袋鼠云
DTSTACK

实时湖仓在袋鼠云的落地实践之路

分享人：修竹

目录 CONTENTS

- 背景介绍
- 实践与探索
- 未来规划

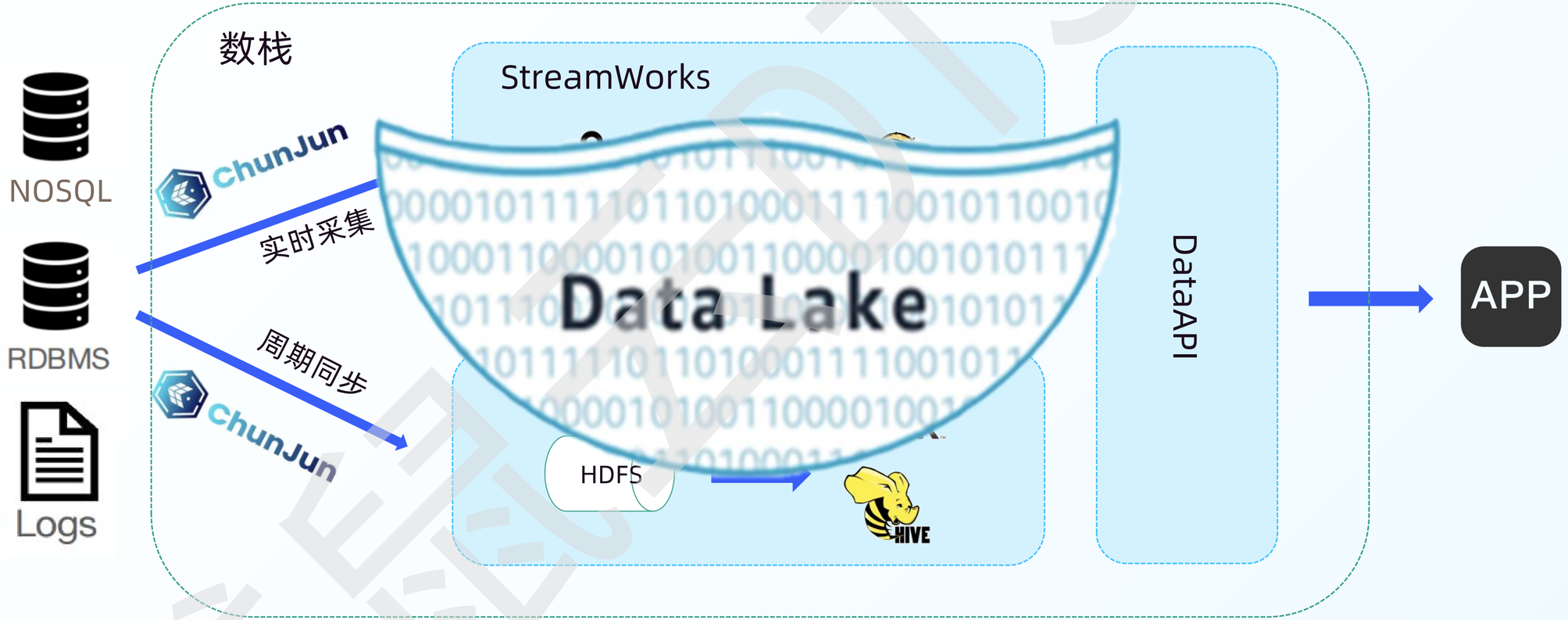
DTSTACK

Part 1 | 背景介绍

痛点分析

未引入数据湖之前数栈提供的是基于lambda架构的开发模式

- 复杂性高，需要维护流批双链路的不同组件
- 实时链路不可查，kafka中间数据查询困难
- 存储成本高，流批两个链路维护两份相同的数据
- 数据口径一致性，不同计算引擎难保证统一的数据口径



为什么会选择数据湖

袋鼠云 DTSTACK

数据湖解读

提供了多样化的分析能力，不限于批处理、流处理、交互式查询和机器学习；

提供了ACID事物能力，可以更好的保障数据质量；

提供了完善的数据管理能力，包括数据格式、数据schema等；

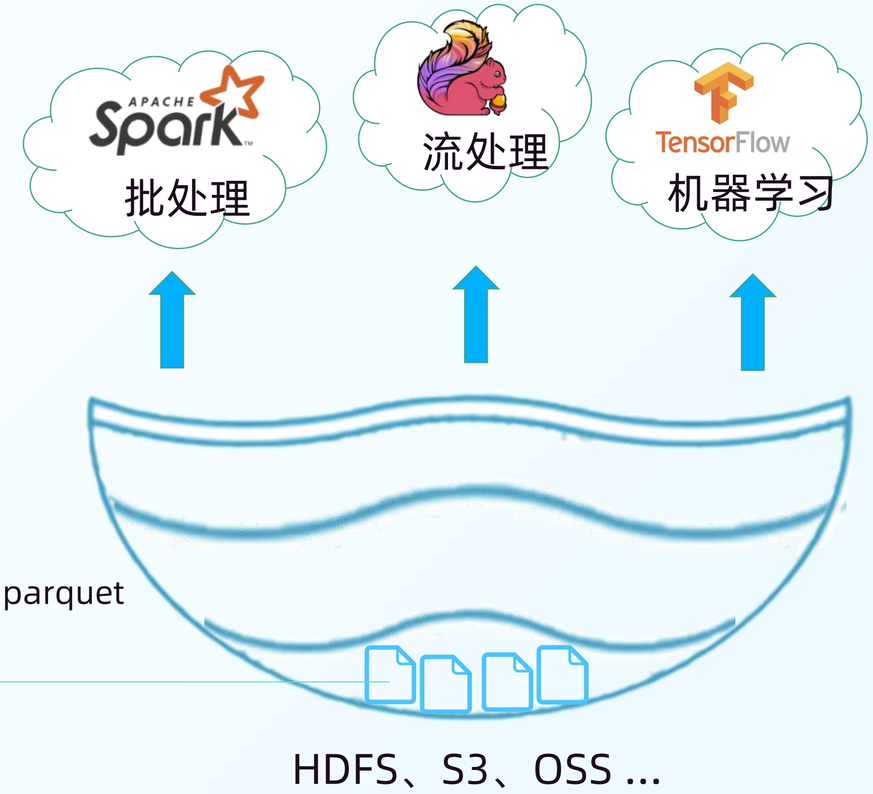
提供了存储介质可扩展的能力，支持HDFS、对象存储等；

价值体现

节省存储成本

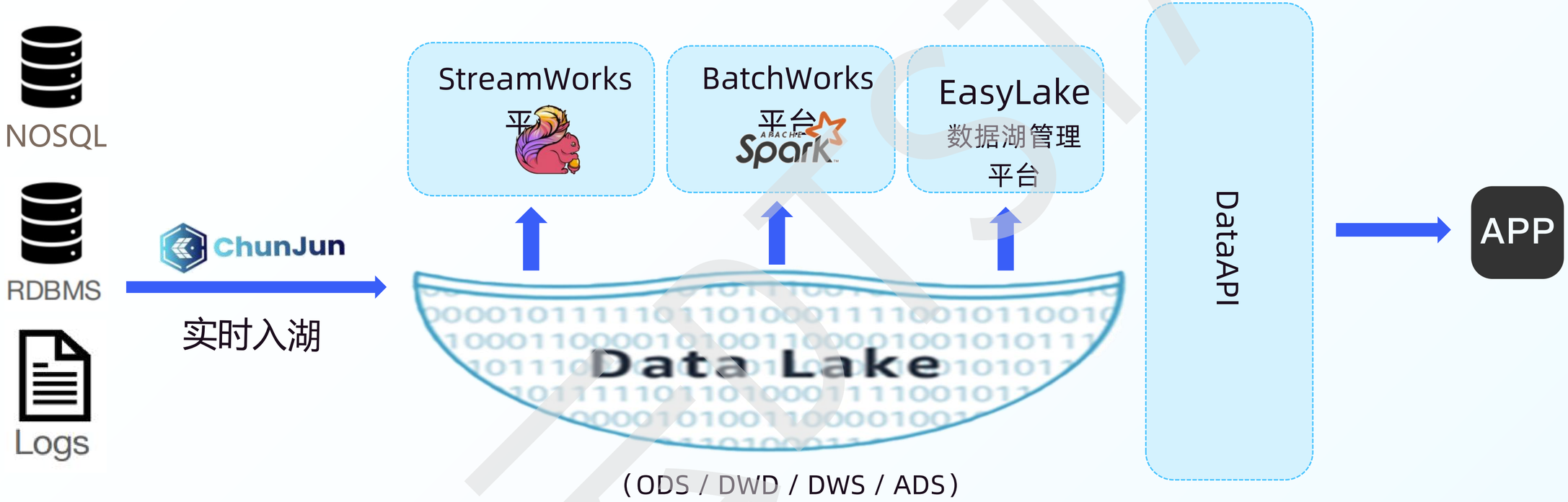
提升开发效率

能够更快更好的挖掘数据价值



DTSTACK

基于数据湖的数栈解决方案

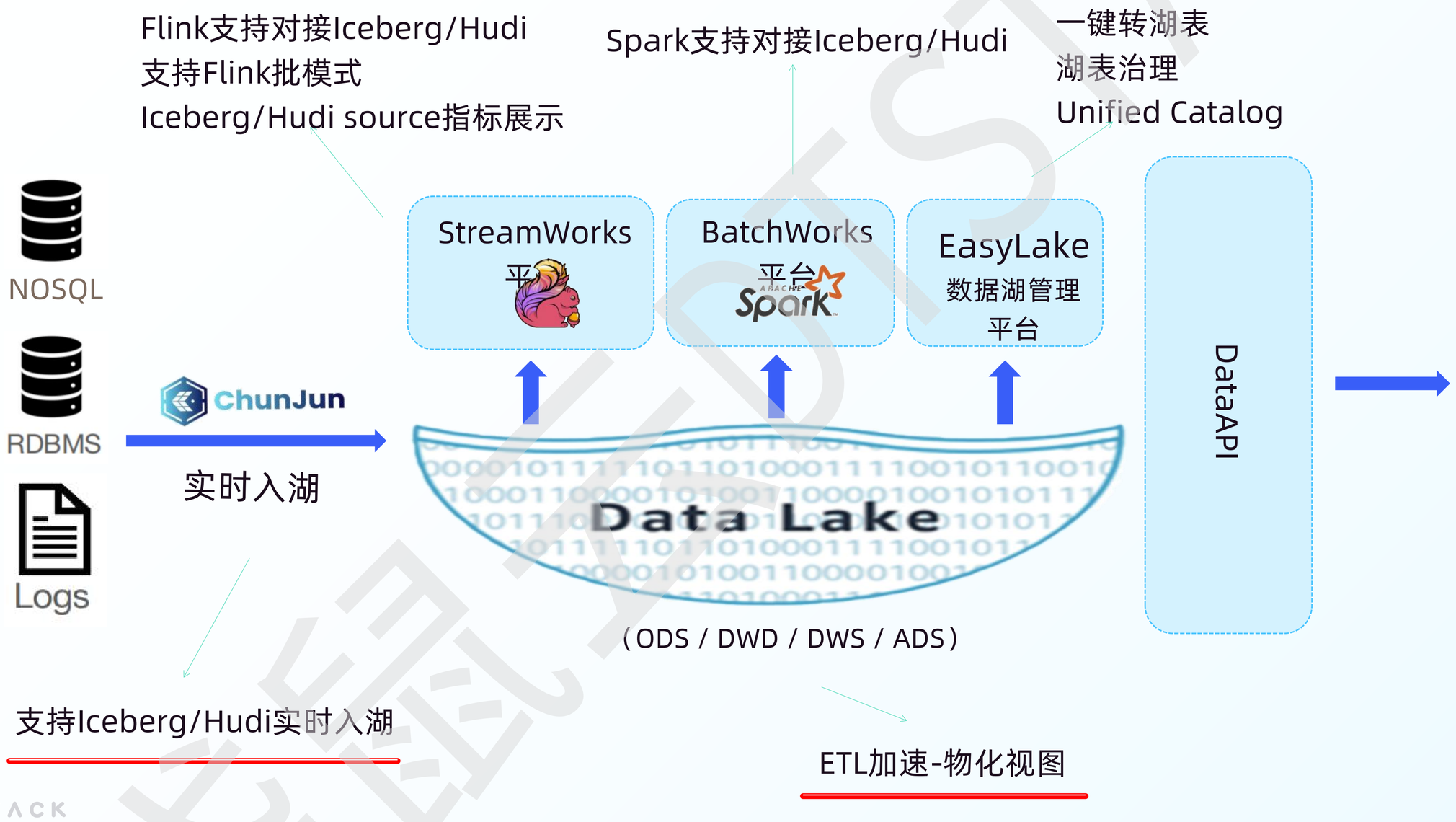


- 存储层流批一体
- 计算层流批一体
- 实时链路中间数据可查
- 统一的数据口径
- 存储低成本

DTSTACK

Part 2 | 实践与探索

数栈基于数据湖的实践


Flink支持对接Iceberg/Hudi
支持Flink批模式
Iceberg/Hudi source指标展示
Spark支持对接Iceberg/Hudi
一键转湖表
湖表治理
Unified Catalog
NOSQL
RDBMS
Logs
ChunJun
实时入湖
StreamWorks
平台
BatchWorks
平台
Spark
EasyLake
数据湖管理
平台
Data Lake
（ODS / DWD / DWS / ADS）
DataAPI
支持Iceberg/Hudi实时入湖
ETL加速-物化视图

CDC实时入湖

袋鼠云 DTSTACK

CDC数据实时入湖

实时性高：CDC数据对实时性要求高，数据新鲜度越高，往往业务价值越高

历史数据量大：数据库的历史数据规模大

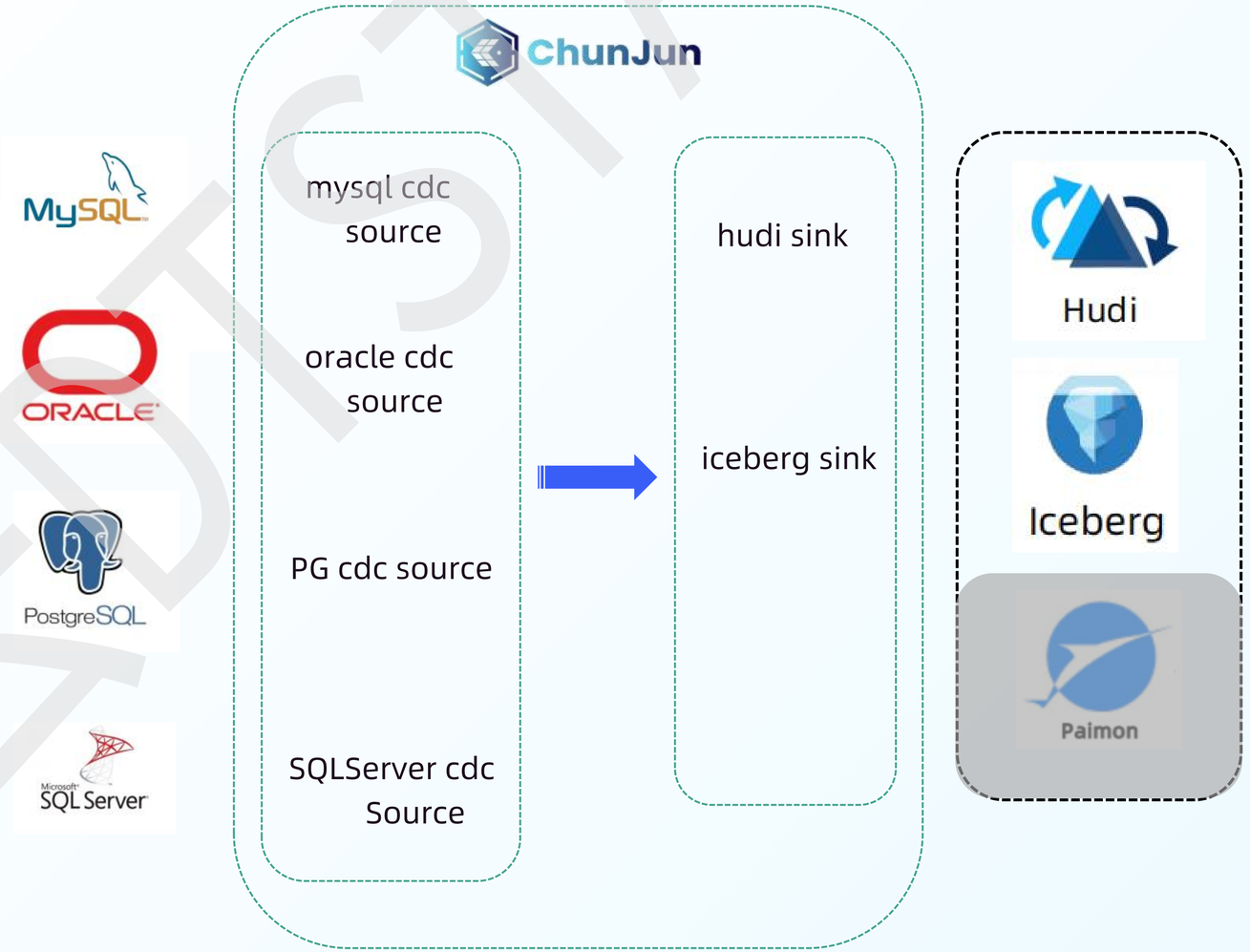
强一致性：数据处理必须要保证有序性而且结果需要一致性

Schema动态演进：数据库对应的Schema会随着业务不断变更

DTSTACK

CDC实时入湖

- 自主可控
- 全增量一体化
- 分钟级时延
- 链路短
- 对业务稳定性无影响



ChunJun：https://github.com/DTStack/chunjun.git

实时入湖落地遇到的问题

袋鼠云 DTSTACK

小文件问题

小文件影响读写效率，HDFS集群稳定性

Hudi适配Flink1.12

客户群体使用的Flink版本大多还停留在1.12

跨集群入湖

存在多套Hadoop集群的场景下存在跨集群的需求

DTSTACK

小文件问题优化-合理设置Checkpoint Interval

袋鼠云 DTSTACK

Checkpoint Interval设置过小会产生一系列问题

- 小文件问题
- 导致hdfs压力变大
- checkpoint失败
- 任务不稳定

Checkpoint Interval设置为1-5min比较合适

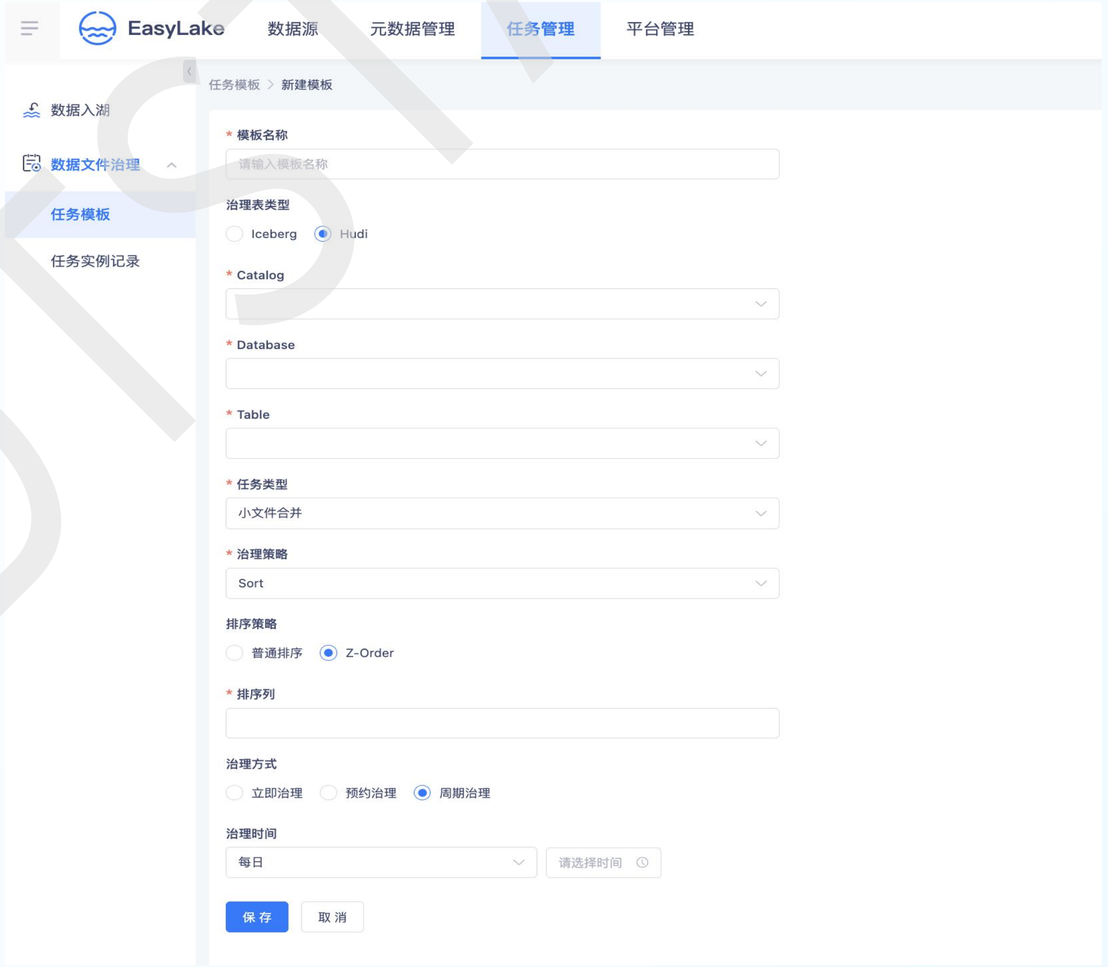
DTSTACK

小文件问题优化-小文件治理

平台化小文件治理，提升治理效率

- 支持数据文件治理
- 支持快照文件治理
- 支持Hudi MOR增量文件合并

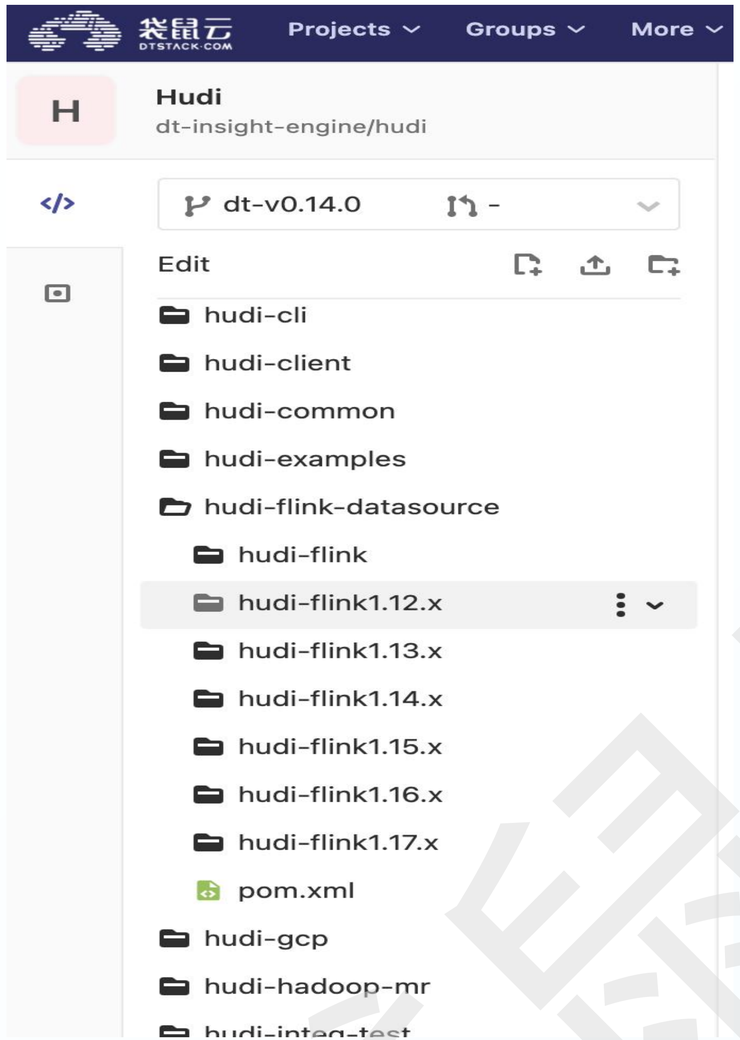
batch compaction　　streming writer



EasyLake平台湖表治理界面

DTSTACK

Hudi适配Flink 1.12版本



适配做法如下：

基于hudi-flink1.13.x模块开发hudi-flink1.12

Flink版本修改成1.12.7

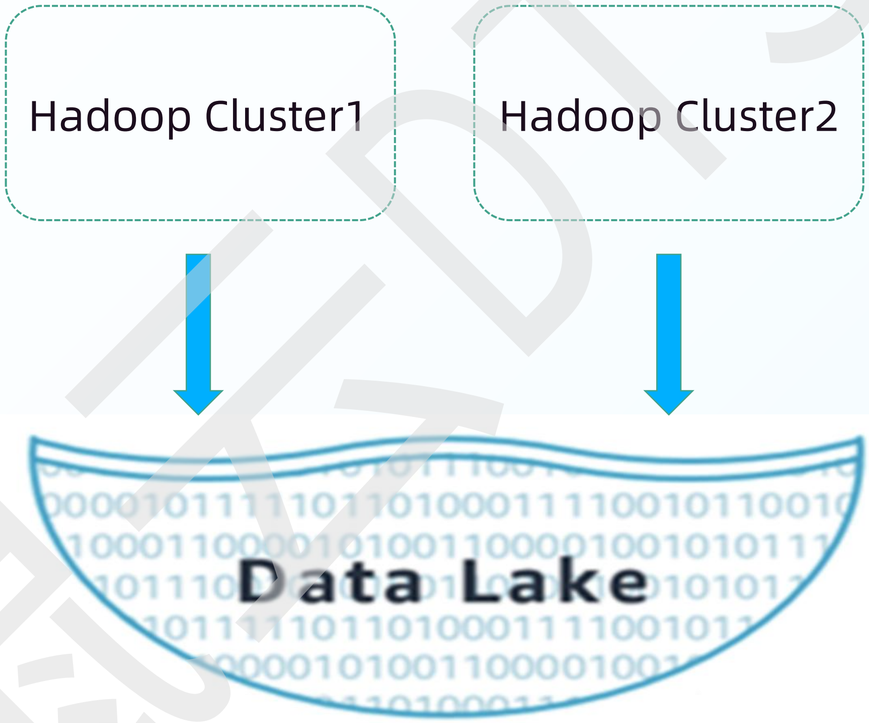
针对不兼容的点逐个进行修复

进行完整的功能测试

DTSTACK

跨集群入湖

背景：多业务部门之间存在多套Hadoop集群，基于这种场景建设数据湖需要具备跨集群入湖的能力



DTSTACK

跨集群入湖方案

袋鼠云 DTSTACK

现状

Hudi和Iceberg Sink默认从HADOOP_CONF_DIR环境变量获取core-site.xml和hdfs-site.xml访问对应的HDFS

改造

在ChunJun iceberg-connector和hudi-connector中对hadoop conf dir的获取方式进行扩展，支持通过指定hadoopConfig自定义参数的方式

```
{
  "parameter": {
    "hadoopConfig": {
      "dfs.ha.namenodes.ns1": "nn1,nn2",
      "dfs.namenode.rpc-address.ns1.nn2": "<nn2.rpc-address>",
      "dfs.client.failover.proxy.provider.ns1": "org.apache.hadoop.hdfs.server.namenode.ha
      "dfs.namenode.rpc-address.ns1.nn1": "<nn1.rpc-address>",
      "dfs.nameservices": "ns1",
      "fs.defaultFS": "hdfs://ns1"
    },
    "tablePath": "${hdfs_table_path}",
    "tableName": "${hudi_table_name}",
    "preCombineField": "ts",
    "recordKeyField": "uuid",
    "partitionPathField": "par",
    "operation": "upsert"
  },
  "name": "hudiwriter"
}
```

收益

能够使集群之间数据流动起来，打破数据孤岛

ETL加速探索-物化视图

为什么需要物化视图？

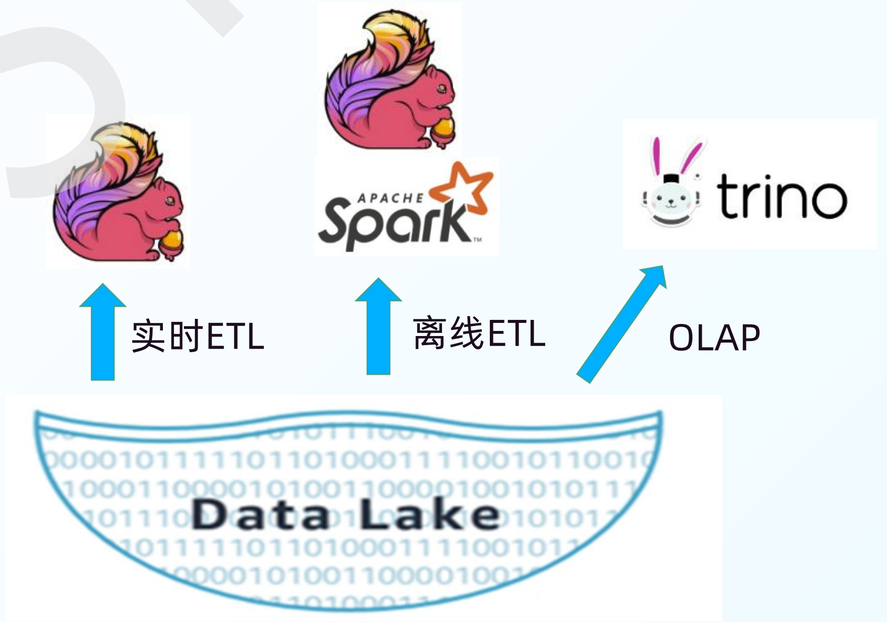
在实时数据湖中包含三类任务：

- 实时ETL
- 离线ETL
- OLAP

以上三类任务在加工过程中ODS -> ADS

- 聚合操作越来越多
- IO越来越密集
- 多个任务SQL中具有相同逻辑的SQL片段

实现物化视图可以为实时加工链路加速，并能节省计算成本



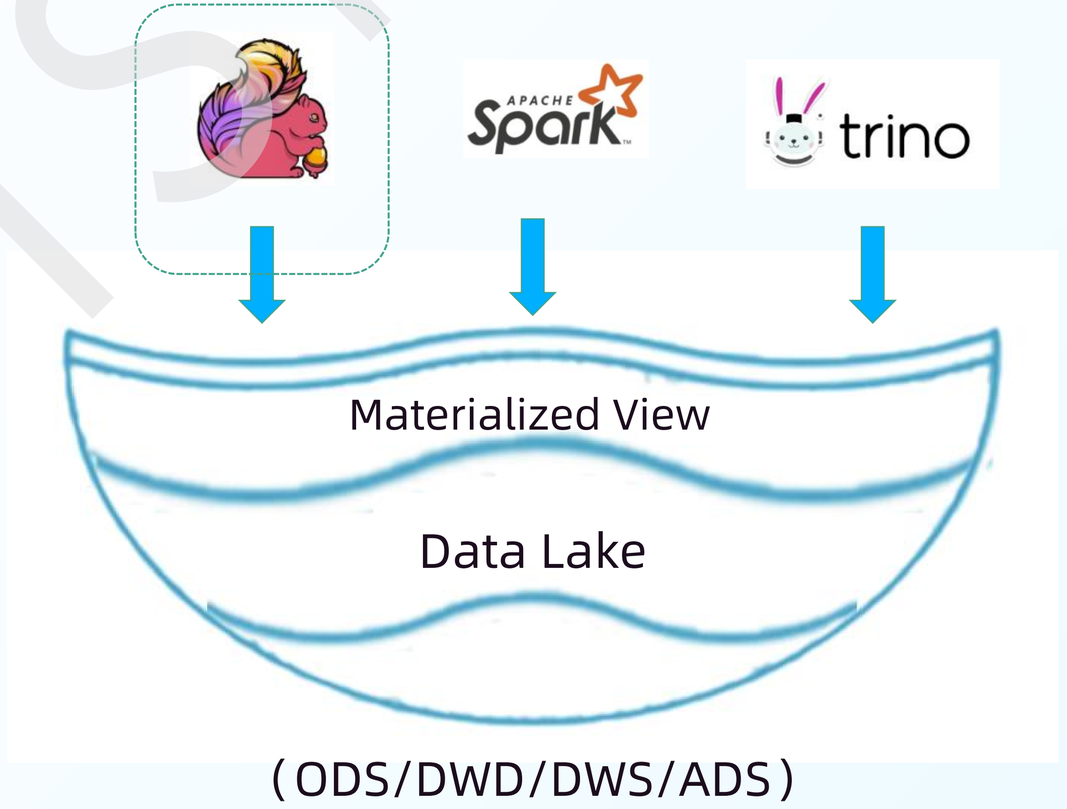
（ODS/DWD/DWS/ADS）

ETL加速探索-物化视图

在实时数据湖中基于数据湖构建的物化视图可实现流、批和OLAP任务之间共享，从而进一步降低实时数据湖中数据在整条链路中的延时。

实时数据湖中落地物化视图需要完成：

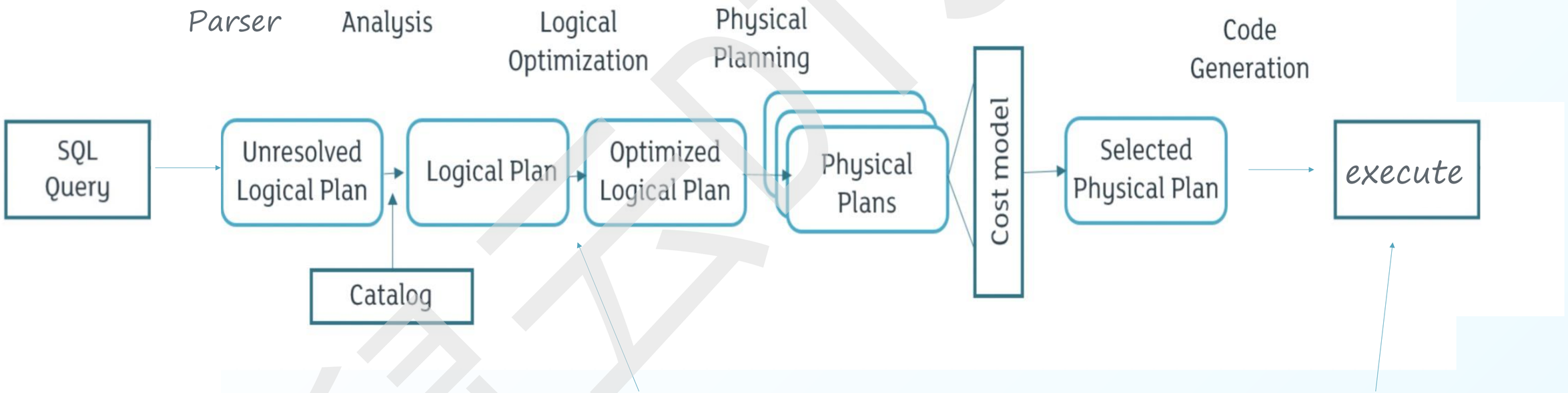
- 平台化数据湖物化视图管理
- Spark支持基于数据湖表格式管理物化视图
- Trino支持基于数据湖表格式管理物化视图
- Flink支持基于数据湖表格式管理物化视图



ETL加速探索-物化视图

袋鼠云 DTSTACK

物化视图实现原理



匹配物化视图
逻辑执行计划重写

失败回退

Part 3 | 未来规划

未来规划

- 易用性：增加平台湖表管理的易用性
- 引入Paimon：平台支持对接Paimon、增加基于Paimon的湖仓一体建设
- 提升入湖性能：深入并增强内核，提升入湖的的性能
- 安全性探索：数据湖提供数据共享、支持多引擎，探索数据湖的安全管理方案

DTSTACK

袋鼠云
DTSTACK

让数据产生价值

袋鼠云服务号

行业交流群

资料获取