



(12) 发明专利申请

(10) 申请公布号 CN 112579307 A

(43) 申请公布日 2021.03.30

(21) 申请号 202011454282.4

(22) 申请日 2020.12.10

(71) 申请人 腾讯科技(深圳)有限公司

地址 518044 广东省深圳市南山区高新区
科技中一路腾讯大厦35层

(72) 发明人 周信静

(74) 专利代理机构 北京同达信恒知识产权代理
有限公司 11291

代理人 李娟

(51) Int. Cl.

G06F 9/52 (2006.01)

G06F 9/50 (2006.01)

G06F 16/21 (2019.01)

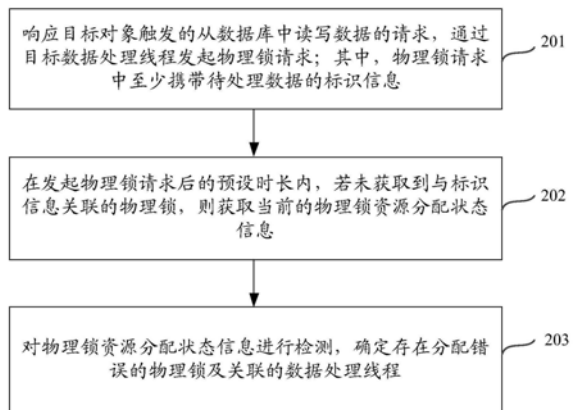
权利要求书2页 说明书13页 附图7页

(54) 发明名称

一种物理锁资源的分配检测方法、装置及电子设备

(57) 摘要

本申请提供一种物理锁资源的分配检测方法、装置及电子设备,涉及云技术领域。本申请实施中,响应目标对象触发的从数据库中读写数据的请求,通过目标数据处理线程发起物理锁请求;物理锁请求中至少携带待处理数据的标识信息;在发起物理锁请求后的预设时长内,若未获取到与标识信息关联的物理锁,则获取当前的物理锁资源分配状态信息;对物理锁资源分配状态信息进行检测,确定存在分配错误的物理锁及关联的数据处理线程。本申请实施例提供一种针对数据库中物理锁资源分配的有效检测方案,可以用于对物理锁资源分配状态信息进行检测,从而在确定存在分配错误的物理锁及关联的数据处理线程后,可以及时修正,提高了数据库中物理锁资源分配的准确性。



1. 一种物理锁资源的分配检测方法,其特征在于,包括:

响应目标对象触发的从数据库中读写数据的请求,通过目标数据处理线程发起物理锁请求;其中,所述物理锁请求中至少携带待处理数据的标识信息;

在发起物理锁请求后的预设时长内,若未获取到与所述标识信息关联的物理锁,则获取当前的物理锁资源分配状态信息;

对所述物理锁资源分配状态信息进行检测,确定存在分配错误的物理锁及关联的数据处理线程。

2. 如权利要求1所述的方法,其特征在于,在获取当前的物理锁资源分配状态信息之前,还包括:

获取在所述数据库中执行数据读写操作的各个数据处理线程;

根据所述各个数据处理线程请求的物理锁以及所述各个数据处理线程占用的物理锁,生成所述物理锁资源分配状态信息。

3. 如权利要求2所述的方法,其特征在于,所述物理锁资源分配状态信息为物理锁资源分配图;

所述根据所述各个数据处理线程请求的物理锁以及所述各个数据处理线程占用的物理锁,生成所述物理锁资源分配状态信息,具体包括:

将所述各个数据处理线程、所述各个数据处理线程请求的物理锁以及所述各个数据处理线程占用的物理锁作为节点;

将多个节点中数据处理线程与请求的物理锁之间连线作为边,以及将多个节点中数据处理线程与占用的物理锁之间连线作为边;其中,数据处理线程与请求的物理锁之间的边,和数据处理线程与占用的物理锁之间的边的方向不同;

根据确定出的节点和边生成物理锁资源分配图。

4. 如权利要求2所述的方法,其特征在于,若所述数据处理线程为处理数据库引擎层中读写数据请求的第一数据处理线程;

在所述根据所述各个数据处理线程请求的物理锁以及所述各个数据处理线程占用的物理锁,生成所述物理锁资源分配状态信息之前,还包括:

从第一对应关系集合中获取各个第一数据处理线程请求的物理锁;其中,所述第一对应关系集合中存储有发起物理锁请求的第一数据处理线程与请求的物理锁之间的第一对应关系,所述第一对应关系为所述第一数据处理线程发起物理锁请求后生成并存储在所述第一对应关系集合中的;

从第二对应关系集合中获取各个第一数据处理线程占用的物理锁;其中,所述第二对应关系集合中存储有第一数据处理线程与占用的物理锁之间的第二对应关系,所述第二对应关系为所述第一对应关系集合中的第一数据处理线程请求的物理锁成功占用后生成并存储在所述第二对应关系集合中的。

5. 如权利要求2所述的方法,其特征在于,若所述数据处理线程为处理数据库服务层中读写数据请求的第二数据处理线程;

在所述根据所述各个数据处理线程请求的物理锁以及所述各个数据处理线程占用的物理锁,生成所述物理锁资源分配状态信息之前,还包括:

遍历各个第二数据处理线程对应的容器数组;

根据各个第二数据处理线程对应的容器数组中存储的二元组信息,确定各个第二数据处理线程请求的物理锁以及各个第二数据处理线程占用的物理锁;其中,所述容器数组中存储的二元组信息包括第二数据处理线程与请求的物理锁、第二数据处理线程与占用的物理锁中的至少一种。

6.如权利要求3所述的方法,其特征在于,所述对所述物理锁资源分配状态信息进行检测,确定存在分配错误的物理锁及关联的数据处理线程,包括:

对所述物理锁资源分配图进行检测;

若根据所述物理锁资源分配图中各个节点以及节点之间连接的边的方向,从所述物理锁资源分配图中检测到按照边的方向将节点依次连接的节点环路;则将所述节点环路中包含的物理锁及数据处理线程确定为存在分配错误的物理锁及关联的数据处理线程。

7.一种物理锁资源的分配检测装置,其特征在于,包括:

请求单元,用于响应目标对象触发的从数据库中读写数据的请求,通过目标数据处理线程发起物理锁请求;其中,所述物理锁请求中至少携带待处理数据的标识信息;

获取单元,用于在发起物理锁请求后的预设时长内,若未获取到与所述标识信息关联的物理锁,则获取当前的物理锁资源分配状态信息;

检测单元,用于对所述物理锁资源分配状态信息进行检测,确定存在分配错误的物理锁及关联的数据处理线程。

8.如权利要求7所述的装置,其特征在于,在获取当前的物理锁资源分配状态信息之前,所述获取单元还用于:

获取在所述数据库中执行数据读写操作的各个数据处理线程;

根据所述各个数据处理线程请求的物理锁以及所述各个数据处理线程占用的物理锁,生成所述物理锁资源分配状态信息。

9.一种电子设备,其特征在于,其包括处理器和存储器,其中,所述存储器存储有程序代码,当所述程序代码被所述处理器执行时,使得所述处理器执行权利要求1~6中任一所述方法的步骤。

10.一种计算机可读存储介质,其特征在于,其包括程序代码,当所述程序代码在电子设备上运行时,所述程序代码用于使所述电子设备执行权利要求1~6中任一所述方法的步骤。

一种物理锁资源的分配检测方法、装置及电子设备

技术领域

[0001] 本申请涉及云技术领域,特别涉及一种物理锁资源的分配检测方法、装置及电子设备。

背景技术

[0002] 随着现在信息化建设进程的稳步推进,数据库技术得到了广泛应用。数据库就是存储数据的仓库,数据按照特定的格式将数据存储起来,用户可以通过数据库中的数据处理线程读写存储在数据库中的数据,而数据处理线程从数据库中读写数据时需要请求该数据对应的物理锁,用于保护该数据,防止其他用户对该数据进行修改或者删除,保证该数据的完整性。

[0003] 目前,在实际情况下,往往存在多个数据处理线程并发请求从数据库中读写数据,因而多个数据处理线程会分配不同的物理锁,此时物理锁资源分配可能会存在分配错误的问题,但是,目前针对数据库的物理锁分配情况并没有一种有效的检测方案。

发明内容

[0004] 本申请实施例提供一种物理锁资源的分配检测方法、装置及电子设备,公开一种有效检测数据库的物理锁资源分配状态的方案,提高了数据库中物理锁资源分配准确性。

[0005] 第一方面,本申请实施例提供一种物理锁资源的分配检测方法,包括:

[0006] 响应目标对象触发的从数据库中读写数据的请求,通过目标数据处理线程发起物理锁请求;其中,所述物理锁请求中至少携带待处理数据的标识信息;

[0007] 在发起物理锁请求后的预设时长内,若未获取到与所述标识信息关联的物理锁,则获取当前的物理锁资源分配状态信息;

[0008] 对所述物理锁资源分配状态信息进行检测,确定存在分配错误的物理锁及关联的数据处理线程。

[0009] 第二方面,本申请实施例提供一种物理锁资源的分配检测装置,包括:

[0010] 请求单元,用于响应目标对象触发的从数据库中读写数据的请求,通过目标数据处理线程发起物理锁请求;其中,所述物理锁请求中至少携带待处理数据的标识信息;

[0011] 获取单元,用于在发起物理锁请求后的预设时长内,若未获取到与所述标识信息关联的物理锁,则获取当前的物理锁资源分配状态信息;

[0012] 检测单元,用于对所述物理锁资源分配状态信息进行检测,确定存在分配错误的物理锁及关联的数据处理线程。

[0013] 第三方面,本申请实施例提供一种电子设备,包括:

[0014] 至少一个处理器;以及

[0015] 与所述至少一个处理器通信连接的存储器;其中,

[0016] 所述存储器存储有可被所述至少一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行本申请提供的物理锁资源的分配检测

方法。

[0017] 第四方面,本申请实施例提供一种计算机可读介质,存储有计算机可执行指令,所述计算机可执行指令用于执行本申请提供的物理锁资源的分配检测方法。

[0018] 本申请有益效果:

[0019] 本申请实施例响应目标对象触发的从数据库中读写数据的请求,通过目标数据处理线程发起物理锁请求;物理锁请求中至少携带待处理数据的标识信息;在发起物理锁请求后的预设时长内,若未获取到与标识信息关联的物理锁,则对当前的物理锁资源分配状态信息进行检测,确定存在分配错误的物理锁及关联的数据处理线程;本申请实施例在目标数据处理线程发起物理锁请求后对等待时长进行检测,由于在目标数据处理线程的等待时长大于预设时长时,说明目标数据处理线程请求的物理锁可能被占用,需要对当前物理锁资源分配状态进行检测,并且由于物理锁资源分配状态信息是实时更新的,在确定目标数据处理线程在发起物理锁请求后的预设时长内,若未获取到与标识信息关联的物理锁,则获取当前的物理锁资源分配状态信息,并对获取到的物理锁资源分配状态信息进行检测,判断是否存在分配错误的物理锁及关联的数据处理线程。因此,本申请实施例提供一种针对数据库中物理锁资源分配的有效检测方案,可以用于对物理锁资源分配状态信息进行检测,从而在确定存在分配错误的物理锁及关联的数据处理线程后,可以及时修正,提高了数据库中物理锁资源分配的准确性。

附图说明

- [0020] 图1为本申请实施例示例性的一种可选应用场景的示意图;
- [0021] 图2为本申请实施例提供的一种物理锁资源的分配检测方法流程示意图;
- [0022] 图3为本申请实施例提供的物理锁资源分配图示意图;
- [0023] 图4为本申请实施例提供的第一对应关系集合和第二对应关系集合示意图;
- [0024] 图5为本申请实施例提供的容器数组示意图;
- [0025] 图6为本申请实施例提供的一种死锁环路示例图;
- [0026] 图7为本申请实施例提供的一种非死锁环路示例图;
- [0027] 图8为本申请实施例提供的从物理锁资源分配图中检测到的死锁环路的示例图;
- [0028] 图9为本申请实施例提供的第一种死锁环路示意图;
- [0029] 图10为本申请实施例提供的第二种死锁环路示意图;
- [0030] 图11为本申请实施例提供的第三种死锁环路示意图;
- [0031] 图12本申请实施例提供的一种物理锁资源的分配检测装置的结构示意图;
- [0032] 图13为本申请实施例中的电子设备的结构示意图;
- [0033] 图14为本申请实施例中的一种计算装置的结构示意图。

具体实施方式

[0034] 为了使本领域普通人员更好地理解本申请公开的技术方案,下面将结合附图,对本申请实施例中的技术方案进行清楚、完整地描述。

[0035] 需要说明的是,本申请的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。应该理解这样使用

的数据在适当情况下可以互换,以便这里描述的本申请的实施例能够以除了在这里图示或描述的那些以外的顺序实施。以下示例性实施例中所述的方式并不代表与本申请相一致的所有实施方式。相反,它们仅是与如所附权利要求书中所详述的、本申请的一些方面相一致的装置和方法的例子。

[0036] 下面对文中出现的一些术语进行解释:

[0037] 1、MySQL数据库:MySQL是一种开放源代码的关系型数据库管理系统(Relational Database Management System,RDBMS),使用最常用的数据库管理语言--结构化查询语言(Structured Query Language,SQL)进行数据库管理。MySQL是开放源代码的,因此任何人都可以在通用公共许可证(General Public License,GPL)的许可下下载并根据个性化的需要对其进行修改。MySQL因为其速度、可靠性和适应性而备受关注。大多数人都认为在不需事务化处理的情况下,MySQL是管理内容最好的选择。

[0038] 2、MySQL的服务(Server)层:MySQL数据库负责处理SQL语法解析、SQL语句优化、SQL语句执行、MySQL协议等功能的组件,它对接了多个用于管理数据的存储引擎。

[0039] 3、MySQL的存储引擎层:负责MySQL中的数据存储和提取。和Linux下的文件系统类似,每种存储引擎都有其优势和劣势。中间的服务层通过API与存储引擎通信,这些API接口屏蔽了不同存储引擎间的差异,使得这些差异对上层的查询过程透明。存储引擎API包含几十个底层函数,用于执行诸如“开始一个事务”或者“根据主键提取一行记录”等操作。但存储引擎不会去解析SQL(InnoDB除外,它会解析外键定义,因为MySQL服务器本身没有实现该功能),不同的存储引擎之间也不会相互通信,而只是简单地相应上层服务器的请求。

[0040] 4、物理锁:保护并发线程对关键代码区域的执行、数据结构的修改不会造成不一致,一般采用操作系统提供的锁原语来实现,如Linux pthread_mutex_t相关API。

[0041] 5、物理锁资源分配图:有向图,有两类节点:<线程>、<物理锁>,分别代表了系统中存在的线程和物理锁资源。有两类边:[请求]、[被占用]。其中[请求]边的起点是<线程>节点,终点是<物理锁>,表示一个线程正在请求获得某个物理锁;[被占用]边的起点是<物理锁>节点,终点是<线程>节点,表示某个物理锁资源被某个线程所占用。

[0042] 6、物理锁死锁环路:在数据库系统中,物理锁资源分配图中包括多个数据处理线程和物理锁,若检测到物理锁资源分配图中存在一个首尾相连的环路,则将该环路称为物理锁死锁环路,物理锁死锁环路中包含的物理锁及数据处理线程为分配错误的物理锁及数据处理线程。

[0043] 7、终端:又称为用户设备(User Equipment,UE)、移动台(Mobile Station,MS)、移动终端(Mobile Terminal,MT)等,是一种向用户提供语音和/或数据连通性的设备,例如,具有无线连接功能的手持式设备、车载设备等。目前,一些终端的举例为:手机(mobile phone)、平板电脑、笔记本电脑、掌上电脑、移动互联网设备(Mobile Internet Device,MID)。

[0044] 8、客户端:既可以指软件类的APP(Application,应用程序),也可以指终端设备。它具有可视的显示界面,能与用户进行交互;是与服务器相对应,为客户提供本地服务。针对软件类的应用程序,除了一些只在本地运行的应用程序之外,一般安装在普通的客户终端上,需要与服务端互相配合运行。因特网发展以后,较常用的应用程序包括了如收寄电子邮件时的电子邮件客户端,以及即时通讯的客户端等。对于这一类应用程序,需要网络中有

相应的服务器和服务程序来提供相应的服务,如数据库服务,配置参数服务等,这样在客户终端和服务端,需要建立特定的通信连接,来保证应用程序的正常运行。

[0045] 下面对本申请实施例的设计思想进行简要介绍:

[0046] 云技术(Cloud technology)是指在广域网或局域网内将硬件、软件、网络等系列资源统一起来,实现数据的计算、储存、处理和共享的一种托管技术。

[0047] 云技术(Cloud technology)基于云计算商业模式应用的网络技术、信息技术、整合技术、管理平台技术、应用技术等的总称,可以组成资源池,按需所用,灵活便利。云计算技术将变成重要支撑。技术网络系统的后台服务需要大量的计算、存储资源,如视频网站、图片类网站和更多的门户网站。伴随着互联网行业的高度发展和应用,将来每个物品都有可能存在自己的识别标志,都需要传输到后台系统进行逻辑处理,不同程度级别的数据将会分开处理,各类行业数据皆需要强大的系统后盾支撑,只能通过云计算来实现。

[0048] 数据库(Database),简而言之可视为电子化的文件柜——存储电子文件的处所,用户可以对文件中的数据进行新增、查询、更新、删除等操作。所谓“数据库”是以一定方式储存在一起、能与多个用户共享、具有尽可能小的冗余度、与应用程序彼此独立的数据集合。

[0049] 数据库管理系统(Database Management System,DBMS)是为管理数据库而设计的电脑软件系统,一般具有存储、截取、安全保障、备份等基础功能。数据库管理系统可以依据它所支持的数据库模型来作分类,例如关系式、可扩展标记语言(Extensible Markup Language,XML);或依据所支持的计算机类型来作分类,例如服务器群集、移动电话;或依据所用查询语言来作分类,例如结构化查询语言(Structured Query Language,SQL)、XQuery;或依据性能冲量重点来作分类,例如最大规模、最高运行速度;亦其他的分类方式。不论使用哪种分类方式,一些DBMS能够跨类别,例如,同时支持多种查询语言。

[0050] 目前,用户可以通过数据处理线程读写数据库中存储的数据,一般的,在实际情况下,往往存在多个数据处理线程并发请求从数据库中读写数据,则此时需要为多个数据处理线程分配不同的物理锁,那么可能会遇到物理锁资源分配错误的问题,然而,目前针对数据库的物理锁分配情况并没有一种有效的检测方案。

[0051] 本申请实施例响应目标对象触发的从数据库中读写数据的请求,通过目标数据处理线程发起物理锁请求;物理锁请求中至少携带待处理数据的标识信息;在发起物理锁请求后的预设时长内,若未获取到与标识信息关联的物理锁,则对当前的物理锁资源分配状态信息进行检测,确定存在分配错误的物理锁及关联的数据处理线程;本申请实施例在目标数据处理线程发起物理锁请求后对等待时长进行检测,由于在目标数据处理线程的等待时长大于预设时长时,说明目标数据处理线程请求的物理锁可能被占用,需要对当前物理锁资源分配状态进行检测,并且由于物理锁资源分配状态信息是实时更新的,在确定目标数据处理线程在发起物理锁请求后的预设时长内,若未获取到与标识信息关联的物理锁,则获取当前的物理锁资源分配状态信息,并对获取到的物理锁资源分配状态信息进行检测,判断是否存在分配错误的物理锁及关联的数据处理线程。因此,本申请实施例提供一种针对数据库中物理锁资源分配的有效检测方案,可以用于对物理锁资源分配状态信息进行检测,从而在确定存在分配错误的物理锁及关联的数据处理线程后,可以及时修正,提高了数据库中物理锁资源分配的准确性。

[0052] 在介绍完本申请实施例的设计思想之后,下面对本申请实施例的技术方案能够适用的应用场景做一些简单介绍,需要说明的是,以下介绍的应用场景仅用于说明本申请实施例而非限定。在具体实施过程中,可以根据实际需要灵活地应用本申请实施例提供的技术方案。

[0053] 如图1所示,其为本申请实施例示例性的一种可选应用场景的示意图,包括目标对象10、终端设备11、远程服务器12;其中,目标对象10可以操作终端设备11上安装的客户端访问位于远程服务器12中的数据库,从远程服务器12中的数据库中读写数据。

[0054] 目标对象10操作终端设备11上安装的客户端,在客户端运行过程中,根据目标对象10在客户端中的操作,若该客户端需要从远程服务器12的数据库中读写数据,则客户端向远程服务器12的数据库发送读写数据的请求;远程服务器12为客户端发起的读写数据请求分配目标数据处理线程,目标数据处理线程发起物理锁请求;其中,物理锁请求中至少携带待处理数据的标识信息;若远程服务器12确定在目标数据处理线程发起物理锁请求后的预设时长内,目标数据处理线程未获取到与标识信息关联的物理锁,则获取当前的物理锁资源分配状态信息,远程服务器12对物理锁资源分配状态信息进行检测,确定存在分配错误的物理锁及关联的数据处理线程。

[0055] 需要说明的是,数据库可以为终端设备11中运行的客户端对应的服务器中的数据库。

[0056] 下面结合上述描述的应用场景,参考图2-图6来描述本申请示例性实施方式提供了一种物理锁资源的分配检测方法。需要注意的是,上述应用场景仅是为了便于理解本申请的精神和原理而示出,本申请的实施方式在此方面不受任何限制。相反,本申请的实施方式可以应用于适用的任何场景。

[0057] 如图2所示,其为本申请实施例提供的物理锁资源的分配检测方法流程示意图,该方法可以包括以下步骤:

[0058] 步骤S201、响应目标对象触发的从数据库中读写数据的请求,通过目标数据处理线程发起物理锁请求;其中,物理锁请求中至少携带待处理数据的标识信息;

[0059] 步骤S202、在发起物理锁请求后的预设时长内,若未获取到与标识信息关联的物理锁,则获取当前的物理锁资源分配状态信息;

[0060] 步骤S203、对物理锁资源分配状态信息进行检测,确定存在分配错误的物理锁及关联的数据处理线程。

[0061] 本申请实施例目标对象触发从数据库中读写数据的请求时,一种可选的应用场景为,例如,目标对象可以为公司管理人员,目标对象通过终端设备上运行的客户端访问公司服务器,从公司服务器的数据库中读取公司员工薪资数据A;目标对象触发从数据库中读写数据的请求,公司服务器分配目标数据处理线程T执行数据读写操作,目标数据处理线程T发起物理锁请求,请求物理锁L对数据A进行加锁处理,用于对数据A进行保护,防止其他对象进行修改或者删除,保证数据A的完整性。

[0062] 在通过目标数据处理线程发起物理锁请求后,一种可选的实施方式为,本申请实施例可以利用物理锁加锁等待超时检测技术,判断目标数据处理线程在发起请求物理锁后的等待时长是否超过预设时长;例如,目标数据处理线程T发起请求物理锁L的请求,此时利用物理锁加锁等待超时检测技术,监测目标数据处理线程T,若监测到目标数据处理线程T

在发起请求物理锁L后的预设时长(假设预设时长为10s)内,并没有获取到物理锁L,则确定目标数据处理线程T在发起请求物理锁L后的等待时长超过预设时长。

[0063] 一种可选的实施方式为,本申请实施例可以根据下列方式生成物理锁资源分配状态信息:

[0064] 获取在数据库中执行数据读写操作的各个数据处理线程;根据各个数据处理线程请求的物理锁以及各个数据处理线程占用的物理锁,生成物理锁资源分配状态信息。

[0065] 需要说明的是,在数据库系统中执行数据读写操作的各个数据处理线程请求的物理锁以及占用的物理锁是实时更新的,物理锁资源分配状态信息根据在数据库中执行读写操作的各个数据处理线程请求或占用的物理锁实时动态生成。

[0066] 在一些实施例中,物理锁资源分配状态信息可以为物理锁资源分配图。

[0067] 一种可选的实施方式为,本申请实施例可以根据下列方式生成物理锁资源分配图:

[0068] 将各个数据处理线程、各个数据处理线程请求的物理锁以及各个数据处理线程占用的物理锁作为节点,并将多个节点中数据处理线程与请求的物理锁之间连线作为边,以及将多个节点中数据处理线程与占用的物理锁之间连线作为边;

[0069] 其中,数据处理线程与请求的物理锁之间的边,和数据处理线程与占用的物理锁之间的边方向不同;

[0070] 一种可选的实施方式为,本申请实施例可以将数据处理线程到请求的物理锁的方向作为用于连接数据处理线程与请求的物理锁之间的边的方向,将数据处理线程占用的物理锁到数据处理线程的方向作为用于连接数据处理线程与占用的物理锁之间的边的方向。

[0071] 实施中,将数据处理线程与请求的物理锁之间的边以<线程T>-[请求]-><物理锁L>这种边的形式体现于物理锁资源分配图中,将数据处理线程占用的物理锁之间的边以<物理锁L>-<被占用>-><线程T>这种边的形式体现于物理锁资源分配图中。

[0072] 需要说明的是,在数据库中,物理锁资源分配图是有向图,物理锁资源分配图根据在数据库中执行读写操作的各个数据处理线程请求或占用的物理锁实时动态生成;且整个数据库中处理不同层数据的数据处理线程及请求或占用的物理锁资源分配情况可以集成于同一个物理锁资源分配图。

[0073] 例如,如图3所示为本申请实施例提供的物理锁资源分配图示意图,处理数据库服务层的数据处理线程以线程 T_1 、线程 T_2 、线程 T_3 为例,处理数据库引擎层的数据处理线程以线程 T_4 、线程 T_5 、线程 T_6 为例,线程 T_1 请求物理锁 L_1 、线程 T_2 占用物理锁 L_1 、线程 T_2 请求物理锁 L_3 、线程 T_3 占用物理锁 L_3 、线程 T_3 请求物理锁 L_4 、线程 T_2 占用物理锁 L_4 、线程 T_4 请求物理锁 L_7 、线程 T_6 占用物理锁 L_7 、线程 T_6 请求物理锁 L_6 、线程 T_5 占用物理锁 L_6 、线程 T_5 请求物理锁 L_2 、线程 T_3 占用物理锁 L_2 ;则上述处理数据库服务层与引擎层数据的数据处理线程请求与占用的物理锁以下列边的形式,集成于同一个物理锁资源分配图:<线程 T_1 >-[请求]-><物理锁 L_1 >、<物理锁 L_1 >-[被占用]-><线程 T_2 >、<线程 T_2 >-[请求]-><物理锁 L_3 >、<物理锁 L_3 >-[被占用]-><线程 T_3 >、<线程 T_3 >-[请求]-><物理锁 L_4 >、<物理锁 L_4 >-[被占用]-><线程 T_2 >、<线程 T_4 >-[请求]-><物理锁 L_7 >、<物理锁 L_7 >-[被占用]-><线程 T_6 >、<线程 T_6 >-[请求]-><物理锁 L_6 >、<物理锁 L_6 >-[被占用]-><线程 T_5 >、<线程 T_5 >-[请求]-><物理锁 L_2 >、<物理锁 L_2 >-[被占用]-><线程 T_3 >。

[0074] 本申请实施例在生成物理锁资源分配状态信息之前,需要获取各个数据处理线程请求的物理锁以及各个数据处理线程占用的物理锁;本申请实施例中针对处理数据库不同层的数据处理线程,获取数据处理线程请求和占用的物理锁的方式不同,下面针对不同类型的数据处理线程,获取数据处理线程请求和占用的物理锁的方式分别进行说明。

[0075] 1、数据处理线程为处理数据库引擎层中读写数据请求的第一数据处理线程;

[0076] 从第一对应关系集合中获取各个第一数据处理线程请求的物理锁;

[0077] 其中,第一对应关系集合中存储有发起物理锁请求的第一数据处理线程与请求的物理锁之间的第一对应关系,第一对应关系为第一数据处理线程发起物理锁请求后生成并存储在第一对应关系集合中的。

[0078] 需要说明的是,在数据库中,当一个数据处理线程读写数据时,且无法立刻获取到该数据对应的物理锁,则该数据处理线程会生成获取该物理锁的物理锁请求。

[0079] 从第二对应关系集合中获取各个第一数据处理线程占用的物理锁;

[0080] 其中,第二对应关系集合中存储有第一数据处理线程与占用的物理锁之间的第二对应关系,第二对应关系为第一对应关系集合中的第一数据处理线程请求的物理锁成功占用后生成并存储在第二对应关系集合中的。

[0081] 需要说明的是,在第一对应关系集合中的第一数据处理线程请求的物理锁成功占用后并存储在第二对应关系集合之前,需要删除第一对应关系集合中存储的发起物理锁请求的第一数据处理线程与请求的物理锁之间的第一对应关系。

[0082] 在物理锁资源分配状态信息为物理锁资源分配图时,第一对应关系集合中的数据处理线程与请求的物理锁之间的对应关系,以及第二对应关系集合中的数据处理线程与占用的物理锁之间的对应关系可以以边的形式进行存储,便于在生成物理锁资源分配图时,根据第一对应关系集合和第二对应关系集合生成物理锁资源分配图;其中,第一对应关系集合中的数据处理线程与请求的物理锁之间的对应关系可以表示为<线程T>-[请求]-><物理锁L>,第二对应关系集合中的数据处理线程与占用的物理锁之间的对应关系可以表示为<物理锁L>-<被占用>-><线程T>。

[0083] 例如,处理数据库引擎层中数据的数据处理线程以线程 T_1 、线程 T_2 、线程 T_3 为例,线程 T_1 请求物理锁 L_1 、线程 T_2 成功占用物理锁 L_1 以及 L_2 、线程 T_2 请求物理锁 L_3 、线程 T_3 成功占用物理锁 L_3 、线程 T_3 请求物理锁 L_4 ,则第一对应关系集合和第二对应关系集合如图4所示;其中,第一对应关系集合中以<线程T>-[请求]-><物理锁L>形式表示第一对应关系,第二对应关系集合中以<物理锁L>-<被占用>-><线程T>形式表示第二对应关系;

[0084] 线程 T_1 请求物理锁 L_1 ,此时发起请求物理锁 L_1 的线程 T_1 与请求的物理锁 L_1 生成第一对应关系并以<线程 T_1L_1 >的形式存储在第一对应关系集合中;

[0085] 线程 T_2 成功占用物理锁 L_1 以及 L_2 ,且线程 T_2 请求物理锁 L_3 ,此时发起请求物理锁 L_3 的线程 T_2 与请求的物理锁 L_3 生成的第一对应关系以<线程 T_2L_3 >的形式存储在第一对应关系集合中,线程 T_2 与占用的物理锁 L_1 之间生成第二对应关系并以<物理锁 L_1 >-<被占用>-><线程 T_2 >形式存储在第二对应关系集合中,线程 T_2 与占用的物理锁 L_2 之间生成第二对应关系并以<物理锁 L_2 >-<被占用>-><线程 T_2 >形式存储在第二对应关系集合中;

[0086] 线程 T_3 成功占用物理锁 L_3 ,且线程 T_3 请求物理锁 L_4 ,此时发起请求物理锁 L_4 的线程 T_3 与请求的物理锁 L_4 之间生成的第一对应关系以<线程 T_3L_4 >的形式存

储在第一对应关系集合中,线程 T_3 与占用的物理锁 L_3 之间生成第二对应关系并以 \langle 物理锁 L_3 \rangle - \langle 被占用 \rangle - \langle 线程 T_3 \rangle 形式存储在第二对应关系集合中。

[0087] 2、数据处理线程为处理数据库服务层中读写数据请求的第二数据处理线程;

[0088] 遍历各个第二数据处理线程对应的容器数组;

[0089] 根据各个第二数据处理线程对应的容器数组中存储的二元组信息,确定各个第二数据处理线程请求的物理锁以及各个第二数据处理线程占用的物理锁;其中,容器数组中存储的二元组信息包括第二数据处理线程与请求的物理锁、第二数据处理线程与占用的物理锁中的至少一种。

[0090] 一种可选的实施方式为,各个第二数据处理线程对应的容器数组中存储的二元组信息可以表示为 \langle ID(L),Status \rangle ,其中ID(L)可以表示为第二数据处理线程请求的物理锁L的ID,Status表示锁请求状态,取值为Requesting(请求)或Owned(占用);

[0091] 需要说明的是,对于各个第二数据处理线程请求物理锁L的加锁请求,以 \langle L,Requesting \rangle 形式存储于容器数组;当物理锁L被成功占用后,此时存储于容器数组中的 \langle L,Requesting \rangle 替换成 \langle L,Owned \rangle 。

[0092] 例如,处理数据库服务层中数据的数据处理线程以线程 T_1 、线程 T_2 、线程 T_3 为例,线程 T_1 请求物理锁 L_1 、线程 T_2 成功占用物理锁 L_1 以及 L_2 、线程 T_2 请求物理锁 L_3 、线程 T_3 成功占用物理锁 L_3 、线程 T_3 请求物理锁 L_4 ,则本申请实施例提供的容器数组示意图如图5所示;

[0093] 在数据库服务层中,针对每一个线程都有对应的容器数组C,例如,线程 T_1 对应的容器数组为 C_1 ,线程 T_2 对应的容器数组为 C_2 ,线程 T_3 对应的容器数组为 C_3 ,线程 T_4 对应的容器数组为 C_4 ;

[0094] 线程 T_1 请求物理锁 L_1 ,此时线程 T_1 对应的容器数组 C_1 中存储的二元组信息为 \langle L_1 ,Requesting \rangle ;

[0095] 线程 T_2 成功占用物理锁 L_1 以及 L_2 ,且线程 T_2 请求物理锁 L_3 ,此时线程 T_2 对应的容器数组 C_2 中存储的二元组信息为 \langle L_1 ,Owned \rangle 、 \langle L_2 ,Owned \rangle 以及 \langle L_3 ,Requesting \rangle ;

[0096] 线程 T_3 成功占用物理锁 L_3 ,且线程 T_3 请求物理锁 L_4 ,此时线程 T_3 对应的容器数组 C_3 中存储的二元组信息为 \langle L_3 ,Owned \rangle 和 \langle L_4 ,Requesting \rangle 。

[0097] 需要说明的是,处理数据库中引擎层与服务层中数据的各个数据处理线程可以占用多个物理锁,当某一数据处理线程占用某一个物理锁后,还可以请求其它物理锁;

[0098] 其中,数据处理线程请求哪一个物理锁,是由数据处理线程读写的数据决定的。

[0099] 本申请实施例在数据处理线程发起物理锁请求后的预设时长内,若未获取到该物理锁,则触发对物理锁资源分配状态信息进行检测,来确定存在分配错误的物理锁及关联的数据处理线程,下述实施例中物理锁资源分配状态信息均以物理锁资源分配图为例;

[0100] 一种可选的实施方式为,本申请实施例可以根据下列方式对物理锁资源分配图进行检测;

[0101] 若根据物理锁资源分配图中各个节点以及节点之间连接的边的方向,从物理锁资源分配图中检测到按照边的方向将节点依次连接的节点环路,则将节点环路中包含的物理锁及数据处理线程确定为存在分配错误的物理锁及关联的数据处理线程。

[0102] 其中,在数据库中物理锁及数据处理线程存在分配错误时,较常见的是当从物理锁分配图中检测到按照边的方向将节点依次连接的节点环路为死锁环路,此时死锁环路中

的物理锁及数据处理线程存在分配错误的问题。

[0103] 下述实施例中存在物理锁及数据处理线程分配错误的节点环路均以死锁环路为例；

[0104] 需要说明的是，从物理锁资源分配图中检测到的死锁环路可以为一条或多条，且每条死锁环路中可以包括多个数据处理线程和物理锁。

[0105] 若物理锁资源分配图中存在一个首尾相连的环路，且首尾为同一数据处理线程，则认为发生死锁，此时检测到的环路为死锁环路。

[0106] 例如，如图6所示，为本申请实施例提供的一种死锁环路示例图，该死锁环路有环路且包含起点，死锁环路依次包括<线程 T_1 >-[请求]-><物理锁 L_1 >，<物理锁 L_1 >-[被占用]-><线程 T_2 >，<线程 T_2 >-[请求]-><物理锁 L_3 >，<物理锁 L_3 >-[被占用]-><线程 T_3 >，<线程 T_3 >-[请求]-><物理锁 L_4 >，<物理锁 L_4 >-[被占用]-><线程 T_1 >这些边，且从图6中可以看出该死锁环路的起始节点和终止节点都为线程 T_1 ；

[0107] 又例如，如图7所示，为本申请实施例提供的一种非死锁环路示例图，该示例图有不包含环路，且依次包括<线程 T_1 >-[请求]-><物理锁 L_1 >，<物理锁 L_1 >-[被占用]-><线程 T_2 >，<线程 T_2 >-[请求]-><物理锁 L_3 >，<物理锁 L_3 >-[被占用]-><线程 T_3 >这些边；

[0108] 如图8所示为本申请实施例从物理锁资源分配图中检测到的死锁环路的示例图，若此时生成的物理锁资源分配图包括下列这些边：<线程 T_1 >-[请求]-><物理锁 L_1 >、<物理锁 L_1 >-[被占用]-><线程 T_2 >、<线程 T_2 >-[请求]-><物理锁 L_3 >、<物理锁 L_3 >-[被占用]-><线程 T_3 >、<线程 T_3 >-[请求]-><物理锁 L_2 >、<物理锁 L_2 >-[被占用]-><线程 T_6 >、<线程 T_6 >-[请求]-><物理锁 L_4 >、<物理锁 L_4 >-[被占用]-><线程 T_4 >、<线程 T_4 >-[请求]-><物理锁 L_6 >、<物理锁 L_6 >-[被占用]-><线程 T_5 >、<线程 T_5 >-[请求]-><物理锁 L_5 >、<物理锁 L_5 >-[被占用]-><线程 T_6 >；从图8中可以看出，从物理锁资源分配图检测到一条死锁环路：<线程 T_6 >-[请求]-><物理锁 L_4 >-><物理锁 L_4 >-[被占用]-><线程 T_4 >-><线程 T_4 >-[请求]-><物理锁 L_6 >-><物理锁 L_6 >-[被占用]-><线程 T_5 >-><线程 T_5 >-[请求]-><物理锁 L_5 >-><物理锁 L_5 >-[被占用]-><线程 T_6 >，且死锁环路的起始节点和终止节点都为线程 T_6 。

[0109] 具体实施中，从物理锁资源分配图中检测到的死锁环路中的物理锁可能存在多种情况，例如，分别为物理锁都为处理数据库引擎层数据的物理锁、处理物理锁都为数据库服务层数据的物理锁、物理锁包括处理数据库引擎层数据的物理锁以及处理数据库服务层数据的物理锁；

[0110] 下面针对上述3种情况分别进行说明。

[0111] 1、死锁环路中的物理锁均为处理数据库引擎层数据的物理锁。

[0112] 如图9所示，为本申请实施例提供的第一种死锁环路示意图，其中，处理数据库引擎层数据的数据处理线程以线程 T_1 、线程 T_2 、线程 T_3 为例，线程 T_1 请求物理锁 L_3 、线程 T_2 占用物理锁 L_3 、线程 T_2 请求物理锁 L_1 、线程 T_3 占用物理锁 L_1 、线程 T_3 请求物理锁 L_2 、线程 T_1 占用物理锁 L_2 ；

[0113] 线程 T_1 响应目标对象触发的从数据库中读写数据A的请求，其中数据A对应的物理锁为 L_3 ，则线程 T_1 生成请求物理锁 L_3 的加锁请求，当监测到线程 T_1 在发起物理锁请求后的预设时长内未获取到物理锁 L_3 ，则获取当前的物理锁资源分配图，启动死锁检测，以线程 T_1 为起点，沿着<线程 T_1 >-[请求]-><物理锁 L_3 >的边，开始对当前获取的物理锁资源分配图进行

检测,在检测过程中发现物理锁 L_3 被线程 T_2 占用,而此时线程 T_2 正在请求获取物理锁 L_1 ,此时物理锁 L_1 被线程 T_3 占用,线程 T_3 正在请求获取物理锁 L_2 ,而物理锁 L_2 被线程 T_1 占用,则此时形成一个死锁环路。

[0114] 2、死锁环路中的物理锁都为处理数据库服务层数据的物理锁。

[0115] 如图10所示,为本申请实施例提供的第二种死锁环路示意图,其中,处理数据库服务层数据的数据处理线程以线程 T_4 线程 T_5 、线程 T_6 为例,线程 T_4 请求物理锁 L_4 、线程 T_5 占用物理锁 L_4 、线程 T_5 请求物理锁 L_6 、线程 T_6 占用物理锁 L_6 、线程 T_6 请求物理锁 L_5 、线程 T_4 占用物理锁 L_5 ;

[0116] 线程 T_4 响应目标对象触发的从数据库中读写数据B的请求,其中数据B对应的物理锁为 L_4 ,则线程 T_4 生成请求物理锁 L_4 的加锁请求,当监测到线程 T_4 在发起物理锁请求的预设时长内未获取到物理锁 L_4 ,则获取当前的物理锁资源分配图,启动死锁检测,以线程 T_4 为起点,沿着<线程 T_4 >-[请求]-><物理锁 L_4 >的边,开始对当前获取的物理锁资源分配图进行检测,在检测过程中发现物理锁 L_4 被线程 T_5 占用,而此时线程 T_5 正在请求获取物理锁 L_6 ,此时物理锁 L_6 被线程 T_6 占用,线程 T_6 正在请求获取物理锁 L_5 ,而物理锁 L_5 被线程 T_4 占用,则此时形成一个死锁环路。

[0117] 3、死锁环路中的物理锁为处理数据库引擎层数据的物理锁以及处理数据库服务层数据的物理锁。

[0118] 如图11所示,为本申请实施例提供的第三种死锁环路示意图,其中,处理数据库引擎层数据的数据处理线程以线程 T_7 线程 T_9 、线程 T_{11} 为例,处理数据库服务层数据的数据处理线程以线程 T_8 、线程 T_{10} 为例,线程 T_9 请求物理锁 L_7 、线程 T_7 占用物理锁 L_7 、线程 T_7 请求物理锁 L_9 、线程 T_8 占用物理锁 L_9 、线程 T_{10} 请求物理锁 L_{11} 、线程 T_9 占用物理锁 L_{11} ;

[0119] 线程 T_9 响应目标对象触发的从数据库中读写数据C的请求,其中数据C对应的物理锁为 L_7 ,则线程 T_9 生成请求物理锁 L_7 的加锁请求,当监测到线程 T_{10} 在发起物理锁请求的预设时长内未获取到物理锁 L_7 ,则获取当前的物理锁资源分配图,启动死锁检测,以线程 T_9 为起点,沿着<线程 T_9 >-[请求]-><物理锁 L_7 >的边,开始对当前获取的物理锁资源分配图进行检测,在检测过程中发现物理锁 L_7 被线程 T_7 占用,而此时线程 T_7 正在请求获取物理锁 L_9 ,此时物理锁 L_9 被线程 T_8 占用,线程 T_8 正在请求获取物理锁 L_{10} ,而物理锁 L_{10} 被线程 T_{10} 占用,线程 T_{10} 正在请求物理锁 L_{11} ,而物理锁 L_{11} 被线程 T_9 占用,则此时形成一个死锁环路。

[0120] 如图12所示,为本申请实施例中的一种物理锁资源的分配检测装置1200的结构示意图,包括:

[0121] 请求单元1200,用于响应目标对象触发的从数据库中读写数据的请求,通过目标数据处理线程发起物理锁请求;其中,物理锁请求中至少携带待处理数据的标识信息;

[0122] 获取单元1201,用于在发起物理锁请求后的预设时长内,若未获取到与所述标识信息关联的物理锁,则获取当前的物理锁资源分配状态信息;

[0123] 检测单元1202,用于对物理锁资源分配状态信息进行检测,确定存在分配错误的物理锁及关联的数据处理线程。

[0124] 可选地,获取单元1201还用于:

[0125] 获取在数据库中执行数据读写操作的各个数据处理线程;

[0126] 根据各个数据处理线程请求的物理锁以及各个数据处理线程占用的物理锁,生成

物理锁资源分配状态信息。

[0127] 可选地,获取单元1201具体用于:

[0128] 将各个数据处理线程、各个数据处理线程请求的物理锁以及各个数据处理线程占用的物理锁作为节点;

[0129] 将多个节点中数据处理线程与请求的物理锁之间连线作为边,以及将多个节点中数据处理线程与占用的物理锁之间连线作为边;其中,数据处理线程与请求的物理锁之间的边,和数据处理线程与占用的物理锁之间的边的方向不同;

[0130] 根据确定出的节点和边生成物理锁资源分配图。

[0131] 可选地,获取单元1201具体用于:

[0132] 从第一对应关系集合中获取各个第一数据处理线程请求的物理锁;其中,第一对应关系集合中存储有发起物理锁请求的第一数据处理线程与请求的物理锁之间的第一对应关系,第一对应关系为所述第一数据处理线程发起物理锁请求后生成并存储在所述第一对应关系集合中的;

[0133] 从第二对应关系集合中获取各个第一数据处理线程占用的物理锁;其中,第二对应关系集合中存储有第一数据处理线程与占用的物理锁之间的第二对应关系,第二对应关系为第一对应关系集合中的第一数据处理线程请求的物理锁成功占用后生成并存储在第二对应关系集合中的。

[0134] 可选地,获取单元1201具体用于:

[0135] 遍历各个第二数据处理线程对应的容器数组;

[0136] 根据各个第二数据处理线程对应的容器数组中存储的二元组信息,确定各个第二数据处理线程请求的物理锁以及各个第二数据处理线程占用的物理锁;其中,容器数组中存储的二元组信息包括第二数据处理线程与请求的物理锁、第二数据处理线程与占用的物理锁中的至少一种。

[0137] 可选地,获取单元1201具体用于:

[0138] 对所述物理锁资源分配图进行检测;

[0139] 若根据物理锁资源分配图中各个节点以及节点之间连接的边的方向,从物理锁资源分配图中检测到按照边的方向将节点依次连接的节点环路;则将节点环路中包含的物理锁及数据处理线程确定为存在分配错误的物理锁及关联的数据处理线程。

[0140] 为了描述的方便,以上各部分按照功能划分为各模块(或单元)分别描述。当然,在实施本申请时可以把各模块(或单元)的功能在同一个或多个软件或硬件中实现。

[0141] 所属技术领域的技术人员能够理解,本申请的每个方面可以实现为系统、方法或程序产品。因此,本申请的每个方面可以具体实现为以下形式,即:完全的硬件实施方式、完全的软件实施方式(包括固件、微代码等),或硬件和软件方面结合的实施方式,这里可以统称为“电路”、“模块”或“系统”。

[0142] 在一些可能的实施方式中,本申请实施例还提供一种电子设备,参阅图13所示,电子设备1300可以至少包括至少一个处理器1301、以及至少一个存储器1302。其中,存储器1302存储有程序代码,当程序代码被处理器1301执行时,使得处理器1301执行本说明书上述描述的根据本申请各种示例性实施方式的物理锁资源的分配检测方法中的步骤,例如,处理器1301可以执行如图2中所示的步骤。

[0143] 在一些可能的实施方式中,本申请实施例还提供一种计算装置,可以至少包括至少一个处理单元、以及至少一个存储单元。其中,存储单元存储有程序代码,当程序代码被处理单元执行时,使得处理单元执行本说明书上述描述的根据本申请各种示例性实施方式的物理锁资源的分配检测方法中的步骤,例如,处理器1301可以执行如图2所示的步骤。

[0144] 下面参照图14来描述根据本申请的这种实施方式的计算装置1400。图14的计算装置1400仅仅是一个示例,不应对本申请实施例的功能和使用范围带来任何限制。

[0145] 如图14,计算装置1400以通用计算装置的形式表现。计算装置1400的组件可以包括但不限于:上述至少一个处理单元1401、上述至少一个存储单元1402、连接不同系统组件(包括存储单元1402和处理单元1401)的总线1403。

[0146] 总线1403表示几类总线结构中的一种或多种,包括存储器总线或者存储器控制器、外围总线、处理器或者使用多种总线结构中的任意总线结构的局域总线。

[0147] 存储单元1402可以包括易失性存储器形式的可读介质,例如随机存取存储器(RAM) 1421或高速缓存存储单元1422,还可以进一步包括只读存储器(ROM) 1423。

[0148] 存储单元1402还可以包括具有一组(至少一个)程序模块1324的程序/实用工具1425,这样的程序模块1424包括但不限于:操作系统、一个或者多个应用程序、其它程序模块以及程序数据,这些示例中的每一个或某种组合中可能包括网络环境的实现。

[0149] 计算装置1400也可以与一个或多个外部设备1404(例如键盘、指向设备等)通信,还可与一个或者多个使得用户能与计算装置1400交互的设备通信,或与使得该计算装置1400能与一个或多个其它计算装置进行通信的任何设备(例如路由器、调制解调器等等)通信。这种通信可以通过输入/输出(I/O)接口1405进行。并且,计算装置1400还可以通过网络适配器1406与一个或者多个网络(例如局域网(LAN),广域网(WAN)或公共网络,如因特网)通信。如图所示,网络适配器1406通过总线1403与用于计算装置1400的其它模块通信。应当理解,尽管图中未示出,可以结合计算装置1400使用其它硬件或软件模块,包括但不限于:微代码、设备驱动器、冗余处理器、外部磁盘驱动阵列、RAID系统、磁带驱动器以及数据备份存储系统等。

[0150] 在一些可能的实施方式中,本申请提供的物理锁资源的分配检测方法的每个方面还可以实现为一种程序产品的形式,其包括程序代码,当程序产品在计算机设备上运行时,程序代码用于使计算机设备执行本说明书上述描述的根据本申请各种示例性实施方式的物理锁资源的分配检测中的步骤,例如,计算机设备可以执行如图2所示的步骤。

[0151] 程序产品可以采用一个或多个可读介质的任意组合。可读介质可以是可读信号介质或者可读存储介质。可读存储介质例如可以是但不限于电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。可读存储介质的更具体的例子(非穷举的列表)包括:具有一个或多个导线的电连接、便携式盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。

[0152] 尽管已描述了本申请的优选实施例,但本领域内的技术人员一旦得知了基本创造性概念,则可对这些实施例做出另外的变更和修改。所以,所附权利要求意欲解释为包括优选实施例以及落入本申请范围的所有变更和修改。

[0153] 显然,本领域的技术人员可以对本申请进行各种改动和变型而不脱离本申请的精

神和范围。这样,倘若本申请的这些修改和变型属于本申请权利要求及其等同技术的范围之内,则本申请也意图包含这些改动和变型在内。

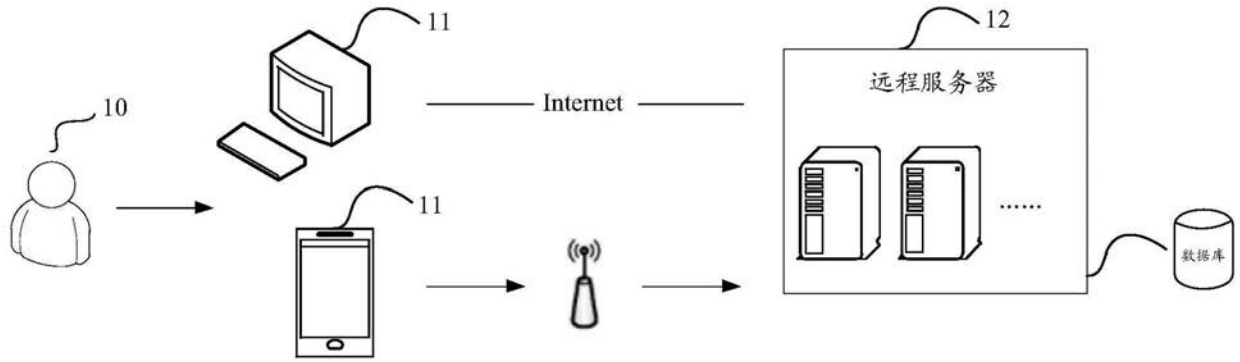


图1

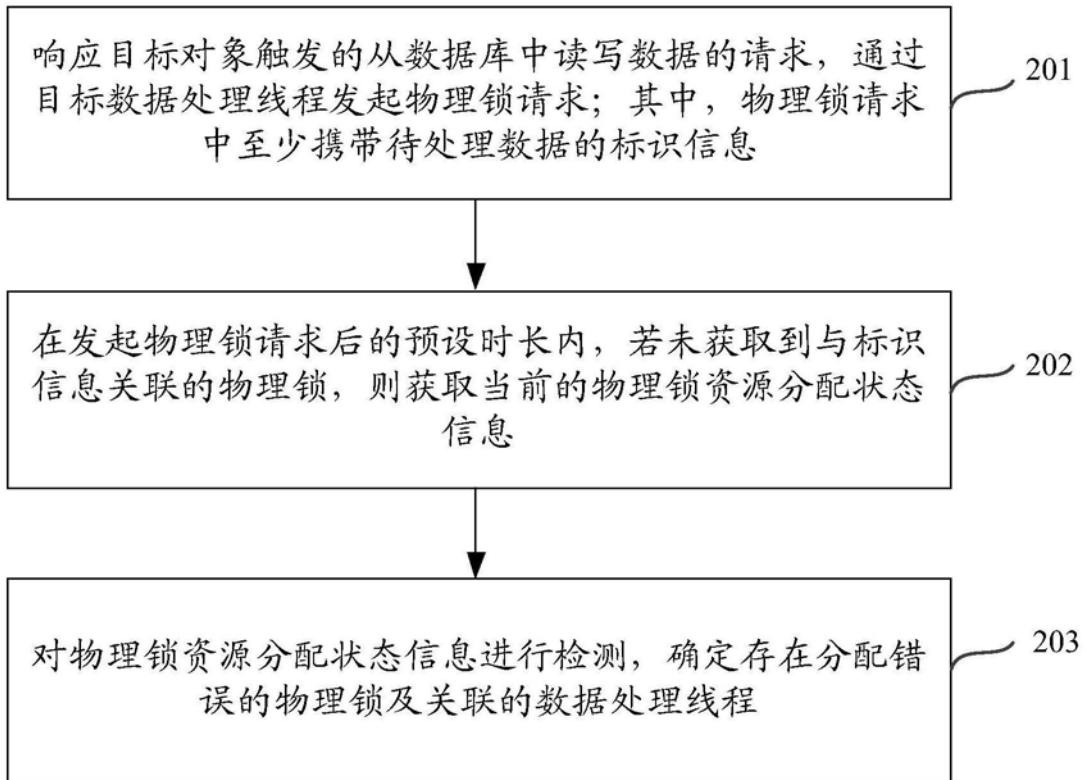


图2

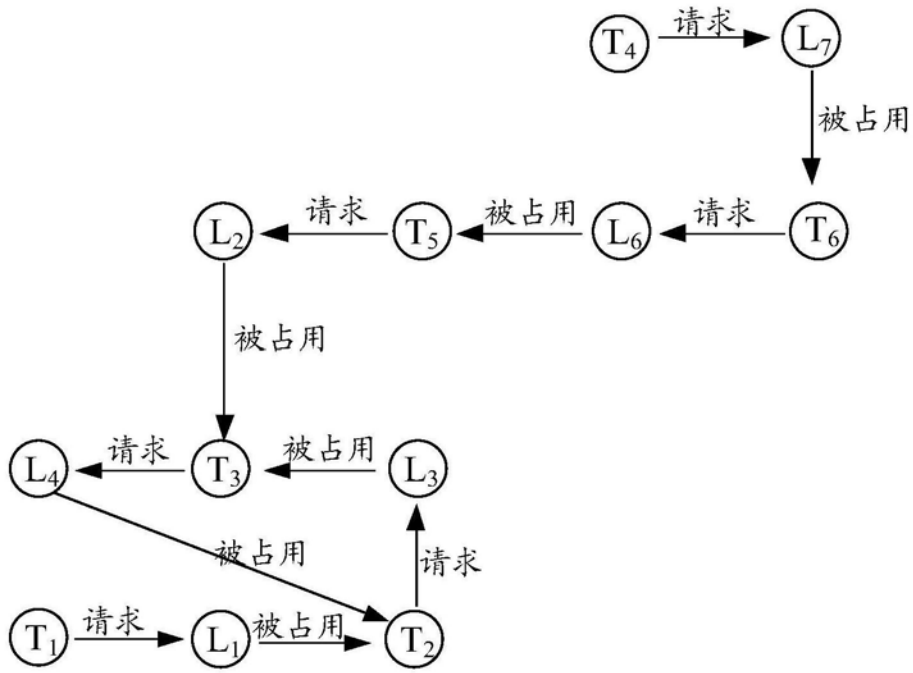


图3

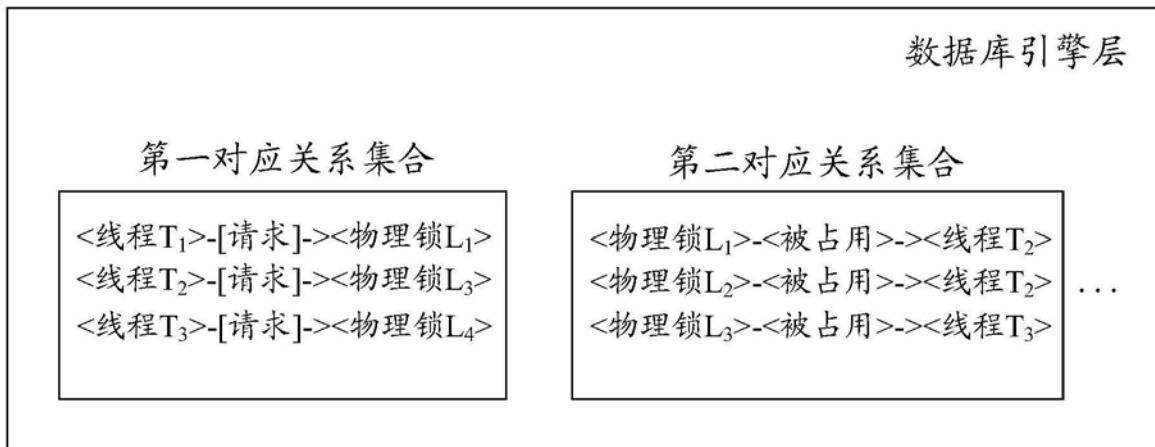


图4

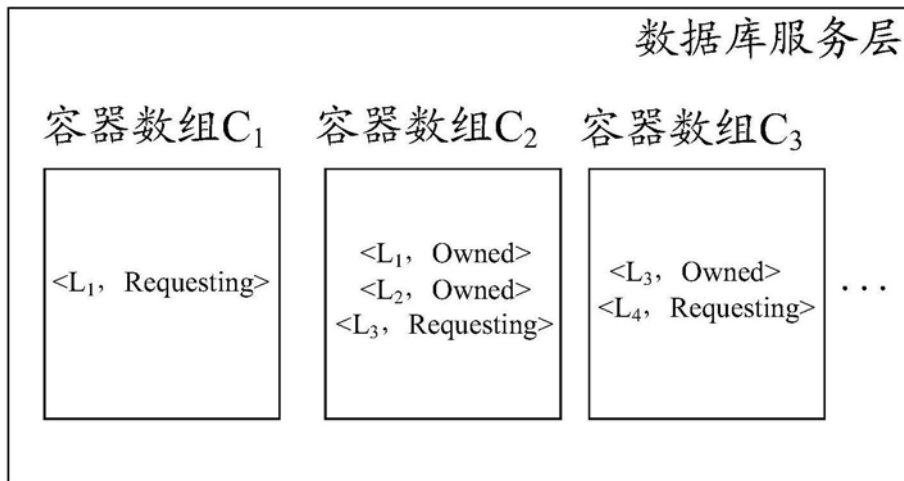


图5

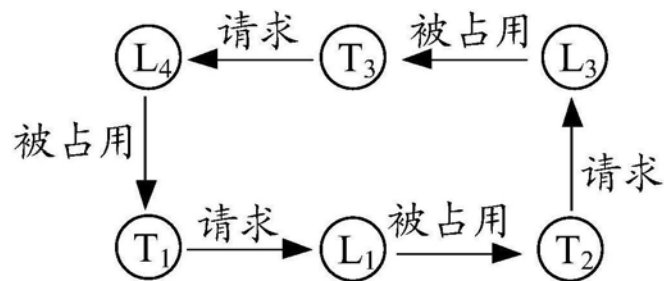


图6

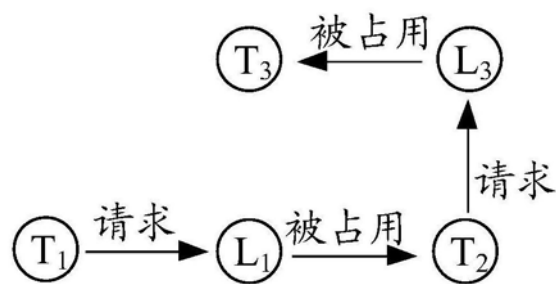


图7

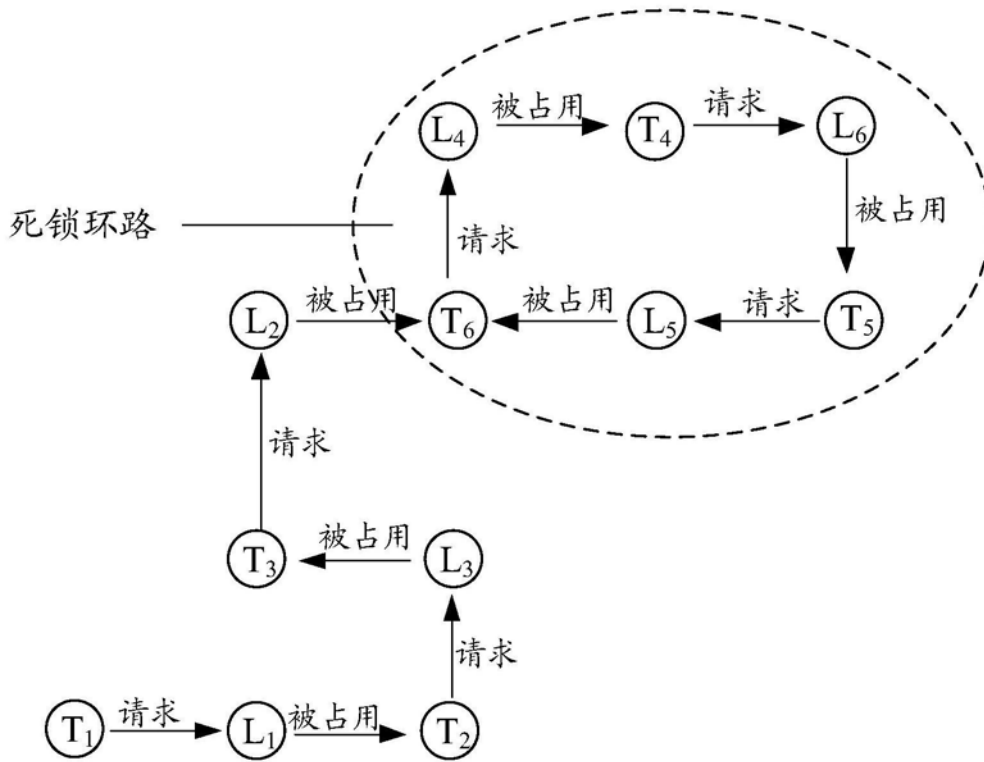


图8

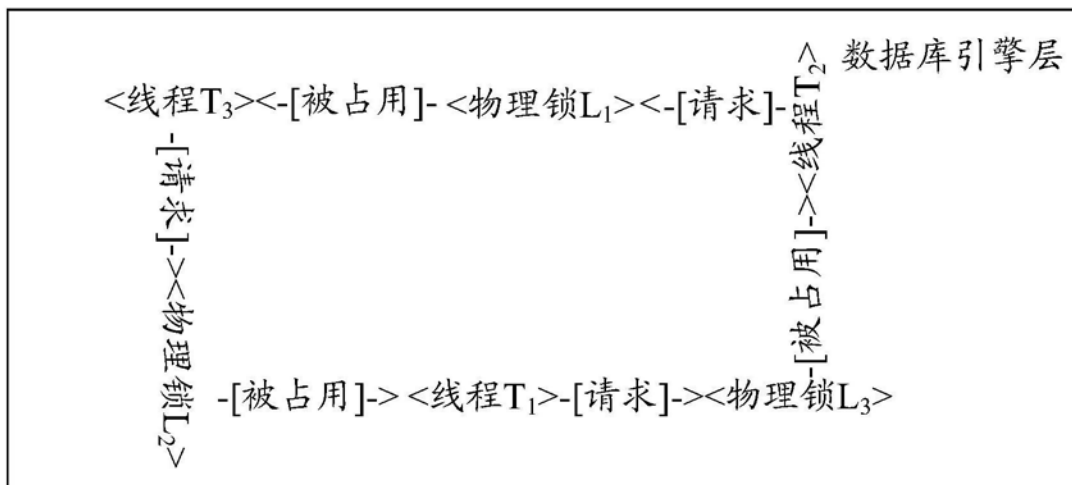


图9

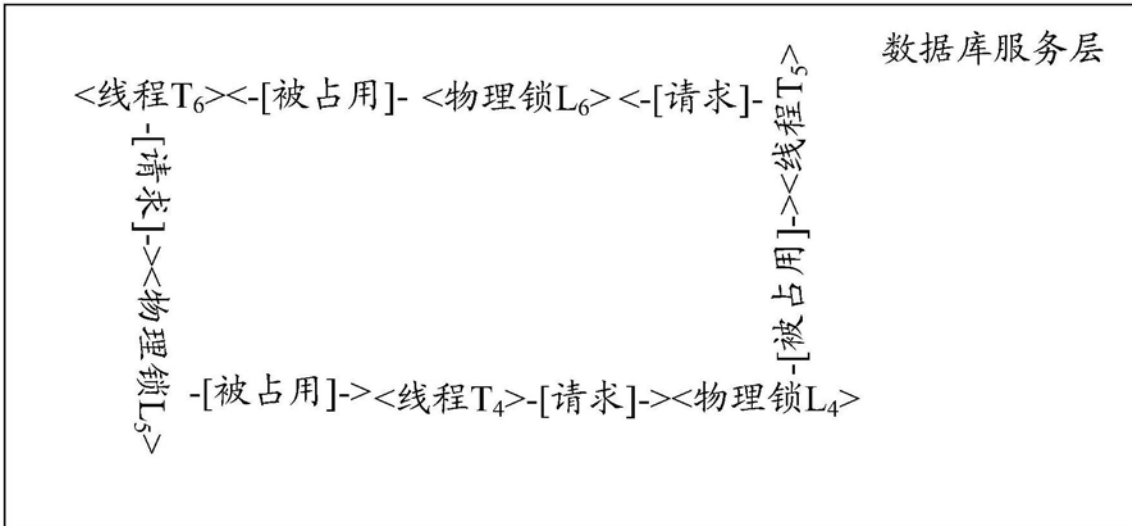


图10

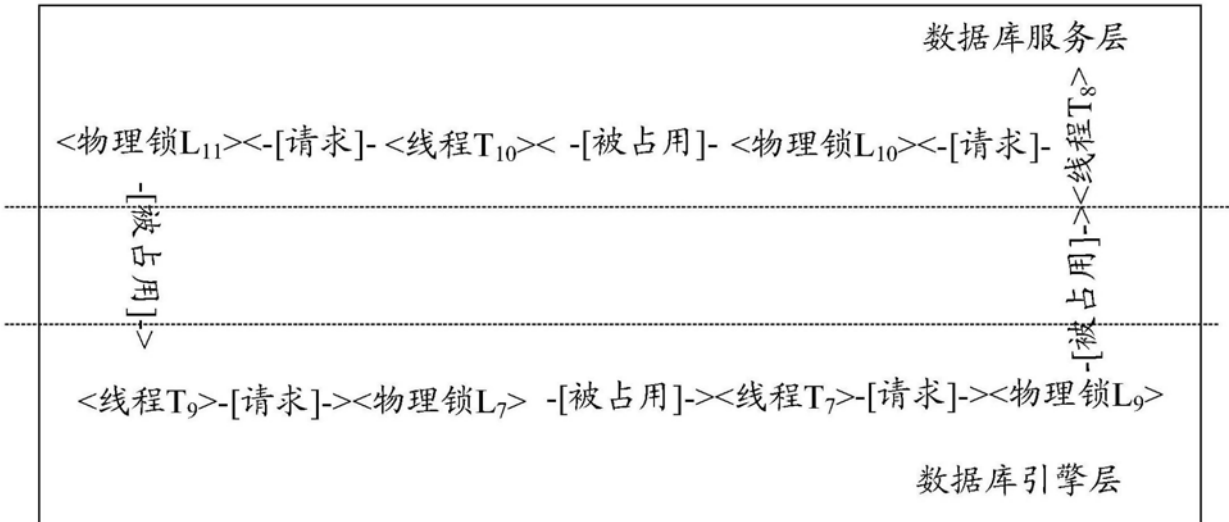


图11

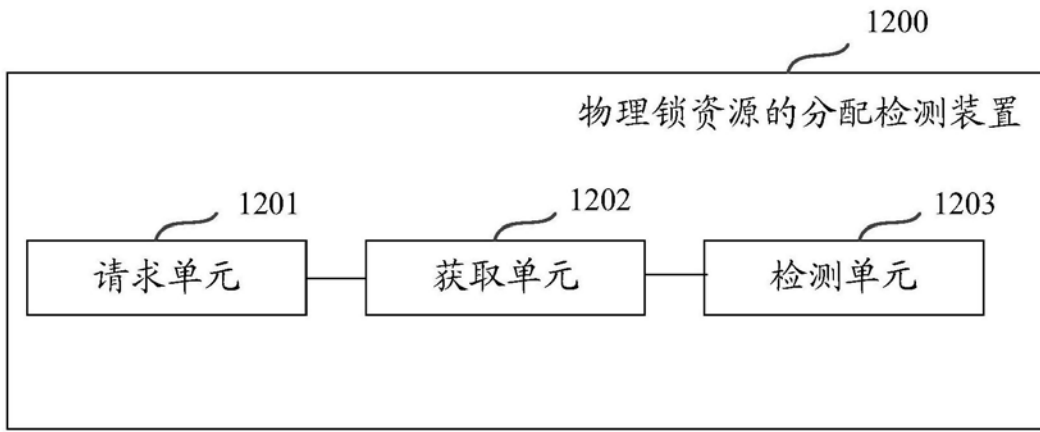


图12

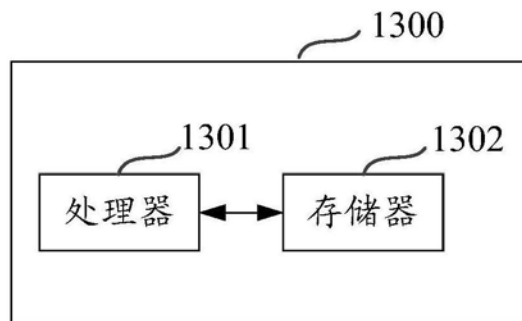


图13

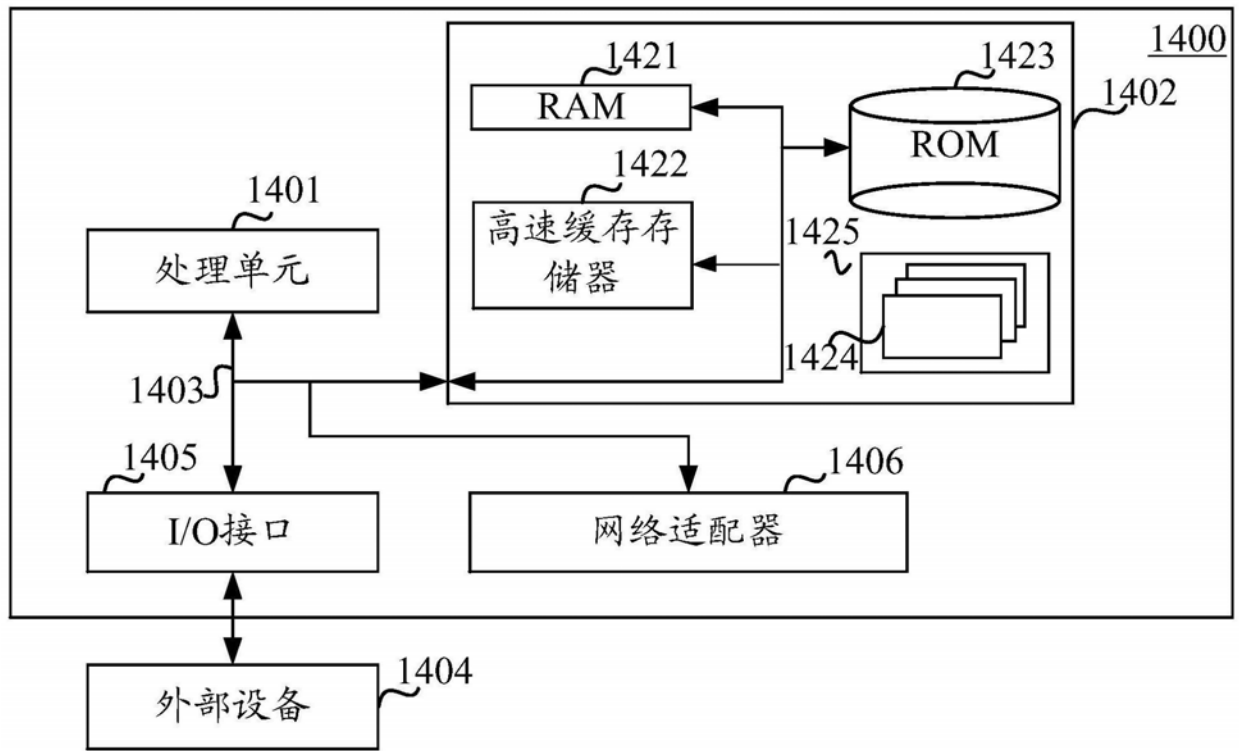


图14