**MyCAT读写分离模式智能优化方案**

**修订记录**

| **日期** | **描述** | **作者** | **邮箱** | **QQ** | **版本号** |
| --- | --- | --- | --- | --- | --- |
| 2015-03-30 | 创建文档 | 杨超 | bobiy@foxmail.com | 654839088 | V1.00 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[1. 前言 3](#_Toc415476422)

[1.1编写目的 3](#_Toc415476423)

[1.2预期读者 3](#_Toc415476424)

[2. MySQL引擎浅谈 3](#_Toc415476425)

[3. 取长补短，兼容并包 4](#_Toc415476426)

[4. MySQL主从配置 5](#_Toc415476427)

[5. 主从配置的MySQL创建表方式 6](#_Toc415476428)

[6. 启用Mycat的读写分离机制 7](#_Toc415476429)

[7. 两种模式的执行时间比照图 7](#_Toc415476430)

## 前言

## 1.1编写目的

通过对mycat的研究和使用，目前已经正式应用于我公司的项目中，为了感谢群里各位朋友的帮助，现对mycat读写分离模式的智能优化方案分享给大家。本文档属于进阶篇，写者建议大家在阅读本文档之前，请先阅读mycat的入门文档，并能初步使用mycat时再阅读。本人技术泛泛，若有不对之处，望大能指出。

## 1.2预期读者

本文的预期读者包括：技术管理人员、技术开发人员、运维人员

## MySQL引擎浅谈

在进入正题前，我这里先与大家谈谈MySql的引擎 InnoDB与MyISAM，这里，本文只考虑这两种引擎，因为这两个是常见的。

MyISAM 是MySQL中默认的存储引擎，一般来说不是有太多人关心这个东西。决定使用什么样的存储引擎是一个很tricky的事情，但是还是值我们去研究一下。

下面先让我们回答一些问题：

* 你需要事务支持吗？
* 你需要全文索引吗？
* 你经常使用什么样的查询模式?

思考上面这些问题可以让你找到合适的方向，但那并不是绝对的。如果你需要事务处理，那么InnoDB 可能是比较好的方式。如果你需要全文索引，那么通常来说 MyISAM是好的选择，因为这是系统内建的。

事务支持，是一个影响你选择什么样存储引擎的重要因素，事务支持趋向于选择InnoDB方式，因为其支持事务处理和故障恢复。另外InnoDB可以利用事务日志进行数据恢复，这会比MyISAM快很多。而MyISAM可能会需要几个小时甚至几天来干这些事，InnoDB 只需要几分钟。

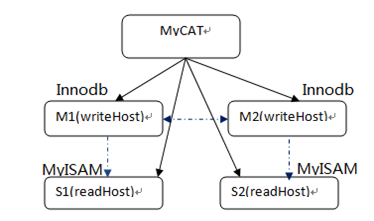
当然操作数据库表的习惯可能也会是一个对性能影响很大的因素。比如： COUNT() 在 MyISAM 表中会非常快，而在InnoDB 表下可能会很痛苦。另外MyISAM存储引擎在筛选大量数据时非常迅速，这是它最突出的优点。MyISAM还提供了大量的特性，包括全文索引、压缩、空间函数(GIS)等，但MyISAM不支持事务和行级锁，有一个毫无疑问的缺陷就是崩溃后无法安全恢复。在读多写少的业务，如果不介意MyISAM的崩溃恢复问题，选用MyISAM引擎是合适的。

所以，到底你检使用哪一个呢？根据应用场景来说：

* MyISAM管理非事务表。它提供高速存储和检索，以及全文搜索能力。如果应用中需要执行大量的SELECT查询，那么MyISAM是更好的选择。
* InnoDB用于事务处理应用程序，具有众多特性，包括ACID事务支持。如果应用中需要执行大量的INSERT或UPDATE操作，则应该使用InnoDB，这样可以提高多用户并发操作的性能。

## 取长补短，兼容并包

如前文所述，MySQL的引擎各有千秋，我设计方案的初衷就是希望能结合两种引擎的优势，摒弃各自的劣势，创造出一种接近“理想”状态的使用方式，既可以做到InnoDB的事务处理，又可以使用到MyISAM特性。而这种方式简单概括起来就是：利用MyCat的读写分离模式，写节点使用MySql的InnoDB引擎，读节点使用的是MySql的MyISAM引擎。如下图所示：



Mycat的入门使用教程，这里我不再赘述，有需要请到

<https://github.com/MyCATApache/Mycat-doc>下载。这里我详细谈谈如何利用Mycat的读写分离机制来最大限度的利用MySQL强大的功能。

## MySQL主从配置

我们的方案设计基于Mysql的主从数据复制功能，以下是基于mysql5.5以上版本最新的主从配置。

开启mysql主从数据复制，主要在mysql的my.ini文件中设置：

windows系统一般在c:\ProgrameData\MySQL中：

Linux系统一般在 /etc/my.cnf中：

**1、主数据库服务器配置：**

server-id=1

log-bin=mysql-bin

binlog-do-db=Industry       //只同步Industry数据库

binlog-ignore-db=mysql    //避免同步mysql用户配置，以免不必要的麻烦

binlog-ignore-db=performance\_schema  
binlog-ignore-db=information\_schema

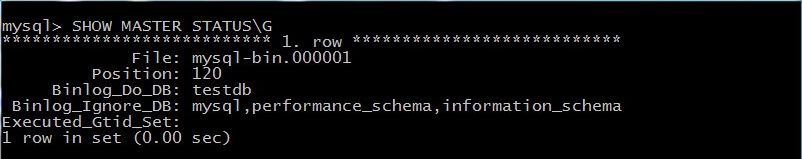
**2、创建一个允许从服务器来访问的用户(主服务器)：**

grant replication slave on \*.\* to 'root'@'192.168.0.58' identified by 'root';

说明：

* root：Slave使用的账号
* IDENTIFIED BY 'root' ：Slave使用的密码
* 192.168.1.58：Slave数据库IP

master端，通过show master status\G 显示信息。如图：



注意上图中的File,和Position。Slave 中会用到这些信息。

**3、从数据库服务器配置：**

server-id=11 //服务Id，注意要和master的server-id不一样

log-bin=mysql-bin

relay-log-index=slave-relay-bin.index

relay-log=slave-relay-bin

sync\_master\_info=1  
sync\_relay\_log=1  
sync\_relay\_log\_info=1

**4、关联上Master信息：**

change master to

master\_host='192.168.0.68',master\_user='root',master\_port=3306,master\_password='root' master\_log\_file='mysql-bin.000001',master\_log\_pos=120;

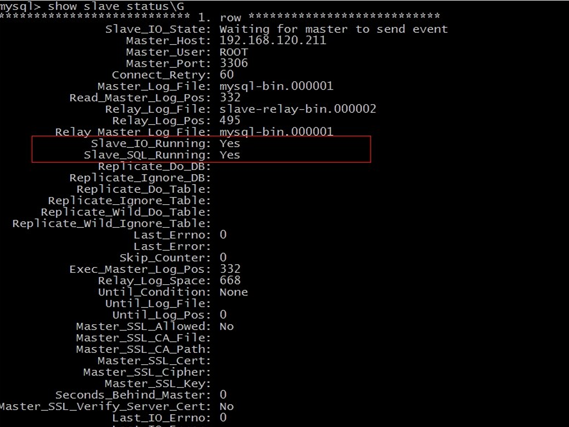
命令说明：

* master\_host：master 数据库服务器ip
* master\_user: master  数据库通过GRANT授权的账号
* master\_port：master  数据库使用的端口号
* master\_password：master  数据库通过GRANT授权的密码
* master\_log\_file：master数据库中通过show master status显示的File名称
* master\_log\_pos：master数据库中通过show master status显示的Position数据

重启Slave 的MySql服务

在命令行中输入start slave.

在命令行中输入show slave status,显示如下：



 注意：只有Slave\_IO\_Running与Slave\_SQL\_Running都为Yes才配置成功。

至此，配置全部完成。这样，Master服务器上Industry里的数据就能同步到slave的Industry数据库中。

## 主从配置的MySQL创建表方式

由于Mysql已经配置好了主从复制，在创建表时，我们要注意，在Master库中创建表后，Mysql会自动在Slave库中创建一张表。这里我们需要做的是，将从库中自动创建的表删除，然后手动创建一张我们符合我们需求的表。具体操作步骤如下：

* 1. 在主库中创建表A，引擎选择Innodb；
  2. 删除从库中自动创建的表A；
  3. 在从库中创建表A，引擎选择MyISAM；

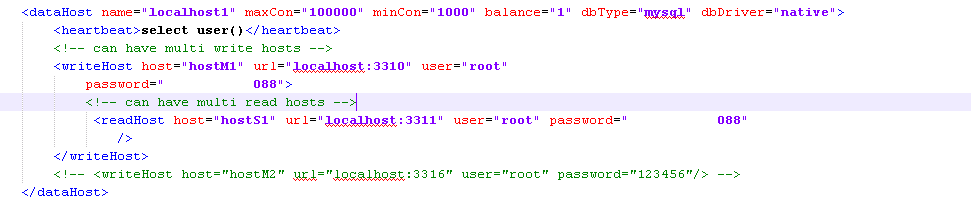
## 启用Mycat的读写分离机制

最后启用Mycat的读写分离机制即可。

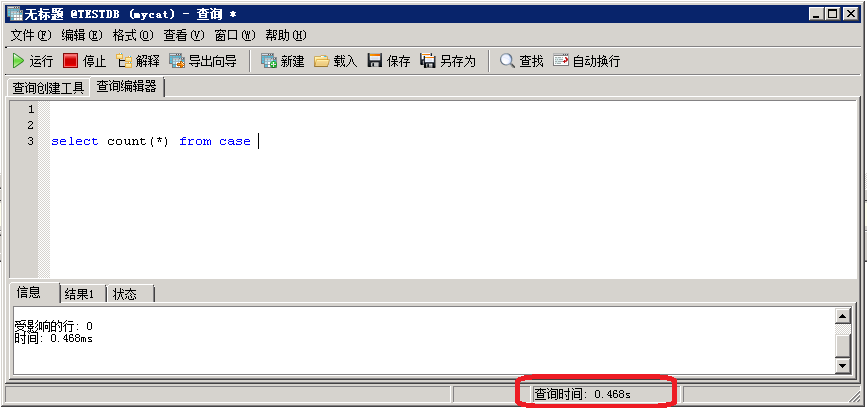
## 两种模式的执行时间比照图

再多的废话也不如实际的测试来的有效，接下来，让我们实际测试利用这种方案能给我们带来多大的执行效率。

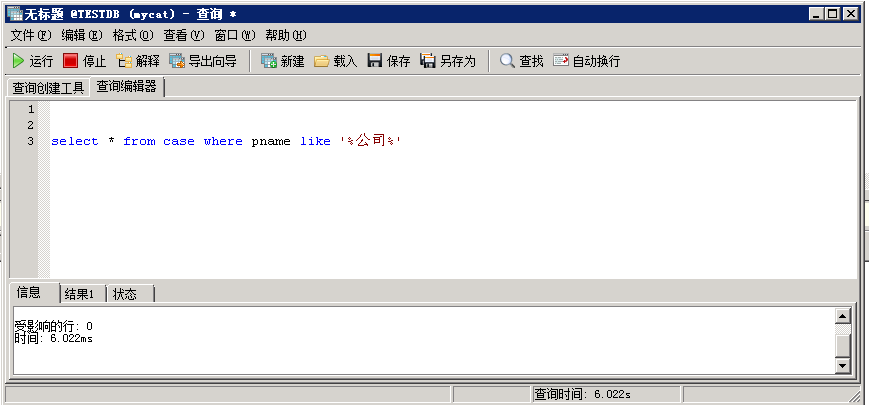
首先，在开启mycat的读写分离机制后，如下图：



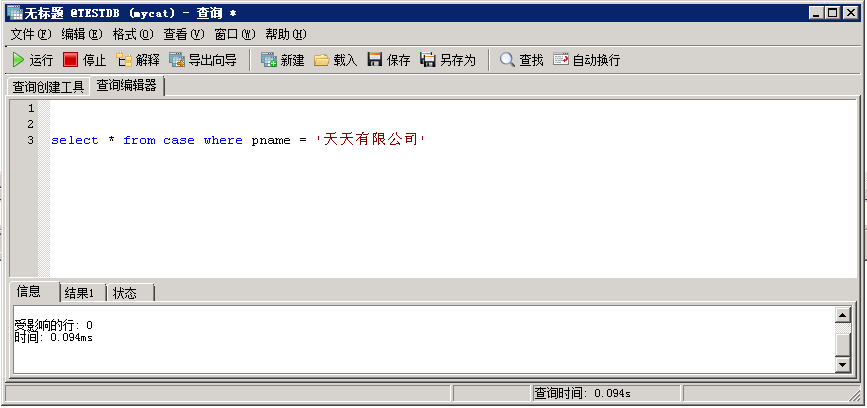
此时写入库的引擎为Innodb，读库的引擎为MyISAM，这时使用count语句,执行时间如下图：



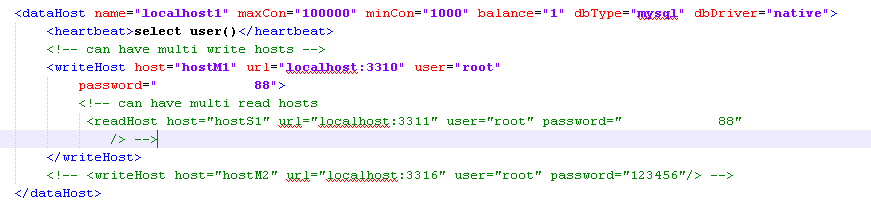
使用like条件语句执行时间如下图：



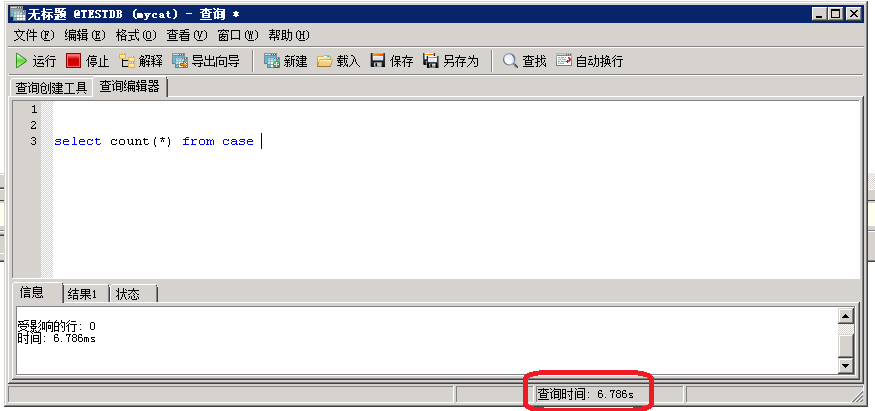
使用等于条件语句执行时间如下图：



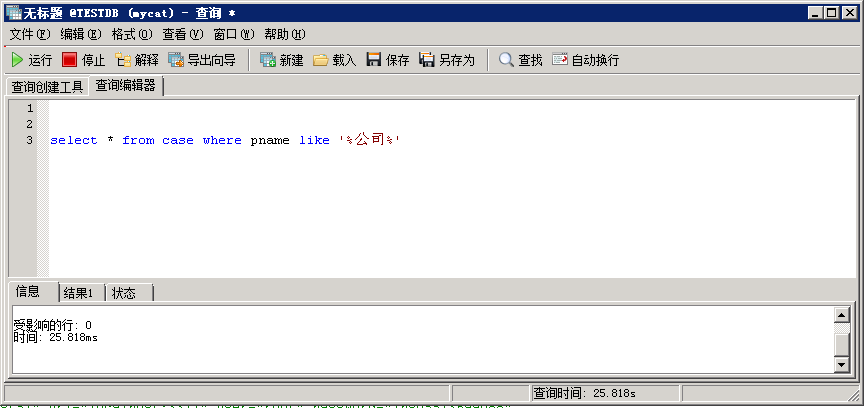
接下来，我们取消读写分离机制，见下图：



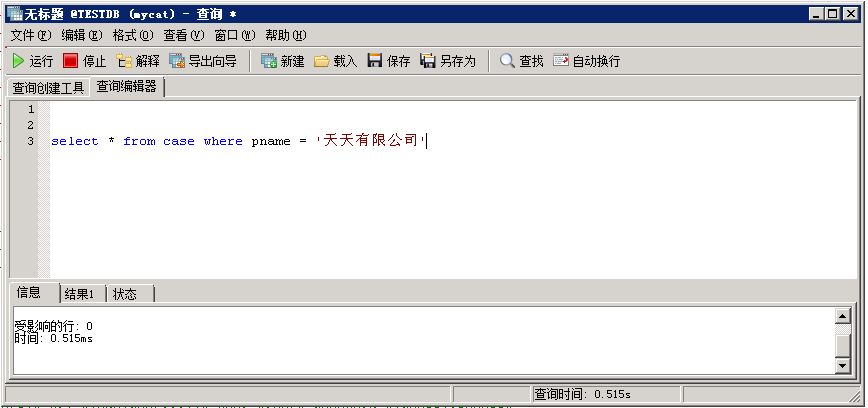
此时只有写入库，并且引擎为Innodb，这时使用count语句,执行时间如下图:



使用like条件语句执行时间如下图：



使用等于条件语句执行时间如下图：



总体的执行时间对比表，见下图：

|  |  |  |
| --- | --- | --- |
| Sql语句 | 启用读写分离机制（读库使用MyISAM） | 取消读写分离机制(读使用的是Innodb) |
| count | 0.468秒 | 6.786秒 |
| like | 6.022秒 | 25.818秒 |
| = | 0.094秒 | 0.515秒 |

杨超 2015-03-28

QQ：654839088

邮箱：bobiy@foxmail.com