后台开发通道面试陈述

梁承希(cyrilliang)
PCG-视频产品技术部-速看中心

2019/5/30

个人经历

- **2009 ~ 2016 -- 西安电子科技大学 本科&硕士**

- 2016 ~ 2017 -- 腾讯视频 - 媒资后台 后台开发
  - **乘风系统，后台开发负责人**
  - **媒资地域播控服务，项目负责人，架构设计**
  - **腾讯视频游戏推荐，项目负责人，架构设计**

- 2018 ~ 至今 -- 腾讯视频 - 速看平台 后台开发
  - **近一年考核 3 + 5**
  - **速看 APP 推荐系统（独立负责，设计&实现）**

业务介绍


关注
精选
最新
电影
金世佳
“展博”金世佳，登《我就是演员》舞台挑战徐峥作品，演技爆棚！
速看
影视综漫排行榜 发布
推荐主题
创造101
Yamy曾一百块跳三支舞，强东...
金世佳
《我就是演员》金世佳演技爆发，雨中扛原车祸真相！
推荐
我的


速看
完整版
06:41
13.1万
你可能还喜欢
美丽人生
光辉岁月
夜总会看门...
滑稽戏
评论
小眼睛糖豆
哈哈啊呜呜
2018-09-04
写评论


关于电影
看正片
豆瓣8.0+以上的佳片推荐
美国电影汇
分享
一个人最大的敌人是自己，一个人最好的朋友也是自己。在绝望的环境当... 详情
写评论...
点赞
评论
分享

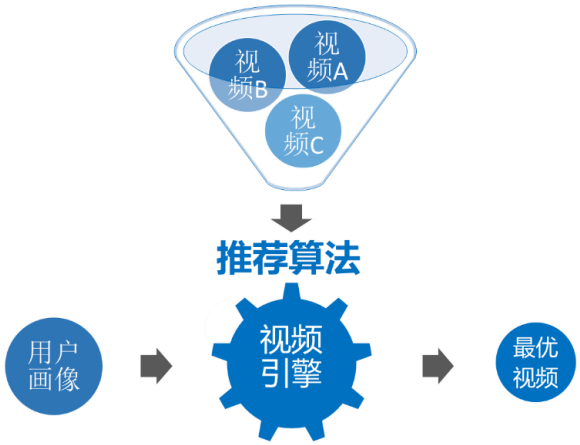
业务介绍

- **精品优质的 影视综漫 短视频内容平台**
- **主打“速看”，几分钟解说、精华版等，节约用户时间，快速消费内容**
- **主要以 feeds 流数据展示，需要进行个性化推荐**
- **多场景 多推荐数据类型 推荐系统**

上线紧迫！人力不够！

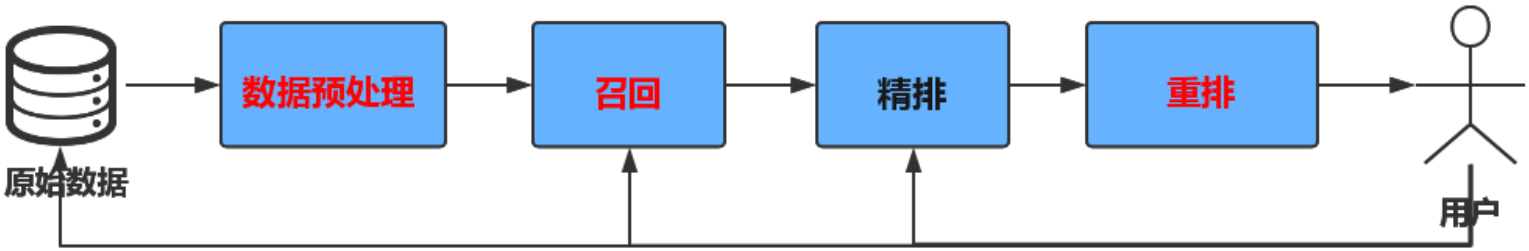
- **资料数据 独立于腾讯视频数据**
- **产品定位 是短视频内容为主，以短带长**
- 跟长视频场景不同，视频本身是短视频内容，比长视频蕴含更多信息
- 跟短视频场景不同，数据内容主要由优质CP产生，量级（百万级）相对较少，日增量K级别，难以适配短视频推荐套路

业务介绍

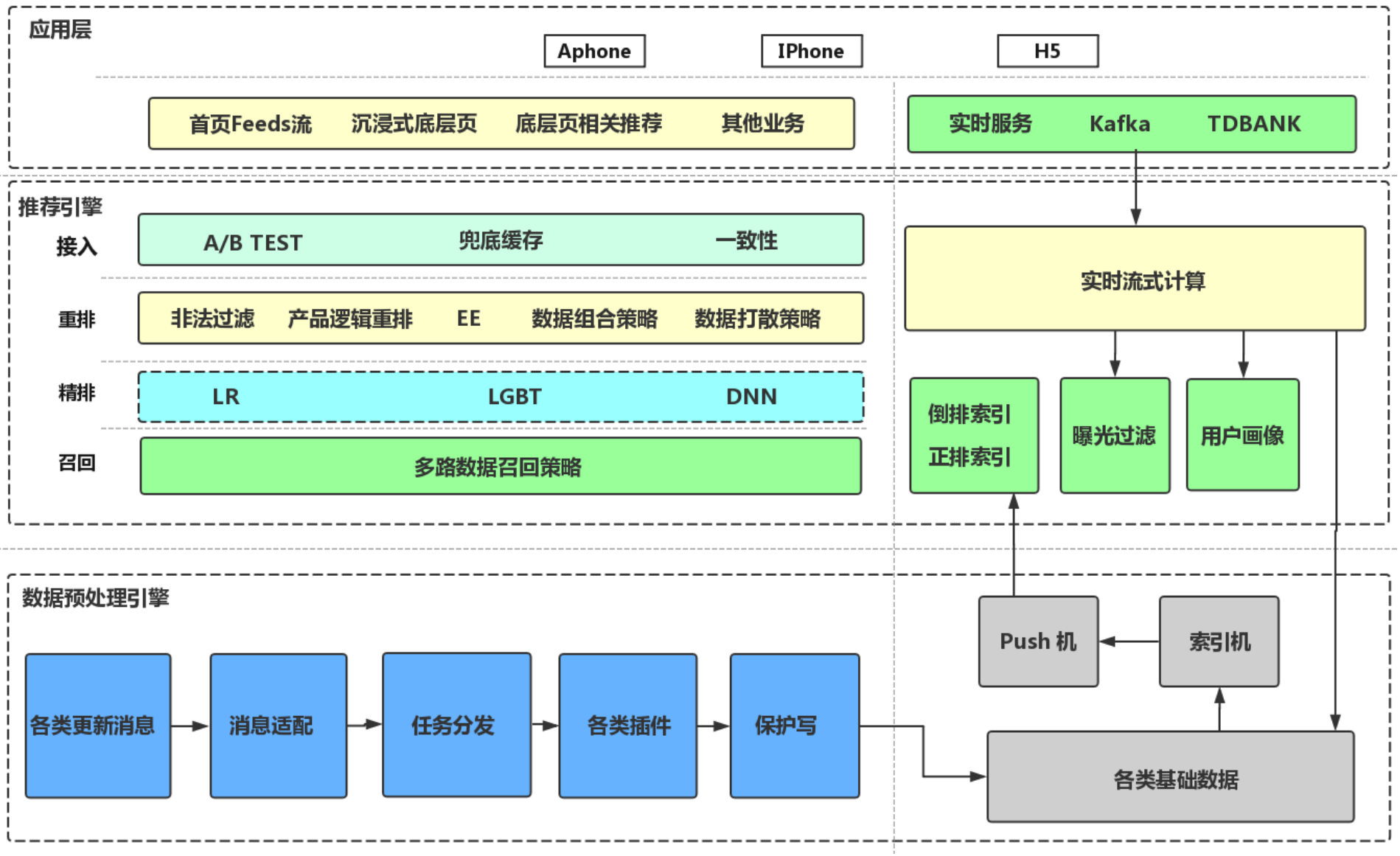


推荐核心三元素：
用户，内容，算法

我的工作：构建速看推荐系统，打通 数据内容、算法、用户的 闭环



系统架构图


应用层
Aphone
IPhone
H5
首页Feeds流
沉浸式底层页
底层页相关推荐
其他业务
实时服务
Kafka
TDBANK
推荐引擎
接入
A/B TEST
兜底缓存
一致性
重排
非法过滤
产品逻辑重排
EE
数据组合策略
数据打散策略
精排
LR
LGBT
DNN
召回
多路数据召回策略
实时流式计算
倒排索引
正排索引
曝光过滤
用户画像
数据预处理引擎
各类更新消息
消息适配
任务分发
各类插件
保护写
Push 机
索引机
各类基础数据

业务难点

① 数据召回

- 多种推荐场景差异性推荐，首页，底层页，正片页等
- 多种召回策略（最新、最热、追剧等）
- 每多增加一个召回数据源， 开发一套服务，成本高
- 召回量大（1000条合法数据），耗时要求高（平均耗时 100ms 内）

① 曝光过滤

- **短期内不能给用户推重复数据**
- **性能要求高**

① 数据预处理

- **数据结构丰富复杂，各种数据类型相互关联**
- **数据之间处理有依赖关系**
- **要求快速开发上线**

难点一：数据召回

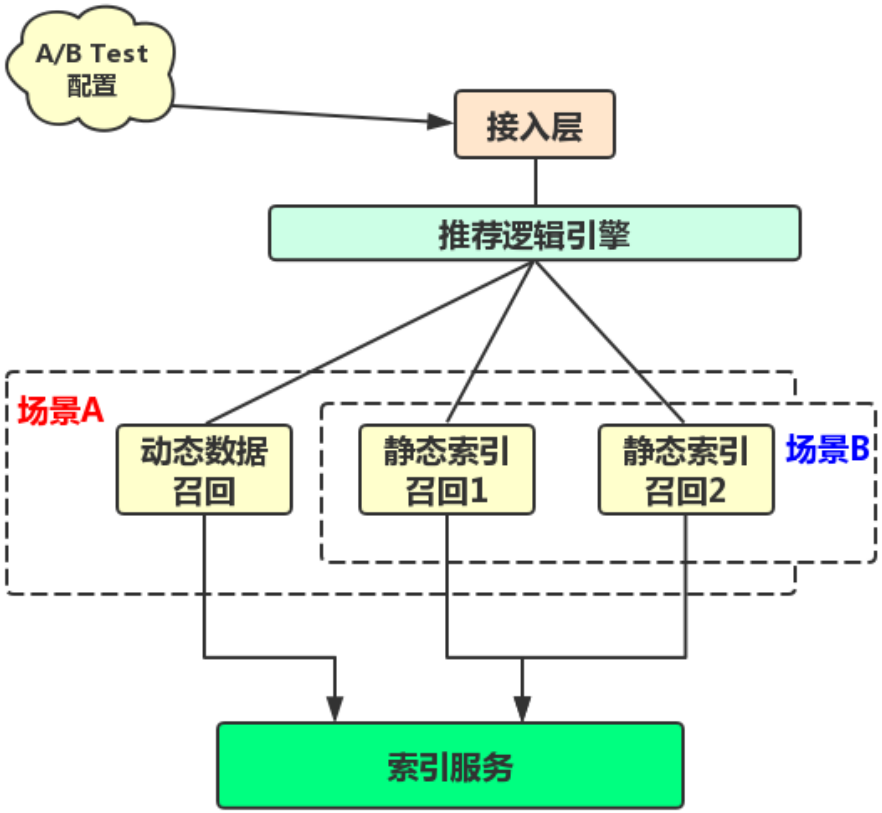
- 问题
  - **需要支持多种召回策略**
  - **场景不同，所需数据源不同，且要支持实验**
  - **模块重用性和通用性**
  - **100 ms 内耗时要保证**

- 分析
  - **多路并发召回**
  - **数据源根据场景配置化**
  - **大部分召回源都是索引式召回，少部分是特殊数据召回**
  - **数据操作，索引查找一定要快**

难点一：数据召回

- 抽象
  - **对上层提供统一的数据接口**
  - **划分为动态数据召回服务和静态召回服务**
  - **大部分召回源都是索引式召回，少部分是特殊数据召回**
  - **服务各自实现定义的逻辑接口**

- 架构方案
  - **数据源互相独立，多路召回，异步并发拉取数据**
  - **A/B Test 配置各个场景下的数据源组合**
  - **独立出来索引服务，提供倒排、正排、曝光过滤功能，索引使用共享内存存储**



难点二：曝光过滤

- **功能诉求**
  - **校验数据量大**
  - **性能要求高**

- **方案对比**

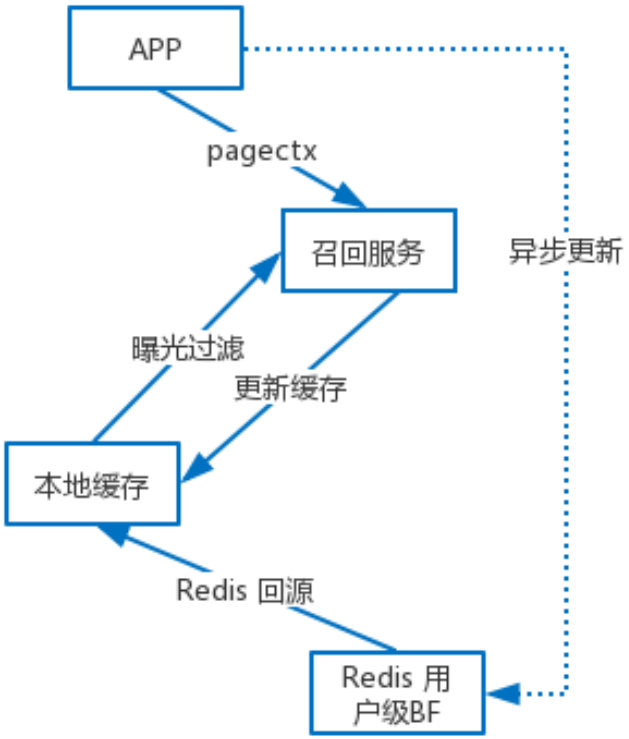
方案	支持大小	网络包量	空间耗损	时间耗损
APP缓存透传	小	大	小	中
Redis 交集	小	大	大	中
Bloom Filter	大	无	中	低

- **Redis set，直接进行差集计算，召回计算耗时 10 ms/万条**
- **使用 BF，对用户ID进行，一致性hash，共享内存，召回计算耗时 3 ms/万条**

- **遭遇问题**

  需要扩缩，数据迁移需要同步各个机器上的BF数据，短时间内，近期曝光过滤失效

难点二：曝光过滤

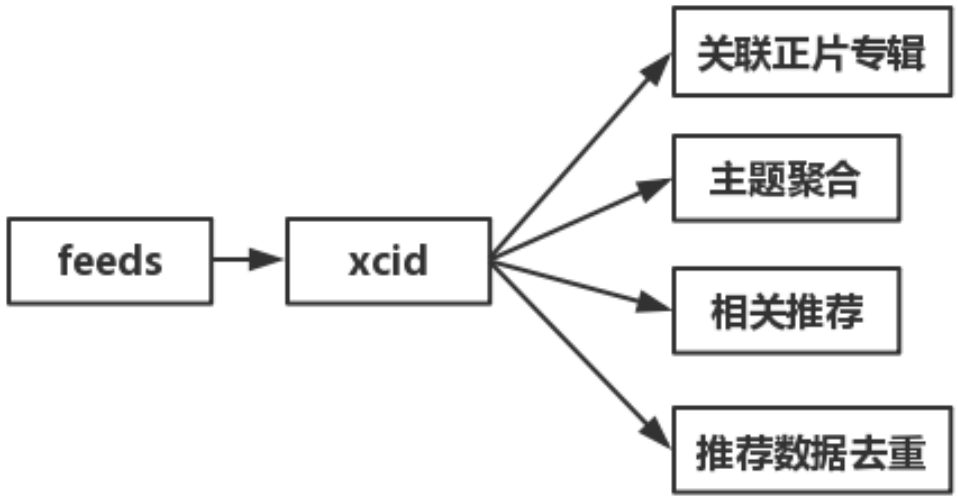


- 分析
  - 本地缓存不能集中化，用户存储**粒度要打散**
  - 同时要保证**实时性**和**高性能**

- **最终方案**
  - **客户端缓存+本地缓存**，app pagectx 携带上一次曝光的少量数据，合并缓存数据做check
  - filter内数量达到阈值或缓存超时，回源redis
  - redis 内的数据非交互关键路径，ms级更新
  - 动态扩展，数据压缩，节省空间

- **效果**
  - 总体空间缩小 **80%**，按需分配
  - 平均召回耗时 **2ms / 万条**
  - 控制粒度更细（个体和策略两个维度）

难点三：数据预处理



- **重要性**
  - **xcid 是多个数据的纽带**
  - **xcid 在推荐逻辑中有重要作用，直接决定了是否可入推荐池**
- **问题**
  - **原匹配规则由产品制定，从各类关联关系找到关联的xcid，但有很多数据会关联不上，匹配率 50% 左右（表格说明）**
  - **综艺&动漫无版权大多数数据关联不到 xcid**

难点三：数据预处理

提升效果的重要手段
除了精湛的技术外，对业务的理解也十分重要
——用巧不用蛮

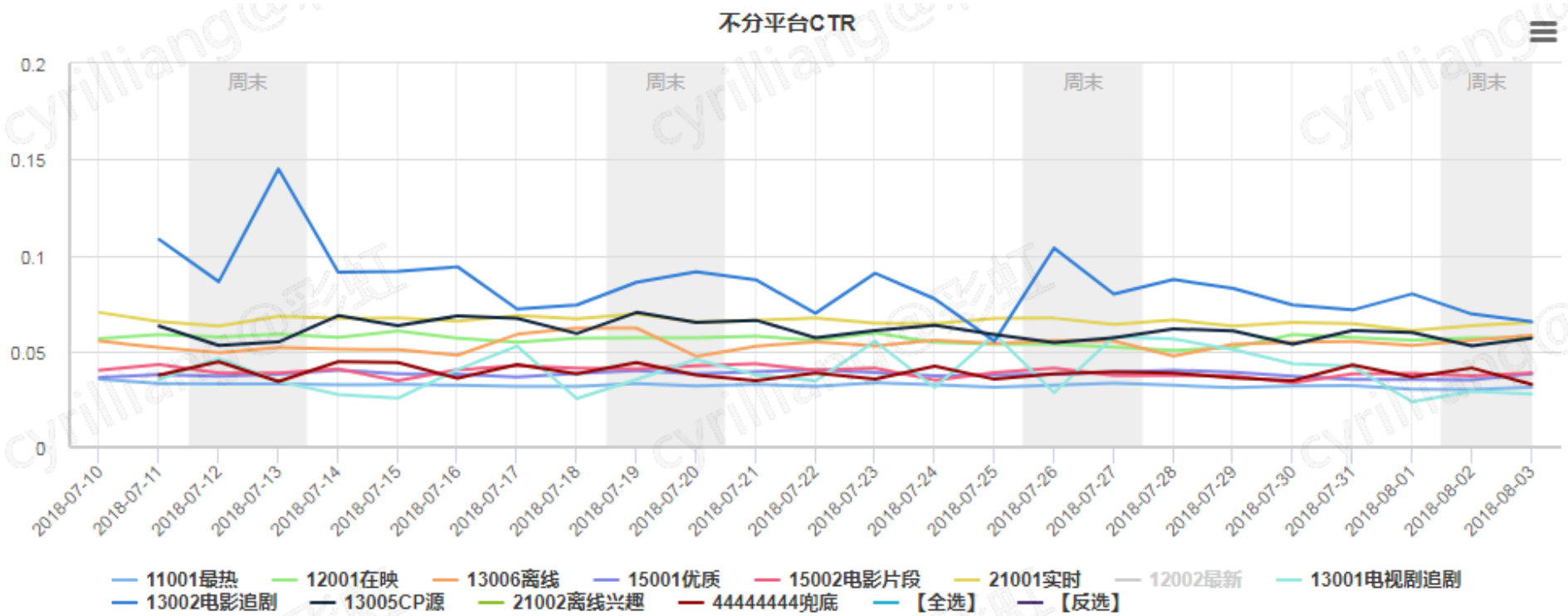
- **分析**
  - 跟踪编辑的操作方式和流程，思考非常规化方案
  - 使用规则的方法准确率高，但是覆盖率低，有没有方法去提升覆盖率
  - xcid 和 feed 上有很多隐性的数据没有用到，如文本、标签等

- **解决方案**
  - 使用规则方法已关联的数据训练出文本标签 -> XCID 的映射模型
  - 使用规则的方法准确率高，但是覆盖率低，有没有方法去提升覆盖率
  - xcid 和 feed 上有很多隐性的数据没有用到，如文本、标签等

- **效果**
  - 匹配耗时秒级别->毫秒级别；
  - 综艺&动漫类目下的匹配率由之前的 20%，提升到 85%，准确率在 95% 以上

运营数据

多数据源召回支持



运营数据

提升数据质量对推荐效果的影响

其他工作内容

培养&培训

- 培养人：stantu、victortang、sullivanzhu、roysun
- 部门课程：从0到1搭建推荐系统
- 部门课程：乘风系统方案介绍

KM文章

- 《网络高性能服务的本质探究》
- 《速看推荐系统搭建》
- 《Let's GO》
- 《初探函数式编程》

公共组件开发&分享

- Python Hippo 生产者&消费者组件
- Python JCE 服务 RPC 工具
- Python 通用开发工具库
- SPP 微线程多类型并发工具库
- Elastic Search 快速导数据工具

谢谢！

索引 TOP K 拉取过程

- 抽象
  - **方案1：顺序 check，直到拉够**
  - **方案2：随机拉取，去掉不合法的（算法不确定）**
- **索引上获取足量数据**
  - **方案3：记录上次索引 offset，避免无效比较**

运维信息——使用机器情况

XX层：X台机器

索引建立&PUSH

索引文件 2G，百M ？ min

用到组件 kvm

rsync + inotify

索引机-》push机，批量push

曝光过滤-》comm_cache，多阶 hash

BF&动态BF实现详细说明

函数个数N的确定 m/n ln(2)

自动扩容BF实现