

字节跳动数据集成引擎BitSail 开源架构演进和实践

李畅 字节跳动大数据工程师

DataFunSummit # 2023

个人简介

- 16年加入字节跳动开发套件团队，从0到1设计、研发了面向字节各业务线的数据集成服务
- 专注大规模数据的分布式计算和传输领域，提供高效、可靠的数据集成解决方案
- 开源社区爱好者，BitSail开源项目负责人之一

目录 CONTENT

01 BitSail 背景介绍

03 BitSail CDC解决方案

02 BitSail 新功能介绍

04 未来展望

01

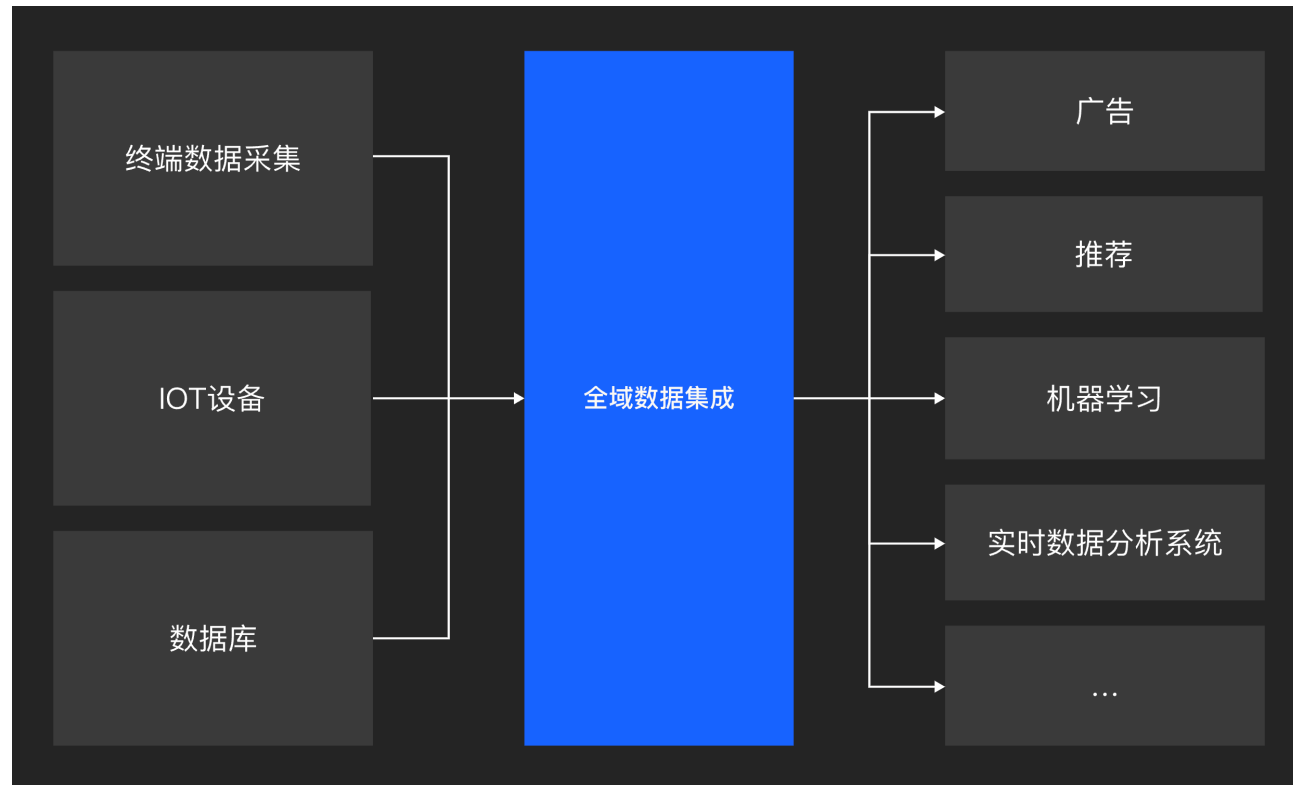
BitSail 背景介绍

DataFunSummit # 2023

BitSail 背景介绍

在数据建设过程中，通常需要把数据从A系统导入到B系统

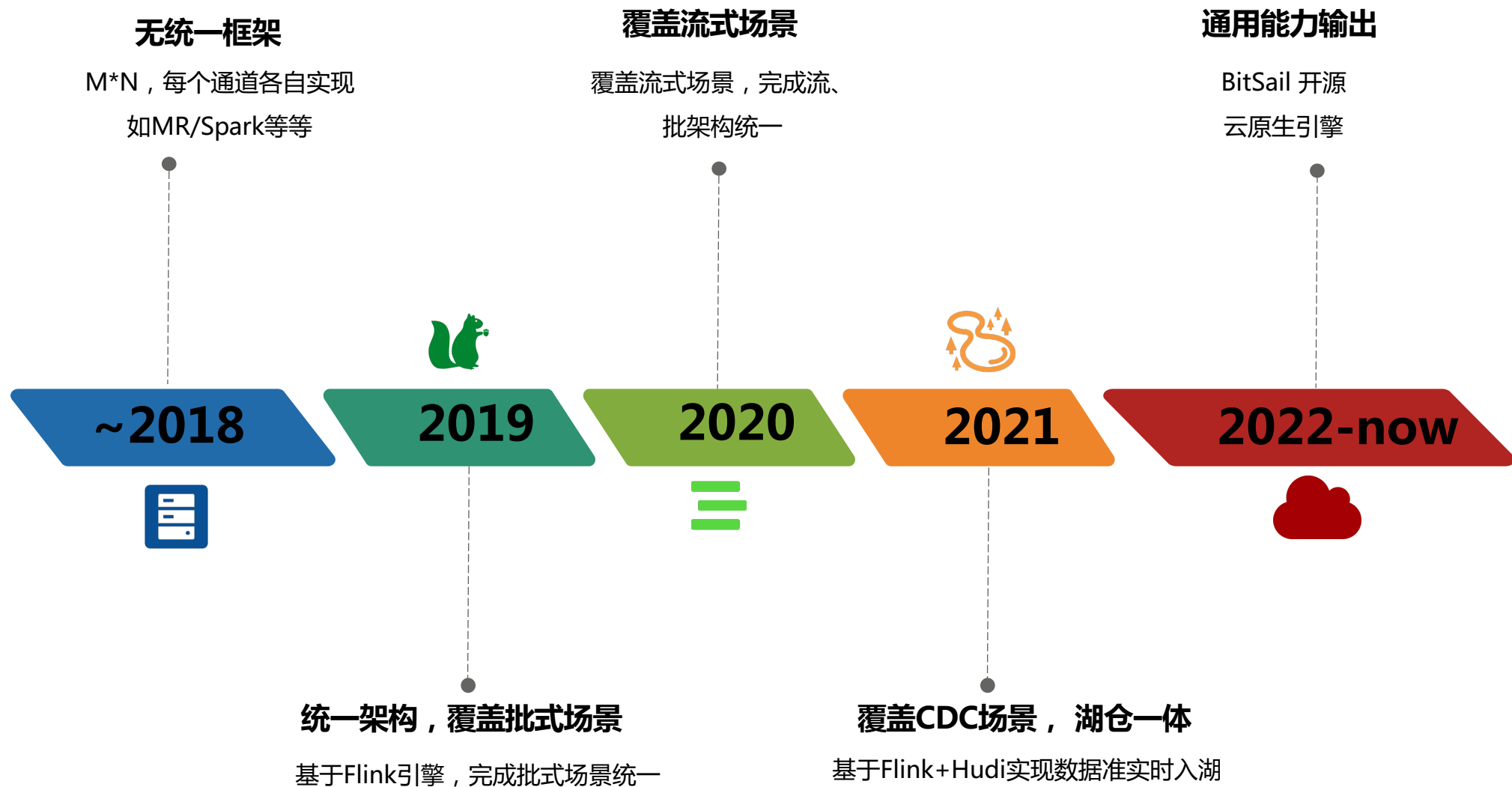
数据集成是数据建设的基础，主要解决异构数据源间数据传输、加工和处理



BitSail 背景介绍

- **BitSail , 又可称为 “比特航行”**
- **BitSail 是字节跳动开源的分布式、高性能数据集成引擎 , 支持多种异构数据源间的数据同步 , 并提供离线、实时、增量场景下全域数据集成解决方案**
- **BitSail 目前已被广泛使用, 并服务于字节内部几乎所有业务线 , 包括抖音、今日头条等 , 每天同步数百万亿条数据 , 同时在火山引擎云原生环境、本地私有云环境等多种场景下得到验证**

BitSail 演进历程



02

BitSail 新功能介绍

DataFunSummit # 2023

BitSail 数据同步架构

架构

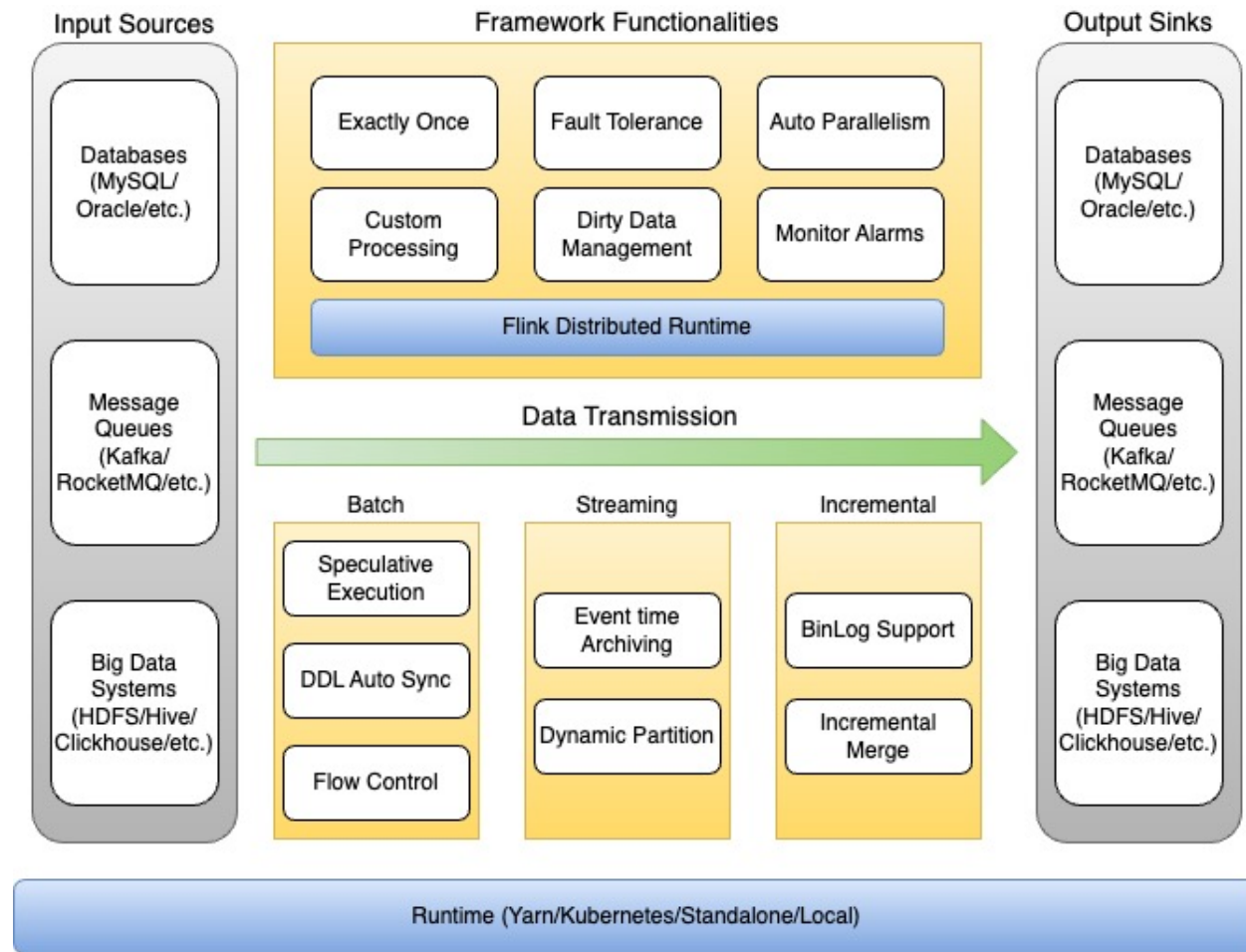
- Connector层、框架层、引擎层
- 每一层都采取可插拔的设计，灵活性较好

框架层

- 丰富基础能力，比如数据源类型转换、脏数据处理、流控、自动并发度推断、运行监控等

场景

- 覆盖离线、实时、增量数据同步场景



BitSail 数据同步架构 – 代码结构

Root

```
bitsail [bitsail-all] ~/Git/bitsail
├── .github
├── .idea
├── bitsail-base
├── bitsail-clients
├── bitsail-common
├── bitsail-components
├── bitsail-connectors
├── bitsail-cores
├── bitsail-dist
├── bitsail-shade
├── bitsail-test
├── output
├── target
├── tools
└── website
```

Connector

```
bitsail-connectors
├── bitsail-connectors-legacy
├── connector-assert
├── connector-base
├── connector-cdc
├── connector-clickhouse
├── connector-doris
├── connector-druid
├── connector-elasticsearch
├── connector-fake
├── connector-ftp
├── connector-hbase
├── connector-kafka
├── connector-kudu
├── connector-larksheet
├── connector-localfilesystem
├── connector-mongodb
├── connector-print
├── connector-redis
├── connector-rocketmq
└── connector-selectdb
```

Components

```
bitsail-components
├── bitsail-component-clients
├── bitsail-component-formats
├── bitsail-component-formats-flink
├── bitsail-component-metrics
├── bitsail-component-progress
├── bitsail-component-security
├── bitsail-conversion-flink
├── bitsail-flink-row-parser
├── target
├── .flattened-pom.xml
└── pom.xml
```

Engine Adapter

```
bitsail-cores
├── bitsail-core-api
├── bitsail-core-entry
├── bitsail-core-flink
├── bitsail-core-flink-1.11-bridge
├── bitsail-core-flink-1.16-bridge
└── bitsail-core-flink-base
```

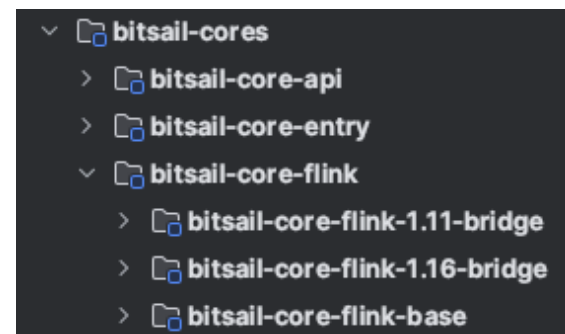
多引擎架构

问题

- Flink 深度绑定，依赖较重，场景受限制
- Flink 定位通用计算引擎，数据集成场景存在资源浪费

解决思路

- 引擎基于可插拔设计，支持轻量化的分布式计算引擎
- Connector 提供引擎无关的读写接口
- 框架层与引擎解耦



多引擎架构 – Source API

Source API

数据读取组件的生命周期管理类，主要负责和框架的交互，它不参与作业真正的执行

- Split

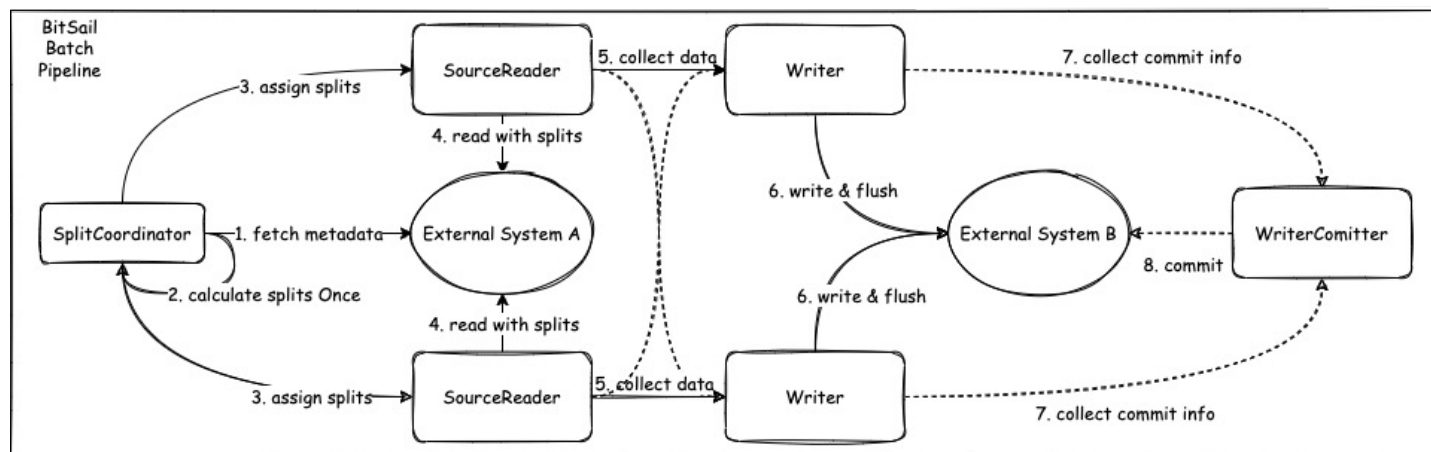
数据分片，大数据处理框架的核心目的就是大规模的数据拆分成多个合理的Split

- Split Coordinator

创建、管理Split，并将Split分发给Source Reader

- Source Reader

真正负责数据读取的组件，在接收到Split后会对其进行数据读取，然后将数据传输给下一个算子



多引擎架构 – Sink API

Sink API

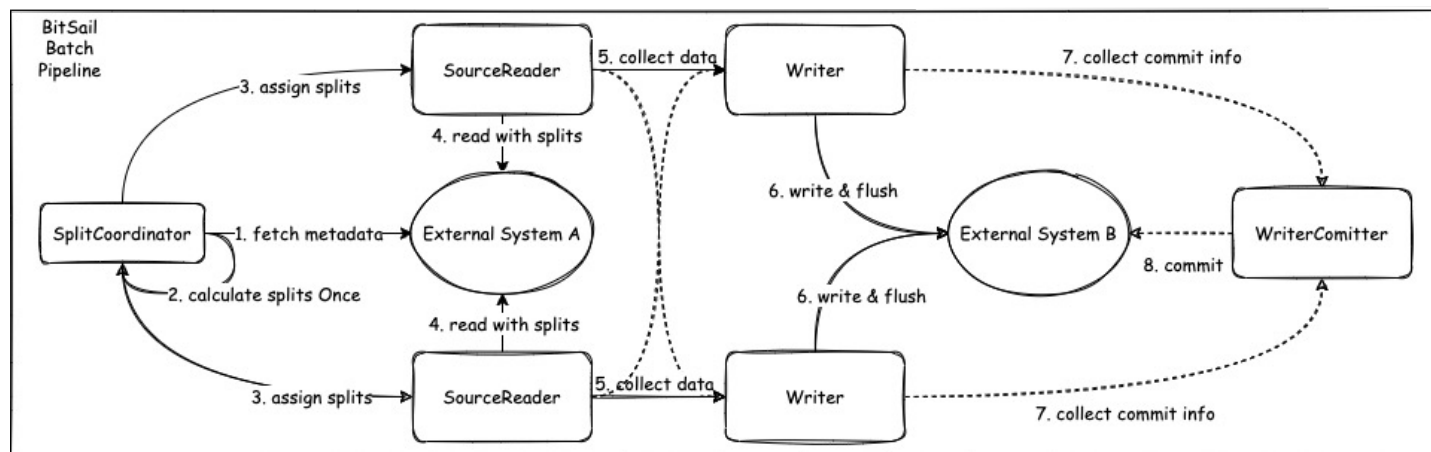
数据写入组件生命周期的管理类，主要负责和框架进行交互，它不参与作业真正的执行

- Writer

主要负责将接收到的数据写到外部存储

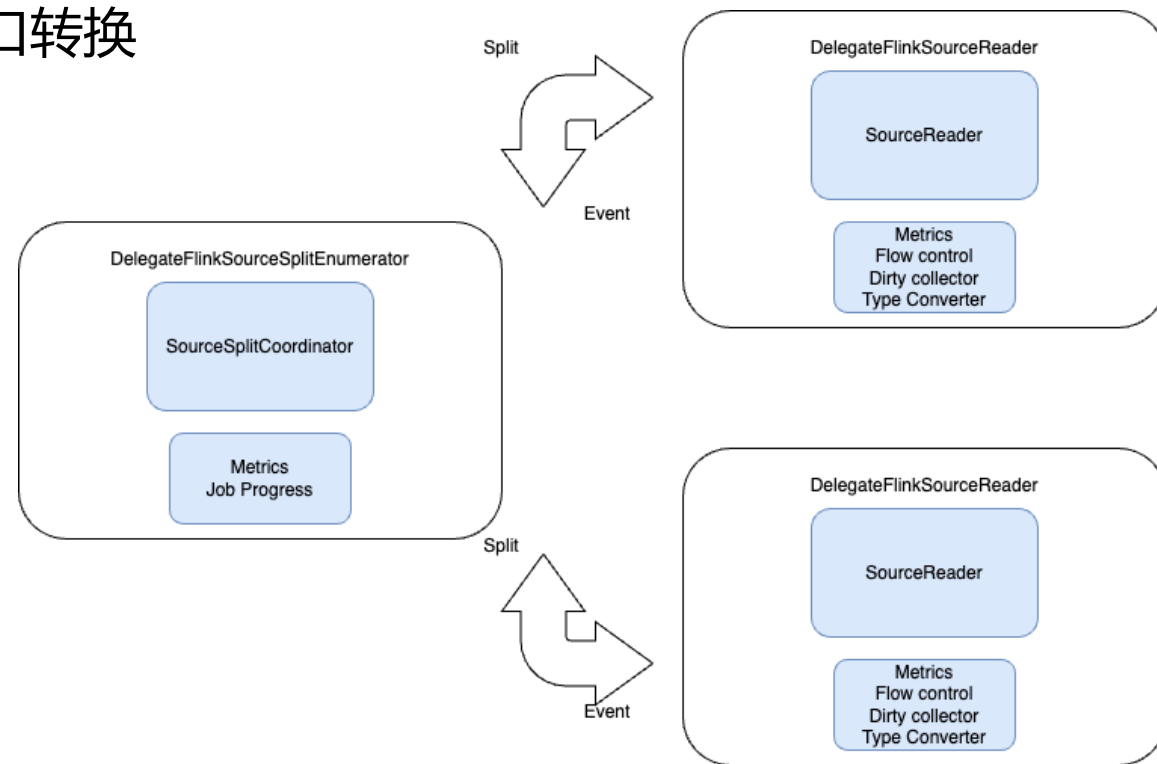
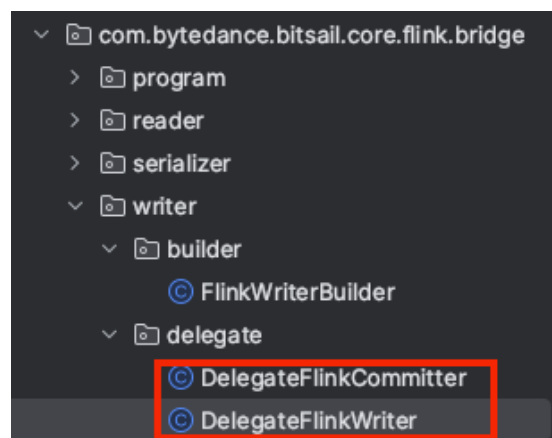
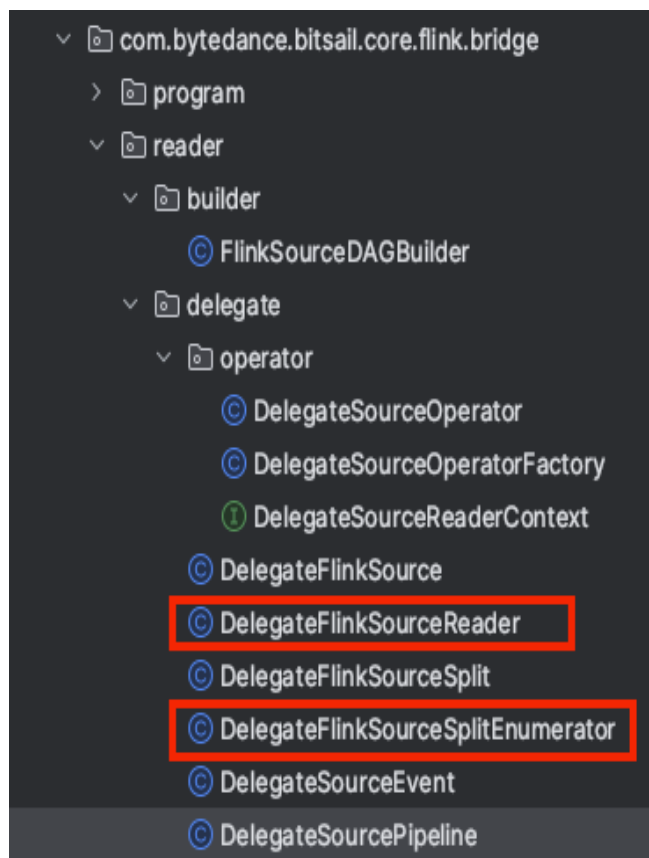
- Writer Committer (可选)

用于对数据进行提交操作，基于两阶段提交，实现Exactly-Once的语义



多引擎架构 – API 接口转换

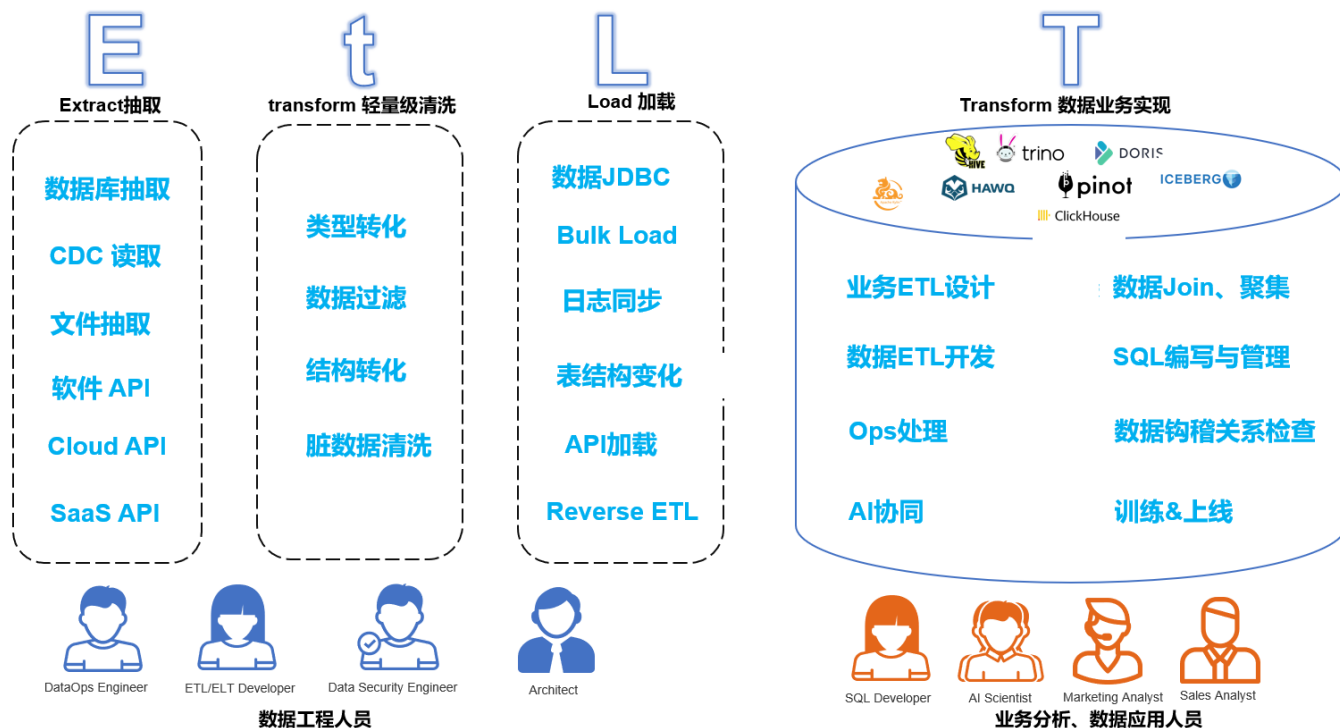
- DelegateXXX 基于引擎接口实现
- DelegateXXX 实现Source/Sink API 接口和引擎接口转换



EtLT架构

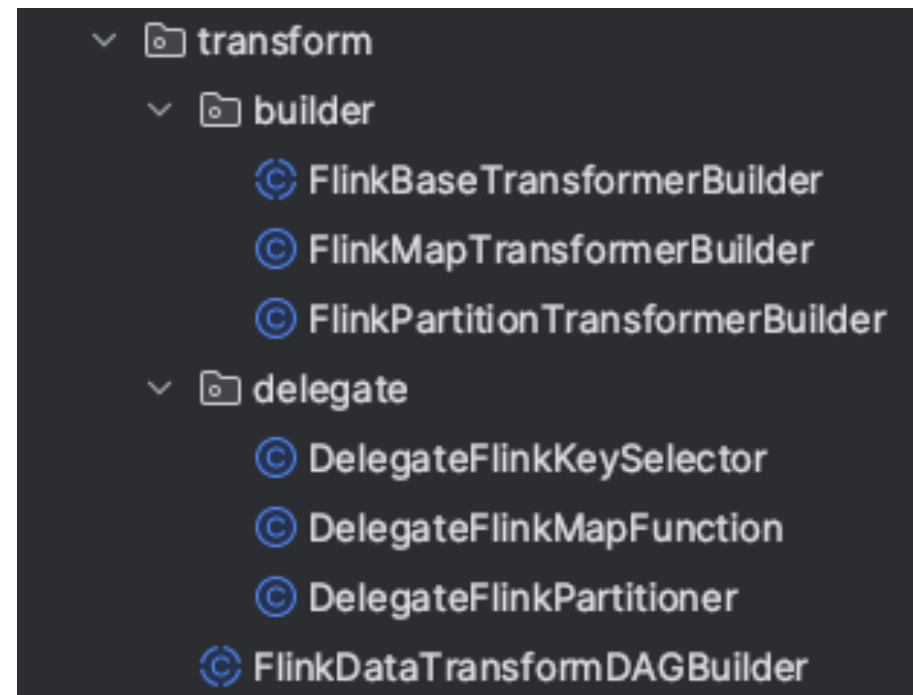
数据处理架构

- ETL -> ELT -> EtLT
- 现代数据栈、实时性



BitSail EtLT架构

- Source、Sink间引入Transform模块
- 流批一体，支持轻量级数据处理



自动化测试引擎

背景

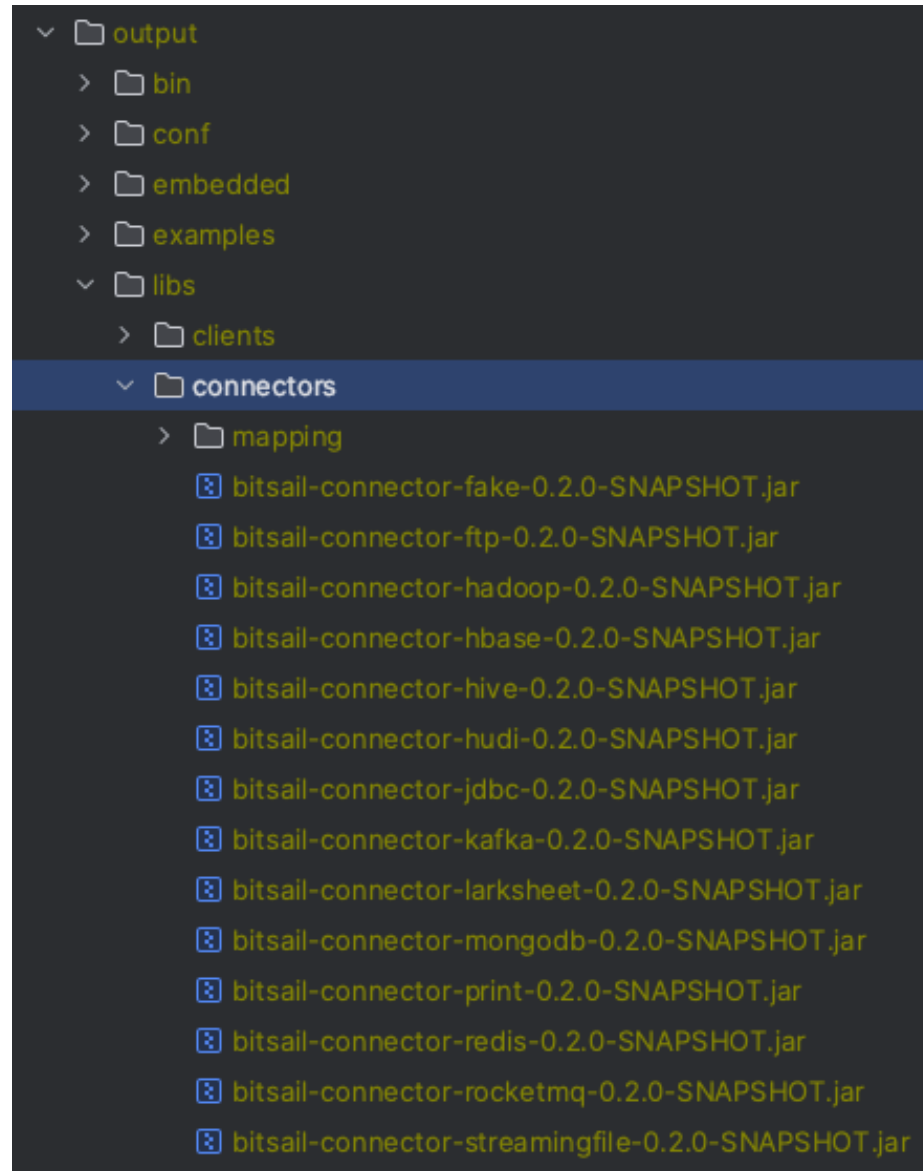
- Fat 包缺点，产物包太大，并且包冲突较为严重
- Connector 独立打包，并支持运行时动态加载

问题

- Connector 动态加载时，存在潜在的包冲突问题

解决思路

- 构建 Connector 自由组合自动化测试引擎
- 打通 CICD 流程，避免 Connector 包冲突问题



自动化测试引擎

基本原理

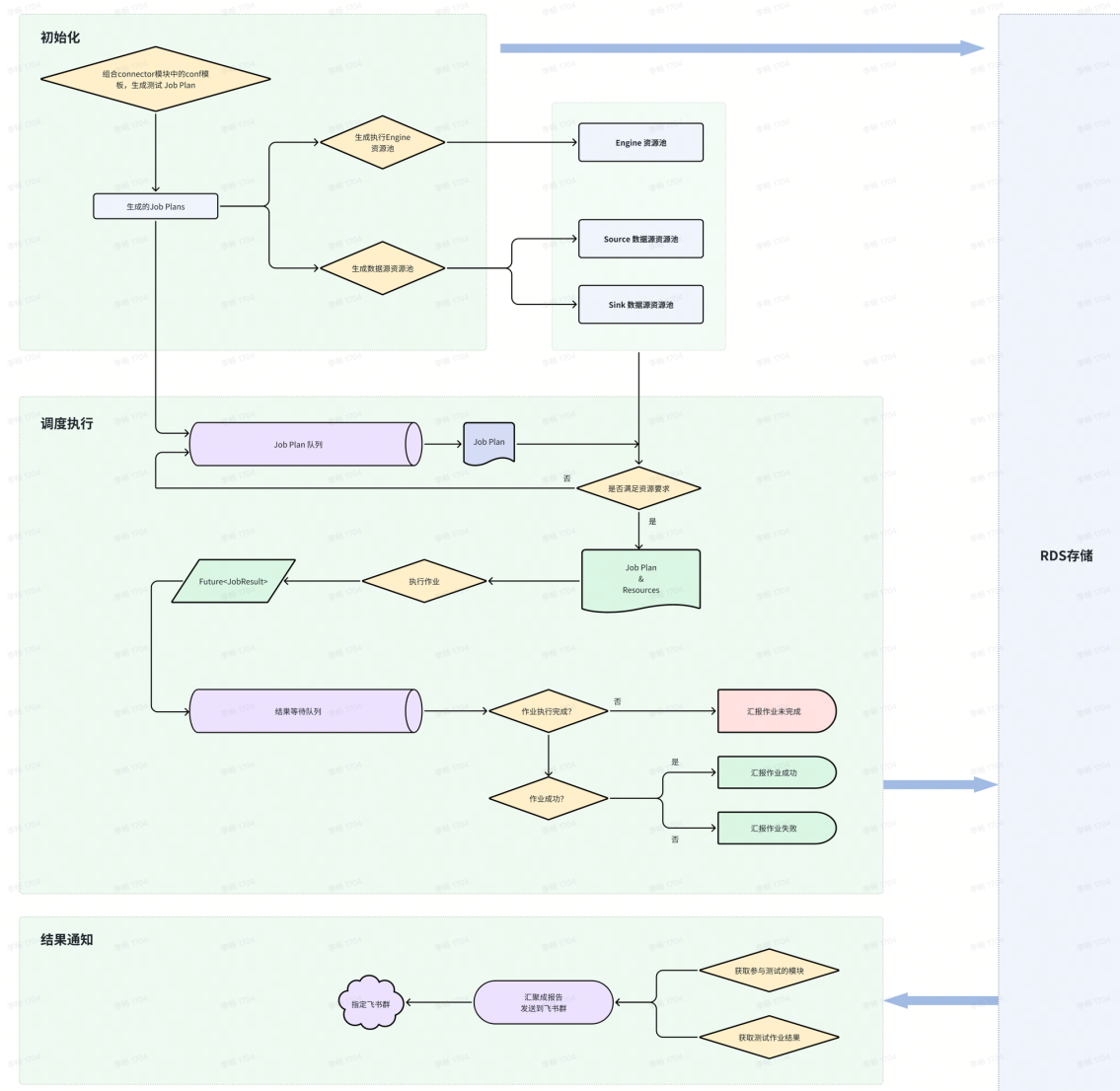
- 单数据源测试用例构建
- $M * N$ 自由组合测试任务生成
- 分布式调度和执行
- 结果通知

多引擎架构

- 测试用例与引擎无关
- 测试任务可复用到不同引擎

流批一体框架

- 兼容离线以及实时场景



03

BitSail CDC解决方案

DataFunSummit # 2023

CDC 背景介绍

CDC (Change data capture) :

通过捕获数据变更日志 (Binlog) 的方式，将更新数据同步到外部数据源

相比于批式变更同步，CDC 变更同步有着延迟更低，对线上数据影响更小的优点



CDC 同步使用场景

- 离线数据分析：离线数仓建设，离线报表

MySQL -> Hive/ClickHouse

- 准实时数据分析：实时看板建设，实时数据分析

MySQL -> Doris/StarRocks

- 搜索数据分析：在线搜索

MySQL -> ElasticSearch

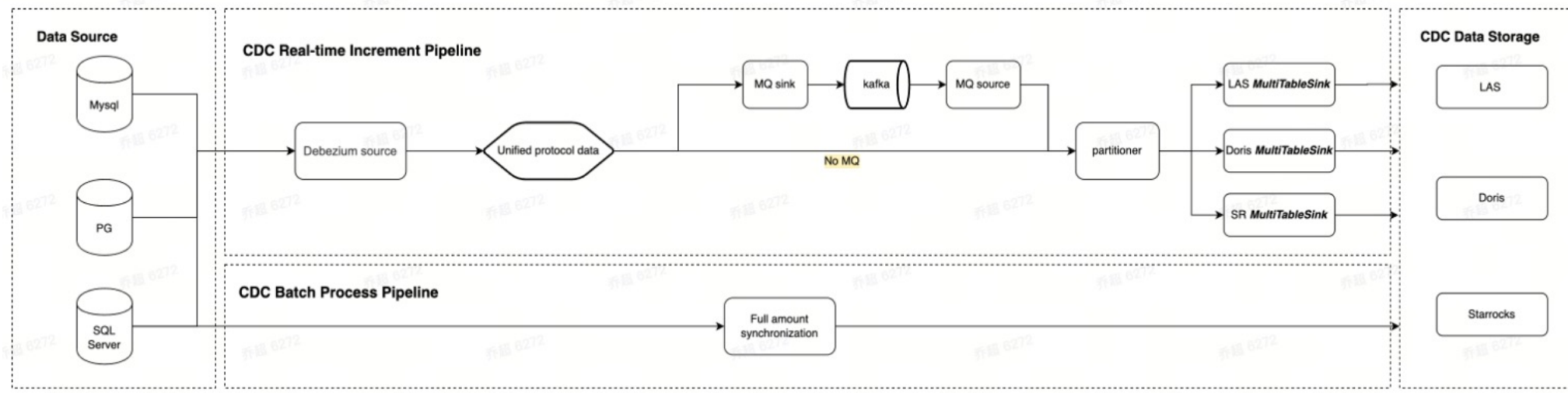
离线整库同步解决方案

痛点

- 延迟高：MySQL 数据 $T + 1$ 导入 Hive，实时性较差
- 运维成本高：需要维护多张表与表之间的映射关系和多个同步任务
- 分库分表场景：分库分表只能分别入仓，入仓后再进行合并
- 一致性问题：同步链路长，乱序导致数据不一致问题



CDC 整库同步解决方案



CDC 整库同步解决方案

离线整库同步 - 延迟高

- MySQL 数据 $T + 1$ 导入 Hive , 实时性较差

CDC 整库同步

- 全量数据批式导入
- 增量数据实时消费 Binlog 写入 , 根据场景需求调整写入延迟
- 以 Doris/StarRocks 为例延迟可以做到秒级

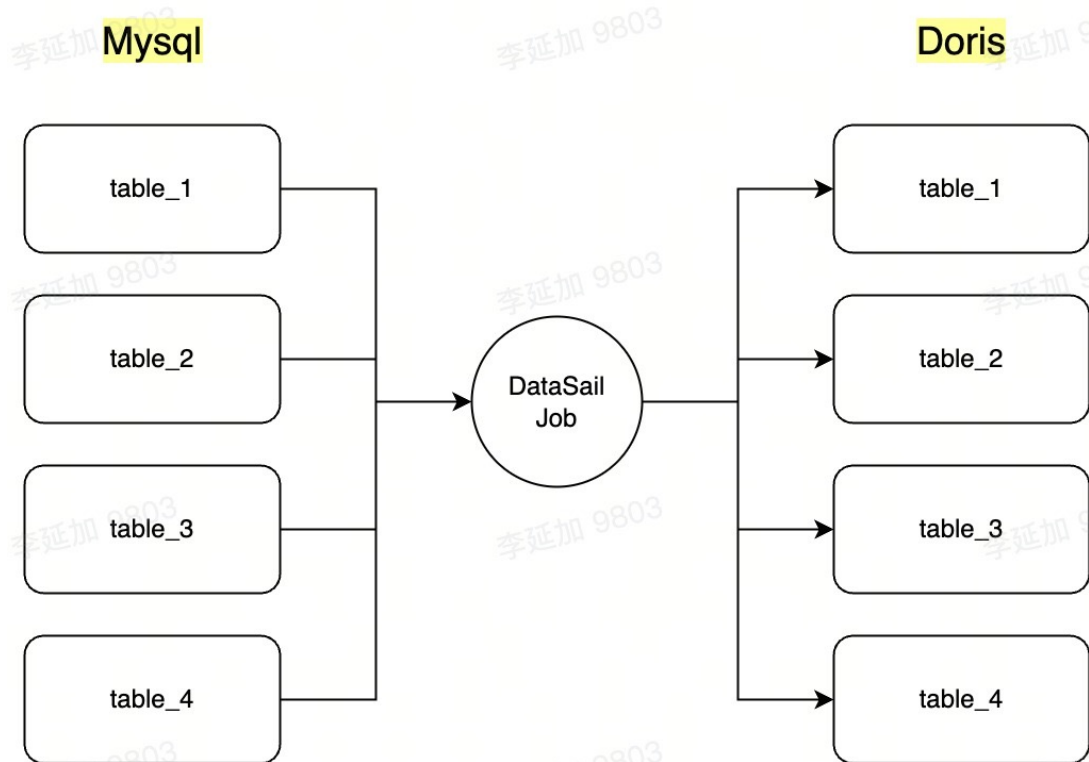
CDC 整库同步解决方案

离线整库同步 - 运维成本高

- 需要维护多张表之间的映射关系和多个同步任务

CDC 整库同步

- 自动建表：自动根据上游表的字段创建下游表
- 全量任务：自动调度单次全量批任务，其中每个批任务实现单表同步
- 增量任务：一个任务写入到多张下游表中



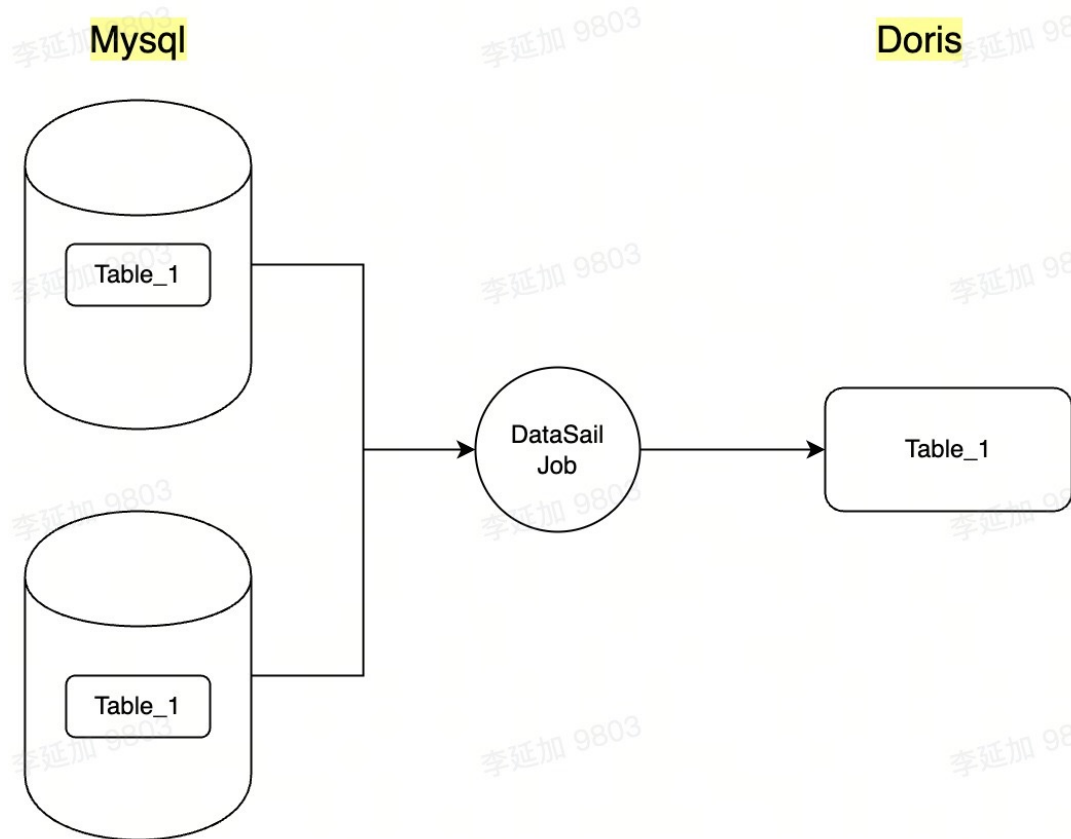
CDC 整库同步解决方案

离线整库同步 - 分库分表

- 分库分表只能分别入仓，入仓后再进行合并

CDC 整库同步

- 全量任务：单个任务读取多个数据源，写入到单张下游表中
- 增量任务：增量任务直连多个数据源拉取Binlog，写入到单张下游表中



CDC 整库同步解决方案

离线整库同步 - 一致性问题

- 同步链路长，乱序导致数据不一致

CDC 整库同步

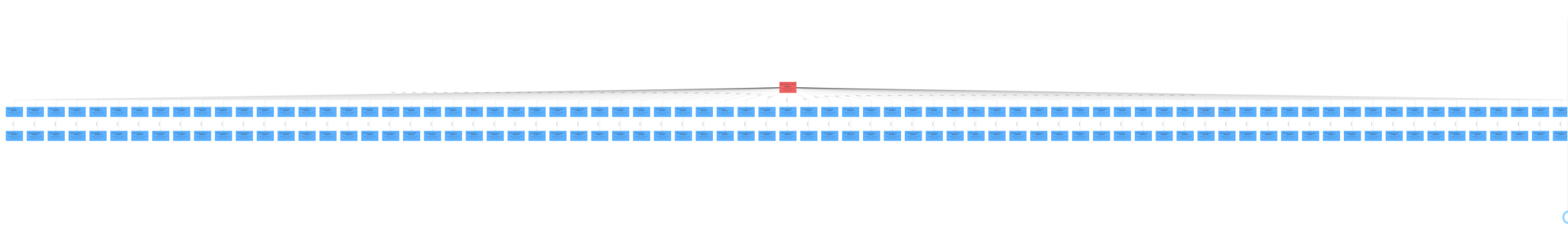
- 基于严格有序的字段，保证新数据不被旧数据覆盖
 - Binlog 原生字段，例如 Binlog Position
 - 业务自定义排序字段，例如 Hudi 的 precombine 字段，Doris 的 sequence 字段





CDC 整库同步解决方案 – 运行流程

▼ 执行步骤

步骤	子任务名称	启动时间	结束时间	耗时	操作状态	运维
1	增量流任务停止	2023-08-28 12:26:55	2023-08-28 12:27:05	10s	● 已完成	执行详情
2	目标表创建	2023-08-28 12:27:05	2023-08-28 12:27:10	5s	● 已跳过	执行详情
3	任务目录创建	2023-08-28 12:27:10	2023-08-28 12:27:16	6s	● 已跳过	执行详情
4	增量流任务位点初始化	2023-08-28 12:27:16	2023-08-28 12:27:20	4s	● 已完成	执行详情
5	一次性全量批任务清理	2023-08-28 12:27:20	2023-08-28 12:27:25	5s	● 已跳过	执行详情
6	一次性全量批任务创建	2023-08-28 12:27:25	2023-08-28 12:27:35	10s	● 已完成	执行详情
7	一次性全量批任务发布	2023-08-28 12:27:35	2023-08-28 12:27:44	9s	● 已完成	执行详情
8	任务quota检测	2023-08-28 12:29:32	2023-08-28 12:29:32	0s	● 已确认	执行详情
9	一次性全量批任务启动	2023-08-28 12:29:32	2023-08-28 17:09:46	4h 40m 14s	● 已完成	执行详情
10	增量流任务创建	2023-08-28 17:09:46	2023-08-28 17:09:51	5s	● 已完成	执行详情
11	增量流任务发布	2023-08-28 17:09:51	2023-08-28 17:09:57	6s	● 已完成	执行详情
12	增量流任务启动	2023-08-28 17:09:57	2023-08-28 17:14:32	4m 35s	● 已完成	执行详情

CDC 整库同步解决方案 - 运行页面



Name	Status	Bytes Received	Records Received	Bytes Sent	Records Sent	Parallelism	Start Time	Duration	Tasks
	RUNNING	48.0 GB	145,590,505	0 B	0	1	2023-09-04 20:22:39.052	21h 33m 51s 647ms	1
	RUNNING	40.0 GB	25,008,994	0 B	0	1	2023-09-04 20:22:39.035	21h 33m 51s 664ms	1
	RUNNING	11.3 GB	7,085,486	0 B	0	1	2023-09-04 20:22:39.147	21h 33m 51s 552ms	1
	RUNNING	7.62 GB	5,739,470	0 B	0	1	2023-09-04 20:22:39.180	21h 33m 51s 519ms	1

04

未来展望

DataFunSummit # 2023



火山引擎 数智平台

 DataFun.

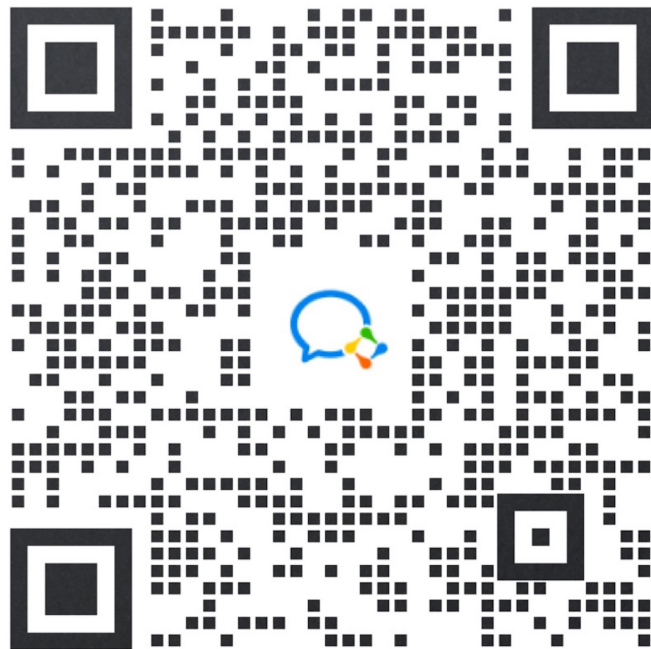
基础能力建设

- Connector生态建设，提供更多的读写数据源
- 多引擎架构，提供更轻量的分布式计算引擎

CDC 同步能力增强

- 自动 DDL 同步，支持自动 Create Table 和 Add Column
- CDC 生态建设，支持更多的上下游数据源，复用 CDC 框架能力
- 端到端数据一致性校验能力构建

了解更多



BitSail技术交流群
最新进展·社区活动·干货分享



BitSail代码仓库
<https://github.com/bytedance/bitsail>



感谢观看



火山引擎数智平台

 DataFun.