

垂直与水平拆分设计原理

垂直拆分：

1.数据库拆分主要指分库分表，其目的主要是分散数据库压力，达到横向扩展，满足均衡访问等。

2.数据库拆分主要有两种方式：垂直拆分和水平拆分。

A.垂直拆分：

将不同业务功能相关的表放到不同的数据库中 也就是类似于 微服务架构中 会员数据库/订单数据库/支付数据库

数据整合就需要通过 RPC 接口通讯

例如 当当网（卖书）

<http://static.mayikt.com/dagang.html>

水平拆分：

B.水平拆分：

将一张表数据拆分 n 多张不同的子表来进行存放

考虑扩容的问题

考虑查询的问题

考虑主键问题（序列）

考虑一张表最多存放多少数据（）

当一张表的业务量行数如果超过 500 万行（阿里巴巴 java 开发手册官方推荐），分页/排序效率还是非常低，可以对同一张表数据实现拆分放到多个不同的表中存放。

分表分库之后 如何查询的呢？

1. 整合数据库中间件 mycat 或者 shardingjdbc;
2. 客户端代码中将表的名称替换成虚拟表的名称(虚拟表的名称是被我们数据库中间件改写的)，客户端发送 JDBC 语句 会被数据库中间件进行拦截
3. 获取到 条件后面的 分片字段 如果获取成功 则根据该分片字段 计算 直接定位到该表----只会查询一次。
4. Sql 后面没有带上条件 分片字段 对每张表都会做操作，在将该结果给数据库中间件整合后返回给客户端。
建议最好查询的过程中 带上分片字段，不然对每张都会查询一次。

千万级数据分表分库策略有哪些？

回答：

数据的插入：

1. 定义一个全局的 id，根据该全局 id 采用分片算法，计算该条数据应该落地存放到具体那张表。
2. 该全局 id 特点：有序且递增 推荐序列

- 1.取余/取模
 - 2.按照范围分片
 - 3.按照日期进行分片
 - 4.按照枚举值分片
 - 5.二进制取模范围分片
 - 6.一致性 hash 分片
 - 7.按照目标字段前缀指定的进行分区
 - 8.按照前缀 ASCII 码和值进行取模范围分片
- 一张表的数据达到 500 万条 开始分表分库

分表分库之后分页如何查询

核心点 还是看分片字段 如果 sql 语句有带上分片字段的话 则只会对一张表来进行分页查询。

如果 sql 语句没有带上 分片字段的话 则会对每张表来做分页查询 将该结果交给我们的数据库中间件来整合 二次分页之后返回给客户端。

单表达到多大量开始分表分库

【推荐】单表行数超过 500 万行或者单表容量超过 2GB，才推荐进行分库分表。

说明：如果预计三年后的数据量根本达不到这个级别，请不要在创建表时就分库分表。

摘自：阿里巴巴 java 开发手册

hash 分表算法分表存在哪些优缺点

缺点：

以后无法支持新增表的扩容

优点：

将数据 可以分摊的形式存放 均匀的形式存放

前提：需要保证用户的 userid 连续 计算 存放 序列或者 设置每张表的步长

User0 1 3 5 7 9

User1 2 4 6 8 10

Uuid

Fajfsaksjfafzxfassaf

Saffsaffsafasfsafsa

索引

根据范围的形式分片

假设一张表存放的数据 500 万

以每张 500 万条数据的形式来进行分表

User0 0-500 万条数据

User1 500-1000 万条数据

User2 1000 1500 万条数据

Kafka 的日志存储

根据日期的形式分片

应用场景在哪些？

例如：订单

例如 根据每天 或者 每年
数据存储不均匀 淡季 旺季

支持扩容

订单号码 带上日期规则

20251010

查询历史数据

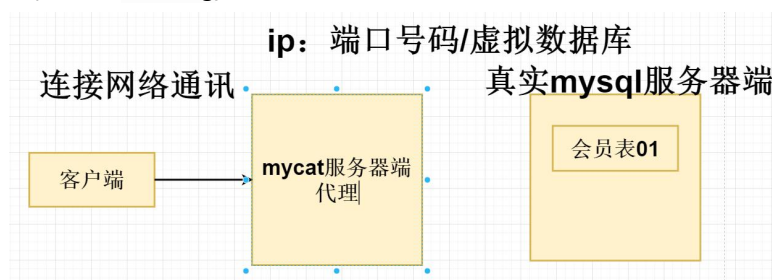
mycat 或者 Shardingjdbc

Mycat 是基于服务器端实现代理；

Shardingjdbc 是基于客户端改写 sql 语句代理；

相对于来说 Shardingjdbc 效率比 Mycat 高。

Mycat 比 Shardingjdbc 更加安全的



如果我们 sql 语句没有带上分页字段，建议 sql 语句后面 带上
Limit 分页查询 避免数据内存溢出

每特教育