大数据技术基础



类目：计算机类
书名：大数据技术基础
主编：董 磊　黎 军　孙 楠
出版社：江苏大学出版社
开本：正 16 开
书号：978-7-5684-2861-3
使用层次：通用
出版时间：2026 年 7 月
定价：49.80 元
印刷方式：双色
是否有资源：有

责任编辑：徐　婷
封面设计：旗语书装

计算机类创新融合精品规划教材
“互联网+”教育改革新理念教材





大数据技术基础

计算机类创新融合精品规划教材
“互联网+”教育改革新理念教材
大数据技术基础

大数据技术基础

主编◎董　磊　黎　军　孙　楠

主编◎董　磊　黎　军　孙　楠

ISBN 978-7-5684-2861-3
定价：49.80元

江苏大学出版社
JIANGSU UNIVERSITY PRESS

计算机类创新融合精品规划教材
“互联网 +”教育改革新理念教材



大数据技术基础

主　编◎董　磊　黎　军　孙　楠
副主编◎唐　晶　蔺晓东　孙业鑫
吕尚征

江苏大学出版社
JIANGSU UNIVERSITY PRESS
镇　江

图书在版编目（CIP）数据

大数据技术基础 / 董磊，黎军，孙楠主编. -- 镇江 ：江苏大学出版社，2026. 7. -- ISBN 978-7-5684-2861-3

Ⅰ. TP274

中国国家版本馆 CIP 数据核字第 202696PG51 号

大数据技术基础

Dashuju Jishu Jichu

主　　编 / 董　磊　黎　军　孙　楠
责任编辑 / 徐　婷
出版发行 / 江苏大学出版社
地　　址 / 江苏省镇江市京口区学府路 301 号（邮编：212013）
电　　话 / 0511-84446464（传真）
网　　址 / http：//press. ujs. edu. cn
印　　刷 / 河北龙大印务有限公司
开　　本 / 787 mm×1 092 mm　1/16
印　　张 / 14. 5
字　　数 / 311 千字
版　　次 / 2026 年 7 月第 1 版
印　　次 / 2026 年 7 月第 1 次印刷
书　　号 / ISBN 978-7-5684-2861-3
定　　价 / 49. 80 元

如有印装质量问题请与本社营销部联系（电话：0511-84440882）

PREFACE 前言

在当今数字化时代，大数据已成为推动社会发展和变革的关键力量。从商业决策到科学研究，从公共服务到日常生活，大数据的应用无处不在，深刻地改变着我们的世界。

大数据技术作为一个融合了计算机科学、统计学、数学等多学科知识的综合性领域，正以其强大的数据分析和处理能力，为各行业带来前所未有的机遇和挑战。随着互联网、移动设备、传感器等技术的飞速发展，数据量呈爆炸式增长，如何从海量的数据中提取有价值的信息，成为亟待解决的问题。大数据技术应运而生，它为我们提供了有效的手段和方法，帮助我们更好地理解和利用数据。

本书旨在搭建全面、系统的大数据技术学习平台。首先，通过理论与实践相结合的方式，详细介绍了大数据的产生背景、发展历程、定义、分类、特征、应用场景等基础知识，使读者对大数据形成清晰的认识。然后，深入探讨了大数据软件架构、采集与预处理、存储与管理、计算、分析与挖掘、可视化、安全等核心技术，以及大数据技术与人工智能、云计算、物联网、区块链等新一代信息技术的融合应用，帮助读者掌握大数据技术的关键要点和实际应用方法。

本书具有以下特点：

系统性强：按照大数据技术的应用流程，从数据的产生、采集、存储、处理、分析到可视化和安全，进行了全面、系统的介绍，形成了完整的知识体系。

实用性高：除介绍大数据技术的基本理论和方法外，还详细阐述了大数据技术在各行业的应用场景和解决方案，使读者能够更好地将所学知识应用到实际工作中。

融入思政内容：在每个项目中都设置了思政目标，将思想政治教育融入专业知识的学习中，培养读者的社会责任感和创新精神。

本书适合作为高等学校计算机科学、数据科学、统计学等相关专业的教材，也可供大数据技术爱好者自学参考，同时能为从事大数据研究、开发和应用的专业人员提供有价值的参考资料。

由于大数据技术发展迅速，书中难免存在不足之处，敬请广大读者批评指正。

编　者

CONTENTS 目录

1

项目一　初识大数据　/001

任务一　大数据的产生背景与发展历程　/003
任务二　大数据的定义、分类与特征　/008
任务三　大数据的应用场景　/012

2

项目二　大数据软件架构　/019

任务一　Hadoop 架构　/020
任务二　Spark 架构　/034
任务三　实时流处理架构　/042

3

项目三　大数据采集与预处理　/051

任务一　大数据采集　/052
任务二　大数据预处理　/060

4 项目四 大数据存储与管理 /069

任务一 传统数据存储与管理技术 /070
任务二 大数据存储与管理技术 /089

5 项目五 大数据计算 /104

任务一 大数据计算基本框架 /105
任务二 大数据批处理计算 /107
任务三 大数据流计算 /114

6 项目六 大数据分析与挖掘 /120

任务一 大数据分析 /122
任务二 大数据挖掘 /134

7 项目七 大数据可视化 /158

任务一 大数据可视化概述 /159
任务二 科学可视化 /169
任务三 信息可视化 /172
任务四 数据可视化的应用 /177

8

项目八　大数据安全　/186

任务一　大数据安全概述　/187
任务二　大数据保护的原则与对策　/193

9

项目九　大数据技术与新一代信息技术的融合应用　/205

任务一　人工智能　/207
任务二　云计算　/210
任务三　物联网　/213
任务四　区块链　/217

参考文献　/223

项目一 初识大数据

项目导读

在数字化浪潮的推动下，数据已成为与土地、劳动力、资本同等重要的生产要素，深刻改变着经济发展、社会治理和日常生活的方方面面。大数据作为数字经济的核心载体，其相关技术与应用正加速渗透到各个行业领域。本项目从大数据的产生背景出发，系统介绍大数据的定义、分类、特征和应用场景，分析其未来发展趋势与面临的挑战，并通过实训任务引导学习者发现身边的大数据应用，帮助学习者建立对大数据的全面认知，为其后续深入学习大数据技术与实践应用奠定坚实的理论基础。

学习目标

知识目标

1. 了解大数据的产生背景，掌握大数据发展的各阶段核心特征。
2. 掌握大数据的定义与核心本质，能准确区分结构化、半结构化和非结构化数据并列举典型案例。
3. 熟练掌握大数据的“4V”（Volume、Variety、Velocity、Value）特征，明确各特征的核心内涵。
4. 了解大数据在电商、交通、医疗、金融、教育、制造等领域的典型应用场景，理解其中的数据流转逻辑与应用价值。

能力目标	1. 能结合具体场景识别大数据应用，准确分析其数据来源、数据类型及体现的“4V”特征，提升知识迁移应用能力。 2. 具备搜索、筛选并分析身边大数据案例的能力，能清晰梳理案例的核心技术与价值实现路径。 3. 能辩证看待大数据的价值与风险，针对具体挑战提出初步应对思路，培养问题分析与解决能力。 4. 具备小组协作完成案例调研、报告撰写与成果展示的能力，提升沟通表达能力与团队协作效率。
素质目标	1. 建立“数据驱动”的思维模式，养成从数据视角观察、分析生活与工作中问题的习惯。 2. 认识大数据作为核心生产要素的战略意义，培养对新兴技术的探索兴趣与持续学习意识。 3. 强化数据安全与隐私保护意识，树立合规使用数据、尊重个人隐私的职业素养与责任担当。 4. 培养辩证思维与系统思维，既能看到大数据带来的便利与价值，也能理性认知其潜在风险与局限。

思政目标

1. 结合大数据国产化替代趋势与行业应用成果，增强科技自信与民族自豪感，激发爱国情怀。

2. 认识大数据在社会治理、民生改善、产业升级中的重要作用，理解技术创新服务于国家发展与人民幸福的使命担当。

3. 树立尊重数据伦理、坚守公平正义的价值导向，自觉抵制算法偏见、数据滥用等不良现象，践行科技向善理念。

4. 培养精益求精的工匠精神与协同合作的团队意识，为未来投身数字经济建设、助力国家数字化转型奠定思想基础。

任务一　大数据的产生背景与发展历程

任务导入

小明是一名大一新生，他发现一些有趣的现象：妈妈买菜时会通过手机 App 查看实时菜价和产地信息，爸爸开车时依赖导航软件躲避拥堵，自己刷短视频时总能看到喜欢的内容。这些现象背后似乎都有一股“看不见的力量”在运作。老师告诉他，这一切都离不开大数据。小明不禁疑惑：大数据是怎么产生的？它经历了怎样的发展过程才走到今天？带着这些问题，我们一起探索大数据的起源与发展。

任务分析

在本任务中，我们将了解大数据的产生背景，明确大数据的出现并非偶然，而是信息技术发展与社会需求升级的必然结果；梳理大数据及其相关技术的发展脉络，掌握不同阶段的技术特征和应用特点，进而理解技术进步与应用需求之间的相互促进关系。

任务实施

1. 分组收集资料：以“大数据产生的技术推手”为主题，分组收集物联网、分布式存储、分布式计算等技术的发展历程资料，每组重点分析一种技术对大数据产生的推动作用。

2. 时间轴制作：结合教材内容，制作大数据发展历程时间轴，标注关键时间节点、标志性事件和核心特征。

3. 讨论分享：每组推选代表分享收集的资料和制作的时间轴，讨论“技术进步”与“需求升级”在大数据发展中的相互作用。

知识讲解

一、大数据的产生背景

自 20 世纪 60 年代以来，信息技术的快速发展已经彻底改变了人们的生活方式、工作方式和社会结构。这种变革不仅对个人产生了深远影响，也给政府、商业和科学领域带来了巨大的机遇和挑战。美国未来学家阿尔文·托夫勒（Alvin Toffler）将这种变革喻为信息化浪潮。之所以将信息化变革比喻为浪潮，是因为它如同自然界的浪潮不断地涌动、起伏，并且永不停息。每一次新的浪潮都会在前一次浪潮的基础上推动社会的发展，形成新的产业和商业模式。

回望历史长河，人类社会已历经多轮信息化浪潮的洗礼。20 世纪 80 年代前后，个人计算机（PC）的普及拉开了第一次信息化浪潮的序幕，计算机逐步渗透至企业办公与家庭生活的各个角落，人们借助计算机与网络实现数据的存储、处理及信息传输，社会生产力由此获得大幅跃升，联想（Lenovo）、微软（Microsoft）、英特尔（Intel）、IBM（国际商业机器公司）等企业成为这一时期的标志性代表。

20 世纪 90 年代中期，互联网时代呼啸而至，网络通信技术让信息传播的速度与便捷性实现了质的飞跃，每个人都能通过互联网实时获取千里之外的信息，原本遥远的世界由此浓缩成一个紧密相连的“地球村”，人类也随之迎来第二次信息化浪潮，谷歌、百度、阿里巴巴、腾讯等互联网巨头在这一时期应运而生。

如今，身处数字化时代的我们，正置身于一场前所未有的信息化浪潮之中。十几年间，物联网（IoT）、云计算、人工智能（AI）等新兴技术如雨后春笋般涌现，海量数据正以远超人类想象的速度持续生成、采集与存储。以大数据为核心的应用场景，不仅重塑了商业领域的生态格局，更深刻变革了传统社会的运作模式。

二、大数据的发展历程

大数据时代的来临并非出于偶然，而是技术发展的必然结果，具体可归结为两个核心驱动力：其一，硬件设备的迭代升级——作为大数据应用的物理基石，升级后的硬件不仅提供了更庞大的数据存储容量、更迅捷的数据传输速率，而且具备了更高效的计算处理能力，为大数据分析及应用奠定了基础；其二，大数据算法架构的创新发展——大数据框架与核心算法的理论研究取得突破性进展，其中并行计算与分布式处理的核心思想，让大数据任务调度、数据处理得以以高效、迅捷且精准的方式进行。下面将从这两个维度出发，简要梳理大数据及相关技术的发展脉络。

（一）硬件设备的升级

1. 更大的数据存储容量

数据一般存储于磁带、磁盘、光盘、硬盘等各种类型的存储设备中，作为数据的存储媒介，存储设备的实际物理存储空间决定了数据存储量级的上限。随着技术的革新和进步，存储设备的工艺制造水平不断提升，物理存储容量不断拓展，存储成本则不断降低，这无疑为大数据存储提供了有利的条件。

传统存储设备的出现可以追溯到 20 世纪 50 年代，IBM 推出了第一代磁带存储设备。这种设备基于电磁感应原理，通过覆盖磁性涂层的塑料带来存储数据，并使用读写头来访问数据，这种物理设计也作为磁存储技术的代表被沿用至今。在计算机外存方面，软盘、机械硬盘等磁存储器都是重要的存储器件。其中，软盘由于性能无法满足市场需求很快遭到淘汰，而机械硬盘凭借存储容量大、价格便宜等优势得以保留并不断演进。

光学存储设备是传统存储设备的另一个分支，也在存储技术发展的历史洪流中有着举足轻重的作用。1965 年，只读光盘存储器的诞生代表着光学存储技术登上了历史舞台。光学存储技术利用激光照射介质，使介质的物理或化学性质发生改变，从而实现数据存储。常说的 CD 光盘、DVD 光盘等均是光学存储设备的代表，主流 CD 光盘的存储容量为 700MB 左右，DVD 光盘的存储容量可达 4.7GB，而蓝光光盘具有性价比高、环境适应性强的特点，存储容量可达 25GB，最大数据传输速率达 54Mbit/s，是光学存储设备的重要发展方向。

与此同时，以闪存为代表的新型存储介质也开始得到大规模的普及和应用。1980 年，东芝最先推出了闪存技术。所谓闪存，其实是一种新兴的半导体存储器，它具有断电非易失、体积小、质量轻、能耗低等特点。以闪存颗粒为基础制造的固态硬盘（SSD）具有存取速度快、稳定性好等优势，在计算机存储领域已经得到了广泛应用。主流的闪存颗粒包括 NAND 闪存（与非型闪存）和 NOR 闪存（或非型闪存），其中 NAND 闪存较 NOR 闪存存储容量更大、读取速度更快。目前，NAND 闪存技术已发展到三维存储单元在垂直方向上层叠在一起，形成多个存储层。3D NAND 闪存芯片在密度、成本等方面较 2D NAND 闪存芯片具有更大优势，三星、海力士等行业知名的半导体厂商都在积极开拓 3D NAND 闪存芯片的市场，华为、EMC（易安信）及 IBM 等存储阵列厂商也在围绕固态硬盘及全闪存阵列开展相关研究。基于闪存的固态硬盘可实现几万甚至更高的 IOPS（Input/Output Operations Per Second，每秒的读写次数），访问延迟只有几十微秒，相信在未来这一指标仍能有所突破。

2. 更快的数据传输速率

我们通常使用网络带宽来表示数据在网络中传输的速率，以每秒传输的数据量来

衡量，单位为比特每秒（bit/s），常用的还有千比特每秒（kbit/s）、兆比特每秒（Mbit/s）、吉比特每秒（Gbit/s）等。

数字信号的传输方式经历了三个发展阶段。

第一阶段：模拟调制解调器时期（1960—1990 年）。在这个时期，数据主要通过电话线传输，调制解调器是最主要的网络接入设备，网络带宽的上限只有 56kbit/s。

第二阶段：数字调制解调器时期（1990—2000 年）。此时期内使用离散二进制数字信号在不同设备和网络间进行通信，相比模拟调制解调器，数字调制解调器有更好的容错性和更低的误码率，网络带宽也在这一时期得到了明显提升，可达到 10Mbit/s、100Mbit/s，甚至更高。

第三阶段：光纤时代（2000 年至今）。随着光纤通信技术的不断发展和普及，网络带宽得到了极大提升，可达到 10Gbit/s、40Gbit/s、100Gbit/s 等级别。同时，5G（第五代移动通信技术）的应用也为移动设备的网络带宽提供了更大的发展空间。在大数据时代，信息传输不会像网络发展初期那样受到数据传输大小的限制及数据传输效率的制约。

3. 更高效的计算处理能力

英特尔公司的创始人之一戈登·摩尔（Gordon Moore）提出了关于计算机核心处理硬件设备发展的著名经验法则，该法则被后人称为“摩尔定律”。他指出，集成电路上可以容纳的晶体管数目每经过 18～24 个月便会增加一倍，换言之，处理器的性能大约每两年提升一倍，同时价格下降为之前的一半。近年来，集成电路中的晶体管尺寸已经突破了纳米级别，逐渐接近原子尺度的物理限制，摩尔定律也因集成电路难以继续增加晶体管的数量而受到挑战。尽管这一在过去几十年中都被证明是正确的经验法则在当前的技术水平下遭到了质疑，但技术领域依然在探索和寻找新的解决方案，试图延续摩尔定律的趋势。其中，三维集成电路、量子计算和其他新兴技术都被视为具有潜力的技术发展方向。

即使摩尔定律的绝对适用性有所减弱，计算机处理能力的发展也绝不会停滞不前。在过去的 30 多年里，得益于制程技术的进步与多核处理器的普及和发展，计算机中央处理器（Central Processing Unit，CPU）的处理速度已经从 10MHz 提高到 5GHz 以上，计算处理效率得到了大幅提升。除此之外，拥有更强并行计算能力、更大内存带宽的图形处理器（Graphics Processing Unit，GPU）也获得了广泛关注，无论是在图形渲染还是科学计算等方面，GPU 都有着绝对的优势，推动了近年来人工智能与深度学习领域的发展。随着云服务的推广和普及，越来越多的人可以接触到海量的计算资源，从而大幅降低大数据分析和处理的门槛。

（二）大数据算法架构的发展

从 2003 年起，谷歌（Google）陆续发表了 3 篇引爆大数据时代的论文，即 *The*

google file system，*MapReduce*：*Simplified data processing on large clusters* 及 *Bigtable*：*A distributed storage system for structured data*，其主题分别为可扩展分布式文件系统、大数据分布式计算方式及分布式数据存储系统。这 3 篇论文影响了当今的大数据生态，作为大数据架构设计的“三驾马车”，其奠定了大数据全球化发展的基础。

2006 年，Apache 软件基金会 Nutch 引擎项目组的道格·卡廷（Doug Cutting）等人受到谷歌 MapReduce（分布式计算框架）计算方式的启发，将其与 NDFS（Nutch Distributed File System）相结合，用来支持 Nutch 引擎的主要算法，Hadoop（分布式系统基础架构）由此诞生。随着应用的不断拓展，Hadoop 的一些问题逐渐暴露出来。由于 MapReduce 编程模型的编写较为复杂，Yahoo 内部开发了一种名为“Pig”的脚本语言，该语言提供类 SQL（结构化查询语言）的语法，开发者可以用 Pig 脚本描述要对数据集进行的编译操作，数据集经过编译后会生成 MapReduce 程序，并在集群中运行。

2007 年，Facebook 基于使用 Pig 脚本语言对 MapReduce 模型进行转换编译的思路开发了 Hive 数据仓库工具，该工具支持使用 SQL 语言进行大数据计算，极大地简化了编写 MapReduce 分布式程序的过程。此外，Hadoop 社区引入 YARN（资源调度系统）作为通用分布式资源管理系统和调度平台，解决了原始 MapReduce 模型只适用于批处理任务、无法进行实时计算和交互式查询等问题。至此，Hadoop 生态环境基本建立。

2012 年前后，加州大学伯克利分校（UC Berkeley）AMP 实验室（AMPLab）开发的 Spark（大数据并行计算框架）逐渐在大数据应用与实践领域崭露头角。Spark 使用弹性分布式数据集和内存计算模型，大大提升了数据并行处理速度，加之 Spark SQL、MLlib、GraphX 等组件的开发，使得 Spark 拥有更加强大的生态系统，逐步取代了 MapReduce 在企业应用中的地位。

随着大数据技术的快速发展，除了上述大数据框架，还涌现出许多其他的大数据算法框架和工具，如基于流式处理框架的 Flink、Storm、Kafka 等。这些框架和工具旨在解决大数据处理中的特定问题，并提供更高效、更灵活的解决方案。

从最初的 MapReduce 技术到 Hadoop 生态系统，再到更加灵活、高效的新算法框架和工具，大数据算法架构的发展向着更高的可扩展性、更快的处理效率及更细化的应用领域迈进。这些框架和工具的发展不断推动着大数据应用的创新，为大规模数据处理问题的解决提供理论层面和方法层面的强大支持。

任务二　大数据的定义、分类与特征

任务导入

某电商平台每天会产生海量数据：用户的订单信息（买家、商品、金额、时间）、浏览记录（点击商品、停留时长）、评价内容（文字、图片）、物流轨迹（位置、配送状态）。这些数据形态各异，有的格式规范，有的杂乱无章。平台如何对这些数据进行分类管理？又如何从这些看似零散的数据中挖掘价值，为用户推荐心仪的商品？要解决这些问题，我们需要先掌握大数据的定义、分类与核心特征。

任务分析

在本任务中，我们将了解大数据的定义，明确大数据的核心属性并非“规模大”而是“价值高”；掌握大数据的分类以及不同类型数据的特点，为后续数据处理技术的学习奠定基础；熟悉大数据的“4V”特征，能够结合实际案例分析每个特征的具体表现，深化对大数据本质的理解。

任务实施

1. 数据分类实操：收集生活中的不同类型数据，分别判断其属于结构化、半结构化还是非结构化数据，并说明判断依据。

2. “4V”特征案例分析：选择一个熟悉的大数据应用（如智能导航、短视频推荐、金融风控）场景，分析该应用如何体现大数据的“4V”特征，填写表 1-1。

表 1-1　“4V”特征在应用中的具体体现

大数据特征	应用中的具体体现
Volume（规模庞大）	
Variety（多样性）	
Velocity（高速性）	
Value（价值密度低）	

3. 小组讨论：结合分析结果，讨论“价值密度低”是否大数据处理的核心挑战，以及如何应对这一挑战。

知识讲解

一、大数据的定义

在深入了解大数据及相关概念之前，首先需了解什么是数据。“数据”（data）在拉丁文里是“已知”的意思，也可以理解为“事实”。从广义上说，数据是指对客观事物的性质状态及相互关系等方面进行记载的符号或符号组合。它可以是可识别的、具体的，也可以是模糊的、抽象的。它不仅指数字，也指具有一定意义的文字、数字符号的组合，如图形、图像、视频、音频等，还是客观事物的属性、数量、位置及其相互关系的抽象表示。尽管人类所能观测到的一切事物均可以用数据来抽象表示，但为了避免超出本书讨论的范围，从计算机科学的角度出发，我们将数据的定义局限为所有能输入计算机中并被计算机程序处理的符号的总称。

了解了数据的概念后，还需要了解以下问题：为什么仅仅在“数据”前加了一个“大”，相关应用和技术发展直到近年来才有所突破？这个所谓的“大”，到底是多大？大数据和传统意义上的数据有什么区别？想要解答这些问题，就要引入数据量级的概念。

目前国际上普遍采用的标准度量衡单位体系中，已有很多量的基本单位，就像长度的基本单位是米（m），数据也有用来衡量数据量大小的基本单位。我们提到的数据需要被编码成计算机可以识别的二进制格式，而数据存储的基本单位字节（Byte，缩写为 B）正好是由 8 个二进制位［即 8 比特（bit）］组成的。因此，可以得到关于数据存储单位的换算关系，即 1 字节＝8 比特，而比特是数据存储的最小单位。为了便于日常使用，我们一般会用一组前缀与基本单位组合形成的整数或小数单位来对世界上一切可以测量的物质进行量化计算。例如，在计算北京至上海的直线距离时，我们通常不用米而用千米做单位；在计算一颗苹果的重量时，我们通常不用千克而用克做单位。当数据规模很大时，用字节作为单位来描述数据量的大小就显得十分不方便。由于硬件设计问题，计算机处理信息时通常使用 2 的整数倍作为进位边界，最接近 1000 的 2 的整数倍是 $2^{10}=1024$，因此，不同于其他单位的换算逻辑，描述数据量级的单位会使用 1024 作为进位标准。表 1-2 中列举了一些数据量的表示单位及对应换算关系。

表 1-2　数据量的表示单位和对应换算关系

词头名称	单位名称	符号	换算关系
千 kilo	千字节	KB	1KB=1024B
兆 mega	兆字节	MB	1MB=1024KB
吉［咖］giga	吉字节	GB	1GB=1024MB
太［拉］tera	太字节	TB	1TB=1024GB
拍［塔］peta	拍字节	PB	1PB=1024TB
艾［可萨］exa	艾字节	EB	1EB=1024PB
泽［塔］zetta	泽字节	ZB	1ZB=1024EB
尧［塔］yotta	尧字节	YB	1YB=1024ZB
布［朗托］bronto	布字节	BB	1BB=1024YB
诺［纳］nona	诺字节	NB	1NB=1024BB

大数据之父维克托·迈尔-舍恩伯格曾提出，大数据是指无法在一定时间范围内用常规软件工具进行捕捉、管理和处理的数据合集。其并不单指数据量达到很高的阈值（如多少 PB 或多少 EB），只不过如今使用全量数据时数据量很容易达到 PB 量级或 EB 量级，这很容易造成读者对大数据量级的误解。大数据的真正意义不在于“数据量大”，而在于通过数据分析、比对、挖掘等一系列操作，发现新知识、创造新价值、培养新能力。

二、大数据的分类

不同类型的数据，需选用不同的存储和处理方式。根据数据的结构化程度，大数据可分为结构化数据、半结构化数据和非结构化数据三类。

（一）结构化数据

结构化数据是指具有固定格式、明确逻辑关系的数据，能够直接存入关系型数据库（如 MySQL、Oracle）的表格中，以行和列的形式组织存储。这类数据的特点是格式规范、易于查询和分析，是传统数据处理的主要对象。

典型例子包括：银行交易记录（账号、交易金额、交易时间、交易对象）、学生学籍信息（学号、姓名、性别、成绩）、企业财务报表（日期、收入、支出、利润）、Excel 表格数据等。

结构化数据在大数据总量中占比约 20%，处理难度较低，可通过 SQL 语句等传统工具进行查询和分析。

（二）半结构化数据

半结构化数据是介于结构化数据与非结构化数据之间的数据，没有严格的固定格式，但包含标签、关键字段等结构化元素，能够通过这些元素实现部分数据的解析和组织。

典型例子包括：XML 文档、Java Script 对象简谱（Java Script Object Notation，JSON）数据、服务器访问日志（包含时间戳、IP 地址、访问路径、状态码等字段）、电子邮件（包含发件人、收件人、发送时间、正文等元素）、HTML 页面等。

半结构化数据在大数据总量中占比约 30%，处理难度中等，需要通过专门的解析工具提取结构化字段，再进行进一步分析。

（三）非结构化数据

非结构化数据是指没有固定格式、无明确逻辑关系的数据，以文本、图像、音频、视频等形式存在，是大数据中占比最高、增长最快的类型。

典型例子包括：社交媒体评论、短视频、语音消息、医学影像（如 CT、X 光片）、手写笔记、扫描件、监控视频、传感器原始数据等。

非结构化数据在大数据总量中占比约 50%，且年增长率达到 60%，处理难度较高，需要采用自然语言处理、图像识别、语音识别等人工智能技术进行解析和价值提取，是当前大数据处理的重点和难点领域。

三、大数据的特征


大数据的特征

大数据的“4V”特征是其与传统数据相比的独特之处，即规模庞大（Volume）、多样性（Variety）、高速性（Velocity）及价值密度低（Value）。

1. 规模庞大

数据规模庞大是针对现有计算和存储能力而言的。一开始人们认为 PB 量级的数据已经是“天文数字”了，而随着互联网用户生成的内容和传感器实时获取的数据的增加，数据规模正以惊人的速度增长。据国际数据公司的统计，2025 年全球数据总量为 213.56ZB，预计 2029 年将达到 527.47ZB，核心增量来自生成式 AI、企业数字化、云端数据扩张。

2. 多样性

大数据的多样性是针对大数据所面对的应用场景而言的。在不同应用场景中，一方面，需要处理的数据可能同时包含结构化、半结构化和非结构化数据；另一方面，同类型的数据中也可能包含具有不同数据范式和存储逻辑的数据子集。数据的异构性

增加了数据处理的复杂性，这对数据处理的逻辑提出了更高的要求。

3. 高速性

大数据的高速性主要体现在两个维度：一方面，高速性是指数据的处理速度快。数据的高速处理是数据量不断增长的必然要求，我们不可能无限制地等待计算机对输入的数据进行处理。如今很多数据均以流的形式进行处理，分布式系统也需要保证批处理结果的快速反馈，这促进了大数据处理算法的迭代升级。另一方面，高速性也指数据的变化速度快。大数据来源于对现实世界的持续观察，而世界是在不断变化发展的，因此采样样本也是动态变化的。

4. 价值密度低

大数据价值密度低是指大数据分析可以通过海量数据之间的相关性获得高价值的但非显而易见的隐含知识，从而挖掘出巨大的数据价值。但大数据的价值并不一定随着数据规模的增加而增加，在很多情况下具有分析处理价值的数据被淹没在大量的无用数据中，有挖掘价值的数据仅占数据总量的一小部分。由于无法探明这部分有价值的数据在海量数据中的具体位置，因此只能像“大海捞针”那样在数据总体中寻觅，因此大数据具有价值密度低的特征。也正是基于此，如何针对具体问题快速定位有价值的数据，并从中更加高效地挖掘出数据的价值，仍是当前大数据领域的核心问题之一。

任务三　大数据的应用场景

任务导入

在日常生活中，我们早已被大数据包围：早晨醒来，天气 App 推送精准的当日预报；通勤时，导航软件规划最优路线；购物时，电商平台推送心仪商品；就医时，医生利用医疗数据辅助诊断。大数据究竟在哪些领域发挥作用？不同领域的应用逻辑是什么？本任务将走进大数据的应用世界，探索其背后的价值实现路径。

任务分析

在本任务中，我们将了解大数据在各行业的典型应用场景，通过学习电商、交通、

医疗、金融等重点领域的大数据应用案例，明确不同场景下的数据来源、处理技术和应用价值；通过实操与讨论，提升识别身边大数据应用的能力，深化对大数据“价值增值”本质的理解。

任务实施

1. 行业应用调研：分组并选择一个行业（如电商、交通、医疗、金融、教育、制造），调研该行业的大数据应用案例。

2. 场景还原分析：选择一个身边的大数据应用（如外卖平台调度、短视频推荐），绘制其“数据流转流程图”，标注数据从采集到价值输出的关键环节。

3. 小组汇报：每组推选代表分享调研结果和流程图，讨论“大数据如何改变该行业的传统模式”。

知识讲解

大数据的应用已渗透到社会经济的各个领域，其核心逻辑是通过“数据采集→清洗处理→分析挖掘→决策优化”的闭环，实现效率提升、成本降低或体验升级。以下是大数据的六大核心应用领域的详细解析。

一、电商零售领域

电商零售是大数据应用最成熟的领域，核心目标是实现“人货精准匹配”和“运营效率优化”。

1. 核心应用场景

个性化推荐：基于用户浏览记录、购买历史、搜索关键词、收藏加购行为等数据，构建用户画像（如“年轻女性、偏好轻奢美妆”）和商品画像（如“保湿型面霜、中端价位”），通过协同过滤、深度学习等算法，向用户推送个性化商品（淘宝的“猜你喜欢”、京东的“为你推荐”）。

智能库存管理：整合历史销售数据、市场趋势、促销计划、天气数据等，通过预测算法（如 ARIMA、LSTM）预测商品销量，优化库存补货策略，减少滞销和缺货。例如，京东通过大数据预测，将商品提前部署到离消费者最近的仓库，实现“次日达”。

精准广告投放：根据用户的地域、年龄、消费偏好、行为轨迹等特征，定向推送广告，提高广告转化率。例如，某服装品牌通过大数据定向投放，广告 ROI（投资回报率）提升 200%。

2. 应用价值

电商平台个性化推荐使转化率提升 30% 以上；智能库存管理使库存周转天数缩短 20%～30%；精准广告投放使企业营销成本降低 40% 左右。

二、智能交通领域

大数据推动交通领域从“被动管理”向“主动优化”转型，构建智慧交通体系。

1. 核心应用场景

智能导航：实时采集用户 GPS 数据、道路传感器数据、交通摄像头数据、天气数据，通过流量分析算法预测拥堵点，为用户规划最优路径（如高德地图的“实时避堵”、百度地图的“智能推荐路线”）。

交通信号优化：基于历史交通数据和实时流量，动态调整信号灯配时。例如，深圳智能交通系统通过大数据优化，使主干道拥堵时长减少 20%，通行效率提升 15%。

网约车调度：实时分析用户打车需求、司机位置、路况信息，通过派单算法实现“就近派单”，缩短用户等待时间和司机空驶里程。例如，滴滴通过大数据调度，将平均候车时间缩短至 3 分钟以内。

2. 应用价值

智能导航使用户通勤时间平均缩短 10%～15%；交通信号优化使道路通行能力提升 20%；网约车调度使司机空驶率降低 10%～15%。

三、健康医疗领域

大数据推动医疗模式从“经验医疗”向“精准医疗”转型，提升诊断效率和健康管理水平。

1. 核心应用场景

临床诊断辅助：整合电子病历、检验报告、医学影像（CT、X 光片）、基因数据、药物数据等，通过机器学习算法（如卷积神经网络 CNN）构建疾病诊断模型，辅助医生识别病灶、判断病情。例如，某医院通过大数据分析医学影像，肺癌早期诊断准确率提升 15%。

疾病预测与预防：分析居民健康数据（如体检报告、生活习惯、基因信息）、环境数据（如空气质量、水质）、传染病数据等，预测疾病发生风险（如糖尿病、高血压），推送个性化预防建议。例如，某健康 App 通过用户运动、饮食数据，预测肥胖风险并提供饮食方案。

医疗资源优化：分析医院就诊数据、患者流量、医护人员配置等，优化门诊排班、

床位分配，减少患者排队等待时间。例如，某三甲医院通过大数据调度，将平均挂号等待时间缩短至 30 分钟以内。

2. 应用价值

临床诊断辅助使误诊率降低 10%～20%；疾病预测与预防使慢性病发病率降低 8%～12%；医疗资源优化使医院运营效率提升 30%。

四、金融服务领域

大数据是金融行业“风险控制”和“服务升级”的核心支撑，广泛应用于信贷、风控、投顾等场景。

1. 核心应用场景

信贷审批：整合用户银行流水、交易记录、信用报告、社交媒体行为、消费数据等，通过风控模型评估用户信用等级和还款能力，实现快速审批。传统信贷审批需 3～7 天，大数据审批可缩短至几分钟。

欺诈检测：实时监测用户交易数据（如交易金额、地点、时间、设备），识别异常行为（如异地大额转账、频繁换设备登录），实时预警欺诈风险。例如，信用卡盗刷实时预警系统每年为银行减少数十亿元损失。

智能投顾：根据用户风险承受能力、理财目标、收入状况等数据，推荐个性化理财组合（如基金、股票、债券配比）。例如，支付宝“余额宝”智能投顾为普通用户提供低门槛理财服务。

2. 应用价值

大数据信贷审批使坏账率下降 30%；欺诈检测使金融机构损失减少 25%～40%；智能投顾使普通用户理财收益率提升 5%～8%。

五、教育领域

大数据推动教育从“批量教学”向“精准教学”转型，实现个性化学习和教育资源优化。

1. 核心应用场景

个性化学习：分析学生答题数据、学习时长、错题分布、知识点掌握情况等，构建学生学习画像，推送个性化学习资源和习题（如作业帮的“智能题库”、新东方的“个性化辅导方案”）。

教学效果评估：分析教师授课数据、学生课堂互动数据、考试成绩数据等，评估教学效果，优化教学方法。例如，某中学通过大数据分析，发现某知识点学生掌握率

低，教师调整教学策略后，学生掌握率提升 40%。

教育资源均衡：通过大数据整合优质教育资源（如名师课件、公开课），向偏远地区推送，缩小教育差距。例如，国家中小学智慧教育平台通过大数据调度，覆盖全国上亿名学生。

2. 应用价值

个性化学习使学生学习效率提升 20%～30%；教学效果评估使教师教学质量提升 15%；教育资源均衡使偏远地区学生成绩平均提升 10%。

六、制造领域

大数据推动制造业从“传统生产”向“智能制造”转型，实现生产效率的提升和成本的降低。

1. 核心应用场景

设备预测性维护：通过传感器采集设备运行数据（如温度、振动、电压），分析设备故障规律，提前预警故障风险，安排维护计划，避免停机损失。例如，某汽车工厂通过大数据预测维护，使设备故障率下降 30%，维护成本降低 25%。

生产流程优化：分析生产过程数据（如工序耗时、物料消耗、产品合格率），优化生产流程和工艺参数，提升生产效率。例如，某电子厂通过大数据优化，使生产效率提升 20%，产品合格率从 95% 提升至 98%。

供应链优化：整合原材料价格数据、库存数据、市场需求数据、物流数据等，优化供应链采购、仓储、物流环节，降低供应链成本。

2. 应用价值

设备预测性维护使企业停机损失减少 40%；生产流程优化使生产成本降低 15%～20%；供应链优化使库存成本降低 20%～30%。

项目小结

本项目首先介绍了大数据的产生背景与发展历程，梳理大数据诞生的时代诱因与各阶段发展脉络；然后讲解大数据的定义、分类与核心特征，明确大数据的概念边界、数据划分方式及典型特征；最后介绍大数据的各类主流应用场景，展示大数据在不同行业领域的落地实践。

通过本项目的学习，学习者能够建立对大数据的基础认知，理解大数据的由来、核心概念与实际落地价值，为后续大数据相关技术的学习打下基础。

项目工单

大数据基础认知综合实训

专业		班级		组别	
姓名		学号		成绩	
实训任务	大数据基础认知综合实训				
实训步骤	（1）小组共同回顾项目一三大任务的核心知识点，梳理知识框架，明确各知识点间的逻辑关联。 （2）围绕三大任务，收集典型案例，完成对应分析表的填写，确保案例与知识点精准匹配。 （3）整合分析表内容，撰写综合实训报告，总结实训收获、问题反思及改进建议。 （4）小组内部审核实训成果，修正错误、完善内容，按要求提交给指导教师。				
实训中遇到的困难及解决办法					
思考与总结					
教师审阅意见： 签名： 年　月　日					

课后练习

一、选择题

1. 以下哪项技术的发展是大数据产生的核心驱动力之一？（　　）

A. 传统纸质记录技术　　B. 计算机存储与网络通信技术

C. 手工数据统计方法　　D. 单机运算技术

2. 大数据发展历程中，“数据爆炸”现象开始凸显的阶段是（　　）。

A. 20 世纪 50 年代前　　B. 20 世纪 80 年代至 21 世纪初

C. 21 世纪 10 年代后　　D. 20 世纪 60 年代至 70 年代

3. 以下属于非结构化数据的是（　　）。

A. 企业财务报表中的数字　　B. 电商平台的订单信息

C. 社交媒体上的用户评论文本　　D. 学生成绩统计表

4. 关于大数据的定义，以下表述最准确的是（　　）。

A. 指数据量超过 100GB 的数据集

B. 指无法用传统工具处理的海量、多样、高速产生的数据集合及其相关技术体系

C. 仅指互联网行业产生的用户行为数据

D. 指只能通过云计算处理的数据

5. 以下哪项不属于大数据在金融行业的典型应用？（　　）

A. 信用风险评估　　B. 高频交易策略优化

C. 客户精准营销　　D. 生产设备故障维修

二、简答题

1. 简要说明互联网、物联网技术的发展与大数据产生之间的关系。

2. 简述大数据发展历程中，从“数据积累”向“数据价值挖掘”转变的核心原因。

3. 解释大数据“4V”特征的具体含义，并举例说明其中的“Variety（多样性）”特征在生活中的体现。

4. 简述大数据按照数据的结构化程度划分主要有哪些类别，并各举一个实际例子。

5. 结合生活实际，举例说明大数据在电商行业和交通出行领域的具体应用，并分析其为用户带来的便利。

项目二 大数据软件架构

项目导读

在大数据技术体系中，软件架构是支撑海量数据存储、高效处理与价值挖掘的核心骨架，直接决定了大数据应用的性能、扩展性与稳定性。随着数据规模的爆炸式增长和业务需求的多样化（从海量批处理到实时监控预警），单一架构已无法满足所有场景需求，Hadoop、Spark、实时流处理等多样化架构应运而生，成为大数据工程实践的基础支撑。

学习目标

知识目标	1. 掌握 Hadoop 架构的核心组件（如 HDFS、MapReduce、YARN）及各组件功能，理解分布式存储与计算的核心逻辑。 2. 熟悉 Spark 架构的核心模块（如 Spark Core、Spark SQL、Spark Streaming、GraphX、MLlib）。 3. 了解实时流处理架构的核心特性及数据处理流程，区分批处理与流处理的适用场景。
能力目标	1. 能够梳理 Hadoop、Spark、实时流处理架构各组件交互流程，独立绘制三类架构拓扑示意图。 2. 可根据业务数据处理需求，合理选择匹配的大数据架构，分析对应架构的数据处理优势。 3. 具备识别离线批处理、内存计算、实时流式业务场景的能力，能简单分析不同场景对应的架构部署方案。 4. 能够小组协作完成架构对比调研，完成架构分析报告撰写与成果展示，提升逻辑梳理与表达能力。

素质目标	1. 建立“架构适配场景”的工程思维，养成从业务需求出发选择技术方案的习惯，避免盲目追求“新技术”。 2. 培养对大数据架构技术的探索兴趣与持续学习意识，适应架构技术快速迭代的行业特点。 3. 强化团队协作与技术沟通能力，能清晰表达架构选择逻辑与技术方案优势，提升工程实践中的协同效率。 4. 树立技术选型的成本意识与合规意识，在架构选择中兼顾性能需求与资源合理配置，践行高效务实的工程理念。

思政目标

1. 结合国产大数据架构（如华为 FusionInsight、阿里 MaxCompute）的发展成果，增强科技自信与民族自豪感，激发投身国产技术创新的使命感。

2. 认识大数据架构在数字经济、国家治理中的底层支撑作用，理解技术创新服务国家发展战略的重要意义。

3. 培养精益求精的工匠精神，在架构学习与选型中注重细节、严谨务实，树立“技术向善”的价值导向，确保架构设计兼顾安全性与可持续性。

4. 强化网络安全与数据主权意识，在架构学习中关注数据存储、传输的安全合规要求，践行维护国家数据安全的责任担当。

任务一　Hadoop 架构

任务导入

某电商平台每天产生 PB 级用户交易数据、浏览日志和商品信息，需要长期存储并进行批量分析（如月度销量统计、用户消费行为分析）。传统的单机存储与计算模式已无法满足如此海量的数据处理需求，技术团队决定采用 Hadoop 架构搭建分布式数据处理平台。那么，Hadoop 架构由哪些核心组件构成？各组件如何协同实现海量数据的存储与批处理？其分布式架构的核心设计思想是什么？本任务将揭开 Hadoop 架构的神秘面纱。

任务分析

在本任务中，我们将了解 Hadoop 架构的核心组成与工作原理，掌握 HDFS（Hadoop 分布式文件系统）、MapReduce（分布式计算框架）、YARN（资源调度系统）三大核心组件的功能定位与交互逻辑；掌握 Hadoop 运行环境，以及 Hadoop 集群的安装与部署，为后续架构应用与优化奠定基础。

任务实施

1. 分组收集资料：以“Hadoop 架构技术”为主题，分组搜集 Hadoop 相关学习资料，梳理 Hadoop 简介、Hadoop 家族各组件功能，重点分析 Hadoop2.0 集群架构的组成，理清各模块之间协作关系。

2. 组件拆解与绘图：分组梳理 Hadoop 三大核心组件（HDFS、MapReduce、YARN）的核心功能、关键角色（如 HDFS 的 NameNode、DataNode），绘制 Hadoop 架构组件交互图，标注组件间的数据流转路径。

3. 讨论分享：每组展示架构图与批处理流程图，讨论 Hadoop 架构协同工作原理，分享对分布式存储与计算核心思想的理解。

知识讲解

一、Hadoop 简介

Hadoop 是 Apache 软件基金会旗下的一个开源分布式计算平台，为用户提供系统底层细节透明的分布式基础架构。该平台以分布式文件系统（DFS）和 MapReduce 为核心，能够对大量数据进行数据挖掘、数据分析、数据存储、数据管理与维护，具有可靠、高效、可伸缩的特点。Hadoop 是基于 Java 语言开发的，具有很好的跨平台特性，并且可以部署在低成本的计算机集群中。

几乎所有主流厂商都围绕 Hadoop 提供开发工具、开源软件、商业化工具和技术服务，如谷歌、微软、思科、淘宝等。

Apache Hadoop 版本分为两代：第一代是 Hadoop 1.0；第二代是 Hadoop 2.0。Hadoop 1.0 包含 0.20.x、0.21.x 和 0.22.x 三大版本，其中，0.20.x 最后演化成了 1.0.x，变成了稳定版，而 0.21.x 和 0.22.x 则增加了 HIDES HA 等重要的新特性；Hadoop 2.0 包含 0.23.x 和 2.x 两大版本，它们完全不同于 Hadoop 1.0，是一套全新

的架构，均包含 HDFS Federation（HDFS 联盟）和 YARN 两个组件。

二、Hadoop 家族成员

Hadoop 的核心组件包含 Hadoop Common、HDFS 和 MapReduce（Hadoop 2.0 还包括 Hadoop YARN）。随着 Hadoop 自身的不断发展和完善，产生了与 Hadoop 密切相关的数据服务类子项目（如 HBase、Hive、Pig、Catalog、Sqoop、Flume、Chukwa 等）、运行维护类子项目（如 Ambari、Oozie、Zookeeper 等）和其他相关类子项目（如 Avro、Mahout 等）。

①Hadoop Common（Hadoop 公共服务模块）。Hadoop Common 是 Hadoop 体系中最底层的模块，为 Hadoop 各子项目提供开发所需的应用程序编程接口（Application Program Interface，API）。在 Hadoop 0.20 及以前的版本中，Hadoop Common 包含 HDFS、MapReduce 和其他项目公共内容，从 Hadoop 0.21 开始，HDFS 和 MapReduce 被分离为独立的子项目，其余内容为 Hadoop Common，如系统配置工具 Configuration、远程过程调用 RPC、序列化机制等。

②HDFS（Hadoop 分布式文件系统）。HDFS 是 Hadoop 体系中数据存储管理的基础，具有高容错性，可以用在低成本的通用硬件上。HDFS 提供了一个可扩展、高可靠、高可用的大规模数据分布式存储管理系统，基于物理上分布在各个数据存储节点的本地 Linux 系统的文件系统，为上层应用程序提供一个在逻辑上成为整体的大规模数据存储文件系统。

③MapReduce（并行计算框架）。MapReduce 是一种计算模型，能自动完成计算任务的并行化处理，自动划分计算数据和计算任务，用于海量大数据的计算。MapReduce 框架可以实现 Map 和 Reduce 两种操作。其中，Map 对数据集上的独立元素进行指定操作，生成“键值对”形式的中间结果；Reduce 则对中间结果中相同“键”对应的所有“值”进行规约，以得到最终结果。

④YARN（资源管理框架）。YARN 是新一代 Hadoop 资源管理器，用户可以运行和管理同一个物理集群机上的多种作业，例如 MapReduce 批处理和图形处理作业。YARN 可以对集群中的各类资源进行抽象，并按照一定的策略将资源分配给应用程序或服务。

⑤HBase（分布式列存储数据库）。HBase 即 Hadoop Database，是一个针对结构化数据的，可伸缩、高可靠、高性能、分布式和面向列的动态模式数据库。HBase 主要用于对大规模数据的随机、实时读写访问，并且 HBase 中保存的数据可以使用 MapReduce 进行处理，它可以将数据存储和并行计算完美地结合在一起。

⑥Hive（数据仓库）。Hive 是基于 Hadoop 的数据仓库工具，可以将结构化的数据文件映射为一张数据库表，通过类 SQL 语句快速实现简单的 MapReduce 统计。Hive

无须开发专门的 MapReduce 应用，适用于数据仓库的统计分析。

⑦Pig（脚本语言工具）。Pig 是一款基于 Hadoop 的大规模数据分析工具，它提供的类 SQL 语言称为 Pig Latin，该语言的编译器会把类 SQL 的数据分析请求转换为一系列经过优化处理的 MapReduce 运算。

⑧Catalog（元数据存储与管理系统）。Catalog 是基于 Hadoop 搭建的数据表和存储管理系统，提供了更好的数据存储抽象和元数据服务。

⑨Sqoop（数据库同步工具）。Sqoop 是 SQL-to-Hadoop 的缩写，主要用于 Hadoop 和关系数据库之间的数据传输，可以将关系数据库（如 MySQL、Oracle、Postgres 等）中的数据导入 Hadoop 的 HDFS 中，也可以将 HDFS 中的数据导出到关系数据库。

⑩Flume（日志收集工具）。Flume 是一个分布的、可靠的、高可用的海量日志聚合系统，用于日志数据的收集、处理和传输。

⑪Chukwa（分布式数据采集系统）。Chukwa 是一个开源的数据收集系统，用于监控和分析大型分布式系统的数据，Chukwa 构建在 Hadoop 的 HDFS 和 MapReduce 框架之上，继承了 Hadoop 的可扩展性。Chukwa 还包含一个灵活且强大的工具集，用于展示、监控和分析已收集的数据。

⑫Ambari（部署管理工具）。Ambari 是一种基于 Web 的工具，也是 Hadoop 核心管理组件之一，支持 Apache Hadoop 集群的供应、管理和监控。Ambari 目前兼容大多数 Hadoop 组件，如 HDFS、MapReduce、Hive、Pig、HBase、ZooKeeper、Sqoop 和 HCatalog 等。

⑬Oozie（作业流调度系统）。Oozie 是一个工作流引擎服务器，用于管理和协调 Hadoop 平台上的 HDFS、MapReduce 和 Pig 任务工作流；同时，Oozie 还是一个 Java Web 程序，运行在 Java Servlet 容器中。

⑭ZooKeeper（分布式协调服务）。ZooKeeper 是一个为分布式应用设计的分布、开源的协调服务，主要用来解决分布式应用中经常遇到的一些数据管理问题，以降低分布式应用协调及管理难度，提供高性能的分布式服务。

⑮Avro（数据序列化系统）。Avro 可以将数据结构或者对象转换成便于存储和传输的格式，适合大规模数据的存储与交换。Avro 提供了丰富的数据结构类型、快速可压缩的二进制数据格式、存储持久性数据的文件集、远程调用 RPC 和简单动态语言集成等功能。

⑯Mahout（数据挖掘库）。Mahout 是一个基于 Hadoop 的机器学习和数据挖掘的分布式框架。Mahout 使用 MapReduce 实现了聚类、分类、推荐引擎（协同过滤）和频繁集挖掘等广泛使用的数据挖掘方法。除了算法，Mahout 还包含数据的输入/输出工具、与其他存储系统（如数据库、MongoDB 或 Cassandra）集成等数据挖掘支持架构。

三、Hadoop 2.0 生态系统的集群架构

Hadoop 集群使用了 Master/Slave 架构模式，其集群架构主要由三部分组成：管理节点（MasterNode）、数据节点（SlaveNode）和客户端（Client）。管理节点主要负责管理集群节点，实现实体分配、负载均衡以及数据节点的失败回复，并负责管理整个集群的 Meta 信息，同时提供 Meta 信息服务；数据节点主要负责处理实际任务，如数据存储、责任无执行等，并通过心跳机制向管理节点定期汇报状态、工作进度等信息；客户端主要负责缓存集群以及 Meta 信息，避免与管理节点频繁通信，并且可以读写 API、进行批量操作等。

Hadoop 2.0 生态系统的集群架构主要以 MapReduce、HDFS 和 YARN 为核心，但总体上仍然采用 Master/Slave 结构，如图 2-1 所示。

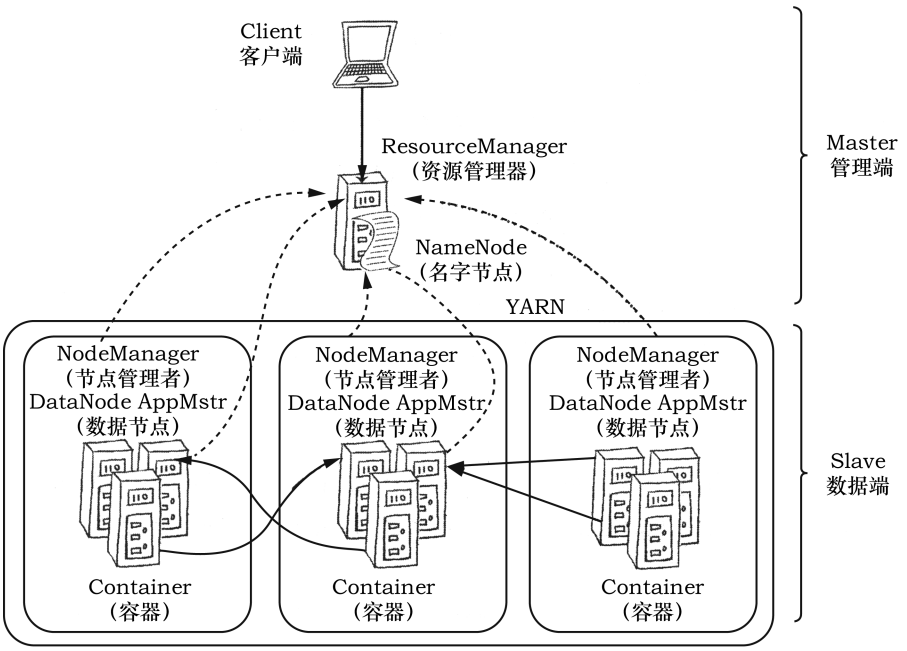


图 2-1　Hadoop 2.0 生态系统的集群架构

从图 2-1 中可以看出，在整个集群中，YARN 为独立的资源管理与分配的通用系统，主要由 ResoureeManager、NodeManager、DataNode AppMstr 以及 Container 等组件构成。其中，ResourceManager 是 Master/Slave 结构中的 Master，NodeManager 是 Master/Slave 结构中的 Slave，ResourceManager 负责对各个 NodeManager 上的资源进行统一管理和调度。

四、Hadoop 运行环境

要想保证 Hadoop 集群能够充分发挥作用，就需要相应的软硬件及网络的支持。

1. 硬件环境

虽然 Hadoop 的一个优势是其可以运行在普通的商用服务器上，但这并不意味着 Hadoop 对硬件环境没有太高要求。用户需要对自己所需处理的问题有全面的了解，并根据 Hadoop 上运行的应用程序的特性确定其硬件环境。例如，对于机器学习、数据挖掘等计算密集型的应用，需要选用计算性能较高的商用服务器；对于索引、检索、统计、聚类等 I/O（Input/Output，输入/输出）密集型的应用，则需要选用 I/O 性能较好的商用服务器等。因此，Hadoop 的硬件环境虽因具体需求的不同而有所差异，但必须保证 Hadoop 运行的硬件环境满足最基本的稳定性和性能要求。

在 Hadoop 生态系统的集群架构中，各节点对硬件环境的要求也不完全相同。对于 Master 节点而言，一旦其出现故障，集群所提供的服务就会中断；对于 Slave 节点而言，其崩溃属于正常现象，不会对集群的可用性造成太大影响；对于客户端节点而言，其故障将会影响作业的批量操作。

2. 软件环境

Hadoop 不仅需要硬件环境的支撑，同样也需要软件环境的支撑。其中，Hadoop 的软件环境主要包括支撑 Hadoop 运行的操作系统、运行环境和节点之间的安全通信协议。

（1）操作系统

由于 Hadoop 是在 Linux 环境下开发的，因此一般会选择 Linux 操作系统运行。任何支持 Java 1.6 的 Linux 操作系统都可以运行 Hadoop，如 Red Hat Enterprise Linux、CentOS、Ubuntu Server Edition、SuSE Enterprise Linux、Debian、Oracle Linux 等。此外，从 Hadoop 2.0 生态系统开始，Hadoop 已经支持在 Windows 操作系统上运行，并且 Horton Works 和 Microsoft 公司合作开发的 HDP 的 Hadoop 版本拥有 Windows 发布版。因此，操作系统的选择主要取决于该系统对硬件的支持能力、系统管理人员对系统的熟悉程度和对目前所使用的商业软件的支持能力等。

目前，Hadoop 系统仍大多运行在 Linux 操作系统上，并已在由 2000 个节点的 GNU/Linux 主机组成的集群系统上得到了验证。在 Windows 环境中安装的 Hadoop 仅作为开发平台支持，因为分布式操作尚未在 Windows 平台上充分测试，所以其只是以学习和研究为目的，暂时不建议将其作为生产平台使用。

（2）运行环境

Hadoop 系统本身是用 Java 语言编写的，但也有少量的 C/C++代码。因此，Ha-

doop 的正常运行需要 JDK（Java Development Kit）的支持。在安装 JDK 时，不建议只安装 JRE（Java Runtime Environment），建议直接安装 JDK。因为 Hadoop 中的 MapReduce 程序的编写和 Hadoop 的编译都需要使用 JDK 中的编译工具，而 JRE 无法满足需求，并且安装 JDK 时还可以同时安装 JRE。

（3）安全通信协议

Hadoop 是一个集群环境，当集群中的管理节点（MasterNode）需要对集群中的其他节点的服务进程进行远程启动和停止时，就需要使用 SSH 协议。SSH 协议能够启动远程命令，通过 SSH 协议用户能够在一个中心的管理节点上远程启动集群中的其他节点的服务进程。例如，名称节点（NameNode）使用 SSH 无密码登录并启动数据节点（DataNode）进程；同样，DataNode 也能使用 SSH 无密码登录 NameNode。

3. 网络环境

Hadoop 中的 MapReduce 在执行作业调度时需要进行 Map 和 Reduce 两个过程，虽然在 Map 阶段进行任务调度时会尽量使任务本地化，但 Reduce 过程仍会产生大量的 I/O，因此 Hadoop 集群网络在任意节点之间的带宽需求都很高，网络拓扑结构一般采用层级很少的 Fabric 网络，以提高集群的网络性能。另外，为使 Hadoop 集群能够正常运行，建议至少使用千兆以太网进行连接，并配置大容量的网络交换机。

五、Hadoop 集群的安装与部署

Hadoop 包括单机模式、伪分布模式和分布式模式三种安装模式。单机模式只在一台计算机上运行；存储采用本地文件系统，而非 HDFS。伪分布模式同样只在一台计算机上运行；存储采用 HDFS，但 HDFS 的名称节点（NameNode）和数据节点（DataNode）都位于同一台计算机上。分布式模式在多台计算机上运行；存储采用 HDFS，且 HDFS 的名称节点（NameNode）和数据节点（DataNode）位于不同的计算机上。在分布式模式下，数据可分布到多个节点上，不同数据节点上的数据计算可以并行执行，这时 MapReduce 的分布式计算能力才能真正发挥作用。

下面以分布式模式为例说明 Hadoop 的部署过程，以方便读者深入理解 Hadoop 的体系架构。

1. 集群环境搭建

为降低分布式模式部署的难度，本书仅使用 2 个节点（2 台计算机）搭建集群环境：一台计算机作为 Master 节点，其局域网 IP 地址为 192.168.255.101；另一台计算机作为 Slave 节点，其局域网 IP 地址为 192.168.255.102。如果是由 3 个或 3 个以上的节点构成的集群，也采用类似的方法完成安装和部署。

2. Hadoop 集群的安装与配置

Hadoop 集群的安装与配置大致包括以下步骤。

①在 Hadoop 官网（http://hadoop.apache.org/releases.html）下载相应版本的 Hadoop，本书采用的版本是 2.7.5，Linux 操作系统采用 CentOS 7 版本。

②选择一台计算机作为 Master 节点，并把主机名改为 Master，设定其局域网 IP 地址为 192.168.255.101。

③在 Master 节点上创建 hadoop 用户，然后安装 SSH 服务端和配置 Java 环境。

④在 Master 节点上安装 Hadoop 2.7.5，并完成基本配置。

⑤选择另一台计算机作为 Slave 节点，并把主机名改为 Slave，设定其局域网 IP 地址为 192.168.255.102（若有多台 Slave 节点计算机，则最好将主机名对应地设置为“Slave+序号”，每台 Slave 节点的计算机的局域网 IP 地址也不同）。

⑥在 Slave 节点上创建 hadoop 用户，然后安装 SSH 服务端和配置 Java 环境（本例仅有一个 Slave 节点，如果有多个 Slave 节点，则需要在多台对应的计算机上完成安装）。

⑦将 Master 节点上的/usr/local/hadoop 目录复制到其他 Slave 节点上（本例仅有一个 Slave 节点，此步骤只需要执行一次，若有多台 Slave 节点，则需要重复操作多次）。

⑧在 Master 节点上开启 Hadoop。

在上述步骤中，hadoop 用户创建、SSH 服务端安装、Java 环境配置、Hadoop 安装等操作，只需要参考相关软件的安装说明即可顺利进行，这里不再赘述。完成①～⑥的操作后，才可以继续进行分布式模式的部署。

3. 分布式模式的部署

（1）配置网络

假设集群所用的 2 个节点（2 台计算机）都位于同一个局域网（LAN）内，Master 节点计算机的 IP 地址为 192.168.255.101，Slave 节点计算机的 IP 地址为 192.168.255.102。

由于集群中只有 2 台计算机需要设置，所以在接下来的操作中一定要注意区分 Master 节点和 Slave 节点。因为步骤②和步骤⑤已经为 2 台计算机修改了主机名，这样在终端窗口的标题和命令行中就可以看到主机名了，从而比较容易区分当前在对哪台计算机进行操作。

①修改 Master 节点中的/etc/hosts 文件。

执行编辑命令，打开并修改 Master 节点中的/etc/hosts 文件。

```
$ sudo vim/etc/hosts
```

在 hosts 文件中增加主机名和对应局域网 IP 的映射关系，本例仅有一个 Master 主机和一个 Slave 主机。

```
192.168.255.101   Master
192.168.255.102   Slave
```

修改完成后重新启动 Master 节点上的 Linux 操作系统，即可完成 Master 节点的配置。

②修改 Slave 节点的配置。

本例只有一个 Slave 节点，主机名为 Slave。参照上述方法修改 Slave 节点上的 /etc/hosts 文件中的内容，在 hosts 文件中增加如下两条 IP 和主机名的映射关系。

```
192.168.255.101   MaSter
192.168.255.102   Slave
```

修改完成后重新启动 Slave 节点上的 Linux 操作系统，即可完成 Slave 节点的配置。

③测试。

完成 Master 和 Slave 节点的配置后，在各个节点上执行 ping 命令，以测试各个节点彼此是否相互 ping 得通，若 ping 不通，则无法顺利配置成功。

（2）设置 SSH 无密码登录节点

为了确保 Master 节点可以通过 SSH 无密码登录到各个 Slave 节点上，需要进行如下设置。

①生成 Master 节点的公钥。

若 Master 节点上已经存在公钥文件，则必须删除原来已存在的公钥再重新生成一次。在 Master 节点上执行如下命令。

```
$ cd ~/.ssh      # 若该目录不存在，则必须先执行一次 ssh localhost
$ rm ./id_rsa*     # 若此前存在公钥，则需要执行命令删除它；若不存在，则可忽略此命令
$ ssh-keygen -t rsa     # 执行该命令后，遇到提示信息一直按 Enter 键即可
```

操作完成后，在 Master 节点上生成了两个文件：一个私钥文件和一个公钥文件。

②SSH 无密码登录本机。

为了确保 Master 节点能够无密码登录本机，还需要在 Master 节点上执行如下命令。

```
Scat ./id_rsa.pub >> ./authorized_keys
```

操作完成后，执行 ssh Master 命令进行验证，若遇到提示信息，则直接输入 yes。测试成功后，执行 exit 命令即可返回原来的终端。

③公钥传输。

在 Master 节点上将公钥传输到 Slave 节点，操作命令如下。

```
scp ~/.ssh/id_rsa.pub Hadoop @Slave: home/hadoop/
```

命令 scp 是 secure copy 的简写，用于在 Linux 环境下远程复制文件。执行 scp 时

会要求输入 Slave 节点上的 hadoop 用户的密码，输入后会提示传输完毕。

④公钥加入授权。

在 Slave 节点上将 SSH 公钥加入授权，操作命令如下。

```
$ mkdir ~/.ssh      #如果文件夹不存在，则需要先创建文件夹；若已存在，则忽略本条命令
$ cat ~/id_rsa.pub >> ~/.ssh/authorized_keys
$ rm ~/id_ras.pub      #用完以后即可删除
```

本例只有一个 Slave 节点，所以步骤③和步骤④只需要执行一次。如果有 2 个或 2 个以上的 Slave 节点，则需要将 Master 公钥传输到所有其他 Slave 节点（注意：需要修改传输命令中对应 Slave 节点的主机号），并给各 Slave 节点授权。

⑤测试。

在 Master 节点上执行以下命令。

```
$ ssh Slave
```

（3）配置 PATH 变量

在 Master 节点上配置 PATH 变量，通过设置 PATH 变量，可以在执行命令时不用带上命令本身所在的路径。首先执行命令 vim ~/.bashrc，用 vim 编辑器打开 ~/.bashrc 文件，然后在该文件最上面的位置处添加如下内容。

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

保存后执行命令 source ~/.bashrc，使配置生效。

（4）配置集群

配置集群时，需要修改对应主机上的/usr/local/hadoop/etc/hadoop 目录下的配置文件，这里仅设置正常启动所必需的设置项，包括 slaves、core-site.xml、hdfs-site.xml、mapred-site.xml、yarn-site.xml 共 5 个文件，更多的设置项可查看官方说明。我们先在 Master 中修改配置相关的文件，然后将其复制到所有 Slave 节点上即可。

①在 Master 节点上修改 slaves 文件。

slaves 文件记录的是集群中所有 DataNode 的主机名，需要把所有数据节点的主机名写入该文件，每行一个，默认为 localhost（即把本 Master 机也作为数据节点）。在进行分布式配置时，可以保留 localhost，让 Master 节点同时充当名称节点和数据节点，或者也可以删除 localhost 这一行，让 Master 节点仅作为名称节点使用。本例让 Master 节点仅作为名称节点使用，因此将 slaves 文件中原来的 localhost 删除，只添加如下内容。

```
Slave
```

②在 Master 节点上修改 core-site.xml 文件。

core-site.xml 文件是全局配置文件，主要配置的是 HDFS 的地址、端口号和 ha-

doop. tmp. dir 的参数，其主要作用是指定名称节点的地址和在使用 Hadoop 时产生的文件的存放目录。core-site. xml 文件的修改内容如下。

```
<configuration>
<property>
<name>fs. default. name</name>
<value>hdfs：//Master：9000</value>
</property>
<Property>
<name>hadoop. tmp. dir</name>
<value>file：/usr/local/Hadoop/tmp</value>
<description>Abase for other temporary directories. </description>
</property>
</configuration>
```

如果没有配置 hadoop. tmp. dir 参数，则系统会默认配置一个临时目录。

③在 Slave 节点上修改 hdfs-site. xml 文件。

hdfs-site. xml 文件的作用是指定 HDFS 保存数据的副本数量和 HDFS 中名称节点与数据节点的存储位置。对于 Hadoop 的分布式文件系统而言，一般采用冗余存储，冗余因子通常为 3，也就是 1 份数据会保存 3 个副本。但本例只有一个 Slave 节点作为数据节点，即整个集群中只有一个数据节点，数据只能保存一份，所以将 dfs. replication 的值设为 1（如果仍然设置为 3 也没有问题）。hdfs-site. xml 文件的修改内容如下。

```
<configuration>
<Property>
<name>dfs. namenode. secondary. http - address</name>
<value>Master：500 90</value>
</property>
<PropertY>
<name>dfs. replication</name>
<value>1</value>
</property>
<property>
<name>dfs. namenode. name. dir</name>
<value>file：/usr/local/hadoop/tmp/dfs/name</value>
</property>
```

```
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/usr/local/hadoop/tmp/dfs/data</value>
</propertY>
</configuration>
```

④在 Master 节点上修改 mapred-site.xml 文件。

在 Master 节点的/usr/local/hadoop/etc/hadoop 目录下有一个 mapred-site.xml.template 文件，需要修改其文件名称，重命名为 mapred-site.xml。

mapred-site.xml 文件的主要作用是配置使用 YARN 计算框架和历史作业的 IPC 的 IP 端口及历史作业的 Web 端访问地址。mapred-site.xml 文件的修改内容如下。

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
<Property>
<name>mareduce.jobhistory.address</name>
<value>Master:10020</value>
</property>
<Property>
<name>mapreduce.jobhistory.webapp.address</name>
<value>Master:19888</value>
</property>
</configuration>
```

⑤在 Master 节点上修改 yarn-site.xml 文件。

这一步是对 YARN 进行设置，主要配置 ResourceManager 地址及 yarn.application.classpath 等。YARN 由 ResourceManager 和 NodeManager 构成，其中 Master 节点充当 ResourceManager，而 Slave 节点充当 NodeManager。yarn-site.xml 文件的修改内容如下。

```
<configuration>
<property>
<name>yarn.resourcemanager.hostname</name>
<value>Master</value>
```

```
</property>
<Property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce shuffle</value>
</property>
</configuration>
```

⑥在 Master 节点上压缩复制/usr/local/hadoop 文件夹。

配置完上述 5 个文件后，需要把 Master 节点上的/usr/local/hadoop 文件夹压缩复制到各个节点上。具体来说，首先需要在 Master 节点上执行如下命令进行压缩和复制。

```
$ cd/usr/local
$ sudo rm -r. /hadoop/tmp        #删除 Hadoop 临时文件
$ sudo rm -r. /hadoop/logs/*        #删除日志文件
$ tar -zcf~/hadoop.master.tar.gz. /hadoop        #先压缩，再复制
$ cd~
$ scp. /hadoop.master.tar.gz Slave: /home/hadoop
```

然后在 Slave 节点上执行如下命令进行解压缩。

```
$ sudo rm -r/usr/local/hadoop        #删除旧的（若存在）
$ sudo tar -zxf~/hadoop.master.tar.gz -C/usr/loca
$ suao chown -R Hadoop/usr/local/hadoop
```

本例仅有一个 Slave 节点，所以步骤⑥只需要执行一次。如果还有其他 Slave 节点存在，则步骤⑥需要重复执行，必须将 hadoop.master.tar.gz 传输到所有 Slave 节点，并在各个 Slave 节点执行解压文件的操作。因此，当有多个 Slave 节点存在时，在传输命令中需要特别注意 Slave 节点主机名的改变。

（5）启动集群

启动 Hadoop 集群时，首次需要先在 Master 节点上执行名称节点的格式化（只需一次），命令如下。

```
$ hdfs namenode -format
```

然后，启动 Hadoop。启动操作需要在 Master 节点上进行，即执行如下命令。

```
$ start -dfS.sh
$ start -yarn.sh
$ mr -jobhistory -daemon.sh start historyserver
```

最后，可以通过执行 jps 命令查看各个节点（包括 Master 节点和所有 Slave 节点）启动的进程。如果已经正确启动，则在 Master 节点执行 jps 命令时可以看到 Master 节