

## 1. Mycat 是什么?

从定义和分类来看,它是一个开源的分布式数据库系统,是一个实现了 MySQL 协议的Server,前端用户可以把它看作是一个数据库代理,用 MySQL 客户端工具和命令行访问,而其后端可以用MySQL 原生(Native)协议与多个 MySQL 服务器通信,也可以用 JDBC 协议与大多数主流数据库服务器通信,其核心功能是分表分库,即将一个大表水平分割为 N 个小表,存储在后端 MySQL 服务器里或者其他数据库里。Mycat 发展到目前的版本,已经不是一个单纯的 MySQL 代理了,它的后端可以支持 MySQL、SQL Server、Oracle、DB2、PostgreSQL 等主流数据库,也支持 MongoDB 这种新型 NoSQL 方式的存储,未来还会支持更多类型的存储。而在最终用户看来,无论是那种存储方式,在 Mycat 里,都是一个传统的数据库表,支持标准的SQL 语句进行数据的操作,这样一来,对前端业务系统来说,可以大幅降低开发难度,提升开发速度,在测试阶段,可以将一个表定义为任何一种 Mycat 支持的存储方式,比如 MySQL 的 MyASIM 表、内存表、或者MongoDB、LevelDB 以及号称是世界上最快的内存数据库 MemSQL 上。试想一下,用户表存放在 MemSQL 上,

Mycat 就是 BigSQL, Big Data On SQL Database。

## 1.1 SQL查询比较慢的原因?

SQL复杂程度,数据量,循环操作次数过多

解决:

通过一些算法或者是方法:比如冒泡算法或者其他规则的算法,在tp框架中,会将这些sq方法(比如insert,update)等写入到方法中,而mycat就相当于一个中间件一样的东西,作为一个中间层

## 1.2

## 1.2.1 对于 DBA 来说,可以这么理解 Mycat:

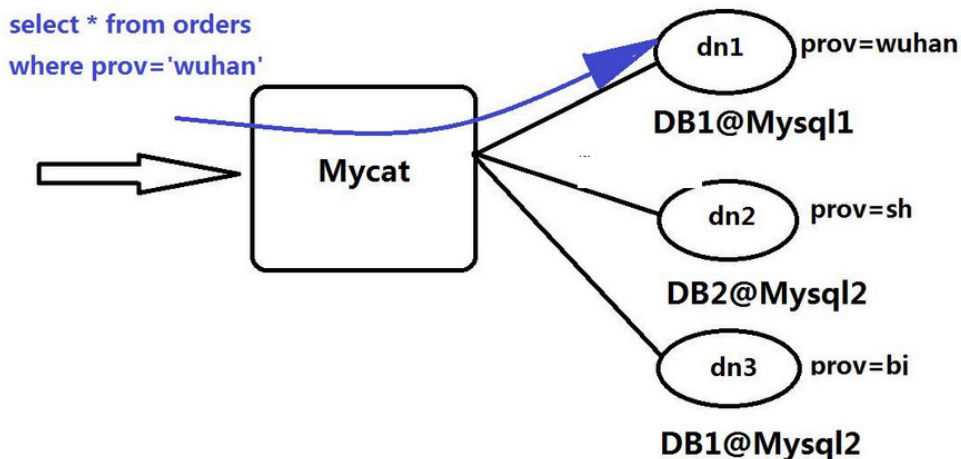
Mycat 就是 MySQL Server,而 Mycat 后面连接的 MySQL Server,就好像是 MySQL 的存储引擎,如 InnoDB, MyISAM 等,因此, Mycat 本身并不存储数据,数据是在后端的 MySQL 上存储的,因此数据可靠性以及事务等都是 MySQL 保证的,简单的说, Mycat 就是 MySQL 最佳伴侣,它在一定程度上让 MySQL 拥有了能跟 Oracle PK 的能力。

## 1.2.2 对于开发人员来说,可以这么理解 Mycat:

Mycat 就是一个近似等于 MySQL 的数据库服务器,你可以用连接 MySQL 的方式去连接 Mycat (除了端口不同,默认的 Mycat端口是 8066 而非 MySQL 的 3306,因此需要在连接字符串上增加端口信息),大多数情况下,可以用你熟悉的对象映射框架使用 Mycat,但建议对于分片表,尽量使用基础的 SQL 语句,因为这样能达到最佳性能,特别是几千万甚至几百亿条记录的情况下。

## 1.3原理

Mycat 的原理中最重要的一个动词是“拦截”,它拦截了用户发送过来的 SQL 语句,首先对 SQL 语句做了一些特定的分析:如分片分析、路由分析、读写分离分析、缓存分析等,然后将此 SQL 发往后端的真实数据库,并将返回的结果做适当的处理,最终再返回给用户。

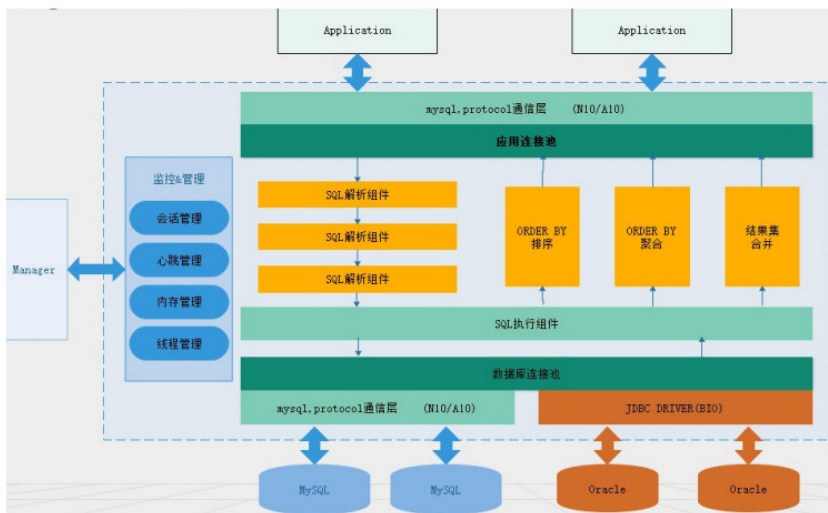


上述图片里, Orders 表被分为三个分片 datanode (简称 dn),这三个分片是分布在两台 MySQL Server 上(DataHost),即 datanode=database@datahost 方式,因此你可以用一台到 N 台服务器来分片,分片规则为 (sharding rule)典型的字符串枚举分片规则,一个规则的定义是分片字段 (sharding column)+分片函数 (rulefunction),这里的分片字段为 prov 而分片函数为字符串枚举方式。

当 Mycat 收到一个 SQL 时,会先解析这个 SQL,查找涉及到的表,然后看此表的定义,如果有分片规则,则获取到 SQL 里分片字段的值,并匹配分片函数,得到该 SQL 对应的分片列表,然后将 SQL 发往这些分片去执行,最后收集和处理所有分片返回的结果数据,并输出到客户端。以 `select * from Orders where prov=?` 语句为例,查到 `prov=wuhan`,按照分片函数, `wuhan` 返回 `dn1`,于是 SQL 就发给了 MySQL1,去取 DB1 上的查询结果,并返回给用户。

如果上述 SQL 改为 `select * from Orders where prov in ('wuhan','beijing')`,那么,SQL 就会发给MySQL1 与 MySQL2 去执行,然后结果集合合并后输出给用户。但通常业务中我们的 SQL 会有 Order By 以及Limit 翻页语法,此时就涉及到结果集在 Mycat 端的二次处理,这部分的代码也比较复杂,而最复杂的则属两个表的 Join 问题,为此, Mycat 提出了创新性的 ER 分片、全局表、HBT (Human Brain Tech)人工智能的 Catlet、以及结合 Storm/Spark 引擎等十八般武艺的解决办法





## 2. 应用

**Mycat** 发展到现在, 适用的场景已经很丰富, 而且不断有新用户给出新的创新性的方案, 以下是几个典型的应用场景:

单纯的读写分离, 此时配置最为简单, 支持读写分离, 主从切换;

分表分库, 对于超过 1000 万的表进行分片, 最大支持 1000 亿的单表分片;

多租户应用, 每个应用一个库, 但应用程序只连接 **Mycat**, 从而不改造程序本身, 实现多租户化;

表系统, 借助于 **Mycat** 的分表能力, 处理大规模报表的统计;

替代 Hbase, 分析大数据;

作为海量数据实时查询的一种简单有效方案, 比如 100 亿条频繁查询的记录需要在 3 秒内查询出来结果, 除了基于主键的查询, 还可能存在范围查询或其他属性查询, 此时 **Mycat** 可能是最简单有效的选择。

## 3. 安装(官网地址:<http://www.mycat.org.cn/>)

### 3.1 这里准备好了素材

链接: <https://pan.baidu.com/s/1YABrx-RMLJ3qU5XB21JGzw>

提取码: m5vc

### 3.2 windows安装

因为是java写的, 所以需要一些拓展支持(jdk)

安装mycat前需要先安装jdk和mysql. mycat1.6版本建议使用的jdk是1.7以上版本, mysql建议使用5.6版本。

mysql安装不在赘述, jdk安装, 获取资源后, 傻瓜式安装, 下一步点击即可

配置一下环境变量 C:\jdk\bin, dos界面输入java, 出现有效信息即可。

mycat安装: 解压 Mycat-server-1.6.7.3-release-20190927161129-win.tar

重命名(可以不用操作), 命名为mycat, 移动到自己喜欢的目录下

#### 3.2.1 测试连接:

1)

在 C:\mycat\bin 双击 startup\_nowrap.bat, 出现

```
C:\mycat\bin>"java" -server -Xms1G -Xmx2G -XX:MaxPermSize=64M -XX:+AggressiveOpts
-XX:MaxDirectMemorySize=1G-DMYCAT_HOME=C:\mycat -cp "..\conf;..\lib\*" io.mycat.MycatStartupJava
HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=64M; support was removed in 8.0
证明安装成功
```

2)

配置一下环境变量

在doc界面输入mycat的默认用户名和密码以及端口连接(默认: user=root port=8066 password=123456)

输入命令:mysql -uroot -p123456 -P8066

```
C:\Users\Administrator>mycat.bat
```

```
Usage: mycat.bat { console : start : pause : resume : stop : restart : install : remove : status }
```

```
C:\Users\Administrator>mycat.bat console
```

```
wrapper | --> Wrapper Started as Console
```

```
wrapper | Launching a JVM...
```

```
jvm 1 | Wrapper (Version 3.2.3) http://wrapper.tanukisoftware.org
```

```
jvm 1 | Copyright 1999-2006 Tanuki Software, Inc. All Rights Reserved.
```

```
jvm 1 |  
jvm 1 | MyCAT Server startup successfully. see logs in logs/mycat.log //证明启动成功
```

注意当启动一次后,在其他窗口再次启动会报错

```
C:\Users\Administrator>mysql -uroot -p123456 -P8066  
mysql: [Warning] Using a password on the command line interface can be insecure.  
ERROR 1045 (HY000): Access denied for user 'root', because password is error  
报错原因是命令行直接使用密码不安全,另外是mysql8.0更新了加密方式 caching_sha2_password ,mycat暂时不支持
```

所以登陆时更改加密方式(指定数据表):

```
C:\Users\Administrator>mysql -uroot -p -h127.0.0.1 -P8066 -DTESTDB
```

```
--default_auth=mysql_native_password
```

这里就只有一张数据表

```
mysql> show databases;
```

```
+-----+
```

```
| DATABASE |
```

```
+-----+
```

```
| TESTDB |
```

```
+-----+
```

```
1 row in set (0.01 sec)
```

不建议使用navicat连接,支持不是很好,要等待很久

### 3.3 Linux安装

下载安装包jdk-8u20-linux-x64.tar.gz/ Mycat-server-1.6.7.3-release-20190828135747-linux.tar.gz

#### 3.3.1 安装jdk(这里尝试手动安装,也可以直接 yum install 安装)

解压 tar -zxvf jdk-8u20-linux-x64.tar.gz

移动到指定目录 mv jdk1.8.0\_20 jdk | mv jdk /usr/local/java

配置环境变量: vim /etc/profile

#指定文件的根目录

```
export JAVA_HOME=/usr/local/java
```

#是java的一些驱动

```
export CLASSPATH=.:${JAVA_HOME}/jre/lib/rt.jar:${JAVA_HOME}/lib/dt.jar:${JAVA_HOME}/lib/tools.jar
```

```
export PATH=$PATH:${JAVA_HOME}/bin
```

刷新操作: source /etc/profile

测试 直接输出 java

mycat的安装:

解压: tar -zxvf Mycat-server-1.6.7.3-release-20190828135747-linux.tar.gz

移动到指定目录: mv mycat /usr/local

进入 /usr/local/mycat/bin下,指定开启/关闭: ./mycat start/stop/console

//默认连接密码123456

连接方法: mysql -uroot -p -h127.0.0.1 -P8066 -DTESTDB --default\_auth=mysql\_native\_password

```
mysql> show databases;
```

```
+-----+
```

```
| DATABASE |
```

```
+-----+
```

```
| TESTDB |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> show tables;
```

```
+-----+
```

```
| Tables in TESTDB |
```

```
+-----+
```

```
| company |
```

```
| customer |
```

```
| customer_addr |
```

```
| employee |
```

```
| goods |
```

```
| hotnews |
```

```
| orders |
```

```
| order_items |
```

```
| travelrecord |  
+-----+  
9 rows in set (0.00 sec) //多了许多用于测试的表
```