



(12)发明专利

(10)授权公告号 CN 105339904 B

(45)授权公告日 2019.10.11

(21)申请号 201480016823.1

(73)专利权人 辛博立科伊奥公司

(22)申请日 2014.01.31

地址 美国新泽西州

(65)同一申请的已公布的文献号

(72)发明人 B·伊格诺米瑞罗

申请公布号 CN 105339904 A

(74)专利代理机构 北京市铸成律师事务所
11313

(43)申请公布日 2016.02.17

代理人 孟锐

(30)优先权数据

(51)Int.CI.

13/756,921 2013.02.01 US

G06F 12/00(2006.01)

13/797,093 2013.03.12 US

13/908,239 2013.06.03 US

(85)PCT国际申请进入国家阶段日

CN 102282544 A, 2011.12.14,

2015.09.18

US 2002/0172203 A1, 2002.11.21,

(86)PCT国际申请的申请数据

US 2007/0101074 A1, 2007.05.03,

PCT/US2014/014209 2014.01.31

US 2008/0133835 A, 2008.06.05,

(87)PCT国际申请的公布数据

审查员 李江

W02014/121102 EN 2014.08.07

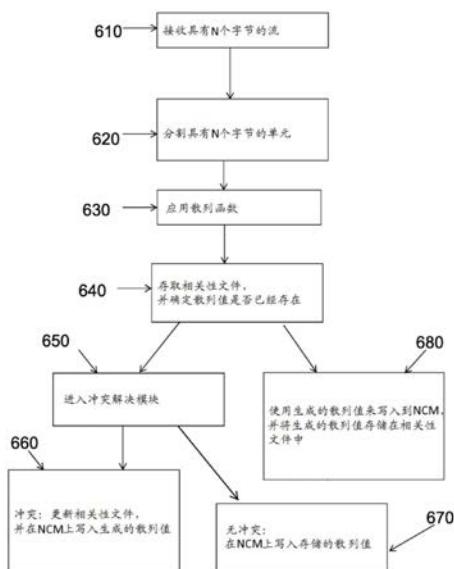
权利要求书4页 说明书31页 附图6页

(54)发明名称

用于存储和检索数据的方法和系统

(57)摘要

通过使用本发明的技术，相关人员能够有效地存储和检索数据。在各种实施方案中，相关人员可以通过编码所述数据以及存储大小上小于起初数据的编码数据，而实现这些效率。



1. 一种用于将数据存储在非高速缓存器记录介质上的方法,所述方法包括:
接收具有N个字节的I/O流;
将所述N个字节分割成具有X个字节的片段单元;
将加密散列函数应用到具有X个字节的每个片段单元,以便针对具有X个字节的每个片段单元来形成生成的散列函数值;
存取相关性文件,其中所述相关性文件使所存储的具有Y个比特的散列函数值与多个X字节存储序列中的每个序列关联,并且对于每个片段单元而言:
如果针对具有X个字节的片段单元的所述生成的散列函数值是处于所述相关性文件中,那么便使用所述所存储的具有Y个比特的散列函数值来在非高速缓存器记录介质上进行存储;以及
如果针对所述具有X个字节的片段单元的所述生成的散列函数值并未处于所述相关性文件中,那么便将所述具有Y个比特的生成的散列函数值与所述具有X个字节的片段单元一起存储在所述相关性文件中,并使用所述生成的散列函数值来在所述非高速缓存器记录介质上进行存储;以及
通过使用比特标记符表来编码用于存储的每个散列函数值,以便针对用于存储的每个散列函数值而生成比特标记符,并且,将每个比特标记符存储在所述非高速缓存器记录介质中。
2. 根据权利要求1所述的方法,其中所述加密散列函数包括MD5消息摘要算法。
3. 根据权利要求1所述的方法,其还包括应用冲突解决模块,其中,如果所述冲突解决模块确定对于具有X个字节的片段单元而言,生成与所述相关性文件中的所有存储的散列函数值相同的散列函数值,但是所述具有X个字节的片段单元不同于与所述所有存储的散列函数值关联的所述X个字节,那么便将不同的Z个比特与所述所有存储的散列函数值以及所生成的散列函数值相关联。
4. 根据权利要求3所述的方法,其中在所述相关性文件中,对于不存在冲突的所有散列函数值而言,关联Z个比特,并且所述Z个比特为8至16个0。
5. 根据权利要求4所述的方法,其中N=4096并且X=512。
6. 根据权利要求5所述的方法,其中Y=128或256。
7. 根据权利要求1所述的方法,其还包括将多个散列函数值存储在所述非高速缓存器记录介质上。
8. 根据权利要求1所述的方法,其还包括通过使用频次转换器来编码用于存储的每个散列函数值,其中,针对用于存储的每个散列函数值而言,生成转换比特串,其中,对于至少两个不同的Y比特串而言,输出的转换比特串具有不同长度。
9. 根据权利要求8所述的方法,其中,对于具有不同大小的每批多个转换比特串而言,存在长度为A个比特的第一转换比特串以及长度为B个比特的第二转换比特串,其中A<B,并且所述第一转换比特串中的所述A个比特的身份与所述第二转换比特串中的前A个比特的身份并不相同。
10. 根据权利要求3所述的方法,其还包括通过使用频次转换器来编码用于存储的每个散列函数值,其中,针对用于存储的每个散列函数值而言,生成转换比特串,其中,对于至少两个不同的Y比特串而言,输出的转换比特串具有不同长度。

11. 根据权利要求10所述的方法,其中所述I/O是从主机接收的,并且所述方法还包括向所述主机确认所述I/O流的接收。

12. 根据权利要求11所述的方法,其中所述主机将所述I/O记录为存储在第一存储地址上,并且所述转换比特串是存储在所述非高速缓存器记录介质中的第二存储地址上,其中所述第一存储地址与所述第二存储地址并不相同。

13. 根据权利要求12所述的方法,其中所述第一存储地址对应于具有第一文件大小的文件,并且所述第二存储地址对应于具有第二文件大小的文件,其中所述第一文件大小是所述第二文件大小的至少四倍大。

14. 根据权利要求12所述的方法,其还包括将所述第二存储地址存储在中介器上,其中所述中介器为非暂时性存储介质。

15. 根据权利要求14所述的方法,其中所述第二存储地址存储在比特字段中。

16. 根据权利要求15所述的方法,其中所述中介器包括:

第一组磁道,其中所述第一组磁道包括文件系统信息、可启动性信息和分区信息;

第二组磁道,其中所述第二组磁道包括所述第一组磁道的复本;

第三组磁道,其中所述第三组磁道包括不同于文件系统信息、可启动性信息和分区信息的元数据;以及

第四组磁道,其中所述第四组磁道包括所述比特字段。

17. 一种用于将数据存储在非高速缓存器记录介质上的方法,所述方法包括:

接收具有N个字节的I/O流;

将所述N个字节分割成具有X个字节的片段单元;

使加密散列函数值与具有X个字节的每个片段单元关联,其中所述关联包括存取相关性文件,其中所述相关性文件使所存储的具有Y个比特的散列函数值与多个X字节存储序列中的每个序列关联,并且

如果所述具有X个字节的片段单元的序列是处于所述相关性文件中,那么便使用所述所存储的具有Y个比特的散列函数值来在非高速缓存器记录介质上进行存储;以及

如果所述具有X个字节的片段单元的序列并未处于所述相关性文件中,那么便将具有Y个比特的新散列函数值与所述具有X个字节的片段单元一起存储在所述相关性文件中,并使用所述具有Y个比特的新散列函数值来在所述非高速缓存器记录介质上进行存储;以及

通过使用比特标记符表来编码用于存储的每个散列函数值,以便针对用于存储的每个散列函数值而生成比特标记符,并且,将每个比特标记符存储在所述非高速缓存器记录介质中。

18. 一种用于将数据存储在非高速缓存器记录介质上的方法,所述方法包括:

接收具有N个字节的I/O流;

将加密散列函数值算法应用到具有N个字节的每个单元,以便针对具有N个字节的每个单元来形成生成的散列函数值;

存取相关性文件,其中所述相关性文件使所存储的具有Y个比特的散列函数值与多个N字节存储序列中的每个序列关联,并且

如果针对所述具有N个字节的单元的所述生成的散列函数值是处于所述相关性文件中,那么便使用所述所存储的具有Y个比特的散列函数值来在非高速缓存器记录介质上进

行存储;以及

如果针对所述具有N个字节的单元的所述生成的散列函数值并未处于所述相关性文件中,那么便将所述具有Y个比特的生成的散列函数值与所述具有N个字节的单元一起存储在所述相关性文件中,并使用所述生成的散列函数值来在所述非高速缓存器记录介质上进行存储;以及

通过使用比特标记符表来编码用于存储的每个散列函数值,以便针对用于存储的每个散列函数值而生成比特标记符,并且,将每个比特标记符存储在所述非高速缓存器记录介质中。

19.一种用于将数据存储在记录介质上的方法,所述方法包括:

接收多个数字二进制信号,其中所述数字二进制信号是组织在多个小盘中,其中每个小盘的长度为N个比特,其中N是大于1的整数并且其中所述小盘具有某个顺序;

将每个小盘划分为具有统一大小的子单元,并从一组X个标记符中指配标记符给每个子单元,以便形成一组多个标记符来编码所述子单元,其中X小于或等于子单元内不同的比特组合的数目,同样的子单元指配相同的标记符,并且至少一个标记符小于子单元的所述大小,其中,所述指配包括存取比特标记符表,其中在所述比特标记符表内,每个独特标记符被识别为对应于独特比特串;以及

将所述组的所述多个标记符以与所述小盘的所述顺序相对应的顺序,存储在非暂时性记录介质上。

20.根据权利要求19所述的方法,其中每个子单元具有第一末端和第二末端,并且在指配所述标记符之前,所述方法还包括分析每个小盘的每个子单元内的一个或多个比特,以便确定所述第二末端处的比特是否具有数值0,并且,如果所述第二末端处的所述比特具有数值0,那么便移除所述第二末端处的所述比特以及具有所述数值0且与所述第二末端处的所述比特形成连续比特串的所有比特,从而针对在所述第二末端处具有0的任何子单元来形成修正子单元。

21.根据权利要求20所述的方法,其中计算机算法:

(a)核查每个子单元以便确定在所述第二末端处是否存在0,且如果存在,便移除所述0,以便形成具有在邻近于所述子单元的所述第二末端的位置处的修正的第二末端的修正子单元;

(b)核查每个修正子单元以便确定在所述修正的第二末端处是否存在0,且如果存在,便移除所述0,以便形成另外的修正第二末端;以及

(c)针对每个修正子单元来重复(b),直到生成在其第二末端处具有1的缩短子单元。

22.根据权利要求19所述的方法,其中每个子单元具有第一末端和第二末端,并且在指配所述标记符之前,所述方法还包括分析每个小盘的每个子单元内的一个或多个比特,以便确定所述第二末端处的所述比特是否具有数值1,并且,如果所述第二末端处的所述比特具有数值1,那么便移除所述第二末端处的所述比特以及具有所述数值1且与所述第二末端处的所述比特形成连续比特串的所有比特,从而针对在所述第二末端处具有1的任何子单元来形成修正子单元。

23.根据权利要求22所述的方法,其中计算机算法:

(a)核查每个子单元以便确定在所述第二末端处是否存在1,且如果存在,便移除所述

1,以便形成具有在邻近于所述子单元的所述第二末端的位置处的修正的第二末端的修正子单元;

(b) 核查每个修正子单元以便确定在所述修正的第二末端处是否存在1,且如果存在,便移除所述1,以便形成另外的修正第二末端;以及

(c) 针对每个修正子单元来重复(b),直到生成在其第二末端处具有0的缩短子单元。

24. 根据权利要求19所述的方法,其中所述标记符存储在频次转换器中,所述标记符具有多个不同大小,并且大小上更小的标记符与更高频次子单元相互关联。

25. 根据权利要求19所述的方法,其中多个不同标记符是由不同数目的比特形成。

26. 一种用于存储电子数据的方法,所述方法包括:

接收一组参数,其中所述参数包括文件系统信息、可启动性信息和分区信息中的一个或多个;

接收元数据;

接收一个或多个文件,其中每个文件具有文件名;

将所述参数和元数据存储在中介器上;

将所述文件中的每个文件存储在非高速缓存器介质上的某个位置中;

在所述中介器上,存储每个文件名与所述非高速缓存器介质上的位置的相关性;以及

在将所述文件存储在所述非高速缓存器介质上之前,编码所述文件,其中所述编码包括使用比特标记符表来创建转换文件。

27. 根据权利要求26所述的方法,其中所述编码还包括使用频次转换器来创建转换文件。

28. 根据权利要求26所述的方法,其中所述转换文件并不含有文件系统信息、可启动性信息或分区信息中的任何信息。

29. 根据权利要求26所述的方法,其中所述参数存储在多个预留磁道中。

30. 根据权利要求26所述的方法,其中所述多个预留磁道为第一组预留磁道,并且所述方法还包括将所述参数复制到第二组预留磁道中。

31. 根据权利要求30所述的方法,其还包括使用所述第二组预留磁道来检查所述第一组预留磁道中的错误。

32. 根据权利要求31所述的方法,其中所述元数据与用于自动精简配置的指令相对应。

33. 根据权利要求26所述的方法,其中所述文件是从主机接收的,所述主机将所述文件记录为存储在虚拟地址上,并且所述虚拟地址与所述文件的所述位置并不相同。

34. 一种用于备份数据的方法,所述方法包括根据权利要求26所述的方法,其中所述中介器为第一中介器且所述位置为第一位置,所述方法还包括生成第二中介器,其中所述第二中介器为所述第一中介器在时间T1上的复本,并且,在T1之后的时间T2上:

接收指令来保存位于所述第一位置处的修正文件;

将新文件存储在第二位置处,其中所述新文件对应于所述修正文件;以及

更新所述第一中介器以便使所述文件名与所述第二位置相互关联,其中在所述第二中介器上,所述文件名与所述第一位置相互关联。

用于存储和检索数据的方法和系统

发明领域

[0001] 本发明涉及数据存储和检索领域。

[0002] 发明背景

[0003] 二十一世纪已经见证了人们和公司生成和存储的数字化信息的量成指数级增长。这类信息由通常存储在磁性表面(如磁盘)上的电子数据组成,所述磁性表面含有尺寸为亚微米并且能够存储数条单独二进制信息的较小区域。

[0004] 由于许多实体生成的大量数据,因此数据存储行业已经转向基于网络的存储系统。这些类型的存储系统可以包括形成处理系统或作为处理系统一部分的至少一个存储服务器,所述处理系统被配置成代表一个或多个实体来存储和检索数据。数据可以作为存储对象(如区块和/或文件)进行存储和检索。

[0005] 用于存储的一个系统是网络附加存储(NAS)系统。在NAS的情境中,存储服务器代表一个或多个客户端进行运作来存储数据和管理数据的文件级存取。文件可以存储在存储系统中,所述存储系统包括海量存储装置(如磁盘或光盘或磁带)的一个或多个阵列。另外,这个数据存储模型可以使用独立磁盘冗余阵列(RAID)技术。

[0006] 用于存储的另一系统是存储区域网络(SAN)。在SAN系统中,通常存储服务器向客户端提供所存储数据的区块级存取,而不是所存储数据的文件级存取。然而,一些存储服务器能够向客户端提供文件级存取和区块级存取。

[0007] 无论相关人员使用NAS还是SAN,所有生成数据的行业都必须考虑存储和检索所述数据的成本。因此,需要新的技术来节约性地存储和检索数据。

发明内容

[0008] 本发明提供用于改善数据存储和检索效率的方法、系统以及计算机程序产品。通过使用本发明的各种实施方案,相关人员可以有效地存储和存取已经可选地转换或编码的数据。另外或替代地,本发明的各种实施方案使用会促进数据有效存储和数据有效存取的中介器(mediator)。

[0009] 本发明的各种实施方案是针对原始数据和/或将元数据与原始数据分离。因此,基于可以结合本发明进行存储和/或检索的文件类型而言,不存在任何限制。可以使用的文件类型的示例包括但不限于JPEG、PDF、WORD文档、MPEG以及TXT文档。借助本发明的各种实施方案,相关人员可以转变数据和/或更改转变后或转换后数据所存储的物理装置。这些实施方案可以借助使用计算机的自动化过程来执行,所述计算机包括或者可操作性地耦接到计算机程序产品,所述计算机程序产品在运行时会使得执行本发明的方法或过程的步骤。这些方法或过程可以(例如)体现在计算机算法或脚本中或包括所述计算机算法或脚本,且可选地由本发明的系统加以执行。

[0010] 根据第一实施方案,本发明针对一种用于将数据存储在记录介质上的方法,所述方法包括:(i)接收多个数字二进制信号,其中所述数字二进制信号是组织在多个小盘(chunklet)中,其中每个小盘的长度为N个比特,其中N是大于1的整数并且其中所述小盘具

有某个顺序; (ii) 将每个小盘划分为具有统一大小的子单元，并从一组X个标记符中指配标记符给每个子单元，以便形成一组多个标记符，其中X等于子单元内不同的比特组合的数目，同样的子单元指配相同的标记符，并且至少一个标记符小于子单元的大小；以及 (iii) 将所述组的所述多个标记符以与所述小盘的所述顺序相对应的顺序，或者以允许重新创建所述小盘的所述顺序的另一配置而存储在非暂时性记录介质上。

[0011] 根据第二实施方案，本发明针对一种用于从记录介质检索数据的方法，所述方法包括：(i) 访问记录介质，其中所述记录介质以某个顺序存储多个标记符；(ii) 将所述多个标记符转译成一组小盘，其中每个小盘的长度为N个比特，其中N是大于1的整数，并且其中所述小盘具有与所述多个标记符的顺序相对应的顺序，而且其中所述转译是通过存取比特标记符表来完成，其中在所述比特标记符表内，每个独特标记符识别为对应于独特比特串；以及 (iii) 生成包括所述组小盘的输出。所述标记符可以按照或可以不按照与所述小盘的顺序相同的顺序来存储，但是无论它们存储的顺序如何，相关人员都可以重新创建所述小盘的顺序以及从中获得这些标记符的任何文件。

[0012] 根据第三实施方案，本发明针对一种用于将数据存储在记录介质上的方法，所述方法包括：(i) 接收多个数字二进制信号，其中所述数字二进制信号组织在小盘中，其中每个小盘的长度为N个比特，每个小盘具有第一末端和第二末端，N是大于1的整数，并且所述小盘具有某个顺序；(ii) 将每个小盘划分为多个子单元，其中每个子单元的长度为A个比特；(iii) 分析每个子单元以便确定所述第二末端处的比特是否具有数值0，并且，如果所述第二末端处的所述比特具有数值0，那么便移除所述第二末端处的所述比特以及具有所述数值0且与所述第二末端处的所述比特形成连续比特串的所有比特，从而针对在所述第二末端处具有0的任何小盘来形成修正小盘；以及 (iv) 在非暂时性记录介质上，以允许按照所述顺序重新构建所述小盘的方式，存储每个修正子单元以及长度为A个比特且在其第二末端处具有1的每个子单元。举例而言，所述修正子单元（以及任何未修正的子单元）可以在修正之前，按照与每个小盘内的子单元的顺序相对应的顺序进行组织，并且所述数据的存储可以与原始文件内的对应小盘处于相同的顺序中。

[0013] 根据第四实施方案，本发明提供一种用于将数据存储在记录介质上的方法，所述方法包括：(i) 接收多个数字二进制信号，其中所述数字二进制信号组织在小盘中，其中每个小盘的长度为N个比特，每个小盘具有第一末端和第二末端，N是大于1的整数，并且所述小盘具有某个顺序；(ii) 分析每个小盘以便确定所述第一末端处的比特是否具有数值0，并且，如果所述第一末端处的所述比特具有数值0，那么便移除所述第一末端处的所述比特以及具有所述数值0且与所述第一末端处的所述比特形成连续比特串的所有比特，从而针对在所述第一末端处具有0的任何小盘来形成第一修正小盘；(iii) 分析每个小盘以便确定所述第二末端处的比特是否具有数值0，并且，如果所述第二末端处的所述比特具有数值0，那么便移除所述第二末端处的所述比特以及具有所述数值0且与所述第二末端处的所述比特形成连续比特串的所有比特，从而针对在所述第二末端处具有0的任何小盘来形成第二修正小盘；(iv) 对于每个小盘而言：(a) 如果所述第一修正小盘和所述第二修正小盘的大小是相同的，那么便存储所述第一修正小盘或所述第二修正小盘，(b) 如果所述第一修正小盘小于所述第二修正小盘，那么便存储所述第一修正小盘，(c) 如果所述第二修正小盘小于所述第一修正小盘，那么便存储所述第二修正小盘，(d) 如果不存在修正小盘，那么便存储所述

小盘，(e) 如果不存在第一修正小盘，但是存在第二修正小盘，那么便存储所述第二修正小盘，(f) 如果不存在第二修正小盘，但是存在第一修正小盘，那么便存储所述第一修正小盘，其中所存储的每个修正小盘是与指示是否从所述第一末端或所述第二末端移除了一个或多个比特的信息一起存储。指示是否从所述第一末端或所述第二末端移除了一个或多个比特的所述信息可以(例如)呈所述子单元的独特性的形式。当生成有待于在任何比较方式中进行比较的所述第一与第二修正小盘中的每个小盘时，优选的是0已经从任一末端移除，而不是从两个末端移除。

[0014] 根据第五实施方案，本发明提供一种用于将数据存储在记录介质上的方法，所述方法包括：(i) 接收多个数字二进制信号，其中所述数字二进制信号组织在小盘中，其中每个小盘的长度为N个比特，每个小盘具有第一末端和第二末端，N是大于1的整数，并且所述小盘具有某个顺序；(ii) 将每个小盘划分为多个子单元，其中每个子单元的长度为A个比特；(iii) 分析每个子单元以便确定所述第一末端处的比特是否具有数值0，并且，如果所述第一末端处的所述比特具有数值0，那么便移除所述第一末端处的所述比特以及具有所述数值0且与所述第一末端处的所述比特形成连续比特串的所有比特，从而针对在所述第一末端处具有0的任何子单元来形成第一修正子单元；(iv) 分析每个子单元以便确定所述第二末端处的比特是否具有数值0，并且，如果所述第二末端处的所述比特具有数值0，那么便移除所述第二末端处的所述比特以及具有所述数值0且与所述第二末端处的所述比特形成连续比特串的所有比特，从而针对在所述第二末端处具有0的任何子单元来形成第二修正子单元；以及(v) 对于每个子单元而言：(a) 如果所述第一修正子单元和所述第二修正子单元的大小是相同的，那么便存储所述第一修正子单元或所述第二修正子单元，(b) 如果所述第一修正子单元小于所述第二修正子单元，那么便存储所述第一修正子单元，(c) 如果所述第二修正子单元小于所述第一修正子单元，那么便存储所述第二修正子单元，(d) 如果不存在修正子单元，那么便存储所述子单元，(e) 如果不存在第一修正子单元，但是存在第二修正子单元，那么便存储所述第二修正子单元，(f) 如果不存在第二修正子单元，但是存在第一修正子单元，那么便存储所述第一修正子单元，其中所存储的每个修正子单元是与指示是否从所述第一末端或所述第二末端移除了一个或多个比特的信息一起存储。指示是否从所述第一末端或所述第二末端移除了一个或多个比特的所述信息可以(例如)呈所述子单元的独特性的形式。

[0015] 根据第六实施方案，本发明提供一种用于从记录介质检索数据的方法，所述方法包括：(i) 访问记录介质，其中所述记录介质在多个位置上存储多个数据单元，其中每个数据单元含有多个比特且所述数据单元的最大大小为N个比特，至少一个数据单元含有少于N个比特，并且所述数据单元具有某个顺序；(ii) 检索所述数据单元，并在长度少于N个比特的任何数据单元的末端处添加一个或多个比特，以便生成一组对应于所述数据单元的小盘，其中每个小盘含有相同数目的比特；以及(iii) 以与所述数据单元的所述顺序相对应的顺序来生成包括所述组小盘的输出。

[0016] 根据第七实施方案，本发明针对一种用于存储电子数据的方法，所述方法包括：(i) 接收一组参数，其中所述参数包括文件系统信息、可启动性信息和分区信息中的一个或多个；(ii) 接收元数据；(iii) 接收一个或多个文件，其中每个文件具有文件名；(iv) 将所述参数和元数据存储在中介器上；(v) 将所述文件中的每个文件存储在非高速缓存器介质上

的某个位置中；以及 (vi) 在中介器上，存储每个文件名与所述非高速缓存器介质上的位置的相关性。本发明的中介器可以服务于下述目的中的一个或多个目的：(1) 存储用于数据编码的协议；(2) 在记录介质上分配物理空间；(3) 充当主机发起器的磁盘几何排列的中心点；(4) 增加安全性；(5) 允许系统内部构件来登记、读取一个或两个预留 (R_1 和 R_2) 并与其进行互动；(6) 提供框架以便采用新的方式来获取快照（也就是，冻结特定时间上所存储的数据）和/或克隆磁盘；以及 (7) 提供元数据。即使没有实现所有这些特征，但只要实现其中的一个或多个特征，也可以有助于提升用于存储数据、从未授权存取中保护数据和/或检索数据的方法的效率。

[0017] 根据第八实施方案，本发明针对一种用于备份数据的方法，所述方法包括：(i) 在第一中介器上，使多个文件名与数据文件的多个位置相互关联，其中所述数据文件的所述位置对应于第一非高速缓存器介质上的位置，并且所述第一中介器被配置成允许识别特定文件名的用户来检索对应于所述特定文件名的数据文件；(ii) 将所述多个数据文件复制到第二非高速缓存器介质；(iii) 生成第二中介器，其中所述第二中介器是所述第一中介器在时间 T_1 上的复本；并且在所述第二中介器内，所述第二非高速缓存器介质上多个数据文件的位置与所述文件名相互关联；(iv) 接收指令来保存数据文件的修正；以及 (v) 在 T_1 之后的时间 T_2 上，在所述第一非高速缓存器介质上保存所述数据文件的所述修正。优选地，所述修正并非保存在所述第二非高速缓存器介质上的对应数据文件中。

[0018] 根据第九实施方案，本发明提供一种数据存储和检索系统，所述系统包括：(i) 非高速缓存器数据存储介质；(ii) 中介器，其中所述中介器从所述非高速缓存器数据存储介质远程存储，并且所述中介器包括：(a) 第一组磁道；(b) 第二组磁道；(c) 第三组磁道；和 (d) 第四组磁道；以及 (ii) 管理器，其中所述管理器被配置来：(a) 将包括文件系统信息、可启动性信息和分区信息中的一个或多个的数据存储在所述第一组磁道中；(b) 将元数据存储在所述第三组磁道中；(c) 将一个或多个文件存储在所述非高速缓存器介质上，其中所述一个或多个文件存储在不含有文件系统信息、可启动性信息和分区信息中的任何信息的所述非高速缓存器介质上；(d) 将所述非高速缓存器介质中每个文件的位置存储在所述第四组磁道中；以及 (e) 存储所述非高速缓存器介质中每个文件的位置与文件的主机名的相关性。因此，所述管理器可以使得将信息存储在所述中介器上，所述信息可以不存储在所述管理器本身上。

[0019] 根据第十实施方案，本发明针对一种用于将数据存储在非高速缓存器记录介质上的方法，所述方法包括：(i) 接收具有 N 个字节的 I/O 流（通过使用（例如）I/O 协议）；(ii) 将所述 N 个字节分割成具有 X 个字节的片段单元；(iii) 将加密散列函数（其为数值算法）应用到具有 X 个字节的每个片段单元，以便针对具有 X 个字节的每个片段单元来形成生成的散列函数值；(iv) 存取相关性文件，其中所述相关性文件使所存储的具有 Y 个比特的散列函数值与多个 X 字节存储序列中的每个序列关联，并且：(a) 如果针对具有 X 个字节的片段单元的所述生成的散列函数值是处于所述相关性文件中，那么便使用所存储的具有 Y 个比特的散列函数值来在非高速缓存器记录介质上进行存储；以及 (b) 如果针对所述具有 X 个字节的片段单元的所述生成的散列函数值并未处于所述相关性文件中，那么便将所述具有 Y 个比特的生成的散列函数值与所述具有 X 个字节的片段单元一起存储在所述相关性文件中，并使用所述生成的散列函数值来在所述非高速缓存器记录介质上进行存储。

[0020] 根据第十一实施方案，本发明针对一种用于将数据存储在非高速缓存器记录介质上的方法，所述方法包括：(i) 接收具有N个字节的I/O流；(ii) 将所述N个字节分割成具有X个字节的片段单元；(iii) 使加密散列函数值与具有X个字节的每个片段单元关联，其中所述关联包括存取相关性文件，其中所述相关性文件使所存储的具有Y个比特的散列函数值与多个X字节存储序列中的每个序列关联，并且：(a) 如果具有X个字节的片段单元的序列是处于所述相关性文件中，那么便使用所存储的具有Y个比特的散列函数值来在非高速缓存器记录介质上进行存储；以及(b) 如果具有X个字节的片段单元的序列并未处于所述相关性文件中，那么便将具有Y个比特的新生成的散列函数值与所述具有X个字节的片段单元一起存储在所述相关性文件中，并使用所述具有Y个比特的新生成的散列函数值来在所述非高速缓存器记录介质上进行存储。

[0021] 根据第十二实施方案，本发明针对一种用于将数据存储在非高速缓存器记录介质上的方法，所述方法包括：(i) 接收具有N个字节的I/O流；(ii) 将加密散列函数应用到具有N个字节的每个单元，以便针对具有N个字节的每个单元来形成生成的散列函数值；(iii) 存取相关性文件，其中所述相关性文件使所存储的具有Y个比特的散列函数值与多个N字节存储序列中的每个序列关联，并且：(a) 如果针对所述具有N个字节的单元的所述生成的散列函数值是处于所述相关性文件中，那么便使用所存储的具有Y个比特的散列函数值来在非高速缓存器记录介质上进行存储；以及(b) 如果针对所述具有N个字节的单元的所述生成的散列函数值并未处于所述相关性文件中，那么便将所述具有Y个比特的生成的散列函数值与所述具有N个字节的单元一起存储在所述相关性文件中，并使用所述生成的散列函数值来在所述非高速缓存器记录介质上进行存储。

[0022] 本发明的各种方法可以(例如)与下述方法结合使用以便配置存储系统来存储电子数据：(i) 接收一组参数，其中所述参数包括文件系统信息、可启动性信息和分区信息中的一个或多个(即便不包括所有这些)；(ii) 接收元数据；以及(iii) 将所述参数和元数据存储在中介器上。在配置所述存储系统之后，所述系统可以准备：接收一个或多个文件，其中每个文件都具有文件名；将所述文件中的每个文件(可选的是借助上述方法中的一种方法进行处理，以便进行转译)存储在非高速缓存器介质上的位置中；以及，将每个文件名与所述非高速缓存器介质上的位置的相关性存储在所述中介器上。

[0023] 当使用本发明的某些方法时，指令可以存储在编码于非暂时性计算机可读介质上的计算机程序产品上，所述指令可以经过操作而致使数据处理设备来执行操作，所述操作包括：(a) 借助I/O协议而从服务器接收具有N个字节的I/O流；(b) 针对具有N个字节的每个单元或者针对多个具有N个字节的片段单元来获取散列函数值；以及(c) 致使存储包括多个散列值或多个转换散列值的数据。通过将所述散列值转换成转换散列值，便可以针对存储文件内信息的未授权存取来提供加强保护。可选地，所述计算机程序产品还包括冲突解决模块，所述冲突解决模块允许识别指配给具有N个字节的不同单元或具有N个字节的不同片段单元的相同散列函数值，并且致使存储对应于所述冲突散列函数值中的每个值的不同数据。当被运行时，所述计算机程序产品可以致使上述特征的启用和自动运行。这些指令可以存储在一个模块中，或者存储在可操作性地彼此耦接的多个单独模块中。

[0024] 另外，本发明的各种实施方案也可以实施在系统上。本发明的系统的示例包括：(a) 非高速缓存器存储介质(其可以是永久性或非暂时性存储装置)；以及(b) 一个或多个处

理器,其可经过操作而与所述非高速缓存器存储介质进行互动。可选地,所述系统还包括一个或多个用户界面(例如,图形用户界面),所述用户界面能够允许用户与所述一个或多个处理器进行互动。所述一个或多个处理器还可经过操作而执行本发明的方法中的一个或多个方法,所述方法可以(例如)存储在计算机程序产品上,所述计算机程序产品存储在非暂时性介质上。

[0025] 在一些实施方案中,所述系统包括:(i)中介器,其中所述中介器是从所述非高速缓存器数据存储介质远程存储,并且所述中介器包括:(a)第一组磁道;(b)第二组磁道;(c)第三组磁道;和(d)第四组磁道;(ii)非高速缓存器介质;以及(iii)管理器,其中所述管理器被配置来:(a)致使将包括文件系统信息、可启动性信息和分区信息中的一个或多个的数据存储在所述第一组磁道中;(b)致使将元数据存储在所述第三组磁道中;(c)致使将一个或多个文件存储在所述非高速缓存器介质上,其中所述一个或多个文件是存储在不含有文件系统信息、可启动性信息和分区信息中的任何信息的所述非高速缓存器介质上;(d)致使将所述非高速缓存器介质中每个文件的位置存储在所述第四组磁道中;以及(e)致使存储所述非高速缓存器介质中每个文件的位置与文件的主机名和/或地址的相关性。

[0026] 借助本发明的各种实施方案,相关人员可以提高存储和检索数据的效率。可以通过比普遍使用的方法使用更小的存储空间以及在存储信息的活动中投入更少的时间和努力,而实现效率的提高。在一些实施方案中,相关人员也可以针对数据文件的未授权检索来提高保护。这些益处可以在远程或本地存储数据时加以实现,并且本发明的各种实施方案可以与RAID技术结合使用或者独立于所述RAID技术来使用。

[0027] 附图简述

[0028] 图1为本发明系统的概况的表示图。

[0029] 图2为中介器和非高速缓存器介质(NCM)的表示图。

[0030] 图3为使用中介器来存储信息的系统的表示图。

[0031] 图4为使用两个中介器来备份所存储的信息的系统的表示图。

[0032] 图5为二进制树的表示图。

[0033] 图6为表示本发明的方法的流程图。

[0034] 发明详述

[0035] 现将详细参考本发明的实施方案,其示例在附图中予以说明。在以下详细描述中,阐述了许多具体细节以提供对本发明的充分理解。然而,除非上下文中另有指示或暗示,否则这些细节都是作为示例,并且不应该视为以任何方式来限制本发明的范围。另外,标题是为了方便读者而使用的,而不是意在限制本发明实施方案中的任何实施方案的范围,也不是意在表明任何特征是限于特定标题内的章节。

[0036] 定义

[0037] 除非上下文中另有陈述或暗示,否则下述术语和短语都具有下文所提供的意义。

[0038] 术语“比特”是指二进制数字。其可以具有两个数值中的一个数值,这个数值可以由0或1表示。

[0039] 术语“区块”是指具有预定长度的数据字节或比特序列。

[0040] 短语“可启动性代码”、“可启动性信息”和“可启动性特征”是指提供相应方式来进入可启动状态并且可以存储在启动扇区上的信息。启动扇区可以含有机器代码,所述机器

代码被配置成由固件加载到RAM(随机存取存储器)中,这又允许启动进程来从存储装置加载程序或将程序加载到存储装置上。举例而言,主启动记录可以含有定位现用分区并调用卷启动记录的代码,其可以含有代码来加载和调用操作系统或其它独立程序。

[0041] 术语“字节”是指八比特序列。

[0042] 术语“高速缓存器”是指为了将来更快提供数据的请求或出于缓冲的目的而临时存储数据的位置。L1高速缓存器(1级高速缓存器)是指(例如)与处理器核心集成的静态存储器。L1高速缓存器可以用来在CPU(中央处理单元)多次存取相同数据的情况下提升数据存取速度。L2高速缓存器(2级高速缓存器)通常比L1高速缓存器更大,并且,如果在L1高速缓存器中寻找但并未发现数据文件,那么便可以在查看外部存储器之前对L2高速缓存器进行搜索。在一些实施方案中,L1高速缓存器并不处于中央处理单元内。相反,其可以位于DDR、DIMM或DRAM内。另外或替代地,L2高速缓存器可以是PCI2.0/3.0的一部分,所述PCI2.0/3.0并入到主板中。因此,L1高速缓存器和L2高速缓存器中的每个高速缓存器可以处于主板的单独部分中。就大小而言,在本发明的一些实施方案中,L1高速缓存器介于2GB与128TB之间或者介于2GB与4TB之间;而L2高速缓存器介于16GB与1PB之间或者介于16GB与3.2TB之间。在一些实施方案中,当实施本发明的方法时,比特标记符表、频次转换器(frequency converter)或散列数值表存在于L2高速缓存器中。

[0043] 术语“小盘”是指可以对应于扇区群集的一组比特。小盘的大小由存储系统确定,并且可以具有某个小盘大小。传统上,小盘大小是通过CHS方案得出,所述CHS方案会经由元组来定址区块,所述元组定义所述区块在硬盘上出现的柱面、磁头和扇区。更近地,小盘大小已经从LBA测量中得出,所述LBA测量是指逻辑区块定址,并且作为另一种方式用以指定存储在计算机存储装置上的数据区块的位置。举例而言,小盘大小可以是512B、1K、2K、4K、8K、16K、32K、64K或1MB。正如本领域普通技术人员所知晓的,1K=1024B。小盘可以从主机作为原始数据进行接收。

[0044] “文件”是具有某个长度的相关字节或比特的集合。文件可以小于小盘、与小盘具有相同大小或者大于小盘。

[0045] 短语“文件名”是指允许计算机识别特定文件并将所述文件与其它文件加以区别的名称或代码。

[0046] 短语“文件系统”是指用来存储、检索和更新一组文件的抽象概念。因此,文件系统是用来管理数据和文件的元数据以及含有数据的存储装置上的可用空间的存取的工具。一些文件系统可以(例如)存在于服务器上。文件系统的示例包括但不限于Unix文件系统和其关联的目录表和索引节点、Windows FAT16和FAT32文件系统(FAT是指文件分配表)、Windows NTFS(其基于主文件表)以及Apple Mac OSX(其使用HFS或HFS plus)。

[0047] 短语“散列函数”、“加密散列函数值算法”和“散列函数值算法”是指针对特定散列函数而将较大数据组(可选地具有可变长度)映射到具有固定长度的更小数据组的算法或子程序。“散列函数值”是指在应用散列函数算法之后返回的输出。所述算法返回的数值也可以称为散列值、散列代码、散列和、校验和或散列数据。当(例如)使用MD5时,输出为128个比特,然而,当使用SHA-1时,输出便为160个比特。

[0048] 术语“主机”和“发起器”可以互换使用,并且是指发送数据以便存储到本发明的数据存储和检索中介系统上的实体或系统。主机可以发送与一种或多种类型的文档或文件以

及接收的数据相对应的数据。优选地,在任何输入/输出(“I/O”)流内,数据对应于单个文档类型的文件。出于便利性目的,短语“I/O流”用来指代数据从一个实体传输到另一实体。第一实体的输出内容可以是第二实体的输入。

[0049] 缩写“LBA”是指逻辑区块定址。LBA是线性定址方案,并且是用来指定存储在某些存储介质(例如,硬盘)中的数据区块位置的系统。在LBA方案中,区块是通过整数来定位,并且只有一个数字用来定址数据。通常,第一区块为区块0。

[0050] 缩写“LUN”是指逻辑单元数字,并且是用来识别逻辑单元的数字。LUN通常用来管理SAN上共享的区块存储阵列。

[0051] 术语“管理器”是指可以存储在非暂时性介质中并且致使采取一个或多个其它动作(例如,接收、传输、存储或处理数据)的计算机程序产品,例如,代码。管理器可以存储在硬件、软件或其组合上。在一些实施方案中,管理器可以是计算机和/或系统的一部分,所述计算机和/或系统被配置成允许管理器执行其预期功能。

[0052] 术语“中介器”是指可以存储在硬件、软件或其组合上并且使至少一个非高速缓存器介质内存储空间的一个或多个单元与文件名相互关联的计算机程序产品。中介器可以是比其所指向的非高速缓存器介质更小的量级。举例而言,中介器可以大约小到典型柱面大小的约0.2%。在一些实施方案中,中介器可以存在于计算云中,然而在其它实施方案中,其存在于非暂时性有形记录介质中。中介器能够组织、转换、转译并控制主机察觉到是处于记录介质的某些磁道中而实际上出现在记录介质的不同磁道中的位置中的数据存储,或者中介器可以可操作性地耦接到管理器,所述管理器即便不会提供所有这些功能,也会提供其中的一个或多个功能。此外,中介器可以包括可位于物理装置或结构内的扇区图、表或其它数据组织,且因此,中介器的内容可以致使物理装置或结构具有某种几何排列。在一些实施方案中,中介器永久性地存在于L2高速缓存器上。在其它实施方案中,其存在于固态驱动器上。

[0053] 术语“元数据”是指关于数据容器的管理信息。元数据的示例包括但不限于:正在读取的文件的长度或字节计数;与文件修改的最后一次有关的信息;描述文件类型和存取许可的信息;以及LUN QoS、VM和WORM。其它类型的元数据包括操作系统信息、自动初始化信息、分组许可以及文档类型内比特的频次。在一些实施方案中,存储的元数据可以(例如)用来允许有效地紧缩或扩展发起器的存储空间,因为发起器力图存储的文档的数目和大小在缩减或增加。

[0054] 缩写“NAS”是指网络区域存储。在NAS系统中,磁盘阵列可以连接到针对局域网络传送而给予访问权限的控制器。

[0055] 短语“可操作性地耦接”意味着系统、装置和/或模块被配置成彼此或互相通信,并且能够在通信时或通信之后执行它们的预期目的。

[0056] 短语“操作系统”是指管理计算机硬件资源的软件。操作系统的示例包括但不限于Microsoft Windows、Linux和Mac OS X。

[0057] 术语“分区”是指将存储介质(例如,磁盘驱动器)划分为多个单元的格式。因此,分区也可以称为磁盘分区。分区的示例包括但不限于GUID分区表和Apple分区图。

[0058] 缩写“RAID”是指独立磁盘冗余阵列。对于相关服务器而言,这组磁盘可能看上去像单个卷。RAID技术通过从多个磁盘牵引单条数据来提升性能,并且建立在一种或多种前

提类型上,诸如:(1)数据的镜像;(2)数据的条状化;或者(3)数据镜像和条状化的组合。

[0059] 术语“记录介质”是指非暂时性有形计算机可读存储介质,在其中相关人员可以存储对应于比特的磁性信号。举例而言,记录介质包括但不限于非高速缓存器介质,如硬盘和固态驱动器。正如本领域普通技术人员所知道的,固态驱动器也具有高速缓存器,并且不需要自旋。非暂时性有形计算机可读存储介质的示例包括但不限于硬驱、硬盘、软盘、计算机磁带、ROM、EEPROM、非易失性RAM、CD-ROM以及穿孔卡。

[0060] 缩写“SAN”是指存储区域网络。这种类型的网络可以用来将计算装置链接到磁盘、磁带阵列以及其它记录介质。数据可以(例如)在SAN上以区块形式进行传输。

[0061] 缩写“SAP”是指系统辅助处理器,其是操作系统所使用的I/O(输入/输出)引擎。

[0062] 缩写“SCSI”是指小型计算机系统接口。

[0063] 术语“扇区”是指磁盘(例如,磁性磁盘)上磁道的细分。每个扇区存储固定量的数据。针对磁盘而言的普通扇区大小为512个字节(512B)、2048个字节(2048B)和4096个字节(4K)。如果小盘的大小为4K且每个扇区的大小为512B,那么每个小盘对应于8个扇区($4 \times 1024 / 512 = 8$)。扇区具有磁道并且位于盘片上。通常,两个或四个盘片构成一个柱面,并且255个柱面构成硬盘和介质装置。

[0064] 短语“扇区图”是指从主机接收调用信号并关联存储装置中文件所存储的位置的工具。扇区图可以(例如)在SCSI协议定义的参数下进行运作。在本发明的一些实施方案中,扇区图可以位于中介器的比特字段中。

[0065] 术语“磁道”是指磁盘内横跨所有扇区的圆形单元。“磁道扇区”为任何一个扇区内的磁道。“磁道群集”跨越一个以上扇区。

[0066] 优选实施方案

[0067] 本发明提供用于将数据存储在非高速缓存器记录介质上的方法、用于执行这些方法的计算机程序产品以及被配置来执行这些方法的系统。借助本发明的各种实施方案,相关人员可以有效地存储和检索数据。

[0068] 比特标记符

[0069] 根据一个实施方案,本发明针对一种用于将数据存储在记录介质上的方法,所述方法使用比特标记符作为原始数据内比特串的表示形式。因此,原始数据被转译成一系列表示所述原始数据的标记符,并且所述方法提供接收含有会被转换成一组信号(可以称为比特标记符)而用于存储的数据的文件或流,并且存储这些信号。

[0070] 文件可以从称为主机的个人或实体进行接收。优选地,主机将以原始数据的形式来发送信号。举例而言,主机可以发送单独或共同形成一个或多个文件(如JPEG、PDF、TIFF或WORD文档)的一个或多个小盘。本发明的各种实施方案开始于接收小盘、接收小盘的子单元或者接收有待转换成小盘的一个或多个文件。优选地,所接收的小盘或小盘的子单元都具有统一大小。可选地,所述方法验证大小的统一性,并且如果检测到不统一性,例如,一个或多个小盘具有过少的比特,那么所述方法便添加0,直到所有小盘都具有统一大小。

[0071] 举例而言,数据可以在长度为N个比特的小盘中进行接收,其中N是大于1的整数。举例而言,N可以是128个字节至16K,例如,4K,这对应于4096B。

[0072] 小盘是按顺序接收的。举例而言,文件可以含有系统连续接收的十个小盘。或者,针对给定文件的多个小盘可以并行或一起传输,条件是它们含有允许其以合适方式彼此重

新关联的信息,所述方式允许通过主机的操作系统重新创建和使用所述文件。因此,在一些实施方案中,本发明的方法以接收小盘的相同顺序来存储标记符。因此,当主机想要检索文件时,对应的检索方法论将会以相同顺序来调回编码数据,并按照合适顺序将所述编码数据解码成小盘。

[0073] 可选地,在编码之前,系统可以将小盘划分为比特群组,也称为子单元,其中每个子单元的长度为A个比特。如果系统将小盘划分为子单元,那么这些子单元可以与比特标记符表进行比较。如果系统并未将小盘划分为子单元,那么每个小盘便可以与比特标记符表进行比较。

[0074] 比特标记符表使独特组的比特与独特标记符相互关联。在一些实施方案中,比特标记符表含有标记符而用于使用子单元时具有大小A的每个独特比特串,或不使用子单元时具有大小N的每个独特比特串。因此,在这种方法下,计算机程序可以接收一组小盘作为输入。所述计算机程序随后可以将每个小盘划分为具有相同大小且每个的长度为A个比特的Y个子单元,其中A/8是整数。对于每个独特的A而言,可以在所述表内存在标记符。

[0075] 因此,借助自动化协议,在接收小盘之后,计算机程序产品会致使存取比特标记符表。因此每个小盘或子单元可以充当输入,并且每个比特标记符可以充当输出,从而形成输出的标记符组。所述输出的标记符组可以称为转译数据、译码数据或编码数据。在每个小盘并未再分的实施方案中,每个小盘会接收一个标记符。如果小盘划分为两个子单元,那么所述小盘会转译或编码成两个标记符。因此,计算机程序产品使用会将标记符与输入相互关联的比特标记符表,以便指配对应于每个小盘的至少一个标记符。计算机程序产品可以经过设计,从而使得生成对应于每个单独标记符的不同输出、生成含有一组对应于每个小盘的标记符的不同输出或者生成含有对应于完整文件的所述组标记符的不同输出。

[0076] 比特标记符表含有X个标记符。在一些实施方案中,X等于长度为N的小盘内不同的比特组合的数目,条件是所述方法并未将小盘划分为子单元,或者X等于长度为A的子单元内不同的比特组合的数目,条件是所述方法划分小盘。如果文档类型已知或预期具有比针对给定长度子单元或小盘的所有比特组合少,那么X(标记符的数目)可以小于可能的比特组合的实际数目。举例而言,在一些实施方案中,所有比特标记符都具有相同大小,并且比特标记符表内的比特标记符的数目等于大小为N或A的比特串内的比特组合的数目。在其它实施方案中,所有比特标记符都具有相同大小,并且比特标记符表内的比特标记符的数目小于大小为N或A的比特串内的比特组合数目的90%、小于其80%、小于其70%或者小于其60%。

[0077] 举例而言,在一些是实施方案中,每个小盘指配有由多个0和/或1组成的代码(也就是标记符)。在其它实施方案中,每个小盘划分为多个子单元,其中每个子单元指配有由多个0和1组成的代码(也就是标记符)。子单元可以由长度A界定,其中N/A=Y且Y为整数。如果任何子单元并不具有此数目的比特,例如,一个或多个子单元比系统经过配置而接收为输入的比特具有更小数目的比特,那么系统便可以添加比特,例如0,直到所有子单元具有相同大小。此步骤可以(例如)在小盘划分为子单元之后,并且在没有首先进行检查来查看是否所有小盘都具有相同大小的情况下执行。或者,且正如上文所描述的,此步骤可以在将小盘划分为子单元之前,在小盘层面上执行。

[0078] 正如上文的描述所表明的,所述算法可以被配置来将比特串转译成一组编码数

据,并且所述算法可以经过设计,从而使得比特串对应于小盘或对应于小盘的子单元。优选地,所述组编码数据小于从主机或客户端所接收的文件。然而,无论所述组编码数据是否小于起初数据,所述组编码数据都能够转回成文件的小盘。正如本领域普通技术人员会认识到的,从主机接收而用于存储的数据将是原始数据,并且因此可以对应于任何文档类型。

[0079] 编码可以服务于两个独立目的。第一,通过编码用于存储的数据,会增强安全性。只有知道代码(也就是对比特标记符表具有访问权限)的个人或实体才能够对所述数据进行解码并且重新构建文档。第二,如果使用比起初文档更少的比特来创建代码,那么便会需要更少的存储空间,并且可以节约成本。如果用于一个或多个文件的存储空间量减小,那么NCM便可以存储对应于更大数目文件和/或更大文件的数据。

[0080] 对于表内的至少多个独特比特组合而言,优选地,如果系统并未将小盘划分为子单元,那么标记符便小于小盘长度N,或者,如果系统确实将小盘划分为子单元,那么标记符便小于子单元长度A。优选地,如果系统并未将小盘划分为子单元,那么不会有任何标记符大于小盘长度N,或者,如果系统确实将小盘划分为子单元,那么不会有任何标记符大于子单元长度A。在一些实施方案中,所有标记符都小于N或者小于A。另外,在一些实施方案中,每个标记符都可以具有相同大小,或者两个或两个以上标记符可以具有不同大小。当存在具有不同大小的标记符时,这些不同大小的标记符可以(例如)处于所述表中。或者,在所述表内,所有标记符都具有相同大小,但是在存储之前,从这些标记符的一个或两个末端移除所有的0。

[0081] 在计算机程序产品将小盘转译成多个标记符之后,其会致使将所述多个标记符(已经或尚未从末端移除0)以与小盘的顺序相对应的顺序存储在非暂时性记录介质上,或者以另一顺序存储在非暂时性记录介质上,其中可以根据此顺序按照其他方式重新创建小盘的顺序。最后,标记符将会存储在非暂时性介质中,所述介质为非高速缓存器介质。然而,可选地,这些标记符可以首先发送给高速缓存器介质,例如L1和/或L2。因此,为了不损失任何信息,比特标记符表可以存储在永久性存储器中,但是在使用时,复本可以处于(例如)L2高速缓存器中,并且在启动或重启后,L2高速缓存器中表的填写可以来自永久性存储器。

[0082] 在一个实施方案中,存储装置将对应于给定文件的多个比特标记符存储在非高速缓存器介质中。这些比特标记符的大小范围为X至Y,其中X小于Y并且至少两个标记符具有不同大小。在检索比特标记符时或之后,计算机算法将0添加到所有小于预定大小Z的比特标记符的一个末端,其中Z大于等于Y。可以查询查找表,其中大小为Z的每个标记符都转译成长度为A的比特串,其中A大于或等于Z。在非限制性示例中,X=4,Y=20,Z=24,A=32。在一些实施方案中,A至少比Z大50%。对应于A的比特串可以是组合成小盘的子单元,或者它们本身可以是小盘。

[0083] 频次转换器

[0084] 正如上文所描述的,比特标记符表可以针对原始数据以随机或非随机的方式将标记符指配给比特串,并且比特标记符可以具有统一或不统一的大小。然而,相关人员可以不使用如上文所述的比特标记符表,而使用频次转换器。

[0085] 正如本领域普通技术人员会认识到的,本公开的示例性章节中的表I和表II以独立于原始数据中比特串的频次的方式来指配比特标记符(也就是转换比特)。然而,正如上文所提及的以及下文在示例3中所解释的,相关人员可以将更小的标记符指配给预期更为

频繁地出现在文档类型或一组文档中的原始数据。这个策略利用如下事实：全部信息的大约80%是包含在最频繁子单元的大约最靠前的20%内。换句话来说，对应于数据的子单元是高度重复的。

[0086] 为了进一步说明频次转换器可以如何使用具有不同大小的比特串，可以参考图5。图5展示针对长度为五个单元并且开始于数字1的所有二进制序列的二进制树。正如所述树展示的，存在16个路径来创建长度为五个数字的序列，这些路径开始于顶行并沿着树向下移动到底部。然而，当沿着树向下移动时，相关人员可以沿着分支向下移动到（例如）第三、第四或第五行。因此，所述方法可以经过设计，从而使得对于一条数据而言，所述方法指配对应于部分沿着树向下移动的代码，然而，对于其它条的数据而言，所述方法指配对应于沿着不同分支移动更大数目单元的代码。举一个非限制性示例，对于特定条的转换数据而言，相关人员可以停止于第三行上，例如，生成序列101，然而，对于所有其它条的数据而言，前三个数字不是101。因此，所述方法从不允许使用：1010、1011、10100、10101、10110和10111（这些处于图5中的方框内）。在从存储器检索数据后，系统会读取数字，直到其将这些数字辨识为独特的，并且在将它们辨识为独特后，所述系统会使用独特串作为容许解码的程序的输入，这会允许为主机重新创建原始数据文件。

[0087] 本发明的各种方法可以在检索过程中或检索之后要求读取最小数目的比特，并在这个最小数目（其对应于最短独特代码）之后，通过与数据文件或表进行比较来检查独特性，而且，如果代码在数据文件或表内不是独特的，那么所述方法便可以继续扩展比特的数目，并且在添加每个单独比特之后检查所扩展代码的独特性直到序列被识别为独特的。前述示例是就3比特、4比特或5比特串来描述的，但是这些大小仅仅用于说明性目的。在一些实施方案中，与频次转换器结合使用的最短独特比特序列的长度为10至15个比特，并且最长比特序列的长度为25至40个比特。

[0088] 在一些实施方案中，对于具有不同大小的每批多个转换比特串而言，存在长度为A个比特的第一转换比特串以及长度为B个比特的第二转换比特串，其中A<B，并且，第一转换比特串的A个比特的身份与第二转换比特串的前A个比特的身份并不相同。

[0089] 正如表I、表II和表III（参看示例）所展示的，信息可以进行转换，并且输出代码可以被配置成比输入占据更少的空间，原因在于标记符用来表示比特群组。因此，优选地在表内，标记符中的至少一个标记符、多个标记符、至少50%、至少60%、至少70%、至少80%、至少90%或至少95%在大小上小于子单元。然而，不存在任何技术障碍来使得转换数据与从主机所接收的数据或根据散列函数值算法而产生的数据具有相同大小或比其更大。

[0090] 分割

[0091] 如上文所阐明的，在接收数据流之后，在一些实施方案中所述方法需要分割所接收的数据。如果所接收的数据呈较大文件的形式，那么所述数据可以分割成小盘。如果所接收的数据呈小盘的形式，那么小盘可以分割成子单元。较大文件可以首先分割成小盘，且然后这些小盘可以分割成子单元。因此，对于来自主机的任何输入而言，可能存在零个、一个或两个分割步骤。

[0092] 通过分割I/O流，所述方法将每个I/O流划分为具有预定大小且优选地具有统一大小的更小单元（其也可以称为区块）。正如本领域普通技术人员会认识到的，使用更小区块大小的一个益处在于，较小区块容许更轻易地进行数据处理。举一个非限制性示例，I/O流

的长度可以是4096个字节，并且所述方法可以将这个流分割成长度为1024、512、256或128个字节的片段单元。在接收数据之后或者数据正被接收时，可以运行某个算法，所述算法将小盘划分为(例如)具有32个比特的子单元。

[0093] 子单元的大小是从主机接收数据的系统的设计者的一个选择。然而，子单元的大小应经过选择，使得将小盘划分为具有一致大小的子单元，并且子单元可以轻易地与比特标记符表或频次转换器的查询结合使用。

[0094] 如果小盘中的任何小盘小于其它小盘，那么可选地，在接收具有更小大小的那个小盘后，所述算法便添加0，从而将所述更小的小盘渲染成与其它小盘具有相同大小。或者，系统可以将小盘划分为子单元，并且在获取小于期望长度的子单元后，将0添加到这个子单元的末端。

[0095] 预处理

[0096] 在转译过程(其也可以称为编码过程)期间，算法针对比特标记符表或频次转换器而用作输入的比特串(也就是小盘或子单元)可以进行预处理。这些比特串中的每个比特串都可以由第一末端和第二末端界定，并且在指配标记符之前，所述方法还包括分析每个比特串，以便确定第二末端处的比特是否具有数值0。如果第二末端处的比特具有数值0，那么所述方法便可以移除第二末端处的比特以及具有数值0且与第二末端处的比特形成连续比特串的所有后续比特，从而形成大小上减小的比特串。预处理步骤的益处在于，可以使用更小的比特标记符表或频次转换器。举例而言，可以不使用表I，而是使用表II来产生相同的编码数据。正如本领域普通技术人员会认识到的，这个预处理可以通过从第二末端搜索和移除1而不是0来完成。

[0097] 截断

[0098] 在本发明的另一实施方案中，相关人员不是通过使用比特标记符表或频次转换器来转译，而是可以按照截断子单元在小盘内存在的相同顺序来存储截断子单元(或者，如果不使用子单元，那么可以截断和存储小盘)。因此，不是使缩短的比特串充当预处理数据来用作(例如)比特标记符表中的输入，而是可以存储比特串本身。

[0099] 因此，在一些实施方案中，存在另一种用于将数据存储在记录介质上的方法。根据这种方法，相关人员接收多个数字二进制信号，其中所述数字二进制信号是组织在呈上文所述格式的小盘中。可选地，每个小盘可以划分为子单元，正如上文所提供的。

[0100] 每个小盘或子单元可以由其长度界定，并且每个小盘或子单元具有第一末端和第二末端。相关人员可以分析每个小盘或子单元以便确定所述第二末端处的比特是否具有数值0，并且，如果第二末端处的比特具有数值0，那么便移除第二末端处的比特以及具有数值0且与第二末端处的该比特形成连续比特串的所有比特，从而针对在第二末端处具有0的任何小盘或子单元来形成修正小盘或修正子单元。因为这些修正比特串比起初比特串更短，所以它们可以称为是截断的。

[0101] 在截断小盘或子单元之后，相关人员可以将截断的信息存储在非暂时性记录介质中。通过存储截断的信息，使用更少的比特来存储相同的信息，所述信息会以其它方式存储在未截断的比特串中。

[0102] 在上文所描述的各种方法中，相关人员可以从每个子单元或每个小盘的第一末端或第二末端移除数字，但是不会从这两个末端同时移除。然而，实践下述方法是处于本发明

的各种实施方案的范围内：相关人员考虑从每个子单元或小盘的第一末端移除数字；相关人员单独考虑从每个子单元或小盘的第二末端移除数字；对于每个子单元或小盘而言，相关人员分析截断发生在第一末端和第二末端中的任一末端、一个末端还是两个末端处，并且，如果截断只发生在一个末端处，那么便保存截断的小盘或子单元，而如果截断发生在两个末端处，那么便保存截断单元中更小的单元。实践可以从小盘或子单元的两个末端处移除数字的方法，也是处于本发明的范围内。

[0103] 因此，相关人员可以接收多个数字二进制信号。二进制信号可以在单元（例如，小盘或者小盘的子单元）中进行接收。每个单元的长度可以是相同数目的比特，并且每个单元具有第一末端和第二末端。单元内的比特数目为大于1的整数，并且这些比特在单元内具有某个顺序，而这些单元也具有某个顺序。

[0104] 相关人员随后可以分析每个单元，从而确定第一末端处的比特是否具有数值0，并且，如果第一末端处的比特具有数值0，那么便移除第一末端处的比特以及具有数值0且与这个比特形成连续比特串的所有比特，从而针对在第一末端处具有0的任何单元来形成第一修正单元。

[0105] 相关人员也可以分析每个单元以便确定第二末端处的比特是否具有数值0，并且，如果第二末端处的比特具有数值0，那么便移除第二末端处的比特以及具有数值0且与这个比特形成连续比特串的所有比特，从而针对在第二末端处具有0的任何单元来形成第二修正单元。

[0106] 对于每个单元而言，可以应用下述决策树：(a) 如果第一修正单元和第二修正单元的大小是相同的，那么便存储第一修正单元（或所述第二修正子单元）；(b) 如果第一修正单元小于第二修正单元，那么便存储第一修正单元；(c) 如果第二修正单元小于第一修正单元，那么便存储第二修正单元；(d) 如果不存在修正单元，那么便存储所述单元；(e) 如果不存在第一修正单元，但是存在第二修正单元，那么便存储第二修正单元；以及(f) 如果不存在第二修正单元，但是存在第一修正单元，那么便存储第一修正单元。

[0107] 相关人员也可以存储指示是否从第一末端或第二末端移除了一个或多个比特的信息，或者，相关人员可以将第一比特标记符表用于从第一末端移除比特的单元，并将第二比特标记符表用于从第二末端移除比特标记符的单元，而且在这两个比特标记符表之间，不存在比特标记符的任何复制形式。这两个不同的比特标记符表可以组织为同一张表的区段，并且包括用于未修正的单元的比特标记符。在所述一张或多张表中，不存在比特标记符的任何复制形式来用于第一修正单元、第二修正单元以及任何未修正的单元，因为（例如）它们在两个末端上都具有数值1。

[0108] 正如本领域普通技术人员会认识到的，尽管上文所描述的方法是结合移除0来描述的，但是系统也可以替代移除1。另外，这些方法被描述成是用作使用比特标记符表或频次转换器的方法的替代方法。然而，将这些方法与比特标记符表或频次转换器结合使用也是处于本发明的范围内，例如，通过截断作为那些协议的输出的标记符，只要它们是以允许检索和重构起初文件的方式存储的。

[0109] 散列值

[0110] 在上文所描述的实施方案中的某些实施方案中，相关人员存取比特标记符表或频次转换器来编码数据。在另一实施方案中，小盘或子单元充当加密散列函数值算法的输入。

[0111] 因此,对于每个输入而言,存在大小上小于所述输入的输出。举例而言,如果存在具有X个字节的子单元(其也可以称为具有X个字节的片段单元),那么输出便是针对每个具有X个字节的片段单元的生成的散列函数值。散列函数值算法的示例包括但不限于MD5散列算法(也称为消息摘要算法)、MD4散列算法和SHA-1。从散列函数值算法中输出的值可以称为校验和或和。在一些实施方案中,所述和大小为64、128、160或256个比特。由于I/O流内数据高度重复的特性,生成冲突和(也就是,相同的但对应于不同片段单元的和)的可能性便相对较低。

[0112] 在一些实施方案中,每个散列值是由多个0和/或1组成的、优选地具有相同长度的比特串或代码。这些散列值可以用来在非高速缓存器记录介质上进行存储。

[0113] 在一个实施方案中,系统可以(例如)执行一种用于将数据存储在非高速缓存器记录介质上的方法,所述方法包括:(i)接收具有N个字节的I/O流;(ii)将所述N个字节分割成具有X个字节的片段单元;(iii)使加密散列函数值与具有X个字节的每个片段单元关联,其中所述关联包括存取相关性文件,其中所述相关性文件使所存储的具有Y个比特的散列函数值与多个X字节存储序列中的每个序列相互关联,并且:(a)如果具有分割的X个字节的序列是处于所述相关性文件中,那么便使用所存储的具有Y个比特的散列函数值来在非高速缓存器记录介质上进行存储;以及(b)如果具有分割的X个字节的序列并未处于所述相关性文件中,那么便生成具有Y个比特的新散列函数值,并将其与所述具有X个字节的片段序列一起存储在所述相关性文件中,并使用所述新散列函数值来在所述非高速缓存器记录介质上进行存储。

[0114] 上述方法要求分割具有N个字节的I/O流。然而,所述方法可以通过将加密散列函数值算法应用到I/O流并且不将其分割而加以应用。因此,一种替代方法包括:(i)接收具有N个字节的I/O流;(ii)使加密散列函数值与具有N个字节的每个单元关联,其中所述关联包括存取相关性文件,其中所述相关性文件使所存储的具有Y个比特的散列函数值与多个N字节存储序列中的每个序列相互关联,并且:(a)如果具有N个字节的序列是处于所述相关性文件中,那么便使用所存储的具有Y个比特的散列函数值来在非高速缓存器记录介质上进行存储;以及(b)如果具有N个字节的序列并未处于所述相关性文件中,那么便生成具有Y个比特的新散列函数值,并将其与所述N个字节一起存储在所述相关性文件中,并使用所述新散列函数值来在所述非高速缓存器记录介质上进行存储。另外,尽管各种方法是出于说明性目的而描述成是应用到一组N个字节或X个字节,但是本领域普通技术人员将会了解,所述方法可以且优选地应用于文件的所有N个字节或X个字节。

[0115] 另外,上述方法只是在字节串已经不处于相关性文件内时才应用所述算法。在一实施方案中,本发明提供一种用于将数据存储在非高速缓存器记录介质上的方法,所述方法包括:(i)接收具有N个字节的I/O流;(ii)将加密散列函数值算法应用到具有N个字节的每个单元,以便针对具有N个字节的每个单元来形成生成的散列函数值;(iii)存取相关性文件,其中所述相关性文件使所存储的具有Y个比特的散列函数值与多个N字节存储序列中的每个序列相互关联,并且:(a)如果针对所述具有N个字节的单元的所述生成的散列函数值是处于所述相关性文件中,那么便使用所存储的具有Y个比特的散列函数值来在非高速缓存器记录介质上进行存储;以及(b)如果针对所述具有N个字节的单元的所述生成的散列函数值并未处于所述相关性文件中,那么便将所述具有Y个比特的生成的散列函数值与

所述具有N个字节的片段单元一起存储在所述相关性文件中，并使用所述生成的散列函数值来在所述非高速缓存器记录介质上进行存储。这个实施方案也可以修改成增加下述步骤：分割所述N个字节，以便形成片段单元；以及将所述加密散列函数值算法应用到所述片段单元，而不是应用到所述具有N个字节的单元。

[0116] 在上文所描述的实施方案中的某些实施方案中，在针对每个小盘、片段单元或子单元获取校验和之后，相关人员存取相关性文件。这些方法可以根据先进先出（“FIFO”）协议来获取校验和，并且在I/O流正在接收时、校验和正在生成时或者在所有I/O流已经接收、分割并经受散列函数值算法之后，开始存取相关性文件。

[0117] 如上文说阐明的，相关性文件使具有Y个比特的不同存储散列函数值与多个X字节存储序列中的每个序列关联。因此，在存在生成的散列函数值的这些情况下：(a) 如果针对具有X个字节的单元的生成的散列函数值是处于相关性文件中，那么所述方法便致使利用所存储的具有Y个比特的散列函数值来在非高速缓存器记录介质上进行存储；以及(b) 如果针对具有X个字节的单元的生成的散列函数值并未处于相关性文件中，那么所述方法便致使将具有Y个比特的生成的散列函数值与具有X个字节的序列存储在相关文件中，并使用生成的散列函数值来在非高速缓存器记录介质上进行存储。

[0118] 冲突解决模块

[0119] 作为应用散列值算法的结果而生成冲突散列值的可能性是较低的。因此，在一些实施方案中，相关人员可以选择使用本文中所描述的方法，而不使用冲突解决模块。然而，在各种实施方案中，本发明应用冲突解决模块，从而解决可能发生冲突的情形。

[0120] 如图6中所示，系统可以接收几组N个字节的流610。系统随后可以致使将所述N个字节分割成具有合意大小的单元620，从而针对每个片段单元来应用散列函数并生成散列函数值630。接着，系统查询是否每个散列值已经处于相关性文件内640。

[0121] 在这些实施方案中，存在冲突解决模块，每当对于具有X个字节的片段单元而言，生成与相关性文件中的散列值相同的散列值时，便会激活所述冲突解决模块650。这个模块随后查询对于所述散列值而言具有X个字节的片段单元是否与已经存储的X字节数值相同。如果冲突解决模块确定对于具有X个字节的片段单元而言，生成与相关性文件中的存储散列函数值相同的散列函数值，但是具有X个字节的片段单元不同于与所存储的散列函数值关联的X个字节，那么便有必要更新相关性文件并将生成的散列值存储在其中，以及写入到NCM 660。举例而言，在下文所描述的一些实施方案中，所述模块会致使存在不同的Z个比特与所存储的散列函数值以及生成的散列函数值关联。

[0122] 如果生成的散列值（与任何关联的Z值（若存在）一起）是独特的，那么其便可以用来在NCM上进行存储，并且相关性文件可以被更新成包括关联性680。此外，如果散列值已经处于相关性文件中，并且所接收的N个字节与相关性文件中的那些字节相同，那么所存储的散列值（与任何关联的Z值（若存在）一起）便可以写入到NCM，并且系统不需要更新相关性文件670。因此，可以在存取相关性文件之后并且只有在新生成的散列值已经处于相关性文件内的情况下，才会作为检查来访问冲突模块。冲突模块随后会确定是否存在冲突，或者来自所接收文件的校验和以及片段单元是否都已经在相关性文件中彼此关联。

[0123] 前述实施方案描述一种方法，在这种方法中，存取相关性文件之前，每个片段比特单元经受加密散列函数值算法。然而，作为替代地，相关人员可以首先检查相关性文件中子

单元或小盘的存在，并且只有在片段单元已经不处于表中的情况下，才会应用加密散列函数值算法。然后，如果小盘或子单元已经处于相关性文件中，那么便使用所存储的校验和值。如果子单元或小盘并未处于文件中，那么相关人员将会应用散列值算法并进入协议来检查冲突。如果存在真实冲突，也就是，相同的散列值与不同的子单元或小盘关联，那么相关人员将会更新相关性文件来解决这个问题。

[0124] 一种用以解决真实冲突问题的方式是，对于不存在冲突的所有散列函数值而言，Z个比特进行关联，并且所述Z个比特具有（例如）8至16个0的统一长度。举一个非限制性示例，N=4096字节（小盘大小），X=512字节（子单元大小），Y=128或256比特（散列值大小），并且Z=8或16比特（散列值扩展大小）。所述方法可以（例如）在校验和与之前存储的校验和并不冲突时，在校验和的末端处关联8个0。在识别冲突后（例如，不同的片段单元与相同的校验和关联），最新的校验和可以指配有不同的Z值。因此，如果存储在相关性文件中的Z值是00000000，那么针对第一冲突校验和的Z值可以是00000001，并且应存在另一冲突校验和00000010。如果存在额外的冲突校验和，那么每个冲突校验和可以随着冲突校验和被识别而指配有下一个Z值。

[0125] 在针对每个片段单元获取校验和后，并且紧接着冲突模块的应用（若存在）之后，多个散列函数值（且在一些实施方案中与Z值一起）可以写入到非高速缓存器记录介质而便于存储。

[0126] 散列值和比特标记符或频次转换器的组合使用

[0127] 在一些实施方案中，不是存储散列值，而是所述方法还包括：通过使用比特标记符表来转换（也称为编码）用于存储的每个散列函数值，以便针对用于存储的每个散列函数值来生成比特标记符；以及，将针对各别散列函数值的每个比特标记符（并且在一些实施方案中与Z值一起）存储在非高速缓存器记录介质中。通过使用比特标记符表，相关人员可以转换或编码散列值。因此，在存取比特标记符表时散列值可以充当输入子单元，并且关联的标记符可以充当输出。

[0128] 子单元可以由长度A界定，其中散列值的长度除以A是整数。如果任何比特串都不具有这个数目的比特，例如，一个或多个比特串比被配置来进行转换的子单元内的数目具有更小数目的比特，那么系统便可以添加比特，例如0，直到所有子单元具有相同大小。或者，可以在将散列值划分为子单元之前，在散列值层面上执行0的添加。此外，如果散列值（并且在一些实施方案中与Z值组合在一起）小于子单元大小，那么它们便可以进行组合来形成子单元，或者进行组合和分割来形成具有适当大小的子单元。

[0129] 散列值向编码数据的转换可以由算法或计算机程序来执行。算法或计算机程序的运行可以（例如）由管理器进行控制，所述管理器也控制散列函数值算法的应用。编码数据（其也可以称为转换数据）也是由二进制信号组成，并且所述编码数据是以允许其被转回成文件的散列值的方式进行编码和存储。因此，在编码过程期间保持信息，所述编码过程允许在不损失信息的情况下进行解码。

[0130] 编码可以服务于两个独立目的。第一，通过比特标记符表来转换用于存储的数据，便会增强安全性。只有知道代码的个人或实体才能够将所述数据解码以及重构文档。第二，如果使用比对应于文档的散列值更少的比特来创建代码，那么使用转换数据和散列值将会致使需要更少的存储空间，并且可以进一步节约成本。

[0131] 在一些实施方案中,不是使用比特标记符表,而是所述方法还包括,通过使用频次转换器来编码用于存储的每个散列函数值。在这些方法中,针对用于存储的每个散列函数值而言,生成转换比特串,其中,对于具有相同大小的散列值而言,所输出的转换比特串具有不同长度。频次转换器可以将多个转换比特串中的每个转换比特串与多个散列函数值中的每个散列函数值关联,并且频次转换器中的所述多个转换比特串是足够地截然不同,从而使得在回读时,转换比特串可以与适当的散列值相互关联。

[0132] 通过将散列值算法的使用与(1)比特标记符表应用或(2)频次转换器应用组合,相关人员便能够带来安全性的加强和/或成本的节约。另外,正如上文所描述的,散列值是针对比特标记符表或频次转换器而输入的。然而,顺序可以颠倒,且小盘或子单元可以进入使用比特标记符表或频次转换器来生成标记符的协议,并且这些标记符可以根据本文所描述的方法中的任何方法来充当散列函数算法的输入,这些方法包括但不限于使用冲突解决模块的方法。

[0133] 中介器和管理器

[0134] 比特标记符表、频次转换器、散列值相关性文件以及用以存取和使用前述文件和协议的程序中任何一者或每一者都可以存储在中介器内、管理器内或其它地方。每当前述协议或文件中的任何一个或多个协议或文件与本发明的实施方案结合使用,而且并未存储在中介器或管理器内时,优选地,其可操作性地耦接至中介器、管理器或这两者。用以与本地或远程存储的文件和程序进行通信的方法和系统对于本领域普通技术人员而言是熟知的。

[0135] 在各种实施方案中,管理器控制本发明的方法。举例而言,在I/O(从主机输出、输入到系统)从主机接收之后,管理器可以致使向主机确认I/O流的接收。

[0136] 主机可以将I/O流的文件记录成是存储在第一存储地址上。然而,转换比特串事实上可以存储在非高速缓存器记录介质中的第二存储地址上,其中第一存储地址与第二存储地址并不相同。此外,第一存储地址可以将文件展示为具有第一文件大小(或者与文件的名称关联),而第二存储地址可以对应于第二文件大小,其中第一文件大小至少是第二文件大小的两倍大或者至少是四倍大。

[0137] 在一些实施方案中,第二存储地址存储在中介器上。在中介器内,第二存储地址可以存储在比特字段中。本文中所描述的中介器可以与所描述的通过比特标记符表、频次转换器和散列值算法中的一者或者来转译数据的方法独立使用或结合使用。

[0138] 在一些实施方案中,中介器为硬件、软件或其组合,所述中介器包括:(i)第一组磁道,其中所述第一组磁道包括或含有与文件系统信息、可启动性信息和分区信息相对应的信息;(ii)第二组磁道,其中所述第二组磁道包括或含有与第一组磁道的复本相对应的信息;(iii)第三组磁道,其中所述第三组磁道包括或含有与不同于文件系统信息、可启动性信息和分区信息的元数据相对应的信息;以及(iv)第四组磁道,其中所述第四组磁道包括或含有对应于比特字段的信息。举例而言,中介器可以存在于L2高速缓存器中,并且比特标记符表、频次转换器或散列值表中的任何一者或者可以在使用时存在于服务器的RAM中,且也可以存在于固态装置的永久性储存器中,所述固态装置可以与上面存储标记符或散列值的NCM相同或不同。如果进行更新,那么会对RAM中的表做出更新,并且也会对永久性储存器做出更新。

[0139] 除了使用中介器之外,本发明的各种实施方案要求使用管理器。管理器可以包括一个或多个模块,并且可以存在于本地计算机上、网络上或云中。管理器被配置来协调某个信息的接收或接收某个信息本身,以及将这个信息传递给中介器,或直接通过中介器来控制信息的接收。因此,所述方法可以经过设计,从而使得来自发起器的信息经过管理器而流向中介器,或者管理器只指引信息流动到系统的其它部件,例如主机或NCM。

[0140] 在本发明的各种实施方案中,管理器可以应用或致使应用散列函数算法或其它协议、任何冲突模块(若存在)以及任何转换模块(若存在),并且致使数据直接流向中介器。管理器也可以通过使用中介器来控制信息的存储以及信息的检索和传输。当存储信息时,LBA编号可以进行识别,并且有待存储的数据可以发送给缓冲器,从而避免或减少瓶颈效应。此外,在存储过程中可以使用L1和/或L2高速缓存器。

[0141] 类似地,当检索数据时,所述方法可以从NCM调用数据、填充缓冲器、获取校验和值且然后重新创建片段单元和进行重组,或者如果没有进行分割,便直接重组来形成呈主机可以接收和核查的形式的信息。如果所存储的数据是转换数据,那么在获取校验和值之前,所述数据可以解码。这些步骤可以借助管理器来协调和控制,所述管理器会执行必要的协议。

[0142] 在一些实施方案中,管理器可以控制一个或多个中介器、与其进行通信并协调其活动。对于每个中介器而言,管理器接收一组参数(或协调这组参数的接收)。这些参数可以包括文件系统信息、可启动性信息和分区信息中的一者、两者或所有三者,基本上由其组成或者由其组成。管理器致使这个信息存储在中介器上的第一组磁道中,所述第一组磁道可以称为预留1或R₁。文件系统将规定如何使用预留区块。举例而言,当使用NTFS,扇区1至2可以用于MBR(主启动记录),而扇区3可以用于\$MFT。可选地,这些磁道可以复制到第二组磁道中,所述第二组磁道可以称为预留2或R₂。

[0143] 除了前一段落中所描述的参数之外,管理器也可以接收元数据。元数据存储在中介器上的第三组磁道中。在管理器接收参数和元数据的时间上或者在稍后时间上,管理器也可以接收一个或多个文件以便存储在非高速缓存器介质上。每个文件都是在带有文件名的情况下接收的。文件名由传输文件的主机生成,并且可以由主机的文件系统进行定义。可以(例如)作为SAN或NAS或其组合,或者作为SAN或NAS或其组合的一部分的管理器,在接收具有文件名的文件后,可以自动执行本文中所描述的步骤来进行存储。

[0144] 本发明的系统可以经过设计,从而使得散列函数值算法和转换算法存储在中介器内、或管理器内,或者存储在可操作性地耦接至中介器或管理器的其它硬件和/或软件内。所述算法中的任一算法或两个算法也可以致使文件名存储在中介器内。对中介器物理上所位于的地方没有任何限制。然而,优选地,中介器被配置成与主机或计算机通信,所述计算机能够与主机通信,所述主机优选地位于远离中介器的地方上。中介器也被配置成直接或间接(例如,借助管理器)与记录介质(例如,存储有所述组编码数据的非高速缓存器介质)通信,所述记录介质可选地远离中介器、任何管理器和主机。正如上文所提到的,中介器允许识别文件名的用户来从非高速缓存器存储介质检索所述组编码数据。

[0145] 高速缓存器

[0146] 正如上文所阐明的,在一些实施方案中,在接收原始数据后,本发明的方法可以致使自动向主机返回接收的确认。在一个QoS(服务质量)协议中,数据文件是借助I/O来接收,

并立即发送给L1高速缓存器。在接收后,借助I/O而从L1高速缓存器发回确认。数据文件可以从L1高速缓存器发送给L2高速缓存器,所述L2高速缓存器将确认回传给L1高速缓存器。L2高速缓存器也可以将数据文件发送给非高速缓存器介质(NCM),以便长期存储。NCM又可以将确认发回给L2高速缓存器。

[0147] 在一些实施方案中,中介器可以存在于L1高速缓存器内的堆积(动态分配存储器)中,或者可操作性地耦接至所述堆积。或者,中介器可以存在于卡片内,或者作为L2高速缓存器的一部分或可操作性地耦接至L2高速缓存器。

[0148] 正如本领域普通技术人员所知悉的,将中介器放置在L1与L2中的决策将受到多个因素的影响,如存储数据的使用频次。因此,L1高速缓存器用来存储系统或最终用户频繁使用的数据,而L2高速缓存器可以用于略微频繁存取的数据。

[0149] 在另一QoS协议中,借助I/O,数据文件由L1高速缓存器加以接收。数据文件从L1高速缓冲器传递给L2高速缓存器和NCM。L2高速缓存器和NCM中的每一者都将确认发送给L1高速缓存器。在从L2高速缓存器和NCM中的一者或两者接收确认之前或之后,L1高速缓存器借助I/O来发送确认。

[0150] 文件相关性

[0151] 在本发明的各种实施方案中,主机将会了解有待存储在第一存储地址上的每个文件。第一存储地址可以由主机存储在扇区图中,并且对应于LUN。第一存储地址也可以包括对应于文件的单元、扇区或区块的起点并且暗示地或明确地包括其末端。第一存储地址将对应于主机认为文件在存储装置或存储区域网络内所处于的位置。主机将使用这个第一地址来跟踪其存储文档或文件,并且对它们进行检索。第一存储地址是虚拟地址,也就是,其并不对应于数据实际存储的位置。

[0152] 正如本领域普通技术人员认识到的,可使用多个方法和系统,其中主机生成第一存储地址,并将其与SCSI命令以及可选的是关联扇区或LBA编号一起发送给本发明的系统。中介器可以将文件名、主机认为是作为从主机接收的文件的位置和文件的存储大小的信息(也就是原始数据)以及任何页眉或页脚数据与第二存储地址相互关联,所述第二存储地址是数据的实际存储地址,这个地址可以含有可供重新创建文件的信息,但是所述信息已经借助(例如)散列值算法、比特标记符表和频次转换器中的一者或者进行了转换。或者,中介器可以只存储文件名,并且可选地,中介器可以不接收用于文件的第一存储地址。正如上文所提到的,因为存储地址是基于数据的线性组织,所以它们可以通过只列举第一用户感知LBA或明确地列举所有位置,而暗示地含有存储信息的大小。

[0153] 尽管上面的段落描述了主机将会提供其认为是第一存储地址的信息,但是所述信息可以由另一实体生成,所述实体是供主机直接或间接与中介器通信的通道、处于主机内或可操作性地耦接至主机的模块,或者是处于中介器和/或管理器内或可操作性地耦接至中介器和/或管理器的模块。正如本领域普通技术人员认识到的,识别存储装置上数据文件位置的存储信息可以称为指针。对于文件而言,指针可以从与NCM上用户认为数据所处的位置不同的位置(例如,区块)来指引数据的检索。

[0154] 可选地,相关人员可以使文件与文件类型关联。文件类型将会指引数据的接收者来知悉应使用哪个操作系统来打开所述数据。在一些实施方案中,与文件类型的关联是在发起器或客户端或主机处完成的。

[0155] 预留磁道

[0156] 正如上文所提到的,中介器可以包括第一组预留磁道(R_1)和第二组预留磁道(R_2)。在一些实施方案中,第二组预留磁道(R_2)是第一组预留磁道(R_1)的复本。另外,在一些实施方案中,相关人员可以使用第二组预留磁道(R_2)来检查第一组预留磁道(R_1)中的错误。

[0157] R_1 可以被配置来充当主机发起的中心点。因此,主机可以选择参数来发送给 R_1 。中介器可以直接从主机或者间接地借助管理器来接收这个信息。 R_2 优选的是从不暴露给主机。因此,只有中介器本身或管理器可以致使将信息存储在 R_2 中。 R_1 和 R_2 中的每一者可以(例如)含有16个扇区,并且填充有真实数据,如主机修改符。按照惯例,编号可以从0开始。因此, R_1 可以(例如)含有扇区(或磁道)0至15,而 R_2 可以含有扇区(或磁道)16至31。然而,中介器可以经过构建,从而允许将 R_1 和 R_2 中的每一者扩展成超过16个磁道的初始大小。

[0158] 在一些实施方案中, R_1 含有独特的预留扇区信息和分区信息。在分区信息内,相关人员可以存储文件系统信息。

[0159] 举一个非限制性示例,并且正如本领域普通技术人员所意识到的,当利用NFTS文件系统格式化某个卷时,相关人员会创建元数据文件,如\$MFT(主文件表)、\$Bitmap、\$Log文件以及其他文件。这个元数据含有与NFTS卷上所有文件和文件夹相关的信息。NFTS卷上的第一信息可以是分区启动扇区(\$Boot元数据文件)信息,并且可以位于扇区0处。这个文件可以描述基本的NTFS卷信息和主要元数据文件\$MFT的位置。

[0160] 格式化程序会为\$Boot元数据文件分配前16个扇区。第一扇区为具有引导程序代码的启动扇区,并且随后的15个扇区为启动扇区的IPL(初始程序加载程序)。

[0161] 除了 R_1 和 R_2 磁道之外,中介器还可以存储额外的元数据。这个元数据可以(例如)对应于允许运行自动精简配置策略的信息,所述自动精简配置策略对应于允许装置看上去比实际可用资源具有更多物理资源的可视化技术,并且,这个元数据可以(例如)包含在 R_2 之后的八个磁道中,所述八个磁道是磁道32至39。元数据也可以提供有多个特征,如LUN QoS、VM和WORM。

[0162] 在一些实施方案中,系统可以含有SAN索引器。SAN索引器可以检查 R_1 和 R_2 中存在什么信息,并且提取所述信息。这个信息可以放入到数据库中,可以通过(例如)文本搜索而轻易地搜索所述数据库。

[0163] 比特字段

[0164] 中介器也可以包括或含有对应于比特字段的信息。比特字段含有指示数据在存储介质内物理上存储的位置的信息,并且如果元数据位于磁道32至39中,那么比特字段的扇区编号可以开始于磁道40。主机的文件名与数据的位置之间的相关性也是存储在中介器的比特字段内。因此,比特字段可以包括扇区图、基本上由扇区图组成或者由扇区图组成。来自中介器的比特字段成分的这个信息可以用来确定任何装置上的实际空间节约。举例而言,所节约空间的百分比=1-[$(\text{实际使用的空间}) / (\text{主机所映射的空间})$]。在一些实施方案中,根据本发明的方法来转换数据而节约的空间至少是50%、至少是60%、至少是70%或至少是80%。这些节约可以针对每个文件或者对于装置上的所有文件取平均值。

[0165] 作为惯例,优选地中介器并不位于存储文件数据的磁盘或记录介质上。另外,优选地中介器只需要对应磁盘或记录介质的总内存的约0.1%至0.2%。

[0166] 除了从空间节约方面提供经济价值之外,本发明的各种实施方案在寻求保护数据

完整性时为提高效率打开了方便之门。因此,本发明的各种实施方案为备份数据提供新颖且非显而易见的技术。

[0167] 因为在许多实施方案中,存储文件将会小于从主机所接收的原始数据文件,所以需要更少的存储空间来用于所述存储文件。因此,重新创建文件所需要的数据可以存储在比主机察觉到正在使用的位置以及第一存储地址所表明的位置更小的位置中。文件所存储的实际位置可以称为第二存储地址。因此,对于每个文件而言,将会存在第一存储地址和第二存储地址,其中第一存储地址是主机认为文件所存储的地方,而第二存储地址则是编码文件实际存储的地方。

[0168] 可能的是,对于给定文件(其可对应于一个或多个区块)而言,第一存储地址和第二存储地址位于存储装置内的同一区块处或者一组或多组重叠区块处。然而,优选地,对于文件中的至少一个文件、至少50%、至少60%、至少70%、至少80%、至少90%或100%而言,第一存储地址和第二存储地址内不存在区块重叠。另外,即使主机与中介器察觉到相同的存储地址,但是当数据编码时,主机也无法在未首先解码数据的情况下重新创建文件。在一些实施方案中,主机并不知道代码,且因此不能够解码存储数据。

[0169] 在一些实施方案中,中介器可以接收对应于文件的小盘,并将它们临时存储在L1或L2高速缓存器中。如果存在L2高速缓存器,那么L2高速缓存器便可以向主机确认接收,并且可选地向主机提供或确认第一存储地址。正如本领域普通技术人员会认识到的,第一存储地址接收和传输的确认可以在第二存储地址上进行存储之前完成,并且,如果执行编码,那么便在编码之前或之后完成。无论何时发送确认,文件的实际存储地址与虚拟存储地址的相关性,优选的是在比特字段内跟踪。

[0170] 数据的备份

[0171] 在某些实施方案中,相关人员可以使用两个中介器来促进数据备份。举例而言,在第一中介器中,相关人员可以使存储在第一记录介质上的第一扇区或第一扇区群集中的数据文件与文件名相互关联。正如上文所描述的,第一中介器被配置成允许识别文件名的用户或实体来从记录介质检索数据文件。

[0172] 可以执行数据保护协议,从而生成第二中介器。第二中介器将会是第一中介器在时间T1上的精确复本。因此,在时间T1上,第一中介器和第二中介器都将指向第一记录介质上的相同扇区或扇区群集(以及其中的位置)。

[0173] 在时间T1之后,例如在时间T2上,主机可以试图更新存储在给定扇区或扇区群集上的给定位置中的文件。主机将不会更改第一存储地址上所存储的数据。不是致使改写扇区或扇区群集上的信息,而是第一中介器可以致使将新的信息写入到对应于不同扇区或扇区群集中的位置的第三存储地址,并使文件名以及第一存储地址与这个新存储地址相互关联。

[0174] 因此,即使主机认为正在特定存储地址上重写信息,但是第一中介器也将指向新的扇区或扇区群集。因此,主机将不需要针对扇区群集来更新其地址。

[0175] 第二中介器将不会被更新,并且其将继续使文件名与文件所存储的第一位置相互关联。这样使用两个中介器将会允许相关人员提供数据存在于时间T1上时的快照,而不会使得主机需要更新其文件系统来指示存在于时间T1和T2上时的文件正在被存储。因此,快照会锁定时间T1上存储的所有数据文件,并且防止任何人删除或改写这些物理文件。然而,

如果主机想要修正这些文件,那么主机可以在持有正在修正文件的印象下工作,事实上这时会存储新的文件。这种方法是结合扇区和扇区群集来描述的。然而,所述方法也会与并非布置在扇区或扇区群集中的非高速缓存器介质一同使用。举例而言,它们可以由LUN中的LBA加以组织。

[0176] 正如上文所表明的,这种方法可以由包括第一中介器、第二中介器和非高速缓存器存储介质的系统来实施。第一中介器、第二中介器和记录介质中的每一者可以存储在单独装置上或者由单独装置形成,所述单独装置包括非暂时性介质、基本上由非暂时性介质组成或者由非暂时性介质组成。前述系统列举同一记录介质的不同扇区的使用,但是也可以通过写入到不同记录介质的不同扇区来使用。另外,在系统内,中介器和记录介质可操作性地彼此耦接,并且可选地耦接至一个或多个计算机或CPU,所述一个或多个计算机或CPU存储指令以便致使它们执行它们的预期功能,并在网络上借助一个或多个门户向一个或多个主机进行通信。此外,尽管这个实施方案是结合两个中介器的使用来描述的,但是相关人员也可以使用同一中介器的两个区段来实施系统,而不是使用两个单独中介器。

[0177] 系统可以进一步配置有锁定模块。锁定模块可以防止在从某个时间起已经写入的一个或多个区块上进行修正、重写或删除。锁定模块也可以经过设计而容许写入新区块以及修正尚未锁定的这些新区块。因此,锁定模块可以配置成允许主机、用户或系统管理员来选择从某个时间起已经写入的某些区块,或者选择从某个时间起已经写入的所有区块都不进行重写。

[0178] 此外,可以存在选择模块,所述选择模块默认借助第一中介器来发送所有的文件检索以及修正、重写或删除请求。选择模块也可以被配置成允许恢复主机可能认为从应用锁定技术的时间起是一个或多个文件的更老版本的文件。可选地,整个选择模块的访问权限可以限制给具有授权的人员,例如,系统管理员。

[0179] 用于备份数据的前述系统是在两个中介器的情境中加以描述的。然而,也可以使用两个以上的中介器来捕获存储文件或文件版本的历史数据。举例而言,可以使用至少三个、至少四个、至少五个、至少十个中介器等等。另外,主机可以使得中介器在有规律间隔上(例如,每周、每月、每个季度或每年)或不规律间隔上(例如根据需要)获取快照。

[0180] 根据另一种用于备份数据的方法,可以制作非高速缓存器介质的克隆形式。在这种方法中,在第一中介器中,相关人员使多个文件名与存储在非高速缓存器存储介质上的数据的多个位置相互关联。第一中介器被配置成允许识别特定文件名的用户来从第一非高速缓存器存储介质检索对应于所述特定文件名的数据文件。特定文件的一部分或整个特定文件可以存储在第一扇区或扇区群集中。

[0181] 相关人员可以将所述多个数据文件(或第一非高速缓存器存储介质的所有数据文件)复制到第二非高速缓存器存储介质和第二中介器。第二中介器是第一中介器在时间T1上的复本,并且可操作性地耦接至第二非高速缓存器存储介质。在时间T1之后的时间T2上,系统可以保存第一非高速缓存器存储介质上的所述第一扇区或扇区群集中所存储的数据文件的修正文件。然而,不会对第二非高速缓存器介质上的对应位置做出任何更改。用户在时间T2之后请求文件时,他或她将会仔细查看第一中介器并检索文件的最近存储版本。然而,系统管理员会对存储在第二非高速缓存器介质上的早先版本具有存取权限,并且可以通过仔细查看第二中介器来检索早先版本。

[0182] 这种方法可以由包括第一中介器、第二中介器、第一非高速缓存器存储介质以及第二非高速缓存器存储介质的系统来实施。用于存储数据文件的第一中介器、第二中介器以及第一和第二记录介质中的每一者可以存储在单独装置上,所述单独装置包括非暂时性介质、基本上由非暂时性介质组成或者由非暂时性介质组成。另外,第一中介器使从主机获得的文件名与第一记录介质的第一LUN相互关联,并且第二中介器使同一文件名与第二记录介质上的第二LUN相互关联。在一些实施方案中,存储在第一非高速缓存器介质中的最近的文件与第二非高速缓存器介质内的遗留文件具有相同的LUN。

[0183] 数据检索

[0184] 根据本发明的方法中的任何方法,以转换形式存储的任何数据都能够将其返回给主机之前进行检索和解码。通过使用允许检索转换数据、存取上述参考表或频次转换器以及转回成统一比特串和小盘的一个或多个算法,便可以为主机重新创建文件。举一个非限制性示例,数据可能已经按照某个格式进行了转换和存储,所述格式含有一个标记符结束位置的指示,例如,使用独特的比特串或者通过使用统一大小的标记符。

[0185] 所存储数据的检索可以借助现在已知的或者将要知悉的并且本领域普通技术人员了解到会与本发明结合使用的过程和技术。可选地,管理器协调文件的存储和检索。在一些实施方案中,来自NCM的数据(如标记符)是借助并行处理来检索。

[0186] 在一个方法中,相关人员从访问记录介质开始。记录介质以某个顺序存储多个标记符,并且相关人员可以根据这些标记符来重新创建文件。访问可以由请求文件的检索并将请求传输给存储区域网络的主机发起,或者由存储区域网络的管理员发起。

[0187] 在从中介器(若存在)识别的位置上来从记录介质检索数据之后,如果数据已经转换,那么相关人员便将所述多个标记符(或已经借助频次转换器进行转换的数据)转译成可以用来形成有待于转回成I/O流或I/O流片段单元的小盘或散列值的比特。标记符可以经过存储,从而使得每个标记符对应于小盘或者每个标记符对应于子单元,并且多个子单元可以组合来形成小盘。在存储格式中,以容许重新创建期望文档或文件的方式、按照允许重新创建小盘(或散列值)内的比特和重新创建小盘(或散列值)顺序的顺序来布置标记符。

[0188] 如果数据经过转换,那么为了将标记符转译成小盘,相关人员可以存取比特标记符表或频次转换器。在比特标记符表或频次转换器内,可以存在与每个独特比特串关联的,或者处于文件内的每个独特比特串内的独特标记符。如果所述表是组织在类似于表II的格式中,那么在转译之后,可以添加0,从而使得每个子单元和小盘具有相同大小。在解码时,相关人员以与其用于编码的方式相反的方式来使用比特标记符表或频次转换器。可选地,不是使用同一个表以及颠倒输入和输出,而是相关人员可以使用单独的表。

[0189] 就其它实施方案而言,每个小盘的长度可以是N个比特,其中N为大于1的整数,并且每个子单元的长度可以是A个比特,其中A为整数。为了将标记符转译成小盘,相关人员可以存取比特标记符表或频次转换器。

[0190] 在形成小盘之后,相关人员将会具有对应于二进制数据的输出,可以根据所述二进制数据来重新组建文档。可选地,相关人员可以使文件与文件类型关联。举例而言,主机可以跟踪MIME转译程序,并在返回后将其与文件重新关联。文件类型将会指引数据的接收者来知悉应使用哪个操作系统打开所述数据。正如本领域普通技术人员认识到的,存储区域网络不需要跟踪文件类型,并且在一些实施方案中也并未跟踪文件类型。

[0191] 在一些实施方案中,形成小盘之后,相关人员将会具有对应于二进制数据的输出,可以根据所述二进制数据来重新组建文件。在其它实施方案中,转换数据对应于散列值,并且散列值必须首先用来创建I/O流的片段单元或I/O流本身。

[0192] 相对于截断数据而言,甚至在不需要利用比特标记符表或频次转换器的情况下,相关人员也可以检索数据。相关人员可以通过访问记录介质来检索数据,其中所述记录介质将多个数据单元存储在多个位置中,其中每个数据单元含有多个比特并且数据单元的最大大小为第一比特数目,至少一个数据单元含有第二比特数目,其中第二比特数目小于第一比特数目。

[0193] 接着,相关人员可以检索数据单元,并且在长度小于N个比特的任何数据单元的末端添加一个或多个比特(例如0),以便生成一组对应于数据单元的小盘,其中每个小盘含有相同数目的比特;以及,以与数据单元的顺序相对应的顺序来生成包括所述组小盘的输出。如果截断数据是通过移除0来形成的,那么在检索数据时,相关人员会将0加回。另外,如果存储的数据单元为小盘的子单元,那么系统可以首先将0加回到截断子单元,从而生成具有统一大小的子单元,且然后组合子单元来形成小盘。

[0194] 当使用查找表时,优选地,所述查找表是存储在计算装置的存储器中。在一些实施方案中,查找表是静态的,并且标记符是预定的。因此,当随着时间的推移来存储一种或多种不同文档类型的多个文档时,可以使用同一个表。可选地,这个表可以存储在主机的位置处,或者作为存储区域网络的一部分加以存储。

[0195] 从主机接收数据

[0196] 为了进一步说明各种实施方案并提供情境,下文将参考相关人员可以使用的特定硬件,所述硬件可以组合来形成系统以便实施本发明的方法。所述硬件可以被配置成容许在从主机接收一个或多个数据文件时,通过一个或多个处理器将上述过程自动并进行控制。

[0197] 在一些实施方案中,主机可以按照任何方式在第一位置处生成文档和计算机文件。文档将由主机的操作系统生成,并且经过组织而由主机的文件系统进行存储。主机的操作系统可以将文件名本地存储在其存储器中。本发明并不受限于主机所使用的操作系统或文件系统的类型。举一个非限制性示例,主机可以包括网络内的计算机或一组计算机,所述计算机或所述组计算机具有下述硬件部件中的一个或多个部件:存储器、储存器、输入装置、输出装置、图形用户界面、一个或多个通信门户以及中央处理单元。

[0198] 在所述第一位置处,SAP执行协议而用以存储与文档或文件相互关联的数据。SAP将数据格式化成大小为(例如)4K的I/O流或小盘。

[0199] 数据可以在SAN上发送给具有一个或多个模块的计算机或网络,或者发送给被配置来接收数据的计算机或一组计算机。这个接收计算机可以包括下述硬件部件中的一个或多个部件:存储器、储存器、输入装置、输出装置、图形用户界面、中央处理单元以及被配置成允许在本地和/或在网络上与一个或多个主机和一个或多个存储装置进行信息通信的一个或多个通信门户。

[0200] 另外,可以存在将可执行计算机代码存储于硬件、软件或硬件与软件的组合上的计算机程序产品。所述计算机程序产品可以划分为一个或多个被配置来执行本发明方法的模块或者能够与所述一个或多个模块通信,并且可以存储在一个或多个非暂时性介质中。

[0201] 举例而言,可以存在1级(L1)高速缓存器和2级(L2)高速缓存器。正如本领域普通技术人员所知晓的,高速缓存器技术的使用在传统上容许相关人员提高数据存储的效率。在本发明中,举例而言,数据可以在SAN上发送给高速缓存器,并且数据可以在存取散列函数算法之前、在查询比特标记符表之前、在查询频次转换器之前以及在将比特截断之前,和/或在查询散列函数算法之后、在查询比特标记符表之后、在查询频次转换器之后以及在将比特截断之后,发送给高速缓存器。

[0202] 假设扇区大小为512B,那么对于大小为4K的每个小盘而言,主机将期望使用8个存储扇区。

[0203] 系统

[0204] 本发明的各种实施方案提供数据存储和检索系统。在一个实施方案中,所述系统包括非高速缓存器数据存储介质、中介器和管理器。这些元件之间的通信以及可选地与发起器之间的通信可以在有线网络、无线网络或其组合上进行。

[0205] 非高速缓存器数据存储介质可以(例如)包括一个或多个磁盘或固态驱动器、基本上由所述一个或多个磁盘或固态驱动器组成或者由所述一个或多个磁盘或固态驱动器组成。在使用时,与尚未根据本发明方法中的一个或多个方法进行处理的文件相比而言,非高速缓存器数据存储介质可以在节约至少50%、至少60%、至少70%或至少80%的空间的情况下存储文件。

[0206] 中介器可以包括四组磁道、基本上由这四组磁道组成或者由这四组磁道组成:第一组磁道、第二组磁道、第三组磁道和第四组磁道。中介器优选地存储在非暂时性介质上,并且位于远离非高速缓存器数据存储介质的地方。因此,中介器和非高速缓存器数据存储介质优选地不是同一装置的部分。

[0207] 系统也可以含有管理器。管理器可以借助中介器来对数据的接收、处理存储以及检索和传输提供控制。因此,优选地,管理器可操作性地耦接至主机和中介器,并且可选的是可操作性地耦接至非高速缓存器数据存储介质。此外,在一些实施方案中,管理器位于远离中介器、非高速缓存器介质和主机中的每一者的地方。

[0208] 管理器可以被配置来执行下述特征中的一个或多个特征:(a)将包括文件系统信息、可启动性信息和分区信息中的一个或多个信息的数据存储在中介器的第一组磁道中;(b)将元数据存储在中介器的第三组磁道中;(c)将一个或多个文件存储在非高速缓存器介质上,其中所述一个或多个文件存储在不具有文件系统信息、可启动性信息和分区信息中的任何信息的非高速缓存器介质上(因此在一些实施方案中,只有原始数据处于非高速缓存器介质上);(d)将非高速缓存器介质中每个文件的位置存储在中介器的第四组磁道中;以及(e)将非高速缓存器介质中每个文件的位置与文件的主机名的相关性存储在中介器上。优选地,非高速缓存器介质中每个文件的位置与文件的主机名的相关性是存储在第四组磁道中,所述第四组磁道对应于比特字段。另外,管理器可以包括或可操作性地耦接至计算机程序产品,所述计算机程序产品含有能够执行本发明方法中的一个或多个方法的可执行指令。

[0209] 在本发明的各种系统中,根据存储在计算机程序产品中的指令,SAN可以存取比特标记符表或频次转换器。这些资源使比特标记符与子单元中的每个子单元相互关联,并生成输出。因为即便不是全部,也会有大部分比特标记符在大小上小于子单元,所以所述输出

便为比从主机接收的输入文件更小的数据文件。因此,虽然从主机接收的文件可以具有大小R,但是SAN所保存的实际数据可以是S,其中R>S。优选地,R至少是S的两倍大,并且更优选地,R至少是S的三倍大。

[0210] SAN取得输出文件并将其存储在非暂时性存储介质(例如,非高速缓存器介质)中。优选地,SAN使所存储的文件与从主机接收的文件相互关联,从而使得主机可以检索文件。

[0211] 出于进一步说明的目的,可以参考图1,其展示用于实施本发明方法的系统。在系统10中,主机1将文件传输给存储区域网络6,所述存储区域网络6含有可操作性地耦接至存储器4的处理器3。可选地,存储区域网络确认接收并回复给主机。

[0212] 在存储器内存储有计算机程序产品,所述计算机程序产品经过设计而取得小盘并将其中的数据划分为子单元。存储器也可以含有或可操作性地耦接至参考表5。所述表含有针对子单元中的一个或多个子单元的比特标记符或频次转换器或相关性文件,并且计算机程序产品创建新数据文件,所述新数据文件含有取代原始子单元的标记符中的一个或多个标记符。

[0213] 处理器接着致使将比特标记符存储在记录介质(如非高速缓存器介质)上,所述记录介质可以(例如)是磁盘2。在一些实施方案中,最初所有标记符都具有相同大小。然而,在一些实施方案中,在存储这些标记符之前,一个或多个标记符、优选的是至少25%、至少50%或至少75%在存储之前被截断。

[0214] 图2展示具有中介器70的系统100,所述中介器70含有R₁40和R₂50以及用于比特字段60和元数据文件30的空间。中介器的表示形式仅仅出于说明性目的,并且不会对中介器的结构或其内的组织施加任何限制。也在图中展示的是非高速缓存器介质(NCM)20。非高速缓存器介质展示为处于中介器的附近,但是它们是单独的结构。在一些实施方案中,中介器位于与NCM相分离的装置上,并且本发明的装置中的一个或多个装置是便携式的。标记符或其它转换原始数据将存储在NCM上。

[0215] 图3展示另一系统200。在这个系统中,发起器(Iⁿ)270将小盘传输给高速缓存管理器230,所述管理器230可选地会安排数据文件的编码并将数据文件传输给中介器210。主机的示例包括但不限于运行Microsoft Windows Server和Desktop、Apple OS X、Linux RHEL和SUSE Oracle Solaris、IBM AIX、HP UX以及VM ESX和ESXi的计算机或计算机网络。对于数据文件的信息最初发送给R₁240,R₁240之前填写有发起器所定义的参数。中介器可以通过使用比特标记符表或频次转换器(未图示)而亲自转译信息,或者中介器可以与远程编码器(其也可以称为远程转换器,而且未图示)通信,并且中介器会将从主机接收的文件名的复本存储在R₁内以及R₂250内。在数据已经转换并且更小大小的文件已经创建之后,在比特字段260的扇区图内记录文件将在磁盘220中所存储的某个位置或某些位置。编码数据可以存储在位置285处。在数据编码之前或者代替数据编码,可以存取散列值算法,并且校验和可以用于存储或转换。

[0216] 图4展示另一系统300,所述系统300是图3系统的实施方案的变型并且提供存储的备份。在这个系统中,发起器370将小盘传输给高速缓存管理器330,高速缓存管理器330将信息转发给第一中介器310,第一中介器310含有数据来修正针对图3所发送的同一文件。在接收修正文件之前或者在接收修正文件之后,但是在将修正文件存储于非高速缓存器介质中之前,从第一中介器310创建第二中介器380。第二中介器是第一中介器在其创建时间上

的精确复本，并且在这个时间上，对于文件名而言，第二中介器指向非高速缓存器介质320内的同一扇区(或扇区群集)385。

[0217] 第一修正文件是在第一中介器的R₁340处接收。第一中介器将再次通过使用比特标记符表或频次转换器(未图示)来转译信息，或者与远程编码器通信。中介器会继续将从主机接收的文件名的复本存储在R₁内以及R₂350内。在数据已经转换并且更小大小的文件已经创建之后，在第一中介器的比特字段360的扇区图内记录文件将在磁盘320中所存储的位置。然而，修正文件将会存储在不同的扇区395中。因此，对第一中介器的更改不会针对第二中介器做出。

[0218] 主机默认与第一中介器通信。因此，当主机想要从储存器重新调用文件时，第一中介器将会从扇区395调回数据。如果主机或系统管理员希望获取数据的前一个版本，那么它可以将文件名提交给第二中介器，所述第二中介器会查看扇区385。

[0219] 根据本发明的另一实施方案，之前已经将元数据(例如，操作系统信息、可启动性信息、分区信息、文档类型信息、QoS信息等)提供给本发明系统的发起器，将对应于文档的比特发送给高速缓存管理器。所述比特可以(例如)组织在小盘中。

[0220] 高速缓存管理器可以将信息发送给L1高速缓存器、L2高速缓存器和中介器。高速缓存管理器也可以向发起器发送接收确认。

[0221] 上面可能已经存储有相关元数据的中介器，将对应于文档的比特发送给转换器，所述转换器将比特转换成编码信息。计算器也可以与转换器关联或者作为转换器的一部分，所述计算器确定转换比特的大小。可以(例如)通过使用比特标记符表或频次转换器来进行转换。

[0222] 转换器随后可以将转换文件的大小告知给中介器，并且中介器可以确定转换文件将在非高速缓存器介质中所存储的位置。紧接着这个步骤之后，中介器可以致使在所述位置上进行存储。

[0223] 本说明书中所描述的各种实施方案的特征中的任何特征，都可以与结合所公开的任何其它实施方案来描述的特征联合使用，除非另有说明。因此，结合各种或特定实施方案来描述的特征不应解释为不适于结合本文所公开的其它实施方案，除非上下文中明确陈述或暗含此类排他性。

[0224] 示例：

[0225] 示例1：比特标记符表(预示性的)

[0226] 在参考定位符表内，每个独特标记符被识别为对应于独特比特串。所述表可以按照普遍熟知的或将要知悉的用于表存储的，并且允许计算机算法来获取指配给每个输入的输出的任何格式来存储。

[0227] 下面的表I提供来自比特标记符表的节选的示例，其中子单元的长度为8个比特。

[0228] 表I

[0229]

比特标记符(所存储的)	子单元= 8 比特 (输入)
0101	00000001
1011	00000010
1100	00000011
1000	00000100
1010	00000101
11111101	11111101

[0230] 举例而言,并且在使用表I中所识别的子单元的情况下,如果输入为00000101 00000100 00000101 00000101 00000001,那么输出便为:1010 1000 1010 1010 0101。当比特标记符输出小于子单元输入时,所述比特标记符输出将在存储介质上占据更少的空间,并且因而节省存储比特所需要的存储空间和时间。

[0231] 正如本领域普通技术人员会认识到的,在给定比特标记符表(如经过节选而产生表I)中,如果要使用所有比特组合,那么便需要 2^N 个条目,其中N对应于子单元内的比特数目。当存在8个比特时,便需要256个条目。当子单元中存在16个比特时,相关人员需要 2^{16} 个条目,这等于是65536个条目。当子单元中存在32个比特时,相关人员需要 2^{32} 个条目,这等于是4294967296个条目。如果相关人员知悉某些比特串将不在文件中使用,那么所述表便可以从最小的标记符开始来分配标记符。

[0232] 示例2:针对预处理子单元的比特标记符表(预示性的)

[0233] 因为随着子单元大小变大,表也会变得更繁琐,所以在一些实施方案中,表可以经过配置,从而使得来自子单元列的一个末端的所有0都是缺失的,并且在存取所述表之前,从每个子单元的所述末端移除所有0。因此,不是配置可供节选成表I的表,而是可以配置可供节选成表II的表。

[0234] 表II

[0235]

比特标记符(输出)	预处理子单元
0101	00000001
1011	0000001
1100	00000011
1000	000001
1010	00000101
11111101	11111101

[0236] 正如相关人员可以看出的,在第二和第四行中,在子单元预处理之后,它们会具有少于八个比特。然而,从主机接收的原始数据中的实际子单元都具有八个比特。因为实施所述方法的系统可以经过设计而理解数字的不存在便意味着是0,而且数字的所有不存在都是处于任何截断子单元的相同末端,所以相关人员可以使用占据更少空间并且仍然保持向独特子单元指配独特标记符的能力的表。因此,所述方法允许系统将00000001(七个0和一个1)与00000001(六个0和一个1)解译为是不同的。

[0237] 为了实施这个方法,相关人员可以将每个子单元(或如果不使用子单元便是将每个小盘)视为具有第一末端和第二末端。第一末端可以是比特串的右侧或左侧,而第二末端则是相反的一侧。出于说明性目的,相关人员可以认为第一末端是最左侧数字而第二末端是最右侧数字。在这个方法下,相关人员随后分析每个小盘的每个子单元内的一个或多个

比特,以便确定第二末端处的比特是否具有数值0。这个步骤可以称为预处理,并且在子单元预处理之后,它们会出现在表II的右列。如果第二末端处的比特具有数值0,那么所述方法便可以移除第二末端处的比特以及具有数值0且与这个比特形成连续比特串的所有比特,从而针对起初在第二末端处具有0的任何子单元来形成修正子单元(表中的预处理子单元)。

[0238] 相关人员可以使用计算机算法,所述计算机算法核查每个子单元以便确定在第二末端处是否存在0,并且如果存在,便移除所述0来形成预处理子单元,所述预处理子单元也可以称为修正子单元,其中修正的第二末端处在邻近于子单元第二末端的位置上。接着,所述算法核查修正子单元以便确定在其现在修正的第二末端处是否存在0,并且如果存在,便移除所述0来形成另外的修正第二末端。在这个方法中,修正的第二末端会是之前邻近于第二末端处的比特的位置。任何另外的修正第二末端都会是远离子单元第二末端的两个或两个以上的地方。因此,术语“修正的”意味着缩短或截断的第二末端。所述算法可以对修正单元重复这个方法,直到生成在其第二末端处具有1的缩短小盘。

[0239] 正如本领域普通技术人员会认识到的,前述方法被描述成是通过如下方式加以应用的:从第二末端移除0,直到1处于修正的第二末端或另外的修正第二末端处。所述方法可以反向设计,从而使得系统从第二末端移除1,直到0处于修正的第二末端或另外的修正第二末端处。另外,在利用本公开的情况下,本领域普通技术人员可以从第一末端移除比特而不是从第二末端来移除,并使用所创建的表来将这些修正子单元转换成比特标记符。

[0240] 示例3:频次互换(预示性的)

[0241] 基于经验性分析,相关人员可以确定从特定主机接收的一类文档或一组文档内或已经在给定时间范围内(例如,过去的一年或过去的两年)接收的一组文档内的每个子单元的频次。利用这个信息,而不是查看如表I或II所示出的、子单元按照数字顺序组织的表,相关人员可以查看频次转换器,其中更小的比特标记符是与预期最有可能出现在从特定主机接收的文件内、一类文档内或一组文档内的子单元关联。因此,在频次转换器内,标记符具有多个不同大小,并且具有更小大小的标记符是与更高频次的子单元相互关联。

[0242] 表III:频次转换器

	比特标记符(输出)	频次	子单元=8 比特(输入)
[0243]	0101	16%	00000001
	1000	15%	00000010
	11011	10%	00000011
	10011101	0.00001%	00000100
	10111110	0.00001%	00000101
[0244]	1100	15%	11111101

[0245] 表III是来自与表I使用相同子单元的频次转换器的节选的示例。然而,相关人员将会注意到,比特标记符并不是依次指配的,并且实际上更大的比特标记符指配给更低频次的子单元。正如表所示出的,指配给子单元00000011的标记符比指配给子单元00000001的标记符大25%,并且对于子单元11111101而言,尽管具有较高数值,但是其仍然接收更小的比特标记符,因为所述子单元在从特定主机接收的这些类型文件中更为频繁地出现。因此,如果相关人员使用表I并且子单元11111101出现在10000个地方,那么所述子单元将会

对于111111010000个比特。然而,如果相关人员使用表III,那么只需要11000000个比特来用于相同信息的存储目的。尽管未展示在这个方法中,但是子单元也可以经过预处理来从一个末端或另一末端移除0,并且所述表可以被设计成含有关联的截断子单元。

[0246] 正如上文所提到的，频次转换器可以基于被视为代表可能从一个或多个主机接收的数据的一组文件的分析来生成。在一些实施方案中，处理信息的算法可以执行其自身的品质控制，并且将用于来自给定时期的文档的子单元的实际频次，与频次转换器中标记符的分配所基于的那些频次进行比较。在使用统计分析的情况下，所述算法随后可以确定，对于将来的使用而言，是否应创建新表来重新安排标记符如何与子单元关联。正如本领域普通技术人员会认识到的，表III是频次转换器的简化节选。然而，在实践中，相关人员可以选择十六进制系统，从而获得相关性。另外，是为了方便读者而包括表所基于的频次的列举，并且这些频次不需要包括在本发明的各种实施方案所存取的表中。

[0247] 示例4:中介器中空间的分配(预示性的)

[0248] 在大小为1MB的假设记录介质中,本领域普通技术人员可以按照如下方式来映射扇区:

[0249] 1MB记录介质具有1024000个字节,这对应于250个扇区。 $(1024000/4096=250)$ 。记录介质的几何特性可以按照如下方式汇总:容量 = $(c*h*spt*ss)$,其中:

[0250] c(柱面数目)=7;

[0251] h(磁头数目)=255;

[0252] spt(每个磁道的扇区数)=63;以及

[0253] ss(扇区大小,单位:字节)=4096。

[0254] 在中介器内，扇区可以按照如下方式进行分配：

[0255] 表IV

地址:	实际非高速缓存器介质 LBA
0-15	中介器<<预留 1>>“启动扇区 0”+15
16-31	中介器位置<<预留 2>> Sys_ 仅限内部
32-35	中介器_元数据
36	映射数据“LBA-nnnnnnnnnnnnnn”
37	映射数据“LBA-nnnnnnnnnnnnnn”
...	映射数据“LBA-nnnnnnnnnnnnnn”
250	映射数据“LBA-nnnnnnnnnnnnnn”

[0257] 示例5:空间节约

[0258] 本发明的系统接收具有4096个字节的I/O流中的42500000个数据区块。所述系统应用MD5散列算法来生成对应于I/O流中的42500000个区块的7800000个散列值区块。

[0259] 这会体现为仅仅使用存储起初的42500000个区块所需要的空间的18.5%。应用冲突模块，并且所述冲突模块会验证不存在任何冲突，也就是不会针对不同区块来生成散列值的复制值。

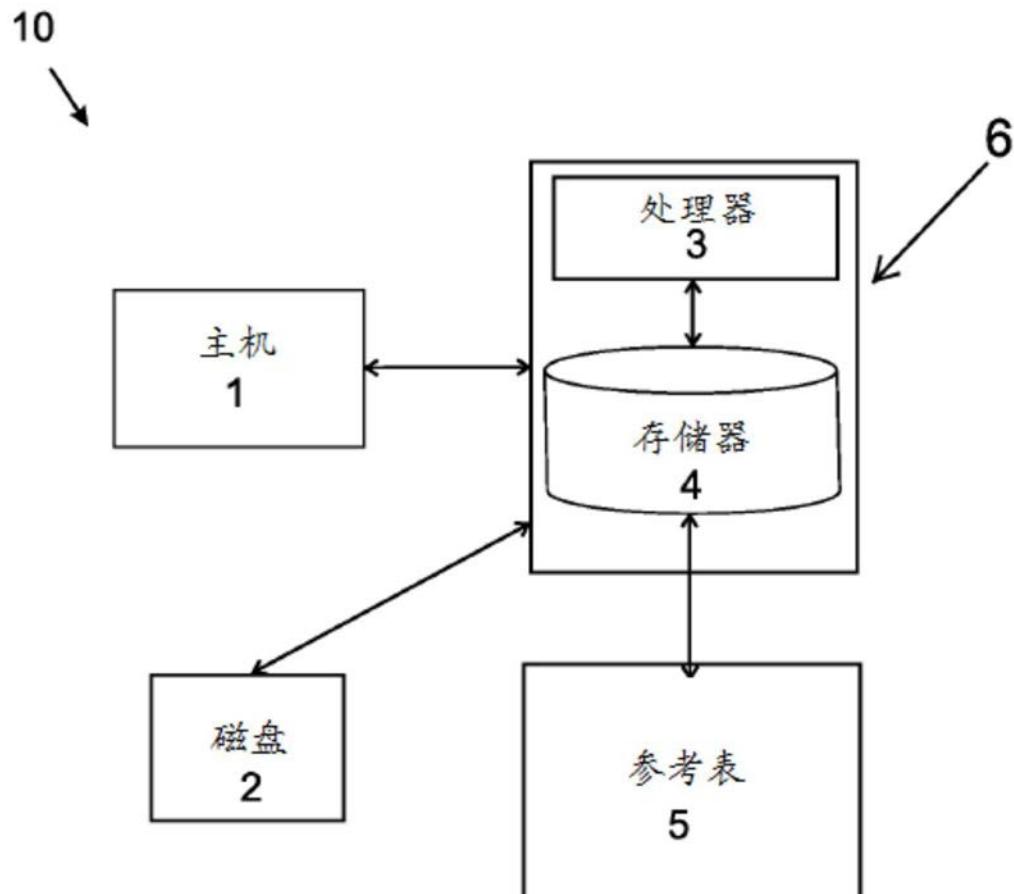


图1

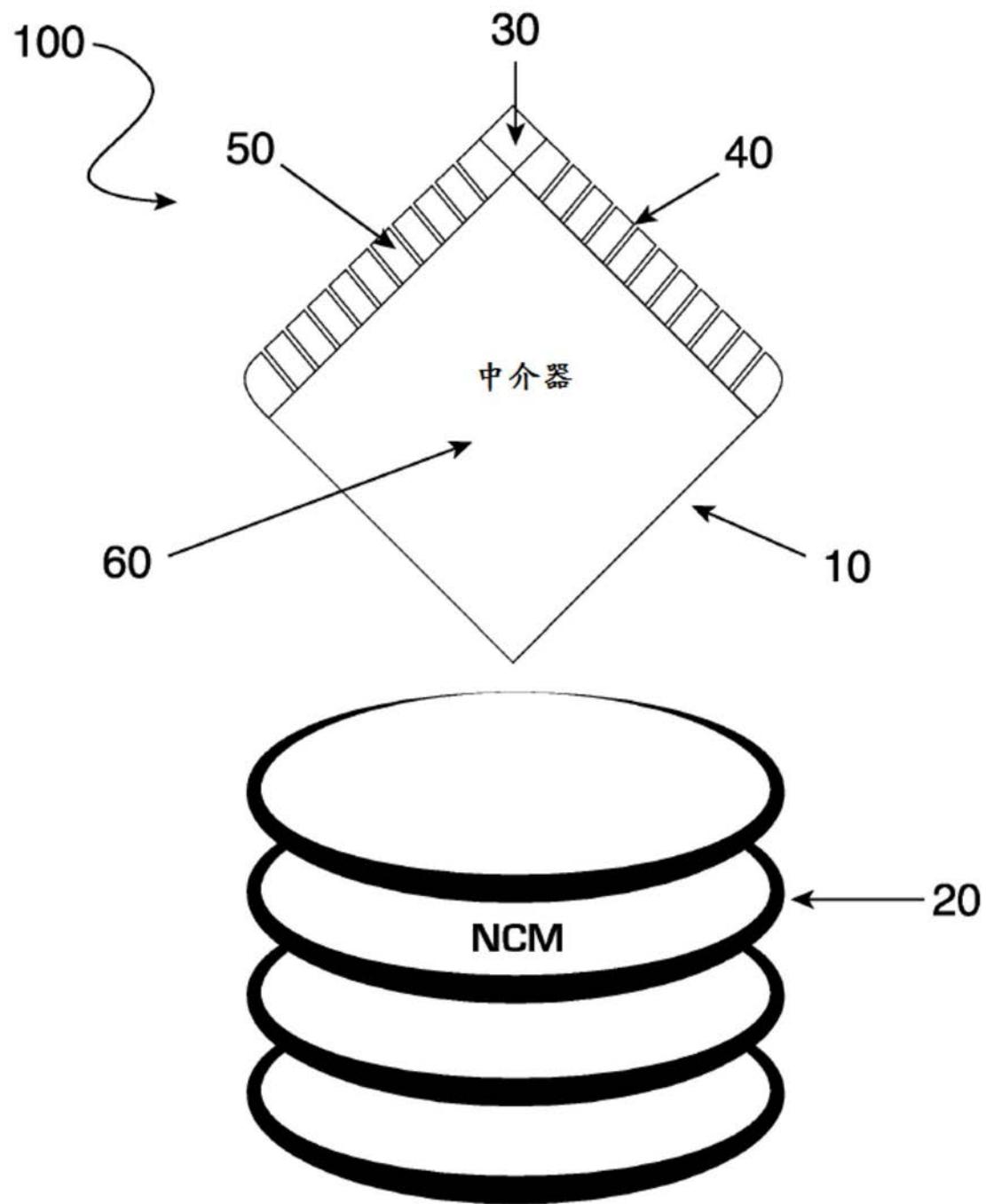


图2

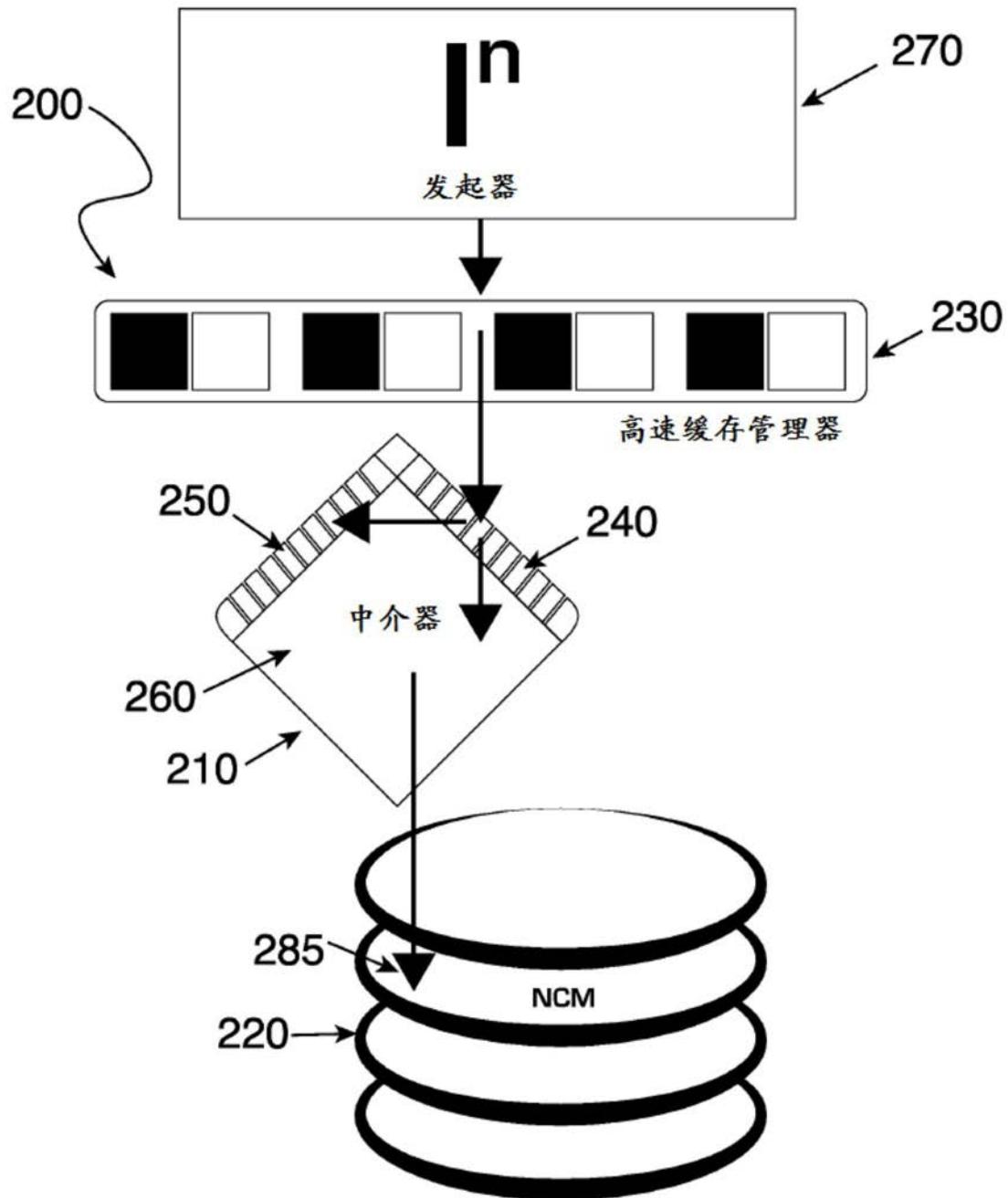


图3

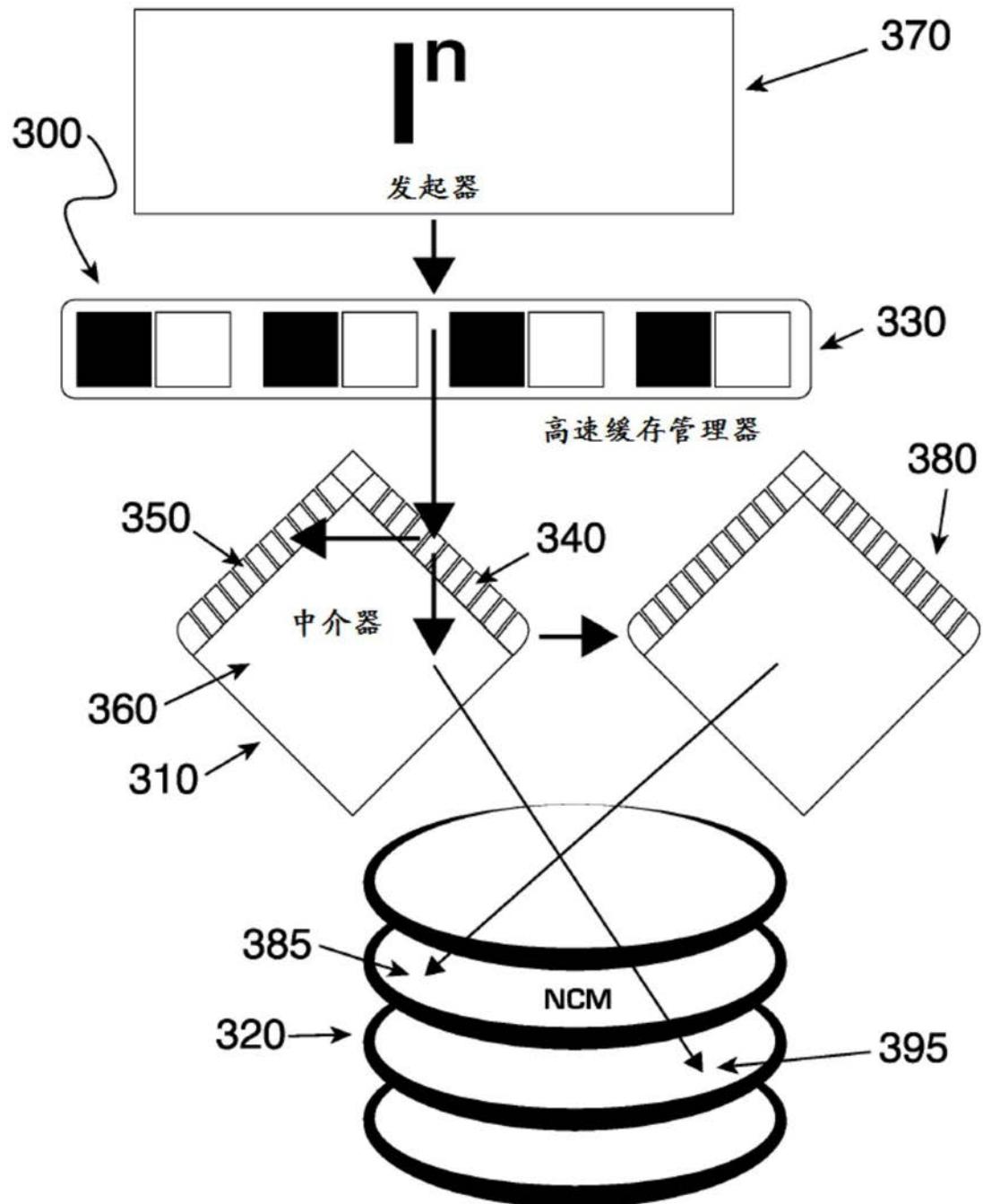


图4

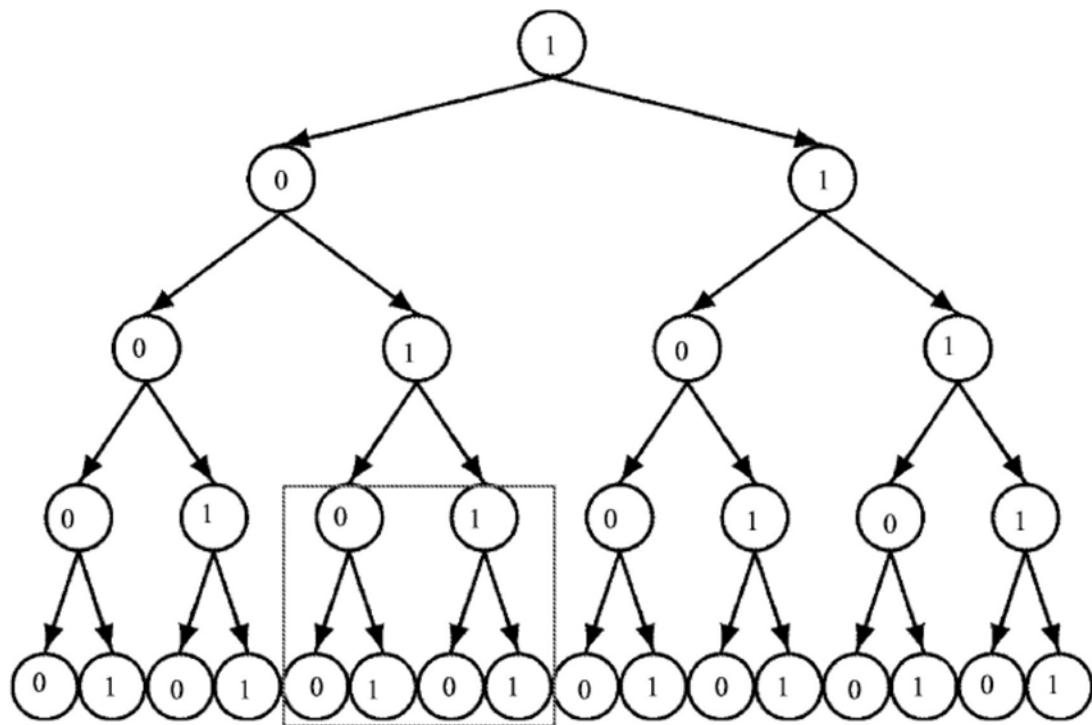


图5

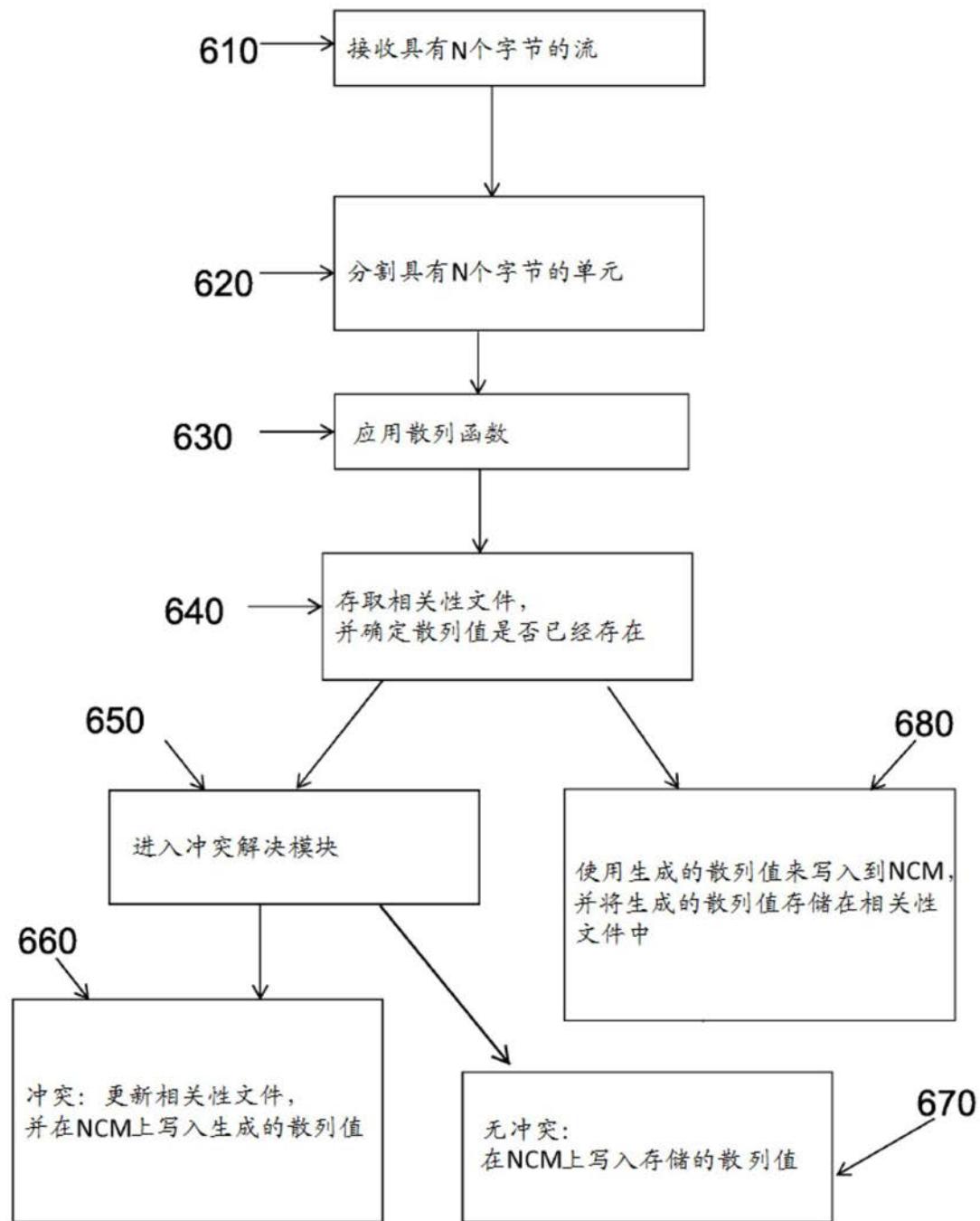


图6