keepalived+db2实现高可用性集群

**第一部分keepalived+共享磁盘HA**

**一．配置环境**

db2-test1 172.16.2.7

db2-test2 172.16.2.8

虚拟Ip 172.16.2.9

共享磁盘及挂载点

mount /dev/vg\_db2/lv\_db2home /db2home/

mount /dev/vg\_db2/lv\_db2data /opt/db2/data/

mount /dev/vg\_db2/lv\_db2arch /opt/db2/archivelog/

mount /dev/vg\_db2/lv\_db2bak /opt/db2/backup/

db2用户和组（注意两个节点要相同）

groupadd -g 999 db2iadm1

groupadd -g 998 db2fadm1

groupadd -g 997 dasadm1

groupadd -g 996 secs

useradd -u 999 -g db2iadm1 -m -d /db2home/db2inst1 db2inst1

useradd -u 998 -g db2fadm1 -m -d /db2home/db2fenc1 db2fenc1

useradd -u 997 -g dasadm1 -m -d /db2home/dasusr1 dasusr1

useradd -u 996 -g secs -m -d /db2home/secs secs

db2-test1的/etc/hosts配置加入：

172.16.2.7 db2-test1.dev.xianglin.com db2-test1 db2host

172.16.2.8 db2-test2.dev.xianglin.com db2-test2

db2-test2的/etc/hosts配置加入：

172.16.2.8 db2-test2.dev.xianglin.com db2-test2 db2host

172.16.2.7 db2-test1.dev.xianglin.com db2-test1

**二．安装及配置db2实例**

分别在两台主机上安装db2软件

首先在db2-test1上挂载共享磁盘后，配置实例，注意修改计算机名

cd /opt/ibm/db2/V9.7/instance/

hostname db2host

./db2icrt -u db2fenc1 db2inst1

切换到db2inst1用户后并创建数据库，完成后关闭实例，umount共享磁盘

在db2-test2上挂载共享磁盘后切换到db2inst1，查看实例能否启动。特别要注意实例启动的时候也需要修改计算机名为db2host（和配置实例是的计算机名要保持一致）。

**三．安装keepalived**

yum install keepalived –y

**四．准备db2脚本**

db2启动脚本：start\_db2ha.sh

#!/bin/bash

#start db2 ha

echo "===========================================================================" >> /etc/keepalived/scripts/keepalived.log

date +"%Y-%m-%d %T" >>  /etc/keepalived/scripts/keepalived.log

# v\_veth="eth3"

# v\_vip="172.16.2.9"

# v\_mask="255.255.254.0"

#ERROR

v\_error\_status="INSTANCE IS OPEN,SHUTDOWN FIRST MANUALLY"

v\_error\_mounting="ERROR FOR MOUNTING SHARE DISK"

v\_error\_db2iupdt="ERROR FOR DB2IUPDT"

v\_error\_db2start="ERROR FOR DB2START"

v\_errror\_vip1="ERROR VIP EXISTS"

function fun\_primary()

 {

 echo "primary" > /etc/keepalived/scripts/status

 }

function fun\_standby()

{

# echo "standby" > /opt/db2/ha/ha\_status

hostname `grep HOSTNAME /etc/sysconfig/network | awk -F '=' '{print $2}'`

exit

}

#1.check status

#1.1check instance status

if [ `ps -ef | grep db2sys | grep -v grep | wc -l` -eq 1 ];then

echo ${v\_error\_status} >>  /etc/keepalived/scripts/keepalived.log

fun\_standby

fi

#2.1set node as primary

 fun\_primary

#2.mount share disk

#2.1umount

if [ `df -hP | grep lv\_db2home | wc -l` -eq 1 ];then

umount -l /dev/vg\_db2/lv\_db2home

fi

if [ `df -hP | grep lv\_db2data | wc -l` -eq 1 ];then

umount -l /dev/vg\_db2/lv\_db2data

fi

if [ `df -hP | grep lv\_db2arch | wc -l` -eq 1 ];then

umount -l /dev/vg\_db2/lv\_db2arch

fi

if [ `df -hP | grep lv\_db2bak | wc -l` -eq 1 ];then

umount -l /dev/vg\_db2/lv\_db2bak

fi

#2.2mount

mount /dev/vg\_db2/lv\_db2home /db2home/

mount /dev/vg\_db2/lv\_db2data /opt/db2/data/

mount /dev/vg\_db2/lv\_db2arch /opt/db2/archivelog/

mount /dev/vg\_db2/lv\_db2bak /opt/db2/backup/

#2.3check mount

if [ `df -hP | grep lv\_db2home | wc -l` -ne 1 ];then

echo ${v\_error\_mounting} >>  /etc/keepalived/scripts/keepalived.log

fun\_standby

fi

if [ `df -hP | grep lv\_db2data | wc -l` -ne 1 ];then

echo ${v\_error\_mounting} >>  /etc/keepalived/scripts/keepalived.log

fun\_standby

fi

if [ `df -hP | grep lv\_db2arch | wc -l` -ne 1 ];then

echo ${v\_error\_mounting} >>  /etc/keepalived/scripts/keepalived.log

fun\_standby

fi

if [ `df -hP | grep lv\_db2bak | wc -l` -ne 1 ];then

echo ${v\_error\_mounting} >>  /etc/keepalived/scripts/keepalived.log

fun\_standby

fi

#3.set hostname

hostname db2host

#4.db2iupdt db2inst1

if [ ! -z $1 ];then

if [ $1 == "db2iupdt" ];then

if [ `/opt/ibm/db2/V9.7/instance/db2iupdt db2inst1 | head -1 | awk '{print $5}'` != "successfully." ];then

echo ${v\_error\_db2iupdt} >>  /etc/keepalived/scripts/keepalived.log

fun\_standby

fi

fi

fi

#5.db2start

if [ `su - db2inst1 -c db2start | tail -1 | awk '{print $5}'` != "successful." ];then

echo ${v\_error\_db2start} >>  /etc/keepalived/scripts/keepalived.log

fun\_standby

fi

# #6.vip

# if [ `ifconfig | grep ${v\_vip} | wc -l` -eq 1 ];then

# echo ${v\_errror\_vip1}

# fun\_standby

# fi

# ifconfig ${v\_veth}:1 ${v\_vip} netmask ${v\_mask} up

echo "set to primary successfully" >>  /etc/keepalived/scripts/keepalived.log

db2关闭脚本：stop\_db2ha.sh

#!/bin/bash

#stop db2 ha

echo "===========================================================================" >> /etc/keepalived/scripts/keepalived.log

date +"%Y-%m-%d %T" >>  /etc/keepalived/scripts/keepalived.log

v\_error\_stopinstance="ERROR FOR DB2STOP"

# v\_veth="eth3"

# v\_vip="172.16.2.9"

#1.shutdown instance

if [ `ps -ef | grep db2sys | grep -v grep | wc -l` -eq 1 ];then

if [ `su - db2inst1 -c "db2stop force" | tail -1 | awk '{print $1}'` != "SQL1064N" ];then

echo ${v\_error\_stopinstance} >>  /etc/keepalived/scripts/keepalived.log

exit

fi

fi

#2.shutdown vip

# if [ `ifconfig | grep ${v\_vip} | wc -l` -eq 1 ];then

# ifconfig ${v\_veth}:1 down

# fi

#3.umount share disk

if [ `df -hP | grep lv\_db2home | wc -l` -eq 1 ];then

umount -l /dev/vg\_db2/lv\_db2home

fi

if [ `df -hP | grep lv\_db2data | wc -l` -eq 1 ];then

umount -l /dev/vg\_db2/lv\_db2data

fi

if [ `df -hP | grep lv\_db2arch | wc -l` -eq 1 ];then

umount -l /dev/vg\_db2/lv\_db2arch

fi

if [ `df -hP | grep lv\_db2bak | wc -l` -eq 1 ];then

umount -l /dev/vg\_db2/lv\_db2bak

fi

#4.set standby

echo "standby" >  /etc/keepalived/scripts/status

hostname `grep HOSTNAME /etc/sysconfig/network | awk -F '=' '{print $2}'`

echo "set to slave successfully" >>  /etc/keepalived/scripts/keepalived.log

db2监控脚本：ha\_monitor.sh

#!/bin/bash

#db2 monitor

function isok()

{

rm -f /tmp/monitor.tmp

su - db2inst1 -c "db2 -tvf /etc/keepalived/scripts/monitor.sql" > /dev/null 2>&1

if [ -f /tmp/monitor.tmp  ]&&[ `cat /tmp/monitor.tmp` -eq 0 ];then

MYSQL\_OK=1

else

MYSQL\_OK=0

fi

rm -f /tmp/monitor.tmp

return MYSQL\_OK

}

#2015-10-27,add ping gateway

v\_gateway=`route | grep default | awk '{print $2}'`

for((i=1;i<=5;i++))

do

if [ `ping -c 1 ${v\_gateway} | grep transmitted | awk '{print $4}'` -eq 1 ];then

break

fi

service keepalived stop

exit 1

done

if [ $HOSTNAME == "db2host" ]&&[ `cat /etc/keepalived/scripts/status` == "primary" ];then

MYSQL\_OK=1

CHECK\_TIME=3

while [ $CHECK\_TIME -ne 0 ]

do

let "CHECK\_TIME -= 1"

isok

if [ $MYSQL\_OK = 1 ] ; then

CHECK\_TIME=0

exit 0

fi

if [ $MYSQL\_OK -eq 0 ] &&  [ $CHECK\_TIME -eq 0 ]

then

service keepalived stop

exit 1

fi

sleep 1

done

fi

monitor.sql

connect to secs;

export to /tmp/monitor.tmp of del select 0 from SYSIBM.SYSDUMMY1;

terminate;

**五．keepalived配置文件**

/etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

   notification\_email {

#

}

#   smtp\_connect\_timeout 3

   smtp\_server smtp.exmail.qq.com

   router\_id db2-ha

}

vrrp\_script check\_run {

#   script "/etc/keepalived/scripts/keepalived\_check\_mysql.sh"

   script "/etc/keepalived/scripts/ha\_monitor.sh"

   interval 20

}

vrrp\_sync\_group VGdb2 {

    group {

          VI\_db2

    }

}

vrrp\_instance VI\_db2 {

#    state MASTER

    state BACKUP

    interface eth3

    virtual\_router\_id 99

    priority 100

    advert\_int 1

    nopreempt

    authentication {

        auth\_type PASS

        auth\_pass 1111

    }

    track\_script {

        check\_run

    }

    virtual\_ipaddress {  #??IP

        172.16.2.9

    }

    notify\_master /etc/keepalived/scripts/start\_db2ha.sh

    notify\_backup /etc/keepalived/scripts/stop\_db2ha.sh

    notify\_fault /etc/keepalived/scripts/stop\_db2ha.sh

    notify\_stop /etc/keepalived/scripts/stop\_db2ha.sh

}

**六．启动keepalived**

分别在两台主机上启动keepalived服务，先启动的自动成为master，后启动的为standby。当master的db2无法正常提供服务后，keepalived会自动将虚拟ip漂移到standby上，并运行start\_db2ha.sh脚本启动db2。

**七．HA缺陷**

1.HA虽然在数据库管理节点上达到了灾备效果，但是数据文件是放在共享存储上的，只有一份，当共享存储挂掉后数据库还是要出问题的

2.为了解决存放在共享存储上的数据只有一份的问题，一般还要在HA的基础上再做个主从的复制，保证即使数据问题除了问题仍然能够切换到主从复制的从机上继续提供数据库服务。这样一来完成这一整个的机器则至少需要三台主机，同一时间点只有一台主机能够提供服务，硬件使用效率比较低

3.双机集群理论上会出现脑裂的问题。当两主机之间出现网络通信问题但主机数据库服务仍然正常的情况下，会出现主备机数据库同时启动并争夺共享磁盘上的数据文件，造成数据文件损坏。

**第二部分keepalived+db2 hadr实现主从复制**

**一．搭建hadr从机简要配置**

**1.环境**

主库 db2-test1 192.168.23.221 60000 secs

备库 db2-test2 192.168.23.222 60000 secs

**2.必要配置**

db2 update db cfg for secs using logarchmeth1 "disk:/opt/archivelog"

db2 update db cfg for secs using trackmod on

db2 update db cfg for secs using logindexbuild on

db2 update db cfg for secs using indexrec restart

**3.设置hadr通讯端口**

主备库都要设置

[root@db2-test1 ~]# vi /etc/services

在最后添加

DB2\_HADR\_SECS1 60006/tcp

DB2\_HADR\_SECS2 60007/tcp

**4.配置主机hadr参数**

db2 update db cfg for secs using hadr\_local\_host db2-test1

db2 update db cfg for secs using hadr\_remote\_host db2-test2

db2 update db cfg for secs using hadr\_local\_svc DB2\_HADR\_SECS1

db2 update db cfg for secs using hadr\_remote\_svc DB2\_HADR\_SECS2 deferred

db2 update db cfg for secs using hadr\_remote\_inst db2inst1

db2 update db cfg for secs using hadr\_syncmode async deferred

**5.备份主机并发送备份到备机进行恢复**

db2 backup database secs to /home/db2inst1/dbbak/

scp SECS.0.db2inst1.NODE0000.CATN0000.20151009230558.001  db2inst1@db2-test2:/home/db2inst1/dbbak

在备机上进行恢复

db2 restore database secs from /home/db2inst1/dbbak/ taken at 20151009230558

**6.配置备机hadr参数**

db2 update db cfg for secs using hadr\_local\_host db2-test2

db2 update db cfg for secs using hadr\_remote\_host db2-test1

db2 update db cfg for secs using hadr\_local\_svc DB2\_HADR\_SECS2

db2 update db cfg for secs using hadr\_remote\_svc DB2\_HADR\_SECS1

**7.启动hadr**

先在备机启动hadr

db2 start hadr on database secs as standby

再在主机启动hadr

db2 start hadr on database secs as primary

**8.主从切换**

普通切换（切换过程都在备机执行），主备关系互换，可保证数据不丢失

db2 takeover hadr on database secs

强制切换，备机升级为主机，与原主机脱离主从关系，根据timeout参数有可能会丢失部分数据

db2 takeover hadr on database secs by force

**二．配置keepalived实现自动切换**

settoslave.sh

#!/bin/bash

date +"%Y-%m-%d %T" >>  /etc/keepalived/scripts/keepalived.log

#if [ `su - db2inst1 -c "db2stop force" | tail -1 | awk '{print $1}'` != "SQL1064N" ];then

# echo "set to slave,but stop db2 failed" >>  /etc/keepalived/scripts/status/keepalived.log

# echo "" >> /etc/keepalived/scripts/status/keepalived.log

# exit

#fi

echo "set to slave successfully" >>  /etc/keepalived/scripts/keepalived.log

echo "" >> /etc/keepalived/scripts/keepalived.log

echo "standby" >  /etc/keepalived/scripts/status

settoprimary.sh

#!/bin/bash

date +"%Y-%m-%d %T" >>  /etc/keepalived/scripts/keepalived.log

if [ `su - db2inst1 -c "db2 get db cfg for secs | grep 'HADR database role'" | awk '{print $5}'` == "STANDBY" ];then

if [ `su - db2inst1 -c "db2 takeover hadr on database secs by force" | awk '{print $1}'` != "DB20000I" ];then

echo "set to primary,but takeover failed" >>  /etc/keepalived/scripts/keepalived.log

echo ""

exit

fi

fi

echo "set to primary successfully" >>  /etc/keepalived/scripts/keepalived.log

echo "" >> /etc/keepalived/scripts/keepalived.log

echo "primary" >  /etc/keepalived/scripts/status

ha\_monitor.sh

#!/bin/bash

#db2 monitor

function isok()

{

rm -f /tmp/monitor.tmp

su - db2inst1 -c "db2 -tvf /etc/keepalived/scripts/monitor.sql" > /dev/null 2>&1

if [ -f /tmp/monitor.tmp  ]&&[ `cat /tmp/monitor.tmp` -eq 0 ];then

MYSQL\_OK=1

else

MYSQL\_OK=0

fi

rm -f /tmp/monitor.tmp

return $MYSQL\_OK

}

#2015-10-27,add ping gateway

v\_gateway=`route | grep default | awk '{print $2}'`

for((i=1;i<=5;i++))

do

if [ `ping -c 1 ${v\_gateway} | grep transmitted | awk '{print $4}'` -eq 1 ];then

break

fi

service keepalived stop

exit 1

done

if [ `cat /etc/keepalived/scripts/status` == "primary" ];then

MYSQL\_OK=1

CHECK\_TIME=3

while [ $CHECK\_TIME -ne 0 ]

do

let "CHECK\_TIME -= 1"

isok

if [ $MYSQL\_OK = 1 ] ; then

CHECK\_TIME=0

exit 0

fi

if [ $MYSQL\_OK -eq 0 ] &&  [ $CHECK\_TIME -eq 0 ]

then

service keepalived stop

exit 1

fi

sleep 1

done

fi

monitor.sql

connect to secs;

export to /tmp/monitor.tmp of del select 0 from SYSIBM.SYSDUMMY1;

terminate;

keepalived.conf

! Configuration File for keepalived

global\_defs {

   notification\_email {

#         yuxiaodong@xianglin.cn

   }

#   notification\_email\_from yuxiaodong@xianglin.cn

   smtp\_connect\_timeout 3

   smtp\_server smtp.exmail.qq.com

   router\_id db2-hadr

}

vrrp\_script check\_run {

#   script "/etc/keepalived/scripts/keepalived\_check\_mysql.sh"

   script "/etc/keepalived/scripts/ha\_monitor.sh"

   interval 20

}

vrrp\_sync\_group VGdb2\_hadr {

    group {

          VI\_db2\_hadr

    }

}

vrrp\_instance VI\_db2\_hadr {

#    state MASTER

    state BACKUP

    interface eth0

    virtual\_router\_id 93

    priority 100

    advert\_int 1

    nopreempt

    authentication {

        auth\_type PASS

        auth\_pass 1111

    }

    track\_script {

        check\_run

    }

    virtual\_ipaddress {  #??IP

        172.16.2.11

    }

    notify\_master /etc/keepalived/scripts/settoprimary.sh

    notify\_backup /etc/keepalived/scripts/settoslave.sh

    notify\_fault /etc/keepalived/scripts/settoslave.sh

    notify\_stop /etc/keepalived/scripts/settoslave.sh

}

操作步骤：

1.首先在主备机上启动db2实例以及hadr服务

2.先在hadr主的机器上启动keepalived，当主机的/etc/keepalived/scripts/status显示为primary后，才能在备机上启动keepalived

注意事项：

1.当需要操作switch切换操作的时间，首先关闭备机的keepalived服务，再关闭主机的keepalived服务，最后再执行takeover操作

2.keepalived所执行的切换都是强制切换。原备机被强制切换为主机后应立即进行备份，待原主机可工作后在上面进行恢复并重新配置hadr

3.强制切换过后两台主机的hadr角色有可能都是primary，此时要通过/etc/keepalived/scripts/status的值来判断此时哪个才是真正的primary

**第三部分 两种方案对比**

共享磁盘ha:

两个节点处于对等状态，只有启动之后才有主备之分。当一台主机发生故障并解决之后可以立刻加入集群。但是要提供最高的ha服务则需要三台主机，硬件利用率较低，而且理论上存在因集群脑裂而产生的数据库文件损坏的问题。

hadr主从复制：

有严格的主备之分，所有的操作都要遵循一定的顺序。当发生强制切换之后需要利用备份重新配置备机。这样集群处于单节点时间要长于共享磁盘ha的方案。

**第四部分 备份机制**

无论采取何种集群方式，组织有效的备份机制都是非常必要的。现在使用的是db2move的逻辑形式备份，建议在此基础上再加上db2 backup的物理形式备份，以达到更好的保护机制，在出现问题后有更多可选的恢复方案。

**一．物理备份的优缺点：**

1.优点

基于数据文件块的备份效率更高，速度更快；

根据数据文件块的变化可以实现增量备份的效果；

通过归档及日志能达到前滚的效果，更大可能地保护数据，减少故障后丢失数据的损失

2.缺点

物理备份恢复的粒度最低也是表空间级别的，相对逻辑备份粒度较大；

无法实现跨平台恢复；

**二．备份脚本**

db2backup.sh

#!/bin/bash

#db2 backup

source /home/db2inst1/.bash\_profile

BASE\_PATH=/opt/dbbak

v\_archpath=/opt/archivelog/db2inst1/SECS/NODE0000

#get dbinfo

source /home/db2inst1/shell/dbinfo

#format as 20150929

v\_date=`date +%Y%m%d`

#format as 201509

v\_month=`date +%Y%m`

mkdir -p ${BASE\_PATH}/${v\_db}/${v\_month}/${v\_date}/{dbbak,dbdump}

cd ${BASE\_PATH}/${v\_db}/${v\_month}/${v\_date}/dbdump

db2look -d ${v\_db} -e -a -o ${v\_db}\_${v\_date}.sql -i ${v\_user} -w ${v\_pass}

db2move ${v\_db} export -u ${v\_user} -p ${v\_pass}

if [ ! -f ${BASE\_PATH}/${v\_db}/${v\_month}/${v\_date}/dbdump/${v\_db}\_${v\_date}.sql ];then

echo "db2look failed"

fi

v\_db2move\_out=`tail -3 EXPORT.out | head -1 | awk '{print $5}' | awk -F '!' '{print $1}'`

if [ ${v\_db2move\_out} != "successful" ];then

echo "db2move failed"

fi

db2 backup database ${v\_db} online to ${BASE\_PATH}/${v\_db}/${v\_month}/${v\_date}/dbbak

v\_db\_upper=`echo ${v\_db} | tr [:lower:] [:upper:]`

v\_backup\_out=`db2ckbkp ${BASE\_PATH}/${v\_db}/${v\_month}/${v\_date}/dbbak/${v\_db\_upper}\*.${v\_date}\* | tail -1 | awk '{print $5}'`

if [ ${v\_backup\_out} != "successful." ];then

echo "db2backup failed"

fi

cd  ${BASE\_PATH}/${v\_db}/${v\_month}

tar zcvf ${v\_db}\_${v\_date}.tar.gz ${v\_date}

rm -rf ${v\_date}

db2 connect to ${v\_db}

db2 prune history "${v\_date}000000"

cd ${v\_archpath}

find ./  -name "\*.LOG" -mtime +60 -exec rm -f {} \;

db2 terminate

**三．必要配置的修改**

当要使用物理备份时，需要对数据库的一些默认配置进行修改。

**1.LOGARCHOPT1**

定义：此参数用于指定已归档日志的主要目标的介质类型及路径。要实现物理热备，必须要开启次参数。

默认值：OFF

建议值：DISK:/opt/db2/archivelog（或其他路径，并建议和数据文件，日志文件分开存储）

**2.LOGBUFSZ**

定义：日志缓冲内存大小

默认值：（4KB）256

建议值：（4KB）512（或更大1024）

**3.LOGFILSIZ**

定义：每个日志（包括主日志和辅助日志）的大小

默认值：（4KB）1024

建议值：（4KB）10240

**4.LOGPRIMARY**

定义：主日志文件的个数

默认值：13

建议值：23

**5.TRACKMOD**

定义：增量备份开关

默认值：NO

建议值：YES