

MYSQL集群+HA高可用【MySQL性能调优】

MySQL Cluster 是用于解决高可用和高可靠性的解决方案。

MySQL Cluster 具有故障恢复、节点修复、数据同步、非单点故障等优点。

MySQL Cluster 是为提供 99.999%以上的高可用性而设计的，采用分布式节点设计技术，不会因为单点故障而使整个 Cluster 瘫痪。

MySQL Cluster 由 3 类节点组成：管理节点、数据节点、SQL 节点。

1: 数据节点

数据节点是整个系统中最主要的节点，它负责存储所有的数据以及数据的同步复制，以防单个或者更多的节点故障而使 MySQL Cluster 瘫痪。

2: 管理节点

管理节点用于管理系统的配置信息，只在启动和重新配置 MySQL Cluster 的时候才起作用。一般情况下只需要 1 个管理节点，当然也可以运行几个管理节点。

3: SQL 节点

SQL 节点用于数据节点存取数据，提供统一的标准 SQL 接口，跟平常的 MySQL Serve 一样，让应用程序和开发人员不用关心系统内部究竟是如何运行的。

os: centos5/rhel5 mysql version:5.1.22 rc

准备:

mysql-max-5.1.5-alpha-linux-i686-glibc23.tar

环境:

管理节点:

IP: 192.168.0.118 主机名: mgm

存?节点 1/sql 节点 1: 192.168.0.146 主机名: node1

存?节点 2/sql 节点 2: 192.168.0.221 主机名: node2

三台主机的 hosts 文件要一样

192.168.0.118 mgm

192.168.0.146 node1

192.168.0.221 node2

这里只有用一台管理节点

存?节点和 SQL 节点分别在在两台电脑上，也就是存?节点同时也当数据节点

=====

安装：分别在三台电脑安装数据库

```
shell>tar xvf mysql-max-5.1.5-alpha-linux-i686-glibc23.tar
shell> groupadd mysql
shell> useradd -g mysql mysql
```

```
shell> cd /usr/local
shell> mv mysql-max-5.1.5-alpha-linux-i686-glibc23 mysql
shell> cd mysql
shell> chown -R mysql .
shell> chgrp -R mysql .
shell> scripts/mysql_install_db --user=mysql
shell> chown -R root .
shell> chown -R mysql data
shell> bin/mysqld_safe --user=mysql &
```

复制代码

CP 启动脚本到/etc/init.d 目录

```
shell>cp support-files/mysql.server /etc/init.d/mysqld
```

复制代码

停止 mysql 数据库

```
shell>/etc/init.d/mysqld stop
```

复制代码

加入 systemV 服务

```
shell>chkconfig --add mysqld
```

复制代码

cp MYSQL 配置文件到/etc/目录

```
shell>cp support-files/my-small.cnf /etc/my.cnf
```

复制代码

测试三台上的数据库启动是否正常

二：配置管理节点

MGM:

1, 配置/etc/my.cnf 文件

在后面添加如下内容

```
[ndb_mgm]
connect-string=mgm
[ndb_mgmd]
config-file=/etc/config.ini
##connect-string=mgm ##ndb_mgm 管理节点的计算机名
##config-file=/etc/config.ini ##ndb_mgmd 的启动所要的配置文件
```

复制代码

2, ndb_mgmd 的配置文件 config.ini 配置如下

```
#
# DEFAULT SETTINGS
#

# Default settings for all data nodes
[NDBD DEFAULT]
NoOfReplicas=2
DataMemory=644245094
IndexMemory=322122547

#
# COMPUTER DEFINITIONS
#

# Datanode Computer #1
[COMPUTER]
Id=1
HostName=node1

# Datanode Computer #2
[COMPUTER]
Id=2
HostName=node2

# Management Node Computer #1
[COMPUTER]
Id=3
HostName=mgm

#
# NODE DEFINITIONS
#

# Data Nodes

# Node group #1

[NDBD]
Id=1
ExecuteOnComputer=1
```

```

Id=2
ExecuteOnComputer=2

# Management node

[NDB_MGMD]
Id=3
ExecuteOnComputer=3

# Anonymous API nodes  ##
[MYSQLD]
[MYSQLD]
[MYSQLD]
复制代码

```

分另在两点节点配置：在/etc/my.cnf 下面添加如下内容

```

[MYSQLD]
ndbcluster
ndb-connectstring=mgm
#default_table_type=NDBCLUSTER

[MYSQL_CLUSTER]
ndb-connectstring=mgm

# my.cnf
[MYSQL_CLUSTER]
ndb-connectstring=mgm #管理节点的主机名

/////////
mysql cluster 实战
////////////////////////////////////
192.168.0.118 是管理节点, 192.168.0.146、192.168.0.221 是数据节点和 sql 节点一体化
复制代码

```

启动管理节点

```

/usr/local/mysql/bin/ndb_mgmd -f /etc/config.ini
复制代码

```

注意，必须用“-f”或“--config-file”选项，告诉 ndb_mgmd 到哪里找到配置文件（详情请参见 17.5.3 节，“ndb_mgmd，“管理服务器”进程”）。

6. 启动数据节点

第一次启动

```
/usr/local/mysql/bin/ndbd --initial
```

复制代码

注意，仅应在首次启动 ndbd 时，或在备份 / 恢复或配置变化后重启 ndbd 时使用 “--initial” 参数，这很重要。原因在于，该参数会使节点删除由早期 ndbd 实例创建的、用于恢复的任何文件，包括恢复用日志文件。

非第一次启动：

```
/usr/local/mysql/bin/ndbd
```

7. 启动 sql 节点

```
service mysqld start start
```

8. 调用 ndb_mgm 管理节点客户端，可对其进行测试

```
#!/usr/local/mysql/bin/ndb_mgm
ndb_mgm> show
Connected to Management Server at: mgm:1186
Cluster Configuration
-----
[ndbd(NDB)]      2 node(s)
id=1      @192.168.0.146  (Version: 5.1.5, Nodegroup: 0)
id=2      @192.168.0.221  (Version: 5.1.5, Nodegroup: 0, Master)

[ndb_mgmd(MGM)]  1 node(s)
id=3      @192.168.0.118  (Version: 5.1.5)

[mysqld(API)]    3 node(s)
id=4      @192.168.0.221  (Version: 5.1.5)
id=5      @192.168.0.146  (Version: 5.1.5)
id=6 (not connected, accepting connect from any host)
```

复制代码

查看网络连接情况

234 管理节点：

```

[root@mgm local]# netstat -anp | grep mgm
tcp        0      0
0.0.0.0:1186                0.0.0.0:*                LISTEN      10059/ndb_mgmd

tcp        0      0
192.168.0.118:47504        192.168.0.118:1186      ESTABLISHED
10302/ndb_mgm
tcp        0      0
192.168.0.118:47503        192.168.0.118:1186      ESTABLISHED
10302/ndb_mgm
tcp        0      0
192.168.0.118:47502        192.168.0.118:1186      ESTABLISHED
10302/ndb_mgm
tcp        0      0
127.0.0.1:34878            127.0.0.1:1186          ESTABLISHED
10059/ndb_mgmd
tcp        0      0
192.168.0.118:1186        192.168.0.221:2689      ESTABLISHED
10059/ndb_mgmd
tcp        0      0
127.0.0.1:1186            127.0.0.1:34878         ESTABLISHED
10059/ndb_mgmd
tcp        0      0
192.168.0.118:1186        192.168.0.146:57511     ESTABLISHED
10059/ndb_mgmd
tcp        0      0
192.168.0.118:1186        192.168.0.118:47502     ESTABLISHED
10059/ndb_mgmd
tcp        0      0
192.168.0.118:1186        192.168.0.118:47503     ESTABLISHED
10059/ndb_mgmd
tcp        0      0
192.168.0.118:1186        192.168.0.118:47504     ESTABLISHED
10059/ndb_mgmd
tcp        0      0
192.168.0.118:1186        192.168.0.221:2356      ESTABLISHED
10059/ndb_mgmd
tcp        0      0
192.168.0.118:1186        192.168.0.221:2355      ESTABLISHED
10059/ndb_mgmd
tcp        0      0
192.168.0.118:1186        192.168.0.146:34588     ESTABLISHED 10059/ndb_mgmd
复制代码

```

数据和 SQL 节点:

```
[root@node1 data]# netstat -anp | grep 192.168.0.146
tcp        0      0
192.168.0.221:1284          192.168.0.146:51311      ESTABLISHED
3053/mysqld
tcp        0      0
192.168.0.221:4182          192.168.0.146:42435      ESTABLISHED
2714/ndbd
tcp        0      0
192.168.0.221:1545          192.168.0.146:55286      ESTABLISHED 2714/ndbd

tcp        0      0
192.168.0.221:2356          192.168.0.118:1186      ESTABLISHED
2714/ndbd
tcp        0      0
192.168.0.221:2355          192.168.0.118:1186      ESTABLISHED
2714/ndbd
tcp        0      0
192.168.0.221:1284          192.168.0.146:51311      ESTABLISHED 3053/mysqld
```

10. 测试

严重注意: 所有建表语句的 TYPE=MyISAM 必须替换为 ENGINE=NDBCLUSTER, mysql 文档说明如下:
对于将成为簇数据库组成部份的每个表, 均需要为其定义执行上述操作。完成该任务的最简单方法是, 简单地在 world.sql 文件上执行“查找-替换”, 并用 ENGINE=NDBCLUSTER 替换所有的 TYPE=MyISAM 实例。

每个 NDB 表必须有一个主键。如果在创建表时用户未定义主键, NDB 簇存储引擎将自动生成隐含的主键

10.1 建库同步测试

146 节点

```
mysql> show databases;

+-----+
| Database          |
+-----+
```

information_schema

mysql

ndb_2_fs

test

+-----+

4 rows in set (0.00 sec)

[复制代码](#)

```
mysql> create database clustertest;
```

Query OK, 1 row affected (0.34 sec)

```
mysql> show databases;
```

+-----+

Database

+-----+

information_schema

clustertest	
mysql	
ndb_2_fs	
test	

+-----+

5 rows in set (0.00 sec)

221 节点验证

mysql> show databases;

+-----+

Database	
----------	--

+-----+

information_schema	
--------------------	--

clustertest	
-------------	--

mysql	
-------	--

ndb_3_fs	
----------	--

```
| test |
```

```
+-----+
```

5 rows in set (0.00 sec)

[复制代码](#)

10.2 建表插入数据测试

221 节点

```
mysql> use clustertest;
```

Database changed

```
mysql> CREATE TABLE test (i INT) ENGINE=NDBCLUSTER;
```

Query OK, 0 rows affected (0.69 sec)

```
mysql> INSERT INTO test () VALUES (233);
```

Query OK, 1 row affected (0.02 sec)

```
mysql> select * from test;
```

```
+-----+
```

```
| i |
```

```
+-----+
```

```
| 233 |
```

```
+-----+
```

1 row in set (0.00 sec)

[复制代码](#)

146 节点验证、建表和插入数据

```
mysql> use clustertest;
```

Database changed

```
mysql> select * from test;
```

```
+-----+
```

```
| i |
```

```
+-----+
```

```
| 233 |
```

```
+-----+
```

1 row in set (0.03 sec)

```
mysql> INSERT INTO test () VALUES (232);
```

```
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from test;
```

```
+-----+
```

```
| i      |
```

```
+-----+
```

```
| 233    |
```

```
| 232    |
```

```
+-----+
```

```
2 rows in set (0.04 sec)
```

```
mysql> CREATE TABLE City (
```

```
    -> ID int(11) NOT NULL auto_increment,
```

```
-> Name char(35) NOT NULL default '',  
  
-> CountryCode char(3) NOT NULL default '',  
  
-> District char(20) NOT NULL default '',  
  
-> Population int(11) NOT NULL default '0',  
  
-> PRIMARY KEY (ID)  
  
-> ) ENGINE=NDBCLUSTER;
```

Query OK, 0 rows affected (0.93 sec)

```
mysql> INSERT INTO City VALUES (1,'Kabul','AFG','Kabol',1780000);
```

Query OK, 1 row affected (0.02 sec)

```
mysql> SELECT * FROM City;
```

```
+-----+-----+-----+-----+-----+  
| ID | Name | CountryCode | District | Population |  
+-----+-----+-----+-----+-----+
```

	1		Kabul		AFG		Kabul		1780000	
--	---	--	-------	--	-----	--	-------	--	---------	--

+	-----	+
---	-------	---

1 row in set (0.05 sec)

复制代码

221 节点验证

```
mysql> select * from test;
```

+	-----	+
---	-------	---

	i	
--	---	--

+	-----	+
---	-------	---

	232	
--	-----	--

	233	
--	-----	--

+	-----	+
---	-------	---

2 rows in set (0.00 sec)

```
mysql> SELECT * FROM City;
```

```

+-----+-----+-----+-----+-----+
| ID | Name | CountryCode | District | Population |
+-----+-----+-----+-----+-----+
| 1 | Kabul | AFG | Kabol | 1780000 |
+-----+-----+-----+-----+-----+

```

1 row in set (0.02 sec)

[复制代码](#)

10.3 删除表测试

146 节点

```

mysql> use cluster;
Database changed
mysql> show tables;
+-----+
| Tables_in_cluster |
+-----+
| huzi               |
| test               |
+-----+
2 rows in set (0.08 sec)

mysql> drop table huzi;
Query OK, 0 rows affected (0.79 sec)
复制代码

```

221 节点验证

```
mysql> use cluster
Database changed
mysql> show tables;
+-----+
| Tables_in_cluster |
+-----+
| test               |
+-----+
1 row in set (0.01 sec)

mysql>
```

10.4 删除数据库测试

146 节点删除 clustertest 数据库

```
mysql> drop database isa1;
Query OK, 0 rows affected (0.01 sec)

mysql> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| cluster           |
| huzi              |
| isa               |
| mysql             |
| test              |
+-----+
6 rows in set (0.00 sec)

复制代码
```

221 节点验证


```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cluster |
| huzi |
| isa |
| mysql |
| test |
+-----+
6 rows in set (0.00 sec)

mysql>
复制代码
```

重起

停止

管理节点可以控制所有数据节点的停止

```
/usr/local/mysql/bin/ndb_mgm -e shutdown
```

Connected to Management Server at: localhost:1186

2 NDB Cluster node(s) have shutdown.

Disconnecting to allow management server to shutdown.

11.2 启动

管理节点

```
/usr/local/mysql/bin/ndb_mgmd -f /etc/config.ini
```

数据节点

```
/usr/local/mysql/bin/ndbd
```

sql 节点可以不重起

lvs + heartbeat + mysql 集群构建真正高可用的 MYSQL 集群

基于上面的 MYSQL 集群，可以达到 MYSQL 数据库完整的备份及同时可读可写，现在我们要实现高可用，通过一个 IP 可以同时访问各个节点的 MYSQL。实现真正的高可用。

上面是两台数据节点和 SQL 节点上做 LVS + heartbeat 来实现高可用！分别在两台服务器上面安装 ipvsadm heartbeat 。我这里采用 yum 源来安装！

1，配置两台服务器/etc/hosts 文件

我的如下

```
127.0.0.1          localhost.localdomain localhost
192.168.0.118      mgm
192.168.0.146      node1
192.168.0.221      node1
```

2、集群软件安装

用两台服务器安装:

```
yum -y install ipvsadm    # ipvs 管理器
yum -y install libnet     # 库文件
yum -y install e2fsprogs  # 库文件
yum -y install heartbeat  # linux-ha
```

```
modprobe softdog    ##加载软件狗驱动程序
```

2- 拷贝配置文件

```
cp /usr/share/doc/heartbeat-2.1.*/ha.cf /etc/ha.d/
cp /usr/share/doc/heartbeat-2.1.*/authkeys /etc/ha.d/
cp /usr/share/doc/heartbeat-2.1.*/haresources /etc/ha.d/
```

```
DRP1=192.168.0.180
```

```
DIP2=192.168.0.181
```

```
VIP=192.168.0.224
```

```
REAL=192.168.0.146
```

```
REAL=192.168.0.221
```

配置 ha.cf authkeys haresources

1.ha.cf 文件内容如下

```
logfile /var/log/ha-log
```

```
logfacility    local0
```

```
keepalive 2
deadtime 30
warntime 10
initdead 120
udpport 694
ucast eth0 192.168.0.221 //备份机这个是主服务器的地址
auto_failback on
watchdog /dev/watchdog
node node1
node node2
ping 192.168.0.118
respawn hacluster /usr/lib/heartbeat/ipfail
apiauth ipfail gid=haclient uid=hacluster
```

2, authkeys 内容如下

```
#
# Authentication file. Must be mode 600
#
#
# Must have exactly one auth directive at the front.
# auth send authentication using this method-id
#
# Then, list the method and key that go with that method-id
#
# Available methods: crc sha1, md5. Crc doesn't need/want a key.
#
# You normally only have one authentication method-id listed in this file
#
# Put more than one to make a smooth transition when changing auth
# methods and/or keys.
#
#
# sha1 is believed to be the "best", md5 next best.
#
# crc adds no security, except from packet corruption.
# Use only on physically secure networks.
#
auth 1
1 crc
#2 sha1 HI!
#3 md5 Hello!
```

权限改为 600

```
chmod 600 authkeys
```

3,修改 haresources 文件 , 在最后一行添加内容

```
node1 IPaddr::192.168.0.224/24/eth0:0 ldirectord
```

至此, heartbeat 配置完成

配置 ldirectord

```
checktimeout=3
checkinterval=1
#fallback=127.0.0.1:80
autoreload=yes
logfile="/var/log/ldirectord.log"
logfile="local0"
emailalert="liuguan269@163.com"
emailalertfreq=3600
emailalertstatus=all
quiescent=yes
```

#Sample configuration for a MySQL virtual service.

```
virtual = 192.168.0.224:3306
#      real=master.up.com->slave.up.com:3306 gate 10
      real=192.168.0.146:3306 gate 10
      real=192.168.0.221:3306 gate 10
#      fallback=127.0.0.1:3306
      service=mysql
      scheduler=wrr
#      #persistent=600
#      #netmask=255.255.255.255
```

```
protocol=tcp
checktype=negotiate
login="client"
passwd="password123"
database="cluster"
request="SELECT i FROM test"
receive="233" //可要可不要
```

四，打开 ip_forward 转发

修改/etc/sysctl.conf 文件中的

把 0 改成 1

```
# Controls IP packet forwarding
net.ipv4.ip_forward = 1
```

在 shell 环境下执行

```
sysctl -p
```

五：关闭 arp 脚本,成员机使用

```
switchdr
```

内容如下

```
#!/bin/sh
```

```
# description: close lo0 and arp_ignore
```

```
VIP=192.168.0.224
```

```
./etc/init.d/functions
```

```
case "$1" in
```

```

stop)

    echo 0>/proc/sys/net/ipv4/conf/all/arp_announce    ##允许 arp 解析虚 ip

    ;;

start)

    echo "start Real Server"

    ifconfig lo:0  $VIP netmask 255.255.255.255 broadcast $VIP up

    /sbin/route add -host $VIP dev lo:0

    echo "1" >/proc/sys/net/ipv4/conf/lo/arp_ignore

    echo "2" >/proc/sys/net/ipv4/conf/lo/arp_announce

    echo "1" >/proc/sys/net/ipv4/conf/all/arp_ignore

    echo "2" >/proc/sys/net/ipv4/conf/all/arp_announce

    sysctl -p

    ;;

*)

    echo "Usage: switchdr { start|stop}"

    exit 1

esac

```

六： 把所有文件拷到另一个备份 SERVER 上

```

scp switchdr 192.168.0.221:/etc/init.d/
scp -r /etc/ha.d 192.168.0.221:/etc
scp /etc/sysconfig/ipvsadm 192.168.0.221:/etc/sysconfig/

```

注意备份节点的 ha.cf 文件

ucast eth0 192.168.0.181 要改成

ucast eth0 192.168.0.180

我这里在 118 电脑上连接数据库

```
[root@mgm yum.repos.d]# /usr/local/mysql/bin/mysql -u client -p -h 192.168.0.224
```

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 5 to server version: 5.1.5-alpha-max

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

```
mysql> show databases;
```

```
+-----+
```

```
| Database |
```

```
+-----+
```

```
| information_schema |
```

```
| cc |
```

```
| cluster |
```

```
| huzi |
```

```
| mysql |
```

```
| ndb_1_fs |
```

```
| test |
```

```
+-----+
```

```
7 rows in set (0.01 sec)
```

```
mysql>
```

在开一个 shell 连接数据库

```
[root@mgm yum.repos.d]# /usr/local/mysql/bin/mysql -u client -p -h 192.168.0.224
```

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 5 to server version: 5.1.5-alpha-max

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

```
mysql> show databases;
```

```
+-----+
| Database          |
+-----+
| information_schema |
| cluster           |
| huzi              |
| mysql             |
| test              |
+-----+
```

5 rows in set (0.00 sec)

```
mysql>
```

查看连接状态

```
[root@huzi ha.d]# ipvsadm -L -n
```

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port	Forward	Weight	ActiveConn	InActConn
TCP 192.168.0.224:3306 wrr				
-> 192.168.0.221:3306	Route	10	0	0
-> 192.168.0.146:3306	Route	0	0	0

IP Virtual Server version 1.2.1 (size=4096)

Prot LocalAddress:Port Scheduler Flags

-> RemoteAddress:Port	Forward	Weight	ActiveConn	InActConn
TCP 192.168.0.224:3306 wrr				
-> 192.168.0.221:3306	Route	10	0	1


```
-> 192.168.0.146:3306          Route    10      0      1
[root@huzi ha.d]#
```

```
[root@stu253 etc]# /usr/local/mysql/bin/mysql -u client -p -h 192.168.0.224
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 620 to server version: 5.1.5-alpha-max
```

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

```
mysql> show
      -> databases;
+-----+
| Database          |
+-----+
| information_schema |
| cluster           |
| huzi               |
| mysql             |
| test              |
+-----+
5 rows in set (0.01 sec)
```

```
mysql> exit
Bye
[root@stu253 etc]# /usr/local/mysql/bin/mysql -u client -p -h 192.168.0.224
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 5.1.5-alpha-max
```

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

```
mysql> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| cluster           |
| huzi               |
| mysql             |
| ndb_1_fs           |
+-----+
```

5 rows in set (0.04 sec)

mysql> exit

Bye

[root@stu253 etc]#

[root@node1 ha.d]# ipvsadm -L -n -c

[root@huzi ha.d]# ipvsadm -L -n -c

IPVS connection entries

pro	expire	state	source	virtual	destination
TCP	14:40	ESTABLISHED	192.168.0.253:8232	192.168.0.224:3306	192.168.0.221:3306
TCP	14:56	ESTABLISHED	192.168.0.253:8234	192.168.0.224:3306	192.168.0.221:3306

已经成功执行：ha+mysql 集群成功应用

用 NDB 管理节点查看 SQL 节点连接状态

[root@mgm yum.repos.d]# /usr/local/mysql/bin/ndb_mgm

-- NDB Cluster -- Management Client --

ndb_mgm> show

Connected to Management Server at: mgm:1186

Cluster Configuration

[ndbd(NDB)] 2 node(s)

id=1 @192.168.0.146 (Version: 5.1.5, Nodegroup: 0, Master)

id=2 (not connected, accepting connect from node2)

[ndb_mgmd(MGM)] 1 node(s)

id=3 @192.168.0.118 (Version: 5.1.5)

[mysqld(API)] 3 node(s)

id=4 @192.168.0.221 (Version: 5.1.5)

id=5 @192.168.0.146 (Version: 5.1.5)

id=6 (not connected, accepting connect from any host)

ndb_mgm>