World of Tech WOT

51CTO传媒

WOT 2015 互联网运维与开发者大会

■ 2015年04月10日-11日 ■ 北京珠三角JW万豪酒店

58同城mysql分库分表实践

技术中心-沈剑
shenjian@58.com

关于我-@58沈剑

- 前百度高级工程师
- 58同城技术委员会主席，高级架构师
- 58同城优秀讲师
- @58沈剑

目录

- 基本概念
- Mysql大数据量下，常见问题与解决方案
- Mysql分库分表实战
- Mysql分库分别后业务实现
- 总结

一、基本概念

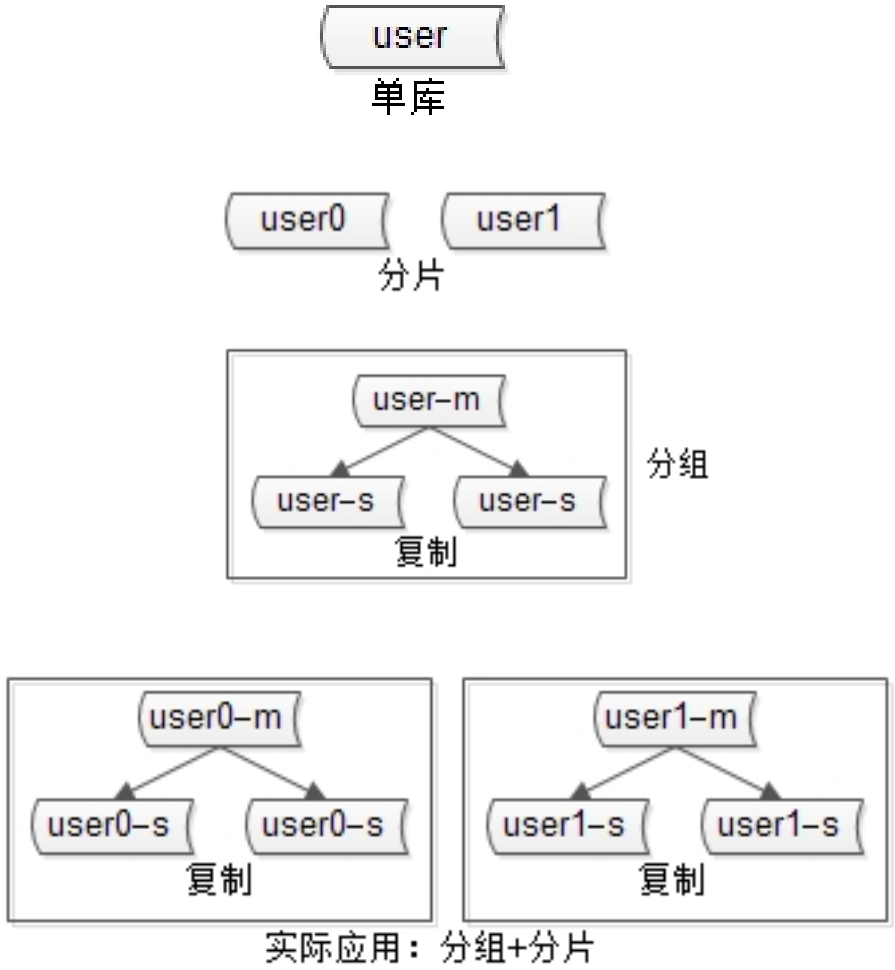
基本概念

- **分片**：sharding
- **复制**：replication
- **分组**：group
- **路由规则**：router rule
- 常用路由方法 + 优缺点 + 扩展性

（1）范围：range

（2）哈希：hash

（3）路由服务：router-config-server



二、Mysql大数据量常见问题与解决方案

Mysql大数据量常见问题解决思路

- 数据量大，怎么解决？ => 水平切分
- 如何保证可用性？ => 复制
- 各色各异的读写比，怎么办？ => 针对性设计
- 如何做无缝倒库？ => 双写+追日志

各色各异的读写比-读多写少

- 读多些少符合大部分业务场景
- 读多些少数据库解决方案，通常有几种：

1）新建索引提高读性能

2）读写分离，增加从库扩展读性能

3）增加缓存来扩展读性能

- 1）2）3）方案存在什么问题？
- 如何解决这些问题？

实践：用户状态读写

- 业务描述：

1）帮帮用户**登录，登出**时会改变用户状态

2）**列表页，最终页**都会查询用户状态

3）**发消息**时，会查询用户状态

- 可能存在的数据问题

1）可用性

2）一致性

- 如何解决

各色各异的读写比-读写相近

- 部分场景-读写比相近
- 实践：**离线消息拉取**
- 少数详见-写多读少
- 实践：**聊天记录备案需求**
- 解决方案？

无缝导库

- 什么情况下会导库？

1）实践：**分库库迁移与拆分**

2）实践：**帮帮数据库迁移**，mongo => mysql

3）实践：**数据库增加字段** => 能alter table么？

- 解决方案

1）停服务 => 业务能接受的话，强烈建议！

2）无缝方案

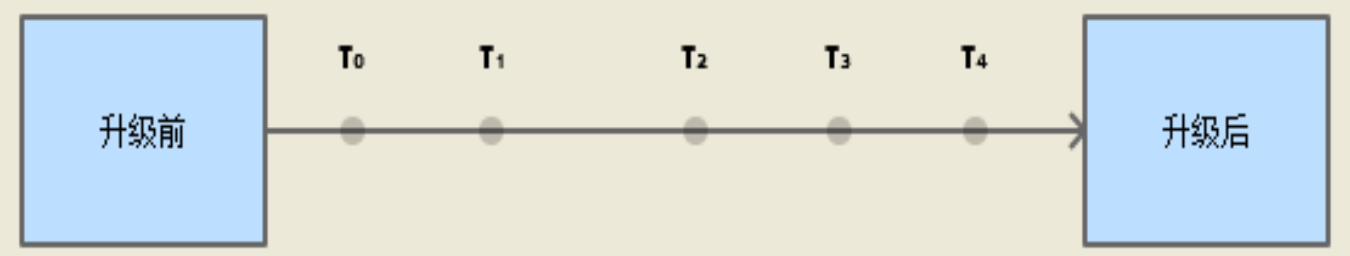
实践：追日志倒库

- 目标

1）1个库分成4个库

2）多个库部署到多台物理机上

- 解决方案
- 如何回滚？



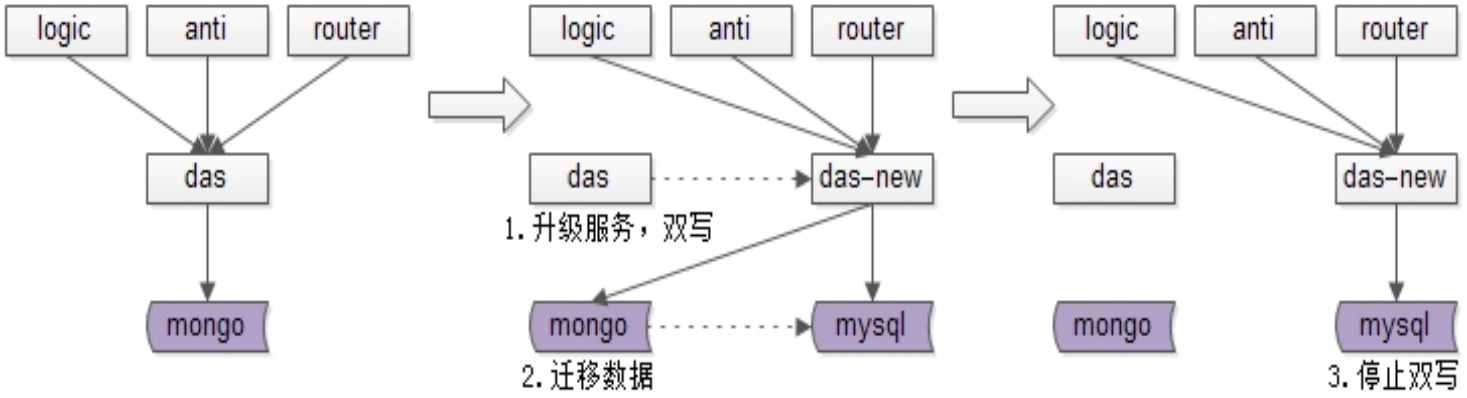
实践：双写倒库

- 目标

1）数据由mongo迁移到mysql（mysql增加字段同样适用）

2）不停止对外服务

- 解决方案



- 如何回滚？

三、数据库设计实战

核心方向：拆库

拆分的依据是什么？

实战-用户库拆分？

- 用户库，10亿数据量

user(**uid**, **uname**, passwd, age, sex, create_time);

- 业务需求如下

（1）1%登录请求 => where uname=XXX and passwd=XXX

（2）99%查询请求 => where uid=XXX

实战-帖子库拆分？

- 帖子库，15亿数据量

tiezi(**tid**, **uid**, title, content, time);

- 业务需求如下

（1）查询帖子详情（90%请求）

SELECT * FROM tiezi WHERE tid=$tid

（2）查询用户所有发帖（10%请求）

SELECT * FROM tiezi WHERE uid=$uid

- 目前imc的做法，目前如何查询用户发帖？
- 潜在优化方案

实战-好友库拆分？

- 好友库，1亿数据量

friend(**uid**, **friend_uid**, nick, memo, XXOO);

- 业务需求如下

（1）查询我的好友（50%请求） => 用于界面展示

SELECT friend_uid FROM friend WHERE uid=$my_uid

（2）查询加我为好友的用户（50%请求） => 用户反向通知

SELECT uid FROM friend WHERE friend_uid=$my_uid

实战-订单库如何拆分？

- 订单库，10亿数据量

order(**oid**, **buyer_id**, **seller_id**, order_info, XXOO);

- 业务需求如下

（1）查询订单信息（80%请求）

SELECT * FROM order WHERE oid=$oid

（2）查询我买的东东（19%请求）

SELECT * FROM order WHERE buyer_id=$my_uid

（3）查询我卖出的东东（1%请求）

SELECT * FROM order WHERE seller_id=$my_uid

四、分库后业务实战

分库后带来的问题？

实战-IN查询怎么做？

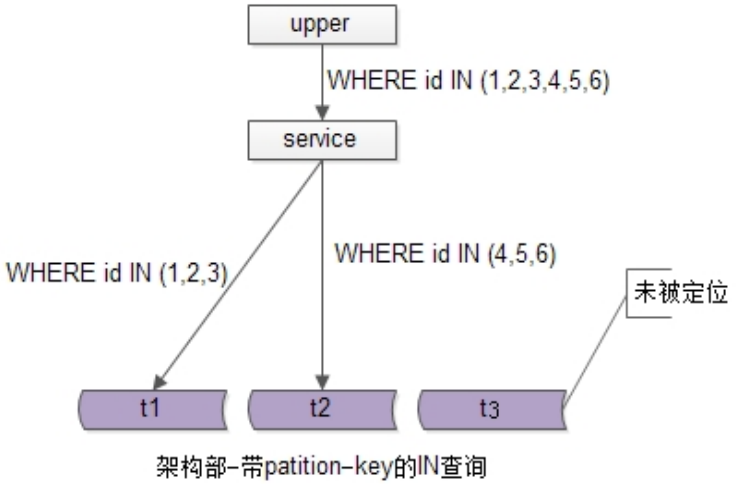
- 用户库如何进行uid的IN查询

user(**uid**, **uname**, passwd, age, sex, photo, create_time);

- Partition key：uid
- 查询需求：IN查询：WHERE uid IN(1,2,3,4,5,6)
- 解决方案：服务做MR

（1）直接分发

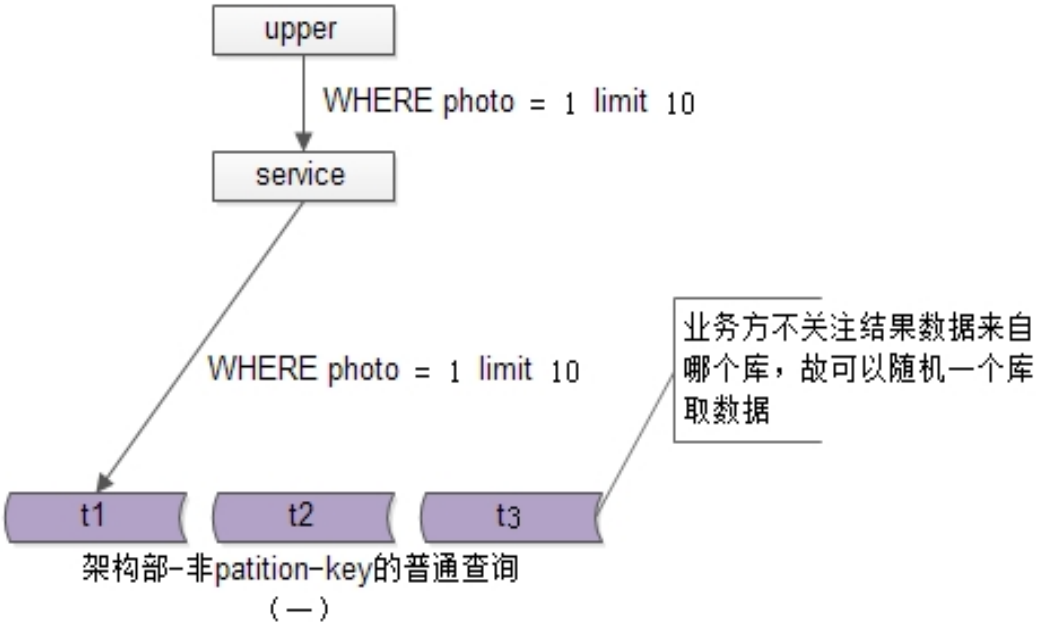
（2）拼装成不同SQL



架构部-带patition-key的IN查询

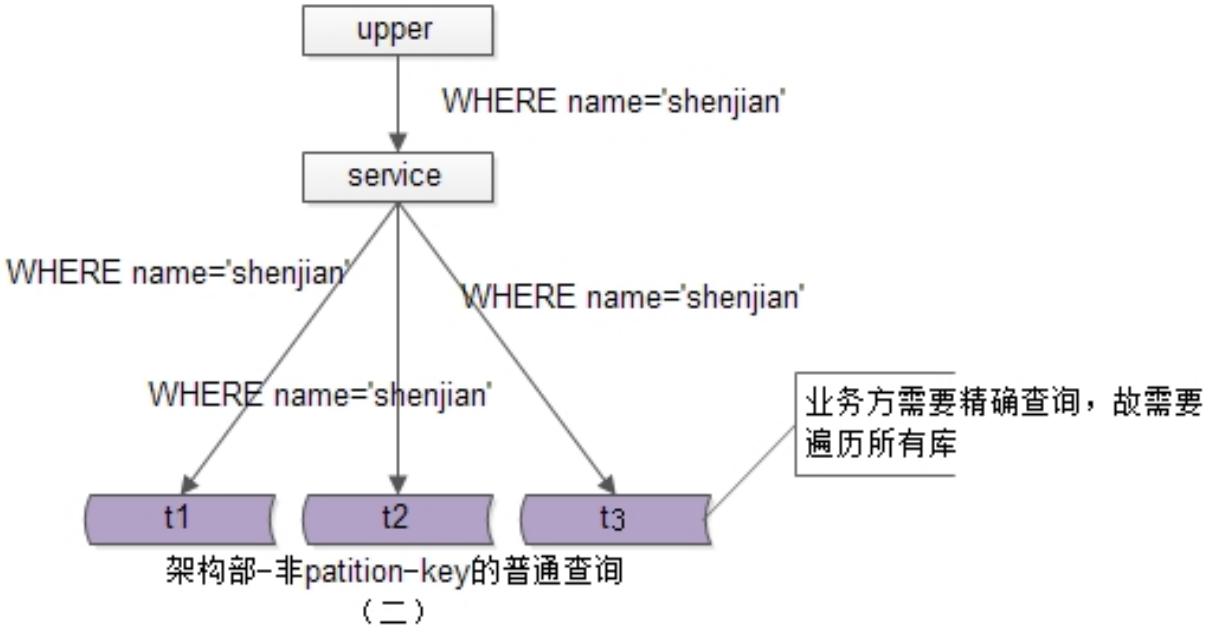
实战-非partition key的查询怎么做？

- 需求：头像查询
- 解决方案：只定位一个库



实战-非partition key的查询怎么做？

- 需求：登录查询
- 解决方案：服务做MR，一条数据返回则返回



实战-跨库分页查询怎么做？

- 需求：ORDER BY xxx OFFSET xxx LIMIT xxx

（1）按时间排序；

（2）每页100条记录；

（3）取第100页的记录；

- 单机方案

ORDER BY time OFFSET 10000 LIMIT 100

- 分库后如何实现？

实战-跨库分页查询怎么做？

- 分库后难点：如何全局排序？
- 传统方案：SQL改写 + 自己排序

（1）ORDER BY time OFFSET 0 LIMIT 10000+100

（2）对20200条记录进行排序

（3）返回第10000至10100条记录

实战-跨库分页查询怎么优化？

- 方案一：

（1）技术上，引入特殊id，作为查询条件（或者带入上一页的排序条件）

（2）业务上，尽量禁止跨页查询

- 单机情况

（1）第一页，直接查

（2）得到第一页的max(id)=123（一般是最后一条记录）

（3）第二页，带上id>123查询：**WHERE id>123 LIMIT 100**

=>

这样每次只要查100条，那分库情况呢？

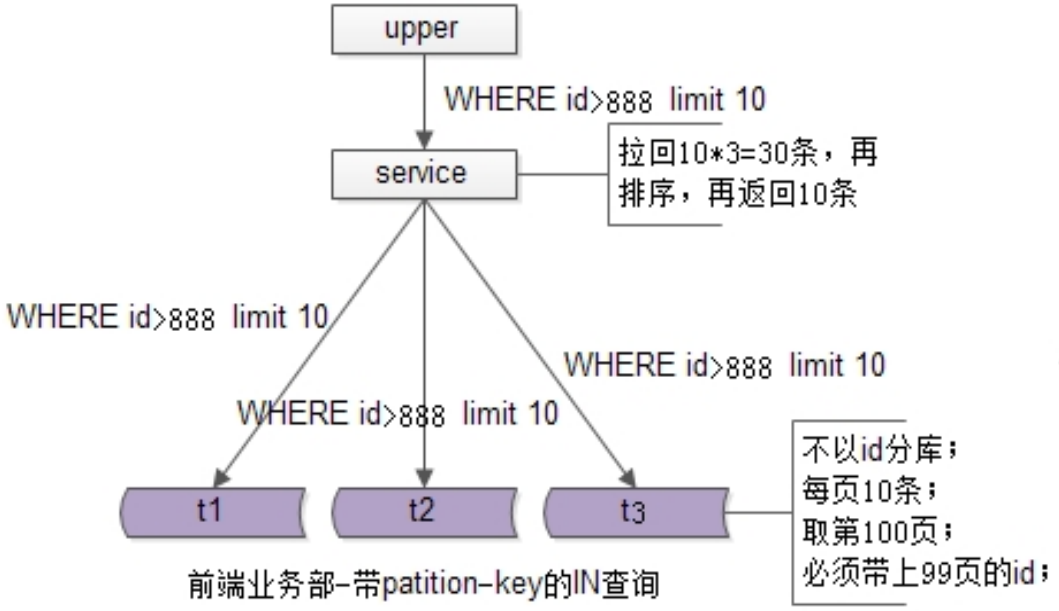
实战-跨库分页查询怎么优化？

- 分库情况（假设3个库）

（1）将WHERE id>xxx LIMIT 100分发

（2）将300条结果排序

（3）返回前100条



实战-跨库分页查询怎么优化？

- 方案二：

（1）业务上：禁止查询XX页之后的数据

（2）业务上：允许模糊返回 => 第100页数据的精确性真这么重要么？

总结

第一章、基本概念

- 总结

1) 分片

2) 复制

3) 分组

4) 路由规则

第二章、常见问题与解决方案

- 总结

1）数据量大，解决思路是**分片**

2）可用性，解决思路是**冗余**

3）读写比

3.1）读多些少：用**从库，缓存，索引**来提高读性能

3.2）业务层控制**强制读主**来解决从库不一致问题

3.3）**双淘汰**来解决缓存不一致问题

3.4）读写相近，写多读少：**不要使用缓存**，该怎么整怎么整

4）无缝导库

4.1）**写日志追数据**

4.2）**双写**

第三章、数据库设计实战

- 总结

（单key）用户库如何拆分： user(uid, XXOO)

（1对多）帖子库如何拆分： tiezi(tid, uid, XXOO)

（多对多）好友库如何拆分： friend(uid, friend_uid, XXOO)

（多key）订单库如何拆分：order(oid, buyer_id, seller_id, XXOO)

第四章、分库后业务实战

- 总结

1）IN查询

1.1）分发MR

1.2）拼装成不同SQL语句

2）非partition key查询

2.1）定位一个库

2.2）分发MR

3）夸库分页

3.1）修改sql语句，服务内排序

3.2）引入特殊id，减少返回数量

3.3）业务优化，允许模糊查询

Q&A&讨论

谢谢！