



(12)发明专利

(10)授权公告号 CN 105718548 B

(45)授权公告日 2019.04.26

(21)申请号 201610032143.X

(22)申请日 2011.08.31

(65)同一申请的已公布的文献号
申请公布号 CN 105718548 A

(43)申请公布日 2016.06.29

(30)优先权数据
12/875,815 2010.09.03 US

(62)分案原申请数据
201180050077.4 2011.08.31

(73)专利权人 赛门铁克公司
地址 美国加利福尼亚州

(72)发明人 F·郭

(74)专利代理机构 中原信达知识产权代理有限
责任公司 11219

代理人 周亚荣 安翔

(51)Int.Cl.
G06F 16/174(2019.01)

(56)对比文件
US 2005065986 A1,2005.03.24,
CN 101232450 A,2008.07.30,
CN 101196920 A,2008.06.11,
CN 101196920 A,2008.06.11,
US 2006047670 A1,2006.03.02,
US 6151685 A,2000.11.21,
US 6687815 B1,2004.02.03,
US 2007153575 A1,2007.07.05,

审查员 毕爽君

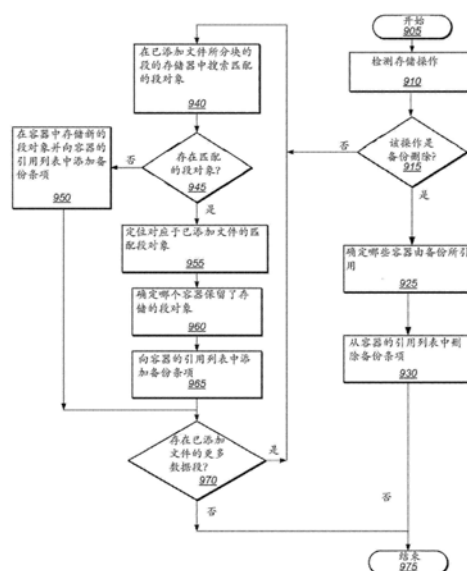
权利要求书2页 说明书11页 附图11页

(54)发明名称

基于去重复存储系统中用于可扩展引用管理的系统和方法

(57)摘要

本发明涉及基于去重复的存储系统中用于可扩展引用管理的系统和方法。一个存储装置被配置成在多个存储容器中存储多个存储对象,所述存储容器的每一个被配置成存储多个所述存储对象。维护一个存储容器引用列表,其中对于该多个存储容器的每一个,该存储容器引用列表识别多个文件中的哪些文件引用了一个给定存储容器内的一个存储对象。响应于检测到一个给定文件的删除,该文件引用该多个存储容器的一个特定存储容器内的一个对象,一个服务器被配置成通过从该存储容器引用列表中移除该给定文件的一个识别来更新该存储容器引用列表。将段对象与引用这些段对象的文件相关联的一个引用列表可以不响应于该删除而进行更新。



1. 一种用于管理数据存储的系统,包括:

去重复服务器,所述去重复服务器被配置成存储与一个或多个存储设备上的多个文件相对应的多个数据段;

多个数据段容器,其中,所述容器中的每一个容器包括两个或更多个数据段;以及

所述容器中的每一个容器的相应的容器引用列表,其中,每一个容器的容器引用列表识别所述多个文件中的哪些文件使用了该容器内的数据段中的至少一个数据段;

其中,响应于确定在特定容器的容器引用列表中所识别的文件的数目已经下降至阈值之下,所述系统被配置成针对所述特定容器中的每一个数据段创建相应的段对象引用列表,其中,所述特定容器中的每一个数据段的段对象引用列表识别所述多个文件中的哪些文件使用了该数据段。

2. 根据权利要求1所述的系统,其中,响应于确定在所述特定容器的容器引用列表中所识别的文件的数目已经下降至所述阈值之下,所述系统被进一步配置成中止维护所述特定容器的相应的容器引用列表。

3. 根据权利要求1所述的系统,其中,响应于确定在给定容器的给定容器引用列表中所识别的文件的数目已经上升到阈值之上,所述系统被配置成:

中止维护在所述给定容器中的每一个数据段的段对象引用列表;以及

维护所述给定容器的相应的容器引用列表。

4. 根据权利要求1所述的系统,其中,所述多个文件的子集被分组成备份,并且其中,对于所述容器中的每一个容器,所述容器引用列表识别所述备份中的哪些备份包括引用在给定容器内的段对象的文件。

5. 根据权利要求1所述的系统,其中,响应于对删除文件的请求,所述系统被配置成:

识别与所删除的文件相对应的第一容器;

确定多少其他文件与所述第一容器相对应;

响应于确定所述其他文件的数目大于给定阈值,而维护所述第一容器的容器引用列表;以及

响应于确定所述其他文件的所述数目小于所述给定阈值,而维护所述第一容器的段对象引用列表。

6. 根据权利要求5所述的系统,其中,响应于检测到所述对删除文件的请求,所述服务器进一步被配置成识别与所删除的文件相对应的其他容器,并且确定对于所述其他容器中的每一个其他容器,是维护容器引用列表还是维护段对象引用列表。

7. 根据权利要求1所述的系统,其中,所述服务器被配置成至少部分地基于以下至少一个来确定所述阈值的值:所述存储设备的存储利用和容器引用列表的大小。

8. 一种计算机实现的方法,包括:

在去重复存储系统中存储多个文件,其中,所述存储所述文件包括接收所述文件的多个数据段以及在一个或多个存储设备上存储所述数据段;

维护多个数据段容器,其中,所述容器中的每一个容器包括两个或更多个数据段;

维护所述容器中的每一个容器的相应的容器引用列表,其中,每一个容器的容器引用列表识别所述多个文件中的哪些文件使用了该容器内的数据段中的至少一个数据段;以及

响应于确定在特定容器的容器引用列表中所识别的文件的数目已经下降至阈值之下,

针对所述特定容器中的每一个数据段创建相应的段对象引用列表,其中,所述特定容器中的每一个数据段的段对象引用列表识别所述多个文件中的哪些文件使用了该数据段。

9.根据权利要求8所述的计算机实现的方法,其中,响应于确定在所述特定容器的容器引用列表中所识别的文件数目已经下降至所述阈值之下,所述方法包括:中止维护所述特定容器的相应的容器引用列表。

10.根据权利要求8所述的计算机实现的方法,其中,响应于确定在给定容器的给定容器引用列表中所识别的文件数目已经上升到阈值之上,所述方法包括:

中止维护在所述给定容器中的每一个数据段的段对象引用列表;以及

维护所述给定容器的相应的容器引用列表。

11.根据权利要求8所述的计算机实现的方法,其中,所述多个文件的子集被分组成备份,并且其中,对于所述容器中的每一个容器,所述容器引用列表识别所述备份中的哪些备份包括引用在给定容器内的段对象的文件。

12.根据权利要求8所述的计算机实现的方法,其中,响应于对删除文件的请求,所述方法进一步包括:

识别与所删除的文件相对应的第一容器;

确定多少其他文件与所述第一容器相对应;

响应于确定所述其他文件的数目大于给定阈值,而维护所述第一容器的容器引用列表;以及

响应于确定所述其他文件的所述数目小于给定阈值,而维护所述第一容器的段对象引用列表。

13.根据权利要求12所述的计算机实现的方法,其中,响应于检测到所述对删除文件的请求,所述方法包括识别与所删除的文件相对应的其他容器,并且确定对于所述其他容器中的每一个,是维护容器引用列表还是维护段对象引用列表。

14.根据权利要求8所述的计算机实现的方法,进一步包括至少部分地基于以下至少一个来确定所述阈值的值:所述存储设备的存储利用和容器引用列表的大小。

基于去重复存储系统中用于可扩展引用管理的系统和方法

[0001] 分案说明

[0002] 本申请属于申请日为2011年8月31日的中国发明专利申请201180050077.4的分案申请。

技术领域

[0003] 本发明总体上涉及备份存储系统,尤其是涉及用于协助基于去重复的存储系统中的资源回收的引用列表。

背景技术

[0004] 各种组织机构正在积累并存储大量电子数据。因此,备份存储系统正在增加尺寸并消耗大量资源。为了应对数量不断增加的数据的存储,去重复技术已经成为在备份存储系统中使存储利用率最大化的一个重要特征。在典型的去重复系统中,将文件分块为数据段并且将冗余数据段从系统中删除。然后,在备份存储介质中将唯一的数据段存储为段对象。随着所存储的段对象数量的增加,段对象的管理要求提高系统资源的共享,这可能影响去重复系统的整体效率和性能。

[0005] 基于去重复系统的目的是减少需要存储大量数据的大量存储容量。去重复技术已经成熟到它们可以在数量上实现所存储数据的显著减少。然而,尽管这些技术可以减少所需要的存储空间,但系统中存储的段对象的数量仍然继续增加。当去重复系统按比例增加以处理更多的数据负载时,段对象的管理和标引就成为影响系统性能的重要因素。

[0006] 典型地,段对象尺寸小,在一些系统中仅有4千字节(KB)。对于存储400兆兆字节(TB)数据的系统(所有段对象大小为4KB),要维护1000亿个段对象。随着存储需求的增长,段对象数量上的增加可能产生无法接受的管理开销。因此,需要高可扩展性管理系统来有效地存储和管理大量段对象。

[0007] 特别具有挑战性的问题包括在从系统删除文件后恢复资源。当删除一个文件时,不能简单地删除组成该文件的段对象,因为系统所存储的一些其他文件有可能引用了那些相同段对象的一个或多个。只要没有其他文件使用那些段对象,就可以将它们删除。某种管理形式需要持续跟踪段对象以及使用这些段对象的所有文件。存在用于管理段对象以及指向它们的文件的各种技术,当小规模操作时这些技术中的大部分可以工作良好。然而,在应对大量段对象时这些方法中的很多可能不会有效。

[0008] 协助资源回收的一个技术是对段对象进行引用计数。引用数存储了一个值,该值表示有多少文件指向或使用该段。每当一个文件使用段对象时它的引用数就递增,并且当删除使用该段的文件时递减,最终当该数降为零时可以回收该段。

[0009] 引用计数具有不适合于去重复的一些限制。一个限制是任何丢失或重复更新会不适当地改变该计数。如果该数偶然减少,当段仍由至少一个文件使用时可能被删除。如果该数偶然增加,那么甚至在使用该段的所有文件都从系统中删除后可能永远不会删除该段。

[0010] 引用计数的另一个缺点是它不允许识别哪些文件使用一个给定段对象。如果段对

象遭到破坏,备份系统需要知道哪些文件正在使用它,这样可以要求该文件恢复破坏的数据。然而,引用计数并不维护哪些文件正使用每个特定段对象的列表,这使得破坏数据的恢复更为困难。

[0011] 可用于协助资源回收的另一个工具是引用列表。维护引用列表不会受到引用计数的内在缺点的影响。引用列表对错误更新具有更强的免疫力,因为可以搜索该列表以查看是否已经执行了添加或移除操作。另外,引用列表具有识别哪些文件正使用每个段对象的能力。然而,引用列表不易于扩展以便处理大量段对象。传统上,根据所存储的每个段对象在一个细等级上来管理引用列表。随着段对象数量的增加,更新引用列表耗费更长的时段,这可能降低系统性能。需要一种新方法,以维护可以有效管理大量段对象的引用列表。

[0012] 鉴于以上问题,维护去重复系统中的引用列表的改进方法和机制是需要的。

发明内容

[0013] 设想了有效管理基于去重复的存储系统中的引用列表的方法和机制的各种实施例。在一个实施例中,引用列表可以包括在备份存储介质中存储的每个容器的粗等级条项组成。由在特定容器内存储的至少一个段对象组成的每个文件可以在该特定容器的引用列表中具有一个条项。当文件添加到基于去重复的存储系统中或从其删除时,条项可以添加到引用列表或从其删除。在另一个实施例中,引用列表可以包括用于容器的粗等级条项以及用于在容器中存储的段对象的细等级条项。引用列表可以在粗等级上进行管理,这样存储系统的文件的删除可以产生更新的容器条项,无需更新段对象条项。当用于一个特定容器的粗等级条项的数量减少时,最终该数量将降至一个阈值以下,在该阈值点上服务器可以切换回在细等级上管理该特定容器的列表。在细等级上管理引用列表可以包括每次从系统中删除一个文件时就更新段对象条项。

[0014] 在又一个实施例中,引用列表可以将每个条项与一个备份事务相关联,而不是将每个条项与一个文件相关联。备份事务可以包括由一个单一客户端向基于去重复的存储系统发送的所有文件以进行一次单一备份操作。引用列表可以包括在备份存储介质中存储的每个容器的粗等级条项。由一个特定容器中存储的至少一个段对象组成的每个备份事务可以在该特定容器的引用列表中具有一个条项。在又一个实施例中,引用列表可以具有一个备份事务所引用的每个容器的粗等级条项以及一个备份事务所引用的每个段对象的细等级条项。引用列表可以仅在粗等级上进行更新,直到一个特定容器的粗等级条项的数量降至一个阈值以下,在该阈值点上服务器可以切换回在细等级上管理该特定容器的列表。根据备份事务组织该引用列表可以进一步减少列表中条项的数量,并且响应于备份事务添加到系统中或从其中删除而减少需要处理列表的处理时间。

[0015] 鉴于以下本文提出方法的详细描述,这些以及其他特征和优点对本领域的普通技术人员将变得显而易见。

附图说明

[0016] 通过参考以下描述并结合附图,本方法和机制的上述和进一步的优点可以得到更好的理解,在附图中:

[0017] 图1展示了一种基于去重复的存储系统的一个实施例。

[0018] 图2展示了一个备份事务的一个实施例,该备份事务作为段对象存储在备份存储中的一个容器内。

[0019] 图3展示了文件及其相关段对象引用的一个实施例。

[0020] 图4展示了一个存储段对象的容器和一个容器引用列表的两个实施例。

[0021] 图5展示了具有粗和细等级条项的一个面向文件的引用列表的一个实施例。

[0022] 图6展示了一种用于维护存储容器引用列表的方法的一个实施例。

[0023] 图7展示了在第一删除操作之后的一个引用列表的一个实施例。

[0024] 图8展示了在第二删除操作之后的一个引用列表的一个实施例。

[0025] 图9展示了一个面向备份的引用列表的一个实施例,该列表具有用于备份事务的条项。

[0026] 图10是一个总体流程图,该流程图展示了一种用于在文件添加或删除操作之后更新引用列表的方法的一个实施例。

[0027] 图11是一个总体流程图,该流程图展示了一种用于更新引用列表的方法的一个实施例。

具体实施方式

[0028] 在以下描述中,给出了很多具体细节以透彻理解本文提出的方法和机制。然而,本领域普通技术人员应当认识到没有这些具体细节也可以实施各种实施例。在一些示例中,并未详细示出众所周知的结构、部件、信号、计算机程序指令以及技术,以避免混淆本文所述的方法。

[0029] 应当认识到为了进行简单而清晰的解释,图中所示的元素不一定按照比例绘出。例如,可以相对于其他元素而放大一些元素的尺寸。另外,在认为适当的时候,参考号在图例之间重复以表示相应的元素。

[0030] 图1展示了一种基于去重复的存储系统100的一个实施例。该基于去重复的存储系统100包括客户端110、120和130,这些客户端代表任意数量的移动或静态客户端。尽管该图例将两个桌上计算机和一个膝上计算机的示例展示为客户端,但包括个人数字助理、蜂窝电话、智能电话、数字相机、摄像机、无线阅读装置以及能够发送和接收数据的任意其他类型电子装置的其他客户端装置是可能的并且是可以设想的。如图1所示,客户端连接至网络140,通过该网络它们还连接至去重复服务器150。该去重复服务器150可用于各种不同目的,例如提供客户端110、120和130访问共享数据和备份关键任务数据。

[0031] 通常,去重复服务器150可以是任意类型的物理计算机或计算装置。去重复服务器150可以包括总线,该总线可以将服务器150的主要子系统或组件相互连接,例如一个或多个中央处理单元(CPU)、系统存储器(随机访问存储器(RAM)、只读存储器(ROM)、闪速RAM等)、输入/输出(I/O)装置、永久存储装置(例如硬盘)、以及通常包含在计算机中的其他外围装置。去重复服务器150可以具有分布式架构,或它的所有组件可以集成为一个单一单元。去重复服务器150可以管理运行软件进程和应用程序的操作系统,并且软件可以运行在服务器的CPU上并且可以存储在服务器的存储器中。另外,该基于去重复的存储系统100可以包括一个或多个去重复服务器150。

[0032] 去重复服务器150还可以连接至备份存储160,在该存储器上可以存储客户端110、

120和130的数据。备份存储160可以包括不同类型的一个或多个数据存储装置,例如硬盘驱动器、光盘驱动器、磁盘驱动器、可换式磁盘驱动器和其他驱动器。备份存储160可以存储引用列表170,并且引用列表170可以由去重复服务器150来管理。在另一个实施例中,引用列表170可以存储在去重复服务器150的存储器中。在又一个实施例中,引用列表170可以由一个实体而非去重复服务器150来管理和存储。引用列表170可以为去重复服务器150提供一种途径,以跟踪客户端110、120和130的多少个文件或备份事务正在使用备份存储160中存储的段对象的每一个。

[0033] 在一个实施例中,引用列表170可以包含在备份存储160中存储的容器的粗等级条项。容器可以是与文件系统的可变尺寸部分相关联的逻辑实体,该文件系统包括数据存储的多个已分配单元。另外,容器可以映射到备份存储介质中的物理单元。对于备份存储介质中的每个容器,引用列表170可以包含用于每个独立文件的不同粗等级条项,该独立文件引用了在该特定容器内存储的多个段对象的一个或多个。因此,容器在引用列表中的粗等级条项的数量可以等于引用该容器内的至少一个段对象的不同文件的数量。在另一个实施例中,引用列表还可以包含用于容器中存储的段对象的细等级条项。对于在容器内存储的每个段对象,引用列表可以包含引用特定段对象的每个文件的细等级条项。因此,段对象在引用列表中的细等级条项的数量可以等于引用该段对象的不同文件的数量。

[0034] 连接至网络140的客户端的一个或多个还可以用作其他客户端的服务器。本文描述的方法可以在各种网络中使用,这些网络包括局域网 (LAN) (例如以太网、光纤分布式数据接口 (FDDI) 网、令牌环网、以及基于电气与电子工程师协会 (IEEE) 802.11标准 (Wi-Fi) 的无线局域网 (WLAN))、和广域网 (WAN) (例如互联网、蜂窝数据网)、以及其他数据通信网络的组合。本文描述方法所服务的网络还可以包含多种备份存储介质160,这依赖于每个特定网络的独有的存储和备份要求。与备份存储160相关联的存储介质可以根据各种存储架构来实施,这些架构包括但不限于附网存储环境、存储区域网络 (SAN)、以及直接连接至去重复服务器150的磁盘组合。

[0035] 客户端110、120和130可以在网络140上向去重复服务器150发送数据。数据可以采用数据段的形式来发送,这些数据段通过将客户端110、120和130上存储的数据分块为一个或多个预定尺寸的块来创建。在不同的实施例中,客户端可以包括协助备份操作(例如,备份代理)的软件。在一些实施例中,去重复服务器150可以对接收到的数据进行去重复。去重复通常涉及确定接收到的数据段是否已经存储在备份存储160中。如果该数据段已经存储在备份存储160中,接收到的数据段可以舍弃,并且可以在适当的位置使用已经存储的数据段(也称为段对象)的指针。以此方式,去重复服务器150可以争取在备份存储160中只保留任意段对象的一个单一备份。在其他实施例中,去重复过程可以在数据段发送到去重复服务器150之前进行,这样只有新的数据段可以发送到去重复服务器150,并且可以在客户端110、120和130上删除所有冗余数据段。基于去重复的存储系统100被示为包括多个客户端和一个服务器,但在替代实施例中,客户端和服务器执行的功能可以由对等配置中的节点来执行,或由客户端、服务器和节点的组合来执行。

[0036] 在其他实施例中,数据还可以作为完整的数据文件、作为从镜像文件或卷拷贝的多个数据文件、作为虚拟机磁盘文件 (VMDK)、作为虚拟硬盘 (VHD)、作为 **SYMANTEC®** BackupExec软件产品创建的磁盘镜像文件 (.V2I)、作为 .TAR 档案文件(该文件进一步包括

将数据文件存储为原始磁盘分区的VMDK文件)、或作为以其他方式可由客户端110、120和130格式化的文件从客户端110、120和130发送至去重复服务器150。

[0037] 现在参见图2,图中示出了一种基于去重复的存储系统。客户端110通过网络140连接至去重复服务器150。去重复服务器150连接至备份存储160,备份存储在逻辑数据存储容器210中将引用列表170和客户端110的数据存储为段对象231-239。在容器中可以存储任意数量的段对象。另外,段对象231-239的尺寸可变。在另一个实施例中,段对象231-239的尺寸可以相同。

[0038] 客户端110具有一组文件241-244,该组文件构成一个单一备份事务250,客户端110可以将该备份事务发送至去重复服务器150以便存储在备份存储160中。在从客户端110发送至去重复服务器150之前或之后可以将文件241-244分块为不同大小的数据段。另外,这些数据段可以由客户端110或去重复服务器150进行去重复。在一个实施例中,在一个单一备份操作中备份事务250可以包括由一个单一客户端备份的所有文件。在另一个实施例中,备份事务250可以包括来自一个单一客户端或来自多个客户端的多个文件。在又一个实施例中,备份事务250可以包括至少部分基于备份存储介质160中的段对象的临近性而集合在一起的多个文件,这些段对象由多个文件引用。将文件组成为备份事务的其他分组方法是可能的并且是可设想的。

[0039] 去重复服务器150可以将从备份存储160中的备份事务250创建的去重复数据段存储为段对象231-239。去重复服务器150可以创建容器210以便存储段对象231-239。去重复服务器150还可以在备份存储160中创建附加容器。在一个实施例中,容器全部可以具有相同的大小。在另一个实施例中,容器可以具有不同的大小。

[0040] 现在转至图3,图中示出了一组文件260和相关段对象引用270。文件241-244被示为它们将从方框260中的段对象231-239来重构。客户端110(图2)的文件241-244可以分块为数据段,然后这些数据段可以在备份存储160(图2)中存储为段对象231-239。每个段对象231-239可以由多于一个的文件来引用。

[0041] 在所示的示例中,文件241可以包括或从5个段对象重构:231、234、235、236以及237。文件242可以从7个段对象重构:231、233、234、236、237、238以及239。文件243可以从6个段对象重构:231、232、234、235、237以及238。文件244可以从4个段对象重构:231、232、233以及234。段对象中的大部分由4个文件241-244至少一次地引用,但每个段对象只有一个备份存储在容器210(图2)内的备份存储160中,从而减少了需要存储4个文件241-244的总存储容量。

[0042] 在图3中还示出了段对象引用270,每个段对象231-239具有引用段对象的文件的相关列表。引用列表270的很多可能实施例是有可能的。例如,在一个实施例中文件的链表可以与每个段对象标识符相关联。B-树结构或其他方式可以用于存储和维护列表270。很多这类实施例是可能的并且是可设想的。在一个实施例中,如果删除了一个文件,可以遍历段对象标识符231-239以便移除那些识别已删除文件的条项/实体。如可以认识到的是,有必要遍历很多条项以完全更新这个或这些数据结构270。一般而言,与这种删除相关的总开销相对较高。在以下讨论中,描述了一种替代方法。

[0043] 现在转至图4,在方框280中示出了包含段对象231-239的容器210。一般而言,存储在系统内的所有段对象在逻辑上可以存储在容器中。在所示的单一示例中,容器210包括6

个段对象。然而,可以对容器进行配置以包括任意数量的段对象——几百、几千或更多。因此,容器的数量是段对象数量的一小部分。除了以上所述,在方框290中示出了容器210(图2)的容器引用列表170的两个实施例。第一实施例示为链表,而第二实施例示为表。

[0044] 容器引用列表识别引用容器内的一个段对象的每个文件。容器引用列表170的第一实施例描绘为与文件241-244相关的容器引用210,这些文件中的每一个引用在该容器中存储的至少一个段对象。如同前文讨论的段对象引用列表,任意适合的数据结构都可用于维护容器引用列表。在所示的第一实施例中,描绘了链表型结构,其中容器标识符210具有文件标识符链表,这些标识符引用容器210内的段对象。如前所述,可以采用B-树、双向链表以及其他数据结构。具有头“容器”和“文件”的容器引用列表170包括用于容器210的粗等级条项。容器引用列表170仅是为了示例的目的而呈现;可以根据本文所述的方法和机制采用其他实现容器引用列表的方式。还需要注意的是,本文描述的引用列表可以作为一个或多个列表或结构来维护。如果维护多个列表,给定的列表可能与特定的数据集、特定类型的数据、数据的用户、特定备份等相关联。

[0045] 除了链型结构外,可以采用更多面向数组类型的结构。例如,在一个实施例中可以采用动态可分配n-维数组。在图4的示例中,所示的2-维数组用于容器210,每个条项用于每个文件241-244。以此方式,在每个文件的引用列表170中存在一个粗等级条项,该文件引用在容器210中存储的段对象的至少一个。4个文件241-244应用容器210中存储的引用段对象。因此,在应用列表中存在容器210的4个粗等级条项——一个条项用于引用该容器内存储的段对象的每一个文件。

[0046] 如上所提及,上文描述的容器引用列表在存储系统中具有段对象引用列表的条项的一小部分。因为采用这种容器引用存储列表,现在描述具有更少开销的维护引用列表的一种方法。图5展示了一种用于维护“面向文件”的容器引用列表的方法的概览的一个实施例。容器列表据称是面向文件的,因为每个容器具有引用该容器中至少一个对象的文件的列表。如前文讨论的,遍历和维护段对象引用列表可能需要相对较多的开销。特别是当删除文件时,段对象引用列表的遍历和更新是相对耗时的。作为这种方法的替代,以下方法描绘了一种途径,在这种途径中常常忽略段对象引用。以此方式,减少了与维护这种列表相关的开销。

[0047] 图5的方法开始于检测文件操作(模块510)。如果该操作不是文件删除操作(决定模块515),那么可以将文件分块并进行搜索以匹配已经在系统内存储的对象(模块540)——这就是在去重复存储系统中的情况。如果存在一个已经存储的匹配段对象(决定模块545),文件的识别被添加到包括该匹配段对象的容器的容器引用列表(模块565),并且如果有该文件的剩余数据段要处理(决定模块570),可以重复该过程。另一方面,如果不存在已经存储的匹配段对象(决定模块545),那么可以将数据存储为新的段对象,并且更新容器应用列表以包括用于容器(包括新的段对象)的文件的标识(模块550)。

[0048] 如果结果表明检测到的文件操作是文件删除操作(决定模块515),那么就从容器引用列表中删除文件的标识(模块530)。应当注意的是,在一个实施例中此时不更新或维护段对象引用列表。相反,只更新容器引用列表以反映检测到的文件。由于在系统中容器远少于段对象,并且容器引用列表包括段对象引用列表的条项的一小部分,与更新容器引用列表相关联的开销远少于段对象列表的开销。在以下讨论中,将展示很多示例,这些示例示出

了容器和段对象引用列表的维护。为了便于说明,示例将展示排序格式的列表和条项。然而,如上文所提及,实际的实施例可以是链结构、树结构或其他方式。另外,尽管这种讨论可以将粗和细条项描绘为单一列表的一部分,但应当理解的是实际上可以存在多个维护的列表。

[0049] 现在参见图6,图中示出了具有粗和细等级条项的用于容器210(图2)的引用列表500。如图4中所示,同时示出了表和链表格式。引用列表500包括用于容器210的粗等级条项,该容器可以位于备份存储160中(图2),并且包括用于容器210内存储的段对象的细等级条项。在另一个实施例中,引用列表500可以包含用于在备份存储160中存储的多个容器的条项。在又一个实施例中,引用列表500可以包含在备份存储160中存储的所有容器的条项。在再一个实施例中,去重复服务器150(图2)可以维护在备份存储160中存储的每个容器的独立引用列表。

[0050] 在每个文件的引用列表500中存在粗等级条项,该文件引用了存储在容器210中的段对象的至少一个。4个文件241-244(图3)引用了在容器210中存储的段对象,因此在引用列表中存在4个用于容器210的条项,一个条项用于指向在容器内存储的段对象的文件的每一个。用于容器210的这些条项是引用列表500的粗等级条项。用于段对象的每一个的条项是引用列表500的细等级条项。每个段对象可以包含引用它的每个文件的细等级条项。如果段对象可用于在检索或恢复操作过程中重新创建一个文件,该文件可以引用此段对象,或者以其他方式构成组成该文件的数据的一部分。

[0051] 在引用列表500中段对象231包含用于4个文件(241-244)的4个细等级条项,这些文件指向该段对象。另外,在该列表中段对象232包含用于文件243和244的2个文件等级条项,段对象233包含用于文件242和244的2个细等级条项,段对象234包含用于文件241-244的4个细等级条项,段对象235包含用于文件241和243的2个细等级条项,段对象236包含用于文件241和242的2个细等级条项,段对象237包含用于文件241-243的3个细等级条项,段对象238包含用于文件242和243的2个细等级条项,而段对象239包含用于文件242的1个细等级条项。

[0052] 如图6中所示,细等级条项可以紧接在引用列表500中的粗等级条项之后。在一个实施例中,如果引用列表500包含不止一个容器的条项,那么第一容器的粗和细等级条项可以集合在一起,接着是第二容器的粗和细等级条项,以此类推是容器的其余粗和细等级条项。在另一个实施例中,所有容器的粗等级条项可以集合在一起,接着是所有容器的所有细等级条项。将粗和细等级条项集合在一起并组织引用列表500的其他方法是可能的并且是可以设想的。

[0053] 现在转至图7,图中示出了从存储系统中删除了文件243后已经更新该列表之后的容器210的引用列表500。另外,图中同时示出了表和链表格式。如图7中所描述,引用列表只对粗等级条项进行更新。这种引用列表的阈值可以是任意所需的数值,例如3。因此,当指向容器210的文件的数量低于3时,引用列表可以切换更新粗和细等级条项。在另一个实施例中,阈值可以采用不同的值。在又一个实施例中,服务器150(来自图1)可以至少部分地基于备份存储160(来自图1)中当前使用的存储空间的百分比来确定阈值。在又一个实施例中,服务器可以至少部分地基于引用列表中的条项的大小或数量来确定阈值。可以使用任意所需的条件来设定或确定阈值。

[0054] 在已经删除文件243的条项之后容器210在引用列表500中具有3个粗等级条项。引用列表500中由文件243引用的段对象的条项仍保留在列表中。由于引用列表500仅对粗等级条项进行更新,当删除一个文件时不会删除细等级条项。在粗等级上更新引用列表的优点是它可以加速更新列表的进程,因为存在更少的条项要处理。在引用列表500的情况中,当删除文件243时仅有一个粗等级条项被删除。另外,可能只有4个条项(粗等级条项)需要进行处理以便确定是否删除的文件引用了容器。如果引用列表500已经在细等级上进行了更新,对于文件243所指向的段对象的每一个,就已经删除了6个额外条项。另外,如果引用列表500已经在细等级上进行了更新,所有的细等级条项就得到处理。在存储大量容器和段对象的大规模基于去重复的存储系统中,仅更新这个或这些引用列表的粗等级条项可能在很大程度上减少在删除一个文件或一组文件之后执行的更新和处理操作的数量。

[0055] 在粗等级上更新引用列表存在一个缺点。如果容器内的段对象的一些不被任何文件所使用,引用列表可能不会显示这一点。这可能产生消耗存储空间的未使用段对象,否则这些存储空间可以得到释放并重新使用。为了减少未使用段对象的存储,当一个特定容器的粗等级条项的数量降到一个阈值以下时,该容器的引用列表条项可以在细等级上进行更新。当只存在特定容器的少量粗等级条项时,段对象更有可能进行回收,因此切换到细等级更新可以比仅仅利用粗等级更新更快地促进资源回收。另外,当存在少量特定容器的粗等级条项时,相比于如果存在大量粗等级条项,切换到细等级更新可能只是稍微增加了更新列表的的处理负担。

[0056] 在引用列表切换到用于具体容器的细等级更新后,新的文件可以添加到备份存储系统,这些新文件引用了在该特定容器存储的引用段对象。如果引用该容器的文件数量超过阈值,引用列表可以切换回该容器的粗等级更新。引用列表从细到粗等级更新切换回和切换去的次数可以与特定容器的粗等级条项的数量在两个方向上跨越阈值的次数一样多。

[0057] 现在参见图8,图中示出了已经从存储系统中删除文件242之后的容器210的引用列表500(表和链表格式)。在删除文件242之后,容器210的粗等级条项的数量是2个。因此,引用列表500可以切换到更新粗和细等级条项,因为条项的数量已经降至阈值3以下。在其他实施例中,引用列表可以具有不同的阈值,并且引用列表可以在不同数量的粗等级条项上从粗等级更新向细等级更新切换。

[0058] 在图8中,可以从引用列表500中删除用于文件242的容器210的粗等级条项。另外,段对象(或细等级)条项也可以进行更新。可以从列表中删除用于文件242的所有细等级条项。另外,因为用于文件243的粗等级条项不再存在,可以在优先操作中删除该条项,可以从列表中删除文件243的所有细条项。当删除了文件243时,如图7中所示,引用列表500处于粗等级更新模式,并且仅从列表中删除文件243的粗等级条项。在所有引用列表500切换到细等级更新之后,可能需要更新细等级条项以匹配容器210的粗等级条项。这允许列表精确地反映有多少文件引用了每个段对象。如图8中所示,在从引用列表500删除与文件242和243相关联的所有细等级条项之后,段对象238和239不会由任意文件引用。因此,可以删除这两个段对象,并且这些对象所占据的存储空间可以重新使用。可以立即删除这些段对象,或者可以对它们做删除标记并随后在涉及其他未使用段对象的批操作中进行删除。在进一步的实施例中,标记和回收段对象的其他方法是可能的并且可以设想。

[0059] 当将多个文件添加到备份存储系统时,可以将这些文件分块为与已经存储的段对

象相同的数据段。可以更新用于存储这些相同段对象的容器的引用列表。在一个实施例中，如果粗等级条项的数量低于阈值，那么只更新粗引用列表。一旦删除了文件并且粗等级引用列表达到阈值，那么可以进行重建细引用列表。以此方式，当需要时只需对细等级引用列表采取动作。如果粗引用列表容器很少达到阈值，就完全不存在细引用更新开销。在一个替代实施例中，当将多个文件添加到容器中时，可以同时也在细和粗等级上更新引用列表，即使是粗等级条项的数量低于阈值。在这种实施例中，新存储的文件所引用的段对象可以存储在正在引用列表的粗等级上进行处理。对于在粗等级上处理的容器，当将一个文件添加到备份存储系统中时，用于这些容器的段对象条项仍然可以进行更新。

[0060] 在一些实施例中，容器可以将其所有粗等级条项从引用列表中删除，无需更新细等级条项。这可以在容器的引用列表只包含粗等级条项时发生。这还可以在容器的引用列表包含粗和细等级条项并且阈值为零时发生。或者这可以在同时删除一组文件并且在一个操作中删除容器的所有粗等级条项时发生。当为特定容器删除所有粗等级条项时，该容器的段对象可以进行回收或标记为准备进行回收，无需对引用列表的细等级条项进行更新或处理。这种做法可以通过回收段对象所使用的资源而节省处理时间和开销，不必处理引用列表的细等级条项。

[0061] 现在转至图9，图中示出了用于容器210的“面向备份的”引用列表800。该容器引用列表是面向备份的，因为每个容器具有备份列表，这些备份引用了容器中的至少一个对象。因此，相比于图5-7中的引用列表500，引用列表800包含与备份事务250相关联的条项。如同前述示例，图中示出了表和链表格式。如图2中所示，备份事务250包含文件241-244。在删除文件243和242之前，图9中的引用列表800对应于图6的引用列表500。根据备份事务而非根据每个单独的文件通过跟踪容器210和段对象231-239，已经减少了引用列表800中的条项数量。减小引用列表800的大小可以降低存储列表所需要的存储空间，并且当将多个备份事务添加到存储系统或从其删除时可以减少需要处理列表中条项的处理时间。在一个实施例中，引用列表800可以包含引用了容器或段对象的备份事务的每个示例的一个条项。在另一个实施例中，去重复服务器150(图2)可以将多个备份事务组织为一组备份事务，并且引用列表800可以包含引用容器或段对象的一组备份事务的每个示例的条项。在进一步的实施例中，文件和备份事务的其他分组可用于确定引用列表800如何记录条项。正如可以认识到的是，尽管描述了包括多个文件的备份事务，还可以使用其他可识别的文件分组方法。

[0062] 另外，引用列表的粗等级条项可以对应于多于一个的容器。例如，在一个实施例中，多个容器可以集合在一起。该多个容器可以存储来自一个备份事务的数据。或者，可以选择该多个容器并基于其他因素而集合在一起。引用列表可以组织为粗等级条项对应于多个容器而不是一个单一容器。以此方式组织引用列表可以产生具有更少条项的较小引用列表，并且在文件或备份事务添加到系统或从其删除时可以产生更快的更新处理。

[0063] 现在转至图10，图中示出了一种用于维护面向备份的引用列表的方法的实施例。为了进行讨论，本实施例中的步骤以顺序次序示出。应当注意的是，在以下描述的方法的各种实施例中，可以按照不同于所示的次序同时执行所描述的要素的一个或多个，或者可以完全忽略。也可以按照所期望的那样执行其他附加要素。

[0064] 图10的方法开始于模块905，然后可以在模块910检测存储操作。由于本图例总体上讨论备份操作，存储操作可以执行新的备份或删除先前的备份。在条件模块915，如果确

定该操作是删除备份,那么它可以确定容器引用列表的哪个容器包括被删除的备份的识别(模块925)。然后,可以删除容器的引用列表中删除的备份的条项(930)。

[0065] 如果检测的操作是新的备份(条件模块915),那么对于添加的每个文件可以在存储器中搜索一个匹配的段对象(模块940),该段对象与从添加的文件分块的一个数据段相同。如果存在一个匹配段对象(条件模块945),那么可以定位该匹配段对象(模块955)。如果不存在匹配的段对象(条件模块945),那么新段对象(对应于所添加文件的数据段)可以存储在容器中并且一个文件条项可以添加到容器的引用列表(模块950)。

[0066] 在定位匹配的段对象后(模块955),可以确定哪个容器保留该匹配的段对象(模块960)。接下来,对应于新文件的备份事务的条项添加到容器的引用列表中(模块965)。如果备份事务已经具有该容器的条项,可以不需要新的条项。如果存在来自自己添加文件的更多数据段(条件模块970),那么该方法可以返回到模块940以搜索匹配的段对象。如果不再有来自自己添加文件的数据段(条件模块970),那么该方法可以在模块975结束。

[0067] 尽管已经讨论了面向文件的和面向备份的容器引用列表的实施例,但在不同实施例中可以同时维护这些实施例(包括段对象应用列表)的组合。在这些实施例中,可以利用不同条件来确定在既定状况中是否进行更新以及更新哪个引用列表。

[0068] 现在参见图11,图中示出了一种基于上述方法和机制的混合方法的一个实施例。在此示例中,描述了一种介于容器引用列表和段对象引用列表之间的混合方法。方法1000展示了用于确定是否维护容器引用列表或段对象引用列表的方法的一个实施例。为了进行讨论,本实施例中的步骤以顺序次序示出。应当注意的是,在以下描述的方法的各种实施例中,可以按照不同于所示次序的次序同时执行所描述的一个或多个要素,或者可以完全忽略。也可以按照所期望的那样执行其他附加要素。

[0069] 在以下讨论中,面向文件的容器引用列表用于讨论的目的——类似于图5中所讨论的内容。然而,该方法还可以应用在面向备份的容器引用列表中。所示的方法1000开始于在模块1010中请求删除一个文件。在模块1020中,去重复服务器(或其他组件)识别由删除的文件所引用的容器(即,该文件包括存储在该容器中的一个段对象)。由于已经识别了该容器,那么去重复服务器可以确定多少其他文件引用该容器(模块1025)。如果文件的数量大于给定的阈值(条件模块1030),那么去重复服务器可以维护容器引用列表并从容器引用列表中删除已删除文件的标识(模块1035)。条项的删除可以如图5或图10中所描述。在面向文件的容器引用列表的情况中,删除的文件的识别可以从该文件的容器引用列表中移除。在面向备份的容器引用列表的情况中,删除的备份的标识可以从该容器引用列表中移除。

[0070] 如果容器引用列表中一个给定容器的文件数量小于阈值(条件模块1030),那么去重复服务器可以维护段对象引用列表并从段对象引用列表中删除对应于删除的文件的条项(模块1040)。在一个实施例中,当从维护容器引用列表切换到维护段对象引用列表时,对应于识别的容器的段对象引用列表条项可能还未存在。例如,如果在添加文件过程中只有容器引用列表正在维护,那么没有对应的段对象引用列表正在维护。因此,如果对一个给定容器切换到段对象引用列表维护时仍然存在引用该容器的文件,那么该容器的段对象引用列表还未存在。在这种情况下,需要创建该容器的段对象引用列表。在一个实施例中,这些段对象引用列表条项的创建可以在决定维护段对象引用列表的时刻发生(模块104)。接着,去重复服务器可以确定是否该容器是删除的文件所引用的最后一个容器(条件模块1045)。

如果是删除的文件所指向的最后一个容器(条件模块1045),那么该方法可以在模块1055结束。如果不是删除的文件所指向的最后一个容器(条件模块1045),那么该方法可以找到删除的文件所指向的下一个容器(模块1050)。接着,服务器可以返回到模块1025以确定多少其他文件指向下一个容器。

[0071] 应当注意,上述实施例可以包括软件。在这样的实施例中,代表所述系统和/或方法的程序指令和/或数据库(这两者都可以称为“指令”)可以存储在一种计算机可读介质上。通常来讲,计算机可读存储介质可以包括可由计算机在使用过程中访问的任何存储介质以便向计算机提供指令和/或数据。例如,计算机可读存储介质可以包括的存储介质有磁或光介质,例如磁盘(固定的或可移动的)、磁带、CD-ROM、DVD-ROM、CD-R、CD-RW、DVD-R、DVD-RW或蓝光(Blu-Ray)。存储介质可以进一步包括易失性或非易失性存储介质,例如RAM(例如同步动态RAM(SDRAM)、双倍数据速率(DDR、DDR2、DDR3等) SDRAM、低功率DDR(LPDDR2等) SDRAM、Rambus DRAM(RDRAM)、静态RAM(SRAM))、ROM、闪存、可通过如USB接口等这类外围接口访问的非易失性存储器(例如闪存)。存储介质可以包括微电子机械系统(MEMS)以及可通过如网络和/或无线链接这样的通信介质访问的存储介质。

[0072] 在不同的实施例中,本文所述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,根据一个或多个不同模型,可以通过互联网将资源作为服务来提供。这类模型可以包括基础设施即服务(IaaS)、平台即服务(PaaS)、软件即服务(SaaS)。在IaaS中,计算机基础设施作为服务进行传送。在这种情况下,计算设备通常由服务提供商拥有并操作。在PaaS模式中,用于开发软件解决方案的开发者所用的软件工具和底层设备可以由服务提供者作为服务来提供并托管。SaaS典型地包括按需作为服务的服务提供商许可软件。服务提供商可以托管软件,或者可以将软件调配给客户一段给定的时间。以上模式的很多组合是可能的并且是可以设想的。

[0073] 尽管展示并描述了多个方法的若干实施例,但对本领域普通技术人员而言显而易见的是可以对所述方法做出很多改变、修改和变更。因此,改变、修改和变更应当视为在本文描述的方法和机制的范围内。还应当强调的是,上述实施例只是实施例的非限定性示例。

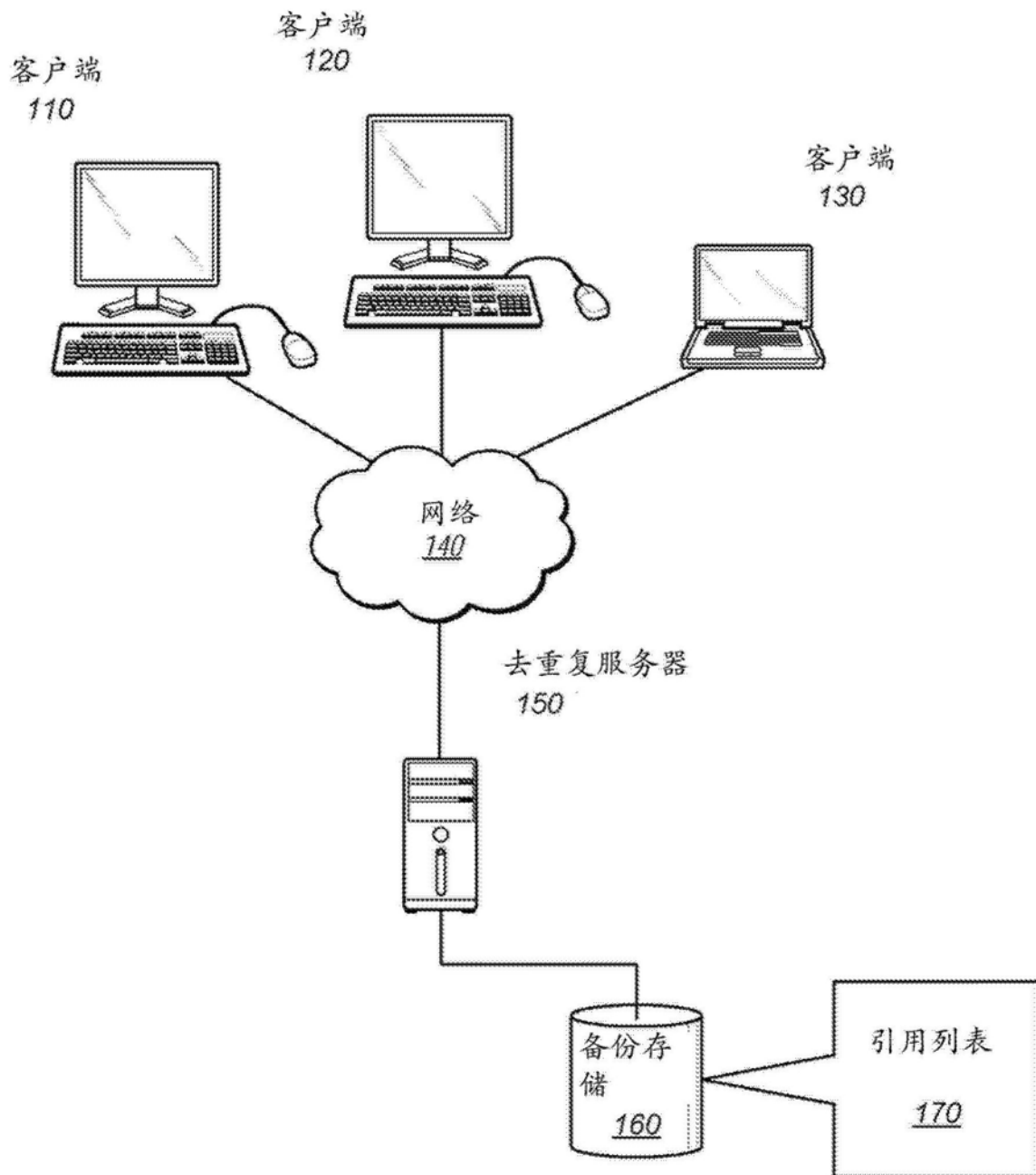
基于去重复的存储系统
100

图1

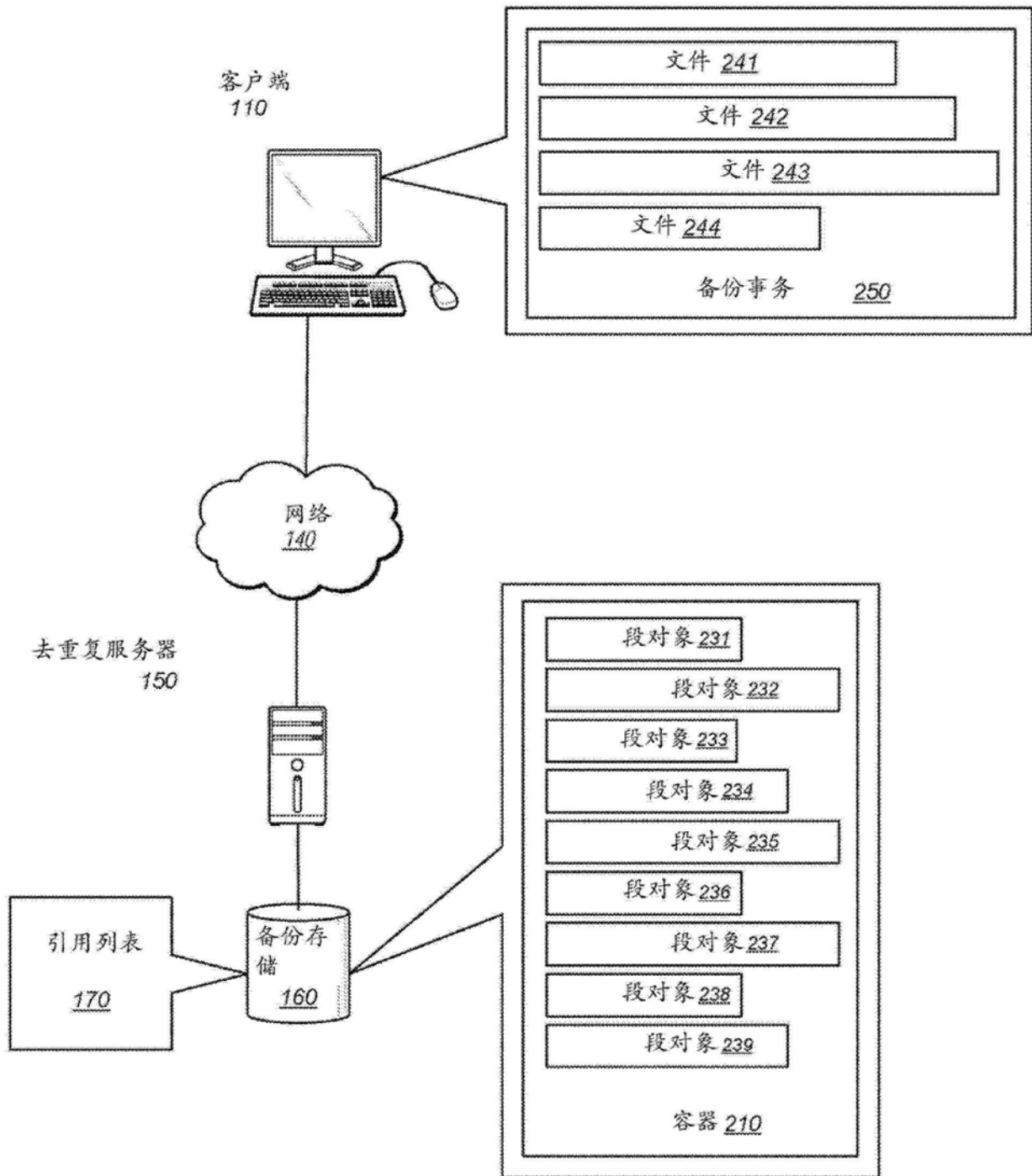


图2

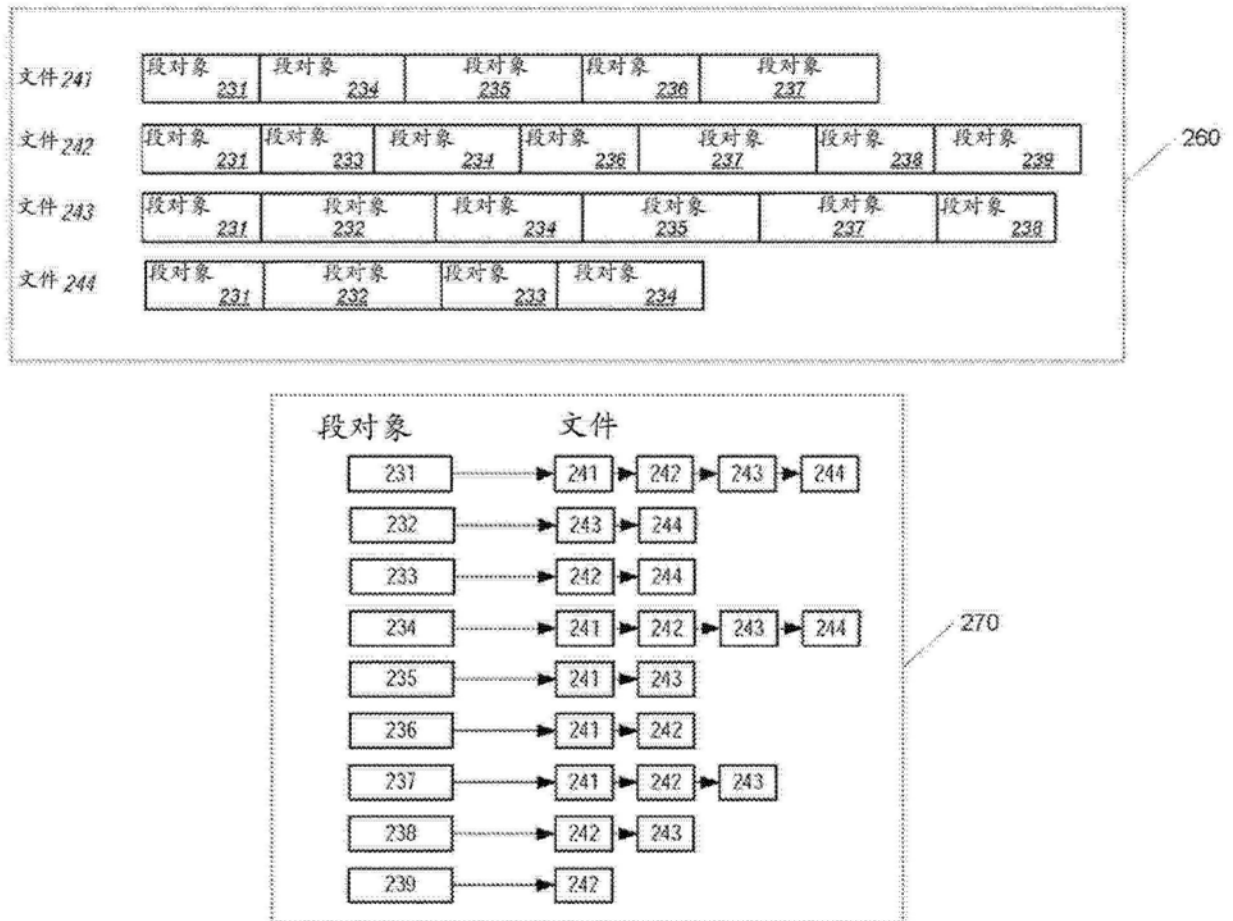


图3

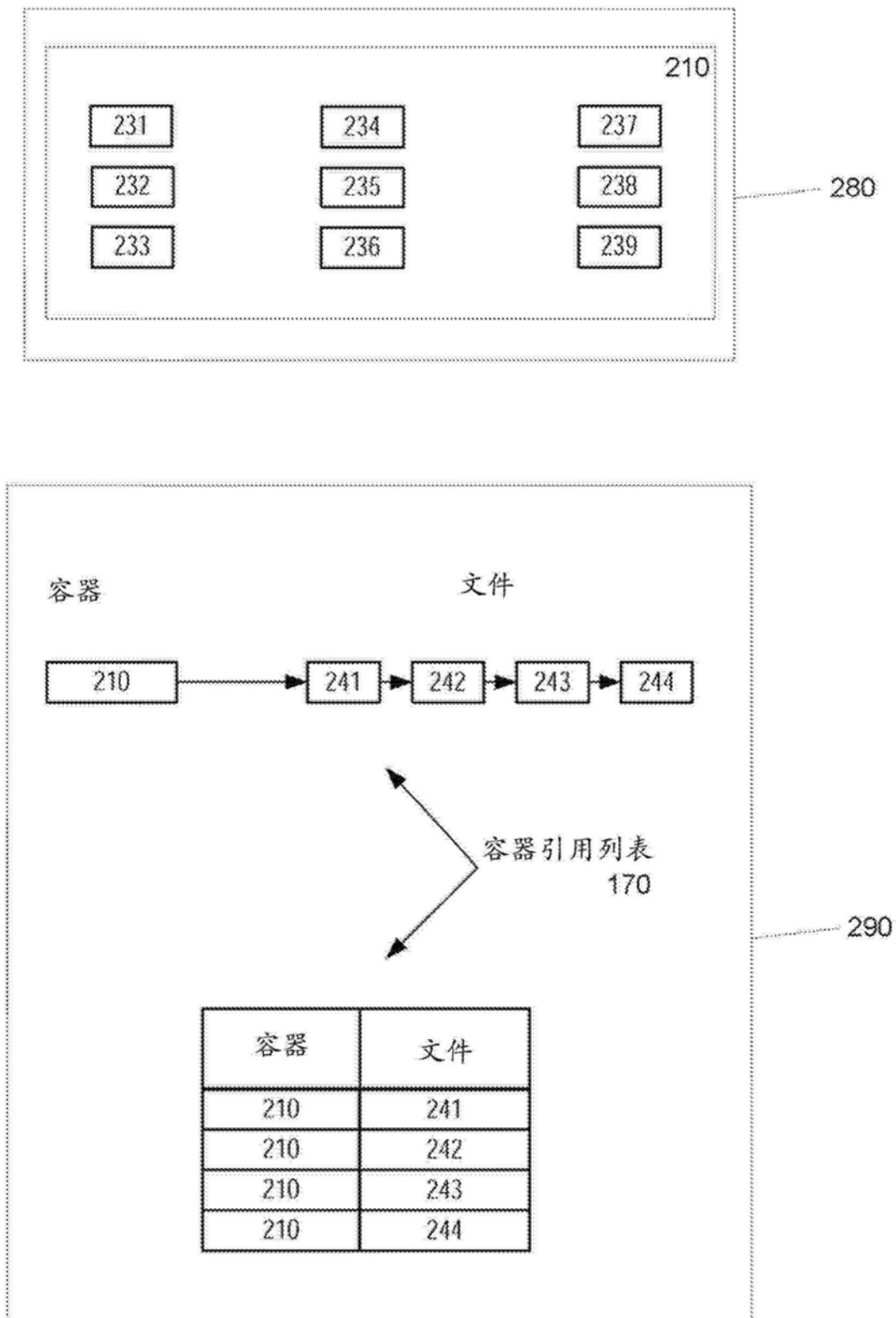


图4

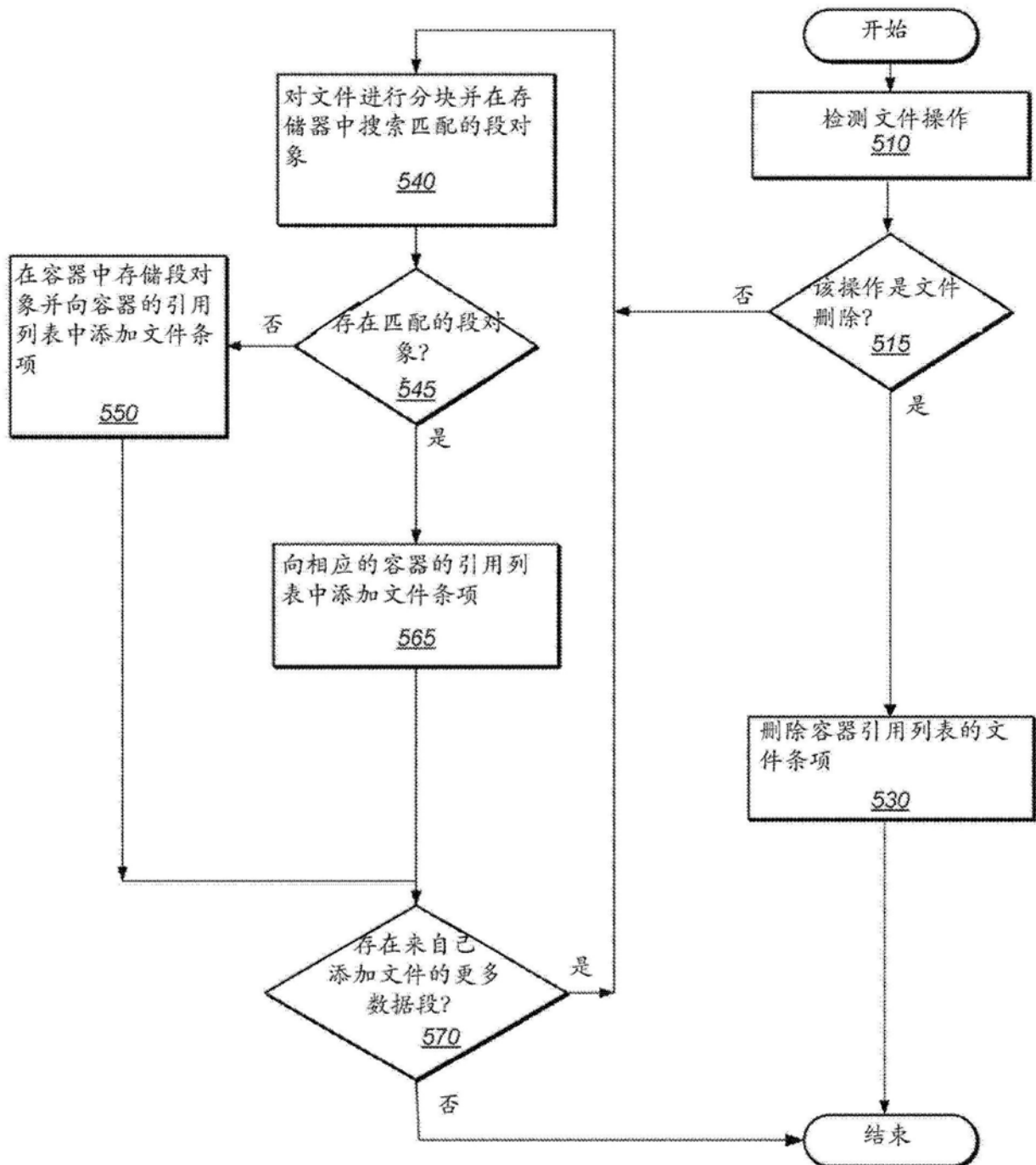


图5

引用列表 500		
容器	段对象	文件
210		241
210		242
210		243
210		244
	231	241
	231	242
	231	243
	231	244
	232	243
	232	244
	233	242
	233	244
	234	241
	234	242
	234	243
	234	244
	235	241
	235	243
	236	241
	236	242
	237	241
	237	242
	237	243
	238	242
	238	243
	239	242

容器

文件

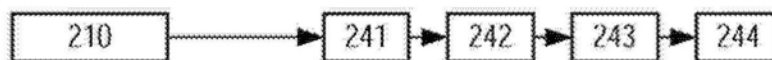


图6

引用列表 500		
容器	段对象	文件
210		241
210		242
210		244
	231	241
	231	242
	231	243
	231	244
	232	243
	232	244
	233	242
	233	244
	234	241
	234	242
	234	243
	234	244
	235	241
	235	243
	236	241
	236	242
	237	241
	237	242
	237	243
	238	242
	238	243
	239	242

容器

文件

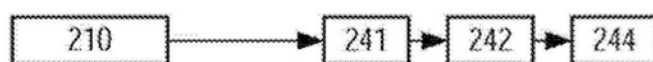


图7

引用列表 500		
容器	段对象	文件
210		241
210		244
	231	241
	231	244
	232	244
	233	244
	234	241
	234	244
	235	241
	236	241
	237	241
	238	None
	239	None

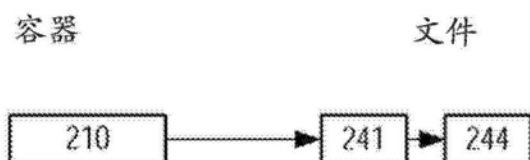


图8

引用列表 800A		
容器	段对象	备份事务
210		250
	231	250
	232	250
	233	250
	234	250
	235	250
	236	250
	237	250
	238	250
	239	250



图9

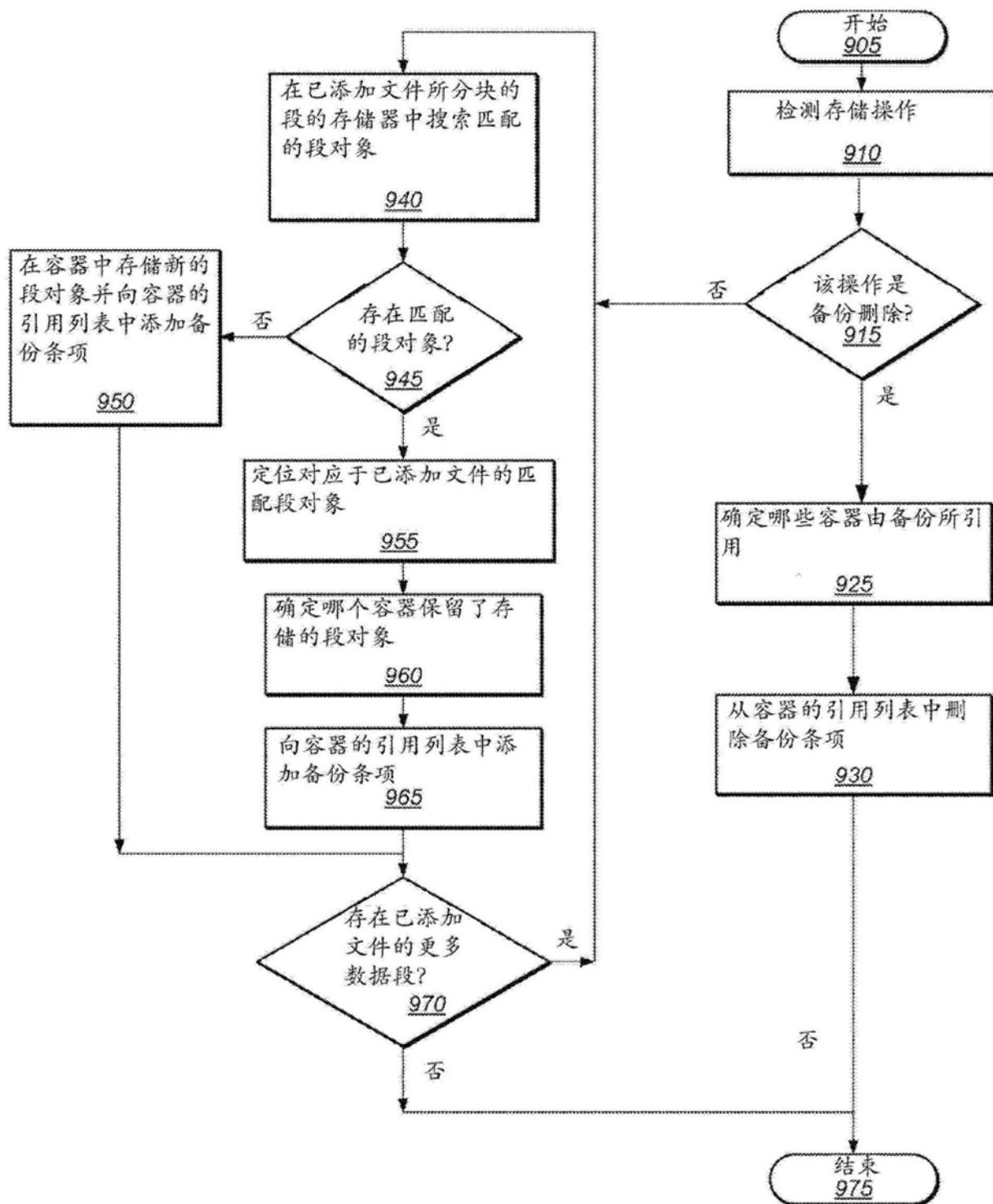


图10

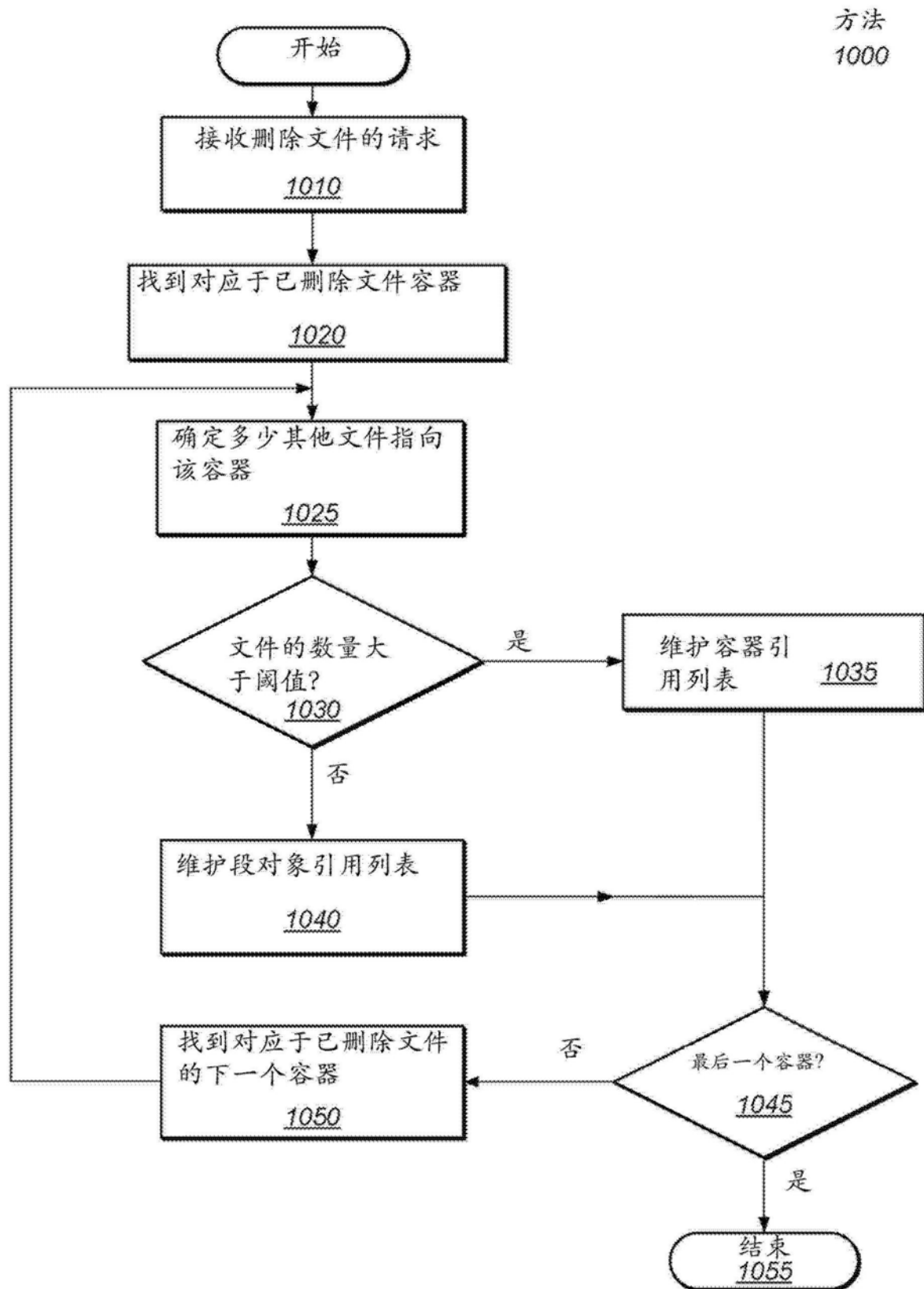


图11