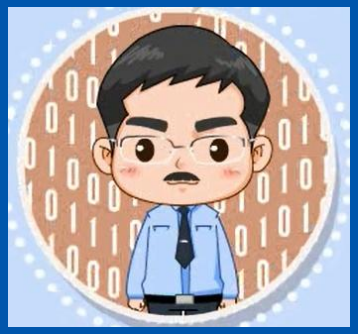


《大数据基础编程、实验和案例教程（第2版）》

教材官网：
http://dblab.xmu.edu.cn/post/bigdatapractice2/

温馨提示：编辑幻灯片母版，可以修改每页PPT的厦大校徽和底部文字

第9章 Spark的安装和基础编程

（PPT版本号：2020年12月版本）

扫一扫访问教材官网

林子雨

厦门大学计算机科学系

E-mail: ziyulin@xmu.edu.cn

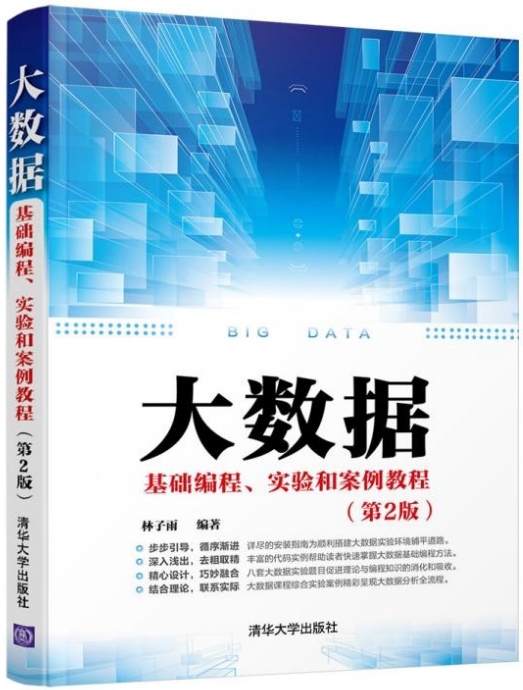
主页：http://dblab.xmu.edu.cn/linziyu

教材简介

本书是与《大数据技术原理与应用（第3版）》教材配套的唯一指定实验指导书

林子雨编著《大数据基础编程、实验和案例教程（第2版）》
清华大学出版社 ISBN:978-7-302-55977-1 定价：69元，2020年10月第2版
教材官网： http://dblab.xmu.edu.cn/post/bigdatapractice2/

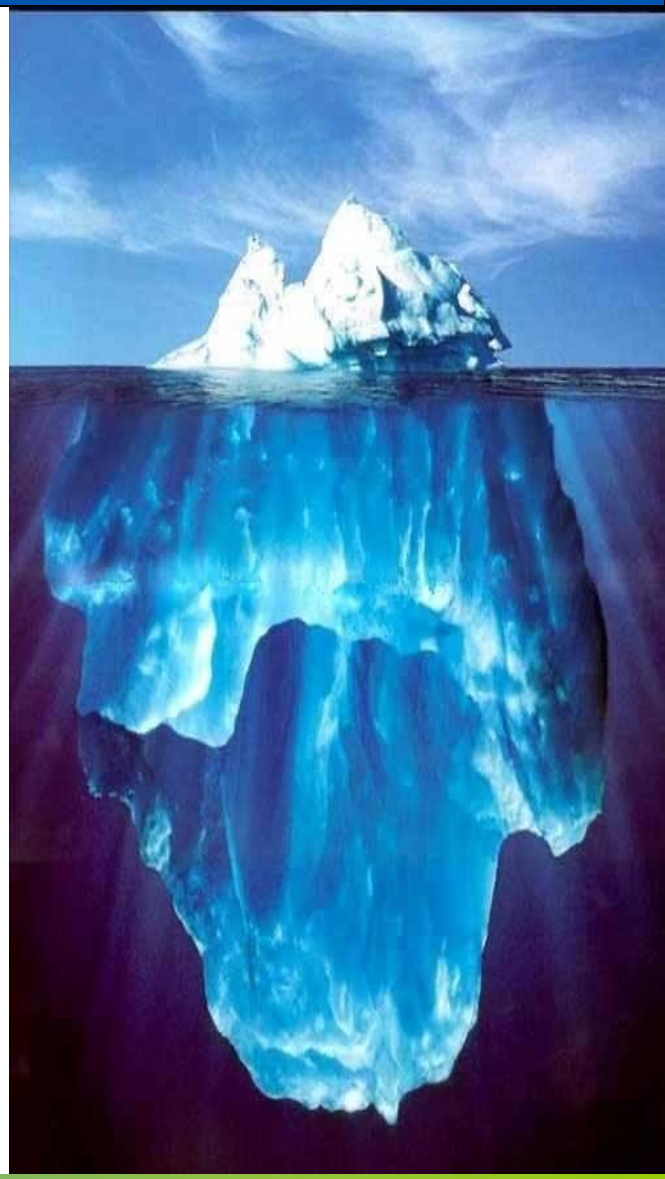
扫一扫访问
教材官网



- 步步引导，循序渐进，详尽的安装指南为顺利搭建大数据实验环境铺平道路
- 深入浅出，去粗取精，丰富的代码实例帮助快速掌握大数据基础编程方法
- 精心设计，巧妙融合，八套大数据实验题目促进理论与编程知识的消化和吸收
- 结合理论，联系实际，大数据课程综合实验案例精彩呈现大数据分析全流程

提纲

9.1基础环境
9.2 安装Spark
9.3 使用 Spark Shell编写代码
9.4 编写Spark独立应用程序



高校大数据课程
公 共 服 务 平 台

百度搜索厦门大学数据库实验室网站访问平台

9.1基础环境

本教程采用如下环境配置。
（1）Linux系统：Ubuntu16.04（或Ubuntu18.04）。
（2）Hadoop：3.1.3版本。
（3）JDK：1.8版本。
（4）Spark：2.4.0版本。
请参照“第2章 Linux系统的安装和使用”完成Linux系统的安装，参照“第3章 Hadoop的安装和使用”完成Hadoop和JDK的安装。

9.2 安装Spark

9.2.1 下载安装文件
9.2.2 配置相关文件

9.2.1 下载安装文件

访问Spark官网（https://archive.apache.org/dist/spark/spark-2.4.0/在页面中选择下载“spark-2.4.0-bin-without-hadoop.tgz”

请使用hadoop用户登录Linux系统，打开一个终端，执行如下命令：

```
$cd ~
$ sudo tar -zxvf ~/Downloads/spark-2.4.0-bin-without-hadoop.tgz -C /usr/local/
$ cd /usr/local
$ sudo mv ./spark-2.4.0-bin-without-hadoop/ ./spark
$ sudo chown -R hadoop:hadoop ./spark    # hadoop是当前登录Linux系统的用户名
```

9.2.2 配置相关文件

安装文件解压缩以后，还需要修改Spark的配置文件spark-env.sh。首先，可以复制一份由Spark安装文件自带的配置文件模板，命令如下：

```
$ cd /usr/local/spark
$ cp ./conf/spark-env.sh.template ./conf/spark-env.sh
```

然后，使用vim编辑器打开spark-env.sh文件进行编辑，在该文件的第一行添加以下配置信息：

```
export SPARK_DIST_CLASSPATH=$(/usr/local/hadoop/bin/hadoop classpath)
```

通过运行Spark自带的实例，可以验证Spark是否安装成功，命令如下：

```
$ cd /usr/local/spark
$ bin/run-example SparkPi
```

9.2.2 配置相关文件

执行时会输出很多屏幕信息，不容易找到最终的输出结果，为了从大量的输出信息中快速找到我们想要的执行结果，可以通过 grep 命令进行过滤:

9.3 使用 Spark Shell编写代码

学习Spark程序开发，建议首先通过Spark Shell进行交互式编程，加深Spark程序开发的理解。Spark Shell提供了简单的方式来学习 API，并且提供了交互的方式来分析数据。你可以输入一条语句，Spark Shell会立即执行语句并返回结果，这就是我们所说的REPL（Read-Eval-Print Loop，交互式解释器），它为我们提供了交互式执行环境，表达式计算完成就会输出结果，而不必等到整个程序运行完毕，因此可即时查看中间结果，并对程序进行修改，这样可以在很大程度上提升开发效率。Spark Shell支持Scala和Python，这里使用 Scala 来进行介绍。Scala是一门现代的多范式编程语言，旨在以简练、优雅及类型安全的方式来表达常用编程模式，它平滑地集成了面向对象和函数语言的特性，运行在JVM（Java 虚拟机）上，并兼容现有的 Java 程序。

9.3.1 启动Spark Shell

可以通过下面命令启动Spark Shell环境：

```
$ cd /usr/local/spark
$ ./bin/spark-shell
```

启动spark-shell后，就会进入“scala>”命令提示符状态

9.3.1 启动Spark Shell

现在，就可以在里面输入Scala代码进行调试了。比如，下面在Scala命令提示符“scala>”后面输入一个表达式“8 * 2 + 5”，然后回车，就会立即得到结果：

```
scala> 8*2+5
res0: Int = 21
```

最后，可以使用命令“:quit”退出Spark Shell，如下所示：

```
scala>:quit
```

9.3.2 读取文件

1.读取本地文件

读取Linux本地文件系统中的文件“/usr/local/spark/README.md”，并显示第一行的内容，命令如下：

```
scala> val textFile = sc.textFile("file:///usr/local/spark/README.md")
scala> textFile.first()
```

9.3.2 读取文件

2.读取HDFS文件

在Spark读取HDFS文件之前，需要首先启动Hadoop，请新建一个Linux终端，执行如下命令：

```
$ cd /usr/local/hadoop
$ ./sbin/start-dfs.sh
```

现在，可以把本地文件“/usr/local/spark/README.md”上传到HDFS的“/user/hadoop”目录下，命令如下：

```
$ cd /usr/local/hadoop
$ ./bin/hdfs dfs –put /usr/local/spark/README.md .
```

上传成功以后，可以使用cat命令输出HDFS中的README.md中的内容，命令如下：

```
$ ./bin/hdfs dfs –cat README.md
```

9.3.2 读取文件

现在请切换回到之前已经打开的Spark Shell窗口，编写语句从HDFS中加载README.md文件，并显示第一行文本内容：

```
scala> val textFile = sc.textFile("hdfs://localhost:9000/user/hadoop/README.md")
scala> textFile.first()
```

如下三条语句都是等价的：

```
scala>  val textFile = sc.textFile("hdfs://localhost:9000/user/hadoop/ README.md ")
scala>  val textFile = sc.textFile("/user/hadoop/ README.md ")
scala>  val textFile = sc.textFile("README.md ")
```

9.3.3 编写词频统计程序

前面我们已经打开了多个Linux终端，现在请切换回到Spark Shell窗口，在“scala>”命令提示符后面输入以下代码：

```
scala> val textFile = sc.textFile("file:///usr/local/spark/ README.md ")
scala> val wordCount = textFile.flatMap(line => line.split(" ")).map(word
=> (word, 1)).reduceByKey((a, b) => a + b)
scala> wordCount.collect()
```

9.4 编写Spark独立应用程序

9.4.1 用Scala语言编写Spark独立应用程序
9.4.2用Java语言编写Spark独立应用程序

9.4.1 用Scala语言编写Spark独立应用程序

1.安装sbt

使用Scala语言编写的Spark程序，需要使用sbt进行编译打包。Spark中没有自带sbt，需要单独安装。可以到“http://www.scala-sbt.org”下载sbt安装文件sbt-1.3.8.tgz。

新建一个终端，在终端中执行如下命令：

```
$ sudo mkdir /usr/local/sbt                    # 创建安装目录
$ cd ~/Downloads
$ sudo tar -zxvf ./sbt-1.3.8.tgz -C /usr/local
$ cd /usr/local/sbt
$ sudo chown -R hadoop /usr/local/sbt          # 此处的hadoop为系统当前用户名
$ cp ./bin/sbt-launch.jar ./  #把bin目录下的sbt-launch.jar复制到sbt安装目录下
```

9.4.1 用Scala语言编写Spark独立应用程序

接着在安装目录中使用下面命令创建一个Shell脚本文件，用于启动sbt：

```
$ vim /usr/local/sbt/sbt
```

该脚本文件中的代码如下：

```
#!/bin/bash
SBT_OPTS="-Xms512M -Xmx1536M -Xss1M -
XX:+CMSClassUnloadingEnabled -XX:MaxPermSize=256M"
java $SBT_OPTS -jar `dirname $0`/sbt-launch.jar "$@"
```

保存后，还需要为该Shell脚本文件增加可执行权限：

```
$ chmod u+x /usr/local/sbt/sbt
```

9.4.1 用Scala语言编写Spark独立应用程序

然后，可以使用如下命令查看sbt版本信息：

```
$ cd /usr/local/sbt
$ ./sbt sbtVersion
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option
MaxPermSize=256M; support was removed in 8.0
[warn] No sbt.version set in project/build.properties, base directory:
/usr/local/sbt
[info] Set current project to sbt (in build file:/usr/local/sbt/)
[info] 1.3.8
```

9.4.1 用Scala语言编写Spark独立应用程序

2.编写Scala应用程序代码

在终端中执行如下命令创建一个文件夹 sparkapp作为应用程序根目录：

```
$ cd ~           # 进入用户主文件夹
$ mkdir ./sparkapp       # 创建应用程序根目录
$ mkdir -p ./sparkapp/src/main/scala     # 创建所需的文件夹结构
```

下面使用vim编辑器在“~/sparkapp/src/main/scala”下建立一个名为SimpleApp.scala的Scala代码文件，命令如下：

```
$ cd ~
$ vim ./sparkapp/src/main/scala/SimpleApp.scala
```

9.4.1 用Scala语言编写Spark独立应用程序

```
/* SimpleApp.scala */
import org.apache.spark.SparkContext
import org.apache.spark.SparkContext._
import org.apache.spark.SparkConf

object SimpleApp {
   def main(args: Array[String]) {
      val logFile = "file:///usr/local/spark/README.md" // Should be some file on your system
      val conf = new SparkConf().setAppName("Simple Application")
      val sc = new SparkContext(conf)
      val logData = sc.textFile(logFile, 2).cache()
      val numAs = logData.filter(line => line.contains("a")).count()
      val numBs = logData.filter(line => line.contains("b")).count()
      println("Lines with a: %s, Lines with b: %s".format(numAs, numBs))
   }
}
```

9.4.1 用Scala语言编写Spark独立应用程序

3.用sbt打包Scala应用程序

SimpleApp.scala程序依赖于Spark API，因此，需要通过sbt进行编译打包。 首先，需要使用vim编辑器在“~/sparkapp”目录下新建文件simple.sbt，命令如下：

```
$ cd ~
$ vim ./sparkapp/simple.sbt
```

simple.sbt文件用于声明该独立应用程序的信息以及与 Spark的依赖关系，需要在simple.sbt文件中输入以下内容：

```
name := "Simple Project"
version := "1.0"
scalaVersion := "2.11.12"
libraryDependencies += "org.apache.spark" %% "spark-core" % "2.4.0"
```

9.4.1 用Scala语言编写Spark独立应用程序

为了保证sbt能够正常运行，先执行如下命令检查整个应用程序的文件结构：

```
$ cd ~/sparkapp
$ find .
```

文件结构应该是类似如下所示的内容：

```
.
./src
./src/main
./src/main/scala
./src/main/scala/SimpleApp.scala
./simple.sbt
```

9.4.1 用Scala语言编写Spark独立应用程序

接下来，可以通过如下代码将整个应用程序打包成 JAR（首次运行时，sbt 会自动下载相关的依赖包）：

```
$ cd ~/sparkapp  #一定把这个目录设置为当前目录
$ /usr/local/sbt/sbt package
```

执行上述命令后，屏幕上会返回如下类似信息：

```
$~/sparkapp$ /usr/local/sbt/sbt package
[info] Set current project to Simple Project
[info] Updating {file:/home/hadoop/sparkapp/}sparkapp...
[info] Done updating.
[info] Compiling 1 Scala source to /home/hadoop/sparkapp/target/...
[info] Packaging /home/hadoop/sparkapp/target/scala-2.11/...
[info] Done packaging.
[success] Total time: 17 s, completed 2020-1-27 16:13:56
```

生成的JAR包的位置为“~/sparkapp/target/scala-2.11/simple-project_2.11-1.0.jar”。

9.4.1 用Scala语言编写Spark独立应用程序

4.通过spark-submit运行程序

最后，可以将生成的JAR包通过spark-submit提交到Spark中运行，命令如下：

```
$/usr/local/spark/bin/spark-submit --class "SimpleApp"
~/sparkapp/target/scala-2.11/simple-project_2.11-1.0.jar
```

上面命令执行后会输出太多信息，可以不使用上面命令，而使用下面命令运行程序，这样就可以直接得到想要的结果：

```
$ /usr/local/spark/bin/spark-submit --class "SimpleApp"
~/sparkapp/target/scala-2.11/simple-project_2.11-1.0.jar 2>&1 | grep "Lines
with a:"
```

最终得到的结果如下：

```
Lines with a: 62, Lines with b: 31
```

9.4.2用Java语言编写Spark独立应用程序

1. 安装Maven

Ubuntu中没有自带安装Maven，需要手动安装Maven。可以访问Maven官网下载安装文件，下载地址如下：

https://downloads.apache.org/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.zip

然后，可以选择安装在“/usr/local/maven”目录中，命令如下：

```
$ sudo unzip ~/下载/apache-maven-3.6.3-bin.zip -d /usr/local
$ cd /usr/local
$ sudo mv apache-maven-3.6.3/ ./maven
$ sudo chown -R hadoop ./maven
```

9.4.2用Java语言编写Spark独立应用程序

2.编写Java应用程序代码

在Linux终端中执行如下命令，在用户主文件夹下创建一个文件夹sparkapp2作为应用程序根目录：

```
$ cd ~ #进入用户主文件夹
$ mkdir -p ./sparkapp2/src/main/java
```

然后，使用vim编辑器在“./sparkapp2/src/main/java”目录下建立一个名为 SimpleApp.java的文件，命令如下：

```
$ vim ./sparkapp2/src/main/java/SimpleApp.java
```

9.4.2用Java语言编写Spark独立应用程序

在SimpleApp.java文件中输入如下代码：

```
/*** SimpleApp.java ***/
import org.apache.spark.api.java.*;
import org.apache.spark.api.java.function.Function;
import org.apache.spark.SparkConf;

public class SimpleApp {
   public static void main(String[] args) {
      String logFile = "file:///usr/local/spark/README.md"; // Should be some file on your system
      SparkConf conf=new SparkConf().setMaster("local").setAppName("SimpleApp");
                  JavaSparkContext sc=new JavaSparkContext(conf);
      JavaRDD<String> logData = sc.textFile(logFile).cache();
      long numAs = logData.filter(new Function<String, Boolean>() {
         public Boolean call(String s) { return s.contains("a"); }
      }).count();
      long numBs = logData.filter(new Function<String, Boolean>() {
         public Boolean call(String s) { return s.contains("b"); }
      }).count();
      System.out.println("Lines with a: " + numAs + ", lines with b: " + numBs);
   }
}
```

9.4.2用Java语言编写Spark独立应用程序

该程序依赖Spark Java API，因此，我们需要通过Maven进行编译打包。需要使用vim编辑器在“~/sparkapp2”目录中新建文件pom.xml，命令如下：

```
$ cd ~
$ vim ./sparkapp2/pom.xml
```

9.4.2用Java语言编写Spark独立应用程序

然后，在pom.xml文件中添加如下内容，用来声明该独立应用程序的信息以及与Spark的依赖关系：

```
<project>
    <groupId>cn.edu.xmu</groupId>
    <artifactId>simple-project</artifactId>
    <modelVersion>4.0.0</modelVersion>
    <name>Simple Project</name>
    <packaging>jar</packaging>
    <version>1.0</version>
    <repositories>
        <repository>
            <id>jboss</id>
            <name>JBoss Repository</name>
            <url>http://repository.jboss.com/maven2/</url>
        </repository>
    </repositories>
    <dependencies>
        <dependency> <!-- Spark dependency -->
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-core_2.11</artifactId>
            <version>2.4.0</version>
        </dependency>
    </dependencies>
</project>
```

9.4.2用Java语言编写Spark独立应用程序

3.使用Maven打包Java程序

为了保证Maven能够正常运行，先执行如下命令检查整个应用程序的文件结构：

```
$ cd ~/sparkapp2
$ find .
```

文件结构应该是类似如下的内容：

```
.
./pom.xml
./src
./src/main
./src/main/java
./src/main/java/SimpleApp.java
```

9.4.2用Java语言编写Spark独立应用程序

接下来，我们可以通过如下代码将整个应用程序打包成JAR包（注意：计算机需要保持连接网络的状态，而且首次运行打包命令时，Maven会自动下载依赖包，需要消耗几分钟的时间）：

```
$ cd ~/sparkapp2    #一定把这个目录设置为当前目录
$ /usr/local/maven/bin/mvn package
```

如果屏幕返回如下信息，则说明生成JAR包成功：

```
[INFO]------------------------------------
[INFO] BUILD SUCCESS
[INFO]------------------------------------
[INFO] Total time: 10.847 s
[INFO] Finished at: 2020-01-07T 16:33:33+08:00
[INFO] Final Memory: 30M/132M
[INFO]------------------------------------
```

4.通过spark-submit 运行程序

最后，可以将生成的JAR包通过spark-submit提交到Spark中运行，命令如下：

```
$ /usr/local/spark/bin/spark-submit --class "SimpleApp"
~/sparkapp2/target/simple-project-1.0.jar
```

上面命令执行后会输出太多信息，可以不使用上面命令，而使用下面命令运行程序，这样就可以直接得到想要的结果：

```
$ /usr/local/spark/bin/spark-submit --class "SimpleApp"
~/sparkapp2/target/simple-project-1.0.jar 2>&1 | grep "Lines with a"
```

最终得到的结果如下：

```
Lines with a: 62, Lines with b: 31
```

9.5 本章小结

Spark是基于内存的分布式计算框架，减少了迭代计算时的IO开销。虽然，Hadoop已成为大数据的事实标准，但是MapReduce分布式计算模型仍存在诸多缺陷，而Spark不仅具备了Hadoop MapReduce的优点，而且解决了Hadoop MapReduce的缺陷。Spark正以其结构一体化、功能多元化的优势逐渐成为当今大数据领域最热门的大数据计算平台。

本章详细介绍了Spark的安装配置方法，并且把Spark配置为和Hadoop一起使用，可以让 Spark访问HDFS中的数据。
Spark Shell提供了简单的方式来学习 API，并且提供了交互的方式来分析数据。本章详细介绍了如何启动Spark Shell、读取文件以及如何编写词频统计程序。

可以使用Spark API编写独立应用程序。使用Scala语言编写的程序需要使用sbt进行编译打包，相应地，使用Java语言编写的Spark程序需要使用Maven进行编译打包。本章最后分别介绍了如何使用Scala和Java两种语言编译运行Spark独立应用程序。

附录A：主讲教师林子雨简介

主讲教师：林子雨

单位：厦门大学计算机科学系
E-mail: ziyulin@xmu.edu.cn
个人网页：http://dblab.xmu.edu.cn/post/linziyu
数据库实验室网站：http://dblab.xmu.edu.cn

扫一扫访问个人主页

林子雨，男，1978年出生，博士（毕业于北京大学），全国高校知名大数据教师，现为厦门大学计算机科学系副教授，曾任厦门大学信息科学与技术学院院长助理、晋江市发展和改革局副局长。中国计算机学会数据库专业委员会委员，中国计算机学会信息系统专业委员会委员。国内高校首个“数字教师”提出者和建设者，厦门大学数据库实验室负责人，厦门大学云计算与大数据研究中心主要建设者和骨干成员，2013年度、2017年度和2020年度厦门大学教学类奖教金获得者，荣获2019年福建省精品在线开放课程、2018年厦门大学高等教育成果特等奖、2018年福建省高等教育教学成果二等奖、2018年国家精品在线开放课程。主要研究方向为数据库、数据仓库、数据挖掘、大数据、云计算和物联网，并以第一作者身份在《软件学报》《计算机学报》和《计算机研究与发展》等国家重点期刊以及国际学术会议上发表多篇学术论文。作为项目负责人主持的科研项目包括1项国家自然科学青年基金项目(No.61303004)、1项福建省自然科学青年基金项目(No.2013J05099)和1项中央高校基本科研业务费项目(No.2011121049)，主持的教改课题包括1项2016年福建省教改课题和1项2016年教育部产学协作育人项目，同时，作为课题负责人完成了国家发改委城市信息化重大课题、国家物联网重大应用示范工程区域试点泉州市工作方案、2015泉州市互联网经济调研等课题。中国高校首个“数字教师”提出者和建设者，2009年至今，“数字教师”大平台累计向网络免费发布超过1000万字高价值的研究和教学资料，累计网络访问量超过1000万次。打造了中国高校大数据教学知名品牌，编著出版了中国高校第一本系统介绍大数据知识的专业教材《大数据技术原理与应用》，并成为京东、当当网等网店畅销书籍；建设了国内高校首个大数据课程公共服务平台，为教师教学和学生学习大数据课程提供全方位、一站式服务，年访问量超过200万次，累计访问量超过1000万次。

附录B：大数据学习路线图

大数据学习路线图访问地址：http://dblab.xmu.edu.cn/post/10164/

附录C：林子雨大数据系列教材

了解全部教材信息：http://dblab.xmu.edu.cn/post/bigdatabook/

附录D：《大数据导论（通识课版）》教材

开设全校公共选修课的优质教材

本课程旨在实现以下几个培养目标：

- 引导学生步入大数据时代，积极投身大数据的变革浪潮之中
- 了解大数据概念，培养大数据思维，养成数据安全意识
- 认识大数据伦理，努力使自己的行为符合大数据伦理规范要求
- 熟悉大数据应用，探寻大数据与自己专业的应用结合点
- 激发学生基于大数据的创新创业热情

高等教育出版社 ISBN:978-7-04-053577-8 定价：32元

教材官网：http://dblab.xmu.edu.cn/post/bigdataintroduction/

附录E：《大数据导论》教材

- 林子雨 编著《大数据导论》
- 人民邮电出版社，2020年9月第1版
- ISBN:978-7-115-54446-9 定价：49.80元

教材官网：http://dblab.xmu.edu.cn/post/bigdata-introduction/

开设大数据专业导论课的优质教材

扫一扫访问教材官网

附录F：《大数据技术原理与应用》教材

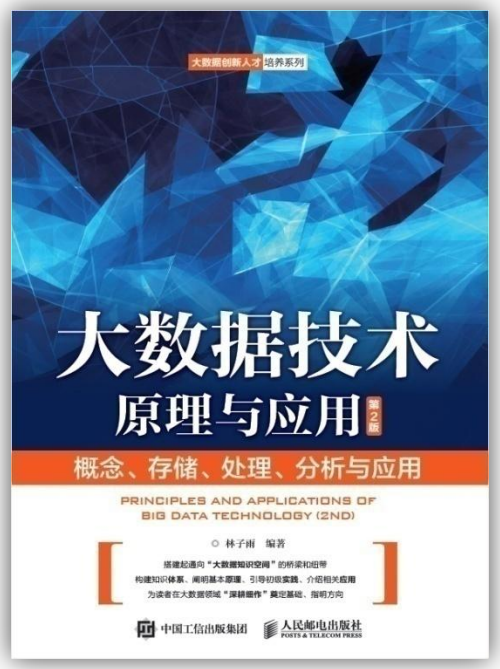
《大数据技术原理与应用——概念、存储、处理、分析与应用（第2版）》，由厦门大学计算机科学系林子雨博士编著，是国内高校第一本系统介绍大数据知识的专业教材。人民邮电出版社 ISBN:978-7-115-44330-4 定价：49.80元

全书共有15章，系统地论述了大数据的基本概念、大数据处理架构Hadoop、分布式文件系统HDFS、分布式数据 库HBase、NoSQL数据库、云数据库、分布式并行编程模型MapReduce、Spark、流计算、图计算、数据可视化以及大数据在互联网、生物医学和物流等各个领域的应用。在Hadoop、HDFS、HBase和MapReduce等重要章节，安排了入门级的实践操作，让读者更好地学习和掌握大数据关键技术。

本书可以作为高等院校计算机专业、信息管理等相关专业的大数据课程教材，也可供相关技术人员参考、学习、培训之用。

欢迎访问《大数据技术原理与应用——概念、存储、处理、分析与应用》教材官方网站：
http://dblab.xmu.edu.cn/post/bigdata

扫一扫访问教材官网



附录G：《大数据基础编程、实验和案例教程（第2版）》

本书是与《大数据技术原理与应用（第3版）》教材配套的唯一指定实验指导书

大数据教材

1+1黄金组合

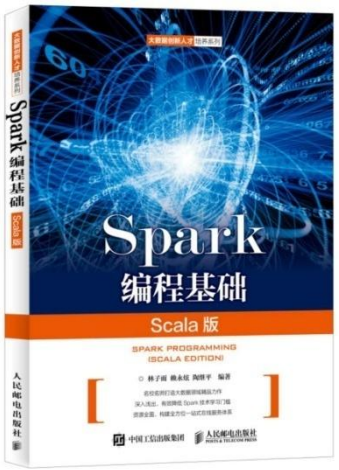
厦门大学林子雨编著

配套实验指导书

- 步步引导，循序渐进，详尽的安装指南为顺利搭建大数据实验环境铺平道路
- 深入浅出，去粗取精，丰富的代码实例帮助快速掌握大数据基础编程方法
- 精心设计，巧妙融合，八套大数据实验题目促进理论与编程知识的消化和吸收
- 结合理论，联系实际，大数据课程综合实验案例精彩呈现大数据分析全流程

林子雨编著《大数据基础编程、实验和案例教程（第2版）》
清华大学出版社 ISBN:978-7-302-55977-1 定价：69元 2020年10月第2版

附录H：《Spark编程基础（Scala版）》



《Spark编程基础（Scala版）》

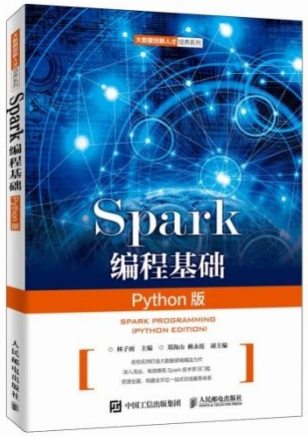
厦门大学 林子雨，赖永炫，陶继平 编著

披荆斩棘，在大数据丛林中开辟学习捷径
填沟削坎，为快速学习Spark技术铺平道路
深入浅出，有效降低Spark技术学习门槛
资源全面，构建全方位一站式在线服务体系

人民邮电出版社出版发行，ISBN:978-7-115-48816-9
教材官网：http://dblab.xmu.edu.cn/post/spark/

本书以Scala作为开发Spark应用程序的编程语言，系统介绍了Spark编程的基础知识。全书共8章，内容包括大数据技术概述、Scala语言基础、Spark的设计与运行原理、Spark环境搭建和使用方法、RDD编程、Spark SQL、Spark Streaming、Spark MLlib等。本书每个章节都安排了入门级的编程实践操作，以便读者更好地学习和掌握Spark编程方法。本书官网免费提供了全套的在线教学资源，包括讲义PPT、习题、源代码、软件、数据集、授课视频、上机实验指南等。

附录I：《Spark编程基础（Python版）》



《Spark编程基础（Python版）》

厦门大学 林子雨，郑海山，赖永炫 编著

披荆斩棘，在大数据丛林中开辟学习捷径
填沟削坎，为快速学习Spark技术铺平道路
深入浅出，有效降低Spark技术学习门槛
资源全面，构建全方位一站式在线服务体系

人民邮电出版社出版发行，ISBN:978-7-115-52439-3
教材官网： http://dblab.xmu.edu.cn/post/spark-python/

本书以Python作为开发Spark应用程序的编程语言，系统介绍了Spark编程的基础知识。全书共8章，内容包括大数据技术概述、Spark的设计与运行原理、Spark环境搭建和使用方法、RDD编程、Spark SQL、Spark Streaming、Structured Streaming、Spark MLlib等。本书每个章节都安排了入门级的编程实践操作，以便读者更好地学习和掌握Spark编程方法。本书官网免费提供了全套的在线教学资源，包括讲义PPT、习题、源代码、软件、数据集、上机实验指南等。

附录J：高校大数据课程公共服务平台

高校大数据课程

公 共 服 务 平 台

http://dblab.xmu.edu.cn/post/bigdata-teaching-platform/

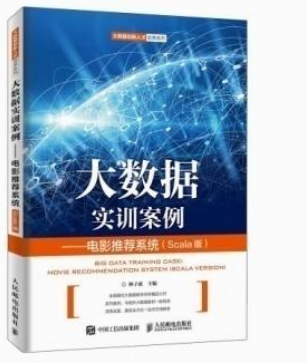
扫一扫访问平台主页

扫一扫观看3分钟FLASH动画宣传片

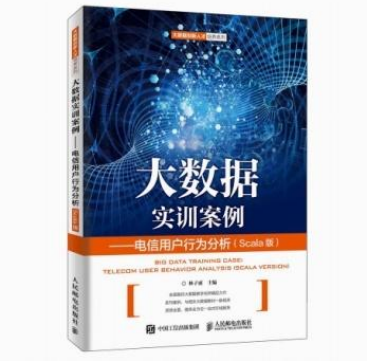
附录K：高校大数据实训课程系列案例教材

为了更好满足高校开设大数据实训课程的教材需求，厦门大学数据库实验室林子雨老师团队联合企业共同开发了《高校大数据实训课程系列案例》，目前已经完成开发的系列案例包括：

《电影推荐系统》（已经于2019年5月出版）
《电信用户行为分析》（已经于2019年5月出版）
《实时日志流处理分析》
《微博用户情感分析》
《互联网广告预测分析》
《网站日志处理分析》



系列案例教材将于2019年陆续出版发行，教材相关信息，敬请关注网页后续更新！
http://dblab.xmu.edu.cn/post/shixunkecheng/



扫一扫访问大数据实训课程系列案例教材主页


Thank You!