斗鱼直播实时风控引擎快速对抗探索实践

演讲人-李瑞-斗鱼直播-风控负责人

DataFunSummit # 2023

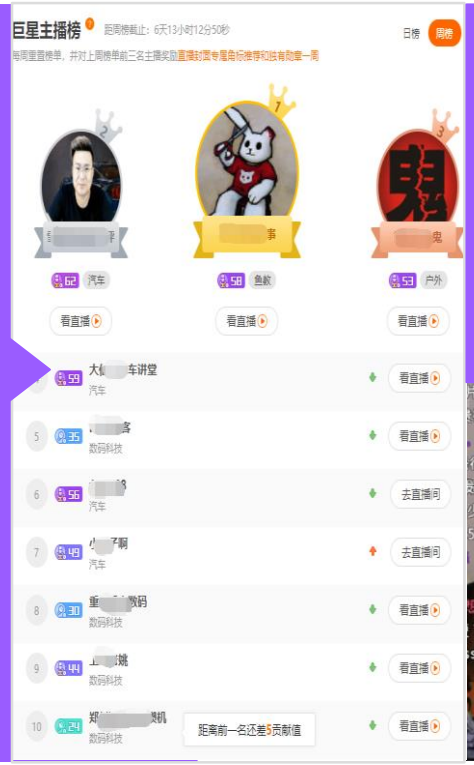
目录
CONTENT

01

直播行业的黑产问题

直播行业常见的黑灰产问题

主播刷榜
虚假开播



广告引流
低俗辱骂

渠道作弊
活动欺诈

欺诈充值
电信诈骗

业务安全的痛点

木桶效应

如果不掌握所有的用户行为入口和数据，总会出现防范的短板，无法识别出黑产账号，也无法有效支撑业务安全。

业务对接成本高

不同类型的业务需要独立的风控名单/接口服务，相应的风控策略也不同，每个业务的策略服务如果单独开发效率低，并且配置凌乱难以管理。

性能要求高

- 对接业务众多，吞吐量巨大，RT不能影响业务
- 实时计算时效性要求高

用户体验差

- 用户被风控后缺少反馈途径的引导。
- 投诉反馈排查效率低、耗费风控人员精力

防御时效性差

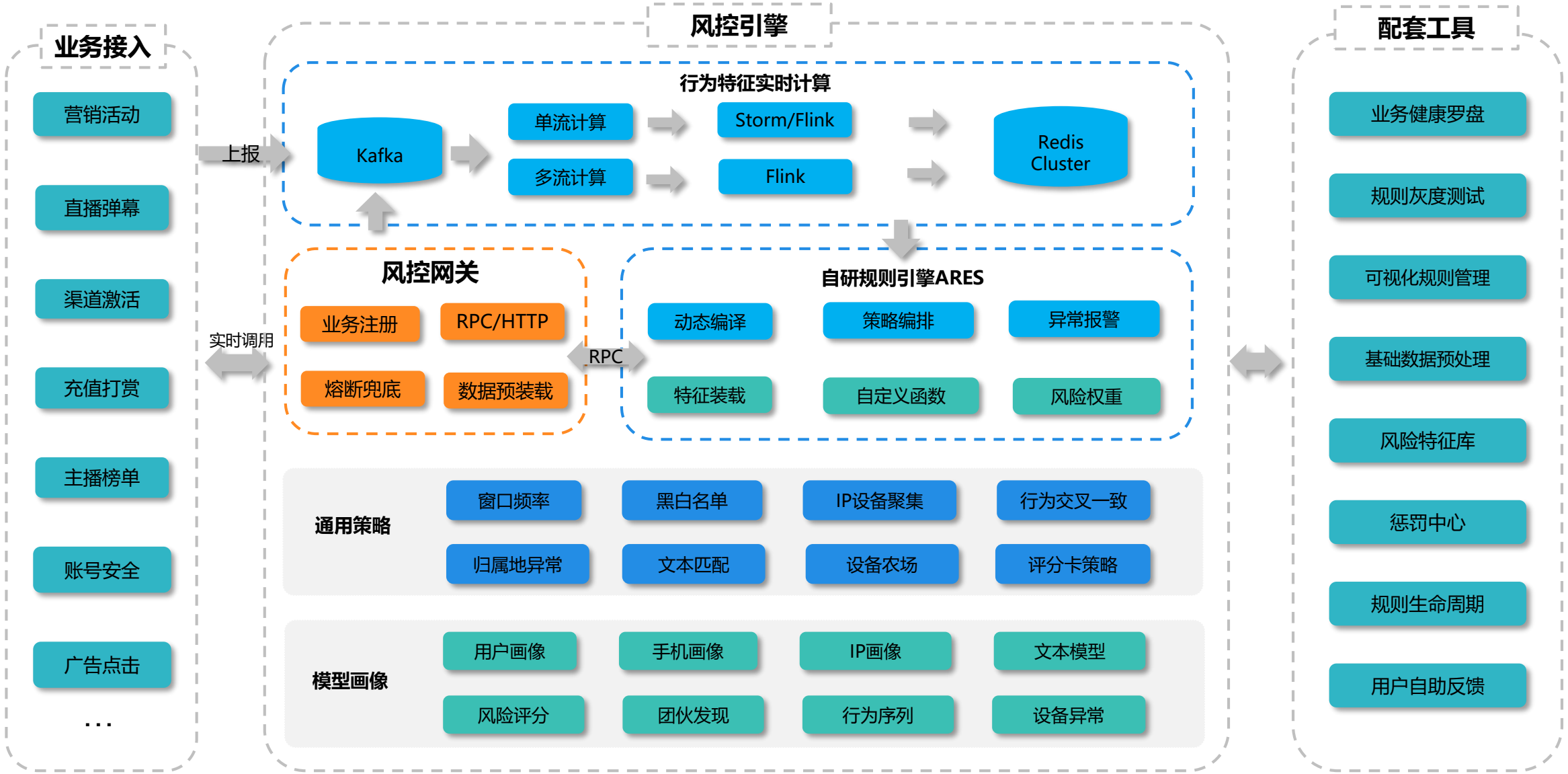
- 风险感知能力不全面，风控迭代慢
- 实时性策略较少，依赖离线挖掘周期长。

02

全栈式风控引擎的建设

DataFunSummit # 2023

全栈式风控引擎架构


业务接入
营销活动
直播弹幕
渠道激活
充值打赏
主播榜单
账号安全
广告点击
...
上报
实时调用
风控引擎
行为特征实时计算
Kafka
单流计算
多流计算
Storm/Flink
Flink
Redis Cluster
风控网关
业务注册
RPC/HTTP
熔断兜底
数据预装载
RPC
自研规则引擎ARES
动态编译
策略编排
异常报警
特征装载
自定义函数
风险权重
通用策略
窗口频率
黑白名单
IP设备聚集
行为交叉一致
归属地异常
文本匹配
设备农场
评分卡策略
模型画像
用户画像
手机画像
IP画像
文本模型
风险评分
团伙发现
行为序列
设备异常
配套工具
业务健康罗盘
规则灰度测试
可视化规则管理
基础数据预处理
风险特征库
惩罚中心
规则生命周期
用户自助反馈

全栈式风控引擎降低对接成本

业务收拢

强运营强宣发，推动业务对接，解决木桶效应。

低成本一站式接入

业务注册调用后，就可以获得全面的风险管控、监控告警、反馈排查等配套服务支持。

业务方注册

监控、调优

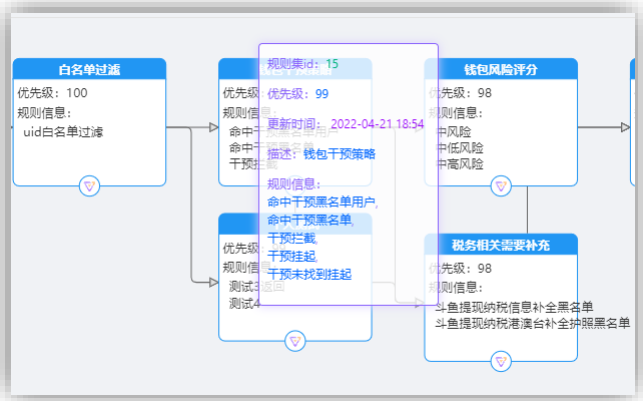
全栈式风控引擎降低对接成本

可视化引擎

- 勾选编辑上手**门槛低**
- 常用策略模型**高度复用**
- 策略发布流程规范
- 支持高度定制类规则
- 高可用高性能

可视化编辑

决策流



定制类策略

通用型策略

实时订阅

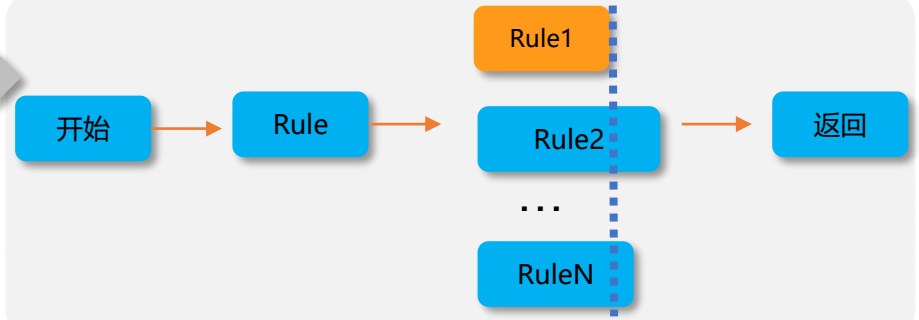
规则编译动态加载

规则文件管理器 → 元信息 JavaFileObject → 源码编译器 → 字节码 JavaFileObject → Class对象 → 注入rpc服务

微服务装载

惩罚中心	风险标签	用户信息/地址解析
风险评分	团伙属性	三方服务

执行计划编排



超时兜底 | 流量控制 | 风险权重 | 对外话术 | 日志收集

智能风控：风控引擎与机器学习平台打通

斗鱼 DOUYU.COM DataFun.

智能风控

- 增强欺诈识别能力
- 提升风险感知能力
- 策略评估更准确
- 提升对抗效率

风控引擎与算法平台打通

实时引擎调用模型微服务

- 实时团伙服务
- 实时评分服务
- 设备异常服务
- 文本实时预测
- 行为序列标签
- 模型解释话术

配套工具后台

- 团伙管理
- 评分管理
- 设备查询
- 行为序列查询

自动分析

- 异常根因分析
- 自动规则生成

机器学习平台

- 算法框架：Tensorflow、SparkML、Pytorch、...
- 任务调度：公有云打通、Docker
- 特征工程：特征构造、特征计算
- 版本管理：升级回退、在线预测
- 模型部署：准召率评估、一键上线

策略融合

- 风险评分：融合评分、单场景评分、团伙评分、白评分
- 团伙发现：可解释性、团伙标签
- 垃圾文本：变体内容、行为属性
- 风险设备：设备标签、唯一性检测

算法层

- 风险评分：GBDT+LR、DeepFM
- 团伙发现：图算法、自研无监督
- 垃圾文本：TextCNN、Wide&Deep
- 风险设备：IForest、自研指纹
- 行为序列：Transformer、自研团伙序列

智能风控：提升对抗效率

监控到波动

提取特征

- **用户信息特征**：注册地、用户等级、手机归属地、区号、手机参数...
- **行为特征**：行为IP/设备、行为类型个数(行为路径)、沉默天数、最近登陆地...
- **风控画像特征**：uid/ip/设备风控记录、团伙分数、用户评分

规则提取

挖掘算法：决策树(GBDT)、KS、方差分析、分位数

提取策略

是什么：哪个IP、哪个手机区号

粒度：　地点：国/省/市, 时间：月/日/时

阈值：　连续：= ≥ ≤ ,　离散：in =

效果校验

策略:
注册月份= ‘202212’ && 沉默天数≥110 && 沉默天数≤120
&& 今日行为类型数==3

评估指标：高质量白名单误杀占比(准确率)
识别量(召回率)

- **提升效率：减少了 监控>排查>策略上线 人力与时间**
- **减少监控噪音，提升监控准确性**

高吞吐设计-行为指标实时计算

诈骗：user.level<10 && user最近n分钟观看房间数==0 && user近m小时订单金额≥1000的扫码ip去重数≥3

基础信息　　单流滑动窗口　　多流滑动窗口

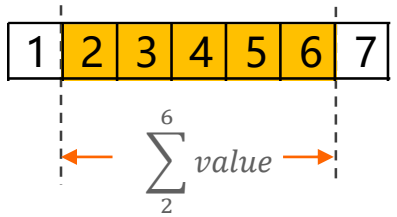
基础信息　RPC调用，两级缓存：caffeine + redis

单流滑动窗口　kafka 12亿/日 → Storm/Flink 100亿/日 → redis 400G

- count
- sum
- avg
- count distinct

1. 滑动窗口选型
   - flink sliding window
   - redis 分钟key拼接

```
select concat(hop_start, uid) as key, count(distinct room_id) from chat_session_kafka
group by HOP(dateline, INTERVAL 1 MINUTE, INTERVAL N MINUTE) ,uid
```



优点：窗口大小灵活，占内存极少
缺点：redis写操作频繁

内存预聚合，
定时刷入redis-cluster
lettuce异步提升并发

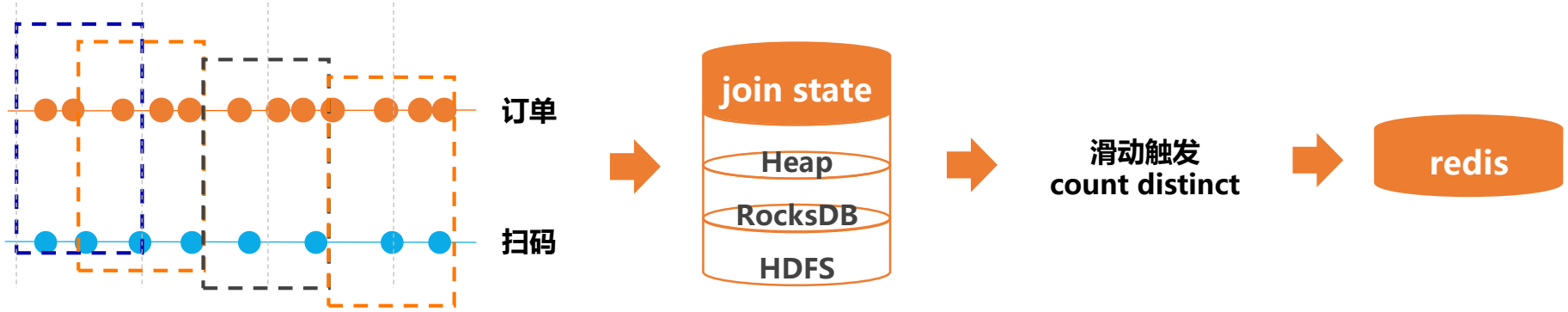
2. 大窗口去重计算
   - HashSet
   - BitMap
   - 布隆过滤器
   - Hyperloglog

功能\算法	HashSet	BitMap	布隆过滤器	HyperLogLog
是否精准	Yes	Yes	误报率低	0.81%误报率
存储消耗	大	小	小	小
效率	快	快	快	快
支持count distinct	Yes	Yes	No	Yes
支持合并去重	Yes	Yes	No	Yes
任意维度count distinct	Yes	不可控	Yes	Yes

高吞吐设计-行为指标实时计算

多流滑动窗口


```
select concat(hop_start, 订单.uid) as key, count(distinct 扫码.ip)
from 订单 join 扫码 on 订单.orderId=扫码.orderId AND 订单.dateline between 扫码.dateline - 60秒 AND 扫码.dateline
where 订单.金额 >= 1000 group by HOP(订单.dateline, INTERVAL 1 MINUTE, INTERVAL N MINUTE)
```



高吞吐设计-规则引擎选型

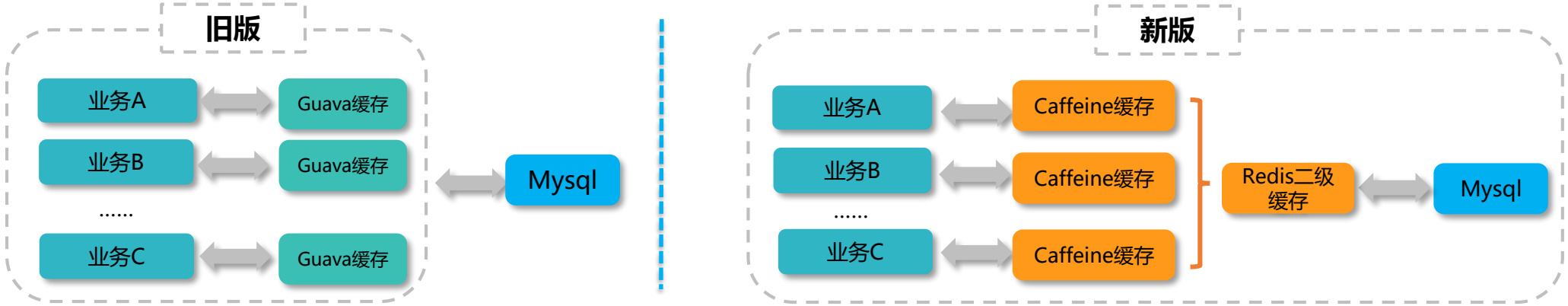
斗鱼 DOUYU.COM DataFun.

引擎选型预研

执行策略	groovy	groovy（@CompileStatic）	aviator	Drools7	java
dubbo（10000次）	12519	/	12638	13214	11670
dubbo（100000次）	111390	/	113666	114066	105755
dubbo（1000000次）	1086102	/	1101762	1188882	1052743
逻辑运算（10000000次）	739	345	2895	2270	321
逻辑运算（100000000次）	6771	3238	23377	22695	2469
递归（O(2^n)）(n=40)	18718	7061	/	7298	6626

- 源代码性能最好
- 规则检错机制友好
- 迁移成本低

高吞吐设计-预装载缓存优化



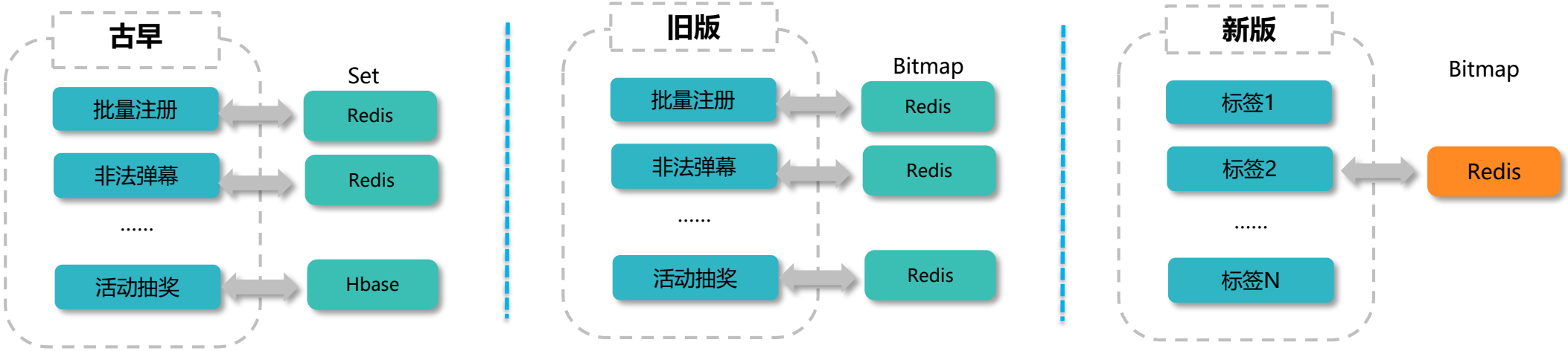

```
@Cacheable(value = "u_i", depict = "用户信息缓存", enableFirstCache = true,
        firstCache = @FirstCache(expireTime = 7200 , timeUnit = TimeUnit.SECONDS ,
initialCapacity = 60000, maximumSize = 60000),
        secondaryCache = @SecondaryCache(expireTime = 18000, timeUnit = TimeUnit.SECONDS,
isAllowNullValue=true))
public UserInfo getUserInfo(long uid){
            …
}
```

请求量：日均2.5E

平均耗时：5ms ->1.5ms

缓存命中率：32%->87%

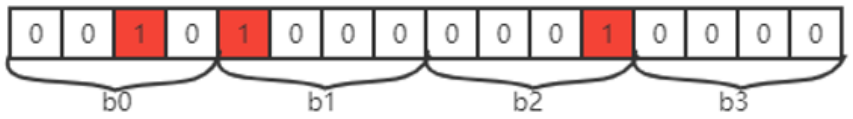
高吞吐设计-风险标签存储优化



古早时期，每种异常行为一个set，也有使用Hbase、Mysql等DB, 空间浪费、管理混乱、慢查询较多

统一使用Redis管理，每种异常标签一个bitmap，缩短查询耗时、减少存储空间，但读取多种标签，就会产生多次IO

- 一个用户开辟一个bitmap, 一把读出所有风险标签
- 所有用户存放在多段bitmap中，分桶存储，进一步节省key开销

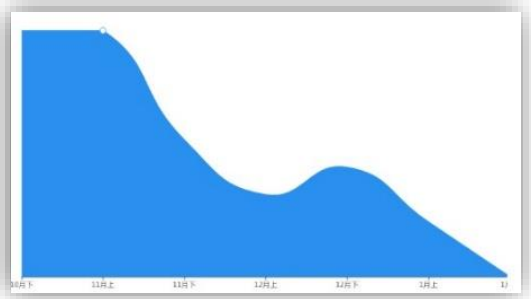


一个用户占位256bit

平均耗时：20ms ->6ms

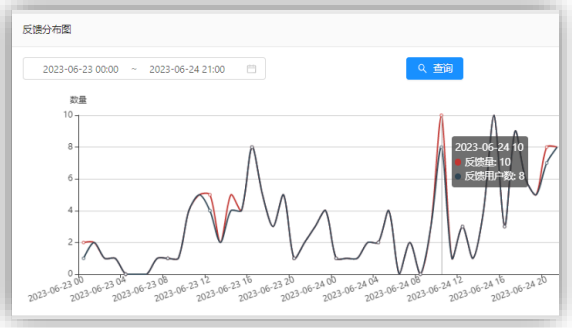
内存使用：250G->30G

提升用户体验



策略优化闭环

持续优化风控策略模型，推导策略实际准确率，分析命中规则误杀情况



客诉量监控

实时监控客诉量，将客诉量维持在较低的水平

系统自动解决客诉

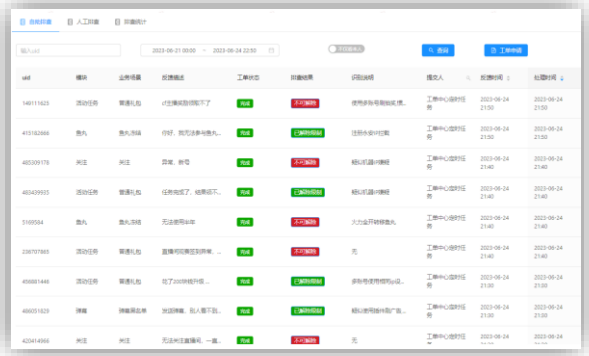
根据命中策略风险程度、团伙规模等自动判定是否解除限制

友好的风控引导提示

- 明确用户行为受限原因
- 提示违规行为的影响
- 提供申诉入口

用户自助申诉

- 用户自主申诉替代人工客服，节约人力提升效率



03

文本识别对抗实践

文本识别挑战

1. 广告变体

- 谐音变体、象形变体、拆字变体
- 联系方式字母数字变体字符
- 联系方式符号间隔
- 拼音混合
- 表情符号代替文字

主播簧薇 **ET28六八**

加 **薇 信** **②⑥O②⑨O⑦③O7**

本人私房【**大chi 度**】激晴自拍视频!

想看加**徽亻 言** baby**1╋3╋7╋8╋7╋0╋2╋0╋1╋8╋0**

点我**头像** 让你爽

❣❣站 À ｊ 6 ， ⓠ ℂ

☺篁沚 Ｊ ９ ᴀ • ⁙ ⁙

2. 低俗辱骂变体

- 谐音变体、象形变体、拆字变体
- 拼音缩写
- 拼音同与同音词混合
- 表情符号代指

p研

拉链夹到蛋

zao屎zao，超生

一拳大事你👹

司马

没母

你顶的我好爽

文本识别服务架构

服务层

事前拦截 | 事中/事后人审 | 错检/漏检监控 | 误杀降级兜底 | 内容回溯平台

算法策略

预处理	规则识别	敏感词匹配	ML/DL模型
标点符号	正则匹配	硬词匹配	char2vec+textcnn
表情符号	字母数字占比	谐音匹配	word2vec+textcnn
拼音特征	文本相似度	模糊匹配	Wide&Deep
特殊符号映射	异形字占比	自动提炼关键词	Bayes

模型管理

语料标注 | 模型自动训练 | 准召率评估 | 样本管理 | 模型版本管理

数据层

弹幕 | 昵称 | 帖子 | 频道聊天 | 私信 | 标题 | 签名

自研敏感词匹配算法

挑战

- 敏感词通配符 ? * 通配逻辑实现
- 通配长度↑ 误杀率↑ 风险↓ ，通配长度↓ 误杀率↓ 风险↑，最大通配长度需在各个场景、时期、用户上分别配置
- 数十万敏感词，调用量大，直接影响C端用户体验，耗时敏感

技术选型

维度 \ 算法	字符串Contains	普通正则引擎	Hyperscan多模正则	AC算法
时间复杂度	O(m×n)	O(m×n)	O(m+n)	O(n)
空间复杂度	O(m)	O(m)	O(m)	O(m)
初始化耗时	低	高	高	低
增量添加删除	√	×	×	√
通配支持	×	√	√	×

其中n为待检测文本长度，m为模式串（敏感词）集合的总长度

自研敏感词算法

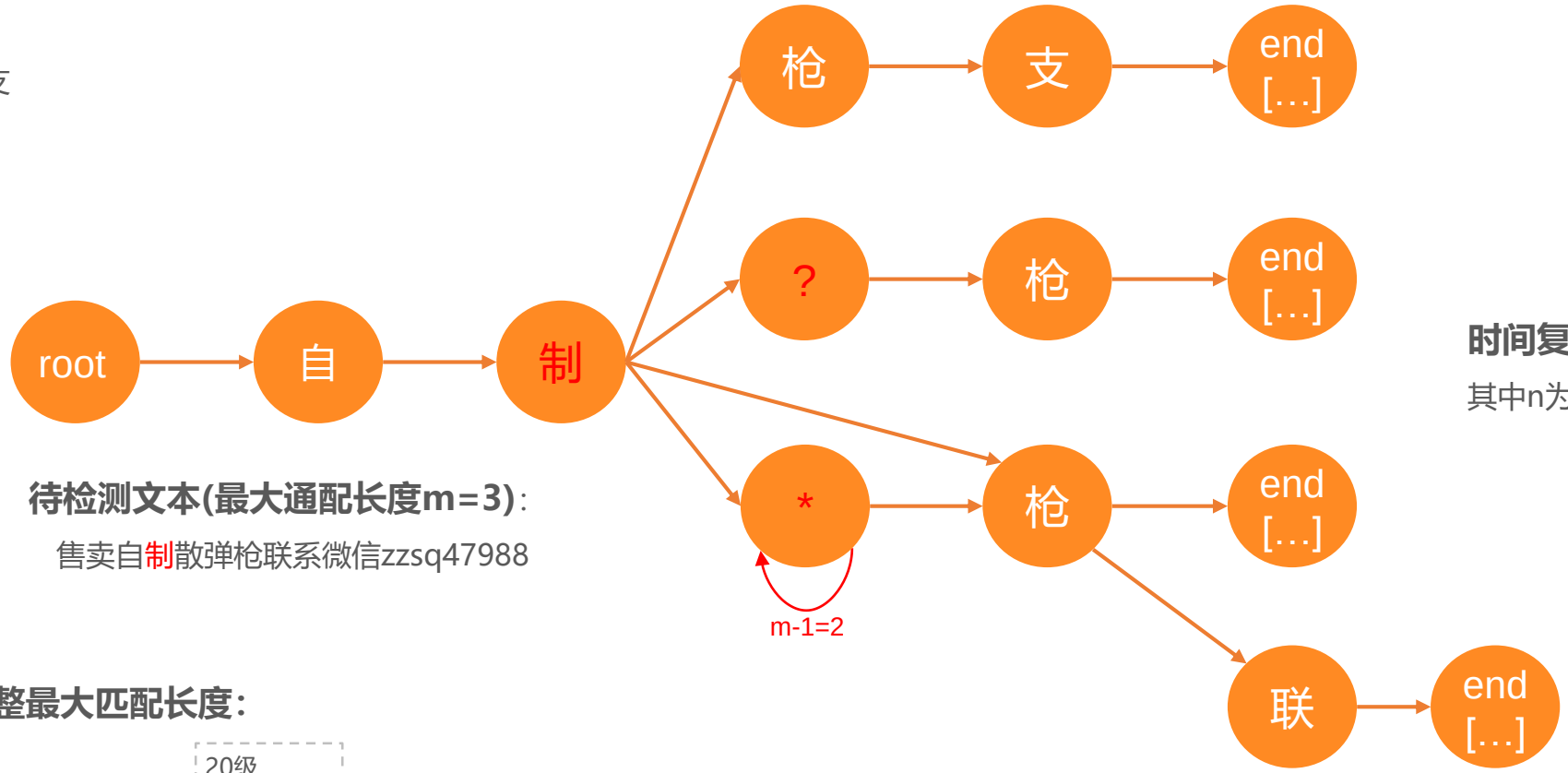
基于NFA的通配敏感词匹配算法

敏感词:

自制枪支

自制?枪

自制*枪



待检测文本(最大通配长度m=3):

售卖自制散弹枪联系微信zzsq47988

时间复杂度：O(n)，

其中n为待检测文本长度

动态调整最大匹配长度：

敏感词：**群*懂*的**

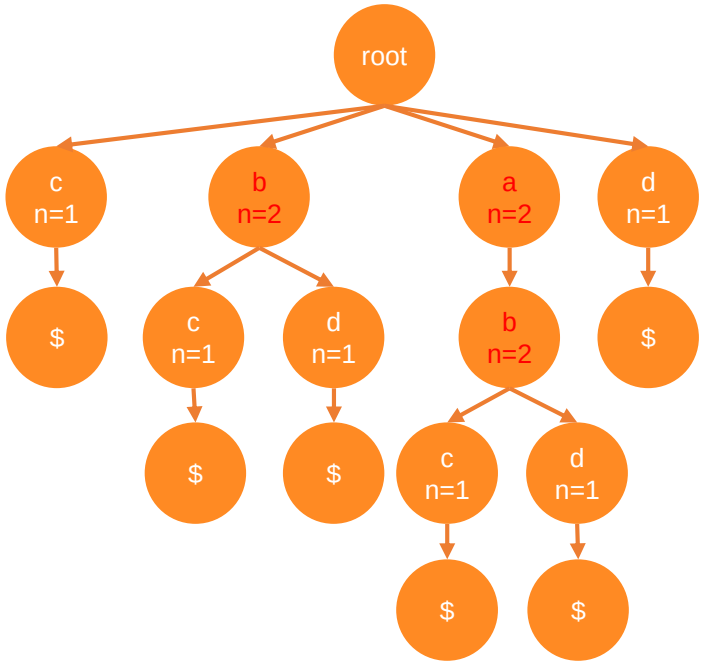
20级
白用户
风险评分5
→ m=2 → 粉丝**群**说啥了，**懂的**人解释一下 → 不命中

2级
非白用户
风险评分80
→ m=15 → 加**群**79648729都**懂❤口的** → 命中

敏感词自动发现

FROM	TO
À	a
Á	a
Â	a
Æ	ae
𝐚	a
j̈	j
ĵ	j
ⓒ	c
𝟙	1
𝟚	2
...	...

变体字符字典（约3k）



后缀树输入[“abc”，“abd”]
得到公共子串：a、b、ab

异形字公共子串

a2 j, 都俛的
a2 j̈, 都俛的
鑝a2j,ⓒⓒ
a2 ĵ cc都俛的

a2jcc都俛的
a2jcc都俛的
鑝a2jcc
a2jcc都俛的

联系方式公共子串

Q群六210、382直接来
来球群6210ℨ82看刺激的
Q群Б21零3 ⑧ 2周姐的视频

q6210382
6210382
q6210382

[a2jcc都俛的, a2jcc]
[2jcc都俛的, jcc都俛的, cc都俛的]
[6210382, 210382]

a2jcc
jcc都俛的
6210382

长度大于5、重复3次及以上

04
思考与展望

思考与展望

1. 自动分析目前还处于半自动挖掘

虽然自动化分析可以给出初步的风险策略建议，但还不够成熟，存在特征重复、阈值不合理、召回率较低等问题，需要持续迭代优化

2. 拥抱向量检索

- 文本相似检索
- 违规行为匹配

3. 大模型应用

- 大模型识别文本变体的能力显著
- 当前特征标签本身还是人工维护创建的，受限于人员的思路宽度，存在无法召回的情况，是否可以借用大模型自动化构建特征标签和策略？

感谢观看