FLINK FORWARD # ASIA 2022 ONLINE | 实时即未来

流批一体
在快手的实践和思考

张静 Apache Flink && Apache Calcite Committer

1 Flink 在快手的发展

2 流批一体在快手的规划

3 第一阶段（加强批能力）的进展

4 第二阶段（业务视角的流批一体）的挑战

1 Flink 在快手的发展

Flink 在快手的发展

13亿/秒
TPS

6000 实时
3000 离线
作业数目

70W
CPU Cores

社科
索引构建
样本拼接
特征计算

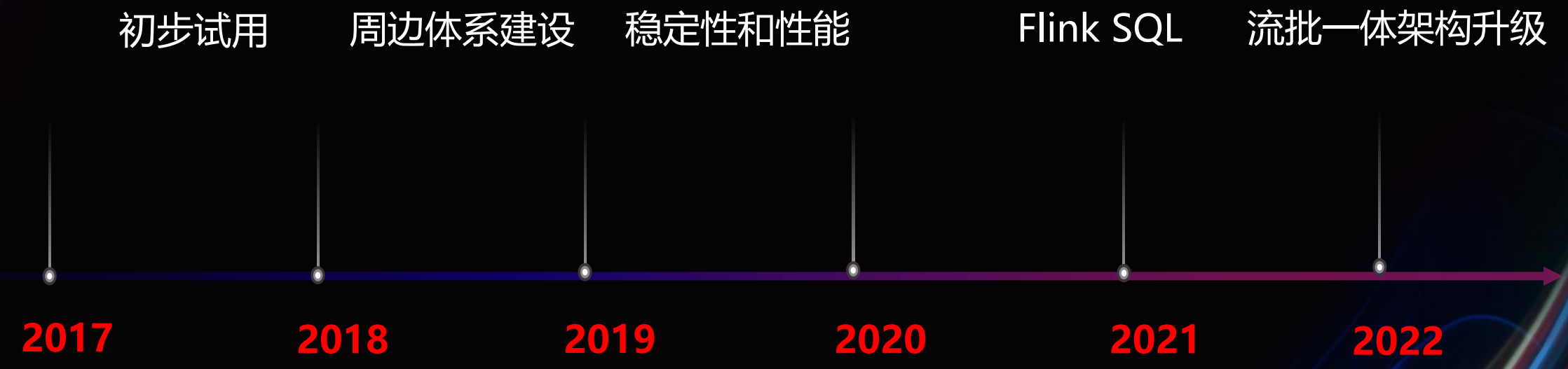
电商
快手小店
直播助手
直播大屏

商业化
投放平台
实时收入健康
特征计算

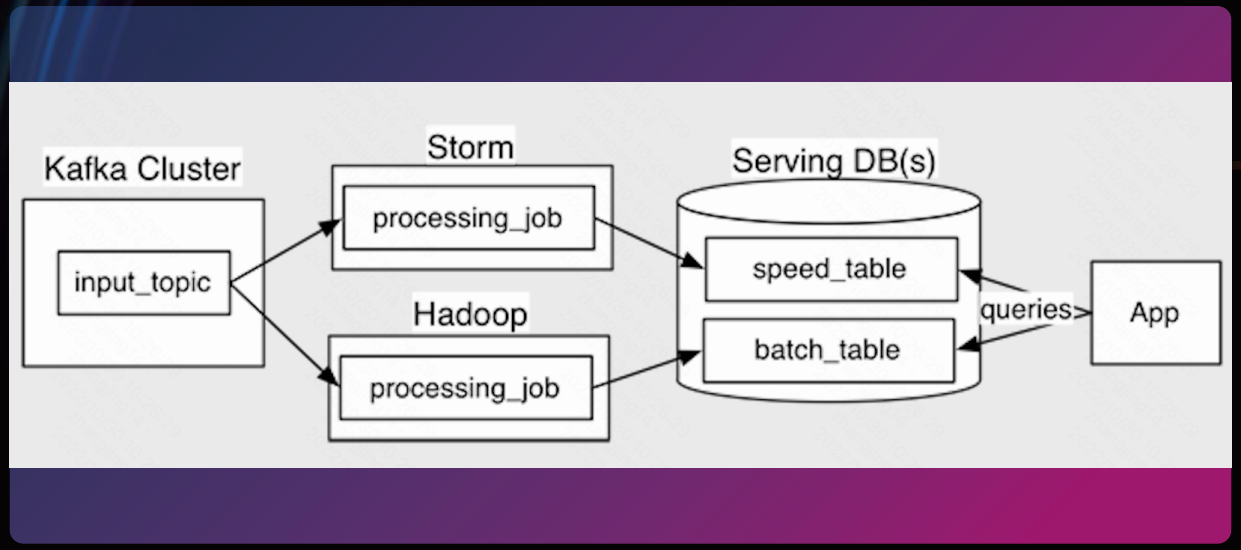
音视频
精彩瞬间
音视频质量监控

实时数据组
实时数仓
实时大屏
实时报表
实时数据同步

Flink 在快手的发展



Lambda 架构



引自 Questioning the Lambda Architecture

缺点

两套计算引擎

两套业务代码

结果一致性难以保证

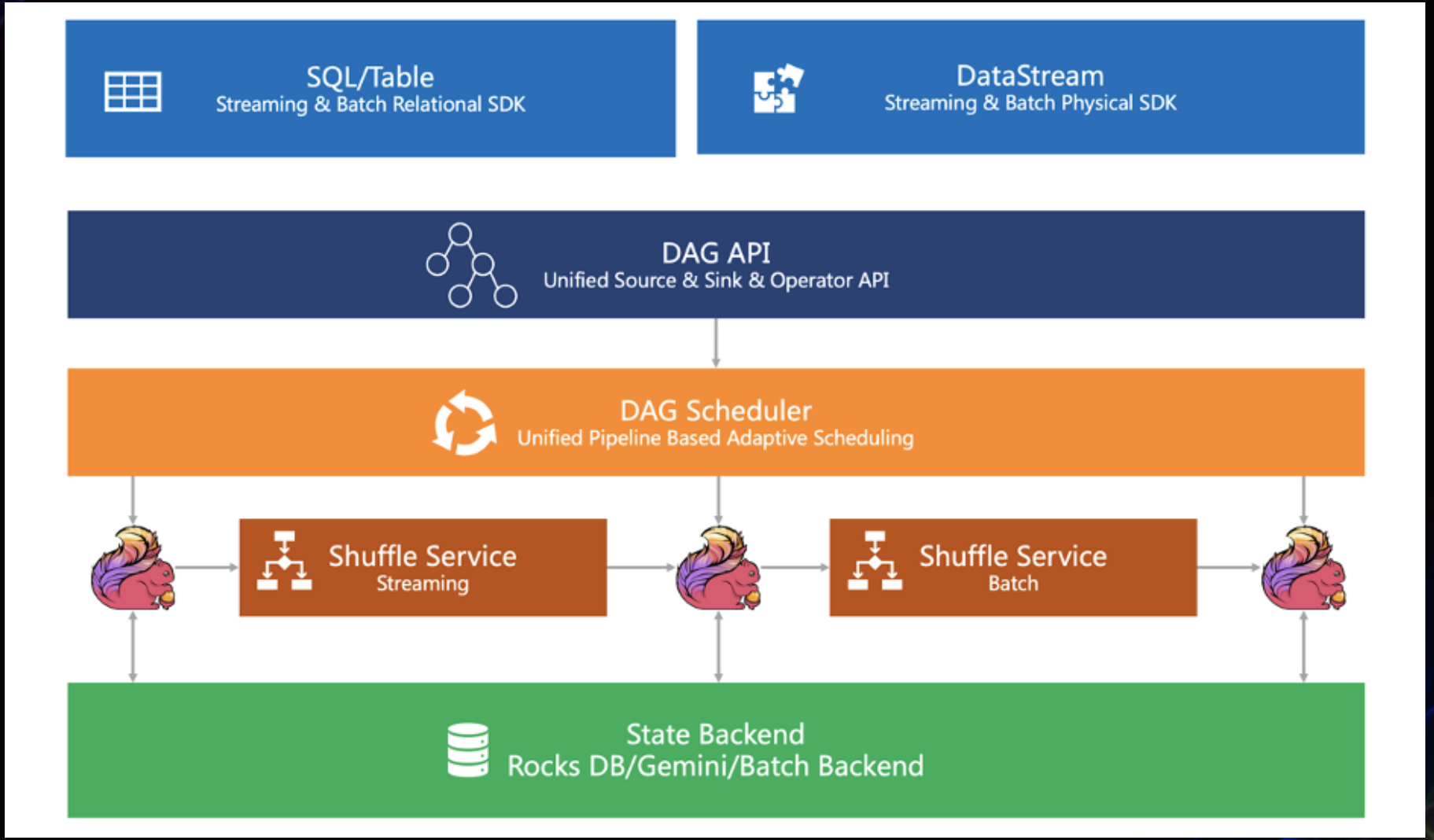
流批一体的方案选型

统一的 API
不同的计算引擎

统一的计算引擎
不满足极致的实时性

统一的计算引擎
批处理能力待加强

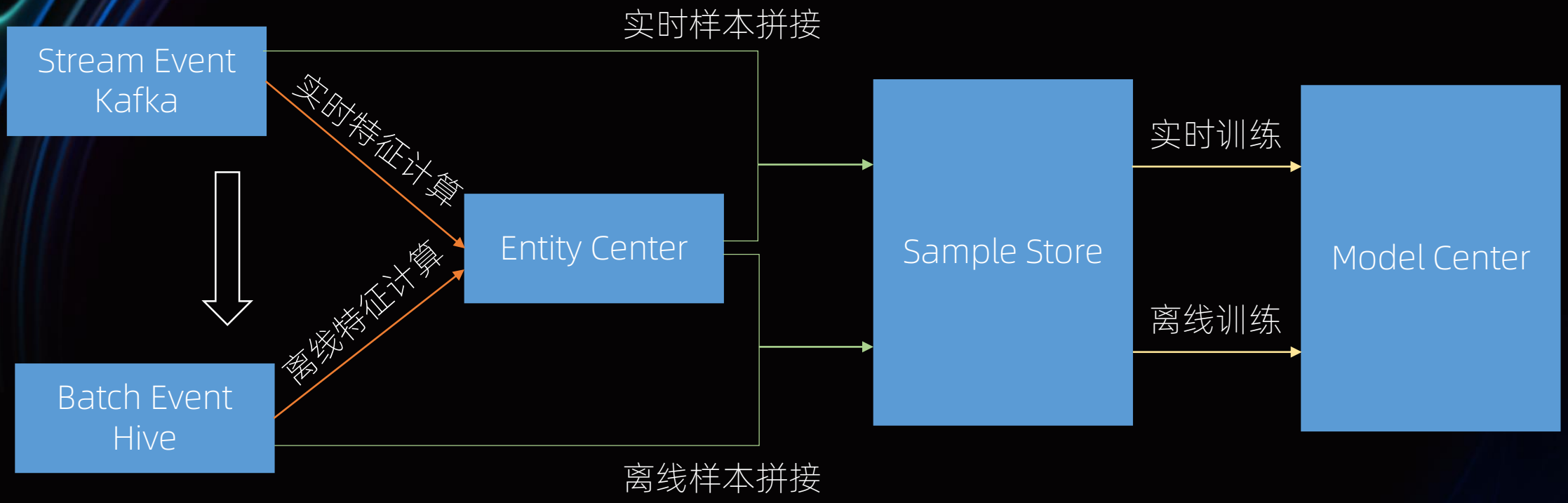
Apache Flink 流批一体



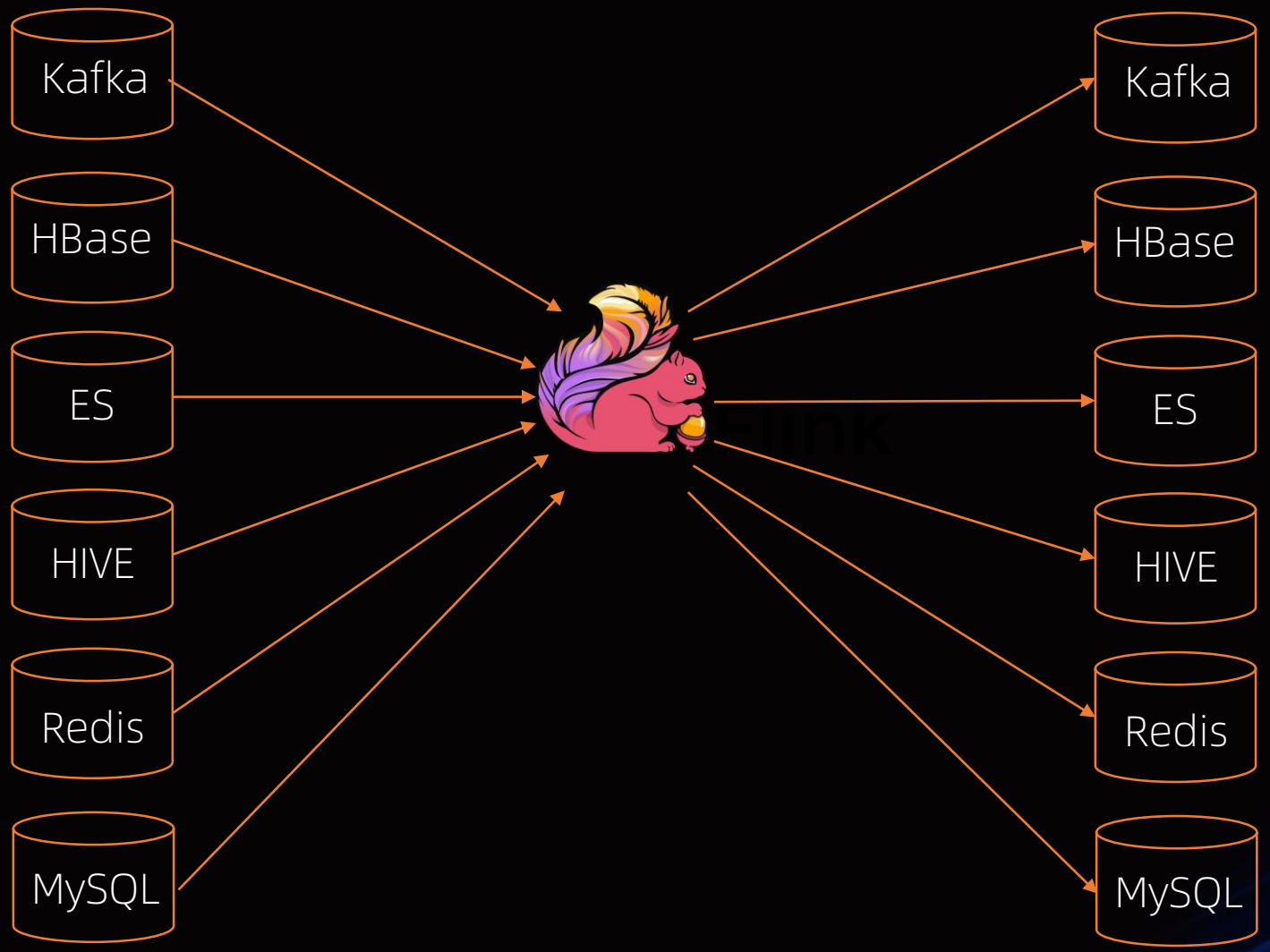
引自 Flink 执行引擎：流批一体的融合之路

2 流批一体在快手的规划

流批一体的初期业务场景1：机器学习



流批一体的初期业务场景2：数据集成



流批一体的目标

统一的用户体验：一套业务逻辑，一套平台入口

统一的引擎：计算引擎 + 存储引擎

智能的引擎：用户只关心业务逻辑和实时性要求

满足更复杂的业务：比如流批融合的需求

流批一体的规划

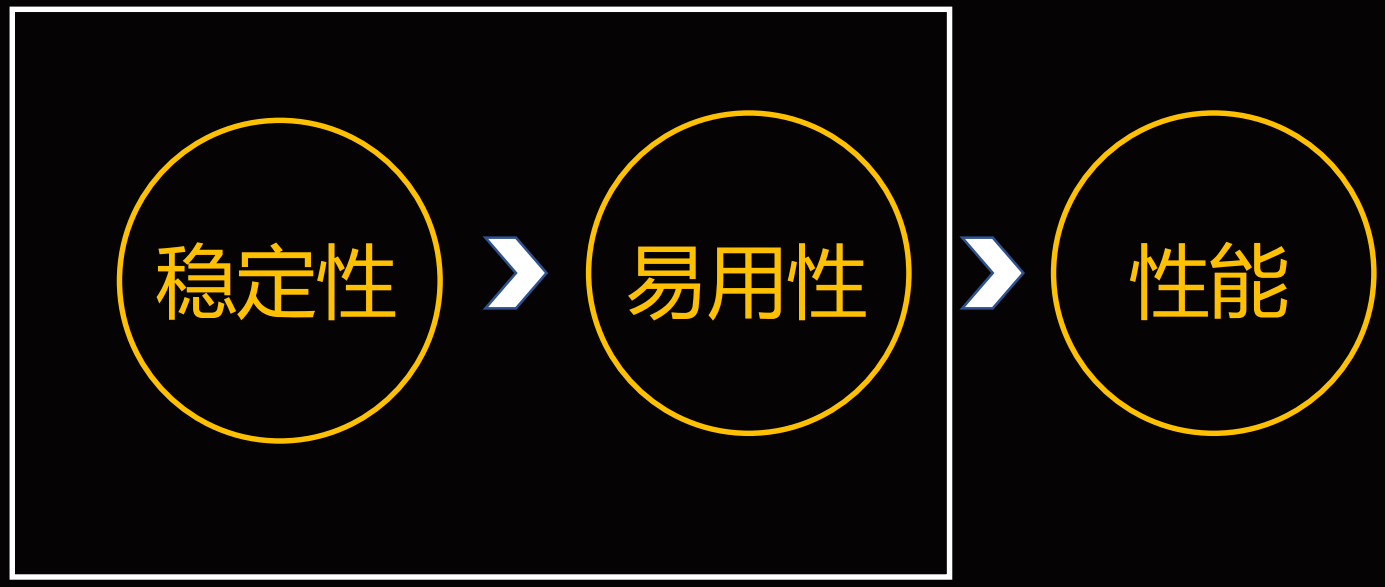

加强 Flink 批能力
打通产品入口
支持业务（AI & 数据同步）
第一阶段
第二阶段
智能的引擎
极致的批处理
统一的流批一体存储引擎
广泛的业务场景

3 第一阶段的进展

初期用户对批能力的要求

稳定性 › 易用性 › 性能

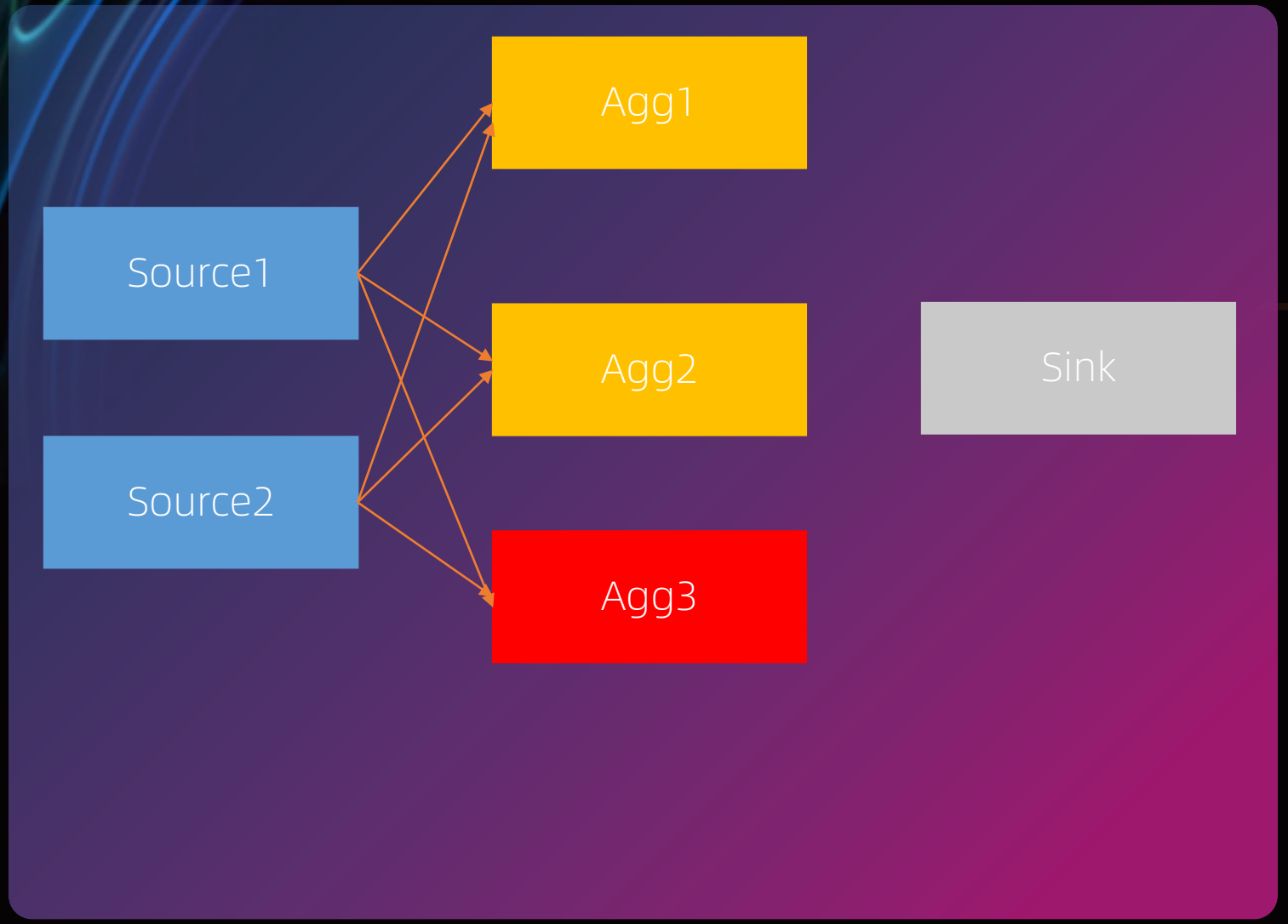
初期用户对批能力的要求



稳定性的核心问题

慢节点问题	• 机器或者网络等造成的长尾问题
TM Shuffle 不稳定	• 节点异常退出导致托管的 Shuffle 文件不可读，影响下游任务
离线任务稳定性差	• 离线集群开启资源抢占，中低优任务的资源频繁被抢占 • 离线集群资源紧张，导致并发间 splits 分配不均匀，failover 开销大

稳定性问题1：慢节点问题

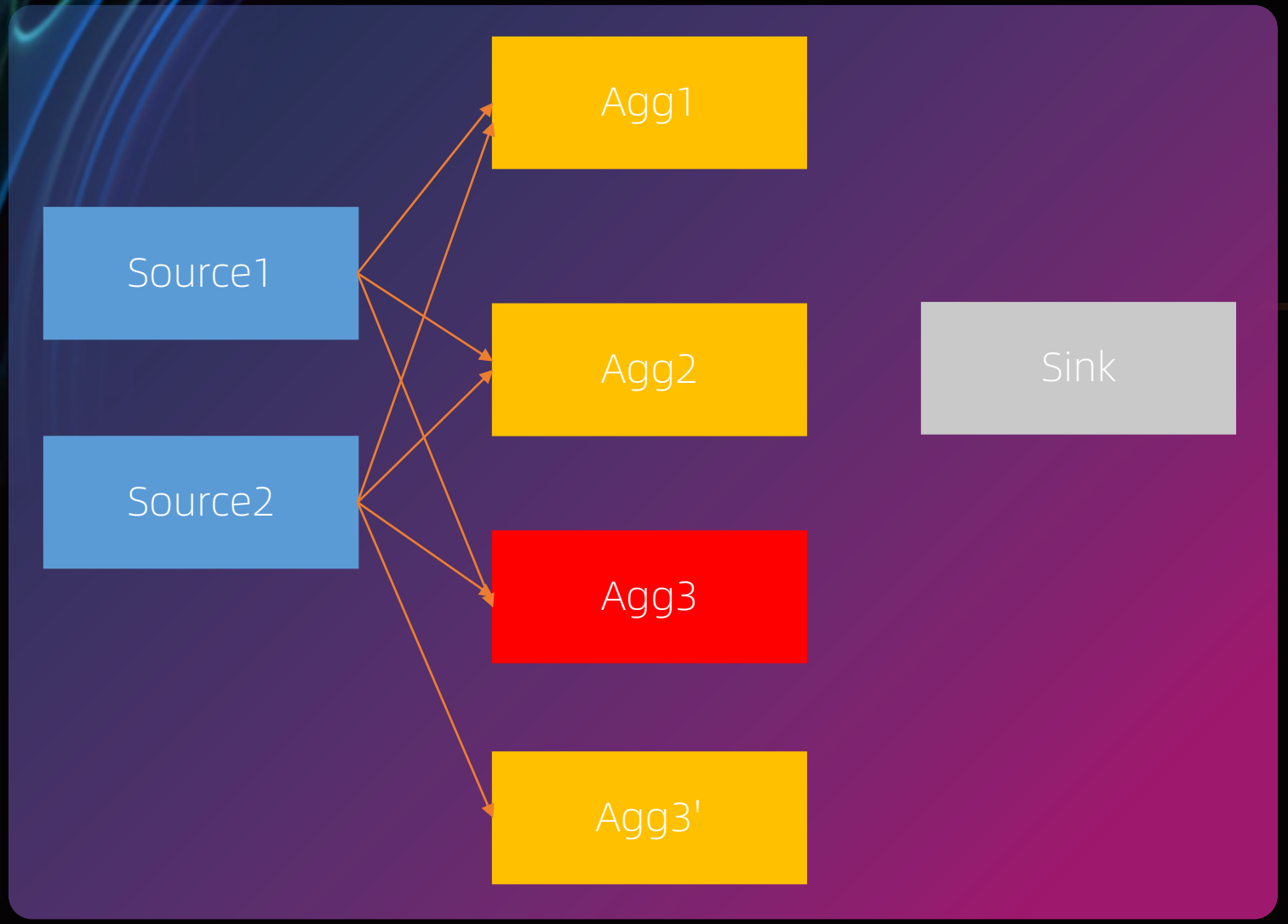


问题描述

个别并发上运行缓慢，可能因为机器原因或者网络问题，影响整个作业的执行时间

Note：针对各个并发处理速度不一致，而非数据倾斜导致的慢节点

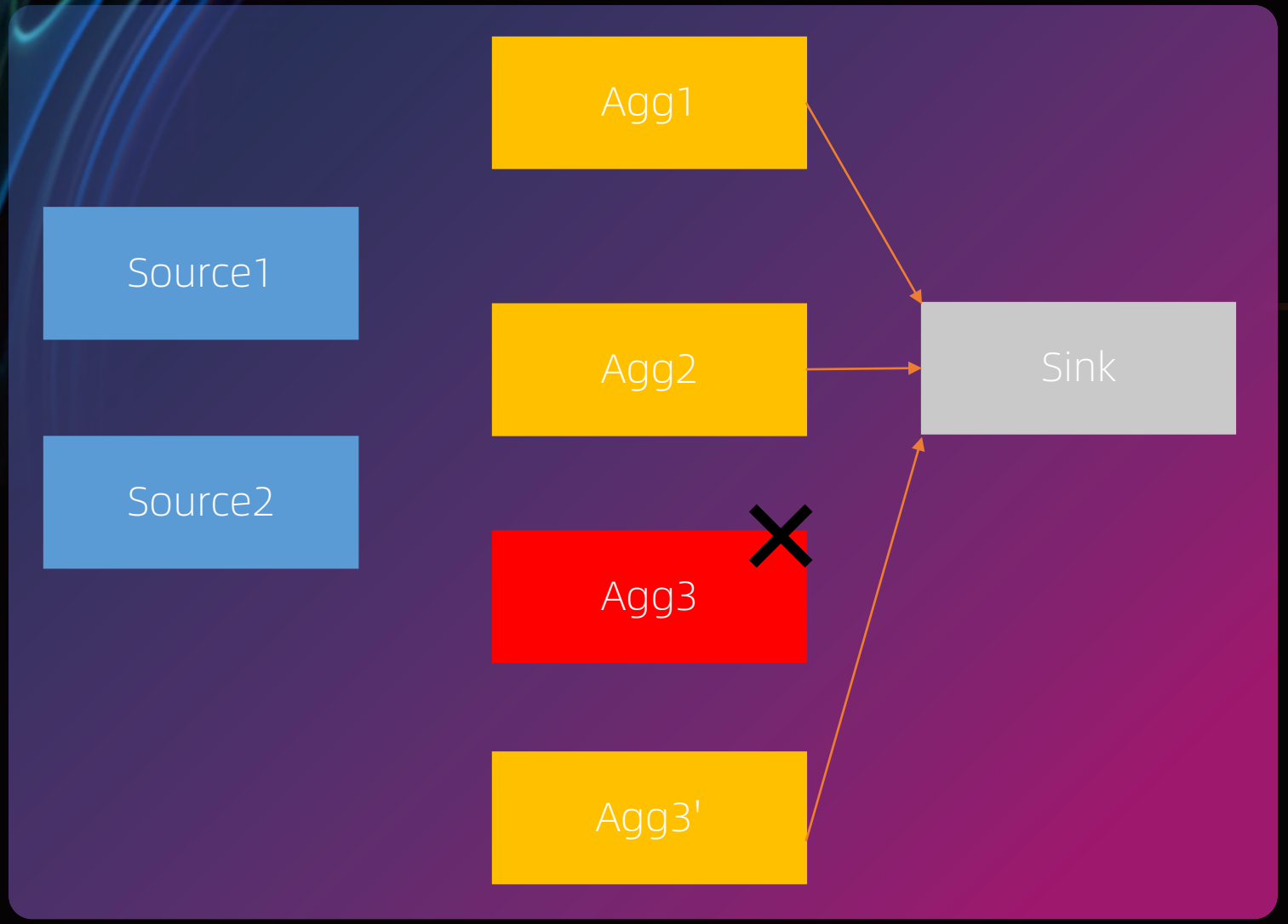
推测执行 （FLIP-168）



方案描述

在非热点机器上启动镜像实例

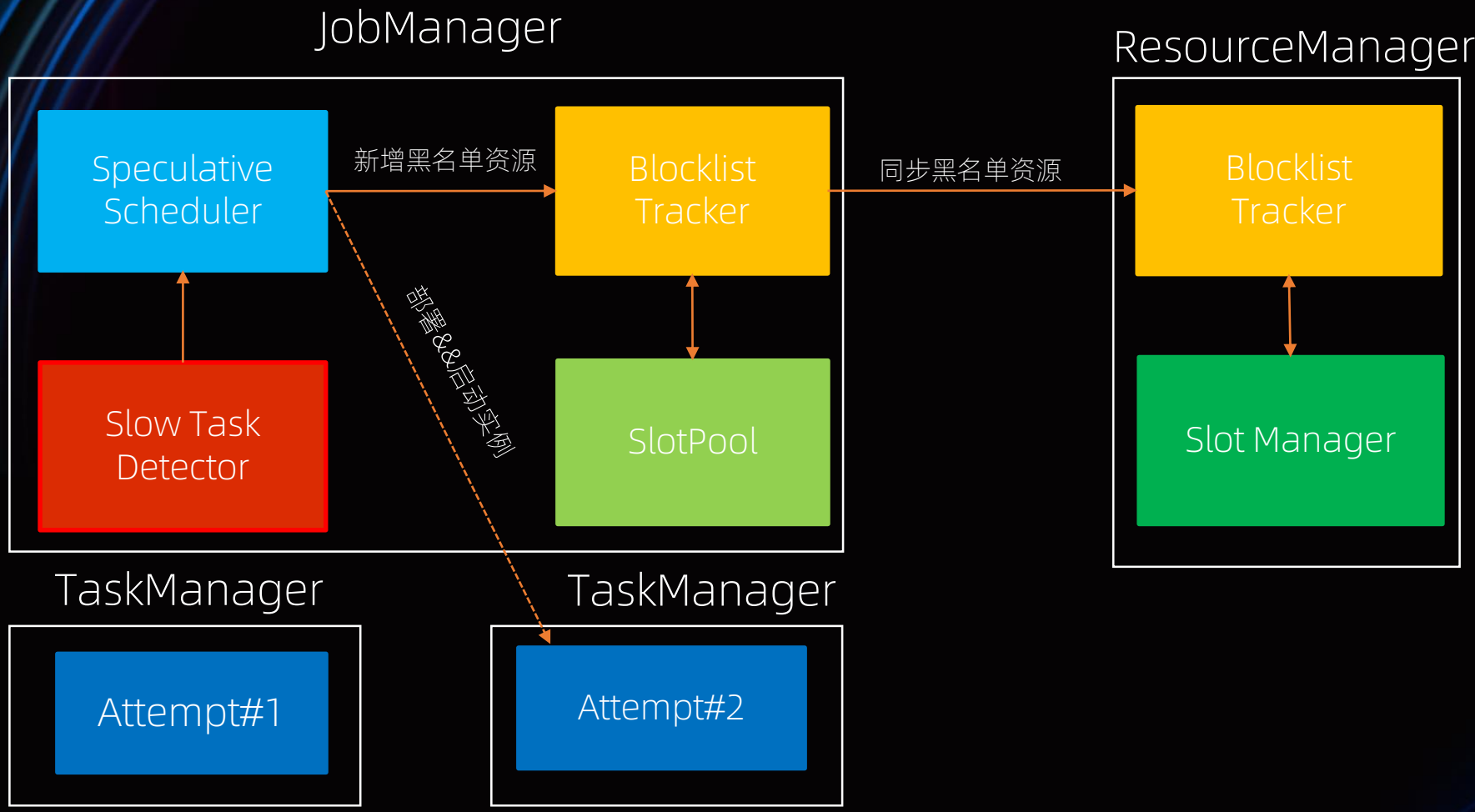
推测执行 （FLIP-168）



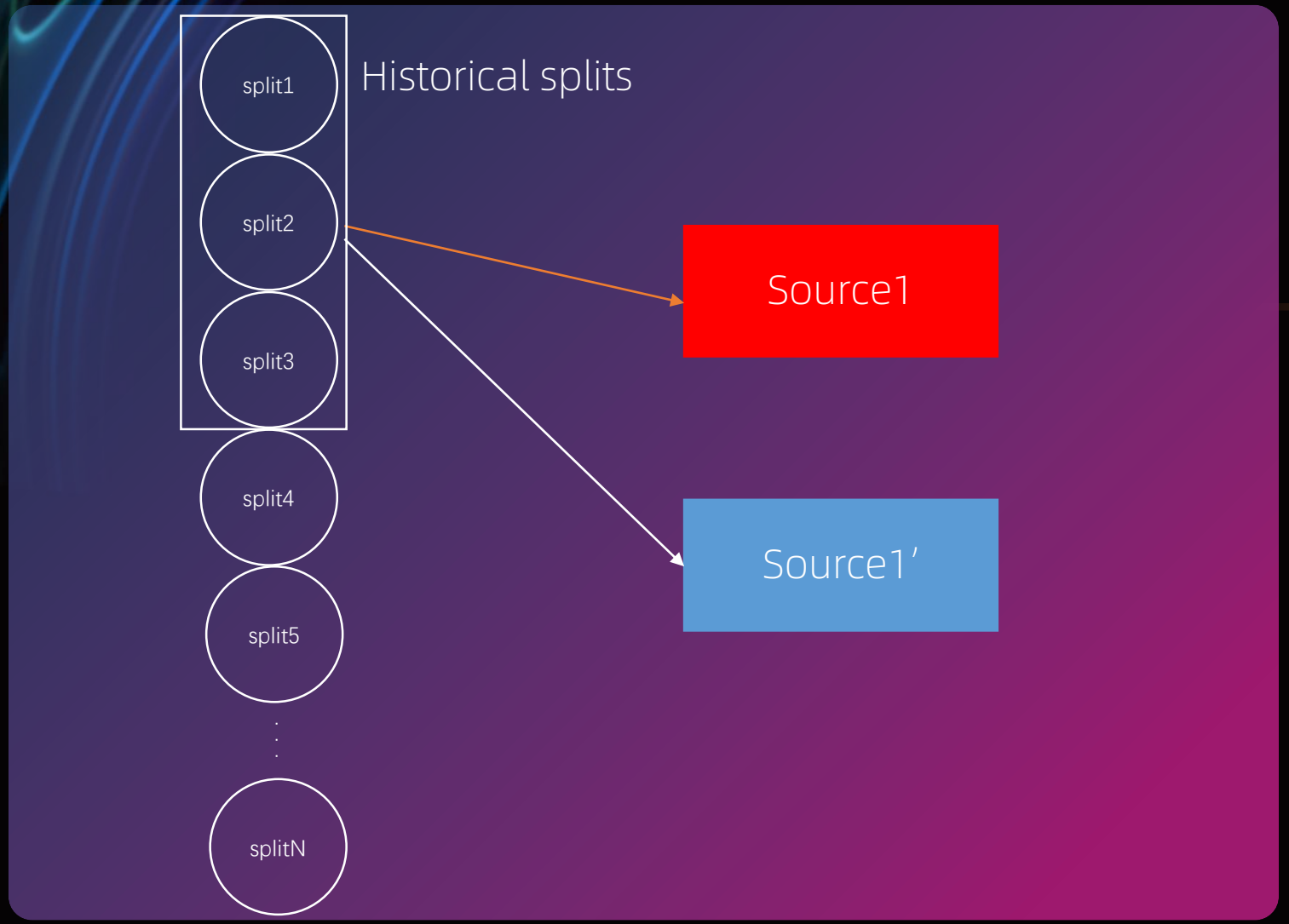
方案描述

哪个先执行完就采用它的结果，并把慢的取消掉

推测执行 （FLIP-168）



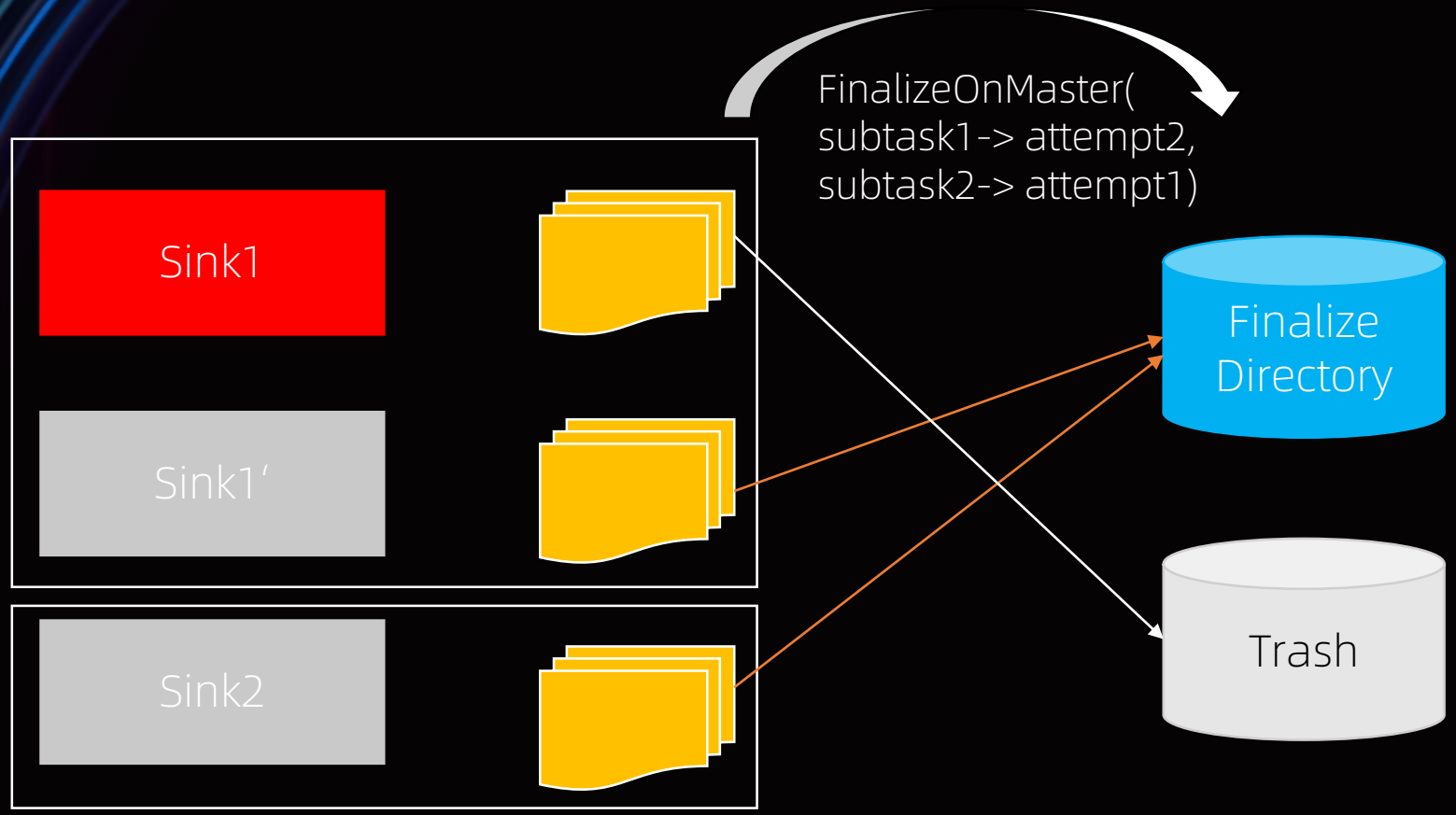
Source 支持推测执行 （FLIP-245）



方案描述

- ✓ 镜像实例和原始的实例，必须生成相同的数据
- ✓ 通过引擎侧的改动让 Source 支持推测执行，不需要每个 source connector 额外开发

OutputFormat Sink 支持推测执行



would be merged in 1.17

稳定性问题2：TM Shuffle 不稳定

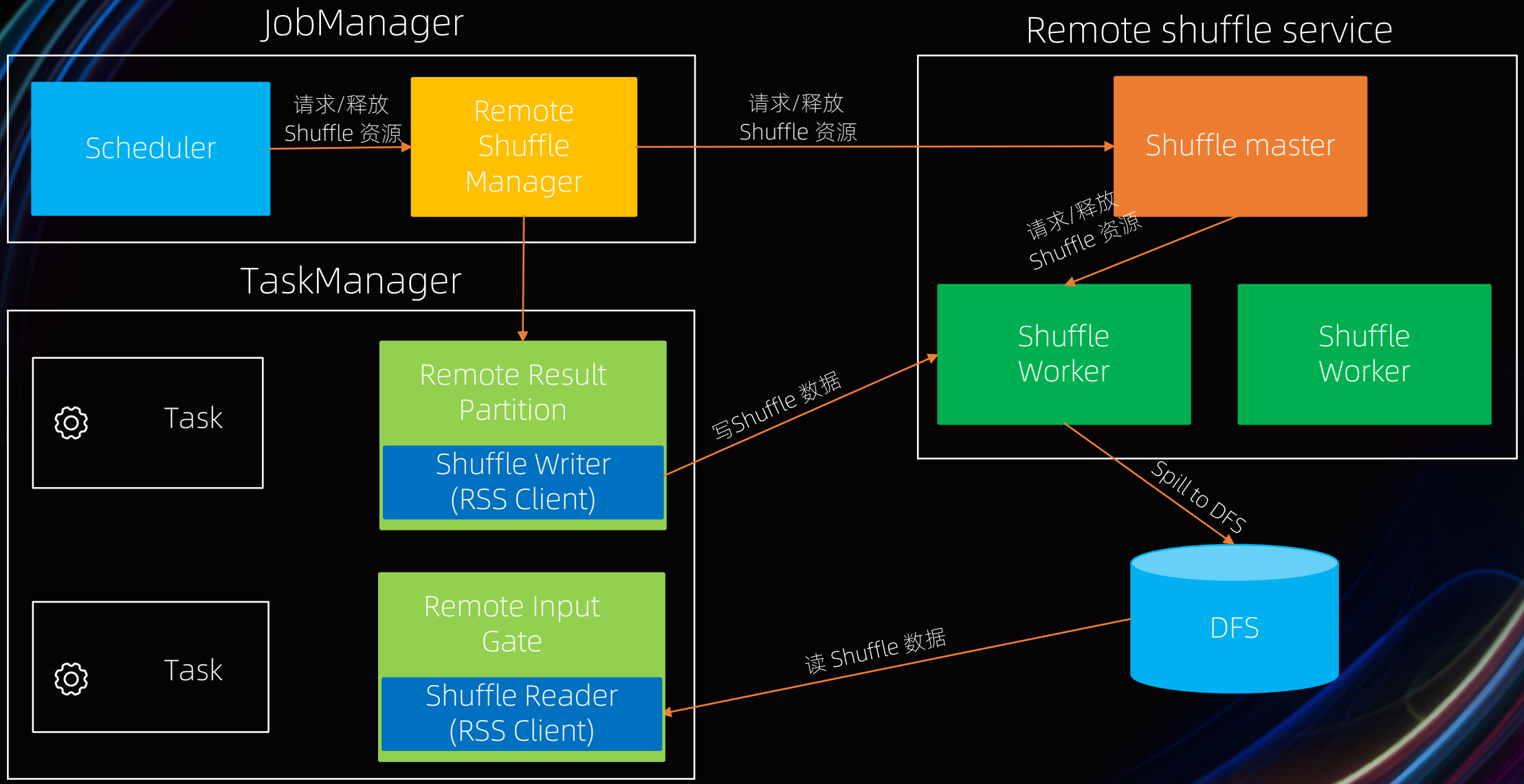
TaskManager Shuffle

- Write to local disk
- Read from upstream TaskManager

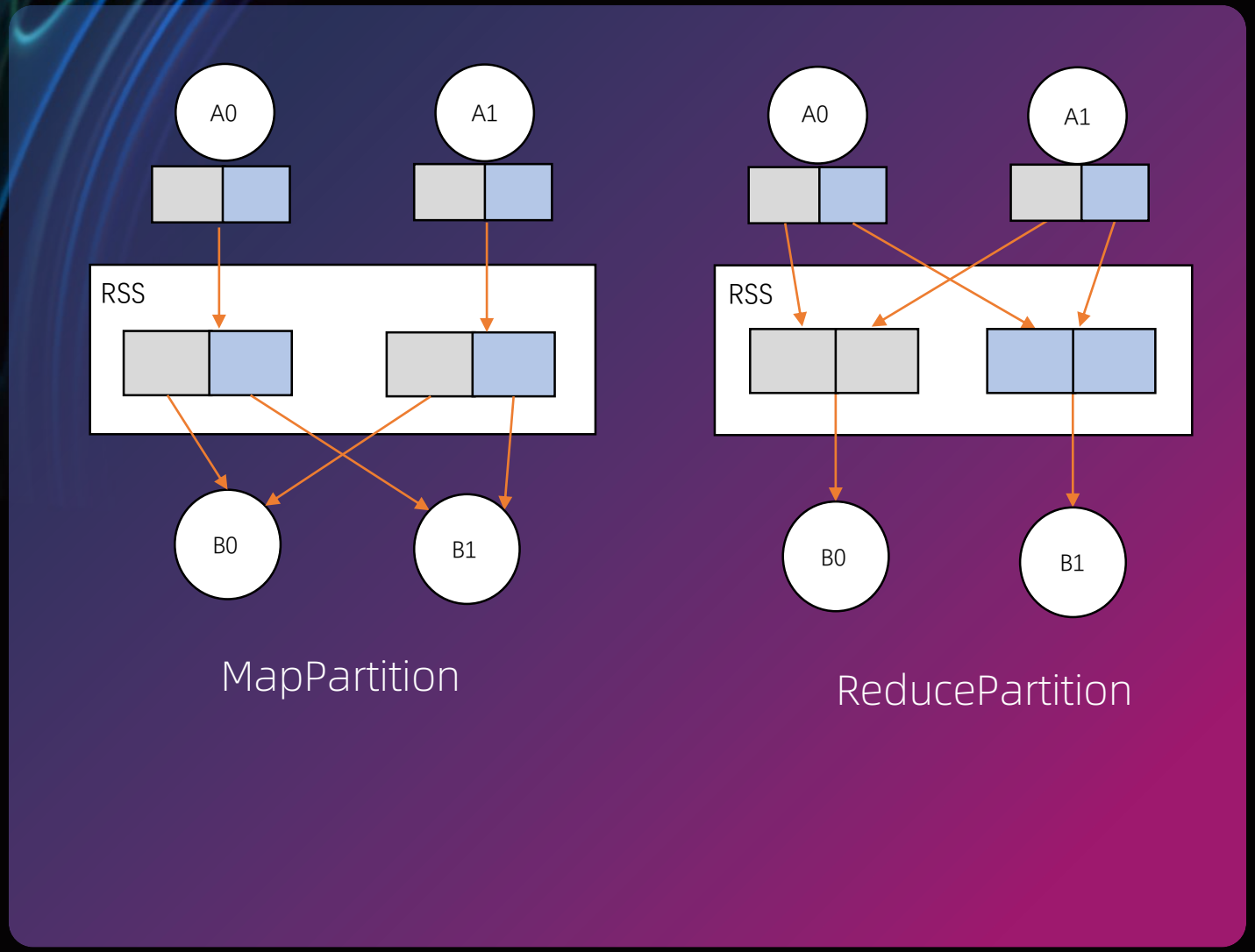
Remote shuffle service

- Write to remote shuffle worker
- Read from remote shuffle worker

Remote shuffle service



Map Partition VS Reduce Partition



Map Partition VS Reduce Partition

- ✓ Reduce Partition 优点：下游消费是顺序读，避免随机小 I/O，同时减少磁盘压力
- ✓ Reduce Partition 缺点：存储开销大，需要多副本，为了避免数据不可用的情况下重新拉起所有上游map

更多内容请关注
Flink Forward Asia 2022 - 生产实践专场
快手 Flink 的稳定性和功能性扩展

稳定性问题3：离线任务稳定性差

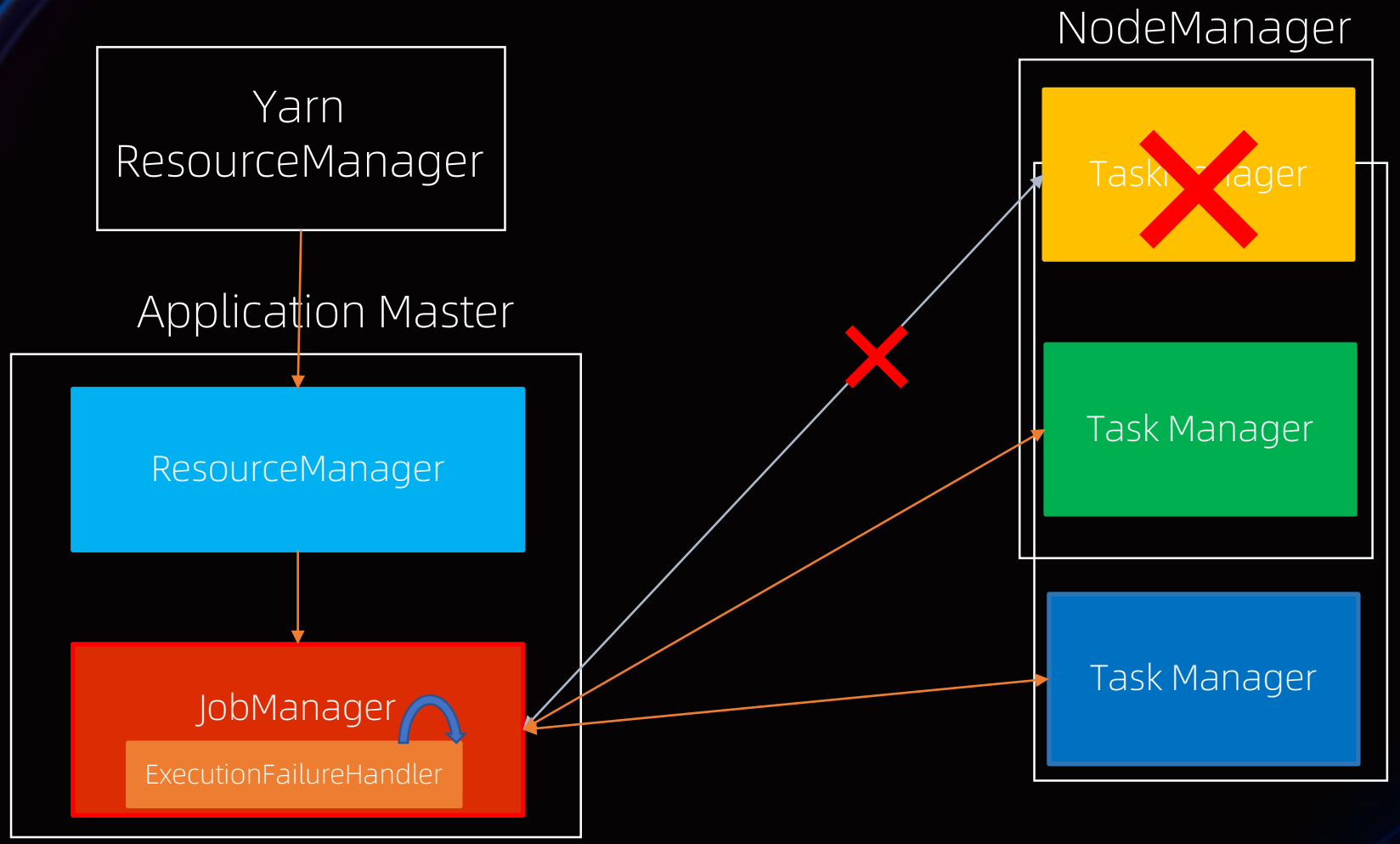
中低优任务频繁失败重启

- 离线集群开启资源抢占，中低优任务的资源频繁被抢占，导致离线任务多次重启

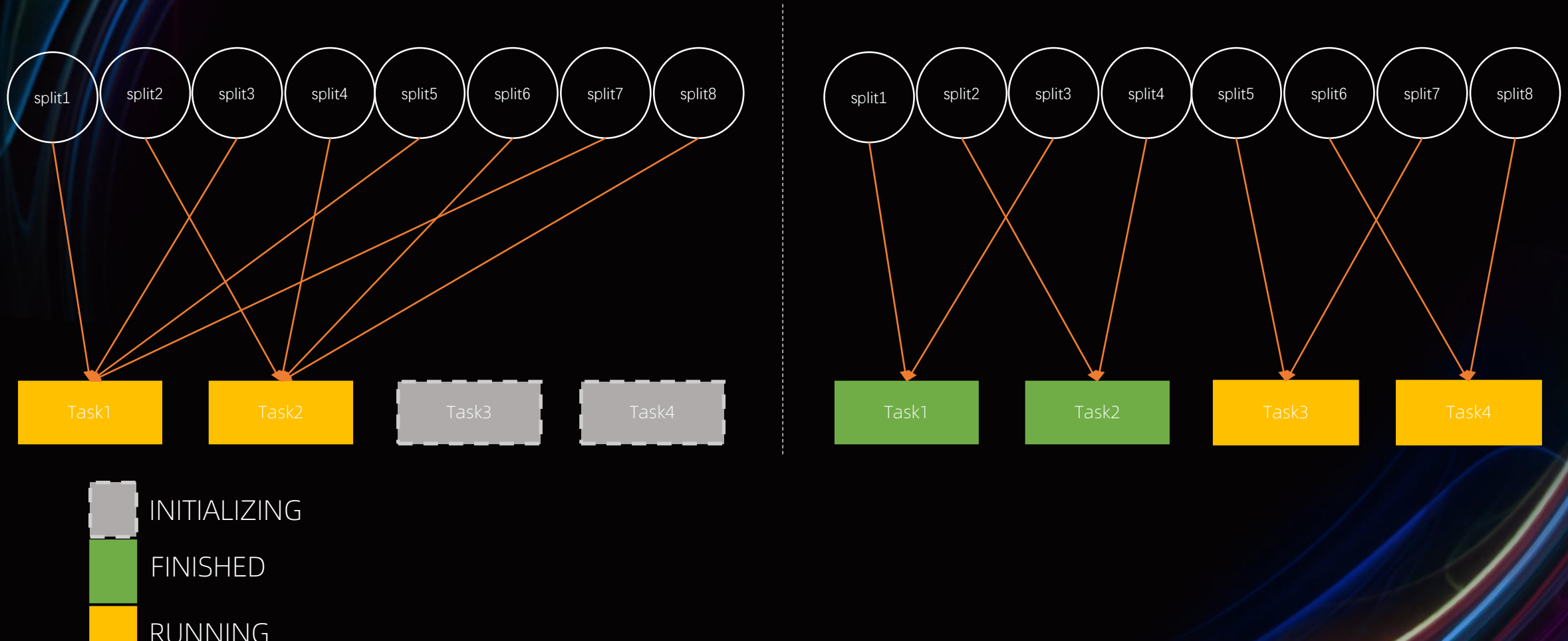
失败恢复开销大

- 离线集群资源紧张，任务可能只能申请到部分资源，已经运行的任务处理太多Splits，一旦发生异常，恢复代价大

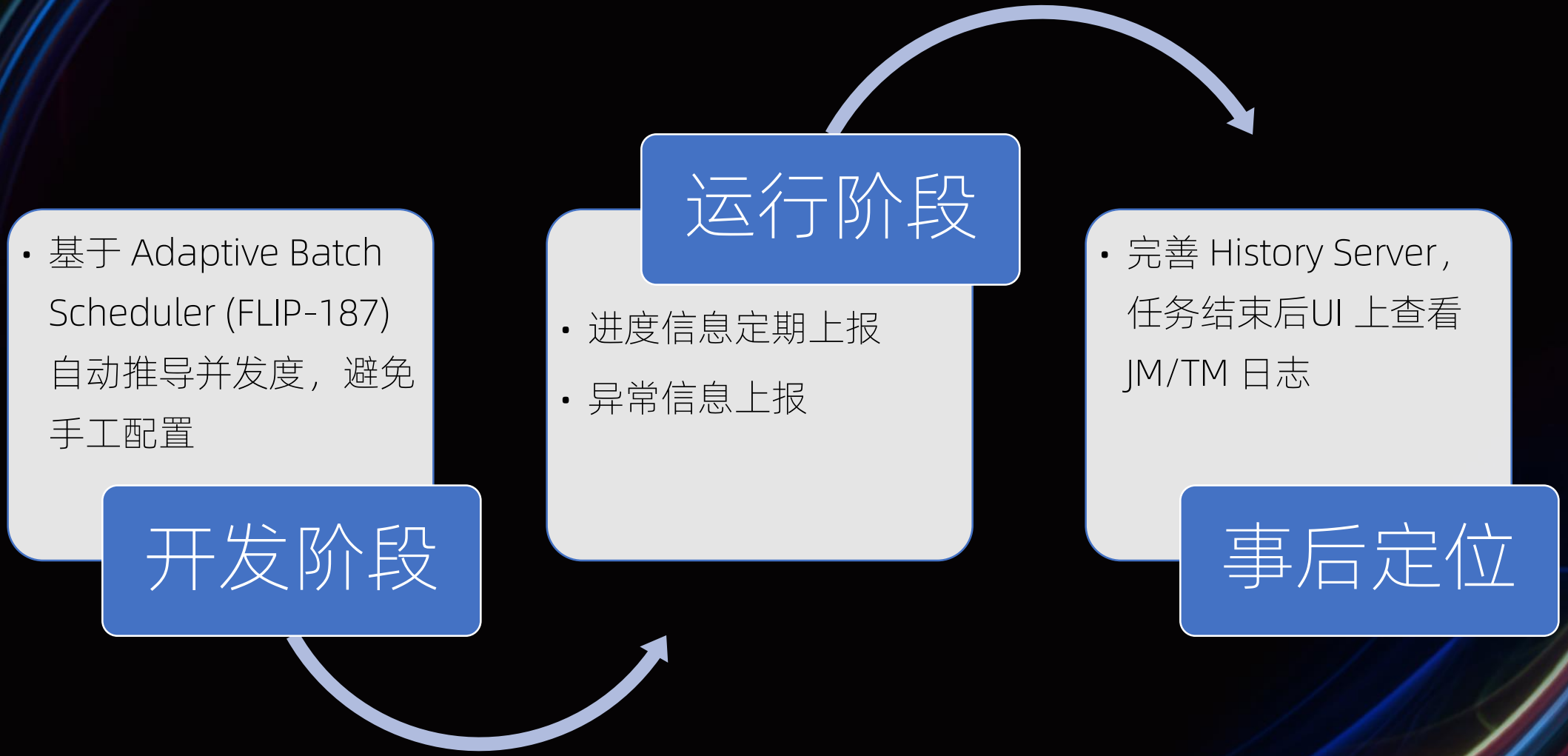
对部分异常（资源抢占类）异常自动重试



限制单个 Task 处理的 splits 个数

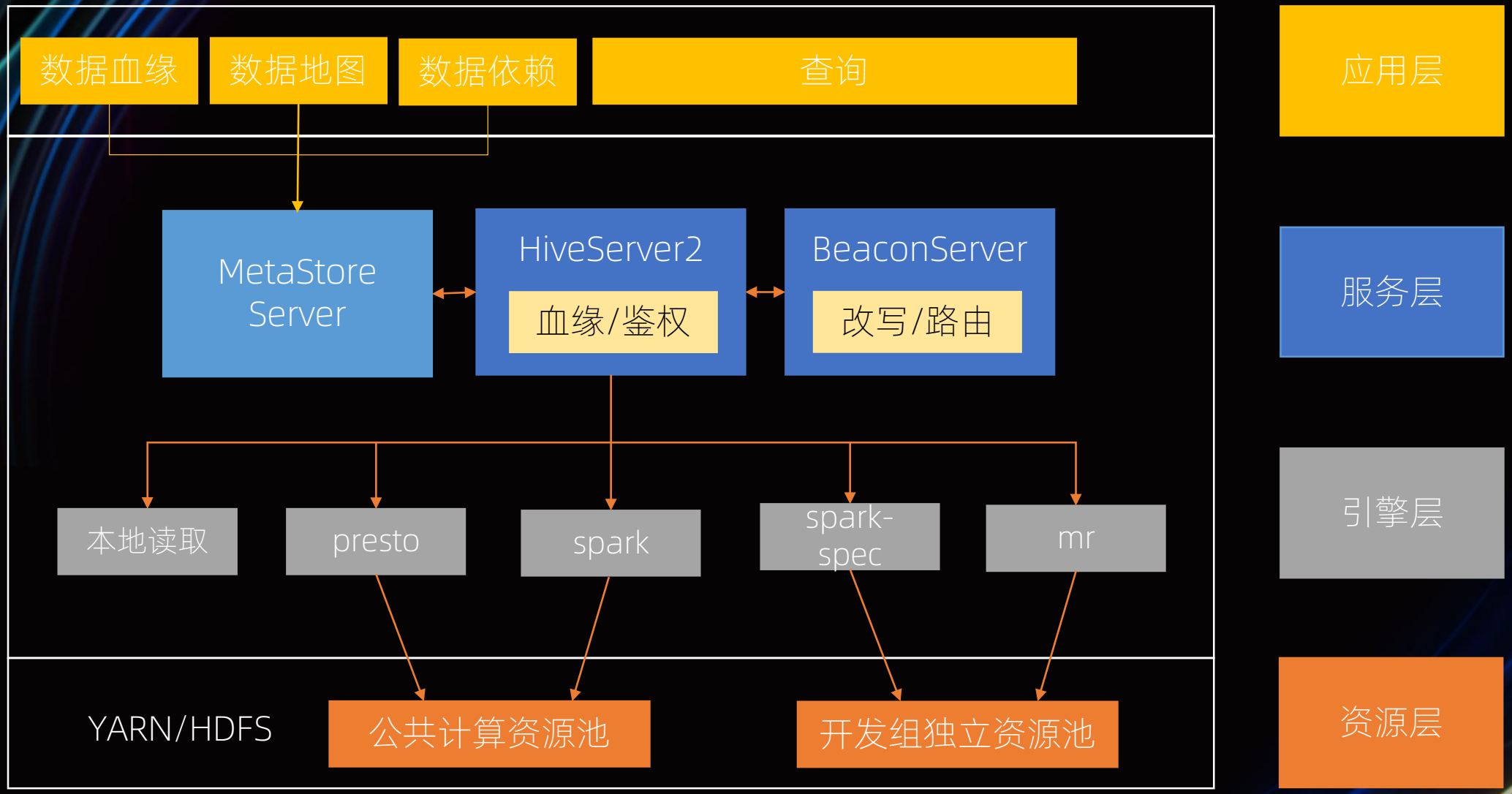


易用性


运行阶段
• 基于 Adaptive Batch Scheduler (FLIP-187) 自动推导并发度，避免手工配置
开发阶段
• 进度信息定期上报
• 异常信息上报
• 完善 History Server，任务结束后UI 上查看 JM/TM 日志
事后定位

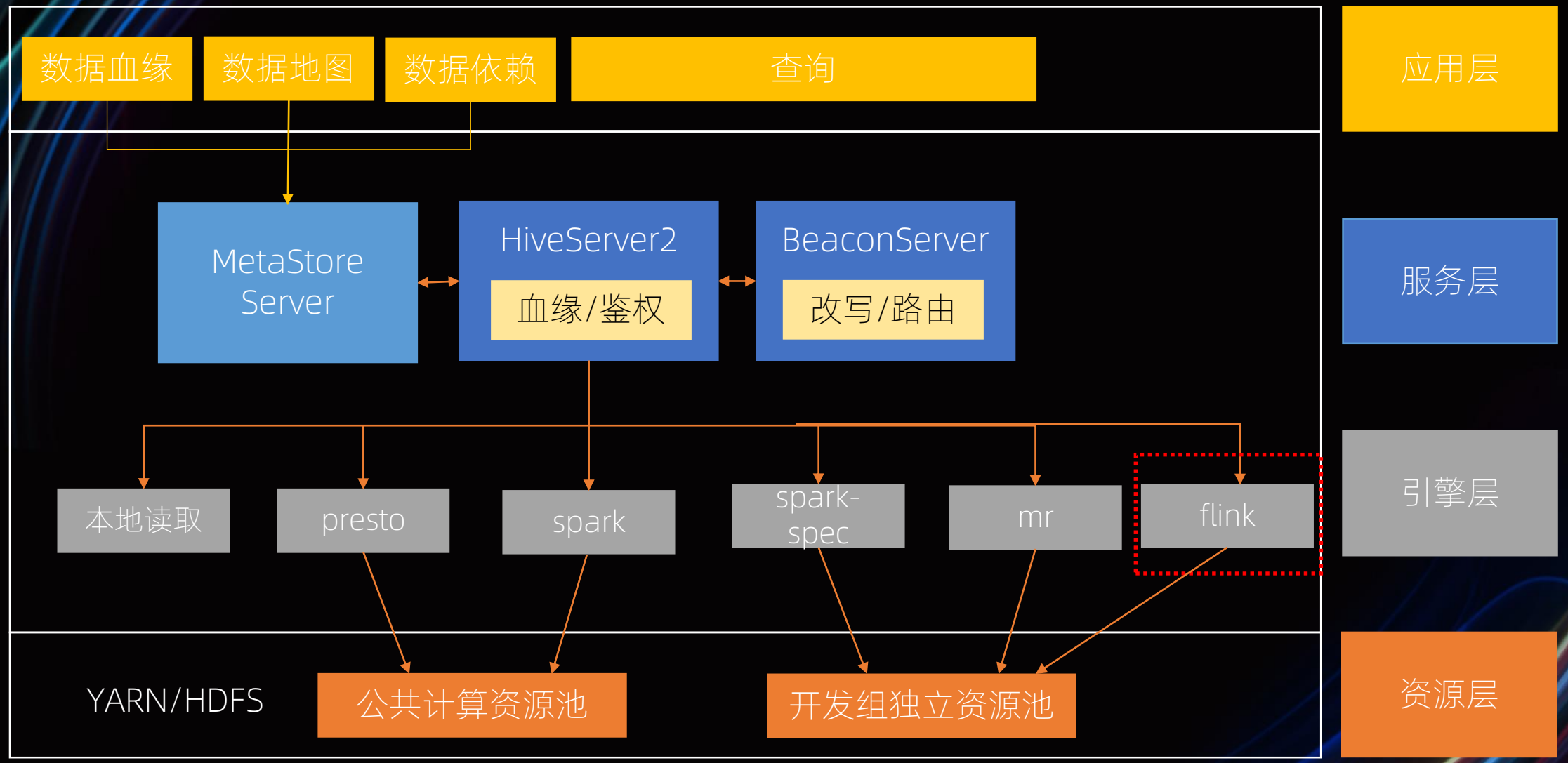
如何大规模落地 Flink Batch

快手离线生产引擎



引自 Hive 实践 | Hive MetaStore 在快手遇到的挑战与优化

Flink 接入离线生产引擎



Flink 接入离线生产引擎：关键工作

引擎能力增强	产品接入	自动化流程
• Hive dialect 兼容性 • Hive connector 能力	• 扩展 HiveServer2 和 Beacon Server，接入 Flink • 接入公司鉴权体系	• 接入双跑平台 • 监控&&报警流程

更多内容请关注
Flink Forward Asia 2022 - 生产实践专场
HiveSQL 迁移 FlinkSQL 在快手的实践

第一阶段的进展： 工作成果

3000+
作业数目

特征生产

数据集成

离线生产引擎

4 第二阶段的挑战

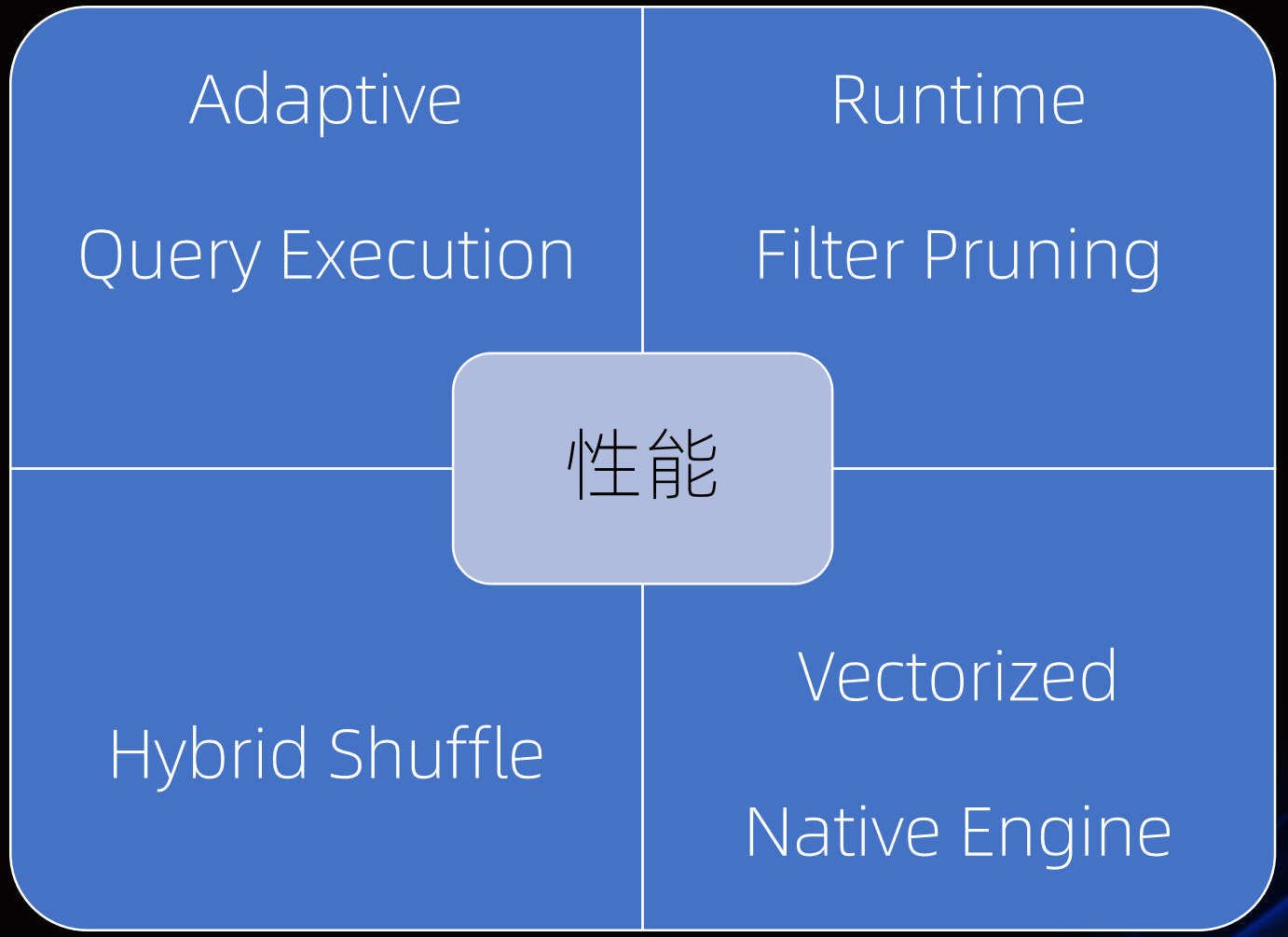
第二阶段的挑战1：手动指定运行模式

runtime mode
batch/streaming

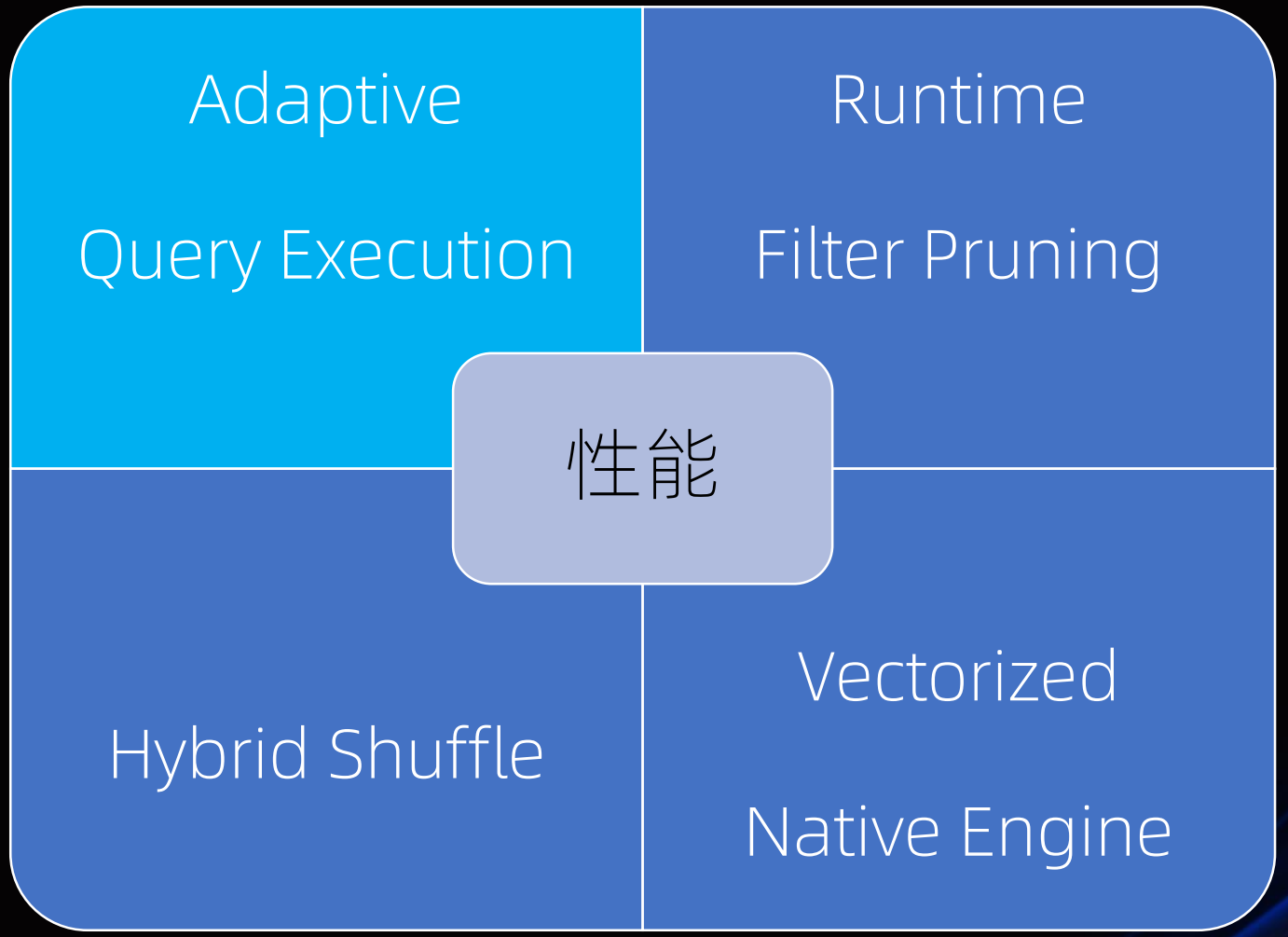
batch shuffle mode
pipeline/blocking/hybrid

scheduler
default/adaptive batch
enable speculative execution

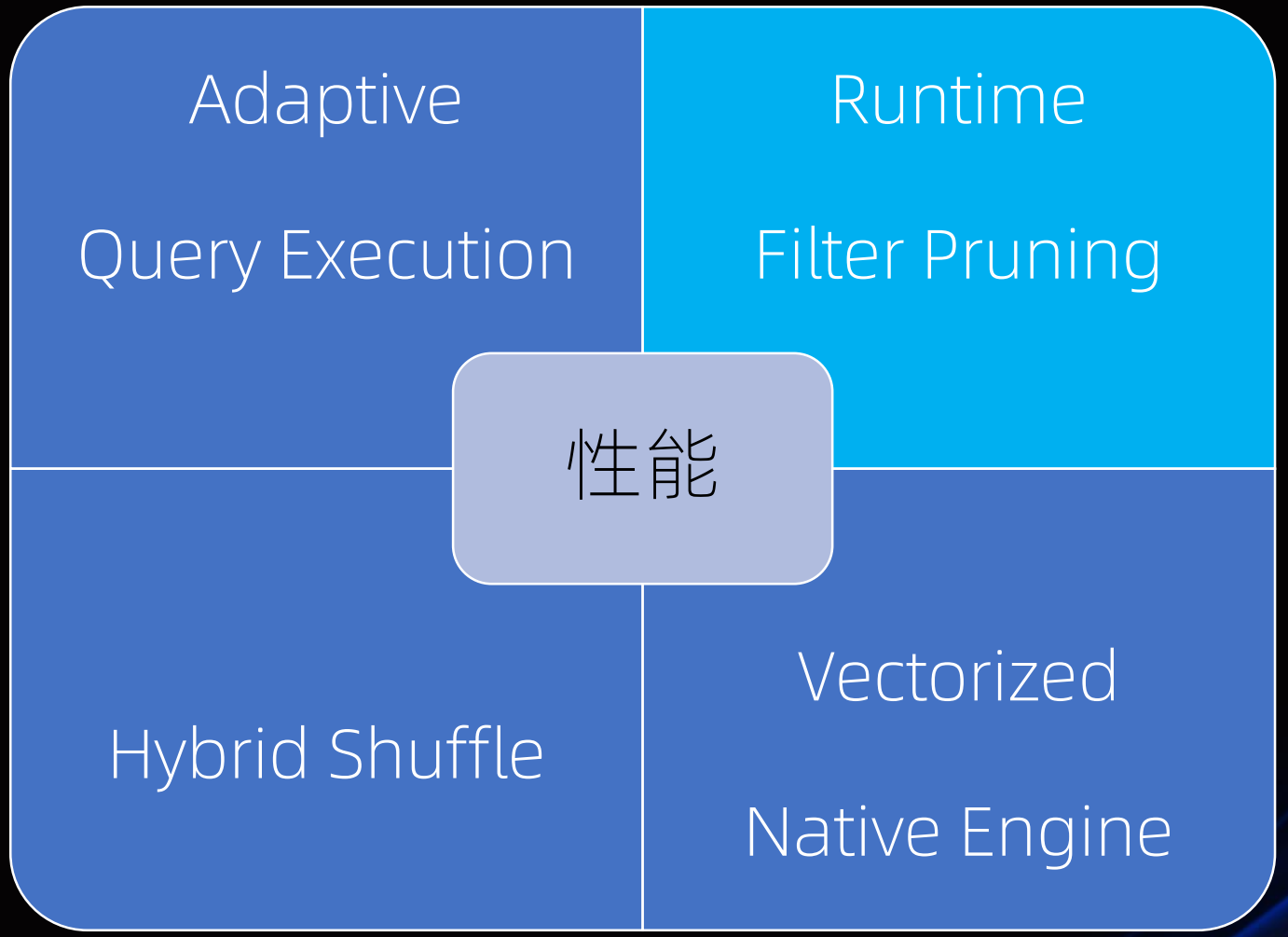
第二阶段的挑战2：批处理的性能



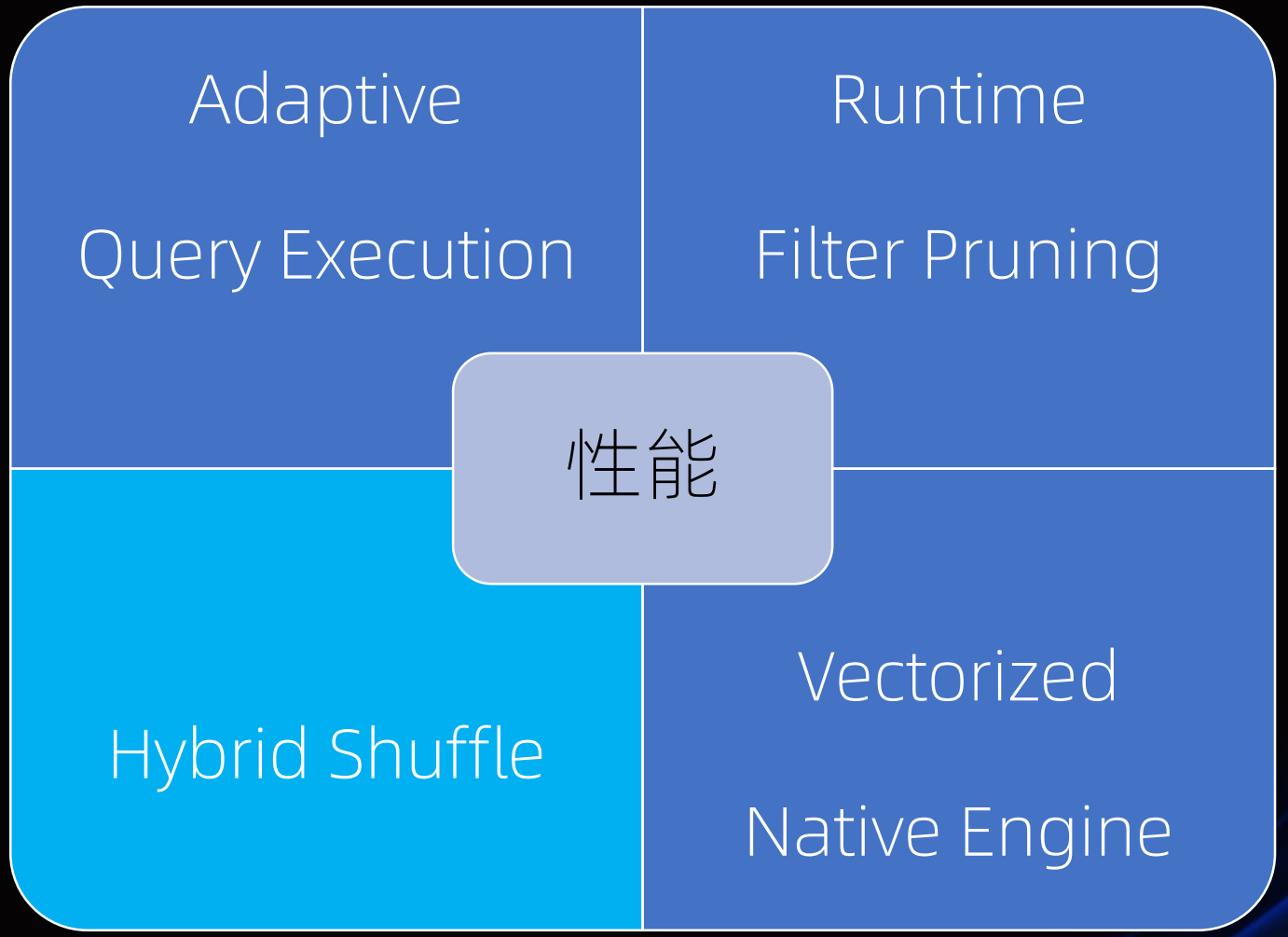
第二阶段的挑战2：批处理的性能



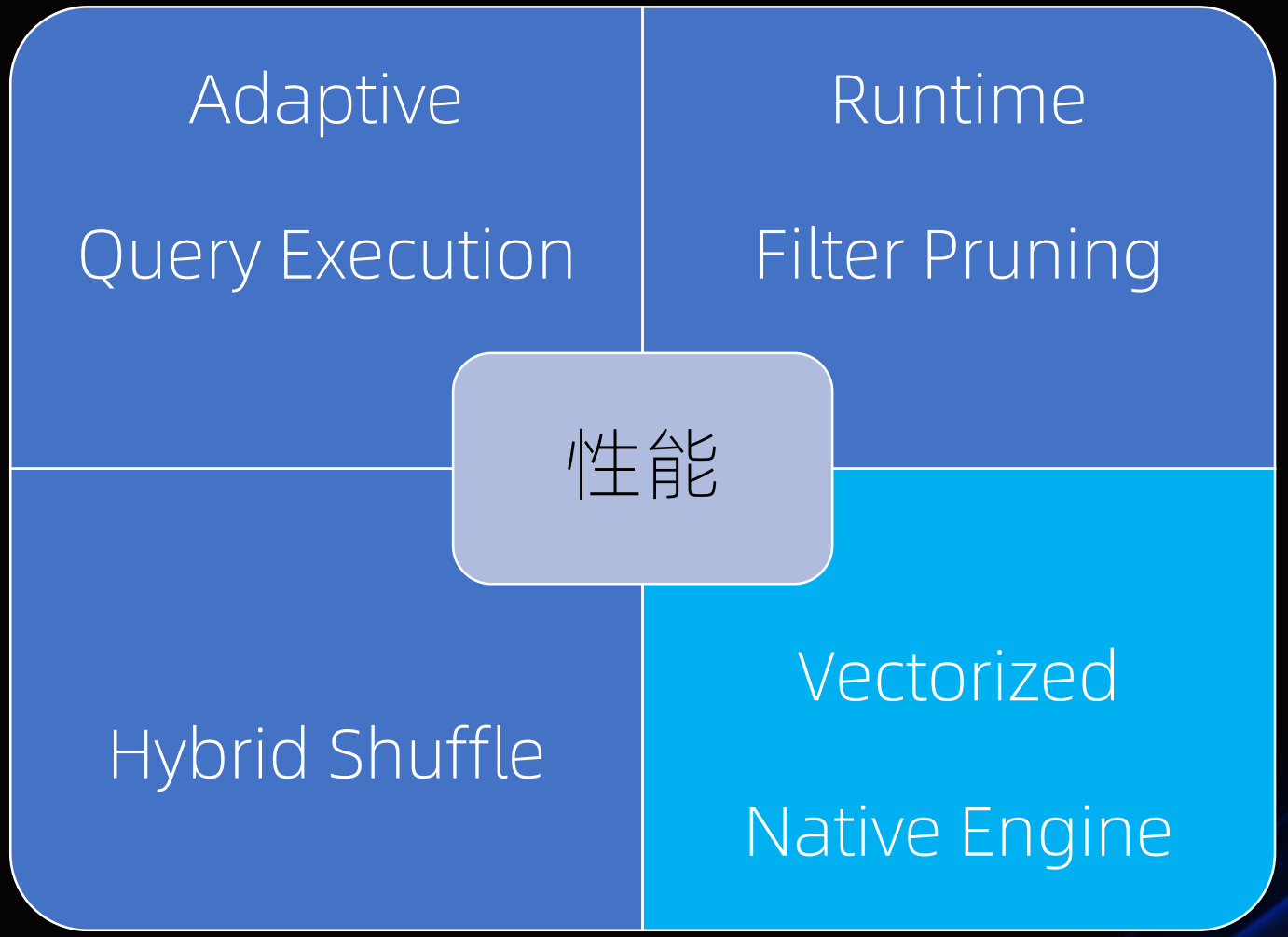
第二阶段的挑战2：批处理的性能



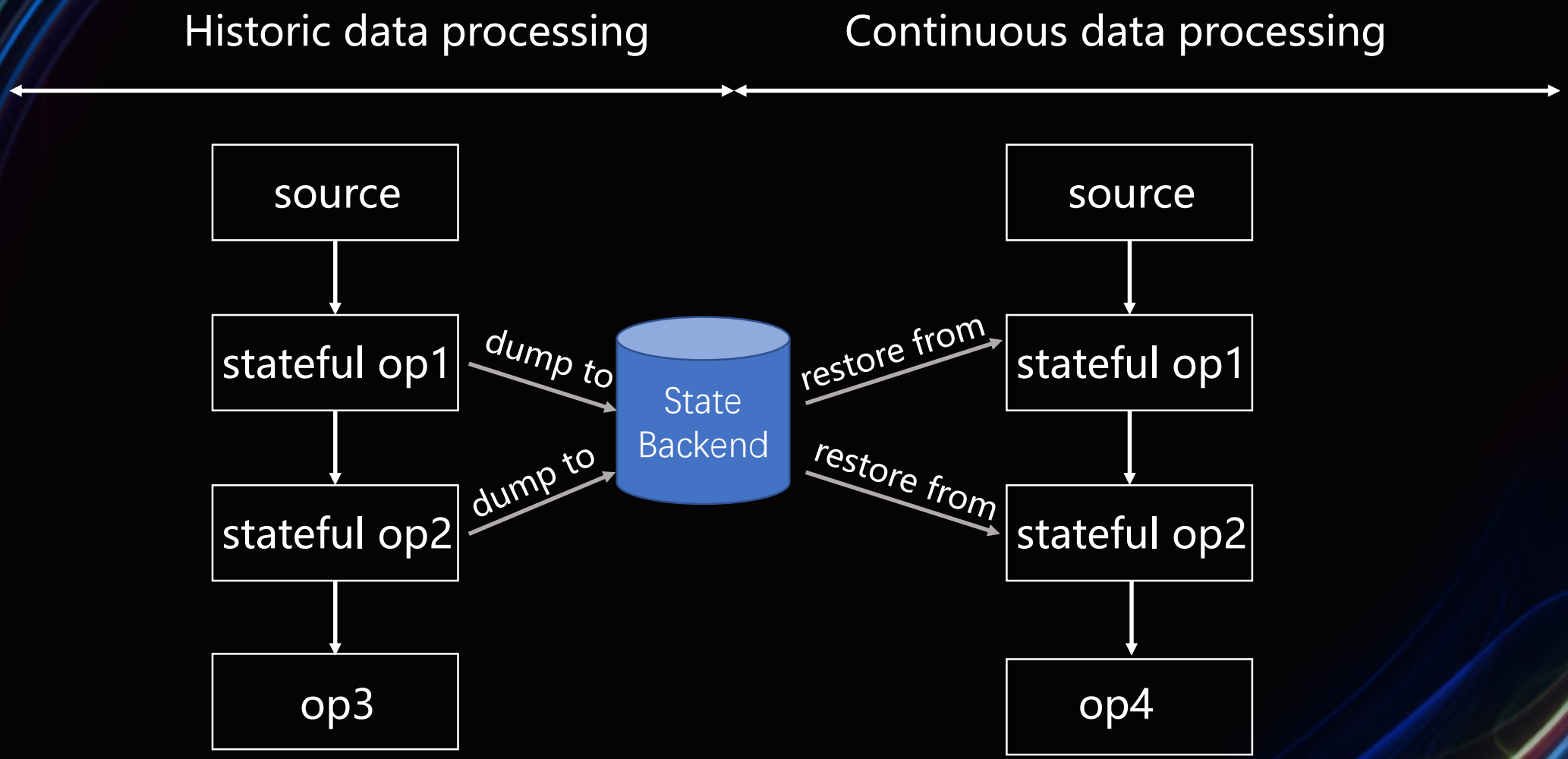
第二阶段的挑战2：批处理的性能



第二阶段的挑战2：批处理的性能



第二阶段的挑战3：流批混跑- 状态无法衔接



第二阶段的挑战4：流批混跑-存储的割裂

缺点

用户抽象出逻辑表

平台层路由到物理表

FLINK FORWARD # ASIA 2022 ONLINE | 实时即未来

THANK YOU

谢 谢 观 看