

database migration

数据库迁移

Kanner

what

- 顾名思义，在A机器对数据库做的变更，需要对B机器产生同样的变更。

why

- 避免人为的操作数据库所带来的失误，特别是操作正式库时
- 保证不同的数据库实例，它们之间的结构（包括表结构、存储引擎、权限等等）是完全一致的

how

- 简单的说，就是将执行数据库变更的所有sql脚本以版本变更的方式管理起来，然后再执行。
- 特点：有序性，原子性，一次性。

“选择什么样的工具，决定了你的生产力水平”

–Kanner

数据库性能优化

场景

- 系统只有一个数据库实例，所有的数据存储都只存储在这一个数据库
- 某一天，用户量激增，这时候数据库成为访问瓶颈，为了适应剧增的用户量，需要提升数据库的性能

怎么办？

方案： 横向扩容

- 单机——>一主三从
- 一主三从——>多主多从

横向扩容的本质

- 横向扩容的方式其实并没有提升我们每台数据库服务器能承受的访问量
- 原因： 每台数据库实例存储的数据都是一样的，也就是代表数据量的一样多的。所以，不管是在哪种架构上（单机、主从、集群），执行同样的sql查询，耗时都是一样的

方案：纵向扩容

- 拆库，根据业务将原先的单一数据库拆成多个数据库
- 拆表，将存储量大的表拆成多个小表
- 需要大量变动上层数据库操作的代码

“人生苦短，我用工具”

–Kanner