(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 1815469 B
(45)授权公告日 2010.04.28

(21)申请号 200510005221.9

(22)申请日 2005.02.01

(73)专利权人 华为技术有限公司
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72)发明人 王仲涛

(74)专利代理机构 北京德琦知识产权代理有限公司 11018
代理人 张颖玲 王琦

(51)Int.Cl.
G06F 17/30(2006.01)

(56)对比文件
CN 1510575 A,2004.07.07, 说明书第7页第5行－第10页第20行,权利要求1,摘要,图2.
US 2003135542 A1,2003.07.17, 全文.

审查员 唐田田

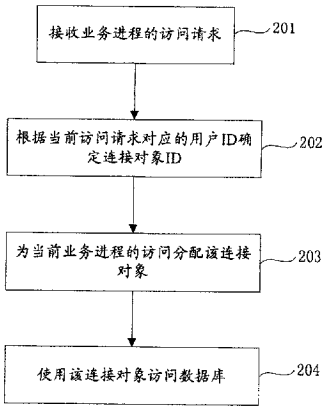
权利要求书 2 页 说明书 9 页 附图 4 页

(54)发明名称

一种数据库连接资源的管理方法

(57)摘要

本发明公开了一种数据库连接资源的管理方法,预先配置应用流程属性和连接对象的对应关系,并包括:A. 数据库连接池接收到业务进程的访问请求后,根据所述访问请求中的应用流程属性和所述预先配置的对应关系,确定与该访问请求对应的连接对象;B. 将步骤A所确定的连接对象分配给所述业务进程的访问。本发明还公开了另一种数据库连接资源的管理方法。采用本发明方法,能够避免污读现象,并保证数据库系统性能。



CN 1815469 B

1. 一种数据库连接资源的管理方法，其特征在于，预先配置应用流程属性和连接对象的对应关系，并包括如下步骤：

A. 数据库连接池接收到业务进程的访问请求后，根据所述访问请求中的应用流程属性和所述预先配置的对应关系，确定与该访问请求对应的连接对象；

B. 将步骤 A 所确定的连接对象分配给所述业务进程的访问。

2. 根据权利要求 1 所述的方法，其特征在于，所述应用流程属性和连接对象的对应关系为：具有逻辑关系的一个或一个以上应用流程对应同一个连接对象。

3. 根据权利要求 2 所述的方法，其特征在于，所述具有逻辑关系的一个或一个以上应用流程为具备相同属性的应用流程。

4. 根据权利要求 1 所述的方法，其特征在于，步骤 A 中，

所述配置应用流程属性和连接对象的对应关系的方法为：构造哈希 HASH 算法；

所述根据应用流程属性和所述预先配置的对应关系，确定与该访问请求对应的连接对象的方法为：根据应用流程属性，通过所构造的 HASH 算法得到连接对象 ID。

5. 根据权利要求 1 所述的方法，其特征在于，步骤 A 中，

所述配置应用流程属性和连接对象的对应关系的方法为：设置应用流程属性与连接对象之间的映射关系表；

所述根据应用流程属性和所述预先配置的对应关系，确定与该访问请求对应的连接对象的方法为：根据应用流程属性查询该映射关系表确定连接对象。

6. 根据权利要求 1 至 5 任一项所述的方法，其特征在于，所述应用流程属性为：用户 ID、或群组 ID。

7. 根据权利要求 1 至 5 任一项所述的方法，其特征在于，该方法进一步包括：数据库连接池周期性调整连接对象数目；

所述调整连接对象数目的方法为：预先设置申请连接对象成功率的合理范围，周期性统计申请连接对象的成功率，如果申请连接对象成功率低于该合理范围的最小值，则增加连接对象数目；如果申请连接对象成功率高于该合理范围的最大值，则减少连接对象数目；如果申请连接对象成功率在该合理范围内，则维持连接对象数目。

8. 根据权利要求 1 至 5 任一项所述的方法，其特征在于，该方法进一步包括：数据库连接池周期性调整连接对象数目；

所述调整连接对象数目的方法为：预先设置连接对象占用率的合理范围，周期性统计连接对象的占用率，如果连接对象占用率低于该合理范围的最小值，则减少连接对象数目；如果连接对象占用率高于该合理范围的最大值，则增加连接对象数目；如果连接对象占用率在该合理范围内，则维持连接对象数目。

9. 一种数据库连接资源的管理方法，其特征在于，建立并实时维护应用流程的连接对象使用状态表；并包括如下步骤：

a、数据库连接池接收到业务进程的访问请求后，根据所述访问请求中的应用流程属性查询所述连接对象使用状态表，判断具有该属性的应用流程是否已经分配过连接对象，如果是，则确定该已分配的连接对象为与该访问请求所对应的连接对象，执行步骤 b；否则确定任意一个连接对象为与该访问请求所对应的连接对象，执行步骤 b；

b、将步骤 a 所确定的连接对象分配给所述业务进程的访问。

10. 根据权利要求 9 所述的方法，其特征在于，所述应用流程属性为：用户 ID、或群组 ID。

11. 根据权利要求 9 所述的方法，其特征在于，该方法进一步包括：数据库连接池周期性调整连接对象数目；

所述调整连接对象数目的方法为：预先设置申请连接对象成功率的合理范围，周期性统计申请连接对象的成功率，如果申请连接对象成功率低于该合理范围的最小值，则增加连接对象数目；如果申请连接对象成功率高于该合理范围的最大值，则减少连接对象数目；如果申请连接对象成功率在该合理范围内，则维持连接对象数目。

12. 根据权利要求 9 所述的方法，其特征在于，该方法进一步包括：数据库连接池周期性调整连接对象数目；

所述调整连接对象数目的方法为：预先设置连接对象占用率的合理范围，周期性统计连接对象的占用率，如果连接对象占用率低于该合理范围的最小值，则减少连接对象数目；如果连接对象占用率高于该合理范围的最大值，则增加连接对象数目；如果连接对象占用率在该合理范围内，则维持连接对象数目。

一种数据库连接资源的管理方法

技术领域

[0001] 本发明涉及数据库访问技术，特别涉及数据库连接资源的管理方法。

背景技术

[0002] 随着计算机技术的发展，数据管理的规模越来越大，不仅数据量急剧增加，数据共享的要求也越来越高，于是出现了数据库系统用于对所有数据实行统一、集中、独立的管理，使数据存储独立于使用数据的程序，实现数据共享。

[0003] 目前，大型中央数据库系统在许多领域，尤其是通信领域，得到了广泛应用。这种大型数据库系统需要并行处理多个业务进程的访问，并为各业务进程的访问分配数据库连接资源，如何保证大量在线事务处理 (OLTP) 并发的情况下数据库系统的高效运行、以及数据的一致性和安全性，成为数据库连接资源管理技术主要解决的问题。

[0004] 所述数据库系统至少包括数据库管理系统 (DBMS) 和物理的数据库 (DataBase)，以及作为支撑的计算机软 / 硬件系统。其中，DBMS 负责接收业务进程的访问请求，为业务进程建立数据库连接，并根据该业务进程的访问需求通过数据库连接对 DataBase 进行操作，比如 ：读取、写入、删除数据库记录等。在业务进程访问数据库时，每个业务进程将独占一条数据库连接，这样对于大量业务进程并发访问的数据库系统来讲，系统的响应时间将过长，很难保证良好的数据库系统性能。因此，在大量业务进程并发访问的数据库系统中，还设置了用于对数据库连接资源进行管理的数据库连接池。

[0005] 所述数据库连接池，在系统初始化时，根据配置文件确定向 DBMS 申请建立数据库连接的数目，并将所建立的每个数据库连接封装成连接对象。当用户的应用流程访问数据库时，系统首先将应用流程随机分配至某个业务进程，再由业务进程访问数据库，数据库连接池将为业务进程的每一次访问分配连接对象。每个业务进程可以由多个应用流程共享，每个连接对象可以由多个业务进程共享，这样，数据库的连接资源不再被单个业务进程的访问独占，从而能够充分利用系统资源访问数据库。也就是说 ：连接对象与多个业务进程的访问建立多个软连接，每个连接对象与数据库之间建立一个硬连接，该多个业务进程的访问共享连接对象与数据库之间的硬连接来访问数据库。但是，当多个应用流程各自通过不同业务进程并使用不同的连接对象访问数据库时，如果该多个应用流程之间存在一定的逻辑关系，就可能出现污读现象。

[0006] 下面结合图 1 对现有多业务进程访问数据库出现的问题加以详细说明。图 1 为现有技术数据库连接资源分配情况示意图。如图 1 所示，业务进程通过数据库连接池与 DBMS 通信，由 DBMS 根据业务进程的访问请求对 DataBase 进行操作。其中，当前进行访问的用户包括 ：用户 1、用户 2、用户 3 ；业务进程有 N 个 ：用户 1 的流程 A 和用户 2 的流程 B 被分配到业务进程 1，用户 1 的流程 B 和用户 2 的流程 A 被分配到业务进程 2，.....，用户 3 的流程 A 和流程 B 被分配到业务进程 N ；数据库连接池中建立了 M 个连接对象，即数据库连接池通过 DBMS 与 DataBase 建立了 M 个数据库硬连接。应用流程每次通过业务进程访问数据库时，数据库连接池将为该业务进程随机分配一个连接对象，即建立数据库软连接。图 1 中，

数据库连接池为用户 1 的流程 A 和流程 B、用户 2 的流程 B 的访问分配连接对象 1，并分别建立连接 1、连接 2 和连接 3；为用户 2 的流程 A、用户 3 的流程 A 和流程 B 分配连接对象 2，并分别建立连接 4、连接 5 和连接 6。

[0007] 其中，用户 2 的流程 A 和流程 B 的访问对应的是不同的连接对象，这样如果该流程 A 与流程 B 之间存在一定的逻辑关系，用户 2 访问数据库时可能出现污读。以无线通讯领域的归属位移寄存器 (HLR) 设备为例，该设备需要处理的消息流程有多种，而且对于同一个用户的不同消息流程而言，它们之间存在逻辑关系。假定流程 A 为位置登记流程，流程 B 为取路由流程，只有成功执行了位置登记流程，取路由流程才能根据位置登记后的数据正确获取用户 2 漫游地信息。由于，流程 A 与流程 B 分别通过连接对象 2 和连接对象 1 使用不同的硬连接访问数据库，也就将分别使用各自硬连接对应的数据缓存区对数据库进行操作。因此，如果在先的流程 A 结束时没有将其对于数据库的改动及时提交到数据库中，那么这些更新的数据就保存在连接对象 2 对应的数据缓存区中，随后的流程 B 通过连接对象 1 的硬连接访问数据库时，读到的是连接对象 1 对应的数据缓存区中的数据，该数据为未经流程 A 更新的“脏”数据，即产生污读。这里，数据库系统为每个数据库硬连接分配数据缓存区，将当前对数据库的修改缓存起来；DBMS 将根据系统性能要求等因素在若干应用流程结束后，将数据缓存区的数据更新提交到 DataBase。

[0008] 当然，为了避免上述污读情况，可以在每一个应用流程结束后均执行提交数据库操作，即每个应用流程对应一个数据库事务。这样从理论上可行，但频繁的提交 (COMMIT) 操作将导致输入 / 输出 (I/O) 迅速成为数据库系统的性能瓶颈。以 Oracle 数据库系统为例，每次执行 COMMIT 操作时，系统的写日志进程 (LGWR) 就会将日志缓存区的重做日志 (redo log) 写入日志文件。由于写日志文件涉及到外存访问，因此系统需要等待的时间就会显著增加。当应用流程即数据库事务非常多，执行 COMMIT 操作非常频繁时，系统等待写 redo log 的时间超过了其它所有操作的时间，也就是说：系统主要是在等待写 redo log 的完成。因此，导致 I/O 显著增长为系统性能的瓶颈，消息流程时延的均值和离散程度都显著增加，系统的运行状态变得非常不稳定。

[0009] 综上所述，现有数据库连接池的数据库连接分配机制不能有效使所有应用流程使用所分配的连接访问到准确的数据，如果要避免污读只能使系统性能显著下降，甚至造成系统崩溃。如何在维持一定系统性能的基础上保证数据库访问的有效性成为目前数据库连接资源管理技术中亟待解决的问题。

发明内容

[0010] 有鉴于此，本发明的主要目的在于提供一种数据库连接资源的管理方法，能够避免多业务进程同时访问数据库时出现的污读，并有效保证数据库系统的 I/O 性能。

[0011] 为达到上述目的，本发明的技术方案是这样实现的：

[0012] 一种数据库连接资源的管理方法，预先配置应用流程属性和连接对象的对应关系，并包括：

[0013] A. 数据库连接池接收到业务进程的访问请求后，根据所述访问请求中的应用流程属性和所述预先配置的对应关系，确定与该访问请求对应的连接对象；

[0014] B. 将步骤 A 所确定的连接对象分配给所述业务进程的访问。

[0015] 其中，所述应用流程属性和连接对象的对应关系为：具有逻辑关系的一个或一个以上应用流程对应同一个连接对象。

[0016] 其中，所述具有逻辑关系的一个或一个以上应用流程为具备相同属性的应用流程。

[0017] 步骤 A 中，所述配置应用流程属性和连接对象的对应关系的方法为：构造哈希(HASH) 算法；所述根据应用流程属性和所述预先配置的对应关系，确定与该访问请求对应的连接对象的方法为：根据应用流程属性，通过所构造的 HASH 算法得到连接对象 ID。

[0018] 步骤 A 中，所述配置应用流程属性和连接对象的对应关系的方法为：设置应用流程属性与连接对象之间的映射关系表；所述根据应用流程属性和所述预先配置的对应关系，确定与该访问请求对应的连接对象的方法为：根据应用流程属性查询该映射关系表确定连接对象。

[0019] 其中，所述应用流程属性为：用户 ID、或群组 ID。

[0020] 其中，该方法进一步包括：数据库连接池周期性调整连接对象数目；所述调整连接对象数目的方法为：预先设置申请连接对象成功率的合理范围，周期性统计申请连接对象的成功率，如果申请连接对象成功率低于该合理范围的最小值，则增加连接对象数目；如果申请连接对象成功率高于该合理范围的最大值，则减少连接对象数目；如果申请连接对象成功率在该合理范围内，则维持连接对象数目。

[0021] 其中，该方法进一步包括：数据库连接池周期性调整连接对象数目；所述调整连接对象数目的方法为：预先设置连接对象占用率的合理范围，周期性统计连接对象的占用率，如果连接对象占用率低于该合理范围的最小值，则减少连接对象数目；如果连接对象占用率高于该合理范围的最大值，则增加连接对象数目；如果连接对象占用率在该合理范围内，则维持连接对象数目。

[0022] 本发明还公开了另一种数据库连接资源的管理方法，其特征在于，建立并实时维护应用流程的连接对象使用状态表；并包括如下步骤：

[0023] a、数据库连接池接收到业务进程的访问请求后，根据所述访问请求中的应用流程属性查询所述连接对象使用状态表，判断具有该属性的应用流程是否已经分配过连接对象，如果是，则确定该已分配的连接对象为与该访问请求所对应的连接对象，执行步骤 b；否则确定任意一个连接对象为与该访问请求所对应的连接对象，执行步骤 b；

[0024] b、将步骤 a 所确定的连接对象分配给所述业务进程的访问。

[0025] 由上述方案可以看出，本发明的关键在于：为分配到各业务进程的应用流程中具备逻辑关系的应用流程的访问分配同一连接对象。因此，本发明所提供的数据库连接资源的管理方法，能够保证数据库访问中数据读取的正确性，使用户得以灵活配置数据库 COMMIT 的频率，保证系统的 I/O 性能，还能够通过动态调整连接对象数目使数据库连接资源的有效利用率达到最优。

附图说明

[0026] 图 1 为现有技术数据库连接资源分配情况示意图；

[0027] 图 2 为本发明中分配数据库连接资源一较佳实施例处理流程示意图；

[0028] 图 3 为本发明中分配数据库连接资源另一较佳实施例处理流程示意图；

[0029] 图 4 为应用本发明技术分配数据库连接资源的总体情况示意图；

[0030] 图 5 为本发明中调整连接对象数目一较佳实施例处理流程示意图；

[0031] 图 6 为本发明中调整连接对象数目另一较佳实施例处理流程示意图。

具体实施方式

[0032] 下面结合附图及具体实施例对本发明再作进一步详细的说明。

[0033] 本发明提供了一种数据库连接资源的管理方法，其主要设计思想为：应用流程通过业务进程访问数据库时，数据库连接池为分配给各业务进程的应用流程中具有逻辑关系的一个或一个以上应用流程的访问分配同一连接对象。

[0034] 其中，可以使用应用流程属性来关联应用流程之间的逻辑关系，因此可以将具备相同属性的应用流程视为具有逻辑关系的应用流程。因此，数据库连接池应为分配给各业务进程的应用流程中具有相同属性的应用流程的访问分配同一连接对象，该分配方法基本有两种：一、预先配置应用流程属性与连接对象的对应关系；在应用流程访问数据库时，数据库连接池根据应用流程属性直接确定连接对象并分配给当前业务进程的访问。二、实时保存应用流程每一次访问数据库时使用连接对象的情况，即维护连接对象使用状态表；当数据库连接池接收到业务进程的访问请求后，根据应用流程属性查询该连接对象使用状态表，判断具有该属性的应用流程是否已分配过连接对象，如果是，分配该连接对象给当前业务进程的访问；否则任意分配连接对象给当前业务进程的访问。这里，所述分配连接对象即为分配数据库连接资源。

[0035] 下面结合图 2 和图 3 分别对本发明中采用上述两种分配连接对象方法的实施例加以详细描述。

[0036] 图 2 为本发明中分配数据库连接资源一较佳实施例处理流程示意图。在本实施例中，数据库连接池采用上述第一种方法来分配连接对象，即：根据应用流程属性直接确定连接对象，所述应用流程属性为用户标识 (ID)。其中，对于通信、金融等领域的大型数据库来说，通常同一用户的多个应用流程之间具备逻辑关系，比如：某用户成功执行了位置登记流程，其取路由流程才能根据位置登记后的数据正确获取该用户的漫游地信息；因此，通常将用户 ID 相同的应用流程视为具备逻辑关系的应用流程。本发明所述应用流程属性可以为用户 ID、群组 ID 等，关于具体采用哪种应用流程属性来关联应用流程之间的逻辑关系与数据库管理系统所支持的应用软件系统功能相关，本发明不进行限定。

[0037] 如图 2 所示，具体处理如下：

[0038] 步骤 201：数据库连接池接收业务进程的数据库访问请求。这里，应用流程访问数据库时，数据库管理系统所支持的应用软件系统将为应用流程随机分配业务进程，该业务进程将向数据库连接池发起数据库访问请求，该请求包含用户 ID、访问需求等信息。

[0039] 步骤 202：数据库连接池根据接收到的数据库访问请求中的用户 ID 确定当前访问对应的连接对象 ID。

[0040] 其中，数据库连接池预先配置用户 ID 与连接对象之间的对应关系，从而可以根据接收到的用户 ID 直接确定其对应的连接对象 ID。所述配置用户 ID 与连接对象 ID 之间对应关系的方法基本有两种：1、构造哈希 (HASH) 算法，该算法根据用户 ID 能够直接得到连接对象索引值，该连接对象索引值通常为连接对象 ID；2、保存用户 ID 与连接对象 ID 之间的

静态映射关系表，从而根据用户 ID 查询此映射关系表就可以得到连接对象 ID。本发明中，一个连接对象可以对应多个应用流程，一个应用流程则只能使用一个连接对象，因此所配置的用户 ID 与连接对象 ID 之间对应关系的基本原则为：一个用户 ID 仅对应一个连接对象 ID，但并不限定每个连接对象 ID 所对应的用户 ID 数量。

[0041] 这里，所述 HASH 算法的具体运算规则有多种，本发明并不限定。比如可以构造 HASH 算法的运算规则为：15 位十进制数的各位数之和除以连接对象数目，将所得到的余数作为连接对象 ID 输出。该算法应用于通信领域时，用户 ID 采用国际移动用户标识 (IMSI)，假定：IMSI 为 460010755123456，当前连接对象数目为 6，该 IMSI 各位数之和为 49，则使用该算法输入该 IMSI，将输出索引值 1，即得到连接对象 ID 为 1。关于具体如何构造 HASH 算法属公知技术，本文不作详述。

[0042] 步骤 203：数据库连接池根据步骤 202 所确定的连接对象 ID，将该连接对象分配给当前业务进程的访问，从而当前业务进程与该连接对象建立数据库软连接。

[0043] 步骤 204：步骤 203 所分配的连接对象根据业务进程的访问请求中的访问需求，使用自身与数据库之间的硬连接访问数据库，从而完成对数据库记录的相应操作。

[0044] 这里，所述访问需求为指示对数据库记录进行哪些操作以及对哪些内容进行操作的信息；所述操作为：查询、增加、修改、或删除等等。

[0045] 上述实施例中，预先通过构造 HASH 算法或保存静态映射关系表配置应用流程属性与连接对象的对应关系，则应用流程属性与连接对象之间的对应关系固定。如果事先并不配置应用流程属性与连接对象之间固定的对应关系，而是在应用流程访问数据库时动态确定连接对象，则可以采用上述第二种方法来分配连接对象，具体处理方法参见图 3。

[0046] 图 3 为本发明中分配数据库连接资源另一较佳实施例处理流程示意图。本实施例中，同样将用户 ID 作为应用流程属性来判决各应用流程是否具备逻辑关系，将用户 ID 相同的应用流程视为具备逻辑关系的应用流程。如图 3 所示，处理步骤包括：

[0047] 步骤 301：数据库连接池接收业务进程的数据库访问请求。

[0048] 步骤 302：数据库连接池根据接收到的数据库访问请求中的用户 ID，判断该用户的应用流程是否已分配过连接对象，如果是，执行步骤 303；否则执行步骤 304。

[0049] 其中，数据库连接池实时维护连接对象使用状态表，每一次为业务进程的访问分配连接对象之后，均将当前访问对应的应用流程属性及所分配的连接对象信息添加至该状态表中。该状态表中至少保存：应用流程属性、具有该属性的应用流程所分配过的连接对象信息。这里，所述应用流程属性为用户 ID，所述连接对象信息为连接对象 ID，则本步骤中，根据用户 ID 查询该连接对象使用状态表能够得知是否已为该用户的应用流程分配过连接对象，并能够得到已分配的连接对象 ID。

[0050] 步骤 303：根据步骤 302 得到的已分配的连接对象 ID，将该连接对象分配给当前业务进程的访问，从而当前业务进程的访问与该连接对象建立数据库软连接，执行步骤 305。

[0051] 步骤 304：任意选择一个连接对象 ID，将该连接对象分配给当前业务进程的访问，从而当前业务进程的访问与该连接对象建立数据库软连接。同时将当前访问对应的用户 ID 及所分配的连接对象 ID 添加至该状态表中。

[0052] 步骤 305：所分配的连接对象根据业务进程的访问请求中的访问需求，使用自身与数据库之间的硬连接访问数据库，从而完成对数据库记录的相应操作。

[0053] 图 2 和图 3 所述实施例为数据库连接池针对一次访问请求分配连接对象的处理，每一次接收到数据库访问请求时，均将执行图 2 或图 3 所述的处理，从而为具备相同属性的应用流程分配同一连接对象。图 4 为应用本发明技术分配数据库连接资源的总体情况示意图。

[0054] 图 4 与图 1 所示类似，业务进程通过数据库连接池与 DBMS 通信，由 DBMS 根据业务进程的访问请求对 DataBase 进行操作。其中，当前进行访问的用户包括 :用户 1、用户 2、用户 3 ;业务进程有 N 个 :用户 1 的流程 A 和用户 2 的流程 B 属业务进程 1，用户 1 的流程 B 和用户 2 的流程 A 属业务进程 2，.....，用户 3 的流程 A 和流程 B 属业务进程 N ;数据库连接池中建立了 M 个连接对象，即通过 DBMS 与 DataBase 建立了 M 个数据库硬连接。应用流程每次通过业务进程访问数据库时，数据库连接池将按本发明方法为该业务进程实时分配一个连接对象，即建立数据库软连接。

[0055] 这里，选择用户 ID 作为关联应用流程之间逻辑关系的应用流程属性，因此，应用图 2 或图 3 所述实施例能够为具有相同用户 ID 的应用流程的访问分配同一连接对象。如图 4 所示，数据库连接池为用户 1 的流程 A 和流程 B 的访问分配连接对象 1，并分别建立连接 1 和连接 3 ;为用户 2 的流程 A 和流程 B 的访问分配连接对象 2，并分别建立连接 4 和连接 2 ;为用户 3 的流程 A 和流程 B 分配连接对象 M，并分别建立连接 5 和连接 6。

[0056] 对于每个用户来说，其各个应用流程通过业务进程访问数据库时，数据库连接池为其分配同一连接对象，通过同一硬连接对数据库进行操作，并共享同一数据缓冲区，这样，就避免了背景技术所述先后执行同一用户不同应用流程的访问时出现的污读现象。以图 4 中用户 2 的应用流程访问为例 :数据库连接池为用户 2 的流程 A 和流程 B 分配连接对象 2，从而该流程 A 和流程 B 可以共享连接对象 2 的数据缓存区，假定流程 A 在先执行，流程 B 在后执行。由于该两个流程共享同一数据缓存区，流程 A 对数据库的改动已保存在数据缓存区中，流程 B 在后执行时将根据该数据缓存区的数据进行操作，这样不管流程 A 结束时是否已将数据更新提交到数据库，都不会对流程 B 的执行造成影响，也就避免了上述污读现象。

[0057] 可见，应用本发明方法，数据库连接池根据业务进程的访问请求分配数据库连接资源时，不必考虑与当前请求的应用流程具备逻辑关系的应用流程对数据库的修改是否已经提交。同样，数据库系统在配置 COMMIT 的频率时也不必关心应用流程之间的逻辑关系，不仅大大简化了系统中数据库连接资源管理部分的开发过程，而且在系统运行时可以灵活调整 COMMIT 的频率，使系统的 I/O 性能保持最优。其中，业务进程每一次访问数据库时，根据当前访问的业务逻辑将生成一条或一组结构化查询语句 (SQL)，所谓 COMMIT 的频率可由每次将数据更新提交数据库时执行的 SQL 语句量来表征 :如果每次执行的 SQL 语句量较多，则说明 COMMIT 频率较低 ;反之，如果每次执行的 SQL 语句量较少，则说明 COMMIT 频率较高。

[0058] 本发明的数据库连接池在保证具有逻辑关系的应用流程的访问使用同一连接对象的同时，还进一步对数据库连接池中连接对象的数目进行动态调整，使数据库连接资源得到有效利用。下面结合图 5 和图 6 对本发明调整连接对象数目的方法加以详细描述。

[0059] 图 5 为本发明中调整连接对象数目一较佳实施例处理流程示意图。在本实施例中，数据库连接池周期性统计申请连接对象的成功率，并根据所统计的申请连接对象成功率来决定如何调整连接对象数目。系统预先配置统计周期时长，并设置申请连接对象成功

率的合理范围。对于每一个统计周期来说，具体处理如图 5 所示：

[0060] 步骤 501：数据库连接池获取本统计周期内，业务进程申请连接对象成功的次数，以及业务进程申请连接对象的总次数。

[0061] 步骤 502：根据步骤 501 所统计的申请连接对象成功的次数与所述申请连接对象总次数，计算得到申请连接对象的成功率。这里，可以采用申请连接对象成功的次数与申请连接对象总次数的比值来表示申请连接对象成功率。

[0062] 步骤 503：判断步骤 502 计算得到的申请连接对象成功率是否在所设置的合理范围内，如果该申请连接对象成功率高于此范围的最大值，则执行步骤 504；如果该申请连接对象成功率低于此范围的最小值，则执行步骤 505；如果该申请连接对象成功率的值在此范围内，则结束当前处理。

[0063] 步骤 504：减少连接对象数目，即释放部分与数据库之间的硬连接，结束当前处理。

[0064] 步骤 505：增加连接对象数目，即建立若干与数据库之间的硬连接，结束当前处理。

[0065] 其中，所述增加或减少连接对象数目的方法属公知技术，并且非本发明解决的问题，因此不作详述。关于每次减少或增加多少个连接对象，由数据库系统实现的具体情况确定，通常可在系统运行前通过测试得到每次增加或减少连接对象的数目，并通过配置文件在系统初始化时设置在数据库连接池中，本发明对此不进行限定。

[0066] 本实施例中，所述申请连接对象的成功率可以表征连接对象使用的忙闲程度，如果该成功率较高，则说明连接对象比较空闲，即数据库连接资源足够使用，那么当成功率高于一定数值时就有必要减少连接对象数目来节约数据库连接资源；如果该成功率较低，则说明连接对象使用频率较高，即数据库连接资源比较紧张，那么当成功率低于一定数值时就有必要增加连接对象数目来补充数据库连接资源，以维持数据库系统的 I/O 性能。

[0067] 因此，本发明预先设置申请连接对象成功率的合理取值范围，当统计得到的申请连接对象成功率高于该范围的最大值，则减少连接对象数目；当统计得到的申请连接对象成功率低于该范围的最小值，则增加连接对象数目；当统计得到的申请连接对象成功率在该范围内，则维持当前连接对象数目。

[0068] 图 6 为本发明中调整连接对象数目另一较佳实施例处理流程示意图。在本实施例中，数据库连接池周期性统计连接对象占用率，并根据所统计的连接对象占用率来决定如何调整连接对象数目。系统预先配置统计周期时长，并设置连接对象占用率的合理范围。对于每一个统计周期来说，如图 6 所示，具体处理包括：

[0069] 步骤 601：数据库连接池获取本统计周期内，业务进程占用连接对象的总时间。

[0070] 其中，可以统计每次数据库访问时连接对象被业务进程占用的时间，将各次数据库访问时连接对象被占用的时间累加，即得到业务进程占用连接对象的总时间。上述统计占用连接对象总时间的具体实现机制有多种，本发明并不限定，这里仅以时间戳机制为例进行说明：预先为各连接对象设置时间戳，当业务进程调用连接对象时，该连接对象将被打上时间戳；当业务进程完成访问释放连接对象时，该连接对象也被打上时间戳，两个时间戳的值之差即为当前数据库访问中连接对象被占用的时间。每个连接对象每次数据库访问时均按此方法记录自身被占用的时间，当统计周期结束时，将各连接对象每次数据库访问时

被占用的时间累加，就得到了所有连接对象被占用的总时间。

[0071] 步骤 602 ：根据步骤 601 所获取的占用连接对象的总时间计算得到连接对象的占用率。这里，所述连接对象的占用率可以由所述占用连接对象的总时间与该统计周期时长的比值来表征。

[0072] 步骤 603 ：判断步骤 602 计算得到的连接对象占用率是否在所设置的合理范围内，如果该连接对象占用率高于此范围的最大值，则执行步骤 604 ；如果该连接对象占用率低于此范围的最小值，则执行步骤 605 ；如果该连接对象占用率的值在此范围内，则结束当前处理。

[0073] 步骤 604 ：增加连接对象数目，结束当前处理。

[0074] 步骤 605 ：减少连接对象数目，结束当前处理。

[0075] 本实施例中，所述连接对象的占用率也可以表征连接对象使用的忙闲程度，如果该占用率较低，则说明连接对象比较空闲，即数据库连接资源足够使用，那么当占用率低于一定数值时就有必要减少连接对象数目来节约数据库连接资源 ；如果该占用率较高，则说明连接对象使用频率较高，即数据库连接资源比较紧张，那么当占用率高于一定数值时就有必要增加连接对象数目来补充数据库连接资源，以保证数据库系统良好的 I/O 性能。

[0076] 因此，本发明预先设置连接对象占用率的合理取值范围，当统计得到的连接对象占用率低于该范围的最小值，则减少连接对象数目 ；当统计得到的连接对象占用率高于该范围的最大值，则增加连接对象数目 ；当统计得到的连接对象占用率在该范围内，则维持当前连接对象数目。

[0077] 应用上述图 5 和图 6 所述的方法来动态调整连接对象数目，使数据库连接池中连接对象的数目始终保持最优，即使在数据库系统总的访问量较大时，也能保证每一个数据库访问都能够得到及时响应，从而有效改善系统的 I/O 性能。

[0078] 由于，当调整连接对象数目时，如果采用所述第一种分配连接对象的方法，则连接对象数目的变化将引发应用流程属性与连接对象之间对应关系的改变。尤其，当使用 HASH 算法配置应用流程属性与连接对象的对应关系时，连接对象数目变化通常将造成 HASH 算法的输出的连接对象 ID 值的改变，比如 ：在连接对象数目调整前，属性为 A 的应用流程对应连接对象 1 ；在连接对象数目调整后，根据 HASH 算法的输出值，该属性为 A 的应用流程对应的连接对象变为连接对象 2。因此，如果连接对象数目的变化将引发应用流程属性与连接对象对应关系的改变，则在应用上述图 5 和图 6 所述的方法动态调整连接对象数目之前，需要将所有连接对象对数据库的改动提交数据库，以确保在应用流程属性与连接对象之间的对应关系发生改变前，所有的数据更新都已经保存到 DataBase 中，从而避免在连接对象数目调整后，由于相同属性应用流程的访问所对应的连接对象的改变而出现的污读现象。这里所述仅为调整连接对象数目时为避免数据库污读，系统所采取的一种实施方式，但并不排除有其它实施方式的可能，具体采取哪种实施方式本发明不进行限定 ；并且，本段所述处理非本发明重点，因此对于其它实施方式本文不再进一步详述，但并不影响本发明的有效性。

[0079] 综上所述，应用本发明方法能够在多个业务进程同时访问数据库时，使分配到各业务进程的具备逻辑关系的应用流程的访问使用同一连接对象，这样应用流程访问数据库时不必考虑其它有关应用流程对数据库的修改是否已执行 COMMIT 操作，并能够防止数据的污读，不仅显著简化了数据库管理系统的开发过程，而且使用户更加灵活的配置数据库

COMMIT 的频率，有效保证数据库系统的 I/O 性能。在此基础之上，本发明还增加了动态调整数据库连接池的连接对象数目的处理，从而有效利用数据库连接资源来提高服务质量。

[0080] 以上所述仅为本发明的较佳实施例而已，并非用于限定本发明的保护范围。凡在本发明的精神和原则之内所作的任何修改、等同替换、改进等，均包含在本发明的保护范围内。

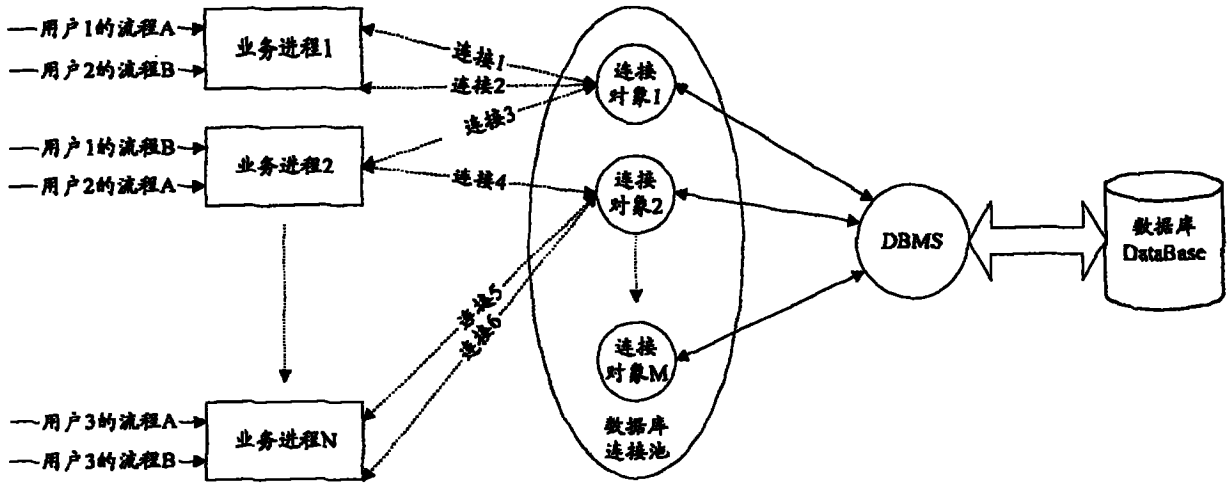

用户1的流程A
用户2的流程B
业务进程1
用户1的流程B
用户2的流程A
业务进程2
用户3的流程A
用户3的流程B
业务进程N
连接1
连接2
连接3
连接4
连接5
连接6
连接对象1
连接对象2
连接对象M
数据库连接池
DBMS
数据库DataBase

图 1

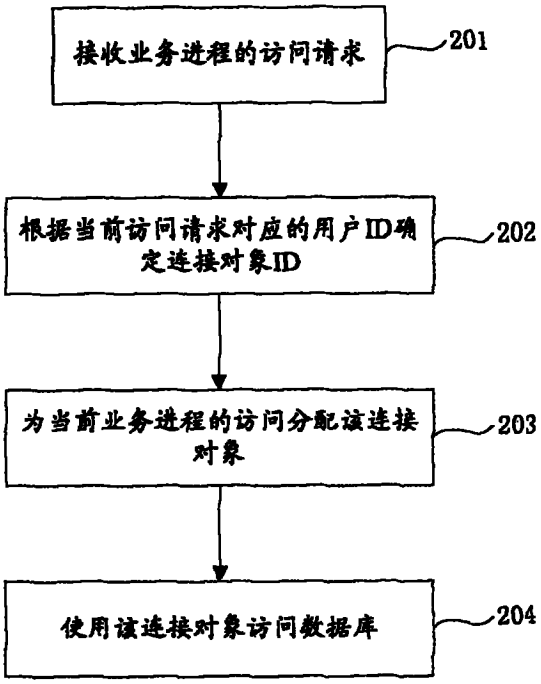

接收业务进程的访问请求
201
根据当前访问请求对应的用户ID确定连接对象ID
202
为当前业务进程的访问分配该连接对象
203
使用该连接对象访问数据库
204

图 2

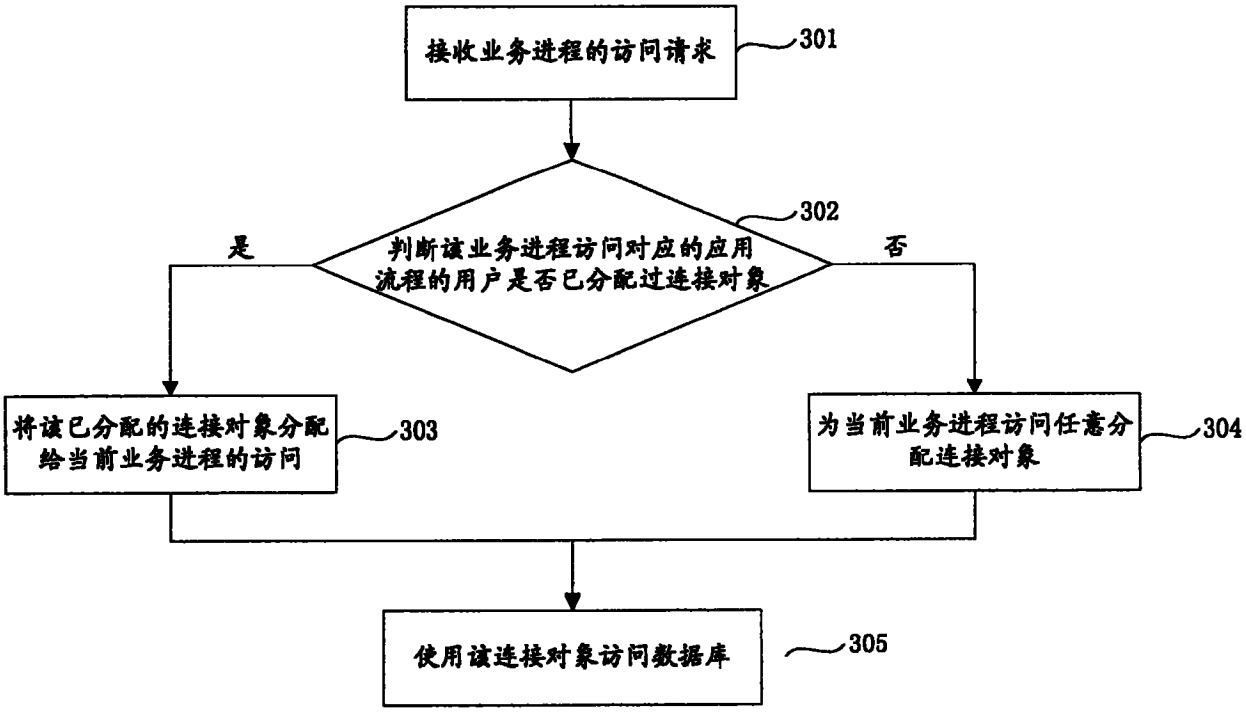

接收业务进程的访问请求
301
判断该业务进程访问对应的应用流程的用户是否已分配过连接对象
302
是
否
将该已分配的连接对象分配给当前业务进程的访问
303
为当前业务进程访问任意分配连接对象
304
使用该连接对象访问数据库
305

图 3

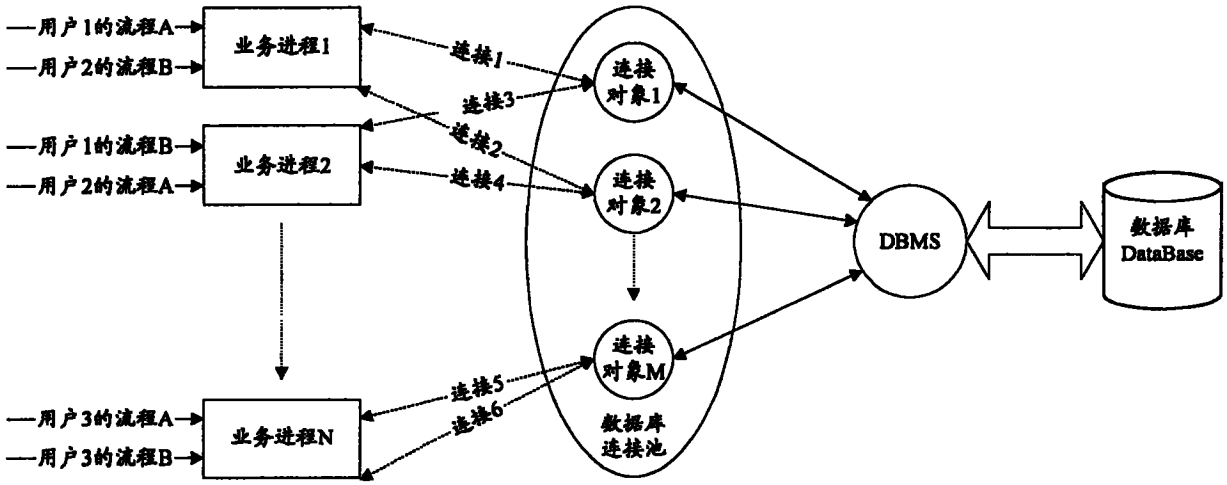

用户1的流程A
用户2的流程B
业务进程1
用户1的流程B
用户2的流程A
业务进程2
用户3的流程A
用户3的流程B
业务进程N
连接1
连接3
连接2
连接4
连接5
连接6
连接对象1
连接对象2
连接对象M
数据库连接池
DBMS
数据库DataBase

图 4

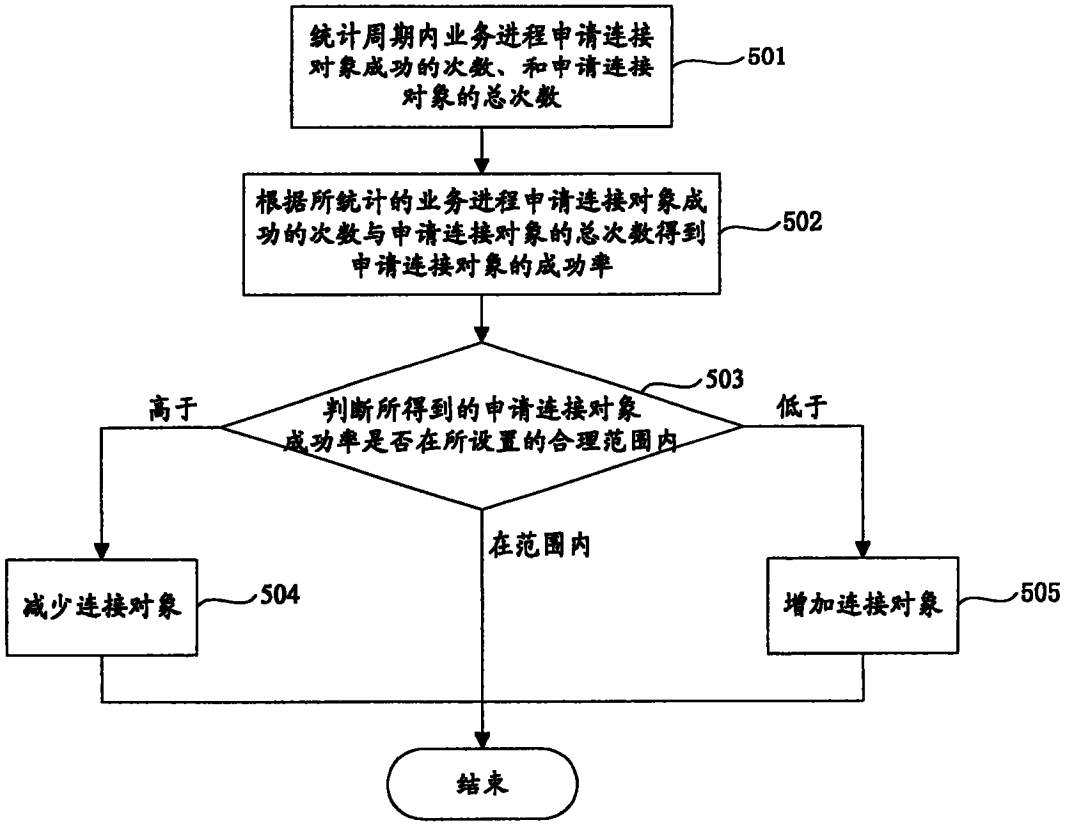

统计周期内业务进程申请连接对象成功的次数、和申请连接对象的总次数
501
根据所统计的业务进程申请连接对象成功的次数与申请连接对象的总次数得到申请连接对象的成功率
502
判断所得到的申请连接对象成功率是否在所设置的合理范围内
503
高于
低于
在范围内
减少连接对象
504
增加连接对象
505
结束

图 5

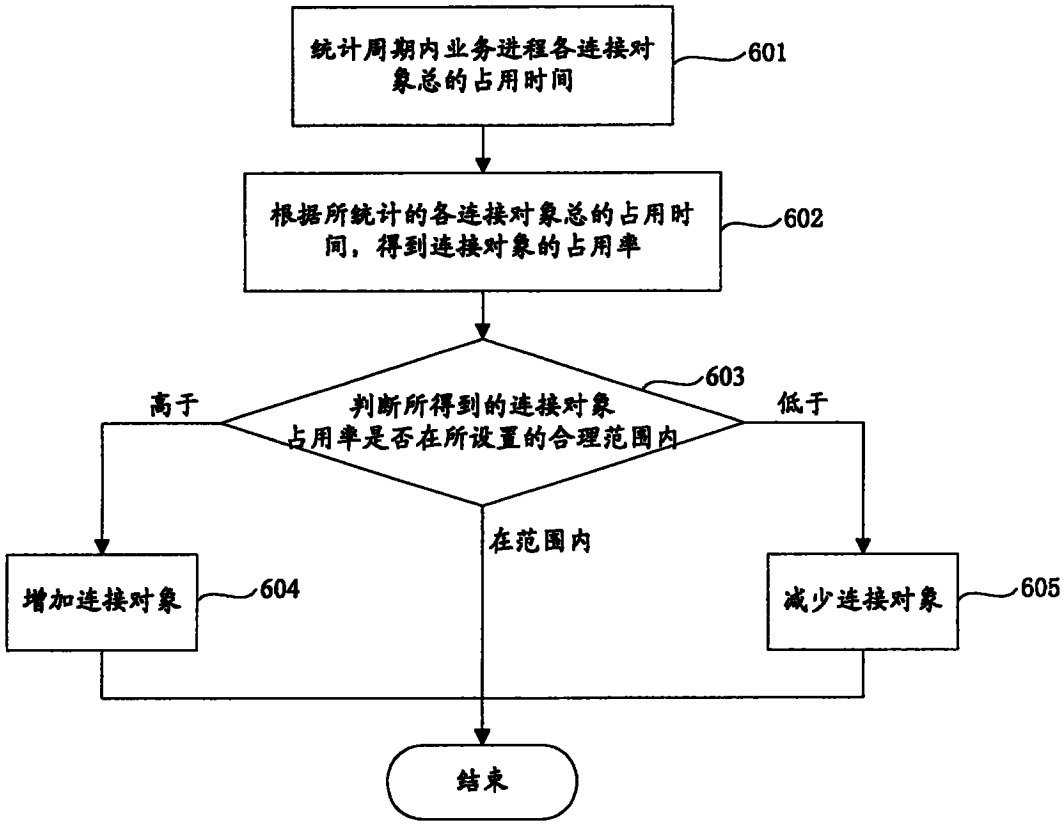

统计周期内业务进程各连接对象总的占用时间
601
根据所统计的各连接对象总的占用时间，得到连接对象的占用率
602
判断所得到的连接对象占用率是否在所设置的合理范围内
603
高于
低于
在范围内
增加连接对象
604
减少连接对象
605
结束

图 6