



## (12) 发明专利申请

(10) 申请公布号 CN 103370686 A

(43) 申请公布日 2013. 10. 23

(21) 申请号 201180054158. 1

代理人 党建华

(22) 申请日 2011. 09. 15

(51) Int. Cl.

(30) 优先权数据

G06F 3/06 (2006. 01)

12/882, 872 2010. 09. 15 US

G06F 11/10 (2006. 01)

G06F 12/02 (2006. 01)

(85) PCT申请进入国家阶段日

2013. 05. 10

(86) PCT申请的申请数据

PCT/US2011/051710 2011. 09. 15

(87) PCT申请的公布数据

W02012/037318 EN 2012. 03. 22

(71) 申请人 净睿存储股份有限公司

地址 美国加利福尼亚

(72) 发明人 J·科尔格洛夫 J·海斯 洪波

王峰 E·米勒 C·哈莫

(74) 专利代理机构 中国国际贸易促进委员会专  
利商标事务所 11038

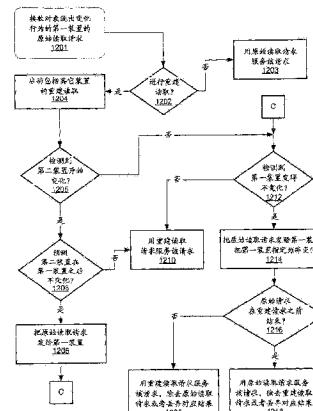
权利要求书3页 说明书15页 附图11页

(54) 发明名称

存储环境中重建 I/O 读取操作的调度

(57) 摘要

一种在多个固态存储装置之间，有效地调度读取操作和写入操作的系统和方法。计算机系统包括借助网络相互耦接的客户端计算机和数据存储阵列。数据存储阵列把固态驱动器和快闪存储器单元用于数据存储。数据存储阵列内的存储控制器包含 I/O 调度器。存储控制器被配置成接收以数据存储介质为目标的读取请求，识别所述多个存储装置中包含读取请求所针对的数据的至少第一存储装置。响应检测到或者预测第一存储装置将表现出可变性能，所述控制器被配置成产生重建读取请求，所述重建读取请求被配置成从多个存储装置中的除第一存储装置以外的一个或多个装置获得所述数据。



1. 一种计算机系统,包括:

数据存储介质,所述数据存储介质包括配置成把数据保存在至少一个 RAID 组中的多个存储装置;和

与数据存储介质耦接的数据存储控制器;

其中数据存储控制器被配置成:

接收以数据存储介质为目标的读取请求;

识别所述多个存储装置中包含读取请求所针对的数据的至少第一存储装置;和

响应检测到或者预测第一存储装置将表现出可变性能,产生重建读取请求,所述重建读取请求被配置成从所述多个存储装置中的除第一存储装置以外的一个或多个装置获得所述数据,其中所述可变性能包括较高的响应等待时间和 / 或较低的吞吐量。

2. 按照权利要求 1 所述的计算机系统,其中存储控制器被配置成至少部分根据 I/O 请求的最近历史,产生所述重建读取请求。

3. 按照权利要求 1 所述的计算机系统,其中重建读取请求包括以所述多个存储装置中的至少两个装置为目标的至少两个读取请求,以及其中存储控制器被配置成调度所述至少两个读取请求,以致它们在大约相同的时间完成。

4. 按照权利要求 1 所述的计算机系统,其中存储控制器还被配置成调度等待时间较长的操作,以致在任何给定时间,所述 RAID 组中的多个装置中的至多 N 个装置正在进行调度的等待时间长的操作。

5. 按照权利要求 4 所述的计算机系统,其中所述等待时间长的操作包括高速缓存清洗操作、修剪操作、擦除区块操作、休眠、写入或者大型读取中的一个或多个。

6. 按照权利要求 4 所述的计算机系统,其中响应检测到接收的请求的比率超过给定阈值,存储控制器被配置成调度等待时间较长的操作,以致在任何给定时间,允许 RAID 组内的多于 N 个装置处于繁忙状态。

7. 按照权利要求 6 所述的计算机系统,其中响应检测到请求的比率已经降到阈值之下,存储控制器被配置成调度等待时间较长的操作,以致在任何给定时间,所述 RAID 组中的多个装置中的至多 N 个装置正在进行调度的等待时间长的操作。

8. 按照权利要求 3 所述的计算机系统,其中所述多个存储装置中的每一个都包括用于保存未决操作的队列,以及其中存储控制器被配置成通过把所述至少两个读取请求中的每一个都保存在对应存储装置的队列中,近似相同的队列深度,或者预测的完成时间近似相同的队列深度,来调度所述至少两个读取请求,以致它们在近似相同的时间完成。

9. 按照权利要求 8 所述的计算机系统,其中读取请求具有给定优先级,以及其中存储控制器被配置成以与所述给定优先级相同的优先级,调度所述至少两个读取请求二者。

10. 按照权利要求 8 所述的计算机系统,其中读取请求具有给定优先级,以及其中存储控制器被配置成调度所述至少两个读取请求中的每一个,以具有彼此不同的优先级。

11. 按照权利要求 10 所述的计算机系统,其中存储控制器被配置成确定优先级,根据所述至少两个读取请求中的读取所针对的装置的状态,以所述优先级调度所述至少两个读取请求中的每一个。

12. 按照权利要求 11 所述的计算机系统,其中所述状态包括对应装置的队列占用率、和对应装置的平均响应等待时间中的一个或多个。

13. 按照权利要求 3 所述的计算机系统, 其中存储控制器还被配置成把接收的请求划分成多个请求, 并在所述多个请求之间插入重建读取请求。

14. 一种在计算系统中使用的方法, 所述方法包括 :

接收以数据存储介质为目标的读取请求, 所述数据存储介质包括配置成把数据保存在至少一个 RAID 组中的多个存储装置 ;

识别所述多个存储装置中包含读取请求所针对的数据的至少第一存储装置 ; 和

响应检测到或者预测第一存储装置将表现出可变性能, 产生重建读取请求, 所述重建读取请求被配置成从所述多个存储装置中的除第一存储装置以外的一个或多个装置获得所述数据, 其中所述可变性能包括较高的响应等待时间和 / 或较低的吞吐量。

15. 按照权利要求 14 所述的方法, 还包括至少部分根据 I/O 请求的最近历史, 产生所述重建读取请求。

16. 按照权利要求 14 所述的方法, 其中所述重建读取请求包括以所述多个存储装置中的至少两个装置为目标的至少两个读取请求, 以及其中所述方法还包括调度所述至少两个读取请求, 以致它们在大约相同的时间完成。

17. 按照权利要求 14 所述的方法, 还包括调度等待时间较长的操作, 以致在任何给定时间, 所述 RAID 组中的多个装置中的至多 N 个装置正在进行调度的等待时间长的操作。

18. 按照权利要求 17 所述的方法, 其中所述等待时间长的操作包括高速缓存清洗操作、修剪操作、擦除区块操作、休眠、写入或者大型读取中的一个或多个。

19. 按照权利要求 17 所述的方法, 其中响应检测到接收的请求的比率超过给定阈值, 所述方法还包括调度等待时间较长的操作, 以致在任何给定时间, 允许 RAID 组内的多于 N 个装置处于繁忙状态, 在 RAID 组内的多于 N 个装置处于繁忙状态时, 重建读取是不可能发生的。

20. 按照权利要求 19 所述的方法, 其中响应检测到请求的比率已降到阈值之下, 所述方法还包括调度等待时间较长的操作, 以致在任何给定时间, 所述 RAID 组中的多个装置中的至多 N 个装置正在进行调度的等待时间长的操作。

21. 按照权利要求 16 所述的方法, 其中所述多个存储装置中的每一个都包括用于保存未决操作的队列, 以及其中所述方法还包括通过把所述至少两个读取请求中的每一个都保存在对应存储装置的队列中, 近似相同的队列深度, 或者预测的完成时间近似相同的队列深度, 来调度所述至少两个读取请求, 以致它们在近似相同的时间完成。

22. 按照权利要求 21 所述的方法, 其中读取请求具有给定优先级, 以及其中所述方法还包括以与所述给定优先级相同的优先级, 调度所述至少两个读取请求二者。

23. 按照权利要求 21 所述的方法, 其中读取请求具有给定优先级, 以及其中所述方法还包括调度所述至少两个读取请求中的每一个, 以具有彼此不同的优先级。

24. 一种包含程序指令的计算机可读存储介质, 其中当被处理装置执行时, 所述程序指令可操作以 :

接收以数据存储介质为目标的读取请求, 所述数据存储介质包括配置成把数据保存在至少一个 RAID 组中的多个存储装置 ;

识别所述多个存储装置中包含读取请求所针对的数据的至少第一存储装置 ; 和

响应检测到或者预测第一存储装置将表现出可变性能, 产生重建读取请求, 所述重建

读取请求被配置成从多个存储装置中的除第一存储装置以外的一个或多个装置获得所述数据，其中所述可变性能包括较高的响应等待时间和 / 或较低的吞吐量。

## 存储环境中重建 I/O 读取操作的调度

### 技术领域

[0001] 本发明涉及计算机网络,更具体地说,涉及计算数据存储系统。

### 背景技术

[0002] 随着计算机存储器存储量和数据带宽的增大,企业管理的数据的数量和复杂性也增大。诸如数据中心之类的大型分布式存储系统一般进行许多业务运营。分布式存储系统可耦接到利用一个或多个网络互连的许多客户端计算机。如果分布式存储系统的任意部分的性能差或者变得不可用,那么公司运营会受到损害或者完全停止。这样的分布式系统试图维持对于数据可用性和高性能功能的高标准。

[0003] 在存储系统本身中,文件系统和存储装置级输入 / 输出 (I/O) 调度器通常确定读取操作和写入操作的顺序,以及提供所述操作将被如何执行的步骤。例如,对存储装置来说,与顺序读取操作和写入操作相比,非顺序读取操作和写入操作可能执行代价更高(例如,就时间和 / 或资源而论)。于是,I/O 调度器会试图减少非顺序操作。另外,I/O 调度器可提供另外的功能,比如防止饥饿、请求合并和进程间公平。

[0004] 在存储装置之间,至少读取和写入响应时间差异相当大。这样的差异可能是技术本身特性。从而,与选择的数据存储装置相关的技术和机制可决定用于进行有效 I/O 调度的方法。例如,为利用硬盘驱动器 (HDD) 的系统,建立了许多现行的算法。HDD 包含一个或多个旋转盘,每个盘覆盖有磁性介质。这些盘以每分钟数千转的转速旋转。另外,电磁致动器负责把磁性读 / 写装置定位到旋转的盘上。装置的机械和机电设计影响其 I/O 特性。不幸的是,摩擦、磨损、振动和机械不对准会产生可靠性问题,以及影响 HDD 的 I/O 特性。考虑到 HDD 的输入 / 输出 (I/O) 特性,设计许多现行的 I/O 调度器。

[0005] 另一种存储介质的一个例子是固态驱动器 (SSD)。与 HDD 相反,SSD 利用固态存储器,而不是磁介质装置来保存持久数据。固态存储器可包含快闪存储器单元。快闪存储器具有不同于硬盘驱动器的许多特征。例如,在被重写或者重新编程之前,快闪存储器单元通常是按较大的区块擦除的。通常还在复杂的布置,比如小片、封装、板和区块中,排列快闪存储器。所选布置的尺寸和平行性、快闪存储器随着时间的磨损、及装置的互连和传送速度都会变化。另外,这样的装置还可包括管理装置上的存储的闪存转换层 (FTL)。FTL 利用的算法可变,并且还会对装置的行为和 / 或性能的变化产生影响。从而,在把基于闪存的 SSD 用于存储,同时利用为诸如具有不同特性的硬盘驱动器之类的系统设计的 I/O 调度器的系统中,通常不能获得较高的性能和可预测的等待时间。

[0006] 鉴于上面所述,需要在多个存储装置之间,有效地调度读取操作和写入操作的系统和方法。

### 发明内容

[0007] 公开了在多个固态存储装置之间,有效地调度读取操作和写入操作的计算机系统和方法的各个实施例。

[0008] 在一个实施例中，计算机系统包括多个客户端计算机，所述多个客户端计算机被配置成通过网络，把读取请求和写入请求传送给借助网络耦接以接收所述读取请求和写入请求的一个或多个数据存储阵列。可预想包含多个存储装置上的多个存储存储位置的数据存储阵列。在各种实施例中，按用于数据存储和保护的独立驱动器冗余阵列（RAID）布置配置存储装置。数据存储装置可包括用于数据存储的固态存储器技术，比如快闪存储器单元。对应存储装置的特性用于调度对存储装置的 I/O 请求。所述特性可包括对 I/O 请求的预测响应时间、装置使用年限、任何对应的高速缓存大小、存取速率、差错率、当前 I/O 请求、完成的 I/O 请求等等。

[0009] 在一个实施例中，I/O 调度器被配置成接收读取请求和写入请求，并调度读取请求和写入请求，以便由多个存储装置处理。取决于所服务的操作，存储装置会表现出变化的等待时间，还会在各个时间表现出非调度的或者不可预知的行为，导致性能不同于预期或者期望的性能。在各种实施例中，这些行为对应于装置正常工作（即，不处于错误状态），但是仅仅根据等待时间和 / 或吞吐量，以低于预期或者期望的水平进行的行为。这样的行为和性能可被称为“可变性能”行为。例如，诸如基于闪存的存储技术之类的技术会表现出这些可变性能行为。可构思存储控制器，所述存储控制器被配置成接收以数据存储介质为目标的读取请求，并识别多个存储装置中包含读取请求所针对的数据的至少第一存储装置。响应或者检测到或者预测所述第一存储装置将表现出可变性能（所述可变性能包含较高的响应等待时间和 / 或较低的吞吐量），所述控制器被配置成产生重建读取请求，所述重建读取请求被配置成从所述多个存储装置中除第一存储装置以外的一个或多个装置获得数据。

[0010] 根据以下的说明和附图，这些和其它实施例将变得明显。

## 附图说明

- [0011] 图 1 是图解说明网络体系结构的一个实施例的广义方框图。
- [0012] 图 2 描述按照计算系统的一个实施例的概念模型。
- [0013] 图 3 是图解说明调整 I/O 调度，以减小数据存储子系统上的不可预测的可变 I/O 响应时间的方法的一个实施例的广义方框图。
- [0014] 图 4 是图解说明隔离发给存储装置的操作的方法的一个实施例的广义方框图。
- [0015] 图 5 是图解说明建立模型，以表征存储子系统中的存储装置的行为的方法的一个实施例的广义流程图。
- [0016] 图 6 是图解说明存储子系统的一个实施例的广义方框图。
- [0017] 图 7 是图解说明装置单元的另一个实施例的广义方框图。
- [0018] 图 8 是图解说明状态表的另一个实施例的广义方框图。
- [0019] 图 9 是图解说明调整 I/O 调度，以减小数据存储子系统上的不可预测的可变 I/O 响应时间的方法的一个实施例的广义流程图。
- [0020] 图 10 是图解说明在共享数据存储器上，维持具有有效等待时间的读取操作的方法的一个实施例的广义流程图。
- [0021] 图 11 是图解说明减少表现出可变 I/O 响应时间的存储装置的数目的方法的一个实施例的广义流程图。
- [0022] 图 12 是图解说明在共享数据存储器上，维持具有有效等待时间的读取操作的方

法的一个实施例的广义流程图。

[0023] 虽然本发明容许各种修改和备选形式,不过在附图中举例表示,并且这里详细说明了具体的实施例。然而应理解,附图及对其的详细说明并不意图把本发明局限于公开的特定形式,相反,本发明覆盖在附加权利要求限定的本发明的精神和范围内的所有修改、等同物和替换物。

## 具体实施方式

[0024] 在下面的说明中,陈述了众多的具体细节,以便充分理解本发明。不过,本领域的普通技术人员会认识到可在没有这些具体细节的情况下,实践本发明。在一些情况下,未详细表示公知的电路、结构、信号、计算机程序指令和技术,以避免模糊本发明。

[0025] 参见图 1,图中表示了网络体系结构 100 的一个实施例的广义方框图。如下进一步所述,网络体系结构 100 的一个实施例包括通过网络 180 互连,并且连接到数据存储阵列 120a-120b 的客户端计算机系统 110a-110b。网络 180 可通过交换机 140 高认接第二网络 190。客户端计算机系统 110c 通过网络 190 高认接客户端计算机系统 110a-110b 和数据存储阵列 120a-120b。另外,网络 190 可通过交换机 150 高认接因特网 160 或者其它外部网络。

[0026] 注意在备选实施例中,客户端计算机和服务器、交换机、网络、数据存储阵列和数据存储装置的数目和种类并不局限于图 1 中所示的数目和种类。在各种时候,一个或多个客户端可以离线工作。另外,在工作期间,各个客户端计算机连接种类可随用户连接、脱离和重新连接到网络体系结构 100 而变化。此外,虽然本说明一般讨论网络附连存储,不过,这里说明的系统和方法也适用于直接附连存储系统,可包括配置成实现所述方法的一个或多个方面的主机操作系统。众多这样的备选方案都是可能的,并可构思众多这样的备选方案。下面简要提供图 1 中所示的各个组件的进一步说明。首先,说明数据存储阵列 120a-120b 提供的一些特征的概况。

[0027] 在网络体系结构 100 中,每个数据存储阵列 120a-120b 可用于不同的服务器和计算机,比如客户端计算机系统 110a-110c 之间的数据的共享。另外,数据存储阵列 120a-120b 可用于盘镜像、备份和恢复、归档和归档数据的取回、和从一个存储装置到另一个存储装置的数据迁移。在备选实施例中,可以使一个或多个客户端计算机系统 110a-110c 通过快速局域网 (LAN) 相互链接,以便形成群集。这样的客户端可以共享存储资源,比如存在于数据存储阵列 120a-120b 之一内的群集共享卷。

[0028] 每个数据存储阵列 120a-120b 包括用于数据存储的存储子系统 170。存储子系统 170 可包括多个存储装置 176a-176m。这些存储装置 176a-176m 可向客户端计算机系统 110a-110c 提供数据存储服务。每个存储装置 176a-176m 利用特殊的技术和机制进行数据存储。在每个存储装置 176a-176m 内使用的技术和机制的种类可至少部分用于确定用于控制和调度相对于每个存储装置 176a-176m 的读取操作和写入操作的算法。在这些算法中使用的逻辑可被包含在基本操作系统 (OS) 116、文件系统 110、存储子系统控制器 174 内的一个或多个全局 I/O 调度器 178、每个存储装置 176a-176m 内的控制逻辑等中的一个或多个中。另外,这里说明的逻辑、算法和控制机制可包括硬件和 / 或软件。

[0029] 每个存储装置 176a-176m 可被配置成接收读取请求和写入请求,并且包括多个数

据存储位置,每个数据存储位置是可作为阵列中的行和列寻址的。在一个实施例中,存储装置 176a-176m 内的数据存储位置可被布置成逻辑的冗余存储容器或者 RAID 阵列(廉价 / 独立盘的冗余阵列)。在一些实施例中,每个存储装置 176a-176m 可以利用与常规硬盘驱动器(HDD)不同的数据存储技术。例如,存储装置 176a-176m 中的一个或多个可包括或者进一步耦接到由固态存储器组成的存储器,以保存持久数据。在其它实施例中,存储装置 176a-176m 中的一个或多个可包括或者进一步耦接到利用其它技术(比如自旋转转移技术、磁阻随机存取存储器(MRAM)技术、瓦状盘、忆阻器、相变存储器或者其它存储技术)的存储器。这些不同的存储方法和技术会导致存储装置之间不同的 I/O 特性。

[0030] 在一个实施例中,包含的固态存储器包括固态驱动器(SSD)技术。通常,SSD 技术利用快闪存储器单元。本领域中众所周知,快闪存储器单元根据捕获并保存在浮棚中的电子的范围,保持二进制值。完全擦除的快闪存储器单元在浮棚中不保存或者保存极少量的电子。特定的二进制值,比如单阶存储单元(SLC)闪存的二进制值 1 与擦除的快闪存储器单元相关。多阶存储单元(MLC)闪存具有与擦除的快闪存储器单元相关的二进制值 11。在对快闪存储器单元内的控制栅施加比给定阈电压高的电压之后,快闪存储器单元在浮棚中捕获给定范围的电子。因而,另一个特定的二进制值,比如 SLC 闪存的二进制值 0,与被编程(写入)的快闪存储器单元相关。取决于对控制栅施加的电压,MLC 闪存单元可具有与被编程的存储器单元相关的多个二进制值之一。

[0031] HDD 技术和 SSD 技术之间在技术和机制方面的差异会导致数据存储装置 176a-176m 的输入 / 输出(I/O)特性方面的差异。一般来说,SSD 技术提供比 HDD 技术低的读取访问等待时间。不过,SSD 的写入性能通常慢于读取性能,并会受到 SSD 内的自由的可编程区块的可用性的极大影响。由于 SSD 的写入性能显著慢于 SSD 的读取性能,因此,就预期与读取类似的等待时间的某些功能或操作来说,会出现问题。另外,影响读取等待时间的长写入等待时间会使调度更困难。因而,对于各个数据存储阵列 120a-120b 中的 I/O 调度,可以使用不同的算法。

[0032] 在其中诸如读取操作和写入操作之类的不同种类的操作具有不同等待时间的一个实施例中,I/O 调度用算法可隔离这些操作,并单独处理这些操作,以便调度。例如,在存储装置 176a-176m 中的一个或多个存储装置中,装置本身可以批处理写入操作,比如通过把写入操作保存在内部高速缓存中。当这些高速缓存达到给定的占用率阈值时,或者在某个其它时间,对应的存储装置 176a-176m 可清洗高速缓存。通常,这些高速缓存清洗会在不可预测的时间,对读取和 / 或写入引入额外的等待时间,从而导致难以有效地调度操作。于是,I/O 调度器可以利用存储装置的特性,比如高速缓存的大小,或者测量的空闲时间,以便预测这样的高速缓存清洗会在何时发生。了解一个或多个存储装置 176a-176m 中的每一个的特性会导致更有效的 I/O 调度。在一个实施例中,全局 I/O 调度器 178 可能检测到一个或多个存储装置 176a-176m 中的给定装置在不可预测的时间,对 I/O 请求表现出较长的响应时间。作为响应,全局 I/O 调度器 178 可调度对所述给定装置的给定操作,以使该装置恢复表现出预期的行为。在一个实施例中,这样的操作可以是高速缓存清洗命令、修剪命令、擦除命令等。下面将讨论关于 I/O 调度的更多细节。

[0033] 网络体系结构的组件

[0034] 再次,如图所示,网络体系结构 100 包括通过网络 180 和 190 彼此互连,并且连接

到数据存储阵列 120a-120b 的客户端计算机系统 110a-110c。网络 180 和 190 可包括各种技术,包括无线连接、直接局域网 (LAN) 连接、诸如因特网之类的广域网 (WAN) 连接、路由器、存储区域网、以太网等等。网络 180 和 190 可包括也可以是无线的一个或多个 LAN。网络 180 和 190 还可包括远程直接存储器存取 (RDMA) 硬件和 / 或软件、传输控制协议 / 网际协议 (TCP/IP) 硬件和 / 或软件、路由器、转发器、交换机、电网和 / 或其它。在网络 180 和 190 中,可以利用诸如光纤通道、以太网光纤通道 (FCoE)、iSCSI 之类的协议。交换机 140 可利用与网络 180 和 190 相关的协议。网络 190 可以与用于因特网 160 的一组通信协议,比如传输控制协议 (TCP) 和网际协议 (IP),或者说 TCP/IP,对接。交换机 150 可以是 TCP/IP 交换机。

[0035] 客户端计算机系统 110a-110c 代表许多的固定或移动计算机,比如桌上型个人计算机 (PC)、服务器、服务器群、工作站、膝上型计算机、手持计算机、服务器、个人数字助手 (PDA)、智能电话机等等。一般来说,客户端计算机系统 110a-110c 包括一个或多个处理器,所述处理器包含一个或多个处理器核心。每个处理器核心包括按照预先定义的通用指令集执行指令的电路。例如,可以选择 x86 指令集体体系结构。另一方面,可以选择 **Alpha®**、**PowerPC®**、**SPARC®** 或者任何其它通用指令集体体系结构。处理器核心可关于数据和计算机程序指令,访问高速缓冲存储器子系统。高速缓存子系统可以耦接到包括随机存取存储器 (RAM) 和存储装置的存储器层次结构。

[0036] 客户端计算机系统内的每个处理器核心和存储器层次结构都可连接到网络接口。除了硬件组件之外,每个客户端计算机系统 110a-110c 可包括保存在存储器层次结构内的基本操作系统 (OS)。基本 OS 可代表多种操作系统任意之一,比如 **MS-DOS®**、**MS-WINDOWS®**、**OS/2®**、**UNIX®**、**Linux®**、**Solaris®**、**AIX(R)®**、DART 等等。因而,基本 OS 可操作为向最终用户提供各种服务,和提供可操作为支持各种程序的运行的软件架构。另外,每个客户端计算机系统 110a-110c 可包括用于支持虚拟机 (VM) 的管理程序。本领域的技术人员众所周知,在桌上型计算机和服务器中可以利用虚拟化,以完全或者部分使诸如 OS 之类的软件与系统的硬件分离。虚拟化可向最终用户提供在相同机器上运行多个 OS 的错觉,每个 OS 都具有它自己的资源,以及可以访问在每个数据存储阵列 120a-120b 内的存储装置 176a-176m 上建立的逻辑存储实体 (例如, LUN)。

[0037] 每个数据存储阵列 120a-120b 可用于不同服务器(比如客户端计算机系统 110a-110c)之间的数据的共享。每个数据存储阵列 120a-120b 包括用于数据存储的存储子系统 170。存储子系统 170 可包含多个存储装置 176a-176m。这些存储装置 176a-176m 都可以是 SSD。控制器 174 可包含处理接收的读取 / 写入请求的逻辑。例如,至少可以在控制器 174 中执行上面简要说明的算法。随机存取存储器 (RAM) 172 可用于批处理操作,比如接收的写入请求。在各种实施例中,当批处理写入操作 (或者其它操作) 时,可以使用非易失性存储器 (例如, NVRAM)。

[0038] 保存在存储介质 130 中的基本 OS132、文件系统 134、任意 OS 驱动器 (未示出) 和其它软件可提供用于提供对文件的访问的功能,和这些功能的管理。基本 OS132 和 OS 驱动器可包含保存在存储介质 130 上,可由处理器 122 执行的指令,以便在存储子系统 170 中进

行与接收的请求对应的一个或多个存储器访问操作。图 1 中所示的系统通常包括一个或多个文件服务器和 / 或块服务器。

[0039] 每个数据存储阵列 120a-120b 可利用网络接口 124 连接到网络 180。类似于客户端计算机系统 110a-110c，在一个实施例中，网络接口 124 的功能可以包含在网络适配卡上。网络接口 124 的功能可以利用硬件和软件二者实现。随机存取存储器 (RAM) 和只读存储器 (ROM) 可被包含在网络接口 124 的网卡实现物上。一个或多个专用集成电路 (ASIC) 可用于提供网络接口 124 的功能。

[0040] 在一个实施例中，可以建立试图优化 I/O 性能的数据存储模型。在一个实施例中，所述模型至少部分基于存储系统内的存储装置的特性。例如，在利用固态存储技术的存储系统中，特定装置的特性可用于建立关于所述装置的模型，所述模型又可用于通报对应的 I/O 调度算法。例如，如果正在使用的特定存储装置表现出与读取等待时间相比，相对高的写入等待时间，那么在调度操作时，可以考虑这样的特性。注意，被认为相对高或低的事物可随给定系统、处理的数据的类型、处理的数据的数量、数据的定时等而变化。一般来说，系统可被编程，以确定什么构成低或高的等待时间，和 / 或什么构成这两者之间的显著差异。

[0041] 一般来说，为装置或者计算系统建立的任何模型将是不完整的。通常，在真实系统中，存在要考虑的太多变量，以至于不能完全模拟给定系统。在一些情况下，能够建立不完整，但是仍然有价值的模型。如下更充分所述，这里说明其中根据存储装置的特性，模拟存储装置的实施例。在各种实施例中，根据关于装置如何行为表现的某些预测，进行 I/O 调度。根据对装置的特性的了解，某些装置行为比其它装置行为更加可预测。为了更有效地调度操作，以便获得最佳的 I/O 性能，需要对系统的行为的更大控制。意外的或者不可预测的装置行为使得更难以调度操作。于是，建立试图使系统中的不可预测或者意外的行为降至最少的算法。

[0042] 图 2 提供被模拟的装置或系统，以及用于使装置或系统内的不可预测的行为减至最少的途径的概念图。在第一个方框 200 中，描述了理想情形。方框 200 中所示的是系统 204 和该系统的模型 202。在一个实施例中，系统可以是单一装置。另一方面，系统可包含许多装置和 / 或组件。如上所述，模型 202 可能不是它试图模拟的系统 204 的完整模型。然而，模型 202 捕捉对模型来说有意义的行为。在一个实施例中，模型 202 试图模拟计算存储系统。在理想情形 200 下，系统 204 的实际行为与模型 202 的行为“对齐”。换句话说，系统 204 的行为通常与模型 202 试图捕捉的那些行为一致。虽然系统行为 204 与模型 202 的行为一致，不过，系统行为通常更加可预测。从而，可以更有效地进行系统内的操作（例如，读取操作和写入操作）的调度。

[0043] 例如，如果期望优化读取响应时间，那么可调度读取，以致更及时地服务于它们，如果系统的其它行为相对可预测的话。另一方面，如果系统行为相对不可预测，那么当需要时，调度这些读取以提供结果的能力的置信度减小。方框 210 图解说明系统行为（较小的圆）未与系统的模型的行为（较大的圆）对齐的情形。在这种情况下，系统表现出超出模型的范围的行为。从而，系统行为不太可预测，操作的调度会变得不太有效。例如，如果在存储系统中使用固态存储装置，并且这些装置独立发起使装置以更大（或者未预料到）的等待时间服务请求的行动，那么可为该装置调度的任何操作也会经历更大或者未预料到的等待时间。这种装置操作的一个例子是内部高速缓存清洗。

[0044] 为了解决意外或者非调度的系统行为和对应的可变性能的问题,建立的模型可包括为了使系统恢复到不确定性较低的状态,它可采取的行动。换句话说,如果系统开始表现出降低模型的预测系统行为的能力的行为,那么模型已在其中加入了为了使系统恢复到特定的意外行为被消除,或者变得不太可能的状态,它可采取的某些行动。在所示的例子中,表示了试图把系统“移动”到更接近地对齐模型的状态的行动 212。行动 212 可被称为“反应性”行动或操作,因为它是响应检测到在模型之外的系统行为而进行的。在执行行动 212 之后,可以达到更理想的状态 220。

[0045] 虽然建立能够对不可预测的行为作出反应,从而把系统移动到更理想的状态的模型是合乎需要的,不过,这些不可预测的行为的存在仍然可能干扰有效的调度操作。于是,理想的是使未预料到的行为或事件的发生减至最少。在一个实施例中,建立一种模型,该模型包括用于避免或减少意外行为的发生的行动或操作。这些行动可被称为“主动性”行动或操作,因为通常主动地进行这些行动,以便避免某种行为或事件的发生,或者改变某种行为或事件的定时。图 2 中的方框 230 图解说明其中系统行为(较小的圆)在模型的行为(较大的圆)之内的情形。尽管如此,模型采取行动 232,以便按照使系统行为保留在模型内,并且可能被更理想地对齐的方式,移动系统行为。方框 230 中的系统行为可被看作接近它表现出在模型之外的行为的状态。在这种情况下,模型可具有认为系统正在接近这种状态的某种根据。例如,如果 I/O 调度器已把许多写入操作传送给给定装置,那么调度器可预料该装置可能在未来的某个时刻,进行内部高速缓存清洗操作。调度器可主动为该装置调度高速缓存清洗操作,以致在调度器选择的时间进行高速缓存清洗,而不是等着这种事件的发生。另一方面,或者除了上述之外,还可在任意时间,进行这样的主动性操作。尽管仍然发生高速缓存清洗,不过其发生不是出乎意外的,它已成为调度器进行的整个调度的一部分,从而可按照更加有效和智能的方式管理。在进行该主动性行动 232 之后,系统通常被认为处于更加可预测的状态 240。这是因为对该装置调度并进行高速缓存清洗,以及该装置独立地自发启动内部高速缓存清洗的可能性被降低(即,其高速缓存已被清洗)。通过在模型内结合反应性和主动性行动或操作二者,可以获得更高的系统可预测性,可同样地实现改进的调度。

[0046] 现在参见图 3,图中表示了进行 I/O 调度,以减少不可预测行为的方法 300 的一个实施例。包含在上述网络体系结构 100 和数据存储阵列 120a-120b 中的组件通常可按照方法 300 工作。本实施例中的各个步骤是按顺序表示的。不过,一些步骤可按和所示顺序不同的顺序发生,一些步骤可以同时进行,一些步骤可以与其它步骤结合,以及在另一个实施例中,一些步骤可不存在。

[0047] 在方框 302, I/O 调度器为一个或多个存储装置,调度读取操作和写入操作。在各种实施例中, I/O 调度器可保持用于每个存储装置的独立队列(物理地或者逻辑地)。另外, I/O 调度器可包括用于对应存储装置所支持的每种操作种类的独立队列。例如, I/O 调度器可至少保持用于 SSD 的独立的读取队列和独立的写入队列。在方框 304, I/O 调度器可监控所述一个或多个存储装置的行为。在一个实施例中, I/O 调度器可包括对应存储装置的模型(例如,行为类模型和 / 或至少部分以装置的模型为基础的算法),并从存储装置接收状态数据,以输入模型中。通过利用已知和 / 或观察的存储装置的特性,I/O 调度器内的模型可模拟和预测存储装置的行为。

[0048] I/O 调度器可以检测影响或者可能影响 I/O 性能的给定存储装置的特性。例如,如下进一步所述,可以保持装置的各种特性和状态,以及 I/O 流量的各种特性和状态。通过观察这些特性和状态,I/O 调度器可预测给定装置不久会进入表现出高 I/O 等待时间的行为的状态。例如,在一个实施例中,I/O 调度器可检测或者预测在存储装置内,将发生影响对存储装置的请求的响应时间的内部高速缓存清洗。例如,在一个实施例中,持续给定时间量处于空闲状态的存储装置可清洗其内部高速缓存。在一些实施例中,给定装置是否空闲可以基于装置之外的观察。例如,如果持续一段时间,还没有为装置调度操作,那么可认为该装置大约持续所述一段时间处于空闲状态。在这样的实施例中,基于装置内的内部启动的活动,该装置事实上可能是繁忙的。不过,在判定装置是否空闲时,不会考虑这种内部发起的活动。在其它实施例中,当判断装置是空闲还是繁忙时,可考虑装置的内部发起的活动。通过观察装置的行为,并且注意到装置已空闲给定一段时间,那么调度器可预测内部高速缓存清洗何时可能发生。在其它实施例中,调度器还具有轮询各个装置,以确定各个装置的各种状态或条件的能力。无论怎样,调度器可被配置成确定非调度行为,比如内部高速缓存清洗的可能性,并启动主动性操作,以便防止所述行为发生。这样,调度器控制装置和系统中的事件的定时,从而能够更好地调度操作。

[0049] 各种特性可以用作进行关于装置行为的预测的基础。在各种实施例中,调度器可以保持对应于存储装置的当前未决操作的状态和 / 或最近操作的历史。在一些实施例中,I/O 调度器可能知道装置内的高速缓存的大小和 / 或高速缓存策略,并保持发送给存储装置的写入请求的数目的计数。在其它实施例中,其它机制可用于确定装置内的高速缓存的状态(例如,对装置的直接轮询式访问)。另外,I/O 调度器可跟踪在发送给存储装置的写入请求中的数据的数量。I/O 调度器随后可检测何时写入请求的数目或者对应于写入请求的数据的总量达到给定阈值。如果 I/O 调度器检测到这样的条件(条件框 306),那么在方框 308,I/O 调度器可为该装置调度特定操作。这样的操作可通常对应于上面说明的主动性操作。例如,I/O 调度器可把高速缓存清洗命令放入对应队列中,以强迫存储装置在调度器选择的时间进行高速缓存清洗。另一方面,I/O 调度器可在队列中放入虚拟读取操作,以便判定对于存储装置的任何高速缓存清洗是否已完成。另外,调度器可查询装置,以获得状态信息(例如,空闲、繁忙等)。这些和其它特性和操作都是可能的,并是可设想的。另外,在各种实施例中,当重新使 SSD 就位时,可以调度主动性操作。在这样的实施例中,SSD 固件和 / 或映射表可能进入请求暂停或者持久缓慢的状态。可以仅仅重置驱动器,或者关断和接通驱动器的电源,以清除固件的障碍。不过,如果该状况持久(即,不能处理映射表的当前状态的固件中的程序缺陷),那么另一种修复方式是重新格式化驱动器,以完全清除和重置 FTL,随后重新填充它或者把它用于某些其它数据。

[0050] 可以进行上述行动,以避免或减少不可预测的可变响应时间的发生数。同时,I/O 调度器可检测在不可预测的时间,给定存储装置的任何可变行为的发生。如果 I/O 调度器检测到这样的条件(条件框 310),那么在方框 312,I/O 调度器可把操作放入存储装置的对应队列中。在这种情况下,所述操作通常对应于上面说明的反应性操作。所述操作既可用于减少存储装置提供可变行为的时间量,又可用于检测可变行为的结束。在各种实施例中,主动性和 / 或反应性操作通常包括能够使装置(至少部分)进入已知状态的任何操作。例如,启动高速缓存清洗操作会导致装置达到清空高速缓存状态。与其高速缓存不空的装置

相比,高速缓存空的装置不太可能启动内部高速缓存清洗。主动性和 / 或反应性操作的一些例子包括高速缓存清洗操作、擦除操作、安全擦除操作、修剪操作、睡眠操作、休眠操作、接通和关断电源、重置操作。

[0051] 现在参见图 4,图中表示了隔离发给存储装置的操作的方法 400 的一个实施例。本实施例中的各个步骤是按顺序表示的。不过,一些步骤可按和所示顺序不同的顺序发生,一些步骤可以同时进行,一些步骤可以与其它步骤结合,在另一个实施例中,以及一些步骤可以不存在。在各种实施例中,可以使第一种类的操作和第二种类的操作分离,以便调度。例如,在一个实施例中,第一种类的操作可被赋予高于第二种类的操作的调度优先级。在这样的实施例中,可以调度第一种的操作,以便相对快地处理,同时使第二种的操作排队,以便稍后处理(实际上,推迟这些操作的处理)。在给定时刻,第一种类的操作的处理可被暂停,同时处理先前排队的(第二种)操作。随后,可以再次停止第二种的操作的处理,同时把处理优先权返还给第一种类的操作。何时暂停对一种操作的处理,而开始对另一种操作的处理可基于时段、累积的数据、事务频率、可用资源(例如,队列利用率)、上述的任意组合,或者酌情基于任意期望的条件。

[0052] 对于随机读取请求和写入请求,SSD 一般展示比 HDD 好的性能。不过,归因于 SSD 的特性,对于随机写入请求来说,SSD 一般表现出比读取请求差的性能。不同于 HDD,读取请求和写入请求的相对等待时间相当不同,写入请求一般用时明显长于读取请求,因为与读取快闪存储器单元相比,它要用更长的时间对快闪存储器单元编程。另外,归因于需要作为写入的一部分而执行的附加操作,写入操作的等待时间相当可变。例如,对于已被修改的快闪存储器单元,可在写入或编程操作之前,进行擦除操作。另外,可基于区块地进行擦除操作。在这种情况下,区块(擦除片段)内的所有快闪存储器单元被一起擦除。由于区块较大,包含多页,因此操作会花费较长的时间。另一方面,FTL 可把区块重新映射到已被擦除的擦除区块。在任何一种情况下,与进行写入操作相关的附加操作都会导致与读取相比,写入具有相当高的等待时间可变性,以及相当长的等待时间。其它存储装置会根据请求类型而表现出不同的特性。除了上面所述之外,如果读取请求和写入请求被混合,那么某些存储装置会提供差和 / 或可变的性能。于是,为了改善性能,各种实施例可分离读取请求和写入请求。注意,尽管上述讨论特别谈到读取操作和写入操作,不过,这里说明的系统和方法也适用于其它操作。在这样的其它实施例中,可同样地识别并分离其它相对高和低等待时间的操作,以便调度。另外,在一些实施例中,读取和写入可被归类为第一种类的操作,而诸如高速缓存清洗和修剪操作之类的其它操作可被归类为对应于第二种类的操作。各种组合都是可能的,并是可以构思的。

[0053] 在方框 402, I/O 调度器可接收和缓存对一个或多个存储装置中的给定存储装置的 I/O 请求。在方框 404,低等待时间的 I/O 请求通常可优先于高等待时间的请求,被发送给存储装置。例如,取决于存储装置使用的存储技术,读取请求可具有比写入请求和其它命令类型低的等待时间,从而可首先发出。因此,写入请求可被累积,同时读取请求被赋予发送优先级(即,在写入请求之前,被传送给装置)。在某个时间点,I/O 调度器可停止向装置发送读取请求,而开始发送写入请求。在一个实施例中,可以一连串的多个写入的形式,发送写入请求。于是,可以在多个写入请求内,分摊与写入请求相关的开销。按照这种方式,可以隔离从而单独处理高等待时间请求(例如,写入请求)和低等待时间请求(例如,读取

请求)。

[0054] 在方框 406, I/O 调度器判定是否存在指示应把高等待时间请求传送给装置的特定条件。例如,在一个实施例中,检测这样的条件可包括检测给定数目的高等待时间 I/O 请求,或者许多的数据已累积并且达到给定阈值。另一方面,收到的高等待时间请求的比率可达到某个阈值。众多的这种条件都是可能的,和是可以预想的。在一个实施例中,高等待时间请求是写入请求。如果出现这样的条件(条件框 408),那么在方框 410, I/O 调度器可开始把高等待时间 I/O 请求发送给给定的存储装置。发送的这种请求的数目可随给定算法而变化。所述数目可对应于固定数目或者可编程数目的写入,或者数据的数量。另一方面,可以持续给定的一段时间,发送写入。例如,所述一段时间可持续到所述特定条件停止存在(例如,接收的写入请求的比率降低),或者出现特定条件为止。另一方面,任意上述的组合可用于确定何时开始和何时停止把高等待时间请求发送给装置。在一些实施例中,与其它读取请求相比,在一连串的写入请求之后的第一个读取请求可能较慢。为了避免在紧接在一连串的写入请求之后的发送时隙中,调度“真实的”读取请求,I/O 调度器可被配置成在一连串的写入请求之后,自动调度“虚拟的”读取。在这种情况下,“真实的”读取是用户或应用程序请求的数据的读取,而“虚拟的”读取是人为产生的其数据可被丢弃的读取。在各种实施例中,在检测到虚拟读取结束之前,不会确定写入请求已完成。另外,在各种实施例中,高速缓存清洗可以跟随一连串的写入请求,并用于确定何时完成了写入。

[0055] 现在参见图 5,图中表示了建立表征存储子系统中的存储装置的行为的模型的方法 500 的一个实施例。本实施例中的步骤是按顺序表示的。不过,一些步骤可按和所示顺序不同的顺序发生,一些步骤可以同时进行,一些步骤可以与其它步骤结合,以及在另一个实施例中,一些步骤可以不存在。

[0056] 在方框 502,可以选择将在存储子系统中使用的一个或多个存储装置。在方框 504,可以识别每个装置的各种特性,比如高速缓存大小、一般的读取和写入响应时间、存储拓扑、装置的使用年限等等。在方框 506,可以识别影响给定存储存储装置的 I/O 性能的一个或多个特性。

[0057] 在方框 508,可以确定影响给定装置的特性的定时和 / 或发生的一种或多种行动。例子可包括高速缓存清洗,和给定操作(比如对 SSD 的擦除操作)的执行。例如,诸如高速缓存清洗之类的强制操作可减少在不可预测时间的 SSD 的可变响应时间的发生。在方框 510,可根据对应的特性和行动,为一个或多个选择的装置中的每个装置,建立模型。该模型可以用在软件中,比如在存储控制器内的 I/O 调度器内使用。

[0058] 现在参见图 6,图中表示了存储子系统的一个实施例的广义方框图。在所示的实施例中,每个存储装置 176a-176m 都被显示在单个装置组内。不过,在其它实施例中,一个或多个存储装置 176a-176m 可被划分到装置组 173a-173m 中的两个或者更多的装置组中。在装置单元 600a-600w 中,可以包括每个存储装置的一个或多个对应的操作队列和状态表。这些装置单元可被保存在 RAM172 中。对于每一个装置组 173a-173m,可以包括对应的 I/O 调度器 718。每个 I/O 调度器 178 可包括跟踪对应的装置组内的每个存储装置的状态数据的监控器 610。调度逻辑 620 可进行把哪些请求发送给对应的存储装置的判定,和确定发送请求的定时。

[0059] 现在参见图 7,图中表示了装置单元 600 的一个实施例的广义方框图。装置单元

600 可包含装置队列 710 和表格 720。装置队列 710 可包括读取队列 712, 写入队列 714 和一个或多个其它的队列, 比如其它操作队列 716。每个队列可包含用于保存一个或多个对应请求的多个条目 730。例如, 对应 SSD 的装置单元可包括至少保存读取请求、写入请求、修剪请求、擦除请求等的队列。表格 720 可包含一个或多个状态表 722a-722b, 每个状态表包含用于保存状态数据的多个条目 730。在各种实施例中, 图 7 中所示的队列可被物理和 / 或逻辑地隔离。另外注意, 尽管队列和表格被显示成包括特定数目的条目, 不过, 条目本身不一定相互对应。另外, 队列和表的数目可不同于图中所示的数目。另外, 给定队列内的或者跨队列的条目可被区分优先顺序。例如, 读取请求可以具有影响把所述请求发送给装置的顺序的高、中或低优先级。另外, 这样的优先级是可随各种条件而变化的。例如, 达到一定年限的低优先级读取可以使其优先级被提高。众多这样的优先化方案和技术已为本领域的技术人员所知。所有这样的途径都是可预期的, 以及可以结合这里说明的系统和方法使用。

[0060] 现在参见图 8, 图中表示了图解说明状态表, 比如图 7 中所示的状态表的一个实施例的广义方框图。在一个实施例中, 这样的表格可包括对应于给定存储装置的状态、差错、磨损水平信息和其它信息的数据。对应的 I/O 调度器可以访问该信息, 从而使 I/O 调度器可以更好地调度对存储装置的 I/O 请求。在一个实施例中, 所述信息可以包括装置使用年限 802、差错率 804、在装置上检测到的差错的总数 806、可恢复的差错的数目 808、不可恢复的差错的数目 810、装置的存取速率 812、保存的数据的年限 814、对应的高速缓存大小 816、对应的高速缓存清洗空闲时间 818、分配空间的一个或多个分配状态 820-822、并发数 824、和各种操作的预期时间 826 中的至少一个或多个。分配状态可包括充满、空闲、错误等。给定装置的并发数可包括关于装置同时处理多个操作的能力的信息。例如, 如果装置具有 4 个闪存芯片, 并且每个芯片每次能够进行一个传送, 那么该装置能够进行最多 4 个并行操作。是否并行地进行特定操作可取决于在该装置上数据是如何布置的。例如, 如果装置内的数据被布置成以致请求所访问的数据都在一个芯片上, 那么可以与访问不同芯片上的数据的请求并行地进行对所述数据的操作。不过, 如果请求所访问的数据被分解在多个芯片上, 那么请求会相互干扰。从而, 装置能够进行最多 N 个并行 / 并发操作 (例如, 在上面说明的装置具有 4 个芯片的情况下, 4 个并行操作)。另一方面, 最大并发数可以基于所涉及的操作的种类。无论如何, 当调度操作时, 调度器可考虑保存的表示并发数 N 和未决事务数 M 的信息。

[0061] 现在参见图 9, 图中表示了调整 I/O 调度, 以减小数据存储子系统上的不可预测的可变 I/O 响应时间的方法 900 的另一个实施例。包含在上述网络体系结构 100 和数据存储阵列 120a-120b 中的组件通常可按照方法 900 工作。为了讨论起见, 本实施例中的各个步骤是按顺序表示的。不过, 一些步骤可按和所示顺序不同的顺序发生, 一些步骤可以同时进行, 一些步骤可以与其它步骤结合, 以及在另一个实施例中, 一些步骤可以不存在。

[0062] 在方框 902, I/O 调度器可监控存储装置中的每个存储装置的行为。条件框 904-908 图解说明如上关于方法 300 的条件步骤 306 所述的, 检测给定装置的可能影响 I/O 性能的特性的一个实施例。在一个实施例中, 如果 I/O 调度器检测到给定装置超过给定的空闲时间 (条件框 904), 或者检测到对应的高速缓存超过占用率阈值 (条件框 906), 或者检测到高速缓存的数据超过数据年限阈值 (条件框 908), 那么在方框 910, I/O 调度器可向该给定存储装置发出强制 (主动性) 操作。在这种情况下, 调度器可预测不久将在不可预

测的时间,发生内部高速缓存清洗。为了避免这种事件的发生,I/O 调度器主动调度操作,以规避该事件。

[0063] 注意,如上所述的事件的规避可意味事件不发生,或者不在不可预测或意外的时间发生。换句话说,调度器通常宁愿给定事件按照调度器的定时发生,而不是按其它方式发生。在这个意义上,因调度器调度长等待时间的事件而发生该事件好于意外地发生这样的事件。可以与监控器 610 结合地利用调度逻辑 620 内的定时器和计数器,以至少进行这些检测。发送给给定存储装置的强制操作的一个例子可包括高速缓存清洗。强制操作的另一个例子可包括擦除请求。作为调度的一部分,可从 I/O 调度器向在对应的装置单元 600 内的装置队列 710 中的对应队列发送强制操作。

[0064] 现在参见图 10,图中表示了在共享数据存储器上,维持等待时间较低的读取操作的方法 1000 的一个实施例。包含在上述网络体系结构 100 和数据存取阵列 120a-120b 中的组件通常可按照方法 1000 工作。为了讨论起见,本实施例中的各个步骤是按顺序表示的。不过,一些步骤可按和所示顺序不同的顺序发生,一些步骤可以同时进行,一些步骤可以与其它步骤结合,以及在另一个实施例中,一些步骤可以不存在。

[0065] 在方框 1002,可以确定存储子系统的 RAID 体系结构中的冗余量,以便在给定装置组 173 内使用。例如,对于 4+2RAID 组来说,存储装置中的 2 个可用于保存纠删码 (ECC) 信息,比如奇偶校验信息。该信息可用作重建读取请求的一部分。在一个实施例中,在检测到许多存储装置表现出可变 I/O 响应时间的时候,可在正常 I/O 调度期间使用重建读取请求,以改善装置组的性能。在方框 1004,确定装置组内可能同时繁忙,或者表现出可变响应时间的装置的最大数。该最大数可被称为目标数。在一个实施例中,存储装置是由于执行写入请求、擦除请求或高速缓存清洗,可能表现出可变响应时间的 SSD。在一个实施例中,选择目标数,以致仍然能够进行重建读取。

[0066] 在一个实施例中,I/O 调度器可检测到为把目标数升高到重建请求不再有效的程度提供充分根据的条件。例如,对于给定装置的未决写入请求的数目可能达到等待阈值(即,写入请求已持续相当长一段时间悬而未决,并且确定这些写入请求不应再等待)。另一方面,可能检测到如上所述,不能被累积以便稍后发送的优先级较高的给定数目的写入请求。如果 I/O 调度器检测到这样的条件(条件框 1006),那么在方框 1008 中,I/O 调度器可根据一个或多个检测到的条件,递增或递减目标数。例如,如果适当数目的高优先级写入请求未决,或者发生了某个其它条件,那么 I/O 调度器可允许目标数超过支持的冗余量。在方框 1010,I/O 调度器可确定装置组内的 N 个存储装置表现出可变 I/O 响应时间。如果 N 大于目标数(条件框 1012),那么在方框 1014,可按照减小 N 的方式调度存储装置。否则,在方框 1016,I/O 调度器可按照改善性能的方式调度请求。例如,I/O 调度器可利用如下进一步说明的重建读取请求的能力。

[0067] 现在参见图 11,图中表示了减少表现出可变 I/O 响应时间的存储装置的数目的方法 1100 的一个实施例。本实施例中的步骤是按顺序表示的。不过,一些步骤可按和所示顺序不同的顺序发生,一些步骤可以同时进行,一些步骤可以与其它步骤结合,以及在另一个实施例中,一些步骤可以不存在。

[0068] 在方框 1102,I/O 调度器可确定减少存储子系统内执行会在不可预测时间导致可变响应时间的高等待时间操作的存储装置的数目 N。在方框 1104,I/O 调度器可选择执行高

等待时间操作的给定装置。在方框 1106, I/O 调度器可暂停给定装置上的高等待时间操作的执行，并递减 N。例如，I/O 调度器可停止向给定存储装置发送写入请求和擦除请求。另外，对应的 I/O 调度器可暂停发送的写入请求和擦除请求的执行。在方框 1108, I/O 调度器可启动给定装置上的低等待时间操作(比如读取请求)的执行。这些读取请求可包括重建读取请求。这样，装置脱离长等待时间响应状态，N 被减小。

[0069] 现在参见图 12, 图中表示了在共享数据存储器上维持具有有效等待时间的读取操作的方法的一个实施例。包含在上述网络体系结构 100 和数据存取阵列 120a-120b 中的组件通常可按照该方法工作。为了讨论起见，本实施例中的各个步骤是按顺序表示的。不过，一些步骤可按和所示顺序不同的顺序发生，一些步骤可以同时进行，一些步骤可以与其它步骤结合，以及在另一个实施例中，一些步骤可以不存在。

[0070] 图 12 的方法可代表为了进行方法 1000 中的步骤 1016 而采取的各个步骤的一个实施例。在方框 1201, I/O 调度器接收以正表现出可变响应时间行为的第一装置为目标的原始读取请求。第一装置可由于接收特定的调度操作(即，已知的原因)，或者由于某个未知的原因，表现出可变响应时间。在各种实施例中，可至少部分根据给定操作的预期等待时间，确定什么被视为可变响应时间。例如，根据装置的特性和 / 或最近的操作历史，可预计在给定的一段时间内，发生对给定读取的响应。例如，可以借助为反映容许的响应等待时间的范围而确定的增量，为装置确定平均响应等待时间。可以选择所述增量，以考虑到 99% 的事务，或者任何其它适当数目的事务。如果在预期的一段时间内没有收到响应，那么可以触发重建读取的启动。

[0071] 一般来说，是否启动重建读取可基于成本效益分析，成本效益分析比较与进行重建读取相关的成本和获得重建读取的结果的(潜在)益处。例如，如果在给定一段时间内，没有收到给定装置中对原始读取请求的响应，那么可以预测该装置正在进行会导致比将要启动的重建读取的等待时间长的等待时间的操作。于是，可以启动重建读取。可以采取这样的行动，以便(例如)维持给定水平的读取服务性能。注意当判定是否启动重建读取时，也可考虑其它因素，比如当前的负载、正接收的请求的种类、请求的优先级、系统中的其它装置的状态、如在图 7 和 8 中说明的各种特性，等等。此外注意，尽管可由于原始读取的较长响应等待时间而启动重建读取，不过预计原始读取请求事实上将会完成。事实上，原始读取和重建读取都可能成功地完成，并提供结果。从而，不需要重建读取，以便使原始请求得到服务。这与由出错条件，比如检测到等待时间和指出事务将(或者可能)不会成功完成的某种出错指示，而引起的等待时间相反。例如，由不能读取给定存储位置而引起的装置超时代表预期不会完成的响应。在这种情况下，为了服务请求，可能需要重建读取。因而，在各种实施例中，系统实际上包括关于给定装置的至少两种超时条件。第一种超时对应于之后即使不一定需要也可启动重建读取的一段时间。这样，重建读取可以作为无差错相关调度进程的普通部分，被并入调度算法中。在第一种超时之后出现的第二种超时代表之后认为出现了出错条件的一段时间。在这种情况下，由于预期指出所述差错的装置不会服务原始读取，也可启动重建读取。

[0072] 鉴于上面所述，I/O 调度器随后可判定是否要启动对应于原始读取的重建读取(判定框 1202)。重建读取一般会使一个或多个读取由除第一装置以外的装置服务。在判定是否要启动重建读取时，可以考虑许多因素。一般来说，I/O 调度器进行成本 / 效益分析，

以判定是尝试用第一装置服务原始读取“更佳”，还是尝试通过发出重建读取来服务原始读取“更佳”。如上所述，当判定是否启动重建读取时，可以考虑许多因素。给定情况下“更佳的”选择是可变化的，可编程的，和可动态确定的。例如，算法可能总是青睐更快的读取响应时间。在这种情况下，可以判定重建读取的服务是否能够（或者可能）在由原始装置服务原始读取之前完成。另一方面，算法可能确定在给定时间青睐降低的系统负载。在这种情况下，I/O 调度器可选择不利用其额外开销启动重建读取 – 即使重建读取会比原始读取更快地完成。此外，在这样的判定中，可以使用速度与开销的更加细微的均衡。在各种实施例中，算法是可用初始加权编程的（例如，总是宁愿选择速度，而不管负载）。这种加权可以是恒定的，或者可以是可编程以按照各种条件而动态变化。例如，条件可包括时间、接收的 I/O 请求的比率、接收的请求的优先级、是否检测到特定任务（例如，当时正在进行备份操作）、故障的发现，等等。

[0073] 如果调度器决定不启动重建读取，那么读取可由最初的目标装置服务（方框 1203）。另一方面，可以启动重建读取（方框 1204）。在一个实施例中，为了服务重建读取而选择的其它装置是被识别为表现出非可变行为的那些装置。通过选择表现出非可变行为（即，更加可预测的行为）的装置，I/O 调度器能够更好地预测该装置服务重建读取可能需要多少时间。除了装置的给定可变 / 非可变行为之外，I/O 调度器还可考虑每个装置的其它方面。例如，在选择用于服务重建读取的特定装置时，I/O 调度器还可评估给定装置的未完成请求的数目（例如，该装置的队列的充满程度），给定装置的当前未决的请求的优先级，装置本身的预期处理速度（例如，一些装置可代表比其它装置更陈旧或者否则固有地较慢的技术）等等。此外，调度器可能期望按照来自每个装置的对应结果在大约相同的时间被返回的方式，调度重建读取。在这种情况下，调度器可能不赞成特定装置服务重建读取，如果预测所述特定装置的处理时间会明显不同于其它装置的处理时间的话 – 即使所述特定装置比其它装置快得多。许多这样的考虑因素和条件都是可能的，和是可设想的。

[0074] 在一个实施例中，重建读取请求可继续原始读取请求的优先级。在其它实施例中，重建读取请求可具有不同于原始读取请求的优先级。如果 I/O 调度器检测到接收对应的重建读取请求的所选第二（另一个）装置现在正表现出可变的响应时间行为（条件框 1205），并且预计第二装置依然可变，直到在预计第一装置变成非可变之后为止（条件框 1206），那么在方框 1208 中，I/O 调度器可把原始读取请求发给第一装置。在一个实施例中，可以利用定时器预测表现出可变响应时间的存储装置何时会再次提供非可变响应时间。方法 1200 的控制流程从方框 1208 经方框 C 转到条件框 1212。如果未预计第二装置比第一装置时间更长地保持可变状态（条件框 1206），那么方法 1200 的控制流程转到方框 1210。在方框 1210，利用发出的重建读取请求，服务读取请求。

[0075] 如果 I/O 调度器检测到给定的可变装置变成非可变装置（条件框 1212），那么在方框 1214，I/O 调度器把原始读取请求发给所述给定装置。I/O 调度器可把该给定装置指定为非可变，并递减 N（检测的提供可变 I/O 响应时间的存储装置的数目）。如果原始读取请求在备选的重建读取请求之前结束（条件框 1216），那么在方框 1218，I/O 调度器利用原始读取请求，服务读取请求。在各种实施例中，调度器可以除去重建读取请求。另一方面，可以完成重建读取请求，但可仅丢弃它们的数据。否则，在方框 1220，I/O 调度器利用重建读取请求服务读取请求，并可除去原始请求读取（或者丢弃原始读取请求的返回数据）。

[0076] 注意,上述实施例可包含软件。在这样的实施例中,实现所述方法和 / 或机制的程序指令可被传送或者保存在计算机可读介质上。可以利用配置成保存程序指令的各种介质,包括硬盘、软盘、CD-ROM、DVD、快闪存储器、可编程 ROM(PROM)、随机存取存储器 (RAM)、和各种其它形式的易失性或非易失性存储器。

[0077] 在各种实施例中,这里说明的方法和机制的一个或多个部分可以构成云计算环境的一部分。在这样的实施例中,可以按照各种模型中的一个或多个模型,以服务的形式通过因特网提供资源。这样的模型可包括基础架构即服务 (IaaS), 平台即服务 (PaaS) 和软件即服务 (SaaS)。在 IaaS 中,以服务的形式提供计算机基础架构。在这种情况下,计算设备通常由服务提供商所有和操作。在 PaaS 模型中,开发人员用于建立软件解决方案的软件工具和底层设备可以服务的形式提供,并由服务提供商托管。SaaS 一般包括作为按需服务的服务提供商授权软件。服务提供商可托管所述软件,或者可以持续给定的一段时间向消费者部署所述软件。上述模型的众多组合都是可能的,和是可以预想的。另外,虽然上面的说明聚焦于网络化存储设备和控制器,不过,上面所述的方法和机制也可用在具有直接附连存储设备的系统、主机操作系统等中。

[0078] 尽管相当详细地说明了上述实施例,不过一旦充分理解上述公开,对本领域的技术人员来说,各种变化和修改将变得显而易见。以下权利要求应被理解为包括所有这样的变化和修改。

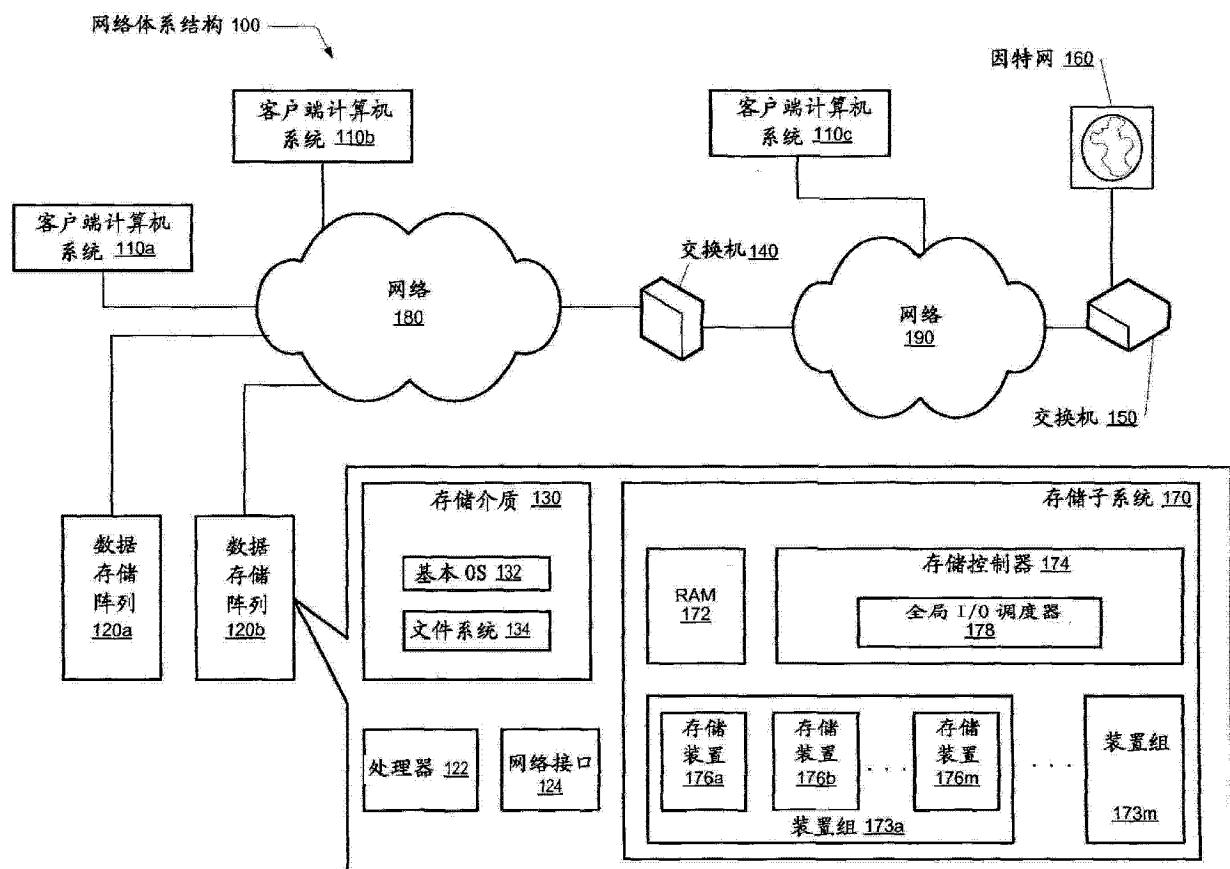


图 1

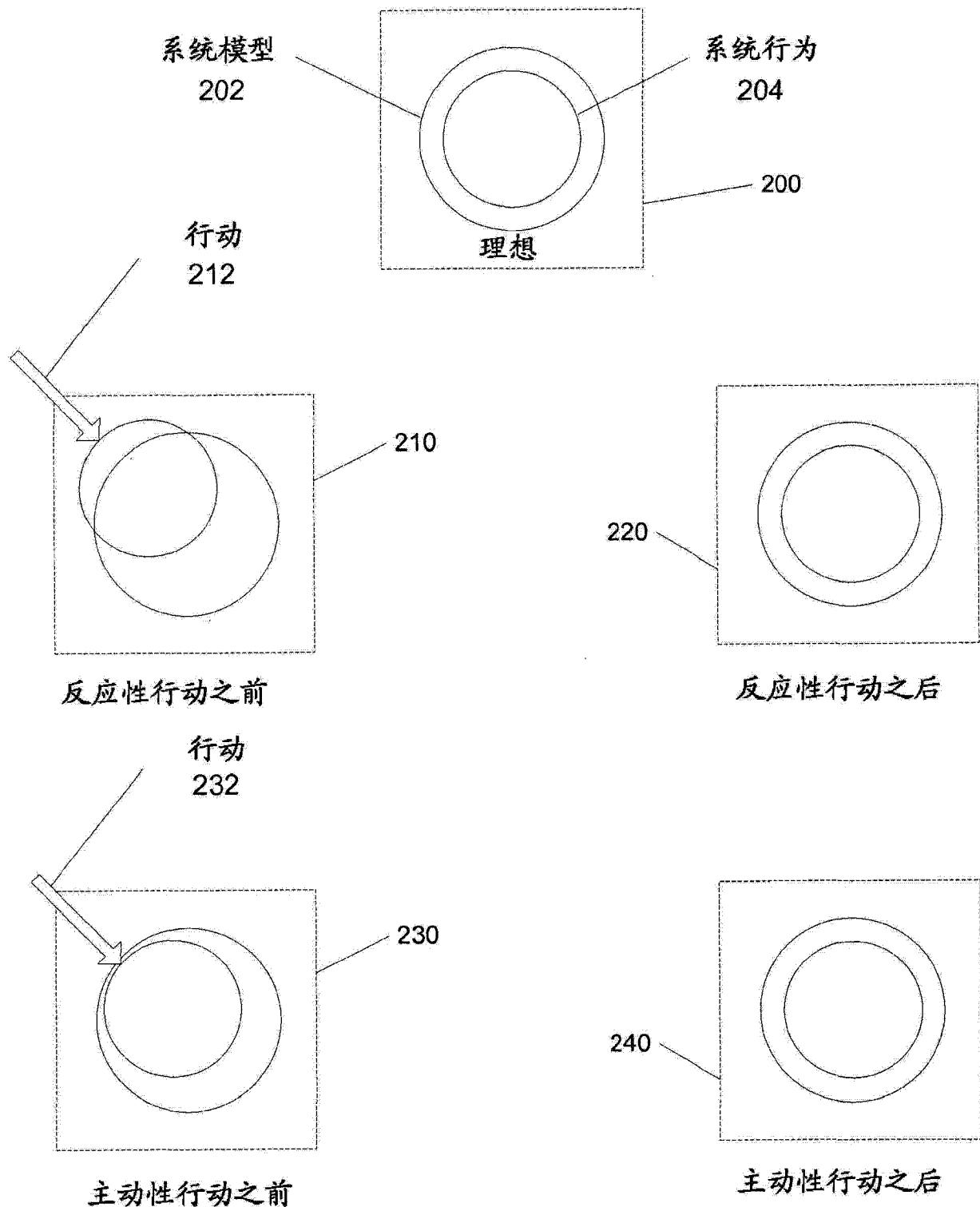


图 2

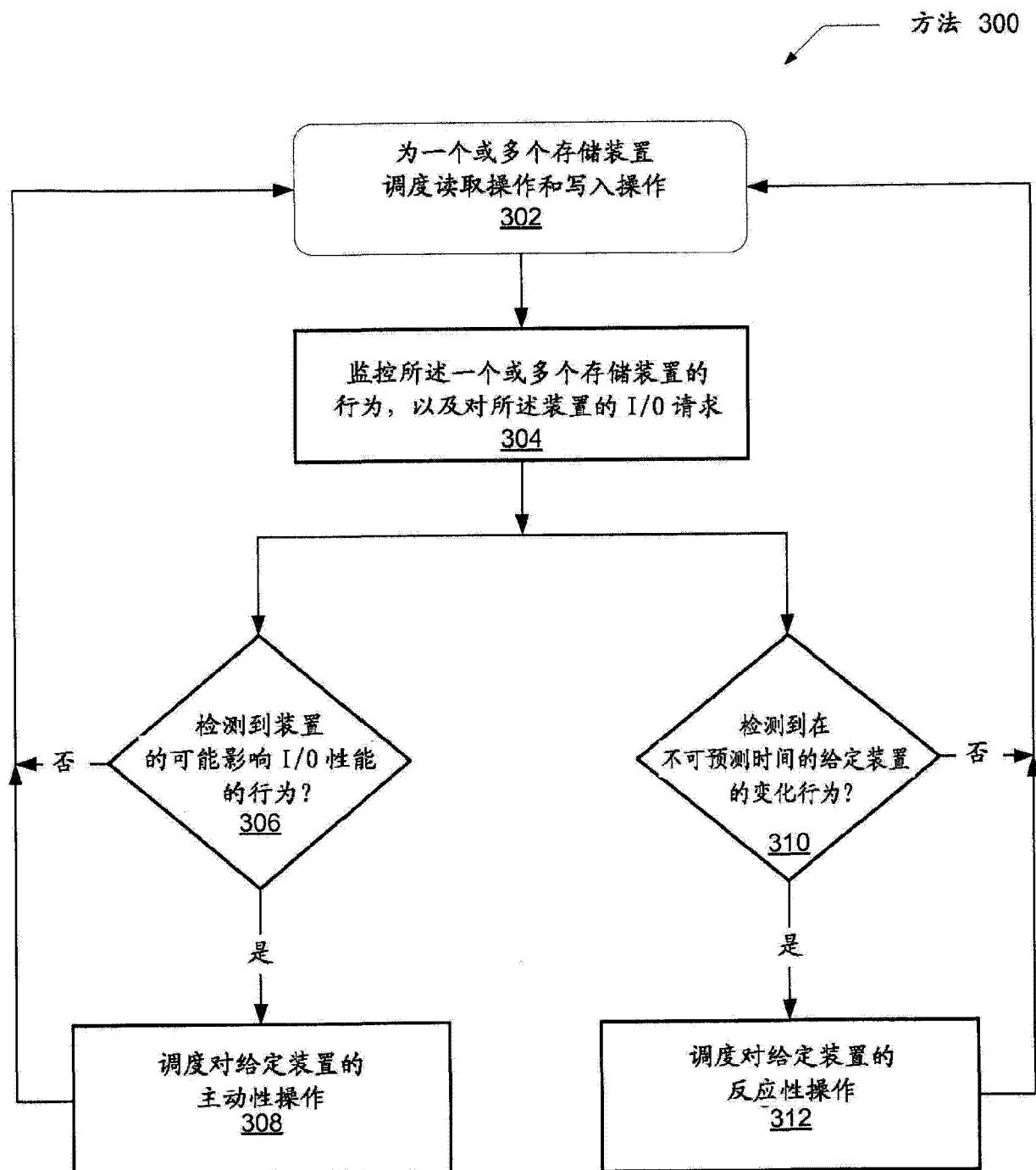


图 3

方法 400

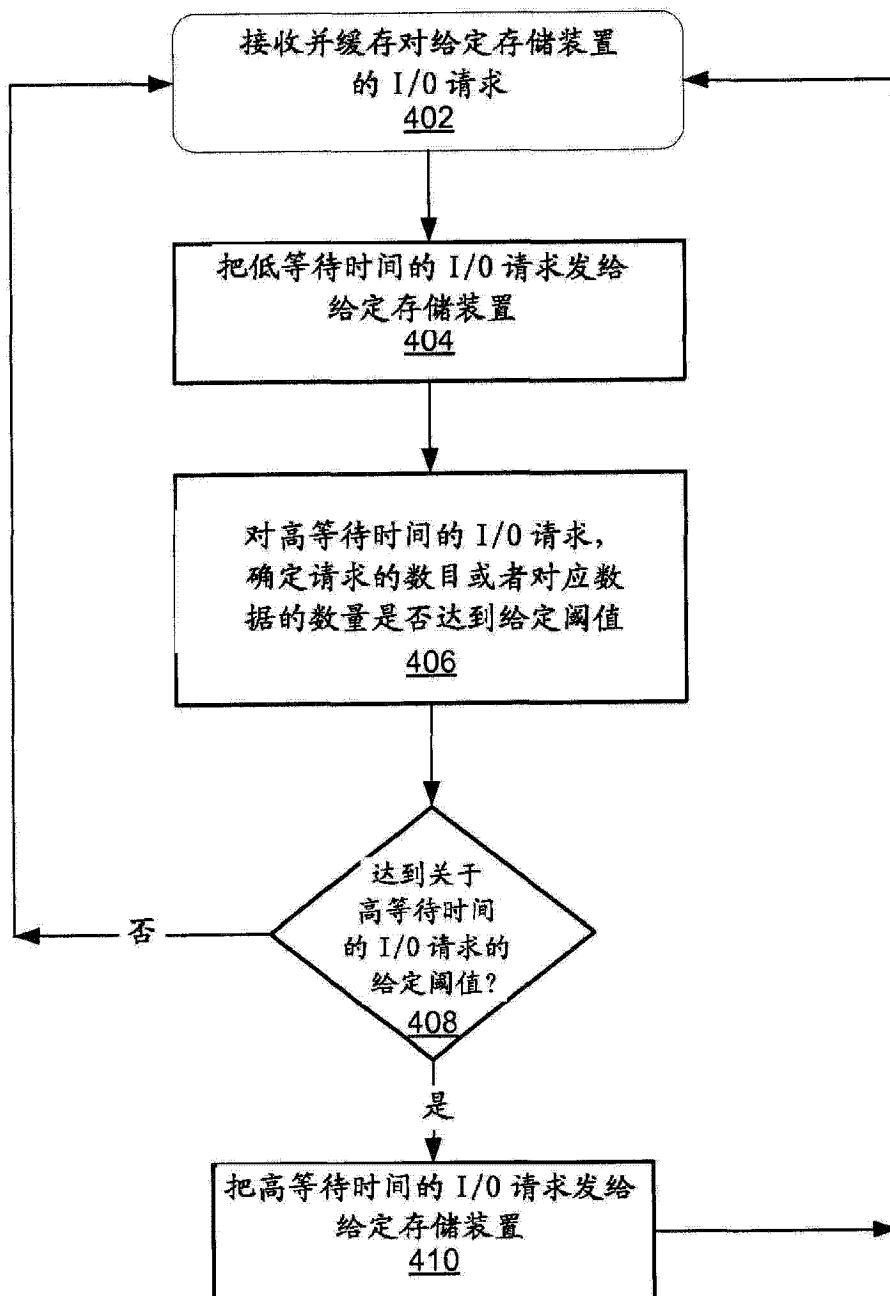


图 4

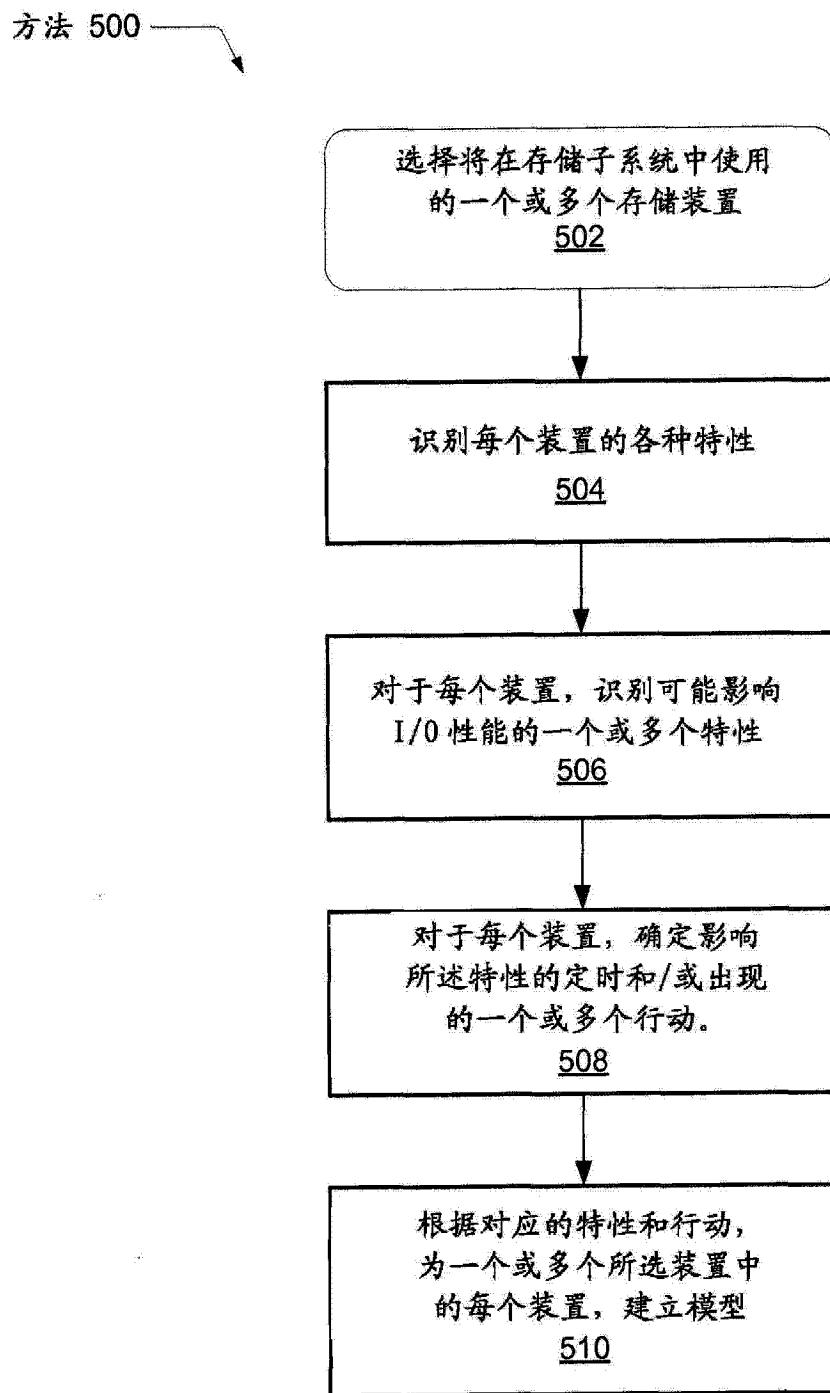


图 5

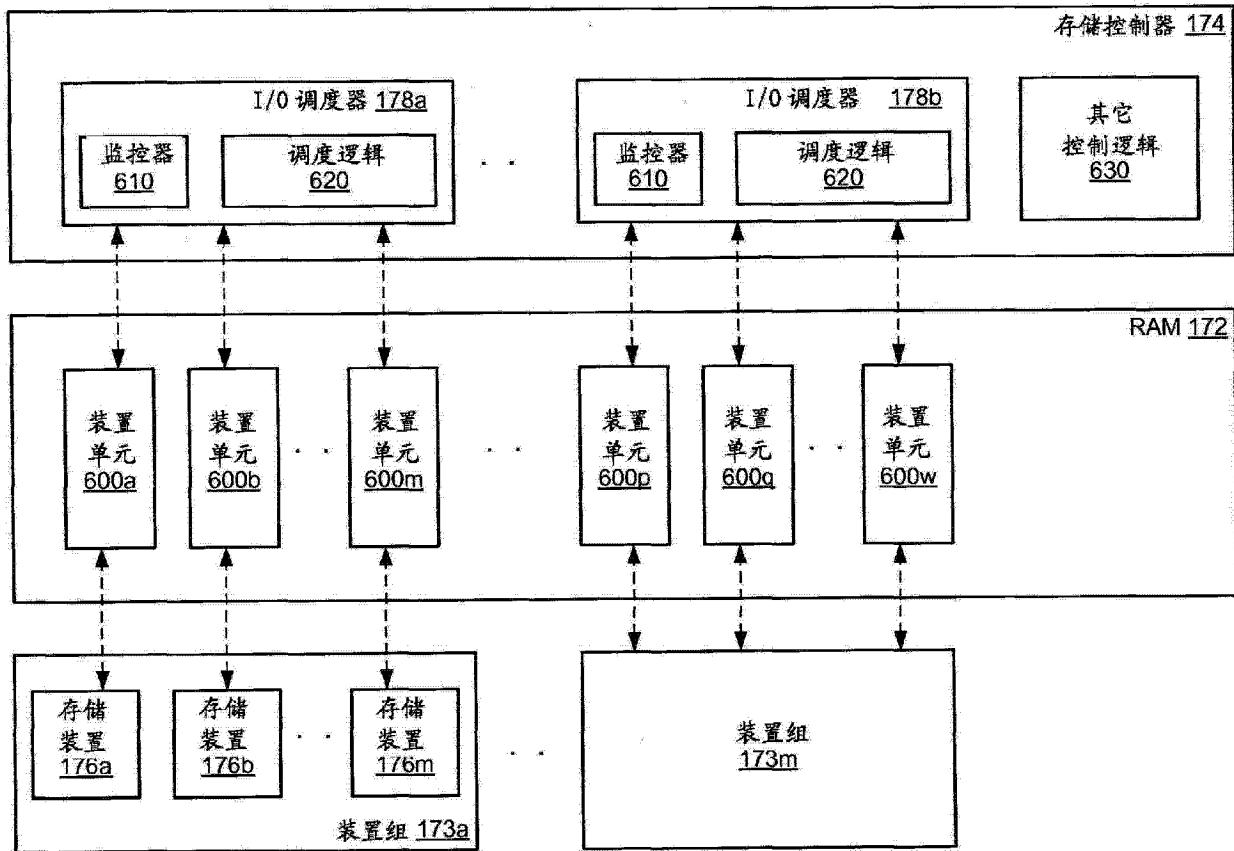


图 6

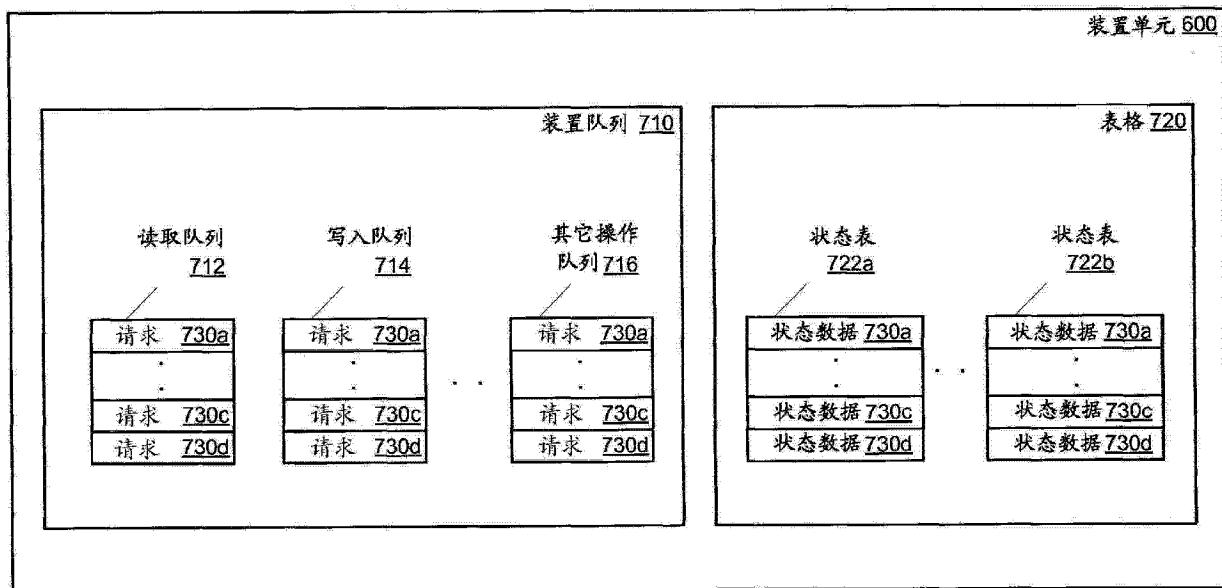


图 7

状态表 722

装置使用年限 802
差错率 804
错误总数 806
可恢复错误的数目 808
不可恢复错误的数目 810
访问速率 812
数据年限 814
高速缓存大小 816
高速缓存清洗空闲时间 818
第一分配空间的分配状态 820
第二分配空间的分配状态 822
并发数 824
预期时间 826

图 8

方法 900

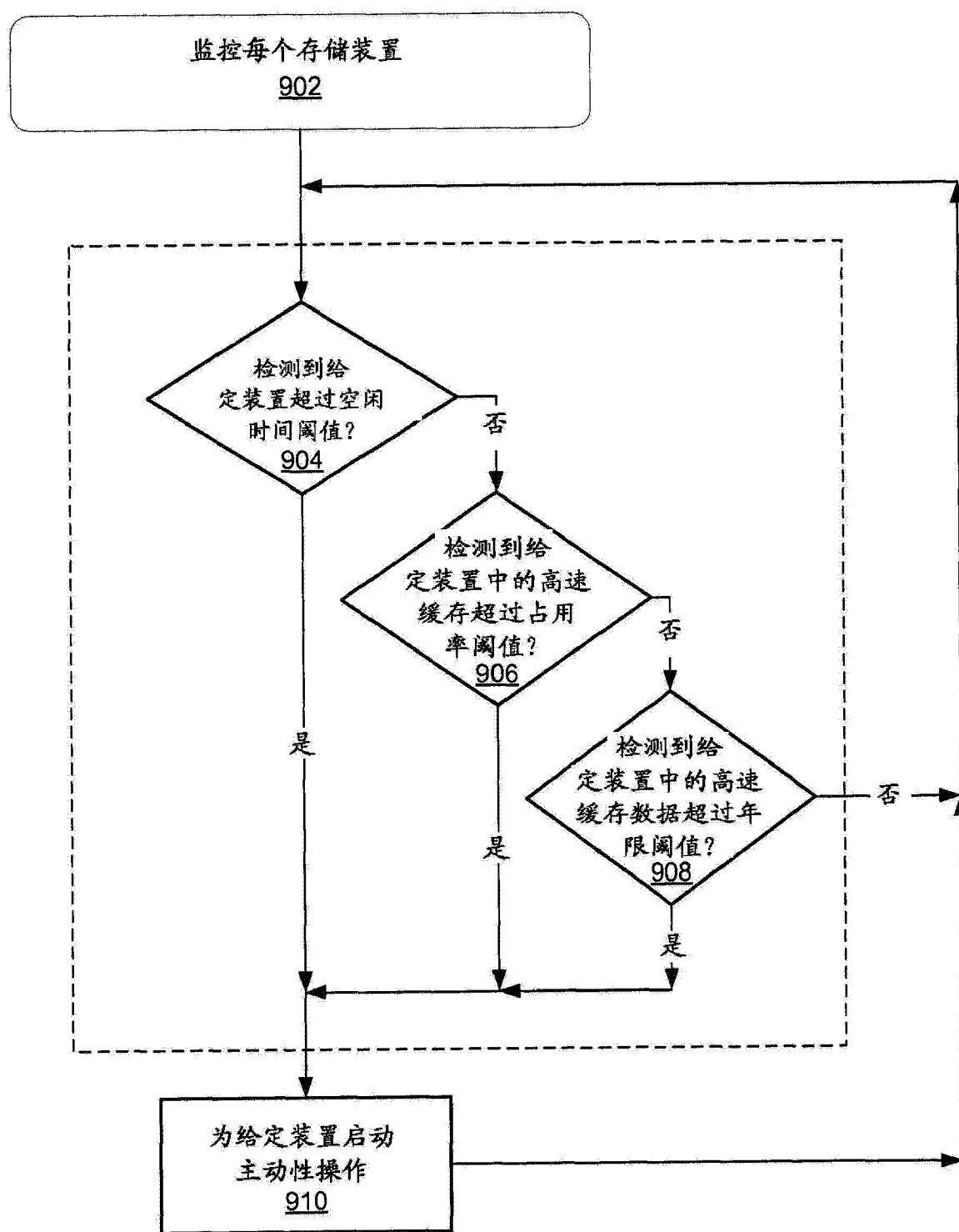


图 9

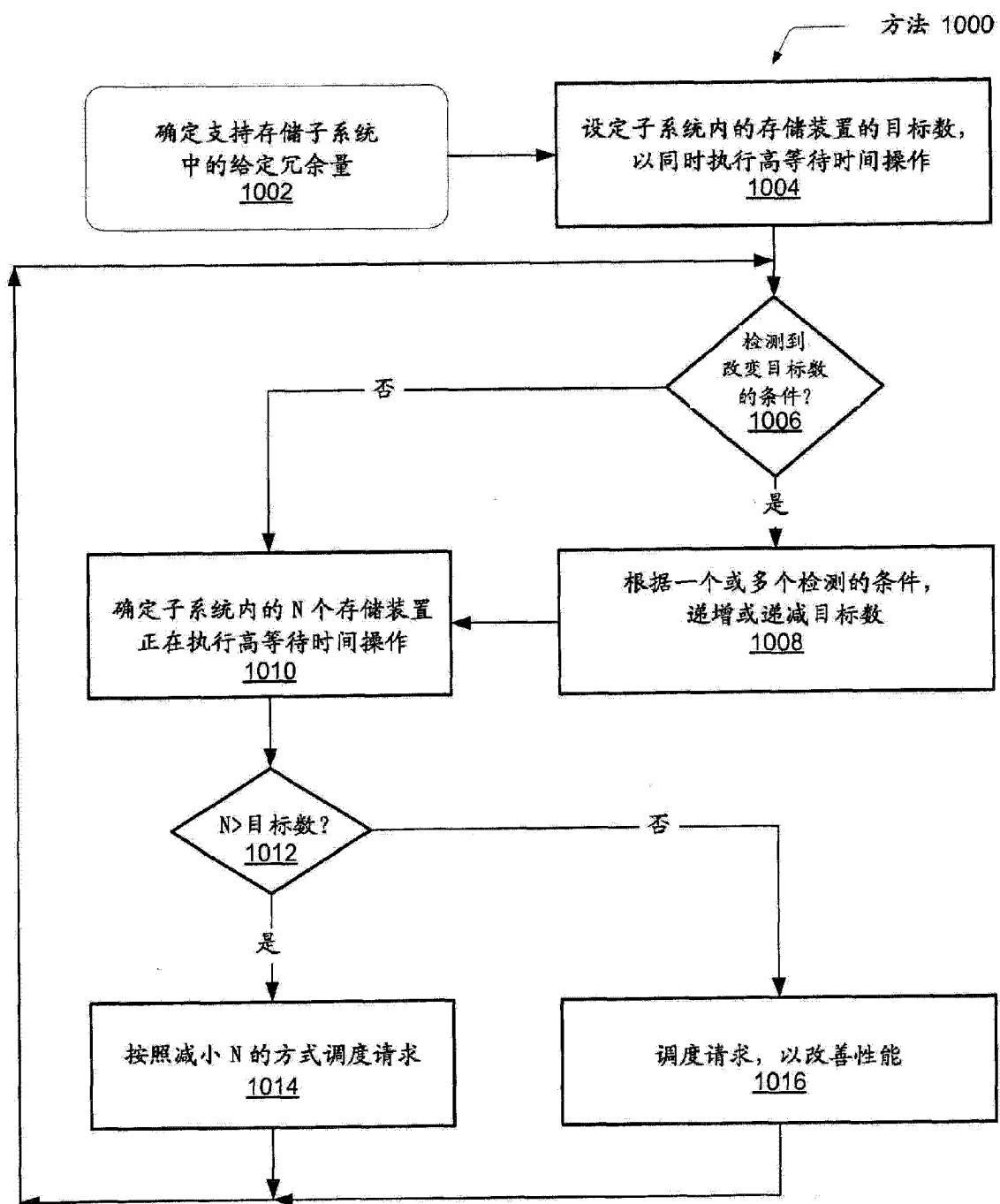


图 10

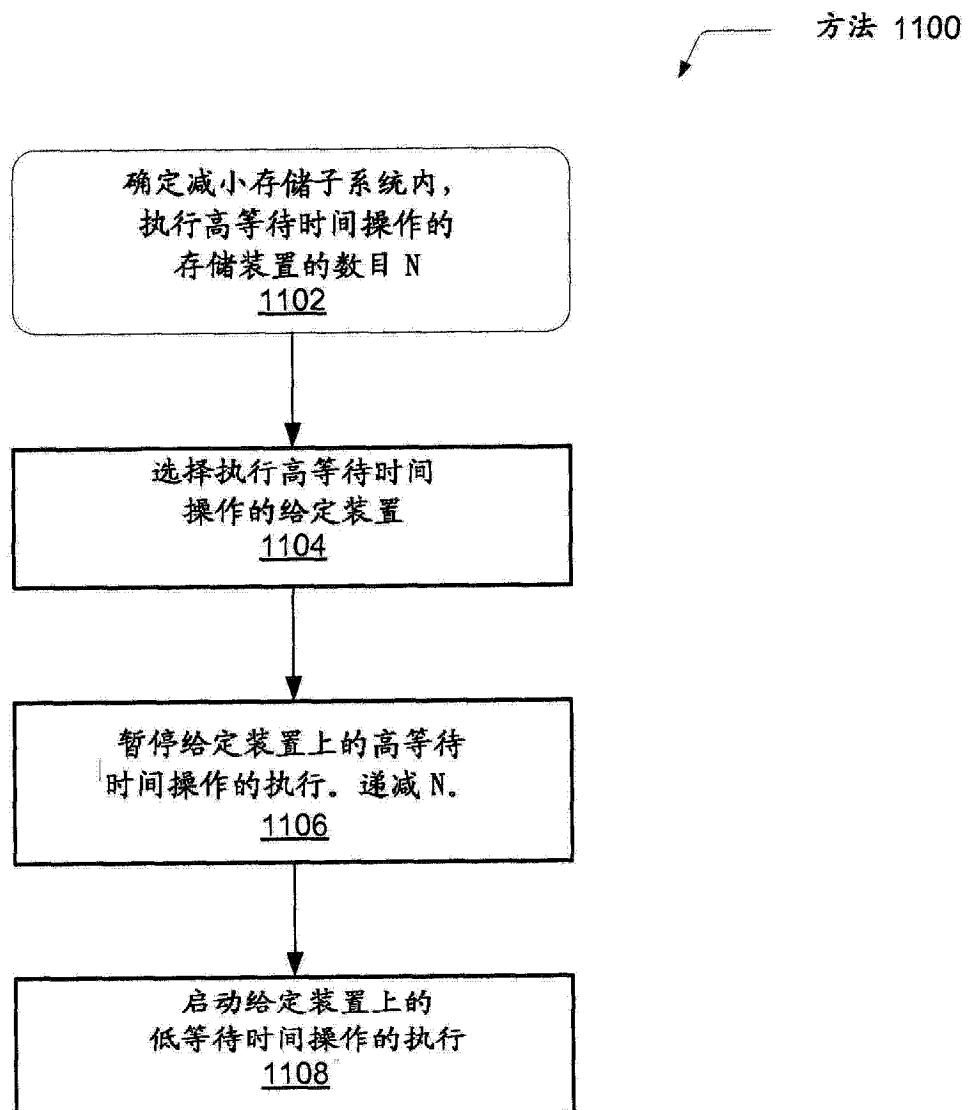


图 11

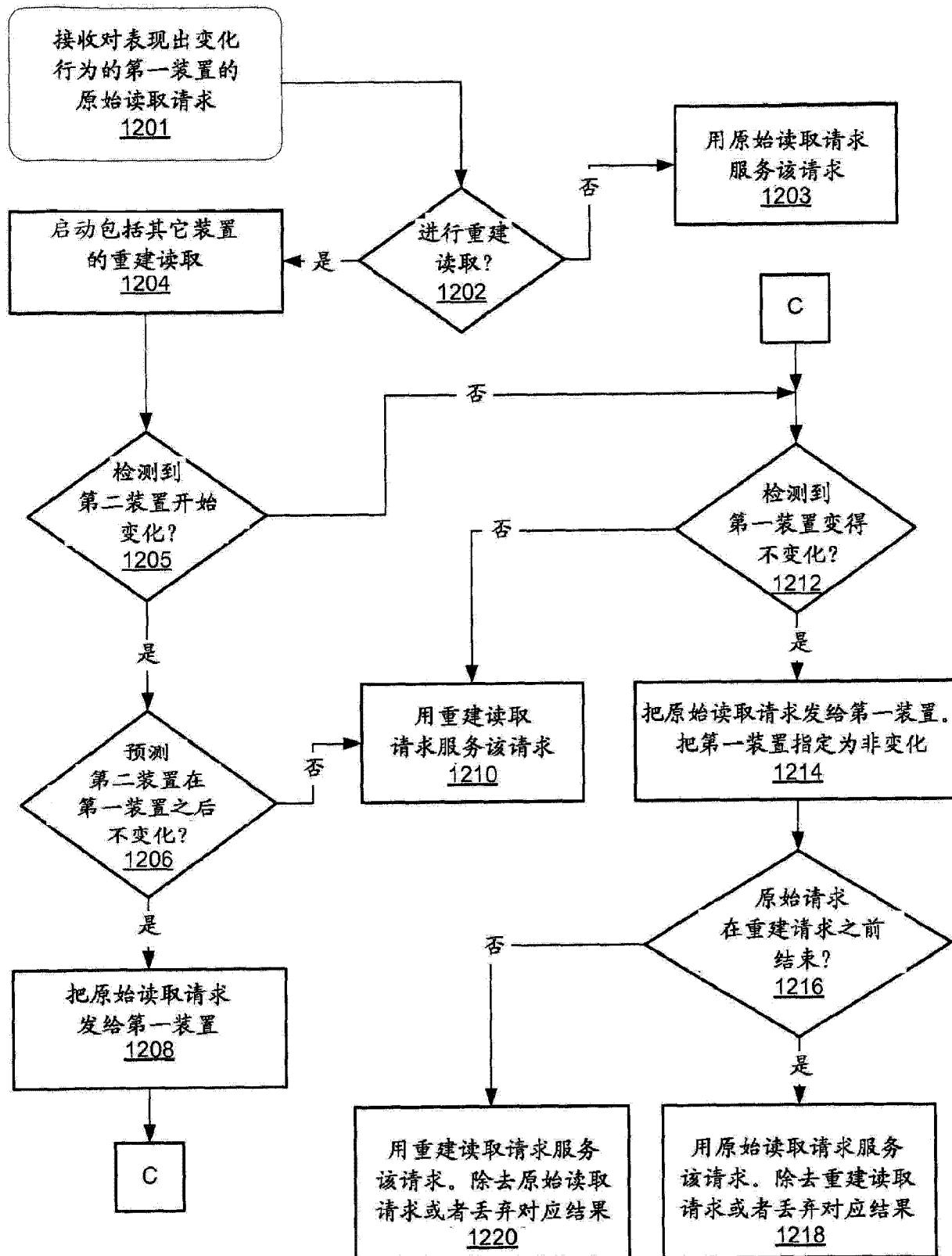


图 12