Apache Flink 新场景
——OLAP 引擎

贺小令
阿里巴巴技术专家

FLINK FORWARD # ASIA
实时即未来 # Real-time Is The Future

FLINK FORWARD

Contents
目录

01 背景介绍
Background

02 Apache Flink OLAP 引擎
Apache Flink OLAP Engine

03 案例介绍
User Cases Introduction

04 未来计划
Future Plans

背景介绍

Background

01

OLAP 及其分类

OLAP & Types

OLAP (online analytical processing) is a **computing method** that enables users to easily and selectively extract and query data in order to analyze it **from different points of view**.

多维 OLAP (MOLAP)

Multi-dimensional OLAP

传统的 OLAP 分析方式，
数据存储在一个多维数据集中

In traditional OLAP analysis, data is stored in a multidimensional cube.

关系型 OLAP (ROLAP)

Relational OLAP

依赖操作 DB 数据，通过 SQL 的 WHERE 条件以呈现传统 OLAP 的切片、切块功能

Relies on manipulating the data stored in the relational database to give the appearance of traditional OLAP's slicing and dicing functionality (equivalent to adding a WHERE clause in the SQL statement)

混合 OLAP (HOLAP)

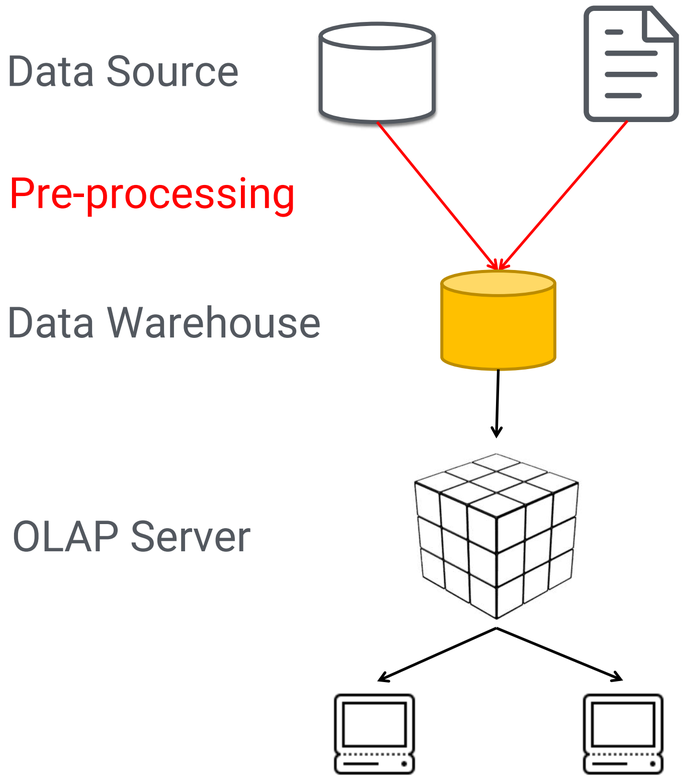
Hybrid OLAP

将 MOLAP 和 ROLAP 的优势结合起来，以获得更快的性能

Combines the advantages of MOLAP and ROLAP for faster performance.

MOLAP

Multi-dimensional OLAP



代表： Kylin, Druid

Representatives: kylin, Druid

优点： (Advantages)

- 卓越的性能 (Excellent performance)
- 支持复杂的计算 (Supports complex computing)

劣势： (Disadvantages)

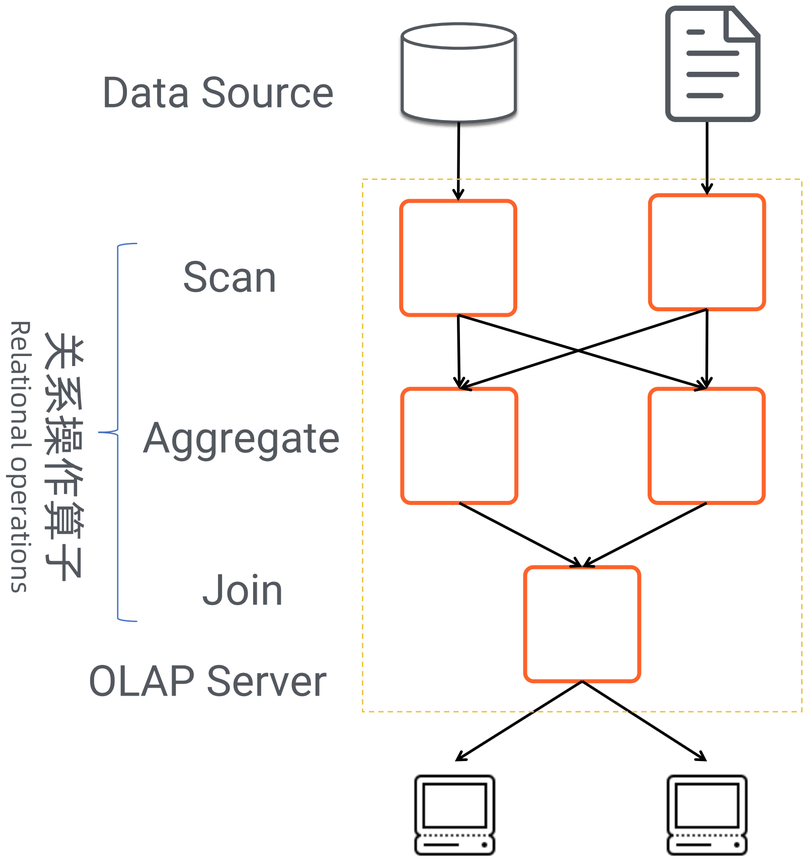
- 需要预先定义维度 (Must define dimensions in advance)
- 不支持明细数据查询 (Does not support query on detailed data)
- 需要更高的成本 (Higher cost)

场景： 适用于对性能要求非常高的场景

(**User Cases:** high performance requirements)

ROLAP

Relational OLAP



代表：Presto, Impala

Representatives: Presto, Impala

优点：(Advantages)

不需要预处理 (No need pre-processing)

可扩展性比较好 (Scalable)

可高效处理大量的数据 (Handle a large amount of data efficiently)

劣势：(Disadvantages)

性能可能比较慢 (Performance may be slow)

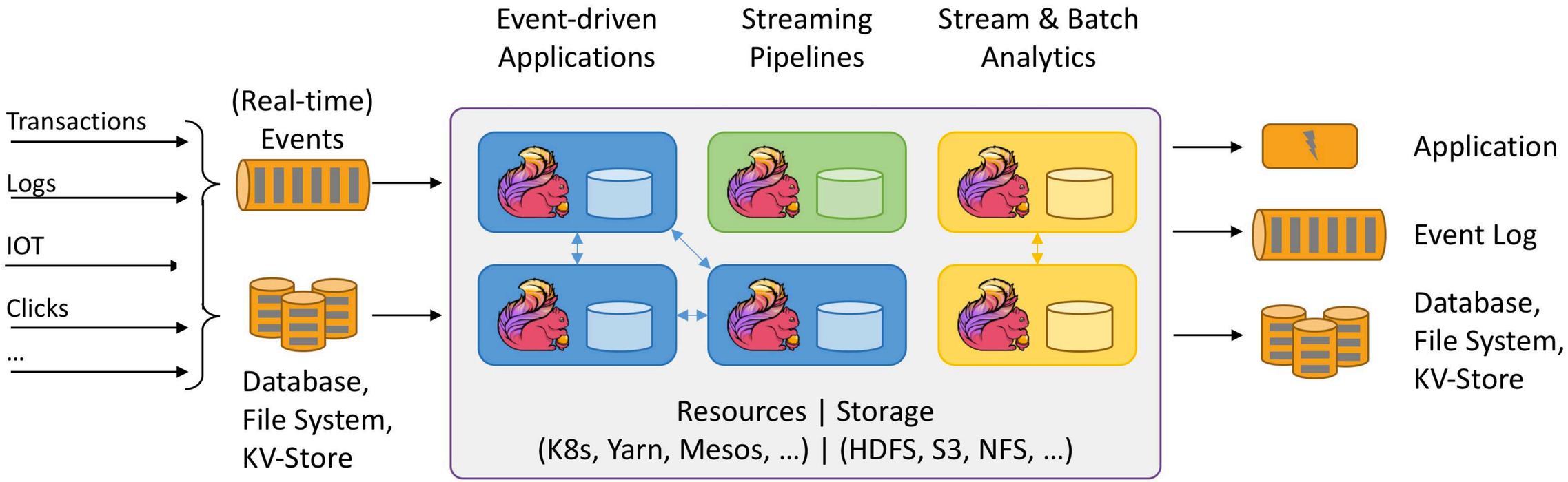
需要更多的计算资源 (Need more computing resources)

场景：适用于灵活性高的即席查询场景

(**User Cases** : flexible ad-hoc queries)

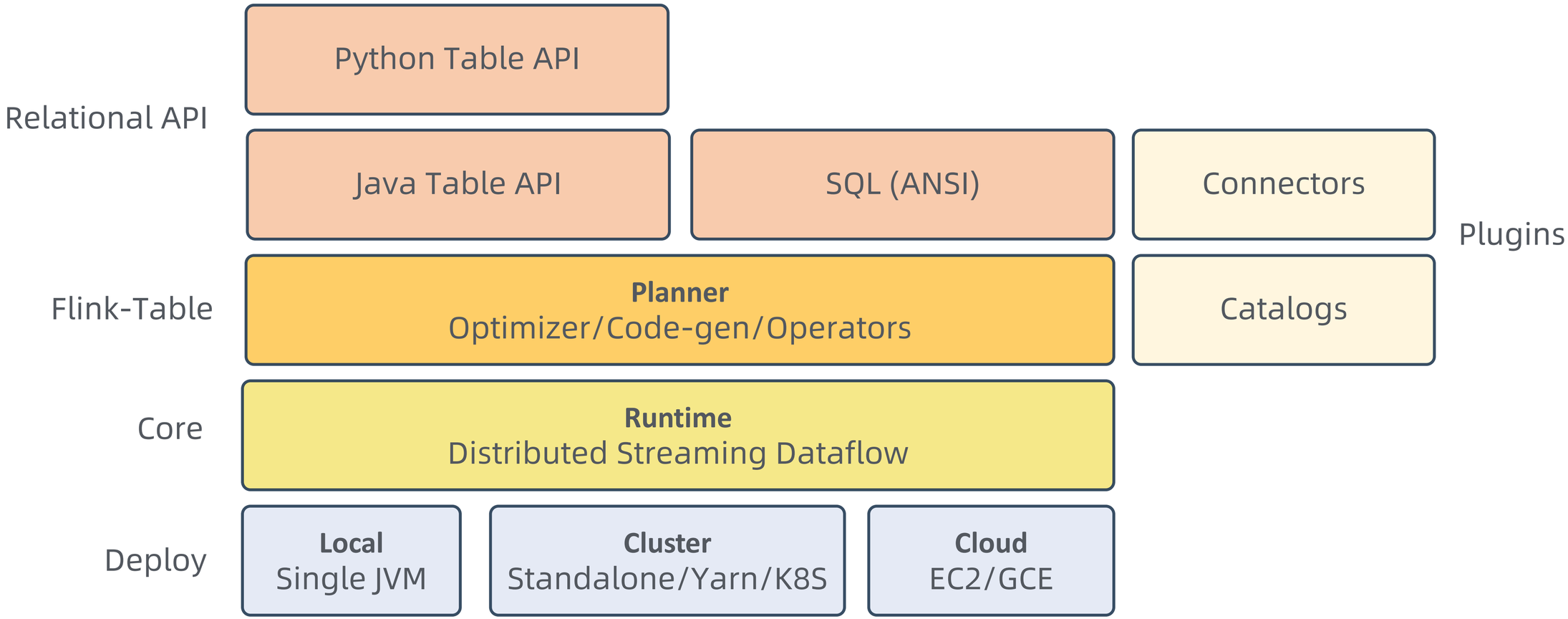
当前 Apache Flink 支持的应用场景

User Cases supported by Apache Flink



Apache Flink 架构

Apache Flink Architecture



Apache Flink 优势

Apache Flink Advantages

统一框架 (Unified framework)

用户 API 统一 (Unified User API)
执行引擎统一 (Unified Runtime Engine)

多层次API (Layered APIs)

ANSI SQL
Java/Python Table API
DataStream API

高性能 (Excellent performance)

内存计算 (In-Memory computing)
代价模型优化器 (Cost-based Optimizer)
代码动态生成 (Code-gen)

方便集成 (Easy integration)

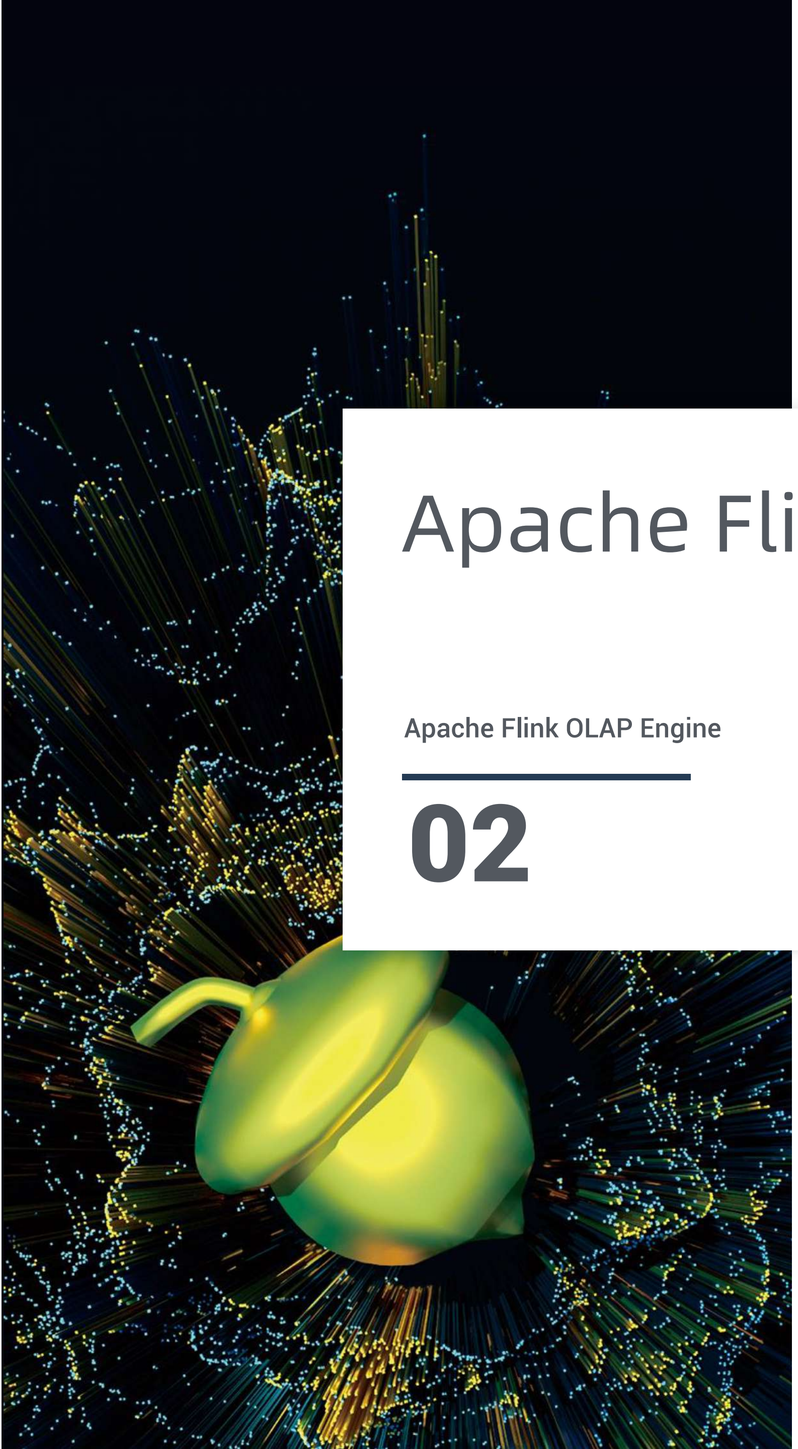
丰富的 Connectors (e.g. file, mq, db) (Various connectors)
方便对接现有 Catalog (Easy to use existing catalog)

灵活的 Failover 策略 (Flexible failover strategy)

Pipeline 支持快速 failover (Fast failover on Pipeline mode)
支持 Shuffle 数据落盘 (Spill to disk for shuffling data)

易部署维护 (Easy deployment & maintenance)

灵活的部署方案 (Flexible deployment)
支持高可用 (High-availability)



Apache Flink OLAP 引擎

Apache Flink OLAP Engine

02

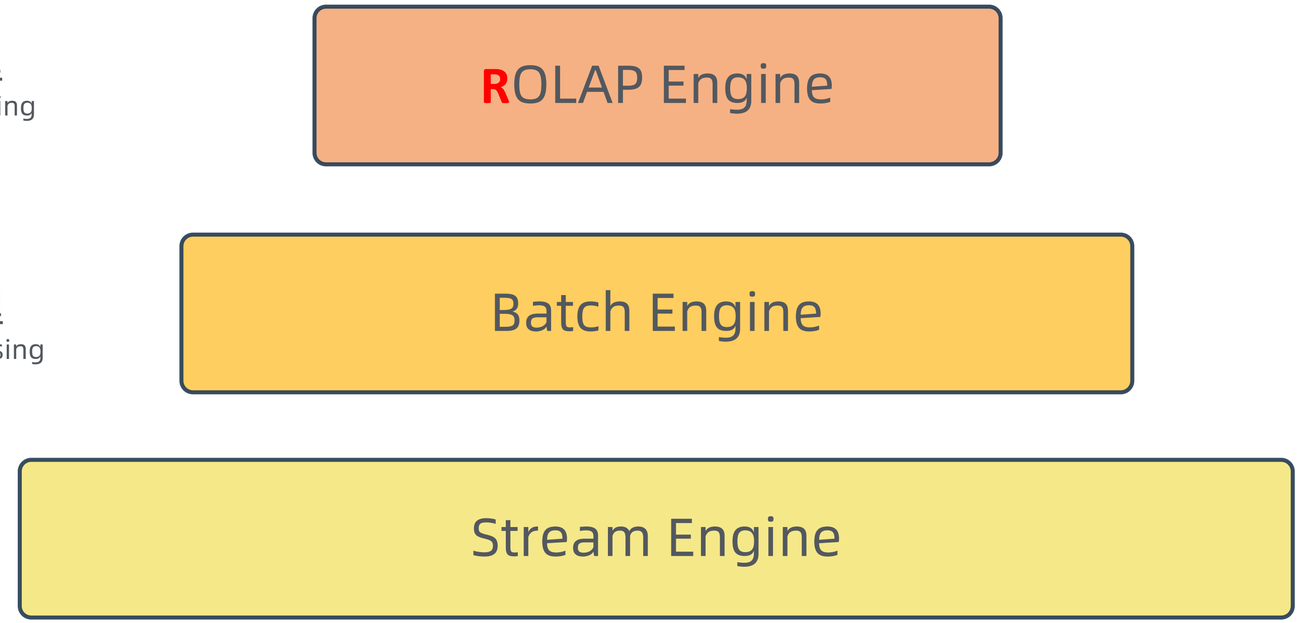
Apache Flink 可以做 ROLAP 引擎?

Could Apache Flink be an ROLAP engine?

特殊的批处理
Special batch processing

特殊的流处理
Special stream processing

基础
Base



短查询（追求极致的性能）
Short query (for extreme performance)

灵活的调度策略
Flexible scheduling strategy

算子支持落盘
Operators support spilling to disk

.....

内存计算
In-Memory computing

Pipeline

.....

Apache Flink 做 OLAP 引擎优势

The advantages of Apache Flink as an OLAP engine

统一引擎

Unified Framework

降低学习成本

Easy to learn

提高开发效率

Improve development efficiency

提高维护效率

Improve maintenance efficiency

既有优势

Existing strengths

内存计算

In-Memory computing

Pipepline

代码动态生成

Code-gen

.....

相互增强

Mutual reinforcement

无统计信息的优化

Optimization without statistic

更高效的算子

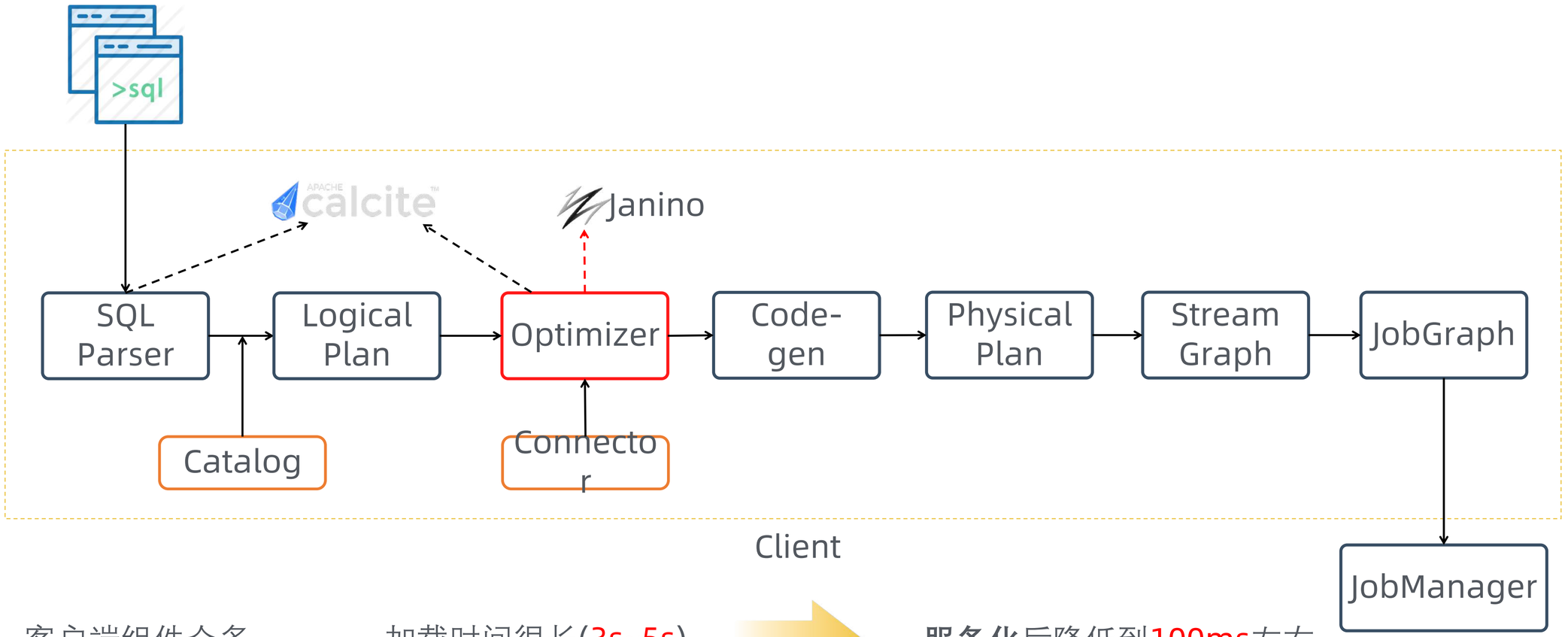
More efficient operators

更通用的框架

More general framework

客户端服务化

Client as a service



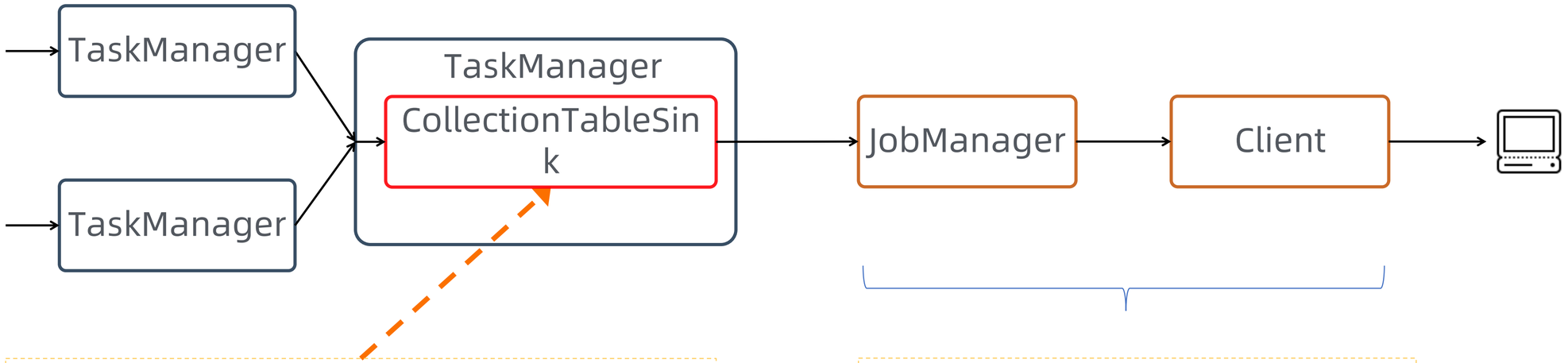
客户端组件众多
Too many components

加载时间很长(3s~5s)
Long loading time

服务化后降低到100ms左右
Reduces to about 100ms after servitization

自定义 CollectionTableSink

Custom CollectionTableSink



验证接收到的**数据条数**是否达到限制
Verify that the number of received records reaches the limit
验证接收到的**数据大小**是否达到限制
Verify that the data size of received records reaches the limit

避免 OOM
Avoid OOM
保证其他正在运行 Query 的稳定性
Guarantee the stability of other running queries

调度优化

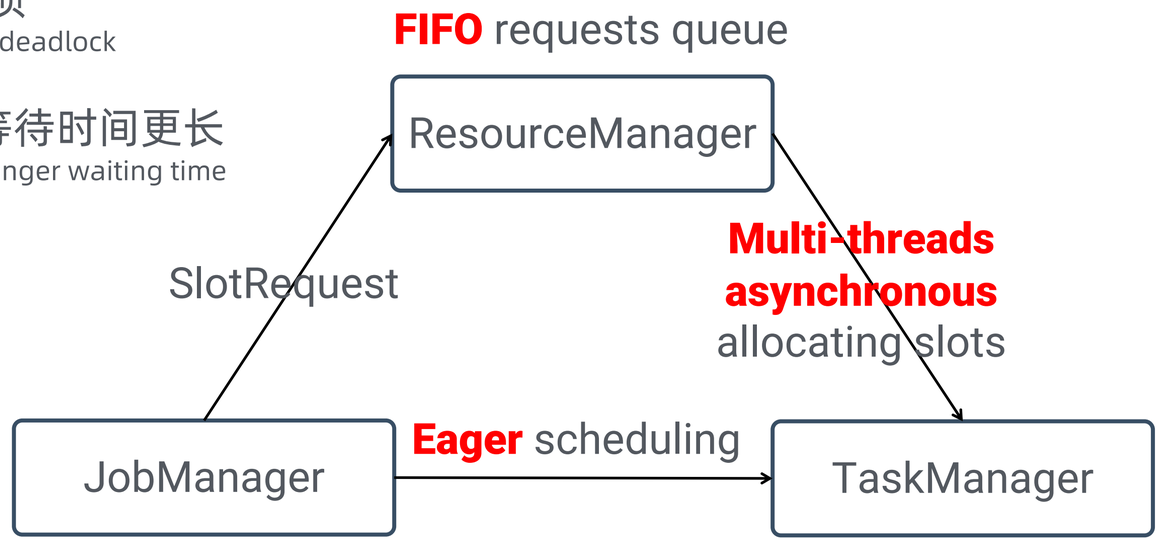
Scheduling Optimization

当前调度模式及问题：
Current scheduling mode and problems:

LAZY_FROM_SOURCES 模式调度 → 整体运行时间变长 & 可能死锁
LAZY_FROM_SOURCES scheduling → Longer execution time and may cause deadlock

RM 按 OnDemand 方式分配 Slot 请求 → 可能死锁
RM assign slot requests in OnDemand mode → May cause deadlock

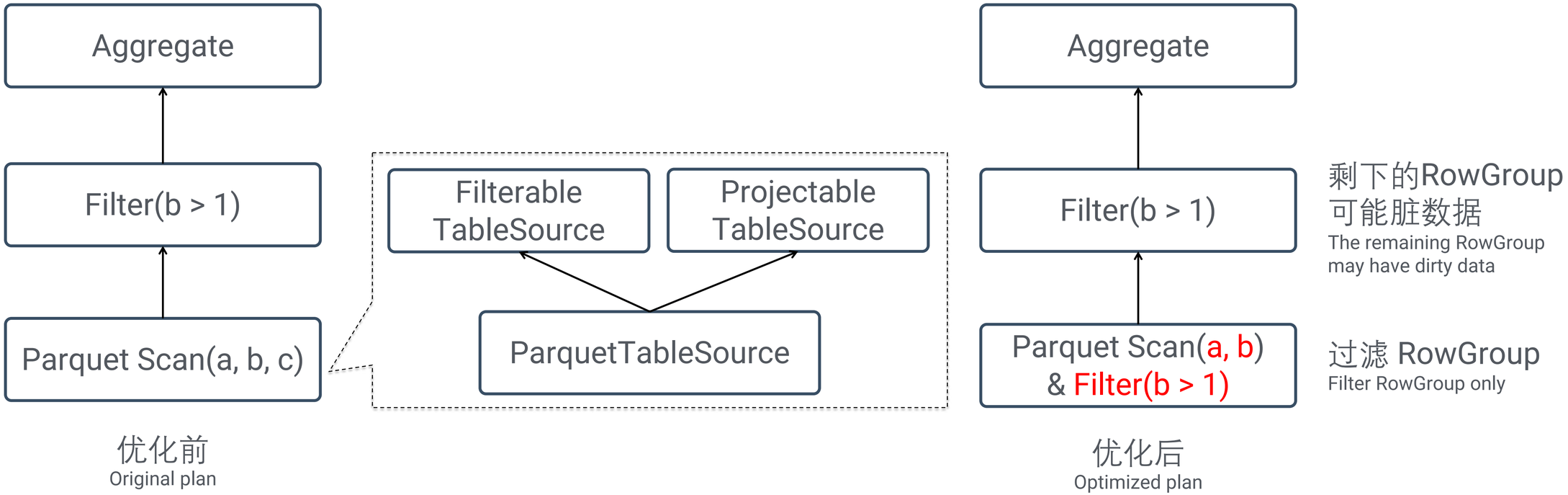
RM 以单线程同步模式向 TM 分配 Slot 请求 → 等待时间更长
RM allocates slot requests to TM in single-thread synchronous mode → Longer waiting time



Project & Filter 下推

Project & Filter Push-down

SELECT a, SUM(b) FROM t WHERE b > 1 GROUP BY a



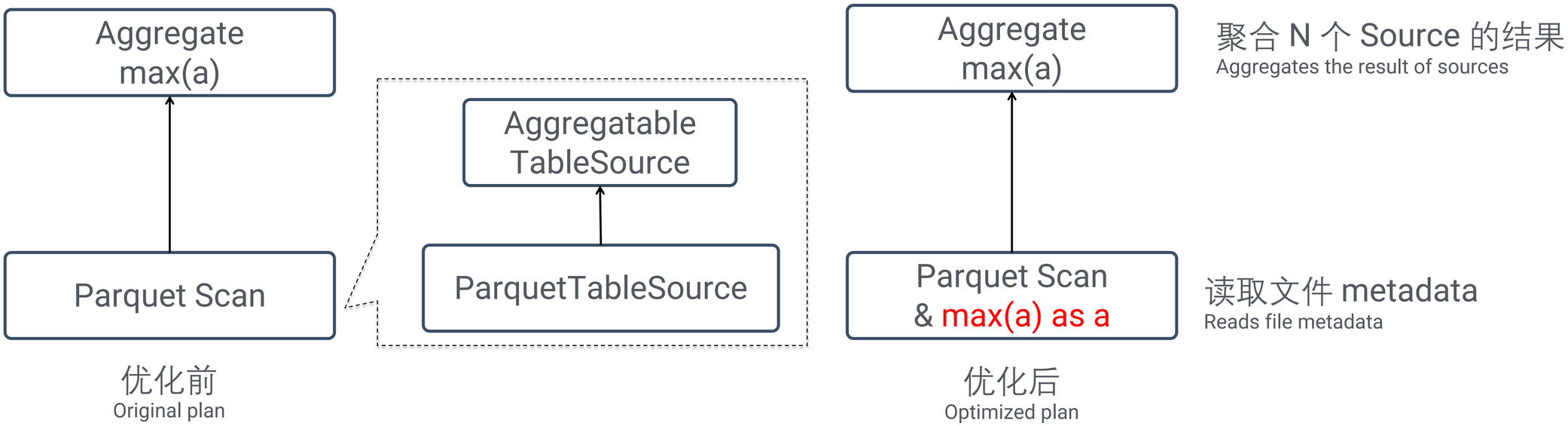
优化前
Original plan

优化后
Optimized plan

Aggregate 下推

Aggregate Push-down

SELECT MAX(a) FROM t



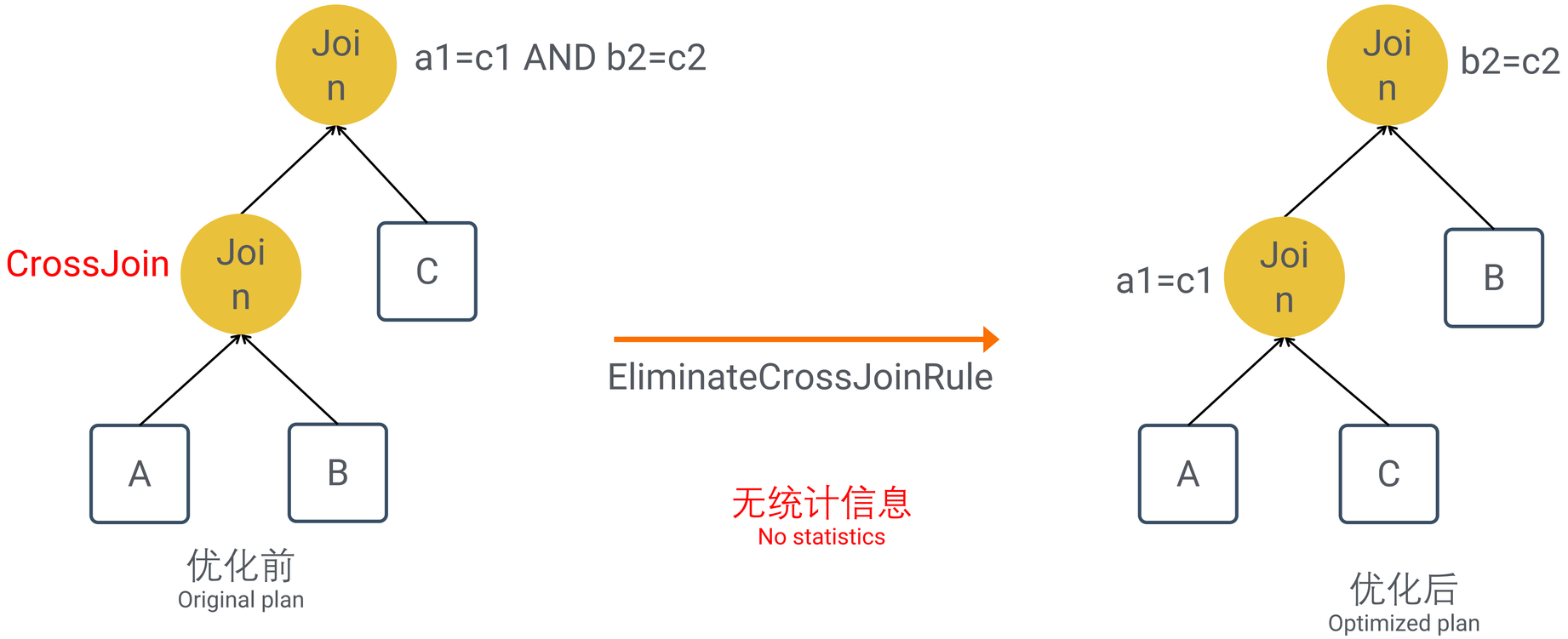
优化前
Original plan

优化后
Optimized plan

消除 CrossJoin

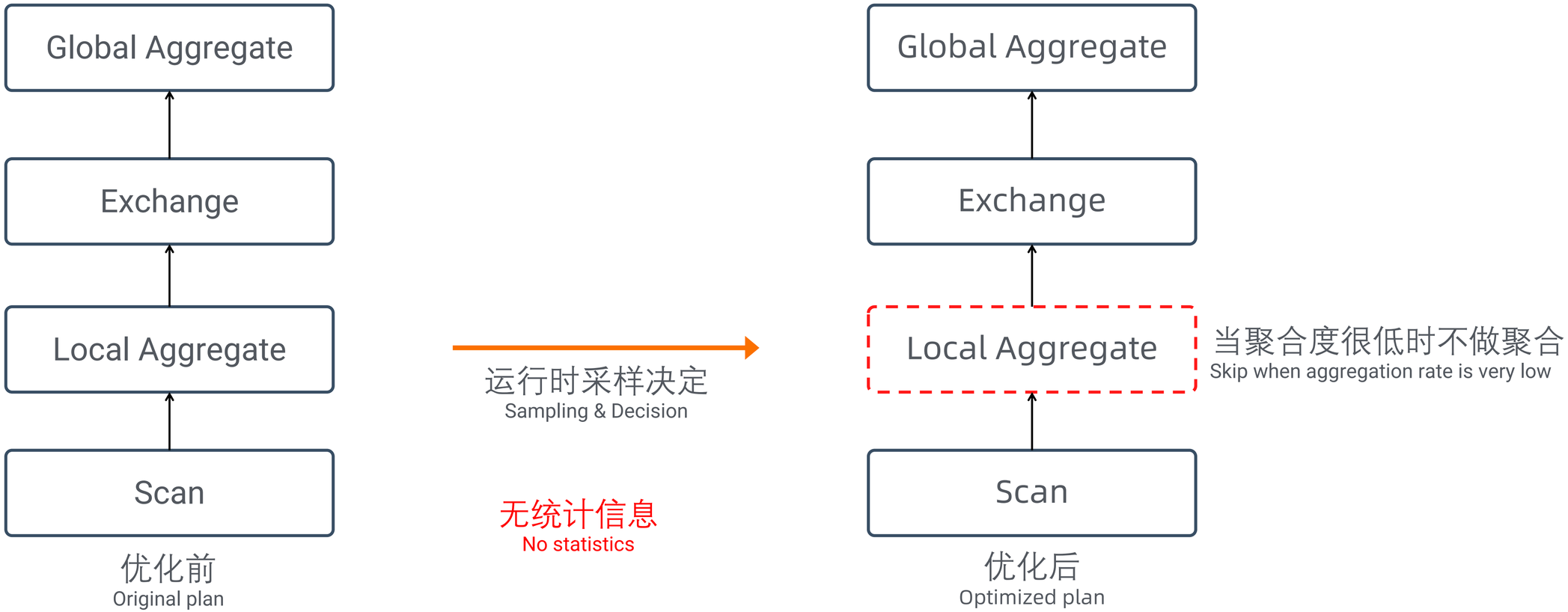
Eliminate CrossJoin

SELECT * FROM A, B, C WHERE a1=c1 AND b2=c2

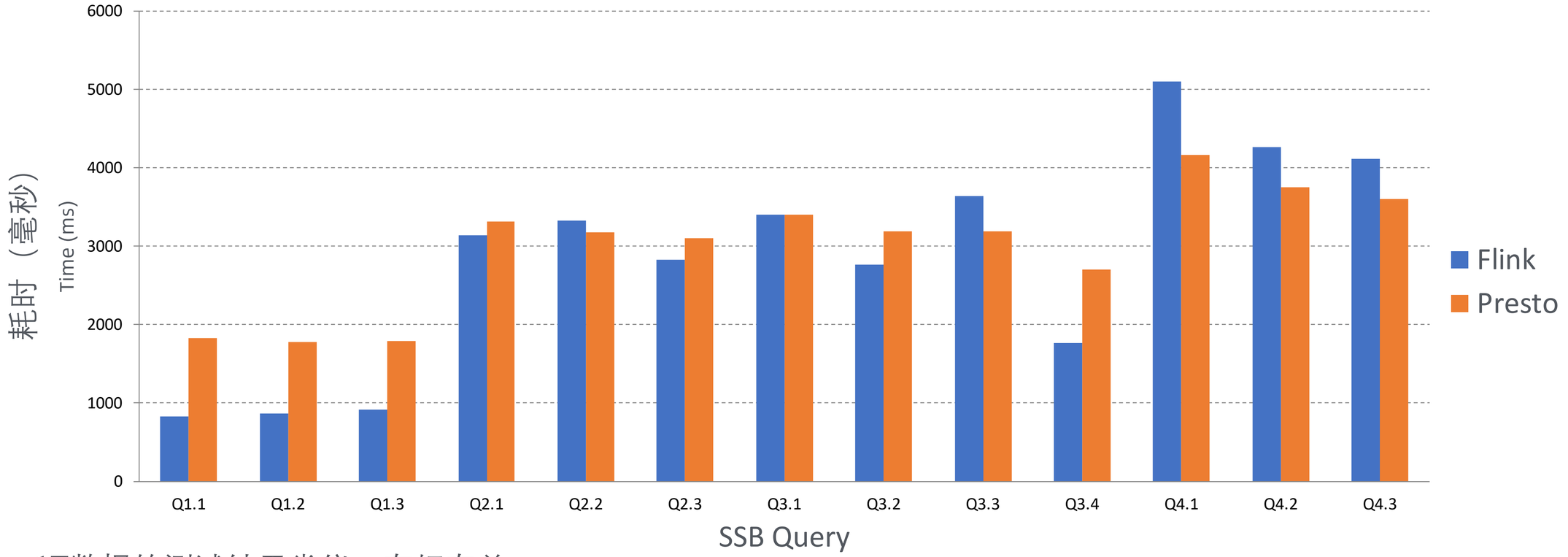


自适应的 Local Aggregate

Self-adaptive Local Aggregate

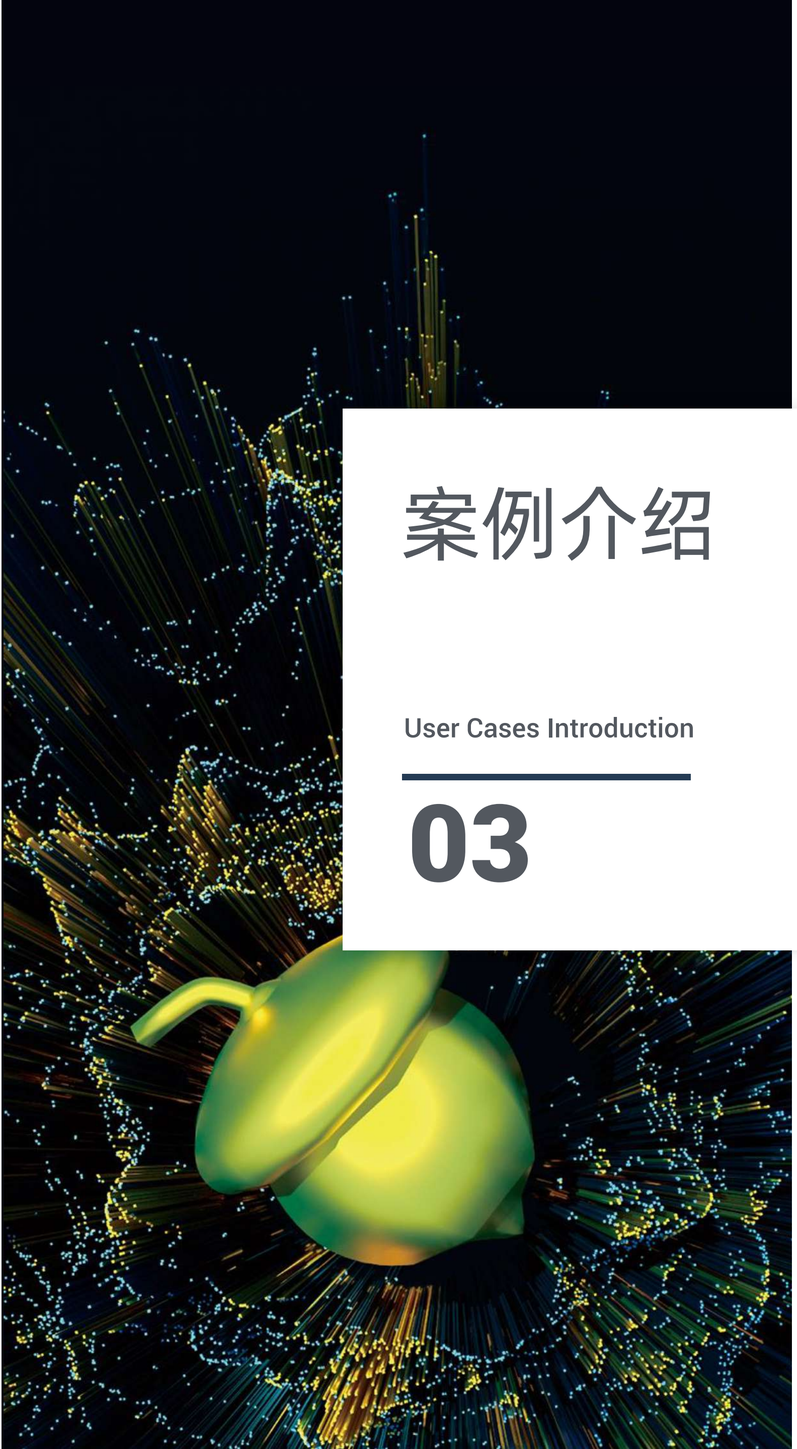


Star Schema Benchmark (100G)



1T数据的测试结果类似：有好有差

The results of 1T are similar: some are better, some are worse



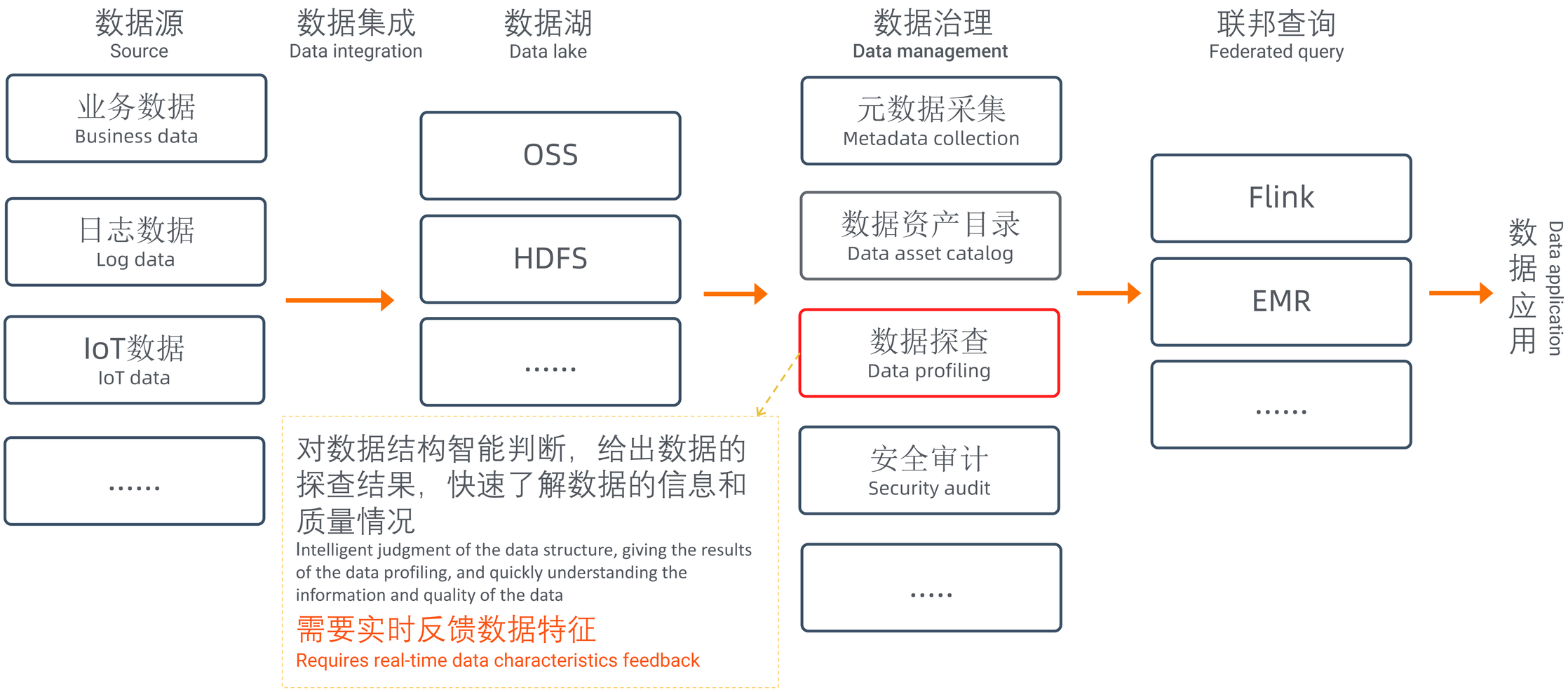
案例介绍

User Cases Introduction

03

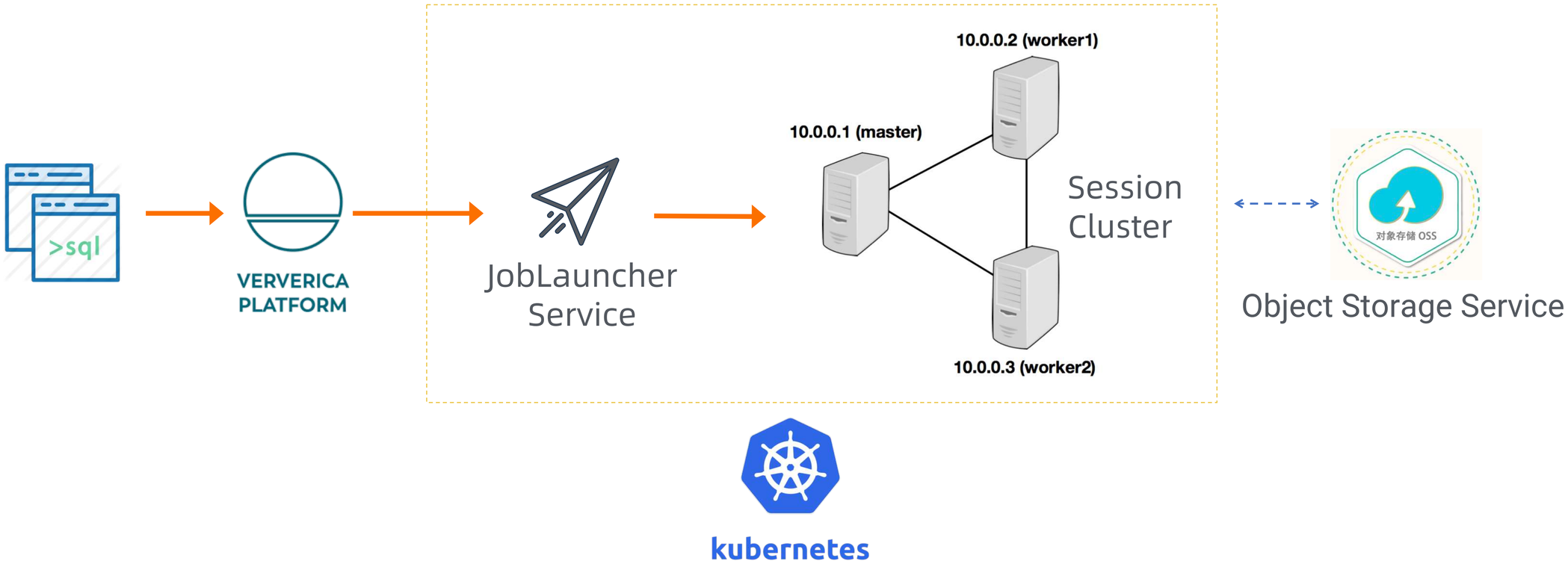
Apache Flink OLAP 在数据探查上的应用

An Application of Apache Flink OLAP on Data profiling



整体架构

Architecture Description



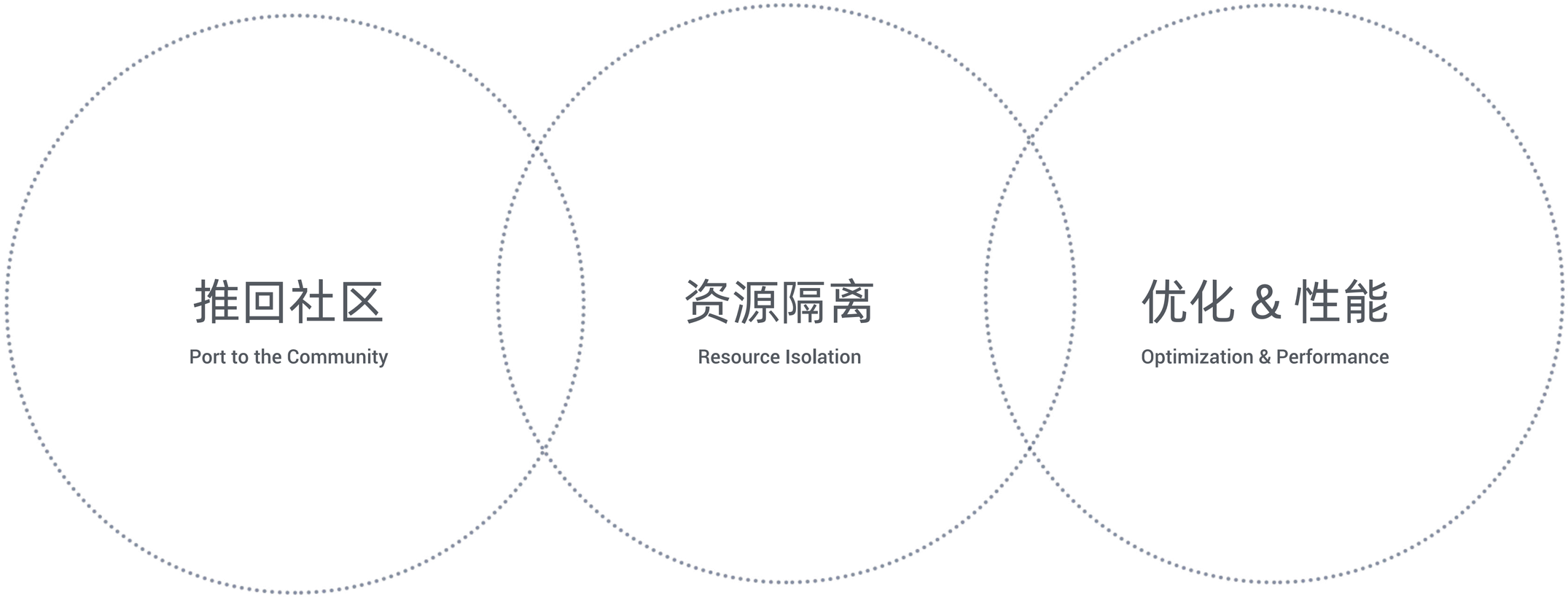
未来计划

Future plans

04

未来计划

Future Plan



FLINK FORWARD

THANKS