(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101676855 B
(45)授权公告日 2014.04.02

(21)申请号 200910174385.2
(22)申请日 2009.09.11
(30)优先权数据
61/095,994 2008.09.11 US
12/511,126 2009.07.29 US
(73)专利权人 日本电气株式会社
地址 日本东京都
(72)发明人 C·杜布尼基 C·昂古里努
(74)专利代理机构 中国专利代理(香港)有限公司 72001
代理人 张晓冬 蒋骏
(51)Int.Cl.
G06F 3/06(2006.01)
G06F 17/30(2006.01)
H04L 29/08(2006.01)

(56)对比文件
US 20040215622 A1,2004.10.28, 全文.
US 2008005334 A1,2008.01.03, 全文.

审查员 郭从征

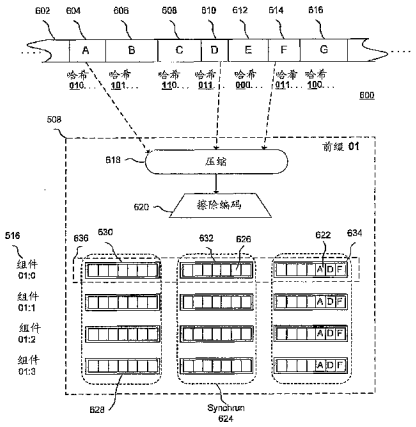
权利要求书2页 说明书19页 附图10页

(54)发明名称

可变动的辅助存储系统和方法

(57)摘要

依照本发明的实施例的示例性系统和方法可以通过采用可分裂的、可合并的和可传输的冗余数据容器链来提供多种数据服务。所述链和容器可以响应于存储节点网络配置的变化而自动地分裂和/或合并，且可以被存储在跨越不同存储节点而分布的经擦除编码的片段中。在利用冗余容器链的分布式辅助存储系统中所提供的数据服务可以包括全局重复删除、动态可变动性、对多种冗余等级的支持、数据定位、数据快速读写和由于节点或磁盘故障的原因而引起的数据重建。



CN 101676855 B

1. 一种用于管理辅助存储系统上的数据的方法，包括：

将数据块分布到位于节点网络中的多个不同物理存储节点中的不同数据容器以在所述节点中生成冗余数据容器链；

检测活动存储节点到所述网络的添加；

响应于检测到所述添加而自动地分裂至少一个容器链，其中所述自动地分裂包括将至少一个数据容器从所述至少一个容器链分离；

将从所述至少一个容器链分裂的数据的至少一部分从所述物理存储节点以及活动存储节点中的一个传输到所述物理存储节点以及活动存储节点中的另一个以提高针对节点故障的系统鲁棒性，其中数据的所述至少一部分在所述分裂之前被存储在所述至少一个数据容器中；和

将所述至少一个数据容器与另一数据容器合并。

2. 如权利要求 1 所述的方法，其中，所述数据容器链中的至少一个包括与其它数据容器链相同的元数据，其中，所述元数据描述数据块信息。

3. 如权利要求 2 所述的方法，还包括：

使用所述元数据响应于故障来重建数据。

4. 如权利要求 2 所述的方法，其中，所述元数据包括所述数据容器中的数据块之间的指针。

5. 如权利要求 4 所述的方法，还包括：

使用所述指针来删除数据。

6. 如权利要求 1 所述的方法，还包括：

对所述数据块进行擦除编码以生成经擦除编码的片段，其中，所述分布包括将经擦除编码的片段存储在所述不同数据容器中，使得源自所述数据块之一的片段被存储在不同存储节点上。

7. 如权利要求 6 所述的方法，还包括：

确定所述冗余数据容器链中的任何一个是否包括漏洞以确定所述数据块中的至少一个在所述辅助存储系统中是否可用。

8. 一种辅助存储系统，包括：

物理存储节点的网络，其中，每个存储节点包括

存储媒体，被配置为将数据块的片段存储在相对于其它存储节点中的数据容器链冗余的数据容器链中；以及

存储服务器，被配置为检测活动存储节点到所述网络的添加，响应于检测到所述添加而通过将所述至少一个数据容器从所述至少一个容器链分离来自动地分裂所述存储媒体上的所述至少一个容器链，将从所述至少一个容器链分裂的数据的至少一部分传输到不同的存储节点以提高针对节点故障的系统鲁棒性，其中数据的所述至少一部分在所述分裂之前被存储在所述至少一个数据容器中；和其中所述不同的存储节点被配置为将所述至少一个数据容器与另一数据容器合并。

9. 如权利要求 8 所述的系统，其中，所述存储服务器还被配置为执行以下各项中的至少一个：数据读取、数据写入、数据可用性确定、数据传输、分布式全局重复去除、数据重建以及数据删除。

10. 如权利要求 8 所述的系统，其中，所述数据容器链中的至少一个包括与其它数据容器链相同的元数据，其中，所述元数据描述数据块信息。

11. 如权利要求 10 所述的系统，其中，所述元数据包括所述数据容器中的数据块之间的指针，并且其中，所述存储服务器被配置为使用所述指针来执行数据删除。

12. 如权利要求 8 所述的系统，其中，依照哈希函数在所述存储媒体上对数据块的所述片段进行内容寻址。

13. 如权利要求 12 所述的系统，其中，内容地址的不同前缀与存储节点的不同子集相关联。

14. 如权利要求 13 所述的系统，其中，所述自动分裂包括扩展所述前缀中的至少一个以生成存储节点的至少一个附加子集。

15. 如权利要求 14 所述的系统，其中，所述传输包括将从所述至少一个容器链分裂的数据的所述至少一部分分布到附加子集。

可变动的辅助存储系统和方法

[0001] 相关申请信息

[0002] 本申请要求于2008年9月11日提交的临时申请序号61/095,994的优先权，在此引入以供参考。

技术领域

[0003] 本发明总体上涉及数据存储，更特别涉及辅助存储系统中的数据存储。

背景技术

[0004] 企业环境的一个重要方面，即辅助存储器技术的发展必须与企业不断增加的空前需求同步前进。例如，此类需求包括为具有不同重要水平的数据同时提供不同程度的可靠性、可用性和保留期限。此外，为了满足诸如萨班斯－奥克斯利(Sarbanes-Oxley，SOX)法案、健康保险携带与责任法案(Health Insurance Portability and Accountability Act，HIPPA)、爱国者法案、以及SEC规制17a-4(t)之类的规章要求，企业环境已需要改善辅助存储系统的安全性、可追溯性和数据审计。结果，理想的辅助存储体系结构严格地定义并制定精确的数据保留和删除程序。此外，它们应保留和恢复数据并根据需要呈现数据，因为不能这样做可能不仅导致经营效率的重大损失，而且可能导致罚款、甚至刑事诉讼。此外，由于商业企业时常采用相对有限的信息技术(IT)预算，所以效率在改善存储器利用率方面和降低安装数据管理成本方面也是最重要的。另外，随着产生的数据和与之相关联的固定备份窗口的量不断增加，明显需要适当地变动(scaling)性能和备份容量。

[0005] 如以磁盘为主体(disk-targeted)的重复删除(de-duplicating)虚拟磁带库(Virtual tape libraries，VTL)、基于磁盘的后端服务器和内容可寻址存档解决方案上的进步所证明的那样，为了解决这些企业需要，已取得了相当大的进展。然而，现有解决方案未充分解决与存储在辅助存储器中的数据量的指数式增加相关联的问题。

[0006] 例如，不像诸如存储区域网络(Storage Area Network，SAN)之类的通常被联网并接受公共管理的主存储器(primary storage)，辅助存储器包括许多高度专业化的专用组件(component)，其中的每一个都是使得需要使用用户化的、精细的、且常常是手动的监督和管理的存储岛(storage island)。因此，大部分的所有权总成本(total cost of ownership，TCO)可能归于较大程度的辅助存储器组件的管理。

[0007] 此外，现有系统对每个存储设备分配固定的容量并使重复去除(duplicate elimination)只限于一个设备，这导致容量利用率差并造成由存储在多个组件上的重复引起的空间浪费。例如，已知的系统包括大型经济磁盘冗余陈列(Redundant Array of Inexpensive Disk，RAID)系统，其提供包含可能是多个、但数目有限的控制器的单控制盒(control box)。这些系统的数据组织是基于固定尺寸的块接口。此外，该系统受到限制，这是因为其采用固定数据冗余模式，利用固定最大容量，并应用即使全部分区是空的也重建全部分区的重构模式。此外，它们未能包括用于提供重复去除的手段，这是因为用此类系统进行重复去除必须在较高层上予以实现。

[0008] 其它已知系统在诸如 DataDomain 之类的单盒或诸如 EMC Centera 之类的集群存储中递送高级存储。这些类型的系统中的缺点是它们提供容量和性能有限，采用与全局一个 (DataDomain) 相反的每盒重复去除或基于全部文件 (EMC Centera)。虽然这些系统给予 (deliver) 了诸如重复删除之类的一些高级服务，但其常常是中心化的，且这些系统所存储的元数据 / 数据不具有超过标准 RAID 模式的冗余。

[0009] 最后，由于这些已知辅助存储设备中的每一个提供固定且有限的性能、可靠性和可用性，所以很难满足对这些尺寸的企业辅助存储的高度总体需求。

发明内容

[0010] 依照本发明的各种示例性实施例的方法和系统通过提供一种便于于实现若干不同数据服务的数据组织模式来解决现有技术的不足。此外，示例性系统和方法提供了对现有技术的改善，这是因为示例性实施方式通过对改变网络配置自动作出反应并通过提供冗余来允许提供动态性。特别地，示例性实施方式可以响应于改变网络配置而分裂、合并和 / 或传输数据容器和 / 或数据容器链，这相对于已知过程是相当有益的。

[0011] 在本发明的一个示例性实施例中，一种用于管理辅助存储系统上的数据的方法包括将数据块分布到位于节点网络中的多个不同物理存储节点中的不同数据容器以在节点中生成冗余数据容器链；检测活动存储节点到网络的添加；响应于检测到所述添加而自动地分裂至少一个容器链；以及将从所述至少一个容器链分裂的数据的至少一部分从所述存储节点中的一个传输到所述存储节点中的另一个以提高针对节点故障的系统鲁棒性。

[0012] 在本发明的替代示例性实施例中，一种辅助存储系统包括物理存储节点的网络，其中，每个存储节点包括存储媒体，该存储媒体被配置为将数据块的片段存储在相对于其它存储节点中的数据容器链冗余的数据容器链中；以及存储服务器，被配置为检测活动存储节点到网络的添加，响应于检测到所述添加而自动地分裂所述存储媒体上的所述至少一个容器链，并将从所述至少一个容器链分裂的数据的至少一部分传输到不同的存储节点以提高针对节点故障的系统鲁棒性。

[0013] 在本发明的可替代示例性实施例中，一种用于管理辅助存储系统上的数据的方法包括将数据块分布到位于节点网络中的多个不同物理存储节点中的不同数据容器以生成节点中的冗余数据容器链；检测网络中的活动存储节点的数目的变化；以及响应于检测到所述变化而自动地将位于所述存储节点之一中的至少一个数据容器与位于不同存储节点中的另一数据容器合并以保证容器的可管理性。

[0014] 通过将结合附图来阅读的本发明的说明性实施例的以下详细说明，这些及其它特征和优点将变得显而易见。

附图说明

[0015] 本公开内容将参照附图提供优选实施例的以下说明的细节，在附图中：

[0016] 图 1 是依照本原理的一种示例性实施方式的辅助存储系统的后端部分中的数据块组织模式的方框 / 流程图。

[0017] 图 2 是依照本原理的一种示例性实施方式的辅助存储系统的方框 / 流程图。

[0018] 图 3 是依照本原理的一种示例性实施方式的物理存储节点的方框 / 流程图。

[0019] 图 4 是依照本原理的一种示例性实施方式的访问节点的方框 / 流程图。

[0020] 图 5 是依照本原理的示例性实施例的根据数据块哈希前缀指示存储节点分组的固定前缀网络的图示。

[0021] 图 6 是依照本原理的一种示例性实施方式的用于分布数据的系统的方框 / 流程图。

[0022] 图 7 是举例说明依照本原理的实施例的响应于存储节点到存储节点网络的添加或者对附加数据进行加载而分裂、连结 (concatenation)、和删除数据并回收存储空间的方框 / 流程图。

[0023] 图 8 是依照本原理的示例性实施方式的用于管理辅助存储系统中的数据的方法的方框 / 流程图。

[0024] 图 9 是举例说明依照本原理的一种示例性实施方式的可以由辅助存储系统来执行的多种数据服务的方框 / 流程图。

[0025] 图 10A ～ 10C 是举例说明依照本原理的示例性实施方式的在用于检测数据容器 (container) 链中的漏洞 (hole) 的扫描操作期间的不同时间帧 (time frame) 的方框 / 流程图。

[0026] 图 11 是依照本原理的可替换示例性实施方式的用于管理辅助存储系统中的数据的方法的方框 / 流程图。

具体实施方式

[0027] 如上所指出的，为了满足商业需求，分布式辅助存储系统应能够执行多种服务，包括：快速确定存储数据的可用性（即确定其是否可以被读取或其是否丢失）；支持多个数据冗余等级 (class)；快速确定任何给定时刻的数据冗余水平（即确定给定数据段可以经受多少次节点 / 磁盘故障而不使该数据丢失）；在冗余水平降低的情况下，快速将数据重建直至指定的冗余水平；以高水平性能来进行数据写入和读取；通过响应于网络配置的变化（例如添加了新节点和 / 或旧节点被移除或出现故障）而调整数据位置来提供动态可变动性；应要求删除数据；以及全局、高效的重复去除。

[0028] 虽然独自实现这些数据服务中的任何一个都是相对简单的，诸如高性能全局重复删除、分布式系统中的动态可变动性、删除服务以及故障排除，但是一起提供它们中的每一个却相当困难。例如，在提供重复删除和动态可变动性之间存在紧张状况，这是因为当存储系统扩大或存储节点的配置改变时难以确定重复数据的位置。另外，在提供重复删除与应要求删除之间存在冲突。例如，为了防止数据丢失，应避免对已被计划删除的数据进行重复删除。在提供容错与删除之间也存在紧张状况，这是因为在发生故障的情况下，删除决定应该是一致的。

[0029] 如上所述，例如，诸如 RAID 之类的当前辅助存储系统未能充分地提供这样的数据服务的组合。本发明的示例性实施方式通过提供用于在保持效率和性能的同时平衡由不同类型的数据服务引起的需求的新颖手段来解决现有技术的不足。例如，下述示例性数据组织模式允许解决这些服务之间紧张和冲突并便于在辅助存储系统中实现这些服务中的每一种。

[0030] 如下文所讨论的，本发明的示例性实施例包括商业存储系统，该商业存储系统包

括被架构为存储节点的网格 (grid) 的后端。前端可以包括针对性能而变动的访问节点层，如本领域的普通技术人员所理解的那样，该访问节点层可以使用诸如像网络文件系统 (network file system，NFS) 或通用因特网文件系统 (common internet file system，CIFS) 协议之类的标准文件系统接口来予以实现。本文所公开的本原理主要针对辅助存储系统的后端部分，其可以是基于内容可寻址的存储。

[0031] 依照本原理的示例性实施方式，辅助存储容量可以在所有客户端和诸如备份数据和档案数据之类的所有类型的辅助存储数据之间动态共享。除容量共享之外，可以应用如下文所述的全系统范围内的重复去除以改善存储容量效率。示例性系统实施方式具有高度可用性，这是因为其可以支持在线扩展和升级，容许多次节点和网络故障，在故障之后自动重建数据并可以将关于所存放数据的可恢复性的信息告知用户。可以由客户端随着每次写入而另外自动地调整存储数据的可靠性和可用性，这是因为如下文更全面地描述的所述后端可以支持多个数据冗余等级。

[0032] 本原理的示例性实施例可以采用如下所述经修改的各种模式和特征来实现诸如数据重建、分布式的和应要求的数据删除、全局重复去除和数据完整性管理之类的高效的数据服务。此类特征可以包括利用已修改内容可寻址存储范例，其使得能够廉价且安全地实现重复去除。其它特征可以包括使用已修改分布式哈希表，其允许建立可变动且耐故障的系统并将重复去除扩展至全局水平。此外，可以采用擦除代码 (erasurecode) 来以期望冗余水平和所得存储开销之间的细粒度控制来向存储数据添加冗余。硬件实施方式可以利用给予大量原始但廉价的存储容量的大型可靠 SATA 磁盘。还可以采用多核 CPU，这是因为它们提供廉价、却强大的计算资源。

[0033] 本原理的示例性系统实施方式还可以是可变动到至少数千个专用存储节点或设备，结果得到几百拍它字节 (petabyte) 量级的原始存储容量，具有潜在较大的配置。虽然系统实施方式可以包括潜在的大量存储节点，但系统可以在外部起到一个大型系统的作用。此外，还应注意的是，下文所讨论的系统实施方式不需要限定一个固定访问协议，而是可以具有灵活性以允许支持使用诸如文件系统接口之类的标准的继承 (legacy) 应用和使用高度专业化的访问方法的新应用这两者。可以用新协议驱动器 (driver) 来在线添加新协议，而不中断使用现有协议的客户。因此，系统实施方式可以支持定制的新应用和商业继承应用这两者，如果它们使用流数据访问的话。

[0034] 辅助存储器实施方式的其它示例性特征可以允许系统在各种情况期间连续运行，这是因为其可以限制或消除故障、升级和扩展对数据和系统可访问性的影响。由于分布式体系结构，甚至常常可以在例如滚动升级之类的硬件或软件升级期间维持连续不停的系统可用性，从而消除对任何昂贵的停机时间的需要。此外，示例性系统能够在在由于磁盘故障、网络故障、功率损耗引起的硬件故障的情况下，甚至从某些软件故障中进行自动恢复。另外，示例性实施例可以经受特定的、可配置次数的故障停止 (fail-stop) 和间歇的硬件故障。此外，可以采用若干层的数据完整性校验来检测随机数据损坏。

[0035] 示例性系统的又一重要功能是确保高数据可靠性、可用性和完整性。例如，可以以用户选择的冗余水平写入每个数据块，从而允许该块存在直到请求次数的磁盘和节点故障。可以通过将每个块擦除编码成片段 (fragment) 来达到用户可选冗余水平。擦除代码通过相同空间开销量内的简单复制而将平均无故障时间增加许多个数量级。在故障之后，

如果块仍然可读，则系统实施方式可以自动地调度数据重建以使冗余恢复回到用户所请求的水平。此外，辅助存储系统实施方式可以确保不会有永久性数据丢失被长时间隐藏。系统的全局状态可以指示所有存储块是否是可读的，且如果是可读的，则其指示在发生数据丢失之前可以经受多少次磁盘和节点故障。

[0036] 现在详细地参照附图，在附图中，相同的标号表示相同或类似元素，首先参照图 1，图示了依照本原理的一种示例性实施方式的辅助存储系统的后端部分中的数据块组织结构的表示 100。用于数据块组织的编程模型是基于海量尺寸可变的、内容寻址的、具有高恢复性的块的抽象 (abstraction)。块地址可以由其内容的哈希导出，所述哈希例如 SHA-1 哈希。块可以包括数据，并且任选地包括指向已写入的块的指针阵列。块可以是尺寸可变的，以容许有较佳重复去除率。另外，可以使指针暴露以便于被实现为“无用单元收集 (garbage collection)”的数据删除，所述“无用单元收集”是回收对象不再使用的存储器的一种类型的存储器管理进程 (process)。此外，辅助存储系统的后端部分可以导出 (export) 协议驱动器所使用的低级块接口以实现新协议和继承协议。提供此类块接口来代替诸如文件系统之类的高级块接口的措施可以简化实施方式并允许后端与前端清晰分离。此外，此类接口还允许高效地实现大范围的许多高级协议。

[0037] 如图 1 所示，辅助存储系统的后端部分中的块可以形成有向无环图 (Directed Acyclic Graph，DAG)。每个块的数据部分被遮蔽，而指针部分未被遮蔽。驱动器可被配置为写入数据块树。然而，由于示例性辅助存储系统的重复删除特征，因此这些树在经重复删除的块处重叠并形成有向图。另外，只要用于块地址的哈希是安全的，在这些结构中就不可能存在环。

[0038] DAG 中的源顶点通常是称为“可搜索保留根”的特殊块类型的块。除规则数据和地址阵列之外，保留根可被配置为包括对块进行定位的用户定义的搜索关键字。此类关键字可以是任意数据。用户可以通过提供可搜索块的搜索关键字而不是其加密块内容地址来检索可搜索块。结果，用户不需要记住内容地址以访问存储数据。例如，相同文件系统的多个快照可以具有被组织为可搜索保留根的每个根，所述可搜索保留根具有包括文件系统名的搜索关键字和随着每个快照而增值的计数器。可搜索块不具有用户可见的地址且不能被指向。同样地，可搜索块不能用来在块结构中产生环。

[0039] 再次参照图 1，该组块 100 包括三个源顶点 102、104 和 106，其中的两个 102、104 是保留根。另一源顶点 106 是规则块 A，其指示此部分 DAG 仍在构造中。

[0040] 如下文所讨论的，应用编程接口 (Application Programming Interface，API) 操作可以包括写入和读取规则块、写入可搜索保留根、基于保留根的搜索关键字来搜索保留根、以及用指定的关键字将保留根标记为将通过写入相关联删除根而删除。应注意的是，可以由驱动器来执行将数据流切割成块。

[0041] 依照本原理的一个示例性方面，在写入块时，用户可以将该块分配以多个可用冗余等级之一。每个等级可以表示数据冗余与存储开销之间的不同权衡。例如，低冗余数据等级中的块只能经受一次磁盘故障，而用于其块尺寸的存储开销是最小的。随后，关键数据等级中的块可以在不同的磁盘和物理节点上被多次复制。本原理的辅助存储系统可以支持这两种极端之间的许多不同冗余等级。

[0042] 还应注意的是示例性辅助存储系统不应提供直接删除单个块的方式，这是因为

此类块可能被其它块引用。相反，API 可以允许用户指示应通过标记保留根来删除哪部分 DAG。为了标记不活动的保留根，用户可以通过对被称为“可搜索删除根”的特殊块分配搜索关键字来将其写入，所述搜索关键字与要删除的保留根的搜索关键字是同样的。

[0043] 例如，再次参照图 1，可以使删除根 108 与保留根 SP1 102 相关联。辅助存储系统所采用的删除算法可被配置为标记删除从活动保留根不可到达的所有块。例如，在图 1 中，如果用户写入删除根 108，则具有虚线的所有块，块 A106、D110、B112、以及 E114 都被标记为删除。请注意，块 A106 也被标记为删除，这是因为不存在指向它的保留根，而块 F116 被保留，这是因为其可从保留根 SP 2 104 到达。这里，块 104 是活动的，这是因为其不具有匹配的删除根。在数据删除期间，存在短暂的只读期，在此期间系统标识要删除的块。在正常读写操作期间，后台可以发生实际空间回收。此外，在进入只读阶段之前，应由活动保留根指向要保留的所有块。

[0044] 现在参照图 2，图示了依照本原理的一个示例性实施例的辅助存储系统 200 的高级方框 / 流程图，所述辅助存储系统 200 可以实现上文所讨论的数据组织模型和 API 操作。应当理解，本文所述的实施例可以完全是硬件的或者可以包括硬件和软件元素这两者。在优选实施例中，本发明是以硬件和软件予以实现的，所述硬件和软件包括但不限于固件、常驻软件、微码等。

[0045] 实施例可以包括可从提供程序代码的计算机可用或计算机可读的媒体访问以供计算机或任何指令执行系统使用或与之连同使用的计算机程序产品。计算机可用或计算机可读的媒体可以包括存储程序以供指令执行系统、装置或设备使用或与之连同使用的任何设备。所述媒体可以是磁性的、光学的、电子的、电磁的、红外的、或半导体系统（或装置或设备）。该媒体可以包括计算机可读媒体，诸如半导体或固态存储器、磁带、可移动计算机磁盘、随机存取存储器 (random access memory，RAM)、只读存储器 (read-only memory，ROM)、硬 (rigid) 磁盘和光盘等。

[0046] 此外，计算机可读媒体可以包括计算机可读程序，其中，当在计算机上执行所述计算机可读程序时，其促使计算机执行本文所公开的方法步骤和 / 或体现存储节点上的一个或多个存储服务器。类似地，有形地体现机器可执行的指令程序的机器可读程序存储设备可被配置为执行下面将更全面地讨论的用于管理辅助存储系统上的数据的方法步骤。

[0047] 系统 200 可以包括被配置为与用户输入交互并实现存储、检索、删除和在其他情况下管理数据的用户命令的应用层 202。应用层 202 下面的 API 204 可被配置为与应用层 202 通信并与前端系统 206 交互以开始数据管理。前端系统 206 可以包括可以与内部网络 210 通信的多个访问节点 208a ～ f。另外，前端系统 206 可以经由应用编程接口 214 与后端系统 212 交互，应用编程接口 214 又可以与协议驱动器 216 交互。后端系统 212 可以包括存储节点 218a ～ f 的网格，该网格可被应用层视为大存储单元中的文件系统集合。此外，虽然这里为了简洁起见示出六个存储节点，但可以依照设计选择，提供任何数目的存储节点。在示例性实施方式中，访问节点的数目可以在一个存储节点与存储节点的一半之间的范围内，但该范围可以根据用来实现存储和访问节点的硬件而改变。后端存储节点还可以经由内部网络 210 来通信。

[0048] 现在，在继续参照图 2 的情况下参照图 3 和 4，图示了依照本原理的实施例的示例性存储节点 218 和示例性访问节点 208。存储节点 218 可以包括一个或多个存储服务器

302、处理单元 304 和存储媒体 306。所述存储服务器可以在被配置为在处理单元 304 和存储媒体 306 上运行的软件中予以实现。类似地，访问节点 208 可以包括一个或多个代理服务器 402、处理单元 404 和存储媒体 406。应当理解，虽然这里将存储节点和访问节点描述为在分开的机器上实现，但可以想到，可以在相同机器上实现访问节点和存储节点。因此，如本领域的技术人员理解的那样，一台机器可以同时实现存储节点和访问节点这两者。

[0049] 应当理解，根据本文所描述的讲授内容，如本领域的技术人员所理解的那样，可以以各种形式的硬件和软件来实现系统 200。例如，合适的系统可以包括被配置为运行一个后端存储服务器并具有六个 500 GBSATA 盘、6GB RAM、两个双核 3GHz CPU 和两个 GigE 卡的存储节点。可替换地，每个存储节点可被配置为运行两个后端存储服务器并具有十二个 1TB SATA 盘、20GB 的 RAM、两个四路 3GHz CPU 和四个 GigE 卡。此外，例如，访问节点 208 可以包括 6GB RAM、两个双核 3GHz CPU、两个 GigE 卡和仅一个小型本地存储。此外，存储和访问节点还可被配置为运行 Linux，Red Hat EL 5.1 版。然而，应当将对硬件元素的详细说明理解为仅仅是示例性的，这是因为根据本文所公开的讲授内容，本领域的技术人员还可以实现其它配置和硬件元素。

[0050] 再次参照图 2 ～ 4，如上文所讨论的，辅助存储系统 200 的组件可以包括存储服务器 302、代理服务器 402 和协议驱动器 216。此外，每个存储节点 218 可被配置为主控 (host) 一个或多个存储服务器 302 进程。在存储节点 218 上运行的存储服务器 302 的数目取决于其可用资源。节点 208 越大，应运行的服务器 302 越多。每个服务器 302 可被配置为排他地负责其存储节点的特定数目的盘。例如，通过使用多核 CPU，每个存储服务器 302 的并行性可以随着核的数目的每次增加而保持恒定，且可以在一个存储节点上放置多个存储服务器。

[0051] 如上文所讨论的，代理服务器 402 可以在访问节点上运行并导出与存储服务器相同的块 API。代理服务器 402 可被配置为提供诸如定位后端节点和执行优化消息路由及缓存之类的服务。

[0052] 协议驱动器 216 可被配置为使用由后端系统 212 导出的 API 214 来实现访问协议。该驱动器可以在运行时间内被加载在存储服务器 302 和代理服务器 402 这两者上。哪个节点将在给定驱动器上加载的确定可以取决于可用资源和驱动器资源需求。诸如文件系统驱动器之类的需要资源的驱动器可以被加载在代理服务器上。

[0053] 存储服务器 302 可被设计为在分布式环境中供多核 CPU 使用。另外，存储服务器 302 的特征可以为多组程序员的并行开发提供支持。此外，存储服务器 302 特征还可以提供所得系统的高可维护性、可测试性和可靠性。

[0054] 为了实现存储服务器 302 特征，可以采用异步管道 (pipelined) 消息传递框架，其包括称为“管道单元”的站。管道中的每个单元可以是单线程的，且不需要与其它单元对任何数据结构进行写共享。此外，管道单元还可以具有某些内部工作者线程。在一个示例性实施例中，管道单元仅仅通过消息传递进行通信。同样地，该管道单元可以在相同物理节点上共同定位 (co-located) 并分布于多个节点。当通信管道单元在相同节点上共同定位时，可以使用只读共享作为优化。同步和并发问题 (issues) 可以仅限于一个管道单元。另外，可以通过提供其它管道单元的接头 (stub) 来单独地测试每个管道单元。

[0055] 为了允许有容易的可变动性，可以采用分布式哈希表 (distributedhash table,

DHT) 来组织数据的存储位置，由于分布式存储系统应包括高效存储利用率和足够的数据冗余，所以应使用 DHT 的附加特征。例如，该附加特征应提供关于存储利用率和所选覆盖网络与诸如擦除编码之类的数据冗余模式的容易集成的保证。由于现有 DHT 并未充分地提供此类特征，所以可以使用固定前缀网络 (Fixed Prefix Network，FPN) 分布式哈希表的修改版。

[0056] 现在参照图 5，图示了依照本原理的示例性实施例的固定前缀网络的表示 500。在 FPN 中，准确地为每个覆盖节点 502、504 分配一个哈希关键字 (hashkey) 前缀，该哈希关键字前缀还是覆盖节点的标识符。所有前缀一起覆盖整个哈希关键字空间，且覆盖网络力图使其保持分离。FPN 节点，例如节点 506 ～ 512 中的任何一个，负责具有等于 FPN 节点标识符的前缀的哈希关键字。图 5 的上部图示了具有以四个 FPN 节点 506 ～ 512 作为叶子的前缀树，所述四个 FPN 节点 506 ～ 512 将前缀空间划分为四个分离的子空间。

[0057] 对于依照本原理的各方面的 DHT，如图 5 所示，可以用“超级节点”来修改 FPN。超级节点可以表示一个 FPN 节点（且同样地，用哈希关键字前缀来对其进行识别）并跨越若干物理节点以提高对节点故障的恢复性。例如，如图 5 所示，如果后端网络包括六个存储节点 514a ～ 514f，则每个超级节点 508 ～ 512 可以包括称为“超级节点基数 (cardinality)”的固定数目的超级节点组件 516，其可以设置在分开的物理存储节点 514a ～ 514f 上。相同超级节点的组件可以称为“同伴 (peer)”。因此，可以将每个超级节点视为存储节点 514a ～ 514f 的子集。例如，超级节点 506 可以由存储节点的子集 514a ～ 514d 组成。此外，每个存储节点可以被包括在少于超级节点总数的数个超级节点中。例如，节点 514a 可以被包括在四个超级节点中的三个中，即超级节点 506、510 和 512 中。

[0058] 依照本原理的示例性实施方式，可以使用固定前缀网络来将数据块的存储分配到存储节点。例如，在对数据进行哈希之后，可以使用哈希结果中的前几个位、在本示例中为前两位来将数据块分布到每个超级节点。例如，可以将具有从“00”开始的哈希值的数据块分配给超级节点 506，可以将具有从“01”开始的哈希值的数据块分配给超级节点 508，可以将具有从“10”开始的哈希值的数据块分配给超级节点 510，且可以将具有从“11”开始的哈希值的数据块分配给超级节点 512。其后，如下文相对于图 6 更全面地讨论的，可以将数据块的各部分分布在超级节点的各组件 516 之间，该数据块被分配给所述超级节点。这里，将超级节点 506 的组件标志为 00:0、00:1、00:2、00:3。其它超级节点的组件类似地予以表示。

[0059] 应当注意，例如，超级节点基数可以在 4 ～ 32 范围内。然而，可以采用其它范围。在优选实施例中，将超级节点基数设置为 12。在示例性实施方式中，对于所有超级节点，超级节点基数可以是相同的，且在整个系统生命期内可以是恒定的。

[0060] 还应当理解，超级节点同伴们可以采用分布式一致性算法来确定应对该超级节点施加的任何改变。例如，在节点故障之后，超级节点同伴们可以确定应在哪些物理节点上重新创建丢失组件的新实体 (incarnation)。另外，超级节点同伴们可以确定哪些替代节点可以取代出现故障的节点。例如，返回参照图 5，如果节点 1 514a 出现故障，则可以用来自超级节点 506 的其它组件的数据，使用擦除编码来在节点 5 514e 处重构组件 00:1。类似地，如果节点 1 514a 出现故障，则可以用来自超级节点 510 的其它组件的数据在节点 3 514c 处重建组件 10:0。此外，可以使用来自超级节点 512 的其它组件的数据在节点 4 514d 处重

建组件 11:2。

[0061] 关于由本原理的辅助存储实施例提供的读和写处理，在写时，可以将数据块路由到超级节点的同伴之一，所述超级节点负责此块的哈希所属的哈希关键字空间。接下来，如下文更全面地讨论的，该写处理同伴可以检查是否已经存储了合适的重复。如果发现重复，则返回其地址；否则，如果用户请求，则对该新块进行压缩，进行分片段并将片段分布到相应超级节点下的其余同伴。依照可替代实施方式，可以通过对访问节点上的块进行哈希并只向存储节点发送哈希值而不发送该数据来执行重复删除。这里，存储节点可以通过将从访问节点接收到的哈希值与存储块的哈希值进行比较来确定该块是否是重复。

[0062] 读请求也被路由到负责数据块哈希关键字的超级节点的同伴之一。该同伴可以首先定位可以在本地找到的块元数据，并可以向其它同伴发送片段读请求以便读取足以依照擦除编码模式来重构数据块的最少数目的片段。如果任何请求超时，则可以读取所有其余片段。在已找到足够数目的片段之后，可以对该块进行重构、解压缩（如果其被压缩）、校验，并在校验成功的情况下返回给用户。

[0063] 总的来说，对于流访问，读取非常高效，因为可以将所有片段顺序地从磁盘预先取到本地高速缓冲存储器。然而，由同伴来确定片段位置可能是相当精细的过程。常常通过对本地节点索引和本地高速缓存加标注来进行片段位置的确定，但在某些情况下，例如，在组件传输期间或间歇的故障之后，所请求的片段可能只出现在其先前位置之一上。在这种情况下，该同伴可以例如通过按颠倒次序搜索先前组件位置的踪迹来指挥对遗失数据的分布式搜索。

[0064] 可以包括在本原理的辅助存储系统实施例中的另一示例性特征是“负载平衡”，其确保数据块的组件充分地遍布于系统的不同物理存储节点。组件在物理存储节点之间的分布改善了系统可生存性、数据恢复性和可用性、存储利用率以及系统性能。例如，如果相应的存储节点丢失，则在一个机器上放置太多同伴组件可能具有灾难性后果。结果，受到影响的超级节点可能不会从故障中恢复，这是因为太多组件不能检索。即使该存储节点可恢复，也可能由于丢失太多片段而不可读取由相关联的超级节点处理的某些或者甚至全部数据。而且，当与可用节点资源成比例地向物理存储节点分配组件时，系统的性能被最大化，这是因为每个存储节点上的负载与被分配给存储节点的组件所覆盖的哈希关键字前缀空间成比例。

[0065] 示例性系统实施方式可被配置为连续地尝试平衡遍及所有物理机器或存储节点的组件分布以达到故障弹性、性能和存储利用率被最大化的状态。可以通过将这些目标区分优先次序的多维函数来测量给定分布的质量，所述多维函数称为系统熵。此类平衡可以由每个机器 / 存储节点来执行，其可被配置为周期性地考虑本地主控的组件到相邻存储节点的所有可能的传输集。如果存储节点发现了会改善分布的传输，则执行此组件传输。另外，可以向系统添加防止同时发生多个冲突传输的安全装置。在组件到达新位置之后，其数据也从旧位置移动到新位置。该数据传输可以在后台执行，这是因为在某些情况下，执行该数据传输可能耗费很长时间。

[0066] 还可以使用负载均衡来管理存储节点机向 / 从辅助存储系统的添加和移除。可以应用上述相同熵函数来测量机器添加 / 移除之后所得的组件分布的质量。

[0067] 示例性辅助存储系统的另一重要特征是超级节点基数的选择，这是因为超级节点

基数的选择可能对辅助存储系统的属性有深远影响。首先，超级节点基数可以确定容许节点故障的最大次数。例如，只要每个超级节点保持活动，后端存储网络就可经受住存储节点故障。如果至少每个超级节点的一半同伴加一个应保持活动以达成一致，则该超级节点保持活动。结果，辅助存储系统至多经受住主控每个超级节点的同伴的物理存储节点之间的超级节点基数的一半减 1 次永久节点故障。

[0068] 超级节点基数还影响可变动性。对于给定基数，每个超级节点幸存的概率是固定的。此外，幸存概率直接取决于超级节点基数。

[0069] 最后，超级节点基数可以影响可用数据冗余等级的数目。例如，用在块仍保持为可重构的同时可能丢失的最大数目的片段对擦除编码进行参数化。如果采用擦除编码且其产生超级节点基数片段，则丢失片段的容许数目可能在一至超级节点基数减一的范围内变化（在后者情况下，可以保持此类块的超级节点基数拷贝）。丢失片段的容许数目的每种此类选择能够定义不同的数据冗余等级。如上文所讨论的，每个等级可以表示例如由于擦除编码的原因而引起的存储开销与故障恢复性之间的不同权衡。可以用丢失片段的容许数目与超级节点基数和丢失片段的容许数目之间的差的比来表征此开销。例如，如果超级节点基数是 12，且块可能丢失不超过 3 个片段，那么此等级的存储开销由 3 与 (12-3) 的比，即 33％，给出。

[0070] 现在参照图 6 和 7，图示了依照本原理的实施方式的使用数据容器链的辅助存储系统的示例性数据组织结构 600、700。存储数据的表示 600、700 允许很大程度上的可靠性、可用性和性能。使用此类数据组织结构的辅助存储系统的实施方式使得快速标识存储数据可用性以及响应于故障而将数据重建至指定冗余水平能够实现。将数据重建至指定冗余水平提供了相对于诸如 RAID 之类的系统的显著优势，所述 RAID 即使在盘不包含有效用户数据的情况下也重建整个盘。如下文所讨论的，由于数据块组件在节点之间移动，后面是数据传输，所以系统可以定位并检索来自旧组件位置的数据，这比重建数据有效得多。应将被写入一个流中的数据块放置得相互接近以使写和读性能最大化。此外，采用下述数据结构实施方式的系统还可以支持应要求分布式数据删除，在应要求分布式数据删除中，其删除从任何活动保留根不可达的数据块，并回收该不可达数据块所占用的空间。

[0071] 图 6 图示了依照本原理的一种示例性实施方式的用于分布数据的系统 600 的方框 / 流程图。如图 6 所示，包括数据块 A604、B606、C608、D610、E612、F614 和 G616 的数据流 602 可以经受诸如 SHA-1 之类的哈希函数或任何其它合适的内容可寻址存储模式。如本领域的技术人员所理解的，内容可寻址存储模式可以对数据块的内容应用哈希函数以获得该数据块的唯一地址。因此，数据块地址可以基于数据块的内容。

[0072] 在继续参照图 5 的情况下回到图 6，在所提供的示例中，数据块 A604、D610 和 F616 的哈希结果具有前缀“01”。因此，可以将块 A604、D610、和 F616 分配给超级节点 508。在根据数据流计算数据块的哈希之后，可以将各个数据块压缩 618 并进行擦除编码 620。如上文所讨论的，可以采用擦除编码来实现数据冗余。可以将一个数据块的不同的所得擦除编码片段 622 分布到超级节点的同伴组件 516，该数据块被分配给所述超级节点。例如，图 6 图示了超级节点 508 的同伴组件 01:0、01:1、01:2 和 01:3，在其上面，存储了来自具有前缀“01”的数据块，即数据块 A604、D610 和 F616，的经擦除编码的不同片段 622。

[0073] 在本文中将示例性辅助存储系统实施例所采用的数据管理的基本逻辑单元定义

为“synchrun”，其是由写处理同伴组件写入并属于给定超级节点的许多连续数据块。例如，synchrun 624包括来自其相应超级节点的每个组件516的许多连续数据块片段626。这里，可以按照块在数据流602中出现的顺序存储每个片段。例如，在存储块D610的片段之前存储块F614的片段，又在块A604的片段之前存储块D610。保留时间顺序(temporal order)使数据的可管理性变得容易并允许推理存储系统的状态。例如，时间顺序的保留允许系统确定在出现故障的情况下某一日期之前的数据是可重构的。

[0074] 这里，写入块基本上是写入其片段622的超级节点基数。因此，可以用synchrun组件的超级节点基数来表示每个synchrun，每个同伴一个。synchrun组件对应于例如片段626之类的经synchrun组件所属的超级节点前缀过滤的片段的时间顺序。容器可以存储一个或多个synchrun组件。对于超级节点的第i个同伴，相应的synchrun组件包括synchrun块的所有第i个片段。synchrun仅仅是逻辑结构，但synchrun组件基本上存在于相应的同伴上。

[0075] 对于给定的写处理同伴，辅助存储系统可被配置为使得在任何给定时间只有一个synchrun是开放的。结果，所有这样的synchrun按照逻辑顺序排列成链，该顺序由写处理同伴来确定。Synchrun组件可以放置在在本文中被称为synchrun组件容器(synchrun component container，SCC)628的数据结构中。每个SCC可以包括一个或多个链相邻的synchrun组件。因此，SCC也形成类似于synchrun组件链的链。此外，在一个同伴中包括多个SCC。例如，同伴01:0可以包括SCC 630、SCC 632和SCC634。因此，按顺序排列在单个同伴上的多个SCC被称为“同伴SCC链”636。另外，可以由同伴SCC链的超级节点基数来表示synchrun链。例如，在图7中的行724～732中图示了同伴链，下文对其进行更全面的讨论。

[0076] 总的来说，相对于synchrun组件/片段622元数据和它们每一个中的片段数目，同伴SCC链可以是同样的，但偶尔可能存在例如由导致链漏洞的节点故障引起的差异。此链式组织允许相对简单且高效地实现辅助存储系统的数据服务，诸如数据检索和删除、全局重复去除、以及数据重建。例如，链漏洞可以用来确定数据是否可用(即所有相关联的块是可重构的)。因此，如果足够数目的同伴链不具有任何漏洞，所述足够数目等于用来完全重构每个块的片段的数目，则可以将该数据视为可用。如果使用冗余等级，则可以对每个冗余等级类似地进行数据可用性确定。

[0077] 此外，应当理解，该系统可以存储不同类型的元数据。例如，数据块的元数据可以包括指向其它数据块的暴露指针，该暴露指针可以用具有指针的数据块的每个片段来复制。其它元数据可以包括片段元数据，该片段元数据例如包括块哈希信息和块恢复信息。片段元数据可以与该数据分开存储，并可以用每个片段来复制。另外，数据容器可以包括与其存储的链的一部分有关的元数据，诸如该容器存储的synchrun组件的范围。由容器保持的此元数据允许快速推理系统中的数据的状态和诸如数据建立和传输之类的数据服务的性能。因此，每个容器包括数据和元数据这两者。如上所述，元数据可以复制，而可以用擦除代码的参数化来维持用户所请求的数据的冗余水平。因此，可以将诸如存储在超级节点508中的组件/存储节点中的每一个上的数据链之类的容器链视为冗余，这是因为可能存在元数据相同但数据不同的多个链。此外，因为数据本身例如由于使用擦除编码来将数据存储在不同链中的原因而在某种意义上是冗余的，所以也可以将容器链视为是冗余的。

[0078] 现在参照图 7，是依照本原理的实施例的数据组织结构 700，其图示了响应于存储节点网络中的存储节点的添加和 / 或存储在存储节点中的数据的添加而分裂、连结和删除数据并回收存储空间。行 702 示出了两个 synchrun A 704 和 B 706，这两者都属于覆盖整个哈希关键字空间的空前缀超级节点。这里，将每个 synchrun 组件放置在一个 SCC 中，还示出了单独的片段 708。具有这些 synchrun 的 synchrun 组件的 SCC 被示为相互前后放置的矩形。如上所述，可以用同伴链 SCC 链的超级节点基数来表示 synchrun 链。在图 7 的其余部分中，仅示出了一个这样的同伴 SCC 链。

[0079] 根据本原理的实施例，可以例如响应于加载数据或添加物理存储节点而最终将每个超级节点分裂。例如，如行 710 所示，所述分裂可以是规则 FPN 分裂，且可以得到包括各自的超级节点 SCC 链 712 和 714 的两个新超级节点，其具有从分别为 0 和 1 的祖先前缀扩展的前缀。在超级节点分裂之后，还可以将每个超级节点中的每个 synchrun 分裂成两半，片段基于其哈希前缀而分布在所述两半 synchrun 之间。例如，行 710 示出了两个此类链，一个链 712 用于具有前缀 0 的超级节点，且另一个链 714 用于具有前缀 1 的超级节点 714。请注意，作为分裂的结果，synchrun A704 和 B706 的片段 708 分布在这两个单独链 712 与 714 之间，所述链 712 和 714 可以存储在不同超级节点下的单独存储节点上。结果，创建四个 synchrun 716、718、720 和 722，但新 synchrun 716、718、720 和 722 中的每一个的尺寸大约分别是原始 synchrun 704 和 706 的一半。

[0080] 此外，应当理解，当向辅助存储系统添加物理存储节点且系统通过分裂超级节点进行响应时，该系统可被配置为以类似于上文相对于图 5 所述的方式向新超级节点和旧超级节点这两者分配物理存储节点。例如，辅助存储系统可以在所有超级节点之中均匀地分布物理存储节点。

[0081] 依照本发明的实施例的另一示例性特征，辅助存储系统可以保持有限数目的本地 SCC。例如，如图 7 的行 724 所示，可以通过将相邻 synchrun 组件合并或连结成一个 SCC，直至达到 SCC 的最大尺寸来保持 SCC 的数目。对本地 SCC 的数目进行限制允许将 SCC 元数据存储在 RAM 中，RAM 随后又使得提供数据服务的动作能够快速的确定。SCC 的目标尺寸可以是配置常数，例如，可以将其设置为 100MB 以下，因此可以在主存储器中读入多个 SCC。在所有同伴上，SCC 连结可以是松散地同步的，以便同伴链保持类似的格式。

[0082] 继续参照图 7，在图 7 的行 726 中图示了数据的删除，其中删除遮蔽的数据片段。随后，分别如行 730 和 732 所示，可以回收存储空间，并可以将分开的 SCC 的剩余数据片段连结。下面更全面地描述删除服务。

[0083] 在静态系统中实现上文相对于图 6 和 7 描述的数据组织相对简单，但在辅助存储系统的动态后端中十分复杂。例如，如果同伴在负载平衡期间被传输到另一物理存储，则其链可能在后台被传输到新位置，每次一个 SCC。类似地，依照示例性实施例，在超级节点分裂之后，并不是超级节点的所有 SCC 都被立即分裂；作为替代，辅助存储系统可以运行后台操作以将链调整到当前超级节点位置和形状。结果，在任何给定时刻，链可能部分被分裂，部分存在于同伴的先前位置上，或者两种情况同时发生。如果一个或者多个物理存储节点出现故障，基本的漏洞可能存在于某些 SCC 链中。由于同伴链可以由于系统中的超级节点基数链冗余的原因而描述相同的数据，所以应呈现足够数目的完整链以使得数据在线、哈希校验的全局重复去除模式。快速近似重复删除可以用于规则块，而可靠的重复去除可以用

于保留根以确保具有相同搜索前缀的两个或更多块指向相同的块。在这两种情况下，如果采用冗余等级，则正在写入的块的潜在副本应具有不弱于该写所请求的等级的冗余等级，且潜在的旧副本应是可重构的。这里，较弱冗余等级指示较低冗余。

[0084] 在写入规则块时，可以对处理写请求的同伴和 / 或已经活动最长时间的同伴实施对重复删除文件的搜索。例如，可以基于该块的哈希来选择该处理写的同伴，以便在此同伴活动时被写入的两个同样的块将由它来处理。因此，在该同伴中可以轻易地将第二块确定为第一块的重复。当近来由于数据传输或组件恢复的原因已经创建了写处理同伴且该同伴由于其本地 SCC 链不完整而尚未具有其应具有的所有数据时，出现更麻烦的情况。在这种情况下，检验与该写处理同伴在相同超级节点中的已活动最长时间的同伴以检查可能的重复。虽然检查活动时间最长的同伴仅仅是启发式措施，但活动时间最长的同伴不具有其完整的适当 SCC 链是不太可能的，这是因为这通常在严重故障之后发生。此外，对于特定块，即使在严重故障的情况下，也只错过一次消除副本的机会；下一个同样的块应被重复去除。

[0085] 对于在保留根上写，辅助存储系统应确保具有相同搜索前缀的两个块指向相同的块。否则保留根在标识快照时将不会是有用的。结果，应对保留根应用准确的、可靠的重复去除模式。类似于对规则块的写入，可以基于块的哈希来选择处理写的同伴，以使任何重复将存在于该同伴上。然而，当在处理写的同伴处不存在本地完整 SCC 链时，写处理同伴可以向其超级节点中的所有其它同伴发送重复去除查询。这些同伴中的每一个在本地检查重复。否定答复还可以包括该答复所基于的 SCC 链的各部分的简要说明。写处理同伴可以收集所有答复。如果存在至少一个肯定答复，则找到重复。否则，当所有答复均是否定的时，写处理同伴可以尝试使用附着于否定答复的任何链信息来创建完整的链。如果可以建立整个 SCC 链，则新块被确定为不是重复。否则，可以用指示数据重建正在进行中的特殊错误状态来拒绝保留根的写入，这可能在严重故障之后发生。如果整个链不能被覆盖，则应稍后提交写入。

[0086] 可以由辅助存储系统实施例执行的另一种数据服务包括数据删除和存储空间回收。如上所述，示例性辅助存储系统可以包括诸如内容可寻址性、分布和容错、以及重复去除之类的特征。这些特征在实现数据删除时引起复杂问题。如下文所讨论的，虽然内容可寻址的系统实施例中的删除稍微类似于得到透彻理解的分布式无用单元收集，但存在相当大的差异。

[0087] 当针对块的另一份旧拷贝来判定该块是否要被重复去除时，示例辅助存储系统实施例应确保该旧块不是预定要删除的。在分布式设定中和在存在故障的情况下，对保持哪个块和删除哪个块的确定应是一致的。例如，不应由于间歇的故障的原因而临时丢失删除确定，这是因为可以消除预定要删除的重复块。此外，数据删除算法的鲁棒性应高于数据鲁棒性。这种属性是理想的，这是因为即使某些块丢失，数据删除也应能够在逻辑上移除该丢失的数据，并在用户明确请求修复系统时执行此类动作。

[0088] 为了简化设计并使该实施方式在辅助存储系统的示例性实施例中可管理，可以将删除分为两个阶段：只读阶段，在此期间，块被标记为删除，且用户不能写数据；以及读写阶段，在此期间，回收被标记为删除的块，且用户可以发出读和写这两者。具有只读阶段简化了删除实施方式，这是因为其消除了写对块标记过程的影响。

[0089] 再次参照图 7，还可以用每块引用计数器来实现删除，所述每块引用计数器被配置

为计算指向特定块的系统数据块中的指针的数目。在某些实施方式中，引用计数器不需要在写时立即更新。作为替代，它们可以在每个只读阶段期间增量地更新，在所述只读阶段期间，辅助存储系统处理自先前只读阶段起所写入的所有指针。对于每个所检测的指针，对该块所指向的块引用计数器进行增值。在检测了所有指针并完成增值之后，可以将具有等于零的引用计数器的所有块标记为删除。例如，如图 7 所示，可以将片段 728 包括在被标记为删除的数据块中。此外，可以对已被标记为删除的块（包括具有相关删除根的根）所指向的块的引用计数器进行减值。此后，可以将具有由于减值的原因而等于零的引用计数器的任何块标记为删除，并可以对已被标记为删除的块所指向的块的引用计数器进行减值。可以重复进行标记和减值过程，直至没有另外的块可以被标记为删除。这时，只读阶段可以结束，且可以在后台移除被标记为删除的块。

[0090] 如上所述的示例性删除过程使用所有块的元数据以及所有指针。可以在所有同伴上复制所述指针和块元数据，因此，即使某些块不再可重构，也可以进行删除，只要在同伴上存在至少一个块片段即可。

[0091] 由于块可以被存储为片段，所以对于每个片段，可以存储块引用计数器的拷贝。因此，给定块的每个片段应具有块的引用计数器的相同值。可以在参与只读阶段的同伴上独立地计算引用计数器。在开始删除之前，每个这样的同伴应具有相对于片段元数据和指针来说完整的 SCC 链。并不是超级节点中的所有同伴都需要参与，但某最小数目的同伴应参与以完成只读阶段。稍后可以在后台将所计算的计数器传播 (propagate) 到其余同伴。

[0092] 计数器计算中的冗余允许任何删除确定经受住物理存储节点故障。然而，删除计算的中间结果不必是持久性的。在某些示例性实施例中，任何中间计算结果都可能由于存储节点故障的原因而丢失。如果存储节点出现故障，那么如果多同伴不能再参与只读阶段则可以重复整个计算。然而，如果每个超级节点中的足够数目的同伴未受到故障的影响，则删除仍可以继续。在只读阶段结束时，可以使新计数器值容许故障，且可以在后台从物理存储器中清除所有死块 (dead blocks)（即引用计数器等于零的块）。例如，可以如图 7 的行 730 所示地清除死块。

[0093] 现在参照图 8、9 和 10a ～ 10c 并继续参照图 2、3、5 和 6，图示了依照本原理的示例性实施方式的用于管理辅助存储系统上的数据的方法 800 和系统 200、600。应当理解，可以在方法 800 中及在系统 200 和 600 中实现示例性辅助存储系统的单独或以任何组合形式所采用的上述每个特征。因此，本文所讨论的方法 800 及系统 200 和 600 的特征仅仅是示例性的，且可以想到，根据本文所提供的讲授内容，如本领域的技术人员所理解的那样，可以将上文所讨论的特征添加到方法和系统实施方式。

[0094] 如例如上文相对于图 6 所述，在步骤 801，方法 800 可以任选地包括对数据块的进入 (incoming) 流应用哈希函数。例如，可以采用 SHA-1 哈希函数。

[0095] 任选地，在步骤 802，如上文相对于图 6 的方框 620 所述，例如可以对数据块进行擦除编码。

[0096] 在步骤 804，如上文相对于图 6 所述，例如可以将数据块分布到位于节点网络中的多个不同物理存储节点中的不同容器以便在节点中生成冗余数据容器链。例如，所述分布可以包括将经擦除编码的片段 622 存储在不同的数据容器 628 中，使得源自数据块之一，例如数据块 A604，的片段存储在不同的存储节点上。例如，所述不同存储节点可以对应于图 5

和 6 中的节点 514b 和 514d ～ f，其在超级节点 508 下。此外，如上所述，可以在存储节点的存储媒介上对数据块的片段进行内容寻址。另外，应注意的是，可以是内容地址的不同前缀与存储节点的不同子集相关联。例如，如上文相对于图 5 和 6 所讨论的，可以是哈希关键字前缀与不同的超级节点 506 ～ 512 相关联，超级节点 506 ～ 512 中的每一个可以跨越多个存储节点。此外，如上文所讨论的，对应于超级节点的数据容器链中的每一个可以包括相同的元数据，该元数据描述诸如像哈希值和恢复水平之类的数据块信息。另外，如上所述，元数据可以包括在数据容器中的数据块之间的暴露指针。

[0097] 在步骤 806，可以由一个或多个存储节点来检测活动存储节点到存储节点网络的添加。例如，可以由同伴组件通过接收指示添加和 / 或移除的管理命令来检测一个或多个存储节点的明确添加或移除。

[0098] 在步骤 808，可以响应于检测到活动存储节点的添加而分裂至少一个容器链。例如，可以如上文相对于图 7 的行 702 和 710 所描述的，分裂一个或多个容器链。应理解，该分裂可以包括分裂一个或多个数据容器 628 和 / 或分裂一个或多个数据容器 /synchrun 链 636。例如，分裂数据容器链可以包括将至少一个数据容器从该容器链分离。另外，可以在容器链分裂期间引用元数据以例如允许维持期望的冗余水平或响应于一个或多个添加的节点而实施负载平衡。另外，还应注意的是，如上文相对于图 7 所述，例如自动分裂可以包括扩展内容地址的至少一个前缀以生成新的超级节点或存储节点的子集。

[0099] 在步骤 810，可以将从至少一个容器链分裂的数据的至少一部分从存储节点之一传输到另一存储节点以提高针对节点故障的系统鲁棒性。如上文相对于图 7 所讨论的，例如可以将存储在 SCC 链的容器中的数据块片段分配和分布到不同的超级节点。如上所述，不同的超级节点可以包括不同的存储节点组件；同样地，从一个超级节点或节点子集到另一超级节点或节点子集的传输可以包括不同存储节点之间的传输。如上文所讨论的，可以响应于新存储节点到辅助存储系统的添加而执行分裂。因此，新节点的生成和它们之间的数据分布允许有效地利用网络中的存重构能够实现。因此，即使在存在传输 / 故障的情况下，在系统中链冗余也允许进行关于数据的扣减 (deduction)。

[0100] 基于上述数据组织结构，本原理的辅助存储系统实施例可以高效地给予数据服务，诸如确定数据可复原性、自动重建数据、均衡负载、删除和空间回收、数据定位、重复删除等。

[0101] 关于数据重建，如果存储节点或磁盘故障发生，那么常驻于其上的 SCC 可能丢失。结果，如果采用冗余水平，则具有属于这些 SCC 的片段的数据块的冗余被最好地降低至用户在写入这些块时所请求的冗余水平以下。在最坏的情况下，如果足够数目的片段未能幸存，则给定的块可能完全丢失。为了确保块冗余处于期望水平，例如，作为对于每个遗失 SCC 的后台工作，辅助存储系统可以扫描 SCC 链以搜索漏洞并基于擦除编码模式来调度数据重建。

[0102] 依照示例性实施例，可以在一个重建会话中重建多个同伴 SCC。例如，基于 SCC 元数据，可以由执行重建的同伴来读取用于数据重建的最小数目的同伴 SCC。其后，批量地对它们应用擦除编码和解码以获得将被包括在（一个或者多个）重建的 SCC 中的丢失片段。可以通过执行任何分裂和连结来将重建的 SCC 配置为具有相同的格式，这允许快速的批量重建。接下来，可以将重建的 SCC 发送到当前目标位置。

[0103] 可以由辅助存储系统实施例执行的另一种服务包括重复去除，其可以跨越存储节点而分散 (decentralize)，并且其可以以许多不同尺度来配置。例如，可以设置检测重复的级别 (level)，诸如整个文件、文件的子集、尺寸固定的块或尺寸可变的块。另外，可以设置检测重复删除的时间，诸如在线执行，当在存储之前检测到有重复时执行，或在其到达磁盘之后在后台中执行。可以调整重复删除的准确度。例如，可以将系统设置为每次检测正在写入的对象的重复存在，这可以称为“可靠的”，或者系统可以在获得更快性能的情况下近似 (approximate) 重复文件的存在，这可以称为“近似”。还可以设置用以校验两个对象的等同性的方式，例如，可以将系统配置为比较两个对象内容的安全哈希，或者可替换地直接比较这些对象的数据。此外，检测的范围可以改变，这是因为其可以是本地的、仅限于存在于给定节点上的数据、或全局的，在全局的情况下使用来自所有节点的所有数据。

[0104] 在优选实施例中，辅助存储系统在存储节点上实现尺寸可变的块、储节点，使得数据存储多样化，从而提供针对一次或多次节点故障的鲁棒性。如果故障发生，冗余数据在不同存储节点上的广泛可用性便于数据重构。

[0105] 在步骤 812 之后，可以将至少一个数据容器与另一数据容器合并。如上文相对于图 7 所讨论的，例如可以将包括数据容器的 synchrun 716 和 720 分别与 synchrun 718 和 722 合并，synchrun 718 和 722 在被分裂为例如保持一定数目的 SCC 之后也包含数据容器。此外，如上文相对于图 7 的行 726、730 和 732 所讨论的，例如还可以在删除和回收之后执行合并。

[0106] 在执行方法 800 的任何部分期间，还可以在步骤 814 执行一种或多种数据服务。虽然步骤 814 被示为在方法 800 的结尾，但其可以在方法 800 的任何其它步骤期间、之间或之后执行。例如，可以执行该方法以帮助实现诸如负载平衡之类的数据服务。如图 9 所示，数据服务的执行可以包括将数据容器链从一个存储节点或超级节点组件传输 902 到另一存储节点或超级节点组件。依照时间顺序来组织同伴链便于数据容器链的传输。

[0107] 在步骤 904 和 908，可以执行数据写和读。例如，如上文所讨论的，在写期间，可以将数据块路由到分配给该块所属的哈希关键字空间的超级节点的同伴之一。另外，还可以在写入时执行重复检测。关于读，例如，如上文所讨论的，可以将读请求路由到负责数据块的哈希关键字的超级节点的同伴之一。如上文所讨论的，读取块可以包括读取块元数据和向其它同伴传输片段读请求以获得足够数目的片段以便依照擦除编码模式来重构该块。

[0108] 在步骤 906，可以确定数据容器链是否具有漏洞。标识数据容器链中的漏洞例如便于数据读取、确定数据可用性、执行数据删除、响应于故障而重建数据、以及执行分布式全局重复去除。例如，漏洞的标识指示存储在容器中的数据片段不可用。结果，存储服务器应在数据重构或重建期间在另一同伴搜索其它数据片段。例如，可以由数据读取来触发数据的重建。类似地，可以在对在系统上是否保持用户定义冗余水平进行系统测试期间执行漏洞的标识。

[0109] 图 10A ～ 10C 图示了存储服务器可以确定容器链是否包含漏洞的一个示例，其指示超级节点的同伴上的 synchrun 链扫描的不同时间帧。例如，在图 10A 中，存储服务器可被配置为在属于超级节点 508 的所有同伴上同时扫描 synchrun 1002。在图 10B 中，系统可以按照数据块流的时间顺序在属于超级节点 508 的所有同伴上同时扫描下一个 synchrun1008，并发现漏洞 1004。类似地，在图 10C 中，系统可以按照数据块流的时间顺序

在属于超级节点 508 的所有同伴上同时扫描下一个 synchrun 1010 并发现漏洞 1006。这样，例如，可以检测到由于节点和磁盘故障而引起的链漏洞。此外，如上文所讨论的，可以使用链和数据冗余来重建该链。

[0110] 返回图 9，如上所述，在步骤 910，可以执行数据可用性的确定。另外，还可以在步骤 912 执行响应于存储节点故障的数据重建。如上文所讨论的，例如系统可以扫描 SCC 链以查找漏洞并调度数据重建以作为每个遗失 SCC 的后台操作。此外，如上所述，可以通过引用片段元数据和 / 或容器元数据来执行重建。

[0111] 在步骤 914，可以执行数据删除。如上文所讨论的，例如可以应要求和 / 或与重复删除相关地执行数据删除。另外，数据删除例如可以包括使用引用计数器并重复地删除不具有指向其的任何指针的任何块。例如，可以通过引用片段元数据来获得指针。

[0112] 在步骤 916，如上文详细地讨论的，可以执行分布式全局重复去除。例如，如上文所讨论的，可以对规则块执行快速近似重复删除，同时可以对保留根执行可靠重复删除。此外，可以在线执行重复去除以作为数据写入的一部分。

[0113] 应当理解，图 9 中描述的所有服务都是任选的，但优选系统包括执行相对于图 9 所提及的所有服务的能力。另外，虽然步骤 902 和 906 被示为在步骤 908 ～ 916 之前执行，但可以在不执行步骤 902 和 906 中的任何一个或多个的情况下执行步骤 908 ～ 916 中的任何一个。

[0114] 此外，返回图 2 和 3，应当理解，在系统 200 的后端中的存储节点上运行的存储服务器可被配置为执行上文相对于图 8 和 9 所述的步骤中的任何一个或多个。因此，应将下文所提供的对系统 200 的后端的说明理解为仅仅是示例性的，且其中可以包括上文所讨论的特征中的任何一个或多个。

[0115] 如上所提及，系统 200 的后端 212 可以包括物理存储节点 218a ～ 218f 的网络。另外，每个存储节点可以包括存储媒体和处理器，其中存储媒体可被配置为将数据块的片段存储在相对于其它存储节点中的数据容器链冗余的数据容器链中。例如，如上文相对于图 6 所讨论的，可以将数据块 604、610 和 616 的片段 622 存储在同伴 01:0 中的相对于存储在其它同伴 01:1 ～ 01:3 中容器链冗余的数据容器链 636 中。

[0116] 此外，如上文相对于图 8 所讨论的，每个存储服务器 302 可被配置为执行步骤 806、808 和 810。另外，每个存储服务器可被配置为执行相对于图 9 所讨论的数据服务中的任何一种。此外，如上文所讨论的，一个或多个数据容器链可以包括与其它数据容器链相同的元数据。所述元数据可以描述存储节点中的数据的状态。而且，如上文所讨论的，存储服务器可以引用元数据以自动地分裂与存储服务器相关联的存储媒体上的至少一个容器链。此外，如上所述，元数据可以包括数据容器中的数据块之间的暴露指针，且存储服务器可被配置为使用该指针来执行数据删除。

[0117] 如上文相对于步骤 801 和图 6 所描述的，可以基于哈希函数对数据块及其相应片段进行内容寻址。类似地，可以将哈希关键字的前缀与不同的超级节点或存储节点子集相关联。此外，如图 5 所示，可以将前缀“00”与存储节点 514a ～ 514d 的子集相关联，可以将前缀“01”与存储节点 514b 和 514d ～ f 的子集相关联。此外，如上文相对于步骤 808 所述，自动地分裂容器链可以包括扩展前缀中的至少一个以生成存储节点的至少一个附加子集。例如，如上文所讨论的，如果分配给超级节点的前缀是“01”，则可以将该超级节点分裂成分

别具有所分配的前缀“010”和“011”的两个超级节点。另外，每个新超级节点可以包括新的且不同的组件或同伴集或与其相关联的存储节点的子集。此外，如上文相对于步骤 810 所讨论的，由存储服务器开始的传输可以包括将从所述一个或多个容器链分裂的数据的至少一部分分布到新生成的或附加的存储节点子集。

[0118] 现在在参考图 7 和 8 的情况下参照图 11，图示了依照本原理的另一示例性实施方式的用于管理辅助存储系统上的数据的方法 1100。应当理解，可以在方法 1100 中实现示例性辅助存储系统和方法的单独或以任何组合形式所采用的上述每个特征。因此，下文所讨论的方法 1100 的特征仅仅是示例性的，且可以想到，根据本文所提供的讲授内容，如本领域的技术人员所理解的那样，可以将上文所讨论的特征添加到方法 1100。

[0119] 类似于图 800，如上文相对于图 8 所讨论的，方法 1100 可以通过执行任选步骤 801 和 802 开始。另外，可以执行步骤 804，在步骤 804 中，如上文相对于图 8 所讨论的，数据块被分布到节点网络中的不同存储节点的不同数据容器以便在节点中生成冗余数据容器链。

[0120] 在步骤 1106，一个或多个存储服务器可以检测存储节点网络中的活动存储节点的数目的变化。活动节点的数目的变化可以包括将至少一个存储节点添加到存储节点网络和 / 或从节点网络移除节点。如上文所讨论的，可以由同伴组件或其相应存储服务器通过接收指示添加和 / 或移除的管理命令来检测节点的添加和移除。此外，应注意的是，可以由同伴组件或其相应存储服务器通过采用 ping 来检测节点故障。例如，同伴组件可被配置为周期性地相互 ping 并在检测到几个 ping 遗失之后推断节点已经出现故障。

[0121] 在步骤 1108，存储服务器可被配置为响应于检测到所述变化而自动地将位于存储节点之一中的至少一个数据容器与位于不同存储节点中的另一数据容器合并。例如，如果向网络添加了节点，则可以如上文相对于图 7 的行 710 和 724 所讨论的合并数据容器。例如，容器 718 在与容器 716 合并之前可能源自不同的存储节点。

[0122] 可替换地，如果从存储系统移除节点，则存储服务器可被配置为在步骤 1106 将来自不同存储节点的数据容器合并。例如，存储服务器可以接收指示节点将被移除的管理命令。在实际移除该节点之前，存储服务器可被配置为将要移除的节点中的数据容器与其余节点中的容器合并。例如，可以简单地将上文相对于图 7 的行 702、710 和 724 所述的过程颠倒，以使将例如容器 718 和 720 的来自不同存储节点的容器合并成较大的 synchrun 链。可以执行该合并以确保容器的可管理性和 / 或改善系统性能。其后，如上文所讨论的，可以执行重新分布或再平衡。此外，例如，如上文相对于图 8 所讨论的，可以在方法 1100 的任何点处执行步骤 814。

[0123] 还应注意的是根据本发明的示例性方法和系统可被配置为将管理性节点添加 / 移除与系统中的节点故障 / 恢复区分开。可以在应在系统中的节点的管理列表中指示管理性节点添加 / 移除。这种区分在自动系统管理中有用。例如，可以采用该区分来检测根据节点的管理列表不应被连接到系统的外来或未经授权节点。例如，当外来节点尝试与系统连接时，该连接可能被拒绝，或者可以通过采用节点的管理列表来移除该外来节点。还可以利用所述区分来计算处于所有节点都活动的健康状态的系统的预期原始容量，并区别健康状态和非健康状态。还可以想到所述区分的其它使用。

[0124] 上述示例性方法和系统便于高效且有效地在辅助存储系统中提供若干数据服务，诸如全局重复删除、动态可变动性、对多个冗余等级的支持、数据定位、数据的快速读写和

由于节点或磁盘故障的原因而引起的数据重建。上文所讨论的基于响应于网络配置变化而被配置为分裂、合并和被传输的冗余数据容器链的示例性数据组织结构允许在分布式辅助存储系统中实现这些服务中的每一种。作为示例性实施例的若干特征之一的链式容器冗余允许在传送数据服务过程中容许故障。例如，如果故障发生，即使数据丢失，也可以进行数据删除，这是因为元数据由于该链中的多份复制品的原因而得以保存。此外，如上文所讨论的，冗余还允许高效的分布式数据重建。另外，数据容器中的数据块的按时间顺序存储和概要容器元数据使得快速度推理系统状态能够实现并允许诸如数据重建之类的操作。包括指向其它块的暴露指针的数据块元数据允许实现分布式容错数据删除。此外，链式容器中的动态性允许高效的可变动性。例如，容器可以被分裂、传输、和 / 或合并以便以可全面优化并利用存储空间来提供数据服务的方式自动地适应于存储节点网络配置的变化。此外，动态性还允许容易地进行数据定位。

[0125] 虽然已描述了系统和方法的优选实施例（其意图是说明性而非限制性的），但应注意的是根据上述讲授内容，本领域的技术人员可以进行修改和变更。因此，应理解的是可以对所公开的特定实施例进行修改，所述特定实施例在随附权利要求所阐述的本发明的范围和精神内。虽然已因此描述了本发明的各方法，以及要求专利法保护的细节和特性，但要求并期望受到专利证书保护的在随附权利要求中阐述。

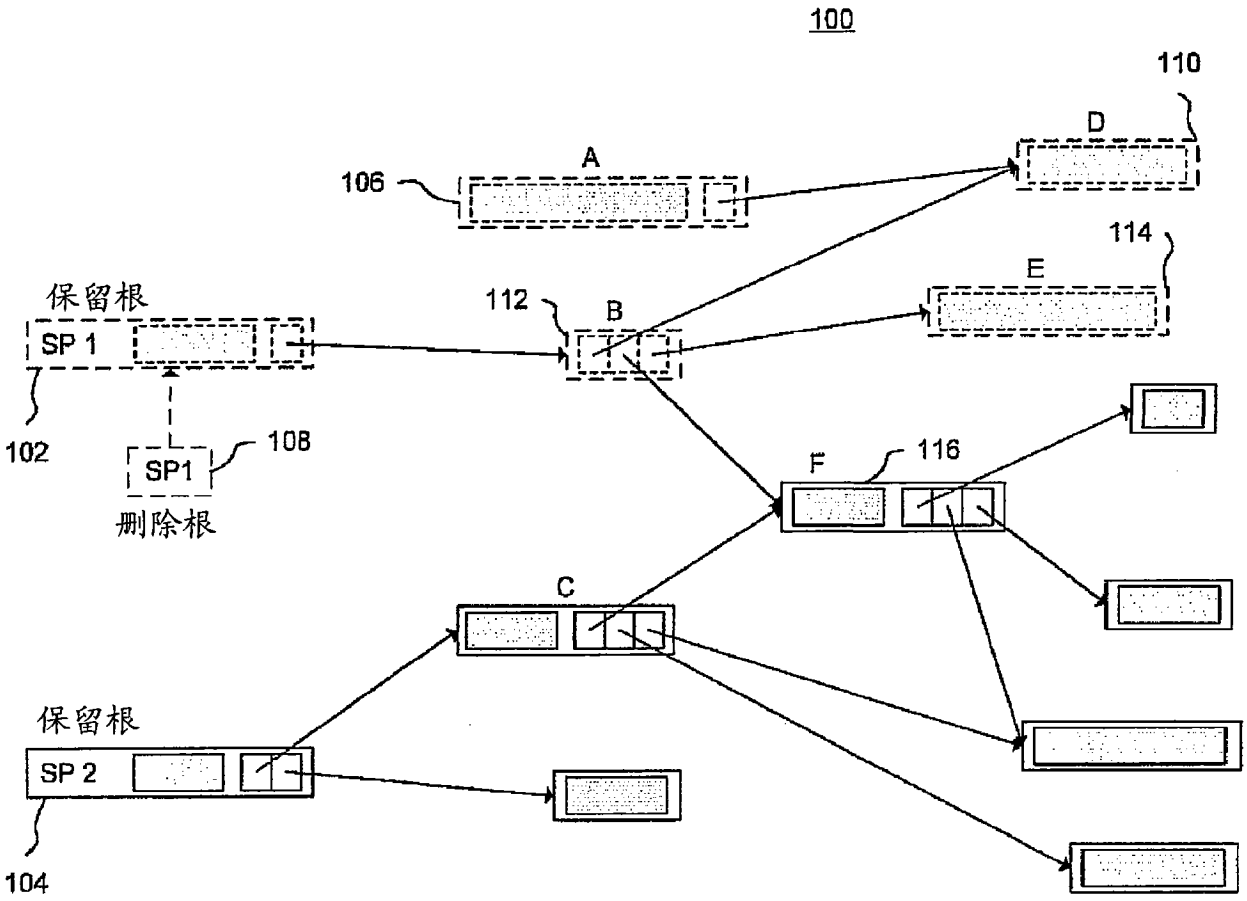

100
110
D
A
106
114
E
保留根
SP 1
112
B
102
SP1
108
删除根
F
116
C
保留根
SP 2
104

图 1

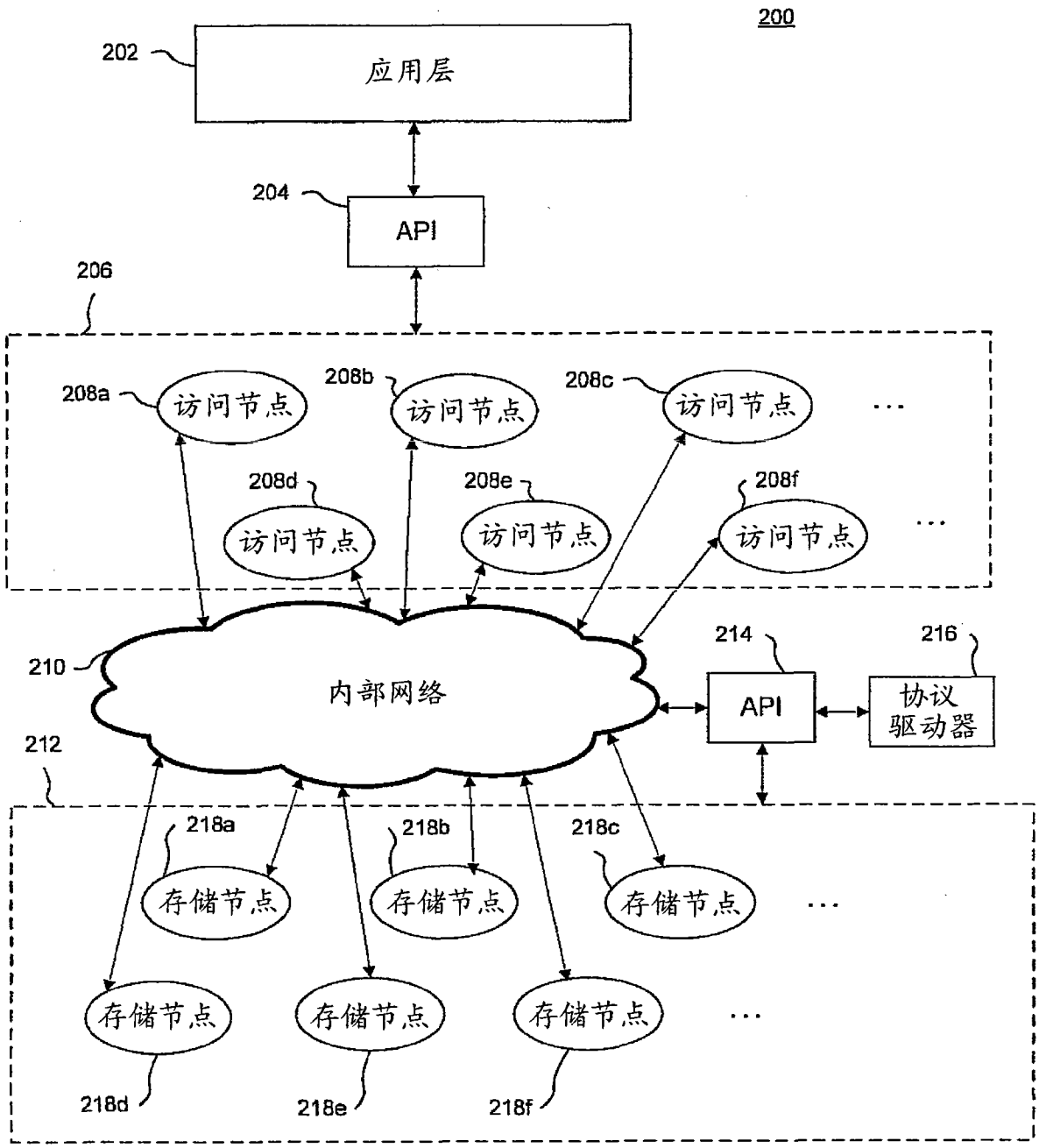

200
202
应用层
204
API
206
208a
访问节点
208b
访问节点
208c
访问节点
208d
访问节点
208e
访问节点
208f
访问节点
210
内部网络
214
API
216
协议驱动器
212
218a
存储节点
218b
存储节点
218c
存储节点
存储节点
存储节点
存储节点
218d
218e
218f

图 2

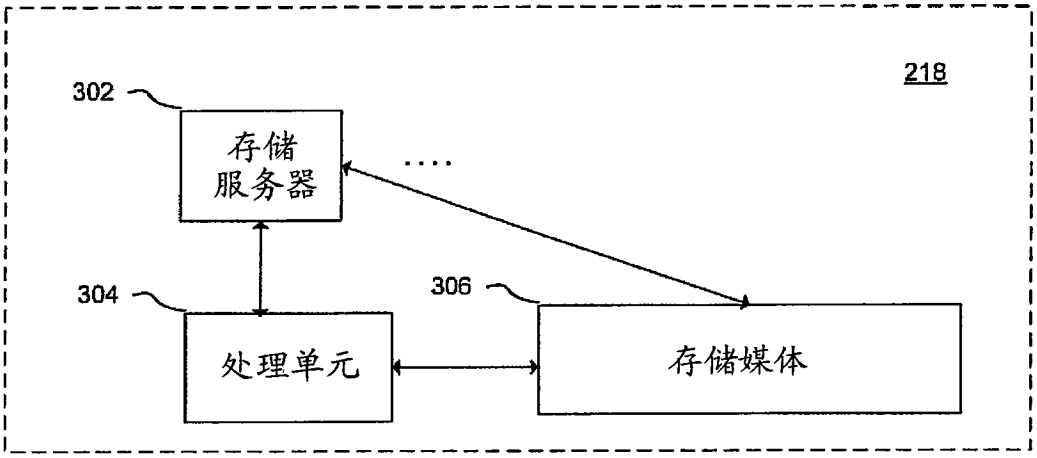

218
302
存储
服务器
304
处理单元
306
存储媒体

图 3

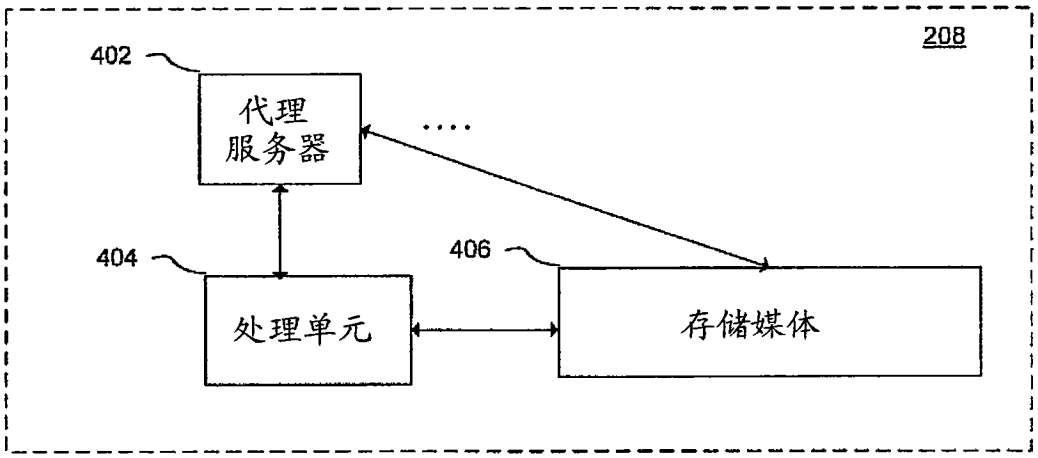

208
402
代理
服务器
404
处理单元
406
存储媒体

图 4

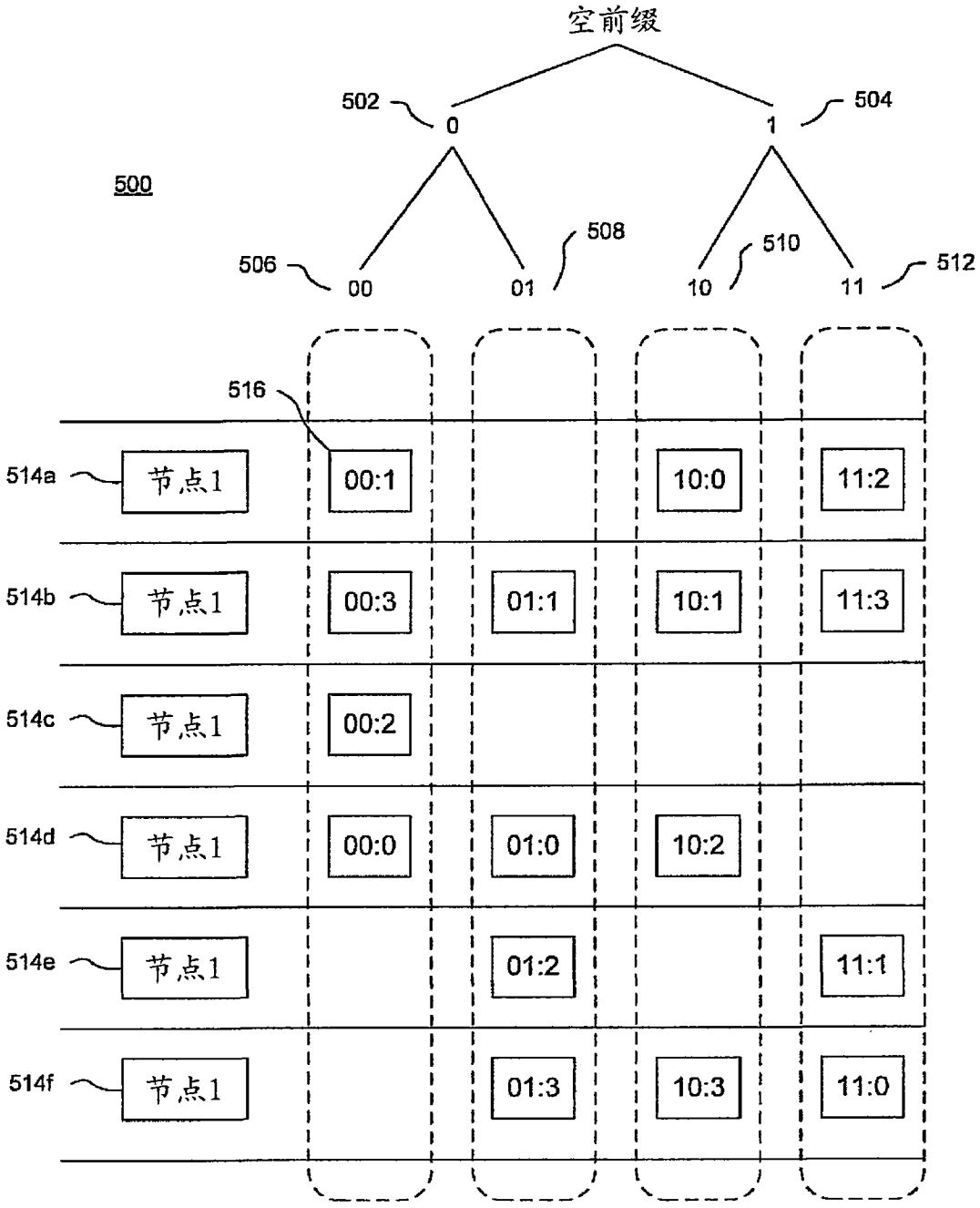

空前缀
502
0
1
504
500
506
00
01
508
10
510
11
512
516
514a
节点1
00:1
10:0
11:2
514b
节点1
00:3
01:1
10:1
11:3
514c
节点1
00:2
514d
节点1
00:0
01:0
10:2
514e
节点1
01:2
11:1
514f
节点1
01:3
10:3
11:0

图 5

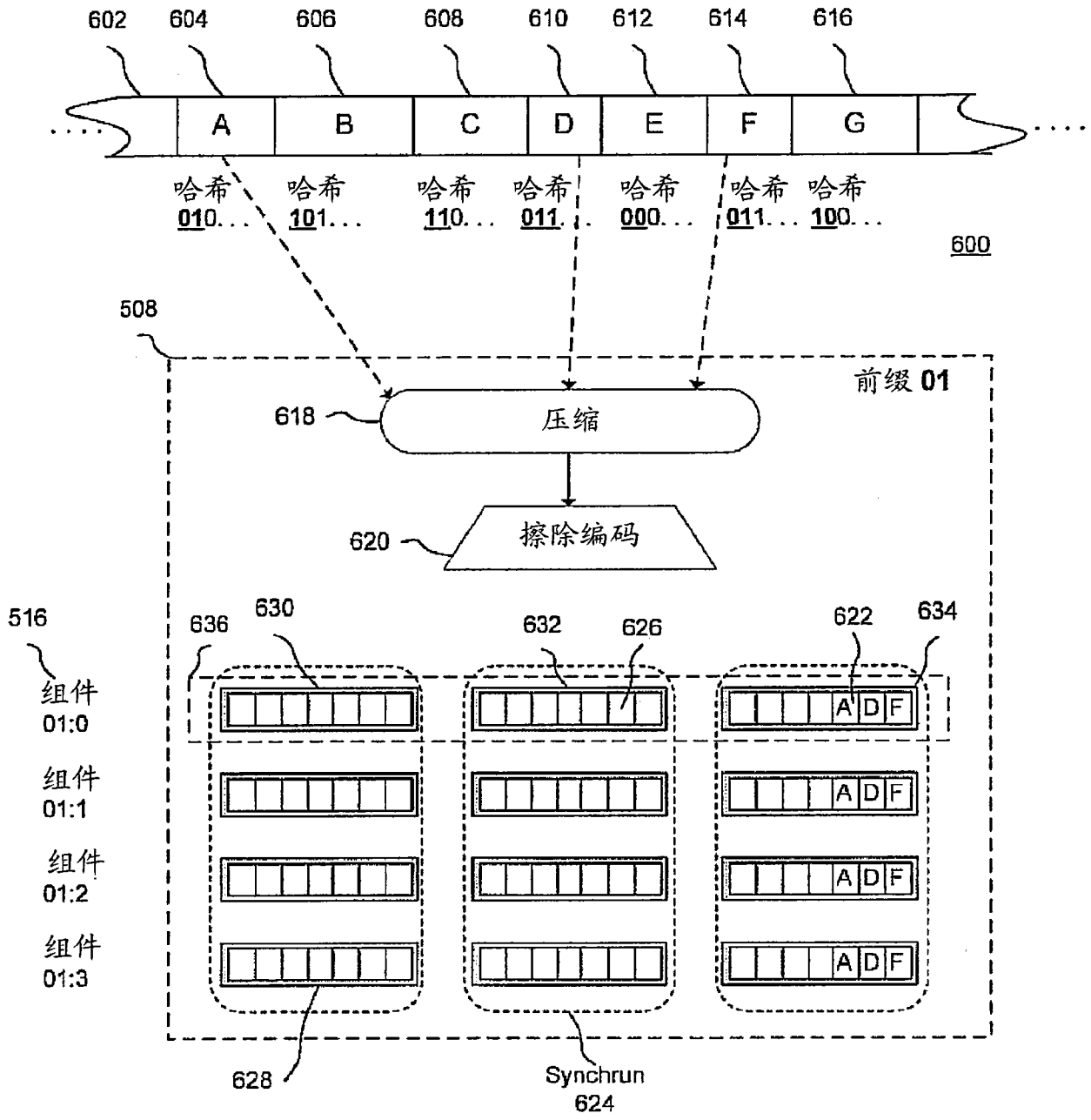

602
604
606
608
610
612
614
616
A
B
C
D
E
F
G
哈希 010...
哈希 101...
哈希 110...
哈希 011...
哈希 000...
哈希 011...
哈希 100...
600
508
前缀 01
618
压缩
620
擦除编码
516
636
630
632
626
622
634
组件 01:0
组件 01:1
组件 01:2
组件 01:3
A D F
628
Synchrun 624

图 6

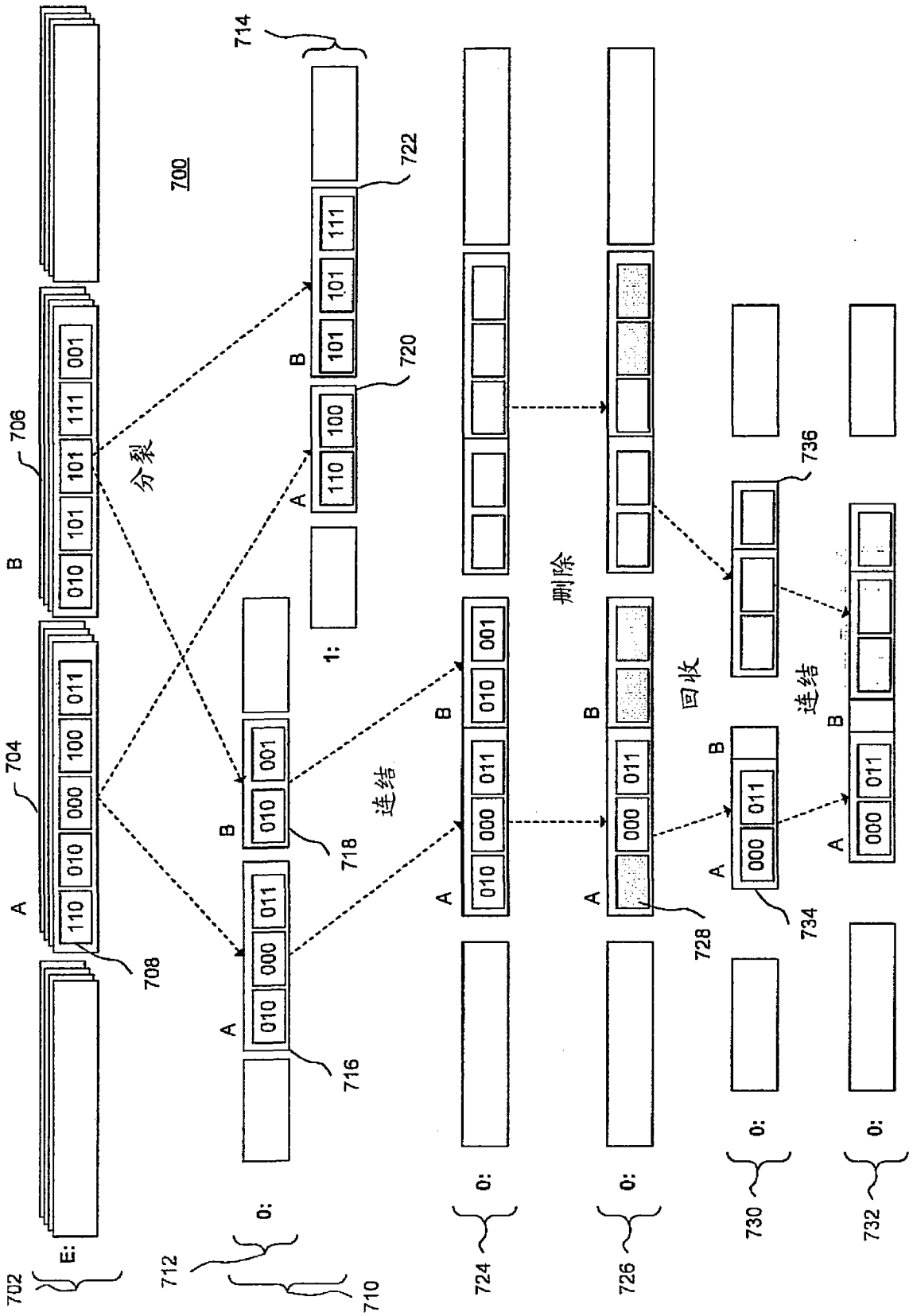

700
分裂
连结
删除
回收
连结
E:
0:
1:
A
B
000
001
010
011
100
101
110
111
702
704
706
708
710
712
714
716
718
720
722
724
726
728
730
732
734
736

图 7

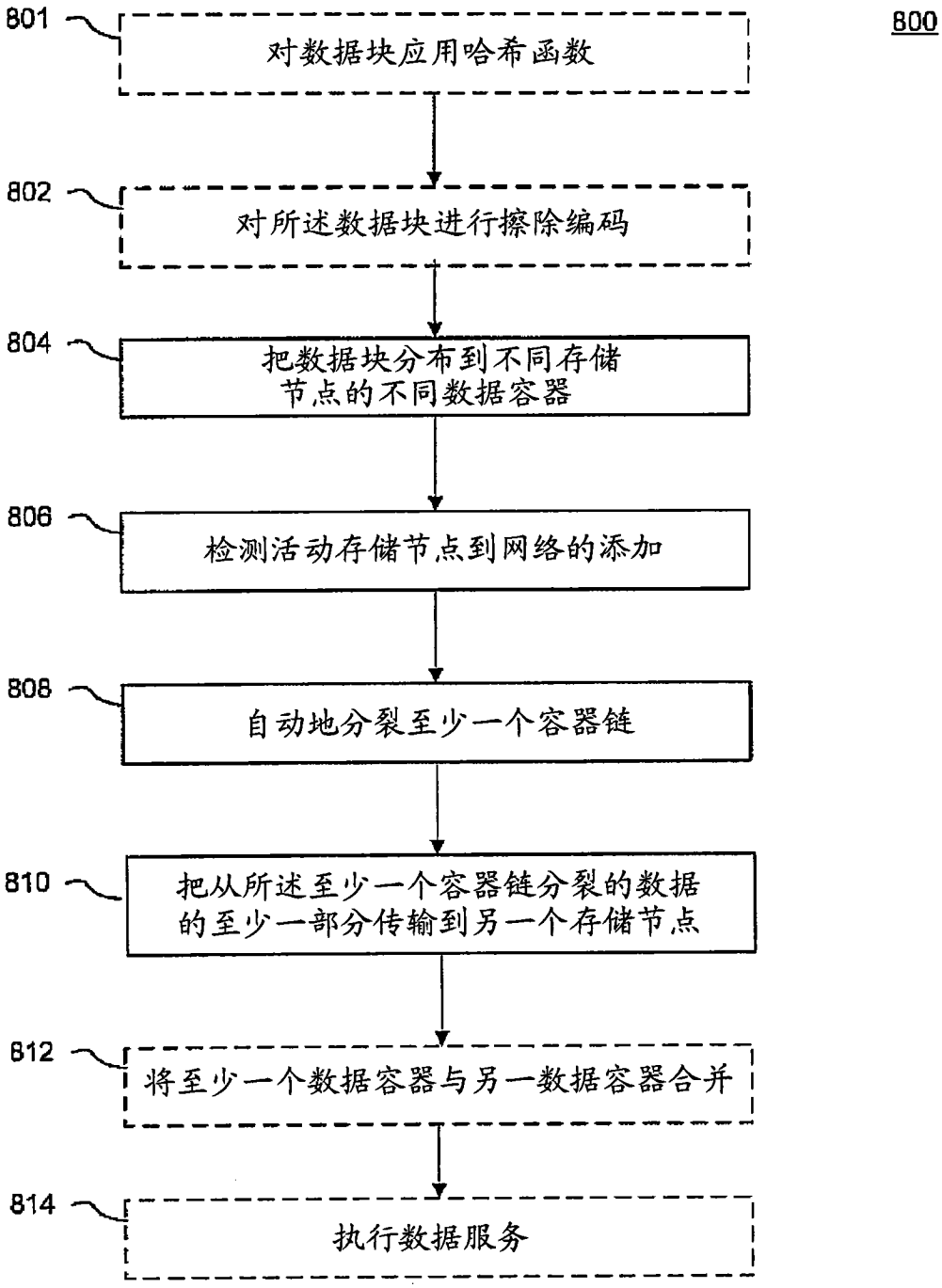

800
801
对数据块应用哈希函数
802
对所述数据块进行擦除编码
804
把数据块分布到不同存储节点的不同数据容器
806
检测活动存储节点到网络的添加
808
自动地分裂至少一个容器链
810
把从所述至少一个容器链分裂的数据的至少一部分传输到另一个存储节点
812
将至少一个数据容器与另一数据容器合并
814
执行数据服务

图 8

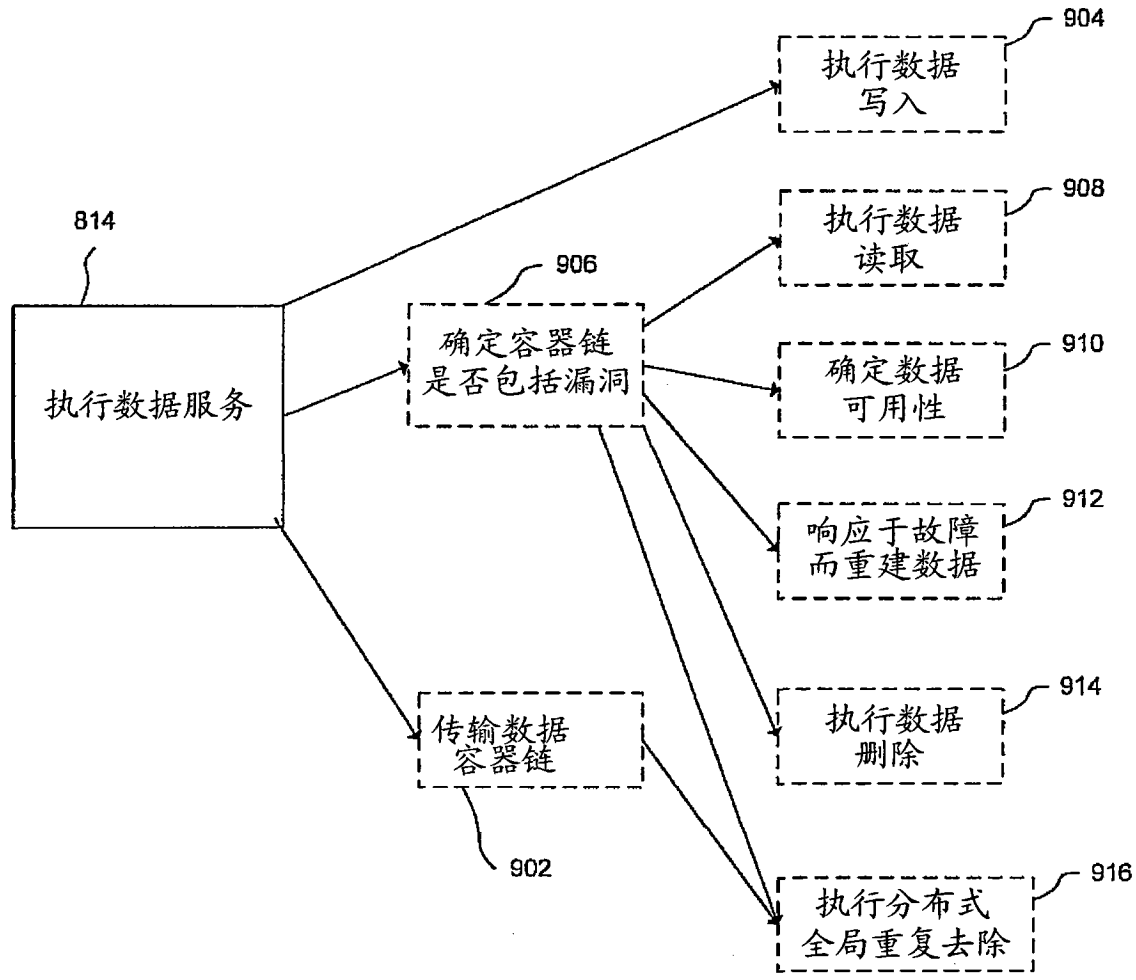

814
执行数据服务
906
确定容器链是否包括漏洞
传输数据容器链
902
904
执行数据写入
908
执行数据读取
910
确定数据可用性
912
响应于故障而重建数据
914
执行数据删除
916
执行分布式全局重复去除

图 9

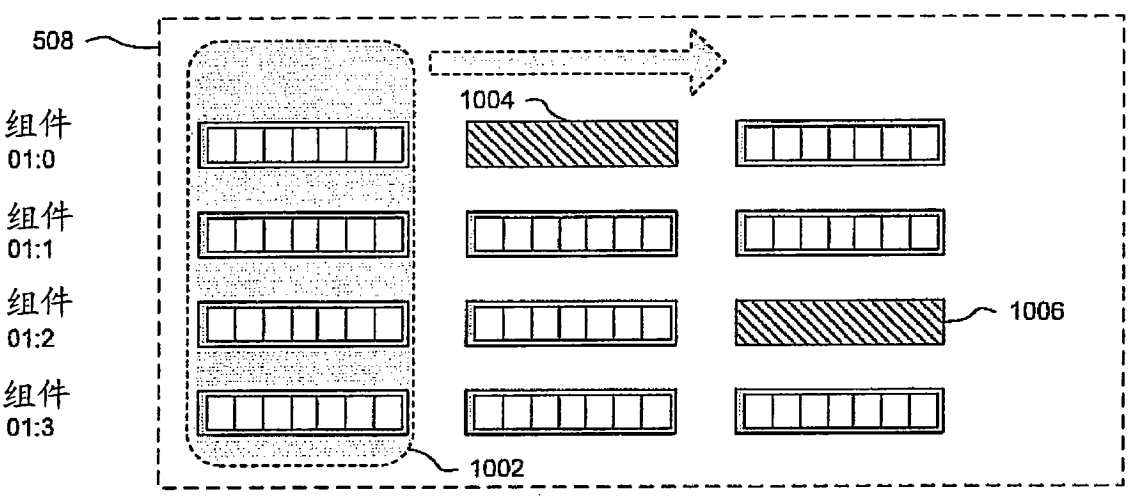

508
组件 01:0
组件 01:1
组件 01:2
组件 01:3
1004
1006
1002

图 10A

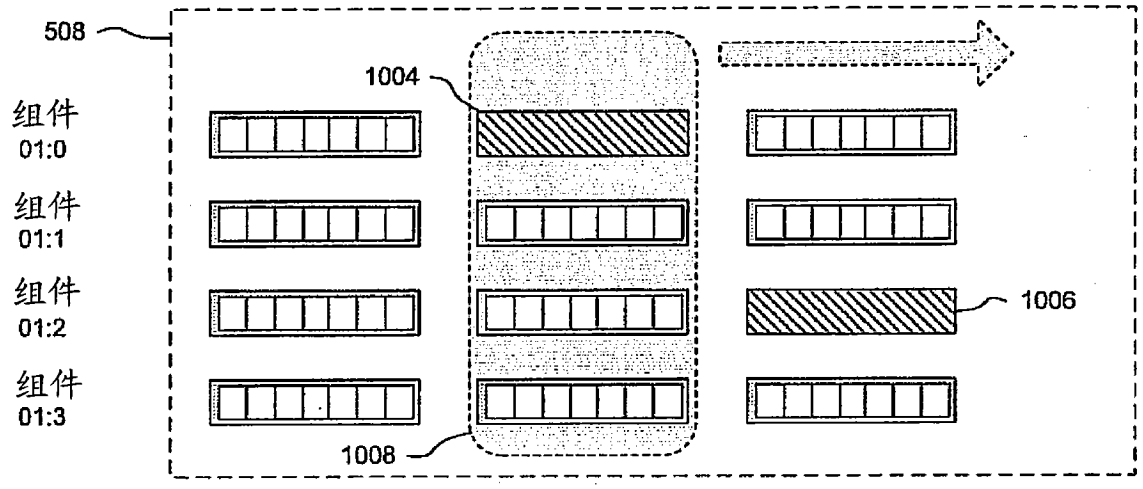

508
1004
组件
01:0
组件
01:1
组件
01:2
组件
01:3
1006
1008

图 10B

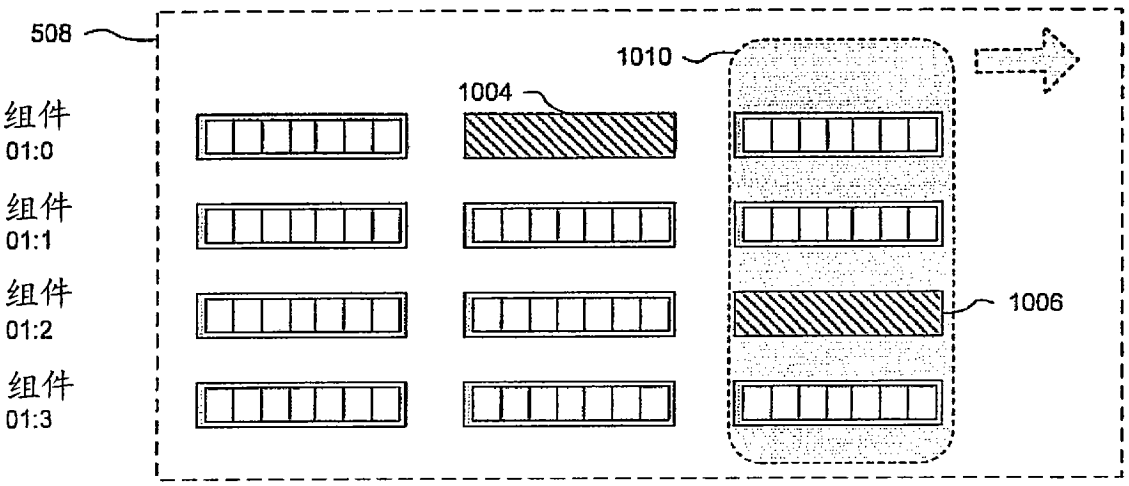

508
1010
1004
组件
01:0
组件
01:1
组件
01:2
组件
01:3
1006

图 10C

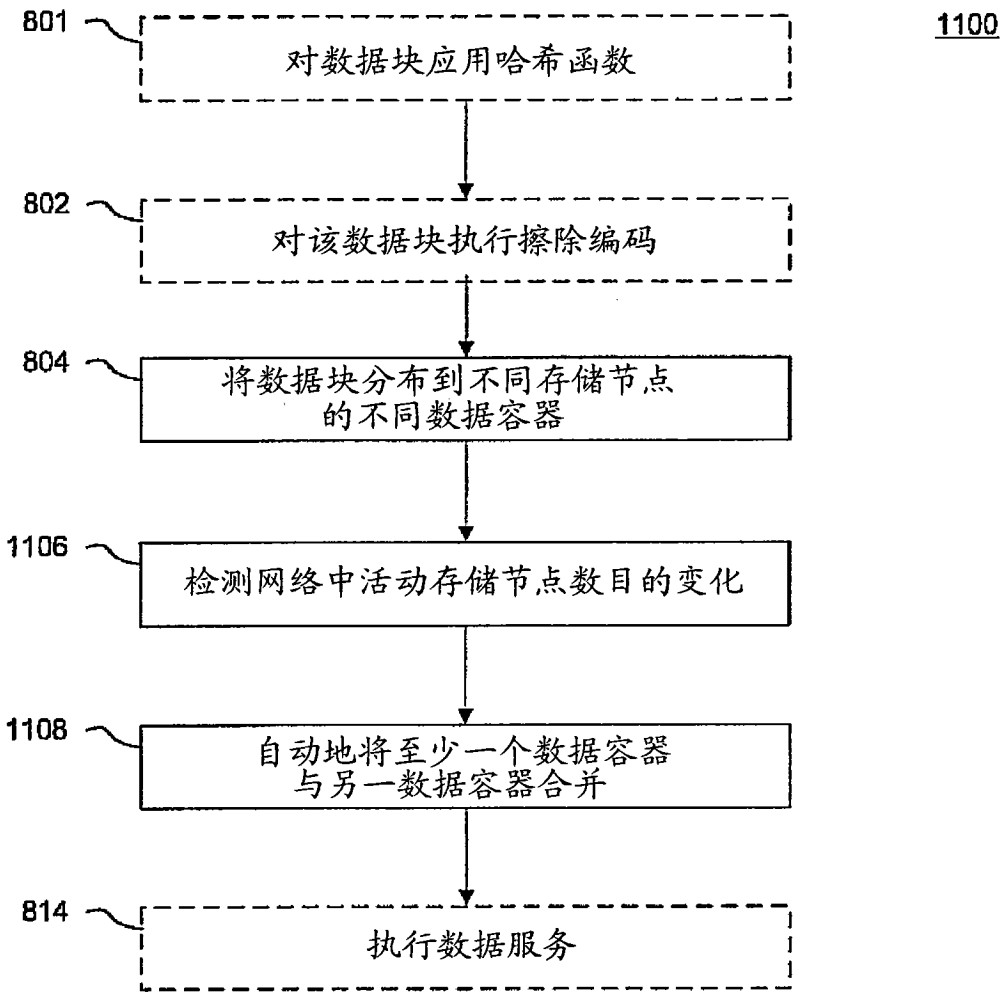

801
对数据块应用哈希函数
1100
802
对该数据块执行擦除编码
804
将数据块分布到不同存储节点的不同数据容器
1106
检测网络中活动存储节点数目的变化
1108
自动地将至少一个数据容器与另一数据容器合并
814
执行数据服务

图 11