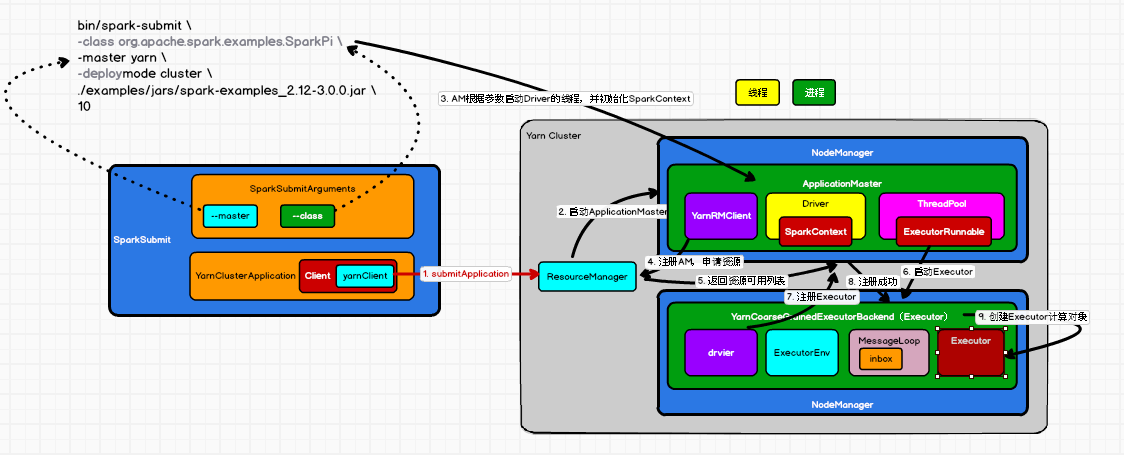
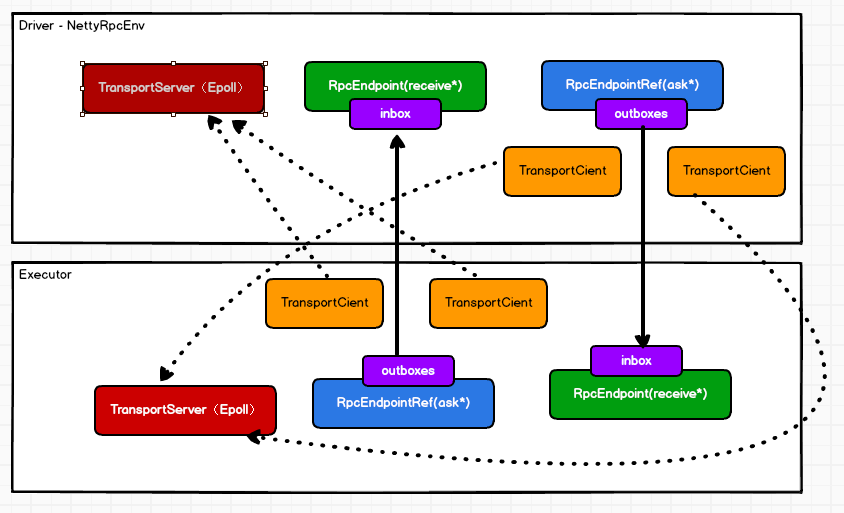
1. 环境准备（Yarn 集群）

Driver，Executor

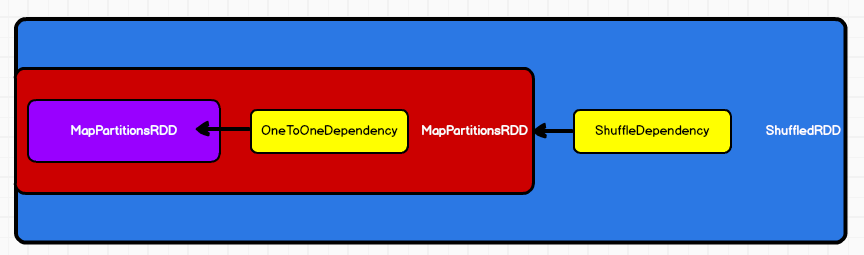


1. 组件通信（Socket）

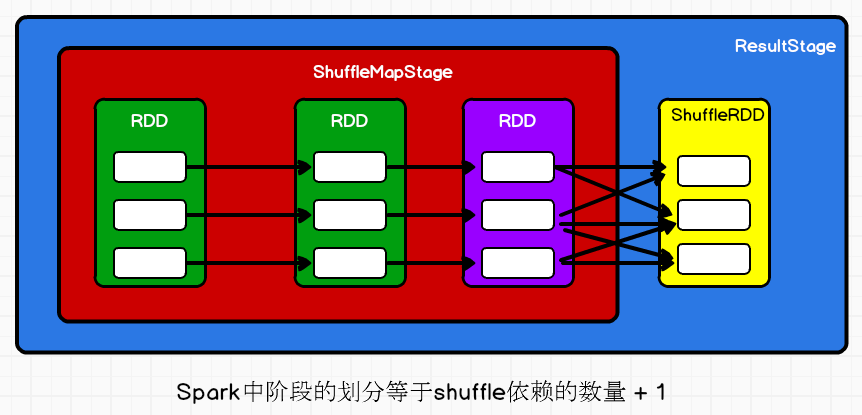
Driver => Executor



1. 应用程序的执行
   1. RDD 依赖



* 1. 阶段的划分



Job 是以 Action 方法为界，遇到一个 Action 方法则触发一个 Job

Stage 是 Job 的子集，以 RDD 宽依赖(即 Shuffle)为界，遇到 Shuffle 做一次划分。

Stage有两个具体子类：

* ShuffleMapStage

1. 是其他 Stage 的输入

2. ShuffleMapStage 内部的转换操作（map、filter等）会组成pipeline，连在一起计算

3. 产生 map 输出文件（Shuffle 过程中输出的文件）

-- Shuffle 内部操作都是窄依赖

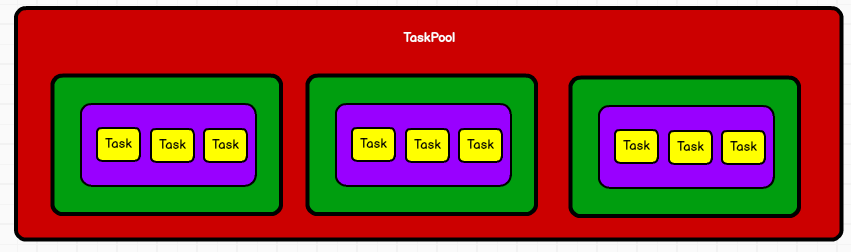
* ResultStage

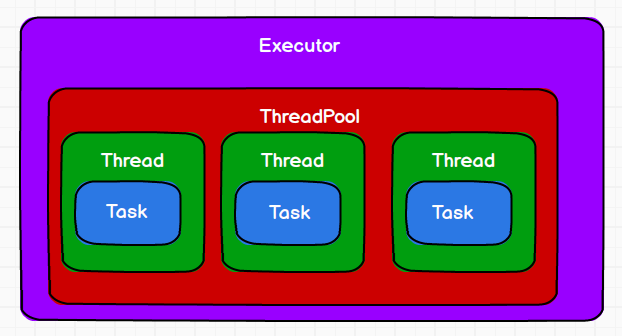
一个job中只有一个ResultStage，最后一个 Stage 即为 ResultStage

Task 是 Stage 的子集，以并行度(分区数)来衡量，分区数是多少，则有多少个 task

Job----> 一个或多个stage---> 一个或多个task(2种类型：1）result task类型，最后一个task，2是shuffleMapTask类型，除了最后一个task都是)

<https://zhuanlan.zhihu.com/p/470482936>

* 1. 任务的切分
  2. 任务的调度
  3. 任务的执行



1. Shuffle
   1. Shuffle 的原理和执行过程
      * shuffleWriterProcessor（写处理器）
      * ShuffleManager: Hash(早期) & Sort（当前）

|  |  |  |
| --- | --- | --- |
| 处理器 | 写对象 | 判断条件 |
| SerializedShuffleHandle | UnsafeShuffleWriter | 1. 序列化规则支持重定位操作（java 序列化不支持，KRYO 支持） 2. 不能使用预聚合 3. 如果下游的分区数量小于或等于   16777216 |
| BypassMergeSortShuffleHandle | BypassMergeSortShuffleWriter | 1. 不能使用预聚合 2. 如果下游的分区数量小于等于 200   （可配） |
| BaseShuffleHandle | **SortShuffleWriter** | 其他情况 |

* 1. Shuffle 写磁盘
  2. Shuffle 读取磁盘

1. 内存的管理
   1. 内存的分类
   2. 内存的配置