118 我们为什么要搭建一套MySQL的主从复制架构？（1）

我们为什么要搭建一套MySQL的主从复制架构？（1）

之前的很长时间里，我们经过了大量内容的讲解，想必大家对于MySQL的内核级工作原理已经有了一个了解了，包括我们的数据是如何写入MySQL服务器的内存以及磁盘的，过程中的事务、锁分别是怎么实现的，多事务并发的时候，隔离机制是如何运作的，MVCC的原理是什么，想必大家都有一个较为透彻的理解了。

同时大家现在对MySQL的索引数据结构以及工作原理，包括SQL查询语句的执行原理以及执行计划的分析，以及SQL语句调优的一些技巧和方法，应该也都有了一个较为透彻的理解了

因此简单来说，现在各位同学假设面对一个单机版的MySQL数据库，对于你的数据是如何执行增删改操作写入数据库的，以及你的索引是如何设计的，如何组织的，你的查询是如何执行的，你的查询应该如何优化，都有了一个较为系统全面的理解，而且这个理解是基于MySQL的内核级的原理的，有一定的深度，是不是?

好，那么如果大家感觉自己对上述提问都有一个肯定的回答，说明大家之前的内容肯定都好好学，而且认真复习过，学习的都较为透彻，其实掌握上述内容，就意味着大家对MySQL的原理以及使用掌握的就比较好了。

那么从今天开始，我们将要进入一个全新的阶段，那就是在MySQL真正的生产环境中，他一定不是一个单机版的架构，因为单机版的MySQL一般仅能用于本地开发环境和测试环境，是绝对不可能运用于生产环境的。

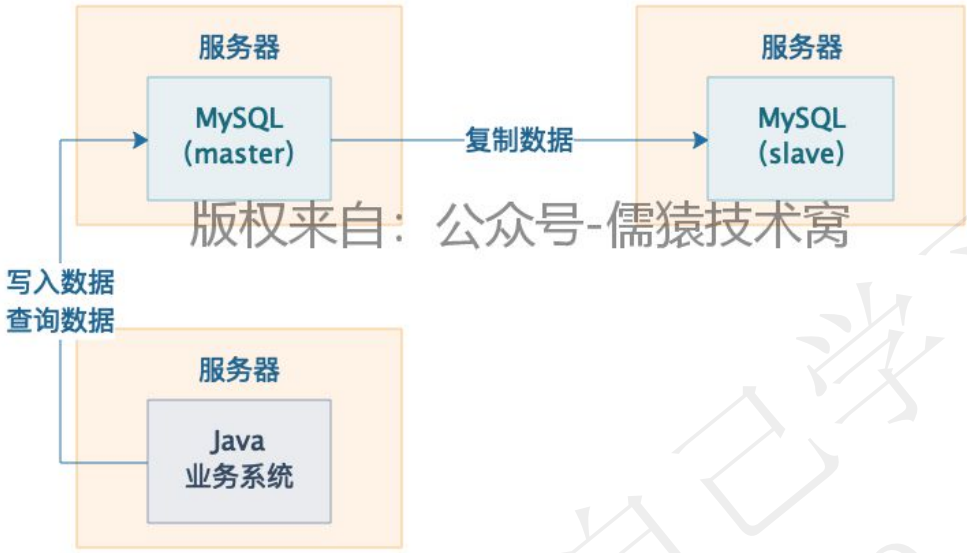
那么生产环境的MySQL架构应该是什么样子的呢？简单来说，MySQL在生产环境中，必须要搭建一套主从复制的架构，同时可以基于一些工具实现高可用架构

另外如果有需求，还需要基于一些中间件实现读写分离架构，最后就是如果数据量很大，还必须可以实现分库分表的架构。

所以当大家把MySQL单机版的内容学完之后，再把后续的这些架构学完，才能说作为一个合格以及优秀的Java工程师/后端工程师，对生产环境下的MySQL架构有了一个全面的理解，能够在自己的生产项目中运用上MySQL的生产级的架构。

那么今天我们就先来给大家讲讲MySQL的主从复制架构，这个主从复制架构，顾名思义，就是部署两台服务器，每台服务器上都得有一个MySQL，其中一个MySQL是master（主节点），另外一个MySQL是slave（从节点）。

然后我们的系统平时连接到master节点写入数据，当然也可以从里面查询了，就跟你用一个单机版的MySQL是一样的，但是master节点会把写入的数据自动复制到slave节点去，让slave节点可以跟master节点有一模一样的数据，如下图所示。

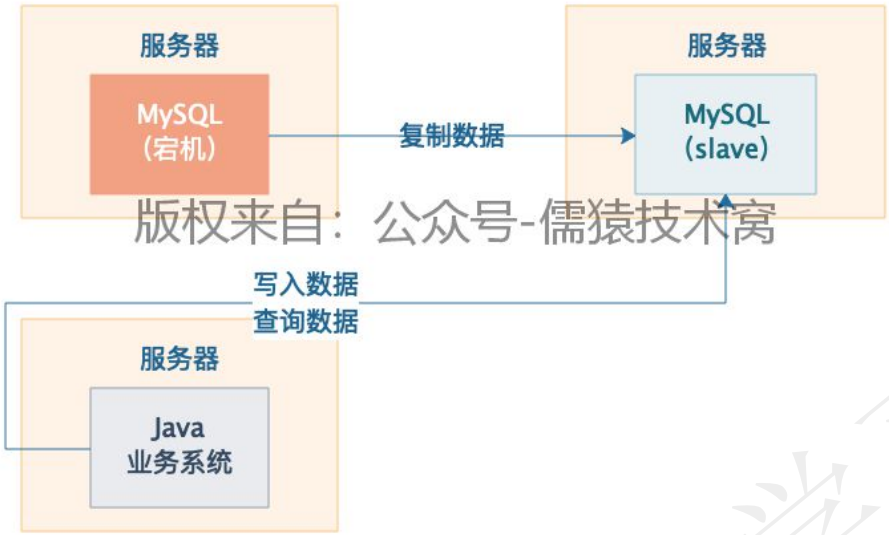


上图其实就是一个典型的MySQL的主从复制架构，那么做这个主从复制架构，意义在哪儿呢？

其实这个架构用处是极为多的，我们来给大家一 一举例，首当其冲的一个需求就是高可用架构。

大家可以想想，如果你的MySQL就单机部署，那么一旦他宕机了，岂不是你的数据库就完蛋了？数据库完蛋了，你的Java业务系统是不是也就完蛋了？所以说，真正生产架构里，MySQL必须得做高可用架构。

那么高可用架构怎么做呢？他的一个先决条件就是**主从复制架构。**你必须得让主节点可以复制数据到从节点，保证主从数据是一致的，接着万一你的主节点宕机了，此时可以让你的Java业务系统连接到从节点上去执行SQL语句，写入数据和查询数据，因为主从数据是一致的，所以这是没问题的，如下图所示。



如果实现这样的一个效果，自然就实现了MySQL的高可用了，他单机宕机不影响你的Java业务系统的运行。但是大家也得注意，这里其实是没这么简单的，因为实际哪怕这套架构运用到生产环境，也是有大量的问题要解决的。

比如主从进行数据复制的时候，其实从节点通常都会落后一些，所以数据不完全一致。另外，主节点宕机后，要能自动切换从节点对外提供服务，这个也需要一些中间件的支持，也没那么容易，这些问题，后续我们都会讲到的。

那么搭建了主从复制架构之后，还有其他什么用处呢？下回我们再继续讲解。

End