DTCC

2016中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2016

数 据 定 义 未 来

SequeMedia 盛拓传媒 IT168 ChinaUnix ITPUB

# 数据库一致性架构设计实践

58沈剑

# 关于-我

- 前百度 - 高级工程师
- 58同城 - 高级架构师，技术委员会主席，技术学院优秀讲师
- 58到家 - 技术总监，技术委员会主席
- “架构师之路”作者，深夜写写技术文章
- 本质：技术人一枚

# 目录

- 为什么数据会存在一致性问题
- “并发写”一致性实践
- “数据冗余”一致性实践
- “主从库”一致性实践
- “数据库与缓存”一致性实践
- 总结

# 为什么数据会存在一致性问题

- 如何保证数据的可用性？
- 当一个数据放到多个地方时，会发生什么？

# “并发写”一致性实践

# “并发写”一致性实践

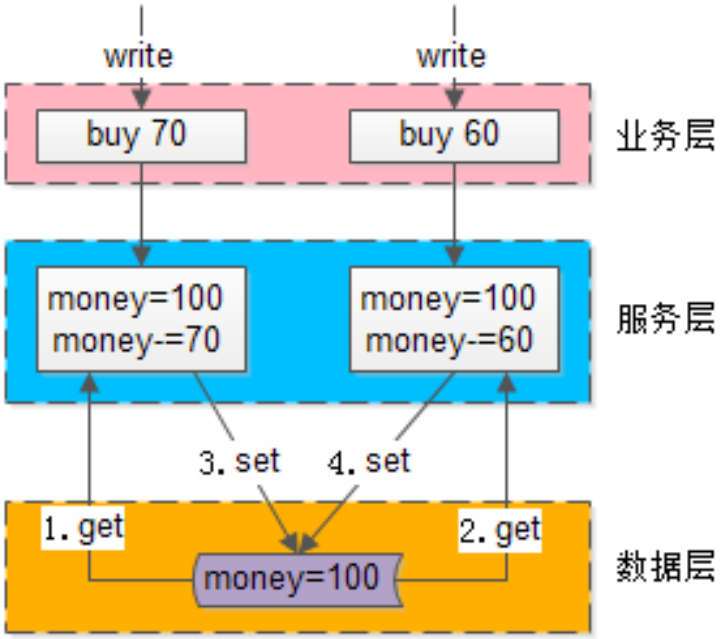
- 业务场景抽象：select & set
- 业务过程

（1）select money

（2）money -= $input

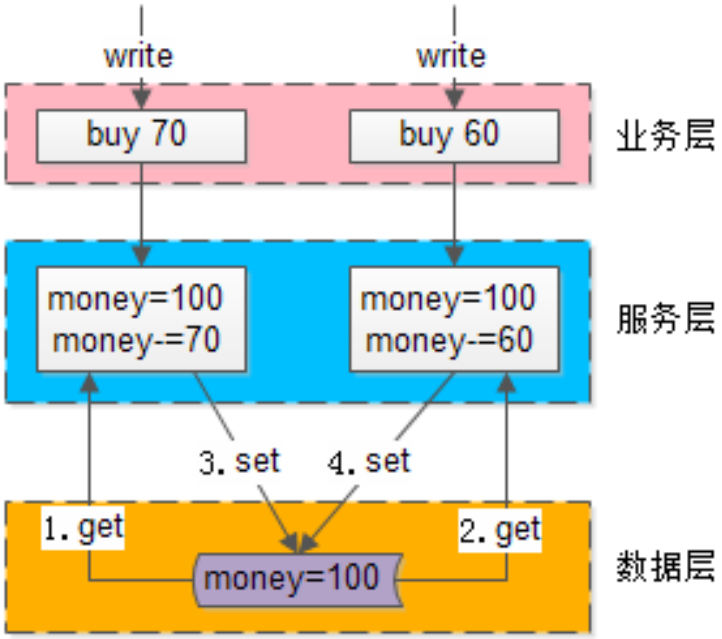
（3）set money

- 为什么会出问题？
- 同一个数据被冗余读出多份
- set -= 能解决么？



# “并发写”一致性实践

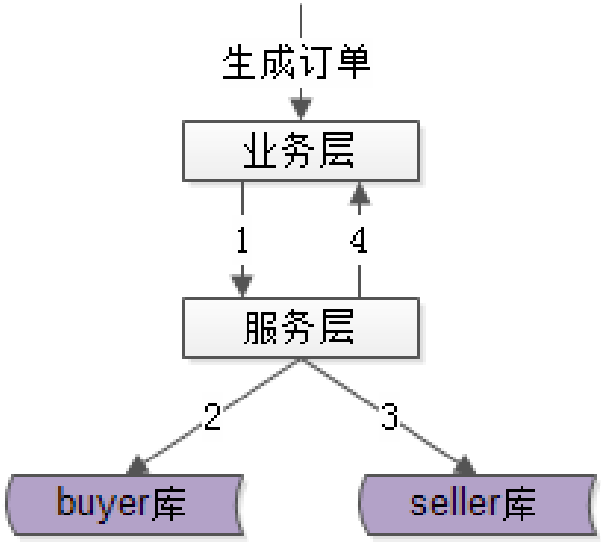
- 不一致根本原因？
- 数据变换时，初始条件已经变换
- 解决方案：数据变更时，比对初始条件
- CAS
- 对于这个例子？



# “数据冗余”一致性实践

# “数据冗余”一致性实践

- 为什么数据要冗余？
- 业务场景抽象：插入正向表，插入冗余表
- 冗余方式：同步、异步、线下异步
- 为什么会不一致？
- 同一个数据被冗余多份



# “数据冗余”一致性实践

解决方案：扫全库

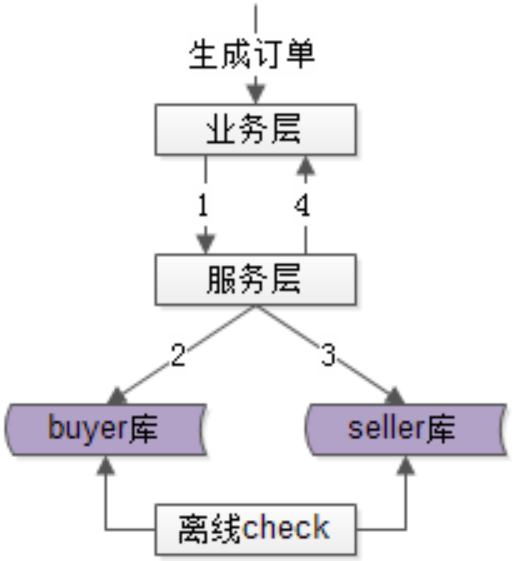
方案缺陷：效率低

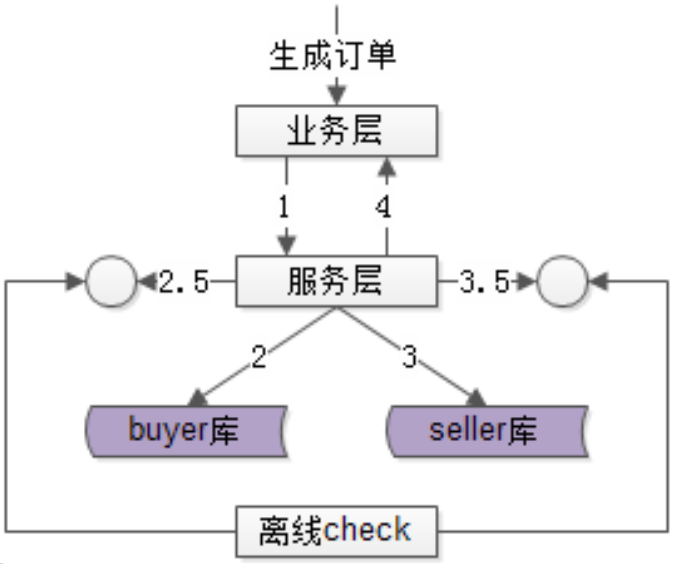
优化方案：扫增量

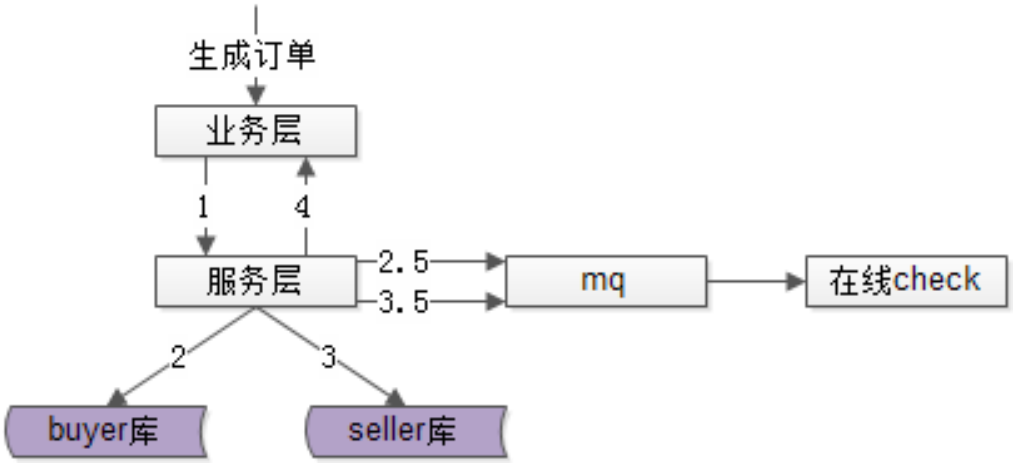
方案缺陷：不实时

优化方案：实时检测

方案缺陷：引入组件较多



# “主从库”一致性实践

# “主从库”一致性实践

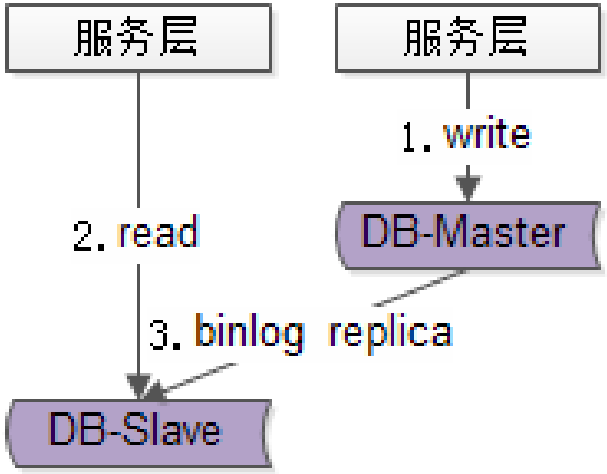
- 主从同步，读写分离的“DB架构”
- 业务过程

（1）写主库

（2）读从库

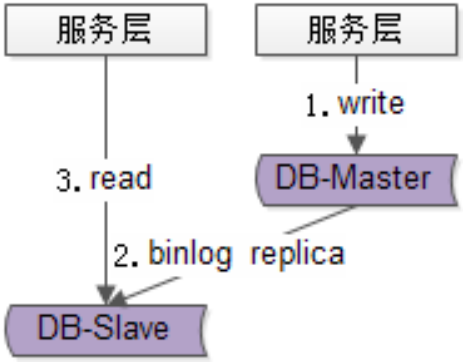
（3）主从同步成功

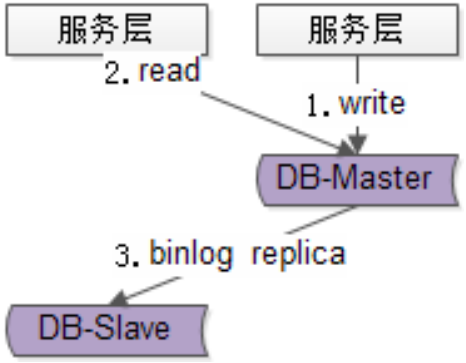
- 为什么会不一致？
- 同一个数据被冗余多份

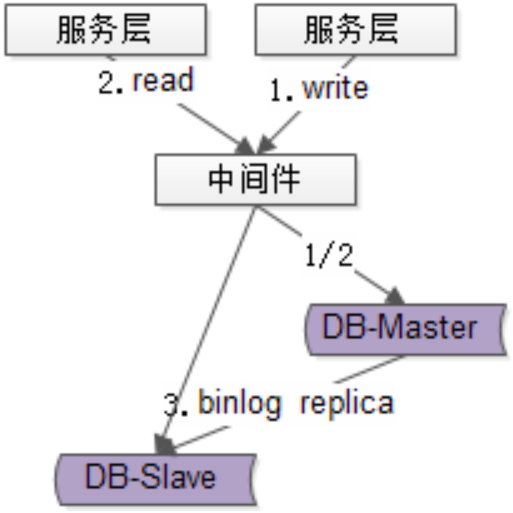


# “主从库”一致性实践

- 主从同步，读写分离的“DB架构”
- 方案一，优缺点？
- 方案二，优缺点？
- 方案三，优缺点？



# “数据库与缓存”一致性实践

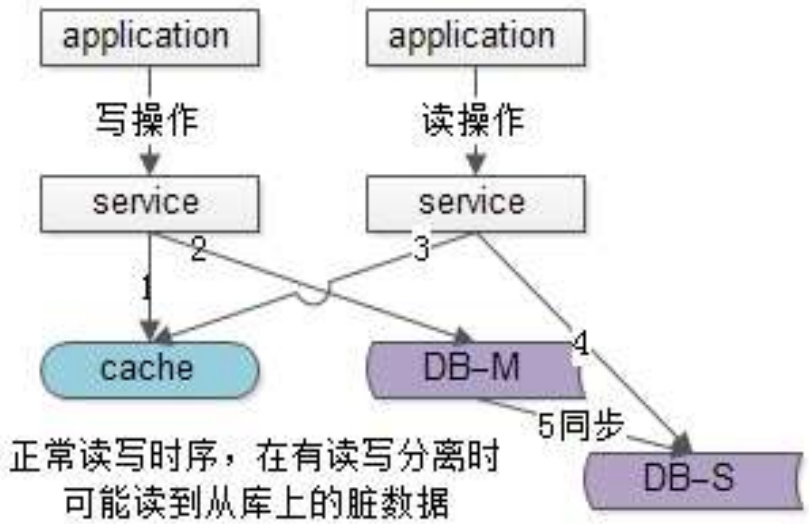
# “数据库与缓存”一致性实践

- 主从同步，读写分离的“DB+缓存架构”
- 业务过程

（1）写操作过程

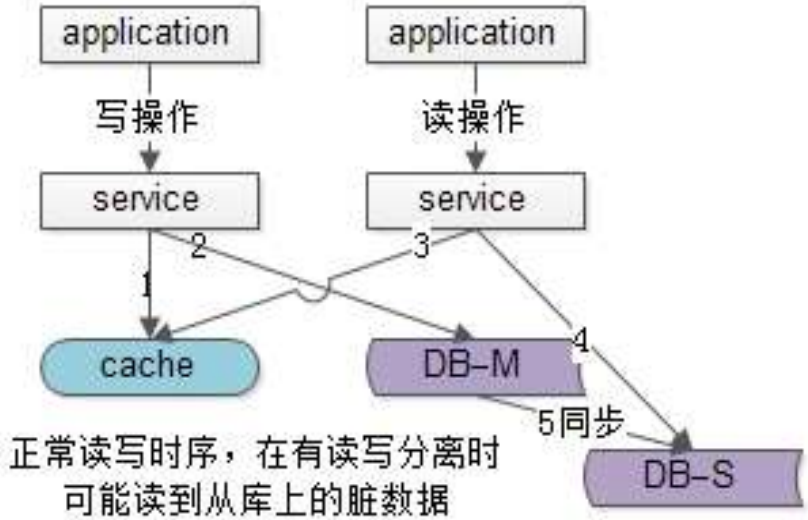
（2）读操作过程

- 为什么会不一致？
- 同一个数据被冗余多份
- 不一致怎么产生的？



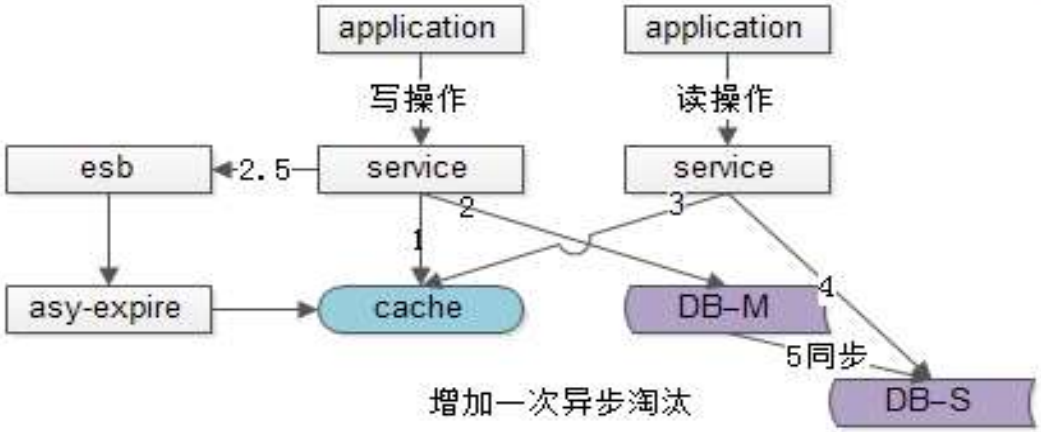
# “数据库与缓存”一致性实践

- 主从同步，读写分离的“DB+缓存架构”
- 不一致是在“数据被写入”时，有可能发生
- 解决思路：二次淘汰
- 方案一：写入阻塞二次淘汰法
- 优缺点？
- 方案二：异步timer二次淘汰法
- 优缺点？

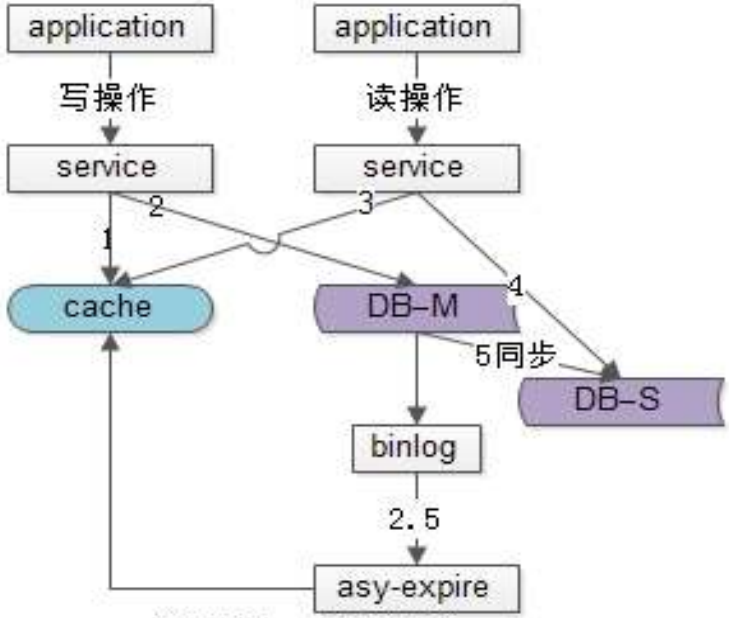


# “数据库与缓存”一致性实践

- 解决思路：二次淘汰
- 方案三，优缺点？
- 方案三，优缺点？




增加一次异步淘汰




通过binlog来异步淘汰

# 总结

# 总结

- “并发写”一致性实践：CAS
- “数据冗余”一致性实践

（1）扫全库修复法
（2）扫增量修复法
（3）实时修复法

- “主从库”一致性实践

（1）半同步复制
（2）读主库
（3）中间件

- “数据库与缓存”一致性实践：双淘汰

（1）写阻塞二次淘汰
（2）异步timer二次淘汰
（3）异步esb二次淘汰
（4）binlog异步二次淘汰

# Q&A

**谢谢！**

“架构师之路”公众号