

详情 评论

### 生产经验：数据库服务器使用的RAID存储架构初步介绍

**如何提问：**每篇文章都有评论区，大家可以尽情留言提问，我会逐一答疑

**如何加群：**购买狸猫技术窝专栏的小伙伴都可以加入狸猫技术交流群，一个非常纯粹的技术交流的地方

具体加群方式，请参见目录菜单下的文档：《MySQL专栏付费用户如何加群》（购买后可见）

今天我们继续给大家讲解生产环境下的MySQL数据库的一些存储技术的原理，之前已经给大家解释了MySQL的磁盘随机读写和顺序读写的场景和原理，包括Linux操作系统的存储系统的原理，那么我们接着就要继续讲解Linux操作系统再底层的存储硬件层面的一些原理

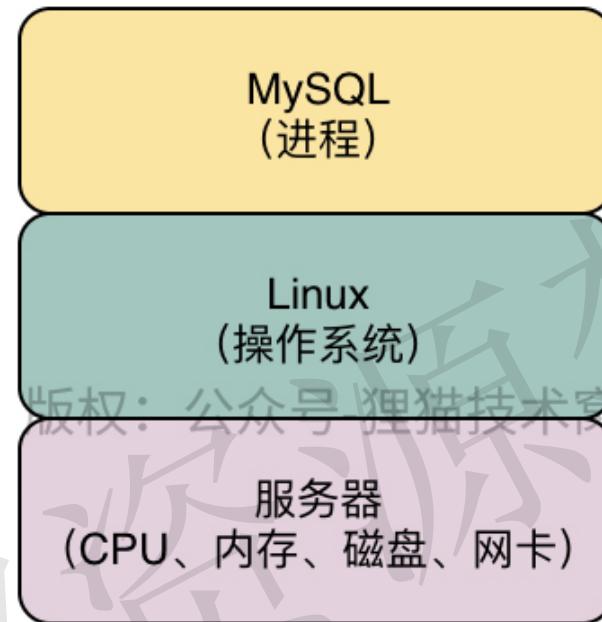
只有把这些都理解透了，才能给大家最终讲清楚一次线上生产环境的MySQL数据库的性能抖动的故障原因。

实际上MySQL数据库就是个软件，大家都知道，他其实就是用编程语言写的一套数据库管理软件而已，底层就是磁盘来存储数据，基于内存来提升数据读写性能，然后设计了复杂的数据模型，帮助我们高效的存储和管理数据。

所以MySQL数据库软件都是安装在一台linux服务器上的，然后启动MySQL的进程，就是启动了一个MySQL数据库

MySQL运行过程中，他需要使用CPU、内存、磁盘和网卡这些硬件，但是不能直接使用，都是通过调用操作系统提供的接口，依托于操作系统来使用和运行的，然后linux操作系统负责操作底层的硬件。

我们下面画了一个图，来给大家表示一下，他们之间的一个关系。



所以之前我们已经把MySQL层面的磁盘读写操作讲完了，同时也把Linux操作系统层面的存储系统的原理讲完了，上一讲不就讲到Linux操作系统的存储系统把IO请求交给机器上的存储硬件了么？

那么今天我们接着来讲讲存储硬件这块的东西。

一般来说，很多数据库部署在机器上的时候，存储都是搭建的RAID存储架构，其实这个RAID很多人以为非常的深奥，确实这个概念比较难以理解，而且说深了其实里面的技术含量很高，但是如果简单说一下，也是每个人都能理解的。

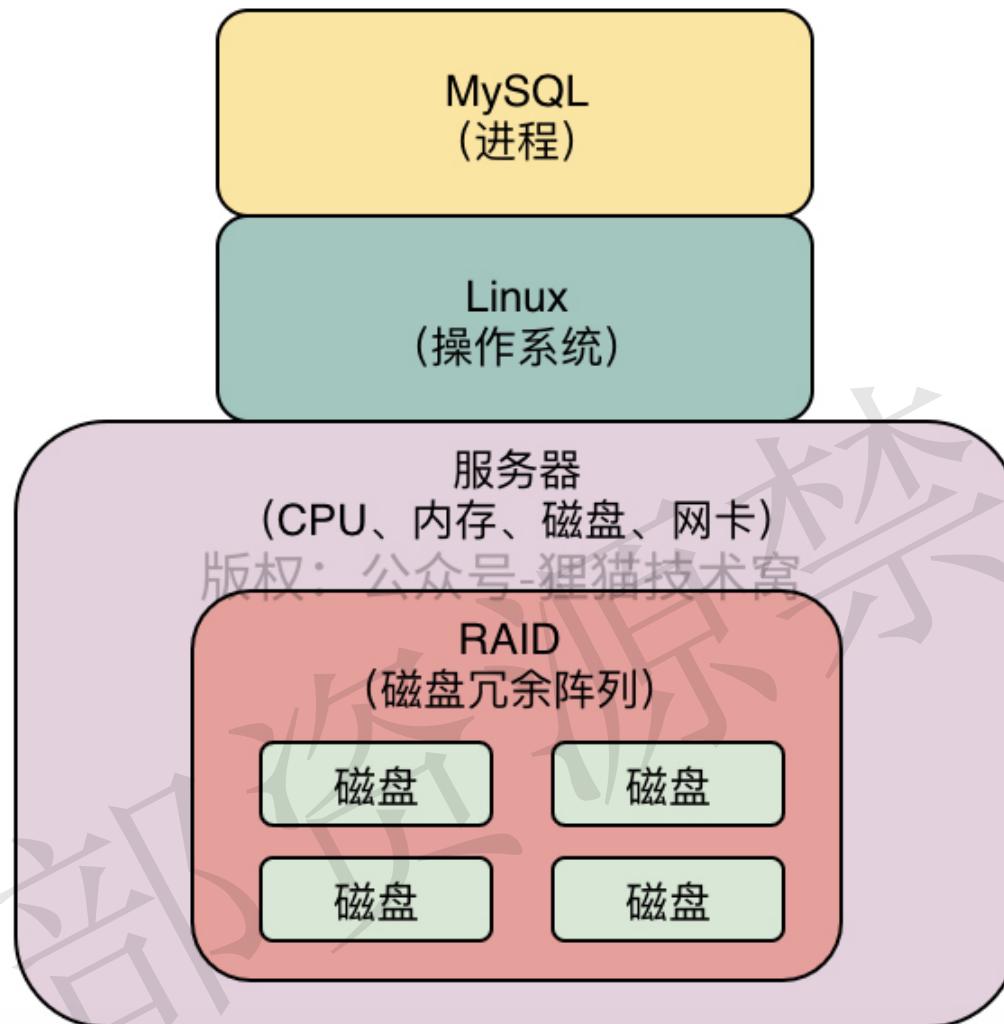
说白了，RAID就是一个磁盘冗余阵列，什么意思呢？

假设我们的服务器里的磁盘就一块，那万一一块磁盘的容量不够怎么办？此时是不是就可以再搞几块磁盘出来放在服务器里

现在多搞了几块磁盘，机器里有很多块磁盘了，不好管理啊，怎么在多块磁盘上存放数据呢？

所以就是针对这个问题，在存储层面往往会在机器里搞多块磁盘，然后引入RAID这个技术，大致理解为用来管理机器里的多块磁盘的一种磁盘阵列技术！

有了他以后，你在往磁盘里读写数据的时候，他会告诉你应该在哪块磁盘上读写数据，如下图。

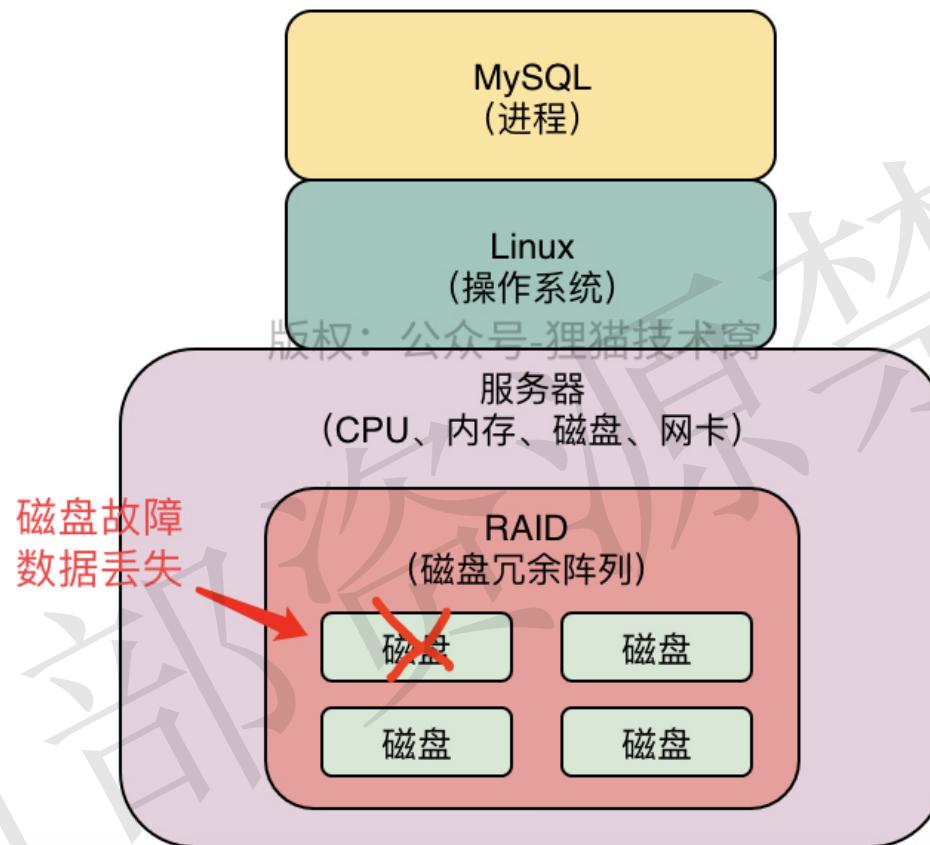


有了RAID这种多磁盘阵列技术之后，我们是不是就可以在一台服务器里加多块磁盘，扩大我们的磁盘存储空间了？

当我们往磁盘里写数据的时候，通过RAID技术可以帮助我们选择一块磁盘写入，在读取数据的时候，我们也知道从哪块磁盘去读取。

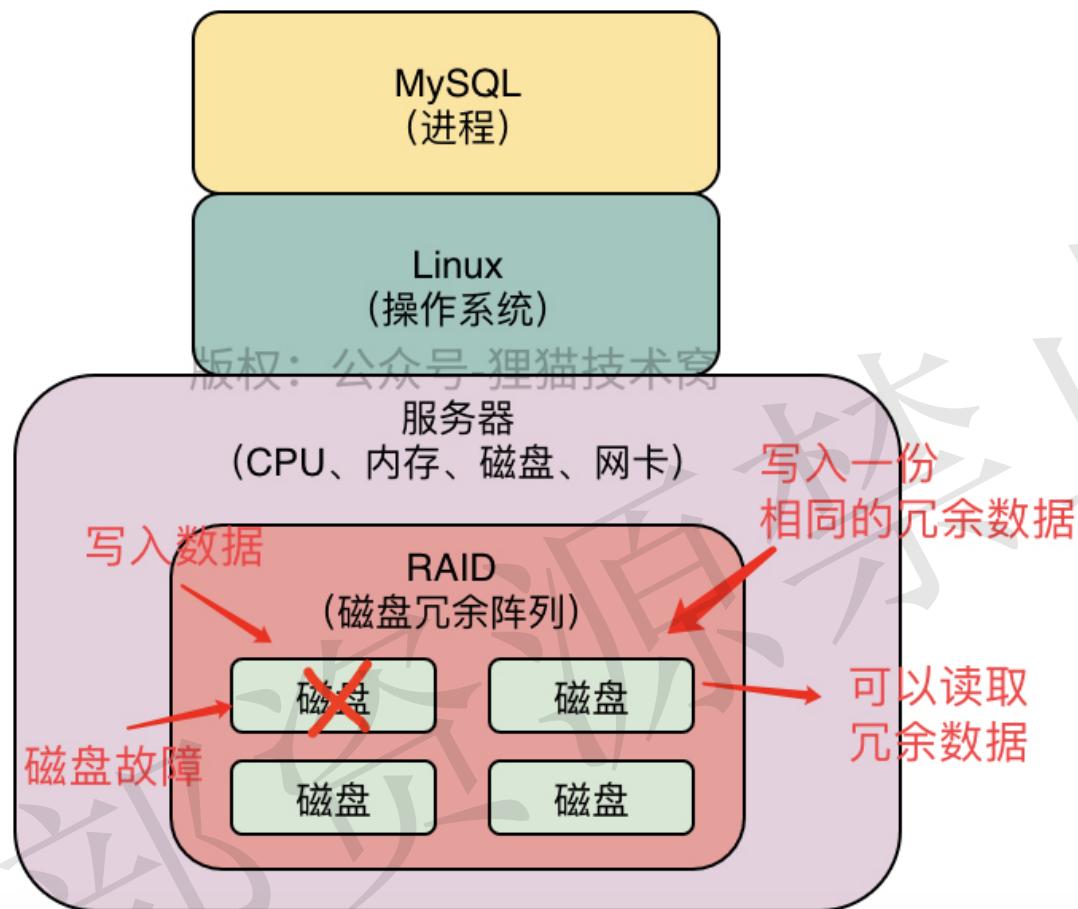
除此之外，RAID技术很重要的一个作用，就是他还可以实现**数据冗余机制**

所谓的数据冗余机制，就是如果你现在写入了一批数据在RAID中的一块磁盘上，然后这块磁盘现在坏了，无法读取了，那么岂不是你就丢失了一波数据？如下图所示



所以其实有的RAID磁盘冗余阵列技术里，是可以把你写入的同样一份数据，在两块磁盘上都写入的，这样可以让两块磁盘上的数据一样，作为冗余备份，然后当你一块磁盘坏掉的时候，可以从另外一块磁盘读取冗余数据出来，这一切

都是RAID技术自动帮你管理的，不需要你操心，如下图。



所以RAID技术实际上就是管理多块磁盘的一种磁盘阵列技术，他有软件层面的东西，也有硬件层面的东西，比如有RAID卡这种硬件设备。

具体来说，RAID还可以分成不同的技术方案，比如RAID 0、RAID 1、RAID 0+1、RAID2，等等，一直到RAID 10，很多种不同的多磁盘管理技术方案。

大家如果有兴趣的，可以自行去搜索对应的资料学习里面的技术细节，但是对于我们来说，这篇文章点到为止

**大家只要了解一下RAID这种多磁盘冗余阵列技术的基本思想就可以了，我们毕竟不是专门讲解存储这块的。**

对于存储的深入学习，主要也是一些运维工程师会去做的。

End

专栏版权归公众号**狸猫技术窝**所有

未经许可不得传播，如有侵权将追究法律责任

---

狸猫技术窝精品专栏及课程推荐：

- [《从零开始带你成为消息中间件实战高手》](#)
- [《21天互联网Java进阶面试训练营》（分布式篇）](#)
- [《互联网Java工程师面试突击》（第1季）](#)
- [《互联网Java工程师面试突击》（第3季）](#)
- [《从零开始带你成为JVM实战高手》](#)