



(12) 发明专利申请

(10) 申请公布号 CN 111880750 A

(43) 申请公布日 2020. 11. 03

(21) 申请号 202010814848.3

(22) 申请日 2020.08.13

(71) 申请人 腾讯科技(深圳)有限公司

地址 518000 广东省深圳市南山区高新区
科技中一路腾讯大厦35层

(72) 发明人 张袁

(74) 专利代理机构 北京派特恩知识产权代理有限公司 11270

代理人 赵翠萍 张颖玲

(51) Int.Cl.

G06F 3/06 (2006.01)

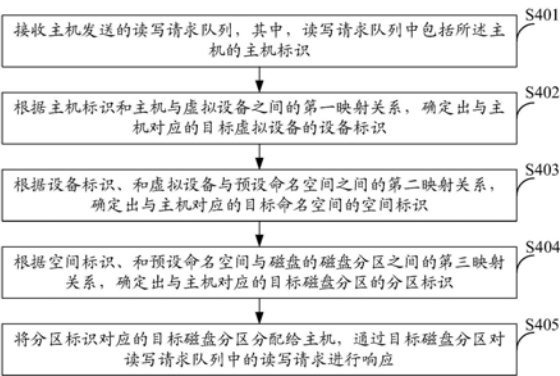
权利要求书3页 说明书19页 附图9页

(54) 发明名称

磁盘读写资源的分配方法、装置、设备及存储介质

(57) 摘要

本申请实施例提供一种磁盘读写资源的分配方法、装置、设备及计算机可读存储介质,涉及云技术领域。该方法包括:依次根据主机标识和主机与虚拟设备之间的第一映射关系、所述虚拟设备与预设命名空间之间的第二映射关系、所述预设命名空间与磁盘的磁盘分区之间的第三映射关系,确定出与所述主机对应的目标磁盘分区的分区标识;将所述分区标识对应的目标磁盘分区分配给所述主机,通过所述目标磁盘分区对所述读写请求队列中的读写请求进行响应。通过本申请,能够实现对读写请求队列所分配资源的动态划分,进而实现了各个预设命名空间的读写负载均衡。



1. 一种磁盘读写资源的分配方法,其特征在于,包括:

接收主机发送的读写请求队列,其中,所述读写请求队列中包括所述主机的主机标识;
根据所述主机标识和所述主机与虚拟设备之间的第一映射关系,确定出与所述主机对应的目标虚拟设备的设备标识;

根据所述设备标识、和所述虚拟设备与预设命名空间之间的第二映射关系,确定出与所述主机对应的目标命名空间的空间标识;

根据所述空间标识、和所述预设命名空间与磁盘的磁盘分区之间的第三映射关系,确定出与所述主机对应的目标磁盘分区的分区标识;

将所述分区标识对应的目标磁盘分区分配给所述主机,通过所述目标磁盘分区对所述读写请求队列中的读写请求进行响应。

2. 根据权利要求1所述的方法,其特征在于,所述接收主机发送的读写请求队列,包括:

接收主机集群中任一主机发送的至少一条读写请求;

将所述至少一条读写请求写入至预设的提交队列或完成队列中,形成所述读写请求队列。

3. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

在接收主机发送的读写请求队列之前,采用至少一个所述预设命名空间,对待池化的磁盘阵列中的每一所述磁盘的磁盘资源进行资源隔离处理,得到隔离后的多个磁盘分区,其中,每一磁盘分区与一个所述预设命名空间之间具有所述第三映射关系;

将具有所述第三映射关系的所述磁盘分区和所述预设命名空间,与虚拟设备集合中的一个虚拟设备进行资源映射处理,以得到所述虚拟设备与所述磁盘分区和所述预设命名空间之间的所述第二映射关系。

4. 根据权利要求3所述的方法,其特征在于,所述方法还包括:

预先设置主机集群中的每一所述主机,与所述虚拟设备集合中的至少一个虚拟设备之间的所述第一映射关系。

5. 根据权利要求3所述的方法,其特征在于,所述采用至少一个所述预设命名空间,对待池化的磁盘阵列中的每一所述磁盘的磁盘资源进行资源隔离处理,得到隔离后的多个磁盘分区,包括:

采用每一所述预设命名空间,对每一所述磁盘中的连续地址对应的连续磁盘资源进行资源隔离处理,得到隔离后的多个磁盘分区;

其中,每一所述磁盘分区的磁盘资源对应连续的地址,且每一所述磁盘分区的容量与其他磁盘分区的容量相同或不同。

6. 根据权利要求3所述的方法,其特征在于,在所述磁盘阵列中的多个磁盘中,每一所述预设命名空间与每一所述磁盘中的一个磁盘分区之间具有所述第三映射关系;所述方法还包括:

对于每一预设命名空间对应的任一磁盘的磁盘分区,根据所述磁盘分区的读写响应速度或所述磁盘分区的容量,确定出所述磁盘分区的第一优先级;

获取所述主机发送的所述读写请求队列的第二优先级;

根据所述第二优先级和每一所述磁盘分区的第一优先级,在所述多个磁盘中匹配出一个磁盘,并将所匹配出的磁盘中的所述磁盘分区确定为所述目标磁盘分区。

7. 根据权利要求3所述的方法, 其特征在于, 在所述磁盘阵列中的多个磁盘中, 每一所述预设命名空间与每一所述磁盘中的一个磁盘分区之间具有所述第三映射关系; 所述方法还包括:

为所述主机添加初始化标签, 所述初始化标签用于标识所述主机的读写请求队列对应的所述目标磁盘分区的目标容量;

根据所述目标容量, 在所述多个磁盘中匹配出一个或多个磁盘;

将匹配出的一个或多个磁盘中的所述磁盘分区, 确定为所述目标磁盘分区。

8. 根据权利要求1至7任一项所述的方法, 其特征在于, 所述根据所述空间标识、和所述预设命名空间与磁盘的磁盘分区之间的第三映射关系, 确定出与所述主机对应的目标磁盘分区的分区标识, 包括:

确定所述目标虚拟设备的页面偏移和所述空间标识对应的所述目标命名空间的地址;

将所述页面偏移对应的偏移值, 确定为所述目标虚拟设备的地址相对于所述目标命名空间的地址的偏移量;

将相对于所述目标命名空间的地址, 偏移所述偏移量之后的偏移地址, 确定为与所述主机对应的目标磁盘分区的地址;

根据所述目标磁盘分区的地址, 确定所述目标磁盘分区的分区标识。

9. 根据权利要求1至7任一项所述的方法, 其特征在于, 所述方法还包括: 确定所述主机的当前的读写请求队列对应的资源分配权重;

所述根据所述空间标识、和所述预设命名空间与磁盘的磁盘分区之间的第三映射关系, 确定出与所述主机对应的目标磁盘分区的分区标识, 包括:

根据所述空间标识和所述资源分配权重, 在磁盘阵列中的多个磁盘中, 确定出一个或多个磁盘为目标磁盘;

在所匹配出的目标磁盘中, 将与第三映射关系对应的磁盘分区确定为所述目标磁盘分区。

10. 根据权利要求9所述的方法, 其特征在于, 所述确定所述主机的当前的读写请求队列对应的资源分配权重, 包括:

在对所述主机的资源分配权重进行调整之后, 确定所述主机的当前读写请求量和所述目标虚拟设备中处于等待状态的待处理读写请求量;

根据所述当前读写请求量和所述待处理读写请求量, 确定所述主机的当前的读写请求队列对应的资源分配权重。

11. 根据权利要求9所述的方法, 其特征在于, 所述确定所述主机的当前的读写请求队列对应的资源分配权重, 包括:

确定历史周期内对所述读写请求进行响应的延迟时间的第一均值、和当前周期内对所述读写请求进行响应的延迟时间的第二均值;

如果所述第二均值大于或等于所述第一均值, 增大所述主机的当前的读写请求队列对应的资源分配权重;

如果所述第二均值小于所述第一均值, 减小所述主机的当前的读写请求队列对应的资源分配权重。

12. 根据权利要求9所述的方法, 其特征在于, 所述确定所述主机的当前的读写请求队

列对应的资源分配权重,包括:

确定所述主机的读写请求队列的延迟敏感类型;

如果所述读写请求队列为尾部延迟敏感型队列,增大所述主机的当前的读写请求队列对应的资源分配权重,且增大后的所述资源分配权重是当前的磁盘读写资源的分配系统中全部读写请求队列中的最大权重。

13. 一种磁盘读写资源的分配装置,其特征在于,包括:

接收模块,用于接收主机发送的读写请求队列,其中,所述读写请求队列中包括所述主机的主机标识;

第一确定模块,用于根据所述主机标识和所述主机与虚拟设备之间的第一映射关系,确定出与所述主机对应的目标虚拟设备的设备标识;

第二确定模块,用于根据所述设备标识、和所述虚拟设备与预设命名空间之间的第二映射关系,确定出与所述主机对应的目标命名空间的空间标识;

第三确定模块,用于根据所述空间标识、和所述预设命名空间与磁盘的磁盘分区之间的第三映射关系,确定出与所述主机对应的目标磁盘分区的分区标识;

分配模块,用于将所述分区标识对应的目标磁盘分区分配给所述主机,通过所述目标磁盘分区对所述读写请求队列中的读写请求进行响应。

14. 一种磁盘读写资源的分配设备,其特征在于,包括:

存储器,用于存储可执行指令;处理器,用于执行所述存储器中存储的可执行指令时,实现权利要求1至12任一项所述的磁盘读写资源的分配方法。

15. 一种计算机可读存储介质,其特征在于,存储有可执行指令,用于引起处理器执行所述可执行指令时,实现权利要求1至12任一项所述的磁盘读写资源的分配方法。

磁盘读写资源的分配方法、装置、设备及存储介质

技术领域

[0001] 本申请实施例涉及计算机领域,涉及但不限于一种磁盘读写资源的分配方法、装置、设备及计算机可读存储介质。

背景技术

[0002] 随着闪存技术的逐渐普及,非易失性内存主机控制器接口规范(NVMe,Non-Volatile Memory Express)技术也随之变成了支持闪存设备与主机(Host)高速访问的关键通道。目前,对于NVMe固态硬盘(SSD,Solid State Disk)的应用,更多的是基于高速独占式的需求,由特定Host实现极致的性能增长。

[0003] 但是,随着NVMe SSD容量的变大,单盘的故障范围及利用率已经无法满足要求,需要考虑通过容量拆分的方式实现资源的复用及共享,然而相关技术中并没有针对NVMe SSD的约束队列管理及分配机制,无法实现对NVMe SSD进行有效的容量拆分以实现资源复用和共享,因此无法实现对NVMe SSD的负载均衡。

发明内容

[0004] 本申请实施例提供一种磁盘读写资源的分配方法、装置、设备及计算机可读存储介质,涉及云技术领域。依次基于主机与虚拟设备之间的第一映射关系、虚拟设备与预设命名空间之间的第二映射关系和预设命名空间与磁盘的磁盘分区之间的第三映射关系,确定出主机对应的目标磁盘分区,从而将目标磁盘分区的资源分配给主机的读写请求队列,实现对磁盘资源的动态划分,实现了各个预设命名空间的读写负载均衡。

[0005] 本申请实施例的技术方案是这样实现的:

[0006] 本申请实施例提供一种磁盘读写资源的分配方法,包括:

[0007] 接收主机发送的读写请求队列,其中,所述读写请求队列中包括所述主机的主机标识;

[0008] 根据所述主机标识和所述主机与虚拟设备之间的第一映射关系,确定出与所述主机对应的目标虚拟设备的设备标识;

[0009] 根据所述设备标识、和所述虚拟设备与预设命名空间之间的第二映射关系,确定出与所述主机对应的目标命名空间的命名空间标识;

[0010] 根据所述命名空间标识、和所述预设命名空间与磁盘的磁盘分区之间的第三映射关系,确定出与所述主机对应的目标磁盘分区的分区标识;

[0011] 将所述分区标识对应的目标磁盘分区分配给所述主机,通过所述目标磁盘分区对所述读写请求队列中的读写请求进行响应。

[0012] 本申请实施例提供一种磁盘读写资源的分配装置,包括:

[0013] 接收模块,用于接收主机发送的读写请求队列,其中,所述读写请求队列中包括所述主机的主机标识;

[0014] 第一确定模块,用于根据所述主机标识和所述主机与虚拟设备之间的第一映射关

系,确定出与所述主机对应的目标虚拟设备的设备标识;

[0015] 第二确定模块,用于根据所述设备标识、和所述虚拟设备与预设命名空间之间的第二映射关系,确定出与所述主机对应的目标命名空间的空间标识;

[0016] 第三确定模块,用于根据所述空间标识、和所述预设命名空间与磁盘的磁盘分区之间的第三映射关系,确定出与所述主机对应的目标磁盘分区的分区标识;

[0017] 分配模块,用于将所述分区标识对应的目标磁盘分区分配给所述主机,通过所述目标磁盘分区对所述读写请求队列中的读写请求进行响应。

[0018] 本申请实施例提供一种页面显示设备,包括:

[0019] 存储器,用于存储可执行指令;处理器,用于执行所述存储器中存储的可执行指令时,实现上述的磁盘读写资源的分配方法。

[0020] 本申请实施例提供一种计算机可读存储介质,存储有可执行指令,用于引起处理器执行所述可执行指令时,实现上述的磁盘读写资源的分配方法。

[0021] 本申请实施例具有以下有益效果:对于接收到的主机发送的读写请求队列,根据主机的主机标识,依次基于主机与虚拟设备之间的第一映射关系、虚拟设备与预设命名空间之间的第二映射关系和预设命名空间与磁盘的磁盘分区之间的第三映射关系,确定出主机对应的目标磁盘分区,从而将目标磁盘分区的资源分配给主机的读写请求队列,通过所述目标磁盘分区对所述读写请求队列中的读写请求进行响应。如此,通过预设命名空间分别与磁盘分区和虚拟设备动态绑定的方式,实现了对磁盘资源的隔离和动态划分,再加上主机的读写请求队列与虚拟设备之间的映射,实现了对读写请求队列所分配资源的动态划分,进而实现了各个预设命名空间的读写负载均衡。

附图说明

[0022] 图1是相关技术中进行磁盘读写资源分配的整体框架图;

[0023] 图2A是本申请实施例提供的云平台的结构示意图;

[0024] 图2B是本申请实施例提供的IaaS层、PaaS层和SaaS层的部署关系示意图;

[0025] 图3是本申请实施例提供的主节点的结构示意图;

[0026] 图4是本申请实施例提供的磁盘读写资源的分配方法的可选的流程示意图;

[0027] 图5是本申请实施例提供的磁盘读写资源的分配方法的可选的流程示意图;

[0028] 图6是本申请实施例提供的磁盘读写资源的分配方法的可选的流程示意图;

[0029] 图7是本申请实施例提供的磁盘读写资源的分配方法的可选的流程示意图;

[0030] 图8是本申请实施例提供的磁盘读写资源的分配方法的可选的流程示意图;

[0031] 图9是本申请实施例提供的系统框架图;

[0032] 图10是本申请实施例提供的资源隔离模块的实现方案结构图;

[0033] 图11是本申请实施例提供的NVMe PRP资源映射图;

[0034] 图12是本申请实施例提供的队列调度图;

[0035] 图13是本申请实施例提供的测试框架图。

具体实施方式

[0036] 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,所描述的实施例不应视为对本申请的限制,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它实施例,都属于本申请保护的范围。

[0037] 在以下的描述中,涉及到“一些实施例”,其描述了所有可能实施例的子集,但是可以理解,“一些实施例”可以是所有可能实施例的相同子集或不同子集,并且可以在不冲突的情况下相互结合。除非另有定义,本申请实施例所使用的所有的技术和科学术语与属于本申请实施例的技术领域的技术人员通常理解的含义相同。本申请实施例所使用的术语只是为了描述本申请实施例的目的,不是旨在限制本申请。

[0038] 为了更好地理解本申请实施例中提供的磁盘读写资源的分配方法,首先对相关技术中的磁盘读写资源的分配方法进行说明,下面以NVMe SSD为待分配资源的磁盘为例进行说明:

[0039] 相关技术中,随着NVMe SSD容量的变大,单盘的故障范围及利用率无法满足要求,需要考虑通过容量拆分的方式实现资源的复用及共享,并且,NVMe规范(NVMe Spec)中的多个命名空间(NameSpace)也应运而生。但命名空间技术当前在NVMe规范中仅是针对设备自身定义的,且命名空间没有约束队列管理及分配机制,只定义了规则。

[0040] 与本申请相似的技术方案,目前业界没有实例,NVMe SSD厂家从设备侧有依据NVMe规范推出类似方案,但只具备简单的资源隔离及高速串行计算机扩展总线标准(PCIe, Peripheral Component Interconnect express)控制器的虚拟设备(VF, Virtual Function)使能能力,且目前的技术受限于设备自身,暂没有进一步推进的演进技术,图1是相关技术中进行磁盘读写资源分配的整体框架图,如图1所示,相关技术中的技术方案的处理流程如下:首先由主控制器(Primary Controller)11明确支持的资源(Resource)大小,由主控制器11做资源的分配管理;再将特定大小的资源挂载到二级控制器(Secondary Controller)12下。其中,在主控制器11中,包括私有资源(Private Resources)111和可分配资源(Flexible Resource)112,可分配资源112是基于灵活分配机制来进行分配的,所有的可分配资源112共同形成资源池(Resource Pool)13,资源池13中包括多个未分配的可分配资源112。每个二级控制器12可以从资源池13中分配到可分配资源112。

[0041] 相关技术中的方法,至少存在以下问题:虽然定义了资源池,但没有约束分配及绑定规则,实际应用难度较大;相关技术的方法没有定义关于NVMe队列的分配机制及池化概念,不具备负载均衡能力;对于不同的应用场景,无法满足NVMe设备均衡化多对多分配需求,资源及队列利用率不足;更多集中在NVMe SSD下的实现,未考虑自定义软硬件结合方案,不具备灵活性及可拓展性。

[0042] 基于相关技术中所存在的上述至少一个问题,本申请实施例提供一种NVM命名空间+队列资源池化的概念,通过软硬件结合的方式,设计实现与PCIe控制器的物理设备(PF, Physical Function)或VF的动态映射,在保证NVMe资源的有效隔离的同时,实现资源的动态拓展及负载均衡的能力。

[0043] 本申请实施例提供一种磁盘读写资源的分配方法,首先,接收主机发送的读写请求队列,其中,读写请求队列中包括主机的主机标识;然后,依次根据主机标识和主机与虚拟设备之间的第一映射关系,确定出与主机对应的目标虚拟设备的设备标识;根据设备标识、和虚拟设备与预设命名空间之间的第二映射关系,确定出与主机对应的目标命名空间的命名空间标识;根据命名空间标识、和预设命名空间与磁盘的磁盘分区之间的第三映射关系,确定

出与主机对应的目标磁盘分区的分区标识;最后,将分区标识对应的目标磁盘分区分配给主机,通过目标磁盘分区对读写请求队列中的读写请求进行响应。如此,通过预设命名空间分别与磁盘分区和虚拟设备动态绑定的方式,实现对磁盘资源的隔离和动态划分,再加上主机的读写请求队列与虚拟设备之间的映射,实现了对读写请求队列所分配资源的动态划分,进而实现了各个预设命名空间的读写负载均衡。

[0044] 下面说明本申请实施例的磁盘读写资源的分配设备的示例性应用,在一种实现方式中,本申请实施例提供的磁盘读写资源的分配方法涉及云技术领域,可以通过云技术来实现,例如,本申请实施例中磁盘读写资源的分配设备可以是云平台中的节点。图2A是本申请实施例提供的云平台的结构示意图,如图2A所示,云平台20中包括主节点21和多个工作节点22,其中,主节点21可以作为本申请实施例的磁盘读写资源的分配设备,即主节点21作为云平台20中的多个节点的主控节点,用于对多个工作节点22进行控制。在一些实施例中,工作节点22可以是发送读写请求的主机(Host),主节点21对工作节点22发送的读写请求队列进行资源分配,将云平台对应的物理资源23(其中,物理资源23可以由多个SSD形成的资源池)进行资源虚拟化后,通过虚拟化资源对应的虚拟设备VF与主机之间的映射关系,将虚拟化资源对应的实际物理资源分配给主机的每一读写请求队列,实现对读写请求队列的读写请求进行响应。

[0045] 本申请实施例中,主节点21作为磁盘读写资源的分配设备,至少还能够实现以下功能:接收每一工作节点22发送的读写请求队列,其中,读写请求队列中包括工作节点22的主机标识;根据主机标识和主机与虚拟设备之间的第一映射关系,确定出与工作节点22对应的目标虚拟设备的设备标识;根据设备标识、和虚拟设备与预设命名空间之间的第二映射关系,确定出与工作节点22对应的目标命名空间的空间标识;根据空间标识、和预设命名空间与磁盘的磁盘分区之间的第三映射关系,确定出与工作节点22对应的目标磁盘分区的分区标识;将分区标识对应的目标磁盘分区分配给工作节点22,以实现通过目标磁盘分区对读写请求队列中的读写请求进行响应。

[0046] 需要说明的是,云技术(Cloud technology)是指在广域网或局域网内将硬件、软件、网络等系列资源统一起来,实现数据的计算、储存、处理和共享的一种托管技术。云技术(Cloud technology)基于云计算商业模式应用的网络技术、信息技术、整合技术、管理平台技术、应用技术等的总称,可以组成资源池,按需所用,灵活便利。云计算技术将变成重要支撑。技术网络系统的后台服务需要大量的计算、存储资源,如视频网站、图片类网站和更多的门户网站。伴随着互联网行业的高度发展和应用,将来每个物品都有可能存在自己的识别标志,都需要传输到后台系统进行逻辑处理,不同程度级别的数据将会分开处理,各类行业数据皆需要强大的系统后盾支撑,只能通过云计算来实现。

[0047] 作为云计算的基础能力提供商,会建立云计算资源池平台,简称云平台,一般称为基础设施即服务(IaaS,Infrastructure as a Service),在资源池中部署多种类型的虚拟资源,供外部客户选择使用。云计算资源池中主要包括:计算设备(为虚拟化机器,包含操作系统)、存储设备和网络设备。按照逻辑功能划分,在IaaS层上可以部署平台即服务(PaaS,Platform as a Service)层,PaaS层之上再部署软件即服务(SaaS,Software as a Service)层,也可以直接将SaaS层部署在IaaS层上。PaaS层为软件运行的平台,如数据库、web容器等。SaaS层为各式各样的业务软件,如web门户网站、短信群发器等。其中,IaaS层

203、PaaS层202和SaaS层201部署关系如图2B所示,一般来说,SaaS层201和PaaS层202相对于IaaS层203是上层。

[0048] 图3是本申请实施例提供的主节点21的结构示意图,图3所示的主节点21包括:至少一个处理器310、存储器350、至少一个网络接口320和用户接口330。主节点21中的各个组件通过总线系统340耦合在一起。可理解,总线系统340用于实现这些组件之间的连接通信。总线系统340除包括数据总线之外,还包括电源总线、控制总线和状态信号总线。但是为了清楚说明起见,在图3中将各种总线都标为总线系统340。

[0049] 处理器310可以是一种集成电路芯片,具有信号的处理能力,例如通用处理器、数字信号处理器(DSP,Digital Signal Processor),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任何常规的处理器等。

[0050] 用户接口330包括使得能够呈现媒体内容的一个或多个输出装置331,包括一个或多个扬声器和/或一个或多个视觉显示屏。用户接口330还包括一个或多个输入装置332,包括有助于用户输入的用户接口部件,比如键盘、鼠标、麦克风、触屏显示屏、摄像头、其他输入按钮和控件。

[0051] 存储器350可以是可移除的,不可移除的或其组合。示例性的硬件设备包括固态存储器,硬盘驱动器,光盘驱动器等。存储器350可选地包括在物理位置远离处理器310的一个或多个存储设备。存储器350包括易失性存储器或非易失性存储器,也可包括易失性和非易失性存储器两者。非易失性存储器可以是只读存储器(ROM,Read Only Memory),易失性存储器可以是随机存取存储器(RAM,Random Access Memory)。本申请实施例描述的存储器350旨在包括任意适合类型的存储器。在一些实施例中,存储器350能够存储数据以支持各种操作,这些数据的示例包括程序、模块和数据结构或者其子集或超集,下面示例性说明。

[0052] 操作系统351,包括用于处理各种基本系统服务和执行硬件相关任务的系统程序,例如框架层、核心库层、驱动层等,用于实现各种基础业务以及处理基于硬件的任务;

[0053] 网络通信模块352,用于经由一个或多个(有线或无线)网络接口320到达其他计算设备,示例性的网络接口320包括:蓝牙、无线相容性认证(WiFi)、和通用串行总线(USB,Universal Serial Bus)等;

[0054] 输入处理模块353,用于对一个或多个来自一个或多个输入装置332之一的一个或多个用户输入或互动进行检测以及翻译所检测的输入或互动。

[0055] 在一些实施例中,本申请实施例提供的装置可以采用软件方式实现,图3示出了存储在存储器350中的一种磁盘读写资源的分配装置354,该磁盘读写资源的分配装置354可以是主节点21中的磁盘读写资源的分配装置,其可以是程序和插件等形式的软件,包括以下软件模块:接收模块3541、第一确定模块3542、第二确定模块3543、第三确定模块3544和分配模块3545,这些模块是逻辑上的,因此根据所实现的功能可以进行任意的组合或进一步拆分。将在下文中说明各个模块的功能。

[0056] 在另一些实施例中,本申请实施例提供的装置可以采用硬件方式实现,作为示例,本申请实施例提供的装置可以是采用硬件译码处理器形式的处理器,其被编程以执行本申请实施例提供的磁盘读写资源的分配方法,例如,硬件译码处理器形式的处理器可以采用一个或多个应用专用集成电路(ASIC,Application Specific Integrated Circuit)、

DSP、可编程逻辑器件(PLD, Programmable Logic Device)、复杂可编程逻辑器件(CPLD, Complex Programmable Logic Device)、现场可编程门阵列(FPGA, Field-Programmable Gate Array)或其他电子元件。

[0057] 下面将结合本申请实施例提供的主节点21的示例性应用和实施,说明本申请实施例提供的磁盘读写资源的分配方法。参见图4,图4是本申请实施例提供的磁盘读写资源的分配方法的一个可选的流程示意图,将结合图4示出的步骤进行说明。

[0058] 步骤S401,接收主机发送的读写请求队列。

[0059] 这里,读写请求队列中包括主机的主机标识。本申请实施例中,读写请求队列可以是提交队列(SQ, Submission Queue)或者完成队列(CQ, Completion Queue)中的任意一种,读写请求队列中包括至少一个读写请求,读写请求队列是将主机发送的读写请求写入至预设的SQ队列或预设的CQ队列中所形成的。每一主机可以对应至少一个SQ队列或CQ队列。

[0060] 步骤S402,根据主机标识和主机与虚拟设备之间的第一映射关系,确定出与主机对应的目标虚拟设备的设备标识。

[0061] 这里,第一映射关系是主机集群中的每一主机与虚拟设备之间的映射关系,对于每个主机上的虚拟机和主机上的应用程序,均会分配至少一个虚拟设备,每个虚拟设备对应一个虚拟磁盘,虚拟磁盘是物理磁盘虚拟化之后的虚拟存储空间,也就是说,虚拟磁盘是对物理磁盘采用虚拟化技术进行处理之后得到的虚拟存储空间。通过虚拟设备对应的虚拟磁盘为该主机的读写请求队列分配虚拟存储空间。

[0062] 本申请实施例中,由于每一主机可以与一个或多个虚拟设备(即虚拟磁盘)之间具有第一映射关系,因此,在一些实施例中,可以根据主机标识确定出一个或多个目标虚拟设备。

[0063] 步骤S403,根据设备标识、和虚拟设备与预设命名空间之间的第二映射关系,确定出与主机对应的目标命名空间的空间标识。

[0064] 这里以目标虚拟设备具有一个为例进行说明,在确定出一个目标虚拟设备之后,根据目标虚拟设备的设备标识和第二映射关系,得到与目标虚拟设备对应的目标命名空间,该命名空间即与主机对应的目标命名空间。

[0065] 本申请实施例中,第二映射关系是指虚拟设备与预设命名空间之间的映射关系,每一虚拟设备对应一个命名空间。

[0066] 步骤S404,根据空间标识、和预设命名空间与磁盘的磁盘分区之间的第三映射关系,确定出与主机对应的目标磁盘分区的分区标识。

[0067] 这里,由于根据第二映射关系可以确定出虚拟设备对应的目标命名空间,因此,可以根据目标命名空间与磁盘分区之间的第三映射关系确定目标磁盘分区。需要说明的是,本申请实施例中可以将磁盘阵列中的多个磁盘进行分区,每一磁盘对应至少一个磁盘分区,并且,对于磁盘阵列中的每一磁盘,目标命名空间可以均对应有一个磁盘分区。也就是说,对于每一预设命名空间可以在磁盘阵列中的每一个磁盘中对应一个磁盘分区,那么,在确定出目标命名空间之后,可以确定出磁盘阵列对应的多个磁盘分区。

[0068] 步骤S405,将分区标识对应的目标磁盘分区分配给主机,通过目标磁盘分区对读写请求队列中的读写请求进行响应。

[0069] 这里,在确定出目标磁盘分区之后,将目标磁盘分区分配给主机,通过该目标磁盘分区,实现对主机的读写请求队列进行读写请求响应。

[0070] 本申请实施例提供的磁盘读写资源的分配方法,对于接收到的主机发送的读写请求队列,根据主机的主机标识,依次基于主机与虚拟设备之间的第一映射关系、虚拟设备与预设命名空间之间的第二映射关系和预设命名空间与磁盘的磁盘分区之间的第三映射关系,确定出主机对应的目标磁盘分区,从而将目标磁盘分区的资源分配给主机的读写请求队列,以实现通过所述目标磁盘分区对所述读写请求队列中的读写请求进行响应。如此,通过预设命名空间分别与磁盘分区和虚拟设备动态绑定的方式,实现了对磁盘资源的隔离和动态划分,再加上主机的读写请求队列与虚拟设备之间的映射,实现了对读写请求队列所分配资源的动态划分,进而实现了各个预设命名空间的读写负载均衡。

[0071] 在一些实施例中,磁盘读写资源的分配系统中包括主节点和主机集群,主机集群中包括至少一个主机,该主机可以是虚拟机,也可以是物理机,主机上运行有至少一个应用程序,通过该应用程序向主节点发送读写请求,以请求主节点对主机的读写请求进行资源分配。主节点可以是软件和硬件结合后得到的节点,主节点上运行有计算机程序产品,该计算机程序产品在运行时能够实现本申请实施例提供的磁盘读写资源的分配方法。

[0072] 下面以磁盘读写资源的分配系统中包括主节点和的主机H1为例进行说明,图5是本申请实施例提供的磁盘读写资源的分配方法的一个可选的流程示意图,如图5所示,方法包括以下步骤:

[0073] 步骤S501,主机H1在运行应用程序的过程中,获取至少一条读写请求。

[0074] 步骤S502,将至少一条读写请求发送给主节点。

[0075] 步骤S503,主节点将至少一条读写请求写入至主机H1的预设的提交队列或完成队列中,形成读写请求队列。读写请求队列中包括主机H1的主机标识H11。这里,提交队列和完成队列可以是磁盘内存中的一个区域,在数据结构原理上,这里的提交队列和完成队列可以是一个环形缓冲区,在这个环形缓冲区中,包括头部和尾部,主机H1的读写请求可以按照环形缓冲区的队列从头部至尾部依次写入读写请求。

[0076] 步骤S504,主节点根据主机标识H11和主机与虚拟设备之间的第一映射关系,确定出与主机H1对应的目标虚拟设备的设备标识。

[0077] 步骤S505,根据设备标识、和虚拟设备与预设命名空间之间的第二映射关系,确定出与主机H1对应的目标命名空间的空间标识。

[0078] 步骤S506,根据空间标识、和预设命名空间与磁盘的磁盘分区之间的第三映射关系,确定出与主机H1对应的目标磁盘分区的分区标识。

[0079] 步骤S507,将分区标识对应的目标磁盘分区分配给主机H1,以实现通过目标磁盘分区对主机H1的读写请求队列中的读写请求进行响应。

[0080] 需要说明的是,步骤S504至步骤S507与上述步骤S402至步骤S405相同,本申请实施例不再赘述。

[0081] 本申请实施例提供的磁盘读写资源的分配方法,主节点可以是实现本申请实施例的方法对应的软件与一硬件载体的结合,通过主机与主节点之间的交互,实现了对主机的读写请求队列的资源分配,且能够自定义软硬件的结合方案,具备灵活性及可拓展性。同时,通过预设命名空间分别与磁盘分区和虚拟设备动态绑定的方式,实现了对磁盘资源的

隔离和动态划分,再加上主机的读写请求队列与虚拟设备之间的映射,实现了对读写请求队列所分配资源的动态划分,进而实现了各个预设命名空间的读写负载均衡。

[0082] 在一些实施例中,可以在接收主机发送的读写请求队列之前就对主机集群中的每一主机分配好磁盘阵列中的磁盘分区,这样,在接收到该主机发送的读写请求时,可以直接在分配的磁盘分区内对读写请求进行响应,实现数据的读出或写入。基于图4,图6是本申请实施例提供的磁盘读写资源的分配方法的一个可选的流程示意图,如图6所示,在步骤S401之前,方法还包括以下步骤:

[0083] 步骤S601,采用至少一个所述预设命名空间,对待池化的磁盘阵列中的每一磁盘的磁盘资源进行资源隔离处理,得到隔离后的多个磁盘分区,其中,每一磁盘分区与一个预设命名空间之间具有所述第三映射关系。

[0084] 这里,在主机发送读写请求之前,预先对磁盘阵列中的资源进行池化、隔离和映射,即预先对资源进行分配,从而能够保证在后续接收到主机发送的读写请求时,按照预先分配的资源对读写请求进行响应,实现了对资源的预分配,保证对主机的读写请求能够进行及时有效的响应。

[0085] 本申请实施例中,在进行资源隔离处理时,可以是一个预设命名空间对应一个磁盘分区,这样通过多个预设命名空间即可实现对磁盘资源的隔离。

[0086] 在一些实施例中,步骤S601可以通过以下步骤实现:

[0087] 步骤S6011,采用每一预设命名空间,对每一磁盘中的连续地址对应的连续磁盘资源进行资源隔离处理,得到隔离后的多个磁盘分区。

[0088] 这里,每一磁盘分区的磁盘资源对应连续的地址,即通过每一预设命名空间对磁盘资源进行隔离时,是将一段连续地址对应的磁盘资源确定为一个磁盘分区,并且每一磁盘分区的容量与其他磁盘分区的容量可以相同,也可以不同。

[0089] 本申请实施例中,采用预设命名空间对磁盘资源进行资源隔离处理,可以理解成是将预设命名空间与磁盘资源的任意一个磁盘分区进行映射。那么,对于磁盘阵列中的每一磁盘,可以均具有一个与同一预设命名空间具有映射关系的磁盘分区。

[0090] 本申请实施例中,通过预设命名空间对磁盘阵列中的磁盘资源进行划分,形成多个磁盘分区,这样,通过多个磁盘的磁盘分区可以形成一资源池,在这个资源池中包括大量的磁盘分区,每一磁盘分区对应一定容量的存储空间,整个资源池的资源容量之和即磁盘阵列中的全部磁盘的资源容量之和,通过对磁盘阵列中的磁盘的资源划分,实现了资源的共享,即资源池中的资源可以被任一主机的应用使用。

[0091] 步骤S602,将具有第三映射关系的磁盘分区和预设命名空间,与虚拟设备集合中的一个虚拟设备进行资源映射处理,以得到虚拟设备与磁盘分区和预设命名空间之间的第二映射关系。

[0092] 每个主机上的虚拟机和主机上的应用程序,均会分配至少一个虚拟设备,那么对于当前系统中的多个主机,可以对应一虚拟设备集群,每一虚拟设备在步骤S601划分后得到的资源池中对一磁盘分区,通过该磁盘分区实现对该虚拟设备的读写请求进行响应。

[0093] 举例来说,磁盘SSD中包括两个SSD磁盘分区,分别是:SSD1和SSD2,预设命名空间包括NS1和NS2,虚拟设备集群中包括四个虚拟设备:VF1、VF2、VF3和VF4,那么本申请实施例的第二映射关系和第三映射关系可以是:VF1与SSD1和NS1对应、VF2与SSD1和NS2对应、VF3

与SSD2和NS1对应、VF4与SSD2和NS2对应。也就是说,对于VF1与SSD1和NS1对应,那么,虚拟设备VF1与磁盘分区SSD1和预设命名空间NS1具有第二映射关系,且磁盘分区SSD1和预设命名空间NS1具有第三映射关系;对于VF4与SSD2和NS2对应,那么,虚拟设备VF4与磁盘分区SSD2和预设命名空间NS2具有第二映射关系,且磁盘分区SSD2和预设命名空间NS2具有第三映射关系。

[0094] 步骤S603,预先设置主机集群中的每一主机,与虚拟设备集合中的至少一个虚拟设备之间的第一映射关系。

[0095] 这里,第一映射关系是主机与虚拟设备之间的映射关系,第一映射关系可以是系统预先设置的,也就是说,预先为每一主机分配好特定的虚拟设备。

[0096] 请继续参照图6,在一些实施例中,在磁盘阵列中的多个磁盘中,每一预设命名空间与每一磁盘中的一个磁盘分区之间具有第三映射关系;因此,在确定目标磁盘分区时,可以从多个磁盘中对应的多个磁盘分区中确定出一个磁盘分区。因此,在步骤S404之后,方法还可以包括以下步骤:

[0097] 步骤S604,对于每一预设命名空间对应的任一磁盘的磁盘分区,根据磁盘分区的读写响应速度或磁盘分区的容量,确定出磁盘分区的第一优先级。

[0098] 这里,磁盘分区的容量越大,该磁盘分区的第一优先级越高;磁盘分区的读写响应速度越高,该磁盘分区的第一优先级越高。

[0099] 步骤S605,获取主机发送的读写请求队列的第二优先级。

[0100] 这里,读写请求队列的第二优先级可以是根据读写请求队列的重要性和处理紧急性来确定的,主机在发送读写请求队列时,可以为每一读写请求队列添加优先级标签或重要级标签,以明确该读写请求队列的第二优先级。

[0101] 步骤S606,根据第二优先级和每一磁盘分区的第一优先级,在多个磁盘中匹配出一个磁盘,并将所匹配出的磁盘中的磁盘分区确定为目标磁盘分区。

[0102] 这里,由于根据主机标识可以在多个磁盘中匹配到多个目标磁盘分区,因此,可以根据读写请求队列的第二优先级,来选择一个合适的目标磁盘分区进行读写请求响应。举例来说,当该读写请求队列是对时延要求比较高或比较重要的读写请求,因此该读写请求队列具有较高的第二优先级,所以可以在多个目标磁盘分区中选择具有较高第一优先级的目标磁盘分区,作为最终对读写请求队列进行响应的目标磁盘分区。

[0103] 请继续参照图6,在一些实施例中,在步骤S401之前,方法还包括以下步骤:步骤S607,为主机添加初始化标签,初始化标签用于标识主机的读写请求队列对应的目标磁盘分区的目标容量。

[0104] 这里,目标容量对应于读写请求队列对存储容量的需求量,如果读写请求队列对存储容量的需求量较高,则初始化标签可以是对应较高目标容量的标签,例如,可以是将标签设置为“High”;如果读写请求队列对存储容量的需求量较低,则初始化标签可以是对应较低目标容量的标签,例如,可以是将标签设置为“Low”。

[0105] 对应地,在步骤S404之后,方法还可以包括以下步骤:

[0106] 步骤S608,根据目标容量,在多个磁盘中匹配出一个或多个磁盘。

[0107] 这里,如果初始化标签为“Low”,则可以在多个磁盘中匹配出一个较低容量的磁盘;如果初始化标签为“High”,则可以在多个磁盘中匹配出一个较高容量的磁盘,或者匹配

出多个磁盘,通过多个磁盘中对应的多个磁盘分区同时对主机的读写请求进行响应。

[0108] 步骤S609,将所匹配出的一个或多个磁盘中的磁盘分区,确定为目标磁盘分区。

[0109] 基于图4,图7是本申请实施例提供的磁盘读写资源的分配方法的一个可选的流程示意图,如图7所示,步骤S404可以通过以下步骤实现:

[0110] 步骤S701,确定目标虚拟设备的页面偏移和空间标识对应的目标命名空间的地址。

[0111] 这里,每一命名空间具有一特定的地址,目标虚拟设备的页面偏移可以是一地址偏移值。

[0112] 步骤S702,将页面偏移对应的偏移值,确定为目标虚拟设备的地址相对于目标命名空间的地址的偏移量。

[0113] 这里,将页面偏移对应的偏移值为地址偏移值,本申请中将地址偏移值确定为目标虚拟设备的地址相对于目标命名空间的地址的偏移量。

[0114] 步骤S703,将相对于目标命名空间的地址,偏移该偏移量之后的偏移地址,确定为与主机对应的目标磁盘分区的地址。

[0115] 这里,以目标命名空间的地址为首地址,确定相对于该首地址偏移了上述偏移量之后的地址为偏移地址,则该偏移地址对应的磁盘分区即为目标磁盘分区。需要说明的是,由于每个磁盘分区对应多个连续的地址,因此,在确定出一个磁盘分区的任一地址后,均可以根据该地址确定出目标磁盘分区。

[0116] 步骤S704,根据目标磁盘分区的地址,确定目标磁盘分区的分区标识。

[0117] 这里,目标磁盘分区的分区标识可以是目标磁盘分区的地址,也可以是能够区分出该目标磁盘分区的任意一种标识,例如,名称、地址区域(即磁盘分区的首地址与尾地址)、与该目标磁盘分区唯一对应的命名空间的标识等。

[0118] 基于图4,图8是本申请实施例提供的磁盘读写资源的分配方法的一个可选的流程示意图,如图8所示,在步骤S404之前,方法还包括以下步骤:

[0119] 步骤S801,确定主机的当前的读写请求队列对应的资源分配权重。

[0120] 本申请实施例中,提供以下三种确定资源分配权重的方式:

[0121] 方式一:步骤S8011,在对主机的资源分配权重进行调整之后,确定主机的当前读写请求量和目标虚拟设备中处于等待状态的待处理读写请求量。步骤S8012,根据当前读写请求量和待处理读写请求量,确定主机的当前的读写请求队列对应的资源分配权重。

[0122] 方式二:步骤S8013,确定历史周期内对读写请求进行响应的延迟时间的第一均值、和当前周期内对读写请求进行响应的延迟时间的第二均值。步骤S8014,如果第二均值大于或等于第一均值,增大主机的当前的读写请求队列对应的资源分配权重。步骤S8015,如果第二均值小于第一均值,减小主机的当前的读写请求队列对应的资源分配权重。

[0123] 方式三:步骤S8016,确定主机的读写请求队列的延迟敏感类型。步骤S8017,如果读写请求队列为尾部延迟敏感型队列,增大主机的当前的读写请求队列对应的资源分配权重,且增大后的资源分配权重是当前的磁盘读写资源的分配系统中全部读写请求队列中的最大权重。

[0124] 对应地,步骤S404可以通过以下步骤实现:

[0125] 步骤S802,根据空间标识和资源分配权重,在磁盘阵列中的多个磁盘中,确定出一

个或多个磁盘为目标磁盘。这里,资源分配权重越高,对应的分配到的磁盘资源越多,资源分配权重越低,对应的分配到的磁盘资源越少。或者,资源分配权重越高,对应的分配到的目标磁盘的数量越大,资源分配权重越低,对应的分配到的目标磁盘的数量越小。

[0126] 步骤S803,在所匹配出的目标磁盘中,将与第三映射关系对应的磁盘分区确定为目标磁盘分区。

[0127] 本申请实施例提供的磁盘读写资源的分配方法,通过配置主机的当前的读写请求队列对应的资源分配权重,对于不同的主机设置不同的资源分配权重,因此可以根据资源分配权重为对应主机分配合理的磁盘资源,实现对资源的合理分配,并且,资源分配权重可以根据当前的读写请求量进行动态调整,因此可以实现对资源的动态分配,提高了系统对每个主机的读写请求队列的快速有效响应。

[0128] 下面,将说明本申请实施例在一个实际的应用场景中的示例性应用。

[0129] 本申请实施例提供一种磁盘读写资源的分配方法,通过将NameSpace (即命名空间,对应上述的预设命名空间) 与PCIe VF/PF动态绑定的机制,实现N VMe设备资源的隔离及动态划分,再加上独占内存对NVMe队列资源 (即读写请求队列所占用的资源) 做动态划分,实现各NameSpace之间的读写负载均衡。

[0130] 本申请实施例的方法能应用到存储端以及高IO裸金属场景,实现NVMe的资源卸载,并极大增强用户对NVMe的使用便捷性。同时,可以保证用户仅需配置各应用优先级的情况下,提供稳定可控的业务带宽及延时体验。

[0131] 本申请实施例典型的应用场景如在高性能IO实例中,本申请实施例的方法可以用于后端存储,通过NVMe队列资源与PF/VF的动态映射,实现IO资源的动态调整;在NVMe映射框架中引入IO队列的调度逻辑,来提高队列资源利用率;克服了当前业界暂无有效的NVMe IO资源共享及虚拟化方案的大背景,且提供了一套NVMe资源池化的方案;解决了NVMe资源使用时必须占用大量CPU资源,同时无法有效的实现NVMe资源池化及动态调度,进而导致的利用率低、性能不可控等问题。

[0132] 图9是本申请实施例提供的系统框架图,如图9所示,本申请实施例的系统从下到上依次包括磁盘阵列91,部署有本申请实施例的硬件资源管理方案的硬件层 (HW, Hardware) 92、即主节点,运行在基础物理服务器和操作系统之间的中间软件层,可允许多个操作系统和应用共享硬件的虚拟机监视器 (Hypervisor) 93和用户层94。

[0133] 其中,磁盘阵列91中包括多个NVMe SSD磁盘911;

[0134] 在硬件层92中,包括NVMe根复合体 (root complex) 设备921、NVMe虚拟化管理器 (NVMe Virtualization Manager) 922、和PCIe控制器的物理设备 (PF) 923和至少一个PCIe控制器的虚拟设备 (VF) 924;在硬件层92和虚拟机监视器93之间,通过读写内存管理单元 (IOMMU, IO Memory Management Unit) 实现硬件层92和虚拟机监视器93的通信。

[0135] 在虚拟机监视器93中,包括:PF驱动器931、基于PCI总线的虚拟设备的数据IO (VFIO-PCI) 932、VF设备的数据IO调节器 (mediated-VFIO) 933、IO控制管理器 (Admin IOCTL) 934、NVMe PF 935、至少一个VF设备的数据IO (VFIO) 936、muser调度937,需要说明的是,PF驱动器931、VFIO-PCI 932、mediated-VFIO 933是三种并列的与硬件层92进行交互的方式,可以选择三者中的任意一种实现与硬件层92的交互,也可以对于不同的主机,采用三者中不同的方式实现与硬件层92的交互。其中,PCIe控制器的物理设备923与虚拟机监视

器93中的PF驱动器931之间通过IO控制 (IOCTL) 实现交互;PCIe控制器的虚拟设备924与VFIO-PCI 932之间进行IO数据交互。

[0136] 在用户层94中,包括多个主机941 (这里的主机,不仅包括主机Host,还包括客户机Guest),每一主机上运行有应用程序(APP),每一主机上还包括标准NVMe (Standard NVMe) 942、和NVMe字符 (NVMe Character) 943。

[0137] 本申请实施例上述系统的关键在于,在底层的硬件 (HW,Hardware) 中部署一套硬件资源管理方案,结合内部软件调度机制,实现对NVMe SSD资源的动态调度。对HW硬件之下的NVMe SSD磁盘阵列,本申请实施例的方案,HW中的硬件资源管理方案体现为对NVMe SSD的管理端口,负责NVMe SSD的初始化、资源配置以及队列和NameSpace的分配,保证NVMe SSD的正常接入;在HW中的硬件资源管理方案的内部,可以将申请的内存资源 (也可做独占资源设计) 分配到对应的队列 (即主机发送的读写请求队列),在同一段内存中实现统一管理和调度;对HW中的硬件资源管理方案之上的虚拟机监视器93和用户层94的管理,实现NVMe控制器 (NVMe Controller) 的模拟及上报 (即将硬件层92模拟为NVMe控制器),并通过特定的绑定机制,将PF/VF与NameSpace、队列资源统一绑定后,提供虚拟的NVMe控制器给上层虚拟机监视器93或用户层94的应用程序直接使用。

[0138] 本申请实施例的磁盘读写资源的分配方法主要包括以下几大部分:

[0139] 第一部分:NVMe队列资源隔离模块。

[0140] 图10是本申请实施例提供的资源隔离模块的实现方案结构图,如图10所示,最底层为待分配的NVMe资源1001,NVMe资源1001之上是硬件层1002,硬件层1002之上是主机1003。

[0141] 其中,在硬件层1002中,硬件层的虚拟机监视器 (VMM,Virtual Machine Monitor) 对应客户端物理地址 (GPA,Guest Physical Address) 和主机物理地址 (HPA,Host Physical Address),GPA和HPA与动态随机存取存储器 (Dynamic Random Access Memory, DRAM) 对应,其中,GPA是上层应用下发下来的地址,GPA需要通过预设的扩展页表 (EPT, Extension Page Tables) 转换为HPA,HPA与DRAM中的实际地址 (即HW DRAM) 对应。在硬件层1002中,还包括直接存储器访问 (DMA,Direct Memory Access) 的缓冲区 (DMA Buffer) 1004,DMA Buffer 1004位于物理队列池 (Physical Queue Pool) 1005中,物理队列池1005中包括多个物理提交队列 (PSQ,Physical Submission Queue) 和多个物理完成队列 (PCQ, Physical Completion Queue)。

[0142] 在主机1003中,每一个主机内部具有至少一个虚拟机 (VM,Virtual Machine),例如图10所示的VM1和VMx,每一虚拟机中具有客户端应用 (Guest APP),在客户端应用中产生虚拟提交队列 (VSQ,Virtual Submission Queue) 和虚拟完成队列 (VCQ,Virtual Completion Queue)。每个虚拟机可以分配一个或多个VF (如图10中的VF1和VFx),虚拟机VM1和VMx通过IOMMU与VF之间进行通信。

[0143] 本申请实施例中,对于不同的NVMe虚拟化请求及资源,通过硬件方式实现对内存及DMA缓冲区 (DMA Buffer) 1004的独占申请,保证资源独立性及池化,对上呈现的是PCIe功能 (PCIe Function),因为访问PCIe Function时对应的是客户端虚拟地址 (GVA,Guest Virtual Address),进而实现了访问地址的隔离。NVMe资源在内存池中通过与要绑定的PF/VF的DMA Buffer做对应,通过VFIO自带的IOMMU机制以及硬件内部自定义维护的EPT表实现

地址转换,IO服务质量(QoS,Quality of Service)配置对应的映射方式,将与硬件队列相关的硬件资源抽象为类似NVMe提交队列/完成队列(SQ/CQ,Sub mission Queue/Completion Queue)的环形队列方式,保证对上的接口不变。

[0144] 内部NVMe SSD资源,通过NameSpace实现逻辑区块地址(LBA,Logical Block Address)的访问隔离,并通过与NVMe集(NVMe Set)及NVMe Die的绑定,实现对各个NVMe NameSpace的IO访问资源隔离。保证多个NameSpace同时并发读写时,不会出现性能冲突及资源抢占。

[0145] 在内部独占内存中,各个NameSpace对应的NVMe CQ/SQ单独分配,每个队列对仅对应一个NameSpace,实现了IO下发过程中的性能及内存隔离。

[0146] VF将绑定后的资源暴露给上层应用后,用户侧的应用程序只能访问到虚拟内存地址,同时,在应用侧的GVA和GPA上,对应的地址偏移并不是实际NVMe SSD设备上的偏移,而是需要依赖映射层(即HW里面进行VF与NameSpace映射处理的映射层)的重新地址映射,进而从应用侧就先保证了内存访问的隔离。

[0147] 第二部分:资源映射模块。

[0148] 图11是本申请实施例提供的NVMe物理请求页面(PRP,Physical Request Page)的资源映射图,如图11所示,在硬件层111中,包括映射层1111,映射层中具有公共内存1112和SSD、NS与VF之间的映射关系。需要说明的是,图11中是基于每个NVMe控制器(即每个硬件层111对应的主节点)的页面基本补偿(Page Base Offset(PerController)),以及,基于每个NameSpace的页面基本补偿(Page Base Offset(PerNamespace)),以及,基于命令补偿(Command Offset)来实现映射的。

[0149] 这里的资源映射主要包括NameSpace(即图中的NS)与PCIe功能(PCIe Function)的对应及映射,在初始化时建立映射关系,在IO过程中实时映射。详细的步骤如下:

[0150] 1) 起始根据用户配置要求,将特定大小的NS及读写请求队列绑定到VF上,对应关系为设备SSD_x的NS_y对应上层模拟的VF_z;

[0151] 2) 主机下发标准的NVMe PRP请求到VF_z,此时默认NS的标识(NSID)为1,方便映射关系快速查询;

[0152] 3) NVMe资源池化层将PRP请求获取,在资源映射模块,将VF_z与SSD_xNS_y的对应关系转换为对应的页面偏移(Page Offset);

[0153] 4) NVMe资源池化层将修改了页面偏移的PRP请求下发到对应的实际NVMe SSD_x,其中保持原有的命令偏移(Command Offset)不变;

[0154] 5) 下端实际NVMe SSD处理完成后,直接返回到与下发命令SQ对应的CQ及命令ID,该部分不涉及地址转换。

[0155] 第三部分:队列调度策略。

[0156] 图12是本申请实施例提供的队列调度图,如图12所示,本申请实施例中通过软硬件结合的方式实现NVMe队列资源池化后,当前采用带虚拟队列VQ与后端物理队列PQ的动态映射及调度方式,如图12所示,不同的应用程序或客户端操作系统(OS,Operating System),对应不同的VQ,通过对各类应用增加不同的标签121(Tag)的方式,实现各个VQ间隔访问标识,并实现对后端PQ的共享占用。该策略在保证访问隔离的同时,最大程度的避免物理队列资源空闲或软件虚拟资源长时间等待的情况。在硬件层122中,包括NVMe主机管理

器1221、独立内存1222和多个PSQ (如图中的PSQ1、PSQ2和PSQ3) 和多个PCQ (如图中的PCQ1和PCQ2)。在主机123中,可以包括主机APP、Guest APP和容器APP。其中,在队列映射中,图中每个PSQ和每个PCQ中不同的填充颜色对应不同的APP,即每个PSQ和每个PCQ中给每个APP分配不同比例的队列内存空间。

[0157] 在本申请实施例中,设计了一种基于软件虚拟队列实时负载的权重分配指标:对任务完成时间敏感(Deadline)的负载,需要尽可能少的发生虚拟队列调度,防止因为其他类型负载的竞争及调度所引起的资源动态调整,所以设置其权重为最大;对吞吐量敏感的负载和延时敏感的负载,则可以考虑采用短期负载预测和反馈修正的方法,进行权重的设置及运行过程中的动态修正。

[0158] 具体地,针对不同主机所配置的指标,可以分别采用以下不同策略:

[0159] 1) 吞吐量敏感负载权重:根据实时调整权重后对应的负载和虚拟软件队列中排队等待的负载计算权重。

[0160] 2) 平均延迟敏感负载权重:根据最近几个周期平均延迟计算得到的均值,与当前周期实际的平均延迟比较。如果比当前的平均延迟低,则说明该虚拟队列的性能已经受到其他虚拟队列的影响,尝试增加权重以保持原有的平均延迟;而如果比当前的平均延迟高,则说明当前权重足以保持良好的平均延迟,可以减少该虚拟队列的权重,以增加其他队列获得物理队列仲裁的概率。

[0161] 3) 尾部延迟敏感负载权重:主要针对对IO完成时间有要求的场景,在该配置下,系统将尾部延迟敏感负载的队列设置为最高优先级,正在处理IO指令的物理队列不再加入到调度系统中,保证物理队列的独占应用。在权重更新时,也将周期内存在尾部延迟敏感IO任务的虚拟队列权重设置为最大值。

[0162] 当前主要针对应用定义了三种优先级的负载:吞吐敏感(默认为低级Low)、延时敏感(默认为中级Normal)及Deadline敏感型(默认为高级High),根据三种类型应用的虚拟队列权重,对对应能够分配的物理队列数量进行设置。设置分为以下几个步骤:

[0163] A、为Deadline敏感负载的虚拟队列分配物理队列。由于在权重更新时,该类型应用对应的虚拟队列会被默认设置为最大权重,所以这些虚拟队列在初始化时默认可以获得物理队列,不受其他类型负载的虚拟队列分配机制的影响;

[0164] B、在运行过程中,按照特定的周期进行轮询,计算该周期内平均延迟敏感负载的虚拟队列对应的应用可以获得的物理队列映射数N1;

[0165] C、根据预先设置好的权重值,为执行平均延迟敏感负载的虚拟队列分配对应比例的物理队列数,同时为了考虑高优先级应用的保障,需要确保分配给平均延迟敏感负载应用的物理队列映射数量不超过N2;

[0166] D、最后,根据权重值分配,为优先级相对较低的吞吐量敏感负载的虚拟队列分配物理队列,这里需要保证负载不出现饿死情况,同时保证所有虚拟队列都有对应的剩余物理队列映射。

[0167] 第四部分:测试过程。

[0168] 图13是本申请实施例提供的测试框架图,如图13所示,本申请实施例主要实现了对NVMe IO资源的重新调度及映射,所以对应的测试框架主要集中在对重新管理后的资源进行IO读写测试。重点需要关注的维度:多Host应用、租户隔离、IO调度以及性能保障。

[0169] 测试开始时,需要对各个模拟设备搭建对应的虚拟机或IO应用,并在虚拟机或Host上运行对应的IO测试软件,不同的虚拟机/应用可以增加不同的Tag标识,本测试中可配置无QoS指标、带宽敏感QoS、延时敏感QoS及长尾敏感QoS四种场景。IO测试过程中,主要需要观察各个虚拟机/应用的IOPS、延迟以及99.99%QoS,对应的性能结果与没有做任何调度及资源隔离的虚拟IO应用做对比。

[0170] 针对负载模型,选用了如下表1中的三种负载模型,分别针对大带宽场景及数据库延时敏感型场景:

[0171] 表1三种负载模型

[0172]	负载	块大小	iodepth	I/O特征
	多线程密集型负载	512K/1024K	32	随机读
	高并发随机SQL	4K	2-16	混合随机读写
	高性能数据库	4K	2-12	混合随机读写

[0173] 上表1中,多线程密集型负载为不限制I/O流量类型的大包负载,更侧重于吞吐量。而高并发随机SQL上的I/O负载,对平均延迟更加敏感。高性能数据库则为模拟高端数据库方案的I/O负载,对截止时间更加敏感。后两种应用的吞吐量主要针对数据库小包随机类型的负载,并且与第一种相比会相对空闲。

[0174] 本申请实施例中,通过对3个VM对应的Tag配置为高、中、低带宽优先级的方式,运行相同的多线程密集型负载,优先级为高的实例,带宽相比中级(Normal)优化10%,优先级为低的实例,带宽相比中级(Normal)劣化15%,未出现性能跌零的情况。因为当前采用软件模型模拟的方式,内存访问及转换效率相对较低,该测试结果符合预期。

[0175] 本申请实施例提供的方法,一个软硬件结合系统,实现NVMe SSD资源的池化及动态映射,并通过整合IO QoS调度策略实现NVMe IO资源的动态可调及NVMe虚拟化能力卸载;提出一种NVMe资源管理的方法,可以在接入NVMe资源的同时,实现资源的池化及动态调度,并通过PF/VF的方式实现资源隔离及动态映射。典型应用场景包括大多数NVMe存储资源访问及裸金属虚拟化场景,解决了如何利用一套标准硬件实现NVMe资源的有效管理,并据此解决NVMe资源共享隔离和IO QoS的问题。

[0176] 本申请实施例规避了传统NVMe存储系统中无法有效隔离的难题,转而设计一套IO虚拟化的框架,通过PCIe Function的角度实现资源隔离。在典型的应用场景如NVMe虚拟化场景中,该方法在资源隔离的基础上实现NVMe设备对应的IO及存储资源动态调用及分配,从而能有效改善系统使用复杂度,并增强可控性。本申请实施例克服了传统NVMe IO系统无法在达成资源共享的情况下实现完全资源隔离的问题,本发明在NVMe存储系统部署的时候不需要提前为应用规划对应的独占物理机。

[0177] 另外一个显著的有益效果是本申请实施例在实现NVMe设备的统一管理后,在系统内部实现了软件虚拟队列与硬件队列的动态绑定层,进而可以有效的添加资源调度机制,并通过各个Host的不同Tag,实现带不同优先级的调度策略,进一步的保证IO平滑。由于该部分对于硬件的配置要求不高,因此本申请实施例较其它已有方法更易于落地到真实的应用场景,如当前业界已有的FPGA或IP方案,支持的NVMe规模有限,且价格昂贵,不具备批量落地的能力。

[0178] 本申请实施例利用NVMe IO共享后无法有效隔离的前提背景,通过整合独立资源

做NVMe设备管理进而消除干扰影响;经过内部资源调度及动态资源绑定,实现QoS平稳及管理归一的优势。

[0179] 最后,本申请实施例对于不同的目标Host,可以实现更细维度的优先级仲裁,对于延时敏感型业务,可以通过配置Deadline方式保证IO的快速响应,保证在各项性能指标和长尾指标上都显著优于其它已有参考系统。

[0180] 本申请实施例通过一套软硬件结合方案,实现对NVMe SSD资源的有效管理。但是,本申请实施例不具体限定NVMe SSD对应的IO传输层,可以增加网络模块,实现对远端设备的互联支持以及集群化的管理。基于该类场景,可以通过分布式存储的技术,在大粒度(至少每个NVMe Set)的块(Chunk)前提下,实现对NVMe资源的互联,但该部分不具备虚拟化Offloading及NV Me访问隔离的能力,更多的需要依赖上层QoS策略保障。

[0181] 本申请实施例的主要创新之处包括:提出一种软硬件结合的方法,利用NVMe队列资源统一管理特征,实现资源的池化及访问隔离,并增加资源动态调度及映射能力。

[0182] 具体地,采用一段独立内存,实现对后端NVMe设备的统一管理,对NV Me SSD呈现一个统一的管理节点;在系统内部,实现队列资源的打散重分配,进而实现资源的调度;对上呈现标准的NVMe控制器,并从PCIe Function级别实现资源访问的隔离及NameSpace的动态映射。

[0183] 更具体地,首先在初始化阶段,在系统内部,构建PCIe Function与Name Space的映射关系,并依据默认规则,对每个NameSpace分配默认的NVMe Queue资源;在运行阶段,内部处理Host端下发的PRP请求,并根据映射关系实现对实际NVMe SSD设备的Memory访问及请求下发;同时,通过每个应用程序附带的Tag,动态调整每个NameSpace对应的虚拟软件队列数以及物理硬件队列数,提供了基于优先级、时间片、拥塞时间等各个角度的调度及仲裁机制,确保不出现长时间堵塞或饿死的情况出现。最后,在与Host应用程序访问时,通过PCIe Function配合IOMMU,实现各个NameSpace对应的PCIe资源完全隔离,确保虚拟化访问及多租户场景下的隔离性及安全性。

[0184] IO QoS调度过程中,首先利用目标Host对IO请求的优先级Tag信息,对相应的仲裁值作加权规整,得到目标Host的优先调度级别;然后在运行过程中,根据虚拟软件队列的占用负载及申请情况,动态调整与物理队列的映射关系,保证在高优先级任务进行的同时,不出现其他Host长时间等待的问题。

[0185] 下面继续说明本申请实施例提供的磁盘读写资源的分配装置354实施为软件模块的示例性结构,在一些实施例中,如图3所示,存储在存储器350的磁盘读写资源的分配装置354中的软件模块可以是主节点21中的磁盘读写资源的分配装置,包括:

[0186] 接收模块3541,用于接收主机发送的读写请求队列,其中,所述读写请求队列中包括所述主机的主机标识;

[0187] 第一确定模块3542,用于根据所述主机标识和所述主机与虚拟设备之间的第一映射关系,确定出与所述主机对应的目标虚拟设备的设备标识;第二确定模块3543,用于根据所述设备标识、和所述虚拟设备与预设命名空间之间的第二映射关系,确定出与所述主机对应的目标命名空间的命名空间标识;第三确定模块3544,用于根据所述命名空间标识、和所述预设命名空间与磁盘的磁盘分区之间的第三映射关系,确定出与所述主机对应的目标磁盘分区的分区标识;分配模块3545,用于将所述分区标识对应的目标磁盘分区分配给所述主机,通

过所述目标磁盘分区对所述读写请求队列中的读写请求进行响应。

[0188] 在一些实施例中,所述接收模块还用于:接收主机集群中任一主机发送的至少一条读写请求;将所述至少一条读写请求写入至预设的提交队列或完成队列中,形成所述读写请求队列。

[0189] 在一些实施例中,所述装置还包括:资源隔离处理模块,用于在接收主机发送的读写请求队列之前,采用至少一个所述预设命名空间,对待池化的磁盘阵列中的每一所述磁盘的磁盘资源进行资源隔离处理,得到隔离后的多个磁盘分区,其中,每一磁盘分区与一个所述预设命名空间之间具有所述第三映射关系;资源映射处理模块,用于将具有所述第三映射关系的所述磁盘分区和所述预设命名空间,与虚拟设备集合中的一个虚拟设备进行资源映射处理,以得到所述虚拟设备与所述磁盘分区和所述预设命名空间之间的所述第二映射关系。

[0190] 在一些实施例中,所述装置还包括:设置模块,用于预先设置主机集群中的每一所述主机,与所述虚拟设备集合中的至少一个虚拟设备之间的所述第一映射关系。

[0191] 在一些实施例中,所述资源隔离处理模块还用于:采用每一所述预设命名空间,对每一所述磁盘中的连续地址对应的连续磁盘资源进行资源隔离处理,得到隔离后的多个磁盘分区;其中,每一所述磁盘分区的磁盘资源对应连续的地址,且每一所述磁盘分区的容量与其他磁盘分区的容量相同或不同。

[0192] 在一些实施例中,在所述磁盘阵列中的多个磁盘中,每一所述预设命名空间与每一所述磁盘中的一个磁盘分区之间具有所述第三映射关系;所述装置还包括:第一优先级确定模块,用于对于每一预设命名空间对应的任一磁盘的磁盘分区,根据所述磁盘分区的读写响应速度或所述磁盘分区的容量,确定出所述磁盘分区的第一优先级;优先级获取模块,用于获取所述主机发送的所述读写请求队列的第二优先级;第一匹配模块,用于根据所述第二优先级和每一所述磁盘分区的第一优先级,在所述多个磁盘中匹配出一个磁盘,并将所匹配出的磁盘中的所述磁盘分区确定为所述目标磁盘分区。

[0193] 在一些实施例中,在所述磁盘阵列中的多个磁盘中,每一所述预设命名空间与每一所述磁盘中的一个磁盘分区之间具有所述第三映射关系;所述装置还包括:添加模块,用于为所述主机添加初始化标签,所述初始化标签用于标识所述主机的读写请求队列对应的所述目标磁盘分区的目标容量;第二匹配模块,用于根据所述目标容量,在所述多个磁盘中匹配出一个或多个磁盘;目标磁盘分区确定模块,用于将所匹配出的一个或多个磁盘中的所述磁盘分区,确定为所述目标磁盘分区。

[0194] 在一些实施例中,所述第三确定模块还用于:确定所述目标虚拟设备的页面偏移和所述空间标识对应的所述目标命名空间的地址;将所述页面偏移对应的偏移值,确定为所述目标虚拟设备的地址相对于所述目标命名空间的地址的偏移量;将相对于所述目标命名空间的地址,偏移所述偏移量之后的偏移地址,确定为与所述主机对应的目标磁盘分区的地址;根据所述目标磁盘分区的地址,确定所述目标磁盘分区的分区标识。

[0195] 在一些实施例中,所述装置还包括:权重确定模块,用于确定所述主机的当前的读写请求队列对应的资源分配权重;所述第三确定模块还用于:根据所述空间标识和所述资源分配权重,在磁盘阵列中的多个磁盘中,确定出一个或多个磁盘为目标磁盘;在所匹配出的目标磁盘中,将与第三映射关系对应的磁盘分区确定为所述目标磁盘分区。

[0196] 在一些实施例中,所述权重确定模块还用于:在对所述主机的资源分配权重进行调整之后,确定所述主机的当前读写请求量和所述目标虚拟设备中处于等待状态的待处理读写请求量;根据所述当前读写请求量和所述待处理读写请求量,确定所述主机的当前的读写请求队列对应的资源分配权重。

[0197] 在一些实施例中,所述权重确定模块还用于:确定历史周期内对所述读写请求进行响应的延迟时间的第一均值、和当前周期内对所述读写请求进行响应的延迟时间的第二均值;如果所述第二均值大于或等于所述第一均值,增大所述主机的当前的读写请求队列对应的资源分配权重;如果所述第二均值小于所述第一均值,减小所述主机的当前的读写请求队列对应的资源分配权重。

[0198] 在一些实施例中,所述权重确定模块还用于:确定所述主机的读写请求队列的延迟敏感类型;如果所述读写请求队列为尾部延迟敏感型队列,增大所述主机的当前的读写请求队列对应的资源分配权重,且增大后的所述资源分配权重是当前的磁盘读写资源的分配系统中全部读写请求队列中的最大权重。

[0199] 需要说明的是,本申请实施例装置的描述,与上述方法实施例的描述是类似的,具有同方法实施例相似的有益效果,因此不做赘述。对于本装置实施例中未披露的技术细节,请参照本申请方法实施例的描述而理解。

[0200] 本申请实施例提供了一种计算机程序产品或计算机程序,该计算机程序产品或计算机程序包括计算机指令,该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令,处理器执行该计算机指令,使得该计算机设备执行本申请实施例上述的方法。

[0201] 本申请实施例提供一种存储有可执行指令的存储介质,其中存储有可执行指令,当可执行指令被处理器执行时,将引起处理器执行本申请实施例提供的方法,例如,如图4示出的方法。

[0202] 在一些实施例中,存储介质可以是计算机可读存储介质,例如,铁电存储器(FRAM, Ferromagnetic Random Access Memory)、只读存储器(ROM, Read Only Memory)、可编程只读存储器(PROM, Programmable Read Only Memory)、可擦除可编程只读存储器(EPROM, Erasable Programmable Read Only Memory)、带电可擦可编程只读存储器(EEPROM, Electrically Erasable Programmable Read Only Memory)、闪存、磁表面存储器、光盘、或光盘只读存储器(CD-ROM, Compact Disk-Read Only Memory)等存储器;也可以是包括上述存储器之一或任意组合的各种设备。

[0203] 在一些实施例中,可执行指令可以采用程序、软件、软件模块、脚本或代码的形式,按任意形式的编程语言(包括编译或解释语言,或者声明性或过程性语言)来编写,并且其可按任意形式部署,包括被部署为独立的程序或者被部署为模块、组件、子例程或者适合在计算环境中使用的其它单元。

[0204] 作为示例,可执行指令可以但不一定对应于文件系统中的文件,可以可被存储在保存其它程序或数据的文件的一部分,例如,存储在超文本标记语言(H TML, Hyper Text Markup Language)文档中的一个或多个脚本中,存储在专用于所讨论的程序的单个文件中,或者,存储在多个协同文件(例如,存储一个或多个模块、子程序或代码部分的文件)中。作为示例,可执行指令可被部署为在一个计算设备上执行,或者在位于一个地点的多个计

算设备上执行,又或者,在分布在多个地点且通过通信网络互连的多个计算设备上执行。

[0205] 以上所述,仅为本申请的实施例而已,并非用于限定本申请的保护范围。凡在本申请的精神和范围之内所作的任何修改、等同替换和改进等,均包含在本申请的保护范围之内。

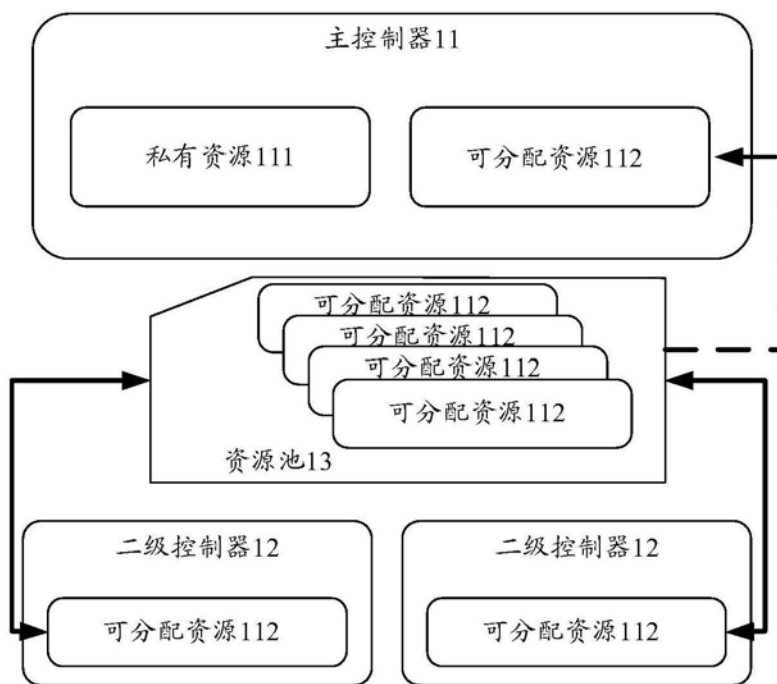


图1

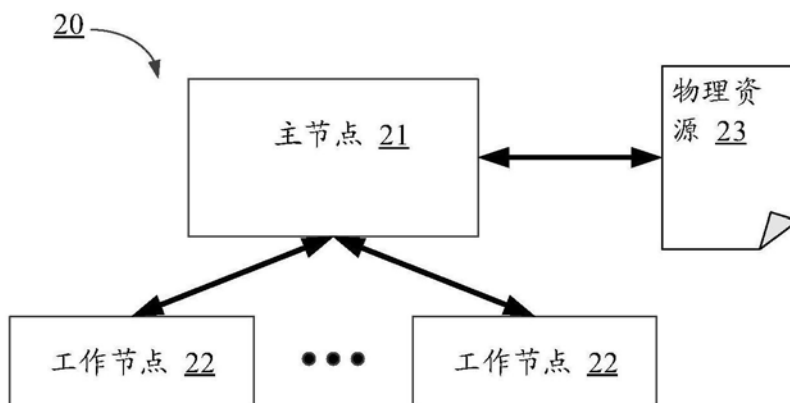


图2A

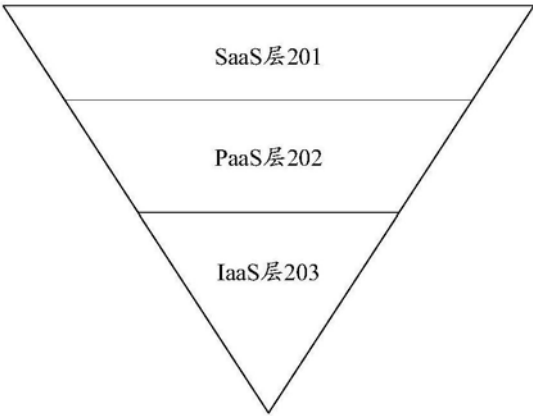


图2B

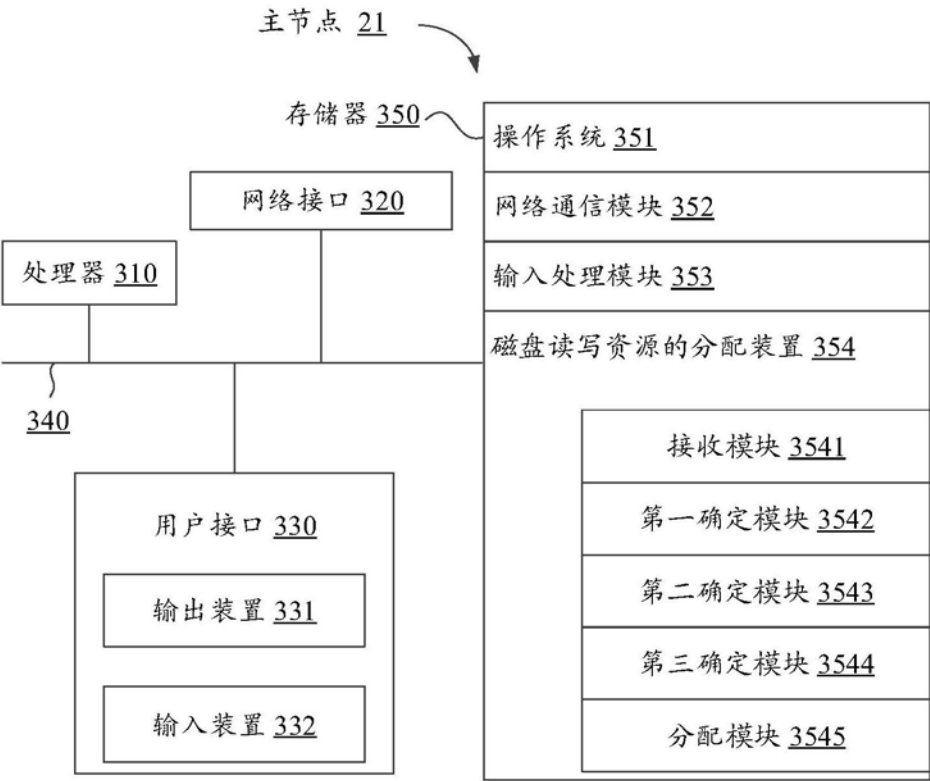


图3

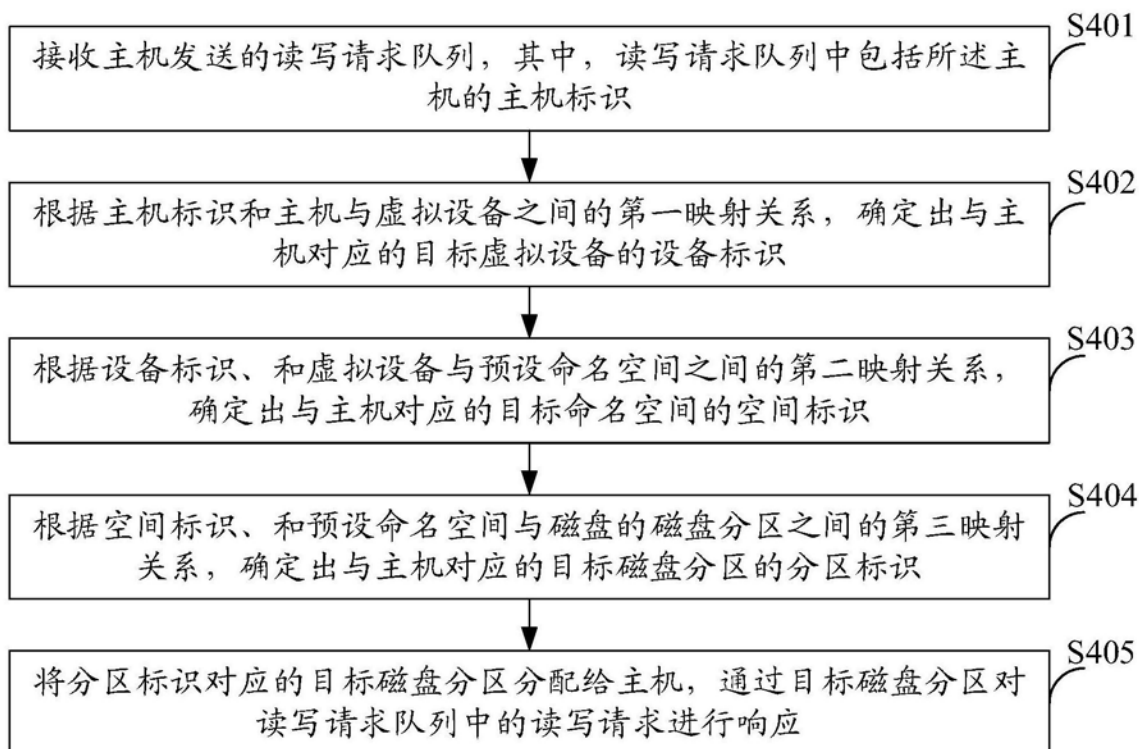


图4

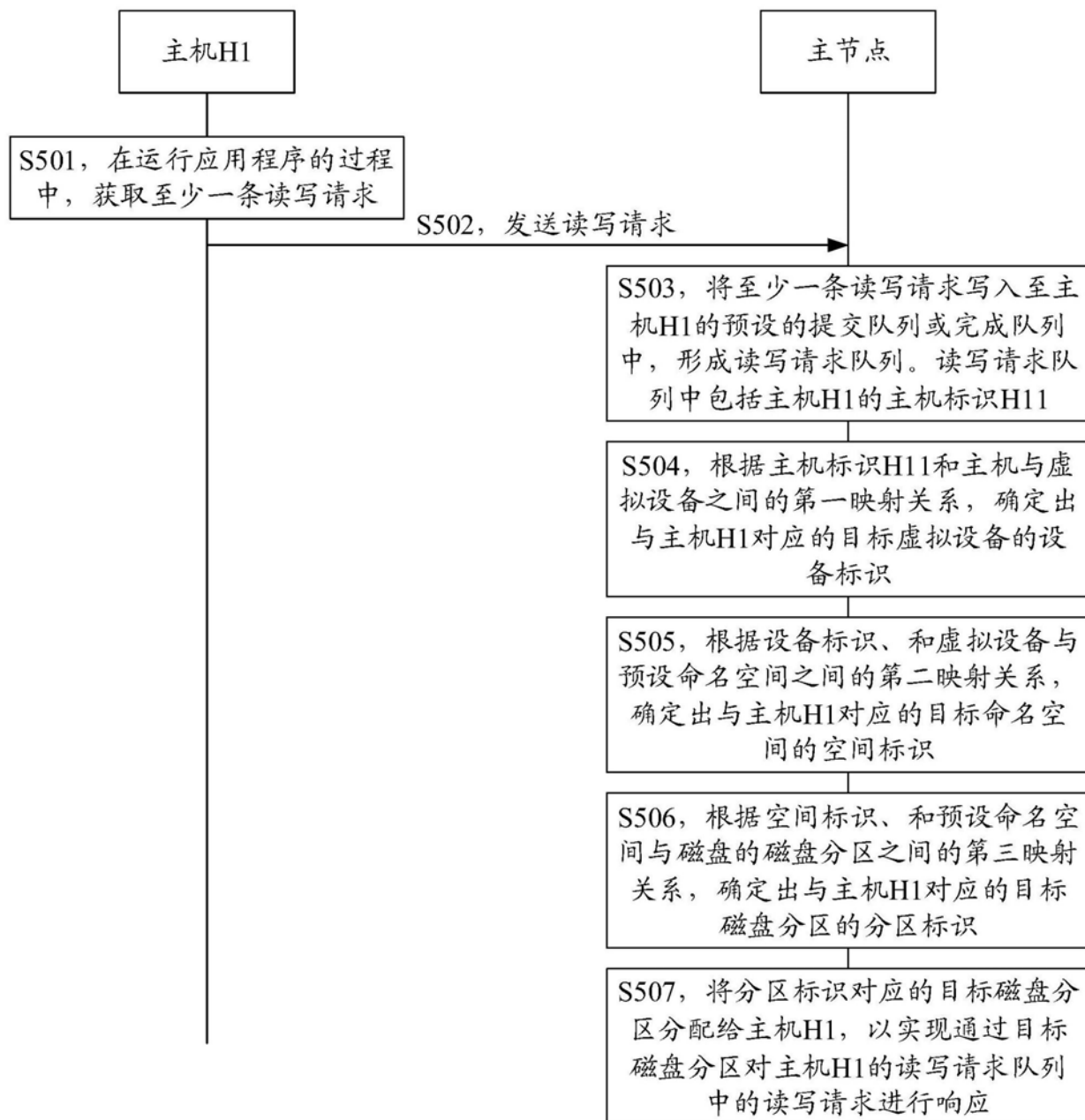


图5

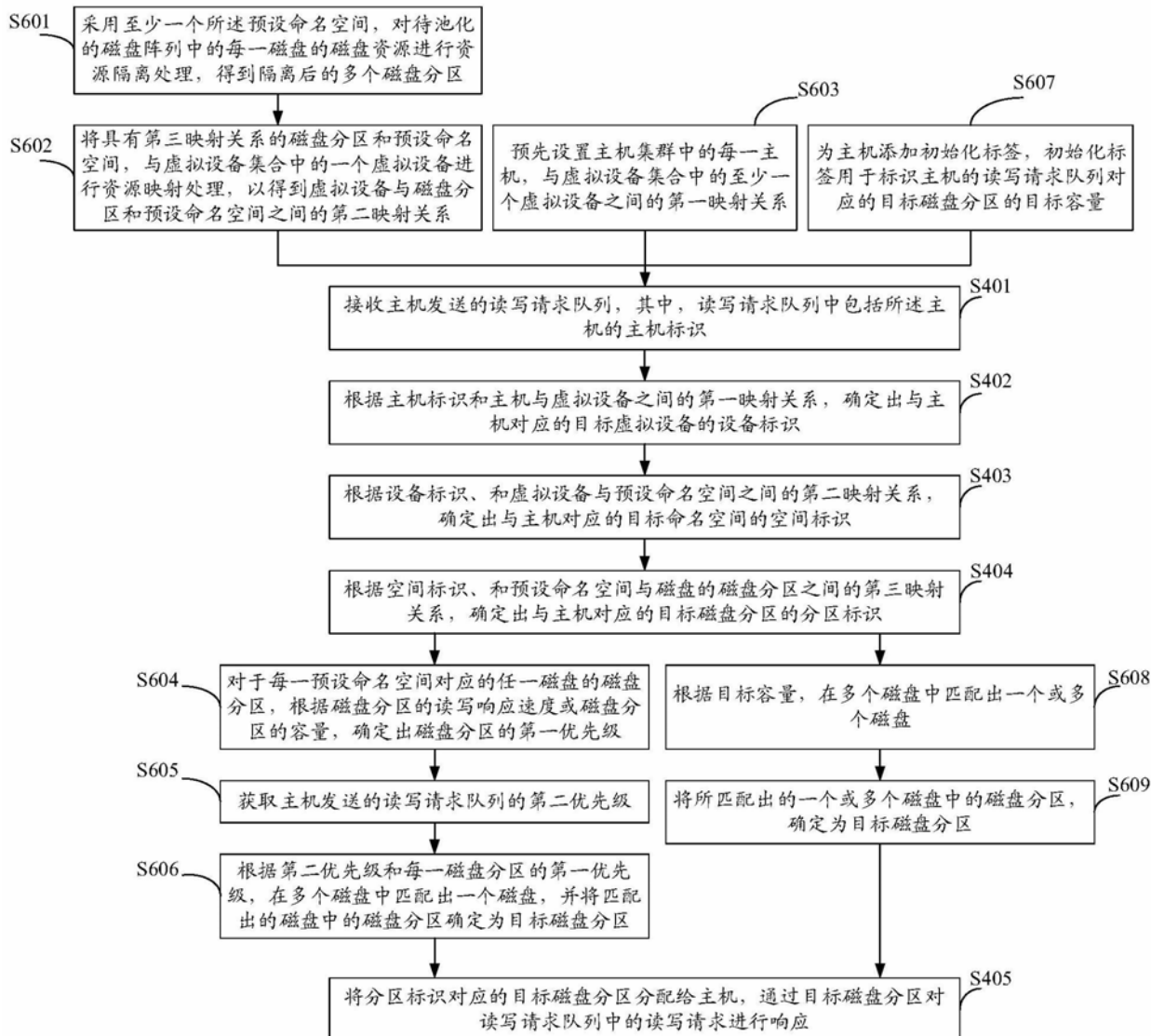


图6

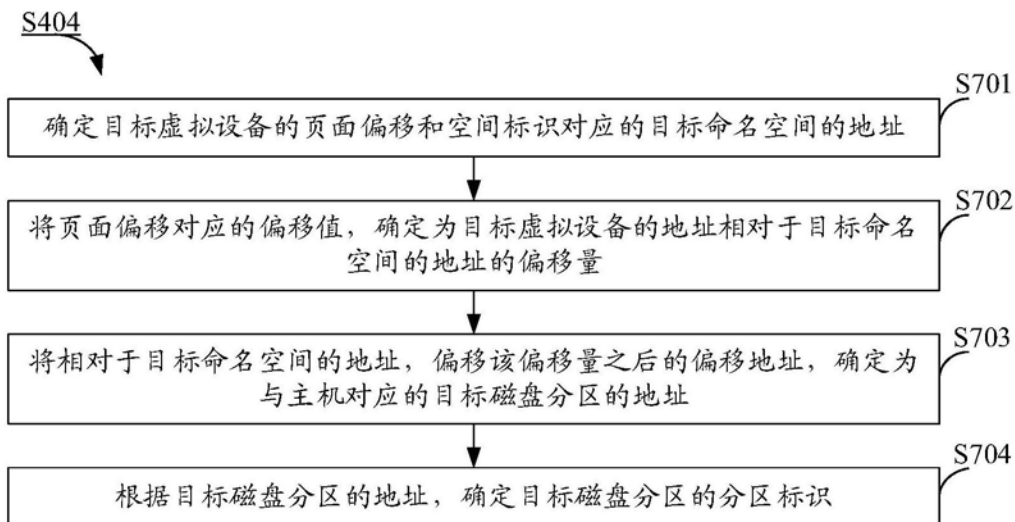


图7

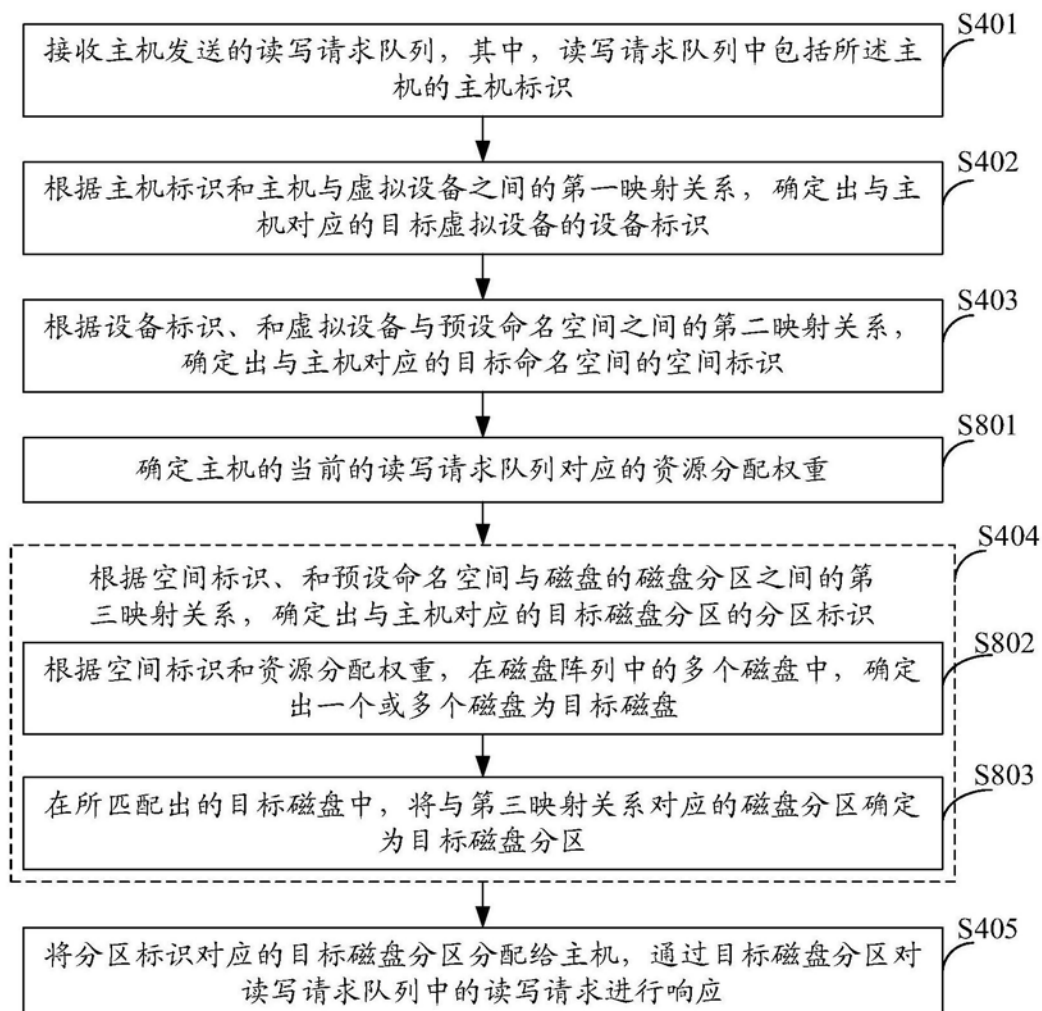


图8

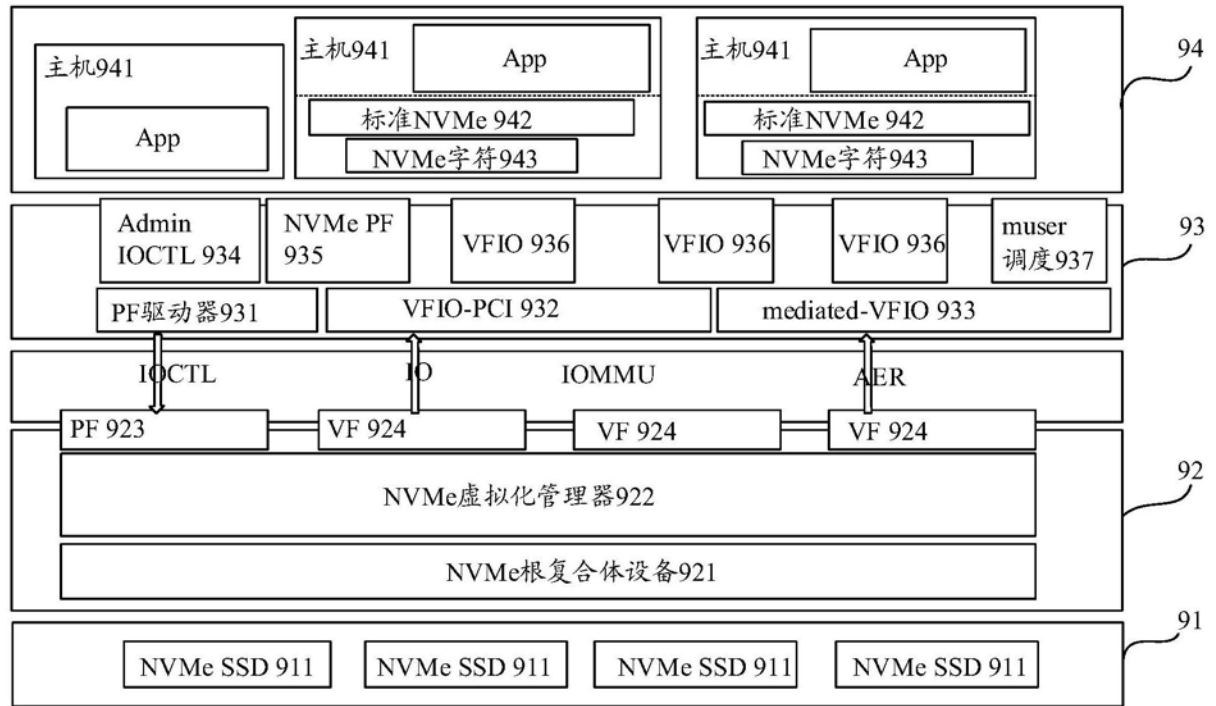


图9

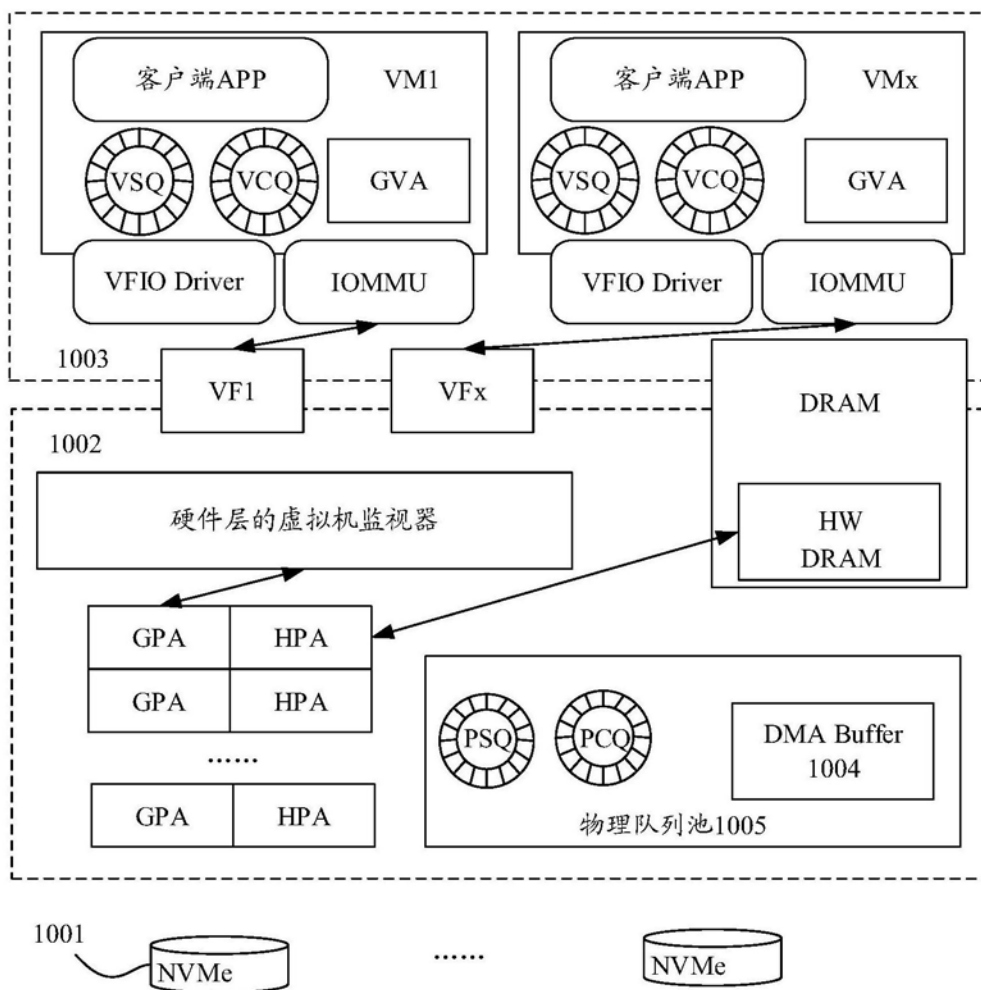


图10

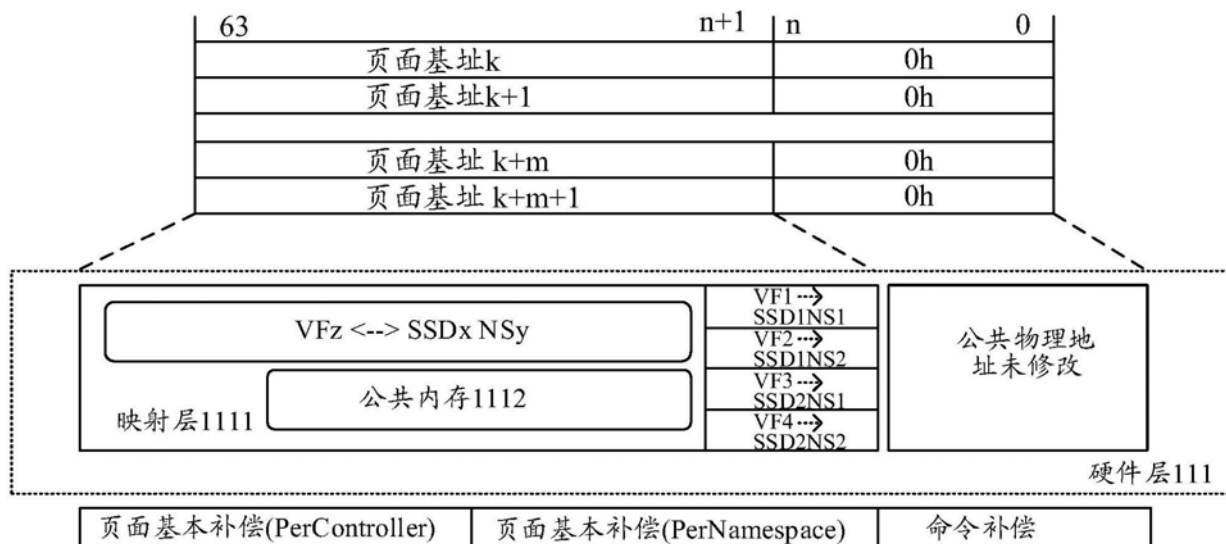


图11

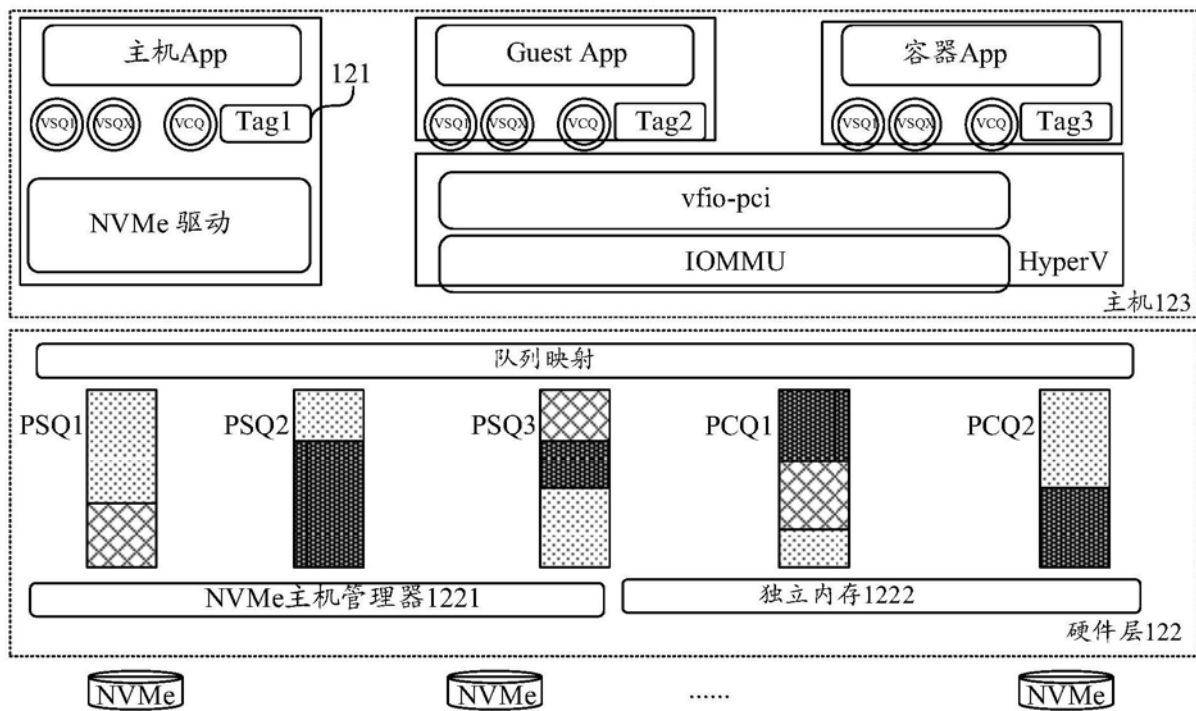


图12

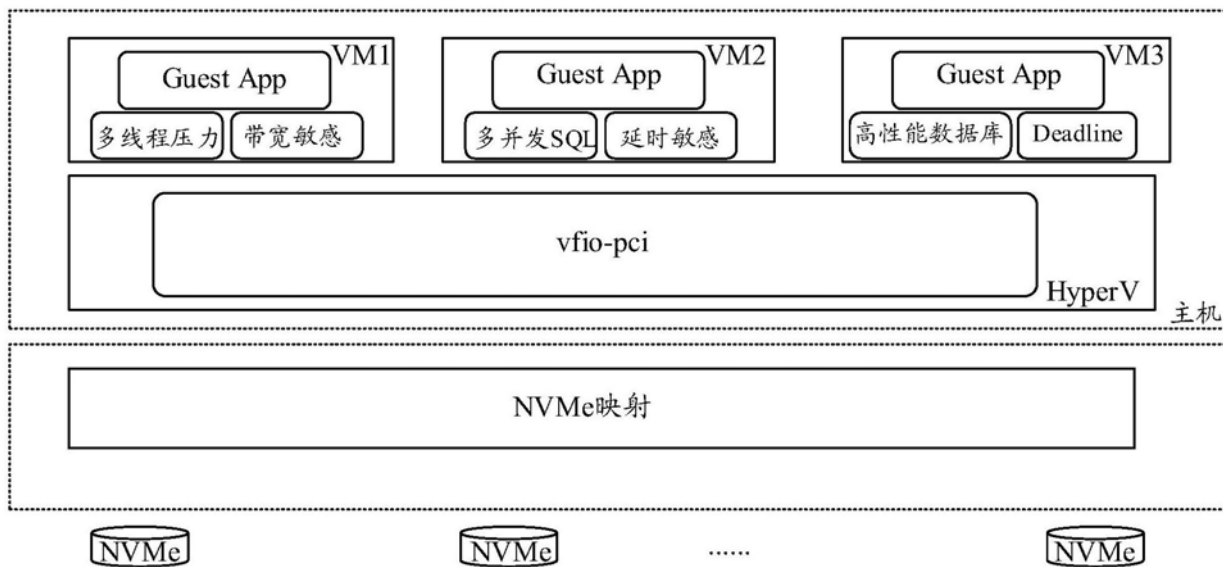


图13