

基于 JavaBean和连接池技术的数据库连接优化方案

A Solution of Optimizing Database Connection Based on JavaBean and Connection Pool Technology

李立锋, 归伟夏, 杨文敏

Li Lifeng, Gui Weixia, Yang Wenmin

(广西大学计算机与电子信息学院, 广西南宁 530004)

(School of Comp., Elec. and Info., Guangxi Univ., Nanning, Guangxi, 530004, China)

摘要: 简单介绍 J2EE数据库连接技术中的 JavaBean技术和连接池技术, 分析它们的优点和缺点, 给出一个利用 JavaBean技术和连接池技术优化数据库连接的解决方案。

关键词: 数据库 连接技术 JavaBean 连接池

中图分类号: TP311.13 文献标识码: A 文章编号: 1002-7378(2005)S0-0107-02

Abstract This paper describes two kinds of methods of database connection technology of J2EE (JavaBean technology and connection-pool technology), analyses their advantage and disadvantage. Finally, we integrate the two technologies to design a solution of optimizing database connection.

Key words database, connection technology, JavaBean, connection pool

J2EE技术是 Sun公司为企业开发推出的一个基于 JAVA语言的服务器应用结构, 支持平台独立、可移植、多用户、安全和标准的企业级应用^[1]。J2EE利用 JAVA2平台来简化诸多与多级企业解决方案的开发、部署和管理相关的复杂问题, 支持提供数据库存取、Internet应用中的数据安全模式、API、CORBA企业 JavaBean规范、JavaServlets API、JSP以及 XML技术的全面支持^[2]。

在构建企业应用信息系统中, 数据库的实现是关键所在。为了实现应用信息系统的数据库访问, 需要有应用服务器与数据库管理系统 (DBMS)之间的接口, 即 J2EE的数据库连接技术。本文简单介绍 J2EE数据库连接技术中的 JavaBean技术和连接池技术, 分析它们的优点和缺点, 提出一个利用 JavaBean技术和连接池技术优化数据库连接的解决方案。

1 优化的相关技术

1.1 JavaBean技术^[3]

JavaBean是一种组件技术, 它的工作原理是将数据的基本处理逻辑封装在 JavaBean包中, 然后通

过一些接口实现 JavaBean之间的相互调用, 它的最大优点是可以实现代码的重复使用。

在基于 J2EE的应用系统中, 通过把进行数据库连接操作的代码封装在 JavaBean中, 可以提高代码的重复使用程度, 大大简化程序设计的复杂度, 提高开发效率。当更换后台数据库时, 不需要对每个用到数据库连接的地方进行修改, 只需在 JavaBean中改变驱动程序以及连接方法即可, 这样就能大大方便程序的维护^[4]。

但是, JavaBean数据库连接技术在建立与数据库的连接时需要耗费较多的时间, 而且数据库所能支持的并发连接数量有限, 过多的并发连接将导致数据库运行效率的下降。随着客户访问数据库的数量不断增加, 数据库服务器的负担也会越来越重, 严重时将导致系统瘫痪。

1.2 连接池技术^[5]

数据库连接池技术是一种可靠、高效的数据库连接管理策略, 它的基本原理是预先在一个连接池中建立好一些连接, 当客户端发来的连接请求时, 可直接从连接池中取出一个空闲的连接, 直接分配给客户端 (而不是花时间去创建并初始化一个连接)。当客户端完成数据库操作时, 便释放连接 (而不是关闭一个连接)。连接池负责回收释放的连接, 供下次复用, 如图 1所示。这样, 我们就不需要频繁的建立、

关闭连接,节省了系统的开销,提高了操作效率。

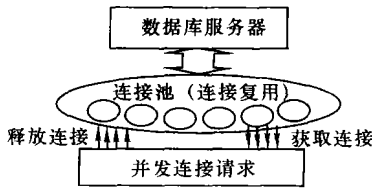


图 1 连接池原理

2 数据库连接的优化方案

2.1 JavaBean与数据库连接池的方案

程序中建立的一般是静态连接池。所谓静态连接池,是指连接池在系统初始化时就已经分配好的,而且不能随意关闭连接。连接池建立时,可通过 Java 中提供的 Vector 容器类一次性地构建相应配置的连接池,装载预设数目的连接对象。这些连接对象作为系统可自由分配的资源,供客户端请求时直接使用,避免了随意建立、释放连接所带来的系统资源的浪费。

我们在 createPool() 中首先保证连接池没有被创建,然后注册 JDBC 驱动程序,创建保存连接的向量 connections,通过 createConnections (this.initialConnections) 建立连接并添加到连接池里。其中 initialConnections 是连接池的初始大小。

连接池建立后,就要对连接池进行管。连接池管理是连接池机制的核心,对系统的性能有很大的影响。连接池管理的策略是:当客户请求数据库连接时,首先看连接池中是否有空闲连接,如果存在空闲连接则把连接分配给客户,并作相应处理;若连接池中并没有空闲连接,则查看当前所开的连接数是否已达到最大连接数 maxConnections,如果没达到就重新创建一个连接给请求的客户,如果到达就按设定的最大等待时间 mSeconds (单位 ms) 进行等待,如果等待 mSeconds 后仍没有空闲连接,就抛出无空闲连接的例外给用户。当客户释放数据库连接时,先判断该连接的引用次数是否超过了规定值,如果超过就删除该连接,并判断当前连接池内总的连接数是否小于 minConnections,若小于就将连接池充满;如果没超过就将该连接标记为开放状态,可供再次复用。这套策略保证了数据库连接的有效复用。如连接服务有更大的通用性,则要考虑多线程环境,即并发问题。在一个多线程环境中,一个线程可能比另一个线程有更高的优先级,必须要保护一个单独的线程不被其他线程过早地中断,即要保证连接管理自身数据的一致性和连接内部数据的一致性。在 Java

中可以使用 synchronized 关键字获得这种保护,当 synchronized 关键字加在方法的特征定义中后(定义一个同步方法),JavaVM 对于该类的实例化的每个实例关联一个锁。当线程在一个特定的实例调用一个同步的方法时,其他线程将不能访问该方法或者该类定义的其他同步方法,这样就可以使连接管理成为线程安全的。如图 2 所示。

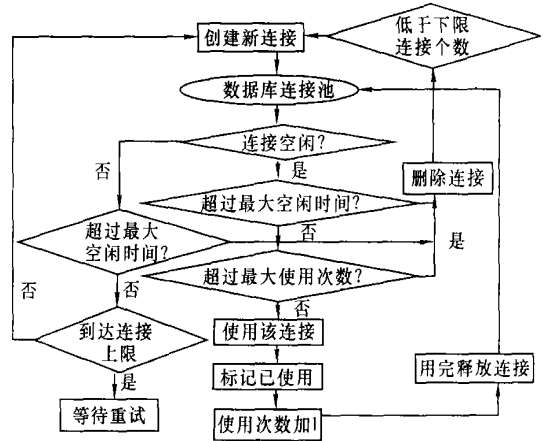


图 2 管理策略原理

当应用程序退出时,应关闭连接池,此时应把在连接池建立时向数据库申请的连接象统一归还给数据库(即关闭所有数据库连接)。这与连接池的建立正好是一个相反过程。

2.2 优化方案

对以上给出的 JavaBean 与数据库连接池的解决方案的性能作进一步的改进和优化。

在 Java Bean 中,用预编译语句

PreparedStatement 取代原来的 Statement,这样可以提高 SQL 语句的执行效率。即将原代码:

“Statement stmt = connect.createStatement();”
改为:

```
PreparedStatement stmt = connect.  
prepareStatement("SQL语句");
```

如果简单地采用上述的连接复用的策略,就会发生问题,因为没有办法控制属于同一个事务的多个数据库操作的动作,可能这些数据库操作是在多个连接上进行的。Connection 本身提供了对于事务的支持,可以通过将 Connection 的 AutoCommit 属性设置为 false,显式地调用 commit 或者 rollback 方法来实现。这种方法可以大大降低事务处理的复杂性。

连接管理服务提供了显式的事务开始、结束的

(下转第 111 页)

行时间和 CPU 占用率两项的优化效果比较明显, SQL 语句执行时间减少 26s, CPU 占用率也减少了 6%, 其他三项参数也都有一定的提高。比较结果表明, Oracle9i 数据库性能优化的效果较好。

4 结束语

优化数据库整体的应用性能是提高计算机系统处理速度的一种行之有效的办法, 本文从对 SGA (系统全局区)、磁盘 I/O 情况以及 SQL 语句的优化等方面的优化进行讨论, 结果表明优化后的数据库性能有所提高, 但在实际应用中要根据系统的实际情况具体分析, 对优化的方法反复实验, 最后再确定

最终的优化方法, 这样才能达到较好的优化效果

参考文献:

- [1] 佚名. Oracle 数据库性能优化技术 [EB/OL]. <http://www.ddvip.net/database/oracle/index4/34.htm>, 2005-03-18.
- [2] 王秋生. 基于 PL/SQL 的 Oracle 数据库性能优化 [J]. 微机发展, 2003, 13: 47-52.
- [3] 薛铭. Oracle 性能调整及优化技术 [J]. 长春工程学院学报, 2002, 3: 70-72.

(责任编辑: 黎贞崇)

(上接第 108 页)

声明, 以及一个事务注册表, 用于登记事务发起者和事务使用的连接的对应关系。通过该表, 可以将使用事务的部分和连接管理部分隔离开来, 因为该表是在运行时根据实际的调用情况动态生成的, 事务使用的连接在该事务运行中不能被复用。当使用者需要使用事务方法时, 首先调用连接管理服务提供的 beginTrans 方法, 该方法主要代码如下:

```
public void beginTrans( ) {
    ...
    conn = getFreeConnection( ); //从连接池中获取一个空闲连接
    userId = getUserId( ); //获取用户标识, 它是唯一的
    registerTrans( userId, conn); //将用户标识 ID 和其获取的连接对应地放入注册表中
}
```

在实现过程中, 用户标识用使用者所在的线程号来标识。所有对数据库的访问都通过查找该注册表和使用已分配的连接来完成。当事务结束时, 从注册表中删除相应表项。

我们把连接池模块与执行数据库各种 SQL 操作的模块分别封装在 DBPool.java 和 DBsql.java 中, 以进一步提高代码的重用性与可扩展性, 有利于进一步开发。

3 结束语

本文介绍了数据库连接的 Java Bean 实现方法。在 Java Bean 中我们用一般的数据库连接方法实现了 SQL 的数据库连接, 实现了代码重用, 提高了开发效率。为了使系统有高效、可靠的数据库连接, 我们在 Java Bean 的基础上引入了连接池技术。通过高效的连接池管理策略, 使数据库连接效率和安全性得到了极大提升, 从而为企业应用系统的开发提供一个高效、方便、可靠、安全的数据库连接解决方案。

参考文献:

- [1] 张晓东. JAV A 数据库高级教程 [M]. 北京: 清华大学出版社, 2004.
- [2] 王强兵, 刘广钟. 基于 J2EE 的 Web 企业计算 [J]. 计算机工程, 2002, 28(1): 262-264.
- [3] 马廷淮, 赵亚伟, 刘忠. 用 EJB 开发 J2EE 应用 [J]. 计算机应用, 2002, 22(4): 111-113.
- [4] Art Taylor [美]. JDBC 数据库编程与 J2EE [M]. 李东升译. 北京: 电子工业出版社, 2004.
- [5] 崔莹峰, 王洪. 数据库连接池技术及其在 web 系统开发中的应用 [J]. 铁路计算机应用, 2005, 14(2): 48-50.

(责任编辑: 邓大玉)