

<<数据访问宝典>>

图书基本信息

书名：<<数据访问宝典>>

13位ISBN编号：9787302220718

10位ISBN编号：7302220719

出版时间：2010-3

出版时间：古德森(John Goodson)、斯图亚特(Robert A. Steward)、王德才 清华大学出版社 (2010-03出版)

作者：(美)古德森 等著,王德才 译

页数：256

译者：王德才

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

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。

<<数据访问宝典>>

内容概要

在当今的企业数据库应用程序中，性能和可伸缩性比过去任何时候更为关键，传统的数据库调整对于解决可能在这些应用程序中遇到的性能问题有些力不从心。

如今，用于处理数据请求的时间通常有75%-95%花费在数据库中间件上。

现在，最严重的性能和可伸缩性问题通常由与网络、数据库驱动程序、宽泛的软件/硬件环境，以及效率低下的数据请求编码等原因造成。

在本书中，数据库访问领域中的两位世界级顶尖专家系统地解决了这些问题，揭示了如何显著地改善实际数据库应用程序的性能。

利用每个先进的数据库系统以及数据库连接API方面的领先经验，John Goodson和Rob Steward揭示了中间件影响应用程序性能的方式，并指导开发人员设计和编写在每个主要环境中能够实现较好性能的代码。

此外还介绍了可以跨数据库系统和API采用的基本概念和技术，并提供了针对ODBC、JDBC和ADO .NET的API示例，以及针对DB2、Microsoft SQL Server、MySQL、Oracle和Sybase等主流数据库系统的示例。

本书包括以下内容：清晰理解数据库中间件的每个组件影响性能和可伸缩性的方式 编写数据库应用程序，减少网络流量、限制磁盘I/O、优化应用程序和驱动程序之间的交互并简化查询——包括针对ODBC、JDBC和ADO .NET的示例更加高效地管理连接、事务以及SQL语句的执行 充分利用连接池和语句池 编写良好的基准测试程序，预测应用程序的性能 系统地解决性能问题——包括8个完整的案例研究示例 如果您是一位软件架构师、系统设计人员或者数据库应用程序开发人员，本书将是您最必不可少的数据库应用程序性能资源。

本书专注于如何实现最大的性能提升——无论是设计新的数据库应用程序还是排除已有数据库应用程序的故障。

作者简介

John Goodson是Progress Software公司DataDirect部门的副总裁和总经理，是数据连接和主机集成方面的权威。

20年来，他与Sun公司、Microsoft公司和其他公司合作开发了数据库连接标准，如J2EE、JDBC、ODBC和ADO。

他是构建了SQL标准的ANSI H2协会的成员，并且现在加入了JDBC专家组和Java Rowsets标准协会。

书籍目录

第1章 性能问题与以前不同了 1.1 现在的, 隋况如何 1.2 本书的目标 第2章 提高性能的设计策略 2.1 应用程序 2.2 静态SQL与动态SQL 2.3 网络 2.4 数据库驱动程序 2.5 理解数据库系统 2.6 使用对象/关系映射工具 2.7 小结 第3章 为什么数据库中间件很重要 3.1 数据库中间件是什么 3.2 数据库中间件影响应用程序性能的原理 3.3 数据库驱动程序 3.4 小结 第4章 为提高性能而调校环境 4.1 运行时环境(Java与.NET) 4.2 操作系统 4.3 网络 4.4 硬件 4.5 小结 第5章 ODBC应用程序:编写良好的代码 5.1 管理连接 5.2 管理事务 5.3 执行SQL语句 5.4 检索数据 5.5 更新数据 5.6 使用目录函数 5.7 小结 第6章 JDBC应用程序:编写良好的代码 6.1 管理连接 6.2 管理事务 6.3 执行SQL语句 6.4 检索数据 6.5 更新数据 6.6 使用数据库元数据方法 6.7 小结 第7章 .NET应用程序:编写良好的代码 7.1 管理连接 7.2 管理事务 7.3 执行SQL语句 7.4 选择.NET对象与方法 7.5 检索数据 7.6 更新数据 7.7 小结 第8章 连接池和语句池 8.1 JDBC连接池模型 8.2 ODBC连接池模型 8.3 ADO.NET连接池模型 8.4 为连接池使用重新认证 8.5 使用语句池 8.6 小结:整体考虑 第9章 开发良好的基准 9.1 开发基准 9.2 基准实例 9.3 小结 第10章 性能问题调试 10.1 从何处开始 10.2 数据库应用程序部署中的改变 10.3 数据库应用程序 10.4 数据库驱动程序 10.5 环境 10.6 案例研究 10.7 小结 第11章 面向服务架构(SOA)环境中的数据访问 11.1 面向服务的架构(SOA)是什么 11.2 SOA环境中数据访问的指导原则 11.3 小结 附录 术语表

章节摘录

插图：3.使用连接池连接池是应用程序能够重复使用的一个或多个数据库物理连接的高速缓存。连接池能够显著提高性能，因为重复使用连接，减少了建立物理连接所需要的相关资源开销。

需要注意的是，为了管理连接池中的所有连接，数据库服务器必须具有足够的内存。

在本书中，我们讨论位于客户端方的连接池（由数据库驱动程序和数据库服务器提供的连接池），而不讨论位于数据库方的连接池（由数据库管理系统提供的连接池）。

一些数据库管理系统提供了连接池，并且这些连接池实现和客户端方的连接池一起工作。

尽管数据库方连接池的具体特征是不同的，但是总的目标都是消除建立和删除连接在数据库服务器上所需要开销。

和客户端方的连接池不同，数据库方的连接池没有优化从数据库到应用程序的网络往返过程。

正如前面所说的，因为需要在数据库驱动程序上分配资源（在驱动程序和数据库之间的网络往返），并且也需要在数据库服务器上分配资源，所以建立到数据库的连接对性能的影响很大。

客户端方的连接池帮助数据库驱动程序以及数据库服务器解决昂贵的资源分配问题。

数据库方的连接池只帮助解决数据库服务器上的问题。

连接池的工作原理在使用连接池的环境中，一旦建立了最初的物理连接，在连接池的生存期中，就很可能不会被关闭。

也就是说，当应用程序断开连接时，物理连接并没有关闭；反而，它位于连接池中以备再次使用。

因此，重新建立连接就变成了一个最快的操作，而不是最慢的操作。

下面是连接池工作过程的基本描述（如图2-1所示）：（1）当应用程序或应用程序服务器启动时，连接池通常包含连接。

（2）应用程序提出连接请求。

（3）驱动程序或者连接池管理器（根据您的架构）为应用程序分配一个连接池中的连接，而不是请求建立一个新的连接。

这意味着没有为连接请求进行网络往返，因为可以在连接池中得到连接。

结果是：应用程序的请求很快。

（4）应用程序连接到数据库。

（5）当关闭连接时，连接被放入连接池中。

编辑推荐

《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》是由清华大学出版社出版的。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>