QCon
全球软件开发大会

InfoQ 极客传媒

湖流一体：
基于 Fluss 和 Lance 构建实时多模态数据湖

徐榜江

阿里云高级技术专家
Apache Flink PMC Member

目录

InfoQ 极客传媒

北京

QCon 全球软件开发大会

会议时间：4月10-12日

- 大模型赋能 AIOps
- 越挫越勇的大前端
- 云上业务架构演进
- 多模态大模型及应用
- 海外 AI 应用创新实践

北京

AiCon 全球人工智能开发与应用大会

会议时间：6月27-28日

- 端侧智能
- AI Agent
- 多模态大模型
- 金融+大模型

上海

QCon 全球软件开发大会

会议时间：10月23-25日

- AI Agent
- 大模型训练推理
- 端侧 AI
- 搜推深度融合
- 数智金融

4月 — 5月 — 6月 — 8月 — 10月 — 12月

上海

AiCon 全球人工智能开发与应用大会

会议时间：5月23-24日

- 通用大模型
- AI Agent
- 垂直领域应用
- 模型可解释性

深圳

AiCon 全球人工智能开发与应用大会

会议时间：8月22-23日

- 模型效率与部署
- 多模态大模型
- 大模型安全
- 智能硬件

北京

AiCon 全球人工智能开发与应用大会

会议时间：12月19-20日

- 通用大模型
- 智能硬件
- LMOps
- 具身智能

极客邦科技 2025 年会议规划

促进软件开发及相关领域知识与创新的传播

参会咨询

查看会议

01

AI时代对数据湖的需求与挑战

AI 应用爆发对数据存储的挑战

多模态数据快速增长

- 非结构化的数据随LLM和GenAI兴起快速增长
- 传统数据湖缺乏对多模态数据、元数据小文件支持

实时数据分析需求激增

- 实时推荐、实时分析、智能客服、AIGC等应用需要持续实时数据
- 特征工程、实时分析等场景需要高效列裁剪等能力，基于行存的传统消息队列很难满足

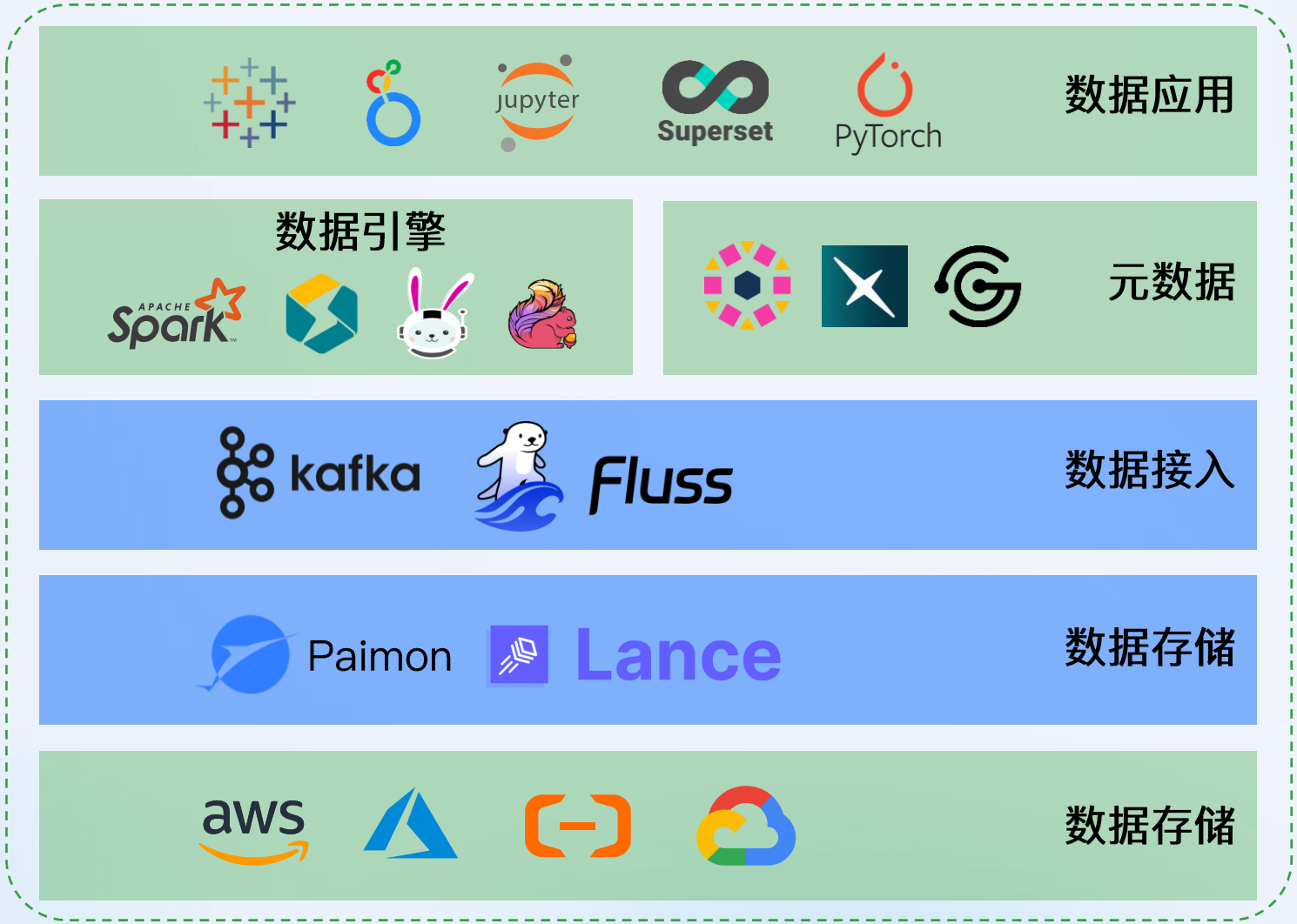
数据质量要求更高

- 模型训练过程需要高质量、版本化的数据集，需要数据溯源、数据血缘等能力
- 相比提升模型复杂度，数据质量更容易提升模型效果

数据合规和治理

- AI模型可能存则泄露隐私、生成幻觉、侵犯版权等问题
- AI应用需要系性地治理数据以满足安全和合规需求

AI 时代的数据湖需求 = 实时 + 多模态

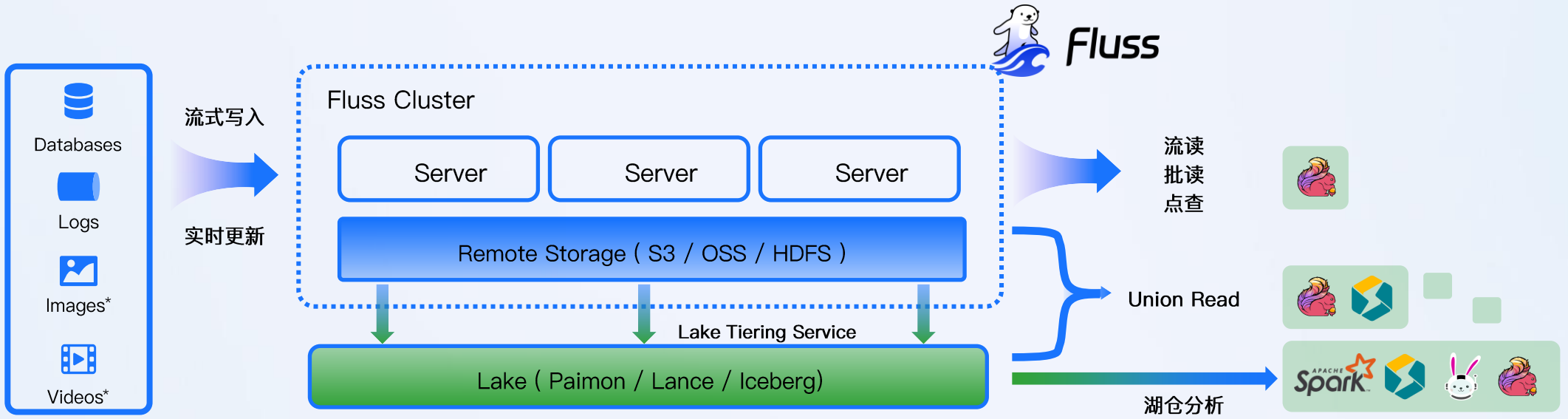


从数据湖到实时多模态数据湖

02

Fluss 流存储 和 Lance 湖格式

Apache Fluss：面向分析和AI的流存储



下一代流存储

- 毫秒级延迟
- 列存设计
- 数据探查
- 支持更新

面向分析

- 原生schema
- 实时CDC
- 列裁剪支持
- 表达式下推

面向AI

- 湖流一体
- AI Lake（Lance）
- Python Client
- 超宽表列裁剪

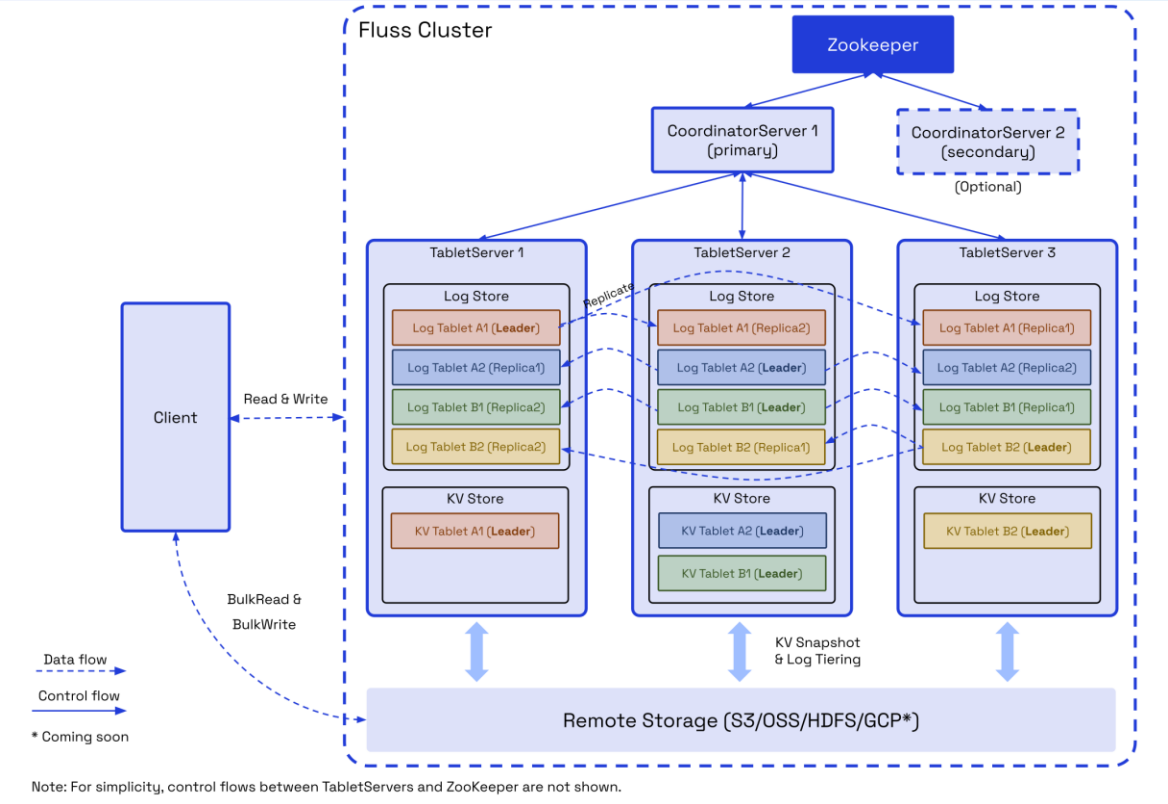
Apache Fluss 架构设计

面向分析场景设计

- 冷热分层：更好的数据管理
- 列式格式：高效列裁剪
- 索引支持：高效 scan
- 支持更新： Log表 + 主键表

面向 AI 场景设计

- Lakehouse 集成:
  - Lance & Paimon & Iceberg
- Python Client:
  - 打通AI生态,PyArrow,Pandas



Apache Fluss 应用: 淘宝数据平台

Kafka 方案

- 存储成本
  - 数据持续增长
  - 需要维护长周期数据
- 网络成本
  - 1 写 + 10 读
  - 跨AZ高流量

Fluss 方案

- 数据共享
  - 冷存入湖，长周期数据友好
  - 数据存储成本降低30%
- 列裁剪
  - 列存格式(Arrow)
  - 分区裁剪

成本 ↘ 30%

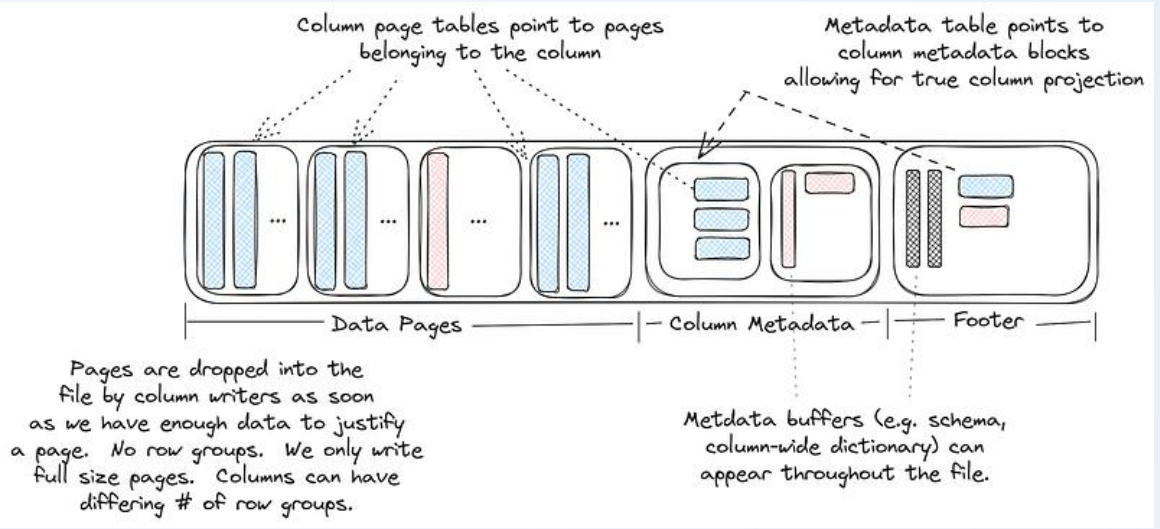
流量 ↘ 70%

3 PB	40 GB/s	500 K	500 B
数据存储规模	数据读取流量	大表点查QPS	单表行数

Lance：面向AI的多模态湖格式

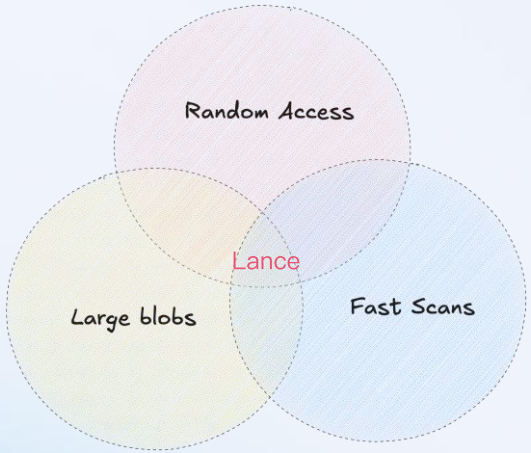
.lance: 为ML和LLM设计的文件格式

- Data Pages :
  - 抛弃row group，page大小存储友好
  - 按照page读取，计算引擎IO友好
- Column Metadata:
  - 列元数据独立存放，大宽表友好
  - 统计信息（skip-list/index）,快速过滤和点查
- footer:
  - 元数据表查询列元数据，真正的列裁剪

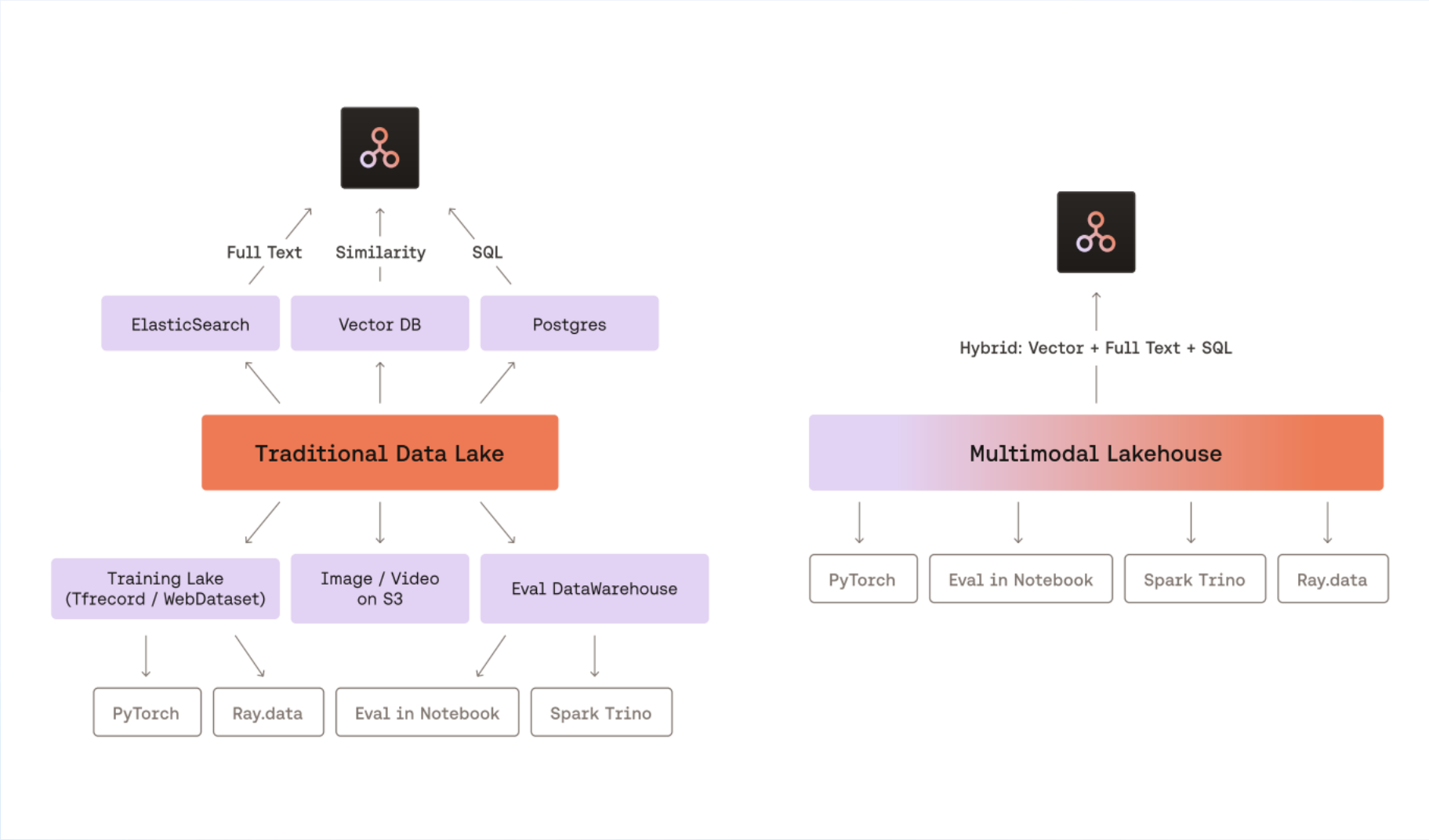


面向 AI 工作负载的 tradeoff

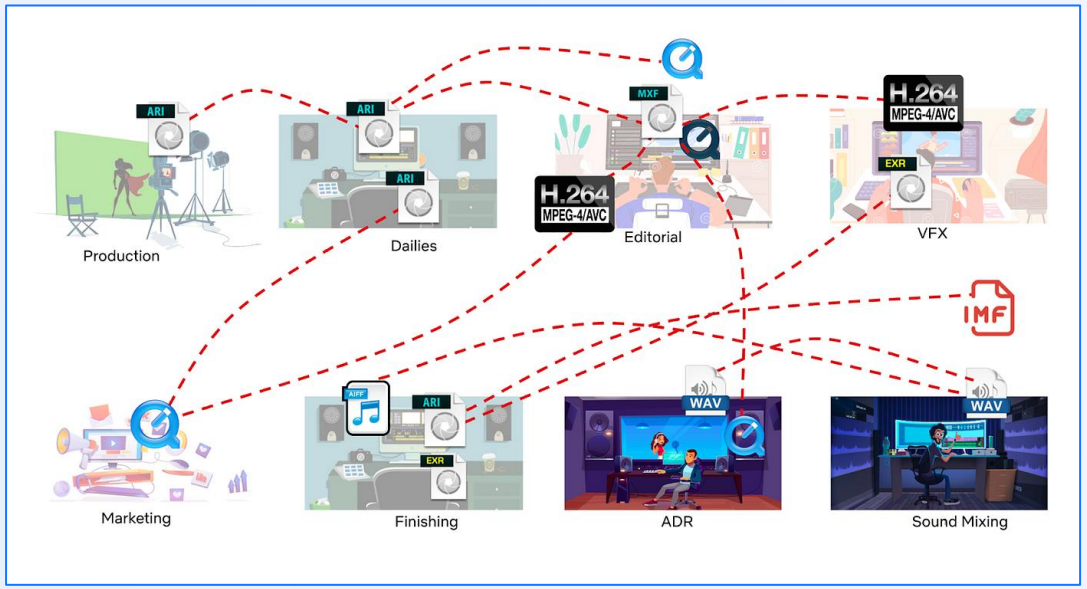
- 多模态数据，超宽列支持
  - 将超大单元作为 large Blob处理，inline 存储，降低读取IO
- 向量搜索、全文搜索场景下的数据查询
  - 列元数据独立存放，大宽表友好
- 特征工程，大宽表支持
  - 元数据是按列独立存储，解决parquet在该场景的元数据瓶颈

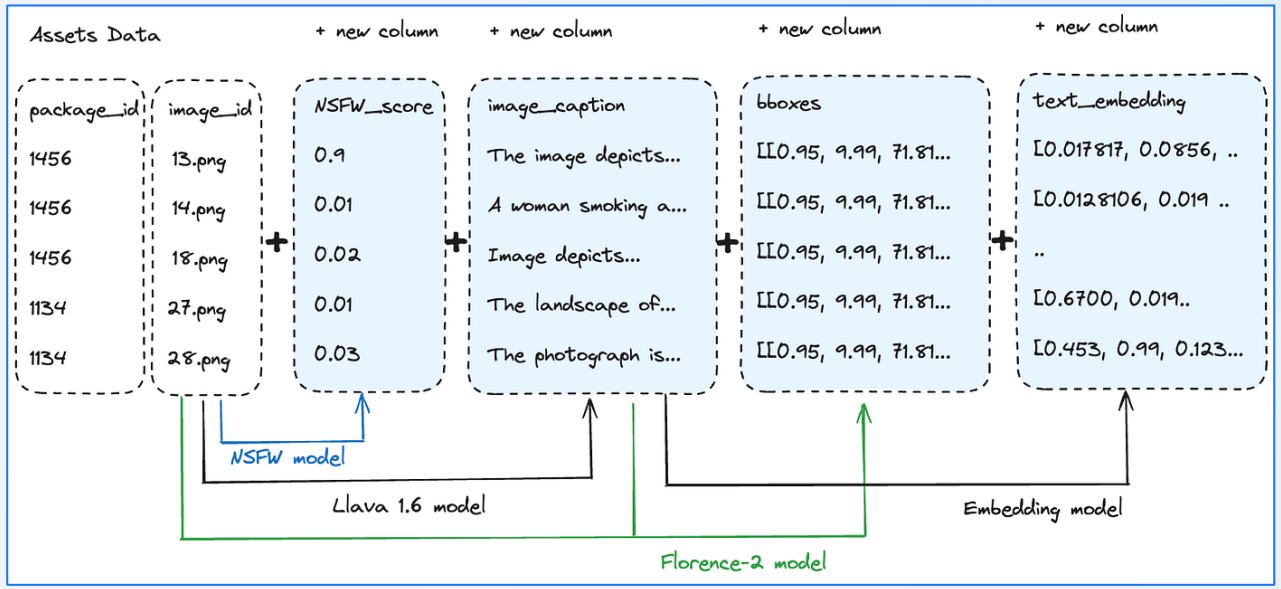


LanceDB：基于 Lance 的一体化多模态数据湖



LanceDB 应用: Netflix Media Data Lake





模型质量提升

模型评估更快

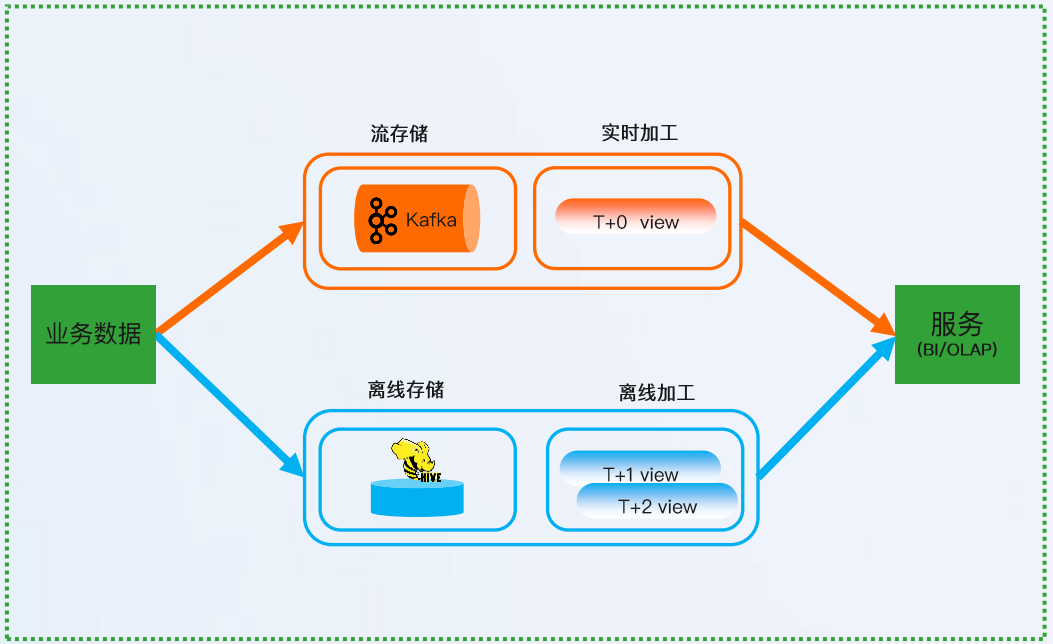
加速 AI 创新

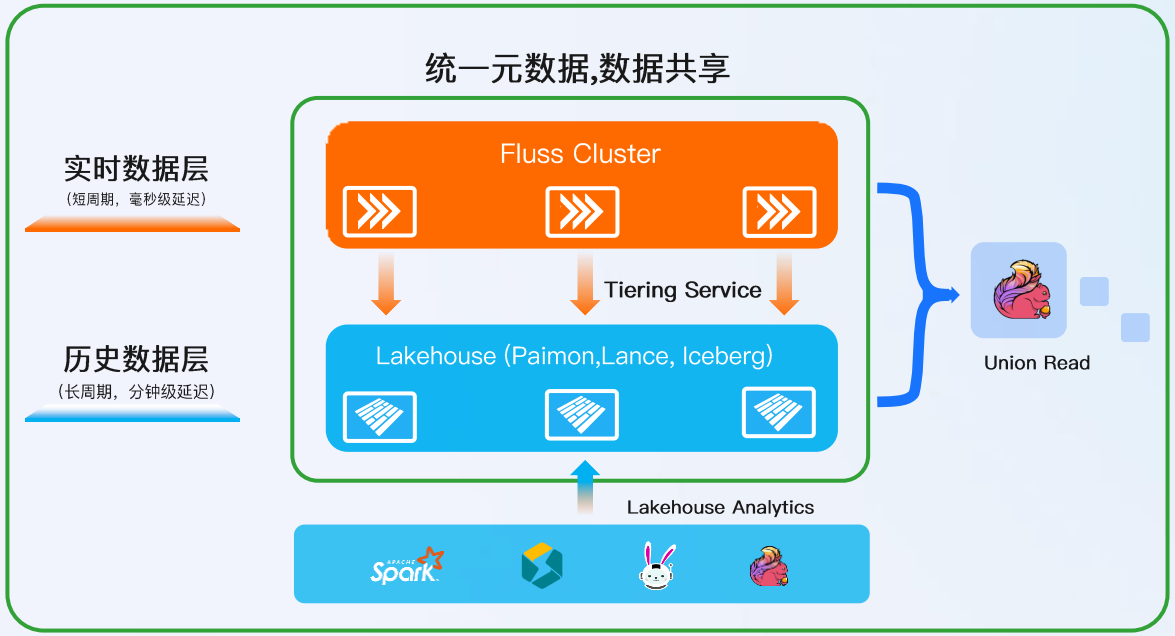
业务洞察更好

03

湖流一体架构和核心收益

湖流一体业务场景





为什么需要湖流一体?

- 两套存储，存储成本高
- 两条链路计算结果不一致
- 两套服务，开发运维成本高

Fluss 解决方案:

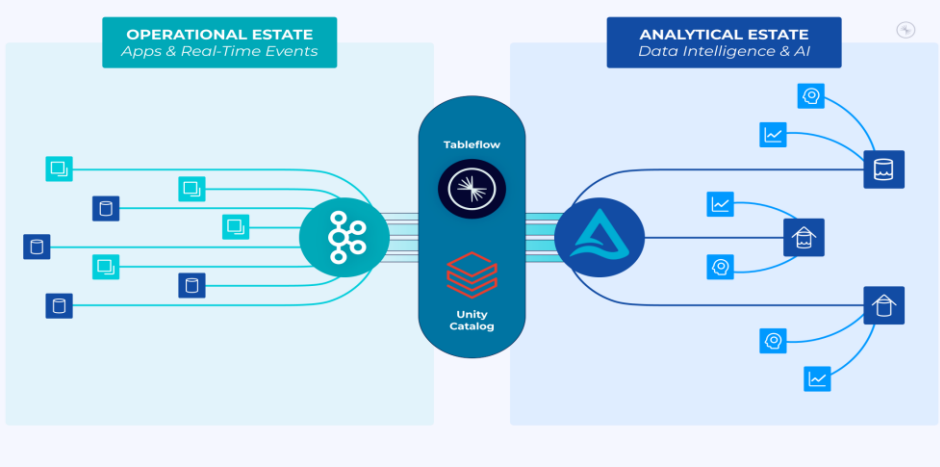
- ✓ 内置湖流通道服务，流存储数据直接入湖
- ✓ 数据格式高效转换， Arrow 到 Parquet
- ✓ 深度集成Flink引擎，Filter&Projection下推

业务收益:

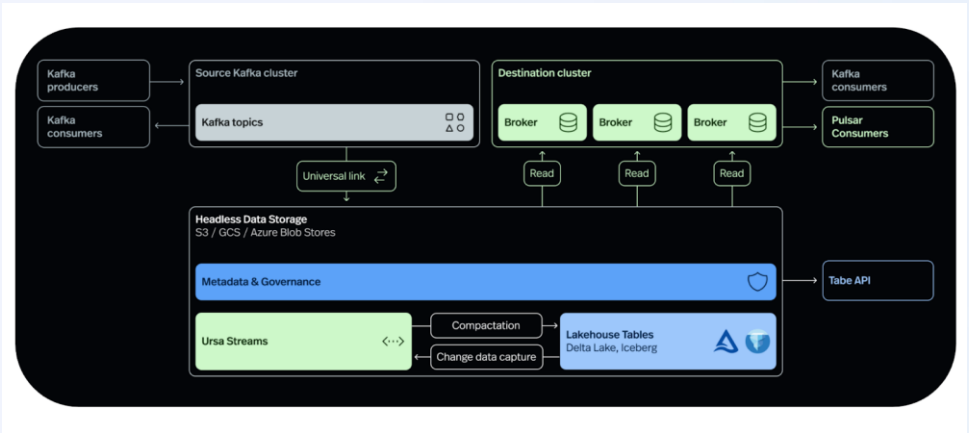
- ✓ 一份存储，流存：7X24h –> 1h
- ✓ 延迟更低，存储开放
- ✓ 链路简化，开发运维成本更低

湖流一体业界趋势

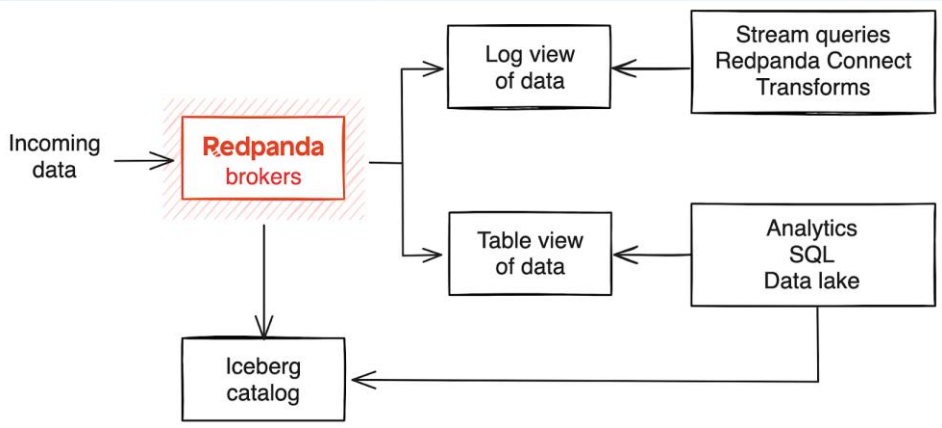
Confluent Kafka Tableflow



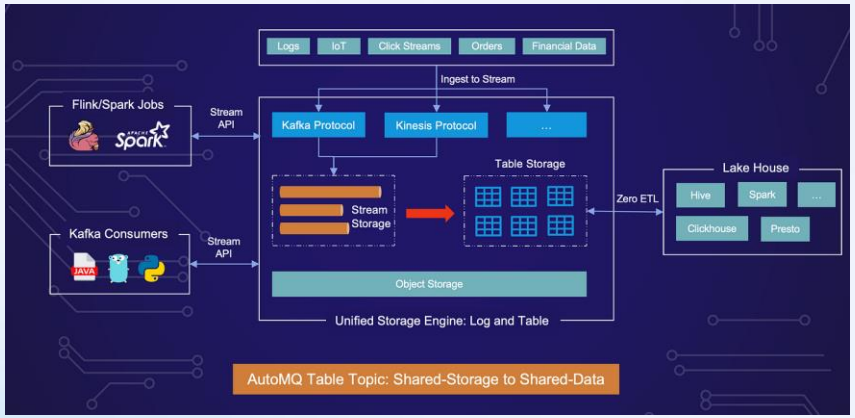
Ursa（VLDB-2025 Best Industry Paper）



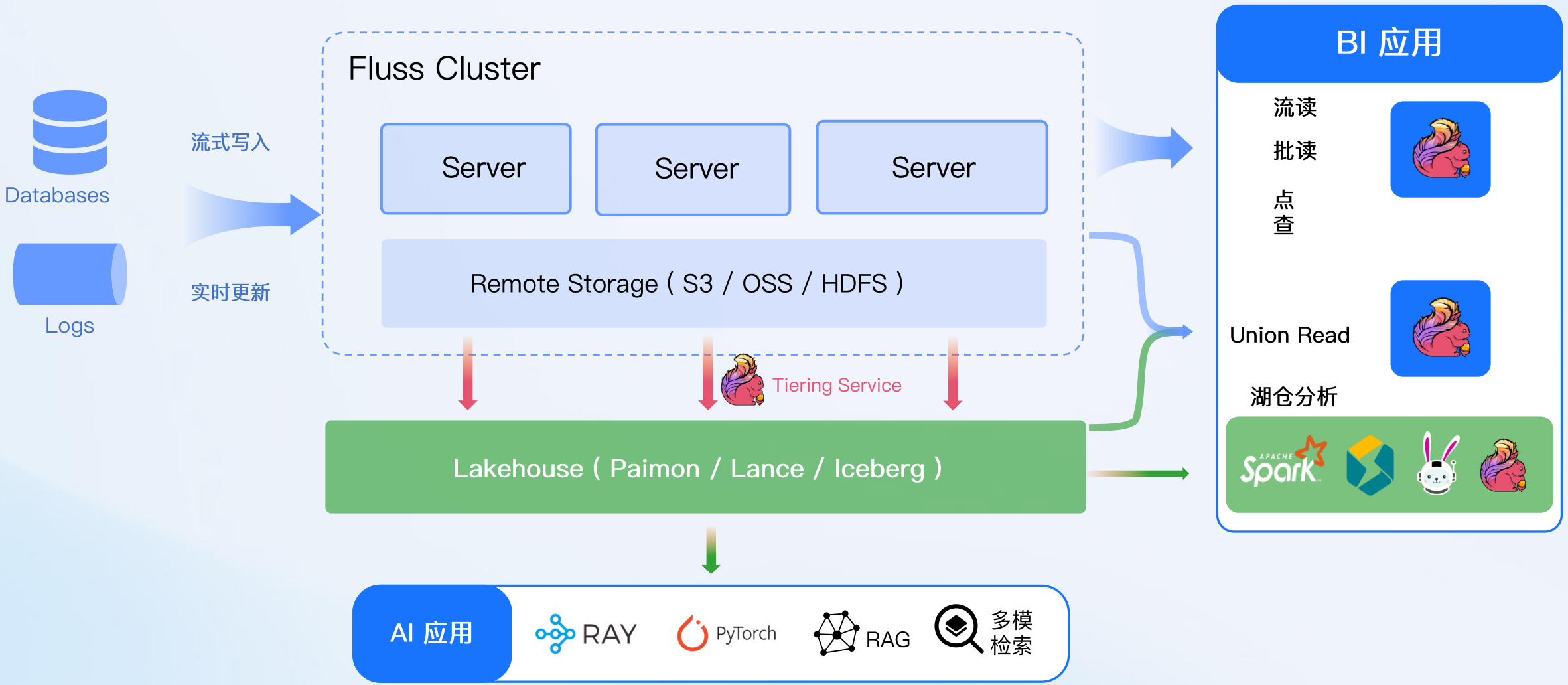
Redpanda Iceberg Topic



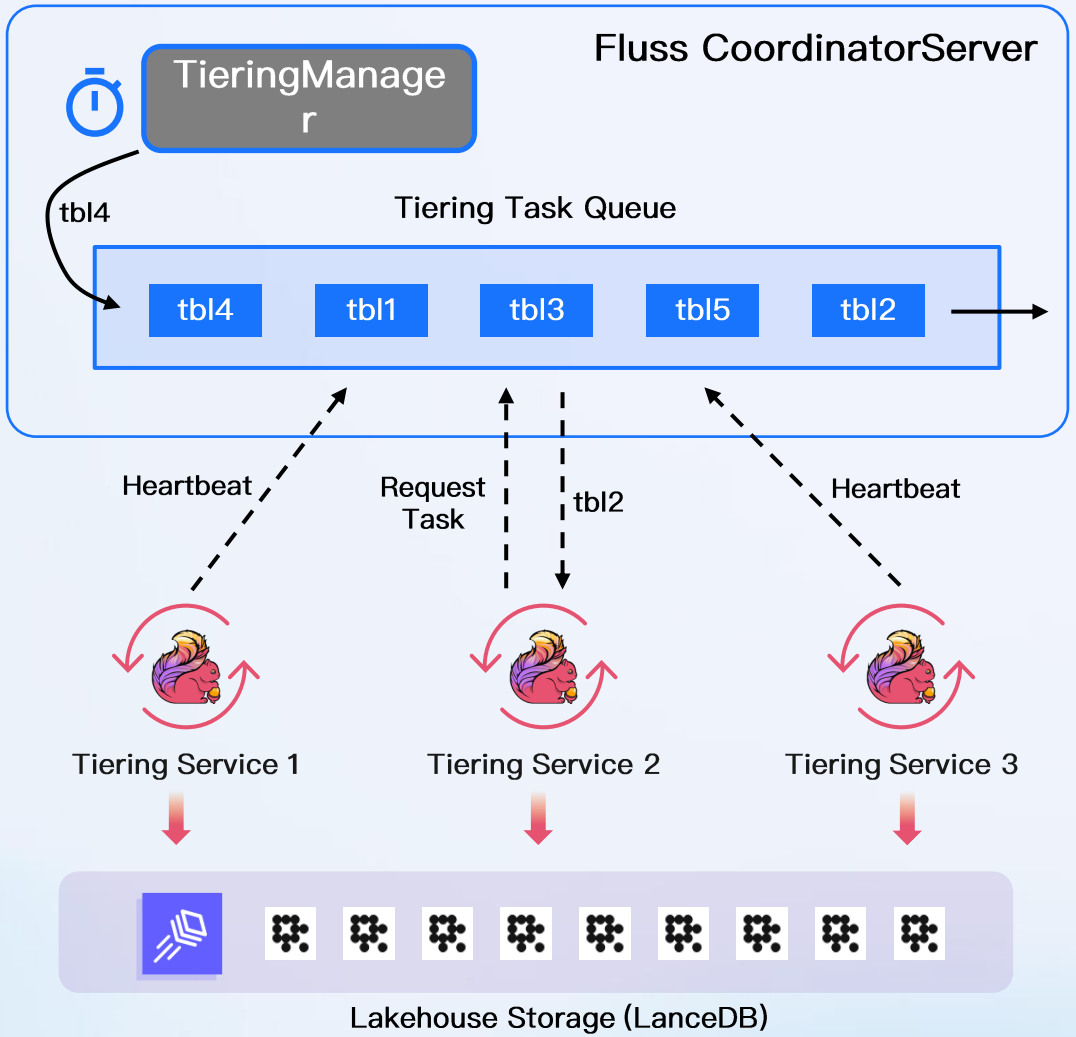
AutoMQ Table Topic



Fluss 湖流一体架构

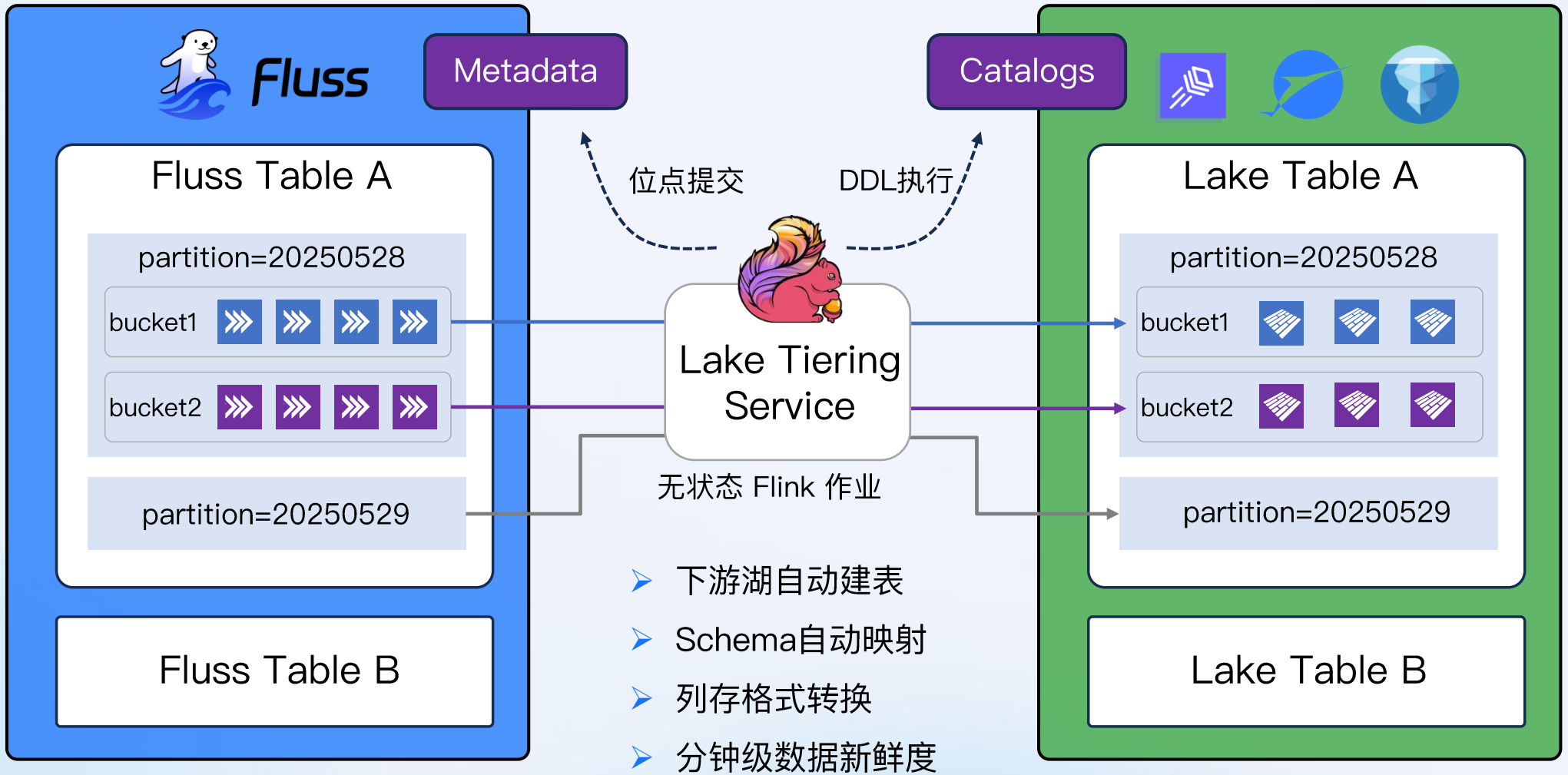


Fluss 湖流一体核心技术：Lake Tiering Service



- ✓ 无状态设计
- ✓ 灵活扩缩容
- ✓ 一键部署启动

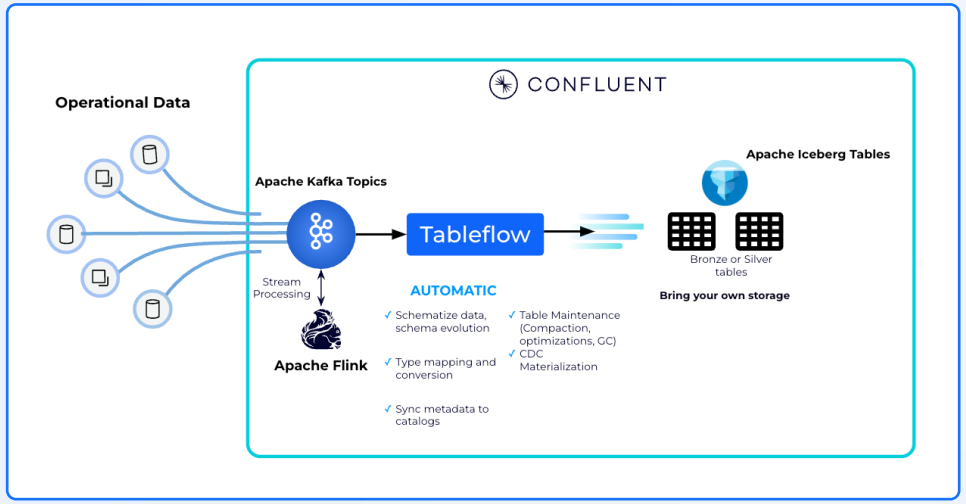
Fluss 湖流一体核心技术： Lake Tiering Service



- 下游湖自动建表
- Schema自动映射
- 列存格式转换
- 分钟级数据新鲜度

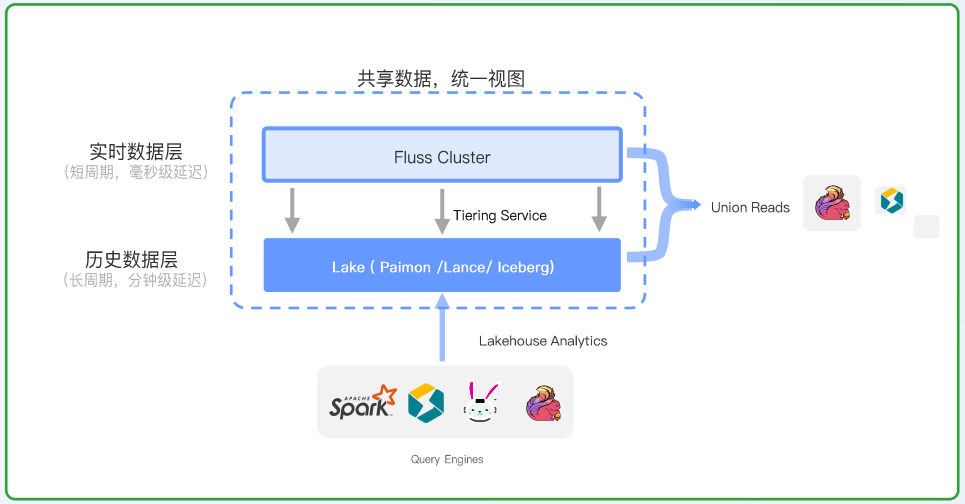
Fluss 湖流一体核心收益

Confluent Tableflow



Tableflow 是连接“数据湖”和“面向事件的流存储”两种系统

Fluss 湖流一体



湖流一体是打通“数据湖”和“面向分析的流存储”两种系统

✗ 数据的单向流动，只是个数据同步工具	✓ 数据双向共享，湖流相互增强
湖和流仍然割裂，并无额外收益	流 增强 湖：湖数据延迟提升到秒级! 湖 增强 流：流数据可分析!
数据的两份拷贝，高昂的成本	消除冗余存储：流存储成本降低10倍!
Kafka 不是为分析设计的，不同数据模型映射转换成本高	面向分析场景设计的流存储：模型对齐，列存文件直转

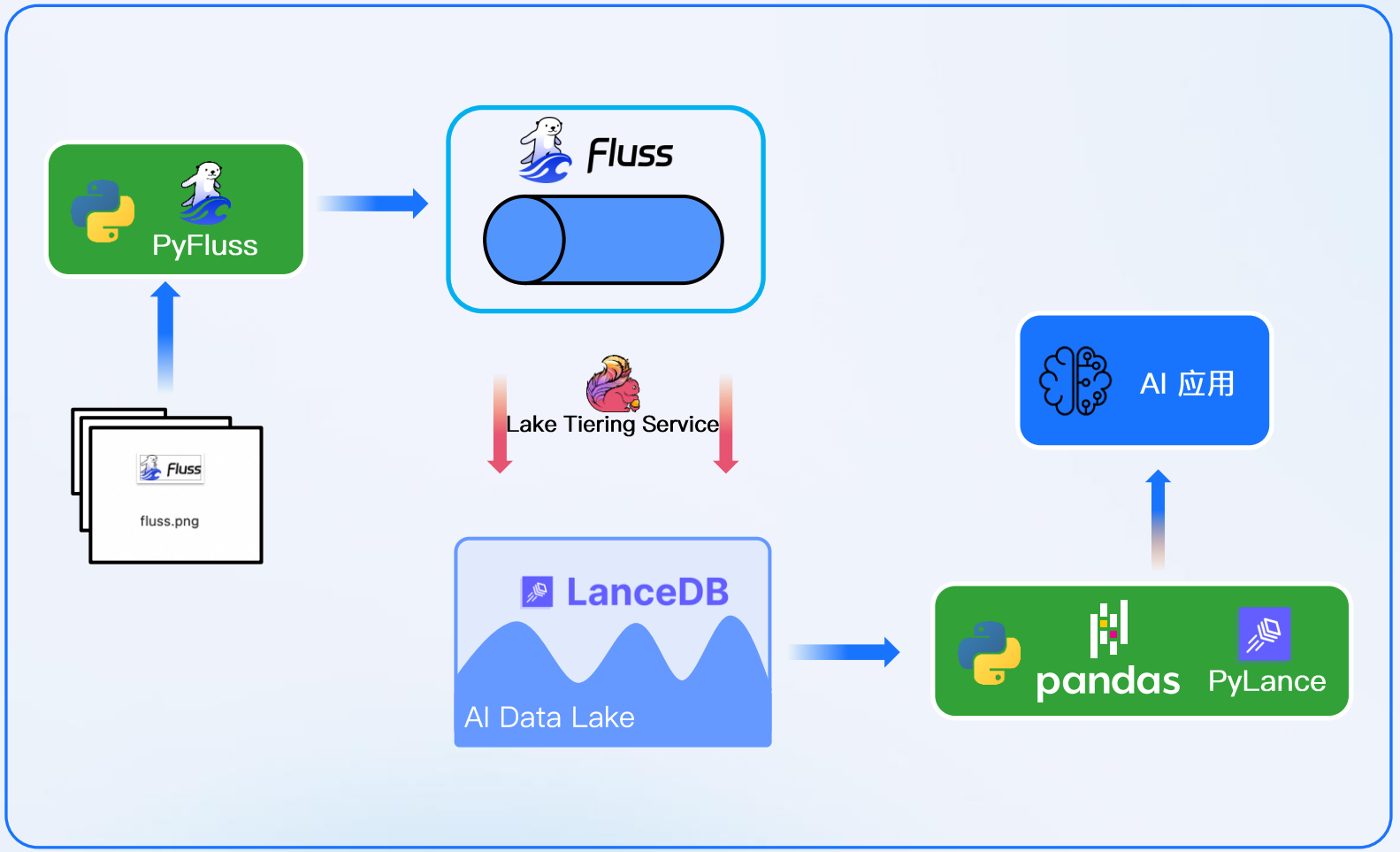
04

Demo: 实时多模态数据湖构建

Demo: Fluss + Lance 构建多模态数据湖

业务场景

1. 多模态数据流式摄入
2. 多模态数据自动入湖
3. AI Data Lakehouse 分析
4. 基于Data Lake构建AI应用



Demo: Fluss + Lance 构建多模态数据湖

主要流程

1.安装Fluss和Lance Python Lib

2. 启动 Fluss 湖流一体服务

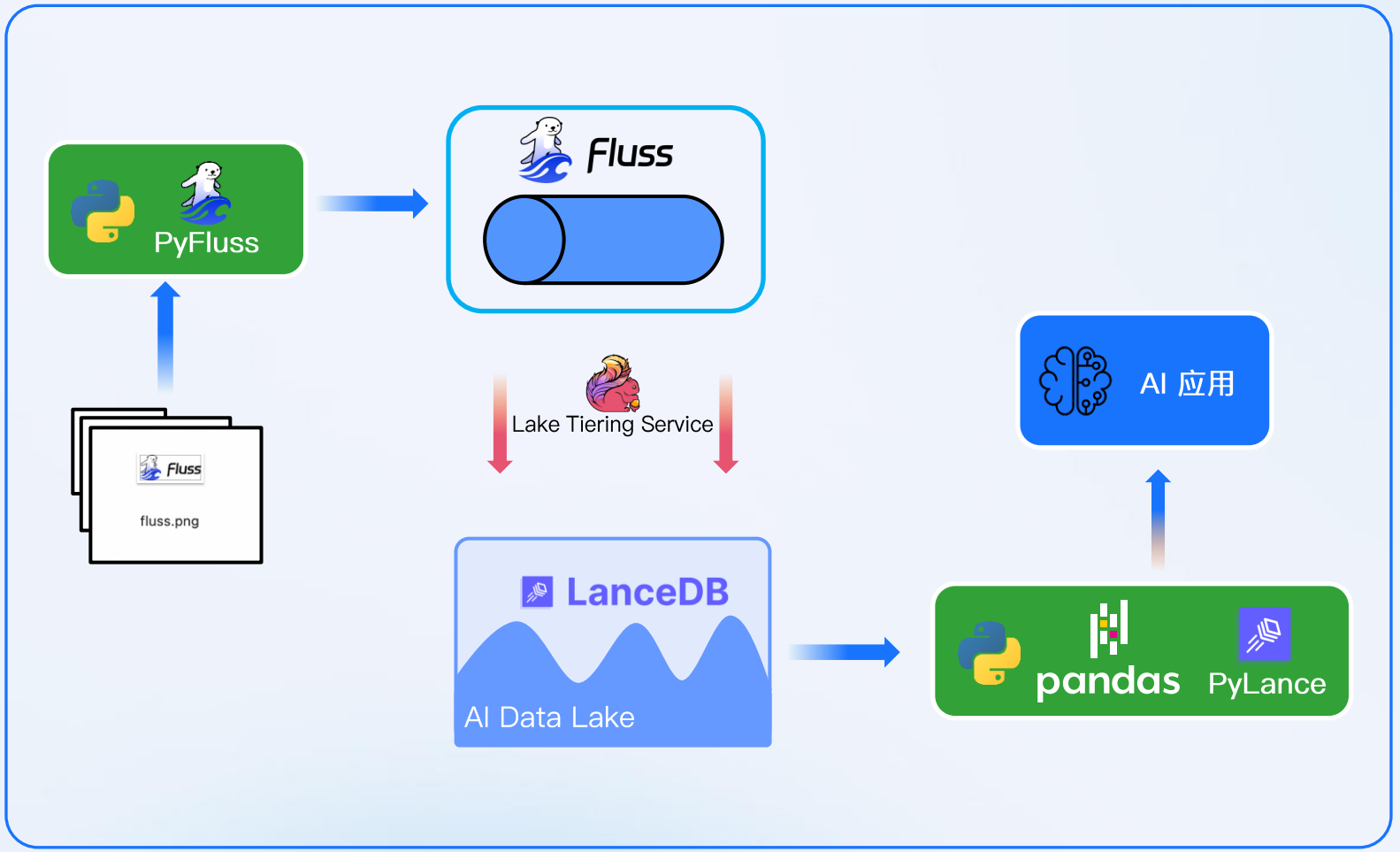
3. 连接Fluss集群并建表

4. 处理多模态数据(图片)

5. 多模态数据写入Fluss 表

6. Fluss 数据自动入湖Lake

7. 加载Lance图片到Pandas分析



Demo: Fluss + Lance 构建多模态数据湖

主要流程

1.安装Fluss和Lance Python Lib

2. 启动 Fluss 湖流一体服务

3. 连接Fluss集群并建表

4. 处理多模态数据(图片)

5. 多模态数据写入Fluss 表

6. Fluss 数据自动入湖Lake

7. 加载Lance图片到Pandas分析

```
$> pip install python-for-fluss pyarrow pylance pandas
```

```
$> ./bin/flink run <path_to_jar>/fluss-flink-tiering-0.8.jar \
    --fluss.bootstrap.servers localhost:9123 \
    --datalake.format lance \
    --datalake.lance.warehouse s3://lance \
    --datalake.lance.endpoint http://localhost:9000 \
    --datalake.lance.allow_http true \
    --datalake.lance.secret_access_key minioadmin \
    --datalake.lance.access_key_id minio \
    --datalake.lance.region eu-west-1 \
    --table.datalake.freshness 30s
```

Demo: Fluss + Lance 构建多模态数据湖

主要流程

1.安装Fluss和Lance Python Lib

2. 启动 Fluss 湖流一体服务

3. 连接Fluss集群并建表

4. 处理多模态数据(图片)

5. 多模态数据写入Fluss 表

6. Fluss 数据自动入湖Lake

7. 加载Lance图片到Pandas分析

```python
import fluss
import pyarrow as pa

async def create_table(conn, table_path, pa_schema):
    # Create a Fluss Schema
    fluss_schema = fluss.Schema(pa_schema)
    # Create a Fluss TableDescriptor
    table_descriptor = fluss.TableDescriptor(
        fluss_schema,
        properties={
            "table.datalake.enabled": "true",
            "table.datalake.freshness": "30s",
        },
    )
    # Get the admin for Fluss
    admin = await conn.get_admin()
    # Create a Fluss table
    await admin.create_table(table_path, table_descriptor, True)
```

Demo: Fluss + Lance 构建多模态数据湖

主要流程

1.安装Fluss和Lance Python Lib

2. 启动 Fluss 湖流一体服务

3. 连接Fluss集群并建表

4. 处理多模态数据(图片)

5. 多模态数据写入Fluss 表

6. Fluss 数据自动入湖Lake

7. 加载Lance图片到Pandas分析

```python
import os
def process_images(schema: pa.Schema):
        # Get the current directory path
        current_dir = os.getcwd()
        images_folder = os.path.join(current_dir, "image")
        # Get the list of image files
        image_files = [filename for filename in os.listdir(images_folder)
         if filename.endswith((".png", ".jpg", ".jpeg"))]
        # Iterate over all images in the folder with tqdm
        for filename in tqdm(image_files, desc="Processing Images"):
                # Construct the full path to the image
                image_path = os.path.join(images_folder, filename)
                # Read and convert the image to a binary format
                with open(image_path, 'rb') as f:
                        binary_data = f.read()
                image_array = pa.array([binary_data], type=pa.binary())
                # Yield RecordBatch for each image
                yield pa.RecordBatch.from_arrays([image_array], schema=schema)
```

Demo: Fluss + Lance 构建多模态数据湖

主要流程

1.安装Fluss和Lance Python Lib

2. 启动 Fluss 湖流一体服务

3. 连接Fluss集群并建表

4. 处理多模态数据(图片)

5. 多模态数据写入Fluss 表

6. Fluss 数据自动入湖Lake

7. 加载Lance图片到Pandas分析

```python
async def write_to_fluss(conn, table_path, pa_schema):
    # Get table and create writer‘
    table = await conn.get_table(table_path)
    append_writer = await table.new_append_writer()
    try:
        print("\n--- Writing image data ---")
        for record_batch in process_images(pa_schema):
            append_writer.write_arrow_batch(record_batch)
    except Exception as e:
        print(f"Failed to write image data: {e}")
    finally:
        append_writer.close()
```

Demo: Fluss + Lance 构建多模态数据湖

主要流程

1.安装Fluss和Lance Python Lib

2. 启动 Fluss 湖流一体服务

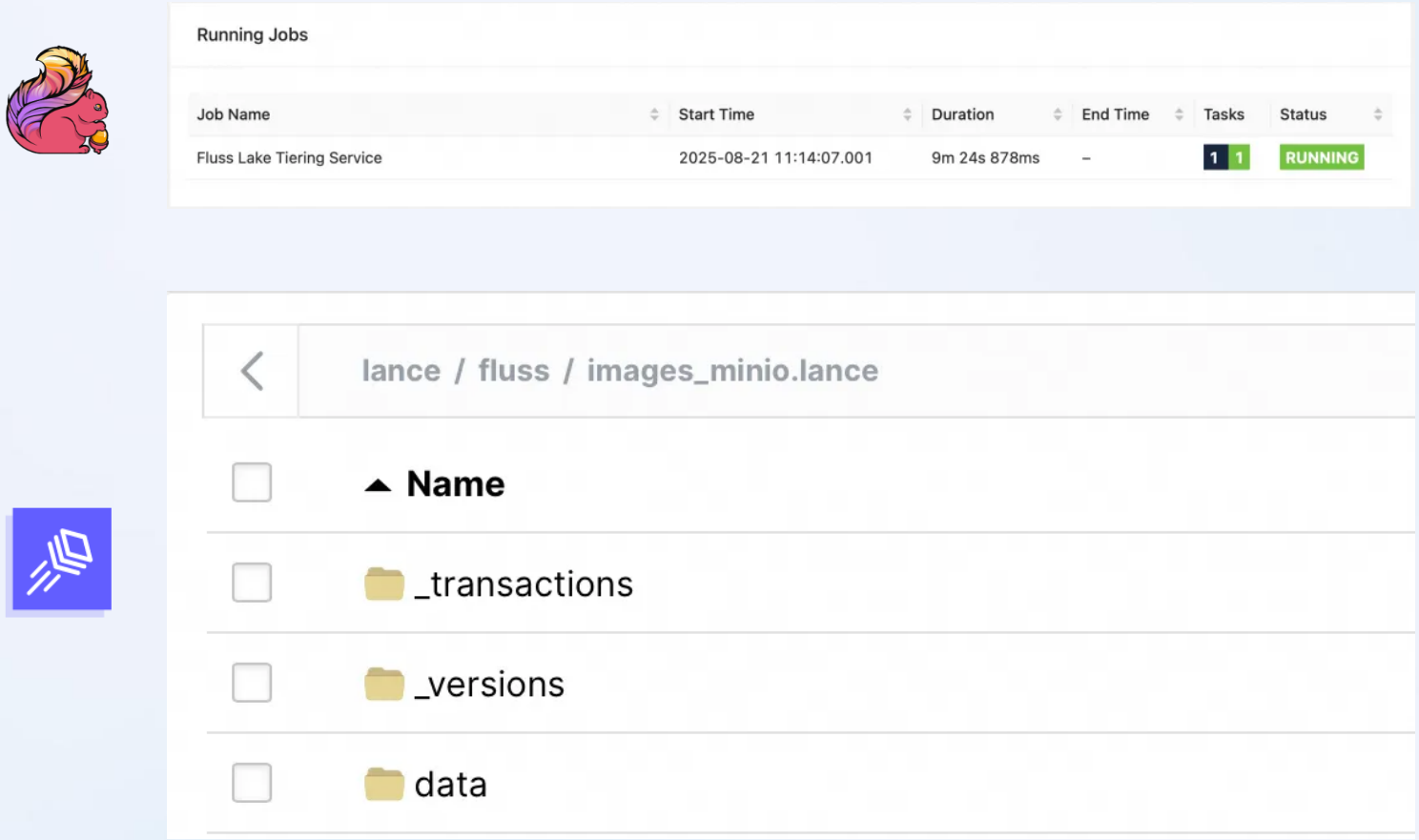
3. 连接Fluss集群并建表

4. 处理多模态数据(图片)

5. 多模态数据写入Fluss 表

6. Fluss 数据自动入湖Lake

7. 加载Lance图片到Pandas分析



Demo: Fluss + Lance 构建多模态数据湖

主要流程

1.安装Fluss和Lance Python Lib

2. 启动 Fluss 湖流一体服务

3. 连接Fluss集群并建表

4. 处理多模态数据(图片)

5. 多模态数据写入Fluss 表

6. Fluss 数据自动入湖Lake

7. 加载Lance图片到Pandas分析

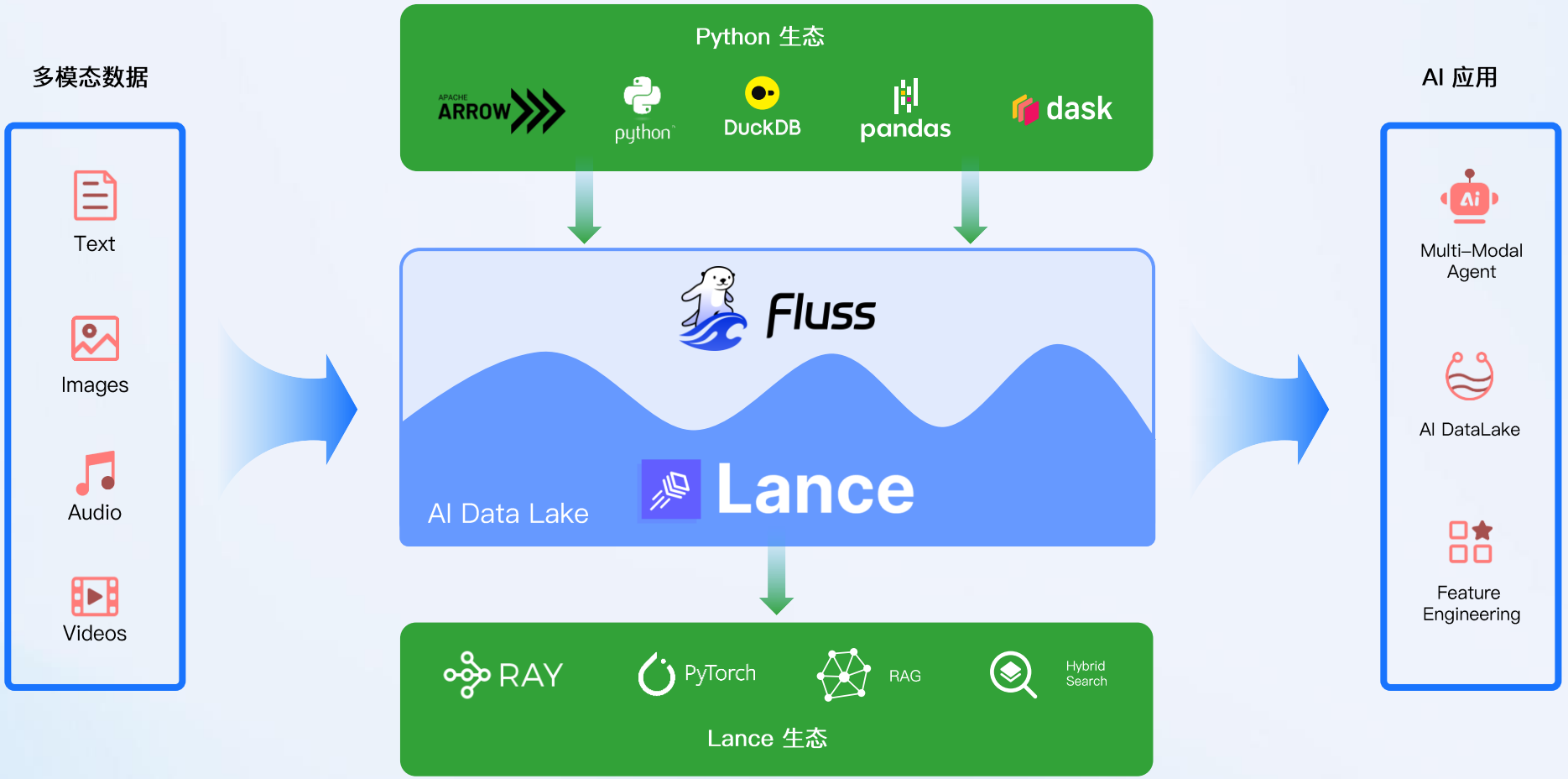
```python
import lance
import pandas as pd
def loading_into_pandas(table_name):
        dataset = lance.dataset(
        "s3://lance/fluss/" + table_name + ".lance",
        storage_options={
            "access_key_id": "minio",
            "secret_access_key": "minioadmin",
            "endpoint": "http://localhost:9000",
            "allow_http": "true",
        },
    )

    df = dataset.to_table().to_pandas()
        print("Pandas DataFrame is ready")
        print("Total Rows: ", df.shape[0])
```

05

总结与展望

Fluss + Lance = 实时多模态数据湖 => AI Data Lake



Fluss 未来规划

完善湖格式集成

Lance(主键表),Hudi*

更多计算引擎

StarRocks*, Spark, DuckDB

更好时效性

Shared Metadata, Deletion Vector

欢迎加入开源社区

Apache Fluss

https://github.com/apache/fluss

https://fluss.apache.org/

LanceDB

https://github.com/lancedb/lancedb

https://lancedb.com/

InfoQ 极客传媒

北京

QCon 全球软件开发大会

会议时间：4月10-12日

- 大模型赋能 AIOps
- 越挫越勇的大前端
- 云上业务架构演进
- 多模态大模型及应用
- 海外 AI 应用创新实践

北京

AiCon 全球人工智能开发与应用大会

会议时间：6月27-28日

- 端侧智能
- AI Agent
- 多模态大模型
- 金融+大模型

上海

QCon 全球软件开发大会

会议时间：10月23-25日

- AI Agent
- 大模型训练推理
- 端侧 AI
- 搜推深度融合
- 数智金融

4月 — 5月 — 6月 — 8月 — 10月 — 12月

上海

AiCon 全球人工智能开发与应用大会

会议时间：5月23-24日

- 通用大模型
- AI Agent
- 垂直领域应用
- 模型可解释性

深圳

AiCon 全球人工智能开发与应用大会

会议时间：8月22-23日

- 模型效率与部署
- 多模态大模型
- 大模型安全
- 智能硬件

北京

AiCon 全球人工智能开发与应用大会

会议时间：12月19-20日

- 通用大模型
- 智能硬件
- LMOps
- 具身智能

极客邦科技 2025 年会议规划

促进软件开发及相关领域知识与创新的传播

参会咨询

查看会议

QCon
全球软件开发大会

InfoQ 极客传媒

THANKS

大模型正在重新定义软件

Large Language Model Is Redefining The Software

完整 Demo 参考：

链接：基于 Apache Fluss（孵化中）与 Lance 搭建实时多模态 AI 分析系统