Understanding Big Data

Analytics for Enterprise Class Hadoop and Streaming Data

理解大数据

企业级 Hadoop 和流数据分析

- 了解 IBM 如何增强企业级 Hadoop 可扩展性和可靠性
- 洞察 IBM 唯一的移动和静止大数据分析平台
- 学习大数据用例和解决方案的技巧和诀窍
- 快速了解 Hadoop

CHRIS EATON
DIRK DEROOS
TOM DEUTSCH
GEORGE LAPIS
PAUL ZIKOPOULOS

理解大数据

关于作者

Paul C. Zikopoulos，文学士，MBA，是 IBM Software Group 信息管理部门的技术主管，同时还领导 World Wide Database Competitive 和 Big Data SWAT 团队。Paul 是一名国际知名的获奖作家和演说家，有着超过 18 年的信息管理经验。Paul 撰写过 320 篇杂志文章和 14 本关于数据库技术的书，包括《DB2 pureScale：Risk Free Agile Scaling》（McGraw-Hill，2010 年）；《Break Free with DB2 9.7：A Tour of Cost-Slashing New Features》（McGraw-Hill，2010 年）；《Information on Demand：Introduction to DB2 9.5 New Features》（McGraw- Hill，2007 年）；《DB2 Fundamentals Certification for Dummies》（For Dummies，2001 年）；《DB2 for Windows for Dummies》（For Dummies，2001 年）等。Paul 是一名 DB2 认证的高级技术专家（DRDA 和 Clusters）和解决方案专家（BI 和 DBA）。在空闲时间，他喜欢各种各样的运动，如带他的狗狗 Chachi 一起跑步；练习综合格斗；试图弄清楚为什么他的高尔夫差点会莫名其妙的上升，以及尝试理解他女儿 Chloë 的世界。您可以通过 paulz_ibm@msn.com 联系他。另外，也可以关注他的微博 @BigData_paul，随时了解大数据发展。

Chris Eaton，理学士，IBM 信息管理产品全球技术专家，主要关注数据库技术、大数据和工作负载优化。Chris 在 Linux、UNIX 和 Windows 平台下的 DB2 领域有 19 年的经验,担任过许多角色，从技术支持到开发，再到产品管理。Chris 的整个职业生涯都在聆听客户的意见，并致力于使 DB2 成为更好的产品。他是几本关于数据管理空间的图书作者，包括《The High Availability Guide to DB2》（IBM Press，2004 年），《IBM DB2 9 New Features》（McGraw-Hill，2007 年）和《Break Free with DB2 9.7：A Tour of Cost-Slashing New Features》（McGraw-Hill，2010 年）。Chris 还是一位国际知名的获奖演说家，常在全球数据管理会议中发表演讲，他在 IT Toolbox 上有最受欢迎的 DB2 博客，http://it.toolbox.com/blogs/db2luw。

Dirk deRoos，理学士，文学士，IBM 全球技术销售团队一员，专攻 IBM 大数据平台。Dirk 于 11 年前加入 IBM，此前在 Toronto DB2 Development 实验室工作，担任信息架构师。Dirk 拥有 New Brunswic 大学计算机科学学士学位，同时还取得了文学士学位（英语成绩优异）。

Thomas Deutsch，文学士，MBA，IBM 大数据业务项目主管。Tom 在过去的几年中致力于帮助客户使用 Apache Hadoop、识别架构机遇、管理多客户参与的早期阶段项目。他在从 IBM Research 到 IBM Software Group 的基于 Hadoop 的技术转换中扮演着结构性角色，他同时参与 IBM Research Big Data 研究以及从研究到商业产品的过渡。在此之前，Tom 在 CTO 办公室信息管理部门工作。在此工作中，Tom 所在的团队致力于新型技术，并帮助客户采用 IBM 创新的 Enterprise Mashups 和云产品。Tom 通过 FileNet 收购进入 IBM，在 FileNet 他负责 FileNet 旗舰 Content

Management 产品和先锋 FileNet 产品创新以及其他 IBM 软件，包括 Lotus 和 InfoSphere。他在业内有着 20 多年的经验，同时是两个初创企业的老员工，Tom 是技术、策略以及今天企业面临的业务信息管理问题方面专家。Tom 从纽约 Fordham 大学获得学士学位，从 Maryland 大学获得 MBA 学位。

George Lapis，MS CS，IBM 硅谷研究和开发实验室大数据解决方案架构师。他在数据库软件领域有着超过 30 年的经验。他是硅谷的 IBM Almaden 研究中心 R* 和 Starburst 研究项目的创办会员，也是编译器开发团队成员之一，进行几个 DB2 版本的研发。他的专长主要包括编译器技术和实现。大约 10 年之前，George 从研究转到开发，他领导当前实验室的编译器开发团队，主要从事 DB2 的 SQL/XML 和 XQuery 功能的研发。George 花费几年时间在 Optim Database 工具集客户实现方面，最近是 IBM 大数据业务。George 最近的角色是领导 IBM's InfoSphere BigInsights 平台的工具开发团队。他也是几本数据库专刊的合著者，也发表过许多文章。他还是一名认证的 DB2 DBA 和 Hadoop 管理员。

关于技术编辑

Steven Sit，理学士，MS，IBM 硅谷研究和开发实验室的一名编程主管，IBM 大数据平台在该实验室开发和设计。Steven 和他的开发团队帮助 IBM 客户和合作伙伴评估、原型化和实现大数据解决方案，以及构建大数据开发模式。在过去的 17 年中，Steven 在许多 IBM 项目中担任重要职务，包括业务智能、数据库加工和内容搜索。Steven 拥有 Western Ontario 大学计算机科学学士学位和 Syracuse 大学计算机科学硕士学位。

理解大数据

企业级 Hadoop 和流数据分析

Paul C. Zikopoulos
Chris Eaton
Dirk deRoos
Thomas Deutsch
George Lapis

纽约　芝加哥　旧金山
里斯本　伦敦　马德里　墨西哥
米兰　新德里　圣胡安
首尔　新加坡　悉尼　多伦多

理解大数据：企业级 Hadoop 和流数据分析

1 2 3 4 5 6 7 8 9 0 DOC DOC 1 0 9 8 7 6 5 4 3 2 1

ISBN 978-0-07-179053-6

MHID 0-07-179053-5

策划编辑
Paul Carlstroem

编辑企划
Patty Mon

项目经理
Sheena Uprety, Cenveo Publisher Services

采购协调员
Stephanie Evans

技术编辑
Lisa Theobald

校对
Paul Tyler

生产主管
George Anderson

排版
Cenveo Publisher Services

插图
Cenveo Publisher Services

封面设计总监
Jeff Weeks

这是我在 IBM 工作 18 年中编著的第五本图书——难以置信时间过得如此之快，信息管理技术不仅已成为我的职业，还在一定程度上成为了我的爱好（插入 Chloe 在两年前学会了通用的"失败者"姿势后阅读本书的照片）。

我的一生致力于向人们贡献我的图书。实际上我希望将本书贡献给我生命中的、在 2011 年 8 月 12 日跨过 100 岁生日的公司：IBM。在这个人才流动频繁的时代，美国劳动部表明普通的学习者到 38 岁时将参加过 10 到 14 个工作，1/4 的员工与其雇主相处的时间不到 1 年，1/2 的员工与其雇主相处的时间不超过 5 年。人们时常询问我在 IBM 的 18 年任期的情况，都觉得难以理解我们这一代。在 IBM 的 18 年里，我很荣幸地学习和参与了最新的技术、营销、销售、技术销售、写作、实用性设计、开发、合作伙伴计划、渠道、教育、支持、服务、公开演讲、竞争分析，并且不断在学习。IBM 始终是一个帮助渴求进步的人实现卓越成就和提供机会的地方，我就是一位永不满足、渴求进步的人。IBM 大力提倡向他人学习——我常常想知道其他人是否希望借助我拥有的这样指导团队（Martin Wildberger、Bob Piciano、Dale Rebhorn 和 Alyse Passarelli）崭露头角。感谢 IBM 提供永无止境的机会和学习经验。

最后，我将本书献给我的两个女儿，她们的天真总是温暖着我的灵魂：Grace Madeleine Zikopoulos 和 Chloë Alyse Zikopoulos。

—Paul Zikopoulos

这是我创作的第四本图书，每次我都会将我的图书献给我的妻子和家庭。本书也不例外，因为正是他们的支持才让本书得以诞生，将无数的个人时间用于写书的任何人都可以作证。

献给我的妻子 Teresa，她对我所做的所有事情都全力支持，包括写书的疯狂想法。她非常清楚写一本书需要多少时间，因为她自己就是一位作家，但在我告诉她我打算编写本书时，她仍然没有任何怨言（您就是一位圣人）。再次献给 Riley 和 Sophia，她们现在的年龄完全能阅读我的图书了

（她们并不是真的对我的任何东西感兴趣，因为它们都还不到 10 岁）。爸爸已完成了这本书，让我们出去尽情玩耍吧。

—Chris Eaton

感谢 Sandra、Erik 和 Anna 对我的支持，为我提供时间完成此工作。另外，感谢 Paul 让本书得以诞生并给了我参与编写的机会。

—Dirk deRoos

我想感谢对我提供大力支持的妻子并将本书献给 Lauren 和 William，既然本书已出版，我可以再次带他们去迪斯尼乐园了。我还要感谢 Anant Jhingran 对我的指导和为我提供这一机会。

—Thomas Deutsch

“如果您热爱您所做的事，您绝不会虎头蛇尾。”我将本书献给在 IBM 与我合作多年的所有同事，是他们陪伴我学习和成长，使这一想法成为现实。

—George Lapis

感谢 IBM 大数据研发部门的同事，是他们创造了供我每天研究的令人激动的技术。我还想感谢 Paul 为我提供参与编写本书的机会。最后也是最重要的，感谢我的妻子 Amy 和双胞胎孩子 Tiffany 和 Ronald，感谢您们为我所做的一切，感谢您们带来的欢乐，感谢您们支持我在本书上投入这么多时间。

—Steven Sit

概　　览

第 I 部分
大数据：业务视角

第 II 部分
大数据：技术视角

目　录

第 II 部分
大数据：技术视角

序言

来自 Rob Thomas 的高管信

有一个古老的故事，讲许多年前两个在铁路上工作的人。一天正午他们在铺设轨道时，一个人开着车路过这里并摇下车窗（没有大到让冷气跑出，但足以听到谈话）。他喊道，“Tom，是您吗？”Tom（其中一个铁路工人）回答道，“Chris，很高兴见到您！我们一定有 20 年没见了……您过得怎么样？”他们继续交谈了一会，最后 Chris 开车离开。当他离开时，另一位工人对 Tom 说，“我知道他是这条铁路的负责人，他的财产接近 10 亿美元。您怎么认识他的？”Tom 回答道，“20 年前的今天，我和 Chris 都在铁路上铺设轨道。我们之间的唯一区别在于我是为了 1.25 美元/小时而工作，而他是为了铁路而工作。”

*****

设想。抱负。雄心。这些都是区别为回报而工作和为改变世界而工作的人的特征。大数据时代的来临是科技世界的每个人决定站在哪一队的一个机会，因为这个时代将为科技公司和个人带来自互联网诞生以来最大的机会。

让我们回头看看本世纪以来，科技世界发生了哪些变化：

- 全球 80% 的信息是非结构化的。
- 非结构化信息正在以 15 倍于结构化信息的速率增长。
- 原始的计算能力正在以极高的速率增长，以至于如今现成的商用机器已开始展现出 5 年前的超级计算机的能力。
- 对信息的访问已民主化：它可供（或者应该供）所有人使用。

这是一个新的标准。单单这些方面就需要更改我们解决信息问题的方法。这是否意味着我们过去 10 年的投资将付诸东流或无关紧要？当然不是！我们仍需要关系型数据存储和仓库，而且占地面积将继续扩大。但是，我们需要通过允许企业从大数据时代获益的技术来改进这些传统方法。

谁能提供适合这一新标准的平台，谁就能领导大数据时代，这个平台包含探索和开发工具集、可视化技术、搜索和发现、原生文本分析、机器学习，以及企业稳定性和安全性等。许多公司都在谈及这一平台，但很少有公司提供。

我在这里是因为我知道我们可以改变科技世界，*这*远远比 1.25 美元/小时令人激动。欢迎来到大数据时代。

Rob Thomas
IBM 副总裁，业务发展部

来自 Anjul Bhambhri 的高管信

上世纪 70 年代，关系型数据库系统的第一个原型 *System R* 在圣何塞的阿尔马登研究实验室创建。System R 为处理具有关系结构的数据的最常见方式（称为 SQL）播下了种子；您会发现它是 DB2、Oracle、SQL/DS、ALLBASE 和 Non-Stop SQL 等产品开发的一个重要推动因素。加上大型机、中型机和个人台式机的计算能力急速提升，数据库已成为一种收集和存储数据的普遍方式。事实上，它们的增殖导致一个围绕“数据仓库”的学科的创立，目的是用一种统一的方式对来自多个数据库的数据进行轻松的管理和关联。它还导致这些仓库被垂直切片为多个数据集市，从而更快地制定与特定业务线需求紧密关联的决策。在上世纪 90 年代的短时间内，这些发展让 IT 部门成为了每次业务冒险的一个关键竞争优势。诞生了数千个应用——某些应用贯穿多个行业，某些应用特定于某个领域，如购买、快递、运输等。ERP（企业资源规划）、SCM（供应链管理）等代号变得家喻户晓。

到上世纪 90 年代末，一个组织的不同部门不可避免地使用不同的数据管理系统来存储和搜索他们的关键数据，这催生了联合数据库引擎（在 IBM 代号 *Garlic* 下）。然后，2001 年进入了 XML 时代。DB2 pureXML 技术提供了复杂的功能，使用其原生的分层格式存储、处理和管理 XML 数据。尽管 XML 具有灵活的模式和易于移植性等重要优势，但电子邮件的广泛使用、后台内容的累积，以及其他技术产生了对内容管理系统的需求，分析企业中的非结构化和半结构化数据的时代诞生。如今，互联网的诞生与使用多种格式创建和分发内容的全面民主化相结合，导致了所有类型数据的激增。数据现在不仅在数量和种类上可观，而且传输速度也很高。我们在需要的准确时刻收集嵌入在这些海量数据中有用信息的能力使它变得非常激动人心。我们处于另一个演化方向的顶端，通常称为*大数据*。

在 IBM，我们的使命是帮助客户通过技术创新实现他们的业务目标，我们在截至 2011 年的一个世纪中一直在这么做。在最后 50 年，IBM 发明了各种技术并实现了多个平台来满足我们客户不断变化的数据管理挑战。IBM 在 30 多年前就发明了关系型数据库，单单在 IBM 提供的多个产品中（例如 DB2、Informix、Solid DB 等），它就已成为了行业标准。关系型数据库进一步专业化为多维数据仓库，其中包含高度并行化的数据服务器、丰富的专用设备（如 Netezza 或 Smart Analytics System），以及分析和报告工具（如 SPSS 或 Cognos）。

在多个行业和领域（消费商品、金融服务、政府、保险、电信等）中，公司正在评估如何管理其未开发的信息数量、种类和传输速度，目的是找到更好的业务决策制定方式。数据的激增源于各种不同的数据源，如传感器、智能设备、社交媒体、数十亿的互联网和智能电话用户等。这些是以最早期和最原始的形式大量涌入的数据。

希望找到更好的方式（让他们在竞争中脱颖而出）的组织希望利用隐藏在他们周围的海量数据中的丰富信息，以改善其竞争力、效率、洞察力和利润等。这些组织认识到了通过分析来自无数内部和外部来源的所有数据（结构化、半结构化和非结构化）所提供的价值。这就是“大数据”的范畴。尽管许多公司认识到最佳的大数据解决方案需要跨多个涉及许多职位的职能小组来运行，但很少有企业知道如何继续发展。企业的挑战是拥有一个利用这些海量数据来获取及时洞察，同时保留他们现有信息管理投资的一个数据平台。实际上，最佳的大数据解决方案还会帮助组织比以往更好地了解他们的客户。

为了解决这些业务需求，本书将探讨一些有关人们和公司如何解决这个现代问题的重要案例。本书将详细描述驾驭大数据的挑战，提供能带来切实业务收益的大数据解决方案示例。

感谢 Paul、George、Tom 和 Dirk 编写本书。他们是一个优秀的团队，对我们客户的贡献是无价的。他们背后的大数据开发团队在十年间在不断克服各种挑战。我很荣幸与这样一个杰出的团队合作，他们对我们客户的成功充满激情，全身心地投入到他们的工作中，并且在不断创新。与他们合作是我的无限荣耀。

谢谢，希望读者喜欢本书。

Anjul Bhambhri
IBM 副总裁，大数据开发

致谢

总体来讲，我要感谢以下人员，没有他们，本书不可能完成：Shivakumar (Shiv) Vaithyanathan、Roger Rea、Robert Uleman、James R. Giles、Kevin Foster、Ari Valtanen、Asha Marsh、Nagui Halim、Tina Chen、Cindy Saracco、Vijay R. Bommireddipalli、Stewart Tate、Gary Robinson、Rafael Coss、Anshul Dawra、Andrey Balmin、Manny Corniel、Richard Hale、Bruce Brown、Mike Brule、Jing Wei Liu、Atsushi Tsuchiya、Mark Samson、Douglas McGarrie、Wolfgang Nimfuehr、Richard Hennessy、Daniel Dubriwny、我们的研究团队，以及我们企业中牺牲每天个人时间来向您提供 IBM 大数据平台的所有其他人。

Rob Thomas 和 Anjul Bhambhri 值得特别提一下，因为他们的激情具有感染力，感谢您们。

衷心感谢我们了不起的杰出工程师 (DE) Steve Brodsky 和两位 BigInsights 首席高级技术研究员 (STSM)：Shankar Venkataraman 和 Bert Van der Linden，没有他们的贡献和努力，本书不可能完成。IBM 是一个令人惊奇的工作场所，它在您每天工作时会变得不同寻常，而且这些人拥有高超的才智并且乐意分享和帮助我们变得更聪明。我们还要感谢 Steven Sit，他最后担任我们的技术编辑（和兼职研究员，但我们在向他委任这个职位时未能告诉它）。

感谢（尽管我们时常说他们的坏话）Susan Visser 和 Linda Currie 为本书所做的准备工作；想法就是一个想法，但它需要这些人帮助我们从杂乱的信息中获取该想法并立即交到您手中。我们的编辑团队（Sheena Uprety、Patty Mon、Paul Tyler 和 Lisa Theobald）都在幕后发挥着重要作用，感谢他们。感谢我们的 McGraw-Hill 领导 Paul Carlstroem——我们希望与您合作是有理由的（顺便说一下，剩下的一星期假期迟早会用上！）。

最后，感谢 Linda Snow 放弃观看费城鹰队球赛的宝贵机会，感谢 Wendy Lucas 百忙之中抽时间审阅本书并让我们保持正轨。您们二位是优秀的同事，我们的客户很幸运有您们在现场，帮助他们取得成功，您们的激情感染着我们整个业务团队。

关于本书

本书的创作团队精通传统的数据库技术，而且尽管我们在 IBM 具有不同的背景和经验，但我们都认可一点：谈到信息技术，大数据无疑是一个拐点：简言之，大数据*是*一笔大交易！事实上，大数据未来将改变您的做事方式、您获取洞察的方式和您的决策方式（这一改变不会取代如今的做事方式，而是一种高价值且众望所归的扩展）。

认识到这个拐点，我们决定在我们最近的时间里深入研究大数据技术，发现本书是让您快速了解它（如果您对它还不熟悉）的不错方式。我们希望向您展示 IBM 如何用户独特的方式支持开源大数据技术（如 Hadoop），并将它扩展到一个企业级的*大数据平台*中。IBM 大数据平台使用 Hadoop 作为其核心（Apache Hadoop 代码没有分支，BigInsights 始终维持着与 Hadoop 的向后兼容性），并将其与能够理解该平台所带来优势的且富有远见的技术领导者所提供的企业功能相结合。IBM 将其丰富的文本分析和机器学习知识产权融入到这样一个平台中，使用一个经过行业试用、测试过真正的企业级文件系统来加固它，提供企业整合、安全等功能。我们肯定您能想像出存在的可能性。IBM 的目标不是让您获得一个可以运行的 Hadoop 集群——我们会在此过程中这么做；它的目标是为您提供一种新方式来获得您之前（也就是在像 Hadoop 这样的技术与像 IBM 这样的分析领导相结合之前）无法轻松利用的海量数据的洞察力。简言之，IBM 的目标是帮助您应对您的分析挑战，为您提供一个平台来创建端到端解决方案。

当然，一个平台越容易使用，获得的投资回报 (ROI) 就越高。查看 IBM 的大数据平台，您可以看到 IBM 通过 Hadoop 压平了分析时间曲线的所有区域。我们可将其与我们如今驾驶的汽车进行类比。在一方面，手动驾驶可带来很多好处（省油、发动机制动和加速），但需要学习更复杂的技术（想想您第一次使用档位）。在另一方面，自动驾驶不会在您需要时提供细粒度的控制，但操作起来要轻松很多。IBM 的大数据平台将本身比作一种类似保时捷的双离合变速箱驾驶——您可在自动模式下使用它来快速对移动数据和静止数据执行文本分析，也可以在需要时接管控制权并扩展或执行您自己的分析来提供本地化功能。无论如何，IBM 都将让您比任何人更快地实现最终目标。

当 IBM 多年前向世界介绍智慧地球中存在的可能性时，该公司认识到世界已变得*物联化*。晶体管已成为数字时代的基本要素。如今，一辆普通的汽车中包含超过 100 万行代码，有 300 万行代码在跟踪您登记的包裹（鉴于这些成就，很难相信我们的包裹仍然常常丢失）；最新的 Airbus 飞机的运行涉及到超过 10 亿行代码。

很简单（且很令人惊讶），我们现在生活的世界里每人拥有超过 10 亿个晶体管，每个晶体管的成本为千万分之一美分；这个世界拥有超过 40 亿个移动电话用户，在两年内全球会产生大约 300 亿个射频识别 (RFID) 标记。这些传感器都在整个生态系统（供应链、医疗设施、网络、城市、自然系统[如水路]等）中生成数据；一些具有井然有序的数据结构，一些没有。这些物联化的设备的一个共同点是它们都在生成数据，而且该数据具有一种机会成本。不幸的是，由于它的大量性和不统一性，以及与它相关的成本，许多数据被简单地丢弃而没有保存任何有意义的时间量，由于缺乏从中获取价值的有效机制而被指定为“噪音”。

我们从物联化中可自然地联想到，智慧地球也是*互联化的*。当然，有将近 20 亿人在使用互联网，但请想想所有这些物联化的设备都拥有彼此通信的能力。将此情景延伸到一万亿个互联且智能的对象构成的景象（从桥梁、汽车、设备、相机、智能电话、道路、管道、家畜，甚至奶瓶），您会发现：所有这些数据生成和度量设备的交互所生成的信息量是前所未有的，但挑战和潜在的机会也是前所未有的。

最后，我们的智慧地球已变得*智能化*。新的计算模型可处理最终用户设备、传感器和传动装置的激增，将它们连接回后端系统。与高级分析相结合，正确的平台可将海量的数据转换为智能，而智能可进一步转换为行动，将我们的系统转化为智能的流程。这意味着，全球的数字和物理基础架构融合在一起了。计算能力可在我们传统上认为不是计算机的设备中找到，这里面蕴含着与全球分享您对任何事物的想法的自由机会。诚然，几乎任何事物（任何大型或小型组织中的任何人、对象、流程或服务）都可变成数字感知的和连网的。拥有如此丰富的技术和网络，我们必须找到经济高效的方式来从这些不断累积的数据中获取洞察力。

多年以前，IBM 向业务部门和领导者介绍了智慧地球：这个方向性的领导思维重新定义了我们考虑技术和它的问题解决能力方式。可以看到 IBM 在定义智慧地球时做出了许多预测，因为所有这些原理似乎都预示着对大数据平台的需求。

大数据具有许多用例，我们的猜测是在不久的将来发现它成为一种普遍的数据分析技术。如果您尝试了解一种品牌态度，最终会找到一种经济高效且强大的框架来度量文化衰退速率、观点等。病毒营销已不是什么新鲜事物。毕竟，它最早的一位实施者是 Pyotr Smirnov（不错，就是那个发明喝伏特加的人）。Smirnov 创立了木炭过滤法，为了推广它的思想，他雇人在每个场所喝他的伏特加，并大声品评它的口味和背后的技术。当然，智慧地球将病毒营销的应用提升到了一个全新的高度，大数据平台提供了一个革命性的信息管理平台，允许您深入洞察它的高效性。

大数据技术可应用于日志分析，以获得您的业务基础架构支撑技术的重要洞察，避免由于我们所谓的*数据废气*太多而丢弃日志。如果您的平台为您提供了轻松将这些有用数据分类为噪音和信号的能力，它将形成一个能保持事态顺利发展且简化的问题解决和预防流程。

在欺诈检测算法和风险建模方面，大数据平台可通过基于越来越多已识别的因果属性的扩展模型、越来越多的历史（用途几乎是无限的）来提供突破性的功能。

本书组织为两部分。“*第 I 部分* – 大数据：业务视角”主要关注大数据的*谁*（它从一个孩子的填充玩具开始——如果这引起了您的好奇心，请阅读本书）、*什么*、*何处*、*为什么和何时*（不算太迟，但是如果您在研究信息管理，您将无法承担更长的延迟）。第 I 部分包含 3 章。

第 1 章探讨定义大数据的 3 个特征：*数量*（数据的增长和运行速率）、*种类*（数据类型，如传感器日志、微博，想想 Twitter 和 Facebook 等）和*速度*（数据从来源传输到您企业的速度）。在 IBM 进行的大数据讨论中，您将看到在多个地方使用了这 3 个术语，所以我们在本书中和演讲活动中经常将它们称为“*3 个 V*”或“V^3”。有了对大数据特征的可靠定义，您就会理解本书剩余部分所列技术的概念、用例和使用理由。例如，想想普通的一天，重点关注一位员工驱车到 IBM 实验室的（大约）30 分钟：在完成这趟旅程的时间内，我们已生成和经历了不计其数的大数据事件。

从您将智能电话从皮套中拿出（是的，这是为您的电话记录的一个事件）到支付养路费，到我们驶过的桥，再到更改 XM 广播电台，到使用媒体印象，到查看电子邮件（当然不是在驾驶时），到刷卡进入办公室，到在一个有趣的 Facebook 帖子上按下“*顶*”，我们一直是大数据 V^3 的一部分。顺便说一下，我们前面已暗示过，您无需呼吸即可生成 V^3 数据。交通系统、桥梁、飞机上的引擎、您的卫星接收器、天气传感器、您的工作 ID 卡等都在生成数据。

在第 2 章中，我们列出一些流行的问题领域和适合大数据技术的部署模式。我们可能无法涵盖所有可能的使用模式，但我们将分享我们在本节前面看到和暗示的一些体验。您将找到一个有关大数据机会反复出现的主题——越来越多的在以前无法轻松分析的数据。此外，

我们将对比和比较大数据解决方案与每个 IT 商店都有的传统仓库解决方案。我们在这里并在本书中经常表明：大数据为现有的分析系统提供了补充，它不会取代它们（在本章中，我们将提供一个不错的类比，它应该能生动地说明我们的意思）。

第 3 章仍未介绍技术方面的内容，只是探讨为什么我们认为 IBM 的大数据平台是目前的最佳解决方案（不错，我们为 IBM 工作，但请阅读本章，它很有说服力！）。如果稍微考虑考虑大数据，您会认识到它不仅仅关乎使用 Hadoop（提供了大数据引擎的重要开源技术）正常运行并使用一个工具集对它进行经营管理。考虑这样一个事实：我们不能仅考虑单个对购买、管理和安装技术充满兴奋的客户。我们的客户对他们的技术允许他们探索收益的机会充满激动；我们的客户拥有他们希望描绘的愿景，我们将帮助您转变为 Claude Monet。IBM 不仅帮助您缩短让大数据解决方案正常运行所需的时间，而且 IBM 在此领域有一款产品的事实意味着它带来了一个全新的平台。例如，如果有一个与 IBM 所表达含义相同的概念，那就是*企业级*。IBM 理解容错、高可用性、安全性、治理和健壮性。所以当您从开源的大数据 Hadoop 产品回头看时，您将看到 IBM 独特地定位于帮助企业使它坚不可摧。但 BigInsights 所做的不仅仅是让 Hadoop 企业版可靠和可扩展，它让存储在 Hadoop 中的数据很容易使用，无需配备 Java 编程人员和统计学哲学博士。考虑到 BigInsights 添加了分析工具包、资源管理、压缩、安全性等功能，您实际上就能够获得一个企业级的坚固 Hadoop 平台，并且无需购买零件或自行构建即可快速实现一个解决方案。

回想一下，在序言中我们探讨了大数据技术不是当前技术的替代品——它们只是一种补充。意思很明确：您必须将大数据与企业基础架构的剩余部分*整合*起来，并且还会有*治理*需求。哪些公司比 IBM 更理解数据整合和治理？它是一家全球公司，所以如果您在考虑*语言国有化*，应该会想到 IBM（文本分析平台不是仅适用于基于英语的分析吗？我们希望不是！）想想获得诺贝尔奖的

世界级研究人员、数学家、统计学家等：IBM 有众多这样的人才，其中许多人都在研究大数据问题。Think Watson（因其拥有在 *Jeopardy!* 中获胜的性能而闻名）就是 IBM 能力的一个有力证明。当然，您会希望支持您的大数据平台，谁可以*用 24×7 的方式在全球提供工程师直接支持？*您将对大数据做什么？分析它！*IBM 的数据分析平台家族*（SPSS、Cognos、Smart Analytics Systems、Netezza、文本注释器、语音到文本转换等——过去 5 年 IBM 仅仅在分析功能采购上就花费了 140 多亿美元）提供了每年对其大数据平台进行扩展的绝佳机会。

当然，我们不会忘记提及 IBM 对开源社区做出了多大贡献。IBM 拥有支持开源的悠久传承。各种开源产品（Eclipse、非结构化信息管理架构 (UIMA)、Apache Derby、Lucene、XQuery、SQL 和 Xerces XML 处理器）中使用的集成开发环境 (IDE) 事实标准等只是众多贡献的冰山一角。我们希望明确一点：IBM 致力于将 Hadoop 开源。事实上，Jaql（您将在第 4 章了解它）是 IBM 捐赠给开源 Hadoop 社区的。而且 IBM 正在继续研究其他可能用于 Hadoop 的相关捐赠技术。我们的研发实验室拥有 Hadoop 贡献者，他们与来自 Facebook、LinkedIn 等的其他 Hadoop 贡献者并肩作战。最后，您可能会在任何 Hadoop 论坛上找到我们的开发人员。我们相信 IBM 对开源 Hadoop 的贡献与它围绕企业需求和分析的丰富知识产权与研究成果，提供了一个真正的大数据平台。

“第 II 部分——大数据：技术视角”首先在第 4 章中介绍大数据开源技术的一些基本知识。本章为类似于大数据的开源技术（最常见的就是 *Hadoop*，一个 Apache 顶级项目，它的执行引擎是大数据运动的幕后推手）奠定“基础”。阅读完本章后您不会成为 Hadoop 专家，但您将对 *Pig*、*Hive*、*HDFS*、*MapReduce* 和 *ZooKeeper* 等术语具有基本的理解。

第 5 章是本书中最重要的章之一。这一章介绍将大数据拆分为两个关键区域的概念，似乎只有 IBM 在定义大数据时探讨过这一概念：*移动大数据*和*静止大数据*。在本章中，我们重点介绍大数据天平的静止端和 IBM 的 InfoSphere BigInsights (BigInsights)，它是来自 IBM 的企业级 Hadoop 平台。我们将探讨在第 3 章中提及的 IBM 技术——仅通过技术解释和插图来说明 IBM 如何通过其大数据平台脱颖而出。您将了解 IBM 的 General Parallel File System (GPFS)（与企业类同义）如何扩展，以 GPFS 无共享集群 (SNC) 的形式融入一个 Hadoop 环境中。您将了解 IBM 的 BigInsights 平台如何在一个功能丰富的注释开发环境中包含一个文本分析工具包，使您无需使用 Java 或其他编程语言即可构建或自定义文本注释器。您将了解没有 Hadoop 世界中 GLP 许可担忧的快速数据压缩、特殊的高速数据库连接器技术、机器学习分析、管理工具、一个灵活的工作负载管理器（它提供了比默认 Hadoop 工作负载管理器的功能更丰富的面向业务策略的管理框架）、安全锁定、通过智能调整增强等。阅读本章后，您希望大数据提供商回答的问题或提供的功能将会改变，您将提出可证明您的供应商实际拥有真正大数据平台的问题。我们确信您的大数据旅程需要从一个大数据平台开始——基于世界级、具有企业级安全性和功能技术的强大分析工具。

在第 6 章中，我们将介绍大数据“硬币”的另一面来结束本书：分析移动数据。第 6 章比较详细地介绍 IBM InfoSphere Streams (Streams) 和一些来自真实客户的示例，介绍他们如何使用 Streams 实现更好的业务成果，进行更准确的预测，为其公司赢得竞争优势，甚至改善我们最脆弱方面的健康状况。我们还将详细介绍 Streams 的工作原理（Streams 是一种特殊的流处理语言，用于缩短编写 Streams 应用所花的时间）、如何配置它，以及一个流的组成部分（也就是运算符和适配器）。利用 BigInsights 让 Hadoop 适用于企业的相同方式，我们在本章末尾将详细介绍让 Streams 适用于企业的功能，如高可用性、可伸缩性、易用性以及它与现有基础架构的整合方式。

我们知道您将花一两个小时的宝贵时间来阅读本书，并且相信在您阅读完后，您能恰当地应对面前的大数据机会，更好的理解将确保您拥有正确的大数据平台需求，并建立有关大数据所带来的业务机会和一些可用技术的牢固基础知识。

在我们编写本书时，我们不得不进行一些艰难的取舍，因为它的篇幅有限。这些决策不容易，有时感觉我们在欺骗技术读者而帮助业务读者，有时却相反。最后，我们希望为您提供一条捷径来掌握大数据知识，理解 IBM 让大数据逐渐成为您业务领域中真实事物的独特定位。

行进在大数据旅程中时，您一定会找到在最开始不太希望看到的东西；因为它不是一部史诗电影，我们将在现在和一年之内向您讲述完，所以请让我知道我们哪些做得对不对。我们相信您将发现，大数据技术不仅将成为企业中一个功能丰富的普通存储库，还将成为一个应用平台（类似于 WebSphere）。您会发现，在比以往更加紧密地整合到数据存储位置的富生态系统中，需要使用声明性语言来构建分析应用。您将发现需要提供了特定分析类型的对象类，您将需要一个允许任意重用组件和自定义内容的开发环境。您需要部署这些应用的方法（一个类似于 Blackberry 的 AppWorld 或 Apple 的 AppStore 的概念）、可视化功能等。

您会看到本书的篇幅不是太长（我们从未打算将它写成一部小说），它有 5 位作者。当我们第一次会面时，其中一个人打趣地说他想到的第一件事是编写本书的方式可能就像一次客户拜访：许多 IBM 成员围坐在一张桌子旁。但您知道吗？这正是此公司的强大之处：它能够将涉及数十亿美元的交易、贯穿不同行业且拥有丰富专家技能的经验集中起来。我们的创作团队拥有超过 100 年的集体经验和数千小时的咨询和客户互动经验。我们拥有各种研究、专利、竞争、管理、开发和各种行业领域的经验。我们希望我们这个小组能在本书中与您分享其中一些经验，作为您的大数据旅程起点。

第 I 部分

大数据：业务视角

1

什么是大数据？提示：您的每一天都是它的一部分

我们应该从何处着手编写有关大数据的图书？从一个定义开始如何，因为“大数据”这个术语的使用不太恰当，它暗示着预先存在的数据比较小（其实不然）或者唯一的挑战只是它的大小（大小是挑战之一，但还有其他许多挑战）。简言之，术语“大数据”指无法使用传统流程或工具处理或分析的信息。如今，组织日渐面临着越来越多的大数据挑战。它们能够访问丰富的信息，但不知道如何从中获得价值，因为这些信息以最原始的形式或半结构化或非结构化格式存在，这导致它们甚至不知道这些信息是否值得保留（甚至是它们能否保留它）。一项 IBM 调查发现，如今有超过一半的业务领导认识到他们无法获取完成自己的工作所需的洞察。尽管如今公司有能力存储任何信息并且正在以前所未有的方式生成信息，但这两方面相结合，带来了一个真正的信息挑战。这是一个复杂的谜题：如今的业务人员能获取比以往更多的潜在洞察，但由于这个潜在的数据金矿堆积成山，企业可处理的数据比例正在迅速下降。我们认为，在探讨您可使用大数据完成的所有重要工作，以及 IBM 拥有一个我们相信会让您能取得更大成功的独特端到端平台之前，需要探讨一下大数据的特征以及它如何融入到当前的信息管理领域中。

非常简单，因为世界在不断改变，现在大数据时代已全面来临。通过*物联化*，我们能够感知更多事物，并且如果我们能感知它，我们会尝试存储它（或者至少其中一部分）。由于通信技术上的进步，人们和事物变得愈加*互联化*，并且不是偶尔，而是一直如此。这种互联速率就像一列失去控制的火车。互联化一般称为*机器间互联 (machine-to-machine, M2M)*，它是导致两位数的年均 (YoY) 数据增长率的主角。最后，因为小型集成电路的价格现在非常低廉，所以我们可向几乎任何事物增添*智能化*。

甚至像火车这样平凡的事物也拥有数百个传感器。在火车上，这些传感器跟踪火车遇到的各种条件、各个零部件的状态，以及用于货运跟踪和物流的基于 GPS 的数据。在火车脱轨导致大量伤亡之后，政府制定了制度，要求存储和分析此类数据以预防进一步的灾难。火车也正变得更加智能化：添加了处理器来解读容易磨损的零部件（如轴承）的传感器数据，以识别需要修理的零部件，从而预防它们发生故障并导致进一步的损失（或者更糟的是导致灾难）。但并非只有火车是智能化的，实际上铁路上每隔几英尺就有传感器。而且，整个生态系统都拥有数据存储需求：火车、钢轨、铁路交叉道口传感器、导致钢轨移动的天气模式等。现在还需要跟踪火车的载货量、到站和离站时间，而且您会很快发现您遇到了大数据问题。即使此数据的每一比特都是关系型的（其实不然），它也可能是原始的并具有迥异的格式，这使得在传统的关系型系统中处理它变得不切实际或根本不可能。火车只是一个例子，我们随处都可以看到速度、数量和种类相结合就会产生大数据问题。

为了帮助业务人员通过其智慧地球平台应对这一更改，IBM 创建了一个整体模型。它是一种不同的思维方式，真正认识到了世界现在是物联化、互联化和智能化的。智慧地球技术和方法促进了对全球数据现状的理解和利用，提供了获得前所未有的洞察力的机会和改变做事方式的机会。要构建智慧的地球，关键在于利用所有数据，IBM 大数据平台专为此目的而设计；事实上它是智慧地球计划一个关键的架构部分。

大数据的特征

可用 3 个特征来定义大数据：*数量*、*种类和速度*（如图 1-1 所示）。这些特征相结合，定义了我们在 IBM 所称的“大数据”。他们创造了一种需求，那就是使用一类新功能来改善当今的做事方式，提供对我们现有的知识领域和驾驭其能力的更有效控制。

除了以前可能完成的工作，IBM 大数据平台还为您提供了在上下文中通过庞大容量、极快速度和种类丰富的数据中获得洞察的独特机会。我们明确定义一下这些术语。

够多吗？数据量

如今存储的数据*数量*正在急剧增长。在 2000 年，全球存储了 800,000 PB 的数据。当然，如今创建的大量数据都完全未经分析，这是我们尝试使用 BigInsights 解决的另一个问题。我们预计到 2020 年，这一数字会达到 35 ZB。单单 Twitter 每天就会生成超过 7 TB 的数据，Facebook 为 10 TB，一些企业在一年中每一天的每一小时就会产生数 TB 的数据。

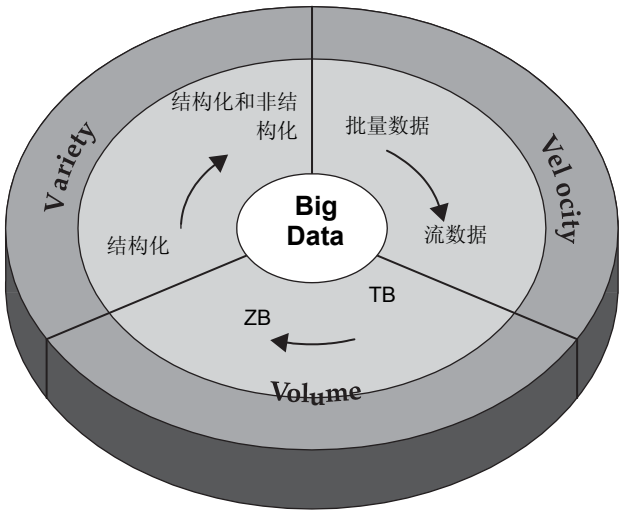

图 1-1 *IBM 按数量、速度和种类或者就是简单的 V*[3] *来定义大数据*

图字：

Variety：种类

Velocity：速度

Volume：数量

现在经常听到一些企业使用存储集群来保存数 PB 的数据。这里我们将列举一些可能的事实：在您阅读本书时这些数据估计已过期，在您阅读完本书后将您的数据增长速率知识告诉朋友和家人时，这些数据会进一步过期。

停下来想想，毫无疑问我们正深陷在数据之中。如果我们可跟踪和记录某个事物，我们通常会这么做。（注意，我们没有提及分析已存储的这些数据，这将是一个大数据主题——对于我们跟踪但未用于决策制定的数据，这是新发现的用途。）我们存储所有事物：环境数据、财务数据、医疗数据、监控数据等。例如，从手机套中拿出您的智能电话会生成一个事件；当您的市郊火车到站开门时，这是一个事件；检票登机，打卡上班、在 iTunes 上购买歌曲、更换电视频道、使用电子收费公路——每一项操作都会生成数据。还需要更多数据？明尼阿波利斯的圣安东尼瀑布大桥（在 2007 年垮塌后被 I35W 密西西比河大桥取代）在重要位置布置了 200 多个嵌入式传感器来提供一个周密的监视系统，它会收集所有类型的详细数据，甚至温度变化和大桥对这一变化的具体反应都可供分析。您一定发现了其中的重点：现在的数据比以往更多，仅仅从个人家庭电脑的 TB 级存储容量即可看出。就在 10 年前，我们知道的超过 1 TB 的数据仓库屈指可数，这足以表明数据量发生了变化。

从术语“大数据”可以看出，组织正面临着过量的数据。不知道如何管理此数据的组织会疲于应对它。但它们有机会使用正确的技术平台，分析几乎所有数据（或者进一步识别对您有用的数据），从而更透彻地理解您的业务、客户和市场。这就导致了所有行业的业务人员如今面临的一个谜题。随着可供企业使用的数据量不断增长，它可处理、理解和分析的数据比例不断下降，因此形成了如图 1-2 中所示的盲区。盲区内是什么？您不知道：它可能是

图 1-2 *如今可供组织使用的数据量不断增长，而它们可分析的数据比例不断下降*
图字：
Data Available ……：可供组织使用的数据
Percent of data an……：组织可处理的数据比例

某种有用的东西，或者可能毫无用处，但“不知道”就是个问题（或者机会，具体取决于您如何看到它）。

有关数据量的对话已从 TB 级别转向 PB 级别，并且不可避免地会转向 ZB 级，而且出于我们将在本章中探讨的原因，所有这些数据都不能存储在传统的系统中。

多样性是生命的调味料

与大数据现象有关的数据量为尝试处理它的数据中心带来了新的挑战：它的*种类*。随着传感器、智能设备以及社交协作技术的激增，企业中的数据也变得更加复杂，因为它不仅包含传统的关系型数据，还包含来自网页、Web 日志文件（包括单击流数据）、搜索索引、社交媒体论坛、电子邮件、文档、主动和被动系统的传感器数据等原始、半结构化和非结构化数据。而且，传统系统可能很难存储和执行必要的分析，以理解这些日志的内容，因为所生成的许多信息并不适合传统的数据库技术。在我们的经验中，尽管一些公司正在朝大数据方向大力发展，但总体而言，大部分公司只是刚开始理解大数据的机会（以及如果不考虑它会有什么风险）。

简言之，*种类*表示所有类型的数据——决策制定和洞察获取流程中的分析需求从传统的结构化数据，到包含原始、半结构化和非结构化数据的一种根本转变。传统的分析平台无法处理多种数据。但是，组织的成功将离不开它从可用的各种类型的数据（同时包括传统和非传统的数据）获取洞察的能力。

当我们回头看看我们的数据库生涯时，有时会羞愧地发现，我们将大部分时间都花在仅 20% 的数据上：格式整齐且符合我们严格模式的关系类型。但事实是，全球 80% 的数据（越来越多的这类数据创造了新的种类和数量的记录）是非结构化的，或者至多是半结构化的。如果查看 Twitter 源，您会在其 JSON 格式中看到结构——但实际的文本不是结构化的，而且理解这些内容会得到回报。视频和图片不能轻松或高效地存储在关系型数据库中，某些事件信息可能动态地更改（如天气模式），它们不太适合严格的模式。要利用大数据机会，企业必须能够分析*所有*类型的数据，包括关系和非关系数据：文本、传感器数据、音频、视频、事务等。

多快才算快？数据的速度

就像我们收集和存储的数据量和种类发生了变化一样，生成和需要处理数据的*速度*也在变化。对速度的传统理解通常考虑数据多快到达并进行存储，及其相关的检索速率。尽管快速管理所有数据没有坏处，并且我们查看的数据量会受数据到达速率的影响，但我们相信速度的概念实际上远远比这些传统的定义更令人信服。

要理解速度，一种思考问题的新方式必须从数据产生的时刻开始。不要将速度的概念限定为与您的数据存储库相关的增长速率，我们建议动态地将此定义应用到数据：数据流动的速度。毕竟我们都同意，如今的企业正在处理 PB 级数据而不是 TB 级数据，而且 RFID 传感器和其他信息流的增加导致了传统系统无法处理的持续的数据流。

有时，领先于您的竞争对手可能意味着在竞争对手之前几秒甚至几微秒识别一个趋势、问题或机会。此外，如今生成的越来越多的数据具有非常短的保存期限，所以如果组织希望在此数据中实现洞察，它们必须能够近乎实时地分析此数据。大数据级的*流计算*是 IBM 推出已久的一个概念，它可用作大数据问题的一种新的解决模式。在传统处理中，您可以考虑对相对静止的数据运行查询：如，查询"向我显示生活在新泽西州洪灾区的所有人"将导致使用一个单一的结果集作为传入的天气模式的警告列表。使用流计算，您可以执行一种类似于持续查询的流程，识别*当前*"在新泽西州洪灾区"的人，但您会得到持续更新的结果，因为来自 GPS 数据的位置信息在实时刷新。

有效处理大数据需要您在数据*变化的过程中*对它的数量和种类执行分析，而不只是在它*静止*后执行分析。考虑从跟踪新生儿健康状况到金融市场的各种示例；在每种情形下，他们都需要以新的方式处理不同数量和速度的数据。大数据的速度特征是让 IBM 成为您最佳大数据平台的一个重要因素。我们将它定义为一种从单纯的批量洞察（Hadoop 风格）到与动态传输的洞察相结合的批量洞察的*内含式*转变，IBM 可能是唯一未将速度局限于数据生成速率（它实际上是数据数量特征的一部分）的供应商。

现在想像这样一种结合式的大数据平台，它可利用两个领域的优点，实时传输洞察，以获得基于新出现数据的进一步研究结果。正如您所想的，我们相信您会跟我们一样，对 IBM 大数据平台所提供的独特主张激动不已。

仓库中的数据和 Hadoop 中的数据（它们不是相对立的）

在我们的经验中，传统仓库是分析来自各种系统的结构化数据，并生成洞察（具有已知且相对稳定的度量指标）的最理想选择。另一方面，我们认为基于 Hadoop 的平台很适合处理半结构化和非结构化数据，以及在需要数据查询流程时。这并不是说，Hadoop 不能用于以原始格式存在的结构化数据；因为它可以，我们将在第 2 章中探讨这一点。

此外，当您考虑数据应该存储在*何处*时，需要理解数据如今的存储*方式*，以及您的持久化选项有哪些特征。考虑将数据存储在传统数据仓库中的体验。通常，此数据会经历严格的审查才能进入仓库。仓库的构建者和使用者总是认为他们在仓库中看到的数据必须具有很高的质量；因此，在准备用于分析之前，会通过清理、扩充、匹配、术语库、元数据、主数据管理、建模和其他服务来整理数据。显然，这可能是一个昂贵的流程。鉴于这一开支，位于仓库中的数据显然不应视为仅具有高价值，它还具有广泛的用途：它将传输到许多位置，将用在数据的准确性至关重要的报告和仪表板中。例如，2002 年推出的 Sarbanes-Oxley (SOX) 法规要求在美国交易的公开上市公司的 CEO 和 CFO 要证明他们的财务报表的准确性（第 302 节，“企业的财务报表责任”）。如果报告的数据不准确或“不真实”，将实施严厉的惩罚（我们指的是存在坐牢的可能）。您认为这些人会查看非原始数据的报告吗？

相反，大数据存储库很少（至少在最初）对注入到仓库中的数据实施全面的质量控制，因为某些具有 Hadoop 用例特征的较新分析方法准备数据时需要很高的成本（我们将在下一章探讨），而且数据不可能像数据仓库数据一样分布。我们可以说，数据仓库数据可得到“公众”的足够信赖，而 Hadoop 数据未得到这样的信赖（*公众*可能表示公司内的广大用户，不适用于外部使用），而且尽管这在未来将可能有所改变，但现在这一体验表明了这些存储库的特征。

我们的经验还表明，在当今的 IT 领域，特定的数据片段是基于所认识到的它们的价值而存储的，因此除了这些预先选择的部分数据外，任何信息将不可用。这与基于 Hadoop 的存储库模式相反，在这种模式下可能会存储完整的业务实体，Tweet、事务、Facebook 帖子等，并且其真实性会得到完整的保留。Hadoop 中的数据可能在目前看来价值不高，或者它的价值未得到量化，但它实际上可能是还未提出的问题的关键所在。IT 部门挑选高价值的数据并执行严格的清理和转换流程，因为它们知道该数据具有*很高的每字节已知价值*（当然这是一种相对的描述）。为什么公司要对数据实施如此多的质量控制流程？当然，因为每字节价值很高，所以业务人员愿意将它存储在成本相对较高的基础架构上，以实现与最终用户社区的交互式且常常公开的互动。CIO 愿意投资对该数据进行清理，以提高其每字节价值。

使用大数据，您应该考虑从相反的视角审视此问题：考虑到目前数据的数量和速度，您绝对无法承担正确清理和记录每部分数据所需的时间和资源，因为这不太经济。而且，您如何知道此大数据是否有价值？您是否要联系 CIO 并要求将资本开支 (CAPEX) 和运营开支 (OPEX) 成本增加到 4 倍，以将仓库的大小扩大到原来的 4 倍？出于此原因，我们喜欢将最初未分析的原始大数据视为拥有*较低的每字节价值*，因此在事实证明它拥有高价值之前，您无法承担仓库调整成本；但是，考虑到庞大的数据量，如果您可分析所有数据，获得有用洞察（进而在市场中获得更高的竞争优势）的潜力将很高。

现在是时候介绍*每计算成本 (cost per compute)* 了，它遵循与每字节价值相同的模式。如果您考虑关注我们之前列出的传统系统中的高质量数据，那么可以得出传统数据仓库中的每计算成本比 Hadoop 的成本（较低）相对较高的结论（这没有问题，因为事实证明它具有较高的每字节价值）。

当然，其他因素可能表明某些数据可能具有很高的价值，但从未添加到仓库中，或者人们希望从仓库中将它转移到更低成本的平台中；无论如何，您可能需要清理 Hadoop 中的部分数据，IBM 可完成此任务（一个重要优势）。例如，非结构化数据无法轻松地存储在仓库中。

诚然，一些仓库在构建时考虑了一个预定义的问题集。尽管这样一个仓库为查询和挖掘提供了一定的自由，但它可能受到架构中的内容（大部分非结构化数据都不在这里）并常常受到一个性能边界（可能是一个硬性的功能/操作限制）的约束。同样，我们将在本书中反复重申，我们没有说 IBM InfoSphere BigInsights 等 Hadoop 平台可取代您的仓库；相反，它只能作为补充。

大数据平台允许您将所有数据存储为其原生的业务对象格式，通过可用组件上的大规模并行性获得其价值。为满足您的交互式导航需求，您可以继续挑选来源，清理该数据，以及将它保留在仓库中。但是可通过分析更多数据（可能甚至在最初似乎毫不相关的数据）来获取更多价值，以绘制所遇问题更可靠的情况。的确，数据可能在 Hadoop 中存在了很长时间，当您发现它的价值时，以及当它的价值得到证明并可持续时，就可以将其迁移到仓库中。

小结

本章最后将使用一个金矿类比来阐述上节的要点，以及在您面前的大数据机会。在“很久以前”（出于某种原因，我们的孩子认为是我们像他们那么大的时期），矿工可实际地*看到*金块或金矿脉；他们能清楚地认识到它的价值，并且在以前发现金矿的位置附近挖掘和筛选，希望发一笔横财。尽管这里有更多黄金（可能位于他们旁边或数英里外的山中），但他们用肉眼看不到，所以这就成了一个赌博游戏。您疯狂地在发现黄金的地方附近挖掘，但您不知道是否会找到黄金。而且尽管历史上有许多淘金热的故事，但没有人会调动数百万人来挖掘每个角落。

相反，如今淘金热的运作方式大不相同。对金矿的挖掘可使用需要巨额资本的设备来执行，用于处理数百万吨无用的泥土。如果要肉眼可看到金矿，通常需要 30 mg/kg (30 ppm) 的矿石品味，也就是说，现在金矿中的大部分黄金是肉眼看不到的。尽管所有黄金（高价值数据）都在整堆泥土（低价值数据）中，但通过使用正确的设备，您可以经济地处理大量泥土并保留您找到的金箔。然后将金箔集中在一起制成金条，存储并记录在安全、受到严密监视、可靠且值得信赖的地方。

这就是大数据的真正含义。您无法承担在传统流程中对所有可用数据进行筛选的成本，有太多的数据具有太少的已知价值和太高的冒险成本。IBM 大数据平台为您提供了一种方式来经济地存储和处理所有数据，找到有价值且值得利用的信息。而且，因为我们探讨的是对静止和移动数据的分析，所以您不仅可通过 IBM 大数据平台充实您可从中获取价值的实际数据，还可以实时、更快地使用和分析这些数据。

2

为什么大数据至关重要？

本章的标题准确说明了我们将介绍的主题：为什么大数据至关重要？我们还将探讨一些真实的客户体验，解释我们如何吸引并帮助客户开发新应用程序和潜在的方法来解决以前困难的（或者甚至无法解决的）挑战。最后，我们将重点介绍急需 IBM 大数据平台（包括 IBM InfoSphere BigInsights (BigInsights) 和 IBM InfoSphere Streams (Streams)）帮助的活动中反复遇到的两种使用模式。

考虑大数据解决方案的时机

术语*大数据*可通过许多不同的方式来解释，这正是我们在第1章中将大数据定义为具有数量、速度和种类 (V^3) 属性特征的原因。注意，大数据解决方案无法取代您现有的仓库解决方案，而且根据我们的经验，任何建议这种取代的供应商可能都没有全面的经验策略或理解您在传统信息管理方面的投资。

在列出一些有关何时使用大数据技术的考虑因素之前，我们认为本节最好首先给出我们希望您记住的两条重要的大数据原则，也就是：

- 大数据解决方案是分析来自各种不同来源的原始结构化数据、半结构化和非结构化数据的理想选择。
- 需要分析所有或大部分数据而不只是一个数据抽样时，或者对一个数据抽样执行分析没有对更大的数据集进行分析更有效时，大数据解决方案是理想的选择。
- 大数据解决方案是在未预先确定数据的业务度量指标时，执行迭代式和探索式分析的理想选择。

谈到使用大数据技术解决信息管理挑战，我们建议您考虑以下问题：

- 传统分析模式的反向模式是否适合您遇到的业务任务？换句话说，您能否找到一个大数据平台可为您当前的分析工具提供补充并实现与现有解决方案的协调一致，以实现更好的业务成果？

  例如，通常放在分析仓库中的数据必须经过清理、记录并且值得信赖，才能规范地放在严格的仓库模式中（当然，如果无法用传统的行和列格式存储它，它在大部分情况下甚至无法放在仓库中）。相反，大数据解决方案不仅会利用通常不适合传统仓库环境且数量庞大的数据，而且它将放弃数据的一些形式和“严格性”。好处是您可保留数据的真实性并能够访问海量的信息，在对信息采取您熟悉的适当行动*之前*探索和发现业务洞察；该数据可包含在一个循环的系统中，以充实仓库中的模型。

- 对于不能使用传统关系型数据库方法处理手头问题的方式来解决的信息挑战，大数据非常适合。

您一定要认识到传统数据库技术*是*整体分析解决方案中一个重要且相关的部分。事实上，它们在与您的大数据平台结合使用时会变得更加重要。

此处一个不错的类比就是您的左手和右手；每只手对于手头的任务具有各自的优势并进行了优化。例如，如果您打过棒球，您就知道一只手更擅长抛球，另一只手更擅长接球。就像这样一种情形，每只手可以尝试执行它天生不适合的任务，但会非常笨拙（试一下，录制下您尝试的视频，您就会明白我们的意思）。而且，您不会看到棒球运动员使用一只手接球，停下来，丢掉他们的手套，然后使用同一只手抛球。棒球运动员的左手和右手会协同实现最佳的结果。这是传统数据库和大数据技术的一个简单类比：没有这两个重要实体的协同工作，您的信息平台不会得到进一步发展，因为就像您协调双手来抛接棒球一样，一个团结一致的分析生态系统才能实现最佳的结果。有些类型的问题不是本来就属于传统数据库的，至少在最初不是。而且我们不确定我们是否希望将一些数据放在仓库中，因为可能我们不知道它是否拥有较高的价值、它是否是非结构化的，或者它是否太庞大了。在许多情形下，我们在投入精力和金钱来将数据放在仓库之后，才能发现数据每字节的价值；但我们希望在投资之前确保该数据值得保存并拥有较高的每字节价值。

大数据用例：大数据部署模式

本章旨在帮助您理解为什么大数据至关重要。我们可以列出有关大数据的大量媒体引证、一夜暴富的商人并长篇累牍地阐述，但这会使它更像一份营销材料，而不是一个拐点。我们认为解释大数据为什么至关重要的最佳方式是，与您分享我们与客户在使用 IBM 大数据平台时面临的使用模式（和他们解决的问题）相关的真实客户经验。这些模式代表着绝佳的大数据机会（以前不容易解决的业务问题），并帮助您理解大数据如何帮助您（或者如果您稍有疏忽，它如何帮助您的竞争对手让您失去竞争优势）。

在我们的经验中，IBM BigInsights 平台（它支持 Hadoop 并使用一些丰富功能扩展了它，我们在本书后面将会介绍）

适合我们所服务的每个行业。我们可以在本章中介绍数百个用例，但限于篇幅，我们选择了6个最典型的用例进行探讨，这些用例代表了最常见的使用模式。尽管对使用模式的解释可能是特定于行业的，但许多使用模式都适用于多个行业（这正是我们选择它的原因）。您将在这里探讨的所有使用模式中找到一个共同的特征：它们都涉及到一种新的做事方式，这种方式现在更加实用且最终使得应用大数据技术成为可能。

用于 IT 部门的 IT 日志分析

日志分析是所创立的大数据项目的一个常见用例。我们喜欢将IT 解决方案操作所生成的所有日志和跟踪数据称为*数据废气 (data exhaust)*。企业拥有大量数据废气，如果仅在紧急情况下存在一两个小时或一两天，然后就被简单地清除，那么它几乎就是污染物。为什么？因为我们认为数据废气具有浓缩的价值，IT 部门需要找到一种方式来存储它并从中提取价值。一些来自数据废气的价值显而易见，并已转换为能够记录网站操作（如每个手势、单击和移动）的增值点击流数据。

一些数据废气的价值不那么明显。在多伦多（加拿大安大略湖）的 DB2 开发实验室，工程师使用 BigInsights 执行性能优化分析，获得了极高的价值。例如，考虑一个大型、集群化的、基于事务的数据库系统，尝试主动找到跨不同服务器的关联活动中何处可能稍加优化。从大量的跨服务器堆栈跟踪日志中寻找性能优化点犹如大海捞针。尝试找到每个核心堆栈跟踪信息的数十 GB 数据之间的关联，这确实是一项艰巨的任务，但大数据平台可以帮助您识别以前未报告的性能优化调试区域。

简言之，IT 部门需要可自由使用的日志，但如今他们无法以经济高效的方式存储足够的日志并分析它们，所以日志通常仅在紧急情况下保留并尽快丢弃。IT 部门在日志中保留大量数据的另一个目的是查找少见的问题。最常见的问题常常已知并很容易处理，但“偶尔”发生的问题通常更加难以诊断和防止再次发生。

我们认为 IT 部门渴望（或应该渴望）日志长期保存。我们还认为业务和 IT 部门都知道这些日志中存在价值，因此我们也看到各个业务部门都会对日志进行复制，但最终却漫无目的地保留，并形成了许多在团队之间差别巨大的非标准（或重复性）分析系统。这不仅会产生很高的费用（需要存储更多的聚合数据——通常在昂贵的系统中），而且因为只有一些数据片段可用，所以根据这些保留时期和视图极其有限的信息来确定整体趋势和问题几乎是不可能的。

如今这些日志历史可以保留，但在大部分情况下，一次仅保留几天或几星期，因为传统系统难以存储如此巨大的数据，这无疑会让您无法根据这些有限保留时期的数据来确定趋势和问题。但除了它的大容量性质，日志分析是一个大数据问题还有更多原因。这些日志是半结构化和原始的，所以它们并不总是适合传统数据库处理。此外，由于硬件和软件升级，日志格式在不断变化，所以它们不能禁锢在严格、僵硬的分析模式中。最后，您不仅需要对长期存在的日志执行分析来确定趋势和模式、查明故障，还需要确保分析是对所有数据执行的。

日志分析实际上是 IBM 在与众多公司（包括一些大型的金融服务领域 (FSS) 公司）合作之后建立的一种模式。从那以来，我们已在许多客户中看到此用例。出于此原因，我们将此模式称为*用于 IT 的 IT*。如果您可以联系起来，我们就无需再说什么了。如果您不熟悉此使用模式，可能会想谁对这种*用于 IT 的 IT*大数据解决方案感兴趣，您应该知道这是一家组织内的内部用例。例如，非 IT 的企业常常希望以一种服务部门的形式向他们提供此数据。*用于 IT 的 IT* 方案非常适合具有较大数据中心的组织，尤其是它相对比较复杂时。例如，具有大量移动部件的面向服务的架构 (SOA) 应用、连锁的数据中心等都存在着本节列出的相同问题。

客户正在尝试获得其客户正在设法更清楚的了解其系统的运行状况，故障是何时、如何发生的。例如，我们合作过的一家金融公司将确定应用程序如何偏离轨道的传统方式亲切地称为“打地鼠”。当他们高度基于 SOA 的环境中出现故障时，很难确定发生了什么，因为某个事务的处理涉及到 20 多个系统，这让 IT 部门很难准确跟踪出现故障的原因和地点（我们都看过这样的电影：每个人都在作战指挥室来回走动并表态“我没有做过！”——电影里还有一种场景，那就是每个人都指着您。）我们帮助此客户每天利用一个大数据平台分析大于 1TB 的日志数据，反应时间不超过 5 分钟。如今，该客户能够准确解密整个堆栈中每个事务上发生的事情。当客户的一个事务（来自他们的移动或网络银行平台）出现问题时，它们能够准确了解是哪个组件在何处导致了该问题。当然，正如您所想的，这为他们节省了大量解决问题的时间，无需与事务一起执行额外的监控，因为此时使用已生成的*数据废气*作为分析来源。但此用例不仅仅包含问题检测：它们能够开发一个知识库（或资料），目的是更好地预测和理解故障之间的交互，它们的服务部门可在发生特定问题时生成最佳实践修复步骤，或者可重新调整基础架构以消除这些问题。这就是可发现的预防性维护，它可能具有更大的影响力。

我们的一些大型保险和零售客户需要知道一些问题的答案，如“故障有哪些先兆？”、“所有这些系统有何关联？”等。您可在这里看到一种跨行业模式，对吗？这些是传统的监控无法回答的问题。大数据平台最终提供了获取所见问题的更新和更透彻洞察的机会。

欺诈检测模式

欺诈检测在金融服务领域由来已久，但如果您环顾四周，会在所有基于索赔或事务的环境中看到它的身影（在线拍卖、保险索赔、保险机构等）。涉及某种金融事务的几乎任何地方都存在着滥用的可能和无处不在的欺诈风险。如果您利用大数据平台，就有机会执行比以往更多的工作来识别它，或者甚至停止它。

欺诈检测平台中的一些挑战可直接归因于对传统技术的单独利用。您将在所有大数据平台上看到的最常见且反复出现的主题是，对可存储的内容以及可用于满足您意图的计算资源的限制。没有大数据技术，这些因素会限制建模的内容。更少的数据意味着受限的建模功能。而且，高度动态的环境常常具有每个几小时、几天或几星期就会出现的新的周期性欺诈模式。如果无法及时获得可识别或支持新欺诈检测模型的数据，那么在您发现这些新模式时为时已晚，一些损失已然发生。

在传统的欺诈案例中，会使用示例和模型来识别具有某种概况的客户。这种方法（这也是您将在大量用例中看到的趋势）的问题是，尽管它有效，但您只能探查部分数据，而达不到各个事务或人员级别的粒度。简言之，基于部分数据进行预测没什么问题，但基于各个事务的实际详情来制定决策显然要更好。为此，您需要处理比传统方法中通常可处理的数据更多的数据。在我们的客户体验中，我们估计可能对欺诈建模有用的可用信息中只有 20%（或许更少）被实际利用了。传统方法如图 2-1 所示。

您可能想知道，“答案不就是将其他 80% 的数据加载到传统的分析仓库中吗？”可要求您的 CIO 对这么做所需的 CAPEX 和 OPEX 进行审批：您很快会发现这个建议的成本太高了。您可能认为它会以更好的欺诈检测模型来弥补，尽管这确实是最终目标，但如何确保新加载、清理、记录和治理的数据从一开始（在花钱之前）就具有价值？这里的关键在于：您可以使用 BigInsights 提供的一个有弹性且经济高效的存储库来确定剩余 80% 的信息中有多少对欺诈建模有用，然后将新发现的高价值信息回馈给欺诈模型（这就是本章前面引用的左手-右手抛接棒球的示例），如图 2-2 所示。

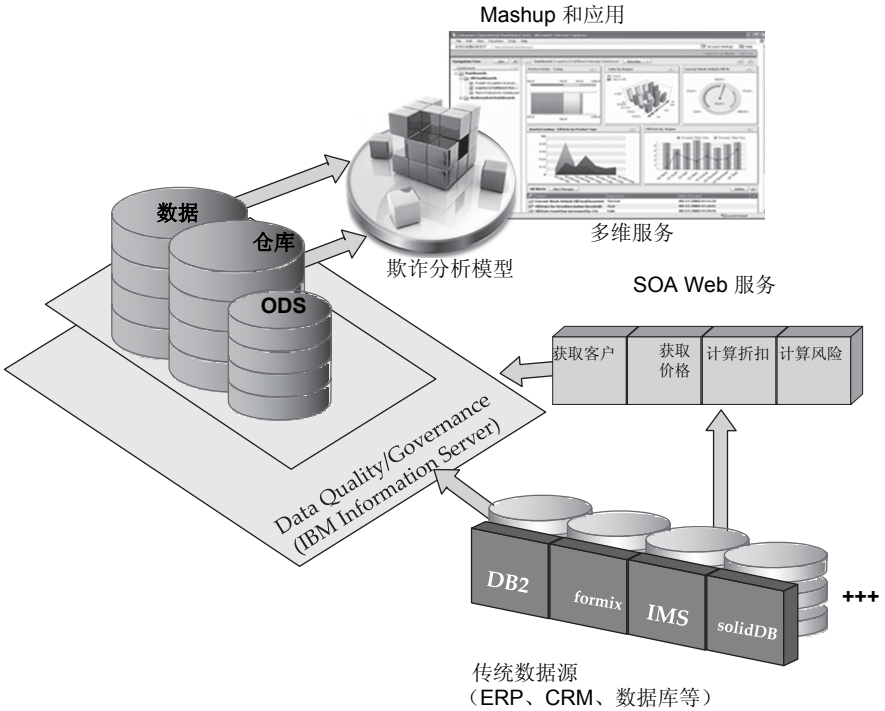

图 2-1 *传统欺诈检测模式仅使用了可用数据的大约 20%*

图字：Data Quality/Governance：数据质量/治理

在图 2-2 中，您可以看到一种现代的欺诈检测生态系统，它提供了一个低成本的大数据平台用于探索式建模和发现。请注意传统系统也可利用此数据，它们可直接利用或通过整合到现有的数据质量和治理协议中来利用。还要注意增加的 InfoSphere Streams（DB2 数据库圆柱构成的圆圈），它展示了只有 IBM 可提供的独特的大数据平台：它是一个生态系统，提供了动态数据和静态数据的分析。

我们与一家大型信用卡发行商合作得出了图 2-2 中的组合，该发行商很快发现它们不仅可改善欺诈检测模型的构建和刷新的速度，其模型还更加庞大和准确（得益于所有的新洞察）。最后，这家客户将此模式应用于某一流程中，将这一流程从事务开始到它实际可供欺诈团队处理的时间从3周缩短至了2个小时。

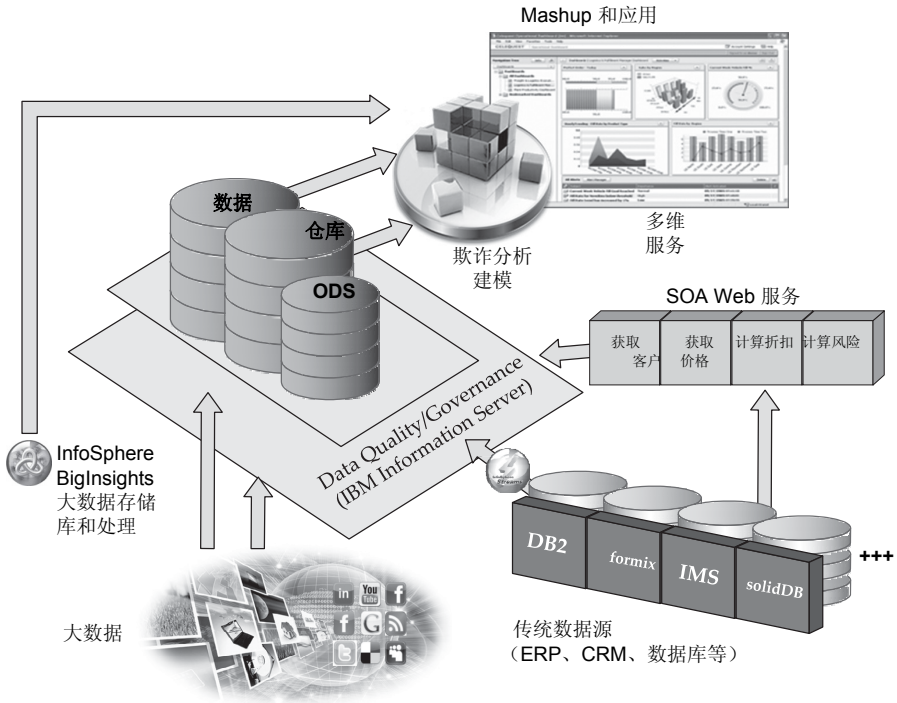

图 2-2 *一个现代的欺诈检测生态系统将一个大数据平台与传统流程结合在一起*

图字：Data Quality/Governance：数据质量/治理

此外，此欺诈检测模型是在比以前的数据集多大约 50% 的数据量上建立的。正如我们在此示例中所见到的，我们提到的未使用的所有“80% 的数据”在最终并不是都有用，但它们通过 BigInsights 平台以一种经济高效的方式确定了哪些数据有用，哪些数据无用。当然，现在构建了欺诈模型后，您会希望应用它们来尝试从一开始就预防欺诈。欺诈的恢复率在所有行业中都比较低，所以最好是预防它，而不是发现它和尝试在发生欺诈后进行恢复。这正是 InfoSphere Streams 的用武之地，如图 2-2 中所示。通常，只有在事务存储后才使用欺诈检测 - 从存储中拉取该事务并执行分析；存储内容后立即再次将其提取出来会让我们感觉到延迟。使用 Streams，您可在事务正在发生时应用欺诈检测模型。

在本节中，我们重点介绍了一家金融服务信用卡公司，因为它是我们在首次启动大数据平台后一种早期的一对一体验。您不应局限于我们在此所提供的信息来考虑本节列出的用例；事实上，我们在本章开头即表明，存在数百个使用模式，我们不能一一列举。实际上，欺诈检测的适用性很大。考虑医疗市场中的欺诈（健康保险欺诈、药物欺诈、医疗欺诈等）和掌控保险商与政府欺诈模式的能力（包括索赔者和提供商）。联邦调查局 (FBI) 估计医疗欺诈一年会耗费美国纳税人 600 多亿美元的税款，因此这是使用此模式的绝佳机会。考虑欺骗性的在线产品或票据销售、货币转移、偷盗的银行卡等：您可以看到此使用模式的适用性非常高。

他们怎么说？社交媒体模式

或许对于大数据使用模式，讨论最多的是社交媒体和客户观点。您可使用大数据确定哪些客户在讨论您（以及他们如何评价您的竞争对手）；而且，您可以始终用这种新洞察来确定这种态度对您制定的决策和公司的服务方式有何影响。更具体来讲，您可以确定这种态度如何影响销量、您的营销活动的效力或可接受性、您的营销组合（产品、价格、促销和更换）的准确性等。

社交媒体分析是一个非常热门的主题，以至于 IBM 专门构建了一个解决方案来方便您对它的使用：Cognos Consumer Insights (CCI)。它是一个在 BigInsights 上运行的单点解决方案，能出色地完成其工作。CCI 可告诉您人们的观点、社交媒体中的主题趋势，以及所有影响您业务的事情，它们都封装在一个富可视化引擎中。

尽管对社交媒体的基本洞察可告诉您人们的态度和态度变化趋势，但它们无法回答一个更加重要的根本问题：“为什么人们这么说和这么认为？”回答此类问题需要使用更多且以不同方式形成的、可能位于其他企业系统中的信息来充实社交媒体源。

简言之，将行为和该行为的驱动因素链接起来需要将社交媒体分析与您的传统数据存储库相关联，无论它们是 SAP、DB2、Teradata、Oracle 还是其他存储库。您的眼光必须超越该数据，您必须看到用户行为、当前的金融趋势、隐藏着的实际事务等之间的相互作用。销售、促销、忠诚度计划、销售组合、竞争对手举措甚至天气等因素都可能影响消费者的感觉和观点。要准确地了解您的客户某种行为的原因，需要以一种动态且经济高效的方式合并信息，尤其是在项目的初始探索阶段。

它有效吗？以下是一个真实的案例：一家客户为它的一个钉书针品牌引入一种不同的环保包装。客户对新包装的态度比较消极，几个月后，在跟踪客户反馈和评论后，该公司发现人们对这一变更存在很大不满，进而采用了一种不同的环保包装。这种方法生效了，我们非常欣赏这家积极进取的公司利用大数据技术来发现、理解和应对大众态度的做法。

我们将假设，如果您的公司没有某种以微博为导向的客户态度监督机制，您的客户可能会转而选择另一家这么做的公司。

备注：*本书的一位作者是一位多产的 Facebook 博主（出于某种原因，它认为全球的人们对他每天的想法和体验很感兴趣）；他在 Facebook 上报道了连续几次航班晚点后，航空公司联系了他，请求解决它检测到的大众态度问题。航空公司承认了晚点问题，尽管我们对航空公司为他做了什么不得而知（或许是更高级的舱位），但事实是它们与他联系意味着有人在倾听，这在一定程度上具有积极的作用。*

我们认为观看每秒 Twitter 上的 tweet 数 (Ttps) 指数这一世界记录是客户态度潜在影响的一个明显指标。Super Bowl 2011 在 2011 年 2 月以 4064 Ttps 创造了一个 Twitter Ttps 世界纪录；它被本拉登死亡公告的 5106 Ttps 所超越，随后又被毁灭性的日本地震的 6939 Ttps 超越。这个 Twitter 记录

在美洲杯四分之一决赛中巴拉圭在任意球环节战胜巴西时以 7166 Ttps 达到顶峰，但这可能还未打破同一天创造的另一个记录：女子世界杯足球赛上美国队一次获胜的 7196 Ttps。当我们将本书付印时，流行歌手 Beyonce 在 Twitter 上的怀孕公告产生了 8868 Ttps，成为了一直保持的记录。我们认为这些记录非常有说服力——不仅因为数据量和速度的增长，还因为所表达的态度说明了一切，包括您的产品和服务。诚然，客户态度无处不在，只需问问 Lady Gaga (@ladygaga)（全球受关注最多的 Tweeter）即可知道。我们从中学到了什么？首先，每个人都能够在几秒内表达他们的反应和态度（常常无需思考或过滤）并让全球的人看到；第二，越来越多的人在表达他们对一切事物的想法或感觉。

呼叫中心口号：“为了保证质量，本次通话可能被录音”

您一定对本节的标题感到很熟悉：很奇怪，似乎为了质量保证的用途，在我们希望记录我们与客户服务代表 (CSR) 的通话时，“*可能*”部分似乎从未生效。呼叫中心效率的挑战在某种程度上类似于我们讨论的欺诈检测模式：欺诈信息延迟对强大的欺诈模型很关键，与之类似，如果您使用过呼叫中心，就会知道呼叫中心的时间/质量决定指标和趋势不满模式可能在事实发生后几星期才显现出来。这一延迟意味着，如果某个人正在打电话且遇到一个问题，从企业的角度讲您不会立刻知道该问题，您不会知道人们正在电话里探讨这个新主题，或者您已在特定领域的交互中看到了新的和可能的消极趋势。底线是：在许多情况下，这个呼叫中心的所有信息太少、来得太迟了，问题被单独留给 CSR 来处理，而没有现成的稳定的、经过检验的修复过程。

许多客户曾要求我们帮助解决此模式的问题，我们相信它非常适合大数据。所有类型的呼叫中心都希望找到

更好的方式来处理信息，以较低的延迟解决业务中发生的问题。这是一个真正有趣的大数据用例，因为它使用了动态分析和静态分析。使用动态分析 (Streams) 意味着您会构建您的模型，基于从语音转换为文本的对话或在通话正在进行时使用语音分析来查找有趣的信息。使用静态分析 (BigInsights)，您可以建立模型，然后将它们传送回 Streams 来实时检查和分析实际发生的呼叫：它是一种真正闭环的反馈机制。例如，您使用 BigInsights 不断对数据运行分析/探索/建模，当发现新模型时，您创建新业务规则并将其推送到 Streams 模型中，以便在某个事件发生时采取及时的操作。或许如果客户提到一个竞争对手，CSR 会收到一个警告，以通知他们当前有竞争力的促销活动并为打电话的客户生成一个最佳推荐。

您可以想像所有有效的用例——捕获态度，这样能知道人们在说什么、表达什么，甚至人们自愿提供的与意向相关的信息，您的公司然后再采取特定的操作；诚然，这是一项宝贵的洞察。此外，随着越来越多的 CSR 外包和接电话的 CSR 极有可能不是您本地口音，不满情绪的细微变化并非总是容易察觉的，这种类型的解决方案可帮助呼叫中心改善它的效率。

一些行业或产品拥有流失率很高且极不忠诚的客户。例如，我们的一家客户所在的行业具有 70% 的年均客户流失率（不同于北美洲，在全球其他地区手机联系方式不受限制）。通过识别哪些类型的客户最有价值、他们在呼叫谁、谁对给定的主题感兴趣等，可实现客户保留率的改进，即使是很小的改进都有可能为客户带来巨大的收益。而对于另一家客户，仅仅 2% 的转换率差异就可能实现其产品的收入加倍。

如果您能够捕获并检测忠诚度的下降并将此趋势融入到您的 CSR 协议、模型，以及您所遇问题的一个现有修复产品中，它可能在损失规避或增加收入（从满意的客户到潜在的交叉销售服务）方面带来非常好的成果。

例如，您在 BigInsights 中完成了语音到文本转换后，可以将该文本与一切事物相关联，从电子邮件到社交媒体和我们在本章中探讨的其他事物；您甚至可以将它与后台服务质量报告相关联，查看人们是否通过您的后端系统呼叫您并表达不满。如果您能够关联和识别一种模式来表明系统在何处运行缓慢或行为异常，而这些异常又恰好是客户打电话要求取消其服务（但他们并未明确说出来）的原因，现在就可通过此模式从客户的言语中找到关联。

我们都可能遇到过下面这个场景：想像在断网两次后呼叫一个客户服务部门，代理反馈说“我们很抱歉，我们遇到了某个电话问题并注意到您断网了两次……”您有多少次打电话向您的高速互联网提供商投诉过服务问题，而 CSR 只是推诿？CSR 除了接听电话没有采取任何措施。这些服务问题是否被真正捕获？或许处理呼叫的代理填了一张表来反映基本的服务投诉，但这些是否已被捕获并与时间点质量报告相关联从而来表明系统的运行状况？而且，我们正在与客户合作利用大数据的不同方面来关联呼叫中心中的趋势与业务操作中的剩余部分。作为一个示例，哪些类型的呼叫和交互与续订、交叉销售、索赔和保险公司的其他多种重要指标相关联？如今很少有公司执行这些关联，但他们需要能够这么做以便随时关注竞争对手的动向。您如何赶上或者甚至引领这一区域的发展趋势？

这是一个真正有趣的大数据用例，因为它应用了如今使用动态分析和静态分析的艺术，是 Watson 等新兴功能的完美用例。使用静态分析 (BigInsights)，意味着基于转换为文本的语音信息或者语音分析来构建模型。然后可以选择继续使用静态分析，用比传统操作节奏低得多的延迟（数小时）来利用呼叫交互，或者将构建后的模型传送回 Streams，立即检查和分析

呼叫，近乎实时地发现正在发生的事情。之后Streams 分析的结果被传送回BigInsights，这表明它确实是一种闭环的反馈机制，因为 BigInsights 之后会迭代结果来改进模型。在不久的将来，我们会看到Watson将被添加到产品组合中来增强 Streams 所期待的模式分析，基于比目前可供使用的呼叫中心议程更丰富的选项来提供如何处理交互问题的专家建议。

您可以从此模式中得出结论，数据中心中存在许多可处理大数据的“一流”功能，而在最开始采取传统的头脑风暴很重要。使用 BigInsights 平台，真正带来了无限可能。有效地分析过去无法捕获的信息是一种既定的大数据模式，可帮助您以全新方式理解事物，最终将这些与您现有的分析系统执行的操作相关联。

风险：建模和管理模式

风险建模和管理是另一个重要的机会和常见的大数据使用模式。在本章中讨论的大数据使用模式上，风险建模关注一个反复出现的问题，“您在建模过程中使用了多少数据？”2008 年的金融危机、相关的次贷危机以及它的后果使风险建模和管理成为金融机构一个重要的关注区域。您通过如今的金融市场可以看到，缺乏对风险的理解可能对财富的创造具有毁灭性的影响。此外，新制定的法规要求全球的金融机构确保他们的风险级别在可接受的阈值之下。

与欺诈检测模式中一样，我们的客户活动表明，在此区域，公司在风险建模中使用了 15～20% 的可用结构化数据。公司并非没有认识到有大量数据可能利用不足，需要在风险模型中制定许多业务规则；只是它们不知道可在剩余数据中的何处找到相关的信息。此外我们已经看到，在许多客户当前的基础架构中找到所需数据的代价非常高，因为客户显然不能因为*可能*（这个词很关键）存在其他一些有用的信息而将仓库的规模扩大两倍、三倍或四倍。

而且，一些客户的系统无法处理高达 80% 的未利用数据可能带来的更高负载，所以即使它们拥有 CAPEX 和 OPEX 预算，可将仓库规模扩大两倍或三倍，但许多传统的系统也无法处理伴随使用“剩余的数据”而导致的数据和分析激增。不要忘记一些数据甚至无法在传统系统中进行处理，但对建模风险可能很有帮助，您很快会遇到一个在根本上需要一种新方法的谜题。

让我们回头想想在一家金融公司的一个交易日结束时会发生什么：它们基本而言会获得所在位置的结算快照。使用此快照，公司可获取洞察并使用它们的模型在两个小时内识别问题和重点，然后报告给监管者进行内部风险控制。例如，您不希望发现您在伦敦预定的业务会在北美的交易日开始后影响纽约的交易。如果您提前知道了风险，就可以对它们采取措施，而不是由于您的纽约办事处不知道或无法准确预测而使问题变得更糟。现在将此示例扩展到更广泛的全球金融市场（例如向投资组合中添加亚洲），您可以看到会发生相同的事情，除了风险和问题是混合的。

有两个问题与此使用模式关联：“您将为您的模型使用多少数据？”（我们已问过这个问题）和“您如何跟上数据的增长速度？”不幸的是，第二个问题的答案常常是“我们无能为力。”最后，考虑金融服务倾向于将它们的风险模型和仪表板转移到一天的中期位置，而不是一天结束时的位置，您可以看到另一个无法单独使用传统系统解决的挑战。如今的金融市场的另一个特征（除了我们继续调整我们计划的退休日期）是存在巨大的交易量。如果您综合考虑庞大的数据量、更好地建模和管理风险的需求，以及无法在模型中使用相关数据（更不用说快速构建数据或在一天的中期运行数据），就会发现您遇到了一个大数据问题。

大数据与能源领域

能源领域在如何处理来自远程安装的大量传感器数据方面带来了许多大数据用例挑战。许多公司仅使用了收集的部分数据，因为它们缺乏基础架构来存储或分析可用的数据。

以一个拥有 20,000 到 40,000 个板载传感器的典型石油钻井平台为例。所有这些传感器都在传输有关石油钻井平台的健康状况、操作质量等数据。不是每个传感器始终都能主动预测，但一些传感器每秒会报告许多次。现在猜猜有效利用了这些传感器的多大比例。如果您认为 10%（或者甚至 5%），那么您是一位出色的猜测者或者您遇到了贯穿多个行业和用例且反复发生的大数据主题：客户未在决策制定流程中使用可供他们使用的所有数据。当然，对于能源数据（或者任何类似数据）收集率，真正的问题是，“如果您已费尽心力配备用户、设备或钻塔（在理论上您会特意这么做），为什么您没有捕获和利用您收集的信息？”

考虑到利润、安全和效率，业务应该不断寻找信号并能够将这些信号与他们潜在的或可能的成果相关联。如果您将 90% 的传感器数据作为噪音而丢弃，就可能无法理解或建模这些关联。“传感器噪音桶”只有在缺乏存储和分析所有数据的能力时才会变得这么大；人们需要一个允许将真实信号与噪音分离的解决方案。当然，它还不足以捕获数据，无论是噪音还是信号。您必须确定洞察（并清除噪音），而且这趟旅程不能就此结束：您必须能够对这些有用的洞察采取操作。这是动态数据分析和静态数据分析形成出色的大数据协调配合的不错示例：您必须在所识别的有用数据静止时对它采取操作（如构建模型），还必须在实际发生事情时采取操作：一个优秀的动态 Streams 用例。

丹麦的一家 BigInsights 客户 Vestas 是能源领域的全球领导企业，它的口号是，“风对我们而言就是全世界。”Vestas 主要参与开发、制造、销售和维护使用风能发电的风轮机发电电力系统。

其产品范围涵盖内陆和海岸风轮机。在我们编写本书时，它在 5 大洲 65 个国家和地区拥有 43,000 多个风轮机。对我们而言，了解 Vestas 非常有用，因为他们的愿景是产生清洁的能源，它们正在使用 IBM BigInsights 平台帮助他们以更高的利润和效率生产更多清洁能源，这使我们引以为豪。

替代能源领域的竞争非常激烈，需求也在激增。它还具有非常有竞争力的定价，所以只要您具有任何优势，就可以进军此市场。事实证明，一台风轮机需数百万美元的投资，具有 20 到 30 年的使用寿命。这击溃了我们的防线。我们没有认识到安置风轮机所需的工作和错误布置风轮机的影响。选择安装和操作风轮机的位置可能对该装置产生的电量以及它能够保持运作的时长具有重大的影响。要确定风轮机的最佳位置，必须考虑大量依赖于位置的因素，如温度、沉降、风速、湿度、气压等。这种数据问题急需一个大数据平台。Vestas 的建模系统预计最初需要 2.6 PB (2600 TB) 的容量，自它们的工程师开始开发其自己的预测并记录已安装的每个风轮机的实际数据，它们的数据容量需求预计将增长到6 PB (6000 TB)这一惊人数字！

Vestas 以前分析此数据的流程不支持使用完整的数据集（大数据平台解决的问题包括这种常见的模式）；而且，它们花了数周时间来执行该模型。Vestas 认识到他们面临着一个大数据挑战，而这一挑战可以使用基于 Hadoop 的解决方案解决。该公司想要一个允许它们利用已收集的所有数据来平缓建模的时间曲线，支持进一步扩展建模技术，以及改善风轮机位置决策的准确性的解决方案。考虑多个其他的供应商之后，Vestas 联系了 IBM，要求提供一个企业级、基于 Hadoop 的大数据分析平台，该平台要支持开源组件并使用本书第 2 部分介绍的 IBM 增强扩展它们（考虑一个完全自动化的安装，Hadoop 的企业级安全性、

基于文本和统计信息的分析、治理、企业整合、开发工具、资源治理、可视化工具等）。

在 IBM System x 服务器上使用 InfoSphere BigInsights，Vestas 能够使用以前不可能实现的方式来管理和分析天气和位置数据。这使得它们能够洞察更好的风轮机位置和操作决策。所有这些分析来自他们的 Wind and Site Competency Center，Vestas 工程师在这里持续建模天气数据，基于一个方圆 1 千米的网格（按国家组织）的全球集合来预测最佳的风轮机位置，这些网格跟踪和分析*数百个*变量（温度、气压、湿度、沉降、风向、从地面一直到 300 英尺高的风速、全球森林开发指标、卫星图像、历史指标、地理空间数据，以及有关月相和潮汐等的数据）。查看 Vestas 在他们的预测模型中包含的一个变量抽样时，您可以看到数据量（PB 级数据）、速度（所有这些数据随天气变化而不断变化并传输到数据中心）和种类（所有不同的格式、结构化、非结构化，还有很多原始格式的数据），这些使此问题成为了可通过与基于 IBM 大数据平台的 IBM 智慧能源计划合作来解决的大数据问题。

3

为什么选择 IBM 的大数据解决方案？

多少次在您听到“这改变了一切”之后，历史却表明事实上并没有太大的改变？我们要在这里明确一点（我们在整本书中都会重复这重要的一点，以确保不存在任何疑问）：大数据技术很重要，并且我们会做到使这些技术成为几乎所有大型组织向前发展的关键路径，但传统的数据平台不会消失，它们只是优秀的合作伙伴。

这里列举一些有用的历史背景：几年前，我们听说 Hadoop 将“改变一切”，并且它将“使传统的数据库过时”。我们自己在心里想，“无稽之谈”。这样的说法要求对某些关键动态进行分析，而这些动态往往会被忽视，其中包括注意大数据技术在成熟度曲线上的位置，挑选合适的合作伙伴来完成这一旅程，了解它如何补充了传统的分析平台（上一章中左右手的比喻），并在选择大数据合作伙伴时考虑到人员组成。综上所述，我们的确认为大数据对于数据中心的整体效益而言是一个改变游戏规则的工具，因为它有可能成为您的信息管理法宝中的强大工具。

大数据对于我们很多人来说都是比较新的，但是您想通过大数据平台收获的价值则不然。打算将 Hadoop 实施到其企业环境的客户对于 MapReduce 编程模型为它们提供的机会感到非常兴奋。虽然它们喜欢对海量数据进行分析 - 这个在过去认为成本过高的想法 - 但它们仍然是带有企业期望和要求的企业。出于这个原因，我们认为 IBM 决不是这个游戏的新人，更不用说

我们与 Google 从 2007 年 10 月已开始全作 MapReduce 项目。

您可以想像，IBM 在集成解决方案方面拥有丰富的资产和经验，确保它们是兼容的、高度可用的、安全的和可恢复的，并且提供了一个框架，供数据在其整个信息供应链中流动，该框架是可信任的（因为没有人会只因为自己喜欢运行软件而购买一个 IT 解决方案）。

想想一个艺术家画一幅画时：一块空白的画布（一个 IT 解决方案）就是一个机会，您要画的图案是最终目标——您需要合适的画笔和颜色（有时您会混合一些颜色来使它完美）来画您的 IT 图案。对于只销售与服务或新进入市场的文件系统绑定的开源 Hadoop 解决方案的企业，讨论开始后会结束于将图画挂到墙上所需的锤子和钉子。您最终不得不走出去采购绘画用品，并依靠自己的艺术技巧来画这幅图画。IBM 大数据平台就像是一个“数字色彩”绘画工具包，其中包括您所需的一切，能帮助您快速地框架，绘画，并悬挂一套充满活力的、详细的图画，以及您认为合适的任何自定义内容。在该工具包中，IBM 提供您所需的一切，包括为开发、自定义、管理和数据可视化所设计的工具集，针对统计数据和文本预构建的高级分析工具包，以及 Hadoop 运行时的企业硬化，这一切都包括在一个自动化安装包内。

IBM 世界级的、屡获殊荣的研究机构，继续通过高度抽象的查询语言、优化、文本和机器学习分析等接受和扩展 Hadoop 领域。利用开源技术的其他公司，特别是规模较小的公司，可能有大量项目（IBM 公司也是如此），但它们所具备的知识深度通常不足以了解对企业至关重要的特性集。例如，开源有文本分析和机器学习组件，但这些工具都还不完善，也不易于使用，并且其扩展性不如 BigInsights 中的工具，这一点对于企业而言真的非常重要。毫无疑问，对于某些客户而言，开源社区就是它们所需要的，并且 IBM 绝对尊重这个事实（这就是您可以从 IBM 单独购买一个 Hadoop 支持合同的原因）。对于希望获得传统支持和交付模型，并使用在文本和机器学习分析以及其他企业特性中数十亿美元投资的其他客户，IBM 提供其大数据平台。IBM 也提供其他优势供您考虑：

24×7 直接工程师支持、以您的母语提供国际化的代码和服务等。我们实际上已拥有数千名员工可以与您配合，帮助您绘出自己的图画。此外，还有来自 IBM 的解决方案，如在 BigInsights 上运行的 Cognos Consumer Insights，它可以推动您的大数据项目。

考虑到 IBM 在 Hadoop 系统上所添加的所有好处时，就可以理解为什么我们将 BigInsights 称为一个平台。在本章中，我们介绍有关 IBM 为大数据解决方案所带来的价值的非技术性细节（在第 5 章中，我们将深入探讨技术细节）。

大数据没有老大哥：它已经准备好，但还很年轻

咨询任何一位有经验的 CIO，他们都会首先告诉您，在许多方面，技术是比较容易的部分。我们希望提供可以反映技术和人员交集的观点。可以举一个很好的例子来说明这一点，我们总是会问自己这个非常务实的问题：“哪些在仓库领域中可行的东西在这里也需要用到？”

注意，我们并没有问什么技术可行；这是一个比技术更广泛的问题。创建和保护数据集市、实施工作负载优先次序，以及扩大业务用户与开发人员的比例，虽然这些都以技术为基础，但它们都是从数千个人的运营经验中产生的最佳实践。这里有一些很好的例子：大型 IT 投资之后，往往是停滞在“科学项目模式”中的项目，停滞的原因不是技术失败，而是技术没有指向要解决的正确问题，或者它不能集成到数据中心供应链的其余部分，以及往往比较复杂的数据中心信息流。我们还看到了许多最初成功的小规模部署，但它们面临的挑战是使其迈过临时阶段，因为其任务中的“企业”部分已经开始呼唤更高的阶段（关于这一点会更详细讨论）。这往往能说明围绕 Hadoop 的热议和缺乏显著大规模使用之间的差异。现在，这听起来似乎有点矛盾，因为 Hadoop 是众所周知的，并且 Twitter、Facebook 和 Yahoo 等巨头都在使用它；但要记住，所有这些公司都有巨大的开发团队，有可能大部分财富 500 强企业都负担不起这些团队，因为它们既不是技术公司，也不希望成为技术公司。

它们希望找到创新的方法来推动自己具有核心竞争力的业务。

有些客户拥有非常庞大的 IT 预算，对于它们想做的任何事情，其资金都可以支持自力更生 (RYO) 的环境，除了这些客户之外，也有大量公司在生产环境中使用 Hadoop，但它们并非传统意识上的企业。例如，有数据质量的要求吗？服务水平协议 (SLA) 是否将 IT 绑定到业务主管方的合同里？数据安全吗？是否执行了隐私政策？是否为关键任务解决方案，并因此具备生存（灾难恢复和高可用性）计划，计划中已定义好平均修复时间 (MTTR) 和恢复点目标 (RPO)？

我们提出这些问题是因为我们从客户那里听说，他们从“使用 Hadoop，但它并没有达到为我们企业中其他解决方案所设定的企业期望水平”的方法开始。我们在这里希望可以澄清一点：我们是 Hadoop 及其社区的超级粉丝和支持者；但是，有些客户要求我们解决特定的需求（我们认为大多数用户最终将提出同样的需求）。IBM 大数据平台的目的是“接受和扩展”IBM 接受这个开源技术（我们已经详细介绍了一长串对开源的贡献，包括 IBM 的 Hadoop 委员会，事实上，我们没有放弃代码，并且我们承诺保持向后兼容性），并围绕我们的客户对我们所提出的需求扩展该框架，即分析的丰富化和一些企业优化特性。我们相信，开源 Hadoop 引擎配合强化和扩展它的丰富生态系统，可以在业务流程中成为满足企业期望的头等公民。毕竟，Hadoop 的重点并不在于响应时间达到思想的速度，它也不是为了在线事务处理 (OLTP)；它针对的是批处理作业，并且大家都知道，批处理窗口正在缩小。虽然企业将扩展它们以获得之前不可能获得的洞察，但您认为要经过多长时间，您的 Hadoop 项目的可用性和性能要求才会得到一个“我爱我的 SLA”纹身呢？这是不可避免的。

Hadoop 解决方案提供给企业的价值越多，它就越接近关键的十字线，这意味着新的期望和新的生产 SLA 水平。试想一下，尝试向负责风险管理的副总裁解释，您不确定自己开放的风险仓位和分析计算是否准确和完整。疯了吗？现在，说句公道话，这些挑战存在于任何系统中，我们并不是说 Hadoop 不好。然而，它在您的业务中变得越流行、越重要，

就会有越多监察将被应用到在其上运行的解决方案。例如，您将会有太多开放端口要进行审计检查，您将被要求职责分离，您可以将最小特权的原则应用到运营中等。

这种情况比您所预期的还更常见，它发生的原因是许多人没有退后一步来看看范围更广的背景以及需要解决的业务问题。使用年轻的技术这个现实也是一个原因，而解决这个问题需要未来的大量创新。

IBM 提供一个合作伙伴关系，不仅使您获得一个可以将时间分析曲线变平并解决许多企业需求的大数据平台，它还真正地提供了关键资源的经验，并理解支持和维护这些关键资源的重要性。例如，IBM Data Server 驱动程序每一天都支持每小时数十亿美元的交易——这就是业务关键性！将它与 Hadoop 等技术创新，以及 IBM 对开源的全力支持结合起来，您就拥有了一个很好的机会。

您的大数据合作伙伴可以为您做些什么？

到目前为止，我们在本章中已简要地暗示了 IBM 在大数据解决方案为您提供的功能——即提供一个平台，而不是一个产品。但是，在任何公司背后，您都需要寻找它可以带上桌面讨论的资源，它如何在您的追求和目标中支持您，它在哪里可以支持您，它是否可行并投资于平台的未来及可提供更多价值的增强功能。抑或只是凑凑热闹，对某个产品提供一些支持，并没有从平台的角度来考虑，让您自己装配它并琢磨大部分的企业挑战。

在本节中，我们将谈论 IBM 为了确保大数据平台获得成功所进行的一些工作以及它所提供的资源。在您看到 BigInsights 时，您看到的是 200 多位 IBM 研究科学家经过 5 年努力所完成的拥有多项专利的获奖作品。例如，IBM 的 General Parallel File System – Shared Nothing Cluster (GPFS-SNC) 赢取了 SC10 Storage Challenge（SC10 存储挑战）奖，该奖项颁发给在竞争中最具创新性的存储解决方案。

IBM 的 1 亿美元大数据投资

作为 IBM 对围绕 Hadoop 平台不断进行创新这个承诺的证明，在 2011 年 5 月，它宣布对大规模分析投资 1 亿美元。这里值得注意的关键词是*分析*。假设有多个供应商提供某种 Hadoop 产品。它们之中有多少人会将它作为一个包括加速器和分析相关功能的平台推出？或者是否有某些东西留给您自己从头开始构建，或分别购买后再集成，并且您的 IT 解决方案会涉及不同的工具、服务支持合同、代码质量等？考虑分析时，请考虑 IBM SPSS 和 IBM Cognos 资产（不要忘记还有 Unica、CoreMetrics 等许多选择），以及 Netezza 或 IBM Smart Analytics System 内的分析知识产权。事实上，IBM 有一个业务分析和优化 (BAO) 部门，这不言而喻，并且代表着 IBM 在其大数据平台中将为分析提供各种长期功能。并且不要忘记，据我们所知，*没有其他*厂商可以同时讨论移动中的大数据（InfoSphere Streams，或简称为 Streams）和静止的大数据 (BigInsights)，*并*同时为两者提供分析。

IBM 可以出色地作出这个尺度的承诺，因为它拥有一个世纪的成功创新纪录。IBM 拥有地球上最大的商业研究机构，如果这还不够，我们将以一个毫不夸张的事实来完成本节，您可以消化一下，像 IBM 这样的合作伙伴可以对您的大数据业务目标产生的影响：在过去的 5 年中，IBM 已经在 24 次分析收购中投入了超过 140 亿美元。今天，IBM Research 中有 8000 多名 IBM 商业顾问致力于分析，还有 200 多名数学家正在开发突破性的算法。这还只是分析；我们还没有谈论针对企业的适用性所进行的 Hadoop 强化、我们对 Apache 项目（包括 Hadoop）的提交等。所以，请您告诉我们，这听起来像是您组建球队时所喜欢的那种球员吗？

大数据创新的历史

阅读本节之前，我们要明确，它是市场营销信息：它听起来像市场营销，看起来像市场营销，并且阅读起来也像市场营销。但 IBM 市场营销的特点是，它就是事实（我们很乐意

在这里对我们某些竞争对手开个明白的玩笑，但我们很确定刚刚已经这么做了）。因此，在下面的章节中所讨论的创新显示，IBM 世世代代都一直在研究和解决问题，其研究实验室通常是超前于市场，并往往在问题发生之前就提供了相应的解决方案。在我们完善本书的业务方面时，让我们花点时间回想一下 IBM 以前、现在及将来可能成为的合作伙伴类型，再花一点点时间回想一下它过去的创新，这些创新使 IBM 现在已准备好成为您的大数据合伙人。

事实上，IBM 在历史上有多个第一，可能您还是第一次听说：从第一个交通灯计时系统，到 Fortran、DRAM、ATM、UPC 条形码、RISC 架构、PC、SQL 和 XQuery，再到关系型数据库技术，以及这些之间的数百个其他创新资产（请在 www.ibm.com/ibm100/ 查阅该历史记录，了解跨一个世纪的创新概要）。让我们看一下 IBM 多年来的一些创新，看看它们如何将 IBM 独特地定位为大数据行业的佼佼者。

1956：第一个磁性硬盘

IBM 推出了世界上第一个用于数据存储的磁性硬盘 Random Access Method of Accounting and Control（会计与控制的随机存取法，RAMAC），允许随机存取分布在 50 个 2 英尺直径的磁盘两侧的一百万个字符，提供了前所未有的性能。在加利福尼亚州圣何塞生产的第一个 IBM 硬盘，每平方英寸可以存储约 2000 比特数据，其购买价格约为每兆字节 10,000 美元。到 1997 年，每兆字节的存储成本已下降到 10 美分左右。IBM 今天仍然是存储游戏的领导者，提供创新的重复数据删除优化、自动根据数据的使用率放置数据（当您计划存储 PB 级数据时是个不错的方法）、固态硬盘等。对大数据来说幸运的是，驱动器的价格继续下降，同时容量在不断提高；但是，如果没有 IBM 发明经济的磁盘驱动器技术，大数据不会有可能实现。

1970：关系型数据库

IBM 科学家 Ted Codd 发表了一篇介绍关系数据库概念的论文。它呼吁在计算机内存储的信息应该被安排在易于理解的表格中，使非技术用户

可以访问和管理大量数据。今天，几乎所有的企业级数据库结构都以 IBM 的关系数据库概念为基础：DB2、Informix、Netezza、Oracle、Sybase、SQL Server 等。您的大数据解决方案不会独自存在，它必须集成，并很可能会增强您的关系型数据库，很少其他公司在这一领域能够声称提供同类的体验——IBM 发明了它。

1971：语音识别

IBM 建立了第一个业务语音识别应用程序，使设备维修工程师可以与计算机交谈，从计算机接收语音回答，该计算机可以识别约 5000 个单词。目前 IBM 的 ViaVoice 语音识别技术的词汇量达到 64,000 个单词，以及一个有 260,000 个单词的备份字典。1997 年，ViaVoice 产品在中国和日本推出。还专门针对在急诊医学、新闻、法律和放射领域工作的人提供了高度自定义的 VoiceType 产品。现在想一想语音识别技术，因为它与在第 2 章中所列的呼叫中心用例有关，并认识到 IBM 拥有该领域的知识产权，这甚至可以追溯到本书某些读者出生之前的年代。

1980：RISC 架构

IBM 成功地构建了第一台采用 RISC（精简指令集计算机）架构的原型计算机。基于 IBM 科学家 John Cocke 在 20 世纪 70 年代初的发明，RISC 概念简化了运行电脑所需给予的指令，使它们速度更快、更强大。现在 RISC 架构是大多数企业服务器的基础，并被普遍认为是未来的主导计算架构。如果您考虑分析和建模在今天所需的计算功能，以及明天将需要的功能，就会希望自己的大数据合作伙伴拥有真正发明了高性能计算 (HPC) 芯片的制造设计商，并且可以在现代的大数据奇迹（如 Watson、Jeopardy！）中找到他，冠军中的冠军。

1988：NSFNET

IBM 与 National Science Foundation (NSF) 以及我们在 MCI 和 Merit 的合作伙伴共同努力，设计、开发并部署了一个新的

高速网络（名称为 NSFNET），以连接大约 200 所美国大学和 6 个总部在美国的超级计算机中心。NSFNET 快速成为 Internet 的主要骨干，以及点燃全球 Internet 革命的火花。NSFNET 大大提高了 Internet 的速度和容量（将主干链路带宽从56kb/s 增加到 1.5Mb/s，再到 45Mb/s），大大提高了 Internet 的可靠性，当 Internet 的控制权在 1995 年 4 月被转移给电信运营商和商业 Internet 服务供应商时，其覆盖范围已达到 93 个国家中超过 5 千万用户。Internet 规模的数据移动专业知识，导致对提供能够在 Internet 规模工作的解决方案所需的硬件和软件都进行了重大投资。此外，我们有大量网络安全和网络监测大数据模式使用了包分析，该技术利用了我们在 NFSNET 上的开创性工作。

1993：可扩展并行系统

IBM 帮助开拓的一项技术可以联合多个计算机处理器，并分拆复杂的数据密集型作业，以加快其完成速度。这项技术被应用于天气预报、石油勘探和制造业。DB2 Database Partitioning Facility (DB2 DPF) — 大规模并行处理 (MPP) 引擎，用于在共享架构上拆分和征服查询问题，可以在 IBM Smart Analytics System 中找到它——它在几十年来一直被用于解决大数据集问题。虽然我们还没有谈到 Hadoop 中的技术，在第II部分中，您将了解名为 MapReduce 的东西，以及它实现并行性（大型独立的分布式计算机处理同一个问题）的方法如何利用一个在概念上与 DB2 DPF 技术非常类似的方法。

1996：深雷

在 1996 年，IBM 开始探索“天气的生意”，超本地化的短期预报，为客户自定义的天气建模。现在，新的分析软件以及像城市和能源事业组织对智慧运营的需求，正在改变这些服务的市场环境。

正如 IBM Research 中深雷项目的首席科学家 Lloyd Treinish 解释的，该方法的重点并不是人们在电视上看到哪一种天气报告，

它关注的是天气可能对特定场所中的企业所带来的运营问题——传统气象学无法解决的挑战。

例如，公共气象资料的目的不是带着合理的信心预测从现在开始三小时的风速对于在 10 米跳台上高风险的竞赛是否可接受。这种有针对性的预测是 IBM 和美国国家气象服务的管理机构，即美国国家海洋和大气管理局 (NOAA)，在 1995 年就接受的挑战。

这种大规模计算问题集与我们每天都在做的客户工作直接相关，包括我们在第 2 章中提到的 Vestas。这也是一个很好的例子，说明 IBM 关注分析结果（通过一个平台产生），而不是一个停止在基础架构的大数据承诺。虽然这里的计算环境肯定是有趣的，但如何将计算基础架构投入工作才是真正的创新——与我们今天在大数据领域中看到的动态完全相同。

1997：深蓝

32 个节点的 IBM RS/6000 SP 超级计算机 Deep Blue（深蓝）击败了国际象棋的世界冠军 Garry Kasparov，这是在锦标赛式的竞赛中计算机战胜世界冠军棋手的第一个已知实例（将它与近二十年后的 Watson 比较，Watson 是一台“学习型”计算机，这又是一个新的转折点）。像前面的例子一样，使用大规模并行处理是 Deep Blue 获得成功的原因。将任务分解成较小的子任务，然后跨多台计算机并行执行它们，这就是 Hadoop 集群的基础。

2000：Linux

2000 年，当 IBM 宣布自己将接受 Linux 作为其系统战略时，Linux 获得了一次重要的推动。一年后，IBM 投资了 10 亿美元支持 Linux 迁移，接受它为 IBM 服务器和软件的一个操作系统，在与其许可有关的不确定性期间加快补偿用户。IBM 的行动吸引了全球各地 CEO 和 CIO 的关注，并帮助 Linux 得到商业世界的接受。Linux 事实上是

面向 Hadoop 的操作系统，您可以看到，您的大数据合作伙伴在 Hadoop 的底层操作系统方面拥有十年以上的经验。

2004：蓝色基因

Blue Gene（蓝色基因）超级计算机架构由 IBM 开发，其目标是 PFLOPS 范围的性能（每秒超过一万亿次浮点操作）。2004 年 9 月，一台 IBM Blue Gene 计算机打破了 PFLOPS 的世界纪录。在接下来的四年中，一台使用 IBM Blue Gene 架构的计算机保持着世界上最快的超级计算机这个头衔。Blue Gene 已在广泛的应用程序中使用，包括人类基因组测绘、医学疗法调查、气候趋势预测等。在 2009 年，因 Blue Gene 所取得的成就，美国总统 Barack Obama 授予 IBM 它的第七个 National Medal of Technology and Innovation （技术创新国家奖章）。

2009：第一个全国性的智能能源和水网格

岛国马耳他向 IBM 求助，以缓解它的两个最紧迫的问题：水资源短缺和能源成本暴涨。结果是智能水资源和智能电网系统的组合，采用物联化的数字仪表监控浪费，奖励资源的高效使用，防止盗用，减少对石油和经过处理的海水的依赖。马耳他和 IBM 正在一起建设世界上第一个国家智能公用设施系统。IBM 已经解决了您今天所面临的许多问题，并且能够带着广泛的领域知识来帮助您。

2009：流计算

IBM 宣布提供其 Streams 计算软件，这是一个突破性的移动数据分析平台。流计算动态收集多个数据流，使用先进的算法来提供近乎瞬时的分析。在传统的数据分析策略中，数据被收集到一个数据库中，并被搜索或查询答案，流计算颠覆了这种策略，可用于需要立即作出决定的复杂动态情况，如预测疫情的蔓延或监测早产儿的条件变化。流计算的工作被转移到 IBM Software Group，并且以 InfoSphere Streams 的身份作为 IBM 大数据平台的一部分被投入商用（我们将在第 6 章中介绍它）。在本书中，我们讨论移动数据和静止数据分析，以及您如何能创建一个

周期性的系统，它可以学习并提供前所未有的愿景；我们相信，只有 IBM 可以在这个时候将这一点作为合作伙伴关系的一部分提供。您可能想知道在运行分析时 Streams 可以维持什么样的吞吐量。在一个客户环境中，Streams 每秒分析 500,000 个呼叫详细记录 （CDR，一种用于蜂窝通信的详细记录），每天处理超过 60 亿个 CDR，每年处理超过 4 PB 数据！

2009：云

IBM 的综合能力使 Enterprise Cloud（企业云）的承诺变成现实。IBM 已经帮助数以千计的客户获得了云计算的好处：仅在 2010 年就有超过 2000 个私有云协议，IBM 每天管理着数十亿个基于云的交易以及数百万的云用户。IBM 本身也在大量使用云计算，并体验着巨大的效益，如创新理念的加速部署，以及从开发中每年节省 1500 万美元以上。然而，要获得可以解决当今市场现实问题的实质效益，并不仅仅是实施云功能这么简单——还涉及组织如何战略性地利用新方法来访问和混合数据。这个巨大的潜力往往未能得到满足，因为之前使用云技术主要是为了使 IT 更容易、更便宜、更快速。IBM 认为，云需要与转型有关。虽然它显然包括如何交付 IT，但愿景被延伸到考虑交付哪些洞察；这样做需要可以处理数据的数量、种类和速度的平台功能，更重要的是，需要能够构建和部署所需的分析，以带来转型功能。

2010 ： GPFS SNC

最初发布于 1998 年的 IBM General Parallel File System (GPFS) 是一个高性能 POSIX 兼容的共享磁盘集群文件系统，在存储区域网络 (SAN) 上运行。如今，许多超级计算机、DB2 pureScale、许多 Oracle RAC 实施等都在使用 GPFS。GPFS 具备跨多个磁盘划出数据块的能力，并且能够并行读取数据块，从而对在多个集群节点上执行的应用程序提供并发的高速文件访问。此外，GPFS 提供高可用性、灾难恢复、

安全性、分层的存储管理等。GPFS 被扩展至无共享集群（称为 GFPS-SNC）上运行，并因其最具创新性的存储解决方案获得了 SC10 Storage Challenge 2010（SC10 存储挑战 2010）奖："它旨在可靠地存储 PB 级到 EB 级的数据，同时以竞争解决方案两倍的速度处理高度并行的应用程序，如 Hadoop 分析。"一个知名的企业级文件系统针对 Hadoop 适用性的扩展，这对于许多组织来说很有吸引力。

2011：Watson

IBM 的 Watson 利用领先的问题-回答（Question-Answering，QA）技术，使计算机可以处理和理解自然语言。Watson 还实施了一个？的学习行为，可以理解以前正确和不正确的决策，它甚至有可能对未来的决策和领域知识应用风险分析。Watson 采用大规模并行分析功能来模仿人类的思维能力，以了解每句话背后的实际意义，区分相关和不相关的内容，并最终表现出提供精确的最终答案的信心。2011 年 2 月，Watson 创造了历史，不仅是在电视台历史悠久的智力竞赛节目 *Jeopardy!* 中与人类对战的第一台计算机，而且还在对阵冠军 Ken Jennings 和 Brad Rutter 时获得了压倒性胜利。在不同知识集上的 Decision Augmentation （决策增强）是大数据技术的一个重要应用，Watson 使用 Hadoop 存储和预处理其知识主体，这种用法是 BigInsights 向前发展的一项基础功能。在这里再强调一下，如果您在选择供应商时只关注是否支持 Hadoop，您就会错过关键的价值（理解和洞察的发现），而不仅仅是处理数据。

IBM Research：InfoSphere BigInsights 战略的核心部分

利用 IBM Research 的创新纪录，这一直是 IBM 大数据战略和平台中深思熟虑的一部分。除了 Streams，IBM 在 IBM Research 实验室中启动了 BigInsights，并在它全面上市的一年多以前将它转移到 IBM Software Group (SWG)。部分 IBM Research 发明，如 *Advanced Text Analytics Toolkit*（之前称为 SystemT）和 *Intelligent Scheduler*

（在 Hadoop 提供的功能上提供工作负载治理，并超越 Hadoop 提供的功能——它之前被称为 *FLEX*）随第一个 BigInsights 版本提供。其他发明，如 GPFS-SNC（超过 12 年来一直是企业性能和可用性的代名词）、*Adaptive MapReduce* 和 *Machine Learning Toolkit*（以前您可能已经听见它被称为 System ML）在今天已可用，或者很快会被发布。（您将注意到 BigInsights 开发团队已经针对特性交付采纳了一个启动性的思维——它们很快就会来到，并且往往不同于传统的软件版本。）我们将在第II部分中介绍所有这些技术。

IBM Research 是推动大数据分析和强化 Hadoop 生态系统的基础引擎。当然，BigInsights 的工作不仅是由 IBM Research 驱动：我们在硅谷、印度和中国的大数据开发团队已经采用了来自 IBM Research 的技术，并进一步增强它们，这带来了我们基于外部和内部客户大量输入的第一个商用版本。

IBM 的内部代码集市和大数据

Hadoop 是 Apache 的顶级项目。IBM 之外没有多少人知道的一件事是，IBM 有自己内部版本的 Apache 模型，在该模型中，团队可以利用其他团队的软件代码和项目，并在自己的解决方案中使用它们，然后将丰富后的代码贡献回中央 IBM 社区。例如，DB2 pureScale 利用了在 Tivoli System Automation、GPFS 和 HACMP 中可以找到的技术。在开发 DB2 pureScale 时，大量增强被放进这些技术中，然后在其各自的市售产品中物化为增强功能。尽管这种共享已经持续了很长一段时间，它与新兴技术关联的程度和速度已大大加快，并将成为 IBM 的大数据合作伙伴关系和旅程的重要组成部分。

正如 IBM 对其创新的 Mashup（混搭）技术所实现的一样，该技术由 Information Management 和 Lotus 合作开发，但很快就被 IBM Enterprise Content Management、Cognos、WebSphere 和 Tivoli 所利用，IBM 大数据团队已经看到关于其 Hadoop 工作的类似代码共享。IBM Cognos Consumer Insight (CCI) 是一个很好的正式上市产品示例，因为 IBM 大数据组合内的这种共享，让它更迅速地进入了市场。CCI 运行在 BigInsights 上（与其他 IBM

产品一样），它通过分析公开提供的大量 Internet 内容，使营销专业人员更精确，更敏捷，并能更迅速地响应客户通过社交媒体表达的需求和意见。CCI 利用 BigInsights 收集、存储和执行在此内容上所需的基础分析流程，并通过应用程序级别的社交媒体分析和可视化来增强该分析流程。BigInsights 可以利用 CCI 收集到的数据执行后续分析作业，包括使用内部企业数据源查找 CCI 识别的行为与企业影响或驱动该行为的做法之间的相关性（优惠券、商品组合、新产品等）。

我们一直鼓励这种级别的代码共享，其原因有若干个，但也许最重要的是用法的多样性带来了需求和领域专业知识的多样性。我们认识到，我们正在旅途中，所以招募尽可能多的导游会有帮助。当然，在旅途中时，挑选经验丰富的导游会有帮助，因为通过一个潜水教练学习跳伞可能不会有什么好结局。相关的专业知识很重要。

领域专业知识很重要

应用深入的领域专业知识，解决以前无法解决的大数据问题，这样的例子有数百个。本节介绍了两个例子，一个来自媒体行业，另一个来自能源行业。

最近，我们使用 BigInsights 帮助一家媒体公司限制其媒体流在没有许可的情况下被分发的频率。答案有很多，事实上，比它所预料的要多得多。这是该技术的成功使用，但它是否解决了业务问题呢？否——这其实只代表着业务问题的开始。简单的应对应该是遏止那些“没有明确书面同意”的尝试，但那会是正确的业务决策吗？不一定，因为这个观众原来没有得到很好的服务，同时又渴求消费，虽然没有经过所有者许可就使用有版权的材料明显是一件坏事，但这是一个变相的机会：市场在告诉该公司，全新地域的人们对其版权资产感兴趣——这代表了一个它们以前没有看到的新机会。从业务战略的角度来看，达成正确的决策极少（如果有的话）是技术决策。这是领域专业知识至关重要的位置，可确保该技术被适当应用在更广泛的业务背景中。有一个很好的例子可以说明这种

专业知识的应用，就是 IBM 的智慧地球工作，它无疑是将领域专业知识应用到了规模和种类都在不断增加的数据集。

让我们来看看能源领域。据估计，到 2035 年，全球能源需求将上升近 50%，而可再生能源将开始对这种增加的需求做出重大的贡献，传统的石油和天然气将仍需要占据总需求增加量中的 50%。鉴于需要获得的石油和天然气储备位置越来越遥远，这将是不小的成就。可盈利且安全地查找、获得、提炼和运输这些储备，要求通过新途径来了解所有部分如何组合在一起。在石油和天然气的生产周期中所产生的数据量和多样性是惊人的。每口油井可能有超过 20,000 个独立传感器，每口油井每天产生数 TB（或更多）的数据。仅仅是存储油田或相关油井的输出就可能是一个每年 10 PB（或更多）的挑战，现在再加上计算需求，以关联所有这些井的行为。虽然明显有可能从所有这些数据中获得重要的了解，但知道从哪里开始，以及如何运用它是至关重要的。

当然，一旦可用于消费，自然就会跟进优化能源的配送方式。IBM Smart Grid 是一个很好的例子，说明领域专业知识与大数据的交集。IBM 帮助公用事业企业在其电网上添加一层数字智能。这些智能电网使用传感器、仪表、数字控制和分析工具，在整个运营（从发电厂到插头）中自动化、监视和控制能源的双向流动。电力公司可以优化电网性能，防止断电，更快地恢复中断，让消费者可以管理每个连接到电网的设备的能源使用。

IBM 大数据平台支持做到这一点所需的所有数据的收集，但更重要的是，该平台的分析引擎可以找到条件的相关性，对电网可以如何优化运行提供了新的认识。正如您可以想像到的，能够存储事件很重要，但能够理解和链接在该领域中的事件则是关键。我们喜欢将这比作在噪声中找到信号。IBM 在这个空间所开发的技术、分析引擎和领域专业知识，同样适用于全面了解客户的情况，当社交媒体成为组合的一部分时尤其如此。

第II部分

大数据：技术视角

4

一切都关于 Hadoop：大数据术语

现在您已经阅读了第 I 部分，这应该是显而易见的，但我们有一种预感，您在拿起这本书之前就已经知道了——在我们的信息中尚未挖掘出来的潜力像山一样高。一直到现在，对这些海量数据进行分析的成本都过于高昂。当然，不挖掘这些信息也有一个惊人的机会成本，因为这个尚未被分析的信息的潜力是接近于无限的。我们在这里不只是谈论无处不在的“差异化竞争”营销口号；我们谈论创新、发现、关联，以及几乎任何其他东西，它们利用来自于您今天的工作方式的更多有形结果和洞察，可能使您明天的工作方式有很大差异。

个人和组织都在试图从多个不同角度解决这个问题。当然，目前在海量数据分析的普及性方面领先的是称为 Hadoop 的一个开源项目，它作为 IBM InfoSphere BigInsights (BigInsights) 平台的一部分提供。很简单，BigInsights 利用企业级安全性、治理、可用性、与现有数据存储的集成、可以简化和提高开发人员的生产力的工具、可扩展性、分析工具包等，接受、强化并扩展 Hadoop 开源框架。

在我们编写这本书的时候，我们认为针对 Hadoop 本身写一章内容是有益的，因为 BigInsights 是（而且将永远是）基于没有分歧的核心 Hadoop 分发内容的，而对

Apache Hadoop 项目的向后兼容性将始终保持。简单来说，为 Hadoop 编写的应用程序始终可以在 BigInsights 上运行。本章无论如何不会让您成为一个 Hadoop 专家，但阅读完之后，您就会明白核心 Hadoop 技术背后的基本概念，对于水冷机的非技术人员来说，您甚至会显得相当厉害。如果您刚接触 Hadoop，本章最适合您。

事实：Hadoop 的历史

Hadoop (http://hadoop.apache.org/) 是 Apache Software Foundation 的一个顶级 Apache 项目，它用 Java 编写。无论从哪一点来看，您可将 Hadoop 想像成一个构建于分布式集群文件系统之上的计算环境，专门针对非常大型的数据操作而设计。

Hadoop 的灵感来自于 Google 在其 Google（分布式）File System (GFS) 和 MapReduce 编程模式上的工作，在该模式中，工作被分解为多个 mapper 和 reducer 任务，以操作在整个服务器集群中存储的数据，实现大规模并行化。MapReduce 并不是一个新概念（IBM 在 2007 年 10 月联手 Google 进行了一些关于 MapReduce 和 GFS 联合大学研究，以了解大规模 Internet 的问题）；然而，Hadoop 已实际被应用到更广泛的用例。与事务系统不同，Hadoop 旨在通过一个高度可扩展的分布式批量处理系统，对大型数据集进行扫描以产生其结果。Hadoop 的重点并不是思想？？速度的响应时间、实时仓库，或创新的事务速度；它的重点是从可扩展性和分析的角度发现和使曾经的几乎不可能变成可能。Hadoop 方法建立于功能到数据的模型，而不是数据到功能的模型；在这个模型中，因为如此多的数据，所以将分析程序发送给数据（我们在本章稍后将详细解释这一点）。

Hadoop 是一个相当奇怪的名字（您将在 Hadoop 世界中找到许多奇怪的名字）。现在阅读任何有关 Hadoop 的书，几乎都从作为这个项目的吉祥物的名字开始，所以让我们也从这里开始吧。Hadoop 实际上是创始人 Doug Cutting 的儿子给自己的毛绒玩具大象起的名字。Cutting 在为他的项目构思一个名称时，显然是在寻找某种很容易说，并且没有什么特别代表意义的东西，所以他儿子的玩具名字似乎非常适合。

Cutting 的命名方法拉开了收集各种奇怪名字的序幕（正如您很快就会发现），但说实话，我们喜欢它。（我们在写这本书时回想起与我们孩子的玩具有关的一些名字，我们很高兴 Cutting 是为这项技术，而不是为我们起绰号；Pinky 和 Squiggles 听起来像是不太好的选择。）

一般认为 Hadoop 有两个部分：一个文件系统 (*Hadoop Distributed File System*) 和编程模式 (*MapReduce*)——稍后再详细讲述它们。Hadoop 中一个关键组件是内置在环境中的冗余性。不仅是数据冗余地存储在整个集群内的多个地方，编程模型也是这样，通过在集群中的多个服务器上运行程序的多个部分，可预期失败并自动解决这种问题。由于这种冗余性，我们可以实现在一个非常大型的商品组件集群中分发数据及其相关的编程内容。众所周知，商品硬件组件将失败（尤其是当您有非常多的商品硬件组件时），但这种冗余性提供了容错，以及让 Hadoop 集群自愈的能力。这使得 Hadoop 可以跨廉价机器的大型集群向外扩展工作负载，以处理大数据问题。

与 Hadoop 相关的项目有许多，我们在本书中介绍了其中一些（还有一些项目因为规模太大，我们没有进行介绍）。一些比较著名的 Hadoop 相关项目包括：Apache Avro（用于数据序列化）、Cassandra 和 HBase（数据库）、Chukwa（一个监控系统，在设计时特别考虑了大型分布式系统）、Hive（为数据聚合与汇总提供类似于 SQL 的专用查询）、Mahout（机器学习库）、Pig（一个高层次的 Hadoop 编程语言，为并行计算提供一个数据流语言和执行框架）、ZooKeeper（为分布式应用程序提供协调服务）等。

Hadoop 的组件

Hadoop 项目包括三部分：*Hadoop Distributed File System (HDFS)*、*Hadoop MapReduce* 模型和 *Hadoop Common*。要理解 Hadoop，您必须理解文件系统的底层基础架构以及 MapReduce 编程模型。让我们先来谈谈 Hadoop 的文件系统，它使应用程序可以跨多个服务器运行。

Hadoop Distributed File System（分布式文件系统）

要理解一个 Hadoop 集群如何可能扩展到数百（甚至数千）个节点，您必须从 HDFS 开始。Hadoop 集群中的数据被分解成多个更小的片（称为块），并分布在整个集群中。通过这种方式，就可以在较大型数据集的较小子集上执行 map 和 reduce 函数，并提供大数据处理所需要的可扩展性。

Hadoop 的目标是在一个非常大型的集群中使用常用的服务器，其中每个服务器都有一套廉价的内部磁盘驱动器。为了实现更高性能，MapReduce 试图将工作负载分配给可存储要处理的数据的这些服务器。这被称为*数据局部性*。（正因为这一原则，不建议在 Hadoop 环境中使用存储区域网络 (SAN)，或网络附加存储 (NAS)。对于使用 SAN 或 NAS 的 Hadoop 部署，额外的网络通信开销可能导致性能瓶颈，对于较大型的集群尤其如此。）现在花点时间，想想一个由 1000 台计算机组成的集群，每台计算机有 3 个内部磁盘驱动器，然后再考虑一个由 3000 个*廉价*驱动器 + 1000 台*廉价*服务器组成的集群的故障率！

我们在这里很可能意见一致了：您将在一个 Hadoop 集群中体验到的组件平均无故障时间 (MTTF) 可能类似于您的孩子夹克上的拉链：它总是会坏的（足够诗意的是，拉链似乎只在您真正需要它们的时候才会坏）。Hadoop 很酷的是，与廉价硬件相关的 MTTF 率这一现实其实得到了充分的理解（如果您愿意，这是一个设计点），Hadoop 的其中一部分强大力量是，它具有内置的容错和故障补偿功能。这对于 HDFS 也是同样，在 HDFS 中，数据被划分成多个块，这些块的副本存储在 Hadoop 集群中的其他服务器上。也就是说，一个单独的文件实际上被存储为多个较小的块，这些小块在整个集群中跨多个服务器被复制。

想像一个包含了美国所有人电话号码的文件；姓氏以 A 开头的人可能被存储在服务器 1 上，姓氏以 B 开头的人在服务器 2 上，以此类推。在 Hadoop 的世界中，这本电话簿的小块将存储在整个集群中，为了重构整个电话簿，您的程序需要来自集群中每台服务器的多个块。为了在组件失败时实现可用性，HDFS 默认将这些较小的块（参见图 4-1）复制到两个额外的服务器上。（这种冗余性可以

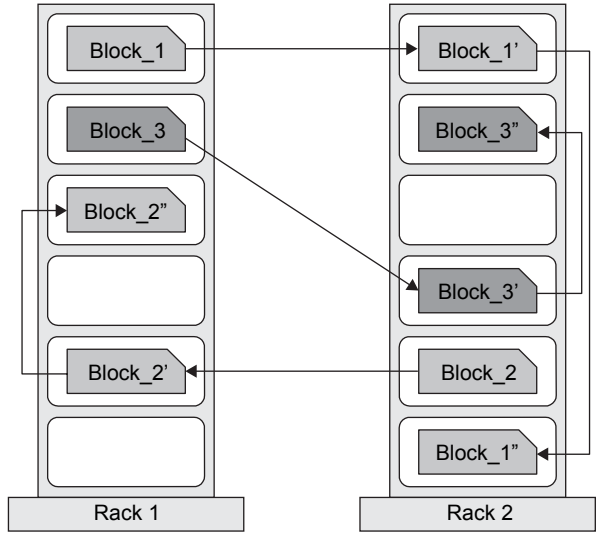

图 4-1 *数据块如何被写入 HDFS 的一个示例。请注意每个块如何（默认）被写入 3 次，并且至少有一个块被写入不同的服务器机架，以实现冗余性*

图字：
Rack：机架

将每一个文件为基础或针对整个环境进行增加或减少；例如，一个开发 Hadoop 集群通常并不需要任何数据冗余）。这种冗余性提供了多种好处，最明显的是可用性更高。此外，这种冗余性使 Hadoop 集群将工作分解成较小的数据块，并在集群中的所有服务器上运行这些作业，实现更好的可扩展性。最后，您可以得到数据局部性的好处，在使用大型数据集时，这是至关重要的。我们稍后将在本章中详细介绍这些重要的优势。

HDFS 中的一个数据文件被划分成多个块，对于 Apache Hadoop，这些块的默认大小为 64 MB。对于较大的文件来说，提高块的大小是一个好办法，因为这将大大减少 NameNode 所需元数据的数量。预期的工作负载是另一个考虑因素，因为非顺序访问模式（随机读取）在使用较小的块大小时会执行得更好。在 BigInsights 中，默认的块大小是 128 MB，因为根据 IBM Hadoop 从业者的经验，最常见的部署涉及较大的文件和顺序读取的工作负载。这个块大小比在其他环境所使用的要大得多，例如，典型的文件

系统磁盘上的块大小是 512 字节，而关系型数据库存储的数据块的大小通常从 4 KB 到 32 KB 不等。记住，Hadoop 旨在扫描非常大的数据集，所以它使用一个非常大的块大小是有意义的，这样每个服务器可以同时处理更大的数据块。在整个集群中进行协调，会产生大量开销，所以不需要将数据发送到其他节点就能够在本地处理大数据块，这将有助于提高性能，以及*开销对实际工作*的比率。还记得每个数据块默认存储在 3 个不同的服务器上；在 Hadoop 中，这是由在幕后工作的 HDFS 所实施的，确保至少有两个块被存储在一个单独的服务器机架，以提高可靠性，在损失了整个机架的服务器的情况下仍然能够提供服务。

Hadoop 的所有数据放置逻辑由一个称为 *NameNode* 的特殊服务器进行管理。这个 NameNode 服务器跟踪在 HDFS 中的所有数据文件，如块的存储位置等。NameNode 的所有信息都存储在内存中，这使得它能够对存储操作或读取请求提供快速的响应。现在，我们知道您在想什么：如果只有一个 NameNode 为整个 Hadoop 集群服务，您需要知道将该信息存储在内存中会创建一个单点故障 (SPOF)。出于这个原因，我们强烈建议您为 NameNode 选择的服务器组件应比 Hadoop 集群中的其他服务器更强健，最大限度地减少故障的可能性。此外，我们也强烈建议您为存储在 NameNode 中的集群元数据执行定期备份流程。此元数据中的任何数据丢失，都将导致在集群中相应数据的永久丢失。在编写本书时，下一个版本的 Hadoop（版本 0.21）准备包括定义 BackupNode 的功能，BackupNode 可以作为 NameNode 的冷备机。

图 4-1 代表由 3 个数据块组成的一个文件，其中一个数据块（表示为 block_n）被复制到两个额外的服务器（表示为 block_n' 和 block_n''）。第二个和第三个副本被存储在一个单独物理机架上的单独节点中，以实现额外的保护。

我们详述了 HDFS 如何存储数据块，以便向您简要介绍这个 Hadoop 组件。Hadoop MapReduce 应用程序框架很棒的一点是，与之前的网格技术不同，开发人员不必处理 NameNode 的概念和数据存储的位置——Hadoop 为您完成这些工作。当您启动一个 Hadoop 作业，并且

应用程序必须读取数据，并开始处理编程的 MapReduce 任务时，Hadoop 将联系 NameNode，找到拥有执行该作业所需访问的数据部分的相应服务器，然后发送您的应用程序，使其在这些节点上本地运行。（我们会在下一节详细介绍 MapReduce）。同样，创建一个文件时，HDFS 会自动与 NameNode 通信，在特定的服务器上分配存储空间并执行数据复制。重要的是要注意，当您使用数据时，您的 MapReduce 代码不需要直接引用 NameNode。与 NameNode 的交互大部分在 Hadoop 集群中各个服务器上调度作业时已完成。这大大减少了作业执行过程中与 NameNode 的通信，有助于提高解决方案的可扩展性。总之，NameNode 处理可描述文件存储位置的集群元数据；MapReduce 作业所处理的实际数据永远不会流过 NameNode。

在本书中，我们谈论 IBM 如何将企业功能带给 Hadoop，在该特定领域中，IBM 使用其数十年的经验和研究，充分利用其无处不在的企业 IBM General Parallel File System (GPFS) 来减轻这些问题的影响。GPFS 最初只在 SAN 技术上运行。在 2009 年，GPFS 被扩展到在无共享集群（称为 GPFS-SNC）上运行，它针对像 Hadoop 等用例。GFPS-SNC 提供了 HDFS 所不具备的许多优点，其中一个优点解决了上述 NameNode 问题。在 GPFS-SNC 内实施的 Hadoop 运行时，不一定要与这个特别的 SPOF 问题进行竞争。GPFS-SNC 使您能够建立一个更加可靠的 Hadoop 集群（其中还包括其他好处，如易于管理和性能）。

除了所提出的有关单一 NameNode 的问题之外，一些客户还指出，HDFS 不是 Portable Operating System Interface for UNIX (POSIX) 兼容的文件系统。这意味着，几乎所有您在与文件进行交互时可能使用的熟悉命令（复制文件、删除文件、写入文件、移动文件等）都以不同形式在 HDFS 中可用（有语法差异，在某些情况下有功能限制）。为了解决这个问题，您必须编写自己的 Java 应用程序执行某些功能，或培训您的 IT 员工，学习不同 HDFS 命令来管理和操作文件系统的文件。我们将在本章的后面更详细地讨论这一主题，但在这里

我们希望您注意，这又是 BigInsights 提供给 Hadoop 环境进行大数据处理的另一种“企业方法”。GPFS-SNC 完全兼容 IEEE 定义的 POSIX 标准，该标准定义了一个 API、shell 和实用程序接口，以提供跨不同的 UNIX（如 AIX、Apple OSX 和 HP-UX）的兼容性。

MapReduce 的基础知识

MapReduce 是 Hadoop 的心脏。正是这种编程模式，实现了跨越一个 Hadoop 集群中数百或数千台服务器的大规模扩展性。MapReduce 概念对于那些熟悉集群向外扩展的数据处理解决方案的人来说相当易于理解。对于刚接触这个主题的人来说，它可能有些难以掌握，因为它不是人们以前一般接触过的某些概念。如果您刚接触 Hadoop 的 MapReduce 作业，别担心：我们打算以一种让您快速了解它的方式来形容它。

术语 *MapReduce* 实际上指的是 Hadoop 程序所执行的两个独立的、不同的任务。第一个是 map 作业，它拿出一组数据，并将它转换成另一组数据，其中每个元素都被分解成多个*元组*（键/值对）。reduce 作业将 map 的输出作为输入，并将那些数据元组组合成较小的元组集。正如 MapReduce 这个名字的顺序所示，reduce 作业始终在 map 作业后执行。

让我们来看看一个简单的例子。假设您有 5 个文件，每个文件包含两个列（在 Hadoop 术语中的一个*键*和一个*值*），分别代表一个城市以及在该城市多个测量日中所录得的相应温度记录。当然，我们已经使这个例子很简单，以便读者能够很轻松地学习。您可以想像得到，一个真正的应用程序不会这么简单，因为它可能含有百万甚至数十亿行数据，并且它们也未必是格式非常整齐的行，事实上，无论您需要分析的数据量有多大或多小，我们在这里介绍的关键原则仍保持不变。无论是哪种方式，在这个例子中，*城市是键，*而*温度是值*。

下面的代码片段显示了来自我们的测试文件的一个示例数据，（顺便说一句，这些温度以摄氏度表示，您不必去拿帽子和手套）：

Toronto, 20
Whitby, 25
New York, 22
Rome, 32
Toronto, 4
Rome, 33
New York, 18

在我们收集的所有数据中，我们跨所有数据文件找出每个城市的最高温度（注意，每个文件中可能有代表多次测量的相同城市数据）。使用 MapReduce 框架，我们可以将这分解为 5 个 map 任务，每个 mapper 处理这 5 个文件的其中一个，mapper 任务遍历数据并返回每一个城市的最高温度。例如，一个 mapper 任务从上述数据所产生的结果看起来如下：

(Toronto, 20) (Whitby, 25) (New York, 22) (Rome, 33)

假设其他 4 个 mapper 任务（处理此处没有显示的其他 4 个文件）所产生的中间结果如下：

(Toronto,	18)	(Whitby,	27)	(New York,	32)	(Rome,	37)
(Toronto,	32)	(Whitby,	20)	(New York,	33)	(Rome,	38)
(Toronto,	22)	(Whitby,	19)	(New York,	20)	(Rome,	31)
(Toronto,	31)	(Whitby,	22)	(New York,	19)	(Rome,	30)

所有这些 5 个输出流将被送入 reduce 任务，reduce 任务综合输入结果，并为每个城市输出单个值，产生的最终结果集如下：

(Toronto, 32) (Whitby, 27) (New York, 33) (Rome, 38)

打个比方，您可以将 map 和 reduce 任务视为在古罗马时期的一种人口普查方式，人口普查局派遣其工作人员到帝国的每个城市。在每个城市中的每个人口普查员都将负责对在这个城市的人口进行计数，然后将其结果返回首都。在那里，来自每个城市的结果将被缩减为单一的计数（所有城市的总和），以确帝国的定总人口。这种将人员并行地 *mapping* 到城市，然后综合结果 (*reducing*) 的方法，比起以串行方式派遣一个人去数帝国的所有人口，要高效得多。

在 Hadoop 集群中，MapReduce 程序指一个*作业*。一个作业按顺序分解成被称为*任务*的多个块来执行。

一个应用程序将作业提交到 Hadoop 集群中的一个特定节点，它运行一个称为 *JobTracker* 的守护进程。JobTracker 与 NameNode 进行通信，在整个集群中找出该作业所需的所有数据的存储位置，然后将作业分解成供集群中每个节点处理的 map 和 reduce 任务。这些任务将被安排在集群中数据所在的节点上。注意，一个节点可能会得到一个任务，该任务所需要的数据可能对该节点而言不在本地。在这种情况下，该节点必须要求数据通过互连网络进行发送，这样才能执行任务。当然，这种方法的效率并不高，所以 JobTracker 试图避免这种情况，并尝试在数据存储的位置安排任务。这就是我们在前面所介绍的数据局部性的概念，在使用大量数据时，它是非常重要的。在 Hadoop 集群中，有一组持续运行的守护进程被称为 TaskTracker 代理，它们监控每个任务的状态。如果一个任务无法完成，会将该失败状态报告给 JobTracker，然后 JobTracker 将该任务重新安排给集群中的另一个节点。（您可以规定在整个作业被取消之前，该任务可以尝试多少次。）

图 4-2 显示了一个 MapReduce 流的示例。从中可以看到，多个 reduce 任务可以有助于提高并行性和提高作业的整体性能。在图 4-2 的示例中，map 任务的输出必须（由键值）定向到相应的 reduce 任务。如果我们将前面最高温的例子应用到该图，所有键值是 Toronto 的记录都必须发送给相同的 reduce 任务，以

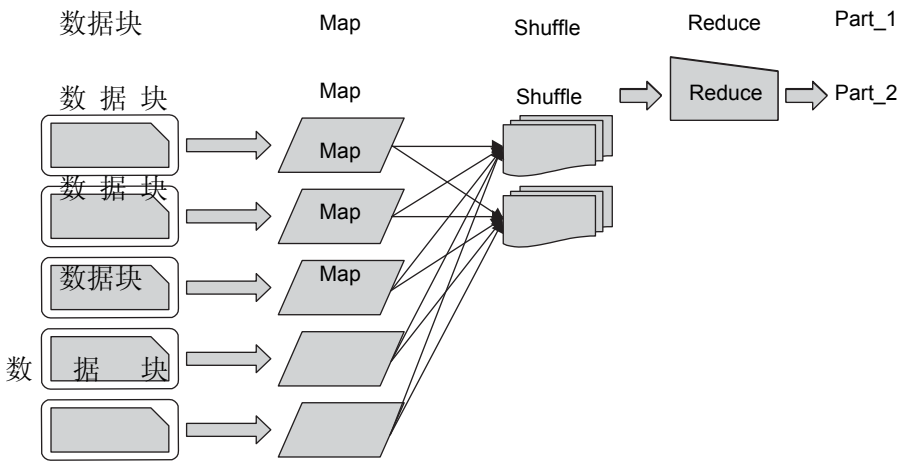

图 4-2 *一个简单 MapReduce 作业中的数据流*

产生一个准确的结果（一个 reducer 必须能够看到 Toronto 的所有温度，以确定该城市的最高温度）。这样将记录定向到 reduce 任务，被称为 Shuffle，它从 map 任务获得输入，并将输出定向到一个特定的 reduce 任务。Hadoop 让您可以选择在每个 map 任务的输出上执行本地聚合，然后再通过称为 *Combiner* 的本地聚合将结果发送给 reduce 任务（但它在图 4-2 中没有显示）。显然，运行多个 reduce 任务会涉及更多工作和开销，但对于非常大型的数据集而言，使用多个 reducer 可以提高整体性能。

所有原生运行在 Hadoop 下的 MapReduce 程序都是用 Java 编写的，它是 Java Archive 文件 (jar)，由 JobTracker 分发到多个 Hadoop 集群节点，以执行 map 和 reduce 任务。有关 MapReduce 的更多详细信息，您可以查看 Apache Hadoop 文档的教程，该教程利用了相当于 Hadoop 无处不在的 Hello World 编程语言：WordCount。WordCount 是一个易于理解的示例，它拥有运行示例所需的全部 Java 代码。

当然，如果您正在寻找最快和最简单的方式利用 Hadoop 启动并运行系统，请查看 BigDataUniversity.com，并下载 InfoSphere BigInsights Basic Edition (www.ibm.com/software/data/infosphere/ biginsights/basic.html)。它有一些非常棒的 IBM 附加功能（例如，为您精简了整个启动和运行体验，让您使用在任何商用软件中相同的方式获得一个正常运行的 Hadoop 集群）等。最重要的是，它是百分之百免费的，您可以选择性地购买 BigInsights 的 Basic Edition 的支持合同。当然，读完本书时，您将完全掌握 IBM InfoSphere BigInsights Enterprise Edition 如何通过接受和扩展 Hadoop 堆栈，提供预期来自其他企业系统的相同功能。

Hadoop 常用组件

Hadoop 常用组件是支持各种 Hadoop 子项目的一组库。在本章前面，我们附带地提到一部分这些组件。在本节中，我们希望花点时间讨论文件系统 shell。如前所述（这真的是非常重要的一点，这就是为什么我们再次注意到它），HDFS *不是* 一个 POSIX 兼容的文件系统，这意味着您不能像和基于 Linux 或 UNIX 的文件系统交互那样与 HDFS 交互。

为了与 HDFS 中的文件进行交互，您需要使用 /bin/hdfs dfs <args> 文件系统 shell 命令接口，其中 args 代表您想在文件系统中的文件上所使用的命令参数。

以下是 HDFS shell 命令的一些参数示例：

cat 将文件复制到标准输出 (stdout)。

chmod 更改某个给定文件或文件集的读写权限。

chown 更改某个给定文件或文件集的所有者。

copyFromLocal 将文件从本地文件系统复制到 HDFS。

copyToLocal 将文件从 HDFS复制到本地文件系统。

cp 将 HDFS 文件从一个目录复制到另一个目录。

expunge 清空回收站中的所有文件。删除一个 HDFS 文件时，数据实际上没有被删除（想像您的 MAC 或 Windows 家用计算机，您就会明白这一点）。已删除的 HDFS 文件可以在回收站中找到，在稍后某个时间点它们会被自动清除。如果您希望立即清空回收站，可以使用 expunge 参数。

ls 显示在给定目录中的文件列表。

mkdir 在 HDFS 中创建目录。

mv 将文件从一个目录移到另一个目录。

rm 删除文件，并将它发送到回收站。如果您想跳过回收站进程并立即从 HDFS 中删除该文件，可使用 rm 命令的 –skiptrash 选项。

Hadoop 中的应用程序开发

您可能已从上一节推断出，Hadoop 平台对于操作非常大型的数据集而言可以说是一个强大的工具。然而，核心的 Hadoop MapReduce API 主要从 Java 调用，这需要熟练的程序员来完成。此外，对于程序员可能更复杂的是，

为需要执行长期和流水线处理的业务应用程序开发和维护 MapReduce 应用程序。

如果您从事编程的时间足够长，就会发现历史总是在重演。例如，我们经常说 XML 是“IMS 的复仇”，原因是其层次性和检索系统。在计算机语言发展领域中，就像汇编让路给结构化编程语言，然后又到 3GL 和 4GL 语言的发展，在 Hadoop 应用程序开发语言的世界中也一样。为了抽象 Hadoop 编程模型的一些复杂性，已经出现了几个在 Hadoop 之上运行的应用程序开发语言。在本节中，我们将介绍三个比较流行的语言，诚然，这听起来就像我们在动物园：Pig、Hive 和 Jaql（顺便说一下，我们在本章还会介绍 ZooKeeper）。

Pig 和 PigLatin

Pig 最初由 Yahoo! 开发，让使用 Hadoop 的人可以更专注于分析大型数据集，并花更少的时间编写 mapper 和 reducer 程序。就像什么都吃的真猪一样，Pig 编程语言旨在处理任何类型的数据——所以用这个名字！Pig 由两个组件组成：首先是被称为 PigLatin 的语言本身（是，命名各个 Hadoop 项目的人的确倾向于保持命名约定中的幽默感），第二个是 PigLatin 程序在其上执行的运行时环境。就像 Java 虚拟机 (JVM) 和 Java 应用程序之间的关系。在本节中，我们将整个实体称为 *Pig*。

让我们首先看看编程语言本身，您就可以看到它明显比必须编写 mapper 和 reducer 程序要容易多少倍。在 Pig 程序中的第一步是从 HDFS LOAD（加载）您要操作的数据。然后通过一组转换（在表面下转换成一组 mapper 和 reducer 任务）来运行数据。最后，您将数据 DUMP（转储）到屏幕，或者将结果 STORE（存储）在某处的一个文件中。

LOAD

与所有 Hadoop 特性一样，正在由 Hadoop 处理的对象都被存储在 HDFS 中。为了让 Pig 程序可以访问该数据，该程序必须先告诉 Pig，它会使用哪个或哪些文件，这可以

通过 LOAD 'data_file' 命令完成（其中 'data_file' 指定一个 HDFS 文件或目录）。如果指定了一个目录，在该目录中的所有文件都将被加载到程序中。如果数据存储在一个 Pig 无法在本机访问的文件格式中，您可以选择将 USING 函数添加 LOAD 语句中，以指定一个可以读入和解析数据的用户自定义函数。

TRANSFORM

转换逻辑是所有数据操作发生的地方。在这里，您可以 FILTER（筛选）掉不感兴趣的行，JOIN（联接）两组数据文件，GROUP（组合）数据以建立聚合，ORDER（排序）结果等。下面是一个 Pig 程序示例，它取出一个由 Twitter 源组成的文件，只选择那些使用 (English) iso_language 代码的微博，然后将它们按正在发 tweet 的用户进行组合，并显示该用户的 tweet 被转发的总次数。

```
L  = LOAD 'hdfs//node/tweet_data';
FL = FILTER L BY iso_language_code              EQ 'en'; G
     = GROUP FL BY from_user;
RT = FOREACH G GENERATE group, SUM(retweets);
```

DUMP 和 STORE

如果您没有指定 DUMP 或 STORE 命令，Pig 程序的结果不会生成。在您调试 Pig 程序时，通常会使用 DUMP 命令，它将输出发送到屏幕上。进入生产环境时，您只需将 DUMP 调用改为 STORE 调用，以便将运行您的程序所生成的任何结果存储到一个文件中，以作进一步处理或分析。注意，您可以在程序中的任意位置使用 DUMP 命令，以便将中间结果集转储到屏幕，这对于调试来说非常有用。

现在，我们已经有了一个 Pig 程序，需要让它运行在 Hadoop 环境中。这里就是 Pig 运行时的用武之地。运行 Pig 程序的方式有三种：嵌入在脚本中、嵌入在 Java 程序中，或者称为 Grunt（这当然是猪发出的声音——我们告诉过您，Hadoop 社区有轻松的一面）的 Pig 命令行。

无论您用这三种方法中的哪种方法来运行程序，Pig 运行时环境都会将程序转换成一组 map 和 reduce 任务，并在代表您的封面下运行它们。这大大简化了

与分析大量数据有关的工作，并让开发人员可以专注于分析数据，而不是每个 map 和 reduce 任务。

Hive

虽然 Pig 语言使用起来相当强大和简单，但它的缺点是，它需要您学习和掌握的新东西。Facebook 的一些人开发了一个运行时 Hadoop 支持结构，使得已经精通 SQL（这对于关系型数据库开发人员来说是司空见惯的）的任何人从一开始就能够利用 Hadoop 平台。他们的创作（被称为 *Hive*）使 SQL 开发人员可以编写与标准 SQL 语句类似的 Hive Query Language (HQL) 语句；现在您应该知道，HQL 仅限于它能理解的命令，但它仍然相当有用。HQL 语句被 Hive 服务分解为 MapReduce 作业并在整个 Hadoop 集群中执行。

对于有 SQL 关系型数据库背景的任何人来说，都会觉得本节看起来很熟悉。与任何数据库管理系统 (DBMS) 一样，您可以通过多种方式运行 Hive 查询。您可以在命令行界面（被称为 *Hive shell*）运行它们，从利用 Hive JDBC/ODBC 驱动程序的 Java Database Connectivity (JDBC) 或 Open Database Connectivity (ODBC) 应用程序运行它们，也可以从被称为 *Hive Thrift Client* 的客户端中运行它们。Hive Thrift Client 与任何数据库客户端都非常相似，它被安装在用户的客户端计算机上（或在三层架构的中间层）：它与在服务器上运行的 Hive 服务通信。您可以在使用 C++、Java、PHP、Python 或 Ruby 编写的应用程序中使用 Hive Thrift Client（就像您可以在嵌入式 SQL 中使用这些客户端语言来访问 DB2 或 Informix 等数据库一样）。下面显示了一个使用 Hive 的示例，其中创建了一个表，填充它，然后查询该表：

```
CREATE TABLE Tweets(from_user STRING, userid BIGINT, tweettext STRING, retweets INT)
  COMMENT 'This is the Twitter feed table' STORED AS
  SEQUENCEFILE;
LOAD DATA INPATH 'hdfs://node/tweetdata' INTO TABLE TWEETS;
SELECT from_user, SUM(retweets)
  FROM
  TWEETS
  GROUP BY from_user;
```

如您所见，Hive 看起来非常像 SQL 访问的传统数据库代码。然而，因为 Hive 以 Hadoop 和 MapReduce 操作为基础，所以存在若干关键差异。首先，Hadoop 适用于长时间的顺序扫描，因为 Hive 是基于 Hadoop 的，您可以预期查询有非常高的延迟（许多分钟）。这意味着 Hive 不适合于需要非常快的响应速度（像您对 DB2 等数据库所预期的响应速度）的应用程序。最后，Hive 是基于读取的，因此不适用于通常涉及高较多写操作的事务处理。

Jaql

Jaql 主要是一个面向 JavaScript Object Notation (JSON) 的查询语言，但它所支持的远远不只是 JSON。它让您可以处理结构化和非传统的数据，它是 IBM 捐献给开源社区的（只是 IBM 对开源所做的许多贡献之一）。具体来说，Jaql 使您可以选择、联接、组合并筛选存储在 HDFS 中的数据，就像 Pig 和 Hive 的混合体。Jaql 查询语言的灵感来自于许多编程和查询语言，包括 Lisp、SQL、XQuery 和 Pig。Jaql 是一个功能性、声明性查询语言，旨在处理大型数据集。为了实现并行性，Jaql 在适当的时候将高层次的查询重写为由 MapReduce 作业组成的“低层次”查询。

在我们介绍 Jaql 语言之前，先看看流行的数据交换格式 JSON，这样我们就可以在它上面构建我们的 Jaql 示例。应用程序开发人员正大量转向，以 JSON 作为其首选的数据交换格式，因为人类可以很轻松地阅读它，并且由于它的结构，它很容易被应用程序解析或生成。

JSON 构建于两种结构之上。第一种是名称/值对的集合（正如您之前在“MapReduce 的基础知识”一节中所了解的，这使它成为 Hadoop 中数据操作的理想选择，这些数据操作处理键/值对）。这些名称/值对可以代表任何东西，因为它们只是文本串（并因此非常适合现有模型），这些文本串可能代表数据库中的一个记录、一个对象、一个关联数组等。第二种 JSON 结构是创建一个值的有序列表的能力，就像在您现有应用程序中可能存在的数组、列表或序列。

JSON 中的对象表示为 { string : value }，而一个数组可以只用 [value, value, …] 表示，其中的 value 可以是字符串、数值、另一个 JSON 对象或另一个 JSON 数组。下面显示了一个示例，这是一个 Twitter 源的 JSON 表示（我们已经删除了在 tweet 语法中可以找到的许多字段，以提高可读性）：

```
results: [
{
created_at: "Thurs, 14 Jul 2011 09:47:45 +0000" from_user:
"eatonchris"
geo: {
                coordinates: [
                        43.866667
                        78.933333
                        ]
                        type: "Point"
    }
iso_language_code: "en"
text: " Reliance Life Insurance migrates from #Oracle to #DB2 and cuts
costs in half. Read what they say about their migration
http://bit.ly/pP7vaT"
retweet: 3 to_user_id: null
to_user_id_str: null
}
```

Jaql 和 JSON 都是面向记录的模型，并因此可以完美地彼此配合。注意，JSON 不是 Jaql 所支持的唯一格式——事实上，Jaql 极为灵活，并且可以支持许多半结构化的数据源，如 XML、CSV、平面文件等。然而，考虑到我们拥有的空间，我们将在以下 Jaql 查询中使用上面的 JSON 示例。正如您在本节所见，Jaql 看起来与 Pig 非常相似，但也与 SQL 有些相近。

Jaql 运算符

Jaql 构建于一组核心运算符之上。让我们看看 Jaql 中一些最常用的运算符，了解它们如何工作，然后通过一些简单的示例，演示如何查询前面所表示的 Twitter 源。

FILTER FILTER 运算符将一个数组作为输入，并根据指定的谓词筛选出感兴趣的元素。对于熟悉 SQL 的人来说，可以将 FILTER 运算符想像为一个 WHERE 子句。例如，如果您

希望只查看 Twitter 源中由用户 eatonchris 所创建的输入记录，可将与下面类似的代码放到您的查询中：

```
filter $.from_user == "eatonchris"
```

如果只想看看被重新转发了两次以上的 tweet，可包括如下面所示的查询：

```
filter $.retweet > 2
```

TRANSFORM TRANSFORM 运算符以一个数组作为输入，并输出另一个数组，其中，第一个数组的元素已通过某种方式被转换。对于 SQL 痴迷者，您会发现这类似于 SELECT 子句。例如，如果输入数组有两个数字，记为 N1 和 N2，TRANSFORM 运算符使用下面代码产生这两个数字的总和：

```
transform { sum: $.N1 + $.N2 }
```

GROUP GROUP 运算符的工作非常像 SQL 中的 GROUP BY 子句，在这里一组数据被聚合输出。例如，如果想计算本节中示例内的 tweet 总数，可以使用这个命令：

```
group into count($)
```

同样，如果想确定每个用户所有转发的 tweet 总和，您会使用一个像这样的 Jaql 查询：

```
group by u = $.from_user into { total: sum($.retweet) };
```

JOIN JOIN 运算符需要两个输入数组，并根据在 WHERE 子句中指定的联接条件产生一个输出数组，类似 SQL 中的联接操作。让我们假设，您有一个 tweet 数组（如 JSON tweet 示例），并且还有来自一组您在 Tweeter 上关注的人有趣数据。这种数组可能看起来像这样：

```
following = { from_user: "eatonchris" },
                        { from_user: "paulzikopoulos" }
```

在该示例中，您可能使用 JOIN 运算符来联接 Twitter *feed* 数据与 Twitter *following* 数据，使产生的结果只来自您关注的人的 tweet，如：

```
join feed, follow
where feed.from_user = following.from_user into {feed.*}
```

EXPAND EXPAND 运算符以一个嵌套数组为输入，并产生单个数组作为输出。让我们假设您有一个地理位置的嵌套数组（以经度和纬度坐标表示），如下所示：

```
geolocations = [[93.456, 123.222],[21.324, 90.456]]
```

在该例中，geolocations -> expand; 命令将在单个数组中返回结果，如下所示：

```
[93.456, 123.222, 21.324, 90.456]
```

SORT 正如您所预期的，SORT 运算符以一个数组作为输入，并产生一个数组作为输出，其中的元素已按顺序排序。默认的 Jaql 排序顺序是升序。您可以使用 sort by desc 关键字按降序来排序 Jaql 结果。

TOP TOP 运算符返回输入数组的前 n 个元素，其中 n 是 TOP 关键字后面的一个 <integer>。

内置的 Jaql 函数

除了核心运算符之外，Jaql 还有许多内置的函数，使您可以读入、操作和写出数据，以及调用外部函数，如 HDFS 调用等。您可以添加自己的自定义函数，它们反过来也可以调用其他函数。要在本书中介绍 100 多个内置函数，这显然是太多了；但是，在基础 Jaql 文档中对它们有详细的记录。

Jaql 查询

MapReduce 作业是一个数据流，与此类似的是，Jaql 也可以被认为是一个管道，数据从源流入，通过一组不同的运算符，然后流出到接收器（目的地）。用于表示流从一个操作数

到另一个操作数的符号是一个箭头： ->。Jaql 与 SQL 不同，SQL 的输出排在最前面（例如，SELECT list），在 Jaql 中，运算按自然顺序排列，先指定源，随后是您想用于操作数据的各种运算符，最后是接收器。

让我们总结 Jaql 这一节，并把它汇总成一个简单 Jaql 示例，按用户计算以英文书写的 tweet 的数量：

```
$tweets = read(hdfs("tweet_log"));
$tweets
-> filter $.iso_language_code = "en"
-> group by u = $.from_user
        into { user: $.from_user, total: sum($.retweet)
};
```

第一行只是打开其中包含数据的文件（目的是读取它），该文件驻留在 HDFS 中，然后赋予该文件一个名称，在本例中是 $tweets。接下来，Jaql 查询读取 $tweets 并将数据传递给 FILTER 运算符。筛选器仅传递 iso_language_code = en 的 tweet。这些记录随后被传递到 GROUP BY 运算符，将每个用户转发 tweet 的值相加在一起，得到每个给定用户的一个总和。

Jaql 引擎在内部将查询转换成 map 和 reduce 任务，可以大幅缩短与在 Hadoop 中分析大量数据相关的应用程序开发时间。注意，我们在本章只说明了 Jaql 和 JSON 之间的关系；重要的是要认识到，这并不是 Jaql 所支持的唯一数据格式。事实恰恰相反：Jaql 是一个灵活的基础架构，可以管理和分析多种半结构化数据，如 XML、CSV 数据、平面文件、关系数据等。此外，从开发的角度而言，请不要忘记 Jaql 基础架构的灵活性和可扩展性非常高，并支持在查询接口和您的首选应用程序语言（例如，Java、JavaScript、Python、Perl、Ruby 等）之间传递数据。

Hadoop Streaming

除了 Java，您还能够以其他语言编写 map 和 reduce 函数，并使用称为 Hadoop Streaming（简写为 Streaming）的API 调用它们。Streaming 以 UNIX 流式传输的概念为基础，其中，

输入从标准输入读取，而输出则被写到标准输出。这些数据流代表 Hadoop 和您的应用程序之间的接口。

Streaming 接口最适合通常会使用 Python 或 Ruby 等脚本语言开发的简短应用程序。主要的原因是数据流具有基于文本的性质，每一行文本代表一条记录。

以下示例显示了使用 Streaming 执行 map 和 reduce 函数（以 Python 编写）：

例如：

```
hadoop  jar contrib/streaming/hadoop-streaming.jar \
      -input input/dataset.txt \
      -output output \
      -mapper text_processor_map.py \
      -reducer text_processor_reduce.py
```

使数据进入 Hadoop

HDFS 的挑战之一是，它不是一个 POSIX 兼容的文件系统。这意味着，您所习惯的与典型文件系统交互的所有事情（复制、创建、移动、删除或访问一个文件等）不会自动适用于 HDFS。要对 HDFS 中的文件执行任何操作，您必须直接使用 HDFS 的接口或 API。这又是使用 GPFS-SNC 文件系统的另一个优势；利用 GPFS-SNC，您就能够使用与任何其他文件系统交互的相同方式来与大数据文件进行交互，因此，在 GPFS-SNC 上运行的 Hadoop 文件处理任务被大大减少。在本节中，我们讨论如何使您的数据进入 HDFS 的基础知识并介绍 *Flume*，这是一个能够将数据导入 Hadoop 集群的分布式数据收集服务。

基本的复制数据

您可能还记得在本章前面的“Hadoop 常用组件”一节中，您必须通过 API 或使用命令 shell，使用特定的命令将文件移动到 HDFS 中。将文件从本地文件系统移动到 HDFS 中，最常见的方式是通过 copyFromLocal

命令。如需将文件从 HDFS 移到本地文件系统，一般会使用 copyToLocal 命令。这两个命令的示例如下：

```
hdfs dfs –copyFromLocal /user/dir/file hdfs://s1.n1.com/dir/hdfsfile
hdfs dfs –copyToLocal hdfs://s1.n1.com/dir/hdfsfile /user/dir/file
```

这些命令都通过 HDFS shell 程序运行，它只是一个 Java 应用程序。该 shell 使用 Java API 将数据移进和移出 HDFS。您可以从任何 Java 应用程序中调用这些 API。

注 *也可以通过由 **hadoop fs** 命令调用的 **Hadoop shell** 发出 **HDFS** 命令。*

用这种方法的问题是，您必须让 Java 应用程序开发者编写逻辑和程序来从 HDFS 读写数据。也可以使用其他方法（如 C++ API，或通过面向跨语言服务的 Thrift 框架），但这些仅仅是基础 Java API 的包装器。如果需要从您的 Java 应用程序访问 HDFS 文件，会使用 org.apache.hadoop.fs 包中的方法。这让您能够在 MapReduce 应用程序中直接包含从 HDFS 读出和写入 HDFS 的操作。不过请注意，HDFS 为顺序读写而设计。这意味着，将数据写到 HDFS 文件时，只可以写到文件的末尾（在数据库世界中它被称为 APPEND）。这里又是使用 GPFS-SNC 作为 Hadoop 集群文件系统骨干的一个优势，因为这个专用的文件系统具有固有的能力，可以在一个文件中而不只是在文件的末尾执行查找和写入。

Flume

水槽 (flume) 是一个通道，能够引导水从源头流到需要水的某个其他位置。正如其巧妙的名字所暗示的，***Flume*** 的创建（在本书出版时，它是 Apache 孵化器项目）是为了让您能够将数据从某个源流入您的 Hadoop 环境。在 Flume 中，您使用的实体被称为 ***source***（*源*）、***decorator***（*装饰器*）和 ***sink***（*接收器*）。***source*** 可以是任意数据源，Flume 有许多预定义的源适配器，我们将在本节讨论它们。***sink*** 是特定操作的目标（在 Flume 以及在使用该术语的其他模式中，一个操作的接收器

可能是下一个下游操作的源）。*decorator* 是在流上的一个操作，能够以某种方式转换流，它可以压缩或解压缩数据，通过添加或删除信息块修改数据等。

Flume 中内置了一些预定义的源适配器。例如，有些适配器允许来自 TCP 端口的任何内容进入流，或来自标准输入 (stdin) 的任何内容进入流。一些文本文件源适配器为您提供了细粒度控制，可以抓取某个特定的文件，并将它馈送进数据流，甚至抓住文件的尾部，并不断将写入到该文件的任何新数据馈送给流。后者对于将诊断或 Web 日志馈送入数据流非常有用，因为它们正在被不断追加，TAIL 运算符将持续从文件中获取最新的条目，并将它们放入流。其他的预定义源适配器，以及作为一个命令的 exit，使您可以使用任何可执行命令来馈送数据流。

在 Flume 中有三种类型的 sink（接收器）。一种接收器基本上是流的最终目的地，被称为 *Collector Tier Event（收集器层事件）*接收器。这是您将一个流（或可能是联接在一起的多个流）放进 HDFS 格式文件系统的位置。在 Flume 中使用的另一种接收器被称为 *Agent Tier Event（代理层事件）*；希望接收器成为另一个操作的输入源时，可使用这种接收器。使用这些接收器时，Flume 也将发回数据已实际到达接收器的确认，从而确保流的完整性。最后一种接收器被称为 *Basic（基本）*接收器，它可能是一个文本文件、控制台显示、简单的 HDFS 路径，或者一个数据被删除的空 bucket。

要想通过多个源流入数据，操作数据，然后将数据放进 Hadoop 环境，请考虑 Flume（它为日志数据而设计，但它也可用于其他类型的数据）。当然，要执行非常复杂的数据转换和清理，应该寻找一个企业级的数据质量工具集，如 IBM Information Server，它为数据转换、提取、发现、质量、整治等提供服务。IBM Information Server 可以在处理 Hadoop 集群的数据之前就处理大规模数据操作，并提供技术（例如能够看到数据沿袭）之间的集成点（即将提供更多）。

其他 Hadoop 组件

其他许多开源项目都属于 Hadoop 的范围，作为 Hadoop 子项目，或作为顶级 Apache 项目，随着时间的推移还会有更多新组件（正如您可能已经猜到了，他们的名字也是同样有趣的：ZooKeeper、HBase、Oozie、Lucene 等）。在本节中，我们再介绍 4 个您可能会遇到的与 Hadoop 相关的项目（它们都会被作为任何 InfoSphere BigInsights 版本的一部分提供）。

ZooKeeper

ZooKeeper 是一个开源 Apache 项目，提供一个集中式基础架构和服务，在整个集群中实现同步。ZooKeeper 维护在大型集群环境中所需的公共对象。这些对象的示例包括配置信息、分层名称空间等。应用程序可以利用这些服务实现跨大型集群协调分布式处理。

想像一个覆盖 500 个或更多商品服务器的 Hadoop 集群。如果您曾经管理过只有 10 个服务器的数据库集群，就知道在名称服务、组服务、同步服务、配置管理等方面都需要在整个集群进行集中式管理。此外，利用 Hadoop 集群的许多其他开源项目都需要这些类型的跨集群服务，在 ZooKeeper 提供它们，这意味着这些项目中的每一个都可以嵌入 ZooKeeper，而不必在每个项目中从头开始构建同步服务。现在我们可以通过 Java 或 C 接口与 ZooKeeper 发生交互（我们的猜测是，将来开源社区将增加与 ZooKeeper 交互的其他开发语言）。

ZooKeeper 提供了一个实现跨节点同步的基础架构，应用程序可以使用该基础架构确保整个集群的任务是序列化或同步的。它通过维护 ZooKeeper 服务器内存中的状态类型信息来实现这一点。ZooKeeper 服务器是一台计算机，保存整个系统的状态副本，并将该信息存放在本地日志文件中。非常大型的 Hadoop 集群可以由多个 ZooKeeper 服务器提供支持（在这种情况下，主服务器同步顶级服务器）。每台客户端计算机与其中一个 ZooKeeper 服务器通信，以获得并更新其同步信息。

在 ZooKeeper 内，应用程序可以创建 znode（一个文件，存放在 ZooKeeper 服务器上的内存中）。集群中的任何节点都可以更新 znode，集群中的任何节点可以注册，获得对该 znode 的变更通知（在 ZooKeeper 的说法中，服务器可以被设置为“观察”某个特定的 znode）。使用这个 znode 基础架构（还有更多内容，我们甚至不能在本节中正确地开始做这件事），应用程序可以通过在 ZooKeeper znode 中更新自己的状态，从而在整个分布式集群同步它们的任务，然后 znode 将特定节点的状态变化通知给集群的其余节点。这个集群范围的状态集中化服务，对于跨一组大型分布式服务器的管理和序列化任务而言是必需的。

HBase

HBase 是一个面向列的数据库管理系统，运行在 HDFS 之上。它非常适合于在许多大数据用例中很常见的稀疏数据集。与关系型数据库系统不同，HBase 不支持 SQL 等结构化查询语言；事实上，HBase 完全不是一个关系型数据存储。HBase 应用程序以 Java 编写，这与典型的 MapReduce 应用程序相似。HBase 不支持以 Avro、REST 和 Thrift 编写的应用程序。（我们在本章结尾将简要介绍 Avro，其他两个在本书中没有介绍，但您可以通过简单的 Google 搜索轻松地找到有关它们的详细信息）。

HBase 系统包括一组表。每个表包含多个行与列，与传统数据库很相似。每个表必须有一个元素被定义为主键，所有对 HBase 表的访问尝试都必须使用这个主键。一个 HBase 列表示一个对象的一个属性；例如，如果该表存储的是环境中服务器的诊断日志，每一行可能是一条日志记录，在这样的一个表中，典型的列将是日志记录被写入的 timestamp（时间戳），或者可能是记录来源的 servername。事实上，HBase 支持将许多属性组合在一起，被称为 *列族 (column family)*，所以一个列族的元素都被存储在一起。这与面向行的关系型数据库不同，给定行的所有列都被存储在一起。利用 HBase，您必须预定义表架构，并指定列族。但是这非常灵活，因为

新列可以随时被添加到列族中，使架构具有灵活性，并因此能够适应不断变化的程序需求。

正如 HDFS 有一个 NameNode 和从节点，MapReduce 有 JobTracker 和 TaskTracker 从属，HBase 也构建于类似的概念之上。在 HBase 中，*master node（主节点）*管理集群，而 *region server（区域服务器）*存储表的部分并执行数据上的工作。以同样的方式，HDFS 因 NameNode 的可用性存在一些企业问题（在其他领域，可以由 BigInsights 针对真实的企业部署"强化"），HBase 对失去其主节点也很敏感。

Oozie

正如您可能在我们对 MapReduce 功能的讨论中已经注意到，许多作业可能需要被链接在一起，以满足复杂应用程序的要求。*Oozie* 是一个开源项目，可以简化工作流和作业之间的协调。它使用户能够定义操作和各操作之间的依赖关系。然后，Oozie 将调度执行工作，在所需的依赖关系已经达到时执行操作。

在 Oozie 中的工作流被定义为 *Directed Acyclical Graph（定向非周期性图形，DAG）*。*非周期性 (Acyclical)* 意味着图形中没有循环（换言之，图形有一个起点和终点），所有任务和依赖关系都从起点指向终点，不会回去。一个 DAG 由*操作节点 (action nodes)* 和*依赖关系节点 (dependency nodes）*组成。操作节点可以是一个 MapReduce 作业、Pig 应用程序、文件系统任务或 Java 应用程序。图形中的流控制由节点元素表示，根据图形中前面任务的输入来提供逻辑。流控制节点的示例有决策、分叉和联接节点。

可以根据某个给定时间或基于文件系统中某些特定数据的到来而调度工作流的开始。开始后，根据在图形中前面操作的完成情况执行更多的工作流操作。图 4-3 是一个 Oozie 工作流示例，其中的节点代表操作和控制流操作。

Lucene

Lucene 是一个非常受欢迎的面向文本搜索的开源 Apache 项目，它被包括在许多开源项目中。 Lucene 早于 Hadoop，

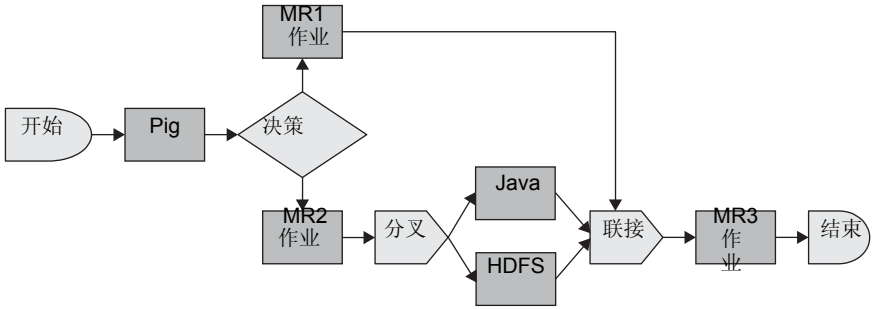

图 4-3 *一个 Oozie 工作流包括多个决策点作为端到端执行的一部分*

自 2005 年起就已经是顶级 Apache 项目。Lucene 提供可以在您的 Java 应用程序中使用的全文本索引和搜索库（注意，Lucene 已经被移植到 C++、Python、Perl 等）。如果您在 Internet 上搜索过，很可能已经与 Lucene 进行过交互（虽然您可能不知道它）。

Lucene 的概念相当简单，但这些搜索库的使用功能却非常强大。简言之，假设您需要在文本集合或一组文档内进行搜索。Lucene 将这些文档分解成文本字段，并对这些字段建立一个索引。该索引是 Lucene 的重要组成部分，因为它形成了快速全文检索功能的基础。然后，使用 Lucene 库内的搜索方法找到文本组件。这个索引和搜索平台随 BigInsights 提供，并且被集成到 Jaql 中，提供在 Jaql 内构建、扫描和查询 Lucene 索引的功能。

BigInsights 提供一个非常强大的文本提取库，可收集非结构化文本的结构，从而添加更强大的功能，它原生运行在 BigInsights 上并利用 MapReduce。甚至还有一个开发框架，可以利用一个完整的工具环境扩展和自定义库，这让它的使用相对容易。通过将这些文本提取器添加到文本索引功能，BigInsights 为 Hadoop 提供当今市场上功能最丰富和最强大的文本分析平台之一。更重要的是，您不能在 HDFS 中存储 Lucene 索引；但是，您可以将它与其他 Hadoop 数据一起存储在 GPFS-SNC 中。

Avro

Avro 是一个 Apache 项目，提供数据序列化服务。将 Avro 数据写入一个文件时，定义该数据的架构总是被写进该文件。这让任何应用程序以后都可以轻松地读取数据，因为文件中存储了定义数据的架构。对于 Avro 流程还有另一个好处：可以对数据进行版本管理，在应用程序中的架构变化可以很容易处理，因为旧数据的架构仍然存储在数据文件内。使用 JSON 可以定义 Avro 架构，我们在前面的 Jaql 一节中已简单讨论过 JSON。

架构定义在一个文件中所包含的数据类型，并且在使用 Avro API 将数据写入文件时会被验证。类似地，在数据被读回文件时，可以根据架构定义对数据进行格式化。该架构使您可以定义两种类型的数据。第一种是*基本数据类型*，如 STRING、INT[eger]、LONG、FLOAT、DOUBLE、BYTE、NULL 和 BOOLEAN。第二种是*复杂类型定义*。复杂类型可以是一个 record、array、enum（为类型定义一个可能值的枚举列表）、map、union（定义一种类型为几种类型之一）或 fixed 类型。

在 C、C++、C#、Java、Python、Ruby 和 PHP 中都提供了 Avro 的 API，使它在与 Hadoop 有关的大部分常见应用程序开发环境中都可用。

小结

如您所见，Hadoop 并不仅仅是单个项目，它更像是一个项目生态系统，这些项目都旨在简化、管理、协调和分析大型数据集。IBM InfoSphere BigInsights 完全接受该生态系统，并且提供代码提交、贡献、无分叉向后兼容性承诺。在下一章中，我们将特别介绍 IBM 所进行的工作，将 Hadoop 及其相关技术扩展至一个通过 IBM 带给该合作伙伴关系的企业级体验所丰富的分析平台。

5

InfoSphere BigInsights：分析静止的大数据

Hadoop 在协助企业驾驭迄今难以管理和分析的数据方面提供了巨大的潜力。具体来说，Hadoop 能够利用各种结构（或根本不使用结构）处理海量数据。尽管如此，Hadoop 从各方面讲仍是一项相当年轻的技术。Apache Hadoop 顶级项目自 2006 年开始启动，虽然采用率在不断上升，并且越来越多的人参与开放源码编写，但 Hadoop 仍然存在不少人所共知的缺点（平心而论，即便是版本 1.0 情况也差不多）。从企业的角度而言，这些缺点可能会妨碍各家公司在生产环境中使用 Hadoop，甚至可能会使它们拒绝采用 Hadoop，因为客户往往会预期在生产过程中实现某些运营指标，如性能、管理功能和稳健性。例如，正如我们在第 4 章所述，Hadoop 分布式文件系统 (HDFS) 具有一个集中元数据存储（以下简称 NameNode），它表示一个会导致失去可用性的单点故障 (SPOF)（版本 0.21 中增加了冷备用）。当 NameNode 恢复之后，可能还需要花费很长时间恢复 Hadoop 集群的正常运行，因为其跟踪的元数据必须加载至 NameNode 的内存结构，而所有内存结构均须重新构建和填充。此外，Hadoop 结构复杂，难以安装、配置和管理，并且目前掌握 Hadoop 技

术的人员还不是很多。同样，掌握 MapReduce 技术的开发人员资源也相当有限。编写在 Hadoop 环境下运行的传统分析算法（如统计或文本分析）难度很大，要求分析师精通 Java 编程，同时还能够熟练运用 MapReduce 技术开展分析算法（Pig 和 Jaql 等高级语言简化了 MapReduce 编程过程，但仍需要经过学习）。内容还有很多，但您只需记住一点：Hadoop 不仅需要一些企业强化，还需要加强工具和功能，使其能够帮助实现 Hadoop 平台提供的各种发展潜力（例如可视化、文本分析及图形管理工具）。

IBM InfoSphere BigInsights (BigInsights) 解决了所有这些问题，更重要的是，还能使 IBM 专心关注以下两个主要产品目标：

- 提供专为企业使用而进行强化的 Hadoop 平台，同时深入考虑高可用性、可扩展性、性能、易用性及公司将要部署的任意解决方案中所体现的其他优势。
- 通过为开发人员提供开发和运行时环境，来构建高级分析应用程序及为企业用户提供工具分析大数据，从而使与大数据分析相关的时间价值曲线变平。

在本章中，我们将会就 IBM 如何筹备 Hadoop 供企业使用进行探讨。诚如人们所知，IBM 一直以来都非常了解企业需求。通过采用 Hadoop（及其开源生态系统）等新型技术，并凭借 IBM 创立至今近百年来累积的深厚经验和智力资本不断扩展，您将会获得成功的产品组合，从而借助可信的平台探索 Hadoop 并且迅速获取收效。

易用性：安装流程简单

BigInsights 安装程序以简约为首要设计原则。IBM 开发团队曾经深刻思考，“IBM 如何削减掌握 Hadoop 所需的时间，同时还无需精通通常安装和运行开源软件所需的工作和技术？”他们的答案是采用 BigInsights 安装程序。

BigInsights 安装程序的主要目标是降低复杂性。这样，您将不必担心各种繁杂的软件预备知识，也不用费力确定需要下载哪种 Apache Hadoop 组件、这些软件的配置及 Hadoop 集群的整体格局。BigInsights 安装程序将代您完成一切，您要做的只是按下按钮。BigInsights 将为您消除几乎全部 Hadoop 开始时的复杂操作。操作相当简便，整个过程与安装商业软件非常类似。

为编写本书，我们共创建了三个不同的 Hadoop 集群：

- 一个集群利用开源软件从零开始创建，我们将之称为自主开发 (RYO) Hadoop 方法
- 一个来自竞争对手，只提供安装程序、某些操作工具及 Hadoop 支持合同
- 一个提供 BigInsights

“自主开发”Hadoop 方法引导我们直接进入 Apache 网站下载 Hadoop 项目，最终会涉及大量工作。具体来说，我们必须完成以下工作：

1. 选择要安装哪些 Hadoop 组件，以及安装这些组件的哪些版本。我们找到了许多组件，并且很难立即明确指出自己需要哪种组件以便开始部署项目，因而需要开展一些初步研究。
2. 创建和设置 Hadoop 用户帐户。
3. 下载我们确定自身所需的各种 Hadoop 组件，并在计算机集群上进行安装。
4. 为 Hadoop 用户帐户配置安全 Shell (SSH)，然后将密钥复制到集群内的各台计算机。
5. 配置 Hadoop 定义我们所期待的运行方式；例如，我们制定了 I/O 设置、JobTracker 及 TaskTracker 级详细信息。
6. 配置 HDFS——特别是要设置及格式化 NameNode 和辅助 NameNode。
7. 定义所有全局变量（例如，HADOOP_CLASSPATH、HADOOP_PID_DIR、HADOOP_HEAPSIZE 和 JAVA_HOME）。

您能够想像得到，从开源组件安装并运行 Hadoop 集群确实相当复杂并且有些费力。在经过一番努力后，我们成功突破了障碍。随后，我们组建了一支经验丰富的 Hadoop 开发团队随时准备答疑解惑。如果您即将踏上 RYO 征程，则需要对整个 Hadoop 生态系统以及基本 Hadoop 管理和配置技术具有不错的了解。您需要首先具备上述知识才能开始考虑运行简单的 MapReduce 作业，更不用说运行任何一种有意义的分析应用程序。

接下来，我们尝试安装竞争对手的 Hadoop 分发程序（注意，这一种分发程序，而不是 BigInsights 等平台）。这款竞争对手的安装程序确实展现出了基本服务开源方法方面的改进，因为它拥有极佳的图形安装程序。但是，它无法安装和配置 Pig、Hive 和 Flum 等需要手动安装的其他 Hadoop 生态系统组件。

这两种体验与 BigInsights 方法大相径庭，BigInsights 能够通过单一安装程序简单拟定并配置整套必要组件。采用 BigInsights 后，只需单击几下即可完成安装，因而不必担心任何 Hadoop 相关组件和版本问题。只需执行极少的配置操作，并且无需下载任何额外的预备内容。更重要的是，您可以使用 IBM 的安装程序以图形的方式构建响应文件，随后以自动化模式利用它在集群内的所有节点上部署 BigInsights。

BigInsights 1.2 中包含的 Hadoop 组件

BigInsights 的特色在于，以 Apache Hadoop 及其相关开源项目作为核心组件。IBM 将继续致力于保持开源项目的完整性，防止它们与核心分离或以其他方式偏离核心。下表列出了 BigInsights 1.2 中包含的开源项目（及其版本），也是编写本书时市场上提供的最新版本：

组件	版本
Hadoop（通用实用程序、HDFS 和 MapReduce 框架）	0.20.2
Jaql（编程和查询语言）	0.5.2
Pig（编程和查询语言）	0.7
Flume（数据收集和聚合）	0.9.1

组件	版本
Hive（数据汇总和查询）	0.5
Lucene（文本搜索）	3.1.0
ZooKeeper（进程协调）	3.2.2
Avro（数据序列化）	1.5.1
HBase（实时读取和写入数据库）	0.20.6
Oozie（工作流程和作业安排）	2.2.2

无论采用哪个版本的 BigInsights，更新开源组件和 IBM 组件均需经过一系列测试周期，以确保各组件协调合作。我们要澄清的另一项特别问题在于：您不能只是简单地将新代码丢入生产环境。实施开源项目期间总会出现一些向后兼容性问题。BigInsights 为您的 Hadoop 组件消除了所有风险和臆断麻烦。像其他 IBM 软件一样，它也经历了严格的回归和质量保证测试流程。所以，问问自己这样一个问题：您愿意成为自己的系统集成商，反复测试所有 Hadoop 组件来确保兼容性？还是愿意委托 IBM 寻找性能稳定的堆栈，以便您部署并确保工作环境充分可靠？

最后，BigInsights 安装程序还部署了额外的基础架构（包括分析工具和组件），确保企业 Hadoop 的稳定性和高品质，从而将 BigInsights 构建成为一个平台而非分发程序。我们将在本章的剩余部分对这些问题展开讨论。

Hadoop 就绪的企业质量文件系统：GPFS-SNC

General Parallel File System (GPFS) 由 IBM Research 于上世纪 90 年代开发，专门用于高性能计算 (HPC) 应用程序。自 1998 年首次发布以来，GPFS 已被全球许多运行速度最快的超级计算机广为采用，其中包括 Blue Gene、Watson（*Jeopardy!* 游戏超级计算机）和 ASC Purple（安装在 ASC Purple 超计算机上的 GPFS 系统支持的数据吞吐量高达 120 GB/秒，十分惊人！）除用于 HPC 外，GPFS 还广泛出现在全球数千种其他任务关键装置中。GPFS 也是 DB2 pureScale 的一个文件系统，甚至在许多 Oracle RAC 装置中也能找到

它的踪迹；您可能发现 GPFS 还为一些高度可扩展的 Web 和文件服务器、财务部门和工程部门内的其他数据库和应用程序等提供基础支持。毋庸置疑，GPFS 因其极高的可扩展性、高性能及可靠性赢得了良好的企业级声誉并实现扬名立户。

现在，妨碍某些企业广泛采用 Hadoop 的一个障碍在于 HDFS。这是一种相对较新的文件系统，目前还存在一些设计相关限制。HDFS 开发指导原则均根据用例定义，假设 Hadoop 工作负载包含对超大文件集的顺序读取操作（并且集群中不存在随机文件写入内容，只有追加写入内容）。相比之下，GPFS 已针对范围广泛的各种工作负载及多种用途进行量身设计，我们将在本节对此展开讨论。

扩展 Hadoop GPFS：GPFS 无共享集群

GPFS 最初只能作为存储区域网络 (SAN) 文件系统使用，不适用于 Hadoop 集群，因为这些集群均使用本地连接磁盘。SAN 技术并非 Hadoop 所需的最佳技术的原因在于，MapReduce 作业当其数据存储在负责处理自身作业的节点上时性能会更好（需要对数据进行局部认知）。在 SAN 中，数据位置是透明的，因而能够实现高网络带宽和磁盘 I/O，特别是在包含大量节点的集群中。

2009 年，IBM 开始扩展 GPFS 处理带有 GPFS-SNC（无共享集群）的 Hadoop。以下是 IBM 为 GPFS 新增的关键功能，使其成为适用于 Hadoop 的最佳文件系统，从而强化了企业 Hadoop：

- **局部性认知** Hadoop 的一个主要特点在于其致力于处理存储数据的节点位置数据。因而最大限度降低了网络流量并提高了性能。为支持这一功能，GPFS-SNC 将提供集群中存储的所有文件的位置信息，以便 Hadoop JobTracker 利用这些位置信息选择需要运行的本地任务副本，进而帮助提高性能。
- **元数据块** 典型 GPFS 数据块的大小为 256 KB，而在 Hadoop 集群中，数据块要比这大得多。例如，BigInsights 的建议数据块大小为 128 MB。在 GPFS-SNC 中，我们将大量 GPFS 数据块叠加在一起创造出了

*元数据块*这一概念。各 map 任务以元数据块为基准执行，而 Hadoop 外的文件操作仍使用较小的普通数据块大小，较小的大小对其他一些类型的应用程序更为有效。这种灵活性能够确保各种应用程序在同一集群上工作，同时还能保持最佳性能。HDFS 并不具备这些优点，其严格限制使用 Hadoop 进行存储，并且只能使用 Hadoop 存储。例如，您无法在 HDFS 上执行 Lucene 全文索引。不过在 GPFS-SNC 中，您可以将 Lucene 全文索引及其文本数据存储在集群中（这种归置具有性能优势）。虽然 Lucene 使用 256 KB 的 GPFS 数据块执行自己的操作，但所有 Hadoop 数据均存储在集群中并以元数据块的形式读取。

- **写入关联和可配置的复制** GPFS-SNC 允许您为文件定义位置策略，包括文件复制期间采用的方法。正常的复制策略是第一个副本为计算机本地副本，第二个副本为机架本地副本（这一点与 HDFS 不同），第三个副本则以条带形式分布在集群中的其他机架之间。例如，您可能会确定一组特定的文件始终存储在一起，从而使应用程序从同一位置访问数据。而在 HDFS 中却无法做到这一点，这种做法能够提高大型连续读取等特定工作负载的性能。第二个副本策略也能够保证这些数据存储在一起。如果主节点发生故障，即可轻松地切换至另一节点，不会造成任何应用程序性能降级。第三个数据副本通常以条带形式存储，在前两个副本中的任意一个必须重建时使用。文件条带化后，文件恢复操作会迅速得多。HDFS 中则无法执行数据条带化，同时您也无法自定义写入关联或复制行为（除非更改复制因子）。
- **可配置恢复策略** 磁盘发生故障后，系统将会复制数据块发生丢失的所有文件。GPFS-SNC 将自动复制集群中的缺失文件，以维持复制水平。GPFS-SNC 可让您自定义策略，决定磁盘发生故障时执行的操作。例如，一种方法是在发生故障时重新条带化磁盘。由于系统通常会条带化其中一个文件的副本，因而缺失数据块的重建非常迅速，并会并行执行读取。

或者，您也可以指定一项磁盘重建策略（例如，或许在更换磁盘时应用）。这些恢复策略不一定要自动完成；您有权决定在维护任务（如交换一组磁盘或节点）时使用手动恢复操作。还可以配置逐步恢复正常工作。例如，如果某个磁盘脱机，随后又恢复联机状态，GPFS-SNC 明白其只需复制缺失数据块，因为它在每个磁盘上都保存了文件列表。在 HDFS 中，NameNode 将会对复制不足的文件启动复制，但无法自定义恢复操作。

GPFS 的所有优势特点（促使其成为大型任务关键 IT 安装环境的首选文件系统）均适用于 GPFS-SNC。毕竟，它依旧是 GPFS，但可以进行 Hadoop 扩展。您可以在 GPFS-SNC 中实现同样的稳定性、灵活性和性能，并可使用从前的所有工具。GPFS-SNC 还可提供分层存储管理 (HSM) 功能，这样就能以不同的检索速度有效管理和使用磁盘驱动器。因而能够管理不同热度带的数据，保证热门数据位于性能最佳的硬件上。HDFS 不具备这项能力。

GPFS-SNC 这项创新技术在 2010 年赢得了 Supercom 主办的著名存储挑战赛大奖，成为本次大赛参赛解决方案中“最具创新性的存储解决方案”。

GPFS-SNC 集群外观如何？

GPFS-SNC 是一种无共享架构的分布式存储集群。其中不包含元数据中央存储区，因为数据已实现在集群中多个节点共享。此外，文件系统管理任务分布于集群中的各个数据节点，因此，如果发生故障，系统将会自动指定替换节点承担这些任务。

在图 5-1 中可以看到，GPFS-SNC 集群包含多个商用硬件机架，存储区附着于计算节点。如果您熟悉 HDFS，将会发现图 5-1 中不包含 NameNode、辅助 NameNode，或者充当元数据中央存储区的任何硬件。*这是 GPFS-SNC 优于 HDFS 的一个显著优势。*

图 5-1 中的集群设计是一个简单示例，假定每个计算节点均遵循相同的 CPU、RAM 和存储规范（实际上，仲裁[Quorum]节点与主集群配置服务器之间可能会存在硬

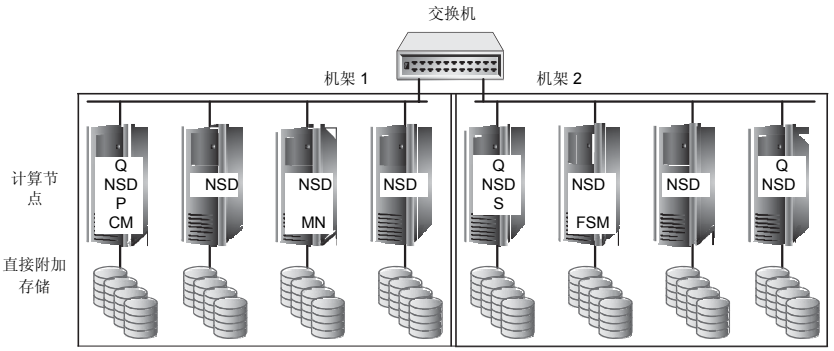

图 5-1 *GPFS-SNC 集群示例*

件差异以便互相强化，从而减少可能发生的中断）。为确保顺利管理集群，GPFS-SNC 集群中的不同计算节点将会承担不同的管理角色，这一点值得探讨。

该图显示 GPFS-SNC 集群中的每个计算节点均提供*网络共享磁盘 (NSD)* 服务器服务，可以访问本地磁盘。当 GPFS-SNC 集群中的某个节点需要访问另一节点上的数据时，该请求将穿越 NSD 服务器。因此，NSD 服务器有助于在集群节点之间移动数据。

仲裁节点 (Q) 能够与 GPFS- SNC 集群中的其他仲裁节点共同工作，以确定集群是否运行及能否用于处理传入的客户端请求。仲裁节点还可用于在节点发生故障时确保整个集群的数据一致性。集群管理员可在创建集群期间或向集群添加节点时，为选定的一组节点指定仲裁节点服务。通常情况下，您会发现每个机架上具有一个仲裁节点，建议仲裁节点最多为 7 个。在设置 GPFS-SNC 集群时，管理员应当将节点数量定义为奇数，如果异构集群，请将该仲裁节点角色分配给故障风险最低的计算机。如果其中一个仲裁节点丢失，其余的仲裁节点仍可相互通信，就像这个仲裁仍然完好无损一样。

每个 GPFS-SNC 集群只有一个 *Cluster Manager (CM)* 节点，该节点由仲裁节点选定（而不是由集群管理员指定）。Cluster Manager 将负责确定仲裁模式、管理

磁盘租赁、检测故障、管理恢复以及选择文件系统管理器节点。如果 Cluster Manager 节点发生故障，仲裁节点将立即检测中断并指定替代节点。

GPFS-SNC 集群还具有一台主集群配置服务器 *(P)*，用于维护集群配置文件（此角色已于集群创建期间指定给单一节点）。如果此节点发生故障，系统将启用自动恢复协议，指定另一节点承担这项任务。*辅助配置服务器 (S)* 是可选的，但我们强烈建议生产集群纳入一个辅助配置服务器，原因在于发生故障时，辅助配置服务器可通过接替主集群配置服务器角色来删除 GPFS-SNC 集群中的 SPOF。如果主集群配置服务器和辅助集群配置服务器同时发生故障，集群配置数据仍将完好无缺（因为所有节点上的集群配置数据均已复制），但需要人工干预才能恢复集群。

每个 GPFS-SNC 集群可拥有一个或多个文件系统管理器 *(FSM)* 节点，由 Cluster Manager 节点以动态形式选定（虽然集群管理员能够为此角色定义可用节点池）。文件系统管理器负责文件系统配置、使用、磁盘空间分配和配额管理。该节点的内存和 CPU 需求比集群中的其他节点要高；通常情况下，我们建议大型 GPFS-SNC 集群具有多个文件系统管理器。

图 5-1 中的最后一项服务是 *Metanode (MN)*。GPFS-SNC 集群中每个打开的文件都有一个 Metanode，负责维护文件元数据的完整性。在几乎所有情况下，Metanode 服务均在打开时间最长的特定文件所在的节点上运行。访问文件的所有节点均可直接读写数据，但只有 Metanode 能够写入元数据更新。各文件的 Metanode 均独立于任何其他文件的 Metanode，并且可以移动到其他节点，以满足各种应用程序需求。

*故障组*定义为共享某个常见故障点的一组磁盘，该故障点可能会导致它们同时失去效用。例如，集群中各节点上的所有磁盘就可组成一个故障组，因为如果这个节点发生故障，节点中的所有磁盘也将立即无法使用。GPFS-SNC 复制方法很好地诠释了故障组的概念，该集群将负责确保不同故

障组中磁盘上的每个复制数据和元数据块均已备份。如果某组磁盘失效，GPFS-SNC 可从其他复制位置恢复数据。

如果您选择 GPFS-SNC 组件进行安装，BigInsights 图形安装程序将负责为您处理 GPFS- SNC 集群创建和配置。安装程序会提示您在即将分配 Cluster Manager 和仲裁节点服务的节点上输入数据。这种安装方法采用 BigInsights 工作负载使用的典型默认配置。GPFS-SNC 高度可定制，因此对于专业化安装，您可以通过修改模板脚本和配置文件在图形安装程序外进行安装和配置（虽然某些自定义配置只在安装程序内部使用）。

GPFS-SNC 故障转移方案

无论您在集群中使用 GPFS-SNC 还是 HDFS，Hadoop MapReduce 框架均可以在文件系统层上运行。运行的 Hadoop 集群根据 GPFS-SNC 或 HDFS 存储层上运行的 TaskTracker 和 JobTracker 服务来为 MapReduce 工作负载提供支持。虽然这些服务器并非文件系统层的特定服务器，但确实是 Hadoop 集群的 SPOF 典型代表。这是因为如果 JobTracker 节点发生故障，所有正在执行的作业也将失败，但这种故障极为罕见，并且很容易恢复。而HDFS NameNode 故障则严重得多，如果磁盘损坏且未进行备份，很可能会导致数据丢失。此外，对于具有数百万兆字节存储空间的集群而言，重新启动 NameNode 可能需要花费几个小时，因为需要从磁盘提取集群元数据并将其读入内存，并且必须重复执行前一检查点的所有更改。而在使用 GPFS-SNC 时，则不需要 NameNode（它仅仅是一个 HDFS 组件）。

集群中可能发生各种形形色色的故障，下面我们逐一介绍 GPFS-SNC 如何处理这些故障情形：

- Cluster Manager **故障** Cluster Manager 发生故障后，仲裁节点会立即检测到这一故障状况，并从仲裁节点池中选出新的 Cluster Manager。集群操作继续执行，对集群整体运作造成的中断时间极短。

- **文件系统管理器节点故障** 仲裁节点首先检测到这一状况，然后要求 Cluster Manager 从集群的各节点中挑选出新的文件系统管理器节点。此类故障对集群整体运作造成的中断时间极短。
- **辅助集群配置服务器故障** 仲裁节点能够检测到这一故障，但需要集群管理员手动指定一个新节点作为辅助集群配置服务器。尽管此节点处于故障状态，集群操作仍将继续执行，但某些同时需要主服务器和辅助服务器提供支持的管理命令可能无法正常运行。
- **机架故障** 其余的仲裁节点将确定集群的哪个部分仍正常运行，哪些节点随之中断。如果 Cluster Manager 位于运行中断的机架上，仲裁节点将在运行正常的集群部分选出一个新的 Cluster Manager。同样，如果文件系统管理器节点位于故障机架上，Cluster Manager 也会选出一个新的文件系统管理器节点。集群将为故障机架上丢失的各数据节点实施标准恢复策略。

GPFS-SNC POSIX 合规

GPFS-SNC 与 HDFS 之间的显著架构差异在于，*GPFS-SNC 是一种内核级文件系统，而 HDFS 在操作系统上运行。*因此，HDFS 本身具有大量限制和低效功能。绝大多数这些限制均源自 HDFS 并未实现全面 POSIX 合规这一事实。而另一方面，GPFS-SNC 则很好地支持了POSIX 。因而，您的 Hadoop 集群更加稳定、更加安全，也更加灵活。

易用性和存储灵活性

所有应用程序均可查看 GPFS-SNC 中存储的文件，就像在一台计算机上查看任何其他文件一样。例如，在复制文件时，所有授权用户均可使用传统操作系统命令列出、复制及移动 GPFS-SNC 中的文件。而在 HDFS 中却无法做到这一点，用户需要登录 Hadoop 才能查看集群中的文件。此外，如果您要在 HDFS 中执行任何文件操作，需要先了解 Hadoop 命令 shell 环境的工作原理，并通晓特定的 Hadoop 文件系统命令。

所有这些都需要对 IT 人员开展额外的培训。经验丰富的管理员或许能够尽快适应，但这不过是学习曲线问题。至于复制或备份，HDFS 只提供的唯一机制，即通过 Hadoop 命令 shell 手动复制文件。

BigInsights GPFS-SNC 的全面 POSIX 合规性有助于您管理 Hadoop 存储，且方法与在 IT 环境中的任何其他计算机上的执行方式并无二致。因此，可以让使用者快速熟悉Hadoop的技能，使工作变得更加轻松。例如，您的传统文件管理工具可以运行，备份与恢复工具和程序也可以运行。GPFS-SNC 实际上对备份功能进行了扩展，纳入了时间点 (PiT) 快照备份、异地复制及其他实用工具。

其他应用程序甚至可以借助 GPFS-SNC 与 Hadoop 共享同样的存储资源。这在 HDFS 中无法实现，您需要预先定义 Hadoop 集群专用磁盘空间。不仅必须估算需要在 HDFS 中存储的数据量，而且还必须猜测 MapReduce 作业输出所需的存储空间（因工作负载不同可能会大相径庭）；同时不要忘记将 Hadoop 系统工具创建的日志文件所占用的空间计算在内！采用 GPFS- SNC 后，您只需要关心磁盘本身，没有必要为 Hadoop 预留专用存储空间。

并发读/写

GPFS-SNC POSIX 合规的另一项额外好处在于，它还会为您提供 MapReduce 应用程序或任何其他应用程序无需追加文件内容即可更新现有的集群文件的能力。此外，GPFS-SNC 还能保证多个应用程序并行写入 Hadoop 集群中的同一文件。同样，HDFS 也无法实现上述任何功能，这些文件写入限制局限了 HDFS 对大数据生态系统的操作范围。例如，BigIndex 或 Lucene 全文索引（对于任何一类有意义的文本处理和分析工具而言都是一个重要组件）已准备就绪，随时可在 GPFS-SNC 中使用。正如我们前面所说，如果您使用 HDFS，Lucene 需要在本地文件系统（而不是在 HDFS）中维护自身索引，因为 Lucene 需要持续更新现有文件，并且所有这一切（您猜对了）均会增加执行复杂度和性能开销。

安全性

如前所述，GPFS-SNC 与 HDFS 不同，GPFS-SNC 是一种内核级文件系统，这意味着它可以享受操作系统级安全防护。您还可以通过 POSIX 访问控制列表 (ACL) 扩展权限，从而实现 HDFS 所无法实现的精确用户特定权限。

使用当前的 Hadoop 版本（0.20，截至本书编写之日），HDFS 尚未意识到操作系统级安全性优势，也就是说具有集群访问权限的任何人均可读取集群数据。虽然 Hadoop 0.21 和 0.22 将各安全功能整合到 HDFS（要求用户进行身份验证并获得授权才能使用集群），这种新安全模型对于管理员而言更加复杂，且灵活性也远不能与 GPFS-SNC 相提并论（我们将在本章后几节就安全问题进行详细探讨）。

GPFS-SNC 性能

GPFS 的最初用途是作为高性能超级计算机的存储系统。凭借这一高性能血统，GPFS-SNC 得以具备三大主要特点，实现了必要的灵活性和功能，因而性能始终优于 HDFS。

第一个特点是*数据条带化*。在 GPFS-SNC 中，集群会对所有数据进行分段和镜像（stripe and mirror everything，SAME），因而会在集群的所有磁盘间条带化数据。由于系统可以并行读取和处理数据，条带功能加快了连续读取速度，因而读取操作快于 HDFS。这大大支持了排序（需要较高的连续吞吐量）等操作。在 HDFS 中，系统根据复制因子复制整个集群的文件，但并未在多个磁盘间条带化各数据块。

GPFS-SNC 的另一个性能推手是*分布式元数据*。在 GPFS-SNC 中，文件元数据广泛分布在整个集群中，通过大量随机数据块读取操作提高了工作负载的性能。而在 HDFS 中，元数据集中存储在 NameNode 上，这不仅形成了单一故障点，而且还成为随机存取工作负载的性能瓶颈。

由于*客户端缓存*，GPFS-SNC 集群的随机存取工作负载实现了额外的性能提升。而 HDFS 中则没有此类缓存。良好的随机存储性能对于 Hadoop 工作负载至关重要，尽管底层设计偏爱顺序存取。例如，Pig 和 Jaql 应用程序探索

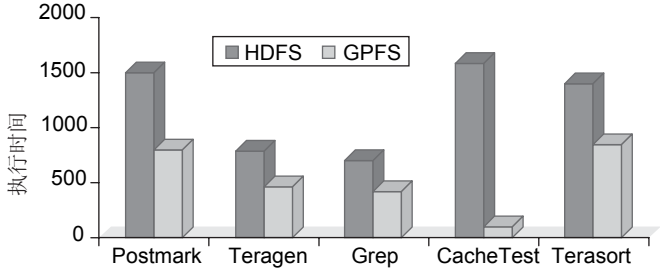

图 5-2 *GPFS 与 HDFS 性能基准比较图*

性分析活动将会大大得益于良好的随机 I/O 性能。

IBM Research 利用同一集群上的标准 Hadoop 工作负载对 GPFS-SNC 和 HDFS 开展了基准测试（一个使用 GPFS-SNC，一个运行 HDFS）。性能提升令人印象深刻（律师要求我们必须发布以下标准声明，否则不允许打印：您的实际结果可能会有所不同），如图 5-2 所示。

从图中可以看出，相较于默认 HDFS 文件系统，GPFS-SNC Hadoop 工作负载实现了显著的性能提升。根据上述结果及我们的内部测试，我们预计 GPFS-SNC 上运行的 10 节点 Hadoop 集群的性能将与带有约 16 个同类节点的 HDFS 上运行的 Hadoop 集群性能水平相同。

GPFS-SNC Hadoop 实现企业质量

为突出重要性，我们花费了大量时间说明 GPFS-SNC 为 Hadoop 集群带来的所有好处。这些优势展现了 IBM 的资产、经验和研究如何强化及补充 Hadoop 开源社区的创新成果，从而为企业级大数据平台奠定基础。总而言之，在 Hadoop 集群中运用 GPFS-SNC 能够全面实现可用性、安全性、性能和可管理性优势。

压缩

在 Hadoop 设置中处理大量数据时，我们呼吁采用压缩理念。一方面，您

可以节省大量空间（尤其是在 Hadoop 中的默认设置下：假设每个存储数据块复制三次）；另一方面，由于写入数据量的减少，数据传输速度也加快了。在选择压缩计划之前，您需要考虑两个重要事项：*拆分压缩*以及采用的压缩算法的*压缩和解压缩速度*。

拆分压缩

在 Hadoop 中，如果文件的大小大于集群数据块设置，将会拆分（分割）文件（通常将一个文件拆分为多个数据块）。对于未压缩的文件，这意味着各拆分文件可以通过不同的 mapper 并行处理。图 5-3 展示了未压缩文件，并用垂直线表示拆分文件和数据块边界（在这种情况下，拆分文件和数据块大小相等）。

当文件（特别是文本文件）经过压缩时，并发症随之出现。对于绝大多数压缩算法，各拆分文件不能独立于同一文件的其他拆分部分独自解压缩。更具体地说，这些压缩算法是“不可拆分的”（请在讨论压缩和 Hadoop 时牢记这一术语）。当前版本的 Hadoop（编写本书时使用的是 0.20.2 版）不支持对已压缩的文本文件的拆分。对于使用序列格式或 Avro 格式的文件而言，这并不算是一个问题，因为这些格式均具有内置同步点，因而可以进行拆分。对于不可拆分的压缩的文本文件，MapReduce 处理功能将仅限于单个 mapper。

例如，假设图 5-3 中的文件是一份 1 GB 的文本文件且位于 Hadoop 集群内，您在 BigInsights 中设置的默认数据块大小为 128 MB，这意味着您的文件将生成 8 个数据块。使用 Hadoop 中的常规算法压缩此文件时，则不能再并行处理每个压缩拆分文件，因为文件只能作为整体解压缩，不可能基于拆分文件作为独立的部分解压缩。图 5-4 对这个处于压缩（二进制）状态的

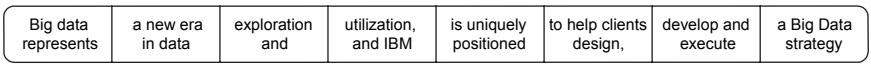

图 5-3 *Hadoop 中的未压缩可拆分文件*

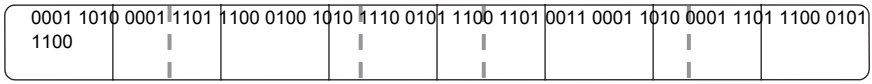

图 5-4 *已压缩的不可拆分文件*

文件进行了描述，拆分文件无法单独解压缩。注意，分割边界用虚线表示，数据块边界用实线表示。

由于 Hadoop 0.20.2 本身不支持可拆分文本压缩功能，压缩文本文件的所有拆分部分只能通过一个 mapper 进行处理。对于许多工作负载而言，这样将会造成重大的性能冲击，因而不是可行方案。但是，Jaql 经过配置后可了解文本文件的可拆分压缩，并会通过并行 mapper 自动处理它们。您可以使用 TextInput- Format 输入格式代替 Hadoop 标准为其他环境（如 Pig 和 MapReduce 程序）手动执行这项操作。

压缩和解压缩

古语有云“天下没有免费的午餐”，这句话用在压缩技术上再适合不过了。世界上并不存在神奇的魔力，实质上，您纯粹是以使用 CPU 周期来节省磁盘空间。所以，我们首先做出这样一个假设：压缩 Hadoop 集群中的数据可能会导致性能下降，因为将数据写入集群后，（CUP 密集型）压缩算法需要使用 CUP 周期和时间来压缩数据。同样，在读取数据时，由于需要使用 CUP 周期和时间来解压缩这些已压缩的数据，针对压缩数据的任何 MapReduce 工作负载均可能会出现性能下降现象。这就产生了一个难题：您需要在节省存储空间与增加性能开支之间进行权衡。

注意 *如果您的应用程序具有严格的输入输出限制（通常针对许多仓库风格的应用程序），可能会发现应用程序性能有所提升，因为具有输入输出限制的系统往往具有备用 CPU 周期（在 CPU 中作为闲置 I/O 等待）可用于运行压缩和解压缩算法。例如，如果您使用闲置 I/O 等待 CPU 周期执行压缩，可以实现较高的压缩率，最终有更多的数据流过 I/O 管道，这意味着性能提速的这些应用程序需要从磁盘提取大量数据。*

BigInsights 的额外优势：IBM LZO 压缩

BigInsights 含有 IBM LZO 压缩编解码器，支持拆分压缩文件，从而通过 MapReduce 作业并行处理各压缩拆分部分。

一些 Hadoop 在线论坛对如何使用 GNU 版本的 LZO 实现可拆分压缩进行了介绍，那么 IBM 为什么要单独创建一个版本，为什么不使用 GNU LZO 代替呢？首先，IBM LZO 压缩编解码器在压缩文件时“不”会创建索引，因为它采用固定长度的压缩数据块。与此相反，GNU LZO 算法则采用可变长度的压缩数据块，由于需要利用索引文件通知 mapper 拆分压缩文件的安全位置，从而增加了操作复杂性（对于 GNU LZO 压缩，这意味着 mapper 需要在解压缩和读取操作期间执行索引查询。采用此类索引还会增加管理开销，因为如果您移动压缩文件，同时也需要移动相应的索引文件）。其次，许多公司（包括 IBM）均制定了法律政策，以防止购买或发布包含 GNU 公共许可证 (GPL) 组件的软件。这意味着，Hadoop 在线论坛上所述的方法需要增加额外的管理开销和配置工作。此外，还有一些企业制定各种策略，严格限制部署 GPL 代码。IBM LZO 压缩与 BigInsights 全面集成，与其余 BigInsights 产品遵循“同样”的企业友好型许可协议，也就是说您可以使用它，并且不会产生太多麻烦，也不会衍生与 GPL 替代产品相关的复杂问题。

在下一版 Hadoop（版本 0.21）中，bzip2 算法也将支持拆分。不过，bzip2 的解压缩速度比 IBM LZO 慢，因此 bzip2 对性能要求较高的工作负载而言并非理想的压缩算法。

图 5-5 显示了先前示例的的压缩文本文件，但文件处于可拆分状态，且各拆分

图 5-5 *可拆分压缩文本文件*

部分可以通过其各自的 mapper 单独解压缩。注意，各拆分部分大小相等，表示压缩数据块长度固定。

压缩编解码器	文件扩展名	是否可拆分	压缩度	解压缩速度
IBM LZO	.cmx	是	中	最快
bzip2	.bz2	是，但只有 Hadoop 0.21 以后的产品才提供	最高	慢
gzip	.gz	否	高	快
DEFLATE	.deflate	否	高	快

在上表中，您可以看到 BigInsights 平台上提供的 4 种压缩算法（IBM LZO、bzip2、gzip 和 DEFLATE）及其各自的某些特点。

最后，下表中展示了 Hadoop 中常用的三种最受欢迎的压缩算法的某些基准测试比较结果（原始资料来源：http://stephane.lesimple.fr/wiki/blog/lzop_vs_compress_vs_gzip_vs_bzip2_vs_lzma_vs_lzma2-xz_benchmark_reloaded）。这项基准测试使用 96 MB 文件作为测试用例。注意，IBM LZO 算法的性能和压缩率与本项基准测试中测试的 LZO 算法不相上下，但还有一项独特优势：无需使用索引即可进行拆分以及依据企业友好型许可证发布。

压缩编解码器	压缩大小 (MB)	压缩速度（秒）	解压缩速度（秒）
LZO	36	1	0.6
bzip2	19	22	5
gzip	23	10	1.3

管理工具

为协助管理您的集群，BigInsights 加入了基于 Web 的管理控制台，让您以互动的方式实时监控集群。BigInsights 控制台还提供了图形工具，用以检查 BigInsights 环境的运行状况，包括集群中的节点、作业（应用程序）运行状态及 HDFS 或 GPFS-SNC 文件系统的内容。该过程将自动纳入 BigInsights 安

装，并且默认情况下在端口 8080 上运行，但您也可以在安装流程期间指定其他端口。

Apache Hadoop 由许多不同的组件构成，每个组件均具有各自的配置和管理。此外，Hadoop 集群往往较大，因而会对管理带来各种各样的挑战。BigInsights 管理控制台提供了单一的一致性集群视图来简化您的工作。您可以通过这一平台添加和删除节点、启动和停止节点、监控应用程序状态、检查 MapReduce 作业的状态、查看日志记录、监控平台（存储、节点和服务器）的整体运行状态、启动和停止可选组件（例如 ZooKeeper）、浏览 BigInsights 集群文件等。

图 5-6 展示了控制台主页的一个片段。从中您可以看到，管理控制台集中处理 Hadoop 集群管理所需的各项任务。仪表板负责概述系统运行状况，您可以深入了解，以获取各个组件的相关详细信息。

在图 5-6 中，您还可以看到一个 HDFS 选项卡，可让您浏览 HDFS 目录结构，查看存储了哪些文件以及创建新目录。同时，您还可以通过此工具将文件上载至 HDFS，但这一操作不太适合大文件。如需将大文件上载至 Hadoop 集群，我们建议采用其他机制，如 Flume。

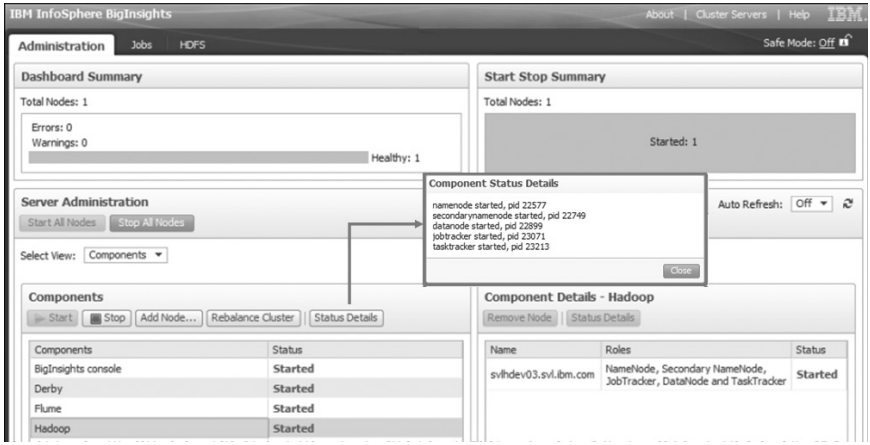

图 5-6 *BigInsights 管理控制台示例*

GPFS 选项卡（如果您使用的是 GPFS-SNC，而不是 HDFS）是 BigInsights 控制台的一个特色，可提供同样的功能，包括浏览数据以及与 GPFS-SNC 文件系统交换数据。

此 Web 控制台的主页还可让您链接至底层开源组件提供的各种集群服务器工具。这种工具非常便于管理员使用，因为它能够让管理员从单一控制台轻松访问各种内置工具。

图 5-7 展示了管理控制台的多个屏幕画面。顶部是 Job Status 页，您可以在其中查看集群摘要信息，如状态、构成集群的节点数量、任务容量、正在执行的作业等。如果某项作业正在执行并且您具有相应的授权，则可以取消正在运行的作业，如本图底部所示。要查看有关某项特定作业的详细信息，请从列表中选择作业，然后在该页的 Job Summary 部分查看作业详细信息。您甚至能够进一步深入，获取更加详细的作业信息。例如，您可以查看作业配置（显示为 XML）和计数器信息，其中详细描述了执行期间使用的 mapper 数，以及完成作业读取/写入的字节数。

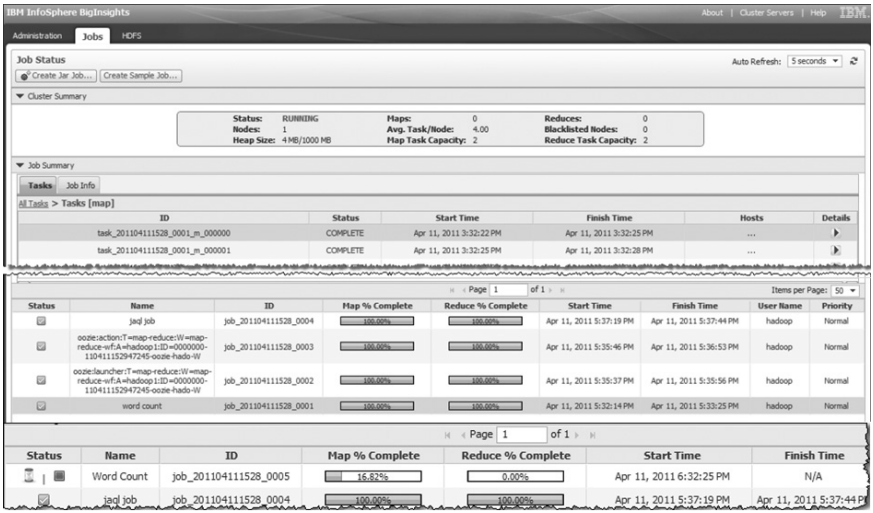

图 5-7 *BigInsights 管理控制台的 Job Status 和 Jobs in Progress 窗口*

BigInsights 还提供了其他一些工具优势，而这在常规 Hadoop 环境是绝对无法实现的。例如，通过彩色的界面显示作业和任务的状态，并且可以根据一定的间隔自动刷新。

安全性

安全性是企业软件的一个重要课题，在运用开源 Hadoop 的情况下，您需要认清采用 Hadoop 的一些明显缺点。BigInsights 通过保护管理接口和关键 Hadoop 服务访问安全解决了这些问题，这的确是一个好消息。

BigInsights 管理控制台构建作为集群的网关。其特色在于，通过支持LDAP身份验证增强安全性。LDAP 和反向代理支持可帮助管理员限制授权用户访问。此外，集群以外的客户端必须使用 REST HTTP 进行访问。相比之下，Apache Hadoop 在集群中的每个节点上均具有多个开放端口。必须开放的端口越多（其中许多端口均位于开源 Hadoop 之内），操作环境的安全性越低，原因在于暴露的区域未实现最小化。

BigInsights 可配置与轻型目录访问协议 (LDAP) 凭据服务器通信，从而进行身份验证。控制台与 LDAP 服务器之间的所有通信均使用 LDAP（默认）或同时使用 LDAP 和 LDAPS (LDAP over HTTPS) 。BigInsights 安装程序可帮助您定义 LDAP 用户、LDAP 组与四个 BigInsights 角色（系统管理员、数据管理员、应用程序管理员和用户）之间的映射关系。BigInsights 安装完成后，您可以向 LDAP 组添加用户，也可以从中删除用户，从而授权或撤销对各种相应控制台功能的访问权限。

某具有竞争性 Hadoop 供应商现已集成 Kerberos 安全协议，该供应商仅提供服务和某些操作工具，但不支持其他身份验证协议（Active Directory 除外）。BigInsights 使用 LDAP 作为默认身份验证协议，且开发团队一直强调使用 LDAP，因为相较于 Kerberos 和其他协议，LDAP 协议更加便于安装和配置。也就是说，BigInsights 确实能够提供可插拔身份验证支持，支持 Kerberos 等其他协议。

BigInsights 采用 GPFS-SNC，其提供的安全产品复杂度更低，从本质上而言比基于 HDFS 的替代产品更加安全。再次重申，由于 GPFS-SNC 是一种内核级文件系统，因而能够很自然地感知操作系统中定义的用户和组。

即将发布的 Apache Hadoop 变革改善了 HDFS 的安全性，但由于 HDFS 并非内核级文件系统，因而依然需要增加复杂度和处理开销。正因如此，IBM 在经营企业级产品的经验成为 BigInsights 的坚强后盾，因而您得以建立更加安全、稳健且更易于维护的多租户解决方案。

企业集成

IBM 版本的大数据关键组件，对于集成所有关联数据源至关重要；促使 Hadoop 引擎满足您的所有存储和处理需求并非偶然。您还必须对企业进行其他投资，因而充分利用自身资产（请参见本书第 2 章的左右手棒球比喻）将成为关键。企业集成是 IBM 较为擅长的又一个领域。就其本身而论，BigInsights 支持与大量数据源（包括 Netezza、DB2 for Linux, UNIX and Windows）进行数据交换，其他关联数据商店均通过 Java Database Connectivity (JDBC) 接口、InfoSphere Streams、InfoSphere Information Server（特别是 Data Stage）、R 统计分析应用程序等进行数据交换。

Netezza

BigInsights 包含一个连接器，用以在 BigInsights 集群与 Netezza 设备之间实现双向数据交换。Netezza Adapter 作为 Jaql 模块实施，从而让您在数据库交互期间充分利用 Jaql 的简便性和灵活性。

Netezza Adapter 支持表拆分（概念类似于文件拆分）。这就需要对表进行分区，并将各部分分配至特定的 mapper。这样，便可以并行处理您的 SQL 语句。

Netezza Adapter 利用 Netezza 的外部表功能，您可以将此比作有形的外部 UNIX 管道。外部表则使用 JDBC。在这种情况下，各 mapper 均作为数据

库客户端。基本上，mapper（作为客户端）会连接 Netezza 数据库，然后开始从 Netezza 基础架构创建的 UNIX 文件读取数据。

DB2 for Linux, UNIX and Windows

您可以通过以下两种方式在 BigInsights 与 DB2 for Linux, UNIX, and Windows 之间交换数据：通过一系列 BigInsights 用户定义函数 (UDF) 从 DB2 服务器交换数据，或通过 JDBC 模块从 BigInsights 集群交换数据（将在下节进行介绍）。

BigInsights 与 DB2 的集成包含以下两个主要组件：一组 DB2 UDF 和一个 Jaql 服务器（用以侦听 DB2 请求，位于 BigInsights 集群之上）。Jaql 服务器则是一款中间件组件，可以从 DB2 9.5 服务器或更高版本接收 Jaql 查询处理请求。具体来说，Jaql 服务器可从 DB2 服务器接收以下几类 Jaql 查询：

- 读取 BigInsights 集群中的数据。
- 上载（或删除）BigInsights 集群中的 Jaql 代码模块。
- 提交即将在 BigInsights 集群上运行的 Jaql 作业（可参阅先前从 DB2 上载的各模块）。

从 DB2 服务器运行这些 BigInsights 函数可让您轻松实现与传统应用程序框架内 Hadoop 的集成。借助这些函数，数据库应用程序（其他方面无法感知 Hadoop）即可运用从 DB2 提取关系数据的同一 SQL 接口访问 BigInsights 集群数据。此类应用程序现已能够利用 BigInsights 集群的并行性和规模化，而无需进行额外配置或投入其他开销。虽然相比常规 Hadoop 应用程序，这种方法会产生额外的性能开销，但却是将大数据处理集成到现有 IT 应用程序基础架构的有效方式。

JDBC 模块

Jaql JDBC 模块可让您从装有标准 JDBC 驱动程序的任何关系型数据库读取和写入数据。这意味着，您可以轻松地使用当今市场上的各种主要数据库仓库产品交换数据以及发出 SQL 语句。

在完成 Jaql MapReduce 集成后，各 map 任务均可访问表的特定部分，从而为分区数据库并行处理 SQL 语句。

InfoSphere Streams

正如您在第 6 章所见，Streams 是一款用于进行流数据实时分析的 IBM 解决方案。Streams 包含一个 BigInsights 接收适配器，从而让您将流数据直接存储至 BigInsights 集群。Streams 还包含一个 BigInsights 源适配器，用于协助 Streams 应用程序读取集群数据。BigInsights 与 Streams 集成衍生了大量有趣的可能性。概括而言，您能够创建一个基础架构以便实时响应事件（就像通过 Streams 处理数据一样），同时利用大量现有数据（通过 BigInsights 进行存储和分析）报告响应。您也可以利用 Streams 作为大型数据插入引擎，以过滤、装饰或以其他方式操控即将存储在 BigInsights 中的数据流。

Streams 应用程序可以借助 BigInsights 接收适配器将控制文件写入 BigInsights 集群。可配置 BigInsights 在出现此类文件时作出响应，这样即可在集群中触发运行更深层次的分析操作。对于更高级的方案，Streams 触发文件还可包含查询参数，以便自定义 BigInsights 分析。

Streams 和 BigInsights 通过 *Advanced Text Analytics Toolkit*（最初以 IBM Research 代码 SystemT 问世）共享同一组文本分析功能。此外，两款产品还共享同一个最终用户 Web 接口（用以参数化和运行工作负载）。未来版本还将陆续统一分析工具。

InfoSphere DataStage

DataStage 是一款数据提取、转换和加载 (ETL) 平台，能够跨越范围广泛的各种数据源和目标应用程序集成海量数据。通过将自身角色扩展为数据集成代理，DataStage 已经能够与 BigInsights 开展协作，并可从 BigInsights 集群来回推送数据。

BigInsights DataStage 连接器已与 HDFS 和 GPFS-SNC 文件系统实现全面集成，充分利用集群架构的优势，以便将所有批量数据并行写入同一文件。在

使用 GPFS-SNC 的情况下，也可以并行写入所有批量数据（GPFS-SNC 与 HDFS 不同，GPFS-SNC 已实现全面 POSIX 合规性）。

DataStage 集成的结果是，BigInsights 现在能够与成功连接 DataStage 的任何其他软件产品快速交换数据。各种计划已部署妥当，目的是加强 Information Server 和 BigInsights 之间的衔接，如能够从 DataStage 精心策划各项 BigInsights 作业，制定强大灵活的 ETL 方案。此外，已设计好扩展 Information Server 信息分析和治理功能，以便纳入 BigInsights。

R 统计分析应用程序

BigInsights 针对 Jaql 纳入了 R 模块，协助您将统计计算 R Project（有关详细信息，请参阅 www.r-project.org）集成到 Jaql 查询。此后，R 查询即可从 Jaql MapReduce 功能中受益，同时还能并行运行 R 计算。

改进工作负载调度：Intelligent Scheduler

开源 Hadoop 自带了先进先出 (FIFO) 基础调度器以及支持替代计划方案的可插拔架构。两种可插拔计划工具均可通过以下 Apache Hadoop 项目获取：Fair Scheduler 和 Capacity Scheduler。两种调度器存在某些相似之处，它们都为小型作业提供最低水平的资源以避免资源匮乏（Fair Scheduler 位于 BigInsights 内，而 Capacity Scheduler 则不是）。这两种调度器均无法提供足够的控制，因而不能确保实现最佳集群性能，也不能为管理员提供满足可自定义工作负载管理需求所需的必要灵活度。例如 FAIR 在确保全面运用各种资源开展工作方面相当擅长，但却无法为您提供类似于 SLA 的同等细粒度控制。

IBM Research 性能专家研究了 Hadoop 中出现的各种工作负载调度问题，并精心设计出一项名为 Intelligent Scheduler（以前称为 *FLEX* 调度器）的解决方案。这款调度器对 Fair Scheduler 进行了扩展，通过不断调整分配执行作业的插槽最低数量来进行操控。Intelligent Scheduler 包含用于进行工作负载优化的各项指标。管理员可根据整个集群的状况选择这些指标，个人用

户也可以根据特定的作业需求做出选择。您可以有选择地权衡这些指标以平衡各优先事项，最大限度地减少或增加独立作业指标总数。

以下就是一些 Intelligent Scheduler 控件示例，您可以借此对工作负载进行优化：

average response time 该调度器尽量为小型作业多分配资源，以确保快速完成这些作业。

maximum stretch 按作业所需的资源量比例为各作业分配资源。换句话说，大型作业具有较高的优先级。

user priority 为特定用户执行的作业分配尽可能多的资源，直至完成作业。

自适应 MapReduce

IBM Research 工作负载管理和性能专家一直与 Hadoop 开展广泛合作，发掘性能优化机遇。IBM Research 开发出了一个全新概念 —— *自适应 MapReduce* (Adaptive MapReduce)，通过实现各 mapper 的自我感知能力以及感知其他 mapper 的能力来扩展 Hadoop。这种做法有助于各 map 任务适应自身环境并做出有效决策。

在即将开始 MapReduce 作业之前，Hadoop 会将数据分割成多个片段，称作拆分片段 (split)。系统会为每个拆分片段分配一个 mapper。为确保工作负载实现平衡，我们以波浪的形式部署这些 mapper，旧 mapper 完成拆分片段处理后新 mapper 立即接续。在这种模式下，拆分片段越小意味着 mapper 越多，从而有助于确保工作负载平衡以及最大限度地降低故障成本。然而，拆分片段小同时也会导致集群开销增加，因为每项映射任务的启动成本也随之提升。对于映射任务启动成本较高的工作负载而言，拆分片段大往往意味着效率高。自适应式 map 任务运行方法使 BigInsights 能够充分发挥两者的长处。

自适应 MapReduce 是 *自适应 mapper* 概念的一次伟大实践。自适应 mapper 通过跟踪中央存储库中的文件拆分片段状态扩展了传统 Hadoop mapper 的各项功能。自适应 mapper 每次完成片段处理操

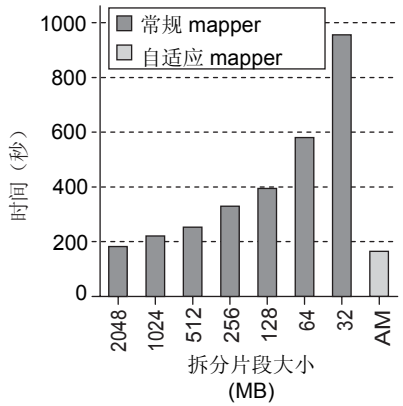

图 5-8 *利用设置类似的联接工作负载与高映射以及任务启动成本与自适应 mapper 进行基准测试的结果*

作均会参照中央存储库，并锁定另一个拆分片段直至处理完成。这意味着，对于自适应 mapper 而言，由于各 mapper 仍保持开放以消耗额外的拆分片段，因而只需部署一波 mapper 。锁定新拆分片段的性能成本远远低于启动新 mapper 的成本，同时还能显著提升性能。图 5-8 展示了设置类似联接工作负载的基准测试结果，自适应 mapper 消除了高昂的映射任务启动成本。自适应 mapper 测试结果（参见 AM 条形图）均基于 32 MB 的小拆分片段得出。测试仅使用了一波 mapper ，由于有效避免了额外 mapper 的启动成本，因此实现了显著的性能成本节约。

对于某些工作负载而言，任何不平衡状况均可能会随拆分片段的增大而放大，进而造成额外的性能问题。采用自适应映射后，只需调整作业降低拆分片段规模即可避免工作负载不平衡现象，同时还不会影响性能。由于仅采用一波 mapper，没有很多额外 mapper 的 mapper 启动成本，因而工作负载也未减少。图 5-9 展示了 TERASORT 记录联接查询基准测试的结果，其中各 map 任务之间的不均衡现象导致大拆分片段工作负载不平衡。自适应 mapper 测试结果（再次参见 AM 条形图）均基于 32 MB 的小拆分片段得出。测试仅使用了一波 mapper，由于有效避免了额外 mapper 的启动成本，因此实现了显著的性能成本节约。

大量新型自适应 MapReduce 性能优化技术目前均处于开发阶段，并将在未来的 BigInsights 版本中陆续发布。

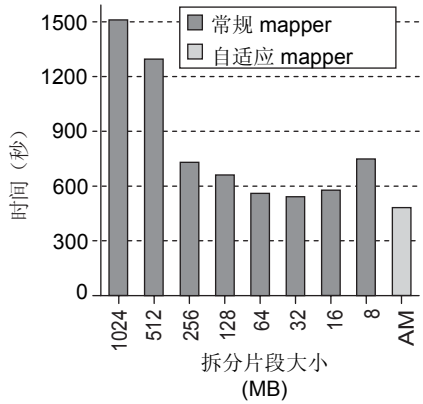

图 5-9 *TERASORT 记录联接查询基准测试结果*

数据发现和可视化：BigSheets

到目前为止，我们已在本章中对 BigInsights 基础架构的各基础层面进行了讨论。这些都是促使 Hadoop 运行更快、更加可靠、更加灵活且适于在企业中使用的重要特色。但存储数据的最终目标在于从中获利，这也正是我们采用 BigInsights 分析功能的原因所在。这是 BigInsights 的另一项主要特色，使它挣脱 Hadoop 分发技术的局限性，成为真正的大数据分析平台。BigInsights 与 Apache Hadoop 组件或竞争性捆绑 Hadoop 分发产品不同，BigInsights 包含专为可视化以及执行大数据组分析而设计的工具。BigInsights 竭尽自身的所有分析功能掩盖 MapReduce 复杂性，促使您的分析师专心开展分析工作，而无需分心处理并行应用程序编程。

虽然 Hadoop 促使大数据分析成为可能，您需要成为一名真正的编程人员，精确掌握 MapReduce 范例才能有效探索数据。BigInsights 包含一款基于浏览器的可视化工具（名为 BigSheets），可协助业务用户利用熟悉的电子表格界面充分发挥 Hadoop 的强大功能。BigSheets 无需编程，也无需特殊管理。只要您会使用电子表格，就可以使用 BigSheets 对海量数据实施分析，不论采用的哪种数据结构。

运用 BigSheets 执行大数据分析分为以下三个简单步骤：

1. *收集数据*。您可以从多种来源收集数据，包括抓取 Web 数据、本地文件或网络文件。支持多种协议和格式，包括 HTTP、HDFS、Amazon S3 本机文件系统 (s3n) 及 Amazon S3 数据块文件系统 (s3)。抓取 Web 数据时，您可以指定要抓取的网页及抓取深度（例如，抓取深度为开始网页及链接开始页面的各页面这两个数据集合）。其中还包含一个 BigSheets 扩展工具，带有自定义数据导入插件。例如，您可以构建插件来搜集 Twitter 数据并将数据纳入 BigSheets 集合。
2. *提取并分析数据*。信息收集完毕后，您可以在电子表格界面中查看数据样本，如图 5-10 所示。此时，您可以使用 BigSheets 中提供的电子

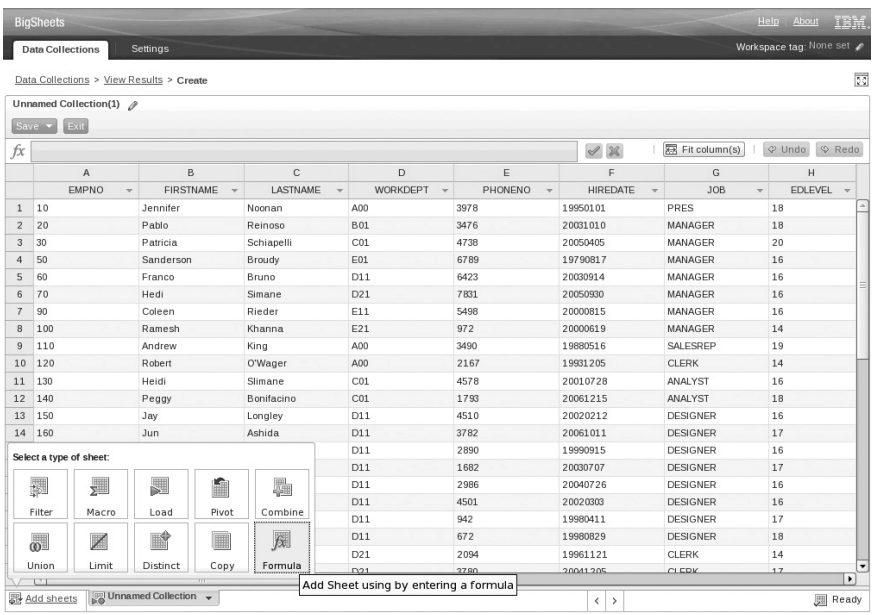

图 5-10 *在 BigSheets 中分析数据*

表格类工具操控数据。例如，您可以合并不同集合的列，运行公式或过滤数据。同时，还可以纳入自定义宏插件，结合自身的数据集合进行使用。在构建工作表和完善分析的同时，您还可以查看样本数据中期分析结果。只有单击 Run 按钮时，才会对整个数据集合运用分析。由于您的数据可能从数 GB、TB 到 PB 不等，所以最好利用小型数据集进行迭代处理。

3. *探索并可视化数据*。从数据表对数据运行分析之后，您可以运用可视化功能帮助自身了解数据。BigSheets 提供了以下一些可视化工具：

   - **标签云** 用于显示字词频率；字词频率越大，在数据表中出现的频率也就越高。请参阅图 5-11 查看示例。

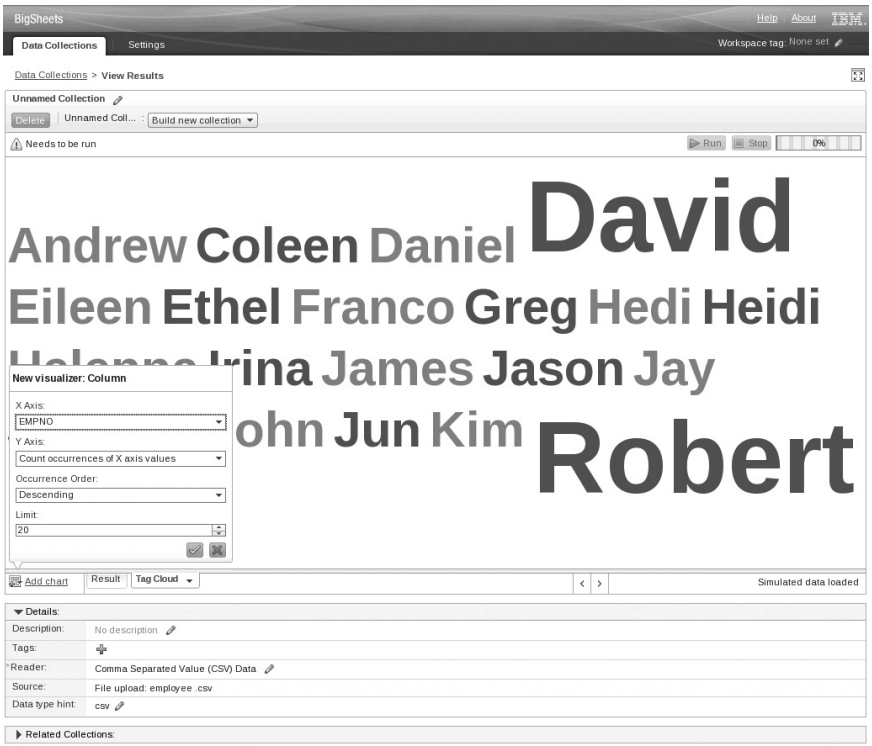

图 5-11 *在 BigSheets 中分析数据*

- **饼图** 用于显示比例关系，片段的相对大小表示数据所占的比例。
- **地图** 用于显示世界地图或美国地图的覆盖数据值。
- **热图** 类似于地图，但比地图多一个维度，用于显示地图覆盖数据值的相对密度。
- **条形图** 用于显示指定列的数值频率。BigSheets 能够利用自身的可视化工具进行全面扩展。因此，您可以纳入自定义插件对数据执行专门的呈现。

Advanced Text Analytics Toolkit

虽然 BigSheets 专用于业务线用户，但 BigInsights 纳入了更加深入的分析功能，如文本分析。

随着企业不断竭力了解自身的庞大文本数据存储库，文本分析技术的重要性也随之不断提升。这项技术可能包含在点击流日志文件中探寻客户 Web 浏览模式、通过电子邮件分析查找欺诈指示因素，或从社交媒体消息评估客户情绪。为应对上述挑战以及未来可能出现的更多挑战，BigInsights 纳入了 Advanced Text Analytics Toolkit，它以文本分析引擎（IBM Research 自 2004 年起开始开发，研发代码：SystemT）为特色。自那时起，IBM 一直在不断发展这个 Advanced Text Analytics Toolkit，现已将其引擎纳入多款 IBM 产品，包括 Lotus Notes、IBM eDiscovery Analyzer、Cognos Consumer Insight、InfoSphere Warehouse 等。截至目前，Advanced Text Analytics Toolkit 一直只是作为嵌入式文本分析引擎发布，最终用户无法真切地看到它。在 BigInsights 中，Advanced Text Analytics Toolkit 将作为文本分析平台提供，其中包含多款开发人员工具、一种易于使用的文本分析语言、一款 MapReduce 就绪文本分析处理引擎以及多个预置文本提取程序。Advanced Text Analytics Toolkit 还提供多语言支持，包括支持多种双字节字符语言。

文本分析的目标在于读取非结构化文本和提炼洞察。例如，文本分析应用

程序可阅读某段文本并根据各种规则派生出结构化信息。这些规则均已在提取程序中预先定义，可以识别文本字段内的人员名称。请考虑以下文本：

In the 2010 World Cup of Soccer, the team from the Netherlands distinguished themselves well, losing to Spain 1-0 in the Final. Early in the second half, Dutch striker Arjen Robben almost changed the tide of the game on a breakaway, only to have the ball deflected by Spanish keeper, Iker Casillas. Near the end of regulation time, winger Andres Iniesta scored, winning Spain the World Cup.

这些提取产品是一系列注释文本，在这段话中以下划线文本的形式显示。

以下是从这段文本样本中派生的结构化数据：

Name	Position	Country
Arjen Robben	Striker	Netherlands
Iker Casillas	Goalkeeper	Spain
Andres Iniesta	Winger	Spain

在提取程序以及与提取程序协作的应用程序开发过程中，面临的挑战在于确保结果的准确性。准确可以分为两个因素：一是“精度”，是指结果中相关项所占的百分比（所得的结果是否有效？），二是“检索率”，是指从文本中检索到的相关结果百分比（原始文本中所有的有效字符串是否均已显示？）分析师开发提取程序和应用程序时，他们会反复精炼，对其精度和检索率进行优化。

当前的替代方法和文本分析基础架构为分析师提出了大量挑战，它们往往表现不佳（包括准确度和速度方面）且难以使用。这些替代方法依赖只能通过提取程序和过滤器体系流动的原始文本。这种方法缺乏灵活性和有效性，往往会产生处理冗余。这是因为工作流程后期应用的提取程序可能会对前期已经完成的任务执行处理操作。现有的工具包也因它们的表现而受到局限（尤其是查询的粒度级别），因而分析师不得不开发自定义代码。反过来又在提炼结果集的准确性（精度和检索率）方面造成了更长的延迟、更高的复杂度和难度。

BigInsights Advanced Text Analytics Toolkit 提供了一种强大而又灵活的文本分析方法。Advanced Text Analytics Toolkit 的核心在于其 Annotator Query Language (AQL)，是一种全声明性文本分析语言，这意味着没有“黑盒”，所有模块均可自定义。换句话说，所有数据均采用同一种语义进行编码，并遵循相同的优化规则。文本分析语言格式的结果极富表现力且速度极快。据我们所知，当今市场上还未出现其他全声明性文本分析语言。您可以找到高级别或中级别的声明性语言，但它们均采用无法自定义的锁定式黑盒模块，限制了产品灵活性，并且难以进行性能优化。

AQL 提供了一种类 SQL 语言用于构建提取程序。它极富表现力且灵活易用，同时还提供常见语法。例如，以下 AQL 代码用于定义提取人员姓名和电话号码的相关规则。

```
create view PersonPhone as select P.name as person, N.number as
phone
from Person P, Phone PN, Sentence S where Follows(P. name.
PN.number, 0, 30)
      and Contains(S.sentence, P.name) and Contains(S. sentence,
PN.number)
      and ContainsRegex(/\b(phone|at)\b/, SpanBetween(P. name,
PN.number));
```

图 5-12 对前面的代码块定义的提取程序进行了可视的再现。

Advanced Text Analytics Toolkit 包含多种 Eclipse 插件，用以提升分析师的工作效率。编写 AQL 代码时，编辑器还提供语法高亮显示和语法错误自动检测功能（参见图 5-13）。

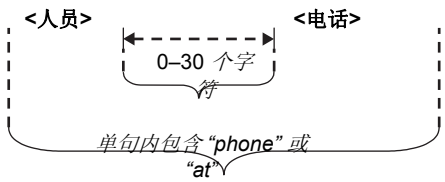

图 5-12 *提取程序规则的可视表达式*

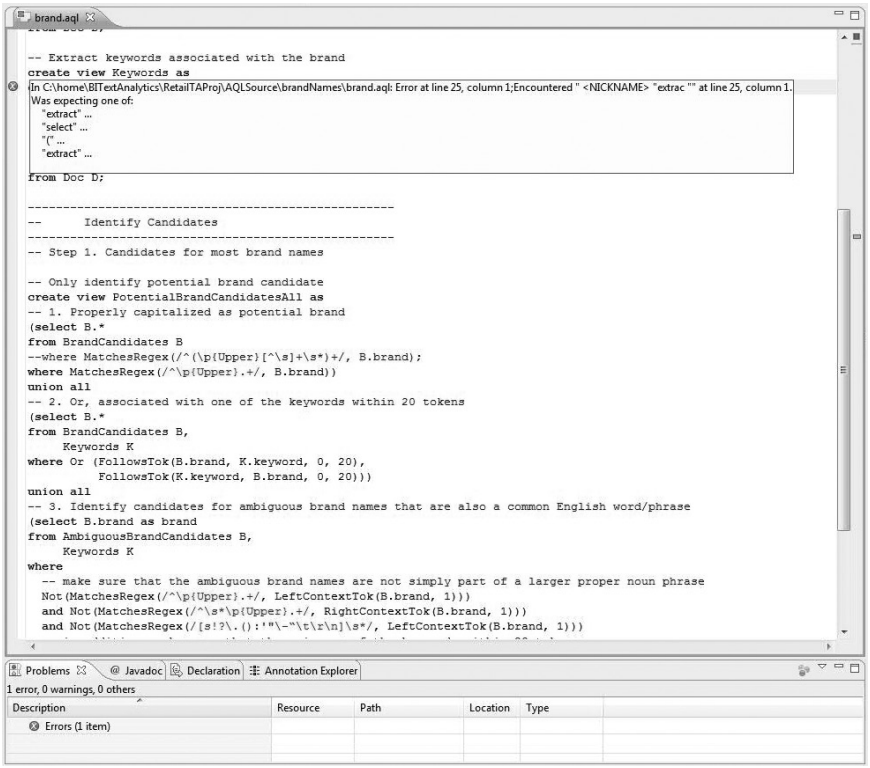

图 5-13 *AQL 编辑器*

同时还纳入了一种以数据子集对提取程序进行测试的工具。此工具对分析师精炼提取程序的精度和检索率至关重要。针对整个数据集（最大数据量：PB）测试逻辑性这一操作可能效率极低并会造成巨大浪费。

分析师面临的一项主要挑战在于确定一直以来的文本变化根源。人们可能会难以辨别哪些提取程序需要进行调整，以便对生成的注释进行相应的调整。为帮助做到这一点，Provenance 查看器（如图 5-14 所示）将提供交互可视化功能，以便显示究竟哪些规则会对生成的注释造成影响。

我们还新增了一种生产工具帮助分析师快速启动和运行，即为 Advanced Text Analytics Toolkit 纳入了预置提取库。其中包括专供提取以下内容的提取程序：

购置	地址	联盟
分析师收益预估	城市	公司盈利公告
公司收益预期	大洲	邮政编码
国家/地区	日期时间	邮箱地址
合资企业	地点	并购
Notes 电子邮件地址	组织机构	人员
电话号码	省/市/自治区	URL

AQL 的全声明性特性使其代码可实现高度优化。与前文所述的更加严格的文本框架方法相比，AQL 优化程序可确定提取指令的执行顺序，以便最大限度地提升效率。因此，Advanced Text Analytics Toolkit 的基准测试结果比处于领先市场地位的各替代框架快十倍（参见图 5-15）。

Advanced Text Analytics Toolkit 集 BigInsights 的高速度与企业稳定性于一身，呈现出无与伦比的价值主张。文本分析开发人员可随意查阅 Advanced Text Analytics Toolkit 与 BigInsights 的集成详细信息（如图 5-16 中所述）。一旦成品 AQL 完成编译，将立即进行性能优化，并生成 Analytics Operator Graph (AOG) 文件。您可以通过 BigInsights Web 控制台提交此 AOG 作为分析作业。一旦提交，系统将

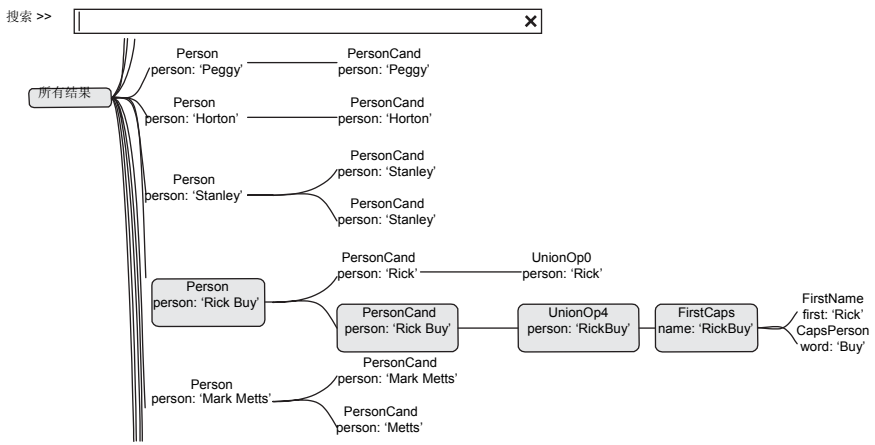

图 5-14 *Provenance 查看器*

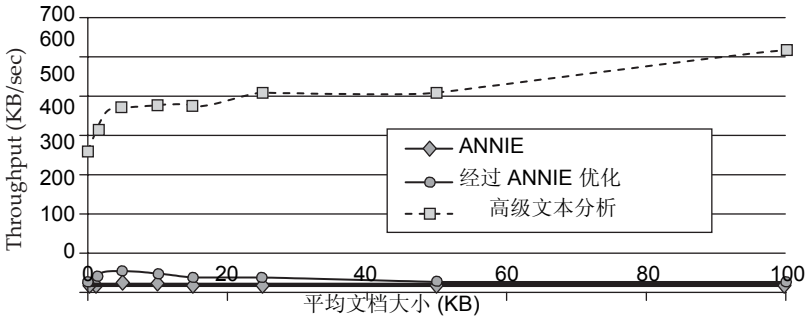

图 5-15 *Advanced Text Analytics Toolkit 性能基准测试*
图字：Throughput (KB/sec)：吞吐量（KB/s）

会使用 BigInsights 集群上即将执行的各 mapper 分发此 AOG。一旦开始执行作业，每个 mapper 均会执行 Jaql 代码，实例化自身的 Advanced Text Analytics Toolkit 运行时，并将其应用至 AOG 文件。各 mapper 文件拆分片段文本均通过工具包的运行时运行，随后将传回注释文档流作为结果集。

添加完所有功能后，BigInsights Advanced Text Analytics Toolkit 将为您提供开发文本分析应用程序所需的一切工具，帮助您从巨量文本数据中实现最大价值。不仅会提供广泛工具支持大型文本分析开发，而且还将进行最终代码高度优化，方便在 Hadoop 集群中进行部署。

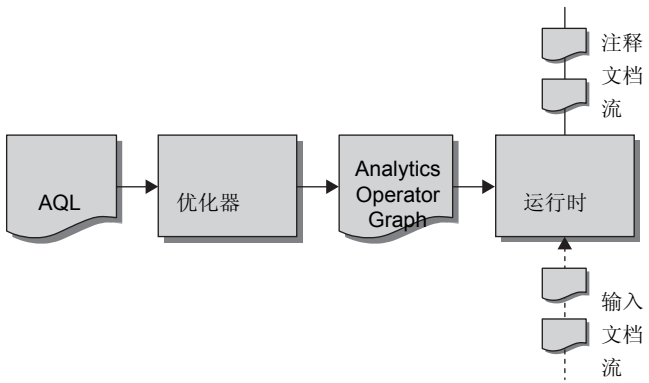

图 5-16 *Advanced Text Analytics Toolkit 与 BigInsights 集成*

机器学习分析

我们预计 2012 年 BigInsights 将纳入 Machine Learning Toolkit，该工具包由 IBM Research 开发（研发代码：SystemML）（免责声明：我们并非保证此功能将于 2012 年首次公开亮相，我们的意思是它迟早会与用户见面）。它将提供一个专用平台供统计师和数学家对 BigInsights Hadoop 集群中的数据执行高性能统计和预测分析。其中还包含一种高级机器学习语言，在语义上类似 R（一种开源统计计算语言），分析师可借此应用统计模型进行数据处理。其中纳入了大量固有数据挖掘算法和统计模型，并且随时可供定制使用。

Machine Learning Toolkit 包含一个引擎，用来将以机器学习语言表达的统计工作负载转换成并行 MapReduce 代码，这样复杂操作会消失无踪，分析师将会感到非常方便。总之，分析师不再需要同时担任 Java 程序员，也不需要将 MapReduce 因素纳入自己的分析应用程序。

Machine Learning Toolkit 由 IBM Research 专家团队精心研发，该专家团队包含多名性能专家、统计博士和数学博士。他们的主要目标在于实现分析师在 Hadoop 环境下执行复杂统计分析所需的高性能和易用性。正因如此，此工具包的最大特色在于其生成低级 MapReduce 执行计划的各项优化技术。这使得相比在 MapReduce 中直接实施算法，统计作业实现了大幅度的性能改进。从此分析师不仅不需要对统计应用程序运用 MapReduce 编码技术，而且其编写的机器学习代码也针对 Hadoop 的出色性能进行了高度优化。

大规模索引编制

为支持其分析工具包，BigInsights 纳入了一种大规模索引和搜索解决方案构建框架，称为 BigIndex。这种索引编制组件包含透过 Hadoop 编制索引、优化、合并以及复制索引所需的各种模块。搜索组件则包含可编程搜索、多面

搜索以及在本地和分布式部署中搜索索引等模块。尤其是在执行文本分析时，强大的索引对于确保分析工作负载的优越性能至关重要。

BigIndex 构建于开源 Apache Lucene 搜索库和 IBM Lucene Extension Library (ILEL) 之上。IBM 是 Lucene 项目的主要参与者，并且长期以来通过 ILEL 开发了大量 Lucene 增强功能。由于这种索引编制引擎的多功能性，BigIndex 中运用的这些技术已被广泛部署到大量产品中。除 BigInsights 以外，我们还将其全面纳入 Lotus Connections、IBM Content Analyzer 和 Cognos Consumer Insight（简单列举几项）。同时，IBM 还利用 BigIndex 驱动其 Intranet 搜索引擎，此项目称为 Gumshoe，在 *Hadoop in Action*（作者：Chuck Lam [Manning Publications, 2010]）一书中进行了大量记载。

BigIndex 目标在于提供大规模索引编制和搜索功能，以便利用并集成 BigInsights。对于大数据分析应用程序而言，这意味着能够在数千 TB 的数据中执行搜索，同时还能保持亚秒级搜索响应速度。BigIndex 实现这一目标的一个主要途径在于，利用各种针对性搜索分布架构为大数据环境要求的不同类型的搜索活动提供支持。BigIndex 可以编制下列类型的索引：

- **分区索引** 这种索引可通过元数据字段分割成单独的标记体（例如客户 ID 或日期）。用户通常只对这些标记体执行搜索，因而系统能够通过运行查询调度程序将查询路由至相应的索引。
- **分布式索引** 该索引会被分发到碎片中，大量碎片集合在一起构成逻辑索引。每次搜索均会对所有碎片进行评估，从而有效地实现了并行索引关键字查询。
- **实时索引** 用户能以近乎实时的速度将实时来源数据（例如 Twitter）添加到索引。并行分析数据，并在分析完成后立即更新索引。

图 5-17 对 BigIndex 部署进行了描述，其中我们运用 BigInsights 集群编制索引，并在其自身的共享集群中执行搜索。

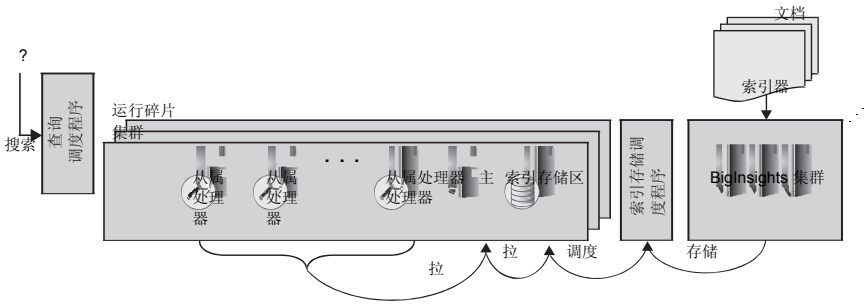

图 5-17 *分布式 BigIndex 部署*

为此类分布式环境生成和部署索引的操作包含以下几个步骤，如图 5-17 所示：

1. **数据注入** 将文档注入 BigInsights 集群。
   此操作可通过任何可用手段完成。例如，通过 Flume 注入日志文件流、Streams 处理的诊断数据或者 HDFS 或 GPFS-SNC 内存储的 Twitter 源。
2. **数据解析** 解析文档，选择需要编写索引的字段。这对于解析选择性算法至关重要：我们需要在大覆盖度（用户可以执行搜索的索引字段）和优质（索引字段越多性能越慢）之间做出权衡。如有需要，可在此处运用文本分析技术。
3. **数据分面** 通过隔离和提取各数据面（如类别）识别当前文档与其他文档的关联方式，用户可能希望借此缩小搜索范围以及深化搜索结果——例如，年、月、日或国家/地区、州、市。
4. **数据索引** 此索引基于 Lucene 文本索引，但具有多种扩展。这些文档均由索引器（部署作为 MapReduce 作业）通过 Hadoop 编写索引。将会生成两个索引：一个 Lucene 索引及一个分布式索引（由多个 Lucene 索引构成，每个索引均代表独立的标记体）。多面索引功能已与单 Lucene 索引和分布式索引全面集成。
5. **索引合并** 索引一旦生成，便会立即被调度至 Runtime Shard Cluster 进行存储。主集群会将这些索引从索引存储区中拉出，并将它们与其本地索引

合并在一起。这一点与常规数据库索引不同，您可以根据需要插入或删除值。此索引是一种经过优化的数据结构。因此，需要将增量更改合并至这一结构。

6. **索引复制** 从属处理器可从主处理器复制更新，并随时可供用户搜索查询。
7. **索引搜索** BigIndex 通过多个接口公开分布式搜索功能，包括 Java API（一种使用 Jaql 的脚本语言）及类 REST HTTP API。

BigInsights 小结

现在让我们对本章所述的各部分内容进行一下小结，BigInsights 是针对静态大数据分析的快速、稳健和易用的平台。借助我们的图形安装、配置和管理工具，集群管理工作变得简便易行。通过利用 GPFS-SNC 存储数据，不仅可以改进性能，而且还能实现数据维护高可用性和灵活性。IBM LZO 压缩模块的加入，促使您无需获得许可即可通过高性能算法压缩数据，省去不少麻烦。此外还提供了其他一些性能特征（如自适应 MapReduce 和 Intelligent Scheduler）帮助您维护稳定的服务级别协议，同时大数据分析技术也将随之衍生大批用户群体。说到分析技术，BigInsights 能够为各种广泛的用户群体提供大量功能。对于业务用户，BigSheets 是一种专为实现大数据可视化的简便工具。而对于深入分析，BigInsights 则可提供业内领先的文本分析工具包和引擎。不久，IBM Research Machine Learning Analytics Toolkit 也将与广大用户见面。我们认为这代表着一场令人难以置信的伟大胜利，在 IT 届堪称独一无二。您可以通过 BigInsights 获取由全球最大的企业研究机构、深谋远虑的开发团队及 IBM 全球支持网络作为强大支持后盾的完整分析解决方案。

6

IBM InfoSphere Streams：分析移动的大数据

鉴于您已经对 IBM 处理专为企业构建强化的 Hadoop 环境中出现的大型数据分析问题的独特方式有所了解，现在我们探讨一下 IBM 大数据解决方案的另一个层面：移动数据分析。采用 BigInsights 后，我们将通过提供信息海洋保障您的竞争优势，IBM InfoSphere Streams (Streams) 可帮助您监控流过环境的海量数据流。您可以深入挖掘数据流，为您的企业获取对时间敏感的竞争优势，也可以像身处海量数据流中的绝大多数用户一样只是充满敬畏地任由它们浩浩荡荡地流过。这就是 Streams 的用武之地。其设计可让您充分利用大规模并行处理 (MPP) 技术来分析数据，同时由于它不断流动，因而您还能实时监控发生的问题并采取行动、更加有效地做出决策，进而提高收益。

在深入研究本章内容之前，让我们首先来澄清一下 *Streams* 和*流 (streams*) 的含义；大写版本是指 IBM InfoSphere Streams 产品，而小写版本则是指数据流。鉴于这一点，让我们来看一下 Streams 的基础知识、使用案例及定义其工作原理的部分技术基础。

InfoSphere Streams 基础知识

Streams 是一个强大的分析计算平台，实现了实时分析数据（仅存在微小的延迟）。Streams 不再像 BigInsights 一样搜集大量数据、操控数据、将数据存储到磁盘上，然后进行分析（换句话说，是指静止数据分析），Streams 可让您对移动数据运用分析技术。在 Streams 中，数据将会流过有能力操控数据流（每秒钟可能包含数百万个事件）的运算符，然后对这些数据执行动态分析。这项分析可触发大量事件，使企业利用即时的智能实时采取行动，最终改善业务成果。当数据流过这些分析组件后，Streams 将提供运算符将数据存储至各个位置（包括 BigInsights 或其他数据仓库），或者如果经过动态分析某些数据被视为毫无价值，则会丢弃这些数据（要么由于数据无意义，要么由于数据虽然存在特定用途，但要求持久性不强）。

如果已经对复杂事件处理 (CEP) 系统非常熟悉，您可能会在 Streams 中发现一些相似之处。不过，Streams 的设计可扩展性更高，并且支持的数据流量也比其他系统多得多。此外，您还将了解 Streams 为何具有更高的企业级特性，包括高可用性、丰富的应用程序开发工具包和高级调度。

您可以将数据流比作一系列连结运算符。初始运算符（或单一运算符）通常是指 Source 运算符。这些运算符可读取输入数据流，然后反过来发送下游数据。中间步骤由执行特定操作的各种运算符组成。最后，每条进入动态分析平台的通道都有多个出口，并且在 Streams 中，这些输出内容被称为 Sink 运算符（就像水一样从水龙头喷涌而出流入厨房水槽）。稍后我们将在本章中对所有这些运算符逐一进行详细介绍。

我们将 Streams 作为一个平台，因为您几乎能够以任何方式构建或自定义 Streams，从而提供应用程序来解决各种业务问题；当然，它还是一个企业支持平台，因为这些运算符中的每一个均可在集群中的独立服务器上运行，从而提高可用性、可扩展性和性能。例如，Streams 提供了丰富的工具环境，

可帮助您设计流应用程序（稍后将在本章进行介绍）。Streams 的另一个好处在于，它与 BigInsights 共享同一 Text Analytics Toolkit，因而能够在整个大数据平台上实现技能和代码段重用。当您的流应用程序部署准备妥当后，Streams 将在运行时根据集群类负载平衡和可用性指标自主确定处理元素 (PE) 的运行位置，从而使其能够重新配置运算符在其他服务器上运行，确保一旦服务器或软件发生故障，数据仍能持续流动。同时，您还能够以编程的方式指定哪些服务器上运行哪些运算符，并可在特定的服务器上运行流逻辑。

这种可自定义的自主式流平台只需增加额外的服务器和分配这些服务器上运行的运算符，即可增加执行数据流分析的服务器数量。Streams 基础架构则负责确保数据在运算符之间的成功流动，无论运算符在不同的服务器上运行还是在同一服务器上运行：这样即可提供从最初建立小平台到根据需要不断发展平台所需的高度敏捷性和灵活性。

Streams 不仅极其适用于结构化数据，而且也适用于其他 80% 的数据（包括传感器数据、语音、文本、视频、财务以及许多其他来源生成的非传统半结构化数据或非结构化数据），这一点与 BigInsights 极为类似。最后，由于 Streams 和 BigInsights 均隶属于 IBM 大数据平台，您将会发现针对移动和静止大数据构建的分析技术均具有大量相同的高效功能。例如，从 Text Analytic Toolkit 构建的提取程序也可以在 Streams 或 BigInsights 中进行部署。

InfoSphere Streams 行业用例

为使您在一定程度上了解 Streams 技术是否适用于您的环境，我们认为有必要提供一些行业用例案例。很显然，我们无法在短短的一本书中全面覆盖各行各业，但我认为这部分内容能够让您对 Streams 技术可能为环境带来的广泛发展可能性产生一定的思考（深呼吸，做好准备，因为您的大脑即将兴奋起来）。

金融服务部门 (FSS)

金融服务部门及其子产业是一个典型例子，流数据分析技术可为该行业提供

无与伦比的竞争优势（以及监管优势，具体情况取决于您的业务状况）。它能够同时跨越多个市场和国家/地区，以极低的延迟分析巨量交易数据和市场数据，因而公司能够实现微秒级响应速度，通过差价交易和业务风险分析制度转亏为盈（例如，某一特殊时刻开展此项交易将会对企业的风险状况造成何种影响？）

FSS 公司还能运用 Streams 实施实时交易监控和欺诈检测。例如，Algo Trading 的平均吞吐率为每秒钟 1270 万条期货买卖消息，并根据情况为其客户生成交易建议，与此同时延迟仅为 130 微秒。正如我们在后文中所说的那样，甚至集成至 Streams 的适配器都能直接通过无处不在的 Financial Information eXchange (FIX) 网关进行连接，并能提供功能丰富的库帮助计算限价买卖理论期权价值。Streams 甚至能够采用多种输入方式。例如，您可以利用 Streams 分析不利模式及其对证券价格的影响，并据此做出短期持券决策。

同样，信用卡公司和零售商也可以利用实时欺诈检测功能开展欺诈和多方欺诈检测（以及识别实时向上销售和交叉销售机遇）。

健康和生命科学

医疗设备旨在快速生成诊断数据。从心电图、温度和血压测量设备到血氧传感器等医疗诊断设备会产生大量数据。实施利用诊断数据并进行分析的意义较之其他任何行业更为重大；除了为公司提供竞争优势外，医疗行业采用 Streams 还有助于挽救生命。

例如，安大略省理工学院 (UOIT) 在多伦多建立了一所智慧医院，利用 Streams 部署新生儿重症监护病房，监控这些我们亲切地称之为小奇迹（“数据婴儿”）的健康状况。这些婴儿在新生儿病房中不断生成数据：每次心跳、每次呼吸、每次异常状况等。每秒钟接收的医疗诊断信息超过 1000 条，

Streams 平台将用作早期预警系统，帮助医生探寻避免威胁生命的恶性感染新途径，速度比过去快高达 24 小时。这里也存在协同效应。独立监控数据流处于正常参数范围（血压和心率等），但将一些数据流与某些特定数值范围结合在一起将会成为一种发病预警，这种情况也时有发生。由于 Streams 对流动数据执行分析，而不只是对绑定值进行审视，因此不仅能够挽救生命，还有助于降低医疗费用（请参阅以下网址：http://www.youtube.com/watch?v=QVbnrlqWG5I）。

电信

电信公司需要管理的通话记录 (CDR) 数量异常惊人。这些信息不仅有助于生成准确的客户账单，而且可以从近期执行的 CDR 中收集大量信息。例如，CDR 分析技术通过分析各社交网络“群体领导者”的访问模式，来帮助防止客户流失。这些群体领导者是指可能会影响其联系人的通信方式倾向，促使改变服务提供商的联系人。通过结合传统分析与社交媒体分析，Streams 能够帮助您识别这些个人、他们所属的网络以及影响的群体。

Streams 还可用于激发实时分析处理 (RTAP) 活动管理解决方案的动力，从而帮助提高活动的有效性、缩短新促销活动与软捆绑的上市时间、帮助发掘新的收入流以及充实客户流失分析。例如，Globe Telecom 利用收集的手机信息制定针对各消费群体的最佳业务促销活动，以及开展活动的最佳时间，进而对其业务产生了深远的影响。Globe Telecom 将新服务上市时间从 10 个月缩短至 40 天，并通过实时促销引擎等大幅提高了销量。

有利于 CDR 分析的功能也同样适用于 Internet 协议详细记录 (IPDR)。IPDR 提供基于 Internet 协议 (IP) 的服务使用和其他活动相关信息，以便运营支持人员借此确定网络质量，检测可能需要提前维护的问题（以免造成网络设

备故障（当然，同样的用例也适用于 CDR）。谈到 CDR 和 IPDR 处理时，Streams 又如何实现实时性与低延迟性呢？我们发现某些详细记录支持的吞吐率峰值可高达 500,000 条/秒，每天分析超过 60 亿条详细记录（是的，您看得没错），每年分析超过 4 PB (4000 TB) 数据；采用 Streams 技术的 CDR 处理产品的持续吞吐率为 1 GBps、X 射线衍射 (XRD) 率为 100 MBps。诚然，Streams 是当之无愧的创新型高效技术。

执法、国防、监督和网络安全

Streams 在加强执法力度以及提高安全性方面带来了巨大的机遇，并为在该领域构建各种应用程序呈现出无限的潜力，如实时名称识别、身份分析、态势感知应用程序、多模式监督、网络安全监测、窃听电话、视频监控、人脸识别等。企业也可以通过流网络和其他系统日志，利用流分析技术监测并预防网络攻击，从而阻止入侵或检测整个网络内的恶意活动。

TerraEchos 利用 InfoSphere Streams 提供隐蔽式传感器监控系统，协助安装敏感设施的公司提前检测入侵者，防止其靠近建筑物或其他敏感设施。他们已经因自身的先进技术赢得了大量奖项（包括 Fiber Optic Sensor System Boarder Application 获得的 Frost & Sullivan 年度创新产品大奖）。最新版本的 Streams 包含一个全新的发展框架，名为 Streams Processing Language (SPL)，将此类应用程序的交付速度提高了 45%，使其启动及交付时间大幅缩短。

本书无法全面介绍的其他领域……

如前所述，我们无法覆盖某款有效产品（如 Streams）可能发挥作用的所有用例和行业，因此我们将在本节中尽量概括性地多介绍一些实际用例。

政府机构可以利用 Streams 的各种广泛实时分析功能，通过监督和天气预报监控野火风险，以及通过实时流量分析管理用水品质和用水量。一些政府还利用

出租车、交通流量摄像头以及嵌入道路来实施智能交通管理的交通传感器传回的 GPS 数据改善某些最拥塞城市的交通状况。这项实时分析技术可帮助他们预测交通模式并调整红绿灯时间来改善交通流量，从而让市民们更加高效地上下班，并借此提高他们的工作效率。

电力行业生成的数据量呈爆炸性趋势增长。现代电网中的智慧电表及各种传感器会以惊人的速度将实时信息传回电力企业。Streams 的内置大规模并行技术可进行实时数据分析，这样能源经销商和电力公司便可随消费者需求的不断变化调整电网发电能力。此外，公司还可以将自然分类系统（如天气或用水管理数据）数据纳入分析流，协助能源交易商满足客户需求，同时预测消费（或消费不足）需求，以实现竞争优势、最大限度地增加公司利润。

制造商需要更加灵敏、准确、富数据质量记录和质量过程控制（例如微芯片制造领域，但适用于任何行业），以便更加有效地预测、避免和遏制既定的违规事件等。太空天气预报、瞬态事件检测和同步加速器原子研究等电子科学领域为 Streams 呈现出另一片广阔的发展空间。正如我们前面所说，从智慧电网到文本分析再到“谈话”分析等 Streams 用例几乎蕴藏着无限可能。

InfoSphere Streams 的工作原理

正如全文所述，Streams 专为开展移动数据分析而设计。您可以将数据流想像成一行多米诺骨牌。如果您推倒一张，那么后面的牌也将产生连锁反应（假设您已将它们正确地排成一行），一张多米诺骨牌倒下的冲力足以导致下一张牌顺势倒下，依此类推。如果您的技术足够好，甚至可以将这一排多米诺骨牌分为多行同时倒下，然后在队列中的某一点重新合并。这样，您就能使多行多米诺骨牌平行倒下，所有冲力均聚集到同一行的下一张多米诺骨

牌（您不要觉得奇怪，根据吉尼斯世界纪录，一次性翻到多米诺骨牌的世界纪录为 430 万张）。Streams 在本质上极为类似，某些数据元素首先在运算符之间流动，一个运算符输出内容变为下一个运算符的输入内容。同样，一个数据记录或数据元组被拆分为多个流，而后可能会在下游合并到一起。整个过程的最大区别在于，玩多米诺骨牌游戏时，一旦某张牌倒下，则直至最后一张都要倒下，而使用 Streams 时，数据将以高速连续不断地流过系统，因此分析的持续信息流将永无休止。

什么是流？

从技术角度而言，流是*通过边缘连接的节点图*。图中的每个节点都是“运算符”或“适配器”，均能够在某种程度上处理流内的数据。节点可以不包含输入和输出，也可以包含多个输入和输出。一个节点的输出与另外一个或多个节点的输入相互连接。图形的边缘将这些节点紧密联系在一起，表示在运算符之间移动的数据流。图 6-1 展示了一个简单的流图，它可以从文件中读取数据，将数据发送到名为 Functor 的运算符（此运算符能够以某种编程方式转换所传入的数据），然后将这些数据传入另一个运算符。在此图片中，流数据被传送至 Split 运算符，而后又将数据传入文件接收器或数据库（具体情况视 Split 运算符的内部状况而定）。

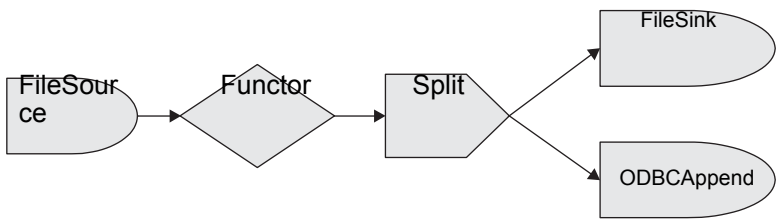

图 6-1 *简单数据流，首先对某些数据进行转换，然后根据某种预定义逻辑将其拆分为两份可行输出内容*

数据流过一个元组流。从关系数据库角度而言，您可以将它们想像成数据组。不过，当 Streams 处理半结构化数据和非结构化数据时，元组是一个表示数据包的抽象概念。您可以将元组看作给定对象的属性集。元组中的每个元素均包含属性值，属性值可以是字符串、数字、日期，甚至是某种二进制对象。

某些运算符处理独立元组、转换数据，然后传递数据。另外一些运算符则需要处理成组的元组，而后才能发送结果。例如，请思考排序操作：每次仅处理一个元组无法进行数据排序。您必须有一组数据才能排序，然后进行传递。出于这个原因，某些运算符将处理数据“窗口”（实际上由一系列元组聚合而成）。运算符本身将根据运算符内的窗口表达式定义窗口中的元素数目。例如，运算符可以定义某个窗口包含接下来进入运算符的 N 个元组，或者也可以定义窗口包含接下来 M 秒进入运算符的所有元组。还有许多其他方法也可以定义窗口，实际上，某些窗口可以移动（因此应定义滑动元组窗口），而另一些窗口则更为固定（它们将聚合成为元组集合，在某个时间点或事件点清空包含所有这些元组的运算符，随后聚合形成下一个集合）。稍后，我们将在本章（各运算符相关主题）中就窗口化问题展开讨论，但我们首先要认清一个重要概念，即 Streams 并非只是讨论每次操作一个元组的问题，而是大型数据集实时分析。

Streams Processing Language

Streams Processing Language (SPL) 是一种结构化应用程序开发语言，Streams 以此来创建应用程序。与过去的各个版本相比，此编程框架对于 Streams 而言收益和效率均有所提高；实际上，有位客户宣称由于采用富 SPL，Streams 应用程序工作效率提高了 45%。毕竟，技术本身非常强大，但如果您无法运用它来满足现行的业务需求，又有什么用呢？

SPL 编写的基于 Streams 的应用程序利用 Streams 编译器进行编译，从而将它们转换为二进制 (bin) 可执行代码，然后在 Streams 环境中运行以完成集群内的各

服务器任务。SPL 程序是一种基于文本的图形表示，我们已在上一节中讨论过：SPL 程序定义来源、接收器及它们之间的运算符，而后反过来定义流处理方法，以及各运算符在流内的行为方式。在本章后面，我们将对这种简化 Streams 应用程序构建过程的应用程序开发工具展开讨论。但是，孩子们在允许使用计算器之前也是以同样的方式学习乘法计算，我们先来看一下 SPL，然后再向您介绍这种简便的应用程序开发工具。例如，下列 SPL 代码段表示一个简易流：只有一个来源，流过单个运算符，最终进入单个接收器：

```
composite toUpper {
    graph
      stream<rstring line> LineStream = FileSource() {
            param    file                    : "input_file";
                     format                  : line;
      }
      stream<LineStream> upperedTxt = Functor(LineStream)
{
        output upperedTxt                    : line = upper(line);
      }
      () as Sink = FileSink(upperedTxt) { param file
                : "/dev/stdout"; format              : line;
      }
}
```

在这个 SPL 代码段中，FileSource 运算符从指定文件读取数据，而后将其置于名为 LineStream 的流内。在本用例中，该运算符被称作 Functor 运算符，负责将流数据转换为大写文本，然后将该元组置于名为 upperedTxt 的输出流上。而后 Sink 运算符将读取 upperedTxt 流数据，最后将其发送至标准输出（单就本用例而言）。

此代码段代表最简单的流，只有一个来源、一个运算符和一个接收器。当然 Streams 的强大之处在于它能够跨越多个大型服务器集群运行大规模并行作业，其中每个运算符（或每组运算符）均可在单独的服务器上运行。但在我们深入探讨企业级 Streams 功能之前，让我们先来看一下这款产品中提供的各种适配器。

来源适配器和接收适配器

毫无疑问，为执行数据流分析，数据必须进入流。当然，分析操作完成后，数据流还必须前往某个位置（即使这个位置定义为空，表明字节没有进入任何位置）。让我们来看一下用于插入数据的最基本的来源适配器，以及可以接收数据的最基本的接收适配器。

FileSource 和 FileSink

顾名思义，FileSource 和 FileSink 是用来读取或写入文件的标准适配器。您可以使用参数指定读取或写入操作所使用的文件名称和位置。另一个参数则负责识别文件内容的格式，可以是以下任意一种：

txt 简单的文本文件，其中每个元组都是文件的行

csv 包含逗号分隔值的文件

bin 包含二进制数据元组的文件

line 包含文本数据行的文件

block 由二进制数据块构成的输入流（与 BLOB 非常类似）

还有其他许多可选参数也可用来指定属性，例如列分隔符、行尾标记、分隔符等。

TCPSource/UDPSource 和 TCPSink/UDPSink

TCPSource 和 TCPSink 适配器是 Streams 用来读取和写入套接字的基本 TCP 适配器。使用这些适配器时，您需要指定 IP 地址（使用 IPv4 或 IPv6）和端口，适配器将会从套接字读取数据，然后生成元组进入流。这些适配器的参数与 FileSource 和 FileSink 适配器在数据流格式（txt 和 csv 等）方面毫无差异。UDPSource 和 UDPSink 适配器从 UDP 套接字读写数据的方式与基于 TCP 的适配器完全相同。

Export 和 Import

export 和 import 适配器在流内搭配工作。您可以使用 export 适配器导

出数据并为导出流分配 streamID。一旦为流分配此 ID，同一实例中的任何其他流应用程序均可使用已分配的 streamID 导入此数据。使用 export 和 import 是在同一 Streams 实例下运行的应用程序之间流动数据的绝妙方式。

MetricsSink

MetricsSink 适配器是一种非常有趣并且有用的接收适配器，因为它可让您设置*指定计量表 (named meter)*，每当有元组到达接收器，这些计量表也会随之增加。您可以将这些计量表看作计量器，您可以使用 Streams Studio 或其他工具实施监控。如果您曾经用过上述某种流量计数器（这些黑色橡皮管看起来毫无用途和韵律，毫无缘由地横在路口或路边），您一定会明白，当流量计数器评估通过某一个兴趣点的流量时，MetricsSink 可用来监控流出数据流的数据量和速度。

运算符

很简单，运算符是 Streams 分析引擎的核心。它们从上游适配器或其他运算符提取数据，操控这些数据，然后将所得的元组向下移至下一个运算符。在本节内容中，我们将会探讨一些较为常见的 Streams 运算符，这些运算符可相互串联构建 Streams 应用程序。

Filter

filter 运算符类似于实际的水流、熔炉或汽车过滤器：其目的在于只允许部分流内容通过。Streams filter 运算符会根据用户定义的条件（指定作为运算符参数）从数据流中删除元组。一旦您通过编程方式指定条件，运算符中定义的首个输出端口将负责接收满足该条件的所有元组。您可以选择指定第二个输出端口接收不满足指定条件的所有元组（如果您熟悉提取、转换和加载 [ETL] 流，会发行此操作与 match 和 discard 操作类似。)

Functor

functor 运算符从输入流读取数据，以某种方式转换数据，然后将这些元组

发送至输出流。运用的转换方法可操控流中的任何元素。例如，您可以从流中提取数据元素，也可以输出通过特定 functor 运算符的各个元素的所有运行元素。

Punctor

punctor 运算符负责为流增加标点，而后下游即可运用这些标点将流分别导入多个窗口。例如，假设某个流负责读取通过该流的联系人目录清单并处理这些数据。您可以通过使用 punctor 运算符在应用程序每次观察到流中的姓氏发生变化时向流中添加标点，从而对联系人目录中的姓氏实施持续运行计数。而后您可以在下游的 functor 聚合运算符中使用此标点发送该姓名的运行总数，而后将该计数重置为 0，开始计算下一组姓氏。

Sort

sort 运算符相当容易理解，它只是负责输出接收到的相同元组，但必须以指定的排序顺序执行。这也是我们在讨论流窗口规范应用部分介绍的首个运算符。请仔细想想：如果流代表不断流动的数据，您如何对数据进行排序，因为您不知道下一个到达的元组是否需要排在必须作为输出内容发送的首个元组前面。为解决这一问题，Streams 允许您指定要执行操作的窗口。您可以采用多种方式指定元组窗口：

count 窗口中包含的元组数量

delta 等到流中某个元素的指定属性变更达到给定的增量值

time 希望等待窗口填满的时间量（秒）

punctuation 用于划定窗口界限的标点，参照 punctor 运算符中的定义

除指定窗口化选项外，您还必须指定数据排序方式表达式（例如，按流中

指定的属性排序）。一旦窗口填满，排序运算符将会根据您指定的元素进行元组排序，并将这些元组以预先定义的排序顺序发送至输出端口（然后返回重新填满窗口）。默认情况下，Streams 会以升序进行排序，但您也可以指定自己希望采用降序排序方式。

Join

您很可能已经猜到，join 运算符需要两个流，根据用户指定的条件匹配元组，然后将匹配元组发送至输出流。当某行到达某个输入流后，系统会将匹配属性与第二个输入流中现有的操作窗口元组进行比较，尝试查找匹配项。与关联数据库一样，用户可以使用多种类型的联接，包括 inner joins（只有匹配元组可以通过）和 outer joins（除两个流的匹配元组外，即使没有匹配元组也可以通过其中一个流元组）。使用 sort 运算符时，您必须在每个流中指定元组存储窗口才能进行联接。

Aggregate

aggregate 运算符可用于汇总窗口元组的指定属性或属性集合值；此运算符也依靠窗口化选项组合元组集，以解决 Sort 一节列举的各种同样挑战。aggregate 运算符还允许使用 groupBy 和 partitionBy 参数分割窗口中的元组，以及对这些较小的元组子集执行聚合操作。您可以使用 aggregate 运算符执行 count、sum、average、max、min、first、last、count distinct 和其他形式的聚合操作。

Beacon

beacon 是一种非常有用的运算符，因为用户可用它来创建动态元组。例如，您可以设置 beacon 以预先定义的各种间隔向流发送元组，其中间隔可以是时间段间隔（每 n/10秒发送一个元组）和/或迭代间隔（发送 n 个元组，然后停止）。beacon 运算符在测试及调试 Streams 应用程序方面非常有用。

Throttle 和 Delay

throttle 和 delay 是两种非常有用的运算符，可帮助您处理指定流的计时和流动。throttle 运算符可帮助您设置数据通过流的“步伐”。例如，用户能以指定的速率将零星抵达的元组发送至 throttle 运算符输出端（设置为元组数/每秒）。同样，delay 运算符也可用于更改流计时。用户可简单设置 delay 在特定的延迟时间后输出元组。但是，使用 delay 运算符时，当元组间隔时间达到他们之间的抵达间隔时，元组将退出运算符。也就是说，如果元组 A 比元组 B 早到达 10 秒钟，而元组 B 比元组 C 早到达 3 秒钟，则 delay 运算符将保留两元组到达的计时时间差，当这些元组的延迟时间达到特定的时间量时即退出。

Split 和 Union

顾名思义，split 运算符将采用一个输入流，将数据流拆分成多个输出流。此运算符对元组中的指定属性采用参数化值列表，将元组属性与此列表值进行匹配分析，以确定在哪个输出流上发送元组。union 运算符恰好相反：它需要多个输入流，负责将输入流中的所有元组合并至输出流。

Streams 工具包

除前文所述的适配器和运算符外，Streams 还附带大量多功能工具包，甚至能够加快应用程序的开发速度。这些工具包还可让您连接至特定的数据源，以及操控数据库、金融市场等领域的常见数据。由于 Streams 工具包能够通过 Streams 缩短分析时间，我们认为最好花少许时间对它们进行更为详细的介绍。具体来说，我们将会介绍 Database Toolkit 和 Financial Markets Toolkit 两节内容。

Database Toolkit：Relational 数据库运算符

Database Toolkit 允许流读写 ODBC 数据库或 SolidDB 数据库。它允许流查询外

部数据库以添加或验证流数据，开展进一步的分析。此 Streams 工具包提供的运算符如下：

ODBCAppend 使用 SQL INSERT 命令将数据插入流中的表

ODBCEnrich 从表中读取数据并将其与流中的元组合并

ODBCSource 从表中读取数据并将各行均添加至流中作为元组

SolidDBEnrich 从 SolidDB 表读取数据并将该信息添加至流元组

Financial Markets Toolkit

Financial Information eXchange (FIX) 协议是一项金融市场数据交换标准。此标准定义了与证券交易相关的信息交换数据格式。Streams Financial Markets Toolkit 可提供大量 FIX 协议适配器，如：

FIXMessageToStream 将 FIX 消息转换为流元组

StreamToFIXMessage 将流元组格式规范为有效 FIX 消息，以便进行转换

除上述运算符外，此工具包还提供市场模拟适配器（用以模拟市场报价、交易、订单等）等其他一些有用组件。其中还包含 WebSphere MQ 消息专用适配器和金融市场专用 WebSphere Front Office。总之，此工具包大大缩短了开发、测试和部署流进程来分析金融类市场数据所需的时间。

企业级

过去构建的许多实时应用程序和并行处理环境均已消失无踪，促使 Streams 与众不同的真正核心是其强大的企业级架构和运行环境，它们足以处理最艰巨的流工作负载。IBM 及其研发团队为应对大数据问题精心研制的“武器”才是真正的价值所在。虽然某些公司投入大量 IT 预算尝试自主应对挑战，但

投入这些预算加强核心竞争力和发展业务不是更有意义吗？

大型大规模并行作业具有独特的可用性要求，由于其位于大型集群中，往往注定要以失败而告终。但 Streams 传来福音，Streams 的内置可用性特点充分考虑了这一需求。同时它也认为，在大规模集群中，应用程序创建、可视化和监控都是保持低管理成本（以及良好企业信誉）的重要成功因素。不必担心：Streams 已考虑这一问题。最后，集成企业架构的其余部分对于构建整体解决方案（而不是炉管或单一孤立应用程序）必不可少。我们在本书中反复强调这一核心：*IBM 提供的是大数据平台，而不是大数据产品。*

我们将在本节介绍一些企业层面的大数据流分析问题：可用性、易用性和集成性。

高可用性

配置 Streams 平台时，您需要通知流哪些主机（服务器）将作为 Streams 实例的一部分。您可以在平台中为每台服务器指定三种类型的主机：

- *应用程序主机*是指运行 SPL 作业的服务器。
- *管理主机*运行管理服务控制 SPL 作业流（但不直接明确运行任何 SPL 作业），控制集群内部安全性并监控所有运行作业等。
- *混合主机*可同时运行 SPL 作业和管理任务。

在典型环境中，您将只有一台管理主机，其余服务器均将作为应用程序主机。

执行流应用程序时，处理元素 (PE) 可在不同的服务器上执行，原因相当简单，因为 PE 实质上是构成流应用程序的运算符和适配器。例如，来源运算符可在一台服务器上运行，而后将元组导入另一服务器运行运算符 B。最后这台服务器上的运算符会将元组导入在另一台服务器上运行的接收运算符。

如果 PE 发生故障，Streams 将会自动检测故障并采取任何可行的补救措施。例如，如果 PE 重新启动并重新定位，则 Streams 运行时将自动挑选可用主机

并于其上运行作业，在该主机上启动 PE（根据需要将输入和输出“重新连接”至其他服务器）。但是，如果 PE 继续不断出现故障（或许由周期性底层硬件问题导致），重试阈值可指定在重试次数达到一定数量后，PE 将停留在 stopped 状态，并且需要手动干预才能解决这个问题。如果 PE 可以重新启动，但已定义为不可重新定位（例如，PE 是一款需要在特定主机上运行的接收器），Streams 运行时将自动尝试在同一主机上重新启动 PE（如果可行的话）。同样，如果管理主机发生故障，您可以在其他位置重新启动管理功能（假设您已将系统配置为 RecoveryMode=ON）在这种情况下，恢复数据库必定早已将重新启动管理任务所需的必要信息存储到集群中的另一台服务器之上。

易用性：促使平台易于使用

可用性意味着可部署。Streams 自带基于 Eclipse 的可视工具集（名为 *InfoSphere Streams Studio (Streams Studio)*），可让您创建、编辑、测试、调试、运行，甚至可视化 Streams 图模型和 SPL 应用程序。与其他基于 Eclipse 的应用程序开发外接程序极为类似，Streams Studio 的“目光”与 Streams 相同，包括利用 Streams Explorer 管理 Streams 开发项目。Streams 目光还包括图形视图，可让您监控从一个或多个数据源到一个或多个接收器的流图，从而让您操控该图形管理应用程序拓扑。

运行 SPL 应用程序时，Streams Studio 还能提供大量额外的优势。内置指标可让您审视流应用程序，识别各种主要运行特性，如进出各运算符的元组数量等。日志查看器可让您查看各 Streams 集群服务器上的各种日志，而交互式调试程序则可让您测试和调试应用程序。

如果您在 Streams Studio 中单击 Streams 运算符，将会打开针对这个特定运算符的 SPL 编辑器，其中包含各种语法和语义相关项，这可简化您在逐步执行开发流程中的编码任务。最后，还有一个集成帮助引擎，让您不费吹灰之力即可

完成 Streams 应用程序开发、调试和部署。总而言之，Streams Studio 可提供您期望从多功能应用程序集成开发环境（IDE，完整 Streams 平台的一部分，而不只是其他供应商提供的并行执行平台）中获得的易用性。

集成性是企业级分析的巅峰

企业级解决方案的最后一个评估标准是测试它与现有企业架构的集成密切程度。正如我们前文所说，大数据并不是要取代传统系统，只是为了增进传统系统。协调传统大数据进程和新时代大数据进程可促使供应商充分了解等式两边。阅读完本章内容后，您可能已经得出这样的结论：Streams 已将广泛的连接功能集成至企业资产，如关联数据库、内存中数据库、WebSphere 队列等。

在上一节中，我们对 Streams 基于 Eclipse 的 IDE 插件和监控基础架构进行了简要介绍，从而使您适应现有的应用程序开发环境，如基于广泛事实的标准开源 Eclipse 框架（IBM 发明并提供大量开放资源，我们可能会予以添加）的 Rational 或其他工具集。但这还只是一个开始：Streams 包含接收适配器，可通过高速并行加载程序将流数据并入 BigInsights Hadoop 环境，从而极其迅速地将流数据交付至 BigInsights（通过面向 Streams 的 BigInsights 工具包）或直接进入数据仓库进行静止数据分析。

我们在整本书中一直在强调，大数据问题需要对静止数据和移动数据进行分析，Streams 与 BigInsights 集成提供了实时数据分析及海量静止数据分析（以便完成复杂的分析工作）平台（而不只是产品）。IBM 可帮助您实现两全其美的效果，利用一个平台全面实现了安全性、企业服务级别协议预期、产品和支持渠道同化、企业性能预期等优势。

其他技术资源

依靠我们提供的大量 IBM 专家、计划和服务，帮助您将大数据技术更上一层楼。请通过 BigInsights wiki 参与我们的在线社区。查阅白皮书、观看视频和演示、下载 BigInsights、链接 Twitter、博客和 Facebook 网站，获取最新最全面的信息。

请访问 **ibm.com/**developerworks/wiki/biginsights

IBM 认证与资格考试

查找业界领先的专业认证和资格考试。我们现已推出 BigInsights (M97) 和 InfoSphere Streams (N08) 全新资格考试。

请访问 **ibm.com**/certify/mastery_tests

IBM 培训

探寻更加绿色和经济高效的在线学习、传统课堂、私人在线培训及世界一流的技术指导。我们经常以各种格式频繁添加新培训资料。

请访问 **ibm.com**/software/data/education 查阅所提供的教育课程。

- InfoSphere BigInsights 基础（使用 Apache Hadoop）
- BigInsights 分析基础 – 第 1 部分
- InfoSphere Streams 编程
- InfoSphere Streams 管理（第 2 版）

信息管理书店

查阅电子书籍、链接市场上最丰富的信息管理书籍，并提供有价值的链接和优惠，以节省资金、加强技能。

请访问 **ibm.com**/software/data/education/bookstore