Flink资源管理机制

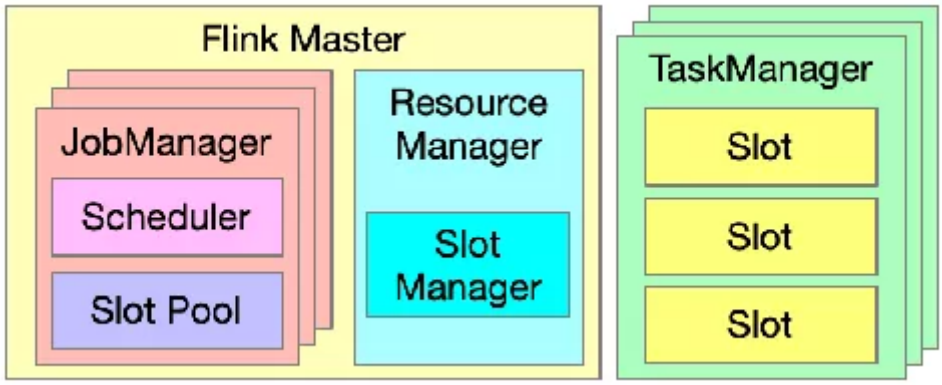
1 基本概念

1.1 相关组件

Flink 资源管理相关的组件：

- 一个Flink Cluster 是由一个 Flink Master 和多个 Task Manager 组成的
- Master 和 TaskManager 是进程级组件

其他的组件都是进程内的组件



如图：

- 一个 Flink Master 中有一个 Resource Manager 和多个 Job Manager
- Flink Master 中每个 JobManager 都**单独管理**一个具体的 Job
- Job Manager 中的 Scheduler 组件负责调度执行该 Job 的 DAG 中所有 Task 发出资源请求，即**整个资源调度的起点**
- Job Manager 中的 Slot Pool 组件持有分配到**该 Job** 的所有资源
- 另外，Flink Master 中唯一的 Resource Manager 负责整个 Flink Cluster 的资源调度以及与外部调度系统**对接**（外部系统：Kubernetes、Mesos、Yarn等资源管理系统）

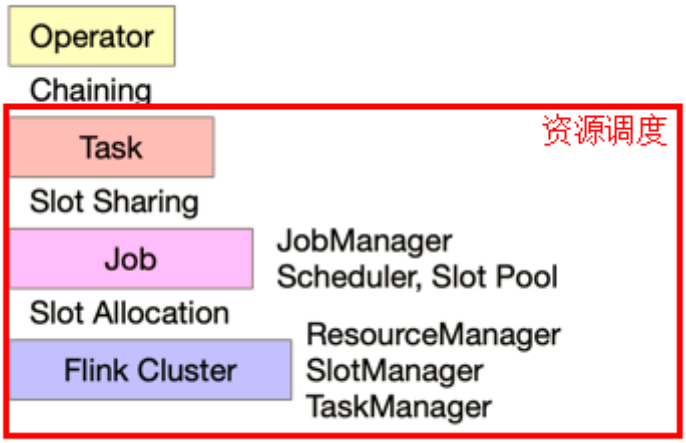
TaskManager 负责 Task 的执行，其中 Slot 是 TaskManager 资源的一个子集，也是 Flink 资源管理的一个基本单位。

Slot 的概念贯穿整个资源调度的过程。

1.2 逻辑层级

组件之间的逻辑关系，共分为4层

- Operator
  - 算子是最基本的数据处理单元
  - 如果两个Operator属于同一个Task，那么不会出现一个Operator已经开始运行另一个Operator还没被调度的情况
- Task
  - Flink Runtime 中真正去进行调度的最小单位
  - 由一系列算子链组成（chained operators）
- Job
  - 对应一个 Job Graph
- Cluster
  - 1 Flink Master + N Task Manager



资源调度的范畴，实际上是图中红框中的内容

- JobManager (Secheduler 和 Slot Pool) 对应于 Job
- Resource Manager (Slot Manager) 和 Task Manager 对应于 Flink Cluster 级别

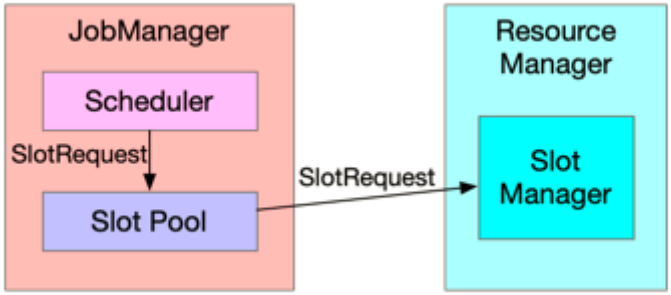
- 在 Operator 和 Task 中间的 Chaining 是指如何用 Operator 组成 Task。
- 在 Task 和 Job 之间的 Slot Sharing 是指多个 Task 如何共享一个 Slot 资源，这种情况不会发生在跨作业的情况中
- Flink Cluster 和 Job 之间的 Slot Allocation 是指 Flink Cluster 中的 Slot 是怎样分配给不同的 Job

1.3 两层资源调度模型

Flink 资源调度是一个经典的两层模型，

- 其中从 Cluster 到 Job 的分配过程是由 Slot Manager 来完成
- Job 内部分配给Task资源的过程是由 Scheduler 来完成

如图: Scheduler 向 Slot pool 发出 Slot Request（资源请求），Slot Pool 如果不能满足该资源需求则会进一步请求 Resource Manager，具体来满足该请求的组件是 Slot Manager

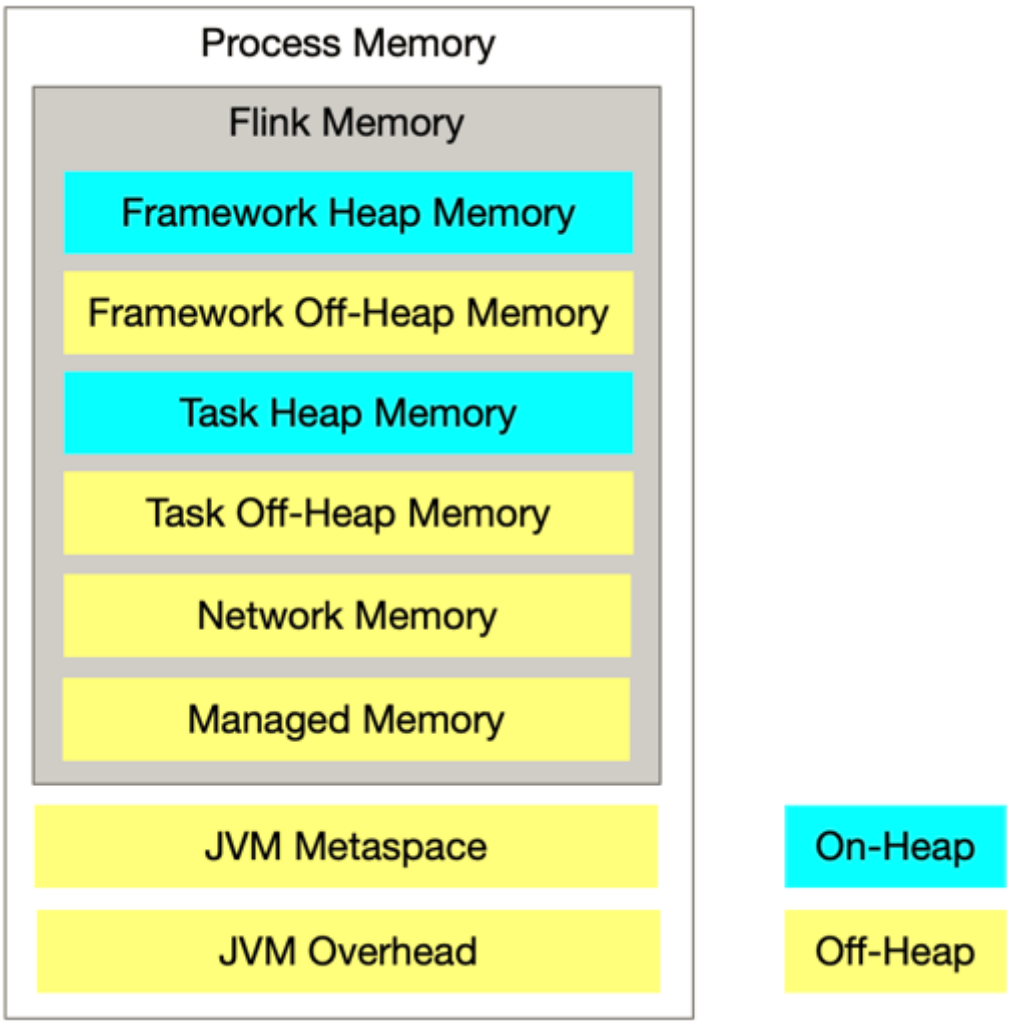


Task 对 Slot 进行复用有两种方式：

- Slot Caching
  - 批处理
  - 流处理的 Failover
  - 多个 Task 先后/轮流使用 Slot 资源
- Slot Sharing
  - 多个 Task 在满足一定条件下同时共享同一个 Slot 资源

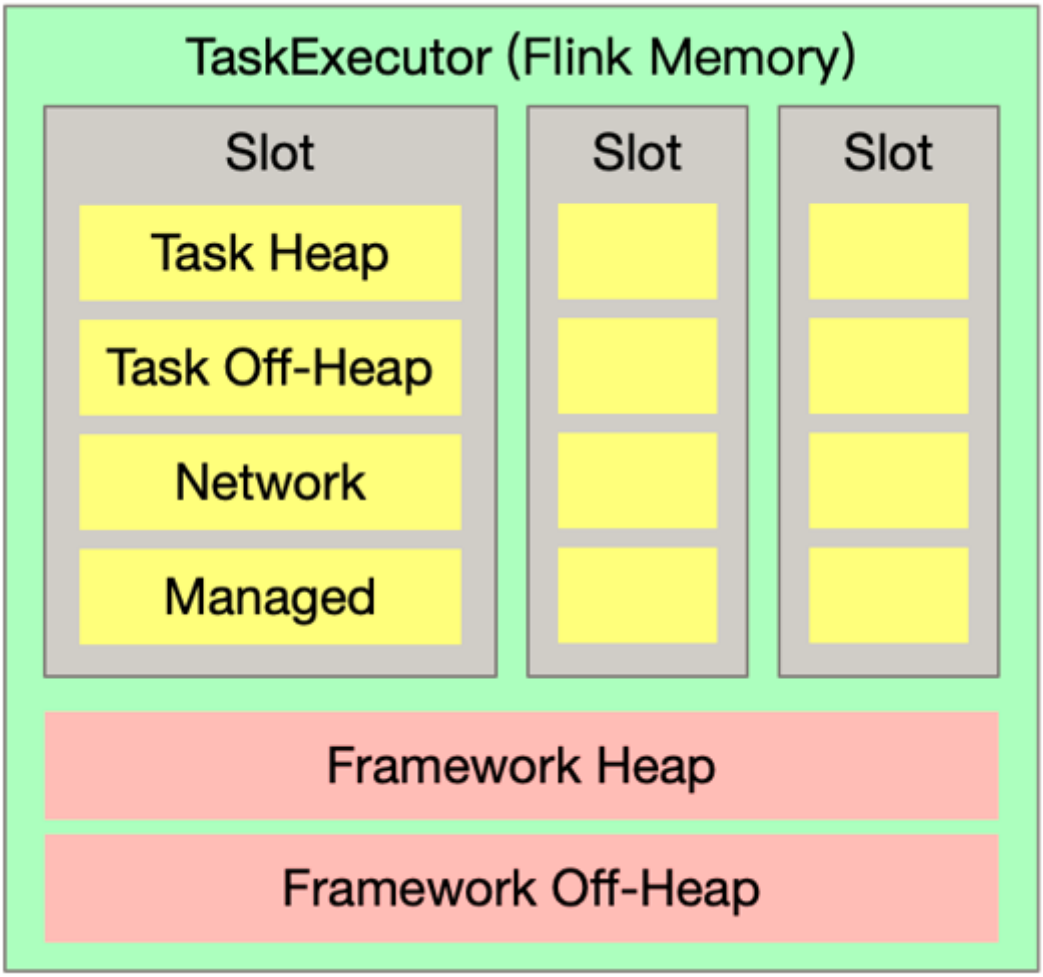
2 当前机制与策略

2.1 TaskManager有哪些资源



- 资源类型
    - GPU（FLIP-108，在Flink1.11版本完成）
    - 内存
    - CPU
    - 其他扩展资源
- Task Manager 资源由配置决定
    - Standalone 部署模式下，TM 资源可能不同
    - 其他部署模式下，所有 TM 资源均相同

2.2 Slot 有哪些资源



Task Manager 中有固定数量的 Slot，Slot 的具体数量由配置决定，同一 TaskManager 上 Slot 之间没有差别，每一个 Slot 都一样大，即资源一样多

2.3 Flink Cluster 有多少 Task Manager

- Standalone 部署模式

在Standalone部署模式下，Task Manager的数量是固定的，如果 start-Cluster.sh 脚本来启动集群，可以通过修改以下文件中的配置来决定TM的数量，也可以通过手动执行 Taskmanager.sh 脚本来启动一个TM。`<FLINK_DIR>/conf/slaves`

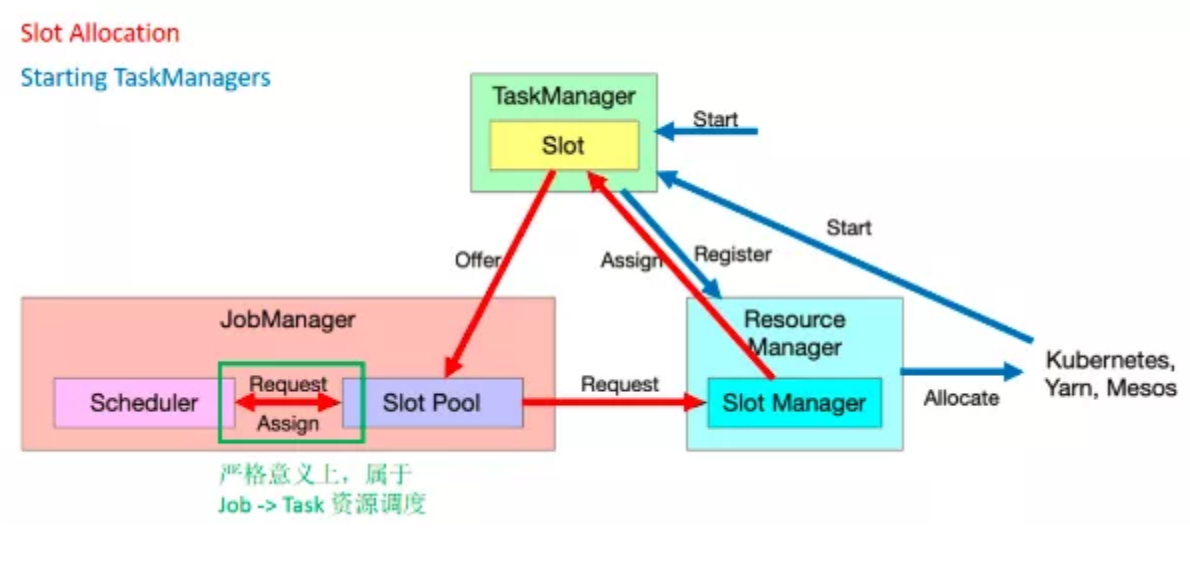
- Active Resource manager 部署模式

- 当前 Slot 数量不能满足新的 Slot Request 时，申请并开启新的 Task manager
- TaskManager 空闲一段时间后，超时则释放
- Kubernetes，Yarn，Mesos
- 由 SlotManager / ResourceManager 按需动态决定

On-Yarn 部署模式不再支持指定固定数量的 TM，即以下命令参数已经失效

```
yarn-session.sh -n <num>
flink run -yn <num>
```

2.4 Cluster -> Job资源调度的过程



Cluster 到 Job 的资源调度过程中主要包括两个过程。

- Slot Allocation（红箭头）

Scheduler 向 Slot Pool 发送请求，如果 Slot 资源足够则直接分配，如果 Slot 资源不够，则由 Slot pool 再向 Slot Manager 发送请求（此时即为 Job 向 Cluster 请求资源），如果 Slot Manager 判断集群当中有足够的资源可以满足需求，那么就会向 Task Manager 发送 Assign 指令，Task Manager 就会提供 Slot 给 Slot Pool，Slot Pool 再去满足 Scheduler 的资源请求

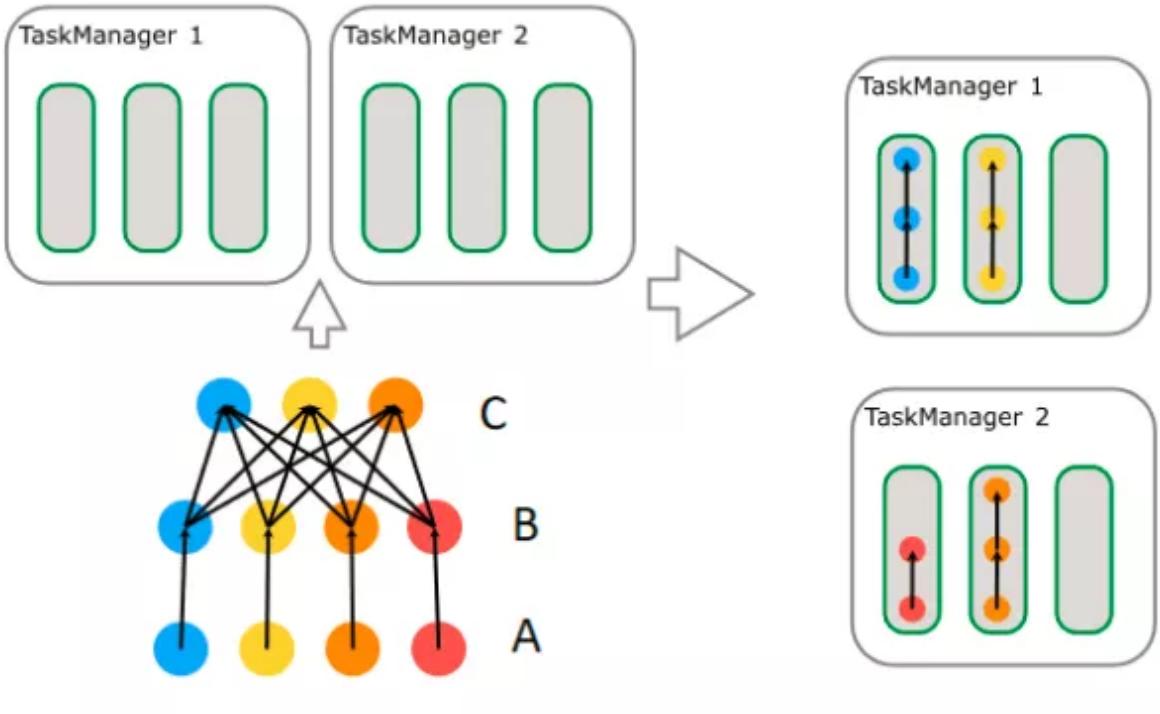
- Starting TaskManagers（蓝箭头）

在 Active Resource Manager 资源部署模式下，当 Resource Manager 判定 Flink Cluster 中没有足够的资源去满足需求时，它会进一步去底层的资源调度系统请求资源，由调度系统把新的 Task Manager 启动起来，并且 TaskManager 向 Resource Manager 注册，则完成了新 Slot 的补充

2.5 Job -> Task 资源调度的过程

- Scheduler
  - 根据 Execution Graph 和 Task 的执行状态，决定接下来要调度的 Task
  - 发起 SlotRequest
  - 决定 Task / Slot 之间的分配
- Slot Sharing
  - 运行一个作业所需的 Slot 数量为最大并发数
  - 相对负载均衡
  - 默认所有节点在一个 Slot Sharing Group 中
  - 一个 Slot 中相同任务只能有一个

- Slot Sharing Group 中的任务可共用 Slot



Slot Sharing 过程如上图 👆 所示（每一行分别是一个 Task 的多个并发，自下而上分别是 A、B、C），A、B、C 的并行度分别是4、4、3，这些 Task 属于同一个 Slot Sharing Group 中，所以不同的 Task 可以放在相同的 Slot 中运行，如上图右侧所示，有3个 Slot 放入了 ABC，而第四个 Slot 放入了 AB 。通过以上过程我们可以很容易推算出这个 Job 需要的 Slot 数是4，也是**最大并发数**。

2.6 资源调优

通过以上介绍的机制，我们容易发现，Flink 所采用的是自顶向下的资源管理，我们所配置的是 Job 整体的资源，而 Flink 通过 Slot Sharing 机制控制 Slot 的数量和负载均衡，通过调整 Task Manager / Slot 的资源，以适应一个 Slot Sharing Group 的资源需求。Flink 的资源管理配置简单，易用性强，适合拓扑结构简单或规模较小的作业。

> https://mp.weixin.qq.com/s/VEOtfvcX3itxMI3w9JuGmw