
PRENTICE
HALL
The Data Access Handbook: Achieving Optimal
Database Application Performance and Scalability
数据访问宝典
——实现最优性能及可伸缩性
的数据库应用程序
(美) John Goodson
Robert A. Steward
王德才
著
译
PEARSON
Education
清华大学出版社

The Data Access Handbook：Achieving Optimal Database Application Performance and Scalability

数据访问宝典

——实现最优性能及可伸缩性的数据库应用程序

清华大学出版社

北　京

北京市版权局著作权合同登记号 图字：01-2009-5136

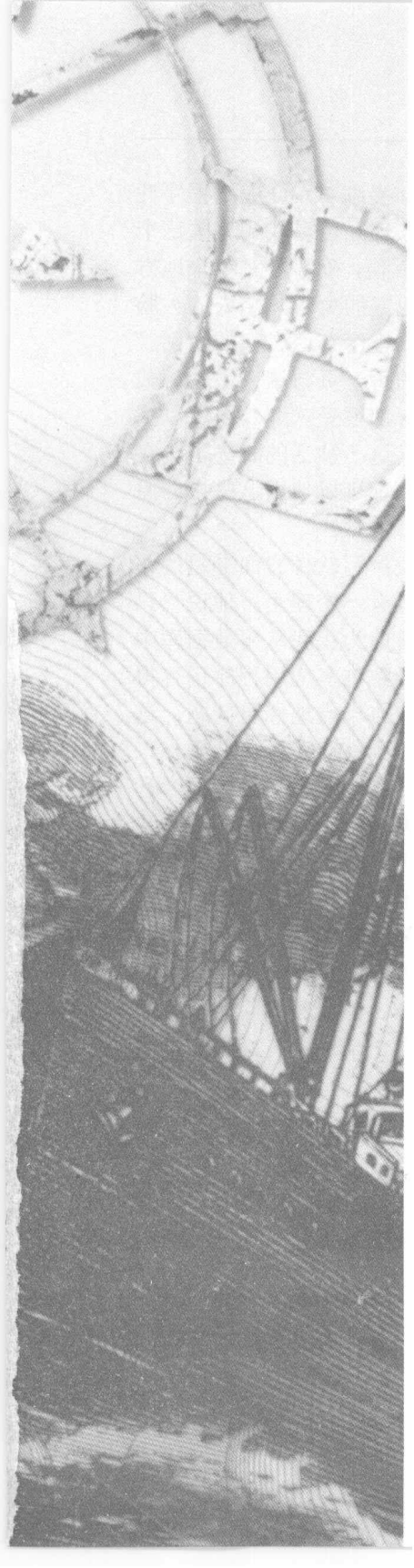
图书在版编目(CIP)数据

数据访问宝典——实现最优性能及可伸缩性的数据库应用程序/(美)古德森(Goodson,J.)，
(美)斯图亚特 (Steward,R.A.) 著；王德才 译. —北京：清华大学出版社，2010.3
书名原文：The Data Access Handbook：Achieving Optimal Database Application Performance and Scalability
ISBN 978-7-302-22071-8
Ⅰ. 数… Ⅱ. ①古… ②斯… ③王… Ⅲ. 数据库系统－程序设计 Ⅳ. TP311.13

中国版本图书馆 CIP 数据核字(2010)第 021438 号

责任编辑：郑雪梅
装帧设计：康 博
责任校对：胡雁翎
责任印制：杨 艳
出版发行：清华大学出版社 **地 址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn **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：北京市清华园胶印厂
经 销：全国新华书店
开 本：185×230 **印 张**：17 **字 数**：350 千字
版 次：2010 年 3 月第 1 版 **印 次**：2010 年 3 月第1 次印刷
印 数：1～4000
定 价：39.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770177 转 3103 产品编号：033098-01



前　言

地球是平的。几千年来，这个世界上的所有数学家、勘探者以及哲学家都确信这是正确的。在6世纪，几个希腊哲学家提出了证据，证明地球是圆的。然而，专家们回避他们的这一思想长达几百年之久。

如果咨询数据库专家，他们会告诉您：所有数据库应用程序的性能和可伸缩性问题，都可以通过调校(tuning)数据库加以解决。他们甚至会说服您每年花费数千到数百万美元，调校数据库软件以解决性能问题。当调校不能解决问题时，他们会告诉您数据库软件或硬件、或者两者的能力不够。但是，如果在调校良好的数据库软件中，实际上只有5%~25%的时间用于处理数据库请求，那么认为这些“系统”功能不足是性能的瓶颈，这种观点合理吗？如果一个商业分析人员为一个查询处理等待10秒，而数据库只使用了其中的半秒，那么花费大量的时间和金钱解决如何提高这半秒的性能合理，还是试图解决如何提高另外9.5秒的性能更加合理呢？绝大部分书籍、咨询报告、Web站点都致力于解决数据库调校问题，但是关于如何设计以数据为中心的应用程序，如何调校数据访问应用程序代码，如何选择和调校数据库驱动程序，以及如何调校在数据库应用程序之间和数据库应用程序与数据库之间的数据流这方面的信息相对很少。我们撰写本书的目的就是为了提供这方面的信息，帮助减少那9.5秒的时间，并展示如何解决那些通过调校数据库所不能解决的数据库应用程序的性能问题和可伸缩性问题。

本书最初由John和Rob分别撰写，最后汇总在一起。

John Goodson：几年以前，我在 IBM 大会上作了一个关于如何提高 JDBC 应用程序性能的报告。作完报告之后，一位 IT 主管找到我，并向我咨询："从哪儿可以找到有关这个主题的更多信息？这类信息是不可能找到的。"我思考了一会，告诉他，确实没有哪个地方可以找到这方面的信息——这方面的信息存在于这个世界上不同地方的少数几个人的大脑中。之后，许多其他 IT 主管也告诉我，"我从来都不知道数据库驱动程序有这么大的区别。"或者"我一直认为数据库应用程序的性能是数据库的问题。"我写的每一页与这个主题相关的技术资料都有很大的需求，并且我做的关于这个主题的每个报告都很受欢迎。之所以撰写本书，是因为现在是消除"性能和可伸缩性仅仅是数据库问题"这一神话的时候了。在本书中提供了许多用于提高性能的指导原则、提示和技巧，所有人都可以使用这些指导原则、提示和技巧。

Rob Steward：我以为我永远不会撰写书籍。我曾经询问过一位撰写软件方面书籍的作者朋友，撰写书籍是否值得。他向我强调，撰写软件方面的书籍只有一个理由。他说"只有当你强烈地感觉到需要说一些问题的时候才会撰写一本书。"到目前为止我从事数据库中间件业务有 15 年了，我曾经看到过许多比较差的数据访问代码，导致许多应用程序运行得非常缓慢，并且必须进行修补。我曾经花费几年的时间帮助他人一点一点地修复问题，这些问题是因为他们缺少有关"当向数据库提交一个调用时在客户端到底会发生什么操作"这方面的知识。John 和我曾经在许多会议上讨论过这个主题，并且撰写了许多技术文章和白皮书，以帮助尽可能多的开发人员理解数据访问代码的错综复杂的关系。当 John 找到我商量共同撰写本书时，我立刻同意了。我强烈地感觉到，应当在更大的范围内分享那些在过去几年里我们曾经在各种论坛上共享的每一部分知识。我衷心地希望每个读者，通过本书都能够找到所需要的知识，从而使将来开发的应用程序可以显著地提高运行速度。

作者希望软件架构设计人员、IT 人员、数据库管理员以及开发人员每天在预测、诊断以及解决数据库应用程序中的性能问题时，能够用到本书中的内容。调校数据库对于获得良好的性能和可伸缩性是很重要的。我们知道这是事实。然而，在一个调校良好的数据库系统中，大部分性能问题是由以下原因造成的：

- 设计不良的数据访问架构
- 没有很好地进行优化的数据访问代码
- 效率低下和协调不良的数据库驱动程序
- 对部署数据库应用程序的环境缺乏理解

本书将解决所有这些问题——地球是圆的。

本书包含以下章节：

第 1 章："性能问题与以前不同了"，描述了数据库中间件的演变，以及识别在哪些地方可能会出现性能瓶颈。

第 2 章：“提高性能的设计策略”，为设计数据库应用程序以及调校连接应用程序与数据库服务器的数据库中间件以优化性能提供指导原则。

第 3 章：“为什么数据库中间件很重要”，解释了什么是数据库中间件，以及它是如何影响性能的。还描述了应当在数据库驱动程序中查找哪些性能调校选项，数据库驱动程序是最重要的数据库中间件组件。

第 4 章：“为提高性能而调校环境”，描述了数据请求和响应经过的不同环境层，解释了环境是如何影响性能的，并且提供了相应的指导原则，以确保环境不会成为性能瓶颈。

第 5 章“ODBC 应用程序：编写良好的代码”，描述了一些良好的编码实践，这些编码实践可以优化 ODBC 应用程序的性能。

第 6 章“JDBC 应用程序：编写良好的代码”，描述了一些良好的编码实践，这些编码实践可以优化 JDBC 应用程序的性能。

第 7 章“.NET 应用程序：编写良好的代码”，描述了一些良好的编码实践，这些编码实践可以优化.NET 应用程序的性能。

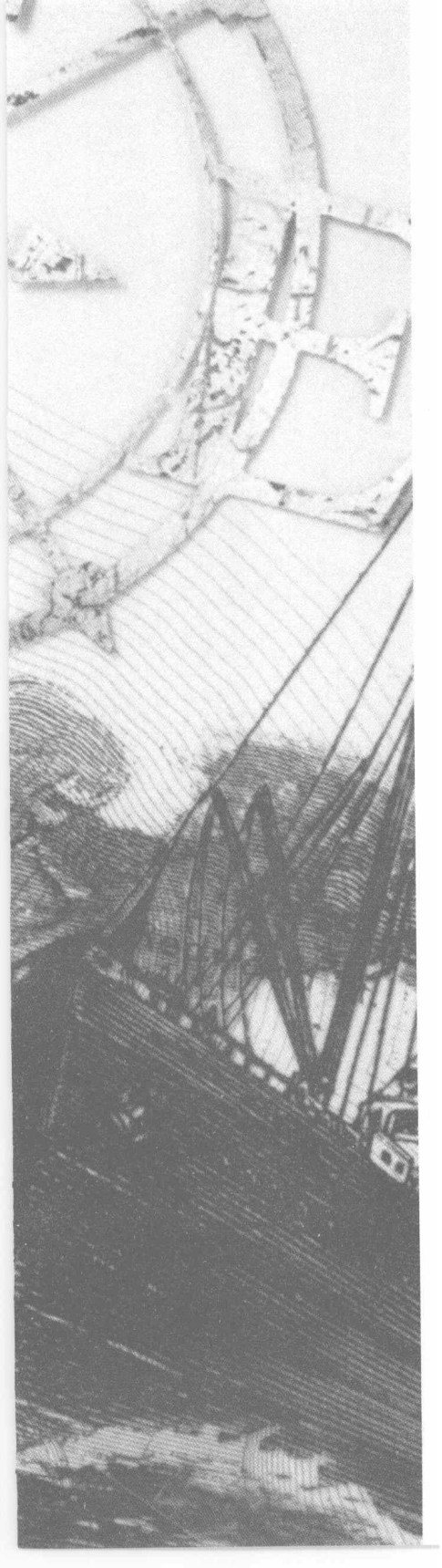
第 8 章：“连接池和语句池”，提供了不同连接池模型的详细内容，描述了如何对连接池进行重新认证工作，以及如何联合使用语句池和连接池，它们可能会消耗比看起来更多的数据库服务器内存。

第 9 章：“开发良好的基准”，为编写基准提供了一些基本的指导原则，虽然许多开发人员不遵循这些原则，但是绝对有必要遵循这些原则。

第 10 章：“性能问题调试”，完整介绍了如何调试性能问题，并且提供了几个案例研究，以帮助分析某些会变化的性能问题，并弄清如何解决这些问题。

第 11 章：“面向服务架构(SOA)环境中的数据访问”，提供了一些通用的指导原则，以确保数据库应用程序在面向服务架构(SOA)的环境中能够良好地执行。

附录“术语表”定义了在本书中使用的术语。



致　　谢

本书是在许多人的贡献、知识以及支持的基础上，历经两年完成的。John Goodson 和 Rob Steward 从大学毕业后，曾经从事过数据库中间件方面的研究。Cheryl Conrad 和 Susan King 出色地将我们的思想变成了文字。Progress Software 公司给予了我们写作本书的机会。

许多人提供了大量的详细信息，我们在整本书中都用到了这些信息，为此，我们表示衷心的感谢。特别是，要感谢 Scott Bradley、John Hobson、Jeff Leinbach、Connie Childrey、Steve Veum、Mike Spinak、Matt Domencic、Jesse Davis 以及 Marc Van Cappellen，感谢他们对本书做出的贡献，这些贡献花费了他们很多时间。没有非常周密的审阅者和其他撰稿者的帮助，是不可能完成一本书的。他们是 Lance Anderson、Ed Long、Howard Fosdick、Allen Dooley、Terry Mason、Mark Biamonte、Mike Johnson、Charles Gold、Royce Willmschen、April Harned、John Thompson、Charlie Leagra、Sue Purkis、Katherine Spinak、Dipak Patel、John De Longa、Greg Stasko、Phil Prudich、Jayakhanna Pasimuthu、Brian Henning、Sven Cuypers、Filip Filliaert、Hans De Smet、Gregg Willhoit、Bill Fahey、Chris Walker、Betsy Kent 以及 Ed Crabtree。

我们还要特别感谢我们的家人、朋友，他们一直支持我们完成这一历险。

如果读者发现了错误或者有什么建议，可以通过 e-mail 发送给我们：Performance-book@datadirect.com，或者 wkservice@vip.163.com，我们将非常感谢。

关 于 作 者

John Goodson：作为 DataDirect Technologies 的执行主管，John 负责日常工作、业务开发、产品研发以及长期企业战略。

John 作为首席工程师在 Data General 工作了 7 年，开发他们的关系数据库产品 DG/SQL。自从 1992 年加入 DataDirect Technologies，他一直负责研发、技术支持和销售。John 是一位非常著名的并且受人尊重的业界杰出人物和数据连接专家。15 年来，他曾经和 Sun Microsystems 公司、Microsoft 公司在开发和发展数据库连接标准方面，包括 J2EE、JDBC、.NET、ODBC 以及 ADO，进行过密切合作。John 曾经参与 ANSI NCITS H2 协会，该协会负责构建 SQL 标准，John 还参与了 X/Open(Open Group) SQL 访问小组(SQL Access Group)，该小组负责为关系数据库构建调用级别的接口。他还积极参与 Java 标准协会，包括 JDBC 专家组。此外，John 发表了大量文章，针对与数据管理相关的主题进行了大量的公开演讲。John 还是 Microsoft SQL Server Java 中间件分布式事务领域的一位专利拥有者。

John 获弗吉尼亚理工学院及州立大学计算机科学学士学位。该学校位于弗吉尼亚洲黑堡(Blacksburg)镇。

Rob Steward：作为 DataDirect Technologies 研发副总裁，Rob 负责公司数据连接产品的开发、策略以及监督，这些产品包括 Shadow 大型主机集成系统客户端软件和在工业上领先的数据库驱动程序与数据提供程序 DataDirect Connect 系列：ODBC 连接、JDBC 连接以及 ADO.NET 连接。他还负责 DataDirect Sequelink 和 DataDirect XQuery 产品的开发，并负责管理 DataDirect 的客户工程开发(Custom Engineering Development)小组。

Rob 花费了超过 15 年的时间开发数据库访问中间件，包括.NET 数据提供程序、ODBC 驱动程序、JDBC 驱动程序以及 OLE DB 数据提供程序。他在 DataDirect Technologies 承担了大量的管理和技术岗位，并且是多种标准协会的成员。在他职业的早期，作为主要软件工程师在 Marconi Commerce Systems 公司工作。

Rob 获北卡罗来纳州立大学计算机科学学士学位，该学校位于北卡罗来纳州首府罗利(Raleigh)。

目　录

第 1 章

性能问题与以前不同了

您公司的一个或多个数据库应用程序正受到性能问题的困扰。这几乎不会使那些应用程序设计人员、开发人员和部署人员感到惊奇。而让人感到惊奇的是，这些问题中的许多问题的根本原因是数据库中间件，数据库中间件是连接应用程序和数据库的软件。

当我们说到性能问题时，是指应用程序正受到不能让人接受的响应时间、吞吐量或者可伸缩性的困扰。响应时间是指在数据请求和数据返回这个过程中经历的时间。从用户的角度来说，响应时间是用户请求数据和他们获得数据这个过程中经历的时间。吞吐量是指在一段时间内从发送程序向接收程序传输的数据的数量。可伸缩性是指当同时访问数据库的用户数量增加时，应用程序维持可以接受的响应时间和吞吐量的能力。

在分析为什么数据库中间件对于优化数据库应用程序的性能是如此重要之前，首先回顾一下在过去 10 年中数据库应用程序的性能情况。

10~15 年之前，如果数据库应用程序存在性能问题，95%的时间问题是由数据库管理软件造成的。在那时，除了少数工程师和数据库专家之外(他们很有可能直接为数据库厂商工作)，对于其他所有人来说调校数据库被认为是一种魔术。

当这些专家开始通过撰写有关数据库调校方面的书籍，并举办公众讲座共享他们的知识时，性能问题开始发生变化了。现在，因为可以通过书店或者互联网获得大量的数据库调校信息、多年的经验以及大量改进了的数据库监视工具，数据库调校工作变得不再那么让人痛苦了。

在这些年里，其他方面正在发生变化，例如硬件价格下降并且计算机的能力上升。随着硬件变得更快并且更便宜，应用程序运行环境从单机转移到网络环境和客户/服务器计算(两层和三层环境)。现在，大多数数据库应用程序通过网络和数据库进行通信，而不是直接通过单个计算机中的内部进程进行通信，如图 1-1 所示。

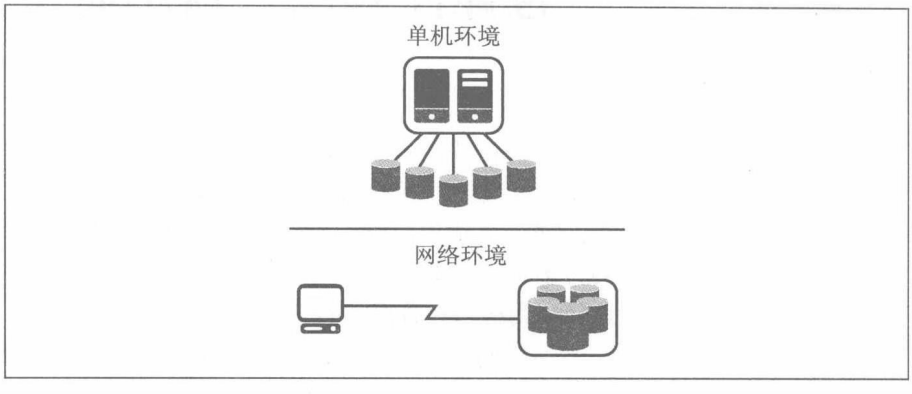


图 1-1　数据库环境

转移到网络环境上之后，软件需要提供在应用程序和数据库之间的连接，现在这些连接位于不同的计算机中。数据库厂商首先提供了这种软件，作为其数据库专门的数据库中间件组件，这为性能问题添加了新的因素。

数据库中间件由所有处理应用程序数据请求的组件构成，直到这些请求被传递给数据库管理软件，如图 1-2 所示。

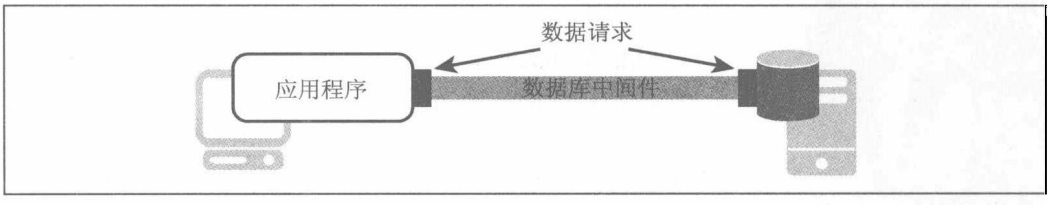


图 1-2　数据库中间件

随着网络环境的引入，数据库中间件层包括以下组件：

- 网络
- 数据库客户端软件，例如 Oracle Net8
- 当应用程序连接到数据库时加载到应用程序地址空间的库(libraries)，例如用于对通过网络传输的数据进行加密的 SSL 库。

很快，在工业上开始出现了对数据库连接标准的需求，数据库连接标准可以为访问多个数据库提供通用的应用程序编程接口(API)。如果没有通用 API，将面临多个数据库连接解决方案；每个数据库厂商提供自己的专用 API。例如，为了开发能够访问 Microsoft SQL Server、Oracle 和 IBM DB2 的应用程序，开发人员必须知道三种完全不同的数据库 API：Microsoft Database Library(DBLIB)、Oracle Call Interface(OCI)和 IBM Client Application Enabler(CAE)，如图 1-3 所示。数据库连接标准的出现，例如 ODBC，解决了这个问题。

随着数据库连接标准的出现，数据库驱动程序被添加到数据库中间件层。和许多其他内容一起，数据库驱动程序处理基于标准的 API 函数调用，向数据库提交 SQL 请求，并向应用程序返回结果。有关数据库驱动程序工作的详细信息，以及选择的驱动程序如何影响数据库应用程序性能的详细信息，请阅读第 3 章。

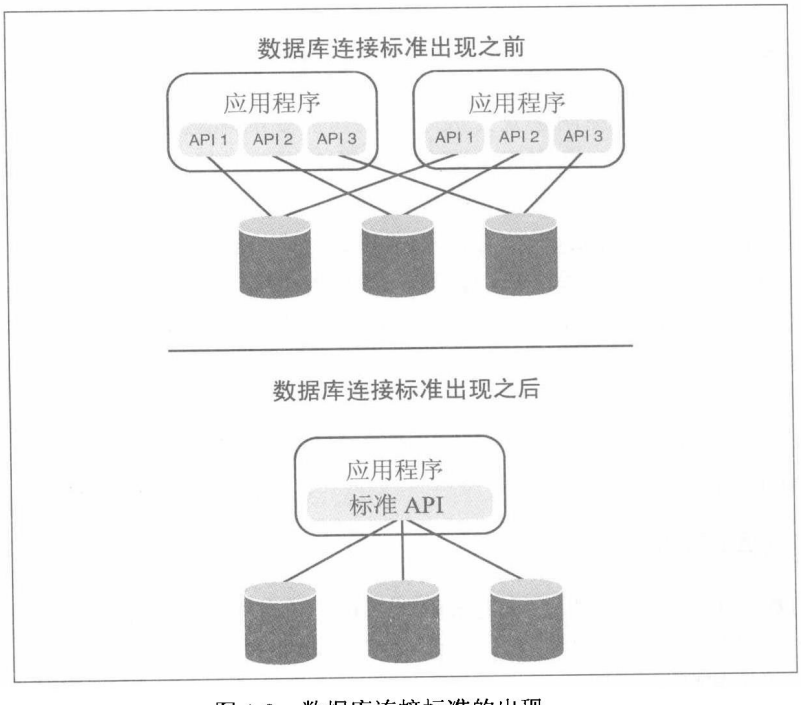


图 1-3　数据库连接标准的出现

1.1　现在的情况如何

现在，我们知道，即使数据库调校得很好，数据库应用程序的运行情况也并不总是尽

如人意。问题出在哪儿呢？现在我们能从哪儿发现性能问题呢？对于多数数据库应用程序，性能问题现在出现在数据库中间件上。与 10~15 年之前相比，现在大多数应用程序，处理数据请求的时间中的 75%~95%花费在数据库中间件上，而在 10~15 年之前，大部分时间花费在数据库管理软件上。

在大多数情况下，数据库应用程序中的性能问题是由以下原因造成的：

- 网络
- 数据库驱动程序
- 环境
- 代码编写不良的数据库应用程序

本书将深入分析数据库中间件和数据库应用程序的性能问题。现在，让我们先看几个例子。

1.1.1 网络

关于网络最常见的性能问题是完成一个操作需要的往返次数。网络包的容量与需要的往返次数有关。网络包通过数据库中间件向数据库传递应用程序的消息，反过来，网络包通过数据库中间件向应用程序传递数据库的消息。网络包的容量会影响数据库应用程序的性能。需要记住的主要概念是，在应用程序和数据库之间发送的数据包越少，数据库应用程序的性能越好；发送的数据包越少，意味着在数据库和应用程序之间的往返次数越少。

思考这样一个过程：Jim 的经理要求他从二楼的办公室向一楼的厨房搬送 5 箱汽水。如果 Jim 每次搬送 6 包而不是 1 箱，他需要往返 20 次，而不是 5 次，这意味着向厨房搬送汽水他将需要更多的时间。

在第 4 章中，我们将讨论关于网络的更多内容，它们影响性能的原理，以及如何才能够优化网络的性能。

1.1.2 数据库驱动程序

所有的数据库驱动程序都不是生来就平等的。在数据库应用程序部署中，选择使用哪个驱动程序，对性能有很大的影响。下面的真实情况说明了一个公司如何仅通过改变数据库驱动程序解决其性能问题。

DataBank 通过检索、存储以及传送数据，为大公司和小公司提供信息需求服务。DataBank 的声誉和财政安全依赖于系统的响应时间和可用性。DataBank 和它的客户之间有关于特定响应时间和系统可用性需求的协议。如果不能满足这些需求，DataBank 必须支付罚金给客户。