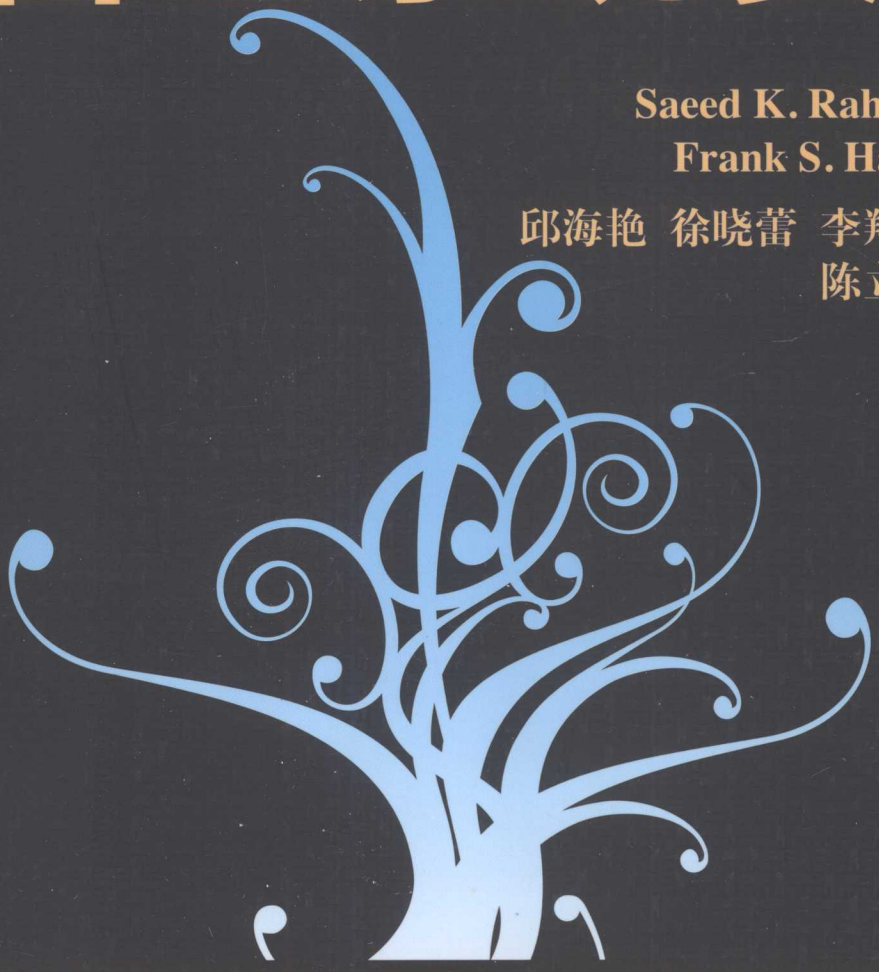
世界著名计算机教材精选

分布式数据库管理系统实践

Saeed K. Rahimi
Frank S. Haug 著

邱海艳 徐晓蕾 李翔鹰 等译
陈立军 主审



DISTRIBUTED DATABASE MANAGEMENT SYSTEMS

A PRACTICAL APPROACH

清华大学出版社

世界著名计算机教材精选

分布式数据库管理系统实践

Saeed K. Rahimi
Frank S. Haug 著
邱海艳 徐晓蕾 李翔鹰 等译

清华大学出版社
北京

Saeed K. Rahimi, Frank S. Haug
Distributed Database Management Systems: A Practical Approach
EISBN: 978-0-470-40745-5

北京市版权局著作权合同登记号　图字号 01-2012-6869

图书在版编目（CIP）数据

分布式数据库管理系统实践/（美）拉希米（Rahimi, S.K.），（美）豪格（Haug, F.S.）著；邱海艳等译. —北京：清华大学出版社，2014
书名原文：Distributed Database Management Systems: A Practical Approach
世界著名计算机教材精选
ISBN 978-7-302-33654-9

Ⅰ. ①分…　Ⅱ. ①拉…　②豪…　③邱…　Ⅲ. ①分布式数据库–数据管理–教材　Ⅳ. ①TP311.133.1

中国版本图书馆 CIP 数据核字（2013）第 204098 号

责任编辑：龙启铭
封面设计：常雪影
责任校对：李建庄
责任印制：杨　艳

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn, http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：36.25　　**字　　数**：905 千字
版　　次：2014 年 1 月第 1 版　　**印　　次**：2014 年 1 月第 1 次印刷
印　　数：1～2500
定　　价：79.00 元

产品编号：048037-01

译 者 序

分布式数据库系统是在集中式数据库系统的基础上发展起来的，是数据库技术和网络技术结合的产物。分布式数据库由一个统一的数据库管理系统，即分布式数据库管理系统。管理分布式数据库作为计算机专业及相关专业的核心课程，在国内外已经出版了大量有关的教科书，Saeed 和 Frank 编写的这本书，清晰地阐述了分布式数据库的基本理论和设计问题，涵盖了分布式数据库系统的设计、实现和管理等方面的诸多专题，内容充实、结构合理、表现形式多样，运用了丰富的讲授方法（如图表说明、程序示例等）。全书的理论体系完整，结构编排有序，其最大特点是理论和实践的完美结合，通过优秀的示例引入最新的分布式数据库技术，深入浅出地引领读者从理论到建模再到体系结构的实现，实用性很强，是高等院校计算机及相关专业本科生或研究生数据库系统课程的理想教材，对相关技术人员也是非常有价值的一本参考书，读者必将从中获益匪浅。

本书共分为 3 部分。第 1 部分（第 1～9 章）着重讨论分布式数据库理论。这些章节面向分布式数据库系统的多个侧面展开讨论，介绍可用于解决这些问题的多种技术和方法。第 2 部分（第 10～14 章）关注于分布式数据库的“实践的状态”。这一部分前面的章节概述一个常规的数据建模方法，并讨论了几个其他的数据建模方法。后面的章节关注于体系结构需求，并提出了几个不同的体系结构，包括传统的、自上而下设计的同构分布式数据库和自底向上设计的异构联邦式数据库，以及两个新的、非传统的体系结构方法，这两个新的方法注重具有更多动态部署特性的环境。第 3 部分（第 15～19 章）关注于分布式数据库的实现。这一部分试验了可用于分布式数据库开发的 3 种平台，这些平台包括 Java 消息服务（Java Message Service，JMS）、Java 2 企业版（Java 2 Enterprise Edition，J2EE）和 Microsoft .NET Framework。还详细研究了 3 种可选方案的实现，对这 3 种实现中的每一个都提供了详细的概览和可扩展的框架，称为入门者工具包（SKIT）。

在翻译的过程中，我们感到，作者在全书内容的组织上可谓用心良苦，3 部分 19 章的内容相对独立，又融会贯通，从理论到实践，使读者可以根据自身情况和不同需求，自由选取 3 部分的章节来学习，也能很好地服务于各种课程设置的需要。

参加本书翻译工作的有邱海艳（第 1～5 章）、徐晓蕾（第 6～11 章）、李翔鹰（第 12～19 章），全书由陈立军老师审校。由于时间仓促，译者水平有限，尽管我们注入了大量心血，但疏忽之处在所难免，恳请读者朋友提出建议和批评，以使得本书在再版时更为完美。联系信箱：hyqiu719@hotmail.com。

前　言

集中式**数据库管理系统**（Database Management System，DBMS）是复杂的软件程序，它允许企业在单机上控制数据。在过去的数十年中，由于一些组织机构拥有多个 DBMS，从而产生了很多合并和组合。这些系统来自不同的厂商。例如，通常会有这样的情况，企业有一个主体数据库，是由 IBM DB2 控制的，还有几个较小的工作组（或部门）数据库，是由 Oracle、Microsoft SQL Server、Sybase 或其他厂商控制的。大多数时间用户需要访问各自的工作组数据库。有时用户还需要访问某些其他工作组数据库，甚或是更大的、企业级数据库中的数据。这种数据共享的需求交叉分散于企业之中，而集中式数据库是不能满足这种需求的。为了满足这种需求，需要一种新的软件来管理分散的（或分布式的）数据，称这种软件为**分布式数据库管理系统**（Distributed Database Management System，DDBMS）。

DDBMS 维护和管理多个计算机上的数据。可以将 DDBMS 看作多个单独的 DBMS 的集合，每个 DBMS 运行在单独的计算机上，为了共享访问企业数据，可以利用某种通信工具来协同它们的行为。数据分散于不同的计算机中，且由不同的 DBMS 产品来控制，这个事实对于用户来说是完全隐藏的。这种系统的用户对数据的访问和使用，就好像数据是本地可用的，且由单一的 DBMS 控制。

本书主要讨论 DDBMS 的设计和开发的问题，并概述解决上述问题的不同方法。我们还会给出两个基于 Java 的框架和一个微软的基于.Net 的框架，它们提供了开发 DDBMS 所需的底层结构。

相关网站

本书的配套站点（www.computer.org/Rahimi_Haug）包含本书的重要信息，例如包括 3 个入门工具包（JMS-SKIT、J2EE-SKIT、DNET-SKIT）在内的入门工具包软件的链接，文字的错误修正以及源代码、本书内容或入门工具包的任何更新。

本书的组织结构

本书划分为 3 个不同的部分。第 1 部分着重讨论分布式数据库理论。这些章节面向分布式数据库系统的多个侧面展开研究，讨论了可用于解决这些问题的多种技术和方法。第 2 部分（第 10～14 章）关注于分布式数据库的“实际状态”。这一部分最初的章节概述了一个常规的数据建模方法，并讨论了其他几个数据建模方法。后面的章节关注于体系结构需求，并提出了几个体系结构，包括传统的、自上而下设计的同构分布式数据库、自底向上设计的异构联邦式数据库，以及两个新的、非传统体系结构的方法，这两个方法注重具有更多动态部署特性的环境。第 3 部分（第 15～19 章）关注于分布式数据库的实现。这一部分的章节试验了可用于分布式数据库开发的 3 种平台。还讨论了 3 个入门工具包，以及

创建它们的平台。这些平台包括 Java 消息服务（Java Message Service，JMS）、Java 2 企业版（Java 2 Enterprise Edition，J2EE）、Microsoft .NET Framework。

第 1 部分：理论

为了设计并最终创建 DDBMS，首先必须了解何为分布式数据库，DDBMS 的作用是什么，这些问题必须得到解决。这一部分首先对 DDBMS 及其所需的功能进行概述。然后详细研究每个主要子集。通过研究“大图片”和“小细节”，这一部分为理解 DDBMS 的内部处理打下理论基础。

第 2 部分：实践

由于大多数实践者都是从零开始，可能不会有设计分布式数据库环境（DDBE）的丰富经验，因此这一部分研究了可选的体系结构方法（和论题），它们是实践者在现实中很有可能会遇到的。首先，回顾了概念、逻辑和物理模型之间的不同。接着，将关注点放在几个逻辑数据建模方法（以及一些实际的物理考虑因素）上，这些方法是当今现实世界中“实际上”使用的。为了描述“实践的状态”，这一部分探索了 4 个（两个传统的、两个可选的）DDBE 体系结构，它们是设计用于现有 DBMS 实现环境的。接着，对于能实现 DDBE 可选体系结构的平台，研究了其一般需求。

第 3 部分：实现

由于体系结构是庞大的、复杂的设计，当试图创建实际的体系结构实现时，即使对“微小细节”的讨论，也好像“英里之上的地面”较之于“地面之下”一样，需要考虑的方面是不同的。软件开发上的最新进展使得实现复杂的、分布式的应用变得更容易。实际上，在平台、开发和部署细节的选择上，有很多可选的方案。这些可选方案都有很多支持分布式应用开发和部署的工具。对于实践者，为了精通任何特殊的可选方案，他们需要在时间、精力，甚至是金钱上花费巨大投资。因此，在现实世界中，虽然有很多可选方案，但是仍然会迫于强大压力而使用特定的一般方法，甚至是特定的专用方法。话虽如此，我们必须认识到，在 DDBE 的实现上没有可选的、特定的目标。这一部分试图减轻混淆，并提供一个具体的场所以开发实际的 DDBE。本部分还详细研究了 3 种可选的实现，对这 3 种实现中的每一个都提供了详细的概览和可扩展的框架，称为入门者工具包（SKIT）。我们对每个 SKIT 均进行了详细讨论。并给出了一个扩展的例子来示范 SKIT 如何可用于开始实现一个实际的 DDBE 项目。

本书的章节

本书由 19 章节组成。

第 1 章：概述

适合实现数据库管理系统（DBMS）体系结构的基本类型有两种。最著名的也是最普遍的体系结构类型是称为集中式 DBMS 的体系结构。它的定义非常合理和成熟，它是如此常见，以至于在使用术语“DBMS 体系结构”时，往往忽略名字中的“集中式”部分，而默认为集中式 DBMS 体系结构。另一种体系结构类型是不常见的，不成熟的，其定义也是

不完善的，称为分布式 DBMS（DDBMS）体系结构。在这个介绍性的章节中，我们对这两种体系结构均进行了描述，然后对它们进行了比较和对照。讨论了每个结构的强势和弱点，促使读者去了解哪种体系结构是最适合他/她的情况。本章还研究了分布式查询执行的不同实现方法，分析了主-从和三方分布式执行控制机制。对于每个执行方法，还使用两个例子来解释如何计算通信代价。

第 2 章：数据分布的方法

本章学习和比较分布式数据库管理系统中数据分布的不同设计方法。分析了可选的分片方法，主要是水平、垂直和混合分片。讨论了支持数据复制所产生的影响，并考虑了数据放置问题。还概述了这些方法的正确性条件，解释了 DDBMS 如何提供分片、位置和分布透明，并讨论了如何设计一个全局的数据字典来支持这些透明。我们给出了分布式设计的一些不同例子，并用它们阐述不同的数据分布透明对用户查询的影响。

第 3 章：数据库控制

本章重点讨论关于集中式和分布式数据库系统的安全性和语义完整性控制。当数据库面向并发事务时，如果没有控制，那么数据库的完整性就会受到危害。使用事务的最基本的原因之一是保持数据的完整性，因此，需要提出和讨论集中式和分布式的语义完整性问题。在实现分布式数据控制时，我们考虑如何平衡本地（集中式）数据库管理系统中的语义完整性控制机制，并讨论在分布式实现和本地强制实现中数据分布的影响。我们研究了 4 个可选的实现方法，它们用于强制实现语义完整性，评估了它们的优点和缺点，并提出了一个在分布式系统中强制语义完整性的代价分析机制，该机制关注于不同实现方法所需的消息数。

第 4 章：查询优化

在集中式系统中，查询优化不是基于规则的，就是基于代价的。除了集中式系统中的优化方法之外，查询优化器的主要目标是最小化查询的响应时间。查询优化的代价要素包括 CPU 时间和磁盘访问时间。从目前的发展来看，磁盘访问时间是查询代价中的主要因素。就此而言，大多数查询优化器注重最小化磁盘 I/O 数，从而最小化磁盘访问时间。本章将关注于分布式数据库管理系统中的查询处理。我们将集中式系统中查询处理的概念扩展为混合分布。在一个分布式数据库管理系统中，查询处理时间代价由本地 CPU 时间、查询中涉及的所有服务器的本地磁盘访问时间，以及同步这些服务器活动所需的通信代价组成。由于在分布式查询中通信代价是主要代价因素，因此分布式查询优化器应注重最小化该代价。我们讨论分布式查询的可选优化方法，这些方法着重减少在分布式系统中执行查询所需的消息数目。为了最小化通信代价，其中一个方法务必确保系统具有最优的分布设计，使得到不同数据库服务器的信息可以配置在合理的位置。已经证明如果将该问题涉及的所有因素都考虑进来，那么解决这个问题是一个 NP 难题。因此，解决该问题的目标不是最优方法，而是最接近最优的方法。

第 5 章：并发控制

本章涵盖了分布式数据库管理系统中的并发问题。首先定义了什么是事务，以及事务在数据库管理系统中必须满足的特性，然后回顾了集中式数据库管理系统中可选的并发控

制方法，并概述了两类并发控制算法（悲观算法和乐观算法）。我们分析了如何改变这些方法以适应分布式数据库管理系统的并发控制要求。由于每类算法都可用锁或时间戳来实现，因此对于基于锁和基于时间戳的并发控制算法的实现，我们评估了它们的优缺点。大多数并发控制方法使用锁作为提供隔离的主要机制。

第 6 章：死锁处理

当对系统中并发运行的事务使用锁时，可能会进入死锁状态。死锁被定义为这样一种系统状态，在该状态中，一个或多个事务陷入无限等待循环中。这意味着循环中的每个事务持有一个或多个锁，并等待得到当前被循环中其他事务之一持有的锁。由于每个事务都不会释放其持有的锁，所有事务将进入无限等待。本章讨论集中式系统中处理死锁问题的 3 种经典方法：死锁预防、死锁避免和死锁检测，如何将这些方法用于分布式数据库管理系统中，以及每种方法的优点和缺点。本章介绍了在死锁预防中使用预先获取所有锁的方法，讨论了等待-死亡和受伤-等待算法如何用于死锁避免。还给出了集中式和分布式死锁检测和消除方法。

第 7 章：复制控制

复制表或分片可以改善数据的可用性、可靠性和可恢复性（与第 2 章所讨论的一样），可以极大地改善系统的性能（特别是第 4 章讨论的查询优化）。不幸的是，关于复制的环境也有很多复杂的限制问题。本章讨论分布式数据库管理系统中复制的作用。重点讨论复制控制，该技术用于确保所有复制的表和表分片相互之间的一致性和同步。复制控制专门针对分布式数据库管理系统的问题。在 DDBMS 环境中有很多可选的复制控制方法，我们将其归类为异步或同步复制控制算法。我们讨论的复制控制算法有无争议的（Unanimous）、主副本的（Primary-Copy）、分布式的（Distributed）、基于投票的（Voting-Based）、基于法定人数的（Quorum-Based）和基于令牌传递的（Token-Passing-Based）复制控制算法。

第 8 章：故障和提交协议

DBMS 必须保证即使当系统出现故障时，能保持数据库内容的一致性。这一章讨论 DBMS 中对事务故障、电源故障和计算机硬件故障的容错能力。研究 DDBMS 用于提供此类保证的提交协议，该协议讨论 DDBMS 如何确定一个特定的事务是否可以安全提交。还研究了当事务不安全时，DBMS 容错机制如何取消事务，以及当事务的所有修改已经成功写入日志时，系统如何提交事务。在集中式系统中，日志管理器负责与事务管理器一起工作以正确实现提交协议。另一方面，在分布式系统中，多个 DBMS 服务器需要协同它们的活动以确保全局事务可以在发生过改变的所有 DBMS 站点上成功提交或取消。因此，我们讨论 DDBMS 如何实现分布式提交协议，研究诸如两阶段和三阶段提交协议这样的可选的分布式提交协议。网络分区是分布式系统中最困难的一种故障（一组计算机通过网络连接在一起，但它们与另一组内部相连的计算机是分隔的）。当发生网络分区时，则每个分区中的计算机可以单方面决定提交或取消事务，这样是危险的。如果两个分区不能对相同的行为做出决定（要么都提交事务，要么都取消事务），那么将损失数据库的完整性。因此，我们还研究了一个基于法定人数的提交协议，用它来处理网络分区。

第 9 章：DDBE 安全

这一章讨论安全性问题，包括改善数据在**分布式数据库环境**（DDBE）中私密性和安全性的认证、授权和编程技术。由于通信是数据库服务器在 DDBE 中协同操作的基础，应该确保服务器的通信和用户对数据的访问是安全的，因此，为研究私钥和公钥安全方法，我们讨论**安全套接层**（Secure Sockets Layer，SSL）和**传输层安全**（Transport Layer Security，TLS），并解释在分布式数据库系统中，许可证书如何用于组件和应用（包括基于 Web 的应用）。我们还考虑“隧道”方法，包括**虚拟专用网**（Virtual Private Network，VPN）和**安全外壳**（Secure Shell，SSH），讨论它们在分布式数据库系统中的潜在功能。

第 10 章：数据建模概述

这一章概述数据建模的概念和技术，研究数据建模并讨论它的目标。考虑数据模型的不同创建和分类方法以及数据建模语言，接下来关注于概念数据建模，并给出了创建概念数据模型和图的 3 种不同的语言。详细探讨了实体关系建模以及其他一些概念建模技术。简要讨论了逻辑数据建模以及它的目标是如何不同于概念数据建模的。类似地，还讨论了物理数据建模以及它的目标是如何不同于概念和逻辑数据建模的，然后简要介绍了一些命名法和符号，它们用于记录这些不同类型的数据模型。最后，研究这些不同类型的数据建模在异构数据库环境中是如何共存的。

第 11 章：逻辑数据模型

这一章研究现实世界中用于数据库的四种逻辑数据建模语言。关系数据模型是本书介绍的第一个建模语言，因为它是很多集中式 DBMS 的基础。它也是最有可能用于 DDBE 的逻辑建模语言，我们将详细研究这种语言。层次数据模型和网状数据模型是可以在一些历史遗留的数据库系统中，尤其是大型机中找到的两种逻辑建模语言。因此，也要了解这些语言。面向对象的程序设计（OOP）已经成为很多组织机构事实上的标准。虽然 OOP 应用可以访问存储为任意格式的数据，但是最自然的格式可以在面向对象的 DBMS（OODBMS）中找到。这种格式也可以在所谓的对象持久性引擎中找到。面向对象数据库使用面向对象建模语言，它与本章中讨论的其他建模语言在本质上是不同的。然而，这些语言之间存在某些相似的地方，我们将对其进行讨论。对于每种建模语言，介绍符号和命名法，以及将概念数据模型转换为使用这些逻辑建模语言之一创建的数据模型时必须遵循的规则。我们简要比较和对比了这 4 种语言，并考虑了在向前和向后推动使用这些语言的数据模型时值得一提的问题。

第 12 章：传统 DDBE 体系架构

在现实世界中，可以使用几种可选的不同的体系结构方法代替传统的、同构的、自顶向下设计的 DDBMS。联邦式数据库或多数据库是将现有的数据库服务器集成起来，它们是自底向上创建的。在联邦式数据库管理系统中，组织机构需要的数据已经存在于多个数据库服务器上。这些数据库也许不是相同类型的数据库（相同数据模型/数据建模语言）。这种异构的数据库集合需要集成起来为联邦式数据库系统的用户提供统一和相关的接口。这种集成涉及的问题是数据模型到关系数据模型的转换，以及将一组关系数据模型集成到一个联邦式视图中。

第 13 章：新 DDBE 体系架构

传统上，DDBE 中的一个组件或子系统不是一个独立的过程。这一章探索分布式数据库环境的体系结构。传统体系结构中的大多数组件都是由一个或多个子系统控制的，而大多数子系统是用一个“指挥系统”设计自身，定义其内部的组件。因此，我们给出的第一个新的体系结构是这样的，它的组件和子系统相互之间以协同方式的关系工作，而不是在相互控制下工作。另一种在大多数体系结构中可以找到的传统性特征是这样的事实，即 DDBE 体系结构中的子系统不是“生而平等的”。体系结构中的某些子系统被设计得比其他子系统更重要、更权威、更强大和更复杂。这也表明即使开发和部署环境提供动态部署和配置服务，体系结构也是略微有些刚性的。因此，这一章还给出了另一种新的体系结构，其子系统是基本对等的。这也使得环境可以更加动态，而对等子系统可以相对容易和灵活地加入和离开环境。

第 14 章：DDBE 平台需求

分布式数据库环境（DDBE）运行在计算机网络之上。这些计算机中的每一个都有自己的操作系统或其他系统，诸如本地 DBMS，需要将这些系统集成进 DDBE 体系结构中。尽管每个单独的计算机有其自身的操作系统，但是完全地分布式操作系统没有多大意义。这意味着我们必须在现有软件之上编写一个软件层，部署在每个单独的计算机上。称这个软件层为 DDBE 平台。可以以多种不同方法构建（和实现）该层。相对于关注实现细节，这一章更侧重于体系结构问题和通用 DDBE 平台需求，特别是我们的 DDBE 平台必须如何处理通信、组件命名、体系安全、部署、分布式事务，以及一般的便携性/互用性问题。这一章还简要探索了可以为 DDBE 平台提供的一些（或许全部）需求的某些实现。我们研究用于远程调用（Remote Call）的 RPC 和 RMI 机制、用于远程消息（Remote Messaging）的 Java 消息服务（Java Message Service，JMS）实现。我们还给出了 XML Web Service 的一个高级概述，这是以交叉平台方式实现服务提供者的一种相对新的技术。

第 15 章：JMS 开发工具包

这一章考虑如何开发分布式数据库管理系统 DDBE，它可用于集成 Oracle、Microsoft、IBM、Sybase 及其他数据库。我们提供一个 DDBE 开发工具包，作为开发我们的 DDBE 组件和子系统的起点，包括一个框架和一个扩展的例子。它使用 Java 2 标准版（J2SE）实现，通过 Java 消息服务（Java Message Service，JMS）的实现来增强。可以用这个框架作为一个基础，在没有额外复杂性（或能力）的完整 J2EE 应用服务器的情况下，使用 Java 来创建异构的分布式数据库管理系统，可以得到快速启动。讨论如何使用这个框架，为实现当前框架中没有提供的附加功能创建新的扩展。最后，我们给出了扩展的例子（用开发工具包编写的），它使用 3 个数据库，讨论了设置和运行这个例子所需的步骤。

第 16 章：J2EE 平台

这一章讨论 J2EE 平台，对平台进行简要概述，介绍不同 DDBE 平台服务的详细实现细节。特别明确了 J2EE 规范中定义的可满足第 14 章所明确需求的 J2EE 服务。还考虑了此时选择的不同实现，其中包括 Apache Geronimo 和 JBoss 的研究，它们研究了 Enterprise Java Beans（EJB）提供的远程支持能力，并讨论了其他 J2EE 工具如何可用于 DDBE 项目。

对工具的讨论包括 Java Naming 和 Directory Interface（JNDI）、XML Web Services 和 RMI，以及 J2EE 提供的安全、部署和事务管理工具。

第 17 章：J2EE 开发工具包

这里给出了另一个 DDBE 开发工具包，它类似于第 15 章给出的那个。但是，这个开发工具包是使用 J2EE 平台实现的，只需最小的修改就可以传送到几个可选的 J2EE 实现中。可以使用这个框架作为一个基础，在使用 J2EE 以及它提供的全功能子系统创建异构的分布式数据库管理系统时，可以得到快速启动。尤其是远程能力和分布式事务管理工具，使得这个实现比我们在第 15 章使用的 JMS/J2SE 方法更强大（也更复杂）。这个开发工具包中包含一个平台和一个扩展的例子。本章讨论如何使用这个框架，为实现当前框架中没有提供的附加功能创建新的扩展。最后，给出了一个简单的扩展的例子（用初学者工具包编写的），它使用 3 个数据库，我们讨论了它的实现。

第 18 章：微软.NET 平台

这一章将微软.Net 框架用作一个可能的 DDBE 开发平台，对该平台进行了简要的概览，介绍了在前面第 14 章中明确的各种 DDBE 平台需求的详细实现细节。特别研究了用于远程代码执行（Remote-Code Execution）的 TCP/IP Remoting 和 HTTP Remoting 实现，还探索了用于 XML Web Services 的通用消息工具和支持。讨论了微软.Net 基于框架的 DDBE 平台如何提供目录服务（Directory Services），展示并研究了该平台是如何支持安全、部署和事务管理的。

第 19 章：DNET 开发工具包

这一章给出了另一个 DDBE 开发工具包，与第 17 章介绍的类似。然而，这个开发工具包是使用微软.Net 框架实现的。因此，我们可以使用这个框架作为一个基础，在使用微软.Net 框架支持的任何编程语言（Visual C#、Visual C++或 Visual Basic）创建我们的异构的分布式数据库管理系统时，可以得到快速启动。这个开发工具包扭转了工具在微软.Net 基于框架的平台中的可用性，我们讨论第 18 章中指定的工具如何用于构建我们的 DDBE 项目。这个开发工具包中包含一个框架和一个扩展的例子，我们讨论如何使用这个框架，为实现当前框架中没有提供的附加功能创建新的扩展。最后，给出了一个简单的扩展的例子（用初学者工具包编写的），它使用 3 个数据库，我们讨论了它的设计。

本书的有效使用

下面描述的是贯穿全书的不同学习路径或“轨迹”：

- 当本书用于课堂环境时，推荐使用第 1 部分的全部内容，并按其顺序介绍。根据课程的范围，从第 2 部分选择包含的内容也应该基于课程中讨论的异构程度。这条轨迹应该最有可能包含第 1 部分的所有内容（第 1～9 章），以及至少第 10 章和第 11 章的部分内容。
- 如果课程中包含任何关于实现的活动或思考，那么同样我们推荐将第 1 部分包含进来，在这种情况下，第 10 章、第 11 章和第 14 章也是很有助益的。根据选择的实

现平台，这条轨迹还应包括第 3 部分的适当章节。换言之，它应包括 J2EE 的章节（第 16 章和第 17 章）、Microsoft .Net Framework-based 的章节（第 18 章和第 19 章），或包含 JMS 的章节（第 15 章）。

- 如果读者计划设计自己的实际的 DDBMS 体系结构，第 1 章和第 2 部分中的章节（第 10～14 章）提供了理解系统体系结构需求的必要背景知识。同样，根据读者实现时选择的开发平台，可以选择包含第 3 部分的适当章节。
- 如果读者使用现有的 DDBMS 体系结构，那么可以将第 1 部分作为其实现中组件和子系统背后的理论支撑参考。第 2 部分可以用于理解体系结构问题或是有助于思考其体系结构的改变。可以根据现有的开发环境，或是在可选的开发环境中的实现原型来选择第 3 部分。

致谢

特别致谢：Brad Rubin 博士，圣托马斯大学软件程序设计专业，对安全方面的章节做出了贡献；Dyanne Haug 和 Patricia Rahimi 为本书初稿提供了很有价值的反馈和建议：非常感激占用了他们的时间和精力；IBM 公司的 Khalid Albarrak 博士，帮助我们学习、安装和调试 InfoSphere Federation Server 软件，并对我们的测试设置和测试结果提供了反馈。圣托马斯大学的 Saladin Cerimagic，软件程序设计专业，感谢他对并发讨论的帮助。

SAEED K. RAHIMI
FRANK S. HAUG

目　录