



(12)发明专利

(10)授权公告号 CN 103353867 B

(45)授权公告日 2018.08.14

(21)申请号 201310189247.8

E·M·瓦格纳

(22)申请日 2006.11.30

(74)专利代理机构 中国国际贸易促进委员会专

(65)同一申请的已公布的文献号

利商标事务所 11038

申请公布号 CN 103353867 A

代理人 赵冰

(43)申请公布日 2013.10.16

(51)Int.Cl.

G06F 17/30(2006.01)

(30)优先权数据

60/754,726 2005.12.29 US

11/371,304 2006.03.08 US

(62)分案原申请数据

200680053577.2 2006.11.30

(73)专利权人 亚马逊科技公司

地址 美国内华达

(56)对比文件

US 6704886 B1,2004.03.09,

US 6353878 B1,2002.03.05,

US 2004230572 A1,2004.11.18,

CN 1602479 A,2005.03.30,

CN 1848849 A,2006.10.18,

审查员 杨庆丽

(72)发明人 A·H·弗穆伦 A·B·阿特拉斯

D·M·巴斯 J·D·科米

A·K·弗西曼 J·C·索伦森三世

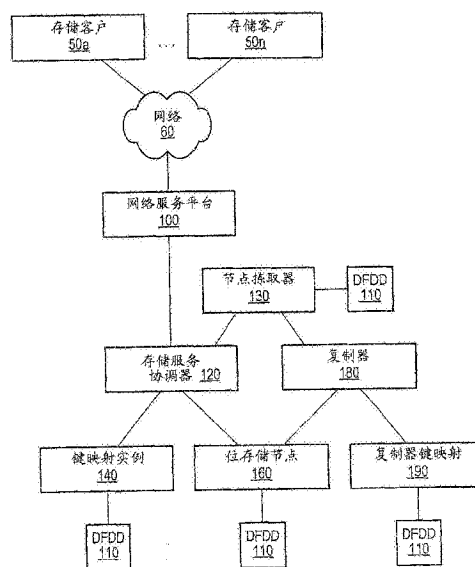
权利要求书3页 说明书59页 附图25页

(54)发明名称

具有网络服务客户接口的分布式存储系统

(57)摘要

一种分布式的基于网络服务的存储系统。系统可包括配置成根据网络服务协议接收对给定的数据对象的访问的给定客户请求的网络服务接口,该请求包括相应于对象的键值。系统还可包括配置成存储对象的复本的存储节点,和配置成为每个对象存储相应的键映射项目的键映射实例,其中通过相应的唯一定位器值可访问每个复本。对于给定的对象,相应的键映射项目包括键值和相应于对象的复本的每个定位器值。协调器可从网络服务接口接收给定的客户请求,响应性地访问键映射实例以识别相应于键值的定位器值,且对于特定的定位器值,从相应的存储节点取回相应的复本。



1. 一种基于网络服务的数据存储系统,包括:

网络服务平台,其配置成向多个客户提供数据存储网络服务,其中所述数据存储网络服务包括一个或多个网络服务端点;

其中每个网络服务端点被配置成实现应用程序编程接口API用于经由基于互联网的通信协议从所述客户接收网络服务调用以访问所述数据存储网络服务,其中至少一些网络服务调用指定相应的数据对象的对应标识符和操作,其中所述API指定至少两个可用操作,一个用于存储相应的数据对象,并且另一个用于检索相应的数据对象;以及

其中响应于经由API接收网络服务调用中的一个,所述网络服务平台配置成指示数据存储网络服务在对应于由所述网络服务调用所指定的标识符的相应数据对象上执行操作;

其中为存储所述相应的数据对象,所述网络服务平台被配置成从客户接收相应的数据对象并指示所述数据存储网络服务存储从所述客户接收的相应的数据对象;

其中为检索所述相应的数据对象,所述网络服务平台被配置成指示所述数据存储网络服务检索所述相应的数据对象并将所述相应的数据对象发送到所述客户;以及

其中所述网络服务平台被进一步配置成:

取决于网络服务调用,确定每个客户的数据存储网络服务的实际使用量;以及

存储所述数据存储网络服务的实际使用量的信息特征。

2. 如权利要求1所述的系统,其中所述数据对象的标识符表明键,并且其中为指示数据存储网络服务执行操作,所述网络服务平台被进一步配置成指定对应于所述数据对象的键。

3. 如权利要求1所述的系统,其中所述数据对象的标识符表明对应于数据存储网络服务的用户的命名空间,并且其中为指示所述数据存储网络服务执行操作,所述网络服务平台被进一步配置成指定与所述命名空间相关的数据对象。

4. 如权利要求3所述的系统,其中所述网络服务平台被进一步配置成:

响应于经由所述API接收所述网络服务调用,取决于访问策略确定所述客户是否具有充分的特权在所述数据对象上执行操作;以及

响应于根据访问策略确定客户不具有充分的特权执行操作,拒绝在所述数据对象上的操作。

5. 如权利要求4所述的系统,其中为确定所述客户是否具有充分的特权,所述网络服务平台被进一步配置成评估所述访问策略所表明的一个或多个项目,其中所述一个或多个项目包括下列中的一个或多个:表明用户具有对至少所述数据对象的访问的用户标识符,配置成对用户进行身份验证的身份验证凭证,或者指定所述用户对至少所述数据对象所具有的访问类型的信息。

6. 如权利要求1所述的系统,其中为接收网络服务调用,所述网络服务平台被配置成接收编码所述数据对象的标识符的统一资源定位符(URL)。

7. 如权利要求1所述的系统,其中为接收网络服务调用,所述网络服务平台被进一步配置成接收编码存储或检索所述数据对象的操作的统一资源定位符(URL)。

8. 如权利要求1所述的系统,其中所述API指定操作,以响应于客户请求在对象命名空间内创建新的存储桶用于为客户存储数据对象。

9. 如权利要求1所述的系统,其中所述API指定操作,以响应于客户请求,列出与给定存

储桶相关联的对象。

10. 如权利要求9所述的系统,其中所述API指定操作,以响应于客户请求,读取与给定存储桶相关联的元数据,并且其中所述元数据包括表明下列中的一个或多个的信息:与所述给定存储桶相关联的使用历史,与所述给定存储桶相关联的计费历史,或者与所述给定存储桶相关联的状态信息。

11. 如权利要求9所述的系统,其中所述API指定操作,以响应于客户请求,读取或写入与给定存储桶相关联的访问策略。

12. 如权利要求1所述的系统,其中所述网络服务平台被进一步配置成确定所述操作是存储数据对象的操作,并且响应性地指示所述数据存储网络服务在多个存储节点的一个或多个中冗余地存储数据对象。

13. 一种基于网络服务接口存储数据的方法,包括:

经由应用程序编程接口API从客户接收网络服务调用以访问数据存储网络服务,所述数据存储网络服务被配置为向多个客户提供数据存储网络服务,其中所述网络服务调用指定数据对象的标识符和用于存储所述数据对象的操作;以及

响应于经由API接收网络服务调用,指示数据存储网络服务存储从所述客户接收的数据对象;

其中指示所述数据存储网络服务存储数据对象包括:指示所述数据存储网络服务跨多个存储设备冗余地存储数据对象。

14. 如权利要求13所述的方法,进一步包括存储所述数据存储网络服务的实际使用量的信息特征,其中所述实际使用量是取决于网络服务调用而确定的。

15. 如权利要求13所述的方法,其中所述数据对象的标识符表明键,并且其中指示数据存储网络服务存储对象包括指定对应于所述数据对象的键。

16. 如权利要求13所述的方法,其中所述数据对象的标识符表明对应于数据存储网络服务的用户的命名空间,并且其中指示所述数据存储网络服务存储对象包括指定与所述命名空间相关的数据对象。

17. 如权利要求16所述的方法,进一步包括:

响应于经由所述API接收所述网络服务调用,取决于访问策略确定所述客户是否具有充分的特权来存储数据对象;以及

响应于根据访问策略确定客户不具有充分的特权存储数据对象,拒绝数据对象的存储。

18. 如权利要求17所述的方法,其中确定所述客户是否具有充分的特权包括评估所述访问策略所表明的一个或多个项目,其中所述一个或多个项目包括下列中的一个或多个:表明用户具有对至少所述数据对象的访问的用户标识符,配置成对用户进行身份验证的身份验证凭证,或者指定所述用户对至少所述数据对象所具有的访问类型的信息。

19. 如权利要求13所述的方法,其中接收网络服务调用包括接收编码所述数据对象的标识符的统一资源定位符(URL)。

20. 如权利要求13所述的方法,其中接收网络服务调用包括接收编码存储或检索所述数据对象的操作的统一资源定位符(URL)。

21. 如权利要求13所述的方法,其中接收网络服务调用包括接收根据一个版本的超文

本传输协议HTTP格式化的HTTP请求。

22. 如权利要求21所述的方法,其中HTTP请求包括与数据存储网络服务的用户相关联的身份验证凭证。

23. 如权利要求21所述的方法,其中所述HTTP请求编码创建新的存储桶的请求,该方法进一步包括响应于接收HTTP请求在对象命名空间内创建新的存储桶。

24. 如权利要求21所述的方法,其中所述HTTP请求编码删除存储桶的请求,该方法进一步包括响应于接收HTTP请求从对象命名空间删除存储桶。

25. 如权利要求21所述的方法,其中所述HTTP请求编码列出与给定存储桶相关联的对象的请求,该方法进一步包括响应于接收HTTP请求标识与给定存储桶相关联的数据存储网络服务中存储的对象。

26. 如权利要求21所述的方法,其中所述HTTP请求编码读取或写入与给定存储桶相关联的元数据的请求,并且该方法进一步包括响应于接收HTTP请求检索或存储与给定存储桶相关联的元数据,其中所述元数据包括表明下列中的一个或多个的信息:与所述给定存储桶相关联的使用历史,与所述给定存储桶相关联的计费历史,或者与所述给定存储桶相关联的状态信息。

27. 如权利要求21所述的方法,其中所述HTTP请求编码读取或写入与给定存储桶相关联的访问策略的请求,并且该方法进一步包括响应于接收HTTP请求检索或存储与给定存储桶相关联的访问策略。

28. 如权利要求21所述的方法,其中所述HTTP请求编码检索数据对象的HTTP GET请求,并且该方法进一步包括响应于接收HTTP请求根据HTTP GET请求检索数据对象。

29. 如权利要求21所述的方法,其中所述HTTP请求编码存储数据对象的HTTP PUT请求,并且该方法进一步包括响应于接收HTTP请求根据HTTP PUT请求存储数据对象。

具有网络服务客户接口的分布式存储系统

[0001] 本分案申请是基于申请号为200680053577.2,申请日为2006年11月30日,发明名称为“具有网络服务客户接口的分布式存储系统”的中国专利申请的分案申请。

技术领域

[0002] 本发明涉及数据存储系统,尤其是涉及配置成提供对存储器的访问作为网络服务的存储系统。

背景技术

[0003] 很多不同的计算应用依赖于用于持久存储各种应用程序数据的一些类型的存储介质。例如,普通办公应用程序和多媒体应用程序产生并使用各种类型和格式的应用数据,例如其中包括文档、电子表格、静止的图像、音频和视频数据。经常地,这样的数据被存储,用于以用户名义的重复的访问或使用。例如,用户可能希望在一段时间内存储和处理很多文档或其它数据,并可能期望在需要时数据以可预测的状态可容易得到。

[0004] 在传统计算系统中,由应用程序用于持久的应用程序数据存储的存储介质通常大部分为磁性固定的驱动器或“硬驱动器”,虽然也可使用光和固态存储设备。这样的设备集成在执行应用程序的计算机系统中或可通过本地外围接口或网络访问该系统。一般来说,用作应用程序存储的设备由管理设备级行为的操作系统管理,以向需要存储访问的各种应用程序提供相容的存储接口,例如文件系统接口。

[0005] 应用程序存储的该常规模型呈现一些限制。首先,它通常限制应用程序数据的可访问性。例如,如果应用程序数据存储在某些计算机系统的本地硬驱动器上,它可能对在其它系统上执行的应用程序是不可访问的。即使数据存储在网络可访问的设备上,在当前网络外部的系统上执行的应用程序可能不能访问该设备。例如,由于安全原因,企业通常限制对其局域网(LAN)的访问,以便企业外部的系统不能访问企业内部的系统或资源。因此,在便携式设备(例如,笔记本或手持式计算机、个人数字助理、移动电话设备等)上执行的应用程序可能经历访问持久地与固定的系统或网络关联的数据的困难。

[0006] 传统的应用程序存储模型还可能不能充分确保所存储的数据的可靠性。例如,传统操作系统一般默认将应用程序数据的一个备份存储在一个存储设备上,如果希望数据冗余,则要求用户或应用程序产生并管理其自己的应用程序数据的备份。虽然个别存储设备或第三方软件可提供一些程度的冗余,但这些功能部件对应用程序可能不是始终如一地可得到的,因为应用程序可得到的存储资源可能在应用程序安装中极大地变化。操作系统作为媒介的传统存储模型还可能限制数据的交叉平台可访问性。例如,不同的操作系统可以不同的、不兼容的格式存储用于相同的应用程序的数据,这可能使在一个平台(例如,操作系统和基本的计算机系统硬件)上执行的应用程序的用户难以访问通过在不同平台上执行的应用程序存储的数据。

发明内容

[0007] 公开了分布式基于网络服务的存储系统的不同实施方式。根据一个实施方式,系统可包括配置成根据网络服务协议来接收对数据对象的访问的客户请求的网络服务接口。对给定数据对象的访问的给定客户请求可包括相应于给定数据对象的键值。系统还可包括配置成存储数据对象的复本的很多存储节点,其中每个复本可通过相应的定位器值(locator value)访问,且其中每个定位器值在系统内是唯一的。系统可进一步包括配置成存储每个数据对象的相应的键映射项目的键映射实例,其中对于给定的数据对象,相应的键映射项目包括键值和相应于给定数据对象的每个存储的复本的每个定位器值。系统还可包括配置成从网络服务接口接收对数据对象的访问的客户请求的协调器。响应于给定的客户请求,协调器可配置成访问键映射实例,以识别相应于键值的一个或更多定位器值,并且对于特定的定位器值,访问相应的存储节点以取回相应的复本。

[0008] 在系统的特定实现中,网络服务接口可进一步配置成根据网络服务协议来接收存储数据对象的客户请求,其中存储特定数据对象的特定客户请求包括相应于特定数据对象的键值。协调器可进一步配置成从网络服务接口接收存储数据对象的客户请求,并响应于特定的客户请求,协调器可配置成将特定数据对象的一个或更多复本存储到一个或更多相应的存储节点。响应于存储特定数据对象的给定复本,给定的存储节点可配置成将相应于给定复本的定位器值返回到协调器。

附图说明

[0009] 图1是示出用于向用户提供存储作为网络服务的存储模型的一个实施方式的结构图。

[0010] 图2是示出存储服务系统体系结构的一个实施方式的结构图。

[0011] 图3是示出存储服务系统组件的物理部署的一个实施方式的结构图。

[0012] 图4是示出存储节点的一个实施方式的结构图。

[0013] 图5是示出配置成组织存储节点内的数据对象的数据结构的一个实施方式的结构图。

[0014] 图6是示出执行对象获取操作的方法的一个实施方式的流程图。

[0015] 图7是示出执行对象放置操作的方法的一个实施方式的流程图。

[0016] 图8是示出执行对象释放操作的方法的一个实施方式的流程图。

[0017] 图9是示出重装对象存储空间的方法的一个实施方式的流程图。

[0018] 图10是示出一组键映射实例数据结构的一个实施方式的结构图。

[0019] 图11A-D示出键映射实例的分级实现的一个实施方式。

[0020] 图12是概述键映射实例内分级层次间的关系的结构图。

[0021] 图13是示出执行键映射项目放置操作的方法的一个实施方式的流程图。

[0022] 图14是示出执行键映射项目获取操作的方法的一个实施方式的流程

[0023] 图15A是示出使用更新传播来使键映射实例同步的方法的一个实施方式的流程图。

[0024] 图15B是示出使用反熵协议(anti-entropy protocol)来使键映射实例同步的方法的一个实施方式的流程图。

[0025] 图16是示出复制器键映射项目的一个实施方式的结构图。

- [0026] 图17示出不平衡索引数据结构的一个实施方式。
- [0027] 图18示出用在不平衡索引数据结构中的索引节点的一个实施方式。
- [0028] 图19示出分层索引数据结构的一个实施方式。
- [0029] 图20是示出遍历不平衡索引数据结构的方法的一个实施方式的流程图。
- [0030] 图21是示出处理FINGERPRINT反熵协议消息的方法的一个实施方式的流程图。
- [0031] 图22是示出处理FILTER反熵协议消息的方法的一个实施方式的流程图。
- [0032] 图23示出发现和故障检测监控进程 (DFDD) 的一个实施方式。
- [0033] 图24示出可由DFDD实例维持的全局操作状态机的一个实施方式。
- [0034] 图25是示出根据基于闲聊 (gossip-based) 的协议使DFDD实例同步的方法的一个实施方式的流程图。
- [0035] 图26是示出在存储服务系统内的存储类的操作的方法的一个实施方式的流程图。
- [0036] 图27是示出根据存储节点的当前状态信息动态地确定用于存储数据对象的一个或更多复本的写入计划的一个实施方式的流程图。
- [0037] 图28是示出动态地确定关于对象的写入计划的一个实施方式的流程图,该对象的一个或更多复本已被存储在存储节点中间。
- [0038] 图29是示出计算机系统的示例性实施方式的流程图。
- [0039] 虽然本发明易受各种更改和可选形式的影响,但是其具体的实施方式作为例子在附图中示出并将在这里被详细描述。然而应理解,附图和对其的详细描述不是用来将本发明限制到所公开的特定形式,而是相反,本发明包括落在如所附权利要求界定的本发明的实质和范围内的所有更改、等效物和可选方案。

具体实施方式

[0040] 当计算应用程序变成更加数据密集的以及地理上分散的时,对应用程序数据的可靠、与位置无关的访问的需要便增加了。例如,多媒体应用程序如授权、存储和重放应用程序要求当多媒体内容的质量和数量提高时逐步增加数据存储的量。进一步地,可能期望从不同位置访问应用程序数据,而不考虑存储数据的设备的位置如何。例如,虽然很多计算机包括相当数量的基于磁盘的存储器,但是以统一和方便的方式远程访问这样的存储器呈现出技术上和安全性上的困难。

[0041] 与配置各个计算机以完全依赖于其自己的内部存储资源或提供基于局域网的存储资源(例如,网络附属存储 (NAS)、存储区域网络 (SAN) 等) 相反,连接互联网的数据存储服务可配置成通过基于互联网的协议例如网络服务 (WS) 协议向客户提供一般存储服务。基于互联网的协议例如网络服务协议一般是独立于平台的,因为它们一般独立于基本的软件或硬件起作用。因此,提供数据存储能力作为网络服务可提供不依赖于在应用程序主机系统内或局域网上实现的存储资源而对任意数量的存储的很多不同类型的应用程序直接访问。此外,通常可从提供互联网接入的任何位置访问网络服务可访问的存储器。网络服务可访问的存储器可促进很多不同计算功能部件的实现,例如通过不同设备或应用程序对公用数据的远程访问、在执行期间通过各个应用程序对广泛分布的数据的远程访问、在协作工作的分布的用户中间对数据的访问和/或共享、在分布的用户中间对应用程序结果数据的分发,以及很多其它类似的功能部件。

[0042] 在下面的讨论中,描述了可用在基于网络服务的存储系统中的可能的数据存储模型的一个实施方式。此后,公开了根据该数据存储模型可配置成提供存储服务的存储服务系统,并详细描述了其各个组件。

[0043] 存储服务用户接口和存储模型的概述

[0044] 用于向用户提供数据存储作为服务例如网络服务的存储模型的一个实施方式在图1中示出。在所示模型中,存储服务接口10设置为到存储服务的面向顾客或用户的接口。根据通过接口10呈现给用户的模型,存储服务可组织为可通过接口10访问的任意数量的存储桶20a-n。每个存储桶20可配置成存储任意数量的对象30a-n,对象30又可存储由存储服务的用户指定的数据。

[0045] 如下面更详细描述,在一些实施方式中,存储服务接口10可配置成根据网络服务模式支持存储服务及其用户之间的交互作用。例如,在一个实施方式中,接口10可作为具有统一资源定位器(URL),例如 `http://storageservice.domain.com` 的网络服务端点而被客户访问,由服务客户产生的网络服务请求可被指引到该URL而进行处理。一般而言,网络服务可指发出请求的客户通过请求接口可得到的任何类型的计算服务,该请求接口包括一个或更多基于互联网的应用层数据传输协议,例如超文本传输协议(HTTP)的一种版本或另外的适当协议。

[0046] 网络服务可使用各种授权服务协议以各种体系结构形式实现。例如,在具象状态传输(REST)形式的网络服务体系结构中,与网络服务请求有关的参数(例如,指定所请求的服务的类型、用户证书、将对其进行操作的用户数据等)可被指定为将网络服务请求调用到网络服务端点的数据传输命令的参数,该数据传输命令例如HTTP GET或PUT命令。在一些实现中,REST形式的网络服务体系结构是无状态的,因为每个网络服务请求可包括处理该请求的所有信息而不参考外部状态信息。与REST形式的网络服务体系结构相反,基于文档或基于消息的网络服务体系结构可将与网络服务请求有关的参数和数据编码为可被传输到网络服务端点并接着被该端点解码和作用的文档。例如,可扩展标记语言(XML)的一种版本或另外的适当标记语言可用于格式化网络服务请求文档。在一些实施方式中,用于格式化请求文档的标记语言可为控制请求的处理的参数定界,而在其它实施方式中,标记语言本身的某些功能部件(例如,某些标志)可直接控制请求处理的方面。此外,在一些实施方式中,最终的文档可封装在另一协议例如简单对象访问协议(SOAP)的一种版本内,例如以便促进端点对网络服务请求的处理。

[0047] 其它协议也可用在网络服务体系结构的不同实施方式中。例如,网络服务描述语言(WSDL)的一种版本可被网络服务端点用于向可能的客户公布其接口技术要求。网络服务端点可通过目录协议例如统一描述、发现和集成(UDDI)协议的一种版本来使自己被可能的客户识别。与通过网络服务接口提供计算服务有关的很多其它类型的协议可能存在,且任何给定的网络服务实现可使用这样的协议的任何适当的组合。

[0048] 可以设想,在一些实施方式中,接口10可支持不同于网络服务接口的、代替网络服务接口的或除了网络服务接口外的接口。例如,企业可实现企业外部的客户以及企业内部的用户使用的存储服务,其中企业外部的客户可通过网络服务协议访问服务,企业内部的用户可使用不同类型的接口(例如,对企业的内联网定制的专用接口)。在一些实施方式中,接口10可支持各种类型的接口连接协议中的每一个,存储服务的任何用户可通过这些协

议访问服务。在其它实施方式中,可为每个不同的接口方案提供接口10的不同实例。注意,在一些实施方式中,与处理与客户的交互(例如,接收和响应服务请求)有关的接口10的那些方面可与实现存储服务一般体系结构的那些方面(例如,进入存储桶和对象的层中的服务的组织)分开地实现。在一些这样的实施方式中,与客户交互作用(例如,通过网络服务协议)有关的接口10的部分可能被某些用户例如企业内部的那些用户绕过,如下面结合图2的描述较详细描述。

[0049] 如图1所示,接口10给存储服务用户提供对存储桶20的访问。一般而言,存储桶20可用作与存储服务的用户关联的对象命名空间的根。例如,存储桶20可能类似于文件系统目录或文件夹。在一些实施方式中,各个存储桶20还可形成用于解释存储服务的使用的的基础。例如,用户可与一个或更多用于记账目的的存储桶20关联,且该用户可使用存储资源(例如对象30的存储)而被开帐单,其中的存储资源分级地驻留在由那些存储桶20建立的命名空间内。

[0050] 在所实施方式中,每个存储桶20a-n包括相关的元数据21a-n以及相应的访问策略23a-n。一般而言,元数据21可包括可用于描述给定存储桶20的方面或特性的任何适当的元数据。例如,元数据21可包括识别存储桶创建的日期、其创建者的身份的信息或其它适当的信息,而不管存储桶是否具有任何与其相关联的对象30。在一些实施方式中,元数据21可包括指示存储桶20的使用特征的信息,例如与存储桶20关联的对象30的总大小、关于存储桶20和/或其相关联的对象30的用户的访问历史纪录、与存储桶20相关联的记账历史纪录、或与存储桶20的当前或过去使用有关的任何其它适当的信息。在一个实施方式中,每个桶20可与相应的唯一标识符关联,该标识符可由用户指定或被存储服务自动分配。唯一的标识符可存储在元数据21中或作为存储桶20的单独的特性或字段。注意,在一些实施方式中,给定存储桶20可不包括显式引用、指针或相应于与给定存储桶20关联的对象30的其它信息。更确切地,如下面更详细描述,对象30的定位和选择可通过使用在这里称为键映射的单独的映射程序(mapping facility)来执行。

[0051] 访问策略23可包括控制对与存储桶20关联的对象30的访问所需的任何信息。访问策略23可包括识别被允许访问存储桶20及其相关联的对象30的客户以及在什么容量内的信息。例如,访问策略23可存储一个或更多客户的用户标识符和/或认证证书(例如,口令),并可进一步指定给定客户是否被允许修改或仅仅读取对象30。访问策略23还可实现默认或面向组的策略(例如,通过对指定的客户或客户组允许对对象30的一般读取访问但限制写入访问)或任何其它期望的安全模型。

[0052] 在所实施方式中,给定存储桶20可与一个或更多对象30关联,其中每个对象都包括各自的元数据31和数据33。一般而言,对象30的数据33可相应于任何比特序列。由存储在对象30内的比特表示的数据的类型对存储服务来说可以是透明的。也就是说,比特可表示文本数据、可执行的程序代码、音频、视频或图像数据或任何其它类型的数字数据,而存储服务可不必在存储和操作对象30时在这些不同的数据类型中间进行区分。在一些实施方式中,可将数据33的大小限制为固定的最高限度(例如,1 千兆字节(GB)),而在其它实施方式中,可允许对象30仅仅以存储服务可用的物理存储资源为条件来在大小上按比例变化。

[0053] 类似于与存储桶21相关的元数据21,元数据31可配置成存储关于其相应的对象30

的任何期望的描述性信息。例如,元数据31可包括关于相应的对象30被创建的日期和/或时间、对象30的大小、被对象30存储的数据33的类型(例如,由多用途互联网邮件扩充(MIME)标准定义的数据类型)的信息或任何其它类型的描述性信息。在一些实施方式中,元数据31可储存指示与相应的对象30的用户交互作用的使用或历史纪录信息,以及访问策略信息(例如,指示不同用户可具有的对对象30的访问的类型的许可信息)、对象费用信息(例如,与对象30关联的开单率或历史纪录),或属于对象30的任何其它适当的信息或信息类型的组合。在一些实例中,客户可提供元数据连同将被存储为元数据31的对象数据,而在其它情况下,元数据31可包括由管理存储服务功能部件的系统(例如,在图2中示出和在下面描述的存储服务系统)产生的元数据。具有对对象30的访问权限的客户可访问元数据31中的一些、全部或根本不能访问元数据31,这取决于元数据的类型、客户访问权限的特定规定或其它适当的因素。

[0054] 在一个实施方式中,可使用两个不同的信息项—键或定位器中的任一项来在存储服务系统内识别各个对象30。一般而言,虽然键和定位器可以不同的方式被解释,但是键和定位器每个可包括在存储服务系统的命名空间的环境内作为整体被解释的字母数字字符串或其它类型的符号。在一个实施方式中,键可在相应的对象30在特定的存储桶20内被创建时由客户指定(例如,响应于客户存储新对象的请求)。如果没有键被用户指定,则键可由存储服务系统分配给新对象30。在这样的实施方式中,可能要求与特定存储桶20的对象30关联的每个相应的键在该存储桶20的命名空间内是唯一的。一般而言,只要相应的对象存在于存储服务系统内,键就可作为有效的标识符继续存在,客户可通过该标识符可访问相应的对象30。

[0055] 在给定的存储桶20内,键可用于产生与传统操作系统的文件系统共用的文件目录或文件夹命名空间类似的分层对象命名空间。例如,可准予客户对具有唯一标识符050739517的特定存储桶20的对象读取和写入访问权限。在一个实施方式中,客户可接着发出对地址<http://storageservice.domain.com/050739717>的网络服务请求,以便在存储桶命名空间内产生相应于存储桶内对象的键。例如,客户可指定使用键“My Document/Email/message.txt”在此特定的存储桶内创建对象30,以便可使用对地址<http://storageservice.domain.com/050739717/My Document/Email/message.txt>的网络服务请求来访问对象30。

[0056] 注意,在一些实施方式中,键所隐含的分层结构可不必在对象存储的下面的等级中反映。例如,在一个实施方式中,与给定存储桶20关联的对象30可以展开的、非分等级的方式存储在存储服务系统中,即使与对象30关联的键可隐含等级也是如此。也就是说,在这样的实施方式中,存储桶20可不分等级地包括其它存储桶20。然而,在其它实施方式中,可支持存储桶20在其它存储桶20内的分等级包含,虽然存储桶的任何这样的层次不需要直接映射到被对象键隐含的层次上。

[0057] 在一个实施方式中,在取回或修改被请求的对象30的基本数据33之前,被键识别的、客户访问对象30的请求可经历客户身份验证程序、访问控制检查,和/或映射过程(例如下面更详细描述)。例如,可请求客户提供口令或其它证书以证明客户的身份,且一旦被识别,就可评估与被请求的存储桶20相关的访问控制参数以确定识别的客户是否被充分给予特权来保证对被请求的键的访问。相反,存储服务系统可支持通过定位器而不是键来访

问对象30的可选方法。一般而言,定位器可表示在存储服务系统已知的所有对象30中间对象30的全局唯一的标识符。也就是说,当键对与特定存储桶20关联的命名空间可能是唯一的时,定位器在所有存储桶20中的所有对象30的全局命名空间内可能是唯一的。例如,定位器可包括由存储服务系统产生的在其它定位器中唯一的字母数字字符串。如下面更详细描述,在一些实施方式中,对象30的多个实例可在用于实现存储服务系统的整个物理存储设备中被复制,例如以增加数据冗余和容错能力。在这样的实施方式中,对给定的对象30的每个复制的实例可存在唯一的定位器。

[0058] 注意,虽然在一些实施方式中,只要对象30存在于存储服务系统中,就可保证键保持有效以访问该对象30,但是这样的保证可能适用或可能不适用于该对象30的任何给定的定位器。例如,如果对象30的复制的实例(或副本)移到不同的物理存储位置(例如,由于其基本存储介质的故障或更换),则涉及该特定实例的定位器可能不再是有效的,虽然在其新位置可产生并使用相应于对象30的移动的实例的另一定位器。在下面关于键映射系统组件的操作的讨论中给出了对键和定位器之间关系的更多细节。

[0059] 作为基于键的对象访问与基于定位器的对象访问相比的例子,被上面给定的键 `http://storageservice.domain.com/050739717/My Document/Email/message.txt` 定位的对象30可具有存储在存储服务系统内的一个或更多实例,其中之一可由格式:

[0060] `http://storageservice.domain.com/3859C89A208FDB5A` 的定位器识别。在此特定的实施方式中,注意,相对于特定的存储桶20 来表示对象30的键参考,而定位器参考被表示为全局定位器空间内的绝对128位的十六进制数字(虽然可使用其它类型的定位器编码或格式)。在一个实施方式中,客户发出的指向定位器的网络服务请求可绕过身份验证、访问权限、翻译或可适用于基于键的网络服务请求的其它步骤中的一些或全部。由于较少层的处理,在一些这样的实施方式中,基于定位器的请求可比基于键的请求被更快地处理。然而,因为对基于定位器的请求可绕过安全措施,因此客户可能需要提供敏感对象30的定位器不受到危害的其自己的保证(例如,使用传送和接收定位器所采用的加密或其它安全方法)。进一步地,因为可能不能保证定位器的持久性(例如,在上面讨论的对象实例移动的情况下),选择执行基于定位器的对象访问的客户可能需要容许定位器在使用期间变得无效的可能性,例如,通过在优先的基础上获得新定位器或相应于发现现有的定位器不再有效。

[0061] 根据客户的存储需要和上面提到的资格证明(caveat),基于定位器的访问相对于基于键的访问可提供改进的处理性能(例如,网络服务请求处理的等待时间和总处理能力中的处理性能)。例如,客户可选择使用基于定位器的访问来参考不是特别敏感的频繁访问的对象30(例如,参考材料、图像或其它适当类型的数据)。注意,在一些实施方式中,基于定位器的访问可在各个对象30的基础上被禁止,因而迫使希望访问这样的对象的客户使用基于键的请求并相应地提交到与这样的请求关联的任何验证和访问权限控制。然而,即使对于基于定位器的访问被启用的对象30来说,不拥有有效定位器的恶意或误操作的客户可能只有成功访问任何给定对象30的随机的机会。通过使用较大的定位器命名空间、用于产生定位器的安全技术(例如,对象数据的安全散列的使用)或其它适当的技术,就可能使得这样的机会在任意情况下都不可能。

[0062] 存储系统体系结构和实现

[0063] 在图2中示出了可配置成实现如图1所示的基于网络服务的存储服务的存储服务体系结构的一个实施方式。在所示实施方式中,很多存储客户50a-n可配置成通过网络60与网络服务平台100进行交互作用。网络服务平台100可配置成与存储服务协调器120(或简单地,协调器120)的一个或更多实例通过接口连接,协调器120又可与一个或更多键映射实例140和位存储(bitstore)节点160通过接口连接。此外,复制器180还可配置成与位存储节点160以及复制器键映射实例190通过接口连接。协调器120和复制器180都可与节点拣取器(nodepicker)服务130通过接口连接。在所示实施方式中,节点拣取器130、键映射140、位存储节点160和复制器键映射190的每个实例都可与发现和故障检测监控进程(DFDD)110的相应实例关联。注意,在给定组件的一个或更多实例可能存在的场合,在下文中可以单数或复数形式对该组件进行引用。然而,任一种形式的使用不是用来排除另一种。

[0064] 在不同实施方式中,图2中示出的组件可直接在计算机硬件中实现为可直接或间接地被计算机硬件(例如,微处理器或计算机系统)执行的指令或这些技术的组合。例如,图2的组件可通过包括很多计算节点(或简单地,节点)的分布式系统实现,例如在图29中示出和下面描述的计算机系统实施方式。在不同实施方式中,给定存储服务系统组件的功能可通过特定的节点或分布式跨越的一些节点实现。在一些实施方式中,给定节点可实现多于一个的存储服务系统组件的功能。在图2的组件的一般功能和如图3所示的存储服务系统的示例性物理配置的概述之后,下面结合图4-28的描述提供了特定存储系统组件的某些实施方式的细节。

[0065] 一般而言,存储客户50可包括可配置成通过网络60向网络服务平台100提交网络服务请求的任何类型的客户。例如,给定存储客户50可包括网络浏览器的适当版本,或配置成作为对网络浏览器提供的执行环境的扩展来执行或在网络浏览器提供的执行环境内执行的插入模块或其它类型的代码模块。可选地,存储客户50可包括应用程序,例如数据库应用程序、媒体应用程序、办公应用程序或可利用持久的存储资源的任何其它应用程序。在一些实施方式中,这样的应用程序可包括用于产生和处理网络服务请求的足够的协议支持(例如,对于超文本传输协议(HTTP)的适当版本),而不必实现对所有类型的基于网络的数据的完全的浏览器支持。也就是说,存储客户50可用是配置成直接与网络服务平台交互的应用程序。如下所述,存储客户50可配置成根据具象状态传输(REST)形式的网络服务体系结构、基于文档或消息的网络服务体系结构或另一种适当的网络服务体系结构来产生网络服务请求。

[0066] 在其它实施方式中,存储客户50可配置成以对其它应用程序透明的方式来向那些应用程序提供对基于网络服务的存储的访问。例如,存储客户50可配置成根据上面描述的存储模型的适当变形来与操作系统或文件系统结合以提供存储。然而,操作系统或文件系统可向应用程序提供不同的存储接口,例如文件、目录和/或文件夹的传统文件系统层次。在这样的实施方式中,应用程序可能不需要修改成利用图1的存储系统服务模型。替代地,网络服务平台100的接口连接的细节可通过存储客户50和代表在操作系统环境内执行的应用程序的操作系统或文件系统来协调。

[0067] 存储客户50可通过网络60将网络服务请求传送到网络服务平台100并从网络服务平台100接收应答。在不同实施方式中,网络60可包含建立客户50和平台100之间的基于网络的通信所必需的联网硬件和协议的任何适当组合。例如,网络60可通常包含共同实现

互联网的各种电信网络和服务供应商。网络60还可包括专用网络,例如局域网(LAN)或广域网(WAN)以及公共或专用无线网络。例如,给定客户50和网络服务平台100可分别在具有其自己的内部网络的企业内被提供。在这样的实施方式中,网络60可包括建立给定客户50和互联网之间以及互联网和网络服务平台100之间的网络链接所必需的硬件(例如,调制解调器、路由器、交换机、负载均衡器、代理服务器等)和软件(例如,协议堆栈、账目管理软件、防火墙/安全软件等)。注意,在一些实施方式中,存储客户50可使用专用网络而不是公共互联网来与网络服务平台100通信。例如,客户50可在作为存储服务系统的同一企业内被提供服务。在这样的情况下,客户50可完全通过专用网络60(例如,可使用基于互联网的通信协议但不可公开访问的LAN或WAN)来与平台100通信。

[0068] 一般而言,网络服务平台100可配置成实现一个或更多服务端点,这些服务端点配置成接收并处理网络服务请求,例如访问由存储服务系统存储的对象30的请求。例如,网络服务平台100可包括配置成实现在前面例子中使用的端点`http://storageservice.domain.com`的硬件和/或软件,以便指向该端点的基于HTTP的网络服务请求被正确接收和处理。在一个实施方式中,网络服务平台100可实现为服务器系统,该服务器系统配置成从客户50接收网络服务请求并将它们转发到协调器120或存储服务系统的其它组件用于进行处理。在其它实施方式中,网络服务平台100可配置为实现负载均衡和其它请求管理功能部件的很多不同的系统(例如,在群集拓扑中),这些请求管理功能部件配置成动态地管理处理负载的大规模网络服务请求。

[0069] 在不同实施方式中,网络服务平台100可配置成支持REST形式或基于文档(例如,基于SOAP)类型的网络服务请求,如上面详细描述。在一个特定的实施方式中,平台100可配置成实现特定的网络服务应用程序编程接口(API),其支持对由存储服务系统管理的实体的各种操作。例如,由平台100实现的API可支持对存储桶或对象的基本客户操作,包括列表(可选地根据过滤模式或标准过滤的)存储桶20或对象30、取回存储桶20或对象30的数据或元数据、以及创建或删除存储桶20或对象30。在一些实施方式中,API可支持更复杂的客户操作,例如对多个存储桶20或对象30的操作的批应用。

[0070] 除了用作客户网络服务请求的可寻址的端点以外,在一些实施方式中,网络服务平台100可实现各种客户管理功能部件。例如,平台100可例如通过跟踪请求的客户50的身份、客户请求的数量和/或频率、代表客户50存储或取回的对象30的大小、被客户50使用的总存储带宽、被客户50请求的存储类,或任何其它可测量的客户使用参数,来协调网络服务包括存储资源的客户使用的记录和账目管理。平台100还可实现财务账目管理和记账系统,或可维护可被用于客户使用行为的报告和记账的外部系统询问和处理的使用数据的数据库。

[0071] 在某些实施方式中,平台100可配置成收集和/或监控各种存储服务系统操作度量,例如反映从客户50接收的请求的速率和类型的度量、这样的请求所利用的带宽、对这样的请求的系统处理等待时间、系统组件利用(例如,在存储服务系统内的网络带宽和/或存储利用)、由请求产生的错误率和错误类型、被请求的对象30的特点(例如,大小、数据类型等)或任何其它适当的度量。在这样的实施方式中,平台100可配置成集合地收集这样的度量,例如作为随着时间的平均值,或作为可进行各种分析的特定数据点。在不同实施方式中,这样的度量可用来以对客户50可见或不可见的方式检验或监控系统性能。例如,在一个

实施方式中,这样的度量可被系统管理员用于调整和维护系统组件,而在其它实施方式中,这样的度量(或这样的度量的相关部分)可暴露给客户50,以使这样的客户能够监控其对存储服务系统的使用。

[0072] 在一些实施方式中,平台100还可实现用户身份验证和访问控制程序。例如,对于对访问与给定存储桶20关联的特定对象30的给定网络服务请求,平台100可配置成确定与该请求关联的客户50是否被授权访问给定存储桶20和特定的对象30。平台100可通过例如对照与给定存储桶20关联的证书评估身份、口令或其它证书,并对照指定对特定对象30的可允许操作的访问控制列表评估对特定对象30的请求的访问,来确定这样的授权。如果客户50没有足够的证书来访问存储桶50或对对象30执行请求操作(例如,客户50试图写入对象30,而只具有读取访问特权),则平台100可例如通过向请求的客户50返回指示错误情况的应答来拒绝相应的网络服务请求。可以设想,在一些实施方式中,每个存储桶20和对象30可具有管理对该存储桶或对象的访问的相关联的访问控制策略。这样的访问控制策略可作为元数据21或31内的访问控制信息的记录或列表被存储,或作为与元数据21和31不同的数据结构被存储。

[0073] 然而在一些实施方式中,存储服务系统例如图2的系统可支持任意大小的对象30,在其它实施方式中,对象30可限制到某种最大大小,该最大大小也称为块大小(chunk size)的确定的最大大小。在一些这样的实施方式中,当客户提供将与键关联存储的数据且该数据超过块大小时,平台100可配置成根据块大小将数据分成两个或更多块。在一个实施方式中,平台100可配置成将每个块生成为具有相关联的键值的相应的对象30。平台100可为每个块生成键值作为客户提供的键的函数,这样当执行涉及客户提供的键的访问请求时,原始的客户数据可由块重建。例如,平台100可配置成由客户数据产生N个块,并可通过将N个不同的模式(pattern)附加到客户提供的键来为这些块产生N个相应的键,其中N个不同的模式以N个块被产生的相同顺序被按字母顺序式地排序。接着N个块中的每个都可使用下面描述的技术作为不同的对象30来管理,且原始数据可通过列出具有键值的所有对象30并以列出的顺序取回那些对象来重新产生,其中对于该键值客户提供的键是前缀。在一些实施方式中,可访问、修改或移除单独的块而不干扰其它块,相对于将数据管理为单个大的对象30来说这可提高系统性能。可以设想,在一些实施方式中,可允许客户50指定器所提供的数据对象是否应被分割成块。

[0074] 如同在图2中示出的很多存储服务系统组件的情况一样,将网络服务平台100的功能从其它组件分离可改善存储服务系统的维护和总的可缩放性。例如,可特别提供附加的硬件和软件资源来用于管理独立于分配给其它任务的资源的附加的网络服务处理负载。进一步地,与平台100关联的任何资源故障的影响可被限制到该特定的功能区域,因而便于故障的隔离和解决。然而,在一些实施方式中,可以设想,平台100的功能可合并到其它组件中。例如,协调器120可配置成包括与平台100关联的任务。

[0075] 还注意到,虽然网络服务平台100可表示主要接口,客户50可通过该接口访问存储服务系统的功能部件,它不需要表示到这样的功能部件的唯一接口。例如,在一些实施方式中,协调器120可配置成支持与网络服务接口不同的供替换的API。这样的供替换的API可例如用于允许提供存储服务系统的企业内部的客户绕过网络服务平台100。在一些情况下,平台100的账目管理和/或证书管理服务对内部客户例如管理客户可能是不必要的。

[0076] 协调器120可配置成协调网络服务平台100和存储服务系统的其它组件之间的行为。在一些实施方式中,协调器120的主要职责可包括响应于指向对象30的网络服务请求,来指导对那些对象30的对象数据33和元数据31的读取和写入行为。例如,如下面更详细描述,对象读取访问可包括执行对键映射实例140的访问,以取回指示存储有给定对象30的复本的位存储节点160的定位器,后面是执行对特定定位存储节点160的访问以便读取被请求的数据。类似地,对象创建或修改可包括将对象30的多个复本存储到不同的位存储节点160,且如果必要则更新键映射实例140,以反映所创建或修改的复本的定位器。在一些实施方式中,协调器120可配置成对键实例140和位存储节点160执行这些读取和写入操作。然而,注意,在某些实施方式中,协调器120可不操作来在对象30创建和修改时创建对象30的全部数量的期望复本。如下面更详细描述,在一些实施方式中,当协调器120完成了写入对象30的确定数量的复本(例如,两个复本)时,该对象30的写入操作可被考虑为结束。该对象30的另外的复制物可作为通过复制器180的带外或异步操作来完成。也就是说,在这样的实施方式中,对象创建或修改操作的带内或同步部分可包括产生少于总期望数量的被影响的对象30的复本。注意,虽然协调器120被示为与键映射实例140、位存储节点160和其它系统组件不同的组件,但是在一些实施方式中有可能协调器120的实例与另一系统组件(例如,作为由单个计算机系统执行的软件组件)一起实现。因此,虽然这里的描述可指将数据存储到或取自于位存储节点160、键映射实例140或另外的组件的协调器120,但是应理解在一些实施方式中,这样的处理可出现在共享的计算系统资源中。

[0077] 如上面关于图1描述的,在一些实施方式中,存储服务系统可包括基于存储桶的存储模型,其中不同对象30的键可为了管理(例如,账目管理、记账)、安全或其它目的而被分组到存储桶20中。在一个实施方式中,协调器120可配置成响应于来自客户50的相应的网络服务请求而处理各种与存储桶相关的操作。例如,协调器120可配置成执行下列存储桶操作中的一些或全部:

[0078] -创建存储桶:为存储桶20产生并储存新的存储桶名称。

[0079] -删除非空存储桶:删除包括相关的元数据21和与给定存储桶20内对象30关联的所有键的给定存储桶20。

[0080] -删除空的存储桶:只有当没有对象30的键与给定存储桶20关联时才删除给定存储桶20和相关的元数据21,否则返回错误情况。

[0081] -写入存储桶数据:将数据(例如,元数据21)写到现有的存储桶20中。

[0082] -列出存储桶键:列出与给定存储桶20关联的对象30的键(可选地根据模式、规则表达式、通配符等存储或过滤的)。

[0083] -列出存储桶:列出与给定签约者(例如,用户或客户50)关联的存储桶20。

[0084] 在一些实施方式中,协调器120可配置成使用适当的随机数算法以及低概率的冲突产生来为新近创建的存储桶20生成标识符。在其它实施方式中,协调器120可配置成例如通过在客户请求存储桶创建时检验被请求的标识符相对于现有的存储桶标识符的唯一性,来支持客户专用的存储桶标识符。

[0085] 如上所述,对象30的实例可跨越不同的位存储节点160而被复制,例如以增加对象数据经得住任何给定节点160或其相关的基础结构的故障的可能性。存储服务系统内的对象复制为在所示实施方式中可被节点拣取器130和复制器180处理的管理和最佳化呈现一

些机会,如下。

[0086] 当协调器120接收写入对象30的请求时,它可在声明写入将要结束之前相应地将对象30写到给定数量的节点160。然而,对象30应被写入的节点160的数量和特定选择可能根据很多不同存储策略考虑因素而变化。例如,在写入操作被认为将要结束之前,要求对象30的确定的最小数量的复本(例如,两个或三个)被成功地写入可能是谨慎的,以便考虑到可能的故障写入的数据是持久存在的。然而,确保选择成储存最小数量的复本的节点60分布在故障的不同可能的位置或区域中也可能是期望有的。例如,位于相同的数据中心的节点160比在地理上分离的节点160可能更有可能同时失效(例如,由于灾难性故障如自然灾害、电源故障等)。

[0087] 可一般称为存储节点选择逻辑的节点拣取器130可配置为协调器120 和复制器180可访问的服务,在一个实施方式中,协调器120和复制器180 可实现用于为对象读取和写入操作选择节点160的算法,以便满足不同的存储策略。例如,在如上概述的写入对象30的情况下,节点拣取器130 可操作来开发写入计划,或对象30应被写入的特定序列的节点160。在开发特定的写入计划中,节点拣取器130可配置成确保写入计划有成功的合理机会—例如,在写入计划中指定的节点160实际上是可操作的并预期具有可用来接受对象30的足够的存储资源—并且写入计划如果被完成将满足与写入操作有关的所有存储策略。示范性写入存储策略可包括下列:

[0088] -持久性策略:如果写入计划成功地完成,则对象30的实例将存储在至少N个不同的节点160上。

[0089] -区域多样性策略:如果可能,写入计划将包括分布在至少M个不同区域中的节点160。

[0090] -本地性策略:如果可能,写入计划将优先选择(例如,在数量上) 在被请求的协调器120本地的区域内的节点160。

[0091] -负载平衡策略:努力使写入请求业务量在节点160中间均衡(例如,以避免“热节点”)。

[0092] -空间平衡策略:努力使存储资源容量利用在节点160中间均衡。

[0093] -最低成本链策略:努力最小化在写入计划中节点序列写入操作的总成本(例如,网络等待时间)。

[0094] 注意,在不同实施方式中,节点拣取器130可配置成当制定给定的写入计划时考虑这些策略或没有列出的其它策略中的一些或全部。进一步地,不同的策略可能被加有不同的优先级。例如,在一个实施方式中,持久性策略可以是所有写入计划必须满足的强制性策略,而剩余的策略可以在尽力服务(best-effort)的基础上得到满足。在一些情况下,一些存储策略可能与另一些存储策略冲突。例如,支持对象实例在不同区域中间的广泛分布的区域多样性特点通常与支持在特定的区域内局部化对象实例的本地性策略相反。如果对象实例的数量不够大,则满足两个策略是可能的。例如,如果要创建对象30的五个实例,则可能将两个实例存储在两个不同的区域且三个实例在被请求的协调器120本地的第三个不同的区域内,因而满足本地性和区域相异性策略。如果不可能满足写入计划所指定的所有策略,则节点拣取器130可试图以优先考虑满足并产生尽力服务写入计划的那些策略,或可将指示对象写入不能被良好地执行的错误指示返回到发出请求的协调器120。

[0095] 在一些实施方式中,节点拣取器还可帮助协调器120读取对象30。例如,对象读取操作可被不同于最初或最近写入请求的对象30的协调器的协调器120请求。因此,相对于写入协调器120可被存储在本地的对象30 的实例可能相对于读取协调器120来说不是本地的。节点拣取器130可配置成识别提供读取协调器120可利用的最佳读取性能的最佳节点160。例如,节点拣取器130可识别最接近于读取协调器120的节点160 (例如,根据地理距离或网络拓扑)或提供最大读取带宽的节点160 (例如,最少负载的节点160或具有较高性能的存储硬件类的节点160),或节点拣取器130 可使用用于选择节点160的其它性能标准,其中从节点160读取对象30。在其它实施方式中,不是相对于读取协调器120最佳化读取操作的性能,节点拣取器可全局地计划并行的读取操作,以便在总体上最佳化系统的性能 (例如,以最大化全局读取总处理能力)。

[0096] 为了开发写入计划并关于对象读取操作向协调器120提出建议,节点拣取器130可配置成例如相对于其操作状态和有效资源来监控节点160的状态。在一个实施方式中,节点拣取器130可配置成与DFDD 100的实例进行交互作用(下面描述的),以便识别存储服务系统内并行操作的节点 160。一旦节点拣取器130意识到操作的节点160,它就可询问那些节点来确定在每个节点处可利用的资源 (例如,存储容量)。因为节点160的操作和资源状态可随着时间而变化,所以在一些实施方式中,节点拣取器130 有时可通过DFDD 110更新操作状态信息并轮询最终的节点160,以更新其资源状态信息。注意,在一些实例中,节点拣取器130可能没有节点160 的状态的完全同步的观察。例如,被认为节点拣取器130可用的特定节点 160可实际上自从状态信息的最后更新以后失效。在这样的实例中,节点拣取器130可能不能保证其读取或写入计划能够由协调器120完成。如果协调器120不能访问由节点拣取器130指定的节点160,则相关的操作可能失败且被协调器120全部重新尝试,或协调器120可与节点拣取器130 协商以修改请求的计划。在一些情况下,如果在写入计划中指定的节点160 的故障只影响可选的或尽力服务的存储策略,同时仍然允许满足强制性的存储策略,则可允许写入计划结束。在一些这样的实施方式中,复制器180 可配置成在以后的时间努力满足未满足的存储策略,如下所述。

[0097] 在一些实施方式中,节点拣取器130的多个实例可在整个存储服务系统中部署。例如,节点拣取器130的相应实例可为协调器120的每个实例部署。虽然节点拣取器130可部署为可从协调器120通过API (和复制器 180) 访问的服务,这种结构不是必要的。在其它实施方式中,节点拣取器130的功能可直接合并并在协调器120和/或复制器180的实例中。

[0098] 如上所述,对象数据的可靠性和有效性可通过在整个存储服务系统中复制对象30来增加。例如,在地理上分散的系统内分布对象30的实例或复本可提高类似地分散的客户50的性能,这些客户50试图通过可能定位较接近于这样的客户的一些对象实例来访问这样的对象30。(注意,在对象复制的情景下,术语“实例”和“复本”可在这里互换地使用)进一步地,对象复制可通常降低由特定对象实例的破坏而产生的数据损失的可能性。然而,在一些实施方式中,情况可能是,在给定的时间点,对象30的有效复本的数量可能小于复本的期望或目标数量。例如,在存储服务系统中实施的复制存储策略可指定每个对象30的特定目标数量的复本 (例如,3或任何其它适当的数量)应在任何给定的时间存在。然而,对于给定的对象 30,由于各种原因,有效复本的实际数量可能小于目标数量。例如,先前有效的复本可能由于存储其的设备的故障而变得不可访问。可选地,在一些实施方式中,被协调器120写

入的对象30的实例的数量可小于该对象 30的复本的目标数量。例如,如上所述,实例可根据由节点拣取器130指定的写入计划而被写入,该写入计划可考虑需要比目标数量少的实例的持久性策略。

[0099] 在一个实施方式中,复制器180可操作来检查对象30以确定每个对象30的有效复本的数量是否满足目标数量(例如,复本的数量是否至少是在做出确定时的目标数量)。特别地,在一个实施方式中,复制器180 可配置成对指定每个对象30的实例的数量和位置的记录连续重复。例如,复制器180可涉及复制器键映射190,复制器键映射190可与在下面更详细描述的关键映射实例140一样可配置成存储对象键和识别复制的对象实例的相应定位器之间的映射。(在其它实施方式中,复制器180可考虑一个键映射实例140,而不是专用的键映射实例)。在一些实施方式中,可以设想,复制器180的多个实例可配置成同时检查键映射空间的不同部分,这可减少检查由存储服务系统管理的所有对象30的状态所需要的总时间量。

[0100] 如果复制器180确定对于给定的对象30来说不满足有效复本的目标数量,则它可配置成以类似于协调器120执行对给定对象30的写入操作的方式来写入给定对象30的额外的复本。例如,复制器180可与节点拣取器130进行交互作用以获得用于产生额外的复本的写入计划如上所述。可选地,复制器180可实现其自己的用于产生对象复本的算法反映(algorithms reflecting)策略。在一些实施方式中,复制器180可根据需要额外的复本的条件来符合不同的优先级以产生对象30的复本。例如,具有少于目标数量的在复制器键映射190中列出的定位器的对象30可最近由协调器120写入。相反,具有目标数量的定位器的对象30可展示基本存储的故障,其中一些定位器是无效的。作为策略的问题,复制器180可试图在后面的情况之前纠正前面的情况,反之亦然。可选地,每当遇到这种条件时,复制器180可试图为具有少于目标数量的有效复本的任何对象 30产生额外的复本,而不管引起该条件的特定情况如何。

[0101] 如上所述,对象30的存储的总可靠性可通过将对象数据的复本存储在例如不同的区域或数据中心内来增加。然而,注意,在一些实施方式中,每个复本不需要相应于对象数据的准确拷贝。在一个实施方式中,对象30 可根据冗余编码方案(例如奇偶性、错误校正代码或其它方案)分成很多部分或“碎片”,以便对象数据可从少于全部的生成的部分重建。例如,使用从对象30产生N个部分的各种方案对象数据可根据编码方案从任何N-1 个部分、N个部分的任何简单的多数或部分的其它组合来重建。在这样的实施方式中,对象30的复本可相应于所生成的部分或这些部分的某些组合。与存储对象数据的多个完整备份比较,这样的方法可提供有效的容错能力,同时减小数据存储要求。然而,注意,在一些实施方式中,冗余编码技术还可结合对象数据的完整复制而被使用。例如,对象数据的多个单独的完整备份可存储在节点160中,作为根据如上所述的适当的冗余编码技术确定的多个部分的相应集合。最后,注意,在一些实施方式中,某些对象30根本不需要以任何程度的复制或容错能力存储。例如,如下面结合存储类的描述所述的,客户可要求对象30根据指定很少或根本没有任何程度的容错能力的存储类来存储,可能具有比指定较高的容错能力程度的存储类更低的成本。

[0102] 一般而言,键映射实例40可提供对象30的键和对象30的特定实例或复本的定位器之间关系的记录。在存储这样的记录中,键映射实例140 还反映对象30在存储系统内复制

的程度(例如,存在多少对象30的实例,以及可如何引用它们)。位存储节点160通常可为对象30的各个实例提供存储,如由定位器识别的。然而,给定节点160可能意识到实例相对于任何其它节点160的状态,或实例的定位器和其相应对象30的键之间的关系。也就是说,一般而言,由键映射实例140维护的状态信息对位存储节点160来说可以是透明的。DFDD 110可操作来检测和传送关于节点160 的操作状态和/或键映射实例140(以及复制器键映射190,如果被实现)的状态信息,以便DFDD110的客户例如协调器120和复制器180可获得所检测的状态的准确、虽然可能延迟的观察。下面更详细地专注于这些部件。

[0103] 在图3中示出了说明图2的存储服务系统体系结构的某些组件的物理部署的一个实施方式。在所示实施方式中,数据中心300显示为包括两个区域310a-b。此外,区域310c-d显示在数据中心300的外部,且区域310a-d 通过网络60相互连接。区域310c-d中的每个都包括各自的协调器实例 120a-d。区域310a-d还可包括位存储节点160和键映射实例140的各种组合,以及在图3中没有示出的图2的其它组件。例如,区域310a包括四个位存储节点160,区域310b包括三个位存储节点160和键映射实例140,区域310c包括两个位存储节点160,以及区域310d包括一个位存储节点 160和一个键映射实例140。

[0104] 如上所述,在一个实施方式中,每个区域310a-d可被考虑为独立的或弱相关故障的位置。也就是说,任何给定区域310经历故障的概率可能通常与任何其它给定区域310的故障的概率无关或不相关,或故障概率的相关性可小于阈值量。例如,两个区域310可表现出小于10%的同时故障的可能性。故障相关性或无关性可使用任何适当的统计或概率技术测量并以各种方法执行。例如,区域310可与独立的效用网格(utility grid)物理分离或连接,可能致使影响一个区域310的灾难将不影响另一个区域。类似地,在数据中心300内,不同的区域310可具有独立的备用电源、网络连接或其它冗余资源,其可起作用来使一个区域310能够继续操作而不管另一区域310的故障。

[0105] 注意,在一些实施方式中,在其各自的故障可能性之间有小的但非零相关性的两个区域310可仍然被认为具有独立的故障可能性。例如,尽管每个区域310都具有用于备用电源、冷却等的稳健的和独立的系统,但是在给定数据中心300内的两个区域310可能在足够大的灾难(例如,足以摧毁整个数据中心300的爆炸)的情况下易受并行故障的影响。然而,足以使这两个区域310同时出故障的事件的概率可能足够小,为了实际的目的,两个区域310可假定为具有无关的故障可能性。

[0106] 区域310可包括层次的附加级别(未示出)。例如,在一个实施方式中,虽然可使用任何适当的区域组织,但是区域310可细分成格栅(rack),格栅进一步细分成单独的节点,例如位存储节点160。一般而言,区域310 可包括足以实现在区域内部署的存储服务系统组件的计算资源。例如,每个位存储节点160可实现为自主的计算机系统,该计算机系统可包括如下面结合图4-9的说明所述的各种硬件和软件组件。类似地,每个键映射实例140可通过多个计算机系统来实现,如下面结合图10-22的说明所述来配置这些计算机系统。

[0107] 在一些实施方式中,组件例如网络服务平台100、协调器120、节点拣取器130、复制器180和DFDD 110可通过每个区域310内分立的计算资源来实现,组件被部署在区域310内。例如,这些组件中的每一个都可实现为可由相应的计算机系统执行的一组指令和数据。可选地,这些组件中的一些或全部可实现为可同时在一个或更多计算机系统上执行的过程。在一些实施方式中,用于实现这些组件中的一些或全部的计算资源可与用于实现位存储节

点160或键映射实例140的那些资源一起被共享。例如,计算机系统可配置成实现键映射140功能以及协调器120功能的一些部分。一般而言,可使用在单独的区域310内部署的计算资源上的图2的组件的任何适当的分割。注意,如图3所示,不同的区域310可包括存储服务系统组件的不同组合,且所示实施方式旨在为说明性的而不是限制性的。

[0108] 此外,不同的存储服务系统组件可根据任何适当类型的通信协议来进行通信。例如,在图2的某些组件被实现为分立的应用程序或可执行的过程的场合,它们可使用标准的过程间通信技术或通过使用标准或专用的与平台无关的通信协议来彼此通信,标准的过程间通信技术可由操作系统或平台(例如,远程程序调用、队列、邮箱、套接字等)提供。这样的协议可包括可支持任意级的信号交换/确认、错误检测和校正或通信组件可能需要或希望的其它通信功能部件的有状态或无状态的协议。例如,在一个存储服务系统实施方式中,相当程度的组件间通信可使用适当的互联网传输层协议例如传输控制协议(TCP)的一种版本、用户数据报协议(UDP)或类似的标准或专用的传输协议来实现。然而,还可以设想,在存储服务系统组件之间的通信可使用在协议抽象的较高层的协议来实现。例如,与客户50和网络服务接口100之间的通信一样,存储服务系统组件之间的通信可使用应用层协议例如在HTTP上的网络服务请求来进行。

[0109] 位存储配置

[0110] 如上所述,在图2所示的存储服务系统体系结构实施方式中,位存储节点160可通常操作来为由存储服务系统管理的不同对象30提供存储。在图4中示出位存储节点160的一个示例性实施方式。在所示实施方式中,位存储节点160包括配置成与存储重装器163和逻辑文件输入/输出(I/O)管理器165通过接口连接的存储节点管理(SNM)控制器161。管理器165配置成与文件系统167通过接口连接,文件系统167又配置成管理一个或更多存储设备169。在不同实施方式中,SNM控制器161、存储重装器163、逻辑文件I/O管理器165或文件系统167中的任何一个可实现为可存储在计算机可访问的介质上或可由计算机执行的指令,以执行下面描述的功能。可选地,这些部件中的任一个都可由专用硬件电路或设备实现。

[0111] 在一个实施方式中,SNM控制器161可配置成向节点160的客户提供对象存储API,以及协调节点160的其它组件的行为以根据API来完成动作。例如,控制器120可配置成通过由SNM控制器161提供的API来将对象30存储到给定节点160并从给定节点160取回对象30。虽然API管理在这里被描述为SNM控制器161的功能部件,可以设想,在一些实施方式中,节点169的API处理功能可以与SNM控制器161不同的模块或组件实现。

[0112] 对象存储API可支持对象放置、获取和释放操作。在一个这样的实施方式中,一般也可称为存储操作或写入操作的对象放置操作可将对象30的数据和/或元数据指定为操作的变元或参数。当对于给定节点160完成操作时,放置操作可向发出请求的客户返回相应于所存储的对象30的定位器,该定位器可相对于在整个存储服务系统中存储的所有其它对象30来唯一地识别在给定节点160上的对象实例。

[0113] 相反地,一般也可称为读取或取回操作的对象获取操作可将对象30的定位器指定为参数。当完成操作时,获取操作可向发出请求的客户返回相应于所指定的定位器的对象数据和/或元数据。在一些实施方式中,获取操作可支持允许发出请求的客户指定对象数据、元数据或两者是否返回到客户的参数。

[0114] 与获取操作一样,一般也可称为删除或消除操作的对象释放操作可将对象30的定

定位器指定为参数。然而,当完成操作时,释放操作可释放以前与参考的对象30关联的存储资源,且这样的资源可接着用于存储其它对象30。在一个实施方式中,一旦定位器被释放,对定位器的随后的获取操作在一段时间内可以接着发生或可以不接着发生。也就是说,释放操作可用作对节点160的信号,它可释放存储资源用于重新使用,但节点160可能不试图立即这么做或通知客户或另外地使这样的重新使用与客户同步。因此,在其释放之后的客户继续试图访问对象30可在任意的时间段内接着发生,其后对象30可变得在没有通知时不可访问。在其它实施方式中,节点160可配置成阻止客户访问以前被释放的定位器,而不管对象数据是否仍然是可用的。

[0115] 可以设想,在不同实施方式中,放置、获取和释放操作可根据任何适当的协议来使用其它参数和/或返回不同的状态、错误或其它指示。例如,如果在节点160上对于待存储的被请求的对象30没有足够的资源,或如果由于某种其它原因放置不能被完成,则放置操作可返回错误情况。还可设想,在一些实施方式中,节点160的对象存储API可包括其它操作。例如,API可配置成通过支持复制操作来促进对象复本的建立。在一个实施方式中,除了不是向目标节点160提供待存储的对象30的数据,而是请求的客户可在不同的节点160上指定该对象30的定位器外,复制操作可与放置操作类似地操作。目标节点160可接着与指定的节点160进行交互作用以获得对象数据和/或元数据,并可将关于目标节点的对象定位器返回给客户。在其它实施方式中,节点160可支持对对象30的其它适当的操作。

[0116] 注意,在实现如上所述的放置、获取和释放操作的一些实施方式中,现有对象30可以不在原位被修改。更确切地,对象30的实例可通过在写入包括修改的数据的新实例之后通过释放现有的实例来被有效的修改。如果对象30的修改致使它小于或大于其原始大小则例如通过减小可能出现的分段存储或对象再定位,这样的方法可简化节点160的基本管理层的实现。如下面关于网络服务平台100更详细描述,在一些实施方式中,存储服务系统可支持将大的对象分裂成块,其中每个块可作为不同的对象30而被管理。这种方法可在处理大的对象中提高节点160的性能,通过限制可能需要被重写的块的范围可频繁地修改该大的对象。然而,可以设想,在其它实施方式中,节点160可包括支持对象30在原位的修改而不是通过刚刚描述的释放-重写方法所必需的那些功能部件。

[0117] 在所示实施方式中,逻辑文件I/O管理器165(或简单地,管理器165)可配置成虚拟化基本设备或文件系统特征,以便给SNM控制器161和重装器163提供一个或更多对象30可驻留的逻辑上邻接的存储空间。例如,给定对象30可根据其在存储空间内的偏移和其距离该偏移的范围(extent)(例如,根据对象大小,包括数据和元数据)来被定位在逻辑存储空间中。通过提供这样的逻辑存储空间,管理器165可向SNM控制器161提供基本存储的一致呈现,而不管这样的基本存储的实现细节。

[0118] 为了便于对逻辑存储空间内的对象30的访问,在一个实施方式中,管理器165可配置成向存储到节点160的每个对象30分配对象索引值(也称为对象索引)。一般而言,任何给定对象30的索引在特定节点60内可以是唯一的。例如,在一个实施方式中,对象索引可通过每当对象30存储到节点160时递增计数器并使用最终的计数器值作为对象索引来获得。(在允许多个对象写入操作同时进行的实施方式中,可通过例如串行化来同步计数器递增,以确保以一致和可预测的方式分配对象索引值。)足够大的计数器值,例如64位无符号整数例如可确保为了实际目的而给每个对象30分配唯一的索引值。这样的计数器可在比方说

264个对象被存储之后滚动循环,其后可重复以前产生的索引值。然而,冲突是非常不可能的,因为在计数器滚动循环之后以前被分配了给定索引值的对象30仍然存在于节点160内是非常不可能的。注意,用于分配对象索引的任何其它适当的方法也可被使用。如下所述,对象索引值可结合节点160的唯一标识符使用,以确定可由协调器120或节点160的其它客户使用来参考特定对象 30的定位器值。

[0119] 管理器165可配置成使用上面描述的唯一对象索引值来以便于对象访问的方式组织定位在逻辑存储空间内的关于对象30的信息。例如,如在图5的上部分中示出的,在一个实施方式中,管理器165可配置成存储表格或类似的数据结构,所述表格或数据结构可被组织成通过对象索引值而被容易地访问。在所示实施方式中,索引表500可包括很多项目510,其中每个项目都可包括很多字段,包括对象索引字段、偏移字段、对象大小字段、元数据大小字段和循环冗余码校验(CRC)字段。如在图5的下部分中,对于一些示例性对象30示出,项目510的偏移字段可指定相应的对象30在逻辑存储空间内的开始位置,以及对象大小和元数据大小字段可指定对象数据和元数据从偏移点延伸所达到的程度。虽然对象数据和对象元数据的顺序在其它实施方式中可颠倒,但是在所示实施方式中,对象数据在对象元数据之前。CRC字段可存储循环冗余码校验算法或其它适当类型的校验和或散列算法的结果。当对象30最初存储到节点160时,可计算最初存储到CRC字段中的值。随后,当访问对象30时,相同的算法可应用于对象数据和/或元数据,且作为结果的值与所存储的CRC字段值相比较。如果该比较导致失配,则可能损害所存储的数据的完整性。注意,在其它实施方式中,项目510可包括附加字段或与所示的那些不同的字段。例如,CRC字段可被省略或在其它地方实现。此外,代替地或除了相对偏移以外,还可存储对象数据和元数据的绝对位置。

[0120] 重装器163可配置成对逻辑对象存储空间进行操作,以消除当对象30 被释放且其相关联的存储资源被重新使用时可能出现的间隙。在一个实施方式中,重装器163可配置成扫描逻辑对象存储空间(例如,周期性地或连续地),以识别已经被SNM控制器161和/或管理器165标记为被前面的释放操作释放的对象30。重装器163可接着使具有出现在被释放的对象 30的索引之后的索引的那些对象30的项目510被更新以反映被释放的对象30的消除,这可有效地产生朝着逻辑对象存储空间的原点移动的那些对象30。例如,如果在图5的下部分中的对象N将被释放,则重装器163 可操作来使相应于对象N+1的项目510被更新以将对象N的偏移字段表现为对象N+1的新的偏移字段。重装器163还可使与对象N关联的项目 510被删除,并可更新在对象N+1之后的对象的偏移以反映移位。在一个实施方式中,管理器165可使在逻辑对象存储空间和/或存储设备169下面的文件或结构内出现对象数据和元数据的相应移位。因此,在一些实施方式中,重装器163的操作可减少下层存储结构的分段存储并相应地提高节点160的对象访问性能。

[0121] 在一些实施方式中,管理器165可配置成在包括不同类型的硬件和软件的多个不同的执行平台上执行。在一些这样的实施方式中,一个或更多另外的抽象层可存在于由管理器165提供给SNM控制器161的逻辑对象存储空间及其客户之间。例如,在所示实施方式中,管理器165可配置成将逻辑对象存储空间实现为由文件系统167管理的一个或更多物理文件。一般而言,文件系统167可配置成将不同类型的物理存储设备169组织到逻辑存储设备中,该逻辑存储设备可以逻辑单元存储数据,所述逻辑单元在这里称为物理文件。由文件系统167管理的逻辑存储设备可实际上是分层的。例如,文件系统167可支持可被定位

(navigated) 来存储和访问物理文件的目录或文件夹的层次。一般而言,文件系统167可配置成跟踪和管理给定物理文件和存储设备169的位置之间的关系,其中物理文件的相应的数据和/或元数据存储存储在存储设备169上。因此,在一个实施方式中,管理器165可管理逻辑对象存储空间到由文件系统167分配的一个或更多物理文件的映射。文件系统167又可管理这些物理文件到存储设备169的可寻址的位置的映射。

[0122] 文件系统167可通常集成在操作系统内,虽然任何给定的操作系统可支持提供用于管理下层设备169的不同功能部件的各种不同的文件系统 167。例如,Microsoft Windows®操作系统的不同版本支持文件系统,例如 NT文件系统 (NTFS) 以及FAT32 (文件分配表-32) 和FAT16文件系统。Linux和Unix操作系统的不同版本可支持文件系统,例如 ext/ext2文件系统、网络文件系统 (NFS)、Reiser文件系统 (ReiserFS)、快速文件系统 (FFS) 和许多其它的文件系统。一些第三方软件厂商可提供用于与各种计算平台集成的专用文件系统,例如VERITAS®文件系统 (VxFS)。不同文件系统可提供对于用于管理下层存储设备169的不同功能部件的支持。例如,一些文件系统167可提供对于实现设备镜像、分散连结 (striping)、快照或其它类型的虚拟化功能部件的支持。

[0123] 注意,在一些实施方式中,在管理器165和存储设备169之间可能仍然存在其它的抽象层。例如,在一些实施方式中,体积 (volume) 管理器层可设置在文件系统167和存储设备169之间,并可配置成执行上面提到的一些或所有虚拟化功能部件。可选地,特定的存储设备169可配置为硬盘驱动器或包括虚拟控制器的其它设备的独立阵列。虚拟控制器可配置成向文件系统167提供磁盘驱动器作为单个物理设备,虽然虚拟控制器在内部可支持设备的存储地址空间到磁盘驱动器的任意复杂的映射,但是类似于如上提到的可由体积管理器支持的或在文件系统167内的虚拟映射。还注意到,在一些实施方式中,可能存在比那些所示的抽象更少的层。例如,在一些实施方式中,管理器165可配置成与例如作为原始物理设备的存储设备169直接进行交互作用,而不使用文件系统167。

[0124] 一般而言,存储设备169可包括可由文件系统167和/或管理器165支持的任何适当类型的存储设备。存储设备169可通常包括硬盘驱动器设备,例如小型计算机系统接口 (SCSI) 设备或AT附加编程接口 (ATAPI) 设备 (其也可称为集成驱动电子 (IDE) 设备)。然而,存储设备169可包含包括基于磁或光介质的设备的任何类型的海量存储设备、固态海量存储设备 (例如,基于非易失性或“闪”存储器的设备)、磁带等。进一步地,存储设备169可通过除了上面提到的那些以外的任何适当的接口类型而被支持,例如遵循通用串行总线或IEEE 1394/Firewire®标准的版本的接口。

[0125] 如上所述,对于存储在存储设备系统内的对象30的任何给定的实例,相应的定位器可唯一地识别在系统内所有节点160上的实例。在一个实施方式中,定位器可生成为对象索引值的级连、组合或其它函数,该对象索引值以及相应于上面存储对象实例的节点160的唯一标识符或“节点ID”可由管理器165分配给对象实例。例如,如上所述,64位对象索引值可与64位节点ID组合以产生128位定位器。这样的定位器允许多达264个唯一的节点160中的每一个存储多达 2^{64} 个唯一的对象实例,虽然在不同实施方式中更小或更大数量的位可用来形成定位器。

[0126] 在一个实施方式中,节点ID可通过唯一的网络地址,例如相应于给定节点160的互联网协议 (IP) 地址的级连接或组合来形成,并具有时间戳或日期戳。例如,根据IP地址 (例

如,在节点启动/初始化时或如果不在初始化期间则在节点ID被分配时)与反映IP地址被分配的时间或IP地址被得知是有效的时间期间的组合,可给节点160分配节点ID。一般而言,属于同一ID地址空间的两个不同的节点160不能在任何给定的时间被有效地分配相同的IP地址。因此,节点的IP地址和时间戳值的组合可产生对该节点来说唯一的标识符。例如,32位IP地址可与32位时间戳(例如,其表示自从某个公共参考时间以来经过的秒数)级连或组合以产生上面提到的64位节点ID,虽然可使用其它位宽度。还可以设想,可使用其它技术来分配不依赖于节点IP地址的唯一的节点ID。例如,授权中心例如命名服务器可在被请求时以保证节点ID的唯一性的方式来委派节点ID,类似于如上所述在节点160内对象索引值的分配。

[0127] 注意,在节点ID得自节点的IP地址的实施方式中,节点ID可不在任何给定的时间反映节点160的当前IP地址。例如,节点ID可继续存在直到节点160被重置,但节点的IP地址可在节点ID产生之后被改变或重新分配。此外,在一些实施方式中,节点ID可以确定性的方式被散列、加密或混乱化,以便防止存储客户50或其它可能恶意的实体解码定位器以确定实际的节点IP地址。

[0128] 图6中示出了关于图4的节点160的实施方式的获取、放置和释放操作的示例性实施方式的操作。首先参考图6,获取操作可在块600中开始,其中操作在节点160处从协调器120或其它客户接收。例如,协调器120可向包括节点ID和对象索引值的特定的定位器发出获取操作,如上所述。例如,如果节点ID反映目标节点160的当前IP地址,则节点ID可用于直接将获取操作路由到适当的节点160。可选地,目录服务例如下面描述的DFDD 110可用于将定位器的节点ID解析到可寻址的端点或目的地中,获取操作可通过该端点或目的地而被路由到适当的节点160。

[0129] 一旦被节点160接收,则获取操作就可被处理以识别节点160的逻辑对象存储空间内的目标对象实例的范围(块602)。例如,控制器161可接收获取操作并将它传送到管理器165。管理器165又可使用通过获取操作所引用的定位器的对象索引部分来访问索引表500,以便获得逻辑对象存储空间内的期望的对象实例的位置。例如,管理器165可获得在对象实例开始位置的逻辑对象存储空间内的偏移,以及该对象实例距离该偏移的长度。在一些实施方式中,获取操作可指定对象数据、元数据或两者是否是期望的。在这样的实施方式中,管理器165可确定与被请求的数据有关的逻辑对象存储范围。例如,如果对象数据和元数据都是期望的,则管理器165可使用对象数据大小和元数据大小来确定距离待取回的对象偏移的范围。如上所述,在其它实施方式中,管理器165可以不同的方式来存储和管理对象实例的存储范围,例如通过绝对位置而不是逻辑对象存储空间内的相对偏移。

[0130] 在逻辑对象存储空间内的对象范围可接着被映射到物理文件存储空间内的一个或更多相应的文件内的范围上(块604)。例如,管理器165可将逻辑对象存储空间映射到由文件系统167管理的一个或更多文件上,并可向文件系统167发出适当的文件访问操作,以例如通过参考一个或更多文件名以及待读取的命名文件内的位置或偏移来获得相应于所期望的对象范围的数据。可以设想,在可选的实施方式中,控制器161可配置成绕过由管理器165管理的逻辑块存储空间功能部件,并可改为直接与文件系统167管理的物理文件进行交互作用。

[0131] 对物理文件的参考可接着被映射到与设备有关的请求(块606)。例如,文件系统

167可配置成生成对存储设备169的特定可寻址的位置的一个或更多读取请求,其中特定可寻址的位置例如逻辑块地址(LBA)或设备几何结构(例如,圆柱体、轨道、扇区和/或磁头)所特有的地址。如上所述,在一些实施方式中,管理器165可配置成绕过文件系统167并直接管理存储设备169。

[0132] 被请求的对象数据可接着被从存储设备169取回(块608)并被返回到发出请求的客户(块610)。例如,取回的数据可被向上返回通过图4所示的请求层次,或可从存储设备169或文件系统167直接返回到控制器161以传送到发出请求的客户。

[0133] 如图7所示,在一个实施方式中,在块700中当操作在节点160处从协调器120或其它客户接收放置操作时,放置操作可以类似于上面对图6的块600描述的方式开始。例如,协调器120可向在由节点拣取器130产生的写入计划中指定的节点160发出放置操作。与获取操作相反,放置操作可包括待存储的对象数据和/或元数据,并可以可选地包括指定数据和/或元数据的长度的附加参数。

[0134] 一旦被节点160接收,则放置操作就可被处理以为逻辑对象存储空间内的对象实例分配存储范围(块702)。在一个实施方式中,管理器165可配置成向新的对象实例分配对象索引值,并在索引表500中记录指定新的对象实例的偏移的新项目510。例如,新项目的偏移可相对于具有最大索引值的现有对象实例的存储范围(例如,偏移和长度)而被确定。如果新的对象实例的数据和/或元数据的长度没有被指定为放置操作的参数,则管理器165或控制器161可配置成计算所述数据和/或元数据的长度以包含在新项目510中。

[0135] 逻辑对象存储空间内新近分配的存储范围可接着被映射到物理文件存储空间内的一个或更多相应的文件内的范围上。例如,新的对象实例的被分配的范围可附加到一个或更多现有物理文件的末尾,或另外地位于现有的或新近分配的物理文件内。物理文件范围可接着例如由文件系统167以类似于上面对获取操作描述的方式而被映射到存储设备范围(块706),且对象实例数据和/或元数据可接着存储到存储设备169(块708)。

[0136] 当确认数据和/或元数据已经被成功地写入到存储设备169时,相应于所存储的对象实例的定位器可返回到发出请求的客户(块710)。例如,管理器165可配置成将产生的对象索引值附加到节点160的节点ID上,并且当从文件系统167指示物理文件写入操作成功地完成时可返回最终的值作为对象定位器。

[0137] 如图8所示,在一个实施方式中,在块800中当操作在节点160处从协调器120或其它客户接收释放操作时,释放操作可以类似于上面对图6的块600描述的方式开始。释放操作可简单地指定待释放的对象实例的定位器,虽然在其它实施方式中也可提供其它变元。

[0138] 与获取操作一样,一旦释放操作被节点160接收,释放操作就可被处理以识别节点160的逻辑对象存储空间内的目标对象实例的范围(块802)。例如,控制器161可接收释放操作并将其传送到管理器165。管理器又可使用被释放操作参考的定位器的对象索引部分来访问索引表500,以便识别所参考的对象实例的相应项目510。所参考的对象可接着被标记为被释放的(块804)。例如,管理器165可配置成将偏移或项目510的另一字段设定为非法值,例如负数,其中负数可表示该项目不再是有效的。指示对象已被释放的确认可接着返回到发出请求的客户(块806)。

[0139] 如上所述,当对象实例被释放时,与对象实例相关联的存储资源不能被立即释放、收回或重新分配以用于其它用途。更确切地,在一个实施方式中,那些资源可继续存在直到

相对于释放操作异步地进行操作的独立过程收回它们为止。图9示出这样的过程的一个实施方式的操作,例如可通过存储重装器163来实现。在块900中,可选择相应于存储在节点160上的特定对象实例的对象索引项目。例如,重装器163可配置成根据存储在项目中的对象索引值来以连续的顺序从索引表500中选择索引项目。随后,可检查所选定的项目以确定相应的对象实例是否已被释放(块902)。例如,重装器163可检查偏移字段或另一字段以确定该字段是否已被设定为指示相应的对象实例已被释放的值,例如负值或一些其它值。

[0140] 如果所选定的对象还没有被释放,则操作可继续后退到块900,其中可选择另一对象。如果所选定的对象已被释放,则逻辑对象存储空间可被重装以收回相应于所释放的对象的存储资源(块904)。例如,重装器163可配置成调节跟随逻辑对象存储空间内的所释放的对象的那些对象实例的索引项目510,以便第一个这样的对象实例的偏移被设置成所释放的对象的偏移,下一个这样的对象实例的偏移被设置为第一个这样的对象实例的数据大小、元数据大小和偏移的函数,等等。然而,在一些实施方式中,不是在所释放的对象实例之后的所有对象实例都需要在新的对象被选择用于检查之前被重装。例如,重装可与对象选择交替进行,以便遇到的每个对象在它被选择用于检查时被重装。

[0141] 在一些实施方式中,管理器165可响应于逻辑对象存储空间的重装而执行物理文件存储空间内类似的重装或合并操作。例如,管理器165可使逻辑对象数据范围被重新映射到不同的物理文件数据范围。类似地,在一些实施方式中,文件系统167可响应于物理文件存储空间的重装而在存储设备169中执行类似的重装或合并操作。在其它实施方式中,物理文件存储空间或存储设备本身的重装可独立于由重装器163发起的逻辑对象存储空间的重装而出现。例如,文件系统167可配置成通过重新排列物理文件存储范围到设备存储范围的映射来为存储在存储设备169上的物理文件整理碎片,以便所映射的设备存储范围相对于存储设备的访问模式来说大部分或全部是连续的。

[0142] 在逻辑对象存储空间的重装之后,相应于所释放的对象的索引项目可被删除(块906),且操作可从另一对象被选择的块900继续。如上所述,在一些实施方式中,重装可在对象被选择时“即时(on the fly)”出现,这可提高逻辑对象存储空间的总利用,同时最小化重新定位对象所需要的操作的数量。

[0143] 注意,在一些实施方式中,可被节点160支持的获取、放置、释放或其它操作中的任何一种操作可支持关于发出请求的客户的各种类型的信号交换、确认或错误处理协议。例如,如果客户请求对于一种操作的残缺的请求(例如,不能提供必要的参数),或如果节点160不能良好地完成操作(例如,没有充足的资源来兑现放置操作),则节点160可将错误指示返回给发出请求的客户。这样的指示可以包括或不包括关于故障情况的性质的特定细节。

[0144] 在一个实施方式中,协调器120可配置成将操作独立传送到该操作所指向的每个相应的节点160,即使当多个操作可具有共同的数据时也是如此。例如,在放置操作的情况下,其中对象30根据写入计划被写到多个节点160,协调器120可与每个指定的节点160独立地进行通信。然而,在可选的实施方式中,可链接具有共同数据和/或参数、旨在多个目标节点160的操作。在一个实施方式中,协调器120或其它客户可通过指定在操作的参数中的每个接收者例如接收者列表来开始所链接的操作。在操作中指示的多个接收者可默认地表示链接,或其它参数可用于将操作标记为被链接的。协调器120或其它客户可接着通过将链接

的操作传送到在操作中指定的第一个目标节点160而开始该链接的操作。

[0145] 当接收该链接的操作时,节点160可处理该操作并可将其转发到在操作中指定的另一个目标节点160。在这样的转发之前,接收者节点160可将其自身从包括在该操作中的目标的列表中删除,以表示接收到并避免循环转发。该操作可与接收者节点的处理同时被转发。可选地,转发可能发生在接收者节点成功地完成处理时。在一些实施方式中,链接的操作可以接收者被指示在该操作中的顺序而被传送到那些接收者。在其它实施方式中,节点160可例如通过确定剩余目标中哪一个是最接近的、最少负载的或满足一些其它选择标准的,来动态地选择下一个接收者。注意,在一些实施方式中,链接的和非链接的操作的组合可由协调器120或其它客户产生。例如,如果相同的数据是前往六个不同的节点160的放置操作的目标,则协调器120可产生指定这六个目标节点的单个链接的操作,或两个链接的操作,其中每个操作指定目标节点中的三个目标节点。其它组合也是可能的,包括产生协调器120可独立地传送到相应的目标节点160中的每一个的六个非链接的操作。

[0146] 键映射配置

[0147] 如上所述,各种位存储节点160可配置成为对象30的实例提供存储。节点160不可单独地为冗余和数据安全提供任何特定的支持;事实上,在一些实施方式中,节点160可使用运行开放资源操作系统(例如,Linux)并通过廉价的商品性硬盘驱动器(例如,ATAPI/IDE硬盘驱动器)来提供存储的一般计算平台来实现。在这样的实施方式中,各个系统可能不是容错能力特别好的。更确切地,数据安全和冗余可通过对象30在很多节点160上的复制而被提供,如上所述。

[0148] 如前面讨论的,给定对象30可相应于可被存储客户指定的键。给定对象30的各个实例可相应于相应的定位器,这些定位器可唯一地识别被包括在存储服务系统中的节点160的集合中的那些实例。在一个实施方式中,部署在存储服务系统内的每个键映射实例140可配置成为存储和维持键和用于给定对象30及其存储在节点160中的复制的实例的所有相应的定位器之间的关系或映射。在下面的讨论中,讨论了键映射实例140的不同实施方式的一般特征和功能,接着是键映射实例140的特定实施方式可如何被实现的描述。

[0149] 在一个实施方式中,给定键映射实例140可配置成将各个键和相关联的定位器之间的关系的细节存储在一个或更多表或任何其它适当类型的数据结构中。例如,在一个实施方式中如图10所示,键映射实例140包括具有很多项目144的键映射数据结构142。每个项目包括相应的键146 以及相关联的记录148。在一些实施方式中,如下面更详细描述,用于组织项目144的数据结构的组织可能是复杂的。然而,从功能性的观点来看,键映射实例140可通常保存给定键144及其相应的记录148之间的一对一的表格状的关系。

[0150] 记录148可一般包括相应于给定键144的定位器,但也可包括其它信息。例如,记录148的一个实施方式可如下构造:

```
Struct KeyRecord{  
    int16_t version;  
    int16_t storageClass;  
    int16_t creationDate;  
    int64_t objectSize;  
[0151]    uint32_t crc32;  
    int8_t numLocators;  
    struct locator{  
        int64_t nodeID;  
        int64_t objectIndex;  
        }replicas[];  
[0152] }
```

[0153] 虽然该示例性数据结构使用C编程语言的语法来表达,它可使用任何适当的语言、表示或格式来实现。记录148的可选实施方式可包括比所示的那些更多、更少或不同的字段。在一些实例中,记录148可称为“i节点(inode)”,将记录148在组织存储空间中的目的的相似性用于某些类型的Unix文件系统中采用的i节点结构。然而,术语“i节点”在本上下文中的使用不是用来调用文件系统或其它存储环境内i节点的实现或使用的特定细节。

[0154] 在上面的实施方式中,记录148可包括七个特定的元素。16位的版本元素可用于存储对于记录148的格式特定的唯一识别的值。例如,记录148 的不同版本可用在键映射实例140的不同实现中,且在一些实施方式中,存储在给定键映射实例140中的记录148可为不同种类的。版本元素可用于区分记录148的不同版本,以便记录的其它元素可被正确解码和使用。

[0155] 16位storageClass元素可用于存储相应于记录148的对象30的存储类的指示。在随后的部分中更详细地描述了存储类。一般而言,对象的给定存储类可识别存储特征和/或策略,其对给定存储类的其它成员可能是共有的,但可不同其它存储类的成员。例如,对于存储服务系统的给定实现可定义“高可靠性”存储类和“低可靠性”存储类。作为高可靠性存储类的成员的对象30可比作为低可靠性存储类的成员的对象30被更大程度地复制,因而降低对各个复本的损失的敏感性,可能是以比对低可靠性存储类的成员估算的更高的使用成本来交换的。存储类的很多其它可能的类型和组合是可能的和预期的。

[0156] 64位的creationDate元素可用于存储相应的对象30在存储服务系统内被创建的日期和时间的指示。该元素可以任何适当的方式格式化。例如,日期和时间可被明确地编码为元素内的不同字段,或表示自公共参考点以来经过的时间单元(例如,秒、微秒等)的数量的单个数字。在一些实施方式中,creationDate元素可包括配置成指示相应的对象30的任何方面的最后修改的日期和时间的附加字段,虽然在其它实施方式中,最后的修改元素可

作为记录148内的不同元素而被包括。

[0157] 64位的objectSize元素可用于例如以字节的形式存储相应对象的大小的指示。在一些实施方式中,该元素可反映对象数据和元数据的大小,而在其它实施方式中,这些可存储为不同的字段。32位的crc32元素可用于存储根据任何适当的校验和算法为对象数据和/或元数据计算的循环冗余码校验(CRC)校验和的指示。例如,可包括校验和以验证数据防讹误或篡改的完整性。在其它实施方式中,除了或代替CRC校验和,还可使用从对象数据和/或元数据计算的任何适当类型的散列信息或签名(signature)。

[0158] 8位的numLocators元素可用于存储包括在replicas[]阵列内的记录148 内的定位器的数量的指示。在该阵列内,每个定位器存储为64位的节点 ID元素以及64位的对象索引值,其可如上所述在对位存储节点160的配置的讨论中得到。在一些实施方式中,定位器可存储为replicas[]阵列内的单个元素。

[0159] 在一个实施方式中,键映射实例140可配置成向键映射客户提供键映射API,例如协调器120,以及配置成执行支持所提供的API所必需的那些功能。例如,控制器120可配置成使用API来对与键映射实例140管理的项目144关联的记录148进行存储、取回、删除或执行其它操作。类似于对如上所述可由节点160支持的对象实例的操作,在一个实施方式中,键映射API可支持对键映射项目144的放置、获取和删除操作。在一个这样的实施方式中,一般也可称为键映射存储操作或键映射写入操作的键映射项目放置操作可指定要存储在键映射项目144内的键146和记录148。在一个实施方式中,指定键146的放置操作可用被指定为放置操作的变元或参数的记录来代替与现有的项目144关联的记录148,其中对于键146来说项目144已经存在。当对于给定的键映射实例140完成时,键映射放置操作可向请求的客户返回状态指示,例如操作是否成功或失败,以及例如出现什么类型的故障(如果有)。在一些实施方式中,如果键映射放置操作导致现有项目144的替换,则键映射实例140可配置成将项目144的前面的值返回给请求的客户。

[0160] 也可一般称为键映射读取或取回操作的键映射获取操作可在一个实施方式中将键指定为参数。当完成时,键映射获取操作可向请求的客户返回与请求的键关联的键映射项目144的记录148,如果这样的项目存在的话。如果没有相应的项目144存在则可将对该结果的指示返回给请求的客户。

[0161] 在一个实施方式中,除了请求的客户不需要指定将写入到项目中的记录之外,键映射项目删除操作可配置成与放置操作类似地操作。当对给定的键映射实例140完成时,键映射删除操作可向请求的客户返回状态指示,类似于键映射放置操作的情况。与放置操作一样,在一些实施方式中,键映射实例140可配置成将删除的项目144的前面的值返回给请求的客户。

[0162] 键映射API还可支持在不同实施方式中的其它类型的操作。例如,键映射API可支持可帮助键映射客户管理键映射项目的操作。在一个实施方式中,键映射API可支持列表操作,其可配置成识别具有与请求的客户指定的一些标准匹配的键146的那些项目144。例如,列表操作可允许客户将字符串或模式指定为操作的参数。当对于给定的键映射实例140完成时,列表操作可给请求的客户返回满足所指定的字符串或模式的那些键146的列表。在一个实施方式中,只有当字符串是键146的正确前缀(例如,对于字符串的所有字符,字符串的第N个字符与键的第N个字符匹配)时,键146才可满足给定的字符串。在其它实施方式中,如

果字符串可在键146 内的任何位置找到,则键146可满足给定的字符串。

[0163] 在一些实施方式中列表操作可支持其它参数。例如,列表操作可允许请求的客户指定对待返回的匹配的数量的限制。此外,请求的客户可例如通过指定可扩充的或封闭式的按字母顺序的范围来指定对将被搜索的键 146的约束条件,其中将被搜索的键146应落在该按字母顺序的范围内。在一些实施方式中,键映射实例140可配置成返回记录148以及满足列表操作标准的键146。此外,在一些操作中,键映射API可支持计数操作,其可支持与列表操作相同类型的参数和执行行为。然而,不是返回满足由请求的客户提供的标准的那些键146和/或记录148,计数操作可返回满足那些标准的键的数量(例如,由相应的列表操作返回的键的数量)。注意,键映射API还可支持没有在上面详述的其它操作。

[0164] 在一些情况下,不同的键映射客户可试图修改相同的键映射项目144。例如,响应于各种客户或系统驱动的操作,两个不同的协调器120可试图同时改变给定记录148的内容(例如,以增加、删除或修改复本的定位器),或一个协调器120可试图修改记录148,同时另一个协调器120试图删除相应的项目144。为了提供用于解决对给定键映射实例144的同时请求的一致方法,在一个实施方式中,键映射API可要求至少那些更新或修改键映射状态的键映射操作(例如,键映射放置和删除操作)提供序列号作为键映射操作的参数。键映射实例140可接着配置成通过比较序列号(数字式或按字母顺序地)和在比较的基础上一致地挑选一个操作来解决对项目 144的冲突更新。在一些实施方式中,所提供的序列号连同如下面更详细描述用于同步恢复的修改的记录148一起可存储在修改的键映射项目 144中。

[0165] 例如,键映射客户可基于时间戳来产生序列号。在一个实施方式中,这样的时间戳可包括如下格式化的64位数字。时间戳的第63位可设定为零(例如,以避免关于时间戳是否有符号的或无符号的数字的混淆)。第62-32位可包括自从时间的参考点(例如,1970年1月1日午夜,格林尼治标准时间,被很多版本的Unix和Linux使用的参考时间)以来经过的秒数。第31-22位可包括自从最后一秒以来经过的微秒数。第21-0 位可包括实质上随机产生的位。在其它实施方式中,时间戳可在不同宽度或类型的字段的基础上产生。可选地,键映射客户可使用完全不同的基础来产生序列号。假定序列号的分辨率高,则在由不同键映射客户为相同的键映射项目144提供的不同序列号中间的冲突的机会可能低。然而,如果冲突将要出现,则键映射实例140可配置成使用任何适当的一致技术来解决冲突。

[0166] 在很多实施方式中,键映射实例140在将键映射到定位器中的抽象功能行为可能相对直接。例如,如上所述,由键映射实例140的一个实施方式支持的一组基本操作可包括配置成处理项目144的放置、获取和删除操作,该项目144反映键146和包括在记录148内的定位器之间的关系。然而,键映射功能在存储设备系统内的实现可提出很多挑战。特别是,如果存储服务系统要支持代表大量客户的大量对象30(例如,总数达1000千兆字节(TB)或1000TB字节(EB)的存储量或更多的成百万或十亿的对象30),则可能要求键映射的实现相应地在容量上调节。然而,实现足够的系统资源以在单个计算机系统内表示包含在键映射内的全部信息可能是不可能的或在经济上是不可行的。此外,对于键映射客户请求的容错能力和增加的总处理能力,可在存储服务系统内以分布式方式部署键映射数据的多个复本。然而,例如如果一个复本在另一个复本被访问时被修改,则键映射数据的复本可导致键映射同步和一致性问题。

[0167] 键映射功能的可缩放性可通过将层次级别引入键映射实例140内来提高。在图

11A-D中示出了这样的层次的一个实施方式。在图11A中,示出了示例性键映射部署1100。如上所述,例如,关于图3,在一些存储服务系统实施方式中,多个键映射实例140可分布在整个系统中,例如在不同的数据中心300或区域310中。通常,键映射实例的集合可称为部署。在一些实施方式中,存储服务系统可包含单个键映射部署1100,单个键映射部署1100包括在系统内提供的所有键映射实例140,虽然在其它实施方式中,系统可包括在键映射层次的其它级别下合并的多个键映射部署1100。

[0168] 在所实施方式中,部署1100包括键映射实例140a-c,其中每个键映射实例都配置成例如根据如下面更详细描述实例同步协议来与其它键映射实例交换键映射信息。如所示,每个键映射实例140包括配置成彼此通信的很多主机400。例如,键映射实例140a包括主机400a-c,键映射实例140b包括主机400d-g,以及键映射实例140c包括主机400h-j。一般而言,每个主机400可包括计算机系统和相关联的软件,并可包括例如处理器、系统存储器、存储设备、联网接口或其它适当的组件的元件。例如,计算机系统或节点的一个实施方式可配置成用作下面结合图29的描述所讨论的主机400。

[0169] 一般而言,对于存储在存储服务系统内的所有对象30,每个键映射实例140可配置成维持键映射数据的完整表示,包括键映射项目144以及用于索引并管理键映射层次的任何其它数据。在键映射实例140内,键映射数据可分布在主机400中,以便各个主机400存储键映射数据的某个(可能冗余的)部分。注意,虽然在图11A中只示出几个主机400,但是在其它实施方式中,每个键映射实例140可具有任何适当数量的主机140。例如,在一些大范围的实现中,许多的或者可能数百的主机140可以被包括在键映射实例140中。还可以设想,虽然在某些实施方式中,用于给定键映射实例140的主机400可局限于给定区域310或数据中心300内,但在其它实施方式中,这样的主机400可分布在不同区域310或数据中心300中。进一步地,虽然主机400可配置成在一些实施方式中只实现与键映射有关的功能,但在其它实施方式中,主机400可实现与存储服务系统的其它元件有关的功能。例如,在一个实施方式中,各个主机400还可配置为位存储节点160,因而可存储键映射数据以及对象数据。

[0170] 图11B更详细地示出键映射实例140a的示例性实施方式。在所实施方式中,包括在键映射实例140a内的每个主机400a-c包括相应的分区索引410a-c和任意数量的程序块(brick)415。一般而言,程序块415可相应于键映射实例140内的中间键映射数据结构。在一些实施方式中,如下面结合图12的描述更详细描述,键映射数据可在程序块415中间分成分区,以及在键映射实例140内分区的复制可出现在程序块级。分区索引410可配置成为程序块415编索引,以便于选择一个或更多特定的程序块415来在键映射操作期间进行处理。例如,分区索引410可配置为树或另一适当的数据结构。在一个实施方式中,分区键映射实例140内的索引410以及较深层的索引级可配置为称为分层的不平衡树或特里结构(trie)的特定类型的数据结构的一部分,这在随后的部分中被详细描述。在所实施方式中,键映射实例140进一步包括键映射协调器412。一般而言,键映射协调器412可配置成实现键映射访问管理、内容管理和同步方法或协议,如下面更详细描述的那些方法或协议。注意,虽然示出键映射协调器412与主机400不同,但在某些实施方式中,它可实现为在一个或更多主机400内的过程或模块。还注意到,在某些实施方式中,分区索引410可在键映射协调器412内而不是分开地在主机400内实现。

[0171] 图11C示出包括程序块415a-n的主机400a的示例性实施方式。如所示,每个程序块

415a-n包括相应的块索引420a-n以及任意数量的块425。一般而言,块425可相应于键映射实例140内的中间键映射数据结构,类似于程序块415,但属于程序块的抽象级。类似于分区索引410,块索引420可为配置成为程序块415内的块425编索引的任何适当的数据结构。例如,块索引420在一个实施方式内可配置为分层不平衡树的一部分。

[0172] 如图11D所示,在一个实施方式中,块425可配置成包括任意数量的各个键映射项目144a-n以及配置成为项目144编索引用于选择的项目索引 430。如前所述,每个项目144a-n都可包括相应的键146a-n以及相应的纪录148a-n的指示。

[0173] 在图12中概述了在图11A-D所示的实施方式的键映射实例140和键映射项目144之间的等级层次中的关系。在包括多个键映射实例140的抽象的部署层中,特定的键映射实例140可参考在抽象的实例层的分区索引 410。参考的分区索引410可识别相应于特定项目144的一个或多个程序块415。例如,在所示实施方式中,所有的键映射项目都由相应于不同的程序块415的三个不同分区复制。给定的程序块又可通过块索引420参考特定的块425(未在图12中示出),且参考的块可通过项目索引430指向特定的项目144。注意,虽然键映射可使用例如在图12中所示的等级实现来实现,但是其它实现是可能的。泛泛地说,可以设想,键映射实例140 可使用将键144与纪录148关联的任何适当的技术来实现。例如,在一个实施方式中,键映射实例140可使用传统的数据库或其它类型的结构化索引来实现。

[0174] 注意,在图12的实施方式中的一些等级层次可配置成提供冗余(例如,在部署层内键映射实例140的复制以及在分区层的程序块415的复制),而其它层可配置成提供可缩放性。例如,当在键映射部署内要被编索引的项目144的数量增加时,在多个不同层中索引(分区索引410、块索引420 和项目索引430)的分布可通过允许索引的每个部分以易于管理的方式扩大来便于数据结构的缩放。注意,在其它实施方式中,可使用更多或更少的等级层次以及冗余和不冗余层的不同组合。

[0175] 如同对象30一样,在键映射等级层次内复制的使用可通过降低对各个复本的损失的敏感度来提高容错能力。然而,如果在修改出现时不试图使键映射数据的复本同步,则键映射的正确(例如,最近的)状态可能变得模棱两可,这又可导致不可预知的或错误的系统操作。在一些实施方式中,键映射数据的复制部分可使用原子的或事务处理的语义(例如,两阶段提交语义)以严格同步的方式被更新,其中不将更新向键映射客户报告为完整的,直到它们关于每个复本被持久和可验证地完成。虽然原子更新语义可最小化或甚至消除了使键映射数据维持在不一致状态中的更新的可能性,但是原子更新的性能可能在相当大规模的分布式环境中相当大地退化。例如,如果键映射数据的复本被广泛地分布,则从客户的观点来看复本访问等待时间可相当大地变化,最慢的复本指示完成更新操作所需要的总时间。而且,如果一个复本失败,则严格的原子更新语义可使客户停机,直到故障被纠正,这可导致客户不可接受的延迟。

[0176] 可提供比原子协议更好的客户性能的其它类型的同步协议可在键映射等级内使用。在一些实施方式中,可实现混合同步方法,其中一种类型的同步协议可相对于特定键映射实例140内的复本(例如,在分区层的复本,如图12所示)使用,而另一类型的协议可用来同步键映射部署内的不同键映射实例140。这样的混合方法可允许尤其针对在键映射等级内的不同层的复本的使用动态来调整同步开销。

[0177] 例如,键映射数据访问可展示参考的位置,以便对特定项目144的重复请求指向比

另一键映射实例140更可能的特定的键映射实例140(例如,根据地理、网络拓扑或另一适当的标准最接近于请求的客户的实例)。也就是说,情况可能是,给定键映射实例140内的键映射数据的复本比在不同的键映射实例140内的相应键映射数据可能更可能被给定的客户访问。相应地,在一些实施方式中,给定键映射实例140内的复本可使用这种协议被同步,即该协议可配置成比用于同步不同键映射实例140的协议更快地收敛(例如,以传播复本中间的变化)。

[0178] 在一个实施方式中,给定键映射实例140内的键映射数据复本的同步可使用适当版本的定额协议(quorum protocol)来执行。一般而言,根据定额协议执行的键映射数据的复本的更新或修改(包括键映射放置和删除操作)在该修改相对于至少定额数量的复本被持久地(例如,完全地和持续地)执行时可被认为相对于请求的客户完成。类似地,当相同的数据从至少定额数量的复制本取时,根据定额协议执行的键映射项目获取操作可被认为完成。在一些实施方式中,定额数量可被定义为出现的复本的数量简单大多数,而在其它实施方式中,可使用任意程度超级大多数。注意,如果不满足定额要求,则定额协议操作可能不能完成。然而,如果复本的定额数量小于复本的总数量,则给定的定额协议操作失败的概率可能小于原子协议操作的失败概率,这有效地需要在复本中间的一致而不是定额。注意,键映射实例140可使用除了这里描述的定额协议以外的定额协议。例如,多阶段提交协议例如Paxos或两阶段提交可用于实现定额类型的键映射语义。

[0179] 在正常的读取操作和根据定额协议的更新操作的过程中,例如由于通讯故障或在复本下层的资源的故障,更新不能传播到每个复本是可能的。在一个实施方式中,复本中间的不一致可在读取操作期间被检测和修复。特别地,如果在键映射项目获取操作期间在特定项目144的不同复本中间检测到不同的值,则可产生键映射放置操作以调整差异。在一个实施方式中,用作放置操作的基础的项目144可为具有在读取的不同值中最近(例如,数字式或按字母顺序地最高的)关联的时间戳的项目。因此,复本中间的差异可“即时”被解决,例如,当键映射项目获取操作被处理时,而不需要不同的过程或操作修复该差异。

[0180] 在图13-14中示出了关于配置成实现定额协议的键映射实例140的键映射项目放置、获取、删除和列表操作的示例性实施方式的操作。在不同实施方式中,这些方法可在键映射协调器过程内实现,这些过程可配置在例如处于包括在键映射实例140内的一个或更多主机400内,或作为键映射实例140内的分离的过程或系统,例如图11B所示的键映射协调器412。首先参考图13,在块1300中,键映射项目放置操作可当操作在键映射实例140从协调器120或其它键映射客户接收时开始。例如,响应于特定对象30的相应对象实例存储到特定的位存储节点160,协调器120可产生键映射项目放置操作,以便更新对象30的项目144来反映存储的对象实例的定位器。

[0181] 键映射实例140的等级可接着被导航以识别相应于键映射项目放置操作的复本(块1302)。例如,对于图12的实施方式,可查阅分区索引410 以确定哪个程序块415复制相应于所关心的对象30的项目144。随后,各个放置操作可指向识别的复本(块1304)。对于每个放置操作,键映射实例140的剩余等级可被导航以访问和修改相应的项目144(块1306)。例如,在给定程序块415内,可遍历块索引420和项目索引430以便访问特定的项目144。一旦项目144的给定复本被成功地写入,则相应的放置操作就可指示成功(块1308)。注意,以项目144的相应复本为目标的各个放置操作可同时执行。相应地,块1306-1308的多个实例被

并行地示出。

[0182] 可监控各个复本放置操作的成功指示以确定复本的定额数量是否被成功地更新(块1310)。例如,在包括三个复本的实施方式中,用于完成键映射项目放置操作的复本的定额数量可能是两个。如果成功地更新了复本的定额数量,则可给请求的客户返回被请求的键映射项目放置操作已完成的指示(块1312)。如果没有更新,则监控可继续。在一些实施方式中,可实施超时,以便如果键映射项目放置操作没有在处理开始之后的特定时间段内完成,则该操作可终止且错误指示可返回给请求的客户。在其它实施方式中,键映射项目放置操作可保持无限地挂起,直到它完成。

[0183] 在一个实施方式中,键映射项目删除操作可实现为放置操作的特殊情况。在这样的实施方式中,键映射项目144可包括配置为删除标记或标志字段的附加字段,且删除操作可作为配置成将删除字段设定到有效状态的放置操作来执行(例如,通过将该字段设定为特定的值,例如“1”)。具有宣称的删除字段的那些项目144可在未来的键映射操作期间不予处理。在一些这样的实施方式中,分离的过程可配置成通过键映射实例144独立地重复,以清除具有宣称的删除字段的那些项目144。在其它实施方式中,这样的项目144可无限地保持为历史键映射行为的日志。

[0184] 图14中示出了键映射项目获取操作的操作方法的一个实施方式。在块 1400中,当获取操作在键映射实例140从协调器120或其它键映射客户接收时操作可开始。例如,响应于来自存储客户50的对相应于特定键的对象数据的请求,节点拣取器130或协调器120可产生键映射项目获取操作,以便获得相应于特定键的定位器,使得位存储节点160可被访问以取回如前面的部分中所述的对象数据。

[0185] 如同键映射项目放置操作一样,键映射实例140的等级可接着被导航以识别相应于键映射项目获取操作的复本(块1402)。随后,各个获取操作可指向识别的复本(块1404)。对于每个获取操作,键映射实例140的剩余等级可被导航以访问和取回相应的项目144(块1406)。一旦项目144 的给定复本被成功地取回,则相应的获取操作就可指示成功(块1408)。注意,如同上述和在图13中所示的各个放置操作一样,以项目144的相应复本为目标的各个获取操作可同时执行,且块1406-1408相应地被并行示出。

[0186] 可监控各个复本获取操作的成功指示以确定定额数量的复本是否被成功地读取(块1410)。如果没有,则监控可继续,直到附加的复本被读取。至于上述键映射项目放置操作,在一些实施方式中,键映射项目获取操作可无限地等待下去,直到定额数量的复本被成功地读取。在其它实施方式中,键映射项目获取操作可在一段时间之后暂停,其后错误指示和/或在此时可得到的最佳数据(例如,具有最近的时间戳的复制数据)可返回到请求的客户。

[0187] 如果定额数量的复本被成功地读取,则可确定取回的复本的内容是否不同(块1412)。例如,对照每个其它取回的复本可比较被请求的项目144 的每个复本的总体,或可只比较项目144的某些字段(例如,记录148的某些字段)。如果根据用于比较的标准,在取回的复本中间没有差异,则取回的数据可连同键映射项目获取操作完成的指示一起返回到请求的客户(块1414)。

[0188] 如果在复本中间存在差异,则可根据选择标准选择一个复本(块 1416)。例如,该标准可包括考虑每个复本的时间戳值,其中可选择具有最高时间戳值的复本。接着可使用

选定复本的数据来开始键映射项目放置操作(块1418)。例如,可根据图13执行放置操作,如上所述。作为放置操作的结果,可写入具有选定复本的内容的最初请求的项目144的定额数量的复本,降低了未来的获取操作将遇到复本中的差异的可能性。在放置操作之后,选定复本的数据可连同返回到键映射项目获取操作完成的指示一起返回到请求的客户(块1414)。在一些实施方式中,当完成被开始以解决差异的放置操作时,在复本中间检测到差异的情况下完成获取操作是可能发生的,而在其它实施方式中,可向请求的客户指示获取操作为完成的,而与随后的放置操作是否完成无关。

[0189] 如上所讨论的,在一些实施方式中,键映射API可支持配置成指示键映射项目144的那些键146的键映射项目列表或计数操作,该键映射项目144满足某个标准例如搜索模式。在一个实施方式中,列表或/或计数操作可实现为键映射项目获取操作的特定情况,其中对满足给定列表或计数操作的标准的项目144执行相应的键映射项目获取操作。然而,根据定额协议实际上从多个复本取回项目数据(例如,记录148)的额外开销可能对键映射项目列表或计数操作是不必要的。因此,在一些实施方式中,与定额协议有关的键映射项目获取操作的那些步骤可从键映射项目列表或计数操作省略。例如,不是识别给定项目的所有复本并为每个复本产生单独的获取操作,如在块1402-1404中的,对于列表或计数操作,单个复本(例如,程序块415)可被任意选择,且其相应的等级被导航以便识别满足列表或计数操作标准的项目144。对于满足该标准的最终的项目144,相应的键146或最终的项目144的计数可返回到请求的客户,绕过图14的与定额有关的处理部分(例如,块1410-1418)。

[0190] 在一些实施方式中,除了用于为项目144编索引的不同数据结构以外,键映射实例140还可实现高速缓存。例如,高速缓存可允许指向频繁使用的项目144的键的键映射操作绕过索引数据结构的导航,以便直接访问相应的项目144,这可提高键映射项目获取操作的效率。此外,高速缓存可帮助阻止与普通的频繁地访问的键关联的主机400变得被键映射请求业务过载。例如,在键映射高速缓存分布在主机400中间的一个实施方式中,键的副本可被高速缓存在与为该键维持索引数据结构的主机400不同的主机400上。通过键的高速缓存在主机400中间这样的分布,键处理工作量可更均匀地在主机400中分配。

[0191] 在一个实施方式中,键映射高速缓存可配置成存储键148的散列并被键148的散列而不是键本身编索引。下面结合不平衡索引数据结构的讨论更详细讨论的数据散列法可构成用于以固定长度的数据结构表示可变长度的数据例如键148的有效技术,该数据结构更容易在键映射高速缓存内管理。此外,各种散列算法可为可能最初不能均匀分布的数据(例如,具有公有的相当大部分的数据的一组键148)产生均匀分布的散列值,这可便于键映射高速缓存数据在主机400中间的均匀分布。在一些实施方式中,项目144的内容可连同相应的键148的散列值一起存储在键映射高速缓存中。在其它实施方式中,项目144的指针或其它参考信息可被存储,而不是项目144本身的内容。

[0192] 一般而言,在包括键映射高速缓存的键映射实施方式中,键映射项目放置和获取操作可在对上面提供的描述有微小修改的情况下操作。在一个实施方式中,键映射项目获取操作可首先咨询高速缓存以确定获取操作是否可根据驻留在高速缓存中的数据而被服务。获取操作在使用用于读取的定额协议继续进行之前可等待来自高速缓存的响应达固定数量的时间。如果高速缓存在定额协议读取开始之后返回值,则从高速缓冲存储器读取的

该值可被处理且相应的项目144返回,以及定额协议读取可终止。如果没有值从高速缓冲存储器返回,则从定额协议读取操作读取的项目144或指向这样的项目144的指针可连同相应的键信息一起安装在键映射高速缓存中。

[0193] 一般而言,在包括高速缓存的键映射实施方式中的键映射项目放置操作可实质上如上所述操作,除了锁定或其它一致性协议可用于阻止多个放置操作同时试图修改相同的高速缓存项目以外。在一个实施方式中,键映射项目放置操作可配置成在开始用于写入的定额协议之前试图锁定相应于键148的高速缓存项目。当接收到来自锁定请求成功的高速缓存的响应时(例如,因为不存在对项目的其它锁定,或因为在高速缓存中没有相应的项目),定额协议可继续下去。在放置操作根据定额协议结束之后,锁定可被释放,且新项目数据可安装在高速缓存中。

[0194] 注意,在一些实施方式中,用于如刚刚描述的键映射放置和获取操作的定额协议可实现用于更新键映射项目状态的强一致性模型。也就是说,定额协议可保证一旦对特定键的放置操作被客户确认为完成,则随后的获取操作将返回最近放置的数据,即使不是每个副本都在获取操作被处理的时间被更新。

[0195] 当键映射操作例如放置和删除操作指向特定的键映射实例140时,在该特定的键映射实例140内的项目144的状态可随着时间的过去而改变。因此,在没有调整它们的任何企图时,在部署内的不同键映射实例140可能随着时间的过去往往变得不同或不一致。如果只有一个存储服务客户50 参考给定的对象30,并且通过相同的键映射实例140这么做,则这样的不同可能没有实际效果。然而,如果多个存储服务客户50通过不同的键映射实例140参考相同的键,则这样的不一致可使客户50在时间的相同点观察到不同的键映射状态和/或不同版本的对象数据。

[0196] 如前所述,强一致性协议例如原子或定额协议可在更新副本时使用,以有效地阻止客户观察到副本不一致性或根本阻止这样的不一致性出现。然而,在不同副本的访问等待时间有时可相当大地变化的分布式环境中,强一致性协议可能有高性能成本。例如,对于原子或定额协议,操作完成所需的时间可相应地为相对于所有副本或定额数量的副本中最慢副本来说完成操作所需的时间的函数。而且,在没有强一致性协议时,副本不一致性变得对客户可见的概率(例如,存储服务客户50获得失效的键映射或对象数据的概率)可能通常为在被访问的副本还没有反映更新时的时间段期间客户访问副本的概率的函数。

[0197] 对于很多对象30,这个后面的概率可能很低。例如,在一些实施方式中,由存储服务系统管理的大多数对象30可通过特定的键映射实例140 被单个客户50访问,在这种情况下从客户的观点来看不一致性可能是未解决的。对于可被多个客户50访问的对象50,可观察的不一致性仍然是不可能的。例如,两个键映射实例140对于特定的键在比方说10秒的时期内不一致。然而,如果在不一致的时期期间对于特定的键没有访问被执行(例如,如果相应的对象30的访问之间的持续时间大于不一致的时期),或如果被执行的访问指向较新近更新的键映射实例140(例如,如果最后更新键的状态的客户50是通过相同的键映射实例140参考键的下一个客户),则不一致性可能对客户50没有可观察的效果。因此,在一些实施方式中,键映射实例140可使用力求将键映射实例140聚合到一致状态的不严格的同步协议,但其可允许在任何给定的时间在键映射实例140中的某种程度的不一致性。这样的同步协议可为大多数客户50提供更好的总性能,更严格的同步对这些客户50可能是不必要的。

在一些实施方式中,对共享的对象30需要键映射数据的更严格的访问同步的客户50可在他们本身中间实现附加的协议,而不需要所有的客户50承受更严格的同步的负担。例如,共享对特定的一组对象30的访问的一组客户50可使用信号 (semaphore)或其它分布式锁定技术来协调他们对键映射数据的访问。

[0198] 在一些实施方式中,在键映射实例140中间的不严格的同步协议可包括可独立地实现同步过程的不同方面的不同同步任务的组合。图15A-B示出包括两个不同的同步任务的不严格的同步协议的操作的方法的一个实施方式:更新传播任务在图15A中示出,而反熵或设定一致任务在图15B 中示出。首先参考图15A,操作在块1500中开始,其中可检测到对一个键映射实例140的更新。例如,键映射实例140可根据如上所述的定额协议来接收和完成键映射项目放置或删除操作。

[0199] 处理键映射更新的键映射实例140可接着将更新操作转发到在存储服务系统内提供的每个其它键映射实例140 (块1504)。例如,如果键映射实例140a处理键映射项目放置操作,则它可将包括变元、参数等的操作转发到键映射实例140b和140c。在一个实施方式中,可在没有验证或确认的情况下执行转发。例如,处理键映射更新操作的键映射实例可使用“发后不理 (fire and forget)”来转发操作,进行一次将操作转发到每个其它键映射实例的尝试,而不试图验证转发的操作是否在其目的地被接收或如果没有接收到则重新发送该操作。使用任何适当的转发策略可出现这样的转发,例如从发起的键映射实例140到多个键映射实例140的并行广播、从发起的键映射实例140到其它实例的顺序转发、基于树的策略,等等。

[0200] 接收转发的操作的那些相关联的主机400可在本地执行更新操作 (块 1506)。例如,如果主机400f成功地接收到从主机400a转发的键映射项目放置操作,则它可执行该操作,好像它从任何键映射客户接收操作一样。如果放置操作在主机400f上成功地完成,则键映射实例140a和140b可相对于放置操作被同步。

[0201] 一般而言,可能期望在主机400中间转发键映射更新操作在大部分时间成功。因此,最小化在转发这样的操作中包含的开销可减少在大多数情况下在键映射实例140中间获得同步所需的时间和/或带宽。例如,从转发过程消除确认响应或其它类型协议的验证或信号交换可释放通信带宽用于其它用途,例如以支持包含较大程度的同步业务的较大规模的键映射实现。在很多情况下,通过键映射部署 (其可通常相应于给定键映射项目144 的复本的可能不一致性的窗口) 传播键映射更新所需的时间可限制为将操作转发到相关联的主机400所需的通信等待时间以及主机400应用转发的操作的处理等待时间。经常地,此总时间可在秒或秒的分数的数量级。

[0202] 然而在一些实施方式中,在主机400中间键映射更新操作的转发可能失败。例如,通信链接失败可能致使一个主机400不可从另一个主机400 到达,或可使转发的操作在传输中丢失、截断或否则损坏。可选地,例如由于暂时的硬件或软件问题,目标主机400可能不能接收或正确处理准确转发的更新操作。如在一个实施方式中的,如果在发起的主机400方面不试图验证或确保转发的键映射更新操作被目标主机400成功地接收并处理,则个别操作的转发失败可能导致在键映射实例140中间相对于某些项目的不一致性。

[0203] 相应地,在一个实施方式中,在键映射实例140中间不严格的同步协议可包括上面提到的并在图15B中示出的反熵或设定调整任务。该任务可称为“反熵”任务,因为通常任务

的操作可用来在不同的键映射实例140中间减小差异并增加相似性,因而在键映射实例140中间降低可由更新传播的随机或系统故障引入的总熵以正确同步实例。在所示实施方式中,操作在块1510中开始,其中发起的键映射实例140随机选择一起执行特定分区的调整的另一键映射实例140,该特定分区如上所述可包括驻留在不同主机400上的很多复制的程序块415。

[0204] 发起的键映射实例140可接着与选定的键映射实例140交换关于在实例内的分区的信息(块1512)。例如,在两个键映射实例140的特定主机 400可配置成交换在每个实例内维持的分区索引410的拷贝,分区索引410 又可识别在每个实例内定义的那些程序块415。

[0205] 根据交换的分区信息,发起的键映射实例140可接着识别两个实例中分区之间的相应性(块1514),并可调整发起的键映射实例140内的每个分区与选定的键映射实例140内的相应分区(块1516)。例如,如前所述,给定键映射实例140内的每个分区可在很多程序块415中复制。在一个实施方式中,发起的键映射实例140可配置成指引分区内的特定程序块415(其可称为“引导程序块”)来与选定键映射实例140内的相应分区的相应或“对等”程序块415通信,以便调整分区之间的差异。在一个实施方式中,两个程序块415的调整可包含程序块交换关于包括在每个程序块415中的键映射项目144内的差异的信息并接着传播每个键映射实例140内的最当前的信息。例如,如果一个程序块415在时间戳信息的基础上确定其项目 144的版本比对等程序块415的项目更近,则它可将项目数据传送到对等程序块415。随后,对等程序块415可执行键映射项目放置操作(例如,根据如上详细描述的同额协议)以更新项目144的拷贝。

[0206] 一旦两个键映射实例140之间的分区调整已经完成,则操作就可从块 1510继续,其中调整过程相对于另一随机的键映射实例140再次开始。在不同实施方式中,每个键映射实例140可配置成以预定的或动态地确定的时间间隔执行该过程。例如,调整可以每分钟一次的静态速率或以根据随机或其它统计概率分布确定的时间间隔出现。在一些实施方式中,调整可在一定数量的键映射访问出现之后或在对某些个别的、类型的或组的键映射项目的访问被检测到之后执行。

[0207] 一般而言,图15A-B所示的更新传播和设定调整或反熵的方法可以互补的方式操作。在大多数情况下,更新传播可良好地同步部署内的不同键映射实例140。在由于更新传播故障出现键映射不一致性的那些实例中,反熵任务可通常操作来调整这样的不一致性。注意,在一些实施方式中,反熵任务的执行可能不能保证两个键映射实例140准确地全部同步。然而,在一个实施方式中,可实现反熵任务以保证其操作不增加两个键映射实例 140之间不一致性的程度。因此,通过重复的应用,反熵任务可便于键映射实例140的收敛。下面结合数据结构的特定实施方式的描述提供了关于反熵任务的一个特定实施方式的更多的细节,键映射实例140可用该数据结构实现。

[0208] 如图12所示和上面讨论的,在一些实施方式中,除了其他键映射实例 140以外,存储服务系统还可包括复制器键映射实例190。在一个实施方式中,复制器键映射实例190可本质上配置成与上述键映射实例140相同,并可使用上面讨论的协议参与键映射同步。然而,在这样的实施方式中,复制器键映射实例190可配置成为复制器180而不是协调器120或其它键映射客户服务。在一些情况下,从其它键映射实例140分离复制器键映射实例190可在总体上提高键映射性能。例如,当复制器180通过键映射重复以检查对象30的复本的状况

和数量时,复制器180可产生相当数量的键映射请求业务。如果与代表存储服务客户50的请求而产生的键映射业务混合,则复制器键映射业务可能消极地影响响应时间或与客户50有关的其它服务质量测量。相反,配置复制器180来利用专用键映射实例190 可将内部产生的键映射业务与客户产生的业务分离。此外,这样的分离可更好地使每种类型的键映射实例的实现能够根据其主要客户的要求而进行调整。例如,复制器键映射实例190的实现可配置成便于大量并行键映射操作的处理,而不是最小化任何给定键映射操作的等待时间,同时键映射实例140可被最佳化而用于服务质量标准的不同组合。然而,注意,不需要以这种方式分离键映射实例140,而在一些实施方式中,复制器180 可为键映射实例140的客户而不是专用复制器键映射实例190的客户。

[0209] 在一个实施方式中,复制器键映射实例190还可配置成便于由客户50 使用的存储服务系统资源的账目管理。特别地,复制器键映射实例190可配置成以指示相应项目的额外数据来增加由键映射实例140存储的项目 144,所述相应项目承担为相应的对象30记账或其它的财务责任。例如,在图16所示的实施方式中,示出了复制器键映射项目194。在复制器键映射实例190内,项目194可相对于键映射实例140的结构和等级与项目144 相同地起作用。然而,在所示实施方式中,项目194包括附加字段,即,存储桶ID196。一般而言,存储桶ID196可包括存储桶20的标识符的指示,该存储桶20包括相应于键146的对象30。这样的标识符可响应于来自客户50的请求,例如如上所述通过网络服务接口100或协调器120来定义,以产生其中存储对象30的存储桶20。注意,在其它实施方式中,账目管理信息不需要完全反映在复制器键映射实例190的项目内。例如,在一个实施方式中,一些或所有键映射实例140的键映射项目144可配置成将存储桶ID196的指示存储为例如记录148或键146内的附加字段。

[0210] 如上讨论的,对象30和存储桶20之间的关系对键映射实例140的一般操作可以是透明的。然而,假定这种关系一般是静态的,则通过复制器键映射项目194存储桶20和对象30的明确关联可便于客户50的账目管理和记账。例如,不是明确地向网络服务接口100询问与每个对象30相关联的存储桶20,账目管理过程(其可包括在复制器180或另一模块中,或实现为系统内的不同模块)可配置成根据存储桶ID196来分类复制器键映射项目194。当完成这样的分类时,与特定的存储桶ID 196相关联的所有键146将容易显而易见。如在记录148内指示的相应对象30的大小可接着被合计以确定与存储桶ID196相关联的总存储资源利用。此外,可考虑对象30的其它特征,例如与特定对象30相关联的存储的类。接着根据适当的记账模型可货币化资源利用。

[0211] 在不同实施方式中,代替或除了可便于各种内部系统维护或账目管理任务的存储桶ID 196之外,复制器键映射项目194还可包括其它字段。注意,在复制器键映射实例190不同于其它键映射实例140的实施方式中,这样的附加字段的存储成本可配置成复制器键映射实例190。然而,可以设想,在缺少专用复制器键映射实例190的实施方式中,可增加键映射实例140的项目144以包括这样的附加字段。

[0212] 分层的不平衡数据结构

[0213] 如前所述,在一些实施方式中,存储服务系统可调整以支持非常大数量的对象30,例如,大约十亿或更多。因此,在这样的实施方式中,每个键映射实例140管理类似数量的项目144。在一些实施方式中,键映射实例140可支持不同类型的分类和/或分组操作,例如在

前面部分中讨论的键映射项目列表和计数操作。此外,为了支持一致的键映射操作,由每个键映射实例140管理的很多键可能需要在其它键映射实例140中间同步,如上所述。

[0214] 在很多情况下,由键映射实例140提供的键映射功能可能对总存储服务系统的操作是重要的。例如,如果客户50选择不执行对于对象30的特定实例的基于定位器的访问,则键映射实例140可调解由客户50执行的每个基于键的对象访问请求。因此,如客户50看到的存储服务系统的性能可直接依赖于键映射实例140访问和处理键映射项目144的效率和速度。键映射实例140的性能又可直接依赖于用于索引和组织项目144的数据结构,例如用于在图12的实施方式中实现分区索引410、块索引420和项目索引430的数据结构。

[0215] 设计索引数据结构来支持大规模键映射实现中的分类和同步操作可能提出相当多的挑战。需要索引大量数据例如数据库的传统应用频繁地使用传统的平衡数据结构,例如B-树或其它类型的平衡树。一般而言,当用于索引给定数量的数据项例如键映射项目144时,平衡数据结构算法试图根据被管理的项的数量在平衡数据结构中分布数据项。例如,给定索引的 10,000个键映射项目144,平衡数据结构算法可试图在项目144中间选择断点,以便项目分成10组,每组大约1,000个项目。平衡的数据结构算法可在每组内产生平衡等级的其它层次,例如将大约1,000个项目的每组再分成五个子组,每个子组大约200个项目。由于数据项被添加到平衡数据结构和从平衡数据结构删除,因此数据结构内的组和/或子组可能变得不平衡。因此,传统的平衡数据结构算法可通过在组中重新分配数据项而使数据结构重新平衡,产生额外的组和/或产生额外的等级层次。这样的重新平衡在数据项被添加或删除时可“即时”发生,或可在一定数量的数据项修改发生或自从最后的重新平衡以来经过一定数量的时间之后出现。

[0216] 通过以平衡的方式分离数据项,平衡数据结构可为数据结构内的任何给定的数据项提供可预测的、大致一致的访问等待时间,这可能是在大量数据项需要被索引的大规模实现中所希望的。然而,例如使用如上所述的不严格的同步模型,有效地调整或同步分布的平衡数据结构的实例可能特别困难。特别地,当平衡数据结构的实例被单独修改时,在每个实例内将数据项分成组的断点可变得不同。作为结果,在不同的平衡数据结构实例的组或子组之间的数据项从属关系方面可能没有直接的相应性。因而为了协调两个这样的实例,彻底比较两个实例的整体可能是必要的,这在每个实例索引大量数据项的情况下可能非常耗费时间。

[0217] 作为根据数量在组中间分布数据项的平衡数据结构的可选方案,在一些实施方式中,键映射实例140的索引数据结构可配置成实现不平衡数据结构(其也可称为特里结构),该不平衡数据结构根据每个组内数据项中间的一些关系在组中间分布数据项。特别地,键映射实例140可配置成根据相应的键146的前缀索引项目144。作为例子,考虑存在具有相应的对情况不敏感的字母数字键146的600个键映射项目144的情况。这600个项目的平衡索引可将项目分成三个不平衡组,每组有200个项目。相反,在一个实施方式中,不平衡索引可定义三个字母数字组,以便以字符a到 l开始的那些项目分配到第一组,以字符m到x开始的那些项目分配到第二组,以及以字符y或z或数字0到9开始的那些项目分配到第三组。

[0218] 项目144可不均匀地分布在不平衡索引的组中。例如,在第一组中可有300个项目,在第二组中250个项目,而在第三组中只有50个项目。然而,注意,对于任何给定的项目144,在不平衡索引的特定组中的给定项目144的从属关系可为其相应的键146的函数,而不依赖

于任何特定的组内的项目144的数量。因此,如果不平衡索引的两个实例维持相同的组定义,则每个组可独立地同步,而不依赖于其它组。例如,在两个实例之间的a-1组可被同步而与m-x组和y-9组无关。相反,如上所述,相同的项目144的集合的平衡索引的两个实例的同步可能需要考虑在所有组中的所有项目。

[0219] 图17示出了说明索引很多数据项的不平衡数据结构的使用的一个例子。在所示实施方式中,不平衡索引200(或简单地,索引200)包括以等级的方式布置来索引为以前缀“a1”开始的很多字符串值的很多节点210。例如,被索引的值可相应于键映射实例140的不同项目144的键146。索引200内的每个节点210包括可以直接相应于或不直接相应于被索引的数据项的相关的标记值。在所示实施方式中,被描绘为椭圆形的节点可相应于没有相应的数据项的索引200的内部节点,而被描绘为矩形的节点可相应于被索引的数据项。因此,例如,节点210a相应于字符串“a1”并与索引200内的很多其它节点有关,但可能不存在相应于字符串“a1”的实际键146。相反,具有标记“Alicia”的节点210n可相应于指定相同字符串的键146。内部和非内部节点210之间的区别可以明确反映或可以不明确反映在节点210的状态中。

[0220] 如下所述,在一些实施方式中,不平衡数据结构可配置为其它索引的索引。在一些这样的实施方式中,在索引200的第一实例内索引的数据项可为另一索引200的根节点210,且第一索引200内的相应节点210可被考虑为非内部节点。也就是说,在一些实施方式中,给定索引200的非内部节点210可一般被定义为与数据值相关联的任何节点210,例如另一索引200的项目144或根节点,所述另一索引在给定索引200的外部。类似地,给定索引200的内部节点可只涉及给定索引200内的其它节点210,并且可不具有与项目144或不同于给定索引200的其它索引200的任何关联。还注意到,如图17所示,非内部节点210不必为叶节点(例如,不涉及在较低等级层次的其它节点的节点)。

[0221] 在不同实施方式中,每个节点210可编码各种信息。在图18中示出了说明可在节点内编码的不同数据字段的一般节点210的一个实施方式。在所示实施方式中,节点210包括标记字段212、计数字段214、指纹字段216以及一个或更多指针字段218。通常,标记212可配置成存储相应于给定节点210的值,该值可被使用在遍历或利用索引200的过程中,如下面更详细描述。在一些实施方式中,标记212可唯一地识别索引200内的所有节点中的一个节点210。此外,在一些实施方式中,给定节点210的标记212可包括作为前缀的索引200内给定节点210的所有直接的先辈的标记。也就是说,通过将某个值附加到该给定节点的直接的父节点210的标记212,可确定任何给定节点210的标记212。例如,考虑图17的具有标记“alicia”的节点210n。节点210n的直接先辈节点210l、210k和210a中的每个都具有形成节点210n的标记的适当前缀的标记(分别为“alic”“ali”和“al”)。

[0222] 如图17所示,某些节点210指向在索引210的等级中进一步在下面的一个或更多子或后继节点210。在一个实施方式中,指针字段218可配置成将数据反射指针或参考从给定节点210存储到另一节点210。例如,给定的指针字段218可包括识别地址空间例如存储器地址空间内参考的节点210的位置的地址。给定的指针字段218还可包括关于参考的节点210的额外的标记信息。例如,如图17所示,从给定节点210到后继节点210的每个弧线标注有后继节点210的标记212的第一字符,该字符不同于由给定节点210的标记212形成的前缀。在一个实施方式中,此附加的标记信息可连同指向该参考的节点210的指针一起存储在相

应的指针字段218内。例如,包括在节点210a中的指针字段218可分别包括对节点210b、210g、210j、210k和210t的参考以及相应的标记数据“a”、“e”、“f”、“i”和“z”。

[0223] 如上面关于图12讨论的,索引例如索引200可用于组织数据项例如键映射项目144用于选择。在一些实施方式中,非内部节点210(即,直接映射到被索引的数据项的节点210)的指针字段218还可包括指向相应的数据项,例如键映射项目144、块425或程序块415的指针。在一些实施方式中,如下面更详细描述,不平衡索引例如索引200可分等级地实现,以便一个索引200的非内部节点210可指向另一索引200。参考被索引的数据项的指针字段218可通过任何适当的技术与参考另一节点210的指针字段218区分开,例如通过对不同类型的指针字段218使用不同的编码。例如,如在前面的段落中描述的,在与到后继节点210的弧线关联的标记信息被在指针字段218内编码的实施方式中,无标记用于将对被索引的数据项的参考与对后继节点210的参考相区分。

[0224] 对于给定的节点210,计数字段214和指纹字段216可配置成反映在给定节点210之下的节点210的状态。在一个实施方式中,计数214可配置成存储作为给定节点210(例如,分层地在下面)的子孙的所有节点的计数。例如,图17的节点210k在索引200内在其下有八个其它节点210。相应地,它的计数214可使用任何适当的编码或格式指示8的值。

[0225] 在不同实施方式中,给定节点210的指纹字段216可配置成存储指示对分等级地在给定节点210下面的节点210的数据的某个部分执行的散列(例如,适当的散列算法的结果)的值。例如,给定节点210的指纹字段216可反映作为给定节点210的子孙的所有节点210的标记212的散列的和。可选地,指纹字段216可根据特定的、一致的遍历顺序(例如,横向优先或深度优先的遍历)来反映后继节点210的标记212的串连的散列。在其它实施方式中,除标记212外的节点210的其它字段可参与散列。在一些实施方式中,与给定节点210相关联的数据可反映在其自己的指纹字段216内,而在其它实施方式中,给定节点210的指纹字段216可根据其后继节点而被严格地确定。为了描述的一致性,如在这里使用的,给定节点210的指纹可表示作为给定节点210的至少一些后继节点的函数的散列值,同时给定节点210的散列可表示作为只与给定节点210关联而不与其子孙关联的数据的函数的散列值。

[0226] 一般而言,散列算法可配置成将可能任意长度的给定的源数据值映射到较小的一般固定长度的散列值上,以便如果两个散列值不同,则原始的源数据值也必须在某个方面不同,其中两个散列值从所述原始的源数据值产生。当散列算法一般不是一对一的函数时,两个散列值之间的一致性不必暗示原始的源数据值之间的一致性。然而,对于一些类别的散列算法,原始的源数据值与给定的完全相同的散列值之间的一致性在统计上可能处于可以计量的概率或信任度内,特别是对于表现某种冗余度的源数据值来说。不同类型的散列算法也可称为符号差、指纹或校验和算法。可以设想,可使用任何适当类型的散列算法来产生将存储在指纹字段216中的散列值,作为非限制性的例子包括任何适当版本的消息摘要5(MD5)算法或安全散列算法(SHA),例如SHA-1、SHA-256、SHA-512等。

[0227] 如在前面部分中描述的,可对键映射实例140执行的基本操作可包括放置和获取操作,其可分别存储和取回相应于被指定为操作的参数的键的项目144。在一些实施方式中,键映射实例140内的不同索引可实现为不平衡索引,例如索引200。

[0228] 在大量数据项需要被编索引的场合,如可能在键映射实例140中共用的,对所有数据项使用索引200的单个实例可能是不实际的。例如,单个大的索引可能不完全适合于处理

索引的系统的存储器,这可消极地影响依赖于索引的操作的性能。在一些实施方式中,使用分层的不平衡数据结构或分层的索引可实现大的索引。一般而言,在分层的索引中,索引200的多个实例可被分等级地定义,其中在等级中较高处的实例可索引其它索引200,以及在等级中较低处的实例可索引特定的项目144或其它项目(例如,块425或程序块415)。

[0229] 在图19中示出了分层索引的一个实施方式。在所示实施方式中,分层索引220包括五个索引200a-e。索引200a包括节点210u-x,其中每个都是涉及索引200b-e之一的相应根节点的非内部节点。索引200b-e每个又包括图17示出的各个节点210a-t。在分层索引220的一些实施方式中,较高层索引例如索引200a可配置成驻留在处理索引的系统的存储器、高速缓存或另一较高层的存储器等级中,而较低层索引例如索引200b-e可主要驻留在磁盘或这样的存储器等级的另一较低层中。在这样的实施方式中,较低层索引可例如使用分页(paging)型技术按需要从存储器等级的较层重新定位到较高层。通过支持大量数据项的索引的等级分区,分层索引220可更高效和有效地使用系统资源。

[0230] 例如,使用上述分页技术,分层的索引220的频繁使用的索引200可保持在存储器等级的较高层,这些较高层一般访问得较快但在容量上被限制,而较不频繁使用的索引200可存储在较低的存储器等级层次中,这些较低层一般访问得较慢但具有比较高层更大的存储容量。可以设想,在一些实施方式中,当节点210添加到分层索引220内的索引200时,各个索引200可增长到超过目标大小(例如,在实现索引的系统上磁盘块或存储器页面的大小)。在这样的实施方式中,如果给定索引200增长到超过目标大小,则它可分裂成两个或更多索引实例。在执行这样的分裂的过程中,节点210可按需要添加到较高层索引200中,以构成新索引实例。

[0231] 响应于键映射项目放置或获取操作,可遍历分层的或不分层的不平衡索引以确定指定的键是否相应于索引200内的节点210。在图20中示出了不平衡索引遍历方法的一个实施方式。在所示实施方式中,操作在块2000中开始,其中索引内待搜索的键值(也称为搜索值)例如通过相关的键映射操作被指定。随后,选择索引的根节点210(例如,没有父节点的节点210)(块2002)。

[0232] 对于选定的节点210,对照搜索值比较该节点的相应标记值212以确定该标记值是与搜索值确切地匹配,是搜索值的前缀,还是两者都不是(块2004)。如果选定节点210的标记值212与搜索值匹配,则检查选定的节点210以确定它是内部还是非内部节点(块2006-2008)。例如,可检查选定节点210的指针218或其它内容以确定节点是否参考由索引200索引的数据值,例如项目144或索引200的另一实例。如果选定的节点210是内部节点,则可能出现索引失败,如下所述(块2022)。

[0233] 如果选定节点210是非内部节点,则取回由选定节点210参考的数据值(块2010)。在支持分层的平衡数据结构的实施方式中,其中一些数据结构实例可索引其它数据结构实例,取回的数据值可相应于项目144或索引200的另一实例的根节点。如果取回的数据值是项目144,则索引遍历可结束,且取回的项目144可被根据发起遍历的键映射操作来处理(块2012-2014)。例如,如果发起的键映射操作是获取操作,则取回的项目144可被作为获取操作的结果返回。如果发起的键映射操作是放置操作,则取回的项目144可根据在放置操作中指定的参数来修改。

[0234] 如果取回的数据值不相应于项目144,则在所示实施方式中,它可相应于另一索引

200的根节点210。相应地该根节点210可被选择(块2012、2016),且操作可从块2004继续进行,在块2004中新选择的索引200的遍历可继续进行。因此,在一个实施方式中,图20的方法的执行可继续进行,直到相应于搜索值的节点210的存在或没有被最后确定。

[0235] 返回到块2006,如果选定节点210的标记212与搜索值不匹配,而是搜索值的前缀,则可检查选定节点210的子孙以确定是否任何子孙相应于搜索值(块2018)。如果是这样,则可选择相应的后继节点210(块2020),且操作可从块2004继续进行。在一个实施方式中,可检查选定节点210的指针218以确定与特定指针218关联的额外的标记信息在与选定节点210的标记212结合时是否也形成搜索值的前缀(或完全匹配)。例如,参考图17,节点210a的标记“al”可被确定为“alibaba”的搜索值的前缀。此外,可由相应的指针218表示的、从节点210a到节点210k的弧线与额外的标记信息“i”相关联。该标记信息在附加到节点210a的标记“al”时形成值“ali”,该值“ali”也是搜索值的前缀。因此,可选择节点210k用于进一步的遍历。

[0236] 返回到块2018,如果选定节点210的子孙不相应于搜索值,则搜索值在索引200内没有相应的项目144,这也可称为索引失败(块2022)。该索引失败可接着根据发起索引遍历的键映射操作的类型来处理(块2024)。例如,键映射项目获取操作可通过将表示失败的适当的状态指示返回到请求的客户来处理索引失败。相反,键映射项目放置操作可通过将相应于待存储的项目144的新节点210插入索引中作为选定节点210的子孙来处理索引失败。例如,新节点210可被创建,且其为待存储的项目144适当地设置不同的字段,以及指向新节点210的指针218可存储在选定节点210内。注意,如果新节点210被添加到索引200或现有的节点210被修改,则添加或修改的节点210的所有先辈节点210的计数字段214和指纹字段216可被更新以反映该变化。

[0237] 返回到块2006,如果选定节点210的标记212与搜索值不匹配且不是搜索值的前缀,则索引失败也可能出现,且处理可从块2022继续。在一些实例中,这种情况可出现在选定节点210不是索引200的根节点时。相应地,在一个实施方式中,响应于此失败情况将新节点210添加到索引200可包括创建具有标记212的新的根节点210,该标记212是搜索值和现有根节点210(在这种情况下为选定节点210)的标记212的共同前缀。(在一些实例中,新根节点210的共同前缀可为零,这可解释为用于任何值的有效前缀。)新的根节点210可接着配置为将选定节点210称为子孙。如果必要,额外的节点210可产生以相应于搜索值并配置为新的根节点210的额外子孙。

[0238] 注意,在一些实施方式中,当遍历分层的不平衡索引200时,如果选定节点210的标记212与搜索值不匹配且不是搜索值的前缀,则索引失败可能不立即出现。在一个实施方式中,如果遇到这种情况,则如果选定节点210有父亲,则选择父节点210。如果父节点210是参考另一索引200的非内部节点,则可选择参考索引200的根节点210,且处理可从块2004继续。否则,可能出现索引失败。(然而,注意,这种情况可能不出现在没有索引其它索引200的不分层的自含式索引200中。)作为这种情况的例子,考虑图19的分层索引,其中搜索值为“alice”。索引200a的遍历可继续进行到具有标记“ali”的节点210w。因为节点210w具有包含相关联的标记信息“c”的指向子节点210x的指针,因此可选择节点210x。然而,节点210x的标记是“alicia”,该标记与搜索值不匹配并且不是搜索值的前缀。因此,遍历可返回到节点210w(节点210x的父亲),其可为参考索引200c的非内部节点。相应地,遍历可继续到节

点210k并最终到节点210m,节点210m具有与搜索值匹配的标记212。

[0239] 在不同实施方式中,不平衡索引200或分层的不平衡索引220可用于索引键映射实例140内的键映射项目144。例如,分层的索引220可用于实现一个或更多分区索引410、块索引420或项目索引430,或可在键映射实例140内实现的任何其它层的索引。如上讨论的,当使用不严格的同步协议时,不同的键映射实例140可在操作的一般过程中不同或不一致。在一些实施方式中,键映射实例140可使用穷举协议来被同步,该协议以一致的顺序(例如,深度优先或横向优先的搜索顺序)遍历相应的索引数据结构中的每个节点以识别在索引结构或被索引的内容中的差异。然而,上述不平衡索引的各种特征,例如根据键信息而不是键的数量的数据的分布以及在索引数据结构内包含计数和/或累积散列信息可便于计算上更有效的同步算法的实现。

[0240] 前面描述的反熵设定调整协议的很多可能的版本可设想成用在由键映射实例140实现的不平衡的、可能分层的索引上。接着是这样的协议的一个实施方式的描述,虽然应理解,对一般协议所设想的变形可展示不同的实现优先级,例如在选择最佳化某些情况而优先于其它情况或使用一种或另一特定类型或类的算法来执行协议的一般步骤中。因此,意图是所述实施方式被视为说明性的而不是限制性的。

[0241] 在一个实施方式中,配置成调整不平衡索引200或分层的不平衡索引 220的不同实例的反熵协议可包括在不同类型的消息的实例之间的交换。反熵协议的一个实施方式可基于的示例性的一组消息可包括DATA消息、REQUEST消息、HASH消息、FILTER消息和FINGERPRINT消息。下面描述了这些消息中的每个的相应的实施方式的一般功能,接着是消息可如何用于实现反熵协议的实施方式的讨论。在下列讨论中,可参考键映射实例140中间数据的交换,虽然应理解,这样的键映射实例可实现包括上述特征中的任何一个的不平衡索引200或分层的不平衡索引220的一个或更多实例。

[0242] DATA消息可用于将关于一个或更多索引节点210的数据从一个键映射实例140传送到另一个。在一个实施方式中,DATA消息可配置成只传送与给定节点210关联的标记212,而在其它实施方式中,DATA消息可传送与给定节点210关联的其它字段。在一些实施方式中,如果给定节点 210是非内部节点,则DATA消息还可包括与给定节点210关联的数据项的所有或一些部分(例如,项目144或关于另一索引200的根节点210的信息)。

[0243] HASH消息可用于将关于一个或更多索引节点210的信息从一个键映射实例140传送到另一个,而没有明确传送给给定节点210的字段或与给定节点210关联的数据项。在一个实施方式中,HASH消息可配置成传送与给定节点210关联的标记212以及根据适当的散列算法计算的给定节点 210的散列。在一些实施方式中,给定节点210的散列还可反映与给定节点210关联的数据项(例如,键映射项目144),但可排除给定节点210的任何子孙。

[0244] REQUEST消息可用于传送对与一个或更多节点210关联的消息的请求。在一个实施方式中,REQUEST消息可配置成传送一个或更多标记前缀值。在响应中,请求的实例可能期望接收关于具有标记212的那些节点 210的信息,对该标记212来说,传送的标记前缀值实际上是前缀。对于给定的节点210,接收的信息可包括给定节点210的相应字段的内容和/或相应于给定节点210的数据项(例如,键映射项目144)。在一些实施方式中,REQUEST消息可支持被请求的标记前缀值的进一步的限定,例如通过指定由特定的标记前缀值界定的结果空间内的值或值的范围应从为该标记前缀值返回的结果中排除。例如,REQUEST消息可指定

关于与标记前缀值“alex”匹配的所有节点210的信息应被返回,除了与前缀“alexe”或“alexj”匹配那些节点210之外。

[0245] 刚刚描述的消息可通常在各个节点210的粒度层(level of granularity) 操作。然而,如果键映射实例140之间的差异通常很小(例如,限制到大多数节点210) 则它可促进同步过程以立刻快速确定多个节点210的状态。在一个实施方式中,FINGERPRINT和FILTER消息可配置成传递关于节点210的集合的消息。特别地,在一个实施方式中,FINGERPRINT消息可配置成将节点210的指纹字段216连同其标记212一起从一个键映射实例140传送到另一个。如上所述,给定节点210的指纹字段216可配置成存储被确定为给定节点210的子孙的函数的散列值。因此,如果在不同键映射实例140中相应节点210的指纹字段216相等,则相应节点210的子孙的布置和内容相同可能是非常可能的(取决于所使用的散列算法的特征)。也就是说,从相应的节点210传下来的键映射实例140的部分被同步是非常可能的。

[0246] 指纹的使用可允许关于包括相当数量的节点210的键映射实例140的部分是否被同步的快速确定。然而,指示相应的部分没有同步的指纹通常可能不提供关于这些部分如何不同的进一步的细节。在一个实施方式中,FILTER消息可配置成将编码相应于特定前缀值的很多节点210的过滤值从第一键映射实例140传送到第二键映射实例140。第二实例可接着使用接收到的过滤值来检测其自己的相应于该前缀值的节点210,以确定第二实例的哪些节点不存在于第一实例中,如果有的话。

[0247] 在一个实施方式中,由FILTER消息传送的过滤值可为Bloom过滤器,虽然可以设想,可使用用于可恢复地将一组数据值编码成过滤值的任何适当的过滤技术。一般而言,一组值(例如,节点210)的Bloom过滤器可相应于M位二进制值,其中M为整数。在任何值编码到Bloom过滤器中之前,它的初始值可为零。也就是说,过滤器的所有位可处于复位无效的状态中。可通过借助于一组k个独立的散列函数中的每个来传递将在过滤器内被编码的每个值可填充(populate)Bloom过滤器,其中每个函数将待编码的值映射到范围[0,M-1]内的值上。对于k个最终的散列值中的每个,Bloom过滤器内的相应位被置位有效(例如,设定为逻辑1值)。根据将在Bloom过滤器内编码的值的数量和类型以及期望的假阳性(false positives) 的概率,M和k可选择为设计参数(下面讨论)。例如,在使用八个散列函数的1,024位Bloom过滤器中,每个散列函数可产生将过滤器的1,024 位的特定一位指定为置位有效的相应的10位散列值。

[0248] 为了测试给定的值是否被编码到Bloom过滤器中,该值被传递通过用于编码过滤器的同一组k个独立的散列函数,且最终的过滤值的k个位被检查。如果最终的过滤值的k个位中的任何一个不是被置位有效的,则测试值确定地不能在过滤器内编码。也就是说,测试值可能最初在过滤器内被编码,或它可为假阳性的。在一些实施方式中,在产生Bloom过滤器的每个独立的时机,可用随机或自动产生的盐值(salt value) 或种子值(例如,作为当前系统时间的函数) 确定散列函数的参数,以减小当给定的一组值被相继地编码到过滤器中时将产生相同的假阳性值的概率。

[0249] 因此,例如,第一键映射实例140可将相应于前缀P的一组节点{A, B,C,D,E} 编码到Bloom过滤器中,并可使用FILTER消息将过滤器传送到第二键映射实例140。在第二键映射实例140中,一组节点{A,B,X, Y,Z} 可相应于前缀P。第二键映射实例140可对照过滤器测试每个节点,并可确定节点A、B和X可在过滤器中编码,而节点Y和Z确定地不能在过滤器中

编码。因此,第二键映射实例140可正确地断定节点Y和Z不存在于第一键映射实例140中,并可断定节点A、B和X可能存在于第一键映射实例140中,其中X是假阳性。作为结果,第二键映射实例140可采取行动来将关于节点Y和Z的信息传送到第一键映射实例140。

[0250] 可以设想,DATA、HASH、REQUEST、FINGERPRINT和FILTER消息可根据任何适当的协议或API来实现和传送,并可包括各种类型的字段或参数,这些字段或参数配置成传送上述信息以及解码并正确地处理消息所必需的任何额外的信息。在一个实施方式中,消息可包括额外的参数,所述额外的参数指示对于包括在该消息中的给定标记值来说,发送键映射实例是具有相应的数据还是需要相应的数据,其分别称为获取数据和需要数据参数。例如,如果键映射实例140为具有标记“a1”和某个数量的子孙的节点210发送FINGERPRINT消息,则实例可包括指示该实例在由“a1”界定的前缀空间内有一些节点210的获取数据参数。实例还可包括需要数据参数,例如如果它的由“a1”界定的前缀空间的拷贝被认为是不完整的情况下。在一些实施方式中,获取数据参数可能在DATA和HASH消息中是隐含的,而需要数据参数可能在FILTER和REQUEST消息中是隐含的,虽然DATA 或HASH消息可明确指定需要数据参数,而FILTER或REQUEST消息可明确指定获取数据参数。在一个实施方式中,可能需要FILTER消息来指定需要数据或获取数据参数中的至少一个。

[0251] 在一个实施方式中,由两个键映射实例140管理的反熵协议可在两个实例建立彼此的联系时开始。每个实例可假定它具有某些数据和缺少某些数据。相应地,每个实例可向其它实例发送FINGERPRINT消息,其指定实例的根节点210的标记212和指纹216并包括获取数据和需要数据参数。例如,在使用分层的不平衡索引220的键映射实例140的实施方式中,根节点210可相应于在索引200内没有父节点的节点210,该索引200没有父索引或上级索引200。

[0252] 在图21中示出了处理FINGERPRINT消息的方法的一个实施方式。在所示实施方式中,操作在块2100中开始,其中FINGERPRINT消息从消息发送器接收。例如,第一键映射实例140可将包括标记值、指纹和一个或更多获取数据或需要数据参数的FINGERPRINT消息传送到第二键映射实例140。在收到FINGERPRINT消息之后,接着遍历消息接收器的索引以识别这种节点210是否存在,即对节点210来说接收的标记值是相应的标记字段212的前缀(或确切地匹配)(块2102)。例如,使用图20的方法或其适当的变形从根节点210开始可遍历键映射实例140的索引。

[0253] 如果接收的标记值不是任何节点210的标记字段212的前缀或确切匹配,则相应于由FINGERPRINT消息参考的节点的节点210可能不存在于消息接收器。相应地,通过将REQUEST消息传送到消息发送器,指定包括在最初接收的FINGERPRINT消息内的标记值,可响应接收器(块2104)。在一个实施方式中,REQUEST消息的处理可继续进行,如下面更详细描述。在一些实施方式中,只有当接收的REQUEST消息指示获取数据参数时才可传送REQUEST消息。

[0254] 注意,在一些实施方式中,在反熵协议的操作期间交换的各个消息的完成可能不依赖于额外的消息是否响应于给定的消息成功地完成而产生的。也就是说,在一些实施方式中,各个消息的处理可相对于其它消息以无状态和异步的方式出现。在这里描述的示例性实施方式的讨论中,采用此无状态的异步模型。因此,在REQUEST消息产生之后,FINGERPRINT 消息本身的处理可被考虑为完成的(块2106)。然而,该模型对反熵协议的一

般操作不是必须的,且可以设想,在可选的实施方式中,任何给定的消息可阻止、等待或另外地维持与从属地或响应于给定消息而产生的消息的同步。例如,在一些实施方式中可使用显式信号交换、确认、重试或其它类型的协议,以将一个消息的完成的状态传送到另一相关的消息。

[0255] 如果接收的标记值确实相应于在消息接收器的特定节点210的标记 212的前缀或匹配,则可对照特定节点210的指纹字段216比较接收的指纹值以确定两个指纹是否匹配(块2108)。如果是这样,则消息发送器和消息接收器相对于接收的标记值同步是非常可能的(例如,服从使用的指纹算法产生冲突的或具有相同的值的两个指纹的概率,尽管指纹是从不同的数据产生)。例如,使接收的标记值作为前缀的任何节点210在 FINGERPRINT消息被发送的键映射实例140和该消息被接收的键映射实例140内处于相同的状态是非常可能的。因此,没有额外的消息可响应于 FINGERPRINT消息而产生,且消息可被考虑为完成的(块2106)。

[0256] 如果指纹不匹配,则消息发送器和消息接收器相对于接收的标记值不同步,可能需要额外的工作来使发送器和接收器一起在状态上较接近。如上所述,FILTER消息在允许发送器将关于某些节点210的特定信息传递到接收器方面可能是有用的。然而,在一些实施方式中,可编码到FILTER 消息中同时保持合理的假阳性率的节点210的数量可被限制到某个阈值。如果后继节点210的数量超过在消息接收器与接收的标记值匹配的节点 210的此阈值,则在发送FILTER消息之前执行额外的FINGERPRINT消息处理可能是更有效的。

[0257] 因此,在所示实施方式中,如果指纹不匹配,则可检查在消息接收器的特定节点210的计数字段以确定它是否超过用于FILTER消息处理的阈值(块2110)。如果是这样,则消息接收器可配置成根据特定节点210的子节点细分其相应于接收的标记值的索引范围的部分,其中对于所述特定节点210,接收的标记值是前缀(块2112)。对于每个子节点210,消息接收器可配置成将相应的FINGERPRINT消息发送回原始的消息发送器,指定相应的子节点210的标记212和指纹字段216(块2114)。此外,如果在相应于接收的标记值的索引范围的部分内有间隔,例如由特定节点210的子节点所示的,则消息接收器可配置成为为相应于该间隔的标记值发送一个或更多REQUEST消息(块2116)。接收的FINGERPRINT消息的处理可接着被考虑为完成的(块2118)。在一个实施方式中,除了上面的行动以外,如果接收的标记前缀值是特定节点210的准确匹配,则相应于特定节点210的HASH消息可返回到消息发送器。

[0258] 例如,如图17所示,消息接收器的索引200的特定节点210a可具有标记“a1”以及具有相应的标记“alan”“alex”“alfred”、“ali”和“alz”的子节点。这暗示着消息接收器具有关于以“alan”和“alex”开始的节点210的但是不关于可能以“alb”“alc”或“ald”开始的节点210的一些信息。相应地,消息接收器可为节点210a的每个子节点传送FINGERPRINT消息,以及用于这些子节点的标记中间的间隔的REQUEST消息。在支持负的REQUEST语法的实施方式中,消息接收器可传送用于除了相应于特定节点的子节点的那些标记以外的标记传送REQUEST消息。例如,消息接收器可传送用于除了以“alan”、“alex”、“alfred”、“ali”和“alz”为前缀的那些标记以外的标记传送 REQUEST消息。

[0259] 如果特定节点210的计数值不超过用于FILTER消息处理的阈值,则如果接收的FINGERPRINT消息包括获取数据参数,则消息发送器可具有关于不存在于消息接收器中的节点210的特定信息。相应地,消息接收器可配置成发送FILTER消息,其中FILTER消息将作

为特定节点210的子孙的每个节点210编码到过滤器(例如,如上所述的Bloom过滤器)中(块2120-2122)。例如,参考图17,如果特定的节点相应于节点2101,则通过 FILTER消息可产生并返回编码每个节点210m-q的Bloom过滤器。在所示实施方式中,如果获取数据参数不包括在原始的FINGERPRINT消息中,则代替FILTER消息,对于特定节点210的每个子节点,相应的FINGERPRINT消息可产生并返回到消息发送器。这些FINGERPRINT消息可包括获取数据参数。在这种情况下,在FILTER或FINGERPRINT消息产生之后,可完成接收的FINGERPRINT消息的处理(块2118)。

[0260] 在图22中示出了处理FILTER消息的方法的一个实施方式。在所示实施方式中操作在块2200开始其中例如响应于如上所述的FINGERPRINT 消息,包括标记值和过滤值的FILTER消息被从消息发送器接收。一旦接收到FILTER消息,就遍历消息接收器的索引以识别相应于接收的标记值的特定节点210(例如,对该特定节点来说接收的标记值是前缀或匹配),以类似于上面关于图21描述的方式(块2202)。在一些实施方式中,如果 FILTER消息响应于另一消息而产生,则通常存在相应于接收的标记值的节点210。

[0261] 消息接收器可接着对照FILTER消息中提供的过滤值检测特定节点 210的每个子孙,以识别不以过滤值编码的那些节点210,如果有的话(块 2204)。对于在消息接收器的没有以过滤值编码的每个节点210,相应的 DATA消息可返回到消息发送器(块2206)。FILTER消息的处理可接着被考虑为完成的(块2208)。如上所述,根据用于FILTER消息的过滤算法的类型和配置,可能出现假阳性。也就是说,消息接收器可能错误地断定其节点210中的某些被以过滤值编码,因而在消息发送器以相同的状态出现,然而实际上它们并不是这样。因此,反熵协议的单次循环不能使得两个键映射实例140相对于每个节点210变得同步是可能的。然而,可能期望,在很多实施方式中,反熵协议的单次循环不能使实例变得更加不同,且可期望协议在某个数量的循环内的重复应用以某种程度的概率收敛,取决于实例不同的程度和所使用的过滤算法的特征(例如,给定用于过滤器编码的阈值的假阳性的概率)。

[0262] 在一些实施方式中,HASH、REQUEST和DATA消息的处理可能比 FILTER和FINGERPRINT消息相当简单。在一个实施方式中,HASH消息接收器可试图识别相应于包括在消息中的标记值的节点210,并可接着计算所识别的节点210的相应的散列值。如果接收的散列值与计算的散列值匹配,则识别的节点210可能已经与消息发送器中相应的节点210同步。否则,包括接收的标记值的REQUEST消息返回到消息发送器以获得更新的数据版本。

[0263] 在一个实施方式中,REQUEST消息的处理可例如使用上述不平衡的索引导航技术,来简单地包括识别每个节点210的消息接收器,对于该节点210,包括在消息中的接收的标记值与相应的标记字段212匹配或是其前缀。对于每个识别的节点210,如上所述配置的相应DATA消息返回到消息发送器。在一个实施方式中,接收的DATA消息的处理可包括识别相应于在消息内指示的标记值的节点210是否存在于消息接收器中。如果不存在,则相应的节点210可被产生并以从消息提取的数据填充(populate)。如果存在,则与现有的节点210关联的数据和/或其相应的数据值可用从消息中提取的数据代替。在一些实施方式中,只有当接收的数据更新时,才可代替现有节点210的数据。例如,DATA消息可包括相应于在消息发送器的节点210的项目144的内容,且项目144可包括可与消息接收器中相应的时间戳信息比较的时间戳信息,以确定接收的项目144是否比现有的项目144更新。如果是这样,则接收的项目144可代替现有的项目144。

[0264] 图21的一般同步协议的变形是可能的和预期的。例如,在使用具有固定长度的分组(packet)来执行键映射实例之间的通信的实施方式中,通过在单个分组内传送用于多个节点210的多个FINGERPRINT消息而不是相应于特定节点210的单个FINGERPRINT消息可提高带宽利用。接收这样的分组的实例可接着能够快速辨别其索引200中的哪个特定索引与发送器失配,而不需要与发送器交换进一步的消息。例如,如果第一个 FINGERPRINT消息不匹配,则接收器可在给分组发送器发布REQUEST、FILTER或其它消息之前考虑分组内的其它FINGERPRINT消息。在这么做时,接收器可以能够将差异缩小到数据结构的特定部分,这可减少交换关于已经被同步的数据结构的其它部分的消息的不必要的网络业务。

[0265] 通常,可以设想,上面描述的用于使用反熵协议和/或更新传播协议来执行键映射实例调整的任何方法或技术,可通过配置成键映射实例140的层上或在实例内的各个主机400操作的键映射协调器过程来实现。注意,用于实现用于不平衡数据结构的反熵协议的上述方法和技术很多变形是可能的和预期的,且上面的讨论旨在是说明性的而不是限制性的。例如,一些项目频繁地与其它、随机选择的项目通信以便在整个网络中分布信息所使用的一般类型的协议可称为基于闲聊的协议,且可使用基于闲聊的协议的其它技术或方面来用在键映射实例140中间的反熵协议中。在不同实施方式中,上述示例性同步消息(或其它适当的消息)可以不同的方法合并以产生具有不同特征的同步协议。

[0266] 此外,虽然上面关于图17-22描述的分层的索引数据结构和同步技术是在实现用在键映射实例140内的有效数据结构的环境中讨论的,但是可以设想,这样的数据结构和同步技术可用在为快速访问而被索引的大量数据的任何应用中。这样的应用不必包括例如如图2的系统的对象存储系统,但可包括数据库系统、搜索系统或可应用数据索引的任何其它应用。

[0267] 注意,在不同实施方式中,这里描述的事件的任何类型的随机产生或选择的实现可对随机数或事件产生使用任何适当的算法或技术。在很多情况下,用于实现随机方法的计算技术可不产生完全随机的结果,而更确切地为伪随机结果。例如,伪随机算法可指定配置成统计地产生随机结果的确定性过程。如这里使用的,“随机”或“实质上随机”的数据的产生旨在包括任何适当的伪随机计算技术以及完全随机的数据源。

[0268] 存储服务组件检测和管理

[0269] 在存储服务系统的大规模高度分布的实现中,可能有分布在整个系统中的图2中所示的大量不同的系统组件。例如,可能有成百或成千的位存储节点160的实例、协调器120和键映射实例140。管理这样规模的分布式系统的状态提出了实际的挑战。例如,由于计划的维护、被组件依赖的计算资源的失效、隔离另外功能的组件的通信故障或其它原因,特定系统组件的不同实例可能在任何给定的时间不能使用。此外,在任意或不可预测的时间在一些情况下,部件的新的或以前不能使用的实例可恢复使用。

[0270] 在一个实施方式中,发现、故障和检测监控进程(DFDD) 110的实例可配置成相应地监控存储服务系统的不同相关联组件的状态,以关于这样的状态彼此通信,并用接口提供DFDD客户应用程序,这样的客户可通过该接口识别可用于执行系统操作例如键映射或位存储操作的可利用的系统组件。通常,DFDD 110可配置成代表其它部件提供存储服务系统组件的当前状态的统一可访问的视图。也就是说,不是以多个不同的接口配置存储服务系统的不同部组,其中所述接口配置与其它、不同组件直接传输状态信息,而是提供或依赖于这

样的信息的每个组件可配置成通过标准 DFDD接口来与DFDD的110的实例通信。在一些实施方式中,DFDD 110 可实现为配置成在由操作系统管理的环境内操作的监控进程。然而,在其它实施方式中,DFDD 110可实现为配置成实现这里描述的功能的独立或自主的硬件或软件,而没有依赖于或从属于操作系统或其它组件的任何必要。

[0271] 一般而言,配置成被DFDD 110的实例发现和监控的存储服务系统组件的每个实例可称为应用程序实例。例如,给定位存储节点160的操作状态或状况可被配置成由给定位存储节点160执行的SNM控制器161的实例指示。因此,SNM控制器161可相应于位存储应用程序实例。类似地,键映射实例140的操作状态可被配置成执行键映射实例内的一个或更多主机400的键映射管理器的实例指示。每个键映射管理器实例可相应于键映射应用程序实例。其它类型的应用程序实例是可能的和预期的。例如,在一个实施方式中,每个计算机系统可包括配置成检测和报告专用系统的操作状态细节,例如处理器、存储器、磁盘、输入/输出(I/O)或其它系统资源的利用的主机监控器应用程序实例,其中一个或更多存储服务组件通过所述计算机系统来部署。在一些实施方式中,DFDD 110的每个实例本身可配置为应用程序实例。也就是说,DFDD实例可配置成监控除了其它应用程序实例的状态外的其自己的操作状态。

[0272] 在存储服务系统内,应用程序实例可一般由应用程序名称识别并唯一地由相应的应用程序实例标识符(ID)识别。例如,特定的应用程序名称可包括识别应用程序实例的一般类型的字符串,例如“键映射管理器”、“位存储管理器”、“主机管理器”或另一适当的名称,同时应用程序实例ID可包括唯一地识别应用程序命名空间内的特定实例的字符串。在一些实施方式中,应用程序实例ID可明确地包括应用程序名称,例如“键映射-管理器 - 4AB8D945”。也可使用应用程序实例ID的其它适当的格式。在一个实施方式中,DFDD 110的给定实例可配置成将多个应用程序实例与相应的状态信息相关联(例如,通过名称和实例ID)。例如,在图23所示的实施方式中,DFDD 110包括多个项目111,其中每个都可将应用程序名称112和实例ID 113与实例状态信息114相关联。在一些实施方式中,DFDD 110 可使用一个或更多表格来反映不同类型的状态信息114与给定的应用程序名称112和实例ID113的关联,而在其它实施方式中,DFDD 110可使用树、如上所述的不平衡索引、或指示在给定应用程序实例和其相应的状态信息之间的关联的任何其它适当类型的数据结构。

[0273] 注意,在一些实施方式中,应用程序实例ID可包括其自己的任意级别的粒度层的命名空间。例如,在一个实施方式中,给定的键映射应用成实例ID可具有形式<mapname>/<instance>/<endpoint>。术语<mapname>可识别键项目关联的特定的键映射字典,该键项目关联可通常相应于给定的键映射部署。(不同键映射部署的键映射应用程序实例在DFDD 110的相同实例内被管理是可能的。)术语<instance>可例如通过唯一的字符串来识别键映射示例140内的指定的主机400。术语<endpoint>可识别在所识别的主机400上操作的多个独立的、在功能上相同的键映射应用程序实例之一(例如,作为不同的过程)。在应用程序实例ID内的其它复杂的命名空间是可能的和预期的。

[0274] 通过DFDD 110而与应用程序实例关联的状态信息114可包括各种不同类型的信息。在一个实施方式中,DFDD 110可配置成将对DFDD 110 管理的所有类型的应用程序实例来说可共有的全局操作状态信息存储在状态信息114内。例如,如下面更详细描述,在一些实施方式中,DFDD 110可实现全局操作状态机,其定义应用程序实例的一组全局操作状

态(或简单地,全局状态)以及在该组状态中间可能的转换。在这样的实施方式中,由DFDD 110管理的每个应用程序实例可在任何给定的时间与该组全局状态中特定的一个关联,且给定应用程序实例的全局状态可根据状态机器和应用程序实例的行为随着时间而变化。

[0275] 在一些实施方式中,除了对大不相同类型的应用程序实例可共有的全局状态信息以外,状态信息114也可反映对特定的应用程序实例或特定类型的实例可为特定的或定制的操作状态信息。例如,如果应用程序实例相应于特定的位存储节点160的位存储管理器,则其状态信息114可包括关于在该特定节点上可利用的存储资源的数量、那些资源的类型的信息(例如,高性能、低性能等),或可能对位存储节点的环境特定的任何其它相关的状态信息。类似地,对于相应于特定键映射实例140的键映射管理器的应用程序实例,其状态信息可包括关于由特定键映射实例管理的项目 144的数量、所使用的或可利用的键映射存储资源的信息,或其它相关的键映射状态信息。在一些实施方式中,选择何种应用程序实例特定的状态信息来包括在相应的DFDD项目111内可根据DFDD客户的要求确定。例如,在帮助协调器120或节点拣取器130从几个选项中选择特定的位存储或键映射应用程序实例中,可能有用的状态信息可包括在那些应用程序实例的DFDD项目111内。

[0276] 在一些实施方式中,应用程序实例的状态信息114还可包括关于 DFDD客户可如何访问实例的信息。例如,状态信息114可包括互联网协议(IP)地址和端口号,DFDD客户可通过该互联网协议(IP)地址和端口号可建立与应用程序实例的连接。一些应用程序实例可支持其它类型的接口,例如网络服务接口、公布/预订接口或其它适当的接口。在这样的实施方式中,状态信息114可包括URL或DFDD客户执行网络服务请求所必需的其它信息,以预订公布/预订通道,或执行建立与应用程序实例的通信所必需的另一种类型的行动。在一些实施方式中,除了或代替应用程序实例访问信息,状态信息114可包括关于实例物理地定位在存储服务系统中的位置的信息。例如,状态信息114可包括与特定的应用程序实例相应的数据中心300或区域310的标识符。

[0277] 如上所述,在一些实施方式中,DFDD 100可为各个应用程序实例维持全局状态信息,该信息可一般指示给定的应用程序实例是否正常地操作,并因而可利用,或处于异常状态中。在一个实施方式中,配置成通过 DFDD 110的实例监控的每个应用程序实例可配置成通常(但不是必需的)以定期的间隔例如某个数量的秒或分钟向DFDD 110报告其状态。这样的报告可称为“心跳”。心跳报告可根据任何适当的协议而被传输(例如,作为 TCP/IP消息、作为网络服务请求、或根据其它标准的或专用的发送消息协议),并可在信息内容中变化。作为最低限度的例子,给定应用程序实例可向仅仅包括应用程序名称和相应于给定实例的应用程序实例ID的 DFDD 110提交心跳。在其它情况下,给定的应用程序实例可包括心跳中的额外状态,例如本地资源利用的特定状态。在一些实施方式中,应用程序实例可配置成执行某个级别的自我诊断或自我验证以在发送心跳之前确定其自己的功能状态,而在其它实施方式中,应用程序实例可发送心跳而不依赖于任何自我评估。

[0278] 一般而言,如预期的,如果应用程序实例向DFDD 110发送心跳,则有正常操作的合理预期。如果心跳应被中断某个长度的时间,则有应用程序实例出现问题的合理预期。图24示出全局状态机的一个实施方式,对于作为心跳活动和/或其它参数的函数的每个应用程序实例,该全局状态机可由DFDD 110维持。在所示实施方式中,新的应用程序实例以NEW状态在线出现在例如,在它开始操作和向DFDD 110的实例通知其存在并提供应用程序名称、

应用程序实例ID和DFDD 110初始化相应的项目111 所必需的任何其它信息之后不久。一旦新的应用程序实例是稳定的并准备开始正常操作,它就进入OK状态。在不同实施方式中,从NEW到OK 状态的转换可为时间(例如,基于应用程序实例的类型的默认设置时间)、应用程序实例自报告、管理员干涉或这些或其它因素的组合的函数。

[0279] 在所示实施方式中,应用程序实例可保持在OK状态中,只要自从实例到DFDD 110的最后心跳以来经过的时间小于故障阈值 T_{fail} 。例如,DFDD 110可为每个应用程序实例维持当每次心跳从相应的实例接收时递增的计数器,并可监控每个计数器(例如,用倒数的计时器)以确定其值在经过 T_{fail} 之前是否改变。在一些实施方式中,除了OK (以及可能NEW) 以外的全局状态可一般称为异常操作状态或故障状态,虽然在这样的状态中间可能有区别,如下所述。

[0280] 如果自从用于应用程序实例的最后心跳以来经过了时间 T_{fail} ,则其全局状态可转换为INCOMMUNICADO。在所示实施方式中, INCOMMUNICADO可用作应用程序实例可能有问题的瞬变状态指示,但它没有被最后地确定为永久失效。例如,应用程序实例可暂时停止或挂起,到DFDD的110的心跳消息可被延时或丢失,或如下面更详细描述,相对于应用程序实例的当前状态,DFDD 110的一个实例可能与DFDD 110 的另一实例不同步。如果在INCOMMUNICADO状态中心跳被从应用程序实例接收,则该实例可转换回OK状态。在一些实施方式中,DFDD客户可自担风险地选择使用处于INCOMMUNICADO状态的应用程序实例。

[0281] 如果应用程序实例不能自动地从INCOMMUNICADO状态恢复,则可能有影响实例的更严重的问题。在所示实施方式中,可能出现两个可能的故障情况。如FAIL状态所示的,单独的应用程序实例可能例如由于基本计算资源不能支持(host) 该单独的实例而失效处于隔离中。可选地,应用程序实例可能由于失去实例和DFDD 110之间网络通信而失效,如NETWORK SPLIT状态所示的。例如,由于使存储服务系统的部分彼此隔离的通信故障,应用程序实例对DFDD 110的一些实例可能是可操作和可访问的,但是对于DFDD 110的其它实例不是可操作和可访问的。

[0282] 毫无疑问地确定给定的应用程序实例故障是否被隔离或由于网络分离可能很难。在一些实施方式中,DFDD 110可使用相应的启发式标准 H_{fail} 和 $H_{netsplit}$,其考虑不同类型的可用信息来进行关于应用程序实例是否应从 INCOMMUNICADO状态转换为FAIL状态或NETWORK SPLIT状态的确定。例如,该标准可要求给定应用程序实例在转换到另一故障状态之前处于INCOMMUNICADO状态达到至少阈值数量的时间 $T_{heuristic}$ 。此外,该标准可考虑其它应用程序实例是否也处于INCOMMUNICADO、FAIL或 NETWORK SPLIT状态中,所述其它应用程序实例与和给定应用程序实例相同的区域310或字符300共享资源或属于区域310或字符300。例如,如果位于与给定应用程序实例相同的IP地址或位于相同的区域310或数据中心300内的其它地址的其它应用程序实例是OK,则可能给定应用程序示例的故障是隔离的。相反,如果多个应用程序示例不是OK,则网络分离情况可能是更可能的,特别是如果应用程序实例状态被根据地理或网络拓扑分组。在一些实施方式中,DFDD 110可配置成除了使用被动地接收的状态信息以便确定故障的性质以外还询问被怀疑发生故障的应用程序实例。在一些实施方式中,启发式标准可配置成根据概率的一些阈值(例如,大于50%的概率、大于90%的概率,等等)来确定应用程序实例在概率上是否可能失效。

[0283] 根据启发式标准,失效的应用程序实例可转换到FAIL状态或 NETWORK SPLIT状

态。在一个实施方式中,如果心跳被接收到,则实例可从这些状态中的任一种转换回到OK状态,而在其它实施方式中,这些状态中的任一个或两个可为不能恢复的。虽然处于INCOMMUNICADO状态中的应用程序实例可被假定为具有故障概率的功能性的或可恢复的,但是处于FAIL或NETWORK SPLIT状态中的应用程序实例可被假定为已经失效了(在一些实施方式中有恢复的可能性)。通常,DFDD客户可避免选择处于这些失效状态的任一个中的那些应用程序实例。在一些实施方式中,DFDD 110可配置成对客户隐藏关于处于这些失效状态的任一个中的应用程序实例的信息。

[0284] 在所示实施方式中,应用程序实例可在转到FORGOTTEN状态之前保持在FAIL或NETWORK SPLIT状态中达到相应的时间段 T_{clean} 和 $T_{recover}$ 。例如,在FAIL状态的一些情况下,与失效的应用程序实例相关联的资源可为了恢复或分析的目的而保存一段时间。如果可能,这样的资源(例如,位存储节点160的存储资源)可接着被初始化,用于重新部署为新的应用程序实例。在NETWORK SPLIT状态的一些情况下,可能需要做出关于是否在没有失效的应用程序实例的情况下继续进行系统操作的决定,且如果这么做,应采取什么类别的恢复行动(例如,在剩余的应用程序实例中间重新产生对象复本,等等)。在一些实施方式中,失效的应用程序实例不能不转到FORGOTTEN状态,直到这样的恢复行动完成为止。

[0285] 应用程序实例的FORGOTTEN状态可以不明确地表现在DFDD 110内。更确切地,在一些实施方式中,它可通过从DFDD 110删除应用程序实例的现有状态信息,例如其DFDD项目111而被标记。一般而言,应用程序实例不能从FORGOTTEN状态恢复,虽然在一些实例中,可使用通过新状态分配给被忽略的实例的相同资源来初始化新的应用程序实例。在一些实施方式中,如果当处于FORGOTTEN状态中时,应用程序实例自动地重新开始发送心跳,则DFDD 110可认识到实例被忽略(例如,不再相应于有效的项目111)并可指示实例停止操作或重新设置或重新初始化本身。

[0286] 注意,在一些实施方式中,将全局状态转换分解的启发式算法和转换时间参数对不同类型的应用程序实例可能不同,且这些参数中的一些或全部可由DFDD客户调节。此外,虽然DFDD客户通常可询问DFDD 110的实例以确定给定应用程序实例的当前的全局状态,但是在一些实施方式中,DFDD 110可支持公布/预订状态变化通知模型。例如,DFDD客户可通过预订过程告知DFDD 110,客户希望被通知特定应用程序实例或实例组的所有或某些类型的全局状态变化。当检测到这样的状态变化时,DFDD 110可接着向预订的DFDD客户传送指示该变化的消息。

[0287] 应用程序实例常常可配置成向最接近于该应用程序实例的DFDD 110的实例发送心跳信息。例如,在一些实施方式中,DFDD 110的实例可在配置成支持一个或更多其它应用程序实例的每个计算机系统上提供,以便应用程序实例可仅仅通过参考其主机的本地IP地址并使用为应用程序实例-DFDD通信保留的公知IP端口,来容易地访问DFDD 110的本地实例。然而,如果应用程序实例向DFDD 110的一些实例而不是其它实例报告其状态,则当没有努力同步其状态时,部署的DFDD 110的实例可变得不同。

[0288] 在一些实施方式中,DFDD 110的实例中间的不同可使用类似于上面关于键映射实例140描述的那些同步协议,例如基于闲聊的协议来解决。然而,在很多情况下,由DFDD 110的实例共同管理的DFDD项目111的数量可能实质上小于由键映射实例140管理的键映射项目144的数量。当是这种情况时,较简单的调整协议可用于同步DFDD 110的实例。在图25中

示出了这样的基于闲聊的协议的一个实施方式的操作的方法。在所示实施方式中,操作在块2500开始,其中称为初始实例的DFDD 110的一个实例随机地选择DFDD 110的另一对等的实例用于同步。在一些实施方式中,初始DFDD实例有时可根据初始DFDD实例的状态信息来从目前处于失效状态(例如,NETWORK SPLIT)中的那些DFDD实例中间慎重地选择对等的DFDD实例。如果初始DFDD实例成功地与明显失效的对等实例联系和同步,则可有助于从明显的故障中恢复。

[0289] 初始实例可接着计算在其项目111内反映的应用程序实例的识别信息的散列值(例如,通过散列应用程序实例名称和ID,以及可能地端点或其它识别信息)(块2502)。散列值可根据任何适当的散列算法例如MD5算法来确定。初始实例接着将计算的散列值连同当前应用程序实例状态信息(例如,心跳计数、全局状态信息和/或包括在状态信息114内的任何其它信息)的分类列表一起传送到对等的实例(块2504)。状态信息的列表可根据在初始和对等实例产生一致的列表的任何标准来分类。例如,列表可根据应用程序实例名称和/或ID来分类。

[0290] 注意,如上所述,在一些实施方式中,与应用程序实例相关的状态信息可从包括在心跳消息内的心跳计数信息得到。相应地,在一些实施方式中,DFDD实例可交换用于应用程序实例的心跳计数信息,并可从接收的心跳计数信息得到应用程序实例状态信息,而不是直接从其它DFDD实例接收状态信息。因此,在一个实施方式中,给定DFDD实例可配置成在接收的心跳计数信息的基础上更新特定应用程序实例的状态(例如,根据图 24的状态机),而不管该信息是直接从特定的应用程序实例接收还是通过同步协议间接地从另一DFDD实例接收。在这样的实施方式中,在DFDD 实例中间应用程序实例操作状态信息的同步可包括心跳信息的同步,而没有直接交换应用程序实例的特定全局操作状态(例如,OK、INCOMMUNICADO,等等),这可简化同步协议的操作。

[0291] 响应于接收状态信息的散列值和列表,对等实例以与由初始实例执行的方式一致的方式来计算其自己的应用程序实例的识别信息的散列值(块 2506),并比较最终的散列值与从初始实例接收的散列值(块2508)。如果两个值一致,则存在初始和对等实例都具有相应于相同组的应用程序实例的项目111的高度可能性。对等实例可接着扫描接收的状态信息的列表并根据接收的列表适当地更新其项目111(块2510)。例如,如果接收的列表中的心跳计数或时间戳比存储在一个对等的项目111中的更大或更新,则对等实例可根据接收的列表中的状态信息更新项目111。在一些实施方式中,对等实例可与从初始实例接收列表并行地或之后将其自己的状态信息的列表发送回初始实例用于类似的处理。

[0292] 如果散列值不一致,则有可能对等和初始实例所知的应用程序实例的组在至少一个项目111上不同。相应地,对等实例可请求初始实例所知的项目111的完整的信息转储(dump)(恰好与那些项目111的状态信息114 相反)(块2512)。对等实例可接着添加它缺少的任何项目111,并同步剩余的项目111的状态(块2514)。如上面的,在一些实施方式中,对等实例可与从初始实例接收信息转储并行地或之后将其项目111的完整的信息转储发送回初始实例。

[0293] 可以设想,在一些实施方式中,存在于系统内的DFDD 110的每个实例可配置成每隔一段时间重复地执行刚刚描述的同步协议,或其适当的变形。例如,协议可大致周期性地以一秒的周期或任何其它适当的周期被 DFDD 110的实例执行。进一步可以设想,在一些实

施方式中,DFDD 110 的实例可以大致相同的周期但相对于彼此不同的相移执行同步协议,以便在任何给定的时间,只有DFDD 110的实例的一部分可开始该协议。

[0294] 注意,在一些实施方式中,DFDD 110的实例可用于为任何分布式系统内的任何类型的应用程序实例,而不仅仅是存储服务系统内限定的那些应用程序实例协调和传递状态信息。此外,在一些实施方式中,DFDD实例的不同组可管理不同的应用程序实例状态信息。在一些这样的实施方式中,通过给作为相同组的成员的DFDD 110的实例分配共同的标识符并要求标识符匹配作为DFDD同步的条件可将组彼此区别开。例如,管理存储服务系统应用程序实例的DFDD实例可具有标识符,该标识符不同于配置成管理与存储服务系统无关的其它应用程序实例的状态的DFDD实例,且只有具有相同标识符的那些DFDD实例可根据图25的同步协议彼此交换信息。

[0295] 在一些实施方式中,DFDD组标识符可用于区别存在于相同的系统中的应用程序实例的不同配置。例如,相应于“生产”标识符的DFDD 110的一组实例可用来管理存储服务系统或另一分布式系统的生产版本,并可反映相应于生产系统的一组应用程序实例,而相应于“测试”标识符的DFDD 110的另一组实例可部属为管理相应于不同的应用程序实例组和状态的系统的测试版本。注意,在一些情况下,相应于任一系统版本的应用程序实例和/或DFDD实例可在相同的基本系统资源(例如,在相同的计算机系统)上执行,但可由于其不同的DFDD组标识符而对彼此变得透明。例如,当执行例如图25中所示的协议的同步协议时,DFDD实例可首先确定它们是否是相同组的成员(例如,通过交换组标识符)并视此确定的情况而执行随后的同步步骤,从而便于组之间应用程序实例状态信息的分离。

[0296] 注意,前述用于同步DFDD 110的实例的基于闲聊的协议可不仅帮助在整个存储服务系统中分布现有的应用程序实例的操作状态,而且也可通过其它系统组件便于新应用程序实例的发现。例如,一旦新应用程序实例被初始化并与DFDD 110的实例(例如,在新应用程序实例被初始化的系统上在本地操作的实例)接触,就可产生相应于新实例的新项目111。由于在其上创建新实例111的DFDD 110的实例使其状态与DFDD 110的不同的其它实例同步,因此新项目111可在整个系统中传播。为了各种目的而询问DFDD 110以识别应用程序实例(例如,以存储新对象30或更新键映射实例140)的DFDD客户可接着被提供关于新应用程序实例以及任何现有的实例的状态信息。

[0297] 还注意到,在上述实施方式中,与故障检测和发现有关的应用程序实例状态信息变化可在整个系统中传播,而不干涉参考那些实例的应用程序实例或DFDD客户的部分。也就是说,给定的应用程序实例只需要知道如何将心跳信息传送到DFDD 110的一个实例。不需要知道系统内DFDD 110 的所有实例、其它应用程序实例或可调用给定应用程序实例的不同客户。类似地,DFDD客户不需要单独地知道其他客户或系统内所有应用程序或 DFDD实例;客户可依赖于DFDD 110的实例,他们与其通信以合理地获得关于系统内可利用的资源的状态的当前信息。通过允许应用程序实例的状态改变而不需要立即通知其它应用程序实例或客户这样的变化,DFDD 110可有助于存储服务系统的可缩放性。

[0298] 存储类

[0299] 在一些存储服务系统实施方式中,对象30可相对于其复制的程度、复本在区字段310中间的分布、复本被存储到的存储资源的类型和/或其它系统特征或策略被统一处理。例如,系统可试图将每个对象30复制相同数量的次数到相同数量的不同区域310。然而,不

同的客户50对不同的对象30可有不同的存储要求。例如,一个客户50可能希望以比默认的存储策略可提供的更高的可靠度(例如,根据复本的数量和分布)存储特定的对象30,而另一客户50可能甚至不需要默认级别的可靠性。可选地,客户50可希望通过限制对象复本被分布到的区域310的数量来增加对象写入性能,可能以可靠性为代价。

[0300] 相应地,在一个实施方式中,存储服务系统例如图2的系统可配置成支持对象30的存储类。一般而言,给定对象30的存储类可指定相对于给定对象30影响服务等级协议(SLA)的存储服务系统特征或特性的任何集合。服务等级协议可通常反映保证或期望的集合,在这些保证或期望下服务供应商向客户提供服务,以换取从客户接收的一些考虑因素(例如,费用或任何其它适当类型的考虑因素)。例如,由存储服务系统管理的对象30的SLA可为服务指定对象可靠性、有效性、访问性能(例如,等待时间、带宽)、费用或速率的不同级别,或客户与对象30的交互作用的任何其它可测量的方面。在一些实施方式中,存储类可只指定SLA特性的特定子集合(例如,如下所述对象复本的数量和分布),而在其它实施方式中,存储类可直接相应于包括关于给定对象30的SLA协议的所有定义的方面的综合SLA。

[0301] 在一个实施方式中,存储服务系统可定义存储类的固定集合,每个存储类都具有特别定义的SLA特性,且客户50可选择使特定的对象30与特定的存储类关联起来。例如,默认的存储类可指定对象30复制至少三次到至少两个不同的区域310。高度可靠性的存储类可指定对象30复制至少五次到至少三个不同的区域310。预算存储类可指定对象30的单个复本存储在单个区域310内。局部存储类可指定对象30复制至少三次到单个区域310。在其它实施方式中,存储服务系统可定义具有其它特性的存储类,或可允许客户50通过指定将应用到给定对象30的存储策略的组合(例如,如上关于节点拣取器130描述的)来为给定的对象30定制存储类。

[0302] 如上所述,SLA特性可扩展而超出复本的数量和区域的数量,复本应分布到该区域。在一个实施方式中,特定存储类的SLA特性可包括相应于与特定存储类关联的对象30的预期的处理等待时间的指示。例如,一个存储类可对给定的成本规定低的预期的处理等待时间,而另一个可为较低的成本指定较高的预期的处理等待时间。不同水平的预期的处理等待时间可以不同方法实现。例如,从给定协调器120的观点来看,由于这种因素,例如节点160到给定协调器120的接近度、在节点160可得到的资源的级别和类型、节点160的处理负载或其它相关的因素,一些节点160可展示比其它节点低的访问等待时间。因此,受由给定存储类指定的其它SLA特性实现的限制的影响,在一些实施方式中,协调器120和/或节点拣取器130可配置成选择为在存储类中的对象30展示较低的访问等待时间的节点160,该存储类规定较低的预期处理的等待时间。在其它实施方式中,协调器120可配置成根据与对象30关联的存储类的预期处理的等待时间来为优先考虑对对象30的客户访问请求的处理。例如,协调器120可实现配置成使处理偏向于有利于具有较低的预期处理等待时间的存储类的不同队列或其它处理控制或数据结构,同时确保对具有较高的预期处理等待时间的存储类的请求最终完成。

[0303] 存储类可由客户50在对象30最初存储在存储服务系统内时指定。可选地,在一些实施方式中,客户50可在对象30存在于存储服务系统内的任何时间改变与对象30关联的存储类。如果当对象30被最初存储时客户50没有指定存储类,则可使用默认的存储类例如上面描述的存储类。如上所述,在一些实施方式中,对象30的存储类可存储在与对象30的键关

联的键映射记录148中。在这样的实施方式中,协调器120和/或复制器180 可配置成当存储、复制和维持对象30的现有复本时考虑对象30的存储类。可以设想,客户50可被要求为与不同存储类关联的对象30支付不同使用费用。例如,高度可靠性存储类可通常使用较多的系统资源,而预算存储类可使用较少的资源。相应地,对于给定大小的对象30,为了使用前面的存储类来储存对象30,客户50可被要求支付得较多,而对后者较少。

[0304] 在图26中示出了存储类在存储服务系统内的操作的方法的一个实施方式。在所示实施方式中,操作在块2600开始,其中客户50指定将与特定的对象30相关联的存储类。随后,存储类持久地与存储服务系统内的特定对象30关联(块2602)。例如,存储类的指示可通过代表客户50的协调器120存储在特定对象30关联的数据结构中,例如键映射记录148。接着根据指定的存储类的特性来配置与对象30关联的对象数据的状态(块 2604)。例如,如果存储类指定对在区域310中间的对象复本的数量和/或分布的某些要求,则协调器120和/或复制器180可操作来产生并分布必要的复本,以便存储系统相对于特定对象30的最终的状态满足存储类的要求。在一些实施方式中,复制器180可配置成确保随着时间的过去而为对象30维持存储类要求。例如,如果复本失效,则复制器180可配置成检测故障并产生额外的复本。

[0305] 可以设想,在一些实施方式中,由给定存储类指定的存储特性可包括通过位存储节点160区分不同类型的存储资源。例如,在一些实施方式中,一些位存储节点160可包括比其它节点更高性能的存储设备,或单独的位存储节点160可包括较高和较低性能设备的组合。在这样的实施方式中,存储类可指定一种或另一类型的设备应用于与该类相关联的对象30。

[0306] 动态复制

[0307] 如上所讨论的,在一些实施方式中,节点拣取器130可配置成产生识别特定对象30的复本应写入的特定位存储节点160的写入计划。这样的写入计划可以这样的方式产生,使得一旦写入计划例如通过协调器120实现,不同的写入策略相对于特定的对象30就被满足。例如,由写入计划规定的节点160的数量可根据特定对象30的复本的最小需要的数量、在其上应分布复本的不同区域310的最小数量,或任何其它存储策略考虑因素来确定。

[0308] 在一些实施方式中,节点拣取器130可配置成以静态的方式产生写入计划,其中节点160根据可预测的程序被一致地选择,而不考虑节点160 的当前状态。例如,节点拣取器130可一致地选择相同组的节点160用于存储复本,或可通过很多节点160以循环的方式旋转。然而,在大规模实现中,存储服务系统可包括可在不同的时间在相当不同的状态中操作的节点160。例如,一些节点160可为无效的,另一些节点可为有效的但充满请求行为或具有少量的自由资源,以及还有另一些可相对空闲或具有相当多的自由资源。

[0309] 此外,从给定协调器120或节点拣取器130的观点来看,节点160可呈现不同级别的通信费用。例如,位于相同区域310或数据中心300内作为协调器120的节点160可通过本地低等待时间网络连通性被访问。相反,位于与协调器120不同的区域310或数据中心300内的节点160可呈现比本地节点160实质上更高的访问等待时间。而且,在一些实施方式中,区域310或数据中心300之间的通信可发生在具有与本地通信不同的经济费用模型的通信网络上。例如,区域310内的通信可发生在具有大量带宽的专用局域网(LAN)上,对于该带宽没有用于数据传输的基于使用的费用。相反,数据中心300之间的通信可发生在设备上,例如租

用的电信设备、公共互联网、专用广域网(WAN)设备或其它远程电信网络。这些设备可一般比LAN设备更具有带宽限制,且在一些实例中可呈现由第三方要求支付的利用费用(例如,基于峰值或合计的带宽利用),该费用可能不适用于LAN通信。

[0310] 不同节点160的操作状态以及对那些节点的通信的费用都可随着时间的过去而变化。例如,曾经有效或空闲的节点160可能在稍后的时间变得无效或繁忙,反之亦然。类似地,通信费用例如等待时间和/或经济费用可能在一些时期期间较高而在另一些时期期间较低(例如,在峰值对非峰值利用的时期期间)。由于这个变化性,曾经有效和费用低的写入计划可能相当不有效、费用较高、或甚至在另一时间不可行(例如,如果在写入计划中指定的节点160变得繁忙、通信起来缓慢或无效)。

[0311] 因此,在一些实施方式中,节点拣取器130可配置成根据与节点160 关联的当前状态信息来动态地确定给定的写入计划,用于写入给定对象30 的复本。一般而言,动态确定的写入计划可考虑关于节点160的可观察的动态状态信息。也就是说,动态确定的写入计划可作为可随着时间的过去而变化的节点状态信息的函数而产生。因此,对给定对象30的动态确定的写入计划本身可根据节点160的基本状态信息随着时间的过去而变化,与可独立于节点160的状态而确定的静态产生的写入计划相反。

[0312] 如上所述,在写入计划的动态产生中可考虑很多不同类型的状态信息。一般来说,节点160的状态信息可包括关于给定节点160的状态信息以及关于通信资源(例如,网络资源)的状态信息,通过该资源可访问给定的节点160。在不同实施方式中,关于给定节点160的状态信息可包括给定节点160的操作状态,例如给定节点160(或与该节点关联的应用程序实例)是否处于由DFDD 110指示的OK、INCOMMUNICADO或其它操作状态中,如上所述。关于给定节点160的状态信息还可包括负载状态信息,其可比操作状态信息更详细地指示给定节点160的行为。例如,在不同实施方式中,负载状态信息可指示处理器利用、存储器利用、存储设备容量利用、存储设备输入/输出带宽利用或相应于给定节点160的网络接口带宽利用的级别,或节点行为的任何其它可测量的方面。如上所述,在一些实施方式中,除了操作状态信息以外,负载状态信息也可通过DFDD 110得到。

[0313] 可称为网络费用信息的通信资源状态信息可包括与到给定节点160的一个或更短通信路径的状态有关的任何适当的信息。在不同实施方式中,网络费用信息可指示与将消息传送到给定节点160和/或从给定节点160传送出来相关的网络通信等待时间,这可根据时间(例如,秒、毫秒等)、网络跳数(例如,传送消息的路由步骤的数量)或另一适当的度量来表达。在一个实施方式中,网络费用信息可包括可用于与给定节点160通信的有效带宽的指示(例如,数据传输的速率)。在另一实施方式中,网络费用信息可包括与和给定节点160的网络通信相关的经济费用。例如,这样的费用可表达为用于传输或接收一定数量的数据的付费率或网络通信的任何其它适当的费用或费率模型。

[0314] 节点拣取器130在动态地确定对象30的写入计划中可通常使用节点 160的状态信息的任何适当的函数。在一些实施方式中,由节点拣取器130 实现的存储策略(其可以是除了或代替前面所述的那些存储策略)可为状态信息指定准则或要求,该状态信息限制可能有资格包括在特定对象30 的写入计划中的节点160。在不同的实施方式中,这些策略可全局地(例如,对于所有的对象30)应用于对象30的特定组(例如,包括在特定存储类或存储桶中的、具有共同的键前缀,或另外地表示为组的成员的对象)、或应用于各个对象30(例如,

相应于规定将与对象30关联的特定策略的客户)、或这些的任何适当组合。作为例子,特定的存储类可规定存储策略,该存储策略要求某个最小数量的复本展示不多于某个最大的通信等待时间。相应地,在为该存储类中的对象30开发写入计划中,节点拣取器 130可配置成根据节点160是否满足指定的最大通信等待时间来选择至少一些节点160。

[0315] 在一些实施方式种,节点拣取器130还可配置成根据对节点状态信息的最佳化的不同类型来产生写入计划。例如,作为指定与写入计划相关的特定的最大网络费用或其它费用的可选方案,存储策略可指定费用应在可在特定的时间得到的资源中间被最小化。相应地,节点拣取器130可配置成例如通过选择具有较低的网络通信或其它相关费用的节点160来最小化与写入计划相关的一个或更多费用。在一些实施方式中,这样的最小化可出现在存在其它限制时,例如指定其它节点状态信息要求的其它存储策略。

[0316] 此外,注意,在一些实施方式中,一些节点状态信息可以可预测的方式随着时间的过去而变化。例如,与数据中心300之间的网络通信相关的带宽费用可根据定义明确的费率计划来变化。在一些这样的实施方式中,最小化与写入计划相关的费用可包括识别一段时间,在该段时间期间应执行写入计划的全部或一些,取决于与特定的时间段相关的费用。例如,节点拣取器130可确定用于与远程数据中心300通信的带宽在某个未来的时间比在当前的时间更廉价,且可进一步确定,包括在远程数据中心300的节点160的写入计划的费用可通过在被确定的未来时间至少执行指向远程数据中心的那些存储操作来被最小化。该过程的一个可能的结果是,由节点拣取器130产生的写入计划可指示应延迟给定对象30的一些(或可能全部)复本的产生直到被确定的未来时间。

[0317] 很多不同的存储策略可应用于特定的对象30是可能的。进一步地,在一些实例中,产生满足与特定对象30关联的每个存储策略的单个写入计划可能是不可能的。例如,与特定对象30关联的存储策略可指定,最小数量的复本应存储和分布在最小数量的不同区域310内。然而,在写入计划为特定的对象30产生时,节点拣取器130正在其中执行的区域310可由于瞬时通信故障而与其它区域310暂时隔离。结果,将复本成功地分布到其它区域310以满足相应的存储策略可能是至少暂时不可能的。

[0318] 在一个实施方式中,节点拣取器130可配置成在最小化可由写入计划满足的存储策略的数量的基础上动态地确定对象30的写入计划。在存在不是最佳的条件时,这可能导致表示满足存储策略的“尽力服务”的写入计划。例如,在刚刚描述的特定情况中,区域多样性策略可能由于通信故障不是可以满足的,但最少复制策略可通过将特定对象310的所要求的最小数量的复本存储在本地区域310内是可以满足的。在一些实施方式中,存储策略的最大化可发生在各种限制下。例如,一些存储策略可被识别为强制性的,以便如果它们不是可以满足的,则写入计划不能被确定,且存储对象的相应客户请求可能失败。其它存储策略可具有优先等级或权重,以便较高优先级的存储策略可优先于较低优先级的策略在最大化过程期间被选择。在另一实施方式中,通过最大化最终的存储计划的总的权重(在所满足的存储策略的重要性的基础上确定),而不是或除了被最终的存储计划满足的存储策略的数量,可执行存储策略的选择。

[0319] 注意,用于动态地确定对象30的写入计划的各种技术不需要只出现在对象30被最初存储时。如上所述,在一些实施方式中,复制器180可配置成检查相应于对象30的键映射项目144以确定对象30的复本是否是可访问的。如果特定对象30的任何复本是不可访问的,

则复制器180可配置成从节点拣取器130请求可用于产生额外的复本的新的写入计划。新的写入计划可由节点拣取器130使用上述技术的任何适当的组合动态地确定。此外,在一些实施方式中,复制器180可配置成相对于各种存储策略更一般地监控对象30的适应性。例如,复制器180可配置成确定除了最少复制策略或策略的任何其它适当集合以外,对象30的现有的复本组是否还满足区域多样性策略。在一个这样的实施方式中,如果复制器180确定被特定对象30的现有的复本满足的策略的数量小于某个阈值,则复制器180可从节点拣取器130请求可被动态地确定以最大化如上所述的被满足的存储策略的数量的新存储计划。在可选的实施方式中,复制器180在确定特定的强制性存储策略不再被满足时或当确定被满足的存储策略的总权重落到阈值权重之下时可请求新的存储方案。

[0320] 图27示出根据位存储节点160的当前状态信息动态地确定用于存储数据对象的一个或更多复本的写入计划的方法的一个实施方式。在所示实施方式中,操作在块2700中开始,其中存储给定对象30的客户请求被接收。在一个实施方式中,根据网络服务协议通过网络服务接口100可接收这样的请求,如上详细描述。

[0321] 随后,根据位存储节点160的当前状态信息来动态地确定用于存储给定对象30的复本的写入计划(块2702)。例如,节点拣取器130可配置成根据可应用于给定对象30的各种存储策略来确定写入计划,其中这些策略考虑任何适当的当前状态信息,例如节点操作状态、节点负载状态信息、网络通信费用或如上面详细描述的任何其它适当的状态信息。此外,如上所述,在一些实施方式中,动态地确定写入计划可包括对于状态信息或存储策略的最佳化,例如通过最小化与写入计划相关的费用或最大化被写入计划满足的存储策略的数量或权重。

[0322] 给定对象30的复本可接着根据动态确定的写入计划存储到一个或更多位存储节点160(块2704)。例如,协调器120可配置成产生指向在写入计划中指定的各个位存储节点160的位存储对象放置操作,如上所述。在一些实施方式中,写入计划的一些存储操作可在与其它操作不同的时间执行,如上所述。

[0323] 如前面提到的,在一些实施方式中,写入计划可相对于对象30被动态地确定,该对象30的一个或更多复本已经被存储在位节点160中间。图28示出这样的方法的一个实施方式。在所示实施方式中,操作在块2800 中开始,其中给定对象30的一个或更多现有复本被检查。例如,如上所述,复制器180的一个实施方式可配置成确定给定对象30的现有复本是否是可访问的,和/或给定对象30的现有复本满足与该对象关联的存储策略的程度。

[0324] 响应于检查给定对象30的复本,可确定需要产生一个或更多额外的复本(块2802)。例如,现有的复本可能失效或另外地变得不可访问,导致少于最小数量的复本。可选地,现有复本的状态可能相对于一个或更多存储策略是不足的。随后,根据位存储节点160的当前状态信息来动态地确定用于存储给定对象30的额外复本的写入计划(块2804)。以类似于前面描述的方式或根据其任何适当变形可确定这样的写入计划。注意,在一些实施方式中,如果确定对给定的对象30不需要产生额外的复本,则可不确定写入计划。

[0325] 给定对象30的复本接着根据动态确定的写入计划被存储到一个或更多位存储节点160(块2806)。例如,复制器180可配置成产生指向在写入计划中指定的各个位存储节点160的位存储对象放置操作,如上所述,或可简单地指引存储给定对象30的现有复本的一个或更多节点160来将其复本拷贝到在写入计划中指定的一个或更多节点160。

[0326] 示例性计算机系统实施方式

[0327] 可以设想, 在一些实施方式中, 上述任何方法或技术可实现为能够通过计算机可访问的介质存储或传送的程序指令和数据。这样的方法或技术可包括例如且没有限制地存储客户50、网络服务平台100、DFDD 110、协调器120、节点拣取器130、键映射实例140、位存储节点160、复制器180 和/或复制符键映射190的功能。这样的方法或技术可进一步包括在图6-9、13-15、20-22和25-8中示出来的任何方法及其任何适当的变形。可执行这样的程序指令以履行特定的计算功能, 例如特定的方法或上述方法的部分, 以及提供更一般的操作系统功能、应用程序功能和/或任何其它适当的功能。注意, 在一些实施方式中, 在上面被描述为独立的组件或方法在其它实施方式中可合并到比所示的那些更少的实体中, 或功能可与从上述的划分不同地在组件和方法中划分。

[0328] 图29中示出了包括计算机可访问的介质的计算机系统的一个示例性实施方式。这样的系统也可称为节点。如前面所讨论的, 在一个实施方式中, 上述各种存储系统组件的任何一个的功能可分布在很多节点中, 以便给定组件可由一个或更多节点实现或在一些节点中划分。虽然在一些实施方式中, 节点可专门实现单个存储服务系统组件的功能, 但是在其它实施方式中, 节点可实现一些不同的系统组件的所有或部分的功能。在所示实施方式中, 计算机系统2900包括通过输入/输出 (I/O) 接口2903耦合到系统存储器2920的一个或更多处理器2910。计算机系统2900进一步包括耦合到I/O接口2930的网络接口2940。

[0329] 在不同实施方式中, 计算机系统2900可为包括一个处理器2910的单一处理器系统, 或包括一些处理器2910 (例如, 两个、四个、八个或其它适当的数量) 的多处理器系统。处理器2910可为能够执行指令的任何适当的处理器。例如, 在不同实施方式中, 处理器2910可为实现各种指令集结构 (ISA) 中的任一种, 例如x86、PowerPC、SPARC或MIPS ISA或任何其它适当的ISA的通用或嵌入式处理器。在多处理器系统中, 每个处理器2910可通常但不是必要地实现相同的ISA。

[0330] 系统存储器2920可配置成存储处理器2910可访问的指令和数据。在不同实施方式中, 系统存储器2920可使用任何适当的存储器技术来实现, 例如静态随机存取存储器 (SRAM)、同步动态RAM (SDRAM)、非易失性/闪型存储器或任何其它类型的存储器。在所示实施方式中, 实现期望功能的程序指令和数据, 例如上面详细描述的那些存储服务系统部件中的任何一个和其它功能显示为存储在系统存储器2920内作为代码2925。

[0331] 在一个实施方式中, I/O接口2930可配置成协调处理器2910、系统存储器2920和设备中任何外围设备之间的I/O业务, 其中的外围设备包括网络接口2940或其它外围接口。在一些实施方式中, I/O接口2930可执行任何必要的协议、计时或其它数据转换, 以将数据信号从一个组件 (例如, 系统存储器2920) 转换成适合于由另一组件 (例如, 处理器2910) 使用的格式。在一些实施方式中, I/O接口2930可包括对通过不同类型的外围总线连接的设备的支 持, 例如外围部件互联 (PCI) 总线标准或通用串行总线 (USB) 标准的变型。在一些实施方式中, I/O接口2930的功能可分成两个或更多分离的组件, 例如北桥和南桥。此外, 在一些实施方式中, I/O接口2930, 例如系统存储器2920的接口的一些或所有功能可直接合并到处理器2910中。

[0332] 网络接口2940可配置成允许数据在计算机系统2900和连接到网络的其它设备例如其它计算机系统之间交换。在不同实施方式中, 网络接口 2940可通过有线或无线一般数

据网络例如任何适当类型的以太网、通过电信/电话网络例如模拟语音网络或数字光纤通信网络、通过存储局域网例如光纤通道SAN或通过任何其它适当类型的网络和/或协议来支持通信。

[0333] 在一些实施方式中,系统存储器2920可为配置成存储如上所述的程序指令和数据的计算机可访问的介质的一个实施方式。然而,在其它实施方式中,对不同类型的计算机可访问的介质可接收、发送或存储程序指令和/或数据。一般而言,计算机可访问的介质可包括存储介质或存储器介质,例如磁或光介质,例如通过I/O接口2930耦合到计算机系统2900的磁盘或CD/DVD-ROM。计算机可访问的介质还可包括任何易失性或非易失性介质例如RAM(例如,SDRAM、DDR SDRAM、RDRAM、SRAM等)、ROM等,其可包括在计算机系统2900的一些实施方式中作为系统存储器 2920或另一类型的存储器。通过计算机可访问的介质存储的程序指令和数据可通过传输介质或信号例如电、电磁或数字信号来传输,这些信号可通过通信介质例如可通过网络接口2940来实现的网络和/或无线链接来传送。

[0334] 虽然以相当多的细节描述了上面的实施方式,当上面的公开被充分认识到时很多变形和更改对本领域的技术人员将变得明显。意图是下列的权利要求被解释为包括所有这样的变形和更改。

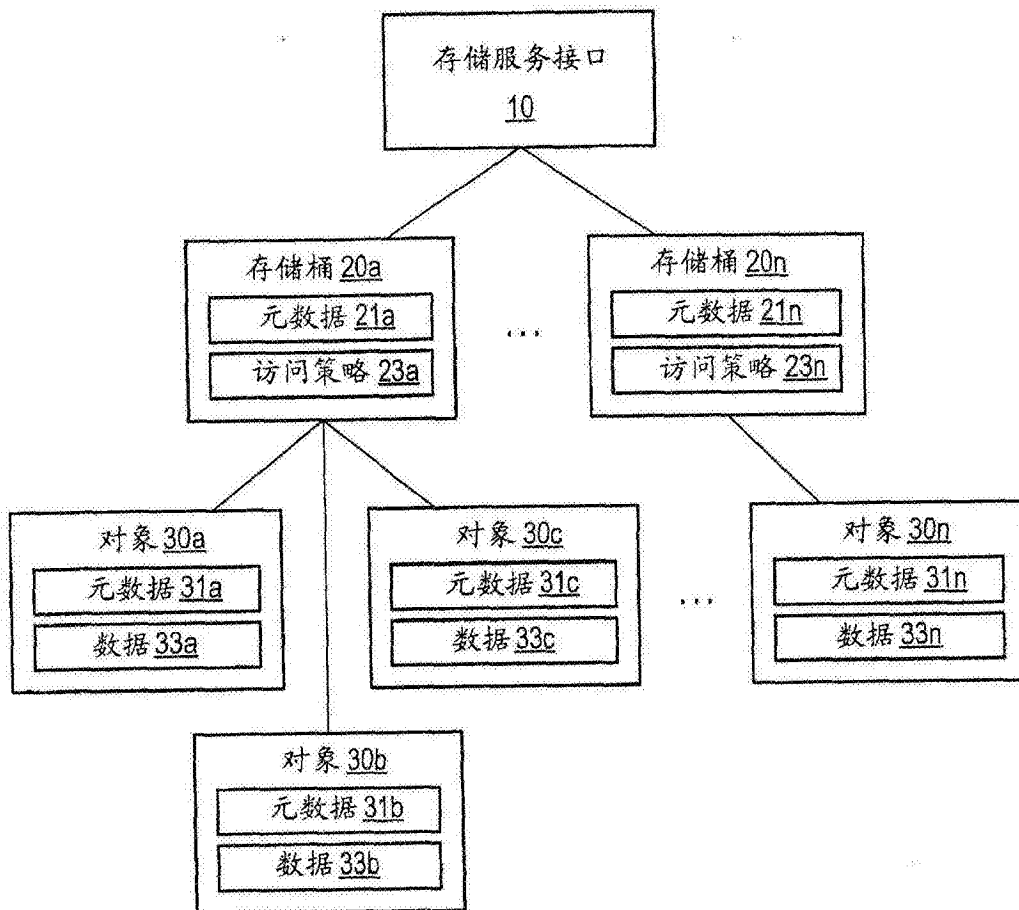


图1

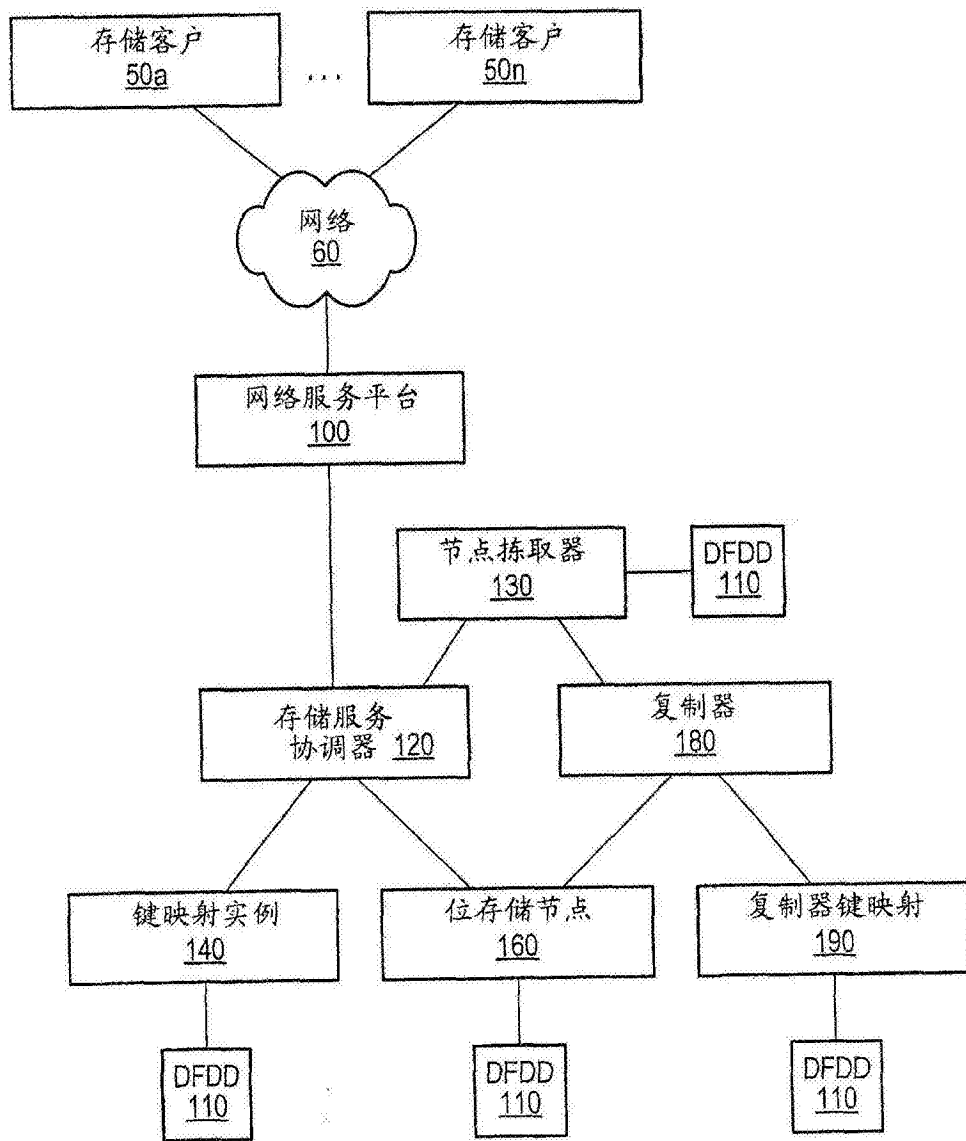


图2

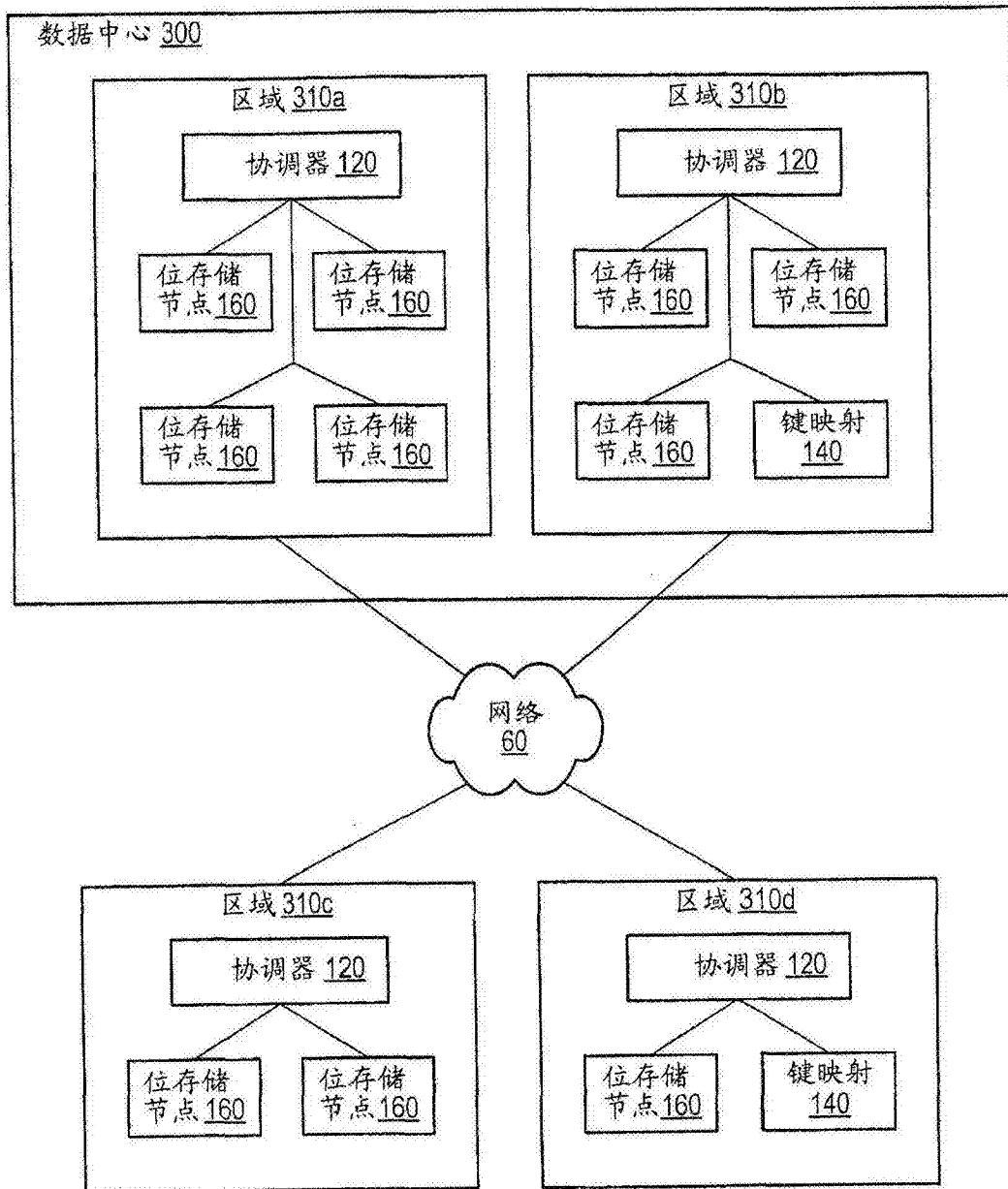


图3

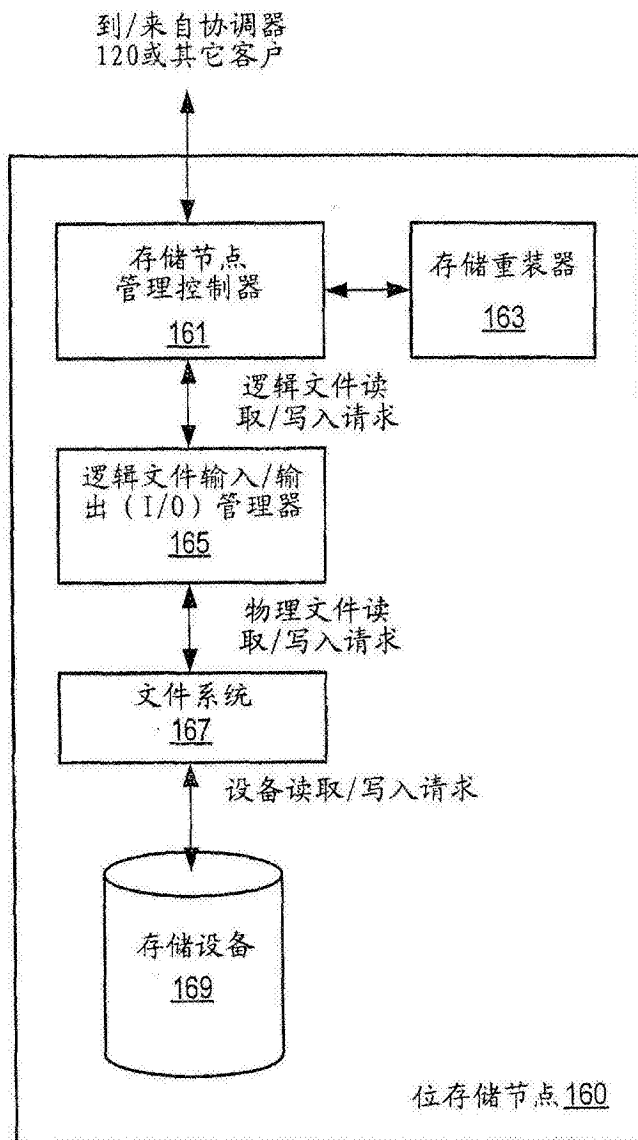


图4

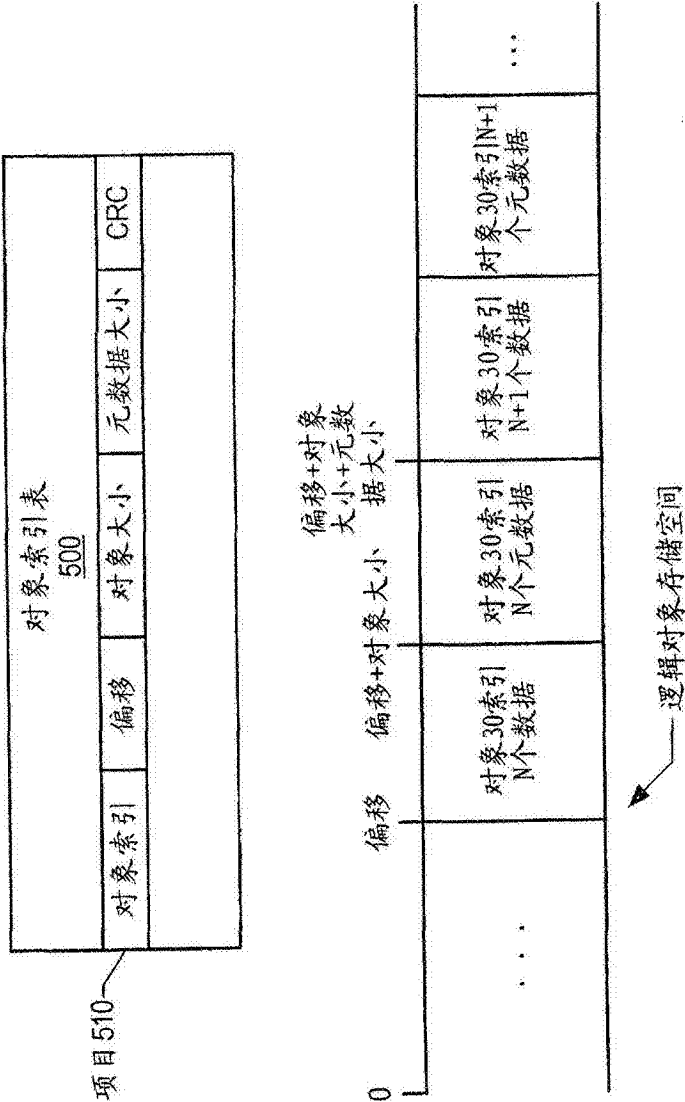


图5

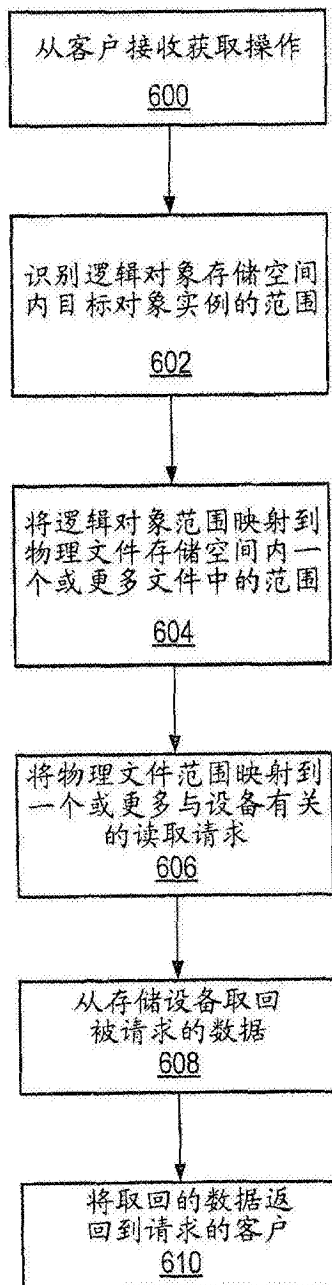


图6

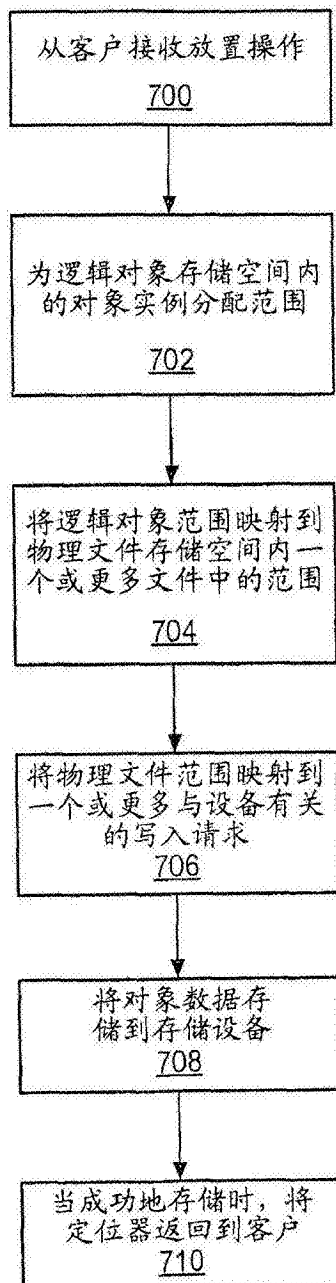


图7

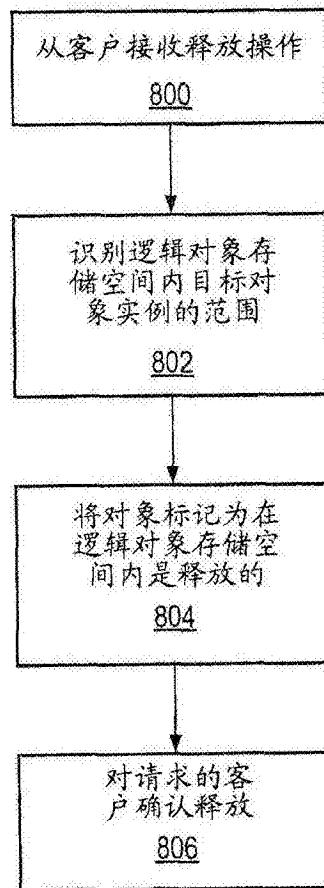


图8

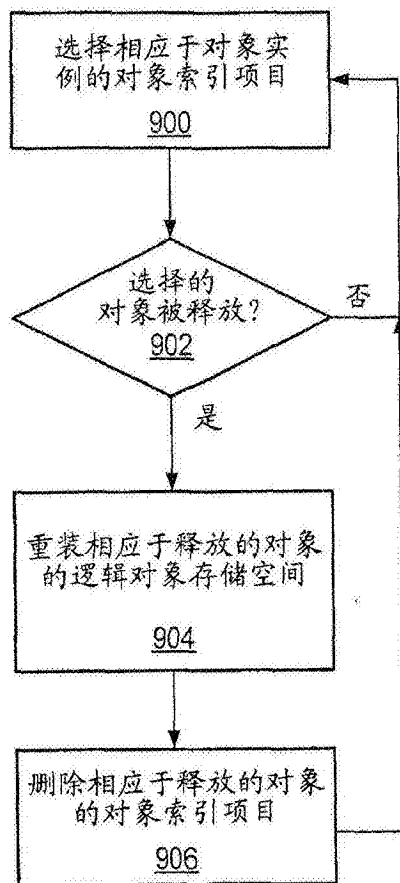


图9

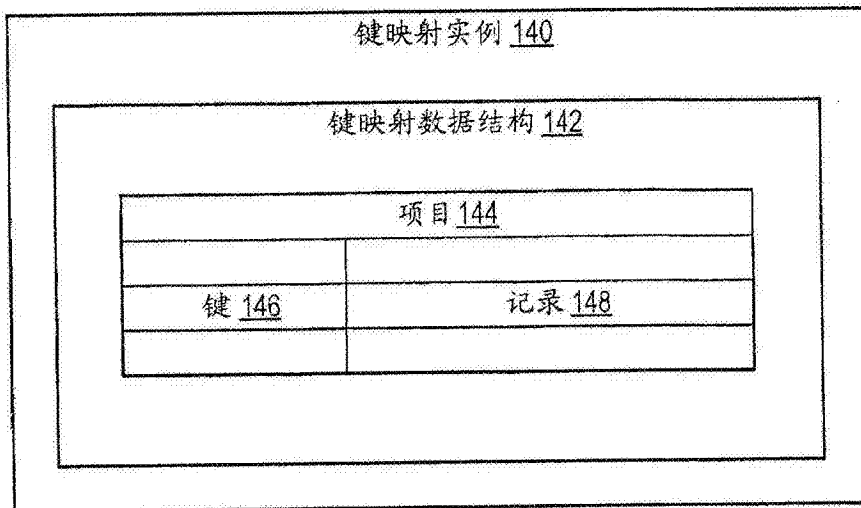


图10

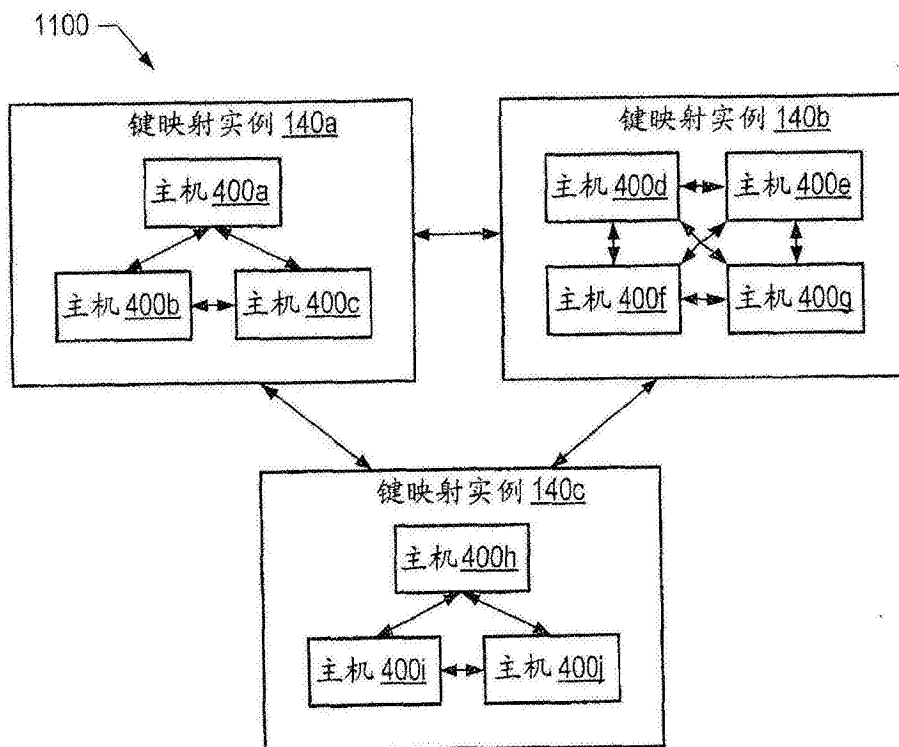


图11A

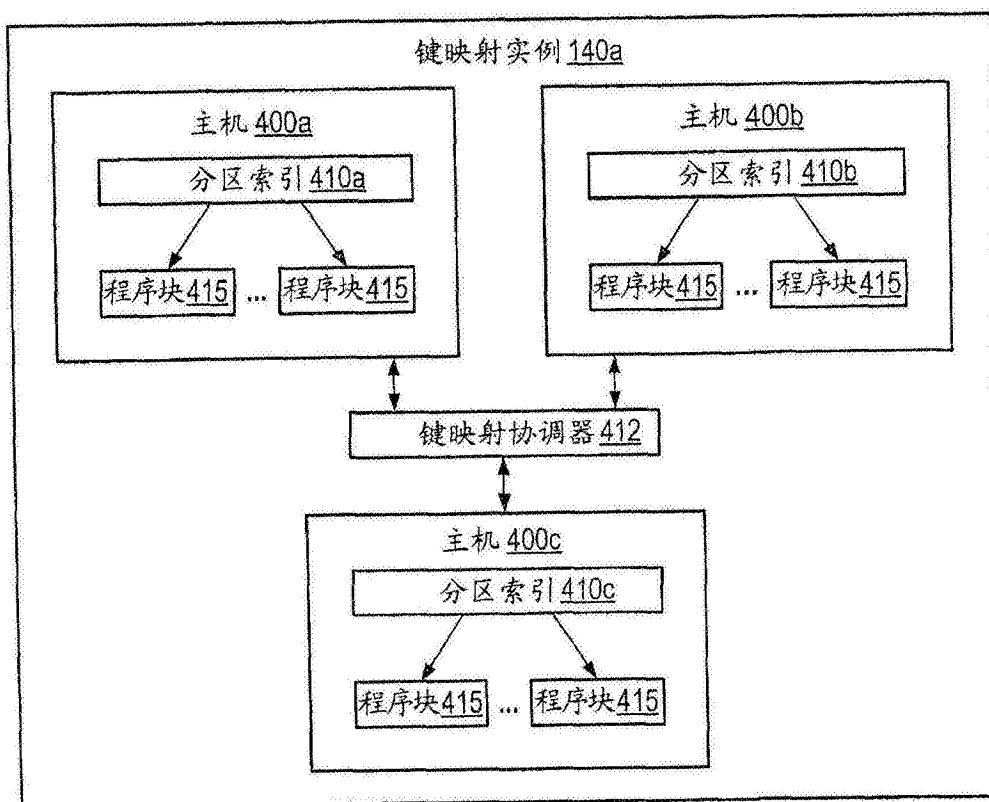


图11B

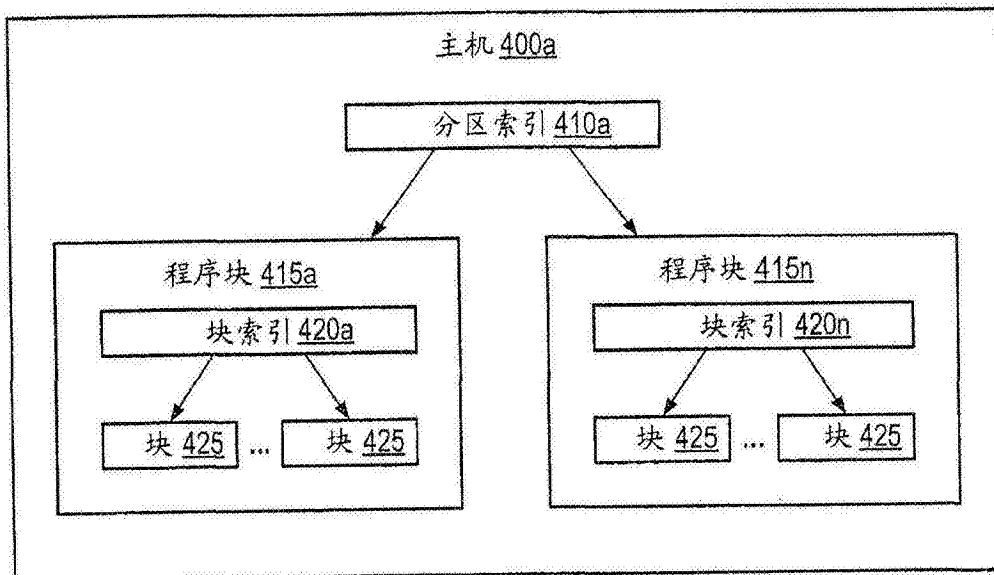


图11C

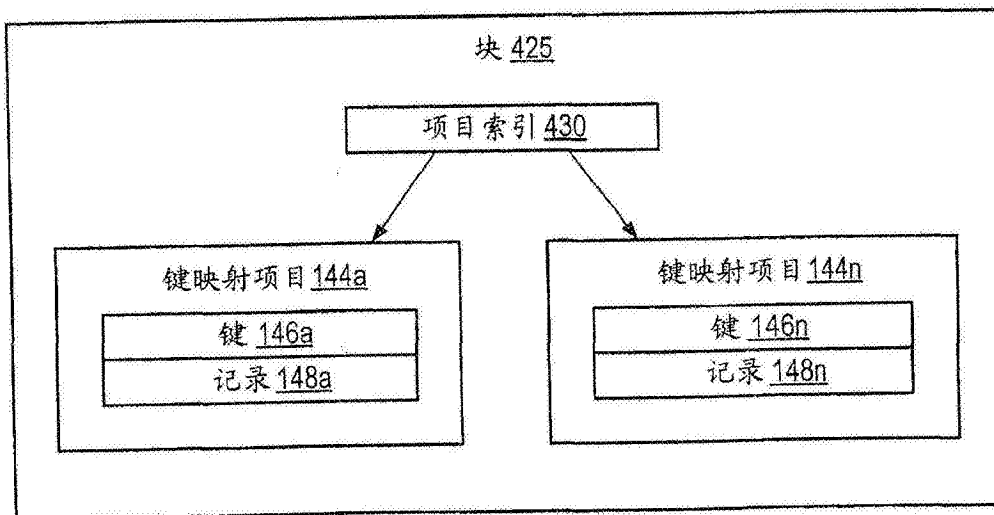


图11D

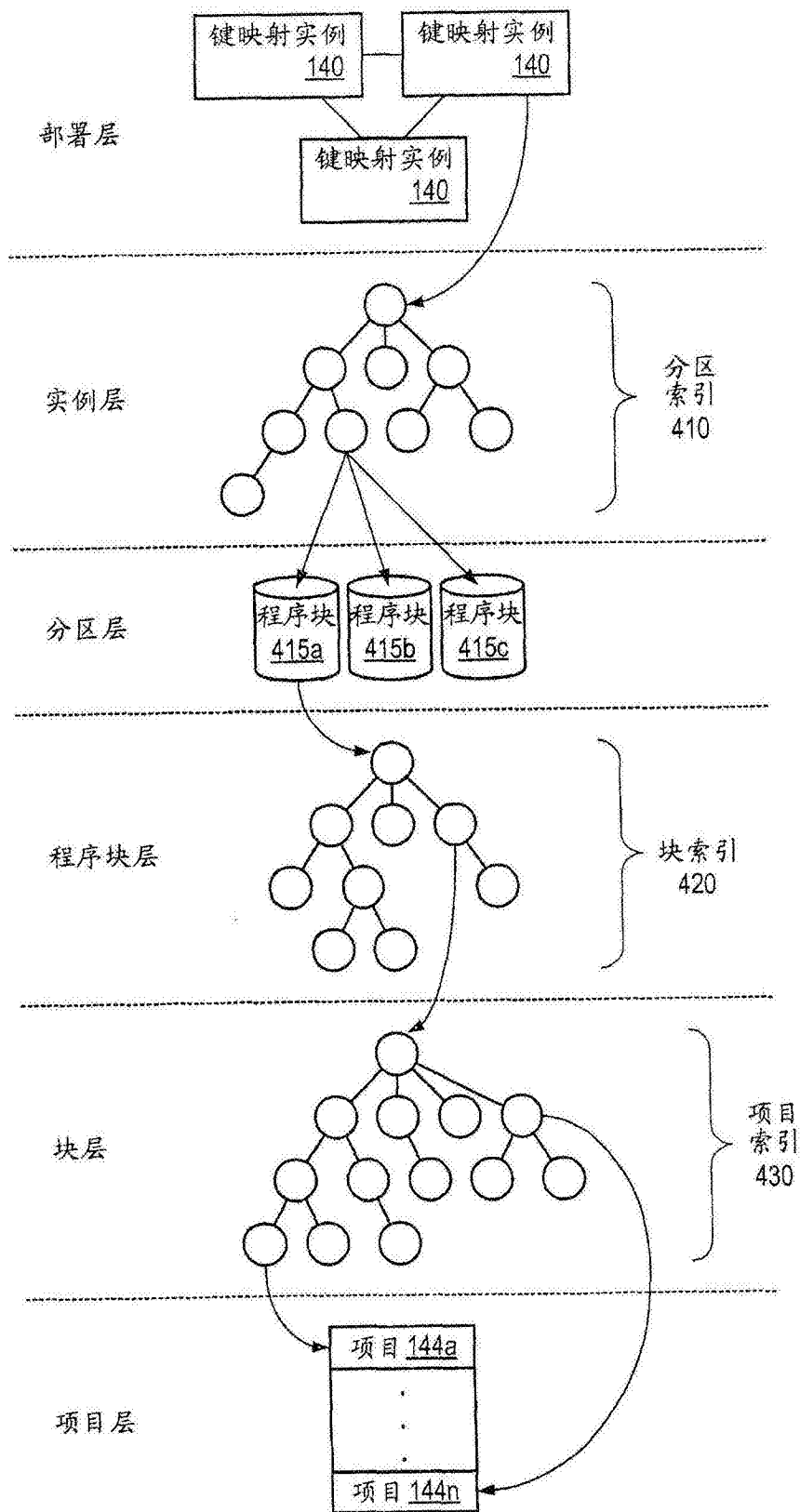


图12

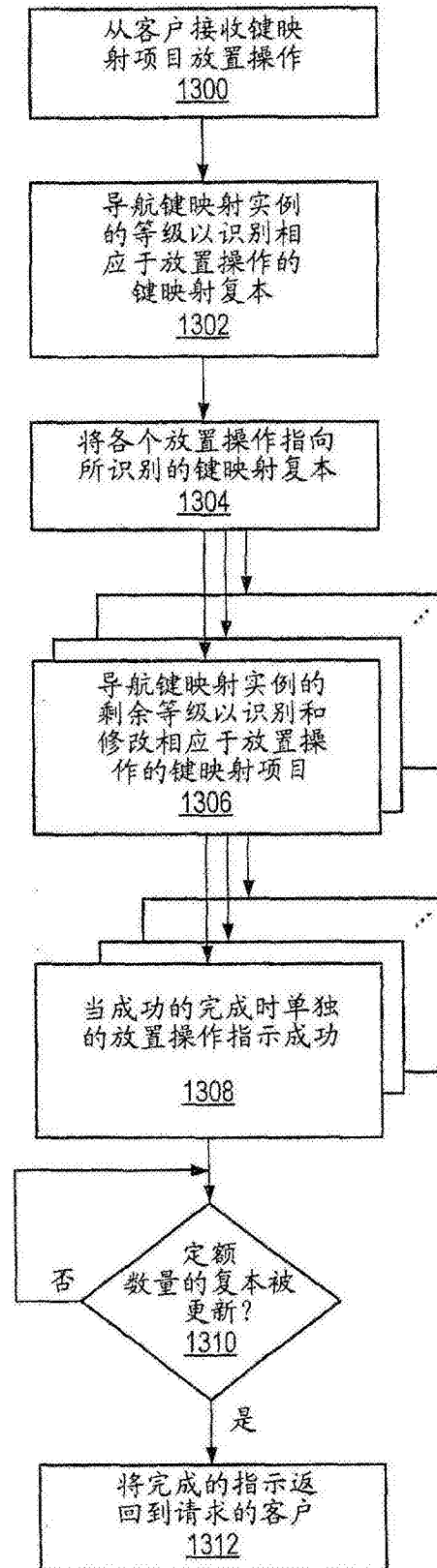


图13

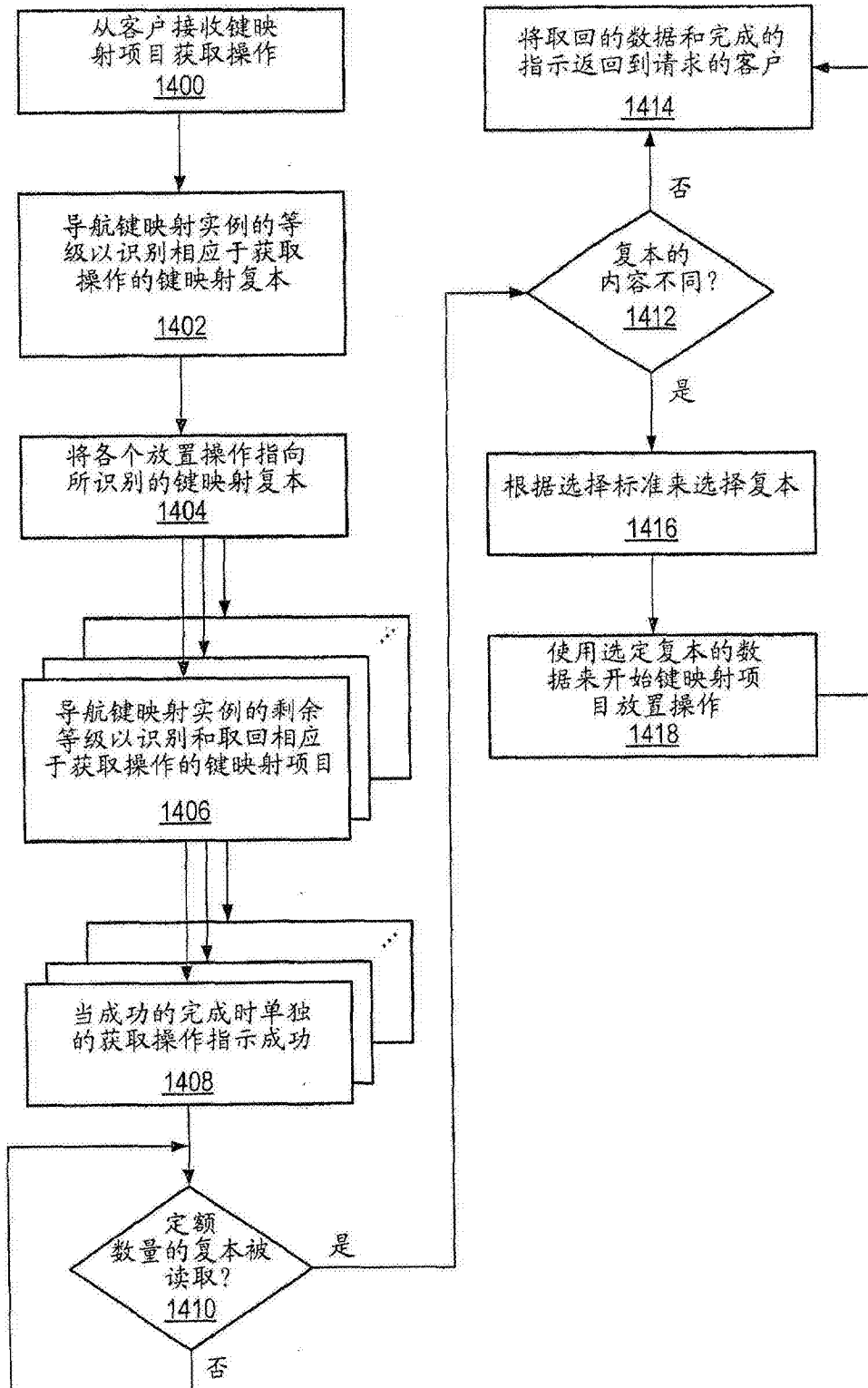


图14

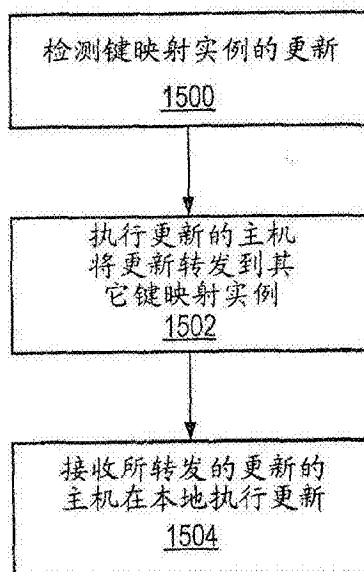


图15A

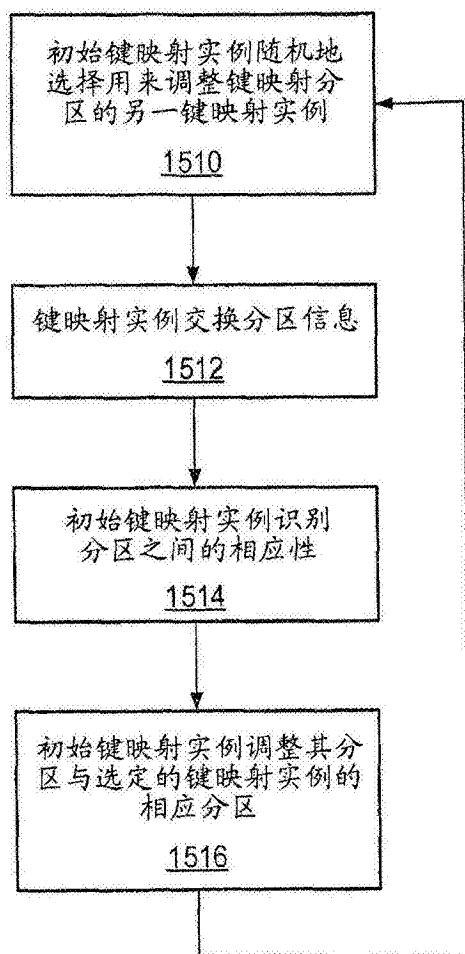


图15B

复制器键映射项目 194		
存储桶 ID 196	键 146	记录 148

图16

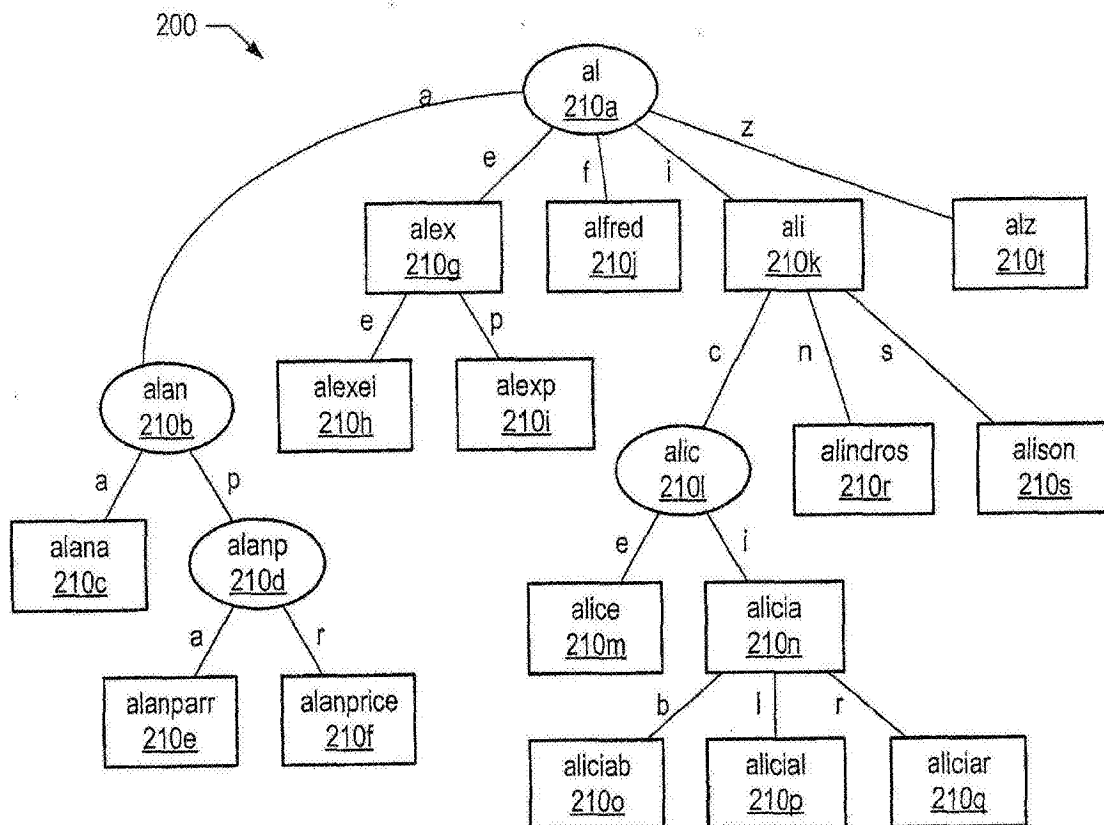


图17

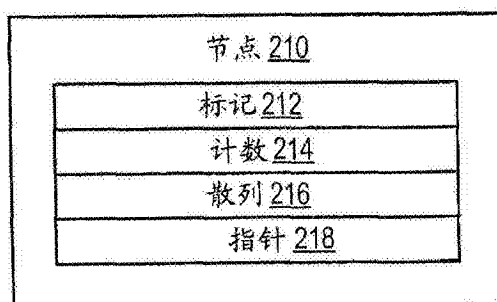


图18

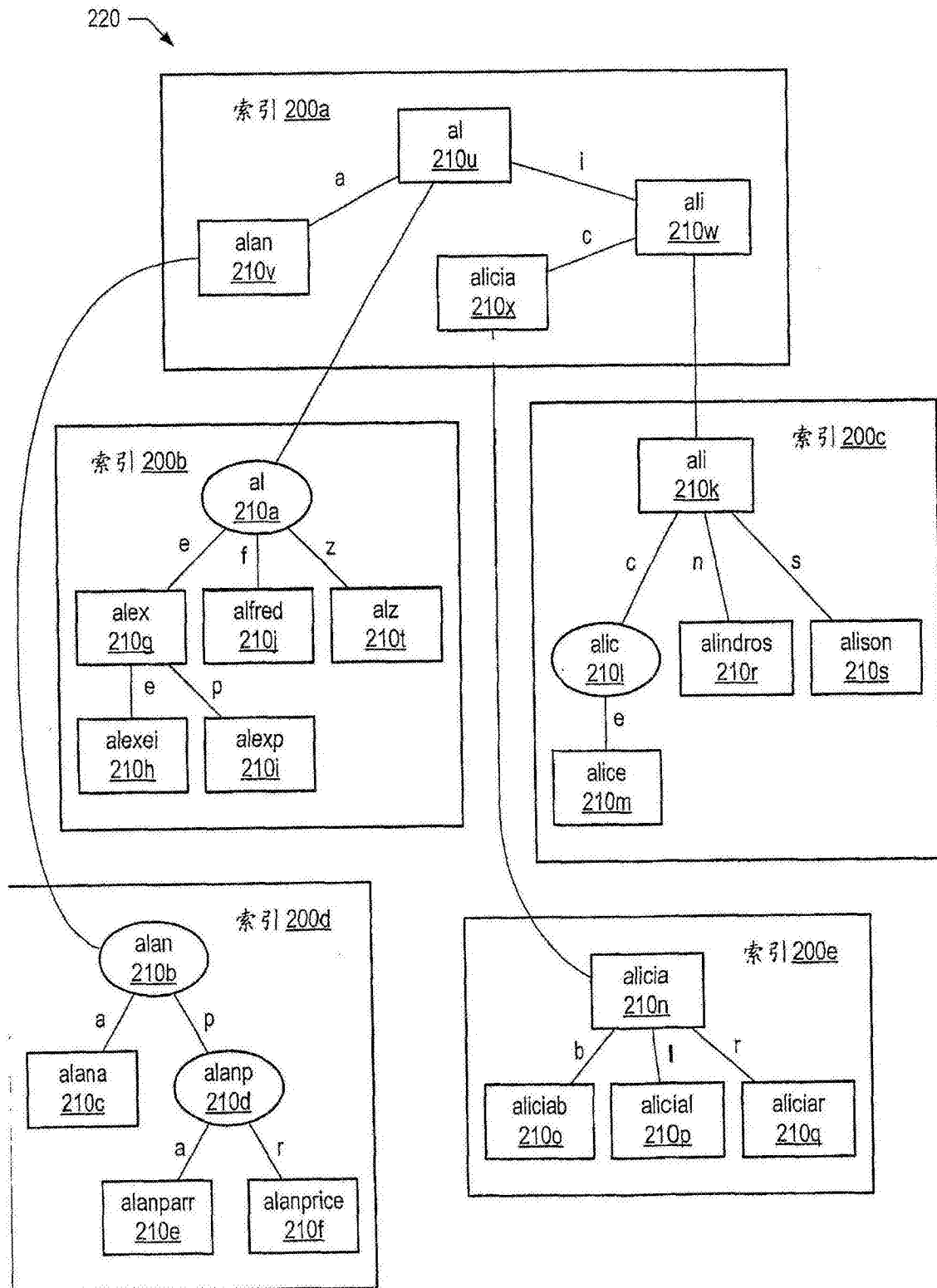


图19

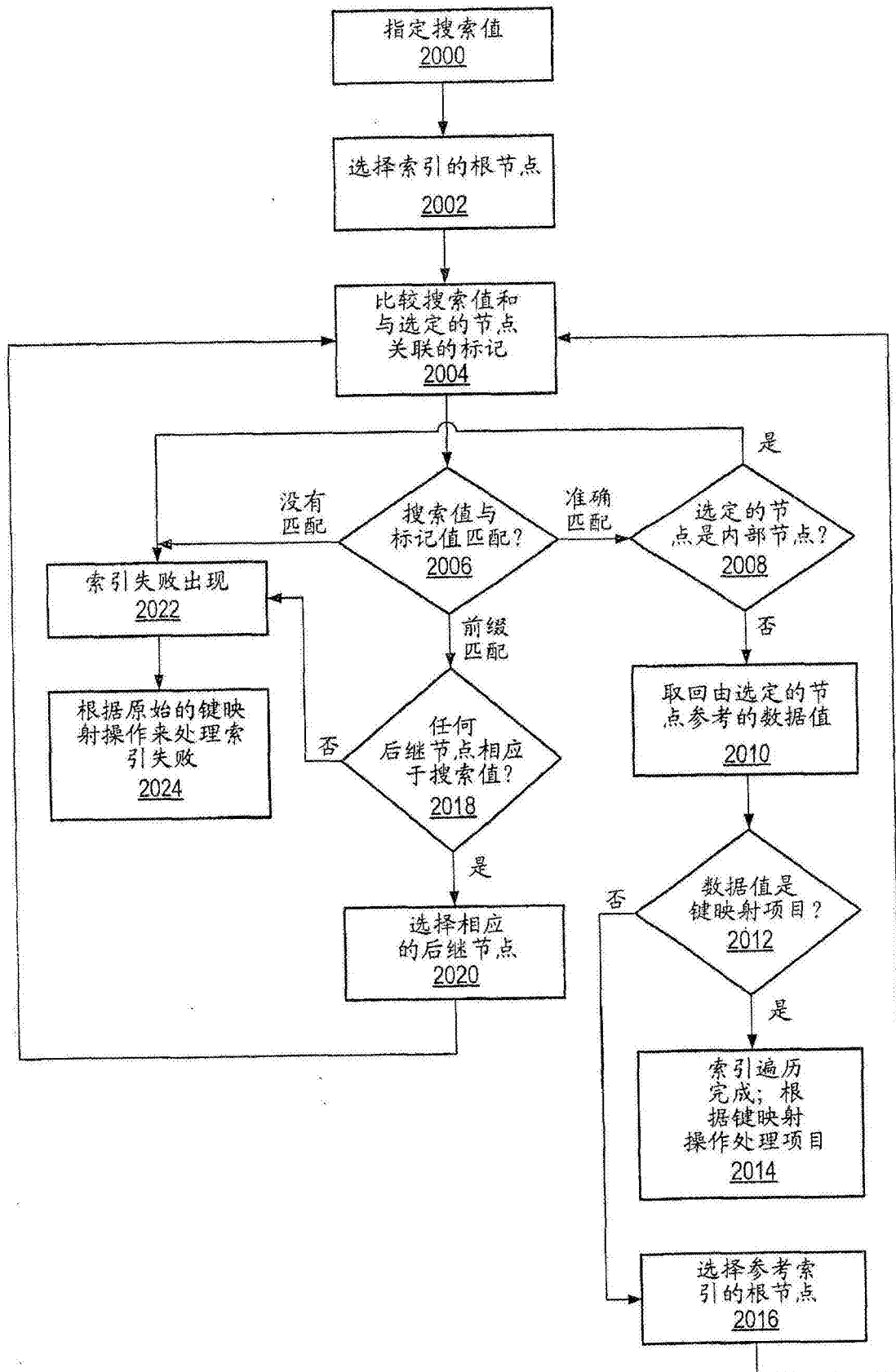


图20

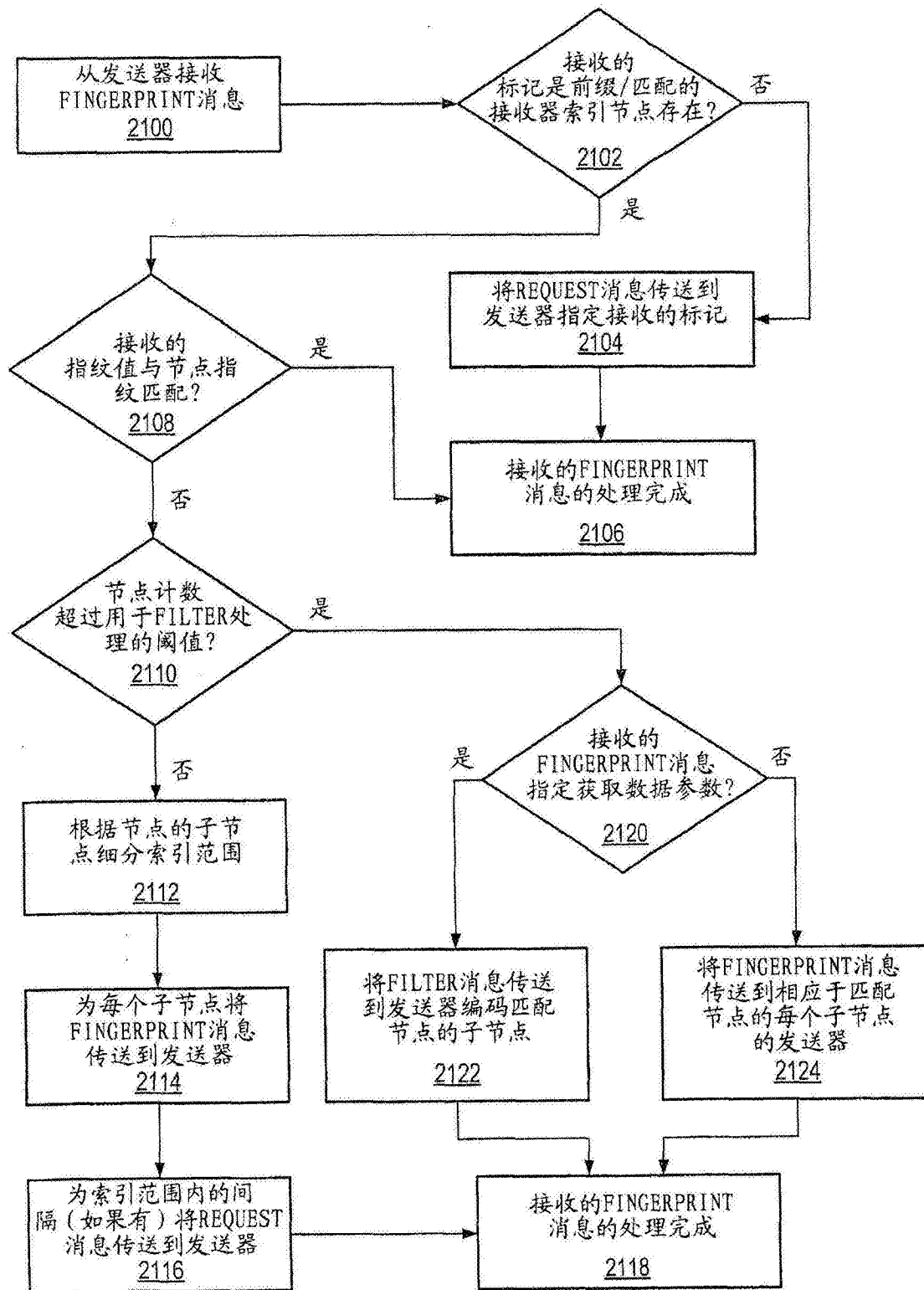


图21

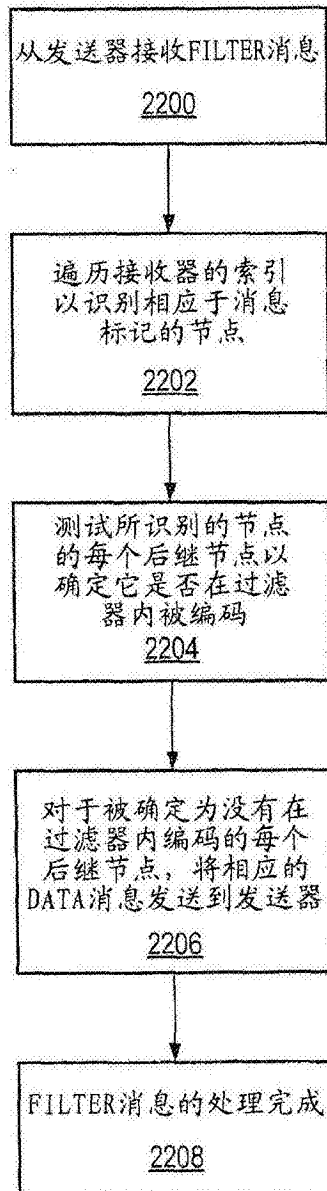


图22

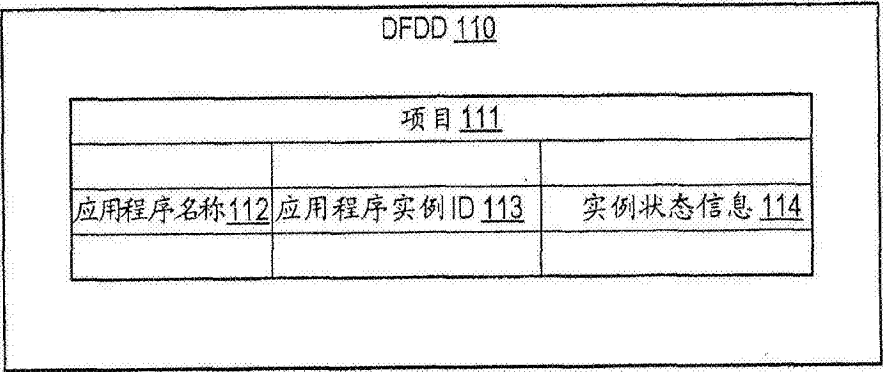


图23

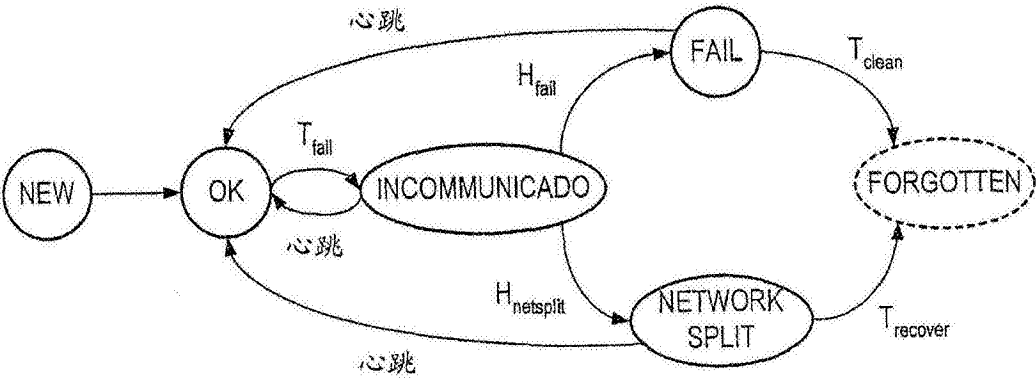


图24

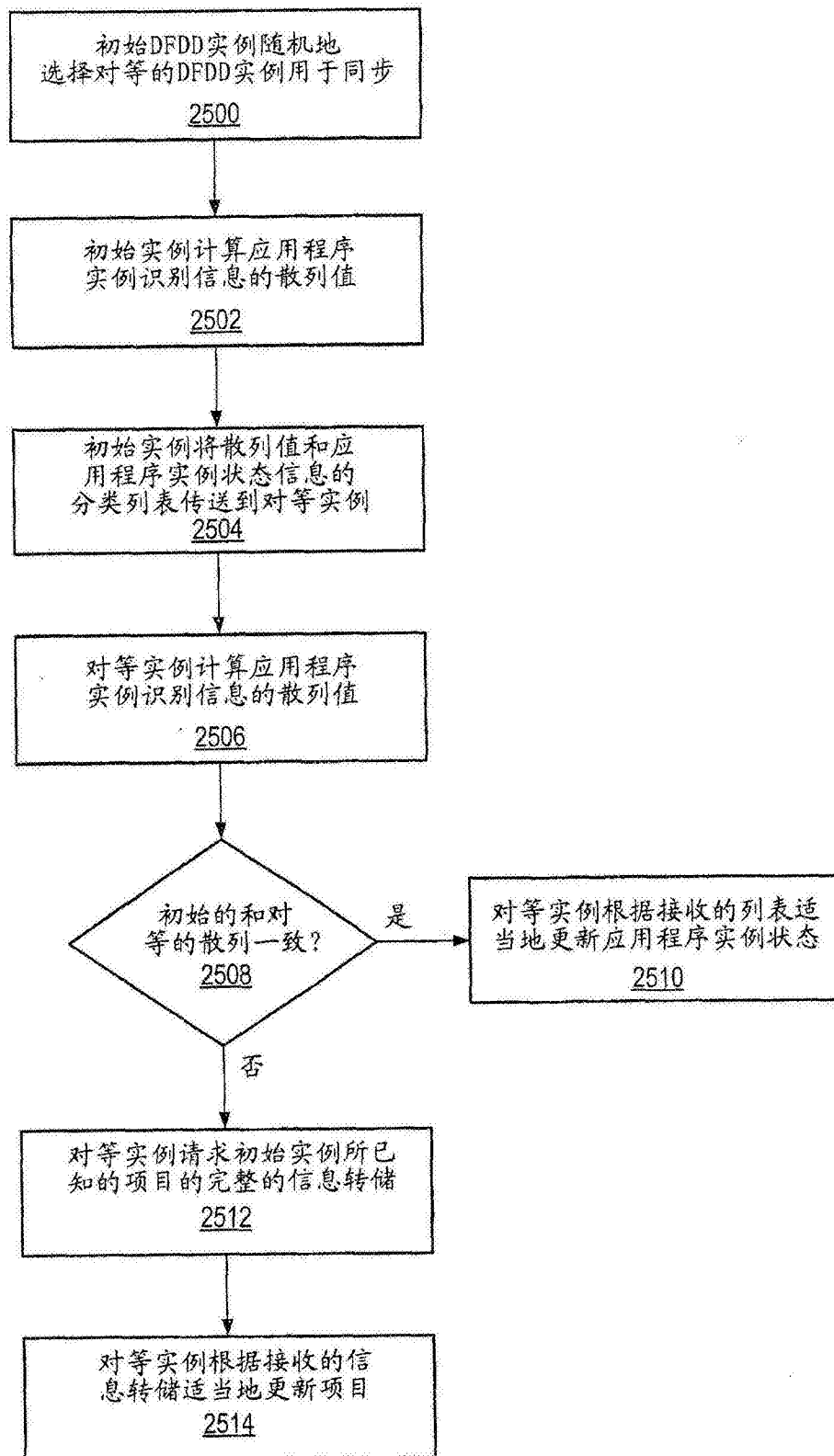


图25

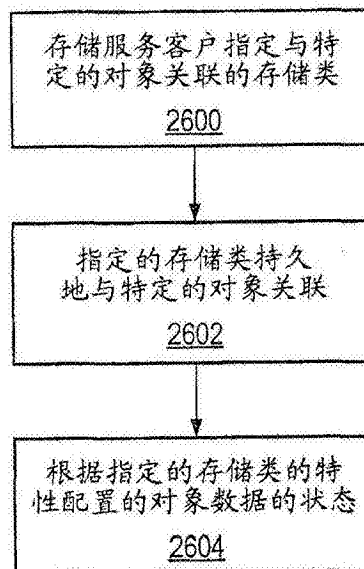


图26

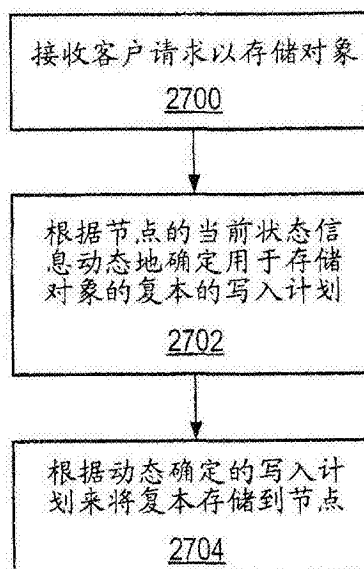


图27

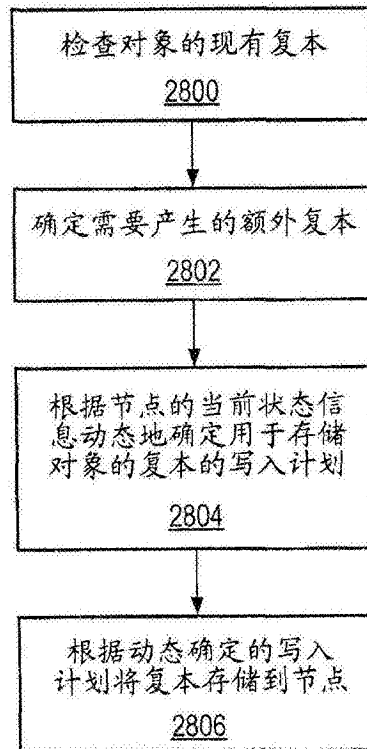


图28

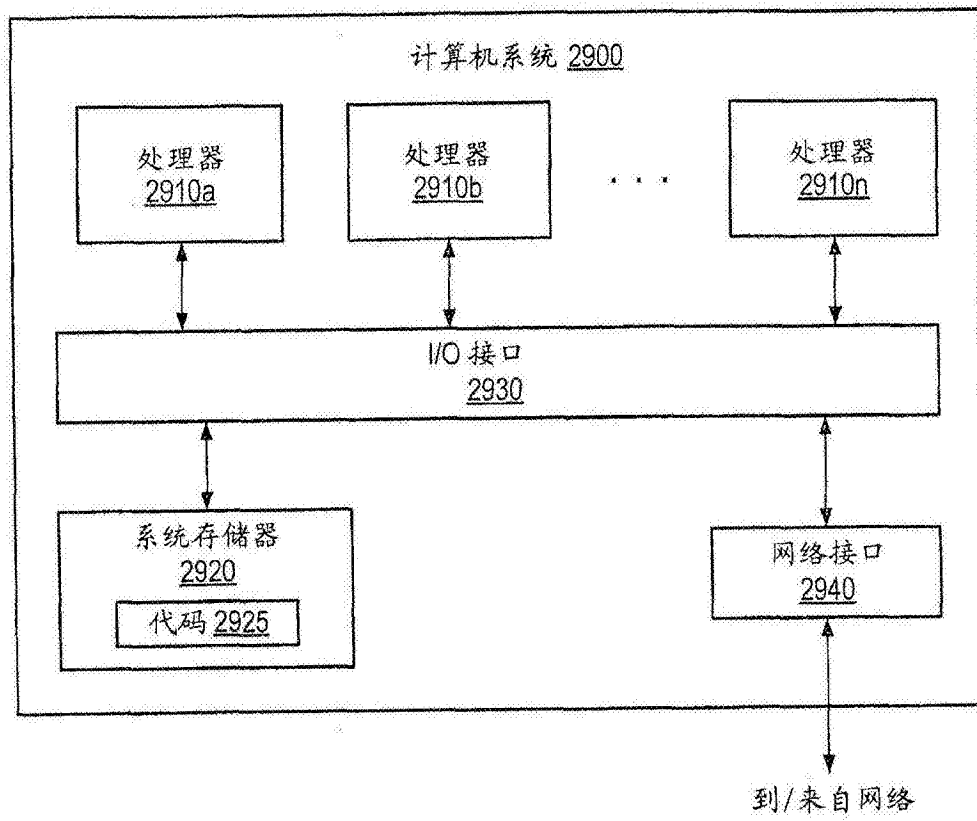


图29