



(12) 发明专利申请

(10) 申请公布号 CN 116166203 A

(43) 申请公布日 2023.05.26

(21) 申请号 202310419925.9

(22) 申请日 2023.04.19

(71) 申请人 苏州浪潮智能科技有限公司

地址 215000 江苏省苏州市吴中经济开发区郭巷街道官浦路1号9幢

(72) 发明人 李飞龙 王见 孙明刚

(74) 专利代理机构 北京连和连知识产权代理有限公司 11278

专利代理师 张元 马鹏林

(51) Int. Cl.

G06F 3/06 (2006.01)

权利要求书3页 说明书16页 附图6页

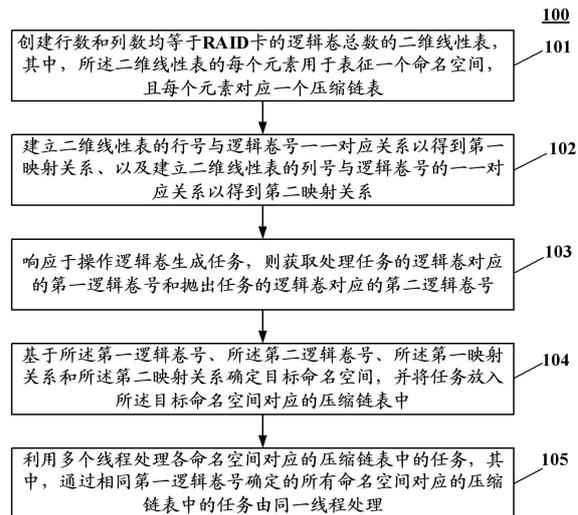
(54) 发明名称

一种RAID卡的命名空间管理方法、装置、设备及介质

(57) 摘要

本发明涉及存储技术领域,尤其涉及一种RAID卡的命名空间管理方法、装置、设备及介质。所述方法包括:创建行数和列数均等于逻辑卷总数的二维线性表,表的每个元素用于表征命名空间且对应压缩链表;建立表的行号、列号分别与逻辑卷号对应关系以得到第一映射关系和第二映射关系;响应于操作逻辑卷生成任务则获取处理任务的第一逻辑卷号和抛出任务的第二逻辑卷号;基于第一逻辑卷号、第二逻辑卷号、第一映射关系和第二映射关系确定目标命名空间并将任务放入目标命名空间对应的压缩链表中;利用多个线程处理各命名空间对应的压缩链表中的任务,相同第一逻辑卷号确定的压缩链表中的任务由同一线程处理。本发明的方案显著提命名空间管理效率。

CN 116166203 A



1. 一种RAID卡的命名空间管理方法,其特征在于,所述方法包括:

创建行数和列数均等于RAID卡的逻辑卷总数的二维线性表,其中,所述二维线性表的每个元素用于表征一个命名空间,且每个元素对应一个压缩链表;

建立二维线性表的行号与逻辑卷号一一对应关系以得到第一映射关系、以及建立二维线性表的列号与逻辑卷号的一一对应关系以得到第二映射关系;

响应于操作逻辑卷生成任务,则获取处理任务的逻辑卷对应的第一逻辑卷号和抛出任务的逻辑卷对应的第二逻辑卷号;

基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间,并将任务放入所述目标命名空间对应的压缩链表中;

利用多个线程处理各命名空间对应的压缩链表中的任务,其中,通过相同第一逻辑卷号确定的所有命名空间对应的压缩链表中的任务由同一线程处理。

2. 根据权利要求1所述的RAID卡的命名空间管理方法,其特征在于,所述基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间的步骤包括:

将第一逻辑卷号和所述第一映射关系进行匹配以确定目标行号;

将第二逻辑卷号和所述第二映射关系进行匹配以确定目标列号;

将二维线性表中行号等于目标行号且列号等于目标列列号的元素作为目标命名空间。

3. 根据权利要求2所述的RAID卡的命名空间管理方法,其特征在于,线程总数等于RAID卡的逻辑卷总数;

所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤包括:

建立线程与二维线性链表的行的一一对应关系以得到第三映射关系;

响应于某一命名空间对应的压缩链表中存在需要处理的任务,则将某一命名空间所在行号与所述第三映射关系进行匹配以得到目标线程;

由所述目标线程处理所述某一命名空间对应的压缩链表中的任务。

4. 根据权利要求1所述的RAID卡的命名空间管理方法,其特征在于,所述基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间的步骤包括:

将第一逻辑卷号和所述第二映射关系进行匹配以确定目标列号;

将第二逻辑卷号和所述第一映射关系进行匹配以确定目标行号;

将二维线性表中行号等于目标行号且列号等于目标列列号的元素作为目标命名空间。

5. 根据权利要求4所述的RAID卡的命名空间管理方法,其特征在于,线程总数等于RAID卡的逻辑卷总数;

所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤包括:

建立线程与二维线性链表的列的一一对应关系以得到第四映射关系;

响应于某一命名空间对应的压缩链表中存在需要处理的任务,则将某一命名空间所在列号与所述第四映射关系进行匹配以得到目标线程;

由所述目标线程处理所述某一命名空间对应的压缩链表中的任务。

6. 根据权利要求3或5所述的RAID卡的命名空间管理方法,其特征在于,所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤还包括:

响应于同一线程对应的多个压缩链表中均存在任务需要处理,则采用轮询的机制处理多个压缩链表的任务。

7.根据权利要求1所述的RAID卡的命名空间管理方法,其特征在于,所述方法还包括:将多个线程均分到RAID卡的多个控制器上执行。

8.根据权利要求1所述的RAID卡的命名空间管理方法,其特征在于,各命名空间对应的压缩链表在二维线性链表创建时同时创建;或者

各命名空间对应的压缩链表在首次放入任务时创建。

9.根据权利要求1所述的RAID卡的命名空间管理方法,其特征在于,所述压缩链表包括:

环形队列,所述环形队列用于采用连续内存空间存放预设数量的任务;

第一指针,所述第一指针用于指向环形队列最新放入的任务;

第二指针,所述第二指针用于指向环形队列当前需要处理的任務。

10.根据权利要求9所述的RAID卡的命名空间管理方法,其特征在于,所述压缩链表还包括:

第三指针,所述第三指针用于指向溢出任务链表的首元素的链表结构体;

第四指针,所述第四指针用于指向溢出任务链表的尾元素的链表结构体。

11.根据权利要求9所述的RAID卡的命名空间管理方法,其特征在于,所述预设数量等于128。

12.根据权利要求10所述的RAID卡的命名空间管理方法,其特征在于,所述链表结构体包括:

任务类型字段,所述任务类型字段用于记录任务类型;

第五指针,所述第五指针用于指向下一个任务;

任务执行函数字段,所述任务执行函数字段用于记录与任务类型对应的执行函数。

13.根据权利要求10所述的RAID卡的命名空间管理方法,其特征在于,所述将任务放入所述目标命名空间对应的压缩链表中的步骤包括:

判断环形队列是否已满;

若环形队列未满,则将产生的任务放入环形队列;

若环形队列已满,则将产生的任务放入溢出任务链表。

14.根据权利要求13所述的RAID卡的命名空间管理方法,其特征在于,所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤包括:

响应于某一线程处理某一压缩链表中的任务时若环形队列已满,则所述某一线程优先处理溢出任务链表中的任务。

15.根据权利要求13所述的RAID卡的命名空间管理方法,其特征在于,任意线程处理环形队列中任务时均遵循先进先出原则。

16.根据权利要求13所述的RAID卡的命名空间管理方法,其特征在于,任意线程处理溢出任务链表中任务时均遵循后进先出原则。

17.根据权利要求1所述的RAID卡的命名空间管理方法,其特征在于,逻辑卷的任务包括卷的格式化、卷的在线重构、卷的扩容、卷的缩容。

18.一种RAID卡的命名空间管理装置,其特征在于,所述装置包括:

创建模块,配置用于创建行数和列数均等于RAID卡的逻辑卷总数的二维线性表,其中,所述二维线性表的每个元素用于表征一个命名空间,且每个元素对应一个压缩链表;

映射模块,配置用于建立二维线性表的行号与逻辑卷号一一对应关系以得到第一映射关系、以及建立二维线性表的列号与逻辑卷号的一一对应关系以得到第二映射关系;

获取模块,配置用于响应于操作逻辑卷生成任务,则获取处理任务的逻辑卷对应的第一逻辑卷号和抛出任务的逻辑卷对应的第二逻辑卷号;

确定模块,配置用于基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间,并将任务放入所述目标命名空间对应的压缩链表中;

处理模块,配置用于利用多个线程处理各命名空间对应的压缩链表中的任务,其中,通过相同第一逻辑卷号确定的所有命名空间对应的压缩链表中的任务由同一线程处理。

19. 一种计算机设备,其特征在于,包括:

至少一个处理器;以及

存储器,所述存储器存储有可在所述处理器中运行的计算机程序,所述处理器执行所述程序时执行权利要求1-17任意一项所述的方法。

20. 一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,其特征在于,所述计算机程序被处理器执行时执行权利要求1-17任意一项所述的方法。

一种RAID卡的命名空间管理方法、装置、设备及介质

技术领域

[0001] 本发明涉及存储技术领域,尤其涉及一种RAID卡的命名空间管理方法、装置、设备及介质。

背景技术

[0002] RAID卡由多个RAID阵列构成,RAID阵列(Redundant Array of Independent Disks,即独立磁盘的冗余阵列)是一种存储技术,是由很多块独立的磁盘组合成一个容量巨大的磁盘组,RAID是一种把多块独立的硬盘(物理硬盘)按不同的方式组合起来形成一个硬盘组(逻辑硬盘),从而提供比单个硬盘更高的存储性能和提供数据备份技术。为了进一步提高RAID卡的I/O(input/output,即输入/输出)速度,业界已使用固态硬盘取代机械硬盘来构成RAID阵列,在使用时RAID卡时用户在RAID卡上创建越来越多的命名空间(即Namespace)以供各类需求使用。

[0003] 目前,传统的命名空间管理大多通过一维线性表对Namespace进行维护,一维线性表常见形式有数组和链表两种数据结构;考虑到数组的元素个数是固定的,而组成链表的结点个数可按需要增减,因此为了线性表的灵活性,现有技术都采取链表的形式,而且链表内存不连续可以任意添加元素,但随着Namespace数量的增加,查找时间复杂度会越来越高,需要对链表中的每个Namespace元素依次进行查找。而且用户在RAID卡上创建越来越多的逻辑块设备(即RAID卡的逻辑卷volume),与之对应维护Namespace的链表长度随之增加,造成查找Namespace的效率会越来越低。此外,Namespace链表元素一旦被一个线程用锁锁定后,其他线程在访问该Namespace链表元素时需要等待锁释放,这样会对RAID卡读写数据性能产生负面影响。不仅如此,由于各个逻辑卷之间会相互抛出任务(例如卷的格式化任务、在线重构任务等等),而采用一维线性链表对逻辑卷进行管理,这样在逻辑卷之间相互抛出任务后,一维线性链表对逻辑卷相互间抛出的任务会更难以管理,很容易造成Namespace元素混乱。

发明内容

[0004] 有鉴于此,有必要针对以上技术问题,提供一种RAID卡的命名空间管理方法、装置、设备及介质。

[0005] 根据本发明的第一方面,提供了一种RAID卡的命名空间管理方法,所述方法包括:

创建行数和列数均等于RAID卡的逻辑卷总数的二维线性表,其中,所述二维线性表的每个元素用于表征一个命名空间,且每个元素对应一个压缩链表;

建立二维线性表的行号与逻辑卷号一一对应关系以得到第一映射关系、以及建立二维线性表的列号与逻辑卷号的一一对应关系以得到第二映射关系;

响应于操作逻辑卷生成任务,则获取处理任务的逻辑卷对应的第一逻辑卷号和抛出任务的逻辑卷对应的第二逻辑卷号;

基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射

关系确定目标命名空间,并将任务放入所述目标命名空间对应的压缩链表中;

利用多个线程处理各命名空间对应的压缩链表中的任务,其中,通过相同第一逻辑卷号确定的所有命名空间对应的压缩链表中的任务由同一线程处理。

[0006] 在一些实施例中,所述基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间的步骤包括:

将第一逻辑卷号和所述第一映射关系进行匹配以确定目标行号;

将第二逻辑卷号和所述第二映射关系进行匹配以确定目标列号;

将二维线性表中行号等于目标行号且列号等于目标列列号的元素作为目标命名空间。

[0007] 在一些实施例中,线程总数等于RAID卡的逻辑卷总数;

所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤包括:

建立线程与二维线性链表的行的一一对应关系以得到第三映射关系;

响应于某一命名空间对应的压缩链表中存在需要处理的任務,则将某一命名空间所在行号与所述第三映射关系进行匹配以得到目标线程;

由所述目标线程处理所述某一命名空间对应的压缩链表中的任务。

[0008] 在一些实施例中,所述基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间的步骤包括:

将第一逻辑卷号和所述第二映射关系进行匹配以确定目标列号;

将第二逻辑卷号和所述第一映射关系进行匹配以确定目标行号;

将二维线性表中行号等于目标行号且列号等于目标列列号的元素作为目标命名空间。

[0009] 在一些实施例中,线程总数等于RAID卡的逻辑卷总数;

所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤包括:

建立线程与二维线性链表的列的一一对应关系以得到第四映射关系;

响应于某一命名空间对应的压缩链表中存在需要处理的任務,则将某一命名空间所在列号与所述第四映射关系进行匹配以得到目标线程;

由所述目标线程处理所述某一命名空间对应的压缩链表中的任务。

[0010] 在一些实施例中,所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤还包括:

响应于同一线程对应的多个压缩链表中均存在任务需要处理,则采用轮询的机制处理多个压缩链表的任务。

[0011] 在一些实施例中,所述方法还包括:

将多个线程均分到RAID卡的多个控制器上执行。

[0012] 在一些实施例中,各命名空间对应的压缩链表在二维线性链表创建时同时创建;或者

各命名空间对应的压缩链表在首次放入任务时创建。

[0013] 在一些实施例中,所述压缩链表包括:

环形队列,所述环形队列用于采用连续内存空间存放预设数量的任务;

第一指针,所述第一指针用于指向环形队列最新放入的任务;

第二指针,所述第二指针用于指向环形队列当前需要处理的任务。

[0014] 在一些实施例中,所述压缩链表还包括:

第三指针,所述第三指针用于指向溢出任务链表的首元素的链表结构体;

第四指针,所述第四指针用于指向溢出任务链表的尾元素的链表结构体。

[0015] 在一些实施例中,所述预设数量等于128。

[0016] 在一些实施例中,所述链表结构体包括:

任务类型字段,所述任务类型字段用于记录任务类型;

第五指针,所述第五指针用于指向下一个任务;

任务执行函数字段,所述任务执行函数字段用于记录与任务类型对应的执行函数。

[0017] 在一些实施例中,所述将任务放入所述目标命名空间对应的压缩链表中的步骤包括:

判断环形队列是否已满;

若环形队列未满,则将产生的任务放入环形队列;

若环形队列已满,则将产生的任务放入溢出任务链表。

[0018] 在一些实施例中,所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤包括:

响应于某一线程处理某一压缩链表中的任务时若环形队列已满,则所述某一线程优先处理溢出任务链表中的任务。

[0019] 在一些实施例中,任意线程处理环形队列中任务时均遵循先进先出原则。

[0020] 在一些实施例中,任意线程处理溢出任务链表中任务时均遵循后进先出原则。

[0021] 在一些实施例中,逻辑卷的任务包括卷的格式化、卷的在线重构、卷的扩容、卷的缩容。

[0022] 根据本发明的第二方面,提供了一种RAID卡的命名空间管理装置,所述装置包括:

创建模块,配置用于创建行数和列数均等于RAID卡的逻辑卷总数的二维线性表,其中,所述二维线性表的每个元素用于表征一个命名空间,且每个元素对应一个压缩链表;

映射模块,配置用于建立二维线性表的行号与逻辑卷号一一对应关系以得到第一映射关系、以及建立二维线性表的列号与逻辑卷号的一一对应关系以得到第二映射关系;

获取模块,配置用于响应于操作逻辑卷生成任务,则获取处理任务的逻辑卷对应的第一逻辑卷号和抛出任务的逻辑卷对应的第二逻辑卷号;

确定模块,配置用于基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间,并将任务放入所述目标命名空间对应的压缩链表中;

处理模块,配置用于利用多个线程处理各命名空间对应的压缩链表中的任务,其中,通过相同第一逻辑卷号确定的所有命名空间对应的压缩链表中的任务由同一线程处理。

[0023] 根据本发明的第三方面,还提供了一种计算机设备,该计算机设备包括:

至少一个处理器;以及

存储器,存储器存储有可在处理器上运行的计算机程序,处理器执行程序时执行

前述的RAID卡的命名空间管理方法。

[0024] 根据本发明的第四方面,还提供了一种计算机可读存储介质,计算机可读存储介质存储有计算机程序,计算机程序被处理器执行时执行前述的RAID卡的命名空间管理方法。

[0025] 上述一种RAID卡的命名空间管理方法,通过一个行数和列数均等于RAID卡的逻辑卷总数的二维线性表维护RAID卡的命名空间,通过建立行号、列号分别与逻辑卷号的对应关系,当操作逻辑卷生成任务时,通过任务的处理器和抛出者使用创建的对应关系找到二维线性链表对应位置的命名空间,然后将任务放入到与命名空间对应的压缩链表中,最后对于多个线程处理压缩链表中的任务,在处理任务时相同任务处理者的任务都由同一个线程处理,可以实现对命名空间的彻底免锁,不需等待其他线程对锁的释放,显著提升命名空间的查找和管理效率,有助于提高RAID卡的I/O性能。

[0026] 此外,本发明还提供了一种RAID卡的命名空间管理装置、一种计算机设备和一种计算机可读存储介质,同样能实现上述技术效果,这里不再赘述。

附图说明

[0027] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,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实施例。

[0028] 图1为本发明一个实施例提供的一种RAID卡的命名空间管理方法的流程图;

图2为本发明一个实施例提供的RAID卡中命名空间、卷和控制器的关系框图;

图3为本发明一个实施例提供的二维线性链表的结构示意图;

图4为本发明一个实施例提供的压缩链表与二维线性链表的关系图;

图5为本发明一个实施例提供的压缩链表详细字段示意图;

图6为本发明一个实施例提供的压缩链表中维护溢出环形阵列的任务的链表结构体详细字段示意图;

图7为本发明另一个实施例提供的一种RAID卡的命名空间管理装置的结构示意图;

图8为本发明另一个实施例中计算机设备的内部结构图。

具体实施方式

[0029] 为使本发明的目的、技术方案和优点更加清楚明白,以下结合具体实施例,并参照附图,对本发明实施例进一步详细说明。

[0030] 需要说明的是,本发明实施例中所有使用“第一”和“第二”的表述均是为了区分两个相同名称非相同的实体或者非相同的参量,可见“第一”“第二”仅为了表述的方便,不应理解为对本发明实施例的限定,后续实施例对此不再一一说明。

[0031] 在一个实施例中,请参照图1所示,本发明提供了一种RAID卡的命名空间管理方法100,具体来说,所述方法包括以下步骤:

步骤101,创建行数和列数均等于RAID卡的逻辑卷总数的二维线性表,其中,所述

二维线性表的每个元素用于表征一个命名空间,且每个元素对应一个压缩链表;

在本实施例中,一维线性表是将数据排成像一条长线一样的结构,一维线性表上的每个数据最多只有前后两个方向,除了第一个和最后一个数据元素之外,其它数据元素都是首尾相接的。二维线性链表指多个一维线性链表以行或列的方式构成的,例如由多行一维线性链表组成一个二维线性链表或者由多列一维线性链表构成一个二维线性链表。二维线性表可以表示为元素NS,行号指的是元素在二维线性表中所在的行值,列号指的是元素在二维线性表中所在的行值。

[0032] 步骤102,建立二维线性表的行号与逻辑卷号一一对应关系以得到第一映射关系、以及建立二维线性表的列号与逻辑卷号的一一对应关系以得到第二映射关系;

在本实施例中,逻辑卷又称Volume,逻辑卷作为容量消费者在RAID阵列中申请全部或者部分容量来创建卷,并映射给主机作为逻辑块设备使用,逻辑卷号是指逻辑卷对应的编号,例如逻辑卷0、逻辑卷1……,通常用户在创建逻辑卷后为了便于区别不同的逻辑卷,通常会采用数字或字符对逻辑卷进行顺序编号,例如,可以1为步长从数字零开始编号,假设有五个卷即可得到逻辑卷0至逻辑卷4,当然逻辑卷号也可以从1开始编号或者以其他步长为间隔进行编号,本实施例不对逻辑卷号的编排规则进行限定。第一映射关系表征的每个行号对应每个逻辑卷号的一一对应关系,例如第一行对应逻辑卷0,第二行对应逻辑卷1,等等以此类推;第二映射关系表征的每个列号对应每个逻辑卷号的一一对应关系,例如第一列对应逻辑卷0,第2列对应逻辑卷1依次类推。

[0033] 需要说明的是,行号由小到大可以按照逻辑卷号由小到大顺序对应映射,当然也可以采用乱序方式映射,同样列号由小到大可以按照逻辑卷号由小到大顺序对应映射,当然也可以采用乱序方式映射。

[0034] 步骤103,响应于操作逻辑卷生成任务,则获取处理任务的逻辑卷对应的第一逻辑卷号和抛出任务的逻辑卷对应的第二逻辑卷号;

在本实施例中,操作逻辑卷生成的任务可以是任何逻辑卷投给自己的任务,也可以是其他逻辑卷抛给其的任务,例如逻辑卷0可以给自己抛任务,逻辑卷4也可以给逻辑卷0抛任务。

[0035] 步骤104,基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间,并将任务放入所述目标命名空间对应的压缩链表中;

在本实施例中,可以使用二维线性变的一行元素作为处理相同逻辑卷任务的一组命名空间,相应的将一系列元素作为抛出逻辑卷任务的一组命名空间,此时第一逻辑卷号需要使用第一映射表确定其在二维线性链表中的位置,第二逻辑卷号需要使用第二映射表确定其在二维线性链表中的位置;此外也可以使用二维线性变的一列元素作为处理相同逻辑卷任务的一组命名空间,相应的将一行元素作为抛出逻辑卷任务的一组命名空间,此时第一逻辑卷号需要使用第二映射表确定其在二维线性链表中的位置,第二逻辑卷号需要使用第一映射表确定其在二维线性链表中的位置。

[0036] 步骤105,利用多个线程处理各命名空间对应的压缩链表中的任务,其中,通过相同第一逻辑卷号确定的所有命名空间对应的压缩链表中的任务由同一线程处理。

[0037] 在本实施例中,相同第一逻辑卷号确定的命名空间指的是二维线性链表中处理任务的逻辑卷对应的逻辑卷号相同元素。

[0038] 本实施例的一种RAID卡的命名空间管理方法,通过一个行数和列数均等于RAID卡的逻辑卷总数的二维线性表维护RAID卡的命名空间,通过建立行号、列号分别与逻辑卷号的对应关系,当操作逻辑卷生成任务时,通过任务的处理器和抛出者使用创建的对应关系找到二维线性链表对应位置的命名空间,然后将任务放入到与命名空间对应的压缩链表中,最后由多个线程处理压缩链表中的任务,在处理任务时相同任务处理者的任务都由同一个线程处理,可以实现对命名空间的彻底免锁,不需等待其他线程对锁的释放,显著提升命名空间的查找和管理效率,有助于提高RAID卡的I/O性能。

[0039] 在一些实施例中,前述步骤104,所述基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间的步骤包括:

将第一逻辑卷号和所述第一映射关系进行匹配以确定目标行号;

将第二逻辑卷号和所述第二映射关系进行匹配以确定目标列号;

将二维线性表中行号等于目标行号且列号等于目标列列号的元素作为目标命名空间。

[0040] 在一些实施例中,线程总数等于RAID卡的逻辑卷总数;

前述步骤105,所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤包括:

建立线程与二维线性链表的行的一一对应关系以得到第三映射关系;

响应于某一命名空间对应的压缩链表中存在需要处理的任務,则将某一命名空间所在行号与所述第三映射关系进行匹配以得到目标线程;

由所述目标线程处理所述某一命名空间对应的压缩链表中的任务。

[0041] 在一些实施例中,前述步骤104,所述基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间的步骤包括:

将第一逻辑卷号和所述第二映射关系进行匹配以确定目标列号;

将第二逻辑卷号和所述第一映射关系进行匹配以确定目标行号;

将二维线性表中行号等于目标行号且列号等于目标列列号的元素作为目标命名空间。

[0042] 在一些实施例中,线程总数等于RAID卡的逻辑卷总数;

前述步骤105,所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤包括:

建立线程与二维线性链表的列的一一对应关系以得到第四映射关系;

响应于某一命名空间对应的压缩链表中存在需要处理的任務,则将某一命名空间所在列号与所述第四映射关系进行匹配以得到目标线程;

由所述目标线程处理所述某一命名空间对应的压缩链表中的任务。

[0043] 在一些实施例中,前述步骤105,所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤还包括:

响应于同一线程对应的多个压缩链表中均存在任务需要处理,则采用轮询的机制处理多个压缩链表的任务。

[0044] 在一些实施中,所述方法还包括:

将多个线程均分到RAID卡的多个控制器上执行。

[0045] 在本实施例中,均分线程到RAID卡的多个控制器指的是让每个控制维护的线程数量相同,假设有十二个线程,而RAID卡有四个控制器,此时每个控制器负责三个线程,由此可以保证RAID卡每个控制器的负载均衡,从而保证任务处理的效率。

[0046] 在一些实施例中,各命名空间对应的压缩链表在二维线性链表创建时同时创建;或者

各命名空间对应的压缩链表在首次放入任务时创建。

[0047] 需要说明的是,在具体实施过程中压缩链表可以是同时创建产生的,也可以是在任务产生后创建产生的,优选的可以根据RAID卡形成的逻辑卷的规模或者逻辑卷之间抛任务情况的多少进行设置,例如对应逻辑卷数量较少或者逻辑卷之间相互抛任务的情形较多的情形可以在创建二维线性时一并建立空的压缩链表,当产生任务时将任务直接投放到预先建好的压缩链表中即可,当逻辑卷数量较多时或者逻辑卷之间相互抛任务的情形较少时,为了尽量减少内存的占用可也在产生任务后再建立。

[0048] 在一些实施例中,所述压缩链表包括:

环形队列,所述环形队列用于采用连续内存空间存放预设数量的任务;

第一指针,所述第一指针用于指向环形队列最新放入的任务;

第二指针,所述第二指针用于指向环形队列当前需要处理的任务。

[0049] 在一些实施例中,所述压缩链表还包括:

第三指针,所述第三指针用于指向溢出任务链表的首元素的链表结构体;

第四指针,所述第四指针用于指向溢出任务链表的尾元素的链表结构体。

[0050] 在一些实施例中,所述预设数量等于128。

[0051] 在一些实施例中,所述链表结构体包括:

任务类型字段,所述任务类型字段用于记录任务类型;

第五指针,所述第五指针用于指向下一个任务;

任务执行函数字段,所述任务执行函数字段用于记录与任务类型对应的执行函数。

[0052] 在一些实施例中,所述将任务放入所述目标命名空间对应的压缩链表中的步骤包括:

判断环形队列是否已满;

若环形队列未满,则将产生的任务放入环形队列;

若环形队列已满,则将产生的任务放入溢出任务链表。

[0053] 在一些实施例中,所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤包括:

响应于某一线程处理某一压缩链表中的任务时若环形队列已满,则所述某一线程优先处理溢出任务链表中的任务。

[0054] 在一些实施例中,任意线程处理环形队列中任务时均遵循先进先出原则。

[0055] 在一些实施例中,任意线程处理溢出任务链表中任务时均遵循后进先出原则。

[0056] 在一些实施例中,逻辑卷的任务包括卷的格式化、卷的在线重构、卷的扩容、卷的缩容。

[0057] 在又一个实施例中,为了便于理解本发明的方案下面请以图2所示的RAID卡为例,

不妨假设RAID卡由三个RAID阵列构成,一个是RAID1,一个是RAID5,另一个是RAID6。在实际使用中RAID卡的存储容量巨大,正常情况是成百上千个磁盘构成几十个不同级别的RAID阵列。由于篇幅有限,此处只展示三个RAID阵列构成的RAID卡,假定该RAID卡由控制器0、控制器1、控制器2构成其逻辑控制单元。下面将详细说明用于该RAID卡的命名空间管理方法,具体实施方案如下:

图2中NS就是Namespace的缩写,由下图2可知,Namespace和逻辑卷一一绑定(即NS0和卷0绑定、NS10和卷10绑定),控制器0、控制器1、控制器2负责控制所有的Namespace,图2中一个圆柱表示一个数据块或校验块,其中数据块用D表示,校验块用P表示,例如D1表示数据块1,P1表示校验块1,逻辑卷可以由多个数据块和校验款构成,一个逻辑卷包括的数据块和校验块的数量可以采用任何现有的划分方式,一个逻辑卷对应一个命名空间Namespace(又可简称NS),NS即为灰色椭圆形状表示的区域。

[0058] 目前现有技术通常采用一维线性链表管理Namespace元素,随着用户在RAID卡上创建越来越多的逻辑块设备(即RAID卡内部的逻辑卷volume),与之对应维护Namespace的链表长度随之增加,造成查找Namespace的效率会越来越低。而且Namespace链表元素一旦被一个线程用锁锁定后,其他线程在访问该Namespace链表元素时需要等待锁释放,这样会对RAID卡读写数据性能产生负面影响。

[0059] 本实施例采用二维线性链表管理Namespace元素以此解决上述缺点,如图3所示:本实施例针对目前现有技术管理Namespace元素内存资源浪费的问题,提出将每一个逻辑卷接收到的本逻辑卷或其他逻辑卷抛出的任务都设计为一个压缩链表,使用二维线性链表管理压缩链表。二维线性链表的一个元素代表一个Namespace元素,二维链表的行数和列数等于逻辑卷总数,Namespace元素(例如图3的NS[0][0]和NS[n-1][n]等),以NS[n-1][n]为例,是n-1号线程维护的n-1号逻辑卷(n-1号逻辑卷与n-1号Namespace绑定,故用NS[n-1]表示),因此NS[n-1][n]为n号逻辑卷向n-1号逻辑卷抛出的各种任务(例如卷的格式化任务、在线重构任务、卷的扩容任务、卷的缩容任务等等);

创建与逻辑卷数量相同的线程数,将线程均分到RAID卡的多个控制器上,分配公式可以采用为n对m取余数,余数得几即由几号控制器负责这些数字的线程(例如在图2示例中卷12对3取余为0则由控制器0负责线程12的卷12,同理控制器1负责线程1、线程4、线程7等)。每个线程维护一个一维线性链表(即每个线程维护一个Namespace负责的所有任务),线程0维护与卷0绑定的0号Namespace链表,其中NS[0][0]是卷0抛给卷0自己的任务;NS[0][1]是卷1抛给卷0的任务,同理NS[n][0]是卷n抛给卷n的任务。逻辑卷由本逻辑卷和其他逻辑卷抛向该逻辑卷的任务用一个一维Namespace线性链表维护管理(例如:NS[0][0]、NS[0][1]...NS[0][n-1]和NS[0][n]),多个线程分别与多个逻辑卷一一绑定的Namespace链表(如图3),由此形成二维线性链表,每个线程只负责其自己负责的Namespace而且不使用锁来形成互斥,二维线性链表可以实现相比一维线性链表更快速查找Namespace元素的优势,并且二维线性链表维护管理Namespace元素不需要锁来进行互斥从而提高查找效率,可以彻底免锁,由于没有使用锁且使用二维线性链表,因此当前线程想要访问Namespace元素时就不需等待其他线程对锁的释放,提高I/O性能。

[0060] 如图4所示,我们使用Compress_List结构体来维护压缩链表(即维护逻辑卷接收到的本逻辑卷和其他逻辑卷抛出的多个任务)。下面请结合图5所示将Compress_List结构

体单独拎出来详细介绍：

producer字段为指向本逻辑卷或其他逻辑卷抛向该逻辑卷生成的最新任务，consumer字段为指向本逻辑卷当前需要处理的任务。Ring[128]为环形队列(Ring Buffer 队列)，可以存放128个任务，与producer字段、consumer字段配合一起形成环形队列，环形队列判断为空的判断条件为producer值等于consumer值，环形队列判断为满的判断条件为 $(consumer+1)\%128$ 等于producer。由上所述可知，环形队列采用一维线性数组，一维线性数组内存空间连续，具有占用内存资源小的优点，本发明提出的压缩链表就是将环形队列和链表结合在一起，以发挥一维线性数组占用内存资源小的优势。当任务量超过环形队列可以容纳的128个任务时，由overflow_tail字段和overflow_head字段维护的链表存放环形队列溢出的任务，其中overflow_tail字段指向链表的末位链表元素、overflow_head字段指向链表的首位链表元素。

[0061] 图6中的Element_Node结构体和图3中环形队列的Ring[128]字段一样，都是逻辑卷要处理的任務，其中任务类型用于区分是卷的格式化任务还是在线重构任务或其他任务，next字段指向链表中的下一个任务节点，任务执行函数则是对应任务类型的执行函数。

[0062] 本实施例的RAID卡的命名空间管理方法至少具备以下有益技术效果：

第一，提出使用二维线性链表维护管理Namespace元素不需要锁来进行互斥从而提高查找效率，可以彻底免锁，由于没有使用锁且使用二维线性链表，因此当前线程想要访问Namespace元素时就不需等待其他线程对锁的释放，提高RAID卡的I/O性能；

第二，在RAID卡中设计一个线程只访问一个Namespace线性链表且只有一个操作，多个线程根据本实施例提出的公式均分到多个控制器上，实现多控制器维护二维Namespace线性链表；

第三，创新性地使用二维线性链表，很好解决逻辑卷之间相互抛出任务后Namespace元素混乱的问题；

第四，将一维线性链表设计为压缩链表，压缩链表将环形队列和链表结合在一起使用，通过环形队列实现减少内存资源消耗的作用且不丢失链表的灵活性；

第五，二维线性链表可以实现相比一维线性链表更快速查找Namespace元素的优势。

[0063] 在一些实施例中，请参照图7所示，本发明还提供了一种RAID卡的命名空间管理装置200，所述装置包括：

创建模块201，配置用于创建行数和列数均等于RAID卡的逻辑卷总数的二维线性表，其中，所述二维线性表的每个元素用于表征一个命名空间，且每个元素对应一个压缩链表；

映射模块202，配置用于建立二维线性表的行号与逻辑卷号一一对应关系以得到第一映射关系、以及建立二维线性表的列号与逻辑卷号的一一对应关系以得到第二映射关系；

获取模块203，配置用于响应于操作逻辑卷生成任务，则获取处理任务的逻辑卷对应的第一逻辑卷号和抛出任务的逻辑卷对应的第二逻辑卷号；

确定模块204，配置用于基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间，并将任务放入所述目标命名空间对应的压

缩链表中；

处理模块205,配置用于利用多个线程处理各命名空间对应的压缩链表中的任务,其中,通过相同第一逻辑卷号确定的所有命名空间对应的压缩链表中的任务由同一线程处理。

[0064] 在一些实施例中,所述确定模块204进一步配置用于:
将第一逻辑卷号和所述第一映射关系进行匹配以确定目标行号;
将第二逻辑卷号和所述第二映射关系进行匹配以确定目标列号;
将二维线性表中行号等于目标行号且列号等于目标列列号的元素作为目标命名空间。

[0065] 在一些实施例中,线程总数等于RAID卡的逻辑卷总数;
所述处理模块205进一步配置用于:
建立线程与二维线性链表的行的一一对应关系以得到第三映射关系;
响应于某一命名空间对应的压缩链表中存在需要处理的任務,则将某一命名空间所在行号与所述第三映射关系进行匹配以得到目标线程;
由所述目标线程处理所述某一命名空间对应的压缩链表中的任务。

[0066] 在一些实施例中,所述确定模块204进一步配置用于:
将第一逻辑卷号和所述第二映射关系进行匹配以确定目标列号;
将第二逻辑卷号和所述第一映射关系进行匹配以确定目标行号;
将二维线性表中行号等于目标行号且列号等于目标列列号的元素作为目标命名空间。

[0067] 在一些实施例中,线程总数等于RAID卡的逻辑卷总数;
所述处理模块205进一步配置用于:
建立线程与二维线性链表的列的一一对应关系以得到第四映射关系;
响应于某一命名空间对应的压缩链表中存在需要处理的任務,则将某一命名空间所在列号与所述第四映射关系进行匹配以得到目标线程;
由所述目标线程处理所述某一命名空间对应的压缩链表中的任务。

[0068] 在一些实施例中,所述处理模块205进一步配置用于:
响应于同一线程对应的多个压缩链表中均存在任务需要处理,则采用轮询的机制处理多个压缩链表的任务。

[0069] 在一些实施例中,所述装置还包括配置用于执行以下步骤的模块:
将多个线程均分到RAID卡的多个控制器上执行。

[0070] 在一些实施例中,各命名空间对应的压缩链表在二维线性链表创建时同时创建;
或者

各命名空间对应的压缩链表在首次放入任务时创建。

[0071] 在一些实施例中,所述压缩链表包括:
环形队列,所述环形队列用于采用连续内存空间存放预设数量的任务;
第一指针,所述第一指针用于指向环形队列最新放入的任务;
第二指针,所述第二指针用于指向环形队列当前需要处理的任務。

[0072] 在一些实施例中,所述压缩链表还包括:

第三指针,所述第三指针用于指向溢出任务链表的首元素的链表结构体;
第四指针,所述第四指针用于指向溢出任务链表的尾元素的链表结构体。

[0073] 在一些实施例中,所述预设数量等于128。

[0074] 在一些实施例中,所述链表结构体包括:

任务类型字段,所述任务类型字段用于记录任务类型;

第五指针,所述第五指针用于指向下一个任务;

任务执行函数字段,所述任务执行函数字段用于记录与任务类型对应的执行函数。

[0075] 在一些实施例中,确定模块204进一步配置用于:

判断环形队列是否已满;

若环形队列未满,则将产生的任务放入环形队列;

若环形队列已满,则将产生的任务放入溢出任务链表。

[0076] 在一些实施例中,所述处理模块205进一步配置用于:

响应于某一线程处理某一压缩链表中的任务时若环形队列已满,则所述某一线程优先处理溢出任务链表中的任务。

[0077] 在一些实施例中,任意线程处理环形队列中任务时均遵循先进先出原则。

[0078] 在一些实施例中,任意线程处理溢出任务链表中任务时均遵循后进先出原则。

[0079] 在一些实施例中,逻辑卷的任务包括卷的格式化、卷的在线重构、卷的扩容、卷的缩容。

[0080] 本实施例的一种RAID卡的命名空间管理装置,通过一个行数和列数均等于RAID卡的逻辑卷总数的二维线性表维护RAID卡的命名空间,通过建立行号、列号分别与逻辑卷号的对应关系,当操作逻辑卷生成任务时,通过任务的处理者和抛出者使用创建的对应关系找到二维线性链表对应位置的命名空间,然后将任务放入到与命名空间对应的压缩链表中,最后对于多个线程处理压缩链表中的任务,在处理任务时相同任务处理者的任务都由同一个线程处理,可以实现对命名空间的彻底免锁,不需等待其他线程对锁的释放,显著提升命名空间的查找和管理效率,有助于提高RAID卡的I/O性能。

[0081] 需要说明的是,关于RAID卡的命名空间管理装置的具体限定可以参见上文中对RAID卡的命名空间管理方法的限定,在此不再赘述。上述RAID卡的命名空间管理装置中的各个模块可全部或部分通过软件、硬件及其组合来实现。上述各模块可以硬件形式内嵌于或独立于计算机设备中的处理器中,也可以以软件形式存储于计算机设备中的存储器中,以便于处理器调用执行以上各个模块对应的操作。

[0082] 根据本发明的另一方面,提供了一种计算机设备,该计算机设备可以是服务器,其内部结构图请参照图8所示。该计算机设备包括通过系统总线连接的处理器、存储器、网络接口和数据库。其中,该计算机设备的处理器用于提供计算和控制能力。该计算机设备的存储器包括非易失性存储介质、内存储器。该非易失性存储介质存储有操作系统、计算机程序和数据库。该内存储器为非易失性存储介质中的操作系统和计算机程序的运行提供环境。该计算机设备的数据库用于存储数据。该计算机设备的网络接口用于与外部的终端通过网络连接通信。该计算机程序被处理器执行时实现以上所述的RAID卡的命名空间管理方法,具体来说,所述方法包括以下步骤:

创建行数和列数均等于RAID卡的逻辑卷总数的二维线性表,其中,所述二维线性表的每个元素用于表征一个命名空间,且每个元素对应一个压缩链表;

建立二维线性表的行号与逻辑卷号一一对应关系以得到第一映射关系、以及建立二维线性表的列号与逻辑卷号的一一对应关系以得到第二映射关系;

响应于操作逻辑卷生成任务,则获取处理任务的逻辑卷对应的第一逻辑卷号和抛出任务的逻辑卷对应的第二逻辑卷号;

基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间,并将任务放入所述目标命名空间对应的压缩链表中;

利用多个线程处理各命名空间对应的压缩链表中的任务,其中,通过相同第一逻辑卷号确定的所有命名空间对应的压缩链表中的任务由同一线程处理。

[0083] 在一些实施例中,所述基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间的步骤包括:

将第一逻辑卷号和所述第一映射关系进行匹配以确定目标行号;

将第二逻辑卷号和所述第二映射关系进行匹配以确定目标列号;

将二维线性表中行号等于目标行号且列号等于目标列列号的元素作为目标命名空间。

[0084] 在一些实施例中,线程总数等于RAID卡的逻辑卷总数;

所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤包括:

建立线程与二维线性链表的行的一一对应关系以得到第三映射关系;

响应于某一命名空间对应的压缩链表中存在需要处理的任務,则将某一命名空间所在行号与所述第三映射关系进行匹配以得到目标线程;

由所述目标线程处理所述某一命名空间对应的压缩链表中的任务。

[0085] 在一些实施例中,所述基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间的步骤包括:

将第一逻辑卷号和所述第二映射关系进行匹配以确定目标列号;

将第二逻辑卷号和所述第一映射关系进行匹配以确定目标行号;

将二维线性表中行号等于目标行号且列号等于目标列列号的元素作为目标命名空间。

[0086] 在一些实施例中,线程总数等于RAID卡的逻辑卷总数;

所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤包括:

建立线程与二维线性链表的列的一一对应关系以得到第四映射关系;

响应于某一命名空间对应的压缩链表中存在需要处理的任務,则将某一命名空间所在列号与所述第四映射关系进行匹配以得到目标线程;

由所述目标线程处理所述某一命名空间对应的压缩链表中的任务。

[0087] 在一些实施例中,所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤还包括:

响应于同一线程对应的多个压缩链表中均存在任务需要处理,则采用轮询的机制处理多个压缩链表的任务。

[0088] 在一些实施例中,所述方法还包括:

将多个线程均分到RAID卡的多个控制器上执行。

[0089] 在一些实施例中,各命名空间对应的压缩链表在二维线性链表创建时同时创建;或者

各命名空间对应的压缩链表在首次放入任务时创建。

[0090] 在一些实施例中,所述压缩链表包括:

环形队列,所述环形队列用于采用连续内存空间存放预设数量的任务;

第一指针,所述第一指针用于指向环形队列最新放入的任务;

第二指针,所述第二指针用于指向环形队列当前需要处理的任務。

[0091] 在一些实施例中,所述压缩链表还包括:

第三指针,所述第三指针用于指向溢出任务链表的首元素的链表结构体;

第四指针,所述第四指针用于指向溢出任务链表的尾元素的链表结构体。

[0092] 在一些实施例中,所述预设数量等于128。

[0093] 在一些实施例中,所述链表结构体包括:

任务类型字段,所述任务类型字段用于记录任务类型;

第五指针,所述第五指针用于指向下一个任务;

任务执行函数字段,所述任务执行函数字段用于记录与任务类型对应的执行函数。

[0094] 在一些实施例中,所述将任务放入所述目标命名空间对应的压缩链表中的步骤包括:

判断环形队列是否已满;

若环形队列未滿,则将产生的任务放入环形队列;

若环形队列已滿,则将产生的任务放入溢出任务链表。

[0095] 在一些实施例中,所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤包括:

响应于某一线程处理某一压缩链表中的任务时若环形队列已滿,则所述某一线程优先处理溢出任务链表中的任务。

[0096] 在一些实施例中,任意线程处理环形队列中任务时均遵循先进先出原则。

[0097] 在一些实施例中,任意线程处理溢出任务链表中任务时均遵循后进先出原则。

[0098] 在一些实施例中,逻辑卷的任务包括卷的格式化、卷的在线重构、卷的扩容、卷的缩容。

[0099] 根据本发明的又一方面,提供了一种计算机可读存储介质,其上存储有计算机程序,计算机程序被处理器执行时实现以上所述的RAID卡的命名空间管理方法,具体来说,包括执行以下步骤:

创建行数和列数均等于RAID卡的逻辑卷总数的二维线性表,其中,所述二维线性表的每个元素用于表征一个命名空间,且每个元素对应一个压缩链表;

建立二维线性表的行号与逻辑卷号一一对应关系以得到第一映射关系、以及建立二维线性表的列号与逻辑卷号的一一对应关系以得到第二映射关系;

响应于操作逻辑卷生成任务,则获取处理任务的逻辑卷对应的第一逻辑卷号和抛出任务的逻辑卷对应的第二逻辑卷号;

基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间,并将任务放入所述目标命名空间对应的压缩链表中;

利用多个线程处理各命名空间对应的压缩链表中的任务,其中,通过相同第一逻辑卷号确定的所有命名空间对应的压缩链表中的任务由同一线程处理。

[0100] 在一些实施例中,所述基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间的步骤包括:

将第一逻辑卷号和所述第一映射关系进行匹配以确定目标行号;

将第二逻辑卷号和所述第二映射关系进行匹配以确定目标列号;

将二维线性表中行号等于目标行号且列号等于目标列列号的元素作为目标命名空间。

[0101] 在一些实施例中,线程总数等于RAID卡的逻辑卷总数;

所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤包括:

建立线程与二维线性链表的行的一一对应关系以得到第三映射关系;

响应于某一命名空间对应的压缩链表中存在需要处理的任務,则将某一命名空间所在行号与所述第三映射关系进行匹配以得到目标线程;

由所述目标线程处理所述某一命名空间对应的压缩链表中的任务。

[0102] 在一些实施例中,所述基于所述第一逻辑卷号、所述第二逻辑卷号、所述第一映射关系和所述第二映射关系确定目标命名空间的步骤包括:

将第一逻辑卷号和所述第二映射关系进行匹配以确定目标列号;

将第二逻辑卷号和所述第一映射关系进行匹配以确定目标行号;

将二维线性表中行号等于目标行号且列号等于目标列列号的元素作为目标命名空间。

[0103] 在一些实施例中,线程总数等于RAID卡的逻辑卷总数;

所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤包括:

建立线程与二维线性链表的列的一一对应关系以得到第四映射关系;

响应于某一命名空间对应的压缩链表中存在需要处理的任務,则将某一命名空间所在列号与所述第四映射关系进行匹配以得到目标线程;

由所述目标线程处理所述某一命名空间对应的压缩链表中的任务。

[0104] 在一些实施例中,所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤还包括:

响应于同一线程对应的多个压缩链表中均存在任务需要处理,则采用轮询的机制处理多个压缩链表的任务。

[0105] 在一些实施例中,所述方法还包括:

将多个线程均分到RAID卡的多个控制器上执行。

[0106] 在一些实施例中,各命名空间对应的压缩链表在二维线性链表创建时同时创建;或者

各命名空间对应的压缩链表在首次放入任务时创建。

[0107] 在一些实施例中,所述压缩链表包括:

环形队列,所述环形队列用于采用连续内存空间存放预设数量的任务;

第一指针,所述第一指针用于指向环形队列最新放入的任务;
第二指针,所述第二指针用于指向环形队列当前需要处理的任务。

[0108] 在一些实施例中,所述压缩链表还包括:

第三指针,所述第三指针用于指向溢出任务链表的首元素的链表结构体;
第四指针,所述第四指针用于指向溢出任务链表的尾元素的链表结构体。

[0109] 在一些实施例中,所述预设数量等于128。

[0110] 在一些实施例中,所述链表结构体包括:

任务类型字段,所述任务类型字段用于记录任务类型;

第五指针,所述第五指针用于指向下一个任务;

任务执行函数字段,所述任务执行函数字段用于记录与任务类型对应的执行函数。

[0111] 在一些实施例中,所述将任务放入所述目标命名空间对应的压缩链表中的步骤包括:

判断环形队列是否已满;

若环形队列未满,则将产生的任务放入环形队列;

若环形队列已满,则将产生的任务放入溢出任务链表。

[0112] 在一些实施例中,所述利用多个线程处理各命名空间对应的压缩链表中的任务的步骤包括:

响应于某一线程处理某一压缩链表中的任务时若环形队列已满,则所述某一线程优先处理溢出任务链表中的任务。

[0113] 在一些实施例中,任意线程处理环形队列中任务时均遵循先进先出原则。

[0114] 在一些实施例中,任意线程处理溢出任务链表中任务时均遵循后进先出原则。

[0115] 在一些实施例中,逻辑卷的任务包括卷的格式化、卷的在线重构、卷的扩容、卷的缩容。

[0116] 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,所述的计算机程序可存储于一非易失性计算机可读存储介质中,该计算机程序在执行时,可包括如上述各方法的实施例的流程。其中,本申请所提供的各实施例中所使用的对存储器、存储、数据库或其它介质的任何引用,均可包括非易失性和/或易失性存储器。非易失性存储器可包括只读存储器(ROM)、可编程ROM(PROM)、电可编程ROM(EPROM)、电可擦除可编程ROM(EEPROM)或闪存。易失性存储器可包括随机存取存储器(RAM)或者外部高速缓冲存储器。作为说明而非局限,RAM以多种形式可得,诸如静态RAM(SRAM)、动态RAM(DRAM)、同步DRAM(SDRAM)、双数据率SDRAM(DDRSDRAM)、增强型SDRAM(ESDRAM)、同步链路(Synchlink) DRAM(SLDRAM)、存储器总线(Rambus)直接RAM(RDRAM)、直接存储器总线动态RAM(DRDRAM)、以及存储器总线动态RAM(RDRAM)等。

[0117] 以上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。

[0118] 以上所述实施例仅表达了本申请的几种实施方式,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,在不脱离本申请构思的前提下,还可以做出若干变形和改进,这些都属于本申请的保护范围。因此,本申请专利的保护范围应以所附权利要求为准。

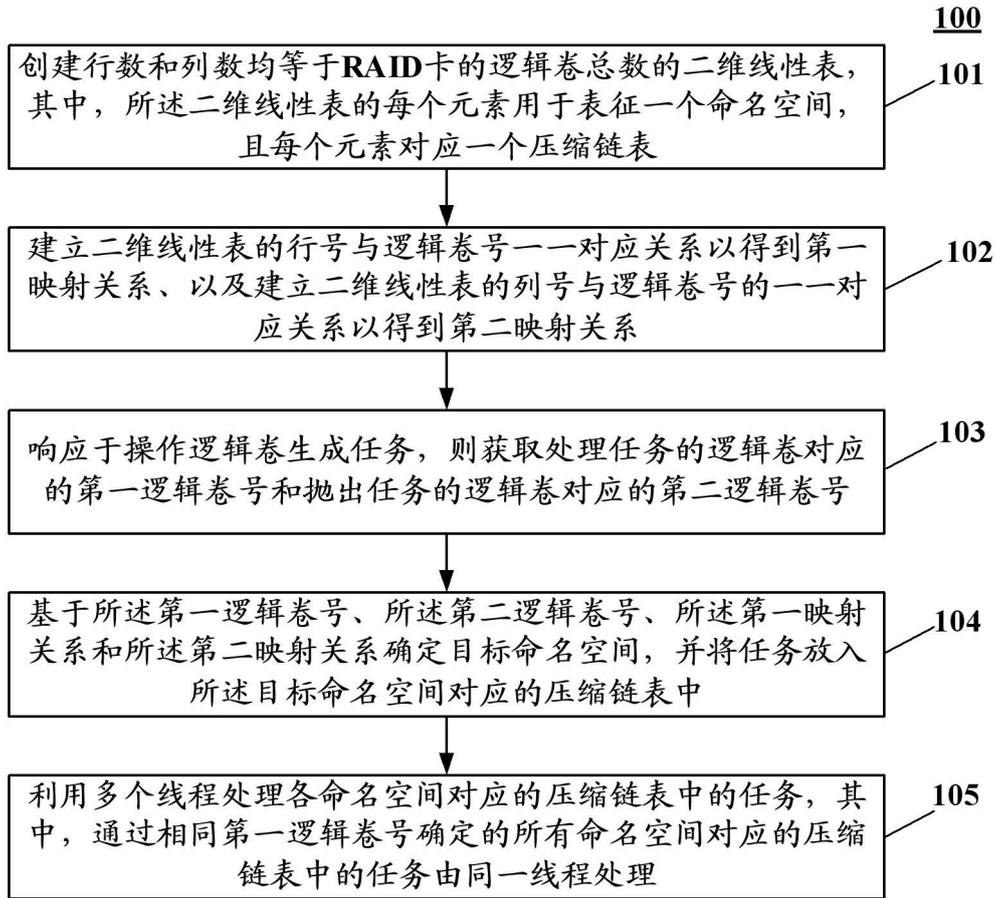


图1

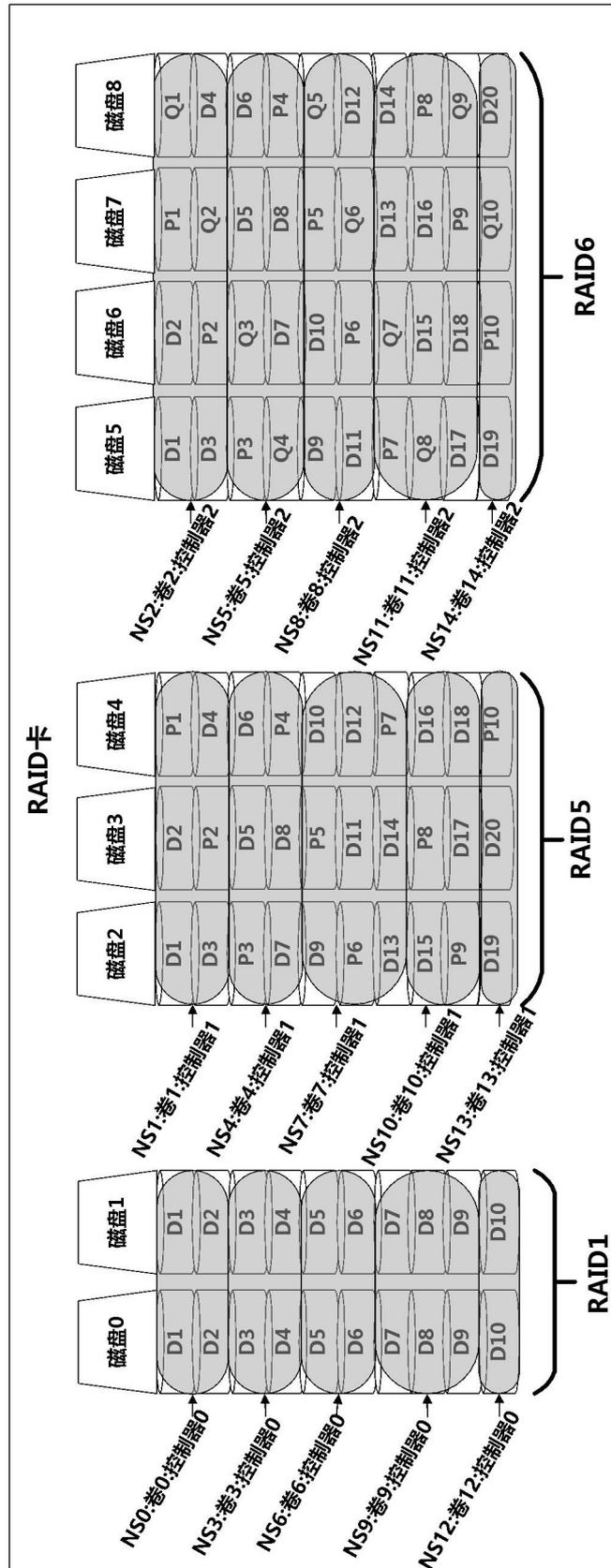


图2

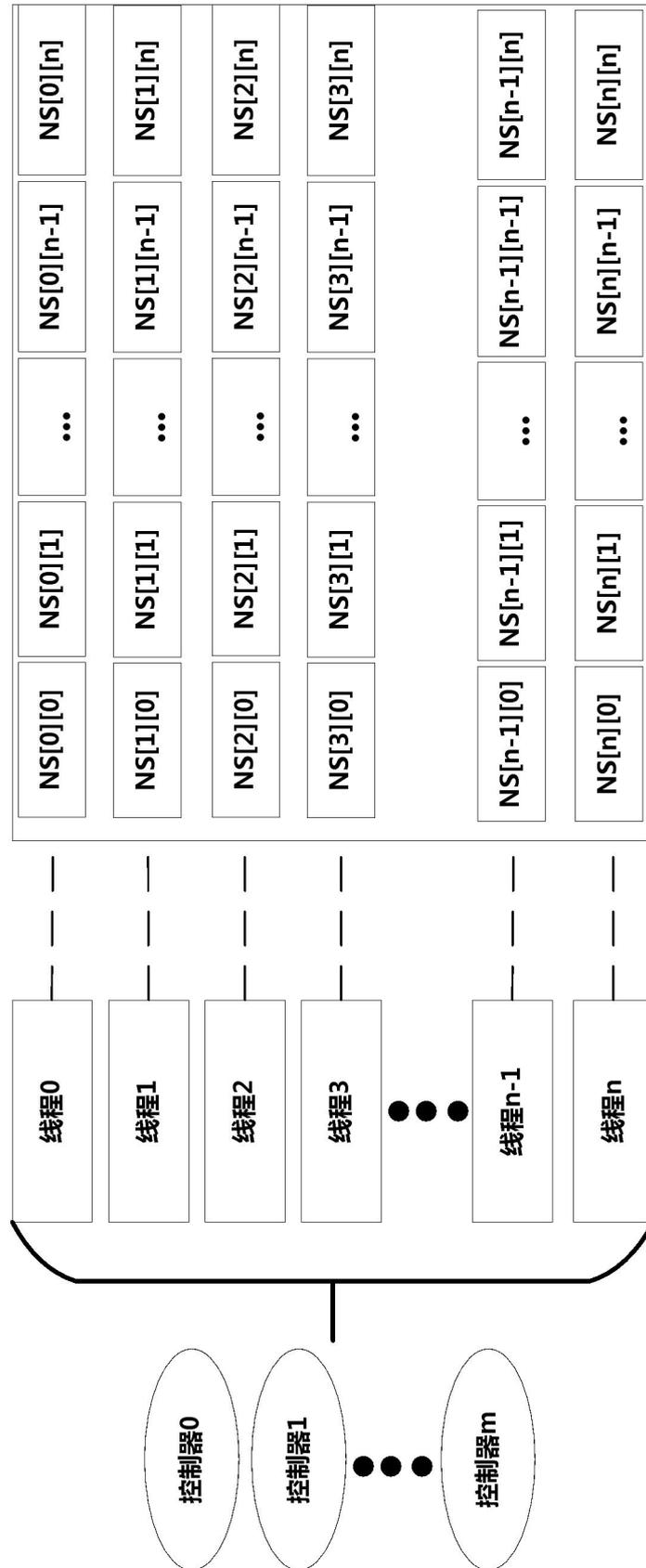


图3

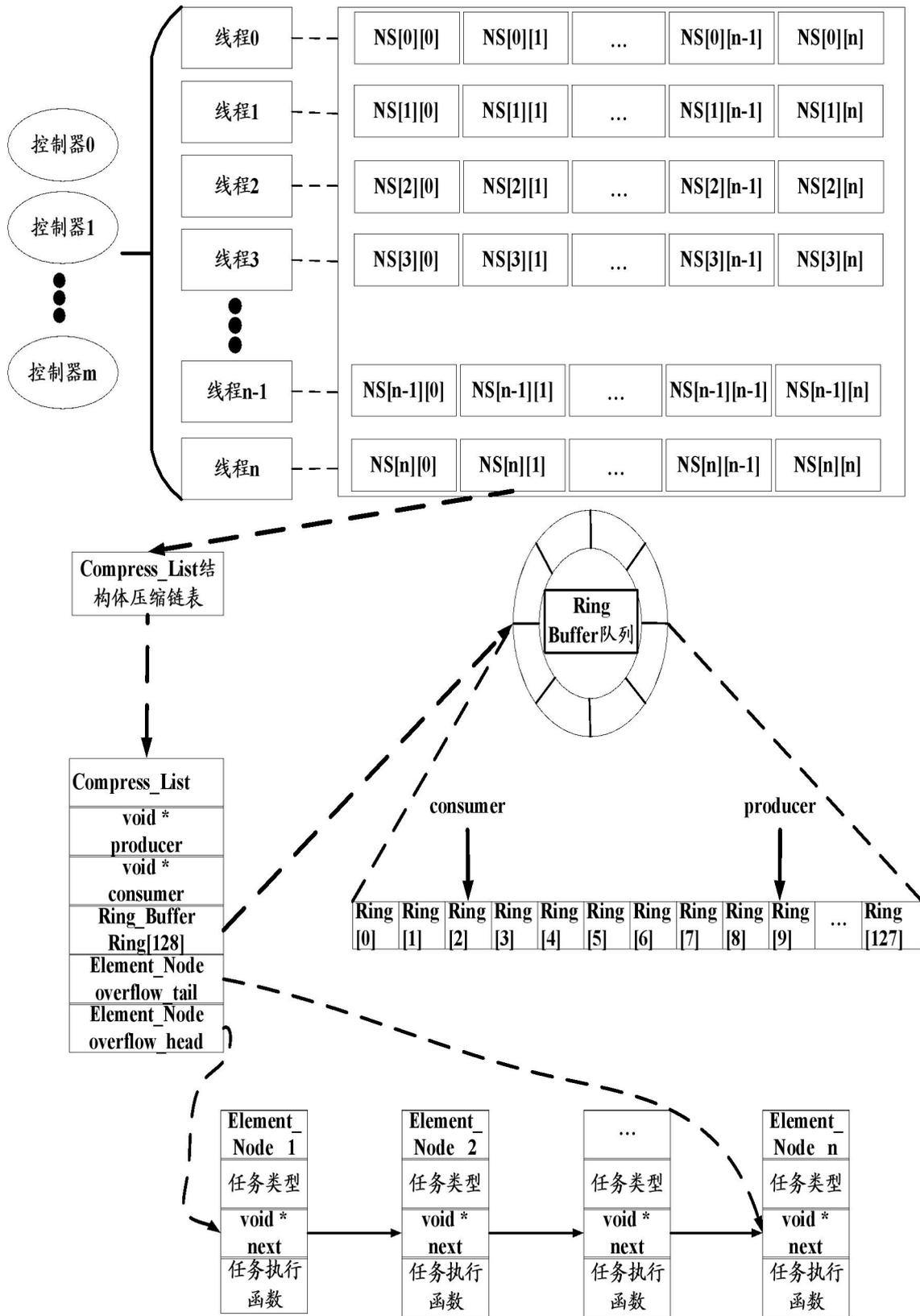


图4

Compress_List	
void *	producer
void *	consumer
Ring_Buffer	Ring[128]
Element_Node	overflow_tail
Element_Node	overflow_head

图5

Element_Node
任务类型
void * next
任务执行函数

图6

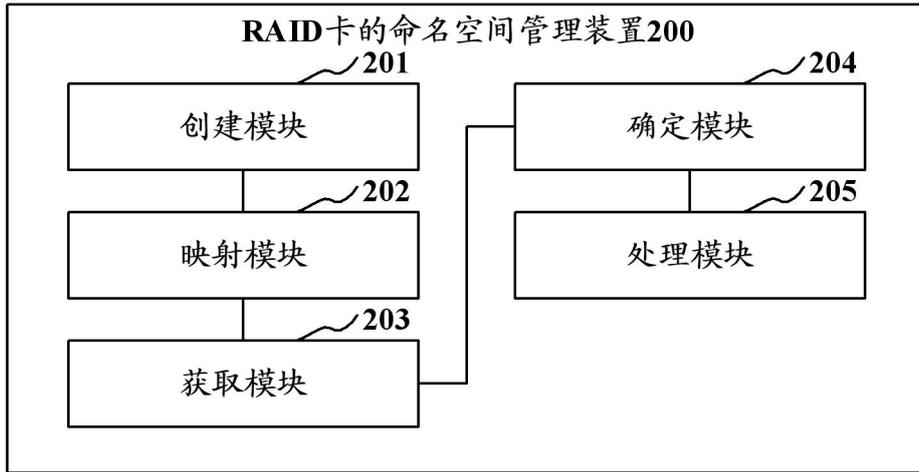


图7

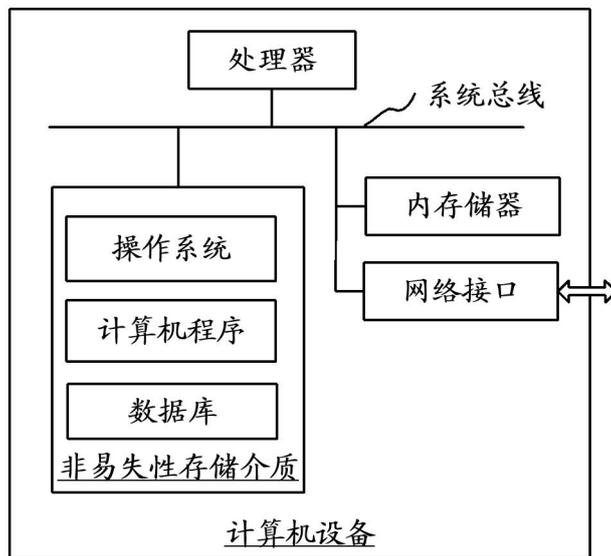


图8