# 第一章

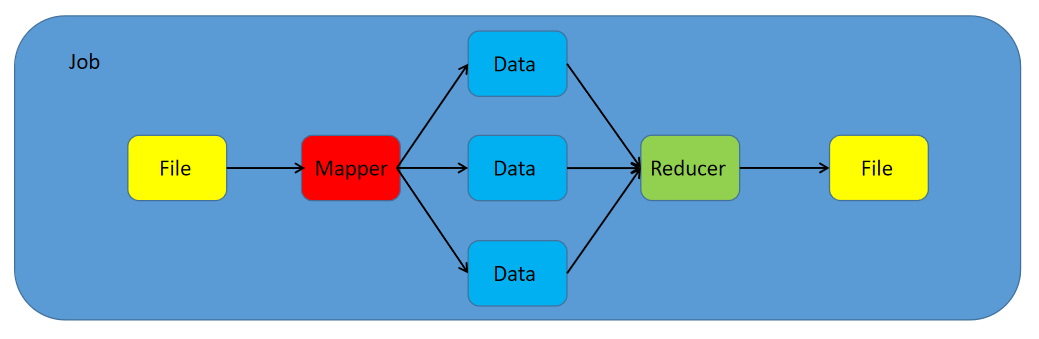
## Spark是什么？

Spark是一种基于内存的快速、通用、可扩展的大数据分析计算引擎。

## Spark VS MapReduce

### MapReduce处理机制

MapReduce设计的初衷主要是用于一次性数据计算，就是框架在处理数据的时候，会从存储设备中读取数据，进行逻辑操作，然后将处理的结果重新存储到介质中。

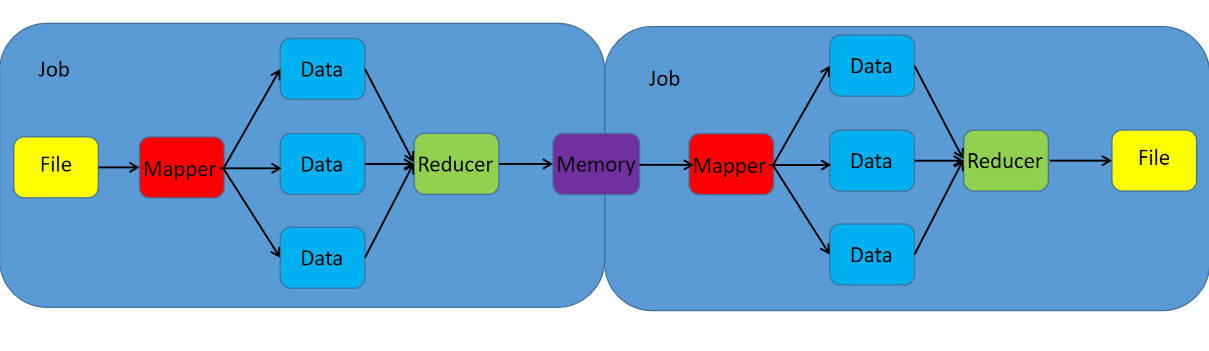


这种计算框架的缺陷在于当我们处理复杂逻辑时的性能是非常低的，因为MapReduce计算引擎中计算模型比较简单，它里面只有Mapper和Reducer，那么这样的话对于我们的上层应用来讲的话，就不得不想方设法去拆分我们的算法，甚至于在上层应用实现多个Job的串联。而且我们可以发现多个Job的之间的关联靠的是磁盘的交互，所以这种我们的磁盘IO会非常的影响性能。

### Spark处理机制

Spark相对于MapReduce提供了更加丰富的数据处理模型，且它可以基于内存来做数据集的多次迭代。如果先不考虑业务复杂度的问题，只考虑数据处理的过程，我们可以发现Spark把计算的结果放到了内存当中，为下次计算提供了更便利的处理方式，这种方式效率非常的高。这也是我们选择Spark而非MapReduce的原因。在内存计算策略和先进的调度机制帮助下，Spark可以更加快速的处理相同的数据集。

而由此带来的问题就是如果Spark部署在共享的集群当中，会遇到资源不足的问题，而且同时会对其他任务产生影响。所以从本质上来看，Spark并不适合和Hadoop堆栈的其他一些组件共同使用。



### Spark和MapReduce的差异

Spark和MapReduce的根本差异是多个作业之间的通信问题：Spark多个作业之间数据通信是基于内存，而MapReduce是基于磁盘。

## Spark核心模块

### Spark Core

Spark Core中提供了Spark最基础与最核心的功能，Spark其他功能如：Spark SQL，Spark Streaming，Graph，MLlib都是在Spark Core的基础上进行扩展的。

### Spark SQL

Spark SQL是Spark用来操作结构化数据的组件。通过Spark SQL，用户可以使用SQL或者Apache Hive版本的SQL方言（HQL）来查询数据。

### Spark Streaming

Spark Streaming是Spark平台上针对实时数据进行流式计算的组件，提供了丰富的处理数据流的API。

### Spark MLlib

MLlib是Spark提供的一个机器学习算法库。MLlib不仅提供了模型评估、数据导入等额外的功能，还提供了一些更底层的机器学习原语。

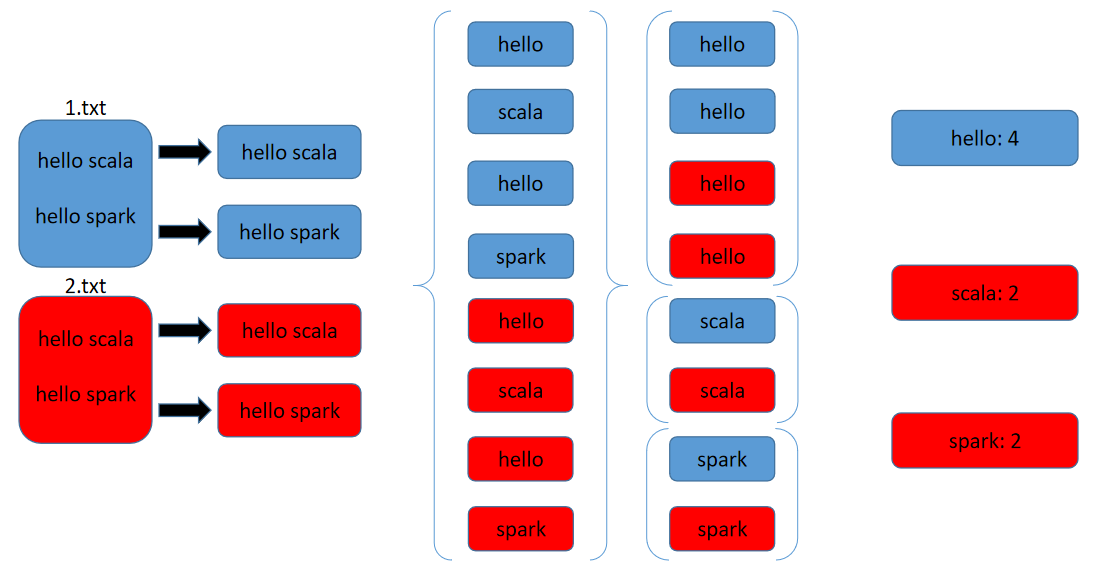
### Spark GraphX

GraphX是Spark面向图计算提供的框架和算法库。

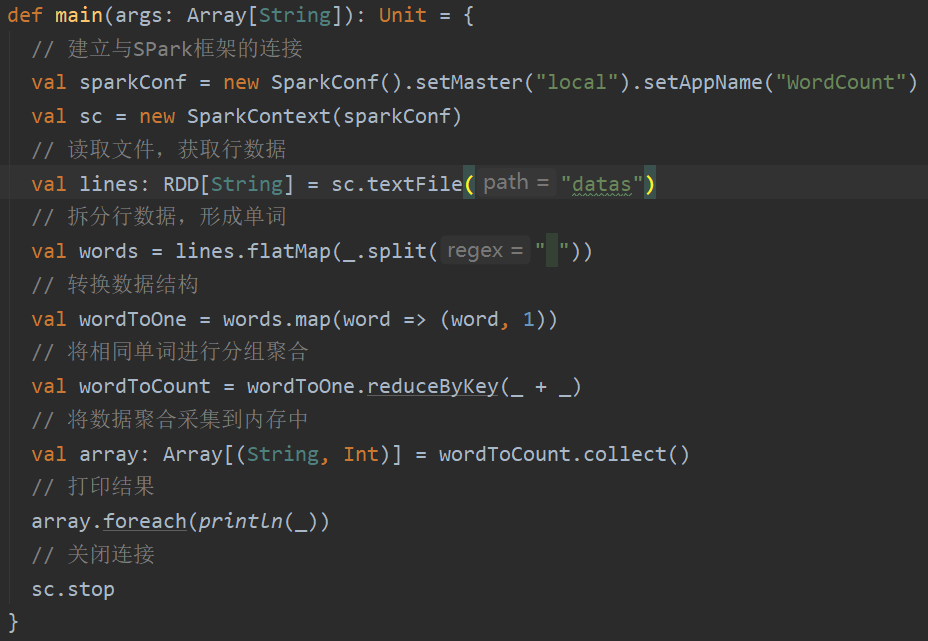
# 第二章

## Word Count

### 逻辑解析

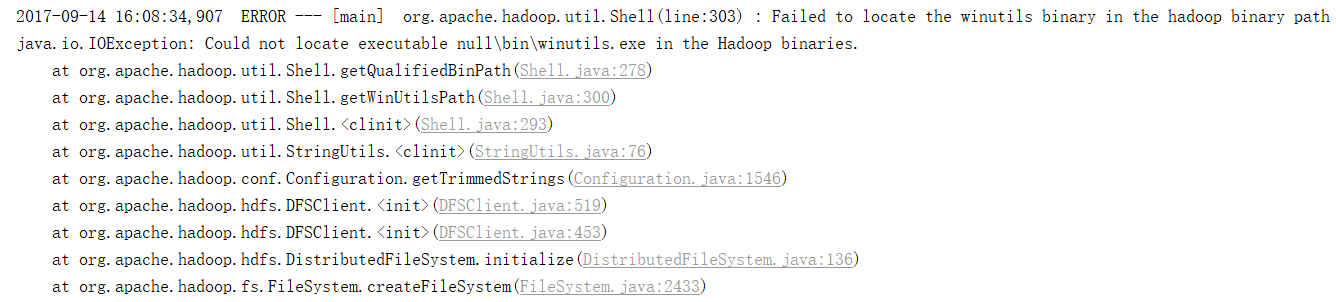


### 代码展示

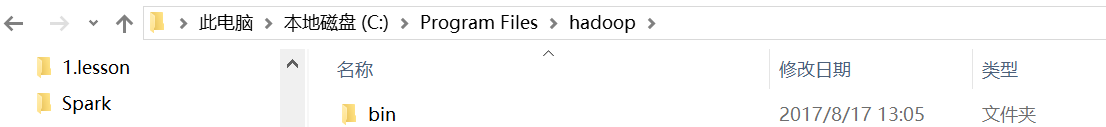


## 异常处理

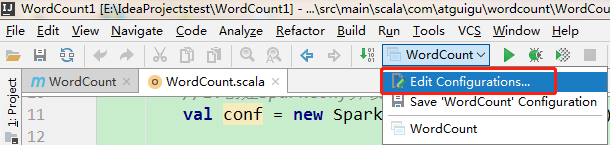
如果本机操作系统是 Windows，在程序中使用了 Hadoop 相关的东西，比如写入文件到HDFS，则会遇到如下异常：

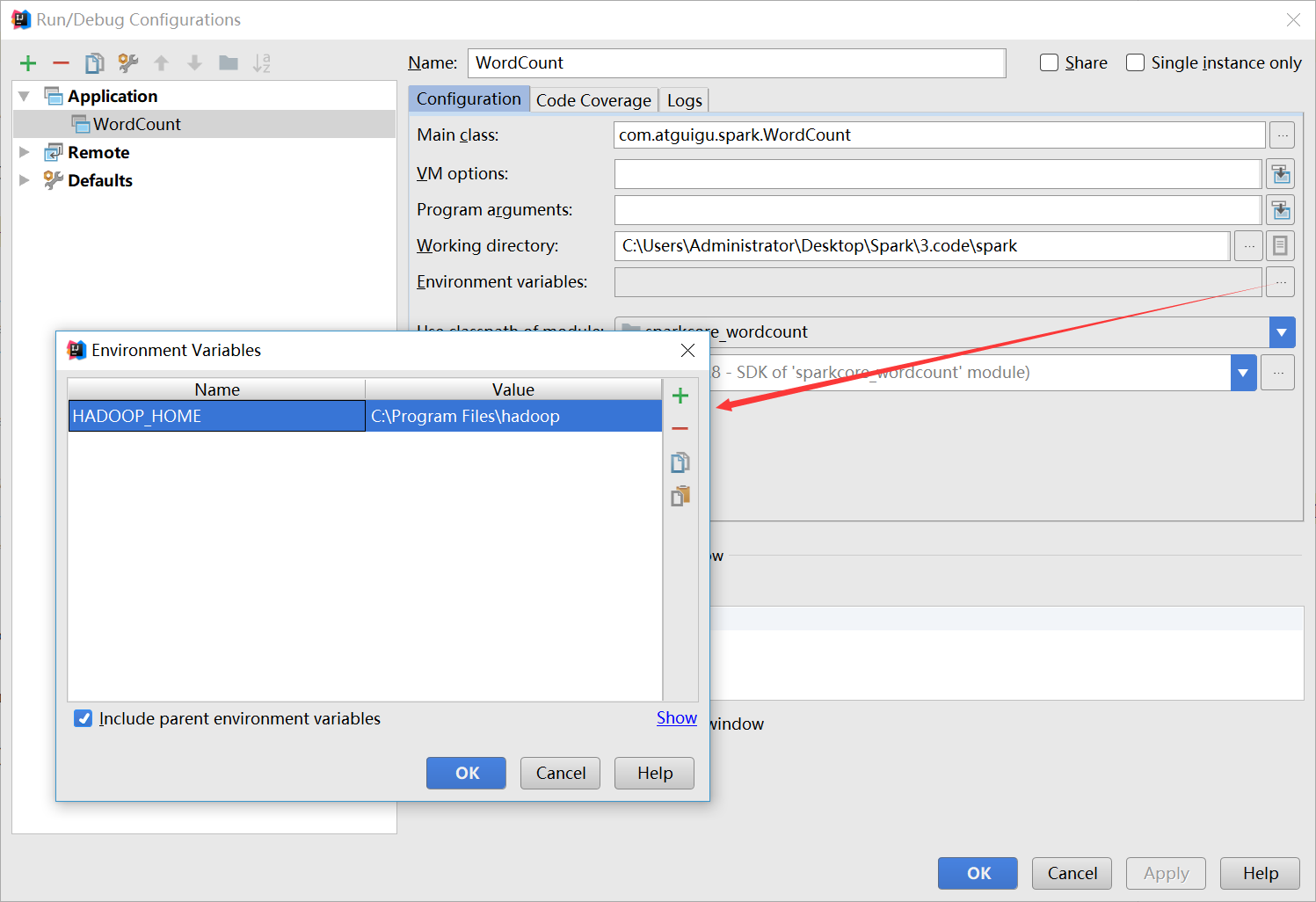


出现这个问题的原因，并不是程序的错误，而是 windows 系统用到了 hadoop 相关的服务，解决办法是通过配置关联到 windows 的系统依赖就可以了。



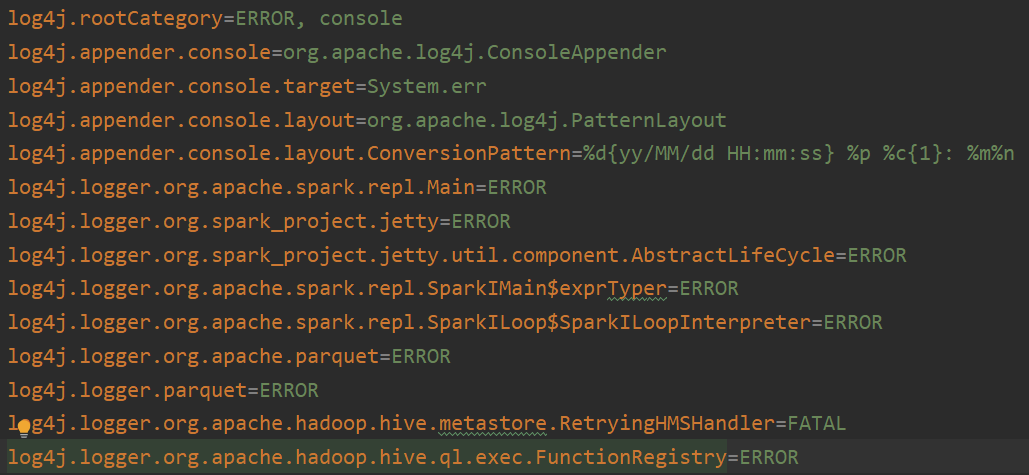
在 IDEA 中配置 Run Configuration，添加 HADOOP\_HOME 变量。





## 日志处理

执行过程中，会产生大量的执行日志，如果为了能够更好的查看程序的执行结果，可以在项目的 resources 目录中创建 log4j.properties 文件，并添加日志配置信息：



# 第三章

## Local模式

不需要其他任何节点就可以在本地执行Spark代码的环境，一般用于教学，演示等。

### 解压缩文件夹

将spark-3.0.0-bin-hadoop3.2.tgz文件上传到Linux并解压缩，放置在指定位置，路径中不要包含中文和空格。

tar -zxvf spark-3.0.0-bin-hadoop3.2.tgz -C /opt/module

cd /opt/module

mv spark-3.0.0-bin-hadoop3.2 spark-local

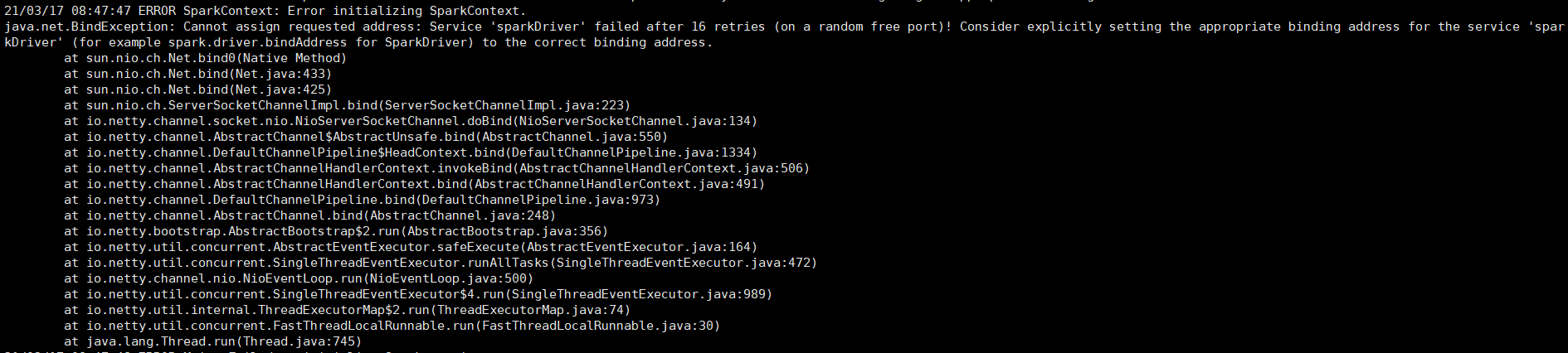
### 启动Local环境

1. 进入解压缩后的路径，执行如下命令

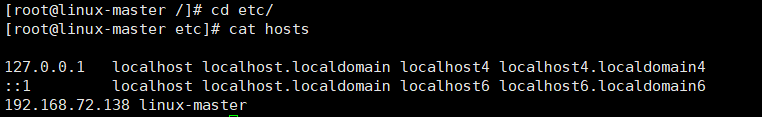
bin/spark-shell

## 异常处理

### 输入bin/spark-shell后报错



修改hosts中的IP地址为虚拟机地址



### Spark UI在虚拟机中可以打开，但是在宿主机上无法访问

原因是VM里防火墙未关闭造成的，执行如下命令关闭防火墙即可

[root@localhost ]# systemctl stop firewalld.service //关闭防火墙服务

[root@localhost ]# systemctl disable firewalld.service //禁止防火墙服务开机启动