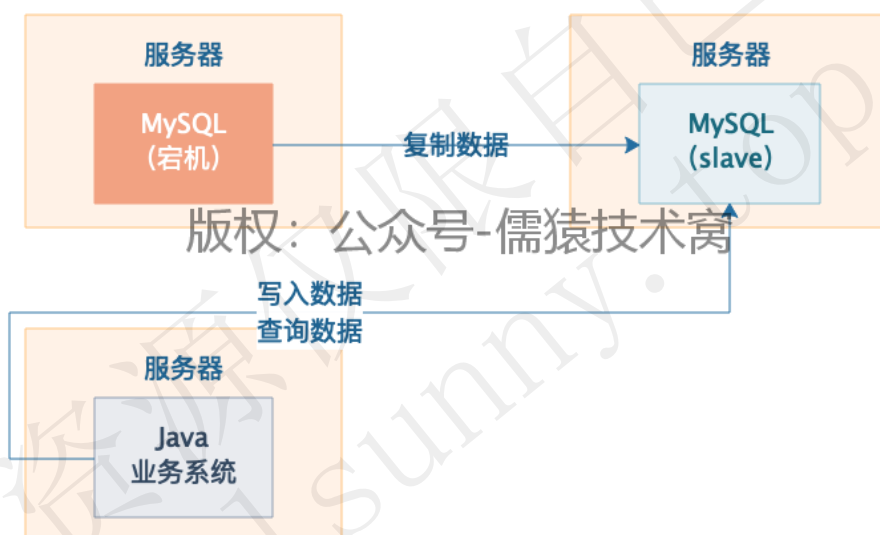


## 我们为什么要搭建一套MySQL的主从复制架构？（2）

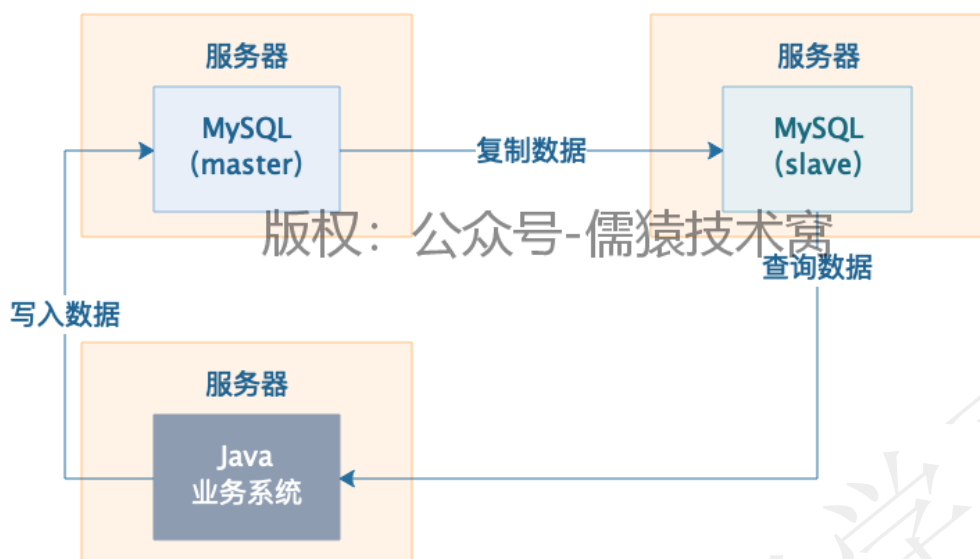
上一次已经讲到，我们搭建一套MySQL主从复制架构之后，可以实现一个高可用的效果，也就是说主节点宕机，可以切换去读写从节点，因为主从节点数据基本是一致的

当然，暂时也就只能说是基本一致的，因为后续大家学习了主从复制的原理之后就知道为什么说是基本了，如下图。



那么我们如果做了这个MySQL主从复制架构之后，除了这个高可用之外，还有什么作用呢？其实这就得说到大名鼎鼎的**读写分离架构**了！这个读写分离架构，也是依赖于MySQL的主从复制架构的。

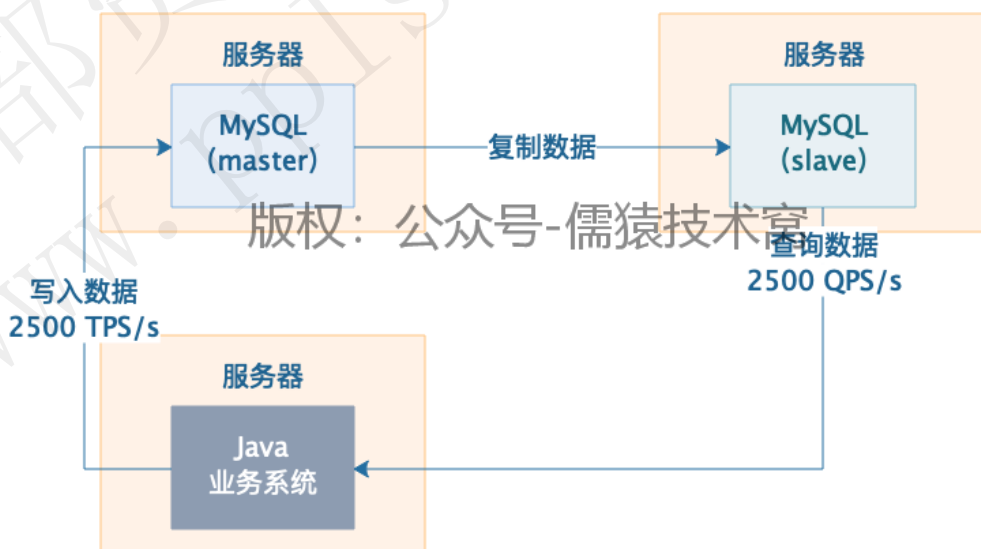
读写分离架构的意思就是，你的Java业务系统可以往主节点写入数据，但是从从节点去查询数据，把读写操作做一个分离，分离到两台MySQL服务器上去，一台服务器专门让你写入数据，然后复制数据到从节点，另外一台服务器专门让你查询数据，如下图所示。



可是好端端的，我们吃饱了没事儿，为什么要做读写分离呢？难道就为了好玩儿吗？

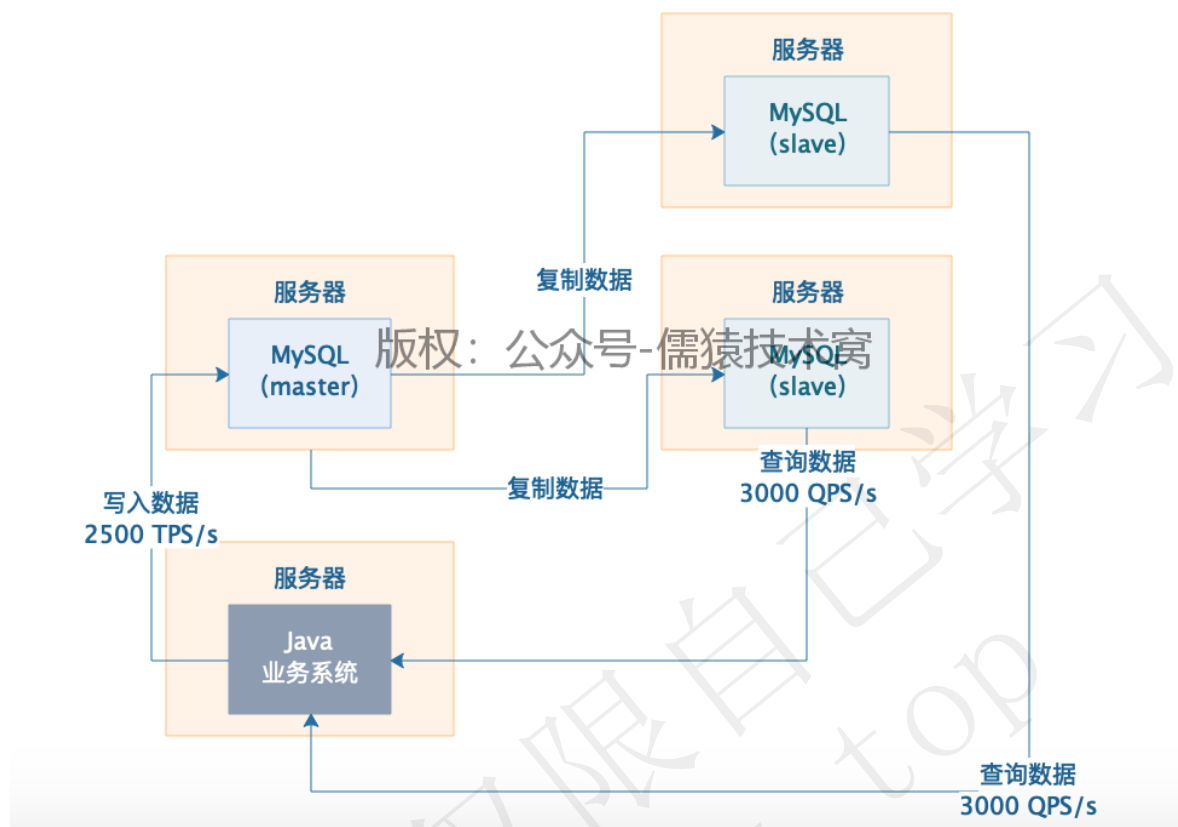
当然不是了！因为假设我们的MySQL单机服务器配置是8核16GB，然后每秒最多能抗4000读写请求，现在假设你真实的业务负载已经到达了，每秒有2500写请求+2500读请求，也就是每秒5000读写请求了，那么你觉得如果都放一台MySQL服务器，能抗得住吗？

必然不行啊！所以此时如果你可以利用主从复制架构，搭建起来读写分离架构，就可以让每秒2500写请求落到主节点那台服务器，2500读请求落到从节点那台服务器，用2台服务器来抗下你每秒5000的读写请求，如下图所示。



接着现在问题来了，大家都知道，其他大部分Java业务系统都是读多写少，读请求远远多于写请求，那么接着发现随着系统日益发展，读请求越来越多，每秒可能有6000读请求了，此时一个从节点服务器也抗不下来啊，那怎么办呢？

简单！因为MySQL的主从复制架构，是支持一主多从的，所以此时你可以再在一台服务器上部署一个从节点，去从主节点复制数据过来，此时你就有2个从节点了，然后你每秒6000读请求不就可以落到2个从节点上去了，每台服务器主要接受每秒3000的读请求，如下图。



如上图，Java业务系统每秒以2500的TPS写入主库，然后主库会复制数据到两个从库，接着你每秒6000 QPS的读请求分散在两个从库上，一切似乎很完美，这就是主从复制架构的另外一个经典的应用场景，就是读写分离，通过读写分离，可以让你抗下很高的读请求。

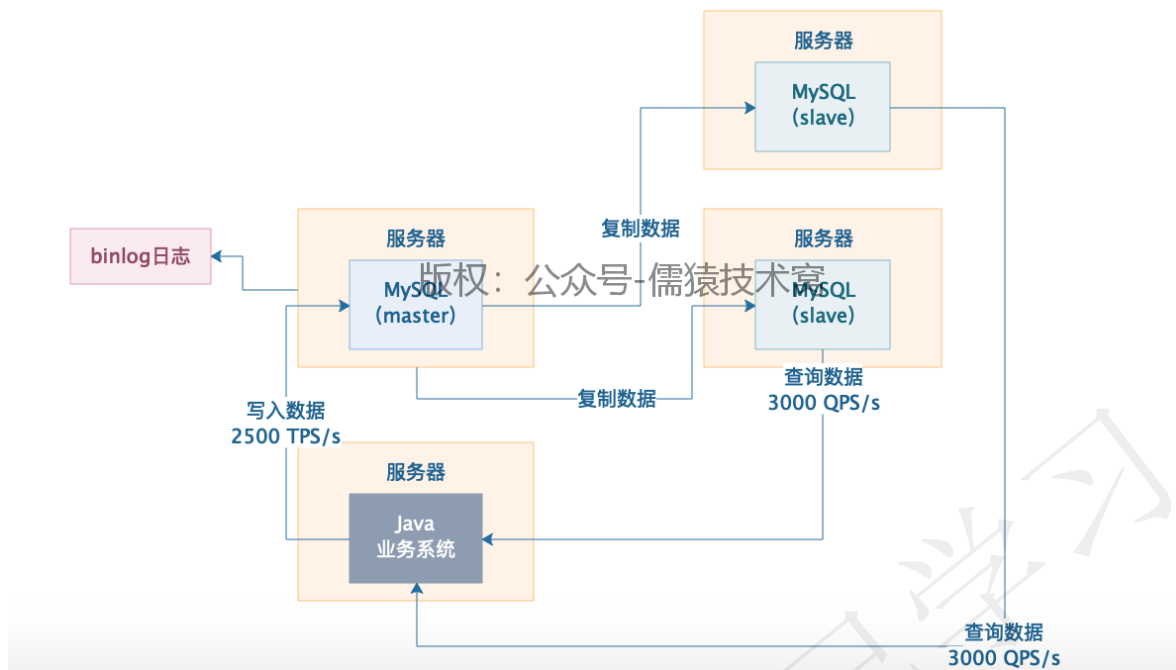
而且在上述架构之下，还可以融合高可用架构进去，因为你有多个从库，所以当你主库宕机的时候，可以通过中间件把一个从库切换为主库，此时你的Java业务系统可以继续运行，在实现读写分离的场景下，还可以同时实现高可用。

不过其实一般在项目中，高可用架构是必须做的，但是读写分离架构并不是必须的，因为对于大多数公司来说，读请求QPS并没那么高，远远达不到每秒几千那么夸张，但是高可用你是必须得做的，因为你必须保证主库宕机后，有另外一个从库可以接管提供服务，避免Java业务系统中断运行。

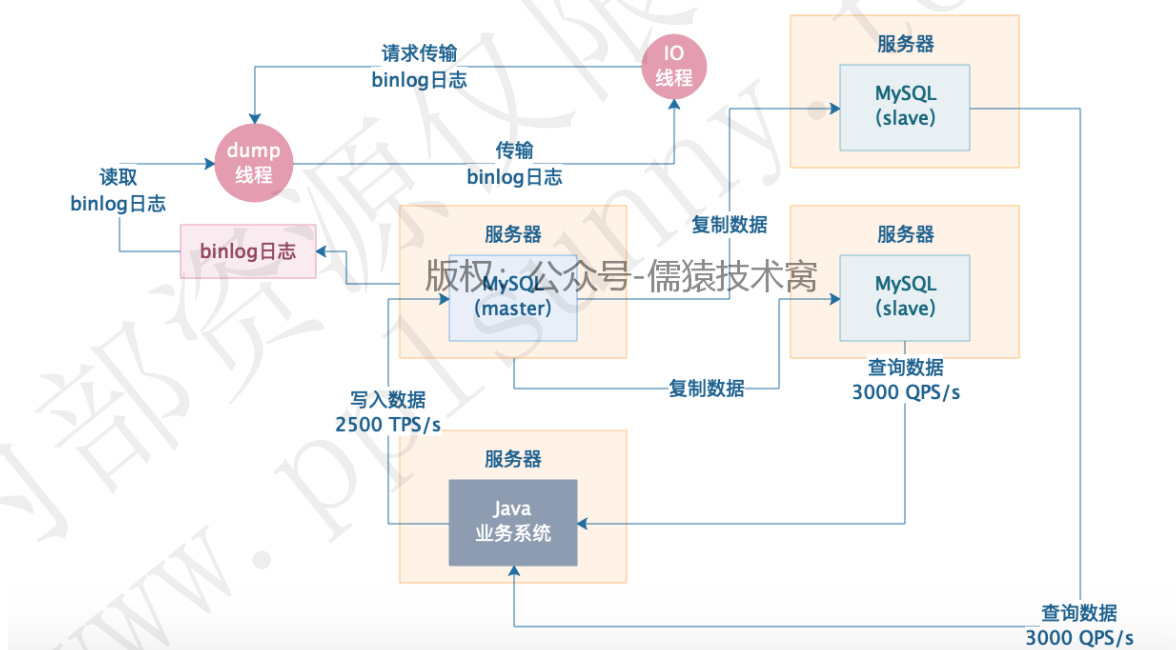
除此之外，这个从库其实还有很多其他的应用场景，比如你可以挂一个从库，专门用来跑一些报表SQL语句，那种SQL语句往往是上百行之多，运行要好几秒，所以可以专门给他一个从库来跑。也可以是专门部署一个从库，让你去进行数据同步之类的操作。

接着我们来说一下MySQL实现主从复制的一个基本的工作原理。

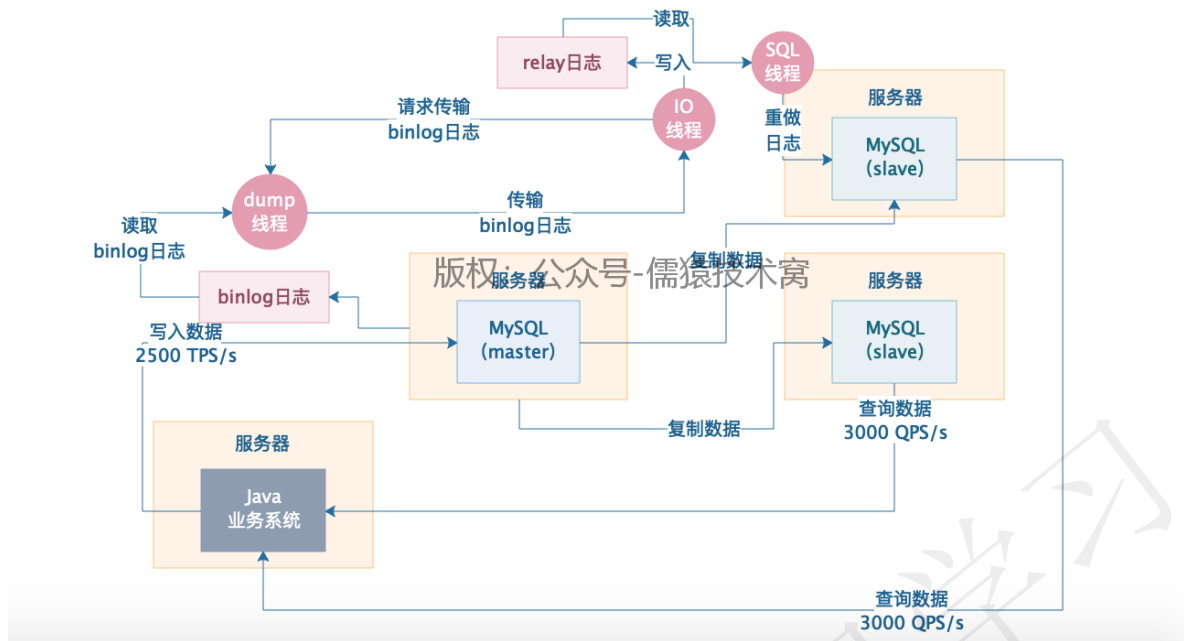
首先呢，大家通过之前的学习，应该都知道，MySQL自己在执行增删改的时候会记录binlog日志，这个大家没问题吧？忘记的同学可以回过头去看看，所以这个binlog日志里就记录了所有数据增删改的操作，如下图。



然后从库上有一个IO线程，这个IO线程会负责跟主库建立一个TCP连接，接着请求主库传输binlog日志给自己，这个时候主库上有一个IO dump线程，就会负责通过这个TCP连接把binlog日志传输给从库的IO线程，如下图所示。



接着从库的IO线程会把读取到的binlog日志数据写入到自己本地的relay日志文件中，然后从库上另外一个SQL线程会读取relay日志里的内容，进行日志重做，把所有在主库执行过的增删改操作，在从库上做一遍，达到一个还原数据的过程，如下图。



到此为止，想必大家对MySQL主从复制的原理也就有一个基本的了解了，简单来说，你只要给主节点挂上一个从节点，从节点的IO线程就会跟主节点建立网络连接，然后请求主节点传输binlog日志，主节点的IO dump线程就负责传输binlog日志给从节点，从节点收到日志后就可以回放增删改操作恢复数据。

在这个基础之上，就可以实现MySQL主从节点的数据复制以及基本一致，进而可以实现高可用架构以及读写分离架构

好了，今天就先讲到这里，下次我们会继续讲解MySQL的各种主从复制模式的搭建方式。

End