

31.MySQL高可用架构–MHA√

1.MHA高可用简介

1.1 什么是高可用？

Bash | Copy

1	企业高可用标准：全年无故障时间				
2	无故障时间	故障时间			
3	99.9%	0.1%	= 525.6 min	KA(keepalived)+双主	: 架构需要人为干预
4	99.99% (与钱无关的互联网行业)	0.01%	= 52.56 min	MHA+增强半同步+GTID	: 半自动化
5	99.999% (金融级别)	0.001%	= 5.256 min	PXC 、 MGR 、 MGC	: 多节点提高读写, 保
6	99.9999%	0.0001%	= 0.5256 min	自动化、云化、平台化	:

1.2 MHA的软件结构(脚本)



1.2.1 manager 组件（脚本）

Bash | Copy

1	masterha_manger	启动MHA
2	masterha_check_ssh	检查MHA的SSH配置状况
3	masterha_check_repl	检查MySQL复制状况
4	masterha_master_monitor	检测master是否宕机 （检查心跳）
5	masterha_check_status	检测当前MHA运行状态
6	masterha_master_switch	控制故障转移（自动或者手动）
7	masterha_conf_host	添加或删除配置的server信息

1.2.2 node 组件（脚本）

Bash | Copy

1	save_binary_logs	保存和复制master的二进制日志（数据补偿）
2	apply_diff_relay_logs	识别差异的中继日志事件并将其差异的事件应用于其他的
3	purge_relay_logs	清除中继日志（不会阻塞SQL线程）

1.3 站在产品经理角度，评估高可用软件设计

Bash | Copy

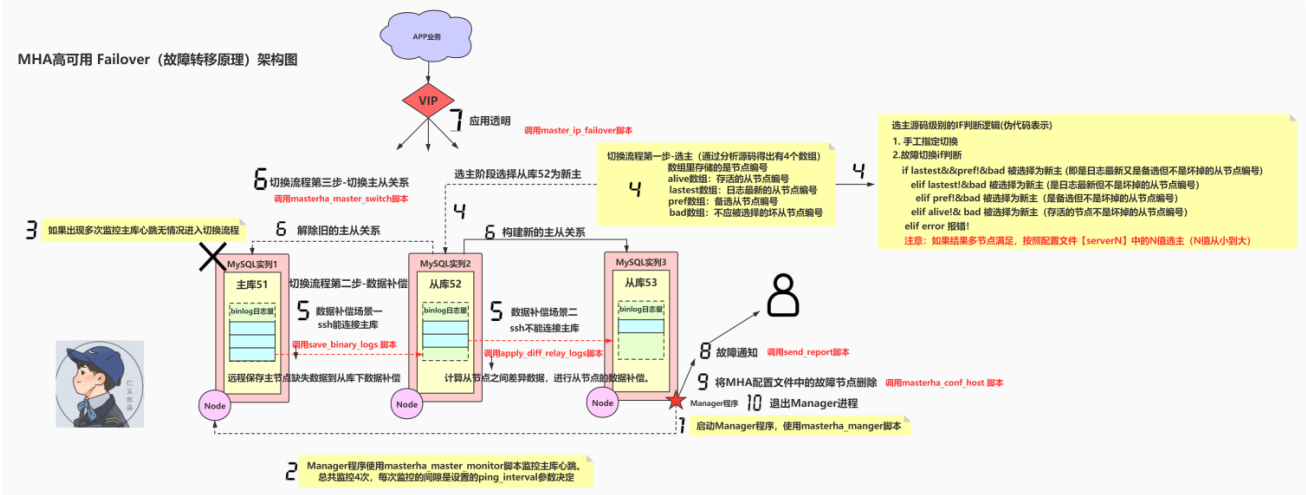
- 1 [3.1 监控](#)
- 2 [3.2 选主](#)
- 3 [3.3 数据补偿](#)
- 4 [3.4 故障转移](#)
- 5 [3.5 应用透明](#)
- 6 [3.6 自动提醒](#)
- 7 [3.7 自愈\(待开发\)--docker](#)

1.4 MHA FailOver 原理

Bash | Copy

```
1  ## 1.启动manager程序，使用masterha_manger脚本
2  ## 2.监控
3  manager程序使用masterha_master_monitor脚本进行监控主库心跳
4  总共监控4次，每次监控时间的间隔是我们设置的ping_interval参数
5  ## 3.如果出现多次监控主库心跳无情况进入切换流程
6  ## 4.切换流程第一步-选主（通过分析源码得出有4个数组）
7  ### 4.0 数组里存储的是节点编号
8      alive数组：存活的从节点编号
9      lastest数组：日志最新的从节点编号
10     pref数组：备选从节点编号
11     bad数组：不应被选择的坏从节点编号
12
13  ### 4.1如何计算数组中的值？
14  #### 4.1.1 lastest数组：
15      show slave status\G;
16      看从库接受到的日志
17      Master_Log_File:
18      Read_Master_Log_Pos:
19  #### 4.1.2 Pref数组
20      与MHA配置文件中的candidate_master参数有关，candidate_master>=0
21  #### 4.1.3 bad数组
22      a.没有打开binlog
23      b.no_master=1参数
24      c.日志差异较大（1个亿）
25
26  ### 4.2 选主-源码级别的IF判断逻辑（伪代码表示）
27  #### 4.2.1 手工指定切换
28  #### 4.2.2 故障切换if判断
29      if lastest&&pref!&bad 被选择为新主（即是日志最新又是备选但不是坏掉的从节点编号）
30      elif lastest!&bad      被选择为新主（是日志最新但不是坏掉的从节点编号）
31      elif pref!&bad         被选择为新主（是备选但不是坏掉的从节点编号）
32      elif alive!& bad       被选择为新主（存活的节点不是坏掉的从节点编号）
33      elif error 报错!
34  注意：如果结果多节点满足，按照配置文件【serverN】中的N值选主（N值从小到大）
35
36  ## 5.数据补偿
37  ### 5.1 ssh能连接主库
38      调用save_binary_logs 脚本，远程保存主库缺失部分日志到从节点/var/tmp/xxx下数据补偿
39  ### 5.1 ssh不能连接主库
40      调用apply_diff_relay_logs脚本，计算从节点之间差异数据，进行从节点的数据补偿。
41  ## 6.切换主从关系
42      调用masterha_master_switch脚本
43  1.解除旧的主从关系
44  mysql> stop slave;reset slave all;
45  2.构建新的主从关系
46  mysql> change master to; start slave;
47  ## 7.vip 应用透明
48      调用master_ip_failover脚本
49  ## 8. 故障通知
50  使用send_report
```

```
51  ## 9. 将MHA的配置文件故障节点删除
52  调用masterha_conf_host 脚本
53  ## 10. 退出程序
54  因为MHA是一次性高可用
```



1.5 MHA版本对应的MySQL版本

```
1  数据库版本对应MHA版本
2  mysql5.6--->MHA 0.56
3  mysql5.7--->MHA 0.57
4  mysql8.0--->MHA 0.58
```

2.搭建MHA架构步骤

2.1 MHA基础环境准备

a.准备gtid主从环境

准备一主两从的GTID模式主从架构

b.配置关键程序软连接(所有节点)

配置的原因是MHA源码种调用程序的格式是/usr/bin

Bash | Copy

```
1 ln -s /usr/local/mysql/bin/mysqlbinlog /usr/bin/mysqlbinlog
2 ln -s /usr/local/mysql/bin/mysql /usr/bin/mysql
```

c.配置各节点互信（各节点之间无密码ssh）

Bash | Copy

```
1 生成公钥私钥(51)
2 ssh-keygen -t rsa
3 ssh-copy-id -i ~/.ssh/id_rsa.pub 10.0.0.51
4 scp -rp ~/.ssh 10.0.0.52:/root/.ssh
5 scp -rp ~/.ssh 10.0.0.53:/root/.ssh
6
7 DB01节点:
8 [root@db01 ~]# ssh 10.0.0.51 hostname
9 db01
10 [root@db01 ~]# ssh 10.0.0.52 hostname
11 db02
12 [root@db01 ~]# ssh 10.0.0.53 hostname
13 db03
14
15 DB02节点:
16 [root@db02 ~]# ssh 10.0.0.51 hostname
17 db01
18 [root@db02 ~]# ssh 10.0.0.52 hostname
19 db02
20 [root@db02 ~]# ssh 10.0.0.53 hostname
21 db03
22
23 DB03节点:
24 [root@db03 ~]# ssh 10.0.0.51 hostname
25 db01
26 [root@db03 ~]# ssh 10.0.0.52 hostname
27 db02
28 [root@db03 ~]# ssh 10.0.0.53 hostname
29 db03
```

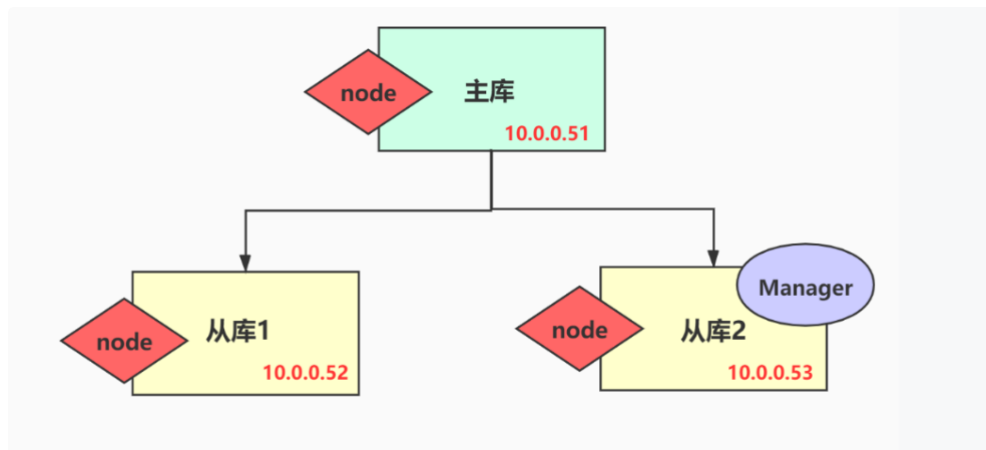
d.安装MHA软件

1.下载mha软件

mha官网：<https://code.google.com/archive/p/mysql-master-ha/> <<https://code.google.com/archive/p/mysql-master-ha/>>

github下载地址：<https://github.com/yoshinorim/mha4mysql-manager/wiki/Downloads> <<https://github.com/yoshinorim/mha4mysql-manager/wiki/Downloads>>

2.上传软件包到数据库（/opt下）



3.Node安装（所有节点）

```
1 1. 因为MHA是perl语言书写，所以先安装perl的依赖包
2
3 yum -y install perl-DBD-MySQL
4 2. 安装node
   rpm -ivh mha4mysql-node*.rpm
```

4.Manager安装（53节点）

```
1 yum install -y perl-Config-Tiny epel-release perl-Log-Dispatch perl-Parallel-ForkManager perl-Time-HiRes
2 yum install -y mha4mysql-manager*.rpm
```

e.主库(51)上创建MHA用户

```
1 create user mha@'10.0.0.%' identified with mysql_native_password by 'mha';
2 grant all privileges on *.* to mha@'10.0.0.%' ;
```

f. MHA配置文件的准备(53节点)

1.创建MHA配置文件目录

```
1 mkdir -p /etc/mha
```

2.创建MHA日志目录

Bash | Copy

```
1  mkdir -p /var/log/mha/app1
```

3.编辑MHA配置文件

Bash | Copy

```
1  和mysql类似也是通过标签控制不同配置信息
2
3  vim /etc/mha/app1.cnf
4  [server default]                第一部分：整个程序共用配置信息
5  manager_log=/var/log/mha/app1/manager      MHA工作日志存放文件
6  manager_workdir=/var/log/mha/app1          MHA工作日志存放目录路径
7  master_binlog_dir=/data/3306/binlog/       主库二进制日志存储路径
8  user=mha                                  MHA程序中manager的管理用户名
9  password=mha                             MHA程序中manager的管理用户密码
10 ping_interval=2                         心跳检测的间隔时间（针对主库检测）
11 repl_password=123                       主从复制的复制用户名称
12 repl_user=repl                           主从复制的复制用户密码
13 ssh_user=root                            节点之间互相通信用户
14
15  节点的标签名必须是server开头，后面的数字表示节点编号
16  [server1]                                第二部分：各个节点的配置信息
17  hostname=10.0.0.51
18  port=3306
19  [server2]
20  hostname=10.0.0.52
21  candidate_master=1                      备选主库
22  port=3306
23  [server3]
24  hostname=10.0.0.53
25  port=3306
```

g.MHA启动前的状态检查（53节点，因为只有53安装了manager）

Bash Copy

```
1  ### 互信检查
2  [root@db03 opt]# masterha_check_ssh --conf=/etc/mha/app1.cnf
3  Sun May 16 19:47:17 2021 - [warning] Global configuration file /etc/masterha_default.cnf not found. Skippi
4  Sun May 16 19:47:17 2021 - [info] Reading application default configuration from /etc/mha/app1.cnf..
5  Sun May 16 19:47:17 2021 - [info] Reading server configuration from /etc/mha/app1.cnf..
6  Sun May 16 19:47:17 2021 - [info] Starting SSH connection tests..
7  Sun May 16 19:47:18 2021 - [debug]
8  Sun May 16 19:47:17 2021 - [debug] Connecting via SSH from root@10.0.0.51(10.0.0.51:22) to root@10.0.0.52
9  Sun May 16 19:47:17 2021 - [debug] ok.
10 Sun May 16 19:47:17 2021 - [debug] Connecting via SSH from root@10.0.0.51(10.0.0.51:22) to root@10.0.0.53
11 Sun May 16 19:47:17 2021 - [debug] ok.
12 Sun May 16 19:47:18 2021 - [debug]
13 Sun May 16 19:47:17 2021 - [debug] Connecting via SSH from root@10.0.0.52(10.0.0.52:22) to root@10.0.0.51
14 Sun May 16 19:47:17 2021 - [debug] ok.
15 Sun May 16 19:47:17 2021 - [debug] Connecting via SSH from root@10.0.0.52(10.0.0.52:22) to root@10.0.0.53
16 Sun May 16 19:47:18 2021 - [debug] ok.
17 Sun May 16 19:47:19 2021 - [debug]
18 Sun May 16 19:47:18 2021 - [debug] Connecting via SSH from root@10.0.0.53(10.0.0.53:22) to root@10.0.0.51
19 Sun May 16 19:47:18 2021 - [debug] ok.
20 Sun May 16 19:47:18 2021 - [debug] Connecting via SSH from root@10.0.0.53(10.0.0.53:22) to root@10.0.0.52
21 Sun May 16 19:47:18 2021 - [debug] ok.
22 Sun May 16 19:47:19 2021 - [info] All SSH connection tests passed successfully.
23
24
25  ### 主从状态检查
26  [root@db03 opt]# masterha_check_repl --conf=/etc/mha/app1.cnf
27  Sun May 16 19:47:37 2021 - [warning] Global configuration file /etc/masterha_default.cnf not found. Skippi
28  Sun May 16 19:47:37 2021 - [info] Reading application default configuration from /etc/mha/app1.cnf..
29  Sun May 16 19:47:37 2021 - [info] Reading server configuration from /etc/mha/app1.cnf..
30  Sun May 16 19:47:37 2021 - [info] MHA::MasterMonitor version 0.58.
31  Sun May 16 19:47:39 2021 - [info] GTID failover mode = 1
32  Sun May 16 19:47:39 2021 - [info] Dead Servers:
33  Sun May 16 19:47:39 2021 - [info] Alive Servers:
34  Sun May 16 19:47:39 2021 - [info] 10.0.0.51(10.0.0.51:3306)
35  Sun May 16 19:47:39 2021 - [info] 10.0.0.52(10.0.0.52:3306)
36  Sun May 16 19:47:39 2021 - [info] 10.0.0.53(10.0.0.53:3306)
37  Sun May 16 19:47:39 2021 - [info] Alive Slaves:
38  Sun May 16 19:47:39 2021 - [info] 10.0.0.52(10.0.0.52:3306) Version=8.0.20 (oldest major version between
39  Sun May 16 19:47:39 2021 - [info] GTID ON
40  Sun May 16 19:47:39 2021 - [info] Replicating from 10.0.0.51(10.0.0.51:3306)
41  Sun May 16 19:47:39 2021 - [info] Primary candidate for the new Master (candidate_master is set)
42  Sun May 16 19:47:39 2021 - [info] 10.0.0.53(10.0.0.53:3306) Version=8.0.20 (oldest major version between
43  Sun May 16 19:47:39 2021 - [info] GTID ON
44  Sun May 16 19:47:39 2021 - [info] Replicating from 10.0.0.51(10.0.0.51:3306)
45  Sun May 16 19:47:39 2021 - [info] Current Alive Master: 10.0.0.51(10.0.0.51:3306)
46  Sun May 16 19:47:39 2021 - [info] Checking slave configurations..
47  Sun May 16 19:47:39 2021 - [info] read_only=1 is not set on slave 10.0.0.52(10.0.0.52:3306).
48  Sun May 16 19:47:39 2021 - [info] read_only=1 is not set on slave 10.0.0.53(10.0.0.53:3306).
49  Sun May 16 19:47:39 2021 - [info] Checking replication filtering settings..
50  Sun May 16 19:47:39 2021 - [info] binlog_do_db= , binlog_ignore_db=
```



```
51 Sun May 16 19:47:39 2021 - [info] Replication filtering check ok.
52 Sun May 16 19:47:39 2021 - [info] GTID (with auto-pos) is supported. Skipping all SSH and Node package che
53 Sun May 16 19:47:39 2021 - [info] Checking SSH publickey authentication settings on the current master..
54 Sun May 16 19:47:39 2021 - [info] HealthCheck: SSH to 10.0.0.51 is reachable.
55 Sun May 16 19:47:39 2021 - [info]
56 10.0.0.51(10.0.0.51:3306) (current master)
57 +--10.0.0.52(10.0.0.52:3306)
58 +--10.0.0.53(10.0.0.53:3306)
59
60 Sun May 16 19:47:39 2021 - [info] Checking replication health on 10.0.0.52..
61 Sun May 16 19:47:39 2021 - [info] ok.
62 Sun May 16 19:47:39 2021 - [info] Checking replication health on 10.0.0.53..
63 Sun May 16 19:47:39 2021 - [info] ok.
64 Sun May 16 19:47:39 2021 - [warning] master_ip_failover_script is not defined.
65 Sun May 16 19:47:39 2021 - [warning] shutdown_script is not defined.
66 Sun May 16 19:47:39 2021 - [info] Got exit code 0 (Not master dead).
67
--
```

h.启动MHA (53节点)

```
1 nohup masterha_manager --conf=/etc/mha/app1.cnf --remove_dead_master_conf --ignore_last_failover < /dev/null>
2                               指定的配置文件                当你主库宕机怎么处理                跳过最后一次切换（不设置这个参数8个小时内不能连续切
```

i.查看MHA状态

```
1 [root@db03 opt]# masterha_check_status --conf=/etc/mha/app1.cnf
2 app1 (pid:3061) is running(0:PING_OK), master:10.0.0.51
```

3.MHA-vip应用透明功能

3.1 简介

```
1 MHA程序中自带vip的功能，通过脚本来实现。
2 也可以使用keepalived实现高可用功能。
3 优点是MHA自带
4 缺点是只能基于局域网使用
```

3.2 应用透明的配置步骤

3.2.1 上传脚本，解压脚本，转移，授权 (db03)

 mha_script.zip (6 kB)

```

1  unzip mha_script.zip
2  [root@db03 opt]# cd mha_script/
3  [root@db03 mha_script]# ll
4  总用量 28
5  -rw-r--r-- 1 root root  581 4月  1 2020 app1.cnf
6  -rw-r--r-- 1 root root 2229 4月  1 2020 master_ip_failover
7  -rw-r--r-- 1 root root 10312 4月  1 2020 master_ip_online_change
8  -rw-r--r-- 1 root root  789 4月  2 2020 mha_check.sh  MH管理脚本(老男孩学生写的)
9  -rw-r--r-- 1 root root 2238 4月  1 2020 send_report
10 [root@db03 mha_script]# cp * /usr/local/bin/
11 [root@db03 mha_script]# chmod +x /usr/local/bin/*
12 [root@db03 mha_script]# ll /usr/local/bin/
13 总用量 28
14 -rwxr-xr-x 1 root root  581 5月 17 19:00 app1.cnf
15 -rwxr-xr-x 1 root root 2229 5月 17 19:00 master_ip_failover
16 -rwxr-xr-x 1 root root 10312 5月 17 19:00 master_ip_online_change
17 -rwxr-xr-x 1 root root  789 5月 17 19:00 mha_check.sh
18 -rwxr-xr-x 1 root root 2238 5月 17 19:00 send_report

```

3.2.2 修改脚本内容

```

1 [root@db03 mha_script]# cd /usr/local/bin/
2 备份一下修改的脚本
3 [root@db03 bin]# cp master_ip_failover master_ip_failover.bak
4 [root@db03 bin]# ll
5 总用量 32
6 -rwxr-xr-x 1 root root  581 5月 17 19:00 app1.cnf
7 -rwxr-xr-x 1 root root 2229 5月 17 19:00 master_ip_failover
8 -rwxr-xr-x 1 root root 2229 5月 17 19:05 master_ip_failover.bak
9 -rwxr-xr-x 1 root root 10312 5月 17 19:00 master_ip_online_change
10 -rwxr-xr-x 1 root root  789 5月 17 19:00 mha_check.sh
11 -rwxr-xr-x 1 root root 2238 5月 17 19:00 send_report
12 [root@db03 bin]# vim master_ip_failover
13 需要修改的脚本内容如下
14 my $vip = '10.0.0.55/24';    vip设置与对外提供服务的是同一网段，vip地址独自使用
15 my $key = '1';
16 my $if='eth0';
17 my $ssh_start_vip = "/sbin/ifconfig $if:$key $vip"; 生成临时地址vip 格式：网卡名:$key:$vip
18 my $ssh_stop_vip = "/sbin/ifconfig $if:$key down"; 关闭临时地址vip 格式：网卡名: $key downw
19 my $ssh_Bcast_arp= "/sbin/arping -I $if -c 3 -A 10.0.0.55"; 及时更新mac地址（局域网中通信使用mac地址）

```

3.2.3 修改MHA的配置文件

```
1 [root@db03 bin]# vim /etc/mha/app1.cnf
2 [server default]
3 master_ip_failover_script=/usr/local/bin/master_ip_failover
```

3.2.4 重启MHA

```
1 1.关闭MHA
2 [root@db03 bin]# masterha_stop --conf=/etc/mha/app1.cnf
3 Stopped app1 successfully.
4 [1]+ 退出 1          nohup masterha_manager --conf=/etc/mha/app1.cnf --remove_dead_master_conf --ignore_
5 2.开启MHA
6 [root@db03 bin]# nohup masterha_manager --conf=/etc/mha/app1.cnf --remove_dead_master_conf --ignore_last_failov
7 [1] 6385
```

3.2.5 查看MHA状态，及主节点是51

```
1 [root@db03 bin]# masterha_check_status --conf=/etc/mha/app1.cnf
2 app1 (pid:6385) is running(0:PING_OK), master:10.0.0.51
```

3.2.5 第一次配置应用透明功能，需要人为在主节点添加vip

```
1 [root@db01 opt]# ifconfig eth0:1 10.0.0.55/24
```

3.3 测试vip功能

3.3.1 查看主节点的vip临时地址

```
[root@db01 ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.51 netmask 255.255.252.0 broadcast 10.0.3.255
    inet6 fe80::bfc4:6341:ed77:5fe7 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:ed:c3:86 txqueuelen 1000 (Ethernet)
    RX packets 36101 bytes 19498059 (18.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 67768 bytes 124495223 (118.7 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.55 netmask 255.255.255.0 broadcast 10.0.0.255
    ether 00:0c:29:ed:c3:86 txqueuelen 1000 (Ethernet)

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.1.51 netmask 255.255.252.0 broadcast 172.16.3.255
    inet6 fe80::6be1:43ad:46d1:8b29 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:ed:c3:90 txqueuelen 1000 (Ethernet)
    RX packets 99 bytes 7474 (7.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 136 bytes 11758 (11.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 194 bytes 34229 (33.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 194 bytes 34229 (33.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

3.3.2 模拟主节点（51）宕机

```
1 [root@db01 ~]# /etc/init.d/mysqld stop
```

3.3.3 再次查看主节点（51）ip,已经没有

```
[root@db01 ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.51 netmask 255.255.252.0 broadcast 10.0.3.255
    inet6 fe80::bfcd:6341:ed77:5fe7 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:ed:c3:86 txqueuelen 1000 (Ethernet)
    RX packets 36556 bytes 19539330 (18.6 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 68048 bytes 124534948 (118.7 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.1.51 netmask 255.255.252.0 broadcast 172.16.3.255
    inet6 fe80::6be1:43ad:46d1:8b29 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:ed:c3:90 txqueuelen 1000 (Ethernet)
    RX packets 99 bytes 7474 (7.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 136 bytes 11758 (11.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 194 bytes 34229 (33.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 194 bytes 34229 (33.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

3.3.4 51的vip地址漂移到新主节点52上

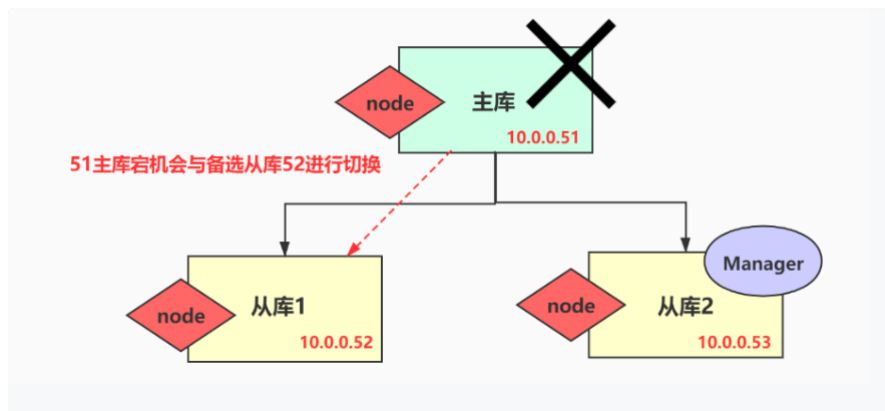
```
[root@db02 opt]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.52 netmask 255.255.252.0 broadcast 10.0.3.255
    inet6 fe80::872:7243:d6d9:4c1a prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:41:78:ee txqueuelen 1000 (Ethernet)
    RX packets 15042 bytes 17690990 (16.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7353 bytes 835276 (815.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.55 netmask 255.255.255.0 broadcast 10.0.0.255
    ether 00:0c:29:41:78:ee txqueuelen 1000 (Ethernet)

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.1.52 netmask 255.255.252.0 broadcast 172.16.3.255
    inet6 fe80::d98d:d093:8012:f6cd prefixlen 64 scopeid 0x20<link>
    inet6 fe80::6be1:43ad:46d1:8b29 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:41:78:f8 txqueuelen 1000 (Ethernet)
    RX packets 18 bytes 1392 (1.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 60 bytes 4782 (4.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 41 bytes 8166 (7.9 KiB)
```

4.MHA故障测试和修复



4.1 查看各个节点状态

Bash | Copy

```
1 51节点宕机
2 52节点查看, 此时52为主, 53为从
3 db02 [(none)]>show slave hosts;
4 +-----+-----+-----+-----+-----+
5 | Server_id | Host | Port | Master_id | Slave_UUID |
6 +-----+-----+-----+-----+-----+
7 | 53 | | 3306 | 52 | a4910121-b634-11eb-9153-000c29b4ef39 |
8 +-----+-----+-----+-----+-----+
9 53节点查看线程, 51主库切换到了52
10 db03 [(none)]>show slave status\G;
11 Master_Host: 10.0.0.52
```

4.2 修复为一主多从结构

4.2.1 修复51坏掉的旧主节点

Bash | Copy

```
1 [root@db01 ~]# /etc/init.d/mysqld start
```

4.2.2 重新构建主从 (51上操作)

Bash | Copy

```
1 因为此时主库为52
2 所以51--指定->52
3 [root@db01 ~]# mysql -e \
4 "CHANGE MASTER TO \
5     MASTER_HOST='10.0.0.52', \
6     MASTER_USER='repl', \
7     MASTER_PASSWORD='123', \
8     MASTER_PORT=3306, \
9     MASTER_AUTO_POSITION=1;"
10 [root@db01 ~]# mysql -e "start slave;"
11 [root@db01 ~]# mysql -e "show slave status \G"|grep "Running:"
12         Slave_IO_Running: Yes
13         Slave_SQL_Running: Yes
```

4.2.3 修改MHA配置文件(53上操作)

```
▼ Bash Copy
1 [root@db03 bin]# vim /etc/mha/app1.cnf
2 [server1]
3 hostname=10.0.0.51
4 port=3306
5 candidate_master=1
6
7 [server2]
8 hostname=10.0.0.52
9 port=3306
10
11 [server3]
12 hostname=10.0.0.53
13 port=3306
```

4.2.4 检查MHA状态

```
▼ Bash Copy
1 [root@db03 bin]# masterha_check_repl --conf=/etc/mha/app1.cnf
2 MySQL Replication Health is OK.
```

4.2.5 开启MHA

```
▼ Bash Copy
1 [root@db03 bin]# nohup masterha_manager --conf=/etc/mha/app1.cnf --remove_dead_master_conf --ignore_last_failov
2 [1] 7338
3 [root@db03 bin]# masterha_check_status --conf=/etc/mha/app1.cnf
4 app1 (pid:7338) is running(0:PING_OK), master:10.0.0.52
5
```

5.MHA-故障提醒功能

5.1 配置故障提醒功能

5.1.1 修改故障提醒脚本

Bash | Copy

```
1 1.备份脚本
2 [root@db03 bin]# cp send_report send_report.bak
3 2.修改脚本内容
4 [root@db03 bin]# vim send_report
5 my $smtp='smtp.qq.com';           #smtp服务器
6 my $mail_from='476764695@qq.com'; #发件者的邮箱
7 my $mail_user='476764695';        #发件者的用户名
8 my $mail_pass='ydfgzxwzowqcjdh'; #发件者邮箱服务器的授权码
9 my $mail_to='1083637963@qq.com';  #收件者的邮箱
10 发送给多个人方式
11 #my $mail_to=['to1@qq.com','to2@qq.com'];
```

5.1.2 修改MHA配置文件

Bash | Copy

```
1 vim /etc/mha/app1.cnf
2
3 [server default]
4 report_script=/usr/local/bin/send_report
```

5.1.3 重启MHA

Bash | Copy

```
1 1.关闭MHA
2 [root@db03 bin]# masterha_stop --conf=/etc/mha/app1.cnf
3 2.开启MHA
4 [root@db03 bin]# nohup masterha_manager --conf=/etc/mha/app1.cnf --remove_dead_master_conf --ignore_last_failov
```

5.2 测试故障提醒

5.2.1 模拟主节点故障

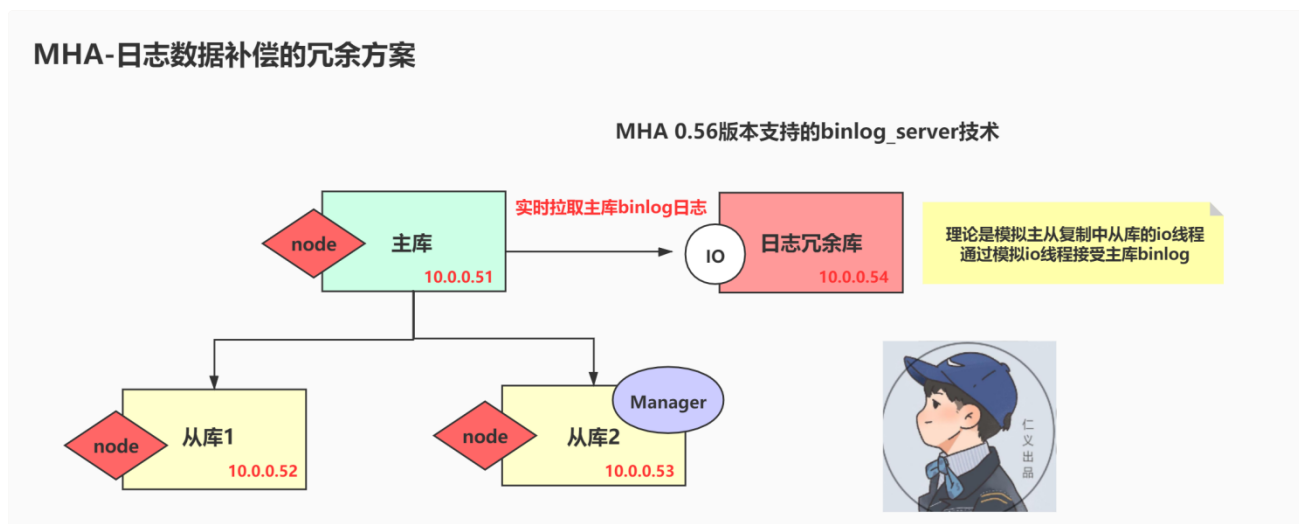
Bash | Copy

```
1 [root@db02 opt]# /etc/init.d/mysqld stop
```

5.2.2 邮箱收到故障提醒



6.MHA-日志数据补偿的冗余方案



6.1 选取一台机器 (binlog_server)

只要机器上具有数据库软件即可

```
CentOS Linux 7 (Core)
Kernel 3.10.0-1127.el7.x86_64 on an x86_64

binlog_server login:
```

6.2 在binlog_server节点上配置

6.2.1 创建binlog_server接受主库binlog的存放目录

一定不能与数据库本身存放binlog的目录重复

▼

Bash | Copy

```
1 mkdir -p /data/binlog_server/
2 chown -R mysql:mysql /data/*
```

6.2.2 binlog_server拉取日志

▼

Bash | Copy

```
1 切换到接受日志路径下
2 cd /data/binlog_server/
3 mysqlbinlog -R --host=10.0.0.51 --user=mha --password=mha --raw --stop-never mysql-bin.
4 远程remote 拉取主库的ip 只要主库不宕机，一直拉取 拉取日志的起点 放
5
6 1.拉取主库节点
7 [root@db03 bin]# masterha_check_status --conf=/etc/mha/app1.cnf
8 app1 (pid:8499) is running(0:PING_OK), master:10.0.0.51
9
10 2.拉取日志的起点
11 需要按照目前从库已经获取到的二进制点为起点
12 [root@db03 bin]# mysql -e "show slave status \G"|grep "Master_Log"
13 Master_Log_File: mysql-bin.000005
14 Read_Master_Log_Pos: 196
15 Relay_Master_Log_File: mysql-bin.000005
16 Exec_Master_Log_Pos: 196
```

6.2.3 修改MHA配置文件（53节点）

```
▼ Bash Copy
1 vim /etc/mha/app1.cnf
2 [binlog1]
3 no_master=1          因为这个节点只负责io接受主库binlog日志，不会回放日志。所以我们把这个节点加入到bad数组里，不允许发生故障时
4 hostname=10.0.0.54    (binlog_server节点)
5 master_binlog_dir=/data/binlog_server/ 接受主库binlog存放的路径
```

6.2.4 重启MHA（53节点）

```
▼ Bash Copy
1 [root@db03 bin]# masterha_stop --conf=/etc/mha/app1.cnf
2 [root@db03 bin]# nohup masterha_manager --conf=/etc/mha/app1.cnf --remove_dead_master_conf --ignore_last_failover &
```

6.3 发生故障binlog_server的修复

6.3.1 模拟主节点发生故障

```
▼ Bash Copy
1 1.先查看现在MHA架构中的主节点是谁？
2 [root@db03 bin]# masterha_check_status --conf=/etc/mha/app1.cnf
3 app1 (pid:9056) is running(0:PING_OK), master:10.0.0.51
4
5 2.模拟主节点故障
6 [root@db01 ~]# /etc/init.d/mysqld stop
7
8 3.故障后出现的现象
9 3.1 主从关系会发生变化，主节点变成52
   3.2 vip地址会飘逸到52
   3.3 会收到故障通知
```

6.3.2 故障修复–构建旧节点的主从结构

Bash | Copy

```
1 [root@db01 ~]# /etc/init.d/mysqld start
2
3 [root@db01 ~]# mysql -e \
4 "CHANGE MASTER TO \
5     MASTER_HOST='10.0.0.52',\
6     MASTER_USER='repl', \
7     MASTER_PASSWORD='123', \
8     MASTER_PORT=3306, \
9     MASTER_AUTO_POSITION=1;"
10
11 [root@db01 ~]# mysql -e "start slave;"
12
13 [root@db01 ~]# mysql -e "show slave status \G"|grep "Running:"
14     Slave_IO_Running: Yes
15     Slave_SQL_Running: Yes
```

6.3.3 故障修复-修复binlog_server

Bash | Copy

```
1 1.将之前接受主库日志路径下的文件进行删除或者备份
2 2.binlog_server节点重新设置接受
3 2.1 查看新主库日志位置点
4 [root@db03 ~]# mysql -e "show slave status \G"|grep "Master_Log"
5     Master_Log_File: mysql-bin.000003
6     Read_Master_Log_Pos: 196
7     Relay_Master_Log_File: mysql-bin.000003
8     Exec_Master_Log_Pos: 196
9
10 2.2 拉取主库节点是52
```

拉取日志

```
mysqlbinlog -R --host=10.0.0.52 --user=mha --password=mha --raw --stop-never mysql-bin.000003 &
```

6.3.4 故障修复-MHA

```
▼ Bash | Copy
1 [root@db03 bin]# vim /etc/mha/app1.cnf
2 [server1]
3 hostname=10.0.0.51
4 port=3306
5 candidate_master=1
6
7 [server2]
8 hostname=10.0.0.52
9 port=3306
10
11 [server3]
12 hostname=10.0.0.53
13 port=3306
14
15
16 检查MHA
17 [root@db03 ~]# masterha_check_repl --conf=/etc/mha/app1.cnf
18 MySQL Replication Health is OK.
19
20 重启MHA
21 1.关闭MHA
22 [root@db03 bin]# masterha_stop --conf=/etc/mha/app1.cnf
23 2.开启MHA
24 [root@db03 bin]# nohup masterha_manager --conf=/etc/mha/app1.cnf --remove_dead_master_conf --ignore_last_fail
25
26 查看MHA
27 [root@db03 ~]# masterha_check_status --conf=/etc/mha/app1.cnf
app1 (pid:10056) is running(0:PING_OK), master:10.0.0.52
```

7.MHA维护操作-手工在线切换

7.1 第一种方法 直接使用materha_master_switch命令（生成环境不推荐）

```

1 masterha_master_switch --conf=/etc/mha/app1.cnf --master_state=alive --new_master_host=10.0.0.51 --orig_master_host=10.0.0.51
2                                     指定MHA配置文件           在线切换           指定切换的主库           把原理切换的主库作
3
4
5 使用步骤
6
7 1. 首先需要先关闭MHA架构
8
9 2. 使用命令
10
11 3. 在主库当中执行FLUSH NO_WRITE_TO_BINLOG TABLES ,禁止binlog刷新, 防止出现同步导致主从数据不一致
12
13 4. 等待主从binglog一致 ----yes
14
15 5. 主从切换---yes
16
17 6.master_ip_online_change_script is not defined. If you do not disable writes on the current master manually,
18 applications keep writing on the current master. Is it ok to proceed? (yes/NO): yes
19
20 6.1 手动设置禁止主库写入操作, 防止主从不一致问题
21
22 6.2 手工切换vip到新的主库上, (vip在设置时是遇到故障会飘逸)

```

7.2 第二种方法 调用master_ip_online_change_script脚本，再使用materha_master_switch命令（生成环境中的正确操作）

7.2.1 准备切换脚本

脚本拥有两个功能，自动切换vip,自动锁定源主库

```

1 [root@db03 ~]# cd /usr/local/bin/
2 [root@db03 bin]# cp master_ip_online_change master_ip_online_change.bak
3 [root@db03 bin]# vim master_ip_online_change
4
5 my $vip = "10.0.0.55";
6 my $key = "1";
7 my $if = "eth0";
8
9 my $ssh_start_vip = "/sbin/ifconfig $if:$key $vip";
10 my $ssh_stop_vip = "/sbin/ifconfig $if:$key $vip down";
11 my $ssh_Bcast_arp = "/sbin/arping -I $if -c 3 -A 10.0.0.55";

```

7.2.2 修改MHA配置文件

```

1 vim /etc/mha/app1.cnf
2 [server default]
3 master_ip_online_change_script=/usr/local/bin/master_ip_online_change

```

7.2.3 停止MHA （可以省略）

▼

Bash | Copy

```
1 [root@db03 bin]# masterha_stop --conf=/etc/mha/app1.cnf
```

7.2.4 检查MHA-REPL (可以省略)

▼

Bash | Copy

```
1 [root@db03 bin]# masterha_check_repl --conf=/etc/mha/app1.cnf
```

7.2.5 在线切换

Bash | Copy

```
1 [root@db03 bin]# masterha_master_switch --conf=/etc/mha/app1.cnf --master_state=alive --new_master_host=
2 Tue May 18 19:06:46 2021 - [info] MHA::MasterRotate version 0.58.
3
4 Tue May 18 19:06:46 2021 - [info] Starting online master switch..
5
6 Tue May 18 19:06:46 2021 - [info]
7 Tue May 18 19:06:46 2021 - [info] * Phase 1: Configuration Check Phase..
8
9 Tue May 18 19:06:46 2021 - [info]
10 Tue May 18 19:06:46 2021 - [warning] Global configuration file /etc/masterha_default.cnf not found. Skipp
11 Tue May 18 19:06:46 2021 - [info] Reading application default configuration from /etc/mha/app1.cnf..
12
13 Tue May 18 19:06:46 2021 - [info] Reading server configuration from /etc/mha/app1.cnf..
14
15 Tue May 18 19:06:48 2021 - [info] GTID failover mode = 1
16
17 Tue May 18 19:06:48 2021 - [info] Current Alive Master: 10.0.0.52(10.0.0.52:3306)
18
19 Tue May 18 19:06:48 2021 - [info] Alive Slaves:
20
21 Tue May 18 19:06:48 2021 - [info] 10.0.0.51(10.0.0.51:3306) Version=8.0.20 (oldest major version betwe
22 Tue May 18 19:06:48 2021 - [info] GTID ON
23 Tue May 18 19:06:48 2021 - [info] Replicating from 10.0.0.52(10.0.0.52:3306)
24
25 Tue May 18 19:06:48 2021 - [info] Primary candidate for the new Master (candidate_master is set)
26
27 Tue May 18 19:06:48 2021 - [info] 10.0.0.53(10.0.0.53:3306) Version=8.0.20 (oldest major version betwe
28
29 Tue May 18 19:06:48 2021 - [info] GTID ON
30
31 Tue May 18 19:06:48 2021 - [info] Replicating from 10.0.0.52(10.0.0.52:3306)
32
33 # It is better to execute FLUSH NO_WRITE_TO_BINLOG TABLES on the master before switching. Is it ok to exe
34 # 原主库执行 FLUSH NO_WRITE_TO_BINLOG TABLES
35
36 Tue May 18 19:10:02 2021 - [info] Executing FLUSH NO_WRITE_TO_BINLOG TABLES. This may take long time..
37
38 Tue May 18 19:10:02 2021 - [info] ok.
39
40 Tue May 18 19:10:02 2021 - [info] Checking MHA is not monitoring or doing failover..
41
42 Tue May 18 19:10:02 2021 - [info] Checking replication health on 10.0.0.51..
43
44 Tue May 18 19:10:02 2021 - [info] ok.
45
46 Tue May 18 19:10:02 2021 - [info] Checking replication health on 10.0.0.53..
47
48 Tue May 18 19:10:02 2021 - [info] ok.
49
50 Tue May 18 19:10:02 2021 - [info] 10.0.0.51 can be new master.
51
52 Tue May 18 19:10:02 2021 - [info]
53
54 From:
55 10.0.0.52(10.0.0.52:3306) (current master)
56 +--10.0.0.51(10.0.0.51:3306)
57 +--10.0.0.53(10.0.0.53:3306)
```



7.2.6 里构Diniog_server

Bash | Copy

```
1 [root@binlog_server binlog_server]# ps -ef |grep mysqlbinlog
2 root      2664   2525   0 17:35 pts/0    00:00:00 mysqlbinlog -R --host=10.0.0.52 --user=mha --password=x x --i
3
4 root      2902   2525   0 19:13 pts/0    00:00:00 grep --color=auto mysqlbinlog
5
6 [root@binlog_server binlog_server]# kill -9 2664
7
8 [root@binlog_server binlog_server]# cd /data/binlog_server/
9 [root@binlog_server binlog_server]# rm -rf *
10 [root@binlog_server binlog_server]# mysqlbinlog -R --host=10.0.0.51 --user=mha --password=mha --raw --stop-ne
```

7.2.7 启动MHA

Bash | Copy

```
1 [root@db03 bin]# nohup masterha_manager --conf=/etc/mha/app1.cnf --remove_dead_master_conf --ignore_last_failo
2
3 [1] 13019
4 [root@db03 bin]# masterha_check_status --conf=/etc/mha/app1.cnf
5
6 appl (pid:13019) is running(0:PING_OK), master:10.0.0.51
```

8.MHA–故障修复总结

Bash | Copy

```
1 a. 检查修复故障节点(硬件 OS 网络).
2
3 b. 检查修复数据库实例
4
5 c. 检查和修复1主2从.
6 db01 <----> db02
7 stop slave ;
8 reset slave;
9
10 db01 ----> db02
11
12 d. 检查和修复配置文件
13 e. 检查和修复VIP
14 f. 检查和修复binlog server
15 g. 使用检查脚本最后确认
16 h. 启动MHA
```

MHA%E2%88%9A%20%7C%201.MHA%E9%AB%98%E5%8F%AF%E7%94%A8%E7%AE%80%E4%BB%8B1.1%20%E4%BB%80%E4%B9%88%E6%98%AF%E9%AB%98%E5%8F%AF%E7%94%A8%3F%E4%B