

缓存通用服务系统技术方案

- 一、背景
- 二、适用场景
- 三、原则
- 四、存储选型
- 五、流程图
- 六、主要模块
- 七、模块详细设计
 - 1、服务注册
 - 2、业务接入
 - 3、消息消费
 - 4、服务查询
- 八、排期
- 九、结论

一、背景

目前各个业务线存在一些接口，严重依赖下游，一旦出现系统级故障，只能粗暴的采取降级策略，给用户带来不好的体验和损失，为了改变这一现状，致力打造一个通用的缓存服务系统，应对突发状况，也提升系统的稳定和可靠性，避免带来重大损失。

二、适用场景

- 1、针对需要缓存接口数据的（接口是读接口），比如外部第三方服务接口、用户分群数据，ab实验数据，签到信息等等、
- 2、可以针对一些内网调用接口比较耗时的，都可以缓存起来，并支持非故障情况下实时调用。

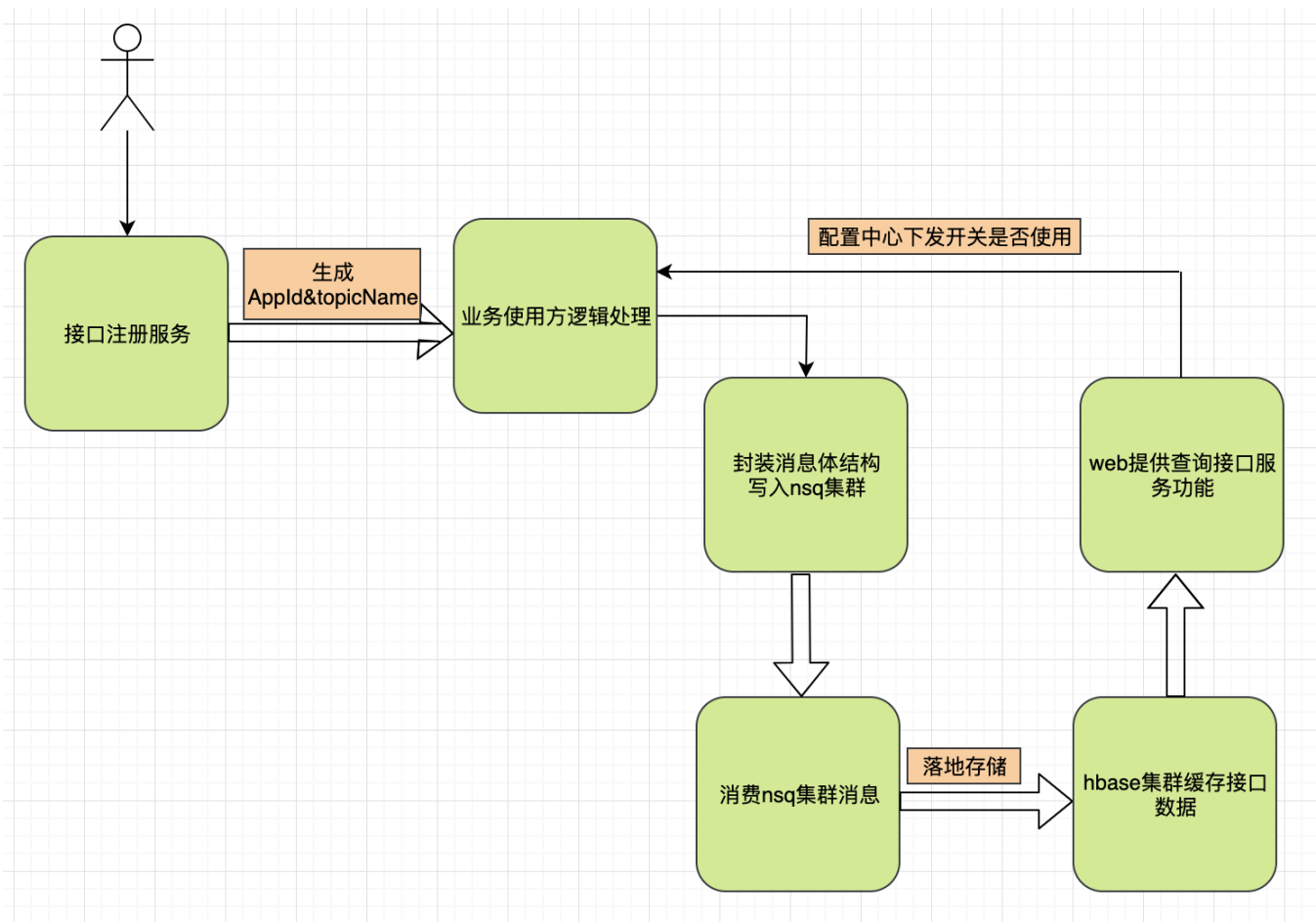
三、原则

- 1、通用服务功能，主要是为了系统的可扩展性强，与业务解耦，并且能够最大化对保障系统对使用方透明，最大程度做到一站式服务。
- 2、针对消费消息的校验，要非常的严格，由于系统是大部分情况下是应对故障情况下，对消息的丢失，丢弃，并不是非常的care，简言之，宁愿不消费，不能乱消费。

四、存储选型

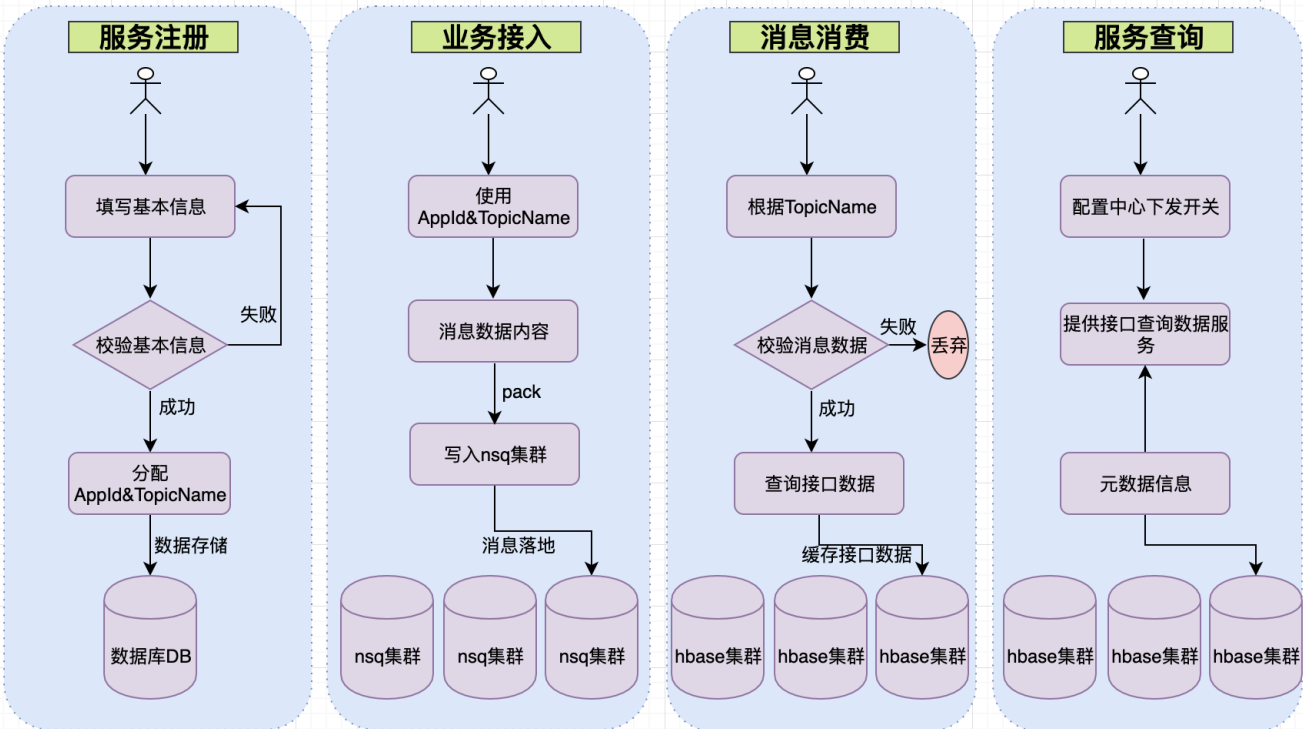
- 1、鉴于接口数据存储的特点，目前比较倾向选择在hbase和Cassandra之间。
- 2、经过与运维和DBA的沟通，目前公司还没开始使用阿里云的Cassandra，只有大数据分析自建的Cassandra，不过年后DBA会再做评估，后续可能会逐步引进。
- 3、选择hbase的优势，场景合适，自己比较熟悉，上手快，公司已有集群支持，其它的存储可讨论再定？

五、流程图



六、主要模块

接口数据缓存通用服务系统



七、模块详细设计

1、服务注册

(1) 功能描述

主要目的是根据接入方提供的接口基本信息，能够清楚知道该接口目前的使用基本情况。

能够自动化的更新接口数据，与业务解耦，不用关系缓存更新接口数据细节，对使用方透明。

还能够避免多方对同一接口缓存冗余情况的发生。

(2) 表的设计

字段	department	group	product_line	path	domain	method	avg_qps	top_qps	time_limit	count_limit	resp_time	query_qps	enable	topic
描述	部门	组	业务线	接口路径名称	线上接口域名	请求方式	业务qps (正常情况下平均)	峰值qps (服务可支撑)	时间限制, 该用户调接口多久一次更新	次数限制, 该用户每请求多少次才会更新一次。	接口的响应时间 (平均响应时间即可)	预估使用缓存查询接口的qps	是否禁用	分配的topic名称
后续可讨论的字段														
1、消费者进程数是否做成可配？														
2、hbase的cell的ttl是否做成可配？														

(3)后续

时间条件允许的提前下，后面可提供后台页面注册功能等可视化界面。

(4) 校验注册信息

- 针对domain和path这两个字段做唯一索引键设置，避免多个业务方注册了相同的服务接口，造成接口请求压力倍增。
- 基本信息填写是否完备，对于给定的接口信息，需要ping探测、qps等相关数据字段必填。

3. 注册成功后，对于有接口默认返回数据的接口，进行轮询与当前接口数据字段是否一致，并企微告知(主要是为了防止后期业务接口上有接口字段新增，未同步到注册表，造成数据不一致的潜在风险)
4. 由于缓存更新严重依赖注册信息数据，因此在对注册信息变更的同时，是否必要设置一个缓冲时间，以防止数据设置的过于扩张或其它未知影响，等待审核期（可以讨论下）？？

2、业务接入

- (1) 使用方式
- 接入方根据注册成功后提供的AppId和topic名称，在业务代码中针对需要缓存接口的地方，使用nsq发送消息即可。
- (2) 消息体字段结构
- nsq发送消息体数据主要有：Appid（必需）、topic（必需）、请求参数param（必需）、接口名称path（非必需，可讨论？）、请求方式method（非必需，可讨论？）
- (3) 后续
- 探讨是否有更好的方案，更高效的消息队列，针对消息发生积压等异常情况，有没有容错机制，自动处理。（可以讨论下，如何做到有效控制写入消息？）

3、消息消费

- (1) 使用方式
- 在系统设计时，已考虑各个场景，并针对各场景接口生成一个唯一的topic，并依据接口的qps，接口的响应时长等关键参数，计算出需要消费者进程的个数（一旦不够，是否支持可配？？？）。
- (2) 校验消息
- 针对消息，需要校验消息是否满足缓存的条件，如校验注册的 time_limit、count_limit、enable、当前的已请求接口的qps是否能够支持，原则上不要超过给定的top_qps-avg_qps的三分之一。
- (3) 已消费查询接口qps
- 需要记录当前消息消息的接口的qps，注意该qps指的是符合校验通过条件，去调接口查询数据产生的qps。（非常重要，不能因为消费，而对原有的接口性能造成影响，做到可控，宁愿丢弃消息。）
- (4) 数据存储
- 已通过校验的消息，根据请求参数，请求接口，并缓存接口数据至hbase中（讨论下，针对cell的ttl设置的必要性，以及是否做成可配？）。
- (5) 消费监控
- 各个接口的消费速度，消息队列的积压情况，各个接口每天更新的数据情况，异常告警，企微通知。

4、服务查询

- (1) 服务方式
- 提供一个查询缓存接口的数据，并做好限流措施，针对各个接口（接入方注册已提供查询预估的qps）也需要做一下限流和降级预防处理。
- (2) 使用方式
- 接入方自己依据各自场景情况，判断是否调用查询缓存接口数据，比如可根据配置中心下发配置开关，来控制是否走缓存业务逻辑。
- (3) 服务监控
- 各个接口的查询qps监控，响应时间的监控，hbase自身的监控，异常报警，企微通知。

八、排期

功能点	进度	日期	备注
技术方案讨论	待讨论	2.1~2.5	暂定年后第一个星期
数据库表的选型与设计	待定		
缓存系统的测试集群与线上集群	待定		
系统注册功能开发	待定		
nsq消费功能开发	待定		
缓存查询功能开发	待定		
监控报警搭建	待定		
系统压力测试	待定		

系统故障演练测试	待定		
后台功能操作可视化	待定		

九、结论

待补充