DORIS | Webinar Release

Apache Doris 4.0

企业级 AI 应用解读（四）

观看直播

01月15日 周四 19:30-20:30

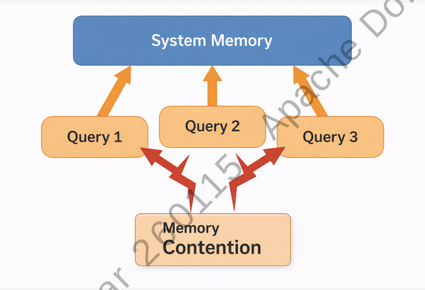
Apache Doris 基于全局内存视角的大查询落盘机制

胡胜刚 – Apache Doris Committer、飞轮科技资深技术专家

目录

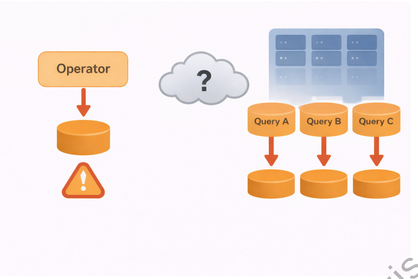
为什么需要基于全局内存视角

并发查询的内存竞争是常态



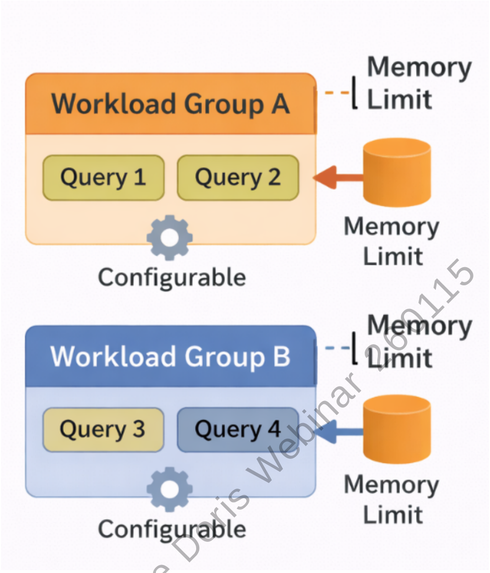
- 系统内存是**全局共享资源**，查询之间天然存在竞争
- 多个查询的内存需求具有**叠加效应**
- 单查询 / 单算子视角下的内存安全，**无法推导系统级安全**
- 局部最优 ≠ 全局稳定
- 必须从全局内存使用情况进行统一判断

单算子/单查询视角无法保证系统稳定



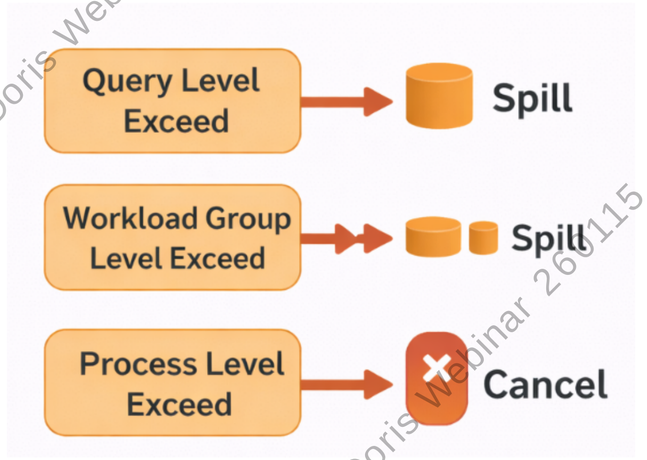
- 单算子、单查询只能感知**局部内存使用情况**
- 难以感知：其他查询的内存占用；内存峰值在时间上的叠加
- 局部判断“无需落盘” ≠ 系统整体安全
- 内存风险往往在**全局层面集中爆发**
- 需要全局内存视角统一决策落盘时机

基于 Workload Group 的全局内存管理



每个 Workload Group：

- 配置独立内存上限
- 可配置 group 内 query 的内存分配策略



1.**Query 级超限**

- 暂停 query 的执行，交给 workload group 处理

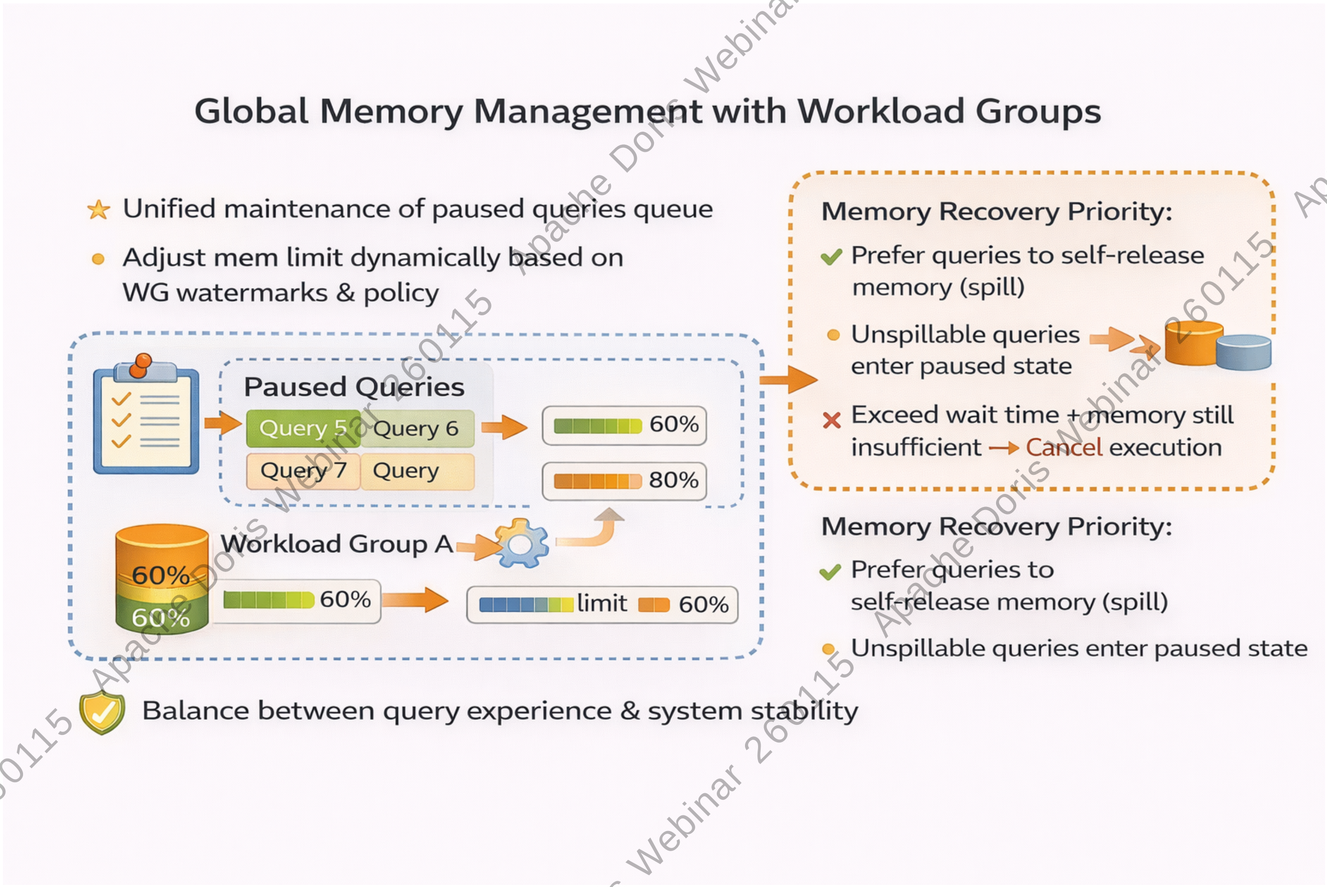
2.**Workload Group 级超限**

- 暂停 group 内所有 query
- 选择可落盘 query 执行落盘
- 协调释放 group 内存

3.**Process 级超限**

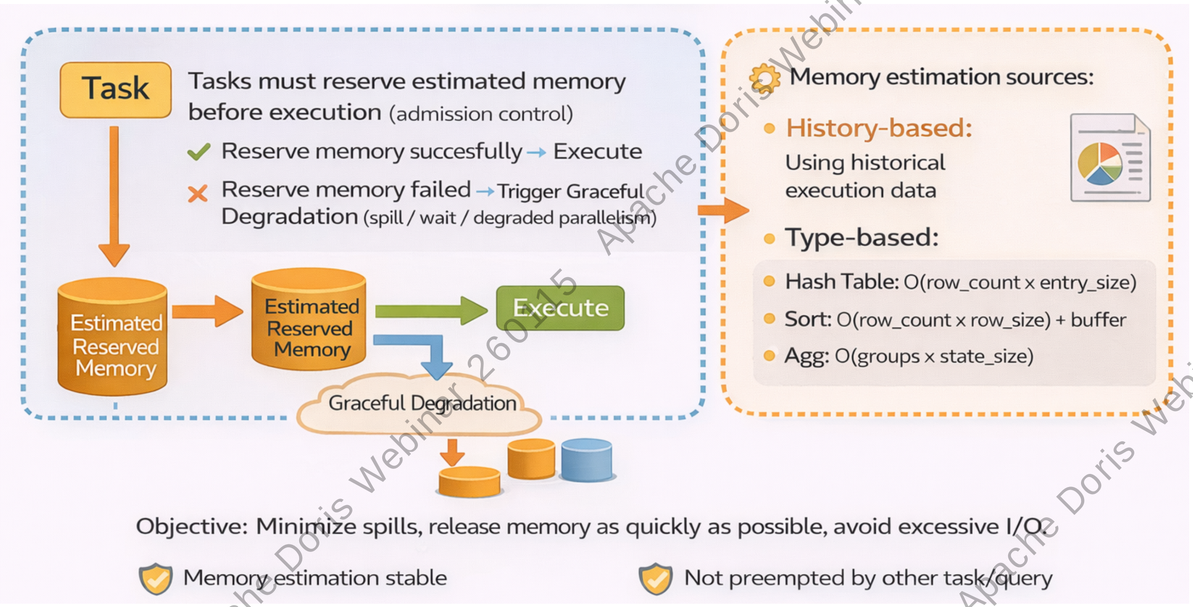
- 直接 cancel 最大内存占用的 query
- 快速释放内存，保障系统稳定

基于 Workload Group 的全局内存管理



- 统一维护 paused queries 队列
- 根据 Workload Group 水位和策略动态调整 mem limit
- 内存回收优先级：
  1. 优先让 query 自行释放内存（spill）
  2. 无法落盘的 query 进入等待状态
  3. 等待超时且内存仍不足 → 取消执行
- 在查询体验与系统稳定性之间实现平衡

Memory reserve 机制

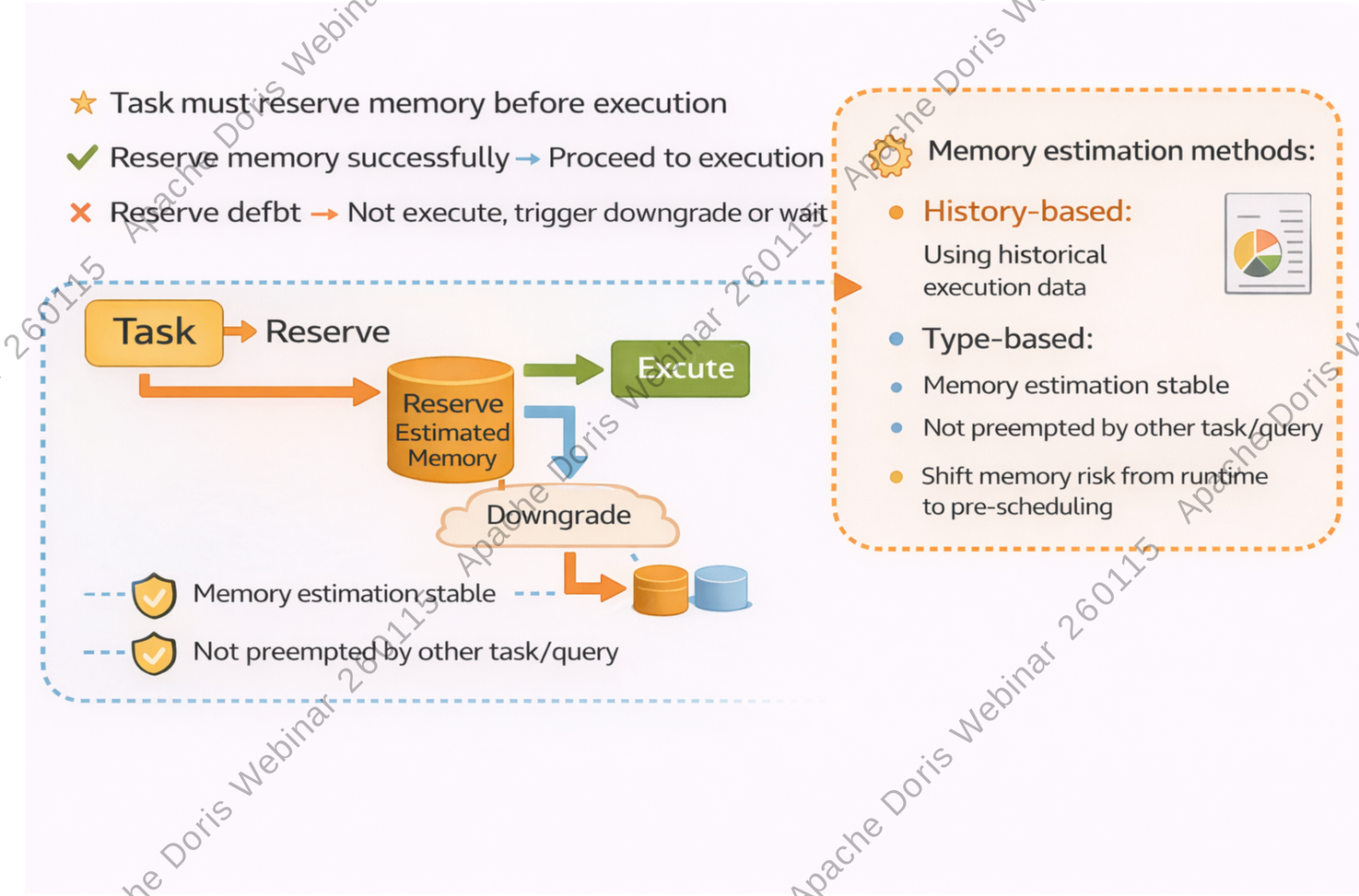


Task 执行前必须 reserve 预计内存（准入机制）

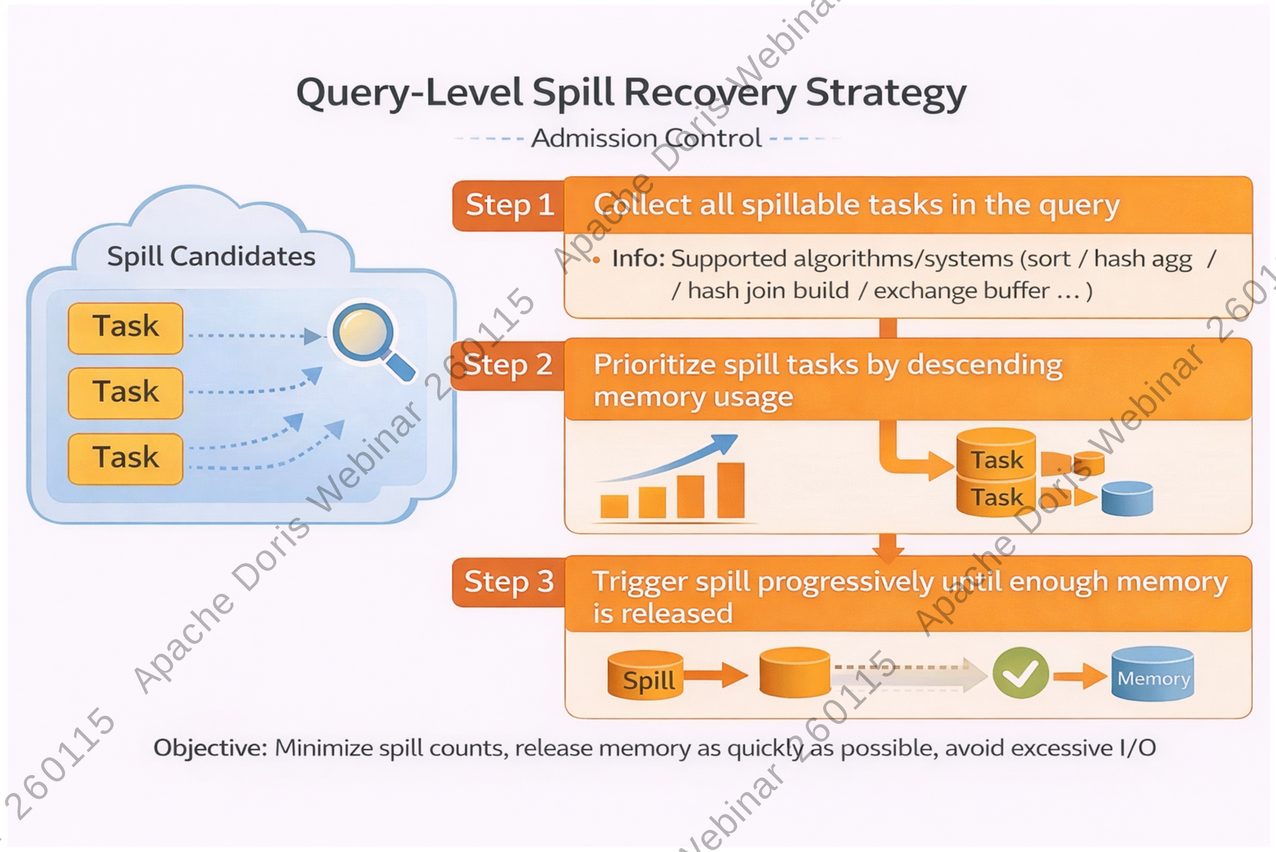
- reserve 成功 → 执行
- reserve 失败 → 触发降级（spill / 等待 / 降低并发）

- 内存预算来自估算：
  - History-based：基于历史执行数据
  - Type-based：基于任务类型

Memory reserve 机制的作用



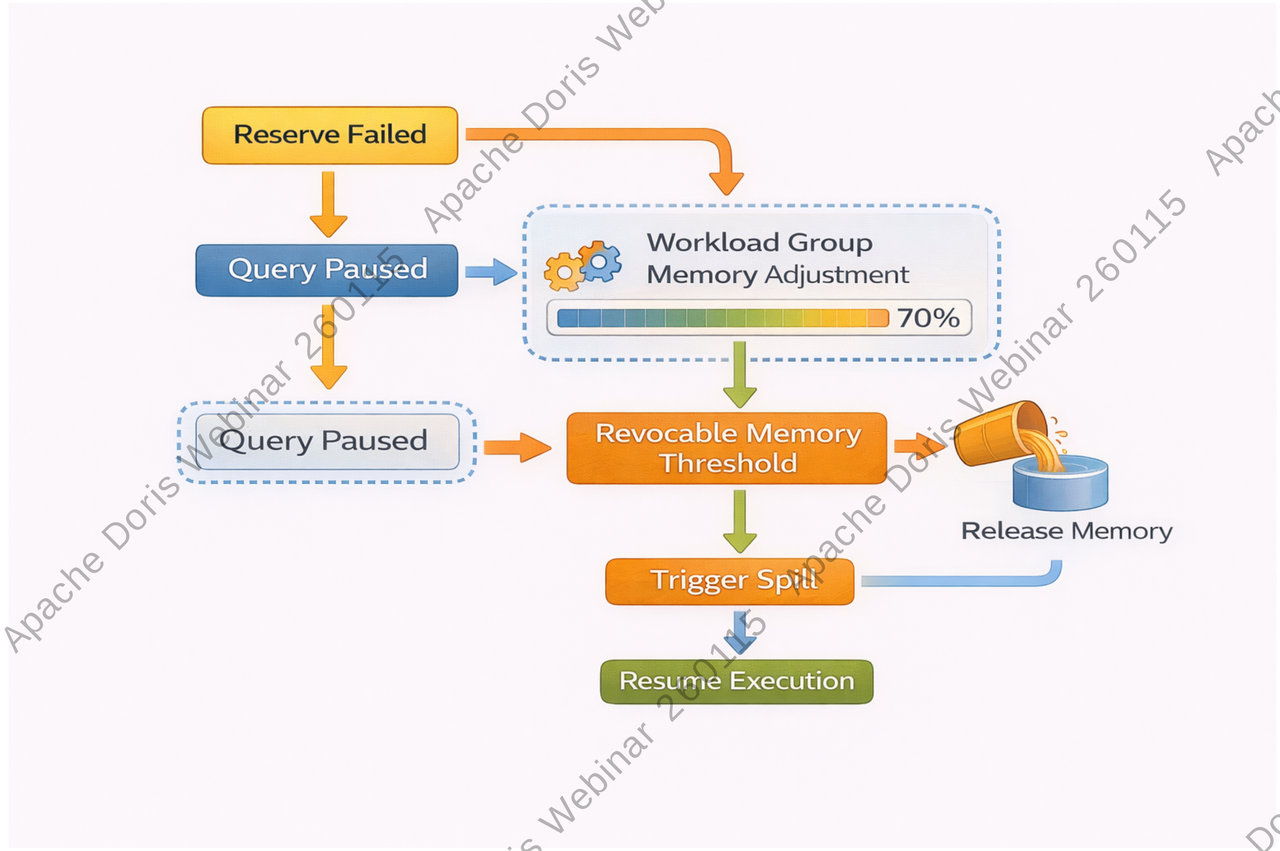
Query 级落盘回收策略



- Step1：收集 Query 内所有可落盘 task（Spill candidates）
- Step2：按可落盘内存大小排序（收益优先）
- Step3：依次触发落盘 → 每次触发后评估内存

  → 直到释放足够内存为止
- 目标：**最少落盘次数，最快释放内存，避免过度 I/O**

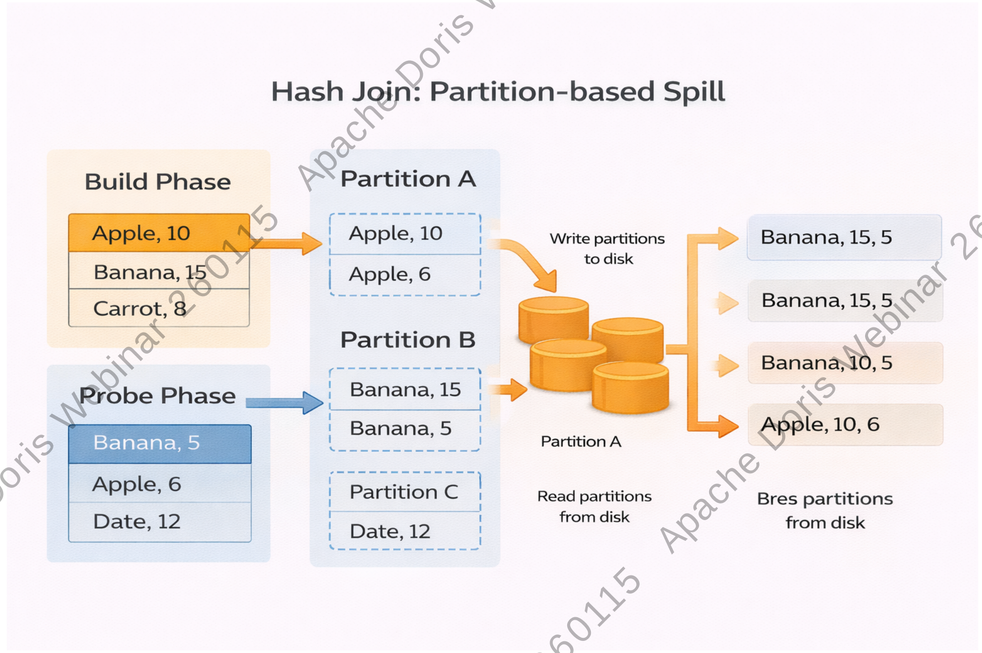
Spill 的触发机制



Spill 的触发机制

- Reserve 失败 → Query 进入 paused 状态
- Workload Group 根据组内情况动态调整内存限制
- Query 的可回收内存达到阈值 → 触发 Spill
- Spill 由全局内存压力驱动，而非局部决策

算子落盘 hash join



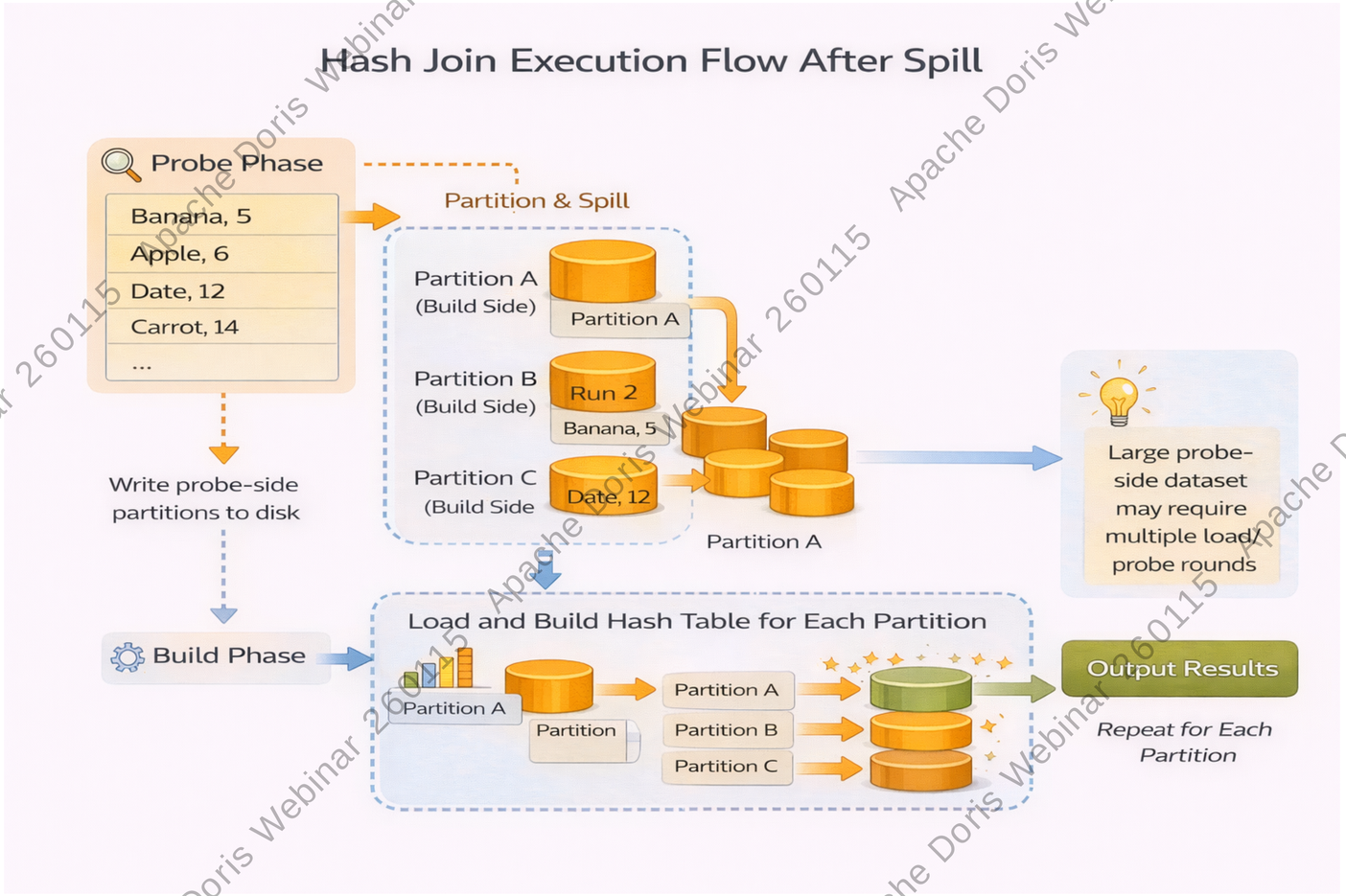
Hash Join 算子：基于 Partition 的落盘

- Hash Join 的 **build / probe 阶段**均支持落盘
- 当内存压力触发 spill 时：
  - 按 join key 对数据进行 **partition**
  - 将部分 partition 写入磁盘
- 后续通过逐个 partition 方式完成 join

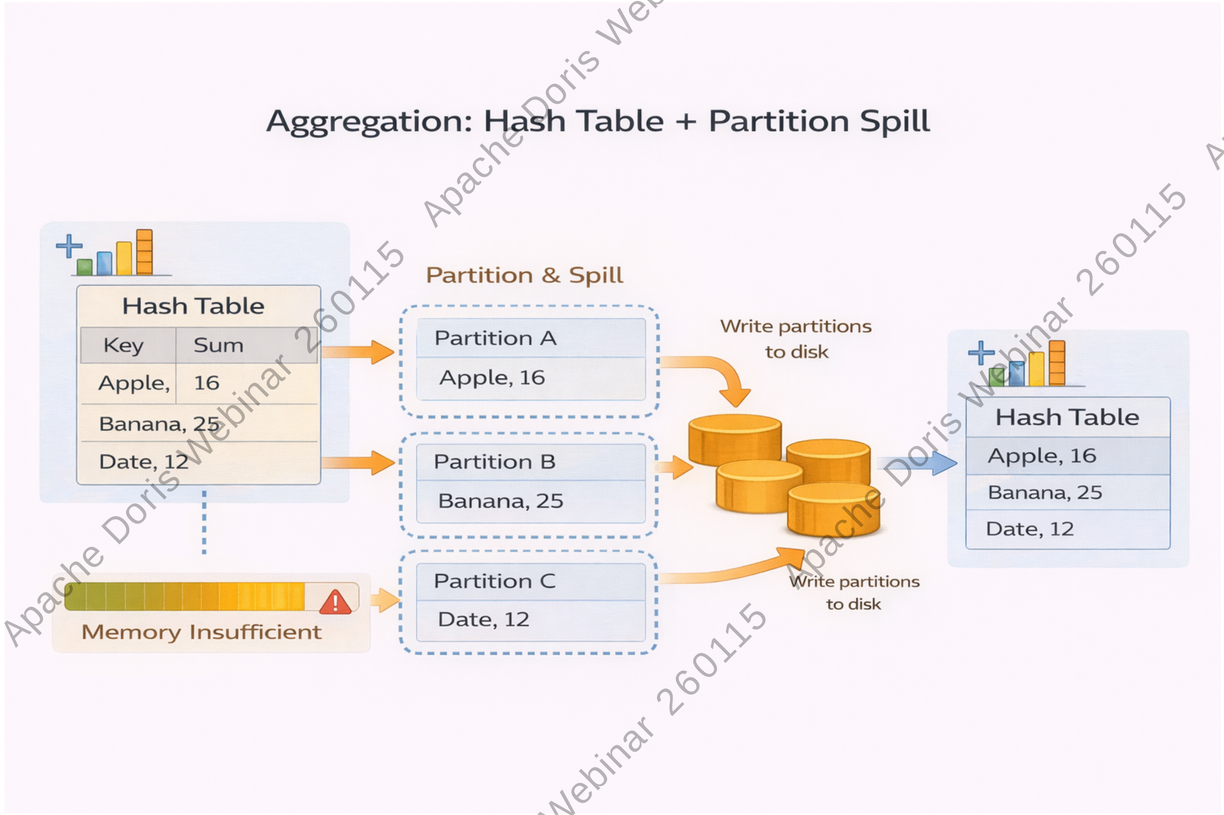
特点：

- 落盘粒度清晰
- 适合处理大规模 join 数据
- 可在保证正确性的同时显著降低内存占用

Hash Join 落盘之后的执行



算子落盘 Agg



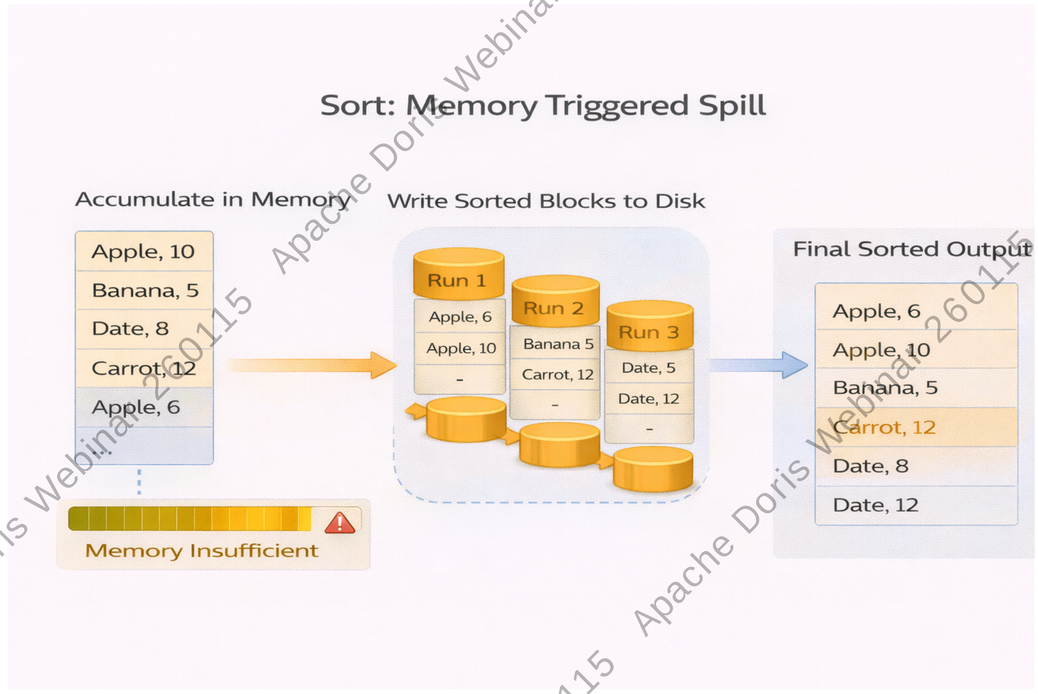
Aggregation 算子：Hash Table + Partition 落盘

- Agg 算子在执行过程中 **实时构建 hash table**
- 当内存不足时：
  - 将部分聚合数据按 key **partition**
  - 将 partition 后的数据落盘
- 后续对落盘数据重新加载并继续聚合

特点：

- 保持在线聚合能力
- 内存压力下逐步退化
- 避免一次性中断查询执行

算子落盘 Sort

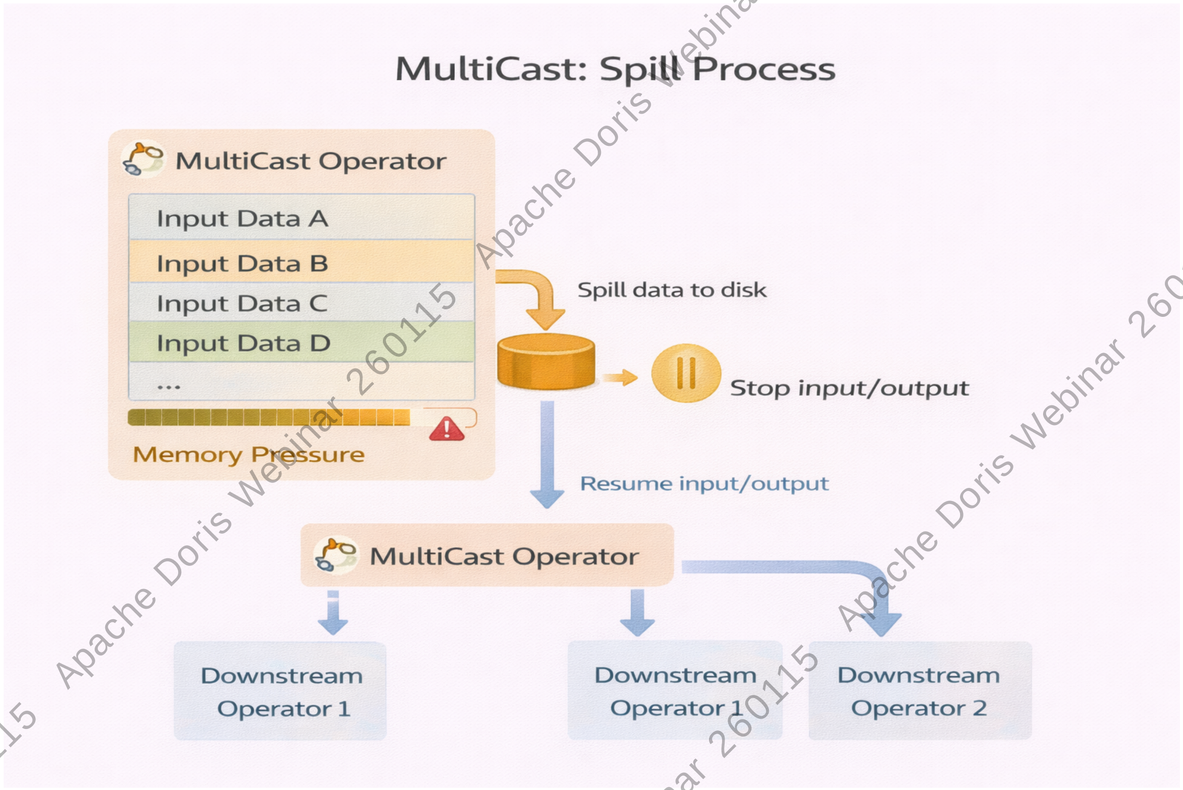


- Sort 算子在内存中持续累积待排序数据
- 当数据量达到阈值或内存不足时：
  - 将当前数据块写入磁盘
  - 生成多个有序 runs
- 最终通过多路归并完成全局排序

特点：

- 经典外部排序模型
- 落盘时机明确
- 对内存使用高度可控

MultiCast 算子落盘

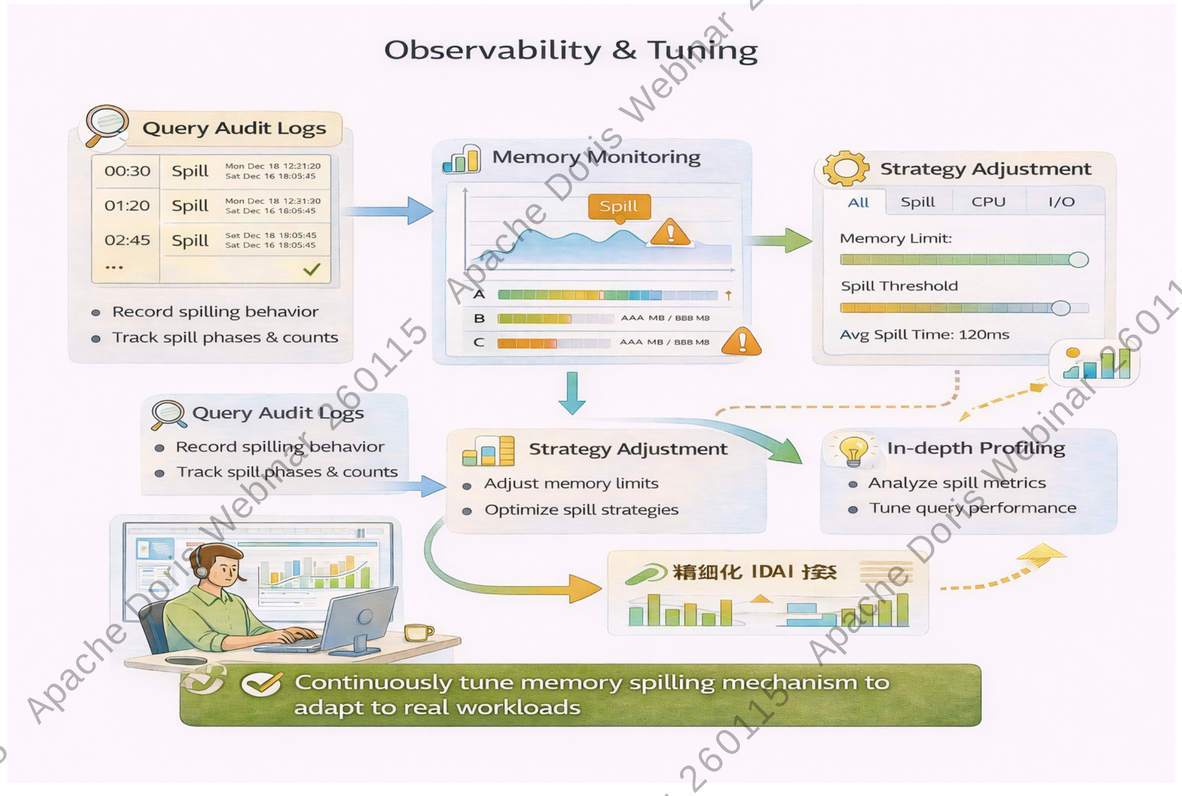


- MultiCast 算子需要将输入数据 **广播到多个下游算子**
- 不同下游算子的 **消费速度不一致**
  - 易导致数据在 MultiCast 处累积
  - 内存压力显著增加

落盘触发与执行流程：

- 当累积数据量达到阈值时，触发落盘
- 落盘过程中：
  - 暂停接收新的输入数据
  - 暂停向下游输出数据
- 当前数据落盘完成后：
  - 恢复数据接收与广播输出
  - 继续正常执行

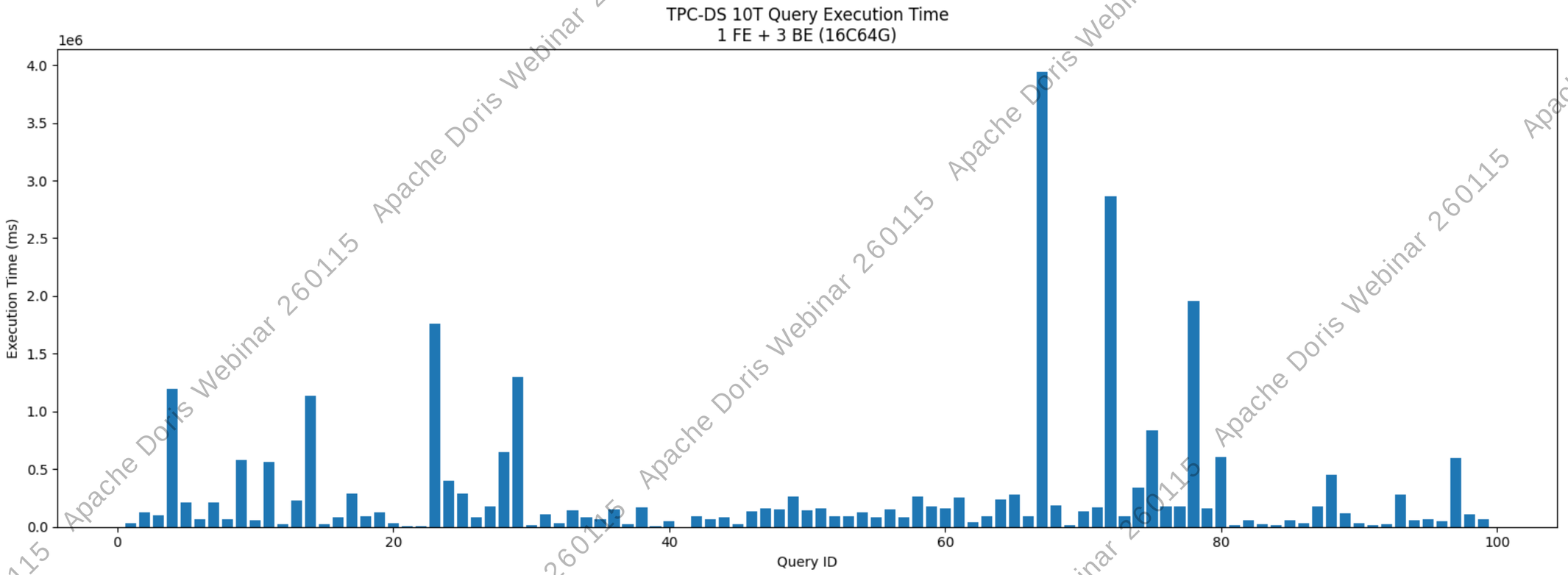
观测与调优



观测与调优（Observability & Tuning）

- **基于审计日志观测落盘行为**
  - 在审计日志中记录每个 query 的落盘情况
  - 包括是否触发落盘、落盘阶段及落盘次数
- **监控系统内存使用情况**
  - 实时监控进程及 Workload Group 级内存水位
  - 评估落盘机制对缓解内存压力的实际效果
- **基于观测结果进行策略调整**
  - 根据运行情况动态调整 Workload Group 的内存限制
  - 优化落盘触发阈值和回收策略
- **利用 Profile 指标进行精细化调优**
  - Profile 中提供丰富的落盘相关指标
  - 支持分析落盘开销、频率及对查询性能的影响

TPC-DS 10T 实测



Thanks !

DORIS | Webinar Release

回放与演讲资料获取

请关注 SelectDB 公众号发送 20260115

联系我们

www.selectdb.com

400-092-6099

微信公众号

免费试用

在线咨询

加入社区