

基于 JDBC 数据库连接池的自适应管理策略研究*

马海燕^{1,2}, 彭宇行²

(1. 台州学院 信息与电子工程学院, 浙江 临海 317000; 2. 国防科学技术大学 计算机学院, 湖南 长沙 410073)

摘要: 通过分析现有数据库连接池在参数配置、管理策略等方面的不足, 提出一种自适应的连接池管理策略。从配置参数调整、设置监听线程和运行日志等管理方案的实施, 弥补了现有连接池技术的不足。运行测试证明, 自适应管理策略能减少系统响应时间, 降低系统资源的开销, 从而提高系统的整体性能。

关键词: 数据库连接池; 自适应; XML; HPF 策略

中图分类号: TP3 文献标识码: A 文章编号: 1001-3695(2006)02-0057-03

Research of Self-Adaptive Management Strategy of Database Connection Pool Based on JDBC

MA Hai-yan^{1,2}, PENG Yu-xing²

(1. School of Information & Electronical Engineering, Taizhou University, Linhai Zhejiang 317000, China; 2. School of Computer Science, National University of Defense Technology, Changsha Hunan 410073, China)

Abstract: Through analyzing the disadvantages of the current database connection pool on the aspects of parameter configuration and management strategy, this paper presents a self-adaptive management strategy of connection pool. It applies configuration parameter, setting up the thread of monitoring and running log of management scheme etc, to solve the inefficiency of current connection pool. Through operations and tests, it proves that system response duration is shorter and expense on system resources is lower by applying this scheme, thus the performance of system is improved.

Key words: Database Connection Pool; Self-adaptive; XML; HPF Strategy

Java 语言的跨平台性、可移植性以及安全性等特性使其成为开发数据库的一种优秀语言。JDBC 是 Java 应用程序与数据库沟通的桥梁^[1]。一般情况下, Java 应用程序访问数据库的基本步骤为: 在主程序中通过 JDBC 建立数据库连接; 进行 SQL 操作, 访问数据库, 存入、取出数据; 断开数据库连接。由于每次 Web 请求, 建立连接要花费 0.5s ~ 1s 时间, 连接使用后必须关闭, 这样频繁的建立与关闭连接, 极大地降低系统性能。在实际开发中, 普遍使用连接池技术将数据库的访问过程管理起来, 以提高系统的效率和稳定性。

1 问题的提出

1.1 数据库连接池组成

连接池技术是将一定数量的数据库连接作为对象存储在 Vector 对象中, 形成一个储存数据库连接的容器, 内部提供一种管理机制控制数据库连接的建立、分配、断开等操作。连接池分为线程池、连接池、数据库操作三大部分^[6,7]。线程池统一对要执行的任务进行合理的线程分配和调度, 执行用户的任务, 管理计算机的线程资源。连接池负责管理数据库连接的建立、释放和调度。连接池中先建立若干个连接, 放置在内存对象中, 当有数据库访问请求时, 只需从连接池的空闲队列中取用连接, 数据库访问完成后, 将连接放回连接池中, 供其他数据

库操作时复用这些连接。数据库操作负责 SQL 语句的执行、结果的返回, 保证事务的完整性和异常处理。

1.2 连接池的调度管理策略

1.2.1 连接池基本属性

连接池的相关属性和参数配置是连接池的重要组成部分, 可以建立一张 XML 的配置表进行参数配置。主要包括: 连接池的名字, 用于标志应用使用的是哪一个连接池; 初始连接数; 连接池的最大、最小容量; 数据库的位置 URL; 驱动程序类的名字; 驱动程序特定的属性。如 BEA 公司的 WebLogic Server 的 Config.xml^[2] 参数配置表形式为 < JDBC ConnectionPoolName = "MMDB" URL = "jdbc:rawstrin:sqlserver://localhost:1433" DriverName = "com.rawstrin.jdbc.sqlserver.SQLServerDriver" Properties = "databasename = MMdb; user = scott; password = tiger" InitialCapacity = "2" MaxCapacity = "8" CapacityIncrement = "2" LoginDelaySeconds = "2" RefreshMinutes = "10" PreparedStatementCacheSize = "4" Targets = "myserver" TestTable Name = "emp_info" / > (其中 1433 为 SQL Server 的端口号)。基于 J2EE 架构的应用程序服务器, 如 IBM 公司的 Websphere、金蝶公司的 Apusic 中大都采纳了连接池机制, 其配置参数与此大同小异。

1.2.2 连接池调度策略

图 1 给出了现有连接池的调度策略流程^[3,5]。

1.3 现有 JDBC 连接池调度策略的不足

通过以上对数据连接池原理的分析, 发现现有连接池技术

的参数配置及管理存在以下问题：连接池的参数配置策略大多是静态的，依据 Config.xml 参数配置表设置池参数，这些参数不能根据应用需求的变化而适时地调整数据库连接池的容量。在系统运行时，客户请求数量的多少往往表现出类似于波峰和波谷的曲线，所以我们在系统允许的范围内，提供对连接池的自适应动态管理。而现有连接池运行中的资源调度管理策略还不能适应应用的动态需求。缺少对连接池的用户管理机制。现有连接池可以完成用户验证、安全上下文配置等任务，但是如何保证连接池资源不释放给未授权用户使用？如何让关键性事务获得优先服务？现有数据库连接池无法解决。

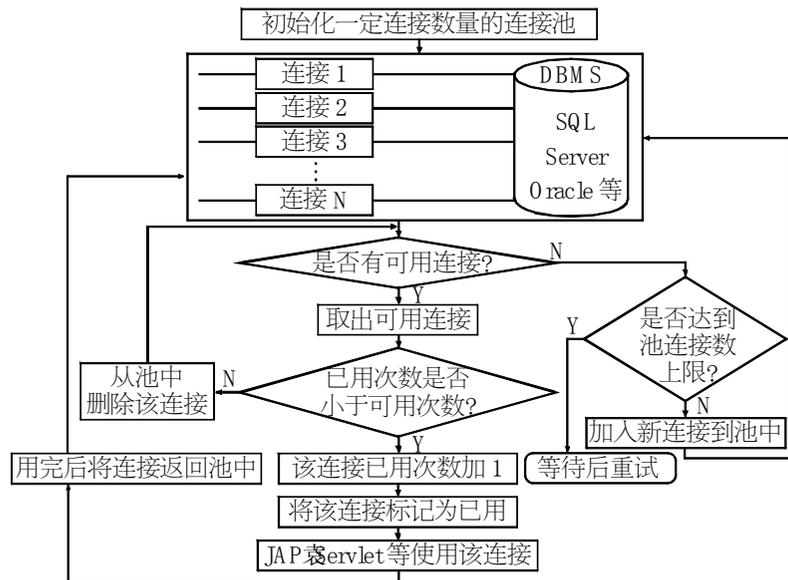


图 1 现有连接池调度策略

针对以上问题，本文提出了数据库连接池的自适应管理方案，其基本思路是：采用基于 XML 配置表，使系统可以根据用户的不同应用需求来配置连接池参数；设置运行日志，在运行过程中能够根据日志来自适应地调整连接池配置参数；连接池调度算法采用类似工作集的策略，根据过去对资源的使用情况决定连接池的大小以及连接池中各连接的优先级；对用户实行优先级管理，高优先级的用户可以剥夺那些低优先级且可被剥夺的资源，尽可能保证高优先级的用户享受优先服务；对安全性要求较高的连接进行验证使用，由用户在配置中指定。通过这一系列的调整策略的实施，可以建立一个高效率、实用的自适应数据库连接池。

2 数据库连接池的自适应管理策略

2.1 连接池参数配置改进

自适应连接池的配置参数包括总的配置参数和连接池参数^[4]。配置文件 Config XML 的主要内容描述如下：

```

<自适应数据库连接池>
  <属性>
    <初始容量>、<最大容量>等是现有池中的参数。根据自适应管理策略的需要增加相应参数：
    <状态>0</状态>
    <调整周期>5</调整周期>
    <最大内存利用率>0.9</最大内存利用率>
    <最小内存利用率>0.5</最小内存利用率>
    <最近一次调整时间>20041118192832</最近一次调整时间>
    <时间频度因子>0.75</时间频度因子>
  </属性>
  <连接>
    <连接 JNDI 名>、<数据库地址>、<数据库驱动程序>、<数据

```

库名>、<用户名>、<密码>等是现有连接池通用参数。我们增加了以下参数：

```

<身份认证> * * * * <身份认证>
<优先级>0.04, 0, ..., 0.07 </优先级>
<权限>1 </权限>
<日志回写时间>b20041018192832e20041018192834...
</日志回写时间>
</连接>
</自适应数据库连接池>

```

其中总连接池各属性参数的作用是：初始容量、最大容量设定连接初始运行容量，实际运行中连接池的大小由工作集策略进行调整；池状态，0 有空余连接，1 有可剥夺连接，2 无可用连接；调整周期，连接池进行一次运行策略调整的时间间隔，以分钟为单位；日志回写时间，系统将阶段运行日志评价结果回写到磁盘的时间段，并清除日志；最大内存利用率，此参数的作用是当进行连接池调度时发现内存的利用率超过此界限则进行紧缩处理；最小内存利用率，此参数的作用是当进行连接池调度时发现内存的利用率还没超过此界限但是已达到最大允许连接数，则可以短期越界；最近一次调整时间；时间频度比因子，在调度策略中，是使用时间的长短重要，还是使用次数的频繁重要，影响调度结果。单连接池参数的作用是：通用参数，是传统的连接池的基本参数；身份认证；优先级值，用于记录评价结果，评价结果 = f(日志，时间频度比因子，...)，其值随系统运行而变化，反映连接池运行的最新动态；权限值，0 可被剥夺，1 可被高优先级剥夺，2 不能被剥夺。

2.2 连接池的自适应管理策略

2.2.1 连接池自适应调整的条件

连接池的调整发生在以下几种情况：连接池调整周期到，是连接池调整的例行工作。系统有警告时，如内存的最大利用率超过了连接池配置参数。发生新用户请求，而缓存没有空余资源，但有可剥夺连接且新请求的优先级高于此可被剥夺连接的优先级。连接池调整流程是：接收到池调整请求后，对其状态进行评价，然后对可连接资源进行排序，将最常使用的资源放在最前面，最容易被淘汰的资源总在最后面。

2.2.2 自适应调整的监视线程

不同时段用户的数据访问量决定数据存取具有波峰和波谷的曲线规则。在峰值请求时将连接池容量动态地放大，提供更多的可供使用的资源；波谷时收缩连接池，进行资源的回收，淘汰那些不再使用或最不可能在将来使用的资源。为连接池对象增加一个后台运行的监视线程^[3]，在池对象创建后，按照一定的算法进行检测连接池对象的运行状态，实现动态调整的功能。连接池调度线程代码段如下：

```

public class FactoryManageThread implements Runnable{
    ConnectionFactory cf = null;
    long delay = 1000;
    public FactoryManageThread( ConnectionFactory obj) {
        cf = obj; }
    public void run()
    {
        while( true) {
            try { Thread . sleep( delay) ;}
            Catch( InterruptedException e) {
                System . out . println( " eeeee" );}
            if( cf . isCreate() ) //判断是否已经关闭了工厂，退出监听
                cf . schedule() ;
            else

```

```

System . exit ( 1 ) ;
} } }

```

2.2.3 日志管理

连接池的动态自适应性主要依据运行日志来决定管理策略的调整。根据用户经常访问的数据主要集中在某些部分数据的规则^[8], 连接池将数据访问的情况都记录在日志中, 这样根据日志运行的结果就可能找到大部分频繁使用的数据。为避免连接池调度频繁地从磁盘上存取数据而消耗额外的系统资源, 这里采用将日志记录在缓存中, 而只在一定的时候将日志评价的结果回写到连接池的配置文件中, 并清除日志。连接池首次启动是根据用户配置参数来运作的, 以后每次启动的时候装入上一次日志运行的结果来动态决定池参数的值, 从而有效地避免了用户盲目设置连接池参数。

2.2.4 用户请求管理

对用户请求实行最高优先级 HPF (Highest Priority First) 法, 具体实现采用剥夺式动态设置优先级的 HPF 算法, 系统响应用户请求连接的算法流程见图 2。其调度性能明显优于非剥夺式静态^[8]设置优先级的 HPF 调度。

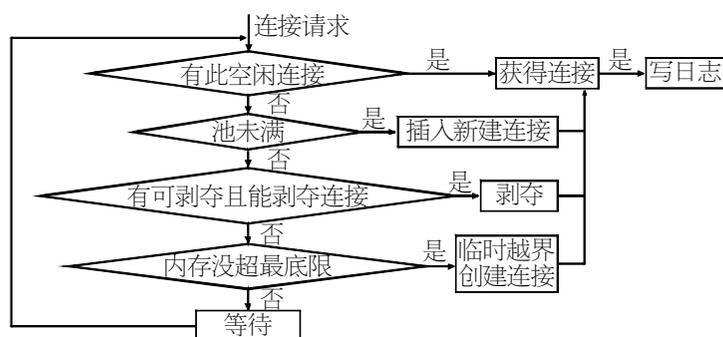


图 2 HPF 算法的用户请求连接过程

3 性能测试

为了与传统连接池比较, 将该文的自适应连接池部署到 WebLogic 服务器上, 同时在服务器还安装了基于 Web 的应用程序供测试使用。我们使用 VB 设计了一个测试 Web 访问速度的工具, 其作用是创建多线程访问指定的 URL, 并返回在指定访问量的情况下系统的响应时间。需指出的是, 实际测试的结果与服务器的硬件和软件配置有很大关系。本文测试结果为: 当服务器内存空余较多时, 传统连接池达到最大连接数就不再响应请求, 而该文的连接池在没超过内存底限时可继续工作; 对于频繁使用且已在连接池中的连接, 传统连接池访问 1 000 次需要 12 353ms, 自适应连接池需要 8 109ms; 对于不经常使用且已在连接池中的连接, 传统连接池访问 1 000 次需要 12 759ms, 自适应连接池需要 9 312ms; 不同时段并发用户访问响应时间对比见图 3。从以上三条测试结果表明自适

应连接池性能明显比传统连接池优越; 从图 3 的曲线对比表明连接池的自适应策略总体响应时间比现有连接池技术在大部分时段有明显的降低。

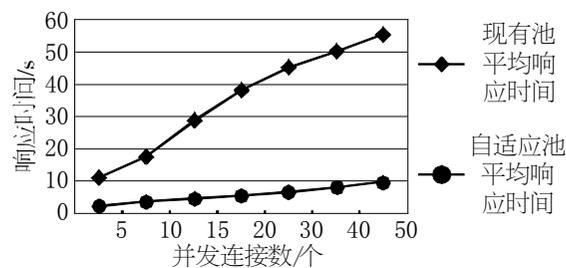


图 3 并发用户访问响应时间对比

4 结束语

连接池技术的关键在于对连接池自身的管理。对于大型应用系统而言, 采用传统连接池是远不能满足需求的。该文提出连接池的自适应管理方案在 TRMIS 教学资源管理系统中运行的结果显示它比传统连接池响应速度快, 更能满足大型应用的需求。同时, 该连接池能够很容易地进行多方面的扩充, 以适应具体系统需求。

参考文献:

- [1] 赵勇超, 郑宁, 葛瀛龙. Java 中连接池的设计与实现 [J]. 计算机应用研究, 2004, 21(6): 219-221, 224.
- [2] Michael Girdley, et al. J2EE 应用与 BEA WebLogic Server [M]. 邢国庆, 等. 北京: 电子工业出版社, 2002. 129.
- [3] 孔哲. 数据库连接策略优化方法 [J]. 山东大学学报 (工学版), 2003, 33(6): 654-657.
- [4] 朱长生. 自适应数据库连接池的研究 [J]. 计算机工程与应用, 2003, (36): 188-189.
- [5] 杨瑞. 数据库连接池原理及 Java 实现 [J]. 微型机与应用, 2003, (4): 12.
- [6] 田翔川, 田忠和, 谢志宇. JSP 数据库连接池的设计 [J]. 计算机应用研究, 2004, 21(2): 198-199.
- [7] Siva Visveswaran. Dive into Connection Pooling with J2EE [EB/OL]. <http://java.sun.com/developer/technicalArticles/J2EE/pooling/>, 2000-10.
- [8] Alka Gupta, Michael Doyle. Turbo-charging Java HotSpot Virtual Machine, v1.4. x to Improve the Performance and Scalability of Application Servers [EB/OL]. <http://java.sun.com/developer/technicalArticles/Programming/turbo/>, 2002-11.

作者简介:

马海燕 (1969-), 女, 江苏人, 讲师, 硕士, 主要研究方向为分布式应用系统开发、多媒体技术与网络教学; 彭宇行 (1963-), 男, 湖南人, 教授, 博士生导师, 博士, 主要从事分布式多媒体的研究。

(上接第 41 页)

- [9] 余金山. 软件开发过程及其模型 (II) [J]. 计算机应用与软件, 1995, 12(4): 1-8.
- [10] 覃成菊, 孙健敏. 科技人员流动分析 [J]. 科学学与科学技术管理, 2004, 25(4): 130-132.
- [11] Gerald M Weinberg. 质量·软件·管理——系统思维 [M]. 邓俊辉. 北京: 清华大学出版社, 2004. 313-334.
- [12] Barry W Boehm. 软件工程经济学 [M]. 李师贤, 等. 北京: 机械工业出版社, 2004. 145-147.
- [13] 张亚莉, 杨乃定. 人员流动风险分析与控制 [J]. 科学学与科学技术管理, 2000, (9): 42-44.

- [14] UMLChina, 潘加宇. 《人件》在计算机行业的实践 [EB/OL]. www.peoplewarecn.com/review/pwcompany.htm, 2003-03.

- [15] 丁玲. 企业人员流动管理 [J]. 经营与管理, 2003, (4): 24-25.

作者简介:

陈荣 (1980), 女, 河北保定人, 硕士研究生, 主要研究方向为软件工程; 谢高岗 (1974-), 男, 浙江衢州人, 副研究员, 博士, 主要研究方向为网络测试、测量与监控、软件工程; 张大方 (1959-), 男, 上海人, 院长, 教授, 博士生导师, 主要研究方向为可信系统与网络、软件工程; 王任栋 (1979-), 男, 山东青岛人, 硕士研究生, 主要研究方向为网络测试、软件工程。