

# 如何设计可以动态扩容缩容的分库分表方案

## 停机扩容（不推荐）

- 这个方案就跟停机迁移一样，步骤几乎一致，唯一的一点就是那个导数的工具，是把现有库表的数据抽出来慢慢倒入到新的库和表里去。但是最好别这么玩儿，有点不太靠谱，因为既然分库分表就说明数据量实在是太大了，可能多达几亿条，甚至几十亿，你这么玩儿，可能会出问题。
- 从单库单表迁移到分库分表的时候，数据量并不是很大，单表最大也就两三千万。那么你写个工具，多弄几台机器并行跑，1小时数据就导完了。这没有问题。
- 如果 3 个库 + 12 个表，跑了一段时间了，数据量都 1~2 亿了。光是导 2 亿数据，都要导个几个小时，6 点，刚刚导完数据，还要搞后续的修改配置，重启系统，测试验证，10 点才可以搞完。所以不能这么搞。

## 优化后的方案

- 一开始上来就是 32 个库，每个库 32 个表，那么总共是 1024 张表。
- 我可以告诉各位同学，这个分法，第一，基本上国内的互联网肯定都是够用了，第二，无论是并发支撑还是数据量支撑都没问题。
- 每个库正常承载的写入并发量是 1000，那么 32 个库就可以承载  $32 * 1000 = 32000$  的写并发，如果每个库承载 1500 的写并发， $32 * 1500 = 48000$  的写并发，接近 5万/s 的写入并发，前面再加一个MQ，削峰，每秒写入 MQ 8 万条数据，每秒消费 5 万条数据。
- 有些除非是国内排名非常靠前的这些公司，他们的最核心的系统的数据库，可能会出现几百台数据库的这么一个规模，128个库，256个库，512个库。
- 1024 张表，假设每个表放 500 万数据，在 MySQL 里可以放 50 亿条数据。
- 每秒的 5 万写并发，总共 50 亿条数据，对于国内大部分的互联网公司来说，其实一般来说都够了。
- 谈分库分表的扩容，第一次分库分表，就一次性给他分个够，32 个库，1024 张表，可能对大部分的中小型互联网公司来说，已经可以支撑好几年了。
- 一个实践是利用  $32 * 32$  来分库分表，即分为 32 个库，每个库里一个表分为 32 张表。一共就是 1024 张表。根据某个 id 先根据 32 取模路由到库，再根据 32 取模路由到库里的表。
- 

orderId	id%32(库)	id/32%32(表)
259	3	8
1189	5	5
352	0	11
4593	17	15

- 刚开始的时候，这个库可能就是逻辑库，建在一个数据库上的，就是一个mysql服务器可能建了 n 个库，比如 32 个库。后面如果要拆分，就是不断在库和 mysql 服务器之间做迁移就可以了。然后系统配合改一下配置即可。
- 比如说最多可以扩展到32个数据库服务器，每个数据库服务器是一个库。如果还是不够？最多可以扩展到 1024 个数据库服务器，每个数据库服务器上面一个库一个表。因为最多是1024个表。
- 这么搞，是不用自己写代码做数据迁移的，都交给 dba 来搞好了，但是 dba 确实是需要做一些库表迁移的工作，但是总比你写代码，然后抽数据导数据来的效率高得多吧。
- 哪怕是要减少库的数量，也很简单，其实说白了就是按倍数缩容就可以了，然后修改一下路由规则。

#### 这里对步骤做一个总结：

- 设定好几台数据库服务器，每台服务器上几个库，每个库多少个表，推荐是 32库 \* 32表，对于大部分公司来说，可能几年都够了。路由的规则， $orderId \text{ 模 } 32 = \text{库}$ ， $orderId / 32 \text{ 模 } 32 = \text{表}$  扩容的时候，申请增加更多的数据库服务器，装好 mysql，呈倍数扩容，4 台服务器，扩到 8 台服务器，再到 16 台服务器。由 dba 负责将原先数据库服务器的库，迁移到新的数据库服务器上去，库迁移是有一些便捷的工具的。我们这边就是修改一下配置，调整迁移的库所在数据库服务器的地址。重新发布系统，上线，原先的路由规则变都不用变，直接可以基于 n 倍的数据库服务器的资源，继续进行线上系统的提供服务。