主从复制、读写分离

读写分离，基本的原理是让主数据库处理事务性增、改、删操作（INSERT、UPDATE、DELETE），而从数据库处理SELECT查询操作。数据库复制被用来把事务性操作导致的变更同步到集群中的从数据库。

4.1、 mysq支持的复制类型

1） 基于语句的复制。在服务器上执行sql语句，在从服务器上执行同样的语句，mysql默认采用基于语句的复制，执行效率高。

2） 基于行的复制。把改变的内容复制过去，而不是把命令在从服务器上执行一遍。

3） 混合类型的复制。默认采用基于语句的复制，一旦发现基于语句无法精确复制时，就会采用基于行的复制。

4.2、 复制的工作过程

1） 在每个事务更新数据完成之前，master在二进制日志记录这些改变。写入二进制日志完成后，master通知存储引擎提交事务。

2） Slave将master的binary log复制到其中继日志。首先slave开始一个工作线程（I/O），I/O线程在master上打开一个普通的连接，然后开始binlog dump process。binlog dump process从master的二进制日志中读取事件，如果已经跟上master，它会睡眠并等待master产生新的事件，I/O线程将这些事件写入中继日志。

3） Sql slave thread（sql从线程）处理该过程的最后一步，sql线程从中继日志读取事件，并重放其中的事件而更新slave数据，使其与master中的数据一致，只要该线程与I/O线程保持一致，中继日志通常会位于os缓存中，所以中继日志的开销很小。

前较为常见的Mysql读写分离分为以下两种：

1）基于程序代码内部实现

在代码中根据select 、insert进行路由分类，这类方法也是目前生产环境下应用最广泛的。优点是性能较好，因为程序在代码中实现，不需要增加额外的硬件开支，缺点是需要开发人员来实现，运维人员无从下手。

1

2） 基于中间代理层实现

代理一般介于应用服务器和数据库服务器之间，代理数据库服务器接收到应用服务器的请求后根据判断后转发到，后端数据库，有以下代表性的程序。

1

（1）mysql\_proxy。mysql\_proxy是Mysql的一个开源项目，通过其自带的lua脚本进行sql判断。实际上是在客户端请求与MySQLServer之间建立了一个连接池。所有客户端请求都是发向MySQLProxy，然后经由MySQLProxy进行相应的分析，判断出是读操作还是写操作，分发至对应的MySQLServer上。对于多节点Slave集群，也可以起做到负载均衡的效果。

（2）Atlas。是由 Qihoo 360, Web平台部基础架构团队开发维护的一个基于MySQL协议的数据中间层项目。它是在mysql-proxy 0.8.2版本的基础上，对其进行了优化，增加了一些新的功能特性。360内部使用Atlas运行的mysql业务，每天承载的读写请求数达几十亿条。支持事物以及存储过程。

1. Amoeba。由阿里巴巴集团在职员工陈思儒使用序java语言进行开发，阿里巴巴集团将其用户生产环境下，但是他并不支持事物以及存储过程。