本书由清华大学、上海大学、河北科技大学、国网河北电科院等科研院所的一线教师和高级工 程师合著,阿里巴巴、腾讯、百度、51talk、昆仑万维、央视网等企业多位工业界同行联合推荐。





V

l



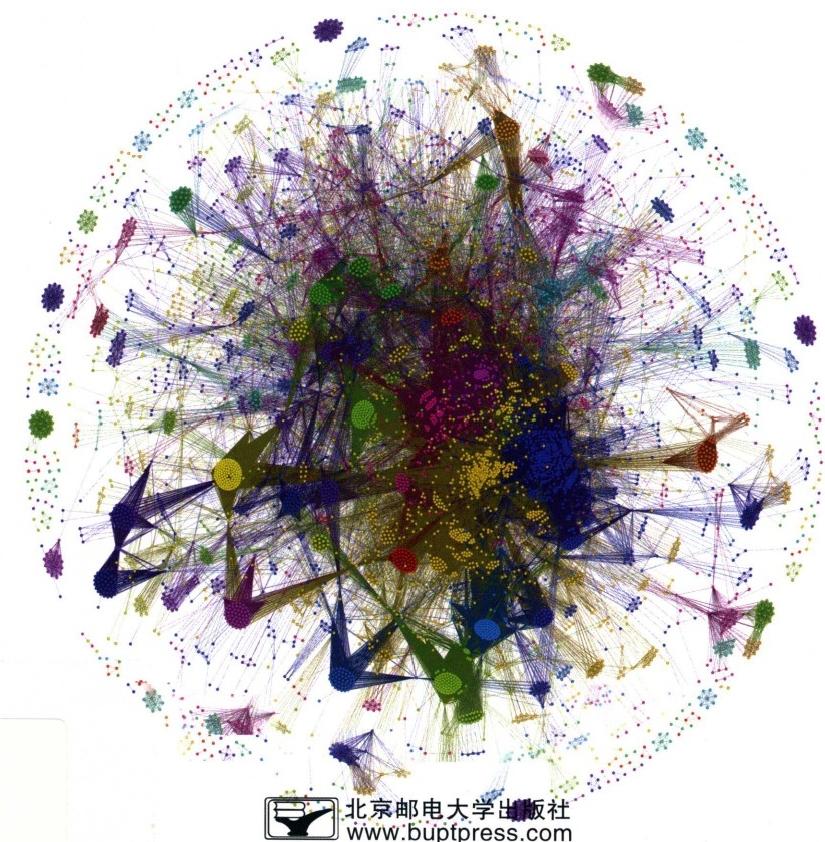
**技术及行业应用**

如何**定** **义**大数据?如何**应** **用大数据?**

什么是大数据思维?如何**学** **习**大数据?

如何**构** **建**大数据平台?如何在行业中**应** **用**大数据?

**许云峰徐华张妍王杨君马瑞◎著**





**大数据技术及行业应用**

许云峰 徐 华 张 妍 王杨君 马 瑞 著

**北京邮电大学出版社**

**www.buptpress.com**



**图书在版编目(CIP)数据**

大数据技术及行业应用/许云峰等著.--北京：北京邮电大学出版社，2016.8

ISBN 978-7-5635-4918-4

I.① 大… Ⅱ.①许… Ⅲ.①数据处理 IV.①TP274

中国版本图书馆 CIP 数据核字(2016)第199031 号

**书** **名：**大数据技术及行业应用

**著作责任者：**许云峰 徐华 张妍 王杨君 马瑞 著

**责** **任** **编** **辑：**王丹丹

**出** **版** **发** **行：**北京邮电大学出版社

**社** **址：**北京市海淀区西土城路10号(邮编：100876)

**发** **行** **部**：电话：010-62282185 传真：010-62283578

**E-mail:** publish@bupt.edu.cn

**经** **销：**各地新华书店

**印** **刷：**保定市中画美凯印刷有限公司

**开** **本：**787 mm×1092 mm 1/16

**印** **张：**16.5

**字** **数：**469千字

**版** **次：**2016年8月第1版 2016年8月第1次印刷

ISBN 978-7-5635-4918-4

**·如有印装质量问题，请与北京邮电大学出版社发行部联系** ·

定 价：35.00元

**序** **一**

自从世界上第一台真正意义上的电子计算机 ENIAC 在宾 夕法尼亚大学的诞生开始，数据的存储与组织就成为了计算机 技术走向实际应用过程中的一个重要课题。在随后的整整七十 年间，以通过高效处理解决实际应用问题为目标的计算机科学 得到了长足的发展，但在增长速度的赛跑中，存储器硬件的容量 远远地被应用问题的规模甩在后面。进入新世纪后，随着这一 矛盾在多样性、价值、速度等方面的日益突出，迫切需要综合已 有技术给出系统性的、可扩展的解决方案，而大数据理论与方法 也自然地应运而生。

近十年来，这方面的专著、教材层出不穷，但摆在案头的这 本《大数据技术及行业应用》却在很多方面令人耳目一新。该书 的五位联合作者虽非名声显赫，但都是来自国内著名高校的青 年才俊，他们活跃在大数据科学的技术前沿，同时善于汲取与融 合来自工业界的先进工程经验和学术界的宝贵学术资源。其中 首章简明地介绍了大数据的基本理论，辨析了相关的重要概念 和理念。接下来的三章，从私有平台、虚拟化平台、综合性平台 等层面系统介绍了现有的成熟技术方法。最后七章是本书的重 点，依次剖析了大数据技术在图算法、环境科学、药物数据聚类、 电子商务、社交网络、文本挖掘与情感分析、电力系统控制等领 域的具体应用。透过这些应用，读者应该能够生动地看到大数 据技术也已取得的巨大成效，以及未来发展的广阔前景。

祝贺五位作者的这次成功合作，更祝愿他们能在这一领域 继续前行，探索和发展出更多的新方法、新技术。

是为序。

邓俊辉

2016年8月于清华园

**序** **二**

2010年，随着中国进入“移动互联网元年”,中国正式进入了移动 端设备的全面上网时代。与传统互联网相比，移动互联网解决了任 何人、任何时间、任何地点以多种方式上网的问题。伴随着移动互联 网的普及，人与人之间、人与物之间、物与物之间实现了全面的互联 互通。通过移动互联网，我们可以获得人和人之间互联的社交数据 (微博、微信等)、人和物之间互联的行为数据(上网日志信息等)、物 与物之间互联的环境数据(智能家居设备采集的环境数据等)。由于 通过移动互联网采集的这些数据，具有高容量(Volume)、高生成速度 (Velocity)、多样性(Variety), 同时具有潜在的应用价值(Value) 等特 点，所以我们常常称之为“大数据”。为了解决“大数据”的存储管理 和计算难题，信息技术领域近些年深入研发了“云”计算技术。“互联 网+”正是在这样一个如何综合利用大数据，实现移动互联网与各个 传统行业的深度融合与创新的背景下应运而生的。

互联网技术与应用的发展对于高校和 IT 行业，特别是信息技 术专业领域的学生和工程技术人员，提出了掌握大数据的处理方法 与技术的基本要求。本书正是针对这样一个技术与应用发展背景 下的要求，系统性的介绍了大数据的相关概念、大数据平台的搭建 与综合解决方案，以及国内相关领域的研究学者在环境、医药、电子 商务、社交网络、文本挖掘和电力系统等分支领域的应用研究成果。

作为教育部支持的高级访问学者，许云峰老师曾在我所在的清 华大学智能技术与系统国家重点实验室从事了为期一年的高级访 问学者研究工作，重点研究了基于社交网络数据的社群发现算法。 作为近年来在国内互联网社交数据挖掘与分析方面较为活跃的研 究学者，他陆续在KBS 和 ESWA 等国际期刊上发表了系列化社群 发现的研究成果。本书既是对许云峰老师过去多年来在大数据应 用方面的技术总结，也是将他对大数据技术的应用经验分享给相关 同学和技术人员的一个很好的形式。相信各位读者能够从中熟悉 大数据技术的深刻内涵。

徐华

2016年8月于清华园

**前** **言**

如何定义大数据?如何应用大数据?什么是大数据思维?如何学习大数据?如何构建 大数据平台?如何在行业中应用大数据?这一系列的问题，是当前在大数据热的时代背景 里，让人感到非常迷茫的问题。本书直面这些问题，在从业者角度解答以上问题，希望能给 大数据行业的初学者提供一些帮助。当然我们的观点并不是放之四海而皆准的唯一真理， 随着大数据行业的发展，会有更全面的答案出现。希望这本书能起到抛砖引玉的作用。

本书第1章阐述了大数据的相关概念。第2章讲解了基于 Hadoop 的私有云平台搭建 案例。第3章讲解了基于OpenStack 和 Docker 的大数据平台的基础虚拟化平台搭建。第 4章讲解了基于CDH 和 HDP 的大数据平台搭建。第5章讲解了基于Spark 平台的大数据 处理应用，并展示了图挖掘中的经典案例。第6章讲解了大数据技术在环境科学中的应用 案例。第7章讲解了一个大数据在 DrugBank 药物数据库聚类方面的应用案例。第8章讲 解了一个大数据在电子商务数据分析中应用的案例。第9章讲解了一个大数据思维在社交 网络数据分析中应用的案例。第10章讲解了大数据技术在情感分类中应用的案例。第11 章讲解了大数据技术在电力数据分析中应用的案例。本书的案例虽然不能囊括大数据应用 的所有领域，但是在不同的角度回答和解决了大部分人在当前大数据环境下面临的问题和 挑战。

本书由清华大学、上海大学、河北科技大学、国网河北电科院等科研院所的一线教师和 高级工程师合著。其中，许云峰完成了第1章、第7章和第9章。徐华完成了第10章，张妍 完成了第3章、第4章、第5章。王杨君完成了第6章。马瑞、陈二松和范辉共同完成了第 11章。许云峰和陈书旺共同完成了第2章。许云峰和李媚共同完成了第8章。

本书由清华大学 MOOC 著名教育专家邓俊辉老师和清华大学智能技术国家重点实验 室徐华老师倾情作序，邓老师多次入选清华大学我印象最深的十大教师(毕业生评选),徐华 老师是我在清华访学期间的指导老师。两位老师治学严谨，是我学习的典范，在此表示深深 的感谢。

本书得到来自工业界同行的宝贵意见和建议，他们是：阿里数据经济研究中心常务副主 任兼秘书长潘永花，腾讯企业云架构师潘晓东，百度云服务架构师董月照，51talk 大数据 专家资深架构师郝伟瑞、刘会山，央视网高级数据分析师孙泉，昆仑万维苗雨顺，北京卓新 思创 CEO 庄严，北京星立方科技发展股份有限公司周长亮，北京图森王路，在此表示诚挚 的感谢。感谢参加本书的编辑和实验工作的同学，他们是：周莹莹、白云、么学媛、赖杰、俞孝 帅、胡江涛、肖明美、刘芳彤、温亚东、李书航、刘利平。同时感谢石家庄京华电子实业有限公 司远松灵总经理提供部分应用场景支持，感谢石家庄吾搜网络科技有限公司提供域名和空 间支持。

全书共46.9万字，许云峰完成了16.9万字，徐华完成了8.2万字，张妍完成了12.1 万字，王杨君完成了3.7万字，马瑞完成1万字，陈书旺完成1万字，李媚完成1万字，陈二 松完成1万字，范辉完成1万字，许云岭完成1万字。

本书在初稿完成后，经多位工业界和学术界同行审阅，获得大家普遍认可。同时各位专 家学者也提出了很多宝贵的意见和建议，我们做了对应修改，但是由于时间仓促，难免会有 疏漏。在本书付梓之际，本人诚惶诚恐，失眠多日，但是在探寻真知的过程中偶得的宝贵知 识和经验不敢私藏，所谓“愚者千虑，必有一得”,希望本书中的工程经验和学术观点能够对 广大读者有所启发，并起到抛砖引玉的效果。同时欢迎读者与我们交流，并提出宝贵意见， 联系信息如下。

本书官方公众号：大数据技术及行业应用



本书官方网站：<http://dxmall.com.cn:2016>

许云峰

2016年8月于河北科技大学

**目** **录**

**第1章** **大数据相关概念** [1](#bookmark2)

1.1 什么是大数据? [1](#bookmark3)

1.2 大数据有多大? [3](#bookmark4)

1.3 大数据是一种思维方式 [3](#bookmark5)

1.4 大数据思维的应用案例 [4](#bookmark6)

1.5 大数据是如何产生的? [6](#bookmark7)

1.6 美国和中国的大数据产业生态系统 [6](#bookmark8)

1.7 如何学习大数据技术 [7](#bookmark9)

本章小结 [8](#bookmark1)

参考文献 [8](#bookmark1)

**第2章** **搭建私有大数据处理平台** [10](#bookmark10)

2.1 FreeBSD 操作系统安装 [10](#bookmark11)

2.2 基础软件安装 [11](#bookmark12)

2.2.1 安装Java 运行环境 [11](#bookmark13)

2.2.2 安装 bash [11](#bookmark14)

2.3 Hadoop 安装配置 [11](#bookmark15)

2.3.1 系统规划 [11](#bookmark16)

2.3.2 配置 conf/ masters 、conf/ slaves 文件 [12](#bookmark17)

2.3.3 Hadoop 安装 [12](#bookmark18)

2.4 Hadoop 开发环境配置 [16](#bookmark19)

2.4.1 编译 Hadoop -eclipse plugin-1.1.2. jar 插件 [16](#bookmark20)

2.4.2 eclipse 配置 [17](#bookmark21)

2.4.3 测试 [17](#bookmark22)

2.5 Hadoop 升级 [18](#bookmark23)

2.6 Zookeeper 安装 [19](#bookmark24)

2.6.1 在 FreeBSD 上安装 Zookeeper [19](#bookmark25)

2.6.2 启动并测试 Zookeeper [20](#bookmark26)

2.7 HBase 安装配置 [21](#bookmark27)

2.8 FreeBSD 上网配置 [26](#bookmark28)

2.8.1 VPN 上网配置 [26](#bookmark29)

2.8.2 网页认证上网配置 [27](#bookmark30)

2.9 配置杀毒软件 [28](#bookmark31)

本章小结 [29](#bookmark32)

**第3章** **大数据平台虚拟化解决方案** [30](#bookmark33)

3.1 Ubuntu 上安装 Docker [30](#bookmark34)

3.1.1 Docker 简介 [30](#bookmark35)

3.1.2 Docker 安装 [31](#bookmark36)

3.1.3 Docker 镜像相关命令 [31](#bookmark37)

3.1.4 Docker 容器相关命令 [32](#bookmark38)

3.1.5 Dockerfile 创建镜像 [34](#bookmark39)

3.1.6 Docker 实现 Spark 集群 [36](#bookmark40)

3.1.7 Docker 集中化 Web 界面管理平台 shipyard [41](#bookmark41)

3.1.8 DockerUI [43](#bookmark42)

3.2 OpenStack 搭建 [45](#bookmark43)

3.2.1 下载工具和镜像 [45](#bookmark44)

3.2.2 配置网桥 [46](#bookmark45)

3.2.3 安装 fuel [47](#bookmark46)

3.2.4 安装 OpenStack 平台 [49](#bookmark47)

3.2.5 使 用OpenStack 平台 [54](#bookmark48)

本章小结 [61](#bookmark49)

参考文献 [61](#bookmark50)

**第** **4** **章** **大数据平台解决方案** [62](#bookmark51)

4.1 大数据平台比较 [62](#bookmark52)

4.2 CDH 大数据平台搭建 [63](#bookmark53)

4.2.1 Cloudera Manager 安装 [63](#bookmark54)

4.2.2 添加服务 [64](#bookmark55)

4.3 HDP 大数据平台搭建 [74](#bookmark56)

4.3.1 部署 Ambari [75](#bookmark57)

4.3.2 用Ambari\_web 部署 HDP 平台 [78](#bookmark58)

本章小结 [86](#bookmark59)

**第5章** **Spark 在大数据处理中的应用** [87](#bookmark60)

5.1 Spark 集群搭建 [87](#bookmark61)

5.1.1 Scala 在 Ubuntu 下的安装和配置 [87](#bookmark62)

5.1.2 Spark 集群搭建 [88](#bookmark63)

5.1.3 Spark 集群启动测试 [89](#bookmark64)

5.2 Spark-shell 统计社交网络中节点的度 [90](#bookmark65)

5.2.1 启动 HDFS 和 Spark [90](#bookmark66)

5.2.2 运行 Spark -shell [91](#bookmark67)

5.2.3 统计社交网络中节点的度 [92](#bookmark68)

5.3 Spark GraphX [94](#bookmark69)

5.3.1 属性图 [95](#bookmark70)

5.3.2 图操作 [98](#bookmark71)

5.3.3 构建图 [108](#bookmark72)

5.3.4 图计算相关算法 [109](#bookmark73)

5.3.5 GraphX 图计算实例 [112](#bookmark74)

本章小结 [113](#bookmark75)

参考文献 [113](#bookmark76)

**第** **6** **章** **大数据技术在环境科学中的应用** [115](#bookmark77)

6.1 大气环境科学的数值模式的介绍 [115](#bookmark78)

6.1.1 气象模式 [115](#bookmark79)

6.1.2 区域空气质量模式 [119](#bookmark80)

6.2 高分辨率实时观测的大数据 [127](#bookmark81)

本章小结 [128](#bookmark82)

参考文献 [128](#bookmark83)

**第** **7** **章** **大数据在** **DrugBank 药物数据库聚类方面的应用** [**130**](#bookmark84)

7.1 简介 [130](#bookmark85)

7.2 开发环境及编程语言 [133](#bookmark86)

7.3 算法设计 [134](#bookmark87)

7.3.1 算法设计流程 [134](#bookmark88)

7.3.2 相似度的计算 [135](#bookmark89)

7.4 算法实现 [138](#bookmark90)

7.4.1 文件的解析 [138](#bookmark91)

7.4.2 对靶标、作用酶的分析 [138](#bookmark92)

7.4.3 对分子中原子百分比的处理过程 [140](#bookmark93)

7.4.4 结果的整合 [145](#bookmark94)

7.4.5 最终结果展示 [146](#bookmark95)

本章小结 [147](#bookmark96)

参考文献 [148](#bookmark97)

**第8章** **大数据在电子商务数据分析中的应用** [150](#bookmark98)

8.1 研究现状 [150](#bookmark99)

8.2 相关技术及概念 [151](#bookmark100)

8.2.1 网络爬虫 [151](#bookmark101)

8.2.2 HtmlUnit 工具包 [152](#bookmark102)

8.2.3 Mahout [152](#bookmark103)

8.2.4 朴素贝叶斯算法 [152](#bookmark104)

8.2.5 文档向量 [153](#bookmark105)

8.2.6 TF- IDF 改进加权 [153](#bookmark106)

8.2.7 中文分词 [154](#bookmark107)

8.3 需求分析 [154](#bookmark108)

8.3.1 系统功能 [154](#bookmark109)

8.3.2 系统界面 [156](#bookmark110)

8.4 概要设计 [157](#bookmark111)

8.4.1 系统模块设计 [157](#bookmark112)

8.4.2 数据库设计 [158](#bookmark113)

8.5 详细设计 [162](#bookmark114)

8.5.1 用户登录模块 [162](#bookmark115)

8.5.2 爬虫管理模块 [163](#bookmark116)

8.5.3 算法管理模块 [165](#bookmark117)

8.5.4 用户管理模块 [166](#bookmark118)

8.6 系统测试 [167](#bookmark119)

8.6.1 训练集准备 [167](#bookmark120)

8.6.2 新数据准备 [168](#bookmark121)

8.6.3 训练模型 [170](#bookmark122)

8.6.4 数据分类 [171](#bookmark123)

8.6.5 分类结果分析 [171](#bookmark124)

本章小结 [173](#bookmark125)

参考文献 [173](#bookmark126)

**第9章** **大数据技术在社交网络研究中的应用** [174](#bookmark127)

9.1 社区发现研究简介 [174](#bookmark128)

9.2 社区发现相关研究工作 [175](#bookmark129)

9.2.1 相关工作 [176](#bookmark130)

9.2.2 研究动机 [177](#bookmark131)

9.3 模型与问题的形式化 [177](#bookmark132)

9.3.1 社区森林模型 [177](#bookmark133)

9.3.2 问题形式化 [179](#bookmark134)

9.4 骨干度算法 [180](#bookmark135)

9.4.1 骨干度算法框架 [181](#bookmark136)

9.4.2 算法的时间复杂度 [183](#bookmark137)

9.4.3 算法比较 [183](#bookmark138)

9.5 实验分析 [183](#bookmark139)

9.5.1 数据集 [183](#bookmark140)

9.5.2 一个特定人际关系网络的测试 [186](#bookmark141)

9.5.3 Zachary 的空手道俱乐部测试 [187](#bookmark142)

9.5.4 美国大学橄榄球队 [189](#bookmark143)

9.5.5 安然电子邮件公司数据集 [189](#bookmark144)

9.5.6 DBLP 合作网络 [191](#bookmark145)

9.5.7 结论 [192](#bookmark146)

本章小结 [192](#bookmark147)

参考文献 [193](#bookmark148)

**第10章** **大数据技术在文本挖掘和情感分类中的应用** [195](#bookmark149)

10.1 研究综述 [195](#bookmark150)

10.1.1 基于产品特征的观点挖掘研究 [195](#bookmark151)

10.1.2 产品评论结构化信息抽取方法 [198](#bookmark152)

10.1.3 评论信息分类相关研究方法 [200](#bookmark153)

10.2 评论文本的结构化信息抽取 [202](#bookmark154)

10.2.1 产品特征抽取 [202](#bookmark155)

10.2.2 基于关联规则抽取评论的隐式特征 [203](#bookmark156)

10.2.3 基于监督学习抽取评论的隐式特征 [207](#bookmark157)

10.3 情感分类研究综述 [209](#bookmark158)

10.3.1 基于词典与语言规则进行情感分类 [209](#bookmark159)

10.3.2 观点挖掘结果归纳 [213](#bookmark160)

10.4 算法评估结果与分析 [215](#bookmark161)

10.4.1 隐式特征抽取实验结果及分析 [215](#bookmark162)

10.4.2 篇章粒度情感分类实验结果及分析 [221](#bookmark163)

10.4.3 语句粒度情感分类实验结果及分析 [222](#bookmark164)

本章小结 [224](#bookmark165)

参考文献 [224](#bookmark166)

**第11章** **大数据技术在电力系统中的应用** [228](#bookmark167)

11.1 一种云可视化机网协调控制响应特性数据挖掘方法 [228](#bookmark168)

11.1.1 技术领域 [229](#bookmark169)

11.1.2 背景技术 [229](#bookmark170)

11.1.3 方案内容 [229](#bookmark171)

11.2 基于电力数据分析的河北南网电力市场化风险对冲方法 [231](#bookmark172)

11.2.1 电网对发电侧市场化风险对冲分析 [232](#bookmark173)

11.2.2 电网对用电侧市场化风险对冲分析 [233](#bookmark174)

11.2.3 基于方差偏离规律的统计套利对冲方法 [236](#bookmark175)

本章小结 [237](#bookmark176)

**附录** **FreeBSD 操作系统安装** [238](#bookmark177)

**第** **1** **章** **大数据相关概念**

大数据就是互联网发展到现今阶段的一种表象或特征而已，我们无须神化它或对它 保持敬畏之心，在以云计算为代表的技术创新大幕的衬托下，这些原本很难收集和使用 的数据开始容易被利用起来了，通过各行各业的不断创新，大数据会逐步为人类创造更 多的价值。



**1.1** **什么是大数据?**

大数据并不是一个新的概念，大数据其实是随着计算机技术、通信技术、物联网技术的 发展而必须面对的一个普遍的问题。类似计算机发展史上的软件危机，信息技术每发展到 一定阶段就会遇到数量太大处理不了的问题，可以说大数据是一种信息技术发展的现象。 20世纪60年代，随着商业软件的发展，原来的文件已经不能满足商业数据存储的要求，于 是产生了关系型数据库。相对于当时的传统文件，关系型数据库是一种处理大数据的经典 解决方案。戏剧性的是，随着互联网技术的发展，传统数据库已不能解决对互联网文档数据 的存储，于是产生了基于文档应用的 NoSQL 技术。可见在信息技术发展的过程中，人类在 不停面对来自大数据的挑战。

麦肯锡咨询公司最早提出大数据时代的到来：“数据，已经渗透到当今每一个行业和业 务职能领域，成为重要的生产因素。人们对于海量数据的挖掘和运用，预示着新一波生产率 增长和消费者盈余浪潮的到来。 ”IBM 将大数据的特征归纳为4个“V”: 量(Volume), 多样 (Variety), 价值(Value), 速度(Velocity) 。 特点有四个层面：第一，数据体量巨大。大数据 的起始计量单位至少是 P(1000 个 T)、E(100 万个 T) 或 Z(10 亿个 T); 第二，数据类型繁 多。比如，网络日志、视频、图片、地理位置信息等；第三，价值密度低，商业价值高；第四，处 理速度快。

麦肯锡咨询公司最早给出了大数据的定义：大数据是超过传统数据库工具的获取、存 储、分析能力的数据集，并不是超过TB 的才叫大数据。维基百科对大数据的定义：大数 据是指无法在可承受的时间范围内用常规软件工具进行捕捉、管理和处理的数据集。作 为多年从事数据处理工作的 IT 从业者，我们给出的大数据定义是：大数据是超过传统数 据库工具、传统数据结构、传统程序设计语言、传统编程思想的获取、存储、分析能力的数 据集。本书对大数据定义的补充是基于我们在长期的大数据处理中遇到的一些具体 挑战。

(1)关于传统数据库的局限。我们都知道 Oracle 数据库是大家常用的企业应用数据

库。但是对于现在的一个小的物流公司来说，使用这个 Oracle 已经不能满足 一个小公司获 取、存储、分析能力的需求。对于一个二线城市中的一个小物流中心来说，每天保守地说如 果是1万件的吞吐量，那么在数据库中会至少产生1万条记录， 一年就会产生365万条记 录，3年就会是1000万多条数据。对于千万级数据的调优，需要一个工作5年以上的数据 库工程师才能胜任，据了解这样一个工程师的年薪应该是20万元以上，那么对于一个小的 物流公司负担这样的薪水就有些困难了，所以大部分小物流公司对这些物流数据的处理措 施是定期删除数据，但是这些数据其实是蕴含着非常宝贵的商业价值的。如果采用大数据 时代的数据库管理工具，比如 MongoDB 等，那么成本就会降低很多，当然会使用基于大数 据需求的数据库管理工具的人才也是非常难找，目前只有一些重点大学成立了大数据专业， 比如清华大学的大数据学院等。

(2)关于传统数据结构的局限。大数据时代的数据具有4V 的特性，数据类型繁多，数 据量大，要求处理速度快，基于内存操作，因而传统的数据结构必须经过优化创新才能适应 大数据时代的应用需求。例如Google 的 BigTable 技术，Apache 软件基金会的 HDFS 文 件 结构，Mahout 中的数据结构，HIVE 数据库，学术界的热点知识图谱技术，这些都是顺应大 数据时代而生的大数据时代的数据结构。这些数据结构相比与传统编程语言中的数组、结 构体、ArrayList 、HashMap 等数据结构具有更好地适应大数据4V 特点的特性。现在已经 广泛应用的大数据的数据结构都是根据大数据的具体应用场景进行优化和创新的，例如 Mahout 中 的 FastByIDMap 、FastIDSet 和 GenericltemPreferenceArray 等，相对于单机环境 的 Java Collections 框架，它们降低了对内存的占用(欧文，2014)。

(3)关于传统的程序设计语言的局限。大数据应用环境下，产生了许多新兴的编程语 言，如 R 语 言 、Python 、Scala 等，这些语言天生具有操作分布式计算环境和进行机器学习运 算的基因。以 R 语言为例，其完成一个逻辑回归分析并生成图像，大概只需要5行代码，而 Java 和 C# 却需要调用N 多类库，写上数十行程序。Python 和 Scala 可以轻松操作Hadoop 和 Spark 平台，而这些在C# 和 Java 环境中，需要 N 多的类库和配置，再加上数十行代码

(4)关于传统编程工具中的报表、datagrid 等控件的数据展示能力是有限制的。如需 要展现上万个节点之间的关系，使用水晶报表或者JFreeChart 是无法完成用户需求的，需 要 EChart 、Three.js 等大数据时代的报表工具。

(5)关于传统编程思想的局限。Jeffrey Scott 等提出在大数据环境下，传统的编程思想 和框架产生了瓶颈(Vitter,2008), 因为传统的编程思想和框架将寄存器、缓存、内存、磁盘 统一编址，是基于所有的存储器具有相同访问时间的假设(Vitter,2008) 。 因而用这样的思 想来处理大数据，会造成效率低下，并且在这种框架和编程思想下开发的应用程序不能适应

大数据时代的应用。例如对600万条数据做一次查询需要十几秒，如果采用大数据时代的 编程思想，可以将查询控制在毫秒级别，当然还有服务器配置高低的问题，这里我们讨论的 是在相同的运算环境下。

综上所述，我们概括了传统数据库、数据结构、程序设计语言、编程思想的局限，提出了 大数据的定义。



**1.2** **大数据有多大?**

大数据到底有多大?这个问题在大数据的定义里是可以界定的，大数据的大是相对于 传统的数据库、编程语言、数据结构、编程思想和框架的处理能力的大，但是相对于现在的数 据增长来说，以前的工具、方法思想都是传统的。当遭遇新的智能硬件革命、新技术浪潮的 时候，现在的技术工具方法等又会在未来成为传统的且阻碍潮流发展的。因而大数据的大 是一个相对的大，人类在未来世界里会不断地遭遇大数据的问题，从而会有新的解决方案提 出。因而大数据不是一个新问题，会是一个一直存在的问题。



**1.3** **大数据是一种思维方式**

在大数据环境下，程序员、决策者、领导层等，应该具有大数据的思维方式，为什么要这 么说?我们接触过教师、官员、企业主、金融从业人员、互联网从业者等很多人，发现大家总 有这么几个思维定式：(1)认为大数据就是将传统的经验知识应用到海量数据里去，当然这 是应用大数据的一种方式，但是大数据其实是可以展示很多现象的，这些现象中有人类在传 统数据环境下不能发现的客观规律。如果一味地坚持已有知识的应用，而错过从全局角度 去发现、理解大数据现象下新的客观规律，就如同一叶障目不见泰山。(2)习惯将传统的思 维方式带入到大数据应用环境中。举个例子， 一个编程能力很强的研究生在数据采集过程 中总是很苦恼，工作进度很慢，我很奇怪，就问他为什么这么苦恼?进度这么慢，这明显不是 他的风格。他告诉我说，在大数据采集过程中，总有一些网站出现莫名其妙的异常，而且这 些异常超出了正常的逻辑。他一直试图把这些异常解决掉，但总是解决不了，所以耽误了进 度。其实这个学生陷入一个传统程序设计思维在大数据应用场景下的误区。因为大数据采 集过程中的异常来源是非常广泛的，如网络异常、网站改版、各种 Web 服务器的访问策略异 常等，有的是人为的，有的是硬件的，还有的是软件造成的。传统程序设计思维因处理的应 用场景单一，异常是可控的，程序员可以将程序写得非常完美，但是在大数据环境下， 一天可 能有上亿条数据的吞吐量，并且应用场景复杂，因而个别数据的不完整是可以忽略不计的， 因为大数据观察的是大趋势、大方向，个例的差异是可以忽略的，如果一味地追求程序的完 美，要处理所有异常，相对于大数据价值密度低、整体价值高的特点，在时间成本和用人成本 上是得不偿失的。

由此可见，大数据环境下的数据思维方式是需要慢慢建立和适应的。那么大数据思维是 什么?大数据思维主要包括两个方面：(1)从什么角度看数据。(2)怎样使用数据。

第一方面，大数据时代的到来，使人类以史无前例的低成本去获取和利用数据，人类的 本能是利用已有的知识和经验去理解、分析和利用这些数据，但是所有新事物的出现都会带 来新科学规律的发现，例如天文望远镜的出现使人类可以更直观地观测宇宙，而抛弃了原来 的猜想方式，发现了木星的四个卫星等客观规律。大数据时代的到来，肯定会帮助人类发现 新的客观规律。例如，我们依赖大数据技术和可视化技术对社交网络进行研究，从而更精准

地对社区概念进行定义(Xu,Xu et al.2015)。可见大数据可以给人类一个更有高度、更全

局的观察视角对客观世界进行分析和发现。

第二方面，怎样使用大数据。首先要明确使用大数据的目的是解决问题、发现规律，那 么如何根据要解决的问题去使用大数据?我们就需要使用数据的方法，通常我们叫作量化 的方法。所谓量化的方法，就是在解决问题的过程要可衡量、可评估、有明确的定义。车品 觉等提出了PIMA 定义(车品觉，2014):要解决的问题是什么,或者说目的是什么? (P); 在 要达到的这个目的的过程中要有非常明确的定义(I); 在解决问题的过程中用的手段必须 是可以量化的(M); 解决问题的结果是可以评估的(A) 。 这实际上 一 个相当严谨的研究问 题的体系，是一种严谨的数据思维。这个 PIMA 框架可以提供 一 个非常易用的大数据 流程。

综上所述，大数据思维是一种从全局角度去明确问题、定义过程、量化过程、评估结果的 数据思维方式。其不但可以跳出已有知识的界限，从全局角度发现新的规律，而且可以将已 有知识、规律应用于大数据的解读过程中，是我们向科学领域进军的新的得力工具和武器。



**1.4** **大数据思维的应用案例**

1 .3节中我们提出了大数据思维的定义，这个定义局限于当前我们对大数据的认识， 学界和工业界的学者和技术专家们一定会有不同于我们定义的大数据思维的意见，但是本 节还是要把一些我们之所以产生这样定义的案例阐述一下。

随着计算和存储成本的降低，人类可将生产、工作、生活中产生的海量数据进行存储和 分析，这些海量数据中蕴含着丰富的关系数据。在已有的计算和可视化条件下，结合云计算 和社区发现技术从全局角度分析和挖掘数据中蕴含的丰富信息，可以为自然科学和商业应 用等领域提供一种新的观察、分析和挖掘数据的视角，为发现数据中蕴含的结构、功能、规律 等信息提供新的强有力的工具和方法。下面从正在研究的两个领域简要介绍下我们提出的 大数据思维定义的产生背景。

(1)药物研究方面。当前国际上对新药物研发都是基于天然产物的活性新化合物的发 现和已知结构化合物二次开发两种思路相结合来开展的。例如我们对 Drugbank 数据库中

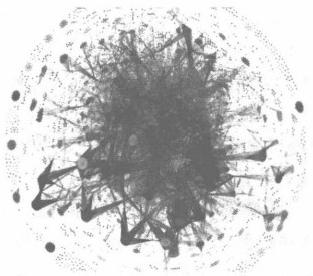


图1-1 DrugBank 数据库中的八千多种药物基于 靶标和作用酶进行聚类后的可视化图

的八千多种药物基于靶标和作用酶进行聚类， 图1-1中展示的是聚类后的可视化图，通过该 图我们可以了解到不同药物种类的分布，药物 聚类的内部结构和外部边界，从而指导新药的 研发。该图是当前4K 屏的可视化条件下的极 限，上部和下部小的类别没有全部显示出来。 聚类算法采用的是我们提出的骨干度算法 (Xu,Xu et al.2015),该算法采用从全局角度 去明确问题、定义过程、量化过程、评估结果的 思维方式，这部分工作已经发表在期刊《Ex- pert Systems with Applications》上，并得到国 际同行的认可。

(2)社交网络方面。近年来，随着时代的进步，社交网络的发展逐渐被人们关注，社交网 络已成为人们生活的一部分，并对人们的信息获得、思考和生活产生巨大的影响，社交网络在 人们的生活中扮演着重要的角色。社交网络成为人们获取信息、展现自我、营销推广的窗口。 社交网络中的用户之间的关系有关注关系、社区中的好友或亲情关系、实时交互过程中因共同 购买或评论产品而结成的共同兴趣关系等。这些关系所带来的信息量是巨大的，如果把这些 信息收集起来并加以分析，就能把虚拟关系转化成利润，为企业提供有价值的关系网络，从而 挖掘出潜藏在社交网络背后的巨大的经济价值，具体体现在：①帮助企业找到潜在的商机，比 如分析某个用户的评论和发表内容，可知他的消费能力、喜好和最近的购买习惯，从而知道他 购买自己产品的概率。②危机预警，根据用户的消息内容可以知道他对自己产品的满意度。 ③带动了消息的传播速度和广度，企业可以利用这一点，为自己的产品更好地做宣传。

图1-2～图1-5 是某高校学术协作网络在1996 — 2015年的演化过程，这是一个真实的 社交网络结构在时间轴上的演化过程的可视化。在该过程中，我们可以从全局视角观察到 社区的产生、发展、融合等演化过程，另外也可以观察到网络中任意节点在这个过程中扮演 的角色，如结构洞 Spanner 、意见领袖等。不同时间跨度上的社交网络聚类算法采用的也是 骨干度算法(Xu,Xu et al.2015)。

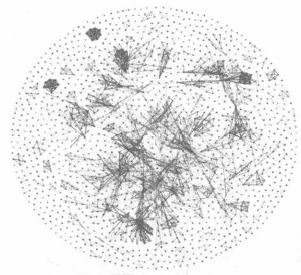


图1-2 1996—2000年某高校学术 协作网络的社交网络结构

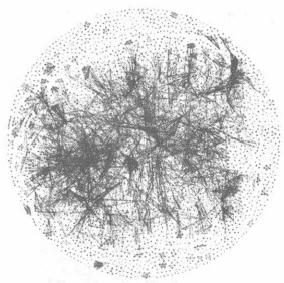


图1-3 2001—2005年某高校学术

协作网络的社交网络结构

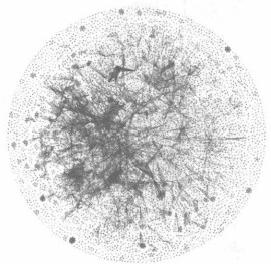


图1-4 2006—2010年某高校学术 协作网络的社交网络结构

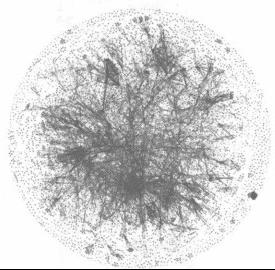


图1-5 2010—2015年某高校学术

协作网络的社交网络结构



**1.5** **大数据是如何产生的?**

大数据的产生是互联网、智能终端、网络通信、物联网等技术发展的必然结果，赵刚等认 为新兴技术的发展带来了数据产生的四大变化：(1)数据产生由企业内部向外部扩展；(2)数 据产生由Web1.0 向 Web 2.0 扩展；(3)数据产生由互联网向移动互联网扩展；(4)数据产 生由计算机/互联网向物联网扩展。

如何获取大数据?大数据获取方式有爬虫收集、行业数据、政府数据、企业自有数据等。 例如，支付宝的用户数据、物流公司的用户数据、电商网站的销售数据等。获取的方式总结 一下就是自身积累，再加上通过各个渠道收集。



**1** **.6美国和中国的大数据产业生态系统**

美国的大数据产业已经形成比较完善的生态体系，各种商业公司、开源机构在这个生态体系 中扮演着重要而不可替代的角色，同时又相互激烈竞争，新旧交替迅速。Lukas Lbiewald 画了一 张美国的大数据生态系统图(Lbiewald,2015) 如图1-6所示，该图非常详细地描绘了美国各种商业 公司和开源机构在大数据生态系统中所扮演的角色和所处阶段。美国的大数据产业分成三个大 环节：数据源、数据处理和数据应用。数据源包括数据库、商业应用和第三方数据。数据处理包括 数据强化、数据转化、数据整合和 API 接口。数据应用包括数据洞察和数据模型。

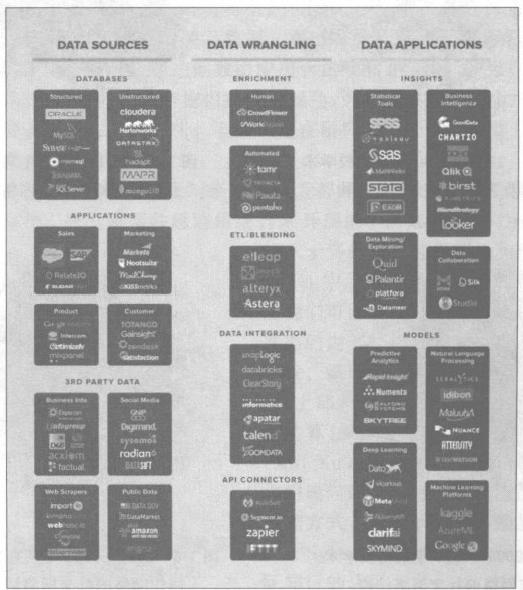


图1-6 美国大数据生态系统全图(Lbiewald,2015)

数据源中，数据库包括结构化数据库和非结构化数据库。传统的结构化数据库，有

Oracle、MySQL、SYbase、memsql、TERADATA 和 SQL Server 等。非结构化数据库，有

cloudera 、Hortonworks 、DATASTAX 、hadapt 、MAPR 和 MongoDB 。 商业应用数据源包括 销售、营销、产品和消费者4种。销售包括 SAP、RelateIQ 等；营销包括 Marketo、Hoot- suite 等；产品包括 Google Analytics 、Intercom 、mixpanel;顾客方向包括 TOTANGO 、 Gainsight 、zendesk 。第三方数据源包括商业信息、社交媒体、网络信息抓取和公共信息。商 业信息包括 infogroup 、D&.G 、Experian(益博睿)、Acxiom (安客诚)、factual 等；社交媒体包 括 GNIP 、Digimind 、Sysomos 、DataSift; 网络信息抓取包括 Import.io 、KimonoLabs 、web- hose.io 、Dataprovider; 公开数据包括 DATAGOV 、DataMarket 、amazon。

数据处理包括数据浓缩、数据转化、数据整合、API 接口。数据浓缩包括人工的 Crowd- flower 、Workfusion; 自动化的 Tamr 、Paxata 、Trifacta 、pentaho 。数据转化包括 etleap 、clo- verETL alteryx Astera。数据整合包括 snapLogic 、databricks 、ClearStory 、informatica 、apa- tar 、talend 等 。API 接口包括 MuleSoft 、Segment.io 、zapier。

数据应用包括数据洞察和模型。数据洞察又包括统计工具、商业智能、数据挖掘和探索和 数据协作。模型包括预测分析、自然语言处理、深度学习和机器学习平台。统计工具包括 SPSS、SAS、STATA 、Excel、MathWorks等。商业智能包括 GoodData、CHARTIO DOMO Qlik birst RJMETRICS MicroStrategy Looker 等。数据挖掘和探索包括 Quid Palantir platfora Datameer。数据协作包括 Silk RStudio 等。预测分析包括 Rapid Insight 、Numenta 、SALFORD SYSTEMS 、SKYTREE 。自然语言处理包括 LEXALYTICS,idibon 、Maluuba 、NUANCE 、AT- TENSITY 、IBMWATSON 等。深度学习包括 Dato、vicaricus、MetaMind、AlchemyAPI 、Clarifai、 SKYMIND 等。机器学习平台包括 kaggle、AzureML、Google等。

美国大数据生态系统的大部分企业是创业公司，当然也不乏原来老牌 IT 企业的子公 司。总体来看美国的大数据行业生态体系已经日趋完善，并且具有蓬勃的创新能力和盈利 能力。

中国 CDO 精英俱乐部在2015年11月汇集上百余大数据专家针对中国近大数据全产 业链近400家公司联合推出《2015年中国大数据公司年度排行榜》,但是该联盟并未将传统 互联网企业 TABLES 列入其中。TABLES 是改变中国互联网未来的六大力量，其中T 代 表腾讯，A 代表阿里巴巴，B 代表百度，L 代表雷军系，E 代表周鸿袆系，S 指以新浪、搜狐为 代表的大型门户网站。该榜单中我们可以发现中国的大数据产业已经形成了比较完善的产 业链条，其中不乏获得B 轮 和C 轮融资的企业，但是真正有实力的还是以 TABLES 为代表 的传统老牌互联网企业。



**1.7** **如何学习大数据技术**

对于大数据技术的初学者，首先要明确学习大数据技术的目的，如就业、学术研究等。 大数据就业机会大致分为三类：大数据平台搭建和维护、大数据软件开发、大数据分析工

作。商业应用主要是对数据进行分析，为商业决策提供支持。学术研究侧重对大数据的软

硬件平台相关算法、性能的评测、分析和创新等。

对于要在大数据浪潮中寻找工作机会的初学者来说，迫切需要一个很好练手的云计 算平台，而这样的平台需要自己搭建的门槛有些高，但的确是一个比较好的练手机会。 Hadoop 和 HBase 是大数据平台中相对容易搭建的平台，建议初学者在自己的笔记本计 算机上搭建。对笔记本的基本要求是固态硬盘和至少8GB 的内存，推荐配置是16GB 内存+256GB 固态硬盘。首先安装 VMWare station,然后创建3台主机，每台2GB 内 存，20GB 硬盘，主机的操作系统可以选择 FreeBSD 或 者 Ubuntu Linux,然 后 在 这 3 个 节点上配置 Hadoop 和 HBase 。 笔记本上配置的大数据的基础环境的性能和集群的性 能差异很大，但是这样的系统确实是学习大数据编程框架和进行小规模实验的方便的 平台。

对于学术研究的大数据初学者来说，同样需要一个上文中提出的小平台，但是如果数据 量巨大，可以考虑在已有的大规模集群上部署私有大数据环境，或者购买多节点的云主机来 实现。对于非计算机专业的大数据初学者，如果数据量不是十分巨大，可以考虑学习 R 语 言，进行大数据分析。

其实还有很多其他目的的大数据学习者，这里就不一一列举，无论什么目的，易用 的大数据平台是进行大数据分析的基础，除了刚才列出的 Hadoop+HBase, 还有 Spark 等新兴的大数据工具。我们在以后相关的章节里还介绍了 Spark 的 安 装 配 置 和 FreeBSD 的安装。



**本** **章** **小** **结**

大数据技术的应用是大数据技术的核心， 一切技术都是在应用中来最终回到应用中 去。触角比较敏感的 IT 从业者们在多年前就已经敏锐地感受到了大数据时代的到来。对 于 IT 工程师来说，每次由于新技术的发展、新智能设备的普及、带来的海量数据对他们都 是新的挑战、机遇或者说灾难。从固态硬盘、内存数据库、分布式运算和存储、接踵而来的大 数据解决方案，需要对新鲜事物有足够的耐心、热情，才能在大数据到来的时候顽强地适应 并引领时代的发展。

**参** **考** **文** **献**

[1] Lbiewald,L.(2015).“THE DATA SCIENCE ECOSYSTEM.”from https://

lukasbiewald.com/2015/06/11/the-data-science-ecosystem/.

[2] Vitter,J.S.(2008).Algorithms and Data Structures for External Memory,Now

Publishers Inc.

[3] Xu,Y.,et al.(2015).“A novel disjoint community detection algorithm for social networks based on backbone degree and expansion.”Expert Systems with Applica- tions 42(21):8349-8360.

[4] 车品觉.决战大数据.杭州：浙江人民出版社，2014.

[5] 欧 文 .Mahout 实战.北京：人民邮电出版社，2014.

**第** **2** **章** **搭建私有大数据处理平台**

工欲善其事，必先利其器。大数据技术的应用，在初期需要一个随时都能使用的计算和 存储环境，这样的环境是进行练兵使用的，需要自己搭建，以节约成本。本章提供一个最简 单的私有大数据环境的搭建方案，该解决方案采用FreeBSD+Hadoop+HBase 作为基础软 件平台，硬件为三台 HP 服务器。



**2.1** **FreeBSD操作系统安装**

FreeBSD 是一个先进的操作系统，其速度快、稳定性高，它是加州大学伯克利分校 BSD 版本的 UNIX 的分支，现在有非常大的社区维护它，FreeBSD 特点如下。

(1)磁盘I/O 速度

FreeBSD 的 UFS2 写入速度并不如 EXT4 快，甚至慢很多，但是读取速度明显快于 EXT4 和 EXT3 。 每读取1GB 的数据，基本要比 EXT4 快0 . 8～1 . 5秒。reroot 初始化执行 被加到 reboot 实用程序中，允许根文件系统挂载到临时源文件系统，而不需要一个完整的 系统重新启动。UEFI 引导器也带来了更好的多设备支持、帧缓冲器驱动的通用图形适配 器(UGA) 和图形输出协议(GOP) 处理改进，以及多个 ZFS 启动环境的支持。Hadoop 的 HDFS 是构建在基本文件系统上的，因此基本文件系统的读写速度决定了 HDFS 的读取速度， 而 Hadoop 处理的都是 TB 级数据，所以1GB 感觉不明显，而1TB 读取就大约快了800～1500 秒，相当于快了15～30分钟。HDFS 的设计初衷就是一次写入多次读取，所以读取速度越快 越好。mkimg 在 GPT 和 MBR 分区方案中支持 NTFS 文件系统。

(2)网络 I/O 速度

FreeBSD 是 TCP/IP 协议的最早实现者，因此在网络数据传输速度和稳定性上， FreeBSD 还是要优于Linux 的。如 ifconfig-v 现在能报告SFP/SFP+ 光模块数据，NIC 驱 动可提供一些信息，比如 cxgbe 、ixgbe 、mlx5en 和 sfxge。

(3)高度精简系统强壮，稳定和高安全性

FreeBSD 的系统在默认安装情况下，是高度精简的，除了内核和常用命令，并没有多余 的东西，非常干净和稳定。其稳定性也是久经考验，10年不关机的故事不是神话，据说目前 Hotmail 还有一部分 FreeBSD 的服务器。网络安全方面得天独厚，被发现的远程漏洞明显 少于 Linux 。 另外，基于以上几方面考虑，可认为，FreeBSD 更适合大数据的挖掘工作。

由于 FreeBSD 的安装过程需要太多的截图，所以我们将其放到“附录”中，如果参考可 以到本书的“附录”中查找。



**2.2** **基础软件安装**

**2.2.1** **安装** **Java** **运行环境**

进入/usr/ports/java/openjdk7 目录，执行 make install clean,等待安装完成。安装完

毕后，输入“java” 命令，如果找到java, 说明成功；如果没有找到，进入/usr/ports/java/open-

jdk6, 执行 make install clean,等待安装完成，安装完毕后，输入“java” 命令，如果找到 java,

说明成功。

**2.2.2** **安装** **bash**

进入/usr/ports /shells/bash 目录，执行 make install clean,等待安装完成，安装完毕 后，输入“bash” 命令，如果找到 bash, 说明成功。



**2.3 Hadoop 安装配置**

**2.3.1** **系统规划**

本节以 Hadoop 的最小配置来演示 Hadoop 平台的搭建，只需使用三台机器来搭建环 境。 一台机器作为 NameNode 和 JobTracker, 另外两台机器作为运行任务的 Datanode 节 点，如表2-1所示

**表2-1** **分布式机器节点说明**

|  |  |  |  |
| --- | --- | --- | --- |
| Node | User | IP address | 备 注 |
| Namenode | Hadoop5.tsinghua.edu.cn |  | Namenode和Jobtracker使用同一台机器 |
| Datanode | Hadoop6.tsinghua.edu.cn |  |
| Datanode | Hadoop8.tsinghua.edu.cn |  |

(1)修改 hosts

s vi/etc/hosts //将所使用到的三台主机及其名称添加到该文件中

(2)配置 ssh

//在 Namenode 节点上运行命令，使得 Namenode 能够无密码访问 Datanode

S scp root@hadoop6.tsinghua.edu.cn:/root/.ssh/id\_dsa.pub 2\_dsa.pub

s cat 2\_dsa.pub>>/root /authorized\_keys

$scp administrator@hadoop8.tsinghua.edu.cn:/root/.ssh/id\_dsa.pub 3\_dsa.pub

s cat 3\_dsa.pub>>/root /authorized\_keys

//在 Datanode节点上运行命令，使得 Datanode能够无密码访问Namenode

S scp administrator@hadoop5.tsinghua.edu.cn:/root /.ssh/id\_dsa.pub 1\_dsa.pub

$cat 1\_dsa.pub>>/root/authorized\_keys

//在 Datanode节点上运行命令，使得 Datanode 之间能够无密码访问

//在 Datanodel(hadoop6.tsinghua.edu.cn 上)

S scp administrator@hadoop6.tsinghua.edu.cn:/home/administrator/.ssh/id\_dsa.pub 3\_dsa.pub

$cat 3\_dsa.pub >>/root /authorized\_keys

//在 Datanode2(Hadoop8.tsinghua.edu.cn) 上

s scp administrator@hadoop6.tsinghua.edu.cn:/root/.ssh/id\_dsa.pub 2\_dsa.pub

$cat 2\_dsa.pub>>/root /authorized\_keys

**2.3.2** **配置** **conf/masters、conf/slaves 文件**

在所有节点上都要配置：

( 1 ) 在 $HADOOP\_HOME/conf/masters

中加入 NameNode IP 、Jobtracker IP,本 节

中添加 一个IP 地址即可。

( 2 ) 在 $ HADOOP\_HOME/conf/slaves

地址即可。

**2.3.3** **Hadoop 安装**

中加入 slavelPs, 本节中添加两个节点的 IP

执行以下步骤：

#cd /usr/ports/shells/bash

#make install

#cd/usr/ports/devel/apache-ant

#make install

# # # 添 加 hadoop 用户

#pw user add hadoop

#chsh hadoop

# # # 将 /bin/sh 改为/usr/local/bin/bash, 保存退出

#chown -R hadoop:hadoop /path/to/your/hadoop

#su hadoop

#export JAVA\_HOME=/usr/local/openjdk7

#export HADOOP\_HOME=/path/to/your/hadoop

然后执行。

注意：现在是 Hadoop 用户，以下事情都要用 Hadoop 用户完成。

#cd S HADOOP\_HOME

# # # 运 行 ant

#ant

编 辑 S HADOOP\_HOME/conf/core-site.xml

#vi $HADOOP\_HOME/conf/core-site.xml

core-site.xml

删除所有内容，加入以下：

<?xml version="1.0"?>

<?xml-stylesheet type="text/xsl"href="configuration.xsl"?>

<configuration>

<property>

<name>fs.default.name</name>

<value><hdfs://hadoop5.tsinghua.edu.cn:9000></value>

</property>

<property>

<name>io.compression.codecs</name>

<value>org.apache.hadoop.io.compress.DefaultCodec,com.hadoop.compression.lzo.LzoCodec,

com.hadoop.compression.lzo.LzopCodec,org.apache.hadoop.io.compress.GzipCodec,org.apache.hadoop.

io.compress.BZip2Codec</value>

</property>

<property> <name>io.compression.codec.1zo.class</name>

<value>com.hadoop.compression.lzo.LzoCodec</value>

</property>

</confiquration

hdfs-site.xml

<?xml version="1.0"?>

<?xml-stylesheet type="text/xsl"href="configuration.xsl"?>

<!--Put site-specific property overrides in this file.-->

<configuration>

<property>

<name>dfs.name.dir</name>

<value>/opt/data/hadoop/hdfs/name,/opt/data/hadoopl/hdfs/name,/opt/data/hadoop2/hdfs/ name</value>

<! - - 定义 hdfs namenode 所使用的硬盘路径名称-->

<description> </description>

</property>

<property>

<name>dfs.data.dir</name>

<value>/opt/data/hadoop/hdfs/data

/opt/data/hadoopl/hdfs/data,/opt/data/hadoop2/hdfs/data</value>

<! - - 定义 hdfs namenode 数据存储的路径-->

<description></description>

</property>

<property>

<name>dfs.http.address</name>

<value>hadoop5.tsinghua.edu.cn:50070</value>

<! - - 定义 hdfs http管理端口-- >

</property>

<property>

<name>dfs.secondary.http.address</name>

<value>hadoopslave-189.tj:50090</value>

<!-- 定义备用节点的管理地址-- >

</property>

<property>

<name>dfs.replication</name>

<value>3</value>

<!--定义数据的复制份数，份数越多越安全，但速度越慢-->

</property>

<property>

<name>dfs.datanode.du.reserved</name>

<value>1073741824</value>

<! - - 定义 du 操作返回 - - >

</property>

<property>

<name>dfs.block.size</name>

<value>134217728</value>

<! - - 定义 hdfs 的存储块大小，默认64M,我用的128M-->

</property>

<property>

<name>dfs.permissions</name>

<value>false</value>

<!--权限设置，最好不要-->

</property>

</configuration>

还有 一 个很重要的配置，MapReduce mapred-site.xml

的配置 ：

|  |  |
| --- | --- |
| <?xml version="1.0"?> |  |
| <?xml-stylesheet type="text/xsl"href | ="configuration.xsl"?> |

<!--Put site-specific property overrides in this file.-- 一

<configuration>

<property>

<name>mapred.job.tracker</name>

<value>hadoop5.tsinghua.edu.cn:9001</value>

<! - - 定义 jobtracker 的端口和主机-- >

</property>

<property>

<name>mapred.local.dir</name>

<value>/opt/data/hadoop1/mapred/mrlocal</value>

<! - - 定义 mapreduce 所使用的本机分发汇总数据存储路径-->

<final>true</final>

</property>

<property>

<name>mapred.system.dir</name>

<value>/opt/data/hadoopl/mapred/mrsystem</value>

<! - - 定义 map reduce系统的路径 - - >

<final>true</final>

</property>

<property>

<name>mapred.tasktracker.map.tasks.maximum</name>

<value>6</value>

<!--这个很重要，定义任务 map 的最大槽位数，太小了会慢，太多了会内存溢出--> <final>true</final>

</property>

<property>

<name>mapred.tasktracker.reduce.tasks.maximum</name>

<value>2</value>

<!--也很重要，定义任务的 reduce最大槽位数，设不好结果同上-->

<final>true</final>

</property>

<property>

<name>mapred.child.java.opts</name>

<value>-Xmx1536M</value>

<!--mapred 每个槽位所使用的内存大小，设定错误结果同上，通常是map 最大数\*虚拟机内存 数，也就是6x1536,超过 Datanode 内存就会溢出-- >

</property>

<property>

<name>mapred.compress.map.output</name>

<value>true</value>

<! - - 定义 map 采用压缩输出-- >

</property>

<property>

<name>mapred.map.output.compression.codec</name>

<value>com.hadoop.compression.1zo.LzoCodec</value>

<! - - 定义 map 压缩输出所使用的编解码器-->

</property>

<property>

<name>mapred.child.java.opts</name>

<value>-Djava.library.path=/opt/hadoopgpl/native/Linux-amd64-64</value>

<!--非常重要，这是能正常使用数据压缩的保证，hadoopgpl-->

</property>

<property>

<name>mapred.jobtracker.taskScheduler</name>

<property>

<name>mapred.jobtracker.taskScheduler</name>

<value>org.apache.hadoop.mapred.CapacityTaskScheduler</value>

</property>

<property>

<name>io.sort.mb</name>

<value>300</value>

</property>

<property>

<name>fs.inmemory.size.mb</name>

<value>300</value>

</property>

<property>

<name>mapred.jobtracker.restart.recover</name>

<value>true</value>

</property>

</configuration>



**2.4** **Hadoop** **开发环境配置**

**2.4.1** **编译** **Hadoop-eclipse-plugin-1.1.2.jar** **插件**

**1.** **编辑{HADOOP\_HOME}/build.xml**

(1)对 build.xml 文件31行的 Hadoop 版本做修改。

<property name="version"value="1.1.2-SNAPSHOT"/>

修改为

<property name="version"value="1.1.2"/>

( 2 ) 对 build.xml 文件2418行的ivy 下载进行注释，因为已经包含了ivy.jar。

<!--target name="ivy-download"description="To download ivy"unless="offline"> <get src="S{ivy\_repo\_url}"dest="s{ivy.jar}"usetimestamp="true"/>

</target-->

(3)对 build.xml 文件2426行去除对ivy-download 的依赖关系，保留如下：

<target name="ivy-init-antlib"depends="ivy-init-dirs,ivy-probe-antlib"

**2.** **编辑{HADOOP\_HOME}/src/contrib/build-contrib.xml**

<projectname="hadoopbuildcontrib"xmlns:ivy="antlib:org.apache.ivy.ant">

<propertyname="eclipse.home"location="eclipse的安装目录"/>

<propertyname="version"value="1.1.2"/> //build的 hadoop 的版本号

<propertyname="name"value="§{ant.project.name}"/>

<propertyname="root"value="${basedir}"/>

<propertyname="hadoop.root"location="${root}/../../../"/>

</project>

**3.building hadoop**

cd ${HADOOP-HOME}

ant compile

building eclipse-plugin for hadoop

**2.4.2** **eclipse 配置**

(1)打开 Window→Preferens, 会发现 HadoopMapReduce 选项，在这个选项里用户需

要配置 Hadoop installation directory,配置完成后退出。

(2)选择 Window→open perspective→Other…,选择有大象图标的 MapReduce, 此

时，就打开了 MapReduce 的开发环境。可以看到，右下角多了一个 MapReduce Locations

的框。

(3)设置 Hadoop 的环境参数。选择 MapReduceLocations 标签，点击该标签最右边的 大象图标，即那个齿轮状图标右侧的大象图标，打开参数设置页面。

·LocationName: 此处为参数设置名称，可以任意填写。

·MapReduceMaster: 此处为 Hadoop 集群的 MapReduce 地址，应该和 mapred-site. xml 中的 mapred.job.tracker 设置相同。

·Host:<10.0.0.211>。

·port:9001。

·DFSMaster: 此处为 Hadoop 的 master 服务器地址，应该和 core-site.xml 中的 fs. default.name 设置相同。

·Host:<10.0.0.211>。

· Port:9000。

设置完成后，点击 Finish 就应用了该设置。

此时，在最左边的 Project Explorer 中就能看到 DFS 的目录。

注意：解决Linux 上运行权限的问题，可以在服务器创建一个和 Hadoop 集群用户名一 致的用户，即可不用修改 Master 的 permissions 策略。

**2.4.3** **测试**

**1.** **新建项目**

File→New→Other→MapReduce Project,项目名可以任意取，如 HadoopTest。

2. 修改 WordCountTest.java

public static void main(String[]args)throws Exception(

//Add these statements.XXX

File jarFile=EJob.createTempJar("bin");

EJob.addClasspath("/opt/hadoop-1.1.1/conf");

ClassLoader classLoader =EJob.getClassLoader();

Thread.currentThread().setContextClassLoader(classLoader);

|  |  |
| --- | --- |
| String | inputPath="<hdfs://hadoop5.tsinghua.edu.cn:9000/user/chenym/input/file>\*"; |
| String | outputPath="<hdfs://hadoop5.tsinghua.edu.cn>;9000/user/chenym/output2"; |

Configuration conf =new Configuration();

String[]otherArgs=new String[]{inputPath,outputPath

};

if(otherArgs.length!=2){

System.err.println("Usage:wordcount System.exit(2);

<in><out>");

Job job=new Job(conf,"word count");

//And add this statement.XXX

((JobConf)job.getConfiguration()).setJar(jarFile.toString());

job.setJarByClass(WordCountTest.class);

job.setMapperClass(TokenizerMapper.class); job.setCombinerClass(IntSumReducer.class); job.setReducerClass(IntSumReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job,new

FileOutputFormat.setOutputPath(job,new

Path(otherArgs[0]));

Path(otherArgs[1]));

System.exit(job.waitForCompletion(true)?0:1);

输入上述程序后保存，然后右击执行：

run as mapreduece project



**2.5 Hadoop 升级**

( 1 ) 下 载 Hadoop1.1.2。

Fetch

<http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-1.1.2/hadoop-1.1.2.tar.gz>

( 2 ) 解 压 Hadoop:

Tar xzvf hadoop-1.1.2.tar.gz

Cd hadoop-1.1.2

Ant compile

(3)将旧版的 conf 下的所有 XML 文件复制到/usr/local/hadoop112/conf/

下，强制覆

盖原来的配置文件。

将/usr/local/hadoop112/文件夹，压缩然后传送到 hadoop6 和 hadoop8

Tar czf hadoop112.tar.gz /usr/local/hadoop112/

Scp hadoop112.tar.gz remote@hoadoop6.tsinghua.edu.cn:/home/remote

Scp hadoop112.tar.gz remote@hoadoop8.tsinghua.edu.cn:/home/remote

(4)重新启动 Hadoop:

/usr/local/hadoop112/bin/start-all.sh



**2.6 Zookeeper 安装**

Zookeeper 的官网文档上指出其在 FreeBSD 上只支持 Client, 但事实上我们在 FreeBSD 上

把 Client 和 Server 全部安装成功了。

**2.6.1** **在** **FreeBSD** **上安装** **Zookeeper**

(1)下载 Zookeeper。

wget <http://mirror.bit.edu.cn/apache//zookeeper/zookeeper-3.4.6/zookeeper-3.4>. 6.tar.gz (本次安装3.4.6 版 本)

其他版本下载地址(最好使用stable版本):<http://zookeeper.apache.org/releases.html> (2)解压。

tar-xzf zookeeper-3.4.6.tar.gz

(3)将 zookeeper-3.4.6/conf 目录下的 zoo\_sample.cfg 文件复制一份，命名为“zoo.cfg”。 (4)修改 zoo.cfg 配置文件。

修改 zoo.cfg 内容为

server.1=hadoop5.tsinghua.edu.cn:2888:3888

#server.2=hadoop5.tsinghua.edu.cn:2888:3888

server.2=hadoop6.tsinghua.edu.cn:2888:3888

server.3=hadoop8.tsinghua.edu.cn:2888:3888

#The number of milliseconds of each tick

tickTime=2000

#The number of ticks that the initial

#synchronization phase can take

initLimit=10

#The number of ticks that can pass between

#sending a request and getting an acknowledgement

syncLimit=5

#the directory where the snapshot is stored.

dataDir=/tmp/zookeeper

#the port at which the clients will connect

clientPort=2181

其中，2888端口号是 Zookeeper 服务之间通信的端口，而3888是 Zookeeper 与其他应用程 序通信的端口。而 Zookeeper 是在 Hosts 中已映射了本机的IP。

initLimit: 这个配置项是用来配置 Zookeeper 接收客户端(这里所说的客户端不是用户 连接 Zookeeper 服务器的客户端，而是 Zookeeper 服务器集群中连接到 Leader 的 Follower 服务器)初始化连接时最长能忍受多少个心跳时间间隔数。当已经超过10个心跳的时间 (也就是 tickTime) 长度后 Zookeeper 服务器还没有收到客户端的返回信息，那么表明这个 客户端连接失败。总的时间长度就是5×2000=10秒。

syncLimit; 这个配置项标识 Leader 与 Follower 之间发送消息，请求和应答时间长度，

最长不能超过多少个 tickTime 的时间长度，总的时间长度就是2×2000=4秒。

server.A=B:C:D:, 其 中A 是一个数字，表示这个是第几号服务器；B 是这个服务器的 IP 地 址 ；C 表示的是这个服务器与集群中的Leader 服务器交换信息的端口；D 表示的是万一 集群中的Leader 服务器挂了，需要一个端口来重新进行选举，选出一个新的 Leader, 而这个端 口就是用来执行选举时服务器相互通信的端口。如果是伪集群的配置方式，由于B 都是一样， 所以不同的 Zookeeper 实例通信端口号不能一样，所以要给它们分配不同的端口号。

(5 ) 创 建 dataDir 参数指定的目录(这里指的是“/tmp/zookeeper”), 并在目录下创建 文件，命名为“myid”。

(6)编辑“myid” 文件，并在对应的 IP 的机器上输入对应的编号。如在 Hadoop 5 上 ， “myid” 文件内容是1。Hadoop 6上 myid 的内容就是2。

至此，如果是多服务器配置，就需要将 Zookeeper-3.4.6 目录复制到其他服务器，然后 按照上述的方法修改 myid。

**2.6.2** **启动并测试** **Zookeeper**

(1)在所有服务器中执行

/usr/local/zookeeper/bin/zkServer.sh start。

(2)输入jps 命令查看进程

namenode 上显示为

hadoop5#jps

60677 JobTracker

33520 Jps

2461 DataNode

5222 HQuorumPeer

5413 HRegionServer

60495 NameNode

5279 HMaster

其中，QuorumPeerMain 是 Zookeeper 进程，启动正常。(HMaster 和 HRegionServer

为已启动的 HBase 进程，其他为安装 Hadoop (3)查看状态

/usr/local/zookeeper/bin/zkServer.sh

hadoop5#/usr/local/zookeeper/bin/zkServer.sh

后启动的进程。)

status。

status

JMX enabled by default

Using config:/usr/local/zookeeper/bin/../conf/zoo.cfg

Mode:follower

(4)启动客户端脚本

/usr/local/zookeeper/bin/zkCli.sh-server zookeeper:2181。

WatchedEvent state:SyncConnected type:None path:null

[zk:zookeeper:2181(CONNECTED)0]

[zk;zookeeper:2181(CONNECTED)0]help

Zookeeper -server host:port cmd args

connect host:port

get path [watch]

ls path [watch]

set path data [version]

rmr path

delquota [-n|-b]path

quit

printwatches on|off

create [-s][-e]path data acl

stat path [watch]

close

1s2 path [watch]

history

listquota path

setAcl path acl

getAcl path

sync path

redo cmdno

addauth scheme auth

delete path [version]

setquota -n|-b val path

[zk:zookeeper;2181(CONNECTED)1]1s /

[hbase,zookeeper]

[zk:zookeeper:2181(CONNECTED)2]

(5)停止 Zookeeper 进程

/usr/local/zookeeper/bin/zkServer.sh stop。



**2.7 HBase 安装配置**

HBase选用的最稳定版本0.94.9。主要的还是配置工作，依然将 HBase 放在/home

下，编辑/usr/local/hbase/conf 下 的 hbase-site.xml,hbase-default.xml,hbase-env.sh 这几

个文件。具体步骤如下。

(1)编辑所有机器上的 hbase-site 文件，命令如下：

vi/usr/local/hbase/conf/hbase-site.xml

注意以下两点：

① 其中首先需要注意 hdfs://hadoop 5.tsinghua.edu.cn:9000/hbase 这里，必须与用

户 的 Hadoop 集 群 的core-site.xml 文件配置保持完全 一 致才行，如果 Hadoop 的 HDFS 使 用了其他端口，请在这里也修改。再者就是 HBase 该 项 并 不 识 别 机 器 IP, 只 能 使 用 机 器

hostname, 即 若 使 用 hadoop5.tsinghua.edu.cn 的 IP 会 抛 出Java 错 误 。

② hbase.zookeeper.quorum 的个数必须是奇数。

|  |  |
| --- | --- |
| <?xml version="1.0"?> |  |
| <?xml-stylesheet type="text/xsl"href | ="configuration.xsl"?> |

<!--

/\*\*

\*Copyright 2010 The Apache Software Foundation

\*Licensed to the Apache Software Foundation(ASF)under one

\*or more contributor license agreements. See the NOTICE file

\* distributed with this work for additional information

\*regarding copyright ownership. The ASF licenses this file \*to you under the Apache License,Version 2.0(the

\*"License");you may not use this file except in compliance x with the License. You may obtain a copy of the License at

% <http://www.apache.org/licenses/LICENSE-2.0>

\*Unless required by applicable law or agreed to in writing,software

\*distributed under the License is distributed on an "AS IS"BASIS,

\*WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,either express or implied. \*See the License for the specific language governing permissions and

\*limitations under the License.

\*/

-->

<configuration>

<property>

<name>hbase.rootdir</name>

<value><hdfs://hadoop5.tsinghua.edu.cn:9000/hbase></value>

<description>The directory shared by region servers.

</description>

</property>

<property>

<name>hbase.cluster.distributed</name>

<value>true</value>

</property>

<property>

<name>hbase.master</name>

<value><hdfs://hadoop5.tsinghua.edu.cn>;60000</value>

</property>

<property>

<name>hbase.zookeeper.quorum</name>

<value>hadoop5.tsinghua.edu.cn,hadoop6.tsinghua.edu.cn,hadoop8.tsinghua.edu.cn</ value

</property>

</configuration>

(2)编辑所有机器的 hbase-env.sh, 命令如下。

vi/usr/local/hbase/conf/hbase-env.sh

修改代码如下所示：

# \*

# \* <http://www.apache.org/licenses/LICENSE-2.0>

# \*

#\*Unless required by applicable law or agreed to in writing,software

#\*distributed under the License is distributed on an "AS IS"BASIS,

#\*WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,either express or implied.

#\*See the License for the specific language governing permissions and

# \*limitations under the License.

# \*/

#Set environment variables here.

#This script sets variables multiple times over the course of starting an hbase process,

#so try to keep thingsidempotent unless you want to take an even deeper look

#into the startup scripts(bin/hbase,etc.)

#The java implementation to use. Java 1.6 required.

#export JAVA\_HOME=/usr/java/jdk1.6.0/

export JAVA\_HOME=/usr/local/openjdk7/

#Extra Java CLASSPATH elements. Optional.

export HBASE\_CLASSPATH=/usr/local/share/hadoop/conf

#The maximum amount of heap to use,in MB.Default is 1000.

#export HBASE\_HEAPSIZE=1000

#Extra Java runtime options.

#Below are what we set by default. May only work with SUN JVM.

#For more on why as well as other possible settings,

#see <http://wiki.apache.org/hadoop/PerformanceTuning>

export HBASE\_0PTS="-XX:+UseConcMarkSweepGC"

#Uncomment one of the below three options to enable java garbage collection logging for the

server-side processes.

#This enables basic gc logging to the.out file.

#export SERVER\_GC\_0PTS="-verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps"

#This enables basic gc logging to its own file.

#If FILE-PATH is not replaced,the log file(.gc)would still be generated in the HBASE\_LOG\_DIR.

#export SERVER\_GC\_0PTS="-verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps - X1oggc:<FILE-PATH>"

#This enables basic GC logging to its own file with automatic log rolling.Only applies to jdk 1.6.0\_34+and 1.7.0\_2+.

#If FILE-PATH is not replaced,the log file(.gc)would still be generated in the HBASE\_LOG\_DIR.

#export SERVER\_GC\_0PTS="-verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps - Xloggc:<FILE-PATH>-XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=1-XX:GCLogFileSize =512M"

#Uncomment one of the below three options to enable java garbage collection logging for the client processes.

#This enables basic gc logging to the .out file.

#export CLIENT\_GC\_OPTS="-verbose:gc-XX:+PrintGCDetails -XX:+PrintGCDateStamps"

#This enables basic gc logging to its own file.

#If FILE-PATH is not replaced,the log file(.gc)would still be generated in the HBASE\_LOG\_

DIR.

#export CLIENT\_GC\_0PTS="-verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps - Xloggc:<FILE-PATH>

#This enables basic GC logging to its own file with automatic log rolling.Only applies to jdk

1.6.0\_34+and 1.7.0\_2+.

(3)编辑所有机器的 HBase 的 HMasters 和 HRegionServers 。 修改/usr/local/hbase/

conf 文件夹下的 regionservers 文件。添加 DataNode 的 IP 地址即可。代码如下：

hadoop5.tsinghua.edu.cn

hadoop6.tsinghua:edu.cn

hadoop8.tsinghua.edu.cn

至此，HBase 集群的配置已然完成。

(4)启动、测试 HBase 数据库。

在 HMaster 即 NameNode 上启动 HBase 数 据 库(Hadoop 集群必须已经启动)。启动

命令：

第2章 搭建私有大数据处理平台



/usr/local/hbase/bin/start-hbase.sh

然 后 输 入 如 下 命 令 进 入 HBase 的 命 令 行 管 理 界 面 ：

/usr/local/hbase/bin/hbase shell

在 HBase shell下 输 入list, 列 举 当 前 数 据 库 的 名 称 ， 如 图 2 - 1 所 示 。 如 果 HBase 没 配

置 成 功 会 抛 出Java 错 误 。

Footadm4:“#/home/hbase/bin/hbase

list

shell

hbase(main):001:0>list

BusinessEntity

BusinessService

hode1

rou(s)in 0.0570 seconds

图2-1 HBase shell 下列举当前数据库的名称

我 们 也 可 以 通 过Web 页 面 来 管 理 查 看 HBase 数 据 库 。 HMaster:<http://hadoop5.tsinghua.edu.cn:60010/master.jsp>

HBase 数据库截图，如图2 - 2所示。

+→C Dhstopssn

此网页为[英文”网页，是否需要前译?翻译否

Master:hadoop5.tsinghua.edu.cn:60000

Lasal laxt.Thread Dim. Lat Lmml.Dchur.

=

选 项 ·

**PBHSE**

**Attributes**

|  |  |  |
| --- | --- | --- |
| **Attribute Mamn** Desrlptlor | | |
| HBane Verricn 0.94.9,r1496211 HBase vernion and revlnion | | |
| an Coplled | Ion Jun 2420:57:30 UTC 2013, Wen sse vertion an canlled and bg shar | |
| Hadoop Veraion 1.0.4,r1393290 | | Hsoop verelen and renlrlan |
| Hadoop Compiled Tha Oet 420 :40:3 urc Men Hadoop veralon wne caplled and by sten | | |
| itsst Roat | hdfs:/MadoogB tstnehua e Location of HDaae hoae directory | |
| Directary  Zookeeper Quarun | hadoop6,tsinghun ethu cn:21B1,I | of all reelatered z servera.For mre, |
| HWaster Start TimefThu Jul 1811:34:95 CST 2013 ate stae of shen thls Hster woe atarted  .  Date stap of whm this Haster becane active  str Aetlv  Thu Jul 18  Tine | | |
| rersm 3.67 | | Averag oglcm ver,Malve |
| HBaae Cluater ID ce2as059-f661- | | identi rated for each iBae |
| Ceprocessore □ | | ded by the |

Tasks

Shon AlL Lenl tarsd Ias Zhat nan-tfC Tast Shn AlL RCH

Taka Sha

Mo tasks currently runing an this node.

Tables

Catalog

RT00TA=

Table

The t0oT-table

he.ETA tshle

Pescription

holds references to all .ETA rericn.

holdn referencee to all Uher Thle rericn

9 table(a)in set.[Detailal



**图2-2** **HBase 数据库** **Web 管理界面**



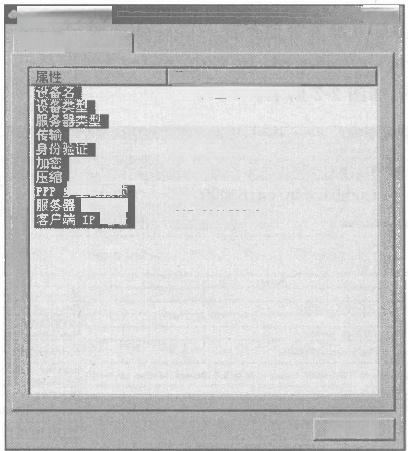
**2.8 FreeBSD上网配置**

FreeBSD 支 持VPN 上网和网页认证上网等多种上网方式，本节将介绍这两种常用的 认证上网方式。

**2.8.1** **VPN上网配置**

在国内很多高校和企事业单位中，使用的都是 VPN 上网，如何让配置 FreeBSD 支 持 VPN 上 网

(1)可以先找 一个 Windows 创建一个 VPN 连接，看这个 VPN 连接里面的属性，VPN 状态界面如图2-3所示。



**tsinghua** **扶态**

常 规 详 细 信 息

值

WAN 微 型 端 口(PPTP)

vpn

PPP

TCP/IP

MS CHAFP V2

MPPE 128

MPPC

关

<192.168.200.5>

<192.168.200.6>

关 闭C)

多重健接顿

IP 地址

地址

2×

图2-3 VPN 状态界面

(2)修改/usr/local/etc/mpd5/mpd.conf

#

为 pptp\_client。

#PPTP client:only outgoing calls,auto reconnect,

#ipcp-negotiated address,one-sided authentication,

#default route points on ISP's end

#

create bundle static B1

set iface route default

set ipcp ranges <0.0.0.0>/<00.0.0.0>/0

set bundle enable compression

set ccp yes mppc

set mppc yes e40

set mppc yes el28

set mppc yes stateless

create link static Ll pptp

set link action bundle B1

set link no pap eap

set link yes mschap

set auth authname "xxxx"

set auth password "xxxxxxxx"

set link max-redial 0

set link mtu 1460

set link keep-alive 2075

set pptp peer xxx.xxx.xxx.xx

set pptp disable windowing

open

(3)运行。

route delete default

mpd5 pptp\_client

**2.8.2** **网页认证上网配置**

首先安装图形界面，其次通过网页上网。通常安装 FreeBSD 选择推荐方式的最小安 装，安装完成后再通过编译源码或 pkg\_add 命令安装其他软件。 一般很少用到图形界面， 以留备用。安装过程中，创建默认一个普通用户并指定组为 wheel, 因为 ssh 远程登录禁用 root 用户，并且只有 wheel 组的用户可以 su 到 root 用户。

(1)通过 sysinstall 命令安装 gonme 和 Xorg。

复制代码，代码如下：

Configure->Packages->CD/DVD->gnome->gonme2-2.28.2\_ 1

Configure->Packages->CD/DVD->X11->Xorg-7.4\_3

安装完成后，不能直接运行 startx 命令，否则启动图形界面后系统无响应。 先编辑/etc/rc.conf 文件，复制代码如下：

dbus\_enable="YES"

hald\_enable="YES"

重新启动后，如果出现“acd0:FAILURE-unknown CMD…” 的错误信息，执行 hal- disable polling-device /dev/acd0 命令即可。再运行 startx 命令，至少我们可以使用Xorg 的简易的图形界面。如果使用gnome 界面，执行 echo“exec gnome-session”>.xinitrc;如 果使用kde 界面，执行 echo“exec startkde”>.xinitrc。

再运行 startx 即可。我们还可以进一步定制图形界面属性，利于配置刷新率、分辨

率等。

(2)使用 root 用户运行 Xorg-configure, 生成 xorg.conf.new 文件。

复制代码，代码如下：

大数据技术及行业应用

Section "Monitor"

Identifier "Monitor0"

VendorName "Monitor Vendor"

ModelName "Monitor Model"

HorizSync 30- [107](#bookmark178)

VertRefresh 48- [120](#bookmark179)

EndSection

Section "Screen"

Identifier "Screen0"

Device "Card0"

Monitor "MonitorO"

DefaultDepth 24

SubSection "Display"

Viewport 00

Depth 24

Modes"1024×768"

EndSubSection

EndSection

再运行 Xorg-config xorg.conf.new-retro 进行测试，如果出现灰格子和X 形鼠标，按Ctrl+ Alt+Backspace 组合键退出。测试完成后，运行 cp xorg.conf.new /etc/X11/xorg.conf 即可更新。 安装 VMware tools 时需要 compat6x, 可以下载 compat6x-i386-6.4.604000.200810.tbz。



**2.9** **配置杀毒软件**

在 FreeBSD 系统中，使用 ports 来安装 Clamav 非常简单方便，下面就具体介绍一下这

种方法。

(1)输入命令“cd/usr/ports/security/clamav”。

(2)输入命令“make”, 选上 clamav-milter。

(3)输入命令“make install”。此时在/usr/local/etc/rc.d 这个目录下我们就能找到

Clamav 的三个文件：clamav-milter 、clamav-clamd 、clamav-freshclam。

(4)输入命令“chmod +x clamd.conf”,并在/etc/rc.conf 加入如下三行命令

clamav\_clamd\_enable="YES"

clamav\_freshclam\_enable="YES"

clamav\_milter\_enable="YES"

如图2-4所示。

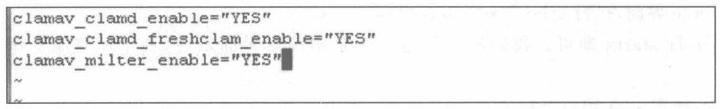


图2-4 修改 clamd.conf

安装成功后，我们再介绍 Clamav 里两个重要命令的使用方法：

(1)执行 freshclam 命令可以更新病毒库，具体位置在/usr/local/bin/freshclam, 输入：

freshclam。

(2)执行 clamscan 和 clamdscan 这两个命令可以查杀病毒，它们的使用方法类似，在这 里主要介绍一下 clamscan 命令的使用参数。

① 扫描指定的文件

clamscan 指定文件名

② 扫描当前目录

clamscan

③ 扫描/home 目录中所有文件(包括子目录)

clamscan-r /home

clamscan-r/ (扫描所有目录以及其子目录)

④ 扫描数据流

cat testfile |clamscan-

⑤ 检查邮件目录

clamscan-r-mbox /var/spool/mail

其他差数：

·-1[路径][文件名]把查杀的总结报告写入指定文件(不包含查杀了具体哪些文件)。 · --move [路径]移动病毒文件到指定目录中。

· --remove [路径]删除病毒文件。

· --unzip(unrar) 解压压缩文件扫描，



**本** **章** **小** **结**

本章介绍了搭建私有大数据处理平台的流程，相信大部分读者看完这一章都对大数据 平台的搭建望而生畏，但是不要放弃，如果想轻松地搭建和维护大数据平台，还有其他更轻 松有效的方案，将会在本书的第4章详细介绍。

**第** **3** **章** **大数据平台虚拟化解决方案**

在第2章学习了一个基于FreeBSD UNIX平台的小型私有云平台的搭建，这样的小型 云平台适合大数据的初学者和小型企业等初级用户使用。这一章我们将介绍适合企业应用 的一些大数据平台虚拟化解决方案。平台虚拟化是大数据平台的基础，因为对大规模集群 的部署和维护需要耗费大量的人力物力，采用虚拟化平台是工业界和学术界的共识。

本章重点介绍两种大数据平台的基础虚拟化平台：Docker 和 OpenStack 。这两种平台 和在第2章中搭建的私有大数据实验平台不同。Docker 和 OpenStack 都是为了解决大数 据平台的管理和维护的问题，解决物理机向虚拟机转化的问题。在第2章中使用的都是物 理机节点，而 Docker 和 OpenStack 可以将物理节点转化成虚拟节点。虚拟化有多个优点， 其解决物理节点维护烦琐的瓶颈，虚拟化具有备份、快照、双机热备等多种功能，这些在物理 节点实现需要大量的人力、物力。但是虚拟化也有其缺点，上述功能很多是以牺牲硬件性能 为代价的。由于硬件发展符合摩尔定律，硬件越来越便宜，当硬件的成本远低于人力成本的 投入时，Docker 和 OpenStack 平台应运而生。用户可以选择将大数据平台部署在 Docker 和 OpenStack 上，也可以选择直接部署到物理节点上。除了 Docker 和 OpenStack 还有 VMWare 、Citrix 、Hyper-v、红山 vGate 等虚拟化软件，读者可以通过其官方网站进行了解， 本书不作赘述。



**3.1 Ubuntu 上安装** **Docker**

**3.1.1** **Docker 简介**

Docker 是一个开源项目，其目标是提供开放工具帮助开发者打造开放 API, 使系统管 理员更好地管理其应用程序。Docker 可以创建非常轻量的“虚拟机”,即容器，其核心思想 是创建应用程序可移植的轻量容器。Docker 具有如下功能：隔离应用，创建、复制应用镜 像，创建容易分发的即启即用的应用，允许实例简单、快速扩展，测试应用并删除等。

Docker 的重要概念包括镜像、容器和数据卷(DockerOne 2015)。

(1)镜像：Docker 中的镜像类似虚拟机的快照，但是更轻量(DockerOne 2015)。 Docker 镜像既可以重新创建，又可以在原有的镜像上创建。Docker 镜像有唯一的 ID, 以方 便用户调用。

(2)容器：容器是独立运行的一个或者一组应用，以及他们的运行态环境(DockerOne 2015)。容器被设计用来运行单进程，无法很好地模拟一个完整的环境。Docker 提供了用

来分离应用与数据的工具，用户可以快捷地更新运行中的系统或代码，而不影响数据。

(3)数据卷：数据卷是一个可供一个或者多个容器使用的特殊目录(DockerOne 2015)。 数据卷可以使用户不受容器生命周期影响进行数据持久化，其表现为容器内的空间，但实际保 存在容器之外，该特性可以使用户在不影响数据的情况下销毁、重建、修改和丢弃容器。

**3.1.2** **Docker 安装**

(1)检测系统内核

使用uname-r 命令检测系统内核，内核需在3.10以上。

s uname -r

3.11.0-15-generic

(2)更新 apt 源

$sudo apt -key adv --keyserver hkp://p80.pool.sks -keyservers.net:80 --recv -keys

58118E89F3A912897C070ADBF76221572C52609D 打开/etc/apt/sources.list.d/docker.list

创建，并在其中添加如下内容：

deb <https://apt.dockerproject.org/repo> 更新 apt 包的 index:

$sudo apt-get update

(3)下载并安装 Docker

s sudo apt -get install docker-engine (4)开启 Docker 服务

$sudo service docker start

(5)验证是否安装成功

$docker info

文件查看是否有 docker.list 文件，若没有则

ubuntu-precise main

**3.1.3** **Docker** **镜像相关命令**

Docker 镜像相关命令如表3- 1所示。

**表3-1** **Docker** **镜像相关的命令列表**

|  |  |  |
| --- | --- | --- |
| 命令 | 使用方法 | 功能 |
| pull/push | docker pull/push <image> | 下载、上传镜像 |
| images | docker images | 显示Docker镜像 |
| import | docker import file|URL|-[REPOSITORY[:TAG]] | 导入Docker镜像 |
| load | docker load filel URLl-[REPOSITORY[:TAG]] | 导入Docker镜像 |
| save | docker save [OPTIONS]IMAGE [IMAGE...] | 导出Docker镜像 |
| rmi | docker rmi IMAGE | 移除指定镜像 |
| tag | docker tag [OPTIONS]IMAGE[:TAG]NAME[:TAG] | 重命名镜像 |

**1.Docke r 获取镜像**

docker pull ubuntu

Docker 默认是从 Docker Hub 上下载，Docker Hub 是 Docker 官网的镜像仓库，也是世 界上最大的镜像仓库，基本能想到的镜像都能在其中找到。

**2.Docker** **镜像的重命名**

可以使用tag 命令对镜像重命名，如“docker tag ubuntu myubuntu”,将 ubuntu 镜像重 命名为 myubuntu

**3.Docker 镜像的导入、导出**

(1)导出

docker save-o/opt/dockerimagesfile/ubuntu-14.04.tar ubuntu:14.04

将 ubuntu:14.04 镜像文件导出到/opt/dockerimagesfile 文件夹中，并命名为ubuntu- 14.04.tar

(2)导入

docker load-i/opt/dockerimagesfile/ubuntu-14.04.tar

**4.Docker** **镜像的删除**

docker rmi ubuntu:14.04

在删除 Docker 镜像前需将由此镜像的容器先删除。 push 命令是推送镜像的意思，可 将做好的镜像推送至 Docker Hub上，也可以推送至个人搭建的私人仓库中去。

**3.1.4** **Docker** **容器相关命令**

Docker 容器相关的命令如表3-2所示。

**表3-2** **Docker** **容器相关的命令列表**

|  |  |  |
| --- | --- | --- |
| 命令 | 使用方法 | 功能 |
| run | docker run [OPTIONS]IMAGE | 从镜像中运行一个容器 |
| start/stop/restart | docker start/stop/restart CONTAINER | 开始、暂停、重启容器 |
| attach | docker attach [OPTIONS]CONTAINER | 进入正在后台运行的容器 |
| build | docker build [OPTIONS]PATH | 从Dockerfile中构建新的Image |
| inspect | docker inspect CONTAINER|IMAGE [CONTAINER|IMAGE..] | 查看容器运行时详细信息，了解一个Image或 者Container的完整构建信息 |
| kill | docker kill [OPTIONS]CONTAINER [CONTAINER...] | 杀死容器的进程 |
| ps | docker ps [-a] | 查看正在运行的容器， -a查看全部的容器 |
| rm | docker rm [OPTIONS]CONTAINER | 移除容器 |
| top | docker top CONTAINER [ps OPTIONS] | 显示容器内部的进程 |
| commi | docker commit [OPTIONS]CONTAIN- ER [REPOSITORY[:TAG]] | Docker容器转换为Docker镜像 |

(1)运行容器

docker run -i -t ubuntu:14.04

在 ubuntu:14.04 镜像的基础上开一个具有标准的输入和输出流， 一个伪终端的容器，

如图3-1所示，其中957d104932f9 为此容器的 ID。

|  |  |  |
| --- | --- | --- |
| **rooteubuntu-jie:/opt/dockerimagesfilet** **docker** **run** **-i** **-t** **ubuntu:14.04** | | |
| **root0957d104932f9:/\*1s**  bin dev home lib64 boot etc lib media | mnt proc run opt root sbin | SrV Var  3y8 usT |

图3-1 运行一个容器

(2)暂停容器

docker stop 957d104932f9

(3)启动容器

docker start 957d104932f9

(4)进入容器

docker attach 957d104932f9

启动和进入容器如图3-2所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **rootEubuntu-jie:/opt/dockerimagesfilef** **docker** **start** **957d104932f9**  957d104932f9  **rootBubuntu-jie:/opt/dockerimagesfilef** **docker** **attach** **957d104932f**9 | | | | |
| **root8957d104932f9:/\***  **root2957d104932f9:/\*1s**  bin dev home lib64 mnt  bootetc lib media opt  **root0957d104932f9:/\*** | proc root | run STV sbin sys | uST | var |

图3-2 启动和进入容器

查看 Docker 内部进程如图3- 3所示。

docker top 957d104932f9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| root@ubuntu-jie:/opt/dockerimagesfilet docker top 957d104932f9   |  |  |  |  | | --- | --- | --- | --- | | UID | PID | PPID | C | | STIME | TTY | TIME | CMD | | root | 2297 | 753 | 1 | | 5:38 | pts/2 | 00:00:00 | /bin/bash |   rootEubuntu-jie:/opt/dockerimagesfilef |

图3-3 查看 Docker 内部进程

(5)移除容器

docker rm 957d104932f9

(6)容器转换为镜像

docker commit 957d104932f9 myubuntu

如图3-4所示。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| root@ubuntu-jie:/opt/dockerimagesfilet docker commit 957d104932f9 myubuntu  64ba00aff6d6a1a60305484db0f59b331a3f0d6ab216e7ac84ea77a16b06e982 rooteubuntu-jie:/opt/dockerimagesfilet docker images   |  |  |  |  | | --- | --- | --- | --- | | REPOSITORY | TAG | IMAGE ID | CREATED | | VIRTUAL SIZE |  |  |  | | myubuntu | latest | 64ba00aff6d6 | 9 seconds ago | | 188.4 MB |  |  |  | | jie/docker-desktop 1.127 GB | latest | 9722f6bd1d46 | 5 months ago | | ubuntu | 14.04 | d2aOecffe6fa | 6 months ago | | 188.4 MB |  |  |  | |

图3-4 容器转换为镜像展示

**3.1.5** **Dockerfile 创建镜像**

**1.Dockerfile** **的基本语法**

**【例3-** **1】** 以 Ubuntu:14.04 为基础创建一个带有 OpenJDK7 的镜像。

#test

#author jie

#install open-jdk-7

#Format:FROM repository[:version]

FORM ubuntu:14.04

#Format:MAINTAINER Name <email@addr.ess>

MAINTAINER Jie <jie147@yahoo.com>

#update apt-get and install openjdk 7

RUN apt-get update &&apt-get install openjdk-7-jre-headless

在例3-1中，相当于有了一个能够运行Java 代码的容器。在例3-2中，将 MongoDB 数 据库安装在 Docker 中 ，Docker 容器具有快速创建和启动等优点，Dockerfile 相关命令列表 如表3- 3所示。

**表3-3** **Dockerfile** **相关命令列表**

|  |  |  |
| --- | --- | --- |
| 命令 | 基本用法 | 功能 |
| FROM | FROM<Image> | 以image为基础 |
| MAINTAINER | MAINTAINER <author name> | Dockerfile作者 |
| RUN | RUN<common> | 容器中运行的命令 |
| ADD | ADD<src><dest> | 将本机的文件添加至容器中 |
| CMD | CMD [“executable""param1""param2"]or CMD ["pa- ram1”“param2”]or CMD command paraml param2 | CMD有三种格式 |
| EXPOSE | EXPOSE<port> | 暴露出容器端口 |
| ENTRYPOINT | ENTRYPOINT ['executable’'param1’'param2’]or ENTRYPOINT command param1 param2 | 与CMD命令类似 |
| WORKDIR | WORKDIR/path/to/workdir | 容器的工作路径 |
| ENV | ENV<key><value> | 环境变量 |
| UID | USER<UID> | 设置UID |
| VOLUME | VOLUME['/data’] | 添加数据卷 |

**2.Dockerfile 创建** **MongoDB**

**【例3-2】** 在 Ubuntu 中创建 一 个MongoDB 数据库的 Docker 镜 像。

#test

#author jie

#build mongodb

#come from <https://docs.docker.com/engine/examples/mongodb/>

#Format:FROM repository[:version]

FROM ubuntu;14.04

#Format:MAINTAINER

#Installation:

Name <email@addr.ess>

#Import MongoDB public GPG key AND create a MongoDB list file

RUN apt-key adv --keyserver <hkp://keyserver.ubuntu.com:80> --recv 7F0CEB10

RUN echo "deb <http://repo.mongodb.org/apt/ubuntu>"$(lsb\_release -sc)"/mongodb-org/3.0 multiverse"|tee /etc/apt/sources.list.d/mongodb-org-3.0.list

#Update apt-get sources AND install MongoDB

RUN apt-get update &&apt -get install -y mongodb-org

#Create the MongoDB data directory

RUN mkdir-p/data/db

#Expose port 27017 from the container to the host

EXPOSE 27017

#Set usr/bin/mongod as the dockerized

ENTRYPOINT ["/usr/bin/mongod"]

使用 Docker build 命令创建 MongoDB

entry-point application

镜像，执行命令docker

build./,如图3-5所示。

FootEubuntu:/home/workspace/docker/dockerfile/mongi

Sending build context to Docker daemon 2.56 KB

Sending build context to Docker daemon

Step 0:FROM ubuntu:14.04

--->d2aOecffe6fa

Step 1:MAINTAINER Jie <jie147Byahoo.com>

--->Running in 92fd57eaf6ed

-->42e5c79blae9

Removing intermediate container 92fd57eaf6ed

[Step 2:RUN apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB](Step2:RUNapt-keyadv--keyserverhkp://keyserver.ubuntu.com:80--recv7F0CEB)

10

--->Running in eba58828a2b9

Executing:gpg --ignore-time-conflict --no-optiona --no-default-keyring --homedi

E /tmp/tmp.mtnfzsOlvn --no-auto-check-trustdb --trust-model alwaye --keyring /et

图3- 5 使 用 Docker build 命令创建 MongoDB 镜 像

使用 Docker images 查看镜像，存在一个 mongodb:1.0, 如图3-6所示。

|  |  |  |  |
| --- | --- | --- | --- |
| root@ubuntu:/home/workspace/docker/dockerfile/mongodbf docker images  REPOSITORY TAG IMAGE ID CREATFD | | | |
| VIRTUAL SIZE  mongodb  368.7 MB | 1.0 | 1cdb36279a5b | 22 minutes ago |

图3-6 使用 Docker images 查看镜像

大数据技术及行业应用

通 过 MongoDB 镜像创建容器，使用如下命令，如图3-7所示。执行命令： docker run -p 27017:2017 --name mongodb\_test mongodb:1.0

|  |  |
| --- | --- |
| root@ubuntu:/home/workspace/docker/dockerfile/mongodbt docker run -p 27017:2017  name mongodb\_test mongodb:1.0  2016-01-09T08:38:59.739+0000 I CONTROL [initandlisten]MongoDB starting:pid=  port=27017 dbpath=/data/db 64-bit host=71dfba78eeba | |
| 2016-01-09T08:38:59.739+0000  2016-01-09T08:38:59.739+0000 e42856924d84e220fbe4a839e605d 2016-01-09T08:38:59.739+0000 187-89-1263.13.0-24-generic  BOOST\_LIB\_VERSION=1\_49  2016-01-09T08:38:59.739+0000 2016-01-09T08:38:59.739+0000 2016-01-09T08:38:59.782+0000  rnal | I CONTROL [initandlisten]db version v3.0.8  I CONTROL [initandlisten]git version:83d8cc25e0d  I CONTROL [initandliaten]build info:Linux ip-10-  \*46-UbuntuSMP Thu Apr 1019:11:08 UTC 2014 x86\_64  I CONTROL [initandliaten]allocator:tcmalloc  I CONTROL [initandlisten]options:{}  I JOURNAL [initandliaten]journal dir=/data/db/joy |
| 016 | |

图3-7 通过 MongoDB镜像创建容器

创建完成界面，如图3-8所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2016-01-09T08:39:16.047+000O 2016-01-09T08:39:16.049+0000 data/db/local.ns,filling  2016-01-09T08:39:16.146+0000 /data/db/local.0,filling  2016-01-09T08:39:16.146+0000  /db/\_tmp  2016-01-09T08:39:16.171+0000 /data/db/local.0,size:64MB, 2016-01-09T08:39:16.183+0000 on nort 27017 | I I with  I with  I  I  I | CONTROL [initandlisten]  INDEX [initandlisten]allocating new ns file /  zeroes.  STORAGE [FileAllocator]allocating new datafile  zeroea... | | | |
| STORAGE  STORAGE  took 0.018  NETWORK | [FileAllocator]creating  [FileAllocator]done secs | directory  allocating | /data  datafile |
| [initandlisten]waiting for connectiong | | |

图3-8 创建完成界面

**3.1.6** **Docker** **实现** **Spark** **集群**

**1.** **基础镜像的制作**

基础镜像是在 ubuntu14.04 的基础上安装了JDK7.0 、Scala 、Hadoop 2.6 、Spark 1.4

等软件。使用 Dockerfile 创建 dspark:basic 镜像，Dockerfile 内容如下：

FROM ubuntu;precise

#build dspark docker image

#Upgrade package index

#install a few other useful packages plus Open Jdk 7

#Remove unneeded /var/lib/apt/lists/\*after install to reduce the

#docker image size(by～30MB)

RUN apt-get update &&apt-get install -y less openjdk-7-jre-headless net-tools vim-ti-

sudo openssh-server &&rm-rf /var/lib/apt/lists/\*

ny

#set environment

ENV SCALA\_HOME/opt/scala

ENV HADOOP\_HOME /opt/hadoop

ENV SPARK\_HOME /opt/spark

ENV PATH §HADOOP\_HOME/bin:$SPARK\_HOME/bin:$SCALA\_HOME/bin:$PATH

#install scala

ADD/home/pkg/scala/scala-2.11.7.tgz /opt

ADD/home/pkg/hadoop/hadoop-2.6.0.tar.gz /opt

ADD /home/pkg/spark/spark-1.4.1-bin-hadoop2.6.tgz /opt

RUN cd/opt &&ln-s./spark-1.4.1-bin-hadoop2.6 spark &&ln-s./hadoop-2.6.0 hadoop &&

ln-s./scala-2.11.7 scala

#add some profiles:hadoop spark host

#

COPY./hadoop//opt/hadoop/etc/hadoop/

COPY./conf//opt/spark/conf/

#ssh login without password

RUN ssh-keygen -tdsa-P"-f～/.ssh/id\_dsa

RUN mkdir-p/opt/data/hadoop/dfs/name &&mkdir -p/opt/data/hadoop/dfs/data

#Hdfs ports

EXPOSE 50010500205007050075500908020

#Mapred ports

EXPOSE 198881002090019010

#Yarn ports

EXPOSE 8030803180328033804080428088

#SSH port

EXPOSE 228080

执行命令 docker build-t dspakr:basic./ 创建镜像，并将镜像命名为 dspark:basic ,如

图3- 9所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| root@ubuntu:-docker images  REPOSITORY TAG  VIRTUAL SIZE  dapark basic2.0  877 MB | | IMAGE ID  3f7a321929d2  58248a18746c | CREATED 11 hours  11 hours | age  ago |
| dspark  877 MB | basic |

图3-9 使用命令 Docker build 创建镜像

接下来将使用 dspark:basic 这个基础镜像制作出整个集群。

**2.** **集群的制作**

集群中包含一个 Master 节点，两个 Slave 节点。设置 MASTER\_IP= <172.17.1.1>,

修

改 slaves 文件，Dockerfile 如下：

FROM dspark;basic

#build dsparkmaster1:1.0

###

# Add slave

###

ENV MASTER\_IP <172.72.1.1>

ADD slaves $HADOOP\_HOME/etc/hadoop/

ADD slaves $SPARK\_HOME/conf/

使 用 docker build -t dsparkmaster:1.0./ 创建镜像，创建过程如图3- 10所示。

|  |
| --- |
| root@ubuntu:/home/workspace/docker/dockerfile/spark/mastert docker build -t dape rkmaster:1.0 ./  Sending build context to Docker daemon 3.072 KB  Sending build context to Docker daemon  Step 0:FROH dspark:basic  --->58248a18746c  Step 1:ENV MASTER\_IP <172.72.1.1>  --->Running in 9850d8bcf9eb  --->f05ff6abf57b  Removing intermediate container 9850d8bcf9eb  Step 2:ADD slaves SEADOOP\_HOME/etc/hadoop/  --->ad57e4bf6f63  Removing intermediate container e6d6180599b1  Step 3:ADD slaves SSPARK\_BONE/conf/  --->ab3146aa5451  Removing intermediate container 41786c94e591  Successfully built ab3146aa5451 |

图3-10 使用命令 Docker build 创建镜像

(1)启动 Master 容器

docker run -dit -p 22 -p 8080 -p 8088 -p 50070 -m 8g -net =none ---name master dsparkmaster /bin/bash

(2)启动 Slave 容器

docker run -dit --net=none --name slavel -m 6g dsparkmaster

docker run -dit --net=none --name slave2 -m 6g dsparkmaster

|  |
| --- |
| root&ubuntu:/home/workapace/docker/dockerfile/apark/mastert docker ps  CONTAINER ID IMAGE COHHAND CREATED STATUS  PORTS NAMES  c295269c880e daparknaster "/bin/bash" About a minute ago Up About a  minute slave2  a3995b76be9d daparknaster “/bin/bash” &hout a minute ago 0p About a  minute alave1  c6438116b0e7 dsparkmaster "/bin/bash" About a minute ago Op About a  minute master |

图3-11 启动 Slave 容器

Docker 并 无 固 定IP 地址的命令或工具，可以通过虚拟网桥对来实现 IP 的固定，方法

如下写成的 shell 文 件 ：

#!/bin/bash

#1 master ip is <172.17.1.1>

#2 slave ip is <172.17.2.1> -<172.17.2.2>

#

if[!-d"/var/run/netns"];then

mkdir/var/run/netns

fi

#for the master

master\_name =master

declare -i master\_ip

master\_ip=1

docker start ${master\_name}

pid=§(docker

inspect pid=$pid"

echo "master

ln-s/proc/spid/ns/net

-f"{{.State.Pid}}"${master\_name})

/var/run/netns/$pid

ip link add veth\_${master\_name)\_b type veth peer name veth\_${master\_name)\_c

brctl addif docker0 veth\_${master\_name}\_b

ip link set veth\_${master\_name}\_b up

ip link set veth\_ §{master\_name)\_c netns spid

ip netns exec spid ip link set dev veth\_S{master\_name}\_c name etho

ip netns exec S pid ip link set eth0 address 12:34:56:78:9a:${master\_ip}

ip netns exec $pid ip link set eth0 up

ip netns exec S pid ip addr add 172.17.1.${master\_ip}/16 dev etho

ip netns exec S pid ip route add default via <172.17.42.1>

#end master set IP

#for the slave network

declare -i slave\_num slave\_i

slave num=2

for((slave\_i=1;$slave\_i<=$slave\_num;slave\_i=$slave\_i+1))

do

slave\_name="slave $slave\_i"

declare -i slave\_ip

slave\_ip=$slave\_i

docker start ${slave\_name}

pid=§(docker inspect -f'{{.State.Pid}}'${slave\_name})

echo"s slave\_name pid=$pid"

ln-s/proc/Spid/ns/net /var/run/netns/$pid

ip link add veth\_${slave\_name}\_b type veth peer name veth\_${slave\_name}\_c brctl addif docker0 veth\_${slave\_name}\_b

ip link set veth\_${slave\_name}\_b up

ip link set veth\_${slave\_name)\_c netns $pid

ip netns exec s pid ip link set dev veth\_S{slave\_name)\_c name etho

ip netns exec s pid ip link set eth0 address 12:34:56:78:9a:${slave\_ip}

ip netns exec $pid ip link set eth0 up

ip netns exec S pid ip addr add 172.17.2.§{slave\_ip}/16 dev eth0

ip netns exec $pid ip route add default via <172.17.42.1>

done

#end slave set IP

将创建的三个容器： 一个 Master、 两 个 Slave 容器启动后运行 shell 文件，即可固定 IP,

如图3- 12所示。

Foot@ubuntu:/home/workspace/docker/dockerfile/spark/mastert ./start\_28lave.sh

masteT

master pid=6620

slave1 pid-6644

3lave2 pid=6667

图3-12 启动设置固定 IP 的脚本

目前 Docker 官网没有提供固定 IP 的方法，容器重启后，又要重新设置固定 IP 相当麻烦。

**3.** **集群启动测试**

(1)先格式化 NameNode 节点，执行如下命令：

hdfs namenode -format

(2)然后执行启动 Spark 集群的命令来启动 Spark 集群。命令如下，界面如图3- [13](#bookmark180)

所示：

cd/usr/local/spark-1.4.1-bin-hadoop2.6/

sbin/start-all.sh

rootemaster:/usr/local/spark-1.4.1-bin-hadoop2.6t sbin/start-all.ah

atarting org.apache.apark.deploy.master.Master,logging to /uar/local/apark-1.4.1-bin-hadoo p2.6/sbin/…/logs/apark--org.apache.apark.deploy.master.Master-1-maater.out

3lave2:atarting org.apache.spark.deploy.worker.Worker,logging to /usr/local/apark-1.4.1-b in-hadoop2.6/sbin/../logs/spark-root-org.apache.apark.deploy.worker.Worker-1-slave2.out

3lave1:starting org.apache.spark.deploy.Worker.Worker,logging to /usr/local/spark-1.4.1-b in-hadoop2.6/sbin/../logs/spark-root-org.apache.apark.deploy.worker.Worker-1-slave1.out

root@master:/usr/local/spark-1.4.1-bin-hadoop2.64 jps

416 SecondaryNameNode

226 NameNode

580 ResourceHanager

1084 Jpa

925 Master

图3-13 启动 Spark 集群

(3)NameNode 节点上 Spark 集群启动界面。

登录 Slave 节点，执行jps 命令，查看 Slave 节点上 Spark 是否启动了，当显示如下界面 时，如图3-14所示，说明 Slave 节点上的 Spark 启动成功。

**Foot@slave1:/usr/local/hadoop-2.6.0#JPB**

**434 Worker**

**540** **Jps**

**124 DataNode**

**238 NodeManager**

图3-14 验证 Slave节点上的 Spark 启动成功

Slave 节点上的 Spark 启动成功界面如图3-15 所示，至此 Docker 上 的 Spark 安装成功。

接下来启动 Spark 集群的 Python 界面，在 NameNode 节点上输入如下命令：“./bin/ pyspark”, 当出现如下界面时，代表 Python 界面启动成功，用户就可以输入 Python 程序进 行实验了。



15/08/1411:38:42 INFO storage.BlockManagerMaster:Trying to register BlockManager

15/08/1411:38:42 INFO storage.BlockManagerMasterEndpoint:Registering block manager localr pst:47350 with 265.1 MB RAM,BlockManagerId(driver,localhost,47350)

15/08/1411:38:42 INFO storage.BlockManagerMaster:Registered BlockManager Melcome to

1\_/\ vergion 1.4.1

Using Python version 2.7.6(default,Mar 22201422:59:56)

SparkContext available as ac,HiveContext available as sqlContext.

图3-15 启动 Spark 集群的 Python 界面

**3.1.7** **Docker 集中化** **Web 界面管理平台** **shipyard**

RethinkDB 是一个 shipyard 项目的 一 个 Docker 镜像，用来存放账号(account) 、 引 擎 (engine) 、服务密钥(service key)、扩展元数据(extension metadata)等信息，但不会存储任 何有关容器或镜像的内容。 一般会启动一个 shipyard/rethinkdb 容 器 ，shipyard-rethinkdb- data 使用它的/data 作为数据卷供另外一个 RethinkDB 挂载，专门用于数据存储。

1. 两个概念

(1)engine

一个 shipyard 管理的 Docker 集群可以包含一个或多个 engine (引擎),一个 engine 就 是监听 TCP 端口的 Docker daemon 。shipyard 管理 docker daemon 、images 、containers 完 全基于 Docker API,不需要做其他的修改。另外，shipyard 可以对每个 engine 做资源限制， 包括 CPU 和内存；因为 TCP 监听相比UNIX socket 方式会有一定的安全隐患，所以 shipy- ard 还支持通过 SSL 证书与 Docker 后台进程安全通信。

(2)RethinkDB

RethinkDB 是 一个 shipyard 项目的一个 Docker 镜像，用来存放账号(account) 、 引 擎 (engine) 、服务密钥(service key)、扩展元数据(extension metadata)等信息，但不会存储任 何有关容器或镜像的内容。 一般会启动一个 shipyard/rethinkdb 容器 shipyard-rethinkdb- data 来使用它的/data 作为数据卷供另外 rethinkdb 一个挂载，专门用于数据存储。

**2.** **搭建过程**

(1)修改 TCP 监听

shipyard 要管理和控制 Docker host 的话需要先修改 Docker host 上的默认配置使其 监听 TCP 端口(可以继续保持 UNIX socket),有以下两种方式。

① 启动 docker daemon。如果为了避免每次启动都写这么长的命令，可以直接在/etc/ init/docker.conf 中修改。

sudo docker -Htcp://<0.0.0.0>:4243 -Hunix:///var/run/docker.sock -d

② 修改/etc/default/docker 的 DOCKER\_OPTS。

|  |  |
| --- | --- |
| DOCKER\_OPTS="-H | tcp:// <127.0.0.1>:4243 -Hunix:///var/run/docker.sock" |
| ·重启服务 |  |
| $sudo docker | -Htcp://<0.0.0.0>:4243 -Hunix:///var/run/docker.sock -d |
| · 验 证 |  |
| s netstat -ant | |grep 4243 |

tcp6 0:::4243 44:# LISTEN

(2)启动 RethinkDB

shipyard (基于Python/Django) 在 v1 版本时安装过程比较复杂，既可以通过在 Host 上 安装，也可以部署 shipyard 镜像(包括 shipyard-agent 、shipyard-deploy 等组件)。v2 版本简 化了安装过程，启动两个镜像就可以完成。

① 获取一个/data 的数据卷。

$sudo docker run-it -d --name shipyard-rethinkdb-data \

--entrypoint /bin/bash shipyard/rethinkdb -1

②使用数据卷/data启动 RethinkDB。

docker run -it -P -d --name shipyard-rethinkdb \

--volumes-from shipyard-rethinkdb-data shipyard/rethinkdb

(3)部署 shipyard 镜像

① 启 动 shipyard 控制器。

sudo docker run -it -p8080:8080 -d --name shipyard \

--link shipyard-rethinkdb:rethinkdb shipyard/shipyard

至此，已经可以通过浏览器访问http://host:8080 来 访 问 shipyard UI 界 面 了 。

② 第 一 次 run 后，关闭再次启动时直接使用。

sudo docker

sudo docker

**3.** **登录**

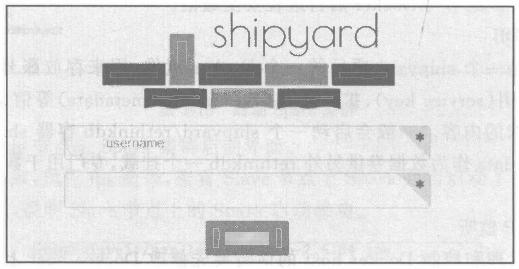
stop

start

shipyard shipyard-rethinkdb shipyard-rethinkdb-data

shipyard-rethinkdb-data shipyard-rethinkdb shipyard

登录界面如图3 - 16所示。



▲

a password

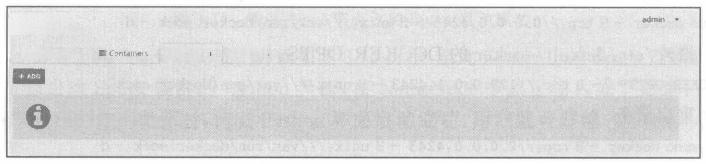
LOGIN

图3-16

默认用户名/密码为 admin/shipyard,

shipyard 登录界面

登录后的界面如图3 - 17所示。

shipyard

Dashbouard Enlne 黑 Events

Engines

There are mo engnes in the duster.

图3-17 shipyard 登录后界面

(1)Dashboard 界面

Dashboard 展示在添加 engine 时指定的 CPU 以及内存的使用情况。

(2)Containers 界面

shipyard 管理的所有 Docker 主机的所有容器，包括 stop 和 running 状态的。可以直接 点击 DEPLOY 按钮来从镜像运行出其他容器，与 Docker run的选项几乎相同，可以限制 CPU 和内存的使用，详见 shipyard 的 containers 文档。

(3)engine 管理

一个 engine 就是一个 Docker daemon,Docker daemon下启动着多个 containers, 可以 对 engine 限制 一 个整体的CPU 和 内 存 限 制 ，shipyard 通 过TCP 端 口 连 接 daemon 。 需 要 注

第3章 大数据平台虚拟化解决方案

意的是 Docker client 与 server 的版本问题(因为 shipyard 目前还在快速的完善过程，不同

版本的 Docker 应该是向下兼容的)。

**3.1.8** **DockerUl**

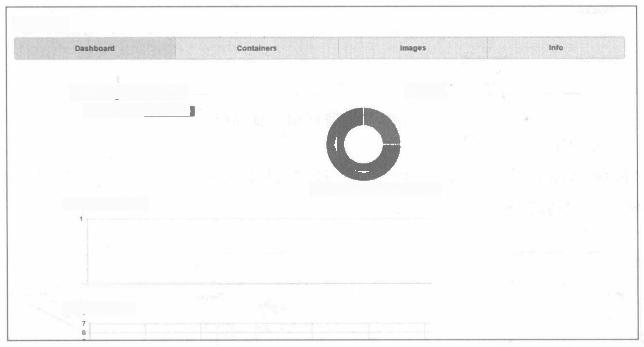
**1.DockerUI 界面搭建**

run cmd docker run-d -p9000:9000-V/var/run/docker.sock:/var/run/docker.sock dockerui/dockerui Open your browser to http://<dockerd host ip>:9000

**2.DockerUI** **管理界面的使用**

(1)Dashboard 界面

Dashboard 界面显示着 Docker 的运行状态，如图3-18 所示，同时能看见正在运行的 容器。

DockerUi

Running Containers Status

·supefed\_bet esmta

■Runing stopped Ghost

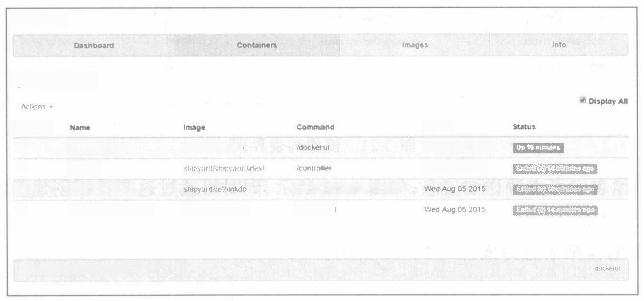
Contalners created

lmages created

图3-18 Docker的运行状态界面

(2)Containers 界面

图3-19显示运行中和暂停的容器，点击容器前面的方框选择action 能够暂停容器、kill 容器等一系列操作。点击容器的名字能显示详细信息，如图3-20所示。



DockerUI

Containers:

Created

Wed Aug 052015

Wad Aug 0s 2ots

Ausrtvrettinaotb -brd al

&ingn

Dseta API Vuninn vt.I7 UI Vamion:wt.7A

hag

thpyaidhelhnkn

snipyare reghiskatb-data

spyanretnmtdn

@

a

■

dockeruvtocieru

Mted

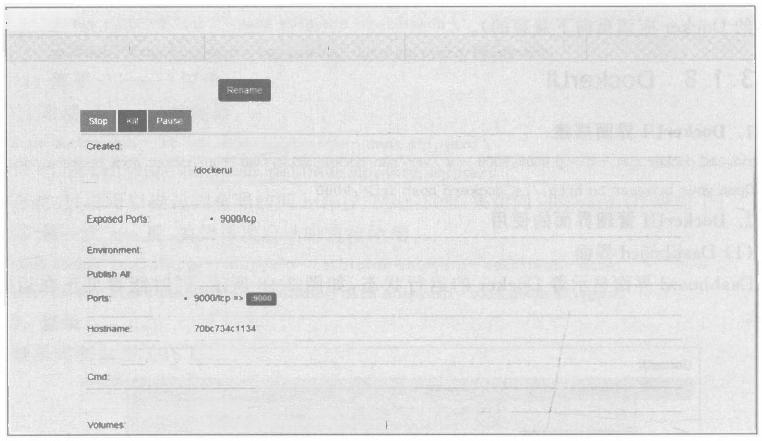


An

**图3-19** **容器列表界面**

43 >>

大数据技术及行业应用



Pockerul

Dashboard Containes

**Container./stupefied\_bell**

2015-08-05T12-00:55.543384423Z

Path

0

talse

<172.17.0.4>

Pdockerun

Cvarrun/docker.socr."Narfrun/docker.sock)

Entrypoint

IPAddress

images

NF

Into

图3-20 容器详细信息界面

(3)Images 界面

管理镜像的界面和 containers 界面差不多，但是本质上是不一样的，不要把镜像和容器

混为一谈，镜像列表界面如图3-21所示。

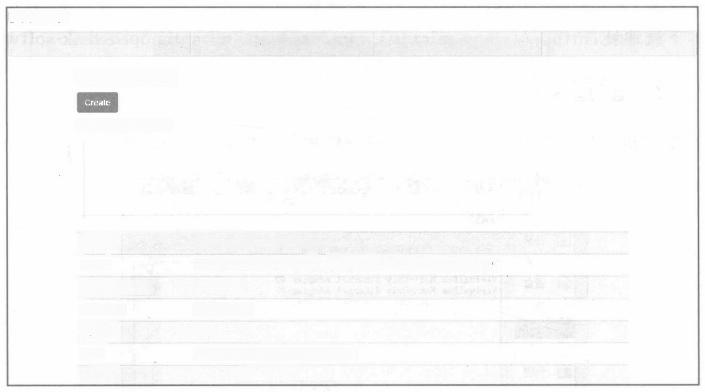
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DockerUI  Dashboard Containen Images Into  **lmages:**  Achons-  Action Id **Repository** rtusist Ceated | | | | | | |
| □ | d2aDec he6ta4efd  39067844403372157 | ubuntr14.0  daocloud io/tbrary/ava.tatest | 180 MB  560 MB | Fri  Wed | Jul  Jun | 102015  172015 |
| 01d0c71830abee950 shipyardretthinkdb:tatest 282 MB Tue Jun 162015 | | | | | | |
| 9ac7996209b0a2333 dockeruidockerut tatest 5 MB Thu May 212015 | | | | | | |
| @ ba94b7a2060699e14 shipyard/deploy.lalest 6 MB Sun Mar 082015 | | | | | | |
| f1be70d106c1171a shipyard/shipyard.tatest 35 MB Tue Mar 032015 | | | | | | |
| b739452c2040507 hipyard/shipyard-clt iatest 140 MB Fri Jan 022015 | | | | | | |

图3-21 镜像列表界面

点击镜像名字可查看详细信息，如图3-22所示，同时可通过这个界面创建新的容器。 (4)Info 界 面

Docker 的详细信息界面，如图3-23所示。

第3章 大数据平台虚拟化解决方案



DockerUI

Dashboard Containers images Into

lmage:ubuntu:14.04

**Containers created:**

2015-07-09T19:28.34.439309646Z

29460ac934423a55802tad24856827050697b4a9133550b193c82762IbEdhBT

nta(180 MB)

da360632003c

User.

Trtin'she,c",W(nop)CMD P'tintbashi'T

Volumes

Created:

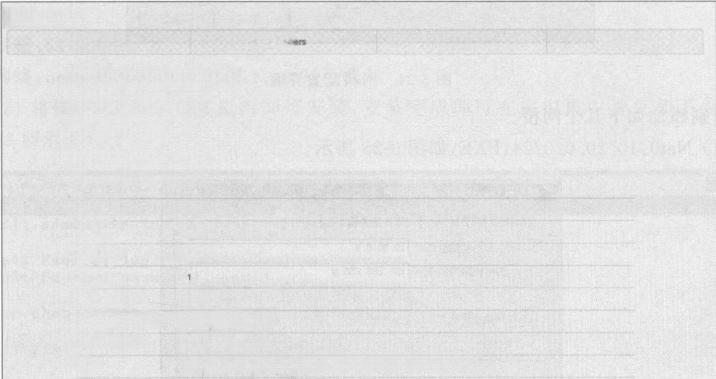
Parent

Sze (Mitual szey

Hostname:

Cmd

图3-22 镜像详细信息界面



DockerUI

Dashboard Contat images infe

**Docker Information**

Endpoint:dockerapi

Api Version:v1.17

Version:1.7.1

Git Commit:786b29d

Go Version:go1.4.2

4

50

take

CPUg:

868 MB

Ubuny 14.04.2LTS

3.16.0-30-penere

3TAVTFJP ATRVLMMM BANL VBOR:7BFALCVC GKLB4C5G.MAO7IFZP

Toul Memory

Operating System.

Kemel Version:

ID:

Contalners

images

Debug

图3-23 Docker的详细信息界面



**3.2 OpenStack 搭建**

本节将进行 OpenStack 的搭建演示，采用VirtualBox 虚拟机和fuel 自动化部署工具。

**3.2.1** **下载工具和镜像**

(1)虚拟化工具：VirtualBox4.3。

(2)下载地址：<https://www.virtualbox.org/wiki/Downloads>。

(3)安装 OpenStack 工具：fuel6.0。

 固 阿

(4)下载地址：<https://www.mirantis.com/products/mirantis-openstack-software/>。

**3.2.2** **配置网桥**

点击虚拟机中的管理 → 全局设定 → 网络 → 仅主机网络，如图3-24所示。



设活

**网络**

MAT网络 仅主机 Oloxt-0nly)网 络 0

VirtualBox Most-Only Ethernet Adapter

VirtualBox Host-Only Etharnet Adaptar

VirtualBom Host-Only Ethernet Adapter

扩展

代理

显示所有可用仅主

机Olost-0nly)网

络.

帮助00

virtxslko. 回 常规 回 热键

更新 语言

显示

确定

取消

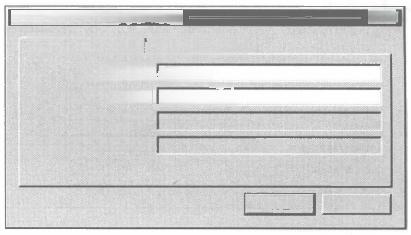
区

分别添加如下几个网桥。

(1)Net0:<10.20.0.0>/24:PXE,

图3-24 网桥配置界面

如图3-25所示。



**仅主机dfost-0nly) 网络明细** 回 区

主机虚拟网络界面QV|mc? 服务器①)1

Trv 地址C):<10.20.0.1>

IPw4 网络掩码 M):|<25.255.255.0>

IPv6地址P):

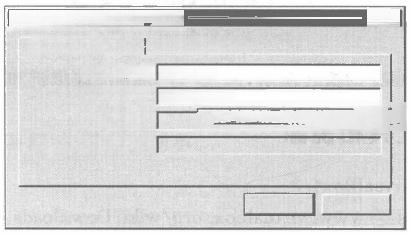
IPv6网络掩码长度L):

确定

取消

图3-25 添加 PXE 网桥

(2)Net1:<192.168.4.0>/24: 管理、存储，如图3-26所示。



**仅主机dtost-0al?)** **网络明细** 区

主机虚拟网络界面N|: DBCP 服务器①)1

IPv4 地址(π):17216.0.1

IPw4 网络掩码(M):|<255.255.255.0>

Iro 地址P): 厂 示主机的好格界的RF

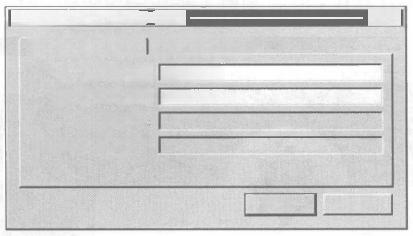
IPv6网络掩码长度L):|

确定

取消

图3-26 添加存储、管理网桥

(3)Net2:<172.16.0.0>/24: 公 网IP、浮动IP, 如图3-27所示。



**仅主机(Kcst-0nly) 网铬明细**

主机虚拟同络界面A)|mer 服 势 器 @ 1

4地址①): 192168.4.1

IPv 网络掩码(@): <255.255.255.0>

IPv8地址①):

IPv6 网络掩码长度L):

确定

区

取消

图3-27 添加公网IP 与浮动 IP 网桥

**3.2.3** **安装** **fuel**

(1)配置 fuel 虚拟机，硬件环境如下。

内存：4 GB。

CPU:2个。

硬盘：64GB。

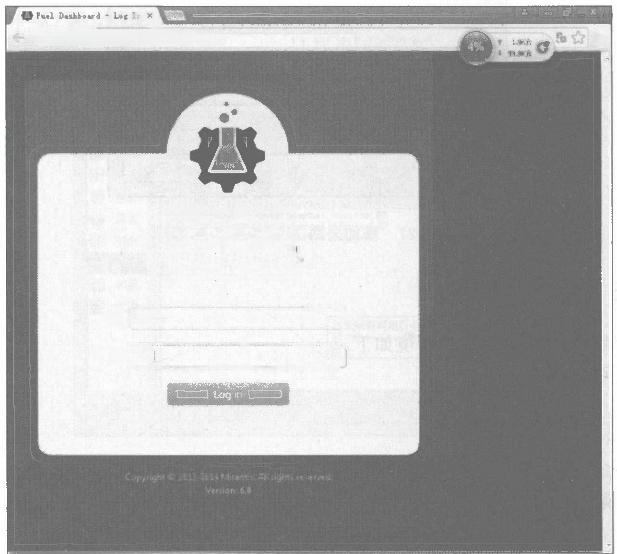
网络：net0(必选)、net1(可选)、net2(可选)。

(2)将镜像放入，启动虚拟机即可安装，安装完成即可在虚拟机中看见如下信息，如 图3-28所示。

|  |  |  |  |
| --- | --- | --- | --- |
| **fuel** | **-master[正在运行]** | **-Oracle** **VI** **VirtualBox** | |
| **控制视图设备帮助**  Default administrator password:r88tme  Default Fuel UI login:admin  Default Fuel UI password:admin  Please change root password on first login. | | | |
| fuel login:  #其非转样非排共其林共共非排非样非其非  # Welcome to the Fuel server  其共#共  Server is running on x86\_64 platform | | | 共共样  # |
| Fuel UI is available on:http://<18.28.0.2>:8888  Default administrator login: root  Default administrator password:r08tme  Default Fuel UI login:admin  Default Fuel UI password:admin  Please change root password on first login.  fuel login: 别 吧回 [Rieht Ctrl | | | |

图3-28 fuel安装完成

( 3 ) 等 待fuel 虚拟机安装完成即可在浏览器中查看，输入 [10 .20 .0 .2](10.20.0.2) :8080 即可进入登 录界面，用户名和密码默认都是 admin, 如图3 - 29所示。

三

C<10.20.0.2>:8000/login

**FUEL**

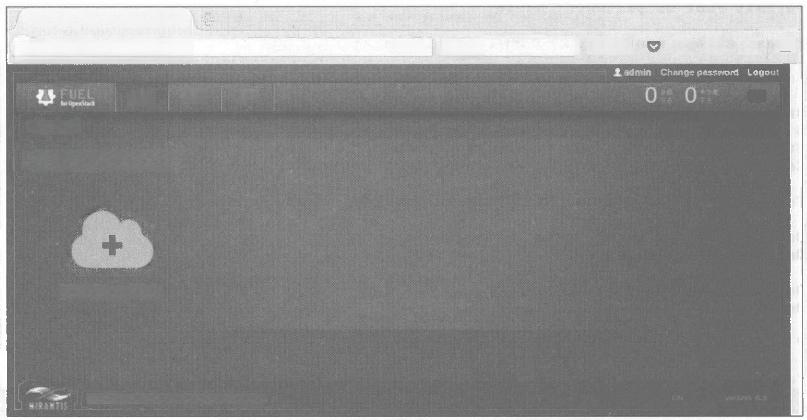
for OpenStack

上 adrin

P ……

**图3-29** **fuel 登录界面**

**(4)登录成功之后的界面，如图3-** **30所示。**



Fuel Dashboard-我 … ×

10.20.0.28000/mclusters

环境 后主

道 五 / 环 境

**我的OpenStack环境**

新建OxenStack环城

Cwwr e ls-old

搜 费

ve

☆ 自

4 舍

支持



三

图3-30 fuel 登录成功

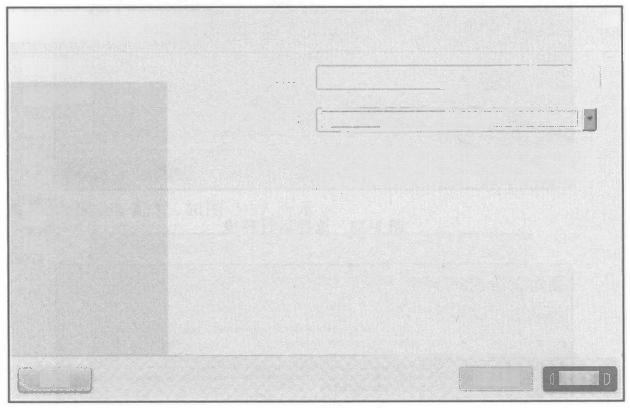
第3章 大数据平台虚拟化解决方案

**3.2.4** **安装** **OpenStack 平台**

**1.** **创建** **OpenStack 环境**

进入 fuel 界面后点击创建 OpenStack 环境即可，创建步骤如下。

(1)填写OpenStack 名称与选择 OpenStack 的版本，如图3-31所示。



×

名称和版本

JlieTest

部署模式

计算

网络

后端存储

附加服务

完成

★后退

Juno on CentOS 6.520142-6.0)

This opton wt nstal the OpenStack Juno pactages usng a

CemtOS based operating system.Wth high avalabdty feathures

bultin,you are getting a robust,enterprse-grade OpenStack

deployment.

**名称**

**OpenStack** **所本**

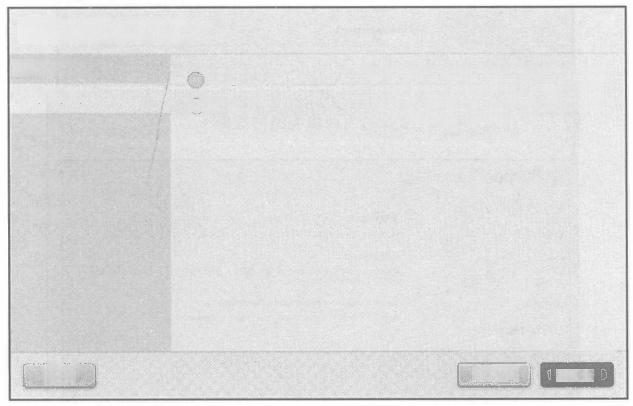
**新建OpenStack环境**

前进 →

取消

图3-31 填写OpenStack 名称与版本

(2)选择部署模式，如图3-32所示。



新建OpenStack 环 境

√ 名称和版本

**HA** **多节点**

部署模式

计算

网络

后端存储

附加服务

完成

前进→

in thes canfquration the OpenStadk controler

deployed separately from the compute and cinder

nodes.This mode asumes the presence of 1 controler

node and 1 or more computeichder nodes.You can

add more nodes to scale your doud later.

**多节点**

◆ 后 退

取消

◎

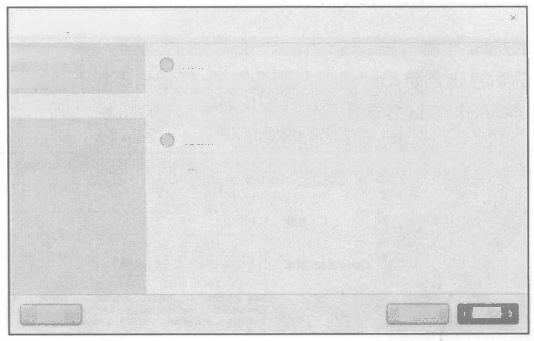
×

图3-32 部署模式

(3)选择运行环境，如图3-33所示。

(4)网络类型的选择，如图3-34所示。

大数据技术及行业应用

**(5)存储模式的选择，如图3** **-** **35所示。**

**新建OpenStack环境**

√ 名称和版本

√ 部署模式

计 算

网路

后端存储

附加照务

完成

**KvM**

如果在件上适行OpenStac, 请选择该管建据类型

**QEMU**

●

妈果在虚机上语行OpenStact,请选择该管建器类型

**vCenter**

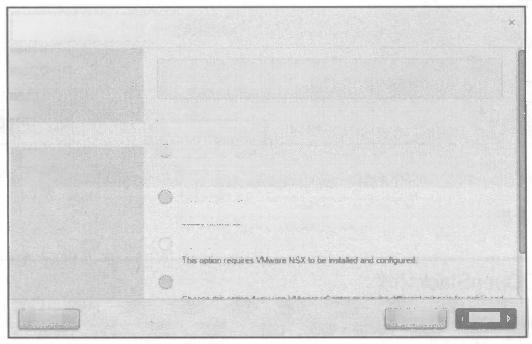
S t p hae aCtr shon Ese

取 清

◆ 后 退

前 进

**图3-33** **选择运行环境**



**新建OpenStack环境**

√ 名称和版本

部署模式

计算

网 络

◎ **Neutron VLAN**

后端存储

附加服务

**Neutron GRE**

完成

seang tnat mut anpen CnE neen The spten stport upo

**Neutron with VMware NSX**

**Nova-Network**

◆ 后 退

Neuron wth NSX can be used with Experimental mode only Please,took atthe documentation to enable t

Choose the prtate taves)network cantquraion.The choce you make here camot be

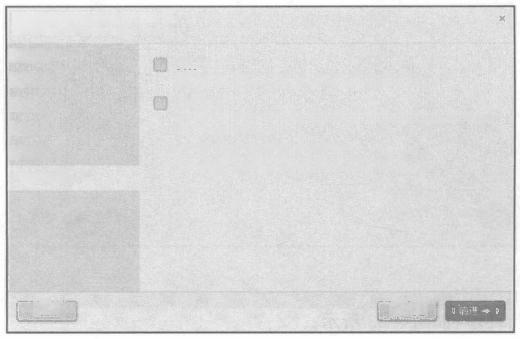
changed aher you fiish the wicned.More intormation see the Mir mntl.CoenStack Panmg Gusde ftr Nehork Topsksr

The netwarlkng equpment must be canfiqured torVLAN seymertaton.This apton wpports up to 40s5 netsarte

前进→

取 消

**图3-34** **网络类型的选择**

**新建OpenStack环境**

|  |  |  |
| --- | --- | --- |
| 名称和瓶本 |  |  |
|  |  | **安** **萄Sahara** |
| 部署模式 |  | 利用Sahua 成在备大顺本厂商的OpenSack上的部密Hadoon#群 。 |
| 计算 |  | **安** **装Murano**  利用Murano能将基于Wndows 的数雨中心服务，加活动目荣、H、Merosot SQL和 |
| √ 网 络 |  | ASPNET 部客到OpenSack上 |
| √ 后端存储 | ☑ | **Install Calometer (Open Suack Telemetry)**  Cetmetrptowtas metering and montorng of an OpenSuact ddutt |
| 附加服务 |  |  |
| 完成 |  |  |

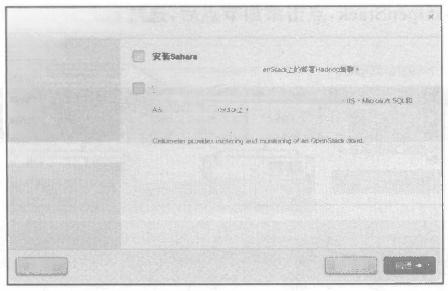
取 消

◆ 后 退

图3-35 存储模式的选择

**第3章** **大数据平台虚拟化解决方案**

**(6)选择服务环境，如图3-36所示。**



新建OpenStack

名称

邮需模式

▼ 计 算

同 备

▼居端存储

附 加 服 务

宽 成

取 消

机用Satars端 在 备 大 厂m)Ope

安Mirang

利用Wug 将基于Wndaw 的数据中心脑劳·如动目、

☑ Inatal Celomotar (Opan Stack Talemetry)

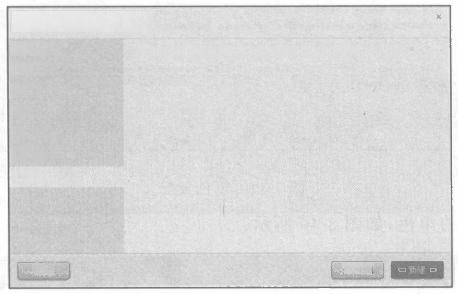
环 境



后 退

图3-36 选择服务环境

(7)完成 OpenStack 配置，如图3-37所示。



新建OpenStack 环 境

名称和版本 部署准备状端!请选择立即部署或者更多配置，请前往开填控制面板

部客模式

计 算

问 路

后端存储

附服

完 成

+ 后 退

取 满

**图3-37** **(8)点击完成后界面，如图3-38所示。**

**完成配置**

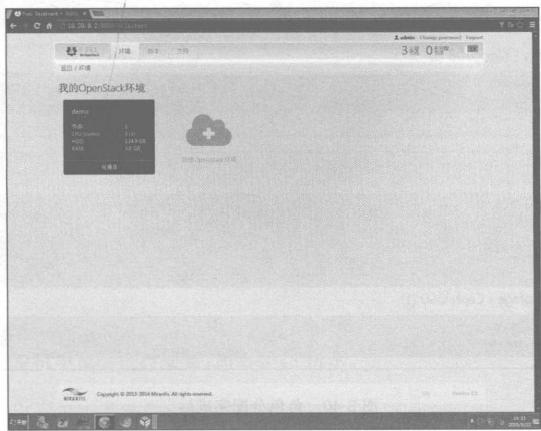


图3-38 新建完成的 OpenStack 环境

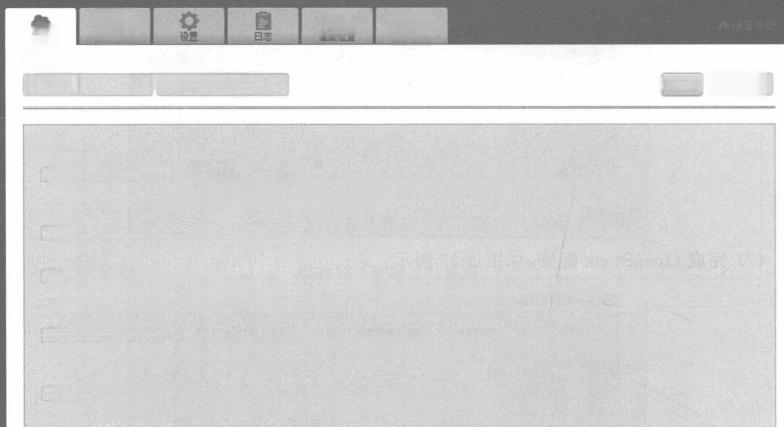


大数据技术及行业应用

**2.** **安** **装** **OpenStack 环境**

( 1 ) 点 击 进 入 testOpenStack, 点击添加节点后，选择一个角色，对应的在下方选择一 台

主机，如图3-39所示。



品

**Filter By**

 Node name/imac

**分配角色**

**Controllr** A

The centreller initiates ochestretion activties and proudes an estemal APt Other components fike Glance fimage storagel.keystenefidentity managememt,Honzon (OpenStack doshboord)and Newo-Schedwler are instolled en the controller as well

**Compute**

*A compute node ceotes,manages and temminates vitvol machine instances.*

**Storage -Cinder**

*Cinder prowidesstheduling dfblactstoroge esowres.bpitally deluvered over sCS(VMWare vCente,and other compatibl boctendstoroge systema.Blct storoge can be wsed for dotabose storoge,erpandoble fie systems,orprwiding a server with ovcess to row block level devices*

**Storage -Ceph OSD**

Ceph stornge can be cenfigumd to provide sternge for bloct volumes iCinderl,imoges fGlancel)and ephemerolinstance sternge (Novel It can also provide ctjett storoge through the 53 and Swift APNjfSee settings to enabile each)

**Telemetry -MongoDB**

*A feature-complete and recommended dotobase for stornge of metening doto from OpenStoct Telemetty (Ceilometer).*

分组

硬件信息

取 消 应用变更



动作

点

**图3-39**

**(2)分配完成后的角色，如图3-40所示。**

**角色分配**

豆 

节点

分组

角 色

健康检查 动作

Flter By

Node name/mac

班盘配置 网答配置 +增加节点

■选择全部

Controller,Telemetry -MongoDB (1)

Untitled (fc:dd)

CONTROLLER·MONGO

D 已就绪

CPU:1(4)HDD:64.0GB RAM:16.2 GB

■选择全部

0

Compute (1)

Untitled (71:dd

COMPUTE

D √ 已就绪

CPU:1(4)HDD:64.0 GB RAM:16.2 GB

■选择全部

☆

Storage

-Cinder,Storage -Ceph

Untitled (90:1b) CINDER·CEPH-OSC

OSD (1)

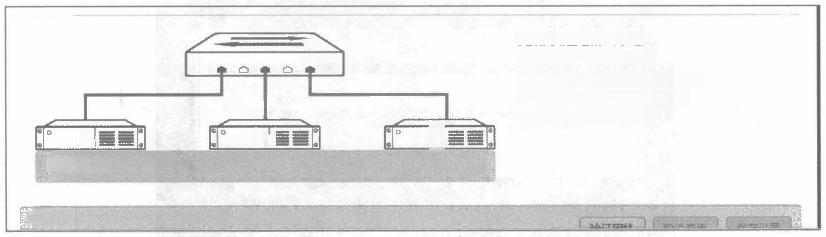
D 已效绪 CPU:1(4)HDD:64.0 GB RAM:16.2 GB

■选择全部



**图3-40** **角色分配完成后**

(3)点击选项栏中的网络，然后点击验证网络，如图3-41所示。

**完成网络验证的4个步囊：**

1. 每个节点启动测试监听帧

2.每个节点发送802. 1Q封 装 的UDP帧

3.每个节点从其他节点注册测试赖

4.在所有端口发送DHCP 消 息

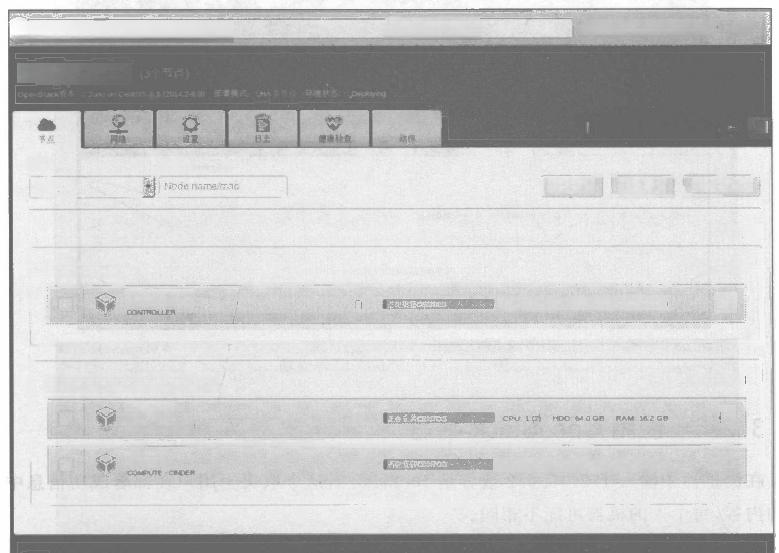
g

gE

验证成功。网络配置正确。

图3 - 41 网 络 验 证 成 功

(4)节点网络和硬盘的调整：在节点页面，选择全部节点，点击网络配置进入网络配置 界面，将管理网段和存储网段放在 net2 网段中；硬盘的调整，用户可根据应用需求进行相应 的配置。在所有配置完成后点击部署，如图3-42所示。

←②<10.20.0.2>:8000/#cluster//nodes

testOpenstark

分 组

Fiter By

角 色

**Controller (1)**

contral (ct:41)

vcQ 搜察 ☆ 自 ☑ 食 

0

础盘配置 网结配置 + 增 加 节 点

选择全部

**■** **选** **择** **全** **部**

D Cpu:1 H006OGB MA B2GB ☆

**Compute,Storage -Cinder (2)**

cs(Sb de)

COMPUTE-CNOER

cs (7:11)

D

D

Pu1Ho G RA 2B

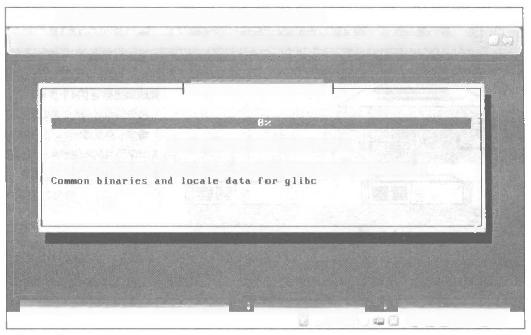
**选择全部**

0

☆

图3 - 42 部 署 过 程

(5)对应虚拟机中也会进行安装，如图3-43所示，



openstack-slave1 [正在运行]-Oracle VM Virtual8ox

虚拉电脑报告其内所安装的接作系统支持我标自动切换这章味教原标不需要为该虚拟电脑为占使用，当原标

Tackage InstalTatron

Packages completed:9 of Z99

Installing glibc-common-Z.12-1.132.e16\_5.Z.xB6\_64(187 HB)

Spact>selects

年

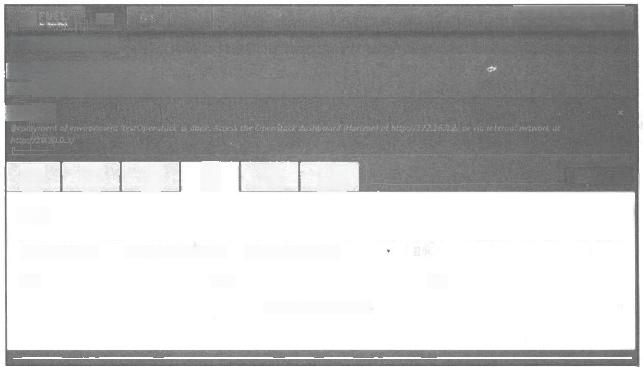
KF12>next xcreen

回左Ctrl+左Al

KTab>Khlt Tab>betueen clements

图3 - 43 虚 拟 机 中 部 署 安 装

(6)漫长的等待后可在Web 界面中看见部署成功的信息，如图3-44所示。



 环填

置近/环缝/tkOpkrtd

testOpenstack ( 个 节

CrStsdB本：1oen-4G026

*Succ4ss*

盘 品 盘

日志

日志 Fug M3sker · 来重 Web hackend

日期

内朝过要图点自把到任间日意。

不 使 式 节 开 ： 可

30■

· 最低级别 ERRO R

盘

◆法器要



继康检置



消息

Level

图3 - 44 OpenStack 部 署 成 功

**3.2.5** **使** **用** **OpenStack 平** **台**

在浏览器中输入<172.16.0.2>或者是<10.20.0.3>,这个取决于用户在部署成功信息中看 见的内容，每个人的机器可能不相同。

用户名和密码默认是 admin, 如图3-45所示。

登录后界面，如图3-46所示。

**1.** **创建虚拟机实例**

(1)点击 Project→Instances→Launch Instance ,过程如图3-47所示。

(2)进入配置虚拟机界面，如图3-48所示。

(3)添加密钥对，在Xshell 中先生成密钥对，再将公钥复制到虚拟机中，点击导入即可 完成，如图3-49所示。

第3章 大数据平台虚拟化解决方案

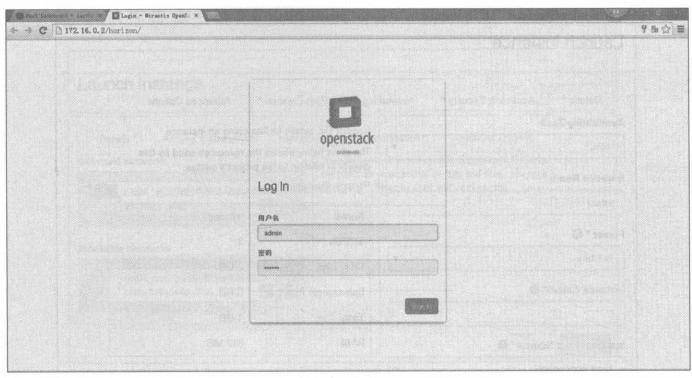
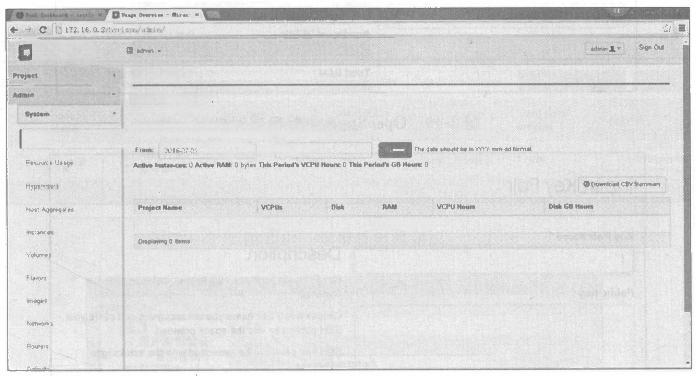


图3-45 OpenStack 登录界面

**openstack**

Overview

Usage Summan

Select a period of time to query its usage:

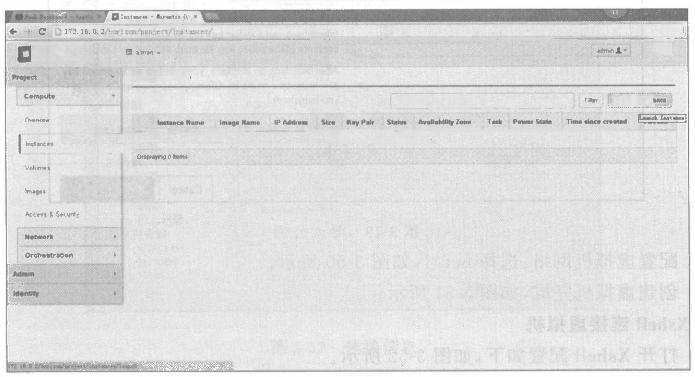
Oaniew

Te:2016-07-9

Usage

Netamato aplap

图3-46 OpenStack 欢迎界面

**openstad**

Instances

Instances latance Reme ·Har

Notame lediapig

☆ 日

Sen Oa

\*Laush m

图3-47 创建虚拟机实例界面

大数据技术及行业应用

|  |  |
| --- | --- |
| x  **Launch Instance** | |
|  | |
| Details\* Access &Securily\* Networking\* Post-Crealion" Advanced Optians | |
| **Avallability Zone**  nova  **Instance Name \***   |  | | --- | | jietest |   **Flavor\*O**   |  | | --- | | m1.tiny |   **Instance Count\*O**    **Instance Boot Source\*○**   |  | | --- | | Boot from image |   **lmage Name**  TestVM (12.6 MB) | Specify the details for launching an instance.  The chart below shows the resources used by this  project in relation to the project's quotas.  Flavor Details   |  |  | | --- | --- | | **Name** m1.tiny | | | **VCPUs** | 1 | | **Root** **Disk** | 1 GB | | **Ephemeral** **Dik** | 0 GB | | **Total** **Disk** | 1 GB | | **RAM** | 512 MB | |
| Project Limits  **Number of Instances**  0 of No Limi Used    **Number of VCPUs**  Umt Used    **Total RAM**  0 of No Limit MB Used |

图3-48 OpenStack 上配置虚拟机



x

**Import Key Pair**

Key Pair Name=

**Description:**

Key Pairs are how you login to your instance after it is launched.

Choose a key pair name you will recognise and paste your SSH public key into the space prowided.

SSH key pairs can be generated with the ssh-keygen command:

ssh-keygen -t rsa -f cloud.key

This generates a pair of keys.a key you keep prvate (cloud.key)and a public key (cloud.key.pub).Paste the contents of the public key fle here.

After launching an instance,you login using the prvate key the usemame might be i depending on the image you launched);:

ssh -i cloud,key cusernane>@kinstance\_ip>

Cancel Import Key Par

**Public Key'**

图3-49 导入公钥

(4)配置虚拟机网络，选择 net04, 如图3-50所示。

(5)创建虚拟机完成，如图3-51所示。

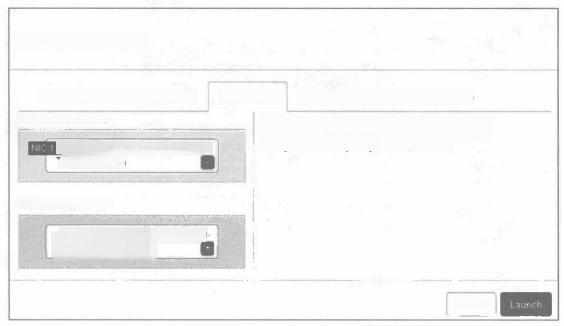
**2.Xshell** **连接虚拟机**

(1)打开 Xshell 配置如下，如图3-52所示。

(2)用户名和密钥选择如图3-53所示。

第3章 大数据平台虚拟化解决方案

(3)配置完成后，连接虚拟机，连接结果如图3-54所示。



x

**Launch Instance**

Networking\* Post-Creation" Aianced Optioms

Choose network from Awailable networks to Selected

networks by push button or drag and drop,you may

change NIC order by drag and drop as well.

**Available networks**

nel04\_ext(et95d596-a32e-44cl-

h+0c54c446a05b)

Cancel

net04(40820610-519d-4bae-bbF

5521fbce1a0e)

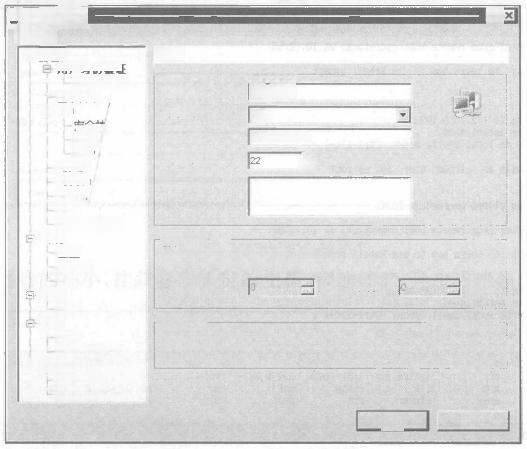
Details\* Access &Securty\*

**Selected networks**

图3-50 虚拟机网络配置

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instances** | | | | | | | | | | | |
| **Instances** Instance Name w Fllter Filter | | | | | | | | Launch Instance | | R U Tenminste instsnces | |
| □ | Instance  Name | lmage  Name | IP Address | Slze | Key  Pair | Status | Availability  Zant | Task | Power  State | Time since  Created | Adions |
| 0 Tes | | TestVM | <192.168.111.2> <172.16.0.133> | m1.smal | jie | Active | nāra | None | Dnfin? | 1分钟 | Create Snapshot |
| Displaying 1 item | | | | | | | | | | | |

图3-51 虚拟机创建完成



T

限制(): 分 钟

取 消

|  |  |
| --- | --- |
| 连接  常规一  名称(N):  协议(P):  主机(H):  端口号(o):  说明(D): | pengar  SSH  <172.16.0.133>  习 |

重新连接

「连接异常关闭时自动重新连接(A) 间隔V): 秒

TCP选 项

「 使 用Nage算法(U)

报

类 别C):

日连拖

用户身份验证

登录提示符 登录脚本

日-5SH

姓全性

隧道

SFTP

TELNET

RLOGIN

SERIAL

代理

保持活动状态

终端

键盘

V 模 式

高级

外 观

边距

高级

跟踪

日志记录

白文件传输

XIVMODEM

ZMODEM

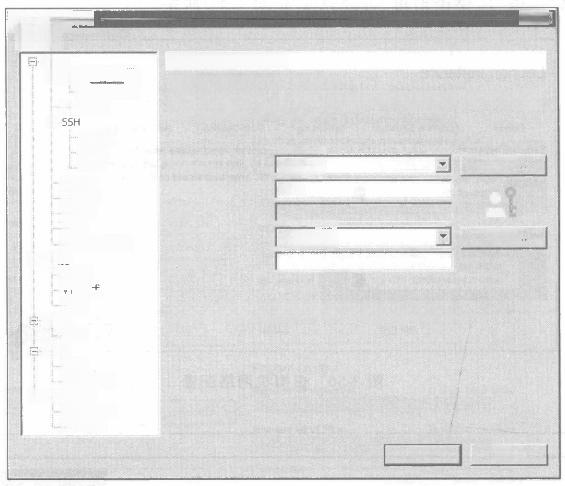
确定

图3-52 连接信息

>>

57

大数据技术及行业应用

**penstack 性**

类 别C):

连

接

B|

用户身份验证

登录提示符

登录脚本

白

安全性 隆道

SFTP

TELNET RLOGIN SERIAL 代理

保持活动状态 白 终!端

键惧式

高级

外观

-边距

高级

跟踪、

日志记录

白文件传输

XIYMODEM ZMODEM

? 区

连接>用户身份验证

请选择身份验证方法和其它参数。

会适属性中此部分是为了登录过程更便捷而提供的。如果需要安全性很高

的状态的话建议您空出此字段。

方法(M):

设置(5)。

Puhk Key

cirros

用户名(U):

密码(P):

用户密钥(K):

浏览(B)…

rsa 2048

密码(A):

注释:公钥和Keyboard Interactve仅在551VSFTP协议中可用。

取 消

确定

图3-53 用户名与密钥选择

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| openstack | -Ishell 5 Cree for Home/School) 回回区 | | | | |
| 文件①)编辑C)查看V)工具T)选项卡B)窗口)钢助O  唱▼时 · ③ · A▼ @R6向  0 sch-//cirros⁴172.16.0133:22  要添加当前会话，点击左侧的前头按扭。  1 openstack | | | | | |
| Disconnected from remote host(openstack)at 10:12:56.  Type help'to learn how to use xshell prompt.  lc:  [c\*  Connecting to <172.16.0.133>:22..  Connection established  To escape to local shell,press 'Ctrl+Alt+]'.  Please login as 'cirros'user,not as root  \*C  Connection closed ty foreign host.  Disconnected frou remote host(openstack)at 10:13:04,  Typehelp'to learn hou to use xshell prompt.  [c:P-]\*  Connecting to <172.16.0.133>:22..  Connection established.  To escape to local shell,press 'Ctrl+hlt+]" | | | | | |
| $1s  $1s  bin  boot  dew  $■ | etc  home  init | initrd.img lost+found old-root root lib media opt Tn  linxrc mnt PTOC sbin | sys  tap  usT | war  mlimz | |
|  | | |  | |  |

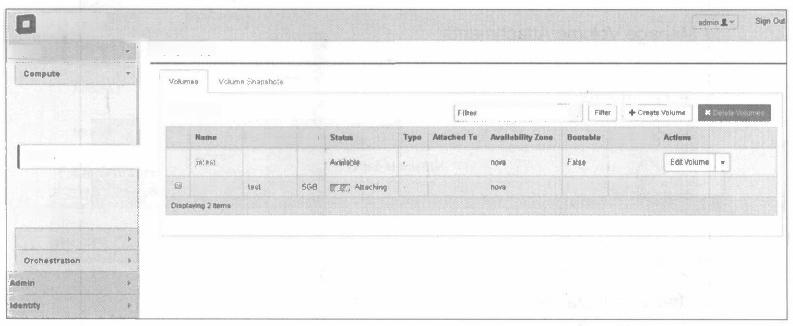
图3-54 连接 OpenStack 虚拟机

第3章 大数据平台虚拟化解决方案

Upload to lmag

**3.** **动态挂载卷**

(1)OpenStack 卷界面如图3-55所示，OpenStack 创建卷如图3-56所示。

**openstack**

Preject

admn ·

**Volumes**

0ewire

hslancs

Valumes

mager

Actats &Seeuty

Natwork

**Volumes**

O

Dchp/tom



TeaValme

Sh

26B

Q

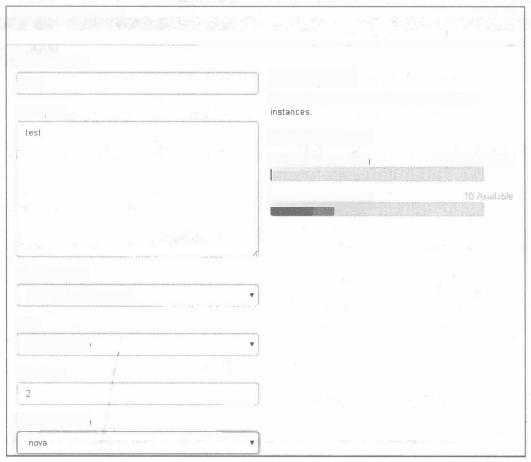
False

Emurptad

否

否

图3-55 OpenStack 卷界面



**Create Volume**

**Volume Name\***

**Description:**

Volumes are block devices that can be attached to

**Description**

**Volume Limits**

Total Gigabytes (7 GB) 1.000 G8 Arafsldle

Humber of Volumes Q)

**Volume Source**

No source,empty volume

**Type**

Nowolume type

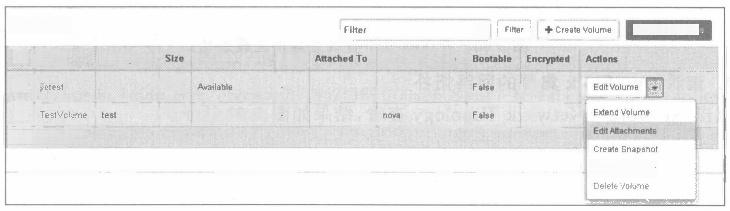
Size (GB)\*

Awallabillty Zont

test

图3-56 OpenStack 创建卷

(2)可将调整卷的大小、挂载卷到特定的主机，创建镜像、导出成镜像，如图3-57所示。



**Volumes**

□ Name

□

□

Displayng 2 lteme

Description Status Type

2GB

5GB 数 力 Attaching

\*Dalete volunss

Awailabillity Zone

否

否

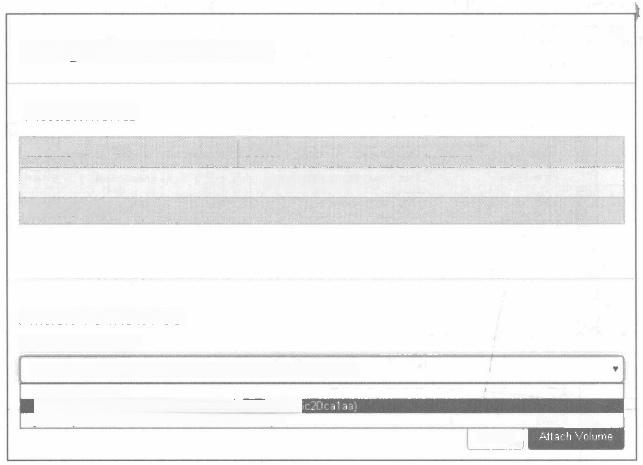
nova

Q

图3-57 OpenStack 卷操作

大数据技术及行业应用

(3)动态的挂载卷到主机，如图3-58所示。



×

**Manage Volume Attachments**

**Attachments**

**Instance**

No items to display.

Displaying O items

**Attach To Instance**

**Attach to Instance\*O**

testOpenstackVolume (353aea5a-2fa8-438-93ca-be36c20calaa)

Select an instance

tasl0rensiackVolrme E5ae:5:2AB4SR9Etthe

jieTest (1516ab20-1571-46249cOb-d1793104dcf)

Cancel

**Actions**

**Device**

图3-58 挂载卷到 testOpenStackVolume 虚拟机中

(4)成功将卷挂载到 testOpenStackVolume 虚拟机中，如图3-59所示。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Volumes** | | | | | | Filte Q | | | Fiiter | +Create Volume | | 8 Delete Voume | |
| □ Name | Descriptlon | Size | Status | Type | Attached To | | Availability Tang | Bootable | | Encrypted | Actlons | | |
| □ jietest |  | 2GB | In-Use |  | AHached to  testOpenstackVolume on dew'vdb | | nova | False | | 否 | Edit Volume | |  |

图3-59 卷挂载成功

(5)在虚拟机中查看，如图3-60所示。

|  |
| --- |
| $1s -al I grep vd  IrWXTWXTX 1 root root 4 Jul 920:52 root ->vdal  brw------- 1 root root 253, 0 Jul 920:52 vda  brw------- 1 root root 253, 1 Ju1920:52 ydal  brw------- 1 root root 253, 16 Ju1 921:01 vdb  $■ |

图3-60 在 Xshell 中查看卷是否挂载成功

**4.** **查看** **OpenStack** **集群的网络拓扑**

点击 Network→Network Topology 查看，结果如图3-61所示。



<12.16.0.0>/24

<192.168.111.0>/2

**Compute**

**Network**

Network Topology

Networks

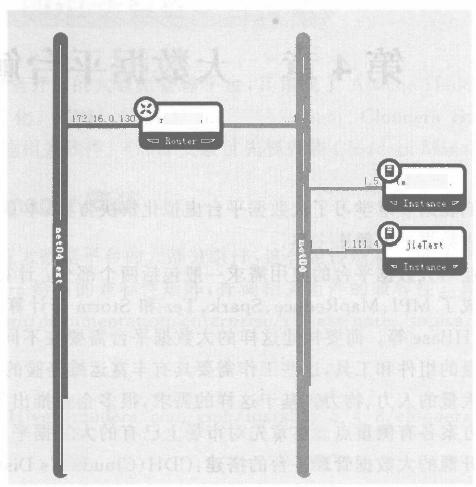
Routers

**Orchestration**

Admin

Identity

Small Normal

outer04 192.168111

92,168.111 xtOpens.,

92 188.

图3-61 集群网路拓扑



**本** **章** **小** **结**

当前工业界普遍将 Docker 和 OpenStack 结合使用，成功案例有蘑菇街、京东等。也有 单独使用 Docker 的如雪球网、灵雀云、DaoCloud 等 。Docker 和 OpenStack 能够提升对生 产环境中大规模集群的部署和维护效率，降低人力物力成本，是大数据平台的基础环境。

**参** **考** **文** **献**

[1] <https://docs.docker.com>.

[2]<http://yaxin-cn.github.io/Docker/docker-container-use-static-IP.html>.

[3]<http://www.cnblogs.com/rilley/archive/2012/02/13/2349858.html>.

[4]<http://www.cnblogs.com/rilley/archive/2012/02/13/2349858.html>.

[5]<http://yuedu.baidu.com/ebook/d817967416fc700abb68fcal?fr=aladdin&.key=docker>.

**第** **4** **章** **大数据平台解决方案**

我们在第3章学习了大数据平台虚拟化解决方案，本章将介绍适合企业应用的生产环 境下的大数据平台解决方案。

企业对大数据平台的应用需求一般包括两个部分：计算和存储。当前主流的计算平台 主要集成了 MPI 、MapReduce 、Spark 、Tez 和 Storm 等计算架构。数据存储集成了 HDFS 、 Impala 、HBase 等。而要构建这样的大数据平台需要在不同的物理和虚拟计算机上安装和 部署大量的组件和工具，这些工作需要具有丰富运维经验的系统管理人员才能完成，并且需 要耗费大量的人力、物力。基于这样的需求，很多企业推出了自己的大数据平台解决方案， 而这些方案各有侧重点。本章先对市场上已有的大数据平台进行分析和对比，然后重点介 绍两个开源的大数据管理平台的搭建：CDH(Cloudera's Distribution Including Apache Ha- doop) 和 HDP(Hortonworks Data Platform)。



**4.1** **大数据平台比较**

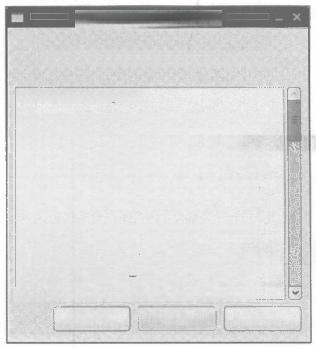
当前主流的大数据平台提供机构包括 Cloudera 、Hortonworks 、MapR、 华为、EMC、 IBM 、Intel等，这些机构有公益的、商业的或者混合的多种形式，提供的大数据平台有开源 的、商业的等多种版本，每个机构提供的大数据平台使用的用户场景各不相同，因而进行平 台选择的时候，用户必须根据自己的需求进行选择。下面我们通过表格的形式，将各个机构 提供的平台的特点列出，如表4-1所示。

**表4-1** **主流大数据平台对比**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 机构名称 | 平台名称 | 是否开源 | 商业版本 | 定价 | 安装管理系统 |
| Cloudera | CDH | 是 | 有 | 4000美元/年/节点 | Cloudera Manager |
| Hortonworks | HDP | 是 | 有 | 12500美元/年/10节点 | Ambari |
| 华为 | Fusionlnsight | 否 | 有 | 不详 | 不详 |
| EMC | Pivotal HD | 部分 | 有 | 不详 | Ambari |
| IBM | InfoSphere Biglnsights | 否 | 有 | 不详 | 不详 |

Hadoop 的发行版除了 Apache 官方的 Apache Hadoop 外 ，Cloudera 、Hortonworks、

MapR、EMC 、IBM 、Intel 和华为等都提供了基于 Apache Hadoop的社区版、试用版和商业 版等。商业版与社区版或试用版的区别是提供了专业的技术支持，这对于从事大数据业务 的企业非常重要。



**cloudera Manager 5**

**Cloudera Manager README**

Cloudera Manager 5

The Cloudera Manager Installer enables you to install Cloudera Manager and

bootstrap an entire CDH cluster,requiring only that you have SSH access to

your cluster's machines,and that those machines have Intemet access.

This installer is only recommended for

demonstration and proof of concept

deployments,but is not recommended for production deployments because it is

not intended to scale and mav reauire

Cancel

Back

Next



**4.2 CDH 大数据平台搭建**

CDH 是 Cloudera 开发的完全开源的大数据管理平台，其集成了 Apache Hadoop 和 一 些 专 门 为 企 业 需 求 所 做 的 优 化 。CDH 包 括 Cloudera Manager 、Cloudera Standard、 Cloudera Enterprise Trial 和其他相关软件。CDH 安装首先要安装 Cloudera Manager。

**4.2.1** **Cloudera Manager 安装**

Cloudera Manager 支持主流大数据平台的大部分组件，这些组件随着 CM 不同版本推 出而更新，用户可以根据Cloudera 给出的兼容性矩阵，查询相关组件的兼容性。具体网址 参见：<http://www.cloudera.com/documentation/enterprise/release-notes/topics/Product>

\_Compatibility\_Matrix.html。

**1.** **下载** **CM** **安装包**

运行命令：wget <http://arcHive.cloudera.com/cm5/installer/5.4.8/cloudera-manag>-

er-installer.bin,

**2.** **运行安装** **CM**

(1)修改权限

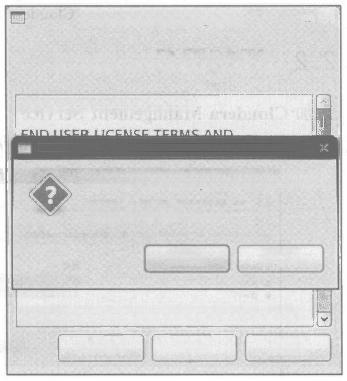
chmod u+x cloudera-manager -installer.bin

(2)执行 cloudera-manager-installer.bin

./cloudera-manager-installer.bin

(3)开始安装 CM 许可认证，点击“是”即可，如图4-1所示。

Cloudera Manager 许可证界面如图4-2所示，选择“是(Y)” 后继续。然后出现 Oracle 二进制代码许可证界面，如图4-3所示，选择 Next 继续。



Cloudera Manager 5 一 x

**Cloudera Express License**

Cloudera Express License

**Cloudera Express License**

Accept this license?

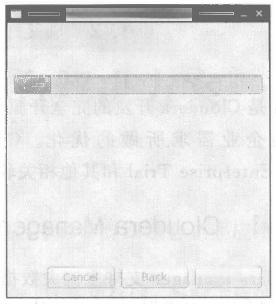
否(N) 是(Y)

IF YOU ("YOU"OR"CUSTOMER")PLAN TO

Cancel Back Next

图 4 - 1 CM Readme 界 面 图4-2 CMLicense界面





**Cloudera Manager 5**

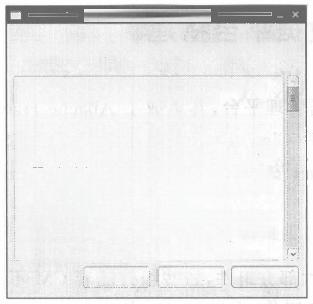
**Installing**

A

Next

大数据技术及行业应用

继续选择 Next, 开始安装 Cloudera Manager 5,界面如图4-4所示。



**Cloudera Manager 5**

**Oracle Binary Code License Agreement**

Oracle Binary Code License Agreement for the Java SE Platform Products and JavaFX

ORACLE AMERICA,INC.("ORACLE"),FOR AND

ON BEHALF OF ITSELF AND ITS SUBSIDIARIES

AND AFFILIATES UNDER COMMON CONTROL,IS WILLING TO UCENSE THE SOFTWARE TO YOU

ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS BINARY CODE UICENSE AGREEMENT AND

SUPPLEMENTAL UICENSE TERMS (COLLECTIVELY "AGREEMENT").PLEASE READ THE AGREEMENT CAREFULY.BY SELECTING THE "ACCEPT

ICENSE AGREEMENT IOR THE EOIIMALFMTL

Cancel

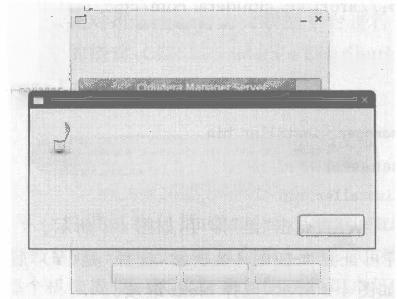
Back

Next

图4- 3 Oracle 二进制代码许可证界面 图 4 - 4 Cloudera Manager 5安装界面

安装完成后，提示打开浏览器访问 CM5 的管理界面，用户名和密码都是“admin”, 如

图4-5所示。至此安装成功，打开浏览器输入http://localhost:7180 即可进入Web 界面。



opt]#

Cloudera Manager

**Starting**

8-1.cdh5.

8-1.cdh5.

iste

**Next step**

Your browser should

localhost:7180/.Log in

usemame:'admin'and

installation.

关 闭(C)

Back

cance!

now open to http://

to Cloudera Manager password:'admin'to

ister

ind

ster ster

ppt]

pkgl

continue

x64.tar.gz



with

Next

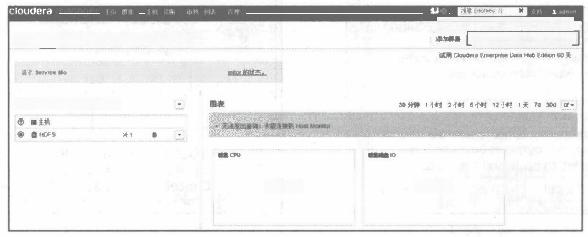
5

图4-5 Cloudera Manager 5 提示界面

**4.2.2** **添加服务**

**1.** **添加** **Cloudera Management Service**

点击右上角添加 CM Service,如图4-6 所示。



0列样平验限HHDE 下 在RCZ 属

*Chnonn ureyentnbante*

nr ,建 网 需 过 源 接 。 宣enkeh

**●Clustwr 1** coHsAA.PiR

查谢错调

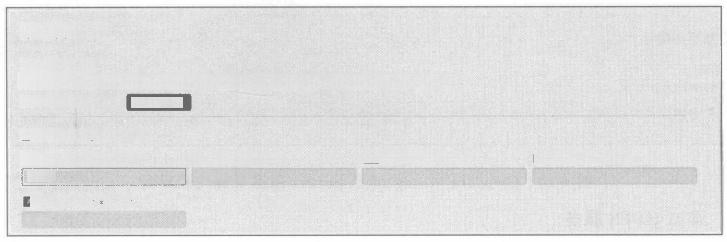
**主** **页** **有深行状况问满配理** **-降有最播命专**

查询错调

图4-6 添 加 Cloudera Management Service 界 面

第4章 大数据平台解决方案

选择 master 作为 Service Monitor 和 Host Monitor,如图4-7所示。



**添加服务向导**

**自定义角色分配**

您可以在此处自定又新服务的确色分配，但请注趣，如果分配不正确(例如，分配到某个主机上的确色大多),性能受到能响。 还可以按主机查看角色分配。 #接于机查着

cleucera Management Sersce

servke Monter x1新建

master

Aert Pubasher 1新建

master

Event Server x1新建

选择一个主机

x 1新建

h Montor

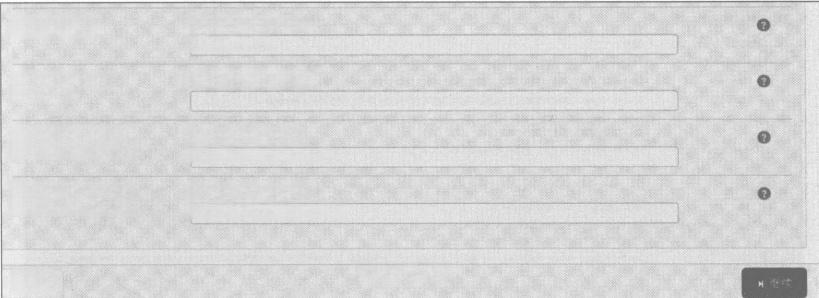
HMontor

master

master

**图4-7** **添加服务向导界面**

**目录的存储配置如图4-8所示。**

**警报：邮件收件人**

**Custom Alert Senpt**

alert scnpt path

**Host Monitor 存铺日录**

frehose storage base direclory

**sarvice Monitor 存前目录**

trehose storage base directory

Aert Publisher Detauft Group

root@locahost

Nat Pudlshar Detautt Goup

HostMonitor Defaut Group

var/b/cloudera-host-monltor

Sarvce Monitor Detautt Group

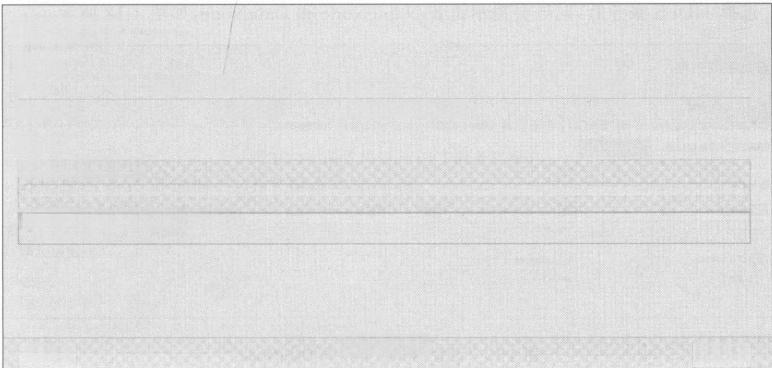
var/lb/cloudera-servce-montor

020**0Bd**

N 返 回

图4-8 目录的存储配置界面

启动服务如图4-9所示。



**添加服务向导**

进度：

命令

2015-11-1716:19.09 CST

**命令进度**

已完球0个步骤(共1个)。

**正在启动Cloudarn** **Managamant** **Sarvies服务**

**详级信息。**

0**D90DD**

N 返 回

**首次运行**

**开给日期**

正在进行

N 继 续

结束于

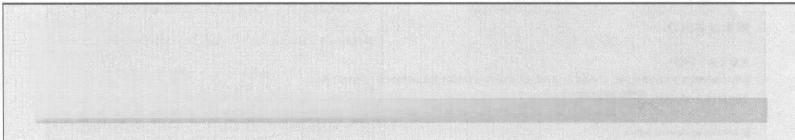
上下文

状态

图4-9 启动Cloudera Management Service 服务界面

大数据技术及行业应用

安装完成界面，如图4-10所示。



**添加服务向导**

**恭喜您!**

**意的新服务已安装在群集中并已配图。**

注意：定可能仍必须息动新服务，建议意在执行此接作之期，先重息所有配图过时的依最关系限务。可以通过单击下面的究减在主项面上执行边业接作。

图4-10

**2.** **添加** **HDFS 服务**

“Cloudera Management

Cloudera Management Service 服务安装完成界面

Service 服务”添加成功后，可以通过“添加服务向导”添加其他

服务。如想添加 HDFS 服务，请选择 HDFS 服务，如图4-11所示。



明

The Apache Accumulo sorted,distrbuted key/value store is a robust,scalable,high pertormance data storage and retrieval system. Flume 从几平所有来源收集数据并将这些数据频合到永久性存储(如HDFS) 中 。

Apache HBase 提供对大型数据集的随机，实时的读写访问权限(需要 HDFS 和 Zookeeper)。

Apache Hadoop 分布式文件系统(HDFS) 是Hadoop 应用屋序快用的主要存储系统。HDFS 创建多个数据块副本并将它们分布在整个群集的计算主机 上，以启用可靠且极其快速的计算功能。

Hihe是一种数据仓库系统，提供名为 HveOL 的 SO 类语言。

Hue 是与包括Apache Hadoop的 Cloudera Dlstrbuton一起配合使用的图形用户界面(需要 HDFS,MapReduce 和Hve)。

lmpala 为存储在 HOFS和 HBase中的数据提供了一个实时 SOL 查询接口。Impala 需要 Hiwe服务，并与Hue 共享 Hve Metastore。 EMC isilon is a dstritbuted nlesystem.

The Hadoop Key Manapement Service wth fle-based Java KeyStore.Malntains a single copy of keys,using simple password-based proleclion. Requires CDH 5.3+.Not recommended for produchion use.

Apache Katta is pubish-subscribe messaging rethought as a distrtbuied commt log Before adaing this servce,ensure thal the Kafta partelis installed

读值 Store Indexer 侦听 HBase 中所合表内的数据变化，并使用Sotr 为其创建素引。

本社外融A 解作内的4删的好集单仁八十十山智/单黑

**0280200**

群 续

**添加服务向导**

**选择您要添加的服务类型。**

服务类型

● a Accumulo 1.6

● Fume

● H HBase 图 HOFS

0 Hve

● Hue

● Y impaa

● isilon

● Java keyStore kMs

0 Kaka

0 Key-Value Store Indexer

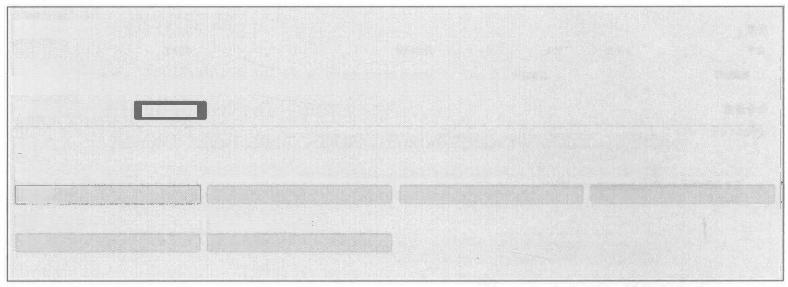


M 返 回

t

图4-11 添加 HDFS 服务

选择 HDFS 服务后，向导会提示设置 NameNode 和 DataNode, 如图4- 12所示。



**添加服务向导**

**自定义角色分配**

您可以在此处自定义新服务的角色分配，但请注量，如果分配不正确《例如，分配到某个主机上的角色太多),性能受到影响。

还可以挖主机查看角色分配。 #报主机或看

HDFS

secondanpNameNote x 1新建

master

[W DataNode x 2新建

slavel1-2]~

NameNode x 1新建 masie

W NFS Gateway

选择主机

mpfs 选择主机

eatancer x 1新建

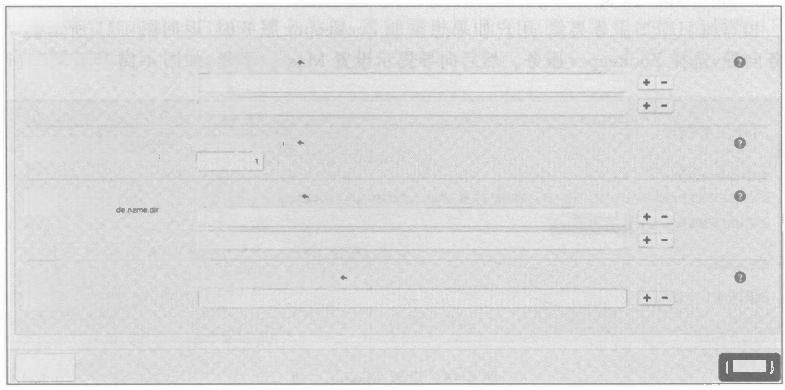
masler

图4-12 设置 NameNode 和 DataNode

设置 NameNode 和 DataNode 后，向导提示设置 DataNode 数据目录和 NameNode 数

据目录，如图4-13所示。

第4章 大数据平台解决方案



**审核更改**

DataNode Detault Group

/dts/dn

lopuatvan

DataNode Datault Group

NameNode Dutait Oroup

iatsnn

loptdtsmn

SacsndanNamaNose Dutault

t.checkpolntar.

/ats/snn

M 返回

**接受的** **DataNode 失败的卷**

dts datanode talled volumes toieratad

**DataNode** **数据目录**

dn dat dir,dh datanode dat dir

**NamaNode 数据目录**

dh names,dts namenod

dts namenode theckpoint air

**HDFS** **检查点目录**

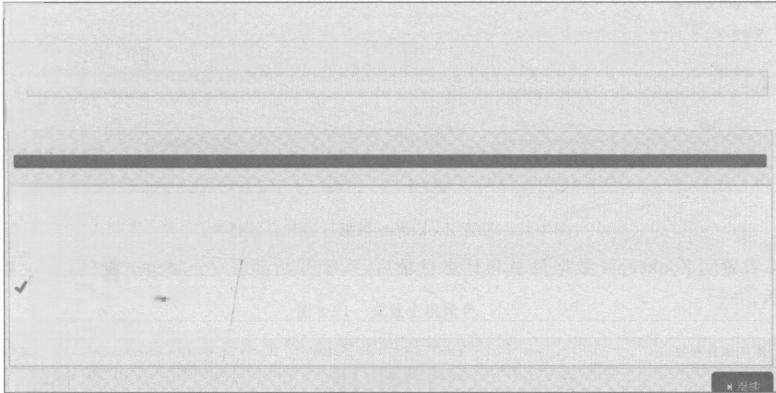
0200000

M维续

roup

图4-13 设置 DataNode数据目录和 NameNode 数据目录

上述设置完成后，系统开始部署服务，如图4-14所示。



进度

命令

√ 首次运行

Findshed First un of all servlces successfully.

**命令进度**

已完成4个步壁(共4个)。

√ 正在部著客户演配用

Successfully deployed all client conflgurations.

详细信息

√正在检查NameNode的名称目录是否为空。仅在为空时格式化HOFS。

Successfully foraatted naselode.

详细信息

正在启动HDFS 服务

Successfully started HoFS dce

连细信息

√ 正 在 创 建 HDFS Amp目录

successfully created wors directory /tap.

详细信息

N返回 **0000000**

结束于

2015-11-1717:48.49 CST

开始日期

2015-11-1717:47.40 CST

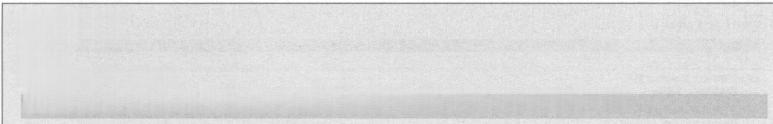
已完成

上下文

状态

图4-14 系统部署 HDFS 服务界面

当出现图4- 15界面时，HDFS 服务部署完成。



**添加服务向导**

**恭** **喜** **您** **!**

您的新服务已安装在群集中并已配图。

注益：您可能仍必须启动断服务。建议意在执行此操作之前，先重启所有配置对时的作赖关系服务。可以通过单击下面的完减在主而面上执行造些接作，

图4-15 HDFS 服务部署完成

大数据技术及行业应用

**3.Zookeeper 安装**

和添加 HDFS 服务类似，用户如果想添加 Zookeeper 服务时，返回图4-11 所示的添加 服务向导，选择 Zookeeper 服务。然后向导提示设置 Master 主机，如图4-16所示。



**添加服务向导**

**自定义角色分配**

您可以在此处自定义新服务的角色分配，但请注意，如果分配不正确(例如，分配到某个主机上的角色太多),性能受到影响。

还可以按主机查看角色分配。 出按主机查看

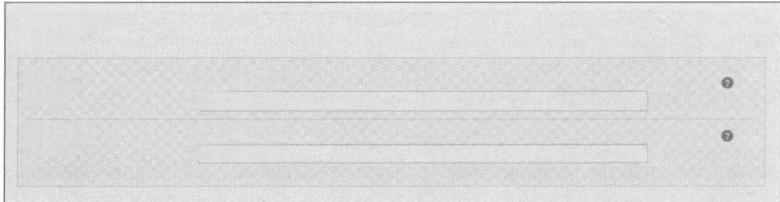
**ZooKeeper**

Sserver x 1新建

master

图4-16 设置 Master 主机

设置完 Master 主机后，系统提示设置 Zookeeper 数据目录和日志目录，如图4-17 所示。

**添加服务向导**

**审** **核** **更** **改**

数据目录

Server Detaut Goup

Nar/Mib/zookeeper

Sarver Datut Goup ar/mtv/zookeeper

dataOr

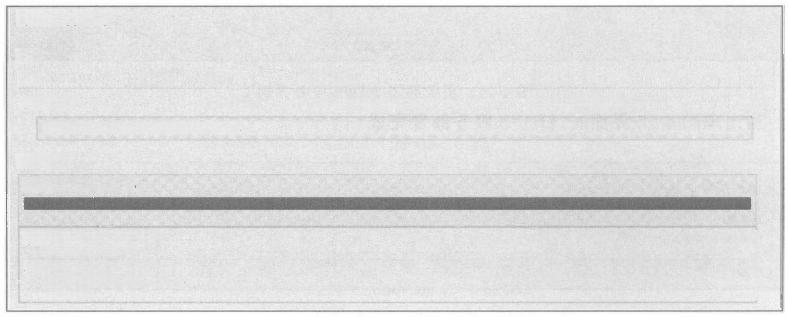
事务日志目景

duatenDir

图4-17 设置 Zookeeper 数据目录和日志目录

设置完 Zookeeper 数据目录和日志目录后，系统开始部署 Zookeeper 服务，如图4-18

所示。



**添加服务向导**

进度

命令

√ 首次运行

PInlahed Pirit Bun of all

**命令进度**

已充或2个步理(共2个)。

√ 正在初始化ZooKeeper 腺务 已成功完成1个步键。

√ 正在启动 Zookeper 服务

Cmpleted 1 steps successfully. 详细信息

结束于

2015-11-1717.5240 CST

**开始日期**

2015-11-1717.5216 CST

上下文 状态

serdces successfully.

已完或

图4-18 系统部署 Zookeeper 服务

第4章 大数据平台解决方案

**4.YARN 安装**

YARN 安装和添加 HDFS 服务类似，返回图4-11所示的添加服务向导，选择 YARN 服务，如图4-19所示。

Apache Hadoop MapReduce 20(MRV2)或YARN 是支持 MapRetuce应用程序的数据计算框架(需要 HDFS)。

出 YARN (MR2 Included) Zookeeper

Apache Zookeeper是用于维护和同步配置数据的集中服务。

**0000000**

W 返回

图4-19 选择 YARN 服务

然后向导提示设置依赖关系，如图4-20所示。

**添加服务向导**

**为新服务选择一组依赖关系**

0 HDFS

Zookeapar

图 HDFS

Zookeeper

●HDFS

图4-20 设置 YARN 依赖关系

设置依赖关系后进行主机配置，如图4-21所示。用户根据需求，设置主机的角色。



**添加服务向导**

**自定义角色分配**

您可以在此处自定义新服务的角色分配，但请注意，如果分配不正确(例如，分配到某个主机上的角色太多),性能受到影响。

还可以按主机查看角色分配。 按王机查看

YARN(MR2 Included)

HS JobHistory Server x 1新建

master

RW ResourceManager

master

NodeManager

x 2新 建

slave[1-2]~

×1新建

图4-21 设置主机角色

设置主机完成后，选择安装路径，如图4-22所示。



添加服务向导

审核更改

NodaManager 本地目录

yam nodamanager lseatdin

wammm

/opUyammm

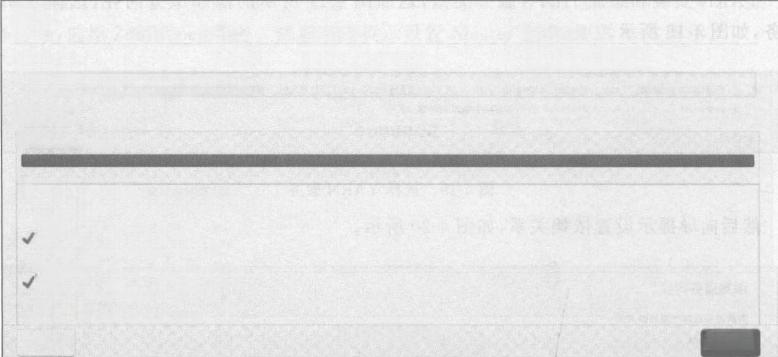
NodaMarapsrDetult Group

O

图4-22 安装路径选择

大数据技术及行业应用

上述步骤设置完成后，系统开始部署并启动，如图4-23和图4-24所示。



进度

命 令

**√** **首** **次** **运** **行**

Hlslsted Firat Bun af all services successtfully.

**命令进度**

已完成4个步程(共4个)。

在部答客户演配置

Successfully depleyed all client canfigurations.

连 组 信 息

正在创建 MR2 作业历史记录目录

Suecessfully created Hors directory.

详 细 信 息

正在创建 NodetManager远程应用程序日志目录 Successfully created hors directory.

详细信息

rrezh vanu nins iarhudsn田R

返 回

结束于

2015-11-1718.01.00 CST

开始日期

2015-11-171759:00

**020000**

继读

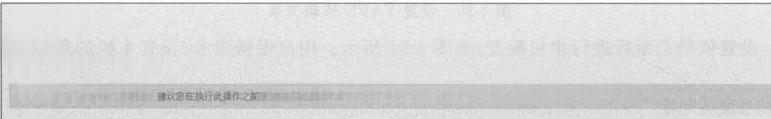
已完或

上下文

状态

CST

图4-23 YARN 部署界面



**添加服务向导**

**参喜您**

**您的新服务已安装在群集中并已配置。**

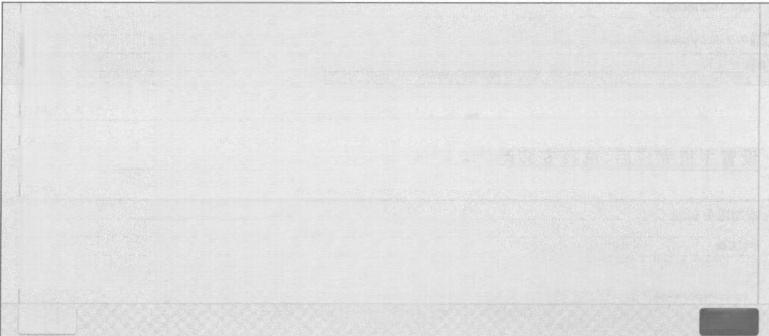
注意：总可能必须启动新服舟，

先重启所有面 过时的体额关系服务。可以通过单击下面的完成在主贡面上执行这些操作。

**图4-24** **YARN 服务安装成功界面**

**5.Hive 安装**

**添加** **Hive 服务和添加** **HDFS 服务类似，如图4-25所示的添加服务向导，选择** **Hive 服务。**



上，以启用可童且极其快速的计算功能。

Hve 是一种数据仓库系统，提供名为 HveOL 的 SQL 类语言。

Hue 是与包括Apache Hadoop的 Cloudera Dlstrtbuton一起配含使用的图形用户界面(需要 HDFS、MapRecuce 和 Hwe)。

impala 为存储在 HDFS和 HBase 中的数据提供了一个实时 SOL 查询接口。tmpaa 需要Hve 服务，并与Hue 共享 Hve Metastore。 EMC islon is a distrbuted flesystem.

The Hadoop Key Management Servce wth Me-based Java KeyStore Mairtains a single copy of keys,using simpte password-based protecton. Requres CDH 5.3+.Not fecommended for produc ion use.

Apache Kafka t putash-sutbscrbe messaying rethought as a dntrbutet commit tog.Before adding this servce,ensure that the Karta parcel s nstaled

键 值 Store Indexer侦听 HBase 中所含表内的数据变化，并使用 Sor 为其创建素引。

Apache Hadoop MiapReduxe变持对整个群集中的大型数据集进行分布式计算(需要HDFS)。 建议改用YARN (包括MapRatucte 2)。包 括

MapReduce 用于向后兼容性。

Ooze 是群集中管理数据处理作业的工作液协调服务。

Sentry服务存情身份验证政策元数据并为客户端提供对请元数据的并发安全访间。

Sor 是一个分布式服务，用于编制存情在HDFS 中的数据的索引并搜索这些数据。

Apacthe Spark s an open source custer compuling system.This servce runs Spark as an applcation an YARN

Apache Spark is an open source cduster computing system.This is the standalone version af the servce which does not use YARN tor resouice management.Cloudera recommends using Spark an YARN instead of this standalone version.

**0000000**

返 回

图 Hve

● Hue

0 Y impae

○ 5 lsion

● Java KeyStore ks

0 Kata

 Key-Vaue Store Indexer

Q ■ MapReduce

0 0oze

0 Sentry

● Sol

0 Spark

● Spark (Standalone)

继续

图4-25 添加 Hive服务界面

选择 Hive 服务的依赖关系，如图4-26所示。

选择 Hive 服务中主机的角色，如图4-27所示。

选择 Hive 服务的数据库，可以选择已经安装好的 MySQL, 设置连接数据的主机名、用 户名和密码，如图4-28所示。 Hive 支持多种数据连接方式，具体可到其官网查询。

<< 70

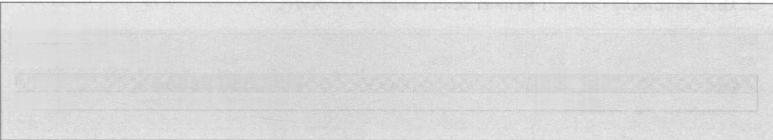
第4章 大数据平台解决方案

**添加服务向导**

**自定义角色分配**

您可以在此处自定义新服务的角色分配，但请注意，如果分配不正确(例如，分配到某个主机上的角色太多),性能受到影响。

还可以按主机查看角色分配。 田

**添加服务向导**

**为新服务选择** **一** **组依赖关系**

**HDFS**

**d ZooKeaper**

Zookeeper

YARN (MR2 Imcluded)

YARN (MR2 Inctuded)

●HDFS

**图4-26** **选择** **Hive 服务的依赖关系**



Hve

0Gateway x 3新建

master stawe[1-2]

mus Hve Metastore Server x 1新建

master

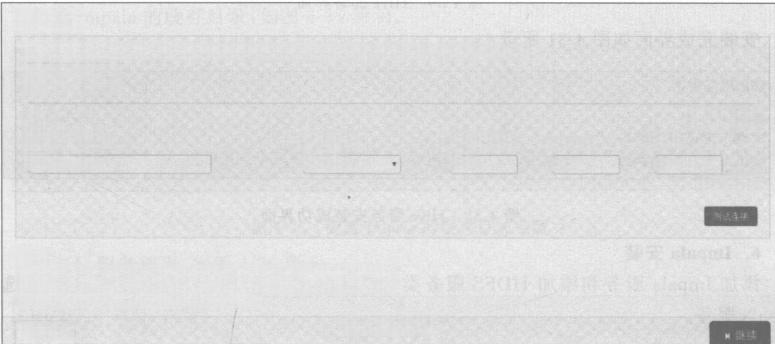
MR HNeServer2 x 1新建

master

Wic WebHCat Server

选择主机

图4-27 选择 Hive 服务中主机的角色界面



**数据库设置**

配度和测试数据库连接。首先根捉Instala tlon Cukte e的Installing and Configuring an Extemal Database小节创建数据库。

●使用自定义数据库

●使用嵌入式激据库

**Hive**

**数据库主机名除** **·**

locathost 3306

显示密码

备注：

·创建数据库时，数据库主机名称字段中的值必领与您用于主机名称的值匹配。了解更备

·如数据库末在其财认施口运行，请使用数据库主机名称字段中的 hostport指定端口号。

·强烈建议将各个数据库与相应角色实例置于同一主机上。

M 返 回

数握库名称：

hve

数报库类型

MySOL

用户名：

root

密码：

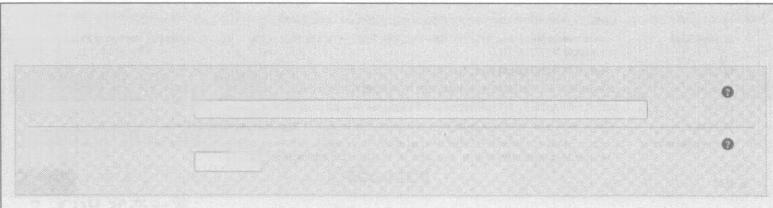
lale

**0000000**

VSueeesstu

图4-28 Hive 设置数据连接界面

设置 Hive 仓库的路径如图4 - 29所示。

**添加服务向导**

**审核更改**

Hive 仓库目景

hve matastore warnhouse dir

Hive Metastore服 务 器 口

hve metastore part

Hhe (服务范围)

lusernlverwarehouse

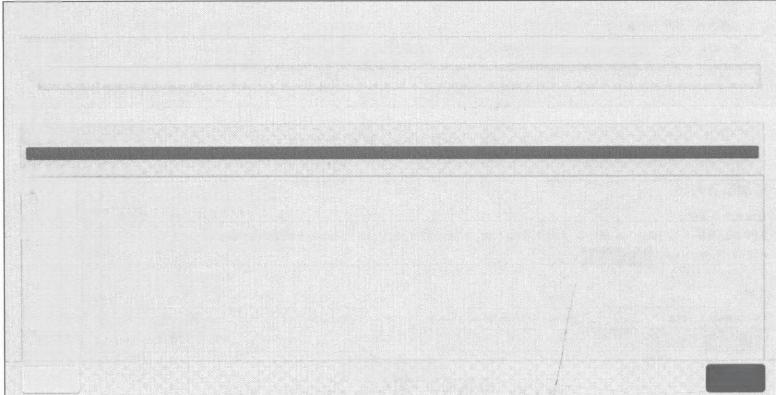
Hre Metastere Server Datut Oroup

9083

图4-29 设置 Hive 仓库的路径界面

大数据技术及行业应用

**上述步骤完成后，系统开始部署安装，如图4-30所示。**



进度

命 令

**√** **首** **次** **运** **行**

Pintshed Flirst Run ef al1 servlces successfully.

**命** **令** **进** **度**

已完成5个步疆(共5个)。

**正在部育客户配**

deployed alI cllent

Metastore料智库来

Created Hlive Hetastore Database Tables successfully.

详 细 信 息

√ 正在创建 Hve 用户目录

Succestfully creatad Hors dlrectary.

连 细 信 息

√ 正在创遽 Hwe 仓库目录

Successfully created hors directory.

Te编2 阳

**000000D**

N 返 园

结束于

2015-11-1718.42:30 CST

**开始日期**

2015-11-1718.40.38 CST

successfully 详细信息#

正在创建Hwe

contigurattong.

已完成

上下文

状 态

继 续

图4-30 Hive 部署界面

**安装完成界面如图4-31所示。**



**添加服务向导**

**恭** **喜** **您** **!**

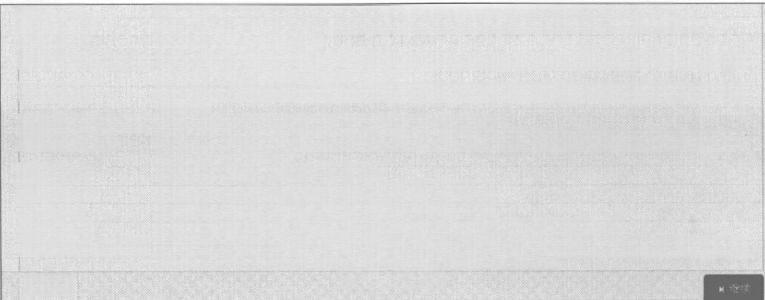
**您的新服务已安装在群集中并已配盟。**

注意；您可能仍必须启动新服务。建议您在执行此操作之前，先重君所有配照过时的依赖关系服务。可以通过单击下面的完减在主页面上执行这些操作。

**图4-31** **Hive 服务安装成功界面**

**6.Impala 安装**

添加 Impala 服务和添加 HDFS 服务类似，返回如图4-32 所示的添加服务向导，选择 Impala 服务。



Hue Hue 是写直括 Apache Hadoop的 Cbudera Dsrbulion一E 配含使用的图形用产界面(需费HOFS,MapAeduce 和 Hve),

9 impala Impala 为存储在 HDFS 和 HBase 中的数据拼供了一个实时 SOL 查询接口。lmpala 需要Hve 服务，并与Hue 共 享 Hve Melastore。

4 Hsllon EMC tsikon is a distnbutedflesystem.

|  |  |
| --- | --- |
| Java keyStore KMS  kaka | The Hadoop Key Management Servke wth Me-based Java KeyStore Mantains a single capy of keys.using simple password-based protection Requres CDH 5.3+.Nol tecommended for productton use.  Apache Katta is putlsh-sutscrbe messaying rethougnt as a distrtbuted commt iog Before adting this servce,ensure that the Kalta parcel is installed. |

将 Key-Value Store Indexer 健/信Sore iIndexer 们听 HBase 中所含表内的数据变化，并使用 Solr 为其创建索引。

图 MapReduce

MapReduce 用于向后兼容性。

Ooze 是群集中管理数据处理作业的工作流协调服务。

Sentry服务存情身份验证政策元数据并为客户端提供对请元数据的并发安全访问。

Sot 是一个分布式服务，用于编制存倘在 HDFS 中的数据的素引并搜素这些数据。

Apache Spark is an open source cluster computng system.This servce runs Spark as an appltcation an YARN

Apache Spark is an open source ctuster computng system.This is the standalone versian of the servce wnich does not use YARN tor fesource management.Cloudera recommends ustng Spark on YARN instead of this standalone verston.

**0200030**

返 回

00oze

Sentry

Sor

spark

Spark (Standalone)

Apache Hadoop MapFRetue支持对整个群集中的大型姓据集进行分布式计算(需要HOFS)。 建议改用YARN《 包括 MapReduce 2)。包括

◎ 图

●

○

●

●

●

e

●

0

0

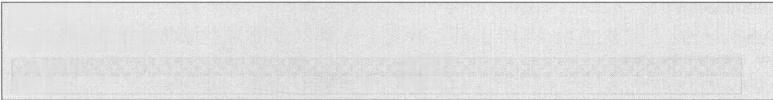
0

M

图4-32 添加 Impala 服务界面

<< 72

设置 Impala 服务依赖关系，如图4-33所示。

添加服务向导

为新服务选择 一 组依赖关系

0 HDFS

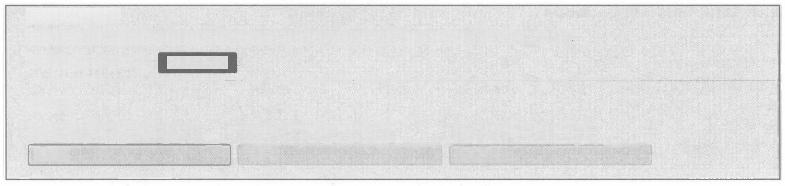
Hve

HDFS

Hve

图4-33 设 置 Impala 服务的依赖关系

配置Impala 的主机角色，如图4-34所示。

自 定 义 角 色 分 配

您可以在此处自定义新服务的角色分配，但请注意，如果分配不正确《例如，分配到某个主机上的角色太多),性能受到影响。

还可以按主机查看角色分配。 按主机查者

4 impala

impala Catalog Server x 1新 建

lmpala Statesltore x 1 新 津

0 impala

slive[1-2]

Daemon x 2新建

master

master

图4-34 配 置 Impala 的主机角色界面

配置 Impala 的缓存目录，如图4-35所示。

**审核更改**

Impala Daamon 著存日录

ratch\_din

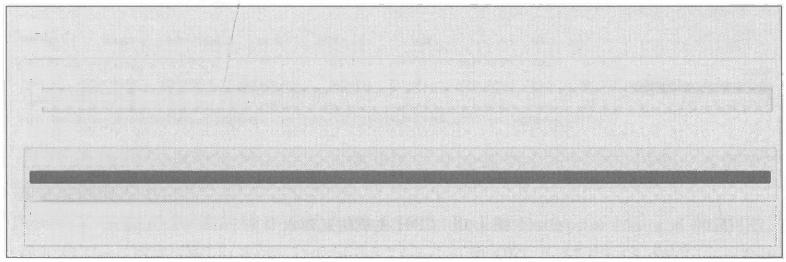
mpats Daemon Detault Orup

/mpslampatad

foptimpala/mpatad

图4-35 配 置Impala 的缓存目录界面

Impala 服务部署，如图4-36所示。



**进** **度**

**命令**

**√** **首次运行**

indahad

**命令进度**

已完成1个步骤(共1个)。

√ 正 在 创 建impae 用户目录

suecessfully created nors directory. 详给信息

结束于

2015-11-1718:49:01 CST

**开始日期**

2015-11-171848.29 CST

Ftrnt hun of all strvlcn

saccanfully.

已完成

上下文

状态

图4-36 Impala 服务部署界面

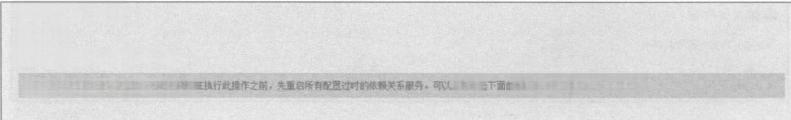
Impala 服务安装成功，如图4-37所示。

**7.CDH 状态一览**

上述配置完成后， 一个生产环境需要的大数据平台基本部署完成，包括 HDFS 、YARN 、 Zookeeper 、Hive 和 Impala 等，如果用户还需要其他服务如 Solr 、Spark 等可以在安装向导

大数据技术及行业应用

中根据提示进行安装。 CDH 对所有安装的服务都有状态报表，如图4- 38所示。

**添加服务同导**

**恭喜您1**

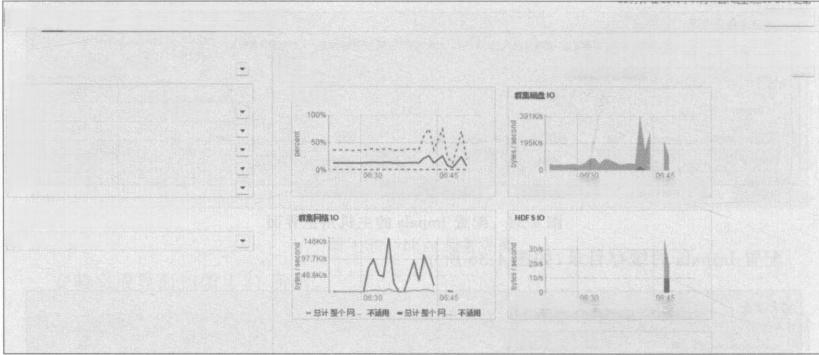
您的新服务已安装在群集中并已观图。

注意：您可能仍必领启动新服丹。建议您

完成在主页面上执行这些操作，

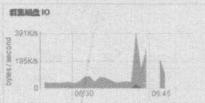
过单

**图4-37** **Impala 服务安装成功**



潭加解集

试 用 Coudera Enterprse Data Hub Edmon 60天





一总计要个D 不适用一总计整个D 不 适 用

已完成的p

**所** **有** **运** **行** **状** **况** **问** **通** **配** **营** **法** **3** **▼** **所有最浙命令**

O Cluster 1 (CDH53.8.Pacen

O ■王 机 \*3

O QHOFS \*1

○ He

O Y impata

O YAFN (MR2 inc

O Zookeeper \*1

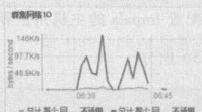
**Cloudera Management Service**

O Coudea Mans

**图** **表**

塞 cPU

=Custer 1.整个主机中的主机CFU 使用率25.6%



3 0 分 钟 1 个 时 2 小 时 6 小 时 1 2 小 时 1 天 7d 300 C

总计整个磁 本 话 用 一 总 计 基 个 碳 不 适 用

**主页**

c

图4-38 CDH 服务状态报表

CDH 主机信息列表，如图4- 39所示。



**主** **机** **获** **态** **配** **置** **模** **板** **根盘述** **Pareal**

**状态(Cluster 1)**

辩选器

图 常

础查使用情况

1130g/n7a

√ 状 鑫

112018171708

11208/7L100

1 # 月

》 内换

》已解除测载

》上 一检测璃号

物 理 内 存

4108/15208

1308/15268

1308/15208

平均仍献

1.631.981.79

0.040.04001

<0.030.040.01>

角 色

>15Ro

>4Roe()

》4 Role(s)

上一检测的号

12.37%之前

14.37s 之前

<192.168.10.201>

<192.168.10.202>

192.168.10203

交换空间

081208

08/208

0B/208

向解集浮加顿主机 Host inspector (Cussen) 重并运行升级内周

master

g ve1

s2

◎状好

且示25 ·条自

■已选定的操作

列 9 已 选 定 ·

14.16%之前

□ 0

+ 名 称



志

O

图4-39 CDH 主机信息列表



**4.3** **HDP 大数据平台搭建**

Hortonworks Data Platform 是 基 于 集 中 式 体 系 架 构(YARN) 的 企 业 级 开 源 Apache

Hadoop 分 发 版 本 。HDP 解 决 静 态 数 据 的 全 部 需 求 ， 支 持 实 时 客 户 应 用 和 提 供 鲁 棒 性 分

析，加快决策和创新。HDP 平台集成了大部分关键的大数据处理组件，并且随着这些组件 的更新而更新，用户可以根据自己的需求，选择安装 HDP 的版本，详见图4-40。 HDP 版 本 和其集成的开源组件的版本对照表如图4-40所示。搭建 HDP 之前先部署 Ambari, 因 为 Ambari 可以方便 HDP 的自动化安装。

Hive

Tez

Solr

a05

sader

HBase

Phoenlx

Falcon

Atlas

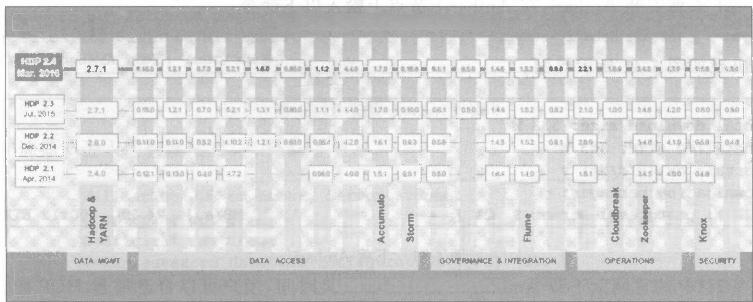
Kafka

doobg

Ambar

ooa

Ranger



Ongoing Innovation in Apache

&

HORTONWORKS DATAPLATFORM

图4-40 HDP 与其他开源项目的版本关系图

**4.3.1** **部署** **Ambari**

**1.Ambari 集群规划**

本节还是以最小的 Hadoop 集群架构来演示 Ambari 的部署。3个节点的主机名列表 如下：

(1)<192.168.10.101> Ambaril

(2)<192.168.10.102> masterl.hadoop

(3)<192.168.10.103> slavel.hadoop

Ambaril 是 http 服务器，masterl.hadoop 是集群的 Master,slavel.hadoop 是集群的

NameNode。

**2.** **下载** **Ambari**

Apache Ambari 是一个完全开源的管理平台，其具有配置、管理、监控和确保 Apache Hadoop 集群的功能。Apache Ambari 是 Hortonworks Data Platform(HDP)的一部分，允 许企业计划、安装和安全配置 HDP 平台，使其更容易地提供不间断的集群维护和管理，并 且不限制集群的大小。

Ambari 目前支持64位的操作系统包括：RHEL(Redhat Enterprise Linux)6 和RHEL 7、 CentOS 6 和 CentOS 7 、OEL(Oracle Enterprise Linux)6 和 OEL 7 、SLES(SuSE Linux Enter- prise Server)11 、Ubuntu 12 和 Ubuntu 14以及 Debian 7。本章我们选择 Ubuntu 14。

Ambari 的官方文档对安装环境提出了最小系统要求。这些要求包括硬件要求、操作系 统要求、浏览器要求、软件要求、数据库要求等。细节可以参照在网站 http://Ambari.a-

pache.org 上的官方文档。其中比较重要的就是软件要求，Ambari 要求 Hadoop 集群的每 个主机上必须安装 yum、rpm 、scp 、curl、wget、pdsh。这些软件都是在 Ambari 的安装脚本 中使用的工具。

Ambari 有2种安装方式：第一种是从公用资源库安装，第二种是源代码安装。第一种 简单方便，第二种因为以源代码编译安装，因而程序的执行效率会更高。

下面我们将介绍第一种安装方式。

**3.** **安装** **Ambari**

接下来就安装 Ambari, 在 Ambaril 节点上输入以下命令：

yum-y install Ambari-server

出现如图4-41所示界面。

**Total download size:333 M**

**Installed size:383 M**

**Is this ok [y/d/N]:y**

**Downloading packages:**

**4/4):postgresql-aerver-922%[==-** ] **20** **MB/s** **1** **74** **MB** **00:12** **ETA**

图4-41 下载 Ambari

在 HDP 的官网上下载 Ambari 的安装需要很长时间，用户可以自己搭建 HDP 的公共 资源服务器，因为如果是上百个节点的集群都到服务器上去下载，确实是一个很漫长的过 程，因而搭建一个本地的公共资源库确实很有必要，互联网上有很多配置本地资源库的文 档，大家可以上网去搜索，本节不作赘述。

下载完成后执行如下命令：Ambari-server setup。

当出现如图提示时，输入 y 继续，如图4-42所示。

|  |
| --- |
| **[root@localhost opt]#ambari-server setup**  Using python /usr/bin/python2.7  Setup ambari-server  Checking SELinux...  SELinux status is 'enabled'  SELinux mode is 'enforcing'  Temporarily disabling SELinux  **WARNING:SELinux is set to 'permissive'mode and temporarily disabled.**  OK to continue [y/n](y)? |

图4-42 安装 Ambari步骤1

因为我们已经安装了JDK 所以选3,继续安装，如图4-43所示。

|  |
| --- |
| Customize user account for ambari-server daemon [y/n](n)?  Adjusting ambari-server permissions and ownership...  Checking firewall status...  Redirecting to /bin/systemctl status iptables.service  Checking JDK...  [1]Oracle JDK 1.8 +Java Cryptography Extension(JCE)Policy Files 8 [2]Oracle JDK 1.7+Java Cryptography Extension(JCE)Policy Files 7 [3]Custom JDK |
| Enter choice(1): |

图4-43 安装 Ambari 步骤2

当出现如图4-43提示时，输入JAVA\_HOME 环境变量的路径，如图4-44所示。

Enter choice(1):3

HARNING:JDK must be installed on all hosts and JAVA\_HONE must be valid on all H psts.

HARNING:JCE Policy files are required for configuring Kerberos security.If you plan to use Kerberos,please make sure JCE Unlimited Strength Jurisdiction Polid r Files are valid on all hoats.

Path to JAVA HOME:/opt/jdk

图4-44 安装 Ambari步骤3

选择 y 进入数据库配置，如图4-45所示。

|  |
| --- |
| **Validating JDK on Ambari Server…done.**  Completing setup...  Configuring database …  **Enter advanced database configuration[y/n](n)?** |

图4-45 安装 Ambari步骤4

这里要选数据库，我们现在的系统里安装了 PostgreSQL, 所以我们选择1,如图4-46 所示。

|  |
| --- |
|  |
| Choose one of the following options:  [1]-PostgreSQL(Embedded)  [2]-Oracle  [3]-MySQL  [4]-PostgreSQL  [5]-Microsoft SQL Server(Tech Preview) |
| Enter choice(1): |

图4-46 安装 Ambari步骤5

图4-47是 Ambari 创建的数据库的名字、用户等。

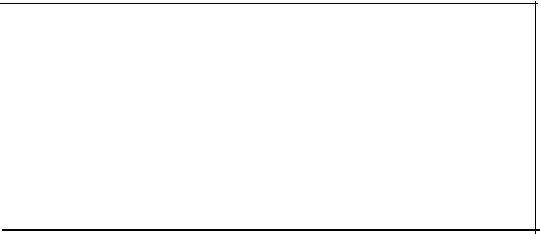
|  |
| --- |
| Enter choice(1):  Database name (ambari):  Postgres schema (ambari):  Username(ambari):  Enter Database Password(bigdata):  Default properties detected.Using built-in database. Configuring ambari database...  Checking PostgreSQL..  Running initdb:This may take upto a minute. |

图4-47 安装 Ambari步骤6

图4-48是安装 Ambari 成功界面，至此安装完成。

**4.** **启动** **Ambari**

启动 Ambari, 输入命令：Ambari-server start。



About to start PostgreSQL

Configuring local database..

Connecting to local database...done.

Configuring PostgreSQL.

Restarting PostgreSQL

Extracting system views...

ambari-admin-2.1.0.1470.jar

-\*=--

Adjusting ambari-server permissions and ownership... Ambari Server 'setup'completed successfully.

图4-48 安装 Ambari 成功

当出现如图4-49界面时，代表启动成功，

[root@localhost opt]#ambari-gerver start

Using python /usr/bin/python2.7

Starting ambari-server

Ambari Server running with administrator privileges.

Organizing resource files at /var/lib/ambari-server/resources. Server PID at:/var/run/ambari-server/ambari-server.pid

Server out at:/var/log/ambari-server/ambari-server.out

Server log at:/var/log/ambari-server/ambari-server.log

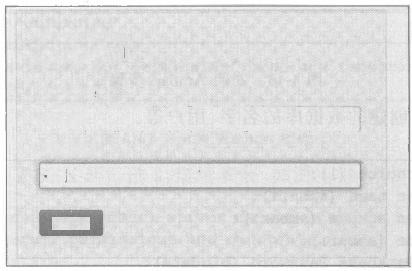
Waiting for server start … … ………

Ambari Server 'start'completed successfully.

图4-49 启动 Ambari

在其他计算机上用浏览器打开，输入 http://Ambaril:8080, 进入 Ambari Web登录界

面，如图4-50所示，默认的Username:admin,password:admin。



**Sign in**

Usemame

admin

Password

…

Sign in

图4-50 Ambari Web 登录界面

Ambari 搭建正式完成，如图4-51所示。接下来使用 Ambari 进行集群搭建。

**4.3.2** **用** **Ambari\_web** **部署** **HDP** **平台**

**1.** **集群规划配置**

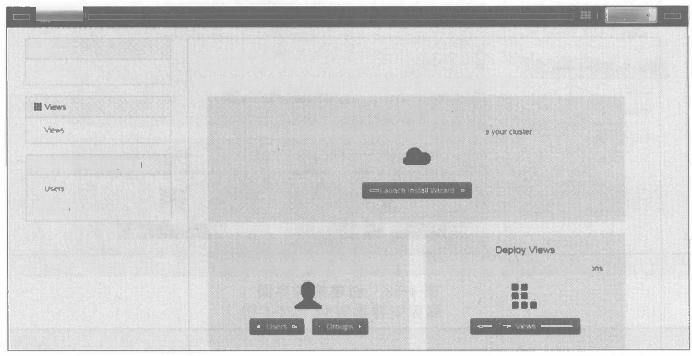
首先应规划好 HDP 集群：

(1)<192.168.10.104> master2

(2)<192.168.10.110> slave2-1.hadoop

(3)<192.168.10.111> slave2-2.hadoop

IP 的规划是101～109为master 保留，110～254是给 Slave 的。这里只做演示用，所以 只搭建一个 Master 和 2 个 Slave 节点。进行集群操作之前，要做ssh 无密钥登录配置，该配 置的详细步骤请自行百度，注意要将 Ambari 主机上的密钥复制到每个节点上。注意：复制 的是密钥，不是公钥。



▲ admh -

Chslen

Welcome to Apache Ambari

No chisteis

Provtion a ttuster,manage who can acces the ciuser,and cusomize veas tor Amban userns.

**Create a Cluster**

Uhe the Ihstal Wzard to selectservkes and contpure

1User\*Groap Mansyemert

GAcu

Manage Users +Groups

Create vew instances and grant permtssinr

日

sanape the users and groups that can access Amban

Amm

图4-51 Ambari 控制台界面

**2.** **配置文件的复制**

在 Ambari 主机上配置好 hosts 文件，使用scp 复制至 Slave 节点上。

除了 hosts 文件还要复制 yum 下载的文件，分别是在/etc/yum.repos.d 下的：Ambari.re- po 、hdp.repo,hdp-util.repo 三个源文件，这是为了方便在本地的 http 服务器上下载安装包。

scp Ambari.repo slave2-1.hadoop:/etc/yum.repos.d/

scp Ambari.repo slave2-2.hadoop:/etc/yum.repos.d/

scp hdp.repo slave2-2.hadoop:/etc/yum.repos.d/

scp hdp.repo slave2-1.hadoop:/etc/yum.repos.d/

scp hdp-util.repo slave2-1.hadoop:/etc/yum.repos.d/

scp hdp-util.repo slave2-2.hadoop:/etc/yum.repos.d/

**3.** **使** **用Ambari 搭建** **HDP 集群**

(1)创建集群

点击 Launch Install Wizard即可开始创建集群，如图4-52所示。

(2)开始搭建 HDP 集群，如图4-53所示。输入集群的Name, 点击Next。 (3)选择 HDP 版本，如图4-54所示。

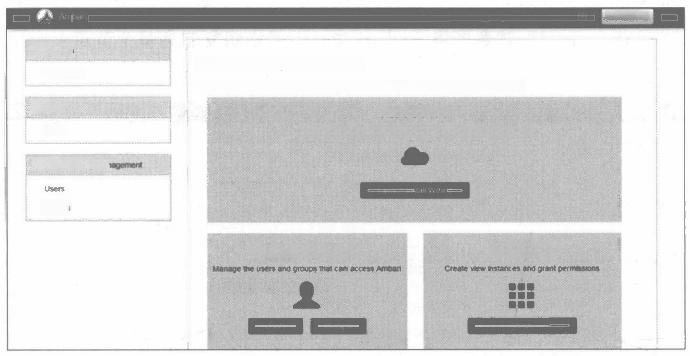
选择搭建 HDP 2.3集群，另外为了加快部署速度，需要选择一下 Repository 选项，如 图4-55 所示。

在这次演示中，我们选择的操作系统是centos7, 因此选择相近的 redhat7, 选择 redhat7 并修改安装源，在进行集群搭建之前，我们已经下载 HDP 并解压到 Ambaril 主机下的/

var/www/html/hdp 目录，并将该目录映射到 http 服务器下的 http://Ambaril/hdp/ 目

大数据技术及行业应用

录，那么 HDP2.3 和 HDP-UTILS 的对应目录就是如图4-55所示。注意：如果不修改本地 安装，那么 Ambari 就会直接从官网下载了。如果要部署的集群节点众多，并且网速跟不上 的话，那么安装过程将会是个非常漫长的过程。



▲ sdmn-



**Welcome to Apache Ambari**

Proweson a causter,manage who tan actess the ttister,and customlte vess trAmbart users.

Create a Clustar

Use he nstat wezart to select sentes and contlgureyour caister

User \*Grap Man

Gio

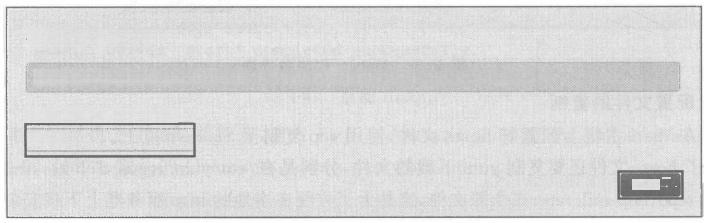
**Manage Users +Groupe** Deplioy Viewa

No cluslers

Vews

Weas

**图4-52** **创建集群界面**



**Get Started**

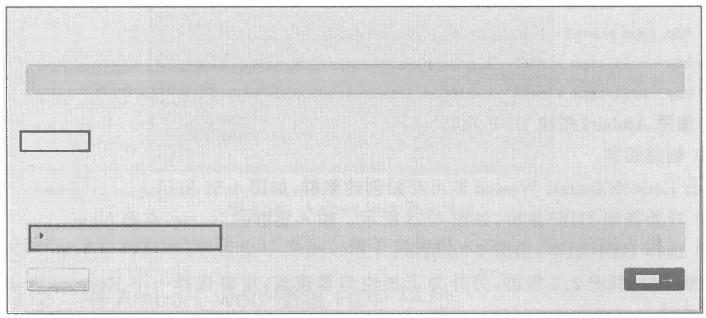
This wzard wll walk you through the cluster installation process.First,start by naming your new cluster.

Name your cluster Leam more

cluster2

Next-

图4-53 设置集群名字



**Select Stack**

Please select the servce stack that you want to use to install your Hadoop cluster.

**Stacks**

HDP 2.3

● HDP 2.2

● HDP2.1

O HDP 2.0

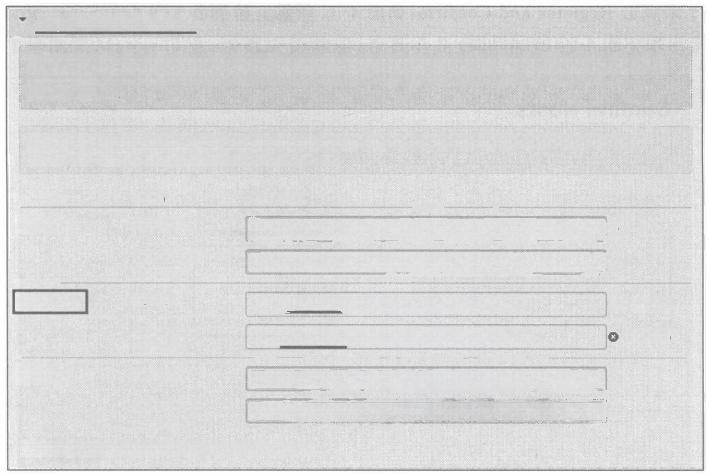
Advanced Repository Options

—Back

Next

图4-54 设置集群版本

**第4章** **大数据平台解决方案**

Advanced Repository Options

Customize the repository Base URLs for downloading the Stack software packages.If your the internet,you wil have to create a local mirror of the Stack repository that is accessible Base URLs here.

Important:When using local mirror repositores,you only need to provide Base URLs for are installing for your Stack Uncheck all other repositortes.

hosts do not have access to

by all hosts and use those

the Operating System you

**Os Name**

□ redhat6 HDP-2.3

HDP-UTILS-

<1.1.0.20>

reahar HDP-2.3

HDP-UTILS-

<1.1.0.20>

suse11 HDP-2.3

HDP-UTILS-

<1.1.0.20>

Skip Repository Base URL validation

**Base URL**

htp //publc-fepo-1 hortonworks comHDP/centos6/2 xiupdates/2.**3○**

<http://publie-fepo-1.hortonworks.com/HDP-UTILS-1.1.0.20frepos/c>( ○

htp //ambant1 hdp/HDPIcentos712 xupdates<2.3.0.0> ○

http//ambant1hdp/HDP-UTILS-<1.1.0.20>repostcentos6

http //publlc-repo-1.hortonworks.com/HDP/suse11sp3/2 x/updates/(O

http //public-repo-1.hortonworks.comvHDP-UTILS-<1.1.0.20>/repos/st o

(Advanced)O

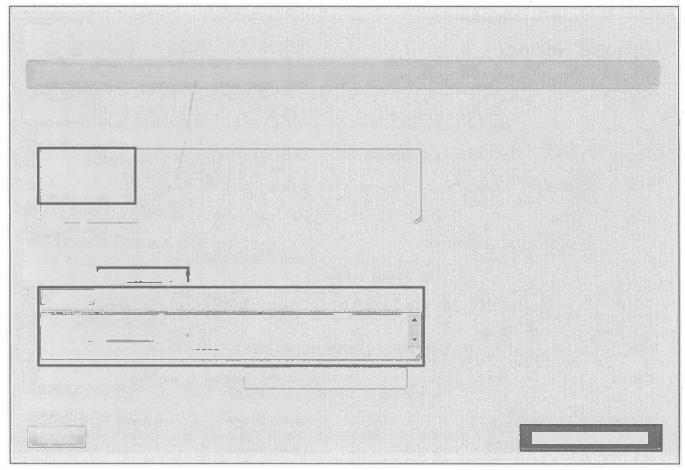
Undo

0Undo

图4-55 设置集群安装源

(4)集群主机配置与检测

接下来就是集群相关的配置，如图4-56所示。填写集群中所有节点的 hostname,一 行 一个主机名。下面填写 Ambari 主机的私钥，可以通过文件导入，也可以通过复制、粘贴的 方式。注意：这里填写的 ssh 不是公钥，而是密钥。公钥在文件 id\_dsa.pub 中，而密钥是 id\_dsa 这个文件中。

**Install Options**

Enter the list of hosts to be included in the cluster and provide your SSH key

**Target Hosts**

Enter a list of hosts using the Fuly Qualihed Domain Name (FQDN),one per line.Or use Pattern Expresstons

masler2

slave2-1.hadoop

slave2-2.hadoop

**Host Registration Information**

●Prowde you[SSH Piwate Keyp automatcaly regster hosts

选择文件 未选择任何文件

5Qu6C

anlrKTtvH/YTSrWWTmtCng

-----END DSA PRIVATE REY-

SSH User Account root

○ Pertorm manual registration on hosts and do not use SSH

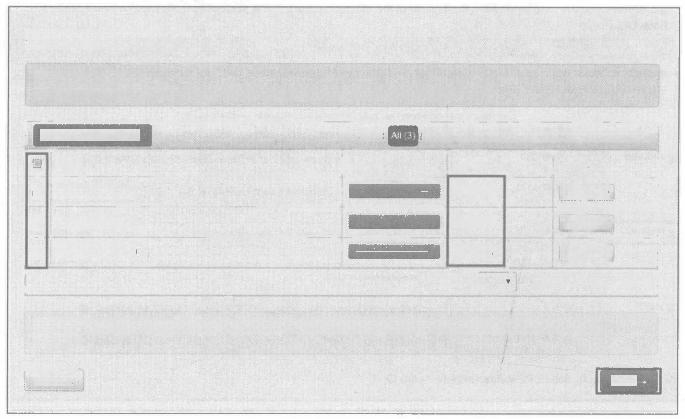
\*-Back

Register and Cenhim-

图4-56 设置集群私钥

大数据技术及行业应用

接下来点击 Register and Confirm,如图4-57所示。检测各个节点是否通，如果检测通 过，就会出现如图4-57所示界面。



**Confirm Hosts**

Registering your hosts.

Please confirm the host ist and remove any hosts that you do nat want to include in the cluster.

Show: nslalling.(Q)I Regskering (Q)I Success 3)1 Fal (@)

Progress Status Action

Success Remove

Success Remove

Sueces Remo

Show: 25 1-3of3 M →M

Some wamings were encountered while performing checks against the 3 registered hosts above Clck here to see the warnings.

Back

**Remove selected**

**Host**

mster2.hadoop sl口ave2-1.hadoop

slve2-2.hadoop

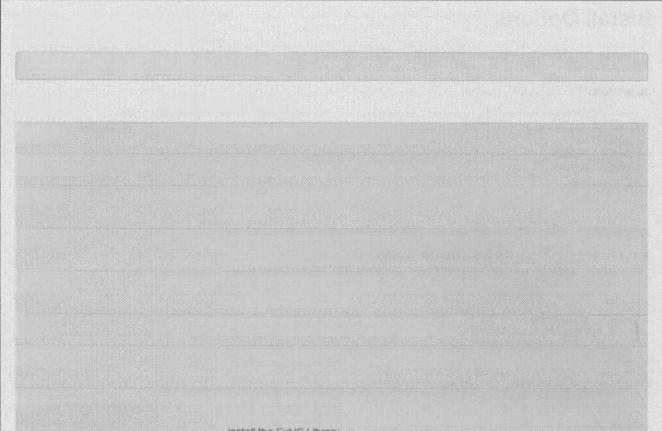
Next

图4-57 测试集群加密通信

把主机都选上，点击下一步。如果用户有 Fail 的情况， 一般来说是 Hostname 和 Hosts 中设置有问题。

(5)选择集群服务

接下来是选择服务，为了方便安装只选择了 HDFS 、YARN+MapReduce2 、Zookeeper 和 Spark 这四个服务，如图4-58所示。



**Choose Services**

Choose which services you want to install on your cluster.

**Descnption**

Apache Hadoop Distrbuted Flle System

Apache Hadoop NextGen MapReduce (YARN)

Tez is the next generatton Hadoop Query Processing framework written on top of YARN.

Data warehouse system tor ad-hoc queries &analysis ot large datasets and table 8 storage management service

A Non-felational distributed database.plus Phoenix,a high performance SOL layer for low latency applications.

Scripting platiorm for analyzing large datasets

Tool for transterring bulk data between Apache Hadoop and structured data stores such as relational databases

System for worktlow coordination and execution of Apache Hadoop jobs.This also includes the installation of the optional Oozle Web Console which relles on and wll

**Service**

HDFS

YARN+MapReduce2

☑ Tez

**Version**

2.7.1.2.3

0.15.0.2.3

4.2.0.2.3

0.7.0.2.3

2.7.1.2.3

1.4.6.2.3

1.2.1.2.3

1.1.1.2.3

Sqoop

E Pg

HBase

Oozle

Hive

**图4-58** **选择集群服务**

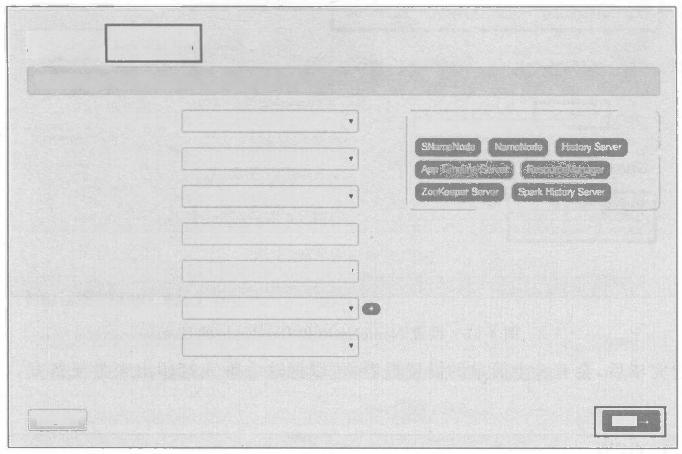
第 4 章 大数据平台解决方案

>)

(6)Master 节 点 和Slave 节点的配置

接下来配置 Master 节点，根据应用需求，选择合适的主机做 Master 节点，如图4-59

所示。



**Assign Masters**

Assign master components to hosts you want to run them on.

master2 hadoop (1.8 GB,1 col

master2.hadoop (1.8

master2 hadoop (1.8

master2.hadoop (1.8

ResourceManager. master2.hadoop (1.8 GB,1 col

ZooKeeper Server master2.hadoop (1.8 GB,1 coi

Spark History Server. master2.hadoop (1.8 GB,1 col

—Back

master2.hadoop (1.8 GB,1 cores)

coi

col

coi

App Tlmeline Server.

History Server.

SNameNode:

NameNode:

GB,1

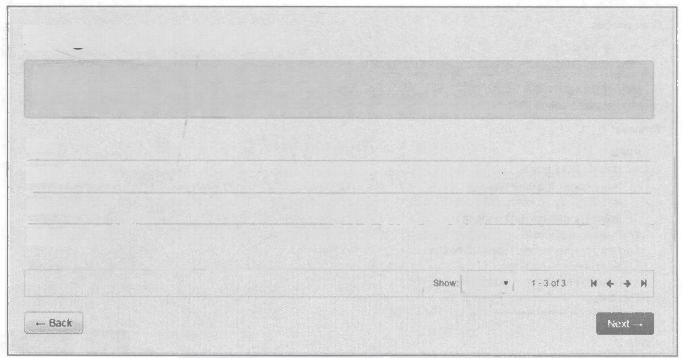
GB,1

GB,1

Next

图4-59 Master 节点配置

设置完 Master 节点，接下来选择 DataNode 节 点 和Client 节点，设置完毕点击Next, 如 图4-60所示。



**Assign Slaves and Clients**

Assign slave and clent componerts to hosts you want to run them on.

Hosts that are assigned master components are shown with \*.

Clent"wil instal HDFS Clent.MapReduce2 Clent,YARN Chent,ZooKeeper Clent and Spark Client

Host all I none all I none all I none

master2.hadoop\* DataNode NFSGateway NodeManage

slave2-1.hadoop DataNode □ NFSGateway NodeManager

slave2-2.hadoop DataNode NFSGateway NodeManager

25

all I none

□ Chent

CMent

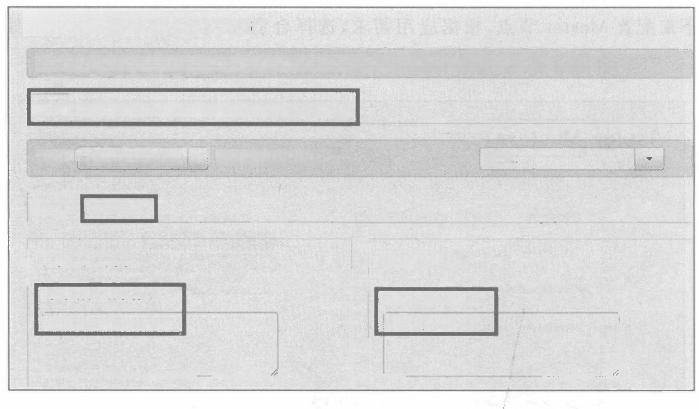
Clent

图4-60 和 slave节点配置

最后是对集群进行详细设置。这里对集群的设置还是相当详细的，可以设置 NameNode 存储位置和 DataNode 的位置，点击Advanced 有更详细的设置。用户可以根据应用需求，进行 详细定制，如图4-61所示。



大数据技术及行业应用



**Customize Services**

We have come up with recommended configurations for the services you selected.Customize them as you see ft.

HDFS MapReduce2 YARN ZooKeeper Spark Misc

Group HDFS Defaut (3)  Manage Contig Groups Fiter..

Settings Advanced

**NameNode**

NameNode directories

hadoop/hdfs/namenode

D Node drectones

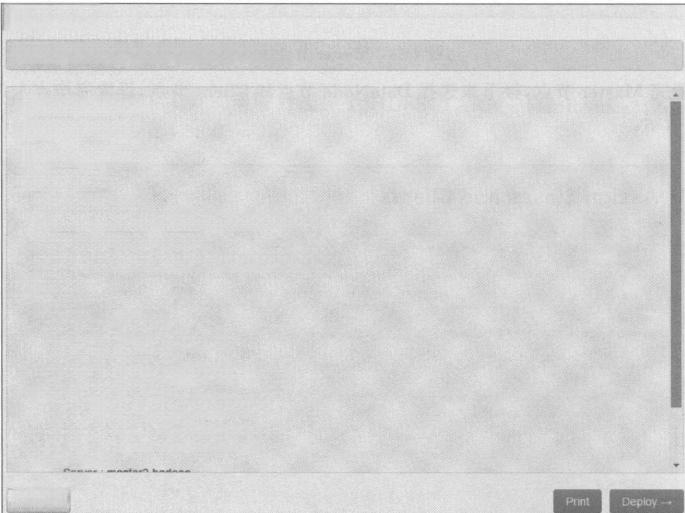
hadoop/hdfs/data

**DataNode**

图4-61 设 置 NameNode 和 DataNode 的目录

配置完毕后，会有一个简单的设置报表，可以把这个报表打印出来方便备案，如图4-62

所示。

**Review**

Please review the configuration before installation

**Admin Name** :admin

**Cluster Name**:cluster2

**Total Hosts:**3(3 new)

**Repositones**

redhat7 (HDP-2.3):

http://ambari¹/hdp/HDPIcentos7/2.x/updates/<2.3.0.0>

redhal7 (HDP-UTILS-<1.1.0.20>):

http.//ambari1/hdp/HDP-UTILS-<1.1.0.20>/repos/centos6

**Services:**

**HDFS**

DataNode:2 hosts

NameNode master2.hadoop

NFSGateway:0 host

SNameNode:master2.hadoop

***YARN+MapReduce2***

App Timelne Server:master2.hadoop

NodeManager:2 hosts

ResourceManager:master2.hadoop

**ZooKeeper**

—Back

图4-62 配置 一 览表

(7)开始部署集群

开始安装，集群部署界面，如图4-63所示。

安装完成，集群部署成功界面，如图4-64所示。

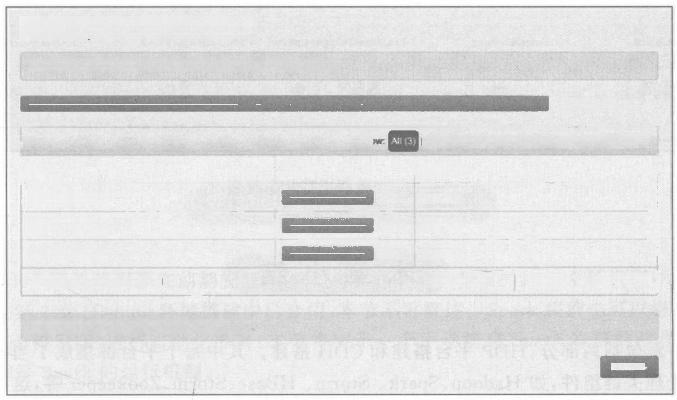
**第4章** **大数据平台解决方案**

>>



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Install,Start and Test**  Pease walt while the selected servces are instaled and started.  **4 %overall**  Show n Proqress.3)1 Warning (0)I Success(Q)I Eat (o   |  |  |  | | --- | --- | --- | | **Host** | **Status** | **Messagt** |   master2 hadoop 4% instaling App Timeine Server   |  |  |  | | --- | --- | --- | | slave2-1.hadoop | 5% | Installing DataNode |   slave2-2.hadoop 5% Installing DataNode | | |
|  | 30f3 hosts showing-Show Al Show 25 1-3013 **H+\*H** |  |
| Next | | |

**图4-63** **集群部署界面**



**Install,Start and Test**

Please wait while the selected services are installed and started.

100 %overall

n Progress (0)1 Wamingt0)I Success(3)1 FalltO)

**Message**

Success

Success

Success

Show 25  1-3gf3 M →

Successtuly installed and started the services

Next=

**Host**

master2.hadoop

slave2-1.hadoop

slave2-2.hadoop

3013 hosts showing-Show All

100%

100%

100%

**Status**

**Sho**

图4-64 集群部署成功界面

点击Conplete, 进入集群部署总结界面，如图4-65所示。

**Summary**

Here is the summary of the install process.

The cluster consists of 3 hosts

Installed and started servces successtuly on 3 new hosts

Master servces installed

SNameNode installed on master2 hadoop

NameNode instaled on master2,hadoop

History Server installed on master2 hadoop

ResourceManager installed on master2.hadoop

Al services siarted

All tests passed

Install and start completed in 26 minutes and 24 seconds

Compete-

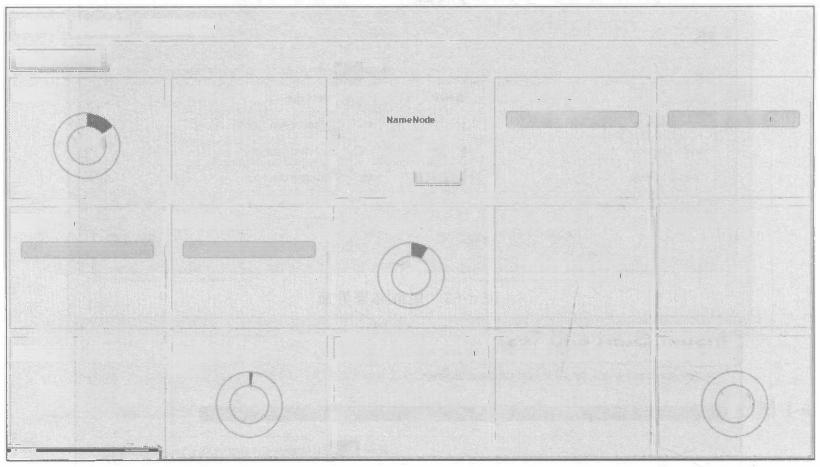
图4-65 集群部署总结界面



大数据技术及行业应用

**(8)部署成功**

**可以查看整个集群的状态，界面直观，如图4-66所示。**



Metrics Heatmaps

Metric Actions ~

**HDFS Links Memory Usage Network Usage**

No Data Avallabie No Data Available

Secondary NameNode

2 DataNodes

More …

**NameNode Heap** NameNode RPC NameNode CPU WIO

No Data Available No Data Avallable

9% **0.50** **ms** n/a

**ResourceManager Uptime** NodeManagers Llve YARN Memory

Heap

**2/2**

**553.8** **s**

**654.4** **s**

**CPU Usage Cluster Load**

**2/2**

**NameNode Uptime**

**HDFS Disk Usage**

**ResourceManager**

Confg History

**DataNodes Live**

**16%**

0%

**2%**

图4-66 集群运行状态报表



**本** **章** **小** **结**

这一章包括两部分：HDP 平台搭建和CDH 搭建。其中每个平台都集成了当前主流的 大数据处理关键组件，如 Hadoop 、Spark 、Storm 、HBase 、Storm 、Zookeeper 等，这些组件满 足企业对计算和存储的需求，从分布式文件存储到数据查询，再到高级机器算法全方位构建 了满足生产环境的大数据平台。

**第** **5** **章** **Spark** **在大数据处理中的应用**

Spark 是 Apache 软件基金会旗下一个快速通用的大规模数据处理引擎。该软件具有 四个特点：运行速度快、支持多种高级语言、通用性强、支持多种分布式文件系统和数据库。

Spark 的运行速度在内存中据称比 Hadoop 快100倍，在硬盘上比 Hadoop 快10 倍。 Apache Spark 拥有先进的 DAG 执行引擎，其支持循环数据流和内存计算。Spark 支持 Java 、Scala、Python 和 R 等高级语言，并且提供80多种方便构建并行应用程序的高效操作。 Spark 集成了 SQL、流计算和复杂网络分析，其支持一系列类库包括 SQL 和数据框架、 MLib 、GraphX 和 Spark Streaming,用户可以把这些类库无缝地封装到一个基于 Spark 的 应用中去。Spark 可以运行在 Hadoop 、Mesos 、standalone 或者是云平台上，其可以访问多 种数据源包括 HDFS 、Cassandra 、HBase 和 S3。



**5.1 Spark 集群搭建**

Spark 集群搭建需要在前期安装JDK、Scala 和 Hadoop 集群。JDK 环境和 Hadoop 集 群环境在第2章 Hadoop 环境搭建中已介绍，本节不再赘述。Spark 集群也可以通过第3章 中的CDH 和 HDP 进行自动化部署，本节将演示一个手动搭建 Spark 集群的过程，方便读 者深入理解 Spark 的运行机制。

**5.1.1** **Scala** **在Ubuntu** **下的安装和配置**

Scala 是一种可扩展性语言，其既支持面向对象编程，也支持函数式编程。其运行在 Java 虚拟机上，可以轻松实现与Java 类库互通。目前支持Scala 的 IDE 环境有 Typesafe 公 司开发的基于 Eclipse 的 IDE 。该 IDE 提供了一个方便的功能 Worksheet 。Worksheet 在 用户输入 Scala 表达式并保存后立即可以得到程序的运行结果，非常方便用户体验 Scala 语 言的各种特性。Spark 为 Scala 提供了脚本执行环境 Spark-shell 。Spark 的内核是用Scala 语言实现的，因而安装 Spark 之前，必须先安装 Scala 语言。

(1)下载 Scala压缩包，下载地址：<http://www.scala-lang.org/download/2.11.7.html> 下载好的 tar 包传到服务器上，解压 Scala 到 opt 目录下。

把

tar zxvf scala-2.11.7.tgz -C/opt/

(2)创建软链接“ls-s/opt/scala-2.11.7/opt/scala”。

(3)配置/etc/profile 文件“vi/etc/profile”。

(4)在 JDK 环境变量后加上：

#set scala environment

export SCALA\_HOME=/opt/scala

export PATH=$PATH:$SCALA\_HOME/bin

使用 source /etc/profile编译，使PATH 环境变量生效，scala-version 查看 Scala 版本， 显示如图5-1 所示内容，即安装成功。



图5-1 查看 Scala 版本

所有要安装 Spark 的节点，都要先安装 Scala 。 也可以通过 scp 命令，将以上配置好的 文件分发到其他节点。

注意：安装 Scala 之前，需提前安装JDK 环境。

**5.1.2** **Spark** **集群搭建**

**1.** **集群规划**

本次 Spark 集群搭建将使用4台主机，规划如下：

<192.168.10.101> <192.168.10.111> <192.168.10.112>

<192.168.10.113>

masterl.jie.com

slavel-1.jie.com

slavel-2.jie.com

slavel-3.jie.com

(master)

(worker)

(worker)

(worker)

其中有一个 Master 节点，三个 Worker 节点。

**2.ssh** **免密钥登录配置**

如果不会 ssh 免密钥登录，请参考 Hadoop 集群搭建中的ssh 免密钥登录，在此不再赘述。 3. 下载安装 Spark

(1)下载 Spark:wget <http://www.apache.org/dyn/closer.lua/spark/spark-1.5.1/> spark-1.5.1.tgz。

(2)解压、安装 Spark:tar zxvf spark-1.5.1.tgz-C /opt/hadoop。

(3)建立软链接：In-s/opt/hadoop/spark-1.5.1/opt/hadoop。

(4)将 Spark 中 bin 目录添加到 PATH 路径中：vi/etc/profile。

(5)在最后添加如下内容：

#set spark environment

export SPARK\_HOME=/opt/hadoop/spark

export PATH =$SPARK\_HOME/bin:MYMPATH

Spark 安装完成。

**4.Spark** **配置**

Spark 的配置文件都在$SPARK\_HOME/conf 文件夹下。

(1)slaves 文件配置。

① 如果在 conf 目录下没有 slaves 文件，复制一份 slaves.template, 重命名为 slaves:cp

slaves.template slaves。

② 编辑 slaves 文件：vi slaves。

③ 将 Worker 节点添加到其中：

slavel-1.jie.com

slavel-2.jie.com

slavel-3.jie.com

(2)spark-env.sh 文件的配置。

cp spark-env.sh.template spark-env.sh

vi spark-env.sh

① 添加如下内容：

export JAVA\_HOME=/usr/java/jdk ###java 安装目录

export SCALA\_HOME=/usr/scala-2.11.4 ##scala 安装目录

export SPARK\_MASTER\_IP= <192.168.52.128> ### 集群中master 机器 IP

exportSPARK\_WORKER\_MEMORY =2g

###指定的 worker节点能够最大分配给 Excutors 的内存大小

export HADOOP\_CONF\_DIR=/opt/hadoop/etc/Hadoop

###Hadoop 集群的配置文件目录

② 保存退出即可

③ 最后通过 scp 命令，将以上配置好的文件分发到其他节点。

**5.1.3** **Spark** **集群启动测试**

**1.Spark 启动**

启动 Spark 集群：

$SPARK\_HOME/sbin/start-all.sh

如果没有报错，即搭建成功，使用jps 查看是否在 Master 节 点 和 Worker 节点上启动 Master 进程和 Worker 进程，使用jps 查 看 Master 节点进程如图5-2所示，使用jps 查 看 Worker 节点进程如图5-3 所示。



**rootmaster1:/opt/hadoop/apark/conf\*jps 1410 Master**

1609 Jps

Foot@slave1 1: #jpa

1315 Jps - -

1180 Worker

图5-2 使用 jps 查看 master 节点进程 图5-3 使用jps 查看 worker节点进程

**2.Spark-shell**

(1)启动 Spark-shell:$SPARK\_HOME/bin/sprk-shell,

如图5-4所示。

|  |
| --- |
| 15/11/0120:25:27 INFO HttpServer:Starting HTTP Server  15/11/0120:25:28 INFO Utils:Successfully started service 'HTTP class server'on port 5357 0.  Helcome to  version 1.4.1  Using Scala version 2.10.4(Java HotSpot (TM)64-Bit Server VM,Java 1.8.0\_51) Type in expressions to have them evaluated.  Type :help for more information  15/11/0120:25:35 INFO SparkContext:Running Spark version 1.4.1  15/11/0120:25:35 INF0 SecurityManager:Changing view acls to:root  5/11/0120:25:35 INFO SecurityManager:Changing modify acls to:root  15/11/0120:25:35 INFO SecurityManager:SecurityManager:authentication disabled;ui acls d isabled;users with view permissions:Set(root);users with modify permissions:Set(root) 15/11/0120:25:36 INFO S1f4jLogger:S1f4jLogger started  15/11/0120:25:36 INF0 Remoting:Starting remoting  15/11/0120:25:37 INFO Remoting:Remoting started;listening on addresses:[akka.tcp://spar kDrivere<192.168.10.101>:33831] |

图5-4 Spark-shell 启动界面



大 数 据 技 术 及 行 业 应 用

(2)Spark 的欢迎界面，进入后的界面如图5-5所示。



engine=mr.

15/11/0120:26:00 INEO SQL context available

Lcala> 工

SparkILoop:Created as aqlContext.

aql context(with Hive support) … .

图 5 - 5 Spark 的 欢 迎 界 面

当出现以上界面的时候，证明 Spark 平台搭建成功，接下来可以去 Spark 官网阅读相关 文档进行实验了，网址如下 <http://spark.apache.org/docs/latest/quick-start.html>。

**5.2** **Spark-shell** **统计社交网络中节点的度**

5 . 1节中我们搭建了 Spark 集群，本节我们通过一个在社交网络计算中经常用到的一 个操作来测试一 下5 . 1节中搭建的 Spark 集群。

**5.2.1** **启动** **HDFS 和** **Spark**

进入 Hadoop 的 sbin 目录，然后运行启动 HDFS 的命令，如图5-6所示。

cd $Hadoop\_HOME/sbin/

./start-dfs.sh

rootemaster:/opt/hadoop/bint cd /opt/hadoop/sbin

root@master:/opt/hadoop/sbin#./start-dfs.sh

15/11/0910:54:49 WARN util.NativeCodeLoader:Unable to load native-hadoop libra

Starting namenodes on [master]

master:starting namenode,logging to /opt/hadoop/logs/hadoop-root-namenode-mast

slave2:starting e2.out

master:starting er.out

slave1:starting e1.out

datanode,logging datanode,logging datanode,logging

to /opt/hadoop/logs/hadoop-root-datanode-slav

to /opt/hadoop/logs/hadoop-root-datanode-mast

to /opt/hadoop/logs/hadoop-root-datanode-slav

slave4:ssh:connect to host slave4 port 22:No route to host

slave5:ssh:connect to host slave5 port 22:No route to host

slave3:ssh:connect to host slave3 port 22:No route to host

Starting secondary namenodes [master]

master:starting secondarynamenode,logging to /opt/hadoop/logs/hadoop-root-secondarynamenode

英 ● , 管 上

15/11/0910:55:06 WARN util.NativeCodeLoader:Unable to load native-hadoop library for your plt

tin-java classes where applicable

rootemaster:/opt/hadoop/sbinw

图 5 - 6 启 动 HDFS

运行jps, 看 NameNode 和 DataNode 是否已经启动。如果没有启动 NameNode, 需 要

对 NameNode 进行格式化，运行如下命令：$ Hadoop\_HOME/bin/hadoop Namenode-for-

mat, 如图5-7所示。

rootamaster:/opt/hadoop/sbin#jps

NameNode

12197 DataNode

13711 Jps

rootemaster:/opt/hadoop/sbin#

图 5 - 7 运 行 jps 查 看 NameNode 和 DataNode 是 否 已 经 启 动

第5章 Spark 在大数据处理中的应用

**进入** **Spark的** **sbin 目录，然后运行** **start-all 的命令，如图5-8所示。**

|  |
| --- |
| rootemaster:/opt/hadoop/sbin#cd /usr/spark/sbin  root@master:/usr/spark/sbin#./start-all.sh  starting org.apache.spark.deploy.master.Master,logging to /usr/spark/sbin/../logs/spark-root-org.apache,spark.dep loy.master.Master-1-master.out  slave2:starting org.apache.spark.deploy.worker.Worker,logging to /usr/spark/sbin/../logs/spark-root-org.apache.s ark.deploy.worker.Worker-1-slave2.out  slave1:starting org.apache.spark.deploy.worker.Worker,logging to /usr/spark/sbin/../logs/spark-root-org.apache.s park,deploy.worker.Worker-1-slave1.out  naster:starting org.apache.spark.deploy.worker.Worker,logging to /usr/spark/sbin/../logs/spark-root-org.apache.s park.deploy.worker.worker-1-master.out  slave4:ssh:connect to host slave4 port 22:No route to host  slave5:ssh:connect to host slave5 port 22:No route to host  slave3:ssh:connect to host slave3 port 22:No route to host |

图 5 - 8 启 动 Spark

cd §SPARK\_HOME/sbin

./start-all.sh

运行jps 查看启动的服务，如图5-9所示。

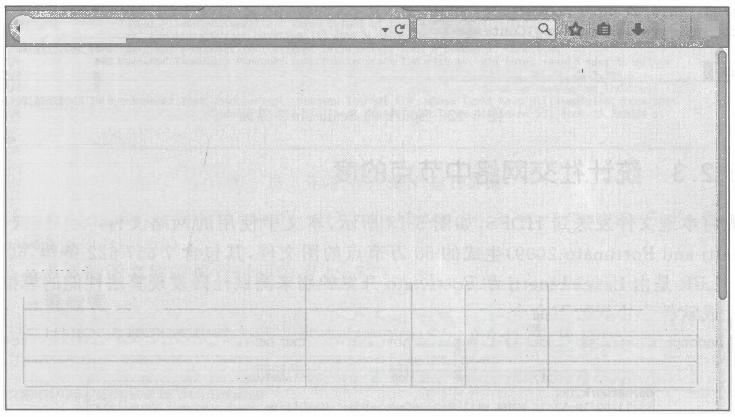
rootamaster:/usr/spark/sbin#jps

14419 Worker

|  |  |
| --- | --- |
| 12020 | NameNode |
| 12405 | SecondaryNameNode |
| 12197 | DataNode |
| 14201 | Master |
| 14782 | Jps |

图 5 - 9 运 行 jps 查 看 启 动 的 服 务

如果已经启动，查看http://master:8080 的界面，如图5-10所示。



mastor2080

**Master at spark://**[**192.168.10.130**](192.168.10.130)**:7077**

URL:spark://<192.168.10.130>:7077

**Workers:3**

**Cores:3** Total,0 Used

**Memory:**24.0 GB Total,0.0 B Used

**Applications:** 0 Running,0 Completed

**Drivers:0 R**unning,0 Completed

**Status:ALIVE**

**Workers**

**ld**

worker-20151109110101-master-38870

worker-20151109110101-slave1-59182

worker-20151109110101-slave2-45221

Memory

8.0 GB (0.0 B Used)

8.0 GB(0.0 B Used)

8.0 GB(0.0 BUsed)

**Address**

master:38870

slave1:59182

slave2:45221

Cores

1 (0 Used)

1 (0 Used)

1(0 Used)

图▼百度 合 三

State

ALIVE

ALIVE

ALIVE

**Spark**

Sp**a\***

图 5 - 1 0 Spark 启 动 后 显 示 状 态 的 Web 界 面

**5.2.2** **运行** **Spark-shell**

运行 Spark-shell, 命令如下：

S SPARK\_HOME/bin/spark-shell

进入欢迎界面，如图5-11所示。

|  |  |
| --- | --- |
| log4j:WARN See <http://logging.apache.org/log4j/1.2/faq.htmlfnoconfig> for more info. Using Spark's default log4j profile:org/apache/apark/log4j-defaults.properties  15/11/0319:16:07 INF0 SecurityManager:Changing view acls to:root  15/11/0319:16:07 INFO SecurityManager:Changing modify acls to:root  15/11/0319:16:07 INFO SecurityManager:SecurityManager:authentication disabled;ui acls d isabled;users with view permissions:Set(root);users with modify permissions:Set(root)  15/11/0319:16:07 INFO HttpServer:Starting HTTP Server  15/11/0319:16:07 INFO Utils:Successfully started service 'HTTP class aerver'on port 3571  o.  Melcome to | |
|  | version 1.4.1 |
| Gsing Scala version 2.10.4(Java HotSpot (TM)64-Bit Server VM,Java 1.8.0\_51) Type in expressions to have them evaluated.  Type :help for more information.  15/11/0319:16:14 INFO SparkContext:Running Spark version 1.4.1 | |

图5-11 Spark 运行界面

Spark 进入后界面，启动的 Spark-shell, 使用的交互语言是 Scala, 如图5-12所示。

|  |
| --- |
| 15/11/0319:16:37 INFO HiveMetaStore:Added public role in metastore  hs/11/0319:16:37 INFO HiveMetaStore:No user is added in admin role,since config is empty 15/11/0319:16:37 INFO SessionState:No Tez session required at this point.hive.execution. engine=mr.  h5/11/0319:16:37 INFO SparkILoop:Created sql context(with Hive support)..  SQL context available as aqlContext.  gcala> |

图5-12 Spark 的 Scala shell 界面

**5.2.3** **统计社交网络中节点的度**

先将本地文件发送到 HDFS, 如图5- 13 所示，本文中使用的网络文件是由LFR(Lan-

cichinetti and Fortunato 2009)生成的50万节点的图文件，其包含7657622条边，50万个 节点。LFR 是 由Lancichinetti 和 Fortunato 开发的用来测试社区发现算法性能的模拟社交 网络生成软件。执行如下命令：

S hadoop\_home$/bin/hadoop fs -put/root/network.txt hdfs://master:9000/network.txt



rootemaster:/opt/hadoop/bin#./hadoop fs -put /root/network.txt hdfs://master:90

00/network.txt

15/11/0910:52:16 WARN util.NativeCodeLoader:Unable to load native-hadoop libre

ry for your platform..using builtin-java classes where applicable

root@master:/opt/hadoop/bin#

图5-13 启动 HDFS

**1.** **读取** **textFile**

Spark shell 使用须知：Spark shell 自动初始化 SparkContext 命名为 sc, 所以不需要重

复初始化 SparkContext。

第5章 Spark 在大数据处理中的应用



首先，将 network 文 件 载 入 Spark 中，读取 network.txt 如图5 - 14所示。

val line=sc.textFile("hdfs://master:9000/network.txt ")

scala>var line=sc.textFile("hdfs://master:9000/network.txt")

15/11/0911:06:21 INFO storage.MemoryStore:ensureFreeSpace(162307)called with

INFO storage.MemoryStore:Block broadcast\_0 stored as values

15/11/0911:06:21

1/0911:06:21

15/11/0911:06:21 67.1 MB)

15/11/0911:06:21 67.2 MB)

15/11/8911:06:21

15/11/0911:06:21

INFO storage.MemoryStore:ensureFreeSpace(23048)called INFO storage.MemoryStore:Block broadcast 8 pieceθ stored

with as

INFO storage.BlockYanagerInfo:Added broadcast 0 piece0 in

INFO storage.BlockManagerMaster:Updated info of block

INFO spark.SparkContext:Created broadcast θ from textFile

ine:org.apache.spark.rdd.RDDlstring]=hdfs://master:9000/network.txt MappedRDD[1]at

curMem=0,maxMem=280248975

in memory(estimated size 158.5 KB,

curMem-162307,maxMem=280248975

bytes in memory(estimated size 22

memory on localhost:47685(size:22.9

broadcast 0 pieced

at <console>:12

textFile at <console>:12

scala>

图5-14 读取 network.txt

**其次，语句将分词、合并、统计一步到位，如图5-15所示。**

|  |  |  |  |
| --- | --- | --- | --- |
| 1571178911:12:38 15/11/0911:12:38 | INF0 scheduler.DAGScheduler:Nissing parents  INFO scheduler.DAGScheduler:Submitting Stage | for Stage I:List(T  1(MappedRDD[9]at saveAsTextFile | at cconsolez;15),which is n |
| ow runnable  5/11/0911:12:38 INFO storage.MemoryStore:ensureFreeSpace(112704)called with curMem=191428,maxMem=288248975  5/11/0911:12:38 INFO storage.MemoryStore:Block broadcast\_2 stored as values in memory (estimated size 110.1 KB,free 267.8 MB)  5/11/0911:12:38 INF0 storage.MemoryStore:ensureFreeSpace(67916)called with curMem-384132,maxHem=280248975  5/11/0911:12:38 INFO storage.MemoryStore:Block broadcast 2 pieceθ stored as bytes in memory(estimated size 66.3 KB,free 2  66.9 MB)  5/11/0911:12:38 INFO storage.BlockNanagerInfo:Added broadcast 2 pieceθ in memory on localhost:47685(size:66.3 KB,free:2 67.2 MB)  15/11/0911:12:38 INFO storage.BlockManagerMaster:Updated info of block broadcast 2 piece0  15/11/0911:12:38 INF0 spark.SparkContext:Created broadcast 2 from broadcast at DAGScheduler,scala:838  5/11/0911:12:38 INFO scheduler.DAGScheduler:Submitting 1 missing tasks from Stage 1(MappedRDD[9]at saveAsTextFile at <con sole>:15)  15/11/0911:12:38 INFO scheduler.TaskSchedulerImpl:Adding task set 1.8 with 1 tasks  15/11/0911:12:38 INF0 scheduler.TaskSetManager:Starting task 0.0 in stage 1.0(TID 1,localhost,PROCESS LOCAL,1056 bytes) 15/11/0911:12:38 INFO executor.Executor:Running task 0.0 in stage 1.8(TID 1)  5/11/0911:12:39 INFO storage.ShuffleBlockFetcherIterator:Getting 1 non-empty blocks out of 1 blocks  5/11/0911:12:39 INFO storage.ShuffleBlockFetcherIterator:Started 0 remote fetches in 14 ms  5/11/0911:12:42 INFO output.FileOutputCommitter;Saved output of task 'attempt\_201511091112\_0001 m\_000000\_1'to hdfs://maste r:9000/network\_results.txt/\_temporary/θ/task\_201511891112\_0001 m 000800  15/11/8911:12;42 INF0 spark.SparkHadoopWriter:attempt 2015110911120001 m 0000001:Committed  5/11/0911:12:42 INFO executor.Executor:Finished task 0.0 in stage 1.0(TID 1).824 bytes result sent to driver  15/11/0911:12:42 INF0 scheduler.DAGScheduler:Stage 1(saveAsTextFile at cconsole>:15)finished in 3.390 s  15/11/0911:12:42 INFO scheduler.TaskSetManager:Finished task 0.0 in stage 1.0(TID 1)in 3398 ms on localhost(1/1) 15/11/0911:12:42 INF0 scheduler.TaskSchedulerImpl:Removed TaskSet 1.0,whose tasks have all completed,from pool  15/11/0911:12:42 INFO scheduler.DAGScheduler:Job 8 finished:saveAsTextFile at <console>:15,took 18.723524 s  scalay | | | |

图5-15 分词、合并、统计运行界面

line.flatMap(\_.split("\\t")).map((\_,1)).reduceByKey( + ).saveAsTextFile("hdfs://master:

9000/network\_results.txt“)

50万节点的图文件的执行时间约为18 . 72s。

**2.** **查** **看** **数** **据**

先在 HDFS 上将目录获取下来。进入 Hadoop 下 的 bin 目录，运行如下命令： "./hadoop fs -get hdfs://master:9000/network\_results.txt"。



Footamaster:/usr/spark/binf cd /opt/hadoop/bin

15/1/0911:18:16 WwARN util.NativeCodeLoader:Unable to load native-hadoop library for your platform.…using builtin-java cla

sses where applicable

**container-exec,h**/a**oop/****in** **h****sfs.cnd mapred.cnd**

**adoof hdfs mapred** metwork\_results.txt

**oot@ma** st er:/opt/hadoop/bint

rootemaster:/opt/hadoop/bing./hadoop fs -get hdfs://master:9000/network\_results.txt

rcc

**test-container-executor**

yarn

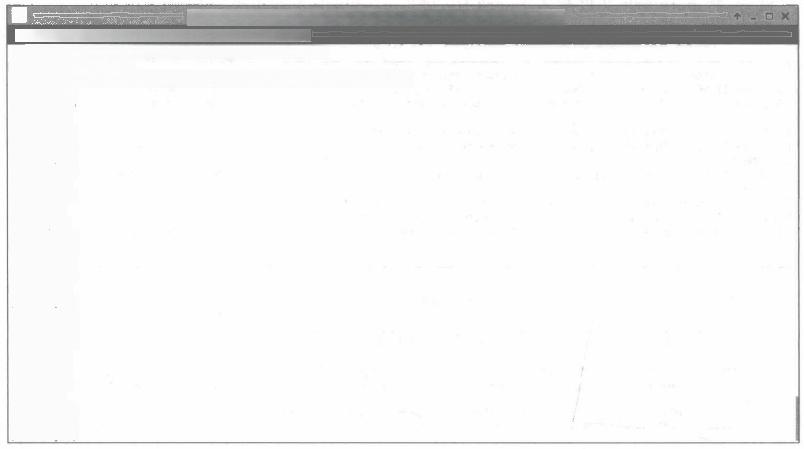
yarn.cnd

图5-16 获取处理结果

接着执行命令“cd network\_results.txt” 。使 用 more 命令查看数据结果，如图5 - 17

所示。

|  |  |  |
| --- | --- | --- |
| rootemaster1:/or  2828,14  400543,44  136547,18  281270,26  43399,14  432234,52  257529,24  302283,28  253895,24  462712,68  6611,14  119687,16  22620,14  348711,34  415932,48  86251,16  94803,16  374499,38  356091,34  137731,18 | esult-5000000.datat | mGT result\_5000000.cav |



四 Termlnal -rootemaster:fopt/hadoop/bin/network\_results.xt

文件(F) 编 强(E) 视 图(V) 终 端(J) 标 签(A) 帮 助(H)

rootamaster:/opt/hadoop/bin/network results.txt#ls

part-00000 SUCCESS

rootemaster:/opt/hadoop/bin/network\_results.txt#more part-00000

(328219,30)

(144176,18)

(43399.14)

(432234,52)

(257529,24)

(304364,28)

(302283,28)

(75768,14)

(156809,18)

(6611,14)

(22620,14)

(203668,20)

(300853,28)

(415932,48)

(326937,30)

(94803,16)

(389549,40)

(142717,18)

(257478,24)

(382686,40)

(26601,14)

(17789,14)

(27639,14)

(1655,14)

(137203,18)

(348125,34)

(278114,26)

图5-17 查看处理结果

使 用 sed 命令去掉括号，保存成 result\_500000.csv,sed 处理后的文件如图5 - 18所示。

使 用 Office 中 的 Excel 打 开 result-500000.csv 文件界面截图，如图5- 19所示。

|  |  |  |  |
| --- | --- | --- | --- |
|  | A | B | C |
| 1 | 2828 | 14 |  |
| 2 | 400543 | 44 |  |
| 3 | 136547 | 18 |  |
| 4 | 281270 | 26 |  |
| 5 | 43399 | 14 |  |
| 6 | 432234 | 52 |  |
| 7 | 257529 | 24 |  |
| 8 | 302283 | 28 |  |
| 9 | 253895 | 24 |  |
| 10 | 462712 | 68 |  |
| 11 | 6611 | 14 |  |
| 12 | 119687 | 16 |  |
| 13 | 22620 | 14 |  |
| 14 | 348711 | 34 |  |
| 15 | 415932 | 48 |  |
| 6 | 86251 | 16 |  |
| 7 | 94803 | 16 |  |
| 18 | 374499 | 38 |  |
| 19 | 356091 | 34 |  |
| 20 | 137731 | 18 |  |
| 21 | 137203 | 18 |  |
| 22 | 213844 | 20 |  |
| 23 | 123383 | 16 |  |
| 24 | 222585 | 22 |  |

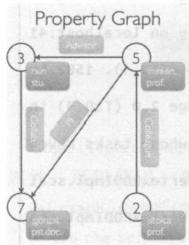
图5-18 sed 处理后的文件

图5-19 使用Office 中的 Excel 打开 result-500000.csv 文件截图



**5.3 Spark GraphX**

GraphX 是 Spark 中针对图计算开发的新组件，可以实现并行图计算。GraphX 利 用 Spark 为计算引擎实现了大规模图计算的功能，并提供了类似Pergel 的 编 程 接 口 。GraphX

将 ETL 、探索性分析和迭代图计算统一在一个系统中。用户可以将相同的数据看成图和集 合，用 RDD 高效地转换和加入图，用PregelAPI 写定制的迭代图算法。“图计算”是以“图 论”为基础的对现实世界的一种“图”结构的抽象表达，以及在这种数据结构上的计算模式。 通常，在图计算中，基本的数据结构表达就是：G=(V,E,D),G 代 表 一 个 网 络 ，V 是该网络 中节点的集合， E 是该网络中边的集合， D 是该网络中权重的集合。

GraphX 扩展了 Spark RDD,通过引入图抽象：带有顶点和边属性的有向多重图。为了 支持图计算，GraphX 提供了很多基本的操作(像 subgraph、joinVertices and aggregateMes- sages)和 Pregel 的 一个优化变种。除此之外，GraphX 包含了一个正在增长的图算法和图 构造的集合来简化图的分析任务。

**5.3.1** **属性图**

属性图是一个多重有向图，连接着每个顶点和边。有向多重图是指有多条平行的边共 同分享着相同的源节点和目的节点的有向图。对多条平行边的支持能简化多重关系的建 模。每个节点有 一个64位长的ID 号唯一标识每一个节点，在 GraphX 中 ID 是不存在先后 顺序的，但是在边中有源节点和目的节点的约束。

构建属性图需要有两个 vertex(VD) 和 edge(ED) 类型的参数。当数据类型为基础类型 时 ，GraphX 会优化节点和边的表示，通过存储在特殊的数组中以减少内存开销。

在实际应用场景中，有些节点具有不同类型的属性，在 GraphX 中可以通过继承 Ver- texProperty 实现。例如，建模一个二分图中的用户和产品，我们可以通过如下代码实现：

class VertexProperty()

case class UserProperty(val name:String)extends VertexProperty

case class ProductProperty(val name:String,val price:Double)extends

VertexProperty

//The graph might then have the type:

var graph:Graph[VertexProperty,String]=null

像 RDDs, 属性图是不变的、分布式的和容错的。改变图的值和结构是通过生成包含改 变的新图来实现。原始图的大部分(不影响结构、属性和索引)都可以在新图中重用，用来减 少固有的功能数据结构的成本。通过executors 方法能够在 一个顶点范围内对 一 张图进行 启发式分区。像 RDDs 一样，当发生故障时，图的每一个分区能被重新创建在不同的机器 上。下面我们看一个属性图例子。

Vertex Table

|  |  |
| --- | --- |
| ld | Property (V) |
| 3 | (rxin,student) |
|  | (gonzal,postdoc) |
| 5 | (frankin,professor) |
|  | (istoica,professor) |

EdgeTable

|  |  |  |
| --- | --- | --- |
| Sreld | Dstdd | Property(E) |
| 3 | 7 | Collaborator |
| 5 | 3 | Adwisor |
| 2 | 5 | Coleague |
| 5 | 7 | PI |

图5-20 合作关系图

假设构建一个包含不同合作者的属性 图，顶点属性包含用户名和职业，边使用字符 串描述和作者之间的关系，如图5-20所示。

从原始文件、RDDs 或者是合成生成器可 以构建一个图，最基本的方法是使用图对象。 例如下面代码所展示的，使用一系列 RDDs 集合构建一个图。

import org.apache.spark.\_

import org.apache.spark.graphx.\_

//To make some of the examples work we will also need RDD

import org.apache.spark.rdd.RDD

rdd 9 0 in memory on localhost:41 0.0 in stage 2.0(TID 4).1582 by

16/06/2822:26:11 INFO Executor:Finished task

16/06/2822:26:11 INFO DAGScheduler:ResultStage 2(reduce at VertexRDDImpl.scal

(6s/.10IaaInfo:Added

//Assume the SparkContext has already been constructed

val sc:SparkContext

//Create an RDD for the vertices

val users:RDD[(VertexId,(String,String))]=

sc.parallelize(Array((3L,("rxin","student")),(7L,("jgonzal","postdoc")),

(5L,("franklin","prof")),(2L,("istoica","prof"))))

//Create an RDD for edges

val relationships:RDD[Edge[String]]=

sc.parallelize(Array(Edge(3L,7L,"collab"), Edge(5L,3L,"advisor"), Edge(2L,5L,"colleague"),Edge(5L,7L,"pi")))

//Define a default user in case there are relationship with missing user

val defaultUser=("Jie","Missing")

//Build the initial Graph

val graph=Graph(users,relationships,defaultUser)

在上面的实例中，我们用到了 Edge 样 本 类 。Edges 有一个 sreld 和 dstId 对应源顶点

和目的顶点标识符。除此之外，Edge 类有一个 attr 成员存储边属性。

使用 graph.vertices 和 graph.edges 解构出一个图对应的顶点和边。

val graph:Graph[(String,String),String]//Constructed from above

//Count all users which are postdocs

graph.vertices.filter{case(id,(name,pos))=>pos=="postdoc"}.count

//Count all the edges where src >dst

graph.edges.filter(e=>e.srcId >e.dstId).count

运行结果如图5-21和图5-22所示，统计图中属性为 postdoc 的节点个数如图5-21 所 示，统计图中源节点大于目的节点边的条数如图5-22所示。

Q⊙Jle@jie-Satellite-L850D:~

16/06/2822:26:11 INFO MemoryStore:ensureFreeSpace(1512)called with curMem=204

088(size:1824.0 B,free:265.1 MB)

14IFO Executor:Finished task 1.0 in stage 2.0(TID 5).1582 by

e602s8u2lt2:e61toIN ivremoryStore:Block rdd 9 0 stored as values in memory(es

i821. NO,fratger:Finished task 1.0 in stage 2.0(TID 5)in

114 ms on localhost(1/2)

02:1toINeaskSetManager:Finished task 0.0 in stage 2.0(TID 4)in

132 ms on localhost(2/2)

all completed,from pool

16/06/2822:26:11 INFO TaskSchedulerImpl:Removed TaskSet 2.8,whose tasks have

16/06/2822:26:11 INFO DAGScheduler:Job o finished:reduce at VertexRDDImpl.sca

a:90)finished in 0.132 s

ha:90,took

1.536573 s =1

res1:Long

图5-21 统计图中属性为 postdoc 的节点个数

6:cal(1/skSetManager:Finished task 0.0 in stage 3.0(TID 6)in

6:cal(2/skSetManager:Finished task 1.0 in stage 3.0(TID 7)in

&● ⊙jle@jle-Satellite-L850D:~

Ealhost,PROCESS\_LOCAL,1529 bytes)

16/06/2822:30:53 INFO TaskSetManager:Starting task 1.0 in stage 3.0(TID 7,lo

Ealhost,PROCESS\_LOCAL,1527 bytes)

16/06/2822:30:53 INFO Executor:Running task 0.0 in stage 3.0(TID 6)

6/06/2822:30:53 INFO BlockManager:Found block rdd 2 0 locally

16/06/2822:30:53 INFO Executor:Running task 1.0 in stage 3.0(TID 7)

16/06/2822:30:53 INFO BlockManager:Found block rdd 2 1 locally

16/06/2822:30:53 INFO Executor:Finished task 0.0 in stage 3.0(TID 6).1830 by

tes result sent to driver

16/06/2822:30:53 INFO Executor:Finished task 1.0 in stage 3.0(TID 7).1830 by

tes result sent to driver

16/06/2822:30:53 INFO TaskSchedulerImpl:Removed TaskSet 3.0,whose tasks have pll completed,from pool

16/06/2822:30:53 hed in 0.051 s 6/06/2822:30:53 0.089417 S

Fes2:Long =1

INFO

INFO

DAGScheduler:Resultstage 3(count

DAGScheduler:Job 1 finished:count

at <console>:37)finis

at <console>:37,took

al az

图5-22 统计图中源节点大于目的节点边的条数

注意：graph.vertices 返回一个 VertexRDD[(String,String)], 其扩展自RDD[(Ver-

texID,(String,String))], 这 样 我 们 可 以 使 用Scala case 表 达 式 来 解 构 元 组 。 另 外 ，graph.

edges 返 回 一 个 EdgeRDD 包 含 Edge[String] 对 象 。 我 们 也 可 以 使 用case 类 类 型 的 构 造 器 ，

如下所示 ：

graph.edges.filter{case Edge(src,dst,prop)=>src >dst }.count

除 了 属 性 图 的 顶 点 和 边 视 图 ，GraphX 也 提 供 了 一 个 triplet 视 图 。triplet 视 图 逻 辑 上

连接了顶点和边属性，产生一个 RDD[EdgeTriplet[VD,ED]], 其包含 EdgeTriplet 类。

join 可 以 表 达 成 下 面 SQL 表 达 式 ：

SELECT src.id,dst.id,src.attr,e.attr,dst.attr

FROM edges AS e LEFT JOIN vertices AS src,vertices AS dst

ON e.srcId=src.Id AND e.dstId =dst.Id

或者表示如图5-23所示。



Vertices: Edges: B Triplets B,

图5-23 Vertices 、Edges 和 Triplets 视图

EdgeTriplet 类 通 过 增 加 srcAttr 和 dstAttr 成 员 扩 展 了 Edge 类，它们包含源和目的顶 点 属 性 。 可 以 使 用 一 个 图 的 triplet 视 图 来 提 供 一 些 字 符 串 描 述 用 户 之 间 的 关 系 。

val graph:Graph[(String,String),String]//Constructed from above

//Use the triplets view to create an RDD of facts.

val facts:RDD[String]=

graph.triplets.map(triplet =>

16/06/2910:04:21 INFO BlockManager:Found block rdd 9 1 locally

triplet.srcAttr.\_ 1+"is

facts.collect.foreach(println(\_))

**5.3.2** **图操作**

the"+triplet.attr+"of"+triplet.dstAttr.\_ 1)

属性图包含 一 些基本的操作，这些操作采用用户自定义函数并产生转换属性和结构的 新图。在 Graph 中定义的核心操作是已经被优化的实现，组合核心操作的便捷操作定义在 GraphOps 中 。Scala 隐 式 转 换 在 GraphOps 中 的 操 作 可 在 Graph 的 成 员 中 自 动 使 用 。 例 如，统计图中所有节点的入度：

val graph:Graph[(String,String),String]

//Use the implicit Graph0ps.inDegrees operator

val inDegrees:VertexRDD[Int]=graph.inDegrees

inDegrees.foreach(println)

结果如图5 - 24所示。

Q● ⊙Jle@jle-Satelllte-L850D:~

16/06/2916:04:21 INFO Executor:Running task 0.8 in stage 6.0(TID 8)

16/06/2910:04:21 INFO Executor:Running task 1.0 in stage 6.0(TID 9)

16/06/2910:04:21 INFO ShuffleBlockFetcherIterator:Getting 2 non-empty blocks o

ut of 2 blocks

ih6/2910:04:21 INFO ShuffleBlockFetcherIterator:Started θ remote fetches in

h6/06/2910:04:21 INFO BlockManager:Found block rdd 9 θ locally

16/06/2910:04:21 INFO ShuffleBlockFetcherIterator:Getting 2 non-empty blocks o

ut of 2 blocks

h6/06/2916:84:21 INFO ShuffleBlockFetcherIterator:Started θ remote fetches in

316/06/2910:04:21 INFO Executor:Finished task 0.0 in stage 6.0(TID 8).20

88 bytes result sent to driver

(7,2)

56,/)6/2910:04:21 INFO Executor:Finished task 1.0 in stage 6.0(TID 9).2008 by

e6s/00:1toINeaskSetManager:Finished task 0.0 in stage 6.0(TID 8)in

23 ms on localhost(1/2)

23 ms on localhost(2/2)

16/06/2910:04:21 INFO TaskSetManager:Finished task 1.0 in stage 6.0(TID 9)in

16/06/2910:04:21 INFO TaskSchedulerImpl:Removed TaskSet 6.0,whose tasks have

图5-24 Graph 入度统计

**1.** **属性图函数操作** **一** **览**

以下是 一 个定义在 Graph 和 GraphOps 中的函数列表，更详细的内容可以参考官方 API, 官 方 网 址 ：<http://spark.apache.org/docs/latest/graphx-programming-guide.html>。

/\*\*Summary of the functionality in the property graph \*/

class Graph[VD,ED]{

//Information about the Graph

val val val val

numEdges:Long

numVertices:Long

inDegrees:VertexRDD[Int]

outDegrees:VertexRDD[Int]

val degrees:VertexRDD[Int]

>>

//Views of the graph as collections

val vertices:VertexRDD[VD]

val edges:EdgeRDD[ED]

val triplets:RDD[EdgeTriplet[VD,ED]]

//Functions for caching graphs ===============

def persist(newLevel:StorageLevel =StorageLevel.MEMORY\_ONLY):Graph[VD,ED] def cache():Graph[VD,ED]

defunpersistVertices(blocking:Boolean=true):Graph[VD,ED]

//Change the partitioning heuristic ==================

def partitionBy(partitionStrategy:PartitionStrategy):Graph[VD,ED]

//Transform vertex and edge attributes ==================

def mapVertices[VD2](map:(VertexID,VD)=>VD2):Graph[VD2,ED]

def mapEdges[ED2](map:Edge[ED]=>ED2):Graph[VD,ED2]

def mapEdges[ED2](map:(PartitionID,Iterator[Edge[ED]])=>Iterator[ED2]):Graph[VD,ED2] def mapTriplets[ED2](map:EdgeTriplet[VD,ED]=>ED2):Graph[VD,ED2]

def mapTriplets[ED2](map:(PartitionID,Iterator[EdgeTriplet[VD,ED]])=>Iterator[ED2])

:Graph[VD,ED2]

//Modify the graph structure ==

def reverse:Graph[VD,ED]

def subgraph(

epred:EdgeTriplet[VD,ED]=>Boolean=(x=>true),

vpred:(VertexID,VD)=>Boolean=((v,d)=>true))

:Graph[VD,ED]

def mask[VD2,ED2](other:Graph[VD2,ED2]):Graph[VD,ED]

def groupEdges(merge:(ED,ED)=>ED):Graph[VD,ED]

//Join RDDs with the graph === === ==

def joinVertices[u](table:RDD[(VertexID,U)])(mapFunc:(VertexID,VD,U)=>VD):Graph [vD,ED]

def outerJoinVertices[U,VD2](other:RDD[(VertexID,U)])

(mapFunc:(VertexID,VD,Option[u])=>VD2)

:Graph[VD2,ED]

//Aggregate information about adjacent triplets

def collectNeighborIds(edgeDirection:EdgeDirection):VertexRDD[Array[VertexID]]

def collectNeighbors(edgeDirection:EdgeDirection):VertexRDD[Array[(VertexID,VD)]]

def aggregateMessages[Msg:ClassTag](

sendMsg:EdgeContext[VD,ED,Msg]=>Unit,

mergeMsg:(Msg,Msg)=>Msg,

tripletFields:TripletFields=TripletFields.All)

:VertexRDD[A]

//Iterative graph-parallel computation =

def pregel[A](initialMsg:A,maxIterations:Int,activeDirection:EdgeDirection)( vprog:(VertexID,VD,A)=>VD,

sendMsg:EdgeTriplet[VD,ED]=>Iterator[(VertexID,A)],

mergeMsg:(A,A)=>A)

:Graph[VD,ED]

//Basic graph algorithms =

def pageRank(tol:Double,resetProb:Double=0.15):Graph[Double,Double]

def connectedComponents():Graph[VertexID,ED]

def triangleCount():Graph[Int,ED]

def stronglyConnectedComponents(numIter:Int):Graph[VertexID,ED]

**2.** **属性图操作**

属性图包括下面操作：

class Graph[VD,ED]{

def mapVertices[VD2](map:(VertexId,VD)=>VD2):Graph[VD2,ED]

def mapEdges[ED2](map:Edge[ED]=>ED2):Graph[VD,ED2]

def mapTriplets[ED2](map:EdgeTriplet[VD,ED]=>ED2):Graph[VD,ED2]

这里每一个操作产生一个新图，其顶点和边被用户定义的map 函数修改了。

注意：上述属性图操作不改变图的结构，这是图属性操作的关键特征，这也就意味着允 许结果图重复利用原始图的结构索引。下面的代码片段逻辑上是等同的，但是第一个没有 保存结构索引，其不会从 GraphX 系统优化中获益：

val newVertices=graph.vertices.map{case(id,attr)=>(id,mapUdf(id,attr))} val newGraph=Graph(newVertices,graph.edges)

使用 mapVertices 保护结构索引：

val newGraph=graph.mapVertices((id,attr)=>mapUdf(id,attr))

这些操作经常用来初始化图为了进行特殊计算或者排除不需要的属性。例如，给定一 个图，它的出度作为顶点属性(之后描述如何构建这样一个图),初始化它为 PageRank:

//Given a graph where the vertex property is the out degree

val inputGraph:Graph[Int,String]=

graph.outerJoinVertices(graph.outDegrees)((vid,\_,degOpt)=>degOpt.getOrElse(0)) //Construct a graph where each edge contains the weight

//and each vertex is the initial PageRank

val outputGraph:Graph[Double,Double]=

inputGraph.mapTriplets(triplet=>1.0/triplet.srcAttr).mapVertices((id,\_)=>1.0) 初始化结果如图5-25所示。

**3.** **结构操作**

当前，GraphX 仅仅支持一个简单的常用结构操作，将来会不断完善。下面是基本结构 操作列表：

class Graph[VD,ED]{

def reverse:Graph[VD,ED]

def subgraph(epred:EdgeTriplet[VD,ED]=>Boolean,

vpred:(VertexId,VD)=>Boolean):Graph[VD,ED]

第 5 章 Spark 在大数据处理中的应用

relationships:org.apache.spark.rdd.RDD[org.apache.spark.graphx.Edge[String]]= ParallelCollectlonRDD[1]at parallelize at <console>:28



def mask[VD2,ED2](other:Graph[VD2,ED2]):Graph[VD,ED]

def groupEdges(merge:(ED,ED)=>ED):Graph[VD,ED]

Q● jle@jle-Satelllte-L850D:~

s Edge(:e(3L,7L,

"pt")))

scala>val defaultuser =("jie","Nissing")

defaultuser:(String,String)=(jte,Missing)

4.g1rraph[(String,string),string]=org.apache.spark

scala>val graph =Graph(users,relationships,defaultuser)

,\_ θ=graph.outerJoinVertices(graph.outDe phspark.graphx.Graph[Int,String]=org.apache.spark.graphx. l>val oippcpAht:r\_bntGraph.maprriplets(trtplet pi.fh7eb.rk.graphx.Graph[Double,Double]=org.apache.spark.gra

图5-25 Graph 初始化为 PageRank

reverse 使所有边的方向反向，操作返回 一 个新图。反转操作没有修改顶点或者边属性 或者改变边数量，因而能够高效地实现反转。

subgraph 操作返回 一个新图，包含满足要求的顶点和边。 subgraph 操作可以用在限制 部分图顶点和边、删除损坏连接等场景。例如，在下面代码中，我们可以移除损坏连接：

//Create an RDD for the vertices

val users:RDD[(VertexId,(String,String))]=

sc.parallelize(Array((3L,("rxin","student")),(7L,("jgonzal","postdoc")),

(5L,("franklin","prof")),(2L,("istoica","prof")),

(4L,("peter","student"))))

//Create an RDD for edges

val relationships:RDD[Edge[String]]=

sc.parallelize(Array(Edge(3L,7L,"collab"), Edge(5L,3L,"advisor"),

Edge(2L,5L,"colleague"),Edge(5L,7L,"pi"),

Edge(4L,0L,"student"), Edge(5L,0L,"colleague")))

//Define a default user in case there are relationship with missing user

val defaultUser =("John Doe","Missing")

//Build the initial Graph

val graph=Graph(users,relationships,defaultUser)

//Notice that there is a user 0(for which we have no information)connected to users //4(peter)and 5(franklin).

graph.triplets.map(

triplet=>triplet.srcAttr.\_ 1+"is

).collect.foreach(println(\_))

//打印出节点之间的关系，如图5-26所示。

the"+triplet.attr+"of"+triplet.dstAttr.\_ 1

**大数据技术及行业应用**

//Remove missing vertices as well as the edges to connected to them

val validGraph=graph.subgraph(vpred=(id,attr)=>attr.\_2!="Missing") //The valid subgraph will disconnect users 4 and 5 by removing user 0

validGraph.vertices.collect.foreach(println(\_))

//打印所用节点的信息，如图5-27所示。

validGraph.triplets.map(

triplet=>triplet.srcAttr.\_ 1+"is the"+triplet.attr+"of"+triplet.dstAttr.\_ 1

).collect.foreach(println(\_))

//去除0号不存在节点后的有效子图，如图5-28所示

|  |
| --- |
| ×●**⊙jie@jie-Satellite-L850D:~**  **16/06/2916:22:30** **INFO** **ShuffleBlockFetcherIterator:Started** **0** **remote** **fetches** **in**  16/M0S6/2916:22:30 INFO Executor:Finished task 0.0 in stage 7.0(TID 14).2113 b ytes result sent to driver  6/06/2916:22:30 INFO TaskSetManager:Finished task 0.0 in stage 7.0(TID 14)t n 13 ms on localhost(1/2)  6/06/2916:22:30 INFO Executor:Finished task 1.0 in stage 7.0(TID 15).2115 b  ytes result sent to driver  16/06/2916:22:30 INFO TaskSetManager:Finished task 1.0 in stage 7.0(TID 15)i n 16 ms on localhost(2/2)  16/06/2916:22:30 INFO TaskSchedulerImpl:Removed TaskSet 7.0,whose tasks have all completed,from pool  16/06/2916:22:30 INFO DAGScheduler:Resultstage 7(collect at <console>:37)fin  ished in 0.816 S  16/06/2916:22:30 INFO DAGScheduler:Job 1 finished:collect at <console>:37,to  ok 0.229255 s  istoica is the colleague of franklin  rxin is the collab of jgonzal  franklin is the advisor of rxin  peter is the student of John Doe  franklin is the colleague of John Doe  franklin is the pi of jgonzal |

图5-26 节点之间的关系

|  |  |
| --- | --- |
| jle@jie-Satellite-L850D:~ | |
| 16/06/2916:24:47 INFO Executor:Running task 1.0 in stage 10.0(TID 17) | |
| 16/06/2916:24:47 INFO BlockManager:Found block  6/06/2916:24:47 INF0 Executor:Finished task 1.0  bytes result sent to driver  16/06/2916:24:47 INF0 TaskSetManager:Finished task  in 21 ms on localhost(1/2)  6/06/2916:24:47 INFO Executor:Finished task 0.0 | rdd\_47\_1 locally  in stage 10.0(TID 17).2047 1.0 in stage 10.0(TID 17)  in stage 10.0(TID 16).2001 |
| bytes result sent to driver  16/06/2916:24:47 INFO TaskSetManager:Finished task 0.0 in stage 10.0(TID [16)](#bookmark181)  in 31 ms on localhost(2/2)  16/06/2916:24:47 INFO DAGScheduler:Resultstage 10(collect at <console>:39)fi  nished in 0.022 s  16/06/2916:24:47 INF0 TaskSchedulerImpl:Removed TaskSet 10.0,whose tasks have  all completed,from pool  6/06/2916:24:47 INFO DAGScheduler:Job 2 finished:collect at <console>:39,to  ok 0.056871 s  (4,(peter,student))  (2,(istoica,prof))  (3,(rxin,student))  (7,(jgonzal,postdoc))  (5,(franklin,prof)) | |

图5-27 打印所有节点信息

第 5 章 Spark 在大数据处理中的应用

3 jle@jle-Satellite-L850D:~

6/06/2916:25:44 INFO ShuffleBlockFetcherIterator:Started θ remote fetches in

ms

6/06/2916:25:44 INFO ShuffleBlockFetcherIterator:Started 0 remote fetches in

ms

2u:2l:Odricutor:Finished task 0.0 in stage 14.0(TID 18).2113

6/06/2916:25:44 INFO Executor:Finished task 1.0 in stage 14.0(TID 19).2040

ytes result sent to driver

6/06/2916:25:44 INFO TaskSetManager:Finished task 0.0 in stage 14.0(TID 18)

n 78 ms on localhost(1/2)

6/06/2916:25:44 INFO TaskSetNanager:Finished task 1.0 in stage 14.0(TID 19)

n 78 ms on localhost(2/2)

6/06/2916:25:44 INFO TaskSchedulerImpl:Removed TaskSet 14.0,whose tasks have

all completed,from pool

6/06/2916:25:44 INFO DAGScheduler:Resultstage 14(collect at <console>:39)fi

6/06/2916:25:44 INFO DAGScheduler:Job 3 finished:collect at <console>:39,to

ished in 0.078 s

k 0.1234465

stoica is the colleague of franklin

xin is the collab of jgonzal

ranklin is the advisor of rxin

ranklin is the pi of jgonzal

图5-28 去除0号不存在节点后的有效子图

mask 操 作 通 过 返 回 原 图 中 部 分 节 点 和 边 ， 构 建 了 一 个 subgraph 。 与 subgraph 操 作 一

同使用能限制 一 个子图的返回。例如，将边数据中丢失的顶点去除：

//Run Connected Components

val ccGraph=graph.connectedComponents()//No longer contains missing field

//Remove missing vertices as well as the edges to connected to them

val validGraph=graph.subgraph(vpred=(id,attr)=>attr.\_2!="Missing")

//Restrict the answer to the valid subgraph

val validCCGraph=ccGraph.mask(validGraph)

validGraph.triplets.map(triplet=>triplet.srcAttr.\_ 1+"is the"+triplet.attr+"of "+

triplet.dstAttr.\_ 1).collect.foreach(println(\_))

返回子图结果如图5-29所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ● jle@jle-Satelllte-L850D:~ | | | | |
| 6/06/2916:55:37 INFO  ut of 2 blocks  6/06/2916:55:37 INFO pytes result sent to 6/06/2916:55:37 INFO  8 MS  6/06/2916:55:37 INF0 | ShuffleBlockFetcherIterator:Getting  Executor:Finished task 0.0 in  driver  ShuffleBlockFetcherIterator:Started  TaskSetManager:Finished task 0.0 | | | 2 non-empty blocks o stage 75.0(TID 44).2113 o remote fetches in  in stage 75.0(TID 44) |
| in 21 ms on localhost(1/2)  6/06/2916:55:37 INFO Executor:Finished task  pytes result sent to driver  16/06/2916:55:37 INFO TaskSetManager:Finished | | 1.0  task | in stage 75.0(TID 45).2040  1.0 in stage 75.0(TID 45) | |
| in 24 ms on localhost(2/2)  6/06/2916:55:37 INFO TaskSchedulerImpl:Removed TaskSet 75.0,whose tasks have  all completed,from pool  6/06/2916:55:37 INFO DAGScheduler:ResultStage 75(collect at <console>:39)fi  ished in 0.026 s  16/06/2916:55:37 INFO DAGScheduler:Job 7 finished:collect at <console>:39,to  bk 8.854691 S  istoica is the colleague of franklin  xin is the collab of jgonzal  franklin is the advisor of rxin  franklin is the pt of jgonzal | | | | |

图5-29 使用 mask 操作返回子图结果

groupEdges 操作能合并了多重图的并行边(例如，顶点之间的重复边)。在一些数值模

拟应用程序中，并行边能被融合，进而减少了图的大小。

**4.Join 操作**

在许多情况中，需要将外部数据集合(RDDs) 添加到图中。例如，有额外的用户属性， 想将其融合到一个已存在图中或者想把数据从一个图复制到另一个图。这些任务可以使用 join 操作来实现。下面我们列出了关键的 join 操作：

class Graph[VD,ED]{

def joinVertices[U](table:RDD[(VertexId,U)])(map:(VertexId,VD,U)=>VD) :Graph[VD,ED]

def outerJoinVertices[U,VD2](table:RDD[(VertexId,U)])(map:(VertexId,VD,Option[U])=>VD2) :Graph[VD2,ED]

joinVertices 操作连接 vertices 和输入 RDD,通过用户自定义的 map 函数处理图中的 节点属性，返回一个新图。若在 RDD 节点中没有一个匹配值，将保留其原始值。

可通过下面的方法加快 join 的执行速度：

val nonUniqueCosts:RDD[(VertexID,Double)]

val uniqueCosts:VertexRDD[Double]=

graph.vertices.aggregateUsingIndex(nonUnique,(a,b)=>a+b)

val joinedGraph=graph.joinVertices(uniqueCosts)(

(id,oldCost,extraCost)=>oldCost+extraCost)

outerJoinVertices 方法和joinVertices 相似，除了可以将用户定义的 map 函数应用到 所有顶点，还可以改变顶点类型。例如，我们可以通过使用出度初始化顶点属性来设置一个 针对 PageRank 的图：

val outDegrees:VertexRDD[Int]=graph.outDegrees

val degreeGraph=graph.outerJoinVertices(outDegrees){(id,oldAttr,outDegOpt)=> outDegOpt match {

case Some(outDeg)=>outDeg

case None=>0//No outDegree means zero outDegree

在上面的示例中使用到了柯里函数模式的多参数列表(例如f(a)(b)) 。 可以将写 f(a)(b) 与 f(a,b) 等同，使用柯里函数模式意味着类型接口b 将不会依赖于a 。 因此用户需 要提供自定义函数类型注释：

val joinedGraph=graph.joinVertices(uniqueCosts, (id:VertexID,oldCost:

Double,extraCost:Double)=>oldCost+extraCost)

**5.** **相邻聚合**

在图分析任务中一个关键步骤就是聚集每一个顶点的邻居信息。例如，我们想知道每 一个用户的追随者数量或者追随者的平均年龄。 一些迭代的图算法(像 PageRank, 最短路 径和联通组件)反复的聚集相邻顶点的属性(像当前 PageRank 值，源的最短路径，最小可到

达的顶点 ID) 。 为了改善原始聚集操作的性能，将 graph.mapReduceTriplets 改 为 新 的

graph.AggregateMessages。

(1)聚合消息

在 GraphX 中 核 心 的 聚 集 操 作 是 aggregateMessages, 它 提 供 了 一 个 用 户 定 义 的 sendMsg 函数到图中的每 一 个边 triplet,然 后 用 mergeMsg 函 数 在 目 的 节 点 聚 集 这 些

信息。

class Graph[VD,ED]{

def aggregateMessages[Msg:ClassTag](

sendMsg:EdgeContext[VD,ED,Msg]=>Unit,

mergeMsg:(Msg,Msg)=>Msg,

tripletFields:TripletFields=TripletFields.All)

:VertexRDD[Msg]

在下面的实例中，我们使用 aggregateMessages 操作来计算每 一个用户更年长追随者

的平均年龄。

//Import random graph generation library

import org.apache.spark.graphx.util.GraphGenerators

//Create a graph with "age"as the vertex property. Here we use a random graph for simplicity.

val graph:Graph[Double,Int]=

GraphGenerators.logNormalGraph(sc,numVertices =100).mapVertices((id,\_)=>id.toDouble) //Compute the number of older followers and their total age

val olderFollowers:VertexRDD[(Int,Double)]=graph.aggregateMessages[(Int,Double)]( triplet=>{//Map Function

if(triplet.srcAttr >triplet.dstAttr){

//Send message to destination vertex containing counter and age

triplet.sendToDst(1,triplet.srcAttr)

},

//Add counter and age

(a,b)=>(a.\_ 1+b.\_ 1,a.\_2+b.\_2)//Reduce Function

)

//Divide total age by number of older followers to get average age of older followers val avgAgeOfOlderFollowers:VertexRDD[Double]=

olderFollowers.mapValues((id,value)=>value

count})

//Display the results

avgAgeOf0lderFollowers.collect.foreach(println(\_)) 结果如图5-30所示。

match{case(count,totalAge)=>totalAge/

|  |
| --- |
| Q● ⊙jle@jie-Satellite-L850D:~  16/06/2920:28:31 INFO DAGScheduler:Job 9 finished:collect at <console>:36,to ok 0.413196 s  (34,69.5)  (52,77.33333333333333)  (4,51.47826086956522)  (16,62.095238095238095)  (82,90.0)  (66,81.4)  (28,64.05882352941177)  (54,80.41666666666667)  (80,93.6)  (30,69.7)  (14,65.85)  (50,76.71428571428571)  (36,69.53333333333333)  (24,64.94444444444444)  (64,88.88888888888889)  (74,80.33333333333333)  (90,93.5)  (72,81.875)  (70,83.77777777777777)  (18,56.75)  (12,44.0)  (38,72.375) |

图5-30 计算每个节点年长追随者的平均年龄

(2)Map Reduce Triplets Transition Guide(Legacy)

在以前的GraphX 版本中，我们计算邻居聚合使用 MapReduceTriplets 操 作 ：

class Graph[VD,ED]{

def mapReduceTriplets[Msg](

map:EdgeTriplet[VD,ED]=>Iterator[(VertexId,Msg)],

reduce:(Msg,Msg)=>Msg)

:VertexRDD[Msg]

MapReduceTriplets 操作应用用户定义的 map 函 数 到 每 一 个 triplet ,使用用户定义的

reduce 函数聚合产生 messages 。 返回迭代器是昂贵的，它抑制了我们应用额外优化(例如，

本地顶点的重新编号)的能力。在 aggregateMessages 中 我 们 引 进 了 EdgeContext, 其 显 示

triplet 属性，也明确了函数发送信息的源和目的顶点。除此之外，我们移除了字节码检测， 取而代之的是要求用户指明哪个 triplet 属性被需要。

下面的代码块使用 MapReduceTriplets:

val

def

graph:Graph[Int,Float]= …

msgFun(triplet:Triplet[Int,Float]):Iterator[(Int,String)]={

Iterator((triplet.dstId,"Hi"))

def reduceFun(a:Int,b:Int):Int=a+b

val result=graph.mapReduceTriplets[String](msgFun,reduceFun)

使 用 aggregateMessages 重写：

val graph:Graph[Int,Float]= …

def msgFun(triplet:EdgeContext[Int,Float,String]){

第 5 章 Spark 在大数据处理中的应用



triplet.sendToDst("Hi")

def reduceFun(a:Int,b:Int):Int=a+b

val result=graph.aggregateMessages[String](msgFun,reduceFun)

(3)计算度

一个普通的聚合任务是计算每个顶点的度。在有向图计算中，经常需要计算节点的入 度、出度和总度。GraphOps 类包含了一系列的度的计算方法。例如，在下面将计算最大入 度、出度和总度，如图5-31、图5-32和图5-33所示。

● ◎ jle@jle-Satelllte-L85oD:-

16/06/2920:54:03 INFO ShuffleBlockFetcherIterator:Started 0 remote fetches in l ms

:0O etoti2 l-l blocks o

9l:03 INFO ShuffleBlockFetcherIterator:Started 0 remote fetches in

task 1.0 in stage 89.0(TID 61).2227

16/06/2920:54:03 INFO Executor:Flnished

pytes result sent to driver

/2:0lct(SetManager:Finished task 1.0 in stage 89.0(TID 61)

0enNdricutor:Finished task 0.0 in stage 89.0(TID 60).2227

/2:0lcst(SetManager:Finished task 0.0 in stage 89.0(TID 60)

16/06/2920:54:03 INFO TaskSchedulerImpl:Removed TaskSet 89.0,whose tasks have

all completed,from pool

16/06/2920:54:03 INFO DAGScheduler:ResultStage 89(reduce at <console>:33)fin

shed in θ.830 s

h6/86/2920:54:03 INFO DAGScheduler:Job 10 finished:reduce at <console>:33,to

pk θ.117077 S

haxInDegree:(org.apache.spark.graphx.VertexId,Int)=(69,35)

图5-31 Graph 最大入度节点

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| &6⊙jie@jle- | Satellite | | -L85oD:~ | | |
| 16/06/2920:54:38 INFO ShuffleBlockFetcherIterator:Started 0 remote fetches in  0 ms  16/06/2920:54:38 INFO BlockManager:Found block rdd\_157\_0 locally  16/06/2920:54:38 INF0 ShuffleBlockFetcherIterator:Getting 2 non-empty blocks o  ut of 2 blocks  6/06/2920:54:38 INFO ShuffleBlockFetcherIterator:Started 0 remote fetches in  0 ms  6/06/2920:54:38 INF0 Executor:Finished task 1.0 in stage 94.0(TID 65).2227  bytes result sent to driver  16/06/2920:54:38 INF0 Executor:Finished task 0.0 in stage 94.0(TID 64).2227  bytes result sent to driver  16/06/2920:54:38 INFO TaskSetManager:Finished task 1.0 in stage 94.0(TID [65)](#bookmark182) | | | | | |
| in 17 ms on 16/06/2920:54:38 in 19 ms on 16/06/2920:54:38 | | localhost(1/2)  INF0 TaskSetManager:Finished task 0.0  localhost(2/2)  INF0 TaskSchedulerImpl:Removed TaskSet | | in stage  94.0,whose | 94.0(TID 64)  tasks have <console>:33)fin  <console>:33,to |
| all completed,from pool  16/06/2920:54:38 INFO DAGScheduler:Resultstage 94(reduce at  ished in 0.011 s  16/06/2920:54:38 INFO DAGScheduler:Job 11 finished:reduce at  ok 0.085733 s  maxoutDegree:(org.apache.spark.graphx.VertexId,Int)=(30,94) | | | | |

图5- 32 Graph 最大出度节点

Ojle@jle-Satellite-L850D:-

16/06/2920:54:38 INFO ShuffleBlockFetcherIterator:Getting 2 non-empty blocks o

6/2920:54:38 INFO ShuffleBlockFetcherIterator:Started 0 remote fetches in

16/06/2920:54:38 INFO BlockManager:Found block rdd\_157\_0 locally

ut of 2 blocks

16/06/2920:54:38 INFO ShuffleBlockFetcherIterator:Started 0 remote fetches in

/t0e64tIricutor:Finished task 0.0 in stage 94.0(TID 64).2227

60s6/2920:54:38 INFO Executor:Finished task 1.0 in stage 94.0(TID 65).2227

bytes result sent to driver

in 17 ms on localhost(1/2)

in 19 ms on localhost(2/2)

all completed,from pool

16/06/2920:54:38 INFO DAGScheduler:Resultstage 94(reduce at <console>:33)fin

16/06/2920:54:38 INFO TaskSetManager:Finished task 1.0 in stage 94.0(TID 65)

16/06/2920:54:38 INF0 TaskSetManager:Finished task 0.0 in stage 94.0(TID 64)

16/06/2920:54:38 INFO TaskSchedulerImpl:Removed TaskSet 94.0,whose tasks have

ished in 0.011 s

16/06/2920:54:38 INFO DAGScheduler:Job 11 finished:reduce at <console>:33,to

ok 0.085733 s

maxOutDegree:(org.apache.spark.graphx.VertexId,Int)=(30,94)

图5-33 Graph 最大总度节点

//Define a reduce operation to compute the highest degree vertex

def max(a:(VertexId,Int),b:(VertexId,Int)):(VertexId,Int)={

if(a.\_2 >b.\_2)a else b

//Compute the max degrees

val maxInDegree:(VertexId,Int) =graph.inDegrees.reduce(max)

val maxOutDegree:(VertexId,Int)=graph.outDegrees.reduce(max)

val maxDegrees:(VertexId,Int) =graph.degrees.reduce(max)

(4)收集邻居

在一些情形下，通过收集每一个顶点的邻居顶点和它的属性来表达计算是更加容易的，

通过使用 collectNeighborlds 和 collectNeighbors 操作。

class Graph0ps[VD,ED]{

def collectNeighborIds(edgeDirection:EdgeDirection):VertexRDD[Array[VertexId]]

def collectNeighbors(edgeDirection:EdgeDirection):VertexRDD[Array[(VertexId,VD)]]

这些操作代价比较高，由于复制信息和要求大量的通信，尽可能直接使用aggregateMessages

操作完成相同的计算。

**5.3.3** **构建图**

GraphX 提供了一些方法来构建一个图，从一个 RDD 的顶点和边或者硬盘上。默认情况 下，构建的图是没有分区的，而是留在默认的分区(像 HDFS 原 始 块 ) 。Graph.groupEdges

要求图重新分区，假定相同的边在同 一个分区，所以必须在调用 groupEdges 之 前 调 用

Graph.partitionBy

object GraphLoader{

def edgeListFile(

sc:SparkContext,

path:String,

canonicalOrientation;Boolean=false,

minEdgePartitions:Int=1)

:Graph[Int,Int]

GraphLoader.edgeListFile 提供了一种方式加载硬盘上边的列表。它解析下面的邻接

对(起始顶点ID 和目的顶点 ID) 列表，跳过#开始的行注释：

#This is a comment

21

41

12

它从指定的边创建一个图，自动创建边涉及的顶点。所有的顶点和边属性默认为1。

**5.3.4** **图计算相关算法**

GraphX 包含 一 系列的图算法来简化分析任务。算法被包含在 org.apache.spark. graphx.lib 包里面，能被 Graph 通 过 GraphOps 直接访问。下面将描述这些算法以及算法 如何使用。

**1.PageRank**

PageRank 测量在图中每一个顶点的重要性， 一条u 到 v 的 边 代 表u 对 v 重要性的 一个 支持。例如，如果一个 Twitter 用户被其他用户浏览，这个用户排名将会升高。

GraphX 自带了静态和动态的 PageRank 实现，作为 PageRank 对象的方法。静态的 PageRank 运行固定的迭代次数，然而动态的 PageRank 运行知道排名收敛(例如，超过设定 容忍值停止迭代)。GraphOps 允许直接调用这些算法作为Graph 的方法。

GraphX 也包含了一个社会网络数据集实例，我们可以在上面运行 PageRank 一 个用户 的集合在 graphx/data/users.txt 中给出，用户之间的关系在 graphx/data/followers.txt 中 给出。我们计算每一个用户的 PageRank 如 下 ：

//Load the edges as a graph

val graph=GraphLoader.edgeListFile(sc,"graphx/data/followers.txt")

//Run PageRank

val ranks =graph.pageRank(0.0001).vertices

//Join the ranks with the usernames

val users=sc.textFile("graphx/data/users.txt").map{line =>

val fields=line.split(",")

(fields(0).toLong,fields(1))

val ranksByUsername=users.join(ranks).map {

0.0

1.0 task task

case(id,(username,rank))=>(username,rank)

//Print the result

println(ranksByUsername.collect().mkString("\n"))

PageRank 算法运行结果如图5-34所示。

Q● ⊙jle@jle-Satellite-L850D:-

INFO

block block

16/06/2921:50:01 16/06/2921:50:01

BlockManager:Found BlockManager:Found

INFO

|  |  |
| --- | --- |
| 66/9y2t0:sultINnt txcruiinished  16/06/2921:50:01 INFO Executor:Finished  226 bytes result sent to driver | task task |

9:5 oNalnager:Finished 95 oNalnager:Finished

rdd\_1165\_1 locally

rdd\_1165\_0 locally

in stage 78488.0(TID 488).2

in stage 78488.0(TID 489).2

0.0 in stage 78488.0(TID 4

1.0 in stage 78488.0(TID 4

16/06/2921:50:01 INF0 TaskSchedulerImpl:Removed

ave all completed,from pool

16/i0: 0.0 DAGScheduler:Resultstage

16/86/2921:50:82 INFO DAGScheduler:Job 65

jstii15r,0s.15)

(matei\_zaharia,0.7013599933629602)

(ladygaga,1.390049198216498)

(Barackobama,1.4588814096664682)

(jerestg,0.9993442038507723)

(odersky,1.2973176314422592)

TaskSet 78488.0,whose tasks h

78488(collect at <console>:38)

finished:collect at <console>:38,t

图5 - 34 PageRank 算 法 运 行 结 果

**2.Connected Components**

连通图算法使用最小编号的顶点标记图的连通体。例如，在一个社会网络，连通图近似 聚类。GraphX 在 ConnectedComponents 对象中包含一个算法实现，我们计算连通图实例， 数据集和 PageRank 部分一样。

//Load the graph as in the PageRank example

val graph=GraphLoader.edgeListFile(sc,"graphx/data/followers.txt")

//Find the connected components

val cc=graph.connectedComponents().vertices

//Join the connected components with the usernames

val users=sc.textFile("graphx/data/users.txt").map{line

=>

val fields=line.split(",")

(fields(0).toLong,fields(1))

val ccByUsername =users.join(cc).map{

case(id,(username,cc))=>(username,cc)

//Print the result

println(ccByUsername.collect().mkString("\n"))

计算连通图如图5 - 35所示。

215 bytes result sent to driver

16/06/2921:52:37 INFO TaskSetManager:Finished task 0.0 in stage 78514.0(TID 5

08)in 6 ms on localhost(2/2)

16/06/2921:52:37 INFO Executor:Flnished task 1.0 in stage

○ ◎ jle@jie-Satellte-L850D:~

16/06/2921:52:37 INFO BlockManager:Found block rdd\_1219\_1

locally

l4y.0(TID 509).2

16/06/2921:52:37 INFO BlockManager:Found block rdd\_1219\_0

16/06/2921:52:37 INFO Executor:Finished task 0.0 in stage 78514.0(TID 508).2

16/06/2921:52:37 INFO TaskSetManager:Finished task 1.0 in stage 78514.0(TID 5

09)in 4 ms on localhost(1/2)

223 bytes result sent to driver

16/06/2921:52:37 INFO TaskSchedulerImpl:Removed TaskSet 78514.0,whose tasks h

ave all completed,from pool

16/06/2921:52:37 INFO DAGScheduler:Resultstage 78514(collect at <console>:38)

finished in 0.008 s

ook 0.043060 S

16/06/2921:52:37 INFO DAGScheduler:Job 68 finished:collect at <console>:38,t

|(bi\_it,3)

(ladygaga,1)

(Barackobama,1)

(jeresig,3)

(odersky,3)

图5-35 计算连通图结果

**3.Triangle Counting**

当顶点有两个邻接顶点并且它们之间有边相连，它就是三角形的 一 部分。GraphX 在 TriangleCount 对象中实现了 一 个三角形计数算法，其确定通过每 一 个顶点的三角形数量， 提供了一个集群的测量。我们计算社交网络三角形的数量，数据集同样使用PageRank 部 分 数 据集。注意：三角形数量要求边是标准方向(srcld<dstId), 图 使 用 Graph.partitionBy 进 行 分区。

//Load the edges in canonical order and partition the graph for triangle count

val graph=GraphLoader.edgeListFile(sc,"graphx/data/followers.txt",true).partitionBy(Par- titionStrategy.RandomVertexCut)

//Find the triangle count for each vertex

val triCounts=graph.triangleCount().vertices

//Join the triangle counts with the usernames

val users=sc.textFile("graphx/data/users.txt").map{line =>

val fields=line.split(",")

(fields(0).toLong,fields(1))

val triCountByUsername=users.join(triCounts).map {case(id,(username,tc))=>

(username,tc)

//Print the result

println(triCountByUsername.collect().mkString("\n"))

● ⊙jleojle-Satellite-L850D:~

16/06/2921:55:04 INFO BlockManager:Found block rdd\_1258\_0 locally

16/06/2921:55:84 INFO BlockManager:Found block rdd\_1278\_0 locally

16/06/2921:55:04 INFO ShuffleBlockFetcherIterator:Getting 2 non-empty blocks o

ut of 2 blocks

/2921:55:04 INFO ShuffleBlockFetcherIterator:Started θ remote fetches in

16/06/2921:55:04 INFO Executor:Finished task 0.0 in stage 78524.0(TID 526).2

218 bytes result sent to driver

16/06/2921:55:84 INFO TaskSetManager:Finished task 0.0 in stage 78524.0(TID 5 26)in 14 ms on localhost(2/2)

16/06/2921:55:04 INFO DAGScheduler:ResultStage 78524(collect at <console>:38)

11 5:0IF TaskSchedulerImpl:Removed TaskSet 78524.0,whose tasks h

ave all completed,from 16/06/2921:55:04 INFO

pool

DAGScheduler:Job

70 finished:collect at <console>:38,t

ook 0.247223 s

(justinbieber,θ)

(matei\_zaharia,1)

a,θ)

(jeresig,1)

(odersky,1)

图5-36 Triangle Counting 运行结果

**5.3.5** **GraphX 图计算实例**

假设我们想从 一 些文本文件构建 一 个图，约束图为重要的人际关系和用户，在子图运行 PageRank, 然 后 返 回 顶 点 用 户 相 关 的 属 性 。 我 们 使 用 GraphX 做 这 些 事 情 仅 仅 需 要 几 行 代 码 。

//Connect to the Spark cluster

val sc=new SparkContext("<spark://master.amplab.org>","research")

//Load my user data and parse into tuples of user id and attribute list

val users=(sc.textFile("graphx/data/users.txt")

.map(line=>line.split(",")).map(parts=>(parts.head.toLong,parts.tail)))

//Parse the edge data which is already in userId->userId format

val followerGraph=GraphLoader.edgeListFile(sc,"graphx/data/followers.txt")

//Attach the user attributes

val graph=followerGraph.outerJoinVertices(users){

case(uid,deg,Some(attrList))=>attrList

//Some users may not have attributes so we set them as empty

case(uid,deg,None)=>Array.empty[String]

//Restrict the graph to users with usernames and names

val subgraph=graph.subgraph(vpred=(vid,attr)=>attr.size==2)

//Compute the PageRank

val pagerankGraph =subgraph.pageRank(0.001)

//Get the attributes of the top pagerank users

val userInfoWithPageRank=subgraph.outerJoinVertices(pagerankGraph.vertices){

case(uid,attrList,Some(pr))=>(pr,attrList.toList)

case(uid,attrList,None)=>(0.0,attrList.toList)

println(userInfoWithPageRank.vertices.top(5)(Ordering.by( . 2.\_ 1)),mkString("\n"))

重 要 的 人 际 关 系 和 用 户 结 果 如 图 5 - 3 7 所 示 。

|  |  |
| --- | --- |
| Q● ⊙jle@jie-Satellite-L850D:- |  |
| 16/06/2922:13:05 INFO BlockManager:Found block rdd\_730\_1 | locally |
| /3nt txinished task 0.0 in stage | 33300.0(TID 308). |
| /3nt txinished task 1.0 in stage | 33300.0(TID 309). |
| :1 onal)anager:Finished task 1.0 in | stage 33300.0(TID |
| :1 onalhanager:Finished task 0.0 in | stage 33300.0(TID |

h6/06/2922:13:05 INFO TaskSchedulerImpl:Removed TaskSet 33300.0,whose tasks

ave all completed,from pool

h6/06/2922:13:05 INFO DAGScheduler:Resultstage 33300(top at <console>:41)fi

16/06/2922:13:05 INFO DAGScheduler:Job 38 finished:top at <console>:41,took

ished in 0.057 s

0.602425 s

(1,(1.453834747463902,List(Barackobana,Barack obama)??

(2,(1.3857595353443166,List(ladygaga,Goddess of Love)))

(7,(1.2892158818481694,List(odersky,Martin Odersky)))

(3,(0.9936187772892124,List(jeresig,John Resig)))

(6,(0.697916749785472,List(matei\_zaharia,Matei Zaharia)))

图5-37 重要的人际关系和用户结果



**本** **章** **小** **结**

本 章 介 绍 了Spark 小 型 集 群 搭 建 ， 使 用 Spark 集 群 对 一 个 5 0 万 节 点 的 仿 真 社 交 网 络 进 行 度 统 计 以 及GraphX 的 简 单 应 用 。 这 些 内 容 可 以 帮 助 大 数 据 初 学 者 熟 悉 如 何 使 用 Spark

这 个 强 有 力 的 工 具 进 行 相 关 工 程 实 战 和 学 术 研 究 。

**参** **考** **文** **献**

[1] <http://spark.apache.org/docs/latest/graphx-programming-guide.html>.

[2] <http://blog.csdn.net/mach>\_learn/article/details/46501351.

[3] 何 为 图 计 算 ：<http://www.csdn.net/article/2015-09-21/2825748>.

[4] 高彦杰.Spark 大数据处理：技术、应用与性能优化.北京：机械工业出版社，2014. [5] <http://blog.csdn.net/sunbow0/article/details/47612291>.

[6] <htps://endymecy.gitbooks.io/spark-programming> guide-zh-cn/content/graphx-programming-

guide/graph-operators,html.

[7] Lancichinetti,A.and S.Fortunato(2009)."Benchmarks for testing community de- tection algorithms on directed and weighted graphs with overlapping communities." Physical Review E Statistical Nonlinear &.Soft Matter Physics 80(1 Pt 2):145-148.

**第** **6** **章** **大数据技术在环境科学中的应用**

环境科学是一门研究环境的地理、物理、化学、生物四个部分的学科。它提供了综 合、定量和跨学科的方法来研究环境系统。随着科学技术的发展，目前的环境科学领域 的某些方面呈现着数据量大、数据分布广的一个趋势，因此借助了大数据的存储和大数 据的计算。如在大气环境方面，针对目前灰霾现象频发大气问题的研究，常用的方法是 对大气污染进行实时的观测和模拟，目前常用的大气模拟的模型是空气质量模型 (CAMx,CMAQ 等),与之相配套的气象模式常用的是MM5/WRF, 这些模型通过对某地 区进行模拟来分析问题的发生原因。大气污染物排放清单的建立也需要大数据平台的 支持进行数据的统计。随着环境问题逐渐得到重视，各地政府也设置了多个环境监测 站，全中国的环境监测站的数量巨大，按照每小时上传的数据量和数据类型来计算的，这 就需要用大数据平台进行云存储和云计算。



**6.1** **大气环境科学的数值模式的介绍**

随着计算机和科学技术的快速发展，以及大气动力学理论和数学物理方法的结合，近 30年来大气数值模式和模拟得到了迅速发展。大气环境领域的数值模式主要包括气象模 式和空气质量模式两大类。

**6.1.1** **气象模式**

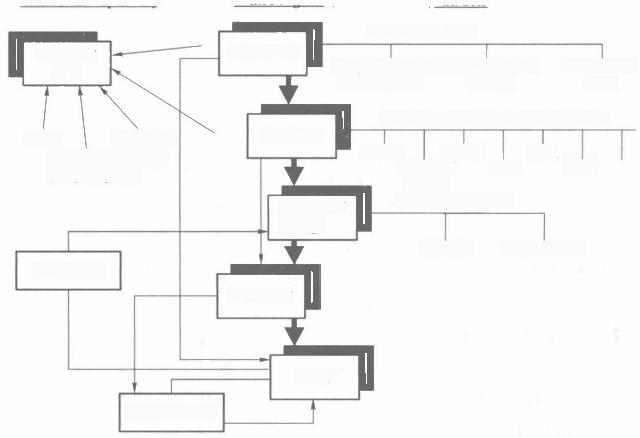
随着大气科学理论、数值计算方法和高性能计算机技术的不断发展，现代天气预报方法 已从传统的建立在大气定性理论、数理统计与预报员经验基础上的半经验方法，发展到以大 气科学理论为基础，综合现代科学技术的最新成果，通过高性能计算平台的模拟计算得到 预测结果的数值预报方法。数值天气预报有效地解决了过去预报产品不够丰富、可用预报 时效太短、各类气象探测资料综合处理能力不强等问题，已经成为气象部门制作天气预报的 重要基础和根本途径，具有其他预报方法不可替代的地位和作用。

气候数值模拟(The weather number imitate)可以概括为在实验室里一定的控制条件 下模拟自然界的气候状况，以及根据控制气候及其变化的基本物理定律，建立起相应的数学 模式，在一定的初始条件和边界条件下进行数值计算，求得气候及其变化的图像。气候主要 与相当长时间的现象有关，从几年到十几年或者更长的时间。随着计算机和数值计算方法 的发展，数值模拟已经成为定量研究气候及其变化的主要方法。

**1.MM5** **中尺度数值模型**

中尺度大气数值模式在20世纪80年代已有相当发展，进入90年代， 一些中尺度模式 和模拟系统已发展得相当先进并在世界范围内广为使用。如美国国家环境预报中心 (NCEP) 的业务预报中尺度模式(Eta) 、 科罗拉多州立大学(CSU) 的区域大气模拟系统 (RAMS) 、海军舰队数值气象和海洋中心(FNMOC) 的耦合海洋/大气中尺度预报系统 (COAMPS)、 英国气象局业务中尺度模式(UKMO)、 加拿大中尺度可压缩共有模式 (MC2) 、法国中尺度非静力模式(MESO-NH) 模式、日本区域谱模式(JRSM) 等。目前，国内 外流传较广、发展的较完善、最具有代表性的中尺度气象模式为美国国家大气研究中心 (NCAR) 和美国宾州大学(PSU) 在原有的流体静力模式 MM4 基础上发展的新一代中尺度 非流体静力模式 MM5 。MM5(Mesoscale Model 5)具有多重嵌套能力、非静力动力模式以 及四维同化的能力，并能在计算机平台上运行，来模拟或预报中尺度和区域尺度的大气环 流。特别是它的非流体静力模式，可以满足中—β(20～200 km) 和中—γ尺度(2～20 km) 强 对流天气系统演变的模拟需要。

MM5 模式结构可分为前处理模块(TERRAIN 、REGRID 、INTERPF 、LITTLE-R) 、 主 模块、后处理及绘图显示等辅助模块(包括 RIP 、GRAPH 、GrADS 、Vis5D)如图6- 1所示。 在每一部分中又有其具体细致的内容，前处理中包括资料预处理、质量控制、客观分析及 初始化，它为 MM5 模式运行准备输入资料，在气象模拟中的输入数据量十分大，而我们 日常使用的计算就在存储上产生困难，无法在短时间下载数据，而 MM5 的前处理模块每 次只能处理一个数据文件，这样在运用中我们就要借助基于大数据平台的云存储和云计 算，直接在“云”上进行前处理；MM5 主模块部分是模式所研究气象过程的主控程序；后 处理及绘图显示模块则对模式运行后的输出结果进行分析处理，包括诊断和图形输出、 解释和检验等。



*Additional Capanolity*

TERRESTRIAL

GRAPH/

RIP

SiB Landuse

GLIBAL/REGIONALANAL YSIS

NCEP NNRP ETA

ECMWF ERA AVN

TOGA

OBSERVATIONS

Surface Rawinsonde

INTERPB

INTERPF

MM5

NESTDOWN

MM5 RAWINS INTERPE/little\_r

NESTDOWN

Old,and USGS Terrain

Other LSM

Data

RAWINS/ little\_r

*Main Programs*

Old,USGS and

*Data Sets*

TERRAIN

REGRID

图 6 - 1 MM5 的 运 行 流 程 图

图6-1 中各模块具体功能如下。

(1)TERRAIN 。 选取模拟区域，生成水平网格，将地形和土地利用资料插值到格点上。 MM5 支持三种地形投影方式：Lambert 正形投影、极地平面投影和赤道平面Mercator 投影，这 三种投影方式分别适用于中纬度、高纬度和低纬度的模拟。TERRAIN 的输入参数包括模拟 区域的中心经、纬度，水平格距和网格数等。从图6-1中我们可以看出 TERRAIN 的主要数据 来源有三个分别是 Old.USGS and Sib Landuse 、Old and USGS Terrain 、Other LSM Date。

(2)REGRID 。 读取气压层上的气象分析资料，将大尺度经、纬度格点的气象、海温和 雪盖资料从原有的格点和地图投影上插值到由 TERRAIN 定义的格点和地图投影上。 REGRID 处理等压面和地面分析资料，并在这些层上进行两维插值。输出结果可作为客观 分析的第一猜值场，或作为分析场被直接插值到 MM5 的模式层上为 MM5 提供初始条件 和边界条件。作为输入的大尺度气象数据有两种来源， 一种是大尺度气象模式的实时预报 场，另一种是用历史观测资料同化得到的再分析气象数据。

(3)INTERPF 。 由于前面的分析都是在标准气压面上进行的，而 MM5 采用的是σ坐

标，因此需要 INTERPF 模块处理分析场和中尺度模式之间的数据转换。它包括垂直插值、 诊断分析并重新指定数据的格式。 MM5 的垂直格点在这一模块内进行定义，由输入参数 提供。INTERPF 将分析好的标准气压面上的数据插值到定义好的 MM5 的垂直格点上作 为初始场，同时生成侧边界条件以及下边界条件。

(4)MM5 主程序。读入 INTERPF 生成的初始条件和边界条件，投入运行根据动力学 方程完成时间积分。

(5)RAWINS/Little\_r 模块。将上一模块插值后的物理量场作为第一猜测场进行地面 和高空客观分析，在模式积分过程中进行四维资料同化，该模块还可以将预报模块(MM5) 的结果作为第一猜测场。

(6)GRAPH 。MM5 的附加程序可以对 MM5 的模拟数据进行绘图，

(7)INTERPB 。 可 将 MM5 模式结果插值到等压面，进而完成四维资料同化循环。

**2.WRF** **数值模型**

MM5 模式曾经是全球用户最多的中尺度大气模式，但由于其开发时间较早，动力学框 架陈旧，程序规范化、标准化程度不高， 一直未被美国最大的用户 NCEP 采 用 ；ETA 模式虽 然作为 NCEP 的业务预报模式，但难以及时吸收各所科研部门和大学的优秀研究成果，因 此其推广也受到限制。为了继承各个研究机构的最新研究成果，自1997年以来，美国多所 科研机构的科学家们共同研发了，业务与研究共用的新一代高分辨率中尺度预报模式 — — WRF 模 式(Weather Research and Forecasting Model,WRF Model)WRF(天气研究和预 报)系统的核心NMM (非静力中尺度模式)由NOAA/NCEP 发 展 而 成 。WRF 模式是 一种 完全可压非静力模式，采用 Arakawa C 网格，集数值天气预报、大气模拟及数据同化于一体 的模式系统，能够更好地改善对中尺度天气的模拟和预报，目前WRF 正逐步取代 MM5, 成 为区域天气研究和业务预报使用最广的气象模式。

WRF 处理分为两部分：前处理系统(WPS) 和后处理(WPP)。

WRF 的前处理系统(WPS) 用于实时的资料处理，功能包括：定义模拟区域；插值地形 资料(如地形、土表和土壤类型)到模拟区域；插值其他模式的资料到模拟区域和模式坐标。 WRF 的核心模块是 WRF-NMM 和 WRF-ARW。

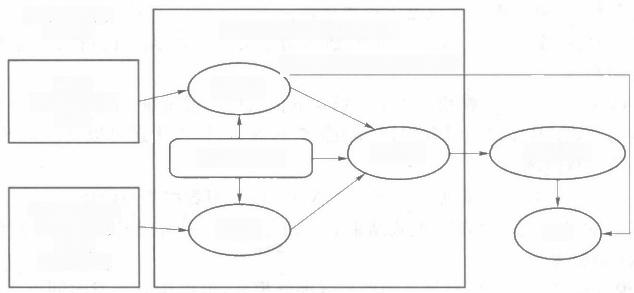
WRF-NMM 的主要特征：完全可压缩，具有静力选项的非静力模式；Arakawa E-网 格 ；

水平传播的快波采用向前向后方案，垂直传播的声波采用隐式方法，水平平流采用 Adams- Bashforth 方案，垂直平流采用Crank-Nicholson 方案；大量一二阶量的守恒，包括能量和位 涡拟能；多物理选项；单方向和双向嵌套。

WRF-ARW 模块的动力框架采用完全可压缩、非静力平衡欧拉模型，模型用具有守恒 性的变量的通量形式表示。水平方向上采用Arakawa C网络点，垂直方向则采用地形跟随 质量坐标，在时间积分方案上采用三阶或者四阶的Runge Kutte算法，对中央差分以及上风 平流方案都具有很好的稳定性，其稳定性时间步长是二阶式蛙跃式时间步长的2～3倍，节 省了运行时间。

WRF 的后处理(WPP) 。 这个子程序可以用来处理 WRF-ARW 和 WRF-NMM 预报， 设计如下：把模式垂直坐标差值到 NWS 标准输出层；把预报网格差值到正常网格；计算诊 断输出量；输出NWS 和 WMO 标准的GRIB1。

WRF 前处理系统(WPS) 是一个由三个程序组成的模块，这三个程序的作用是为真实 数据模拟准备输入场。三个程序的各自用途：geogrid 确定模式区域并把静态地形数据插值 到格点；ungrib 从 GRIB 格式的数据中提取气象要素场；metgird 则是把提取出的气象要素 场水平插值到由 geogrid 确定的网格点上。把气象要素场垂直方向插值到 WRF eta 层则是 WRF 模块中的 real 程序的工作。WRF 的前处理流程图，如图6-2所示。



WRF Preprocessing System

(static file(s)for nested runs)

geogrid

*Data*

namelist.wps metgrid real\_nmm

Gridded Data:

NAM,GFS,

RUC,

AGRMET,

External Data Sources

Static Geographical

ungrib

wrf

图6- 2 WRF 的前处理流程图

在图6-2中给出了数据在 WPS 的三个程序之间的转换关系。正如图像所示，WPS 里 每个程序都会从一个共同的 namelist 文件里读取参数。这个 namelist 文件按各个程序所 需参数的不同分成了三个各自的记录部分及一个共享部分，它们分别定义了 WPS 系统所 要用到的各种参数。

运行 WRF 的 Prepossessing System(WPS)有如下三个步骤：

(1)利用 geogrid 模块确定一个模式的粗糙区域(最外围的范围),及其他嵌套区域；

(2)利用 ungrib 把模拟期间所需的气象要素场从GRIB 资料集中提取出来；

(3)利用 metgird 把上述的气象要素场(第二步所做的工作)水平插值到模式区域(第 一步所做的工作)中。当多次模拟在同一区域重复进行时，只需要做一次第一步的工作即可 (也就是说 geogrid.exe 所做出的地形资料——geo\_em.d0\*.nc 可以重复利用);因此，只有 随时间改变的数据才需要在每次模拟时用第二、三步来处理。类似地，如果在多次模拟中， 气象输入数据是一样的，但是地形区域却不断改变的话，那第二步是可以省略的。

WRF-NMM 是 一 个 可 压 缩 非 静 力 的 中 尺 度 模 式 ( 有 静 力 选 项 ) 。 该 模 型 采 用 地 形 跟 随

第6章 大数据技术在环境科学中的应用

的 sigma 和 P 混合垂直坐标系，垂直方向上有18层，网格采用 Arakawa E-格点。所有过程 采用相同的时步，包括能量和涡度拟能的一些一阶和二阶量都是动量守恒的。WRF-NMM

代码包含初始化程序(real\_nmm.exe) 和数值积分程序(wrf.exe) 。WRF-NMM 第三版有 如下功能。

· 实时数据模拟

· 非静力的和静力的(运行选项)

· 全部物理过程选项

· 单向和双向嵌套

· 适用范围从几米到几千千米

下面举个 WRF 的应用案例。

在孙贞等人对青岛2006年8月的一次海风环流进行模拟，更清楚地找出了海风过程发 生发展的特征，研究了其经过测站的准确时间、风速风向变化、维持时段、转向方式等方面的 特征。为探讨海风环流的时空演变特性，采用WRF 三层嵌套模式对8月21日的海风环流 过程进行了高时空分辨率的数值模拟嵌套区域格距为15km 、5km 和1.67 km, 垂直方向 分为35层，在低层划分更细致 一 些采用的地形场分辨率为30 s, 中 心 在 青 岛 站 附 近 (36°N,120°E) 。 采 用NCEP AVN实况分析资料和预报场资料(分辨率1°×1°)作为初值 化的背景场和边界场，在输入过程中就需要从数据库将这些同步数据进行下载，以及在后 续模拟中需要将这些数据上传服务器进行模拟，这其中就涉及在日常监测中将监测站数据 同步上传，以及卫星等其他工具的监测数据的上传问题，原本借助单一的数据库在上传及下 载上比较麻烦，而现在在大数据的平台下数据的同步及下载是都将变得简单起来，模拟中加 入了常规的高空地面资料，由于大气环流背景相对稳定，采用显式水汽方案 KF 积云参数化 方案，YSUPBL 边界层方案、RRTM 辐射参数等方案，在模拟过程中现有 WRF 的前处理 系统进行前处理，而后WRF 的主模块进行模拟并将数据传到 WRF 的后处理模块进行可视 化的处理。

**6.1.2** **区域空气质量模式**

空气污染已经成为世界上许多城市面临的最严重的环境问题之一 ，强化大气环境管理， 防治空气污染及优化空气污染防治措施是城市环境保护的一项紧迫任务。随着我国工业、 交通和建筑业的蓬勃发展，以二氧化碳、氮氧化物和悬浮颗粒物为主的大气污染问题也日趋 严重，已经成为我国政府和社会共同面临的严峻问题。

目前国际上空气质量预报的方法有两种： 一种是以统计学方法为基础，利用现有数据基 于统计分析，研究大气环境的变化规律，建立大气污染浓度与气象参数间的统计预报模型， 来预测大气污染物浓度，称之为统计预报；另一种则是以大气动力学理论为基础，基于对大 气物理和化学过程的理解，建立大气污染浓度在空气中的输送扩散数值模型，借助计算机来 预报大气污染物浓度在空气中的动态分布，称之为数值预报。

空气质量模型是基于人类对大气物理和化学过程科学认识的基础上，运用气象学原理 及数学方法，从水平和垂直方向在大尺度范围内对空气质量进行仿真模拟，再现污染物在大 气中输送、反应、清除等过程的数学工具，是分析大气污染时空演变规律、内在机理、成因来 源、建立“污染减排”与“质量改善”间定量关系及推进我国环境规划和管理向定量化、精细化 过渡的重要技术方法。近年来，空气质量模拟技术发展迅速，相比其他环境要素的数学模拟

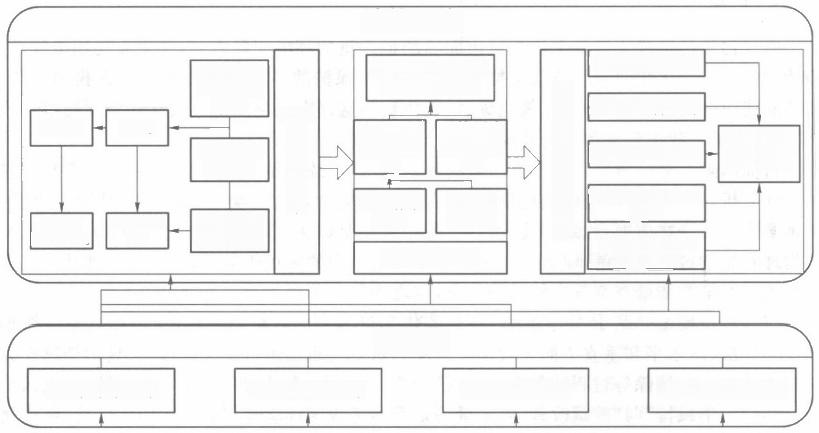
技术最为成熟，当前各种空气质量模型已被广泛应用于环境影响评价、重大科学研究及环境 管理与决策领域，已成为模拟臭氧、颗粒物、能见度、酸雨甚至气候变化等各种复杂空气质量 问题及研究区域复合型大气污染控制理论的重要手段之一，并发展成为一门学科方向。

**1.CMAQ**

CMAQ 模式是我国应用最广泛、最为成熟的第三代空气质量模型，由 USEPA 于1998 年第一次正式发布。CMAQ 最初设计的目的在于将复杂的空气污染问题如对流层的臭氧、 PM、毒化物、酸沉降及能见度等问题综合处理，为此 Models-3/CMAQ 模式最大的特色即 采用了“One-Atmosphere” 的设计理念，能对多种尺度、各种复杂的大气环境污染问题进行 系统模拟，CMAQ 模型目前已成为美国 EPA 应用于环境规划、管理及决策的准法规化模 型。该模型的特点在于：(1)可以同时模拟多种大气污染物，包括臭氧、PM、酸沉降以及能 见度等各种环境污染问题在不同空间尺度范围内的行为；(2)充分利用了最新的计算机硬件 和软件技术，如高性能计算、模块化设计、可视化技术等，使空气质量模拟技术更高效、更精 确，且应用领域趋于多元化。

Models-3/CMAQ 系统由排放清单处理模型(SMOKE) 、 中尺度气象模型(MM5 模型或 WRF 模型等)和通用多尺度空气质量模型(CMAQ) 三部分组成，其中CMAQ 是整个系统 的核心。CMAQ 模型主要由边界条件模块(BCON) 、初始条件模块(ICON) 、光解速率模块 (JPROC) 、气象—化学预处理模块(MCIP) 和化学输送模块(CCTM) 构成。CMAQ 模型的 关键部分是化学输送模块(CCTM), 污染物在大气中的扩散和输送过程、气相化学过程、气 溶胶化学过程、液相化学过程、云化学过程以及动力学过程都由该模块模拟完成。其他模块 的主要功能主要是为 CCTM 提供输入数据和相关参数。CCTM 模块提供了多种气相化学 机制和气溶胶化学机制供使用者选择，输出结果包括各种气态污染物和气溶胶组分在内的 污染物逐时浓度，以及逐时的能见度和干湿沉降。CMAQ 模式需要 MM5 或 WRF 气象模 式提供模拟所需的气象资料。CMAQ5.0 版本已实现气象模式与化学传输模式在线耦合， 吸收了 WRF-CHEM 模型部分优点。

CMAQ 的主要框架结构如图6-3所示。



**计算系统**

适用于空气质量

模型排放清单

BCON边界条件

本地与背

理

本地清单

处理

高时空分辨率

排放清单

数据库

WRF参数数据库 清单参数数据库

内置地形

土地利用

数据

再分析资 料下载

ADP地面 数据下载

天然源清

单自动生

成

背景清 单处理

MCIP 气象-化学

接口

cCrM 化学 传输模块

CMAQ参数数据库

C M A o 空 气 质 量 模 拟

JPROC 光解速率

ARW

MODEL

基本参数数据库

OBSGR

ID

ICON初始条件

排放清单处理

气象场模拟

REAL WPS

景融合处

WRF

图6- 3 CMAQ 的结构图

下面是CMAQ 的 一个应用案例。

珠江三角洲是我国一个经济快速发展的重要城市群地区，光化学烟雾污染是当地主要 的大气污染问题之一。对珠江三角洲地区臭氧污染的研究此前开展了许多工作，区域空气 质量模式在该地区臭氧污染的特征研究、控制与预报方面具有重要作用。在 Shen Jin 等 人 的研究中对珠江三角洲地区2004年10月的臭氧浓度进行模拟来分析污染物的来源。在 模拟中分为三个部分。首先对模拟地区的气象场进行模拟，在对气象场的模拟中首先要 进行数据的输入，在目前这个数据量大的时代气象场数据的输入量也越来越大其中包括 地质数据、气压层数据等，在这种情况下单个的计算机已经无法满足这种数据量的处理， 此时我们借助大数据平台的云存储进行数据的存储和统计，将大量的数据从专业的地质 网站、气象网站的数据库进行导出，在他们的研究中主要导出2004年10月模拟三层网 格区域的地质及气象数据，再将导出的数据输入到 CMAQ 模型气象模块(WRF) 中 地 区 的范围如图6-4所示。

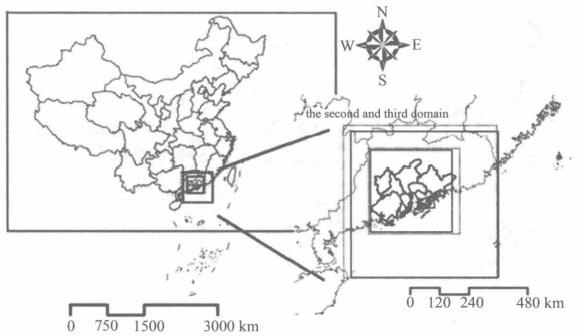


图6-4 模拟网格区域图

7501500

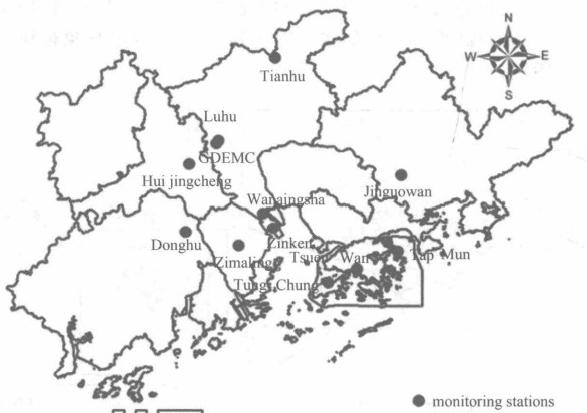
在图6-4中我们可以看出，模拟的第一层区域是包括中国在内的亚洲地区以36千米的 网格分辨率进行输入，第二层区域是包括珠江三角洲及周边城市的区域以12千米的网格分 辨率进行输入，第三层网格也就是珠江三角洲区域以4千米的网格分辨率进行输入，由此可 见在进行本次模拟中的气象场的模拟要输入的数据量之大是单个或几个计算机无法处理 的，数据的输入结束后，以 WRF 的气象模拟流程进行模拟，由于数据量之多，同样单台或多 台计算机进行模拟的计算是较为不便的，借助大数据的平台进行数据的云计算和云存储使 得 WRF 模拟具有更高效率。

第二部分进行 SMOKE 源排放清单的统计及计算，在该研究中第一重区域(36 km 网 格)的污染源输入数据来自 TRACE-P 源排放清单，能够满足为第二重嵌套网格提供边界条 件的模拟需要。第二、三层区域(12 km 与 4km 网格)的污染源输入主要依据广东省环境监 测中心和香港环保署提供的排放清单，在科学研究上，排放清单是大气污染模式重要的起始 输入数据，是研究空气污染物在大气中物理化学过程的先决条件。

第三部分进行 CMAQ 的空气质量模拟。CMAQ 的空气质量模拟是空气质量模拟的核 心模块，它主要将气象场数据和排放清单的数据整合，并通过对污染物的化学反应以及该污

染物在该气象条件下的传输转化进行模拟计算，在原来大量的数据基础上进行数值计算，可 见其计算量及数据处理量之大也是单台计算机无法承受的，大数据的云计算可帮助大大地 提高计算效率。

在模拟结果出来后，我们要将大量的数据进行可视化的一个处理，整合成方便查看 的污染源的 一 个分析图。模拟的部分结束后，我们要对此次模拟的 一 个准确性进行分 析，目前我们采取的分析方法一般是通过在模拟区域内观测站的数据采集，然后用模拟 时间段的观测站的采集数据与模拟数据中观测站位置的污染数据进行对比。观测站的 数据一般是每小时出一套数据，并将这些数据同步到数据库中进行存储。图6- 5 所示是 观测站位置的分布图。



64530 60 km □PRD

图6-5 观测站位置的分布图

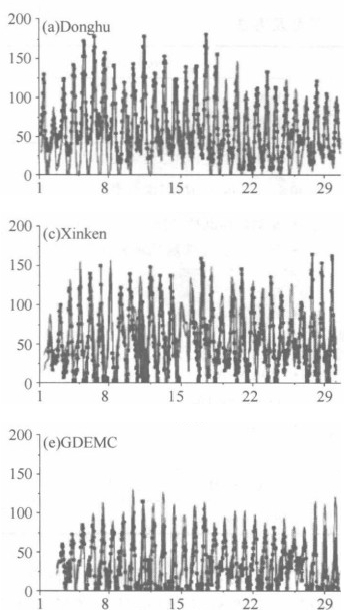
模拟性能的评估需要对数据库中大量的数据进行调用和统计，然后模拟数据进行对比

一般是绘制成折线图，图6-6是对珠江三角洲地区模拟数据的一个对比分析图。

图中红蓝为模拟数据，黑色为观测数据，由此图可以看出模拟数据与观测数据在发 展趋势上是相同的，只是在数值大小还存在一 定的误差，通过该对比我们可以判定本次 数据模拟大致是正确的，也就是我们可以应用该次模拟中得到的源解析结果，后处理得 到每种污染物来源的解析，如表6- 1所示是对珠江三角洲地区的源解析的分类及模拟数 据。通过模拟还可以获得各类污染源的排放比例，以及本地的贡献度等数据，来对当地的 环境政策的制定和环境污染的治理提供帮助。

**2.CAMx**

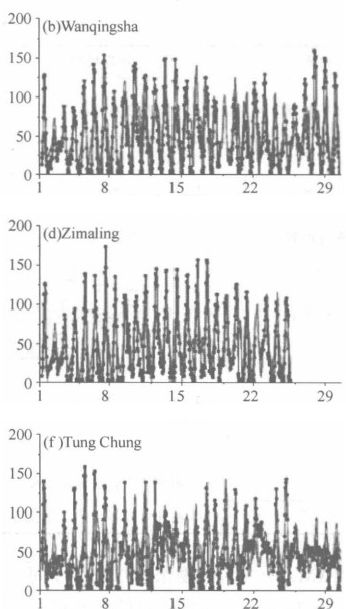
CAMx 模式是美国 ENVIRON 公司在 UAM-V 模式基础上开发的综合空气质量模式， 它将“科学级”的空气质量模型所需要的所有技术特征合成为单一系统，可用来对气态和颗 粒物态的大气污染物在城市和区域的多种尺度上进行综合性评估。CAMx 除具有第三代 空气质量模型的典型特征之外，CAMx 最著名的特点包括：双向嵌套及弹性嵌套、网格烟羽 (PiG) 模块、臭氧源分配技术(OSAT) 、 颗粒物源分配技术(PSAT) 等。

第6章 大数据技术在环境科学中的应用

臭氧浓度(10VN)

臭氧浓度(10°V/V)

臭氧浓度(10°V/V)

日期

日期

日期

日期

日期

日期

图6-6 对珠江三角洲地区模拟数据的对比分析线图

**表6-1** **各类污染源的排放量**

|  |  |  |  |
| --- | --- | --- | --- |
| 源类型 | 源类型 | 排放量/(10⁴t·a-1) | |
| NOx | VOCs |
| 大点源 | 火电点源和其他工业点源 | 68.1 | 10.8 |
| 流动源 | 汽油车和柴油车 | 51.6 | 77.1 |
| 天然源 | 植被排放 | 1.1 | 21.2 |
| 溶剂使用 | 油漆、建筑涂料、家用溶剂、印刷、石油精炼、加油站、溶剂储运 | 0.0 | 57.2 |
| 生活及废物处理 | 生活面源、废物焚烧 | 3.7 | 8.0 |
| 生物质燃烧与农业活动 | 生物质燃烧、农业活动 | 3.8 | 33.1 |
| 其他源 | 农用机械、运输车、工程机械、铁路、飞机、船舶 | 15.1 | 0.8 |

CAMx 可 以 在 三 种 笛 卡 儿 地 图 投 影 体 系 中 进 行 模 拟 ： 通 用 的 横 截 墨 卡 托 圆 柱 投 影 (Universal Transverse Mercator)、旋 转 的 极 地 立 体 投 影(Rotated Polar Stereo Graphic)和 兰伯特圆锥正形投影(Lambert Conic Conformal) 。CAMx 也 提 供 在 弯 曲 的 线 性 测 量 经 纬 度网格体系中运算的选项。此外，垂直分层结构是从外部定义的，所以各层高度可以定义为 任意的空间或时间的函数。这种在定义水平和垂直网格结构方面的灵活性，使CAMx 能 适 应任何用来为环境模型提供输入场的气象模型。

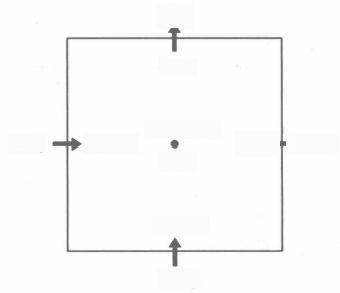


大数据技术及行业应用

**表6-2** **CAMx** **的模型及方法**

|  |  |  |
| --- | --- | --- |
| 过程 | 物理模型 | 数值方法 |
| 水平对流 | 欧拉连续性方程 | ·Bott  ·PPM |
| 水平扩散 | K理论 | 明确的同时发生的二维解决方式 |
| 垂直对流 | 欧拉连续性方程 | 绝对后向欧拉(时间)混合中心逆流(空间)解决方式 |
| 垂直扩散 | K理论或非局地混合 | ·绝对后向欧拉(时间)中心(空间)扩散  ·绝对ACM2非局地对流/扩散 |
| 气相化学 | ·Carbon Bond IV  ·Carbon Bond 2005  ·SAPRC99 | ·EBI  ·IEH  ·LSODE |
| 气溶胶化学 | 干湿有机无机化学，热力学，静电2 模式或多层剖面模型 | ·RADM-AQ  ·ISORROPIA  ·SOAP  ·CMI |
| 干沉降 | 气体和气溶胶各自的阻力模型 | 对于垂直扩散，沉降速度作为表面边界条件 |
| 湿沉降 | 气体和气溶胶各自的消除模型 | 指数式衰减 |

CAMx 中的污染物浓度都是处于每一个网格的中心的，由此代表着整个网格的平均浓 度。气象场提供给模型的目的是衡量每一个网格的大气状态，目的是计算传输和大气化学。 CAMx 内部是用 一种叫作 Arakawa C 的配置方式来运行这些变量的，如图6-7所示。



v,K,

*(i,J)*

*T,p,q,K,*

(i,)+ →u,K,

(i,j)

(i,j-1)

v,K,

u,K₇ (i-1,j)

图6-7 Arakawa C网格点

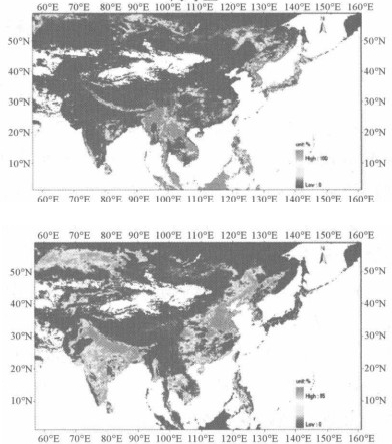
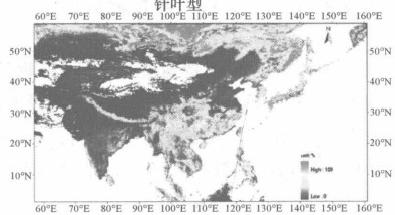
这些变量像温度、压力、水汽、云与污染物浓度 都位于栅格点的中心，代表着栅格点的平均状态条 件，风和扩散系数运行在栅格点的表面，用以描述 每一个栅格点物质的源和汇。

因为要依赖气象数据源，建议提供按照 Arakawa C的配置方式网格化的水平风场。垂直方向上，绝 大部分变量都是运行在每 一 高度层的中间位置。 当然也有例外的变量，如描述层间质量传输速率的 变量，包括垂直混合系数 K 。和垂直传输率η。这 些变量在水平位置上都在栅格点的中心运行，但在 垂直方向上都是运行于每 一层的层顶。敏感性分 析方法估算模拟结果对某一输入改变量的反应，例

如，由于硫氧化合物排放量的改变，引起的硫酸盐模拟浓度的变化。通常，如果模拟输入和 输出之间是非线性的，那么敏感性方法不会规定污染源的分配。例如，如果硫酸盐的形成与 硫氧化物的排放之间不存在线性关系时，那么对于所有的硫氧化物源，硫酸盐(SO₄) 的总量 不会等于模拟的总硫酸盐浓度。

下面是 CAMx 的应用实例。

在宁文涛等人对挥发性有机物的源解析中采用的是CAMx 模型进行模拟，对排放区域

的网格图如图6-8所示。

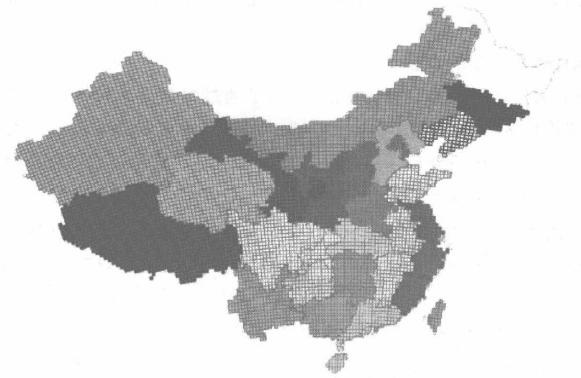


图6-8 对模拟区域的网格分布

在图中我们可以明显看出，在中国这个大的模拟区域的网格分布情况按网格的数据进 行计算，模拟数据的收集、导入、预处理、计算及统计的量都是相当大的，并且要与气象条件、 地形条件相匹配，在计算量上是很大的挑战，只有借助于大数据的存储计算平台，对数据的 统计计算才能实现，

模拟的结果如图6-9所示。

阔叶型

农作物

**灌木型**

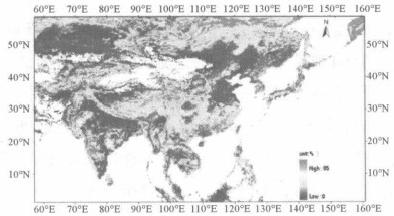


图6-9 对挥发性有机物模拟不同植被的排放情况

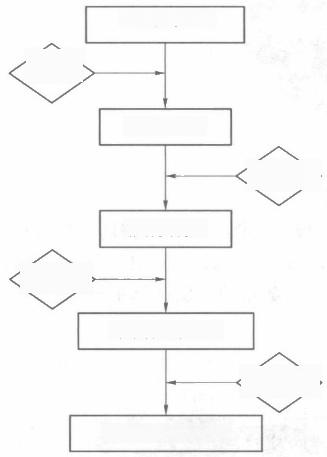
从模拟结果可以看出模拟的范围包括了亚洲的大部分，这样在数据计算结束后的一个

可视化处理上都需要大量的计算，模拟的分辨率越高可视化处理的数据量也就越大，对处理 能力的要求也就越高，大数据的运算能力和处理能力都为数据的模拟、可视化提供了优越的 条件。

**3.SMOKE** **排放清单的建立**

排放清单的大气污染物排放源清单的开发，是通过对某一地区一种或几种污染物排放 量的估算，了解该地区污染源排放特征及不同污染源对大气污染的贡献。大气污染物排放 源清单是利用空气质量模型分析大气污染物在大气中物理化学过程的特征及模拟不同污染

减排效果环境效应的基础输入数据。



农作物年产量

谷草比

秸秆年产量

缺烧比例

秸秆焚烧量

排放因子

污染物排放总量

星火点

秸秆焚烧点源信息

图6-10 排放清单建立的流程

编制过程：以某年为基准年，收集该地区各企业 的燃料消耗量、产品类型、产品产量、地理位置经纬 度等相关信息以及人口、能源消耗、经济、机动车、道 路、土地使用、植被等有关的社会、经济、生活的信息 和统计资料。利用各类源的排放因子和活动水平数 据，编制主要大气污染物源排放清单。建立主要与 需要调查的污染物相关，不同的污染物的来源不同， 因此在建立排放清单时候，先要根据要调查的污染 物来找出该污染物的可能来源，然后进行计算。例 如在朱等人研究的长江三角洲地区秸秆焚烧污染物 的排放清单时，污染物的来源很明确，就是秸秆的焚

烧，所以该研究的第一步就是要对秸秆的量进行统 计，图6- 10是对秸秆焚烧污染物排放清单的 一 个 流程。

由该流程图我们可以清楚地看出，对污染物的 排放清单的建立首先要对污染物的来源进行研究，

然后对污染源进行统计，上面的研究中只有一个污染源，而往往在大多数的研究中污染源的 种类是几个甚至更多，在这种情况下对数据的统计就需要借助大数据平台对污染物的来源 进行统计，然后按照研究人员的需要根据统计数据，计算出来做出排放清单。

例如在余宇帆等人的珠江三角洲地区重点 VOC 排放行业的排放清单的研究中，污染 源要基于珠三角地区本地排放情况。珠三角关注的重点 VOC 排放行业包括：家具制造业、 玩具制造业、印制电路板制造业、印刷业、制鞋业、船舶制造业、汽车制造业、涂料及油墨生产 行业、炼油与石化行业、加油站、建筑涂料使用、家用溶剂使用。数据的来源主要有四种： 一 是来自统计年鉴以及统计信息网上公布的信息，如玩具制造业、家用溶剂使用等；二是通过 对行业数据进行调研获得，如加油站汽油销售量、建筑涂料使用量等；三是通过调研珠三角 各城市政府网站获取公开数据，如印刷业、家具制造业、制鞋业；四是通过对生产企业进行调 研获得，如汽车与船舶制造业、炼油与石化行业等。目前研究中的数据主要来源是前三种， 前三种也是依靠大数据来挖掘有效信息，在庞大的数据库里面依靠“云”来进行数据的挖掘 和统计，得到我们所需要的信息，然后根据不同行业的排放因子计算出它的污染物的排放

量，从而得到污染物的一个排放清单。图6-11 是余等人研究得到的不同行业排放所占比

例图。

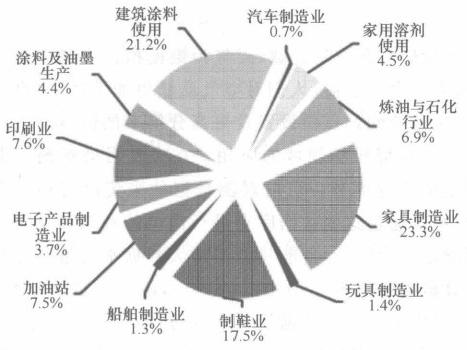


图6-11 2006年珠江三角洲地区重点挥发性有机物

排放行业 VOC 排放分担率

通过该图我们可以清楚地看到各行业在挥发性有机物排放中所占的比例，为政府和相

关部分出台相关的政策提供了帮助。



**6.2** **高分辨率实时观测的大数据**

污染物浓度数据的观测统计是进行大气环境科学研究的重要一部分，不仅是前期进行 模拟的背景输入，而且也是后期模拟的校验所需。要完成一个准确度高的模拟数据条件便 成为一个重要因素。

目前中国政府是对 SO₂ 、NO₂,PM10,PM2.5,CO 和 O₃ 进行观测并进行实时的发布，

发布内容包括空气污染物指数、首要空气污染物、空气质量级别和空气质量状况。随着社会 的进步和人们对大气环境的重视，目前大气环境的观测点的数量也在迅速的增加。另外随 着环保法污染物的排放监测也越加严密。大气污染和排放源的实时观测数据量和数据类型 越来越多，这些都对数据的采集和处理能力提出更高的要求。

来自各观测站高分辨率的实时观测数据具有数据量大、数据密度高、数据来源多的特 点，各个不同观测点的数据在同步过程中单位时间的数据量大，数据的采集依靠多个数据库 来完成，存储过程中数据的同步都是依靠大数据的平台来实现的。在丁超等人采用了高分 辨率观测数据对西安的PM2.5 及含碳气溶胶特性进行了研究。高分辨率数据也就具有了 很好的研究价值，为大气环境科学的研究提供了十分重要的基础数据。



**本** **章** **小** **结**

在大气环境科学的研究和应用领域，大气污染模拟技术的应用及监测数据采集均涉 及大量数据，并且随着社会的发展，人们对空气质量更加关注，更高分辨率的动态污染排 放清单数据，更加密集的监测站点位设置及更高分辨率的区域空气质量模拟均促进了大 数据科学与技术在大气环境科学领域的应用。在大气污染观测数据的收集和模拟是大 气模拟的一个重要组成部分部分。目前观测数据的采集已有利用多个数据库采集监测 站端口的数据，用户可以通过这些数据库来进行简单的查询和处理工作。在数据采集过 程中数据同步的问题也尤为突出，目前只能借助大数据平台进行处理。对数据进行导入 和预处理也是统计分析和数值模拟中重要的一环，随着模拟准确性的提高，模拟的数据 量大，导入与预处理过程中导入的数据量常达到每秒百兆，甚至千兆级别，这通常是单个 计算机或是小型服务器无法满足的，在此基础上利用大数据的来进行处理是非常必要 的。另外，在大气模拟的发展进程中模拟数据量大和模拟分辨率高的特点也日趋显著, 这对计算和统计分析能力的要求也有所提高，借助于大数据的计算与分析主要利用分布 式数据库或者分布式计算集群来对存储的海量数据进行快速的分析和分类汇总。大数 据技术和平台逐步成为大气污染观测数据采集与大气污染数值模拟的研究的重要科技 支持，具有长远意义。

**参** **考** **文** **献**

[1] Georg A.Grell,Jimy Dudhia David R.Staffer,A Description of the Fifth Generation

Penn State/NCAR mesoscale Model(MM5).NCAR Technical Notes,2004. [2] MM5 Version 3 Tutorial Presentations,NCAR Technical Notes,2004.

[3] Liu Xiang,Jiang Guorong,Zhuo Haifeng.Numerical experiment for the impact of

SST to typhoon“Chanchu”[J].Marine Fore-casts,2009,26(3):1-11.

[4] Skamarock W C,Klemp J B,Dudhia J,et al.A description of the advanced research WRF Version 3[Z].Ncar Technical Note,NCAR/TN-475+STR,2008.

[5] 孙贞，高荣珍.张进，徐晓亮，盛春岩.青岛地区8月一次海风环流实例分析和 WRF 模拟，METEOROLOGICAL MONTHLY,Vol.35 No.8August,2009.

[6] 洪钟祥，胡非.大气污染预测的理论和方法研究进展[J]. 气候与环境研究，1999, 4(3):225.

[7 ] 王自发，谢付莹，王喜全，等.嵌套网格空气质量预报模式系统的发展与应用[J]. 大气 科学，2006,30(5):779.

[8] 薛文博，王金南，杨金，等.国内外空气质量模型研究进展，环境与可持续发展.

第6章 大数据技术在环境科学中的应用 

[9] Chan CK,Yao X.Air pollution in mega cities in China.Atmos Environ,2008,42:

1-42.

[10] SHEN Jin,WANG XueSong\*,LI JinFeng,LI YunPeng &.ZHANG Yuan Hang, Evaluation and intercomparison of ozone simulations by Models-3/CMAQ and CAMx over the Pearl River Delta,SCIENCE CHINA Chemistry,November 2011 Vol.54

No.11:1789-1803.

[11] Streets DG,Bond TC,Carmichael GR,Fernandes SD,Fu Q,He D,Klimont Z, Nelson SM,Tsai NY,Wang MQ,Woo JH,Yarber KF.An inventory of gaseous and primary aerosol emissions in Asia in the year 2000.J Geophys Res,2003,108 (D21):8809.

[12] 赵斌，马建中.天津市大气污染源排放清单的建立[J]. 环境科学学报，2008,28(2):

368-375.

[13] 宁文涛，赵善论.天然源 VOC 的排放量估算和对区域空气质量影响的研究.山东师

范大学，2012年博士论文.

[14] 朱佳雷，王体健，邓君俊，等.长三角地区秸秆焚烧污染物排放清单及其在重霾污染天 气模拟中的应用[J]. 环境科学学报，32(12):3045-3055.

[15] 余宇帆，卢清，郑君瑜，等.珠江三角洲地区重点 VOC 排版行业的排放清单[J]. 中

国环境科学，2011,31(2):195-201.

[16] 丁超，张承中.基于高分辨率观测数据对西安市 PM2.5 及其碳气溶胶的污染特性研 究.西安建筑科技大学，2012年博士论文.

**第7章** **大数据在** **DrugBank 药物** **数据库聚类方面的应用**

本章将结合大数据思维对 DrugBank 中的西药进行聚类分析。该案例使用了 Java 语 言和 R 语言，结合近年来学术界对药物聚类的研究成果，集成化学分子、靶标和作用酶的相 似性对药物之间的相似度提出新的度量标准，并采用算法对药物进行聚类，该算法是我们自 己研究出来的一种新型算法，稍后会有详细介绍。

随着科学的发展，各个领域的知识相互渗透，对分子的研究不再局限于在实验室中看不 见的分子，计算机科学的发展对这个问题提供了极大的便利，分子的结构被描述成可在计算 机上存储的格式，比如分子结构的2-D、3-D 形式，用编码表示、分子式表示等，有了这些描 述，就可以将分子的结构分解成纯数学形式的元素进行整合分析，将具有相似性质的分子划 分为一类的行为成为聚类，再对聚类的结果进行分析的过程就成为聚类分析。近年来分子 聚类的算法被提出许多，这些算法的依据分别有原子对、指纹集、分子结构、分子量等，但综 合这些算法能得到一个共同点，即都是对分子结构方面的聚类分析，这样纯数学的物理聚类 分析方法对生物活性的预测和聚类不够准确，新的算法急需被提出来弥补这一缺憾。本章 将介绍一个分析生物活性的算法，该算法针对分子的靶标、作用酶和原子的百分比等方面对 分子进行了分析，在分子的预测和新药的研制上都有着启发性的作用。



**7.1** **简** **介**

当前市场上的药物基本分为中药、西药和中成药三类，中药一般是对中草药做处理得到 的汤药，根据个体差异调节用药，针对性较强， 一般是熬制成药服用，部分中药的熬制过程比 较复杂，其成分也十分复杂。中成药是以中草药为原料，通过一些工艺流程处理成不同剂型 的中药制品，相比中药来说免去了煎药的烦琐过程，比较方便携带和使用，效果和中药相似， 但可能添加有西药的成分，成分复杂程度和中药类似。西药一般用化学合成方法制造或是 直接从天然产物中提炼形成，成分比较单一，具有明确的分子式和分子结构，可以对其分析 研究，对小分子的聚类和分析十分方便，随着基因、蛋白质、遗传变异、化学化合物、疾病和药 物数据源的激素增长，这些数据源之间的集成和识别具有很高的利益可寻。

西医在研究基因、化合物的差异和联系方面的技术越来越成熟，随着信息量逐渐增大， 算法越来越精确，得到的结果越来越清晰，生物医学的基础和应用研究吸引了很多高通量技 术的研究，比如cDNA 或寡核苷酸的微阵列实验被用来识别基因表达差异高度并行的方 式，在基因研究方面取得了很大的成就。

第7章 大数据在 DrugBank 药物数据库聚类方面的应用

西药在当今快节奏的社会中优势极大，相比中药来讲，西药使用方便，效果明显，这 个趋势对药物的研制有促进作用，另外有些进口药物或者不在国内没有销售市场的药 物，虽然有着很好的疗效，但售价高，买不到导致病情不能及时治疗，错过最佳治疗时间。 聚类算法的核心是指针对具有相似的特征将其划分为一类，聚类结果的好坏程度可以从 多方面进行检验。在西方的研究当中，生物、化学、生物与化学，聚类等方面都做有研究， 而且研究价值也十分显著,不仅有对聚类的专门研究，提出了很多行之有效的算法，还发 布了含有西药分子详细信息的化学空间—DrugBank, 对生物小分子方面的聚类的 Chem- mine tools、化合物的数据库 PubChem、为新型化合物作研究的 ChemBank、还有对化学物 和生物学结合分析的Chem2bio2rdf 等一系列可供科学家等人研究的便捷的网站，这些网 站直接搜索对应的名字即可找到，特别是Chem2bio2drf, 其中链接了PubChem 数据库，对 药物的靶向抑制通路、共同作用的靶细胞等做了研究，是结合化学和生物共同特点而做 出的分析，是目前国内外少有的综合性质的网站，但分析的对象比较单一，不针对聚类 分析。

本章主要对 DrugBank 中的药物进行分析。DrugBank 是一个丰富的注解资源，综合了 现有基本的药物数据和完善的各种药物涵盖的信息。自从2006年的第一个版本，Drug- Bank 已被药师，药物化学家，医药研究人员，医生，教育工作者和广大公众广泛使用。并且 广泛用于硅片药物靶点发现，药物设计，药物对接或筛选，以方便药物代谢预测，药物相互作 用预测及一般药物教育。DrugBank 包含小分子和大分子两个药物的命名法，本体、化学、 结构、功能、动作、药理学、药物动力学。代谢和药物特性的大量数据。它还包含对目标疾 病、蛋白质、基因和生物对这些药物作用的综合信息。在2008年这一次更新中，DrugBank 增加了很多新的药物，新的药物和药物的各种基本信息被加入该数据库的字段中，这些新增 功能的数据字段包括说明药物的作用途径、药物转运数据、药物代谢数据、药物不良反应数 据、ADMET 数据，计算性能数据和化学分类数据等。

在分子研究领域中虽然有不少有效且使用方便的算法，但大多数是针对分子结构的分 析，例如 Thomas Girke等人在2008年发表的期刊中提出的基于最大公共子结构的算法， Jean-Louis Reymond 等人在2012年发表的期刊中提出的分子量子数的算法，主要针对的 是分子的化学特性和物理特性的而分析，是属于纯数学的物理分析聚类方法，没有考虑到分 子的生物活性方面，所以虽然算法多，使用方便，但对分子活性的预测效果并不显著,我们在 研究了各算法后，找到了新的聚类角度，于是我们提出了新的聚类算法，该算法是针对 DrugBank 中分子的靶标和作用酶，以及分子中各原子所占百分比的分析处理，不仅对分子 的生物活性方面做出了分析，还结合了分子的物理特性一分子的质量组成，是一个较全面的 聚类方法。

经过相关资料的查找和验证，分子对应的靶标和作用酶对分子的生活活性有着直接的 影响。ALTIERIDC 在2008年发表的一篇期刊中解释说明了 Survivin 这一靶标对癌细胞 活性的影响以及中国科学院的江寿平对酶和分子之间的关系作了详细的实验分析，结果都 表明这两因素对分子活性有直接的影响。通过对靶标和作用酶的聚类分析对分子活性的分 析更加具有一定的合理性和现实意义性。药物靶标是指自身具有药效作用并能被药物作用 的生物大分子，靶标具有两面性：有效的和反作用性。有效性能通过调节靶标的生理活性有 效地改善病症，如果对靶标的生理活性的调节没有达到事先预计的效果甚至是相反的效果，

则将其选作药物作用靶标是不合适的，靶标的发现和确认需要严谨的流程以确保准确性，事 先确定靶向特定疾病的靶标分子是现代新药开发的基础。酶是指具有生物催化功能的高分 子物质，酶的特性有催化作用、专一性、不参与反应、条件比较温和、活性可调节、催化性和辅 因子又关、易变形等特性，在本章中主要对其中的催化作用进行分析。

目前市场上药物种类繁多，中药的研究过程十分烦琐，且在很多方面有着局限性和不 足，西药发展的虽然十分迅速，效果显著,但仍然有不能根治疾病的药物，这类药物在顽固性 疾病中所占的分量还不小，有些药物有不错的治疗效果，但不在国内市场销售，这样就导致 了有些人本来能治好的病，因为国内市场没有或者要价太高负担不起而错过了治疗时间，这 是十分遗憾的事情，所以自己研制新药和制造有类似疗效的药物是十分必要的，这就为分析 分子的聚类算法提供了发展的空间，目前国内外的分子分析的聚类算法大多数是纯数学的 物理分析方法，对分子活性的预测准确度不够，基于这点我们设计了专门针对生物活性的算 法，该算法对 DrugBank 中现存的分子库中的分子进行了分析，聚类基准是分子的靶标、作 用酶和分子中原子所占的百分比，其聚类结果展示的是具有相同活性的分子的不同区域的 划分。

参考和对比国内外相关的算法，可以发现现有的大部分算法只是在分子的组成结构方 面进行了分析，但是通过大量的网络、期刊查阅并且请教多位相关专家，最终发现影响分子 活性的不只是分子的结构特征，也与结构和分子的性质有关，但主要关系在于物理性质和化 学性质，生物活性从分子的分子结构中很难看到，我们经过多次讨论后提出了一种新的分析 方法，将分子的化学性质、物理性质、生物活性综合在一起进行分析，但侧重点是放在分子的 生物活性方面，因为分子的生物活性代表了它在临床上的表现和由该分子制作的药物的药 效。找出在这三方面具有共同相似的群体进行分析，得出的结论将更加具有现实意义。

对以往研究过的算法分析结果如表7-1 所示，从表中可以看出三个算法的匹配结构越 来越细微、精确，从模糊的全部结构匹配到共有的最大结构再到组成分子的基本单元，有了 很大的突破，对分子结构的组成有了更透明的分析。从它们的共同点看出这些算法都是对 分子的组成进行的分析，不同点是分析越来越细化，但在分子活性的预测上结果并不十分 显著。

**表7-1** **算法的分析对比**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 算法名称 | 作用原理 | 共同点 | 不同点 | 活性的预测 |
| HTS | 以完整图的结构为匹配标准 |  | 全部结构 |  |
| MCS | 以部分图的结构为匹配标准 | 分子结构组成的分析 | 部分结构 |  |
| MQN | 以分子的组成单元为匹配标准 |  | 相似的组成单元 | 20% |

我们算法的设计思路以靶标、作用酶和原子所占比作为影响因素，处理过程分为三个步 骤，如表7-2所示：处理原理，初步处理过程得到单个因素的结果；中间结果，综合处理的第 一步；最终值，算法的最终处理结果。进行药物之间相似度的计算。

**表7-2** **算法的设计思路**

|  |  |  |  |
| --- | --- | --- | --- |
| 影响因素 | 处理原理 | 中间结果 | 最终值 |
| 靶标 |  | T=T,+T, | Tw=(T)\*80%+(Tp)\*20% |
| 作用酶 |  |
| 原子  所占比 |  | Tp |



**7.2** **开发环境及编程语言**

**1.** **开发环境**

该算法所用到的开发环境是 Eclipse,Eclipse 最 开 始 是 由IBM 公司用来代替商业软件 Visual Age for Java 而开发的下一代开发环境，是一个开发源代码的，基于Java 的可扩展开 发平台，其本身是一个框架和一组服务，由包括 Java 开发工具包的插件组件构建的开发 环境。

JDK, 是 Java SE Development Kit的简称，即 Java 的开发工具包，是安装 Java 环境时 需要安装的工具。安装 Eclipse 开发环境之前需要安装 Java 环境。

(1)搭建 Java 的运行环境，即安装 JRE 和 JDK

进 入 Oracle 官 网 ：<http://www.oracle.com/technetwork/java/javase/downloads/>

jdk8-downloads-2133151.html, 下载对应系统版本的安装包，本机是 Windows 10 的 6 4 位

系统，所以下载：jdk-8u91-windows-x64.exe。

首先安装JDK, 双 击jdk-8u91-windows-x64.exe 程序，按照安装指示安装JDK 和 JRE,

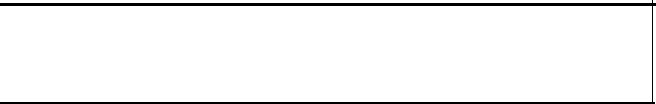
其 中JDK 的安装路径和 JRE 的安装路径不能在同 一 文件夹下，安装好后就要配置环境 变量。

(2)环境变量的配置

右击我的电脑 → 属性 → 高级系统设置 → 环境变量，在用户变量中新建JAVA\_HOME, 添

加变量值“C:\Program Files\Java\jdk1.8.0\_91”。再 新 建 path, 添加变量值“%JAVA

HOME%\bin;%JAVA\_HOME%\jre\bin;” 点击确定，为确保环境变量配置完成，使用 cmd 命 令行检测一下，在 cmd 命令行中输入java-version, 再点击回车键，若出现JDK 的版本信息，输 入Java, 回车显示的有Java 的基本信息，则说明Java 环境配置完成，如图7-1所示。



C:\>java -version

java version“1.8.0\_91”

Tava(TM)SE Runtime Environment(build 1.8.0\_91-b14)

Fava HotSpot(TM)64-Bit Server VM(bui1d 25.91-b14,mi xed mode)

图7-1 Java 环境的配置

(3)Eclipse 的安装

下载 Eclipse 的安装包后双击 Eclipse 的程序图标按照安装指示一步一步安装，中间需 要注意的是安装的路径和工作空间的选择，最好不要选择系统盘安装， 一般是所用的软件安 装在一个固定的盘中，工作空间可以另选一个专门放置运行程序。好的安装习惯是每个程 序员的必备，可以对自己的程序和运行的程序有个更好的了解。选好安装的目录和工作空 间的位置后一般默认其他的选项便可以安装好环境。安装完后，打开开发环境界面新建项 目便可开始算法的编写过程。

**2.** **编程语言**

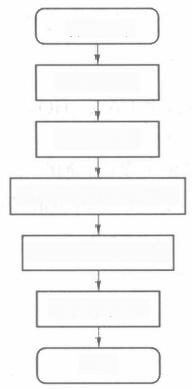
编程语言主要为Java 语言，另外使用R 语 言 ，Java 语言用于编写对数据处理的功能，将 分子的化学性质、物理性质、生物活性对应的数据统一处理为纯数据的过程，以计算数字之 间的关系表示分子的各种活性之间的关系。R 语言主要处理在Java 语言下处理的统 一 化 的数据，将数据之间的关系展示出该数据之间的关系代表的将是个分子之间的各种性质的 关系，以可视化的结果展示出来比在表格之中的数据更加容易观察和理解，数据库使用的是 SQL Server 2012。

(1)Java 编程语言

Java 编程语言是 SUN 公司开发的一种简单的，跨平台的，面向对象的，分布式的，解释 的，健壮的安全的，结构中立的，可移植的，性能很优异的多线程的，动态的语言。Java 是 一 个十分成熟的语言，大多数的高端企业应用都使用该平台。

(2)R 编程语言

R 是用于统计分析、绘图的语言和操作环境。R 是 属 于GNU 系统的一个自由、免费、源代 码开放的软件，它是一个用于统计计算和统计制图的优秀工具，且可以在 UNIX 、Windows 和 Mac OS等平台下运行，主要运行方式是命令行方式，但现在也有开发出来的界面形式。



算法设计

数据的获取

数据的解析

对靶标、作用酶的处理

对原子百分比的分析

相似度的整合

结束

图7-2 算法的基本流程



**7.3** **算法设计**

在研究了多种算法原理和思路之后，发现对分子结构的分析 并不能很好地体现分子的生物活性，而分子对应的靶标、作用酶 等能很好地体现分子的生物特性，于是算法的核心思想就是对靶 标、作用酶、原子百分比的计算和分析，主要是对靶标、作用酶、原 子百分比的权重的计算和处理，它们的权重可以分别代表它们之 间的联系程度，权重越大关系越紧密，反之关系越松散。

**7.3. 1 算法设计流程**

为方便理解该算法的思想和处理过程，用一个流程图来说明 该算法的设计过程，如图7-2 所示。

图7-2中主要说明了该算法设计涵盖两个方面： 一是数据的 获取，二是数据的分析。本算法的核心部分是数据的分析，在分 析过程中又分两个并行的步骤：第一步是对靶标、作用酶的分析，

第二步是对分子式的处理过程。

7.3.2 相似度的计算

要进行药物的聚类必须首先确定相似度的计算方法，首先确定相似度的计算公式，然后 介绍聚类的详细步骤。

1. 相似度的计算公式

靶标和作用酶作为分析分子活性的主要成分，所使用的相似度结算方法为

(7-1)

其中N₄ 是化合物A 的靶标或者作用酶的总个数，Ng 是化合物B 的靶标或者作用酶的总个

数 ，Nc 是化合物A 和 B 共同有的靶标或者作用酶的总个数，T 是化合物A 和 B 之间的权重。 分子百分比之间的相似度量方法：

(7-2)

其中，A, 指的是化合物A 分子中每个原子的相对原子质量，B, 指的是化合物B 中每个原子 的相对原子质量，i 从零到n 是指分子中对应的第一个原子到最后一个原子， T, 指的是分 子百分比的权重。

2. 聚类计算的详细步骤

(1)数据的获取

处理的数据是从 DrugBank 官网下载的 XML 文件，该文件个免费获得，现在需要注册， 但依然是免费的，网址为 <http://www.drugbank.ca/drugs/DB00001>。

(2)数据的解析

因为文件是XML 文件，我们选择用dom4j 方法解析，形成.txt 格式的文件，里面存放 的是之后分析所要用到的相关数据，比如分子的编号、名称、靶标的编号、作用酶的编号、分 子式、分子量等。

(3)数据的分析

①对分子靶标的分析。

在 DrugBank 药物库中每个分子都对应0～n 个不同个数的靶标，通过对不同分子靶标 之间的计算得出每个分子和其他分子靶标之间的关系得到分子之间的关系，该关系所用到 的计算公式为

 (7-3)

其 中 ，Nc 是两个分子之间靶标的共同个数，N₄ 是分子A 所对应的靶标个数，Ng 是分子B 所对应的靶标个数，Nc 是分子A 和 B 所共有的化靶标个数，计算得到的 T 为 分 子A 和 B 之间的权重，权重代表了该两分子之间的联系强度，T 的值越大两分子之间的联系越强，T 的值越小两分子之间的联系越弱。

② 对分子作用酶的分析。

作用酶的分析方法和分子靶标的分析方法一样，在这里就不重复了，其计算公式为

 (7-4)

最后结果将得到两个权重，其中一个是靶标的权重，另一个是作用酶的权重，将其两权 重相加得到的 T=T,+T. 是该分子的一个相似度划分标准。

③对分子中各原子百分数的分析。

每个分子式由多个不同的原子组合而成，每个原子在分子中都有一定的所占比，计算出 每个分子的所占比，计算公式为 ,其中 subE 是分子中的单个元素的相对原子质 量，E 是该分子的相对分子质量，计算出每个分子中的每个原子的百分比后还需计算每个分 子之间的关系，该公式为

 (7-5)

其中，n 表示比较的两分子共有的原子数目，A, 表示分子 A 的单个元素的百分比，B; 表示 分子B 的单个元素的百分比。这样计算出来的值为每个分子和其他分子之间的权重。

将靶标、作用酶、元素百分比计算完之后将得到三个不同的权重，为了计算的确定性和方便 性，需要将三个权重进行量化，即在同一个标准下，其取值都在0～1。最终的权重计算公式为

Tw=(T)\*80%+(Tp)\*20% (7-6) 其中，80%和20%是根据科学的研究所设定的。在MQN 算法中，经过 SF 和 MQN 相似性 度量的分析和比较，最终得出与生物活性的相似度在20%左右的结论。

最后计算每个分子之间的关系并以最直观的视觉接受方式——图形方式展示出来。 (4)数据库的设计

所用的数据库包含5个表，其中表 Dbo.Relation 中存放的是靶标、作用酶的权重；表 Dbo.DrugBank 中存放的是该课题所处理的所有信息，包括药物的编号、名字、分子量、分子 式；表 Dbo.DrugFormula 中存放的是每个分子中各原子所占的分子百分比数；表 Dbo.EM 中则存放的是语元素中期表中每个元素的相对原子质量；表 Dbo.PResult 中存放的是分子 百分比的平方差和最终计算结果(边的大小)。为了使程序能够清楚明确的执行，在 SQL Server 中建立了数据库 DrugBank2, 在该数据库中建立了5个表。

表 DrugBank 用于存储药物的基本信息，其结构如表7-3所示，其中主键id 中存放的是 药物条数的编号，记录有多少条数据。DRUGID 中存放的是药物在药物库中存放的编号， 每一个编号对应一个药物以及该药物分子的信息。Name 中存放的是药物分子的名字。 Wt 和 Formula 中存放的分别是分子的相对原子质量和分子式，这两个量将在分子式中原 子的占用比计算中被使用。

表7-3 DrugBank 表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 宽度 | 是否可空 | 说明 |
| id(primary key) | int | 1000 | not null | 编号 |
| DRUGID | int | Not null | 分子编号 |
| Name | Varchar | Null | 分子名称 |
| Wt | Float | 100 | Null | 相对分子质量 |
| Formula | Varchar | Null | 分子式 |

第7章 大数据在 DrugBank 药物数据库聚类方面的应用

表 Relation 用于存储各个药物之间靶标、作用酶之间的权重和关系，其结构如表7-4所 示 。Tid 和 Eid 在后期的结果中是以节点的形式存在，表示被分析的药物的编号，Weight 代

表的是被分析的两分子之间靶标和作用酶的权重和。

**表7-4** **Relation 表结构**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 宽度 | 是否可空 | | 说明 |
| id(primary key) | int |  | not | null | 关系编号 |
| Tid | float |  | Not | null | 药物分子id |
| Eid | float |  | Not | null | 药物分子id |
| Weight | Float |  | Not | null | 药物间关系的权重 |

表 DrugFormula 用于存储各原子在分子中的质量百分比，如表7- 5 所示。

**表7-5** **DrugFormula 表结构**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 宽度 | 是否可空 | 说明 |
| id(primary key) | int |  | not nul | 关系编号 |
| H | Float |  | null | H的质量百分比 |
| He | Float |  | null | He的质量百分比 |
| …… | Float |  | null | ……的质量百分比 |
| Xe | Float |  | null | Xe的质量百分比 |
| Tl | Float |  | null | Tl的质量百分比 |

表 EM 用于存储元素周期表中各元素的相对原子质量，其结构如表7- 6所示。

**表7-6** **EM 表结构**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 宽度 | 是否可空 | | 说明 |
| Element | Varchar | 10 | not | null | 元素名称 |
| Countl | float | Not | null | 元素相对原子质量 |

表 PResult 中存放的是药物分析的最终结果数据，如表7- 7所示。

**表7-7** **PResult** **表结构**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 宽度 | 是否可空 | 说明 |
| Id(primary key) | int |  | not nul | 编号 |
| Precent | Varchar | 10 | not null | 元素的平方差 |
| Result | float |  | Not null | 三因素的综合结果 |



**7.4** **算法实现**

**7.4.1** **文件的解析**

本算法要分析的文件类型是 XML 文件，该类型文件的结构一般是N 叉树的结构，由 于该文件较大，里面含有数万个节点，每个节点下面还有数万个节点，要把握该文件的整 体结构是个极大的工程，解析该文件的方法有两种， 一种是 DOM 解析， 一种是 SAX 解析。

DOM 解析的方式是将要解析的文档整个放在内存中，对文档的结构有一个全局的掌 控，每个节点和节点的内容都有条理的被存储以便被提取，其优点是，在 dom 方式下对文档 的增删改查比较方便，其缺点是对内存的消耗比较大，如果文档过大，超过了内存的极限就 不能被解析。

SAX 解析的特点是将文档的内容从上向下一行一行读，读一行解析一行，这样的解析 方式就决定了它的优点是内存的消耗比较小，但是却不适合增删改查，只适合读取，若想对 文档进行增删改查请选用 DOM 方式。目前对文档的解析大部分是使用 Dom4j 解析包进 行解析。Dom4j 是一个简单的、灵活的开放源代码的库，是一个非常优秀的 Java XML API, 具有性能优异、功能强大和极易使用的特点，使用Dom4j 开发需下载Dom4j 相应的jar 文件，解析 XML 文件时将 Dom4j 的 jar 导入工程中即可调用相应的解析方法，十分方便。

在对文件解析之后接下来就要处理的是对各因素的分析，该部分是本算法的核心，在本 章中分析的影响分子活性的各因素分别是分子的靶标、作用酶、分子中各原子的百分比。

**7.4.2** **对靶标、作用酶的分析**

靶标是指自身具有药效作用并能被药物作用的生物大分子，能通过适当调节靶标的生 理活性有效的改善疾病症状，所以对靶标的分析是具有实际意义的，作用酶对分子的生物活 性具有催化作用，在适合的条件下可以加快分子与其他分子之间的作用速度，提高作用效 率。分子百分比是指每个分子中原子所占的百分比例。分子是由一个一个原子组成的，每 个原子都有自己的相对原子质量，每个原子的质量占它们组成的分子的总质量的比例就是 该原子的分子百分比。选作这些因素作为衡量的指标是因为靶标和作用酶体现出的是分子 的生活特性，分子百分比则体现的是分子的物理特性，物理特性虽然对分子活性的影响不 大，但仍然有一定的影响，加上该特性聚类结果会更加准确。

对靶标和作用酶的处理过程：每个分子都有一个或者多个作用的靶标，和一个或者多个 起相辅助作用的作用酶。分子在作用酶的帮助下作用于靶标起到相应的作用，对机体产生 一定的影响。该段对靶标和作用酶的分析做一下详细的讲解。要分析的分子的靶标和作用 酶是以它们对应的编号存储在外部文件中以便调用分析，分析所使用的公式在前面已经做 了相关的说明，这章将在表7-8中用一个实例对其进行进一步说明：

表7-8 靶标、作用酶分析实例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 药物编号 | 靶标编号 | 公共靶标编号 | 计算权重 | 结果 |
| A | 1192、1143 | 1192 | T,=1/(2+3-1) | 0.25 |
| B | 4797、4852、1192 |

化合物 A 和化合物 B 用 A 和 B 表示，A 的靶标个数为2,其中对应的编号分别是 1192、1143,B 的靶标个数为3,其对应的编号分别是4797、4852、1192,权重的计算方法是

根据5-1介绍的公式所得。



(7-7)

计算得到 T, 的值为0.25,即化合物A 和 B 之间的联系强度为0.25,为保证数据的准 确性，数据类型选择的是 Double 型，在整个数据集的分析过程中为了避免数据分析的冗余， 采用的分析方法是使用两个 for 循环嵌套对分子进行遍历分析，既不遗漏也没有冗余。靶 标和作用酶之间的关系列表，如表7-9 所示。

表7-9中Weight 列内的数据表示的是靶标和作用酶的权重之和。前两列是被分析的 药物编号，在后期分析中也是节点之间的边。

表7-9 靶标和作用酶的权重

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Tid | Eid | Weight |
| 0 | 100001 | 100006 | 1 |
| 1 | 100001 | 100055 | 0.083333333333333329 |
| 2 | 100001 | 100100 | 0.14285714285714285 |
| 3 | 100001 | 100170 | 0.076923076923076927 |
| ……… | \*\*\*\*\*\* | \*\* |  |
| 100 | 100002 | 100005 | 0.73333333333333328 |
| 101 | 100002 | 100028 | 0.35294117647058826 |
| 102 | 100002 | 100051 | 0.84615384615384615 |
| 103 | 100002 | 100054 | 0.73333333333333328 |
| 104 | 100002 | 100056 | 0.84615384615384615 |
| … | \*\*\*#\*\* | \*\*\*\*\*\* | \* · |
| 142 | 100003 | 100127 | 0.25 |
| 143 | 100003 | 100131 | 0.0625 |
| 144 | 100003 | 100173 | 0.083333333333333329 |
| 145 | 100003 | 100194 | 0.25 |

**续表**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | \*\*\*\*\* |  |
| 87027 | 100996 | 101412 | 0.142857142857143 |
| 87028 | 100996 | 101708 | 0.0588235294117647 |

表7-10所示是表7-9的表间关系。

**表7-10** **表7-9的表间关系**

|  |  |  |
| --- | --- | --- |
| 字段名 | 中文含义 | 意义 |
| Id | 编号 | 数据的总个数 |
| Target | 靶标编号 | 靶标的编号 |
| Enzyme | 作用酶编号 | 作用酶的编号 |
| Weight | 权重 | 靶标作用酶的权重和，代表两分子之间的关系 |

该模块单流程图，如图7-3所示。

**7.4.3** **对分子中原子百分比的处理过程**

分子中的原子组成一部分的反映了分子的结构信息，通过对多个算法研究之后发现对 分子的分析基本都是对分子的结构的分析，虽然分析形式不同，但分析的方向确是一致的， 只是分析的深浅不同，2012年发表的一个根据分子量数分析 DrugBank 中的分子的算法将 组成分子的基本单元拆分到了不可拆分的地步，相比之前的那些对分子结构、最大子结构和 指纹集等分析的算法来说有了一个很大的提升，之后还对分子进行了分子活性的预测，但是 预测的结果仅有20%是重叠的，根据这个结论得出分子的结构对分子活性的影响只占一部 分比例，而分子百分比是借鉴这个方法得出的，同样分析了分子的结构组成，下面就用一个 实例解释一下分子百分比的分析方法。

在上节中介绍了分子百分比的计算方法，在这里用一个实例再次进行具体的介绍，以 H₂O 和CO₂ 为例计算它们之间的相似度权重，首先在数据库中查到各原子的相对原子质 量，H 的质量是1.0079,0的质量是15.999,C 的质量是12.011,H₂O 的相对分子质量是 18.0148,CO₂ 的相对原子质量是44.009,得到各原子的相对原子质量后开始计算单个分子 中各原子所占的百分比，根据上节描述的计算方法得到 H 在 H₂O 中 的 质 量 比 为 0.11189688,0在H₂O 中的质量比为0.88810312,C 在CO₂ 中的质量比为0.27292145,0

在CO₂ 中的质量比为0.72707856.其分子之间的元素关系为表7-11所示。

因此可以得到H₂O 和CO₂ 之间的权重为

T,=1/3 (0.111896-0)^2+(0.888103-0.727078)²+(0-0.0272921)^2

(7-8)

第7章 大数据在 DrugBank 药物数据库聚类方面的应用

开始

参数初始化

判断参数值是 否小于分子数，

否

是

判断参数值是 否小于分子数

否

否

是

参数是否小于 前一个酶数量

否

参数是否小于

前一个靶标数

是

否 参数是否小于前

一个酶数量

是

否

酶数量前者和

否

后者是否一样

是

参数2加1

是

参数是否小于

后一个靶标数

是

靶标数前者和后 者是否一样

是

参数1加1

参数1加参数2得

到参数3

否

否

参数3大于0

是

将值加入数据库

结束

图7-3 计算靶标、作用酶权重的流程图

**表7-11** **分子元素之间的质量比**

|  |  |  |  |
| --- | --- | --- | --- |
| Id | C | H | O |
| H₂O | 0 | 0.111896 | 0.888103 |
| CO₂ | 0.272921 | 0 | 0.727078 |

大数据技术及行业应用

|  |  |  |  |
| --- | --- | --- | --- |
| 第531个分子相对分子质量：824.9664 | | | |
| 单子  原子  H  cl  N  原子 | 厚子质量 1.8079  14.007  15.999  原子质量 12.811  1.0079  35.453  14.007  15.999  原子质量 | 车本车章  原子个故 单车安案  28 该元素所占总质里的比重：0.06648268 2 谈元素所占总质里的比重：0.06599457 5 该元素所占总质里的比重：0.18844987  第532个分子相对分子质量：424.49417  重主 | |
| 原子个数  17  16  1  1 | 电事单率  该元素所占总质里的比重：0.6507309  该元素所占总质量的比重：0.051393807 该元素所占总质量的比重：0.11298644 该元素所占总质理的比重：0.13391824 该元素所占总质霍的比重：0.050987788 |
| 第533个分子相对分子质量：313.7864  事  原子个数 | |

图7-4 分子百分比的部分结果

计 算 的 结 果 为 T,=0. 037635706, 即 H₂O 和 CO₂ 之间的权重为0 .037635706,通 过这个结果可以看到这两个分子之间的权重 值很小，可以得到的信息是这两个分子之间在 结构上的相似度很低，事实上也确实如此。同 样为了避免数据处理时候冗余，处理方法和处 理靶标和作用酶的方式是 一样的，即用两个 for 循环嵌套，得到的处理结果如图7-4所示， 为了使得每个原子的百分比都准确地被计算 进去，表中添加的是所有出现的原子，其他分

子中没有出现的值直接跳过，内容为 null, 表示没有值，但后来在调试过程中发现如果单独 的将分子百分比和靶标作用酶的分析分开计算，分子A 和 分 子B 之间的靶标作用酶和分子 百分比在数据表中是不对应的，即同样是一行数据，但靶标和作用酶的数据是分子 A 和 B 的，分子百分比的数据则可能是分子 A 和 C 的，经过仔细研究算法逻辑和分析方法，最终选 择以靶标和作用酶的分析结果为限制，在分析完靶标和作用酶的情况下计算百分比，这样得 到的结果是具有实际意义的。

单独模块的运行过程中的结果：该部分是单独把分子式拿出来进行分析，得到准确的结

果后被封装进主函数中被调用。

图7-4中的结果主要起到两方面的作用： 一是为了测试程序正常的运行，确保得到的数

据是正确无误的，二是可以在运行过程中随时查看运行结果，清楚当前运行的进度。

每个分子中各原子所占的百分比，如表7-12所示。

**表7-12** **原子所占分子的质量比**

|  |  |  |  |
| --- | --- | --- | --- |
| H | C | N |  |
| 0.063686482608318329 | 0.49503758549690247 | 0.16092082858085632 |  |
| 0.069428049027919769 | 0.53421914577484131 | 0.16641417145729065 |  |
| 0.0688726007938385 | 0.5423731803894043 | 0.16231589019298554 |  |
| 0.070669949054718018 | 0.53338420391082764 | 0.16473877429962158 |  |
| 0.068360678851604462 | 0.52137243747711182 | 0.16977386176586151 |  |
| 0.063794486224651337 | 0.53987336158752441 | 0.15418532490730286 | = |
| 0.070004723966121674 | 0.58595168590545654 | 0.18530867993831635 |  |
| 0.0670541524887085 | 0.52261275053024292 | 0.17697855830192566 |  |
| 0.0738392248749733 | 0.53296655416488647 | 0.18354068696498871 |  |
| 0.070873744785785675 | 0.53684353828430176 | 0.16524986922740936 |  |

<< 142

**续表**

|  |  |  |  |
| --- | --- | --- | --- |
| H | C | N |  |
| 0.072156831622123718 | 0.53212177753448486 | 0.1774088591337204 |  |
| 0.06945674866437912 | 0.53096669912338257 | 0.1723509281873703 |  |
| 0.0666952058672905 | 0.55825042724609375 | 0.19861660897731781 |  |
| 0.0680001974105835 | 0.526681125164032 | 0.17654871940612793 |  |
| 0.072156831622123718 | 0.53212177753448486 | 0.1774088591337204 |  |
| 0.070485532283782959 | 0.50747948884963989 | 0.17958460748195648 |  |
| 0.070132680237293243 | 0.52121514081954956 | 0.16862128674983978 |  |
| 0.067258827388286591 | 0.51650756597518921 | 0.16494885087013245 |  |
| 0.069727621972560883 | 0.53511178493499756 | 0.17037473618984222 |  |
| 0.069266259670257568 | 0.52825123071670532 | 0.16882163286209106 |  |
| 0.07496488094329834 | 0.63172310590744019 | 0.14882898330688477 |  |
| 0.069410324096679688 | 0.53302794694900513 | 0.16610220074653626 |  |
| 0.0670541524887085 | 0.52261275053024292 | 0.17697855830192566 |  |
| 0.06646447628736496 | 0.53147846460342407 | 0.15675878524780273 |  |

表7- 12是数据库 DrugBank2 中的表 DrugFormula 中的部分数据，这个表中有52列， 代表的是组成 DrugBank 库中所有分子的原子，行数共有8188行，表示的是分子的个数。 为了准确地得到这52列原子的值，在调试过程中分了三个步骤实现该目标：第一步是以主 观想法将组成分子的常用原子加入数据库的表中；第二步是在程序的调试过程中将表中没 有的元素加入；最后一步是将表中多余的元素删除，这样就完成了参与分子的所有元素的 添加。

各原子的相对原子质量是常量，可以通过元素周期表查找，然后将它们存放在另一个表 中以供计算该所占比时调用。原子所占比之间的差异十分微小，为了精确地表示各原子之 间的所占比，DrugFormula 表中的各元素的类型和计算结果使用Double 型的数据记录，没 有所占比的位置使用null 值表示为空。

各分子中原子百分比的平方差(Precent 列)的结果，如表7-13所示。

**表7-13** **原子百分比的平方差的结果**

|  |  |  |
| --- | --- | --- |
| Id | Precent | Result |
| 0 | 0.00013920368873634212 | 0.80002784073876687 |
| 1 | 3.163758403444139 E-07 | 0.0666667319286546 |
| 2 | 0.0002234974043045879 | 0.1143304188764887 |
| 3 | 0.00053934167276690592 | 0.061646332161035391 |

**续** **表**

|  |  |  |
| --- | --- | --- |
| Id | Precent | Result |
| 4 | 0.00078374681115574044 | 0.80015674936585135 |
| 5 | 0.00087008327833777536 | 0.20017401665681975 |
| 6 | 0.0014560999038187232 | 0.26695789459627123 |
| 7 | 0.0031306766997029308 | 0.40062613533809782 |
| 8 | 0.00030320018617874489 | 0.26672731465077959 |
| 9 | 0.00035218985529171835 | 0.80007043796940713 |
| 10 | 9.6746686765652619E-05 | 0.80001934933679886 |
| 11 | 0.0017510518069197649 | 0.26701688496395948 |
| 12 | 0.00086471417582478251 | 0.26683961744420231 |



开 始

初始化参数

获取数据

数据是

否为空

否

参数小于数据 集总行数?

是

参数小于数据 集总数减1

是

参数小于分子 元素个数?

1 是 计算前一行和 后一行对应原 子的平方差

更新数据库

结束

是

否

否

否

图7 - 5 原 子 平 方 差 的 简 单 流 程 图

表7-13中的两个数据列的行数为173283,是 8188个分子之间无重叠的所有分析结果。Precent 是根据 DrugFormula 表中的数据计算的分子之间 元素平方差的值，Result 中的数据是该算法设计中 数据处理的最后一步，把对分子活性有影响的三因 素进行了综合处理。靶标和作用酶是主要因素，影 响因子是0.8,分子结构所对应的性质影响因子为 0.2,将表 Relation 中 Weight 的 8 0 % 和 PResult 中

Precent 的20%相加得到 Result 中的结果。

决定影响因子的值是根据 MQN 算法赋值的， MQN 算法是基于分子量子数的一种算法，是对组 成分子的具有独立性质的单元的一种相似度度量， 根据这个度量预测生物活性，得到的相似度值维持 在20%左右，即表明分子结构对分子的活性中所起 到的作用在20%左右，用这个数值作为分子活性的 影响因子是比较合理的。

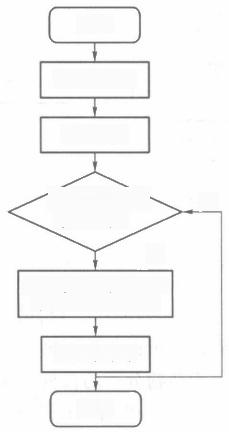
原子平方差的处理函数流程图，如图7-5所示。 该流程图简单描述了计算原子平方差的逻辑过程， 从这个流程图中可以清楚地了解到计算的基本过 程和计算思路。

相对原子质量：每个原子的相对原子质量被存 储在表 EM 中，程序运行时每处理一个分字中的原 子，都要在这个表中调用对应的相对原子质量，如 表7-14所示。

**表7-14** **元素的相对原子质量**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | 1 | 2 | 3 |  | n |
| Element | H | C | O | \*\*\*\* | Ca |
| Count1 | 1.0079 | 12.011 | 15.999 | \*\*\*\*\*\* | 40.08 |

该模块的简单流程图，如图7-6所示。



开始

初始化参数

获取数据

参数是否小于 数据行数?

是

解析分子元素，计算 单个原子的百分比

插入数据库

结束

否

图7-6 分子百分比的处理过程

**7.4.4** **结果的整合**

在分析完靶标、作用酶、分子百分子之后，最重要的是对分析数据的整合和对最后结果

的分析，表7- 15描述了数据整合分析的结果。

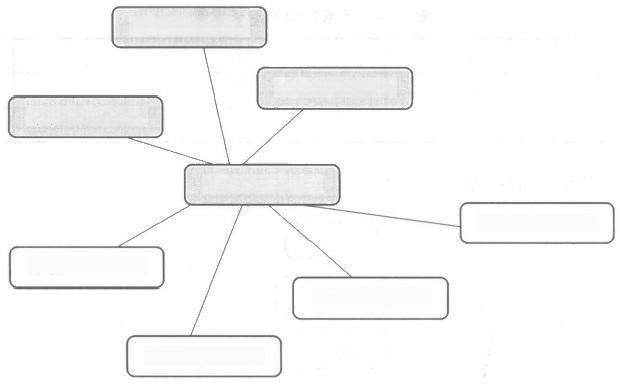
**表7-15** **数据的整合分析**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tid | Eid | Weight | Precent | Result(Weight\*0.8+precen\*0.2) | |
| 100001 | 100006 | 0.5 | 0.437287 | 0.4874574 | 100001 |
| 100002 | 100005 | 0.33 | 0.436781 | 0.3513562 | 100002 |
| 并 号 性 善 号 | ## | 普 |  |  | 黄 黄 号 苷 苷 |
| 109330 | 109570 | 1 | 0.374223 | 0.8748446 | 109330 |

Tid 和 Eid 的值做分析的两分子的编号，Result 是指处理之后的最终数据结果，其结果

数据在表7 - 15中的 Result 中 显 示 。

结果数据的部分运行展示，如图7- 7所示。

分子id:1017030

分子id:1017630

分子id:1001700

分子id:1000016

分子id:1040070

分子id:1005550

分子id:1047060

分子id:1036470

图7-7 部分运行结果

**7.4.5** **最终结果展示**

在 R 语言的运行环境下对 DrugBank 中分子的分析结果聚类如图7-8所示，其中的三个坐 标分别是 rt$Tid 、rt$Eid 和 rt$Weight, 前两个是药物分子的编号，另一个坐标是影响药物分 子生物活性的三因素的综合值，也称为药物分子的权重。该三维散点图在空间中展示了分子 聚类的结果。在权重的坐标平面上的 一条线中分布比较密集的代表这些分子之间的关系比较 紧密，相似度较高，分布疏松的代表这些分子之间的关系联系较弱，相似性较低。图7-8是使 用 R 语言中的 rgl 函数和 一 些简单的处理语句生成的三维散点图，在这个散点图中可以全方 位的观察数据之间的关系。

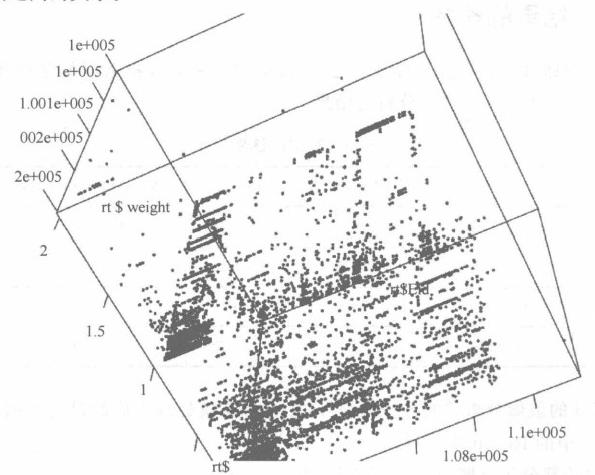


图7-8 R 语言结果展示

第7章 大数据在 DrugBank 药物数据库聚类方面的应用

图7-9 的结果中线的长度表示划分的 不同的类，同一种颜色的节点具有相似的 性质，每一类中的分子是具有一定程度相 似靶标、作用酶和分子百分比的，这些具 有相似性的分子在 DrugBank 中可能具有 不同或者相同的分子结构，这些具有相同 活性却具有不同分子结构的则十分具有 研究价值，根据这个结果可以研究探索新 的活性的分子，从而制造新的药物，此分 析结果对科学研究具有启发性作用。

本节详细介绍了算法的核心思想，算 法的设计模块主要分数据的获取、数据的 解析、数据的分析、数据库的连接和设计 和最后的结果展示等步骤。其中解析数 据使用的是 dom4j 方法，数据分析分为对

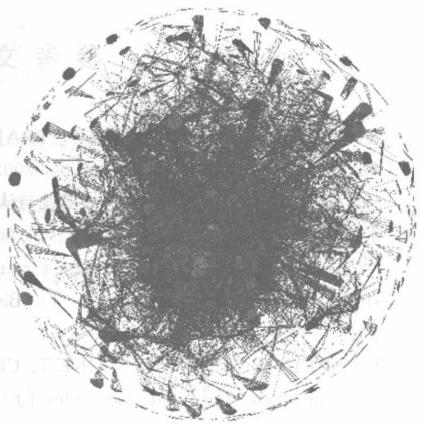


图7-9 结果展示

靶标、作用酶的分析和对分子组成的分析，数据库的连接和设计则是根据数据分析的需求建 立合适的数据表，结果以二维图和三维图两种图形展示。



**本** **章** **小** **结**

分子的聚类在生物化学领域是一直被关注的话题，特别是分子的聚类对药物的发现和 研究有着重要的作用，基于大数据思维，从全局角度对西药进行聚类分析对新药的研制具有 指导意义。

本章实现的算法是对影响分子生物活性的三因素——靶标、作用酶、分子结构的综合分 析。靶标和作用酶是主要因素，分子结构作为辅助因素，成分占用比为8:2,主要因素占 80%,辅助因素占20%。对各因素的处理方法根据因素的不同有各自的特点，主要因素是 根据靶标和作用酶的相似比例，辅助因素则根据组成原子平方差值。

该算法的创新点在于对现有的聚类算法在新的方向上有所突破和对西药的研发具有 重要的参考价值。在化学分子领域中现有算法聚类分析的方向在于分析分子的组成，进 而预测分子的生物活性，尽可能地在分子的结构组成中找到对生物活性有影响的基本单 元。而本节提出的算法是直接对影响生物活性的因素进行聚类，对同类中具有相同和不 同结构的分子分别进行分析，找到影响分子活性的组成。相对以往的算法来讲是反向思 维的过程并且十分有效，根据分析的结果表明该算法对科学的研究和发展具有启发性 意义。

**参** **考** **文** **献**

[1] STURN A,QUACKENBUSH J,TRAJANOSKI Z.Genesis:cluster analysis of microarray data [J].Bioinformatics,2002,18(1):207-208.

[2] 方开泰.聚类分析(I)[J]. 数学的实践与认识，1978,(1):25-29.

[3] 孙吉贵，刘杰，赵连宇.聚类算法研究[J]. 软件学报，2008,19(1):48-61.

[4]AWALE M,REYMOND JL.Cluster analysis of the DrugBank chemical space u- sing molecular quantum numbers [J].Bioorganic &.medicinal chemistry,2012,20 (18):5372-5378.

[5]BACKMAN TW,CAO Y,GIRKE T.ChemMine tools:an online service for analy-

zing and clustering small molecules [J].Nucleic acids research,2011,39(Web

Server issue):W486-491.

[6] CHEN B,DONG X,JIAO D,et al.Chem2Bio2RDF:a semantic framework for linking and data mining chemogenomic and systems chemical biology data [J].Bmc Bioinformatics,2010,11(20):3011-3015.

[7] LOU H,DEAN M.Targeted therapy for cancer stem cells:the patched pathway and ABC transporters [J].Oncogene,2007,26(9):1357-1360.

[8] WISHART D S,CRAIG K,AN CHI G,et al.DrugBank:a knowledgebase for drugs,drug actions and drug targets [J].Nucleic acids research,2008,36(Data- base issue):D901-906.

[9] CRAIG K,VIVIAN L,TIMOTHY J,et al.DrugBank 3.0:a comprehensive re- source for‘omics’research on drugs [J].Nucleic acids research,2011,39(Data- base issue)::D1035-D1041.

[10] ALTIERI D C.Survivin,cancer networks and pathway-directed drug discovery[J].

Nature Reviews Cancer,2008,8(8):61-70.

[11] 江寿平.酶和底物分子之间相互作用力对扩散控制反应速率的影响[J].Acta Bio-

chimica Et Biophysica Sinica,1977,1):329-338.

[12] 高新亮.脂肪酶固定化及其催化性能研究[D]. 大连理工大学，2006.

[13] 姚新生，叶文才，栗原博.阐明中药科学内涵推进中药现代化与创新药物研究进程

[J]. 化学进展，2009,(1):2-13.

[14] 石海信，黄冬梅，谭铭基，et al. 拓扑学原理在化学化工中的应用[J]. 化学工程师， 2010,24(7):38-41.

[15] 邓亚明，杨邦荣.基于 ECLIPSE 图形插件开发的研究[J]. 电脑开发与应用，2009,22

(2):19-21.

[16] 白晓旸 .J2EE 在企业信息系统建设中的应用[J]. 机械设计与制造，2004,(4):18-20.

[17] 高德霖.分子连接性方法及其在结构—活性相关中的应用[J]. 江苏化工，1998,(4):

37-40.

[18] CAO Y,JIANG T T.A maximum common substructure-based algorithm for

searching and predicting drug-like compounds [J].Bioinformatics,2008,24(13):

i366-i74.

[19] 林大海，万常选.基于区间编码方案分裂大型 XML 文档到关系存储[J]. 计算机应

用，2004,24(2):141-145.

[20] 张孟旺.基于VTD-XML 技术的异构数据库数据交换系统的研究与设计[D]. 电子

科技大学，2010.

[21] 陈凯先，蒋华良，罗小民，et al.基因组时代的新药发现：趋势和实践.proceedings of

the 全国抗生素学术会议，F,2 005[C].

**第** **8** **章** **大数据在电子商务数据** **分析中的应用**

随着互联网的飞速发展，各种信息呈爆炸式的增长。从中获取有用的信息对国家安全、 政策出台、经济发展、科学研究、企业决策、个人生活方式等各方各面都将产生巨大的影响， 除此之外，对于大数据的挖掘成为时下最火的商机，其潜在的商业价值无可限量。然而互联 网中的信息杂乱无章，如何从互联网中获取有用信息，如何将获取的信息进行分类成为首要 解决的问题。因此针对以上情况，我们研发了本系统，该系统使用网络爬虫对特定网页进行 爬取，使用流行的第三方包对网页进行解析，获得有价值的信息并格式化保存，之后对分类 器进行训练、测试，对获取的数据进行分类，分析得出结果，并将结果图表化，给用户以直观 展示，此外该系统还包含了用户管理功能，实现了对于用户的权限，增、删、改、查等功能。本 章将详细介绍该系统详细的开发过程。



**8.1** **研究现状**

大数据具有数据体量巨大，数据类型繁多，价值密度低，产生速度快，政治、经济价值巨 大的特点，因此受到人们的高度关注。它是继物质，石油能源后的又一个引起世界重视的新 能源。是在2008年谷歌提出云计算概念、2009年欧盟提出物联网计划之后的又一次网络 科技的进步。对大数据的研究成为 IT 界的又一次重大技术变革。大数据的价值体现在利 用数据分析的方法从数据中获取有用信息，为世界经济，政治，生活等各个方面服务。比如 离我们生活最近的就要数网购了，我们的淘宝账号里充斥着大量的推荐信息，应用大数据技 术可以根据我们以往购买的经历和相似人的购买经历，分析出我们很可能要买的东西并推 送给我们，而我们十有八九会去买，这就是大数据最真实的体现。在国外，美国奥巴马政府 在2012年3月发布了“大数据发展计划”,并将大数据定义为“未来的新石油”。在我国， 2012年7月国务院颁布了《“十二五”国家战略性新兴产业发展规划》将物联网、云计算设为 重点发展方向和主要任务。2013年5月，国家科技部在香山第二次会议中讨论了“数据科 学与大数据的科学原理及发展前景”并且设立了关于大数据的专项研究计划，投入大量的人 力、财力。通过这些大事件表明了世界各国政府对“大数据”的重视程度，以及未来发展的 方向。

由于大数据的潜在价值是无可估量的，数据挖掘近年来成为商业界普遍关注的对象， 它对商家提升客户服务，研发新产品，制定发展方向等等有着重要的作用。数据挖掘是 由海量的数据、机器学习两大方面来支持的。机器学习的研究主要是指使用计算机来模

第8章 大数据在电子商务数据分析中的应用

拟人类的学习能力，通过使计算机根据已有的数据进行算法的训练学习，对新数据进行 预测分析，通过此方法来不断的修改算法，完善算法，来提高对新数据预测分析的正 确率。

进入信息时代后，随着大数据时代来临，互联网中出现大量信息并涌入了我们的生活， 我们时时刻刻接收着来自互联网的各种信息，但多数并不是我们需要的。如何对信息分类 成为一大难题，传统的手工分类耗时、耗力、耗财、效率低，但是凭借人类对语言的浅层及深 层意义的理解无可替代，能将文本精确的分类，于是我们模仿人类，开发出能将文本智能分 类的系统，在效率和成本方面弥补人工的不足，文本智能分类的精确程度是信息科学界仍不 断解决的问题与进步的目标，使之尽可能地代替人类工作。对于互联网丰富的信息，文本智 能分类使我们能快速、准确地获取到有用的信息，排除无用的垃圾数据，整理杂乱无章的信 息，成为全面、广泛、快捷地吸收信息的有效方法。下面介绍一下文本分类国内外研究现状：

**1.** **国外的研究现状**

国外的文本分类的研究起源比较早。卢恩早在1958年，就首先提出了将词频思想应用 在文本分类中。之后很多学者在文本分类领域进行研究，比如 Maron 在1961 年正式发表 了一篇关于文本自动分类的文章，取得巨大研究成果。到了1990年左右，机器学习逐渐成 为主流的文本分类技术，通过提取文本特征，自动训练出分类模型，从而大大减少了人力资 源，能够快速准确地进行文本分类。

**2.** **国内的研究现状**

由于国内发展落后，文本分类研究比较晚，直到20世纪80年代初才对文本分类进行了 研究，不过近年来，机器学习在国内迅猛发展，结合中文的分词在中文文本分类领域有了飞 跃的发展。目前较为成熟和流行的分类算法有朴素贝叶斯、支持向量机(SVN) 、K 临 近 (KNN) 、 决策树等，其中朴素贝叶斯算法简单、效率高，被广泛使用，如垃圾邮件处理、图书 管理、新闻分类、情感分析等。



**8.2** **相关技术及概念**

**8.2.1** **网络爬虫**

网络爬虫就是一个可以从互联网中不断下载各种网页的程序，通用网络爬虫的工作方 式是从网页集合中获得链接，之后发送 http 请求，开始下载网页，分析获得最初网页上的链 接，继续下载这些链接的网页，再从当前页面中获得新的链接，如此重复执行，获取每个网页 上的每个链接，不断深入，直到满足爬虫爬取的停止条件为止。但是出于不同的原因，不同 的用户往往具有不同的信息需求，通用爬虫所爬取的页面结果往往包含着大量用户不需要 的信息，比如广告。因此，本系统中的爬虫一方面继续延用通用爬虫的方式抓取网页，另一 方面又并不追求大面积覆盖所有网页，而将目标定为抓取与用户特定主题内容相关的网页， 面向用户需求，为有特定目标的用户提供数据资源。

**8.2.2** **HtmlUnit 工具包**

HtmlUnit 是 一 个给Java 开发用的浏览器，这种浏览器没有界面，由一个 WebClient 对 象来模拟浏览器，通过该对象的API 来操作页面上的各种元素，比如链接、按钮、表单等，还 可以执行页面上的动作，比如单击、提交等，非常实用方便。本系统采用 HtmlUnit 作为网 页解析工具，它具有一个异于其他解析工具的特点，就是 XPath 。 例如某个页面有如下部分 html 标签：

<p class="productTitle">

<a href ="//detail.tmall.com/item.htm?amp;is\_b=1"target ="\_blank">

<span class=H>子女夏天折叠防晒遮脸</span>

<span class =H>遮阳</span>

<span class=H>夏季防紫外线沙滩太阳帽</span>

<span class =H>面纱可拆卸</span>

</a>

*</p>*

HtmlUnit 提供解析网页的XPath 方法，比如要获取第一个三个 span 内的文字“夏季防紫 外线沙滩太阳帽”,XPath 就可以写为//p[@class=’productTitle’]/span[3], 其中“//”代表 html 中的任意位置，“/”代表子节点，“@”代表标签属性。如果要获取标签的属性值，比如要获 取 a 标签的 href 属性，那么就可以写为//p[@class='productTitle’]/a/@href。

它是唯 一 一 个可以执行网页中JavaScript 的解析器，能够返回到页面一些动态数据，但 需要付出惨重的时间代价。如果不使用 HtmlUnit 的 JavaScript 功能，获取一条数据的时 间大概是0 . 5秒，反之获取 一 条数据的时间大概是1～2秒，甚至更多。除此之外，Htm- IUnit 对 于JavaScript 的执行效率不太高，会报出大量的错误。因此本系统尽可能平衡优缺 点，使之达到最好的效果。

本系统中使用的爬虫规则即为 HtmlUnit 中 的 XPath, 将页面链接或者字段的 XPath 当做爬虫获取信息的依据，来获取所需的信息。

**8.2.3** **Mahout**

Mahout, 意为驯象人，是 Apache 基 于Java 的开源机器学习库，里面高效地实现了多种 经典的机器学习算法，比如 KMeans 聚类算法、LDA 聚类算法、SVN 向量机分类算法等。 它作为Java 项目的一个类库，既可以运行在本地，也可以运行在分布式系统，因为 Mahout 基 于 Hadoop 实现的，这是它最大的优点，将很多原来运行在本地的方法改成了MapReduce 的模式，通过这样的办法大大提升了算法可以处理的数据量和处理效率。本系统中使用 Mahout 中的朴素贝叶斯算法来实现文本的分类。

**8.2.4** **朴素贝叶斯算法**

朴素贝叶斯是一种非常简单的分类算法，通俗地来讲，比如看到一个人，我们要判断是 男人是女人，方法很简单，根据以往经验，就是看头发长短、服饰、声音，如果那个人有一头长 发，穿着裙子，说话声音比较细，那十之八九就是女人，当然也有可能男扮女装，通常情况下 我们会选择概率比较大的那个类别，这就是朴素贝叶斯的思想，根据以往经验来预测信息。

朴素贝叶斯分类的定义：

。

首先，设x={a₁,a₂,…,an} 为一个待分类项的集合，其中每个a 为 x 的一个特征属 性，通俗的理解为文本中的一个单词。其次，再假设有一个类别集合y={b₁,b₂,…,bm}。

再次，计算P(b₁|x),P(b₂|x),…P(bm|x), 即待分类项属于类别y 的概率。最后，如果

P(b₄)=max{P(b₁|x),P(b₂|x),…P(bm|x)}, 则x ∈b₄ 。 那么问题来了，我们如何得到

P(b₁|x),P(b₂|x),…P(bm|x) 呢，其实可以转换一下方法，直接方法行不通就绕道而 行，方法是：第一，手工收集大量的训练样本集并分好类，训练集的质量决定着分类的效 果，所以需要耐心认真的分类。第二，计算训练集中每个特征属性对应到每个类别的条

件概率，其中P(ax|b₁) 的计算方法为特征属性a, 在 b; 类别内出现的次数除以b; 类别内 样本的个数，最终形成特征属性到分类映射的分类模型。第三，因为 P(b;|x) 是算不出 来的，但是根据贝叶斯公式 ,假定各个特征属性之间的条件独 立的，那我们就可以转而去求 ,因为 P(x) 对于所有类别都是常数，因此 只需要取分子 P(x|b₁)P(b₁) 值最大就可以了。最后将待分类数据做P(x|b;)P(b₁)=



P(x|b₁)P(b₁) 即为待分类数据的所属类别。

8.2.5 文档向量

本系统主要功能是根据朴素贝叶斯算法将文本信息分类，既然是算法，那肯定需要数 字，我们如何将文字变成数字来使用算法呢，在 Mahout 中使用向量这一概念，将文档向量 化的常见方法是VSM(Vector Space Model,向量空间模型)。其基本原理是将所有文档看 成一个具有N 个没有重复单词的集合，每个单词被分配一个唯一编号，每个文档就变成了 一个具有N 维的向量，每个单词的编号，就是该单词所在的维度。比如 people 这一单词被 编为2000,那么文档集合中所有包含 people 单词的第2000维度上的主键就是 people 。当 然有主键就有值，向量维度上的值就是单词在每篇出现的次数(Term Frequency,词频),叫 做 TF 权重。至此，我们就可以把文档转换成数字来使用朴素贝叶斯算法了。

8.2.6 TF-IDF 改进加权

在文本分类中，某些领域的专有词对分类结果非常有利，我们一看到就会知道该文本大 致属于哪一类，比如“原子弹”,我们会把它归为军事一类。然而一些无意义的单词将会严重 影响分类结果，比如“爆炸的原子弹”中的“的”字，在所有文章中都有，在每篇文章中占的比 例几乎最大，算法会将它归为出现“的”字最多的类别，显然结果90%不会正确。为了不让 这些停用词影响分类结果，它们在文章中的权重就必须要小，而使专有词的权重增大。在文 本分类中最常使用的就是IDF(Inverse Document Frequency,逆文档频率),公式为 IDF = ,其中N 为文档集合中文档的个数，DF 为拥有该单词的文档的个数，文档个数越多 说明该单词在分类中的意义越小。我们重新定义文档向量中每个维度上的值为 W=

TF ·

**8.2.7** **中文分词**

在英文文章中，区分单词的方法是通过空格，所以英文文本分类会很容易地拆散为一组 单词集合，比如“I am a gril”,计算机很容易将句子拆分为“I”“am”“a”“gril”。但是中文文档 是没有空格的，比如“我是个女生”,计算机没有办法识别“个女”是个单词还是“女生”是个单 词，中文中也没有什么其他自然分隔符能将文章变成单词或字的集合，因此中文分词要比英 文分词困难得多。但是如今的各方各面均已离不开中文分词，比如我们打字时用的输入法、 百度搜索等。但是如何做到将中文分为单词呢，最简单的办法就是将词典中的所有词在文 章中逐字搜索直到文章结束，但是假如一篇文章只有几行长，也要将大约三十六万个词遍历 一遍，效率相当低下。随着人们不断的研究，至今已研发出多种中文分词工具，比如庖丁解 牛、IKAnalyzer 等都是不错的分词工具。分词是文本分类重要的一步，分词的效果直接影 响分类结果。本系统采用IKAnalyzer 因为它的优势是不仅速度快而且还于有一 定的歧义 识别能力。比如，“质量和服务”,在切分的时候可以有两个候选结果集，分别是“质量”“和” “服务”,也可以是“质量”“和服”“务”,如图8-1所示。



cterminated>test [Java Applicatioon]J F:Vava\ire7\bin

质量和服务

质量|和服|服务|

图8-1 IKAnalyzer分词

目前 IK 在测试数据统计中可以做 到95%的准确率。除了准确率以外，管 理它的扩展词典很容易，添加 IK 的词库 很简单，如果有新词就可以往词典中加， 只需在配置文件中写好词典的位置，它 的停用词库也很好管理，有什么新的停

用词就可以加入停用词词典中，同样只需在配置文件中正确配置词典位置即可，它不需要什 么词频度、词性等等复杂的定义，所以对普通用户来说的入门的门槛比较低。它的配置也是 几个分词器中最简单的一种，更容易被大家所接受。



8.3 需求分析

**8.3.1** **系统功能**

大数据的价值我们已经了解，那么大数据从何而来呢?我们不是政府机构，也不是阿里 巴巴，没有海量的数据来进行分析。对于我们，数据的来源就是互联网，数据获取的方式多 种多样，网络爬虫是比较好的选择。对于普通用户来说，没有学习编程的经历，编写爬虫是 一件不可能的事，因此开发本系统的目的就是为普通用户提供一个获取信息的便捷的平台。 但是由于互联网中的页面是具有商业价值的，各种广告的投放以及许多杂乱的信息充斥着 页面，通用的爬虫往往抓取的信息不太准确，例如我们要搜索家居的信息，结果返回的是一 些服饰信息。针对以上情况，系统采用三种爬虫模式：分页方式、起止页方式、通用方式，用 户可针对不同的网站进行不同的选择。

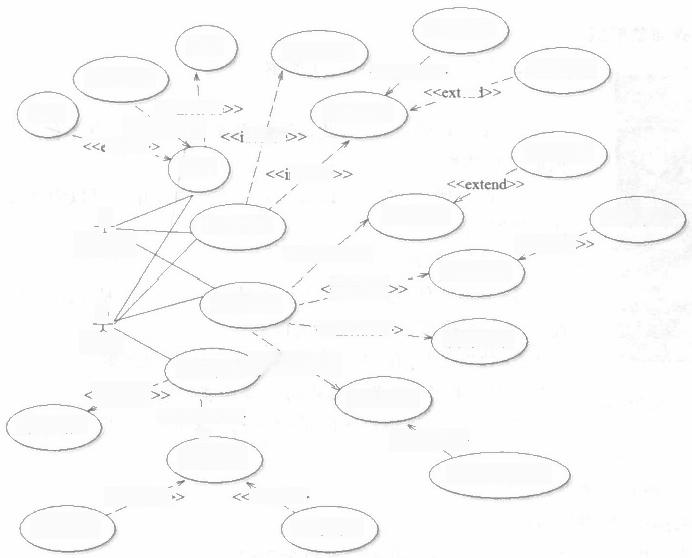
无论是为了百度排名，提高网站竞争力，或是维护企业形象，或是为了迎合用户的审美、 体验，或是为了更大的商业价值，每个网站都会经历改版阶段。然而对于普通的爬虫，网站

第8章 大数据在电子商务数据分析中的应用

的改版就面临着要重新改写大量代码，重复的劳动势必会增加人力的成本和时间成本，我们 就这一问题提出解决方案：系统采用B/S 架构，将爬虫代码封装到后台，将参数界面展示给 用户，普通用户不用了解代码如何实现，只需花费几分钟的时间学习一下 XPath 和简单了 解系统流程即可使用。XPath 是一种 XML 的语言，它是一种用来定位标签元素路径的语 言 。XML 的是树状结构的，XPath 提供在数据结构——树，中找寻元素节点的能力。 HT- ML 与 XML 类似，都是由标签元素组成的树状结构文件。对于 HTML 页面同样可以使用 XPath 进行分析，XPath 只需通过一条路径， 一次就可直接定位所需的信息，而其他页面解 析器一次只能确定信息的范围，通过几次解析才可以定位信息，因此相比较而言，XPath 具 有强大的优势。

朴素贝叶斯算法简单易学，而且分类效果较为理想，将作为本系统的分类算法。数据分 类也是采用简单的界面操作，选取爬虫获取到的数据，交给后台处理，最后将分类结果保存， 并且以图表的形式给用户直观的展示。

以往的数据信息的获取与分类是分开进行的，不利于普通用户的使用，我们打破了这一界 限，将两部分整合到一个系统当中。采用B/S 架构，可以随时随地都登录系统浏览业务，系统 维护简单，用户不必下载客户端，只需网页操作即可，而且可以通过添加网页即实现功能扩充 功能。用户登录界面后填写爬虫规则开始爬取数据或选择已经存在的爬虫规则，爬虫爬取完 毕之后通过界面来选择需要进行分类的数据，将数据提交给分类系统，分类系统使用已经训练 好的模型开始分类文本，或是用户自己重新上传训练集，并训练出模型。最终用户可通过图像 来直观地感受到分类的效果。用户在友好的界面，简单的操作中完成对数据的获取、分类以及 分析。系统用例图如图8 - 2所示。

注销

忘记密码

<<include:

注册

<<extend> extend>:

登录

修改规则

添加规则

<<extend>>

end

查看规则

nclude:

nclude

删除规则

训练数据

9

普通用户

爬虫管理

<<include>>

上传数据

<<extend 查看数据

删除数据

9

管理员

<inchude

添加用户

<inchude

算法管理

<<inchude>:

训练结果

|  |  |
| --- | --- |
| 用户管理 include>  <<include>>  Y  查看用户 | 数据分类  <<extend>>  查看分类结果 |

<<extend

修改用户

extend>>

删除用户

图8-2 系统用例图

**8.3.2** **系统界面**

**1.** **界面结构概述**

本系统界面采用 Framset 来将页面分为上、下、左、右四个部分，上面为系统 logo 部分， 下面为版权信息，右面为菜单栏，左面显示信息页面，给习惯看屏幕左中部分的用户直观的 展示。界面如图8-3所示。

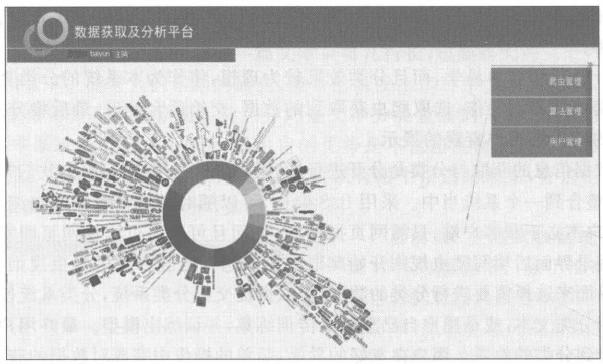


图8-3 系统界面

**2.** **界面功能概述**

系统界面功能，如图8-4所示。

液虫管理 (1)爬虫管理选项

海知点列

直香烘可 ① 添加规则：用户从此选项的页面中填入网站基本信息及该网站

算法管理 抓取信息的规则。

生传文件

② 查看规则：用户可以查看填写的规则，并且可以对规则进行修

就据分析

块 改，删除操作。

(2)算法管理选项

① 上传文件：用户从此页面上传训练集，必须是压缩文件，用户从

清烟用P

童善研产

此页面上查看训练集训练后的信息。

② 数据分析：用户从此页面查看上传的训练集，并且可以对训练

集进行删除操作，对训练集进行训练功能也在此页面上。 图8-4 界面功能视图

③ 数据分类：用户从此页面上选择要进行分类的数据进行分类，

并查看分类结果。

(3)用户管理选项

① 添加用户：添加用户。

② 查看用户：修改用户、删除用户、设置管理员全在次选项中。

③ 个人信息：用户可以在此页面修改个人信息。

第8章 大数据在电子商务数据分析中的应用

④ 修改密码：用户可以在此页面修改个人密码。

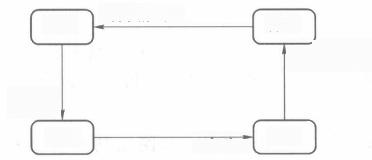


**8** **.** **4** **概** **要** **设** **计**

**8.4.1 系统模块设计**

该系统共分四个模块：登录模块、爬虫管理模块、算法管理模块、用户管理模块。各个模

块之间的关系如图8-5所示，系统总模块图如图8-6所示。



算法

《爬取信息》

《登录》

系统

《分类展示》

《启动》 |

用户

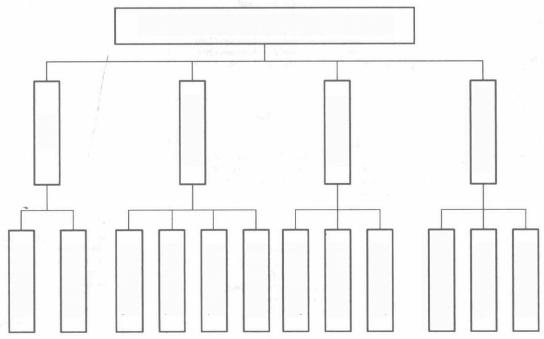
爬虫

图8-5 系统模块关系图

(1)登录模块：通过用户名与密码登录系统，注册新用户，找回密码，通过注销账户登出

系统。

(2)爬虫管理模块：通过用户填写的爬虫规则，将规则保存到数据库中。启动爬虫时， 系统将从数据库中检索出爬取网站的所有规则交给爬虫，爬虫将根据特定网站的链接规则 获取页面链接交给线程，将页面信息获取规则同样交给线程，线程通过信息获取规则获取页 面信息，最终将获取到的信息存入数据库。爬虫结构设计如图8-7所示。



数据分类

数据分析

上 传 训 练 集

删除规则

修改规则

添加规则

修改密码

删除用户

用户管理

查看用户

添加用户

用户登出

登 录

用户登录

大数据电商数据获取及分析平台开发

算 法 管 理

爬虫管理

图8-6 系统总模块图

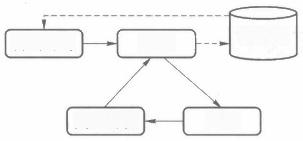
(3)算法管理模块：系统采用朴素贝叶斯算法来对网络爬虫获取到的信息进行分类。

朴素贝叶斯属于监督学习，需要训练集训练分类器得到预测模型，并通过预测模型将网页信 息进行分类，分类结构设计如图8-8所示。

(4)用户管理模块：普通用户无权使用此项功能，管理员可以对所有用户进行增、改、查

大数据技术及行业应用

的操作，但不可以删除自己。



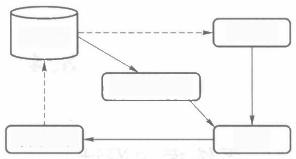
爬虫规则

页面解析 网站

数据库

线程

图8-7 爬虫结构设计



数据库

待分类数据

分类结果

训练集

模型

图8-8 分类结构设计

**8.4.2** **数据库设计**

**1.** **概念结构设计**

网站基本信息表(baseruledetail) 和主表信息规则表(mainruledetail) 与主表链接规则表 (mainanchordetail) 之间都是一对多的关系，即要抓取一个网站需要多条链接的获取规则和 多条信息获取规则。

用户表(users) 与上传文件表(uploadfiles) 和网站基本信息表之间是一对多的关系，即 一个用户可以上传多个文件， 一个用户同样可以添加多条网站抓取规则。

用户表(users) 与上传文件表(uploadfiles) 之间还有一个训练关系， 一个用户可以训练 多个文件形成多个分类模型， 一个文件只对应一次训练，中间训练信息表为(triandetail)。

上传文件表(uploadfiles) 与分类目录表(classifyitem) 表之间是一对多的关系，即一个 上传的文件中，分类的目录是多个的。

分类目录表(classifyitem)表与分类信息表(classifyinfo)表之间是一对多的关系，即一 个分类目录下包含多个数据分类的结果。

综上所述，本系统的数据库设计如图8-9所示。

id

category nulename baseurl

PPRtag startpage

endpage

maiutablesum

tkublevun

layer

regex

pe

id

id

usema

password isadmin

hirhday email

name

sex

registerdate

id <pi>

baseruledetail

spie ntegsr ≤Me

Integer

Variable characters ([200)](#bookmark183)

Variable characters (2000)

Varsble charactn ( [100)](#bookmark184)

Variable characters (100)

Variable characters (100)

loteger

lntege

Variable characters (10)

Variable characters (255)

FK B\_U

users

Spie lntsger ≤M

Variable characters ([20)](#bookmark185)

Variable characters ( [10)](#bookmark186)

Characters ( [1)](#bookmark187)

Date

Variable characters ([255)](#bookmark188)

Variable chancten ([255)](#bookmark189)

Characters ( [1)](#bookmark190)

Daie

mainruledetail

subpageid

cme Llegg

Integer

Variable characters (200) Variable characters (200) Variable charnacters (255)

ae

id

FK\_MR R=firltgere

id <pi

mainanchondetail

B spie lnteger ≤Me

EK MA

subpageid lnteger

name VWarisble chancters ([200)](#bookmark191)

filter Variable characters ([200)](#bookmark183)

regex Variable characters ([255)](#bookmark192)

squence loteger

id <pi>

elassifyitem

id

spie

Integer

sMe

traindetail

Variable characters (255)

Integer

Date &Time

classifyname

uphm

time

id<pie

correctrate Decimal(3)

fflecount Integer

ln

traintime *Date Time*

uploadftleid Integer

Userid Integer

cr u

FK

FK\_cn iN cr

\_

.

FK\_UPU

classiino

mpludfles

id

tuhlenamt mainableid subtableid

classifyid time

id cpie

spe Integer ≤Me

Variable characters (255) Variable charactens (255)

Wrigble chanctens (255) Imteget

Date &Time

spie Integer

id

name

path

type

uploahine userid

trainid

SMe

Variable characters(200)

Variable characters (2000)

Variable characters (200)

Date &Time

tete

Integer

id<pi

数据库 E-R 图

图8-9

<< 158

**第8章** **大数据在电子商务数据分析中的应用**

**2.** **表结构设计**

(1)baseruledetail 表

用于存放爬取网站的基本信息，分为三种模式：模式1必填字段是前四个字段，模式2 必填字段是1～7,模式3必填字段是1～4和10～11。表结构如表8-1 所示。

**表8-1** **网站基本信息表(baseruledetail)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 要求 | 描述 | 举例 |
| 1 | id | int | 必填 | 主键 | 1 |
| 2 | category | int | 必填，必显 | 模式分为1,2,3 | 1/2/3 |
| 3 | rulename | varchar(200) | 必填，必显 | 网站名称 | jrj\_stock |
| 4 | baseurl | varchar(2000) | 必填，必显 | 种子链接地址 | <http://news.qq.com/> |
| 5 | pagetag | varchar(100) | 必显 | 页标 | page |
| 6 | startpage | varchar(100) | 必显 | 起始页码 | 1 |
| 7 | endpage | varchar(100) | 必显 | 终止页码 | 6000 |
| 8 | maintablesum | int | 必填 | 主表链接层数 | 2 |
| 9 | subtablesum | int | 必填 | 子表链接层数 | 2 |
| 10 | layer | varchar(10) | 必显 | 爬取层数 | 2 |
| 11 | regex | varchar(255) | 必显 | 链接规则 | <http://stock.jrj.com.cn/> (.\*?)/[0-9]\*\.shtml |
| 12 | userid | int | 必填，必显 | 添加此规则的用户 | 1 |

(2)mainanchordetail 表

用于存放主表的链接规则，其中 subpagid 为 baseruledetail 表中 id 的外键，意义为：获 取一个网站的信息是通过多层链接的访问获取到的，即获取链接的规则也应该是多层的。 表结构如表8-2 所示。

**表8-2** **主表信息规则表(mainanchordetail)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 要求 | 描述 | 举例 |
| 1 | id | int | 必填 | 主键 | 1 |
| 2 | subpageid | int | 必填，必显 | baseruledetail表的外键 | 1 |
| 3 | name | varchar(200) | 必填，必显 | 主表链接名称 | mainnextfilter mainanchorfilte |
| 4 | filter | varchar(200) | 必显 | 主表连接XPath规则 | //a[@title='设计师] |
| 5 | regex | varchar(200) | 必显 | 通常情况不用此属性 |  |
| 6 | sequence | int | 必显 | 主表链接是第几层 | 1 |

(3)mainruledetail 表

用于存放获取主表信息的规则，其中 subpageid 为 baseruledetail 表中 id 的外键，意义 为：获取一个网站的信息首先要通过多层链接到达存有信息的页面，其次就是要根据信息页 面上信息的XPath 来获取信息，因此此表与 mainanchrodetail 中的 subpageid 都为 baser-

159 >>

大数据技术及行业应用

uledetail 中 id 的外键。表结构如表8-3所示。

**表8-3** **主表链接规则表(mainruledetail)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 要求 | 描述 | 举例 |
| 1 | id | int | 必填 | 主键 | 1 |
| 2 | subpageid | int | 必填，必显 | baseruledetail表的外键 | 1 |
| 3 | name | varchar(200) | 必填，必显 | 用于做主表的字段 | name |
|  | filter | varchar(1000) | 必填，必显 | 信息获取规则 | //div[@id='name'] |
| 5 | regex | varchar(100) | 必显 | 正则表达式，辅助规则 | [0-9]\* |

(4)uploadfiles 表

用于存放用户上传的训练集。表结构如表8-4所示。

**表8-4** **上传文件表(uploadfiles)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 要求 | 描述 | 举例 |
|  | id | int | 必填 | 主键 | 1 |
| 2 | name | varchar(200) | 必填，必显 | 文件名称 | train |
| 3 | path | varchar(2000) | 必填，必显 | 文件存放的实际地址 | /usr/baiyunsource |
| 4 | type | varchar(200) | 必填，必显 | 文件类型 | rar,zip |
| 5 | uploadtime | datetime | 必填，必显 | 上传时间 | 2012-2-139:38:28 |
| 6 | userid | int | 必填，必显 | 上传用户 | 1 |

(5)classifyitem 表

用于存放训练集的目录类别。表结构如表8-5 所示。

**表8-5** **分类目录表(classifyitem)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 要求 | 描述 | 举例 |
| 1 | id | int | 必填 | 主键 | 1 |
| 2 | classifyname | varchar(255) | 必填，必显 | 分类名称 | 军事 |
| 3 | uploadfileid | int | 必填，必显 | 所属训练集id | 1 |
| 4 | time | datetime | 必填 | 创建时间 | 2012-3-328:38:10 |

(6)users 表

用于存放用户的基本信息，其中username 、name 、email 是必填项，在使用忘记密码时使 用该些字段进行密码找回。表结构如表8-6所示。

**表8-6** **用户信息表(users)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 要求 | 描述 | 举例 |
| 1 | id | int | 必填 | 主键 | 1 |
| 2 | username | varchar(20) | 必填，必显 | 用户名 | baiyun |

第8章 大数据在电子商务数据分析中的应用

**续表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 要求 | 描述 | 举例 |
| 3 | Password | varchar(10) | 必填 | 用户密码 | 123 |
| 4 | isadmin | char(1) | 必填，必显 | 是否是管理员 | y |
| 5 | birthday | varchar(255) | 必显 | 生日 | 2000-1-1 |
| 6 | email | varchar(255) | 必显 | 邮箱 | 155050529@gg,com |
| 7 | name | varchar(255) | 必填，必显 | 真实姓名 | 白云 |
| 8 | sex | char(1) | 必填，必显 | 性别 | 1 女 0 男 |
| 9 | registerdate | date | 必填 | 注册日期 | 2015-04-13 |

(7)traindetail 表

用于存放训练集训练后的详细信息。表结构如表8-7 所示。

**表8-7** **训练信息表(traindetail)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 要求 | 描述 | 举例 |
| 1 | id | int | 必填 | 主键 | 1 |
| 2 | uploadfileid | int | 必填，必显 | 训练集id | 1 |
| 3 | filecount | int | 必填，必显 | 测试的文档个数 | 760 |
| 4 | traintime | datetime | 必填，必显 | 训练时间 | 2012-4-323:39:32 |
| 5 | userid | int | 必填，必显 | 训练的用户 | 1 |
| 6 | correctrate | double | 必填，必显 | 正确率 | 0.822 |
| 7 | modelpath | varchar(255) | 必填 | 模型存放路径 | /usr/baiyunsource |
| 8 | modelname | varchar(255) | 必填，必显 | 模型名称 | 27\_model\_1 |

(8)classifyinfo 表

用于存放使用模型对新数据进行分类后的详细信息。表结构如表8-8所示。

**表8-8** **分类信息表(classifyinfo)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 要求 | 描述 | 举例 |
| 1 | id | int | 必填 | 主键 | 1 |
| 2 | tablename | varchar(255) | 必填，必显 | 新数据所属的表名 | tiezi |
| 3 | maintableid | varchar(255) | 必填，必显 | 新数据所属的主表id | 1 |
| 4 | subtableid | varchar(255) | 必填，必显 | 新数据的id | 3 |
| 5 | classifyid | int | 必填，必显 | 所属分类 | 1 |
| 6 | time | int | 必填，必显 | 分类时间 | 2014-2-312:34:52 |

(9)classify\_item\_info

这是本系统中使用的唯一一个视图，可以指定用户所需数据，简化数据库操作，它用来 显示 classifyitem 表 和 classifyinfo 表的联合信息，主要用来显示分类数据来自哪个表，被分

161 >>

大数据技术及行业应用

到哪个训练集的哪个类中，用户可通过查询此视图就可以直观看到。表结构如表8-9所示。

表8-9 分类视图(classify\_item\_info)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段 | 类型 | 要求 | 描述 | 举例 |
| 1 | uploadfileid | int | 必填 | 训练集文件id | 1 |
| 2 | uploadfilename | varchar(255) | 必填，必显 | 训练集名称 | train |
| 3 | maintableid | varchar(255) | 必填，必显 | 新数据所属的主表id | 1 |
| 4 | subtableid | varchar(255) | 必填，必显 | 新数据的id | 3 |
| 5 | classifyid | int | 必填 | 所属分类id | 1 |
| 6 | classifyname | varchar(255) | 必填，必显 | 所属分类名称 | 军事 |
| 7 | tablename | varchar(255) | 必填，必显 | 数据来自哪个表 | tengxunxiwen |

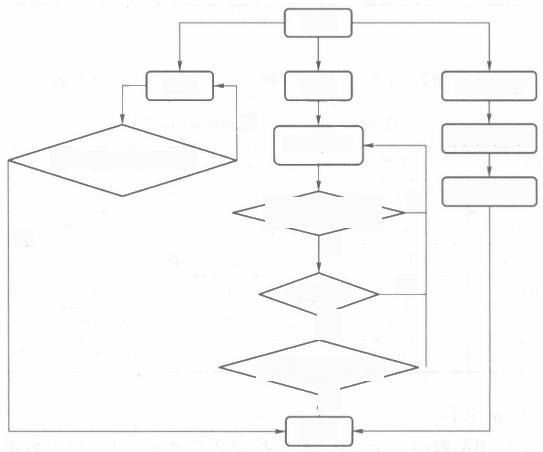


8.5 详细设计

系统的开发环境：Eclipse Java EE IDE for Web Developers(4.4.2);开发语言：Java 语言；系统 环境：Linux;框架选择：Struts2+Hibernate4+Spring3(B/S 架构);数据库：Mysql5。

**8.5.1** **用户登录模块**

用户通过用户名和密码进行登录，如果忘记密码可以通过用户名、真实姓名和邮箱来修改 密码，如果用户名、真实姓名、邮箱正确匹配则可以修改密码，否则不能修改密码。新用户可以 注册账户，用户名是唯一的，如果用户名重复则不能注册用户。新注册的用户是普通用户，没 有查看用户和添加用户的权限，需要管理员修改权限。用户登录流程如图8 - 10所示。



开始

注册

否

填写信息

用户名密码是否正确

填写新密码

是

密码-一致

是

必填信息是否为空

否

结束

忘记密码

填写用户名

用户名是否存在 否

登录

是

否

图8-10 用户登录流程图

**8.5.2** **爬虫管理模块**

爬虫管理包括对爬虫爬取规则的添加、修改、删除、查看、删除数据。其中添加功能有些 复杂，其网页获取方式包括三种模式。

(1)分页方式：适用于对有分页网站进行信息抓取，因为程序中使用了网站的分页标签 来进行网页链接的获取，此方法获取的信息较为准确，排除了网页中其他干扰链接，比如广 告，绝大部分为用户需要的数据。

(2)起始页方式：适用于下一页方式不适用的情况，没有“下一页”标签，或者标签名称 不为“下一页”的情况。给出起始页码，终止页码，但是此方法必须使用页标，即 URL 中 用 于分页的标志，有时不太好区分，需要用户细心分辨。此方法同样获取信息准确。

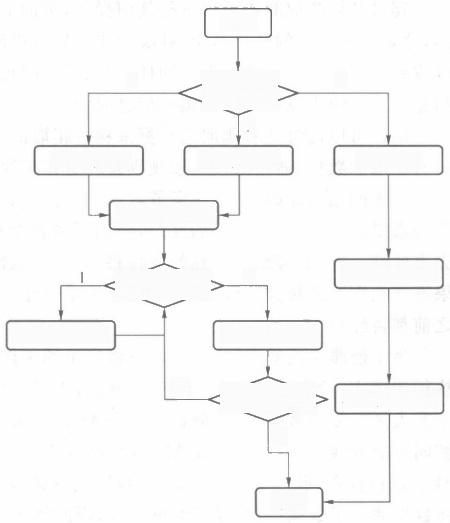
(3)通用方式：适用于普通页面的爬取，用户只需填入一个连接的正则表达式和爬取的 层数即可，该方法返回的数据可能有相当部分是与用户需求无关的。

对于前两种方式而言，其方法都是通过XPath 获取页面上的链接，分别进入每个链接中仍然 使用XPath 获取所需信息。值得注意的是，如果进入获取的链接后仍不能获得所需的信息页面， 就继续添加 XPath 来获取该页面上的链接，直到进入链接后可以获取信息为止，必须在最后一层 链接中填入“下一页”标签的 XPath 规则，程序通过判断有无该规则来判断是否是最后一层链接。 此外，这两种方式还可以获取子页面信息，因为进入子表页面需要链接，所以需要同样填写进入 子表的链接规则，与主页面相似，如果进入该链接后不能直接获取到子页面的信息，还需继续填

写链接规则，直到可以找到子页信息，在最 后一层链接规则中填写“下一页”的规则。 对于通用方式来说，只需填写链接的正则 表达式，爬取层数，获取信息的 XPath 即 可。该方式较为简单，但是信息获取不准 确成为一大弊端。

分页方式是根据分页标签进行爬取 页面，只可以获得具有“下一页”或者“下 页”标签的网站，而对于有些网站是通过 英文 Next 等相关字眼或者图片来设置 分页标签的，很难进行统一 。而使用起 止页方式可弥补这一不足，但是起止页 方式也有它的缺点，由于普通用户不了 解 URL, 对于有些冗长的 URL 不 知 道 该如何从中获取到分页的标志，因此这 三种方式各有利弊，需酌情选择。

添加规则完成后将自动创建表，用 来存放抓取的数据，表的字段即为填写 信息抓取规则的名称，表名称为抓取网

开始

3

1

抓取方式

2

起止页方式

下一页方式

普通方式

填写基本信息

填写基本信息

2

缝接或信息

填写信息规则

是否有子

否

填写链接规则

填写信息规则

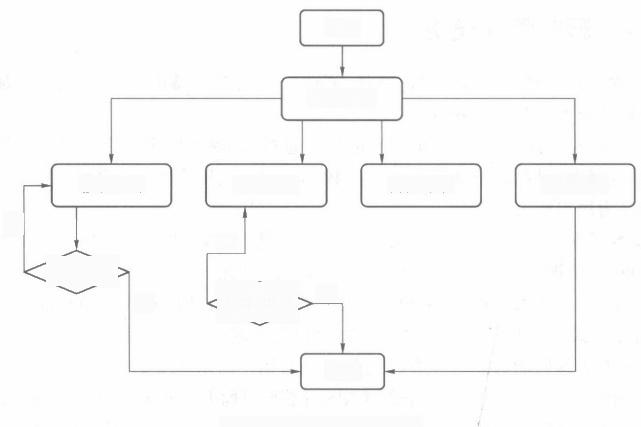
是

添加

图8-11 规则添加流程图

站名称。规则管理添加功能流程如图8-11所示。查看功能如图8-12所示。

图8-12 规则查看流程图



开始

查看规则

开始抓取

否

否

是否修改>

是

是否删险

结束

停止抓取

删除规则

修改规则

是

爬虫的修改同添加类似，可以修改网页基本信息，抓取规则，测试规则是否可用，但是不 能添加新的连接获取规则和信息获取规则。

爬虫删除时会将爬取的数据一并删除，如果该表内的数据被分过类，那么该表的分类信

息也会被删除。

爬虫开始爬取时不可以进行任何操作，页面上的所有功能都将失效，除了开启和停止操 作，这是因为爬虫的开启是由Ajax 异步向后台提出请求，开启爬虫，只有停在此页面上才可 以收到爬虫爬取完毕的信息。同样，爬虫的关闭也需要停留在此页面上来获取停止爬取的 信息。当系统中没有爬虫在爬取信息时，页面上将恢复所有功能。

用户可以通过查看爬取的数据来检查爬取的信息是否符合要求，如果不符合可以删除 所有爬取的数据，通过修改爬取规则重新进行爬取。

由于网站的原因， 一个页面或者多个页面上有相同的链接，致使爬虫获取到的链接很多 都是重复的，为了减少数据的冗余，使用了数据结构 Map 来存储已获取到的网页链接作为 去重的唯一标识，爬虫获取到的链接将会与此集合内的链接进行匹配，如果存在则丢弃，如 果不存在则将链接放入Map 中。此外为了减小内存的使用，网页链接在放入 Map 或匹配 之前都将经过 MD5 加密。

为了使爬虫高效率的工作，本系将爬虫的工作交给了多线程来进行，通过多线程的并发 执行的特点，充分利用CPU 的空闲时间，提高爬取速度。但是线程安全很难把握，数据库 中有大量重复的数据，经过分析发现是Map 的原因，Map 是非线程安全的，假如有两条抓取 相同页面的线程，其中一条在 Map 中查重，还没等放入 Map 中，被另一条线程打断，也使用 Map 进行查重，此时集合中没有重复项，则这条线程将信息插入数据库，而原来那条线程也 将数据插入了数据库中，导致数据重复，因此将 Map 的迭代和存取操作都设置为方法，加上 同步锁后，问题解决。

**8.5.3** **算法管理模块**

算法管理模块包括以下几部分。

(1)训练集的上传：上传的训练集必须是已压缩包的方式上传，格式为\* .rar 或者 \*.zip, 压缩名必须与其内文件名保持一致，因为程序只能判断压缩文件名是否相同，但压 缩文件的内部文件的名字程序无法判断，如果两个压缩文件内的文件名称相同，在解压文件 时会导致文件存在错误。程序中上传的压缩文件，和解压后的文件，算法的输入/输出文件 都存在实际的地址中，不会放到项目的目录中，因为项目重启可能会导致上传文件丢失，有 些文件太大不方便放入数据库中，因此放在实际地址有利于文件的备份，防止数据丢失。

(2)训练集删除：删除之后，与其为直接外键和间接外键的数据都将被删除。

对训练集进行模型训练：将上传的训练集传给分类算法进行训练，训练后输出的参数及 模型文件存放在实际的目录中。

(3)新数据的预测分类：将待分类的数据从数据库中读出并写入文件中，作为分类算法 的输入，将分类的结果直接保存到数据库中。

(4)分类结果的图表展示：通过条件筛选查出分类信息，并且可以查看分类内容的来源 信息。

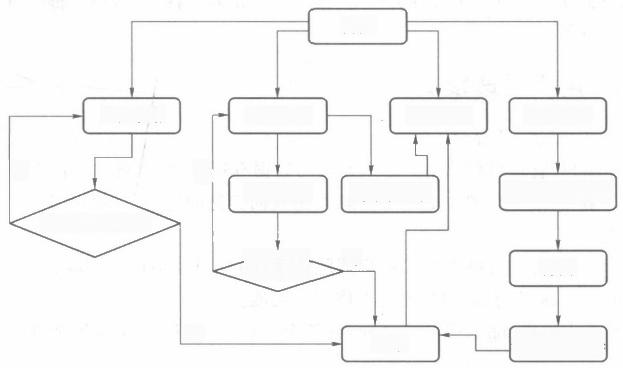
(5)获得发帖人数最多的用户：获得发帖人数最多的前20名发帖人并查看其基本

信息。

(6)将分类的结果导入训练集：形成新的训练集和训练模型，对爬虫获取的其他信息进

行分类，还可以从训练集删除导入的数据。

算法管理功能流程如图8-13所示。



开始

上传数据 查看数据 训练结果 数据分类

否

删除数据 训练数据集 选择待分类数据

是否为.rar或.zip

是否删除

查看分类结果

结束

分类

是

是

否

图8-13 算法管理流程图

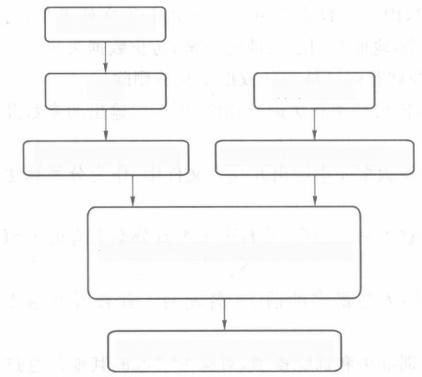
本系统文本分类器核心是使用Mahout 中的朴素贝叶斯分类算法，该算法简单易懂，分 类效果好，是普遍使用的分类算法。本系统的分类器分为三个阶段。

第一阶段：为分类算法准备样本集合，此阶段需要人工搜集大量文本信息，并根据特征

属性手工进行分类形成样本集，样本集的质量决定着模型的质量和最终分类的效果，因此要

尤为仔细。

第二阶段：训练分类器，此阶段不需要人工介入。首先要将训练集进行中文分词，去掉 一些常见的停用词，保留有意义的单词，最终将训练集内的每一篇文档都拆分为单词集合。 其次要将每篇文档进行文档向量化，即将文档集合的所有单词进行编码，作为文档向量的主 键，计算文档中每一个单词出现的次数作为文档向量的权值。再次将每篇文档向量的权值 TF-IDF 重新加权，降低无用词的权重，使之在模型计算中占有比较轻的影响。最后主要计 算每个类别中特征属性对该类别的条件概率，形成预测模型。



获取训练样本

分类

对每个类别计算P(bi) 对每个类别计算P(X|bi)P(bi)

计算每个特征属性对类别的条件概率

P(X|bi)P(bi)最大值即为X的类别

待分类数据

图8-14 分类流程图

第三阶段：使用模型分类新数据，此阶 段同样需要将爬虫爬取到的待分类数据进 行中分分词，去停用词，计算词频，TF-IDF 计算权重，最终形成文档向量。之后根据上 一步计算出的预测模型，计算待分数据的特 征属性在每 一个分类中的概率乘以类别的 概率，之后取最大值即为待分类数据应该被 映射到的类别。分类流程如图8-14所示。

Mahout 工具包中的朴素贝叶斯算法 的输入/输出都是文件，训练集大概有2万 个文件，算法需要将2万个文件读到内存中 计算并将结果写回文件中。另外，待分类的 数据同样需要以文件的方式被算法读取，这 样爬虫爬取的数据就不仅仅是2万个了，需

要将数据从数据库中读取出来写到文件中，读写硬盘是非常耗时的，如何将读写文件改为读

写内存还有待分析。

**8.5.4** **用户管理模块**

用户管理包括以下几部分。

(1)添加用户：管理员添加用户必须要填的数据有用户名、密码、邮箱，不必须填的信息 是用户生日。其中用户名、真实姓名和邮箱作为找回密码的凭证，必须填写。普通用户不可 以使用此功能。

(2)查看用户：管理员可以查看用户删除其他用户，但是不可以删除自己，管理员可以

设置普通用户为管理员，普通用户不可以使用此功能。

(3)修改个人资料：除了用户名之外，其他个人信息均可以修改，普通用户可以使用此

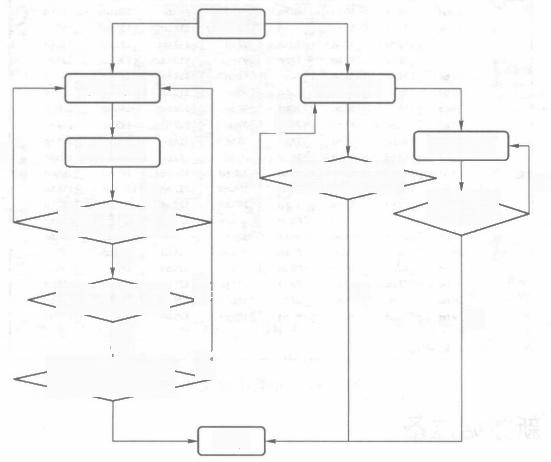
功能。

(4)修改密码：如果输入的原密码与数据库中的密码匹配，则用户可以修改密码，否则

不可以，修改密码成功后必须重新登录，普通用户可以使用此功能。

(5)用户的权限管理：本系统采用两层权限机制，分为普通用户和管理员。普通用户除 了不可以对其他用户进行操作外，其他功能均可以使用。当前用户可以查看自己的资料信 息，修改自己的密码后需要用新密码进行登录。用户管理功能流程如图8-15所示。

第 8 章 大 数 据 在 电 子 商 务 数 据 分 析 中 的 应 用



开 始

添加用户

否

是 填 写 信 息

是

是 否 删 除

用户名是否存在

否

否

密 码 是 否 一 致

是

是

必填信息是否为空

结 束

是否保存修改

删除用户

修改用户

否

是

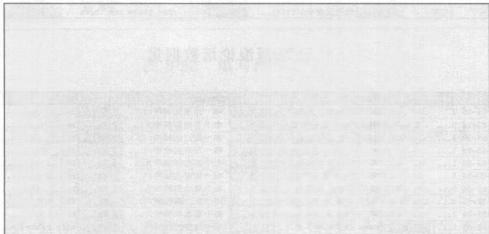
否

图 8 - 1 5 用 户 管 理 流 程 图

8.6 系统测试

**8.6.1** **训练集准备**

本次测试系统使用的是搜狗标准的训练集，其中包含10个分类，分别是IT 类、财经类、 健康类、教育类、军事类、旅游类、上访类、体育类、文化类、招聘类，如图8-16所示。每个分 类中有大约200个文本文档，如图8-17所示，出于测试需要，此次试验又添加了家居装修类 别，用来获取家居装修的信息，同样手工搜集了大约200个文本。到此为止，数据分类的训 练集准备工作已经完毕。



2010/4/714:59 2010/4/714:59 2010/4/714:59 2010/4/715:00 2010/4/715:00 2010/4/715:00 2010/4/1011:49 2010/4/715:00 2010/4/715:00 2010/4/715:00

文件夹 文件夹 文件夹 文件夹 文件夹 文件夹 文件夹 文件夹 文件夹 文件夹

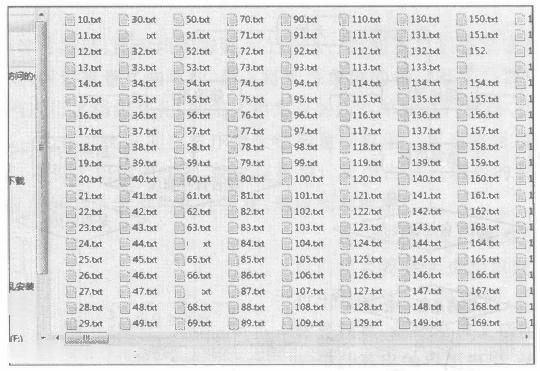
π

财经 健康 数育 军事 旅游 上访 体育 文化 招聘

图 8 - 1 6 训 练 集 分 类 目 录

167 >>

大数据技术及行业应用



31.ty

.tat 153.bt

64.tx

67.tb

amc

E:)

200个对象 状态： 路已共享

图8-17 分类中的文本

**8.6.2** **新数据准备**

进入互联网时代以来，越来越多的网民从互联网中获取信息，也有越来越多的网民通过 社交平台来发布消息，这反而成为互联网信息的主要来源。社交平台主要有 QQ、 微博、论 坛等，其中论坛具有 QQ 的即时通信功能，又有微博可以发表文章的功能，更是一个开放交 流的平台。论坛中的用户不一定互相认识，但是都是因为一个话题相互讨论，比如家居装修 话题，大家会把自己的装修心得拿出来交流，其潜在的商业价值显而易见。本系统数据收集 了大致有三个网站： 一是新浪论坛20000条，如图8-18所示；二是天涯论坛20000条，如图 8-19 所示；三是一起装修论坛10000条，如图8-20所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | detail |  | webatte | ?gaintim |
|  | 飘基的秋千 16 b. | S2篇贴子 0位好友… 94 b | 模主发表于2015642 … 4 | <hutp://bbs.J1eju.com/chrea>…43 | 2015-06-0520:30:18 19 b. |
|  | 丛中第.102608 17.b.L | ²6贴子 0 位 好 友 … 3 | 楼主发表于2015642 … 1 … | <http://bbs.Jtaju.com/threa>… 43 | 2015 :30 |
|  | ty1928104522 13 b..J | 13篇贴于 0 位 好 友 … 9 3 b | 模主发表于2015641…4 | http://bbs.Jiaju.con/threa… 43 k | 2015-06-0 20:30:2 19 |
|  | 场劳伦斯 13D../ | 4 9 3 子 0 位 好 . . 96 b | 模主发表于2015641…2 Rh | <http://bbs.Jtaju.cam/threa>… 43 | 2015-06-0520:30:2 9克， |
|  | 安徽扬于地板 19b,,5 | 21算贴子 0 收 好 101 B, | 楼主发表于2015641.…5Rh,. | <http://bha.JiaJu.com/threa>… 43 b. | 2015- 20:30:2 19 B |
|  | 华贝软装 3b,[ | 37篇贴子 0位好友…104 b | 楼主发表于2015641…2阳， | <http://bbs.J3aJu.com/three>… 43R | 2015-06-0 20:30:23 9b. |
|  | 直高钟诚装饰小唐25 b. | 21篇贴于 0 位 好 友 … 9 3 b | 棱主发表于2015641…。 | <http://bbs.J1aJu.com/threa>… 43D | 2015-06-0520:30:22 9B. |
|  | 劳卡衣柜Cnrlaty 20b | 982篇贴子口收好 . 4b, | 楼主发表于2015641…1 Kb… | <http://bbs.Jiaju.com/three>… 43B. | 2015 20:30:23 19t |
|  | 139xxxx4934 12b/ | 43第贴子 0 位 好 友 … 9 4 b . | 模主发表于2015641…6 | http://bbs.)iaju.cum/three… , | 01 9 b. |
|  | 江中红中式蔬修22 b,u | 693篇贴子 0位好.… B, | 楼主发表于2015641…2 | <http://bbr.jiaju.com/thren>… 42 B。 | 2015-06-0520:30:24 9b |
|  | anhua2014425 4B,J⁴ | 算 子 0位好友.. 2b. | 模主发表于2015631…1kB… | <http://bbs.jiaju.com/threa>.. B, | 2015-06-0 20:30:2 9b. |
|  | 用户3224181593 17b.J | 1算贴子 0位好友.….93b, | 楼主发表于2015641…1R | http://bha.]1aju.com/threm… 43 b, | 2015-06-0520:30:26 19 |
|  | 可商诺德装饰 12bJ | 311篇贴于0位好 … 101 B | 楼主发表于2015641…2… | <http://bhs.jiaju.com/thres>… 3 | 2015-06-0520:30:27 9k |
|  | 浙江多意电题.,1b. | t第 z 0 收 好 灰 2b | 楼主发表于2015641…2地 | bttp://bba.]1aju.com/threa… a | 2015-06-05 20:30:2 19 b |
|  | 山高高 10 b,J | 034前贴子 0位好.….96 b, | 楼主发表于2015641.…2 | hetp://bha.]1aju.com/threa… 3.b. | 2015-06-0520:30:28 19 |
|  | 劳卡衣枢Chriaty 20 b. | 982算贴子 0位好， .. 94 | 楼主发表于2015641...4, | <http://Bb:.31aju.com/threa>. B | 2015-06-0520:30:2 |

**图8-18** **新浪论坛数据集**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | \*ame | time\_Lab1ao | t1FL | | CeE | website | gaint1m |
|  | 龙凰茶苑器 互 b , | 时间：2015-04-061…29 b. | OPP0 N3乎机给抢怎…32 b | | 唯一指定Q接待：(…3Rb, | http://bbn.t1….45b | 2015-06-06 38:3 |
|  | 龙照茶苑员5b.] | 时间：2015-02-160.…29b | 联想VIBE Z2平机被…46 D | | 每个嘉边部携带大…2次， | http://bba,t1.…5上 | 2015-06-0621:57:12 |
|  | 龙的42 B, | 时间：2015-03-301 … 29R | opno 3手机给抢怎…32 b | | 唯一指定的接待：(…3K | http://bba.t1..45E | 2015-06-0621:41:13 |
|  | 龙解42 Lb. | 时间：2015-03-261…29 b | wivo Xshot乎机掉…35 b | | 唯一指定Q接待：(…3K | http://bbe.ti.45h | 2015-06-0621:43:34 |
|  | 龙行天支下 5b.J | 时间：2015-02-201…29 b. | 草果6手机掉了怎么…32 b, | | 每个离边都携带大…之 | http://bba.ti…45 b. | 2015-06-062156:29 |
|  | 龙行天支下5 b,J | 时间：2015-02-162…29 b | 草果6手机掉了怎么…规及 | | 唯一指定的按待：(…3 | httpi//bbs.t1…45 b… | 2015-06-0621:56:58 |
|  | 龙行天支下 5b. | 时间：2015-02-060…29 b | 室尼Z3手机掉了怎 … 3b | | 每个嘉边部携带大 … 2 … | hetp://bbs.t1…45 b | 2015-06-0621:59:14 |
|  | 龙行天支下 5b, | 时间：2015-02-051…39 b. | 小米4乎机被抢怎么… b… | | 每个裹边都携带大…2 b… | hetp://bba.t1…45 | 2015-06-0621:59:37 |
|  | 龙行天支下 点 b , | 时间：2015-02-041…29 b | 草果6乎机掉了怎么……30 b. | | 每 个 裹 边 部 携 带 大 … 2 R | http://bba.t1.…45. | 2015-06-0622:00:01 |
|  | 北行天支下 b.] | 时国：2015-02-031. 29b. | vivo X5手机掉了怎…32 b | | 唯 一 指定0按待：( … 3 | http://bba.t1.…45 | 2015-06-0622:00:18 |
|  | 龙行天支下 5b, | 时间：2015-02-011 … 29b | OPPoR3手机掉了怎…31 b | | 唯一指定Q接待：(…3K | http://bbs.ti.…45 b | 2015-06-0622:01:01 |
|  | 龙虾程允 2b. | 时间：2014-08-021…29 k | 为何女神眼略看上…43 b. | | 近日，涯椒上一篇…3 R | httpi//bbs,ti…44 b | 2015-06-0622:55:47 |
|  | 龙睛装怖12132D, | 时间：2014- 1- 221 … 29 b . | 飘宙可以做成这样的 28B. | | 11 b | http://bbe.t1…45 b… | 2015-06-0622:21:20 |
|  | 龙渊释天下 5b, | 时间：2014-10-021…29 b… | 装修必看-部市果园… | 32 b. | 一、果购计划心 … 2 | [http://bba.ti..4s b | 2015-06-0622:34:10 |
|  | 龙同学通录点D, | 时间：2015-04-061 … 29 b … | 华为p⁷乎机被偷怎 … 39b … |  | 唯一指定Q接待：(…3Kb, | hetpt//Ba,tirn.45 b,. | 2015-06-0621:38:40 |
|  | 龙同学逢录 6b. | 时间：2015-03-<132...29>b.. | 华为p7乎机被的怎， 33 b | | 唯一指定Q橡待：(…3.R | hetp://bhs,t1…45 b. | 2015-06-0621:49:48 |

**图8-19** **天涯论坛数据集**

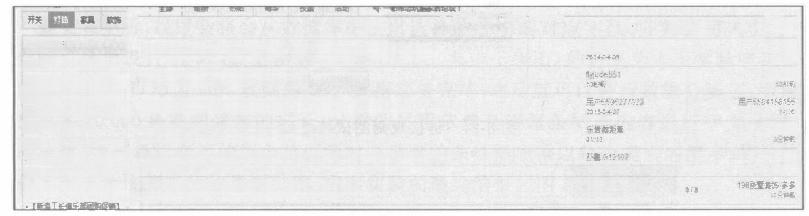
<< 168

**第8章** **大数据在电子商务数据分析中的应用**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  | 4 Laiso tine | content | website | | qadneine |  |
|  | 4 月 2 日 。 一 起 网 答 装 签 害 69 b, | | P 律 料 | 2012-431…B, | 本 帖 最 后 由 网 … 5 K , | http://bbs,17h…42B | | 2015-06-10…②B. |
|  | 也 器 6b. | | 阴态除料 b | 2012-3-15 …2b | 本 帖 最 后 由 网 … 2 K  , | hctp://bbs.17h… | 47b | 2015-06-10…2b |
|  | 四上网 的意3:水电改。 . 6 | | 郑小丫 | 2 0 1 1 - 9 - 1 2 … 至 b | ,本 帖 最 后 由 郑 … K | http://bbs.17h…47 b |  | 2015-06-11…2b |
|  | 道请“宽美装峰计划大关… b | | 壶冥艹 b。 | 2 0 1 1 - 6 - 3 1 . … 豆 b | ,前 光 非 常 感 ”. | http://bbs.17h…46b, | | 2015-06-11…2b. |
|  | [ 一 检格 KM6A3. 04 Ba | | 网态淋料 | 2011- 5- 27 … 豆b,J | 本 帖 最 形 由 网 … 2 K , | netp://bba.17h…46b | | 2015-06-11…22 |
|  | [一起检 楼 起 网 6 月 3 . 。 95 b, | | 闻峦除料 b | 2011-5-27 …②bJ | 本 帕 最 后 由 网 … 5 R h | hetp://bbs.17h…4b |  | 2015-06-11… |
|  | 一封 推 验 | 积 用 A b | 阿枣神料 瓦 | 2011-5-27 | 本 帖 最 后 由 . S K h | http://bbs.17h 46D | | 2015-06-11 2b |
|  | 超 检 楼 95h, | | 同密决料 | 2011-5-27 …2b | 本 帖 最 后 由 网 … 5 R , | nttp//bbs.17h…46 b.… | | 2015-06-11…2b. |
|  | 【 一 起维 楼 R 6 月 3 . 95B | | 即垂市料 b | 2011-5-27 …2b | ,本 帖 最 后 由 网 … R D , | hetp://bbs.17h…46b | | 2015-06-11…2b |
|  | 【 一 起检 楼 心 R A R 95b | | 阿态涂料 bJ | 2011 - 5 - 27 . 立b可 | 本帖最后由 5RD,, | hetp://bbs.17h… | 40D, | 2015-06-11.…2b |
|  | 4 月 9日 阴 市 神科立旺情 6b | | 阿枣神料 b | 2011-4-25 …9b | , 本 帖 最 后 由 网 … 5 R b , | hctp://bbs.17h… | 46 b, | 2015-06-11…9 b. |
|  | 4 月 2 9 日 西 津 科 立 证 性 96 b, | | 四志除料 | 2 0 1 1 - 4 - 2 5 … 豆 b | 本 能 最 后 由 … R b  , | nttp://bbs.17h… | 36b | 2015-06-11…D |
|  | H2 料真狂抢 6k | | 闻枣除料 | 2011-4-25…… | 本 帖 最 后 由 … 5 R h , | hetp://bbs.17h… | 46B, | 2015-06-11. b |
|  | 月2 日 间 科 市石 CE | | 间态涂料 b | 2011-4- 25 … 豆b,J | 本 帖 最 形 由 阿 … 三 R D , | hetpt//bbs.17h… | 46 D | 2015 - 06 - 11 … 王b |
|  | 4 月 2 9 日 间 市 神 和 卫 征 枪 ， 86b, | | 闻杰涂料 b | 2011-4-25…2b | 本 帖 最 后 由 … 5 , | hetp://bbs.17h… | 46 D | 2015-06-11…2 |
|  | 4 H 2 9 日 市 除 料 范 安 位 ， 66B | | 门态库科 5 | 2011-4-25…b | 本 帖 最 后 由 网 … 5 R b , | httpi//bbs.17h… | 40b | 2015-06-11…2b |
|  | 检 活 i n g1 一 起 质 端 。 11C | | 煮其种 | 011 b | t ,17 |  | 6E₄ | 01 |
|  | 拉 楼 活 动 i n q 一起检楼 1FB | | 壶冀艹 b,J |  |  |  |  | 2015-06-11…2b |
|  | 检楼活动 植 B | | 煮真艹 b,J | 2011-3-22 …2b,http://ww.17..2 httpi//bbs.17h…46 b.… | | | | 2015-06-11…9b. |

图8-20 一起装修论坛数据集

本次实验选择新浪论坛为数据的来源。用来分类的新数据是根据爬虫规则从新浪家居 (<http://bbs.jiaju.com/>) 中获取到的帖子。如图8-21 所示。



标 题



房 价 下 块 是 全 民 狂 欢 打 观 血 离 星 跌 还 有 多 远 限 病 ] 题 D1234.12

·520看家日1敢不数让地常

战 小 户 型 开 静 粉 连 工 作 区 紧 殊 静 间 ]

别 被 中 国 楼 市 想 注 您 然 了 ， 中 国 人 其 实 浅 解 了 几 十 年 欧 润 D12

用产5128099958

[ 6 月 1 0B】 张 棒 论 坛 打 卡 签 到 专 用 贴 0 1 2

天 时 地 利 人 和 一 样 不 缺 ， 楼 市 才 能 涵 序 等 进 做 到 最 好 润 D1234.5

·日凿路1装排干万部这么

1%题塑费物多多

六 大 减 系 决 战 工 期 载 派 工 艺 川 板 次 城 [ 紧 停 游 间 ]

·万区一心![新流玻排抢工长 · 老 田o 教守我与白都史先中 ,同米相柜法国年将的—寻味

·你有投诉我们来籍

·3499元地爱依璃断沙发，我购

·你难折扣新流地工长器季千人

回 复 / 查 看

18/342

4/1

最 后 发 表

如 显 玉

\*

se551

· 但天不伯果施地板寄小步

用中5504158155

*tg\*\**

树 优 息

砌/

1 0 分 \*

22150121

21149

20150

作 者

图8-21 新浪家居图

要爬取帖子信息，首先要填写爬取规则，然后交由爬虫去爬取数据，填写爬取规则的步 骤如下：

(1)打开系统的添加规则页面，选择抓取模式，因为该论坛没有“下一页”的按钮标志，

因此不适用分页方式，而是使用起止页方式，所以选择2“起止页方式”。之后填写网站的基 本信息，如图8-22 所示。此时其他两种模式的信息填写区域是不可编辑的。

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | |
| 网站名称 | | | | 种子链接 | 起始页标 | 起始页码终止页码 终止页标 | | | |  | | |
| sina | | | | [http://bbs.jia ju.com/](http://bbs.jiaju.com/) | forun-5121- | 1 | 500 | .html | | 提交 | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

图8-22 填写网站基本信息

(2)点击下一步，由于我们要获得所有帖子的链接，通过链接进入该帖子，在该帖子页 面内进行帖子信息的抓取，所以我们在提示框中选择1“链接”,用来填写帖子链接的抓取规 则。填写的规则如图8-23所示，测试结果如图8-24所示。

(3)点击下一步，这一步代表我们已经进入某个帖子页面内，提示框内选择2“信息”,填写

帖子信息的抓取规则，对于此网站我们主要抓取用户名，帖子内容和发表时间。规则填写如图 8-25所示。

(4)提交表单，至此新浪家居的爬虫已经填写完毕，启动该爬虫，爬取数据。

169

>>

大数据技术及行业应用

|  |  |
| --- | --- |
|  |  |
| **测试连接** |  |
| **规则名称** | **过滤规则** |
| **链接过添规则** | //tbody[eid]/tr/th[eclass='common threadlist\_subject']/span[ei |
|  |  |
| **下一页过滤规则** |  |

图8-23 填写链接规则

[http://bbs jiaju.com/thread](http://bbsjiaju.com/thread)-

17960819-1html

http://bbs jiaju.com/thread-

17960808-L.html

httpu//bbs jiaju.com/thread-

17908675-Lhtml

http://bbr jinju.com/thread-

17960687-1.html

httpi//bbs jiaju.com/thread-

17960682-1.html

httpi//bbs.jiaju.com/thread-

17960274-1.html

http://bbs jiaju.com/thread-

17960671-1.html

httpi//bbs jiaju.com/thread-

17960616-1.html

http://bbs jiajucom/thread-

图8-24 链接规则测试结果

|  |  |
| --- | --- |
| **测试连接**  **规则名称** |  |
| **过滤规则** |
| name l/div[@id='dfview]div[1/table/tbody/trAd[@class='postauthorYcite/a | |
| detail | lldiv[@id='dfview ydiv(1Ytable/tbody/tr/td|@class='postauthorydl[@class= |
| content | lldiv[@id='dfview]div[1/table/tbody/tr/td[@class='postcontent] |

图8-25 帖子信息规则

**8.6.3** **训练模型**

上传前面步骤准备好的训练集，必须是压缩文件并且是 zip 或 者 rar 格式的，上传完毕 后如图8-26所示，程序会自动将上传的压缩文件进行解压操作，之后选择该训练集进行 训练。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 文件名称 文件费型 | | 文件地址 | 上传时间 | 操作 | 上传用户 |
| ain\_jiju | af | uar BaiyunSource/BYBSAT | 15-6-69:59-01.000 | 训练结果 | baiyun |
| 1 供 有 项 | | | | | |

图8-26 训练集列表

训练时经过文档的中文分词、文档向量化，计算每个分类下每个单词的先验概率，形成 最终的模型，并对最终模型进行测试，测试数据是从训练集中随机选择的20%的数据，测试 结果如图8-27 所示，正确率为88.6%。

第8章 大数据在电子商务数据分析中的应用

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 文件名称 | 测试文件个数 | 正确車 用户 模型地址 | | 模型名称 | 训练时间 |
| train\_jiajo | 797 | 0.886 | baiyun Ausr/BaryunSource/BYB | 27\_model\_1 | 15-6-69-5947.000 |
| 1供有项 | | | | | |

图8-27 训练结果

**8.6.4** **数据分类**

从界面中选择待分类数据的表名、字段名，以及模型名称。模型名称即为上传的训练集 结果的名称，表名为上面步骤获取到的新浪家居信息的表名，字段名为 sina 表内存放着帖 子内容的列名，如图8-28所示。

列名 Content ☑ 模型名称27\_model\_1

表名sina

☑[开始分类

图8-28 分类信息选取

提交表单后，程序会将会从数据库中读出选择的表中的对应字段，并将其写入文件中， 文件的目录为表名/字段名/主表ID/ 子表 ID.txt,其中子表 ID 的意义为本次实验中每个帖 子的 ID, 主表 ID 即为 sina 规则的 ID 。数据准备完毕，之后就可以进行数据分类了，本次实 验抓取了20000条新浪家居的帖子，分类的过程依然是将每条帖子内容进行中文分词，文 档向量化，计算每篇文档的每个单词到对应类别的条件概率乘以每个类别的概率，将计算结 果最大的视为该篇文章的最终分类，本次实验的最终结果如图8-29所示。

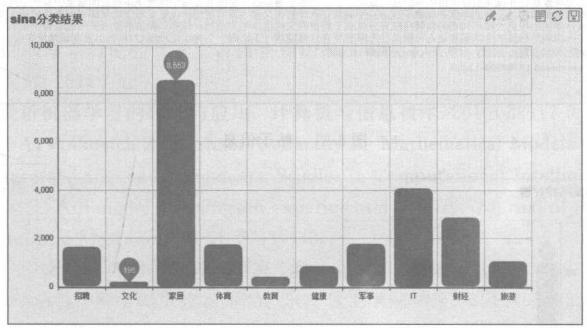


图8-29 分类结果

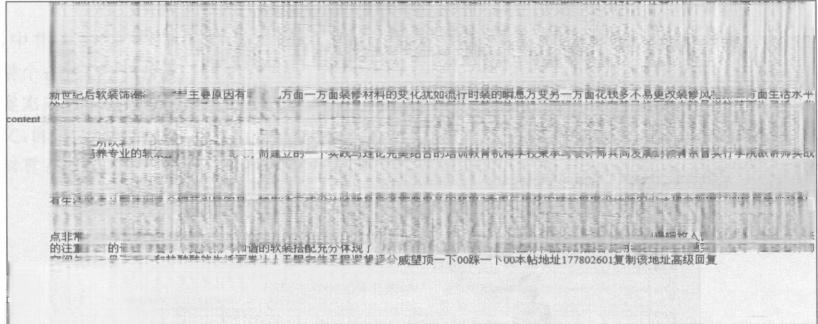
**8.6.5** **分类结果分析**

如图8-30 所示，根据分类结果显示，有8553篇帖子被分到了家居类别下，通过双击信 息条目可以查看该条目的详细信息，比如文章信息、用户信息等，如图8-31 所示。此外，还 可以查看发帖最多的前20个用户的详细信息，如图8-32所示。

**大数据技术及行业应用**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 模型名称27\_model\_ 1 | | 分费名种IT 表名sina | | 族106排育174,素居833;文化193;军事1733,健康43,教育3%封经2859招牌1621a1200 | | | |
| 操作 | 模型名称 | 分费 |  | 子丧D | 得分 | 到名 | 表名 |
| 主表D |
|  | 27\_model1 | 素居 | 32 | 9fAc161-4567-40a1-9415-8e22bf27be32 | 0.324875684766 | coated |  |
| 四 | 27\_model\_1 | 素居 | 32 | 0cb47221-00am-4154-8407-42112cf2a6 | 0297386016461 | content |  |
| 四 | 27\_model\_1 | 家居 | 32 | 8eb48b2b-5254-449d-a2me-49e68m21c99 | 0.274156151850 | conterd |  |
| 口 | 27\_mode\_1 | 家居 | 32 | 9ca8e6bb-0359-4138-443e-b951d714ba78 | 0.227393593120. | conlent | ima |
| 口 | 27\_model\_1 | 家居 | 32 | 91be196-7586-4806-9562-b69a9caddsn | 0221253564778 | conbent |  |
| □ | 27\_model\_1 | 家居 | 32 | ced4De-41af-4618-9495-43b4ae0418e | 0.216893857738 | omleat |  |
| 四 | 27\_model\_1 | 家居 | 32 | eda82e6.559m-4c1-89e4-74275456aBe | 0.202554314316. | centeat | ina |
| ■ | 27\_model\_1 | 家居 | 32 | 720c53e-m064b38-a7e-94404594312e | 0.200681898704 | conteat |  |
| … | 27\_modet\_1 | 素居 | 32 | 8e353002-0926-4596-8678-de264b188cBr | 0.196160280532. | conteat | tiaa |
| … | 27\_model\_1 | 家居 | 32 | b8be8c7-9889-4859-823fe719723Neff | 0.193152387536 | centea | ia |
| … | 27\_model\_1 | 家居 | 32 | 2cf34dbb-c668-4389-8546-40166dN775 | 0.191675096554 | coatenl | itnn |
| … | 27\_modal\_1 | 家居 | 32 | 5d50074-c2e24c38-b729-cf991bd68913 | 0.187241001974 | content | ina |
| 0 | 27\_model\_1 | 素居 | 32 | b203c0b6-d48f-42ee-81a9-10567a155b6 | 0.186644658318 | ontent |  |
| 四 | 27\_model\_1 | 素居 | 32 | af65a9e2-e362-4035-86a2-8280a80ce70e | 0.183715093572 | conteat | a |
| □ | 27\_mode[\_1 | 素居 | 32 | 4860265e-0787-4340-9c91-7e0c309aa26 | 0.176671327236 | Cfn | in |
| trainjinu | +批量导入训练集-从训练基中删障 | | | | 你想知通的Top20 | | |

**图8-30** **分类结果**



是100定制并使用层压板制成不同色调的灰色和黑色以及面约的实木家具造就了充满凝聚力的开放式起居和用餐空间由设计的经非典单 扶 58非常适 合这种简约又不失优雅的装饰速人的黄色和绿色细节增强色彩对比并增添了个性化118283副本20145141547近代软装饰艺术发源于现代欧洲又被称为装 派艺术也称现代艺术它兴起于20世纪20年代随着历史的发展和社合的不断进步在新技术方光来艾的背景下华会的审美意识普遗觉壁人们的团些归情结也 在日益强化经过10年左有的发展形成了声热情古的软装饰艺术图赛是几何型或电县象形式演化而成所用材料丰高县以普重的居多除无然原料如王银象到

和 

美习 量第安人埃及人和早期古典主文艺术的借整软装饰艺术在第二次世界大战后的数年黑已不再流行但从60年代后期开始互工新引起了人们的注意并

高 慧 的 报 开 第四 四 方 面 老 房 新 装 可 通 过 饰 的 ：

己 圆 是 营 少面对 

不 为 装 

导师的2对1教学方式力求培养出具备正式项目的操作能力具体方菜的设计能力以及共际产品的接放能力于 一 身的专业轼装配饰改计师我们这个专业的

告训团队会让你们每二个人具正的型资到学习到了东西不甚此行软装饰在样板属中的应用整体考总1结空间设定主想设计师自先置身于空间的思考这个 给上即蒋上演的故!事并为这些动人的故事设计场景也就是为宫间设定 一 个主题使其成为有思想的房子赋子她生命样板房不仅是简单的摆设展示而需具

文果图气氧整体考应3软装饰设计整体考成家具灯饰面市挂画地税床品花艺饰品贯穿整个样板房设计的每 一 个细节通过细节的处理充分够显出主人的尊

典雅与 家  一 个亮点由于样板房的视觉焦一贯i

甲 计 手法 小 中 见 大 精 巧 的立优品味和生 估有 程 次 色 和 和 配 饰 的 配 合 使 有余主 出证人一走进这样的

detail 31篇贴子= 0位好友50元公积金居住，北京个人中心 | 微博 | 短消息

website http://bbs jiaju com thread-17780260-1 html

me 红 太 阳 8 8

Eaintime 2015-06-0611:35.52

洲 了 方 压

 复兴到现阶段软装饰已经到了比较成熟的程度我国在软装饰方面于二十 一 世纪伊始慢慢的受到人们的关注和研究现在正以快捷的步伐飞速发展着

的 下 第三：



主冗

图8-31 帖子信息

**sina 发帖排行榜**



C 回



图8-32 发帖排行



**本** **章** **小** **结**

本章从举例说明到公式定义，详细阐述了朴素贝叶斯分类算法的概念，并将这一算法应 用到本系统中，完成对爬虫获取到的数据进行分类的功能。首次将爬虫转型为 Web 爬虫， 即从界面控制爬取工作的开始与结束，并使用爬虫规则这一概念将传统的面向专一网站的 爬虫通用化，又不以主题准确率为代价，通过增加爬虫规则来改变爬虫的爬取目标，减少了 重复代码的编写，节约了人力成本，提高了代码的利用率。分类部分的核心算法是由Ma- hout 算法包内的朴素贝叶斯算法实现的，它是一种已经工业化的、成熟的、效率高的分类算 法之一。至此，本系统将爬虫与分类器有机地结合在一起，实现了数据从获取到分析最后展 示结果的一条龙化，使用普通户操作起来方便，结果直观。

**参** **考** **文** **献**

[1] 张引，陈敏，廖小飞.大数据应用的现状与展望.计算机研究与发展2013(S2):33-216.

[2] 严霄凤，张德馨.大数据研究.计算机技术与发展.2013(04):72-168.

[3] 刘光金.大数据处理对电子商务的影响分析.计算机光盘软件与应用.2014(17):6-25.

[4] 徐宗本，张维，刘雷，等.“数据科学与大数据的科学原理及发展前景”——香山科学会 议第462次学术讨论会专家发言摘登香山科学会议第次学术讨论会专家发言摘登.科 技促进发展.2014(1).

[5] 王文.浅析机器学习的研究与应用.计算机与信息技术.2010(Z2):7-9.

[6] Bolshoy A,Volkovich Z,Kirzhner V,Barzily Z.Mathematical Models for the Analy- sis of Natural-Language Documents.Studies in Computational Intelligence.2010.

[7] Maron ME.Automatic indexing:an experimental inquiry.Journal of the Association for Computing Machinery.1961(3):17-404.

\_8] 王香港.中文文本自动分类算法研究[硕士论文].上海交通大学，2008. [9] 于成龙，于洪波.网络爬虫技术研究.东莞理工学院学报.2011(03):25-9.

[10] 陈永江，仲兆满，陈宗华.HTMLUNIT 在网络信息采集系统中的应用.淮海工学院

学报(自然科学版).2013(04):5-31.

[11] Sean Owen RA,Ted Dunning,Ellen Friedman mahout in action:Manning Publica-

tions:2010-11-28.

[12] Lewis DD.Naive(Bayes)at forty:The independence assumption in information re-

trieval.Lecture Notes in Computer Science.1998:4-15.

13] Aas K,Eikvil L.Text Categorisation:A Survey.Raport Nr.1999.

[14] Allen J.Natural language understanding.Benjamin/cummings.1988,30(12):400-3397.

[15] 黄文博，燕杨.C/S 结构与 B/S 结构的分析与比较.长春师范学院学报：自然科学 版.2006,8-56.

**第** **9** **章** **大数据技术在社交** **网络研究中的应用**

社交网络是近年来学术界和工业界关注的焦点，将大数据思维应用到社交网络研究中， 揭示了以前不曾发现的新规律、新现象。例如许云峰等通过对大规模社交网络的全局分析， 重新对社区进行了定义，并结合膨胀度和骨干度提出了解决社区发现问题的新模型：社区森 林模型(Xu,Xu et al.2015)。除此之外，近年来很多学者(Cui,Wang et al.2014,Cui, Wang et al.2014,Li,Wang et al.2014,Eustace,Wang et al.2015,Ding,Zhang et al. 2016)对社交网络中的社区定义产生一致的共识，我们认为这些共识都是在社交网络的数据 日益增大，大家在观察这些海量数据时产生的一致认识。

社区发现是社交网络中的一个关键问题，用来发现社交网络的结构和功能。当前学术 界已经涌现了大量关于重叠社区发现和非重叠社区发现的研究工作，同时这些工作采用了 很多技术，如谱聚类、模块化最大化、随机分布、微分方程，以及统计力学等，但是这些工作绝 大多数采用纯数学和物理的方法来从社交网络中发现社区，反而忽略了社区和社交网络的 社会学和生物学特性。这个缺点是时代局限性造成的，因为在此之前的数年间，社交网络的 数据规模较小，数据的可视化条件有限，因而以前的研究者很难对数据有全局化的观察视 角。近年来随着社交网络的蓬勃发展，产生了海量的社交网络数据，并且数据可视化技术也 产生了蓬勃的发展。因而近年来的研究者将众多强有力的大数据技术及可视化应用到对海 量社交网络数据的探索中，从而对社交网络的结构产生了新的认识。这些大数据技术包括 奇异向量分解(SVD)、子空间分解(Subspace Decomposition)等。

在本章中，我们将大数据思维应用到社交网络的非重叠社区发现研究中。我们提出了 基于这些社会学和生物学特性的社区森林模型来描述真实世界的大型网络结构，其次我们 定义了一个叫作骨干度的新度量，用来衡量边的强度和顶点的相似性，并且基于膨胀度给社 区一个新的公式化定义，最后提出了一个新的算法：基于骨干度和膨胀度从真实的社交网络 中发现非重叠的社区。该算法和CNM 算法、GN 算法相比，在成本和清晰度上有着更好的 性能和效果，并且实验表明该算法在 Email-Enron、美国大学足球队、跆拳道俱乐部等数据 集里运行良好。



**9.1** **社区发现研究简介**

随着社交网络的大规模出现，社区发现在大型社交网络中变得越来越重要，因为它可以 帮助发现大型社交网络中的隐藏知识。隐藏的知识包括结构、功能、规律等。在大型网络系

第9章 大数据技术在社交网络研究中的应用

统中，网络的结构是图挖掘的关键。代谢网络中的社区可能对应于功能单元、周期，或者执 行某些任务的回环(Newman,2010) 。 社区发现通常被用作发现和理解网络中大型结构的 工具(Easley and Kleinberg,2010)。在过去的十年里，社区发现(有时也叫作图的划分)已 经被应用于现实世界的很多领域，例如生物网络、网络图、超大规模集成电路设计、社交网络 和任务调度。

很多把网络划分为社区的社区发现算法已经被提出来，其中有一些通常是选择描述社 区特征的目标函数，然后优化它们，但是这些目标函数精确优化的难度通常是NP 级别的

((Arora,Rao et al.2004);(Schaeffer,2007))。另外 一 些算法采用探索((Girvan and

Newman 2002);(Karypis and Kumar 2006))或进制值((Leighton and Rao 1999);(New- man,2013);(Spielman and Teng 1996))算法，近似地优化一些目标函数，这些目标函数作 为现实世界中社区的解释。但是这些算法的着重点是如何通过拉普拉斯矩阵和特征值分解 把网络划分为社区，这些算法十分简练，但是却不能高效、准确地处理大型网络，他们大多采 用了纯数学和物理的方法，例如谱聚类、模块化最大化、随机游动、微分方程和统计力学等， 反而忽略了社会和社区以及网络的生物学特性。社会和社区以及网络的生物学特性指的是 研究具体网络的围观特点所得到的特征。例如，社交网络有许多社会属性：强连接和弱连 接、桥、捷径、邻里互惠度(Easley and Kleinberg 2010)、权 重 、hub weight(Kleinberg 1999),K-component 等。如果我们把一个社交网络视为一座森林，那么社区在这个森林中 就作为树、灌木、草等存在，因此，社交网络和森林有许多功能是相似的。例如，社交网络是 增长的，社区也是增长的，这一特点和森林一样。我们把这些特点看作社交网络的生物学 特性。

如果我们深入挖掘网络的社会学和生物学特性，那么我们是否能得到一个更简单的模 型来描述社区和网络，然后基于此模型得到一个有效的算法? Newman 讨论了现实世界中 大型网络的结构通过看上去的核心的大小，认为大多数网络的结构是一个填满网络绝大部 分的大核心，有时候是填满整个网络，有时候也有些小核心没有链接到网络主体(Newman, 2009)。我们可以认为现实世界中的大型网络是由一些 K-components 组成的，K 的值越 高 ，K-components 就有更高的连通性，更像一个社区。 一个 K-components 和一个 K-core 一样，这就像一个微小的社区，但是 K-components 不是一个完整的社区，因为所有顶点度 小于 K 的顶点都已经被移除，然后 K-components 可以描述真实世界的大型网络的部分网 络结构，并非全部。因此，我们需要一个新的模型来表征现实世界里的大型网络。我们考虑 了社交网络的很多特征，然后提出了社区森林模型和一种基于社区森林模型的高效算法。

我们的工作分为三个部分。首先，我们提出了基于社会学和生物学特性的社区森林模 型，来表征现实世界大型网络的结构。其次，我们定义了一个叫作骨干度的新度量，用来衡 量边的强度和顶点的相似性，并且基于膨胀度给社区定义了一个新的定义。最后，我们提出 了一个基于骨干度和膨胀度的新算法，用来在现实世界中的社交网络里进行社区发现。



**9.2** **社区发现相关研究工作**

在这一节中，我们先调查相关工作，然后给出一个本章重点研究背景和动机。

**9.2.1** **相关工作**

在大规模网络中，有很多社区发现算法，这些算法主要可以分为两个大类：重叠社区发 现和非重叠社区发现。重叠社区发现算法被审查和分类成五个：社团渗透、线图和链接划 分、本地膨胀度和优化、模糊发现、基于代理和动态的算法(Xie,Kelley et al.2011),这些算 法是基于一致性研究，因为在一个社交网络中，人们多自然的属于多个社区。关于重叠社区 发现相关算法的细节，可以看 Xie 等人的研究(2013)。非重叠社区发现算法被审查和分类 之后分为五个研究方向(Leskovec,Lang et al.2010),他们用了很多技术，例如光谱聚类、 模块化最大化、随机散布、微分方程、统计力学等，来确定一组节点作为一个社区，这组节点 的成员之间有比该网络中其他节点之间更多或更好的链接(Leskovec,Lang et al.2010)。 对于非重叠社区发现算法的细节，可以参看(Leskovec,Lang et al.2010)。

以上算法中有 一 部分给了我们研究工作 一 些启发。与我们的工作相关的是 Newman and Clauset(2004),Newman,and Moore(2004),Kleinberg(1999),Leskovec et al. (2010)。

Newman 等人分析了一种聚类分层算法，通过社区的模块度来描述一个社区的概念。 Newman 等人根据社区的模块度，分析出来了一种分层聚类算法，用来描述社区的概念。 Kleinberg 提出了一个权威度方面的算法构想，基于一组相关权威页面之间的关系和把这些 页面通过一定的链接结构链接到一起的路由页面(Kleinberg 1999)。我们把权威度和 hub 度这两个概念整合为一个新的概念，命名为无向网络的权重。 Leskovec 等人2010年定义 了网络社区简介(NCP), 通过网络社区的大小和功能等特点来刻画网络社区的质量。灵感 来源自于网络社区简介(NCP), 我们使用散点图来描述整个社会发现算法的质量，散点图 的横轴是社区的大小，纵轴是传导性和膨胀度。

Kannan 等人指出了膨胀度的概念。膨胀度是社区中所有指向外面的边，比上社区内 部的边的总数社区最小的比。在研究中，我们发现膨胀度应该从社区中心到社区边缘逐渐 变小。我们用这个特征和骨干度来从社区中心逐渐增加新的节点到社区里，直到社区的膨 胀度开始变大，这个过程可以把社区划分为社交网络。

Kannan 等人提出了一个名为传导性的度量，用来提供集群个体的质量的度量。(Kan- nan,Vempala et al.2000)传导性是 一个类似膨胀度的属性，Leskovec 等人考虑到传导性 是一个很好的度量，用来表征网络社区的质量(Leskovec,Lang et al.2010)。在研究中，我 们用传导性和膨胀度作为度量社区发现算法的主要评价指标。

Easley 和 Kleinberg 定义了闭包、结构洞、弱连接和强连接、桥、捷径、邻里互惠度等。 邻里互惠度表征了边缘强度(Easley and Kleinberg 2010)。在这些概念的基础上，我们提出 了骨干度的概念，来表征节点和社区之间的连接强度。

Palla 等人(2005)提出了社团渗透的思想，他们的社区定义为一个典型的社区由几个完 整的(全连接)的子图组成的，这些子图趋向于分享更多的节点，更准确地说， 一个K 分支社 区是 一个全部 K 分支(完全子图的大小为 K) 的集合，可以从彼此相邻的 K 分支相互连通 (邻接的定义为分享K 分支的 一个节点)。这种方法是基于第 一 定位网络的所有派系最大 完全子图，并通过不断点击矩阵进行对一个标准成分的分析来确定社区的展开。在本章中， 我们提出了邻里互惠度的概念来测量两节点之间的连接强度，这一观念和三分支中两节点

间的连接度是一致的。

我们研究了以上的重叠社区发现算法和非重叠社区发现算法。虽然重叠社区发现领域 是目前研究的重点领域，但我们仍然认为非重叠社区发现算法领域有很大的提高空间和广 阔的应用前景。我们目前的工作主要集中在无向网络中的非重叠社区发现算法，我们将探 讨相邻非重叠社区发现的相关问题。

**9.2.2** **研究动机**

在网络中，社区概念的界定决定如何发现社区，那么社区的概念该如何界定?从直觉来 讲，社区是一组节点的集合，在这个集合中的顶点之间的联系和与集合之外的顶点间连接相 比要远远密集得多(Leskovec,Lang et al.2010);(Radicchi,Castellano et al.2003)。 Radicchi 提出社区概念：在一个强社区中，节点和社区内的子图有更多的连接。在弱社区 中，所有度的总和在社区范围内的总和大于网络其他部分的度的总和(Radicchi,Castellano

et al.2003)。1969年 Luccio 和 Sami 提出社区成为最小群体的概念，1973年 Lawler 将 其 改名为 LS(Radicchi,Castellano et al.2003) 。LS 定向像一个强社区，另一种定义叫做 K 核心，图G 的 一 个 K 核 心 是 图 G 中所有顶点的度都至少为 K 的最大连通子图(Seidman 1983)。K 核心像一个弱社区。通过上面的讨论，我们发现那些社区的概念是简洁明了的， 但并没有详细地描述内部结构，不能可视化。那么我们是否可以有一个概念能让我们想象 社区的具体形象?意味着这个概念可以给出明确的边界和内部结构。

随着社会网络研究的发展，近年来人们提出了许多关于网络社区结构的新概念，例如弱 连接和强连接、桥、捷径、邻里互惠度等。如果我们分析网络中节点之间的联系，我们会发现 这些链接包括：弱连接和强连接、桥、捷径等。如果联系如同树枝和树的联系，那么这样一个 群落可以被看作是一棵树， 一个社交网络可以看作一个社区森林，这个网络所包含的强社区 可以看作是一个社区的森林，而其他弱社区则被看作是灌木丛。根据这一假设，我们可以给 出社区赋于许多生物学的特性，并重新定义社区观念。整合社交网络的概念，本文提出了一 个新的指标——骨干度，并且重新定义了社区的概念，提出了社区森林这一社区概念的可视 化模型。



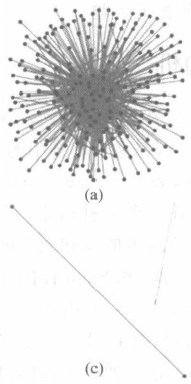
**9.3** **模型与问题的形式化**

在这一节中，我们提出了社区森林模型，确定了社区发现的难题，并介绍了几个相关概

念和必要的符号。

**9.3.1** **社区森林模型**

社交网络和森林在形态上具有相似的特征和结构。图9- 1是一个拥有超过30000用户 的社交网络的可视化图，它就像一个茂密的森林。在巨型组件周围，有一些边缘社区。社交 网络中的社区通常由核心顶点、核心骨干和边缘顶点组成，它们的形态及结构和森林里的树 木、灌木和草相似。图9-2是六个小组件的可视化图，这些图有的像稀疏的灌木和草，有的

像稠密的树木。如果我们把社交网络看作一个森林，森林群落中有树、灌木、草。

社交网络中的社区之间，有的有联系，有的没有联系，这些特点就像森林中的树、灌木和 草一样。社交网络中的大型社区可以派生出新的小社区，此功能就像是森林中的树木、灌木 和草一样。在社交网络和森林之间，还有很多这样的相似之处。社区被定义为一个顶点的

密度比网络的其他部分更密集的 一 个子集 (Radicchi,Castellano et al.2003)。 为 什 么 我们把社区比作一棵树?一棵树由根、树干、 树叶组成， 一个社区由顶点和边组成。有些 联系比较强，有些联系比较弱； 一些顶点是核 心顶点，而另外一些是边缘顶点，像树叶。我 们把强链接看作树干， 一些核心顶点看作树 根， 一些边缘顶点就是树叶。

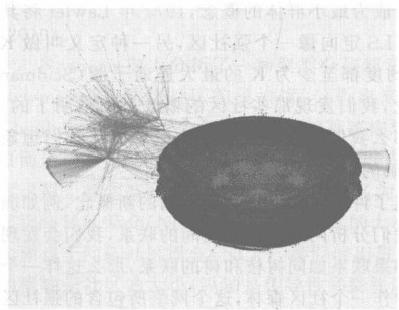
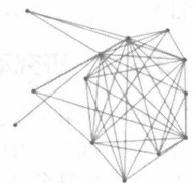
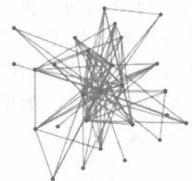


图9-1 一个包括超过30000用户的完整的 在线社交网络的可视化

(b)

(e)

(f)

图9-2 图9-1中社交网络可视化图的六个子图

网络中社区发现的难题是把一个给定网络中的顶点划分为不重叠的组，组内的连接都 是比较密集的，而群体之间则十分稀疏(Newman,2013) 。 在网络中发现社区，就像在森林 中发现树一样。但是怎样在森林中发现一棵树呢?我们只有边和顶点，需要一个度量来测 量边的强度，我们称这个度量为骨干度。让社区作为树，然后边是树木的枝干。 一条边由两 个顶点和一个关系组成，所以骨干度必须衡量这三个因素。边更像竹子的竹节，每一段竹节 由两个关节和一段中空组成，关节之间的联系是这个中空，邻里互惠度可以用来表示这段中 空的强度，网络中的权重可以用来表示连接的强度。如果这个度量工作已经做好了，那么从 一个网络中进行社区发现会有这样的过程：首先寻找骨干度最大的边，其次在骨干度的基础 上寻找最近的顶点直到这个社区的边缘，重复上面的步骤划分其他顶点，直到所有顶点都被 划分。控制骨干的选择，可以使算法在操作上更具有可扩展性。例如，如果该算法不允许骨 干两边的顶点再次被划分，那社区发现是没有重叠的，否则就是可重叠的。在本章中，我们 主要讨论非重叠社区发现。

9.3.2 问题形式化

给定一个有|V| 个顶点和|E| 条边的无向图G(V,E)。 让 n=|V|,m=|E|。 让 C 表

示社区中的一组顶点，C, 是 C 的顶点数，C,=|C|。 让 E={(u,v)∈E:u∈C,v∈C},Cm

是C 中边的数目，Cm=|E.| 。 让C={(u,v)∈E:u∈C,v4C},|C| 是 C 边缘的边的数

量。让d。表示顶点u 的度，让NB。表示顶点u 相邻的一组顶点的集合。让NBc 表示社区 C 的邻域点集。

NBc={v:(u,v)∈E,u∈C,v≠C}

定义1(网络权重)把顶点v 的标识记为i,图 G 中任意顶点的网络权重可以记为x;。

我们能用 NW。来表示v 的网络权重。 

网络权重根据 HIT 算法的定义(Kleinberg 1999),但是 HITS 算法中，网络权重需要大 量的计算来平衡，为了节省计算量和计算时间，顶点的相对权重可以被认为是 ,此外



定义2(社区膨胀度) 这个指标度量边缘点以外的社区数(Kannan,Vempala et al.

2000)。

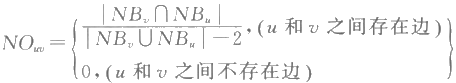


定义3(社区膨胀度差异) 在加入一个新的顶点i 之后，社区 C 的膨胀度的改变。 DE(i)=EXcnuy-EXc

定义4(顶点 i属于社区C 的概率)。



定义5(邻里互惠度) 给定两个顶点u 和v,让 NB, 表示和顶点u 相邻的顶点的集合， 让NB。表示和顶点v 相邻的顶点的集合，让NO 表示顶点u 和顶点v 的邻里互惠度。



定义6(骨干) 一个骨干由一条边和两个连接到该边的顶点组成，如果一个骨干两端 的顶点一个在当前社区内部，另一个在当前社区外部，那么我们把当前社区内部的这个顶点 命名为内部顶点，当前社区外面的这个顶点命名为外部顶点。

定义7(骨干度) 顶点u 和顶点v 之间的这条边的骨干度为

D…=(NW+NW)×NO+6

D 可以测量边的强度和节点相似度。当顶点 u 和顶点v 不相邻的时候NO=0, D =δ,δ 是平滑常数的参数，基于经验我们让δ=0.01。

定义8(社区C 的最大骨干度) 让 CDm 表示社区 C 的最大骨干度，CD 所表示的骨 干即为社区C 的核心骨干。

CDmax =max{Dm,u∈C,v∈C}

定义 9(社区新定义) 为社区做出了一个新的意义上的定义。社区是一组从核心骨干 D…逐渐向外膨胀的顶点们，并且随着膨胀，膨胀度不断降低，直到 EX 变为最小值。

C={u:u∈C,v∈C,(u,v)∈E,EX(cuv)>EXc}

定义 10(社区森林) 图 G 中的社区们，被定义为社区森林。

CF={C:C∈V,v≠C,EX(cuv)>EXc}

定 义 11(三元闭包) 如果两个人在社交网络上有一个共同的朋友，那么他们在未来的 某个时候成为朋友的可能性将会增加(Rapoport 1953)。

定义12(共同好友数) 在一个社交网络中，如果两个人之间有一个共同好友，那么这 两个人变成好友的可能性会增加，因此采取共同好友数可以测量两个顶点之间的相似度。

定 义 13(会员闭包) 在一个社交网络中，加入一个人的朋友参加了一个社区，那么这 个人参加同一社区的可能性会提高(Rapoport 1953)。

定义14 (顶点i 到社区C 的会员闭包数) 假如顶点i 在社区C 中 有x 个好友，那么 顶点i 到社区C 的会员闭包数等于x 。x=|NB)∩C|, 顶点i 的会员闭包数可以用来测 量其他社区和社区C 的相似度。

定义15(社区C 边缘的定义) 让 Cw 表示社区C 的边缘顶点的集合， |Cw| 表示社区 C 边缘顶点的数目。

BVc={v:(u,v)∈E,u∈C,v C,EX(cuv)>EXc}

定义 16(从顶点v 到社区C 的骨干度总和) 这个指标可以测量顶点 v 到社区C 的

距离。



定义17(到社区C 的最近顶点) 让 MAX 作为{NC 。:v ∈{NBc-BVc},EX(cuw< EXc} 的最大值，加入NC 。=MAXw, 那么顶点v 是到社区C 的最近顶点。

v={v:(u,v)∈E,u∈C,v4C,v∈{NBc-BVc},EX(cuo)<EXe,NC 。=MAX}

定义 18(非重叠社区发现中顶点i 和j 之间的关系预测) NO。 是顶点对(i,j)之间的 骨干度，θ。是一个向量，表征顶点u 周围的会员关系(Airoldi,Blei et al.2008),0。是一个向 量，表征顶点 v 周围的会员关系，K 是图G 中所有社区的数量。



NOm一定要松散：当 u 和 v 之间没有边存在的时候， NO 。=0, 否 则 让 NO 。=





9.4 骨干度算法

根据社区森林模型，社区发现的过程可以这样定义：寻找每个社区的核心骨干并且寻找 每个社区的边界。如果找到了每个社区的核心骨干，那么便能由此确定这个社交网络中的 社区数量。确定社区数和每个社区的核心骨干之后，算法可以扩展到大规模的社交网络中 进行社区发现。

第9章 大数据技术在社交网络研究中的应用

为什么要从核心骨干开始，而不是从核心顶点开始?因为我们把社交网络中的社区看 作森林里的树，而树木都是从树干开始扩张的，所以社区也必须从核心骨干开始扩张。名为 CDmx 的这个骨干是社区 C 的核心骨干。当社区C 只包涵一个核心骨干边的时候，



然后d ₄+d₀=2(EXc+ 1)并且

D =(NWm+NWo)×NO+δ





在这里m 和δ是常量，EXc 和 NO…是变量，m=|E|, 基于经验我们让δ=0.01来作为 平滑参数。整合 D…、EXc 和NOm, 以便于更准确地选择社区的核心，同时还能避免选中节 点之间产生的结构洞。 NO。这个集合越大，社区内部的连接越密集，这一点恰如 Radicchi 等人在社区发现框架中所提到的(2004)。如果我们通过核心顶点来进行社区发现，那么这 唯一能度量决定核心顶点的是顶点的权重。在无向网络中，核心顶点的权重和核心顶点的 度有着密切联系，但是在社交网络中， 一个有着很大度数的顶点也有可能是个结构洞，如果 让有最大骨干度的骨干作为社区的核心，将会避免选中结构洞的问题，因为骨干度包括两个 顶点的权重和邻里互惠度，所以加入一个有很大骨干度的骨干，这个骨干很可能是一个社区 的核心或者靠近一个社区的核心。

我们的算法首先计算社交网络中每一个骨干的骨干度，并且以降序的方式把这些骨干 度保存在骨干度列表里。让初始社区是空的，然后选择骨干度列表中骨干度最大的骨干作 为当前社区的初始骨干，接着选择这个集合中剩余骨干中骨干度最大的骨干轮流连接到当 前社区。如果加入一个骨干到当前社区之后膨胀度变小了，那么不断地把骨干度最大的骨 干增加到连接到当前社区的骨干集合中，其他外部顶点添加到当前社区边界的骨干集合中， 继续寻找同当前社区有连接的顶点中有着最大骨干度的顶点，直到没有符合标准的顶点在 当前社区的邻居集中，此刻一个新的社区就完全划分出来了。按照上述方法继续迭代，把剩

下的顶点划分到新的社区中，直到骨干度列表中不再有骨干度大于骨干度阈值 f 的骨干存

在，或者其余顶点的数目小于参数w 。这 里w 的值是根据|V| 的值得到的，例如

因为在大型社交网络中，当剩余顶点在社交网络中极少的时候，这些剩余顶点中真正有价值 的社区不多，如果再使用上述步骤，就会发现很多很小的和无用的社区。在这个时候，这些 剩余顶点可以用一些简单算法被收集起来，例如使用会员闭包来确定一个顶点是属于哪个 社区。骨干度阈值 f 可以通过经验或者需求或者使用者来设定，例如使用者想划分核心骨 干的骨干度在0.3以上的社区，这个时候便让 f=0.3。

9.4.1 骨干度算法框架

给定一个无向图 G(V,E), 该图有 |V| 个顶点和 |E| 条边，给定一个节点列表 NL 来保 存集合V 中的顶点，把当前社区记为C,,C, 邻接的社区的集合是NBc, 把 C; 边界的集合记



大数据技术及行业应用



为BVc, 给定骨干列表 BL来保存集合E 中的骨干。骨干度算法的实现如下：

|  |
| --- |
| Algorithm 1.Backbone degree algorithm implementation |
| Data:An undirected G(V,E).Result:The community set CF in G. |
| begin |
| 1 NL=V,BL=edges with backbone degree≥fin E,CF=null,i=0.SORT BL according to de- |
| scending order,indexn=0. |
| 2 Get a backbone b from BL according to index indexm++,get vertices u and v from b,note |
| the backbone degree of b as BD,, |
| noet the size of NL as nl.  3 while BD≥fand nl≥w do |
| 4 if u∈NL and v∈NL then |
| 5 C;=(u,v} |
| 6 Ec\_PRE=the Expansion degree of C;. |
| 7 calculate the NBc,of C,BVc,=null |
| 8 if{NBc-BVc}=Null,then |
| 9 add C;to CF;i++;goto step2. |
| 10 else |
| 11 find the nearest vertex nv from {NBu -BVc,) |
| based on  backbone degree,add vertice nv to C;,calculate the  Expansion  degree of C;and note it as Ec-cur.  12 if(Ec-cur-Ec-PRE<0),then |
| 13 remove vertice nv from NL and add vertice nv to C,  goto step11. |
| 14 else delete vertice nv from C,and vertice nv to BVc, |
| 15 if(NBc-BVc}=Null,then |
| 16 add C;to CF,i++,goto step2. |
| 17 else |
| 18 goto step11 |
| 19 end if |
| 20 end if |
| 21 end if |
| 22 else |
| 23 goto step2 |
| 24 end of |
| 25 end while |
| 26 Collect all vertices that divided into no community or several communities, |
| 27 return CF.  end |

**9.4.2** **算法的时间复杂度**

我们的算法使用合并排序的骨干列表，其运行的时间复杂度为 O(mlogm), 而社区发 现过程中时间复杂度为 O(n+m), 所以我们的算法在有 n 个 顶 点m 条边的社交网络中进 行社区发现的时间复杂度为O(mlogm+n+m) 。 因为不是所有的骨干都是核心骨干，所以 如果我们根据一个阀值f 来过滤骨干列表，那么骨干列表中的骨干数目将会急剧地下降， O(n+m) 也会随之急剧下降，所以我们的算法运行的时间复杂度约为 O(n+m)。 我们分 析了五大数据集的骨干度。表9-2是具有最大骨干度的边。表9-3是f≥0.2 或 f≥0.3 的 边。我们发现的最大骨干度是1 .282727,最小骨干度是0 .01,当f 的值变化的时候，骨干 列表也会急剧变化，这些在表9-3中表现了出来。

**9.4.3** **算法比较**

CNM 算法在有 n 个 顶 点 m 条 边 的 社 交 网 络 中 进 行 社 区 发 现 的 时 间 复 杂 度 为

O(mdlogn), 其 中d 是 图 的 深 度 。Girvan Newman 的算法运行的时间复杂度为 O(n3)。

骨干度算法运行的时间复杂度大约为 O(n+m), 但 是 Girvan Newman算 法 和 CNM 算 法 、

骨干度算法在优化模块的试图不同，所以很难说哪个算法时间复杂度方面更优越。

骨干度算法的目标是在骨干度和社区森林模型的基础上，在社交网络里进行社区发现， 具有可扩展性，可应用于大型社交网络。骨干度算法可以通过骨干度阈值 f 和 参 数w 的 改 变从不同的深度进行社交网络中的社区发现，这使得它非常灵活，同时它能预测社交网络中 任意两个顶点i 和 j 之间的关系，以便于进行重叠和非重叠的社区发现。



**9.5** **实** **验** **分** **析**

在这一节中，主要研究骨干度算法的有效性和准确性，主要与 CNM 算 法(Clauset,

Newman et al.2005)进行比较，此外还同 Martin Rosvall 的 算 法(Rosvall and Bergstrom 2007)和 GN 算 法 进 行 简 单 的 比 较 。CNM 算 法 的 实 现 来 源 于 斯 坦 福 网 络 分 析 平 台 (SNAP) 。SNAP 是一个通用的网络分析与图挖掘库。它是用C++ 写的，容易扩展到具有 数以百万计节点和数十亿边的大规模网络(Leskovec 2014)。

**9.5.1** **数据集**

我们使用了一个人造网络和一些标准数据集：美国大学足球俱乐部数据集、安然公司电 子邮件的数据集、DBLP 合作网络数据集。美国大学足球俱乐部和空手道俱乐部是标准的 数据集，以证明社区发现算法的有效性，DBLP 合作网络数据集是地面实况网络，这些数据 集的详细描述如表9-1～表9-3所示。

人造网络是由星状结构、网状结构、线形结构组成，并且人造网络边的骨干度为0.01 如图9-3 所示，该人造网络数据集的描述如表9-4所示。

大数据技术及行业应用



**表9-1** **数据集描述**

|  |  |  |  |
| --- | --- | --- | --- |
| Date set | Vertices | Edges | Known communities |
| An artificial network | 19 | 21 | 3 |
| Zachary's Karate Club | 34 | 78 | 2 |
| American College Football | 115 | 613 | 12 |
| Enron email communication network | 36692 | 183831 | Unknown |
| DBLP computer science bibliography | 317080 | 1049866 | 13477 |

**表9-2** **最大骨干度的边列表**

|  |  |  |
| --- | --- | --- |
| Date set | Biggest backbone edge | Backbone degree |
| An artificial network | Anyone | 0.0 |
| American College Football | 763689 | 1.282727 |
| Zachary's Karate Club | 3331 | 1.01346 |
| Enron email communication metwork | 76136 | 0.392861 |
| DBLP computer science bibliography | 55885.286328 | 0.660222 |

**表9-3** **骨干度≥0.2和≥0.3边的数量**

|  |  |  |
| --- | --- | --- |
| Data set | f≥0.2 | f≥0.3 |
| An artificial network | None | None |
| American College footbal | 449 | 411 |
| Zachary's Karate Club | 31 | 9 |
| Enron email communicaton network | 106592 | 12 |
| DBLP computer science bibliography | 625721 | 12536 |

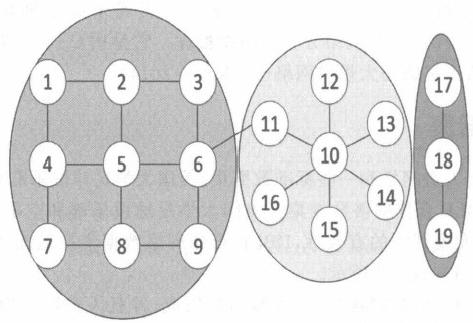


图9-3 一个由星状结构、网状结构和线性结构组成的人造网络

<< 184

第9章 大数据技术在社交网络研究中的应用



**表9-4** **人造网络中的骨干度列表**

|  |  |  |
| --- | --- | --- |
| Edge name | Backbone degree | Belonged to structure |
| 1,2 | 0.01 | Mesh |
| 1,4 | 0.01 | Mesh |
| 2,3 | 0.01 | Mesh |
| 2,5 | 0.01 | Mesh |
| 3,6 | 0.01 | Mesh |
| 4,5 | 0.01 | Mesh |
| 4,7 | 0.01 | Mesh |
| 5,6 | 0.01 | Mesh |
| 5,8 | 0.01 | Mesh |
| 6,9 | 0.01 | Mesh |
| 6,11 | 0.01 | Mesh |
| 7,8 | 0.01 | Mesh |
| 8,9 | 0.01 | Mesh |
| 10,11 | 0.01 | Star |
| 10.13 | 0.01 | Star |
| 10,13 | 0.01 | Star |
| 10,14 | 0.01 | Star |
| 10,15 | 0.01 | Star |
| 10,16 | 0.01 | Star |
| 17,18 | 0.01 | Line |
| 18,19 | 0.01 | Line |

Zachary 的“空手道俱乐部”数据集是1970年关于美国一所大学34名成员之间友谊的 社交网络。Wayne Zachary 观察到在美国一所大学的空手道俱乐部成员之间的社会交往。 他在1970年早期建立了包含34个顶点和78条边的社交网络。由于偶然的机会，俱乐部创 始人和空手道教练之间产生了分歧，把该社区分裂成了两个小社区，并分别出任小社区的领 导人员。

美国大学橄榄球队是美国足球运动会在2000 年常规赛期间对 IA 院校划分时候的数 据集。

安然公司的电子邮件数据集包含大量的电子邮件 (Leskovec,Lang et al.2015)。这 些数据起初是公开到网络上的，在联邦能源管制委员会的调查报告中。网络的节点是电子 邮件地址，如果一个地址 i 至少向地址j 发送过至少一个电子邮件，那么这个图中就包含了 一条从i 到 j 的无向边。

185 >>

将骨干度算法应用于 Zachary 空手道俱乐部社交网络图，如图9-4所示，黄色的结果是 社区0,红色的则是社区1(对于本图标题中提及的颜色的解释，读者可以参考本文的网络 版，网址为 <http://www.sciencedirect.com/science/article/pii/S0957417415004443>)。

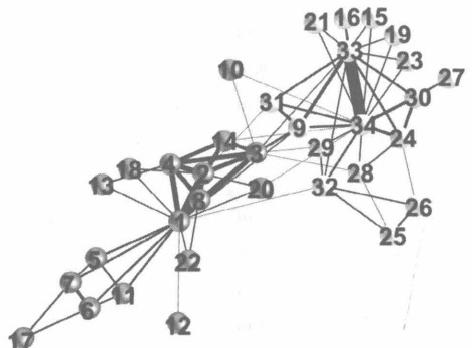


图9-4 将骨干度算法应用于 Zachary 空手道俱乐部社交网络

采 用CNM 算法对 Zachary 空手道俱乐部进行划分的结果如图9 - 5 所示。黄色的表 示社区0,蓝色的表示社区1,绿色的表示社区2(对于本图标题中提及的颜色的解释，读 者 可 以 参 考 本 文 的 网 络 版 ， 网 址 为 <http://www.sciencedirect.com/science/article/pii/>

S0957417415004443)。

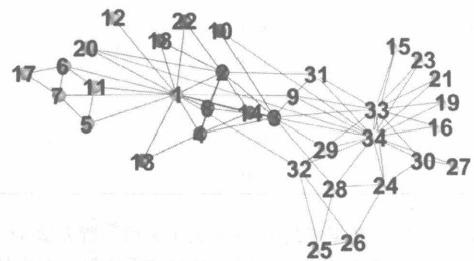


图9-5 采用 CNM 算法对 Zachary空手道俱乐部进行划分的结果

**9.5.2** **一个特定人际关系网络的测试**

将我们的算法应用到一个特定的人际网络中，包括星状图、网状图和线性图，如图9-3 所示。这是一个很简单的网络，当δ=0.01时，这个网络边缘的骨干度为0.01。表9-4是人 际关系网中的骨干度列表，表9-5是人际关系网络通过骨干度划分的结果。测试表明我们 的算法在骨干度非常小的一些稀疏网络中，仍然具有很好的适应性。

第9章 大数据技术在社交网络研究中的应用



**表9-5** **人际关系网络通过骨干度划分的结果**

|  |  |  |  |
| --- | --- | --- | --- |
| Vertex number | Belonged to structure | Neighbor vertices | Community ID |
| 1 | Mesh | 2,4 |  |
| 2 | Mesh | 1,3,5 | 2 |
| 3 | Mesh | 2,6 | 2 |
| 4 | Mesh | 1,5,7 | 2 |
| 5 | Mesh | 2,4,6,8 | 2 |
| 6 | Mesh | 3,5,9,11 | 2 |
| 7 | Mesh | 4,8 | 2 |
| 8 | Mesh | 5,7,9 | 2 |
| 9 | Mesh | 6,8 | 2 |
| 10 | Star | 11,12,13,14,15,16 | 1 |
| 11 | Star | 6,10 | 1 |
| 12 | Star | 10 | 1 |
| 13 | Star | 10 |  |
| 14 | Star | 10 |  |
| 15 | Star | 10 |  |
| 16 | Star | 10 |  |
| 17 | Line | 18 |  |
| 18 | Line | 17,19 | 0 |
| 19 | Line | 18 | 0 |

**9.5.3** **Zachary** **的空手道俱乐部测试**

将我们的算法应用于Zachary 空手道俱乐部的人际关系网络中，图9-4是应用我们的 算法即骨干度算法划分的结果。我们的算法将网络划分成了两个社区，表9-7 是 Zachary 空手道俱乐部应用骨干度算法之后的膨胀度和传导性(Girvan and Newman 2002)图9- 5 是 采用CNM 算法划分的结果，CNM 算法把该网络划分成了三个社区，在社区1和社区2之 间，没有明确的边界和内部结构。采用CNM 算法，该网络的膨胀度和传导性如表9-6(Ros- vall and Bergstrom 2007) 所示。采用骨干度算法，该网络的膨胀度和传导性如表9-7 所示。

表9-7 的结果明显优于表9-6。

**表9-6** **CNM 算法对Zachary 空手道俱乐部的划分结果**

|  |  |  |  |
| --- | --- | --- | --- |
| Community ID | Community size | Conductance | Expansion |
| 1 | 9 | 0.308952 | 1.777778 |
| 2 | 8 | 0.333333 | 1.5 |
| 0 | 17 | 0.128205 | 0.588235 |

**表9-7** **骨干度算法对** **Zachary** **空手道俱乐部的划分结果**

|  |  |  |  |
| --- | --- | --- | --- |
| Community ID | Community size | Conductance | Expansion |
| 0 | 17  17 | 0.128205  0.128205 | 0.588235  0.588235 |

大数据技术及行业应用



表9-8展示了骨干度算法对 Zachary 的空手道俱乐部数据集实施过程，这个过程不包 括步骤26。我们发现步骤26之前，社区0和社区1是重叠的在顶点31、10、9。如果我们运 行步骤26,结果如图9-4所示，与标准结果完全相同，我们还发现，随着顶点的扩展，社区逐 渐减少连接，曲线如图9-9所示，社区和顶点们的连接是基于骨干度。这一现象在本书得到 了充分的验证，跟踪顶点的顺序和它们的骨干度的记录如表9-8所示，表9-8中的 MAXNc 在定义17中有定义。

**表9-8** **骨干度算法对Zachary** **俱乐部数据集实施过程**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Vertex | ID | Current expansion | Community ID | Joining order | MAXnc |
| 34 |  | 13.5 | 0 | 1 | 1.013 |
| 33 |  | 13.5 | 0 | 1 | 1.013 |
| 9 |  | 9.333 | 0 | 2 | 0.414 |
| 31 |  | 6.5 | 0 | 3 | 0.544 |
| 30 |  | 5.2 | 0 | 4 | 0.408 |
| 24 |  | 4.166 | 0 | 5 | 0.624 |
| 32 |  | 3.857 | 0 | 6 | 0.233 |
| 27 |  | 3.125 | 0 | 7 | 0.207 |
| 29 |  | 2.666 | 0 | 8 | 0.177 |
| 28 |  | 2.4 | 0 | 9 | 0.177 |
| 19 |  | 2 | 0 | 10 | 0.165 |
| 23 |  | 1.666 | 0 | 11 | 0.165 |
| 21 |  | 1.384 | 0 | 12 | 0.165 |
| 15 |  | 1.143 | 0 | 13 | 0.165 |
| 16 |  | 0.933 | 0 | 14 | 0.165 |
| 25 |  | 0.8125 | 0 | 15 | 0.108 |
| 26 |  | 0.588 | 0 | 16 | 0.236 |
| 10 |  | 0.555 | 0 | 17 | 0.01 |
| 2 |  | 11.5 |  | 1 | 0.653 |
| 1 |  | 11.5 |  | 1 | 0.653 |
| 4 |  | 8.333 |  | 2 | 0.844 |
| 3 |  | 7.25 |  | 3 | 1.153 |
| 8 |  | 5 |  | 4 | 1.189 |
| 14 |  | 3.666 |  | 5 | 1.134 |
| 9 |  | 3.286 |  | 6 | 0.249 |
| 31 |  | 2.875 |  | 7 | 0.232 |
| 13 |  | 2.333 |  | 8 | 0.185 |
| 22 |  | 1.9 |  | 9 | 0.171 |
| 18 |  | 1.545 |  | 10 | 0.171 |
| 20 |  | 1.333 |  | 11 | 0.168 |
| 5 |  | 1.307 |  | 12 | 0.159 |
| 11 |  | 1.143 |  | 13 | 0.287 |
| 7 |  | 1.067 |  | 14 | 0.27 |
| 6 |  | 0.875 |  | 15 | 0.515 |
| 17 |  | 0.706 |  | 16 | 0.255 |
| 12 |  | 0.579 |  | 17 | 0.01 |

**9.5.4** **美国大学橄榄球队**

美国大学橄榄球队是美国足球运动会在2000年常规赛期间的数据集。骨干度算法可 以将这个数据集划分为12个网络社区，CNM 算法将网络划分为5个社区。美国大学橄榄 球队应用骨干度算法的结果如图9-6所示。在这一结果中，六个社区同标准数据集比较其 结果完全一致， 一个社区比标准数据集少了一个顶点，两个社区比标准数据集多了2个顶 点， 一个社区有3个顶点和标准数据集不同，两个社区同标准数据集有5个顶点的差异。美 国大学橄榄球队数据应用CNM 算法的结果如图9-7所示。在表9-9和表9- 10 中分别展现 骨干度算法和 CNM 算法对该数据集的膨胀度和传导性。通过比较骨干度算法结果优于 CNM 算法的结果，因为虽然 CNM 算法的传导性和膨胀度表现更好，但是 CNM 算法的结 果不能反映美国大学橄榄球队人际关系网的实际结构。这证明了骨干度算法能准确地在社 交网络中进行社区发现。

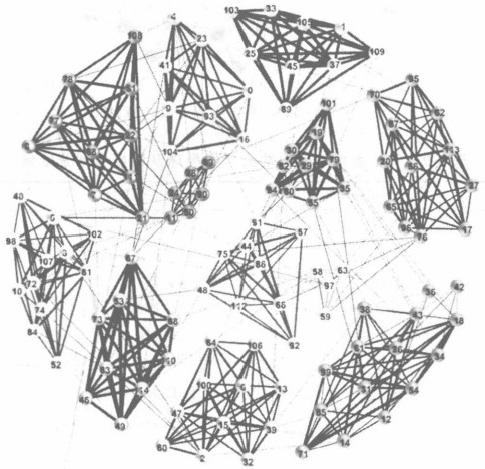


图9-6 美国大学橄榄球队应用骨干度算法的结果

**9.5.5** **安然电子邮件公司数据集**

安然电子邮件通信网络涵盖了所有电子邮件通信范围内的数据集约50万封电子邮件 (Leskovec,Lang et al.2010)。这一数据最初是公开的，并有联邦能源管理委员会在调查 过程中公布到网络上。网络的节点是电子邮件的地址，如果一个地址 i 至少向地址j 发 送 过至少一个电子邮件，那么这个图中就包含了一条从 i 到 j 的无向边。

将 CNM 算法和骨干度算法应用于安然电子邮件公司数据集，其传导性的散点图如 图9-8所示，在这一结果中，发现我们的算法略好于 CNM 算法在传导性方面。我们的算法 的传导性大多是比 CNM 低，所以比CNM 算法更紧凑、更稳定。

大数据技术及行业应用



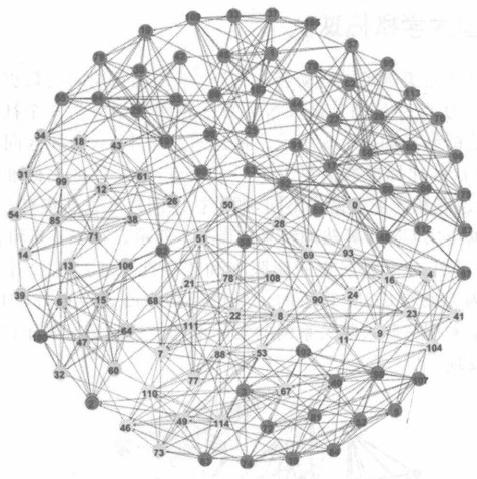


图9-7 美国大学橄榄球队应用CNM 算法的结果

**表9-9** **骨干度算法对美国大学橄榄球队数据集划分结果**

|  |  |  |  |
| --- | --- | --- | --- |
| Community ID | Community size | Conductance | Expansion |
| 0 | 9 | 0.258 | 2.778 |
| 1 | 10 | 0.352 | 3.800 |
| 2 | 9 | 0.294 | 3.333 |
| 3 | 10 | 0.273 | 3.000 |
| 4 | 15 | 0.240 | 2.400 |
| 5 | 8 | 0.364 | 4.000 |
| 6 | 9 | 0.354 | 3.778 |
| 7 | 12 | 0.262 | 2.833 |
| 8 | 12 | 0.250 | 2.667 |
| 9 | 11 | 0.290 | 3.273 |
| 10 | 6 | 0.483 | 4.667 |
| 11 | 4 | 0.657 | 5.750 |

**表9-10** **CNM** **算法对美国大学橄榄球数据集划分结果**

|  |  |  |  |
| --- | --- | --- | --- |
| Community number | Community size | Conductance | Expansion |
| 0 | 19 | 0.239024 | 2.578947 |
| 1 | 32 | 0.193084 | 2.09375 |
| 2 | 15 | 0.329268 | 3.60000 |
| 3 | 22 | 0.262712 | 2.818182 |
| 4 | 27 | 0.218978 | 2.222222 |

第9章 大数据技术在社交网络研究中的应用

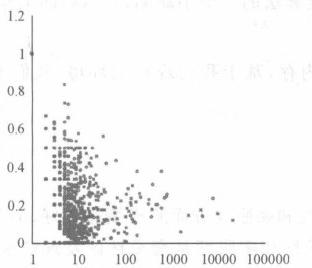
Conductance

Expansion

Conductance

Expansion

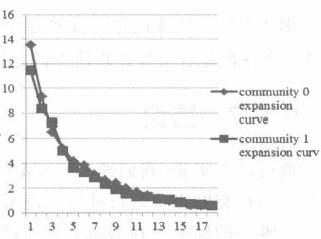
应用CNM 算法之后，安然电子邮件公司的网络膨胀散点图如图9-9所示。从结果中 发现骨干度算法传导率略小于CNM 算法，而膨胀度远远高于CNM 算法，但是骨干度算法 比CNM 算法更紧凑和稳定。



Community size

·CNM Conductance·Our Conductance

图9-8 应用骨干度算法和 CNM 算法在安然电子 邮件公司数据集中的传导性散点图



xth join

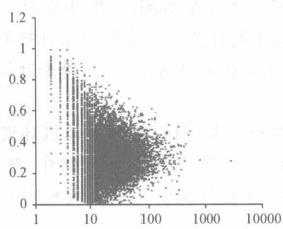
图9-9 膨胀曲线

我们发现，骨干度算法和 CNM 算法一般在传导率和膨胀度有相同的分布。另外，这些 结果表明安然电子邮件公司通过骨干度算法进行社区发现的结果非常接近 CNM 算法。在 大规模的社交网络中，很难说哪一种算法比较好，因为除了社交网络数据之间的关系外，还

有很多其他的隐性因素。

**9.5.6** **DBLP 合作网络**

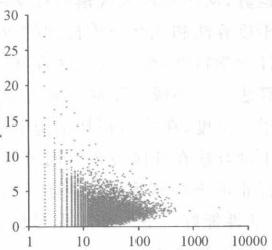
将骨干度算法应用于 DBLP 网络的传导率散点图如图9-10所示，横轴是社区规模的大 小，纵轴是社区的传导性。随着社区规模的增大，传导性逐渐减小，并在大约0.3的地方趋 于稳定。在 DBLP 网络中，大多数社区大小是10～100,这一结果也符合大多数社交网络中 人数在100人以下的常识。



Community size

Our\_conductance

图9-10 DBLP的传导性散点图



Community size

Our\_expansion

图9-11 DBLP的膨胀度散点图

将骨干度算法应用于 DBLP 网络的膨胀度散点图如图9-11所示，横轴是社区规模的大 小，纵轴是社区的膨胀。随着社区规模的增加，膨胀度逐渐减小，并在大约2的时候趋于稳 定。我们发现在10个顶点以内规模的社区中，会有更大的膨胀度，这和 CNM 算法划分的 安然电子通信公司网络的结果一样。这是骨干度算法的一个小缺陷，在以后的工作中，我们

将弥补这个缺陷。

因为 CNM 算法比骨干度算法需要更多的内存，基于我们现在的环境，我们不能得到

DBLP 合作网络在 CNM 算法下的运行结果。

**9.5.7** **结论**

通过以上实验，我们发现 CNM 算法在传导性和膨胀度方面的成绩非常好，但是在小数 据集上的社交网络中进行社区发现不准确。传导性和膨胀度是衡量社区发现结果的标准， 但不是唯一的标准。在 DBLP 合作网络和安然电子通信公司两个数据集中，骨干度算法和 CNM 算法具有类似的性能，但是在相同的实验条件下，骨干度算法可以处理的数据比 CNM 算法更多。因此，骨干度算法在社交网络的社区发现中，在发现社区结构和内存需求 方面优于 CNM 算法。但是对于那些10顶点以内的社区，骨干度算法膨胀度比 CNM 算 法 更大，这是一个缺陷。我们将在后期的工作中，弥补这个缺陷。



**本** **章** **小** **结**

在本章中，我们专注于社交网络中的非重叠社区的社区发现问题，这是理解网络功能和 结构的关键工具，在这方面，已经有了很多的研究和开发，我们已在本章中讨论了这些研究 的瓶颈。我们基于骨干度和膨胀度，开发骨干度算法，实验证明，在发现社交网络的结构方 面，骨干度算法优于CNM 的算法，同时骨干度算法需要的内存小于CNM 算法。骨干度算 法基于骨干度寻找核心骨干，然后根据核心骨干的膨胀来寻找社区发展的趋势。它从本地 网络开始，然后扩展到全球网络。首先，最大限度地减少每个社区的膨胀度，保证整个网络 中每个社区的膨胀度是最小的，从而达到全球最佳。这些特征可以保证骨干度算法的扩展 到并行运算，从而处理大规模的社交网络。

骨干度算法和9.2节中提到的所有的社区发现算法都是不同的，因为它是一个基于生 物学和社会学特性，命名为社区森林的模型，同时骨干度算法是一个简单而直接的进行社区 发现的算法。它集成了膨胀度和骨干度，膨胀度是用来区分社区的界限，骨干度综合网络权 重和邻里互惠度，在网络拓扑结构中最为平衡。

骨干度算法在社区发现方面具有良好的有效性和精度，但它目前在单机环境中无向网 络中进行非重叠社区发现，并有一 点小缺陷，如在10个顶点以下的网络中，膨胀度不如 CNM 算法性能好。下一步工作是优化骨干度算法，并使其调整为适应大规模并行环境下 重叠社区发现的算法。

**参** **考** **文** **献**

[1]Cui,Y.,et al.(2014).“Detecting community structure via the maximal sub- graphs and belonging degrees in complex networks."Physica A Statistical Mechan- ics &.Its Applications 416(C):198-207.

[2] Cui,Y.,et al.(2014).“Detecting overlapping communities in networks using the maximal sub-graph and the clustering coefficient."Physica A Statistical Mechanics &.Its Applications 405(405):85-91.

[3] Ding,Z.,et al.(2016).“Overlapping Community Detection based on Network De-

composition.”Scientific Reports 6.

[4] Easley,D.and J.Kleinberg(2010).Networks,Crowds,and Markets,Cambridge

University Press.

[5] Eustace,J.,et al.(2015).“Overlapping community detection using neighborhood ratio matrix.”Physica A Statistical Mechanics &.Its Applications 421:510-521.

[6] Li,J.,et al.(2014).“Uncovering the overlapping community structure of complex networks by maximal cliques."Physica A Statistical Mechanics &.Its Applications

415:398-406.

[7] Newman,M.(2010).Networks:An Introduction,Oxford University Press,Inc. [8] Xu,Y.,et al.(2015).“A novel disjoint community detection algorithm for social

networks based on backbone degree and expansion.”Expert Systems with Applica- tions 42(21):8349-8360.

[9] Airoldi,E.M.,et al.(2008).“Mixed Membership Stochastic Blockmodels.”Jour-

nal of Machine Learning Research Jmlr 9(5):1981-2014.

[10] Arora,S.,et al.(2004).“Expander flows,geometric embeddings and graph par- titioning.”Mutation Research/fundamental &.Molecular Mechanisms of Mutagene- sis 56(2):181-197.

[11] Clauset,A.,et al.(2005).“Finding community structure in very large net- works.”Physical Review E Statistical Nonlinear &.Soft Matter Physics 70(6 Pt 2): 264-277.

[12] Easley,D.and J.Kleinberg(2010).Networks,Crowds,and Markets,Cambridge

University Press.

[13] Girvan,M.and M.E.J.Newman(2002).“Community structure in social and bi- ological networks."Proceedings of the National Academy of Sciences of the United States of America 99(12):7821-7826.

[14] Kannan,R.,et al.(2000).“On Clusterings:Good,Bad and Spectral.”Founda- tions of Computer Science Annual Symposium on 51(3):367-377.

[15] Karypis,G.and V.Kumar(2006).“A Fast And High Quality Multilevel Scheme

For Partitioning Irregular Graphs.”Siam Journal on Scientific Computing 20(1):

359-392.

[16] Kleinberg,J.(1999).“Authoritative Sources in a Hyperlinked Environment.”

Journal of the Acm 46(5):604-632.

[17] Leighton,T.and S.Rao(1999).“Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms."Journal of the Acm 46(6):

787-832.

[18] Leskovec,J.(2014).“Can Cascades be Predicted?”.

[19] Leskovec,J.,et al.(2015).“Community Structure in Large Networks:Natural Cluster Sizes and the Absence of Large Well-Defined Clusters.”Internet Mathemat- ics 6(1):29-123.

[20] Leskovec,J.,et al.(2010).“Empirical Comparison of Algorithms for Network

Community Detection.”Proc Www':631-640.

[21] Radicchi,F.,et al.(2003)."Defining and identifying communities in networks." Proceedings of the National Academy of Sciences of the United States of America 101(9):2658-2663.

[22] Rapoport,A.(1953).“Spread of information through a population with socio- structural bias:I.Assumption of transitivity.”Bulletin of Mathematical Biology 15 (4):523-533.

[23] Rosvall,M.and C.T.Bergstrom(2007).“Maps of Random Walks on Complex Networks Reveal Community Structure."Proceedings of the National Academy of Sciences of the United States of America 105(4):1118-1123.

[24] Seidman,S.B.(1983).“Network structure and minimum degree ☆.”Social Net-

works 5(3):269-287.

[25] Spielman,D.A.and S.H.Teng(1996).“Spectral partitioning works:Planar

graphs and finite element meshes ☆.”Foundations of Computer Science Annual

Symposium on 421(s 2-3):96-105.

[26] Xie,J.,et al.(2011).“Overlapping community detection in networks:The state-

of-the-art and comparative study.”Acm Computing Surveys 45(4):115-123.

[27] Yang,J.and J.Leskovec(2012).“Defining and Evaluating Network Communities

Based on Ground-Truth.”Knowledge &Information Systems 42(1):745-754.

**第10章** **大数据技术在文本挖掘** **和情感分类中的应用**

随着互联网和社交网络的蓬勃发展，人类习惯于在互联网和社交网络上发表自己的观 点，这些观点文本中蕴含着丰富的信息，可作为商业、政治和文化等研究的样本数据，同时这 些数据又是海量的，并且伴随着自媒体的兴起，这些海量信息每时每刻都在产生，依靠传统 的手段去对这些文本进行分析和挖掘是无法满足用户需求的。因此必须采用大数据的相关 工具和方法对这些海量数据进行量化、分析和挖掘。我们通过对国内相关电商平台商品评 论的抓取、评论信息抽取、建立相关模型，从而挖掘海量评论信息中的观点，为厂商提供决策 支持。

观点挖掘(Opinion Mining)近年来已成为互联网信息应用领域的热门研究方向。最初 观点挖掘的研究集中在对评论文本进行简单的褒贬分类，随着研究的深入以及实际应用的 需要，基于被评论对象特征进行观点挖掘由于更具实用价值，因此也吸引了越来越多研究者 的关注。然而，互联网上针对某一被评论对象的评论(Reviews) 往往是海量、杂乱、非结构 化的，甚至充斥着相互矛盾的观点以及垃圾评论内容。要从这些海量的评论中挖掘出有价 值的观点也绝非易事，无论是将非结构化或者半结构化的文本信息结构化，还是对文本信息 进行情感分类，都面临诸多挑战。



**10.1** **研究综述**

本节对基于被评论对象特征进行观点挖掘的问题进行了形式化定义，并结合具体评论 示例进行了阐述，然后概括了针对该问题的三个不同层次的研究内容。

**10.1.1** **基于产品特征的观点挖掘研究**

互联网上的文本信息通常可被分为两类：事实(facts) 和观点(opinions) 。 事实是对事 物、现象、话题等被评论内容的客观表述，观点则是作者通过评论表露出的带有个人主观色 彩的内容。互联网上的观点可以针对任何事物，例如某种商品、某个服务、某些机构、某类社 会事件、某个话题，或者某些人等。表达观点的途径也多种多样，可以通过论坛、博客、邮件、 微博、评论网站等各种媒介。如果用对象(object) 一词来形容被评论的实体，则这个对象往

往由许多部件组成，或包含不同的属性。因此被评论的对象可以按照不同层次进行拆分①。 例如， 一个产品通常由许多部件组成(一个手机由屏幕、电池、外壳、通信模块等组成),一个 大的社会事件通常也包含许多小的事件， 一个话题可以含有许多子问题等。在英文领域，针 对基于产品特征进行观点挖掘的问题，Liu 等人的工作最有代表性，尤其是 Ding 与 Liu 在 文献[9]中对该问题给出了更加形式化的定义，参考其内容在本章中把评论中被描述的对象 定义如下。

**定义10** **.** **1** 对 象(object): 一 个对象 O 指的是 一 个区别于属性(attribute) 而存在的实 体，可以是一件商品、 一个事件、 一个人、 一个机构或一个话题等。它由相关联的两部分组 成 ：(T,A), 其 中 T 是指的这个对象按层次划分时的组成部件，A 是 O 的属性集合，而每个 部件也都有其子部件和属性集合。

以手机评论为例，每款手机都是一个对象，有一个部件集合，如电池、屏幕、外壳等，还有 一个属性集合、如大小、价格、通话质量等。而它的屏幕也有自己的属性集合，如像素、屏幕 大小等。本质上来说， 一个对象可以用一个树形结构来表示。根节点为这个对象本身，其他 节点都是组成这个对象的一个部件或者子部件，每一条边表示一个组成关系，每个节点同样 对应一个属性结合。而用户所表达的观点可以只针对这个节点本身，也可以针对这个节点 的任何属性。例如，评论者可以表达自己对这个手机本身的观点(根节点),如“我不喜欢这 个手机”;也可以表达对这个手机的属性的观点，如“这个手机的通话质量很差”。同样，评论 者可以表达任何对于这款手机的部件或者部件属性的观点。为了简化问题的表示方法，在 本章中用特征(feature) 来统一表示对象的部件和属性，这样可以省略掉分层的工作。用特 征来描述一件产品是非常实用的，因此对于普通用户来说，用分层的结构来描述产品的不同 特征和观点会使信息获取变得复杂。

**定义10** **.2** 显式特征和隐式特征(explicit and implicit feature):如果 一个特征 f 直接 出现在一条评论中，则称f 为一个显式特征。如果 f 没有直接出现在评论r 中，但是通过其 他内容可以暗示出该评论所描述的是这个特征，则称 f 为一个隐式特征。

例10.1和例10.2是两条关于手机的评论，其中被评论对象是“这款手机”,它被评论的 特征分别是“外观”“价格”“屏幕”和“电池”,其中“外观”“屏幕”和“电池”都直接出现在了评 论文本中，属于显式特征，而“价格”并未直接出现，“便宜”一词暗示了该语句所描述的是“价 格”,因此属于隐式特征。此处的“便宜”一词可以被称为特征指示器，可以由它来推断和定 位评论所描述的隐式特征。

**【例10.1】** 这款手机外观很时尚，也很便宜。

**【例10.2】** 屏幕非常细腻，但是电池刚用了一天就没电了。

更进一步，可以定义显式评论与隐式评论。

**定义10** **.3** 显式评论和隐式评论(explicit and implicit sentence):如果一条评论语句至 少包含一个显式特征，则称该语句为显式评论。而如果一条评论语句仅仅包含隐式特征，称 其为隐式评论。

① 本章提出的研究方法是在电商网站的产品评论上进行实验验证与结果分析的，并可直接应用于其他领域的评 论分析。为了简化起见，本章叙述时都基于产品评论分析这一具体方向进行阐述，此时的“对象”为某一“产品”,特别以 手机评论分析作为文中的具体示例。

第10章 大数据技术在文本挖掘和情感分类中的应用

如上所述，评论可以被分为事实和观点两类，然而，描述事实的评论也能在一定程度上 暗示作者的某些观点。在例10.2中，“屏幕非常细腻”表达了一个显式的主观色彩强烈的观 点，而第二句——“但是电池刚用了一天就没电了”——虽然陈述的是一个客观事实，但是它 仍然暗示了一个针对手机电池性能的贬义情感。因此，客观事实也能反映主观情感，正如定 义10.4中所述。本章的研究工作，不仅考虑了包含了显式观点的评论，也处理了评论中的 隐式观点，这也使得分析难度更加复杂。

**定义10.4** 显式观点和隐式观点(explicit and implicit opinion):针对一个特征 f 的显 式观点是直接由表达褒义或者贬义观点的主观性句子给出，而针对 f 的隐式观点是由暗示 某个观点的客观性句子给出。

**定义10.5** 观点持有者(opinion holder),针对一个特定观点的持有者是指持有这个观 点的个人或者组织。

定义10.5阐述了观点持有者的含义，在产品评论、博客、论坛的帖子中，观点的持有者 通常是发表这些内容的作者。然而也有例外，例如，在一篇酒店评论中可能有以下内容“小 李给我说这家酒店性价比很高，环境也很不错，可是现在看起来根本不是这么回事啊，不但 价格贵，房间也不干净”。在前两个句子中，观点的持有者就是“小李”而不是作者本身。

评论观点挖掘的最终目的是分析其中针对产品特征所透露出的情感倾向，观点的情感 倾向定义如下。

**定义10.6** 观点的情感倾向(sentiment orientation of an opinion):某个观点针对某个 特征 f 的语义倾向是指这个观点是否为褒义，贬义或中性。

综合以上内容，可以针对基于对象特征的观点挖掘问题给出一个形式化的模型定义。

被评论对象D 是由一个有限的特征集合来描述的，F={fi,f₂,…,f,} 。 每个特征 f; 可以

由一个有限的同义词或者短语的集合W; 来表达。也就是说可以用一个同义词集合的集合 W={W₁,W₂,…,Wn} 来表达n 个特征，f;∈W; 。 每条评论的作者或者观点持有者j 都是针 对特征集合F 的子集S, 发表评论，S, F 。对于每一个观点持有者j 所评论的特征f(f

∈S,), 他/她可能会从W, 中选择一个词或者短语来描述他/她所评论的这个特征，也可能 通过其他内容暗示他/她所评论的特征，并且表达出褒义、贬义或者中性的观点。

上述模型涵盖了产品评论中的绝大多数情况，但是也有例外。例如，“这个手机的耳机 插孔和它的开关按钮离得太近了”。此句对手机的两个不同组成部件间的距离表达了一种 否定的情感，但是却不是针对某个特征进行单独评论的，上述模型并未涵盖这种情况。不过 由于此类情况在实际评论中较少，为了简化模型，在本章工作中直接忽略了这类情况，

针对以上模型，给定一个评论集 D 合作为输入，有以下任务需要处理：鉴别并抽取每条 评论 d(d∈D) 中所讨论的产品特征，由于不同评论作者通常会用不同的词语表达同一特 征，因此需要对所有特征词进行相似概念聚类；鉴别并抽取描述每个特征的所有产品评论； 判断针对每个特征的所有观点的情感倾向；归纳高置信度的评论热点，生成情感文摘。本章 讨论的内容针对每条评论文本d 的最终输出是一个特征、情感倾向以及其他相关信息的集 合，可以用结构化的数据表示。该集合中的每个元素是一个四元组，表示为(oy,fx,sou,

h₁), 其中o; 是被描述的对象，fα 是该对象的一个特征，sog 表示文本d 中观点持有者i 针对 对象o; 的特征fx 所表达的情感倾向(sentiment orientation),可以是褒义或贬义(由于中性 的观点实际用途不大，此处可以忽略),h; 表示观点持有者i。

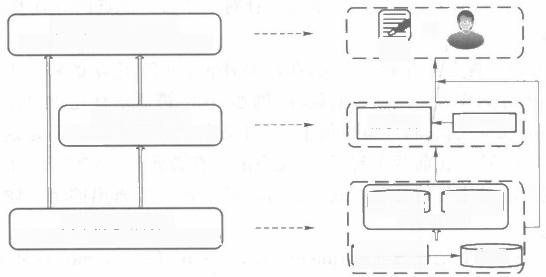
评论文本观点挖掘的主要内容是对文本进行处理、分析、归纳、总结，最后以结构化的信 息表示方法展现实验结果。综合上述定义，整个观点挖掘的处理流程可以概括性地分为三 个层次：

· 文本信息结构化

· 评论信息分类

· 观点挖掘结果归纳

其中评论信息分类是在结构化信息基础上进行的，而观点挖掘结果归纳则综合了前两 步的结果。本章主要的研究内容与这三个层次的对应关系如图10- 1所示。后文将分别给 出这几个部分当前的研究现状综述，并引出本章的核心研究内容。



观点挖掘结果归纳

(特征的情感倾向，用户的讨论特点)

摘要

评论信息分类

(褒义类分类，主客观性分类)

愿示评论 隐式评论 特征相关的评论

产品评论 特 征

文本信息结构化

(产品特征及对应评论，观点持有者，情感词训

情感检测

中文词典

-

图10-1 观点挖掘研究的层次框架

**10.1.2** **产品评论结构化信息抽取方法**

互联网上的产品评论往往是海量的、杂乱的、非结构化的，而基于产品特征进行观点挖 掘，需要以机器方便处理的结构化信息为基础。文本信息结构化的主要工作包括抽取对象 的所有被评价的特征，每个特征对应的所有评论，观点持有者，以及情感词语等有实际意义 的信息单元。例10.3这条手机评论中所包含的产品特征包括：“外观”“音质”“价格”“屏 幕”,观点持有者是作者本身，情感词语包括“大方”“不错”“划算”“漂亮”,其中“价格”是隐式 特征。这些基本信息是后续基于对象特征进行情感分类的基础，而对每类信息的抽取，当前 已有许多成熟的研究方法。

**【** **例** **10.3】** 这款手机外观大方，音质不错，我是2980元买的，所以觉得比较划算，可是

屏幕分辨率太小了。

**1.** **产品特征抽取方法**

产品的组成部件和属性统一用特征表示，产品特征抽取就是要抽取评论文本中作者所 描述的主体。例如，在手机评论中，部分作者可能只评论手机屏幕，而其他作者则会评论手 机音质或者电池等。在现有的产品特征提取工作中，大部分研究都主要提取了频繁出现的 名词或者名词短语，然后进行处理，通过同义词处理，相似概念聚类等方法，提取出用户主要 评论的产品特征。例如 Hu 和 Liu 通过关联规则挖掘的方法提取了评论文本中频繁出现的 名词以及名词短语，再通过定义规则进行剪枝去除噪声，从而提取出被描述的产品特征。这 种方法的缺点是无法提取出被评论比较少的产品特征。然而，用户评论较少的产品特征往

第10章 大数据技术在文本挖掘和情感分类中的应用

往也是不重要的，或者不被用户关注的特征，因此再继续挖掘非频繁出现的产品特征的实际 价值已经不大。Zhai 和 Liu 等人通过添加软性限制，对提取出的产品特征进行了聚类，合并 了相似概念(例如“picture”和“photo”在相机评论中都指的是相机拍摄的照片)。Popesc 和 Etzioni 则通过考察提取的候选特征与领域指示词之间的关联度来抽取产品特征，然而领域 指示词的获取却是一个难点。

**2.** **特征相关评论抽取方法**

与对象特征相关的显式评论可以轻易获得，而隐式评论抽取却比较复杂。隐式特征指 的是未直接出现在评论中，而通过其他内容暗示的对象特征。Hu 和 Liu 最早在文献[8]中 提出了隐式特征这一概念，然而，他们在文献[6][7]中直接忽略了此问题，在文献[8]中，他 们提出可以通过规则挖掘的方法解决此问题，却并没有进行深入讨论，也未给出任何实验结 果。本章实验部分抓取的数据显示， 一般的产品评论中大约有 25% ～30%左右的隐式评 论，如果直接忽略这一部分内容，无疑会显著降低观点挖掘的召回率，并且遗漏一些用户关 注的焦点问题。

Su 等人在文献[12]中利用点对互信息(Point-wise Mutual Information,PMI)分析语义 关联度，并推断每个情感词所评论的产品特征。然而，该文章中仅仅给出了部分示例情感词 的 PMI 计算结果和少部分的规则挖掘结果，并未详细评估该方法的有效性。在文献[13] 中 ，Su 等人利用产品特征和情感词的内容信息以及情感关联信息分别对其进行了聚类，并 通过内部关联度将产品特征簇与情感词语映射起来。虽然该工作能发现产品特征与情感词 语之间的一些隐藏关系，但是他们提出的方法仅仅考虑了形容词作为情感词的情况，而且并 未定量分析隐式特征挖掘的效果。Hai 等人在文献[14]中利用共现的关联规则挖掘方法来 定位隐式特征，他们首先为所有在显式评论文本中出现的情感词构建了一个(情感词，显式 特征)的对应列表，当在评论语句中出现情感词并且不含显式特征时，在之前的列表中搜索 并选出频率最高的规则将其作为该句的隐式特征。他们的方法取得了比较好的实验结果， 然而，他们的实验数据集相对偏小，并且评判实验结果时仅仅计算了和情感词有关的隐式评 论的准确率与召回率。该工作的一个不足之处是只处理了由情感词所暗示的隐式特征，而 正如前文所述，即便是事实性陈述，也能暗示作者的某些情感观点，因此该方法会降低隐式 特征挖掘的召回率，从而遗漏部分评论热点。

**3.** **观点持有者抽取方法**

在产品评论中，作者往往表达的是本人对该产品的观点，观点持有者一般是作者自己， 因此很少有学者研究产品评论领域的观点持有者抽取。然而也不尽然，在例10.4中，前两 句话的观点持有者都是“很多人”,却非作者自己。这种情况实际上会对情感分类的结果带 来误差，因为这条评论的作者对这款手机是持贬义态度的。然而，这类特殊情况在产品评论 中所占比例非常低，因此本文的工作中直接忽略了这类情况，默认产品评论的观点持有者就 是作者本人。

**【例10.4】** 很多人觉得这款手机屏幕很华丽，外形也很漂亮，我却没啥感觉。

与产品评论分析不同的是，在新闻评论分析中，定位观点持有者却意义重大，因为新闻 评论一般涉及了许多不同组织或个人对于某事件的不同态度，所以需要通过观点持有者的 抽取来确定不同组织个人的观点以及情感倾向。现在的一些研究主要集中在使用命名实体 识别技术来进行抽取，或者利用机器学习的方法，如条件随机场(Conditional Random

Fields,CRF) 、 最大熵模型(Maximum Entropy,ME)来进行抽取。

**4.** **情感词抽取方法**

情感词语在文本情感倾向的判断中非常重要，因此情感词语的抽取一直是非常热门的 研究课题。主流的研究工作有两个方向：基于词典计算和基于语料库统计。

基于词典计算的情感词抽取方法主要通过计算不同词语之间的语义相似度来抽取情感 词语，常用的词典是WordNet (英文)和 HowNet (中文)。该方法一般先设定一个情感种子 词典，然后通过语义计算其他词语与种子词的语义相似度，从而得到一个扩充过的情感词 典。然而，由于种子词典中词语数量、质量、涵盖范围的限制，用这种方法进行扩充得到的情 感词典往往精度不高，而且个别词语由于有多种语义，往往也会同时在褒义词典和贬义词典 中出现。在中文领域，Ku 等人在情感词典的基础上提出了 一套计算情感词语极性权重的 方法，得到了带有权重的情感词典。Fu 等人则通过在现有的情感词典基础上，使用改进的 卡方统计的算法计算了单个汉字的情感权重，以此为基础计算了词语、短语的情感权重。总 的来说，基于词典的方法，由于涵盖面广，因此召回率较高，但是由于存在一词多义的情况， 往往会出现一些歧义词。例如，“这家饭店卫生状况很好”和“这家饭店一点都不卫生”,此处 “卫生”做名词时没有情感色彩，但是做形容词时则是一个褒义词。基于词典的方法还有一 个缺点是不能处理情感极性随着不同领域以及不同的产品特征动态变化的情况。在中文领 域，虽然前人已有相当多的研究工作，然而当前仍然没有一个比较权威的中文情感词典。

不同于基于词典的方法，基于语料库统计的情感词抽取方法主要是通过从大的语料库 中进行统计，发现一些包含情感色彩的词语并进行褒贬义分类。早期的工作包括利用连接 词“but”“and”等进行统计，例如，“excellent and awesome”“beautiful but thin”。Turney 基 于信息检索的方法提出了一种新的解决思路，通过计算搜索引擎检索结果中目标词语与种 子词之间的互信息，来衡量词语的情感倾向以及权重。基于语料库统计的方法虽然操作简 单，但是由于高质量的评论语料库比较少，并且和领域密切相关，情感词分布也不均匀，因此 效果不是尽如人意。

**10.1.3** **评论信息分类相关研究方法**

评论信息分类是在结构化的文本信息基础上，对用户的观点进行分类，按照不同的分类 目的，可分为褒贬义情感倾向分类，文本主客观性分类。其中，情感倾向分类又可以按照不 同的粒度分为单字、词语、短语、句子、篇章等不同等级的分类。在例10.3中，按照主客观性 分类，“这款手机外观大方，音质不错”和“所以觉得比较划算，可是屏幕分辨率太小了”都是 主观性文本，而“我是2980元买的”是客观性文本。按照情感的倾向分类，显然“这款手机外 观大方”“音质不错""所以觉得比较划算”都是针对不同产品特征的褒义评论。

当前文本的主客观分类研究主要有两大类方法， 一类方法是判断文本中是否含有情感 信息，另一类方法是将主客观文本分类看成是一个二分类问题，然后抽取文本特征，并利用 机器学习的算法进行分类。如前文所述，由于客观事实也能反映主观情感，因此本章的情感 分类工作不仅考虑了主观性评论，也处理了客观性评论，故没有专门研究文本主客观性分类 问题。

情感分类作为评论文本信息分类的热门研究课题，其起源甚至可以追溯到20世纪70 年代。随着 Web2.0 的发展壮大，该研究方向真正开始广泛进入人们的视野是从20世纪初

第10章 大数据技术在文本挖掘和情感分类中的应用

开始的。文本的情感倾向分类工作按照不同粒度可以分为单字、词语、短语、句子、篇章等不 同等级的分类。其中单字、词语、短语的情感分类实际上就是情感词语抽取以及分类的过 程。评论文本中句子、篇章的情感分类一般都被看成是褒贬义二分类的问题。产品评论的 情感分类主要有两个研究方向。其中一类研究方向是侧重于将文档整体分为褒义、贬义或 中性，这类工作仅仅在篇章粒度上对产品评论进行了分类，而忽略了与产品特征相关联的具 体情感信息，因此实用价值不大。另一大类研究方向专注于基于产品特征进行情感分类，这 类工作通常是基于语句粒度对文本进行情感分类，最终目的是生成一个针对不同特征的情 感摘要，该方向也是本章的主要研究内容。文献[6][7][40 ]是该研究方向比较有开拓性的 一些工作，其他相关研究有[11][41][9]等。

为了实现文本的情感分类，主流的研究工作一般有两大类的方法：基于词典的无监督分 类方法和基于属性选择、机器学习的监督分类方法，也有 一 些研究工作将这两种方法相 结合。

**1.** **基于情感词典的情感分类方法**

基于情感词典的情感分类方法由于能在句子粒度上对文本进行情感分类，因此是基于 产品特征进行评论挖点挖掘的主流研究方法。Hu 和 Liu 在情感词典基础上抽取了含有褒 贬色彩的观点词语，然后加权求和，计算了文本的情感得分。在此工作基础上，Ding 和 Liu 等人进一步提出了一套整合的观点挖掘算法。该工作不仅利用了现有的情感词典，还针对 一些特殊的句子结构制定了不同的规则，包括否定结构、转折结构等，并且为了判断部分情 感倾向随不同领域以及不同特征动态变化的词语情感倾向，他们还提出了一种通过一些特 殊句式在语料库中进行统计继而判断这些词语情感倾向的方法，得到了不错的实验结果。 在中文信息处理方面，Ku 等人研究了中文情感词的权重计算方法，进而依据情感词语权重 对文本进行了情感分类。

**2.** **基于机器学习的情感分类方法**

在基于机器学习的监督分类方法中，Pang 等人在文献[36]中开拓性的首次将机器学习 的方法运用于文本情感分类领域，通过向量空间模型(Vector Space Model,VSM)将文本特 征转换为数学向量，并利用三种机器学习领域的分类器进行了测试。结果表明在抽取的几 种不同的特征中，unigram 的分类效果最好。在后续的研究中，许多研究者也尝试提取了其 他的文本特征来进行分类。基于机器学习的方法进行文本情感分类的一个局限就是用一个 领域的已标定文本进行训练，得到的分类模型应用到另一个新领域，往往分类效果较差。这 是由于很多词语的情感倾向是和具体领域以及语境相关的。例如，“这个房间的墙壁太薄 了”和“这款手机很薄”,“薄”这个词在这两句话中就有不同的情感倾向。

**3.** **结合情感词典与机器学习的情感分类方法**

基于情感词典的情感分类方法由于受情感词典的完整度与准确度的影响，效果会有一 定降低，而基于机器学习的方法由于受到训练语料的限制，应用范围有所局限，为了解决这 个问题，Tan 等人在文献[42]中提出了一种将基于词典的技术和基于机器学习的技术相结 合的情感分类方法。该方法首先利用基于词典的方法对目标领域的部分文本进行分类，选 取高置信度的分类结果作为训练集，再训练分类模型，对剩余的文本进行褒贬义分类，实验 结果显示这种方法能在一定程度上克服利用机器学习进行分类的领域局限性。然而，这种 策略依然没有摆脱利用机器学习方法进行情感分类粒度较粗的局限性。

本节对基于产品特征进行观点挖掘的若干子问题的研究现状进行了系统的综述，并分 析了当前的一些主流研究方法的优缺点。在从评论文本抽取结构化信息的过程中，当前研 究在中文领域的产品特征抽取工作仍然有所欠缺，而绝大多数研究工作都忽略了隐式特征 抽取这个问题，直接影响了系统的召回率。针对这种情况，本章后续内容重点研究了评论文 本结构化信息抽取过程中的特征抽取，尤其是隐式评论抽取的问题，从而有效弥补了当前研 究工作中一个被广泛忽略的关键问题。

针对评论信息分类的问题，当前也有丰富的研究成果，在中文领域前人也已做了相当多 的研究工作。然而，其中的大部分研究都聚焦于对整篇文档进行褒贬分类，而忽略了评论内 部的许多具体信息，尤其是在中文领域基于对象特征进行情感分类的工作依然乏善可陈。 本章的另一个研究工作致力于解决中文领域的文本细粒度情感分类问题，并重点研究了上 下文相关情感词的情感倾向判断方法，从而有效提高了情感分类的准确度，为生成全面客观 的情感摘要奠定了坚实基础。



**10.2** **评论文本的结构化信息抽取**

10.1节对观点挖掘的相关研究现状进行了系统介绍，并将基于产品特征的评论观点挖 掘工作概括性地分为了三个阶段。经调研发现，在产品评论结构化信息抽取过程中，当前针 对中文领域的产品特征抽取工作相对研究较少，而目前主流的研究工作大多忽略了和特征 相关的隐式评论的抽取。为了实现产品评论结构化信息抽取工作，本节提出了针对中文评 论的产品特征抽取方法，并重点介绍了两种不同的隐式特征抽取策略，包括算法思想、流程 以及相应细节，以期解决和特征相关的隐式评论分析问题，提高整个观点挖掘系统的召 回率。

**10.2.1** **产品特征抽取**

产品特征抽取，即概念提取，是基于对象特征进行观点挖掘的第一步①。针对中文文本 的一些特殊性，本部分所设计的观点挖掘系统首先通过关联规则挖掘的方法提取了评论文 本中频繁出现的名词以及名词短语，再定义规则进行了剪枝以便去除噪声，从而提取出频繁 被描述的产品特征。由于本部分重点关注的是被评论的热点概念，而这类名词性短语往往 具有规律性特征，根据这些特征，可定义所提取的名词性短语的语法形式(例如，形容词+名 词、名词+名词、代词/动词/形容词/名词+“的”+名词，名词+“的”+动词等)。根据这种 语法形式识别和划分名词性短语，进而提取。经过上述名词性短语的提取，得到了最原始的 候选概念集合。再通过频繁项集提取从最原始的候选集合中提取评论者最为关注、评论最 多的名词或者名词短语，作为热点概念候选集。通过频繁项提取，虽然获取了被评论最为频 繁的名词或名词短语，但这些频繁项不都是与被评论对象相关的概念特征。其中存在那些 在任何领域都被高频率提到的常用词语，而这些词语都是与评论总结无意义的频繁项，应该

① 概念提取非本文核心内容，主要基于本课题组前期的研究工作，故此仅作简略介绍，详细内容参见：吉宇婷，文 本评论相似概念检测，清华大学，2011。

第10章 大数据技术在文本挖掘和情感分类中的应用

剔除。对非评论产品属性、特征或者相关评论总结无意义的频繁项，应该剔除。因此，再对 非评论产品属性、特征或者相关概念的频繁项进行剪枝，通过分别对频繁项集中的单字单 词、多字单词(至少含有两个汉字的中文单词)和名词性短语进行剪枝，就能得到概念集合。

经过概念提取后，从原始文本评论信息中获取了与被评论产品最为相关的热点概 念。这些概念中可能有多个概念都指代评论对象的同一属性、特征或者相关概念，例如， “价格”“价钱”“价位”和“售价”都指代的同一个产品特征。为了获得高效的评论总结，需 要对这些相似概念进行聚类，将相似概念聚合为同一产品特征簇。在10.1节中已经介 绍过，文献[10]的工作是当前相似概念聚类比较前沿的研究方法，该方法假设评论中经 常用相同的内容来描述同一个产品特征，并利用热点特征概念的离散上下文信息对概念 进行了聚类。然而，其准确率却仍然未达到能投入实际应用的程度。因此，文本的观点 挖掘系统具体实现时，在提取了所有被评论的热点概念之后，并未利用机器全自动对相 似概念进行聚类，而是人工对同一个领域产品的相似特征概念进行了聚类，以满足实用 系统的准确率要求。

提取出产品特征后，需要从评论中抽取描述该特征的所有评论文本，从而分析评论者针 对该特征的所有评论的情感倾向。当前多数研究工作都仅仅提取了直接包含产品特征短语 的评论进行分析。然而，这种方法往往召回率较低，例如，“这款手机很便宜”,这句评论描述 的是手机的“价格”,而在分析评论者针对“价格”这个特征的评价时，却无法直接将这条评论 纳入和“价格”相关的评论集合中，因此需要进行隐式特征抽取。

对于上述抽取评论文本所描述的隐式特征的问题，本章中采用了两种方法： 一种是基于 关联规则挖掘的方法；另一种是基于监督学习的方法。

**10.2.2** **基于关联规则抽取评论的隐式特征**

**1.** **算法设计思想与流程框架**

对于文本所描述的隐式特征，由于并没有一个包含所有领域和所有产品特征的先验知 识库，因此需要从具体的评论文本中挖掘对应的关联信息。正如上例 “这款手机很便 宜”,虽然该评论显而易见是描述“价格”这个特征的，然而我们并没有一个包含该信息的先 验知识库，即便人工进行总结，也不可能穷举所有领域、所有产品以及所有特征。因此，针对 每一个特定领域，直接从评论文本中挖掘这类关联信息是一个比较可行的方法。如果能通 过关联规则进行挖掘，抽取出形如“便宜”→ “价格”这类规则，那么就能轻易地从隐式评论中 挖掘对应的特征了。

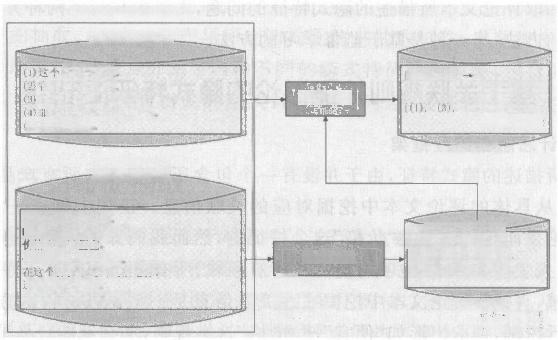
本部分通过关联规则挖掘抽取隐式特征时，首先根据描述特征的词和短语从原始数据集 中抽取包含显式特征的评论，然后利用词频统计和频繁项集挖掘的方法从中抽取出和目标特 征关联密切的词语以及多维频繁项，再利用自然语言处理领域的搭配提取方法对规则进行过 滤，从而得到了比较可靠的关联规则。将此规则用于不含显式特征的评论，就能抽取出相关特 征的隐式评论了。其中，规则的前项(antecedent) 是从和某个具体特征相关的所有显式评论文 本中得到的频繁出现的词语或者短语，这些先行词被称作特征指示器(indicator) 。 不同于前人 的研究工作，在本章中，特征指示器不仅考虑了情感词语，也考虑了其他的非情感词语，并且 还包括从显式评论中抽取的二维频繁项集。在例10.5中，“一块”是一个二维的频繁项，并 且经常被用来描述“电池”。这个句子虽然是描述的事实，但是它隐含地表露了一个贬义的

大数据技术及行业应用

情感。因此，本章中特征指示器考虑了词语以及二维频繁项等信息。生成的关联规则的结 果项(consequent) 是产品特征，整个关联规则挖掘的主要目的是找出产品特征与特征指示 器的共现关系，以达到挖掘隐式特征的目的。

**【例10.5】** 这个手机只配了一块电池。

图10-2描述了算法在一个简单的评论集上进行挖掘的流程。该算法的主要目的是从 每个特征的显式评论中挖掘特征指示器与特征词相对应的关联规则，并应用这些规则挖掘 隐式特征。在得到了分词、词性标注以及产品特征簇之后，能够轻易得到与每个特征簇相关 联的所有显式评论，其中，每特征簇被冠以了一个有代表性的描述词。在关联规则挖掘阶 段，由于已经得到每个特征的所有显式评论，算法针对每个特征的分析流程都是在该特征的 显式评论上独立进行的，因此不需要用传统的关联规则挖掘算法在整个评论文本中挖掘特 征与指示词之间的对应规则。在抽取候选特征指示器时，本部分不仅统计了单个词语，还使 用 FP-tree 来从显式评论集合中筛选二维频繁项作为特征指示器。为了计算候选特征指 示器与产品特征之间的关联程度，本部分使用了一些在自然语言处理领域常用的搭配提取算 法，包括频率(frequency) 、点互式信息(Pointwise Mutual Information,PMI)、频率\*点互式信息 (frequency \*PMI),t 检 验(t test)和x² 检 验(chi-square test)。针对每一种方法都设定了不同 的阈值以筛选有效的规则(特征指示器 →特征),如果某个特征指示器对应了多个特征，那么只 保留关联度最高的那条规则，最后，这些筛选出的关联规则被用于挖掘隐式特征。



用户评论(句子粒度)

隐式评论语句

手机很便直。

它看起来很时尚。

便宜。我正好能买得起

常便宜，(5)非常时尚。

6)真的很划算

隐式特征

显式评论语句

它的外观非常时尚。

时尚的外观，我很喜欢。

价格真的根便宜。

价格便宜， 质量可靠。

价位立来)说真的很划算。

这个价格很划算。

时尚的外观，售价也便宜。

产 品 特 证

城景端 一产品特院

时的一(外外局

便宜→ (价格，售价) 划算→ (价格，售价}

关联规则：

{外观、外形} (4),(6)] 十

价格，售价]

*复\**

*至位庭*

*师事\*互信息*

*t检验*

*x盐验*

应用规则

搭配提取

(2),(5))

图10-2 利用关联规则挖掘隐式特征

算法1以伪代码的形式总结了利用关联规则挖掘隐式特征的主要步骤。在算法中有许 多参数需要设置，并且会直接影响实验结果。当从显式评论中抽取候选特征指示器时(第6 行),所有候选特征指示器都通过词性以及最小出现次数进行了剪枝。算法中用到的搭配提 取方法(第7行)将在下一小节详细介绍。所有比置信度阈值高的候选规则都被筛选出来作 为该特征的关联规则(第8～12行)。高的阈值可以剪枝掉低频度的规则，提高准确率，但是 也会降低一些召回率。在实际应用中，需要通过实验挑选合适的阈值，最后筛选出的规则被 用于挖掘隐式特征(第13～22行)。

第10章 大数据技术在文本挖掘和情感分类中的应用

|  |  |  |
| --- | --- | --- |
| Algorithm 1利用关联规则挖掘隐式特征 | | |
| 1: | for each f;in feature clusters do |  |
| 2: | ES←extract explicit sentence set from the corpus |  |
| 3: | end for |  |
| 4: | IS implicit sentence set with no feature |  |
| 5: | for each f;in feature clusters do |  |
| 6: | CFI₁-select candidate feature indicators from ES |  |
| 7: | CR,←generate candidate rules from CFI;by collocation | selection method N |
| 8: | for each cr in CR,do |  |
| 9: | if the value of cr >threshold.then |  |
| 10: | Add cr to fs rule set R |  |
| 11: | end if |  |
| 12: | end for |  |
| 13: | for each s in implicit sentence set IS do |  |
| 14: | if the feature of S is null then |  |
| 15: | for each r in R;do |  |
| 16: | if s satisfies r then |  |
| 17: | the feature of s←f; |  |
| 18: | break |  |
| 19: | end if |  |
| 20: | end for |  |
| 21: | end if |  |
| 22: | end for |  |
| 23: | end for |  |

**2.** **利用搭配提取方法挖掘关联规则**

通过关联规则挖掘抽取隐式特征，非常类似于自然语言处理领域的搭配提取，因为此时 的输入数据集已经是和某个特征相关的所有显式评论。在自然语言处理领域， 一个搭配通 常是由两个或多个字/词按照习俗组合在一起用于表达特定的含义。搭配可以有多种形式， 例如，“make up”“strong tea”。搭配提取和关联规则挖掘有一个显著地相似之处就是两者 都是用于发现经常同时出现的元素。常用的搭配提取方法有频率、点互式信息、频率\*点互 式信息、t 检验和x² 检验等，本部分创新性的将这些方法应用到了隐式特征抽取问题中，下 面将逐一介绍。

很显然最简单的提取固定搭配的方法是计算某个搭配出现的频率。如果特征指示器和 产品特征经常同时出现，那么就有迹象表明该特征指示器经常用于修饰这个产品特征。特 征指示器必须通过词性进行剪枝，仅仅保留那些会经常被用来描述产品特征的词，例如形容 词、动词、名词等。

点互式信息是在信息论中用来衡量有趣搭配的一个指标，也经常被用于自然语言处理 的各个领域。在本章的隐式特征抽取问题中，产品特征f 和特征指示器w 之间的点互式信 息用下式计算：

 (10-1)

其 中Pgw 是特征f 和指示器w 在显式评论中同时出现的概率，P, 是特征f 在显式评 论中出现的概率，P 。是指示器w 在显式评论中出现的概率。点互式信息是一个非常好的 用于衡量独立性的指标，但是却不是一个合适的用于衡量依赖性的指标。因为对于依赖性 来说，上式得分主要取决于f 和w 各自出现的频率。在其他条件相等的情况下，f 和 w 单 独出现次数较少时得分会更高。这与提取常用搭配的初衷相悖，因为如果词语出现的频率 高，表示这个搭配很常用，可信度更好，也应该给一个更高的得分。 一种改进的方法是将频

率的信息添加进去，即频率\*点互式信息：



(10-2)

另一种经典的搭配提取方法是假设检验，这种方法经常被用来判断某个事件是否是一 个偶然事件。判断特征 f 和指示器w 同时出现是否是偶然事件，可以用假设检验的方法。 先用虚假设 H。表示 f 与w, 除了偶然同时出现之外没有其他关联，然后计算如果 H。为 true 时事件发生的概率 P, 最后根据 P 的值确定之前假设的真假。

一种常用的假设检验方法是检验。它假设样本服从均值为h 的正态分布，然后计算样 本的均值与方差。通过比较实际计算的均值与期望均值之间的差异来确定是否接受这个假 设。在本问题中i 检验可以通过式(10-3)计算：

(10-3)

其中，x是样本均值，s² 是样本方差，N 是样本空间大小，μ为分布的均值。将这种方法用到

搭配提取中，μ=P,P 。, 由 于P 非常小，近似的方差g²=P (1-Pm)≈P,X 取值为

语料库中特征 f 和指示器w 实际同时出现的概率P&m 。如果检验的值足够大，那么之前 提出的虚假设就为 false。

t 检验假设样本服从正态分布，而在实际情况中不一定总是成立，另一种不需要样本服 从正态分布的假设检验是x²(卡方)检验。基于x² 检验的评价法通过计算x 值来评估产品 特征以及指示器的关联程度。在最简单的情况下，以2×2的表10-1来说明x²检验的用法。 x²检验的实质是比较期望的独立分布频率与表中观测到的频率。如果观测到的频率与期望 的频率相差很大，那么就可以拒绝之前的独立分布的虚假设。

表10-1 一个2×2的表来展示特征“外观”与指示器“漂亮”之间的依赖性

|  |  |  |
| --- | --- | --- |
|  | w=“漂亮” | w≠“漂亮” |
| f=“外观” | 8  (例如，“外观漂亮”) | 202 |
| (例如，“外观不错”) |
| f≠“外观” | 509  (例如，“屏幕漂亮”) | 7986  (例如，“屏幕不错”) |

第10章 大数据技术在文本挖掘和情感分类中的应用

x²检验对每个表格中观测频率与期望频率的差异进行求和，并通过期望频率进行了归

一化，计算公式如下：



(10-4)

其中，i 是表的行数，j 是表的列数，o 是表格(i,j)的观测频率，e,是其期望值。期望频率e; 通过边际概率计算。例如表格10-1的期望频率(“外观漂亮”)通过用“漂亮”作为搭配的一 半的边界概率乘以“外观”作为搭配的一半的边界概率：

 (10-5)

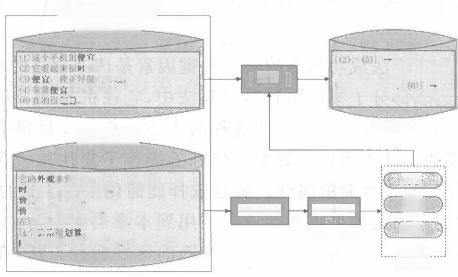
通常情况下，在搭配提取的问题中，t检验和x² 检验的区别并不大。然而，x²检验也同样 适用于大概率的情况，而这时t 检验的正态分布假设是不成立的。这也是为何x² 检验被广 泛应用于搭配提取的问题之中。

10.2.3 基于监督学习抽取评论的隐式特征

1. 算法设计思想与流程框架

和利用关联规则挖掘进行隐式特征抽取的前提类似，与每个特征簇相关的所有显式评 论是很容易得到的。正如前文所述，对于文本所描述的隐式特征，并没有一个针对所有领域 和所有产品特征的先验知识库。然而，在得到与每个特征相关联的所有显式评论之后，经常 被用于描述该特征的一些评论信息，都是包含在对应的显式评论集合中的。因此，如果对该 集合进行学习，从中抽取出用于评论该特征的有用信息，然后建立一个针对该特征的分类模 型，就能对隐式评论语句进行分类了。

鉴于此，本章提出了一种利用监督学习训练分类模型的算法用于挖掘隐式特征，将包含 显式特征的评论作为训练集，其他评论则作为预测集，算法流程如图10-3所示。该方法主 要包括以下步骤：抽取包含显式特征的相关文本；将相关文本数字化；利用机器学习的方法 训练分类模型；对其他语句是否包含隐式特征进行分类，进而判断每条评论所描述的隐式 特征。



用户评论(句子粒度)

尚

买得起。

t(5)非常时尚。

划 算

显式评论语句

常时*时勤。*

尚 的 外 观 。我很喜欢。

格 真的根便宜。

格便立，质量可靠。

这个价位来说真的很划算。

个 价 格

时尚的外藏， 售价也便宜。

分英幕

(州观外形 (),(3),(4).

价格， 售价

价格 外观 屏幕

\*

训练模型 wsy →SNY

隐式评论语句

属性选择

隐式特征

分类

图10-3 利用监督学习挖掘隐式特征

算法2以伪代码的形式总结了利用监督学习挖掘隐式特征的主要步骤。在训练时，训 练集中的正例是当前特征的所有显式评论(第6行),而负例则是其他特征的显式评论(第7

大数据技术及行业应用



行)。在模型训练阶段，本部分使用了不同的方法进行属性选择①,并用VSM 将评论文本数

字化(第8行)。

|  |  |
| --- | --- |
| Algorithm 2 利用监督学习挖掘隐式特征 | |
| 1: | for each f;in feature clusters do |
| 2: | ES←extract explicit sentence set from the corpus |
| 3: | end for |
| 4: | IS-implicit sentence set with no feature |
| 5: | for each f,in feature clusters do |
| 6: | PE←explicit sentences in ES;as positive examples |
| 7: | NE;-explicit sentences in ES,(j≠i)as negative examples |
| 8: | PD,-generate training data based on PE;and NE;by VSM |
| 9: | C;←use SVM to train classification model with TD |
| 10: | for each s in implicit sentence set IS do |
| 11: | if the feature of s is mull then |
| 12: | result -use C;to classify s |
| 13: | If result is positive then |
| 14: | the feature of s-f; |
| 15: | end if |
| 16: | end if |
| 17: | end for |
| 18: | end for |
| **2.** **监督学习方法中的属性选择以及机器学习模型**  利用监督学习挖掘隐式特征时，需要设置一些参数。在利用向量空间模型生成训练数 据时，所有的属性都通过词性以及最小出现次数进行了剪枝。不同的属性选择方法会得到 不同的实验结果。例如，如果仅仅选择形容词作为训练属性，得到的结果很可能比同时选择 名词、动词以及形容词时得到的实验结果差，因为产品特征经常使用不同词性的词语或短语 进行描述。  机器学习中所有分类方法成功与否的一个关键因素是选择合适的输入变量。属性选择 指的是从数据集合中选取部分子集应用到学习算法中，它在机器学习中通常是一个重要的 步骤，可以有效地避免高维诅咒。关键属性选择的主要工作是度量每个属性与目标值的关 联度，并选择关联度高的属性作为训练数据。最好的训练子集往往包含最少的能提高分类 准确率的属性，并舍弃其他不重要的属性。属性选择能简化预测模型的训练过程，并增强其 对未知数据的泛化性能。因此如果将属性选择应用到本章的隐式特征抽取算法中，应该能 在一定程度上提高系统准确率。当前已有许多成熟的属性选择方法，在文本中采用了信息 增益(information gain)作为度量属性重要程度的方法。 | |

① 在机器学习领域，经常用“属性”或者“特征”一词来表述数据的每一维度上的具体信息，为了区别于文本中的 “产品特征”,故本章中仅用“属性”一词表示训练集和测试集中的数据的每一维度的信息。

第10章 大数据技术在文本挖掘和情感分类中的应用 

支持向量机(Support Vector Machine,SVM)在传统的文本分类中 一 直表现出了优异 的性能。它本质上是一个最大间隔分类器，在二分类问题中，该方法的思想是找出一个用支 持向量表示的超平面，该超平面不仅能对已有的数据进行分类，并且要求间隔越大越好。由 于支持向量机在文本分类问题中性能表现突出，文本中选择其作为机器学习模型，并用 SVM-light① 进行训练和测试，所有参数均采用默认参数。

在产品评论结构化信息抽取的问题中，隐式特征往往是被前人忽略的问题，本节重点介 绍了两种截然不同的隐式特征抽取算法。第一种方法基于关联规则挖掘，并将自然语言处 理领域广泛使用的搭配提取方法应用到该问题中。第二种方法利用机器学习算法为每个特 征训练了分类模型，首次将监督学习的方法应用到隐式特征提取的问题中。最后将详细评 估本章所介绍的算法的实验结果。



**10.3** **情感分类研究综述**

在抽取了结构化评论信息的基础上，需要对评论文本基于产品特征进行情感分类，以挖 掘市场的评论热点以及舆论导向。作为产品评论观点挖掘的核心内容，对评论文本进行情 感分类一直是热门的研究方向，现有研究工作中也有许多成熟的方法。本节对现有工作的 优点与不足之处均进行了仔细分析，提出了一套完整的在语句粒度上基于产品特征的情感 分类的算法。基于评论的结构化信息以及情感分类的结果，可以对整个观点挖掘的结果进 行归纳，生成情感文摘。

本节内容如下：首先，介绍了评论信息分类算法的处理细节；其次，介绍了观点挖掘结果 归纳的方法；第三，在完成了所有观点挖掘算法之后，介绍一套中文产品评论观点挖掘系统； 第四，对其进行了简要介绍；最后，对本部分内容进行了总结。

**10.3.1** **基于词典与语言规则进行情感分类**

情感分类主流的研究工作主要有两类方法：基于词典的无监督分类方法；基于属性选择 和机器学习的监督分类方法。基于监督学习的分类方法通常在篇章粒度上对产品评论进行 分类，而由于单个语句信息含量少，且无法处理上下文关系，因此基于监督学习的方法无法 在句子粒度上对评论文本进行情感分类，从而会忽略与产品具体特征相关联的情感信息，故 实际应用价值不大。本部分基于产品特征进行观点挖掘的工作，采用了基于词典的无监督 分类方法对评论文本进行了情感分类。本小节将详细介绍分类算法的具体细节，以及词典 等输入语料的来源，并提出了一种判断与上下文相关情感词的情感倾向的方法，然后分析了 整个情感分析算法的计算开销。

**1.** **在句子粒度对评论进行情感分类**

基于词典的情感词抽取方法是情感分析的重要方法之一，该方法主要通过句子中的情 感词以及情感短语来判断该句的情感倾向。本节提出的算法在语句粒度上对评论进行情感

① <http://svmlight.joachims.org/>

分类，如果该句含有产品特征，则该句的情感倾向即为针对该产品特征的情感倾向。算法首 先通过情感元素判断语句的基本情感倾向，再加入用户的语言表述习俗进一步调整语句的 情感倾向。例10.6、例10.7、例10.8、例10.9共描述了“外观”“价格”“屏幕”“内存”四个产 品特征，后续将以这几个例句说明本文情感分类算法的工作流程。

**【例10.6】** 这款手机外观很时尚[+1],但价格太高[-1]。

**【例10** **.7】** 屏幕非常细腻[+1],内存也很大[+1]。

**【例10.8】** 价格很低[+1],但是屏幕太小[-1]。

**【例10.9】** 屏幕稍微发黄[- 1],但是价格低[+1]。

算法首先定位评论中的所有情感元素，主要是情感词语以及短语，通过评论中的情感词 语能初步确定部分语句的情感色彩，例如“时尚”“细腻”。通过这一步，可以确定“这款手机 外观很时尚”,“屏幕非常细腻”的情感倾向为褒义。然后上述4个例子中的其他语句情感色 彩尚未确定，主要原因是其中包含许多未出现在情感词典之中的情感词。这类词语的情感 色彩随着上下文语境以及所修饰的产品特征动态变化，需要特殊处理。因此，算法加入了一 些常用的语言习俗规则，进一步推断其余评论的情感色彩。

(1)否定：句子中的否定词或短语会使该句的情感倾向发生反转。在处理包含否定词 或短语的语句时，有以下几条规则需要纳入情感分类算法中：(1)否定+褒义 → 贬义；(2)否 定+贬义 → 褒义；(3)否定+中性 → 贬义。上述三条规则中值得一提的是，如果语句中本身 没有情感元素，但是出现了表示否定的词或短语，则该句的情感倾向绝大多数时候是贬义 的。例如，在手机评论中，“这款手机没有 GPS”, 该句陈述了一个事实，但是加上了否定前 缀却隐含了一个贬义的情感倾向。当句子中包含多重否定时，对每一层否定都套用上述规 则，即可推断出该句的情感色彩。

(2)转折：句子中出现表达转折的词或短语(例如，“但是”“可是""却”等)时，该句的情 感倾向通常和上一句相反。应用这一条规则之后，在例10.6中，“但价格太高”的情感倾向 与上一句相反，为贬义。而与该规则对应的是，如果一条语句表达了一定的情感色彩，但是 该句中并不含任何转折词或短语，则该句的情感色彩通常和上一句相同。在例10.7中，第 二句——“内存也很大”——虽不含明显的情感词，但是，“小”是和上下文语境以及产品特征 相关联的情感词，说明该句是有一定的情感色彩的，由于该句并未含转折词，则可推断其情 感倾向与上一句相同，为褒义。应用本条规则，如果知道了后一条语句的情感色彩，同样可 以推断出前一条语句的情感色彩。

(3)程度：句子中包含表程度/强度的词或短语时，该句通常含有一定的情感色彩。在 例10.9中，“屏幕稍微发黄”,虽然该句并不包含任何情感词，依然可以由“稍微”推断出该句 表达了一定的情感倾向。而例10.6、例10.7、例10.8的第二句不仅可以从与上下文相关的 情感词推断出它们包含一定的情感色彩，也可以从表达程度的词语(“太”“很”)推断出其含 有一定的情感色彩。

通过以上几条语言习俗规则，可以基本推断出评论中每条语句的情感倾向。然而，以上 仅仅是几条概念性的规则，在实际评论中可能以各种不同的形式出现，因此在实际应用系统 中需要灵活处理各种情况。而更为特殊的是，即便某条语句满足的上述规则，该句也并不是 一定包含相应的情感倾向，这也使得基于词典的情感分类越发复杂。随着研究的深入，以上 规则也可以做适当的调整，并加入新的针对特殊情况进行处理的规则，以提高分类的准

确率。

在情感词典的基础上，将以上规则应用于例10.6、例10.7、例10.8、例10.9,可以推断 出例10.6和10.7中所有语句的情感色彩，然而，例10.8和10.9依然无法处理，因为以上

两句都不含明显的情感词，需要从上下文推断出其情感色彩。

**2.** **相关中文词典构建**

基于词典进行情感分类，需要以完备的情感词库为基础。而当前中文情感词典的情感 词数量的局限性，往往制约了情感分类的效果。在中文情感词典方面已经有许多相关研究 工作，也有许多研究机构总结了各自的情感词库，然而均不够完备。我们在前人工作的基础 上，综合了已有的情感词典、语言学情感研究成果等资源构建了相对完整的情感词典。主要 筛选了台湾大学自然语言处理实验室总结的中文情感词典①,清华大学自然语言处理组的 中文褒贬义词词典②,以及知网(HowNet) 发布的情感词典与评价词典③。通过对上述情感 词典进行筛选与补充，构建了一个相对完备的中文褒贬义词词典，有效地提高了情感分类准 确率。除此之外，还整理了三个与10.3.1节提出的语言习俗规则相关的中文词典，分别是 否定词典、转折词典与程度词典。以上词典均应用到了本节的情感分类算法中。

**3.** **为产品特征构建上下文相关情感词典**

虽然大部分情感词都能直接通过情感词典进行抽取和情感倾向判断，然而评论中却经 常存在少数情感随着不同领域和所描述的特征动态变化的词语，这些词语往往会给情感分 类过程带来很大的干扰。例如，“这个酒店周围噪声很大”“这款手机屏幕很大”,在这两个句 子中，“大”针对不同的描述对象表露了不同的情感色彩。因此，这类情感色彩与上下文相关 的情感词无法直接放在普通褒贬义情感词典中，它们严重制约了情感分类算法的准确率，故 需要在前述情感分类算法的基础上，对这类上下文相关情感词进行特殊处理。虽然已有研 究者提出了通过一些特殊句式在英文语料库中进行统计继而判断这些词语情感倾向的方 法，但是，由于这些特殊句式往往比较少，而且很难定位哪些是情感倾向动态变化的词语，所 以这方面研究仍然是一大难点。

在10.3节介绍的情感分类算法中，针对例10.6进行分析，第一句通过“时尚”一词推断 出其为褒义，第二句由于包含转折词“但是”,因此为贬义。在该句中，“高”是一个上下文相 关的情感词，从而可以推断出其在该句中针对“价格”表达了一个贬义的情感色彩。同理，在 例10.7中，“大”针对“内存”表达了一个褒义的情感色彩。以上结果可以表示为：价格+高 →贬义；内存十大→褒义。由于“高”和“低”,“大”和“小”互为反义词，因此可得：价格+低→ 褒义；内存+小 →贬义。而将此推断出的结论用于例10.8,则第一句“价格很低”为褒义，由

“但”可以推断出第二句“但是屏幕太小”为贬义，从而可得：屏幕+小 →贬义；屏幕+大 →褒

义。同理，可以推断出例10.9第一句为贬义，第二句为褒义。

本节提出的情感分类基础算法，仅仅能对例10.6、例10.7进行情感分类，而以上推理 过程则从例10.1、例10.2的情感分类结果中推断出了例10.8、例10.9的情感倾向。简言

之，在不同领域中，与上下文相关的情感词针对不同的产品特征所表达的情感倾向是动态变

① <http://nlg18.csie.ntu.edu.tw:8080/lwku/pub1.html>

② <http://nlp.csai.tsinghua.edu.cn/site2/>

③ <http://www.keenage.com/html/e>\_index.html

大数据技术及行业应用



化的。当前并不存在一个完备的先验知识库来描述这些信息，然而，这类信息却可以从海量 的产品评论中通过上下文推断出来。因此，本部分提出了一种新的推断上下文相关情感词 的情感色彩的方法，通过对语料库进行统计，可以为每个产品特征建立一个情感极性随着上 下文动态变化的词典，例如(大—小，高—低，厚—薄),再通过对评论的上下文语境分析，推 断出这些词语在该领域中针对某个对象特征的情感倾向。

一些特殊情况在该算法中需要做相应的处理，在一款产品的所有评论中，同一个词针对 同一个产品特征，也可能表露出不同的情感倾向。例如， 一个消费者可能很喜欢“大”屏幕， 而另一个消费者却可能觉得屏幕太“大”会不方便。因此，在本文的算法中，为每一个特征构 建上下文相关情感词典时，采用了“遵循大多数人的意见”的策略。如果同一个词针对同一 个产品特征在不同评论中有不同的情感倾向，则对整个评论语料进行统计，最后采用多数人 的意见作为该词语针对该产品特征的情感倾向。该算法的另一个难点是如何定位上下文相 关的情感词。幸运的是，在中文领域，该类词语实际上并不多，因此本章中人工整理了一个 中文领域的比较全面的与上下文相关的情感词典。

通过上述介绍的算法推理流程可以发现，在算法第一轮进行时，“价格”的上下文相关情 感词典里还没有任何信息，“高”针对“价格”的情感倾向是未知的，因此在例10.8中，是无法 推断出“小”针对“屏幕”的情感色彩的。如果将上述介绍的算法重复执行一次，此时“价格” 的上下文相关情感词典中，“高”为贬义，“低”为褒义，通过该词典便可以推断出例10.8中所 有语句的情感色彩。因此，本节中构建上下文相关的情感词典的过程与情感分类过程需要 迭代进行，情感分类时针对每个特征可以用到上下文相关情感词典中的信息。这两步一直 迭代，直到上下文相关的情感词典不再变化，就能得出更精确的情感分类结果。算法3以伪 代码的形式总结了本节提出的算法的主要步骤。

|  |  |  |
| --- | --- | --- |
| Algorithm3为每个产品特征构建与上下文相关的情感词典 | | |
| 1: | while | true do |
| 2: |  | for each r in Review set do |
| 3: |  | Sentiment detection on r |
| 4: |  | end for |
| 5: |  | change-false |
| 6: |  | for each f in Feature set do |
| 7: |  | Build sentiment lexicon for f |
| 8: |  | if f's lexicon changes then |
| 9: |  | change \*true |
| 10: |  | end if |
| 11: |  | end for |
| 12: |  | if!change then |
| 13: |  | Break |
| 14: |  | end if |
| 15: | end while | |

第10章 大数据技术在文本挖掘和情感分类中的应用

综合以上分析，本节提出的文本情感分类算法的计算开销主体上包含两部分，其一是对 评论文本进行情感分类，其二是为每个特征构建上下文相关情感词典。在基于词典对评论 进行情感分类时，只需顺序扫描所有评论一遍即可，计算开销为 O(MN), 其 中M 为情感词 数量，为评论中语句数量。而在为每个特征构建上下文相关情感词典时，也只需对语料库顺 序扫描一次进行统计，计算开销为 O(N) 。 故整个情感分析算法每轮迭代的时间开销均为 O(MN)。 而算法在进行情感分类时， 一个原则是优先考虑情感词典，其次考虑语言习俗规 则。即如果该句含有情感元素，如情感词和短语，与特征和上下文相关的情感词等，则优先 通过情感元素判断该句情感倾向；如果该句不包含情感元素，则通过上下文推断其情感色 彩。在采用情感词典优先的原则下进行情感分析，算法经过每轮迭代，与特征和上下文相关 的情感词典都得到进一步完善，并将被用于下一轮的情感分类。因此，每一轮迭代之后，上 下文相关的情感词典都会趋于完善，而不会呈现波动状态，故算法必定会一直收敛，并在数 次迭代后结束。

**10.3.2** **观点挖掘结果归纳**

观点挖掘的最终目的是对产品的全部评论进行分析总结，并从中提炼出主流的情感倾 向和评论热点。因此，在完成情感分析工作之后，需要将情感分析结果进行归纳，以更直观、 更简洁的方式向用户展现，比较主流的一种方式是生成产品评论的情感文摘。由于本研究 中基于产品特征对评论文本进行了观点挖掘，因此基于产品特征为每个产品的所有评论生 成了情感文摘。情感文摘主要包含两方面的内容。

(1)产品特征的情感总结：在观点挖掘算法中，抽取了评论语句所描述的产品特征，并 对所有评论进行了情感分类。因此，针对每一个产品特征，可以统计所有用户针对该特征的 所有评论中，褒义评价和贬义评价的百分比，从而直观地反映出该产品在该特征上的市场口 碑如何。

(2)产品特征的评论热点：评论热点是指比较有代表性的评论。给出每个产品特征的 褒义评论和贬义评论的百分比，仅仅能反映出该特征的普遍评价情况，而用户却并不了解为 何该产品特征会收到如此的评价，因此需要挑选出部分有代表性的评论语句加入情感摘要， 以便用户能从一定程度了解到该特征的优势，以及为人诟病的不足之处在哪里。

通过10.3.1节提出的评论情感分类算法，能够得到每个特征的褒义评价和贬义评价百 分比，因此只需再提取部分有代表性的评论热点，即可生成简洁的情感摘要。在从每个特征 的评论中提取评论热点时，首先对评论在篇章粒度上进行了筛选，去除了内容较短，或重复 内容过多的评论，因为这类评论的作者往往评价态度不够认真，因此评论内容的可信度并不 是很高。其次，对该特征的所有褒义和贬义评论按照情感得分进行了排序，并依据褒义和贬 义评论各自的百分比，选出了部分情感色彩强烈的语句作为该产品特征的情感摘要。

图10-4展示了所开发的观点挖掘系统中，以图形化的方式所展现的基于产品特征的评 论情感分类结果。从图中可以直观地看出该产品不同特征的市场口碑，非常方便潜在的消 费者更直观地了解此产品的优势与不足之处，而将不同产品的评论情感分类结果图合并也 能非常直观地将两款产品在不同特征维度上进行对比。图中每个特征的褒义和贬义评论也 都各自对应了少量有代表性的热点评论作为其摘要。

基于前几节中所提出的观点挖掘算法，我们开发了一套针对中文的基于产品特征的互

**大数据技术及行业应用**

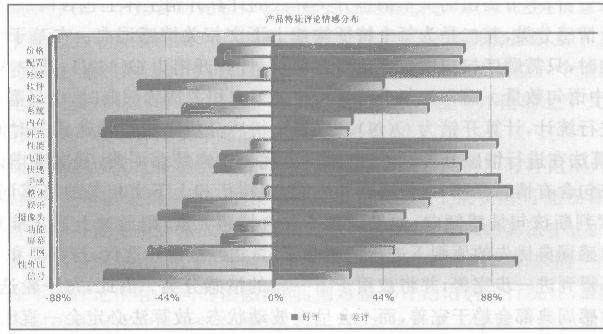
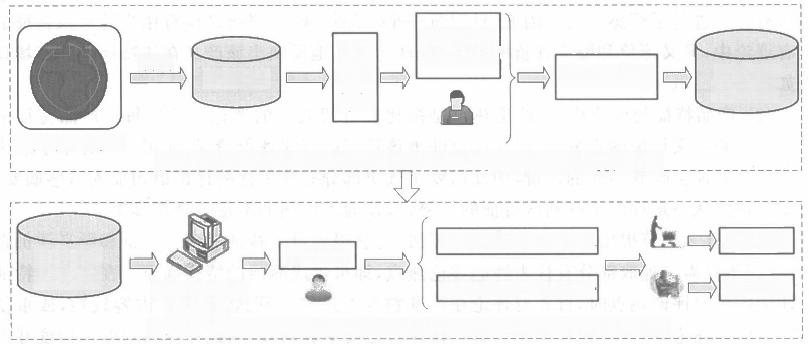
132%

图10-4 系 统 的 具 体 实 现

联网评论观点挖掘系统，流程如图10-5所示。该系统对互联网产品评论进行观点挖掘主要 包含两个模块：首先是系统的准备阶段；其次是系统的使用阶段。在系统准备阶段，主要需 要完成后台的产品评论舆情分析。首先，系统针对某些特定领域，抓取相应产品评论，然后 对不同产品的评论进行产品特征抽取以及观点挖掘，形成舆情分析结果。在系统的使用阶 段，用户可以查询自己所关注的产品的评论分析结果，发现产品的优点与不足，并对比不同 产品的优势与劣势。该系统的分析结果不仅能给潜在的消费者的购物决策提供参考，还能 给产品的生产厂商提供改进建议。



产品特征相

似概念聚类

观点挖掘算法

互联网产品评论

后台观点挖掘算法的目的是抓取并分析某个领域比较全面的产品评论信息

产品市场反映：优点与不足

用户查询

产品评论舆情

分析系统

前台观点挖掘系统目的是为了给不同类型用户展现不同层次的观点挖掘结果

购买决策

改进产品

不同产品对比：优势与劣势

产 品 特征 抽取 算法

评论舆情 分析结果

评论舆情 分析结果

评论数据

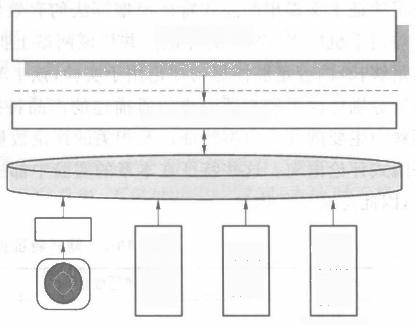
爬虫

图10 - 5 观 点 挖 掘 系 统 详 细 实 施 流 程 图

所开发的情感分析系统的具体实现架构如图10-6所示。整个系统自顶向下可以分成 三大主要层次：顶层是用户界面模块；中间是数据库接口模块；底层是各个算法功能模块。 用户界面模块主要是给观点挖掘系统的使用者提供一个图形化的友好的用户操作界面，以 方便用户浏览自己感兴趣的产品的评论分析结果。数据库接口模块提供了整个系统的数据 库读写接口，方便其他各个不同的功能模块进行数据的 I/O 操作。底层算法模块主要包 括：(1)爬虫模块。用于从互联网上抓取产品评论数据；(2)产品特征抽取模块。用于从评论

中抽取频繁被描述的产品特征；(3)观点 挖掘模块。用于基于产品特征对评论进 行细粒度的情感分类；(4)情感文摘生成 模块。根据情感分类结果生成热点情感 文摘。

针对评论观点挖掘中的核心问题，提 出了 一 套完整的基于词典的评论情感分 类算法，能在句子粒度上基于产品特征对 评论文本进行情感分类，并且还提出了上 下文相关情感词的情感倾向判断方法。 整合的情感分类算法不仅满足了细粒度 的基于产品特征进行情感分类的要求，还 解决了与上下文相关的词语情感倾向判

图形用户界面

前台界面模块接口

系统数据库

|  |  |  |
| --- | --- | --- |
| 产品  特征  抽取  算法 | 观点  挖掘  算法 | 情感 文摘 生成 算法  产品评价 |

图10-6 观点挖掘系统的实现架构图

断的难点，有效提高了情感分类算法的准确率。在上述算法基础上，开发了一套完整的产品 评论观点挖掘系统，将文中提出的算法真正应用到实际系统中。



**10.4** **算法评估结果与分析**

10.3节针对基于产品特征进行观点挖掘的问题，介绍了产品评论的结构化信息抽取方 法，并重点介绍了两种不同的隐式特征提取方法，以期能使观点挖掘系统获得更高的召回 率。在产品评论的结构化信息基础上，本节重点介绍了基于产品特征在语句粒度上对评论 文本进行情感分类的算法，并提出了一种分析与上下文相关情感词的情感倾向的方法。在 本节中将对以上算法进行详细评估，以验证其有效性。

本节主要内容：首先，将介绍评论结构化信息提取工作中，隐式特征抽取算法的实验结 果；其次，介绍了在篇章粒度上情感分类算法的实验结果；再次，介绍了在语句粒度上情感分 类算法的实验结果；最后，对本章内容进行了总结。

**10.4.1** **隐式特征抽取实验结果及分析**

本节将主要介绍在产品评论结构化信息抽取中，所提出的两种不同的隐式特征抽取算 法的实验结果，主要内容包括实验数据集介绍，基于搭配提取方法所获得的关联规则结果分 析，以及基于关联规则和基于机器学习算法抽取隐式特征的实验结果。实验结果验证了前 几部分中提出的隐式特征抽取算法的有效性。

**1.** **实验数据集**

当前中文评论观点挖掘领域的公开实验数据集还稍显匮乏，大部分数据集仅仅标注了 评论整体的褒贬义情感倾向，主要用于验证基于篇章粒度的情感分类算法效果。而就当前 所了解到的情况，当前中文领域并没有公开的基于产品特征进行观点挖掘的权威数据集。

为了验证上文提出的隐式特征挖掘算法的有效性，从国内著名电子商务网站上①抓取了一 款热门手机中兴 V880 的评论。共从该网站上抓取了4545条评论，并从中选取了前1343 的足够长、内容足够详尽的评论用于实验，该1343条评论共包含10148条评论语句。人工 协助分别标注了所有评论语句所描述的产品特征并互相进行了验证。表10-2描述了评论 语料中主要的几个典型特征以及相关的评论数量， 一共包含了2775条显示评论语句和993 条隐式评论语句。这些特征在本章的实验中都进行了评估，其它产品特征由于显式评论较 少，因此忽略了其内容。

**表10-2** **隐式特征抽取实验数据集**

|  |  |  |  |
| --- | --- | --- | --- |
| 产品特征 | 显式评论数量 | 隐式评论数量 | 评论总数 |
| 屏幕 | 810 | 183 | 993 |
| 质量 | 135 | 63 | 198 |
| 电池 | 343 | 127 | 470 |
| 价格 | 950 | 418 | 1368 |
| 外观 | 164 | 128 | 292 |
| 软件 | 329 | 51 | 380 |
| 摄像头 | 44 | 23 | 67 |

**2.** **基于搭配提取的关联规则挖掘结果与讨论**

在获得了分词、词性标注以及每个特征簇的显式评论语句集合之后，本部分用5种不同 的搭配提取方法挖掘了特征指示器与对应的产品特征之间的关联规则。当从显式评论语句 集合中筛选候选特征指示器时，所有的特征指示器都通过词性与最少出现次数进行了剪枝， 在部分的实验中，特征指示器至少需要出现两次。表10-2展示了“价格”“外观”和“电池”三 个产品特征的部分关联规则挖掘结果。由于篇幅所限，表中仅仅列出了这三个特征在每种 方法中排名靠前的8条规则。

当通过词性对候选规则进行剪枝时，仅仅保留了名词(nouns,n) 、 动 词(verbs,v) 和形容 词(adjectives,a)作为候选词。不同的剪枝策略往往会得到迥异的实验结果。在前人的相 关研究中，大多数工作仅仅选取了情感词作为特征指示器，通常是形容词。然而，从表10-2 中可以发现， 一些合乎常理的规则却不是形容词。在“价格”的挖掘结果中，“跌”是动词，也 是一条合理的规则。例如，在例10.10中，第二条语句显然是针对价格的评论，尽管它陈述 了一个事实，但是却明显地表达了一个贬义的情感倾向。

**【例10.10】** 很漂亮，但是我刚买的第二天就跌了20%。真倒霉!

在表10-3 中，下划线的词是一些不太合理的规则。有些错误是由于分词或词性标注程 序的错误引起的。例如，“超”在“价格”的评论中多数时候是副词，但是词性标注结果却是动 词。从表中还可以发现，当使用frequency\*PMI 时，这三个特征的前6条规则均是合理的。 而 t 检验和x² 检验的结果虽然与frequency\*PMI 的结果比较接近，但是这两种方法抽取的 前6条规则中仍然有不合理的。从表中还可以发现，frequency 和 PMI 两种方法的结果均

① <http://www.360buy.com>

第10章 大数据技术在文本挖掘和情感分类中的应用

不如其他三种方法。尽管在这两种方法的结果中错误并不多，但是它们却遗漏了很多重要 的规则，例如，“便宜”→ “价格”。从表10-3中还可以发现，在“外观”的所有规则结果中，仅 仅出现了一个错误的结果。这个现象不难解释，针对“外观”的绝大多数规则均是形容词，并 且也都是情感词。不同于其他名词或动词，这些含有情感色彩的形容词通常均是用来形容 “外观”的，用法比较单一。而与形容词不同的是，动词和名词往往有许多用法。例如，“能 力”既可以用来形容电池的续航能力，也可能用来形容 CPU 的处理能力。因此，“能力”并 不是一条合理的描述“电池”的规则。从中也可以看出，不仅处理了情感词，也处理了其他词 语，这使得该工作比前人的研究难度更高。

**表10-3** **“价格”“外观”和“电池”的部分关联规则挖掘结果(仅仅显示前八条规则)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 产品特征 | frequency | | PMI | | frequency\*PMI | | t test | | x²test | |
| 规则 | 值 | 规则 | 值 | 规则 | 值 | 规则 | 值 | 规则 | 值 |
| 价格 | 对得起 | 33 | 足 | 3.86 | 便宜 | 306.21 | 便宜 | 8.59 | 便宜 | 866.64 |
| 实惠 | 31 | 对得起 | 3.64 | 对得起 | 119.99 | 对得起 | 5.28 | 对得起 | 382.98 |
| 贵 | 10 | 对得起 | 3.63 | 实惠 | 108.41 | 实惠 | 5.07 | 实惠 | 320.97 |
| 对得起 | 8 | 实惠 | 3.50 | 值 | 70.59 | 值 | 4.18 | 值 | 178.52 |
| 具 | 5 | 跌 | 3.45 | 贵 | 29.64 | 贵 | 2.75 | 对得起 | 93.11 |
| 划算 | 5 | 划算 | 3.45 | 对得起 | 29.13 | 对得起 | 2.60 | 贵 | 65.39 |
| 无敌 | 5 | 合理 | 3.45 | 划算 | 17.25 | 超 | 2.09 | 足 | 55.65 |
| 足 | 4 | 优势 | 3.45 | 具 | 15.93 | 划算 | 2.03 | 划算 | 49.65 |
| 外观 | 大气 | 13 | 时尚 | 5.99 | 大气 | 73.15 | 大气 | 3.53 | 大气 | 628.65 |
| 时尚 | 10 | 酷 | 5.83 | 漂亮 | 65.02 | 漂亮 | 3.38 | 时尚 | 625.25 |
| 好看 | 9 | 轻薄 | 5.83 | 时尚 | 59.85 | 时尚 | 3.11 | 漂亮 | 499.01 |
| 大方 | 4 | 大方 | 5.83 | 薄 | 55.70 | 薄 | 3.10 | 好看 | 452.43 |
| 酷 | 3 | 好看 | 5.68 | 好看 | 51.13 | 好看 | 2.94 | 大方 | 224.18 |
| 轻薄 | 3 | 大气 | 5.63 | 大方 | 23.33 | 大方 | 1.96 | 酷 | 168.12 |
| 瑕疵 | 2 | 薄 | 5.57 | 酷 | 17.50 | 酷 | 1.70 | 轻薄 | 168.12 |
| 美观 | 2 | 漂亮 | 5.42 | 轻薄 | 17.50 | 轻薄 | 1.70 | 美观 | 112.07 |
| 电池 | 一块 | 19 | 耗电 | 5.32 | 耐用 | 296.31 | 耐用 | 7.54 | 耐用 | 1716.14 |
| 容量 | 15 | 容量 | 4.90 | 一块 | 91.77 | 一块 | 4.21 | 一块 | 521.51 |
| 续航 | 11 | 续航 | 4.90 | 一天 | 74.50 | 一天 | 3.84 | 容量 | 434.75 |
| 能力 | 9 | 电量 | 4.90 | 容量 | 73.56 | 容量 | 3.74 | 续航 | 318.70 |
| 短 | 7 | 耐用 | 4.86 | 续航 | 53.94 | 续航 | 3.21 | 能力 | 210.08 |
| 待机 | 7 | 一块 | 4.83 | 时间 | 51.49 | 能力 | 2.88 | 耗电 | 157.08 |
| 配 | 5 | 一天 | 4.65 | 能力 | 41.53 | 短 | 2.53 | 短 | 154.61 |
| 耗电 | 4 | 能力 | 4.61 | 短 | 31.79 | 待机 | 2.52 | 待机 | 137.78 |

在本节提出的算法中，特征指示器不仅仅包含了单个的词语，还包括了在评论中经常出 现的二维频繁项。例如，“充满电只能用一天”,该句中“一天”是由数词与量词组成的二维频 繁项，并且经常被用来描述电池的续航能力，是针对“电池”的一条合理的规则。表10-2 中 的部分规则也验证了算法的有效性，例如“一天”“一块”都是针对“电池”的合理规则。在抽 取二维频繁项作为候选特征指示器时，词性被限定为名词、动词、形容词、代词(pronoun,

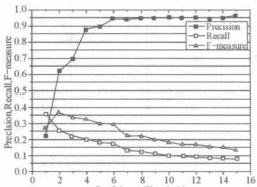
p)、数 词(numeral,m) 和量词(quantifier,q)。

**3.** **基于关联规则的隐式特征抽取结果与讨论**

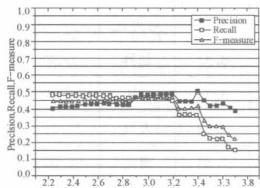
在本章提出的关联规则挖掘策略中，所有的候选规则均通过置信度阈值进行了过滤，低 于阈值的候选规则或者在多个产品特征的候选规则集中同时出现的规则均被移除。高的阈 值能去除低频出现的规则和置信度低的规则，可以提高算法准确率，但同时也降低了召回 率。因此，只有设定合适的阈值才能得到最佳的 F 值。在文部分的实验中主要评估了经常 用于信息检索领域的传统的F 值，也被称作F₁measure①, 此处准确率和召回率是同等权重 的。计算公式如下：

 (10-6)

在实际应用中，合适的置信度阈值需要通过实验进行调整。图10-7展示了利用5种不 同的搭配提取算法在本部分的实验数据集上挖掘隐式特征的实验结果。

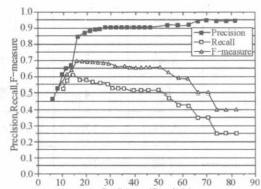
Confidence Threshold

(a)frequency

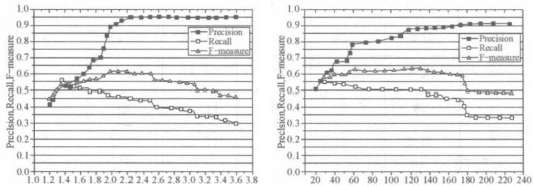


Confidence Theshold

(b)PMI

Confidence Threshold

(c)frequency\*PMI



Confidence Threshold Confidence Threshold

(d)r test (e)x test

图10-7 不同搭配提取算法的实验结果

从图10-7中可以明显看出算法准确率随着阈值升高而升高，同时召回率随之降低。在 这5种方法中，使用 frequency\*PMI 时并且当阈值在20左右时，算法获得了最高的 F 值 — — 69 . 23%。t 检验和x² 检验两种方法的最好结果比较接近，但是均没有 frequency\*

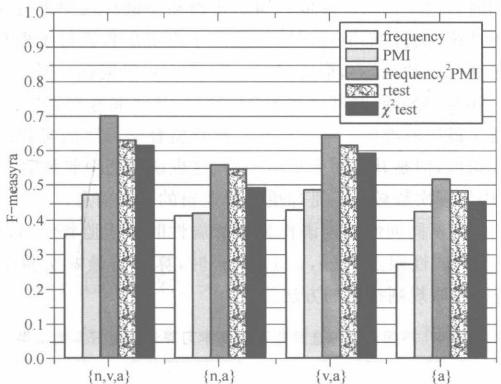
PMI 高。而使用PMI 挖掘关联规则时，算法性能却不如前三种方法，此结果正如本部分在

① <http://en.wikipedia.org/wiki/F-score>

第10章 大数据技术在文本挖掘和情感分类中的应用

10.2.2节中所阐述的，PMI 并不是一个好的衡量依赖性的指标。当使用 PMI 从文本中提 取搭配时，其他条件相同的情况下，出现次数较少的特征指示器反而会比出现次数较高的特 征指示器获得更高的得分。然而，在本部分的应用场景中，出现次数越多更能证明该特征指 示器经常用来描述对应的产品特征。正因如此 frequency\*PMI 的实验结果要明显优于 PMI 。与 Hai 等人的实验结果不同的是，利用frequency 提取关联规则的实验结果是5种方 法 中F 值最低的。最重要的一个原因是 Hai 等人在其工作中仅仅处理了情感词，并且在评 估实验结果时也仅仅评估了包含情感词的评论语句的情况。从10.4.1节中介绍的“外观” 的关联规则提取情况可知，仅仅处理情感词是比较容易获得高的准确率和召回率的。而本 部分的研究工作致力于提取所有隐式评论语句，并且在实验中既评估了直接包含情感观点 的语句，也评估了只包含事实的语句，这也使本项工作更难获得高的召回率。也因为这个原 因，利用 frequency 提取关联规则时，虽然召回率很低，但是其准确率非常高。

当从显式评论语句集合中筛选候选特征指示器时，所有特征指示器均通过词性进行了 剪枝。图10-8展示了使用不用的词性过滤规则时，算法的最好结果。从图中可以看出，当 候选词的词性限定为名词、动词和形容词时，即{n,v,a}, 使 用frequency\*PMI 方法获得了 最高的 F 值。仅保留{v,a} 的剪枝策略的实验结果和{n,v,a} 非常接近，表明了形容词和动 词在评论语句中出现时，通常会用来暗示某些产品特征。然而，名词也同样可以用来暗示产 品特征，因为从实验结果可以发现，保留{n,a} 的剪枝策略的实验结果要优于仅保留{a} 的 实验结果。综上分析可知，相比于前人工作，本部分所提出的算法能在所有评论整体上获得 高的召回率。

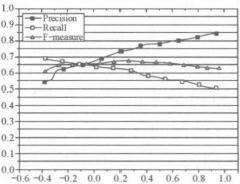
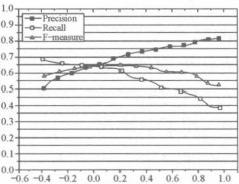


POS

图10-8 使用不同词性过滤规则时不同算法的最好实验结果

**4.** **基于监督学习的隐式特征抽取结果与讨论**

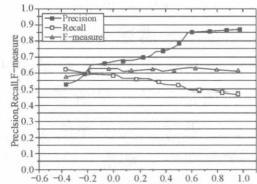
基于监督学习的隐式特征抽取算法为每个产品特征训练了一个分类器，再用这些分类 器预测每个隐式评论语句所描述的特征。在为每个特征训练分类器时，训练数据集为显式 评论语句，其中正例为当前产品特征的显式评论语句，负例为其他所有特征的显式评论语 句。由于正例与负例数据严重不均衡，本文中使用了 SVM-light 中的支持向量回归机

(Support Vector Regression,SVR) 来对隐式评论进行分类，所有参数都使用默认值。 图10-9展示了使用不用的词性筛选策略下监督学习算法的实验结果。

nRanF-mcsuc

cionRecall.F-measure

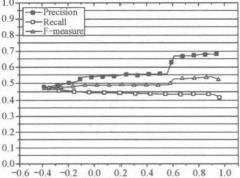
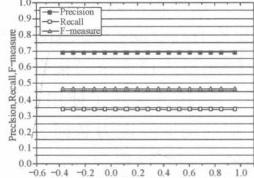
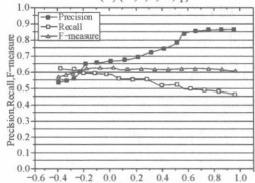
PeebionRealr-mesn

Confidence Threshold

Confidence Threshold

Confidence Threshold

(a)所有实词 (b){n,v,a,m,q} (c){n,v,a}

Confidence Threshold Confidence Threshold Confidence Thre shold

(d){n,a} (e){v,a} (f){a}

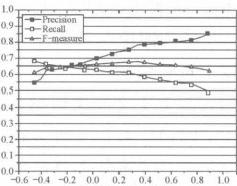
图10-9 不同词性筛选策略下基于监督学习的算法实验结果

实验得到的 SVR 预测结果需要通过置信度阈值划分为正例和负例，并且准确率会随着 阈值升高而升高，同时召回率会相应降低。当从正例和负例评论中进行属性选择生成训练 数据集时，所有用于训练的属性都通过词性以及最小出现次数进行了剪枝。当用于训练的 词语词性限定为{名词(n), 动 词(v), 形 容 词(a), 数 词(m), 量 词(q)} 时，算法在阈值0.25处 获得了最高的值 — — 67 .43%,此时准确率为74 .33%,召回率为61 .70%。当置信度阈值接 近1.0时，算法取得了最好的准确率 — — 85%。尽管监督学习算法实验结果中最高的 F 值 比 frequency\*PMI 稍低，但是其召回率更高一些，这也说明利用监督学习的方法，能够更深 入的挖掘训练语料中出现次数较低但是却更符合逻辑的规则信息。

表10-4展示了使用不同词性的词语作为训练属性时基于监督学习的隐式特征抽取算 法的最好实验结果。当词性限定为{n,v,a,m,q} 时，算法的最好实验结果非常接近利用 frequency\*PMI 进行关联规则挖掘的方法。

**表10-4** **不同词性筛选策略下监督学习算法的最好实验结果**

|  |  |  |  |
| --- | --- | --- | --- |
| 词性 | Precision | Recall | F-measure |
| 所有实词 | 70.12% | 62.20% | 65.92% |
| (n,v,a,m,q) | 74.34% | 61.70% | 67.43% |
| {n,v,a} | 67.75% | 58.40% | 62.72% |
| {n,a} | 74.64% | 51.50% | 60.95% |
| {v,a} | 68.17% | 43.70% | 53.26% |
| {a} | 69.08% | 34.40% | 45.93% |

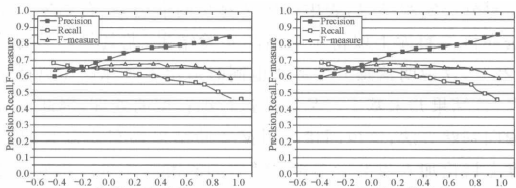
第10章 大数据技术在文本挖掘和情感分类中的应用

喻

唱

在机器学习算法中，属性选择通常是对训练数据进行降维的重要步骤，它可以简化预测 模型的学习过程，并且增大其对于未知数据的泛化性能。因此，如果在本部分提出的基于监 督学习挖掘隐式特征的算法中加入属性选择，排除不相关的或多余的属性，应该能提升算法 的性能。在实验过程中，利用信息增益(information gain)进行了属性选择，当所有用于训 练的词语词性限定为{n,v,a,m,q} 时，实验数据中总的属性数量为1330。图10- 10展示了 选取相关度评分靠前的不同数量的属性作为训练数据集时算法的实验结果。当属性数量为 600时，算法在置信度阈值为0 . 2处获得了最好的 F 值 — — 67 . 91%,此时准确率为75 .

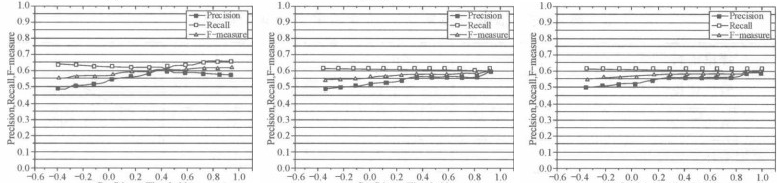
37%,召回率为61.80%。加入属性选择策略后，算法的最好结果比未加入属性选择之前提 高了0 .5%。尽管算法性能经过属性选择之后有所提升，但是提升效果并不显著。在所研 究问题中，产品特征可以通过任意词语暗示，而并不仅仅是形容词，因此算法中的属性数量 并非越多越好，而需要合适的取值。



Confidence Threshold

Confidence Threshold

Confidence Threshold



(a)900 (b)600 (c)300

Confidence Thre shold

Confidence Threshold

Confidence Threshold

(d)150 (e)100 (f)50

图10-10 不同属性数量下基于监督学习的算法实验结果

**10.4.2** **篇章粒度情感分类实验结果及分析**

情感分类算法核心是在语句粒度上基于产品特征对评论进行情感分类。然而，当前中 文领域大多数情感分类研究工作都集中于在篇章粒度上对评论进行情感分类。为了验证本 章提出情感分类算法的有效性与通用性，本小节将介绍用本文的算法对评论在篇章粒度上 进行情感分类的实验结果，包括实验数据集介绍，情感分类的实验结果以及讨论分析。

**1.** **实验数据集**

当前中文领域评论观点挖掘的公开实验数据集并不多，不过仍然有相关的研究。其中， 中科院计算所的谭松波博士所整理的酒店和笔记本计算机的情感分类语料①可以用来进行

① <http://www.searchforum.org.cn/tansongbo/senti>\_corpus.jsp

大数据技术及行业应用

本项实验。他们公布的语料包括10000篇非平衡的酒店评论以及4000篇平衡的笔记本计 算机评论。其中，酒店评论包含7000篇褒义评论和3000篇贬义评论，而笔记本计算机评 论中，褒贬各占二分之一 。本小节将在此数据集上对情感分类的算法进行评估。

**2.** **情感分类实验结果及讨论**

本章所提出的情感分类算法主要是在语句粒度上对评论进行情感分类，分类结果中每 条语句都会获得相应的情感得分，而得分的权重根据语句中程度词的不同而有所不同。为 了在篇章粒度上对评论进行情感打分，只需将每条语句的得分求和，便能得到该评论的整体 情感得分。例如，“这款手机非常漂亮，但是屏幕有点差”,该例子中第一句为褒义，第二句为 贬义，然而第一句中有表示程度增强的词“非常”,在本部分算法中，受程度词加权，该句得分 为1.75,而第二句中由于有表示程度减弱的词“有点”,该句得分为一0.5,因此在本部分算 法中该评论整体得分为1.25,为褒义。表10-5展示了情感分类算法在篇章粒度上对评论进 行情感分类实验结果，表中列出了算法的准确率、召回率与 F 值。从表中的结果可以看出， 算法在该语料库上取得了不错的实验结果，尤其是酒店评论中，褒义评论的准确率达到了 94.22%,这说明情感分类算法也同时适用于对评论在篇章粒度上进行情感分类。需要说明 的是，该结果并未基于产品特征进行细粒度的处理，因此没有加入与上下文相关情感词的情 感倾向判断方法，这也间接表明该实验结果还有非常大的提升空间，10.4.3节将会在语句 粒度上讨论加入与上下文相关情感词的情感倾向判断方法之后，整个情感分类算法的提升 效果。

**表10-5** **情感分类算法在篇章粒度上的实验结果**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 评论语料 | 褒/贬 | Precision | Recall | F-measure |
| 酒店评论 | 褒义 | 94.22% | 76.34% | 84.34% |
| 酒店评论 | 贬义 | 64.40% | 86.37% | 73.79% |
| 笔记本计算机评论 | 褒义 | 82.72% | 86.15% | 84.40% |
| 笔记本计算机评论 | 贬义 | 87.84% | 74.40% | 80.56% |

**10.4.3** **语句粒度情感分类实验结果及分析**

本小节将主要介绍情感分类算法在语句粒度上基于产品特征进行情感分类的实验结 果，包括实验数据集介绍，基于词典进行情感分类的实验结果，以及加入上下文相关情感词 处理之后算法的提升效果。

**1.** **实验数据集**

由于当前中文评论观点挖掘领域在细粒度上进行情感分类的公开实验数据集比较少， 在评估情感分类算法性能时，依然在所介绍的手机评论数据集上进行了评估，并且对实验数 据集进行了进一步扩充标注。共标注了1844篇评论，每条评论语句都标注了其所描述的 产品特征以及情感倾向。整个语料一共包含14565条评论语句，其中包含8432条褒义评 论，4216条贬义评论和3917条中性评论。在所有评论语句中共有1905条句子既包含上 下文相关情感词又描述了至少一个产品特征。而本实验所用到的上下文相关情感词典共包 含50个左右的与上下文相关的情感词语，其中每个词语都整理了对应的同义词与反义词。

第10章 大数据技术在文本挖掘和情感分类中的应用

F-measyra

**2.** **与上下文相关的情感分类实验结果及讨论**

表10-6展示了与上下文相关的情感词的情感分类算法实验结果，该结果仅针对包含上 下文相关情感词的语句，表中列出了算法的准确率、召回率与 F 值。在没有为每个产品特 征构建上下文相关情感词典之前，该部分语句值仅为36.56%,这也表明所提出的上下文相 关词典构建算法非常重要。通过一 次迭代为每个特征建立上下文相关情感词典之后，F 值 升高到了85 . 10%,此结果证明所提出的算法在中文领域非常有效。算法经过第二轮迭代 后 ，F 值进一步提升到了86 .21%,多一轮迭代之后算法效果的提升也证明了前文对迭代算 法分析的正确性。此后实验结果不再发生变化，表明每个特征的上下文相关情感词典已经 趋于稳定。因此，所提出的迭代构建情感词典的方法能在一定程度上提升算法性能，而迭代 次数取决于实验数据。

**表10-6** **与上下文相关的情感分类算法实验结果**

|  |  |  |  |
| --- | --- | --- | --- |
| 迭代次数 | Precision | Recall | F-measure |
| 0 | 26.98% | 76.24% | 36.56% |
| 1 | 84.15% | 86.95% | 85.10% |
| 2 | 85.83% | 87.08% | 86.21% |
| 3 | 85.83% | 87.08% | 86.21% |

图10-11展示了算法经过不同迭代次数之后，在所有语句和在上下文相关语句上的值。 在情感检测阶段，既处理了明显的观点型评论，也处理了包含一定情感色彩的事实性评论， 这也使得算法很难取得非常高的分类效果。从图10-11可以发现，上下文相关情感词的处 理方法使情感分类在所有语句上的整体值从74 . 80%提升到了82 . 06%,提升效果非常显 著。尽管如此，情感分类算法在所有语句上的整体 F 值依然不是非常令人满意，仍然有一 些提升的空间。评论文本中的一些语言表述习俗对情感分类的效果有很大影响，例如虚拟



Iterations

图10-11 不同迭代次数下的 F 值

语气等，这类语句基于情感词典往往会得到错误的情感倾向，从而降低了基于词典的算法的

整体准确率，因此这类语句也需要进一步的处理。

本节对隐式特征抽取算法以及产品评论情感分类算法进行了详细的实验评估。隐式特 征抽取作为评论结构化信息抽取过程中被前人忽略的问题，它的效率高低直接影响了评论 观点挖掘的召回率。首先，详细评估了两大类隐式特征抽取算法的实验结果，证实了所提出 的基于关联规则挖掘隐式特征的算法的有效性。其次，针对评论观点挖掘的核心问题，详细 评估了前文提出的基于词典在语句粒度进行情感分类的算法准确率，并对所提出的上下文 相关情感词的情感倾向判断方法进行了评测。实验结果表明情感分类算法在细粒度上对评 论文本进行情感分类的结果非常优异，并且有效解决了与上下文相关的情感词的情感倾向 判断问题，使算法整体的准确率有了大幅提升。



**本** **章** **小** **结**

基于大数据思维对文本进行挖掘，进而采用相关机器学习的算法挖掘文本中蕴含的丰 富情感和观点信息，这些工作对于产品和服务的提供者具有重要的决策指导作用。同时帮 助企业在海量的信息中精准的定位用户的个性化需求，并把握用户的痛点，对于新产品的推 出和营销等活动具有至关重要的作用。

**参** **考** **文** **献**

[1]Pang B,Lee L.Opinion mining and sentiment analysis.Foundations and Trends in

Information Retrieval,2008,2(1-2):1-135.

[2] Liu B.Sentiment Analysis and Subjectivity.Handbook of Natural Language Pro-

cessing,2010,(1):1-38.

[3] Han J,Kamber M.Data mining:Concepts and techniques.Morgan Kaufmann,2006. [4] Mitchell T.Machine Learning.New York:The McGraw-Hill Companies,Inc,1997.

[5] Manning C,Schütze H,MITCogNet.Foundations of statistical natural language

processing.MIT Press,1999.

[6] Hu M,Liu B.Mining and summarizing customer reviews.2004:168-177.

[7] Hu M,Liu B.Mining opinion features in customer reviews.2004:755-760.

[8] Liu B,Hu M,Cheng J.Opinion observer:analyzing and comparing opinions on the Web.2005:342-351.

[9] Ding X,Liu B,Yu P S.A holistic lexicon-based approach to opinion mining.2008:

231-240.

[10] Zhai Z,Liu B,Xu H,et al.Grouping product features using semi-supervised learning with

soft-constraints.2010:1272-1280.

[11] Popescu A,Etzioni O.Extracting product features and opinions from reviews.

2005:339-346.

[12] Su Q,Xiang K,Wang H,et al.Using Pointwise Mutual Information to Identify

Implicit Fea-tures in Customer Reviews.2006,4285:22-30.

[13] Su Q,Xu X,Guo H,et al.Hidden sentiment association in chinese web opinion

mining.2008:959-968.

[14] Hai Z,Chang K,Kim J.Implicit Feature Identification via Co-occurrence Associa-

tion Rule Mining.2011,6608:393-404.

[15] Kim S,Hovy E.Determining the sentiment of opinions.2004.1367.

[16] Choi Y,Cardie C,Riloff E,et al.Identifying sources of opinions with conditional

random fields and extraction patterns.2005.355-362.

[17] Kim S,Hovy E.Identifying and analyzing judgment opinions.2006.200-207.

[18] Kim S,Hovy E.Automatic detection of opinion bearing words and sentences.

2005.61-66.

[19] Zhu Y,Min J,Zhou Y,et al.Semantic orientation computing based on HowNet.

Journal of Chinese Information Processing,2006,1:14-20.

[20] Ku L,Liang Y,Chen H.Opinion extraction,summarization and tracking in news

and blog corpora.2006,(2001).

[21] Fu G,Wang 10.Chinese sentence-level sentiment classification based on fuzzy

sets.2010.312-319.

[22] Hatzivassiloglou V,McKeown K.Predicting the semantic orientation of adjectives.

1997.174-181.

[23] Turney P.Thumbs up or thumbs down?semantic orientation applied to unsuper-

vised classifi-cation of reviews.2002.417-424.

[24] Hatzivassiloglou V,Wiebe J.Effects of adjective orientation and gradability on

sentence sub-jectivity.2000.299-305.

[25] Yu H,Hatzivassiloglou V.Towards answering opinion questions:Separating facts

from opin-ions and identifying the polarity of opinion sentences.2003.129-136.

[26] Yao T,Peng S.A study of the classification approach for Chinese subjective and

objective texts.Proc.of the NCIRCS,2007,2007:117-123.

[27] Carbonell J.Subjective Understanding:Computer Models of Belief Systems.1979.

[28] Wilks Y,Bien J.Beliefs,points of view,and multiple environments.Cognitive

Science,1983,7(2):95-119.

[29] Das S,Chen M.Yahoo!for Amazon:Extracting market sentiment from stock

message boards.2001,35:43.

[30] Dini L,Mazzini G.Opinion classification through information extraction.2002.

299-310.

[31] Morinaga S,Yamanishi K,Tateishi K,et al.Mining product reputations on the

web. 2002.341-349.

[32] Cardie C,Wiebe J,Wilson T,et al.Combining low-level and summary representa-

tions of opinions for multi-perspective question answering.2003.20-27.

[33] Dave K,Lawrence S,Pennock D.Mining the peanut gallery:Opinion extraction

and semantic classification of product reviews.2003.519-528.

[34] Liu H,Lieberman H,Selker T.A model of textual affect sensing using real-world

knowledge.2003.125-132.

[35] Nasukawa T,Yi J.Sentiment analysis:Capturing favorability using natural lan-

guage process-ing.2003.70-77.

[36] Pang B,Lee L,Vaithyanathan S.Thumbs up?sentiment classification using ma-

chine learning techniques.2002.79-86.

[37] Riloff E,Wiebe J.Learning extraction patterns for subjective expressions.2003.

105-112.

[38] Gamon M,Aue A,Corston-Oliver S,et al.Pulse:Mining Customer Opinions

from Free Text.2005,3646:741-741.

[39] Pang B,Lee L.Seeing stars:exploiting class relationships for sentiment categori-

zation with respect to rating scales.2005.115-124.

[40] Hu M,Liu B.Opinion feature extraction using class sequential rules.2006.

[41] Archak N,Ghose A,Ipeirotis P G.Show me the money!:Deriving the pricing

power of product features by mining consumer reviews.2007.56-65.

[42] Tan S,Wang Y,Cheng 10.Combining learn-based and lexicon-based techniques

for sentiment detection without using labeled examples.2008.743-744.

[43] Han J,Pei J,Yin Y.Mining frequent patterns without candidate generation.Pro-

ceedings of Proceedings of the 2000 ACM SIGMOD international conference on Management of data,New York,NY,USA:ACM,2000:1-12.

[44] Quinlan J R.Induction of Decision Trees.Machine Learning,1986,1:81-106.

[45] Joachims T.Text categorization with Support Vector Machines:Learning with

many relevant features.1998,1398:137-142.

[46] Ku L,Chen H.Mining opinions from the Web:Beyond relevance retrieval.Jour-

nal of the American Society for Information Science and Technology,2007,58

(12):1838-1850.

[47] Boelsma E,Vijver L,Goldbohm R,et al.Human skin condition and its associa-

tions with nutrient concentrations in serum and diet.The American journal of clini- cal nutrition,2003,77(2):348-355.

[48] Grove G,Zerweck C,Heilman J,et al.Methods for evaluating changes in skin

condition due to the effects of antimicrobial hand cleansers:Two studies comparing a new waterless chlorhexidine gluconate/ethanol-emollient antiseptic preparation with a conventional water-applied product.American journal of infection control, 2001,29(6):361-369.

[49] Redman T.Data quality:management and technology.Bantam Books,Inc.,1992.

[50] Liu H,Hussain F,Tan C,et al.Discretization:An enabling technique.Data min-

第10章 大数据技术在文本挖掘和情感分类中的应用

ing and knowledge discovery,2002,6(4):393-423.

[51] Kononenko I,Hong S.Attribute selection for modelling.Future Generation Com-

puter Sys-tems,1997,13(2):181-195.

[52] Quinlan J.C4.5:Programs for machine learning.Morgan kaufmann,1993.

[53] Breiman L.Classification and regression trees.Chapman &.Hall/CRC,1984.

[54] Sokal R,Rohlf F.Biometry(2nd ed).New York:WH Feeman and Company,1981.

[55] Pearson E.The choice of statistical tests illustrated on the interpretation of data

classed in a 2×2 table.Biometrika,1947,34(1/2):139-167.

[56] Sun Y.Iterative RELIEF for feature weighting:algorithms,theories,and applica-

tions.Pattern Analysis and Machine Intelligence,IEEE Transactions on,2007,29

(6):1035-1051.

[57] Johnson R,Wichern D.Applied multivariate statistical analysis,volume 4.Pren-

tice hall Upper Saddle River,NJ,2002.

[58]Elman J.Finding structure in time.Cognitive science,1990,14(2):179-211.

[59] Pham D,Liu 10.Dynamic system modelling using partially recurrent neural net-

works.Journal of Systems Engineering,1992,2:90-97.

[60]Shi X,Liang Y,Lee H,et al.Improved Elman networks and applications for con-

trolling ultrasonic motors.Applied Artificial Intelligence,2004,18(7):603-629.

**第** **1** **1** **章** **大数据技术在电力系统中的应用**

本章主要介绍两个内容： 一种云可视化机网协调控制响应特性数据挖掘方法；基于电力 数据分析的河北南网电力市场化风险对冲方法。这两种方法均基于大数据思维对电力系统 中数据进行了深入挖掘和分析。

11.1 一种云可视化机网协调控制响应

特性数据挖掘方法

2015年6月24日召开的国务院常务会议通过了《“互联网+”行动指导意见》(下称《意 见》)。“互联网+”这一新兴产业模式正式成为中国的国家行动计划。2015年7月1日国 务院发布的《关于积极推进“互联网+”行动的指导意见》明确提出“互联网+”智慧能源，这 里+的是智慧能源，而不是仅仅限于已有的智能电网、能源互联网现有概念，具有更深刻含 义，包含了我国能源创新的核心竞争力。因此在智能发电技术领域开展云计算技术研究已 经刻不容缓，能够在保证现有电力系统硬件基础设施基本不变的情况下，对当前系统的数据 资源和处理器资源进行整合，从而大幅提高网内机组对特高压电网实时响应和高级分析的 能力，为智能电网技术的发展提供有效的支持。

火电机组 DCS 系统实时控制的运行数据点在10000点以上，对于每时每刻都在产生的 实时数据，已经能够达到海量数据级别。传统的方法无法提供运行人员更为丰富的三维立 体数据互动关系，只能根据生产过程经验判断；火电机组过程控制，是一个复杂的多变量的 过程控制系统，传统的方法由于被控对象本身信息不足，均是以先验知识和局部知识为基础 的，无法发现深层次的数据关系；电网调度人员之前只能通过调度指令来对电厂进行经验性 操作，无法深入了解发电机组的内部具体情况，从而做出对电网整体性能最优的调度决策， 网侧和源侧缺乏直观有效的沟通机制。

在河北南网，利用成果和方法实现了河北南网主力发电机组的实际数据与调度仿真模 型对接，有机紧密地将生产实际与科研联系起来，同时结合电网运行的可靠性和解决遇到的 各种问题初步分析，并开展了系列化的优化策略体系研究。利用网源协调数据骨干度可视 化分析方法，建立了河北南部电网网源能量平衡仿真调度平台，制定了电网安全稳定裕度评 价体系，最终达到以生产带来的效益推动科技研发的目的，进一步促进科研成果的孵化，开 启了“以科研奠定生产，以生产促进科研”的良性循环的研发模式。通过整合先进控制策略 的迭代优化工作，减少调试时间和增加试验安全性，同时取得了较好的经济效益。

网源协调数据骨干度可视化分析方法，将原有的机组仿真平台改进为可以满足电网调

第11章 大数据技术在电力系统中的应用

度验证的试验仿真平台，针对河北南部电网内所有机组建立网源协调仿真模型，使仿真平台 成为可以真正服务于电网调度运行的技术支撑平台；应用基于 R 语言的超临界机组增网源 协调数据骨干度可视化分析方法对电网调度试验划分边界进行仿真建模评估后，将网内机 组响应电网调度平均速率提高了7%,缩减机组运行成本5%;电网安全稳定裕度提高了 11%,对机组和电网冲击幅度降低了81%。

**11.1.1 技术领域**

本节内容属于智能电网的云计算领域，具体涉及一种云可视化机网协调控制响应特性 数据挖掘方法。

**11.1.2** **背景技术**

火电机组 DCS 系统实时控制的运行数据点在10000点以上，对于每时每刻都在产生的 实时数据，已经能够达到海量数据级别。在以往常规的控制过程中，海量的实时数据都是以 2D 平面的关系映射到运行画面上，或者根据传统控制方案和现代控制理论，采用其中有限 的一组关键运行数据点，作为运行和监控方案的输入输出，提供给运行人员和 DCS 机组进 行合理的控制；传统的方法无法提供运行人员更为丰富的三维立体数据互动关系，只能根据 生产过程经验判断。

火电机组过程控制，是一个复杂的多变量的过程控制系统，关键的运行数据，比如主汽 温和主汽压，往往是经过一个比较明显的滞后时间后，才会发生相应的变化，而给实际控制 性能带来影响，对循环流化床锅炉的机组尤其如此。归根到底，是由于被控对象本身信息不 足引起的；而传统的方法，都是基于热力平衡关系或者传递函数关系得出的分析模型，均是 以先验知识和局部知识为基础的，无法发现深层次的数据关系。

火电机组信息复杂，技术门槛高，电网调度人员之前只能通过调度指令来对电厂进行经 验性操作，无法深入了解发电机组的内部具体情况，从而作出对电网整体性能最优的调度决 策，网侧和源侧缺乏直观有效的沟通机制。

**11.1.3** **方案内容**

本方案所要解决的技术问题是提供了一种符合机组网源协调实际生产、能够反映电网 最优调度决策的云可视化机网协调控制响应特性数据挖掘方法。为解决上述问题，所采取 的技术方案是： 一种云可视化机网协调控制响应特性数据挖掘方法，实施例以600MW 超 临 界机组为例，进行网源协调可视化数据挖掘分析。

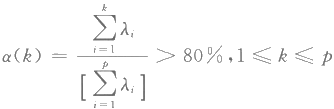
**1.** **具体实施步骤**

(1)在在线热力性能数据校验处理分析平台基础上，实时采集机组的热力性能数据并 对所采集机组的热力性能数据进行规范化校验；

(2)实现机理仿真模型建立：在机组建模过程中将其划分为不同的功能组，对每个功能 组建立子模型，所述子模型建好后通过模型合并，搭建出整个机组模型；

(3)对机理仿真模型进行热力性能精度校验：将在线热力性能数据校验处理分析平台

采集到的实时机组的热力性能数据输入(2)中的机理模型仿真模型，通过机理仿真模型计算

得到热力性能指标计算值，对热力性能指标计算值与最优热力性能曲线求得偏差A, 对机理 仿真模型中已输入的热力性能指标理想值与最优热力性能曲线求得偏差 B, 针对同一功率 所对应的偏差A 和偏差B 进行比较，判断并采用偏差 A 和偏差 B 中较小值所对应的热力 性能指标，得到最优机理仿真模型。

(4)根据机网协调控制原理，确定机网协调控制响应特性为响应时间k;; 机网协调响应 时间的变化由各控制指标的独立变化所引起的响应变化速率叠加而成，从(3)的最优机理仿 真模型中获取样本，建立如下机网协调控制响应特性方程如下(11-1):

(11-1)

其中，k; 为机网协调 AGC 响应时间；x; 为影响机网协调 AGC 响应时间的控制指标的变化 速率；n 为采集数据样本的个数。

(5)确定机网协调响应特性的原始样本数据矩阵 Rxp, 其中脚标 p 为影响机网协调 AGC 响应时间的控制指标的个数；对原始样本数据矩阵Rx, 中的每个值进行标准化归一 处理的计算公式(11-2)如下，消除数据不同量级对计算的影响。

(11-2)

其中，x; 为原始样本数据矩阵Rxp 中第i 样本的第j 个控制指标原始数据；i, 为标准化归 一处理后的值；E(x;) 为原始样本数据矩阵Rx, 中 第j 个控制指标原始样本数据的平均值；

√var(xg) 为 第j 个控制指标原始样本数据方差；原始样本数据矩阵 R,x。中的每个值标准 化归一处理后得到矩阵R\*。

(6)在labview 平台上，使用矩阵和簇工具箱，求解矩阵R”, 获得机网协调控制响应特 性 yì,y₂,…,y, 个主成分，同时得到 p个非负的特征值，将所述 p 个非负的特征值从大到小 排列为λ₁,λ₂,…,λ,,同时得到分别与λ₁,λ₂,…,λ,对应的特征向量u₁,uz,…,up。

(7)利用步骤6中从大到小排列的特征值λ₁,λ₂,…,λ,,计算满足累积方差贡献率α(k) 大于80%所对应的k 值，计算公式(11-3)如下。

(11-3)

求 得k 值后，取λ₁,λ₂,…,λ。中前k 个特征值所对应的主成分代替原来p 个主成分。

(8)以机网协调控制响应特性前k 个特征值所对应的主成分为基础，在 R 语言平台，对 (3)的仿真模型数据进行聚类可视化分析，采用k-medoids 聚类方法，根据(3)的仿真模型数 据，绘制机网协调控制响应特性主成分数据散点图。

(9)以(8)中所述网源能量平衡主成分数据散点图为基础，根据主成分特征值关联程 度，绘制机网协调控制响应特性数据骨干度可视化模型图。

所述(2)中不同的功能组包括风烟系统、主蒸汽系统、高压缸及高旁系统、高压缸及高旁 系统、高加及抽汽系统。

所述(4)中影响机网协调 AGC 响应时间的控制指标包括锅炉控制指标和汽机控制 指标。

第11章 大数据技术在电力系统中的应用

所述锅炉控制指标包括：主蒸汽压力、主蒸汽温度、磨煤机出口温度、烟气排烟温度、 一 次风温度、二次风温度、炉膛负压、再热器入口温度、再热器出口温度、省煤器入口温度、省煤 器出口温度、烟气含氧量、主给水温度和过热度。

所述汽机控制指标包括：调节级压力、调节级温度、#1高加排汽压力、#1高加排汽温 度、#2高加排汽压力、#2高加排汽温度、#3高加排汽压力、#3高加排汽温度、#4除氧 器排汽压力、#4除氧器排汽温度、#5低加排汽压力、#5低加排汽温度、#6低加排汽压 力、#6低加排汽温度、#7低加排汽压力、#7低加排汽温度、凝结水温度和凝结水真空度。

所述(9)中采用 labview 语言将所有数据转换成 CSV 格式，供 R 语言使用。将在线机 组数据和仿真模型对接，建立实际机组特性相似度满足精度要求的仿真。

采用上述技术方案所产生的有益效果在于：本方法实现了按照网内运行机组响应网侧 能量需求调度的云可视化数据挖掘仿真技术和仿真平台进行实时数据传输，将仿真平台调 制至和实际机组与电网特性逼近，验证高级优化算法并建立网源安全稳定裕度评估体系，为 电网调度服务。

**2.** **模型的具体方法**

将在线机组数据按照工况分为若干工况预置条件，首先按照任一工况运行静态数据对 机组进行仿真设计校验，使得仿真模型的设计工况满足实际生产试验的精度要求，比如原有 仿真模型只要求在50%和100%负荷工况满足仿真操作要求精度即可，那么为了满足仿真 试验的要求，就进一步将仿真机工况细分，以10%为档位，分别设置10%、30%、50%、70%、 100%等负荷工况条件，需要将机组在线数据对仿真模型进行修正，提高模型的精确性和可 用性；其次，针对所述仿真模型的动态特性，利用在线机组数据和仿真模型输出数据的偏差， 迭代优化最优的仿真模型参数数值，完成实际机组特性相似度满足精度要求的仿真模型。

对数据进行预处理的具体方法如下：根据网源能量平衡的关系，将仿真模型的所有有关 网源协调数据的原始变量进行标准化处理；输入原始数据矩阵或相关系数矩阵列到 princi- pal() 和 fa() 函数中，在计算前确保数据中没有缺失值。

在新华 OC6000E 仿真平台上，基于实际机组特性高度吻合的600MW 超临界火电机组 仿真模型，针对网源协调控制回路，使用R 语言平台对网源协调控制回路产生的海量实时 数据进行主成分分析和k-medoids 聚类分析，最终确立基于骨干度的网源协调数据云可视 化深度数据挖掘方法。

11.2 基于电力数据分析的河北南网电力

市场化风险对冲方法

随着电力市场化改革的深度开展，发电、用电、配电各方企业，迫切需要一种能够迅速 反应市场供求情况的价格发现工具，帮助电力价格从国家定价模式转化为市场定价模式。 由于电力不能存储的特殊性，无法采用常规方式来进行电价市场化调节，所以采用金融资本 市场作为电力行业改革和发展的突破口，加快电力市场化的进程，成为必行之路。本节从电 网对发电侧、用电侧两个方向进行开展电力期货可行性分析，提出切实可行的河北南网电力 市场化风险对冲方法。

在传统的电力工业管理体制下，政府统一管理电价，电价的波动很小，几乎没有独立发、 输、配电企业，因此不会面临由于电价波动造成的风险。但随着电力体制向市场化方向改革 的进行，电力市场中批发电价和零售电价都将逐步放开。

2015年11月30日，《中共中央国务院关于进一步深化电力体制改革若干意见》公布以 来，新一轮电力体制改革正向核心区快速推进。

发电和售电环节引入竞争，建立购售电竞争新格局；建立市场化价格机制，引导资源优 化配置电价；建立“管住中间、放开两头”的体制架构在发电侧和售电侧引入竞争机制，将建 立购售电竞争新的格局。在可竞争环节充分引入竞争，符合准入条件的发电企业、售电公司 和用户可自主选择交易对象，确定交易量和价格，打破电网企业单一购售电的局面，形成“多 买方一多卖方”的市场竞争格局；电价通过市场竞价方式来确定，将不可避免地导致市场价 格的波动。日内负荷处于高峰时的实时电价与负荷处于低谷时的实时电价可以相差几倍， 而不同日期、不同月份的电价则相差更大。由此带来的电价的剧烈波动将使电力市场的参 与者面临巨大的价格风险。事实证明，电力市场一旦出现大的价格风险，其严重程度一点也 不亚于国际金融业中一些十分著名的金融风险事件。

价格是资源配置最重要的手段，计划经济条件下，价格信号是失真的，不可能实现资源 的优化配置；进入市场经济后，单一的现货市场也无法实现；只有期货市场与现货市场相结 合，才能使资源的优化配置得到充分实现。缺乏期货市场的预期价格，政府只能根据现货价 格进行宏观调控，电力发展缺乏科学的规划能力，造成社会资源的巨大浪费。期货市场产生 的价格具有真实性、超前性和权威性。政府可以依据其来确定和调整宏观经济政策，引导企 业调整生产经营规模与方向，使其符合国家宏观经济发展的需要。

**11.2.1** **电网对发电侧市场化风险对冲分析**

为了更科学、严谨地推进煤电联动以及更准确地监测电煤价格，国家发改委于2015年 9月30日推出中国电煤价格指数，作为煤电联动价格基础。

对煤电价格实行区间联动。以5000大卡/千克代表规格品电煤价格为标准，当周期内 电煤价格与基准煤价相比波动不超过每吨30元(含)的，成本变化由发电企业自行消纳，不 启动联动机制。当周期内电煤价格与基准煤价相比波动超过每吨30元的，对超过部分实施 分档累退联动，即当煤价波动超过每吨30元且不超过60元(含)的部分，联动系数为1;煤 价波动超过每吨60元且不超过100元(含)的部分，联动系数为0.9;煤价波动超过每吨100 元且不超过150元(含)的部分，联动系数为0 .8;煤价波动超过每吨150元的部分不再联 动。按此测算后的上网电价调整水平不足每千瓦时0.2分钱的，当年不实施联动机制，调价 金额并入下一周期累计计算。按煤电价格联动机制调整的上网电价和销售电价于每年1月 1 日实施。

以此次全国燃煤发电上网电价下调3分钱为例，煤价将有50元/吨左右的下调空间，后 期幅度有可能更大。也就是说，煤价下调引发电价下调时，将会压缩煤电企业利润，对煤电 企业均为利空消息。反之，当煤价上涨触发煤电联动机制时，煤价上涨依然提高煤企收入 而电价上涨将提高电力企业收入，利好煤电企业。

新的煤电联动机制则直接提到工商业用电价格相应调整，对降低企业成本有好处。 最新的煤电联动价格机制是基于现货的一种风险对冲方法，要求电网公司和发电企业必

第11章 大数据技术在电力系统中的应用

须100%从现货市场购电购煤，无法提供反映长期供求关系的价格信号。 一旦发生长期 的下跌局面，则对电网企业和发电企业造成长期的不利影响，所以经过分析，采用动力煤 期货合约和焦煤期货合约，分析动力煤和焦煤的期货价格走势，根据季节性规律和统计 套利回归，利用期货价格发现功能，合理的参与动力煤和焦煤期货，可以对冲煤电联动的 风险，具体如表11-1 所示。

**表11-1** **电力风险对冲煤电联动关系示意表**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 煤炭企业 | 发电企业 | 电网企业 |
| 风险对 冲方法 | 煤电联动的同时，根据期货发现远期煤炭下  跌前兆，做空动力煤期货和焦煤期货空头头  寸，对冲煤炭现货下跌带来的风险，同时在现  货跌势稳定后，逐步建立期货多头头寸，以弥  补煤炭企业因为市场不景气带来的利润损失 | 根据季节性规律，在 淡季做多动力煤期货， 在旺季做空动力煤期  货，期现套保规避风险 | 根据季节性规律和统计套利回  归，在淡季等待统计套利底部回归 做多动力煤期货，在旺季等待统计 套利顶点回归动力煤期货，利用煤 电联动机制，规避风险 |

**11.2.2** **电网对用电侧市场化风险对冲分析**

由于工业用电远超于民用电，所以在本章中，用电企业以工业用电企业为研究对象，煤 炭企业作为特殊的用电企业另作讨论，发电企业主要以火力发电分析。在实际社会生产体 系中，关系如下：

煤炭企业 →火电厂 → 电网输配 →工业用电企业

根据生产供求关系，存在能体现电价的关联组合，如表11-2所示。

**表11-2** **电网对用电侧市场化风险对冲组合表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 电网 | 煤炭 | 钢铁 | 化工 | 焦化 |
| 煤电联动 | 动力煤/焦煤 | 动力煤/螺纹钢 | 动力煤/(塑料、聚丙烯、PTA) | 动力煤/焦炭 |

以上各品种在期货市场中均存在相对应的品种。这就为采用金融手段建立对冲机制提 供了现实可能性。

季节性图表法是指在研究价格季节性变动时，计算出相应的价格变动指标，并绘制成图 表来发现商品的季节性变动模式。它主要包括的指标有上涨年数百分比、月度收益率、平均 最大涨幅与平均最大跌幅、平均百分比等。

(1)月度收益率=(本月底价格一上月底价格)/上月底价格

(2)最大涨幅=期间最高价一期初价

(3)最大跌幅=期间最低价一期初价

(4)期初百分比=(期初价格一该期最低价)/(该期最高价价格一该期最低价)

(5)期末百分比=(期末价格一该期最低价)/(该期最高价价格一该期最低价)

运用季节图表法可以对煤钢焦化工价格的季节运行规律进行分析，总结出动力煤及其 上下游价格在一年中不同月份的强弱关系，并对三者之间的季节性传导及主导这种传导机 制的钢铁和化工行业的季节性进行指导。

按照以上方法，结合煤电联动机制可以建立电网与用电企业的风险对冲关系。

由于商品的供需层面存在着某些季节性变化特点，即随着季节的转换商品供给或需求 的增减趋势相对固定，这些商品的价格也因此带有季节性波动特性，我们将这种波动特性称 为季节性波动规律。

下面以动力煤和铜为分析对象进行煤电联动季节性分析。季节性分析的基本逻辑为： 淡旺季交替形式的价格波动。

**1.** **动力煤季节性规律分析**

动力煤跨季节风险对冲基本逻辑：淡旺季交替形成的价格波动。

动力煤活跃的三个合约1月、5月、9月在时间跨度上恰好经历几个季节性波动周期：旺 季夏季用电高峰和冬储用煤高峰。

夏季用电高峰源自工业用电与居民商户用电高峰的叠加，夏季高温时期空调的广泛使 用及工厂生产及降温措施通常是夏季耗电的主要因素，由于中国主要发电来源于火电，因此 7、8月份也是煤炭需求旺盛的季节。

冬储用煤高峰主要源自冬季取暖用煤的增加及春节放假的生产安排。进入冬季以后取 暖用煤需求增加，在北方，都有冬季屯煤的习惯。每年2月份前后，多数煤矿安排放假，即使 生产也考虑到安全性问题而减少产量，港口作业几乎停滞。通常进入十月份冬储活动就开 始了， 一直持续到12月月底，旺季结束。

穿插在两个旺季之间是淡季，通常用煤下降，港口船只数量较少，价格也进入低谷。

动力煤的三个活跃合约1、5、9月份合约，交割期处于一个旺季1月和两个淡季5月和 9月之间。随着交割期的临近旺季合约价格贴近现货而淡季合约价格下跌，价差逐渐拉开。 形成独特的套利风险对冲机会。图11-1显示临近旺季合约和淡季合约价差逐渐放大，形成 套利风险对冲机会。

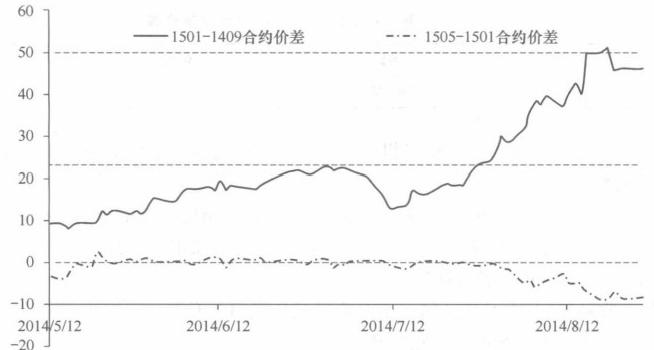


图11-1 动力煤跨期季节性风险对冲分析图

**2.** **铜季节性周期规律分析**

如图11-2所示，每年的2月份到5月月初，铜消费处于春季旺季，铜价通常持续大幅走 高， 一年当中的最高点也常常出现在这个阶段；随后春季高峰结束，到6月中旬期间，铜价常 常会经历较大幅度的调整；接着铜价一年中的第二轮上涨开始， 一般从6月月底持续到9月 份，迎来秋季消费高峰，涨幅通常不及春季旺季，现年内次高点；随后铜价回落，进入休整阶 段，这一阶段的调整持续的时间往往较长。

第11章 大数据技术在电力系统中的应用



图11-2 铜季节性风险对冲分析图

对以往16年数据进行了涨跌概率的统计，如表11-3所示。

**表11-3** **铜历年风险分析统计概率表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 月份 | 上涨概率 | 平均涨幅 | 上涨年数 | 下跌年数 |
| 1 月 | 75.00% | 2.40% | 12 | 4 |
| 2 月 | 68.75% | 2.61% | 11 | 5 |
| 3 月 | 68.75% | 2.26% | 11 | 5 |
| 4 月 | 50.00% | 3.65% | 8 | 8 |
| 5 月 | 31.25% | —0.62% | 5 | 11 |
| 6 月 | 43.75% | -1.77% | 7 | 9 |
| 7 月 | 62.50% | 0.44% | 10 | 6 |
| 8 月 | 62.50% | 1.45% | 10 | 6 |
| 9 月 | 68.75% | 1.69% | 11 | 5 |
| 10月 | 31.25% | -3.55% | 5 | 11 |
| 11月 | 56.25% | 一1.17% | 9 | 7 |
| 12月 | 50.00% | 0.21% | 8 | 8 |

**3.** **动力煤铜季节性风险对冲分析**

可以从上面的对动力煤和铜的季节性规律分析看出，从2013—2015年，每一年的10月 中旬开始，到12月上旬，都会有规律地出现铜和动力煤的价差的下跌，即使当时时间段的动

大数据技术及行业应用

力煤和铜出现波动都会不同，但是铜和动力煤的价差的规律却是固定的，所以根据图11-3 所示，在每一年的10～12月周期内，采取手段，做多动力煤1月份期货合约，做空铜1月份 期货合约，则可以对电网企业和发电企业以及铜用电企业进行风险对冲。该风险对冲方法 可以重复且稳定，对风险波动有很强的抗性。

沪钢1701-郑煤1701(自设营利0)日线 36902 928/2.589%

M



2013年12月止句4

2014年10月中旬

46154

2013年10月中司

% ”N

43104

晰，

2014年12月上旬



2015年10月中旬

38982

38015

552天

2014/03 2014/11

2015年12月中旬

2015/07

32901

图11-3 动力煤&铜季节性风险对冲分析图

**11.2.3** **基于方差偏离规律的统计套利对冲方法**

统计套利策略是一种市场中性策略，它通过对相关品种进行对冲来获得与市场相独立 的稳定性收益。统计套利策略背后的基本思想就是均值回归，也就是说两个相关性很高的 投资标的价格之间如果存在着某种稳定性的关系，那么当它们的价格出现背离走势的时候 就会存在套利机会，因为这种背离的走势在未来会得到纠正。在实际投资中，在价格出现背 离走势的时候买进表现相对差的，卖出表现相对好的，就可以期待在未来当这种背离趋势得 到纠正时获得相对稳定的收益。

不过单纯的统计套利存在一定的问题，即单纯的数学方差统计判断风险对冲的方式，由 于存在相当程度的统计随机波动性，并不是完全适用于实际操作环境，因此综合季节性风险 对冲分析方法，对统计套利体系进行周期和方向性的筛选，可以提高风险对冲体系的可 行性。

首先，建立风险对冲的监控体系，对各种风险对冲组合进行实时监控，如图11-4所示。

其次，在季节性规律的前提下，对风险对冲组合进行统计套利对冲分析。以动力煤和铜 为例，可以看到如图11-5 所示，该策略是可以在已经确定明确季节性规律的前提下，获得统 计对冲风险收益的成功的。

本节从电网对发电侧、电网对用电侧两个方向进行开展电力市场化风险应对开展可行 性分析，提出切实可行的河北南网电力市场化风险对冲方法。提出了一种基于季节性规律 的风险对冲方法，进一步提出一种基于方差偏离规律的统计套利风险对冲方法。并开发了 一套完整的风险对冲工具，可以实时通过对电网侧、发电侧、用电侧各期货相关品种的分析 跟踪，利用金融手段合理有效的实施风险对冲，用以达到帮助电力价格从国家定价模式转化 为市场定价模式的目的。

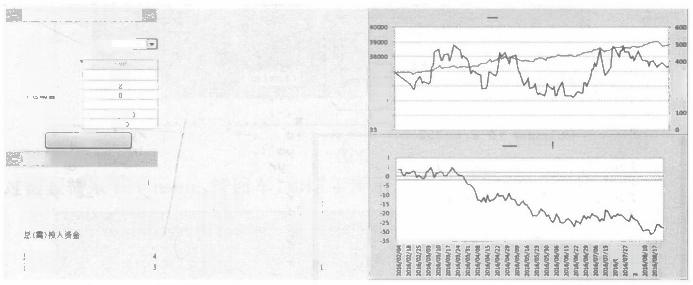
**第11章** **大数据技术在电力系统中的应用**

>>

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 同品种对冲 | 主 力 | 次主力 | 价差模式 | 价差 | 持 仓 量 比 | 相关性 | 波动率 | 冲击成本 | 价差走势 |
| PTA | TA1605  01605  ZC1605  RM1605  WH1605  SR1605  CF1609  R11605  MA1605  FG1605 | TA1609  01609  2C1701  RM1609  WH1705  SR1609  CF1701 | 协整  协整  协整  协整  协整  协整  协整  协整  协整  协型 | 0.79 | 0.19  0.79  16.61 0.58 1.00 1.17 0.39 3.33 0.14 0.16 | 92.1% 85.3%  88.3%  89.8% 100.0% 98.0% 91.2% 91.2%  97.7% 97.0% | 0.007 0.00 0.015 0.009 0.000 0.00 0.017 1.555 0.008  0.009 |  |  |
| 冀油 | -1.57 |
| 动力煤 | 1.32 |
| 菜粕 | -1.59 |
| 强 查 | \*DIV/0! |
| 白 | -0.32 |
| 魏花 | 0.00 |
| 早轴稻 | RII607  MA1609  FG1609 | U |
| 郑醉 | 0.50 |
| 玻墙 | 0.84 |
| PVC | V1605  P1605  M1605  11605  101605  P1mS  L1605  Y1605  C1609  A1605  11605  41605 | V1609  P1609  M1609  11609  J01609  PP1609  11609  Y1609  C1701  A1609  11609 | 协型  协整  协整  协整  协整  协整  协整  协型  协整  协整  协整  协整 | 0.39 | 0.24 0.3 1.1 0.4 0.3 0.4 0.3 0.8  088 0.70 0,3 0.2 | 99.5% 94.79 89.79 99.0% 68.35 99.09  99.4% 93.49  77.2% 97.4% 99.29 98.2% | 00  0.009 0.009 0.017 .019 0.00 0.00  0.0  0.019 0.006 0.00 0:00 | 5 |  |
| 朝 | 0.69 |
| 豆粮 | 0.88 |
| 秩矿石 | 2.97 |
| 鸡蛋 | 1.01 |
| PP | 3.13 |
| 型料 | 2.20 |
| 豆油 | 1.57 |
| 王米 | 1.08 |
| 豆一 | 2.33 |
| 焦寒 | 2.30 |
| 蕉焊 | 2.26 |
| 沪焰 | AL1605  RU1605  ZN1605  CU1605  AU1606  RB1605  PB1604  AG1606 | AL1604  RU1609  2N1604  Cu1701  AU1612  R81610  PB1605  AG1612 | 协型  协型  协整  协整  协整  协整  协整  协整 | 0.79 | 0.5 0.59 0.66 55.6 0.10 0.68  0.73 0.02 | 99.7 79.6%  100.09  58.49 100.09  99.5  96.3% 97.8% | 0.0  0.015 0.001 0.013 0.001  0.005 0.000 0.004 | 100 |  |
| 槽胶 | 0.48 |
| 户体 | 1.12 |
| 户镇 | 235 |
| 黄金 | 0.37 |
| 螺政钢 | -2.18 |
| 沪烟 | 0.09 |
| 自银 | -0 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 跨品种对冲 | | 品种1 | 品种2 | 价差模式 | 价差 | 相关性 | | 波动率 | 冲击成本 | 价差走势 |
| 金属 | 沪调/沪铝 | C11605 C01605  C01605  AL1605 AL1605 ZN1605 AU1606 | AL1605 ZR1605  PD1504  ZN1605 PB1604 PB1604 AG1606 | 协整  协整  协整  协整  协整  协整  协整 | 2.36 |  | 73.8%  86.9%  88.4% | 0.011 0.008 0.008  0.012 0.014  0.013 0.030 | 95  95  90  1.05 |  |
| 沪调/沪梓 | 2.52 |
| 沪钢/沪铅 | 3.06 |
| 沪铝/沪锌 | 0.28 | 89.1% |
| 沪铝/沪始 | 0.81 | 83.8% |
| 户掉/沪铅 | 1.08 | 95.1% |
| 黄金/白银 | 2.92 | 82.4% |

**图11-4** **风险对冲组合实时监控系统**



铜动力煤对冲交易测试 一u1701 —E1F@1

测试方式

可海详本

份整碳式

五自**分雨词**

止播面

段入

最 长 特 仓 愿

000

测 试

”

测 试 时 间： 201624~2016VE

主平仓错员

总亏损

手续费

资金回报率

胜 率

亏损次数

178,158.05 175,5677

428.72

500.000.00

-409.52

400.52

2,405,15

35.05%

净 利 闲 总 盈 利 量 号 比

交易测试报告

总交易次数 量利次数

s4000

35000

4000



500,000

1000

-标化价



级



75.00%

srooo

图11-5 风险对冲组合品种关联度分析示意图



**本** **章** **小** **结**

大数据思维在电力系统中的应用尚有很大空间可以挖掘 ， 本章提出了两种方法对电力

大数据进行挖掘分析，希望能对读者有抛砖引玉的启示。

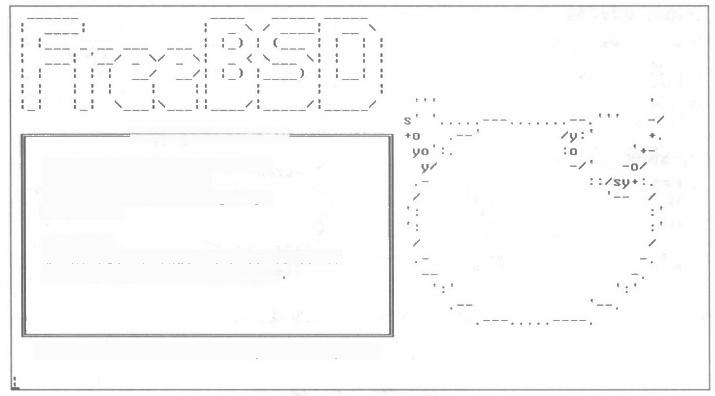
**附录** **FreeBSD 操作系统安装**

在 FreeBSD 上有许多免费的软件，这些软件大都已移植收录于 FreeBSD ports 中，这 就使得 FreeBSD 的安装变得十分轻松。除了传统的光盘安装外，也可以使用网络安装、 MS-DOS 分割区安装等。当然，我们也可以在计算机中同时安装多种不同的操作系统，例 如 Windows XP和 FreeBSD 同时并存也是件非常容易的事情。在这里我们主要介绍如何 使用光盘进行安装。注意，本例子是使用整个硬盘来创建的，这样做之后原硬盘的所有 数据都将丢失，所以安装之前 一 定要考虑清楚，有必要的话最好把重要的数据都复制 下来!

安装前的准备：FreeBSD 的光盘一张，如果没有可以到 www.freebsd.org 下载一张量 新版本的镜像文件(ISO)。

然后设置主版以及光驱的启动，具体操作如下：

首先，在开机时按 DEL 键，在ADVANCED 选项中把第一启动改为cd-rom, 然后放入 光盘并重新启动电脑，此时就会看到下面的界面，如图1所示。

Helcone to FreeBSD

1.Boot Multi User [Enter]

2.Boot Single User

3.Escape to loader prompt

4.Reboot

t:default/kernel(1 of

2)

6.Configure Boot Options..

Autoboot in 9 seconds.[Space]to pause

图 1

此时进入到了安装模式，等待几秒后会自动进入下一个界面。由于 FreeBSD 是以文本 模式进行安装的，初学者可能不太习惯，但是认真学习，会发现其实它很简单，也很安全。

**注意：**选项之间的选择是通过键盘的方向键或 Tab 键进行切换的，按回车键表示选择 默认方式。

第11章 大数据技术在电力系统中的应用

通过鼠标上下键选45 China, 然后切换到 OK, 按回车键，如图2所示。

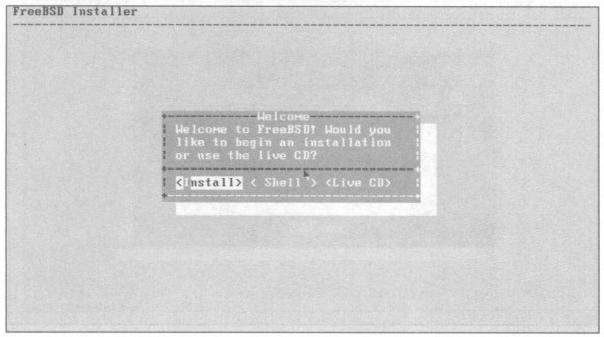
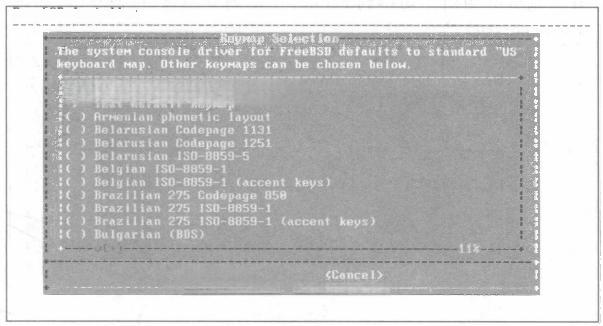


图 2

接下来是键盘布局选择，这里选默认，按回车键，如图3所示。

FreeBSD Installer

**cont imte tith doefault keyrap**

**<select>**

**PregS aT**TUHS

TaB Or KNTER]

图3

通过键盘输入 hostname, 按回车，如图4所示。

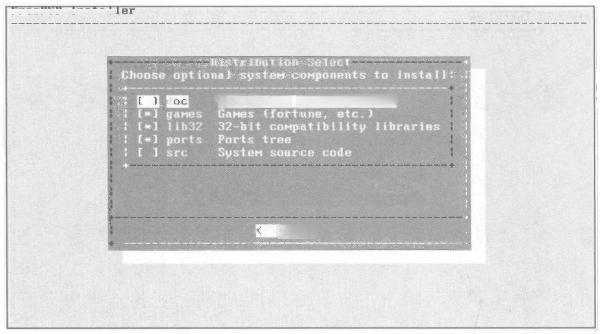
|  |
| --- |
| FreeHST TnstalTer  HostnaHe-  Please choose a hostnare for this nachine.  It,yon are rummtno en.t.rnco neteorkPlonse. |

图 4



大数据技术及行业应用

进入下 一 步后，选择 OK, 如 图 5 所 示 。



FreeBSU InstaN

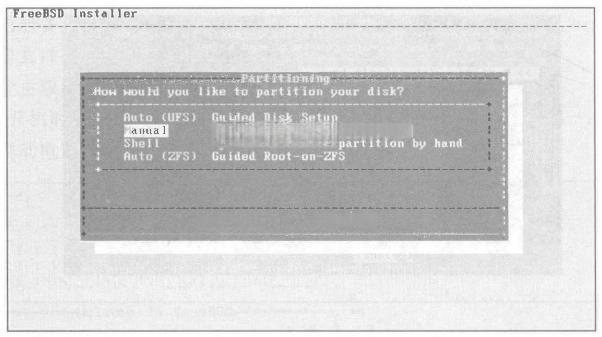
**Additional documentation**

R 2

图 5

马上就要到最危险的地方了， 一 般数据的丢失都是由于这里的错误操作引起的，所以 一

定 要 认 真 地 把 硬 盘 划 分 弄 明 白 。 切 换 到 Manul 手动划分磁盘，按回车键，如图6所示。

Hanual Disk Setup(experts)

R

KK ) <Cancel>

图 6

下 一 步 切 换 到 Create, 选默认项，用 Tab 键 切 换 到 OK, 按 回 车 键 ， 如 图 7 所 示 。

|  |  |
| --- | --- |
| FreeBSI |  |
| Bootable on Host xB6 systers | |

**图** **7**

附录：FreeBSD 操作系统安装



这里仍然选择 OK, 按回车键，如图8所示。

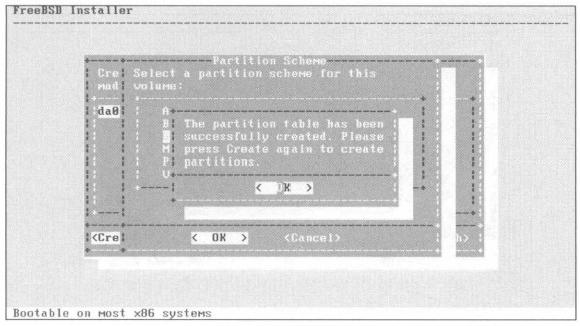
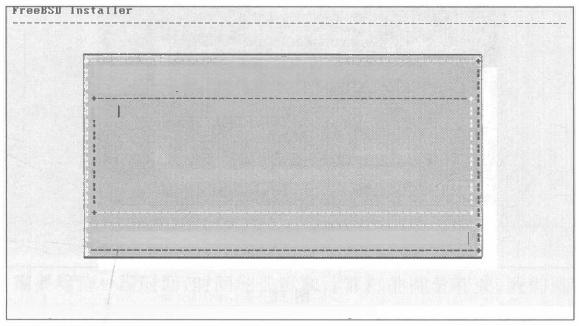


图 8

下面就要开始划分硬盘空间了，要特别注意这里的操作，不合理的划分可能会带来很多 麻烦， 一定小心。这里是分给：根目录“/”16G, 交 换 分 区“swap”4G 。 当 然 用 户 也 完 全 可 以 根据自己的需求划分， 一般 swap 是 内 存(RAM) 的2倍，内存容量太大是没有必要的。用 Tab 切换到输入框，输入要划分的大小，例如这里写16G, 如 图 9 和 图 1 0 所 示 。



Partition Editor

 unptairlti sctTBisShD.No changes ulll be

:tn8

<Create><Delete><Modify><Revert><Auto >Fnish?

Exit partitioner (uill ask whether to save changes)

28 GB

图 9

|  |
| --- |
| **TreeHSD InstalTer** |

**图10**

大数据技术及行业应用



按回车键，如图11所示。

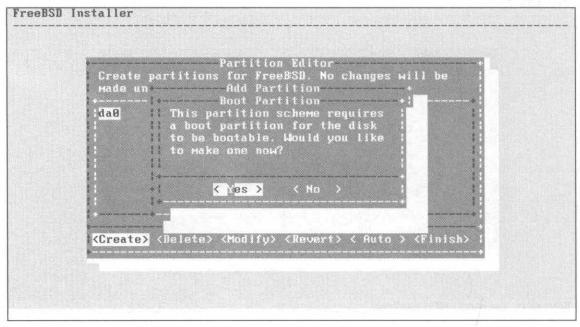


图 1 1

类似以上步骤，现在给 swap 划分空间，首先切换到 Create 创建，再切换到输入框，输入 需要划分的大小( 一 般是物理内存(RAM) 的 2 倍 ) , 这 里 输 入 的 是 4G, 如 图 1 2 所 示 。

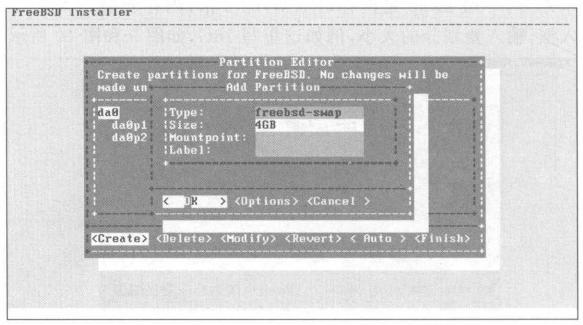


图 1 2

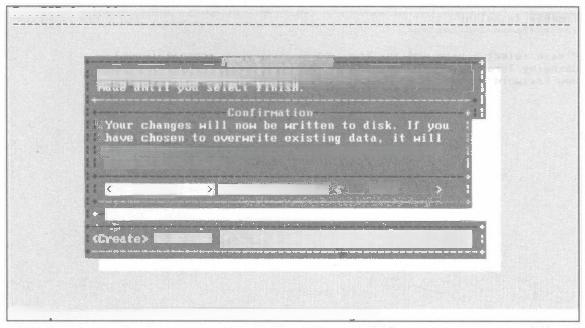
按 Q(Finish) 完成，如图13所示。

|  |  |  |
| --- | --- | --- |
| TreeHST  Partition Editor  edaeunpttseflct **F**riesShD.Mo changes will be | | |
| :1ad  daBp1 daBp2 daBp3 | 28 BB 512 NB 16 GB 4.8 GB | GPT  fraebsd-hoot  freebsd-ufs  freebsd-suap |
| <Create>(Delete>(Modify><Revert><Auto > Fmis  Exit partitioner(ufll ask uhether to save changes) | | |

图 1 3

附 录 ：FreeBSD 操 作 系 统 安 装

是否要提交锁做的改变，按回车键，如图14所示。



TreeBSD InstalTer

artItton EdItor

Crete rtition,for FreetSD.Mo changes will be

be PERMANENTLY ERASED.Are you sure you nant to

cohnit your changes?

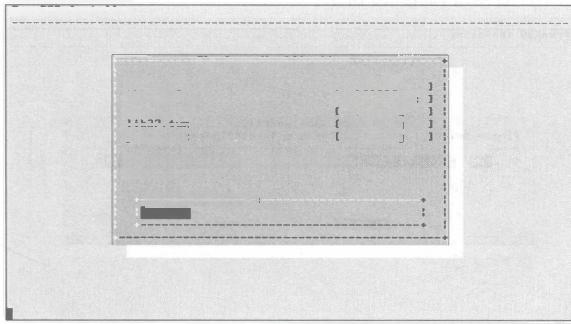
oit <Revert &Exit> Back

<Delete> <hod1fy><Revert?<Ruto>KFinish)

Exit partitioner (uill ask uhether to save changes)

图14

接下来，初始化硬盘分区及文件系统，如图15所示。

FreeBSD InstalTer

Checksun

base.txz

kernel.txz ganes.txz

ports.t

Uerifying checksuns distributions.

Overall Progress

UerIfIcatlon

[ Passed

[In Progress

Pandlno

Pending

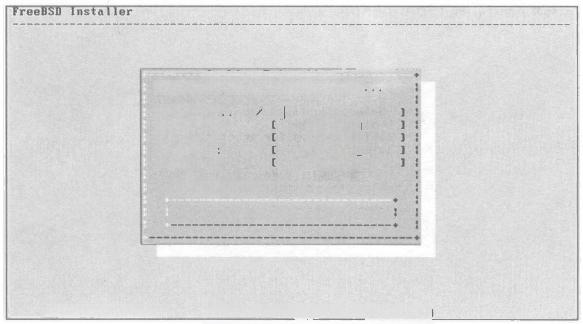
Pending

of selacted

28%

图 1 5

正在安装，需要等待一段时间(时间的长短和计算机的配置有关，我们可以看进度条来 进行观察),如图16所示。

Archive Extraction

Extracting distribution files.

base.txz.

[

Pon%ing

Pending

Pending

Pending

kernel.txz

games.txz

iib32.txz

ports.txz

Overa11 Progress:

1%

1821 files read @ 178.8 files/sec.[1/4 busy/uait1

图 1 6

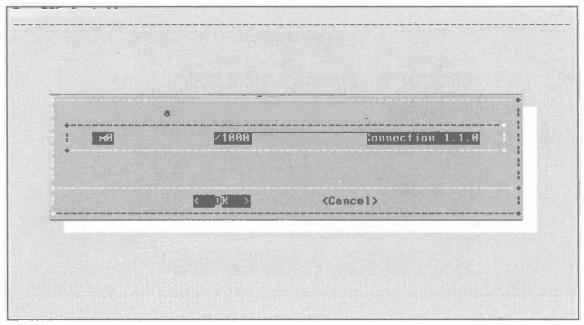
大数据技术及行业应用

安装成功，提示输入密码，如图17所示。

|  |
| --- |
| FreebSD Installer  Please select a passuord for the syster Hanagenent account(root):  Changing local passhord for root  NeH PassHord: |

图17

选择一块网卡配置网络，如图18所示。

FreeBSD InstaIler

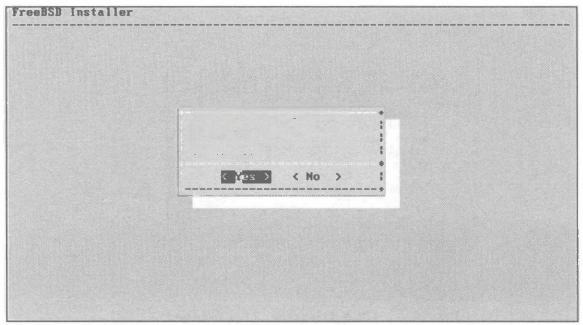
Netuork Configuration

Please select netuork interface to configure:

[mteI(R)PRDZ Legacy Nettork C

**图18**

**配置** **IP, 按回车键，如图19所示。**



Netuork Configuration

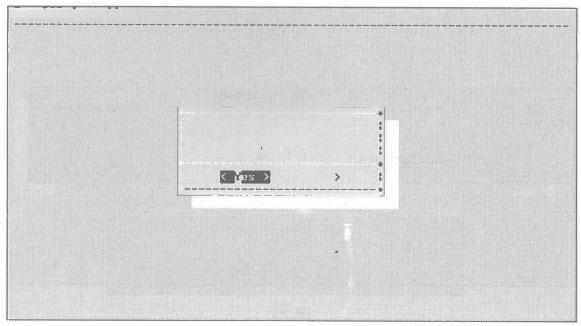
g\_I1P1kue4 tor this

Interface?

图19

附录：FreeBSD操作系统安装

是否使用DHCP 服务自动获取IP, 选择 Yes 按回车键继续，如图20所示。

TreeBSD InstalTer

Netuork Configuratlon

Hould you like to use

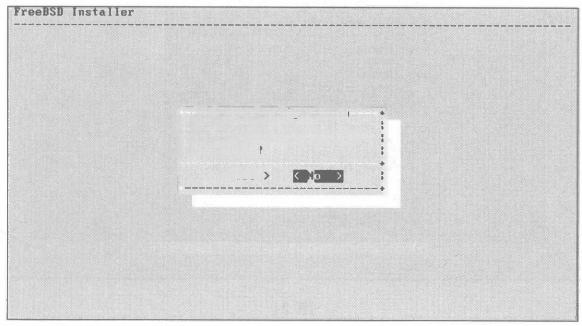
DHCP ta configure this

interface?

<No

图 2 0

是否要设置 IPv6, 选择 No, 按回车键继续，如图21所示。



Netuork Configuration

Nould you 11ke to

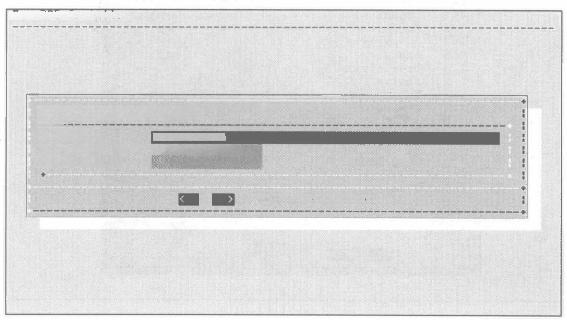
configure IPu6 for this

Intnrtace'

<Yes

图21

配置 DNS 服务器，默认即可，按回车键，如图22所示。

TreeBSD InstaITer

Netuork Configuration

Resolver

Configuratlon

\*=

localdora in

!Search

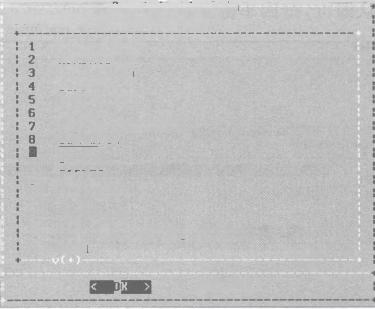
IIPv4 DNS 81 1IPv4 DNS 02

<192.168.228.2>

p

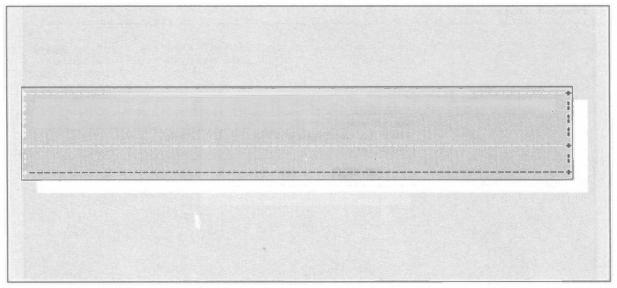
《Cancel>

图22

大数据技术及行业应用

Atlantic Ocean Australia

设置时钟，切换到 No, 按 回 车 键 继 续 ， 如 图 2 3 所 示 。

Select local or UTC(Greenuich

Time)clock is set to

Mean If it

machine's CMOS clock set to UTC?

Is this or you

local tine,

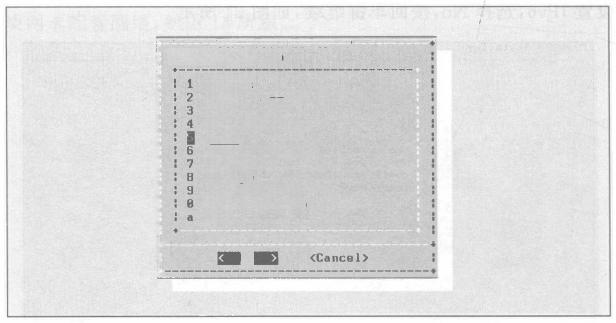
don't knou,please choose NO heret

K

<No >

图 2 3

到这个界面，用方向键选择 Asia ( 亚 洲 ) , 切 换 到 OK, 按 回 车 键 ， 如 图 2 4 所 示 。

Tine ZoneSelector

a region

Select

Africa

Arer ica North and South

Antarctica

Arctic Ocean

asia

Europe

Indian Ocean

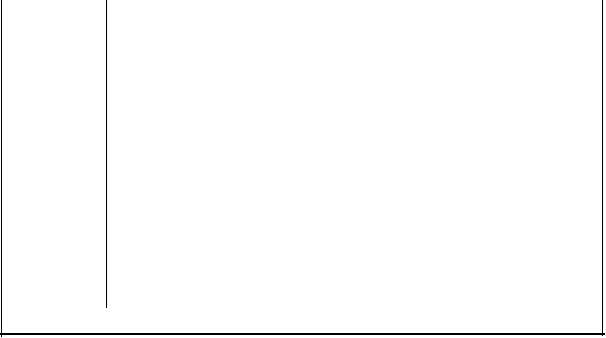
Pacific Ocean

UTC

pg

图 2 4

选 择 China ( 中 国 ) , 切 换 到 OK, 按 回 车 键 ， 如 图 2 5 所 示 。



Countries in Asia

Select a country or region

Afghanistan

Arhenia

Azerbaljan

Bahrain

Bang ladesh

Bhutan

Brunel Darussalan

Cambod la

china

Cyprus

Georgia

Hong Nong

India

Indones ia

Iran,Islamic Republic of

Irag

32%

<Cancel》

1e

11

12

13

14

15

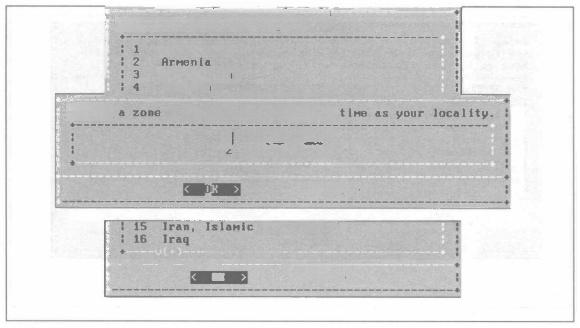
16

图 2 5

附录：FreeBSD操作系统安装



根据自己的地理位置进行选择，这里我们选1,切换到 OK, 按回车键，如图26所示。

Countries in Asia

Select

a country or region

Afghanistan

Azerba1jan

Bahrain

China Tine Zones

Select

thich observes the sarHe

Reitiny Tirm

Xinjlang Tire

<Cancel>

Republic of

32\*

<Cancel>

OK

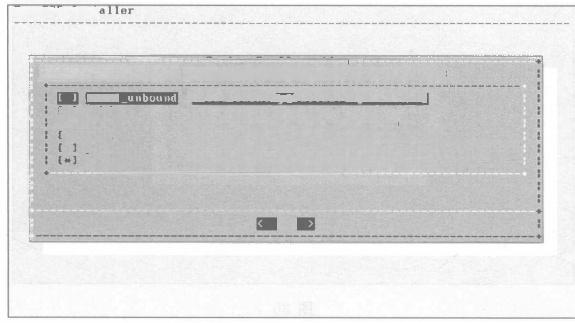


图26

在选择开机自启动的服务，可以根据自己的情况进行选择，如图27所示。

TreeBSD Tnsta

yoytldCiegainstarted at boot:

Local Cachlng Valfdatiny Fesolver

Secure shell daeron

PS/2 Mouse pointer on console

Synchranize systen and netuork tine

Ad just Cpu frequency dynanically if supported

pg

图27

询问是否要添加新用户，选择 No, 如图28所示。

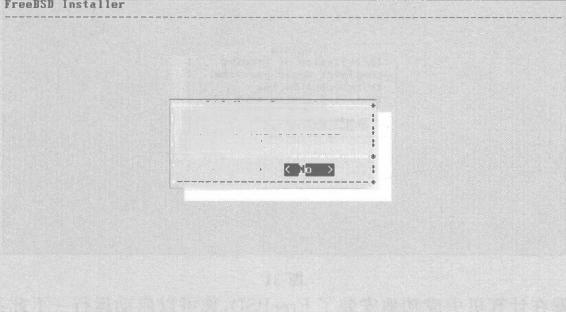
Enable kernel crash dumps to /var/crash

Choose the services

[ocal

[\*] sshd []moused 1 ntpd

pouerd dumpdev



Add User Accounts

Hould you like to add

users to the installed

<Yes>

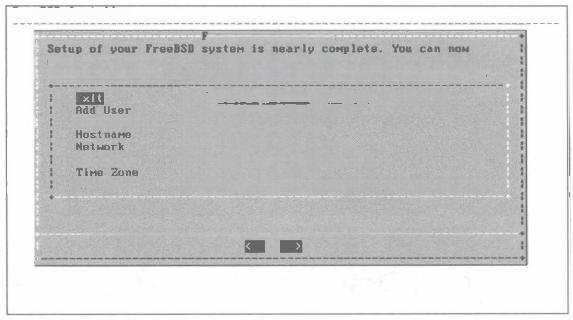
systen nou?

图 2 8



大 数 据 技 术 及 行 业 应 用

如果有还需要设置从这里可以重新进行设置，没有的话，按回车继续，如图29所示。



TreeBSD Tnstaller

inal

wodify your configuratlon

have an opportunity to nake

**w**d**p****y**a **r**t**a**h**!**s**a**t **exit installer**

Change root passuord

Handbook

pR

Set systen hostname

MetuorkIng configuration

Set daenons to run an startup

Sot systen tinezone

Install FreeBSD Handbook (requlres netuork)

Configurat ion choices.After

More complex

Root Passuord

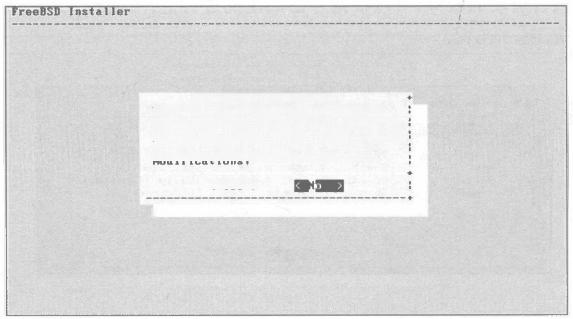
Services

this scraen,you changes using a

wlll shell.

图 2 9

这里选择 No, 按回车键继续，如图30所示。



Manual Configuration

The installatlon is nou finished.

you like to open a shell in the neu

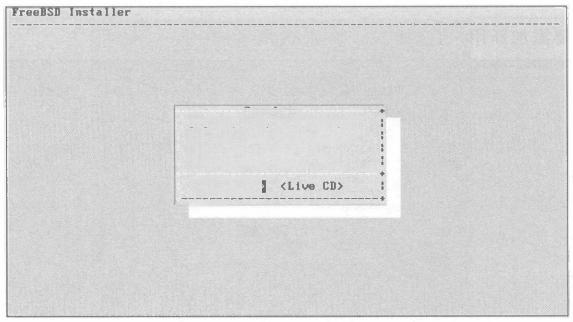
syste te nakg any fimal nanual

<Yes >

Before exiting the installer,Hould

图 3 0

安装成功，选择重启，按回车键继续，如图31所示。



Complete

Installation of FreeBSD

conplete!Hould you like

to reboot into the

installed systen nou?

BR2Doot

图 3 1

到此我们就在计算机中成功地安装了FreeBSD, 你可以启动运行一下此系统，熟悉一下

它的界面环境，实践一些基本的命令。

策划编辑 王丹丹 刘春棠

责任编辑 王丹丹

封面设计

**工业界同行力** **荐**

随着云计算、物联网、移动互联网等技术的发展，大数据将成为具有深远影响的技术变革力量，人类从IT(Infor- mation Technology)时代走向DT(Data Technology)时代，云计算+大数据正在替代硬件+软件成为新的方向，在此基 础上产生的“数据经济”将是全球产业升级的新方向。本书从大数据概念、技术到应用研究的角度详细阐述了大数据技 术方案以及落地的研究方法和实践经验，对于很多用户尤其是研究人员能够在具体场景中快速利用大数据技术分析、挖 掘出应用价值具有实际的指导意义。

**阿里数据经济研究中心常务副主任兼秘书长** **潘永花**

本书作者从行业应用的角度切入，分别介绍了大数据在科学计算、生物医药、电子商务、社交网络、数据挖掘和能 源系统等上面的相关应用，文章理论充分，逻辑严谨，给我们大数据应用有很好的启示。同时，本书在FreeBSD 的环境 之下，通过详尽的实验，对所用环境和案例进行了实例演练，让读者能够轻松入手搭建自己的数据平台，尤其是UFS2文 件系统下，高性能的读取对大数据计算有较大的性能提升。对大数据从业者，非常值得一读。

**腾讯企业云架构师潘晓东**

大数据带来的信息风暴正在变革我们的生活、工作和思维，大数据开启了一次重大的时代转型。大数据浪潮的汹涌 来袭，与互联网的发明一样，这绝不仅仅是信息技术领域的革命，更是在全球范围加速企业创新、引领社会变革的利 器。本书以各行业中实际案例来细解数据创新给公民、政府、社会带来的种种挑战和变革。工欲善其事，必先利其器， 结合本书中的工具搭建、算法运用，让我们在大数据的汪洋中开始探求新知吧!

**百度云服务架构师董月照**

本书重新诠释了大数据的定义，简洁而又通俗易懂。书中对大数据分析中常用的软件介绍得很全面，几乎囊括了所 有主流的解决方案，同时对这些软件的环境搭建介绍非常详细，通过实际操作一步步再现搭建过程，配合适当的图片说 明，就算之前没有基础也能很快搭建。后半部分的大数据行业应用场景的介绍让抽象的概念实例化，其中介绍的各种算 法更是做研究不可多得的资源。相信这本书对想要了解大数据并做深入学术研究的人会非常有帮助。

**北京昆仑万维科技股份有限公司资深架构师苗雨顺**

数据的产生，不局限于特定的领域，不依赖于特殊的载体，它就这样不停地产生、堆叠、以海量，多样的姿态呈现 在我们面前。这本书结合世界上最先进的大数据研究成果和最具代表性的应用案例，化繁为简，为读者解开了大数据研 究中的种种困惑，从而更科学有效地完成对大数据的分析和挖掘。

**51talk** **大数据专家、资深架构师** **郝伟瑞**

感知即存在，人类诞生以来，对于数据的处理从未止步。信息时代，人类本身也将以大数据的形式而存在，不了解 大数据、掌握处理大数据方法的人，等于是新时代的原始人!本书深入浅出，避开深奥烦琐的概念，从工程实践出发， 一步步将人带入到大数据开发的最前沿，对大数据在一些行业的应用进行了深入讲解。同时用简单明了的语言，图文并 茂，对理论进行了阐释。本书适合广大初学者、本科生以及各行业对大数据有兴趣的的从业者。

**央视网** **高级数据分析师** **孙泉**

本书官方公众号：

大数据技术及行业应用

本书官方网站：

<http://dxmall.com.cn:2016>