提纲

1. RDD设计背景
2. RDD概念
3. RDD特性
4. RDD之间的依赖关系
5. 阶段的划分
6. RDD的运行过程

2.3.5 RDD运行原理（阶段的划分和RDD运行过程）

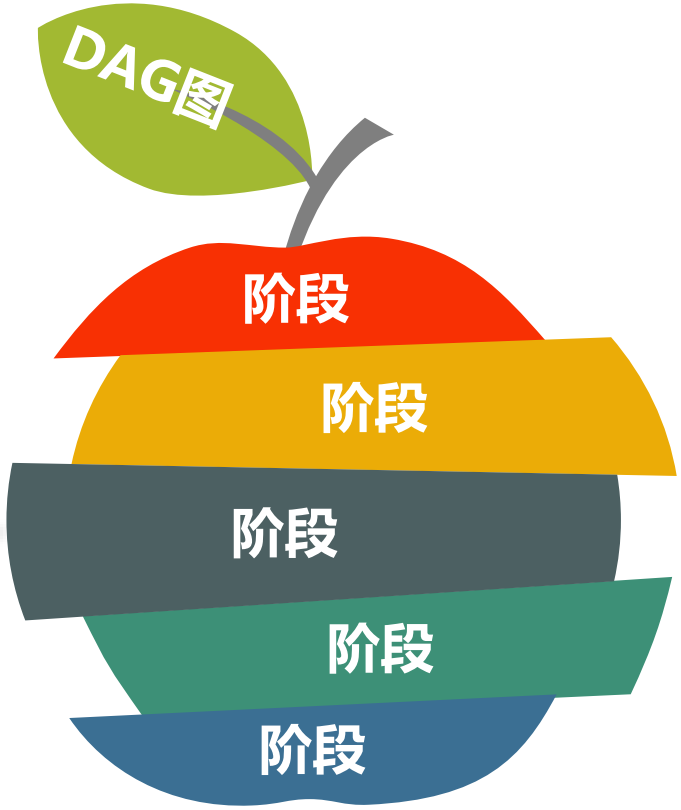


划分阶段的依据?



2.3.5 RDD运行原理（阶段的划分和RDD运行过程）





2.3.5 RDD运行原理（阶段的划分和RDD运行过程）

优化原理fork/join机制



为什么?



可以进行流水线优化

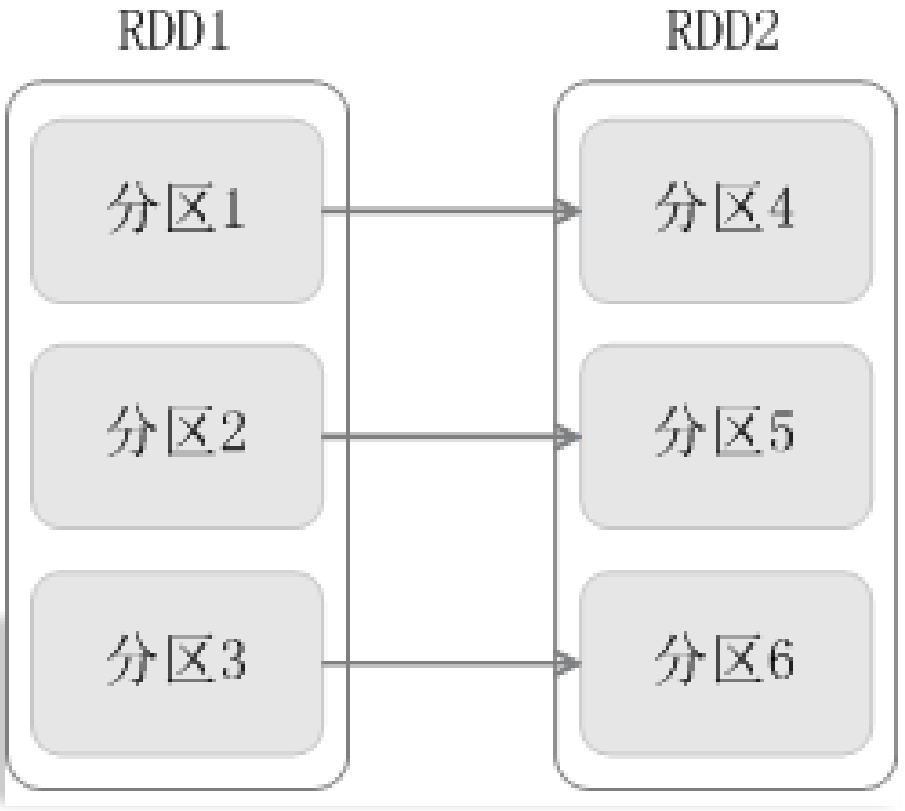
不能进行流水线优化

2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



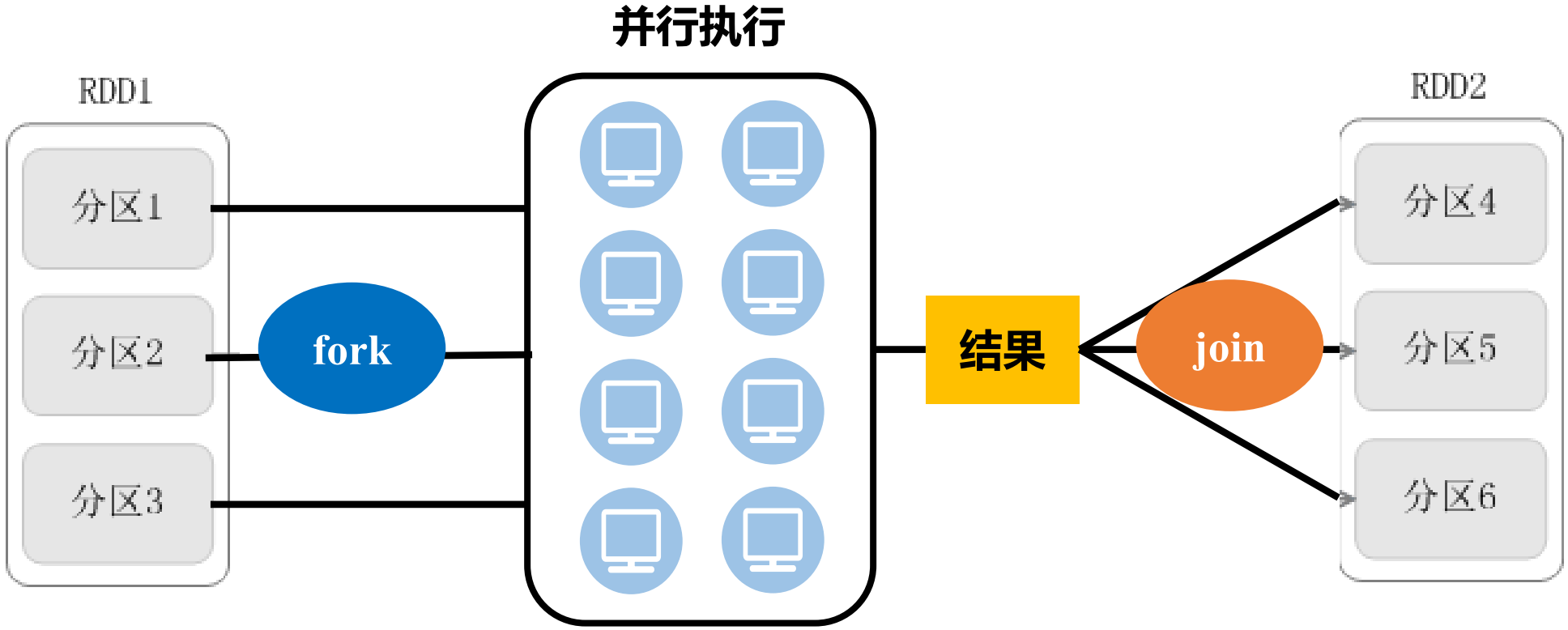
fork/join 机制

并行执行任务的框架



2.3.5 RDD运行原理（阶段的划分和RDD运行过程）

fork/join 机制



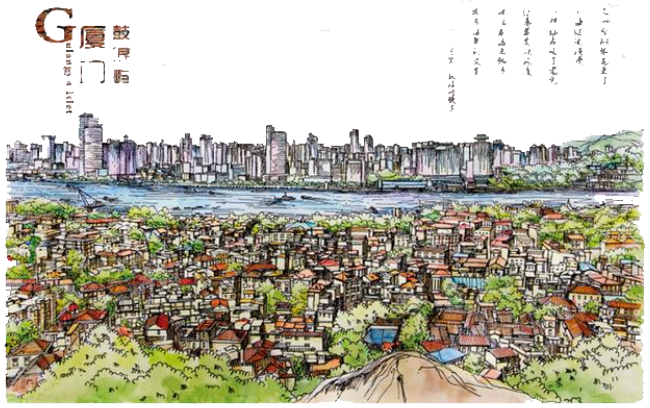
2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



北京

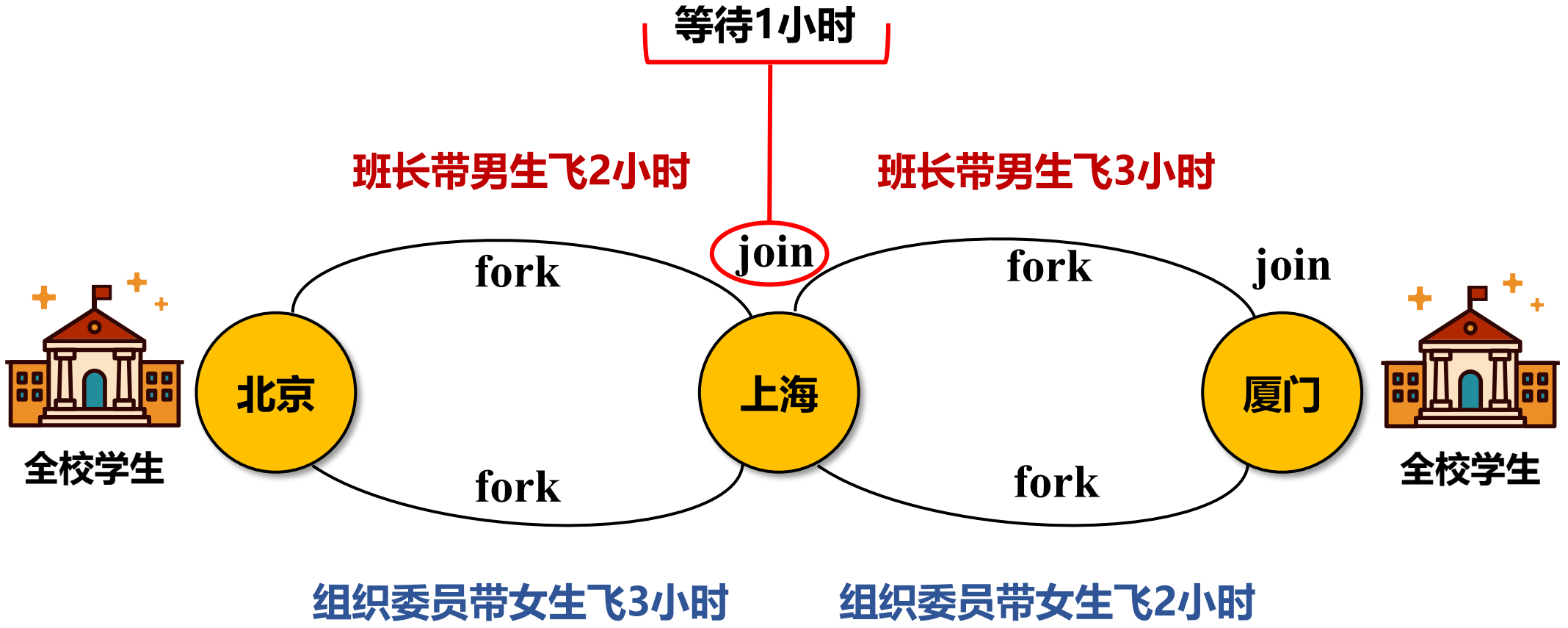


全校学生



厦门

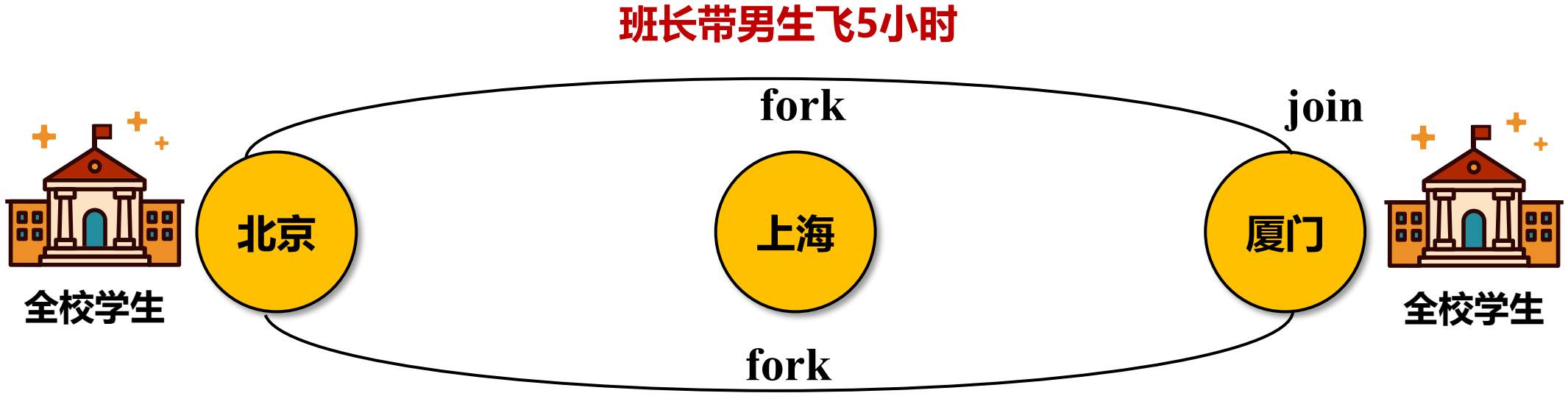
2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



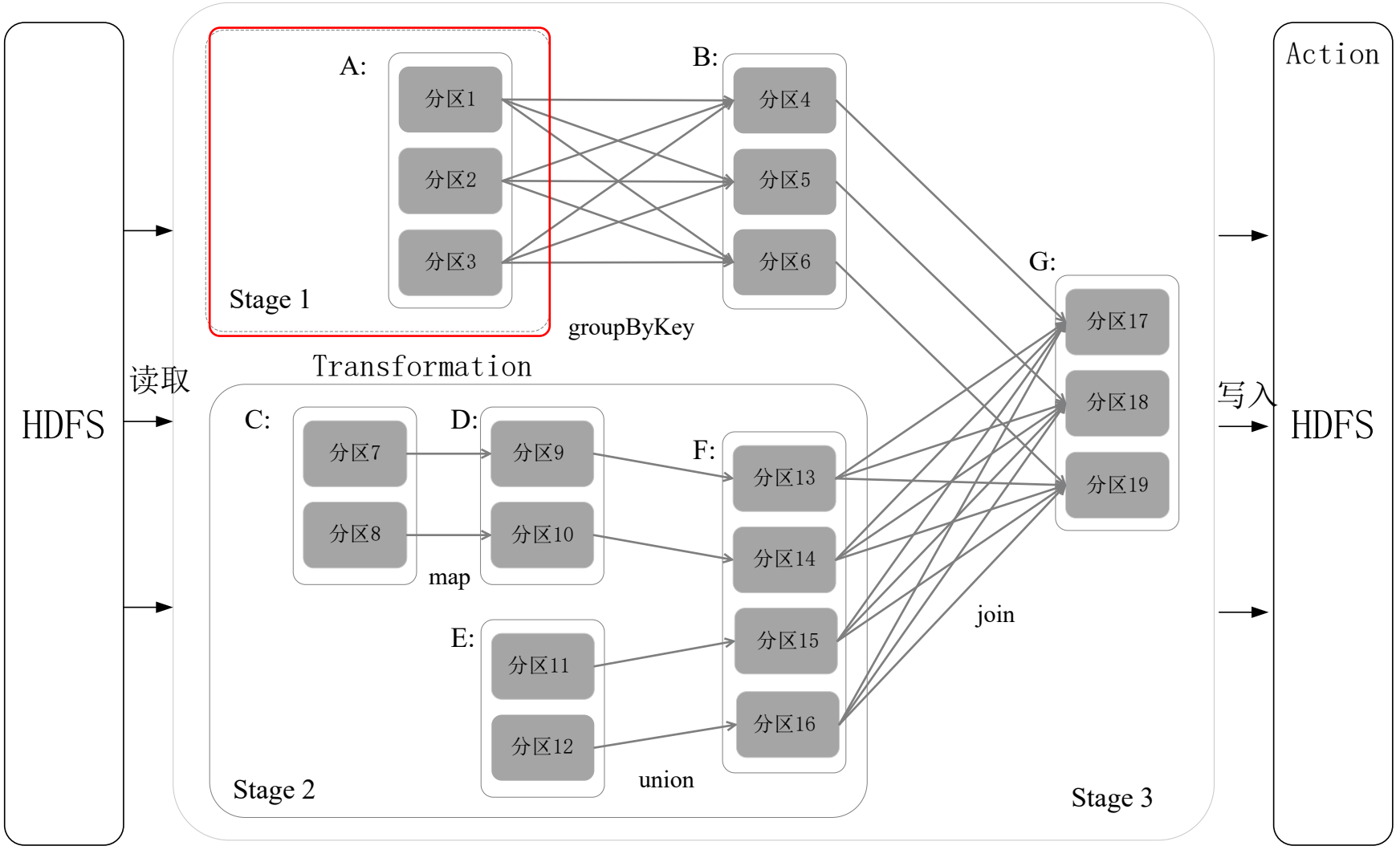
2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



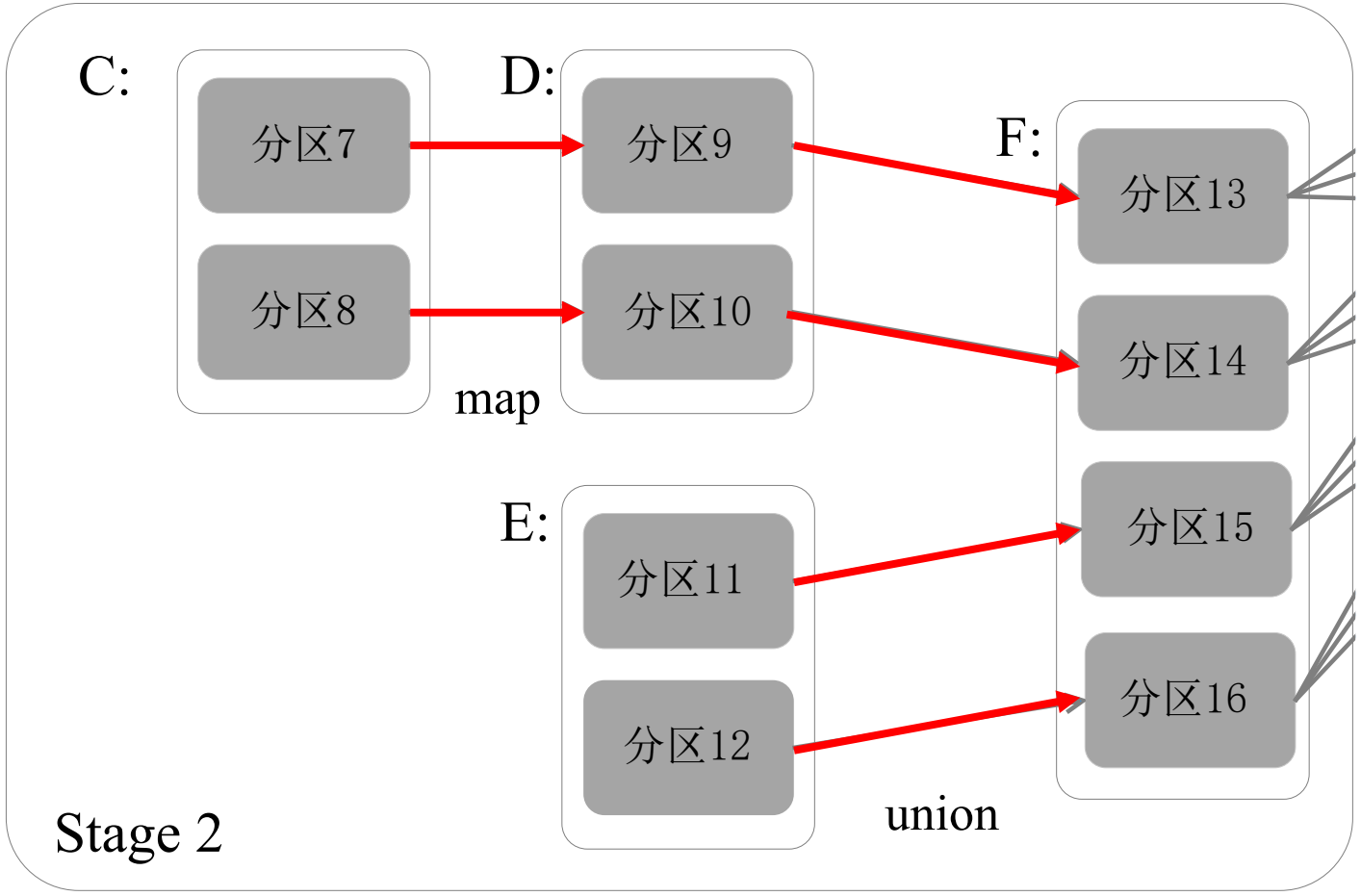
不发生无意义的等待



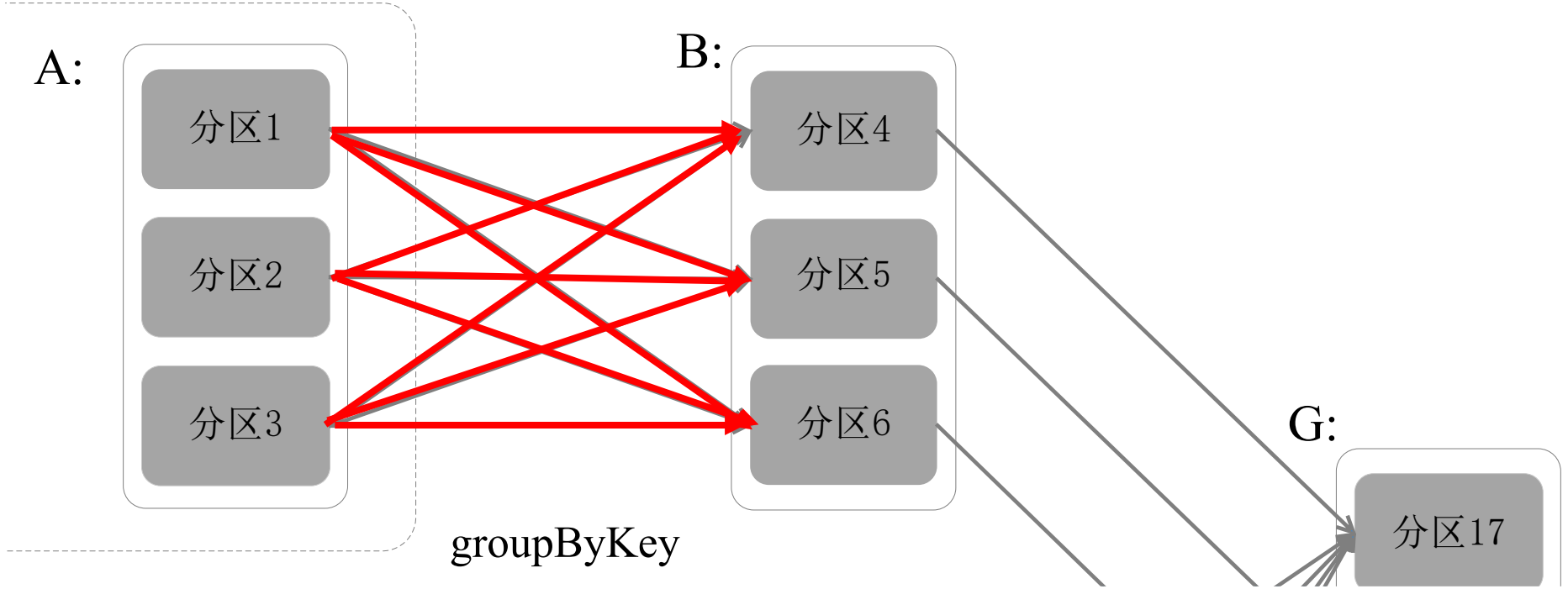
2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



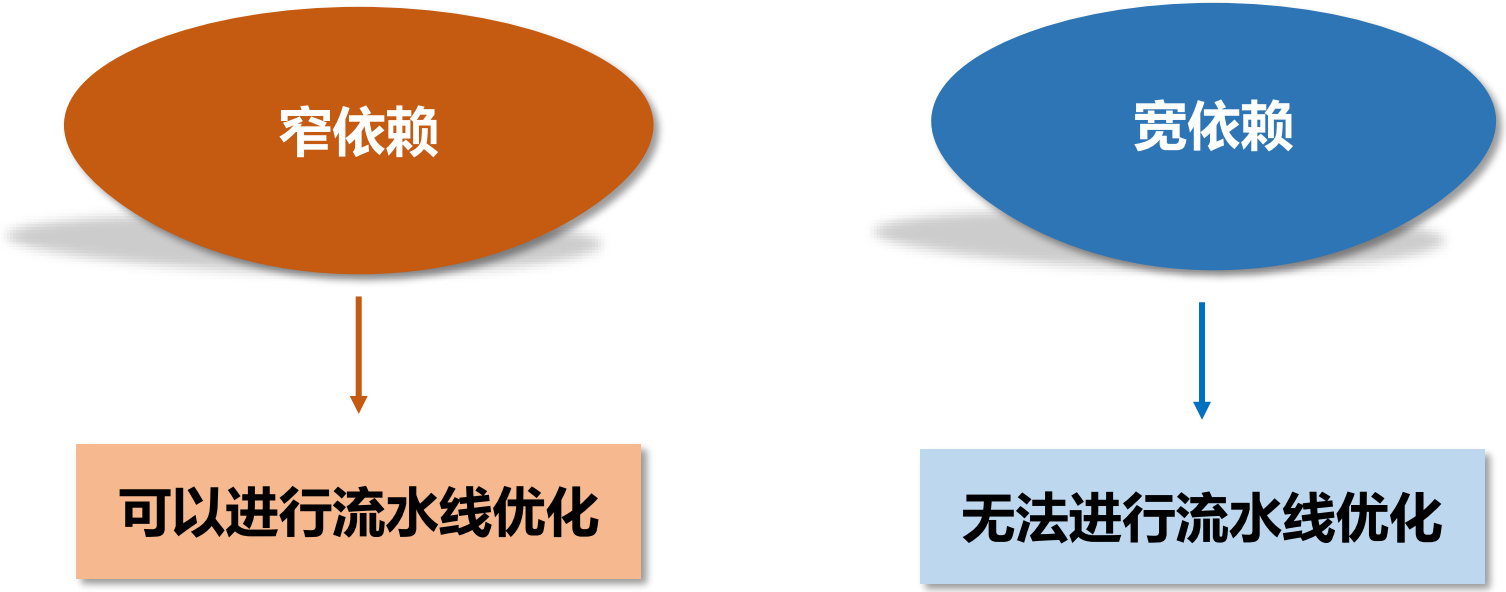
2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



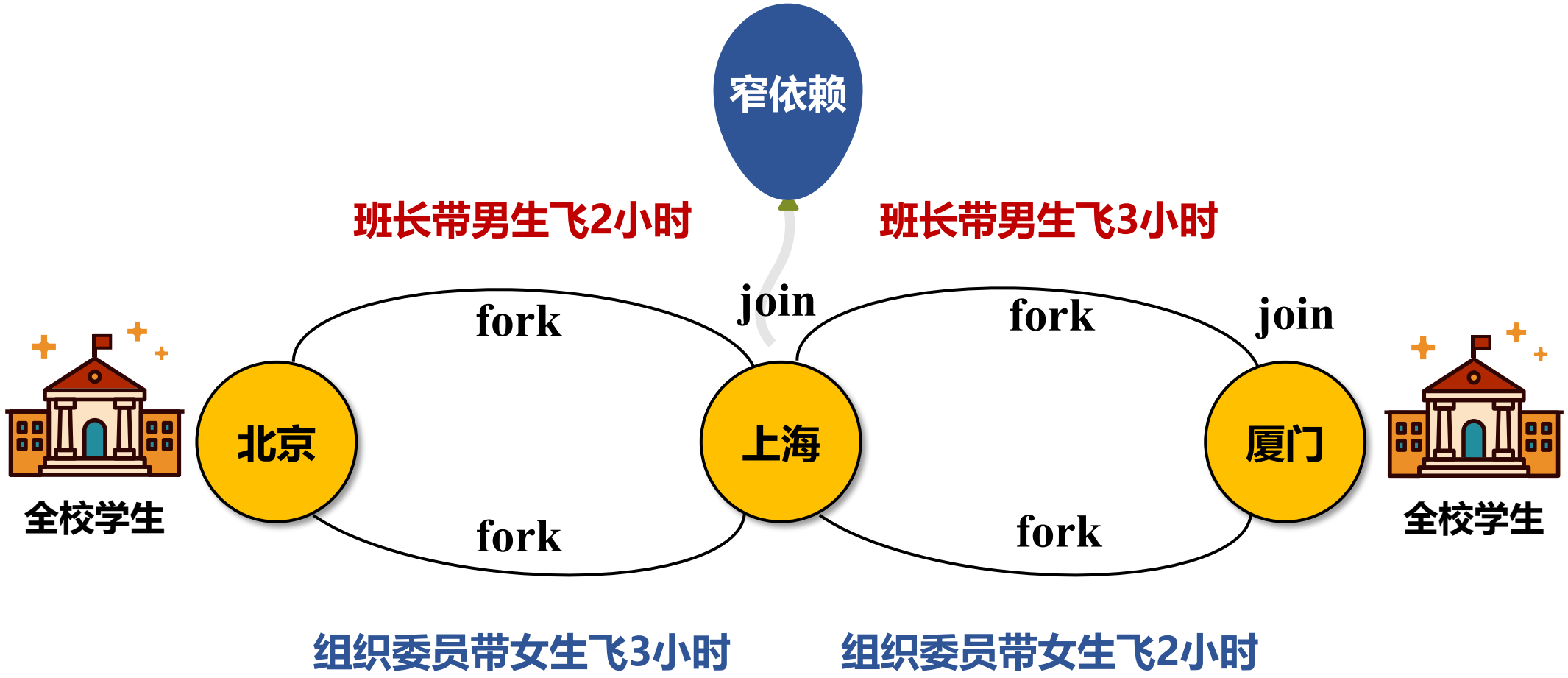
2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



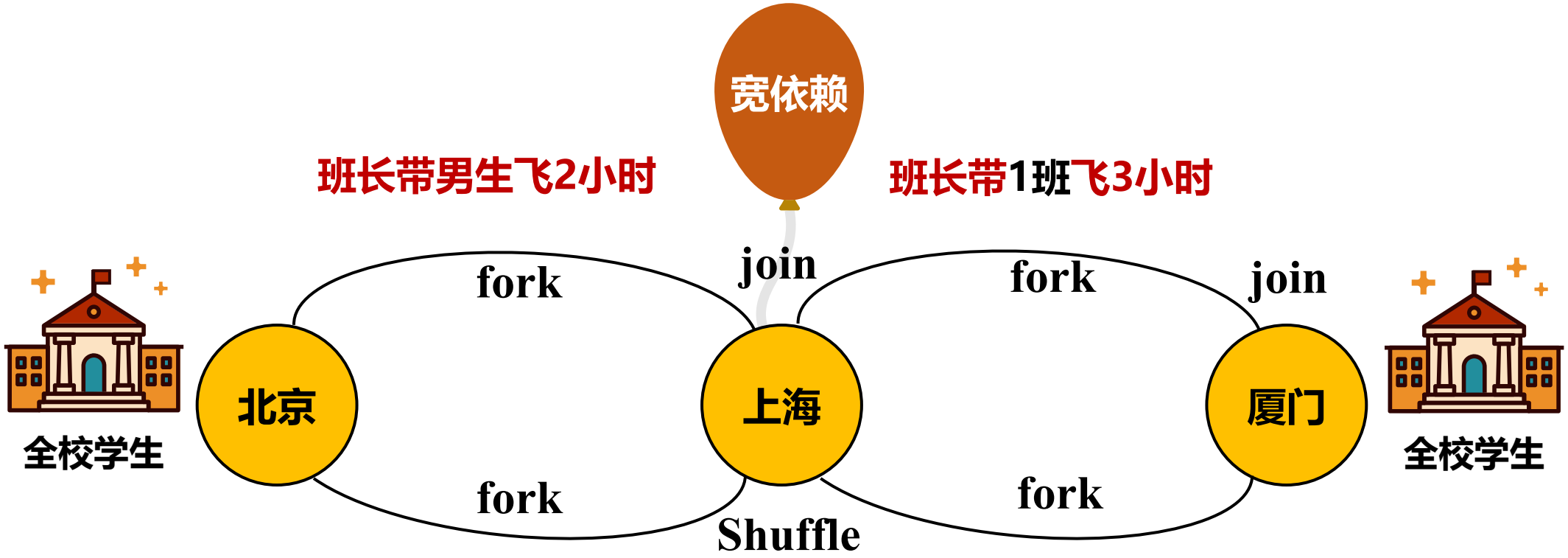
只要发生Shuffle一定会写磁盘



2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



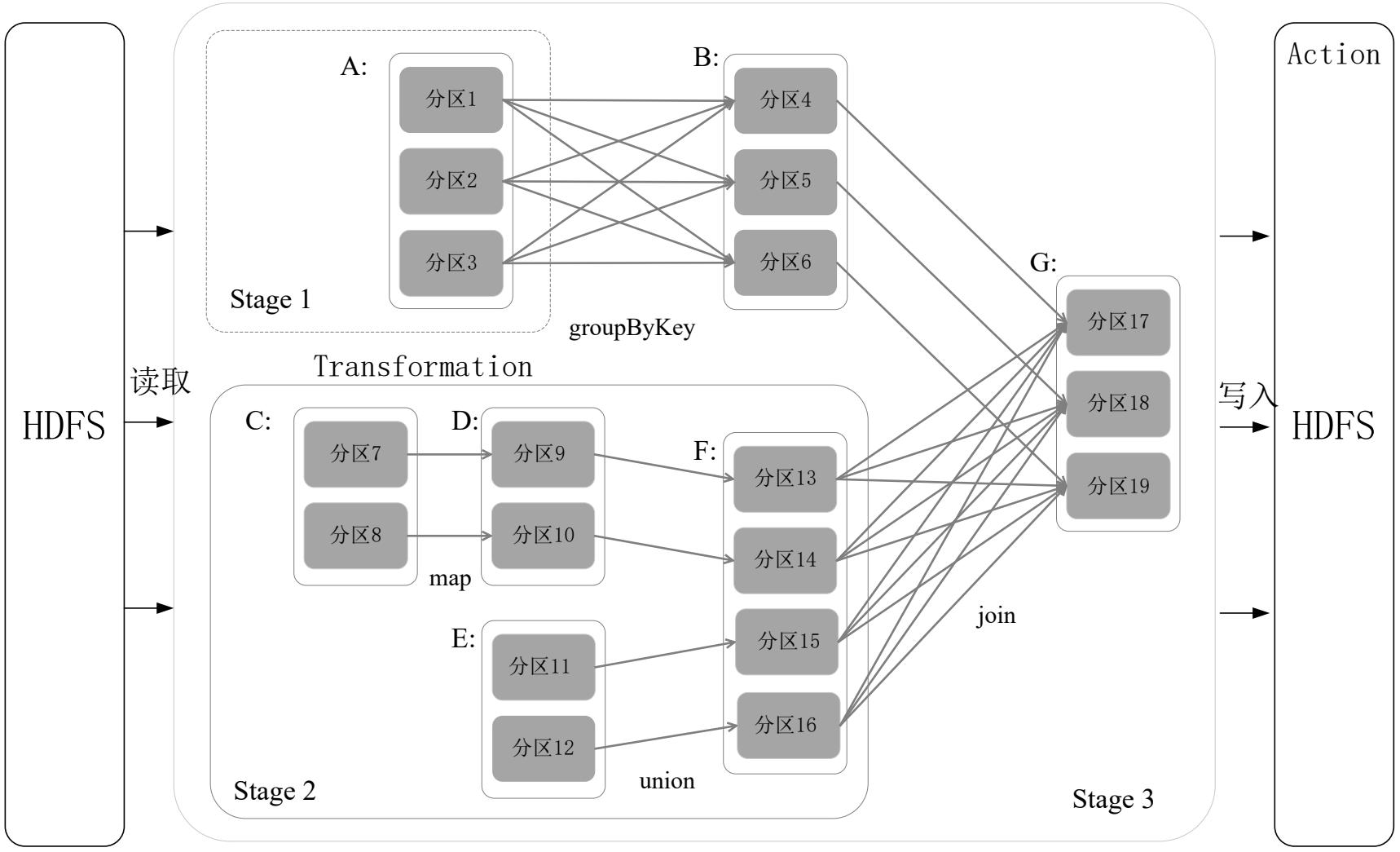
2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



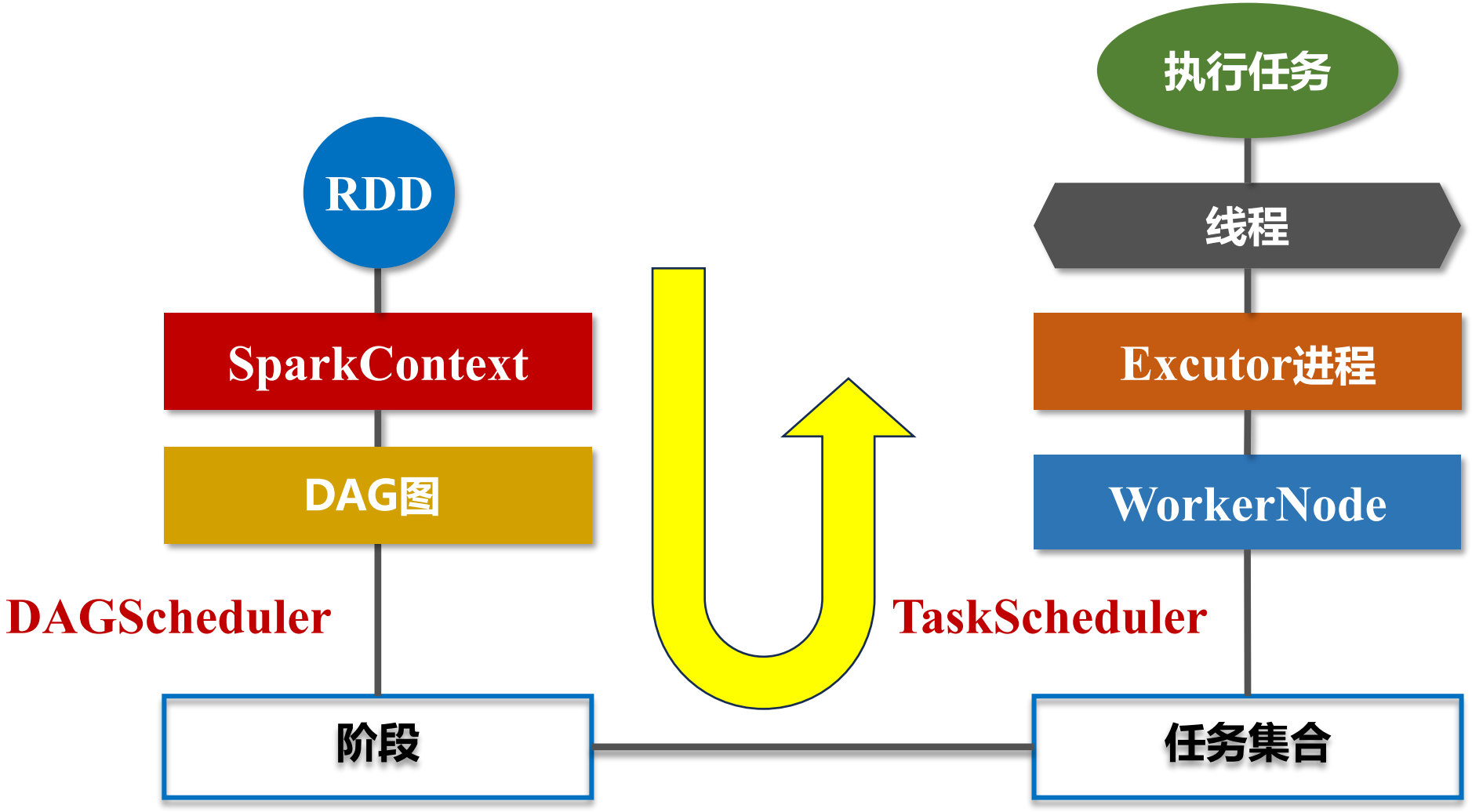
DAG图反向解析



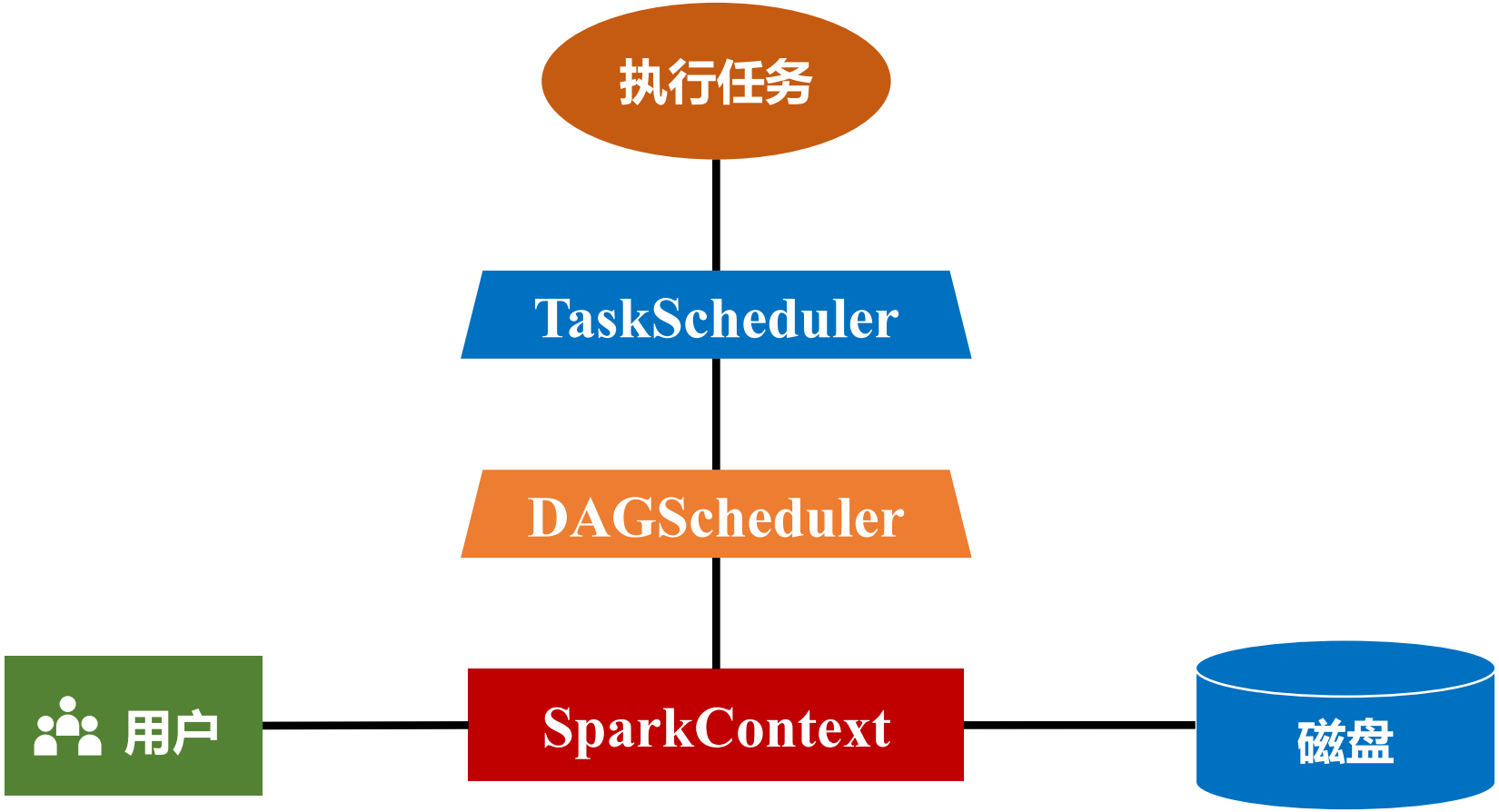
2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



2.3.5 RDD运行原理（阶段的划分和RDD运行过程）





2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



1. Spark中划分Stage的主要依据是什么？

A. RDD的分区数量
B. RDD之间的依赖关系（窄依赖与宽依赖）
C. Transformation操作的类型
D. Action操作的类型



2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



2. 以下哪种依赖关系允许进行流水线优化？

A. 宽依赖
B. 窄依赖
C. Shuffle依赖
D. 全依赖



2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



3. 在窄依赖中，父RDD的一个分区通常对应子RDD的几个分区？

A. 0个
B. 1个
C. 多个
D. 不确定



2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



4. 以下哪个操作通常会导致宽依赖和Shuffle过程？

A. map
B. filter
C. union
D. groupByKey



2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



5. Spark的DAGScheduler的主要职责是什么？

A. 直接在每个Executor上执行任务
B. 将DAG图根据宽依赖划分为多个Stage
C. 管理集群资源
D. 与HDFS进行数据交换



2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



6. 根据Fork/Join机制，Spark的主要优化目标是什么？

A. 减少磁盘使用量
B. 增加数据存储容量
C. 最大化并行执行，减少不必要的等待
D. 避免使用内存计算



2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



7. 当发生Shuffle操作时，数据通常会发生什么行为?

A. 直接在内存中进行流水线处理
B. 被写入磁盘以进行重新分布
C. 被自动压缩以减少网络传输
D. 立即被发送到Driver端



2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



8. 以下哪一项是窄依赖的例子？

A. groupByKey
B. reduceByKey
C. join（未经分区优化的）
D. map



2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



9. Spark中最终的Stage划分和任务调度是由哪个组件完成的？

A. Executor
B. SparkContext
C. DAGScheduler
D. TaskScheduler



2.3.5 RDD运行原理（阶段的划分和RDD运行过程）



10. 为什么宽依赖不能进行流水线优化？

A. 因为数据不需要重新分布
B. 因为父RDD分区可以被多个子RDD分区使用，需等待所有数据准备完毕
C. 因为窄依赖更简单
D. 因为Shuffle操作不需要磁盘IO



Thank you