



(12) 发明专利申请

(10) 申请公布号 CN 112000480 A

(43) 申请公布日 2020. 11. 27

(21) 申请号 202010863452.8

(22) 申请日 2020.08.25

(71) 申请人 深圳忆联信息系统有限公司
地址 518067 广东省深圳市南山区蛇口街
道蛇口后海大道东角头厂房D24/F-02

(72) 发明人 沈荣娟 郭芳芳 杨锦平 韩道静
吴娴

(74) 专利代理机构 深圳市精英专利事务所
44242
代理人 蒋学超

(51) Int. Cl.
G06F 9/50 (2006.01)
G06F 11/22 (2006.01)
G06F 11/34 (2006.01)

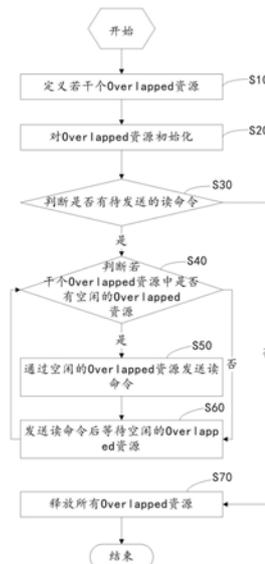
权利要求书2页 说明书8页 附图7页

(54) 发明名称

提升SSD全盘扫描效率的方法、装置、设备及介质

(57) 摘要

本发明公开了提升SSD全盘扫描效率的方法、装置、设备及介质,方法包括以下步骤:定义若干个Overlapped资源;对Overlapped资源初始化;判断是否有待发送的读命令;若是,则判断若干个Overlapped资源中是否有空闲的Overlapped资源;若是,则通过空闲的Overlapped资源发送读命令;发送读命令后等待空闲的Overlapped资源。本发明取代了多线程功能,Overlapped I/O模型是OS (Operating System) 为用户传递数据,完成上下文切换,在处理完之后通知用户。在程序中的处理,变成OS的处理,内部也是用线程来处理的。此方法可大大减少开发难度,提高开发效率,降低软件维护成本。



1. 提升SSD全盘扫描效率的方法,其特征在于,所述方法包括:
 - 定义若干个Overlapped资源;
 - 对Overlapped资源初始化;
 - 判断是否有待发送的读命令;
 - 若是,则判断若干个Overlapped资源中是否有空闲的Overlapped资源;
 - 若是,则通过空闲的Overlapped资源发送读命令;
 - 发送读命令后等待空闲的Overlapped资源。
2. 根据权利要求1所述的提升SSD全盘扫描效率的方法,其特征在于,所述判断是否有待发送的读命令的步骤之后,还包括以下步骤:
 - 若否,则释放所有Overlapped资源。
3. 根据权利要求1所述的提升SSD全盘扫描效率的方法,其特征在于,所述判断若干个Overlapped资源中是否有空闲的Overlapped资源中,获取空闲的Overlapped资源具体包括以下步骤:
 - 判断定义的若干个Overlapped资源中空闲的Overlapped资源数是否大于零;
 - 若是,则依次索引定义的若干个Overlapped资源;
 - 判断当前索引的Overlapped资源中事件内核对象成员是否为非法句柄;
 - 若是,则通过CreateEvent函数为当前Overlapped资源创建一个事件句柄,且相应的定义的若干个Overlapped资源中空闲的Overlapped资源减少一个。
4. 根据权利要求1所述的提升SSD全盘扫描效率的方法,其特征在于,所述发送读命令后等待空闲的Overlapped资源的步骤中,等待释放空闲的Overlapped资源具体包括以下步骤:
 - 从第零个开始依次索引定义的若干个Overlapped资源;
 - 通过GetOverlappedresult()函数查看当前索引的Overlapped资源的结果,判断其命令是否发送完毕;
 - 若是,则释放当前的Overlapped资源为空闲的Overlapped资源,且相应的定义的若干个Overlapped资源中空闲的Overlapped资源增加一个。
5. 提升SSD全盘扫描效率的装置,其特征在于,包括定义单元、初始化单元、第一判断单元、第二判断单元、读命令发送单元以及等待单元;
 - 所述定义单元,用于定义若干个Overlapped资源;
 - 所述初始化单元,用于对Overlapped资源初始化;
 - 所述第一判断单元,用于判断是否有待发送的读命令;
 - 所述第二判断单元,用于判断若干个Overlapped资源中是否有空闲的Overlapped资源;
 - 所述读命令发送单元,用于通过空闲的Overlapped资源发送读命令;
 - 所述等待单元,用于发送读命令后等待空闲的Overlapped资源。
6. 根据权利要求5所述的提升SSD全盘扫描效率的装置,其特征在于,还包括释放单元;
 - 所述释放单元,用于释放所有Overlapped资源。
7. 根据权利要求5所述的提升SSD全盘扫描效率的装置,其特征在于,第二判断单元包括第一判断模块、第一索引模块、第二判断模块以及事件句柄创建模块;

所述第一判断模块,用于判断定义的若干个Overlapped资源中空闲的Overlapped资源数是否大于零;

所述第一索引模块,用于依次索引定义的若干个Overlapped资源;

所述第二判断模块,用于判断当前索引的Overlapped资源中事件内核对象成员是否为非法句柄;

所述事件句柄创建模块,用于通过CreateEvent函数为当前Overlapped资源创建一个事件句柄,且相应的定义的若干个Overlapped资源中空闲的Overlapped资源减少一个。

8. 根据权利要求5所述的提升SSD全盘扫描效率的装置,其特征在于,所述等待单元包括第二索引模块、第三判断模块以及释放模块;

所述第二索引模块,用于从第零个开始依次索引定义的若干个Overlapped资源;

所述第三判断模块,用于通过GetOverlappedresult()函数查看当前索引的Overlapped资源的结果,判断其命令是否发送完毕;

所述释放模块,用于释放当前的Overlapped资源为空闲的Overlapped资源,且相应的定义的若干个Overlapped资源中空闲的Overlapped资源增加一个。

9. 一种计算机设备,其特征在于,包括存储器、处理器以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,所述处理器执行所述计算机程序时实现如权利要求1~4中任意一项所述的提升SSD全盘扫描效率的方法步骤。

10. 一种存储介质,其特征在于,所述存储介质存储有计算机程序,所述计算机程序包括程序指令,所述程序指令被处理器执行时,使得所述处理器执行如权利要求1~4任意一项所述的接提升SSD全盘扫描效率的方法步骤。

提升SSD全盘扫描效率的方法、装置、设备及介质

技术领域

[0001] 本发明涉及固态硬盘,更具体地说是一种提升SSD全盘扫描效率的方法、装置、设备及介质。

背景技术

[0002] 近年来,SSD(Solid State Disk,固态硬盘)产品因其高速的读写性能,得到了空前的发展,众多厂家投入了大量的人力和精力来做SSD产品,对于SSD的性能要求也越来越高,为了保证产品的质量,需要经过一系列的测试与验证。其中全盘扫描或指定存储范围对盘扫描则是其中一项测试,通常会对测试时间又一定的要求。

[0003] 对SSD进行全盘扫描的实现并不复杂,只要按照协议发送NVMe(Non-Volatile Memory express)读命令即可,但由于对测试时间的要求,通常单线程的发送读命令耗时过长,也没有将SSD的性能发挥出来。使用多线程的方式可行,但是麻烦,多线程涉及同步机制,错误处理,在成千上万个线程I/O中,线程上下文切换是一个十分消耗CPU资源的。

发明内容

[0004] 本发明的目的在于克服现有技术的不足,提供提升SSD全盘扫描效率的方法、装置、设备及介质。

[0005] 为实现上述目的,本发明采用以下技术方案:

[0006] 第一方面,提升SSD全盘扫描效率的方法,所述方法包括:

[0007] 定义若干个Overlapped资源;

[0008] 对Overlapped资源初始化;

[0009] 判断是否有待发送的读命令;

[0010] 若是,则判断若干个Overlapped资源中是否有空闲的Overlapped资源;

[0011] 若是,则通过空闲的Overlapped资源发送读命令;

[0012] 发送读命令后等待空闲的Overlapped资源。

[0013] 其进一步技术方案为:所述判断是否有待发送的读命令的步骤之后,还包括以下步骤:

[0014] 若否,则释放所有Overlapped资源。

[0015] 其进一步技术方案为:所述判断若干个Overlapped资源中是否有空闲的Overlapped资源中,获取空闲的Overlapped资源具体包括以下步骤:

[0016] 判断定义的若干个Overlapped资源中空闲的Overlapped资源数是否大于零;

[0017] 若是,则依次索引定义的若干个Overlapped资源;

[0018] 判断当前索引的Overlapped资源中事件内核对象成员是否为非法句柄;

[0019] 若是,则通过CreateEvent函数为当前Overlapped资源创建一个事件句柄,且相应的定义的若干个Overlapped资源中空闲的Overlapped资源减少一个。

[0020] 其进一步技术方案为:所述发送读命令后等待空闲的Overlapped资源的步骤中,

等待释放空闲的Overlapped资源具体包括以下步骤:

[0021] 从第零个开始依次索引定义的若干个Overlapped资源;

[0022] 通过GetOverlappedresult()函数查看当前索引的Overlapped资源的结果,判断其命令是否发送完毕;

[0023] 若是,则释放当前的Overlapped资源为空闲的Overlapped资源,且相应的定义的若干个Overlapped资源中空闲的Overlapped资源增加一个。

[0024] 第二方面,提升SSD全盘扫描效率的装置,包括定义单元、初始化单元、第一判断单元、第二判断单元、读命令发送单元以及等待单元;

[0025] 所述定义单元,用于定义若干个Overlapped资源;

[0026] 所述初始化单元,用于对Overlapped资源初始化;

[0027] 所述第一判断单元,用于判断是否有待发送的读命令;

[0028] 所述第二判断单元,用于判断若干个Overlapped资源中是否有空闲的Overlapped资源;

[0029] 所述读命令发送单元,用于通过空闲的Overlapped资源发送读命令;

[0030] 所述等待单元,用于发送读命令后等待空闲的Overlapped资源。

[0031] 其进一步技术方案为:还包括释放单元;

[0032] 所述释放单元,用于释放所有Overlapped资源。

[0033] 其进一步技术方案为:第二判断单元包括第一判断模块、第一索引模块、第二判断模块以及事件句柄创建模块;

[0034] 所述第一判断模块,用于判断定义的若干个Overlapped资源中空闲的Overlapped资源数是否大于零;

[0035] 所述第一索引模块,用于依次索引定义的若干个Overlapped资源;

[0036] 所述第二判断模块,用于判断当前索引的Overlapped资源中事件内核对象成员是否为非法句柄;

[0037] 所述事件句柄创建模块,用于通过CreateEvent函数为当前Overlapped资源创建一个事件句柄,且相应的定义的若干个Overlapped资源中空闲的Overlapped资源减少一个。

[0038] 其进一步技术方案为:所述等待单元包括第二索引模块、第三判断模块以及释放模块;

[0039] 所述第二索引模块,用于从第零个开始依次索引定义的若干个Overlapped资源;

[0040] 所述第三判断模块,用于通过GetOverlappedresult()函数查看当前索引的Overlapped资源的结果,判断其命令是否发送完毕;

[0041] 所述释放模块,用于释放当前的Overlapped资源为空闲的Overlapped资源,且相应的定义的若干个Overlapped资源中空闲的Overlapped资源增加一个。

[0042] 第三方面,一种计算机设备,包括存储器、处理器以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,所述处理器执行所述计算机程序时实现如上述的提升SSD全盘扫描效率的方法步骤。

[0043] 第四方面,一种存储介质,所述存储介质存储有计算机程序,所述计算机程序包括程序指令,所述程序指令被处理器执行时,使得所述处理器执行如上述的接提升SSD全盘扫

描效率的方法步骤。

[0044] 本发明与现有技术相比的有益效果是：本发明取代了多线程功能，Overlapped I/O模型是OS (Operating System) 为用户传递数据，完成上下文切换，在处理完之后通知用户。在程序中的处理，变成OS的处理，内部也是用线程来处理的。此方法可大大减少开发难度，提高开发效率，降低软件维护成本。

[0045] 上述说明仅是本发明技术方案的概述，为了能够更清楚了解本发明技术手段，可依照说明书的内容予以实施，并且为了让本发明的上述和其它目的、特征及优点能够更明显易懂，以下特举较佳实施例，详细说明如下。

附图说明

[0046] 图1为现有技术的单线程方式发送读命令示意图；

[0047] 图2为本发明使用Overlapped I/O方式发送读命令示意图；

[0048] 图3为本发明提升SSD全盘扫描效率的方法具体实施例的流程图一；

[0049] 图4为本发明提升SSD全盘扫描效率的方法具体实施例的流程图二；

[0050] 图5为本发明提升SSD全盘扫描效率的方法具体实施例的流程图三；

[0051] 图6为本发明提升SSD全盘扫描效率的装置具体实施例的示意性框图一；

[0052] 图7为本发明提升SSD全盘扫描效率的装置具体实施例的示意性框图二；

[0053] 图8为本发明提升SSD全盘扫描效率的装置具体实施例的示意性框图三；

[0054] 图9为本发明一种计算机设备的示意性框图。

具体实施方式

[0055] 为了更充分理解本发明的技术内容，下面结合具体实施例对本发明的技术方案进一步介绍和说明，但不局限于此。

[0056] 下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，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，都属于本发明保护的范围。

[0057] 应当理解，当在本说明书和所附权利要求书中使用时，术语“包括”和“包含”指示所描述特征、整体、步骤、操作、元素和/或组件的存在，但并不排除一个或多个其它特征、整体、步骤、操作、元素、组件和/或其集合的存在或添加。

[0058] 还应当理解，在此本发明说明书中所使用的术语仅仅是出于描述特定实施例的目的而并不意在限制本发明。如在本发明说明书和所附权利要求书中所使用的那样，除非上下文清楚地指明其它情况，否则单数形式的“一”、“一个”及“该”意在包括复数形式。

[0059] 还应当进一步理解，在本发明说明书和所附权利要求书中使用的术语“和/或”是指相关联列出的项中的一个或多个的任何组合以及所有可能组合，并且包括这些组合。

[0060] 现有单线程方式发送读命令，请参考图1，Host是单通道往SSD发送读命令，一个命令完成了，再发下一个，这种方式可以保证读命令的顺序执行，但没有将NVMe的性能发挥出来，NVMe支持的命令队列可达到64K。而本发明则是通过Overlapped I/O方式发送读命令，请参考图2，Overlapped是异步I/O模型。异步I/O和同步I/O不同，同步I/O时，程序被挂起，

一直到I/O处理完,程序才能获得控制。异步I/O,调用一个函数告诉OS,进行I/O操作,不等I/O结束就立即返回,继续程序进行,操作系统完成I/O之后,发通知消息。Overlapped I/O取代多线程功能,OS传递数据,完成上下文的切换,在处理完之后通知。Overlapped的数据结构如下:

```
[0061] typedef struct_OVERLAPPED {  
[0062]     DWORD Internal;  
[0063]     DWORD InternalHigh;  
[0064]     DWORD Offset;  
[0065]     DWORD OffsetHigh;  
[0066]     HANDLE hEvent;  
[0067] } OVERLAPPED, *LPOVERLAPPED;
```

[0068] 通过发送多个Overlapped请求,每个overlapped处理独立的请求操作,对于同一个文件Handle,系统有多个异步操作时,可以通过Overlapped中的hEvent来标识事件内核对象。CreateEvent为每个请求创建一个事件,初始化每个请求的hEvent成员,对同一文件多个读写请求,每个操作绑定一个event对象,调用GetOverlappedresult来等待其中一个(或全部)完成。

[0069] 请参考图3,本发明提升SSD全盘扫描效率的方法,该方法包括以下步骤:

[0070] S10、定义若干个Overlapped资源,下一步执行S20;

[0071] S20、对Overlapped资源初始化,下一步执行S30;

[0072] S30、判断是否有待发送的读命令,若是,则执行步骤S40,若否,则执行步骤S70;

[0073] S40、判断若干个Overlapped资源中是否有空闲的Overlapped资源,若是,则执行步骤S50,若否,则执行步骤S60;

[0074] S50、通过空闲的Overlapped资源发送读命令,下一步执行S60;

[0075] S60、发送读命令后等待空闲的Overlapped资源。

[0076] S70、释放所有Overlapped资源。

[0077] 具体的,通过首先定义一个宏,用来标识需要开多少个overlapped资源,定义N个资源,然后做初始化,如果存在待发送的读命令,则去查找是否有空闲的OVP (overlapped缩写)资源,如果有的话,则通过OVP发送读命令,即在deviceIoControl中最后一个参数Overlapped赋值为获取到的空闲OVP资源。发送完后等待空闲OVP资源释放,有空闲资源时继续发送读命令,没有空闲的话,继续等待,如此循环,直至所有的待发送读命令都发送完成。

[0078] 请参考图4,判断若干个Overlapped资源中是否有空闲的Overlapped资源中,获取空闲的Overlapped资源具体包括以下步骤:

[0079] S401、判断定义的若干个Overlapped资源中空闲的Overlapped资源数是否大于零,若是,则执行步骤S402,若否,则结束整个流程;

[0080] S402、依次索引定义的若干个Overlapped资源,下一步执行S403;

[0081] S403、判断当前索引的Overlapped资源中事件内核对象成员是否为非法句柄,若是,则执行步骤S404,若否,则结束整个流程;

[0082] S404、通过CreateEvent函数为当前Overlapped资源创建一个事件句柄,且相应的

定义的若干个Overlapped资源中空闲的Overlapped资源减少一个。

[0083] 具体的,当如果有空闲资源,则索引这N个OVP,从第0个开始,判断OVP中hEvent成员是否为INVALID(非法)句柄,初始化为非法句柄,用来标识空闲状态。如果是非法句柄,则通过CreateEvent函数创建一个事件句柄,将其赋值给OVP的hEvent成员,相应的OVP空闲资源数减1,返回该OVP资源。

[0084] 请参考图5,发送读命令后等待空闲的Overlapped资源的步骤中,等待释放空闲的Overlapped资源具体包括以下步骤:

[0085] S601、从第零个开始依次索引定义的若干个Overlapped资源,下一步执行S602;

[0086] S602、通过GetOverlappedresult()函数查看当前索引的Overlapped资源的结果,判断其命令是否发送完毕,若是,则执行步骤S603,若否,则执行步骤S604;

[0087] S603、释放当前的Overlapped资源为空闲的Overlapped资源,且相应的定义的若干个Overlapped资源中空闲的Overlapped资源增加一个;

[0088] S604、判断是否为最后一个Overlapped资源,若是,则结束整个流程,若否,则进入步骤S601。

[0089] 具体的,首先索引这N个OVP,从第0个开始,通过GetOverlappedresult()函数查看该OVP结果,将最后一个参数bWait置为False,表示不等待I/O执行完,异步操作。如果结果为完成,则释放当前的OVP资源,将OVP空闲资源数加1,同时将hEvent成员事件句柄置为非法句柄,表示空闲资源。如果一直等待不到空闲资源,则直接返回结果,在主程序中继续调用该模块,继续等待空闲资源。

[0090] 应理解,上述实施例中各步骤的序号的大小并不意味着执行顺序的先后,各过程的执行顺序应以其功能和内在逻辑确定,而不应对本发明实施例的实施过程构成任何限定。

[0091] 对应于上述的提升SSD全盘扫描效率的方法,本发明还提供了提升SSD全盘扫描效率的装置。请参考图6,该装置包括定义单元1、初始化单元2、第一判断单元3、第二判断单元4、读命令发送单元5、等待单元6以及释放单元7;

[0092] 定义单元1,用于定义若干个Overlapped资源;

[0093] 初始化单元2,用于对Overlapped资源初始化;

[0094] 第一判断单元3,用于判断是否有待发送的读命令;

[0095] 第二判断单元4,用于判断若干个Overlapped资源中是否有空闲的Overlapped资源;

[0096] 读命令发送单元5,用于通过空闲的Overlapped资源发送读命令;

[0097] 等待单元6,用于发送读命令后等待空闲的Overlapped资源;

[0098] 释放单元7,用于释放所有Overlapped资源。

[0099] 进一步的,请参考图7,第二判断单元4包括第一判断模块41、第一索引模块42、第二判断模块43以及事件句柄创建模块44;

[0100] 第一判断模块41,用于判断定义的若干个Overlapped资源中空闲的Overlapped资源数是否大于零;

[0101] 第一索引模块42,用于依次索引定义的若干个Overlapped资源;

[0102] 第二判断模块43,用于判断当前索引的Overlapped资源中事件内核对象成员是否

为非法句柄；

[0103] 事件句柄创建模块44,用于通过CreateEvent函数为当前Overlapped资源创建一个事件句柄,且相应的定义的若干个Overlapped资源中空闲的Overlapped资源减少一个。

[0104] 进一步的,请参考图8,等待单元6包括第二索引模块61、第三判断模块62以及释放模块63;

[0105] 第二索引模块61,用于从第零个开始依次索引定义的若干个Overlapped资源;

[0106] 第三判断模块62,用于通过GetOverlappedresult()函数查看当前索引的Overlapped资源的结果,判断其命令是否发送完毕;

[0107] 释放模块63,用于释放当前的Overlapped资源为空闲的Overlapped资源,且相应的定义的若干个Overlapped资源中空闲的Overlapped资源增加一个。

[0108] 如图9所示,本发明还提供了一种计算机设备,包括存储器、处理器以及存储在存储器上并可在处理器上运行的计算机程序,处理器执行计算机程序时实现如上述的提升SSD全盘扫描效率的方法步骤。

[0109] 该计算机设备700可以是终端或服务器。该计算机设备700包括通过系统总线710连接的处理器720、存储器和网络接口750,其中,存储器可以包括非易失性存储介质730和内存存储器740。

[0110] 该非易失性存储介质730可存储操作系统731和计算机程序732。该计算机程序732被执行时,可使得处理器720执行任意一种提升SSD全盘扫描效率的方法。

[0111] 该处理器720用于提供计算和控制能力,支撑整个计算机设备700的运行。

[0112] 该内存存储器740为非易失性存储介质730中的计算机程序732的运行提供环境,该计算机程序732被处理器720执行时,可使得处理器720执行任意一种提升SSD全盘扫描效率的方法。

[0113] 该网络接口750用于进行网络通信,如发送分配的任务等。本领域技术人员可以理解,图9中示出的结构,仅仅是与本申请方案相关的部分结构的框图,并不构成对本申请方案所应用于其上的计算机设备700的限定,具体的计算机设备700可以包括比图中所示更多或更少的部件,或者组合某些部件,或者具有不同的部件布置。其中,所述处理器720用于运行存储在存储器中的程序代码,以实现以下步骤:

[0114] 定义若干个Overlapped资源;

[0115] 对Overlapped资源初始化;

[0116] 判断是否有待发送的读命令;

[0117] 若是,则判断若干个Overlapped资源中是否有空闲的Overlapped资源;

[0118] 若是,则通过空闲的Overlapped资源发送读命令;

[0119] 发送读命令后等待空闲的Overlapped资源。

[0120] 其进一步技术方案为:所述判断是否有待发送的读命令的步骤之后,还包括以下步骤:

[0121] 若否,则释放所有Overlapped资源。

[0122] 其进一步技术方案为:所述判断若干个Overlapped资源中是否有空闲的Overlapped资源中,获取空闲的Overlapped资源具体包括以下步骤:

[0123] 判断定义的若干个Overlapped资源中空闲的Overlapped资源数是否大于零;

[0124] 若是,则依次索引定义的若干个Overlapped资源;

[0125] 判断当前索引的Overlapped资源中事件内核对象成员是否为非法句柄;

[0126] 若是,则通过CreateEvent函数为当前Overlapped资源创建一个事件句柄,且相应的定义的若干个Overlapped资源中空闲的Overlapped资源减少一个。

[0127] 其进一步技术方案为:所述发送读命令后等待空闲的Overlapped资源的步骤中,等待释放空闲的Overlapped资源具体包括以下步骤:

[0128] 从第零个开始依次索引定义的若干个Overlapped资源;

[0129] 通过GetOverlappedresult()函数查看当前索引的Overlapped资源的结果,判断其命令是否发送完毕;

[0130] 若是,则释放当前的Overlapped资源为空闲的Overlapped资源,且相应的定义的若干个Overlapped资源中空闲的Overlapped资源增加一个。

[0131] 应当理解,在本申请实施例中,处理器720可以是中央处理单元(Central Processing Unit,CPU),该处理器720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)、现成可编程门阵列(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、分立硬件组件等。其中,通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0132] 本领域技术人员可以理解,图9中示出的计算机设备700结构并不构成对计算机设备700的限定,可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件布置。

[0133] 所述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,可以存储在一个计算机可读取存储介质中。基于这样的理解,本发明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,该计算机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台计算机设备(可以是个人计算机,服务器,或者网络设备等)或处理器(processor)执行本发明中各个实施例所述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(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0134] 所属领域的技术人员可以清楚地了解到,为了描述的方便和简洁,仅以上述各功能单元、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单元、模块完成,即将所述装置的内部结构划分成不同的功能单元或模块,以完成以上描述的全部或者部分功能。实施例中的各功能单元、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。另外,各功能单元、模块的具体名称也只是为了便于相互区分,并不用于限制本申请的保护范围。上述装置中单元、模块的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0135] 本领域普通技术人员可以意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员

可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本发明的范围。

[0136] 在本发明所提供的实施例中,应该理解到,所揭露的装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,所述模块或单元的划分,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之间的耦合或直接耦合或通讯连接可以是通过一些接口,装置或单元的间接耦合或通讯连接,可以是电性,机械或其它的形式。

[0137] 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。

[0138] 上述仅以实施例来进一步说明本发明的技术内容,以便于读者更容易理解,但不代表本发明的实施方式仅限于此,任何依本发明所做的技术延伸或再创造,均受本发明的保护。本发明的保护范围以权利要求书为准。

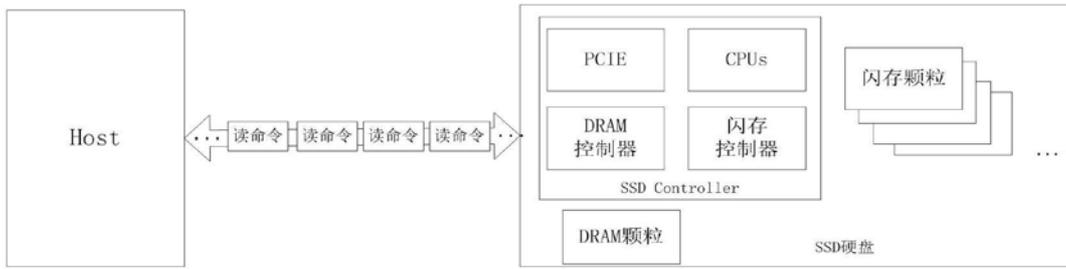


图1

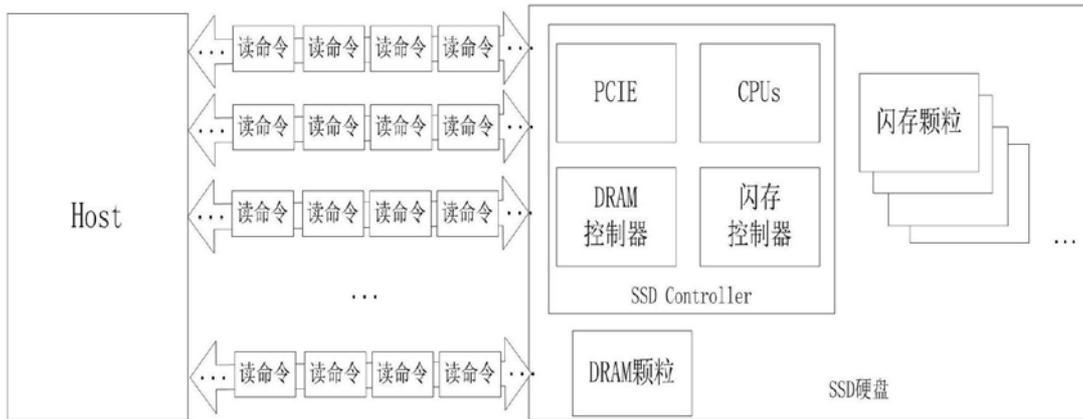


图2

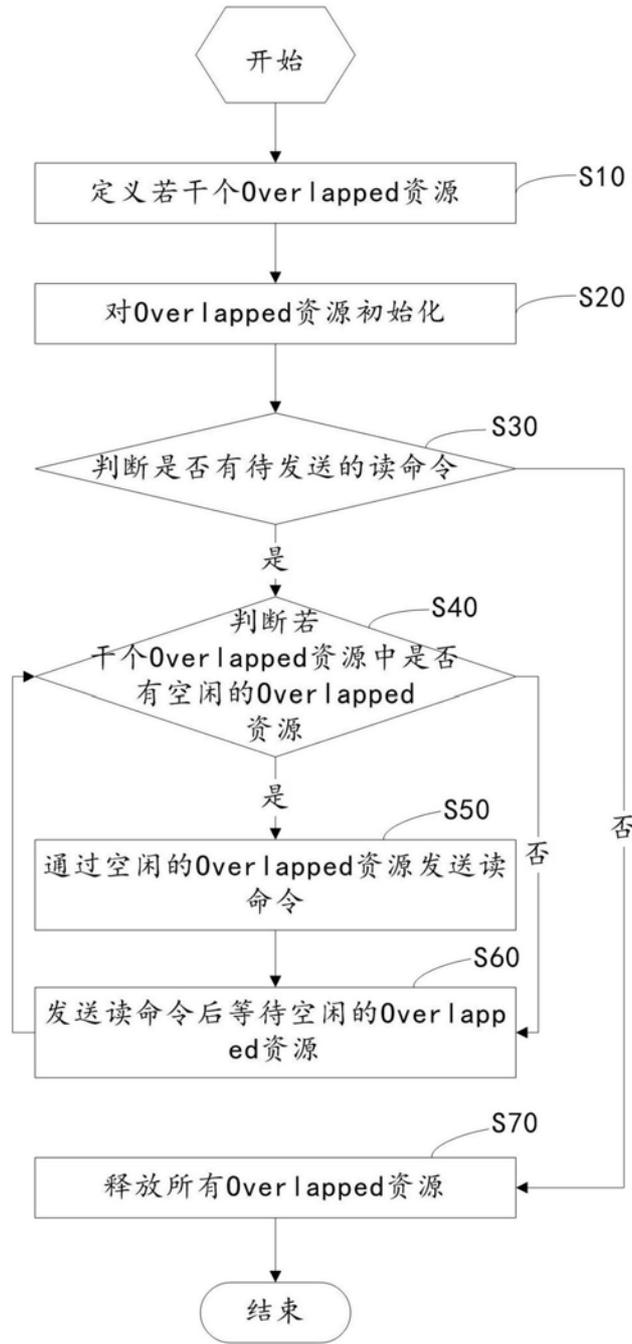


图3

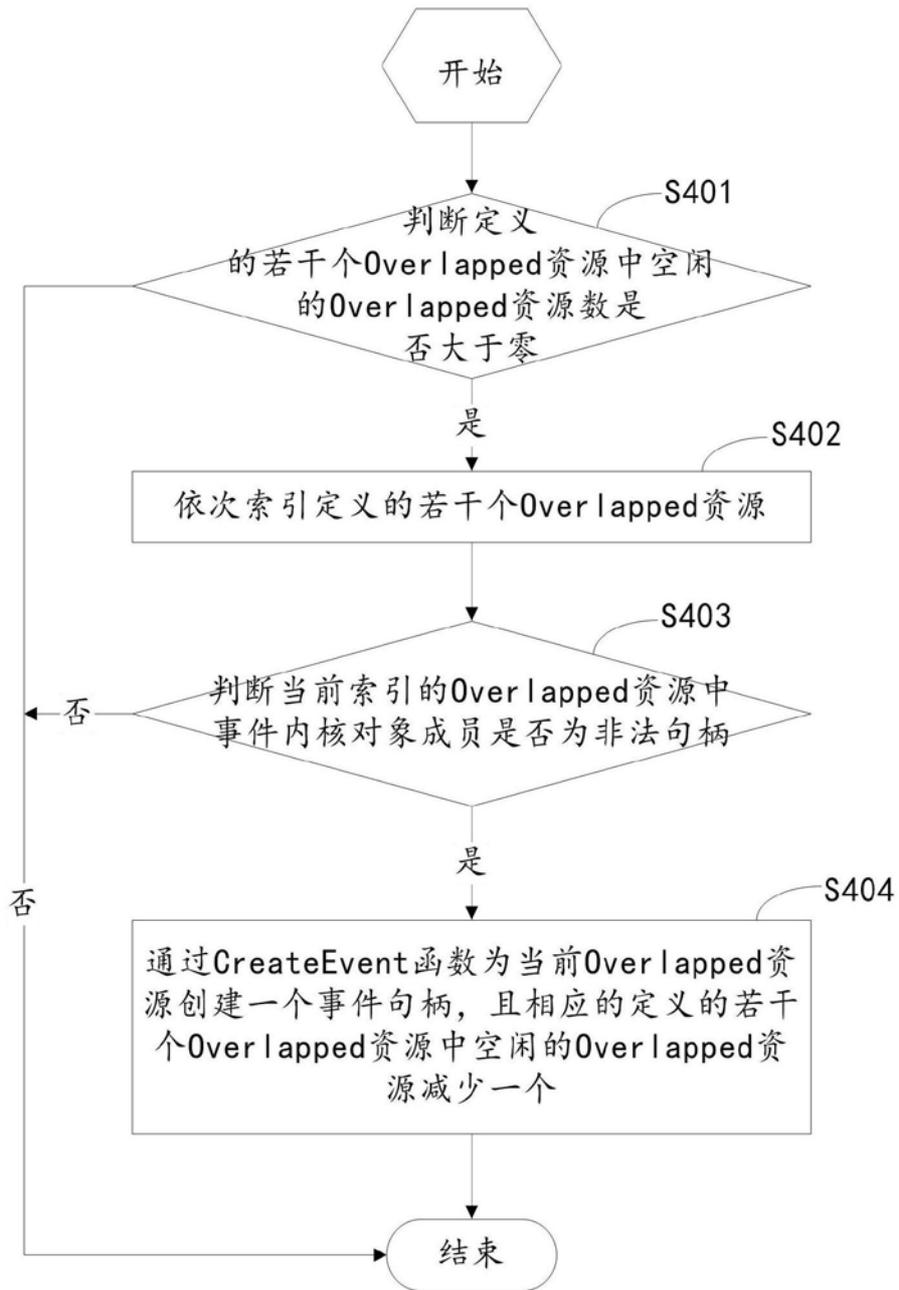


图4

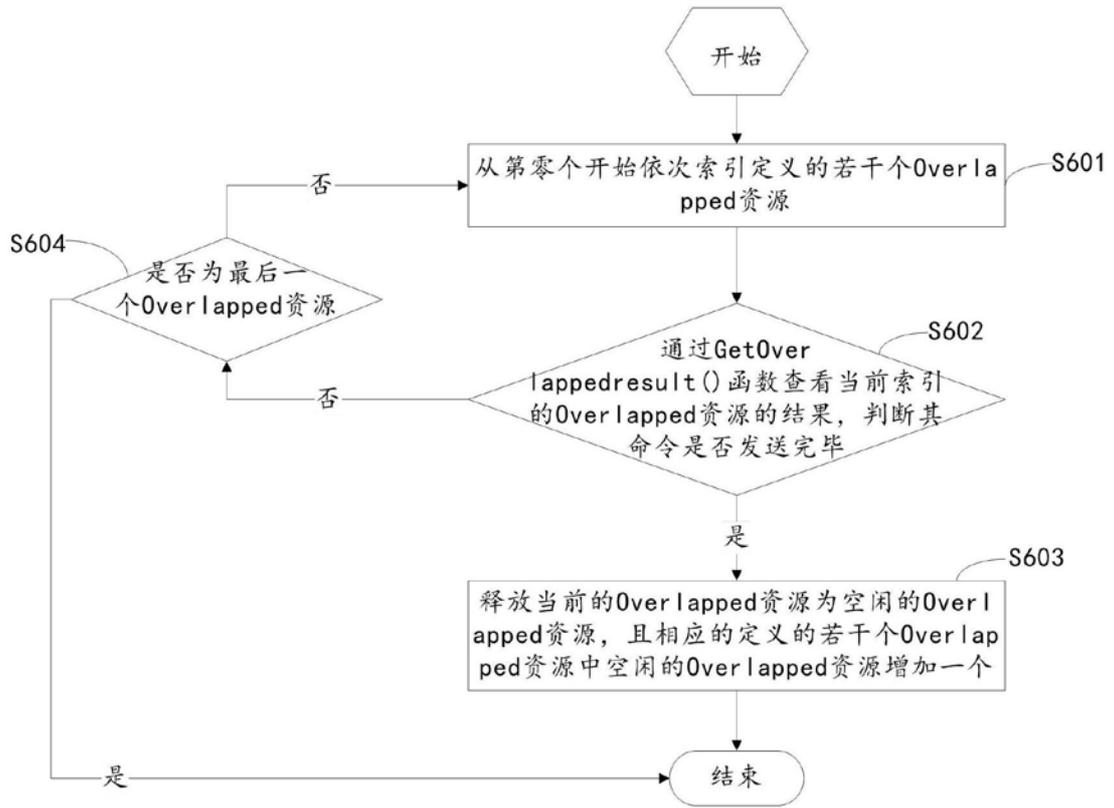


图5



图6

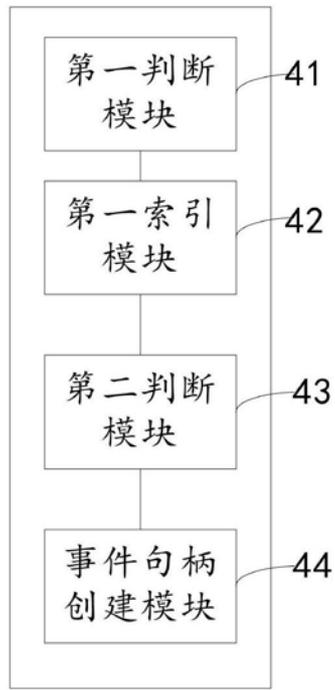


图7

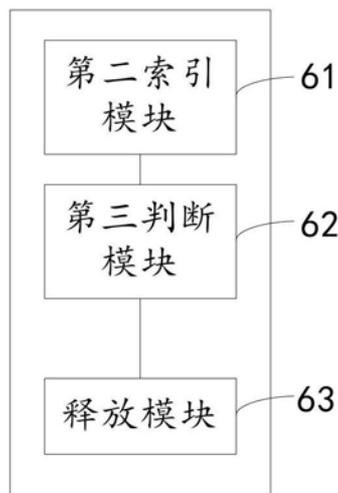


图8

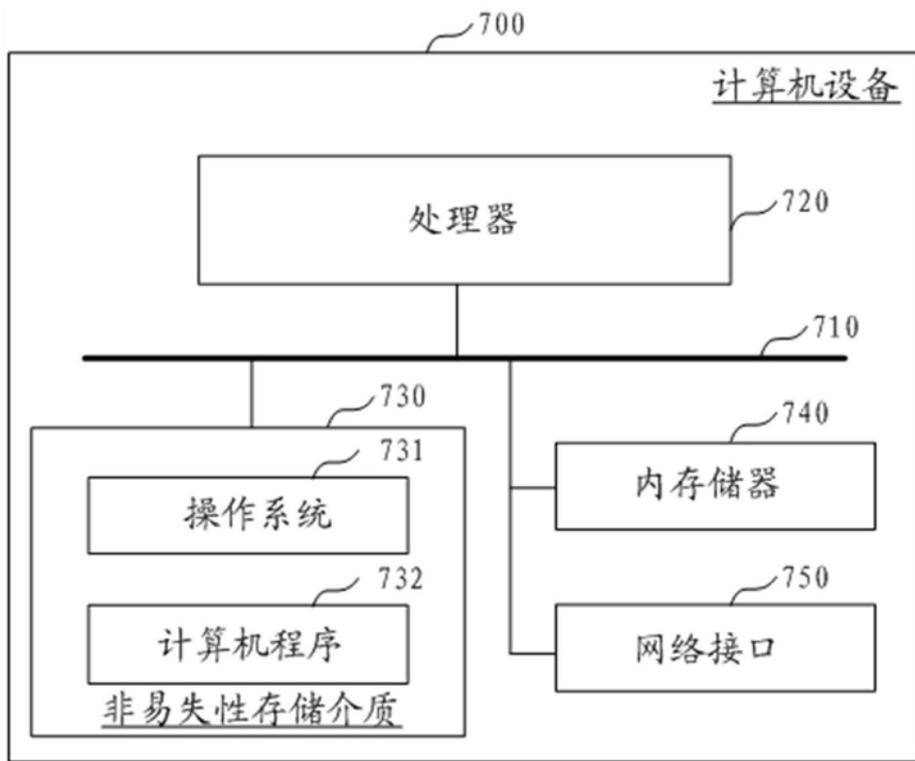


图9