袋鼠云
DTSTACK

# 企业为什么需要实时湖仓

主讲人：灵江

DTSTACK

# 目录 CONTENTS


- 当前的企业困境
- 三步教你如何使用“Flink+数据湖”构建实时湖仓
- 为什么企业需要实时湖仓

# 统一结构化/半结构化数据的流批计算

袋鼠云 DTSTACK

当谈到数据湖的时候，大家都在说，可以把所有数据（结构化/半结构化/非结构化）一股脑都丢进去，进行统一的元数据管理，然后上层计算对接，进行流批计算/OLAP分析/算法分析。

这个没问题，数据湖确实能承接底层的这部分能力。

不过，这次我们只关注讨论：
**利用湖仓架构，统一结构化/半结构化数据的流批计算。**

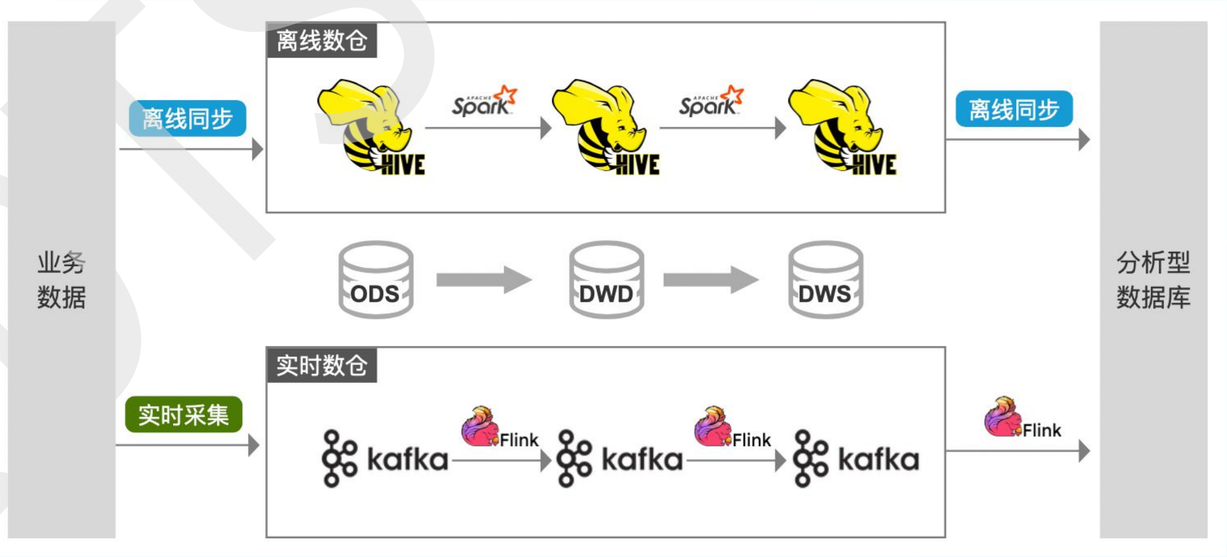
非结构化的视频/图片/文本等数据的存储和计算不在讨论范围内。

# 经典的 Lambda 架构

右图是一个经典的 Lambda 架构，这套架构的优点很明显：技术方案成熟、应用实践广泛。适用于企业发展过程中各阶段、各场景下的大数据开发。

**但是，**

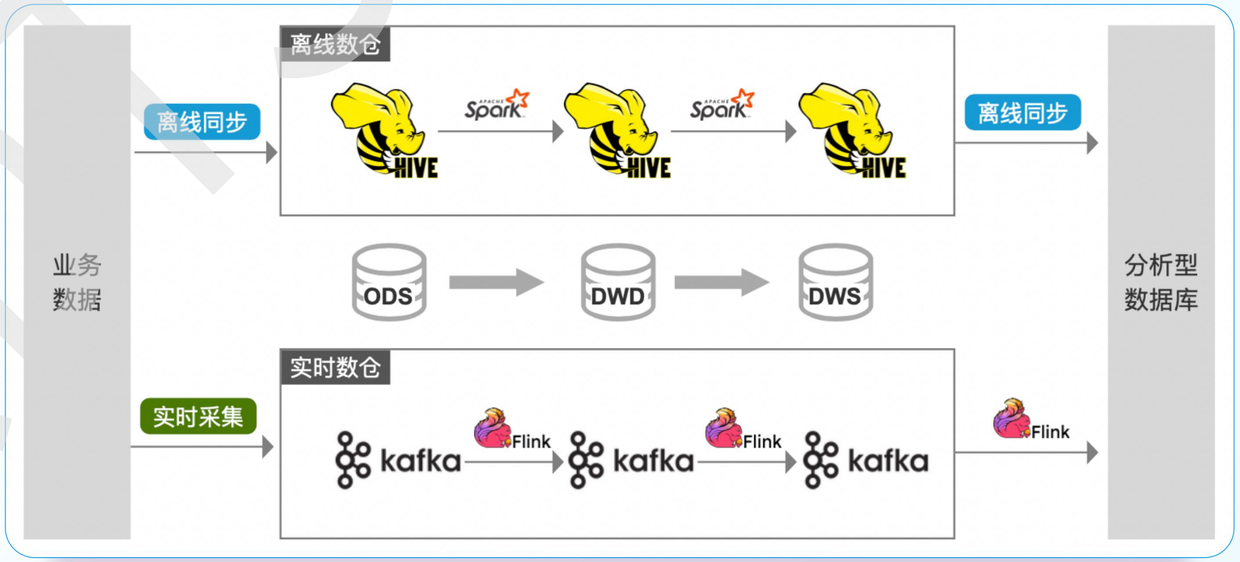
随着业务对数据时效性要求的提高，许多企业的实时任务体量，正在逐步接近存量离线任务。在数开和运维资源有限的情况下，这套架构的问题正在逐渐暴露出来。



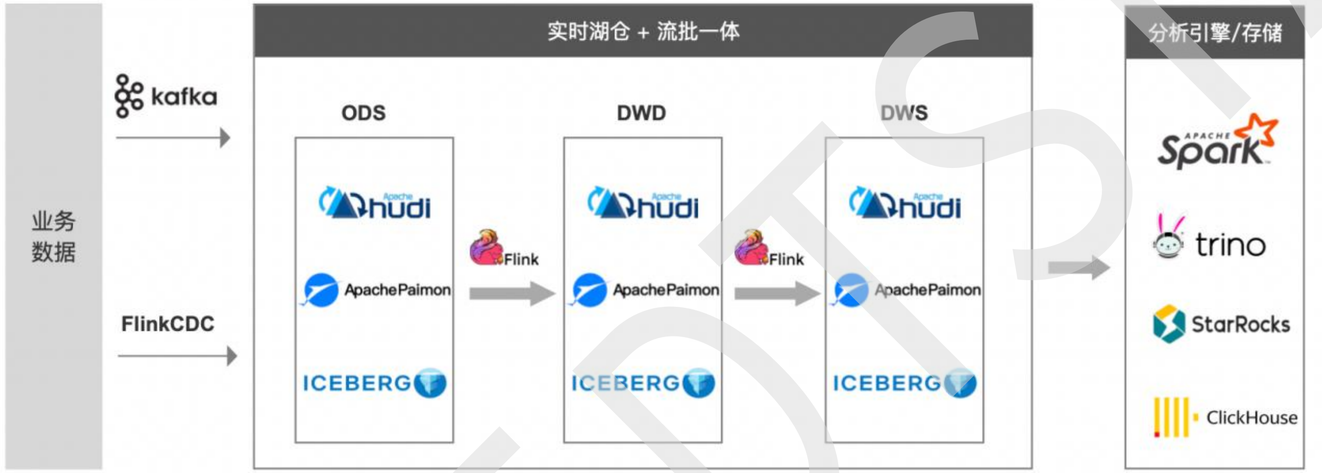
# 当前的企业困境

## Lambda 架构的问题

- 离线开发链路中的数据更新问题，在当前技术环境下显得越来越难以容忍；
- 实时开发链路中的数据不落地问题，无法支持历史数据回溯、查询分析等场景；
- 多种计算引擎，造成数开学习成本和运维管理成本的居高不下；
- 多种存储介质，造成数据存储冗余、批/流数据不一致的问题；
- .....



# 实时湖仓解决方案--Paimon



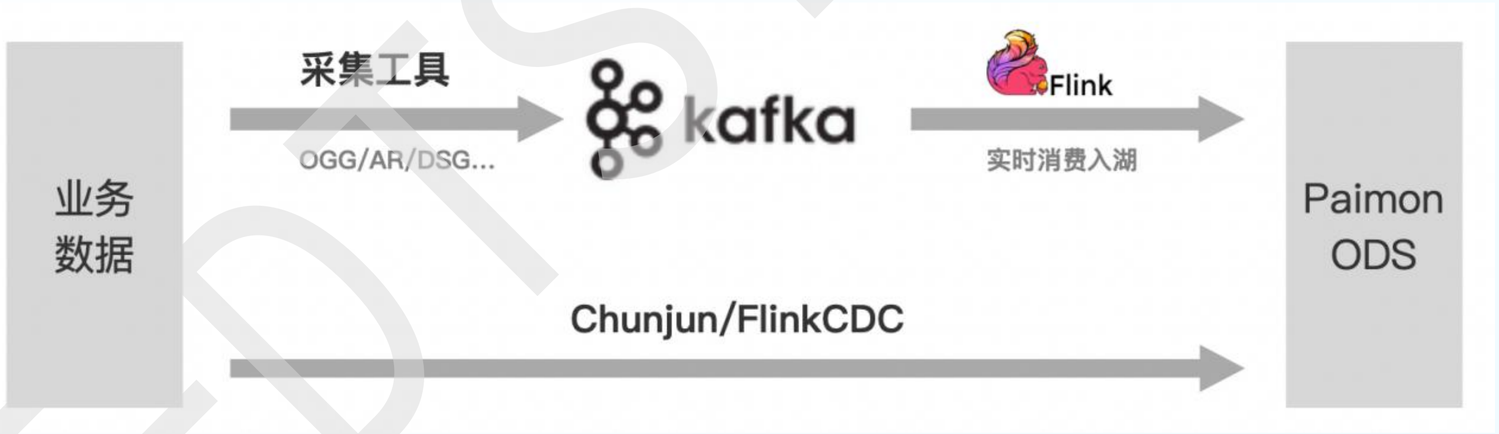

上图是一种实时湖仓解决方案，利用**湖存储的特性和Flink的流批计算能力**，统一存储和计算，解决 Lambda 架构的问题。

下面将以 Paimon 为例，**Paimon 是 Flink 内部基于 Flink Tablestore 孵化的一款湖存储产品**。和 Hudi/Iceberg 相比，Paimon 和 Flink 引擎有着更完整的兼容能力。

# 三步教你如何使用“Flink+数据湖”构建实时湖仓

袋鼠云 DTSTACK

## Step1：搭建实时ODS层




不管是通过 Flink 消费 Kafka，还是通过 FlinkCDC 采集日志，都可以将源库数据实时同步至 Paimon 中。

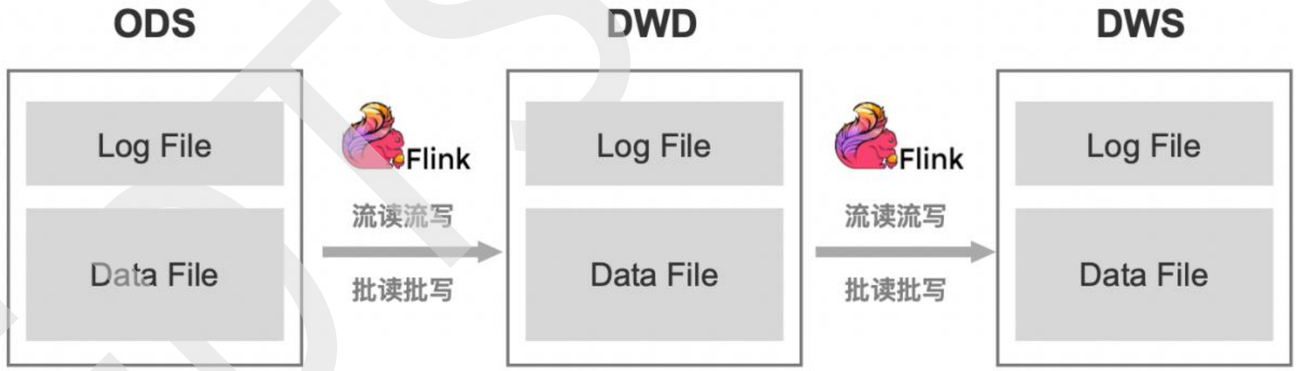
这样，无论上层是要做批计算还是流计算，都有份统一的实时 ODS 数据做基础，避免了数据不一致和存储冗余的问题。

# 三步教你如何使用“Flink+数据湖”构建实时湖仓

袋鼠云 DTSTACK

## Step2：加工湖仓中间层

关于实时湖仓的层级设计，可以参考成熟的离线数仓划分方案。




架构图中可以看出，Paimon 存储将文件分为 DataFile 和 LogFile：

- DataFile 用于存量数据的批计算。

- LogFile 用于增量数据的流计算。但毕竟是一种文件存储格式，其实时性只能做到分钟级别。如果业务场景对实时性有秒级/毫秒级要求，Paimon 也支持将 Kafka 外挂为 LogFile 使用，同时对上层应用暴露的，仍然只有一张 Paimon 表。

# 三步教你如何使用“Flink+数据湖”构建实时湖仓

袋鼠云 DTSTACK

**基于上面的特性，如何在实际应用体现出流/批一体能力，可以参考如下几种开发场景：**

- 流、批独立任务

根据实际业务场景需要，使用 Flink+Paimon 的统一技术栈，进行离线任务和实时任务的独立开发。

- 批流一体任务

在很多实时统计类的数开场景下，往往需要在完成存量数据统计的基础上，再衔接实时增量计算。传统的 Lambda 架构要完成这种场景，实现上相对比较复杂。而使用 Flink+Paimon，一个任务即可满足。

- 流批一体任务

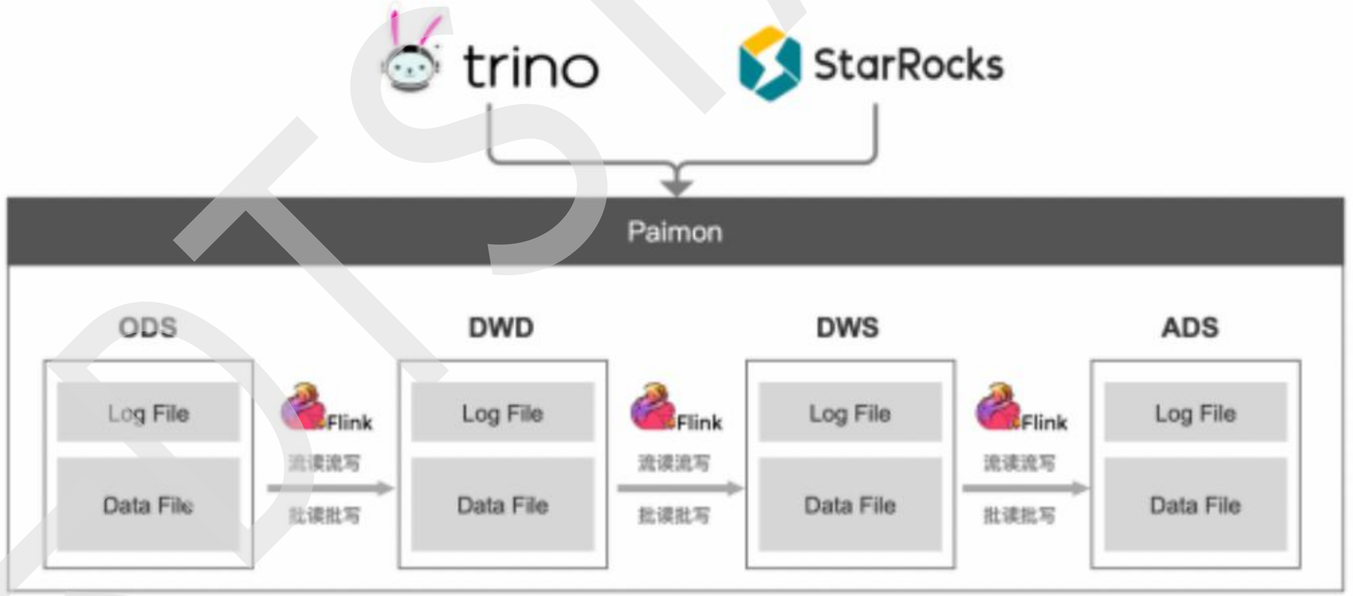
传统的 Lambda 架构中，为了保障 Flink+Kafka 实时计算的准确性，往往需要将 Kafka 数据双写一份到离线存储中。然后通过离线定时任务对实时计算结果做一次覆盖修正。而使用 Flink+Paimon，一个任务即可满足。

# 三步教你如何使用“Flink+数据湖”构建实时湖仓

袋鼠云 DTSTACK

## Step3：湖仓分析应用层

这层有两种不同的落地方案，
可以根据企业技术栈自由选型



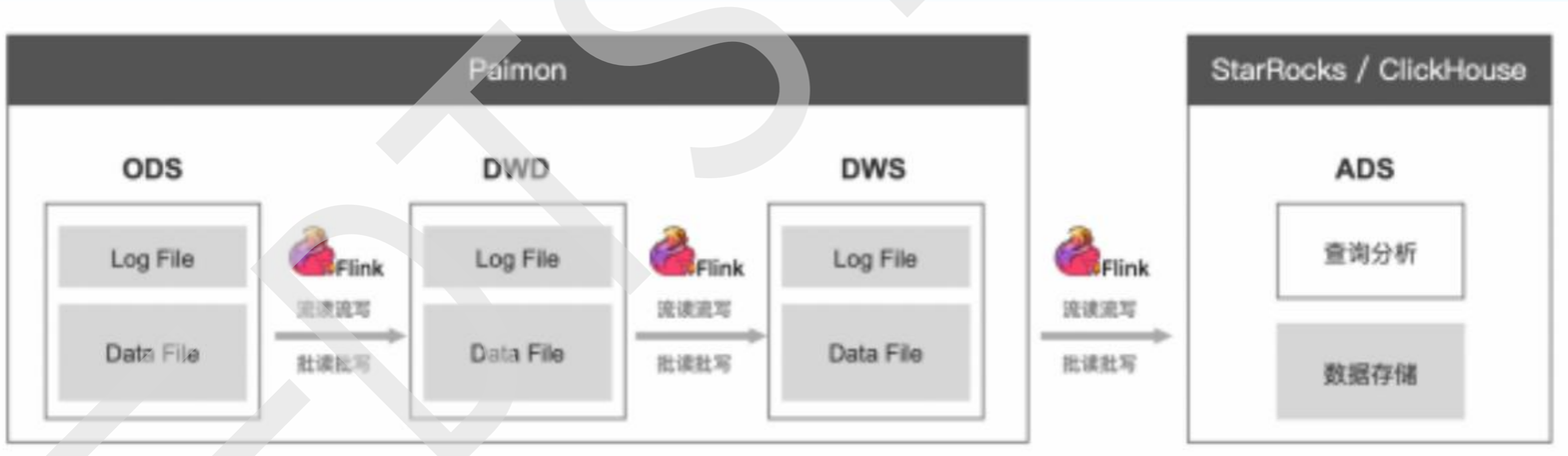

ADS 层数据也在数据湖加工落地，然后使用 OLAP 引擎如 Trino、StarRocks 直接对接数据湖，向上层提供数据分析能力。

这样做可以实现存储的完全统一，但是在查询分析性能上会有一定的牺牲。

# 三步教你如何使用“Flink+数据湖”构建实时湖仓

袋鼠云 DTSTACK

## Step1：搭建实时ODS层



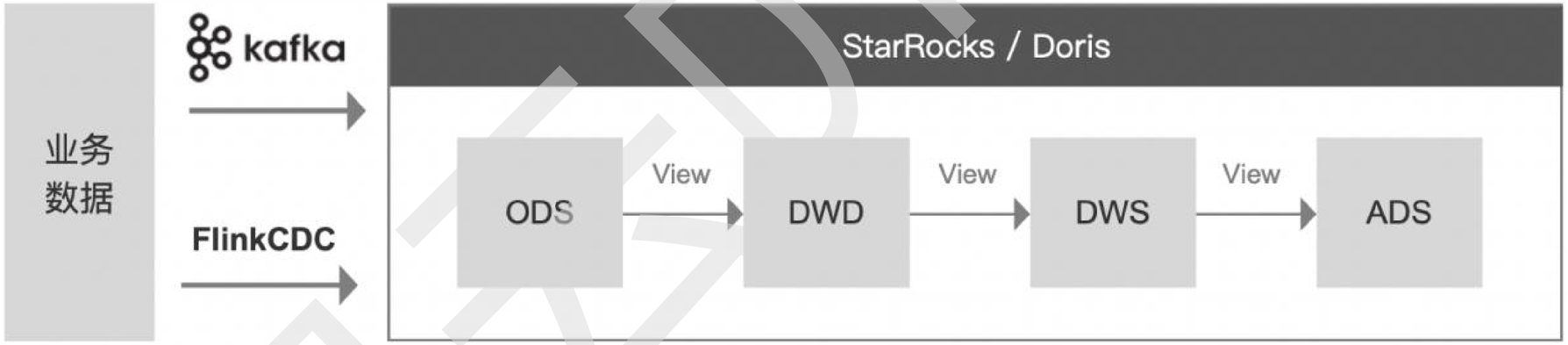

将 DWS 层数据加工后打入 StarRocks 或者 ClickHouse 这类存储+分析的统一引擎。

该方案可以充分利用起这类引擎的查询加速能力，对于 OLAP 场景有较高要求的企业，是个比较合适的方案。

# 其他选择？

目前业内比较热门的探索实践，不依赖 Hadoop 体系，仅利用 StarRocks/Doris 构建实时数仓的方式，

大致的架构图如下：



# 为什么企业需要实时湖仓？

理论上，该方案确实可行。

StarRocks/Doris 本身作为计算+存储一体的引擎，具备向量化、MPP 架构、CBO、智能物化视图、可实时更新等能力，在一定程度上可以满足构建实时数仓的要求。

但是，在我们接触过的一些金融客户的实际应用中发现，当数据体量较大、视图逻辑较复杂时，该方案存在明显的性能瓶颈。

而根据 StarRocks/Doris 官网自身高性能分析型数仓的定位，将它作为企业 OLAP 的选型，完全没有问题。但是寄希望于它承担全链路的大数据计算，目前来看还有很长的路要走。

所以，**将实时湖仓部分层级的计算，前移至“Flink+数据湖”的架构中，仍然是当前技术方案中最优的选择。**

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



DTSTACK