

# 支持动态负荷调整的管理信息系统构建研究

饶 浩

(韶关学院 信息管理系,广东 韶关 512005)

**摘 要:**数据访问层中的数据库操作通常是现行多层式架构管理信息系统中制约系统整体运行效率的瓶颈。为提高多层式管理信息系统的效率,提出了支持动态负荷调整的多层式管理信息系统构建方案。该方案提出在数据访问层内设立连接池,建立数据生产使用流程的异步模型来实现信息数据的生产者与使用者的相对分离,以有效降低系统整体负荷,提高利用率;通过基于过程调整参数设计建立动态负荷调整因子算法和“放大缩小”处理策略,根据用户连接次数和系统环境需求等因素动态设置调整因子,以加快动态负荷调整过程的收敛性、响应性和系统的稳定性。通过采用先进先出(FIFO)和随机(Random)调度算法对已建支持动态负荷调整的系统架构进行了模拟测试。测试结果表明,基于动态调整连接池的系统构建方式不仅能够有效提高信息系统的承载能力,而且还有利于信息系统的维护修改和功能扩充,能满足信息系统平台未来多元化和综合化发展的需求。

**关键词:**动态调整;管理信息系统;业务逻辑层;数据访问层;系统架构;连接池

中图分类号:C931.6

文献标识码:A

文章编号:1673-629X(2017)03-0185-04

doi:10.3969/j.issn.1673-629X.2017.03.039

## Constitution Investigation of Management Information System Based on Dynamic Load Adjustment

RAO Hao

(Department of Information Management, Shaoguan University, Shaoguan 512005, China)

**Abstract:** Database operation in data access layer is often the technical bottleneck of management information system with multi-layer structure, which restricts its promotion of operation efficiency. In order to enhance the efficiency of the information system, a scheme for constructing multi-layer information system supported by dynamic load adjustment has been presented where connection pool is designed and built to lower the whole load of the information system as well as a asynchronous model involving the generation and exploitation of information data is constructed to improve the system utilization rate. Furthermore, a dynamic load adjustment factor and its algorithm have been designed and established based on several process adjustment parameters according to the number of user connection and requirement for environments of the information system, besides a treating strategy named zoom-in-and-out has been determined to promote the constringency and response of dynamic load adjustment and reliability of the whole system. A simulation test of the constructed multi-layer information system supported by dynamic load adjustment has been conducted employing both First-in-First-out (FIFO) and random regulating algorithm. Results of the test shows that the method for building information system based on dynamic load adjustment pool can not only boost the effective carrying capacity but also can make modification of system maintenance and enlargement of system functions easier and that the method can satisfy technical requests on multi-functional and integrative development of future information system.

**Key words:** dynamic adjustment; management information system; service logic tier; date access tier; system architecture; connection pool

## 0 引 言

随着信息时代的快速发展,政府、机构、企业等部门所涉及的各类有价值的数据、信息、文档,成为愈发重要的战略资源。由于信息用户的需求不断演进变

更,促使管理信息系统的功能和结构从单一趋于多元<sup>[1]</sup>,系统平台也逐渐向大型化、综合化发展<sup>[2]</sup>。信息系统的规划者和设计者所面临的问题是,在系统负荷增长的情况下,如何构建能尽量保持访问稳定性的管

收稿日期:2016-04-20

修回日期:2016-08-11

网络出版时间:2017-02-17

基金项目:教育部人文社会科学研究青年基金项目(13YJCZH144)

作者简介:饶 浩(1976-),男,副教授,研究方向为网络技术、信息系统分析。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20170217.1628.024.html>

理信息系统,以满足信息用户需求的扩展,维持较长的软件生命周期。

在当前管理信息系统的设计中,分层式架构是最主流、最重要的一种系统设计方式<sup>[3]</sup>。基本的分层方法遵循三层式架构,从上至下分别为:表示层、业务逻辑层、数据访问层。

表示层是用户进入系统时的界面环境,亦即系统向用户展现的界面,多采用 Web 方式表达,包括接收用户的请求,向用户显示数据结果,为客户端提供应用程序的交互式操作<sup>[4]</sup>,等等。业务逻辑层介于表示层与数据访问层之间,主要负责对数据业务逻辑的处理,是针对具体问题的操作,在数据交换中起到承上启下的作用。其功能是将数据访问层的存储逻辑组合起来形成业务规则,涉及业务规则的制定、业务流程的实现、合法性校验等与业务需求相关的设计。数据访问层是对原始数据(数据库或者文本文件等存放数据的形式)的操作层<sup>[5]</sup>,直接负责数据的添加、删除、修改、查找等操作,为上两层提供数据服务。如果要加入对象关系映射的元素,就需要建立对象和数据表之间的映射,把对象实体持久化到数据库。

分层式架构具有比较明显的优势。系统开发人员只需关注系统整体中的某一层,有利于开发团队的分工和开发工作齐头并进;便于以新的实现替代原有的实现,有利于系统业务能力的拓展和提高<sup>[6]</sup>;数据与业务逻辑之间相对分离,有利于降低系统内模块的耦合性,有效降低层与层之间的依赖,既有利于标准化,也有利于各层逻辑的复用,从而提高系统开发的效率<sup>[7]</sup>。在业务逻辑层设计合理的前提下,表示层即使发生改变,逻辑层也同样能完善地提供服务。对于分层式架构的信息系统,数据访问层中的数据库操作往往成为系统整体效率的瓶颈<sup>[8]</sup>。一般对数据库操作的步骤是:首先向数据申请一条数据库连接,然后执行操作语句,最后关闭此连接<sup>[9]</sup>。如果频繁的数据库操作都按照这种做法,每次操作所建立的数据库连接时间开销会非常大,而且降低了数据库操作的并发量。这使得数据访问层的设计,既要能提高数据库访问的并发量,又要保证成功建立数据连接的效率。

为实现上述目的,提出在数据访问层中设立连接池以降低系统负荷,并通过建立动态负荷调整因子,结合用户连接次数和系统环境需求等因素,动态设置调整因子以加快动态负荷的调整过程,同时采用“放大缩小”策略保证系统的稳定性。

## 1 连接池的设计

为了避免系统频繁申请数据库连接而产生的巨大时间开销,需要在数据访问层中建立一个连接池中间

件<sup>[10]</sup>,提供给持久化框架数据库连接,负责分配、管理和释放数据库连接<sup>[11]</sup>。通过预先建立一定数量的连接,每次根据用户需求分配一条连接,完成后则把该连接回收池中,以提供给下一个客户,这样可以节省每次数据库操作都需要建立连接的时间开销。连接池在整个系统结构上扮演的是一个中间件角色,不仅要像组件一样,有简单的接口提供功能性的要求,而且能够提供一定的可扩展性等非功能性要求<sup>[12]</sup>。

动态调整连接池模型如图 1 所示。

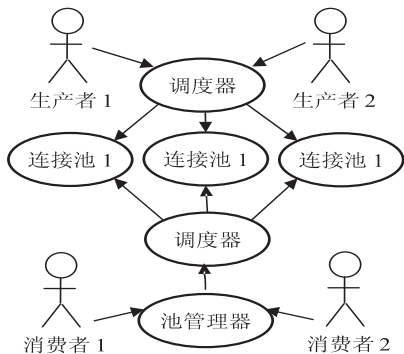


图 1 动态调整连接池模型

系统连接池采用的调用方式是异步生产者和消费者模型:生产者只管生产数据,然后放入连接池缓冲区,不管数据是否会被处理;消费者则从缓冲区中依次读取数据并按照自己的节奏进行处理<sup>[13]</sup>。当缓冲区没有数据时,消费者要等待生产者生产数据;当缓冲区满时,生产者要等待消费者取出数据,以便空出数据存放位置。异步生产者和消费者模型是多线程的,多线程必须要平衡线程之间的协作。数据操作的异步方式通过调度器实现,连接的管理、评估、调控由系统监听线程负责,用户只需以某一特定的重试次数,简单地向连接池管理器尝试连接,从而减少了请求者的动作负荷和申请释放连接的时间,提高了请求连接的并发量。

在获取数据库连接的过程中,为了让系统监听线程可以实时获取所产生的事件,并且做出及时响应,连接池需要一个事件监听模型。事件、事件触发器、事件处理机捆绑注册到事件源中,事件监听器以特定频率扫描所有的连接池事件触发器。当产生某一项事件,则调用该事件注册时所对应的事件处理机进行处理。

事件监听模型如图 2 所示。

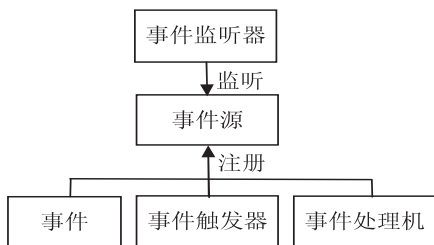


图 2 事件监听模型

需要由连接池事件触发器监听的事件包括:

- 检查连接池是否被关闭,是则强制销毁所有连接,并移除触发器。
- 检查忙队列中是否有连接需要被销毁,有则销毁。
- 检查忙队列中是否有连接使用次数达到最大值,是则回收。
- 检查忙队列中是否有被关闭的连接,是则关闭。
- 检查忙队列中是否有连接已经超过最大活动时间,是则回收。
- 检查忙队列+空闲队列的连接数是否超过最大连接数,是则向某一个池的空闲队列中减少一个连接。
- 检查忙队列+空闲队列的连接数是否小于最小连接数,是则向某一个池的空闲队列增加一个连接。
- 检查池管理器是否被置失效,是则关闭所有池并移除此触发器。
- 根据部分最近的用户重试次数作评估并调节连接池的连接数量。

## 2 动态负荷调整因子

连接池的动态负荷调整关键在于调整因子。其过程所涉及的调整参数包括:ListenerInterval, ListenerAmount, AdjustGeneRaise, Reduce, EvaluateRange, ReConnectTimes。其中,ListenerInterval 和 ListenerAmount 用于保证监听者的效率,使之可以及时对新的用户信息进行评估,或者是对监听的事件及时执行。连接池根据监听队列数量进行动态调整,监听者应该及时清空监听队列以取得最新的客户信息作评估,否则可能会因为评估不及时而延迟调整。

动态负荷调整因子的设置思想,是用最近的部分用户尝试取得连接的次数(也就是重连次数),除以这些用户的最大重试数的平均数作为调整因子,把此百分比对应到最小连接数到最大连接数的区间内。调整因子的计算过程如下:

(1)  $\_MaxConn = MaxConn + (MaxConn - MinConn) * EvaluateRange$ ; //根据用户配置的评估范围计算出用于调整的最大值,以便让连接数量可以达到最大数量。MaxConn 和 MinConn 分别是用户设置的最大连接数量和最小连接数量。

(2)  $\_MinConn = MinConn - (MaxConn - MinConn) * EvaluateRange$ ; //根据用户配置的评估范围计算出用于调整的最小值。\_MaxConn 和 \_MinConn 分别是用于调整的最大连接数量和最小连接数量。

(3) 调整因子 =  $(User_1 + User_2 + User_3 + User_4 + User_5) / (5 * ReConnectTimes)$ ; //将最近 5 个用户尝试取得连接的次数的平均值作为调整因子。

(4) 放大缩小处理 = 调整因子的放大缩小处理; //根

据配置对调整因子作放大或缩小处理。

(5) 目标连接数 =  $(\_MaxConn - \_MinConn) * \text{调整因子} + \_MinConn$ ; //计算出此调整因子所得到的应有目标连接数,目标连接数是根据真正的调整因子计算出的应有连接数量。

(6) 下限 = 目标连接数 -  $((\_MaxConn - \_MinConn) * (EvaluateRange / 2.0))$ ; //根据用户设置的评估范围,计算出连接数可波动范围的下限。

(7) 上限 = 目标连接数 +  $((\_MaxConn - \_MinConn) * (EvaluateRange / 2.0))$ ; //根据用户设置的评估范围,计算出连接数可波动范围的上限。

根据上述调整因子计算,得出连接数量合理范围的上限和下限。如果现有的连接数量比下限少,则需要增加连接数量;如果大于此范围的上限,则需要减少连接,向下调整,以此保持连接的数量。

这种计算方法,对于缓慢改变的连接量有比较好的效果。但如果连接量的改变具有突发性,例如,连接数量突然增加,甚至达到最大负荷,然后又剧减,那么根据以上的“调整因子”计算,连接数量会快速减少,甚至到最小负荷。如此重复跃变式的连接量变化会导致系统不稳定。

为了保证系统的稳定工作,同时让尽可能多的用户取得正常连接,降低“减少连接”的速度,系统采取“放大缩小处理”策略:先对“调整因子”作“放大处理”,乘一个放大系数(需根据具体运行环境测试得出),让“调整因子”更加倾向于增加连接,保证尽量大的用户数量正常取得连接。当放大后的“调整因子”跟上一次的“调整因子”比较起来呈现出“向下调整”的趋势,此时需要减少连接数量,则作“缩小处理”,把本次的“调整因子”按某比例降低(需根据实际运行环境测试得出,例如 200,则新的调整因子 =  $(\text{这次的调整因子} + 199 * \text{上次的调整因子}) / 200$ )。从而可以缓慢减少连接的速度,保留趋向于提高“增加连接”的机会,让系统倾向于使更多的用户取得正常连接。

## 3 并发压力测试

连接池并发压力测试在 MyEclipse 环境中配置执行,使用 MySQL 数据库,测试文件包含 3 个类:Counter,用于记录“无法取得连接的客户个数”和“某时刻的并发量”;takeConn,通过获取连接后睡眠一定的时间来模仿事务处理;test,测试的主程序,按照一定的睡眠时间来模仿一定量的并发客户连接。测试量为 10 000 个连接,即 10 000 个事务(模拟用户),事务长度是 1 000 ms。调度算法方面使用 FIFO(先进先出)和 RANDOM(随机);最小连接数量是  $6 * 5$  个池 = 30;最大连接数量是  $15 * 5$  个池 = 75;重试次数是 30。



测试配置文件如下:

```
<ConnectionPool>
<DataBase>
<Name>MySql</Name>
<MaxConnection>15</MaxConnection>
<MinConnection>6</MinConnection>
<TimeForWait>100</TimeForWait>
<TimeForConnActivity>10000</TimeForConnActivity>
<Driver>com. mysql. jdbc. Driver</Driver>
<URL>jdbc: mysql: // localhost: 3306/ test</URL>
<User>root</User>
<password>123456</password>
<MutiPool>5</MutiPool>
<Scheduler>fifo</Scheduler>
<ConnMaxUse>20</ConnMaxUse>
<CheckConnAvail>select 1 from user; </CheckConnAvail>
<ListenerInterval>2</ListenerInterval>
<ListenerAmount>2</ListenerAmount>
<ReConnectTimes>30</ReConnectTimes>
<EvaluateRange>0. 2</EvaluateRange>
<AdjustGeneRaise>1. 2</AdjustGeneRaise>
<Reduce>200</Reduce>
</DataBase>
</ConnectionPool>
```

测试结果如表 1 和表 2 所示。

表 1 使用 FIFO 调度算法的测试结果

FIFO 算法	理论平均 并发量	实际并发量 波动范围	无法连接 的客户数
小负荷测试	33	[ 23 , 43 ]	0
中负荷测试	50	[ 30 , 70 ]	1
满负荷测试	76	[ 46 , 106 ]	16
超负荷测试	100	[ 60 , 140 ]	89

表 2 使用 RANDOM 调度算法的测试结果

RANDOM 算法	理论平均 并发量	实际并发量 波动范围	无法连接 的客户数
小负荷测试	33	[ 25 , 41 ]	0
中负荷测试	50	[ 28 , 72 ]	2
满负荷测试	76	[ 41 , 111 ]	24
超负荷测试	100	[ 43 , 157 ]	175

对于 5 个池应用的情况,使用 FIFO 比 RANDOM 效果好。此动态调整连接池的配置,对于设置的最小连接数和最大连接数,介乎于小到中的连接数量可以取得令人满意的效果,但当并发量一旦超过了中上的连接数量,越偏近于最大负荷,动态调整的设置就越显得紧迫。所以更适合于把“最小连接数”设置在“实际平均”值附近或者稍微偏下,而让“最大连接数”尽量地设置大点,使其有足够的空间作调整。因此,对于调整因子系数的设置,必须参照于实际环境的测试效果。

4 结束语

基于动态负荷调整的信息系统开发,对管理信息系统开发人员有更高的技术要求,确定调整因子参数所需的测试也要花费额外的工作量。然而,一旦成功建立起系统,在系统运行过程中,通过动态调整的连接池设计,多个客户端可以通过共享少量的物理数据库连接来获得需求,尽可能多地满足信息用户业务需求,减少业务逻辑层对数据库频繁的连接操作。将连接池独立分层的系统,能够更自如地应对业务规则的变化,提高业务的扩展能力和系统性能。层次的细化和独立,支持并行开发,提高了软件开发的效率和质量,降低了模块间的耦合度,拥有更好的可维护性,符合现代大中型管理信息系统的高稳定性、高可扩展性等要求。

参考文献:

[1] 王金光,陈贞翔. 基于 Web 的网络流量分类管理系统[J]. 济南大学学报:自然科学版,2014,28(1):71-76.

[2] 宋 静,何 伟,李浙昆. 基于 Web 的情报信息系统的设计与实现[J]. 计算机技术与发展,2014,24(9):209-212.

[3] 李良斌,王劲林,陈 君. 网络服务系统可生存性分析与验证[J]. 沈阳工业大学学报,2012,34(6):677-684.

[4] Lai I K W, Tong V W L, Lai D C F. Trust factors influencing the adoption of Internet-based interorganizational systems[J]. Electronic Commerce Research and Applications, 2011,10(1):85-93.

[5] 张 龙,肖琬蓉. 集群数据库内容管理系统的设计与实现[J]. 情报杂志,2012,31(2):130-135.

[6] 霍占强,张锦程,王志衡. 数据库连接池的数学建模与性能分析[J]. 计算机工程,2014,40(10):32-36.

[7] Chu R, Lai I K W, Lai D C F. Trust factors influencing the adoption of cloud-based interorganizational systems: a conceptual model[C]//Proc of ICEMSI & iCETS. [s. l.]: [s. n.],2013.

[8] 谷 伟. 基于网络的自适应测试系统的设计与实现[J]. 湘潭大学自然科学学报,2013,35(2):98-102.

[9] Singh A K. Ajax asynchronous database refresh[J]. International Journal of Information and Communication Technology Research,2012,2(8):669-703.

[10] 吕健波,戴冠中,慕德俊. 绝对延迟保证在 Web 应用服务器数据库连接池中的实现[J]. 计算机应用研究,2012,29(5):1838-1841.

[11] Hair J F, Jr Anderson R E, Tatham R L, et al. Multivariate data analysis[M]. 7th ed. Englewood Cliffs, NJ: Prentice-Hall Inc.,2010.

[12] 王倩宜,欧阳荣彬,龙新征. SaaS 云服务环境下的管理信息系统方案研究[J]. 华东师范大学学报:自然科学版,2015(6):134-142.

[13] 孟培超,胡圣波,舒 恒,等. 基于 ADO 数据库连接池优化策略[J]. 计算机工程与设计,2013,34(5):1706-1710.