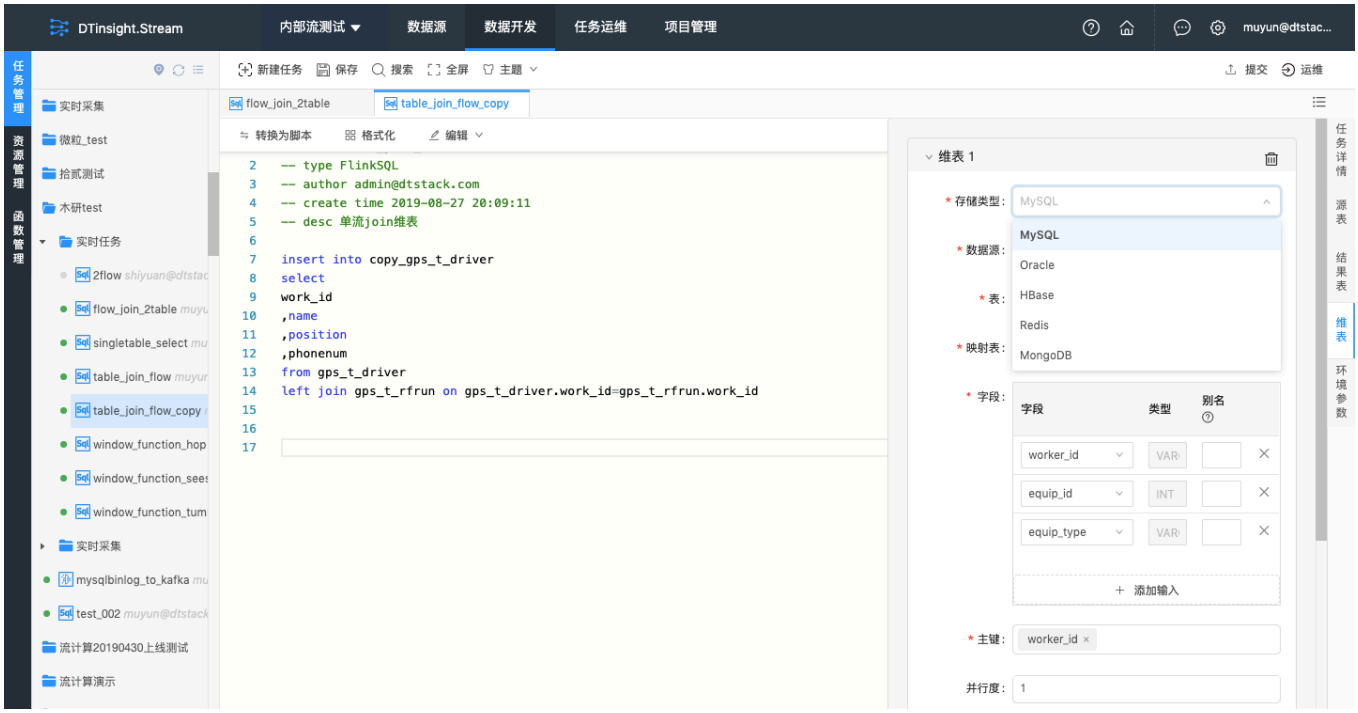
基于扩展FlinkSQL实现流计算的源表导入、维表关联与结果表导出

Streamworks，袋鼠云基于SQL的流计算开发平台，其通过扩展FlinkSQL，实现FlinkSQL与界面化配置映射结合的方式，完成Kafka源数据的读入，并支持流数据与Mysql/Oracle/MongDB等数据源进行维表关联，将最终结果数据导出至Hbase/ES/Greenplum/Oracle/OceanBase等目标数据库，进行一站式的流数据开发。



为什么扩展Flink-SQL?

Flink 本身的SQL语法并不提供对接输入源和输出目的的SQL语法，数据开发在使用过程中需要根据其提供的API接口编写Source和 Sink，不仅需要了解FLink 各类Operator的API，还需要对各个组件的相关调用方式有了解（比如Kafka，Redis，Mongo、Hbase等），异常繁琐。并且在需要关联到外部数据源的时候Flink也没有提供SQL相关的实现方式，若数据开发直接基于原生的Flink SQL进行实时的数据分析，需要较大的额外工作量。

袋鼠云的SteamWorks则聚焦于数据开发人员使用Flink SQL时专注于业务逻辑，只需要关心做什么，而不需要关心怎么做。研发团队对FlinkSQL进行了扩展，用户只需通过可视化配置，完成源表到导入、维表的关联、结果表的导出。

扩展了哪些Flink相关SQL

1.创建源表语句

CREATE TABLE tableName(

```
  colName colType,
  ...
  function(colNameX) AS aliasName,
  WATERMARK FOR colName AS withOffset( colName , delayTime )
```

)WITH(

```
  type ='kafka09',
  bootstrapServers ='ip:port,ip:port...',
  zookeeperQuorum ='ip:port,ip:port/zkparent',
  offsetReset ='latest',
  topic ='topicName',
  parallelism ='parllNum'
```

);

2.创建输出表语句

CREATE TABLE tableName(

colName colType,

...

colNameX colType

)WITH(

type ='mysql',

url ='jdbcUrl',

userName ='userName',

password ='pwd',

tableName ='tableName',

parallelism ='parllNum'

);

3.创建自定义函数；

Create (scala|table) FUNCTION name WITH com.xx.xx

4.维表关联语句；

CREATE TABLE tableName(

```
    colName cloType,
    ...
    PRIMARY KEY(keyInfo),
    PERIOD FOR SYSTEM_TIME
)WITH(
    type='mysql',
    url='jdbcUrl',
    userName='dbUserName',
    password='dbPwd',
    tableName='tableName',
    cache ='LRU',
    cacheSize ='10000',
    cacheTTLMs ='60000',
     parallelism ='1',
     partitionedJoin='false'
 );
```

各个模块是如何翻译到Flink，进行数据处理

1.如何将创建源表的SQL语句转换为Flink的operator

Flink中表的都会映射到Table这个类。然后调用注册方法将Table注册到environment, StreamTableEnvironment.registerTable(tableName, table)。

当前我们只支持kafka数据源。Flink本身有读取kafka 的实现类（FlinkKafkaConsumer09），所以只需要根据指定参数实例化出该对象，并调用注册方法注册即可。

另外需要注意在Flink SQL经常会需要用到Rowtime、Proctime, 所以我们在注册表结构的时候额外添加Rowtime、Proctime。当需要用到rowtime的时候需要额外指定DataStream.watermarks(assignTimestampsAndWatermarks)，自定义watermark主要做两个事情1：如何从Row中获取时间字段。 2：设定最大延迟时间。

2.如何将创建的输出表sql语句转换为Flink的operator

Flink输出Operator的基类是OutputFormat, 我们这里继承的是RichOutputFormat, 该抽象类继承

OutputFormat，额外实现了获取运行环境的方法getRuntimeContext(), 方便于我们之后自定义metric等操作。

我们以输出到Mysql插件Mysql-Sink为例。

分两部分

（1）将create table 解析出表名称，字段信息，mysql连接信息。

该部分使用正则表达式的方式将create table 语句转换为内部的一个实现类。该类存储了表名称，字段信息，插件类型，插件连接信息。

（2）继承RichOutputFormat将数据写到对应的外部数据源。

主要是实现writeRecord方法，在mysql插件中其实就是调用jdbc 实现插入或者更新方法。

3.如何将自定义函数语句转换为Flink的operator;

Flink对udf提供两种类型的实现方式：

继承ScalarFunction
继承TableFunction
需要做的将用户提供的jar添加到URLClassLoader, 并加载指定的class (实现上述接口的类路径),然后调用TableEnvironment.registerFunction(funcName, udfFunc)；即完成UDF的注册。之后即可使用改定义的UDF;

4.维表功能是如何实现的?

流计算中一个常见的需求就是为数据流补齐字段。因为数据采集端采集到的数据往往比较有限，在做数据分析之前，就要先将所需的维度信息补全，但是当前Flink并未提供join外部数据源的SQL功能。

实现该功能需要注意的几个问题

（1）维表的数据是不断变化的

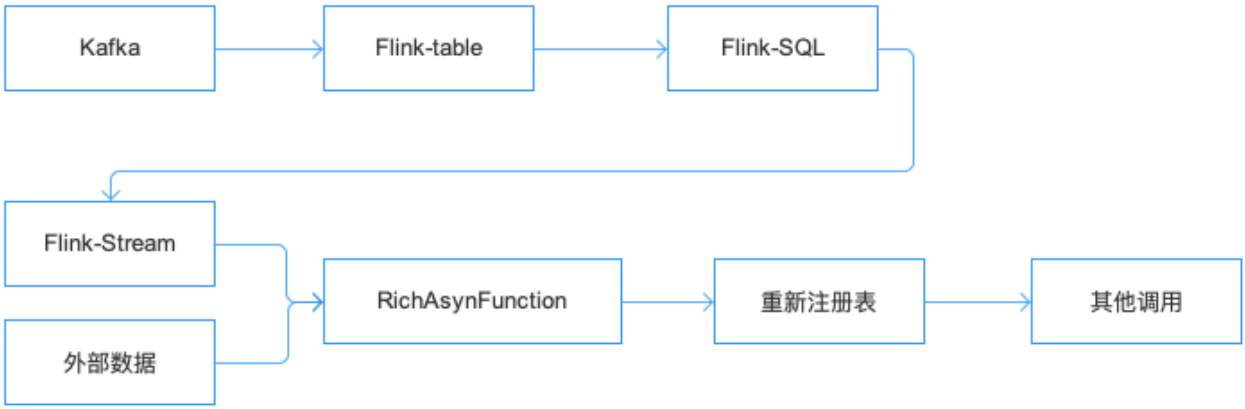
在实现的时候需要支持定时更新内存中的缓存的外部数据源，比如使用LRU等策略。

（2）IO吞吐问题

如果每接收到一条数据就串行到外部数据源去获取对应的关联记录的话，网络延迟将会是系统最大的瓶颈。这里我们选择阿里贡献给flink社区的算子RichAsyncFunction。该算子使用异步的方式从外部数据源获取数据，大大减少了花费在网络请求上的时间。

（3）如何将SQL 中包含的维表解析到Flink operator

为了从SQL中解析出指定的维表和过滤条件，使用正则明显不是一个合适的办法，需要匹配各种可能性，将是一个无穷无尽的过程。查看Flink本身对SQL的解析，它使用了calcite做为sql解析的工作。将sql解析出一个语法树，通过迭代的方式，搜索到对应的维表，然后将维表和非维表结构分开。



通过上述步骤便可以通过Flink SQL完成常用的从kafka源表读取数据，join部数据源，并写入到指定的外部目标结构中，且袋鼠云开源了相关实现：https://github.com/DTStack/flinkStreamSQL

Demo：Kafka流数据关联Oracle维表，写入PostgreSQL 。

读取Kafka源数据：

```
CREATE TABLE MyTable(
    name varchar,
    channel varchar,
    pv INT,
    xctime bigint,
    CHARACTER_LENGTH(channel) AS timeLeng
 )WITH(
    type ='kafka09',
    kafka.bootstrap.servers ='172.16.8.198:9092',
    kafka.zookeeper.quorum ='172.16.8.198:2181/kafka',
    kafka.auto.offset.reset ='latest',
    kafka.topic ='nbTest1,nbTest2,nbTest3',
    --kafka.topic ='mqTest.*',
    --patterntopic='true'
    parallelism ='1',
```

```
    sourcedatatype ='json' #可不设置
 );
```

维表关联Oracle语句

```
create table sideTable(
    channel varchar,
    info varchar,
    PRIMARY KEY(channel),
    PERIOD FOR SYSTEM_TIME
 )WITH(
    type='oracle',
    url='jdbc:oracle:thin:@xx.xx.xx.xx:1521:orcl',
    userName='xx',
    password='xx',
    tableName='sidetest',
    cache ='LRU',
    cacheSize ='10000',
    cacheTTLMs ='60000',
    cacheMode='unordered',
    asyncCapacity='1000',
    asyncTimeout='10000'
    parallelism ='1',
    partitionedJoin='false',
    schema = 'MQTEST'
 );
--创建PostgreSQL输出表语句
```

Greenplum、LibrA数据库写入方式与PostgreSQL类似

```
CREATE TABLE MyResult(
    channel VARCHAR,
    info VARCHAR
 )WITH(
    type ='postgresql',
    url ='jdbc:postgresql://localhost:9001/test?sslmode=disable',
    userName ='dtstack',
    password ='abc123',
    tableName ='pv2',
    parallelism ='1'
 );
--基于FlinkSQL进行Kafka流数据读入关联Oracle维表，并写入PostgreSQL的实现
insert
into
    MyResult
    select
        a.channel,
        b.info
    from
        MyTable a
    join
        sideTable b
            on a.channel=b.channel
```