



(12) 发明专利

(10) 授权公告号 CN 102667720 B

(45) 授权公告日 2016. 06. 01

(21) 申请号 201080056154. 2

(22) 申请日 2010. 11. 11

(30) 优先权数据

12/635725 2009. 12. 11 US

(85) PCT国际申请进入国家阶段日

2012. 06. 11

(86) PCT国际申请的申请数据

PCT/US2010/056311 2010. 11. 11

(87) PCT国际申请的公布数据

W02011/071656 EN 2011. 06. 16

(73) 专利权人 微软技术许可有限责任公司

地址 美国华盛顿州

(72) 发明人 T. J. 米勒 J. M. 卡吉尔

W. R. 蒂普顿 S. 费尔马

(74) 专利代理机构 中国专利代理(香港)有限公

司 72001

代理人 谢建云 刘鹏

(51) Int. Cl.

G06F 9/44(2006. 01)

G06F 3/06(2006. 01)

(56) 对比文件

US 2005/0204108 A1, 2005. 09. 15,

CN 1517885 A, 2004. 08. 04,

CN 1776595 A, 2006. 05. 24,

审查员 谢萍

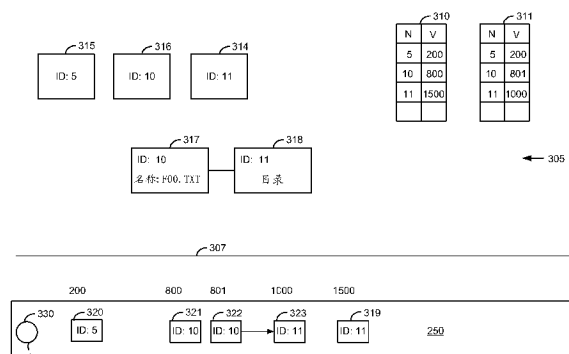
权利要求书3页 说明书12页 附图4页

(54) 发明名称

没有排序依赖的一致性

(57) 摘要

本文中所描述的主题的各方面涉及维护存储系统中的一致性。在各方面中,可以在事务的情境中更新一个或多个对象。结合更新这些对象,可以获取并修改对象的逻辑副本。写所更新的逻辑副本的请求被发送到存储控制器。该逻辑副本没有重写原始副本。结合发送该请求,提供用于存储控制器在磁盘上存储的数据结构。该数据结构指示应被写到磁盘的一个或多个对象,并且可以包括验证数据以指示应被写到磁盘的内容。在恢复期间,该数据结构可用来确定是否所有一个或多个对象都被正确地写到磁盘。



1. 一种至少部分由计算机实现的方法,该方法包括:
获取参与事务的对象的一个或多个指示;
为参与所述事务的对象定位存储设备的第一存储位置集合;
创建在所述事务的情境中被修改的对象的逻辑副本;
为所述逻辑副本确定所述存储设备的第二存储位置集合;
创建将所述对象的逻辑副本绑定在原子单元中的关联;
发送将所述关联和所述逻辑副本写到所述存储设备的一个或多个请求,同时在所述存储设备上维护所述对象的正本;以及
在出现故障后,执行恢复操作,该恢复操作包括:
确定在所述事务的情境中被修改的所有对象的逻辑副本是否在所述存储设备中,以及
当确定与在所述事务的情境中被修改的对象之一相对应的对象的逻辑副本不在所述存储设备中时,丢弃在所述存储设备中在所述事务的情境中被修改的所有对象的逻辑副本的修改。
2. 如权利要求 1 所述的方法,其中所述第二存储位置集合中的每个存储位置不同于所述第一存储位置集合中的每个存储位置,并且其中创建所述对象的逻辑副本包括创建在所述事务期间修改的对象的部分的逻辑副本。
3. 如权利要求 1 所述的方法,其中创建将所述对象的逻辑副本绑定在原子单元中的关联包括创建成对的逻辑副本之间的关联,每个链接包括指示所述成对的逻辑副本中的至少一个的正确内容的散列值。
4. 如权利要求 1 所述的方法,其中发送写所述关联和所述逻辑副本的一个或多个请求包括发送一个或多个写指示所述逻辑副本的每个存储位置的数据结构的请求,一个或多个将其中之一除外的所有逻辑副本写到所述存储设备的请求,和等待发送写所述逻辑副本之一的请求直到出现预定事件。
5. 如权利要求 4 所述的方法,其中等待发送写所述逻辑副本之一的请求直到出现预定事件包括等待以下中的一个或多个:直到另一事务已完成,直到已经过预定时间段和直到到达检查点。
6. 如权利要求 1 所述的方法,其中,所述恢复操作进一步包括:
获取指示对象在非易失性存储设备中的最后已知正确的存储位置的第一数据结构;
获取指示其中计划存储所述逻辑副本的所述非易失性存储设备的存储位置的另一数据结构,所述另一数据结构还指示表明所述逻辑副本的正确内容的第一散列集合;
读取位于所述存储位置的内容;
计算所述内容的第二散列集合;
如果所述第一和第二散列集合的对应成员是相等的,则更新所述第一数据结构以指示所述逻辑副本存储在所述存储位置处;以及
如果所述第一和第二散列集合的任意所述对应成员不等,则结束所述恢复操作。
7. 如权利要求 1 所述的方法,进一步包括指示与所述对象的正本相关联的存储位置可供使用,并指示与所述逻辑副本相关联的存储位置在成功地将所有逻辑副本冲刷到所述存储设备的非易失性存储器之后有效。
8. 一种至少部分由计算机实现的方法,该方法包括:

获取参与一个或多个事务更新的对象的指示；
在存储设备的第一存储位置中定位所述对象；
创建所述对象的逻辑副本，其包括在所述一个或多个事务的情境中对所述对象做出的改变；

确定用于存储所述逻辑副本的所述存储设备的第二存储位置；
发送将改变的所述逻辑副本写到所述第二存储位置的请求；
结合发送所述请求，提供指示下一位置的数据，属于另一事务的逻辑副本，如果有的话，将被写在该下一位置处；

更新易失性数据结构以指示所述逻辑副本被存储在所述第二存储位置中；以及
发送将易失性数据结构的改变写到所述存储设备的请求，并结合发送将易失性数据结构的改变写到所述存储设备的该请求，向所述存储设备发送冲刷命令，所述冲刷命令指示所述存储设备的控制器将与所述存储设备相关联的缓存写到所述存储设备的非易失性存储器。

9. 如权利要求 8 所述的方法，其中在存储设备的第一存储位置中定位所述对象包括接收所述对象的标识符，并使用所述标识符来在将标识符与存储位置相关联的数据结构中查找关联。

10. 如权利要求 8 所述的方法，进一步包括执行恢复动作，包括：
定位链表的头，所述链表的每个成员与将被写到由所述链表指定的所述存储设备的位置的逻辑副本相关联；

遍历所述链表直到链表的成员与没有被写到与所述成员相关联的位置的逻辑副本相关联；以及

结合遍历所述链表，对于每个成员更新数据结构以指示与所述逻辑副本相关联的对象的新位置，针对所述每个成员逻辑副本被写到由该成员指定的位置。

11. 在计算环境中，一种系统包括：
接口，可操作地接收修改参与事务的对象的请求；
存储控制器，其潜在地允许使写到存储设备的非易失性存储器以不同于接收针对所述写的请求的顺序的顺序进行，所述存储设备在其上具有所述非易失性存储器和易失性存储器；以及

I/O 管理器，可操作地执行动作，包括：
在所述存储设备的第一存储位置中定位所述对象，
创建所述对象的逻辑副本，其包括在所述事务的情境中对所述对象做出的改变，
确定用于存储所述逻辑副本的所述存储设备的第二存储位置，
发送将改变的所述逻辑副本写到所述第二存储位置的请求，以及
更新易失性数据结构以指示所述逻辑副本被存储在所述第二存储位置中；以及
恢复管理器，可操作地执行包括以下各项的方法：
确定所改变的所述对象的逻辑副本和所改变的与所述对象原子绑定的至少一个其它对象中的每一个的逻辑副本是否被写到所述第二存储位置，以及

当确定与所述对象的逻辑副本和所述至少一个其它对象中的每一个的逻辑副本中的至少一个相对应的改变未被写到所述第二存储位置，则丢弃与所述对象的逻辑副本和所述

至少一个其它对象的逻辑副本相对应的写到所述第二存储位置的改变。

12. 如权利要求 11 所述的系统,其中所述接口进一步可操作地接收修改参与所述事务的所述至少一个其它对象中的一个的请求,并且其中所述 I/O 管理器进一步可操作地执行另外的动作,包括:

创建将所述至少一个其它对象中的所述一个和所述对象绑定在一起的关联;以及

结合发送将改变的所述逻辑副本写到所述第二存储位置的请求,发送将所述关联写到所述存储控制器的请求。

13. 如权利要求 11 所述的系统,进一步包括检查点管理器,其可操作地发送将所述易失性数据结构写到所述存储控制器的请求,并将冲刷请求发送到所述存储控制器,所述冲刷请求指示所述存储控制器将数据从所述存储设备的易失性存储器冲刷到所述存储设备的非易失性存储器。

14. 如权利要求 11 所述的系统,其中,恢复管理器进一步可操作地基于比较所改变的所述对象的逻辑副本的存储的散列和根据从所述第二存储位置读取的内容计算的散列来确定所改变的所述对象的逻辑副本是否将被写到所述第二存储位置,如果所述散列相等,则所述恢复管理器进一步可操作地更新另一数据结构以指示所述对象的逻辑副本被存储在所述第二存储位置中。

没有排序依赖的一致性

背景技术

[0001] 在提高磁盘性能的努力中,磁盘制造商已经创造了具有缓存的磁盘。虽然以前在从磁盘读取数据的时候使用了这些缓存,但是最近在将数据写到磁盘时也已经使用它们。具体地,当操作系统请求将数据写到磁盘时,磁盘控制器可以在数据真正从磁盘缓存被传送到非易失性磁盘存储器之前报告该数据已被写入。同时,数据被写到非易失性磁盘存储器的顺序可以不同于通过磁盘控制器接收数据的顺序。这些行为对于尝试维护一致性的系统是有问题的。

[0002] 本文中要求保护的主体不限于解决任何缺点或者仅在诸如上述环境中操作的实施例。相反,提供此背景仅以示出其中可以实践本文中描述的一些实施例的一个示例性技术区域。

发明内容

[0003] 简言之,本文中描述的主体的各方面涉及维持存储系统中的一致性。在各方面中,可以在事务(transaction)的情境(context)中更新一个或多个对象。结合更新这些对象,可以获取并修改对象的逻辑副本。写所更新的逻辑副本的请求被发送到存储控制器。该逻辑副本没有重写原始副本。结合发送该请求,提供用于存储控制器将其存储在磁盘上的数据结构。该数据结构指示应被写到磁盘的一个或多个对象,并且可以包括验证数据以指示应被写到磁盘的内容。在恢复期间,该数据结构可用来确定是否所有一个或多个对象都被正确地写到磁盘。

[0004] 提供此发明内容来以简要形式标识主题的一些方面,主题将在以下具体实施例中进一步描述。此发明内容不旨在标识要求保护的主体关键特征或必要特征,也不旨在用来限制要求保护的主体范围。

[0005] 除非上下文另有明确指出,否则短语“在此所述的主体”指代在具体实施例中描述的主题。术语“各方面”将被解读为“至少一个方面”。标识具体实施例中描述的主题的各方面不旨在标识要求保护的主体关键特征或必要特征。

[0006] 上述各方面和本文中描述的主体的其他方面以示例的方式示出并且不限于附图,在附图中,相同的附图标记指示相似的元件并且其中。

附图说明

[0007] 图 1 是表示本文中所描述的主题的各方面可以包含其中的示例性通用计算环境的框图;

[0008] 图 2 是表示本文中所描述的主题的各方面可以在其中操作的系统组件的示例性布置;

[0009] 图 3 是示出本文中描述的主题的各方面的框图;

[0010] 图 4 是根据本文中所述的主体各方面的一般地表示当单个对象在事务的情境中修改时可能出现的示例性动作的流程图;以及

[0011] 图 5 是根据本文中所述的主题的各方面的一般地表示当多个对象在事务的情境中修改时可能出现的示例性动作的流程图。

具体实施例

[0012] 定义

[0013] 本文中使用的术语“包括”及其变形将被解读为开放式术语,其意指“包括,但不限于”。除非上下文另有明确指出,否则术语“或者”将被解读为“和 / 或”。术语“基于”将被解读为“至少部分地基于”。术语“一个实施例”和“实施例”将被解读为“至少一个实施例”。术语“另一实施例”将被解读为“至少一个其他实施例”。其他显式和隐式的定义可以在下文包括。

[0014] 示例性操作环境

[0015] 图 1 示出本文中描述的主题可以在其中实现的合适的计算系统环境 100 的示例。计算系统环境 100 仅是合适的计算环境的一个实例,并不旨在表明对于本文中描述的主题各方面的功能或使用的范围的任何限制。计算环境 100 不应被解释为具有涉及示例运行环境 100 中示出的任意一个组件或组合的任何依赖性要求。

[0016] 本文中描述的主题的各方面可用于众多其他通用或专用计算系统环境或配置。适合于与本文中描述的主题的各方面一起使用的众所周知的计算系统、环境或配置的实例包括个人计算机、服务器计算机、手持或膝上设备、多处理器系统、基于微控制器的系统、机顶盒、可编程消费电子设备、网络 PC、小型计算机、大型计算机、个人数字助理(PDA)、游戏设备、打印机、包括机顶、媒体中心的器具或其他器具、嵌入汽车或连接汽车的计算设备、其他移动设备、包括以上任意系统或设备的分布式计算环境等。

[0017] 本文中描述的主题的各方面可以在由计算机可执行的诸如程序模块的计算机可执行指令的一般背景中描述。通常,程序模块包括例程、程序、对象、组件、数据结构等,其执行特定的任务或实现特定抽象数据类型。本文中描述的主题的各方面还可以在分布式计算环境中实践,其中通过经由通信网络链接的远程处理设备来执行任务。在分布式计算环境中,程序模块可以位于包括存储器存储设备的本地和远程计算机存储介质两者中。

[0018] 参考图 1,用于实现本文中描述的主题的各方面的示例性系统包括以计算机 110 为形式的通用计算设备。计算机可以包括能够执行指令的任意电子设备。计算机 100 的组件可以包括处理单元 120、系统存储器 130 和将包括系统存储器的各种系统组件耦接到处理单元 120 的系统总线 121。系统总线 121 可以是若干类型的任意的总线结构,包括存储器总线或存储器控制器、外围总线和使用多种总线架构的任意的局部总线。以实例而非限制的方式,这样的架构包括工业标准架构(ISA)总线、微通道架构(MCA)总线、增强 ISA(EISA)总线、视频电子标准协会(VESA)局部总线、外围组件互连(PCI)总线(也称为夹层总线(Mezzanine bus))、扩展外围组件互连(PCI-X)总线、高级图形端口(AGP)和 PCI express(PCIe)。

[0019] 计算机 100 典型地包括多种计算机可读介质。计算机可读介质可以是能够通过计算机 110 访问的任意可获得的介质,并且包括易失性和非易失性介质两者,以及可移动和不可移动介质。以示例而非限制的方式,计算机可读介质可以包括计算机存储介质和通信介质。

[0020] 计算机存储介质包括以任意方法和技术实现的易失性和非易失性、可移动和不可移动介质以用于存储诸如计算机可读指令、数据结构、程序模块或其他数据的信息。计算机存储介质包括 RAM、ROM、EEPROM、闪存存储器或其他存储器技术、CD-ROM、数字多功能盘 (DVD) 或其他光盘存储装置、磁盒、磁带、磁盘存储装置或其他磁存储设备或者可以用来存储期望的信息并可以通过计算机 110 访问的任意其他介质。

[0021] 通信介质典型地体现计算机可读指令、数据结构、程序模块、或诸如载波或者其他传输机构的调制的数据信号中的其他数据,并包括任意信息递送介质。术语“调制的数据信号”意指使其特性的一个或多个以便于在信号中编码信息的方式来设置或改变的信号。以示例而非限制的方式,通信介质包括诸如有线网络或直接有线连接的有线介质,和诸如声学、RF、红外的无线介质以及其他无线介质。以上任意组合也应当被包括在计算机可读介质的范围内。

[0022] 系统存储器 130 包括易失性和 / 或非易失性存储器形式的计算机存储介质,例如只读存储器 (ROM) 131 和随机存取存储器 (RAM) 132。基本输入 / 输出系统 133 (BIOS) 典型地存储在 ROM 131 中,该基本输入 / 输出系统 133 包含有助于例如在启动期间在计算机 110 内的元件之间传送信息的基本例程。RAM 132 典型地包含可立即由处理单元 120 访问和 / 或目前正在被处理单元 120 操作的数据和 / 或程序模块。以示例而非限制的方式,图 1 示出操作系统 134、应用程序 135、其他程序模块 136 和程序数据 137。

[0023] 计算机 110 还可以包括其他可移动 / 不可移动、易失性 / 非易失性计算机存储介质。以示例的方式,图 1 示出读取或写到不可移动、非易失性磁介质的硬盘驱动器 141、读取或写到可移动、非易失性磁盘 152 的磁盘驱动器 151、读取或写到可移动、非易失性光盘 156 (例如 CD ROM 或其他光学介质) 的光盘驱动器 155。可以用在示例性操作环境中的其他可移动 / 不可移动、易失性 / 非易失性计算机存储介质包括磁带盒、闪存卡、数字多功能盘、其他光盘、数字视频带、固态驱动器、固态 RAM、固态 ROM、包括两个或更多个存储介质的混合设备、逻辑上被视为单个设备的存储设备集合 (其中每个设备可以包括相似和 / 或不同的特性) 等等。硬盘驱动器 141 典型地通过诸如接口 140 的不可移动存储器接口连接到系统总线 121,并且磁盘驱动器 151 和光盘驱动器 155 典型地通过诸如接口 150 的可移动存储器接口连接到系统总线 121。

[0024] 除了寻址本地存储装置的接口,接口 140 可以包括基于存储区域网络 (SAN) 的接口、基于网络寻址存储 (NAS) 的接口、包括 SAN 和 NAS 的混合接口等。SAN 和 / 或 NAS 可以使用光纤通道、SCSI、iSCSI、PCI-X、以太网、USB 或与存储设备进行通信的一些其他互联技术。

[0025] 以上讨论并在图 1 中示出的驱动器及其关联的计算机存储介质为计算机 110 提供计算机可读指令、数据结构、程序模块和其他数据的存储。在图 1 中,例如,硬盘驱动器 141 被示出为存储操作系统 144、应用程序 145、其他程序模块 146 和程序数据 147。请注意,这些组件可以与操作系统 134、应用程序 135、其他程序模块 136 和程序数据 137 相同或者不同。本文中对操作系统 144、应用程序 145、其他程序模块 146 和程序数据 147 给予不同的编号以说明它们至少是不同的副本。

[0026] 用户可以通过输入设备 (例如键盘 162 和指点设备 161,通常称为鼠标、轨迹球或触摸板) 将命令和信息输入到计算机 110 中。其他输入设备 (未示出) 可以包括麦克风、操

纵杆、游戏垫、碟形卫星天线、扫描仪、触敏屏幕、书写板等。这些或其他输入设备通常通过耦接到系统总线的用户输入接口 160 连接到处理单元 120,但是可以通过诸如并口、游戏端口或通用串行总线(USB)的其他接口和总线结构连接。

[0027] 监视器 191 或其他类型的显示设备还经由诸如视频接口 190 的接口连接到系统总线 121。除了监视器,计算机还可以包括其他外围输出设备,例如扬声器 197 和打印机 196,其可以通过输出外围端口 195 来连接。

[0028] 计算机 110 可以在使用到一个或多个远程计算机(例如远程计算机 180)的逻辑连接的网络环境中运行。虽然在图 1 中仅示出存储器存储设备 181,但是远程计算机 180 可以是个人计算机、服务器、路由器、网络 PC、对等设备或其他公共网络节点,并且典型地包括许多或者所有与计算机 110 有关的上述元件。图 1 中描绘的逻辑连接包括局域网(LAN) 171 和广域网(WAN) 173,但是还可以包括其他网络。这样的联网环境在办公室、企业范围的计算机网络、内联网和因特网中是司空见惯的。

[0029] 当使用在 LAN 联网环境中时,计算机 110 通过网络接口或适配器 170 连接到 LAN 171。当使用在 WAN 联网环境中时,计算机 110 可以包括调制解调器 172 或用于在诸如因特网的 WAN 173 上建立通信的其他装置。内部或外部的调制解调器 172 可以经由用户输入接口 160 或其他适当的机构连接到系统总线 121。在网络的环境中,所描绘的与计算机 110 有关的程序模块或其部分可以存储在远程存储器存储设备中。以示例而非限制的方式,图 1 将远程应用程序 185 示出为驻留在存储器设备 181 上。将会明了,所示出的网络连接是示例性的,并且可以使用其他手段来建立计算机之间的通信链路。

[0030] 一致性

[0031] 如前所述,缓存和乱序写入磁盘对于试图维持一致性的系统来说是问题。本文中有时使用了术语事务。事务是这样的一组操作,其可以包括各种属性,例如包括原子的(atomic)、一致的、隔离的和持久的。本文中使用的术语至少包括原子属性,并可以包括以上其他属性的一个或多个。

[0032] 原子属性用来指代这样的一组操作,其中或者该组中的每个操作都成功,或者该组中这些操作的实质性效果(例如,文件改变)被撤销、丢弃或者未被应用。为简单起见,本文中有时使用术语丢弃(discarded)来指代采取任何适当的动作来确保在该事务的情境中做出的任何改变没有反映在与这些改变相关联的对象中。丢弃可以包括撤销、废弃、不应用更新操作等。

[0033] 例如,银行转账可以实现为两个操作的原子集合:从一个账户借记和存入另一账户。如果两个操作没有都成功,则该转账不公平地对银行或者开户人有利。因而,或者事务中的两个操作都成功,或者任何确实成功的操作的实质性效果(例如,存储到磁盘或存储器的数据)被丢弃。

[0034] 当在此使用“事务”时,它可以但未必意指参与该事务的组件理解事务正在发生。例如,组件可以被明确地通知两个或更多个对象参与事务。该组件随后可以将原子属性强加在如上所述的针对对象的操作上。作为另一实例,组件可以不必被通知事务正在发生。而是,组件可以确定或者被指示给予它的两个或更多个操作或者被成功地完成或者操作的效果将被丢弃。

[0035] 在以上第一实例中,组件被给予对象参与事务的明确信息。在第二实例中,组件可

能没有被通知事务影响两个或更多个操作。相反,组件可能已被指示将原子属性强加在操作上。本文中描述的主题的各方面适用于以上两个实例。

[0036] 此外,当一个或多个对象“在事务的情境中”被修改时,这意味着存在这样的假设,即将对于被发出用来修改一个或多个对象的更新操作强加原子属性。例如,在事务的情境中请求修改的应用可以安全地假设做出修改的所有更新操作将成功,或者完成的更新或本应成功的更新将被丢弃。

[0037] 图 2 是表示其中可以操作本文中所述的主题各方面的系统的组件的示例性布置的框图。图 2 中示出的组件是示例性的而且并非意指包括所有可能需要或可能被包括的组件。在其他实施例中,在没有脱离本文中描述的主题的各方面的精神或范围的情况下,结合图 2 描述的组件和 / 或功能可以被包括在(示出或未示出的)其他组件中,或者被放置在子组件中。在一些实施例中,结合图 2 描述的组件和 / 或功能可以跨多个设备分布。

[0038] 转向图 2,系统 205 可以包括一个或多个应用 210、API 215、一致性组件 220、存储设备(store) 250、通信机构 255 和其他组件(未示出)。系统 205 可以包括一个或多个计算设备。这样的设备可以例如包括个人计算机、服务器计算机、手持或膝上型计算机、多处理器系统、基于微控制器的系统、机顶盒、可编程消费电子设备、网络 PC、小型计算机、大型计算机、手机、个人数字助理(PDA)、游戏设备、打印机、包括机顶、媒体中心的器具或其他器具、嵌入汽车或连接汽车的计算设备、其他移动设备、包括任意以上系统和设备的分布式计算环境、等等。

[0039] 在系统 205 包括单个设备的场合,可以被配置为充当系统 205 的示例性设备包括图 1 的计算机 110。在系统 205 包括多个设备的场合,所述多个设备的每一个可以包括图 1 的相似地或不同地配置的计算机 110。

[0040] 一致性组件 220 可以包括恢复(recovery)管理器 225、检查点管理器 230、I/O 管理器 235 和其他组件(未示出)。本文中使用的术语组件将被解读为包括设备的全部或部分、一个或多个软件模块选集或其部分、一个或多个软件模块的某组合或其部分以及一个或多个设备或其部分等等。

[0041] 通信机构 255 允许系统 205 与其他实体进行通信。例如,通信机构 255 可以允许系统 205 与远程主机上的应用进行通信。通信机构 255 可以是网络接口或者适配器 170、调制解调器 172、或者用于建立结合图 1 所述的通信的任意其他机构。

[0042] 存储设备 250 是能够提供数据访问的任意存储介质。该存储设备可以包括易失性存储器(例如,缓存)和非易失性存储器(例如,永久存储设备)。术语数据将被宽泛地解读为包括可以由一个或多个计算机存储元件表示的任意事物。逻辑上,数据可以表示为易失性和非易失性存储器中的一系列的 1 和 0。在具有非二进制存储介质的计算机中,数据可以根据存储介质的能力来表示。数据可以被组织成不同类型的包括简单数据类型,例如数字、字母等、分级的、链接的、或其他相关的数据类型的数据结构,包括多个其他数据结构或简单数据类型的数据结构,等等。数据的一些实例包括信息、程序代码、程序状态、程序数据、其他数据等。

[0043] 存储设备 250 可以包括硬盘存储装置、其他非易失性存储装置、诸如 RAM 的易失性存储装置、其他存储装置、以上的某组合等,并且可以跨多个设备分布。存储设备 250 可以是外部的、内部的、或者包括在系统 205 的内部和外部两者的组件。

[0044] 存储器 250 可以经由存储控制器 240 访问。本文中使用的访问可以包括读数据、写数据、删除数据、更新数据、包括以上两个或更多个的组合等。存储控制器 240 可以接收访问存储设备 250 的请求,并可以适当地满足这样的请求。存储控制器 240 可以被布置为使得其不保证数据将按其被接收的顺序被写到存储设备 250。此外,存储控制器 240 可以在该存储控制器 240 真正将数据写到存储设备 250 的非易失性存储器之前指示其已经写所请求的数据。

[0045] 一个或多个应用 210 包括可能参与创建、删除或更新资源的事物的任意进程。这样的进程可以以用户模式或内核模式执行。本文中使用的术语“进程”及其变形可以包括执行任务的一个或多个传统的进程、线程、组件、库、对象等。进程可以以硬件、软件或者硬件和软件的组合来实现。在实施例中,进程是任意机构,无论如何调用,都能够执行动作或用于执行动作。进程可以分布在多个设备或单个设备上。该一个或多个应用 210 可以通过 I/O 管理器 235 的 API 215 来(例如经由功能 / 方法调用)做出文件系统请求。

[0046] I/O 管理器 235 可以确定什么 I/O 请求或多个 I/O 请求发出给存储控制器 240(或者一些其他中间组件)。I/O 管理器 235 还可以随着与文件系统请求关联的操作进行、完成或失败而将数据返回到该一个或多个应用 210。当文件系统请求涉及事务时,I/O 管理器 235 可以通知事务管理器(未示出)以使得事务管理器可以恰当地管理事务。在一些实施例中,事务管理器的功能可以包括在 I/O 管理器 235 中。

[0047] 以下,提出了一种用于以促进一致性和故障后恢复的方式将数据写到存储设备 250 的示例性算法。如该算法中提出的,每个对象用 D_n 表示,其中 n 标识系统的对象。假设对象是可序列化的(即,能够表示为存储设备 250 上的数据)。对象表将每个对象标识符与其在存储设备 250 上的位置相关联。

[0048] 当 D_n 首次在修改事务中更新,通过使用 n 在对象表中查找 D_n 的位置来定位 D_n 。为便于在此实例中使用,存储设备 250 上的存储位置 D_n 被称为 L_1 。

[0049] 随后从存储设备 250 中读取 L_1 的内容,对象可以被解序列化(例如,从序列化格式转换成对象的结构),并且将被修改的对象的部分复制到主系统存储器。对存储器中的各部分(及其副本)执行更新。结合存储器中被修改的各部分,为被修改的部分指定存储设备 25 上的一个或多个新位置(将之称为 L_2)。

[0050] 主系统存储器中的这些副本在本文中有时称为对象的“逻辑副本”。对象的逻辑副本包括可以用来表示对象的一个或多个数据结构。逻辑上,逻辑副本是对象的复制品。物理上,逻辑副本可以包括可用来创建对象的复制品的数据(包括指向其他数据的指针)。例如,在一个实施方式中,逻辑副本可以是对象的真实副本(例如,逐位复制)或者包括可用来创建对象的数据的数据结构。在另一实施方式中,未修改的逻辑副本可以包括指代原始对象的一个或多个指针。随着逻辑副本被修改,逻辑副本中的指针可以指代新的存储器位置(例如,针对逻辑副本的改变部分)同时其他指针可以指代原始对象的部分(例如,针对逻辑副本的未改变部分)。利用指针,可以使用修改的数据和原始对象的未修改的数据一起来构造修改的副本。可以执行创建逻辑副本来降低创建对象的复制品所需的存储量。

[0051] 此外,虽然在本文中有时提及序列化和解序列化,但是并非意在将本文中所述的主题的各方面限制于通常所认为的序列化和解序列化。在一个实施例中,序列化的版本与解序列化的版本可以是逐位相等的。在另一实施例中,序列化的版本的位可以以与解序列

化的版本不同的格式和顺序来打包。事实上,在一个实施例中,序列化和解序列化将被理解为意指用于存储和从存储设备取回表示对象的数据的任意机构。其他机构例如可以包括以文本格式将对象的属性写到存储设备,在存储设备中以标记语言来编码对象的属性、将对象的存储属性和其他特征存储在存储设备上的其他方式等。

[0052] 在系统裁量(discretion)时(例如,在事务约定(commit)或某其他时间之后),系统可以将修改的逻辑副本序列化回到稳定的介质,但是在位置 L_2 处进行该序列化。将修改的逻辑副本写回到新位置的意图称为写计划。写计划可以包括任意数目的对一个或多个对象的更新。写计划可以参考在多于一个事物中出现的改变。多个写计划可以组合成单个写计划。

[0053] 当恰在检查点之后出现修改时,称为恢复块的块(其可以在多个位置被复制)被修改以指向经修改的逻辑副本的开始(即 L_2)。在 L_2 处的对象中的字段指向接下来将被写入的位置。该字段表示写计划链中的链路。

[0054] 结合发送写逻辑副本的请求,可以对对象表进行修改。特别地,由对象的标识符索引的位置值可以被设置为将要存储修改的逻辑副本的位置的值(即, L_2)。这样做以使得对象 D_n 的位置后续查找将引用位置 L_2 , 对象的新版本。

[0055] 如果事务修改多于一个对象,例如 D_i 和 D_j , 则这些对象将被视为彼此“原子绑定”, 并且在一个写计划中写入。写计划可以指定此关系(例如,链接到所涉及的对象)。

[0056] 任意数目的对象可以以此方式被持久化。周期地,对象表也可以以与任意其他对象相同的方式写到存储设备 250。

[0057] 结合发送将对象表写到存储设备 250 的请求,还可以将冲刷命令(flush command)发送到存储控制器 240。冲刷命令指示存储控制器 240 将来自其易失性存储器的还没有被写的所有数据写到存储设备 250 的非易失性存储器。

[0058] 检查点可以周期地出现。检查点可以通过由存储设备 250 存储的检查点记录来指示。检查点可以在任意时间被写入并且可以在冲刷后变得稳定和持久。稳定/持久是指检查点存储在存储设备的非易失性存储器上。

[0059] 在检查点稳定/持久之后,可以重用用于任何旧的和未使用的对象副本(或其部分)的空间。在冲刷完成后,然后将恢复块指向下一个写计划链的开始。在一个实施例中,恢复块可以将写计划链的开始指向对象表的新位置。

[0060] 结合图 3 描述更具体的实例,其为示出本文中所述主题的各方面的框图。如图所示,图 3 示出主存储器 305 和存储设备 250。线 307 表示主存储器 305 和存储设备 250 之间的分界线。线 310 之上的对象在主存储器中,而线 310 之下的对象在存储设备 250 的易失性或非易失性存储器中。

[0061] 主存储器 305 中示出对象 314-316。在实施中,对象 314-316 可以分别是对对象 319-321 的解序列化的逻辑副本。对象 319 位于存储设备 250 上的位置 1550 处,对象 320 位于存储设备 250 上的位置 200 处,并且对象 321 位于存储设备 250 上的位置 800 处。

[0062] 对象表 310 包括指示存储设备 250 上的对象 314-316 的位置的键值对。使用对象 314-316 的一个或多个标识符来索引键值对。

[0063] 当事务修改对象 316 (例如通过将它的名字改为 foo.txt) 时,一致性组件(例如图 2 的一致性组件 220)可以确定更新的对象的新的存储位置(例如,位置 801)。如果对象

是文件,在事务的情境中更新它的名字还使得包括该文件的目录也参与该事务。例如,当文件名称被改变时,代表文件的对象和代表包括该文件的目录的对象两者可能需要参与该事务。在此情况下,包括该对象的目录被表示为对象 314,而更新的目录的逻辑副本(例如,对象 318)表示为存储设备 250 中的对象 323。同时,表 310 已被逻辑地更新到表 311 以指示修改的对象(即,对象 317 和 318)的新的存储位置(即,801 和 1000)。

[0064] 事务情境中的对象的修改也影响另一对象可以例如通过图 2 的 I/O 管理器 235 或某其他组件来明确地指示或确定。

[0065] 当两个或更多对象参与事务更新时,对象被视为如前所述的“原子绑定”。在恢复操作中,除非在存储设备 250 中发现该事务情境中改变的所有对象的改变,否则丢弃所有发现的改变。换言之,如果发现对象之一的改变但是没有发现对象的另一个的改变,则丢弃该对象之一的改变。

[0066] 为了原子地绑定两个或更多对象,在一个实施例中,指针可以存储在存储设备 250 中,或者与存储设备 250 中的每个对象关联。指针可以指示参与事务的另一对象(或其部分)的存储位置。如果没有附加的对象参与该事务,则指针可以指向“死块”或者指示另一写计划的“头”对象的存储位置。该头对象可以包括写计划、写计划的修改的对象(或其部分)等。

[0067] 因为文件系统可能参与许多事务,出于性能的原因,期望的是等待请求针对已完成的事务的“最后”对象的写改变直到另一事务已完成。事务的最后对象可以与将指向另一写计划的“头”对象的指针相关联。然而,“头”对象的存储位置可能是未知的直到另一事务完成。因而,为了继续写计划链,写计划中的“最后”对象可以不被写入直到已知另一写计划的头对象的存储位置。

[0068] 如果磁盘断电或另外出现故障,在事务的“最后”对象被写到非易失性存储器之前,以上方法指示丢弃写计划中的改变。为了避免事务频率低的系统中的这样的结果,在一些实施例中,一致性组件可以等待直到以下中的较早者:1)另一事务完成;2)在将该“最后”对象连同它的指针写到下一个存储位置前经过的超时时段;或者 3)可以总是将“最后”对象连同它的指针写到下一个存储位置。在上面的 2)和 3)中,一致性组件可以确定“最后”对象将指向的存储位置。于是该存储位置可以具有指向另一写计划的“头”对象的指针。该“最后”对象指向的这个存储位置有时称为“死块”。以这样的方式,可以减少在事务期间做出的改变的丢失。

[0069] 除了指向下一个存储位置的指针,数据还可以存储在存储设备 250 中以指示所“指向”的对象的正确内容。例如,可以存储指示被指向的对象的正确内容的散列(hash)。

[0070] 在图 3 中呈现的实例中,与对象 322 相关联的指针可以指向与对象 323 相关联的存储位置。指针将两个对象绑定在一起。如果在恢复期间,没有发现对象的其中之一,或者它们没有正确的内容,则可以丢弃由被发现的对象表示的改变。

[0071] 由于存储设备 250 的性质,所以可能不保证哪个对象将首先被写到存储设备 250 的非易失性存储器。如果首先写入对象 322 而没有写入对象 323,则来自对象 322 的指针将指向可能具有伪造数据的存储位置。然而,通过计算存储位置处的数据的散列并将该散列与和对象 322 一起存储的散列进行比较,一致性组件可以检测对象 323 的无效数据。在此情况下,在恢复期间,一致性组件可以丢弃由对象 322 和 323 表示的变化。

[0072] 恢复块 330 指向第一存储位置(在此情况下为 801),在该处数据应存储在检查点之后。恢复块 330 还可以包括使用存储在第一存储位置处的对象的正确内容计算的散列,或者与该散列相关联。

[0073] 在恢复期间,首先,还原对象表的最后已知的好的版本。该最后已知的好的版本是已经成功地存储在存储设备 250 的非易失性存储器中的逻辑对象表。然后,检查从由恢复块 330 指向的一个开始的写计划。

[0074] 回顾一下,写计划是意在将参与事务的所有修改的逻辑副本写到存储设备 250。可以经由指示包含在写计划中的对象数目的数据连同到包含在该写计划中的对象的存储位置的链接,将写计划编码在存储设备上。在一个实施例中,指示对象数目的数据可以存储在写计划的“头”对象中,同时链接到下一个存储位置的数据可以与写到存储设备 250 的每个元素一起存储。在另一实施例中,包括对象的数目并链接到存储位置的数据结构可以存储在与写计划相关联的存储位置之一(例如,“头”存储位置)中,或者与这些存储位置相分离地存储。在一个实施例中,写计划可以仅包括下一存储位置。

[0075] 在一个实施例中,出现在检查点之后的写计划可以被写到非易失性存储器上的已知位置而无需具有写计划之间的“链路”。在该实施例中,在恢复期间,恢复管理器可以从已知位置读取每个写计划,并且如果由写计划指示的写入成功,则适当地更新对象表和其他存储数据结构。

[0076] 以上实例并非旨在包括所有或穷尽可用来指示写计划元素的数据结构类型。基于本文中的教导,本领域技术人员可以认识到可用来指示写计划元素的其他数据结构而未脱离本文中描述的主题的各方面的精神或范围。

[0077] 随着每个写计划被检查,确定写计划的对象是否被成功地写到存储设备 250。例如,这可以通过比较与写计划相关联的每个存储位置的内容的散列和与到该位置的链接相关联的散列来实现。如果这些散列对于写计划的全部对象是相等的,则更新对象表来反映对象的新位置。如果有任意散列是不等的,则恢复停止。

[0078] 在恢复结束之后,对象表具有通过成功的写计划更新的所有对象的根位置,其中当与该计划相关联的所有对象已被成功地写到磁盘的非易失性存储器中时,成功的写计划出现。

[0079] 返回至图 2, API 215 可以接收修改参与事务的对象的请求。作为响应, I/O 管理器 235 可以在存储器的存储位置(例如, L_1) 中定位对象,创建对象的逻辑副本,在事务的情境中对对象做出改变,确定用于存储改变的逻辑副本的第二存储位置(例如, L_2),发送写改变的逻辑副本的请求到存储控制器 240,并更新易失性数据结构(例如,对象表 310)来指示该逻辑副本存储在第二存储位置中。

[0080] 如果 API 215 接收修改参与事务的另一对象的请求,则 I/O 管理器 235 可以执行另外的动作,包括创建将另一对象和第一对象绑定在一起的关联(例如,写计划)。随后,结合发送将对象的修改写到存储的请求, I/O 管理器 235 还可以发送将该关联写到存储控制器 240 的请求。

[0081] 检查点管理器 230 可操作地发送将易失性数据结构(例如,对象表)写到存储控制器 240 的请求,并将冲刷请求发送到存储控制器 240。冲刷请求指示存储控制器 240 将来自存储设备 250 的易失性存储器的数据冲刷到存储设备 250 的非易失性存储器。

[0082] 恢复管理器 225 可以执行之前描述的恢复动作。在确定所指向的位置是否包括正确的内容中,基于比较改变的逻辑副本的存储的散列和根据从第二存储位置读取的内容计算的散列,恢复管理器 225 可以确定改变的逻辑副本是否被写到第二存储位置。如果散列相等,则恢复管理器可进一步操作地更新另一数据结构(例如,根据最后的已知的好的状态构造的对象表)以指示逻辑副本存储在第二存储位置中。存储设备上分配的 / 释放的数据结构还可以被更新为与现已知将是有效写入(例如,因为写计划成功)的位置一致。

[0083] 图 4-5 是根据本文中所述的主题的各方面的通常表示可能出现的动作的流程图。为了便于解释,将结合图 4-5 描述的方法描绘和描述为一系列的行为。将会理解并明了,本文中所述的各方面不受示出的行为和 / 或行为的顺序的限制。在一个实施例中,各行为以如下所述的顺序出现。然而,在其他实施例中,各行为可以并行地出现,以另一顺序出现、和 / 或利用未在本文中描述和呈现的其他行为来出现。此外,可以不需要所有示出的行为以实现根据本文中所述的主题的各方面的方法。另外,本领域技术人员将理解和明了,可替换地,该方法可以表示为经由状态图的一系列不相关的状态或者表示为事件。

[0084] 图 4 是根据本文中所述的各方面的一般地表示当单个对象在事务的情境中被修改时可能出现的示例性动作。在框 405 处,动作开始。例如,参考图 2-3,API 215 可以接收更新对象 316 中的数据的请求。

[0085] 在框 410 处,接收参与事务的更新的对象的指示。例如,参考图 2-3,I/O 管理器 235 可以接收对象 316 的标识符 10。

[0086] 在框 415 处,对象位于存储设备的第一存储位置中。例如,参考图 2-3,I/O 管理器 235 可以查阅对象表 310 以确定对象 321 的位置 800。对象 321 例如可以对应于对象 316 的序列化表示。

[0087] 在框 420 处,创建对象的逻辑副本,其包括在事务情境中对对象所作出的改变。例如,参考图 3,创建逻辑副本(例如,由对象 317 表示),并利用在事务的情境中指定的更新来更新该逻辑副本。

[0088] 在框 425 处,确定用于存储更新的逻辑副本的新位置。例如,参考图 2-3,I/O 管理器 235 可以确定位置 801 将被用来存储更新的逻辑副本的序列化表示(例如,通过对象 317 表示)。

[0089] 在框 430 处,将逻辑副本写到新位置的请求发送到存储控制器。例如,参考图 2-3,I/O 管理器 235 将请求发送到存储控制器 240 以将对象 317 写到存储设备 250 的位置 801。

[0090] 在框 435 处,提供了到下一个位置的链接数据。如前所述,如果有的话,这可以包括将数据发送到指示由另一事务做出的改变的头部的存储位置的存储设备,或者发送指示可用来发现在下一个事务中做出的改变的“死块”的数据。可以在与写逻辑副本的请求相同的请求中或者在不同的请求中发送该链接数据。例如,参考图 2-3,I/O 管理器 235 可以发送链接数据 1000,其指示在另一事务情境中做出的数据更新可以起始于存储设备 250 的存储位置 1000 来存储。

[0091] 在框 440 处,更新对象表来反映对象的逻辑副本被请求存储在新的存储位置。例如,参考图 2-3,I/O 管理器 235 可以更新对象表 310 以使得具有 ID 10 的对象的值被更新至 801。

[0092] 在框 445 处,如果有的话,可以执行其他动作。例如,在冲刷命令之前的将改变写

到对象表的请求可以被发送到存储控制器。作为另一实例,可以执行恢复动作。如前所述,在一个实施方式中,这样的动作可以例如包括定位链表的头部、遍历链表直到发现与该链表的成员相关联的对象没有被写到存储设备、结合遍历链表,更新对象表以指示对象的新位置。

[0093] 图 5 是根据本文中描述的主题的各方面的一般地表示在多个对象在事务的情境中修改时可能出现的示例性动作的流程图。在框 505 中,动作开始。例如,参考图 2-3,API 215 可以接收更新文件名称的请求。更新文件名称例如可包括更新目录的数据结构。

[0094] 在框 510 中,获取参与事务的对象的一个或多个指示。例如,参考图 2-3,I/O 管理器 235 可以接收对象 316 的标识符 10 和改变与对象 316 相对应的文件的名称的请求。因为文件的名称被改变,所以 I/O 管理器 235 (或另一组件)可以确定目录(例如与对象 318 相对应的目录)也参与该事务。还可以获取目录(11)的 ID。

[0095] 在框 515 处,对象的位置位于存储设备中。例如,参考图 2-3,I/O 管理器 235 可以询问对象表 310 来分别确定对象 319 和 321 的位置 1500 和 800。对象 321 例如可以对应于对象 316 的序列化表示,而对象 319 可以对应于对象 314 的序列化表示。

[0096] 在框 520 处,创建在事务的情境中修改的对象的逻辑副本。例如,参考图 3,利用事务的情境中指定的更新来创建和更新逻辑副本(例如,由对象 317 和 318 表示)。

[0097] 在框 525 处,新位置被确定以存储改变的对象。例如,参考图 2-3,I/O 管理器 235 可以确定位置 801 将被用来存储文件的更新的逻辑副本的序列化表示(由对象 317 表示),而位置 1000 被用来存储目录的更新的逻辑副本的序列化表示(由对象 318 表示)。

[0098] 在框 530 处,创建将对象的逻辑副本绑定在原子单元中的关联。例如,可以创建链接逻辑副本的写计划。例如,参考图 2-3, I/O 管理器 235 可以创建链接对象 317 和 318 的写计划。作为另一示例,可以创建包括存储位置之间的引用连同散列值的链表。如前所述,这些散列值可以指示包括在写请求中的逻辑副本的正确内容。

[0099] 在框 535 中,发出将该关联和对象写到存储设备的请求。如前所述,该请求将制作新的副本并在存储设备上维护对象的正本。例如,参考图 2-3, I/O 管理器 235 可以发送将写计划连同对象 317-318 写到存储设备 250 的请求到存储控制器 240。

[0100] 作为另一实例,发送写入该关联和对象的请求可以包括发送一个或多个写入指示逻辑副本的每个存储位置的数据结构的请求,一个或多个将其中之一除外的所有逻辑副本写到存储设备的请求,和等待发送写入逻辑副本之一的请求,直到出现预定事件,例如,直到另一事务已完成,直到已经过预定时间段,和直到到达检查点。如果另一事务完成,可以执行另外的动作,包括发送写入“最后的”逻辑副本连同在另一事务情境中修改的对象的逻辑副本的存储位置的引用的请求。

[0101] 在框 540 处,更新对象表。例如,参考图 3,对象表 310 可以利用新的位置来更新以创建对象表 311。

[0102] 在框 545 处,如果有的话,可以执行其他动作。例如,可以执行恢复动作。这些恢复动作可以包括:

[0103] 1. 获取指示非易失性存储设备中的最后已知的正确的存储位置的第一数据结构;

[0104] 2. 获取指示其中计划存储逻辑副本的非易失性存储设备的存储位置的另一数据

结构。该数据结构还指示表明逻辑副本的正确内容的第一散列集合；

[0105] 3. 读取位于存储位置处的内容；

[0106] 4. 计算该内容的第二散列集合；

[0107] 5. 如果第一和第二散列集合的相应成员是相等的，则更新第一数据结构以指示逻辑副本被存储在存储位置处，并针对存储在存储设备上的更新的逻辑副本的下一个集合重复以上步骤 1-5，如果有的话；并且

[0108] 6. 如果第一和第二散列集合的任意相应成员是不等的，则结束恢复动作。

[0109] 另一可能出现的示例性动作包括指示在成功地将所有逻辑副本冲刷到存储设备的非易失性存储器之后，与对象的正本相关联的存储位置可供使用。

[0110] 这些其他示例性动作不旨在穷尽或包括所有可能出现的动作，并且这里仅给出说明性的可能出现的一些其他动作。

[0111] 根据前面的具体实施例可以看出，已经描述了有关在存储系统中维护一致性的各方面。但是本文中描述的主题的各方面容许各种修改和可替换的构造，其某些示出的实施例在附图中显示并且在上文中详细描述。然而，应当理解的是，这不意在将所要保护的主题的各方面限制于所公开的特定形式，而是相反意在覆盖所有的改变、可替换的构造和落入在本文中描述的主题的各方面的精神和范围的等价物。

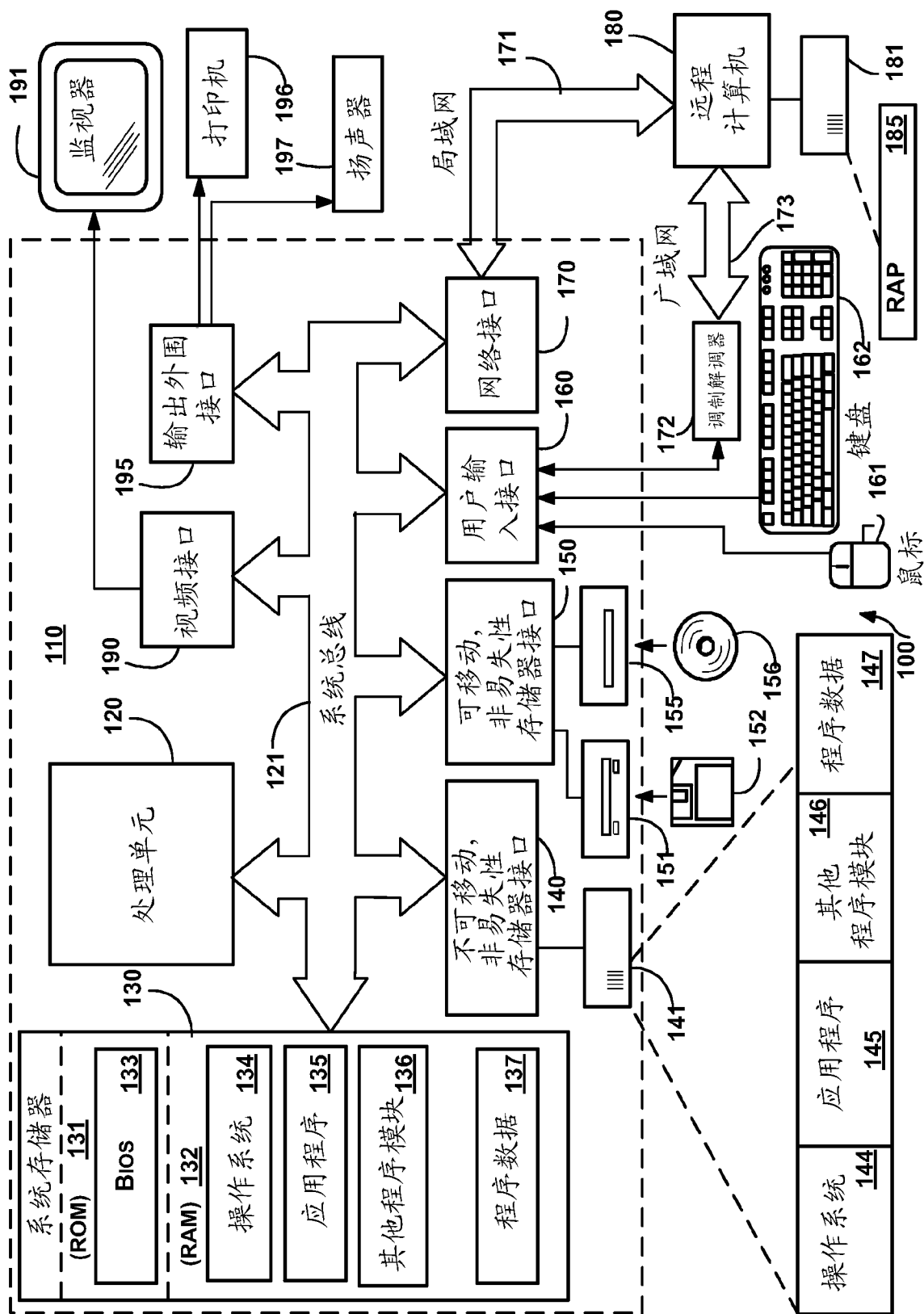


图 1

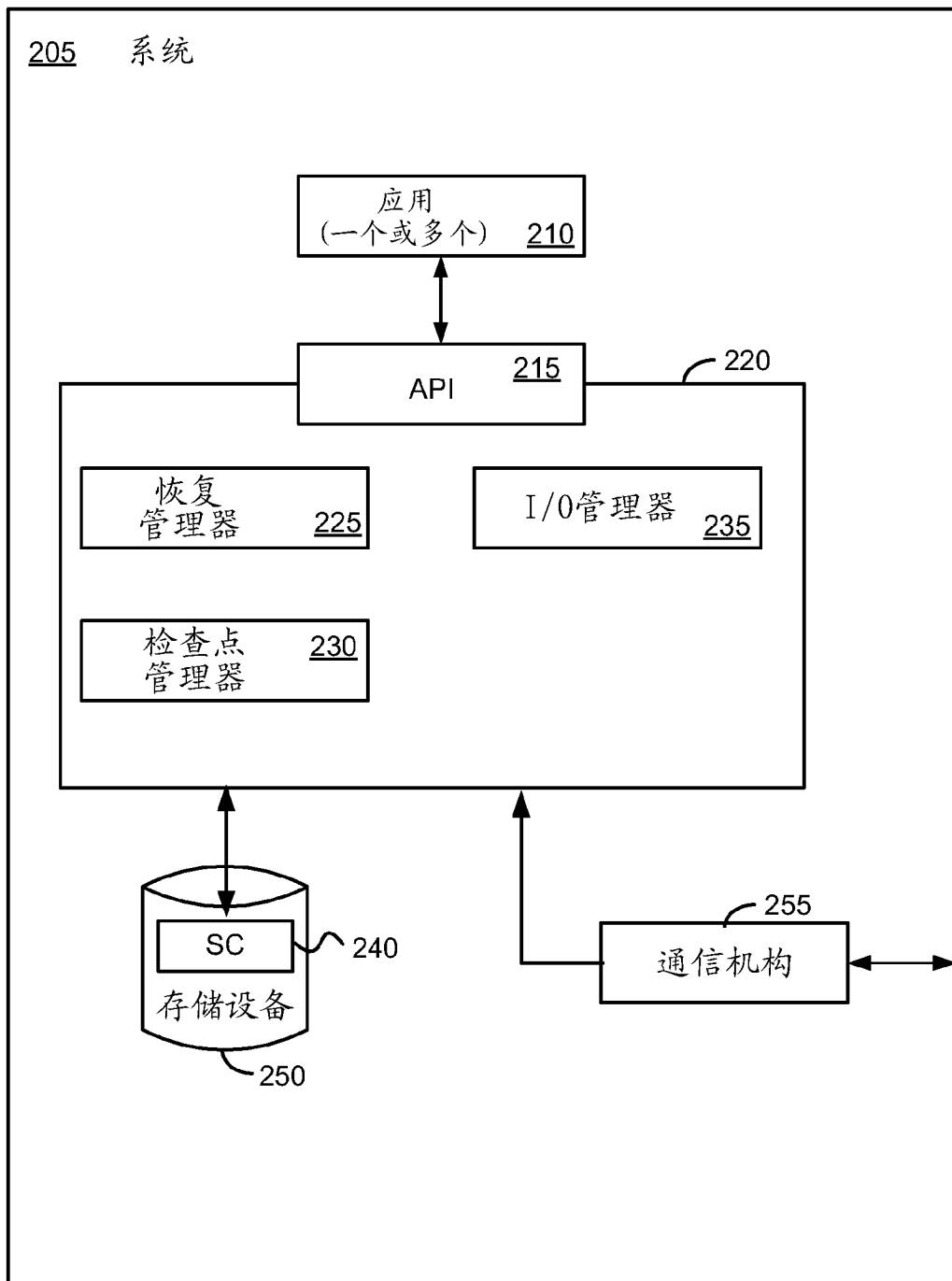


图 2

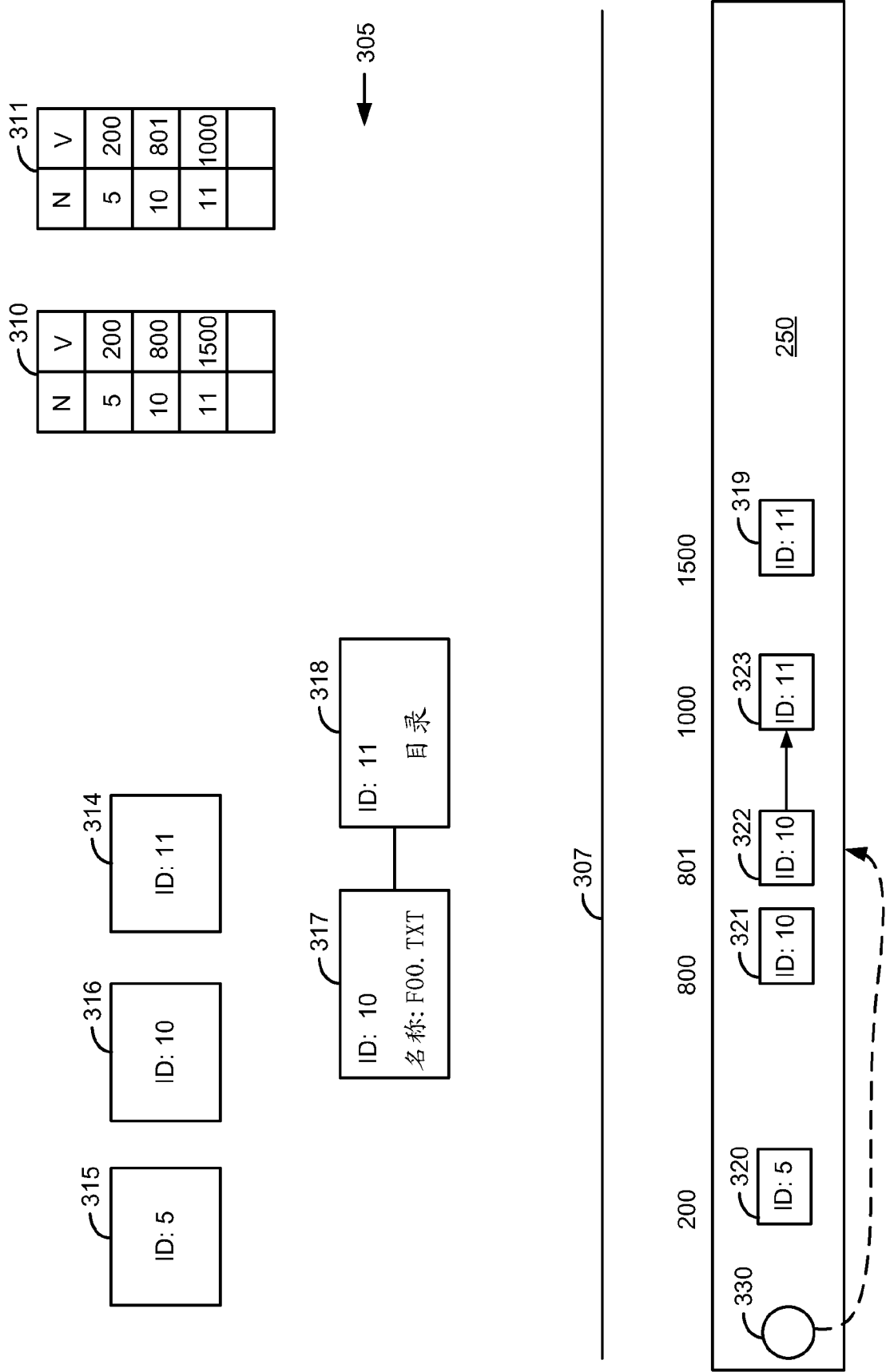


图 3

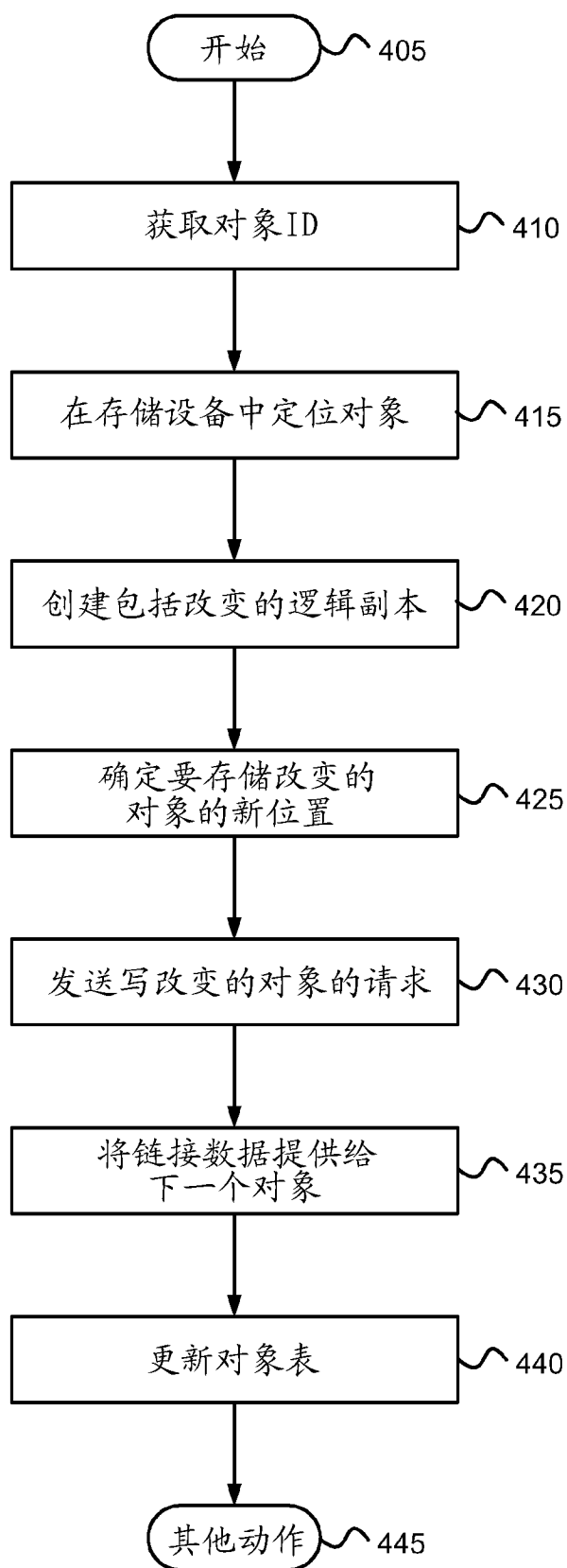


图 4

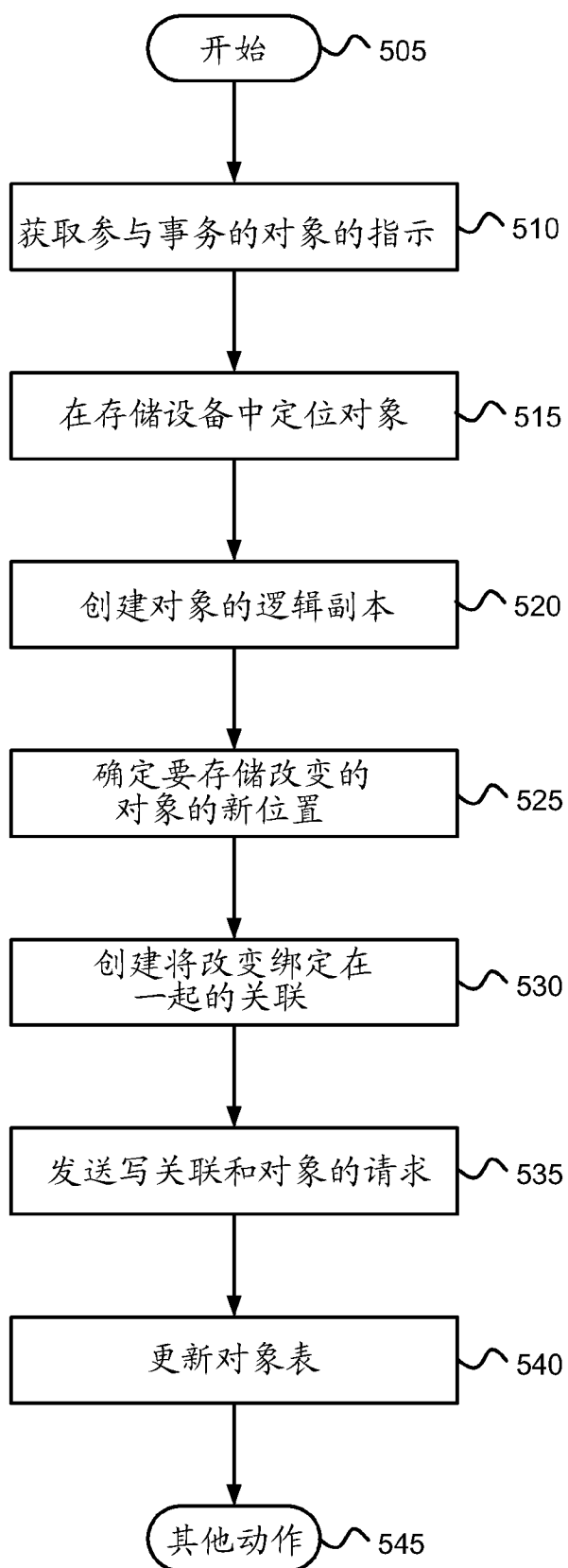


图 5