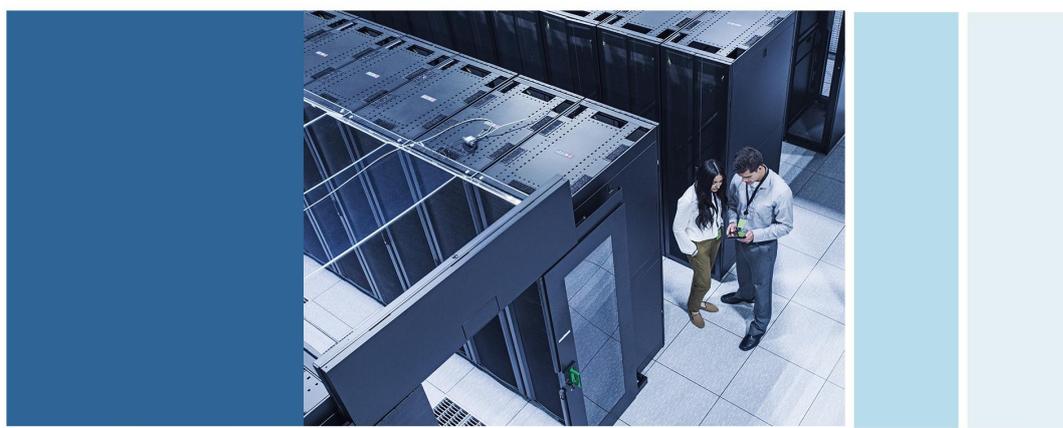


# Edu教学实验系统说明书

Edu teaching experiment system specification



曙光瑞翼教育合作中心 制  
二零一八年三月

# 声 明

本手册适用于V1.0版本，其用途在于帮助您正确地使用曙光瑞翼教育合作中心产品(以下称“本产品”)，在安装和第一次使用本产品前，请您务必先仔细阅读随机配送的所有资料，特别是本手册中所提及的注意事项。这会有助于您更好和安全地使用本产品。请妥善保管本手册，以便日后参阅。

本手册的描述并不代表对本产品规格和软硬件配置的任何说明。有关本产品的实际规格和配置，请查阅相关协议、装箱单、产品规格配置描述文件，或向产品的销售商咨询。

如您不正确地或未按本手册的指示和要求安装、使用或保管本产品，或让非曙光瑞翼教育合作中心授权的技术人员修理、变更本产品，曙光瑞翼教育合作中心将不对由此导致的损害承担任何责任。

本手册中所提供照片、图形、图表和插图，仅用于解释和说明目的，可能与实际产品有些差别，另外，产品实际规格和配置可能会根据需求不时变更，因此与本手册内容有所不同。请以实际产品为准。

本手册中所提及的非曙光瑞翼教育合作中心网站信息，是为了方便起见而提供，此类网站中的信息不是曙光瑞翼教育合作中心产品资料的一部分，也不是曙光瑞翼教育合作中心服务的一部分，曙光瑞翼教育合作中心对这些网站及信息的准确性和可用性不做任何保证。使用此类网站带来的风险将由您自行承担。

本手册不用于表明曙光瑞翼教育合作中心对其产品和服务做了任何保证，无论是明示的还是默示的，包括(但不限于)本手册中推荐使用产品的适用性、安全性、适销性和适合某特定用途的保证。对本产品及相关服务的保证和保修承诺，应按可适用的协议或产品标准保修服务条款和条件执行。在法律法规的最大允许范围内，曙光瑞翼教育合作中心对于您的使用或不能使用本产品而发生的任何损害(包括，但不限于直接或间接的个人损害、商业利润的损失、业务中断、商业信息的遗失或任何其他损失)，不负任何赔偿责任。

对于您在本产品之外使用本产品随机提供的软件，或在本产品上使用非随机软件或经曙光瑞翼教育合作中心认证推荐使用的专用软件之外的其他软件，曙光瑞翼教育合作中心对其可靠性不做任何保证。

曙光瑞翼教育合作中心已经对本手册进行了仔细的校勘和核对，但不能保证本手册完全没有任何错误和疏漏。为更好地提供服务，曙光瑞翼教育合作中心可能会对本手册中描述的产品软件和硬件及本手册的内容随时进行改进或更改，恕不另行通知。

# 目 录

|                             |   |
|-----------------------------|---|
| 1、 用户指南.....                | 1 |
| 1.1 目标用户.....               | 1 |
| 1.2 入门.....                 | 1 |
| 2、 产品简介.....                | 1 |
| 2.1 运行环境.....               | 1 |
| 2.2 Sugon_edu 教学实验系统简介..... | 2 |
| 2.2.1 简介.....               | 2 |
| 2.2.2 内容.....               | 2 |
| 3、 课程管理界面.....              | 3 |
| 4、 课程及实验介绍.....             | 4 |
| 4.1 Linux与Shell基础.....      | 4 |
| 4.1.1 课程界面.....             | 4 |
| 4.1.2 Linux与Shell简介.....    | 4 |
| 4.2 大数据概论.....              | 5 |
| 4.2.1 课程界面.....             | 5 |
| 4.2.2 大数据概论.....            | 6 |
| 4.3 Hadoop安装部署.....         | 7 |
| 4.3.1 课程界面.....             | 7 |
| 4.3.2 Hadoop安装部署.....       | 7 |
| 4.4 Hadoop运维管理.....         | 8 |
| 4.4.1 课程界面.....             | 8 |

|                              |    |
|------------------------------|----|
| 4.4.2 Hadoop运维管理 .....       | 8  |
| 4.5 HDFS基本原理 .....           | 9  |
| 4.5.1 课程界面 .....             | 9  |
| 4.5.2 HDFS基本原理 .....         | 10 |
| 4.6 Hbase基本原理 .....          | 10 |
| 4.6.1 课程界面 .....             | 10 |
| 4.6.2 Hbase基本原理 .....        | 11 |
| 4.7 Hive基本原理 .....           | 11 |
| 4.7.1 课程界面 .....             | 11 |
| 4.7.2 Hive基本原理 .....         | 11 |
| 4.8 ZooKeeper基本原理 .....      | 12 |
| 4.8.1 课程界面 .....             | 12 |
| 4.8.2 ZooKeeper基本原理 .....    | 12 |
| 4.9 Yarn基本原理 .....           | 13 |
| 4.9.1 课程界面 .....             | 13 |
| 4.9.2 Yarn基本原理 .....         | 13 |
| 4.10 MapReduce基本原理 .....     | 15 |
| 4.10.1 课程界面 .....            | 15 |
| 4.10.2 MapReduce基本原理 .....   | 15 |
| 4.11 MapReduce高级编程实战 .....   | 16 |
| 4.11.1 课程界面 .....            | 16 |
| 4.11.2 MapReduce高级编程实战 ..... | 16 |

---

|                         |    |
|-------------------------|----|
| 4.12 Spark编程实战.....     | 19 |
| 4.12.1 课程界面.....        | 19 |
| 4.12.2 Spark编程实战.....   | 20 |
| 4.13 ETL工具使用.....       | 21 |
| 4.13.1 课程界面.....        | 21 |
| 4.13.2 ETL工具使用.....     | 21 |
| 4.14 数据可视化工具实战.....     | 22 |
| 4.14.1 课程界面.....        | 22 |
| 4.14.2 数据可视化工具实战.....   | 22 |
| 4.15 日志采集分析.....        | 23 |
| 4.15.1 课程界面.....        | 23 |
| 4.15.2 日志采集分析.....      | 23 |
| 4.16 用户画像与用户行为分析.....   | 24 |
| 4.16.1 课程界面.....        | 24 |
| 4.16.2 用户画像与用户行为分析..... | 24 |

# 1、用户指南

## 1.1 目标用户

《Edu教学实验系统操作手册》是为需要利用教学实验系统进行Linux与Shell基础、大数据概率、Hadoop安装部署、Hadoop运维管理、HDFS基本原理、Hbase基本原理、Hive基本原理、ZooKeeper基本原理、Yarn基本原理、MapReduce基本原理、MapReduce高级编程实战、Spark编程实战、ETL工具使用、数据可视化工具实战、日志采集分析、用户画像与用户行为分析课程的学习。

## 1.2 入门

Edu教学实验系统为用户提供了教学服务和实验服务。在这里用户可以了解到hadoop、HDFS、Hbase等的大数据有关平台和工具的具体功能和使用方法。

# 2、产品简介

## 2.1 运行环境

本系统运行于LAMP环境下。目前使用的LAMP环境版本如下

表1 LAMP环境

| 名称     | 版本             |
|--------|----------------|
| Ubuntu | 14.04.5        |
| Apache | 2.4.7 (Ubuntu) |
| Mysql  | 5.5.52         |
| PHP    | 5.5.9-lubuntu  |

## 2.2 Edu教学实验系统简介

### 2.2.1 简介

采用 B/S架构，无需安装客户端软件，使用户更容易获取 web教学资源。

### 2.2.2 内容

具体大数据实验教学环节包括如下内容：

- (1)Linux与Shell基础
- (2)大数据概论
- (3)Hadoop安装部署
- (4)Hadoop运维管理
- (5)HDFS基本原理
- (6)Hbase基本原理
- (7)Hive基本原理
- (8)ZooKeeper基本原理
- (9)Yarn基本原理
- (10)MapReduce基本原理
- (11)MapReduce高级编程实战
- (12)Spark编程实战
- (13)ETL工具使用
- (14)数据可视化工具实战
- (15)日志采集分析
- (16)用户画像与用户行为分析

### 3、课程管理界面

进入实验系统登陆界面，浏览器输入：IP/edu

点击界面右上角的LogIn通过用户名和密码就能进入教学实验系统界面。



图3-1 系统登陆界面

登陆后的界面如下图



图3-2 后台系统

## 4、课程及实验介绍

### 4.1 Linux与Shell基础

#### 4.1.1 课程界面



图4-1 Linux与Shell基础

#### 4.1.2 Linux与Shell简介

Linux是一套免费使用和自由传播的类Unix操作系统，是一个基于POSIX和UNIX的多用户、多任务、支持多线程和多CPU的操作系统。它能运行主要的UNIX工具软件、应用程序和网络协议。它支持32位和64位硬件。Linux继承了Unix以网络为核心的设计思想，是一个性能稳定的多用户网络操作系统。

Linux操作系统诞生于1991年10月5日（这是第一次正式向外公布时间）。Linux存在着许多不同的Linux版本，但它们都使用了Linux内核。Linux可安装在各种计算机硬件设备中，比如手机、平板电脑、路由器、视频游戏控制台、台式计算机、大型机和超级计算机。

严格来讲，Linux这个词本身只表示Linux内核，但实际上人们已经习惯了用Linux来形容整个基于Linux内核，并且使用GNU工程各种工具和数据库的操作系统。

统。

Bash (GNU Bourne-Again Shell) 是许多Linux发行版的默认Shell。事实上，还有许多传统UNIX上用的Shell，例如tsh、csh、ash、bsh、ksh等等，Shell Script大致都类同，当您学会一种Shell以后，其它的Shell会很快就上手，大多数的时候，一个Shell Script通常可以在很多种Shell上使用。

bash是大多数Linux系统以及Mac OS X默认的shell，它能运行于大多数类Unix风格的操作系统之上，甚至被移植到了Microsoft Windows上的Cygwin系统中，以实现Windows的POSIX虚拟接口。此外，它也被DJGPP项目移植到了MS-DOS上。

bash的命令语法是Bourne shell命令语法的超集。数量庞大的Bourne shell脚本大多不经修改即可以在bash中执行，只有使用了Bourne的特殊变量或内置命令的脚本才需要修改。bash的命令语法很多来自Korn shell (ksh) 和 C shell (csh)，例如命令行编辑，命令历史，目录栈，\$RANDOM 和 \$PPID 变量，以及POSIX的命令置换语法：\$(...)。作为一个交互式的shell，按下TAB键即可自动补全已部分输入的程序名，文件名，变量名等等。

## 4.2 大数据概论

### 4.2.1 课程界面

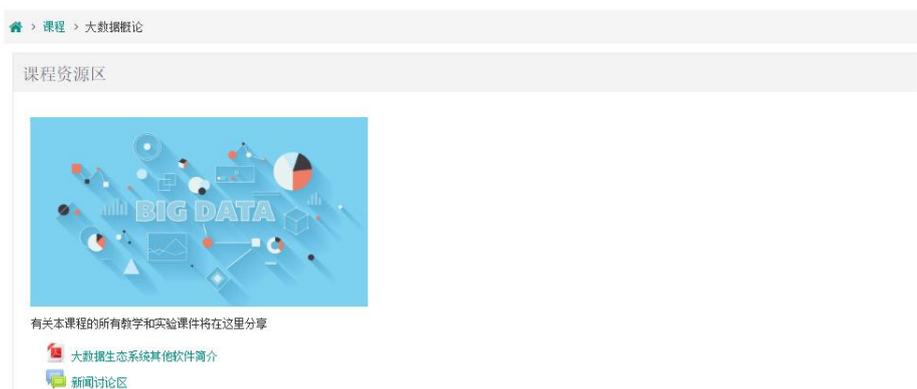


图4-2 大数据概论

## 4.2.2 大数据概论

对于“大数据”（Big data）研究机构Gartner给出了这样的定义。“大数据”是需要新处理模式才能具有更强的决策力、洞察发现力和流程优化能力来适应海量、高增长率和多样化的信息资产。

麦肯锡全球研究所给出的定义是：一种规模大到在获取、存储、管理、分析方面大大超出了传统数据库软件工具能力范围的数据集合，具有海量的数据规模、快速的数据流转、多样的数据类型和价值密度低四大特征。

大数据技术的战略意义不在于掌握庞大的数据信息，而在于对这些含有意义的数据进行专业化处理。换言之，如果把大数据比作一种产业，那么这种产业实现盈利的关键，在于提高对数据的“加工能力”，通过“加工”实现数据的“增值”。

从技术上看，大数据与云计算的关系就像一枚硬币的正反面一样密不可分。大数据必然无法用单台的计算机进行处理，必须采用分布式架构。它的特色在于对海量数据进行分布式数据挖掘。但它必须依托云计算的分布式处理、分布式数据库和云存储、虚拟化技术。随着云时代的来临，大数据（Big data）也吸引了越来越多的关注。分析师团队认为，大数据（Big data）通常用来形容一个公司创造的大量非结构化数据和半结构化数据，这些数据在下载至关系型数据库用于分析时会花费过多时间和金钱。大数据分析常和云计算联系到一起，因为实时的大型数据集分析需要像MapReduce一样的框架来向数十、数百或甚至数千的电脑分配工作。

大数据需要特殊的技术，以有效地处理大量的容忍经过时间内的数据。适用于大数据的技术，包括大规模并行处理（MPP）数据库、数据挖掘、分布式文件系统、分布式数据库、云计算平台、互联网和可扩展的存储系统。

## 4.3 Hadoop安装部署

### 4.3.1 课程界面

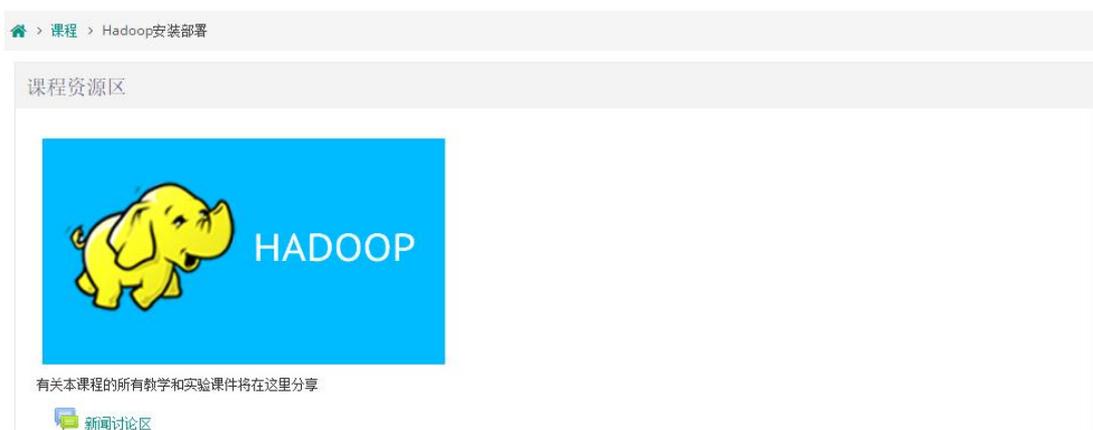


图4-3 Hadoop安装部署

### 4.3.2 Hadoop安装部署

Hadoop是一个由Apache基金会所开发的分布式系统基础架构。

用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力进行高速运算和存储。

Hadoop实现了一个分布式文件系统（Hadoop Distributed File System），简称HDFS。HDFS有高容错性的特点，并且设计用来部署在低廉的（low-cost）硬件上；而且它提供高吞吐量（high throughput）来访问应用程序的数据，适合那些有着超大数据集（large data set）的应用程序。HDFS放宽了（relax）POSIX的要求，可以以流的形式访问（streaming access）文件系统中的数据。

Hadoop的框架最核心的设计就是：HDFS和MapReduce。HDFS为海量的数据提供了存储，则MapReduce为海量的数据提供了计算。

## 4.4 Hadoop运维管理

### 4.4.1 课程界面

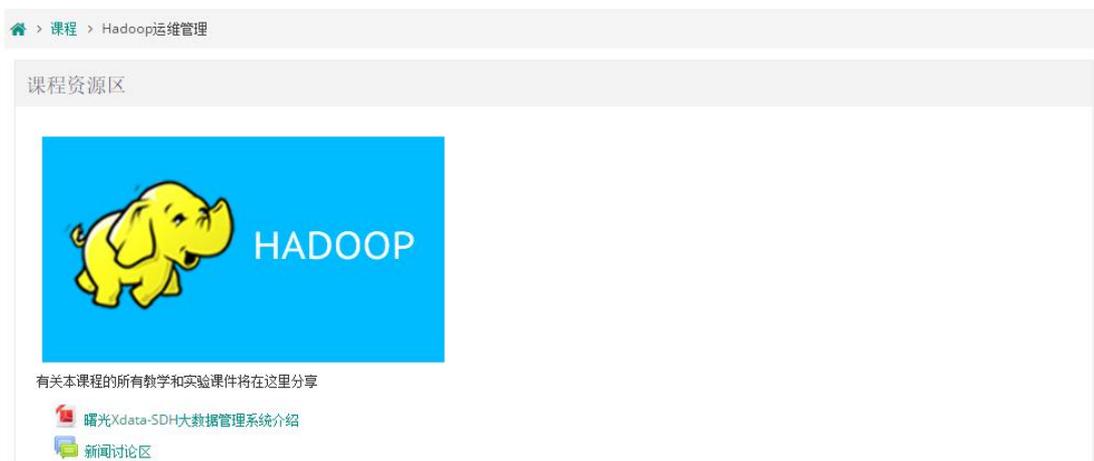


图4-4 Hadoop运维管理

### 4.4.2 Hadoop运维管理

Hadoop是一个能够对大量数据进行分布式处理的软件框架。Hadoop以一种可靠、高效、可伸缩的方式进行数据处理。

Hadoop是可靠的，因为它假设计算元素和存储会失败，因此它维护多个工作数据副本，确保能够针对失败的节点重新分布处理。

Hadoop是高效的，因为它以并行的方式工作，通过并行处理加快处理速度。

Hadoop还是可伸缩的，能够处理PB级数据。

此外，Hadoop依赖于社区服务，因此它的成本比较低，任何人都可以使用。

Hadoop是一个能够让用户轻松架构和使用的分布式计算平台。用户可以轻松地在Hadoop上开发和运行处理海量数据的应用程序。它主要有以下几个优点：

高可靠性。Hadoop按位存储和处理数据的能力值得人们信赖。

高扩展性。Hadoop是在可用的计算机集簇间分配数据并完成计算任务的，这些集簇可以方便地扩展到数以千计的节点中。

高效性。Hadoop能够在节点之间动态地移动数据，并保证各个节点的动态平衡，因此处理速度非常快。

高容错性。Hadoop能够自动保存数据的多个副本，并且能够自动将失败的任务重新分配。

低成本。与一体机、商用数据仓库以及QlikView、Yonghong Z-Suite等数据集市相比，hadoop是开源的，项目的软件成本因此会大大降低。

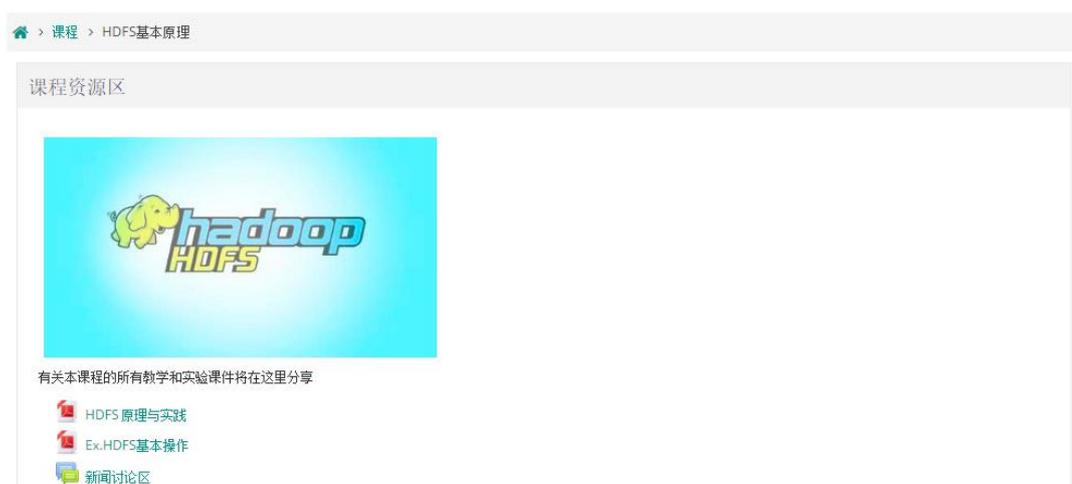
Hadoop带有用Java语言编写的框架，因此运行在 Linux 生产平台上是非常理想的。Hadoop 上的应用程序也可以使用其他语言编写，比如 C++。

#### hadoop大数据处理的意义

Hadoop得以在大数据处理应用中广泛应用得益于其自身在数据提取、变形和加载(ETL)方面的天然优势。Hadoop的分布式架构，将大数据处理引擎尽可能的靠近存储，对例如像ETL这样的批处理操作相对合适，因为类似这样操作的批处理结果可以直接走向存储。Hadoop的MapReduce功能实现了将单个任务打碎，并将碎片任务(Map)发送到多个节点上，之后再以单个数据集的形式加载(Reduce)到数据仓库里。

## 4.5 HDFS基本原理

### 4.5.1 课程界面



4-5 HDFS基本原理

## 4.5.2 HDFS基本原理

Hadoop分布式文件系统(HDFS)被设计成适合运行在通用硬件(commodity hardware)上的分布式文件系统。它和现有的分布式文件系统有很多共同点。但同时,它和其他的分布式文件系统的区别也是很明显的。HDFS是一个高度容错性的系统,适合部署在廉价的机器上。HDFS能提供高吞吐量的数据访问,非常适合大规模数据集上的应用。HDFS放宽了一部分POSIX约束,来实现流式读取文件系统数据的目的。HDFS在最开始是作为Apache Nutch搜索引擎项目的基础架构而开发的。HDFS是Apache Hadoop Core项目的一部分。

HDFS有着高容错性(fault-tolerant)的特点,并且设计用来部署在低廉的(low-cost)硬件上。而且它提供高吞吐量(high throughput)来访问应用程序的数据,适合那些有着超大数据集(large data set)的应用程序。HDFS放宽了(relax)POSIX的要求(requirements)这样可以实现流的形式访问(streaming access)文件系统中的数据。

## 4.6 Hbase基本原理

### 4.6.1 课程界面

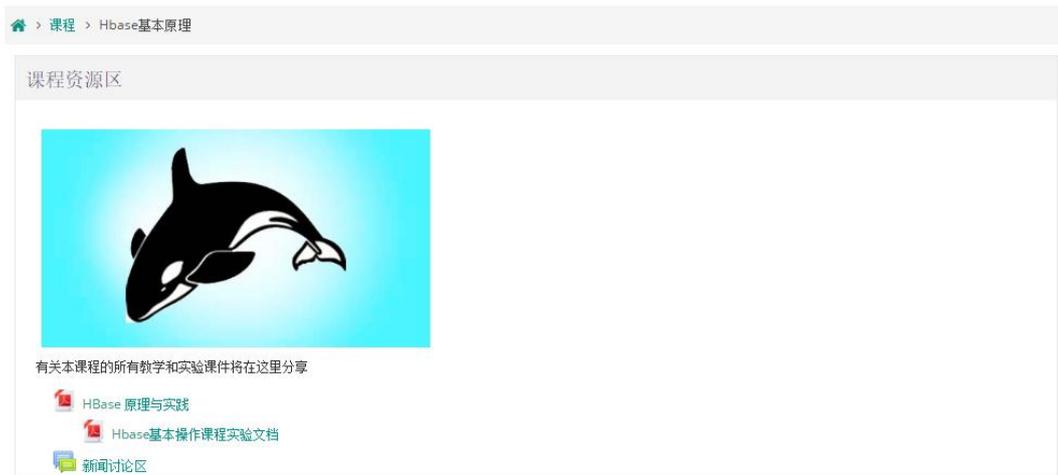


图4-6 Hbase基本原理

## 4.6.2 Hbase基本原理

HBase是一个分布式的、面向列的开源数据库，该技术来源于 Fay Chang 所撰写的Google论文“Bigtable: 一个结构化数据的分布式存储系统”。就像Bigtable利用了Google文件系统（File System）所提供的分布式数据存储一样，HBase在Hadoop之上提供了类似于Bigtable的能力。HBase是Apache的Hadoop项目的子项目。HBase不同于一般的关系数据库，它是一个适合于非结构化数据存储的数据库。另一个不同的是HBase基于列的而不是基于行的模式。

## 4.7 Hive基本原理

### 4.7.1 课程界面



图4-7 Hive基本原理

### 4.7.2 Hive基本原理

Hive是建立在 Hadoop 上的数据仓库基础构架。它提供了一系列的工具，可以用来进行数据提取转化加载（ETL），这是一种可以存储、查询和分析存储在 Hadoop 中的大规模数据的机制。Hive 定义了简单的类 SQL 查询语言，称为 HQL，它允许熟悉 SQL 的用户查询数据。同时，这个语言也允许熟悉 MapReduce 开发者的开发自定义的 mapper 和 reducer 来处理内建的 mapper 和 reducer 无法完成的复杂的分析工作。

Hive 没有专门的数据格式。Hive 可以很好的工作在 Thrift 之上，控制分隔符，也允许用户指定数据格式。

## 4.8 ZooKeeper基本原理

### 4.8.1 课程界面



图4-8 ZooKeeper基本原理

### 4.8.2 ZooKeeper基本原理

ZooKeeper是一个分布式的，开放源码的分布式应用程序协调服务，是Google的Chubby一个开源的实现，是Hadoop和Hbase的重要组件。它是一个为分布式应用提供一致性服务的软件，提供的功能包括：配置维护、域名服务、分布式同步、组服务等。

ZooKeeper的目标就是封装好复杂易出错的关键服务，将简单易用的接口和性能高效、功能稳定的系统提供给用户。

ZooKeeper包含一个简单的原语集，提供Java和C的接口。

ZooKeeper代码版本中，提供了分布式独享锁、选举、队列的接口，代码在zookeeper-3.4.3\src\recipes。其中分布锁和队列有Java和C两个版本，选举只有Java版本。

ZooKeeper是以Fast Paxos算法为基础的，Paxos 算法存在活锁的问题，即当有

多个proposer交错提交时，有可能互相排斥导致没有一个proposer能提交成功，而Fast Paxos作了一些优化，通过选举产生一个leader（领导者），只有leader才能提交proposer，具体算法可见Fast Paxos。因此，要想弄懂ZooKeeper首先得对Fast Paxos有所了解。[3]

ZooKeeper的基本运转流程：

- (1) 选举Leader。
- (2) 同步数据。
- (3) 选举Leader过程中算法有很多，但要达到的选举标准是一致的。
- (4) Leader要具有最高的zxid。
- (5) 集群中大多数的机器得到响应并follow选出的Leader。

## 4.9 Yarn基本原理

### 4.9.1 课程界面



图4-9 Yarn基本原理

### 4.9.2 Yarn基本原理

Apache Hadoop YARN（Yet Another Resource Negotiator，另一种资源协调者）是一种新的 Hadoop 资源管理器，它是一个通用资源管理系统，可为上层应用

提供统一的资源管理和调度，它的引入为集群在利用率、资源统一管理和数据共享等方面带来了巨大好处。

YARN的基本思想是将JobTracker的两个主要功能（资源管理和作业调度/监控）分离，主要方法是创建一个全局的ResourceManager（RM）和若干个针对应用程序的ApplicationMaster（AM）。这里的应用程序是指传统的MapReduce作业或作业的DAG（有向无环图）。

YARN 分层结构的本质是 ResourceManager。这个实体控制整个集群并管理应用程序向基础计算资源的分配。ResourceManager 将各个资源部分（计算、内存、带宽等）精心安排给基础 NodeManager（YARN 的每节点代理）。ResourceManager 还与 ApplicationMaster 一起分配资源，与 NodeManager 一起启动和监视它们的基础应用程序。在此上下文中，ApplicationMaster 承担了以前的 TaskTracker 的一些角色，ResourceManager 承担了 JobTracker 的角色。

ApplicationMaster 管理一个在 YARN 内运行的应用程序的每个实例。ApplicationMaster 负责协调来自 ResourceManager 的资源，并通过 NodeManager 监视容器的执行和资源使用（CPU、内存等的资源分配）。请注意，尽管目前的资源更加传统（CPU 核心、内存），但未来会带来基于手头任务的新资源类型（比如图形处理单元或专用处理设备）。从 YARN 角度讲，ApplicationMaster 是用户代码，因此存在潜在的安全问题。YARN 假设 ApplicationMaster 存在错误或者甚至是恶意的，因此将它们当作无特权的代码对待。

NodeManager 管理一个 YARN 集群中的每个节点。NodeManager 提供针对集群中每个节点的服务，从监督对一个容器的终生管理到监视资源和跟踪节点健康。MRv1 通过插槽管理 Map 和 Reduce 任务的执行，而 NodeManager 管理抽象容器，这些容器代表着可供一个特定应用程序使用的针对每个节点的资源。YARN 继续使用 HDFS 层。它的主要 NameNode 用于元数据服务，而 DataNode 用于分散在一个集群中的复制存储服务。

要使用一个 YARN 集群，首先需要来自包含一个应用程序的客户的请求。ResourceManager 协商一个容器的必要资源，启动一个 ApplicationMaster 来表示

已提交的应用程序。通过使用一个资源请求协议，ApplicationMaster 协商每个节点上供应用程序使用的资源容器。执行应用程序时，ApplicationMaster 监视容器直到完成。当应用程序完成时，ApplicationMaster 从 ResourceManager 注销其容器，执行周期就完成了。

## 4.10 MapReduce基本原理

### 4.10.1 课程界面

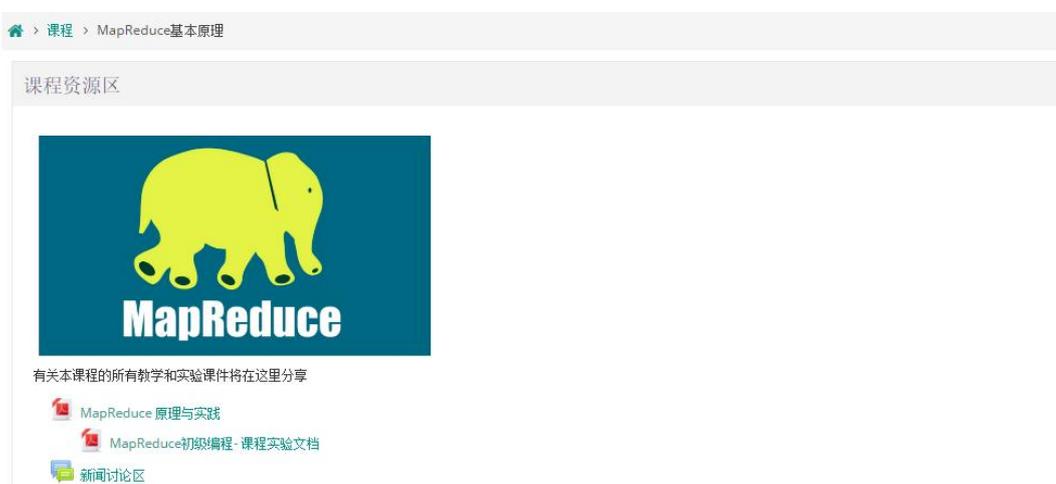


图4-10 MapReduce基本原理

### 4.10.2 MapReduce基本原理

MapReduce是一种编程模型，用于大规模数据集（大于1TB）的并行运算。概念“Map（映射）”和“Reduce（归约）”，是它们的主要思想，都是从函数式编程语言里借来的，还有从矢量编程语言里借来的特性。它极大地方便了编程人员在不会分布式并行编程的情况下，将自己的程序运行在分布式系统上。当前的软件实现是指定一个Map（映射）函数，用来把一组键值对映射成一组新的键值对，指定并发的Reduce（归约）函数，用来保证所有映射的键值对中的每一个共享相同的键组。

## 4.11 MapReduce高级编程实战

### 4.11.1 课程界面

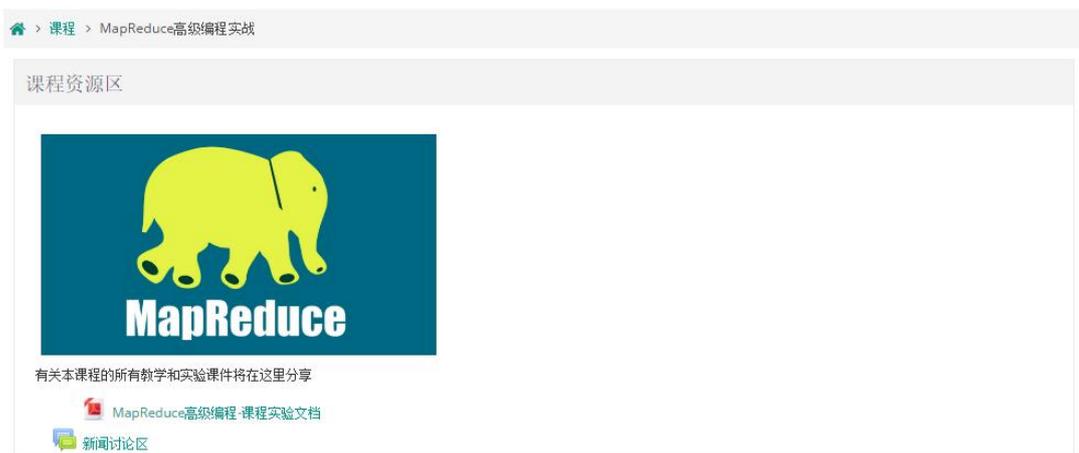


图4-11 MapReduce高级编程实战

### 4.11.2 MapReduce高级编程实战

MapReduce提供了以下的主要功能：

#### (1) 数据划分和计算任务调度：

系统自动将一个作业（Job）待处理的大数据划分为很多个数据块，每个数据块对应于一个计算任务（Task），并自动调度计算节点来处理相应的数据块。作业和任务调度功能主要负责分配和调度计算节点（Map节点或Reduce节点），同时负责监控这些节点的执行状态，并负责Map节点执行的同步控制。

#### (2) 数据/代码互定位：

为了减少数据通信，一个基本原则是本地化数据处理，即一个计算节点尽可能处理其本地磁盘上所分布存储的数据，这实现了代码向数据的迁移；当无法进行这种本地化数据处理时，再寻找其他可用节点并将数据从网络上传送给该节点（数据向代码迁移），但将尽可能从数据所在的本地机架上寻找可用节点以减少通信延迟。

### (3) 系统优化:

为了减少数据通信开销，中间结果数据进入Reduce节点前会进行一定的合并处理；一个Reduce节点所处理的数据可能会来自多个 Map节点，为了避免Reduce计算阶段发生数据相关性，Map节点输出的中间结果需使用一定的策略进行适当的划分处理，保证相关性数据发送到同一个 Reduce节点；此外，系统还进行一些计算性能优化处理，如对最慢的计算任务采用多备份执行、选最快完成者作为结果。

### (4) 出错检测和恢复:

以低端商用服务器构成的大规模MapReduce计算集群中，节点硬件（主机、磁盘、内存等）出错和软件出错是常态，因此 MapReduce需要能检测并隔离出错节点，并调度分配新的节点接管出错节点的计算任务。同时，系统还将维护数据存储的可靠性，用多备份冗余存储机制提高数据存储的可靠性，并能及时检测和恢复出错的数据。

## 主要技术特征编辑

MapReduce设计上具有以下主要的技术特征:

### (1) 向“外”横向扩展，而非向“上”纵向扩展

即MapReduce集群的构建完全选用价格便宜、易于扩展的低端商用服务器，而非价格昂贵、不易扩展的高端服务器。

对于大规模数据处理，由于有大量数据存储需要，显而易见，基于低端服务器的集群远比基于高端服务器的集群优越，这就是为什么MapReduce并行计算集群会基于低端服务器实现的原因。

### (2) 失效被认为是常态

MapReduce集群中使用大量的低端服务器，因此，节点硬件失效和软件出错是常态，因而一个良好设计、具有高容错性的并行计算系统不能因为节点失效而影响计算服务的质量，任何节点失效都不应当导致结果的不一致或不确定性；任何一个节点失效时，其他节点要能够无缝接管失效节点的计算任务；当失效节点恢复后应能自动无缝加入集群，而不需要管理员人工进行系统配置。

MapReduce并行计算软件框架使用了多种有效的错误检测和恢复机制，如节点自动重启技术，使集群和计算框架具有对付节点失效的健壮性，能有效处理失效节点的检测和恢复。

### (3) 把处理向数据迁移

传统高性能计算系统通常有很多处理器节点与一些外存储器节点相连，如用存储区域网络（Storage Area, SAN Network）连接的磁盘阵列，因此，大规模数据处理时外存文件数据I/O访问会成为一个制约系统性能的瓶颈。

为了减少大规模数据并行计算系统中的数据通信开销，代之以把数据传送到处理节点（数据向处理器或代码迁移），应当考虑将处理向数据靠拢和迁移。

MapReduce采用了数据/代码互定位的技术方法，计算节点将首先尽量负责计算其本地存储的数据，以发挥数据本地化特点，仅当节点无法处理本地数据时，再采用就近原则寻找其他可用计算节点，并把数据传送到该可用计算节点。

### (4) 顺序处理数据、避免随机访问数据

大规模数据处理的特点决定了大量的数据记录难以全部存放在内存，而通常只能放在外存中进行处理。由于磁盘的顺序访问要远比随机访问快得多，因此MapReduce主要设计为面向顺序式大规模数据的磁盘访问处理。

为了实现面向大数据集批处理的高吞吐量的并行处理，MapReduce可以利用集群中的大量数据存储节点同时访问数据，以此利用分布集群中大量节点上的磁盘集合提供高带宽的数据访问和传输。

### (5) 为应用开发者隐藏系统层细节

软件工程实践指南中，专业程序员认为之所以写程序困难，是因为程序员需要记住太多的编程细节（从变量名到复杂算法的边界情况处理），这对大脑记忆是一个巨大的认知负担，需要高度集中注意力；而并行程序编写有更多困难，如需要考虑多线程中诸如同步等复杂繁琐的细节。由于并发执行中的不可预测性，程序的调试查错也十分困难；而且，大规模数据处理时程序员需要考虑诸如数据分布存储管理、数据分发、数据通信和同步、计算结果收集等诸多细节问题。

MapReduce提供了一种抽象机制将程序员与系统层细节隔离开来，程序员仅需描述需要计算什么（What to compute），而具体怎么去计算（How to compute）就交由系统的执行框架处理，这样程序员可从系统层细节中解放出来，而致力于其应用本身计算问题的算法设计。

#### (6) 平滑无缝的可扩展性

这里指出的可扩展性主要包括两层意义上的扩展性：数据扩展和系统规模扩展性。

理想的软件算法应当能随着数据规模的扩大而表现出持续的有效性，性能上的下降程度应与数据规模扩大的倍数相当；在集群规模上，要求算法的计算性能应能随着节点数的增加保持接近线性程度的增长。绝大多数现有的单机算法都达不到以上理想的要求；把中间结果数据维护在内存中的单机算法在大规模数据处理时很快失效；从单机到基于大规模集群的并行计算从根本上需要完全不同的算法设计。奇妙的是，MapReduce在很多情形下能实现以上理想的扩展性特征。

多项研究发现，对于很多计算问题，基于MapReduce的计算性能可随节点数目增长保持近似于线性的增长。

## 4.12 Spark编程实战

### 4.12.1 课程界面

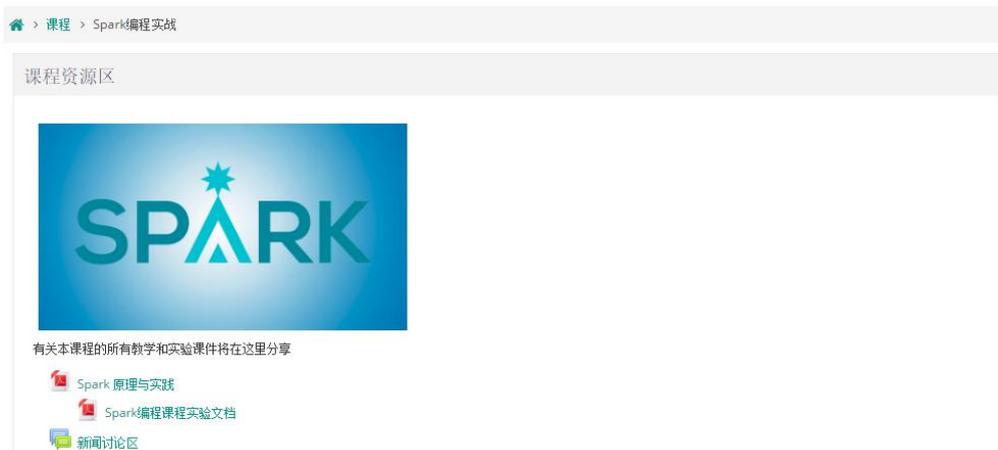


图4-12 Spark编程实战

## 4.12.2 Spark编程实战

Spark是UC Berkeley AMP lab (加州大学伯克利分校的AMP实验室)所开源的类Hadoop MapReduce的通用并行框架, Spark, 拥有Hadoop MapReduce所具有的优点;但不同于MapReduce的是Job中间输出结果可以保存在内存中, 从而不再需要读写HDFS, 因此Spark能更好地适用于数据挖掘与机器学习等需要迭代的MapReduce的算法。

Spark 是一种与 Hadoop 相似的开源集群计算环境, 但是两者之间还存在一些不同之处, 这些有用的不同之处使 Spark 在某些工作负载方面表现得更加优越, 换句话说, Spark 启用了内存分布数据集, 除了能够提供交互式查询外, 它还可以优化迭代工作负载。

Spark 是在 Scala 语言中实现的, 它将 Scala 用作其应用程序框架。与 Hadoop 不同, Spark 和 Scala 能够紧密集成, 其中的 Scala 可以像操作本地集合对象一样轻松地操作分布式数据集。

尽管创建 Spark 是为了支持分布式数据集上的迭代作业, 但是实际上它是对 Hadoop 的补充, 可以在 Hadoop 文件系统中并行运行。通过名为 Mesos 的第三方集群框架可以支持此行为。Spark 由加州大学伯克利分校 AMP 实验室 (Algorithms, Machines, and People Lab) 开发, 可用于构建大型的、低延迟的数据分析应用程序。

## 4.13 ETL工具使用

### 4.13.1 课程界面



图4-13 ETL工具使用

### 4.13.2 ETL工具使用

ETL，是英文 Extract-Transform-Load 的缩写，用来描述将数据从来源端经过抽取（extract）、转换（transform）、加载（load）至目的端的过程。ETL一词较常用在数据仓库，但其对象并不限于数据仓库。

ETL是构建数据仓库的重要一环，用户从数据源抽取出所需的数据，经过数据清洗，最终按照预先定义好的数据仓库模型，将数据加载到数据仓库中去。

信息是现代企业的重要资源，是企业运用科学管理、决策分析的基础。目前，大多数企业花费大量的资金和时间来构建联机事务处理OLTP的业务系统和办公自动化系统，用来记录事务处理的各种相关数据。据统计，数据量每2~3年时间就会成倍增长，这些数据蕴含着巨大的商业价值，而企业所关注的通常只占在总数据量的2%~4%左右。因此，企业仍然没有最大化地利用已存在的数据资源，以致于浪费了更多的时间和资金，也失去制定关键商业决策的最佳契机。于是，企业如何通过各种技术手段，并把数据转换为信息、知识，已经成了提高其核心竞争力的主要瓶颈。而ETL则是主要的一个技术手段。

## 4.14 数据可视化工具实战

### 4.14.1 课程界面

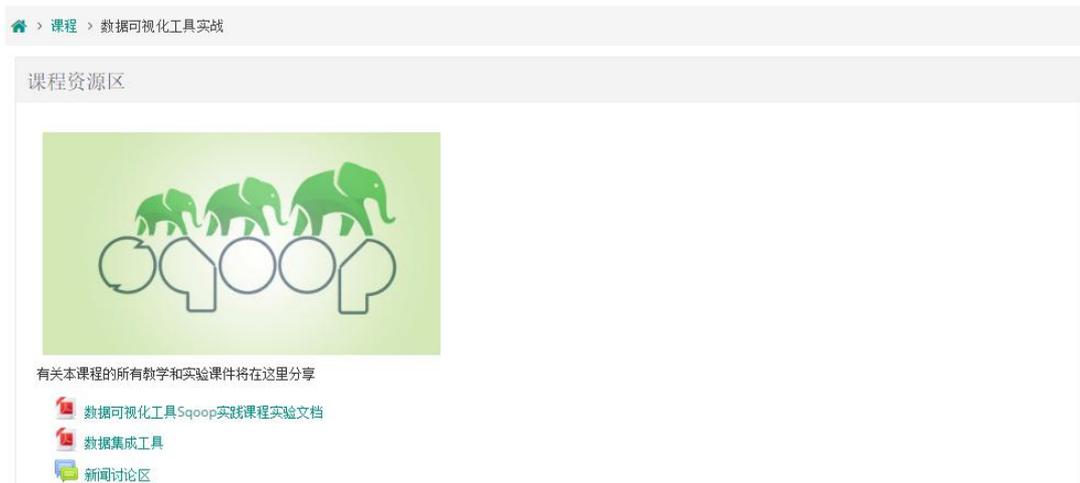


图4-14 数据可视化工具实战

### 4.14.2 数据可视化工具实战

数据可视化，是关于数据视觉表现形式的科学技术研究。其中，这种数据的视觉表现形式被定义为，一种以某种概要形式抽提出来的信息，包括相应信息单位的各种属性和变量。

它是一个处于不断演变之中的概念，其边界在不断地扩大。主要指的是技术上较为高级的技术方法，而这些技术方法允许利用图形、图像处理、计算机视觉以及用户界面，通过表达、建模以及对立体、表面、属性以及动画的显示，对数据加以可视化解释。与立体建模之类的特殊技术方法相比，数据可视化所涵盖的技术方法要广泛得多。

## 4.15 日志采集分析

### 4.15.1 课程界面

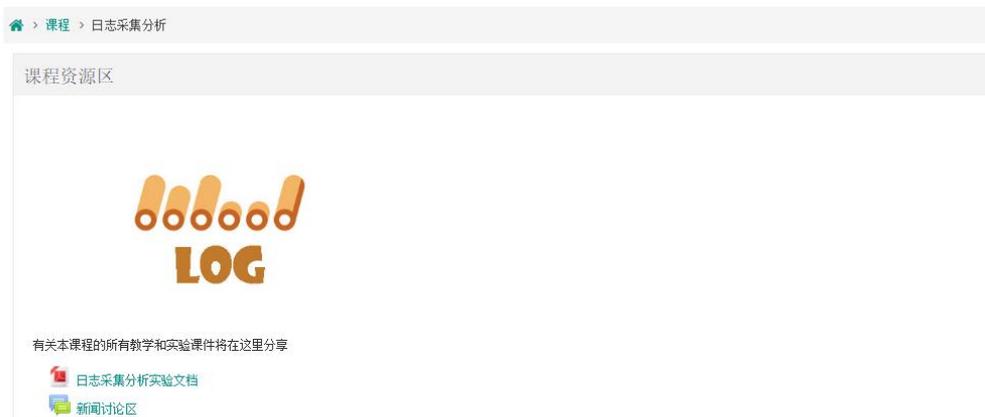


图4-15 日志采集分析

### 4.15.2 日志采集分析

网络设备、系统及服务程序等，在运作时都会产生一个叫log的事件记录；每一行日志都记载着日期、时间、使用者及动作等相关操作的描述。

Windows网络操作系统都设计有各种各样的日志文件，如应用程序日志，安全日志、系统日志、Scheduler服务日志、FTP日志、WWW日志、DNS服务器日志等等，这些根据你的系统开启的服务的不同而有所不同。我们在系统上进行一些操作时，这些日志文件通常会记录下我们操作的一些相关内容，这些内容对系统安全工作人员相当有用。比如说有人对系统进行了IPC探测，系统就会在安全日志里迅速地记下探测者探测时所用的IP、时间、用户名等，用FTP探测后，就会在FTP日志中记下IP、时间、探测所用的用户名等。

当然，日志数据对于实现网络安全的价值有多大取决于两个因素：第一，你的系统和设备必须进行合适的设置以便记录你需要的数据。第二，你必须有合适的工具、培训和可用的资源来分析收集到的数据。

当前的计算机病毒越来越复杂，对于网上求助这种远程的判断和分析来说，必须借助第三方的软件分析。目前流行的辅助分析工具有sawmill、HiJackthis 及

SREng。

通过专业的日志分析工具的完整性分析，让你从数据的海洋中摆脱出来，并可以直接的以WEB界面的形式查看所需信息，解除专业性烦恼。

## 4.16 用户画像与用户行为分析

### 4.16.1 课程界面

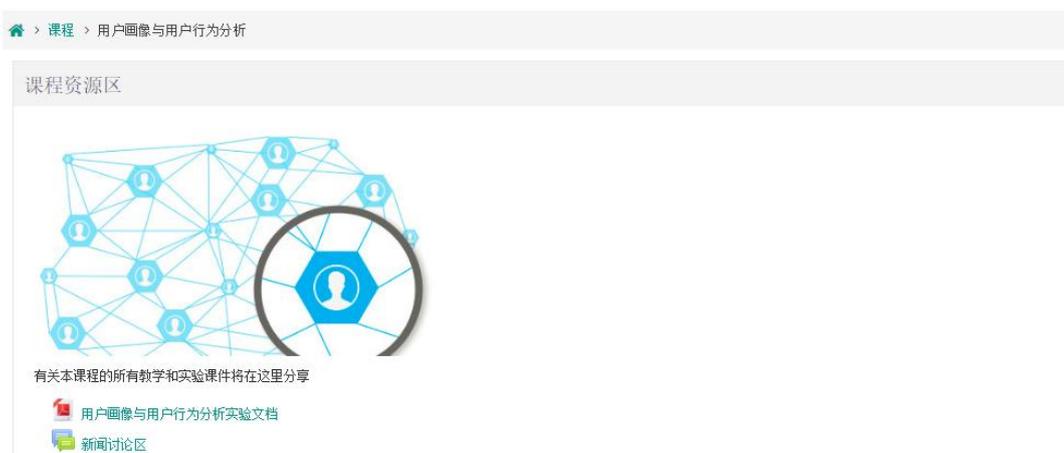


图4-16 用户画像与用户行为分析

### 4.16.2 用户画像与用户行为分析

简而言之，用户画像是根据用户社会属性、生活习惯和消费行为等信息而抽象出的一个标签化的用户模型。构建用户画像的核心工作即是给用户贴“标签”，而标签是通过对用户信息分析而来的高度精炼的特征标识。

举例来说，如果你经常购买一些玩偶玩具，那么电商网站即可根据玩具购买的情况替你打上标签“有孩子”，甚至还可以判断出你孩子大概的年龄，贴上“有5-10岁的孩子”这样更为具体的标签，而这些所有给你贴的标签统在一次，就成了你的用户画像，因此，也可以说用户画像就是判断一个人是什么样的人。

除去“标签化”，用户画像还具有的特点是“低交叉率”，当两组画像除了权重较小的标签外其余标签几乎一致，那就可以将二者合并，弱化低权重标签的异。

用户画像的作用。

罗振宇在《时间的朋友》跨年演讲上举了这样一个例子：当一个坏商家掌握了你的购买数据，他就可以根据你平常购买商品的偏好来决定是给你发正品还是假货以提高利润。且不说是不是存在这情况，但这也说明了利用用户画像可以做到“精准销售”，当然了，这是极其错误的用法。

其作用大体不离以下几个方面：

精准营销，分析产品潜在用户，针对特定群体利用短信邮件等方式进行营销；

用户统计，比如中国大学购买书籍人数 TOP10，全国分城市奶爸指数；

数据挖掘，构建智能推荐系统，利用关联规则计算，喜欢红酒的人通常喜欢什么运动品牌，利用聚类算法分析，喜欢红酒的人年龄段分布情况；

进行效果评估，完善产品运营，提升服务质量，其实这也就相当于市场调研、用户调研，迅速下定位服务群体，提供高水平的服务；

对服务或产品进行私人定制，即个性化的服务某类群体甚至每一位用户（个人认为这是目前的发展趋势，未来的消费主流）。比如，某公司想推出一款面向5—10岁儿童的玩具，通过用户画像进行分析，发现形象=“喜羊羊”、价格区间=“中等”的偏好比重最大，那么就给新产品提供类非常客观有效的决策依据。

业务经营分析以及竞争分析，影响企业发展战略。