



(12)发明专利

(10)授权公告号 CN 106020822 B

(45)授权公告日 2019.03.26

(21)申请号 201610332883.5

(22)申请日 2016.05.18

(65)同一申请的已公布的文献号
申请公布号 CN 106020822 A

(43)申请公布日 2016.10.12

(73)专利权人 腾讯科技(成都)有限公司
地址 610041 四川省成都市成都高新区拓
东东街81号天府软件园二期C区4号楼

(72)发明人 邓君

(74)专利代理机构 广州华进联合专利商标代理
有限公司 44224
代理人 何平 邓云鹏

(51)Int.Cl.
G06F 8/20(2018.01)
G06F 8/30(2018.01)

(56)对比文件

CN 101727355 A,2010.06.09,
US 2007113014 A1,2007.05.17,
US 5398334 A,1995.03.14,
CN 101963914 A,2011.02.02,
CN 101566945 A,2009.10.28,
CN 101593131 A,2009.12.02,
US 2014040326 A1,2014.02.06,
US 6047295 A,2000.04.04,

审查员 张忠钊

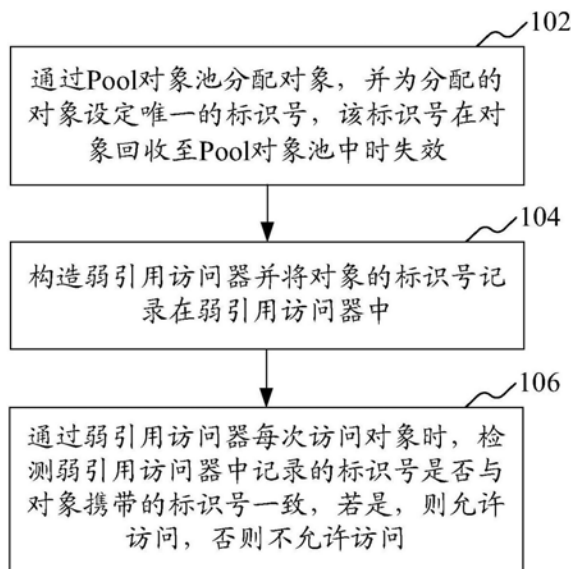
权利要求书1页 说明书6页 附图5页

(54)发明名称

面向Pool对象的弱引用实现方法和装置

(57)摘要

本发明涉及一种面向Pool对象的弱引用实现方法,所述方法包括:通过Pool对象池分配对象,并为分配的所述对象设定唯一的标识号,所述标识号在所述对象回收至所述Pool对象池中时失效;构造弱引用访问器并将所述对象的标识号记录在所述弱引用访问器中;通过所述弱引用访问器每次访问所述对象时,检测所述弱引用访问器中记录的标识号是否与所述对象携带的标识号一致,若是,则允许访问,否则不允许访问。采用该方法,能在Pool对象上实现弱引用机制。此外,还提供了一种面向Pool对象的弱引用实现装置。



1. 一种面向Pool对象的弱引用实现方法,所述方法包括:

通过Pool对象池分配对象,并为分配的所述对象设定唯一的标识号,所述标识号在所述对象回收至所述Pool对象池中时失效;

构造弱引用访问器,将所述对象的指针保存在所述弱引用访问器中,以及将所述对象的标识号保存在所述弱引用访问器的标识号变量中;

通过所述弱引用访问器每次访问所述对象时,检测所述弱引用访问器中记录的标识号是否与所述对象携带的标识号一致,若是,则允许访问,否则不允许访问。

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:

当回收所述对象时设置所述对象携带的标识号为无效值,将所述对象归还至所述Pool对象池中。

3. 根据权利要求2所述的方法,其特征在于,所述检测弱引用访问器中记录的标识号是否与所述对象携带的标识号一致,若是,则允许访问,否则不允许访问,包括:

检测被访问的所述对象的指针是否不为空且所述弱引用访问器中记录的标识号是否与所述对象携带的标识号一致,若是,则允许访问,否则不允许访问。

4. 根据权利要求1至3任一项所述的方法,其特征在于,所述标识号为全局唯一的序列号。

5. 一种面向Pool对象的弱引用实现装置,其特征在于,所述装置包括:

对象分配模块,用于通过Pool对象池分配对象,并为分配的所述对象设定唯一的标识号,所述标识号在所述对象回收至所述Pool对象池中时失效;

访问器构造模块,用于构造弱引用访问器并将所述对象的标识号记录在所弱引用访问器中;

对象访问模块,用于通过所述弱引用访问器每次访问所述对象时,检测所述弱引用访问器中记录的标识号是否与所述对象携带的标识号一致,若是,则允许访问,否则不允许访问。

6. 根据权利要求5所述的装置,其特征在于,所述装置还包括:

对象回收模块,用于当回收所述对象时设置所述对象携带的标识号为无效值,将所述对象归还至所述Pool对象池中。

7. 根据权利要求5所述的装置,其特征在于,所述对象访问模块用于检测被访问的所述对象的指针是否不为空且所述弱引用访问器中记录的标识号是否与所述对象携带的标识号一致,若是,则允许访问,否则不允许访问。

8. 根据权利要求5至7任意一项所述的装置,其特征在于,所述标识号为全局唯一的序列号。

9. 一种服务器,包括存储器和处理器,所述存储器存储有计算机程序,所述计算机程序被所述处理器执行时,使得所述处理器执行如权利要求1至4中任一项所述方法的步骤。

10. 一种存储介质,存储有计算机程序,所述计算机程序被处理器执行时,使得所述处理器执行如权利要求1至4中任一项所述方法的步骤。

面向Pool对象的弱引用实现方法和装置

技术领域

[0001] 本发明涉及计算机技术领域,特别是涉及一种面向Pool对象的弱引用实现方法和装置。

背景技术

[0002] 弱引用是主要用于程序某个模块拥有程序对象的所有权(生命周期控制权)但其他模块需要访问该对象时,用于隔离和检查所持有的引用(指针)是否有效的一种方法。

[0003] 传统技术中,程序对象的生命期通过引用计数控制,强引用访问器控制着计数的加减,弱引用访问器不能修改对象的引用计数。当计数不为0时对象有效,对该对象的弱引用访问器有效,当计数为0时对象生命期结束会被系统释放,弱引用访问器失效。每个对象的弱引用访问器维护在每个对象绑定的弱引用列表中,当对象被真正释放时修改该列表中的访问器状态从而使其失效。

[0004] 然而,传统的这种弱引用实现方法并不能适用于Pool对象,Pool对象是指被分配在内存池中可以被反复申请使用和归还的程序对象,对于加快内存访问和减少内存碎片很有用处。对于Pool对象,对象受对象池管理,对象的引用计数最小是1,因此对象不会被释放,也就无法让弱引用访问器失效。

发明内容

[0005] 基于此,有必要针对上述技术问题,提供一种能在Pool对象上实现弱引用机制的面向Pool对象的弱引用实现方法和装置。

[0006] 一种面向Pool对象的弱引用实现方法,所述方法包括:

[0007] 通过Pool对象池分配对象,并为分配的所述对象设定唯一的标识号,所述标识号在所述对象回收至所述Pool对象池中时失效;

[0008] 构造弱引用访问器并将所述对象的标识号记录在所述弱引用访问器中;

[0009] 通过所述弱引用访问器每次访问所述对象时,检测所述弱引用访问器中记录的标识号是否与所述对象携带的标识号一致,若是,则允许访问,否则不允许访问。

[0010] 一种面向Pool对象的弱引用实现装置,所述装置包括:

[0011] 对象分配模块,用于通过Pool对象池分配对象,并为分配的所述对象设定唯一的标识号,所述标识号在所述对象回收至所述Pool对象池中时失效;

[0012] 访问器构造模块,用于构造弱引用访问器并将所述对象的标识号记录在所弱引用访问器中;

[0013] 对象访问模块,用于通过所述弱引用访问器每次访问所述对象时,检测所述弱引用访问器中记录的标识号是否与所述对象携带的标识号一致,若是,则允许访问,否则不允许访问。

[0014] 上述面向Pool对象的弱引用实现方法和装置,通过Pool对象池分配对象,在分配对象时为分配的对象设定唯一的标识号,构造弱引用访问器时将对象的标识号记录在弱引

用访问器中,这样,弱引用访问器每次访问对象时,检测弱引用访问器中记录的标识号是否与对象携带的标识号一致,若是,则允许访问,否则不允许访问。由于对象分配时为对象设定唯一的标识号,且该标识号在对象回收至Pool对象池中时失效,从而实现对象回收是弱引用访问器失效的目的。相对于传统的引用计数方式,该方法和装置可以适用于Pool对象,实现Pool对象的弱引用机制。

附图说明

- [0015] 图1为一个实施例中面向Pool对象的弱引用实现方法的流程图;
- [0016] 图2为另一个实施例中面向Pool对象的弱引用实现方法的流程图;
- [0017] 图3为一个实施例中面向Pool对象的弱引用实现方法的实现框架图;
- [0018] 图4为一个实施例中面向Pool对象的弱引用实现方法的时序图;
- [0019] 图5为一个实施例中面向Pool对象的弱引用实现装置的结构框图;
- [0020] 图6为另一个实施例中面向Pool对象的弱引用实现装置的结构框图;
- [0021] 图7为一个实施例中服务器的内部结构图。

具体实施方式

[0022] 为了使本发明的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本发明进行进一步详细说明。应当理解,此处所描述的具体实施例仅仅用以解释本发明,并不用于限定本发明。

[0023] 如图1所示,在一个实施例中,提供了一种面向Pool对象的弱引用实现方法,该方法适用于Pool对象,使Pool对象能够实现弱引用机制。该方法包括:

[0024] 步骤102,通过Pool对象池分配对象,并为分配的对象设定唯一的标识号,该标识号在对象回收至Pool对象池中时失效。

[0025] 对象是指计算机中对事物概念、数据等的抽象表达,是一种资源,需分配一定的内存资源来进行记录。Pool对象是指被分配在内存池中可以被反复申请使用和归还的程序对象。Pool对象池是Pool对象的管理器,Pool对象池可维护一个空闲对象的列表,当需要分配对象时为对象拥有者分配一个对象。在分配对象时为分配的对象设定唯一的标识号,该标识号在对象的生命续存期间有效,在对象回收至Pool对象池中时失效,对象被访问时携带对应的标识号。

[0026] 步骤104,构造弱引用访问器并将对象的标识号记录在弱引用访问器中。

[0027] 访问器是对程序对象的访问承载者,可以是指针、智能指针、强引用和弱引用等。弱引用是指程序某个模块拥有程序对象的所有权但其他模块需要访问该对象时,用于隔离和检查所持有的引用或指针是否有效的一种方法。弱引用访问器是对弱引用方法的封装,可以提供与被访问对象间的有效性判断和访问接口。在构造弱引用访问器时需要将分配的对象传递给弱引用访问器,具体的可将对象的指针和携带的标识号传递给弱引用访问器,弱引用访问器获取该对象的标识号,将该对象的标识号记录在弱引用访问器中。

[0028] 步骤106,通过弱引用访问器每次访问对象时,检测弱引用访问器中记录的标识号是否与对象携带的标识号一致,若是,则允许访问,否则不允许访问。

[0029] 本实施例中,在对象被释放前是对象的生命续存期间。由于弱引用访问器提供了

与被访问的对象的有效性判断和访问接口,在对象的生命续存期间,弱引用访问器每次访问对象时,都会对标识号进行有效性判断。具体的,检测若引用访问器中记录的标识号是否与对象携带的标识号一致,若是,则允许访问,否则不允许访问,弱引用访问器失效。

[0030] 本实施例中,通过Pool对象池分配对象,在分配对象时为分配的对象设定唯一的标识号,构造弱引用访问器时将对象的标识号记录在弱引用访问器中,这样,弱引用访问器每次访问对象时,检测弱引用访问器中记录的标识号是否与对象携带的标识号一致,若是,则允许访问,否则不允许访问。由于对象分配时为对象设定唯一的标识号,且该标识号在对象回收至Pool对象池中时失效,从而实现对象回收是弱引用访问器失效的目的。相对于传统的引用计数方式,该方法和装置可以适用于Pool对象,实现Pool对象的弱引用机制。

[0031] 在一个实施例中,上述面向Pool对象的弱引用实现方法还包括:当回收对象时设置对象携带的标识号为无效值,将对象归还至Pool对象池中。当对象拥有者需要回收对象时可调用对象的相应接口,该接口将对象携带的标识号设置成无效值。具体的,可以更改对象所携带的标识号,这样,弱引用访问器记录的标识号与对象携带的标识号则变为不同,弱引用访问器不能访问对象。

[0032] 在一个实施例中,构造弱引用访问器并将对象的标识号记录在弱引用访问器中,包括:构造弱引用访问器,将对象的指针保存在弱引用访问器中,以及将对象的标识号保存在弱引用访问器的标识号变量中。这样,便可以将对象的标识号记录在弱引用访问器中。

[0033] 进一步的,在一个实施例中,检测弱引用访问器中记录的标识号是否与对象携带的标识号一致,若是,则允许访问,否则不允许访问,包括:检测被访问的对象的指针是否不为空且弱引用访问器中记录的标识号是否与对象携带的标识号一致,若是,则允许访问,否则不允许访问。

[0034] 本实施例中,被访问对象的指针不为空,则表明对象实际存在,进一步判断弱引用访问器中记录的标识号是否与对象携带的标识号一致,如果一致,则表明弱引用访问器未失效,可以访问该对象,如果不一致,则表明弱引用访问已失效,无法访问该对象。

[0035] 在一个实施例中,分配对象时为对象设定的标识号是全局唯一的序列号。

[0036] 如图2所示,在另一个实施例中,提供了一种面向Pool对象的弱引用实现方法,该方法包括:

[0037] 步骤202,通过Pool对象池分配对象,并为分配的对象设定全局唯一的序列号。

[0038] Pool对象池是Pool对象的管理器,维护着一个空闲对象列表,当外部需要分配对象时调用Pool对象池的分配接口,该接口从空闲对象列表的队尾取出一个对象,如果Pool对象池中沒有空闲对象则再向系统申请。Pool对象池在取出一个对象时为对象赋予一个全局唯一的序列号。

[0039] 图3示出了面向Pool对象的弱引用实现方法的实现框架,该框架主要包括泛型对象池模块(即PoolObjectControl<T>)、支持Pool弱引用框架的基类(PoolObjectBase)以及弱引用访问器(PoolObjectHandle<T>)。其中,泛型对象池模块负责给定类型的对象进行分配和回收,对象使用方可通过Get接口获取一个对象,当对象池中沒有空闲对象时则可以从系统上重新分配一个对象给外部使用。泛型对象池模块再返回对象之前设定一个唯一的序列号,并将该序列号赋予返回的对象的成员变量,调用其初始化虚函数OnUse完成对象初始化。

[0040] 步骤204,构造弱引用访问器,将对象的指针保存在弱引用访问器中,以及将对象的序列号保存在弱引用访问器的标识号变量中。

[0041] 弱引用访问器是对弱引用方法的封装,提供了与别访问对象间的有效性判断和访问接口。结合图3所示,弱引用访问器(PoolObjectHandle<T>)构造函数完成RefObj和SeqNum的赋值操作:

[0042] `this.RefObj=obj;this.SeqNum=obj.SeqNum。`

[0043] 其中,`this.RefObj=obj`表示将对象的指针保存在弱引用访问器中,`this.SeqNum=obj.SeqNum`表示将对象的唯一序列号记录下来保存在弱引用访问器的SeqNum变量中。

[0044] 步骤206,通过弱引用访问器每次访问对象时,检测弱引用访问器中记录的序列号是否与对象携带的序列号一致,若是,则允许访问,否则不允许访问。

[0045] 弱引用访问器在每次访问对象时,需要进行有效性判断。结合图3所示,operator bool()实现弱引用有效性判断,operator T()实现对引用对象的透明访问。其中,有效性判断中有效的条件为:`this.RefObj`不为null且`this.SeqNum`等于`RefObj.SeqNum`。也就是判断被访问的对象指针不为空且对象携带的序列号等于弱引用访问器中记录的序列号,则弱引用访问器有效,允许访问。

[0046] 步骤208,当回收对象时,设置对象携带的序列号为无效值,将对象归还至Pool对象池中。

[0047] 当弱引用访问器关联的对象回收时,需要设置对象携带的序列号为无效值,例如可以更改对象携带的序列号,则对象的序列号会与关联的弱引用访问器中记录的序列号不一致。这里的弱引用访问器关联的对象是指弱引用访问器访问过的对象。

[0048] 结合图3所示,PoolObjectBase是支持Pool弱引用框架的基类,只有从该类型继承才能提供基于Pool的弱引用机制,提供OnUse虚函数用于类型初始化。OnRelease虚函数用于对象回收前的善后处理,Release接口用于对象的回收,实现类的其他自身需求在派生类中实现。

[0049] 在通过Pool对象池的Get方法获取对象时调用OnUse虚函数用于让该对象初始化。当外部需要回收对象时则调用对象的Release接口,该接口触发OnRelease调用过程,首先调用OnRelease虚函数给类型实现者一次回收善后处理,然后将对象的SeqNum设置为无效值,最后通过泛型对象池管理模块PoolObjectControl<T>的Release接口将对象放入对象列表(PoolList)中等待下次使用。

[0050] 本实施例中,由于当回收对象时会更改对象携带的序列号,使得对象携带的序列号会与对象之前关联的弱引用访问器中记录的标识号不一致,从而使得弱引用访问器在访问对象时判断不通过而达到失效的目的。相对于传统的引用计数的方式,该方法适用于Pool对象的弱引用访问机制,能够解决Pool对象的弱引用的失效问题。

[0051] 为了更加清楚的理解上述各实施例中的面向Pool对象的弱引用实现方法的原理,下面以一个时序图进行详细说明。

[0052] 如图4所示,当外部(对象拥有者)需要使用对象时,向对象池申请获取对象。对象池分配可用对象,并为分配的对象设定唯一的序列号SeqNum,以及调用OnUse虚函数初始化Pool对象,向对象拥有者返回对象强引用。所谓强引用,也是一种对对象的引用方式,它与对象的生命期有效性强绑定,如果对象不存在又用强引用访问会引起程序崩溃。对象池向

对象拥有者返回对象强引用,也就是将对象的指针返回给对象拥有者,以便弱引用访问者在构造弱引用访问器时能够从对象拥有者获取到对象的指针。进一步的,弱引用访问者构造弱引用访问器,以使用弱引用访问器访问对象。在构造弱引用访问器的过程中将分配的对象唯一序列号记录在其中。在对象拥有者释放对象之前都为对象的生命存续期间(如图4中的阴影部分的时序区间)。弱引用访问器在访问对象之前会进行有效性判断,如果对象携带的序列号与弱引用访问器中记录的一致,则访问。当对象拥有者需要释放对象时,调用Release接口回收对象,调用OnRelease虚函数对Pool对象进行善后处理,然后设置对象的序列号SeqNum为无效,将对象归还至Pool对象池中,从而对象回收完成。如果弱引用访问器访问对象之前进行有效性判断不通过,也就是记录的序列号与对象携带的序列号不一致,则不可访问,表示弱引用访问器已经失效。

[0053] 本方法可以解决利用Pool技术无法使用原生弱引用机制的问题,采用本方法后可以实现Pool对象的弱引用访问机制,杜绝了程序开发过程中因Pool对象生命周期长,对象可被反复重用的引用错乱等bug,极大提高了代码质量和程序安全,提升了项目开发效率。

[0054] 如图5所示,在一个实施例中,提供了一种面向Pool对象的弱引用实现装置,包括对象分配模块502、访问器构造模块504和对象访问模块506,其中:

[0055] 对象分配模块502用于通过Pool对象池分配对象,并为分配的所述对象设定唯一的标识号,该标识号在所述对象回收至Pool对象池中时失效。

[0056] 访问器构造模块504用于构造弱引用访问器并将对象的标识号记录在所弱引用访问器中。

[0057] 对象访问模块506用于通过弱引用访问器每次访问对象时,检测弱引用访问器中记录的标识号是否与对象携带的标识号一致,若是,则允许访问,否则不允许访问。

[0058] 对象是指计算机中对事物概念、数据等的抽象表达,是一种资源,需分配一定的内存资源来进行记录。Pool对象是指被分配在内存池中可以被反复申请使用和归还的程序对象。Pool对象池是Pool对象的管理器,Pool对象池可维护一个空闲对象的列表,当需要分配对象时为对象拥有者分配一个对象。在分配对象时为分配的对象设定唯一的标识号,该标识号在对象的生命续存期间有效,在对象回收至Pool对象池中时失效,对象被访问时携带对应的标识号。

[0059] 访问器是对程序对象的访问承载者,可以是指针、智能指针、强引用和弱引用等。弱引用是指程序某个模块拥有程序对象的所有权但其他模块需要访问该对象时,用于隔离和检查所持有的引用或指针是否有效的一种方法。弱引用访问器是对弱引用方法的封装,可以提供与被访问对象间的有效性判断和访问接口。在构造弱引用访问器是将要访问的对象的标识号记录在弱引用访问器中。

[0060] 本实施例中,在对象被释放前是对象的生命续存期间。由于弱引用访问器提供了与被访问的对象的有效性判断和访问接口,在对象的生命续存期间,弱引用访问器每次访问对象时,都会对标识号进行有效性判断。具体的,检测若引用访问器中记录的标识号是否与对象携带的标识号一致,若是,则允许访问,否则不允许访问,弱引用访问器失效。

[0061] 在一个实施例中,如图6所示,该面向Pool对象的弱引用实现装置还包括:对象回收模块508,用于当回收对象时设置对象携带的标识号为无效值,将对象归还至所述Pool对象池中。当对象拥有者需要回收对象时可调用对象的相应接口,该接口将对象携带的标识

号设置成无效值。具体的,可以更改对象所携带的标识号即可,这样,弱引用访问器记录的标识号与对象携带的标识号则变为不同,弱引用访问器不能访问对象。

[0062] 在一个实施例中,访问器构造模块504用于构造弱引用访问器,将对象的指针保存在弱引用访问器中,以及将对象的标识号保存在弱引用访问器的标识号变量中。

[0063] 在一个实施例中,对象访问模块506用于检测被访问的对象的指针是否不为空且弱引用访问器中记录的标识号是否与对象携带的标识号一致,若是,则允许访问,否则不允许访问。其中,标识号为全局唯一的序列号。

[0064] 本实施例中,被访问对象的指针不为空,则表明对象实际存在,进一步判断弱引用访问器中记录的标识号是否与对象携带的标识号一致,如果一致,则表明弱引用访问器未失效,可以访问该对象,如果不一致,则表明弱引用访问已失效,无法访问该对象。

[0065] 在一个实施例中,如图7所示,还提供了一种服务器,该服务器包括通过系统总线连接的处理器、非易失性存储介质、内存储器、网络接口,非易失性存储介质中存储有操作系统和一种面向Pool对象的弱引用实现装置,该面向Pool对象的弱引用实现装置用于执行一种面向Pool对象的弱引用实现方法。该处理器用于提高计算和控制能力,支撑整个服务器的运行。内存储器用于为非易失性存储介质中的面向Pool对象的弱引用实现装置的运行提供环境,该内存储器中可储存有计算机可读指令,该计算机可读指令被处理器执行时,可使得该处理器执行一种面向Pool对象的弱引用实现方法。网络接口用于与终端进行网络通信,接收或发送数据。

[0066] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成,所述的程序可存储于一计算机可读取存储介质中,该程序在执行时,可包括如上述各方法的实施例的流程。其中,所述的存储介质可为磁碟、光盘、只读存储记忆体(Read-Only Memory,ROM)或随机存储记忆体(Random Access Memory,RAM)等。

[0067] 以上所述实施例的各技术特征可以进行任意的组合,为使描述简洁,未对上述实施例中的各个技术特征所有可能的组合都进行描述,然而,只要这些技术特征的组合不存在矛盾,都应当认为是本说明书记载的范围。

[0068] 以上所述实施例仅表达了本发明的几种实施方式,其描述较为具体和详细,但并不能因此而理解为对发明专利范围的限制。应当指出的是,对于本领域的普通技术人员来说,在不脱离本发明构思的前提下,还可以做出若干变形和改进,这些都属于本发明的保护范围。因此,发明专利的保护范围应以所附权利要求为准。

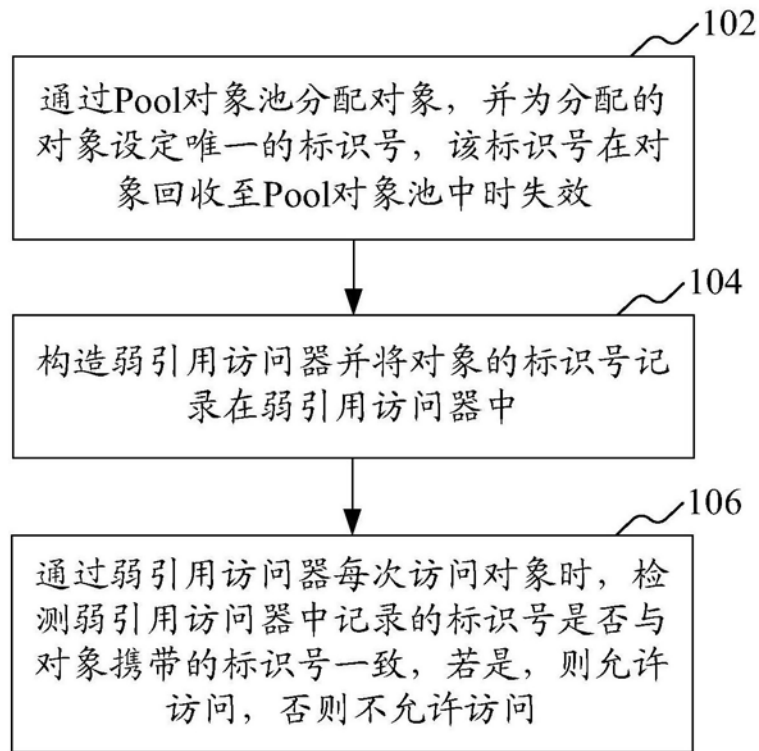


图1

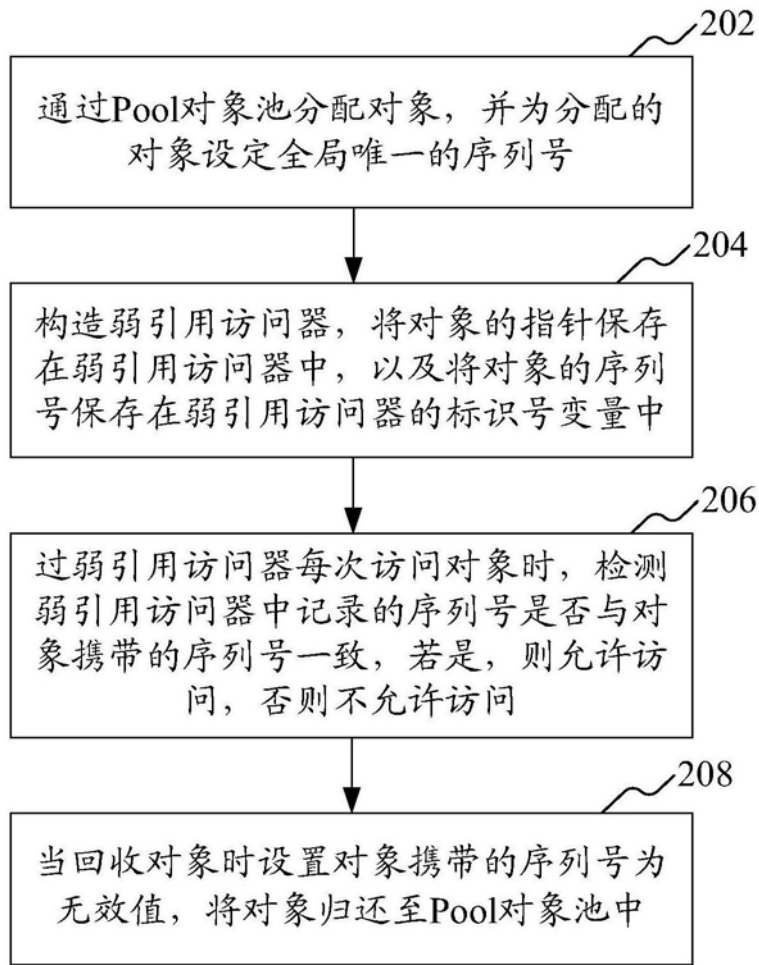


图2

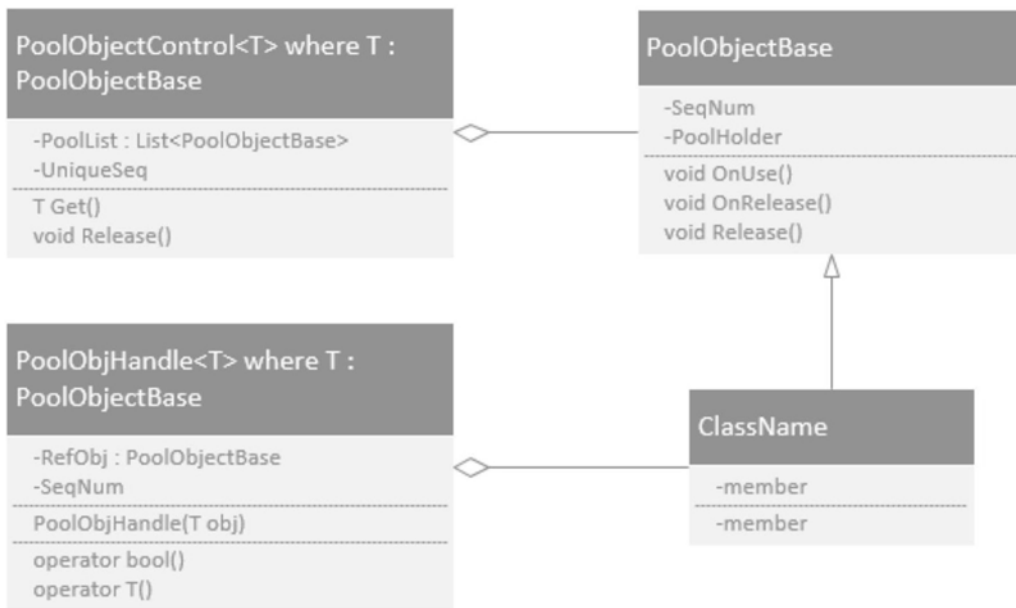


图3

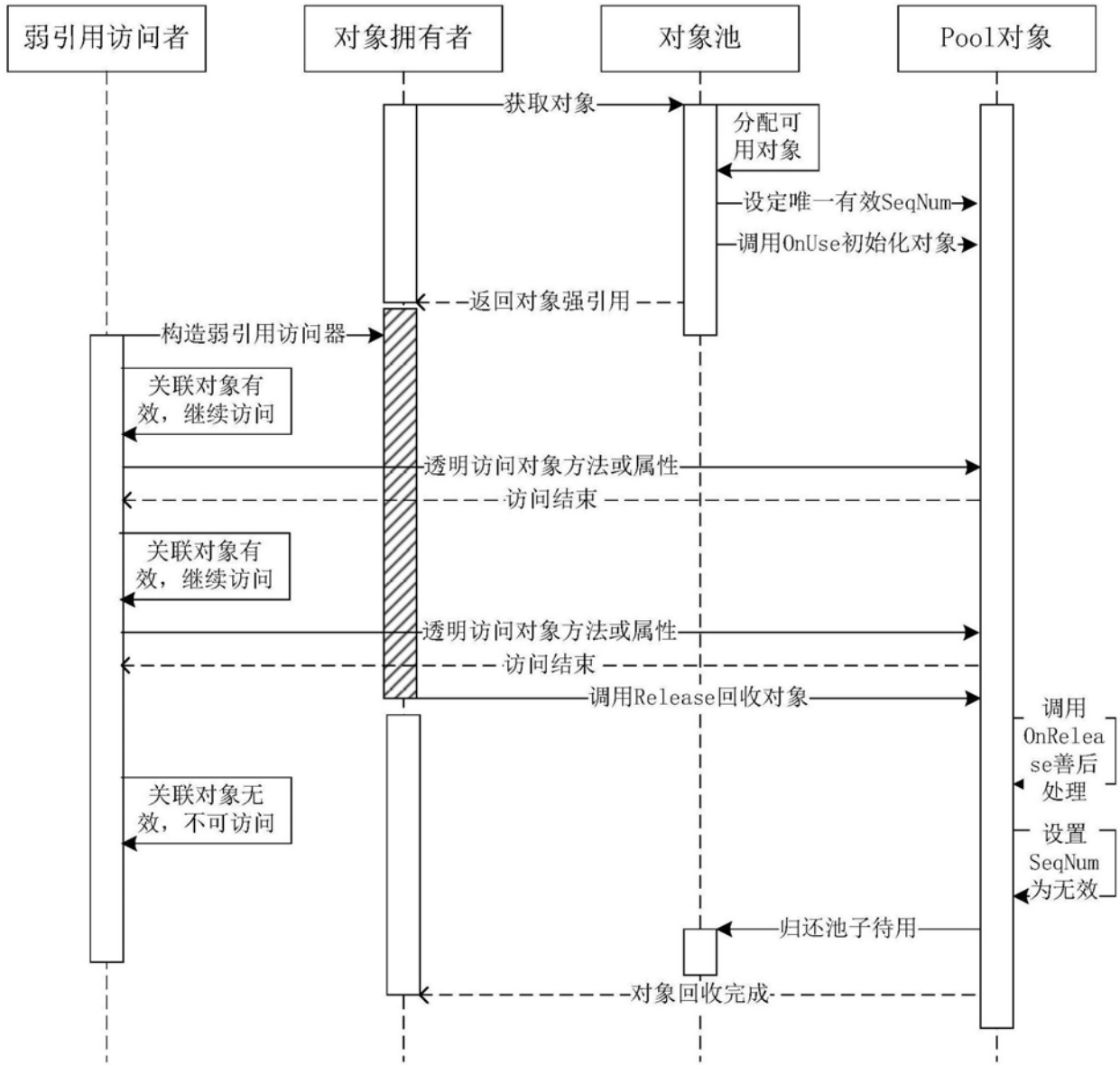


图4

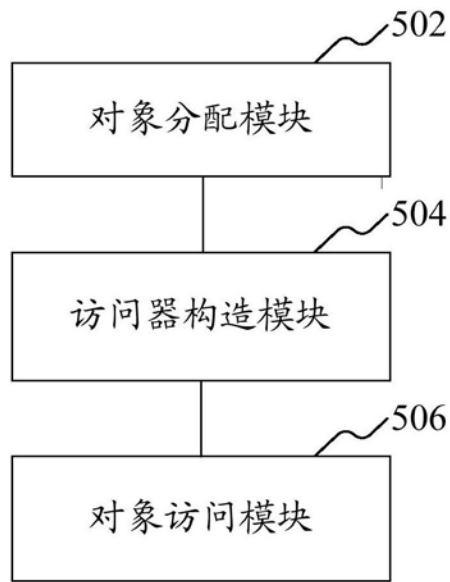


图5

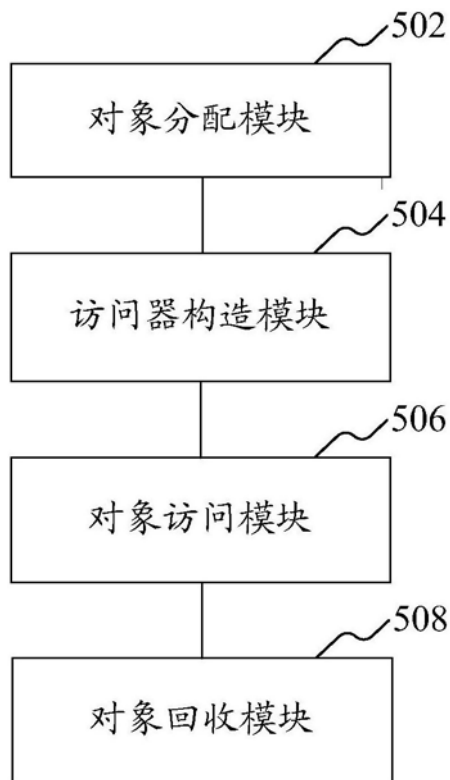


图6

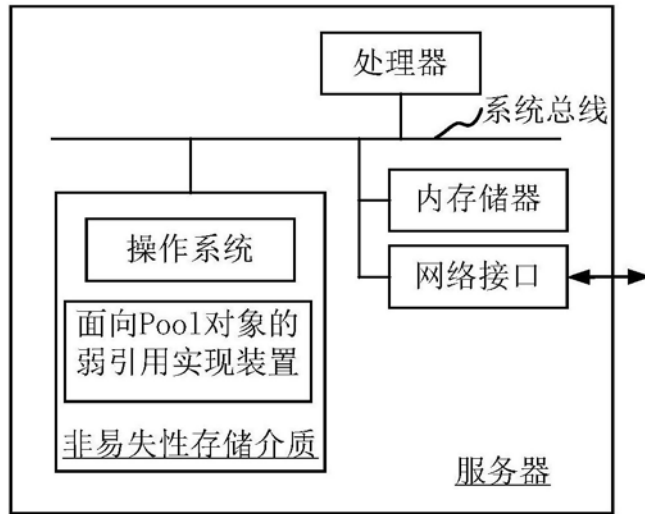


图7