

数据库的使用你可能忽略了这些 (续)

原创：IT米粉 IT米粉 8月23日

前言

之前写过一篇文章《数据库的使用你可能忽略了这些》，主要是从一些大家使用使用时容易忽略的地方，如：字段长度、表设计等来说明，这篇文章同样也是这样的主题，只是从另外的几个方面来说说数据库使用中，容易忽略，导致入坑的地方。

合理预估数据量

在数据库进行表设计的时候，就应该评估可能产生的数据量，数据量会对整个开发和代码的健壮性有很大的影响。开发一个数据量万级别、十万级别、百万级别、千万以上级别数量的应用，在开发思路、技术选型、架构都能都要很大的差别。

基本上的我的原则是：

- 万级别的数据库，可以随意一点，SQL编写有好的习惯；
- 十万级别，注意索引，注意联表性能；
- 百万级别，尽量减少联表，尽量不要做汇总查询，如查总数；
- 千万以上级别，除缓存之外，使用分表分库；

很多系统因为在设计表的时候，没有很好的预估的后期系统的发展，导致上线不久就出现无法支撑的情况，代码上太多的联表查询，不在乎基础的SQL性能，导致数据库的瓶颈很快就显现出来，不得不重构系统。设计数据库的时候，一定是基于业务进行设计的，对业务的发展有一定的预估，看得长远一点。

合理预估并发访问量

数据库有天然的瓶颈，就是并发量。我们一般会通过缓存来减少数据库的并发连接，以及对数据库的操作，数据库的并发，不是只有大型平台才会遇到，很多中小平台其实也会面临这样的问题，例如：

循环进行数据库的操作

这个问题，上一篇文章我也提到过，不要在循环里进行数据库的操作，这个会直接导致数据库连接数暴增，影响非常严重。虽然是个比较低级的问题，但是出现的概率其实是非常高的，在我身边看到很多很多这种案例了，这种问题，就是需要程序员自己本身避免这些问题，当然，也可以通过一些手段去监控，找到这些问题，只是会比较麻烦一点。

业务本身的高频次数据请求

其实有些业务，即使是中小型的平台，也会有高并发请求数据库的情况，常见的例子如：日志。例如，我们需要抓取到所有人的操作日志，或者所有模块的加载时间，并且持久化保存。如果，当初选型通过Mysql去记录这些数据，那么就很容易遇到高并发的问題。这种就是属于选型的错误了。

数据库对高并发的处理一直是短板，所以应该尽量避免高并发的数据库操作，查询通过缓存处理，增删改这可以通过MQ或者Kafka这样的工具异步进行处理，如果对数据库的结构化要求不高，则可以用hbase或者hive进行数据库的保存。

数据库线程池的合理使用

现在数据库的操作都是使用线程池的，线程池主要是用来控制数据库的连接数，其实连接池是不属于数据库范畴，但是，一般我们使用和数据库结合非常紧密，所以在这里一并说明。

一般线程池都会有这样的几个参数：

参数	说明
最小连接数	不管是否有数据库的操作，这几个连接都会一直存在，

参数	说明
最大连接数	允许的最大的连接数，如果超过了这个数据，则无法申请连接，只能等待，或者异常
回收时间	多长时间会对所有的连接进行一次断开，然后重新连接。
释放时间	多长时间没有进行操作的连接，会释放

基本所有的连接池都会有这几个参数，可能不同的连接池参数名不同，但是作用是一样的。这里我们重点说一下最大连接数，这个是很容易忽略的一个设置。

很多人设置最大连接数的时候，喜欢设置的很大，例如设置为5000，但是一般mysql的数据库一个实例连接默认才1000，连接数超过这个了数据库也无法处理，设置的再大其实是没有用的。

服务器数量 * 最大连接数 < 数据库最大连接数

而且，这还是在一个实例，一个数据库的情况下，至于多个数据库：

我建议

服务器数量 * 最大连接数 * 数据库数量 < 数据库最大连接数

如果单个数据库占用了太多的数据库连接，会影响到其他数据库，导致其他数据库也无法使用。

当然，这个值大家可以根据业务去进行合理的估算，高频的业务分配多一点，低频的业务分配少一点。不要盲目的一味设置连接池的最大值。

总结

如今，虽然各种各样的存储方式出现，但是关系数据库一直是我们系统的最重要的组成部分，尽量不要过早暴露数据库应对并发的短板，设计数据库和操作数据库在我们的开发中应该是一件很神圣的事情，认证对待关系的数据库的每一个操作才是明智之举。

扩展阅读：

数据库的使用你可能忽略了这些

学会数据库读写分离、分表分库——用Mycat，这一篇就够了！