16： 同时支持的扩展的表的数量，可以根据机器的性能调整

7.4 查看扩容状态

查看扩容状态
psql -d databasename -c 'select * from gp_segment_configuration;'

查看数据分布前的状态
psql -d databasename -c 'select * from gpexpand.expansion_progress;'

查看扩容进度状态
psql -d databasename -c 'select * from gpexpand.status;'

查看数据分布过程会话
psql -d databasename -c 'select datid, datname, procpid, sess_id ,usesysid, usename, current_query from pg_stat_activity;'

databasename : 数据库的名字

7.5 数据重分布

调整重分布优先级，更新 gpexpand.status_detail 表将数据量为 TB 级别的表设置 rank=10

UPDATE gpexpand.status_detail SET rank=10;
UPDATE gpexpand.status_detail SET rank=1 WHERE fq_name = 'tablename';
UPDATE gpexpand.status_detail SET rank=2 WHERE fq_name = 'tablename'';
****

tablename ： 数据表的名字，格式是 schema.table

7.6 执行数据再平衡

gpexpand -a -D databasename -S -v -n 16 -t /tmp

databasename : 数据库的名字
-n 16 : 同时支持的扩展的表的数量，可以根据机器的性能调整

7.7 查看表的重分布的进度和状态

psql -d databasename -c 'select * from gpexpand.status_detail;';
psql -d databasename -c "select distinct(status) from gpexpand.status_detail where dbname=''databasename ";
psql -d databasename -c 'select * from gpexpand.expansion_progress;'

databasename : 数据库的名字

7.8 数据重分布后查询

7.8.1 查看新集群的配置及数据状态

gpstate -m

或通过以下两个命令查看
select * from gp_segment_configuration order by 1;

SELECT dbid, content, role, mode, hostname, port FROM gp_segment_configuration order by dbid;

7.8.2 确认数据分布后的状态

psql -d databasename -c 'SELECT * FROM gpexpand.expansion_progress;'
psql -d databasename -c 'SELECT * FROM gpexpand.status order by updated;'

databasename : 数据库的名字

7.9 重启集群

停数据库时，最好不好直接使用 gpstop -af 命令直接停整个数据，我们应该先把 master 给停掉,避免一些数据不一致，数据库无法启动的问题。
gpstop -amf
gpstart -am
gpstop -af

7.10 清除扩展临时的 schema

gpexpand -c

8 升级异常处理

先启动数据库
gpstart -R

再执行回滚
gpexpand -r -D databasename

扩展失败时,进行回滚
gpexpand --rollbackup
gpexpand -r

databasename : 数据库的名字