### 想用数据库“读写分离” 请先明白“读写分离”解决什么问题

有一些技术同学可能对于“读写分离”了解不多，认为数据库的负载问题都可以使用“读写分离”来解决。

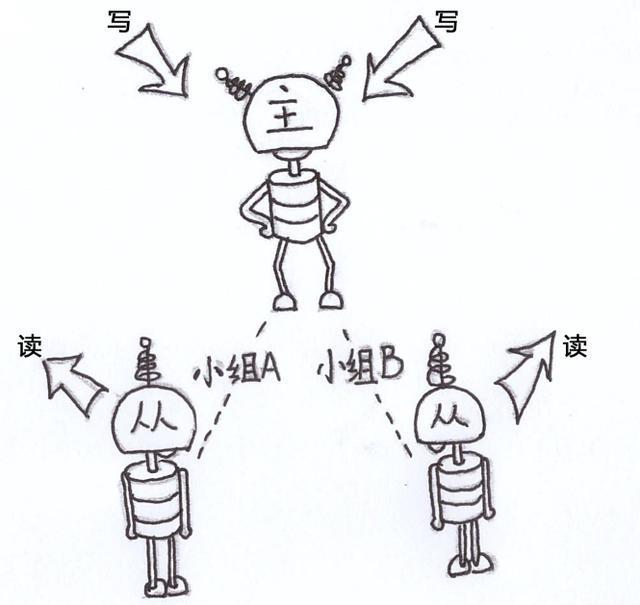


这其实是一个非常大的误区，我们要用“读写分离”，首先应该明白“读写分离”是用来解决什么样的问题的，而不是仅仅会用这个技术。

#### 什么是读写分离？

其实就是将数据库分为了主从库，一个主库用于写数据，多个从库完成读数据的操作，主从库之间通过某种机制进行数据的同步，是一种常见的数据库架构。

一个组从同步集群，通常被称为是一个“分组”。



#### 数据库分组架构解决什么问题？

大多数互联网业务，往往读多写少，这时候，数据库的读会首先称为数据库的瓶颈，这时，如果我们希望能够线性的提升数据库的读性能，消除读写锁冲突从而提升数据库的写性能，那么就可以使用“分组架构”（读写分离架构）。

用一句话概括，读写分离是用来解决**数据库的读性能瓶颈**的。



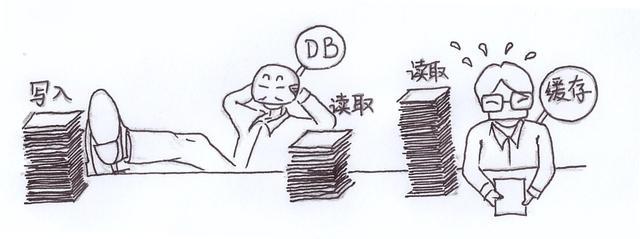
但是，不是任何读性能瓶颈都需要使用读写分离，我们还可以有其他解决方案。

在互联网的应用场景中，常常数据量大、并发量高、高可用要求高、一致性要求高，如果使用“读写分离”，就需要注意这些问题：

数据库连接池要进行区分，哪些是读连接池，哪个是写连接池，研发的难度会增加；为了保证高可用，读连接池要能够实现故障自动转移；主从的一致性问题需要考虑。在这么多的问题需要考虑的情况下，如果我们仅仅是为了解决“数据库读的瓶颈问题”，为什么不选择使用缓存呢？

#### 为什么用缓存

缓存，也是互联网中常常使用到的一种架构方式，同“读写分离”不同，读写分离是通过多个读库，分摊了数据库读的压力，而存储则是通过缓存的使用，减少了数据库读的压力。他们没有谁替代谁的说法，但是，如果在缓存的读写分离进行二选一时，还是应该首先考虑缓存。

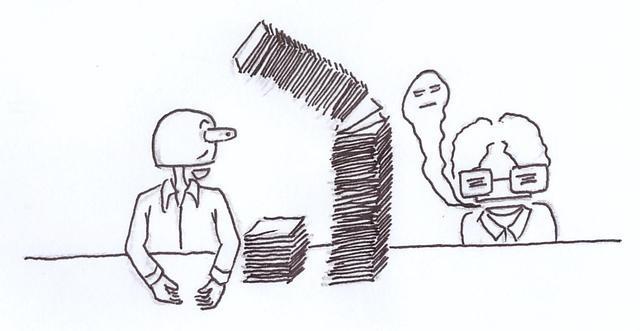


**为什么呢？**

缓存的使用成本要比从库少非常多；缓存的开发比较容易，大部分的读操作都可以先去缓存，找不到的再渗透到数据库。当然，如果我们已经运用了缓存，但是读依旧还是瓶颈时，就可以选择“读写分离”架构了。简单来说，我们可以将读写分离看做是缓存都解决不了时的一种解决方案。

#### 当然，缓存也不是没有缺点的

对于缓存，我们必须要考虑的就是高可用，不然，如果缓存一旦挂了，所有的流量都同时聚集到了数据库上，那么数据库是肯定会挂掉的。



#### 对于常见的数据库瓶颈是什么呢？

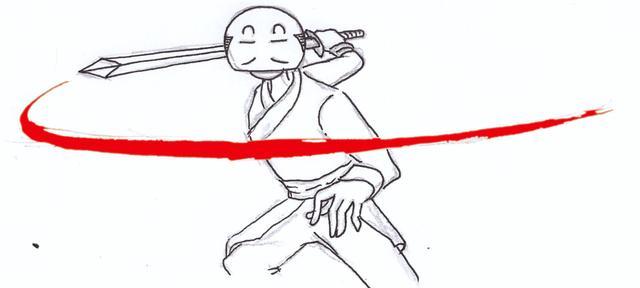
其实是数据容量的瓶颈。例如订单表，数据量只增不减，历史数据又必须要留存，非常容易成为性能的瓶颈，而要解决这样的数据库瓶颈问题，“读写分离”和缓存往往都不合适，最适合的是什么呢？



数据库水平切分

#### 什么是数据库水平切分？

数据库水平切分，也是一种常见的数据库架构，是一种通过算法，将数据库进行分割的架构。一个水平切分集群中的每个数据库，通常称为一个“分片”。每一个分片中的数据没有重合，所有分片中的数据并集组成全部数据。



#### 水平切分架构解决什么问题呢？

大部分的互联网业务，数据量都非常大，单库容量最容易成为瓶颈，当单库的容量成为了瓶颈，我们希望提高数据库的写性能，降低单库容量的话，就可以采用水平切分了。

而有少部分程序员，会没有分析数据库的性能瓶颈是什么，就贸贸然的使用“读写分离”，殊不知“水平切分”才是正道。