键和组

在火花

大数据 H/M 2022

Richard McCreadie

动机

- 到目前为止，我们已经讨论了应用无键转换函数
  - 即起作用的功能固定数量的数据项，而不是可变大小 项目组共享一个密钥

- 但是，正如 Google 较早的 Map-Reduce 提案所展示的那样，在许多用例中，您可能希望单个转换器调用处理按有意义的键分组的许多数据项
  - 按平台分析视频游戏
  - 按 GPA 分组学生
  - 按主机分析网页
  - *……还有更多*

- 基于键的分组也有效率优势，因为单个键的所有操作都可以位于单个执行器上，从而减少了数据传输的需要

按关键字分组数据

- 正如我们所看到的，当我们在 spark 中加载数据时，我们会得到一个 Dataset<Row>，它是无键的，我们之前已经看到如何使用 map 函数将这些数据集转换为其他类型

- 为了处理键，Spark 实现了一种更专业的 Dataset 类型：
  - 键值分组数据集<键类型，值类型>

- 顾名思义，这只是一个由一组用户指定的分组键进行逻辑分组的数据集

从数据集到KEYVALUEGROUPEDDATA

- 那么我们如何从 Dataset<V> 转换到 KeyValueGroupedDataset<K,V> 呢?
  - 答案是使用 MapFunction<V,K>，即我们定义了一个新的 MapFunction，它从数据集中的每个项目中提取一个键

    地图(v1) → (k1)

- 数据集支持 groupByKey 方法，该方法使用上面的 MapFunction 来执行转换

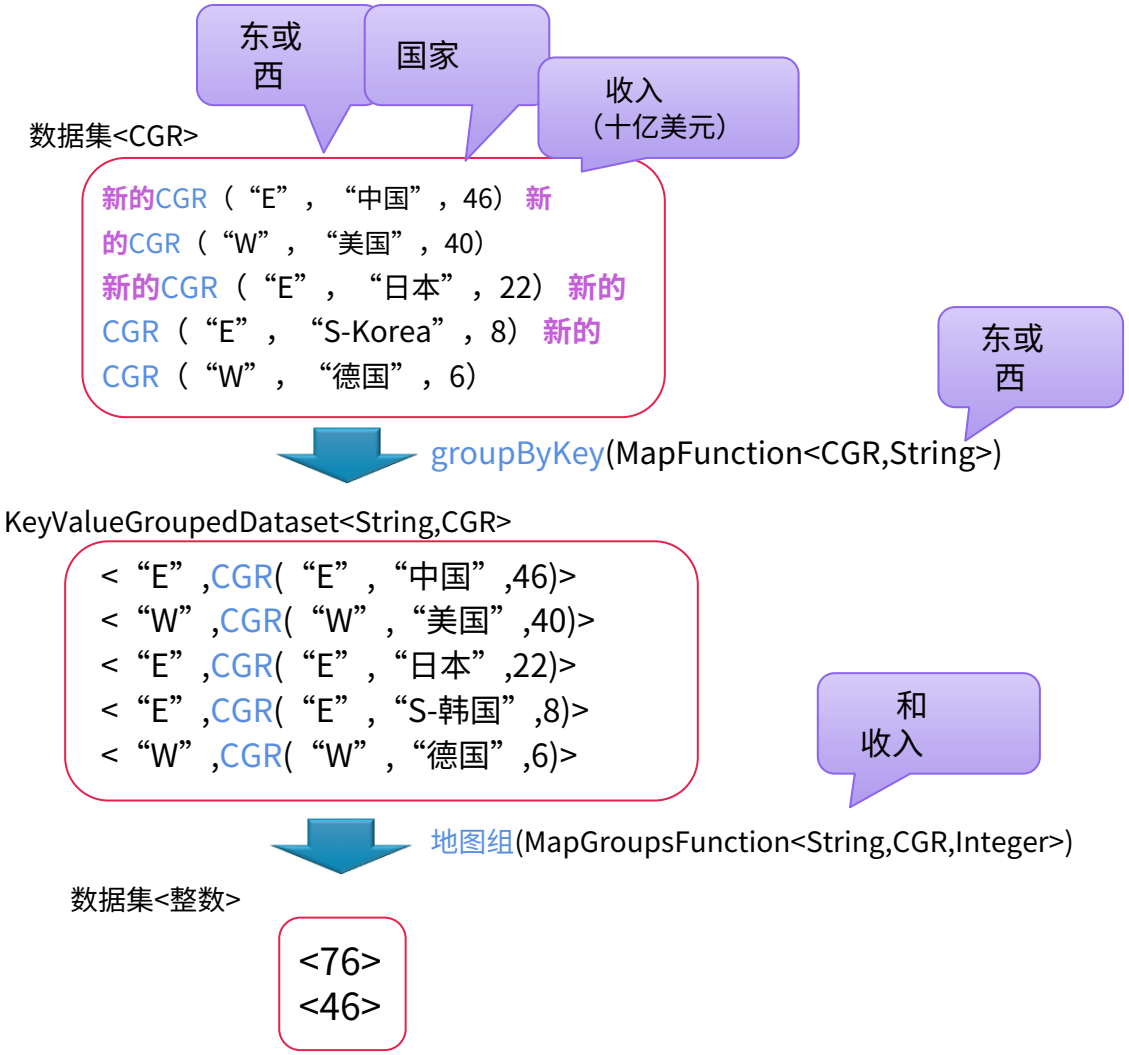
数据集<V>.groupByKey(MapFunction<V,K>, Encoder<K>) → KeyValueGroupedDataset<K,V>

地图组

- KeyValueGroupedDatasets 支持与普通数据集相同的操作类型，只是基于组

- 地图组充当具有相同键的所有项目的聚合器

地图组(k1,Iterator<v1>) → (v2)



平面图组

- FlatMap 组的工作方式类似于地图组，但返回一个迭代器，以便可以返回 0、1 或多个项目

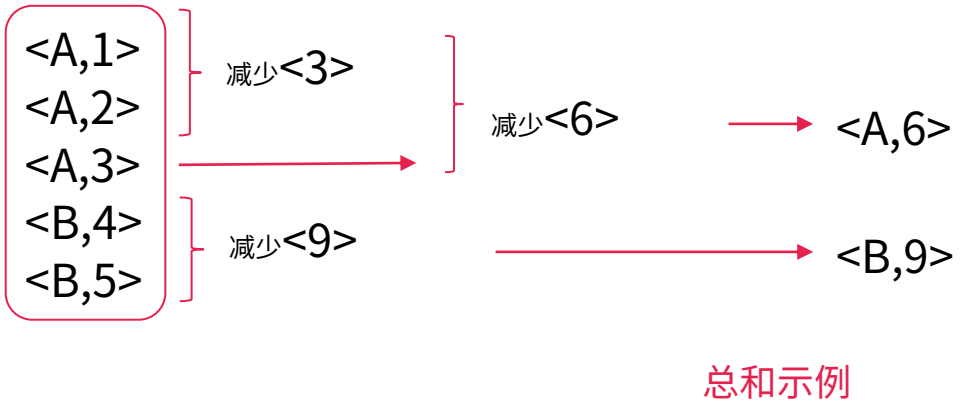
  地图组(k1,Iterator<v1>) → (v2)

  flatMapGroups(k1,Iterator<v1>) → 迭代器<v2>

- 地图功能将 KeyValueGroupedDatasets 转换回普通数据集 通过组合共享每个密钥的所有项目

减少组

- reduce 的基于组的等价物对每个键应用 reduce，返回每个键的减少输出
  - 正因为如此，有**不**特殊的ReduceGroupsFunction，一个ReduceFunction就足以执行reduceGroups
  - reduceGroups 操作的输出是 数据集<Tuple2<K,V>>
    - Tuple2 只是一个用于保存键值对的 Java 类



键值分组数据集。减少组(ReduceFunction<V>) → 数据集<Tuple2<K,V>>