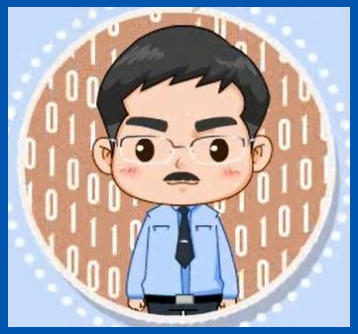


《大数据基础编程、实验和案例教程（第2版）》

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13章 大数据课程综合实验案例

（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

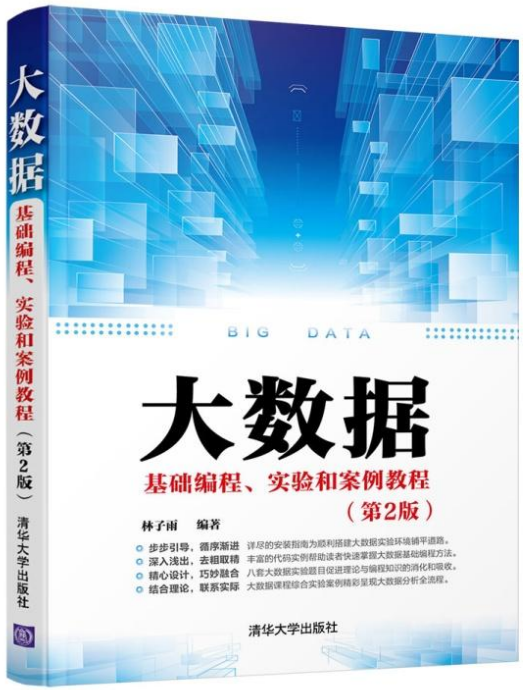
主页：http://dblab.xmu.edu.cn/linziyu

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

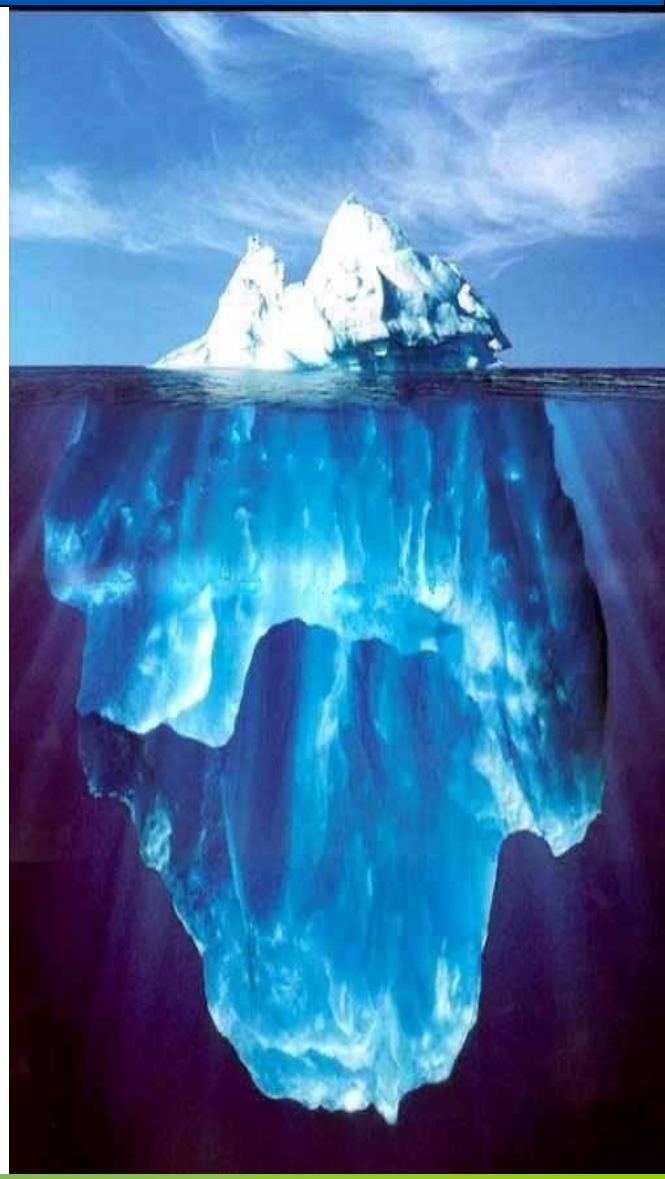
林子雨编著《大数据基础编程、实验和案例教程（第2版）》
清华大学出版社 ISBN:978-7-302-55977-1 定价：69元，2020年10月第2版
教材官网： http://dblab.xmu.edu.cn/post/bigdatapractice2/

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



13.1 案例简介

13.1.1 案例目的

- 熟悉Linux系统、MySQL、Hadoop、HBase、Hive、R、Eclipse等系统和软件的安装和使用；
- 了解大数据处理的基本流程；
- 熟悉数据预处理方法；
- 熟悉在不同类型数据库之间进行数据相互导入导出；
- 熟悉使用R语言进行可视化分析；
- 熟悉使用Eclipse编写Java程序操作HBase、Hive和MySQL。

13.1.2适用对象

- 高校（高职）教师、学生
- 大数据学习者

13.1.3 时间安排

本案例可以作为大数据入门级课程结束后的“大作业”，或者可以作为学生暑期或寒假大数据实习实践基础案例，建议在一周左右完成本案例。

13.1.4 预备知识

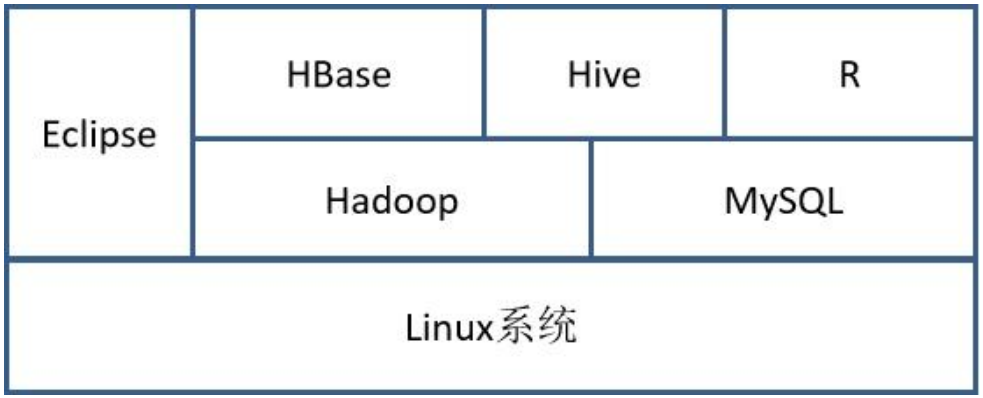
需要案例使用者，已经学习过大数据相关课程（比如入门级课程《大数据技术原理与应用》），了解大数据相关技术的基本概念与原理，了解Windows操作系统、Linux操作系统、大数据处理架构Hadoop的关键技术及其基本原理、列族数据库HBase概念及其原理、数据仓库概念与原理、关系型数据库概念与原理、R语言概念与应用等。

不过，由于本案例提供了全部操作细节，包括每个命令和运行结果，所以，即使没有相关背景知识，也可以按照操作说明顺利完成全部实验。

13.1.5 硬件要求

本案例可以在单机上完成，也可以在集群环境下完成。单机上完成本案例实验时，建议计算机硬件配置为：50GB以上硬盘，8GB以上内存。

13.1.6 软件工具

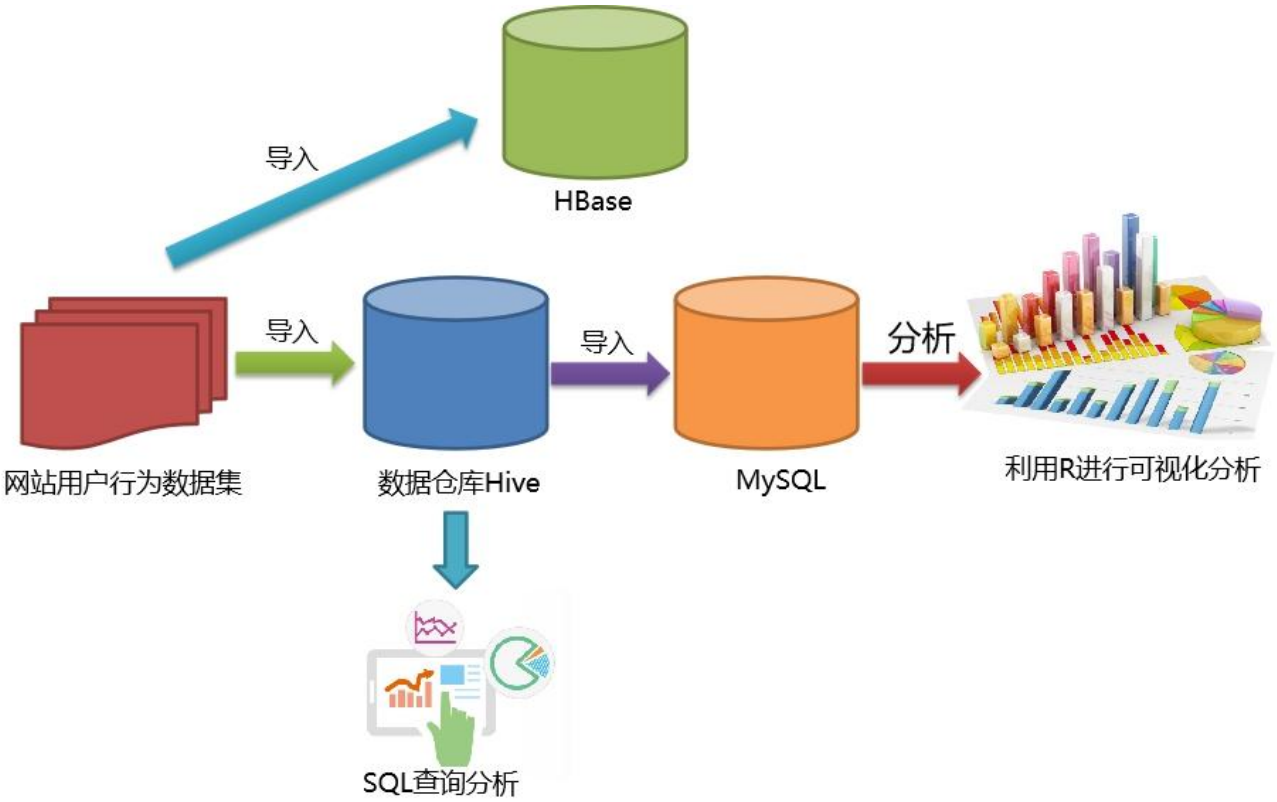


相关软件的版本建议如下：
Linux: Ubuntu16.04（或18.04）
MySQL: 5.7.29
Hadoop: 3.1.3
HBase:2.2.2
Hive:3.1.2
R:3.2.3
Eclipse:3.8

13.1.7 数据集

网站用户购物行为数据集，包括2000万条记录。

13.1.8 案例任务


导入
HBase
导入
导入
分析
网站用户行为数据集
数据仓库Hive
MySQL
利用R进行可视化分析
SQL查询分析

13.2 实验环境搭建

为了顺利完成本案例各项实验，需要完成以下系统和软件的安装：

- 安装Linux系统：如果未安装，请参照“第2章 Linux系统的安装和使用” 的相关内容，完成Linux系统的安装。
- 安装Hadoop：如果未安装，请参照“第3章 Hadoop的安装和使用” 的相关内容，完成Hadoop的安装。
- 安装MySQL：如果未安装，请参照“附录B：Linux系统中的MySQL安装及常用操作”的相关内容，完成MySQL的安装。
- 安装HBase：如果未安装，请参照“第5章 HBase的安装和基础编程”中的相关内容，完成HBase的安装。
- 安装Hive：如果未安装，请参照“第8章 数据仓库Hive安装和使用”的相关内容，完成Hive的安装。
- 安装Eclipse：如果未安装，请参照“第2章 Linux系统的安装和使用”的相关内容，在Linux系统的安装Eclipse。

13.3 实验步骤概述

本案例共包含4个实验步骤：
步骤一：本地数据集上传到数据仓库Hive
步骤二：Hive数据分析
步骤三：Hive、MySQL、HBase数据互导
步骤四：利用R进行数据可视化分析

13.4 步骤一：本地数据集上传到数据仓库Hive

13.4.1 实验数据集的下载
13.4.2 数据集的预处理
13.4.3 导入数据库

13.4.1 实验数据集的下载

本案例采用的数据集为user.zip，包含了一个大规模数据集raw_user.csv（包含2000万条记录），和一个小数据集small_user.csv（只包含30万条记录）。小数据集small_user.csv是从大规模数据集raw_user.csv中抽取的一小部分数据。之所以抽取出一少部分记录单独构成一个小数据集，是因为在第一遍跑通整个实验流程时，会遇到各种错误、各种问题，先用小数据集测试，可以大量节约程序运行时间。等到第一次完整实验流程都顺利跑通以后，可以最后用大规模数据集进行最后的测试。

把数据集user.zip文件下载到Linux系统的“/home/hadoop/下载/”目录下面

13.4.1 实验数据集的下载

下面需要把user.zip进行解压缩，需要首先建立一个用于运行本案例的目录bigdatacase，请执行以下命令：

```
$ cd /usr/local
$ ls
$ sudo mkdir bigdatacase
#这里会提示你输入当前用户（本教程是hadoop用户名）的密码
#下面给hadoop用户赋予针对bigdatacase目录的各种操作权限
$ sudo chown -R hadoop:hadoop ./bigdatacase
$ cd bigdatacase
#下面创建一个dataset目录，用于保存数据集
$ mkdir dataset
#下面就可以解压缩user.zip文件
$ cd ~  //表示进入hadoop用户的目录
$ cd 下载
$ ls
$ unzip user.zip -d /usr/local/bigdatacase/dataset
$ cd /usr/local/bigdatacase/dataset
$ ls
```

现在就可以看到在dataset目录下有两个文件：raw_user.csv和small_user.csv。我们执行下面命令取出前面5条记录看一下：

```
$ head -5 raw_user.csv
```

13.4.2 数据集的预处理

1. 删除文件第一行记录（即字段名称）

```
$ cd /usr/local/bigdatacase/dataset
#下面删除raw_user中的第1行
$ sed -i '1d' raw_user.csv
#上面的1d表示删除第1行，同理，3d表示删除第3行，nd表示删除第n行
#下面删除small_user中的第1行
$ sed -i '1d' small_user.csv
#下面再用head命令去查看文件的前5行记录，就看不到字段名称这一行了
$ head -5 raw_user.csv
$ head -5 small_user.csv
```

13.4.2 数据集的预处理

2.对字段进行预处理

下面要建一个脚本文件pre_deal.sh，请把这个脚本文件放在dataset目录下，和数据集small_user.csv放在同一个目录下：

```
$ cd /usr/local/bigdatacase/dataset
$ vim pre_deal.sh
```

13.4.2 数据集的预处理

```
#!/bin/bash
#下面设置输入文件，把用户执行pre_deal.sh命令时提供的第一个参数作为输入文件名称
infile=$1
#下面设置输出文件，把用户执行pre_deal.sh命令时提供的第二个参数作为输出文件名称
outfile=$2
#注意，最后的$infile> $outfile必须跟在}'这两个字符的后面
awk -F "," 'BEGIN{
srand();
        id=0;
        Province[0]="山东";Province[1]="山西";Province[2]="河南";Province[3]="河北";Province[4]="陕西";Province[5]="
内蒙古";Province[6]="上海市";
        Province[7]="北京市";Province[8]="重庆市";Province[9]="天津市";Province[10]="福建";Province[11]="广东
";Province[12]="广西";Province[13]="云南";
        Province[14]="浙江";Province[15]="贵州";Province[16]="新疆";Province[17]="西藏";Province[18]="江西
";Province[19]="湖南";Province[20]="湖北";
        Province[21]="黑龙江";Province[22]="吉林";Province[23]="辽宁"; Province[24]="江苏";Province[25]="甘肃
";Province[26]="青海";Province[27]="四川";
        Province[28]="安徽"; Province[29]="宁夏";Province[30]="海南";Province[31]="香港";Province[32]="澳门
";Province[33]="台湾";
    }
    {
        id=id+1;
        value=int(rand()*34);
        print id"\t"$1"\t"$2"\t"$3"\t"$5"\t"substr($6,1,10)"\t"Province[value]
    }' $infile> $outfile
```

下面就可以执行pre_deal.sh脚本文件，来对small_user.csv进行数据预处理，命令如下：

```
$ cd /usr/local/bigdatacase/dataset
$ bash ./pre_deal.sh small_user.csv user_table.txt
```

可以使用head命令查看前10行数据：

```
$ head -10 user_table.txt
```

13.4.3 导入数据库

1.启动HDFS

执行下面命令启动Hadoop：

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

然后，执行jps命令看一下当前运行的进程：

```
3800 Jps
3261 DataNode
3134 NameNode
3471 SecondaryNameNode
```

13.4.3 导入数据库

2. 把user_table.txt上传到HDFS中

首先，需要在HDFS的根目录下面创建一个新的目录bigdatacase，并在这个目录下创建一个子目录dataset，具体命令如下：

```
$ cd /usr/local/hadoop
$ ./bin/hdfs dfs -mkdir -p /bigdatacase/dataset
```

然后，把Linux本地文件系统中的user_table.txt上传到分布式文件系统HDFS的“/bigdatacase/dataset”目录下，命令如下：

```
$ cd /usr/local/hadoop
$ ./bin/hdfs dfs -put /usr/local/bigdatacase/dataset/user_table.txt /bigdatacase/dataset
```

现在可以查看一下HDFS中的user_table.txt的前10条记录，命令如下：

```
$ cd /usr/local/hadoop
$ ./bin/hdfs dfs -cat /bigdatacase/dataset/user_table.txt | head -10
```

13.4.3 导入数据库

3. 在Hive上创建数据库

首先启动MySQL数据库，可以在终端中输入如下命令：

```
$ service mysql start  #可以在Linux的任何目录下执行该命令
```

在这个新的终端中执行下面命令进入Hive：

```
$ cd /usr/local/hive
$ ./bin/hive   #启动Hive
```

需要在Hive中创建一个数据库dblab，命令如下：

```
hive> create database dblab;
hive> use dblab;
```

13.4.3 导入数据库

4.创建外部表

在hive命令提示符下输入如下命令：

```
hive>  CREATE EXTERNAL TABLE dblab.bigdata_user(id INT,uid
STRING,item_id STRING,behavior_type INT,item_category
STRING,visit_date DATE,province STRING) COMMENT 'Welcome to
xmudblab!' ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE LOCATION '/bigdatacase/dataset';
```

13.4.3 导入数据库

5.查询数据

在“hive>”命令提示符状态下执行下面命令查看表的信息：

```
hive> use dblab; //使用dblab数据库
hive> show tables; //显示数据库中所有表
hive> show create table bigdata_user; //查看bigdata_user表的各种属性；
```

还可以执行下面命令查看表的简单结构：

```
hive> desc bigdata_user;
```

现在可以使用下面命令查询相关数据：

```
hive> select * from bigdata_user limit 10;
hive> select behavior_type from bigdata_user limit 10;
```

13.5 步骤二：Hive数据分析

13.5.1 简单查询分析
13.5.2 查询条数统计分析
13.5.3 关键字条件查询分析
13.5.4 根据用户行为分析
13.5.5 用户实时查询分析

13.5.1 简单查询分析

首先执行一条简单的指令：

```
hive> select behavior_type from bigdata_user limit 10; #查看前10位用户对商品的行为
```

如果要查出每位用户购买商品时的多种信息，输出语句格式如下：
select 列1，列2，….，列n from 表名；
比如查询前20位用户购买商品时的时间和商品的种类，语句如下：

```
hive> select visit_date, item_category from bigdata_user limit 20;
```

有时在表中查询可以利用嵌套语句，如果列名太复杂可以设置该列的别名，以简化操作的难度，举例如下：

```
hive> select e.bh, e.it from (select behavior_type as bh,
item_category as it from bigdata_user) as e  limit 20;
```

13.5.2 查询条数统计分析

1.用聚合函数count()计算出表内有多少行数据

```
hive> select count(*) from bigdata_user;
```

2.在函数内部加上distinct，查出uid不重复的数据有多少条

```
hive> select count(distinct uid) from bigdata_user;
```

3.查询不重复的数据有多少条(为了排除客户刷单情况)

```
hive>select count(*) from (select
uid,item_id,behavior_type,item_category,visit_date,province from
bigdata_user group by
uid,item_id,behavior_type,item_category,visit_date,province
having count(*)=1)a;
```

13.5.3 关键字条件查询分析

1.以关键字的存在区间为条件的查询

(1)查询2014年12月10日到2014年12月13日有多少人浏览了商品。

```
hive> select count(*) from bigdata_user where behavior_type='1'
and visit_date<'2014-12-13' and visit_date>'2014-12-10';
```

（2）以月的第n天为统计单位，依次显示第n天网站卖出去的商品的个数

```
hive> select count(distinct uid), day(visit_date) from bigdata_user
where behavior_type='4' group by day(visit_date);
```

2.关键字赋予给定值为条件，对其他数据进行分析

取给定时间和给定地点，求当天发出到该地点的货物的数量。

```
hive> select count(*) from bigdata_user where province='江西' and
visit_date='2014-12-12' and behavior_type='4';
```

13.5.4 根据用户行为分析

1. 查询一件商品在某天的购买比例或浏览比例

```
hive> select count(*) from bigdata_user where visit_date='2014-12-11'and behavior_type='4';#查询有多少用户在2014-12-11购买了商品
```

```
hive> select count(*) from bigdata_user where visit_date ='2014-12-11';#查询有多少用户在2014-12-11点击了该店
```

2.查询某个用户在某一天点击网站占该天所有点击行为的比例（点击行为包括浏览、加入购物车、收藏、购买）

```
hive> select count(*) from bigdata_user where uid=10001082 and visit_date='2014-12-12';#查询用户10001082在2014-12-12点击网站的次数
```

```
hive> select count(*) from bigdata_user where visit_date='2014-12-12';#查询所有用户在这一天点击该网站的次数
```

13.5.4 根据用户行为分析

3.给定购买商品的数量范围，查询某一天在该网站的购买该数量商品的用户id

```
hive> select uid from bigdata_user where behavior_type='4' and
visit_date='2014-12-12' group by uid having count(behavior_type='4')>5;#
查询某一天在该网站购买商品超过5次的用户id
```

13.5.5 用户实时查询分析

查询某个地区的用户当天浏览网站的次数，语句如下：

```
hive> create table scan(province STRING,scan INT) COMMENT 'This is the search of bigdataday' ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS TEXTFILE;#创建新的数据表进行存储
hive> insert overwrite table scan select province,count(behavior_type) from bigdata_user where behavior_type='1' group by province;#导入数据
hive> select * from scan;#显示结果
```

13.6 步骤三：Hive、MySQL、HBase数据互导

13.6.1 Hive预操作
13.6.2 使用Java API将数据从Hive导入MySQL
13.6.3使用HBase Java API把数据从本地导入到HBase中

13.6.1 Hive预操作

1. 创建临时表user_action

```
hive> create table dblab.user_action(id STRING,uid STRING, item_id STRING, behavior_type STRING, item_category STRING, visit_date DATE, province STRING) COMMENT 'Welcome to XMU dblab! ' ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS TEXTFILE;
```

现在可以新建一个终端，执行命令查看一下，确认这个数据文件在HDFS中确实已经被创建，请在新建的终端中执行下面命令：

```
$ cd /usr/local/hadoop
$./bin/hdfs dfs -ls /user/hive/warehouse/dblab.db/
```

13.6.1 Hive预操作

2.将bigdata_user表中的数据插入到user_action

下面把dblab.bigdata_user数据插入到dblab.user_action表中，命令如下：

```
hive> INSERT OVERWRITE TABLE dblab.user_action select * from dblab.bigdata_user;
```

然后执行下面命令查询上面的插入命令是否成功执行：

```
hive> select * from user_action limit 10;
```

13.6.2 使用Java API将数据从Hive导入MySQL

1.将前面生成的临时表数据从Hive导入到 MySQL中

(1)登录 MySQL

请在Linux系统中新建一个终端，执行下面命令：

```
$ mysql -u root -p
```

(2)创建数据库

```
mysql> show databases; #显示所有数据库
mysql> create database dblab; #创建dblab数据库
mysql> use dblab; #使用数据库
```

13.6.2 使用Java API将数据从Hive导入MySQL

注意：请使用下面命令查看数据库的编码：

```
mysql>show variables like "char%";
```

```
+--------------------------+----------------------------+
| Variable_name            | Value                      |
+--------------------------+----------------------------+
| character_set_client     | utf8                       |
| character_set_connection | utf8                       |
| character_set_database   | latin1                     |
| character_set_filesystem | binary                     |
| character_set_results    | utf8                       |
| character_set_server     | latin1                     |
| character_set_system     | utf8                       |
| character_sets_dir       | /usr/share/mysql/charsets/ |
+--------------------------+----------------------------+
8 rows in set (0.00 sec)
```

13.6.2 使用Java API将数据从Hive导入MySQL

修改了编码格式后，再次执行“show variables like ”char%“”命令会得到如图所示的结果。

```
+--------------------------+----------------------------+
| Variable_name            | Value                      |
+--------------------------+----------------------------+
| character_set_client     | utf8                       |
| character_set_connection | utf8                       |
| character_set_database   | utf8                       |
| character_set_filesystem | binary                     |
| character_set_results    | utf8                       |
| character_set_server     | utf8                       |
| character_set_system     | utf8                       |
| character_sets_dir       | /usr/share/mysql/charsets/ |
+--------------------------+----------------------------+
8 rows in set (0.00 sec)
```

13.6.2 使用Java API将数据从Hive导入MySQL

(3)创建表

下面在MySQL的数据库dblab中创建一个新表user_action，并设置其编码为utf-8：

```
mysql> CREATE TABLE `dblab`.`user_action` (`id` varchar(50),`uid`
varchar(50),`item_id` varchar(50),`behavior_type`
varchar(10),`item_category` varchar(50), `visit_date`
DATE,`province` varchar(20)) ENGINE=InnoDB DEFAULT
CHARSET=utf8;
```

创建成功后，输入下面命令退出MySQL：

```
mysql> exit
```

13.6.2 使用Java API将数据从Hive导入MySQL

(4)导入数据

通过JDBC连接Hive和MySQL，将数据从Hive导入MySQL。通过JDBC连接Hive，需要通过Hive的thrift服务实现跨语言访问Hive，实现thrift服务需要开启hiveserver2。

首先，在Hadoop的配置文件core-site.xml中添加以下配置信息：

```
<property>
      <name>hadoop.proxyuser.hadoop.hosts</name>
      <value>*</value>
</property>
<property>

<name>hadoop.proxyuser.hadoop.groups</name>
      <value>*</value>
</property>
```

13.6.2 使用Java API将数据从Hive导入MySQL

然后，开启Hadoop以后，在目录“/usr/local/hive”下，执行以下命令开启hiveserver2，并且设置默认端口为10000。

```
$ cd /usr/local/hive
$ ./bin/hive --service hiveserver2 -hiveconf hive.server2.thrift.port=10000
```

启动时，当屏幕上出现“Hive Session ID = 6bd1726e-37c5-41fc-93ea-ef7e176b24f2”信息时，会停留较长的时间，需要出现几个“Hive Session ID=...”以后，Hive才会真正启动。启动成功以后，会出现如图所示信息。

```
Hive Session ID = 6bd1726e-37c5-41fc-93ea-ef7e176b24f2
Hive Session ID = 2470f88b-5f51-4cb8-8b92-d1284f1faa93
Hive Session ID = b9c62916-3395-402f-9017-32eb81f025e0
OK
```

13.6.2 使用Java API将数据从Hive导入MySQL

启动结束后，使用如下命令查看10000号端口是否已经被占用：

```
$sudo netstat -anp|grep 10000
```

如果显示10000号端口已经被占用（如图所示），则启动成功。

```
hadoop@ubuntu:/usr/local/hadoop$ sudo netstat -anp|grep
10000
[sudo] password for hadoop:
tcp6       0      0 :::10000                :::*
            LISTEN      5238/java
tcp6       0      0 127.0.0.1:10000         127.0.0.1:45
572         ESTABLISHED 5238/java
tcp6       0      0 127.0.0.1:45572         127.0.0.1:10
000         ESTABLISHED 5607/java
```

13.6.2 使用Java API将数据从Hive导入MySQL

启动Eclipse，建立Java工程，通过Build Path添加“/usr/local/hadoop/share/Hadoop/common/lib”下的所有jar包，并且添加“/usr/local/hive/lib”下的所有jar包。然后，编写Java程序HivetoMySQL.java，把数据从Hive加载到MySQL中，HivetoMySQL.java的具体代码如下：

```
import java.sql.*;
import java.sql.SQLException;

public class HivetoMySQL {
    private static String driverName = "org.apache.hive.jdbc.HiveDriver";
    private static String driverName_mysql = "com.mysql.jdbc.Driver";
    public static void main(String[] args) throws SQLException {
        try {
            Class.forName(driverName);
        }catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            System.exit(1);
        }
```

13.6.2 使用Java API将数据从Hive导入MySQL

```
 Connection con1 =
DriverManager.getConnection("jdbc:hive2://localhost:10000/default",
"hive", "hive");//后两个参数是用户名密码

        if(con1 == null)
            System.out.println("连接失败");
        else {
            Statement stmt = con1.createStatement();
            String sql = "select * from dblab.user_action";
            System.out.println("Running: " + sql);
            ResultSet res = stmt.executeQuery(sql);
```

13.6.2 使用Java API将数据从Hive导入MySQL

```
            //InsertToMysql
            try {
                Class.forName(driverName_mysql);
                Connection con2 = DriverManager.getConnection("jdbc:mysql://localhost:3306/dblab","root","root");
                String sql2 = "insert into user_action(id,uid,item_id,behavior_type,item_category,visit_date,province)
values (?,?,?,?,?,?,?)";
                PreparedStatement ps = con2.prepareStatement(sql2);
                while (res.next()) {
                    ps.setString(1,res.getString(1));
                    ps.setString(2,res.getString(2));
                    ps.setString(3,res.getString(3));
                    ps.setString(4,res.getString(4));
                    ps.setString(5,res.getString(5));
                    ps.setDate(6,res.getDate(6));
                    ps.setString(7,res.getString(7));
                    ps.executeUpdate();
                }
                ps.close();
                con2.close();
                res.close();
                stmt.close();
            } catch (ClassNotFoundException e) {
                e.printStackTrace();
            }
        }
        con1.close();
    }
}
```

13.6.2 使用Java API将数据从Hive导入MySQL

上面程序执行以后，在MySQL中执行select count(*) from user_action; 如果输出如图所示信息，则表示导入成功。

```
mysql> select count(*) from user_action;
+----------+
| count(*) |
+----------+
|   300000 |
+----------+
1 row in set (0.10 sec)
```

13.6.2 使用Java API将数据从Hive导入MySQL

2.查看MySQL中user_action表数据

下面需要再次启动MySQL，进入“mysql>”命令提示符状态：

```
$ mysql -u root -p
```

然后执行下面命令查询user_action表中的数据：

```
mysql> use dblab;
mysql> select * from user_action limit 10;
```

13.6.3使用HBase Java API把数据从本地导入到HBase中

1.启动Hadoop集群、HBase服务

请首先确保启动了Hadoop集群和HBase服务。如果还没有启动，请在Linux系统中打开一个终端。首先，按照下面命令启动Hadoop：

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

然后，按照下面命令启动HBase：

```
$ cd /usr/local/hbase
$ ./bin/start-hbase.sh
```

13.6.3使用HBase Java API把数据从本地导入到HBase中

2.数据准备

首先，请将之前的user_action数据从HDFS复制到Linux系统的本地文件系统中，命令如下：

```
$ cd /usr/local/bigdatacase/dataset
$ /usr/local/hadoop/bin/hdfs dfs -get
/user/hive/warehouse/dblab.db/user_action .
 #将HDFS上的user_action数据复制到本地当前目录，注意'.'表示当前目录
$ cat ./user_action/* | head -10   #查看前10行数据
$ cat ./user_action/00000* > user_action.output #将00000*文件复制一份
重命名为user_action.output，*表示通配符
$ head user_action.output  #查看user_action.output前10行
```

3.编写数据导入程序

这里采用Eclipse编写Java程序实现HBase数据导入功能，具体代码如下：

```
import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.List;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.*;
import org.apache.hadoop.hbase.client.*;
import org.apache.hadoop.hbase.util.Bytes;
public class ImportHBase extends Thread {
    public Configuration config;
    public Connection conn;
    public Table table;
    public Admin admin;
    public ImportHBase() {
        config = HBaseConfiguration.create();
//      config.set("hbase.master", "master:60000");
//      config.set("hbase.zookeeper.quorum", "master");
        try {
                        conn = ConnectionFactory.createConnection(config);
            admin = conn.getAdmin();
            table = conn.getTable(TableName.valueOf("user_action"));
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

13.6.3使用HBase Java API把数据从本地导入到HBase中

```
public static void main(String[] args) throws Exception {
    if (args.length == 0) {      //第一个参数是该jar所使用的类，第二个参数是数据集所存放的路径
        throw new Exception("You must set input path!");
    }
    String fileName = args[args.length-1];  //输入的文件路径是最后一个参数
    ImportHBase test = new ImportHBase();
    test.importLocalFileToHBase(fileName);
}
public void importLocalFileToHBase(String fileName) {
    long st = System.currentTimeMillis();
    BufferedReader br = null;
    try {
        br = new BufferedReader(new InputStreamReader(new FileInputStream(
                fileName)));
        String line = null;
        int count = 0;
        while ((line = br.readLine()) != null) {
            count++;
            put(line);
            if (count % 10000 == 0)
                System.out.println(count);
        }
    } catch (IOException e) {
        e.printStackTrace();
```

13.6.3使用HBase Java API把数据从本地导入到HBase中

```
} finally {
        if (br != null) {
            try {
                br.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        try {
            table.close(); // must close the client
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    long en2 = System.currentTimeMillis();
    System.out.println("Total Time: " + (en2 - st) + " ms");
  }
  @SuppressWarnings("deprecation")
  public void put(String line) throws IOException {
    String[] arr = line.split("\t", -1);
    String[] column = {"id","uid","item_id","behavior_type","item_category","date","province"};
```

13.6.3使用HBase Java API把数据从本地导入到HBase中

```
        if (arr.length == 7) {
            Put put = new Put(Bytes.toBytes(arr[0]));// rowkey
            for(int i=1;i<arr.length;i++){
                put.addColumn(Bytes.toBytes("f1"),
Bytes.toBytes(column[i]),Bytes.toBytes(arr[i]));
            }
            table.put(put); // put to server
        }
    }
```

13.6.3使用HBase Java API把数据从本地导入到HBase中

```
public void get(String rowkey, String columnFamily, String column,
        int versions) throws IOException {
    long st = System.currentTimeMillis();
    Get get = new Get(Bytes.toBytes(rowkey));
    get.addColumn(Bytes.toBytes(columnFamily), Bytes.toBytes(column));
    Scan scanner = new Scan(get);
    scanner.readVersions(versions);
    ResultScanner rsScanner = table.getScanner(scanner);
    for (Result result : rsScanner) {
        final List<Cell> list = result.listCells();
        for (final Cell kv : list) {
            System.out.println(Bytes.toStringBinary(kv.getValueArray()) + "\t"
                    + kv.getTimestamp()); // mid + time
        }
    }
    rsScanner.close();
    long en2 = System.currentTimeMillis();
    System.out.println("Total Time: " + (en2 - st) + " ms");
  }
}
```

13.6.3使用HBase Java API把数据从本地导入到HBase中

请参照“第5章 HBase的安装和基础编程”的内容，在Eclipse中编写上述代码，并打包成可执行jar包，命名为ImportHBase.jar。然后，请在“/usr/local/bigdatacase/”目录下面新建一个hbase子目录，用来存放ImportHBase.jar。

13.6.3使用HBase Java API把数据从本地导入到HBase中

4.数据导入

现在开始执行数据导入操作。使用上面编写的Java程序ImportHBase.jar，将数据从本地导入HBase中。注意，在导入之前，请先清空user_action表。请在之前已经打开的HBase Shell窗口中（也就是在“hbase>”命令提示符下）执行下面操作：

```
hbase> truncate 'user_action'
```

下面就可以运行hadoop jar命令，来运行刚才的Java程序：

```
$ /usr/local/hadoop/bin/hadoop jar
/usr/local/bigdatacase/hbase/ImportHBase.jar ImportHBase
/usr/local/bigdatacase/dataset/user_action.output
```

13.6.3使用HBase Java API把数据从本地导入到HBase中

5.查看HBase中user_action表数据

下面，再次切换到HBase Shell窗口，执行下面命令查询数据：

```
habse> scan 'user_action',{LIMIT=>10}  #只查询前面10行
```

13.7 步骤四：利用R进行数据可视化分析

13.7.1安装R
13.7.2安装依赖库
13.7.3可视化分析

13.7.1安装R

Ubuntu自带的APT包管理器中的R安装包总是落后于标准版，因此，需要添加新的镜像源，把APT包管理中的R安装包更新到最新版。
请登录Linux系统（这里假设是Ubuntu16.04），打开一个终端，并注意保持网络连通，可以访问互联网，因为安装过程要下载各种安装文件。
首先，利用vim编辑器打开“/etc/apt/sources.list”文件，命令如下：

```
$ sudo vim /etc/apt/sources.list
```

把文件里的原始内容清空，在文件中添加阿里云的镜像源，即把如下内容添加到文件中：

```
deb http://mirrors.aliyun.com/ubuntu/ xenial main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ xenial-security main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ xenial-updates main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ xenial-backports main restricted universe multiverse
```

13.7.1安装R

保存文件退出vim编辑器，然后，执行如下命令更新软件源列表：

```
$ sudo apt-get update
```

如果更新软件源出现“由于没有公钥无法验证签名”的错误，请执行如下命令：

```
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 51716619E084DAB9
```

接下来，执行如下命令安装R语言：

```
$ sudo apt-get install r-base
```

系统会提示“您希望继续执行吗？[Y/n]”，可以直接键盘输入“Y”，就可以顺利安装结束。安装结束后，可以执行下面命令启动R：

```
$ R
```

最后，可以执行下面命令退出R：

```
>q()
```

13.7.2安装依赖库

请启动R进入R命令提示符状态，执行如下命令安装RMySQL：

```
> install.packages('RMySQL')
```

RMySQL安装成功以后，执行如下命令安装绘图包ggplot2：

```
> install.packages('ggplot2')
```

接下来，继续运行下面命令安装devtools：

```
> install.packages('devtools')
```

最后，在R命令提示符下，再执行如下命令安装taiyun/recharts：

```
> devtools::install_github('taiyun/recharts')
```

13.7.3可视化分析

1.连接MySQL,并获取数据

请在Linux系统中新建另外一个终端，然后执行下面命令启动MySQL数据库：

```
$service mysql start
```

执行如下命令进入MySQL命令提示符状态：

```
$ mysql -u root -p
```

接下来，可以输入一些SQL语句查询数据：

```
mysql> use dblab;
mysql> select * from user_action limit 10;
```

13.7.3可视化分析

然后，切换到刚才已经打开的R命令提示符终端窗口，使用如下命令让R连接到MySQL数据库：

```
>library(RMySQL)
>conn <-
dbConnect(MySQL(),dbname='dblab',username='root',password='hadoop',
host="127.0.0.1",port=3306)
>user_action <- dbGetQuery(conn,'select * from user_action')
```

13.7.3可视化分析

2.分析消费者对商品的行为

summary()函数可以得到样本数据类型和长度，如果样本是数值型，我们还能得到样本数据的最小值、最大值、四分位数以及均值信息。首先使用summary()函数查看MySQL数据库中表user_action的字段behavior_type的类型，命令如下：

```
>summary(user_action$behavior_type)
```

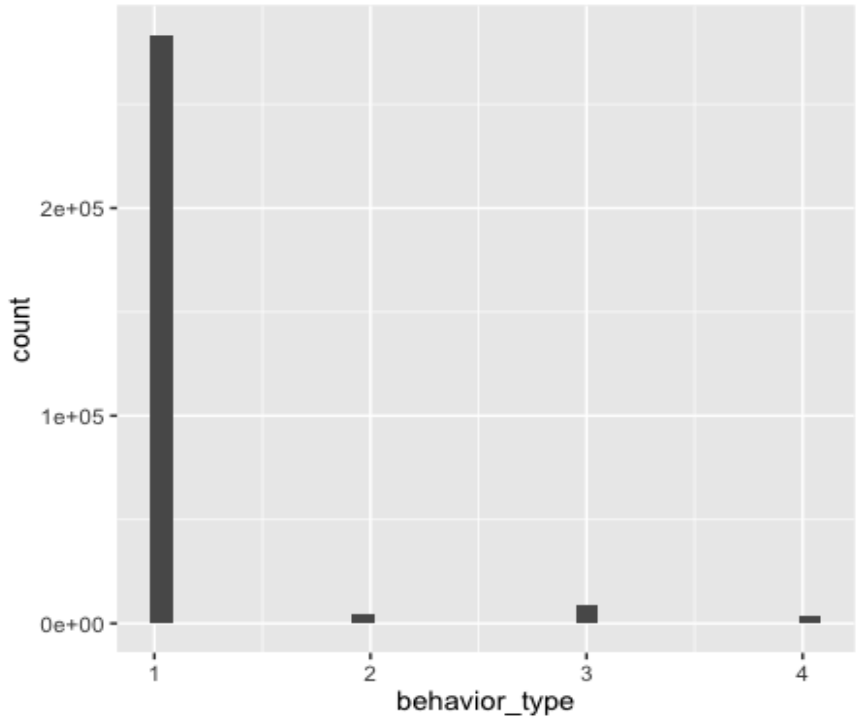
需要把behavior_type的字段类型转换为数值型，命令如下：

```
v>summary(as.numeric(user_action$behavior_type))
```

13.7.3可视化分析

接下来，用柱状图展示消费者的行为类型的分布情况，命令如下：

```
>library(ggplot2)
>ggplot(user_action,aes(as.numeric(behavior_type)))+geom_histogram()
```



13.7.3可视化分析

3.分析销量排名前十的商品及其销量

分析销量排名前十的商品及其销量，可以采用如下命令：

```
>temp <- subset(user_action,as.numeric(behavior_type)==4) # 获取子数据集
>count <- sort(table(temp$item_category),decreasing = T) #排序
>print(count[1:10]) # 获取第1到10个排序结果
```

4.分析每年的哪个月份销量最大

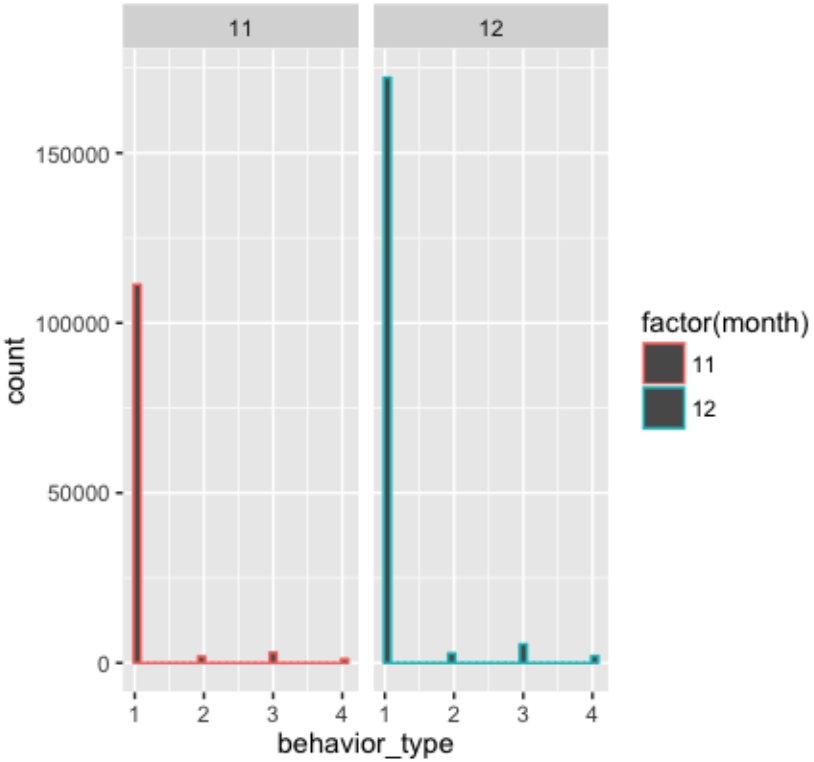
从MySQL直接获取的数据中visit_date变量都是2014年份，并没有划分出具体的月份，因此，需要在数据集中增加一列关于月份的数据，命令如下：

```
>month <- substr(user_action$visit_date,6,7)  # visit_date变量中截取月份
>user_action <- cbind(user_action,month)  # user_action增加一列月份数据
```

13.7.3可视化分析

接下来，用柱状图展示消费者在一年的不同月份的购买量情况，命令如下：

```
>ggplot(user_action,aes(as.numeric(behavior_type),col=factor(month)))+
geom_histogram()+facet_grid(.~month)
```



13.7.3可视化分析

5.分析国内哪个省份的消费者最有购买欲望

可以使用如下语句来分析国内各个省份的消费者的购买情况：

```
>library(recharts)
>rel <- as.data.frame(table(temp$province))
>provinces <- rel$Var1
>x = c()
>for(n in provinces){
>x[length(x)+1] = nrow(subset(temp,(province==n)))
>}
>mapData <- data.frame(province=rel$Var1,count=x, stringsAsFactors=F)
# 设置地图信息
>eMap(mapData, namevar=~province, datavar = ~count) #画出中国地图
```

13.8 本章小结

综合实验案例是大数据技术体系学习的重要内容，可以帮助读者形成对大数据技术综合运用方法的全局性认识，让前面各个章节所学的技术有效融会贯通，通过多种技术的组合来解决实际应用问题。本章的综合实验案例涵盖了Linux、MySQL、Hadoop、HBase、Hive、R、Eclipse等系统和软件的安装和使用方法，这些软件的安装和使用方法，被有效融合到实验的各个流程，可以有效加深对各种技术的理解。

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

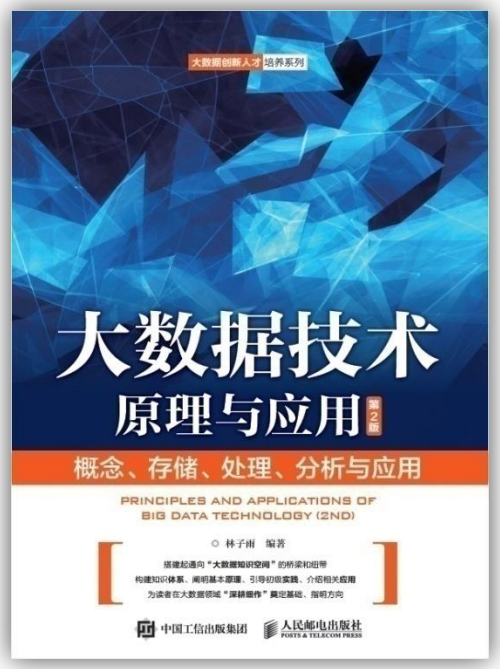
《大数据技术原理与应用——概念、存储、处理、分析与应用（第2版）》，由厦门大学计算机科学系林子雨博士编著，是国内高校第一本系统介绍大数据知识的专业教材。人民邮电出版社 ISBN:978-7-115-44330-4 定价：49.80元

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

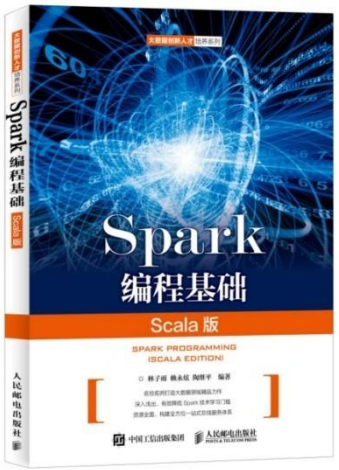
厦门大学林子雨编著

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）》

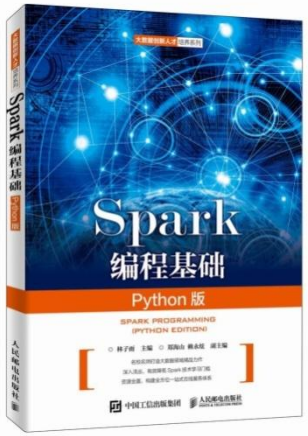
厦门大学 林子雨，赖永炫，陶继平 编著

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/

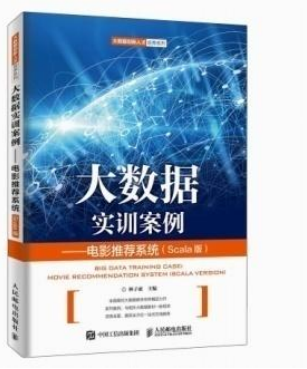
扫一扫访问平台主页

扫一扫观看3分钟FLASH动画宣传片

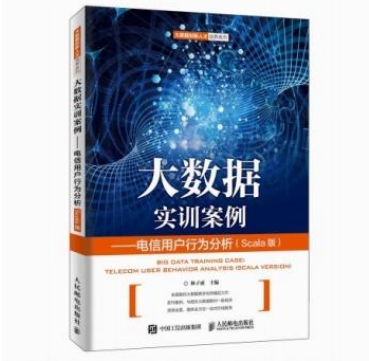
附录K：高校大数据实训课程系列案例教材

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!