滴滴实时计算平台架构与实践

The Application and Architecture of the Real-Time Computing Platform in DiDi

公司：滴滴出行

Company: DiDi Chuxing

职位：技术研究员

Title：Director of BigData Infrastructure Department

演讲者：罗李

Lecturer： Luo Li

FLINK FORWARD

Agenda

- 背景及现状
- 实时规则匹配实践
- StreamSQL介绍
- 实时计算平台框架
- 总结和规划

- Background
- The CEP practice
- StreamSQL
- Stream Platform Infrastructure
- Future plans

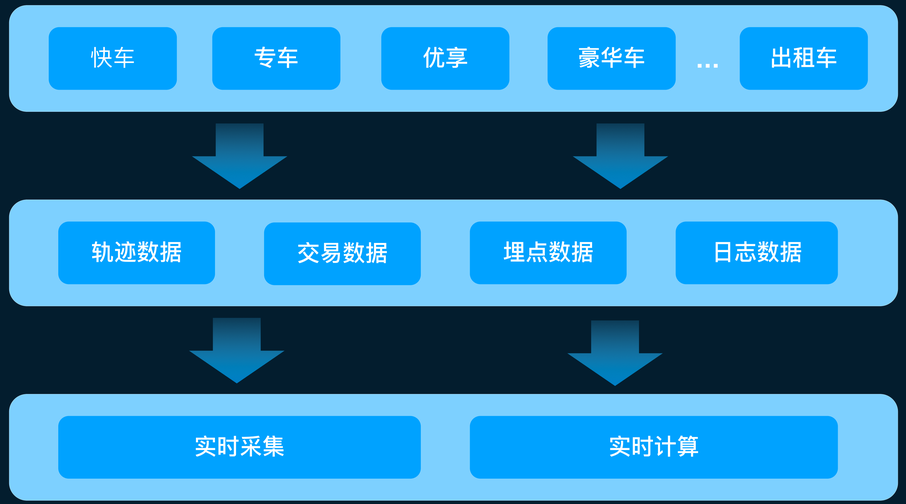
FLINK FORWARD

业务特点

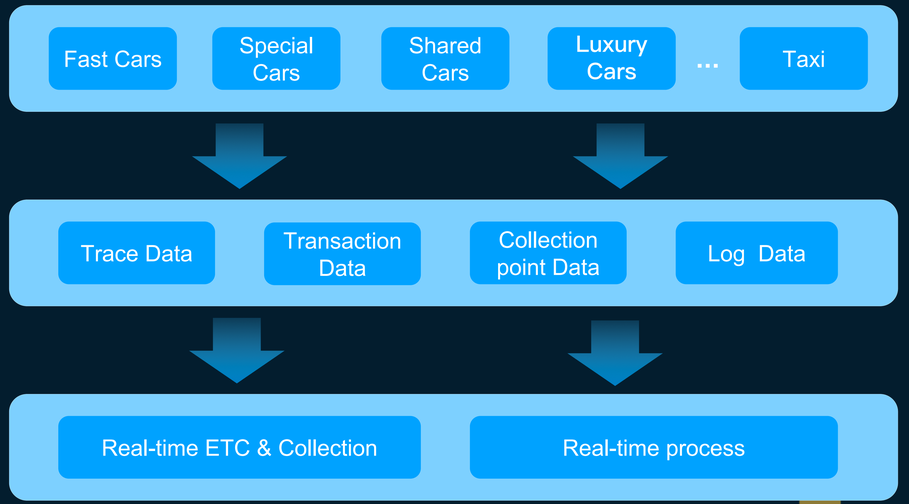
characteristic of business

FLINK FORWARD

- 滴滴本身是一个实时交易引擎
- 滴滴的数据和场景天然是实时的



- DiDi is a realtime Trade Engine
- Didi's data and scenes are naturally real-time



FLINK FORWARD

流计算技术演进

第一阶段：业务方自建小集群

第二阶段：集中式大集群、平台化

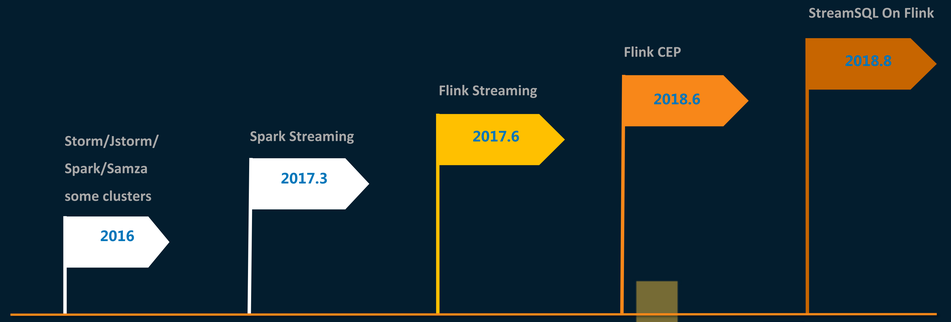
第三阶段：流计算开发SQL化

Didi's Real-time develop steps

First Step：A lot of some Clusters

Second Step：Single Big Shared Cluster

Third Step：SQL On Stream process



FLINK FORWARD

应用场景

- 实时发券、实时风控、实时异常检测等
- 交易监控、服务监控、工单监控等

- 数据从消息队列到HBase、Druid、ES等存储
- 实时乘客特征、司机特征、订单特征等

Scenarios

- 实时发券、实时风控、实时异常检测等
- 交易监控、服务监控、工单监控等

- 数据从消息队列到HBase、Druid、ES等存储
- 实时乘客特征、司机特征、订单特征等

FLINK FORWARD

应用规模

服务业务线　集群规模　流计算任务数　每日处理数据量

Data Scale

Applications　Cluster Scale　Streaming Jobs　Data Scale per day

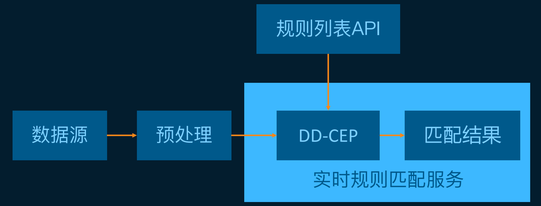
FLINK FORWARD

实时规则匹配（CEP）实践

The CEP practice

实时规则匹配（CEP）

```
Pattern
.begin(s1)
   .where(key=bubble && city=166)
.notFollowedBy(s2)
   .where(key=order_send && city = 166)
.timeEnd(s3)
   .timeShiftMore(s1, Time.ms(5000))
```



个性化营销

- “某城市乘客冒泡后5秒没有下单”
- “某城市下单后30秒没人接单”
- 峰值3000+规则

异常工单检测

- “订单状态长时间(10分钟)停留在–司机已抢单”
- “订单状态长时间(1小时)停留在–开始计费”
- “拼车与dos订单状态不一致–司机已抢单”

CEP功能&性能改进

CEP功能改进

- 支持wait算子
- 支持DSL语言
- 支持单任务多规则
- 支持规则动态修改

CEP性能改进

- SharedBuffer重构
- 增加访问缓存（已贡献社区）
- 简化event time语义处理
- 复用Condition Context（已贡献社区）

FLINK FORWARD

StreamSQL

FLINK FORWARD

StreamSQL建设

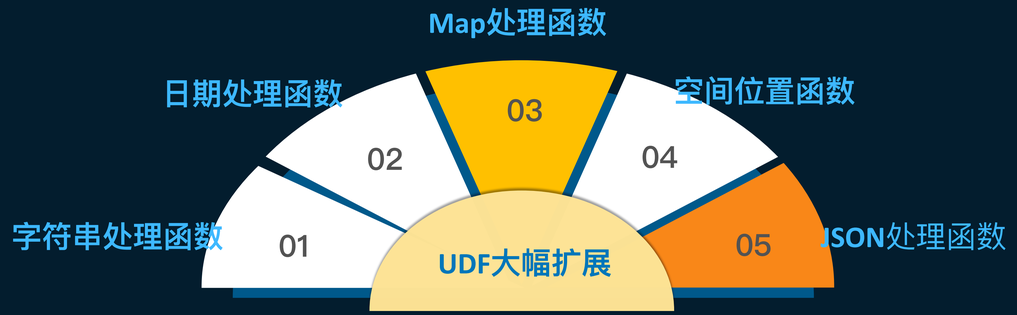
扩展DDL

- 打通滴滴内部主流消息队列和实时存储
- 通过扩展DDL语句描述Source和Sink
- 内置JSON、标准日志、PB格式解析

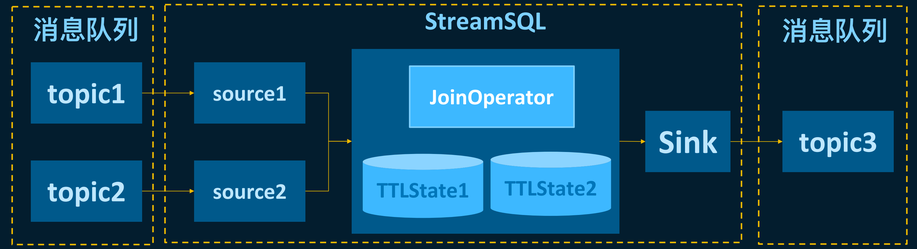
保存 | 语法检查 | + 任务配置 | + 发布任务 | 查看任务

```
CREATE STREAM SOURCE source (
    uid '$.uid' VARCHAR ,
    biz_type '$.biz_type' INT,
    data '$' VARCHAR
) WITH (
    stype = 'kafka',
    topic = 'evcollector_gs_driver',
    brokers= 'ip:port,ip:port',
    encode = 'json',
    parallelism = 1,
    groupId = 'flinksql1',
    auto.offset.reset = 'latest'
);
```

UDF大幅扩展


Map处理函数
日期处理函数
空间位置函数
02
03
04
字符串处理函数
01
UDF大幅扩展
05
JSON处理函数

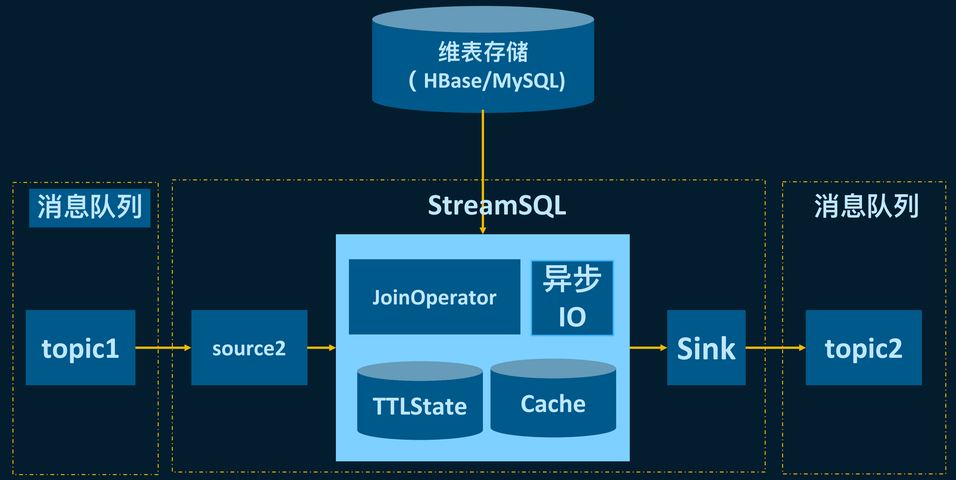
支持基于TTL的双流Join



- inner join
- Left join
- Full join

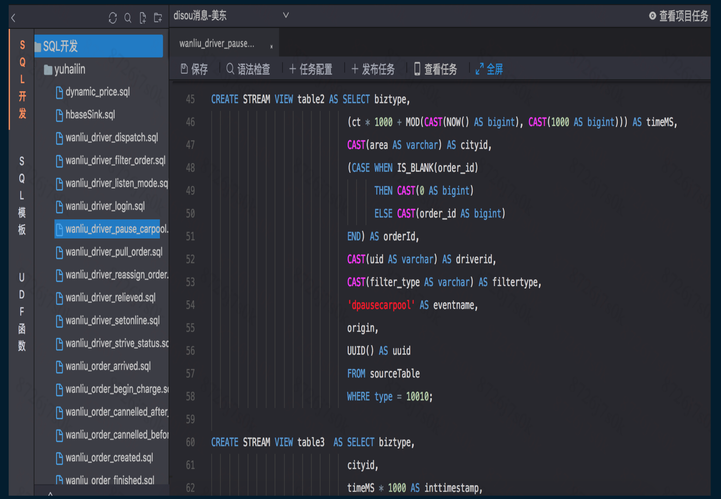
```
SELECT maopao_source.user_id AS user_id,
       maopao_source.bundle_id AS bundle_id,
       maopao_source.src,maopao_source.dst,
       maopao_source.price,
       order_created_source.customer_name,
       order_created_source.order_id,
       order_created_source.ts
FROM
 order_created_source
INNER JOIN
 maopao_source
ON  order_created_source.user_id = maopao_source.user_id;
```

维表JOIN


维表存储
(HBase/MySQL)
消息队列
StreamSQL
消息队列
JoinOperator
异步
IO
topic1
source2
Sink
topic2
TTLState
Cache

FLINK FORWARD

SQL IDE



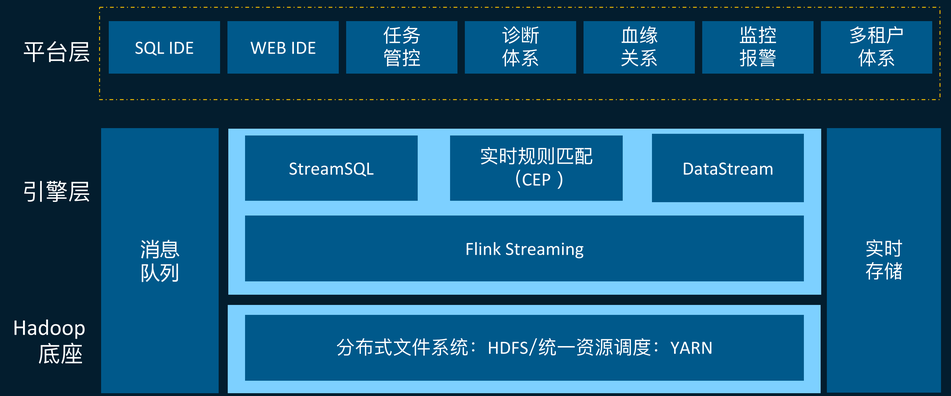
- StreamSQL Editor
- SQL模板
- UDF函数说明
- 语法检测
- DEBUG
- 版本管理

FLINK FORWARD

实时计算平台架构

Stream Platform Infrastructure

实时计算平台架构


平台层
SQL IDE
WEB IDE
任务
管控
诊断
体系
血缘
关系
监控
报警
多租户
体系
引擎层
消息
队列
StreamSQL
实时规则匹配
（CEP ）
DataStream
Flink Streaming
实时
存储
Hadoop
底座
分布式文件系统：HDFS/统一资源调度：YARN

FLINK FORWARD

三种流计算开发模式

本地IDE开发

Web IDE开发

SQL IDE开发

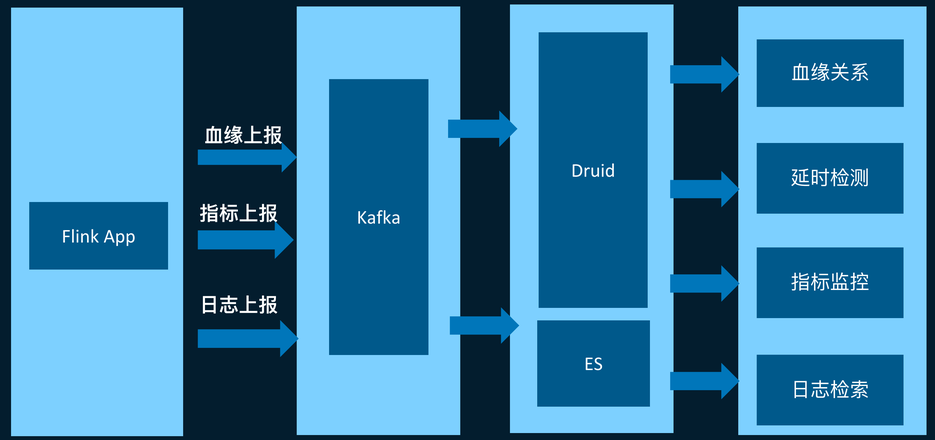
Web化任务管控

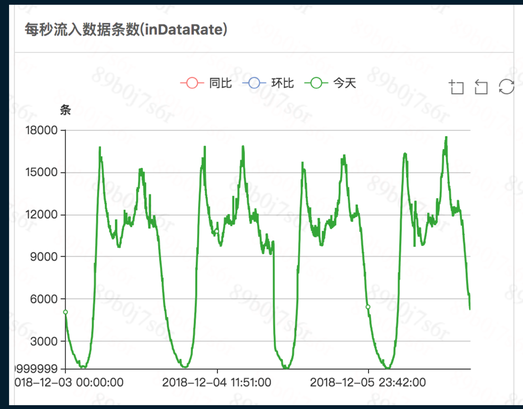
消除客户机

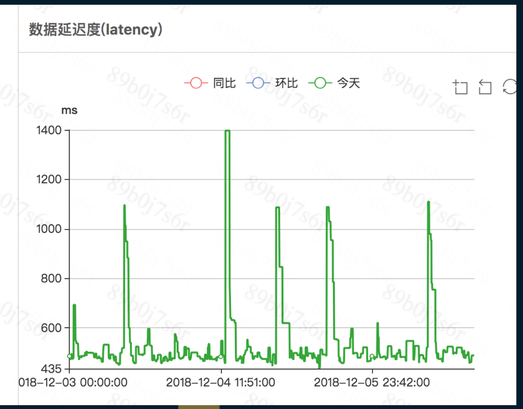
提供入门模板

内置参数调优

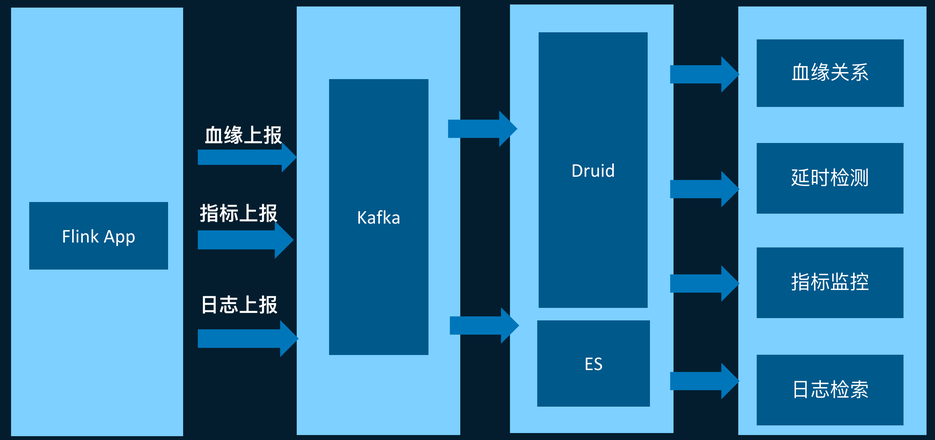
内置指标上报


Flink App
血缘上报
指标上报
日志上报
Kafka
Druid
ES
血缘关系
延时检测
指标监控
日志检索


每秒流入数据条数(inDataRate)
同比
环比
今天
条
18000
15000
12000
9000
6000
3000
999999
018-12-03 00:00:00
2018-12-04 11:51:00
2018-12-05 23:42:00


数据延迟度(latency)
同比
环比
今天
ms
1400
1200
1000
800
600
435
018-12-03 00:00:00
2018-12-04 11:51:00
2018-12-05 23:42:00

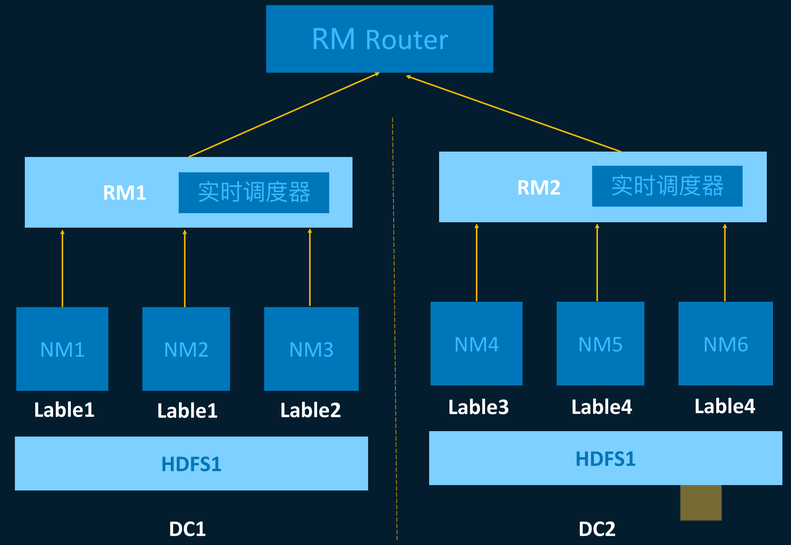
数据血缘关系


Flink App
血缘上报
指标上报
日志上报
Kafka
Druid
ES
血缘关系
延时检测
指标监控
日志检索


MySQL/日志
Topic
流计算任务
Datasource
实时指标
监控面板
报警任务
请输入节点名称关键字...
Backward
Forward
H布局
V布局
上游
下游

Hadoop底座

- 独立建设实时计算集群，避免离线任务干扰
- 实时集群尽量靠近业务，降低网络时延
- 定制调度器
- 调度均衡
- 基于CPU调度
- 基于Node Label和CGroup双重隔离机制
- 多机房部署



总结与规划

Future plans

FLINK FORWARD

总结

- StreamSQL 大幅降低流计算开发门槛
- 实时规则匹配（CEP） 在个性化营销、实时风控等领域落地
- 实时计算平台 一站式流计算开发平台

Summary

- StreamSQL makes the Streaming develop much easier
- Real-time CEP very useful in the risk control & Personalized marketing
- A Big Shared Stream Processing platform

FLINK FORWARD

Flink规划

- 完成Spark Streaming到StreamSQL的改造
- IoT领域发力
- StreamML建设

Flink Plans

- Streaming Job Spark Streaming -> StreamSQL upgrade
- In IoT (Cars/bicye/Monitor in Cars)
- StreamML

THANKS