Apache Flink

Flink 和 Iceberg 如何解决数据入湖面临的挑战

胡争 Apache Iceberg Committer 2021-4-17

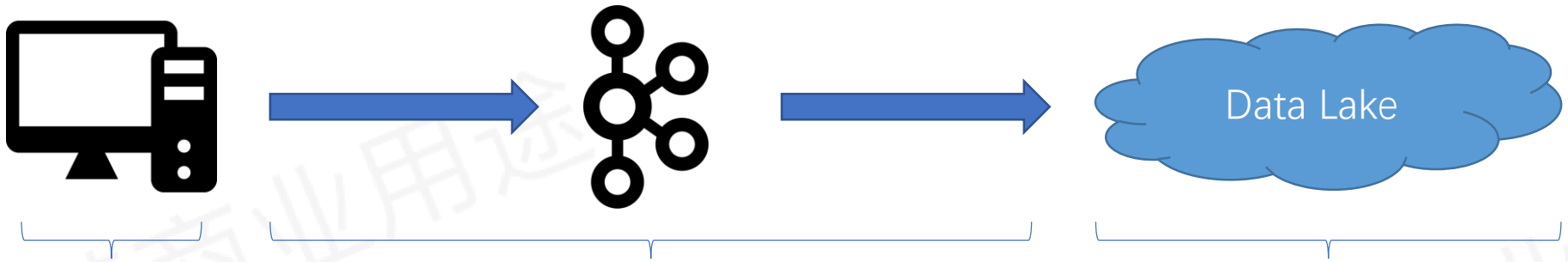
Apache Flink x Iceberg Meetup ·上海站

CONTENT

目录 >>

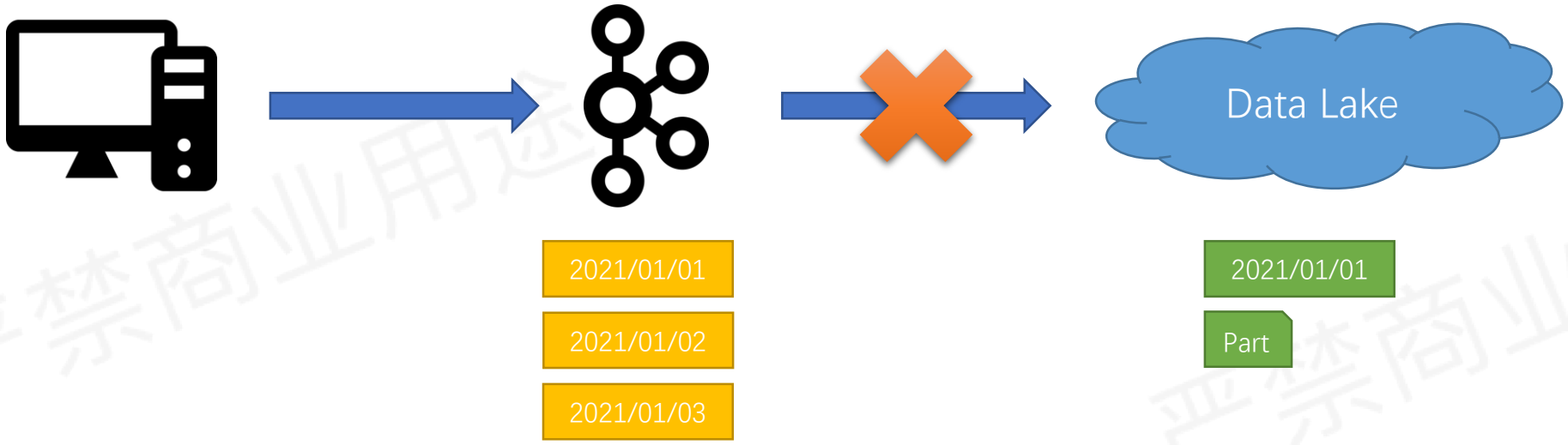
#1 数据入湖的核心挑战

数据实时入湖

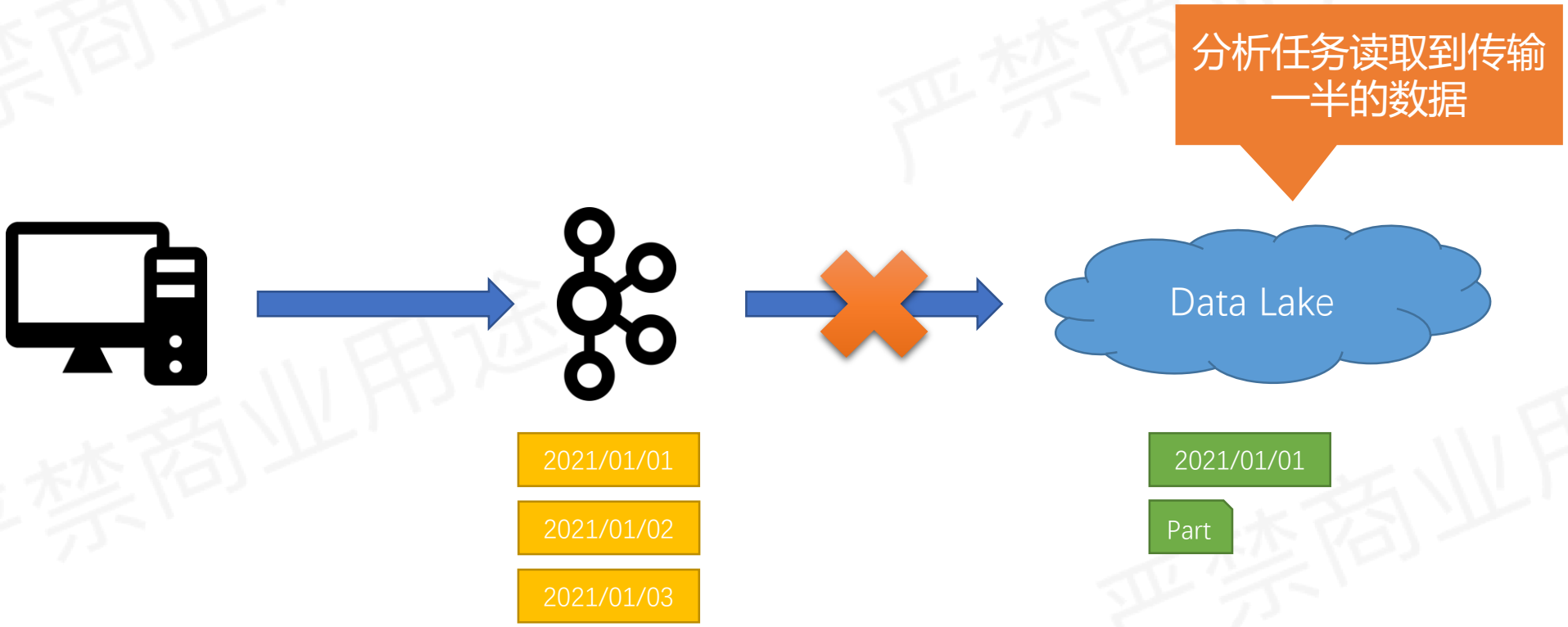


Apache Flink

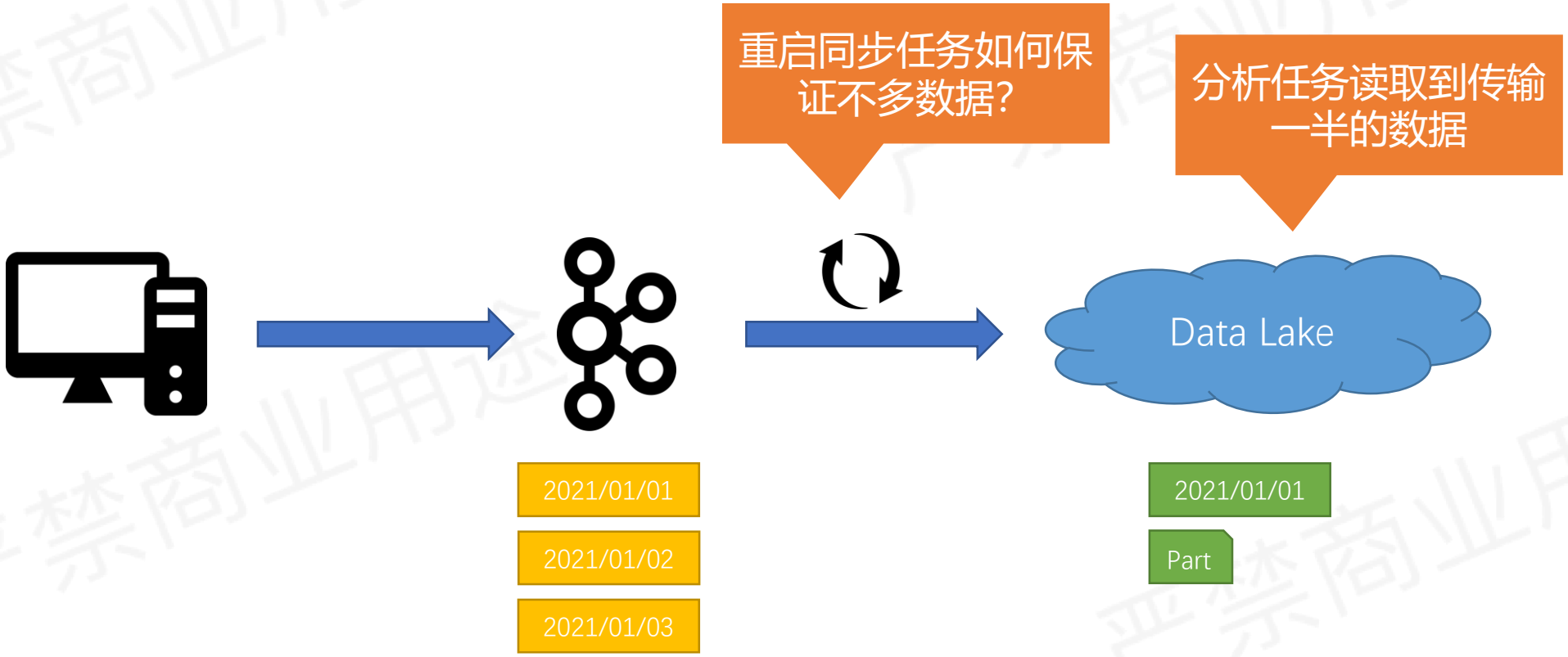
Case #1: 程序BUG导致数据传输中断



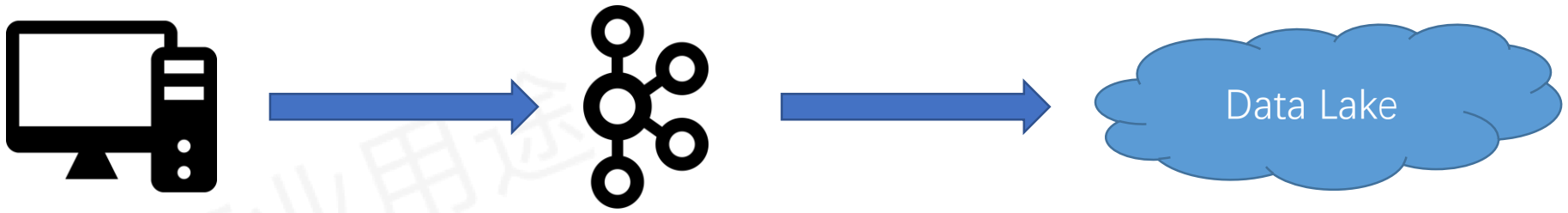
Case #1: 程序BUG导致数据传输中断



Case #1: 程序BUG导致数据传输中断

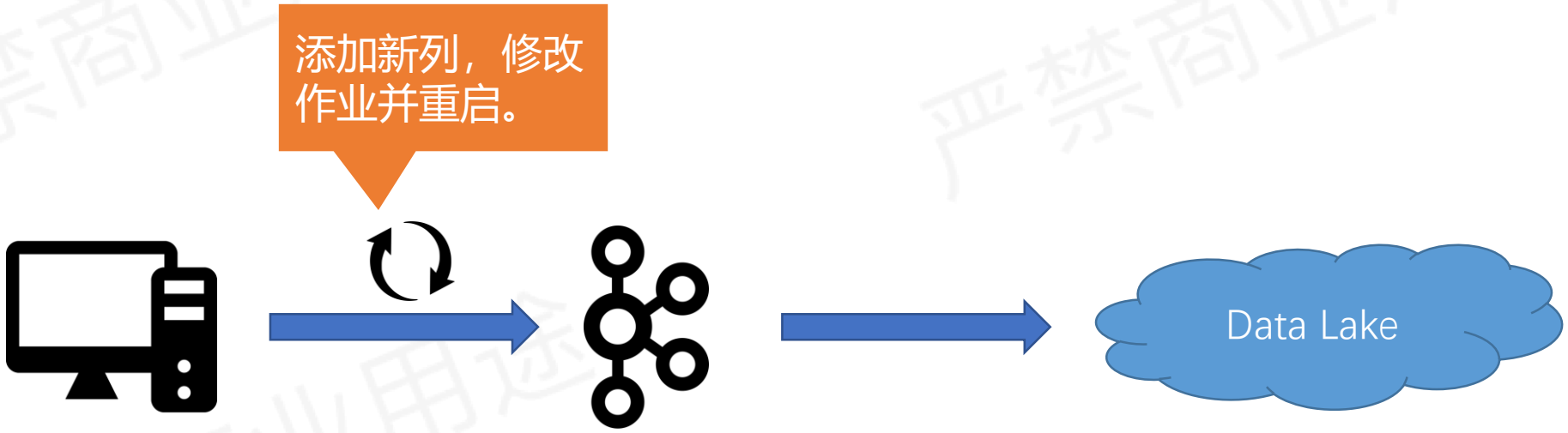


Case #2: 数据变更太痛苦了



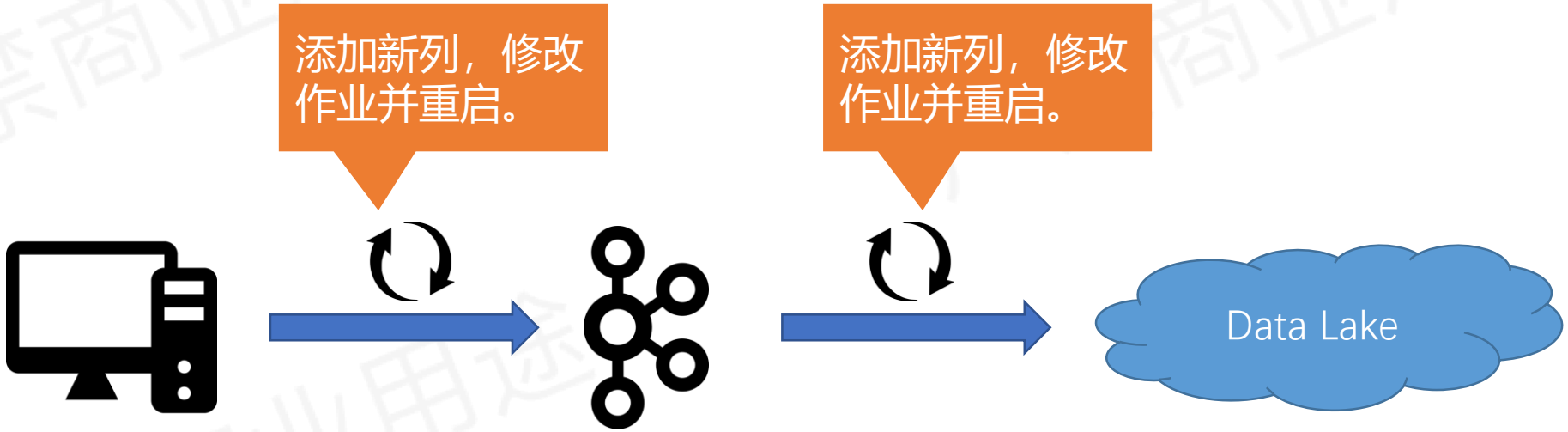
ID	NAME
1001	Alex
1002	Tom

Case #2: 数据变更太痛苦了



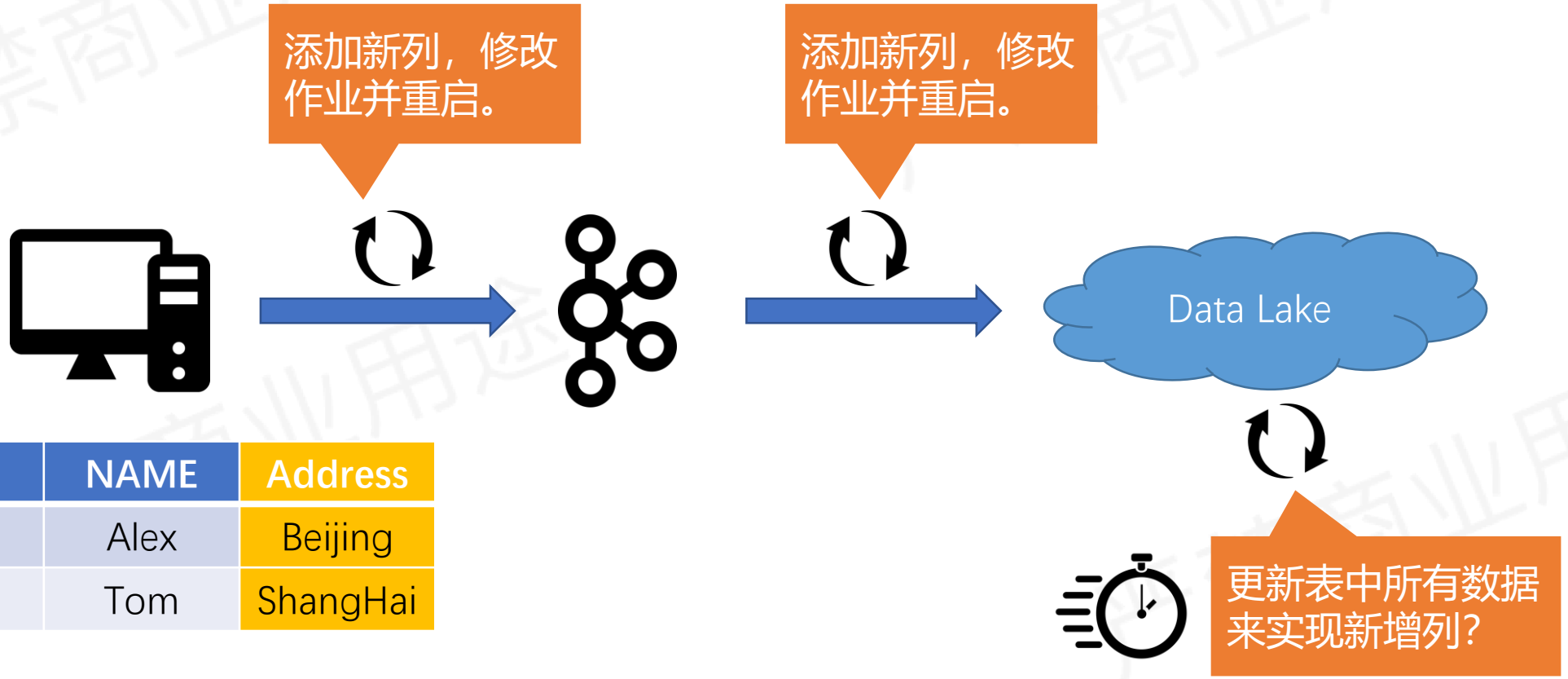
ID	NAME	Address
1001	Alex	Beijing
1002	Tom	ShangHai

Case #2: 数据变更太痛苦了



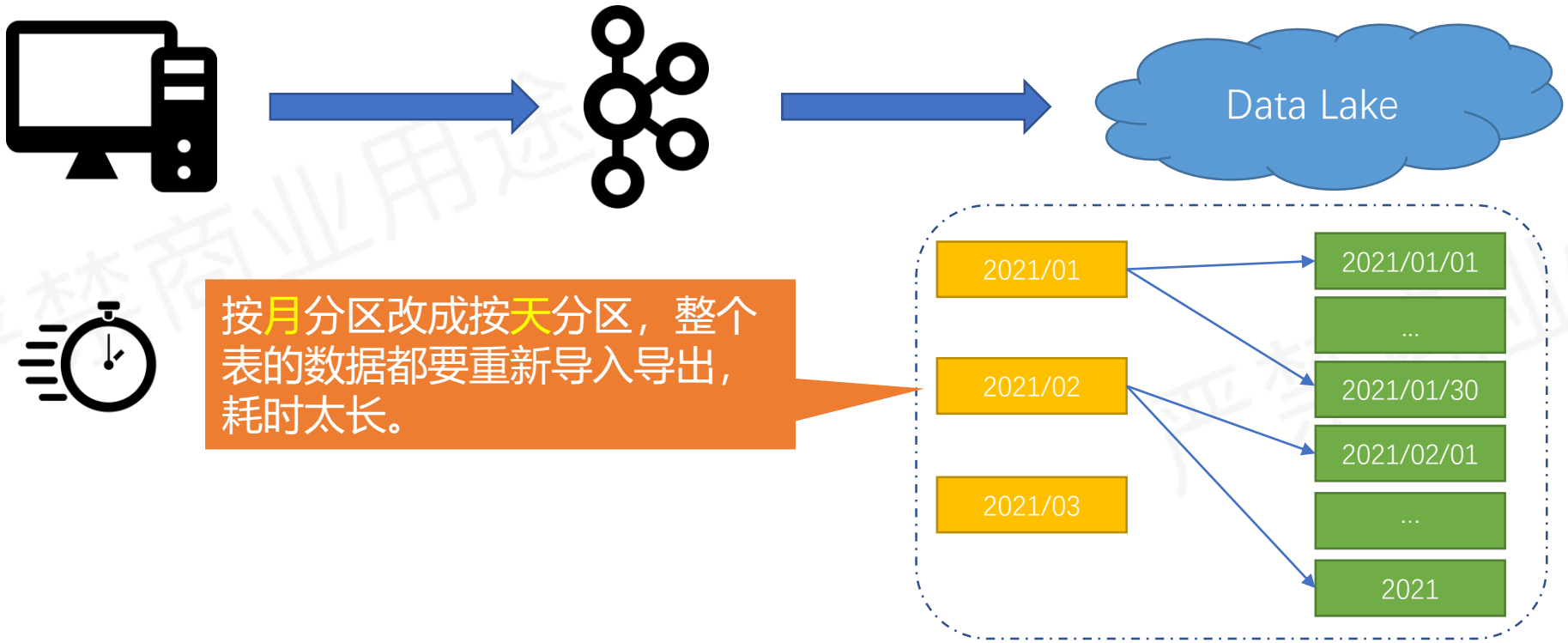
ID	NAME	Address
1001	Alex	Beijing
1002	Tom	ShangHai

Case #2: 数据变更太痛苦了



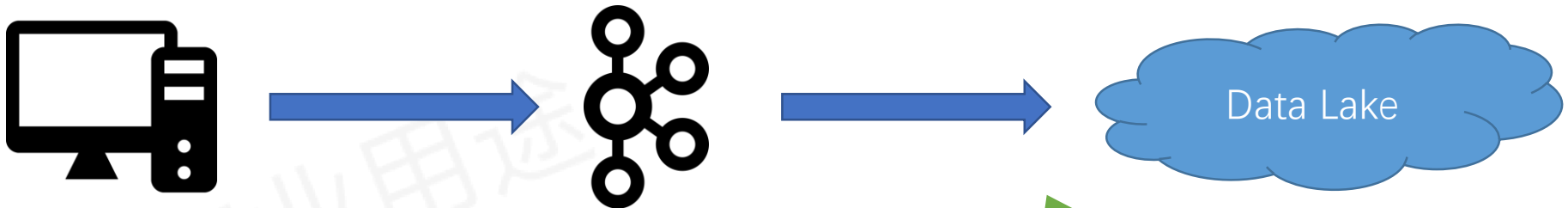
ID	NAME	Address
1001	Alex	Beijing
1002	Tom	ShangHai

Case #2: 数据变更太痛苦了



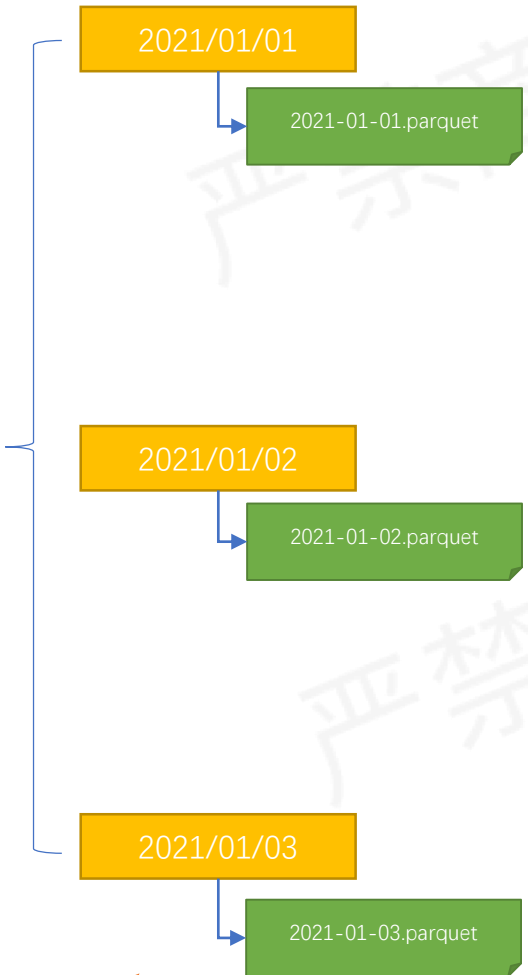
按月分区改成按天分区，整个表的数据都要重新导入导出，耗时太长。

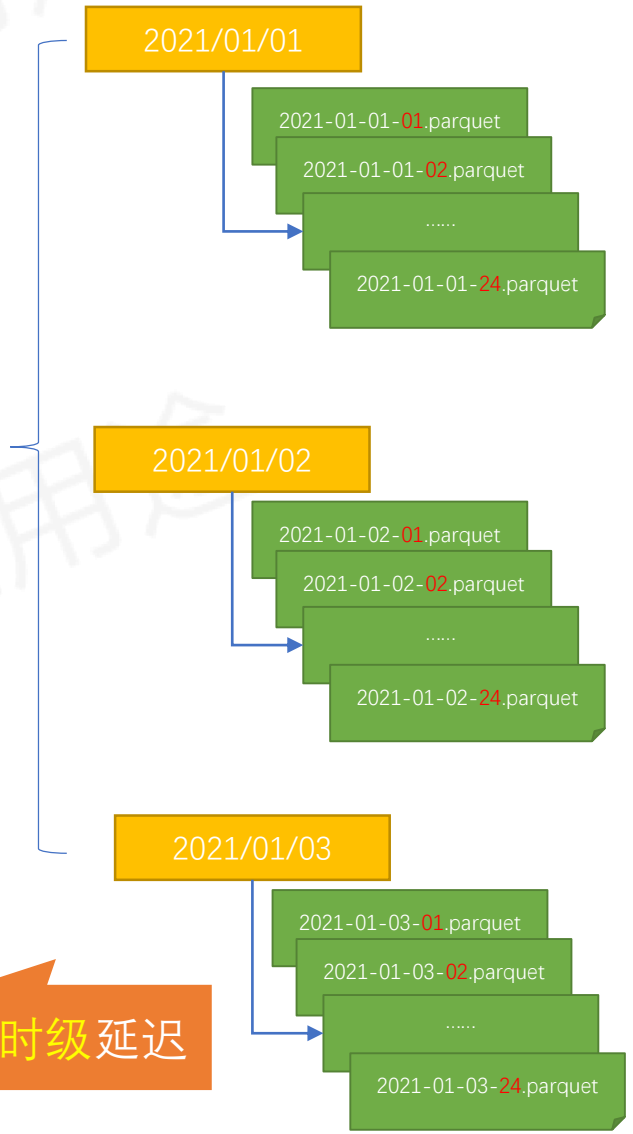
Case #3:越来越慢的近实时报表?

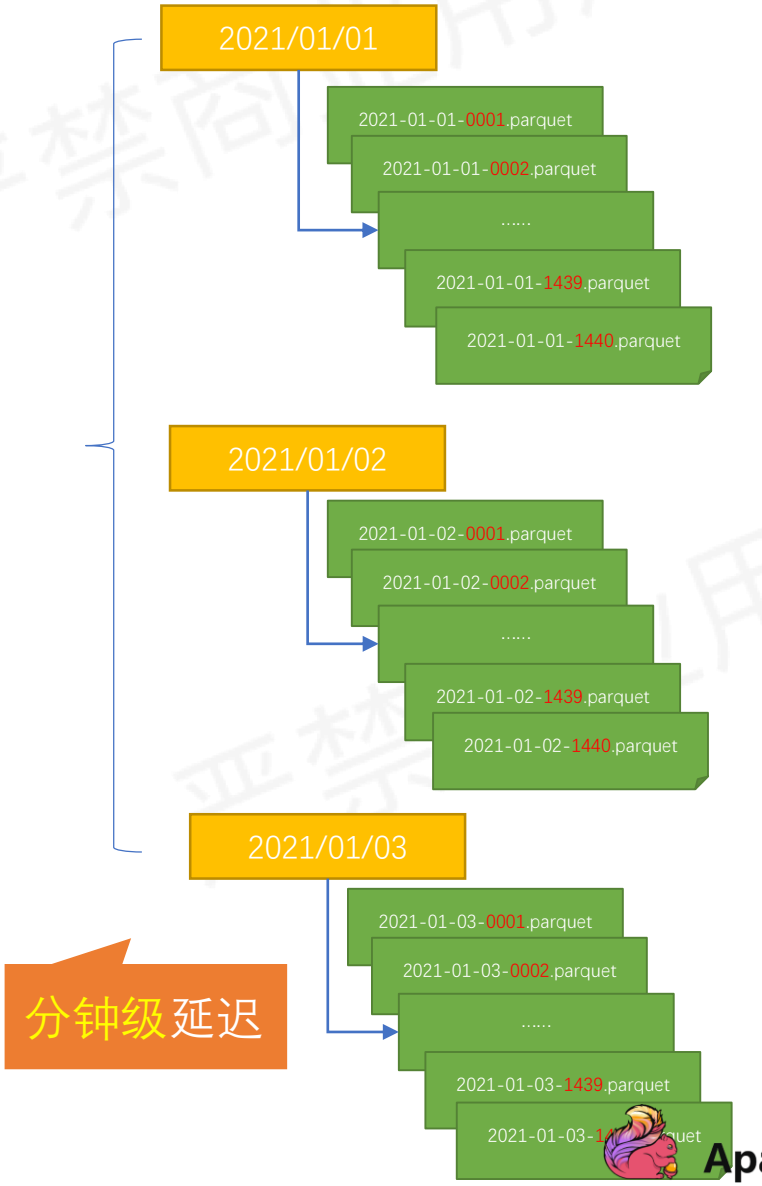


为了实现更实时的报表，一般会把数据导入周期从天级改成小时级甚至分钟级。

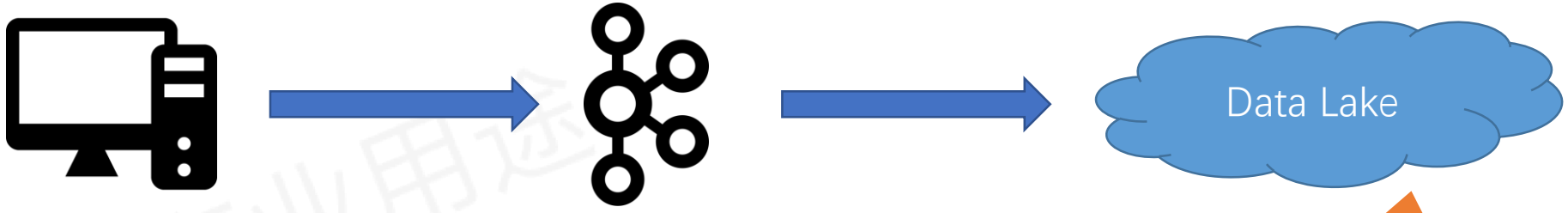
Case #3:越来越慢的近实时报表?







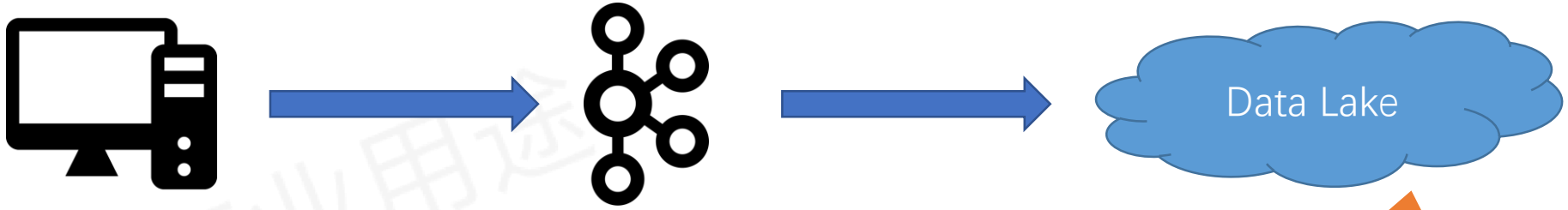
Case #3:越来越慢的近实时报表?



1. 启动分析作业越来越慢
2. Hive Metastore面临扩展难题

随着小文件越来越多，中心化的metadata瓶颈越来越严重。

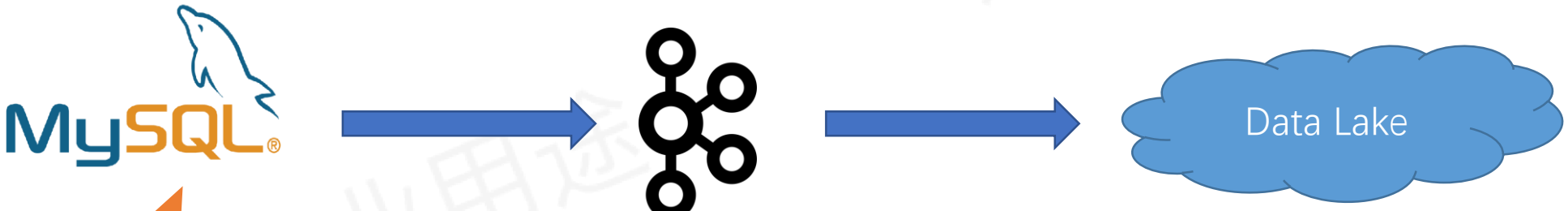
Case #3:越来越慢的近实时报表?



小文件越来越多，导致单个扫描Task频繁地在多个Datanode之间切换，扫描越来越慢。

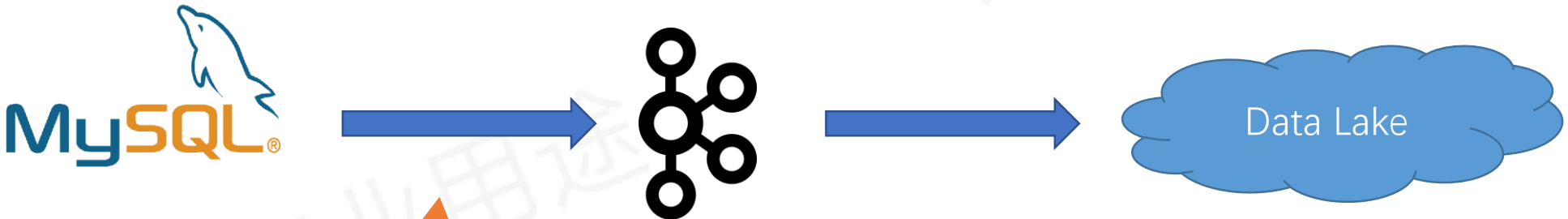
1. 分析作业扫描越来越慢

Case #4: 实时地分析CDC数据很困难



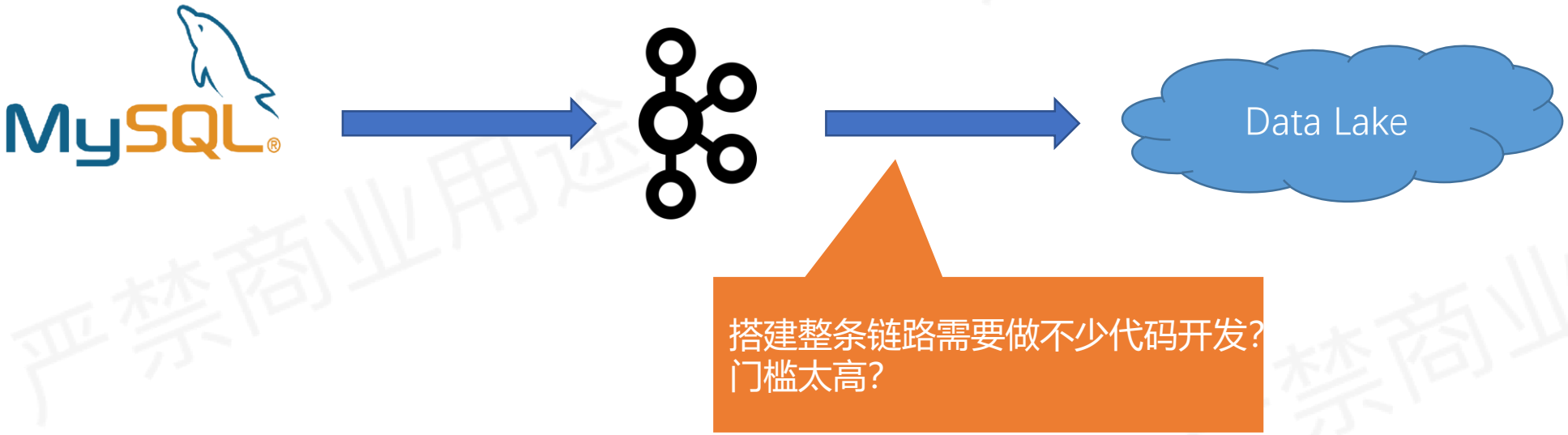
如何完美地同步全量和增量数据到数据湖中?

Case #4: 实时地分析CDC数据很困难

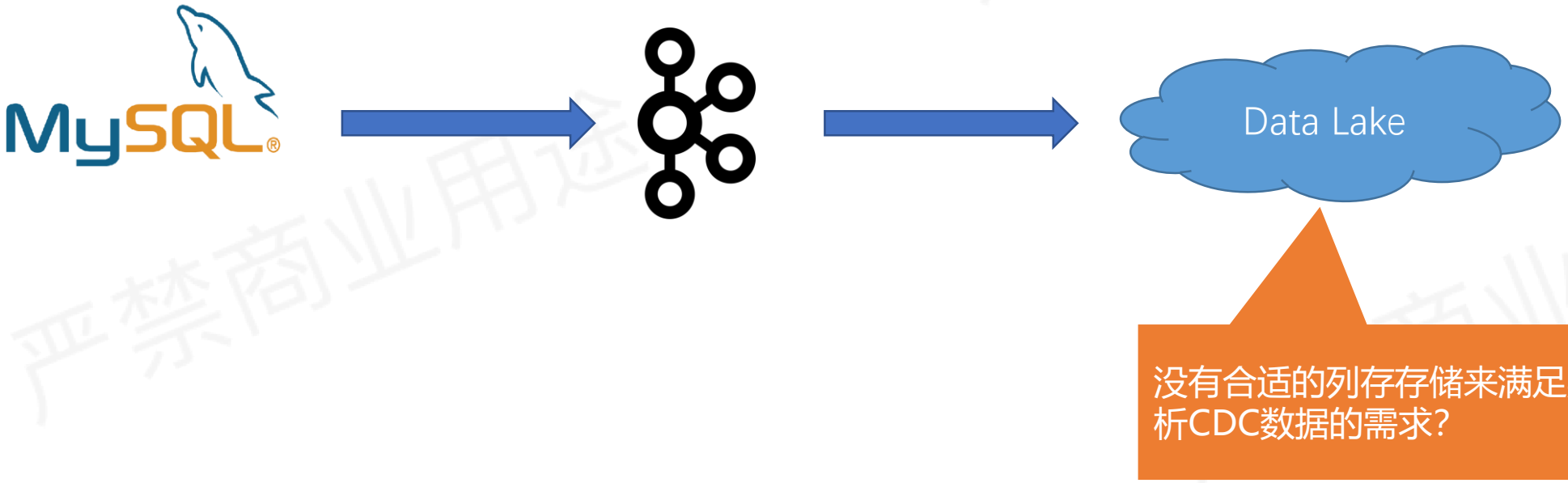


在同步过程中，如何保证Binlog一行不少地同步到下游？（即使中间碰到异常）

Case #4: 实时地分析CDC数据很困难



Case #4: 实时地分析CDC数据很困难



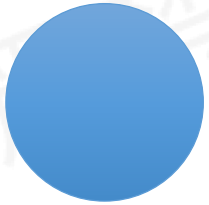
数据入湖面临的核心挑战

数据同步任务中断

无法有效隔离写入对分析的影响；
同步任务不保证exactly-once语义。

端到端数据变更

DDL导致全链路更新升级复杂；
修改湖/仓中存量数据困难。



越来越慢的近实时报表

频繁写入产生大量小文件；
Metadata系统压力大, 启动作业慢；
大量小文件导致数据扫描慢。

无法近实时分析CDC数据

难以完成全量到增量同步的切换；
涉及端到端的代码开发，门槛高；
开源界缺乏高效的存储系统。

#2 Apache Iceberg 介绍

Netflix: Hive上云痛点总结

数据变更和回溯困难

1、不提供ACID语义。在发生数据改动时，很难隔离对分析任务的影响。典型操作如：INSERT OVERWRITE；修改数据分区；修改Schema。

2、无法处理多个数据改动者造成冲突问题。

3、无法有效回溯历史版本。

替换HDFS为S3困难

1、数据访问接口直接依赖HDFS API。

2、依赖RENAME接口的原子性，这在类似S3这样的对象存储上很难实现同样的语义。

3、大量依赖文件目录的list接口，这在对象存储系统上很低效。

太多细节问题

1、Schema变更时，不同文件格式行为不一致。不同 FileFormat 甚至连数据类型的支持都不一致。

2、Metastore仅维护partition级别的统计信息，造成不task plan开销；Hive Metastore难以扩展。

3、非partition字段不能做partition prune。

Apache Iceberg核心特性

通用化标准设计

完美解耦计算引擎
Schema标准化
开放的数据格式
支持Java和Python

完善的Table语义

Schema定义与变更
灵活的Partition策略
ACID语义
Snapshot语义

丰富的数据管理

存储的流批统一
可扩展的META设计支持
批更新和CDC
支持文件加密

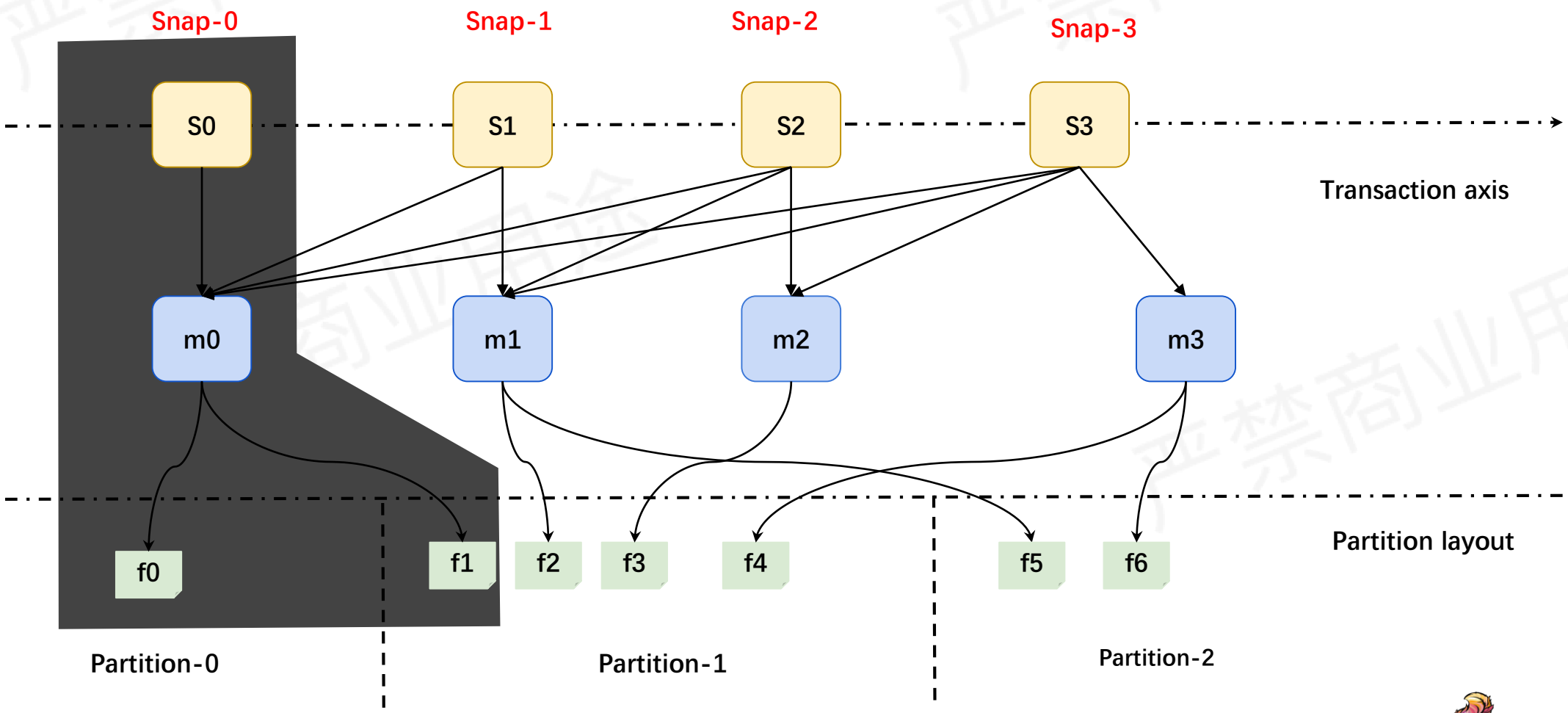
性价比

计算下推设计
低成本的元数据管理
向量化计算
轻量级索引

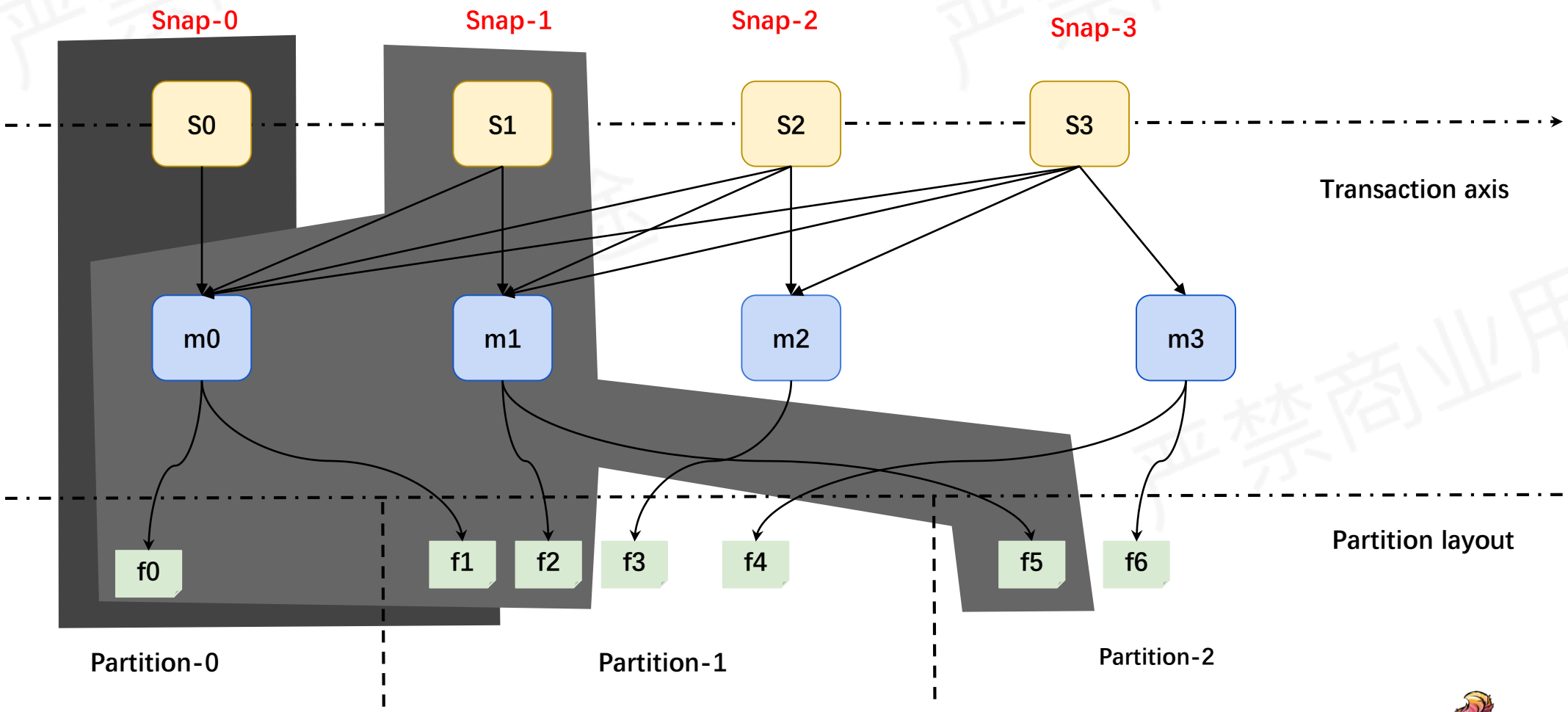
Apache Iceberg File Layout

Apache Flink

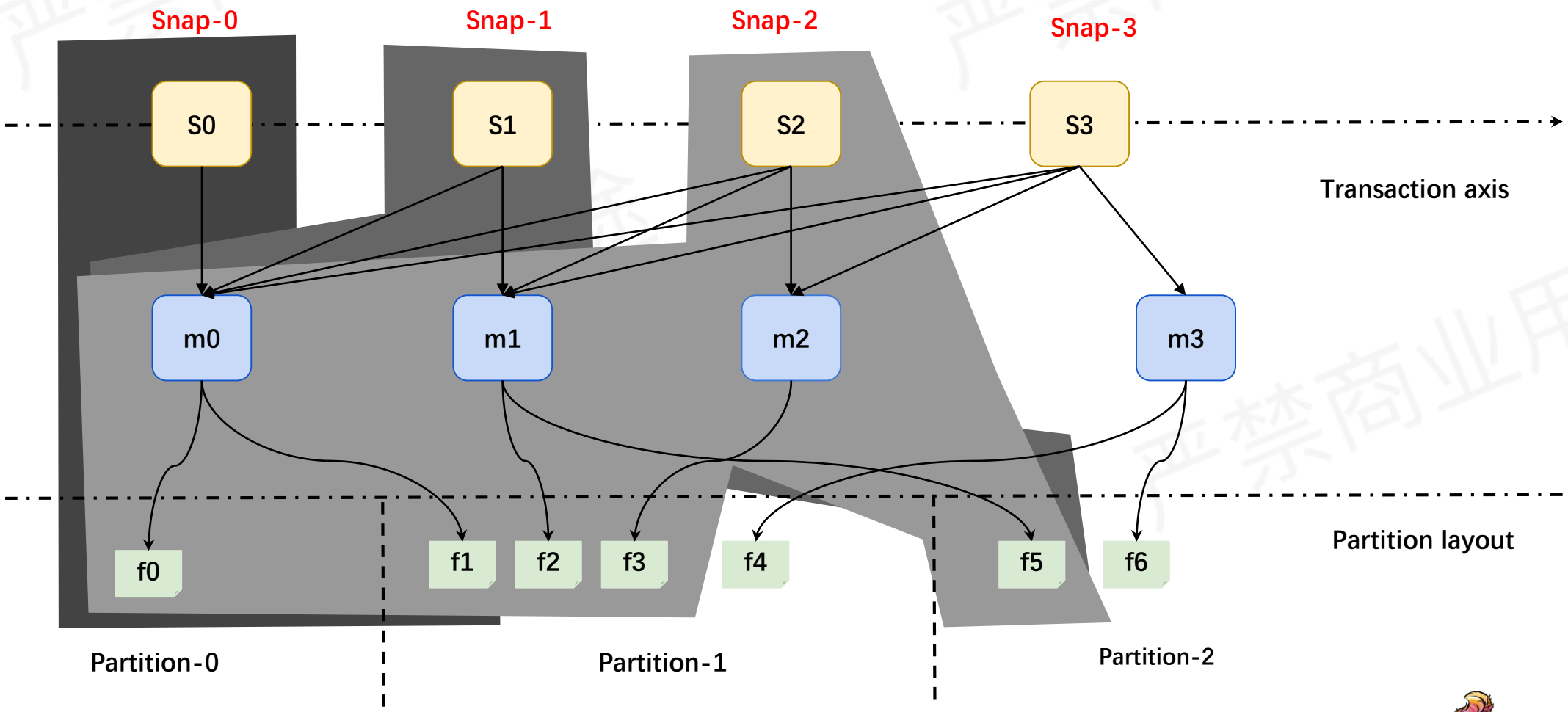
Apache Iceberg Snapshot View



Apache Iceberg Snapshot View

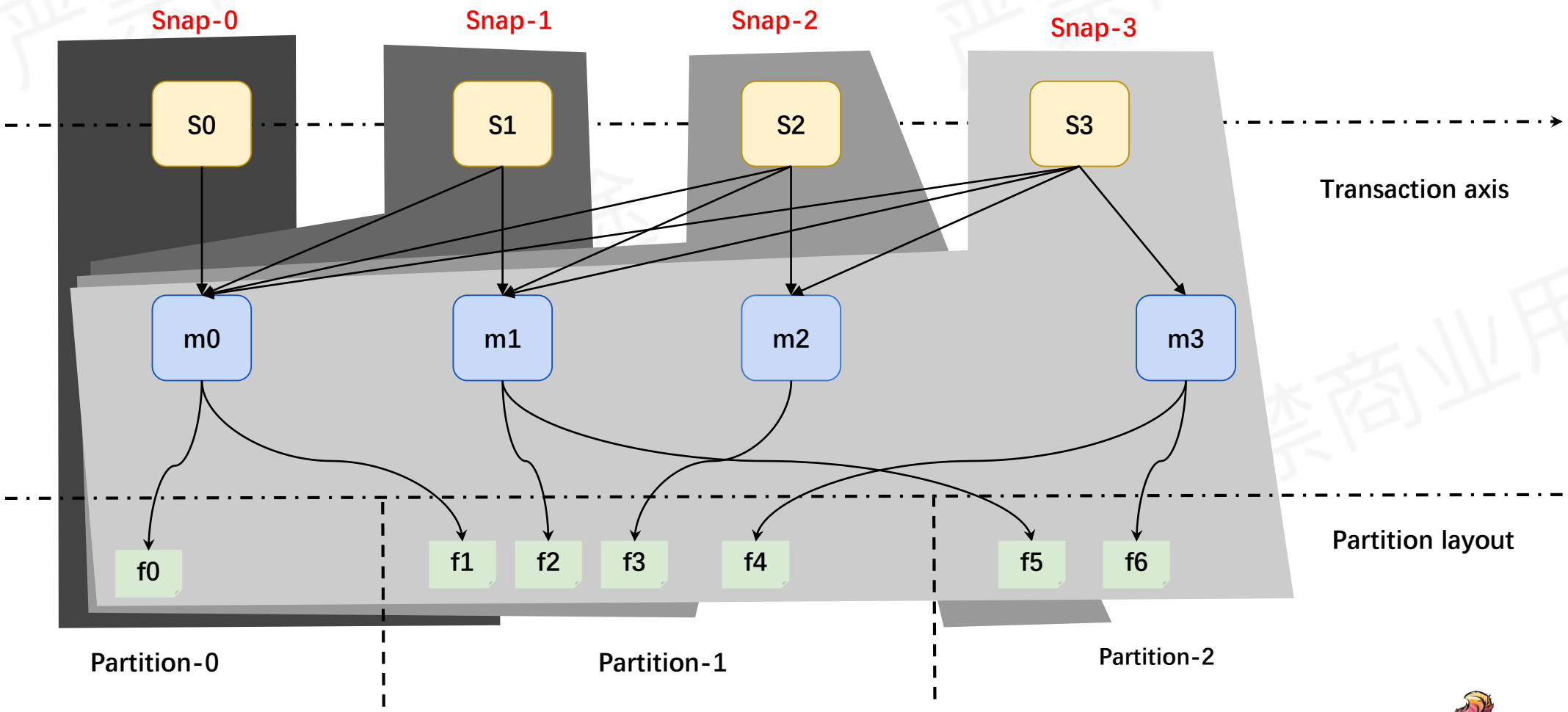


Apache Iceberg Snapshot View



Apache Flink

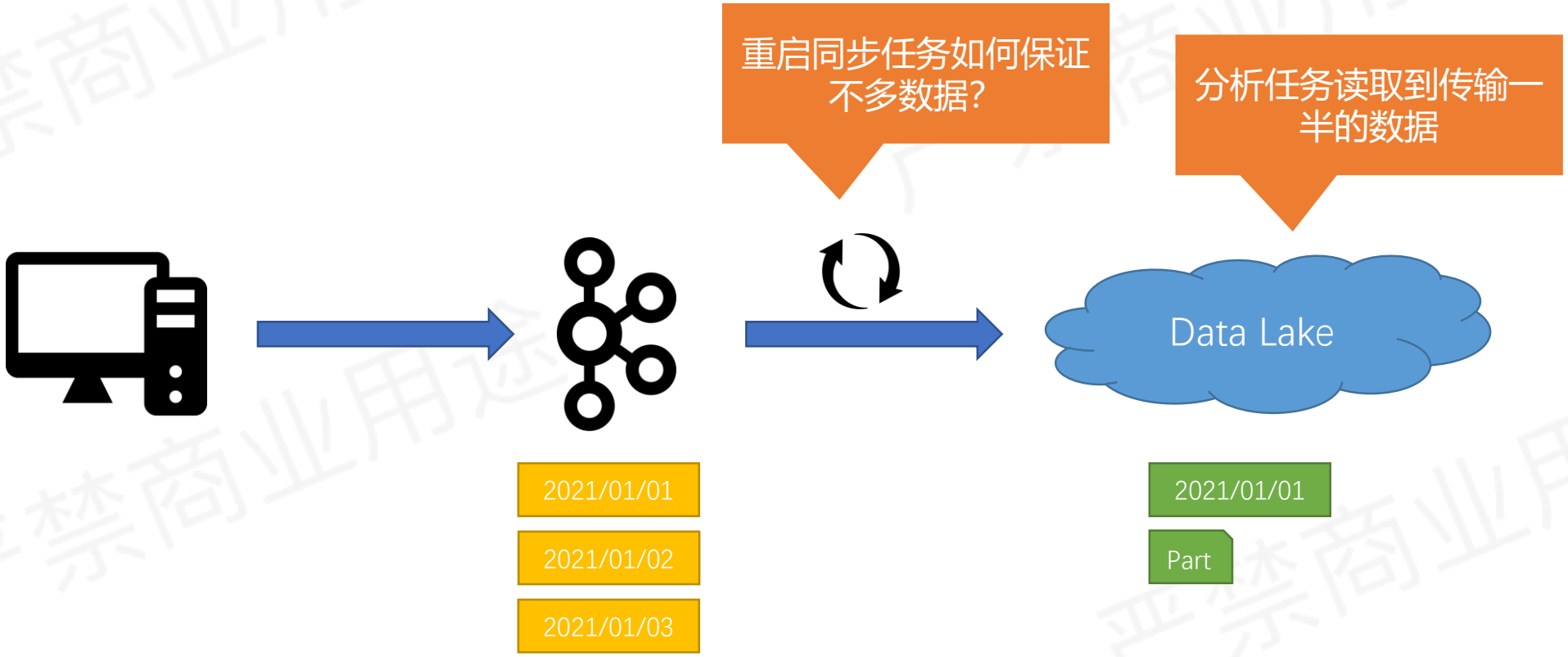
Apache Iceberg Snapshot View



选择Apache Iceberg的公司

#3 Flink 和 Iceberg 如何解决问题

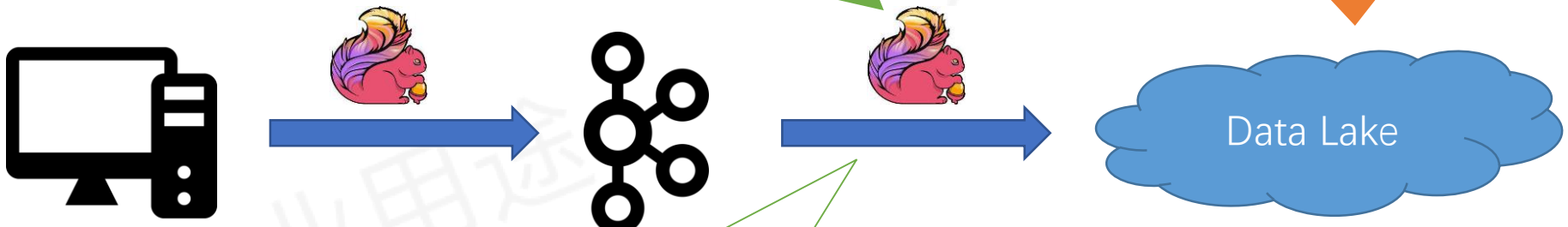
Case #1: 程序BUG导致数据传输中断

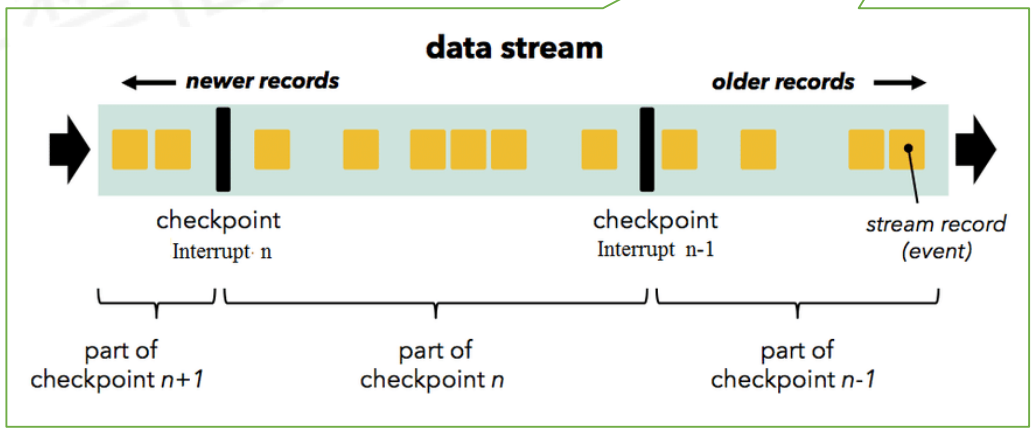


Case #1: 程序BUG导致数据传输中断

借助Flink实现的exactly once语义和故障恢复能力，实现数据严格一致性。

分析任务读取到传输一半的数据



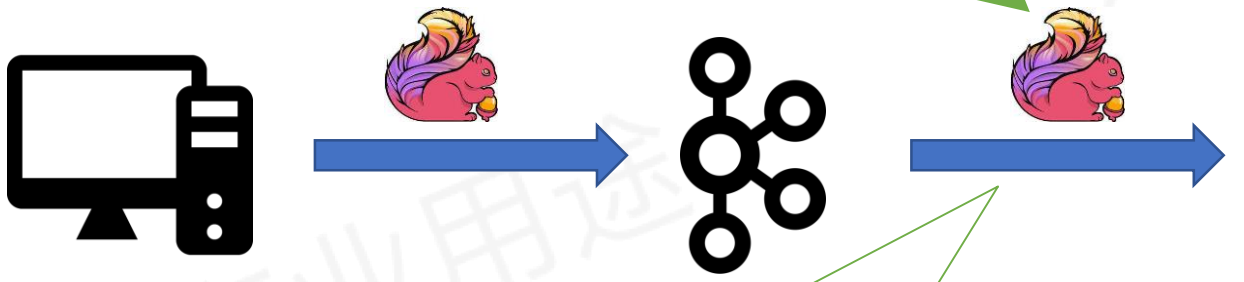


Apache Flink

Case #1: 程序BUG导致数据传输中断

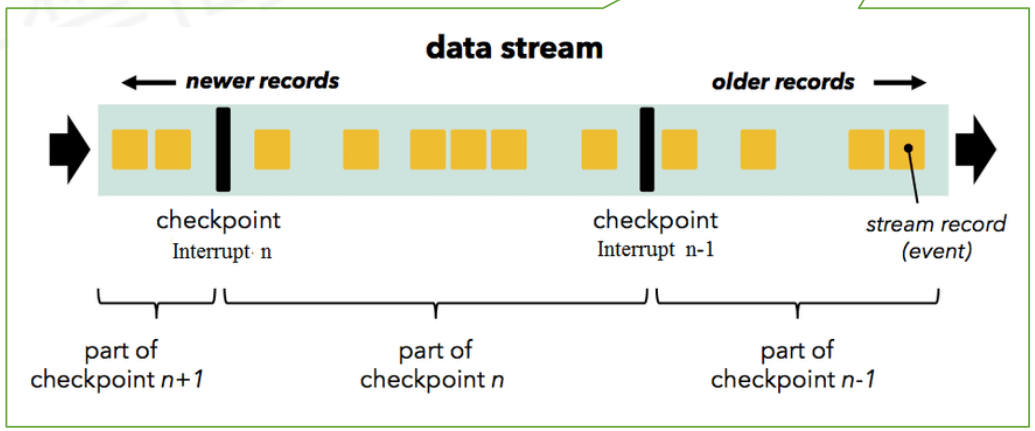
借助Flink实现的exactly once语义和故障恢复能力，实现数据严格一致性。

借助Iceberg ACID能力来隔离写入对分析任务的不利影响。



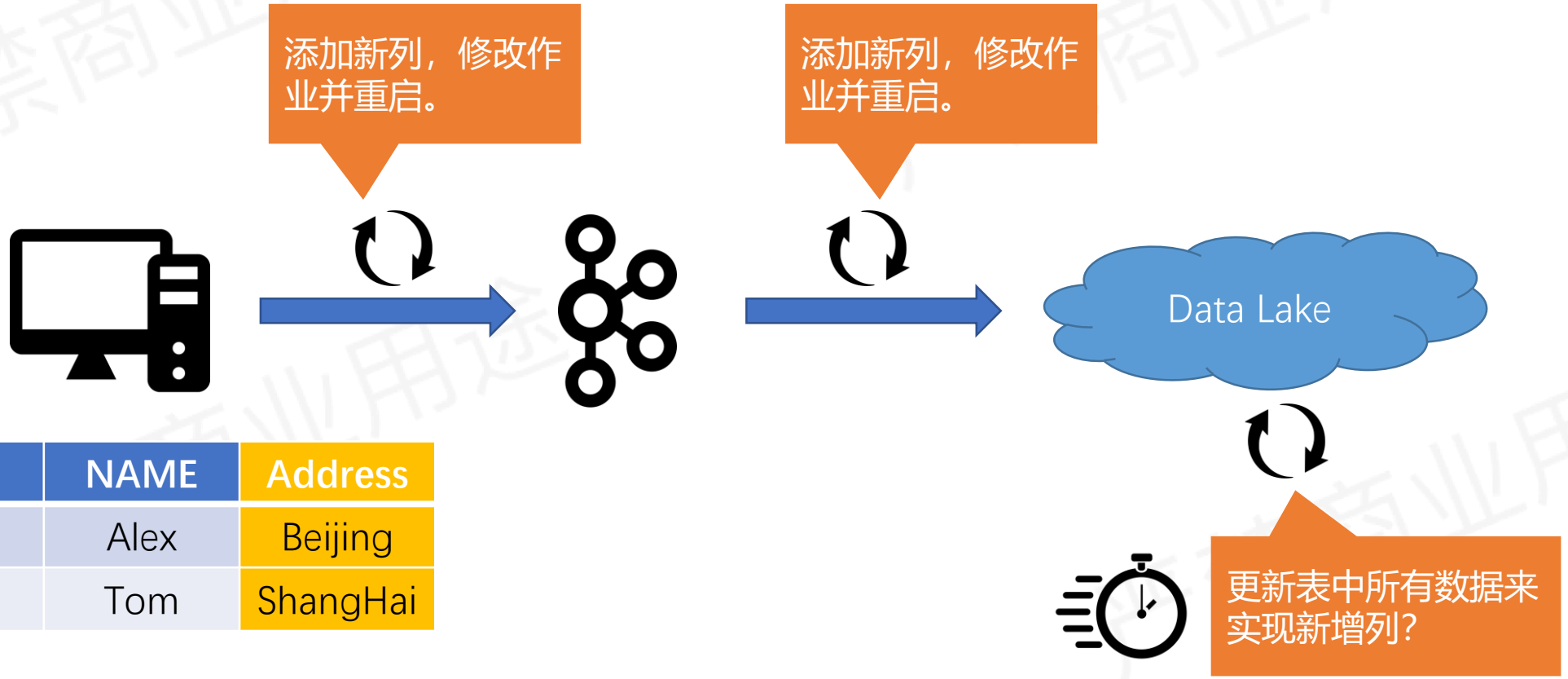
Data Lake

ICEBERG



Apache Flink

Case #2: 数据变更太痛苦了

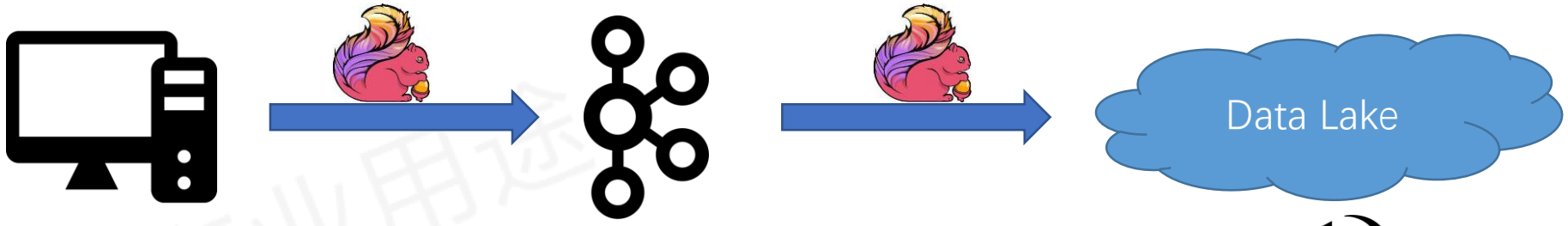


ID	NAME	Address
1001	Alex	Beijing
1002	Tom	ShangHai

Case #2: 数据变更太痛苦了

捕捉数据源的Schema变更事件，并同步到下游。

添加新列，修改作业并重启。



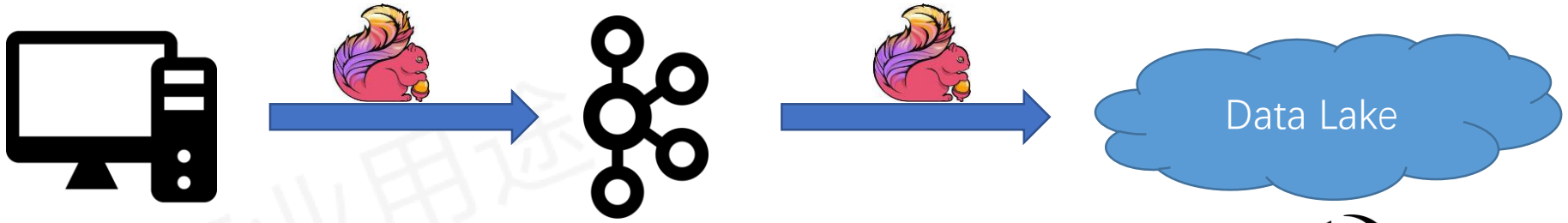
ID	NAME	Address
1001	Alex	Beijing
1002	Tom	ShangHai

更新表中所有数据来实现新增列?

Case #2: 数据变更太痛苦了

捕捉数据源的Schema变更事件，并同步到下游。

直接转发DDL事件到下游即可。



ID	NAME	Address
1001	Alex	Beijing
1002	Tom	ShangHai

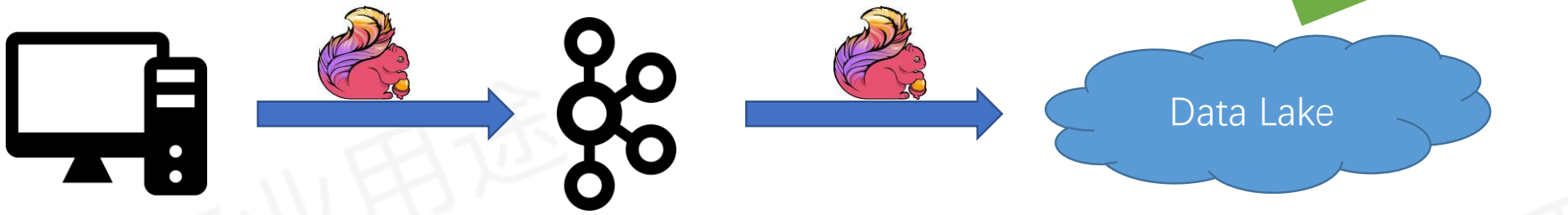
更新表中所有数据来实现新增列?

Case #2: 数据变更太痛苦了

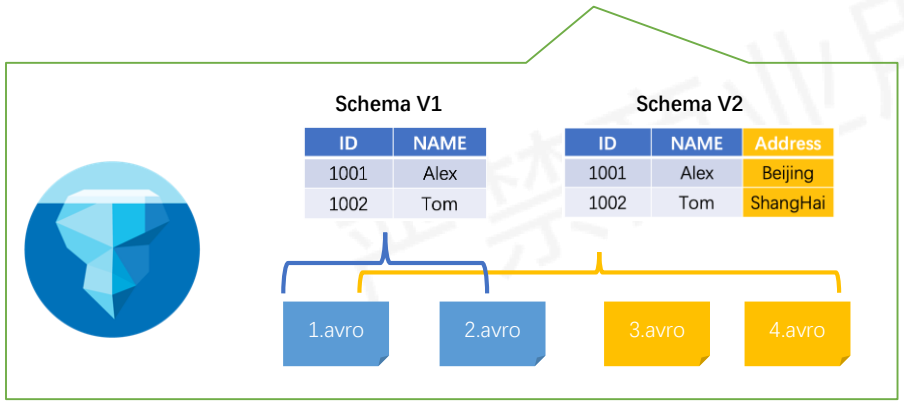
捕捉数据源的Schema变更事件，并同步到下游。

直接转发DDL事件到下游即可。

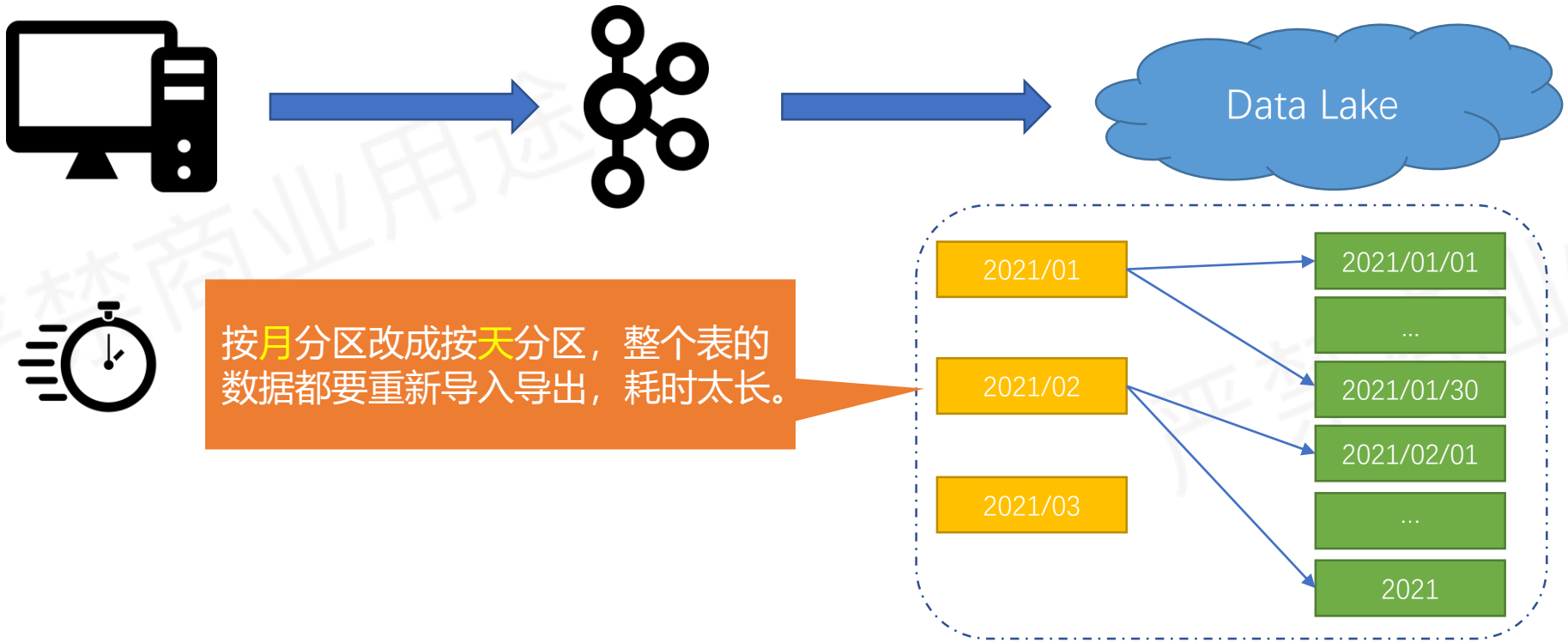
Iceberg通过多版本机制瞬间完成schema变更同步。



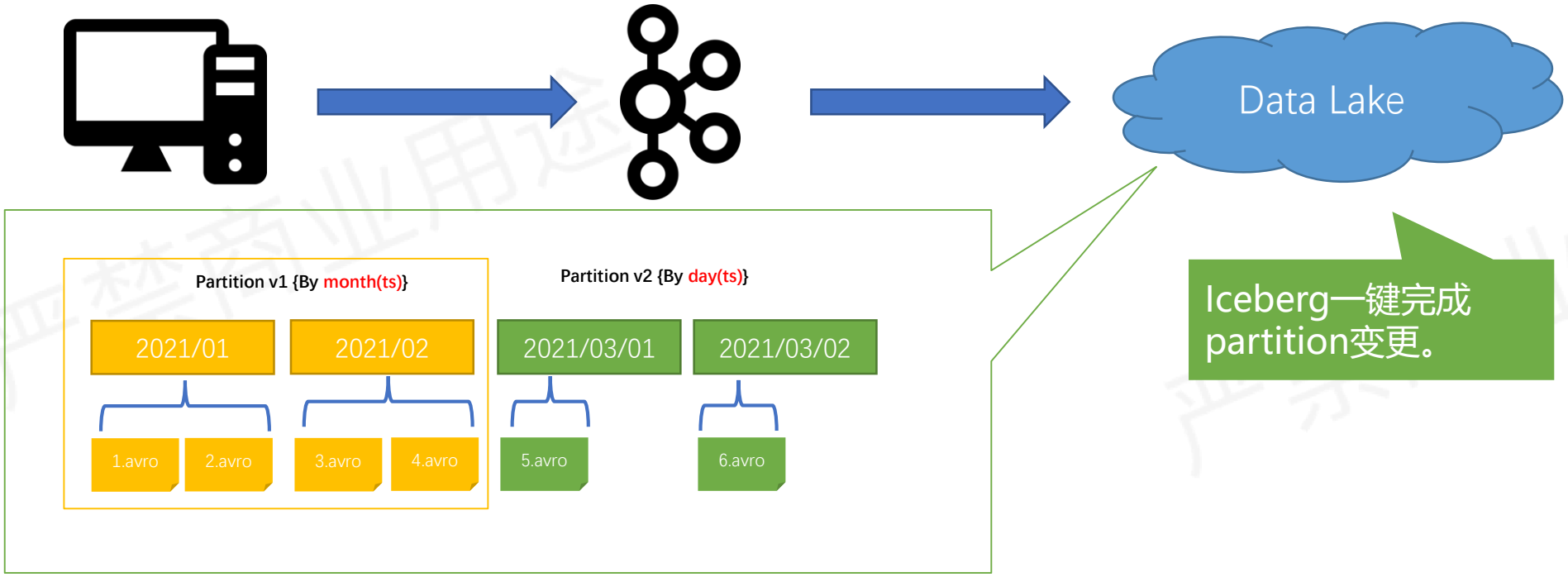
ID	NAME	Address
1001	Alex	Beijing
1002	Tom	ShangHai



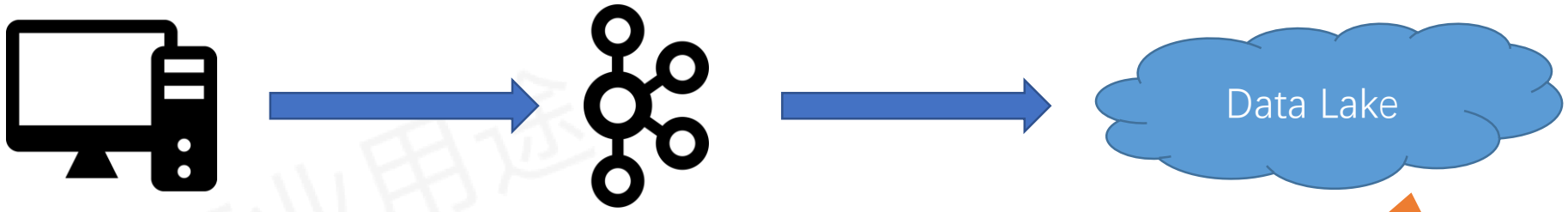
Case #2: 数据变更太痛苦了



Case #2: 数据变更太痛苦了



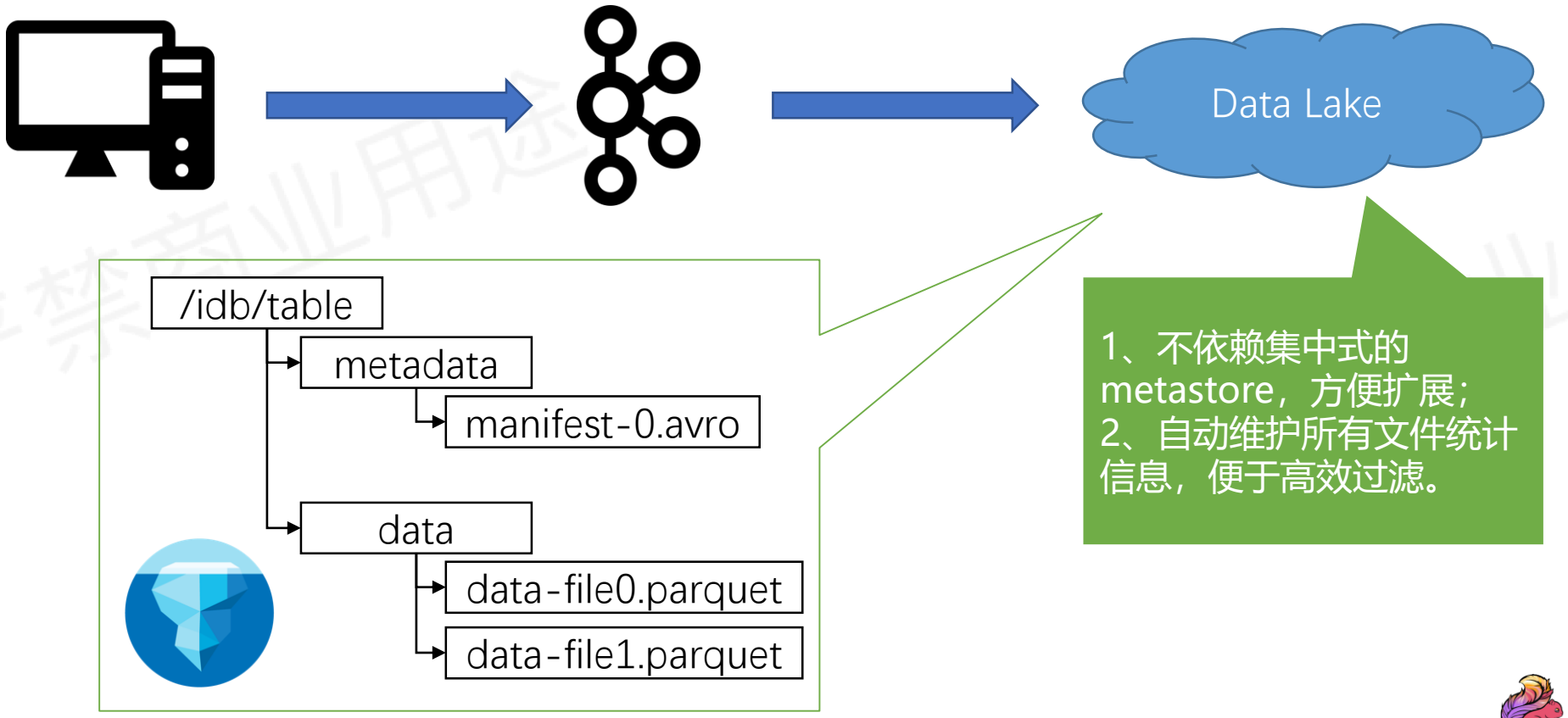
Case #3:越来越慢的近实时报表?



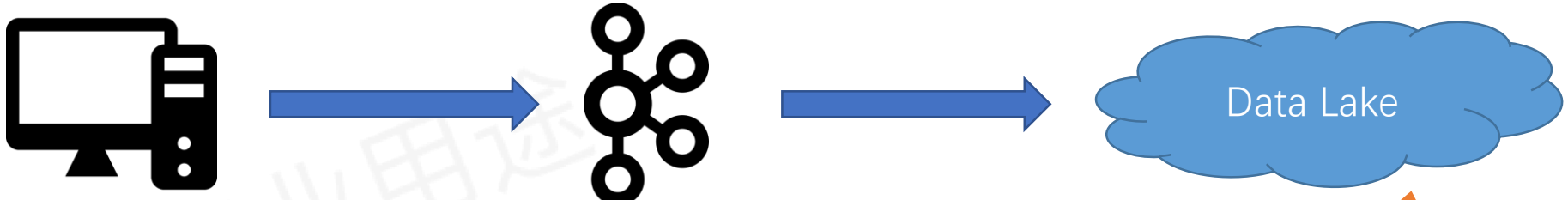
1. 启动分析作业越来越慢
2. Hive Metastore面临扩展难题

随着小文件越来越多，中心化的metadata瓶颈越来越严重。

Case #3:越来越慢的近实时报表?



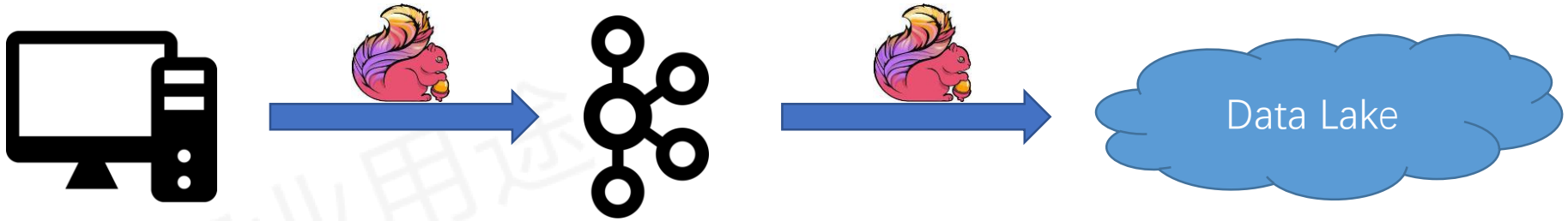
Case #3:越来越慢的近实时报表?

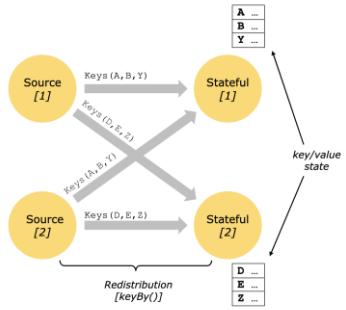


1. 分析作业扫描越来越慢

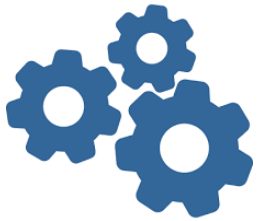
小文件越来越多，导致单个扫描Task频繁地在多个Datanode之间切换，扫描越来越慢。

Case #3:越来越慢的近实时报表?





1. 按照Bucket来Shuffle方式写入

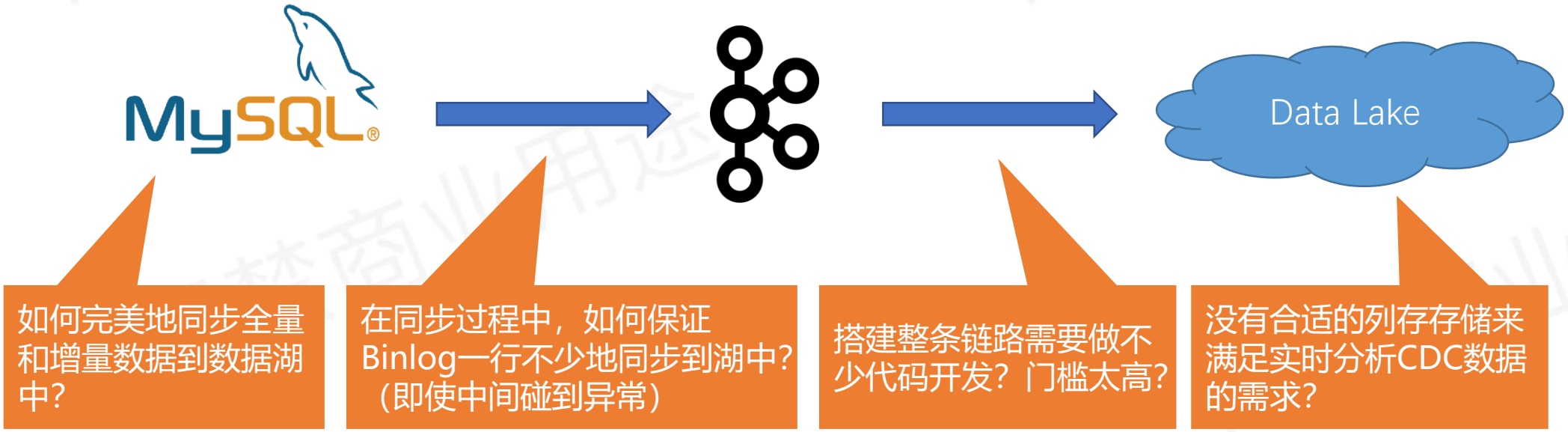


2. 批作业定期合并小文件

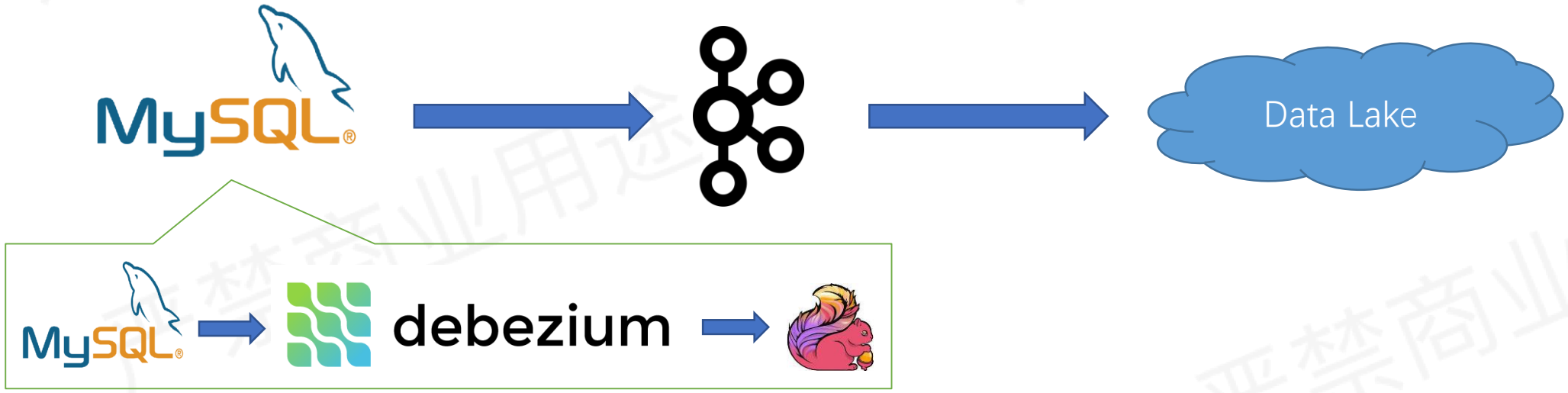
AutOptimize

3. 自动增量地合并小文件

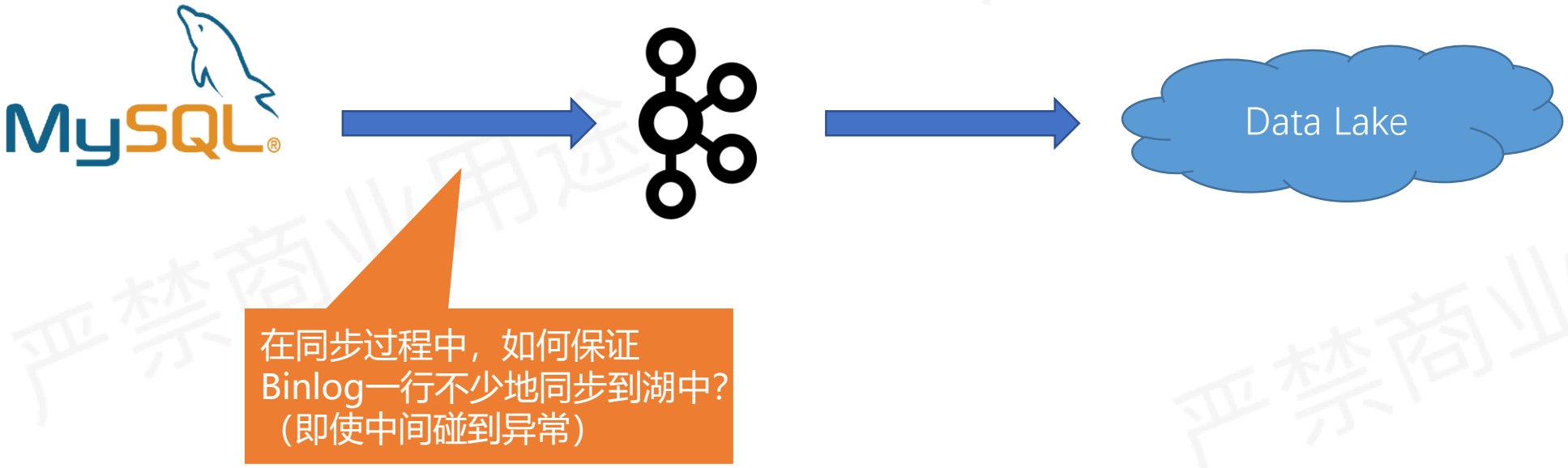
Case #4: 实时地分析CDC数据很困难



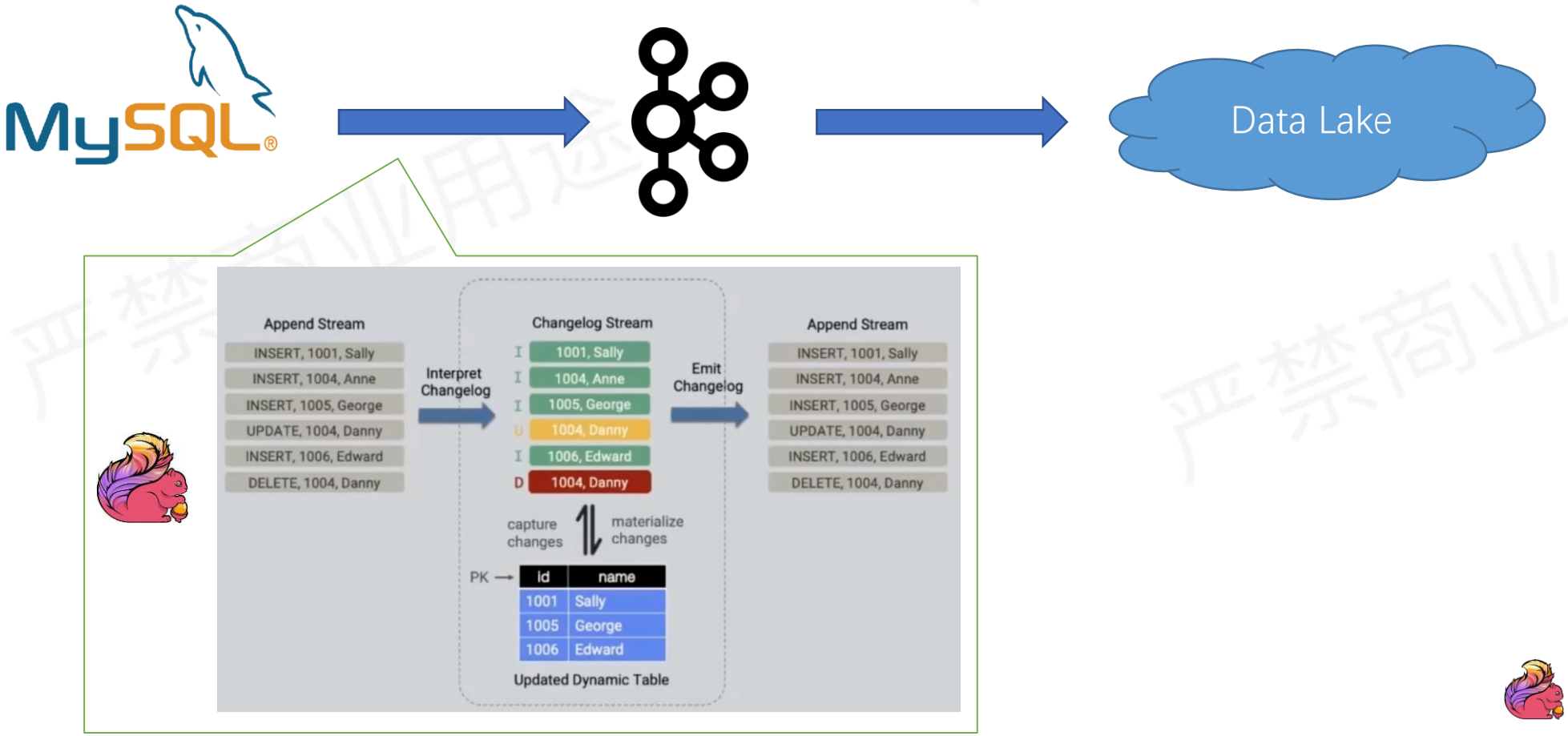
Case #4: 实时地分析CDC数据很困难



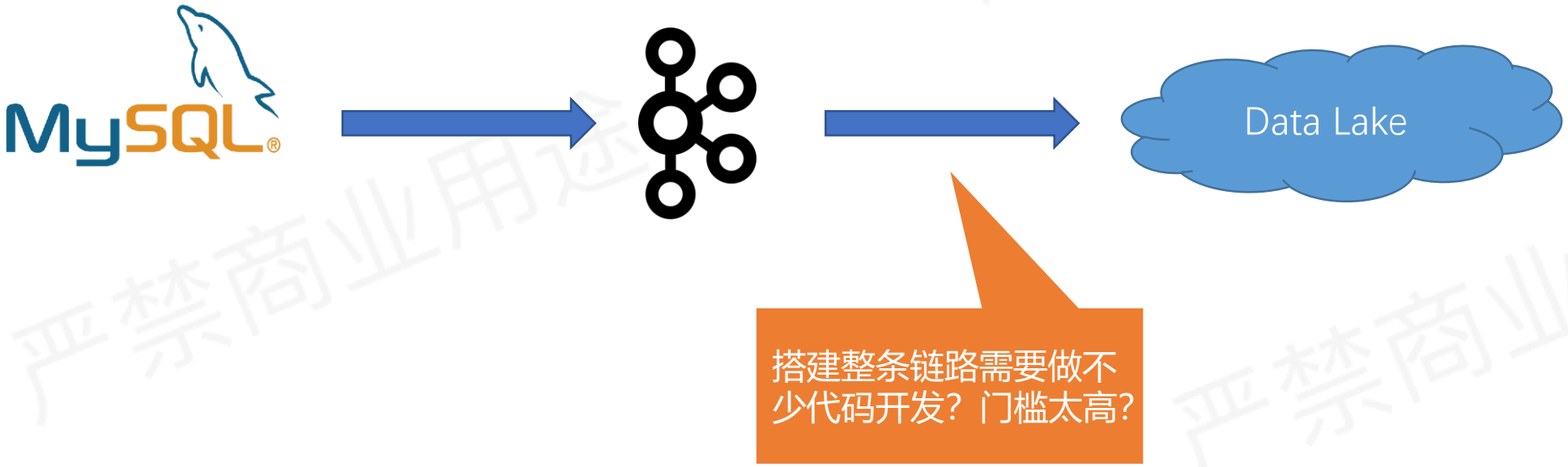
Case #4: 实时地分析CDC数据很困难



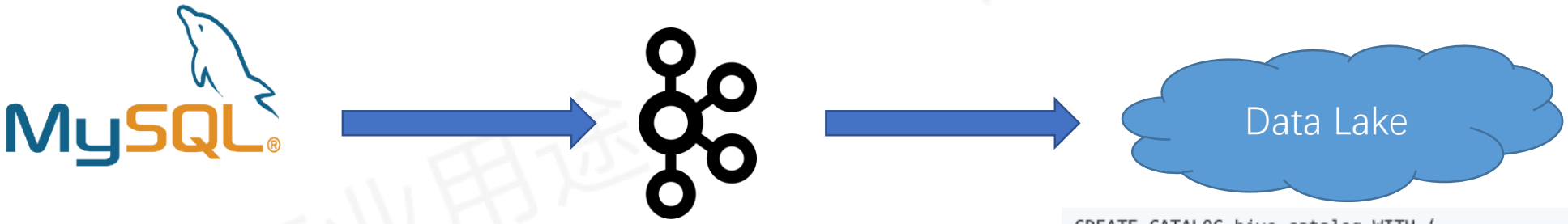
Case #4: 实时地分析CDC数据很困难



Case #4: 实时地分析CDC数据很困难



Case #4: 实时地分析CDC数据很困难




```
CREATE TABLE sbtest1(
  `id`  INT NOT NULL,
  `k`   INT NOT NULL,
  `c`   CHAR(120) NOT NULL,
  `pad` char(60) NOT NULL
) WITH (
  'connector' = 'mysql-cdc',
  'hostname' = 'localhost',
  'port' = '3306',
  'username' = '<your-mysql-user>',
  'password' = '<your-mysql-password>',
  'database-name' = 'test',
  'table-name' = 'sbtest1'
);
```

第一步：定义flink source表

```
INSERT INTO iceberg_sbtest1 SELECT * FROM sbtest1 ;
```

第三步：启动 Flink 作业导入数据

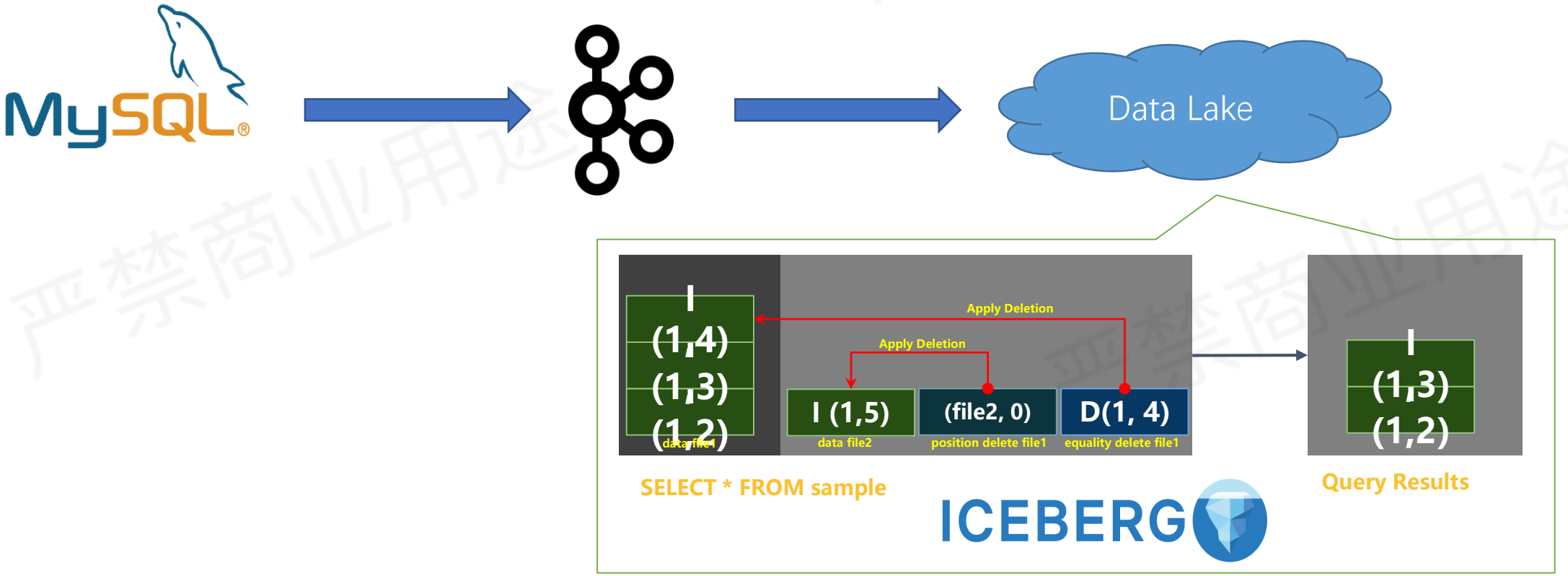
```
CREATE CATALOG hive_catalog WITH (
   'type'='iceberg',
   'catalog-type'='hive',
   'uri'='thrift://localhost:9083',
   'clients'='5',
   'property-version'='1',
   'warehouse'='file:///Users/openinx/test/iceberg-warehouse'
);
USE hive_catalog;

CREATE DATABASE mysql_db;
USE mysql_db;

CREATE TABLE iceberg_sbtest1(
  `id`  INT NOT NULL,
  `k`   INT NOT NULL,
  `c`   CHAR(120) NOT NULL,
  `pad` char(60) NOT NULL,
  PRIMARY KEY(id) NOT ENFORCED
);
```

第二步：定义iceberg sink 表

Case #4: 实时地分析CDC数据很困难



#4 Apache Iceberg Roadmap

Apache Iceberg Roadmap

Apache Iceberg	Core Features
Apache Iceberg 0.7.0	Release 2019/10/26 1. Support Spark 2.4/Presto, Python, Parquet/Avro File Format, File Encryption.
Apache Iceberg 0.8.0	Release 2020/05/07 1. Support ORC File Format. 2. Incremental scan API. 3. Write data in MapReduce.
Apache Iceberg 0.9.0	Release 2020/07/14 (Graduated as Apache Top Level Project) 1. Support Spark 3 2. Vectorized reads for flat schemas in Spark
Apache Iceberg 0.10.0	Release 2020/11/13 1. Flink Integration: Writing into iceberg table, Read in batch mode. 2. Hive Integration: Read iceberg table, filter push down etc. 3. Add Format v2.
Apache Iceberg 0.11.0	Release 2021/01/27 1. Spark 3 SQL extension: MERGE INTO, DELETE FROM, ALTER TABLE, Procedures. 2. Flink: Support filter pushdown, writing CDC, streaming reader. 3. Integration: aws s3, aws glue, nessie catalog.
Apache Iceberg 0.12.0	Release (?) 1. Apache Beam sink 2. Flink CDC/Upsert phase v2. 3. Integration: aliyun oss

Apache Iceberg Roadmap

	Apache Flink	Apache Iceberg	Powered by
Phase #1 (Connect to iceberg)	Apache Flink 1.11.0	Apache Iceberg 0.10.0 (Oct 2020) 1. Flink streaming sink 2. Flink batch sink 3. Flink batch source	1. Tencent 2. Netflix (flink+iceberg on AWS S3) 3. Apple Siri 4. Yilong.com (~ 100 iceberg tables) 5. autohome.com (Replacing hive tables)
Phase #2 (Replace hive table format)	Apache Flink 1.11.0	Apache Iceberg 0.11.0 (Jan 2021) 1. Flink source improvement - filter/limit push down 2. Flink streaming source 3. Format v2: CDC/Upsert (Phase#1) - write & read correctness data. 4. Major Compaction (Batch Mode).	1. autohome.com - CDC/Upsert POC
Phase #3 (Batch/Stream row-level delete)	Apache Flink 1.12.0	Apache Iceberg 0.12.0 (~ Apr 2021) 1. Format v2: CDC/Upsert (Phase#2) - performance & stability 2. Flink SQL imports CDC to iceberg.	
Phase #4 (More powerful data lake)	Apache Flink 1.13.0 (?) 1. SQL DDL 2. SQL time travel.	Apache iceberg 0.13.0 (?) 1. Integrate Ranger & Altas. 2. Integrate with Alluxio 3. SQL on everything (snapshots/manifests/partitions) 4. More things.	

Apache Flink

Apache Flink

我们在招聘

阿里云实时计算团队欢迎感兴趣的朋友
一起探索大数据的世界。

kete.yangkt@alibaba-inc.com

Apache Flink

Thanks

Apache Flink x Iceberg Meetup · 上海站