# 第五章 MapReduce 应用开发

MapReduce编程遵循一个特定的流程，首先写map函数和reduce函数，最好使用单元测试来确保函数的运行符合预期。然后写一个驱动程序来运行作业。

在开始写MapReduce程序之前，要先学会如何配置和设置开发环境；

## 5.1 用于配置的API

Configuration类实例。

### 5.1.1资源合并

多个配置文件合并，后添加的配置文件属性覆盖之前添加的配置文件属性；

被声明final类型的属性不能被覆盖

### 5.1.2 可变的属性

配置属性可以用其他属性或系统属性定义。

系统属性的优先级大于资源文件中定义的属性。

这种特性适合使用JVM参数 -Dproperty=value来覆盖属性。

## 5.2 配置开发环境

Maven POM 说明了编译和测试MapReduce程序时所需要的依赖项。

### 5.2.1 管理配置

开发MapReduce程序经常要在本地运行和集群运行之间切换。

面对这样的变化的一种方法是是Hadoop配置文件包含每个集群的连接设置，在运行hadoop应用或工具时指定使用哪个一个连接设置。

之后就可以通过-conf命令行开关来使用各种配置。

$ hadoop fs -conf hadoop-localhost.xml -ls .

Hadoop的自带工具也支持-conf选项，也可以在程序中使用Tool接口来支持-conf 选项。5.2.2 辅助类 GenericOptionsParser ，Tool 和ToolRunner

GenericOptionParser是一个类，用来解释Hadoop命令行选项。并根据需要，为Configuration对象设置相应的取值。但是通常不直接使用，更方便的方式是**实现Tool接口，通过ToolRunner来运行程序**。ToolRunnner 内部调用GenericOptionsParser。

## 5.3 用MRUnit来写单元测试

Map函数和reduce函数测试非常方便，这是由函数风格决定的。MRUnit是一个测试库，可以与Junit一起使用。

### 5.3.1 关于Mapper

### 5.3.2 关于reducer

## 5.4 本地运行测试数据

现在在mapper和reducer已经能在受控的输入上进行工作了，下一步写一个作业驱动程序（job driver），在开发机器上使用测试数据运行它。

### 5.4.1 在本地作业运行机器上运行作业

使用前面介绍的Tool接口，可以轻松写一个MapReducer作业的驱动程序。

本地作业运行器只能用于测试，不同于完全的MapReduce实现，最大的区别是它不能运行多个reducer。

在MapReduce 2 中，等价的设置是**mapreduce.framework.name，它必须设置为local**

### 5.4.2 测试驱动程序

## 5.5 在集群上运行

### 5.5.1 打包作业

在分布式环境中，作业的类必须打包进作业的JAR文件中发送给集群。

1. 客户端的类路径
2. 任务的类路径
3. 打包依赖
4. 任务类路径的优先权

### 5.5.2 启动作业

### 5.5.3 MapReduce 的Web界面

### 5.5.4 获取界面

### 5.5.5 作业调试

### 5.5.6 Hadoop日志

### 5.5.7 远程调试

## 5.6 作业调优

作业调优检查表：

1. mapper的数量 如果mapper的运行时间只有几秒则看能不能用更少的mapper运行更长的时间，通常是**一分钟左右**。
2. reducer的数量，在2.x中已经没有reducer的任务槽这种概念了。
3. combiner 作业是否充分利用combiner来减少通过shuffle传输的数据量。
4. 中间值的压缩 对map的输出进行压缩几乎是作业执行的更快
5. 自定义序列化
6. 调整shuffle

### 5.6.1 分析任务

（1）HPROF工具

（2）其他分析工具

## 5.7 MapReduce的工作流

### 5.7.1 将问题分解成MapReduce作业

### 5.7.2 关于JobControl

如何管理这些作业按照顺序执行？

主要考虑是否有个线性的作业链或者一个有向无环图（DAG）

JobControl的实例表示一个作业的运行图

### 5.7.3 关于Apache Oozie

Ooize作为服务器运行，工作流式有一个工作节点和控制流节点组成的DAG。