# Percona-XtraDB-Cluster互为见证发生脑裂情况的测试

#### 测试目的

      测试Percona-XtraDB-Cluster互为见证架构可靠性。

#### 测试环境

      1.PXC集群：

      PXC-node1:192.168.4.40

      PXC-node2:192.168.4.41

      2.见证服务器:

      garbd-node1:192.168.4.40

      garbd-node2:192.168.4.41

      3.Percona软件版本：

      Percona-XtraDB-Cluster版本：Percona-XtraDB-Cluster-57.x86\_64

      Percona-XtraDB-Garbd版本：Percona-XtraDB-Cluster-garbd-57.x86\_64

#### 测试步骤

      pxc搭建过程不再赘述，如有需要请参考[Percona XtraDB Cluster安装部署手册](http://wiki.yooli-inc.com/pages/viewpage.action?pageId=2819151)，路径：DBA团队—>How-to 文档。

   1.1 node1:my.cnf配置如下：

 Expand source

[mysqld\_safe]

pid-file =/data/mysqldata/mysql.pid

[mysqld]

# GENERAL

#basedir = /usr/local/mysql

datadir = /data/mysqldata

tmpdir = /tmp

socket = /data/mysqldata/mysql.sock

pid\_file = /data/mysqldata/mysql.pid

user = mysql

port = 3306

character-set-server = utf8

bind-address = 0.0.0.0

server-id = 3232236584

skip-name-resolve = 1

# INNODB

# This changes how |InnoDB| autoincrement locks are managed and is a requirement for Galera

default\_storage\_engine = InnoDB

innodb\_buffer\_pool\_size = 50G

innodb\_autoinc\_lock\_mode = 2

innodb\_buffer\_pool\_instances = 8

innodb\_thread\_concurrency = 16

innodb\_log\_buffer\_size = 32M

innodb\_log\_file\_size = 1024M

innodb\_online\_alter\_log\_max\_size = 512M

innodb\_open\_files = 1024

innodb\_purge\_threads = 2

innodb\_data\_home\_dir = /data/mysqldata

innodb\_data\_file\_path = ibdata1:256M:autoextend

innodb\_read\_io\_threads = 8

innodb\_write\_io\_threads = 8

innodb\_file\_per\_table = 1

innodb\_flush\_method = O\_DIRECT

innodb\_flush\_log\_at\_trx\_commit = 2

innodb\_max\_dirty\_pages\_pct = 90

innodb\_file\_format = Barracuda

innodb\_file\_format\_max = Barracuda

innodb\_buffer\_pool\_dump\_at\_shutdown = ON

innodb\_buffer\_pool\_load\_at\_startup = ON

innodb\_undo\_log\_truncate = ON

innodb\_undo\_tablespaces = 4

innodb\_strict\_mode = OFF

# MyISAM

key\_buffer\_size = 32M

# LOGS

#general\_log = 1

#general\_log\_file = /data/logs/mysql/mysql\_general.log

log\_timestamps = system

log\_warnings = 2

log\_error = /data/logs/mysql/mysql\_error.log

slow\_query\_log = ON

slow\_query\_log\_file = /data/logs/mysql/mysql\_slow.log

log\_queries\_not\_using\_indexes = 0

long\_query\_time = 1

expire\_logs\_days = 15

log-bin = mysql-bin.log

innodb\_print\_all\_deadlocks = 1

relay-log = relay-log

relay-log-index = relay-log

# BINLOG

# In order for Galera to work correctly binlog format should be ROW

binlog\_format = ROW

binlog\_cache\_size = 16M

max\_binlog\_size = 512M

# OTHER

tmp\_table\_size = 32M

max\_heap\_table\_size = 128M

query\_cache\_type = 0

query\_cache\_size = 0M

max\_connections = 1024

thread\_cache\_size = 200

open\_files\_limit = 65535

optimizer-switch = "mrr=on,mrr\_cost\_based=off,batched\_key\_access=on" #开启mmr功能

auto\_increment\_offset = 1

join\_buffer\_size = 5M

sort\_buffer\_size = 5M

sql\_mode = STRICT\_TRANS\_TABLES,NO\_ENGINE\_SUBSTITUTION

performance\_schema = ON

default\_password\_lifetime = 0

# WSREP

wsrep\_auto\_increment\_control = OFF

# Path to Galera library

wsrep\_provider = /usr/lib64/libgalera\_smm.so

# Cluster connection URL

wsrep\_cluster\_address = gcomm://192.168.34.40:4567,192.168.34.41:5567

# Node #1 address

wsrep\_node\_address = 192.168.34.40 # 本机IP地址

# SST method

wsrep\_sst\_method = xtrabackup-v2

# Cluster name

wsrep\_cluster\_name = Yooli\_TEST\_Cluster

# Authentication for SST method

wsrep\_sst\_auth = "sstuser:sstuser"

wsrep\_max\_ws\_rows = 0

wsrep\_max\_ws\_size = 2147483647

wsrep\_slave\_threads = 16

wsrep\_provider\_options = "base\_port=4567;gcache.size=50G; gcache.page\_size=512M; gcs.fc\_limit = 256;gcs.fc\_master\_slave = yes"

pxc\_strict\_mode = PERMISSIVE

#Replication

log\_slave\_updates = 1

gtid-mode = on

enforce\_gtid\_consistency = true

binlog\_checksum = CRC32

slave\_allow\_batching = 1

master\_verify\_checksum = 1

slave\_sql\_verify\_checksum = 1

master\_info\_repository = TABLE

relay\_log\_info\_repository = TABLE

slave\_parallel\_type = LOGICAL\_CLOCK

slave\_parallel\_workers = 16

relay\_log\_recovery = ON

[client]

socket = /data/mysqldata/mysql.sock

port = 3306

[mysql]

default-character-set = utf8

prompt ="\\u@\\h : \\d \\R:\\m:\\s>"

no-auto-rehash

   1.2 node1 garb配置文件如下(garb配置文件中ip与my.cnf不在同一个网段，彼此不能通信)：

# Copyright (C) 2012 Coedership Oy  
# # # This config file is to be sourced by garb service script.  
# # # A space-separated list of node addresses (address[:port]) in the cluster  
GALERA\_NODES="192.168.4.41:5567 192.168.4.40:4567"  
# # # Galera cluster name, should be the same as on the rest of the nodes.  
GALERA\_GROUP="Yooli\_TEST\_Cluster"  
# # # Log file for garbd. Optional, by default logs to syslog  
LOG\_FILE="/tmp/garbd.log"

   2.1 node2:my.cnf配置如下：

 Expand source

[mysqld\_safe]

pid-file =/data/mysqldata/mysql.pid

[mysqld]

# GENERAL

#basedir = /usr/local/mysql

datadir = /data/mysqldata

tmpdir = /tmp

socket = /data/mysqldata/mysql.sock

pid\_file = /data/mysqldata/mysql.pid

user = mysql

port = 3306

character-set-server = utf8

bind-address = 0.0.0.0

server-id = 3232236585

skip-name-resolve = 1

# INNODB

# This changes how |InnoDB| autoincrement locks are managed and is a requirement for Galera

default\_storage\_engine = InnoDB

innodb\_buffer\_pool\_size = 50G

innodb\_autoinc\_lock\_mode = 2

innodb\_buffer\_pool\_instances = 8

innodb\_thread\_concurrency = 16

innodb\_log\_buffer\_size = 32M

innodb\_log\_file\_size = 1024M

innodb\_online\_alter\_log\_max\_size = 512M

innodb\_open\_files = 1024

innodb\_purge\_threads = 2

innodb\_data\_home\_dir = /data/mysqldata

innodb\_data\_file\_path = ibdata1:256M:autoextend

innodb\_read\_io\_threads = 8

innodb\_write\_io\_threads = 8

innodb\_file\_per\_table = 1

innodb\_flush\_method = O\_DIRECT

innodb\_flush\_log\_at\_trx\_commit = 2

innodb\_max\_dirty\_pages\_pct = 90

innodb\_file\_format = Barracuda

innodb\_file\_format\_max = Barracuda

innodb\_buffer\_pool\_dump\_at\_shutdown = ON

innodb\_buffer\_pool\_load\_at\_startup = ON

innodb\_undo\_log\_truncate = ON

innodb\_undo\_tablespaces = 4

innodb\_strict\_mode = OFF

# MyISAM

key\_buffer\_size = 32M

# LOGS

#general\_log = 1

#general\_log\_file = /data/logs/mysql/mysql\_general.log

log\_timestamps = system

log\_warnings = 2

log\_error = /data/logs/mysql/mysql\_error.log

slow\_query\_log = ON

slow\_query\_log\_file = /data/logs/mysql/mysql\_slow.log

log\_queries\_not\_using\_indexes = 0

long\_query\_time = 1

expire\_logs\_days = 15

log-bin = mysql-bin.log

innodb\_print\_all\_deadlocks = 1

relay-log = relay-log

relay-log-index = relay-log

# BINLOG

# In order for Galera to work correctly binlog format should be ROW

binlog\_format = ROW

binlog\_cache\_size = 16M

max\_binlog\_size = 512M

# OTHER

tmp\_table\_size = 32M

max\_heap\_table\_size = 128M

query\_cache\_type = 0

query\_cache\_size = 0M

max\_connections = 1024

thread\_cache\_size = 200

open\_files\_limit = 65535

optimizer-switch = "mrr=on,mrr\_cost\_based=off,batched\_key\_access=on" #开启mmr功能

auto\_increment\_offset = 1

join\_buffer\_size = 5M

sort\_buffer\_size = 5M

sql\_mode = STRICT\_TRANS\_TABLES,NO\_ENGINE\_SUBSTITUTION

performance\_schema = ON

default\_password\_lifetime = 0

# WSREP

wsrep\_auto\_increment\_control = OFF

# Path to Galera library

wsrep\_provider = /usr/lib64/libgalera\_smm.so

# Cluster connection URL

wsrep\_cluster\_address = gcomm://192.168.34.41:5567,192.168.34.40:4567

# Node #1 address

wsrep\_node\_address = 192.168.34.41 # 本机IP地址

# SST method

wsrep\_sst\_method = xtrabackup-v2

# Cluster name

wsrep\_cluster\_name = Yooli\_TEST\_Cluster

# Authentication for SST method

wsrep\_sst\_auth = "sstuser:sstuser"

wsrep\_max\_ws\_rows = 0

wsrep\_max\_ws\_size = 2147483647

wsrep\_slave\_threads = 16

wsrep\_provider\_options = "base\_port=5567;gcache.size=50G; gcache.page\_size=512M; gcs.fc\_limit = 256;gcs.fc\_master\_slave = yes"

pxc\_strict\_mode = PERMISSIVE

#Replication

log\_slave\_updates = 1

gtid-mode = on

enforce\_gtid\_consistency = true

binlog\_checksum = CRC32

slave\_allow\_batching = 1

master\_verify\_checksum = 1

slave\_sql\_verify\_checksum = 1

master\_info\_repository = TABLE

relay\_log\_info\_repository = TABLE

slave\_parallel\_type = LOGICAL\_CLOCK

slave\_parallel\_workers = 16

relay\_log\_recovery = ON

[client]

socket = /data/mysqldata/mysql.sock

port = 3306

[mysql]

default-character-set = utf8

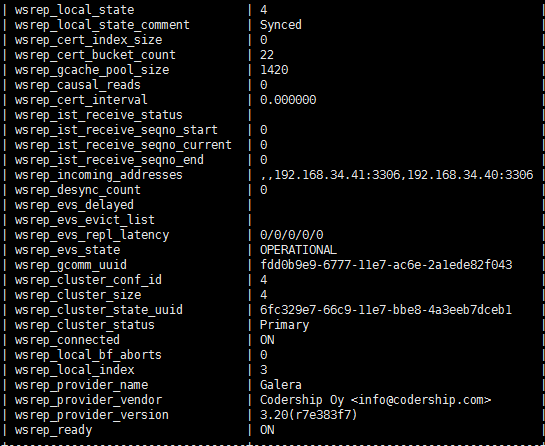
prompt ="\\u@\\h : \\d \\R:\\m:\\s>"

no-auto-rehash

   2.2 node2 garb配置文件如下(garb配置文件中ip与my.cnf不在同一个网段，彼此不能通信)：

# Copyright (C) 2012 Coedership Oy  
# # # # This config file is to be sourced by garb service script.  
# # # # A space-separated list of node addresses (address[:port]) in the cluster  
GALERA\_NODES="192.168.4.41:4567 192.168.4.40:5567"  
# # # # Galera cluster name, should be the same as on the rest of the nodes.  
GALERA\_GROUP="Yooli\_TEST\_Cluster"  
# # # # Log file for garbd. Optional, by default logs to syslog  
LOG\_FILE="/tmp/garbd.log"

   3.node1，node2节点启动mysql、garb服务，在node1上show status like 'wsrep%';状态如下：



   4.在node2上同时kill -9 mysql，garb进程

   5.node2 kill -9 进程之后，node1上mysql\_error.log内容摘要如下（注意观察后6行）：

 Expand source

2017-07-13T15:18:03.387139+08:00 0 [Note] WSREP: (fdd0b9e9, 'tcp://0.0.0.0:4567') turning message relay requesting on, nonlive peers: tcp://192.168.34.41:4567 tcp://192.168.4.41:4567

2017-07-13T15:18:04.517477+08:00 0 [Note] WSREP: (fdd0b9e9, 'tcp://0.0.0.0:4567') reconnecting to 220df60c (tcp://192.168.34.41:4567), attempt 0

2017-07-13T15:18:04.517712+08:00 0 [Note] WSREP: (fdd0b9e9, 'tcp://0.0.0.0:4567') reconnecting to 82cd007a (tcp://192.168.34.41:5567), attempt 0

2017-07-13T15:18:04.517926+08:00 0 [Note] WSREP: (fdd0b9e9, 'tcp://0.0.0.0:4567') reconnecting to 220df60c (tcp://192.168.4.41:4567), attempt 0

2017-07-13T15:18:08.520194+08:00 0 [Note] WSREP: evs::proto(fdd0b9e9, OPERATIONAL, view\_id(REG,080650e4,41)) suspecting node: 220df60c

2017-07-13T15:18:08.520272+08:00 0 [Note] WSREP: evs::proto(fdd0b9e9, OPERATIONAL, view\_id(REG,080650e4,41)) suspected node without join message, declaring inactive

2017-07-13T15:18:08.520336+08:00 0 [Note] WSREP: evs::proto(fdd0b9e9, OPERATIONAL, view\_id(REG,080650e4,41)) suspecting node: 82cd007a

2017-07-13T15:18:08.520374+08:00 0 [Note] WSREP: evs::proto(fdd0b9e9, OPERATIONAL, view\_id(REG,080650e4,41)) suspected node without join message, declaring inactive

2017-07-13T15:18:09.521537+08:00 0 [Note] WSREP: Current view of cluster as seen by this node

view (view\_id(NON\_PRIM,080650e4,41)

memb {

080650e4,0

fdd0b9e9,0

}

joined {

}

left {

}

partitioned {

220df60c,0

82cd007a,0

}

)

2017-07-13T15:18:09.521714+08:00 0 [Note] WSREP: declaring 080650e4 at tcp://192.168.34.40:5567 stable

2017-07-13T15:18:09.521717+08:00 0 [Note] WSREP: New COMPONENT: primary = no, bootstrap = no, my\_idx = 1, memb\_num = 2

2017-07-13T15:18:09.521823+08:00 0 [Note] WSREP: Flow-control interval: [256, 256]

2017-07-13T15:18:09.521846+08:00 0 [Note] WSREP: Received NON-PRIMARY.

2017-07-13T15:18:09.521860+08:00 0 [Note] WSREP: Shifting SYNCED -> OPEN (TO: 10)

2017-07-13T15:18:09.522020+08:00 1 [Note] WSREP: New cluster view: global state: 6fc329e7-66c9-11e7-bbe8-4a3eeb7dceb1:10, view# -1: non-Primary, number of nodes: 2, my index: 1, protocol version 3

2017-07-13T15:18:09.522117+08:00 1 [Note] WSREP: wsrep\_notify\_cmd is not defined, skipping notification.

2017-07-13T15:18:09.522143+08:00 0 [Note] WSREP: Current view of cluster as seen by this node

view (view\_id(NON\_PRIM,080650e4,42)

memb {

080650e4,0

fdd0b9e9,0

}

joined {

}

left {

}

partitioned {

220df60c,0

82cd007a,0

}

)

2017-07-13T15:18:09.522303+08:00 0 [Note] WSREP: New COMPONENT: primary = no, bootstrap = no, my\_idx = 1, memb\_num = 2

2017-07-13T15:18:09.522353+08:00 0 [Note] WSREP: Flow-control interval: [256, 256]

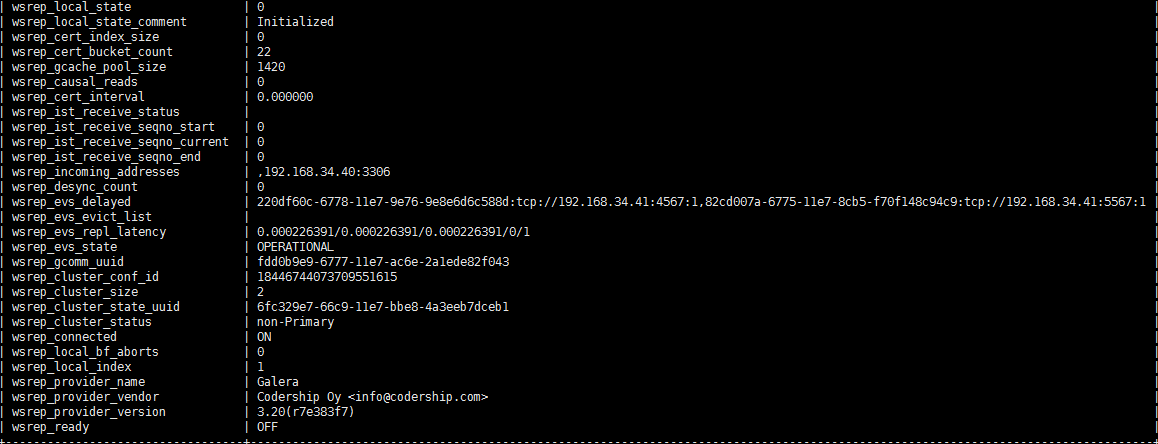
2017-07-13T15:18:09.522382+08:00 0 [Note] WSREP: Received NON-PRIMARY.

2017-07-13T15:18:09.522439+08:00 1 [Note] WSREP: New cluster view: global state: 6fc329e7-66c9-11e7-bbe8-4a3eeb7dceb1:10, view# -1: non-Primary, number of nodes: 2, my index: 1, protocol version 3

2017-07-13T15:18:09.522473+08:00 1 [Note] WSREP: wsrep\_notify\_cmd is not defined, skipping notification.

2017-07-13T15:18:48.025178+08:00 0 [Note] WSREP: (fdd0b9e9, 'tcp://0.0.0.0:4567') reconnecting to 220df60c (tcp://192.168.34.41:4567), attempt 30

   6.node1上发现keepalived已停止工作，show status like 'wsrep%';状态如下：



通过show status like 'wsrep%';发现:

       wsrep\_cluster\_size:2

               wsrep\_cluster\_status:non-Primary  
               [wsrep\_ready:off](http://wsrep_readyoff)

       从这三个状态来看node1数据库不可用，已发生脑裂，原因是集群的成员本来是4个，node2上mysql和garb服务都意外挂掉，集群成员还剩2个，小于等于总数量的50%，所以发生脑裂。

   7.恢复步骤：

      7.1 node1停掉mysql，并将grastate.dat中safe\_to\_bootstrap改为1，然后执行service mysql bootstrap-pxc启动mysql；重启见证：service garb restart；启动keepalived：service keepalived start。

      7.2 node2参考节点崩溃避免SST方法启动mysql（详情路径：DBA团队-->故障排除文章），启动见证：service garb start;启动keepalived：service keepalived start。

   8.修改node1，node2 garb配置文件

      8.1将node1 garb配置文件ip修改成跟my.cnf同一网段：

# Copyright (C) 2012 Coedership Oy  
# # # This config file is to be sourced by garb service script.  
# # # A space-separated list of node addresses (address[:port]) in the cluster  
GALERA\_NODES="192.168.34.41:5567 192.168.34.40:4567"  
# # # Galera cluster name, should be the same as on the rest of the nodes.  
GALERA\_GROUP="Yooli\_TEST\_Cluster"  
# # # Log file for garbd. Optional, by default logs to syslog  
LOG\_FILE="/tmp/garbd.log"

      8.2将node2 garb配置文件ip修改成跟my.cnf同一网段：

# Copyright (C) 2012 Coedership Oy  
# # # # This config file is to be sourced by garb service script.  
# # # # A space-separated list of node addresses (address[:port]) in the cluster  
GALERA\_NODES="192.168.34.41:4567 192.168.34.40:5567"  
# # # # Galera cluster name, should be the same as on the rest of the nodes.  
GALERA\_GROUP="Yooli\_TEST\_Cluster"  
# # # # Log file for garbd. Optional, by default logs to syslog  
LOG\_FILE="/tmp/garbd.log"

   重启node1，node2见证服务，然后在node2上再次同时kill -9 mysql,garb进程，结果发现node1还是会发生脑裂。

#### 测试结论

   1.互为见证架构只能保证在mysql服务不可用时，整个集群健康可用；如遇服务器宕机，仍会发生脑裂造成集群不可用。

   2.garb配置文件跟my.cnf配置文件中IP不在同一网段不会对服务器因宕机发生的集群不可用造成影响，仅当在bond0被流量打满的情况下，会造成集群不可用。