机遇与挑战：Apache Flink 资源管理机制解读与展望

宋辛童 · 阿里巴巴 / 高级开发工程师

Apache Flink China Meetup 北京
- 2019年09月21日

Contents

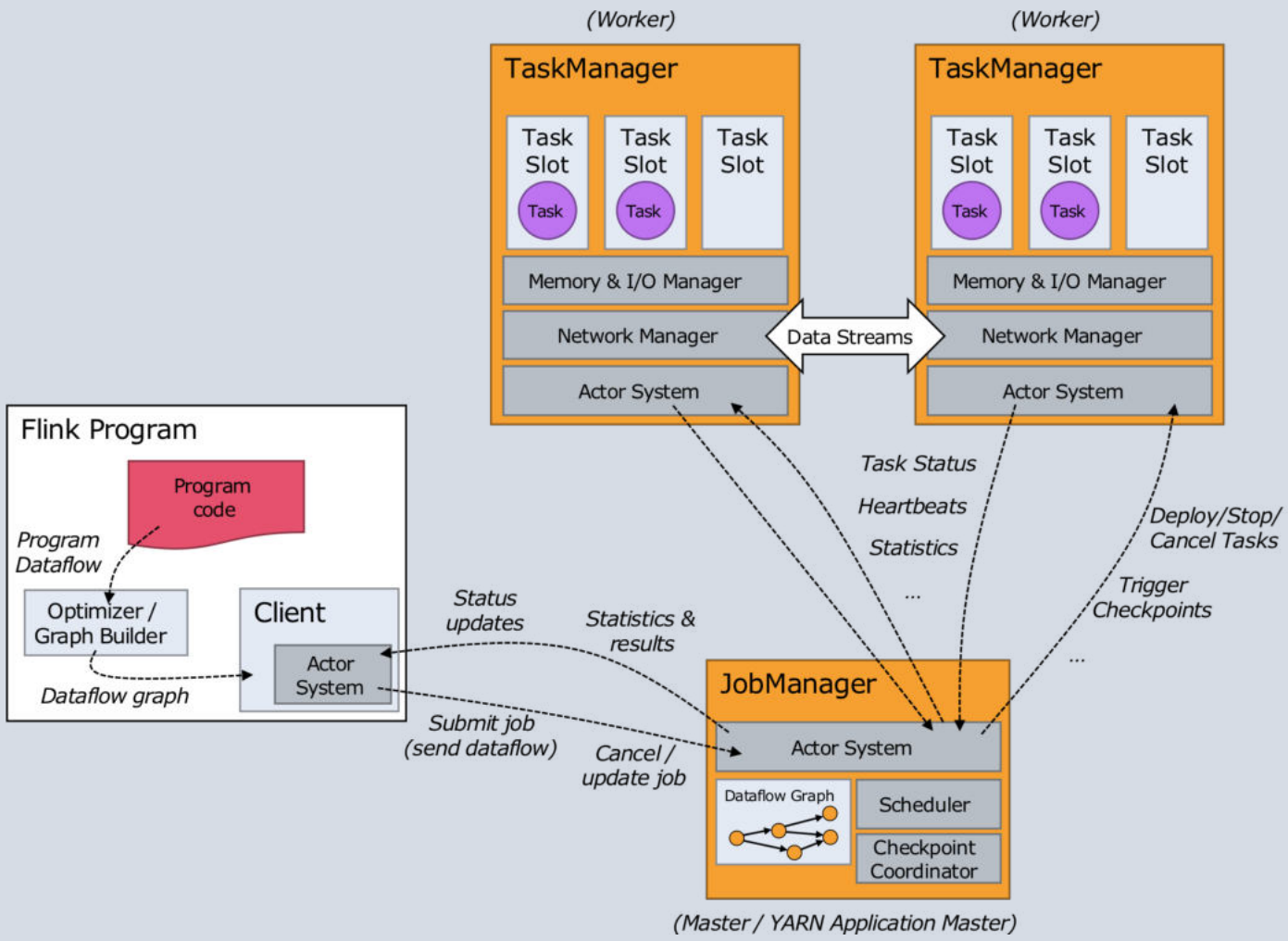
目录 >>

PART 01

Flink 资源管理

TaskManager & Slot

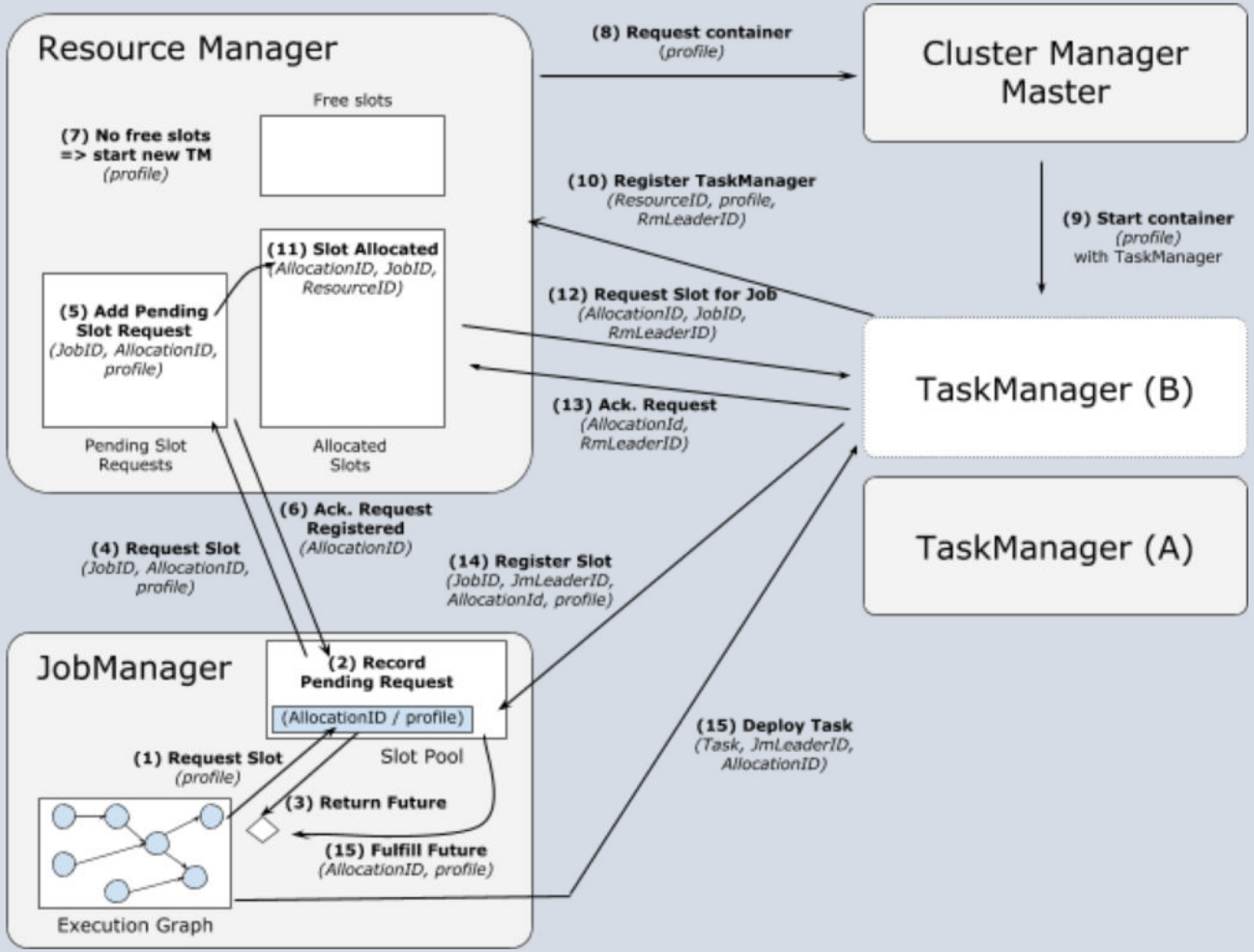
Apache Flink



- TaskManager
  - Task 的执行器
  - 进程级（容器）
- Slot
  - TM 资源的子集
  - 线程级

- Slot 间没有资源隔离
  - 例外：Managed Memory

Slot Allocation

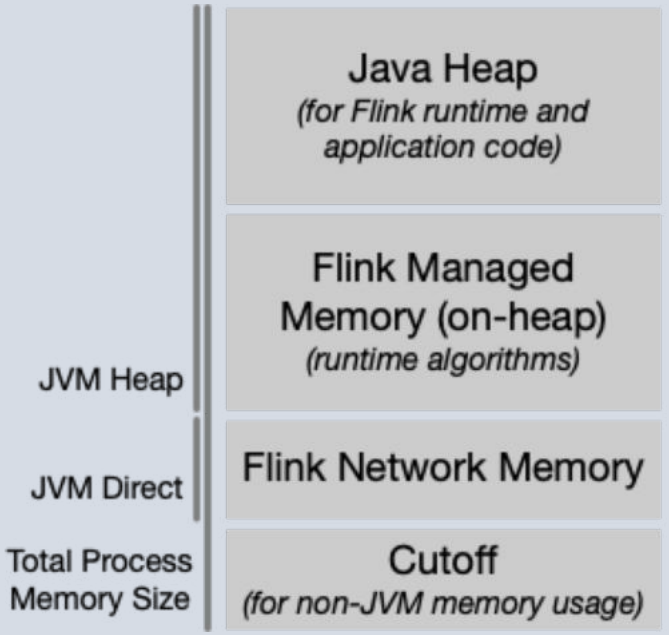


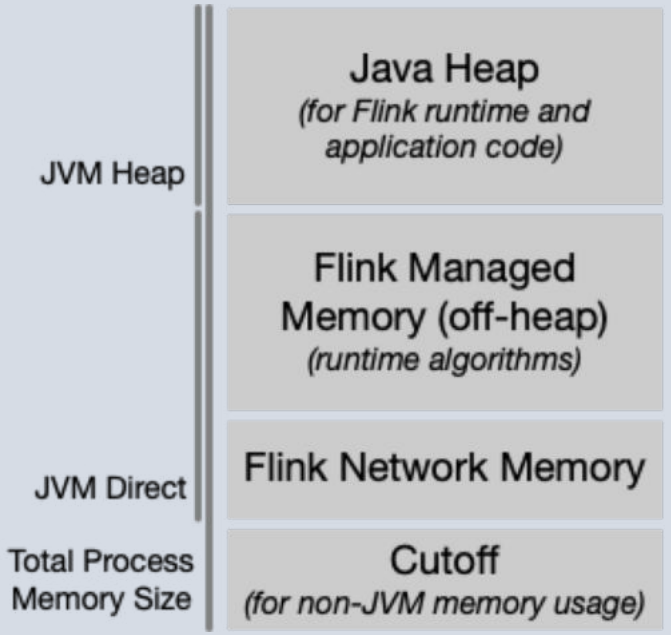
- TaskManager 资源、Slot数量由用户配置决定
- TaskManager 数量
  - Standalone – 固定数量
  - Yarn/Mesos – 按需申请

Resource Types

Apache Flink

- CPU
- Memory
  - Heap Memory
    - JVM 管理的内存
    - 用途：一般的java对象
  - Network Memory
    - Shuffle Service 中 network buffers 占用的 (direct) 内存
  - Cutoff
    - 容器化环境中，JVM以外的内存开销

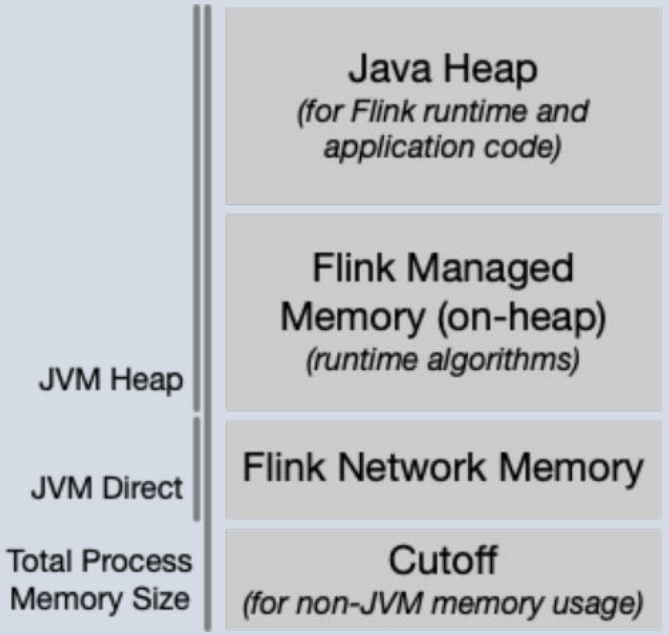


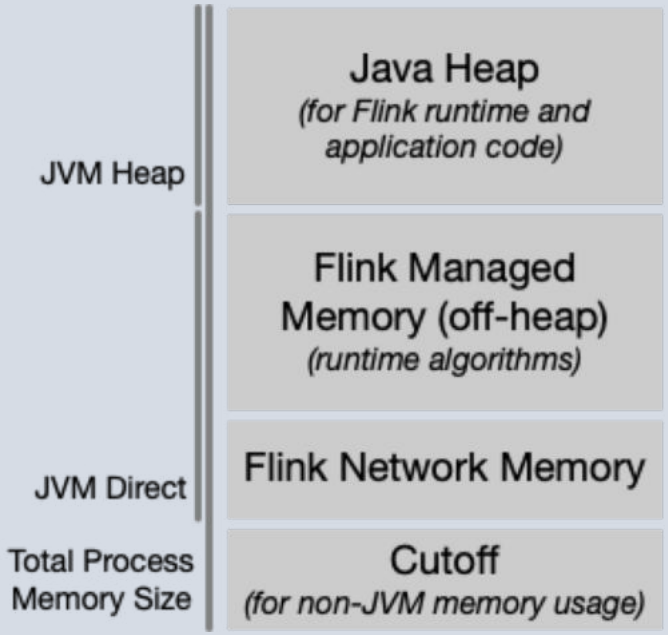


Resource Types

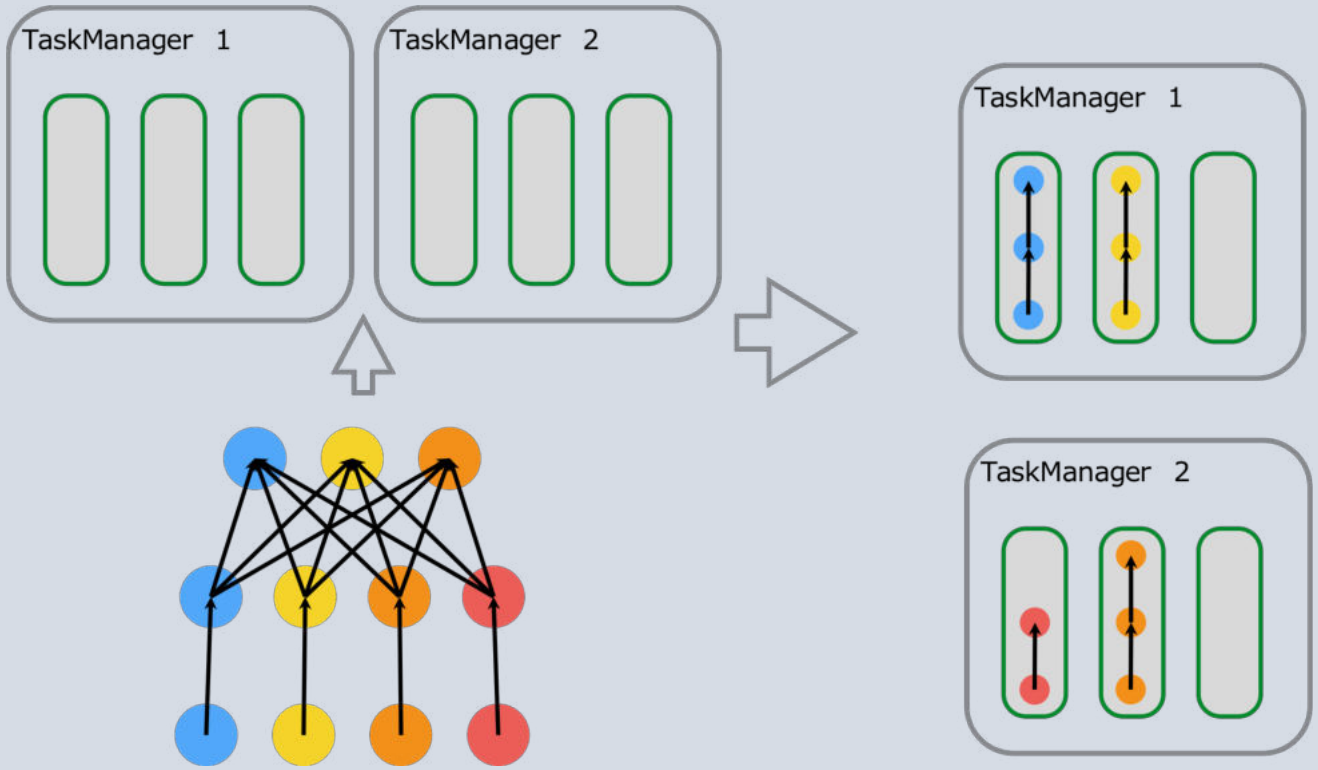
Apache Flink

- Memory
  - Managed Memory
    - 由 TM 的 MemoryManager 负责管理的内存
    - 以 MemorySegment 的形式交给 operator 使用
    - 可配置为使用 on-heap / off-heap (direct) 内存
    - 用途：Batch operator (hash join, hash agg, sort 等)





Slot Sharing

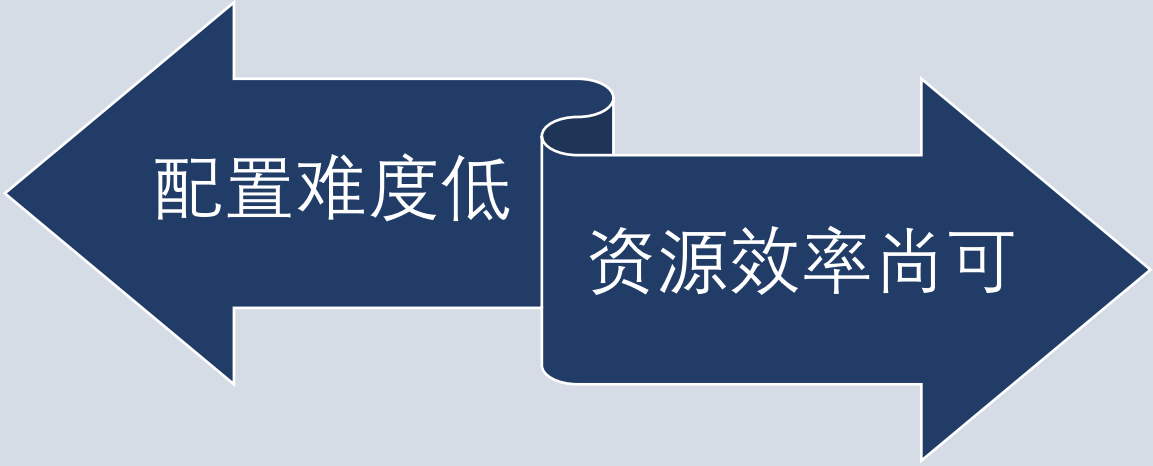


- Slot Sharing
  - 同一 slot sharing group 中的 vertex，其 task 可以共用一个 slot
    - 默认情况下，一个作业的所有 vertex 为一个 slot sharing group
  - 一个 slot 中，对于每个 vertex 最多只能有一个该 vertex 的 task
- 优点
  - 运行一个作业所需的 slot 数量为该作业的最大并发数
  - 相对负载均衡

Flink Philosophy

Apache Flink

1. 通过 Slot Sharing 机制控制 slot 的数量和负载均衡
2. 调整 slot 的资源（即 TM 的资源），以适应一个 Slot Sharing Group 的资源需求



问题 1: 一个 Flink Cluster 中运行多个作业时，如何配置 TM / Slot 资源？

Apache Flink

- 例：
  - Job1 一个 Slot Sharing Group 需要：
    - Java Heap: 500MB
    - On-Heap Managed: 500MB
    - Cutoff: 100MB
  - Job2 一个 Slot Sharing Group 需要：
    - Java Heap: 100MB
    - Off-Heap Managed: 1GB
    - Cutoff: 500MB

不同作业的资源需求差异，可能导致集群资源配置困难，造成资源浪费。

问题 2: 如何精细调整各个维度的资源大小？

Apache Flink

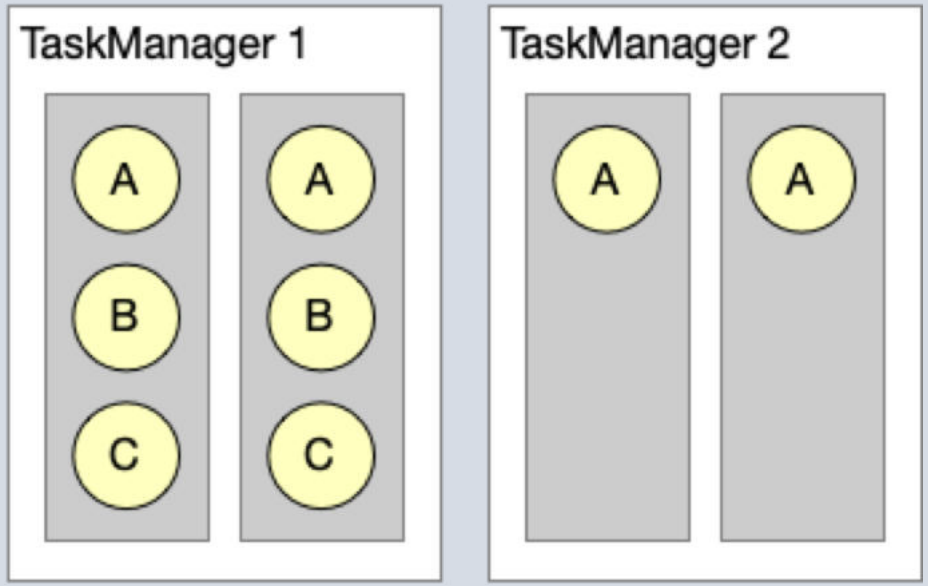
- 资源配置计算
  - JVM Heap (Xmx)
    - Standalone
      - Network Memory = JVM Memory (taskmanager.heap.size) * Network Memory Fraction
      - Managed Memory = (JVM Memory – Network Memory) * Managed Memory Fraction
      - Use on-heap managed memory:
        - JVM Heap = JVM Memory – Network Memory
      - Use off-heap managed memory:
        - JVM Heap = JVM Memory – Network Memory – Managed Memory
    - Containerized
      - JVM Memory = Container Memory * (1 – Cutoff Ratio)
  - TM Started
    - On-Heap Managed Memory = Free Heap Memory * Managed Memory Fraction
- 以上为经过简化的配置计算过程，计算结果与实际情况会有所差异

- 复杂难懂
- 不确定
- 不一致

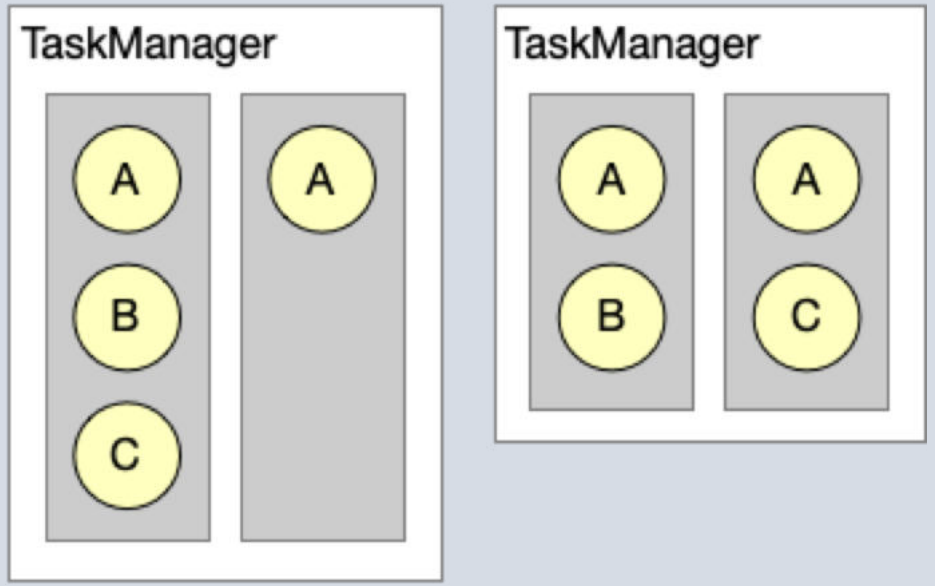
问题 3: Slot Sharing 的资源碎片与不确定性

Apache Flink

资源碎片



不确定性



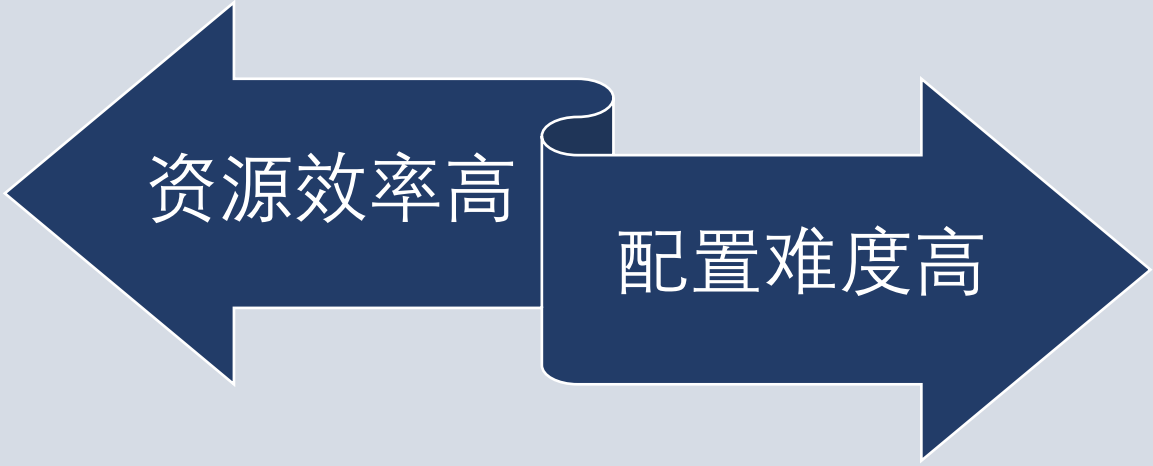
PART 02

Blink 资源管理

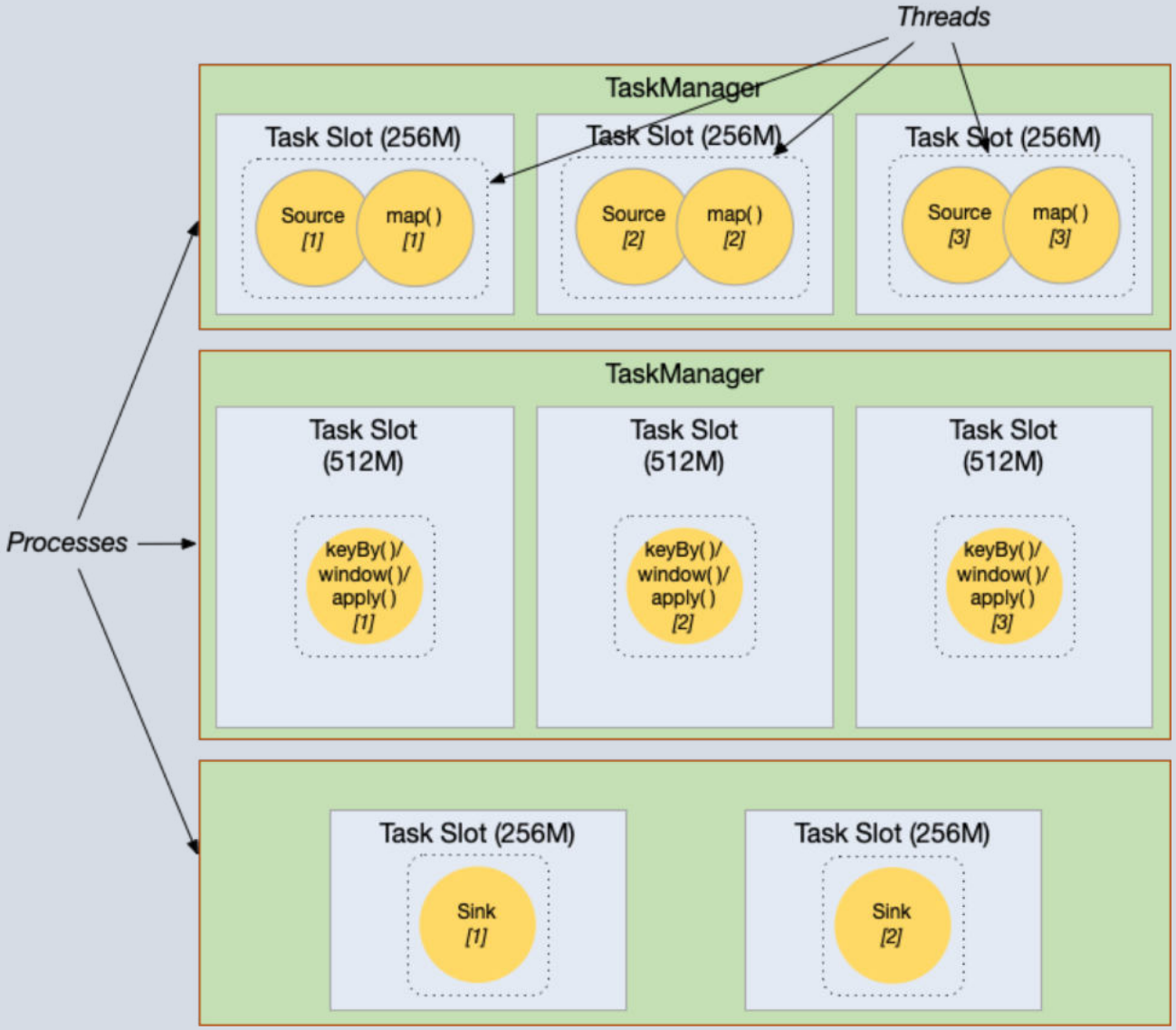
Blink Philosophy

Apache Flink

1. 调整 task 的资源配置，以适应其实际的资源需求
2. 根据各个 task 的资源需求进行调度，优化整体资源利用率



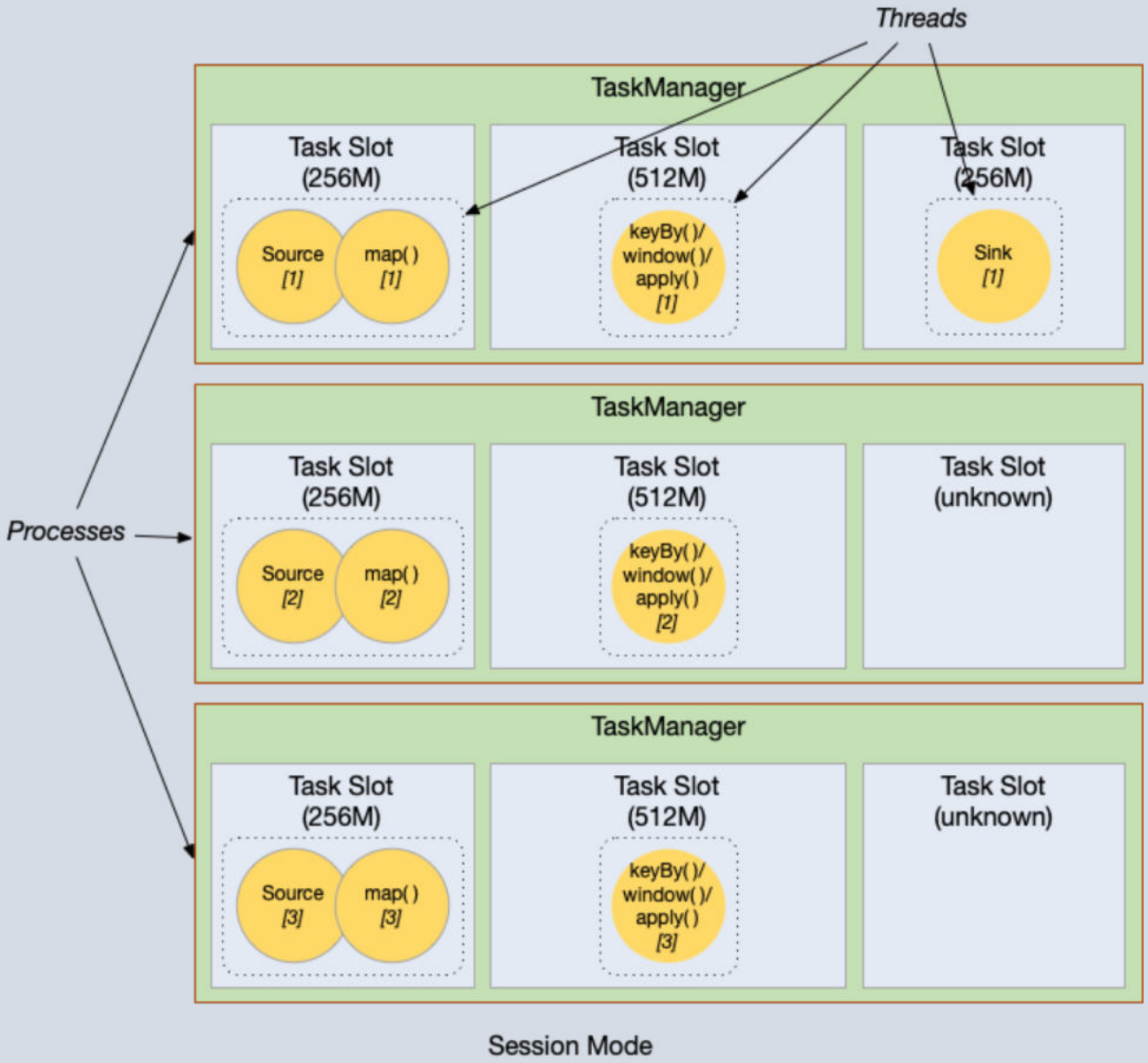
Yarn Perjob Mode



- 一个 Flink Cluster 只运行一个作业
- TaskManager 数量 – 按需申请
- TaskManager 资源 – 根据 Task 的资源需求定制
- 调度策略
  - Task 与 Slot 资源严格匹配
  - 一个 TaskManager 上只有一种规格的 Slot
- 极大程度地避免了资源碎片

Yarn Session Mode

Apache Flink



- 一个 Flink Cluster 可运行多个作业
- TaskManager 数量 – 固定数量
- TaskManager 资源 – 根据配置决定
- 调度策略
  - 根据 Task 的资源需求动态切割 TaskManager 资源给 Slot
  - Task 在 TaskManager 间尽量平铺
- 调度 Task 时无需等待 TM 启动

如何配置 Task 的资源需求

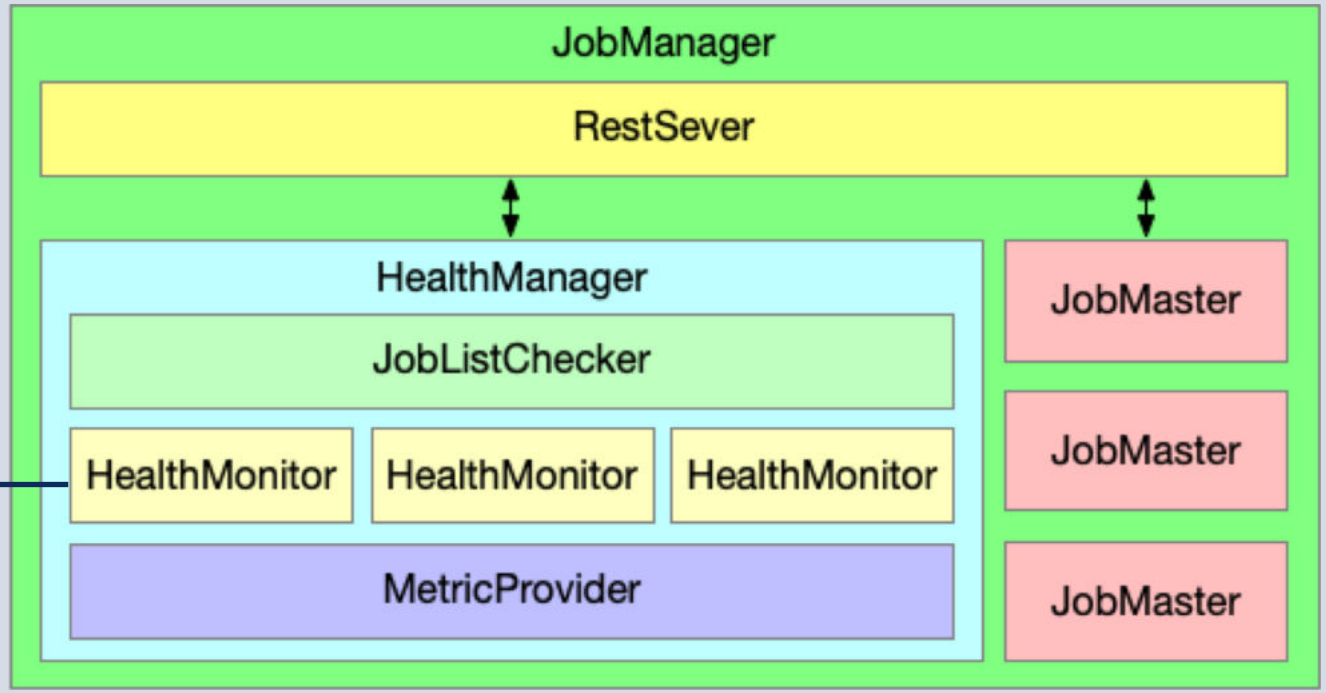
Apache Flink

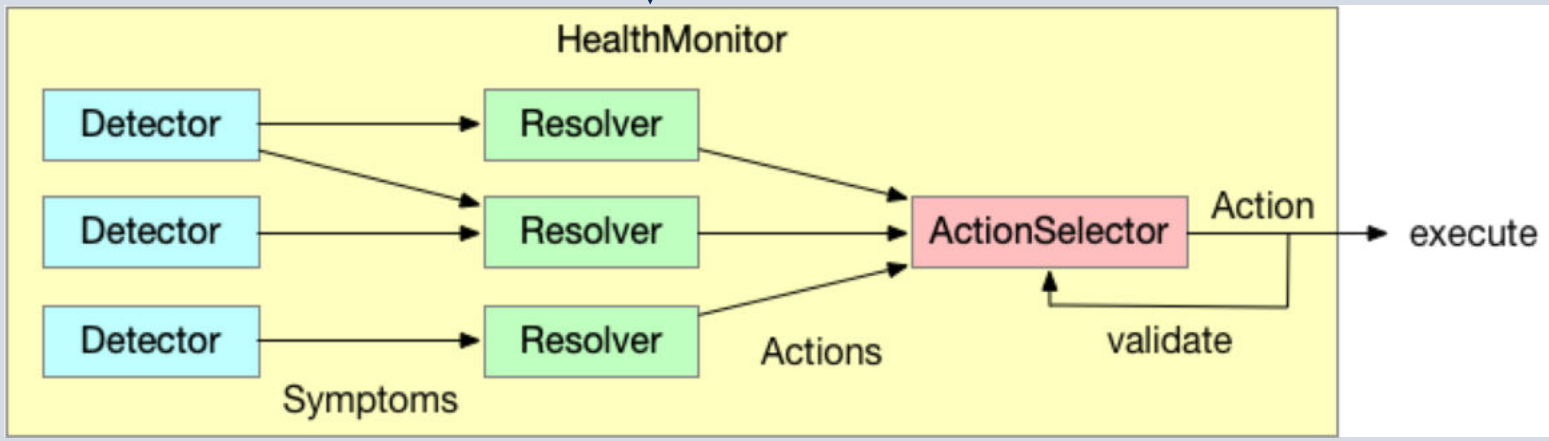
- 基于 JSON 的资源配置文件
- 图形化的资源配置界面
- AutoConfig & AutoScale
  - 基于作业运行期间的 Metrics，对作业配置进行自动调优
  - 调优目标：在满足预设的业务延迟标准的前提下，优化资源开销
  - 调优内容：Task的资源配置及并发数

AutoConfig & AutoScale

Apache Flink

- 应用场景
  - 新作业上线，资源配置
  - 高并发追数据后，释放多余资源
  - 数据流量突增，扩容保障性能





问题 1: 作业配置复杂

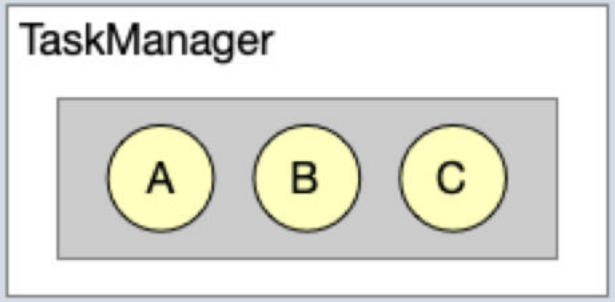
Apache Flink

- 资源管理高度依赖准确的 Task 资源需求配置
- AutoConfig & AutoScale 尚不成熟
  - 针对部分拓扑结构及 operator 类型调优效果不理想
  - 影响作业稳定性
  - 目前仍无法完全代替手动配置
- 手工配置难度大、成本高

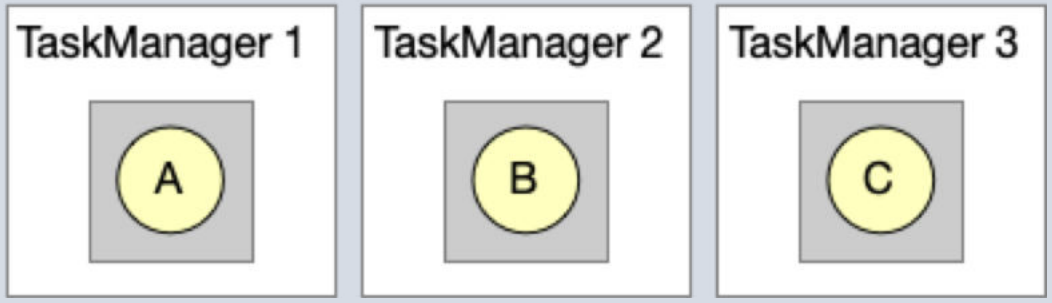
问题 2: 小规模作业框架开销较高

Apache Flink

Flink



Blink



Flink 资源管理 vs. Blink 资源管理

Apache Flink

Flink

- 以 Slot Sharing 为核心
- 作业整体资源决定 Task 资源占用
- 配置简单，易用性强
- 适合基础用户、小规模作业

Blink

- 细粒度资源管理
- Task 资源需求决定作业整体资源
- 配置难度高，资源效率高
- 适合深度用户、大规模作业

两种资源管理策略反映出了不同场景下的需求差异

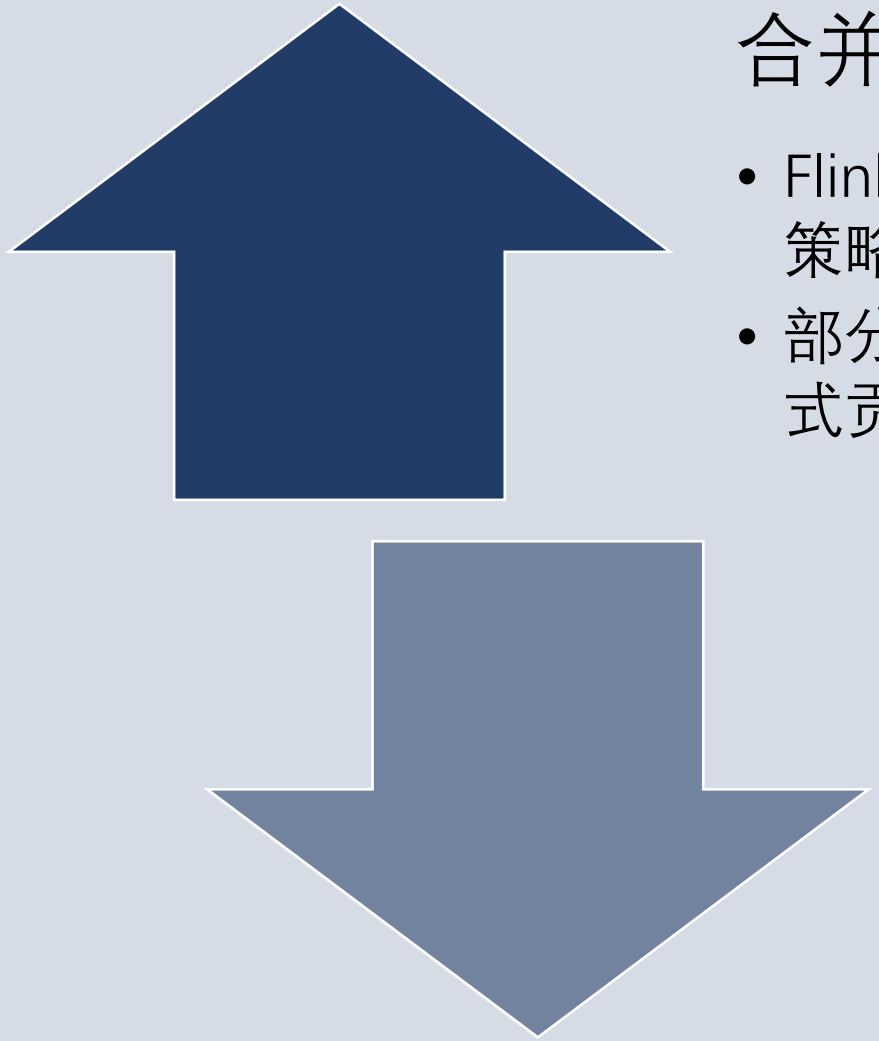
两种策略均有存在的价值

PART 03

近期社区规划

Flink 1.10 周期目标

Apache Flink



合并 Flink / Blink 资源管理机制

- Flink 1.10 将支持 Flink / Blink 两种资源管理策略
- 部分 Blink 调度策略将在后续版本以插件形式贡献 Flink

解决部分 Flink 遗留问题

- 资源配置梳理与简化
- 流、批一体的内存管理

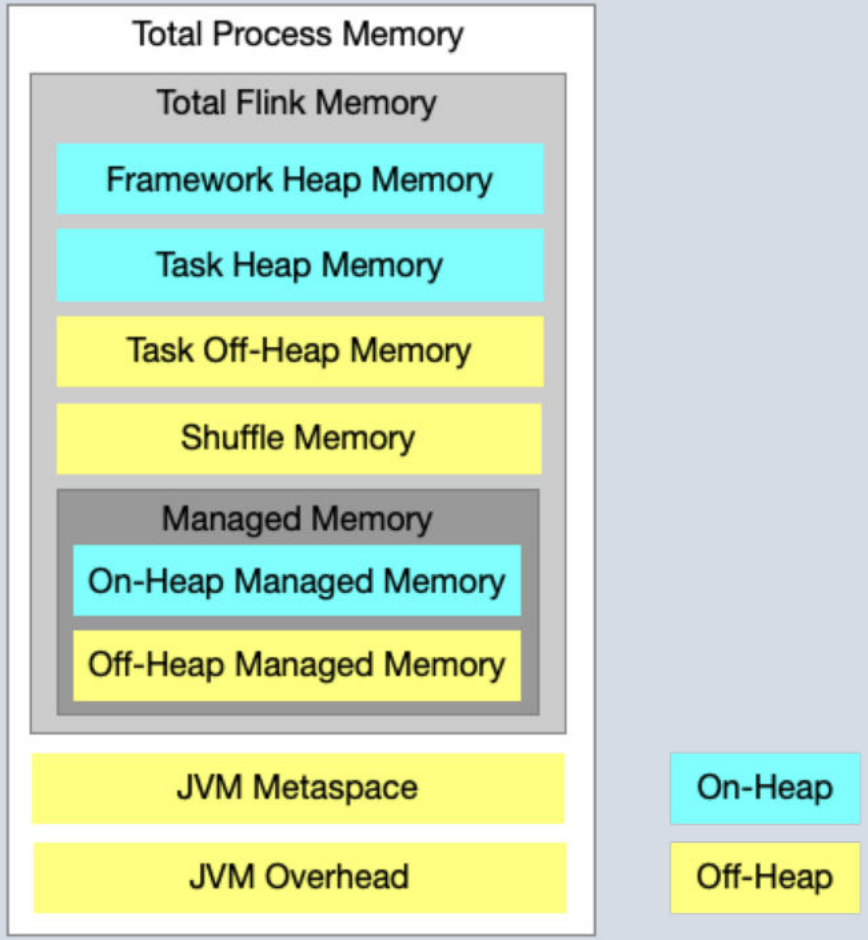
FLIP-49: Unified Memory Configuration for TaskExecutors

Apache Flink

- 简化 TaskManager 资源配置
- 统一 Streaming / Batch 作业的内存管理
- 排除资源计算中的不确定性、不一致性

- 主要变化
  - 整理、细化 TM 中各 Memory Pool 的划分
  - 支持 StateBackend Reserve Managed Memory

- 目前进展：开发中



FLIP-53: Fine Grained Operator Resource Management

Apache Flink

- 支持 Flink / Blink 两种策略的资源管理机制
- 侧重 operator 的资源需求管理与内存分配

- 主要变化
  - 以 Slot Sharing Group 为单位管理可能同时运行在同一 Slot 的 Task
  - 基于 Fraction 的 Operator 内存配额管理
  - 编译阶段，根据 Operator 的资源需求（可能是 unknown）计算基于相对内存配额

- 目前进展：方案已采纳，待开发

FLIP-56: Dynamic Slot Allocation

Apache Flink

- 支持 Flink / Blink 两种策略的资源管理机制
- 侧重 Slot 的调度与分配
- 主要变化
  - 切割 TaskManager 资源，动态创建、回收 Slot
  - 对于未知资源需求的 Task，分配默认资源的 Slot，与之前版本保持行为一致
- 目前进展：方案投票中

- 调度策略插件化：已规划，待开发

FLINK FORWARD

Organized by 阿里云

Created by ververica

Flink Forward Asia

全球最大的 Apache Flink 官方会议

预计 **2000+** 参会人员， 2019年11月28-30日 @北京国家会议中心

国内外一线厂商悉数参与

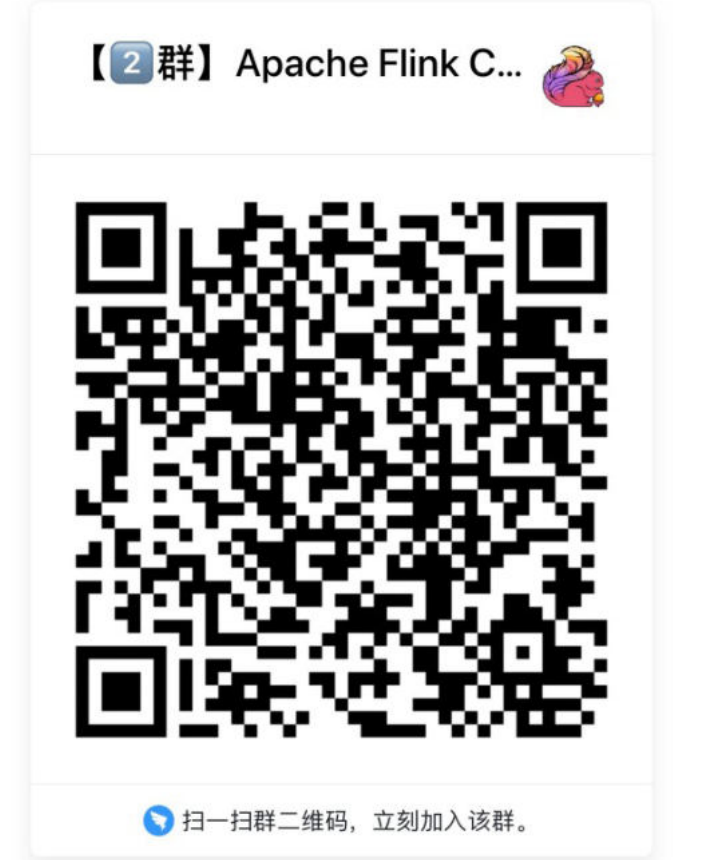
阿里巴巴、腾讯、字节跳动、intel、 DellEMC 、Uber、美团点评、Ververica ...

大会官网，查看更多

Apache Flink

Apache Flink 社区微信公众号「 Ververica」

Meetup动态 / Release 发布信息 / Flink 应用实践



THANKS

Apache Flink China Meetup

· BEIJING