



(12) 发明专利申请

(10) 申请公布号 CN 116257366 A

(43) 申请公布日 2023. 06. 13

(21) 申请号 202111498169.0

(22) 申请日 2021.12.09

(71) 申请人 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 廖志坚 余博伟

(74) 专利代理机构 北京同达信恒知识产权代理有限公司 11291

专利代理师 落爱青

(51) Int. Cl.

G06F 9/52 (2006.01)

G06F 9/54 (2006.01)

G06F 9/50 (2006.01)

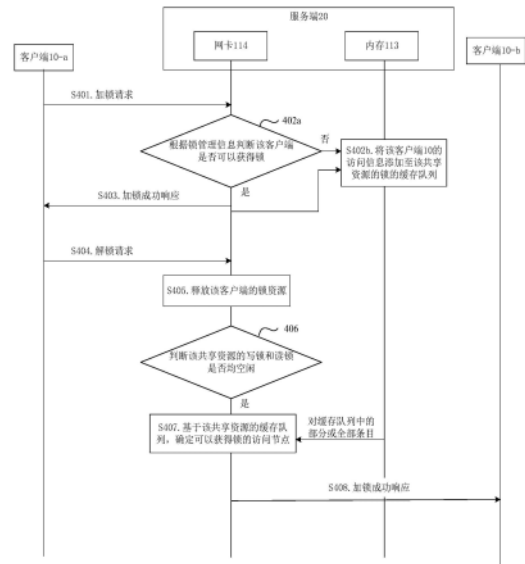
权利要求书2页 说明书14页 附图3页

(54) 发明名称

一种锁管理方法、装置及系统

(57) 摘要

本申请提供了一种锁管理方法、装置及系统。根据该方法，在服务端设备的网卡获取客户端的锁请求后，网卡根据记录的锁信息判断该客户端是否能够获得该共享资源的锁，其中，该锁请求用于请求获得共享资源的访问权限。网卡在确定该客户端能够获得该共享资源的锁的情况下，向该客户端发送加锁成功响应。本申请中，网卡确定客户端是否可以获得共享资源的锁，不需要服务端设备的CPU参与，降低了CPU的占用率，有利于提高CPU的资源利用率，提高响应速度和系统性能。



1. 一种锁管理方法,其特征在于,所述方法包括:

服务端设备的网卡获取客户端的加锁请求,所述加锁请求用于请求获得共享资源的访问权限;

所述网卡根据记录的锁信息判断所述客户端是否能够获得所述共享资源的锁;

所述网卡在确定所述客户端能够获得所述共享资源的锁的情况下,向所述客户端发送加锁成功响应,所述加锁成功响应用于指示所述客户端成功获得所述共享资源的锁。

2. 如权利要求1所述的方法,其特征在于,所述锁信息包括请求获得所述共享资源的写锁的第一数量以及请求获得所述共享资源的读锁的第二数量;

所述网卡根据记录的锁信息判断所述客户端是否能够获得所述共享资源的锁,包括:

当所述第一数量指示所述共享资源的写锁为空闲状态时,所述网卡确定所述客户端能够获得所述共享资源的读锁;或

当所述第一数量指示所述共享资源的写锁为空闲状态,且所述第二数量指示所述共享资源的读锁为空闲状态时,所述网卡确定所述客户端能够获得所述共享资源的写锁。

3. 如权利要求1所述的方法,其特征在于,所述锁信息包括锁状态信息、节点状态信息;其中,所述节点状态信息用于指示访问节点是否已获得锁;所述锁状态信息用于指示所述访问节点所申请的锁是否为解锁状态;

所述网卡根据所述锁信息判断所述客户端是否能够获得所述共享资源的锁,包括:

所述网卡根据所述节点状态信息、所述锁状态信息确定已获得所述共享资源的锁的访问节点均为解锁状态,则所述网卡确定所述客户端能够获得所述共享资源的锁。

4. 如权利要求3所述的方法,其特征在于,所述加锁成功响应包括锁位置信息,所述锁位置信息用于指示所述客户端的锁信息在所述缓存队列中的位置;

所述方法还包括:

所述网卡接收所述客户端发送的解锁请求,所述解锁请求用于请求释放所述共享资源的锁,所述解锁请求包括所述锁位置信息;

所述网卡根据所述锁位置信息,将所述缓存队列中的所述锁位置信息指示的锁的状态修改为解锁状态。

5. 如权利要求1-3任一项所述的方法,其特征在于,还包括:

所述网卡确定所述客户端发生故障,则释放所述客户端的锁。

6. 如权利要求1-5任一项所述的方法,其特征在于,所述服务端设备的网卡获取所述客户端的加锁请求,包括:

所述网卡接收所述客户端发送的加锁请求;或

所述网卡从缓存队列中获取所述客户端的加锁请求。

7. 一种锁管理装置,其特征在于,所述装置包括:

获取模块,用于获取客户端的加锁请求,所述加锁请求用于请求获得共享资源的访问权限;

处理模块,用于根据记录的锁信息判断所述客户端是否能够获得所述共享资源的锁;在确定所述客户端能够获得所述共享资源的锁的情况下,通过发送模块向所述客户端发送加锁成功响应,所述加锁成功响应用于指示所述客户端成功获得所述共享资源的锁。

8. 如权利要求7所述的装置,其特征在于,所述锁信息包括请求获得所述共享资源的写

锁的第一数量以及请求获得所述共享资源的读锁的第二数量；

所述处理模块在根据锁信息判断所述客户端是否能够获得所述共享资源的锁时，具体用于：当所述第一数量指示所述共享资源的写锁为空闲状态时，确定所述客户端能够获得所述共享资源的读锁；或

当所述第一数量指示所述共享资源的写锁为空闲状态，且所述第二数量指示所述共享资源的读锁为空闲状态时，确定所述客户端能够获得所述共享资源的写锁。

9. 如权利要求7所述的装置，其特征在于，所述锁信息包括锁状态信息、节点状态信息；其中，所述节点状态信息用于指示访问节点是否已获得锁；所述锁状态信息用于指示所述访问节点所申请的锁是否为解锁状态；

所述处理模块在根据所述锁信息判断所述客户端是否能够获得所述共享资源的锁时，具体用于：根据所述节点状态信息、所述锁状态信息确定已获得所述共享资源的锁的访问节点均为解锁状态，则所述网卡确定所述客户端能够获得所述共享资源的锁。

10. 如权利要求9所述的装置，其特征在于，所述加锁成功响应包括锁位置信息，所述锁位置信息用于指示所述客户端的锁信息在所述缓存队列中的位置；

所述获取模块，还用于接收所述客户端发送的解锁请求，所述解锁请求用于请求释放所述共享资源的锁，所述解锁请求包括所述锁位置信息；

所述处理模块，还用于根据所述锁位置信息，将所述缓存队列中的所述锁位置信息指示的锁的状态修改为解锁状态。

11. 如权利要求7-9任一项所述的装置，其特征在于，所述处理模块，还用于确定获得锁的所述客户端发生故障，则释放所述客户端的锁。

12. 如权利要求7-11任一项所述的装置，其特征在于，所述获取模块获取客户端的加锁请求时，具体用于：接收所述客户端发送的加锁请求；或从缓存队列中获取所述客户端的加锁请求。

13. 一种锁管理装置，其特征在于，所述装置包括处理器和存储器；

所述存储器存储计算机程序指令；

所述处理器执行所述存储器中的计算机程序指令，以实现如权利要求1至6中任一项所述的方法。

14. 一种计算机可读存储介质，其特征在于，包括程序代码，所述程序代码包括的指令用于实现如权利要求1至6中任一项所述的方法。

15. 一种计算设备，其特征在于，所述计算设备包括锁管理装置；

所述锁管理装置，用于执行如权利要求1至6任一所述的方法。

16. 一种分布式系统，其特征在于，所述系统包括客户端和服务端设备，所述服务端设备包括锁管理装置；

所述客户端，用于向所述计算设备发送加锁请求，所述加锁请求用于请求获得共享资源的访问权限；

所述服务端设备的所述锁管理装置，用于执行如权利要求1至6任一所述的方法。

## 一种锁管理方法、装置及系统

### 技术领域

[0001] 本申请涉及计算机技术领域,尤其涉及一种锁管理方法、装置及系统。

### 背景技术

[0002] 锁是实现多个对象对同一共享资源的有序化访问的一种基本控制机制,该多个对象例如可以是同一个分布式系统内的多个计算设备。

[0003] 目前,一种基于锁的数据访问方式包括:为每个共享资源设置一个对应的锁,访问者需要先获得锁,才可以对共享资源进行访问,访问完成之后,再释放锁,从而保证多个访问者能够有序地访问共享资源。

[0004] 以分布式系统为例,假设该分布式系统包括客户端和服务端设备,当客户端发起对某共享资源的访问之前,可以首先向服务端设备发送用于获得该共享资源访问权限的加锁请求,服务端通过网卡接收该加锁请求,并由网卡将该加锁请求发送至服务端设备的处理器,然后由处理器对该锁执行加锁操作,锁的释放操作也是相同的流程,这里不再赘述。上述操作,不仅占用了较多处理器的计算资源、网络 and I/O 资源,而且处理器的处理速度还会影响数据处理请求,导致整个数据处理请求无法满足高性能的业务要求。

### 发明内容

[0005] 本申请提供一种锁管理方法、装置及系统,用于降低CPU的处理操作,提高系统性能。

[0006] 第一方面,本申请实施例提供了一种锁管理方法,该方法可以应用于分布式系统中,该分布式系统包括客户端和服务端设备。在该方法中,服务端设备的网卡获取客户端的加锁请求,该加锁请求用于请求获得共享资源的访问权限;网卡根据记录的该共享资源的锁信息判断该客户端是否能够获得该共享资源的锁,网卡在确定该客户端能够获得该共享资源的锁时,向该客户端发送加锁成功响应,该加锁成功响应用于指示客户端成功获得该共享资源的访问权限。

[0007] 通过上述方法,网卡获取客户端的加锁请求,并根据记录的锁信息判断该客户端是否能够获得共享资源的锁,由网卡确定客户端是否可以获得共享资源的锁,不需要CPU参与,降低了对CPU的占用率,有利于提高CPU的资源利用率。更进一步地,网卡在确定能够获得时,可以直接向客户端发送加锁成功响应,不需要CPU来指示该操作,因此,可以提高对加锁请求的响应速度,缩短时延,提高系统性能。

[0008] 在一种可能的实施方式中,网卡确定请求获得共享资源的锁的客户端时,一种方式可以为网卡接收该客户端的加锁请求。另一种方式可以为网卡从共享资源对应的缓存队列中获取该客户端的加锁请求。

[0009] 通过上述方法,网卡可以直接响应客户端发送来的加锁请求,也可以暂时将加锁请求放入缓存队列,网卡可以从缓存队列中获得加锁请求,从而提供管理加锁请求的灵活性。

[0010] 在一种可能的实施方式中,该共享资源的锁信息包括请求获得该共享资源的写锁的节点的数量(记为第一数量)以及请求获得该共享资源的读锁的节点的数量(记为第二数量);

[0011] 网卡根据锁信息判断该客户端是否能够获得所述共享资源的锁,包括:若该客户端的加锁请求用于请求获得共享资源的读操作权限(即读锁),当第一数量指示共享资源的写锁为空闲状态时,比如第一数量为0,也即请求获得该共享资源的写锁的节点的数量为0(仅为一种示例),网卡确定该客户端能够获得共享资源的读锁;或若该客户端的加锁请求用于请求获得共享资源的写操作权限(即写锁),当第一数量指示该共享资源的写锁为空闲状态,且该第二数量指示所述共享资源的读锁为空闲状态时,比如第二数量为0,也即请求获得该共享资源的读锁的节点的数量为0(仅为一种示例),网卡确定该客户端能够获得该共享资源的写锁。

[0012] 通过上述方法,网卡接收到客户端的加锁请求之后,可以基于第一数量、第二数量快速判断该客户端是否能够获得锁,提高响应速度,并且在本申请中,由网卡来确定客户端是否可以获得共享资源的锁,不需要CPU参与,降低了对CPU的占用率,有利于提高CPU的资源利用率。

[0013] 在一种可能的实施方式中,网卡从所述共享资源的锁的缓存队列中顺序获取所述客户端的加锁请求,其中,所述缓存队列包括请求获得所述共享资源的锁的每一访问节点的加锁请求,所述加锁请求包括所述访问节点的标识,标识用于唯一标识该访问节点;访问节点包括客户端。

[0014] 通过上述方法,网卡接收到多个客户端的加锁请求时,可以将这些加锁请求缓存在缓存队列中,这样,网卡可以通过缓存队列对请求获得共享资源的锁的一个或多个节点的加锁请求进行有序管理,并且,客户端不需要向服务端设备多次发送加锁请求来获得锁,从而节省网络资源,减少带宽占用。另外,网卡可以顺序从缓存队列中获取加锁请求,以为缓存队列中排队的节点加锁,减少了无序加锁的情况,减少请求获得共享资源的锁的节点因长期得不到锁而被饿死的情况。

[0015] 在一种可能的实施方式中,所述锁信息还包括节点状态信息、锁状态信息;其中,所述节点状态信息用于指示访问节点是否已获得锁;所述锁状态信息用于指示访问节点所请求的锁是否为解锁状态;网卡根据所述锁信息判断客户端是否能够获得所述共享资源的锁,包括:网卡根据节点状态信息、锁状态信息确定已获得共享资源的锁的访问节点均为解锁状态,则网卡确定客户端能够获得该共享资源的锁。

[0016] 通过上述方法,网卡可以根据锁信息确定已获得共享资源的锁的节点,以及这些节点是否已释放锁,从而确定客户端的所请求的锁是否为空闲状态,进一步确定该客户端是否可以获得共享资源的锁,这种方式不需要CPU参与,降低CPU的占用率。

[0017] 在一种可能的实施方式中,该加锁成功响应包括锁位置信息,锁位置信息用于指示该客户端的锁信息在共享资源的缓存队列中的位置;在该方法中,网卡还可以接收客户端发送的解锁请求,该解锁请求用于请求释放所述共享资源的锁,该解锁请求包括该锁位置信息;网卡根据该锁位置信息,将缓存队列中的锁位置信息指示的锁的状态修改为解锁状态。

[0018] 通过上述方法,网卡根据解锁请求中携带的锁位置信息可以快速定位到缓存队列

的对应锁信息,并将其中记录的锁的状态修改为解锁状态,从而为该客户端解锁,提高解锁速度。

[0019] 在一种可能的实施方式中,网卡确定获得锁的客户端发生故障,则释放该客户端的锁。

[0020] 通过上述方法,网卡检测到获得锁的客户端发生故障,可以主动释放客户端的锁,避免发生故障的客户端长期占有锁,可以提高系统性能。

[0021] 第二方面,本申请实施例还提供了一种锁管理装置,该锁管理装置具有实现上述第一方面的方法实例中网卡的功能,有益效果可以参见第一方面的描述此处不再赘述。所述功能可以通过硬件实现,也可以通过硬件执行相应的软件实现。所述硬件或软件包括一个或多个与上述功能相对应的模块。在一个可能的设计中,所述锁管理装置的结构中包括获取模块、处理模块,这些模块可以执行上述第一方面方法示例中的相应功能,具体参见方法示例中的详细描述,此处不做赘述,有益效果可以参见第一方面的描述此处不再赘述。

[0022] 第三方面,本申请实施例还提供了一种锁管理装置,该锁管理装置具有实现上述第一方面的方法实例中网卡的功能,有益效果可以参见第一方面的描述此处不再赘述。所述装置的结构中包括处理器和存储器,可选的,还可以包括通信接口。所述处理器被配置为支持所述锁管理装置执行上述第一方面方法中相应的功能。所述存储器与所述处理器耦合,其保存所述通信装置必要的计算机程序指令和数据(如共享资源的锁信息)。所述锁管理装置的结构中还包括通信接口,用于与其他设备进行通信,如可以接收加锁请求,有益效果可以参见第一方面的描述此处不再赘述。

[0023] 第四方面,本申请还提供一种计算机可读存储介质,所述计算机可读存储介质中存储有指令,当其在计算机上运行时,使得计算机执行上述第一方面以及第一方面的各个可能的实施方式中网卡的方法,有益效果可以参见第一方面的描述此处不再赘述。

[0024] 第五方面,本申请还提供一种包含指令的计算机程序产品,当其在计算机上运行时,使得计算机执行上述第一方面以及第一方面的各个可能的实施方式中网卡的方法,有益效果可以参见第一方面的描述此处不再赘述。

[0025] 第六方面,本申请还提供一种计算机芯片,所述芯片与存储器相连,所述芯片用于读取并执行所述存储器中存储的软件程序,执行上述第一方面以及第一方面的各个可能的实施方式中网卡的方法,有益效果可以参见第一方面的描述此处不再赘述。

[0026] 第七方面,本申请实施例还提供了一种计算设备,该计算设备包括锁管理装置,锁管理装置具有实现上述第一方面的方法实例中网卡的功能,有益效果可以参见第一方面的描述此处不再赘述。

[0027] 第八方面,本申请实施例还提供了一种分布式系统,该分布式系统包括客户端和服务端设备,所述服务端设备包括锁管理装置;客户端,用于向所述服务端设备发送加锁请求,所述加锁请求用于请求获得共享资源的访问权限;服务端设备的锁管理装置具有实现上述第一方面的方法实例中网卡的功能,有益效果可以参见第一方面的描述此处不再赘述。

[0028] 本申请在上述各方面提供的实现方式的基础上,还可以进行进一步组合以提供更多实现方式。

## 附图说明

- [0029] 图1为本申请实施例提供的一种分布式系统架构示意图；  
[0030] 图2为本申请实施例提供的一种服务端20的结构示意图；  
[0031] 图3为本申请实施例提供的一种锁管理信息的帧结构示意图；  
[0032] 图4为本申请实施例提供的一种锁管理方法的流程示意图；  
[0033] 图5为本申请实施例提供的一种锁管理装置的结构示意图；  
[0034] 图6为本申请实施例提供的另一种锁管理装置的结构示意图。

## 具体实施方式

[0035] 为了便于理解本申请实施例所提供的锁管理的方法,首先对本申请实施例所涉及的概念和术语进行简单说明。

[0036] 1) 共享资源,可以被多个对象共享的资源,如允许被分配给多个对象共享的数据或数据所在的存储空间可以称为一种共享资源,上述多个对象均可以对共享数据发起读或写等访问操作。其中,该多个对象可以是同一通信系统中的不同计算机设备,或者该多个对象可以指同一计算机设备中的多个进程或多个线程。

[0037] 当多个对象发起对同一共享资源的访问操作时,通常利用锁来控制多个对象对共享资源的有序化访问。

[0038] 2) 锁,是与共享资源相对应的概念,用于控制共享资源的访问权限。对共享资源的访问操作包括读操作和写操作,与之对应的,用于控制读操作的锁资源可以称为读锁,用于控制写操作的锁资源称为写锁,执行写操作的对象可以称为写者,执行读操作的对象称为读者。具体的,写者需要先获得共享资源的写锁,才能够对共享资源执行写操作。同理,读者需要获得共享资源的读锁,才能对共享资源执行读操作。写锁具有排他性,一次仅允许一个写者获得,也即一次仅允许一个写者对共享资源执行写操作;而读锁具有共享性,一次可以由多个读者获得,也即一次允许多个读者同时读取共享资源对应的数据。并且,读锁和写锁不能同时被分配,即同一时间仅允许一个写者对同一共享资源执行写操作,或允许多个读者对该共享资源执行读操作,以此避免数据不一致的问题。

[0039] 锁具有至少两种状态,分别为加锁状态和解锁状态,对象获得锁,则该锁为加锁状态,获得锁的对象执行完访问操作之后再释放锁,则该对象的锁为解锁状态。由于写锁只能一个写者获得,因此写锁的解锁状态也可以称为空闲状态,而对于读锁而言,只有当所有获得读锁的读者均释放读锁之后,该读锁的状态才可以称为空闲状态。

[0040] 为了使本申请的目的、技术方案和优点更加清楚,下面将结合附图对本申请作进一步地详细描述。方法实施例中的具体操作方法也可以应用于装置实施例或系统实施例中。

[0041] 图1为本申请实施例提供的一种分布式系统的架构示意图。该系统包括一个或多个客户端10(图1中示出了三个客户端10,分别为10-a、10-b、10-c,但不限于三个客户端10)和服务端20。

[0042] 客户端10,可以是部署在用户侧的一种计算设备,或计算设备上的软件或服务,其中,计算设备包括服务器(如应用服务器)、台式计算机、笔记本电脑或移动终端等。用户可以通过客户端10向服务端20发送请求,如用于请求获得共享资源的锁的加锁请求,或用于

请求释放共享资源的锁的解锁请求。客户端10和服务端20可以通过网络150通信。网络150可以为有线或无线通信方式。示例地,网络150通常表示任何电信或计算机网络,包含例如企业内部网、广域网(wide area network,WAN)、局域网(local area network,LAN)、个域网(personal area network,PAN)、因特网或无线网络(如WIFI、第5代(5<sup>th</sup> Generation,5G)通信技术)。具体的,客户端10可以使用多种网络协议与服务端20通信,例如传输控制协议/网际协议(transmission control protocol/internet protocol,TCP/IP)协议、用户数据报(user datagram protocol,UDP)/IP协议、远程内存直接访问(remote direct memory access,RDMA)协议等。除了通过上述网络150的示例外,客户端10和服务端20也可以通过光纤交换机进行通信。或者,光纤交换机也可以替换成以太网交换机、无限带宽(InfiniBand)交换机、基于融合以太网的远程直接内存访问(RDMA over converged ethernet,RoCE)交换机等。

[0043] 服务端20可以是一种计算设备,如服务器、台式计算机等设备。服务端20可以用于管理共享资源的锁,例如为请求加锁的客户端分配锁资源、以及为请求释放锁的客户端解锁等。

[0044] 请参见图2,图2为本申请实施例提供的服务端20的硬件结构示意图,如图2所示,该服务端20至少包括处理器112、内存113和网卡114。其中,处理器112、内存113和网卡114之间通过总线115连接。总线115包括但不限于:高速串行计算机扩展总线标准(peripheral component interconnect express,PCIe)总线、双数据速率(double data rate,DDR)总线、支持多协议的互联总线、串行高级技术附件(serial advanced technology attachment,SATA)总线和串行连接SCSI(serial attached scsi,SAS)总线、控制器局域网络总线(controller area network,CAN)、计算机标准连接(computer express link,CXL)标准总线、缓存一致互联协议(cache coherent interconnect for accelerators,CCIX)总线等。示例性地,处理器112与内存113之间可以通过DDR总线连接,网卡114与内存113可以通过但不限于:多协议互联总线连接、PCIe总线、CCIX总线连接。本申请实施例服务端对各组件之间所使用的总线协议不做限定,任何可以令组件之间通信的连接方式或通信协议均适用于本申请实施例。

[0045] 处理器112可以是一个中央处理器(central processing unit,CPU)、专用集成电路(application specific integrated circuit,ASIC)、现场可编程门阵列(field programmable gate array,FPGA)、人工智能(artificial intelligence,AI)芯片、片上系统(system on chip,SoC)或复杂可编程逻辑器件(complex programmable logic device,CPLD)、图形处理器(graphics processing unit,GPU)、神经网络处理单元(neural network processing unit,NPU)等。

[0046] 内存113,是指与处理器112直接交换数据的内部存储器,它可以随时读写数据,而且速度很快,作为操作系统或其他正在运行中的程序的临时数据存储器。内存113包括至少两种存储器,例如内存113既可以是随机存取存储器,也可以是只读存储器(read only memory,ROM)。举例来说,随机存取存储器是动态随机存取存储器(dynamic random access memory,DRAM),或者存储级存储器(storage class memory,SCM)。内存还可以包括其他随机存取存储器,例如静态随机存取存储器(static random access memory,SRAM)等。而对于只读存储器,举例来说,可以是可编程只读存储器(programmable read only memory,

PROM)、可抹除可编程只读存储器(erasable programmable read only memory,EPR0M)等。另外,内存113还可以是双列直插式存储器模块或双线存储器模块(dual in-line memory module,DIMM),即由DRAM组成的模块。实际应用中,服务端20中可配置多个内存113,以及不同类型的内存113。本实施例不对内存113的数量和类型进行限定。

[0047] 本申请实施例中,内存113可以用于存储一个或多个共享资源的锁的缓存队列(图2仅示出一个共享资源的锁的缓存队列,但本申请对此不做限定),该缓存队列用于存放请求获得该共享资源的锁的访问节点(如客户端10)的访问信息,如图2中,该缓存队列中顺序存储有一个写者的访问信息和两个读者的访问信息,访问信息用于记录访问节点的信息、访问节点所请求的锁的类型(如读锁、写锁)等信息,下文会进行详细介绍,此处不做重复说明。需要说明的是,图2所示的缓存队列位于内存113的环形缓冲区仅是示例,本申请中缓存队列还可以是其他形式存在于内存113中,本申请实施例对此不做限定。

[0048] 网卡114,用于与服务端20外部或服务端20内部的组件之间进行通信。在硬件上,继续参见图2所示,网卡114包括锁管理模块1141、存储器1142和通信接口1143。其中,锁管理模块1141、存储器1142以及通信接口1143之间可以通过总线连接,这里的总线可以参见对总线115的描述,此处不再赘述。

[0049] 其中,锁管理模块1141,可以为硬件逻辑电路、处理核、ASIC、AI芯片或可编程逻辑器件(programmable logic device,PLD)实现,上述PLD可以是复杂程序逻辑器件(complex programmable logical device,CPLD),现场可编程门阵列(field-programmable gate array,FPGA),通用阵列逻辑(generic array logic,GAL)、片上系统(system on chip,SoC)或其任意组合。或者锁管理模块1141也可以为软件模块,或软件模块和硬件模块的组合。

[0050] 锁管理模块1141能够通过通信接口1143与客户端10通信,如通信接口1143可以用于接收客户端10发送的加锁请求或解锁请求。或与服务端20中的其他组件(如处理器112或内存113)交互,如通信接口1143可以通过总线115读取内存113中的数据(如缓存队列中的访问信息)。示例性地,网卡114可以通过直接存储器访问(direct memory access,DRM)协议读取内存113中锁的缓存队列所包括的信息。

[0051] 在本申请中,锁管理模块1141,还可以用于根据接收到的加锁请求/解锁请求生成、管理共享资源的缓存队列,如网卡114接收到客户端10-a发送加锁请求后,锁管理模块1141根据该加锁请求,在对应共享资源的锁的缓存队列中插入该客户端10-a的访问信息。若锁管理模块1141接收到多个访问节点针对该共享资源的加锁请求,则可以按照接收加锁请求的顺序,将各访问节点的访问信息顺序添加至该共享资源对应的缓存队列中。举例来说,网卡114顺序接收到客户端10-a、客户端10-b发送的加锁请求,则锁管理模块1141顺序将客户端10-a的访问信息、客户端10-b的访问信息添加至该缓存队列中,其中,客户端10-b的访问信息位于客户端10-a的访问信息之后,下文会对访问信息进行详细介绍,此处不做赘述。锁管理模块1141还可以根据接收到的解锁请求,来释放缓存队列中对应客户端的锁资源,下文均会进行详细介绍。

[0052] 下文会对访问信息进行详细介绍,此处不做赘述。锁管理模块1141可以按照接收加锁请求的顺序,将各访问节点的访问信息顺序添加至访问队列中。锁管理模块1141还可以根据网卡114接收到的解锁请求,来释放访问队列中对应条目的锁资源。

[0053] 存储器1142,是指可以与锁管理模块1141直接交换数据的内部存储器,它可以随时读写数据,而且速度很快,作为运行在锁管理模块1141上正在运行中的程序的临时数据存储单元。存储器1142的硬件组件与内存113类似,可以参见前述对内存113的介绍。

[0054] 在一种可选的实施方式中,存储器1142可以用于存储共享资源对应的锁管理信息。图3为本申请实施例提供的一种锁管理信息的帧结构示意图。如图3所示,该锁管理信息包括但不限于下列参数中的一项或多项:读者的数量(Rlock\_cnt)、写者的数量(Wlock\_cnt)、缓存队列的头部指针(C\_cnt)、缓存队列的队尾指针(P\_cnt)。其中,读者的数量是指请求获得该共享资源的读锁的读者的数量,示例性地,每增加一个读者,则该计数+1,每减少一个读者,该计数-1。如在一种实施方式中,每当网卡114接收到一个请求获得读锁的加锁请求,则该计数+1,值得注意的是,同一个客户端的重复的加锁请求仅记录一次。同理,当网卡114接收到读者发送的一个解锁请求,则在解锁成功后,将该计数-1。类似的,写者的数量是指请求获得共享资源的写锁的写者的数量。该参数的计数方式参见上文的描述,此处不再赘述。该锁管理信息可以由锁管理模块1141管理。示例性地,该锁管理信息中每个参数的大小可以均为16bit(仅为一种示例,本申请对此不做限定),也即锁管理信息为一个64bit的原子数,由于该锁管理信息的数据量较小,不会占用较多的存储空间,存储器1142有能力存储该锁管理信息,网卡114可能不需要扩容即可实现该功能,这一点使得该设计具有较强的应用性、兼容性。

[0055] 需要说明的是,(1)图1是以客户端10为台式计算机、服务端20是服务器为例示出的,实际上,本申请实施例对客户端10和服务端20的具体形态不做限定。(2)图2所示的服务端20的结构仅为示例,在实际产品中,服务端20或网卡114可能具有更多或更少的组件,如服务端20还可以包括键盘、鼠标、显示屏等输入/输出设备以及硬盘,硬盘如非易失性存储器(non-volatile memory),例如ROM,硬盘驱动器(hard disk drive,HDD)或固态驱动器(solid state disk,SSD)等。又如,网卡114也可以包括前述的硬盘。本申请实施例对服务端20以及网卡114的具体结构不做限定。(3)图2所示的缓存队列存储于内存113中仅为一种示例。在另一种可选的实施方式中,缓存队列还可以存储于网卡的存储器1142中,可以理解的是,这种方式对存储器1142的容量具有一定的要求。在一种实施方式中,可以将网卡114频繁访问的缓存队列放置在存储器1142中,以此提高锁管理模块1141获取缓存队列的速度,减少时延。

[0056] 接下来,以图1所示的系统、图2所示的服务端20的结构为例,对本申请实施例提供的锁管理方法进行说明。为了便于描述,本申请的以下实施例,以服务端20的网卡114执行本申请实施例提供的锁管理方法为例进行说明。在该锁管理方法中,将从为客户端10加锁(步骤401~步骤403或步骤406~步骤408),以及为客户端10解锁(步骤404~步骤405)两个方面进行描述。

[0057] 图4为本申请实施例提供的一种锁管理方法的流程示意图,如图4所示,该方法包括如下步骤:

[0058] 步骤401,网卡114接收客户端10(如客户端10-a)发送的加锁请求。

[0059] 该加锁请求用于请求获得共享资源的锁,具体的,该加锁请求携带的信息包括但不限于下列中的一项或多项:该客户端10所请求的锁的类型(如写锁或读锁)、该客户端10的节点标识(用于唯一标识一个客户端)、锁的地址信息(如该锁的缓存队列在内存的地

址)。

[0060] 步骤402a,网卡114基于锁管理信息判断该客户端10是否能够获得该共享资源的锁,如果可以,则执行步骤403。

[0061] 网卡114可以基于锁管理信息中的Rlock\_cnt和Wlock\_cnt判断该客户端10是否可以获得锁。示例性地,若该加锁请求用于请求获得共享资源的写锁,若Wlock\_cnt指示该写锁为空闲状态,且Rlock\_cnt指示该共享资源的读锁为空闲状态,则确定该客户端10可以获得该写锁。再示例性地,若该加锁请求用于请求获得该共享资源的读锁,由于读锁可以共享,若Wlock\_cnt指示该共享资源的写锁为空闲状态,换言之只要该共享资源的写锁为空闲状态,即可以确定该客户端10可以获得该读锁。

[0062] 具体的,若Wlock\_cnt用于记录该共享资源对应的缓存队列中写者的数量,若缓存队列中写者的数量为0,则表示该共享资源的写锁为空闲状态,同理,若Rlock\_cnt用于记录该缓存队列中读者的数量,若读者的数量为0,则表示该共享资源的读锁为空闲状态。

[0063] 步骤402b,网卡114根据该客户端10的加锁请求更新该共享资源的锁的缓存队列。

[0064] 在本申请中,不论该客户端10是否可以获得该共享资源的锁,网卡114均可以将该客户端10的访问信息添加至该共享资源的缓存队列中,这一操作可以称为入列。为便于理解,在介绍入列过程之前,首先对缓存队列进行详细介绍,如前所述,缓存队列用于顺序排列一个或多个请求获得共享资源的锁的访问节点的访问信息,每个访问信息可以包括但不限于下列中的一项或多项:

[0065] 1) 节点标识:访问节点(如客户端)的标识,用于唯一标识一个访问节点。

[0066] 2) 锁类型:访问节点请求获得的锁的类型,包括读锁、写锁。

[0067] 3) 节点状态信息:用于指示访问节点是否已获得所请求的锁,包括已获得(acquired)、未获得(not acquire)。

[0068] 4) 锁状态信息:用于指示访问节点是否已释放锁,包括:加锁状态(active)、解锁状态(inactive)。

[0069] 5) 位置标识(或称为位置索引):用于指示该访问信息在环形缓冲区中的位置。如该位置标识可以是环形缓冲区中的条目的索引,假设该环形缓冲区最多有8个条目,每个条目用于存储一个访问信息,条目的编号可以从一个预设的初始值如0开始编制,每个条目的编号依次加1,如8个条目的编号顺序为0—7,当该8个条目占满之后,如果编号0位置被释放,则可以将新的访问信息插入该条目,并继续顺序编号为9,编号9之后的下一个条目的编号为10,依此类推。

[0070] 示例性地,网卡114将该客户端10的访问信息入列的过程可以包括:网卡114根据该共享资源对应的锁管理信息中的队尾指针P\_cnt,找到内存113的环形缓冲区中该共享资源对应的缓存队列的队尾,将该客户端10的访问信息添加至队尾的下一个空白条目中。若该客户端10可以获得锁,该客户端10的访问信息中节点状态信息为acquired,锁状态信息为active,其他参数可以根据加锁请求生成,或为缺省值,此处不再赘述。若该客户端10不能获得锁,则该客户端10的访问信息中节点状态信息为not acquire,其他参数可以为缺省值。网卡114还用于维护锁管理信息,若该客户端10请求获得写锁,则将Wlock\_cnt计数+1,若该客户端请求获得读锁,该将Rlock\_cnt\_计数+1,同时根据更新后的缓存队列更新队尾指针P\_cnt,更新后的队尾指针指向该客户端的访问信息所在的条目。

[0071] 需要说明的是,步骤402b为可选的步骤,并非必须执行的步骤。另外,步骤402b可以在步骤402a之前执行,或在步骤402a之后执行,也可以步骤402a与步骤402b同时执行,本申请实施例对此不做限定。如果在步骤402a之前执行,由于先将客户端10的访问信息入列,因此,在基于锁管理信息判断该客户端10是否可以获得锁时,如果客户端10请求获得写锁,则Wlock\_cnt=1时指示写锁为空闲状态,Rlock\_cnt=0指示读锁为空闲状态;如果客户端10请求获得读锁,则Wlock\_cnt=0时指示写锁空闲,Rlock\_cnt=1指示读锁空闲。上述设计,当网卡接收到多个访问节点的加锁请求时,可以使用缓存队列来管理这些访问节点的加锁请求,如网卡114可以将接收到加锁请求顺序添加至缓存队列,以使网卡114可以基于缓存队列顺序为请求锁资源的访问节点加锁,这样客户端便不需要与服务端20多次交互,避免客户端盲目竞争锁资源,可以减少客户端与网卡114之间的交互次数,节省网络资源开销。以顺序为这些客户端加锁。

[0072] 步骤403,网卡114向该客户端10发送加锁成功响应,该加锁成功响应用于指示该客户端10成功获得该共享资源的锁。

[0073] 网卡114向该客户端10发送加锁成功响应,通知客户端10获得该共享资源的写锁或读锁。对应的,该客户端10接收到该加锁成功响应,便可以对共享资源执行写操作或读操作。

[0074] 在一种可选的方式中,若执行了步骤402b,则在步骤403中,加锁成功响应中还可以包括用于指示该客户端10的访问信息在缓存队列中位置的位置标识。该位置标识可以在客户端10发送解锁请求时一起返回,以方便网卡114快速在缓存队列中定位到该客户端10所在的条目,加快解锁速度。

[0075] 上述方式,网卡114接收到客户端10的加锁请求之后,可以基于锁管理信息快速判断该客户端10是否能够获得锁,即由网卡114确定客户端10是否可以获得共享资源的锁,不需要CPU112参与,降低了对CPU112的占用率,有利于提高CPU112的资源利用率。更进一步地,由于该锁管理信息可以存储于网卡114的存储器1142中,因此,锁管理模块1141不需要跨总线访问该锁管理信息,可以提高获取该锁管理信息的速度,降低响应时延,提高系统性能。

[0076] 下面对解锁过程进行介绍:

[0077] 步骤404,网卡114接收客户端10(如客户端10-a)发送的解锁请求。

[0078] 该解锁请求用于请求释放共享资源的锁。具体的,该解锁请求携带的信息包括但不限于下列中的一项或多项:该客户端10所请求释放的锁的类型(如写锁或读锁)、该客户端10的节点标识(用于唯一标识一个访问节点)、锁的地址信息(如该锁的缓存队列在内存的地址)、用于指示该客户端的访问信息在缓存队列中位置的位置标识(从加锁响应中获取并记录下来的)。

[0079] 步骤405,网卡114为该客户端10解锁。

[0080] 具体的,网卡114根据解锁请求中携带的位置标识,将缓存队列中对应位置的条目中的锁状态信息修改为inactive,以标记该客户端已释放锁。除此之外,网卡114还可以向客户端10发送解锁响应(图4未示出)。

[0081] 另外,网卡114根据该解锁请求更新锁管理信息,如该解锁请求用于请求释放写锁,则将Wlock\_cnt计数-1,若该客户端的访问信息排列在缓存队列的队首,则还可以更新

头部指针(C\_cnt),令更新后的C\_cnt指向缓存队列的下一个条目。同理,若该解锁请求用于请求释放读锁,则将Rlock\_cnt计数-1,由于读锁可以共享,且每个读者执行读操作所需时间不同,因此,可以在所有获得读锁的读者均释放读锁之后再更新C\_cnt、P\_cnt。

[0082] 在一种可选的实施方式中,可以检测客户端10的状态,如通过心跳检测客户端10是否正常,若检测到获得锁的客户端10发生故障,则可以通知网卡114为发生故障的客户端10解锁,以及更新锁管理信息,具体实现方式参见上文的描述,此处不再赘述。基于该设计,可以避免发生故障的客户端10长期占有锁,以此提高系统可靠性和性能。值得注意的是,监测客户端10状态这一操作可以贯穿整个方法实施过程,并非限定于步骤405执行,本申请实施例对此不做限定。另外,上述检测客户端10心跳的执行主体可以是处理器112或网卡114或其他节点,如其他服务端设备(该其他服务端设备可以与服务端20通信)等等,本申请实施例对此不做限定。

[0083] 步骤406,网卡114判断该共享资源的写锁以及读锁是否均空闲,如果是,则执行步骤407。

[0084] 网卡114在为客户端10解锁之后,首先判断该共享资源的写锁以及读锁是否均为空闲状态,如果是,则可以为其余未获得锁的访问节点加锁,如下通过两种情况来介绍如何判断共享资源的写锁以及读锁是否均为空闲状态:

[0085] 情况一:解锁请求为请求释放写锁,应理解的是,由于写锁具有排他性,一次只能一个访问节点获得,因此,可以确定,网卡114在释放该客户端10的写锁之后,该共享资源的写锁和读锁均为空闲状态。

[0086] 情况二:解锁请求为请求释放读锁,由于读锁可以共享,可能有多个读者均获得了读锁,因此,在一种实施方式中,网卡114根据缓存队列的头部指针(C\_cnt)和缓存队列的队尾指针(P\_cnt),从缓存队列的头部开始,顺序读取部分或全部访问信息(至尾部结束)。若基于获得的访问信息确定,全部获得锁的读者均已释放锁(即节点状态信息为acquire的访问信息中锁状态信息均为inactive),则确定该共享资源的写锁和读锁均为空闲状态。如果有一个或多个节点状态信息为acquire的访问信息中锁状态信息为active,即有个读者获得读锁,则网卡114可能重复上述步骤404至步骤406的过程,直至所有获得读锁的读者均释放锁。在另一种实施方式中,网卡114还可以根据锁管理信息确定写锁以及读锁是否均空闲,参见上文的相关介绍,此处不再赘述。

[0087] 步骤407,网卡根据该共享资源的缓存队列,确定可以获得锁的访问节点。

[0088] 网卡114根据缓存队列的头部指针(C\_cnt)和缓存队列的队尾指针(P\_cnt),从缓存队列的头部开始,顺序读取部分或全部访问信息。

[0089] 若该缓存队列中排列的首个未获得锁的访问节点请求获得写锁,则确定该访问节点可以获得写锁。值得注意的是,这种情况下,网卡114只需要读取缓存队列的首个访问信息,不需要全部读取,以节省网络资源开销。

[0090] 若该缓存队列中排列的首个未获得锁的访问节点请求获得读锁,由于读锁可以共享,因此,在一种实施方式中,该缓存队列中未获得锁的全部读者均可以获得读锁,比如,假设该缓存队列中首个未获得锁的访问节点的位置标识为9,位置标识12、位置标识13的访问信息均指示为未获得锁的读者,则这几个读者均可以获得读锁,这种方式可以提高读性能。

[0091] 为了避免写者长时间无法获得锁,在另一种实施方式中,该缓存队列中,以首个未

获得锁的访问节点为起点的连续多个读者才可以获得读锁,比如,假设该缓存队列中首个未获得锁的访问节点的位置标识为9,若之后的位置标识10和位置标识11的访问信息均为未获得锁的读者,位置标识12的访问信息为未获得锁的写者,则位置标识9~位置标识11该3个读者均可以获得读锁。这种方式,可以在保证顺序加锁的基础上,提高读性能,减少对写者的影响。

[0092] 步骤408,针对确定的任一可以获得锁的访问节点,网卡114向该访问节点发送加锁响应,用于指示该访问节点成功获得锁。

[0093] 可以理解的是,在图1所示的系统中,这里的访问节点是指客户端。

[0094] 上述方式,网卡114接收到客户端10的解锁请求后,由网卡114来释放该锁资源,之后,网卡从缓存队列读取访问信息,以确定可以获得共享资源的锁的访问节点,并且网卡114可以顺序为缓存队列中排队的访问节点加锁,减少了无序加锁的情况,避免请求共享资源的锁的访问节点被饿死。另外,由网卡114确定访问节点是否可以获得锁,不需要CPU参与,降低了对CPU的占用率,有利于提高CPU的资源利用率。

[0095] 基于与方法实施例相同的构思,本申请实施例还提供了一种锁管理装置,该锁管理装置用于执行上述图4所示的方法实施例中网卡114所执行的方法。如图5所示,该锁管理装置500包括获取模块501、处理模块502,发送模块503,具体的,在锁管理装置500中,各模块直接通过通信通路建立连接。

[0096] 获取模块501,用于获取客户端的加锁请求,该加锁请求用于请求获得共享资源的访问权限;具体实现方式请参见图4中的步骤401的描述,此处不再赘述。

[0097] 处理模块502,用于根据记录的锁信息判断该客户端是否能够获得该共享资源的锁;具体实现方式请参见图4中的步骤402的描述,或参见步骤406的描述,此处不再赘述。在确定该客户端能够获得该共享资源的锁的情况下,通过发送模块503向该客户端发送加锁成功响应,该加锁成功响应用于指示客户端成功获得共享资源的锁。具体实现方式请参见图4中的步骤403的描述,此处不再赘述。

[0098] 作为一种可能的实施方式,获取模块501获取客户端的加锁请求时,具体用于接收该客户端的加锁请求;或从缓存队列中获取客户端的加锁请求。

[0099] 作为一种可能的实施方式,锁信息包括请求获得共享资源的写锁的第一数量以及请求获得共享资源的读锁的第二数量;处理模块502在根据锁信息判断客户端是否能够获得共享资源的锁时,具体用于:当第一数量指示共享资源的写锁为空闲状态时,确定客户端能够获得共享资源的读锁;或当第一数量指示共享资源的写锁为空闲状态,且第二数量指示共享资源的读锁为空闲状态时,确定客户端能够获得共享资源的写锁。

[0100] 作为一种可能的实施方式,锁信息还包括锁状态信息、节点状态信息;其中,锁状态信息用于指示访问节点所申请的锁是否为解锁状态;节点状态信息用于指示访问节点是否已获得锁;处理模块502在根据锁信息判断客户端是否能够获得共享资源的锁时,具体用于:根据节点状态信息、锁状态信息,确定已获得共享资源的锁的访问节点均为解锁状态,则确定客户端能够获得共享资源的锁。

[0101] 作为一种可能的实施方式,加锁响应包括锁位置信息,锁位置信息用于指示客户端的锁信息在缓存队列中的位置;获取模块501,还用于接收客户端发送的解锁请求,解锁请求用于请求释放共享资源的锁,解锁请求包括锁位置信息;具体实现方式请参见图4中的

步骤404的描述,此处不再赘述。处理模块502,还用于根据锁位置信息,将缓存队列中的锁位置信息指示的锁的状态修改为解锁状态。具体实现方式请参见图4中的步骤405的描述,此处不再赘述。

[0102] 作为一种可能的实施方式,处理模块502,还用于确定获得锁的客户端发生故障,则释放客户端的锁。

[0103] 在一个简单的实施例中,本领域的技术人员可以想到如图4所示的实施例中网卡114可采用图6所示的形式。

[0104] 如图6所示的锁管理装置600,包括至少一个处理器601、存储器602,可选的,还可以包括通信接口603。

[0105] 存储器602可以是易失性存储器,例如随机存取存储器;存储器也可以是非易失性存储器,例如只读存储器,快闪存储器,HDD或SSD、或者存储器602是能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由计算机存取的任何其他介质,但不限于此。存储器602可以是上述存储器的组合。

[0106] 本申请实施例中不限定上述处理器601以及存储器602之间的具体连接介质。

[0107] 处理器601可以为中央处理器,该处理器601还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现场可编程门阵列(Field Programmable Gate Array,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件、人工智能芯片、片上芯片等。通用处理器可以是微处理器或者是任何常规的处理器等。处理器601在与其他设备进行通信时,可以通过通信接口603进行数据传输,如接收加锁请求或解锁请求。

[0108] 当所述装置采用图6所示的形式时,图6中的处理器601可以通过调用存储器602中存储的计算机执行指令,使得所述检测装置可以执行上述任一方法实施例中的网卡114执行的方法。

[0109] 具体的,图5的获取模块501、处理模块502以及发送模块503的功能/实现过程均可以通过图6中的处理器601调用存储器602中存储的计算机执行指令来实现。

[0110] 本申请实施例还提供一种计算机存储介质,该计算机存储介质中存储有计算机指令,当该计算机指令在锁管理装置上运行时,使得锁管理装置执行上述相关方法步骤以实现上述实施例中的网卡114所执行的方法,参见图4各步骤的描述,此处不再赘述,此处不再赘述。

[0111] 本申请实施例还提供了一种计算机程序产品,当该计算机程序产品在计算机上运行时,使得计算机执行上述相关步骤,以实现上述实施例中的网卡114所执行的方法,参见图4各步骤的描述,此处不再赘述,此处不再赘述。

[0112] 另外,本申请的实施例还提供一种装置,这个装置具体可以是芯片,组件或模块,该装置可包括相连的处理器和存储器;其中,存储器用于存储计算机执行指令,当装置运行时,处理器可执行存储器存储的计算机执行指令,以使芯片执行上述各方法实施例中的网卡114所执行的方法,参见图4各步骤的描述,此处不再赘述,此处不再赘述。

[0113] 其中,本申请实施例提供的存储设备、计算机存储介质、计算机程序产品或芯片均用于执行上文所提供的网卡114对应的方法,因此,其所能达到的有益效果可参考上文所提供的对应的方法中的有益效果,此处不再赘述。

[0114] 通过以上实施方式的描述,所属领域的技术人员可以了解到,为描述的方便和简洁,仅以上述各功能模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能模块完成,即将装置的内部结构划分成不同的功能模块,以完成以上描述的全部或者部分功能。

[0115] 在本申请所提供的几个实施例中,应该理解到,所揭露的装置和方法,可以通过其他的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,模块或单元的划分,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装置,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的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其他的形式。

[0116] 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一个物理单元或多个物理单元,即可以位于一个地方,或者也可以分布到多个不同地方。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。

[0117] 另外,在本申请各个实施例中的各功能单元(或模块)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。

[0118] 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,可以存储在一个可读取存储介质中。基于这样的理解,本申请实施例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,该软件产品存储在一个存储介质中,包括若干指令用以使得一个设备(可以是单片机,芯片等)或处理器(processor)执行本申请各个实施例方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(read only memory,ROM)、随机存取存储器(random access memory,RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0119] 可选的,本申请实施例中的计算机执行指令也可以称之为应用程序代码,本申请实施例对此不作具体限定。

[0120] 在上述实施例中,可以全部或部分地通过软件、硬件、固件或者其任意组合来实现。当使用软件实现时,可以全部或部分地以计算机程序产品的形式实现。所述计算机程序产品包括一个或多个计算机指令。在计算机上加载和执行所述计算机程序指令时,全部或部分地产生按照本申请实施例所述的流程或功能。所述计算机可以是通用计算机、专用计算机、计算机网络、或者其他可编程装置。所述计算机指令可以存储在计算机可读存储介质中,或者从一个计算机可读存储介质向另一个计算机可读存储介质传输,例如,所述计算机指令可以从一个网站站点、计算机、服务器或数据中心通过有线(例如同轴电缆、光纤、数字用户线(DSL))或无线(例如红外、无线、微波等)方式向另一个网站站点、计算机、服务器或数据中心进行传输。所述计算机可读存储介质可以是计算机能够存取的任何可用介质或者是包括一个或多个可用介质集成的服务器、数据中心等数据存储设备。所述可用介质可以是磁性介质,(例如,软盘、硬盘、磁带)、光介质(例如,DVD)、或者半导体介质(例如固态硬盘(Solid State Disk,SSD))等。

[0121] 本申请实施例中所描述的各种说明性的逻辑单元和电路可以通过通用处理器,数

字信号处理器,专用集成电路(ASIC),现场可编程门阵列(FPGA)或其它可编程逻辑装置,离散门或晶体管逻辑,离散硬件部件,或上述任何组合的设计来实现或操作所描述的功能。通用处理器可以为微处理器,可选地,该通用处理器也可以为任何传统的处理器、控制器、微控制器或状态机。处理器也可以通过计算装置的组合来实现,例如数字信号处理器和微处理器,多个微处理器,一个或多个微处理器联合一个数字信号处理器核,或任何其它类似的配置来实现。

[0122] 本申请实施例中所描述的方法或算法的步骤可以直接嵌入硬件、处理器执行的软件单元、或者这两者的结合。软件单元可以存储于RAM存储器、闪存、ROM存储器、EPROM存储器、EEPROM存储器、寄存器、硬盘、可移动磁盘、CD-ROM或本领域中其它任意形式的存储媒介中。示例性地,存储媒介可以与处理器连接,以使得处理器可以从存储媒介中读取信息,并向存储媒介存写信息。可选地,存储媒介还可以集成到处理器中。处理器和存储媒介可以设置于ASIC中。

[0123] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0124] 尽管结合具体特征及其实施例对本申请进行了描述,显而易见的,在不脱离本申请的精神和范围的情况下,可对其进行各种修改和组合。相应地,本说明书和附图仅仅是所附权利要求所界定的本申请的示例性说明,且视为已覆盖本申请范围内的任意和所有修改、变化、组合或等同物。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,则本申请也意图包括这些改动和变型在内。

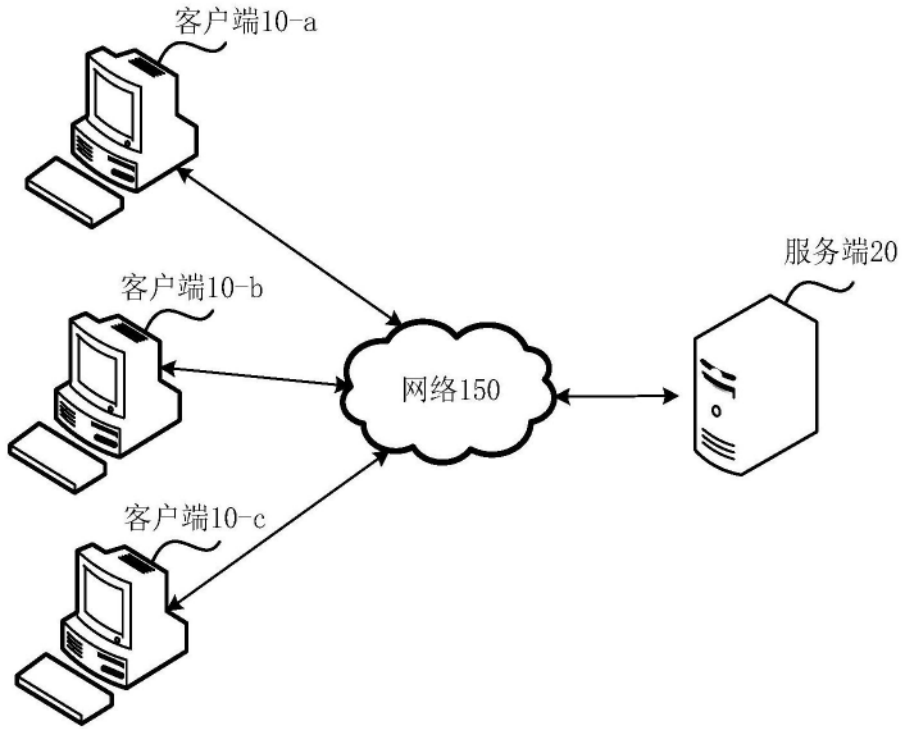


图1

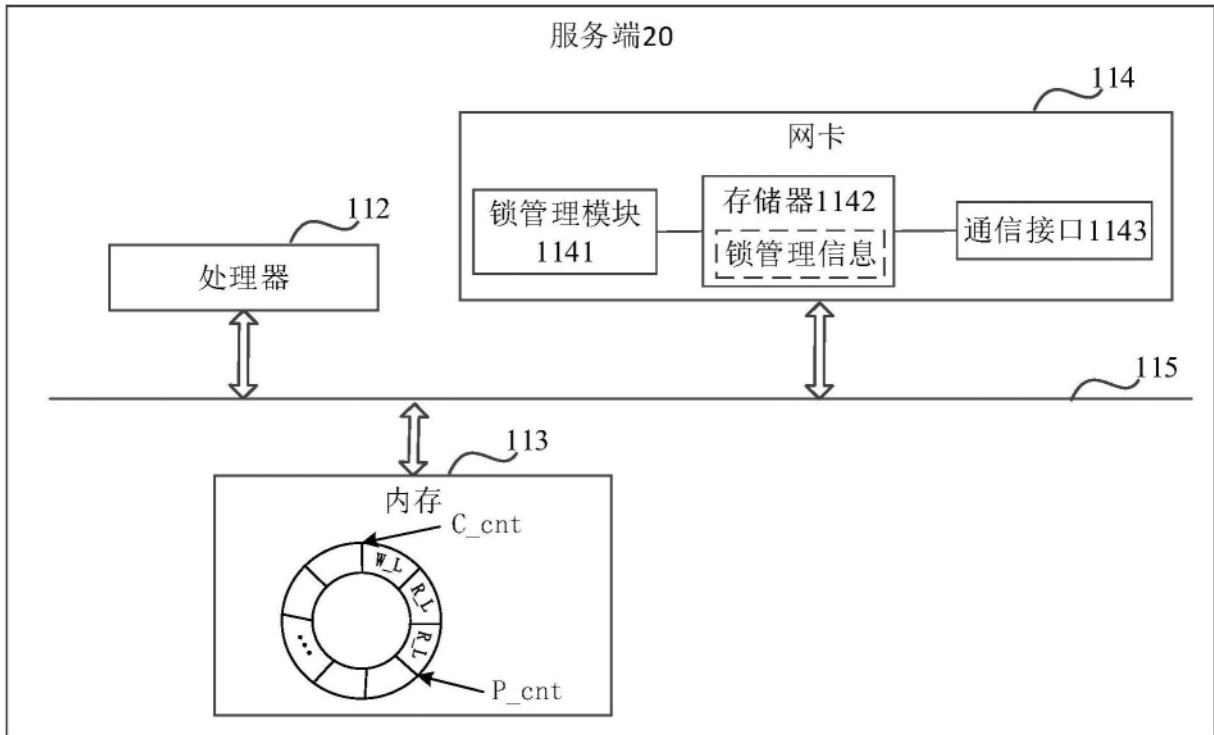


图2

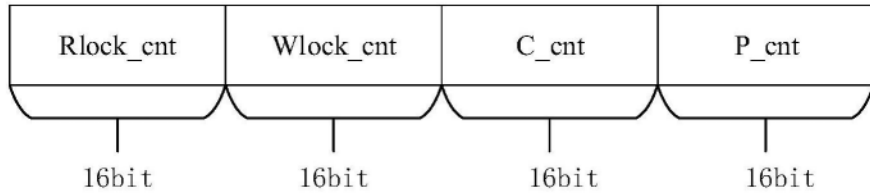


图3

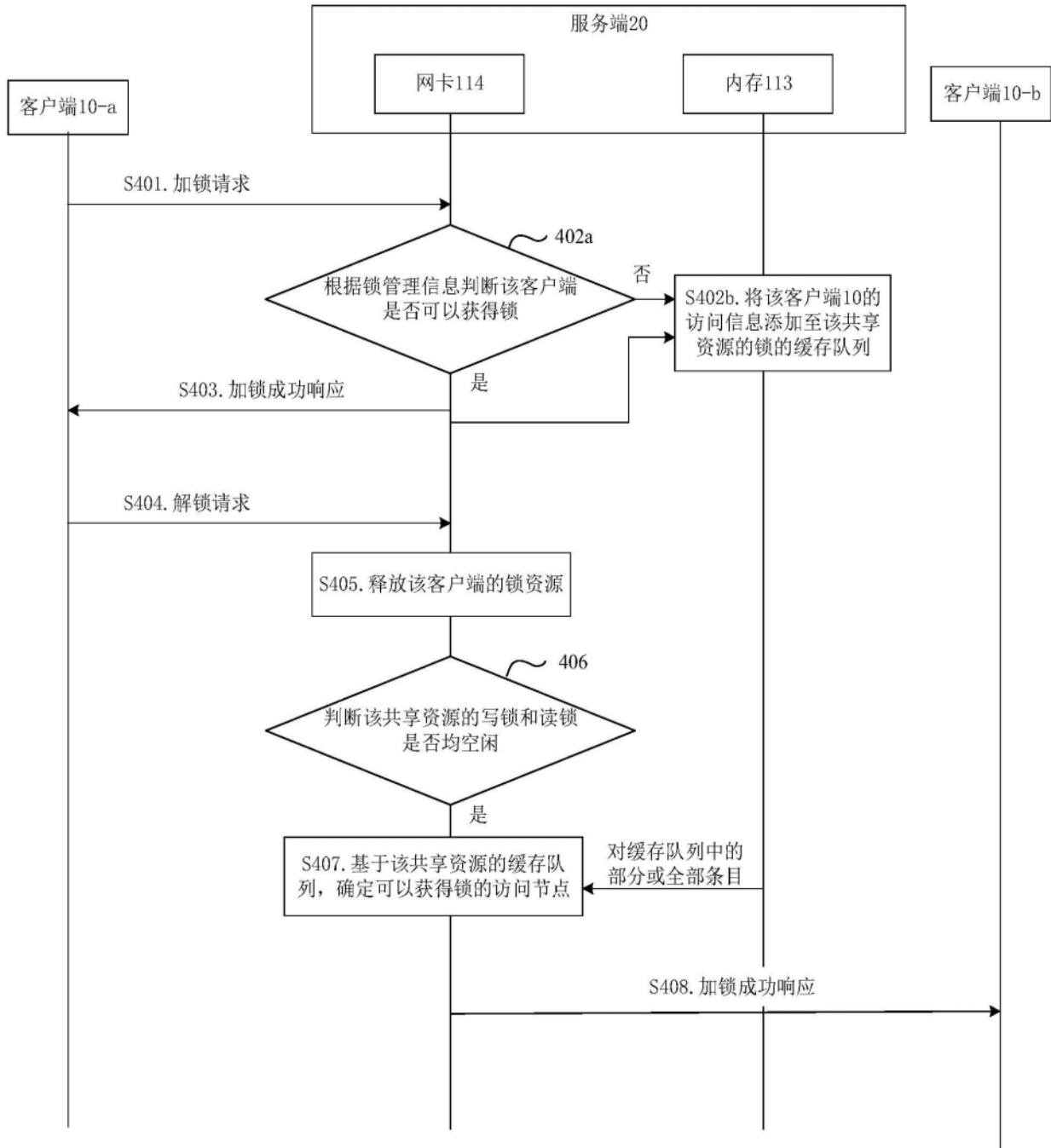


图4

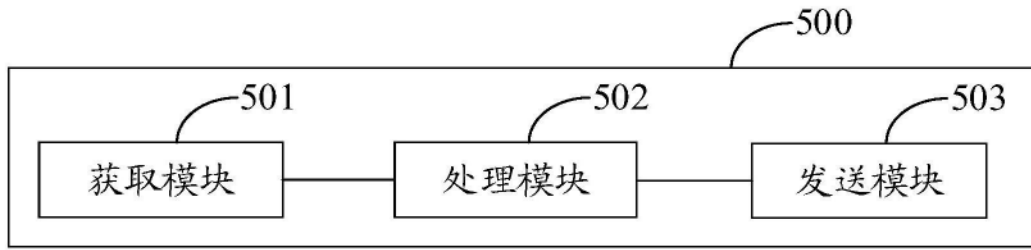


图5

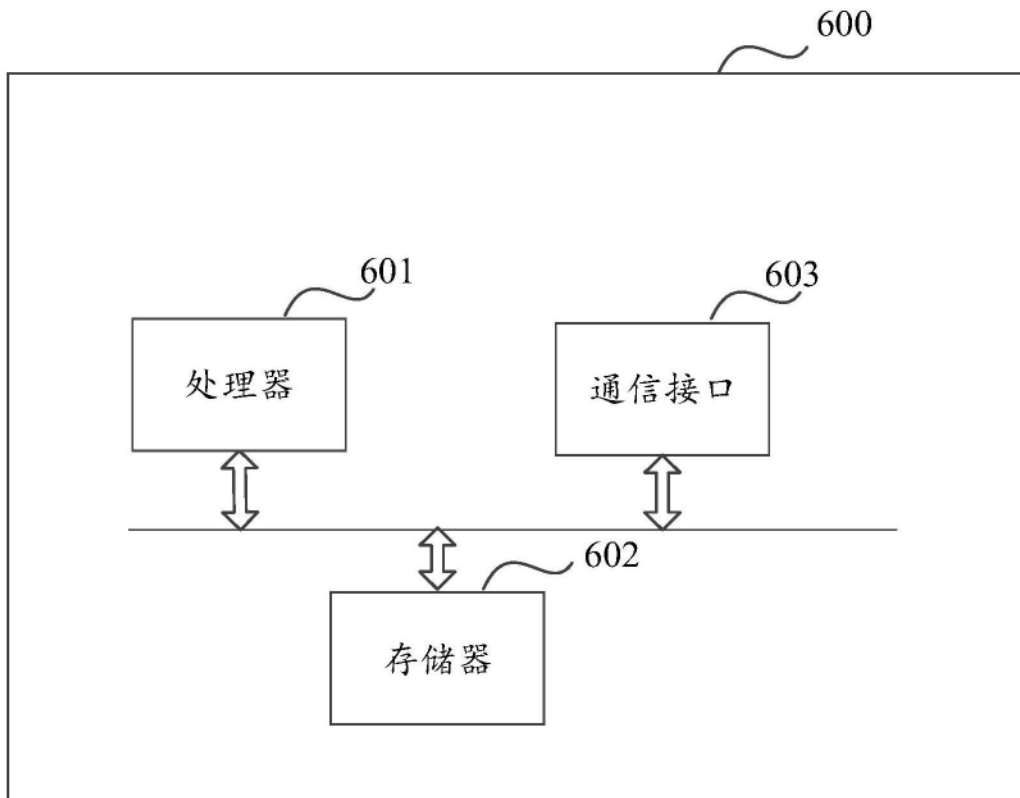


图6