

## 基于 Flink 的实时特征平台在携程的应用

本文来自7月26日在上海举行的 Flink Meetup 会议，分享来自于刘康，目前在大数据平台部从事模型生命周期相关平台开发，现在主要负责基于flink开发实时模型特征计算平台。熟悉分布式计算，在模型部署及运维方面有丰富实战经验和深入的理解，对模型的算法及训练有一定的了解。

本文主要内容如下：

- 在公司实时特征开发的现状基础上，说明实时特征平台的开发背景、目标以及现状
- 选择Flink作为平台计算引擎的原因
- Flink的实践：有代表性的使用示例、为兼容Aerospike（平台的存储介质）的开发以及碰到的坑
- 当前效果&未来规划

### 实时特征平台的开发背景、目标以及现状

#### 1、原实时特征作业的开发运维

1.1、选择实时计算平台：依据项目的性能指标要求（latency，throughput等），在已有的实时计算平台:Storm Spark flink进行选择

1.2、主要的开发运维过程：

- 80%以上的作业需要用到消息队列数据源，但是消息队列为非结构化数据且没有统一的数据字典。所以需要通过消费对应的topic，解析消息并确定所需的内容
- 基于需求中的场景，设计开发计算逻辑
- 在实时数据不能完全满足数据需求的情况，另外开发单独的离线作业以及融合逻辑；例如：在需要30天数据的场景下，但消息队列中只有七天内的数据时（kafka中消息的默认保留时间），剩下23天就需要用离线数据来补充。
- 设计开发数据的校验和纠错逻辑
- 消息的传输需要依赖网络，消息丢失和超时难以完全避免，所以需要有一个校验和纠错的逻辑。
- 测试上线
- 监控和预警

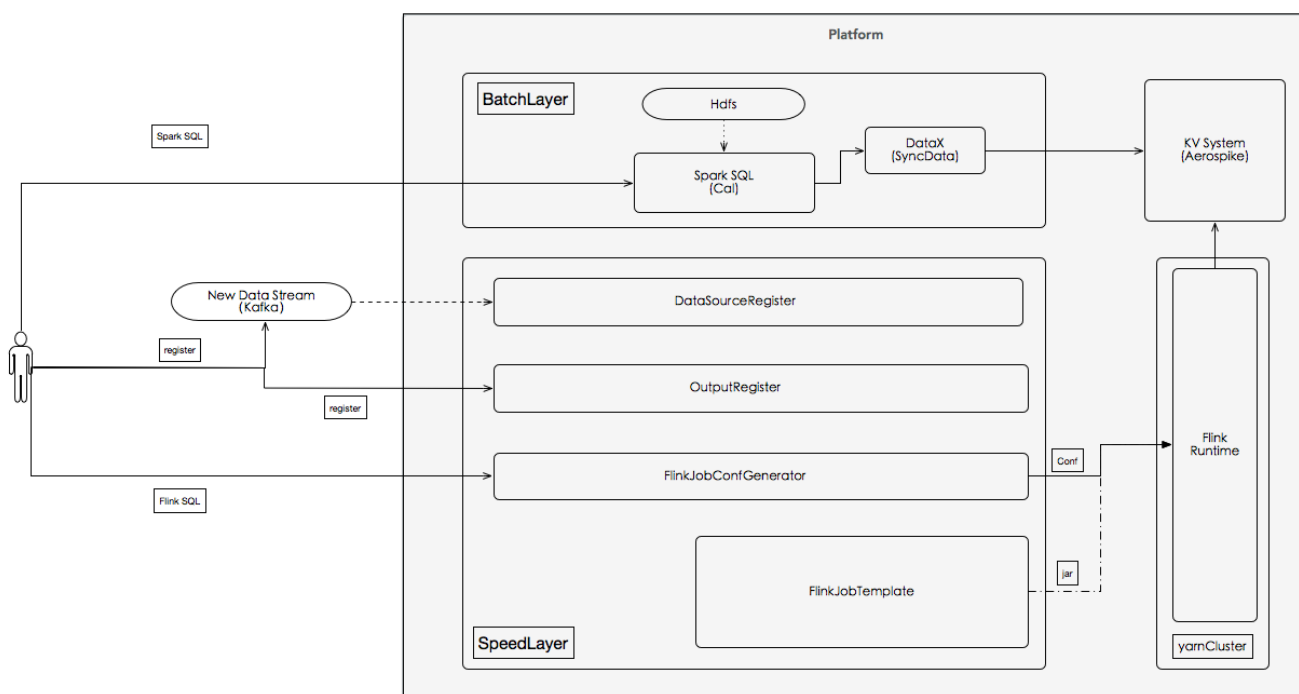
#### 2、原实时特征作业的开发痛点

- 消息队列数据源结构没有统一的数据字典
- 特征计算逻辑高度定制化，开发测试周期长
- 实时数据不能满足需求时，需要定制离线作业和融合逻辑
- 校验和纠错方案没有形成最佳实践，实际效果比较依赖个人能力
- 监控和预警方案需要基于业务逻辑定制

### 3、基于整理的痛点，确定下来的平台目标

- 实时数据字典：提供统一的数据源注册、管理功能，支持单一结构消息的topic和包含多种不同结构消息的topic
- 逻辑抽象：抽象为SQL，减少工作量&降低使用门槛
- 特征融合：提供融合特征的功能，解决实时特征不能完全满足数据需求的情况
- 数据校验和纠错：提供利用离线数据校验和纠错实时特征的功能
- 实时计算延迟：ms级
- 实时计算容错：端到端 exactly-once
- 统一的监控预警和HA方案

### 4、特征平台系统架构



如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog\_hadoop

现在的架构是标准lamda架构，离线部分由spark sql + dataX组成。现在使用的是KV存储系统Aerospike，跟redis的主要区别是使用SSD作为主存，我们压测下来大部分场景读写性能跟redis在同一个数据量级。

实时部分：使用flink作为计算引擎，介绍一下用户的使用方式：

- 注册数据源：目前支持的实时数据源主要是Kafka和Aerospike，其中Aerospike中的数据如果是在平台上配置的离线或者实时特征，会进行自动注册。Kafka数据源需要上传对应

的schemaSample文件

- 计算逻辑：通过SQL表达
- 定义输出：定义输出的Aerospike表和可能需要的Kafka Topic,用于推送Update或者Insert的数据的key

用户完成上面的操作后，平台将所有信息写入到json配置文件。下一步平台将配置文件和之前准备好的flinkTemplate.jar(包含所有平台所需的flink功能)提交给yarn，启动flink job。

## 5、平台功能展示

### 平台功能展示-数据源注册

注册数据源信息

数据源名称\*

数据源名称

数据源名称必填

Topic名称\*

Topic名称

Topic名称必填

Topic类型\*

☐ HermesKafka
 ☒ Kafka

Schema类型\*

☐ multiply
 ☒ single

Topic编码\*

☐ Avro
 ☒ Json

Topic的Table别名(用于SQL)\*

Topic的Table别名(用于SQL)

Table别名必填

上传JsonSchema文件

选择文件

vac\_test.json

ts:long

custom\_key:string

pkgid:int

slcityid:int

dtcityname:string

bu:string

slcityname:string

dtcityid:int

列投影设置

显示 10 行

搜索:

Field名称	Tab列名	类型	自定义类型类型	默认EventTime属性
bu	bu	STRING		<input type="radio"/>
custom_key	custom_key	STRING		<input type="radio"/>
dtcityid	dtcityid	INT		<input type="radio"/>
dtcityname	dtcityname	STRING		<input type="radio"/>
pkgid	pkgid	INT		<input type="radio"/>
slcityid	slcityid	INT		<input type="radio"/>
slcityname	slcityname	STRING		<input type="radio"/>
ts	ts	LONG		<input type="radio"/>

显示 1 to 8 of 8 行

上一页

1

下一页

数据源类型\*

☒ 注册
 ☐ 转换

创建

取消

如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog\_hadoop

### 实时特征编辑-基本信息

## 特征编辑与测试

首页 / 特征管理 / 特征编辑与测试

离线特征

在线特征

1 基本信息

2 数据源信息

3 计算信息

4 存储信息

Tip: 首次创建特征请先在配置管理中配置数据源、分类与特征输出等配置

实时特征名称\*

实时特征名称

实时特征名称必填

实时特征别名\*

实时特征别名

首次使用AS作为数据源时别名将被作为实时名称使用,需向一线AS表申请

是否公开\*

公开

版本信息\*

1

所属分类

已创建

取消

如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog\_hadoop

## 实时特征编辑-数据源选择

## 特征编辑与测试

首页 / 特征管理 / 特征编辑与测试

离线特征

在线特征

1 基本信息

2 数据源信息

3 计算信息

4 存储信息

KAFKA 数据源位置信息

ds\_0

+

选择数据源\*

test

AeroSpoke 数据源 (可选)

已经存储到As的表

default\_online\_features.ops\_user\_info

已创建

取消

如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog\_hadoop

## 实时特征编辑-SQL计算

特征编辑与测试

首页 / 特征管理 / 特征编辑与测试

离线特征

在线特征

1 基本信息

2 数据源信息

3 计算信息

4 存储信息

计算引擎:  
FLINK

实时计算描述:  
计算描述

实时计算sql:  

```
1 select ID, cityId,cityName from test
2 LEFT JOIN LATERAL TABLE(asGetBins(PRODUCTID)) as T(cityId,cityName) ON TRUE
```

#	列名	类型	映射主键
0	ID	Key Value	userid
1	cityId	Key Value	city
2	cityName	Key Value	

是否进行Checkpoint:  
☒  
checkpoint触发间隔(ms):  
100  
checkpoint间最小间隔(ms):  
50

如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog\_hadoop

## 实时特征编辑-选择输出

## 特征编辑与测试

首页 / 特征管理 / 特征编辑与测试

离线特征

在线特征

1 基本信息

2 数据源信息

3 计算信息

4 存储信息

输出到AeroSpike(必填)

输出到Kafka(可选)

获取注册的存储地址

存储到As的表\*  
default\_online\_features.ops\_forTest

存储到As的TTL(time to live: 单位-秒) ⓘ \*  
-1

创建

取消

如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog\_hadoop

## 选择Flink的原因

我们下面一个我们说一下我们选择flink来做这个特征平台的原因。



Flink China

	storm V1.2.2	spark Structured Streaming v2.3.1	flink v1.5
延迟	Streaming ms级	MicroBatch 100ms级 Streaming ms级 (实验)	Streaming ms级
容错	Ack atLeastOnce	CheckPoint&WAL exactlyOnce	CheckPoint&SavePoint exactlyOnce
SQL成熟度 (Unsupported Functions)	aggregation、join	distinct、limit、order by(partial)	distinct aggregate

如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog\_hadoop

分为三个维度：最高延迟、容错、sql功能成熟度

- 延迟：storm和flink是纯流式，最低可以达到毫秒级的延迟。spark的纯流式机制是continuous模式，也可以达最低毫秒级的延迟，
- 容错：storm使用异或ack的模式，支持atLeastOnce。消息重复解决不。spark通过check point和WAL来提供exactlyOnce。flink通过checkpoint和SavePoint来做到exactlyOnce。
- sql成熟度：storm现在的版本中SQL还在一个实验阶段,不支持聚合和join。spark现在可以提供绝大部分功能，不支持distinct、limit和聚合结果的order by。flink现在社区版中提供的sql，不支持distinct aggregate

## Flink实践

### 实践示例

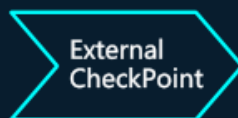


## 使用用例



会话窗口用途广泛，可以用于推荐召回、用户召回等；原设计方案需要使用分布式锁做进程间的并发控制，复杂度高

```
select user,count(ts) from vac_ts group by SESSION(userEventTime_ts,INTERVAL '10' SECOND),user
```



实时计算job执行异常设置的重启策略无法恢复需要修复bug时，如果设置了externalCheckPoint，很大几率保证exactly Once的语义

如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog\_hadoop

### 兼容开发

flink现在没有对Aerospike提供读写支持，所以需要二次开发





Flink China

## 兼容开发

AS Async  
Upsert  
TableSink



基于flink Asynchronous I/O开发（支持exactlyOnce语义）  
异步写入AS  
写入AS成功后可选择push key至Kafka  
支持Upsert Stream的TableSink

UDF-AS  
Table  
Functions



通过定义Table Function类型的UDF支持，SQL中用到AS数据的情况  
示例：  
`SELECT ID, cityId, cityName FROM product LEFT JOIN LATERAL  
TABLE(asGetBins(productId)) as T(cityId,cityName) ON TRUE`

AS=Aerospike

如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog\_hadoop

碰到的坑



Flink China

## 碰到的坑

SQL\_TIMESTAMP



内部默认反序列化JSON中SQL\_TIMESTAMP类型时，值必须是格式为yyyy-MM-dd' T'HH:mm:ss.SSS'Z'的字符串  
场景：直接使用StreamTableEnvironment的方法registerTableSource注册表时，表中的时间属性（eventTime）字段的类型需要配置为SQL\_TIMESTAMP

SlidingWin  
Offset



窗口的offset需要小于winSize，不能用于时区适配  
下面是当前计算win的start值的公式  
$$\text{timestamp} - (\text{timestamp} - \text{offset} + \text{windowSize}) \% \text{windowSize}$$

如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog\_hadoop

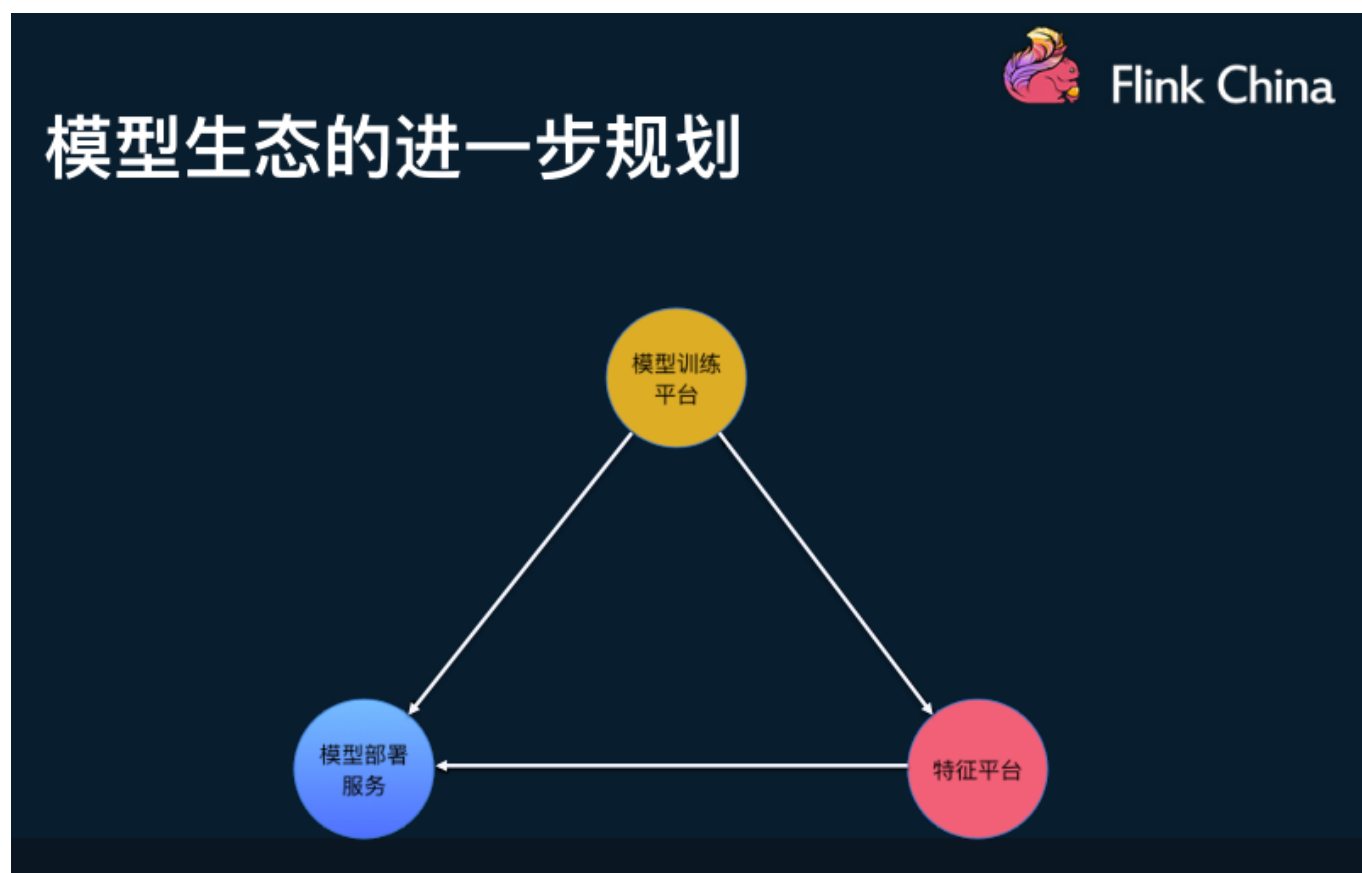
## 平台当前效果&未来规划

当前效果：将实时特征上线周期从原平均3天-5天降至小时级。

未来规划：

- 完善特征平台的功能：融合特征等
- 简化步骤，提高用户体验
- 根据需求，进一步完善SQL的功能例如支持win的开始时间offset，可以通过countTrigger的win等

下一步的规划是通过sql或者DSL来描述模型部署和模型训练



如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog\_hadoop

## 本文 PPT 下载

本文的 PPT 可以到 [《Flink China社区线下 Meetup:上海站 PPT 资料分享》](#) 里面进行下载。

本博客文章除特别声明，全部都是原创！

转载本文请加上：转载自过往记忆 (<https://www.iteblog.com/>)

本文链接: [【】 \( \)](#)