

消息治理初步方案

一：背景介绍

1. 针对消息队列中的消息进行记录

目前大陆中的MQ发送消费记录只能通过日志查看消息的发送消费情况，包括查看MQ的控制台也是通过日志查询messageId才能看到实际情况。因此针对MQ的消息记录需要一个数据存储系统采集，此场景下，采用MongoDB保存MQ的记录。

2. 使用场景介绍

主要场景：

- 1) 网站实时数据处理。适合实时的插入、更新与查询，并具备网站实时数据存储所需的复制及高度伸缩性。
- 2) 缓存。由于性能很高，它适合作为信息基础设施的缓存层。在系统重启之后，由它搭建的持久化缓存层可以 避免下层的数据源过载。
- 3) 高伸缩性的场景。非常适合由数十或数百台服务器组成的数据库，它的路线图中已经包含对MapReduce引擎的内置支持。

二：业务场景

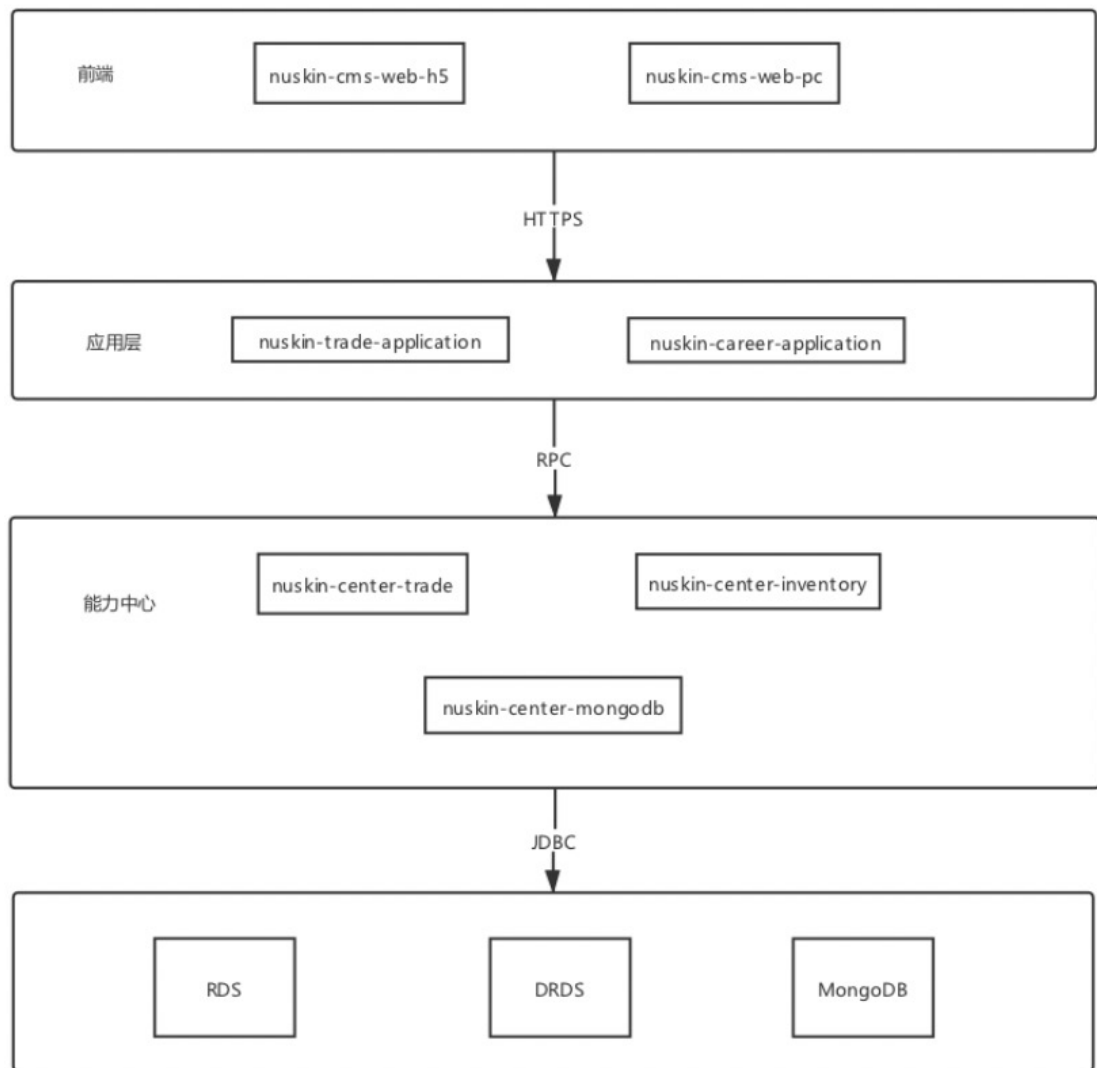
1. 大陆内部消息业务场景

- 1) MQ的发送方，发送消息异步发送MQ到MongoDB工程，发送的记录落库。包括发送成功、发送失败、重新发送。
- 2) MQ的消费方，MQ消费记录异步发送MQ到MongoDB工程，消费的记录包括消费成功、消费失败、重新消费、重复消费。

三：大陆相关设计

1. 新增 nuskin-center-mongodb 中心

- 1) 封装对MongoDB的curd操作，数据采用JSON格式存储，集成MQ，异步消费消息，存储消息记录。
- 2) 业务发送或消费MQ时，异步发送消息到MongoDB中心，记录数据。



2.MQ发送消费记录MongoDB场景

- 1) MQ发送消息的数据，包括MQ的topic、tag，实际发送的数据，存储到MongoDB。
- 2) MQ消费消息的数据，包括MQ的topic、tag，messageId，实际消费的数据，存储到MongoDB。
- 3) MQ消费幂等性处理，采用业务唯一ID加messageId，作为key放入Redis中。消费成功，则删除key。

3.待调研的问题

- 1) 消费失败，多次消费的次数记录多次还是一次。
-- MQ自动重试是MQ内部，目前不能记录自动重试次数。如果是业务开发中，代码重试的，可以记录重试次数。
- 2) 消费成功，记录到MongoDB失败，怎么处理。

-- 存储MongoDB失败，可以记录日志，后续可以考虑作为告警处理。

3) MQ发送失败或消费失败，需不需要告警通知。

-- 对内、对外梳理。对内暂时不需告警。对外的按需告警。

4) 梳理业务处理的MQ发送消费处理场景。

-- 梳理大陆台湾的业务发送MQ

5) 对内的消息可以不作记录MongoDB。

-- 目前的方案，业务中的内部MQ不记录到MongoDB。

6) MongoDB性能调研。