

pg集群搭建-20190607

一， 方案分析

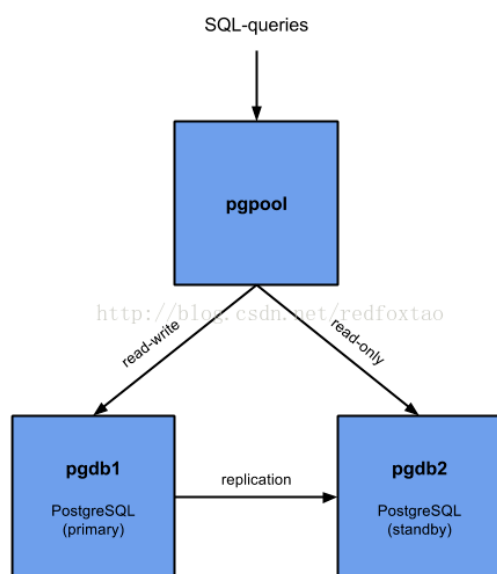
最近研究了PG的两种集群方案，分别是Pgpool-II和Postgres-XL，在这里总结一下二者的机制、结构、优劣、测试结果等。

1、 Pgpool-II和Postgres-XL简介

据我目前的了解，Pgpool-II和Postgres-XL是PG集群开源实现中比较成功的两个项目，其中Pgpool-II的前身的Pgpool-I，Postgres-XL的前身是Postgres-XC。

1.1、 Pgpool-II

Pgpool-II相当于中间件，位于应用程序和PG服务端之间，对应用程序来说，Pgpool-II就相当于PG服务端；对PG服务端来说，Pgpool-II相当于PG客户端。由此可见，Pgpool-II与PG是解耦合的，基于这样的机制，Pgpool-II可以搭建在已经存在的任意版本的PG主从结构上，主从结构的实现与Pgpool-II无关，可以通过slony等工具或者PG自身的流复制机制实现。除了主从结构的集群，Pgpool-II也支持多主结构，称为复制模式，该模式下PG节点之间是对等的，没有主从关系，写操作同时在所有节点上执行，这种模式下写操作的代价很大，性能上不及主从模式。PG 9.3之后支持的流复制机制可以方便的搭建主从结构的集群（包括同步复制与异步复制），因此Pgpool-II中比较常用的模式是流复制主从模式（也可以一主多从）。

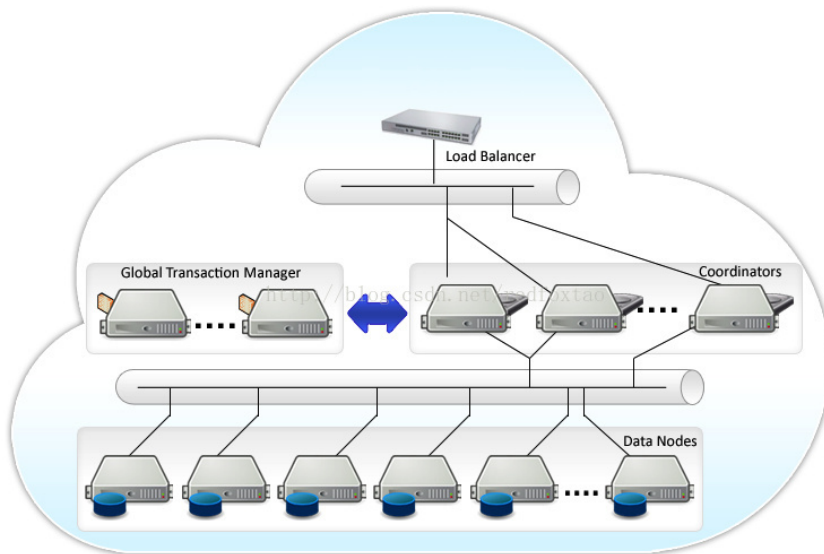


既然PG可以通过自身的流复制机制方便的搭建主从结构集群，为什么还要在它上面搭建Pgpool-II呢？因为简单的主从结构集群并不能提供连接池、负载均衡、自动故障切换等功能，Pgpool-II正好可以做到这些，当然负载均衡只针对读操作，写操作只发生在主节点

上。为了避免单点故障，Pgpool-II自身也可以配置为主从结构，对外提供虚拟IP地址，当主节点故障后，从节点提升为新的主节点并接管虚拟IP。

1.2、Postgres-XL

Postgres-XL的机制和Pgpool-II大不相同，它不是独立于PG的，是在PG源代码的基础上增加新功能实现的。简单来说，Postgres-XL将PG的SQL解析层的工作和数据存取层的工作分离到不同的两种节点上，分别称为Coordinator节点和Datanode节点，而且每种节点可以配置多个，共同协调完成原本单个PG实例完成的工作。此外，为了保证分布模式下事务能够正确执行，增加了一个GTM节点。为了避免单点故障，可以为所有节点配置对应的slave节点。Postgres-XL结构图见下图，来自官网。



Postgres-XL的Coordinator节点是整个集群的数据访问入口，可以配置多个，然后在它们之上通过Nginx等工具实现负载均衡。Coordinator节点维护着数据的存储信息，但不存储数据本身。接收到一条SQL语句后，Coordinator解析SQL，制定执行计划，然后分发任务到相关的Datanode上，Datanode返回执行结果到Coordinator，Coordinator整合各个Datanode返回的结果，最后返回给客户端。

Postgres-XL的Datanode节点负责实际存取数据，数据在多个Datanode上的分布有两种方式：复制模式和分片模式，复制模式下，一个表的数据在指定的节点上存在多个副本；分片模式下，一个表的数据按照指定的规则分布在多个数据节点上，这些节点共同保存一份完整的数据。这两种模式的选择是在创建表的时候执行CREATE TABLE语句指定的，也可以通过ALTER TABLE语句改变数据的分布方式。

2、Pgpool-II和Postgres-XL对比

	Pgpool-II	Postgres-XL
实现机制	独立于 PG 的中间件	在 PG 的特定版本上改进实现
适用场景	适合主从结构的小型集群	适合较大规模集群
扩展性	不支持动态增加节点	支持动态增减节点，但有一些限制
功能	提供连接池、虚拟 IP 配置、负载均衡、自动完成故障切换的功能	多个 Coordinator 节点时需要使用其他工具实现负载均衡；故障切换需要手动触发
数据分布	每个节点一份数据，数据冗余分布	支持数据冗余分布，也支持分片式分布。
存在的问题	写操作性能受限，没有单机 PG 的好	1、无法使用任意版本的 PG，目前最新版本 Postgres-XL 9.5 R1.3 是基于 PG 9.5 的； 2、添加 Datanode 节点后，之前创建的数据库无法分布到新的节点上。

3、Pgpool-II和Postgres-XL的性能测试

我分别使用pgbench和benchmarksql测试了Pgpool-II集群和Postgres-XL集群的性能，为了对比，还测试单机PG的性能。

测试条件：Pgpool-II集群是搭建在两台虚机上的主从复制（异步）集群；Postgres-XL集群也是搭建在相同条件上的两台虚机的集群，其中包含两个Coordinator节点和两个Datanode节点。单机PG也是运行在相同条件的虚机上。操作系统是CentOS 6.6，单机PG和Pgpool-II集群种的PG版本号是9.5，Postgres-XL的版本号是Postgres-XL 9.5 R1.3，也只基于PG 9.5的。

3.1、pgbench测试

pgbench是PG自带的一款简单的PG性能测试工具，测试指标是TPS，表示每秒钟完成的事务数。测试过程如下：

1) 建库

```
psql -h 10.192.33.244 -p7777 -c "create database pgbench"
```

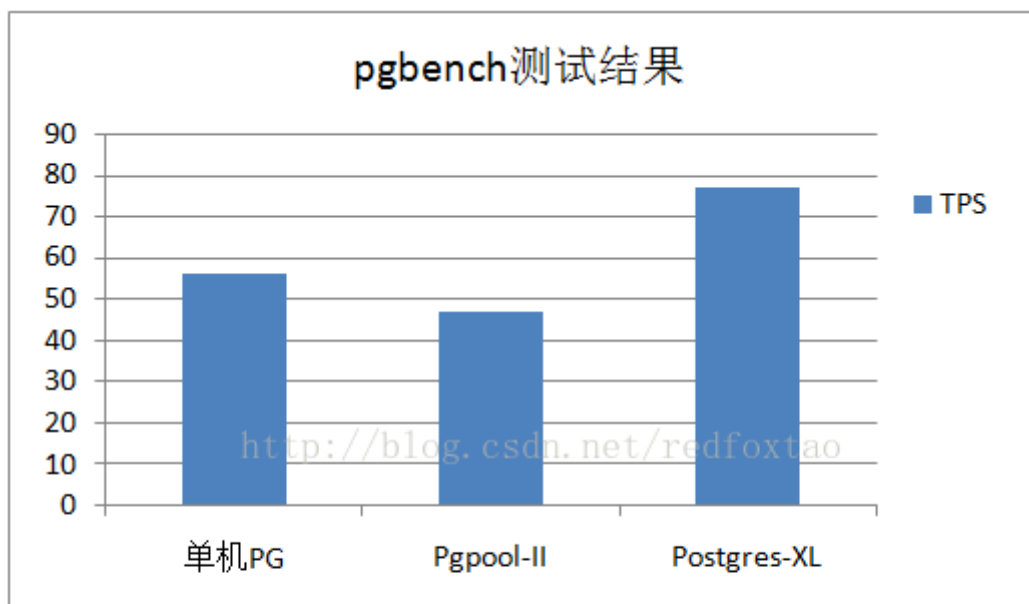
2) 生成数据

1. pgbench -i -s 1000 -h 10.192.33.244 -p 7777 pgbench
2. #参数-s指定数据量，这里使用1000，最终生成的数据量大小约16G。

3) 测试

1. pgbench -h 10.192.33.244 -p7777 -c30 -T300 -n
2. #测试时间5分钟，连续测试3次。

pgbench测试结果：



	测试 1	测试 2	测试 3	均值
单机 PG	60	43	66	56
Pgpool-II	48	43	49	47
Postgres-XL	96	75	61	77

pgbench的测试结果显示，Pgpool-II集群的性能比单机PG的性能差一些，约为84%；Postgres-XL集群的性能比单机PG的性能好一些，约为137%。

3.2、benchmarksql测试

benchmarksql的是一款常用的TPC-C测试工具，TPC-C测试衡量的是数据库的OLTP性能。测试过程如下：

1) 建库

```
psql -h 10.192.33.244 -p7777 -c "create database tpcc"
```

2) 生成数据

1. `./runDatabaseBuild.sh props.pg`

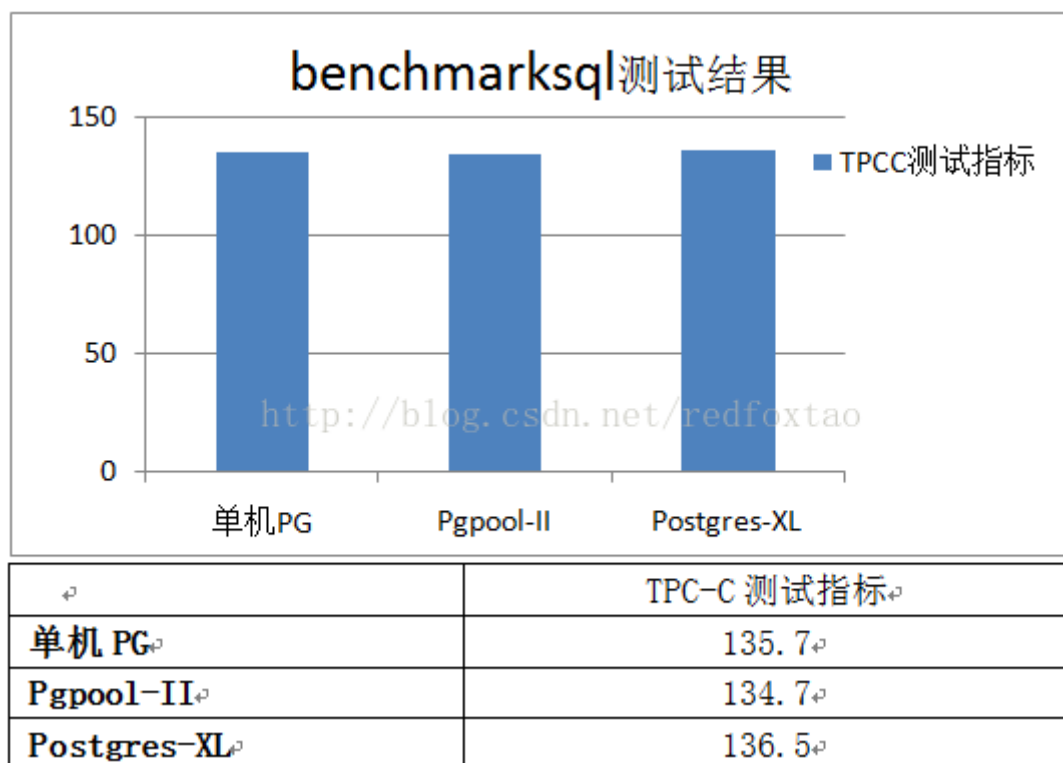
2. `#props.pg`为配置文件，配置数据库链接信息以及测试数据量、测试时间等，

3. #这里配置的数据量是100 warehouse，最终生成的数据约10G，测试时间1小时。

3) 测试

```
./runBenchmark.sh props.pg
```

benchmarksql测试结果：



benchmarksql测试结果显示，两种集群与单机PG的性能指标几乎一致，无法分辨高下。出现这种结果的可能原因之一是：测试数据量较小，无法发挥集群的性能优势，尤其像Postgres-XL这个集群在设计上针对大数据处理做了一些优化，应该更加适合大数据处理的场景。鉴于benchmarksql测试生成数据十分耗时，这里就不再进行较大数据量的测试了。最后，综合来看，我更倾向于Postgres-XL，如果公司今后打算用的话，我会推介。

二，准备环境搭建

1搭建linux环境

选取centos7做系统

root: 123456

1.1永久修改主机名：hostnamectl set-hostname postgres1/2/3

重启系统reboot

查看hostname：cat /etc/hostname 或者 hostname

1.2设置网卡：ifconfig 查看虚拟机网关名称ens160 发现没有ip地址

cat /etc/sysconfig/network-scripts/ifcfg-ens160修改网络配置

```
1 TYPE=Ethernet
2 BOOTPROTO=static
3 DEFROUTE=yes
4 PEERDNS=yes
5 PEERROUTES=yes
```

```
6 IPV4_FAILURE_FATAL=no
7 IPV6INIT=yes
8 IPV6_AUTOCONF=yes
9 IPV6_DEFROUTE=yes
10 IPV6_PEERDNS=yes
11 IPV6_PEERROUTES=yes
12 IPV6_FAILURE_FATAL=no
13 IPV6_ADDR_GEN_MODE=stable-privacy
14 NAME=ens32
15 UUID=29c76cc1-6688-404f-8537-81a3561c3bf9
16 DEVICE=ens32
17 ONBOOT=yes
18 IPADDR=192.168.2.130
19 NETMASK=255.255.255.0
20 GATEWAY=192.168.2.1
21 DNS1=210.22.70.3
22 DNS2=8.8.8.8
```

/etc/init.d/network restart

ping 内网外网地址 正常 用xshell连接查看正常

os: centos 7

pgxl:pg.version '10.3 (Postgres-XL 10alpha2)

pgxl 是一款非常实用的横向扩展的开源软件，继承了很多pgxc的功能，在replication 和 sharding 方面有着非常棒的用处。

pgxl 不严格的说是 pgxc的升级加强版。是对官方 postgresql 的版本的修改提升，为大牛点赞。

Global Transaction Monitor (GTM)

全局事务管理器，确保群集范围内的事务一致性。 GTM负责发放事务ID和快照作为其多版本并发控制的一部分。

集群可选地配置一个备用GTM，以改进可用性。此外，可以在协调器间配置代理GTM，可用于改善可扩展性，减少GTM的通信量。

GTM Standby

GTM的备节点，在pgxc,pgxl中， GTM控制所有的全局事务分配，如果出现问题，就会导致整个集群不可用，为了增加可用性，增加该备用节点。当GTM出现问题时， GTM Standby可以升级为GTM，保证集群正常工作。

GTM-Proxy

GTM需要与所有的Coordinators通信，为了降低压力，可以在每个Coordinator机器上部署一个GTM-Proxy。

Coordinator

协调员管理用户会话，并与GTM和数据节点进行交互。协调员解析，并计划查询，并给语句中的每一个组件发送下一个序列化的全局性计划。

为节省机器，通常此服务和数据节点部署在一起。

Data Node

数据节点是数据实际存储的地方。数据的分布可以由DBA来配置。为了提高可用性，可以配置数据节点的热备以便进行故障转移准备。

总结：

gtm是负责ACID的，保证分布式数据库全局事务一致性。得益于此，就算数据节点是分布的，但是你在主节点操作增删改查事务时，就如同只操作一个数据库一样简单。

Coordinator是调度的，将操作指令发送到各个数据节点。

datanodes是数据节点，分布式存储数据。

规划如下：

node1 192.168.2.164 gtm

node2 192.168.2.165 gtm-proxy,coordinator,datanode

node3 192.168.2.166 gtm-proxy,coordinator,datanode

下载

Postgres-XL 9.5 R1.6发布 - 2017年8月24日

Postgres-XL 10 R1.1 发布2019年2月18日

官方瞎咋地址<https://www.postgres-xl.org/>

<https://git.postgresql.org/gitweb/?p=postgres-xl.git;a=summary>

[git://git.postgresql.org/git/postgres-xl.git](https://git.postgresql.org/git/postgres-xl.git)

三.安装集群环境

以下操作除了 `mkdir {gtm,gtm_slave,pgxc_ctl}` 命令外都需要在node2、node3 上执行

node1 需要安装依赖包

```
1 # yum install -y bison flex perl-ExtUtils-Embed readline-devel zlib-devel  
pam-devel libxml2-devel libxslt-devel openldap-devel python-devel gcc gcc-  
c++ openssl-devel cmake openjade docbook-style-dsssl uuid uuid-devel
```

node1 节点上关闭防火墙, selinux

```
1 # systemctl stop firewalld.service
2 # systemctl disable firewalld.service
3 # vim /etc/selinux/config
4 disabled
```

node1 节点上创建用户

```
1 # groupadd postgres
2 # useradd postgres -g postgres
3 # passwd postgres
4
5 # mkdir -p /usr/pgxl-10
6 # chown -R postgres:postgres /usr/pgxl-10
7
8 # mkdir -p /var/lib/pgxl
9 # cd /var/lib/pgxl
10 # mkdir {gtm,gtm_slave,pgxc_ctl}
11 # chown -R postgres:postgres /var/lib/pgxl
```

修改host文件

```
1 [root@postgres1 ~]# vi /etc/hosts
2 添加各个节点地址
3 192.168.2.164 node1
4 192.168.2.165 node2
5 192.168.2.166 node3
```

node1 节点 postgres 用户的环境变量

```
1 # su - postgres
2 $ vi ~/.bash_profile
3 export PGUSER=postgres
4 export PGHOME=/usr/pgxl-10
5 export PGXC_CTL_HOME=/var/lib/pgxl/pgxc_ctl
6
7 export LD_LIBRARY_PATH=$PGHOME/lib
8 export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/lib:/usr/lib:/usr/local/lib
9 export PATH=$PGHOME/bin:$PATH
10
11 export TEMP=/tmp
12 export TMPDIR=/tmp
13
14 export PS1="\[\e[32;1m\][\u@\h \W]$>\[\e[0m\]"
```


node1 上编译安装

```
1 $ cd /tmp
2 $ git clone git://git.postgresql.org/git/postgres-xl.git
3 $ cd postgres-xl
4 $ git branch -r
5 origin/HEAD -> origin/master
6 origin/XL9_5_STABLE
7 origin/XL_10_STABLE
8 origin/master
9 origin/xl_dbt3_expt
10 origin/xl_doc_update
11 origin/xl_test
12 $ git checkout XL_10_STABLE
13 Branch XL_10_STABLE set up to track remote branch XL_10_STABLE from origin.
14 Switched to a new branch 'XL_10_STABLE'
15 $ git status
16 # On branch XL_10_STABLE
17 nothing to commit, working directory clean
18 $ ./configure --prefix=/usr/pgxl-10 --with-perl --with-python --with-openssl --with-pam --with-ldap --with-libxml --with-libxslt
19 $ make
20 $ make install
21 $ cd contrib
22 $ make
23 $ make install
```

2.另外在 node2、node3节点上还需要运行如下命令

```
1 # su - postgres
2 $ cd /var/lib/pgxl
3 $ mkdir {gtm_proxy}
4 $ mkdir {coord,coord_slave,coord_archlog}
5 $ mkdir {dn_master,dn_slave,dn_archlog}
```

node1、node2、node3配置ssh相互免密登录

```
1 su - pgxl
2
3 ssh-keygen -t rsa
4
5 cat ~/.ssh/id_rsa.pub>> ~/.ssh/authorized_keys
6
```

```
7  chmod 600 ~/.ssh/authorized_keys
8
9  将刚生成的认证文件拷贝到另外2台服务器:
10
11  scp ~/.ssh/authorized_keys postgres@192.168.2.165:~/.ssh/
12
13  scp ~/.ssh/authorized_keys postgres@192.168.2.166:~/.ssh/
```

node1、node2、node3同步下时间

```
1  # ntpdate asia.pool.ntp.org
```

node1,node2,node3 节点修改环境变量 (~/.bashrc和 ~/.bash_profile都要修改

```
1  /etc/hosts
2  ps -aux | grep pgxl
3  echo $PGHOME

1  [postgres@postgres2 ~]$>vi ~/.bashrc
2  # .bashrc
3  export PGUSER=postgres
4  export PGHOME=/usr/pgxl-10
5  export PGXC_CTL_HOME=/var/lib/pgxl/pgxc_ctl
6
7  export LD_LIBRARY_PATH=$PGHOME/lib
8  export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/lib:/usr/lib:/usr/local/lib
9  export PATH=$PGHOME/bin:$PATH
10
11  export TEMP=/tmp
12  export TMPDIR=/tmp
13
14  export PS1="\[\e[32;1m\][\u@\h \W]$>\[\e[0m\]"
15
16  # Source global definitions
17  if [ -f /etc/bashrc ]; then
18  . /etc/bashrc
19  fi
20
21  # Uncomment the following line if you don't like systemctl's auto-paging
  feature:
22  # export SYSTEMD_PAGER=
23
24  # User specific aliases and functions
25
```

```

26 #####
27 [postgres@postgres2 ~]$>vim ~/.bash_profile
28 # .bash_profile
29 export PGUSER=postgres
30 export PGHOME=/usr/pgxl-10
31 export PGXC_CTL_HOME=/var/lib/pgxl/pgxc_ctl
32
33 export LD_LIBRARY_PATH=$PGHOME/lib
34 export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/lib:/usr/lib:/usr/local/lib
35 export PATH=$PGHOME/bin:$PATH
36
37 export TEMP=/tmp
38 export TMPDIR=/tmp
39
40 export PS1="\[\e[32;1m\][\u@\h \W]$>\[\e[0m\]"
41
42 # Get the aliases and functions
43 if [ -f ~/.bashrc ]; then
44     . ~/.bashrc
45 fi
46
47 # User specific environment and startup programs
48
49 #PATH=$PATH:$HOME/.local/bin:$HOME/bin
50
51 #export PATH
52

```

不修改环境会出现 command not found 的错误提示

这里是依赖 ~/.bashrc

```

1 Current directory: /var/lib/pgxl/pgxc_ctl
2 Initialize GTM master
3 ERROR: target directory (/var/lib/pgxl/gtm) exists and not empty. Skip GT
M initialization
4 bash: gtm: command not found
5 bash: gtm_ctl: command not found
6 Done.
7 Start GTM master
8 bash: gtm_ctl: command not found
9 Initialize GTM slave

```

```
10 bash: initgtm: command not found
```

pgxc_ctl 生成配置文件

```
1 $ which pgxc_ctl
2 /usr/pgxl-10/bin/pgxc_ctl
3
4 $ pgxc_ctl prepare
5 /bin/bash
6 Installing pgxc_ctl_bash script as /var/lib/pgxl/pgxc_ctl/pgxc_ctl_bash.
7 ERROR: File "/var/lib/pgxl/pgxc_ctl/pgxc_ctl.conf" not found or not a regular file. No such file or directory
8 Installing pgxc_ctl_bash script as /var/lib/pgxl/pgxc_ctl/pgxc_ctl_bash.
9 Reading configuration using /var/lib/pgxl/pgxc_ctl/pgxc_ctl_bash --home /var/lib/pgxl/pgxc_ctl --configuration /var/lib/pgxl/pgxc_ctl/pgxc_ctl.conf
10 Finished reading configuration.
11 ***** PGXC_CTL START *****
12
13 Current directory: /var/lib/pgxl/pgxc_ctl
14
15 $ ls -l /var/lib/pgxl/pgxc_ctl
16 total 24
17 -rw-r--r-- 1 postgres postgres 246 Jul 18 16:42 coordExtraConfig
18 -rw-r--r-- 1 postgres postgres 17815 Jul 18 16:42 pgxc_ctl.conf
```

pgxc_ctl 修改配置文件: \$ vi /var/lib/pgxl/pgxc_ctl/pgxc_ctl.conf

```
1 #!/usr/bin/env bash
2 #
3 # Postgres-XC Configuration file for pgxc_ctl utility.
4 #
5 # Configuration file can be specified as -c option from pgxc_ctl command.
6 # Default is
7 # $PGXC_CTL_HOME/pgxc_ctl.conf.
8 #
9 # This is bash script so you can make any addition for your convenience to
10 # configure
11 # your Postgres-XC cluster.
12 #
13 # Please understand that pgxc_ctl provides only a subset of configuration
14 # which pgxc_ctl
```

```
12 # provide. Here's several several assumptions/restrictions pgxc_ctl depends on.
13 #
14 # 1) All the resources of pgxc nodes has to be owned by the same user. Same user means
15 # user with the same user name. User ID may be different from server to server.
16 # This must be specified as a variable $pgxcOwner.
17 #
18 # 2) All the servers must be reachable via ssh without password. It is highly recommended
19 # to setup key-based authentication among all the servers.
20 #
21 # 3) All the databases in coordinator/datanode has at least one same superuser. Pgxc_ctl
22 # uses this user to connect to coordinators and datanodes. Again, no password should
23 # be used to connect. You have many options to do this, pg_hba.conf, pg_ident.conf and
24 # others. Pgxc_ctl provides a way to configure pg_hba.conf but not pg_ident.conf. This
25 # will be implemented in the later releases.
26 #
27 # 4) Gtm master and slave can have different port to listen, while coordinator and datanode
28 # slave should be assigned the same port number as master.
29 #
30 # 5) Port number of a coordinator slave must be the same as its master.
31 #
32 # 6) Master and slave are connected using synchronous replication. Asynchronous replication
33 # have slight (almost none) chance to bring total cluster into inconsistent state.
34 # This chance is very low and may be negligible. Support of asynchronous replication
35 # may be supported in the later release.
36 #
37 # 7) Each coordinator and datanode can have only one slave each. Cascaded replication and
38 # multiple slave are not supported in the current pgxc_ctl.
39 #
40 # 8) Killing nodes may end up with IPC resource leak, such as semaphore and shared memory.
41 # Only listening port (socket) will be cleaned with clean command.
```

```

42 #
43 # 9) Backup and restore are not supported in pgxc_ctl at present. This is
    a big task and
44 # may need considerable resource.
45 #
46 #=====
    =====
47 #
48 #
49 # pgxcInstallDir variable is needed if you invoke "deploy" command from
    pgxc_ctl utility.
50 # If don't you don't need this variable.
51 pgxcInstallDir=/usr/pgxl-10
52 #--- OVERALL -----
    -----
53 #
54 pgxcOwner=postgres # owner of the Postgres-XC databaseo cluster. Here, w
    e use this
55 # both as linux user and database user. This must be
56 # the super user of each coordinator and datanode.
57 pgxcUser=$pgxcOwner # OS user of Postgres-XC owner
58
59 tmpDir=/tmp # temporary dir used in XC servers
60 localTmpDir=$tmpDir # temporary dir used here locally
61
62 configBackup=n # If you want config file backup, specify y to this
    value.
63 configBackupHost=pgxc-linker # host to backup config file
64 configBackupDir=$HOME/pgxc # Backup directory
65 configBackupFile=pgxc_ctl.bak # Backup file name --> Need to synchronize
    when original changed.
66
67 #--- GTM -----
    -----
68
69 # GTM is mandatory. You must have at least (and only) one GTM master in
    your Postgres-XC cluster.
70 # If GTM crashes and you need to reconfigure it, you can do it by pgxc_u
    pdate_gtm command to update
71 # GTM master with others. Of course, we provide pgxc_remove_gtm command
    to remove it. This command
72 # will not stop the current GTM. It is up to the operator.
73

```

```

74
75 #---- GTM Master -----
76
77 #---- Overall ----
78 gtmName=node1_gtm
79 gtmMasterServer=node1
80 gtmMasterPort=6666
81 gtmMasterDir=/var/lib/pgxl/gtm
82
83 #---- Configuration ---
84 gtmExtraConfig=none # Will be added gtm.conf for both Master and Slave
      (done at initialization only)
85 gtmMasterSpecificExtraConfig=none # Will be added to Master's gtm.conf
      (done at initialization only)
86
87 #---- GTM Slave -----
88
89 # Because GTM is a key component to maintain database consistency, you may
      want to configure GTM slave
90 # for backup.
91
92 #---- Overall -----
93 gtmSlave=y # Specify y if you configure GTM Slave. Otherwise, GTM slave
      will not be configured and
94 # all the following variables will be reset.
95 gtmSlaveName=node1_gtm_slave
96 gtmSlaveServer=node1 # value none means GTM slave is not available. Give
      none if you don't configure GTM Slave.
97 gtmSlavePort=6667 # Not used if you don't configure GTM slave.
98 gtmSlaveDir=/var/lib/pgxl/gtm_slave # Not used if you don't configure GTM
      slave.
99 # Please note that when you have GTM failover, then there will be no slave
      available until you configure the slave
100 # again. (pgxc_add_gtm_slave function will handle it)
101
102 #---- Configuration ----
103 gtmSlaveSpecificExtraConfig=none # Will be added to Slave's gtm.conf (done
      at initialization only)
104
105 #---- GTM Proxy -----
      -----
106 # GTM proxy will be selected based upon which server each component runs
      on.

```

```

107 # When fails over to the slave, the slave inherits its master's gtm proxy. It should be
108 # reconfigured based upon the new location.
109 #
110 # To do so, slave should be restarted. So pg_ctl promote -> (edit postgresql.conf and recovery.conf) -> pg_ctl restart
111 #
112 # You don't have to configure GTM Proxy if you don't configure GTM slave or you are happy if every component connects
113 # to GTM Master directly. If you configure GTL slave, you must configure GTM proxy too.
114
115 #---- Shortcuts ----
116 gtmProxyDir=/var/lib/pgxl/gtm_proxy
117
118 #---- Overall -----
119 gtmProxy=y # Specify y if you configure at least one GTM proxy. You may not configure gtm proxies
120 # only when you don't configure GTM slaves.
121 # If you specify this value not to y, the following parameters will be set to default empty values.
122 # If we find there're no valid Proxy server names (means, every servers are specified
123 # as none), then gtmProxy value will be set to "n" and all the entries will be set to
124 # empty values.
125 gtmProxyNames=(gtm_pxy1 gtm_pxy2) # No used if it is not configured
126 gtmProxyServers=(node2 node3) # Specify none if you don't configure it.
127 gtmProxyPorts=(6668 6668) # Not used if it is not configured.
128 gtmProxyDirs=($gtmProxyDir $gtmProxyDir) # Not used if it is not configured.
129
130 #---- Configuration ----
131 gtmPxyExtraConfig=none # Extra configuration parameter for gtm_proxy. Coordinator section has an example.
132 gtmPxySpecificExtraConfig=(none none)
133
134 #---- Coordinators -----
135
136 #---- shortcuts -----
137 coordMasterDir=/var/lib/pgxl/coord
138 coordSlaveDir=/var/lib/pgxl/coord_slave

```



```

139 coordArchLogDir=/var/lib/pgxl/coord_slave
140
141 #---- Overall -----
142 coordNames=(coord1 coord2) # Master and slave use the same name
143 coordPorts=(20004 20005) # Master ports
144 poolerPorts=(20010 20011) # Master pooler ports
145 coordPgHbaEntries=(192.168.2.0/24) # Assumes that all the coordinator (master/slave) accepts
146 # the same connection
147 # This entry allows only $pgxcOwner to connect.
148 # If you'd like to setup another connection, you should
149 # supply these entries through files specified below.
150 # Note: The above parameter is extracted as "host all all 0.0.0.0/0 trust". If you don't want
151 # such setups, specify the value () to this variable and supply what you want using coordExtraPgHba
152 # and/or coordSpecificExtraPgHba variables.
153 #coordPgHbaEntries=(::1/128) # Same as above but for IPv6 addresses
154
155 #---- Master -----
156 coordMasterServers=(node2 node3) # none means this master is not available
157 coordMasterDirs=($coordMasterDir $coordMasterDir)
158 coordMaxWALsender=10 # max_wal_senders: needed to configure slave. If zero value is specified,
159 # it is expected to supply this parameter explicitly by external files
160 # specified in the following. If you don't configure slaves, leave this value to zero.
161 coordMaxWALSenders=($coordMaxWALsender $coordMaxWALsender)
162 # max_wal_senders configuration for each coordinator.
163
164 #---- Slave -----
165 coordSlave=y # Specify y if you configure at least one coordinator slave. Otherwise, the following
166 # configuration parameters will be set to empty values.
167 # If no effective server names are found (that is, every servers are specified as none),
168 # then coordSlave value will be set to n and all the following values will be set to
169 # empty values.
170
171 coordUserDefinedBackupSettings=n # Specify whether to update backup/recovery

```

```

172 # settings during standby addition/removal.
173
174 coordSlaveSync=y # Specify to connect with synchronized mode.
175 coordSlaveServers=(node3 node2) # none means this slave is not available
176 coordSlavePorts=(20004 20005) # Master ports
177 coordSlavePoolerPorts=(20010 20011) # Master pooler ports
178 coordSlaveDirs=($coordSlaveDir $coordSlaveDir)
179 coordArchLogDirs=($coordArchLogDir $coordArchLogDir)
180
181 #---- Configuration files---
182 # Need these when you'd like setup specific non-default configuration
183 # These files will go to corresponding files for the master.
184 # You may supply your bash script to setup extra config lines and extra
  pg_hba.conf entries
185 # Or you may supply these files manually.
186 coordExtraConfig=coordExtraConfig # Extra configuration file for coordin
  ators.
187 # This file will be added to all the coordinators'
188 # postgresql.conf
189 # Please note that the following sets up minimum parameters which you may
  want to change.
190 # You can put your postgresql.conf lines here.
191 cat > $coordExtraConfig <<EOF
192 #=====
193 # Added to all the coordinator postgresql.conf
194 # Original: $coordExtraConfig
195 log_destination = 'stderr'
196 logging_collector = on
197 log_directory = 'pg_log'
198 listen_addresses = '*'
199 max_connections = 100
200 EOF
201
202 # Additional Configuration file for specific coordinator master.
203 # You can define each setting by similar means as above.
204 coordSpecificExtraConfig=(none none)
205 coordExtraPgHba=none # Extra entry for pg_hba.conf. This file will be ac
  ded to all the coordinators' pg_hba.conf
206 coordSpecificExtraPgHba=(none none)
207
208 #---- Additional Slaves ----

```

```

209 #
210 # Please note that this section is just a suggestion how we extend the c
onfiguration for
211 # multiple and cascaded replication. They're not used in the current ver
sion.
212 #
213 coordAdditionalSlaves=n # Additional slave can be specified as follows:
where you
214 coordAdditionalSlaveSet=(cad1) # Each specifies set of slaves. This
case, two set of slaves are
215 # configured
216 cad1_Sync=n # All the slaves at "cad1" are connected with asynchronous m
ode.
217 # If not, specify "y"
218 # The following lines specifies detailed configuration for each
219 # slave tag, cad1. You can define cad2 similarly.
220 cad1_Servers=(node08 node09 node06 node07) # Hosts
221 cad1_dir=$HOME/pgxc/nodes/coord_slave_cad1
222 cad1_Dirs=($cad1_dir $cad1_dir $cad1_dir $cad1_dir)
223 cad1_ArchLogDir=$HOME/pgxc/nodes/coord_archlog_cad1
224 cad1_ArchLogDirs=($cad1_ArchLogDir $cad1_ArchLogDir $cad1_ArchLogDir $ca
d1_ArchLogDir)
225
226
227 #---- Datanodes -----
-----
228
229 #---- Shortcuts -----
230 datanodeMasterDir=/var/lib/pgxl/dn_master
231 datanodeSlaveDir=/var/lib/pgxl/dn_slave
232 datanodeArchLogDir=/var/lib/pgxl/dn_archlog
233
234 #---- Overall -----
235 #primaryDatanode=datanode1 # Primary Node.
236 # At present, xc has a priblem to issue ALTER NODE against the primay n
de. Until it is fixed, the test will be done
237 # without this feature.
238 primaryDatanode=node2 # Primary Node.
239 datanodeNames=(datanode1 datanode2)
240 datanodePorts=(20008 20009 ) # Master ports
241 datanodePoolerPorts=(20012 20013) # Master pooler ports
242 datanodePgHbaEntries=(192.168.2.0/24) # Assumes that all the coordinator
(master/slave) accepts

```

```

243 # the same connection
244 # This list sets up pg_hba.conf for $pgxcOwner user.
245 # If you'd like to setup other entries, supply them
246 # through extra configuration files specified below.
247 # Note: The above parameter is extracted as "host all all 0.0.0.0/0 trust". If you don't want
248 # such setups, specify the value () to this variable and supply what you
  want using datanodeExtraPgHba
249 # and/or datanodeSpecificExtraPgHba variables.
250 #datanodePgHbaEntries=(::1/128) # Same as above but for IPv6 addresses
251
252 #---- Master -----
253 datanodeMasterServers=(node2 node3) # none means this master is not available.
254 # This means that there should be the master but is down.
255 # The cluster is not operational until the master is
256 # recovered and ready to run.
257 datanodeMasterDirs=($datanodeMasterDir $datanodeMasterDir)
258 datanodeMaxWalSender=10 # max_wal_senders: needed to configure slave. If
  zero value is
259 # specified, it is expected this parameter is explicitly supplied
260 # by external configuration files.
261 # If you don't configure slaves, leave this value zero.
262 datanodeMaxWALSenders=($datanodeMaxWalSender $datanodeMaxWalSender)
263 # max_wal_senders configuration for each datanode
264
265 #---- Slave -----
266 datanodeSlave=y # Specify y if you configure at least one coordinator slave. Otherwise, the following
267 # configuration parameters will be set to empty values.
268 # If no effective server names are found (that is, every servers are specified as none),
269 # then datanodeSlave value will be set to n and all the following values will be set to
270 # empty values.
271
272 datanodeUserDefinedBackupSettings=n # Specify whether to update backup/recovery
273 # settings during standby addition/removal.
274
275 datanodeSlaveServers=(node3 node2) # value none means this slave is not available

```

```

276 datanodeSlavePorts=(20008 20009) # value none means this slave is not available
277 datanodeSlavePoolerPorts=(20012 20013) # value none means this slave is not available
278 datanodeSlaveSync=y # If datanode slave is connected in synchronized mode
279 datanodeSlaveDirs=($datanodeSlaveDir $datanodeSlaveDir)
280 datanodeArchLogDirs=($datanodeArchLogDir $datanodeArchLogDir)
281
282 # ----- Configuration files -----
283 # You may supply your bash script to setup extra config lines and extra pg_hba.conf entries here.
284 # These files will go to corresponding files for the master.
285 # Or you may supply these files manually.
286 datanodeExtraConfig=none # Extra configuration file for datanodes. This file will be added to all the
287 # datanodes' postgresql.conf
288 datanodeSpecificExtraConfig=(none none)
289 datanodeExtraPgHba=none # Extra entry for pg_hba.conf. This file will be added to all the datanodes' postgresql.conf
290 datanodeSpecificExtraPgHba=(none none)
291
292 #----- Additional Slaves -----
293 datanodeAdditionalSlaves=n # Additional slave can be specified as follows: where you
294 # datanodeAdditionalSlaveSet=(dad1 dad2) # Each specifies set of slaves. This case, two set of slaves are
295 # configured
296 # dad1_Sync=n # All the slaves at "cad1" are connected with asynchronous mode.
297 # If not, specify "y"
298 # The following lines specifies detailed configuration for each
299 # slave tag, cad1. You can define cad2 similarly.
300 # dad1_Servers=(node08 node09 node06 node07) # Hosts
301 # dad1_dir=$HOME/pgxc/nodes/coord_slave_cad1
302 # dad1_Dirs=($cad1_dir $cad1_dir $cad1_dir $cad1_dir)
303 # dad1_ArchLogDir=$HOME/pgxc/nodes/coord_archlog_cad1
304 # dad1_ArchLogDirs=($cad1_ArchLogDir $cad1_ArchLogDir $cad1_ArchLogDir $cad1_ArchLogDir)
305
306 #----- WAL archives -----

```

```

307 walArchive=n # If you'd like to configure WAL archive, edit this
section.
308 # Pgxc_ctl assumes that if you configure WAL archive, you configure it
309 # for all the coordinators and datanodes.
310 # Default is "no". Please specify "y" here to turn it on.
311 #
312 # End of Configuration Section
313 #
314 #=====
=====
315
316 #=====
=====
317 # The following is for extension. Just demonstrate how to write such ext
ension. There's no code
318 # which takes care of them so please ignore the following lines. They ar
e simply ignored by pgxc_ctl.
319 # No side effects.
320 #=====<< Beginning of future extension demonstration >> =====
=====
321 # You can setup more than one backup set for various purposes, such as c
isaster recovery.
322 walArchiveSet=(war1 war2)
323 war1_source=(master) # you can specify master, slave or ano other additi
onal slaves as a source of WAL archive.
324 # Default is the master
325 wal1_source=(slave)
326 wal1_source=(additional_coordinator_slave_set additional_datanode_slave_s
et)
327 war1_host=node10 # All the nodes are backed up at the same host for a gi
ven archive set
328 war1_backupdir=$HOME/pgxc/backup_war1
329 wal2_source=(master)
330 war2_host=node11
331 war2_backupdir=$HOME/pgxc/backup_war2
332 #=====<< End of future extension demonstration >> =====
=====
333

```

pgxc_ctl 的一些操作
在 node1 节点上操作

```

1 #初始化集群（安装）
2 $ pgxc_ctl -c /var/lib/pgxl/pgxc_ctl/pgxc_ctl.conf init all
3 #启动集群
4 $ pgxc_ctl -c /var/lib/pgxl/pgxc_ctl/pgxc_ctl.conf start all
5 关闭集群
6 $ pgxc_ctl -c /var/lib/pgxl/pgxc_ctl/pgxc_ctl.conf stop all

```

重新安装

```

1
2 停止进程ps -aux | grep pgxl
3 kill -9 xxxx
4
5 ERROR: target directory (/var/lib/pgxl/gtm) exists and not empty. Skip GT
M initialization
6 清理/var/lib/pgxl/gtm路径
7 node1节点
8 # cd /var/lib/pgxl
9 # mkdir {gtm,gtm_slave,pgxc_ctl}
10 node2 node3节点
11 su - postgres
12 cd /var/lib/pgxl/gtm_proxy
13 $ mkdir gtm_proxy
14 $ mkdir {coord,coord_slave,coord_archlog}
15 $ mkdir {dn_master,dn_slave,dn_archlog}
16 清理路径
17
18 问题处理
19 gtm_ctl: could not send stop signal (PID: 26076): No such process
20

```

EXECUTE DIRECT ON (node2) 'CREATE NODE coord1 WITH (TYPE="coordinator",
HOST="node2", PORT=20004);

ERROR: PGXC Node node2: object not defined

EXECUTE DIRECT ON (node2) 'CREATE NODE coord2 WITH (TYPE="coordinator",
HOST="node3", PORT=20005);

ERROR: PGXC Node node2: object not defined

```
EXECUTE DIRECT ON (node2) 'ALTER NODE node2 WITH (TYPE="datanode",
HOST="node2", PORT=20008, PRIMARY, PREFERRED)';
ERROR: PGXC Node node2: object not defined
EXECUTE DIRECT ON (node2) 'CREATE NODE node3 WITH (TYPE="datanode",
HOST="node3", PORT=20009, PREFERRED)';
ERROR: PGXC Node node2: object not defined
EXECUTE DIRECT ON (node2) 'SELECT pgxc_pool_reload()';
ERROR: PGXC Node node2: object not defined
EXECUTE DIRECT ON (node3) 'CREATE NODE coord1 WITH (TYPE="coordinator",
HOST="node2", PORT=20004)';
ERROR: PGXC Node node3: object not defined
EXECUTE DIRECT ON (node3) 'CREATE NODE coord2 WITH (TYPE="coordinator",
HOST="node3", PORT=20005)';
ERROR: PGXC Node node3: object not defined
EXECUTE DIRECT ON (node3) 'CREATE NODE node2 WITH (TYPE="datanode",
HOST="node2", PORT=20008, PRIMARY, PREFERRED)';
ERROR: PGXC Node node3: object not defined
EXECUTE DIRECT ON (node3) 'ALTER NODE node3 WITH (TYPE="datanode",
HOST="node3", PORT=20009, PREFERRED)';
ERROR: PGXC Node node3: object not defined
EXECUTE DIRECT ON (node3) 'SELECT pgxc_pool_reload()';
ERROR: PGXC Node node3: object not defined
```

此问题纠结很久 处理方法

修改host文件:vi /etc/hosts

```
1 127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdoma
in4
2 ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
3 192.168.2.164 node1
4 192.168.2.165 node2
5 192.168.2.166 node3
```

修改脚本238 239行为

primaryDatanode=datanode1 # Primary Node.

datanodeNames=(datanode1 datanode2)

问题解决

下面是 的输出日志，记录一下


```
1 [postgres@postgres1 ~]$ pgxc_ctl -c /var/lib/pgxl/pgxc_ctl/pgxc_ctl.conf
init all
2 /bin/bash
3 Installing pgxc_ctl_bash script as /var/lib/pgxl/pgxc_ctl/pgxc_ctl_bash.
4 Installing pgxc_ctl_bash script as /var/lib/pgxl/pgxc_ctl/pgxc_ctl_bash.
5 Reading configuration using /var/lib/pgxl/pgxc_ctl/pgxc_ctl_bash --home
/var/lib/pgxl/pgxc_ctl --configuration /var/lib/pgxl/pgxc_ctl/pgxc_ctl.conf
6 Finished reading configuration.
7 ***** PGXC_CTL START *****
8
9 Current directory: /var/lib/pgxl/pgxc_ctl
10 Initialize GTM master
11 The files belonging to this GTM system will be owned by user "postgres".
12 This user must also own the server process.
13
14
15 fixing permissions on existing directory /var/lib/pgxl/gtm ... ok
16 creating configuration files ... ok
17 creating control file ... ok
18
19 Success.
20 Done.
21 Start GTM master
22 server starting
23 Initialize GTM slave
24 The files belonging to this GTM system will be owned by user "postgres".
25 This user must also own the server process.
26
27
28 fixing permissions on existing directory /var/lib/pgxl/gtm_slave ... ok
29 creating configuration files ... ok
30 creating control file ... ok
31
32 Success.
33 Done.
34 Start GTM slaveserver starting
35 Done.
36 Initialize all the gtm proxies.
37 Initializing gtm proxy gtm_proxy1.
38 Initializing gtm proxy gtm_proxy2.
39 gtm_ctl: could not send stop signal (PID: 20104): No such process
```

```
40 The files belonging to this GTM system will be owned by user "postgres".
41 This user must also own the server process.
42
43
44 fixing permissions on existing directory /var/lib/pgxl/gtm_proxy ... ok
45 creating configuration files ... ok
46
47 Success.
48 gtm_ctl: could not send stop signal (PID: 12004): No such process
49 The files belonging to this GTM system will be owned by user "postgres".
50 This user must also own the server process.
51
52
53 fixing permissions on existing directory /var/lib/pgxl/gtm_proxy ... ok
54 creating configuration files ... ok
55
56 Success.
57 Done.
58 Starting all the gtm proxies.
59 Starting gtm proxy gtm_proxy1.
60 Starting gtm proxy gtm_proxy2.
61 server starting
62 server starting
63 Done.
64 Initialize all the coordinator masters.
65 Initialize coordinator master coord1.
66 Initialize coordinator master coord2.
67 The files belonging to this database system will be owned by user "postgres".
68 This user must also own the server process.
69
70 The database cluster will be initialized with locale "zh_CN.UTF-8".
71 The default database encoding has accordingly been set to "UTF8".
72 initdb: could not find suitable text search configuration for locale "zh_CN.UTF-8"
73 The default text search configuration will be set to "simple".
74
75 Data page checksums are disabled.
76
77 fixing permissions on existing directory /var/lib/pgxl/coord ... ok
78 creating subdirectories ... ok
```

```
79 selecting default max_connections ... 100
80 selecting default shared_buffers ... 128MB
81 selecting dynamic shared memory implementation ... posix
82 creating configuration files ... ok
83 running bootstrap script ... ok
84 performing post-bootstrap initialization ... creating cluster informatio
n ... ok
85 syncing data to disk ... ok
86 freezing database template0 ... ok
87 freezing database template1 ... ok
88 freezing database postgres ... ok
89
90 WARNING: enabling "trust" authentication for local connections
91 You can change this by editing pg_hba.conf or using the option -A, or
92 --auth-local and --auth-host, the next time you run initdb.
93
94 Success.
95 The files belonging to this database system will be owned by user "postg
res".
96 This user must also own the server process.
97
98 The database cluster will be initialized with locale "zh_CN.UTF-8".
99 The default database encoding has accordingly been set to "UTF8".
100 initdb: could not find suitable text search configuration for locale "zh
_CN.UTF-8"
101 The default text search configuration will be set to "simple".
102
103 Data page checksums are disabled.
104
105 fixing permissions on existing directory /var/lib/pgxl/coord ... ok
106 creating subdirectories ... ok
107 selecting default max_connections ... 100
108 selecting default shared_buffers ... 128MB
109 selecting dynamic shared memory implementation ... posix
110 creating configuration files ... ok
111 running bootstrap script ... ok
112 performing post-bootstrap initialization ... creating cluster informatio
n ... ok
113 syncing data to disk ... ok
114 freezing database template0 ... ok
115 freezing database template1 ... ok
```

```
116 freezing database postgres ... ok
117
118 WARNING: enabling "trust" authentication for local connections
119 You can change this by editing pg_hba.conf or using the option -A, or
120 --auth-local and --auth-host, the next time you run initdb.
121
122 Success.
123 Done.
124 Starting coordinator master.
125 Starting coordinator master coord1
126 Starting coordinator master coord2
127 2019-06-11 14:08:22.834 CST [21489] LOG: listening on IPv4 address "0.0.0.0", port 20004
128 2019-06-11 14:08:22.834 CST [21489] LOG: listening on IPv6 address ":::", port 20004
129 2019-06-11 14:08:22.837 CST [21489] LOG: listening on Unix socket "/tmp/.s.PGSQL.20004"
130 2019-06-11 14:08:22.846 CST [21489] LOG: redirecting log output to logging collector process
131 2019-06-11 14:08:22.846 CST [21489] HINT: Future log output will appear in directory "pg_log".
132 2019-06-11 14:08:22.801 CST [13773] LOG: listening on IPv4 address "0.0.0.0", port 20005
133 2019-06-11 14:08:22.801 CST [13773] LOG: listening on IPv6 address ":::", port 20005
134 2019-06-11 14:08:22.804 CST [13773] LOG: listening on Unix socket "/tmp/.s.PGSQL.20005"
135 2019-06-11 14:08:22.814 CST [13773] LOG: redirecting log output to logging collector process
136 2019-06-11 14:08:22.814 CST [13773] HINT: Future log output will appear in directory "pg_log".
137 Done.
138 Initialize all the coordinator slaves.
139 Initialize the coordinator slave coord1.
140 Initialize the coordinator slave coord2.
141 Done.
142 Starting all the coordinator slaves.
143 Starting coordinator slave coord1.
144 Starting coordinator slave coord2.
145 2019-06-11 14:08:28.443 CST [14091] LOG: listening on IPv4 address "0.0.0.0", port 20004
146 2019-06-11 14:08:28.443 CST [14091] LOG: listening on IPv6 address ":::", port 20004
```

```
147 2019-06-11 14:08:28.446 CST [14091] LOG: listening on Unix socket "/tmp/.s.PGSQL.20004"
148 2019-06-11 14:08:28.457 CST [14091] LOG: redirecting log output to logging collector process
149 2019-06-11 14:08:28.457 CST [14091] HINT: Future log output will appear in directory "pg_log".
150 2019-06-11 14:08:28.464 CST [21806] LOG: listening on IPv4 address "0.0.0.0", port 20005
151 2019-06-11 14:08:28.464 CST [21806] LOG: listening on IPv6 address ":::", port 20005
152 2019-06-11 14:08:28.472 CST [21806] LOG: listening on Unix socket "/tmp/.s.PGSQL.20005"
153 2019-06-11 14:08:28.484 CST [21806] LOG: redirecting log output to logging collector process
154 2019-06-11 14:08:28.484 CST [21806] HINT: Future log output will appear in directory "pg_log".
155 Done
156 Initialize all the datanode masters.
157 Initialize the datanode master datanode1.
158 Initialize the datanode master datanode2.
159 The files belonging to this database system will be owned by user "postgres".
160 This user must also own the server process.
161
162 The database cluster will be initialized with locale "zh_CN.UTF-8".
163 The default database encoding has accordingly been set to "UTF8".
164 initdb: could not find suitable text search configuration for locale "zh_CN.UTF-8"
165 The default text search configuration will be set to "simple".
166
167 Data page checksums are disabled.
168
169 fixing permissions on existing directory /var/lib/pgxl/dn_master ... ok
170 creating subdirectories ... ok
171 selecting default max_connections ... 100
172 selecting default shared_buffers ... 128MB
173 selecting dynamic shared memory implementation ... posix
174 creating configuration files ... ok
175 running bootstrap script ... ok
176 performing post-bootstrap initialization ... creating cluster information ... ok
177 syncing data to disk ... ok
178 freezing database template0 ... ok
```

```
179 freezing database template1 ... ok
180 freezing database postgres ... ok
181
182 WARNING: enabling "trust" authentication for local connections
183 You can change this by editing pg_hba.conf or using the option -A, or
184 --auth-local and --auth-host, the next time you run initdb.
185
186 Success.
187 The files belonging to this database system will be owned by user "postgres".
188 This user must also own the server process.
189
190 The database cluster will be initialized with locale "zh_CN.UTF-8".
191 The default database encoding has accordingly been set to "UTF8".
192 initdb: could not find suitable text search configuration for locale "zh_CN.UTF-8"
193 The default text search configuration will be set to "simple".
194
195 Data page checksums are disabled.
196
197 fixing permissions on existing directory /var/lib/pgxl/dn_master ... ok
198 creating subdirectories ... ok
199 selecting default max_connections ... 100
200 selecting default shared_buffers ... 128MB
201 selecting dynamic shared memory implementation ... posix
202 creating configuration files ... ok
203 running bootstrap script ... ok
204 performing post-bootstrap initialization ... creating cluster information ... ok
205 syncing data to disk ... ok
206 freezing database template0 ... ok
207 freezing database template1 ... ok
208 freezing database postgres ... ok
209
210 WARNING: enabling "trust" authentication for local connections
211 You can change this by editing pg_hba.conf or using the option -A, or
212 --auth-local and --auth-host, the next time you run initdb.
213
214 Success.
215 Done.
216 Starting all the datanode masters.
```

```
217 Starting datanode master datanode1.
218 Starting datanode master datanode2.
219 2019-06-11 14:08:35.227 CST [22292] LOG: listening on IPv4 address "0.0.0.0", port 20008
220 2019-06-11 14:08:35.227 CST [22292] LOG: listening on IPv6 address ":::", port 20008
221 2019-06-11 14:08:35.327 CST [22292] LOG: listening on Unix socket "/tmp/.s.PGSQL.20008"
222 2019-06-11 14:08:35.405 CST [22292] LOG: redirecting log output to logging collector process
223 2019-06-11 14:08:35.405 CST [22292] HINT: Future log output will appear in directory "pg_log".
224 2019-06-11 14:08:35.196 CST [14575] LOG: listening on IPv4 address "0.0.0.0", port 20009
225 2019-06-11 14:08:35.197 CST [14575] LOG: listening on IPv6 address ":::", port 20009
226 2019-06-11 14:08:35.296 CST [14575] LOG: listening on Unix socket "/tmp/.s.PGSQL.20009"
227 2019-06-11 14:08:35.376 CST [14575] LOG: redirecting log output to logging collector process
228 2019-06-11 14:08:35.376 CST [14575] HINT: Future log output will appear in directory "pg_log".
229 Done.
230 Initialize all the datanode slaves.
231 Initialize datanode slave datanode1
232 Initialize datanode slave datanode2
233 Starting all the datanode slaves.
234 Starting datanode slave datanode1.
235 Starting datanode slave datanode2.
236 2019-06-11 14:08:42.349 CST [14892] LOG: listening on IPv4 address "0.0.0.0", port 20008
237 2019-06-11 14:08:42.349 CST [14892] LOG: listening on IPv6 address ":::", port 20008
238 2019-06-11 14:08:42.352 CST [14892] LOG: listening on Unix socket "/tmp/.s.PGSQL.20008"
239 2019-06-11 14:08:42.361 CST [14892] LOG: redirecting log output to logging collector process
240 2019-06-11 14:08:42.361 CST [14892] HINT: Future log output will appear in directory "pg_log".
241 2019-06-11 14:08:42.377 CST [22609] LOG: listening on IPv4 address "0.0.0.0", port 20009
242 2019-06-11 14:08:42.377 CST [22609] LOG: listening on IPv6 address ":::", port 20009
243 2019-06-11 14:08:42.379 CST [22609] LOG: listening on Unix socket "/tmp/.s.PGSQL.20009"
```

```

244 2019-06-11 14:08:42.389 CST [22609] LOG: redirecting log output to logg
ng collector process
245 2019-06-11 14:08:42.389 CST [22609] HINT: Future log output will appear
in directory "pg_log".
246 Done.
247 ALTER NODE coord1 WITH (HOST='node2', PORT=20004);
248 ALTER NODE
249 CREATE NODE coord2 WITH (TYPE='coordinator', HOST='node3', PORT=20005);
250 CREATE NODE
251 CREATE NODE datanode1 WITH (TYPE='datanode', HOST='node2', PORT=20008, P
RIMARY, PREFERRED);
252 CREATE NODE
253 CREATE NODE datanode2 WITH (TYPE='datanode', HOST='node3', PORT=20009);
254 CREATE NODE
255 SELECT pgxc_pool_reload();
256 pgxc_pool_reload
257 -----
258 t
259 (1 row)
260
261 CREATE NODE coord1 WITH (TYPE='coordinator', HOST='node2', PORT=20004);
262 CREATE NODE
263 ALTER NODE coord2 WITH (HOST='node3', PORT=20005);
264 ALTER NODE
265 CREATE NODE datanode1 WITH (TYPE='datanode', HOST='node2', PORT=20008, P
RIMARY);
266 CREATE NODE
267 CREATE NODE datanode2 WITH (TYPE='datanode', HOST='node3', PORT=20009, P
REFERRED);
268 CREATE NODE
269 SELECT pgxc_pool_reload();
270 pgxc_pool_reload
271 -----
272 t
273 (1 row)
274
275 Done.
276 EXECUTE DIRECT ON (datanode1) 'CREATE NODE coord1 WITH (TYPE=''coordinat
or'', HOST=''node2'', PORT=20004)';
277 EXECUTE DIRECT
278 EXECUTE DIRECT ON (datanode1) 'CREATE NODE coord2 WITH (TYPE=''coordinat
or'', HOST=''node3'', PORT=20005)';

```



```

279 EXECUTE DIRECT
280 EXECUTE DIRECT ON (datanode1) 'ALTER NODE datanode1 WITH (TYPE=''datanode'', HOST=''node2'', PORT=20008, PRIMARY, PREFERRED)';
281 EXECUTE DIRECT
282 EXECUTE DIRECT ON (datanode1) 'CREATE NODE datanode2 WITH (TYPE=''datanode'', HOST=''node3'', PORT=20009, PREFERRED)';
283 EXECUTE DIRECT
284 EXECUTE DIRECT ON (datanode1) 'SELECT pgxc_pool_reload()';
285 pgxc_pool_reload
286 -----
287 t
288 (1 row)
289
290 EXECUTE DIRECT ON (datanode2) 'CREATE NODE coord1 WITH (TYPE=''coordinator'', HOST=''node2'', PORT=20004)';
291 EXECUTE DIRECT
292 EXECUTE DIRECT ON (datanode2) 'CREATE NODE coord2 WITH (TYPE=''coordinator'', HOST=''node3'', PORT=20005)';
293 EXECUTE DIRECT
294 EXECUTE DIRECT ON (datanode2) 'CREATE NODE datanode1 WITH (TYPE=''datanode'', HOST=''node2'', PORT=20008, PRIMARY, PREFERRED)';
295 EXECUTE DIRECT
296 EXECUTE DIRECT ON (datanode2) 'ALTER NODE datanode2 WITH (TYPE=''datanode'', HOST=''node3'', PORT=20009, PREFERRED)';
297 EXECUTE DIRECT
298 EXECUTE DIRECT ON (datanode2) 'SELECT pgxc_pool_reload()';
299 pgxc_pool_reload
300 -----
301 t
302 (1 row)
303
304 Done.
305

```

四.验证

登录 node2节点的 coordinator, 发现都不用再手动 create node

```

1 $ psql -p 20004
2 psql (PGXL 10alpha2, based on PG 10.3 (Postgres-XL 10alpha2))
3 Type "help" for help.
4

```

```

5 postgres=#
6 postgres=# select * from pgxc_node;
7  node_name | node_type | node_port | node_host | nodeis_primary | nodeis_preferred | node_id
8  -----+-----+-----+-----+-----+-----+-----
9  coord1   | C        | 20004    | node2    | f              | f                | 1885696643
10 coord2   | C        | 20005    | node3    | f              | f                | -1197102633
11 datanode1 | D        | 20008    | node2    | t              | t                | -927910690
12 datanode2 | D        | 20009    | node3    | f              | f                | 914546798
13 (4 rows)

```

再登录下 node3 节点的 coordinator。

```

1 $ psql -p 20005
2 psql (PGXL 10alpha2, based on PG 10.3 (Postgres-XL 10alpha2))
3 Type "help" for help.
4
5 postgres=# select * from pgxc_node;
6  node_name | node_type | node_port | node_host | nodeis_primary | nodeis_preferred | node_id
7  -----+-----+-----+-----+-----+-----+-----
8  coord1   | C        | 20004    | node2    | f              | f                | 1885696643
9  coord2   | C        | 20005    | node3    | f              | f                | -1197102633
10 datanode1 | D        | 20008    | node2    | t              | f                | -927910690
11 datanode2 | D        | 20009    | node3    | f              | t                | 914546798
12 (4 rows)

```

在任意一个 coordinator 执行如下操作

```

1 postgres=# create database peiybdb;
2 postgres=# \c peiybdb
3 peiybdb=# create table tmp_t0(c0 varchar(100),c1 varchar(100));
4 peiybdb=# insert into tmp_t0(c0,c1) SELECT id::varchar,md5(id::varchar) FROM generate_series(1,10000) as id;
5 INSERT 0 10000
6 peiybdb=# select xc_node_id,count(1) from tmp_t0 group by xc_node_id;
7  xc_node_id | count
8  -----+-----
9  -927910690 | 5081
10  914546798 | 4919
11 (2 rows)

```

五.参考文章

<https://blog.csdn.net/ctypyb2002/article/details/81104535#comments>

<https://www.jianshu.com/p/41d857a5d743>

<https://blog.csdn.net/linuxchen/article/details/81509397>

<https://www.cnblogs.com/lottu/p/5646486.html>

官方文档

<https://www.postgres-xl.org/documentation/>

<https://www.postgres-xl.org/documentation/runtime.html>

<https://www.postgres-xl.org/documentation/runtime-config.html>

<https://www.postgres-xl.org/documentation/pgxc-ctl.html>

<https://www.postgres-xl.org/>

<https://www.postgres-xl.org/overview/>

<https://www.postgres-xl.org/download/>

<https://git.postgresql.org/gitweb/?p=postgres-xl.git;a=summary>