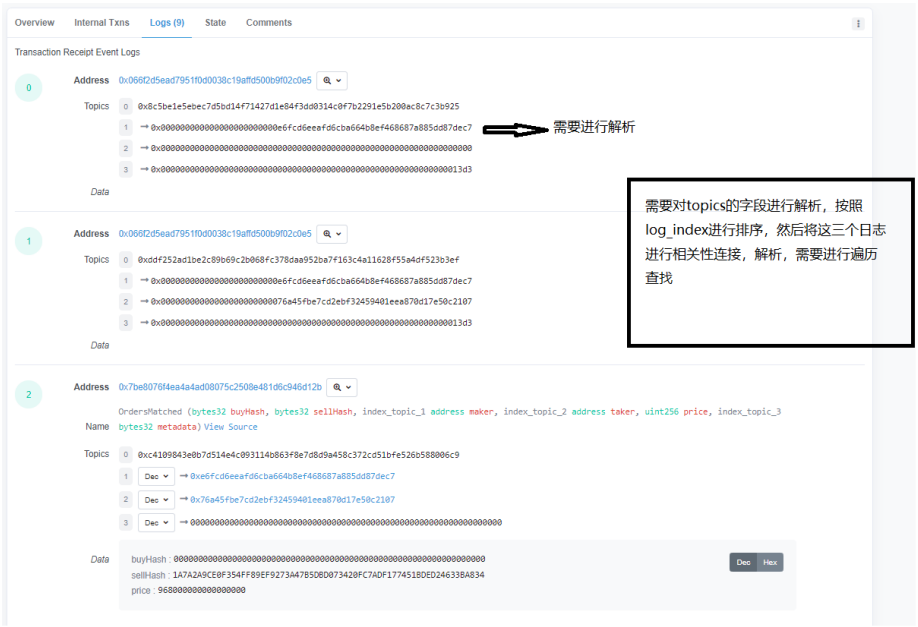
说明文档

1. 需求

解析出以太坊中的opensea的字段，需要进行局部排序，数据提取转换



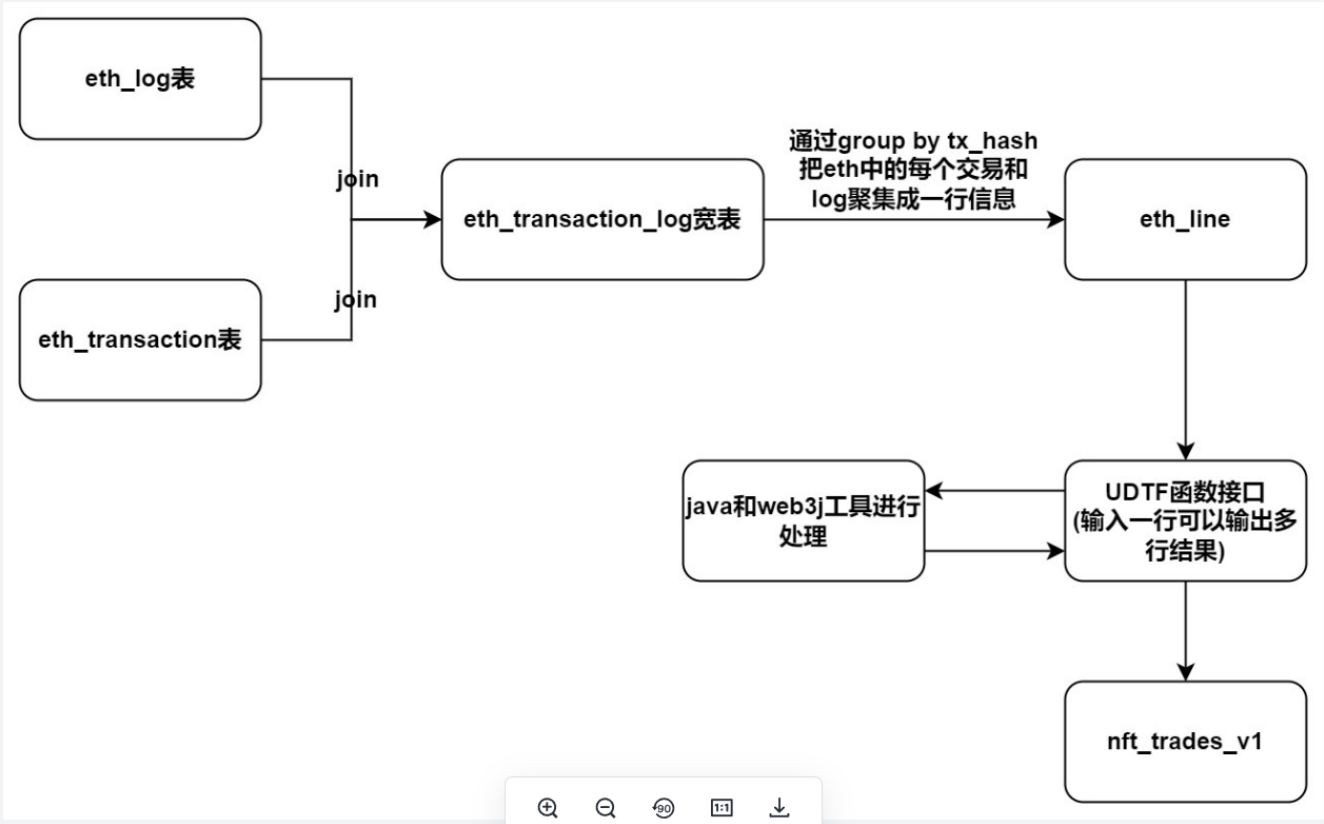
2. 概要

在大数据中，一些复杂的需求，仅凭sql是不能实现的。在这种情况下，一般用spark(flink)的算子来实现。但这种方法有些局限性：

1. 当我们改某个需求(如分析的天数改变)的时候，需要改源代码，不好维护。
2. 对于其他开发的人员来说，需要进行源码阅读，才能进行代码更改

3. 解决流程

首先将要分析的数据按最小粒度聚集成一行，然后利用hive的自定义函数的接口，将数据写入接口中，然后在接口里面进行分析，然后再将数据从接口中输出



4. 实现流程

a. 将数据拼接，并聚集成一行

SQL

```sql
 with
log_tmp as
(
    select txn_hash,
           concat_ws('_', collect_set(concat_ws('-', address, topics, data, erc_type, symbol, cast(decimals as string), cast(log_index as string)))) as log_line
    from eth.dwd_eth_log_erctoken
    where txn_hash = '0xdea04ee76bd891ff04b73c48e21bef2550dc9513267876398d5bf77ca519dd17' and dt = '0000-00-09'
    group by txn_hash
),
txn_tmp as
(
   select txn_hash ,
          concat_ws('#', cast(block_height as string), `timestamp`, txn_hash, txn_from, txn_to, cast(txn_chainid as string)) as txn_line
   from eth.dwd_eth_block_transaction
   where txn_hash = '0xdea04ee76bd891ff04b73c48e21bef2550dc9513267876398d5bf77ca519dd17' and dt = '0000-00-09'
),
source_tmp as
(
   select concat_ws('#', tt.txn_line, lt.log_line) as line
   from log_tmp lt join txn_tmp tt on lt.txn_hash = tt.txn_hash
)
```

b. 实现自定义函数接口(explode_nft_trades)，将数据输入并输出得出结果

SQL

```sql
select opensea_line
from source_tmp lateral view explode_nft_trades(line) result_table as opensea_line;
```

c. 全部sql

SQL

```sql
with
log_tmp as
(
    select txn_hash,
           concat_ws('_', collect_set(concat_ws('-', address, topics, data, erc_type, symbol, cast(decimals as string), cast(log_index as string)))) as log_line
    from eth.dwd_eth_log_erctoken
    where txn_hash = '0xdea04ee76bd891ff04b73c48e21bef2550dc9513267876398d5bf77ca519dd17' and dt = '0000-00-09'
    group by txn_hash
),
txn_tmp as
(
   select txn_hash ,
          concat_ws('#', cast(block_height as string), `timestamp`, txn_hash, txn_from, txn_to, cast(txn_chainid as string)) as txn_line
   from eth.dwd_eth_block_transaction
   where txn_hash = '0xdea04ee76bd891ff04b73c48e21bef2550dc9513267876398d5bf77ca519dd17' and dt = '0000-00-09'
),
source_tmp as
(
   select concat_ws('#', tt.txn_line, lt.log_line) as line
   from log_tmp lt join txn_tmp tt on lt.txn_hash = tt.txn_hash
)
select opensea_line
from source_tmp lateral view explode_nft_trades(line) result_table as opensea_line;
```

5. 结论

上述方法拼接的格式是确定的，跟使用spark(flink)算子比较，多了先需要拼接个字段然后再拆开一步，但这是在查找的时候可以完成的，没有进行shuffle，执行时间不会明显增加。而将处理流程全部放到自定义函数中，当我们需求改变时，我们只要改变where条件，就可以得到我们想要的数据，对于开发人员来说，并不需要知道 explode_nft_trades()的实现流程，达到了解耦性开发