====================

Mysql代理软件Amoeba

====================

Amoeba的作用是在于Mysql之上，用于代理应用与Mysql的连接，可以理解为中间件，他的作用是将连接细化，如：将读写进行分离，分别指向不同的数据库，也可用根据不同的数据进行指向不同的数据库，还可以做读或写的负载均衡

mysql 代理层有mysqlproxy, amoeba, cobar等

Mysqlproxy：数据量很小的时候用的还不错，可数据量一但大起来，这个软件就会暴露出很多BUG问题，Mysql官方给出的建议是适应轻量级的应用

Amoeba：适用中大型数据量的应用负载，可做读写分离，数据切分，负载均衡等技术，是目前企业中用的最广的数据库代理

Cobar：Cobar的分布式主要是通过将表放入不同的库来实现。是阿里巴巴开发的软件，本地的软件和Amoeba一样

Amoeba代理Mysql读写分离，架构图：

客户

｜

｜

web应用程序，游戏程序（c,php,java.......)客户端

|

|

代理层 (mysqlproxy,amoeba) 读写分离/数据切分

|

|

mysql主 <----> mysql从

准备4台服务器

192.168.224.10 --客户端

192.168.224.11 --amoeba

192.168.224.12 --mysql主

192.168.224.13 --mysql从

-u 指定用户名

-p 指定用户密码

-h 指定服务器IP

-S 指定socket文件路径

-P 指定mysql服务器的端口号

-e

架构前所有节点准备：

1，主机名三步，互相绑定

2，时间同步

3，关闭iptables,selinux

4, 配置好yum

5, 静态ip地址

6，yum install -y mariadb mariadb-server 安装这两个包就可以登陆mariadb

第一步：安装amoeba服务器

因为是java开发的，需要java运行环境，需要先安装JDK

1、安装jdk1.8版本，tar包源码包，解压即能用。

# wget http://192.168.2.254/mysql/pack/amoeba/jdk-8u181-linux-x64.tar.gz

解压jdk包

# tar xf jdk-8u45-linux-x64.tar.gz -C /usr/local/

cd到解压目录

cd /usr/local

把刚解压的JDK改名为java

mv ./jdk1.8.0\_45 ./java

完整的目录为：/usr/local/java

验证一下刚解压安装的的jdk版本

# /usr/local/java/bin/java -version

java version "1.8.0\_45"

Java(TM) SE Runtime Environment (build 1.8.0\_45-b14)

Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)

到此为止JDK已经好了，下面amoeba要用到这个jdk目录

2、安装amoeba软件

下载amoeba

# wget http://192.168.2.254/mysql/pack/amoeba/amoeba-mysql-binary-2.2.0.tar.gz

a、新建一个amoeba目录，用于存放amoeba解压指定用

# mkdir /usr/local/amoeba

b、解压到上一步新建的目录里

# tar xf amoeba-mysql-binary-2.2.0.tar.gz -C /usr/local/amoeba

c、使用ls查看解压的目录

# ls /usr/local/amoeba/ －－amoeba的所有文件在此目录下

benchmark changelogs.txt lib README.html

bin conf LICENSE.txt

关注下面二个目录

bin 是启动脚本目录

conf 是配置文件目录

配置文件目录里我们要关注有三个文件:

amoeba.xml --配置amoeba的全局配置文件

dbServers.xml --配置amoeba连接mysql数据库的文件

rule.xml --配置我们数据切分的文件

d、开始配置amoeba连接mysql数据库

# cd /usr/local/amoeba --进到amoeba目录

先修改./conf/amoeba.xml 文件

vi /usr/local/amoeba/conf/amoeba.xml

11 <property name="port">8066</property> --修改端口(可以改可以不改)

30 <property name="user">amoeba</property> --客户端访问amoeba时的用户名

32 <property name="password">123</property> --客户端访问amoeba时的密码

117 <!-- --删除该行，去掉注释

118 <property name="writePool">server1</property> --这里指定写走server1

119 <property name="readPool">server2</property> --把server1改成server2,指定读

120 --> --删除该行，去掉注释

再修改 ./conf/dbServers.xml

vim /usr/local/amoeba/conf/dbServers.xml

20 <property name="port">3306</property> --修改端口，默认是3306，可以不修改

23 <property name="schema">aaa</property> --指定访问数据库

26 <property name="user">mysql</property> --登陆数据库的用户名

29 <property name="password">Mysql138#$</property> --设置登陆密码 把注释打开，或者复制这行

45 <dbServer name="server1" parent="abstractServer"> --服务器1，mysql主服务器，45-50行是一段

46 <factoryConfig>

47 <!-- mysql ip -->

48 <property name="ipAddress">192.168.224.12</property> --mysql主服务器IP

49 </factoryConfig>

50 </dbServer>

51

52 <dbServer name="server2" parent="abstractServer"> --服务器2,mysql从服务器，52-57行是一段

53 <factoryConfig>

54 <!-- mysql ip -->

55 <property name="ipAddress">192.168.224.13</property> --mysql从服务器IP

56 </factoryConfig>

57 </dbServer>

如果要再定义一台从服务器，可以再这里增加内容，（45-50行）6yy复制6行 p粘贴，把name="server2"改成"server3"再修改对应IP就可以了



3、修改amoeba启动文件

启动文件/usr/local/amoeba/bin/amoeba 是需要jdk的支持才能启动，下面我们配置启动文件指定访问jdk

在第1步时我们解压了jdk，这里要配置amoeba访问jdk

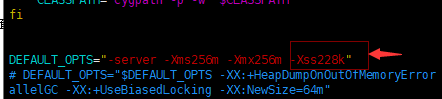
vim /usr/local/amoeba/bin/amoeba --在10行下面，加上以下代码

11 JAVA\_HOME=/usr/local/java

12 PATH=$PATH:$JAVA\_HOME/bin

13 export JAVA\_HOME PATH

62 DEFAULT\_OPTS="-server -Xms256m -Xmx256m -Xss228k" --在62行把 Xss128k 改成228k，因启动时要求最低内存是228k



注：-Xms256m --分配256m物理内存给amoeba软件用，连接数据库时初始化内存就要256m

-Xmx256m --这个是amoeba软件最大可用的物理内存，（32位的JDK最大只能是2G，64位的JDK无限制但不能大于本机的物理内存大小）

-Xss128k --默认是128k，但amoeba软件要求是228k，这个启动amoeba软件就要228k的内存

再使用nohup方法启动amoeba服务

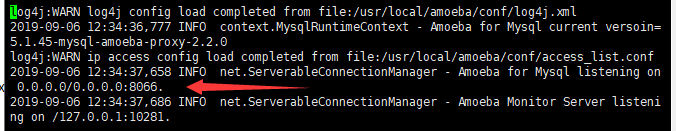
nohup ./bin/amoeba start & --这个启动方法把启动的信息写进nohup.out文件里，并在后台运行。建议用这种方法，方便我们排错。

绝对路径： nohup /usr/local/amoeba/bin/amoeba start &

启动的另外二个方法：建议不使用

./bin/amoeba start & --这个启动的方法是把启动的信息打印在终端窗口。

./bin/amoeba start >/dev/null & --把启动的信息输出到/dev/null 空洞里面。



第二大步:在mysql主从服务器上授权并验证

为了方便验证，把两台mysql复制关闭，然后两台mysql都新建下面的测试表

下面的步骤，两台数据库都要操作执行。

mysql> create database aaa;

Query OK, 1 row affected (0.00 sec)

mysql> use aaa;

Database changed

mysql> create table emp (id int,ename varchar(20));

Query OK, 0 rows affected (0.01 sec)

mysql> insert into emp values(1,"master"); --主服务器插入这一条

mysql> insert into emp values(1,"slave"); --从服务器插入这一条

--并对mysql进行授权，授权的IP为amoeba的IP

mysql> grant all on aaa.\* to 'mysql'@'%' identified by 'Mysql138#$';

Query OK, 0 rows affected (0.04 sec)



8.0版本需要先创建用户，不能使用简单密码

mysql> create user 'mysql'@'%' identified with mysql\_native\_password by 'Mysql138#$';

Query OK, 0 rows affected (0.00 sec)

再授权

mysql> grant all on aaa.\* to 'mysql'@'%';

Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;

Query OK, 0 rows affected (0.00 sec)

授权后,可以去amoeba(192.168.224.11)这台去连接一下这两个mysql,需要能连才行，如果不能连则把上面再做一遍

在amoeba服务器测试连接两台mysql

mysql -umysql -h192.168.224.12 -P3306 -pMysql138#$

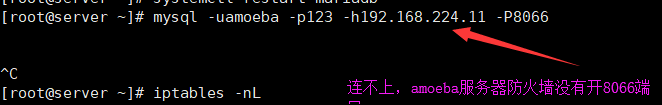
mysql -umysql -h192.168.224.13 -P3306 -pMysql138#$

如果防火墙清空后还是连不上192.168.224.13的话， 可以把13这台mysql重启

第三大步：在客户端进行测试

客户端使用mysql命令连接 amoeba服务器登陆测试

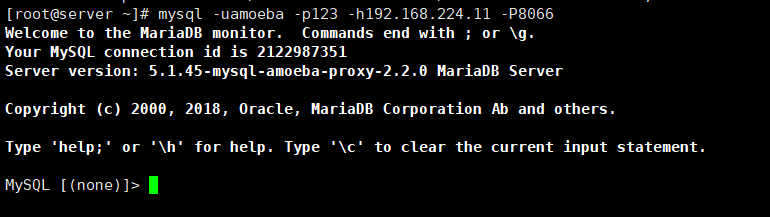
mysql -uamoeba -p123 -h192.168.224.11 -P8066



切换到amoeba服务器，开放对应端口



然后就可以登录了



mysql> use aaa;

Database changed

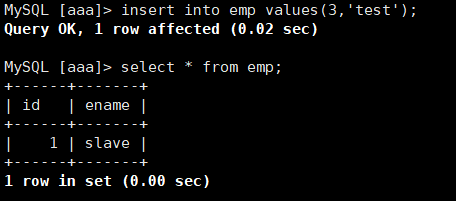
mysql>

mysql>

mysql> insert into emp values(3,'test'); --这里往数据库插入了一条数据，插入的动作是指定到了mysql主

Query OK, 1 row affected (0.01 sec)

mysql> select \* from emp; --这里再查时，看不到刚插入的数据，是因为读我们指定到了mysql从



=======================================================================

================

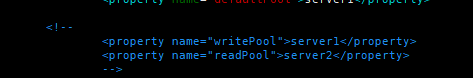
Mysql 数据切分

================

是按照不同的数据访问不同的数据库

第一步：搭建前，先关闭读写分离和主从复制

# vim /usr/local/amoeba/conf/amoeba.xml --在117行前面加上<!--，在119行下面加上--> 注释到117、118行，读写分离成功关闭。



116 <!--

117 <property name="writePool">server1</property>

118 <property name="readPool">server2</property>

119 -->

第二步：配置数据切片

--下面我是按aaa.emp表里的emp\_id这一列来切分，当它为奇数就给server2,为偶数就给server1

# vim /usr/local/amoeba/conf/rule.xml --修改第三个配置文件，定义数据切分的规则

<?xml version="1.0" encoding="gbk"?>

<!DOCTYPE amoeba:rule SYSTEM "rule.dtd">

<amoeba:rule xmlns:amoeba="http://amoeba.meidusa.com/">

<tableRule name="emp" schema="aaa" defaultPools="server1,server2"> --name代表表名称，schema代表数据库名，defaultPools代表可支持的服务器

<rule name="rule1" ruleResult="POOLNAME"> --name 代表切片动作的名称

<parameters>id</parameters> --代表aaa.emp表里哪个字段要做切片的判断

<expression><![CDATA[ --切片开始

var division = id % 2; --定义division变量，值为 id % 2 ，也就是取余

case division when 1 then 'server1'; --变量值不能被2整除的转到server1

when 0 then 'server2'; --变量值能被2整除的转到server2

end case;

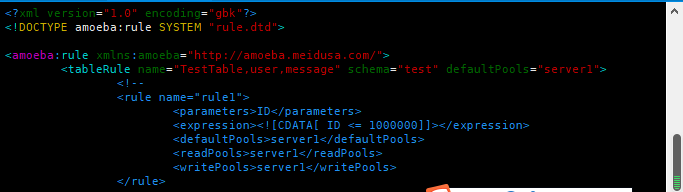
]]> --切片结束

</expression>

</rule>

</tableRule>

</amoeba:rule>



===========================================

<?xml version="1.0" encoding="gbk"?>

<!DOCTYPE amoeba:rule SYSTEM "rule.dtd">

<amoeba:rule xmlns:amoeba="http://amoeba.meidusa.com/">

<tableRule name="emp" schema="aaa" defaultPools="server1,server2">

<rule name="rule1" ruleResult="POOLNAME">

<parameters>id</parameters>

<expression><![CDATA[

var division = id % 2;

case division when 1 then 'server1';

when 0 then 'server2';

end case;

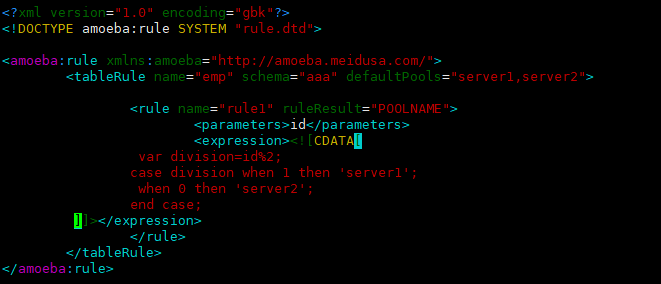
]]>

</expression>

</rule>

</tableRule>

</amoeba:rule>



--直接复制粘贴上去，格式空格之类的太多；可以先在vim里:set paste再insert模式粘贴上去就可以了

备：复制到rule.xml文件后，要把后面的中文注释删除，使用命令d$

第三步:重启amoeba

如果配置文件出错会启动不了：

绝对路径： /usr/local/amoeba/bin/amoeba stop

绝对路径： nohup /usr/local/amoeba/bin/amoeba start &

# cd /usr/local/amoeba --进入到amoeba安装目录

# ./bin/amoeba stop --停止服务

# nohup ./bin/amoeba start & --启动服务

第四步:客户端测试

在客户端连接进行测试，连接的IP为amoeba的IP，端口为8066

# mysql -h192.168.224.11 -uamoeba -p123 -P8066

注意:amoeba是根据sql解析来进行数据切分的，所以需要把切分的关键字段（这里是emp表的id字段）加入到sql中.否则切分规则无效。无效后,会在 server1,server2 均都插入数据。

mysql> insert into aaa.emp values (1,'aaa'); --这样做是错误的，会在两个后台数据库里都插入数据

--正确的插入方法

mysql> insert into aaa.emp(id,ename) values (1,'aaa');

mysql> insert into aaa.emp(id,ename) values (2,'bbb');

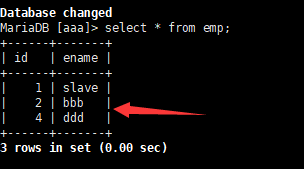
mysql> insert into aaa.emp(id,ename) values (3,'ccc');

mysql> insert into aaa.emp(id,ename) values (4,'ddd');

然后去server1(192.168.224.12)上查看，只有1,3两条数据

去server2(192.168.224.13)上查看，只有2,4两条数据

数据切分成功



delete和update操作也是同理

---------------------------------------------------------------

--按照上面的配置文件，只修改rule.xml规则文件如下，做成以emp\_id值的范围来进行水平切分

--下面实现的是id小于等于100的任何操作都在server1,大于100的任何操作都是在server2

<?xml version="1.0" encoding="gbk"?>

<!DOCTYPE amoeba:rule SYSTEM "rule.dtd">

<amoeba:rule xmlns:amoeba="http://amoeba.meidusa.com/">

<tableRule name="emp" schema="aaa" defaultPools="server1,server2">

<rule name="rule1"> # 注意: (ruleResult="POOLNAME") 这句不要添加， 否则客户端插入数据会报错

<parameters>id</parameters>

<expression><![CDATA[ id <= 100]]></expression>

<defaultPools>server1</defaultPools>

<readPools>server1</readPools>

<writePools>server1</writePools>

</rule>

<rule name="rule2">

<parameters>id</parameters>

<expression><![CDATA[ id > 100]]></expression>

<defaultPools>server2</defaultPools>

<readPools>server2</readPools>

<writePools>server2</writePools>

</rule>

</tableRule>

</amoeba:rule>

--直接复制粘贴上去，格式空格之类的太多；可以先在vim里:set paste再insert模式粘贴上去就可以了

重启amoeba

绝对路径： /usr/local/amoeba/bin/amoeba stop

绝对路径： nohup /usr/local/amoeba/bin/amoeba start &

======================================================================

===========================

Amoeba数据切分 之 垂直切分:

===========================

架构图

client 192.168.224.10

|

|

amoeba 192.168.224.11

|

|

|------------------｜

mysqlA mysqlB

192.168.224.12 192.168.224.13

aaa.aaa表 aaa.bbb表

第一步：准备测试环境

1、先把两个mysql（不做AB复制的）以前的数据删掉，新建两个表来做测试

192.168.224.12上

mysql> create database aaa;

mysql> use aaa;

mysql> create table aaa.aaa (id int);

192.168.224.13上

mysql> create database aaa;

mysql> use aaa;

mysql> create table aaa.bbb (id int);

第二步：修改rule.xml配置文件

# vim /usr/local/amoeba/conf/rule.xml --清空文件内容，加上下面一段

<?xml version="1.0" encoding="gbk"?>

<!DOCTYPE amoeba:rule SYSTEM "rule.dtd">

<amoeba:rule xmlns:amoeba="http://amoeba.meidusa.com/">

<tableRule name="aaa" schema="aaa" defaultPools="server1"/>

<tableRule name="bbb" schema="aaa" defaultPools="server2"/>

</amoeba:rule>

第三步:重启amoeba

# cd /usr/local/amoeba --进入到amoeba安装目录

# ./bin/amoeba stop --停止服务

# nohup ./bin/amoeba start & --启动服务

绝对路径： /usr/local/amoeba/bin/amoeba stop



报这种 可以手动把amoeba进程kill 掉，然后再启动

绝对路径： nohup /usr/local/amoeba/bin/amoeba start &

第四步：在客户端测试

# mysql -uamoeba -p123 -h192.168.224.11 -P8066

mysql> use aaa;

mysql> insert into aaa values (1); --这一条被插入到192.168.224.12上的aaa.aaa表

mysql> insert into bbb values (2); --这一条被插入到192.168.224.13上的aaa.bbb表

mysql> select \* from aaa.aaa;

+------+

| id |

+------+

| 1 |

+------+

1 row in set (0.01 sec)

mysql>

mysql> select \* from aaa.bbb;

+------+

| id |

+------+

| 2 |

+------+

1 row in set (0.03 sec)

mysql>

======================================================================

Amoeba集群 之 读集群

集群的好处是，可以让服务器平均负载我们连接请求

Amoeba集群分发算法有三种：

1=ROUNDROBIN , 2=WEIGHTBASED , 3=HA

ROUNDROBIN --代表轮询，1台访问一次

WEIGHTBASED --代表加权轮询，是以哪一台性能好就承载多一点连接

HA --代表高可用

把读写分离改成读集群的条件是至少2台服务器做读，

第一步：配置dbServers.xml文件:

dbServer.xml文件里的 59行~67行 是设置集群的段

我们关注的行有59，62，65行

59行的 name="multiPool"，这个是设置服务器集群名称，在amoeba.xml文件里设备读服务器时要用到

62行是设置集群的算法，默认是1轮询

65行是设置服务器列表。

以下是配置代码：

vim /usr/local/amoeba/conf/dbServers.xml

59 <dbServer name="multiPool" virtual="true">

60 <poolConfig class="com.meidusa.amoeba.server.MultipleServerPool">

61 <!-- Load balancing strategy: 1=ROUNDROBIN , 2=WEIGHTBASED , 3=HA-->

62 <property name="loadbalance">1</property>

63

64 <!-- Separated by commas,such as: server1,server2,server1 -->

65 <property name="poolNames">server1,server2</property>

66 </poolConfig>

67 </dbServer>

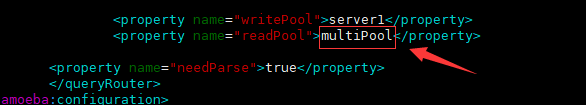
第二步：配置amoeba.xml文件：

amoeba.xml文件里我们要修改118行。

vim /usr/local/amoeba/conf/amoeba.xml --修改配置文件

117 <property name="writePool">server1</property> ---设置写的服务器，

118 <property name="readPool">multiPool</property> ---设置读的服务器。将原来的server2改成multiPool。multiPool就是第一步设置的集群服务器名称



第三步：重启amoeba代理软件。

1、先将amoeba代理软件重启

命令：

--cd到amoeba安装目录：cd /usr/local/amoeba/

--停止：./bin/amoeba stop

--启动：nohup ./bin/amoeba start &

绝对路径： /usr/local/amoeba/bin/amoeba stop

绝对路径： nohup /usr/local/amoeba/bin/amoeba start &

2、查看amoeba软件是否启动成功

命令：

--查看nohup.out文件，里面有启动的信息，如果有以下这段就代表启动成功。

2016-09-05 22:24:30,897 INFO net.ServerableConnectionManager - Amoeba for Mysql listening on 0.0.0.0/0.0.0.0:3066.

2016-09-05 22:24:30,934 INFO net.ServerableConnectionManager - Amoeba Monitor Server listening on /127.0.0.1:19659.

--查看amoeba进程，判断是否启动。

命令：

--使用ps命令：ps -ef|grep mysql，如果能查出来进程，就代表已有进程，如果无则启动失败。

[root@node1 amoeba]# ps -ef|grep amoeba

root 12855 12471 0 22:24 pts/3 00:00:09 /usr/local/java/bin/java -server -Xms256m -Xmx256m -Xss228k -Damoeba.home=/usr/local/amoeba -Dclassworlds.conf=/usr/local/amoeba/bin/amoeba.classworlds -classpath /usr/local/amoeba/lib/classworlds-1.0.jar org.codehaus.classworlds.Launcher start

第四步：在客户机上测试读集群是否生效。

先在sql服务器上创建emp表

mysql> create table emp (id int, ename varchar(20));

在客户机上使用mysql命令登陆amoeba服务器

#mysql -uamoeba -p123 -h192.168.224.11 -P8066

--往emp表里插入两条数据

mysql> insert into emp values(10,"test1");

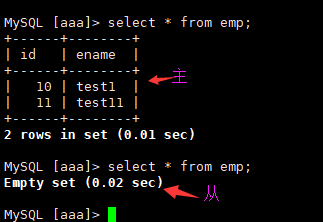
mysql> insert into emp values(11,"test11");

注：配置文件定义的是客户端写入数据是进入server1服务器上，server2服务器上就没有数据，

客户端也可以读server1服务器上的数据。

--再查询emp表的数据

mysql> select \* from emp; --查询的时候一次查询mysql 1上面的数据，再查询一次又跳到另外一台mysql服务器上了



到此mysql读集群配置完毕。

====================================================================================

end