

# 小红书大数据在推荐中的应用

郭一

小红书推荐架构负责人

# 小红书：找到你想要的生活



8500万

月活

300%

年同比增长

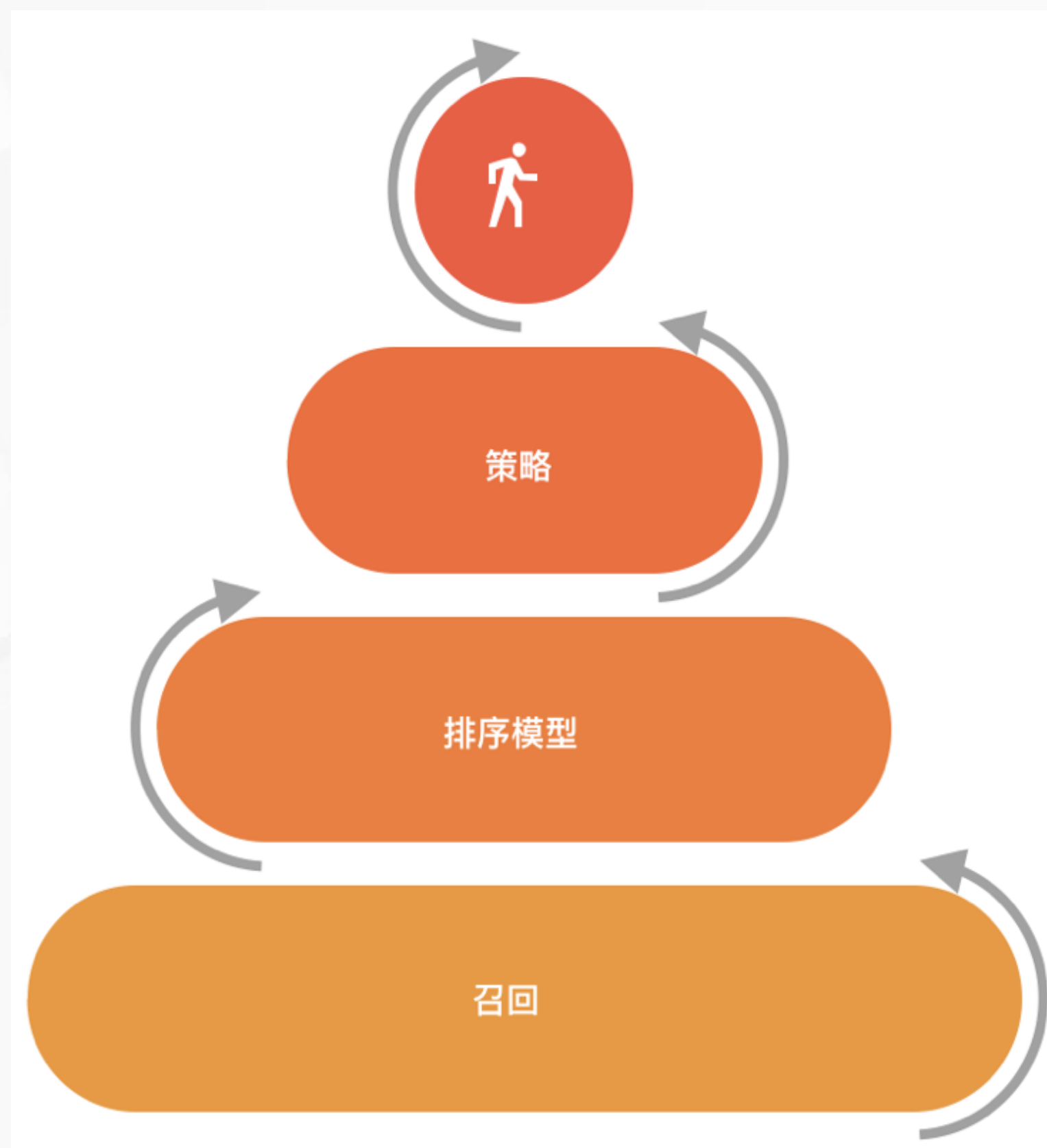
30亿

每天笔记展示

## 目录

- 实时计算在推荐业务中的场景
- 实时归因: 一个推荐业务的Flink实时计算任务的实践
- Red Flink实时流计算平台
- Red ML平台
- 总结和展望

## 推荐 & 实时计算



## 线上推荐过程

分页显示, 有交互之后重排

笔记位置多样性调整

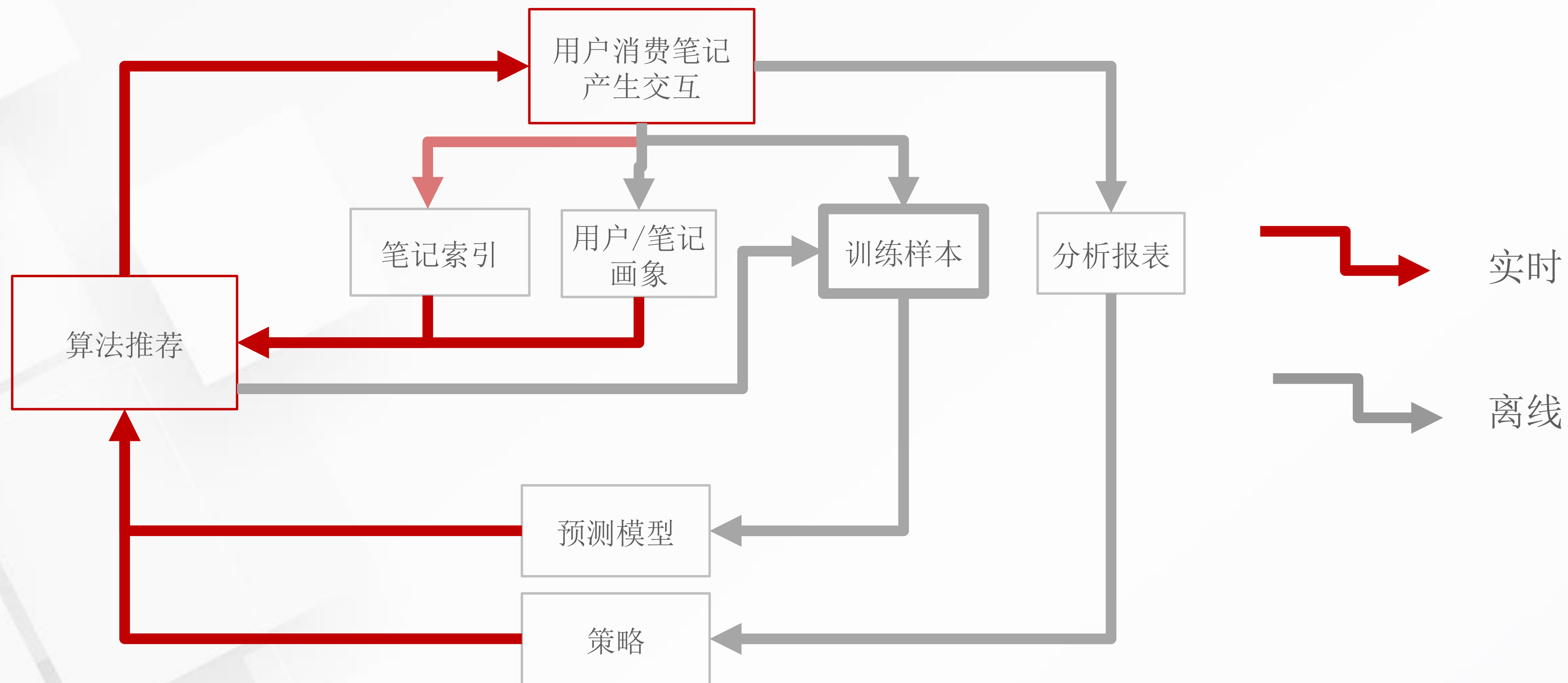
给每个笔记打分

$\text{Score} = \text{pCTR} * (\text{pLike} * \text{Like权重} + \text{pCmt} * \text{Cmt权重} \dots)$

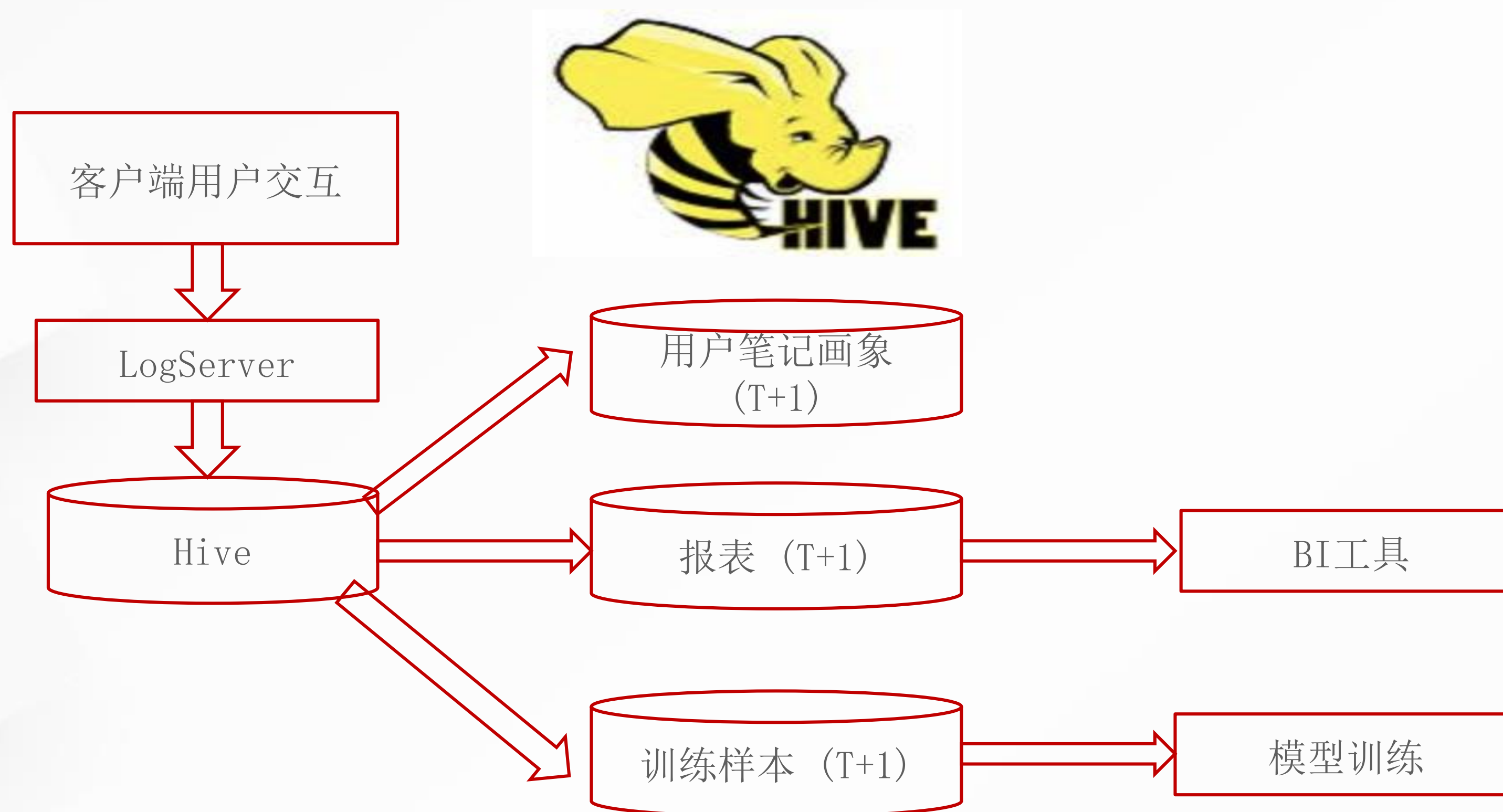
根据用户画像从笔记池中召回（多种策略）



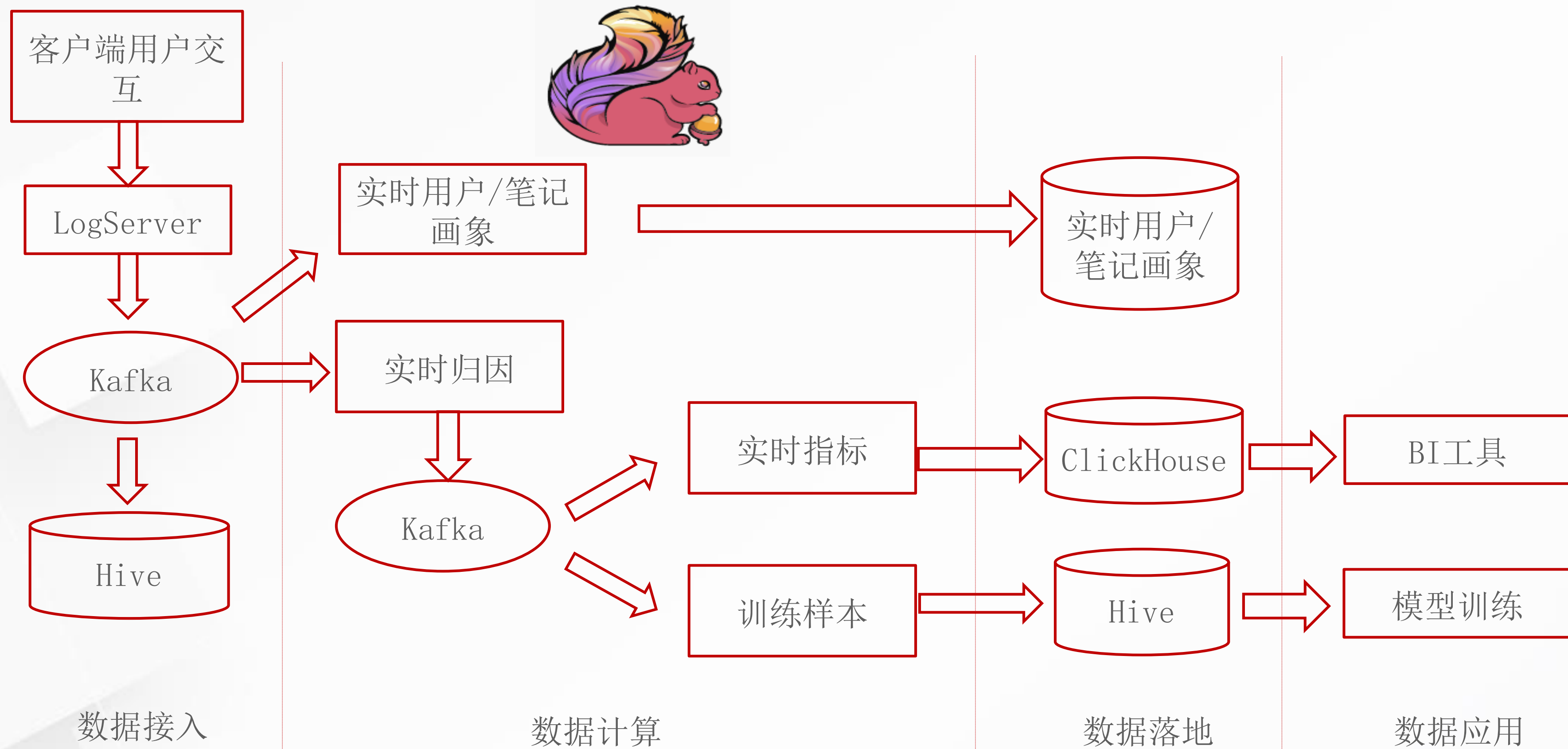
## 推荐系统架构



## 离线批处理



# 实时流处理





# 实时归因

# 实时流计算：实时归因



1. 曝光

2. 点击

3. 点赞

4. 点击

## 用户行为流:

笔记1 曝光  
笔记2 曝光  
笔记3 曝光  
笔记4 曝光  
笔记2 点击  
笔记2 点赞  
笔记2 回退, 时长20s  
笔记4 点击  
笔记4 回退, 时长10s  
...

点击模型数据标签

曝光笔记	有无点击
1	No
2	Yes
3	No
4	Yes

点赞和时长模型数据标签

点击笔记	有无点赞	停留时长
2	Yes	20s
4	No	10s

## Flink Job – Session Labeler

- Flink任务
  - Kafka Source => Kafka Sink
  - keyBy (user\_id, note\_id, 曝光/点击)
  - 使用Process Function API  
(SessionStateProcessor)
- SessionStateProcessor
  - 创建一个定时的20分钟窗口
  - 创建维护ValueState<SessionState>状态
  - 窗口结束的时候
    - 根据SessionState输出下游记录
    - 清除ValueState<SessionState>

## 实际生产需要解决的问题

- Flink 集群管理
- Checkpoint & 状态持久化
- Backfill



# Flink集群部署 on K8s (Standalone)

Flink 集群管理 App 配置 后端日志业务 Flink 任务管理 Flink 数据字典 Schema 变更 审计 用户管理						
列表 (49) 创建 选中的- Search: 集群名称, 报警:						
		集群名称	任务状态/监控	是否 HA	Status	报警
	查看详情 编辑 只看此项	<div>tk-ch-test</div> <div>Total CPU Core 10</div> <div>Total Mem 10GB</div> <div>Num TM 1</div> <div>Total Slots 10</div> <div>40.0%</div> <div>K8S JobManager Pod</div> <div>K8S TaskManagers Pods</div>	<div>RUNNING Flink Web Flink Metrics</div> <div><div>Home</div><div>Home Dashboard</div></div>		run_job_success	Emf
	查看详情 编辑 只看此项	<div>anti-es-log</div> <div>Total CPU Core 4</div> <div>Total Mem 4GB</div> <div>Num TM 2</div> <div>Total Slots 8</div> <div>75.0%</div> <div>K8S JobManager Pod</div> <div>K8S TaskManagers Pods</div>	<div>RUNNING Flink Web Flink Metrics</div> <div><div>Source Record Latency by Partition P99</div><div>1.0 ms</div><div>0.5 ms</div><div>0 ms</div><div>-0.5 ms</div><div>-1.0 ms</div><div>No data points</div><div>current</div></div>		init_deploy_success	p1
	查看详情 编辑 只看此项	<div>dzy-session-v2</div> <div>Total CPU Core 336</div> <div>Total Mem 480GB</div> <div>Num TM 12</div> <div>Total Slots 240</div> <div>75.0%</div> <div>K8S JobManager Pod</div> <div>K8S TaskManagers Pods</div>	<div>RUNNING Flink Web Flink Metrics</div> <div><div>Home</div><div>Home Dashboard</div></div>		init_deploy_start	Emf



## Checkpoint 和 State 持久化

FsStateBackend	RocksDBStateBackend
<ul style="list-style-type: none"><li>• state 存储在HDFS</li><li>• 不支持增量checkpoint</li></ul>	<ul style="list-style-type: none"><li>• state 存储在RocksDB on HDFS</li><li>• 利用RocksDB的compaction机制合并，支持增量Checkpoint</li><li>• 适用于state比较大的情况</li></ul>

## RocksDB参数挑优

- 磁盘I/O高，反压上游
- 内存OOM

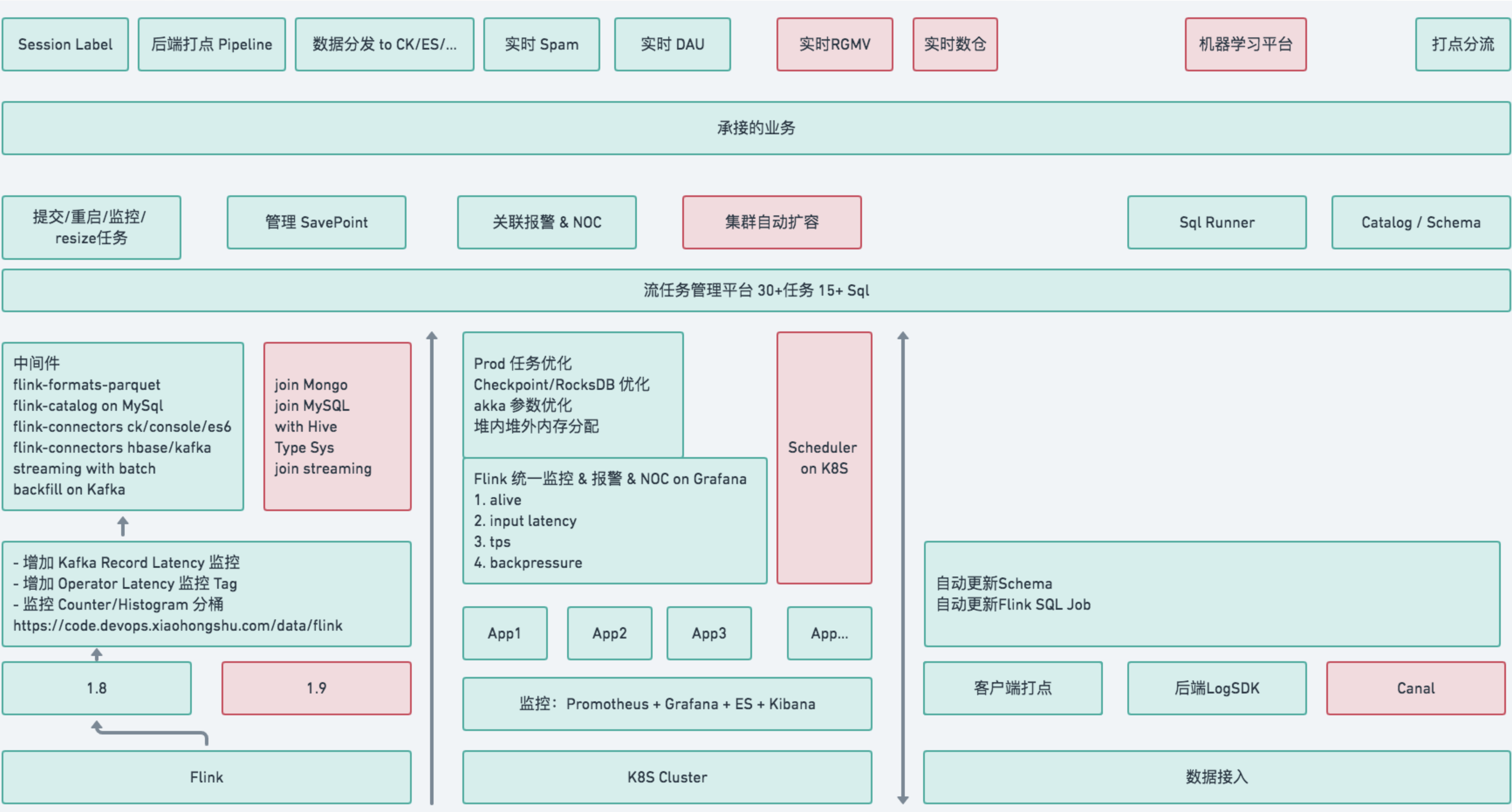
- 本地 HDD => SSD
- 堆外内存 = MemTable \* #Parallelism
- Checkpoint频率 1min = > 10 min
- RocksDB Compaction Threshold => 1G

# Backfill

- Source流控
- 支持Kafka Source和文件
- 期待Flink 1.9 对批的支持加强

# Red Flink实时流计算平台







Flink 集群管理 App 配置 后端日志业务 Flink 任务管理 Flink 数据字典 Schema 变更 审计 用户管理

列表 创建 详情

筛选器

业务	client_data_crashed_ios
Destination	aws-hive, rex-clickhouse
负责人	
下游表	algo-emr: <code>hdfs:///agent-smith/client_data_crashed_ios</code> --> ['aws-hive: <code>redo</code> tencent-press-kafka: <code>client_data_crashed_ios</code> --> ['rex-clickhouse: <code>client_data_</code>
状态	PROD
关联的 Flink 任务	sql-00045
监控	<div>detail</div> <div></div>
Schema	<div>timestamp TIMESTAMP crashed_app_duration INT os_version VARCHAR session_id VARCHAR crashed_event_id VARCHAR start_time BIGINT release_stage VARCHAR app_build VARCHAR crashed_module VARCHAR is_rooted VARCHAR crashed_time BIGINT create_time_format BIGINT app_version VARCHAR start_time_format BIGINT crashed_time_format BIGINT device_id VARCHAR crashed_user VARCHAR is_simulator VARCHAR crashed_experiment VARCHAR crashed_type VARCHAR os_build VARCHAR model VARCHAR</div>

配置目的地

集群名称	任务状态/监控	是否 HA	Status	报警级别
sql-00045 Total CPU Core 1 Total Mem 1GB Num TM 1 Total Slots 1  K8S JobManager Pod K8S TaskManagers Pods 关联的 Sql 任务	<div>RUNNING Flink Web Flink Metrics</div> <div>Source Record Latency by Partition P99 current 1.243 s</div>		run_job_success	p0

自动产生Flink SQL ETL命令

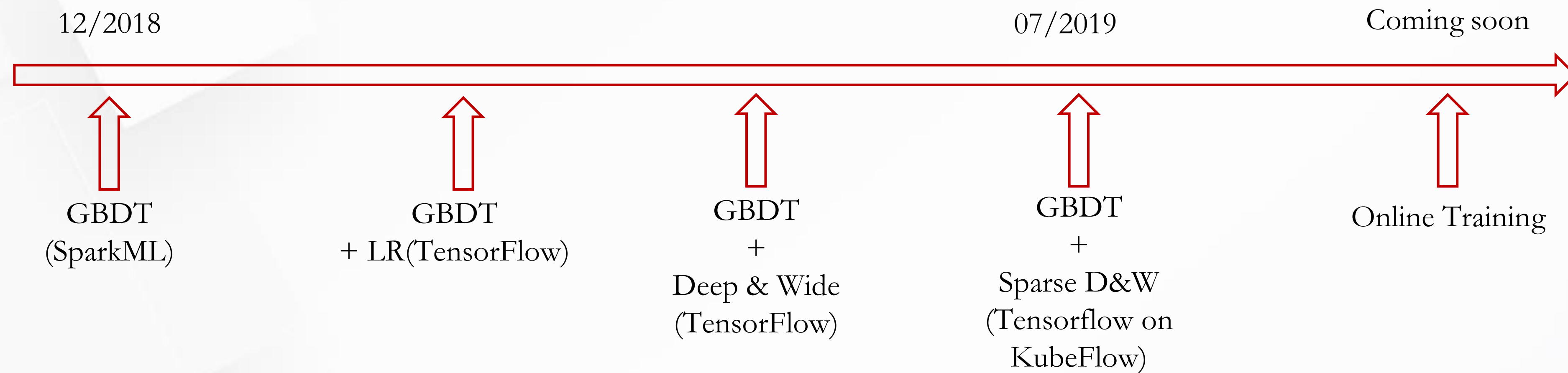
任务	DML	并行	Jar
sql-ck_client_data_crashed_ios	<pre>INSERT INTO `db.sink_ck_client_data_crashed_ios` SELECT `timestamp`, `crashed_app_duration`, `app_version`, `crashed_experiment`, `crashed_type`, `os_build`, `model`, `crashed_app_foreground`, `create_time`, `crashed_message`, `os_version`, `session_id`, `crashed_event_id`, `start_time`, `release_stage`, `app_build`, `crashed_module`, `is_rooted`, `crashed_time`, `create_time_format`, `start_time_format`, `crashed_time_format`, `device_id`, `crashed_user`, `is_simulator` FROM db.client_data_crashed_ios</pre>	1	sql-runner

自动识别格式

# Red ML平台

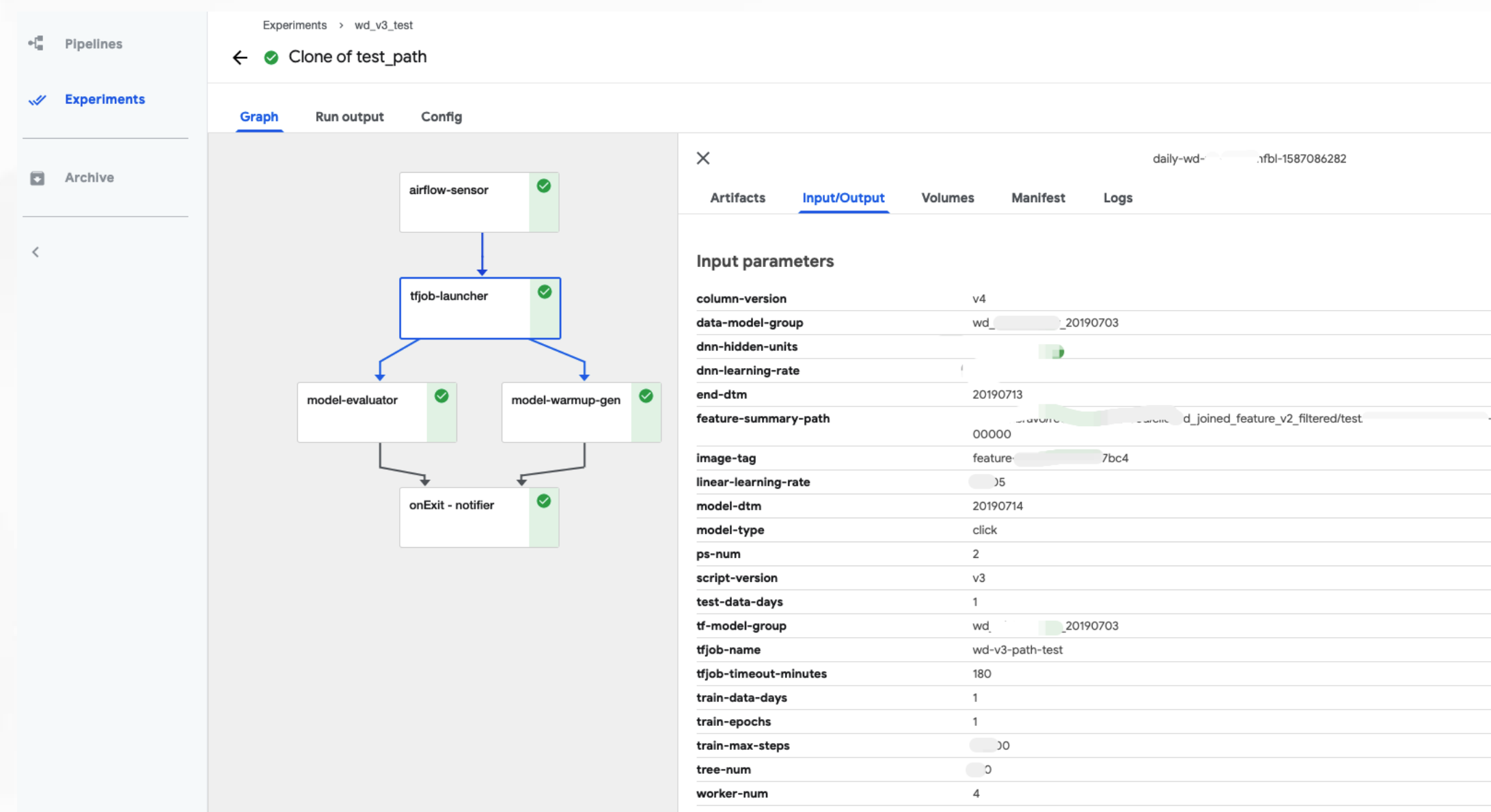
## 小红书推荐预测模型的演进

- 9个预测任务 (click, hide, like, fav, comment, share, follow, ...)
- Click模型规模：5亿样本/天，1T数据/天



# Red ML on KubeFlow

- 开源社区支持
- TFJob支持TensorFlow的分布式训练



## 总结与展望

- 依赖开源社区
- 期待Flink 1.9新功能
- 模型训练算力需求增加, 效率敏感
- Flink + AI





阿里云开发者社区

扫码加入社群  
与志同道合的码友一起  
Code Up

大数据计算开发者...



该群属于“阿里云ACE”部门群，仅组织内部成员可以加入，如果组织外部人员收到此分享，需要先申请加入该组织。

谢谢！