

# Hadoop MapReduce 设计构思

MapReduce 是一个分布式运算程序的编程框架，核心功能是将用户编写的业务逻辑代码和自带默认组件整合成一个完整的分布式运算程序，并发运行在 Hadoop 集群上。

既然是做计算的框架，那么表现形式就是有个输入 (input)，MapReduce 操作这个输入 (input)，通过本身定义好的计算模型，得到一个输出 (output)。

对许多开发者来说，自己完完全全实现一个并行计算程序难度太大，而 MapReduce 就是一种简化并行计算的编程模型，降低了开发并行应用的入门门槛。

Hadoop MapReduce 构思体现在如下的三个方面：

- 如何对付大数据处理：分而治之

对相互间不具有计算依赖关系的大数据，实现并行最自然的办法就是采取分而治之的策略。并行计算的第一个重要问题是如何划分计算任务或者计算数据以便对划分的子任务或数据块同时进行计算。不可分拆的计算任务或相互间有依赖关系的数据无法进行并行计算！

- 构建抽象模型：Map 和 Reduce

MapReduce 借鉴了函数式语言中的思想，用 Map 和 Reduce 两个函数提供了高层的并行编程抽象模型。

Map: 对一组数据元素进行某种重复式的处理；

Reduce: 对 Map 的中间结果进行某种进一步的结果整理。

MapReduce 中定义了如下的 Map 和 Reduce 两个抽象的编程接口，由用户

去编程实现:

$$\text{map: } (k1; v1) \rightarrow [(k2; v2)]$$
$$\text{reduce: } (k2; [v2]) \rightarrow [(k3; v3)]$$

Map 和 Reduce 为程序员提供了一个清晰的操作接口抽象描述。通过以上两个编程接口，大家可以看出 **MapReduce 处理的数据类型是<key,value>键值对**。

- 统一构架，隐藏系统层细节

如何提供统一的计算框架，如果没有统一封装底层细节，那么程序员则需要考虑诸如数据存储、划分、分发、结果收集、错误恢复等诸多细节；为此，MapReduce 设计并提供了统一的计算框架，为程序员隐藏了绝大多数系统层面的处理细节。

MapReduce 最大的亮点在于通过抽象模型和计算框架把需要做什么(what need to do)与具体怎么做(how to do)分开了，为程序员提供一个抽象和高层的编程接口和框架。**程序员仅需要关心其应用层的具体计算问题，仅需编写少量的处理应用本身计算问题的程序代码**。如何具体完成这个并行计算任务所相关的诸多系统层细节被隐藏起来,交给计算框架去处理：从分布代码的执行，大到数千小到单个节点集群的自动调度使用。