

# 映射和减少

## 在火花

大数据 H/M 2022

*Richard McCreadie*

# 快速提醒

## MAP-REDUCE 的起源

- Map-Reduce 最初是 Google 在 2004 年提出的一种处理范式

原始打印

### MapReduce: simplified data processing on large clusters

[J Dean](#), [S Ghemawat](#) - Communications of the ACM, 2008 - dl.acm.org

... the performance of **MapReduce** on two computations running on a **large cluster** of machines. One computation searches through approximately one terabyte of **data** looking for a particu...

☆ Save ↗ Cite Cited by 22910 Related articles All 97 versions Import into BibTeX

### [HTML] MapReduce: Simplified data processing on large clusters

[J Dean](#), [S Ghemawat](#) - 2004 - usenix.org

... Our implementation of **MapReduce** runs on a **large cluster** of ... **MapReduce** computation **processes** many terabytes of **data** on ... the system **easy** to use: hundreds of **MapReduce** programs ...

☆ Save ↗ Cite Cited by 12349 Related articles All 368 versions Import into BibTeX ⌕

- 这是他们如何管理超大型数据集（例如他们的网络爬虫）的分布式处理的解决方案之一

# GOOGLE 的 MAP-REDUCE

- 最初的 Map-Reduce 论文关注的是用于一般大规模并行分布式计算的接口/API 可能是什么样的
- Map-Reduce 论文的第一个主要贡献是规范了两个 API 方法，中间有一个 shuffle 步骤：

地图(k1, v1) → 列表(k2, v2) shuffle (排序和分组) 减少(k2, 列表(v2)) → 列表(v2)

- 这些是转换函数
  - Map 接受一个 <key,value> 对，对其进行一些处理，然后输出 0 个或多个 <key,value> 对，它们可以是与输入不同的类型
  - shuffle 步骤按键组映射输出
  - Reduce 获取 shuffle 输出（一个键的一批 map 输出），然后对其进行处理，生成一个（较小的）相同类型的项目列表

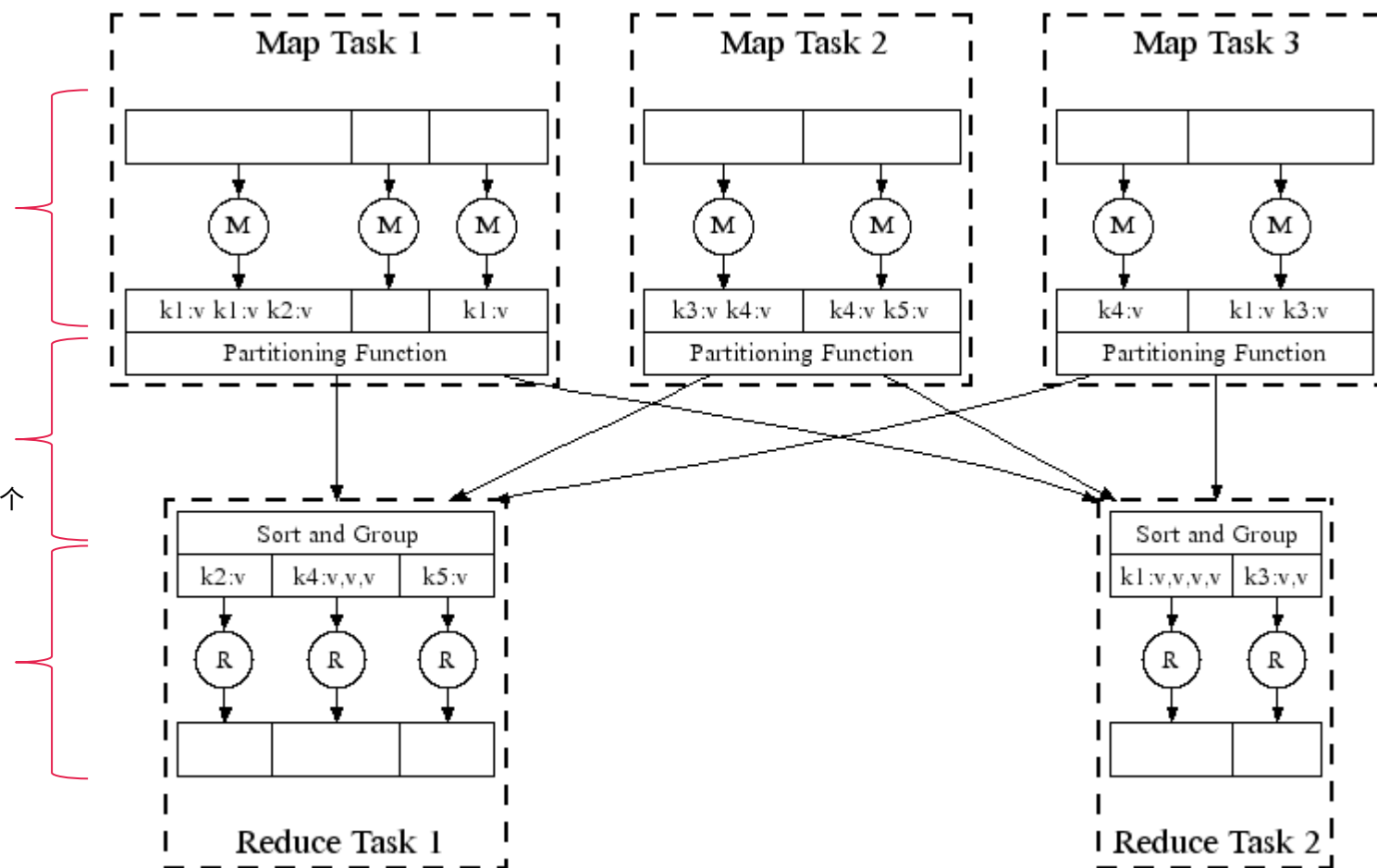
# 可视化

地图(k1, v1) → 列表(k2, v2)

shuffle (排序和分组)

(这涉及到每个 map 任务写入共享数据存储, 然后每个 reducer 从中读取, 此时内存小且昂贵)

减少(k2, 列表(v2)) → 列表(v2)



火花的工作方式不同！





# 火花数据结构：RDD'S

- Google 的 Map-Reduce 范式根据键值对列表定义计算
- Spark 的开发人员指出，并非所有转换都需要有明确的键，而是根据数据元组的集合定义计算，称为弹性分布式数据集(RDD)

```
<1," john" , "smith" , "MSc" , 3.12>  
<2," xi" , "wang" , "MSc" , 3.25>  
<3," jin" , "lin" , "MSc" , 3.11 >
```

RDD

- 元组可以是任何长度并包含任何类型（可以序列化）
- 它们是无模式的（没有说明每个元组中应该包含什么）

# 火花数据结构：

## 数据帧和数据集

- Spark SQL 是后来添加的，它引入了一种替代数据结构，称为数据框
  - 这可以简单地认为是一个带有模式的 RDD，就像一个表头
  - 添加它们是为了使在 Spark 中执行 SQL 操作更容易
- 数据集于 2016 年添加，充当 RDD 和 DataFrame 的抽象
  - 一个数据集<行> == 一个数据框
  - Dataset<AnythingElse> 引入了强类型检查

ID	名	姓	程度	平均绩点
1	约翰	史密斯	理学硕士	3.12
2	习	王	理学硕士	3.25
3	斤	林	理学硕士	3.11

数据框

新的人（“约翰·史密斯”） 新的人（“习”、“王”） 新的人（“晋”、“林”）

由 Person 变量名推断的模式

数据集<人>

名	姓
约翰	史密斯
习	王
斤	林

数据集<人>  
序列化后

# 火花地图和减少

- Apache Spark 也可以不是使用与 Google 原始论文中相同的 map 和 reduce 定义
  - Spark 10 年后发布
  - 硬件环境发生了变化，内存便宜得多
  - 只有线性的map-序列  
> reduce 操作被认为过于严格
- 相反，Spark 定义了广泛的 转换函数

Transformation	Meaning
<b>map</b> ( <i>func</i> )	Return a new distributed dataset formed by passing each element of the source through a function <i>func</i> .
<b>filter</b> ( <i>func</i> )	Return a new dataset formed by selecting those elements of the source on which <i>func</i> returns true.
<b>flatMap</b> ( <i>func</i> )	Similar to map, but each input item can be mapped to 0 or more output items (so <i>func</i> should return a Seq rather than a single item).
<b>mapPartitions</b> ( <i>func</i> )	Similar to map, but runs separately on each partition (block) of the RDD, so <i>func</i> must be of type <code>Iterator&lt;T&gt; =&gt; Iterator&lt;U&gt;</code> when running on an RDD of type T.
<b>mapPartitionsWithIndex</b> ( <i>func</i> )	Similar to mapPartitions, but also provides <i>func</i> with an integer value representing the index of the partition, so <i>func</i> must be of type <code>(Int, Iterator&lt;T&gt;) =&gt; Iterator&lt;U&gt;</code> when running on an RDD of type T.
<b>sample</b> ( <i>withReplacement</i> , <i>fraction</i> , <i>seed</i> )	Sample a fraction <i>fraction</i> of the data, with or without replacement, using a given random number generator seed.
<b>union</b> ( <i>otherDataset</i> )	Return a new dataset that contains the union of the elements in the source dataset and the argument.
<b>intersection</b> ( <i>otherDataset</i> )	Return a new RDD that contains the intersection of elements in the source dataset and the argument.
<b>distinct</b> ( <i>numPartitions</i> )	Return a new dataset that contains the distinct elements of the source dataset.
<b>groupByKey</b> ( <i>numPartitions</i> )	When called on a dataset of (K, V) pairs, returns a dataset of (K, Iterable<V>) pairs. <b>Note:</b> If you are grouping in order to perform an aggregation (such as a sum or average) over each key, using <code>reduceByKey</code> or <code>aggregateByKey</code> will yield much better performance.

…还有很多



# 地图和平面图在火花

- 这地图spark中的操作符可以被认为是对Dataset（或RDD或Dataframe）中的每个项目执行“foreach”的分布式方式
  - 将数据集中的每个项目从一种类型转换为另一种类型

地图(v1) → (v2)

- 然后 Spark 引入了第二个平面图更像谷歌版地图的功能
  - 它不是执行 1-1 映射，而是执行 1-many 映射

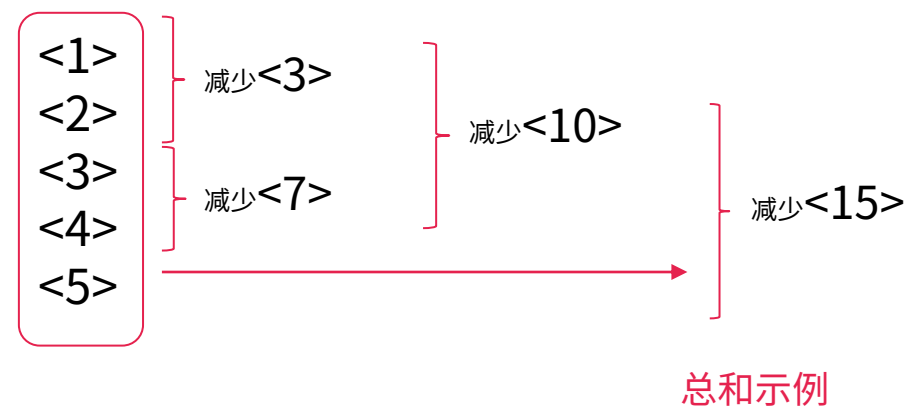
平面图(v1) → 迭代器<v2>

这个迭代器可以输出任意数量的项目，包括 0 如果你需要过滤您的数据集

# 减少在火花

- 与 Google 的设计不同，减少 Spark 中的运算符以成对的方式定义计算，而不是合并列表
  - reduce 操作合并两个相同类型的项目并输出一个相同类型的新项目

减少(v1, v2) → (v3)



- 将对 Dataset 中的每一对迭代地调用 Reduce，直到获得单个输出值。
  - reduce 函数的一个经典示例可能是将值相加