携程机票数仓建设之路

演讲人 张振华
2019年11月9日

张振华

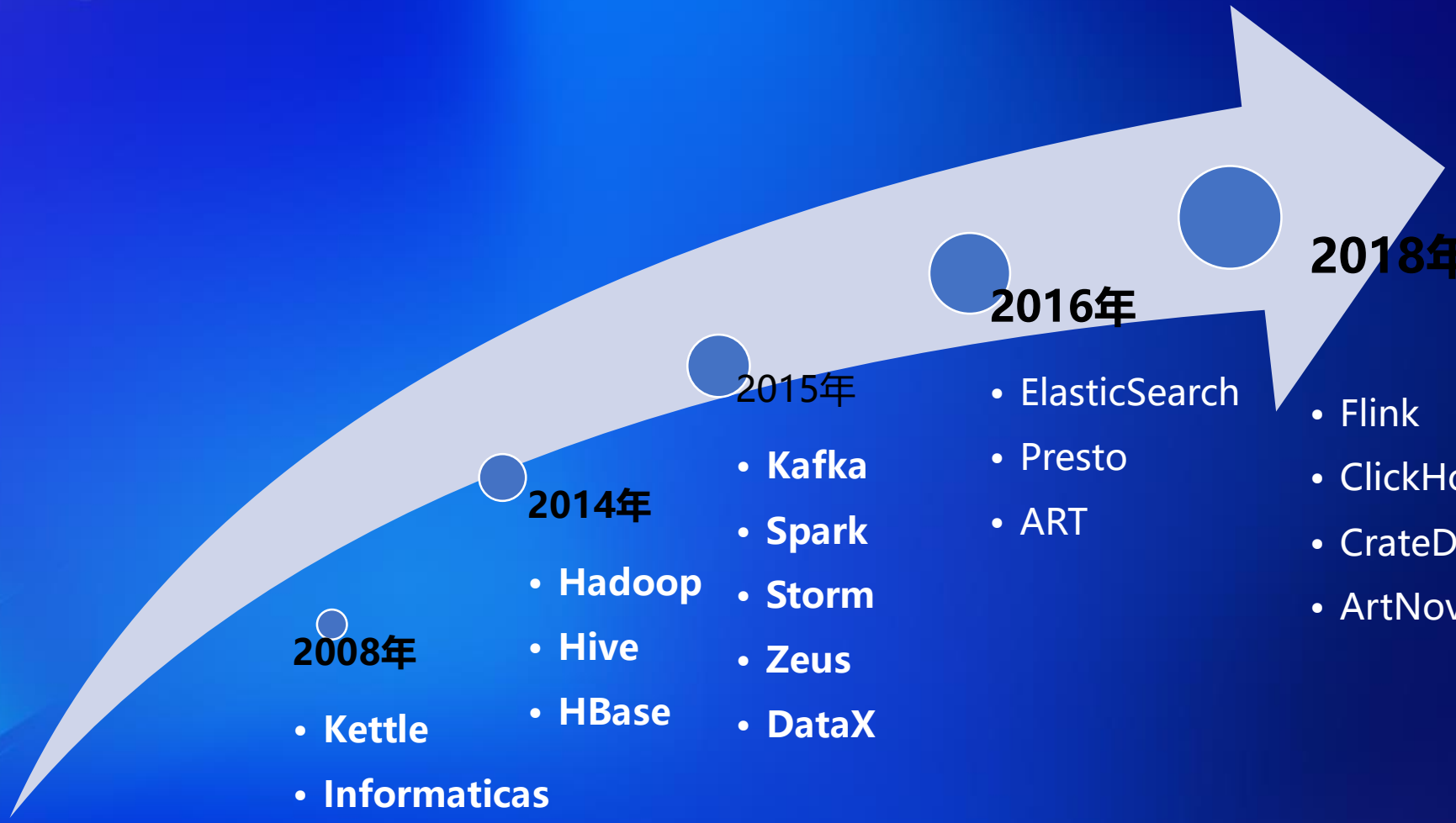
本科、博士分别就读于中国科技大学、中国科学院大学计算机专业

2015年加入携程机票研发部，负责大数据技术调研及在机票的落地。

2018年起加入机票数据仓库团队，主要负责数据仓库技术架构、性能优化、数仓规范制定、数据模型设计以及数据应用开发

目录

数仓技术演进历史


2008年
Kettle
Informaticas
SQLSever
BO
2014年
Hadoop
Hive
HBase
2015年
Kafka
Spark
Storm
Zeus
DataX
2016年
ElasticSearch
Presto
ART
2018年
Flink
ClickHouse
CrateDB
ArtNova

当前技术栈


生产
MySQL
SQLServer
Kafka
QMQ
Spark
Flink
DataX
DataX
Hamal/Gobblin
线上服务
Spark Job
HIVE SQL/SPARK SQL
HIVE
HBASE
elasticsearch.
CRATE.IO
ES Hadoop
Spark Job
Python/R
pyspark
作业调度系统
模型
模型服务
用户行为分析
运营可视化工具
用户画像
生产应用
价格监控
Presto-search
Kibana
日志追踪

数仓的团队合作

产品

- 用户行为分析
- 运营可视化系统
- 埋点管理系统
- 大数据查询工具
- 取数、报表、数据分析

业务、财务

- 报表
- 运营可视化系统
- Adhoc取数
- 业财一体化系统

开发、测试

- 用户画像服务
- 大数据查询前端、服务
- 日志追踪系统
- 报文解析系统
- 流量回放系统

数据分析师

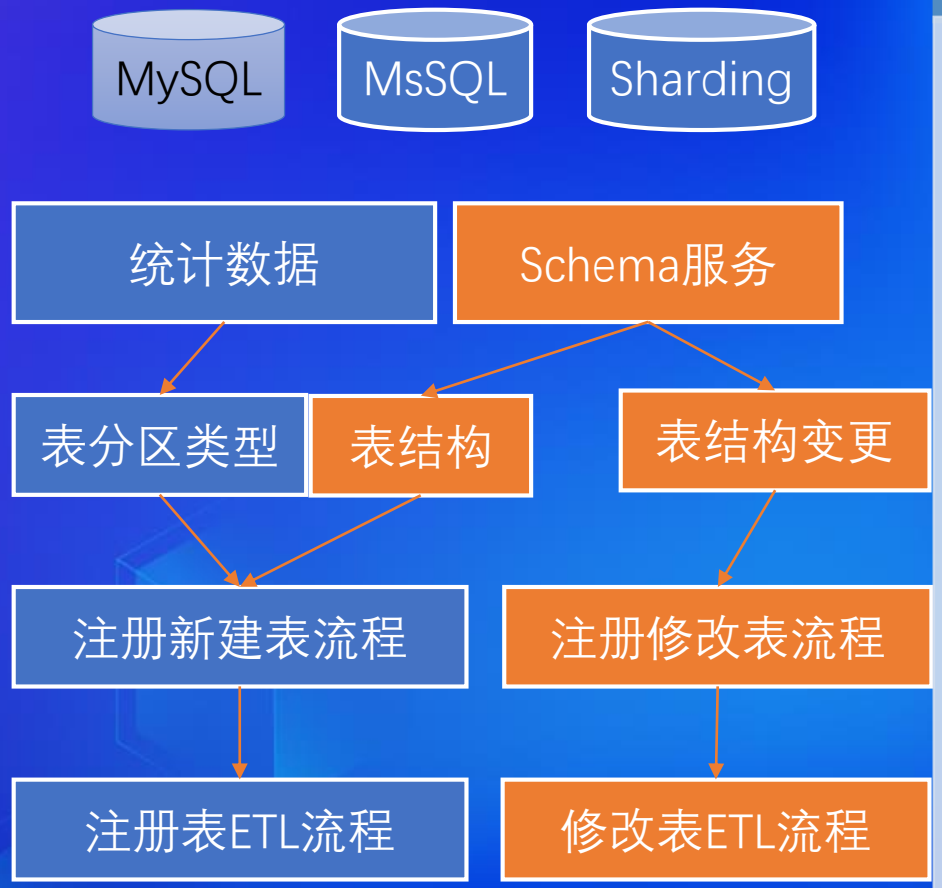
- 底表
- 自定义函数集
- 数据处理
- 分布式算法实现

营销

- 定投营销系统
- 效果分析

数仓建设内容

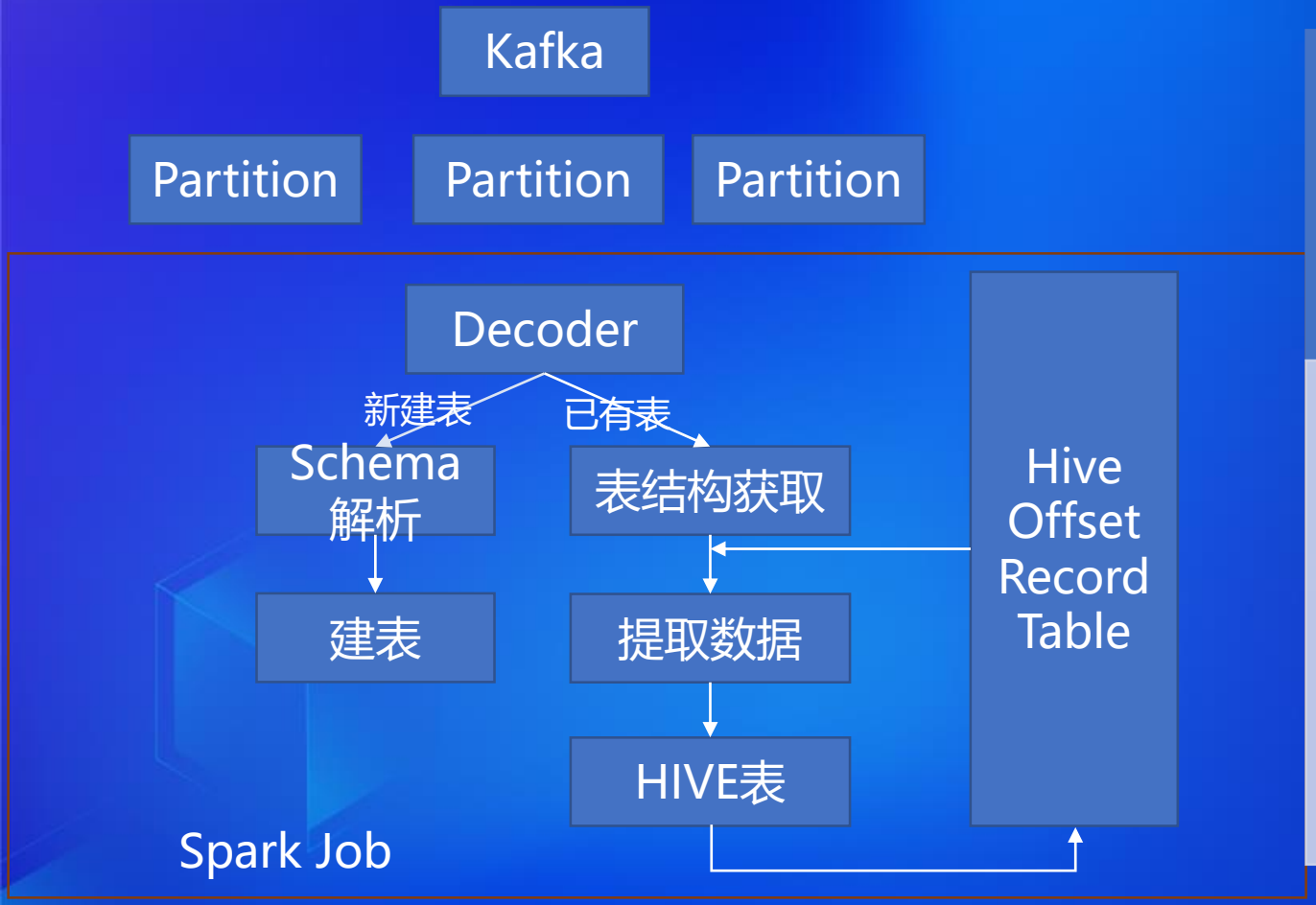
数据同步



DB->HIVE

- **覆盖所有表**
- **DataX**
- 获取DB SCHEMA，映射成HIVE SCHEMA,自动注册ETL zeus脚本,定期同步生产schema的变更
- 初始化脚本
  - 全量同步
- 分区类型
  - 全量分区
    - UPDATE DML的表
  - 增量分区
    - INSERT ONLY的表
- 无分区
  - 基础数据表
- Hint: 尽量不要物理删除，采用逻辑删除、批量物理删除

数据同步



基于spark-sql-kafka的二次开发

- 配置简单
- 稳定
- 方便异常监控

```
cat <<EOF > hamal.properties

cluster.name=
topic.name=
data.db.name=
log.db.name=
log.table.name=
create.table=false

EOF
/opt/app
--class
hamal-
master yarn --deploy-mode client  \
hamal.properties"
```

示例

数仓分层设计


dm
流量
产量
收益
生产KPI
业务考核
主题聚合
dws
订单宽表
出退改票宽表
收益宽表
搜索结果宽表
前端行为宽表
Dws..
维度整合
dwd
订单、航段、乘客处理表
辅营产品处理表
出退改处理表
日志解析表
Dwd..
数据清洗、字段筛选
ods
订单、航段、乘客
辅营产品
出票、退票、改签
机型、城市、机场基础数据
各类日志表
ods_...
维表
基础数据维表

宽表设计


便捷
Trade off
冗余


宽表
不惧冗余
便利使用
不变粒度
兼顾性能
丰富维度


订单宽表
主订单号
用户行为还原
用户原始行程
拆单类型
销售渠道
出票日期
Clientcode
Sid
Vid
前端信息
设备标识
经纬度
串联UUID
三字码翻译
行程信息
中转时长
行程概况
往返停留时长

数据分层-解析处理框架

1. 对非大数据开发同学虚化大数据处理框架
2. 各司其职，处理需求迭代周期短

聚合层数据解析

1. 聚合层返回给前端的用户查询结
2. 数据格式zstd/gzip+ProtoBuf
3. 解析需求：
   将航组、产品类型、乘客类型、票台、价格数据展平
4. 一些数值
   a.17年初上线，积累800TB
   b.日增80亿

元数据管理

基础信息

表结构描述

字段注释

存储

- 位置
- 格式

分区表

- 分区类型 增量/全量
- 分区统计

生命周期ttl

创建时间

表修改时间

使用热度

血缘关系

表与表

表与报表

字段

表与作业

标签

层级标签

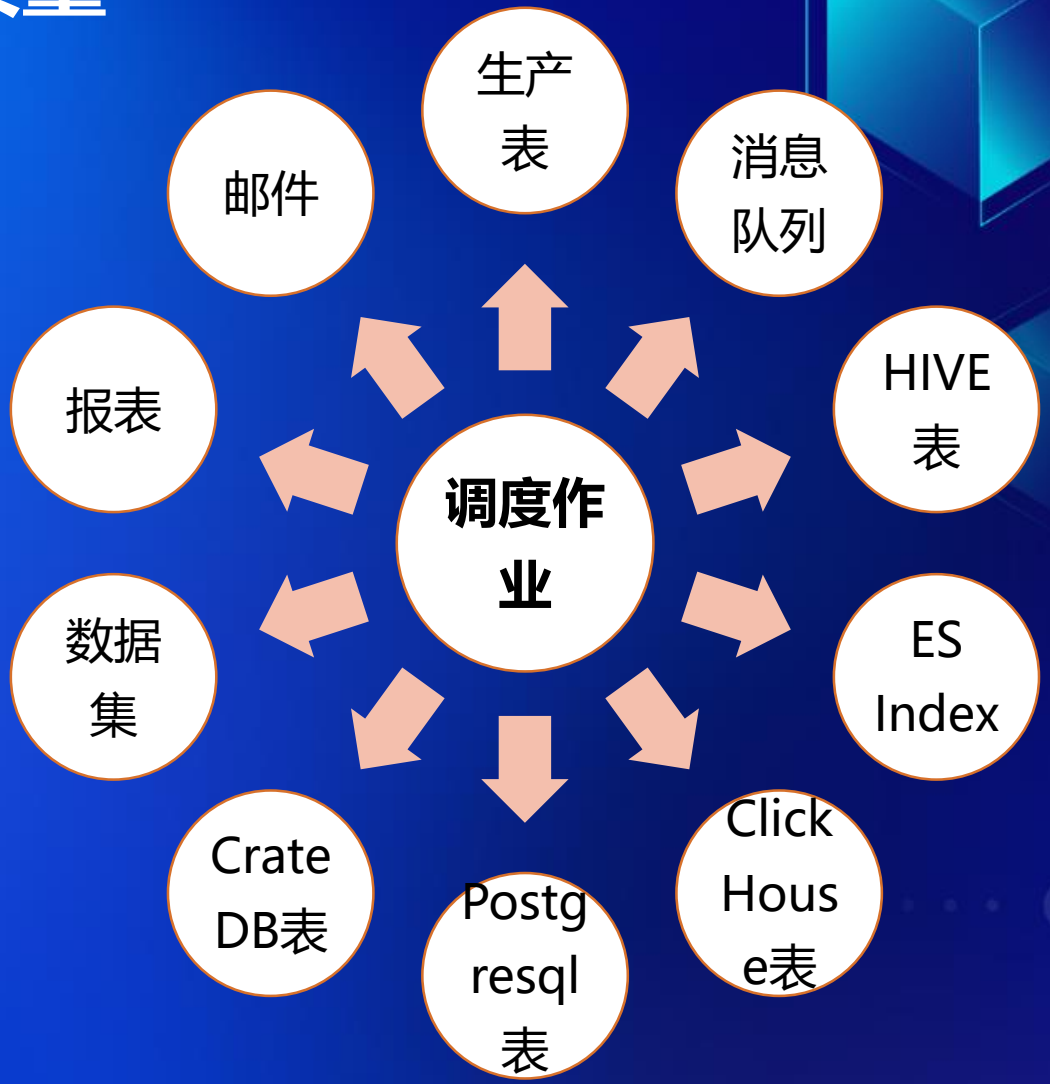
安全标签

重要等级标签

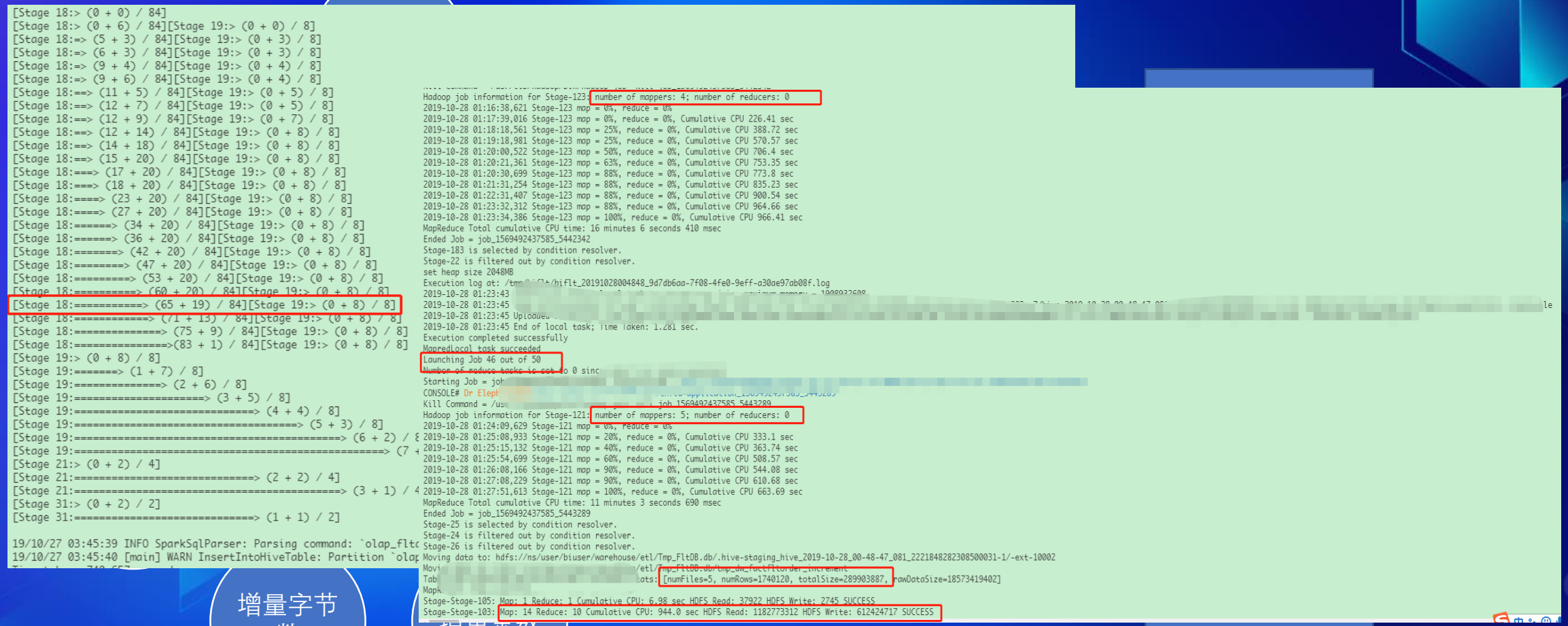
业务标签

数据质量

报表为啥又不对？



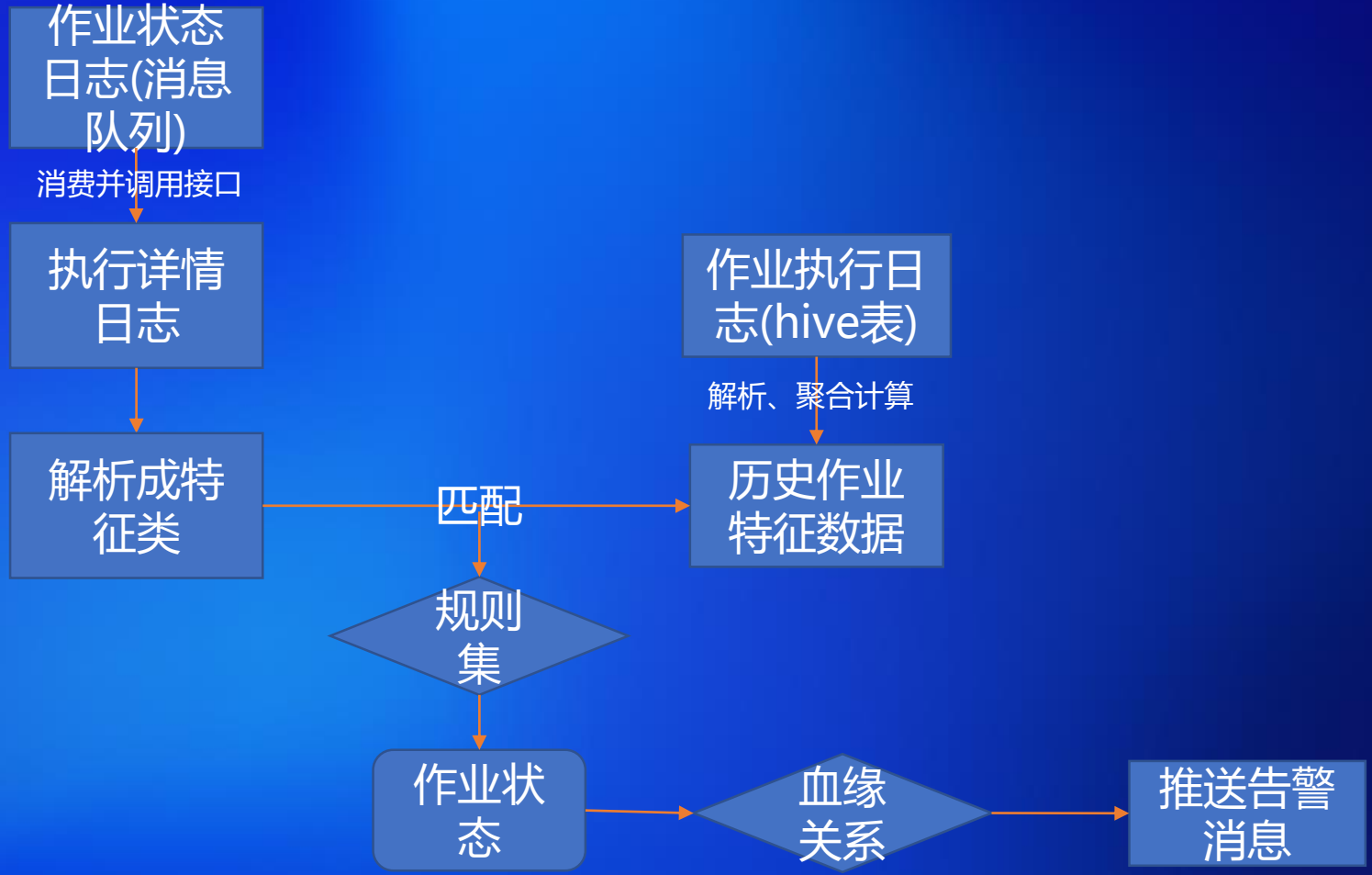
数据质量



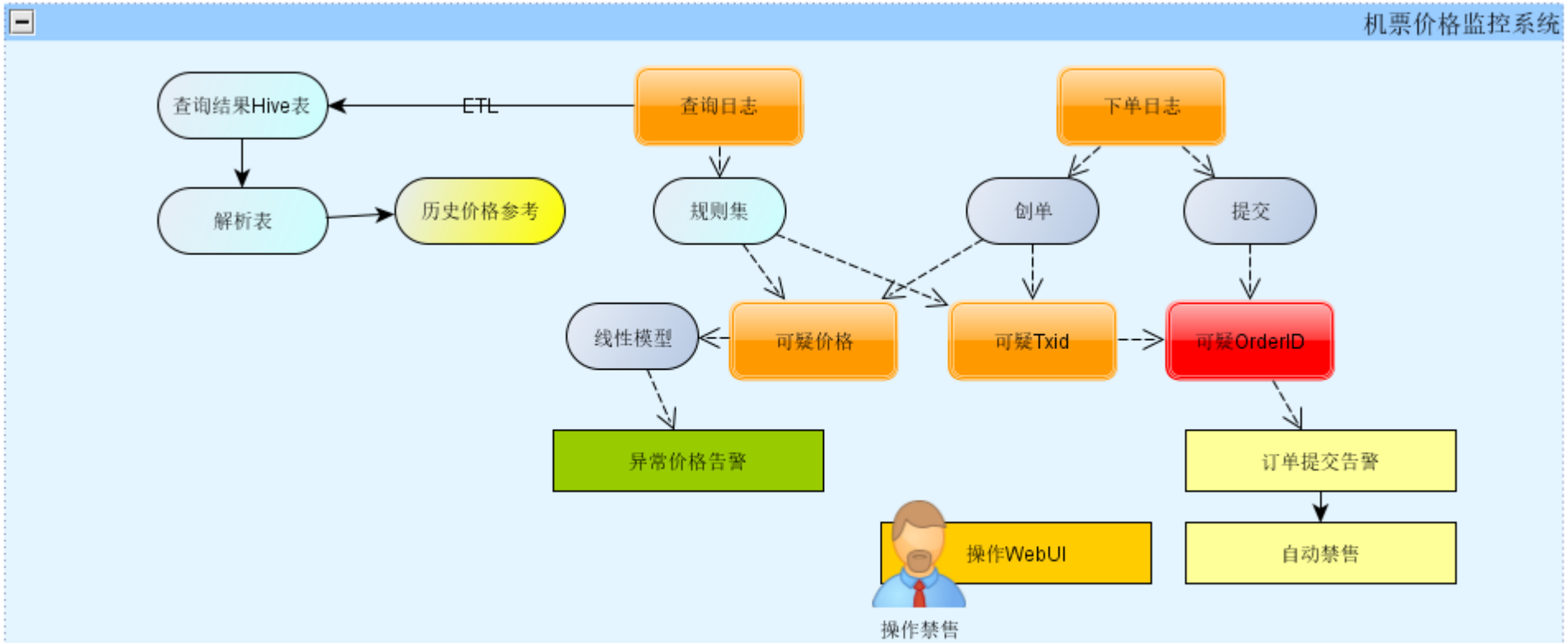
增量字节数

增量条数

数据质量检验流程


作业状态日志(消息队列)
消费并调用接口
执行详情日志
解析成特征类
匹配
作业执行日志(hive表)
解析、聚合计算
历史作业特征数据
规则集
作业状态
血缘关系
推送告警消息

数仓应用-价格监控



- 毫秒级响应
- 发现数十起价格异常

数仓应用-日志追踪

Service-A　Service-B　Service-C

Topic-A　Topic-B　Topic-C

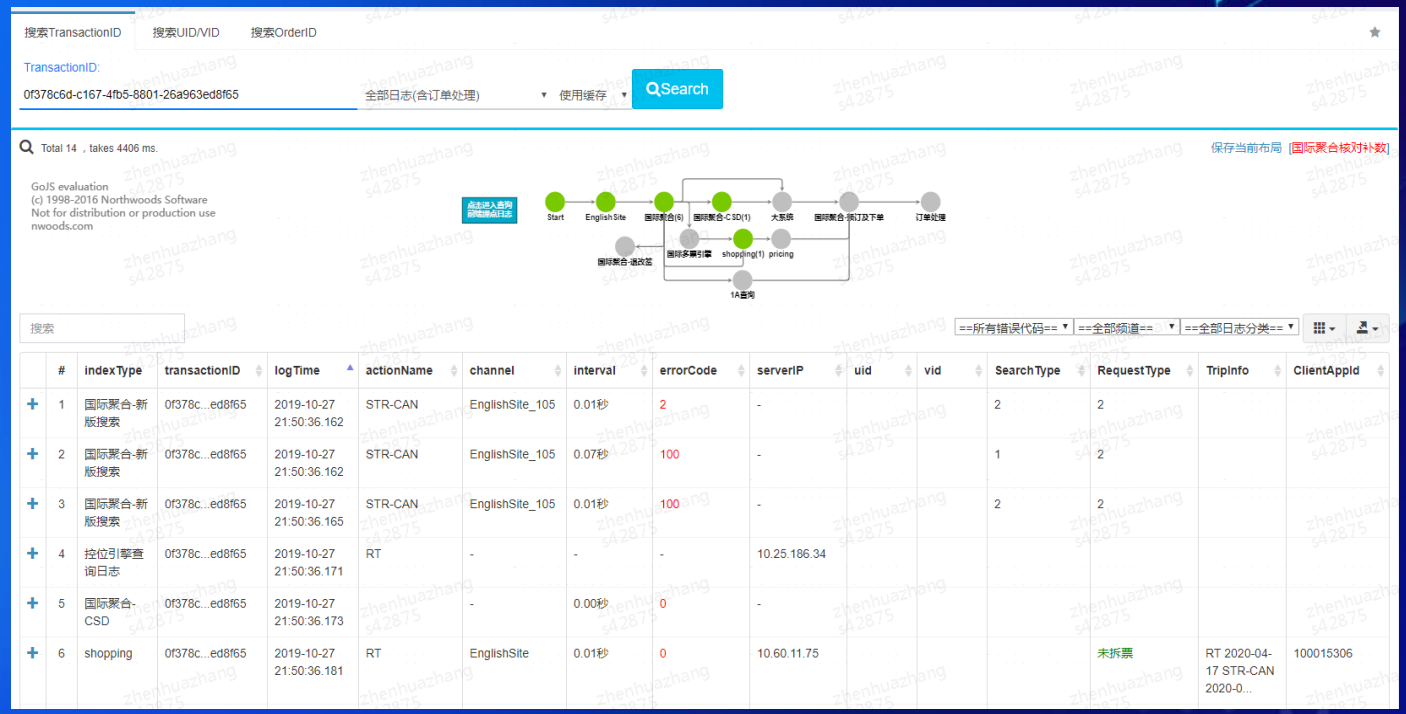
Flink

Txid:[Date List]

同步，topic+date=>index

Redis

ElasticSearch



	#	indexType	transactionID	logTime	actionName	channel	interval	errorCode	serverIP	uid	vid	SearchType	RequestType	TripInfo	ClientAppId
+	1	国际聚合-新版搜索	0f378c...ed8f65	2019-10-27 21:50:36.162	STR-CAN	EnglishSite_105	0.01秒	2	-			2	2		
+	2	国际聚合-新版搜索	0f378c...ed8f65	2019-10-27 21:50:36.162	STR-CAN	EnglishSite_105	0.07秒	100	-			1	2		
+	3	国际聚合-新版搜索	0f378c...ed8f65	2019-10-27 21:50:36.165	STR-CAN	EnglishSite_105	0.01秒	100	-			2	2		
+	4	控位引擎查询日志	0f378c...ed8f65	2019-10-27 21:50:36.171	RT	-	-	-	10.25.186.34						
+	5	国际聚合-CSD	0f378c...ed8f65	2019-10-27 21:50:36.173		-	0.00秒	0	-						
+	6	shopping	0f378c...ed8f65	2019-10-27 21:50:36.181	RT	EnglishSite	0.01秒	0	10.60.11.75				未拆票	RT 2020-04-17 STR-CAN 2020-0...	100015306

数仓应用-报表取数

统一数据源

- 自身准确性
  - 业务规则校验
  - 历史数据规律
  - 其他数据源佐证

口径

- 建立不同口径取数模板
- 标准化口径管理

临时取数

- 中心化需求描述、脚本及数据
- 交叉review

报表管理

- 分类
- 僵尸报表识别
  - 埋点

数仓规范

命名规范

- 正式/临时 库/表
  - 正式表: {层级}[_类别]_{业务}_{描述}
  - 临时表: tmp_{编号}_{业务}
- 视图
  - v_{相关表名}
- 流程名称
  - 与写入表名一致

脚本规范

- 建表语句
  - comment
- 脚本描述
  - 功能
  - 变更历史
  - 是否支持无改动刷数
- 提行
- 注释

检测工具

规范文档、维护及培训

数仓面临的挑战

- 非数据生产者，但必须理解数据
- 生产数据逻辑变更没法事先主动获知
- 貌似做不完的数据需求
- 数据源/种类繁杂
- 历史遗留问题
- 流程过多，维护成本超高，僵尸报表难以辨别，资源浪费

可能带来的问题

1. 需求繁杂琐碎，资源碎片化
2. 数仓人员永远都在被动承接需求状态，无积极性和成就感
3. 人员晋升乏力，流失严重
4. 需求响应不及时，数据部门和需求方互相吐槽

数仓团队能做的

1. 让数仓工程师了解一类业务并熟悉其数据
2. 取数口径、脚本及数据定期分享
3. 数仓规范严格执行，对历史遗留问题做到不增加，有计划有节奏地去除
4. 自定义函数封装公用口径
5. 开发数仓实用工具，提升数据开发的效率
   - ① 脚本查询
   - ② 赋权流程
   - ③ 数据解析工具
   - ④ SQL助手
   - ⑤ 维表导入
6. 培训数据需求方SQL技能
7. 主动了解各需求方看数据视角，总结凝练，建设底表及可视化系统

本PPT来自2019携程技术峰会

更多信息请关注“携程技术中心”微信公众号~