第3章 大数据实验环境搭建 - 中文讲课脚本

Slide1：课程封面

大家好！欢迎来到我们的大数据课程。今天我们要学习第三章的内容，也就是大数据实验环境搭建。

我是李阳源老师，如果大家有什么问题的话，可以通过邮箱ywlee1212@gmail.com联系我，或者访问我的个人主页获取更多资料。

今天这节课对大家来说非常重要，因为我们要从零开始搭建一个完整的大数据实验环境。这个环境将是我们后续所有实验和项目的基础。

Slide2：课程大纲

好，我们先来看一下今天的学习内容。今天我们一共要学习五个主要部分：

首先是3.1，Linux系统的安装。为什么要先学Linux呢？因为大数据生态系统基本上都是在Linux环境下运行的，所以Linux是我们的基础。

接下来是3.2，Hadoop（哈杜普）的安装。Hadoop是大数据处理的核心框架，我们必须要掌握它的安装和配置。

然后是3.3，MySQL（迈艾思库艾尔）的安装。虽然我们主要学习大数据技术，但是传统的关系型数据库仍然很重要，特别是在数据的存储和管理方面。

第四部分是3.4，Kafka（卡夫卡）的安装。Kafka是实时数据流处理的重要工具，在现代大数据架构中扮演着关键角色。

最后是3.5，Anaconda（阿纳康达）的安装和使用方法。Anaconda是Python科学计算和数据分析的重要平台，我们后面的很多实验都会用到。

每个部分我都会带着大家一步一步地操作，确保大家都能跟上。

Slide3：Linux系统安装概述

现在我们开始第一部分，Linux系统的安装。

大家可以看到，Linux系统安装主要分为三个步骤：

第一步是3.1.1，下载安装文件。我们需要先获取Linux的安装镜像文件。

第二步是3.1.2，了解Linux系统的安装方式。我会给大家介绍几种不同的安装方法，帮助大家选择最适合自己的方式。

第三步是3.1.3，安装Linux虚拟机（Virtual Machine)。这是我们今天的重点，我会详细演示整个安装过程。

对于初学者来说，我强烈推荐使用虚拟机的方式来安装Linux。这样既安全又方便，不会影响大家现有的Windows系统。

Slide4：下载安装文件

好，现在我们来看第一个具体步骤，下载安装文件。

首先我要跟大家解释一下，为什么我们选择Ubuntu Kylin这个发行版。

大家都知道，Linux有很多不同的发行版，比如Ubuntu、CentOS、RedHat等等。我们选择Ubuntu是因为它对新手比较友好，安装和使用都相对简单。

而Ubuntu Kylin呢，是专门针对中国用户定制的版本。它有什么好处呢？首先，它内置了中文输入法，这样大家使用起来会更方便。其次，它根据中国用户的习惯优化了一些系统设置，界面也更符合我们的使用习惯。

关于版本选择，虽然最新的Ubuntu Kylin是22.04 LTS，但是我在实际测试中发现，这个版本对计算机资源的要求比较高。如果大家的电脑配置不是特别高的话，运行起来可能会比较慢。

所以我建议大家使用Ubuntu Kylin 16.04 LTS。这个版本虽然不是最新的，但是它运行稳定，资源消耗相对较少，而且完全能够满足我们大数据实验的需要。

Slide5：下载途径和版本选择

现在我来告诉大家具体怎么下载安装文件。

大家可以通过两种方式来下载Ubuntu Kylin。第一种方式是直接访问Ubuntu的官方网站，网址是https://www.ubuntu.org.cn/download/ubuntu-kylin。

进入下载页面后，大家会看到两个下载选项：32位版本和64位版本。

那么应该选择哪个版本呢？这主要取决于大家电脑的硬件配置。

如果你的电脑内存小于2GB，建议选择32位版本，因为32位系统对内存的要求比较低。

如果你的电脑内存大于4GB，那就选择64位版本。64位系统能够更好地利用大内存，性能也会更好。

大部分现代电脑都支持64位系统，所以我建议大家优先选择64位版本。

下载完成后，大家会得到一个ISO镜像文件，这个文件就是我们安装Linux系统需要的安装盘。

Slide6：Linux系统安装方式

在开始安装之前，我需要给大家介绍一下Linux系统的两种主要安装方式。

第一种是虚拟机安装方式。这种方式是在你现有的Windows系统上安装虚拟机软件，比如VirtualBox或者VMware，然后在虚拟机里面安装Linux系统。

这种方式有什么好处呢？首先，它不会影响你原来的Windows系统，两个系统是相互独立的。其次，如果Linux系统出现问题，你可以很容易地重新安装，不会影响到Windows。

但是这种方式也有缺点，就是性能会有一定的损失，因为需要同时运行两个操作系统。

第二种是双系统安装方式。这种方式是直接在电脑的硬盘上安装Linux系统，与Windows系统并存。

开机的时候，系统会让你选择要启动哪个操作系统。这种方式的好处是性能更好，因为每次只运行一个系统。

但是风险也比较大，如果操作不当，可能会影响到Windows系统，甚至导致数据丢失。

对于初学者，我强烈建议使用虚拟机的方式。等大家熟练掌握Linux之后，再考虑双系统安装。

Slide7：虚拟机软件选择

既然我们选择虚拟机方式，那么首先需要安装虚拟机软件。

市面上常用的虚拟机软件主要有两种：VMware和VirtualBox。

VirtualBox是开源免费的软件，功能也很强大。但是从易用性的角度来说，VMware要更胜一筹。VMware的界面更友好，操作更简单，稳定性也更好。

虽然VMware是收费软件，但是它提供了试用版本，对于我们的学习来说是完全够用的。

大家可以到VMware的官方网站下载，网址我已经放在Slide上了。也可以到我们课程的官方网站下载，在"下载专区"的"软件"目录下有VMware-workstation-full-17.0.1.exe这个文件。

下载完成后，大家需要在Windows系统中安装VMware。安装过程比较简单，基本上一路点击"下一步"就可以了。

Slide8：创建虚拟机

好，现在假设大家已经安装好了VMware，我们来开始创建虚拟机。

安装Linux系统主要分为两个大步骤：首先创建一个虚拟机，然后在虚拟机上安装Linux系统。

打开VMware软件，在主页选项卡中，大家可以看到几个选项。我们点击"创建新的虚拟机"。

大家看到屏幕上这个界面了吗？这就是VMware的主界面。中间有三个大按钮：创建新的虚拟机、打开虚拟机、连接远程服务器。

我们需要点击第一个，"创建新的虚拟机"。

Slide9：选择配置类型

点击之后，会弹出新建虚拟机向导。

在这个界面中，系统会问你希望使用什么类型的配置。这里有两个选项：

第一个是"典型（推荐）"，这是默认选择的选项。对于初学者来说，选择这个选项就足够了。

第二个是"自定义（高级）"，这个选项会提供更多的高级设置，但是对新手来说比较复杂。

我建议大家选择"典型（推荐）"，然后点击"下一步"。

这样VMware会使用默认的设置来创建虚拟机，既简单又不容易出错。

Slide10：选择安装来源

接下来这个步骤很重要，我们需要告诉VMware从哪里安装操作系统。

大家看到这个界面有三个选项：

第一个是"安装程序光盘"，如果你有Linux的安装光盘，可以选择这个选项。

第二个是"安装程序光盘镜像文件(iso)"，这就是我们要选择的选项。

第三个是"稍后安装操作系统"，这个选项会创建一个空的虚拟机，稍后再安装系统。

我们选择第二个选项，然后点击"浏览"按钮。

在弹出的文件浏览对话框中，我们需要找到之前下载的Ubuntu安装镜像文件，文件名是ubuntukylin-16.04-desktop-amd64.iso。

找到这个文件后，点击"打开"，然后点击"下一步"。

大家一定要确保选择了正确的ISO文件，否则后面的安装会失败。

Slide11：设置用户信息

现在我们来设置Linux系统的用户信息。这个步骤也很重要，因为这些信息将决定你登录Linux时使用的用户名和密码。

大家看到这个界面有几个字段需要填写：

"全名"这里，我建议大家填写"xmudblab"。这个名字只是一个显示名称，大家也可以根据自己的喜好来设置。

"用户名"这里填写"dblab"。这个用户名将是你登录Linux系统时使用的用户名，建议使用简单易记的名字。

"密码"和"确认"这两个字段，我建议设置为"123456"。当然，在实际的生产环境中，大家应该使用更复杂的密码来保证安全性。但是在学习阶段，使用简单的密码会更方便。

填写完成后，点击"下一步"。

请大家一定要记住这些用户名和密码，因为安装完成后需要用它们来登录系统。

Slide12：设置虚拟机名称

接下来我们需要给虚拟机起一个名字，并选择虚拟机文件的保存位置。

在"虚拟机名称"这个字段，我建议填写"hadoop01"。为什么叫这个名字呢？因为我们后面会在这个Linux系统上安装Hadoop，所以起这个名字比较有意义。

下面的"位置"字段显示的是虚拟机文件将要保存的路径。默认情况下，VMware会把虚拟机文件保存在用户文档目录下。如果大家想要更改保存位置，可以点击"浏览"按钮来选择其他目录。

我建议大家选择一个有足够空间的磁盘分区，因为虚拟机文件会比较大。

设置完成后，点击"下一步"。

Slide13：设置磁盘大小

现在我们来设置虚拟机的磁盘大小。这个设置非常重要，因为它决定了我们在Linux系统中可以使用多少存储空间。

大家看到默认的磁盘大小是100GB。对于我们的大数据实验来说，这个大小是比较合适的。

为什么需要这么大的空间呢？因为我们后面要安装很多大数据软件，比如Hadoop、Spark（斯帕克）、Kafka等等，这些软件加起来至少需要40GB的空间。

另外，我们还需要存储实验数据，处理大数据文件，所以总的来说，50GB到100GB的空间是比较合理的。

在下面的选项中，我们要选择"将虚拟磁盘存储为单个文件"。这样做的好处是管理起来比较方便，性能也会更好。

设置完成后，点击"下一步"。

不过大家要注意，虽然我们设置了100GB，但实际上VMware使用的是动态分配的方式，也就是说，只有当我们真正使用存储空间时，才会占用物理硬盘的空间。

Slide14：自定义硬件设置

在完成虚拟机创建之前，我们还需要对硬件进行一些调整，特别是内存设置。

大家看到这个界面显示了虚拟机的基本配置信息：名称是hadoop01，操作系统是Ubuntu 64位，硬盘是100GB，内存是4096MB，也就是4GB。

对于大数据实验来说，4GB的内存可能不够用。我建议大家点击"自定义硬件"按钮来调整内存大小。

为什么内存很重要呢？因为大数据处理通常需要在内存中处理大量数据，内存越大，处理速度就越快。

特别是当我们运行Hadoop、Spark这些软件时，它们对内存的需求比较高。

所以我们需要根据自己电脑的实际情况来调整虚拟机的内存大小。

Slide15：调整内存大小

现在我们来调整虚拟机的内存大小。

在硬件配置界面中，大家可以看到左边有一个硬件列表，我们需要选择"内存"这一项。

在右边的设置面板中，有一个内存大小的滑动条。大家可以根据自己电脑的实际内存大小来调整。

如果你的电脑有32GB内存，那么可以给虚拟机分配16GB内存。如果你的电脑有16GB内存，可以分配8GB给虚拟机。

但是无论如何，我建议至少分配4GB内存给虚拟机，否则运行大数据软件时可能会很慢。

需要注意的是，分配给虚拟机的内存越多，留给Windows系统的内存就越少。所以大家要在两者之间找到一个平衡。

一般来说，给虚拟机分配总内存的50%左右是比较合适的。

调整完成后，点击"关闭"按钮，然后在之前的界面中点击"完成"。

这时候，VMware就会开始自动安装Ubuntu系统了。

Slide16：系统安装完成

好，经过一段时间的等待，Ubuntu系统安装完成了。

大家看到屏幕上出现了Ubuntu的登录界面。这个蓝紫色背景的界面就是Ubuntu Kylin的典型界面风格。

在登录界面中，我们可以看到之前创建的用户"xmudblab"。点击这个用户图标，然后输入密码"123456"，就可以登录到Ubuntu系统了。

登录成功后，大家会看到Ubuntu Kylin的桌面环境。这个桌面环境叫做UKUI，是专门为Ubuntu Kylin设计的。

桌面的风格和Windows有些相似，这样可以帮助大家更快地适应Linux环境。

在桌面的底部，有一排应用程序图标，这就是任务栏。大家可以通过点击这些图标来启动不同的应用程序。

到这里，我们的Linux系统就安装完成了。接下来我会给大家介绍一些基本的使用方法。

Slide17：终端的使用

在Linux系统中，有一个非常重要的工具，就是终端（Terminal)。

大家看到桌面底部任务栏中有一个黑色的图标，这就是终端按钮。点击这个按钮，就可以打开一个新的终端窗口。

终端是什么呢？简单来说，终端就是Linux的命令行界面，也叫做Shell环境。在终端中，我们可以通过输入命令来控制系统，完成各种操作。

对于学习大数据技术来说，熟练使用终端是非常重要的。因为很多大数据软件的安装、配置和运行都需要通过命令行来完成。

除了点击图标，大家还可以使用快捷键来打开终端。按住Ctrl+Alt+T这三个键，就可以快速打开一个新的终端窗口。

我建议大家现在就试一试，点击终端图标或者使用快捷键来打开终端，熟悉一下这个界面。

在后面的课程中，我们会大量使用终端来进行各种操作，所以大家一定要熟练掌握。

Slide18：文件管理器

除了终端，另一个经常使用的工具是文件管理器。

大家看到任务栏中有一个文件夹图标，这就是文件管理器的启动按钮。点击这个按钮，就可以打开文件管理界面。

文件管理器的作用就像Windows中的"我的电脑"或者"文件资源管理器"一样，可以帮助我们浏览和管理文件系统中的文件和文件夹。

VMware有一个很棒的功能，就是支持Windows系统和Linux虚拟机之间的双向复制。这意味着什么呢？

你可以在Windows中复制一个文件，然后直接粘贴到Linux中。反过来也是一样，你可以把Linux中的文件复制到Windows中。

这个功能对我们的学习和实验非常有用。比如，你可以在Windows中下载一些数据文件，然后直接复制到Linux中进行处理。

大家可以试试这个功能，在两个系统之间复制一些文件，体验一下这种便利性。

Slide19：文件管理器界面详解

现在我们来详细看一下文件管理器的界面。

大家可以看到，文件管理器的界面分为两个部分：

左边是导航面板，显示了各种常用的位置，比如：

- Home：用户的主目录
- Desktop：桌面
- Documents：文档文件夹
- Downloads：下载文件夹
- Pictures：图片文件夹
- Music：音乐文件夹
- Videos：视频文件夹

右边是文件和文件夹的显示区域，以图标的形式显示各个文件夹。

这个界面和Windows的文件资源管理器很相似，所以大家应该很容易上手。

在Linux中，用户的文件通常存储在/home/用户名这个目录下。比如我们的用户名是dblab，那么个人文件就存储在/home/dblab目录下。

大家可以在这个文件管理器中创建新的文件夹，用来存储我们后面实验需要的文件和数据。

我建议大家创建一个专门的文件夹，比如叫"bigdata"，用来存储所有与大数据相关的文件。

Slide20：系统关机和注销

最后，我来给大家介绍一下如何正确地关机和注销。

在Ubuntu系统中，关机和注销的操作和Windows有些不同。

大家看到桌面右上角有一个齿轮图标，这是系统菜单按钮。点击这个按钮，会弹出一个菜单。

在这个菜单中，有几个重要的选项：

- About This Computer：显示系统信息
- System Settings：系统设置
- Lock：锁定屏幕
- Log Out：注销当前用户
- Suspend：挂起系统
- Shut Down：关机

如果你想暂时离开，但不想关闭系统，可以选择"Lock"来锁定屏幕。

如果你想切换到其他用户，可以选择"Log Out"来注销当前用户。

如果你完成了所有工作，想要关闭系统，就选择"Shut Down"。

需要特别注意的是，这里的"Shut Down"只是关闭Ubuntu系统，而不是关闭整个计算机。关闭Ubuntu后，你还是会回到Windows系统。

如果你想完全关闭计算机，需要先关闭Ubuntu，然后再在Windows中关机。

课程小结

好，今天我们完成了Linux系统的安装和基本使用介绍。

让我来总结一下我们今天学到的内容：

1. 我们了解了为什么要使用Ubuntu Kylin，以及如何下载安装文件
2. 我们学习了虚拟机安装方式和双系统安装方式的区别
3. 我们详细演示了使用VMware创建虚拟机和安装Ubuntu的整个过程
4. 我们介绍了Ubuntu系统的基本使用方法，包括终端、文件管理器等重要工具

这个Linux环境将是我们后续所有大数据实验的基础。在下节课中，我们将在这个环境上安装Hadoop，开始我们的大数据之旅。

大家回去后一定要多练习，熟悉Linux的基本操作。特别是终端的使用，这对我们后面的学习非常重要。

如果在安装过程中遇到任何问题，可以随时联系我或者查看课程网站上的资料。

今天的课程就到这里，谢谢大家！

3.2 Hadoop的安装 - 教学讲义脚本

开场白

同学们好！今天我们要学习的是Hadoop(哈杜普)的安装。这是一个非常重要的章节，因为Hadoop是大数据处理的基础平台。我们会从最基本的概念开始，一步一步地带大家完成Hadoop的安装和配置。

【Slide1】3.2 Hadoop的安装 - 目录

大家看一下今天的学习内容。我们一共要学习7个部分：

首先是Hadoop简介，让大家了解什么是Hadoop；然后是安装前的准备工作，这个很重要，如果准备工作没做好，后面的安装就会出问题；接下来我们会学习Hadoop的三种安装模式；然后是下载安装文件；最后我们会实际操作三种不同的配置模式：单机模式、伪分布式模式和分布式模式。

好，我们开始第一部分。

【Slide2】3.2.1 Hadoop简介

首先，什么是Hadoop呢？

Hadoop是Apache软件基金会旗下的一个开源分布式计算平台(Open Source Distributed Computing Platform)。大家注意，这里有几个关键词：第一个是"开源"，意思是我们可以免费使用；第二个是"分布式"，意思是可以在多台计算机上运行；第三个是"计算平台"，意思是它提供了一整套的大数据处理解决方案。

Hadoop为用户提供了系统底层细节透明的分布式基础架构。什么叫"透明"呢？就是说用户不需要关心数据具体存储在哪台机器上，也不需要关心计算任务在哪台机器上执行，Hadoop会自动帮我们处理这些复杂的问题。

Hadoop是基于Java语言开发的，所以具有很好的跨平台特性。无论是Windows、Linux还是Mac，都可以运行Hadoop。而且可以部署在廉价的计算机集群中，这是Hadoop的一个重要优势。

Hadoop的核心是什么呢？主要有两个部分：第一个是分布式文件系统HDFS(Hadoop Distrib

uted File System)，第二个是MapReduce。HDFS负责存储，MapReduce负责计算。

【Slide3】Hadoop版本和商业发行版

现在我们来看看Hadoop的版本历史。

Apache Hadoop版本分为三代：Hadoop 1.0、Hadoop 2.0和Hadoop 3.0。每一代都有重要的改进和新功能。

除了免费开源的Apache Hadoop以外，还有一些商业公司推出了Hadoop的发行版。大家可能听说过Cloudera、Hortonworks这些公司。2008年，Cloudera成为第一个Hadoop商业化公司，2009年推出了第一个Hadoop发行版。后来很多大公司也加入了，比如MapR、星环等。

有趣的是，2018年10月，Cloudera和Hortonworks宣布合并，这说明Hadoop市场的竞争非常激烈。

商业发行版虽然也是基于Apache Hadoop，但是它们通常有更好的易用性、更多的功能以及更高的性能。不过对于我们学习来说，使用免费的Apache Hadoop就足够了。

【Slide4】3.2.2 安装Hadoop前的准备工作 - 创建hadoop用户

好，现在我们开始准备安装Hadoop。第一步是创建一个专门的用户。

在这个教程中，我们需要创建一个名称为hadoop的普通用户。为什么要创建专门的用户呢？这是为了安全考虑，也是为了管理方便。后续所有操作都会使用这个用户名登录到Linux系统。

首先，我们使用已有的用户（比如xmudblab用户）登录Linux系统，然后打开一个终端。大家可以使用快捷键Ctrl+Alt+T来打开终端。

现在我们执行这个命令：

sudo useradd -m hadoop -s /bin/bash

让我解释一下这个命令的参数：

- sudo表示以管理员权限执行
- useradd是添加用户的命令

- -m表示为用户创建主目录
- hadoop是我们要创建的用户名
- -s /bin/bash表示指定这个用户使用bash作为默认shell

这条命令执行后，就创建了一个可以登录的hadoop用户。

接下来我们要为hadoop用户设置密码：

sudo passwd hadoop

系统会提示你输入密码，大家输入一个容易记住的密码就可以了。

然后，我们给hadoop用户增加管理员权限：

sudo adduser hadoop sudo

这样做是为了方便部署，避免一些权限问题。

【Slide5】更新APT

第二步是更新APT软件包管理器。

为了确保Hadoop安装过程顺利进行，我们建议先更新一下系统的软件包列表：

sudo apt-get update

这个命令会从软件仓库下载最新的软件包信息，确保我们安装的软件都是最新版本。这个过程可能需要几分钟时间，大家耐心等待一下。

【Slide6】安装SSH

第三步是安装SSH服务。

SSH(Secure Shell)是什么呢？它是一种网络协议，用于在网络上安全地远程登录和执行命令。Hadoop集群中的机器需要通过SSH相互通信，所以SSH是必不可少的。

Ubuntu默认已经安装了SSH客户端，但是我们还需要安装SSH服务端。在Linux终端中执行以下命令：

sudo apt-get install openssh-server

安装完成后，我们可以测试一下SSH是否工作正常：

ssh localhost

这个命令是让我们通过SSH连接到本机。第一次运行时会出现一个安全提示，问你是否确定要连接。你输入"yes"，然后按提示输入密码hadoop，就可以登录到本机了。

大家看到界面上显示的信息了吗？这表示SSH服务正常工作。

【Slide7】配置SSH免密码登录

现在我们要配置SSH免密码登录。为什么要这样做呢？因为Hadoop集群中的机器需要频繁地相互通信，如果每次都要输入密码，会很麻烦。

首先，我们退出刚才的SSH连接：

exit

然后，我们使用ssh-keygen生成密钥对：

cd ~/.ssh/

ssh-keygen -t rsa

注意，如果没有.ssh目录，请先执行一次ssh localhost。

执行ssh-keygen命令时，会有一些提示信息，大家都按回车键就可以了，使用默认设置。

然后，我们把公钥添加到授权文件中：

cat ./id_rsa.pub >> ./authorized_keys

现在再执行ssh localhost命令，应该就不需要输入密码了。大家可以试试看。

成功的话，大家会看到欢迎信息，这说明免密码登录配置成功了。

【Slide8】安装Java环境 - 创建JVM目录

第四步是安装Java环境。这是非常重要的一步，因为Hadoop是用Java开发的，没有Java环境就无法运行Hadoop。

首先，我们创建一个目录来存放JDK文件：

```
cd /usr/lib
sudo mkdir jvm
```

这里我们在/usr/lib下创建了一个jvm目录，这是存放Java虚拟机的标准位置。

接下来，我们要解压JDK安装文件。假设大家已经下载了JDK安装包到Downloads目录：

```
cd ~
cd Downloads
sudo tar -zxvf ./jdk-8u371-linux-x64.tar.gz -C /usr/lib/jvm
```

这个命令会把JDK压缩包解压到我们刚才创建的jvm目录中。

【Slide9】设置Java环境变量

解压完成后，我们需要设置环境变量。环境变量是什么呢？简单来说，就是告诉系统在哪里可以找到Java命令。

我们使用vim编辑器打开环境变量配置文件：

```
vim ~/.bashrc
```

在这个文件的开头位置，添加以下几行内容：

```
export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_371
export JRE_HOME=${JAVA_HOME}/jre
```

export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib

export PATH=${JAVA_HOME}/bin:$PATH

让我解释一下这些环境变量：

- JAVA_HOME指向JDK的安装目录
- JRE_HOME指向Java运行时环境的目录
- CLASSPATH告诉Java在哪里查找类文件
- PATH告诉系统在哪里查找可执行文件

【Slide10】验证Java安装

设置完环境变量后，我们需要让配置立即生效：

source ~/.bashrc

现在我们可以测试Java是否安装成功：

java -version

如果安装成功，你会看到类似这样的输出：

java version "1.8.0_371"

Java(TM) SE Runtime Environment (build 1.8.0_371-b12)

Java HotSpot(TM) 64-Bit Server VM (build 25.162-b12, mixed mode)

看到这个信息，说明Java环境配置成功了！

【Slide11】3.2.3 安装Hadoop的三种模式

现在我们来了解Hadoop的三种安装模式。理解这三种模式很重要，因为它们适用于不同的场景。

第一种是单机模式：只在一台机器上运行，存储采用本地文件系统，没有采用分布式文件系统HDFS。这种模式主要用于开发和测试。

第二种是伪分布式模式：存储采用分布式文件系统HDFS，但是HDFS的名称节点(NameNode)和数据节点(DataNode)都在同一台机器上。这种模式可以让我们在单台机器上体验Hadoop的分布式特性。

第三种是分布式模式：存储采用分布式文件系统HDFS，而且HDFS的名称节点和数据节点位于不同机器上。这是真正的生产环境模式。

我们今天会学习所有这三种模式的配置方法。

【Slide12】3.2.4 下载Hadoop安装文件

现在我们开始下载和准备Hadoop安装文件。

本教程采用的Hadoop版本是3.3.5，这是一个比较稳定的版本。大家可以到Hadoop官网下载安装文件。官网地址是：https://archive.apache.org/dist/hadoop/core/hadoop-3.3.5

下载完成后，我们来解压和安装：

```
sudo tar -zxf ~/Downloads/hadoop-3.3.5.tar.gz -C /usr/local
cd /usr/local/
sudo mv ./hadoop-3.3.5/ ./hadoop
sudo chown -R hadoop ./hadoop
```

让我解释一下这几个命令：

- 第一个命令把下载的压缩包解压到/usr/local目录
- 第二个命令进入/usr/local目录
- 第三个命令把文件夹名改为hadoop，这样比较简洁
- 第四个命令修改文件权限，让hadoop用户可以访问这些文件

Hadoop解压后就可以使用了。我们可以验证一下：

```
cd /usr/local/hadoop
./bin/hadoop version
```

如果能看到版本信息，说明Hadoop安装成功了。

【Slide13】3.2.5 单机模式配置

现在我们开始配置第一种模式：单机模式。

Hadoop默认就是非分布式模式，也就是本地模式，无需进行其他配置即可运行。这是最简单的模式。

Hadoop附带了丰富的例子程序，我们可以看看都有哪些例子：

cd /usr/local/hadoop

./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.5.jar

执行这个命令后，会显示所有可用的示例程序，比如wordcount、pi、grep等等。

【Slide14】运行单机模式示例

现在我们来运行一个实际的例子。我选择grep程序，这是一个文本搜索程序。

首先，我们准备输入数据：

cd /usr/local/hadoop

mkdir input

cp ./etc/hadoop/*.xml ./input

这里我们把Hadoop的配置文件复制到input目录作为测试数据。

然后运行grep程序：

./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar grep ./input ./output 'dfs[a-z.]+'

这个命令会在输入文件中搜索匹配'dfs[a-z.]+'这个正则表达式的内容。

运行完成后，我们可以查看结果：

cat ./output/*

大家会看到搜索结果，比如"dfsadmin"这样的词。

注意：Hadoop默认不会覆盖结果文件，所以如果要再次运行，需要先删除output目录：

rm -r ./output

【Slide15】3.2.6 伪分布式模式配置 - 修改配置文件

现在我们学习第二种模式：伪分布式模式。这种模式会使用HDFS文件系统，但所有组件都运行在同一台机器上。

首先我们需要修改配置文件。Hadoop的配置文件都在etc/hadoop目录下。

第一个要修改的是core-site.xml文件。修改后的内容如下：

<configuration>

<property>

<n>hadoop.tmp.dir</n>

<value>file:/usr/local/hadoop/tmp</value>

<description>Abase for other temporary directories.</description>

</property>

<property>

<n>fs.defaultFS</n>

<value>hdfs://localhost:9000</value>

</property>

</configuration>

这里我们设置了两个重要参数：

- hadoop.tmp.dir指定了Hadoop的临时目录
- fs.defaultFS指定了默认的文件系统，这里我们使用HDFS

【Slide16】修改hdfs-site.xml文件

接下来我们修改hdfs-site.xml文件：

<configuration>

<property>

<n>dfs.replication</n>

<value>1</value>

</property>

<property>

<n>dfs.namenode.name.dir</n>

<value>file:/usr/local/hadoop/tmp/dfs/name</value>

</property>

<property>

<n>dfs.datanode.data.dir</n>

<value>file:/usr/local/hadoop/tmp/dfs/data</value>

</property>

</configuration>

这里设置了三个参数：

- dfs.replication设置为1，因为我们只有一台机器
- dfs.namenode.name.dir指定名称节点的数据目录
- dfs.datanode.data.dir指定数据节点的数据目录

【Slide17】执行名称节点格式化

配置文件修改完成后，我们需要格式化名称节点。这是一个重要步骤，只在第一次安装时执行。

cd /usr/local/hadoop

./bin/hdfs namenode -format

这个命令会初始化HDFS文件系统。如果格式化成功，你会看到"successfully formatted"的提示信息。

大家要注意，这个操作只能在第一次安装时执行，以后不要再次格式化，否则会丢失HDFS中的所有数据！

【Slide18】启动Hadoop

现在我们可以启动Hadoop了：

cd /usr/local/hadoop

./sbin/start-dfs.sh

启动过程中可能会出现SSH提示，输入"yes"即可。

启动完成后，我们可以使用jps命令查看Java进程：

jps

如果启动成功，你应该看到这些进程：

- DataNode：数据节点
- NameNode：名称节点
- SecondaryNameNode：辅助名称节点
- Jps：jps命令本身

如果缺少任何一个进程，说明启动失败了。

【Slide19】使用Web界面查看HDFS信息

Hadoop提供了Web界面来监控系统状态。我们可以在浏览器中访问：

http://localhost:9870/

这个页面会显示HDFS的详细信息，包括：

- 集群概览
- 数据节点状态
- 文件系统统计信息

这是一个非常有用的监控工具，大家要记住这个地址。

【Slide20】运行伪分布式实例

现在我们来运行伪分布式模式的例子。

首先，我们需要在HDFS中创建用户目录：

```
cd /usr/local/hadoop
./bin/hdfs dfs -mkdir -p /user/hadoop
```

注意这里使用的是hdfs dfs命令，这是HDFS的文件系统命令。

然后创建输入目录并上传文件：

```
./bin/hdfs dfs -mkdir input
./bin/hdfs dfs -put ./etc/hadoop/*.xml input
```

这里我们把配置文件上传到HDFS中作为测试数据。

【Slide21】运行MapReduce作业

现在我们可以运行MapReduce作业了：

```
./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.5.jar grep input output 'dfs[a-z.]+'
```

这次程序运行在HDFS上，你会看到更详细的执行日志，包括Map和Reduce任务的进度。

运行结束后，查看HDFS中的结果：

./bin/hdfs dfs -cat output/*

结果应该和单机模式类似，但这次是真正在分布式文件系统上运行的。

【Slide22】关闭Hadoop和配置PATH变量

使用完毕后，我们可以关闭Hadoop：

cd /usr/local/hadoop

./sbin/stop-dfs.sh

为了使用方便，我们可以配置PATH环境变量。编辑~/.bashrc文件，添加：

export PATH=$PATH:/usr/local/hadoop/sbin:/usr/local/hadoop/bin

然后执行：

source ~/.bashrc

这样以后我们就可以在任意目录直接使用hadoop命令了。

【Slide23】3.2.7 分布式模式配置 - 集群规划

最后我们学习最复杂也是最实用的分布式模式。这种模式需要多台机器。

在我们的例子中，我们使用三个节点来搭建集群环境，主机名分别为hadoop01、hadoop02和hadoop03。

让我们看看组件分布：

hadoop01节点（主节点）：

- HDFS：NameNode + DataNode
- YARN：ResourceManager + NodeManager

hadoop02节点（从节点）：

- HDFS：DataNode
- YARN：NodeManager

hadoop03节点（从节点）：

- HDFS：DataNode + SecondaryNameNode
- YARN：NodeManager

这是一个典型的小型Hadoop集群配置。

【Slide24】安装虚拟机

要搭建分布式集群，我们需要准备三台虚拟机。

如果你已经有了hadoop01虚拟机，可以通过"克隆"的方式快速创建hadoop02和hadoop03。由于hadoop02和hadoop03是从节点，不需要安装很多软件，所以配置可以低一些：

- 内存：4GB
- 磁盘：20GB

对于每台虚拟机，都需要：

1. 创建hadoop用户
2. 安装SSH服务端
3. 安装Java环境

这些步骤和前面讲的单机安装是一样的。

【Slide25-26】网络配置 - 修改主机名

现在我们要配置网络。首先修改每台机器的主机名。

在hadoop01上执行：

sudo vim /etc/hostname

把内容改为"hadoop01"，注意区分大小写。

同样地，把hadoop02和hadoop03的主机名分别改为"hadoop02"和"hadoop03"。

修改完成后需要重启Linux系统才能生效。

接下来查看每台机器的IP地址：

ifconfig

假设我们得到的IP地址是：

- hadoop01：192.168.91.128
- hadoop02：192.168.91.129
- hadoop03：192.168.91.130

【Slide27-33】配置静态IP地址

为了集群稳定运行，我们需要设置静态IP地址。动态IP地址会在重启后改变，导致集群无法正常通信。

首先查看网关地址：

netstat -nr

假设网关是192.168.91.2。

然后在Ubuntu系统中：

1. 点击右上角的设置图标
2. 选择"System Settings"
3. 点击"Network"
4. 选择网络连接，点击"Options"
5. 在IPv4设置中选择"Manual"
6. 设置IP地址、子网掩码、网关和DNS

具体设置：

- Address：192.168.91.128（对应每台机器）
- Netmask：255.255.255.0
- Gateway：192.168.91.2
- DNS servers：114.114.114.114

设置完成后重启系统，并测试网络连接是否正常。

【Slide34-36】配置hosts文件

接下来配置hosts文件，建立IP地址和主机名的映射关系。

在每台机器上编辑/etc/hosts文件：

sudo vim /etc/hosts

添加以下内容：

192.168.91.128 hadoop01

192.168.91.129 hadoop02

192.168.91.130 hadoop03

注意：确保hosts文件中只有一个127.0.0.1条目，并且不能有"127.0.0.1 hadoop01"这样的映射。

修改完成后重启系统，然后测试网络连通性：

ping hadoop01 -c 3

ping hadoop02 -c 3

ping hadoop03 -c 3

每台机器都应该能ping通其他所有机器。

【Slide37-41】配置SSH免密码登录

对于分布式集群，hadoop01（主节点）需要能够免密码SSH登录到所有节点。

首先确保所有节点都安装了SSH服务端：

```
sudo apt-get install openssh-server
```

在hadoop01上生成SSH密钥：

```
cd ~/.ssh
rm ./id_rsa*                    # 删除旧的密钥（如果存在）
ssh-keygen -t rsa               # 生成新密钥，所有提示都按回车
```

配置hadoop01到自己的免密码登录：

```
cat ./id_rsa.pub >> ./authorized_keys
```

将公钥复制到其他节点：

```
scp ~/.ssh/id_rsa.pub hadoop@hadoop02:/home/hadoop/
scp ~/.ssh/id_rsa.pub hadoop@hadoop03:/home/hadoop/
```

在hadoop02和hadoop03上配置授权：

```
mkdir ~/.ssh                    # 如果目录不存在
cat ~/id_rsa.pub >> ~/.ssh/authorized_keys
rm ~/id_rsa.pub                 # 清理临时文件
```

测试免密码登录：

```
ssh hadoop02
ssh hadoop03
```

应该都能直接登录，不需要输入密码。

【Slide42-44】下载安装文件和配置PATH

在hadoop01上下载并解压Hadoop：

```
sudo tar -zxvf ~/Downloads/hadoop-3.3.5.tar.gz -C /usr/local
cd /usr/local/
sudo mv ./hadoop-3.3.5/ ./hadoop
sudo chown -R hadoop:hadoop ./hadoop
```

配置PATH环境变量，编辑~/.bashrc：

```
export PATH=$PATH:/usr/local/hadoop/bin:/usr/local/hadoop/sbin
```

然后执行：

```
source ~/.bashrc
```

这样就可以在任意目录使用hadoop命令了。

【Slide45-52】配置集群环境

现在开始配置分布式环境。我们需要修改5个配置文件：

1. 修改workers文件

删除localhost，添加所有数据节点：

hadoop01

hadoop02

hadoop03

2. 修改core-site.xml

<configuration>

<property>

<n>fs.defaultFS</n>

<value>hdfs://hadoop01:9000</value>

</property>

<property>

<n>hadoop.tmp.dir</n>

<value>file:/usr/local/hadoop/tmp</value>

<description>Abase for other temporary directories.</description>

</property>

</configuration>

3. 修改hdfs-site.xml

<configuration>

<property>

<n>dfs.namenode.secondary.http-address</n>

<value>hadoop03:50090</value>

</property>

<property>

<n>dfs.replication</n>

<value>3</value>

</property>

<property>

<n>dfs.namenode.name.dir</n>

<value>file:/usr/local/hadoop/tmp/dfs/name</value>

</property>

<property>

<n>dfs.datanode.data.dir</n>

<value>file:/usr/local/hadoop/tmp/dfs/data</value>

</property>

</configuration>

注意这里dfs.replication设置为3，表示每个数据块保存3个副本。

4. 修改mapred-site.xml

<configuration>

<property>

<n>mapreduce.framework.name</n>

<value>yarn</value>

</property>

<property>

<n>mapreduce.jobhistory.address</n>

<value>hadoop01:10020</value>

</property>

<property>

<n>mapreduce.jobhistory.webapp.address</n>

<value>hadoop01:19888</value>

</property>

<property>

<n>yarn.app.mapreduce.am.env</n>

<value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>

</property>

<property>

<n>mapreduce.map.env</n>

<value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>

</property>

<property>

<n>mapreduce.reduce.env</n>

<value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>

</property>

</configuration>

5. 修改yarn-site.xml

<configuration>

<property>

<n>yarn.resourcemanager.hostname</n>

<value>hadoop01</value>

</property>

<property>

<n>yarn.nodemanager.aux-services</n>

<value>mapreduce_shuffle</value>

</property>

</configuration>

这些配置文件设置了集群的基本参数，指定了各个组件的位置和端口。

【Slide53-54】分发配置文件和格式化

配置文件修改完成后，我们需要把hadoop01的配置分发到其他节点。

首先清理临时文件：

```
cd /usr/local/hadoop
sudo rm -r ./tmp                    # 删除临时文件
sudo rm -r ./logs/*                 # 删除日志文件
```

然后打包并分发：

```
cd /usr/local
tar -zcf ~/hadoop.master.tar.gz ./hadoop
cd ~
scp ./hadoop.master.tar.gz hadoop02:/home/hadoop
scp ./hadoop.master.tar.gz hadoop03:/home/hadoop
```

在hadoop02和hadoop03上解压：

```
cd ~
sudo rm -r /usr/local/hadoop      # 删除旧的安装（如果存在）
sudo tar -zxf ~/hadoop.master.tar.gz -C /usr/local
sudo chown -R hadoop /usr/local/hadoop
```

最后，在hadoop01上格式化名称节点：

```
cd /usr/local/hadoop
./bin/hdfs namenode -format
```

注意：这个操作只在第一次安装时执行！

【Slide55-56】启动Hadoop集群

现在我们可以启动Hadoop集群了。所有启动命令都在hadoop01（主节点）上执行：

```
cd /usr/local/hadoop
./sbin/start-dfs.sh                    # 启动HDFS服务
./sbin/start-yarn.sh                   # 启动YARN服务
./sbin/mr-jobhistory-daemon.sh start historyserver  # 启动作业历史服务器
```

启动完成后，我们用jps命令检查各个节点的进程：

在hadoop01上应该看到：

- NameNode
- ResourceManager
- NodeManager
- JobHistoryServer
- DataNode
- Jps

在hadoop02上应该看到：

- DataNode
- NodeManager
- Jps

在hadoop03上应该看到：

- DataNode
- SecondaryNameNode
- NodeManager
- Jps

如果缺少任何进程，说明启动有问题。

我们还可以用这个命令检查集群状态：

./bin/hdfs dfsadmin -report

如果看到"Live datanodes (3)"，说明三个数据节点都正常运行。

【Slide57-58】Web界面监控

Hadoop提供了丰富的Web监控界面：

HDFS监控界面： http://hadoop01:9870/

这个页面显示HDFS文件系统的状态，包括：

- 集群概览信息
- 数据节点状态
- 文件系统使用情况

YARN监控界面： http://hadoop01:8088/

这个页面显示资源管理和作业执行情况，包括：

- 集群资源使用情况
- 正在运行的应用程序
- 作业历史记录

大家一定要记住这两个地址，它们是监控集群的重要工具。

【Slide59-62】执行分布式实例

现在我们来运行一个真正的分布式MapReduce作业。

首先创建HDFS用户目录：

hdfs dfs -mkdir -p /user/hadoop

注意这里我们直接使用hdfs命令，因为已经配置了PATH环境变量。

创建输入目录并上传测试数据：

hdfs dfs -mkdir input

hdfs dfs -put /usr/local/hadoop/etc/hadoop/*.xml input

运行分布式grep作业：

hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.5.jar grep input output 'dfs[a-z.]+'

这次作业会在三个节点上并行执行，你会看到详细的执行日志，包括：

- Map任务的分配和执行
- Reduce任务的执行
- 数据在节点间的传输

在执行过程中，大家可以访问YARN的Web界面查看作业进度。点击"ApplicationMaster"链接可以看到更详细的执行信息。

执行完成后查看结果：

hdfs dfs -cat output/*

结果应该包含搜索到的匹配项，比如：

- dfsadmin
- dfs.replication
- dfs.namenode.name.dir
- dfs.datanode.data.dir

【Slide63】关闭Hadoop集群

使用完毕后，我们需要正确关闭集群。关闭命令也在hadoop01上执行：

```
stop-yarn.sh                                    # 停止YARN服务
stop-dfs.sh                                     # 停止HDFS服务
mr-jobhistory-daemon.sh stop historyserver  # 停止作业历史服务器
```

关闭顺序很重要：先停止YARN，再停止HDFS，最后停止历史服务器。

总结

好的，同学们，今天我们学习了Hadoop的完整安装过程。让我总结一下要点：

主要内容回顾：

1. **Hadoop简介**：了解了Hadoop是什么，它的核心组件HDFS和MapReduce
2. **安装准备**：创建用户、安装SSH、配置Java环境
3. **三种模式**：
    - 单机模式：适合开发测试
    - 伪分布式模式：单机体验分布式特性
    - 分布式模式：生产环境的真正集群

重要经验：

1. **配置的重要性**：配置文件决定了集群的行为
2. 网**络的关键性**：SSH免密码登录是集群通信的基础
3. **监控的必要性**：Web界面帮助我们了解集群状态
4. **操作的规范性**：启动和关闭都有固定的顺序

实际应用建议：

1. 开发环境使用伪分布式模式
2. 生产环境使用分布式模式
3. 定期检查集群健康状态
4. 做好数据备份

今天的内容比较多，建议大家课后多练习，特别是分布式模式的配置。下次课我们将学习如何在Hadoop上进行实际的大数据处理。

有什么问题吗？

MySQL与Kafka安装配置讲课脚本

슬라이드 1: 课程内容概览

大家好！今天我们要学习的是大数据环境搭建的重要组成部分 - MySQL和Kafka的安装与配置。

现在请大家看一下今天的学习内容。我们主要分为两大部分：

首先是MySQL的安装，包括四个步骤：

- 执行安装命令
- 启动MySQL服务（MySQL Service）
- 进入MySQL Shell界面
- 解决MySQL出现的中文乱码问题

然后是Kafka的安装，包括：

- Kafka简介
- Kafka的安装和使用

这些都是我们后面学习Spark（Apache Spark）和大数据处理必备的基础环境。大家准备好了吗？我们开始吧！

슬라이드 2: MySQL安装命令执行

好，现在我们开始第一部分 - MySQL的安装。

请大家注意，在安装MySQL之前，我们必须要做一个重要的步骤，就是更新软件源。为什么要这样做呢？因为我们要确保能够获得最新版本的MySQL。

请跟我一起输入这个命令：

sudo apt-get update

大家看到了吗？这个命令会更新我们系统的软件包列表。等这个命令执行完成后，我们再执行MySQL的安装命令：

sudo apt-get install mysql-server

这个命令会自动下载并安装MySQL服务器。安装过程中系统可能会要求你设置root用户的密

码，请大家记住自己设置的密码，我们后面会用到。

슬라이드 3: MySQL服务启动

安装完成后，我们来学习如何启动MySQL服务（MySQL Service）。

正常情况下，MySQL安装完成后会自动启动。但是作为开发者，我们需要掌握如何手动控制MySQL服务的启动和停止。

请看这两个命令：

service mysql stop

service mysql start

第一个命令是停止MySQL服务，第二个是启动MySQL服务。

那么，我们怎么确认MySQL服务是否真的启动成功了呢？请看屏幕上的这个命令：

sudo netstat -tap | grep mysql

执行这个命令后，如果你看到输出结果中MySQL处于LISTEN状态，就像图片中显示的这样，那就说明MySQL服务启动成功了。

大家可以看到这里显示的是：tcp 0 0 localhost:mysql *:* LISTEN 1088/mysqld

这表明MySQL正在监听连接，服务已经正常运行了。

슬라이드 4: 进入MySQL Shell界面

现在MySQL服务已经启动了，我们来学习如何进入MySQL的命令行界面，也就是MySQL Shell。

请大家跟我一起输入这个命令：

mysql -u root -p

让我解释一下这个命令的参数：

- -u root 表示我们要以root用户身份登录
- -p 表示需要输入密码

执行这个命令后，系统会提示你输入密码。在我们的教程中，我统一把密码设置为"hadoop"，大家输入的时候注意密码是不会显示在屏幕上的，这是正常的安全机制。

输入正确的密码后，你会看到像屏幕上显示的这样的欢迎信息，最重要的是最后会出现"mysql>"这个提示符。

看到这个提示符，就说明我们已经成功进入了MySQL的命令行环境！

슬라이드 5: 中文乱码问题诊断

好，现在我们来解决一个在中文环境下经常遇到的问题 - 中文乱码问题（Character Encoding Issues）。

当我们向MySQL数据库导入包含中文的数据时，可能会出现乱码。这是为什么呢？原因是MySQL默认的字符编码设置是latin1，而不是支持中文的UTF-8编码。

我们先来检查一下当前的字符编码设置。请在MySQL提示符下输入这个命令：

```
show variables like 'char%';
```

执行后你会看到像屏幕上这样的结果。请大家特别注意character_set_server这一行，你会发现它的值是latin1，这就是问题所在！

我们需要把它改成utf8才能正确处理中文。

其他的编码设置像character_set_client、character_set_connection等都已经是utf8了，只有这个服务器端的编码还是latin1。

슬라이드 6: 临时解决方案与永久解决方案

现在我来给大家介绍两种解决方案。

第一种是临时解决方案，我们可以在MySQL命令行中直接设置：

```
set character_set_server=utf8;
```

但是请注意！这种方法有一个很大的问题 - 一旦重启MySQL服务，这个设置就会失效，又会变回latin1。

所以我推荐大家使用永久解决方案，就是修改MySQL的配置文件。

我们需要做以下步骤：

(1) 修改配置文件

首先，我们需要退出MySQL，回到Linux终端。然后使用vim编辑器来编辑配置文件：

vim /etc/mysql/mysql.conf.d/mysqld.cnf

这个文件路径很重要，大家要记住。

슬라이드 7: 配置文件修改实践

请大家注意，刚才的vim命令是在Linux的Shell命令提示符下执行的，不是在MySQL的"mysql>"提示符下执行！这一点很重要，不要搞混了。

打开mysqld.cnf文件后，我们需要找到[mysqld]这个部分。然后在这个部分下面添加一行：

character_set_server=utf8

请大家看屏幕上的示例，我们可以看到配置文件的结构：

[mysqld]

#

* Basic Settings

#

user = mysql

pid-file = /var/run/mysqld/mysqld.pid

socket = /var/run/mysqld/mysqld.sock

port = 3306

basedir = /usr

datadir = /var/lib/mysql

tmpdir = /tmp

lc-messages-dir = /usr/share/mysql

character_set_server=utf8 ← 这一行是我们添加的

skip-external-locking

添加完成后，保存并退出vim编辑器。

슬라이드 8: 重启服务并验证

配置文件修改完成后，我们需要重启MySQL服务使配置生效。

(3) 重启MySQL服务

请在Linux终端（注意不是MySQL提示符）下执行：

```
service mysql restart
```

等待服务重启完成。

(4) 登录MySQL查看当前编码格式

重启完成后，我们再次登录MySQL来验证配置是否生效：

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

进入MySQL后，再次执行字符编码查询命令：

```
show variables like 'char%';
```

슬라이드 9: 验证配置成功

太好了！现在我们可以看到配置已经成功了。

请大家对比一下修改前后的结果：

- 修改前：character_set_server 的值是 latin1
- 修改后：character_set_server 的值是 utf8

现在所有的字符编码设置都是utf8了，这样我们就可以正常处理中文数据，不会再出现乱码问题。

这种修改是永久性的，即使重启MySQL服务，设置也不会丢失。

好，MySQL的安装和配置就到这里。接下来我们学习Kafka的安装。

슬라이드 10: Kafka课程内容概览

现在我们开始学习Kafka的安装和配置。

Kafka是现代大数据处理中非常重要的组件，特别是在流数据处理（Stream Processing）中。我们今天要学习的内容包括：

1. Kafka简介 - 了解Kafka的基本概念和作用
2. Kafka的安装和使用 - 实际动手安装和测试Kafka

为什么我们要学习Kafka呢？因为在后面的Spark Streaming课程中，Kafka将作为我们重要的数据源（Data Source）。

슬라이드 11: Kafka核心概念详解

首先，我来给大家介绍一下Kafka的核心概念。Kafka是一个高吞吐量的分布式发布订阅消息系统（Distributed Publish-Subscribe Messaging System）。

让我逐一解释这些重要概念：

Broker：这是Kafka集群中的服务器节点。一个Kafka集群可以包含一个或多个Broker。

Topic：这是消息的分类标签。每条发布到Kafka的消息都属于某个Topic。你可以把Topic理解为消息的频道或者分类。

Partition：这是Topic的物理分区。每个Topic可以分为多个Partition，这样可以提高并行处理能力。

Producer：消息生产者，负责向Kafka发送消息。

Consumer：消息消费者，负责从Kafka读取消息。

Consumer Group：消费者组。多个Consumer可以组成一个Consumer Group，同一个Consumer Group中的Consumer会协作消费消息，避免重复消费。

这些概念之间的关系是：Producer将消息发送到指定的Topic，Consumer从Topic中读取消息进行处理。

슬라이드 12: Kafka版本选择与下载

现在我们开始实际安装Kafka。首先要注意一个非常重要的问题 - 版本兼容性。

请大家访问Kafka官网的下载页面：https://kafka.apache.org/downloads

在选择Kafka版本时，有一个关键点：**Kafka的版本号必须与你系统上已安装的Scala版本号一致**。

为什么这么重要呢？因为我们后面要用Spark来处理Kafka的数据，而Spark是用Scala开发的。如果版本不匹配，就会出现兼容性问题。

在我们的教程环境中：

- Spark版本是3.4.0
- Scala版本是2.12.17

所以我们必须选择文件名以"2.12"开头的Kafka版本，比如kafka_2.12-3.5.1.tgz。

这里的命名规则是：kafka_[Scala版本]-[Kafka版本].tgz

- 前面的2.12是Scala版本号
- 后面的3.5.1是Kafka自身的版本号

大家一定要注意这个版本对应关系！

슬라이드 13: Kafka安装步骤

好，现在我们开始安装Kafka。请大家跟着我一步步操作：

假设你已经下载了kafka_2.12-3.5.1.tgz文件到Downloads目录。

```
cd ~/Downloads  # 进入下载目录
sudo tar -zxvf kafka_2.12-3.5.1.tgz -C /usr/local  # 解压到/usr/local
cd /usr/local  # 切换到/usr/local目录
sudo mv kafka_2.12-3.5.1 kafka  # 重命名为kafka
sudo chown -R hadoop ./kafka  # 改变所有者为hadoop用户
```

安装完成后，我们需要启动Kafka。但是请注意，Kafka依赖于Zookeeper，所以我们必须先启动Zookeeper。

现在请打开第一个终端，执行以下命令启动Zookeeper：

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

重要提醒：执行这个命令后，终端会显示很多信息，然后就停在那里不动了，没有回到命令提示符。请不要以为是死机了！这说明Zookeeper服务已经启动并正在运行。**千万不要关闭这个终端窗口**，否则Zookeeper服务就会停止。

슬라이드 14: 启动Kafka服务

Zookeeper启动后，现在我们来启动Kafka服务。

请打开第二个终端窗口，执行以下命令：

cd /usr/local/kafka

./bin/kafka-server-start.sh config/server.properties

同样地，执行这个命令后，终端窗口会返回很多信息，然后停住不动，没有回到Shell命令提示符状态。这同样不是死机，而是Kafka服务器已经启动并处于服务状态。

再次提醒：不要关闭这个终端窗口，一旦关闭，Kafka服务就停止了。

现在我们有两个终端在运行：

- 第一个终端运行Zookeeper
- 第二个终端运行Kafka

这两个服务都必须保持运行状态。

슬라이드 15: 创建Topic进行测试

现在Kafka服务已经启动，我们来测试一下是否工作正常。

请打开第三个终端，我们来创建一个Topic进行测试：

cd /usr/local/kafka

./bin/kafka-topics.sh --create --zookeeper localhost:2181 ₩

--replication-factor 1 --partitions 1 --topic wordsendertest

让我解释一下这个命令的参数：

- --create：创建新的Topic
- --zookeeper localhost:2181：指定Zookeeper地址，2181是默认端口
- --replication-factor 1：备份因子为1，因为我们是单机环境
- --partitions 1：分区数为1
- --topic wordsendertest：Topic名称

创建成功后，我们可以验证一下Topic是否创建成功：

./bin/kafka-topics.sh --list --zookeeper localhost:2181

如果能看到"wordsendertest"出现在列表中，说明Topic创建成功了。

슬라이드 16: 测试Producer功能

现在我们来测试Kafka的Producer功能。

在当前终端（第三个终端）中继续输入以下命令：

./bin/kafka-console-producer.sh --broker-list localhost:9092 ₩

--topic wordsendertest

注意这里的端口号：

- Zookeeper使用2181端口
- Kafka Broker使用9092端口

执行这个命令后，你会发现光标停在那里等待输入。现在你可以输入一些测试数据，比如：

hello hadoop

hello flink

每输入一行按回车，这行数据就会被发送到"wordsendertest"这个Topic中。

这个终端现在变成了一个Producer，专门用来发送消息。

슬라이드 17: 测试Consumer功能

最后，我们来测试Consumer功能，验证Producer发送的消息是否能被正确接收。

请打开第四个终端，执行以下命令：

cd /usr/local/kafka

./bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 ₩

--topic wordsendertest --from-beginning

参数说明：

- --bootstrap-server localhost:9092：指定Kafka服务器地址
- --topic wordsendertest：指定要消费的Topic
- --from-beginning：从头开始消费所有消息

执行后，你应该能在屏幕上看到之前在Producer终端输入的内容：

hello hadoop

hello flink

如果你看到了这些内容，恭喜你！这说明Kafka的Producer-Consumer模型工作正常。

现在你可以在Producer终端继续输入新的消息，你会发现Consumer终端会实时显示这些新消息，这就是Kafka的实时消息传递功能！

课程总结

好，今天的课程就到这里。我们学习了：

1. **MySQL的完整安装配置流程**，特别是解决中文乱码问题的永久方案
2. **Kafka的基本概念**和**完整的安装测试流程**

这些环境配置是我们后续学习Spark、Spark Streaming的重要基础。

课后作业：

1. 请大家在自己的环境中完整重现今天的所有操作
2. 尝试创建不同名称的Topic，测试多个Producer和Consumer
3. 思考一下：在实际的大数据应用中，Kafka可以解决什么问题？

下节课我们将学习如何在Spark中使用Kafka作为数据源。谢谢大家！

3.5 Anaconda的安装和使用方法 - 强化讲义脚本

슬라이드 1: Anaconda 다운로드 및 기본 설치 소개

大家好，今天我们来学习第三章第五节的内容，就是Anaconda的安装和使用方法。

首先，什么是Anaconda呢？Anaconda是一个Python数据科学平台（Python Data Science Platform），它包含了Python解释器（Python Interpreter）以及很多用于数据分析的包（Data Analysis Packages）。对于我们后面要学习的大数据处理（Big Data Processing）和Apache Spark来说，Anaconda是非常重要的工具。

现在我们看看怎么下载Anaconda。理论上，我们可以从官方网站 https://www.anaconda.com/ 下载，但是呢，大家都知道，访问国外网站的速度比较慢，所以我们一般选择国内的镜像网站。

我推荐大家使用清华大学的镜像网站：https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/。在这个网站上，我们可以选择下载Anaconda3-2023.07-2-Linux-x86_64.sh这个版本。

下载完成后，我们在Linux终端（Linux Terminal）中执行这两个命令：

```
$ cd /home/hadoop/Downloads  # 假设安装文件在这个目录下
$ sh ./Anaconda3-2023.07-2-Linux-x86_64.sh
```

注意，这里的路径可能根据你们的实际情况有所不同，重要的是要进入到下载文件所在的目录。

슬라이드 2: 라이선스 동의 과정 설명

好，当我们执行了刚才的安装命令之后，会出现这样的界面。大家可以看到屏幕上显示的内容。

首先，系统会要求我们阅读许可文件（License Agreement)。这个许可文件比较长，我们可以按回车键开始阅读。如果觉得太长，可以按空格键向下翻页，或者直接按住空格键快速浏览。

当出现"Do you accept the license terms? [yes|no]"这个提示的时候，我们需要输入"yes"。这一步是必须的，因为我们需要同意Anaconda的使用条款才能继续安装。

大家看到终端显示的内容了吗？

Welcome to Anaconda3 2023.07-2

In order to continue the installation process, please review the license

agreement.

Please, press ENTER to continue

>>>

这就是安装程序的欢迎界面，我们按回车键继续就可以了。

슬라이드 3: 설치 경로 설정 및 초기화 과정

接下来，安装程序会询问我们安装路径的问题。

大家看这个界面：

Anaconda3 will now be installed into this location:

/home/hadoop/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/hadoop/anaconda3] >>>

系统建议我们把Anaconda安装到"/home/hadoop/anaconda3"这个目录。一般来说，我们使用默认路径就可以了，直接按回车键确认。当然，如果你们有特殊需求，也可以指定其他路径。

然后，非常重要的一步来了！当安装程序询问是否需要初始化（Initialize）的时候：

Do you wish the installer to initialize Anaconda3

by running conda init? [yes|no]

[no] >>> yes

这里我们一定要输入"yes"！为什么呢？因为这个初始化过程会自动配置我们的环境变量（Environment Variables），让我们能够在任何地方使用conda命令（conda commands）。如果选择"no"，后面我们就需要手动配置，会比较麻烦。

슬라이드 4: 설치 확인 및 conda 설정

安装完成后，我们需要重新打开一个终端窗口。为什么要重新打开呢？因为环境变量的修改需要重新加载才能生效。

大家注意看，新打开的终端提示符变成了这样：

(base) hadoop@hadoop01:~$

看到前面多了一个"(base)"了吗？这说明我们的Anaconda安装成功了，现在我们处在base环境（base environment）中。

但是，还有一个问题需要解决。由于Anaconda是国外的软件，当我们使用conda安装其他包（packages）的时候，默认会从国外的服务器下载，速度会很慢。所以我们需要配置国内的镜像源（Mirror Sources）。

我们执行这个命令来创建配置文件：

$ vim ~/.condarc

这个.condarc文件是conda的配置文件（Configuration File），我们可以在里面设置镜像源的地址。

슬라이드 5: conda 미러 소스 설정 방법

现在我们来配置镜像源。在刚才打开的.condarc文件中，我们需要输入以下内容：

```
channels:
  - defaults
show_channel_urls: true
channel_alias: https://mirrors.tuna.tsinghua.edu.cn/anaconda
default_channels:
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/pro
  - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/msys2
custom_channels:
  conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  msys2: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  bioconda: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  menpo: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
  simpleitk: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
```

大家注意，这个配置文件的格式是YAML格式，缩进（indentation）很重要，一定要保持正确的缩进。所有这些地址都指向清华大学的镜像服务器，这样我们下载包的速度就会快很多。

配置完成后，记得保存文件。在vim中，我们按Esc键，然后输入":wq"保存并退出。

슬라이드 6: Python 버전 확인 및 Python 3.8 설치

现在我们来检查一下Python的版本。在终端中输入"python"命令：

(base) hadoop@hadoop01:~$ python

Python 3.11.4 (main, Jul 5 2023, 14:15:25) [GCC 11.2.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

>>>

大家看到了，当前的Python版本是3.11.4。但是这里有一个问题！

Python 3.11.4这个版本对于我们后面要学习的Apache Spark 3.4.0来说，版本太高了，会有兼容性问题（Compatibility Issues)。所以我们需要安装一个与Spark兼容的Python版本，也就是Python 3.8。

现在我们使用conda来创建一个新的环境（Environment)：

$ conda create -n pyspark python=3.8

这个命令的意思是：创建一个名叫"pyspark"的新环境，并在这个环境中安装Python 3.8。这里"pyspark"是我们给这个环境起的名字，你们也可以起其他名字。

安装过程中，系统会询问"Proceed ([y]/n)?"，我们输入"y"确认就可以了。

安装完成后，我们可以切换到这个新环境：

$ conda activate pyspark

슬라이드 7: 환경 전환 확인 및 Python 종료

现在我们来验证一下环境切换是否成功。输入"python"命令：

```
$ python
```

如果切换成功，我们会看到这样的输出：

```
(pyspark) hadoop@hadoop01:~$ python
Python 3.8.18 (default, Sep 11 2023, 13:40:15)
[GCC 11.2.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

大家注意两个变化：

1. 终端提示符从"(base)"变成了"(pyspark)"
2. Python版本从3.11.4变成了3.8.18

这说明我们成功创建并切换到了pyspark环境！

最后，我们可以使用exit()函数退出Python交互模式（Interactive Mode）：

```
>>> exit()
```

这样我们就又回到了Linux终端。

슬라이드 8: 본 장 요약

好，我们来总结一下这一节的内容。

首先，我们学习了为什么需要安装Anaconda。在大数据处理和机器学习（Machine Learning）项目中，我们经常需要使用各种Python包，Anaconda为我们提供了一个很好的包管理系统(Package Management System)。

其次，我们学习了Anaconda的完整安装过程：

1. 从清华镜像网站下载安装包
2. 执行安装脚本并同意许可协议
3. 选择安装路径并进行初始化
4. 配置国内镜像源以加快下载速度

然后，我们解决了一个重要的兼容性问题：由于Spark（Apache Spark）3.4.0与Python 3.11版本存在兼容性问题，我们创建了一个专门的pyspark环境，安装了Python 3.8版本。

最后，我们学会了如何在不同的conda环境之间切换，这对于管理不同的项目非常有用。

在下一节课中，我们将开始学习如何在这个环境中安装和配置Spark，为后面的大数据分析课程做准备。

大家对今天的内容有什么问题吗？如果没有问题，我们就到这里，下课！