(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 110888858 B
(45)授权公告日 2023.06.30

(21)申请号 201911040315.8

(22)申请日 2019.10.29

(65)同一申请的已公布的文献号
申请公布号 CN 110888858 A

(43)申请公布日 2020.03.17

(73)专利权人 北京奇艺世纪科技有限公司
地址 100080 北京市海淀区海淀北一街2号鸿城拓展大厦10、11层

(72)发明人 高子叶 刘羽

(74)专利代理机构 北京华夏泰和知识产权代理有限公司 11662
专利代理师 韩来兵

(51)Int.Cl.
G06F 16/21(2019.01)
G06F 9/50(2006.01)

(56)对比文件
CN 110032424 A,2019.07.19
CN 108874552 A,2018.11.23

审查员 靳苗苗

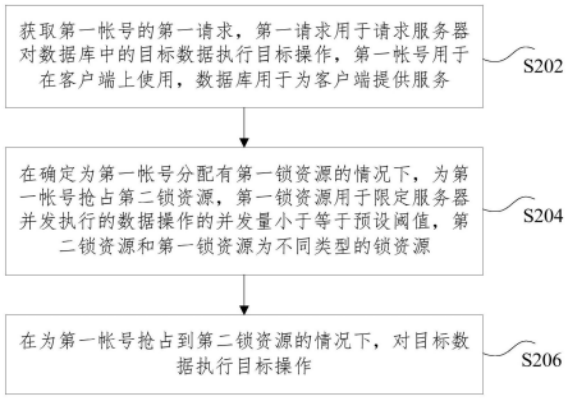
权利要求书2页 说明书10页 附图3页

(54)发明名称

数据库的操作方法和装置、存储介质、电子装置

(57)摘要

本申请公开了一种数据库的操作方法和装置、存储介质、电子装置。其中，该方法包括：获取第一帐号的第一请求，其中，第一请求用于请求服务器对数据库中的目标数据执行目标操作，第一帐号用于在客户端上使用，数据库用于为客户端提供服务；在确定为第一帐号分配有第一锁资源的情况下，为第一帐号抢占第二锁资源，其中，第一锁资源用于限定服务器并发执行的数据操作的并发量小于等于预设阈值，第二锁资源和第一锁资源为不同类型的锁资源；在为第一帐号抢占到第二锁资源的情况下，对目标数据执行目标操作。本申请解决了相关技术中数据库的数据不一致的技术问题。



CN 110888858 B

1.一种数据库的操作方法,其特征在于,包括:

获取第一帐号的第一请求,其中,所述第一请求用于请求服务器对数据库中的目标数据执行目标操作,所述第一帐号用于在客户端上使用,所述数据库用于为所述客户端提供服务;

在确定为所述第一帐号分配有第一锁资源的情况下,为所述第一帐号抢占第二锁资源,其中,所述第一锁资源用于限定所述服务器并发执行的数据操作的并发量小于等于预设阈值,所述第二锁资源和所述第一锁资源为不同类型的锁资源,所述第一锁资源为所述服务器上用于控制并发量预先生成的多个ReetrantLock实例,所述ReetrantLock实例的数量与要控制的并发量的数量一致;生成多个ReetrantLock实例包括:在服务启动后生成从0开始顺序自增的锁id,每个锁id对应生成一个ReetrantLock实例;为所述第一帐号分配所述第一锁资源包括:对所述第一帐号的哈希码取模,得到一个锁id,将该锁id对应的ReetrantLock实例分配给所述第一帐号;

在为所述第一帐号抢占到所述第二锁资源的情况下,对所述目标数据执行所述目标操作;

在对目标数据执行目标操作之后,解除第一锁资源、第二锁资源与第一帐号之间的关联,解除与第一帐号之间的关联的第一锁资源和第二锁资源允许被任意的其他帐号抢占。

2.根据权利要求1所述的方法,其特征在于,所述方法还包括按照如下方式为所述第一帐号分配所述第一锁资源:

确定多个实例中的目标实例,其中,所述目标实例是为所述第一帐号分配的实例,所述多个实例是在所述服务器上生成的用于分配所述第一锁资源的实例;

将所述目标实例所属的所述第一锁资源分配给所述第一帐号。

3.根据权利要求2所述的方法,其特征在于,确定多个实例中的目标实例包括:

在所述多个实例中存在已分配给所述第一帐号的实例的情况下,将所述已分配给所述第一帐号的实例作为所述目标实例;

在所述多个实例中不存在已分配给所述第一帐号的实例的情况下,从所述多个实例中分配所述目标实例给所述第一帐号,其中,所述目标实例对应的锁标识与所述第一帐号的帐号标识匹配。

4.根据权利要求2所述的方法,其特征在于,将所述目标实例所属的所述第一锁资源分配给所述第一帐号包括:

向所述目标实例发送第一抢占请求,其中,所述第一抢占请求用于请求所述目标实例为所述第一帐号分配所述第一锁资源;

在接收到的所述目标实例的第一响应消息表明抢占成功的情况下,将所述第一响应消息所指示的所述第一锁资源分配给所述第一帐号。

5.根据权利要求2所述的方法,其特征在于,在确定多个实例中的目标实例之前,所述方法包括:

按照所述预设阈值生成所述多个实例,其中,所述预设阈值用于限定所述服务器并发执行的数据操作的并发量;

保存所述多个实例和所述多个实例中每个实例对应的锁标识。

6.根据权利要求1至5中任意一项所述的方法,其特征在于,为所述第一帐号抢占第二

锁资源包括：

向所述数据库发送第二抢占请求，其中，所述第二抢占请求用于请求所述数据库为所述第一帐号分配与所述目标数据对应的所述第二锁资源；

在接收到的所述数据库的第二响应消息表明抢占成功的情况下，将所述第二响应消息所指示的所述第二锁资源分配给所述第一帐号。

7.根据权利要求1至5中任意一项所述的方法，其特征在于，在对所述目标数据执行所述目标操作之后，所述方法还包括：

解除所述第一锁资源、所述第二锁资源与所述第一帐号之间的关联，其中，解除与所述第一帐号之间的关联的所述第一锁资源和所述第二锁资源允许被第二帐号抢占。

8.一种数据库的操作装置，其特征在于，包括：

获取单元，用于获取第一帐号的第一请求，其中，所述第一请求用于请求服务器对数据库中的目标数据执行目标操作，所述第一帐号用于在客户端上使用，所述数据库用于为所述客户端提供服务；

抢占单元，用于在确定为所述第一帐号分配有第一锁资源的情况下，为所述第一帐号抢占第二锁资源，其中，所述第一锁资源用于限定所述服务器并发执行的数据操作的并发量小于等于预设阈值，所述第二锁资源和所述第一锁资源为不同类型的锁资源，所述第一锁资源为所述服务器上用于控制并发量预先生成的多个ReetrantLock实例，所述ReetrantLock实例的数量与要控制的并发量的数量一致；生成多个ReetrantLock实例包括：在服务启动后生成从0开始顺序自增的锁id，每个锁id对应生成一个ReetrantLock实例；为所述第一帐号分配所述第一锁资源包括：对所述第一帐号的哈希码取模，得到一个锁id，将该锁id对应的ReetrantLock实例分配给所述第一帐号；

操作单元，用于在为所述第一帐号抢占到所述第二锁资源的情况下，对所述目标数据执行所述目标操作；

解除单元，用于在对目标数据执行目标操作之后，解除第一锁资源、第二锁资源与第一帐号之间的关联，解除与第一帐号之间的关联的第一锁资源和第二锁资源允许被任意的其他帐号抢占。

9.一种存储介质，其特征在于，所述存储介质包括存储的程序，其中，所述程序运行时执行上述权利要求1至7任一项中所述的方法。

10.一种电子装置，包括存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序，其特征在于，所述处理器通过所述计算机程序执行上述权利要求1至7任一项中所述的方法。

数据库的操作方法和装置、存储介质、电子装置

技术领域

[0001] 本申请涉及互联网领域，具体而言，涉及一种数据库的操作方法和装置、存储介质、电子装置。

背景技术

[0002] 随着信息化建设的高速发展，信息系统已经成为企业维持业务运转的关键，企业迫切需要提高信息系统的可用性，保证业务的连续性，最大限度地减少因灾难或故障所带来的损失。另一方面，多样化的业务类型导致数据访问需求的日趋复杂化，数据量的急剧攀升也导致数据库服务器不堪重负，企业同样迫切需要提高信息系统的运行效率。

[0003] 在大型活动(如公司年庆、购物节)时，由于数据读写量极大，例如：用户互动时可以发起换牌邀请，邀请好友帮助替换卡牌，以达到卡牌总和为固定数(如为9)的目的，该用户好友可根据自己持有卡牌情况，替换卡牌，帮助完成活动任务，这样就可能出现多个用户同时更新同一张卡牌的情况，造成数据更新不一致的问题，而且用来替换的卡牌在去替换别人卡牌的时候，自己也有被替换的可能，此时需要保证多个数据更新的一致性。

[0004] 针对上述的问题，目前尚未提出有效的解决方案。

发明内容

[0005] 本申请实施例提供了一种数据库的操作方法和装置、存储介质、电子装置，以至少解决相关技术中数据库的数据不一致的技术问题。

[0006] 根据本申请实施例的一个方面，提供了一种数据库的操作方法，包括：获取第一帐号的第一请求，其中，第一请求用于请求服务器对数据库中的目标数据执行目标操作，第一帐号用于在客户端上使用，数据库用于为客户端提供服务；在确定为第一帐号分配有第一锁资源的情况下，为第一帐号抢占第二锁资源，其中，第一锁资源用于限定服务器并发执行的数据操作的并发量小于等于预设阈值，第二锁资源和第一锁资源为不同类型的锁资源；在为第一帐号抢占到第二锁资源的情况下，对目标数据执行目标操作。

[0007] 根据本申请实施例的另一方面，还提供了一种数据库的操作装置，包括：获取单元，用于获取第一帐号的第一请求，其中，第一请求用于请求服务器对数据库中的目标数据执行目标操作，第一帐号用于在客户端上使用，数据库用于为客户端提供服务；抢占单元，用于在确定为第一帐号分配有第一锁资源的情况下，为第一帐号抢占第二锁资源，其中，第一锁资源用于限定服务器并发执行的数据操作的并发量小于等于预设阈值，第二锁资源和第一锁资源为不同类型的锁资源；操作单元，用于在为第一帐号抢占到第二锁资源的情况下，对目标数据执行目标操作。

[0008] 根据本申请实施例的另一方面，还提供了一种存储介质，该存储介质包括存储的程序，程序运行时执行上述的方法。

[0009] 根据本申请实施例的另一方面，还提供了一种电子装置，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，处理器通过计算机程序执行上述的方

法。

[0010] 在本申请实施例中，采用“获取第一帐号的第一请求，第一请求用于请求服务器对数据库中的目标数据执行目标操作，第一帐号用于在客户端上使用，数据库用于为客户端提供服务；在确定为第一帐号分配有第一锁资源的情况下，为第一帐号抢占第二锁资源，第一锁资源用于限定服务器并发执行的数据操作的并发量小于等于预设阈值，第二锁资源和第一锁资源为不同类型的锁资源；在为第一帐号抢占到第二锁资源的情况下，对目标数据执行目标操作”的方式，通过第一锁资源的数量限定了每个服务器上的数据操作的并发量，通过第二锁资源对同一目标数据的操作数，通过抢占锁资源的形式来获得目标数据的操作权限，避免了大量请求同时请求操作同一目标数据时出现数据更新不一致，可以解决相关技术中数据库的数据不一致的技术问题。

附图说明

[0011] 此处所说明的附图用来提供对本申请的进一步理解，构成本申请的一部分，本申请的示意性实施例及其说明用于解释本申请，并不构成对本申请的不当限定。在附图中：

[0012] 图1是根据本申请实施例的数据库的操作方法的硬件环境的示意图；

[0013] 图2是根据本申请实施例的一种可选的数据库的操作方法的流程图；

[0014] 图3是根据本申请实施例的可选的服务器和数据库系统的示意图；

[0015] 图4是根据本申请实施例的一种可选的数据库的操作方法的流程图；

[0016] 图5是根据本申请实施例的一种可选的数据库的操作装置的示意图；

[0017] 以及

[0018] 图6是根据本申请实施例的一种终端的结构框图。

具体实施方式

[0019] 为了使本技术领域的人员更好地理解本申请方案，下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本申请一部分的实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都应当属于本申请保护的范围。

[0020] 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。此外，术语“包括”和“具有”以及他们的任何变形，意图在于覆盖不排他的包含，例如，包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元，而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0021] 首先，在对本申请实施例进行描述的过程中出现的部分名词或者术语适用于如下解释：

[0022] Redis(全称:Remote Dictionary Server远程字典服务)是一个开源的使用ANSI C语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value数据库，并提供多种语

言的API。

[0023] Cookie,有时也用其复数形式Cookies,指某些网站为了辨别用户身份、进行跟踪而储存在用户本地终端上的数据(通常经过加密)。

[0024] 根据本申请实施例的一方面,提供了一种数据库的操作方法的方法实施例。

[0025] 可选地,在本实施例中,上述数据库的操作方法可以应用于如图1所示的由终端101和服务器103所构成的硬件环境中。如图1所示,服务器103通过网络与终端101进行连接,可用于为终端或终端上安装的客户端提供服务(如游戏服务、多媒体服务等),可在服务器上或独立于服务器设置数据库105,用于为服务器103提供数据存储服务,上述网络包括但不限于:广域网、城域网或局域网,终端101并不限定于PC、手机、平板电脑等。本申请实施例的数据库的操作方法可以由服务器103来执行,还可以是由服务器103和终端101共同执行。

[0026] 图2是根据本申请实施例的一种可选的数据库的操作方法的流程图,如图2所示,该方法可以包括以下步骤:

[0027] 步骤S202,获取第一帐号的第一请求,第一请求用于请求服务器对数据库中的目标数据执行目标操作,第一帐号用于在客户端上使用,数据库用于为客户端提供服务。

[0028] 上述的客户端为安装在上述终端上的客户端,可以为多媒体内容的客户端、游戏客户端、直播客户端、通讯客户端等,后续以视频客户端为例进行说明;上述的第一帐号即在客户端中使用的帐号,可以为手机号、即时通讯帐号、临时帐号等;上述数据库用于为客户端提供服务所需的数据,包括上述的目标数据;目标操作包括对目标数据的更新、删除、替换等操作。

[0029] 步骤S204,在确定为第一帐号分配有第一锁资源的情况下,为第一帐号抢占第二锁资源,第一锁资源用于限定服务器并发执行的数据操作的并发量小于等于预设阈值,第二锁资源和第一锁资源为不同类型的锁资源。

[0030] 上述的服务器可以为多个,对于每个服务器,在该服务器上的第一锁资源的数量限定了该服务器并发执行的数据操作的并发量,例如,有N个第一锁资源,并发量就为N(N的取值为大于等于1的整数,如为5);对于第一帐号而言,只有获得了第一锁资源,才能进行下一步操作,即抢占第二锁资源,在抢占到第二锁资源之后进行目标操作;数据操作为对数据的操作(包括上述目标操作),如修改、删除、增加、补充等。

[0031] 上述锁资源相当于模拟“实体锁”的虚拟锁,锁住的对象为“数据”,此处虚拟锁可以通过软件来管理,如软件实例,具体实例如JAVA中通过new的形式运行起来的具备“生成锁、分配锁、回收锁”等逻辑功能的程序代码。

[0032] 步骤S206,在为第一帐号抢占到第二锁资源的情况下,对目标数据执行目标操作。

[0033] 通过上述步骤,通过第一锁资源的数量限定了每个服务器上的数据操作的并发量,通过第二锁资源对同一目标数据的操作数,通过抢占锁资源的形式来获得目标数据的操作权限,避免了大量请求同时请求操作同一目标数据时出现数据更新不一致,可以解决相关技术中数据库的数据不一致的技术问题。下面结合图2所示的步骤进一步详述本申请的技术方案。

[0034] 在步骤S202提供的技术方案中,获取第一帐号的第一请求,第一请求用于请求服务器对数据库中的目标数据执行目标操作。

[0035] 该数据库为分布式数据库,包括但不局限于Key-value NoSQL类型的数据库(例如Redis Riak等)、column family NoSQL类型的数据库(也称wide column store,例如Hbase Cassandra等)、document NoSQL(例如mongodb等)。后续以数据库为Redis为例进行说明。

[0036] 在步骤S204提供的技术方案中,在确定为第一帐号分配有第一锁资源的情况下,为第一帐号抢占第二锁资源。

[0037] 对于单台服务器,在确定多个实例中的目标实例之前,可以生成顺序自增的锁标识ID(如1至N),按照预设阈值(即N)生成多个实例,通过预先设定的预设阈值来限定服务器并发执行的数据操作的并发量;在哈希容器(如ConcurrentHashMap)中保存多个实例和多个实例中每个实例对应的锁标识。

[0038] ConcurrentHashMap容器里有多把锁(即锁资源),每一把锁用于锁容器其中一部分数据(即锁资源对应的数据),那么当多线程访问容器里不同数据段的数据时,线程间就不会存在锁竞争,从而可以有效的提高并发访问效率,即锁分段技术,首先将数据分成一段一段的存储,然后给每一段数据配一把锁,当一个线程占用锁访问其中一个段数据的时候,其他段的数据也能被其他线程访问。

[0039] 在上述实施例中,第一帐号可以按照如下步骤S2042-步骤S2044获取第一锁资源:

[0040] 步骤S2042,确定多个实例中的目标实例,目标实例是为第一帐号分配的实例,多个实例是在服务器上生成的用于分配第一锁资源的实例。

[0041] 可选地,在确定多个实例中的目标实例时,在多个实例中存在已分配给第一帐号的实例(换言之,第一帐号之前发起过请求,本次请求是重复发起的请求,且已经为第一帐号之前的请求分配过实例)的情况下,将已分配给第一帐号的实例作为目标实例;在多个实例中不存在已分配给第一帐号的实例的情况下,从多个实例中分配目标实例给第一帐号,目标实例对应的锁标识与第一帐号的帐号标识匹配,例如,对第一帐号的帐号标识hashcode进行取模,得到一个锁标识ID,进而以该锁ID锁标识的实例为目标实例。

[0042] 步骤S2044,将目标实例所属的第一锁资源分配给第一帐号。

[0043] 可选地,在将目标实例所属的第一锁资源分配给第一帐号时,向目标实例发送第一抢占请求,第一抢占请求用于请求目标实例为第一帐号分配第一锁资源;在接收到的目标实例的第一响应消息表明抢占成功的情况下,将第一响应消息所指示的第一锁资源分配给第一帐号,在第一响应消息表明抢占失败的情况下,向第一帐号返回操作失败的提示。

[0044] 与前述抢占第一锁资源的方式类似,为第一帐号抢占第二锁资源包括:向数据库发送第二抢占请求,第二抢占请求用于请求数据库为第一帐号分配与目标数据对应的第二锁资源;在接收到的数据库的第二响应消息表明抢占成功的情况下,将第二响应消息所指示的第二锁资源分配给第一帐号,在第二响应消息表明抢占失败的情况下,向第一帐号返回操作失败的提示。

[0045] 可选地,第二锁资源是与数据库的类型适配的,例如数据库是Redis时,第二锁资源可以为setnx实现的锁资源;数据库是Hbase时,第二锁资源可以为jdk中的ReadWriteLock类实现的锁资源,其余类型的数据库与此类似,后续继续以数据库是Redis、第二锁资源为setnx实现的锁资源为例进行说明。第二锁资源与第一锁资源的类型不同,第二锁资源是针对分布式数据库使用的,如setnx,而第一锁资源是针对服务器使用的,如ReentrantLock。

[0046] 在步骤S206提供的技术方案中，在为第一帐号抢占到第二锁资源的情况下，对目标数据执行目标操作，如执行对卡牌的更新、交换、删除等操作。

[0047] 可选地，在对目标数据执行目标操作之后，解除第一锁资源、第二锁资源与第一帐号之间的关联，解除与第一帐号之间的关联的第一锁资源和第二锁资源允许被任意的其他帐号抢占，当然，第一帐号也可以重新发起新的抢占。

[0048] 本申请的技术方案可以用于高并发场景中的数据更新，通过服务器端的第一类锁(ReentrantLock)结合Redis分布式锁(如setnx)，解决了高并发场景中一次操作中更新多份数据，如何保证数据更新一致性的问题。

[0049] 作为一种可选的实施例，下面以将本申请的技术方案应用于保证卡牌数据的一致性为例进行说明：

[0050] 若仅仅采用分布式锁，比如Redis的setnx方式，可以保证在并发不大的情况下，保证多请求抢占锁的行为顺利进行，但是在服务器的压力测试过程中，偶尔还是会出现同一个卡牌被使用两次或者多次的情况，说明高并发场景下，抢占一个Redis分布式锁的数据在Redis集群内部多机器同步过程中，可能会出现多次抢占导致不一致的情况，所以需要一种方法解决这种问题。

[0051] 为了保证在高并发场景下多份数据更新的一致性，在如图3所示的系统架构中，包括服务器集群(以包括2台服务器为例)、Redis数据库集群(以一主一从为例)，基于服务中已使用的Redis分布式锁的情况下，在单台服务器中加入ReentrantLock，来限制单台服务器内部的并发情况，在试图抢占Redis分布式锁之前，需要先抢占相应的ReentrantLock(即容器中的锁，或称锁资源)，从而对单台服务器上的并发情况做控制，保证数据更新存储的准确性以及服务的稳定性。具体实现步骤如图4所示：

[0052] 单台服务器上，服务启动后预先生成一定数量ReetrantLock实例(即多个实例)，例如，生成从0开始顺序自增的锁id，每个锁id对应生成一个ReetrantLock实例，锁id和ReetrantLock实例保存到ConcurrentHashMap中，以利用ConcurrentHashMap适用于高并发场景的特性。

[0053] 步骤S402，用户访问接口，例如，用户更新卡牌时访问服务器提供的数据更新接口。

[0054] 步骤S404，在容器ConcurrentHashMap查找预先生成的锁。

[0055] 如用户执行卡牌数字更新操作的时候，需要先得到相应的ReetrantLock实例，通过用户cookie获取到登录用户id(即用户的帐号)，根据登录用户id的哈希码hashCode，按照ReetrantLock的数量取模，得到一个锁id，通过ConcurrentHashMap得到锁id对应的ReetrantLock实例，例如，用户id为123，ReetrantLock的数量为10，取模后的结果为3，那么就尝试将锁id为3的锁分配给该用户。

[0056] 步骤S406，得到ReetrantLock实例之后，尝试抢占第一锁资源，解决单台服务器的并发量过大的问题。

[0057] 通过指令ReetrantLock.tryLock()抢占锁资源，只有抢占锁资源成功的情况下，才能进行后续操作。

[0058] 需要说明的是，若仅仅使用分布式锁，在并发较高的情况，会出现单台服务器瞬时存在多个请求的情况下，同时获得了Redis的分布式锁，导致出现脏数据，故在单台服务器

中加入ReentrantLock,控制了单台服务上的并发问题,为保证数据准确性,防止同时更新同一份数据的情况,需要通过抢占锁来控制并发;多台服务器之间继续使用Redis分布式锁,解决了服务的并发数据更新问题。

[0059] 步骤S408,抢占锁失败的情况下,为了服务性能,直接返回相应操作失败code码。

[0060] 步骤S410,抢占锁成功的情况下,获取第一锁资源ReetrantLock。

[0061] 步骤S412,获取第一锁资源ReetrantLock失败的情况下,直接返回相应操作失败code码,并释放ReetrantLock锁资源。

[0062] 步骤S414,在ReetrantLock锁抢占成功后,则继续进行Redis分布式锁的抢占操作,即尝试第二次抢占,以解决服务中多台服务器之间的并发量过大的问题。

[0063] 通过Redis的setnx方式,尝试抢占要修改的卡牌数据id对应的Redis分布式锁,抢占Redis分布式锁成功的情况下,才能进行数据更新存储操作。

[0064] 步骤S416,抢占Redis分布式锁失败的情况下,直接返回相应操作失败code码,并释放Redis的分布式锁资源和ReetrantLock锁。

[0065] 步骤S418,抢占Redis分布式锁成功的情况下,获取要修改的卡牌数据id对应的Redis分布式锁。

[0066] 步骤S420,在获取要修改的卡牌数据id对应的Redis分布式锁时,若出现异常导致获取失败,则返回相应操作失败code码,并释放Redis的分布式锁资源和ReetrantLock锁。

[0067] 步骤S422,在获取到要修改的卡牌数据id对应的Redis分布式锁之后,进行数据更新。

[0068] 如果涉及到两个用户的卡牌互换数字的操作,需要对更新的两个用户都进行上述操作,保证两张卡牌数字更新的一致性。

[0069] 步骤S424,操作后,无论成功失败,需要依次对已经抢占成功的锁进行锁释放。

[0070] 在Redis中,清除卡牌id对应的分布式锁数据,在ReetrantLock实例中执行unLock()方法以释放ReetrantLock。

[0071] 上述实施方式以卡牌活动为例进行说明,活动中用户参与之后可以获得卡牌,通过互相换牌来达成条件进行抽奖。本申请的方案可用在控制换牌流程中的并发控制,需要说明的是,其他需求并发更新同一份数据的场景同样适用于本申请的技术方案。

[0072] 上述方案使用ReentranLock结合Redis分布式锁实现了高并发场景下(如视频、阅读、社交、教育、金融等各端产品及其应用)的数据一致性更新。

[0073] 需要说明的是,对于前述的各方法实施例,为了简单描述,故将其都表述为一系列的动作组合,但是本领域技术人员应该知悉,本申请并不受所描述的动作顺序的限制,因为依据本申请,某些步骤可以采用其他顺序或者同时进行。其次,本领域技术人员也应该知悉,说明书中所描述的实施例均属于优选实施例,所涉及的动作和模块并不一定是本申请所必须的。

[0074] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到根据上述实施例的方法可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。基于这样的理解,本申请的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质(如ROM/RAM、磁碟、光盘)中,包括若干指令用以使得一台终端设备(可以是手机,计算

机,服务器,或者网络设备等)执行本申请各个实施例所述的方法。

[0075] 根据本申请实施例的另一个方面,还提供了一种用于实施上述数据库的操作方法的数据库的操作装置。图5是根据本申请实施例的一种可选的数据库的操作装置的示意图,如图5所示,该装置可以包括:

[0076] 获取单元501,用于获取第一帐号的第一请求,其中,第一请求用于请求服务器对数据库中的目标数据执行目标操作,第一帐号用于在客户端上使用,数据库用于为客户端提供服务;

[0077] 抢占单元503,用于在确定为第一帐号分配有第一锁资源的情况下,为第一帐号抢占第二锁资源,其中,第一锁资源用于限定服务器并发执行的数据操作的并发量小于等于预设阈值,第二锁资源和第一锁资源为不同类型的锁资源;

[0078] 操作单元505,用于在为第一帐号抢占到第二锁资源的情况下,对目标数据执行目标操作。

[0079] 需要说明的是,该实施例中的获取单元501可以用于执行本申请实施例中的步骤S202,该实施例中的抢占单元503可以用于执行本申请实施例中的步骤S204,该实施例中的操作单元505可以用于执行本申请实施例中的步骤S206。

[0080] 此处需要说明的是,上述模块与对应的步骤所实现的示例和应用场景相同,但不限于上述实施例所公开的内容。需要说明的是,上述模块作为装置的一部分可以运行在如图1所示的硬件环境中,可以通过软件实现,也可以通过硬件实现。

[0081] 通过上述模块,通过第一锁资源的数量限定了每个服务器上的数据操作的并发量,通过第二锁资源对同一目标数据的操作数,通过抢占锁资源的形式来获得目标数据的操作权限,避免了大量请求同时请求操作同一目标数据时出现数据更新不一致,可以解决相关技术中数据库的数据不一致的技术问题。

[0082] 可选地,抢占单元可包括:确定模块,用于确定多个实例中的目标实例,其中,目标实例是为第一帐号分配的实例,多个实例是在服务器上生成的用于分配第一锁资源的实例;分配模块,用于将目标实例所属的第一锁资源分配给第一帐号。

[0083] 可选地,确定模块还可用于:在多个实例中存在已分配给第一帐号的实例的情况下,将已分配给第一帐号的实例作为目标实例;在多个实例中不存在已分配给第一帐号的实例的情况下,从多个实例中分配目标实例给第一帐号,其中,目标实例对应的锁标识与第一帐号的帐号标识匹配。

[0084] 可选地,分配模块还可用于:向目标实例发送第一抢占请求,其中,第一抢占请求用于请求目标实例为第一帐号分配第一锁资源;在接收到的目标实例的第一响应消息表明抢占成功的情况下,将第一响应消息所指示的第一锁资源分配给第一帐号。

[0085] 可选地,本申请的装置还可包括,生成单元,用于在确定多个实例中的目标实例之前,按照预设阈值生成多个实例,其中,预设阈值用于限定服务器并发执行的数据操作的并发量;保存多个实例和多个实例中每个实例对应的锁标识。

[0086] 可选地,抢占单元还可用于:向数据库发送第二抢占请求,其中,第二抢占请求用于请求数据库为第一帐号分配与目标数据对应的第二锁资源;在接收到的数据库的第二响应消息表明抢占成功的情况下,将第二响应消息所指示的第二锁资源分配给第一帐号。

[0087] 可选地,本申请的装置还可包括,释放单元,用于在对目标数据执行目标操作之

后，解除第一锁资源、第二锁资源与第一帐号之间的关联，其中，解除与第一帐号之间的关联的第一锁资源和第二锁资源允许被第二帐号抢占。

[0088] 此处需要说明的是，上述模块与对应的步骤所实现的示例和应用场景相同，但不限于上述实施例所公开的内容。需要说明的是，上述模块作为装置的一部分可以运行在如图1所示的硬件环境中，可以通过软件实现，也可以通过硬件实现，其中，硬件环境包括网络环境。

[0089] 根据本申请实施例的另一个方面，还提供了一种用于实施上述数据库的操作方法的服务器或终端。

[0090] 图6是根据本申请实施例的一种终端的结构框图，如图6所示，该终端可以包括：一个或多个(图6中仅示出一个)处理器601、存储器603、以及传输装置605，如图6所示，该终端还可以包括输入输出设备607。

[0091] 其中，存储器603可用于存储软件程序以及模块，如本申请实施例中的数据库的操作方法和装置对应的程序指令/模块，处理器601通过运行存储在存储器603内的软件程序以及模块，从而执行各种功能应用以及数据处理，即实现上述的数据库的操作方法。存储器603可包括高速随机存储器，还可以包括非易失性存储器，如一个或者多个磁性存储装置、闪存、或者其他非易失性固态存储器。在一些实例中，存储器603可进一步包括相对于处理器601远程设置的存储器，这些远程存储器可以通过网络连接至终端。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

[0092] 上述的传输装置605用于经由一个网络接收或者发送数据，还可以用于处理器与存储器之间的数据传输。上述的网络具体实例可包括有线网络及无线网络。在一个实例中，传输装置605包括一个网络适配器(Network Interface Controller,NIC)，其可通过网线与其他网络设备与路由器相连从而可与互联网或局域网进行通讯。在一个实例中，传输装置605为射频(Radio Frequency,RF)模块，其用于通过无线方式与互联网进行通讯。

[0093] 其中，具体地，存储器603用于存储应用程序。

[0094] 处理器601可以通过传输装置605调用存储器603存储的应用程序，以执行下述步骤：

[0095] 获取第一帐号的第一请求，其中，第一请求用于请求服务器对数据库中的目标数据执行目标操作，第一帐号用于在客户端上使用，数据库用于为客户端提供服务；

[0096] 在确定为第一帐号分配有第一锁资源的情况下，为第一帐号抢占第二锁资源，其中，第一锁资源用于限定服务器并发执行的数据操作的并发量小于等于预设阈值，第二锁资源和第一锁资源为不同类型的锁资源；

[0097] 在为第一帐号抢占到第二锁资源的情况下，对目标数据执行目标操作。

[0098] 处理器601还用于执行下述步骤：

[0099] 确定多个实例中的目标实例，其中，目标实例是为第一帐号分配的实例，多个实例是在服务器上生成的用于分配第一锁资源的实例；

[0100] 将目标实例所属的第一锁资源分配给第一帐号。

[0101] 采用本申请实施例，采用“获取第一帐号的第一请求，第一请求用于请求服务器对数据库中的目标数据执行目标操作，第一帐号用于在客户端上使用，数据库用于为客户端提供服务；在确定为第一帐号分配有第一锁资源的情况下，为第一帐号抢占第二锁资源，第

一锁资源用于限定服务器并发执行的数据操作的并发量小于等于预设阈值,第二锁资源和第一锁资源为不同类型的锁资源;在为第一帐号抢占到第二锁资源的情况下,对目标数据执行目标操作”的方式,通过第一锁资源的数量限定了每个服务器上的数据操作的并发量,通过第二锁资源对同一目标数据的操作数,通过抢占锁资源的形式来获得目标数据的操作权限,避免了大量请求同时请求操作同一目标数据时出现数据更新不一致,可以解决相关技术中数据库的数据不一致的技术问题。

[0102] 可选地,本实施例中的具体示例可以参考上述实施例中所描述的示例,本实施例在此不再赘述。

[0103] 本领域普通技术人员可以理解,图6所示的结构仅为示意,终端可以是智能手机(如Android手机、iOS手机等)、平板电脑、掌上电脑以及移动互联网设备(Mobile Internet Devices,MID)、PAD等终端设备。图6其并不对上述电子装置的结构造成限定。例如,终端还可包括比图6中所示更多或者更少的组件(如网络接口、显示装置等),或者具有与图6所示不同的配置。

[0104] 本领域普通技术人员可以理解上述实施例的各种方法中的全部或部分步骤是可以通过程序来指令终端设备相关的硬件来完成,该程序可以存储于一计算机可读存储介质中,存储介质可以包括:闪存盘、只读存储器(Read-Only Memory,ROM)、随机存取器(Random Access Memory,RAM)、磁盘或光盘等。

[0105] 本申请的实施例还提供了一种存储介质。可选地,在本实施例中,上述存储介质可以用于执行数据库的操作方法的程序代码。

[0106] 可选地,在本实施例中,上述存储介质可以位于上述实施例所示的网络中的多个网络设备中的至少一个网络设备上。

[0107] 可选地,在本实施例中,存储介质被设置为存储用于执行以下步骤的程序代码:

[0108] 获取第一帐号的第一请求,其中,第一请求用于请求服务器对数据库中的目标数据执行目标操作,第一帐号用于在客户端上使用,数据库用于为客户端提供服务;

[0109] 在确定为第一帐号分配有第一锁资源的情况下,为第一帐号抢占第二锁资源,其中,第一锁资源用于限定服务器并发执行的数据操作的并发量小于等于预设阈值,第二锁资源和第一锁资源为不同类型的锁资源;

[0110] 在为第一帐号抢占到第二锁资源的情况下,对目标数据执行目标操作。

[0111] 可选地,存储介质还被设置为存储用于执行以下步骤的程序代码:

[0112] 确定多个实例中的目标实例,其中,目标实例是为第一帐号分配的实例,多个实例是在服务器上生成的用于分配第一锁资源的实例;

[0113] 将目标实例所属的第一锁资源分配给第一帐号。

[0114] 可选地,本实施例中的具体示例可以参考上述实施例中所描述的示例,本实施例在此不再赘述。

[0115] 可选地,在本实施例中,上述存储介质可以包括但不限于:U盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0116] 上述本申请实施例序号仅仅为了描述,不代表实施例的优劣。

[0117] 上述实施例中的集成的单元如果以软件功能单元的形式实现并作为独立的产品

销售或使用时，可以存储在上述计算机可读取的存储介质中。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来，该计算机软件产品存储在存储介质中，包括若干指令用以使得一台或多台计算机设备(可为个人计算机、服务器或者网络设备等)执行本申请各个实施例所述方法的全部或部分步骤。

[0118] 在本申请的上述实施例中，对各个实施例的描述都各有侧重，某个实施例中没有详述的部分，可以参见其他实施例的相关描述。

[0119] 在本申请所提供的几个实施例中，应该理解到，所揭露的客户端，可通过其它的方式实现。其中，以上所描述的装置实施例仅仅是示意性的，例如所述单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口，单元或模块的间接耦合或通信连接，可以是电性或其它的形式。

[0120] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0121] 另外，在本申请各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用软件功能单元的形式实现。

[0122] 以上所述仅是本申请的优选实施方式，应当指出，对于本技术领域的普通技术人员来说，在不脱离本申请原理的前提下，还可以做出若干改进和润饰，这些改进和润饰也应视为本申请的保护范围。

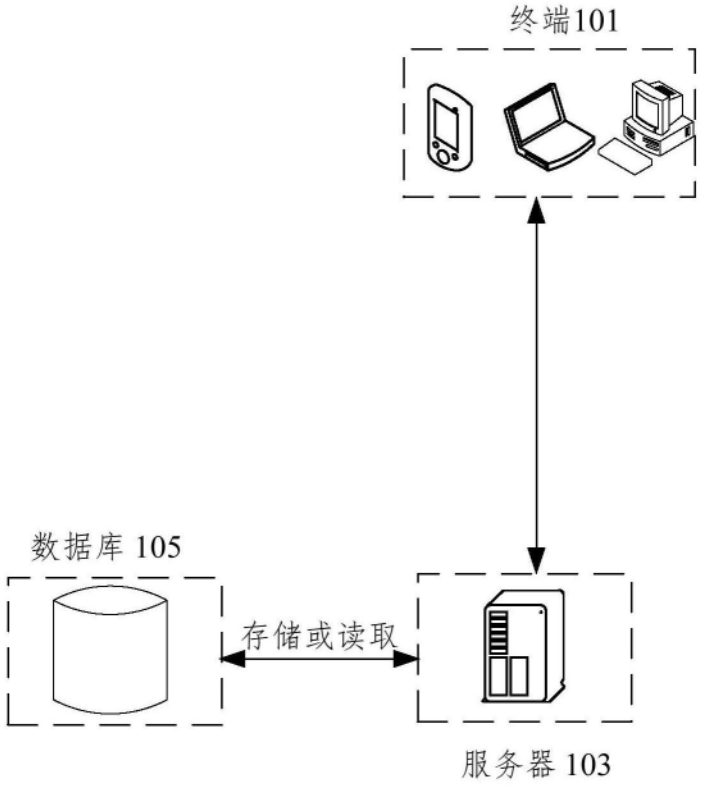

终端101
数据库 105
存储或读取
服务器 103

图1

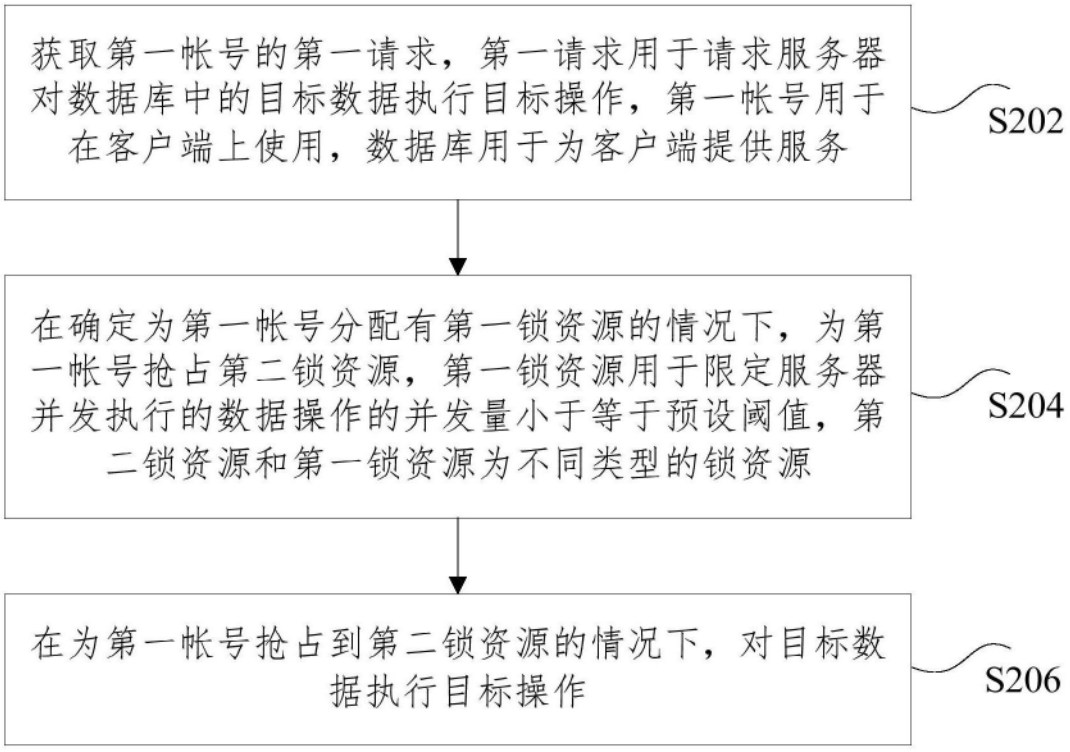

获取第一帐号的第一请求，第一请求用于请求服务器对数据库中的目标数据执行目标操作，第一帐号用于在客户端上使用，数据库用于为客户端提供服务
S202
在确定为第一帐号分配有第一锁资源的情况下，为第一帐号抢占第二锁资源，第一锁资源用于限定服务器并发执行的数据操作的并发量小于等于预设阈值，第二锁资源和第一锁资源为不同类型的锁资源
S204
在为第一帐号抢占到第二锁资源的情况下，对目标数据执行目标操作
S206

图2

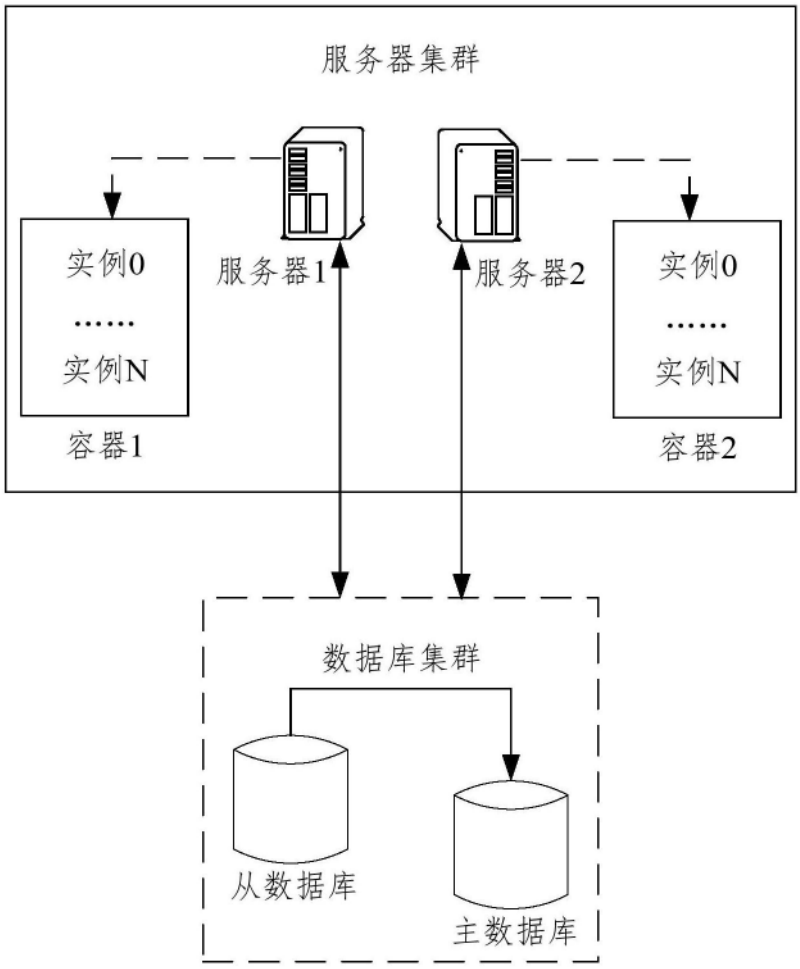

服务器集群
实例0
……
实例N
容器1
服务器1
服务器2
实例0
……
实例N
容器2
数据库集群
从数据库
主数据库

图3

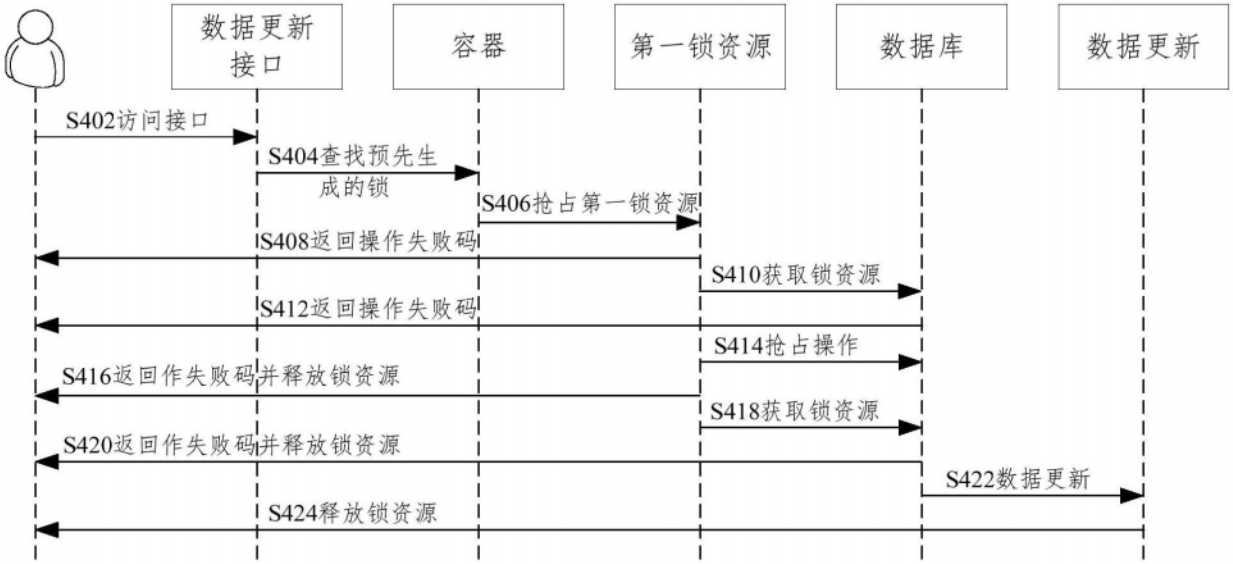

数据更新接口
容器
第一锁资源
数据库
数据更新
S402访问接口
S404查找预先生成的锁
S406抢占第一锁资源
S408返回操作失败码
S410获取锁资源
S412返回操作失败码
S414抢占操作
S416返回作失败码并释放锁资源
S418获取锁资源
S420返回作失败码并释放锁资源
S422数据更新
S424释放锁资源

图4

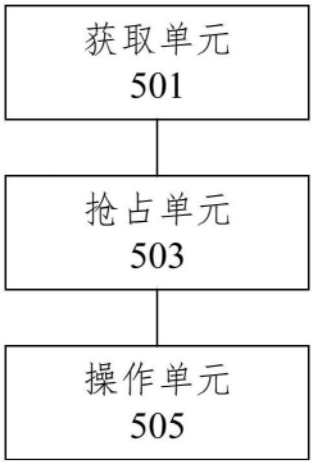


图5

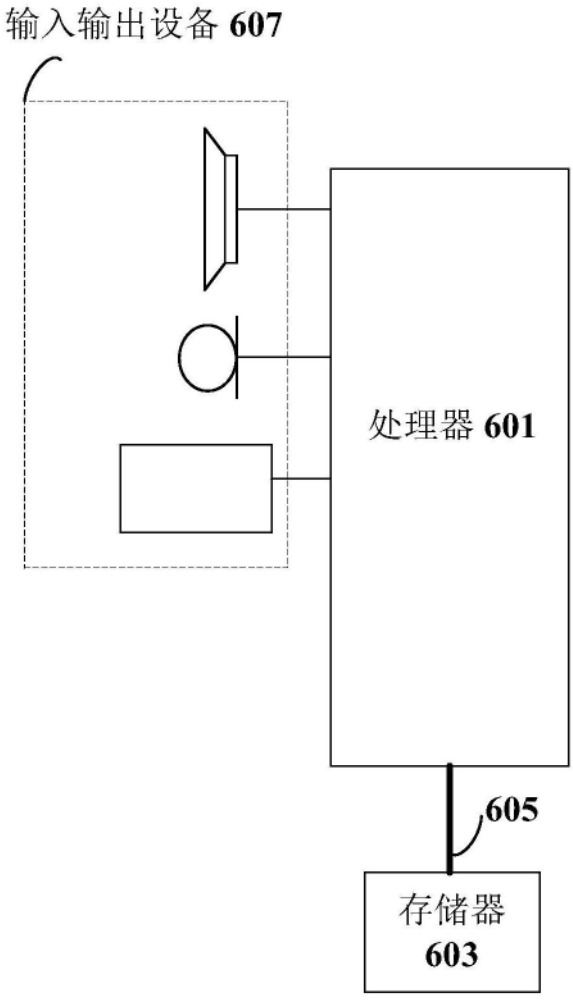


图6