# 背景

为什么要做主从复制？

1、在业务复杂的系统中，有这么一个情景，有一句sql语句需要锁表，导致暂时不能使用读的服务，那么就很影响运行中的业务，使用主从复制，让主库负责写，从库负责读，这样，即使主库出现了锁表的情景，通过读从库也可以保证业务的正常运作。

2、做数据的热备

3、架构的扩展。业务量越来越大，I/O访问频率过高，单机无法满足，此时做多库的存储，降低磁盘I/O访问的频率，提高单个机器的I/O性能。

# 原理

mysql主从复制的原理是什么？

binlog: binary log，主库中保存所有更新事件日志的二进制文件。

主从复制过程：





主从复制的基础是主库记录数据库的所有变更记录到binlog。binlog是数据库服务器启动的那一刻起，保存所有修改数据库结构或内容的一个文件。

mysql主从复制是一个异步的复制过程，主库发送更新事件到从库，从库读取更新记录，并执行更新记录，使得从库的内容与主库保持一致。

在主库里，只要有更新事件出现，就会被依次地写入到binlog里面，之后会推到从库中作为从库进行复制的数据源。

**binlog输出线程**。每当有从库连接到主库的时候，主库都会创建一个线程然后发送binlog内容到从库。

对于每一个即将发送给从库的sql事件，binlog输出线程会将其锁住。一旦该事件被线程读取完之后，该锁会被释放，即使在该事件完全发送到从库的时候，该锁也会被释放。

在从库里，当复制开始的时候，从库就会创建两个线程进行处理：

**从库I/O线程**。当START SLAVE语句在从库开始执行之后，从库创建一个I/O线程，该线程连接到主库并请求主库发送binlog里面的更新记录到从库上。

从库I/O线程读取主库的binlog输出线程发送的更新并拷贝这些更新到本地文件，其中包括relay log文件。

**从库的SQL线程**。从库创建一个SQL线程，这个线程读取从库I/O线程写到relay log的更新事件并执行。

可以知道，对于每一个主从复制的连接，都有三个线程。拥有多个从库的主库为每一个连接到主库的从库创建一个binlog输出线程，每一个从库都有它自己的I/O线程和SQL线程。

从库通过创建两个独立的线程，使得在进行复制时，从库的读和写进行了分离。因此，即使负责执行的线程运行较慢，负责读取更新语句的线程并不会因此变得缓慢。比如说，如果从库有一段时间没运行了，当它在此启动的时候，尽管它的SQL线程执行比较慢，它的I/O线程可以快速地从主库里读取所有的binlog内容。这样一来，即使从库在SQL线程执行完所有读取到的语句前停止运行了，I/O线程也至少完全读取了所有的内容，并将其安全地备份在从库本地的relay log，随时准备在从库下一次启动的时候执行语句。

查看主从复制的状态

当主从复制正在进行中时，如果想查看从库两个线程运行状态，可以通过执行在从库里执行”show slave statusG”语句，以下的字段可以给你想要的信息：

Master\_Log\_File — 上一个从主库拷贝过来的binlog文件

Read\_Master\_Log\_Pos — 主库的binlog文件被拷贝到从库的relay log中的位置

Relay\_Master\_Log\_File — SQL线程当前处理中的relay log文件

Exec\_Master\_Log\_Pos — 当前binlog文件正在被执行的语句的位置

# 流程

整个主从复制的流程可以通过以下图示理解：





步骤一：主库db的更新事件(update、insert、delete)被写到binlog

步骤二：从库发起连接，连接到主库

步骤三：此时主库创建一个binlog dump thread，把binlog的内容发送到从库

步骤四：从库启动之后，创建一个I/O线程，读取主库传过来的binlog内容并写入到relay log

步骤五：还会创建一个SQL线程，从relay log里面读取内容，从`Exec\_Master\_Log\_Pos`位置开始执行读取到的更新事件，将更新内容写入到slave的db

注：上面的解释是解释每一步做了什么，整个mysql主从复制是异步的，不是按照上面的步骤执行的。

# 特点

## 好处

做数据的热备，作为后备数据库，主数据库服务器故障后，可切换到从数据库继续工作，避免数据丢失。

架构的扩展。业务量越来越大，I/O访问频率过高，单机无法满足，此时做多库的存储，降低磁盘I/O访问的评率，提高单个机器的I/O性能。

读写分离，使数据库能支持更大的并发。在报表中尤其重要。由于部分报表sql语句非常的慢，导致锁表，影响前台服务。如果前台使用master，报表使用slave，那么报表sql将不会造成前台锁，保证了前台速度。

**好处一:实现服务器负载均衡**

通过服务器复制功能，可以在主服务器和从服务器之间实现负载均衡。即可以通过在主服务器和从服务器之间切分处理客户查询的负荷，从而得到更好地客户相应时间。通常情况下，数据库管理员会有两种思路。

一是在主服务器上只实现数据的更新操作。包括数据记录的更新、删除、新建等等作业。而不关心数据的查询作业。数据库管理员将数据的查询请求全部 转发到从服务器中。这在某些应用中会比较有用。如某些应用，像基金净值预测的网站。其数据的更新都是有管理员更新的，即更新的用户比较少。而查询的用户数 量会非常的多。此时就可以设置一台主服务器，专门用来数据的更新。同时设置多台从服务器，用来负责用户信息的查询。将数据更新与查询分别放在不同的服务器 上进行，即可以提高数据的安全性，同时也缩短应用程序的响应时间、提高系统的性能。

二是在主服务器上与从服务器切分查询的作业。在这种思路下，主服务器不单单要完成数据的更新、删除、插入等作业，同时也需要负担一部分查询作 业。而从服务器的话，只负责数据的查询。当主服务器比较忙时，部分查询请求会自动发送到从服务器重，以降低主服务器的工作负荷。当然，像修改数据、插入数 据、删除数据等语句仍然会发送到主服务器中，以便主服务器和从服务器数据的同步。

**好处二：通过复制实现数据的异地备份**

可以定期的将数据从主服务器上复制到从服务器上，这无疑是先了数据的异地备份。在传统的备份体制下，是将数据备份在本地。此时备份 作业与数据库服务器运行在同一台设备上，当备份作业运行时就会影响到服务器的正常运行。有时候会明显的降低服务器的性能。同时，将备份数据存放在本地，也 不是很安全。如硬盘因为电压等原因被损坏或者服务器被失窃，此时由于备份文件仍然存放在硬盘上，数据库管理员无法使用备份文件来恢复数据。这显然会给企业 带来比较大的损失。

而如果使用复制来实现对数据的备份，就可以在从服务器上对数据进行备份。此时不仅不会干扰主服务气的正常运行，而且在备份过程中主服务器可以继 续处理相关的更新作业。同时在数据复制的同时，也实现了对数据的异地备份。除非主服务器和从服务器的两块硬盘同时损坏了，否则的话数据库管理员就可以在最 短时间内恢复数据，减少企业的由此带来的损失。

**好处三：提高数据库系统的可用性**

数据库复制功能实现了主服务器与从服务器之间数据的同步，增加了数据库系统的可用性。当主服务器出现问题时，数据库管理员可以马上让从服务器作为主服务器，用来数据的更新与查询服务。然后回过头来再仔细的检查主服务器的问题。此时一般数据库管理员也会采用两种手段。

一是主服务器故障之后，虽然从服务器取代了主服务器的位置，但是对于主服务器可以采取的操作仍然做了一些限制。如仍然只能够进行数据的查询，而 不能够进行数据的更新、删除等操作。这主要是从数据的安全性考虑。如现在一些银行系统的升级，在升级的过程中，只能够查询余额而不能够取钱。这是同样的道理。

二是从服务器真正变成了主服务器。当从服务器切换为主服务器之后，其地位完全与原先的主服务器相同。此时可以实现对数据的查询、更新、删除等操 作。为此就需要做好数据的安全性工作。即数据的安全策略，要与原先的主服务器完全相同。否则的话，就可能会留下一定的安全隐患。

## 问题

主从复制也带来其他一系列性能瓶颈问题：

1、写入无法扩展

2、写入无法缓存

3、复制延时

4、锁表率上升

5、表变大，缓存率下降

针对上述问题，引入数据库垂直分区和水平分区解决。