

腾讯金融实时计算平台实践

张作峰 | 腾讯专家工程师、金融实时计算平台负责人
Apache InLong Committer, Apache Hudi/Flink Contributor



01 腾讯金融实时大数据

02 高效易用

03 稳定可靠

04 数据准确

05 未来展望

腾讯金融科技 - 科技连接金融, 用户为本、科技向善

让每一个人乐享金融的美好

财付通

微信支付

QQ钱包

移动支付

腾讯理财通

腾讯自选股

投资理财

腾讯区块链

腾讯手机充值

微信信用卡还款

微企付

民生服务

WeChat Pay HK

智汇鹅

国际化

实时计算驱动数据价值变现

实时计算——大数据"革命": 数据驱动、数据保鲜、价值变现

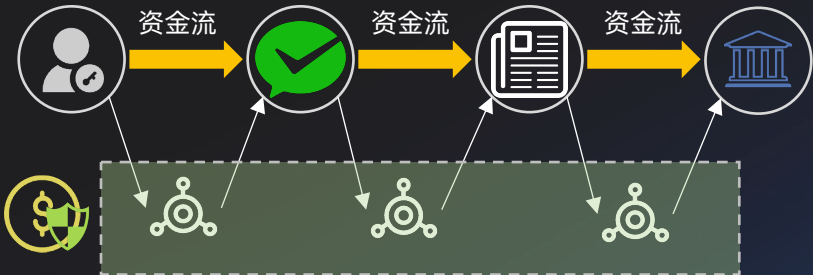
营销分析、指标提升



产品改进、体验升级



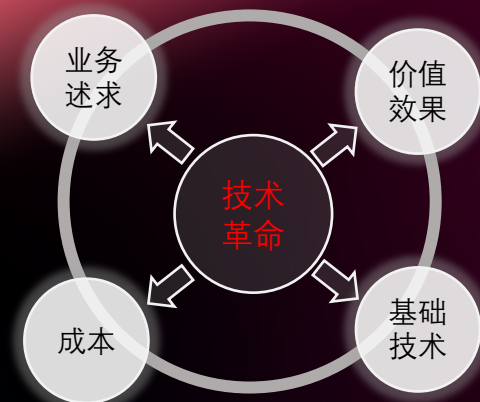
交易链路提效: 资金安全、风控



运营监控、商业决策



腾讯金融实时大数据发展



链路闭环

打通7个平台权限
打通5个开发工具
打通4个监控平台

提升易用性

降低开发门槛
提升开发效率
面向非BI人员

全链路稳定性

指标监控
运维工具
可观测性

数据准确性

实时对账
数据回补
数据审计

大数据的“最后一公里”

专业服务伙伴，聚力技术深耕

腾讯金融实时大数据平台架构



服务层:

对接各业务平台，提供任务开发、治理等服务

API层:

开放核心能力，支撑业务适配

内核层: 屏蔽计算引擎的复杂性

- 链路闭环
- 降低门槛
- 适配金融场景
- 全链路保障
- 模块化与扩展性: 无缝切换计算引擎

公共服务层:

公司和Oteam提供的公共服务能力
开源协同

业务对实时能力的挑战

某直播



半日内开发完成新报表
分钟级调整运营策略
4个9可用性
99.9%准确率
影响业务收入

某提现补贴



用户支付后, 5S内发放权益
4个9可用性
100%准确率
用户投诉、资金损失

基金经理人看板



根据交易量调整备付金
分钟级延迟
4个9可用性
99.9%准确率
资金损失

实时大数据赋能金融业务

高效
易用

- 接入成本高、耗时长
- 场景26+/需求复杂
- 沟通成本高, 反复对接
- 开发效率低, 上线周期长

稳定
可靠

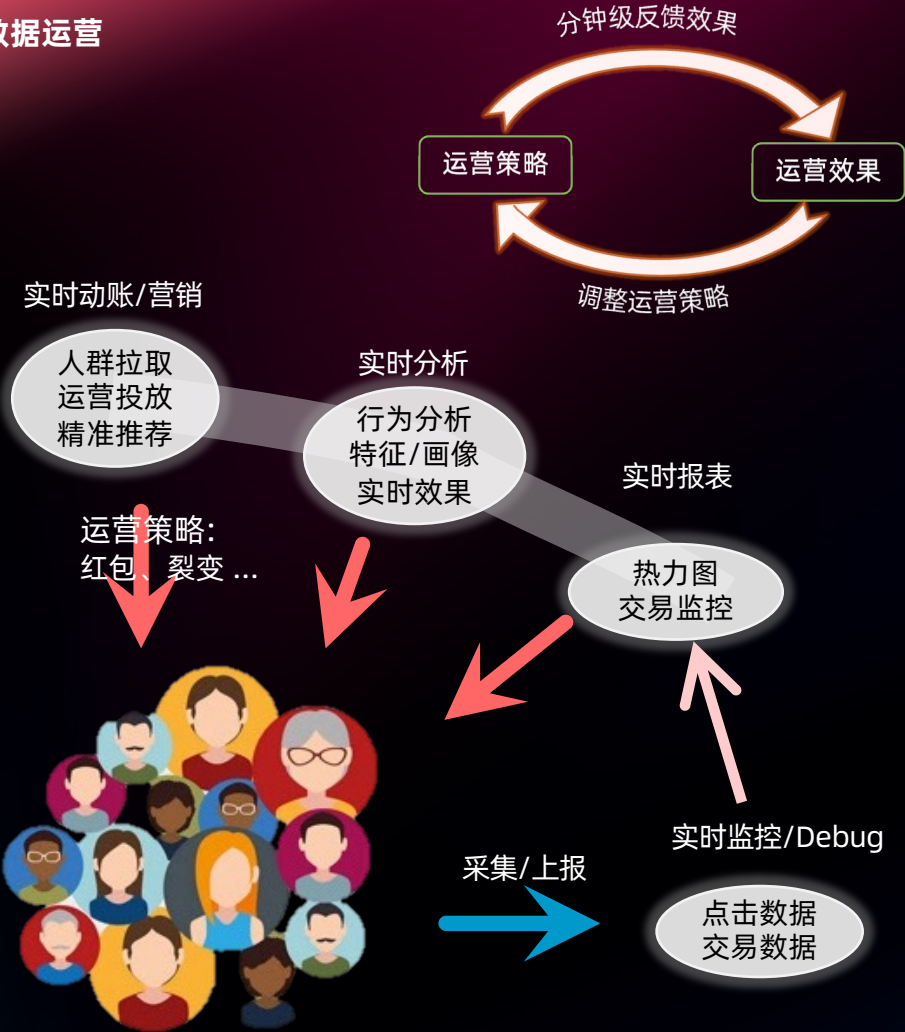
- 链路复杂、组件多、参数多
- 故障定位难、门槛高
- 全链路可靠性要求3~4个9
- 要求端到端秒级延迟

数据
准确

- 任务异常, 造成数据错误或丢失
- 数据缺失和错误, 导致资金损失

高效易用

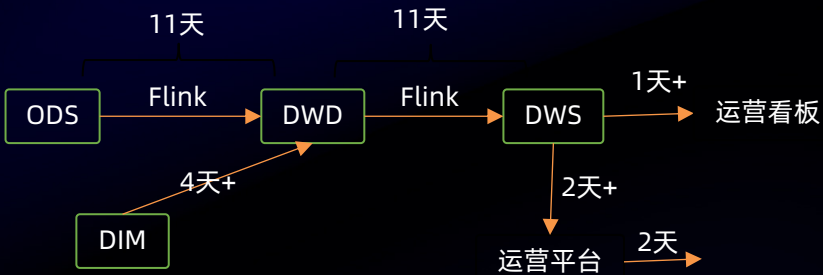
1. 数据运营



2023年，某业务直播预计 1000+ 场次

从方案到上线，至少需要11天，无法实现敏捷的活动运营

单任务开发耗时分析	
需求与方案	0.5天
代码开发	2.6天
任务调试	1.5天
数据验证	1.2天
查询接口与报表开发	1.8天
*活动配置(策略、权益)	2天
CR、压测、对账	3天



- ① 单任务开发(开发、测试、核对、压测、灰度、上线等)耗时 11天
- ② 业务侧无法承担开发，依赖数据侧排期，耗时进一步拉长

2. 挑战

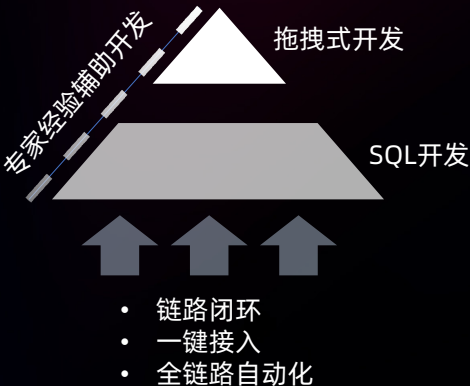
- 数据量大：日增数据几千亿、数十T
- 主播效果报表分钟级延迟、to C 秒级延迟
- 快上快下：开发周期短
- 数据维度和口径变化大

某直播场景-要求半日内开发完成并上线

高效易用

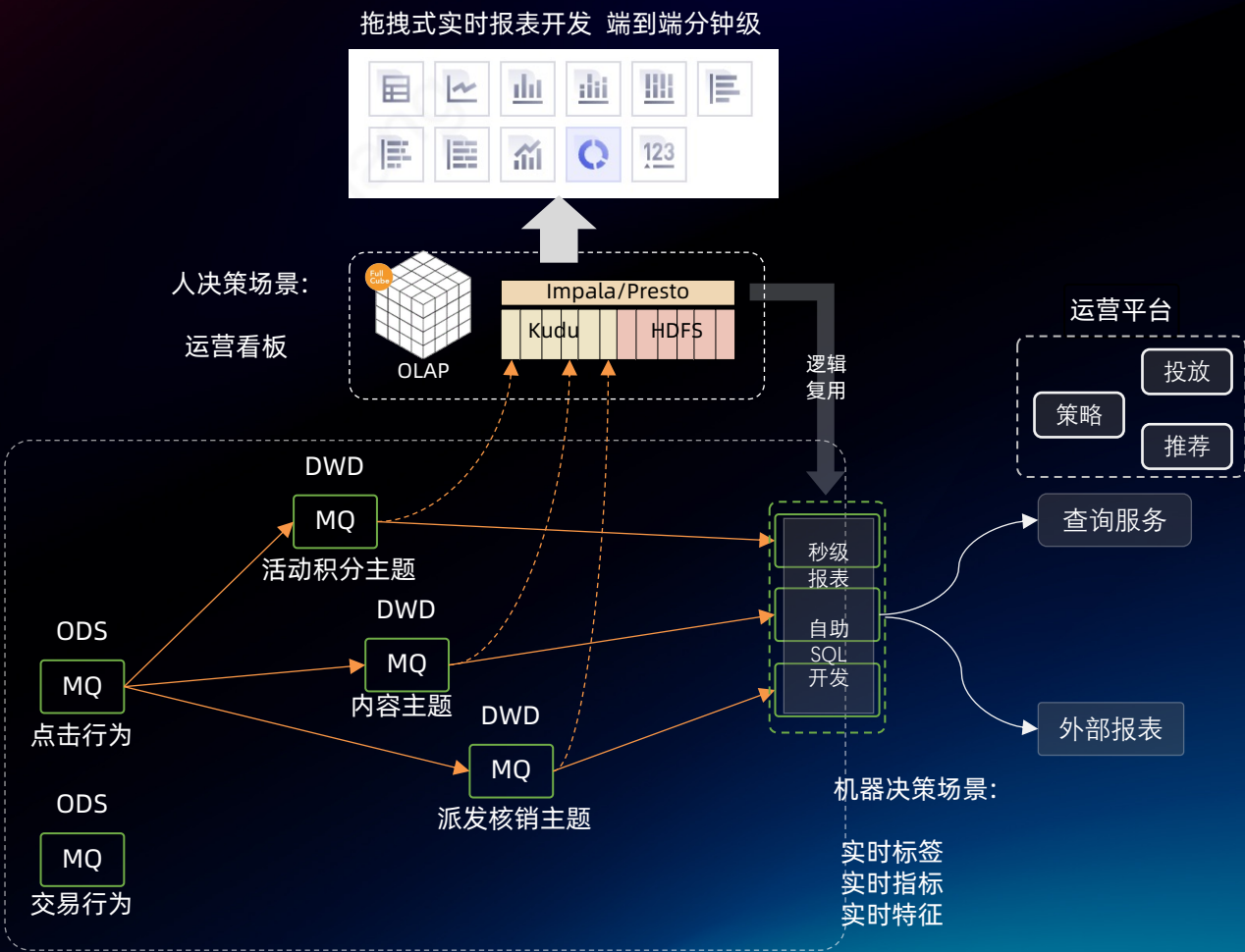
3. 解决方案

需求与方案	0.5天	拖拽式报表开发
代码开发	2.6天	提升SQL场景适配率
任务调试	1.5天	开发过程Fail Fast
数据验证	1.2天	配置化对账压测
查询接口与报表开发	1.8天	拖拽式创建
活动配置(策略、权益)	2天	
CR、压测、对账	3天	



封装存储和计算能力，对用户透明，打包成为完整解决方案。

3.1 总体架构

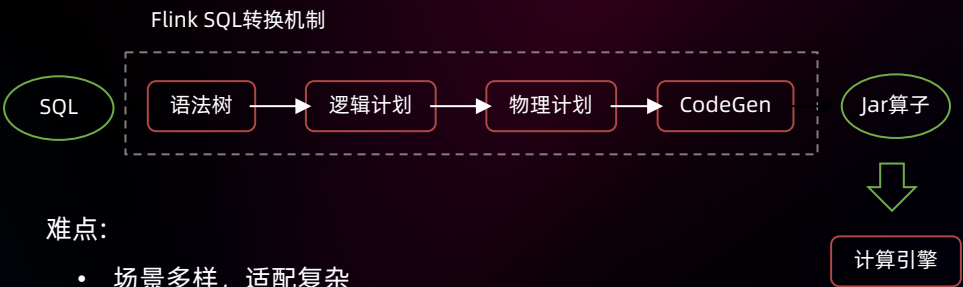


高效易用

3.2 SQL替代代码开发：高阶用户

Jar开发：学习和维护成本高
SQL开发：相对学习成本低，但场景适配率仅支持3.35%

目标：通过提升SQL场景覆盖率，完善开发工具链，让开发同学更专注业务逻辑

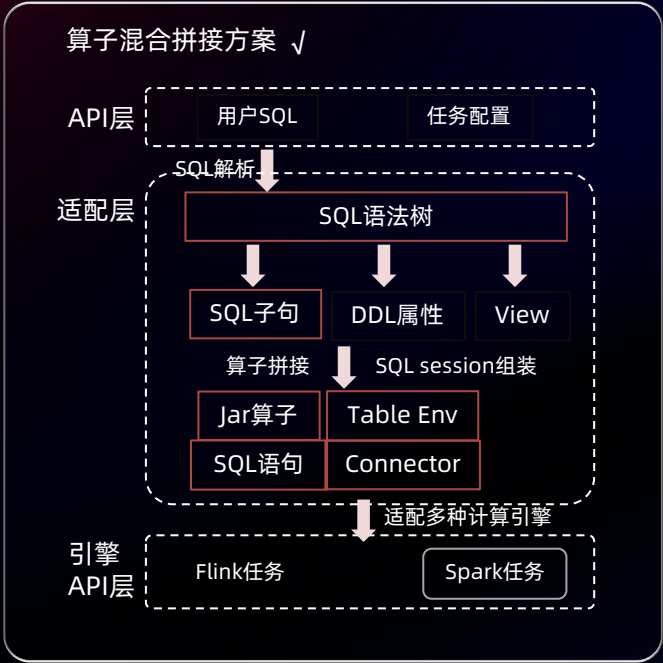


- 难点：
- 场景多样，适配复杂
 - 需遵循SQL标准，兼容Flink版本升级
 - SQL扩展点太少

方案对比：

方案	开发成本	维护成本	引擎兼容性	代表平台
扩展Flink SQL	高	高	很差	x
Flink SQL扩展点	非常高	高	好	未来Flink版本
重写SQL API	高	高	好	x
算子混合拼接	低	低	好	x

改造Flink：开发复杂、无法兼容社区版本、维护成本高、新版本适配成本高



Top5 SQL场景的适配方案

SQL场景 Top5	场景占比	扩展点
点击数据嵌套格式	28%	View
ODS去重	8%	View + DDL属性
MQ游标重置	5%	任务配置
自动化压测	3%	DDL属性
过滤回撤数据	%0.2	View + DDL属性

月新增任务SQL占比超过82%



场景接入后，沉淀为26个模板

通过SQL占比的提升，开发效率提升约150%

如何突破SQL固有的调优门槛？

3.3 专家经验辅助开发

【核心矛盾】

- 1. 用户SQL存在: 性能差、健壮性差、可维护性差
- 2. 试错成本高
- 3. CR耗时长, 依赖排期

【目标】

提前发现异常, 专家经验辅助用户开发

开发阶段	措施	Fail Fast手段
编辑	SQL即时校验	词法和语法检验, 元数据校验, CR规则校验, 专家经验
	数据源采样	真实数据采样校验
编译	预编译	元数据校验
调试	调试模式	Datagen模拟生成数据, Console输出阶段结果, 开启Debug日志 定位问题
	在线调试	真实数据试跑任务
对账	配置化对账	一键流批一体对账, 生成对账报告
压测	配置化压测	流量重放, 评估资源瓶颈

实时计算平台告警

风险等级: 中风险

时间: 2023-02-24 08:46:00

项目ID: 10021

任务ID:

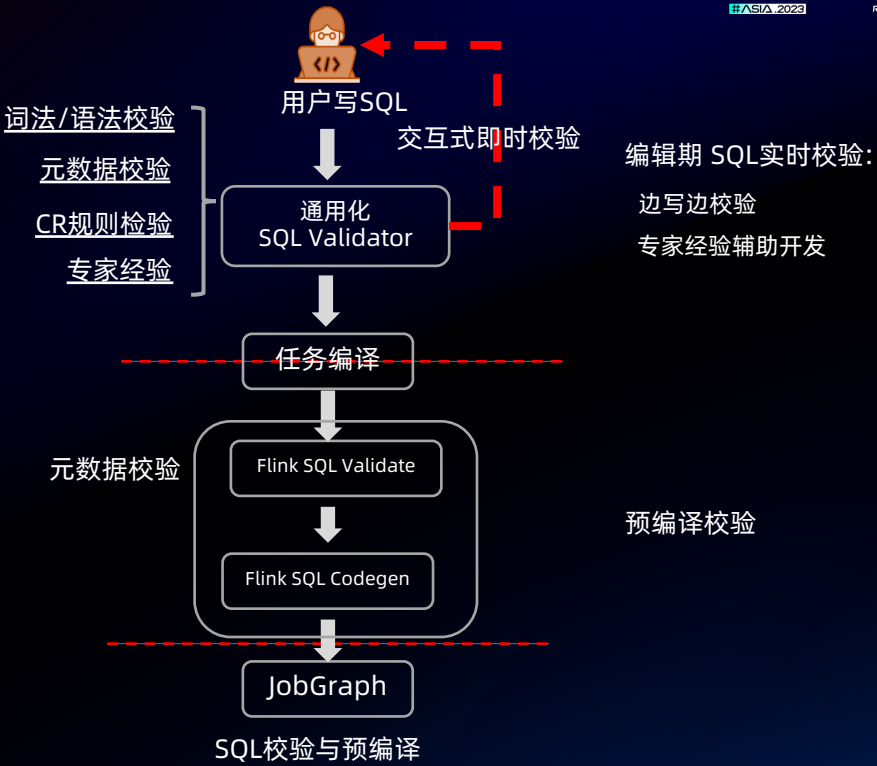
责任人:

任务名称:

任务异常: 检查点失败

诊断建议: checkpoint生成失败, 可能存在反压、状态过大或checkpoint保存超时等问题。建议增加计算资源、调整checkpoint的最小时间间隔或采用增量checkpoint或压缩状态数据

任务链接 修复指引



【效果展示】



1. 背景：大数据从规模化走向实时化、在线化的挑战

2. 原因分析

开发过程设计和实现不合理

设计出合理的方案依赖经验积累

需根据数据分布情况调整实现方式和资源

故障发现难

任务指标覆盖不全，定位工具缺乏

链路长，相互影响

组件多，指标多，需要综合分析

故障定位难

需了解全部组件的原理，甚至源码

需积累定位经验

需了解数据分布

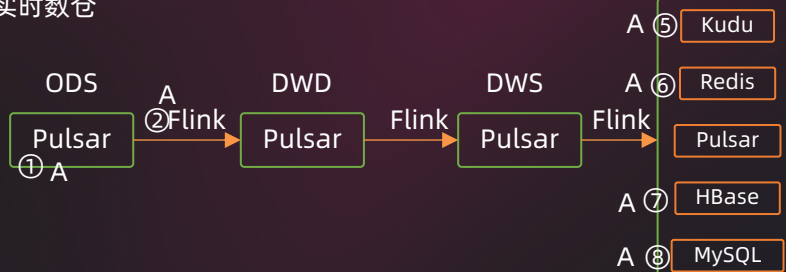
故障修复难

任务重启、调整资源、回放数据耗时长

数据恢复难保障一致性

跨任务复用checkpoint困难

实时数仓



实时计算平台告警
风险等级：高风险
时间：2023-02-16 11:06:00
项目ID：
任务ID：
责任人：
任务名称：
任务异常：作业重启
诊断建议：任务自动重启，可能存在第三方存储访问超时、TM心跳超时、slot丢失、class not found 或内存溢出等问题
任务链接 修复指引

这个还是之前那个null的原因吗

1-29 上午 10:46
这个任务可以帮忙看看吗，checkpoint 失败，但是没有重启，导致线上统计任务受影响

2020-10-6 上午 11:16
线上任务，非常紧急

某链路SLA要求：

链路总体延迟2S以内

实时看板类链路总体延迟小于10S

可用性要求达到3个9以上

数据可信率不低于 99%

- 必须找专业人员排查
- 定位问题耗时长，但实时任务延迟敏感
- 手动修复复杂
- 运维人力紧张，如何减少人肉运维

3. 解决方案



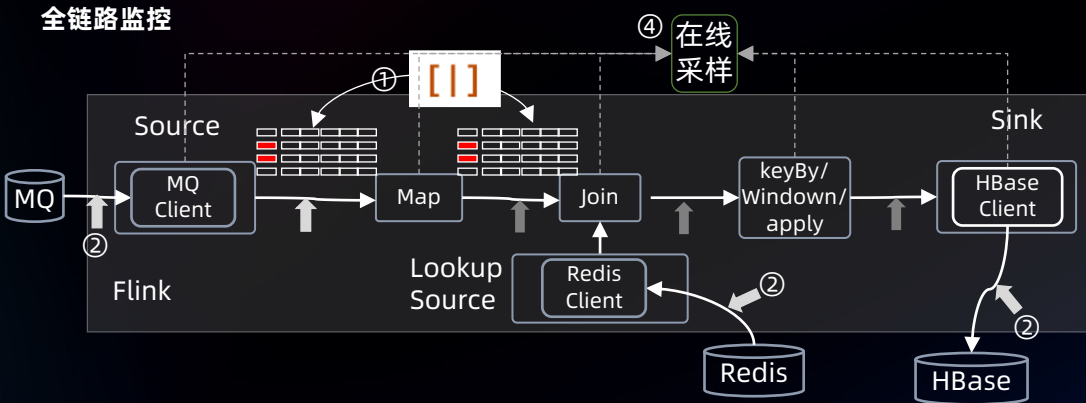
3.1 任务监控、自动诊断与任务巡检

【核心矛盾】

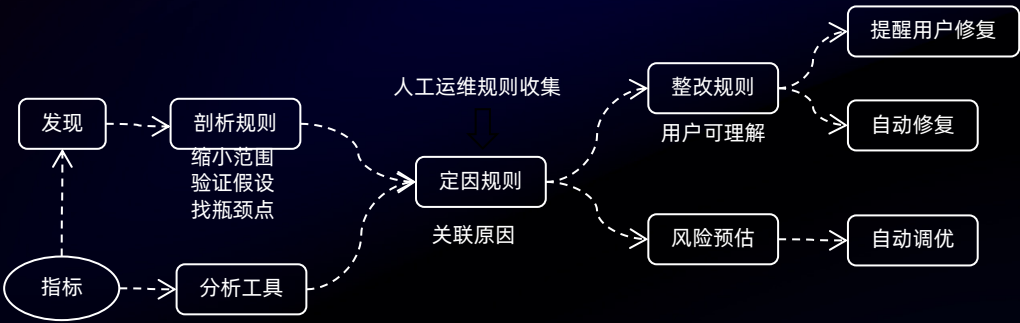
- 任务故障定位，人工定位门槛高、人力不足、耗时长
- 任务failover会造成流量叠加，加上算子重启的耗时，数据就会断流或超时

【解决方案】

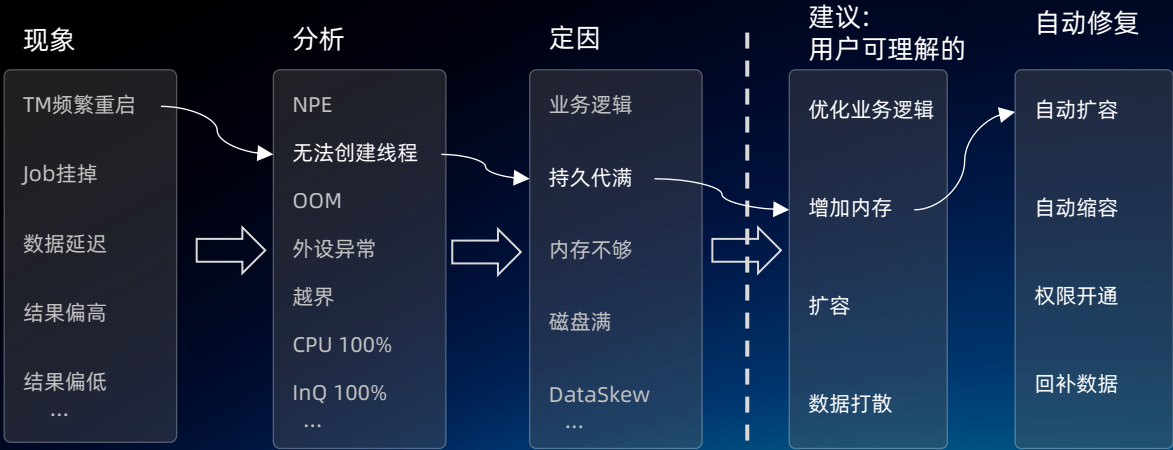
- 任务全链路监控：算子、节点和上下游等的全部指标打通和联动
- 任务诊断：结合专家经验，分析指标，实现故障的自动快速定位，部分问题自动修复
- 任务健康巡检：例行化巡检任务，发现隐患问题(避免failover)，并提供整改建议



故障自动修复与任务巡检



故障自动修复与隐患发现样例

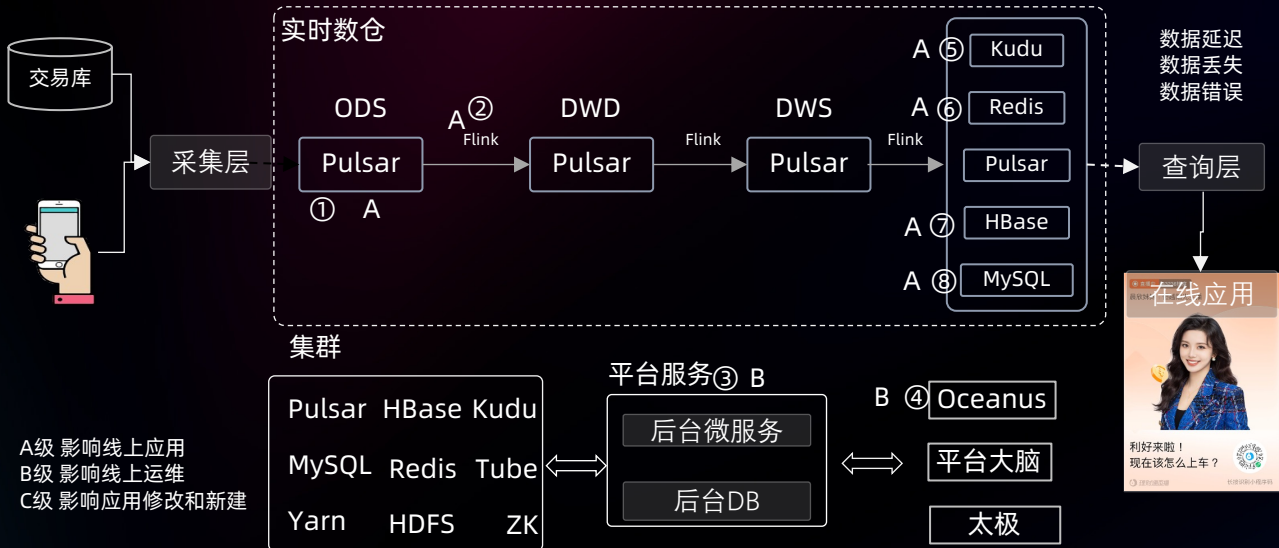


与Oceanus、平台大脑共建任务诊断和巡检能力

3.2 金融大数据自治体系

【核心矛盾】

- ✓ 链路复杂：定位难，诊断难，修复难
- ✓ 需要结合任务、集群、链路、数据内容等多种的指标
- ✓ 运维门槛高，需要各个环节的专业技术经验
- ✓ 运维工具缺乏



实时线上链路，涉及12个集群，其中8个集群一旦出问题会影响线上业务。

自动化运维，势在必行！

【解决方案】

金融大数据自治体系



- ✓ 平台大脑Oteam的 首批发起者
- ✓ 自动化运维能力：截止2022年10月，已覆盖3000+物理节点

自动化数据质量管理

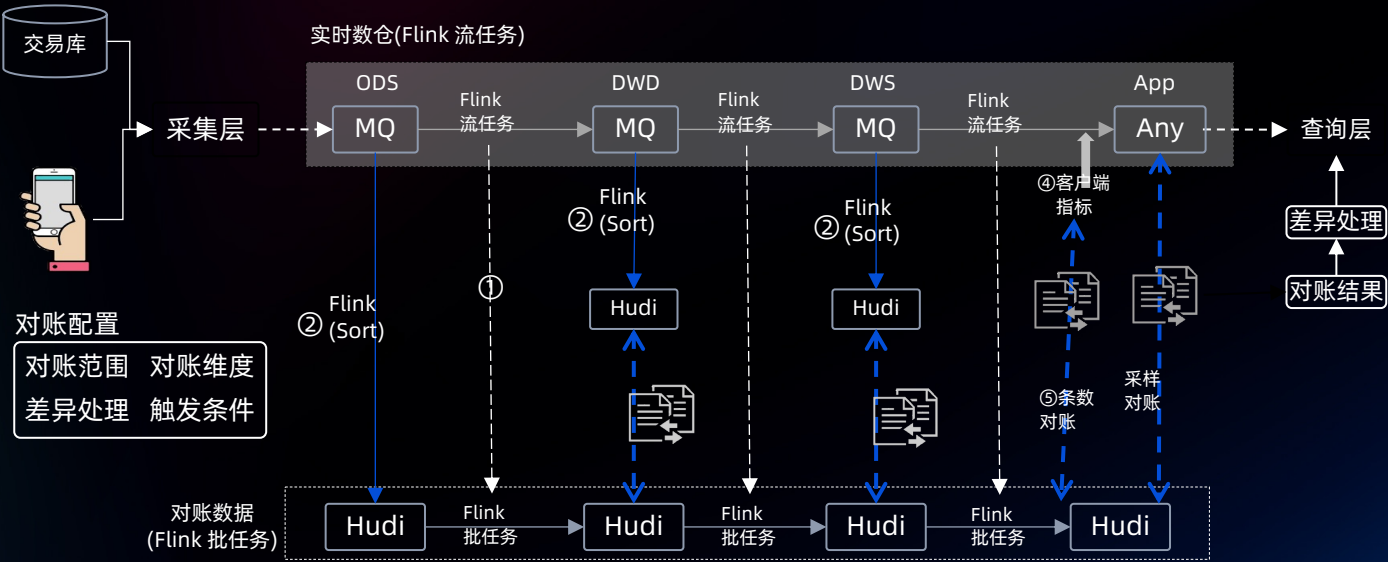
【核心矛盾】

- 任务链路长，依赖关系复杂，计算逻辑多样，数据异常难以发现
- 数据一旦出现异常，无法回补线上应用
- 现有组件缺乏对账和数据质量管理工具

【解决方案】

- 数据源监控：监控数据源的乱序、延迟、元数据等，保证不影响下游处理
- 实时对账：与离线数据核对，处理异常数据
- 数据趋势分析：监控数据变化趋势，预估数据问题
- 数据审计：追踪异常数据流向，定位问题

方案	优势	劣势	代表平台
与离线链路核对	需管理规范 数据流批复用	依赖离线链路的质量 离线链路开发成本高 数据口径难管理 双倍存储	xx-DQA
Flink Jar流批一体对账	易于管理	只用于分析引擎 任务需基于相同SDK	xx
Flink SQL流批一体对账	开发和维护成本低 对账数据延迟低	双倍存储 Flink引擎改造成本高	xx



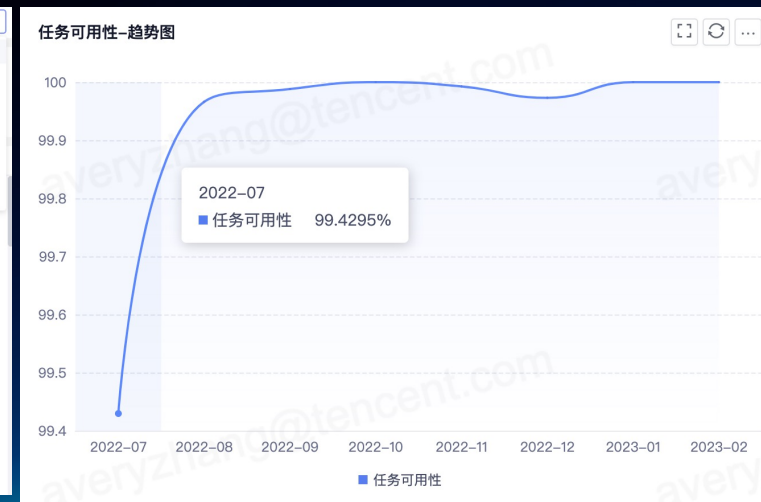
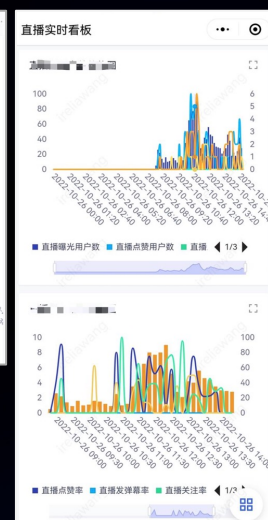
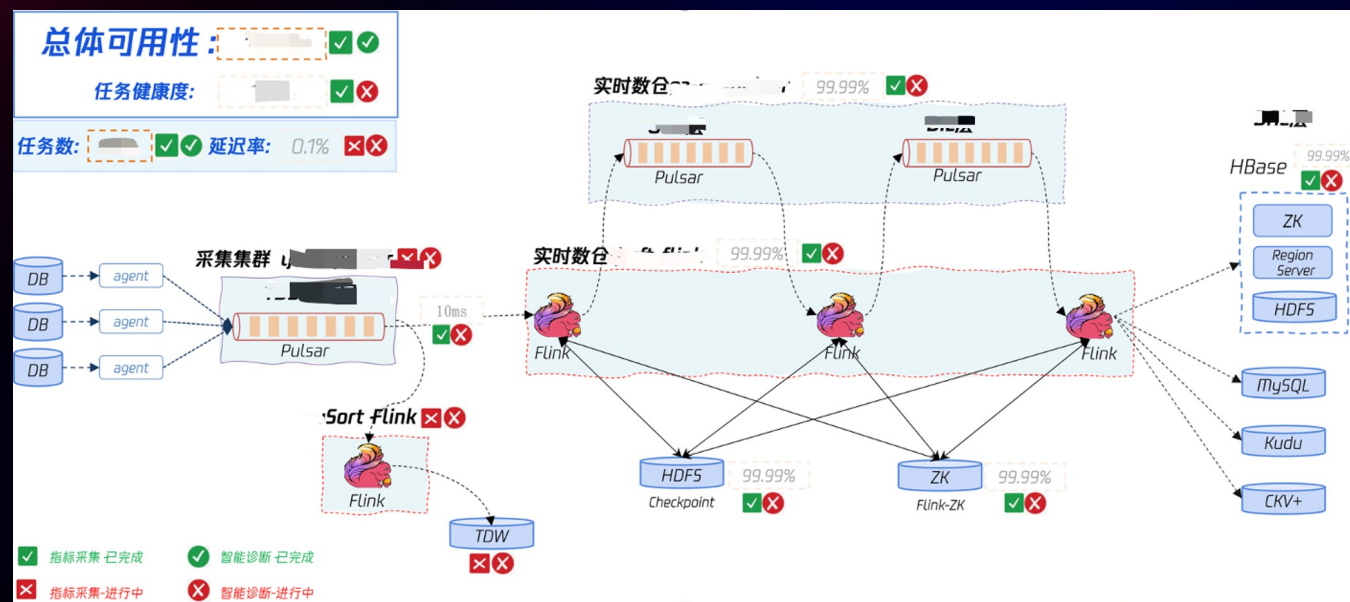
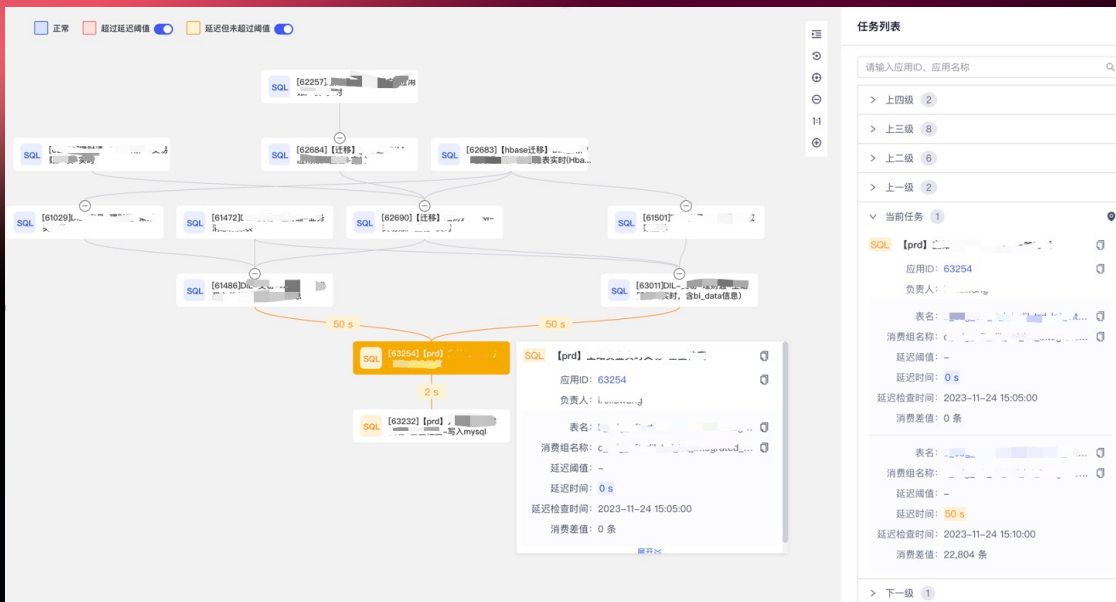
【开源协同】

- 与Oceanus合作，基于Flink SQL的流批一体对账

【业务效果】

- 对账开发耗时从3天提升到分钟级

效果展示



背景

大数据贴近业务
大数据在线化

催化剂

降本增效

改变

【述求&价值】

- 精细化
- 敏捷化
- 低成本化

赋能

【全面实时化】

- 核心交易场景
- 证券交易场景
- 跨境交易场景

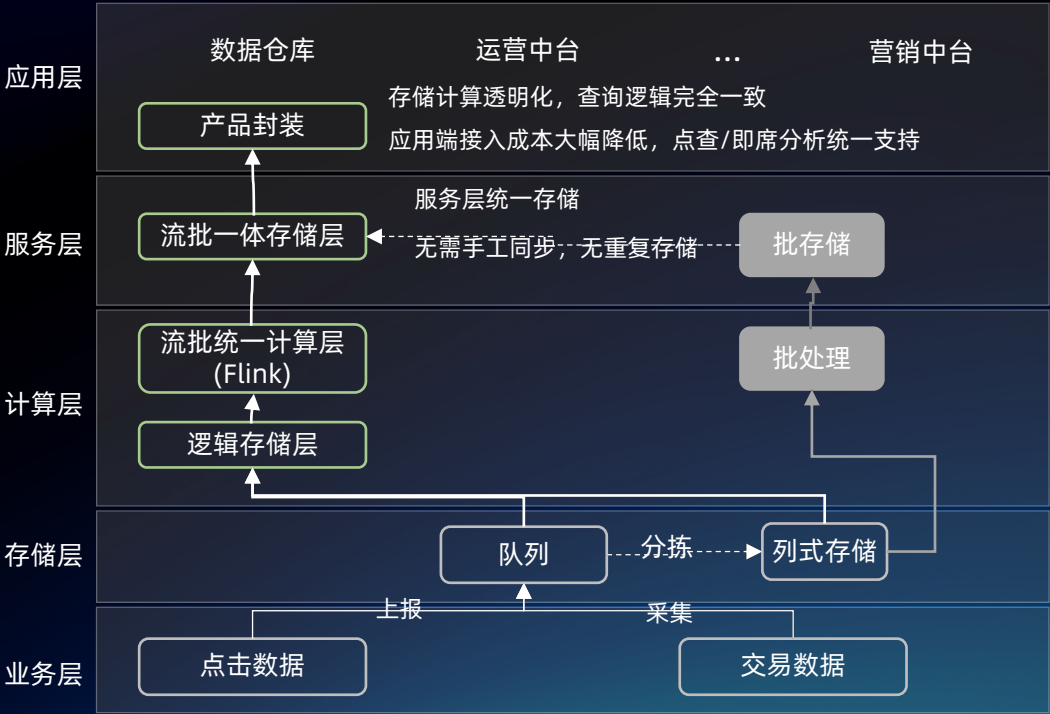
赋能

【架构升级】

- 湖仓一体
- 数仓提速
- 云原生
- 数仓智能化

未来

金融大数据 3.0 --- 云原生流式数仓



THANKS

FLINK FORWARD #ASIA.2023



一站式海量数据集成
github.com/apache/inlong



腾讯实时开发引擎



腾讯大数据自动驾驶