一次线上生产库的全流程切换完整方案

京东零售技术 2025-01-24 159 阅读5分钟

智能总结 复制 重新生成

本文主要介绍了一次线上生产库的全流程切换方案。包括对陈旧技术栈系统的现状梳理，迁移方案如节奏、代码改造、数据源选型、存量和增量数据处理，以及上线的“三板斧”，即保障可监控、可灰度、可回滚，最终实现平滑切换。

关联问题: 如何选数据源 回滚怎样操作 灰度切量注意啥

基于该文章内容继续向AI提问

一、现状梳理

本篇介绍了一次数据库迁移的完整方案。 本次需要改造的系统为一个较为陈旧的技术栈系统，其中MongoDB作为核心数据存储中间件，承担着存储全部核心数据的重要任务。该系统目前的配置为1主1副本模式，涉及1个数据库和2张表，服务于7个不同的应用。尽管系统架构相对简单，但其在日常运营中发挥着不可或缺的作用。目前需要将MongoDB存储在其它介质中，**如何能够保障在不影响线上使用的情况下，平滑切流到新库，是本文主要探讨的问题。**

二、迁移方案

2.1 迁移节奏

整体节奏分为

1.梳理范围，因为系统内不仅有mongo还同时有mysql数据源，需要梳理出使用mongo的所有业务范围

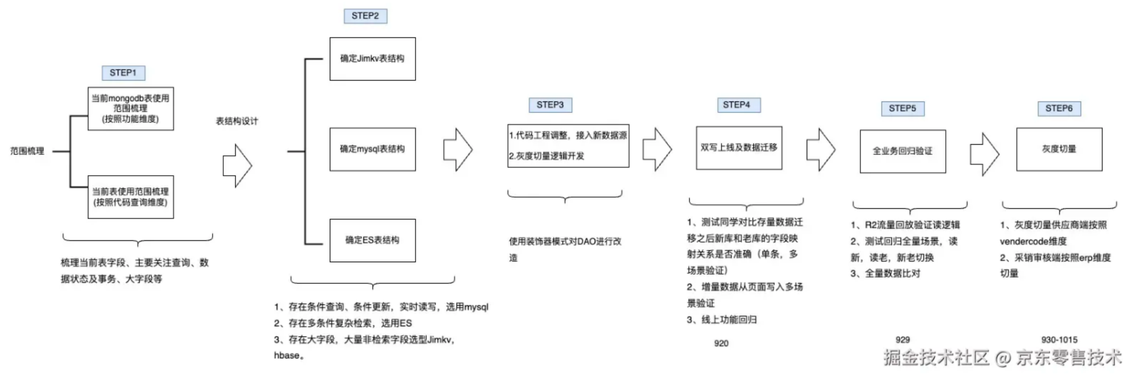
2.确定好原有的数据，应该存储在哪个介质中，确定好存储标准，需要能够cover住原有的所有业务，包括读写性能

3.对原有数据结构的DAO层进行改造

4.需要对数据进行双写并进行数据迁移

5.R2流量验证/测试回归/数据比对 进行验证

6.切量:放量节奏



2.2 代码改造/数据异构

采用装饰器模式，统一控制双写逻辑（主写，辅写），统一控制切量逻辑，下线逻辑， 抽取代码中原有的直接调用底层mongodb API的代码，将其不改业务逻辑的情况下迁移到Dao层。这样做的目的是为了后续做切流适配逻辑。不改逻辑及出入参的目的是为了避免对当前业务造成影响。

选用数据源的依据为

特性	JimKV	HBase
优势	- 支持多存储引擎（SSD、AEP）- 基于 Raft 协议的强一致性和多机房容灾 - 完善的运维监控和弹性伸缩能力	- 支持PB级别的存储容量- 云原生架构，支持单集群和主备集群 - 高吞吐性能，适合写密集型应用
劣势	- 由于Raft协议，写性能低于JimDB	- 故障恢复时间较长，约1—2分钟
适用场景	- 数据一致性和可靠性要求高- 数据存储量大- 读流量大于写流量	- 存储量非常大（PB级别）- 写密集且性能要求高的场景
技术选型推荐理由	- JimKV满足存储量和吞吐量要求- 数据一致性和可靠性优于HBase - 适合读流量大于写流量的应用场景	- 适用于存储量极大的场景，但对一致性要求较高的场景不如JimKV适合

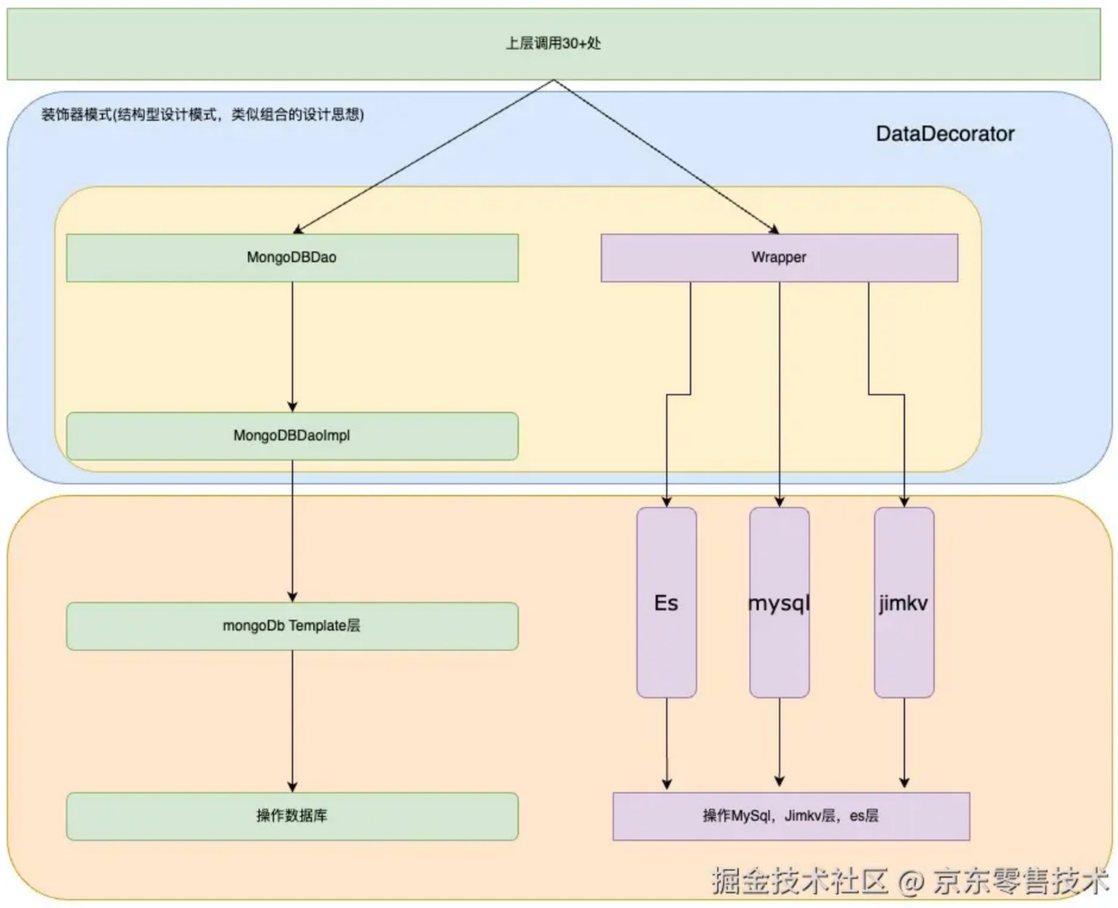
京东零售技术 LV.5
技术运营 @京东
作者榜No.17 优秀作者
147 文章 108k 阅读 1.1k 粉丝
关注 私信

目录 收起

相关推荐
高性能队列 Disruptor 在 IM 系统中的实战
123阅读 · 1点赞
配置管理方法论：国际支付参数运营原...
107阅读 · 2点赞
深入解析Java反射机制在JVM中的实现...
48阅读 · 1点赞
TKE & Ingress-Nginx Controller 之客户...
62阅读 · 1点赞
Grafana 快速搭建go-metrics 仪表盘备忘
85阅读 · 0点赞

精选内容
🚀 NestJS 11 正式发布！核心内容更新...
元兮 · 628阅读 · 5点赞
@SneakyThrows：是Java异常处理的"...
加瓦点灯 · 115阅读 · 1点赞
RocketMQ实战—10.营销系统代码优化
东阳马生架构 · 23阅读 · 0点赞
从Java程序员视角看电商SEO优化：重...
Asthenia0412 · 43阅读 · 1点赞
基于SpringBoot养老院平台系统功能实...
115432031q · 55阅读 · 0点赞

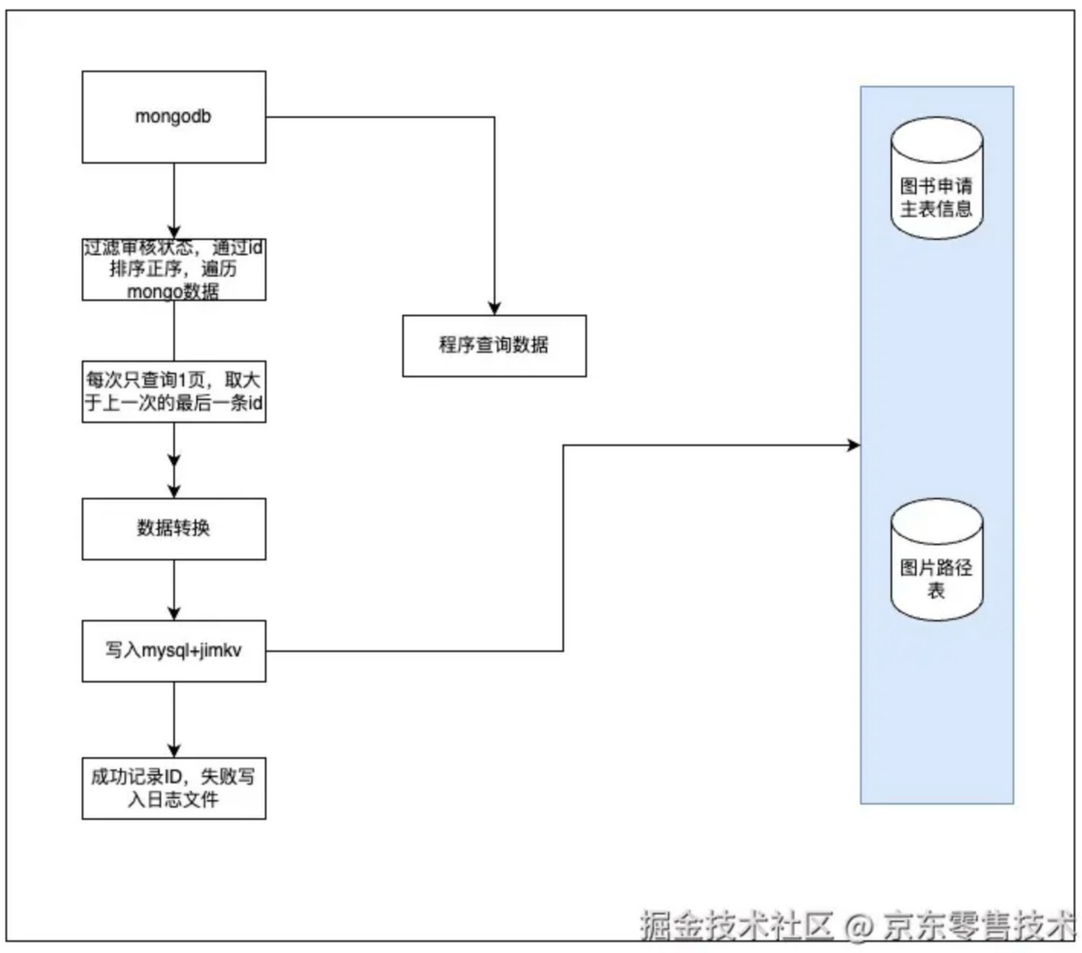
基于以上原则,我们选用JImKV(京东自研中间件)，Mysql和ES作为MongoDB的替换的数据源，数据源切换Dao层的改造方式如下:



2.3 存量数据迁移

方案	是否可实现	难度
使用大数据抽数任务	是	易
使用代码异步任务的方式	是	易
DRC同步	从mongo到数据库不支持	略

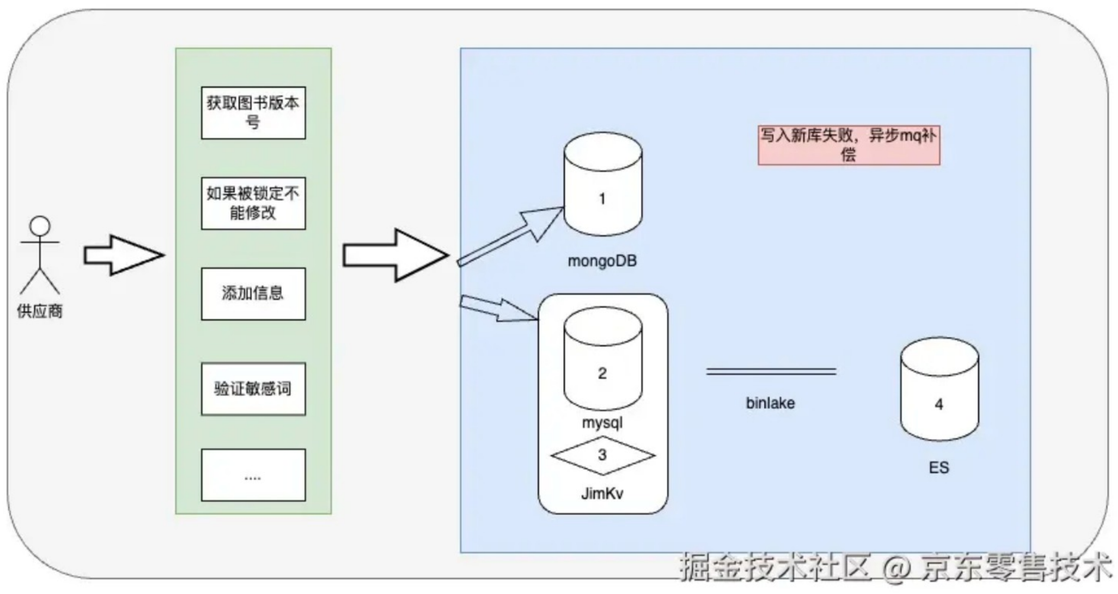
考虑整体的数据量并不大单表300w，通过大数据离线表的方式效率并不高，通过代码更加的灵活，可以随时调整速度和范围存量数据分了两部分1、已经审核通过，申请单不会在有任何变更，可以随时迁移，比对2、申请单处于过程中的数据，数据随时会变更。凌晨迁移，打开双写

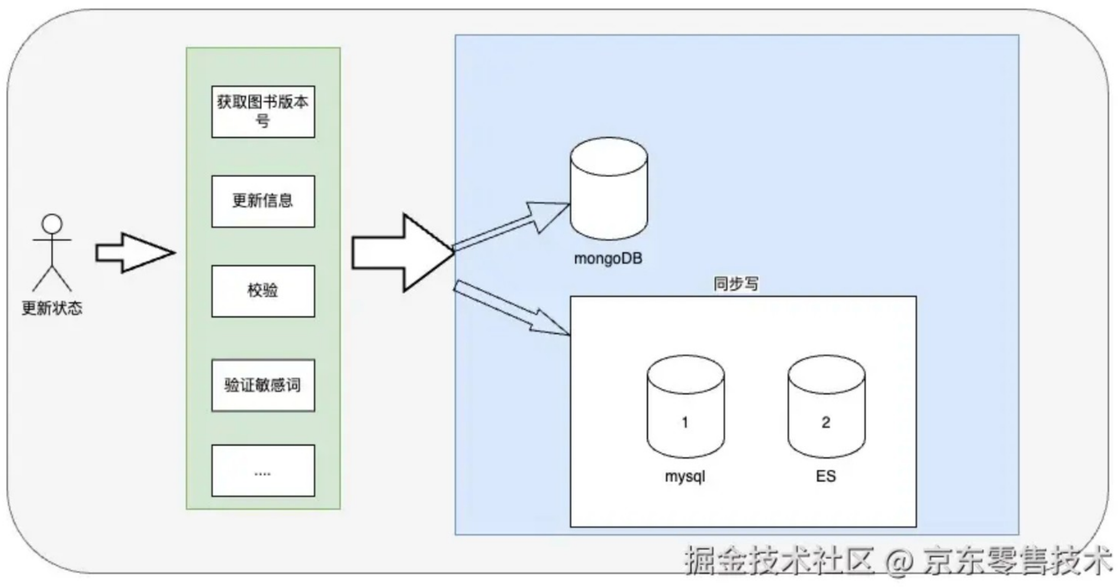


2.4 增量数据同步

创建申请单和更新不包含状态字段时的操作

先写mongo再写mysql，以mongo写入成功为准，写mysql失败，mq异步补偿





三、上线三板斧(灰度/监控/回滚)

本章节主要探讨在进行数据迁移和代码改造这些基础工作完成之后，如何保障上线没有线上问题，如何保障平滑切流和听写，工作主要聚焦于上线三板斧，可灰度，可回滚，可监控等方面，具体工作如下:

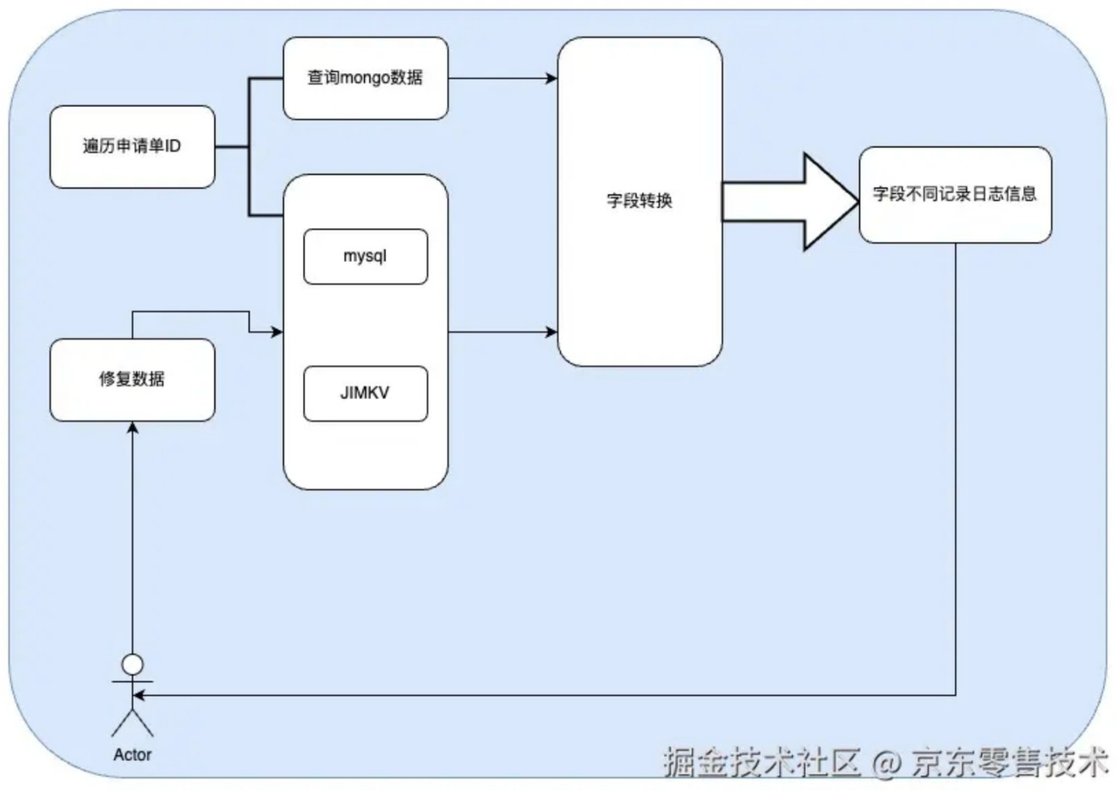
3.1 可监控(数据对比读逻辑)

增量数据比对

双写数据完成后发送MQ，消息里面查询新库，老库的数据进行实时比对，不一致数据记录不一致字段，关键字业务报警，写入日志文件，导出分析

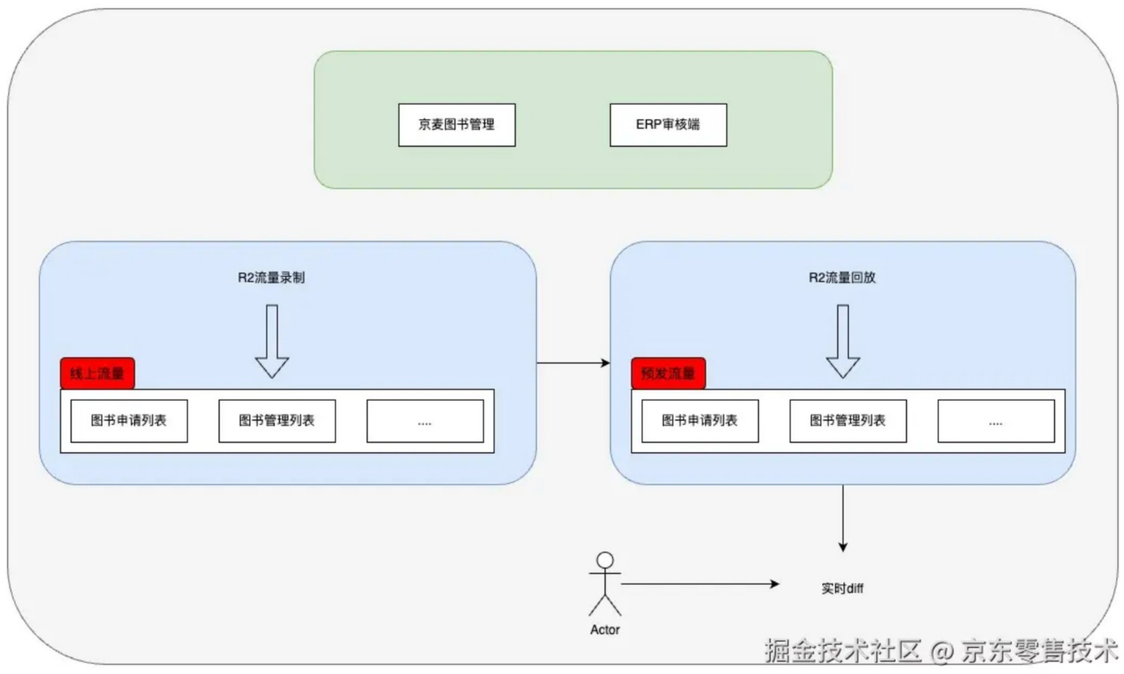
存量数据比对

遍历全量老库数据，与新库查出数据，转换成相同对象对比数据一致性，异常数据写入日志文件分析



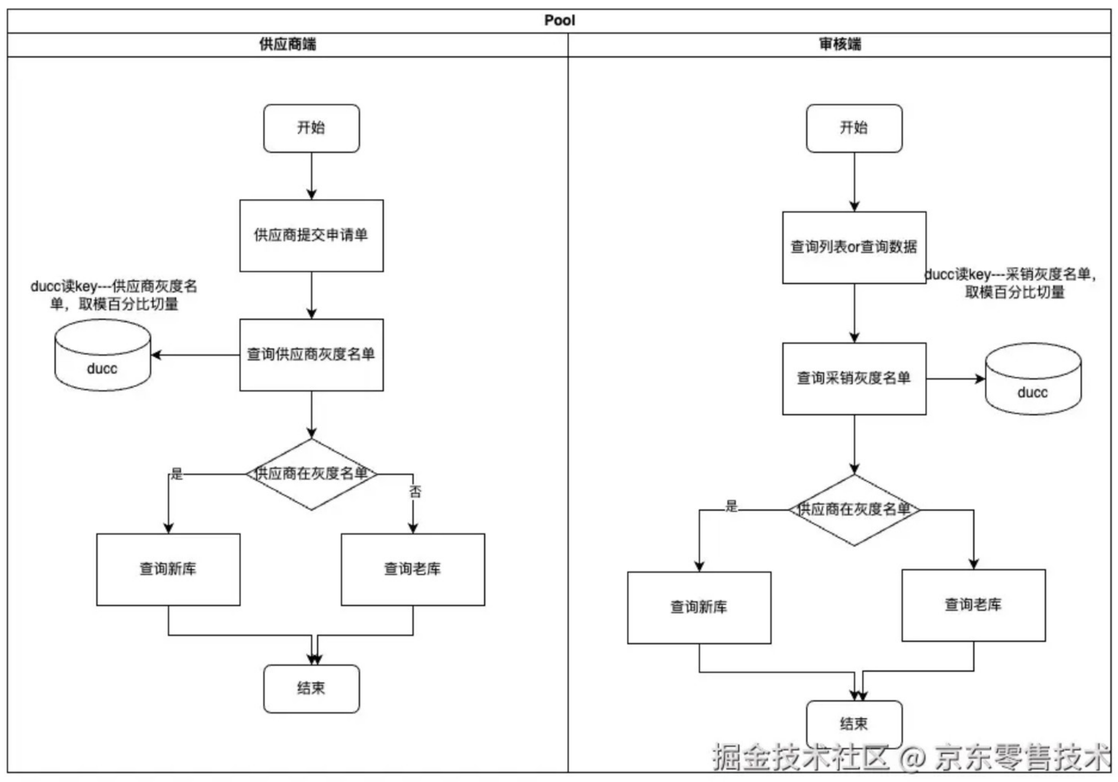
3.2 可监控(对比读逻辑)

对比逻辑，引入R2流量回放对比，提高对比速度，

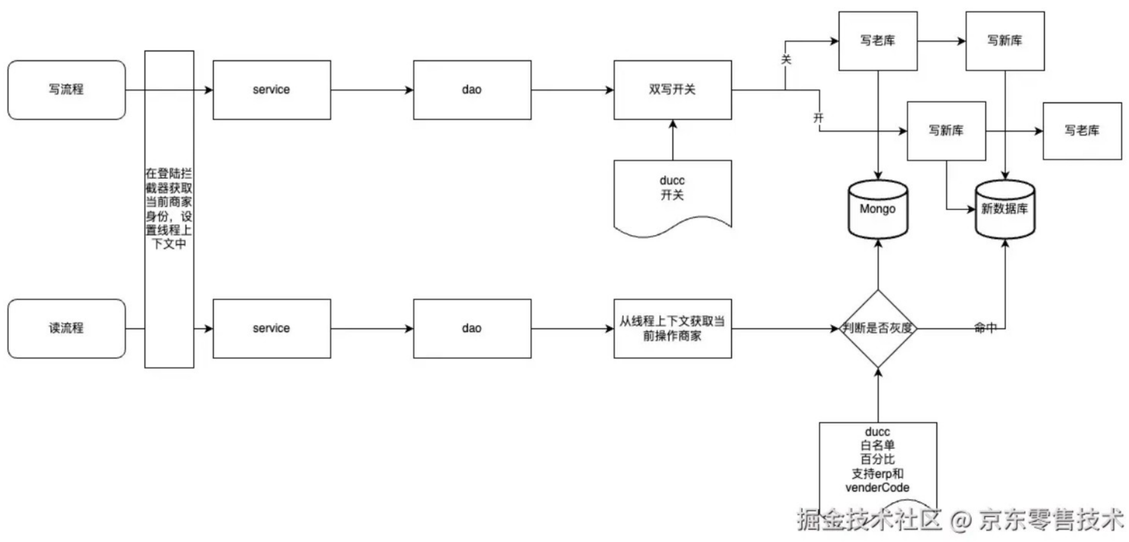


3.3 可灰度(灰度切量读)

读切流，按照供应商和采销白名单+百分比来切流



切流时，由于需要根据pin对流量分散，但是不在同一线程内，使用threadlocal对商户信息进行设置和读取



3.4 可回滚(灰度切量写)

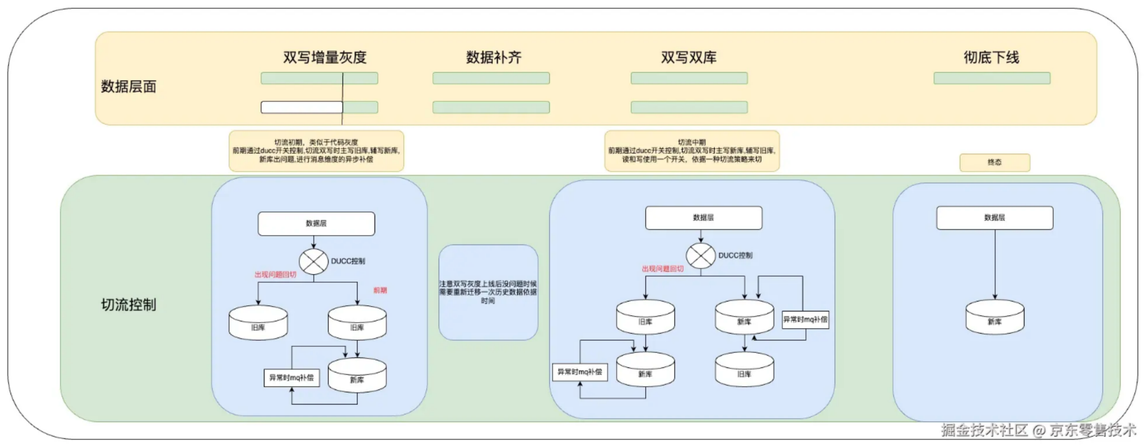
写切流 分为四步

1.首先验证 写新库没问题 相当于对新加代码进行灰度 如果有问题 进行回切

2.当验证写新库没问题，需要补齐数据库数据

3.当数据补齐后 转换为主写新库

4.后续如果读写新库都没问题 可以彻底下线旧库存



四、总结

本文详细梳理了线上生产环境的全流程，包括迁移和切换的灰度方案对比。在数据源选型方面，根据实际业务需求选择合适的中间件是整个工作的基石。在代码改造和数据异构方面，选择恰当的设计模式和合理的架构方案是关键所在。存量数据迁移和增量数据同步是不可或缺的步骤。上线过程中，确保系统具备可监控、可回滚和可灰度的能力，是实现平滑切换的保障。欢迎各位同学与我交流探讨。

标签： 后端

评论 0

登录 / 注册 即可发布评论！

暂无评论数据

为你推荐

互联网公司的完整开发流程是怎样的?
Java鱼仔 | 3年前 | 1.6k | 1 | 评论 — Java 后端

全链路压测：影子库与影子表之争
阿里云云栖号 | 2年前 | 2.1k | 4 | 评论 — 后端

如何设计可靠的灰度方案
阿里云云栖号 | 3年前 | 3.2k | 19 | 评论 — 数据结构

全链路压测：影子库与影子表之争
阿里云云栖号 | 2年前 | 2.9k | 9 | 评论 — 后端

B站稿件生产平台高可用建设分享
哔哩哔哩技术 | 8月前 | 3.8k | 6 | 评论 — 后端

全链路压测：影子库与影子表之争
阿里云云栖号 | 2年前 | 1.6k | 4 | 评论 — 后端

大厂钟爱的全链路压测有什么意义？四种压测方案详细对比分析
博学谷_狂野架构师 | 2年前 | 2.3k | 4 | 评论 — Java 后端

「完整FX」Python Flask 全流程全栈项目实战
用户79950538757 | 11月前 | 109 | 点赞 | 评论 — Python

记一次完整的上云经历
码海 | 2年前 | 172 | 点赞 | 评论 — 后端 Java

有赞全链路压测实战
有赞技术 | 6年前 | 1.9k | 22 | 2 — 有赞

线上事故不要慌！来看一次教科书级的危机处置示范

沐洒 | 1年前 | 1.9k | 29 | 16 | 前端 | Redux | React.js

iOS 性能优化实践：头条抖音如何实现 OOM 崩溃率下降50%+

字节跳动技术团队 | 4年前 | 19k | 165 | 14 | 性能优化 | iOS

记一次JAVA 线上故障排查完整套路

黎杜 | 4年前 | 2.1k | 14 | 3 | Java

记一次生产事故的排查流程

hellohello_tom | 3年前 | 833 | 13 | 1 | 后端

全链路压测：影子库与影子表之争

阿里云云原生 | 2年前 | 2.6k | 3 | 评论 | 云原生