



(12) 发明专利申请

(10) 申请公布号 CN 117519598 A

(43) 申请公布日 2024. 02. 06

(21) 申请号 202311577985.X

(22) 申请日 2023.11.23

(71) 申请人 成都芯忆联信息技术有限公司

地址 610000 四川省成都市中国(四川)自由贸易试验区成都高新区天府四街300号6栋A座7层703-719号

(72) 发明人 梁绍宁 刘金华 叶秀阳

(74) 专利代理机构 深圳市精英专利事务所

44242

专利代理师 迟珊珊

(51) Int. Cl.

G06F 3/06 (2006.01)

权利要求书3页 说明书9页 附图6页

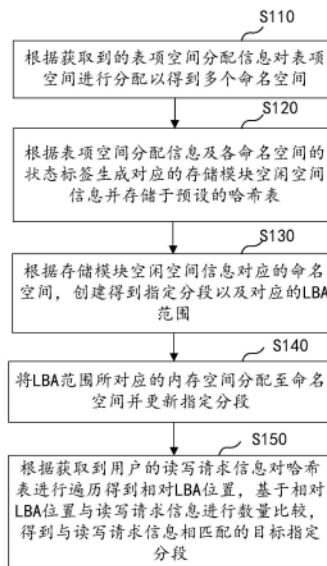
(54) 发明名称

命名空间分段存储优化方法、装置、计算机设备及介质

(57) 摘要

本发明公开了命名空间分段存储优化方法、装置、计算机设备及介质,方法包括根据获取到的表项空间分配信息对表项空间进行分配以得到多个命名空间;根据表项空间分配信息及各命名空间的状态标签生成对应的存储模块空闲空间信息;根据存储模块空闲空间信息对应的命名空间创建得到指定分段以及对应的LBA范围;将LBA范围所对应的内存空间分配至命名空间并更新指定分段;根据获取到用户的读写请求信息对哈希表进行遍历得到相对LBA位置,基于相对LBA位置与读写请求信息进行数量比较,得到与读写请求信息相匹配的目标指定分段。上述方法通过让NS支持配置多段LBA范围,使其能够处理不连续的空间,同时以较小的代价避免剩余空间不连续导致创建NS失败的问题。

CN 117519598 A



1. 一种命名空间分段存储优化方法,其特征在于,包括:

根据获取到的表项空间分配信息对表项空间进行分配以得到多个命名空间;所述表项空间分配信息包括每个命名空间支持的最大分段数量、存储模块支持的最大命名空间数以及分配结构数据;

根据所述表项空间分配信息及各所述命名空间的状态标签生成对应的存储模块空闲空间信息并存储于预设的哈希表;

根据所述存储模块空闲空间信息对应的所述命名空间,创建得到指定分段以及对应的LBA范围;

将所述LBA范围所对应的内存空间分配至所述命名空间并更新所述指定分段;

根据获取到用户的读写请求信息对所述哈希表进行遍历得到相对LBA位置,基于所述相对LBA位置与所述读写请求信息进行数量比较,得到与所述读写请求信息相匹配的目标指定分段。

2. 根据权利要求1所述的命名空间分段存储优化方法,其特征在于,所述根据获取到的表项空间分配信息对表项空间进行分配以得到多个命名空间,包括:

根据每个命名空间所支持的最大分段数以及存储模块所支持的最大命名空间数计算得到每个所述命名空间所需的表项空间大小;

将所有所述命名空间所需的所述表项空间大小累加得到总表项空间大小;

根据所述总表项空间大小及各所述命名空间的表项空间大小对所述表项空间进行空间分配,以得到多个命名空间;

遍历每个所述命名空间并标记各所述命名空间的使用状态。

3. 根据权利要求1所述的命名空间分段存储优化方法,其特征在于,所述根据所述存储模块空闲空间信息对应的所述命名空间,创建得到指定分段以及对应的LBA范围,包括:

根据所述存储模块空闲空间信息计算每个所述存储模块的可用空间大小;

确定所述存储模块中包含的连续空闲空间对应的空间位置,以得到空闲空间信息;

根据预设的分段组合规则对所述命名空间进行分段组合,得到命名空间分段;

根据所述命名空间分段及预设的分配策略确定每个所述命名空间分段的分段起始LBA数量和分段连续LBA数量;

根据所述命名空间分段的分段起始LBA数量和分段连续LBA数量,创建对应的指定分段并确定所述指定分段的LBA范围。

4. 根据权利要求1所述的命名空间分段存储优化方法,其特征在于,所述将所述LBA范围所对应的内存空间分配至所述命名空间并更新所述指定分段,包括:

根据所述LBA范围确定需要分配的所述内存空间大小,所述内存空间与所述LBA范围相匹配;

更新所述指定分段;

根据所述内存空间大小及所述哈希表中记载的所述命名空间的起始LBA数量、连续LBA数量将对应的内存空间分配至所述命名空间。

5. 根据权利要求1所述的命名空间分段存储优化方法,其特征在于,所述根据获取到用户的读写请求信息对所述哈希表进行遍历得到相对LBA位置,基于所述相对LBA位置与所述读写请求信息进行数量比较,得到与所述读写请求信息相匹配的目标指定分段,包括:

根据所述读写请求信息中的命名空间标识符对所述哈希表进行定位；

获取所述哈希表中的相对LBA位置的位置信息,所述位置信息与所述命名空间内的存储段相关联；

检索所述相对LBA位置所关联的所述指定分段的起始LBA数量和连续LBA数量作为第一数量信息；

将所述第一数量信息与所述读写请求信息中的预设数量进行比较,以根据比较结果判断所述指定分段是否为待更新状态；

若所述指定分段为待更新状态,根据所述用户的读写请求信息更新所述指定分段的起始LBA数量和连续LBA数量,并将更新后的所述指定分段作为目标指定分段；

若所述指定分段不为待更新状态,将当前的所述指定分段作为目标指定分段。

6. 一种命名空间分段存储优化装置,其特征在于,所述装置用于执行如权利要求1-5任一项所述的命名空间分段存储优化方法,所述装置包括:

命名空间获取单元,用于根据获取到的表项空间分配信息对表项空间进行分配以得到多个命名空间；

第一信息生成单元,用于根据所述表项空间分配信息及各所述命名空间的状态标签生成对应的存储模块空闲空间信息并存储于预设的哈希表；

分段创建单元,用于根据所述存储模块空闲空间信息对应的所述命名空间,创建得到指定分段以及对应的LBA范围；

分段更新单元,用于将所述LBA范围所对应的内存空间分配至所述命名空间并更新所述指定分段；

目标指定分段获取单元,用于根据获取到用户的读写请求信息对所述哈希表进行遍历得到相对LBA位置,基于所述相对LBA位置与所述读写请求信息进行数量比较,得到与所述读写请求信息相匹配的目标指定分段。

7. 根据权利要求6所述的命名空间分段存储优化装置,其特征在于,所述装置还包括:

第一计算单元,用于根据每个命名空间所支持的最大分段数以及存储模块所支持的最大命名空间数计算得到每个所述命名空间所需的表项空间大小；

第二计算单元,用于将所有所述命名空间所需的所述表项空间大小累加得到总表项空间大小；

表项空间分配单元,用于根据所述总表项空间大小及各所述命名空间的表项空间大小对所述表项空间进行空间分配,以得到多个命名空间；

状态获取单元,用于遍历每个所述命名空间并标记各所述命名空间的使用状态。

8. 根据权利要求6所述的命名空间分段存储优化装置,其特征在于,所述装置还包括:

第三计算单元,用于根据所述存储模块空闲空间信息计算每个所述存储模块的可用空间大小；

第二信息生成单元,用于确定所述存储模块中包含的连续空闲空间对应的空间位置,以得到空闲空间信息；

分段获取单元,用于根据预设的分段组合规则对所述命名空间进行分段组合,得到命名空间分段；

LBA信息获取单元,用于根据所述命名空间分段及预设的分配策略确定每个所述命名

空间分段的分段起始LBA数量和分段连续LBA数量；

LBA范围确定单元,用于根据所述命名空间分段的分段起始LBA数量和分段连续LBA数量,创建对应的指定分段并确定所述指定分段的LBA范围。

9. 一种计算机设备,其特征在于,所述设备包括处理器、通信接口、存储器和通信总线,其中,处理器、通信接口、存储器通过通信总线完成相互间的通信;

存储器,用于存放计算机程序;

处理器,用于执行存储器上所存放的程序时,实现权利要求1-5中任一项所述的命名空间分段存储优化方法的步骤。

10. 一种计算机可读存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如权利要求1-5中任一项所述的命名空间分段存储优化方法的步骤。

## 命名空间分段存储优化方法、装置、计算机设备及介质

### 技术领域

[0001] 本发明涉及智能存储技术领域,尤其涉及一种命名空间分段存储优化方法、装置、计算机设备及介质。

### 背景技术

[0002] 在NVMe SSD(Non-Volatile Memory Express Solid State Drive,非易失性内存固态硬盘)领域,创建NS(Namespace,命名空间)时需要确定LBA(Logical Block Address,逻辑块地址)的数量,通常每个LBA表示512或4096字节。这涉及到管理文件或对象的元数据信息,其中表项空间分配和哈希表用于高效管理存储模块空闲空间。LBA的分配和更新是关键步骤,用于标识存储设备上的数据块。分段存储技术通过分段组合规则和分配策略对命名空间进行优化,确保有效利用存储空间。读写请求信息的处理涉及对用户请求的定位和相应分段信息的更新。然而,在实践中,随着Namespace的反复创建和删除,盘片剩余空间可能不再是连续的,而是离散的多段空闲空间,导致创建Namespace失败。为解决这一问题,一些厂商采用按照一定粒度(如256M、512M、1G等)管理空间,并使每个Namespace支持多个管理粒度整数倍的分段。尽管这种方案能够解决不连续空间的挑战,但仍存在单段空间至少是一个管理粒度大小的问题。此外,在用户创建Namespace时指定的LBA数不是管理粒度的整数倍时,可能造成空间浪费。因此,对于NVMe SSD领域,寻求更精细的管理策略以克服创建命名空间时处理不连续空间分布准确性不足的问题是至关重要的。

### 发明内容

[0003] 本申请实施例提供了一种命名空间分段存储优化方法、装置、计算机设备及介质,旨在解决固态硬盘应用过程中创建命名空间时,存在处理不连续空间分布准确性不足的问题。

[0004] 第一方面,本申请实施例提供了一种命名空间分段存储优化方法,其包括根据获取到的表项空间分配信息对表项空间进行分配以得到多个命名空间;表项空间分配信息包括每个命名空间支持的最大分段数量、存储模块支持的最大命名空间数以及分配结构数据;根据表项空间分配信息及各命名空间的状态标签生成对应的存储模块空闲空间信息并存储于预设的哈希表;根据存储模块空闲空间信息对命名空间,创建得到指定分段以及对应的LBA范围;将LBA范围所对应的内存空间分配至命名空间并更新指定分段;根据获取到用户的读写请求信息对哈希表进行遍历得到相对LBA位置,基于相对LBA位置与读写请求信息进行数量比较,得到与读写请求信息相匹配的目标指定分段。

[0005] 第二方面,本申请实施例还提供了一种命名空间分段存储优化装置,装置包括命名空间获取单元,用于根据获取到的表项空间分配信息对表项空间进行分配以得到多个命名空间;第一信息生成单元,用于根据表项空间分配信息及各命名空间的状态标签生成对应的存储模块空闲空间信息并存储于预设的哈希表;分段创建单元,用于根据存储模块空闲空间信息对应的命名空间,创建得到指定分段以及对应的LBA范围;分段更新单元,用于

将LBA范围所对应的内存空间分配至命名空间并更新指定分段;目标指定分段获取单元,用于根据获取到用户的读写请求信息对哈希表进行遍历得到相对LBA位置,基于相对LBA位置与读写请求信息进行数量比较,得到与读写请求信息相匹配的目标指定分段。

[0006] 第三方面,本申请实施例还提供了一种计算机设备,其包括存储器及处理器,存储器上存储有计算机程序,处理器执行计算机程序时实现上述方法。

[0007] 第四方面,本申请实施例还提供了一种计算机可读存储介质,存储介质存储有计算机程序,计算机程序包括程序指令,程序指令当被处理器执行时可实现上述方法。

[0008] 本申请实施例提供了一种命名空间分段存储优化方法、装置、计算机设备及存储介质。其中,方法包括根据获取到的表项空间分配信息对表项空间进行分配以得到多个命名空间;表项空间分配信息包括每个命名空间支持的最大分段数量、存储模块支持的最大命名空间数以及分配结构数据;根据表项空间分配信息及各命名空间的状态标签生成对应的存储模块空闲空间信息并存储于预设的哈希表;根据存储模块空闲空间信息对命名空间,创建得到指定分段以及对应的LBA范围;将LBA范围所对应的内存空间分配至命名空间并更新指定分段;根据获取到用户的读写请求信息对哈希表进行遍历得到相对LBA位置,基于相对LBA位置与读写请求信息进行数量比较,得到与读写请求信息相匹配的目标指定分段。

[0009] 本申请实施例由于采用了为了更清楚地说明本申请实施例技术方案,下面将对实施例描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图是本申请的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0010] 图1为本发明实施例提供的命名空间分段存储优化方法的方法流程图;

[0011] 图2为本发明实施例提供的命名空间分段存储优化方法的子步骤示意图;

[0012] 图3为本发明实施例提供的命名空间分段存储优化方法的另一子步骤示意图;

[0013] 图4为本发明实施例提供的命名空间分段存储优化方法的又一子步骤示意图;

[0014] 图5为本发明实施例提供的命名空间分段存储优化方法的再一子步骤示意图;

[0015] 图6为与本发明实施例提供的现有技术方法的应用效果图;

[0016] 图7为本发明实施例提供的命名空间分段存储优化方法的应用效果图;

[0017] 图8为本发明实施例提供的命名空间分段存储优化装置的示意性框图;

[0018] 图9为本发明实施例提供的计算机设备的示意性框图。

## 具体实施方式

[0019] 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,所描述的实施例是本申请一部分实施例,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,都属于本申请保护的范围。

[0020] 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。

[0021] 还应当理解,在此本申请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申请。如在本申请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。

[0022] 还应当进一步理解,在本申请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。

[0023] 请参阅图1,如图所示,本发明申请的实施例提供了一种命名空间分段存储优化方法,该命名空间分段存储优化方法应用于终端设备中,该方法通过安装于终端设备的应用软件进行执行,终端设备可以是台式电脑、笔记本电脑、平板电脑或手机等设备。如图1所示,该方法包括步骤S110~S150。

[0024] S110、根据获取到的表项空间分配信息对表项空间进行分配以得到多个命名空间;表项空间分配信息包括每个命名空间支持的最大分段数量、存储模块支持的最大命名空间数以及分配结构数据;

[0025] S120、根据表项空间分配信息及各命名空间的状态标签生成对应的存储模块空闲空间信息并存储于预设的哈希表;

[0026] S130、根据存储模块空闲空间信息对应的命名空间,创建得到指定分段以及对应的LBA范围;

[0027] S140、将LBA范围所对应的内存空间分配至命名空间并更新指定分段;

[0028] S150、根据获取到用户的读写请求信息对哈希表进行遍历得到相对LBA位置,基于相对LBA位置与读写请求信息进行数量比较,得到与读写请求信息相匹配的目标指定分段。

[0029] 在上述方法的具体实施例中,根据表项空间分配信息,实现对表项空间的有效分配,得到多个命名空间。使用获取到的表项空间分配信息,考虑每个命名空间支持的最大分段数量、存储模块支持的最大命名空间数以及分配结构数据。进行分段信息的合理分配,确保在系统中可以动态管理并满足不同命名空间的需求。根据表项空间分配信息和命名空间的状态,生成存储模块空闲空间信息,并存储于哈希表。结合表项空间分配信息和命名空间的状态标签,生成对应的存储模块空闲空间信息。利用哈希表实现对生成信息的存储,以便后续实时管理各命名空间的状态和可用空间。根据存储模块空闲空间信息,为指定命名空间创建分段及对应的LBA范围。利用存储模块空闲空间信息,确定特定命名空间的创建规则和分段组合方式。计算分段的起始LBA数量和连续LBA数量,以确保命名空间的有效利用。将LBA范围对应的内存空间分配至命名空间,并更新指定分段的信息。根据计算得到的LBA范围,进行内存空间的分配,确保与指定分段的对应关系。实时更新指定分段的信息,包括分段状态、使用情况等,以维护命名空间的动态状态。根据用户读写请求信息,通过哈希表快速定位目标指定分段,满足用户的读写需求。遍历哈希表,获取相对LBA位置,基于相对LBA位置与读写请求信息进行数量比较,找到与请求相匹配的目标指定分段。根据读写请求信息,判断是否需要更新指定分段的LBA信息,以满足用户请求的读写操作。

[0030] 在NVMe SSD中,由于NS空间在创建时需要通过指定LBA数量来确定空间大小,而传统的要求NS空间是连续的方式导致了盘片剩余空间可能是离散的多段,使得创建NS时容易面临连续空间不足的问题。当前厂商解决方案中,通过按照一定粒度管理空间,但存在单段空间至少是一个管理粒度大小的问题,同时可能会导致空间浪费。支持多段LBA范围配置,允许NS配置多段LBA范围,不再限制于连续空间。提供更灵活的空间管理,允许在非连续空间中创建NS,避免了传统方式中对连续空间的严格要求。不限制单段空间大小,摒弃了对单

段空间至少一个管理粒度大小的限制。允许更小的单段空间,有效减少了空间浪费,提高了存储资源的利用率。适应不等整数倍的LBA数,允许NS创建时指定的LBA数不是管理粒度的整数倍。避免了在用户创建NS时多分配空间造成的浪费,提高了空间利用的精细度。

[0031] 相较于当前厂商的实现方式,不限制单段空间大小,避免了传统方式中对于单段空间至少一个管理粒度大小的限制,减少了空间浪费。支持多段LBA范围配置,提供更灵活的空间管理,使得NS可以在非连续空间中创建,降低了因连续空间不足而导致创建NS失败的风险。适应不等整数倍的LBA数,允许用户创建NS时指定的LBA数不是管理粒度的整数倍,减少了多余空间的分配,提高了空间利用效率。

[0032] 在具体的实施过程中,如图6-7所示,图6为现有技术方法的实施过程示意图,图7为命名空间分段存储优化方法的应用效果图,通过图6及图7的技术效果对比,上述方法在实际SSD设备上的应用产生了显著的技术效果。

[0033] 如图6,总共 $6n$ 个LBA的空间被分为5个 $n$ 个LBA的NS,删除NS2和NS4后,空闲空间为 $3n$ 个LBA,但不连续。利用现有技术方法进行创建时,由于NS不支持多段空间,所以无法创建一个连续的 $3n$ 个LBA的NS。

[0034] 因此,根据本发明实施例所公开的方法,如图7所示,以每个NS支持5段空间为例,删除NS2和NS4后,空闲空间拥有 $3n$ 个LBA。此时,可以按照以下步骤重新创建一个 $3n$ 个LBA的NS空间:首先,将空闲空间划分为5个段,每个段包含 $n$ 个LBA。这样,每个段都是一个连续的 $n$ 个LBA的空间。从剩下的三段空间中,选取任意一段作为新的NS的第一个段。将该段空间分配给新NS。从剩下的四段空间中,选取两段分别作为新NS的第二和第三个段。将这两段空间分配给新NS。最后,将剩下的两段空间合并,作为新NS的第四个段。通过以上步骤成功创建了一个连续的 $3n$ 个LBA的NS空间。在这种情况下,NS支持多分段空间的能力可以灵活地调整和重新分配空闲空间,从而实现更高效的空间利用。

[0035] 通过支持最多128个命名空间和每个命名空间最多64个分段的配置,该方法提供了对存储资源更灵活的管理。这使得系统管理员可以根据实际需求动态配置和调整命名空间和分段,以满足不同应用场景的需求。通过对命名空间的随机大小进行创建和删除的方式进行测试,该方法展现了在动态环境下对存储资源进行高效管理的能力。这对于应对不断变化的存储需求,特别是在虚拟化和云计算环境下,具有重要意义。该方法成功解决了前述场景中提到的问题,如表项空间的合理分配、命名空间的动态管理、分段的创建与更新,以及对读写请求的高效响应。这表明该技术方案在实际应用中确实能够应对复杂的存储管理问题。通过实际测试,该方法对SSD设备的性能进行了优化。通过合理的分段创建和LBA范围分配,有效减少了对存储资源的浪费,并提高了存储系统的整体性能。通过在实际设备上的测试,该方法展现了对存储系统的稳定性和可靠性的良好支持。这对于长时间运行和数据持久性要求高的应用场景非常重要。在实际SSD设备上的应用表现出了对复杂存储管理问题的有效解决能力,为存储系统的灵活性、性能和可靠性提供了显著的提升。这些效果使得该技术方案在实际应用中具有实用性和商业价值。

[0036] 综上,上述方法解决了NVMe SSD中命名空间(NS)创建过程中的表项空间的动态分配、存储模块空闲空间信息的高效管理、命名空间分段和LBA范围的连续创建、内存空间分配与分段更新、以及用户读写请求的快速处理等技术问题,提高了空间分配的灵活性,优化了存储模块信息管理,确保了命名空间的连续性,实现了实时的存储信息更新,并大幅提升



了用户读写请求的处理效率。整体上技术方法为NVMe SSD的性能和效率提供了全面的优化方案。

[0037] 更进一步地,如图2所示,执行步骤S110还具体包括执行步骤S111-S114:

[0038] S111、根据每个命名空间所支持的最大分段数以及存储模块所支持的最大命名空间数计算得到每个命名空间所需的表项空间大小;

[0039] S112、将所有命名空间所需的表项空间大小累加得到总表项空间大小;

[0040] S113、根据总表项空间大小及各命名空间的表项空间大小对表项空间进行空间分配,以得到多个命名空间;

[0041] S114、遍历每个命名空间并标记各命名空间的使用状态。

[0042] 具体地,在进一步执行步骤S110的细分步骤(S111-S114)中,成功解决了NVMe SSD中命名空间(NS)创建过程中的关键问题。首先,在S111中,通过考虑每个命名空间支持的最大分段数和存储模块支持的最大命名空间数,计算了每个命名空间所需的表项空间大小。接着,在S112中,通过累加计算得到了总表项空间大小,确保足够容纳所有命名空间的信息。随后,在S113中,进行了有效的表项空间分配,以确保每个命名空间都有足够的空间存储相关信息,实现了动态分配表项空间。最后,在S114中,通过遍历每个命名空间并标记其使用状态,为后续步骤提供了有效的管理和分配存储资源的信息。这一系列细分步骤有力地解决了表项空间的动态分配问题,为命名空间的创建提供了可靠的基础。

[0043] 更进一步地,如图3所示,执行步骤S130还具体包括执行步骤S131-S135:

[0044] S131、根据存储模块空闲空间信息计算每个存储模块的可用空间大小;

[0045] S132、确定存储模块中包含的连续空闲空间对应的空间位置,以得到空闲空间信息;

[0046] S133、根据预设的分段组合规则对命名空间进行分段组合,得到命名空间分段;

[0047] S134、根据命名空间分段及预设的分配策略确定每个命名空间分段的分段起始LBA数量和分段连续LBA数量;

[0048] S135、根据命名空间分段的分段起始LBA数量和分段连续LBA数量,创建对应的指定分段并确定指定分段的LBA范围。

[0049] 具体地,在进一步的执行步骤S130中,具体包括执行步骤S131-S135,为了更全面地优化NVMe SSD中的命名空间管理,进一步解决了存储模块空间和命名空间分段的关键问题。

[0050] 通过分析存储模块的空闲空间信息,S131计算了每个存储模块的可用空间大小。这一步骤为后续命名空间的合理分配提供了基础,确保了存储资源的有效利用。S132确定存储模块中包含的连续空闲空间对应的位置,以获得详细的空闲空间信息。这个步骤为存储模块的空间规划提供了准确的空间位置信息,以支持后续的命名空间分段组合。基于预设的分段组合规则,S133对命名空间进行分段组合,得到具体的命名空间分段。这一步骤确保了对命名空间的灵活管理,以满足不同应用场景的需求。S134根据预设的分配策略,确定了每个命名空间分段的起始LBA数量和连续LBA数量。这确保了对命名空间分段的有效分配,使得存储模块能够高效地存储和检索数据。最后,在S135中,根据命名空间分段的起始LBA数量和连续LBA数量,创建了对应的指定分段,并确定了该分段的LBA范围。这一步骤为系统提供了明确的分段信息,以支持命名空间的精准管理和存储操作。通过S130及其细分

步骤,系统进一步完善了对NVMe SSD中存储模块和命名空间的管理,实现了对存储资源的高效分配和利用。

[0051] 更进一步地,如图4所示,执行步骤S140还具体包括执行步骤S141-S143:

[0052] S141、根据LBA范围确定需要分配的内存空间大小,内存空间与LBA范围相匹配;

[0053] S142、更新指定分段;

[0054] S143、根据内存空间大小及哈希表中记载的命名空间的起始LBA数量、连续LBA数量将对应的内存空间分配至命名空间。

[0055] 具体地,通过根据LBA范围确定需要分配的内存空间大小,确保内存空间与LBA范围相匹配。这一步骤保证了内存分配的准确性,使得内存空间与相应命名空间的存储需求一致。在S142中,对指定的分段进行更新。这一步骤确保了系统能够实时地反映命名空间分段的变化,保持存储模块的实时状态。S143根据内存空间大小及哈希表中记录的命名空间的起始LBA数量和连续LBA数量,将相应的内存空间分配至命名空间。这一关键步骤实现了内存空间的精准分配,确保了每个命名空间都能够获得其所需的存储资源。通过S140及其细分步骤,系统更进一步优化了NVMe SSD中的存储资源管理和命名空间分配策略,实现了对存储资源的高效利用和命名空间的动态管理。

[0056] 更进一步地,如图5所示,执行步骤S150还具体包括执行步骤S151-S154:

[0057] S151、根据读写请求信息中的命名空间标识符对哈希表进行定位;

[0058] S152、获取哈希表中的相对LBA位置的位置信息;位置信息与命名空间内的存储段相关联。

[0059] S153、检索相对LBA位置所关联的指定分段的起始LBA数量和连续LBA数量作为第一数量信息;

[0060] S154、将第一数量信息与读写请求信息中的预设数量进行比较,以根据比较结果判断指定分段是否为待更新状态;

[0061] 具体地,若指定分段为待更新状态,则执行步骤S1541、根据用户的读写请求信息更新指定分段的起始LBA数量和连续LBA数量,并将更新后的指定分段作为目标指定分段;

[0062] 若指定分段不为待更新状态,将当前的指定分段作为目标指定分段。

[0063] 在进一步的执行步骤S150中,系统不仅实现了命名空间的动态更新和读写请求的高效处理,还加入了一些优化措施以提升系统性能和响应速度:在S154后,系统可能采取一系列优化措施,如预读数据、写缓存策略等,以优化读写请求的处理过程。这确保了系统能够更加高效地响应用户的读写操作,提升存储性能。在S150执行的过程中,系统可能实施错误处理和回滚机制,以应对意外错误或异常情况。这包括在更新指定分段时的数据完整性检查和错误修复,确保系统的稳定性和可靠性。为了跟踪系统状态和进行故障排除,S150涉及记录操作日志,包括指定分段的更新记录、存储模块的空间使用情况等。这为系统维护和管理提供了重要的信息。通过引入这些优化和安全措施,S150不仅实现了基本的动态更新和高效读写请求处理,还进一步提升了系统的可靠性、性能和可维护性。

[0064] 综上,本案所公开的技术方案的关键目标是让NS支持配置多段LBA范围,使其能够处理不连续的空间,同时以较小的代价避免剩余空间不连续导致创建NS失败的问题。允许每个NS配置多个不连续的LBA范围,以适应各种应用场景。提供了灵活的配置选项,使用户能够根据实际需求灵活定义每个NS的空间分布。通过对NS的分段表按全局LBA的顺序从小

到大排列,实现了有效的不连续空间处理。在IO读写处理中,通过遍历分段信息表,正确计算相对LBA位置,处理了跨越多个分段的空间访问。动态空间分配方法充分考虑SSD的空闲空间情况,通过算法自动匹配合适的分段组合成NS的空间。在分配空间时,尽量减少对SSD空间的碎片化,以降低创建NS失败的概率。与常见厂商的实现方式相比,该方案不限制单段空间的大小,更加灵活适应各种应用场景。通过动态空间分配方法,避免了不必要的空间浪费,确保SSD空间的高效利用。方案经过在实际SSD设备上的测试,支持最多128个NS,每个NS最多支持64分段,并通过随机大小的NS创建和删除方式进行测试,证明了在实际应用中的有效性和可行性。综合来看,该技术方案在提高NS空间利用率、处理不连续空间、避免创建NS失败等方面取得了显著的技术效果。

[0065] 本发明实施例还提供一种命名空间分段存储优化装置,该装置可配置于终端设备中,该命名空间分段存储优化装置用于执行前述的命名空间分段存储优化方法的任一实施例。具体地,请参阅图8,图8为本发明实施例提供的命名空间分段存储优化装置的示意性框图。

[0066] 如图8所示,命名空间分段存储优化装置100包括命名空间获取单元110,用于根据获取到的表项空间分配信息对表项空间进行分配以得到多个命名空间;第一信息生成单元120,用于根据表项空间分配信息及各命名空间的状态标签生成对应的存储模块空闲空间信息并存储于预设的哈希表;分段创建单元130,用于根据存储模块空闲空间信息对应的命名空间,创建得到指定分段以及对应的LBA范围;分段更新单元140,用于将LBA范围所对应的内存空间分配至命名空间并更新指定分段;目标指定分段获取单元150,用于根据获取到用户的读写请求信息对哈希表进行遍历得到相对LBA位置,基于相对LBA位置与读写请求信息进行数量比较,得到与读写请求信息相匹配的目标指定分段。进一步的,该装置还具体包括第一计算单元,用于根据每个命名空间所支持的最大分段数以及存储模块所支持的最大命名空间数计算得到每个命名空间所需的表项空间大小;第二计算单元,用于将所有命名空间所需的表项空间大小累加得到总表项空间大小;表项空间分配单元,用于根据总表项空间大小及各命名空间的表项空间大小对表项空间进行空间分配,以得到多个命名空间;状态获取单元,用于遍历每个命名空间并标记各命名空间的使用状态;第三计算单元,用于根据存储模块空闲空间信息计算每个存储模块的可用空间大小;第二信息生成单元,用于确定存储模块中包含的连续空闲空间对应的空间位置,以得到空闲空间信息;分段获取单元,用于根据预设的分段组合规则对命名空间进行分段组合,得到命名空间分段;LBA信息获取单元,用于根据命名空间分段及预设的分配策略确定每个命名空间分段的分段起始LBA数量和分段连续LBA数量;LBA范围确定单元,用于根据命名空间分段的分段起始LBA数量和分段连续LBA数量,创建对应的指定分段并确定指定分段的LBA范围。上述单元均用于执行命名空间分段存储优化方法中的对应步骤。

[0067] 在本发明实施例所提供的命名空间分段存储优化装置应用上述方法,以实现NVMe SSD中命名空间(NS)创建过程中的表项空间的动态分配、存储模块空闲空间信息的高效管理、命名空间分段和LBA范围的连续创建、内存空间分配与分段更新、以及用户读写请求的快速处理,提高了空间分配的灵活性,优化了存储模块信息管理,确保了命名空间的连续性,实现了实时的存储信息更新,并大幅提升了用户读写请求的处理效率。

[0068] 上述命名空间分段存储优化装置可以实现为计算机程序的形式,该计算机程序可

以在如图9所示的计算机设备上运行。

[0069] 请参阅图9,图9是本发明实施例提供的计算机设备的示意性框图。该计算机设备可以是用于执行命名空间分段存储优化方法的终端设备。

[0070] 参阅图9,该计算机设备500包括通过通信总线501连接的处理器502、存储器和网络接口505,其中,存储器可以包括存储介质503和内存存储器504。

[0071] 该存储介质503可存储操作系统5031和计算机程序5032。该计算机程序5032被执行时,可使得处理器502执行命名空间分段存储优化方法,其中,存储介质503可以为易失性的存储介质或非易失性的存储介质。

[0072] 该处理器502用于提供计算和控制能力,支撑整个计算机设备500的运行。

[0073] 该内存存储器504为存储介质503中的计算机程序5032的运行提供环境,该计算机程序5032被处理器502执行时,可使得处理器502执行命名空间分段存储优化方法。

[0074] 该网络接口505用于进行网络通信,如提供数据信息的传输等。本领域技术人员可以理解,图9中示出的结构,仅仅是与本发明方案相关的部分结构的框图,并不构成对本发明方案所应用于其上的计算机设备500的限定,具体的计算机设备500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。

[0075] 其中,处理器502用于运行存储在存储器中的计算机程序5032,以实现上述的命名空间分段存储优化方法中对应的功能。

[0076] 本领域技术人员可以理解,图9中示出的计算机设备的实施例并不构成对计算机设备具体构成的限定,在其他实施例中,计算机设备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。例如,在一些实施例中,计算机设备可以仅包括存储器及处理器,在这样的实施例中,存储器及处理器的结构及功能与图9所示实施例一致,在此不再赘述。

[0077] 应当理解,在本发明实施例中,处理器502可以是中央处理单元(Central Processing Unit,CPU),该处理器502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。

[0078] 在本发明的另一实施例中提供计算机可读存储介质。该计算机可读存储介质可以为易失性或非易失性的计算机可读存储介质。该计算机可读存储介质存储有计算机程序,其中计算机程序被处理器执行时实现上述的命名空间分段存储优化方法中所包含的步骤。

[0079] 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,上述描述的设备、装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。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、计算机软件或者二者的结合来实现,为了清楚地说明硬件和软件的可互换性,在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。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。

[0080] 在本发明所提供的几个实施例中,应该理解到,所揭露的设备、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,单元的划分,仅仅为逻辑功能划分,实际实现时可以有另外的划分方式,也可以将具有相同功能的单元集成成一个单元,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。另外,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通过一些接口、装置或单元的间接耦合或通信连接,也可以是电的,机械的或其它的形式连接。

[0081] 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发明实施例方案的目的。

[0082] 另外,在本发明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,也可以是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。

[0083] 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存储介质中。基于这样的理解,本发明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计算机可读存储介质中,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)执行本发明各个实施例方法的全部或部分步骤。而前述的计算机可读存储介质包括:U盘、移动硬盘、只读存储器(ROM, Read-Only Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0084] 以上,仅为本发明的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到各种等效的修改或替换,这些修改或替换都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应以权利要求的保护范围为准。

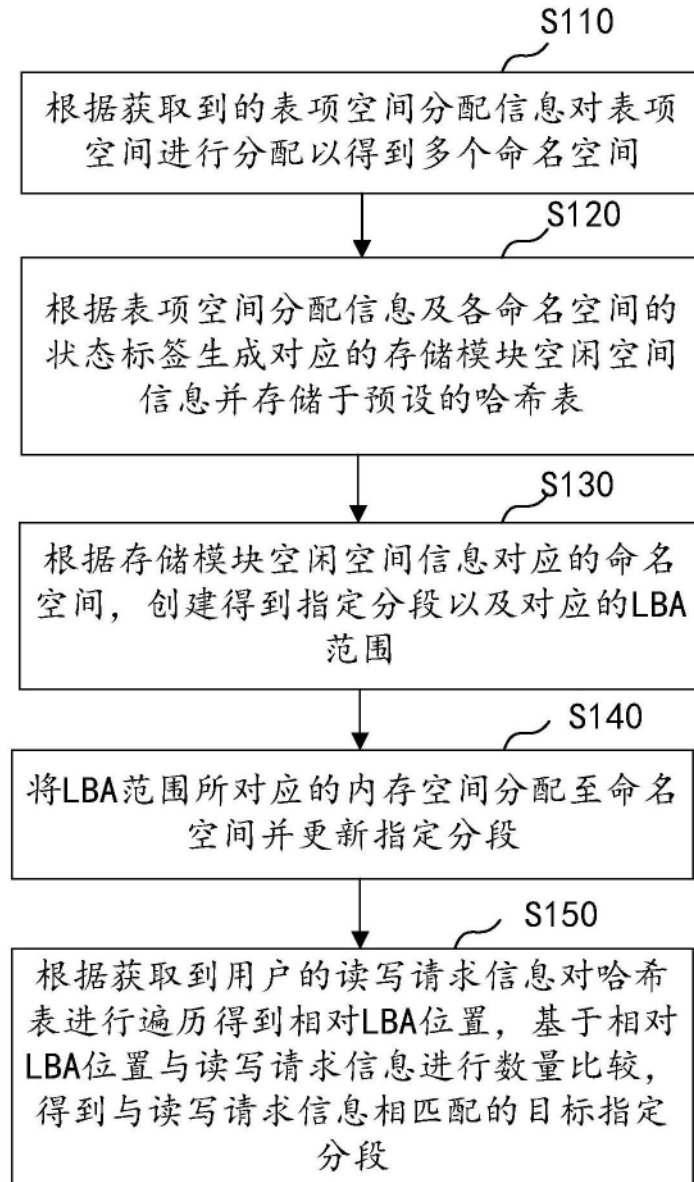


图1

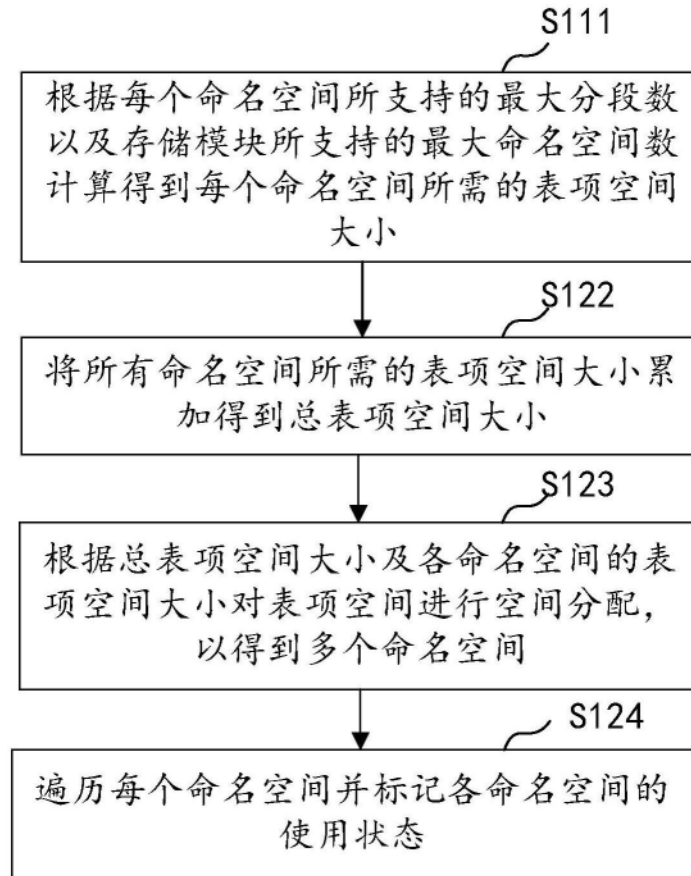


图2

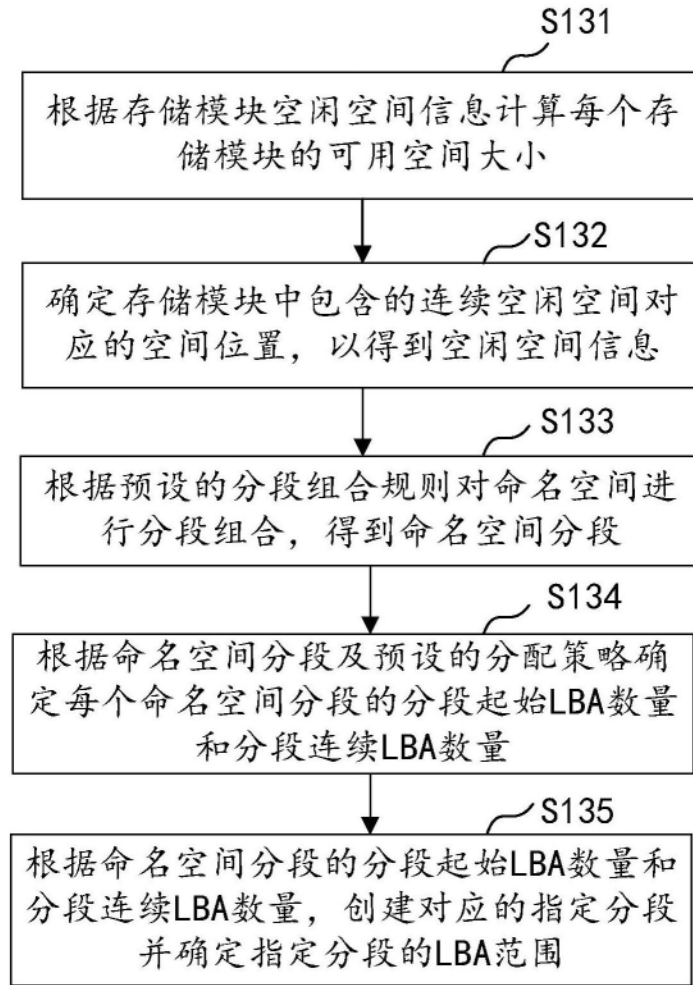


图3

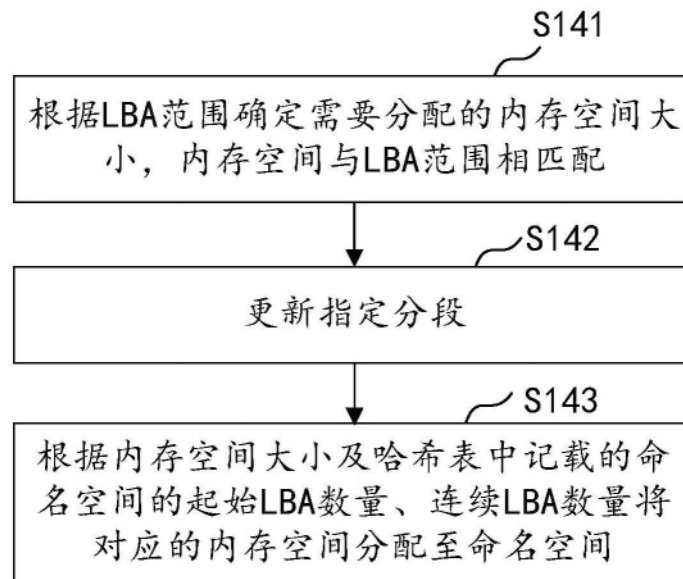


图4



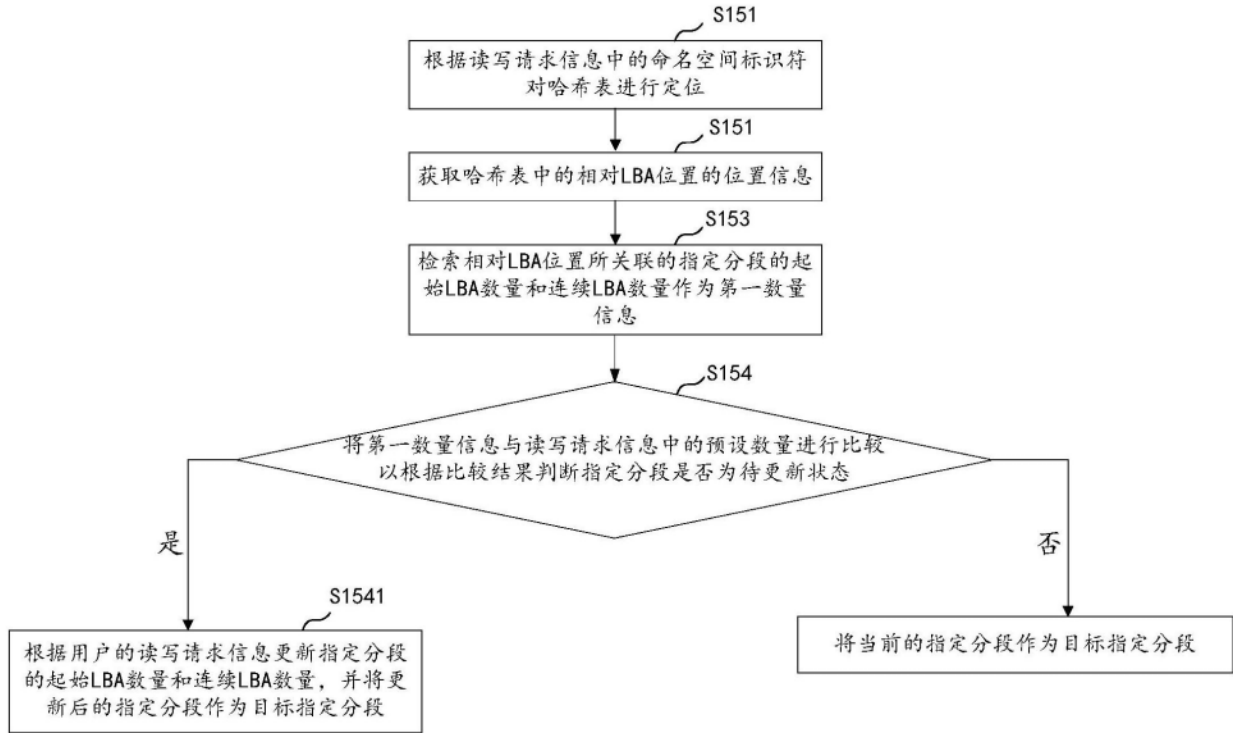


图5

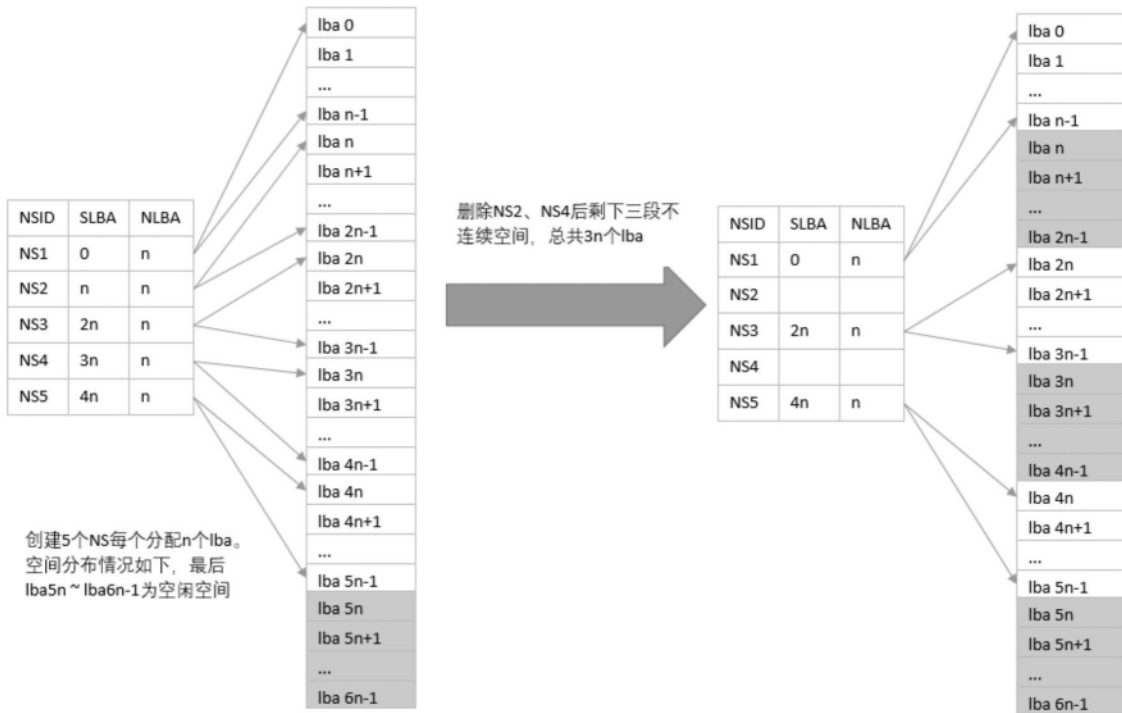


图6

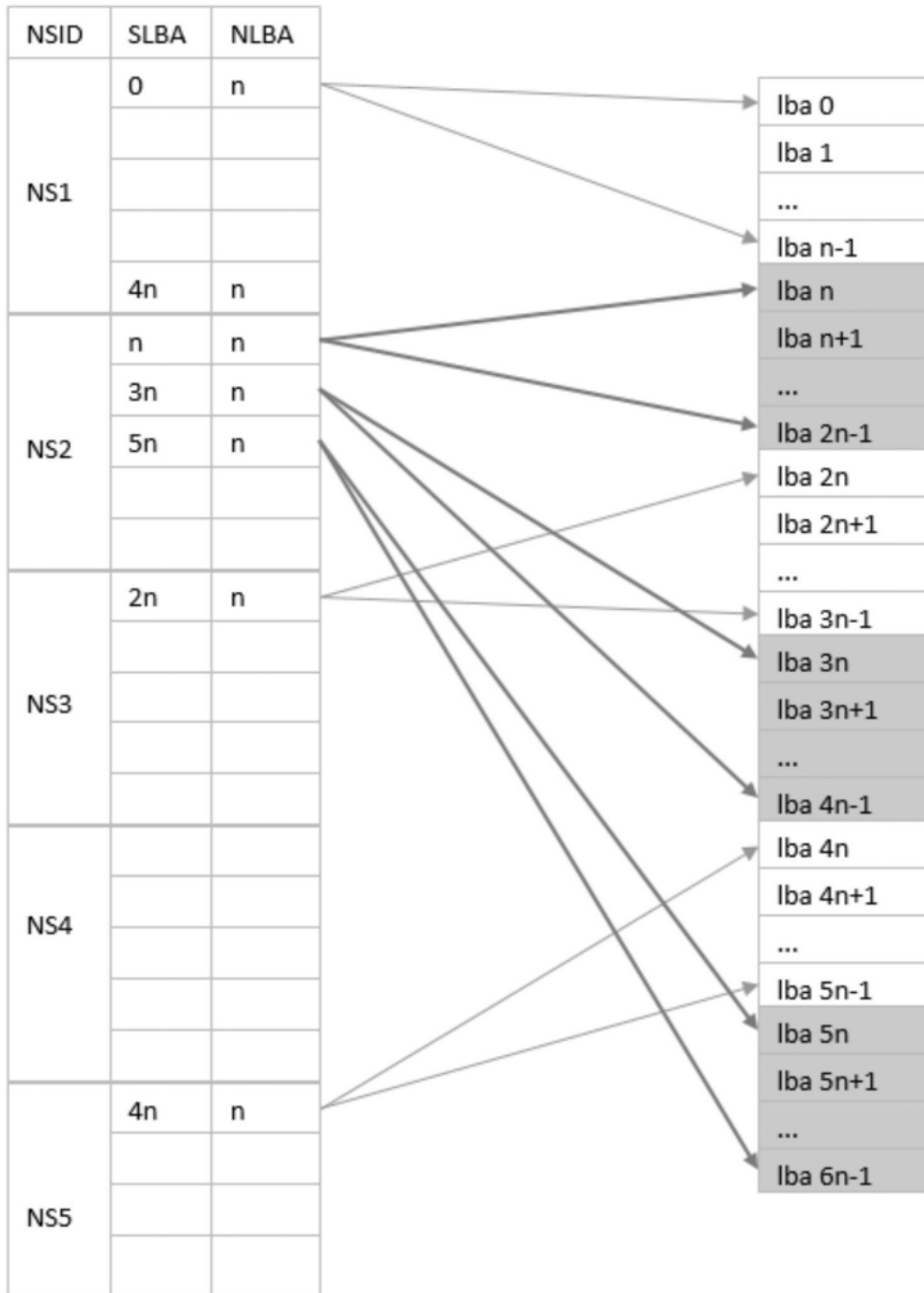


图7

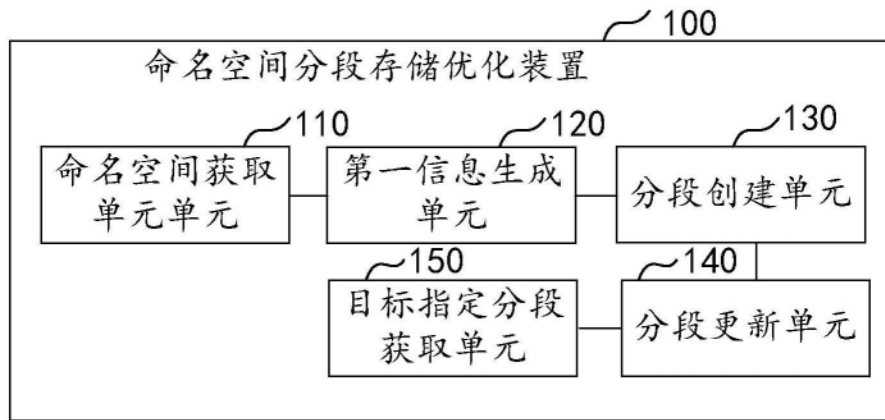


图8

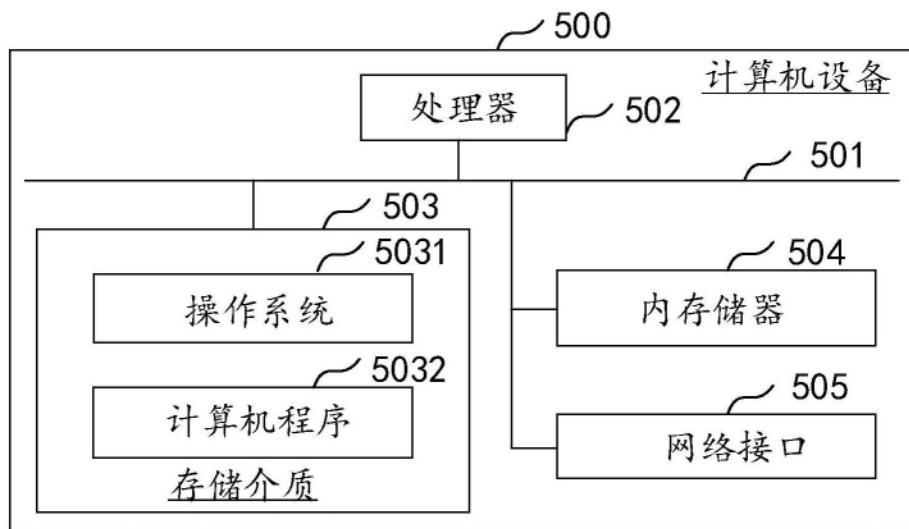


图9