**Cobar安装文档**

**概述：**

Cobar是关系型数据库的分布式处理系统，它可以在分布式的环境下看上去像传统数据库一样提供海量数据服务

vi 详解

http://wenku.baidu.com/link?url=lwNWeoLEG0UnzoMhdw7X2Vek0sNkIqAaRVk0Uh7CkB51yAbXXD8ZarQuIZjuKfGRtEoRXBwn5Yq5zF74EcA7SHIVs84ocI57Vpnid7ruFQ7

说明：

所有软件以及环境依据此文档为准，并且需要用root超级用户来完成，另外需要关闭防火墙。不保证其他环境能成功

**步骤一：环境准备**

1. 软件准备

操作系统：Linux或者windows（推荐在Linux环境下运行Cobar，在这里我们用Ubuntu系统）

JDK：<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Cobar：<https://github.com/alibaba/cobar/wiki/%E4%B8%8B%E8%BD%BD（下载tar.gz>或者ZIP）

1. 安装JDK

1）解压JDK

tar xzvf jdk-8u5-linux-x64.gz

mv jdk1.8.0\_05/ /usr/local/java

2）配置环境变量

vi ~/.bashrc

export JAVA\_HOME=/usr/local/java

export JRE\_HOME=/usr/local/java/jre

export CLASSPATH=.:$JAVA\_HOME/lib:$JRE\_HOME/lib:$CLASSPATH

export PATH=$JAVA\_HOME/bin:$JRE\_HOME/bin:$PATH

source ~/.bashrc

执行以下命令

update-alternatives --install /usr/bin/java java /usr/local/java/bin/java 300

update-alternatives --install /usr/bin/javac javac /usr/local/java/bin/javac 300

测试java

java –version

1. 数据库准备

数据库准备

Ubunt下用apt-get安装mysql

apt-get install mysql-server

apt-get install mysql-client

启动方式

　 1、使用 service 启动：service mysqld start

　　 2、使用 mysqld 脚本启动：/etc/inint.d/mysqld start

　　 3、使用 safe\_mysqld 启动：safe\_mysqld&

停止

　　 1、使用 service 启动：service mysqld stop

　　 2、使用 mysqld 脚本启动：/etc/inint.d/mysqld stop

　　 3、mysqladmin shutdown

重启

　　 1、使用 service 启动：service mysqld restart

　　 2、使用 mysqld 脚本启动：/etc/inint.d/mysqld restart

用户名为myuser，密码为空，需要创建

mysql -uroot -p

grant all on \*.\* to 'myuser'@'%' identfied by 'bjtest1234';

grant all on \*.\* to 'myuser'@'localhost' identfied by 'bjtest1234';

flush privileges;

schema:dbtest1、dbtest2、dbtest3，table：tb1、tb2，脚本如下：

|  |
| --- |
| 数据库创建脚本 |
| #创建dbtest1  drop database if exists dbtest1;  create database dbtest1;  use dbtest1;  #在dbtest1上创建tb1  create table tb1(  id int not null,  gmt datetime);    #创建dbtest2  drop database if exists dbtest2;  create database dbtest2;  use dbtest2;  #在dbtest2上创建tb2  create table tb2(  id int not null,  val varchar(256));    #创建dbtest3  drop database if exists dbtest3;  create database dbtest3;  use dbtest3;  #在dbtest3上创建tb2  create table tb2(  id int not null,  val varchar(256)); |

**步骤二：部署和配置Cobar**

请确保机器上设置了JAVA环境变量JAVA\_HOME

下载Cobar压缩文件并解压，进入conf目录可以看到schema.xml, rule.xml, server.xml等相关的配置文件

tar xzvf cobar-server-1.2.7.tar.gz

mv cobar-server-1.2.7/ /usr/local/cobar

cd /usr/local/cobra/conf

schema.xml配置如下(注意：schema.xml包含MySQL的IP、端口、用户名、密码等配置，您需要按照注释

替换为您的MySQL信息。)

|  |
| --- |
| schema.xm |
| <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE cobar:schema SYSTEM "schema.dtd">  <cobar:schema xmlns:cobar="http://cobar.alibaba.com/">    <!-- schema定义 -->  <schema name="dbtest" dataNode="dnTest1">  <table name="tb2" dataNode="dnTest2,dnTest3" rule="rule1" />  </schema>    <!-- 数据节点定义，数据节点由数据源和其他一些参数组织而成。-->  <dataNode name="dnTest1">  <property name="dataSource">  <dataSourceRef>dsTest[0]</dataSourceRef>  </property>  </dataNode>  <dataNode name="dnTest2">  <property name="dataSource">  <dataSourceRef>dsTest[1]</dataSourceRef>  </property>  </dataNode>  <dataNode name="dnTest3">  <property name="dataSource">  <dataSourceRef>dsTest[2]</dataSourceRef>  </property>  </dataNode>    <!-- 数据源定义，数据源是一个具体的后端数据连接的表示。-->  <dataSource name="dsTest" type="mysql">  <property name="location">  <location>192.168.0.1:3306/dbtest1</location> <!--注意：替换为您的  MySQL IP和Port-->  <location>192.168.0.1:3306/dbtest2</location> <!--注意：替换为您的  MySQL IP和Port-->  <location>192.168.0.1:3306/dbtest3</location> <!--注意：替换为您的  MySQL IP和Port-->  </property>  <property name="user">myuser</property> <!--注意：替换为您的MySQL用户名-->  <property name="password">bjtest1234</property> <!--注意：替换为您的MySQL密码-->  <property name="sqlMode">STRICT\_TRANS\_TABLES</property>  </dataSource>  </cobar:schema> |

**注意：这里的IP地址一定是本机的.实际地址。**

rule.xml配置如下(本文仅以数字类型的id字段作为拆分字段，将数据拆分到两个库中。)

|  |
| --- |
| rule.xml |
| <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE cobar:rule SYSTEM "rule.dtd">  <cobar:rule xmlns:cobar="http://cobar.alibaba.com/">  <!-- 路由规则定义，定义什么表，什么字段，采用什么路由算法。-->  <tableRule name="rule1">  <rule>  <columns>id</columns>  <algorithm><![CDATA[ func1(${id})]]></algorithm>  </rule>  </tableRule>    <!-- 路由函数定义，应用在路由规则的算法定义中，路由函数可以自定义扩展。-->  <function name="func1"  class="com.alibaba.cobar.route.function.PartitionByLong">  <property name="partitionCount">2</property>  <property name="partitionLength">512</property>  </function>  </cobar:rule> |

server.xml配置如下

|  |
| --- |
| server.xm |
| <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE cobar:server SYSTEM "server.dtd">  <cobar:server xmlns:cobar="http://cobar.alibaba.com/">    <!--定义Cobar用户名，密码-->  <user name="test">  <property name="password">test</property>  <property name="schemas">dbtest</property>  </user>  </cobar:server> |

**步骤三：启动和使用Cobar**

1）、启动Cobar，进入bin目录可以看到Cobar的启动、停止与重启脚本./startup.sh #Cobar进程名为CobarStartup

2）、查看logs目录下stdout.log, 启动成功日志如下

10:54:19,264 INFO ===============================================

10:54:19,265 INFO Cobar is ready to startup ...

10:54:19,265 INFO Startup processors ...

10:54:19,443 INFO Startup connector ...

10:54:19,446 INFO Initialize dataNodes ...

10:54:19,470 INFO dnTest1:0 init success

10:54:19,472 INFO dnTest3:0 init success

10:54:19,473 INFO dnTest2:0 init success

10:54:19,481 INFO CobarManager is started and listening on 9066

10:54:19,483 INFO CobarServer is started and listening on 8066

10:54:19,484 INFO ===============================================

3）、访问Cobar同访问MySQL的方式完全相同, 常用访问方式如下(注意：本文将Cobar部署在192.168.0.1这台机

器上，否则请替换为您的Cobar所在IP，其他信息不变)

#命令行

mysql -h192.168.0.1 -utest -ptest -P8066 -Ddbtest

4）、SQL执行示例，执行语句时与使用传统单一数据库无区别mysql>show databases;

#dbtest1、dbtest2、dbtest3对用户透明

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

| DATABASE |

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

| dbtest |

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

mysql>show tables; #dbtest

中有两张表tb1和tb2

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

| Tables\_in\_dbtest1 |

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

| tb1 |

| tb2 |

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

mysql>insert into tb1 (id, gmt) values (1, now()); #向表

tb1插入一条数据

mysql>insert into tb2 (id, val) values (1, "part1"); #向表

tb2插入一条数据

mysql>insert into tb2 (id, val) values (2, "part1"), (513, "part2"); #向表

tb2同时插入多条数据

mysql>select \* from tb1; #查询表

tb1，验证数据被成功插入

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

| id | gmt |

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

| 1 | 2012-06-12 15:00:42 |

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

mysql>select \* from tb2; #查询

tb2，验证数据被成功插入

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

| id | val |

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

| 1 | part1 |

| 2 | part1 |

| 513 | part2 |

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

mysql>select \* from tb2 where id in (1, 513); #根据id

查询

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

| id | val |

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

| 1 | part1 |

| 513 | part2 |

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

至此，配置成功完成。