

【订单领域】如果订单要分库分表，如何确认最佳库表数量？

来自：芋道快速开发平台 Boot + Cloud



随着业务发展，订单单库单表到达性能瓶颈，需要进行分库分表。

分库分表需要重点考虑 sharding key 分片键、库表数量。一般情况下，订单可以使用 id、user\_id、merchant\_id 作为 key，这点我们已经在《订单日增 1000w+ 单，如何实现海量数据存储？如何实现高并发、高性能读写？》分享过，就不重复讲解。

因此，本帖的重心放在**如何计算库表数量**。需要考虑的因素有：单表性能、硬盘容量、在线 DDL 耗时、数据库连接数。

1、单表性能

这是最关键的因素。MySQL 单表最大记录数，行业内没有定论，有说 500w 的，有说 2000w 的。本质的原因，数据量过大，会超过 B+ 树的建议层高 3-4 层，影响 IO 性能。不了解的球友，建议阅读《为什么生产环境中 B+ 树的高度总是3-4层？》文章：

总的来说，我们可以设置一个标准：

- 优秀：500w ~ 1000w
- 良好：1000w ~ 2000w
- 较差：2000w ~ 5000w

基于这个标准，我们期望分库分表后，**单表控制在 1000w 以内，可进（优秀）可退（良好）**。

现在，我们来定义下这次分库分表的目标：

- 单表控制在 1000w 以内
- 日增订单量 100w
- 满足近 1 年的用户订单存储

那么，1 年总的订单量是 365 天 x 100w = 36500w 单。计算下，需要多少个表如下：

	500w	1000w	2000w
总共表数	73 个	36.5 个	18.25 个

这样的话，我们可以考虑分 4 个库，每个库 8 个表，基本控制在 1000w 单，运维成本也可控。

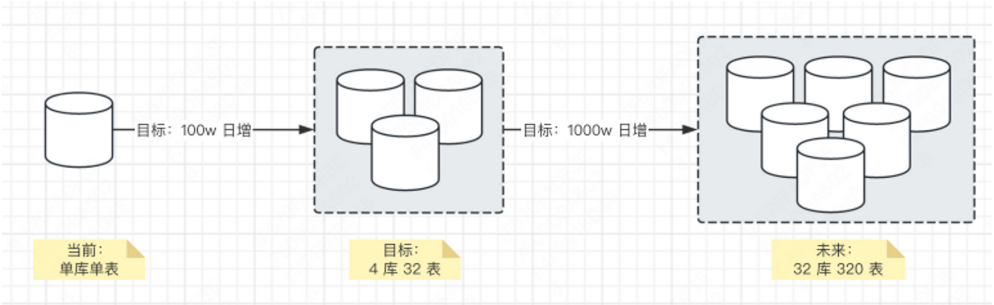
问：为什么不考虑存量订单，只考虑增量订单？

答：这个主要看场景，在较大的订单日增下，存量基本可以忽略不计，例如说上面这个案例。

问：会不会存在 1 年后，现有分库分表又达到瓶颈？

答：首先，确实存在这种可能性，但是尽量不要过度设计，硬件是有成本的。并且我们在做上面这个案例设计时，虽然定的目标是日增 100w 订单，但是可能当下日增实际值有 10w 左右，提前做的技术预研。

其次，架构是持续迭代的，如果真碰到日增从 100w 朝着 500w 或者 1000w 时，恭喜你，继续做下一轮拆分吧。如下图所示：



另外，订单是有明显冷热之分的，超过 90 天的订单往往就不再查询和变更，会考虑归档订单历史库。例如说：某东、每日优鲜、淘宝、收钱吧等公司。这样，我们可以进一步控制订单表的单表大小。

2、在线 DDL 耗时

对 MySQL 大表的 DDL 操作时，一般放在业务低峰期，往往是半夜，避免可能的锁表影响业务。在我们把单表控制在 1000-2000w 记录以内的话，半夜 6 小时内肯定可以完成。

另外，也可以使用一些开源的工具来进行 DDL 操作，尽可能避免锁表。例如说：pt-online-schema-change、gh-ost 等等。

3、数据连接数

分库分表后，单个数据库实例会有多个库，因此 MySQL 的连接数会相应的增加。不过一般情况下，也就乘以几倍，基本问题不大，可以不用考虑太多。

4、硬盘容量

需要预留 20% 硬盘容量，避免数据继续增长，数据库无法写入。例如说：华为云 RDS 的“磁盘空间满”情况：

随着业务数据的增加，原来申请的数据库磁盘容量可能会不足，需要为云数据库RDS实例进行扩容。实例扩容的同时备份空间也会随之扩大。  
1TB以下磁盘空间使用率达到95%，或1TB及以上磁盘空间使用量剩余50GB，实例将显示“磁盘空间满”，此时数据库不可进行写入操作。您需要扩容至磁盘空间使用率小于85%才可使实例处于可用状态，使数据库恢复正常的写入操作。

虽然比较少出现这种情况，但是还是要稍微注意下，= = 我就踩过这样的生产事故。

**题外话：**如果订单日增 5000w，你会拆分多少库表呢？可以分别计算保留 90 天、180 天、365 天不同的结果哟~

“学习是一个输入输出的过程，一定要勤动手，动手才会动脑”

仅供参考，欢迎评论区留言，一起讨论。