horizontal line

**Decathlon**

Lan Song Road NO. 555

PostgreSQL PGPOOL-II Work Manual

**12th August 2019**

|  |  |  |
| --- | --- | --- |
| Document version | V 1.0.0 |  |
| Doc maintenance | Asia Paas DBA Group |  |

# Environment Prerequisite:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Service | Server Host | Server Path | Parameter | Port |
| Primary | pg-pgpool2-01 | /u01/app/postgres/data | pgport/backend\_port | 60901 |
| Standby | pg-pgpool2-02 | /u01/app/postgres/data | pgport/backend\_port | 60901 |
| Pgpool-II | 01/02 | /opt/pgpool-II | port | 60999 |
| PCP | pcp\_port | 60898 |
| Watchdog | wd\_heartbeat\_port | 60694 |

# How to build PG replication

<https://docs.google.com/document/d/1AzJN11TOuiv9ZTgqo6pt-cbrb--PlLhPMPcgQ7zZsvU/edit>

# Infrastructure

# 

# In this version, we only have two nodes for pgpool-II, while they’re lost connection, each node may think it is the master node. Which can cause some problems.

# Install pgpool-II on all nodes

Official download site

<https://www.pgpool.net/mediawiki/index.php/Downloads>

Note:

In fact, on test environment, v4.0.6 works fine with PG v11

curl -o pgpool-II-4.0.6.tar.gz http://www.pgpool.net/mediawiki/images/pgpool-II-4.0.6.tar.gz

tar zxvf pgpool-II-4.0.6.tar.gz -C /opt

#add pgs bin to environment

vim ~/.bash\_profile

PATH=$PATH:$HOME/bin:/usr/pgsql-10/bin:/opt/pgpool-II/bin

cd /opt/pgpool-II-4.0.6

./configure --prefix=/opt/pgpool-II/ --with-pgsql=/usr/pgsql-11/bin/ --with-pgsql-includedir=/usr/pgsql-11/include --with-pgsql-libdir=/usr/pgsql-11/lib --with-openssl

make && make install

Only for PG 11

curl -o llvm-toolset-7-llvm-libs-5.0.1-8.el7.x86\_64.rpm http://mirror.centos.org/centos/7/sclo/x86\_64/rh/llvm-toolset-7/llvm-toolset-7-llvm-libs-5.0.1-8.el7.x86\_64.rpm

curl -o llvm-toolset-7-runtime-5.0.1-4.el7.x86\_64.rpm http://mirror.centos.org/centos/7/sclo/x86\_64/rh/llvm-toolset-7/llvm-toolset-7-runtime-5.0.1-4.el7.x86\_64.rpm

curl -o llvm-toolset-7-llvm-5.0.1-8.el7.x86\_64.rpm http://mirror.centos.org/centos/7/sclo/x86\_64/rh/llvm-toolset-7/llvm-toolset-7-llvm-5.0.1-8.el7.x86\_64.rpm

curl -o llvm-toolset-7-clang-5.0.1-4.el7.x86\_64.rpm http://mirror.centos.org/centos/7/sclo/x86\_64/rh/llvm-toolset-7/llvm-toolset-7-clang-5.0.1-4.el7.x86\_64.rpm

curl -o llvm-toolset-7-clang-libs-5.0.1-4.el7.x86\_64.rpm http://mirror.centos.org/centos/7/sclo/x86\_64/rh/llvm-toolset-7/llvm-toolset-7-clang-libs-5.0.1-4.el7.x86\_64.rpm

curl -o devtoolset-7-gcc-c++-7.3.1-5.15.el7.x86\_64.rpm http://mirror.centos.org/centos/7/sclo/x86\_64/rh/devtoolset-7/devtoolset-7-gcc-c++-7.3.1-5.15.el7.x86\_64.rpm

curl -o devtoolset-7-libstdc++-devel-7.3.1-5.15.el7.x86\_64.rpm http://mirror.centos.org/centos/7/sclo/x86\_64/rh/devtoolset-7/devtoolset-7-libstdc++-devel-7.3.1-5.15.el7.x86\_64.rpm

curl -o llvm-toolset-7-compiler-rt-5.0.1-2.el7.x86\_64.rpm http://mirror.centos.org/centos/7/sclo/x86\_64/rh/llvm-toolset-7/llvm-toolset-7-compiler-rt-5.0.1-2.el7.x86\_64.rpm

curl -o llvm-toolset-7-libomp-5.0.1-2.el7.x86\_64.rpm http://mirror.centos.org/centos/7/sclo/x86\_64/rh/llvm-toolset-7/llvm-toolset-7-libomp-5.0.1-2.el7.x86\_64.rpm

wget http://mirror.centos.org/centos/7/sclo/x86\_64/rh/devtoolset-7/devtoolset-7-gcc-7.3.1-5.15.el7.x86\_64.rpm

wget http://mirror.centos.org/centos/7/sclo/x86\_64/rh/devtoolset-7/devtoolset-7-runtime-7.1-4.el7.x86\_64.rpm

wget http://mirror.centos.org/centos/7/sclo/x86\_64/rh/devtoolset-7/devtoolset-7-binutils-2.28-11.el7.x86\_64.rpm

wget https://download-ib01.fedoraproject.org/pub/epel/7/x86\_64/Packages/l/llvm5.0-libs-5.0.1-7.el7.x86\_64.rpm

wget <https://download-ib01.fedoraproject.org/pub/epel/7/x86_64/Packages/l/llvm5.0-5.0.1-7.el7.x86_64.rpm>

rpm -Uvh llvm-toolset-7-libomp-5.0.1-2.el7.x86\_64.rpm devtoolset-7-gcc-7.3.1-5.15.el7.x86\_64.rpm devtoolset-7-runtime-7.1-4.el7.x86\_64.rpm devtoolset-7-binutils-2.28-11.el7.x86\_64.rpm llvm5.0-5.0.1-7.el7.x86\_64.rpm llvm5.0-libs-5.0.1-7.el7.x86\_64.rpm llvm-toolset-7-clang-libs-5.0.1-4.el7.x86\_64.rpm llvm-toolset-7-clang-5.0.1-4.el7.x86\_64.rpm devtoolset-7-gcc-c++-7.3.1-5.15.el7.x86\_64.rpm devtoolset-7-libstdc++-devel-7.3.1-5.15.el7.x86\_64.rpm llvm-toolset-7-compiler-rt-5.0.1-2.el7.x86\_64.rpm

## On primary node create extension

# Function for recovery node online.

<http://www.pgpool.net/docs/latest/en/html/install-pgpool-recovery.html>

cd /opt/pgpool-II-4.0.6/src/sql/pgpool-recovery

make && make install

# handling of duplicate table names in different schema

<http://www.pgpool.net/docs/latest/en/html/install-pgpool-regclass.html>

cd /opt/pgpool-II-4.0.6/src/sql/pgpool-regclass/

make && make install

# prevent pgpool and vacuum call lock at same time

cd /opt/pgpool-II-4.0.6/src/sql

make && make install

# Allow SQL access PCP commands

<http://www.pgpool.net/docs/latest/en/html/pgpool-adm.html>

cd /opt/pgpool-II-4.0.6/src/sql/pgpool\_adm

ln -s /opt/pgpool-II/lib/libpcp.so.1 /lib64/

ldconfig -p

make install

su - postgres

psql -f /usr/pgsql-11/share/extension/pgpool-recovery.sql template1

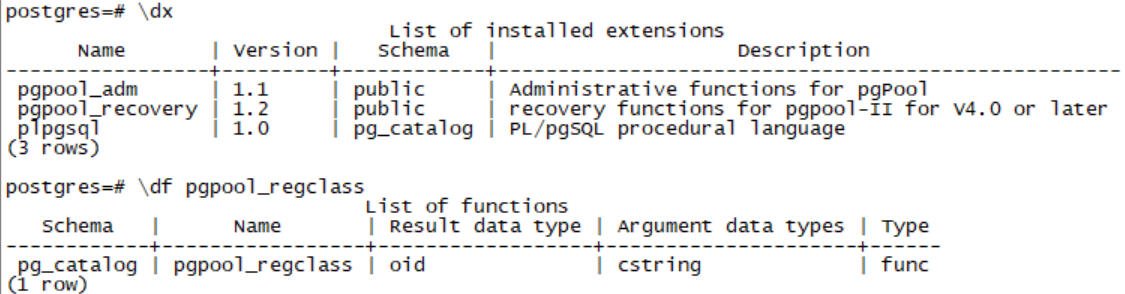
psql -f /usr/pgsql-11/share/extension/pgpool-regclass.sql template1

psql -f /opt/pgpool-II-4.0.6/src/sql/insert\_lock.sql

psql > create extension pgpool\_recovery;

psql > create extension pgpool\_regclass;

psql > create extension pgpool\_adm;



# Configure pgpool-II only for Load balance

## Copy template configuration

cd /opt/pgpool-II/etc

cp pool\_hba.conf.sample pool\_hba.conf

cp pcp.conf.sample pcp.conf

cp pgpool.conf.sample-stream pgpool.conf

# Working for Loadbalance only

# [pgpool.conf](https://docs.google.com/document/d/1CafYm8VKdectl_xXRqv1PRMfCUS8ZzE97TErGVYWbaM/edit)

### Key configuration for each node:

# watchdog hostname, local ip address

wd\_hostname

# watchdog port default 9000, we define it to 60900

wd\_port

# watchdog Weightiness, determine how much workload will this node taken.

wd\_priority

# heartbeat hostname, remote node ip address.

heartbeat\_destination0

# heartbeat port, remote node port.

heartbeat\_destination\_port0

# heartbeat device name

heartbeat\_device0

# if there are more than 2 nodes

heartbeat\_destination1

heartbeat\_destination\_port1

heartbeat\_device1

# pgpool manage configuration

other\_pgpool\_hostname0

other\_pgpool\_port0

other\_wd\_port0

# [pcp.conf](https://docs.google.com/document/d/1SLM8K_lbjNVSPTYfbj4qufGw95-IO5nfyQM6f1vOs28/edit)

# store the password of pcp manager user,use following command-line to generate password and paste them inside pcp.conf

pg\_md5 <password>

# [pool\_hba.conf](https://docs.google.com/document/d/1YXzDsh2lOpmHNUEfyOFrAdUvrolrEZWWCt_n2xfKN_Y/edit)

# pool\_hba.conf should configuration same as pg\_hba.conf, if you need pgpool to manager the access policy.

<http://www.pgpool.net/docs/latest/en/html/auth-pool-hba-conf.html>

# pool\_passwd

# add pgpool management password entry to allow pool access to pgs

<http://www.pgpool.net/docs/latest/en/html/auth-methods.html>

pg\_md5 -p -m -u <User>

# Startup and shutdown the pgpool

pgpool -n -d > /opt/pgpool-II/logs/pgpool.log 2>&1 &

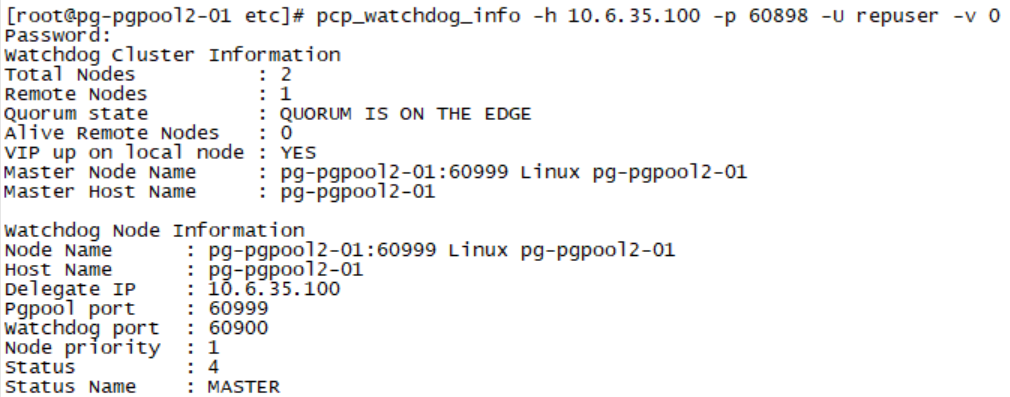
pgpool -m fast stop

# Reload configuration

pgpool reload

# verify pcp watch dog status

[root@pg-pgpool2-01 etc]# pcp\_watchdog\_info -h 10.6.35.100 -p 60898 -U repuser -v



## 

## Configure nodes authorized access

vim /etc/hosts

10.6.35.6 pg-pgpool2-02

10.6.35.4 pg-pgpool2-01

# on each node

ssh-keygen -t rsa

# copy public key from 02 to 01 which allow 02 access 01 without password

[root@pg-pgpool2-02 .ssh]# ssh-copy-id -i ~/.ssh/id\_rsa.pub root@pg-pgpool2-01

# Way 2 Paste the context of ~/.ssh/id\_rsa.pub into ~/.ssh/authorized\_keys on pg-pgpool2-01

# Verify the ssh connection

ssh pg-pgpool2-01

# Verify Load Balance Function

# Aim to confirm the result of LB

<http://www.pgpool.net/docs/latest/en/html/tutorial-testing-load-balance.html>

# create a database on primary database.

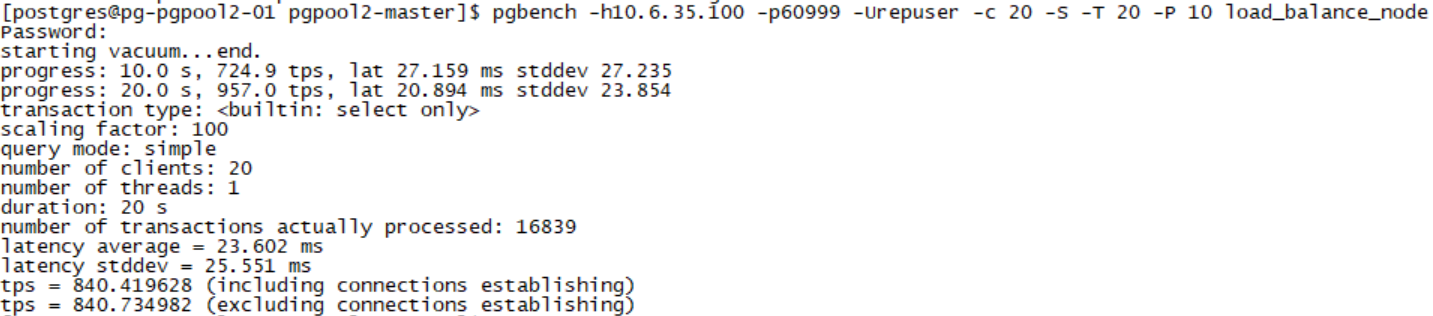
createdb -h10.6.35.4 -p60901 -Urepuser load\_balance\_node

# prepare the tables for LB test.

pgbench -i -s 100 -F 80 -h10.6.35.4 -p60999 -Urepuser -d load\_balance\_node

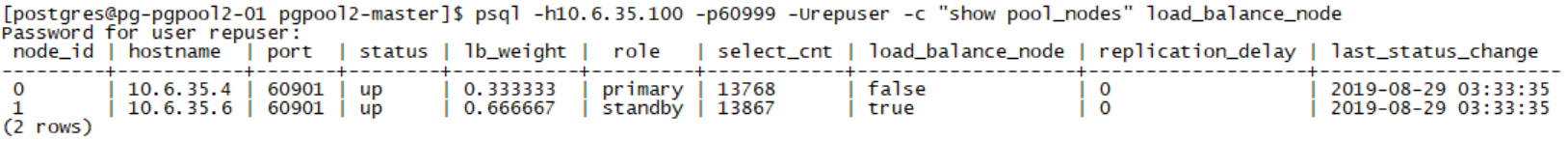
# launch simple query on pgpool connection.

pgbench -h10.x.x.100 -p60999 -Urepuser -c 20 -S -T 20 -P 10 load\_balance\_node

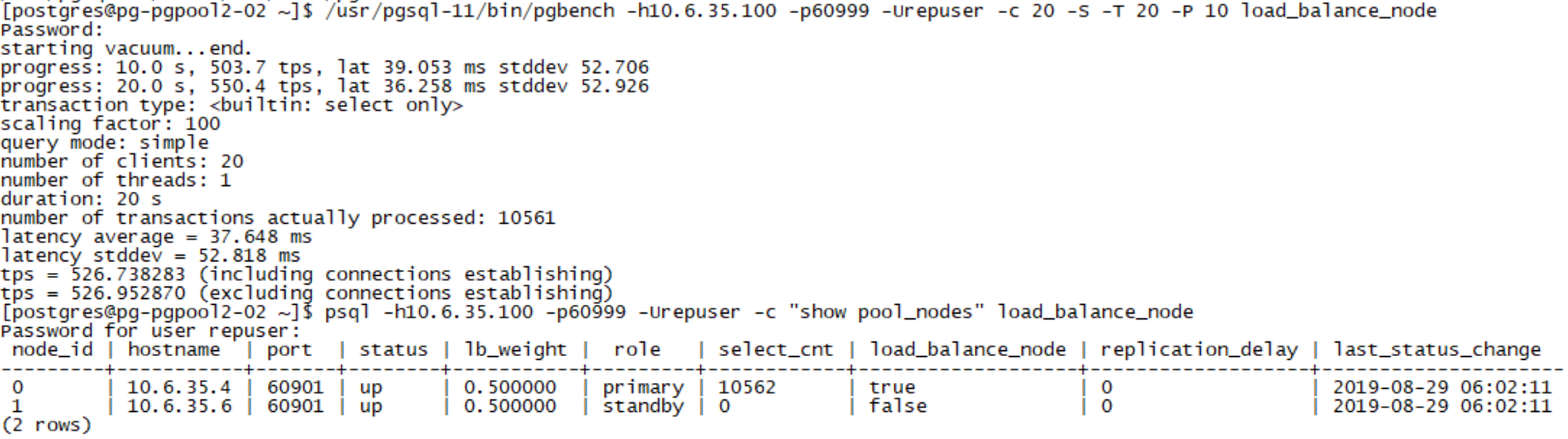


# Review result

Psql -h -p -U -c “show pool\_nodes”



# Stop load balance



## Configure pgpool-II for Failover

### http://www.pgpool.net/docs/latest/en/html/runtime-config-failover.html

### Key configuration

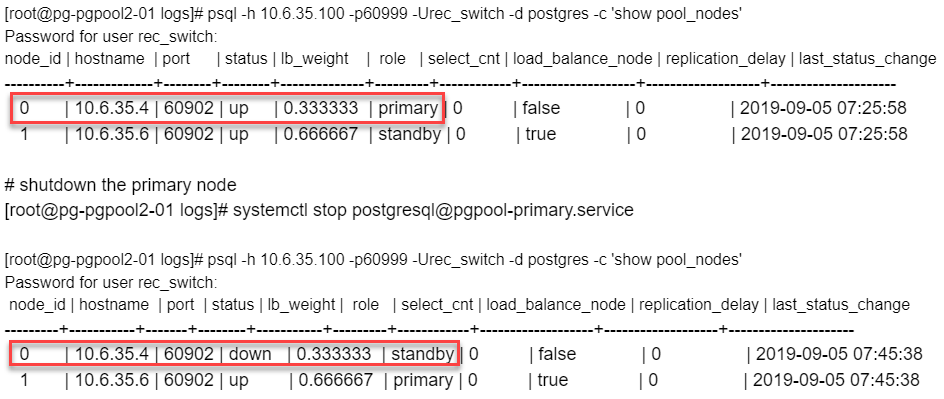
# failover and failback scripts

failover\_command = '/opt/pgpool-II/scripts/failover.sh 10.6.35.4'

failback\_command = '/opt/pgpool-II/scripts/failback.sh 10.6.35.4'

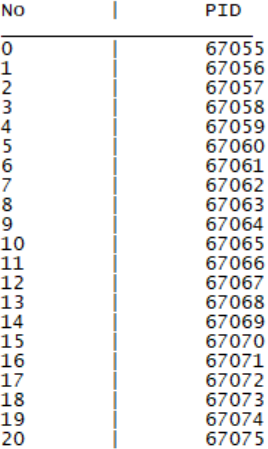
# Verify Failover Function

# We can make primary node shutdown to pretend the database can not provide services, in the case, pgpool will automatically do failover.



# Check pgpool processor

# pcp\_proc\_count -h 10.6.35.100 -p 60898 -Urepuser -v



# Defect of PGPOOL

<https://blog.csdn.net/weixin_34268843/article/details/85511897>

SimpleQuery　→　pool\_send\_and\_wait →　send\_simplequery\_message

/\*

\* Process Query('Q') message

\* Query messages include an SQL string.

\*/

POOL\_STATUS SimpleQuery(POOL\_CONNECTION \*frontend,

POOL\_CONNECTION\_POOL \*backend, int len, char \*contents){

……

/\* log query to log file if necessary \*/

if (pool\_config->log\_statement){

pool\_log("statement: %s", contents);

}else{

pool\_debug("statement2: %s", contents);

}

……

string = query\_context->original\_query;

if (!RAW\_MODE){

……

/\*

\* Query is not commit/rollback

\*/

if (!commit){

char \*rewrite\_query;

……

/\*

\* Optimization effort: If there's only one session, we do

\* not need to wait for the master node's response, and

\* could execute the query concurrently.

\*/

if (pool\_config->num\_init\_children == 1){

/\* Send query to all DB nodes at once \*/

status = pool\_send\_and\_wait(query\_context, 0, 0);

/\*

free\_parser();

\*/

return status;

}

/\* Send the query to master node \*/

if (pool\_send\_and\_wait(query\_context, 1, MASTER\_NODE\_ID)

!= POOL\_CONTINUE) {

free\_parser();

return POOL\_END;

}

}

/\*

\* Send the query to other than master node.

\*/

if (pool\_send\_and\_wait(query\_context, -1, MASTER\_NODE\_ID)

!= POOL\_CONTINUE{

free\_parser();

return POOL\_END;

}

……

}else{

……

}

return POOL\_CONTINUE;

}

/\*

\* Send simple query and wait for response

\* send\_type:

\* -1: do not send this node\_id

\* 0: send to all nodes

\* >0: send to this node\_id

\*/

POOL\_STATUS pool\_send\_and\_wait(POOL\_QUERY\_CONTEXT \*query\_context,

int send\_type, int node\_id)

{

……

/\* Send query \*/

for (i=0;i<NUM\_BACKENDS;i++){

……

per\_node\_statement\_log(backend, i, string);

if ( send\_simplequery\_message(CONNECTION(backend, i),

len, string, MAJOR(backend)) != POOL\_CONTINUE) {

return POOL\_END;

}

}

/\* Wait for response \*/

for (i=0;i<NUM\_BACKENDS;i++){

……

if (wait\_for\_query\_response(frontend, CONNECTION(backend, i),

MAJOR(backend)) != POOL\_CONTINUE){

/\* Cancel current transaction \*/

CancelPacket cancel\_packet;

cancel\_packet.protoVersion = htonl(PROTO\_CANCEL);

cancel\_packet.pid = MASTER\_CONNECTION(backend)->pid;

cancel\_packet.key= MASTER\_CONNECTION(backend)->key;

cancel\_request(&cancel\_packet);

return POOL\_END;

}

……

}

return POOL\_CONTINUE;

}

经过对程序的进一步分析和试验，可以得出以下的结论：

在 Master Node 和其他各Node之间，对SQL文的执行是串行的。

在 Master Node以外的其他各Node之间，是并行执行的。其实是

/\* Send query \*/ 一段，无阻塞方式向各节点发送SQL文。

/\* Wait for response \*/ 一段，虽然也是个循环，但是是串行。

不过好在向各节点发SQL文的时候，几乎是同时地发送命令，

所以 Wait for response 对一个节点检查获得SQL文执行结束消息以后，

几乎同时也会获得下一个节点SQL文执行结束的消息。

综合以上：如果对一个节点单独执行一段批处理耗时1小时，那么在replication mode 多个节点运行条件下，执行时间将变成 2小时。

至于为何 pgpool-II把对 Master Node和 其他Node的执行分开，也许有特殊考虑，也许是为了保证Master Node的正确性。