《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12章 数据采集工具的安装和使用

（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

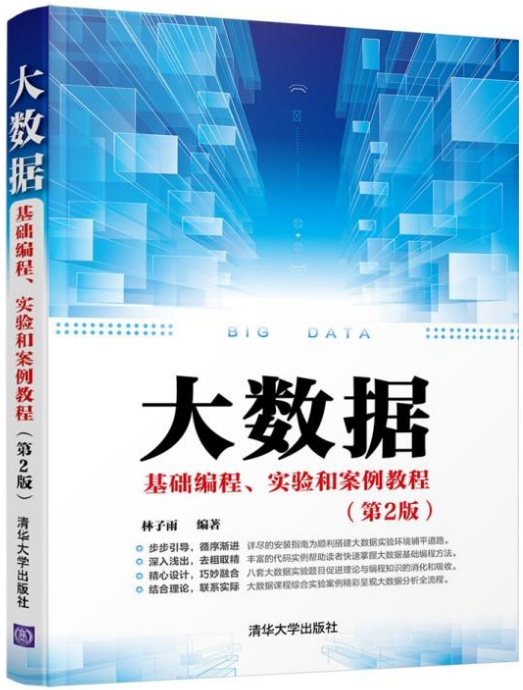
主页：**http://dblab.xmu.edu.cn/linziyu**

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/

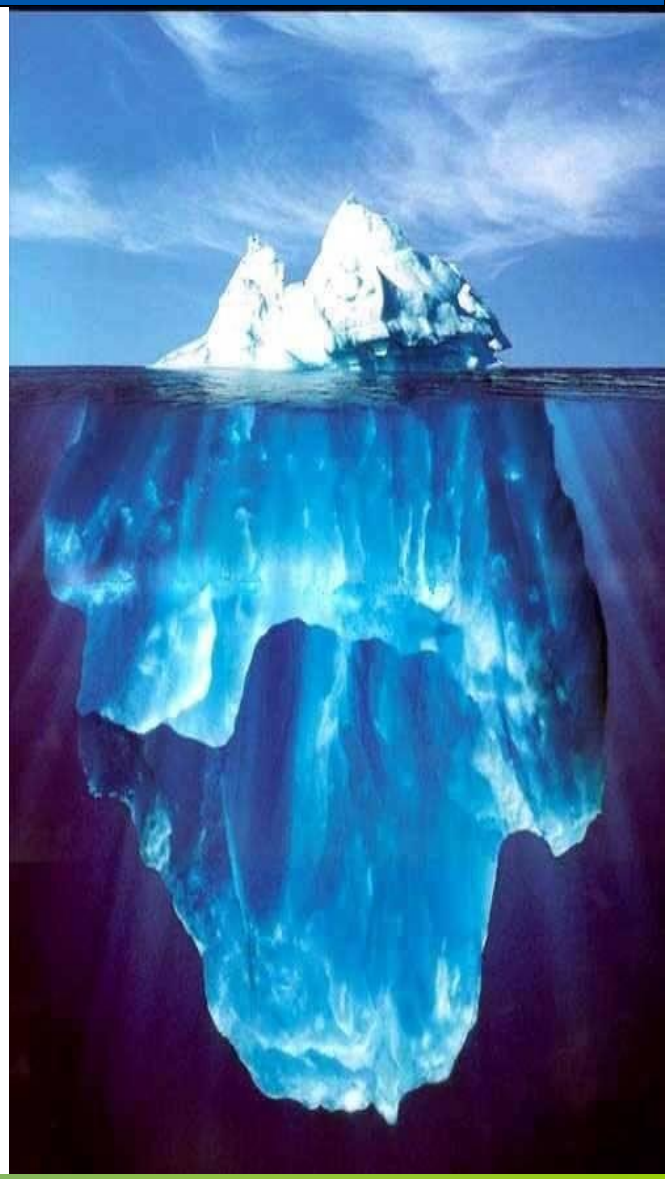
扫一扫访问
教材官网



- 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

12.1 Kafka
12.2 实例：编写Spark程序使用Kafka数据源



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

12.1 Kafka

12.1.1 Kafka相关概念
12.1.2 安装Kafka
12.1.3 一个实例

12.1.1 Kafka相关概念

为了更好地理解和使用 Kafka，这里介绍一下Kafka的相关概念：

•Broker：Kafka集群包含一个或多个服务器，这些服务器被称为Broker；

•Topic：每条发布到Kafka集群的消息都有一个类别，这个类别被称为Topic。物理上不同Topic的消息分开存储，逻辑上一个Topic的消息虽然保存于一个或多个Broker上，但用户只需指定消息的Topic，即可生产或消费数据，而不必关心数据存于何处。

•Partition：是物理上的概念，每个Topic包含一个或多个Partition。

•Producer：负责发布消息到Kafka Broker。

•Consumer：消息消费者，向Kafka Broker读取消息的客户端。

•Consumer Group：每个Consumer属于一个特定的Consumer Group，可为每个Consumer指定group name，若不指定group name则属于默认的group。

12.1.2 安装Kafka

访问Kafka官网下载页面（https://kafka.apache.org/downloads），下载Kafka0.10.2.0的安装包kafka_2.11-0.10.2.0.tgz，此安装包内已经附带Zookeeper，不需要额外安装Zookeeper。

打开一个终端，执行如下命令：

```
$ cd ~/下载
$ sudo tar -zxvf kafka_2.11-0.10.2.0.tgz -C /usr/local
$ cd /usr/local
$ sudo mv kafka_2.11-0.10.2.0/ ./kafka
$ sudo chown -R hadoop ./kafka
```

12.1.3 一个实例

新建一个Linux终端，执行如下命令启动Zookeeper：

```
$ cd /usr/local/kafka
$ ./bin/zookeeper-server-start.sh config/zookeeper.properties
```

新建第二个终端，输入如下命令启动Kafka：

```
$ cd /usr/local/kafka
$ ./bin/kafka-server-start.sh config/server.properties
```

新建第三个个终端，输入如下命令：

```
$ cd /usr/local/kafka
$ ./bin/kafka-topics.sh --create --zookeeper localhost:2181 --
replication-factor 1 --partitions 1 --topic dblab
```

12.1.3 一个实例

可以用list命令列出所有创建的topics，来查看刚才创建的topic是否存在，命令如下：

```
$ cd /usr/local/kafka
$ ./bin/kafka-topics.sh --list --zookeeper localhost:2181
```

可以在结果中查看到，dblab这个topic已经存在。接下来用producer生产一些数据，命令如下：

```
$ cd /usr/local/kafka
$ ./bin/kafka-console-producer.sh --broker-list localhost:9092 --topic dblab
```

该命令执行后，可以在该终端中输入以下信息作为测试：

```
hello hadoop
hello xmu
hadoop world
```

12.1.3 一个实例

然后，再次开启新的第四个终端，输入如下命令使用consumer来接收数据：

```
$ cd /usr/local/kafka
$ ./bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic
dblab --from-beginning
```

执行该命令以后，就可以看到刚才在另外一个终端的producer产生的三条信息“hello hadoop”、“hello xmu”和“hello world”。说明Kafka安装成功。

12.2 实例：编写Spark程序使用Kafka数据源

12.2.1 Kafka准备工作
12.2.2 Spark准备工作
12.2.3 编写Spark程序使用Kafka数据源

12.2.1 Kafka准备工作

1.启动Kafka

首先需要启动Kafka。请登录Linux系统（本教程统一使用hadoop用户登录），打开一个终端，输入下面命令启动Zookeeper服务：

```
$ cd /usr/local/kafka
$ ./bin/zookeeper-server-start.sh config/zookeeper.properties
```

打开第二个终端，然后输入下面命令启动Kafka服务：

```
$ cd /usr/local/kafka
$ ./bin/kafka-server-start.sh config/server.properties
```

12.2.1 Kafka准备工作

2.测试Kafka是否正常工作

请再另外打开第三个终端，然后输入下面命令创建一个自定义名称为“wordsendertest”的topic：

```
$ cd /usr/local/kafka
$ ./bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic wordsendertest
#这个topic叫wordsendertest，2181是Zookeeper默认的端口号，partition是topic里面的分区数，replication-factor是备份的数量，在Kafka集群中使用，这里单机版就不用备份了
#可以用list列出所有创建的topics，来查看上面创建的topic是否存在
$ ./bin/kafka-topics.sh --list --zookeeper localhost:2181
```

12.2.1 Kafka准备工作

下面，用producer来产生一些数据，请在当前终端内继续输入下面命令：

```
$ ./bin/kafka-console-producer.sh --broker-list localhost:9092 --topic wordsendertest
```

上面命令执行后，就可以在当前终端内用键盘输入一些英文单词，比如可以输入：

```
hello hadoop
hello spark
```

打开第四个终端，输入下面命令：

```
$ cd /usr/local/kafka
$./bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic wordsendertest --from-beginning
```

可以看到，屏幕上会显示出如下结果：

```
hello hadoop
hello spark
```

12.2.2 Spark准备工作

1.添加相关jar包

打开一个新的终端，然后启动spark-shell，命令如下：

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

启动成功后，在spark-shell中执行下面import语句：

```
scala> import org.apache.spark.streaming.kafka._
<console>:25: error: object kafka is not a member of package org.apache.spark.streaming
       import org.apache.spark.streaming.kafka._
                                  ^
```

可以看到，马上会报错，因为找不到相关的jar包。所以，需要下载spark-streaming-kafka-0-8_2.11-2.4.0.jar。

12.2.2 Spark准备工作

访问Spark官网（https://mvnrepository.com/artifact/org.apache.spark/spark-streaming-kafka-0-8_2.11/2.4.0），里面有提供spark-streaming-kafka-0-8_2.11-2.4.0.jar文件的下载，其中，2.11表示Scala的版本号，2.4.0表示Spark的版本号。

现在，需要把这个文件复制到Spark目录的lib目录下。请新打开一个终端，输入如下命令：

```
$ cd /usr/local/spark/jars
$ mkdir kafka
$ cd ~/Downloads
$ cp ./spark-streaming-kafka-0-8_2.11-2.4.0.jar /usr/local/spark/jars/kafka
```

12.2.2 Spark准备工作

下面还要继续把Kafka安装目录的lib目录下的所有jar文件复制到“/usr/local/spark/lib/kafka”目录下，请在终端中执行下面命令：

```
$ cd /usr/local/kafka/libs
$ ls
$ cp ./* /usr/local/spark/jars/kafka
```

2.启动spark-shell

然后，执行如下命令启动spark-shell：

```
$ cd /usr/local/spark
$ ./bin/spark-shell --jars /usr/local/spark/jars/*:/usr/local/spark/jars/kafka/*
```

启动成功后，再次执行如下命令：

```
scala> import org.apache.spark.streaming.kafka._
//会显示下面信息
import org.apache.spark.streaming.kafka._
```

12.2.3 编写Spark程序使用Kafka数据源

1.编写生产者（producer）程序

请新打开一个终端，然后，执行如下命令创建代码目录和代码文件：

```
$ cd  /usr/local/spark/mycode
$ mkdir  kafka
$ cd  kafka
$ mkdir  -p  src/main/scala
$ cd  src/main/scala
$ vim  KafkaWordProducer.scala
```

12.2.3 编写Spark程序使用Kafka数据源

```
package org.apache.spark.examples.streaming
import java.util.HashMap
import org.apache.kafka.clients.producer.{KafkaProducer, ProducerConfig, ProducerRecord}
import org.apache.spark.SparkConf
import org.apache.spark.streaming._
import org.apache.spark.streaming.kafka._
object KafkaWordProducer {
  def main(args: Array[String]) {
    if (args.length < 4) {
      System.err.println("Usage: KafkaWordCountProducer <metadataBrokerList> <topic> " +
        "<messagesPerSec> <wordsPerMessage>")
      System.exit(1)
    }
    val Array(brokers, topic, messagesPerSec, wordsPerMessage) = args
    // Zookeeper connection properties
    val props = new HashMap[String, Object]()
    props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, brokers)
    props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG,
      "org.apache.kafka.common.serialization.StringSerializer")
    props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG,
      "org.apache.kafka.common.serialization.StringSerializer")
    val producer = new KafkaProducer[String, String](props)
   // Send some messages
    while(true) {
      (1 to messagesPerSec.toInt).foreach { messageNum =>
        val str = (1 to wordsPerMessage.toInt).map(x => scala.util.Random.nextInt(10).toString)
          .mkString(" ")
                    print(str)
                    println()
        val message = new ProducerRecord[String, String](topic, null, str)
        producer.send(message)
      }
     Thread.sleep(1000)
    }
  }
}
```

12.2.3 编写Spark程序使用Kafka数据源

2.编写消费者（consumer）程序

```
package org.apache.spark.examples.streaming
import org.apache.spark._
import org.apache.spark.SparkConf
import org.apache.spark.streaming._
import org.apache.spark.streaming.kafka._
import org.apache.spark.streaming.StreamingContext._
import org.apache.spark.streaming.kafka.KafkaUtils
object KafkaWordCount{
def main(args:Array[String]){
StreamingExamples.setStreamingLogLevels()
val sc = new SparkConf().setAppName("KafkaWordCount").setMaster("local[2]")
val ssc = new StreamingContext(sc,Seconds(10))
ssc.checkpoint("file:///usr/local/spark/mycode/kafka/checkpoint") //设置检查点，如果存放在HDFS上面，则写成类似ssc.checkpoint("/user/hadoop/checkpoint")这种形式，但是，要启动Hadoop
val zkQuorum = "localhost:2181" //Zookeeper服务器地址
val group = "1"  //Topic所在的group，可以设置为自己想要的名称，比如不用1，而是val group = "test-consumer-group"
val topics = "wordsender"  //topics的名称
val numThreads = 1  //每个topic的分区数
val topicMap =topics.split(",").map((_,numThreads.toInt)).toMap
val lineMap = KafkaUtils.createStream(ssc,zkQuorum,group,topicMap)
val lines = lineMap.map(_._2)
val words = lines.flatMap(_.split(" "))
val pair = words.map(x => (x,1))
val wordCounts = pair.reduceByKeyAndWindow(_ + _,_ - _,Minutes(2),Seconds(10),2) //这行代码的含义在下一节的窗口转换操作中会有介绍
wordCounts.print
ssc.start
ssc.awaitTermination
}
}
```

12.2.3 编写Spark程序使用Kafka数据源

3.编写日志格式设置程序

```
package org.apache.spark.examples.streaming
import org.apache.spark.internal.Logging
import org.apache.log4j.{Level, Logger}
/** Utility functions for Spark Streaming examples. */
object StreamingExamples extends Logging {
  /** Set reasonable logging levels for streaming if the user has not
configured log4j. */
  def setStreamingLogLevels() {
    val log4jInitialized =
Logger.getRootLogger.getAllAppenders.hasMoreElements
    if (!log4jInitialized) {
      // We first log something to initialize Spark's default logging,
then we override the
      // logging level.
      logInfo("Setting log level to [WARN] for streaming example." +
        " To override add a custom log4j.properties to the classpath.")
      Logger.getRootLogger.setLevel(Level.WARN)
    }
  }
}
```

12.2.3 编写Spark程序使用Kafka数据源

4.编译打包程序

请执行下面命令新建一个simple.sbt文件：

```
$ cd  /usr/local/spark/mycode/kafka/
$ vim  simple.sbt
```

在simple.sbt中输入以下代码：

```
name := "Simple Project"
version := "1.0"
scalaVersion := "2.11.12"
libraryDependencies += "org.apache.spark" %% "spark-core" % "2.4.0"
libraryDependencies += "org.apache.spark" % "spark-streaming_2.11"
% "2.4.0"
libraryDependencies += "org.apache.spark" % "spark-streaming-kafka-
0-8_2.11" % "2.4.0" exclude("net.jpountz.lz4", "lz4")
```

12.2.3 编写Spark程序使用Kafka数据源

然后执行下面命令，进行编译打包：

```
$ cd  /usr/local/spark/mycode/kafka/
$ /usr/local/sbt/sbt  package
```

5.运行程序

首先，请启动Hadoop，因为如果前面KafkaWordCount.scala代码文件中采用了ssc.checkpoint("/user/hadoop/checkpoint")这种形式，这时的检查点是被写入HDFS，因此需要启动Hadoop。启动Hadoop的命令如下：

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

启动Hadoop成功以后，就可以测试刚才生成的词频统计程序了。

12.2.3 编写Spark程序使用Kafka数据源

请新打开一个终端，执行如下命令，运行“KafkaWordProducer”程序，生成一些单词（是一堆整数形式的单词）：

```
$ cd  /usr/local/spark
$ /usr/local/spark/bin/spark-submit  \
> --driver-class-path /usr/local/spark/jars/*:/usr/local/spark/jars/kafka/*  \
> --class "org.apache.spark.examples.streaming.KafkaWordProducer"   \
> /usr/local/spark/mycode/kafka/target/scala-2.11/simple-project_2.11-1.0.jar  \
> localhost:9092  wordsender  3  5
```

执行上面命令后，屏幕上会不断滚动出现类似如下的新单词：

```
3 3 6 3 4
9 4 0 8 1
0 3 3 9 3
0 8 4 0 9
8 7 2 9 5
……
```

12.2.3 编写Spark程序使用Kafka数据源

不要关闭这个终端窗口，让它一直不断发送单词。然后，请新打开一个终端，执行下面命令，运行KafkaWordCount程序，执行词频统计：

```
$ cd  /usr/local/spark
$/usr/local/spark/bin/spark-submit  \
> --driver-class-path /usr/local/spark/jars/*:/usr/local/spark/jars/kafka/*  \
> --class "org.apache.spark.examples.streaming.KafkaWordCount"  \
> /usr/local/spark/mycode/kafka/target/scala-2.11/simple-project_2.11-1.0.jar
```

运行上面命令以后，就启动了词频统计功能，屏幕上就会显示如下类似信息：

```
-------------------------------------------
Time: 1488156500000 ms
-------------------------------------------
(4,5)
(8,12)
(6,14)
(0,19)
……
```

12.3 本章小结

数据采集工具Kafka经常用在Hadoop和Spark生态系统中，用来进行日志信息的实时采集。本章介绍了数据采集工具Kafka的安装和使用方法，并给出了几个实例演示工具的具体用法。最后详细介绍了如何编写Spark Streaming应用程序来“消费”Kafka的数据源，通过这个实例，可以从总体上了解Spark和Kafka等工具之间的组合使用方法。

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

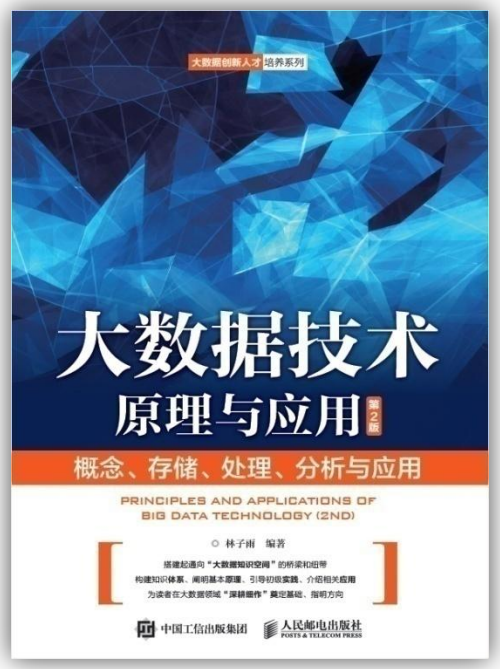
《大数据技术原理与应用——概念、存储、处理、分析与应用（第2版）》，由厦门大学计算机科学系林子雨博士编著，是国内高校第一本系统介绍大数据知识的专业教材。人民邮电出版社 ISBN:978-7-115-44330-4 定价：49.80元

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

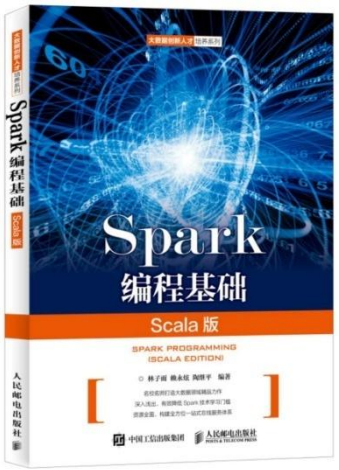
厦门大学林子雨编著

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）》

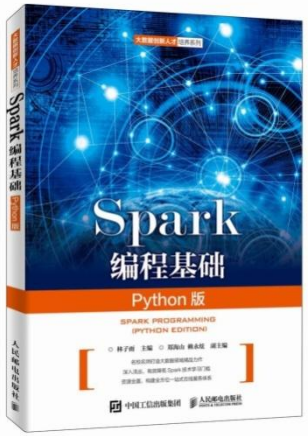
厦门大学 林子雨，赖永炫，陶继平 编著

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/

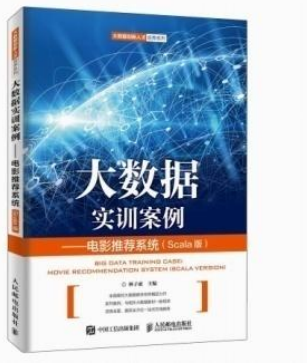
扫一扫访问平台主页

扫一扫观看3分钟FLASH动画宣传片

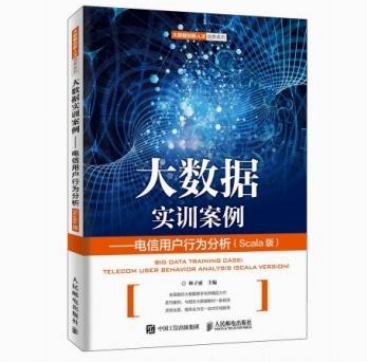
附录K：高校大数据实训课程系列案例教材

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!