(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 103930881 B
(45)授权公告日 2017.08.29

(21)申请号 201280055672.1

(22)申请日 2012.09.13

(65)同一申请的已公布的文献号
申请公布号 CN 103930881 A

(43)申请公布日 2014.07.16

(30)优先权数据
13/232,450 2011.09.14 US

(85)PCT国际申请进入国家阶段日
2014.05.13

(86)PCT国际申请的申请数据
PCT/US2012/055133 2012.09.13

(87)PCT国际申请的公布数据
WO2013/040189 EN 2013.03.21

(73)专利权人 谷歌公司
地址 美国加利福尼亚州

(72)发明人 E·V·J·艾卡斯 D·厄布

(74)专利代理机构 中原信达知识产权代理有限责任公司 11219
代理人 周亚荣 安翔

(51)Int.Cl.
G06F 15/16(2006.01)

(56)对比文件
CN 102016834 A,2011.04.13,说明书第34-68、108-114段,图1-6,15-16.
CN 102016834 A,2011.04.13,说明书第34-68、108-114段,图1-6,15-16.
US 7797276 B1,2010.09.14,说明书第40-42,61-68段.
US 6944642 B1,2005.09.13,全文.
CN 101689164 A,2010.03.31,全文.
US 7409390 B2,2008.08.05,全文.

审查员 李慧

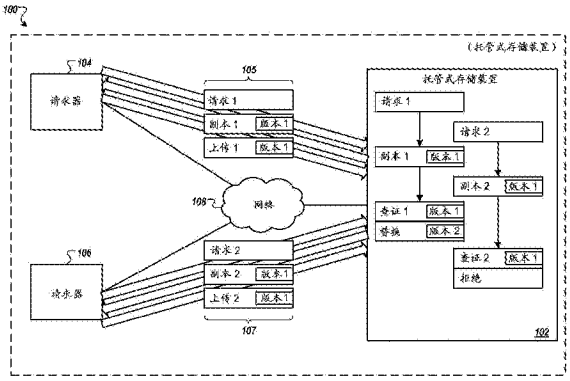
权利要求书3页 说明书20页 附图7页

(54)发明名称

托管存储锁定

(57)摘要

一种托管式存储系统从请求器接收对于访问托管的资源的请求。托管的资源包括二进制大型对象。托管式存储系统向耦合到托管式存储系统的多个客户端提供托管式存储。创建托管的资源的副本。向请求器提供包括托管的资源的副本的副本消息。在请求器修改托管的资源的副本之时在托管式存储系统中维护托管的资源。从请求器接收消息。消息包括托管的资源的修改的副本。确定自从托管的资源的副本被创建起托管的资源尚未被修改。用托管的资源的修改的副本替换托管的资源。



CN 103930881 B

1.一种由一个或者多个处理器执行的方法,所述方法包括:

在托管式存储系统从请求器接收对于访问托管的资源的请求,所述托管的资源包括二进制大型对象,所述托管式存储系统向耦合到所述托管式存储系统的多个客户端提供托管式存储,其中所述托管式存储系统包括:

数据存储库,被配置用于存储托管的资源;

元数据存储库,被配置用于存储与托管的资源有关的多个元数据字段;

创建所述托管的资源的副本;

向所述请求器提供包括所述托管的资源的所述副本的副本消息;

向所述请求器提供所述托管的资源的版本指示符,所述版本指示符包括:i)与所述托管的资源的版本关联的值,以及ii)对于与所述托管的资源关联的所述多个元数据字段中的每个元数据字段,与对应的元数据字段的版本关联的对应的值;

在所述请求器修改所述托管的资源的所述副本之时在所述托管式存储系统中维护所述托管的资源;

从所述请求器接收消息,所述消息:i)指示所述托管的资源的元数据字段的修改,并且ii)包括提供的所述版本指示符;

通过比较对应于提供的所述版本指示符的指示的所述元数据字段的值与对应于所述托管的资源的当前版本指示符的指示的所述元数据字段的当前值来确定自从所述托管的资源的所述副本被创建起所述托管的资源的指示的所述元数据字段尚未被修改;

响应于所述确定并且在所述托管式存储系统中根据从所述请求器接收的所述消息来修改所述托管的资源的指示的所述元数据字段;并且

其中来自请求器的所述请求、所述副本消息和来自所述请求器的所述消息是超文本传送协议或者超文本传送协议安全消息。

2.根据权利要求1所述的方法,其中:

修改包括递增对应于指示的所述元数据字段的所述值。

3.根据权利要求1所述的方法,其中比较对应于提供的所述版本指示符的指示的所述元数据字段的值与对应于所述托管的资源的当前版本指示符的指示的所述元数据字段的当前值包括仅比较对应于提供的所述版本指示符的指示的所述元数据字段的所述值与对应于所述托管的资源的当前版本指示符的指示的所述元数据字段的当前值。

4.根据权利要求1所述的方法,其中作为单个操作的部分执行所述确定和指示的所述元数据字段的所述修改。

5.根据权利要求1所述的方法,所述方法还包括:

在所述托管式存储系统并且在指示的所述元数据字段的所述修改之前从第二请求器接收对于访问托管的资源的第二请求;

创建所述托管的资源的第二副本;

向所述第二请求器提供所述托管的资源的所述第二副本以用于修改;

接收所述托管的资源的第二修改的副本,所述第二修改的副本由所述第二请求器创建;

确定自从所述托管的资源的所述第二副本被创建起所述托管的资源已经被修改;以及

向所述第二请求器提供修改拒绝消息。

6.根据权利要求5所述的方法，所述方法还包括：

修改所述托管的资源的一部分；并且

其中所述修改拒绝消息指示所述托管的资源的修改的一部分。

7.一种由一个或者多个处理器执行的方法，所述方法包括：

在托管式存储系统从请求器接收对于访问托管的资源的请求，所述托管式存储系统向耦合到所述托管式存储系统的多个客户端提供托管式存储，其中所述托管式存储系统包括：

数据存储库，被配置用于存储托管的资源；

元数据存储库，被配置用于存储与托管的资源有关的多个元数据字段；

创建所述托管的资源的副本；

向所述请求器提供包括所述托管的资源的所述副本的副本消息；

向所述请求器提供所述托管的资源的版本指示符，所述版本指示符包括：i)与所述托管的资源的版本关联的值，以及ii)对于与所述托管的资源关联的所述多个元数据字段中的每个元数据字段，与对应的元数据字段的版本关联的对应的值；

在所述请求器修改所述托管的资源的所述副本之时在所述托管式存储系统中维护所述托管的资源；

从所述请求器接收消息，所述消息：i)指示所述托管的资源的元数据字段的修改，并且ii)包括提供的所述版本指示符；并且

通过比较对应于提供的所述版本指示符的指示的所述元数据字段的值与对应于所述托管的资源的当前版本指示符的指示的所述元数据字段的当前值来确定自从所述托管的资源的所述副本被创建起所述托管的资源的指示的所述元数据字段尚未被修改；

响应于所述确定并且在所述托管式存储系统中根据从所述请求器接收的所述消息来修改所述托管的资源的指示的所述元数据字段；以及

响应于所述确定并且在所述托管式存储系统中递增所述版本指示符。

8.一种计算机系统，包括：

托管式存储系统，包括：

数据存储库，被配置用于存储托管的资源；

元数据存储库，被配置用于存储与托管的资源有关的多个元数据字段；

其中所述托管式存储系统被配置用于：

向耦合到所述托管式存储系统的多个客户端提供托管式存储；

从请求器接收对于访问托管的资源的请求，所述托管的资源包括二进制大型对象；

创建所述托管的资源的副本；

向所述请求器提供包括所述托管的资源的所述副本的副本消息；

向所述请求器提供所述托管的资源的版本指示符，所述版本指示符包括：i)与所述托管的资源的版本关联的值，以及ii)对于与所述托管的资源关联的所述多个元数据字段中的每个元数据字段，与对应的元数据字段的版本关联的对应的值；

在所述请求器修改所述托管的资源的所述副本之时在所述托管式存储系统中维护所述托管的资源；

从所述请求器接收消息，所述消息：i)指示所述托管的资源的元数据字段的修改，并且

ii)包括提供的所述版本指示符；

通过比较对应于提供的所述版本指示符的指示的所述元数据字段的值与对应于所述托管的资源的当前版本指示符的指示的所述元数据字段的当前值来确定自从所述托管的资源的所述副本被创建起所述托管的资源的指示的所述元数据字段尚未被修改；

响应于所述确定并且在所述托管式存储系统中根据从所述请求器接收的所述消息来修改所述托管的资源的指示的所述元数据字段；以及

请求器，被配置用于：

向所述托管式存储系统提供所述请求；

从所述托管式存储系统接收所述副本消息；

修改所述托管的资源的所述副本；并且

向所述托管式存储系统提供所述消息；并且

其中来自请求器的所述请求、所述副本消息和来自所述请求器的用于访问托管的资源的所述消息是超文本传送协议或者超文本传送协议安全消息。

9.根据权利要求8所述的系统，其中：

修改所述托管的资源的指示的所述元数据字段包括递增对应于指示的所述元数据字段的所述值。

10.根据权利要求8所述的系统，其中比较对应于提供的所述版本指示符的指示的所述元数据字段的值与对应于所述托管的资源的当前版本指示符的指示的所述元数据字段的当前值包括仅比较对应于提供的所述版本指示符的指示的所述元数据字段的所述值与对应于所述托管的资源的当前版本指示符的指示的所述元数据字段的当前值。

11.根据权利要求8所述的系统，其中作为单个操作的部分执行所述确定和指示的所述元数据字段的所述修改。

12.根据权利要求8所述的系统，其中所述托管式存储系统还被配置用于：

在所述托管式存储系统并且在指示的所述元数据字段的所述修改之前从第二请求器接收对于访问托管的资源的第二请求；

创建所述托管的资源的第二副本；

向所述第二请求器提供所述托管的资源的所述第二副本以用于修改；

接收所述托管的资源的第二修改的副本；

确定自从所述托管的资源的所述第二副本被创建起所述托管的资源已经被修改；并且

向所述第二请求器提供修改拒绝消息；并且

所述系统还包括所述第二请求器，所述第二请求器被配置用于：

向所述托管式存储系统提供所述第二请求；

从所述托管式存储系统接收所述托管的资源的所述副本；

修改所述托管的资源的所述副本；并且

向所述托管式存储系统提供所述托管的资源的所述第二修改的副本；以及

接收所述修改拒绝消息。

13.根据权利要求12所述的系统，其中所述托管式存储系统还被配置用于：

修改所述托管的资源的一部分；并且

其中所述修改拒绝消息指示所述托管的资源的修改的一部分。

托管存储锁定

[0001] 有关申请的交叉引用

[0002] 本申请要求对通过引用将其公开内容结合于此、名称为HOSTED STORAGE LOCKING、提交于2011年9月14日的第13/232,450号美国申请的优先权。

技术领域

[0003] 本文涉及托管式存储装置。

背景技术

[0004] 托管或者基于云的存储装置是指通常由第三方提供的站点外或者远程数据存储装置。第三方可以在数据中心中实施托管式存储装置，并且通过网络(比如因特网)提供对托管式存储装置的访问。

发明内容

[0005] 在一个方面中，一种方法由一个或者多个处理器执行。该方法包括在托管式存储系统从请求器接收对于访问托管的资源的请求。托管的资源包括二进制大型对象。托管式存储系统向耦合到托管式存储系统的多个客户端提供托管式存储。该方法还包括创建托管的资源的副本。该方法还包括向请求器提供包括托管的资源的副本的副本消息。该方法还包括在请求器修改托管的资源的副本之时在托管式存储系统中维护托管的资源。该方法还包括从请求器接收消息。消息包括托管的资源的修改的副本。该方法还包括确定自从托管的资源的副本被创建起托管的资源尚未被修改。该方法还包括响应于该确定并且在托管式存储系统中用托管的资源的修改的副本替换托管的资源。来自请求器的请求、副本消息和来自请求器的消息是超文本传送协议消息。

[0006] 实现方式可以包括以下特征中的任何特征、所有特征或者不包括以下特征中的任何特征。该方法还包括：提供包括向请求器提供托管的资源的版本指示符以用于修改。该方法还包括：接收托管的资源的修改的副本包括接收提供的版本指示符。该方法还包括：确定包括比较提供的版本指示符与托管的资源的当前版本指示符。该方法还包括：替换包括递增版本指示符。版本指示符包括多个值，每个值与托管的资源的版本或者关联于托管的资源的元数据值关联。比较提供的版本指示符与托管的资源的当前版本指示符包括比较提供的版本指示符的多个值的子集与当前版本指示符的多个值的子集。托管式存储系统包括：数据存储库，被配置用于存储托管的资源；元数据存储库，被配置用于存储与托管的资源有关的元数据，元数据包括访问控制列表；以及接口，被配置用于从请求器接收对于访问托管的资源的请求、向请求器提供副本消息并且从请求器接收包括托管的资源的修改的副本的消息。作为单个操作的部分执行确定和替换。该方法还包括在托管式存储系统并且在替换之前从第二请求器接收对于访问托管的资源的第二请求。该方法还包括创建托管的资源的第二副本。该方法还包括向第二请求器提供托管的资源的第二副本以用于修改。该方法还包括接收托管的资源的第二修改的副本，第二修改的副本由第二请求器创建。该方法还包

括确定自从托管的资源的第二副本被创建起托管的资源已经被修改。该方法还包括向第二请求器提供修改拒绝消息。该方法还包括修改托管的资源的一部分；并且其中修改拒绝消息指示托管的资源的修改的一部分。

[0007] 在一个方面中，一种方法由一个或者多个处理器执行。该方法包括在托管式存储系统从请求器接收对于访问托管的资源的请求，托管式存储系统向耦合到托管式存储系统的多个客户端提供托管式存储。该方法还包括创建托管的资源的副本。该方法还包括向请求器提供包括托管的资源的副本和托管的资源的版本指示符的副本消息，版本指示符包括多个值，每个值与托管的资源的版本或者关联于托管的资源的元数据值关联。该方法还包括在请求器修改托管的资源的副本之时在托管式存储系统中维护托管的资源。该方法还包括从请求器接收包括托管的资源的修改的副本和提供的版本指示符的消息。该方法还包括通过比较提供的版本指示符与托管的资源的当前版本指示符来确定自从托管的资源的副本被创建起托管的资源尚未被修改。该方法还包括响应于该确定并且在托管式存储系统中用托管的资源的修改的副本替换托管的资源。该方法还包括响应于确定并且在托管式存储系统中递增版本指示符。

[0008] 在一个方面中，一种计算机系统包括：托管式存储系统，被配置用于向耦合到托管式存储系统的多个客户端提供托管式存储。托管式存储系统还被配置用于从请求器接收对于访问托管的资源的请求。托管的资源包括二进制大型对象。托管式存储系统还被配置用于创建托管的资源的副本。托管式存储系统还被配置用于向请求器提供包括托管的资源的副本的副本消息。托管式存储系统还被配置用于在请求器修改托管的资源的副本之时在托管式存储系统中维护托管的资源。托管式存储系统还被配置用于从请求器接收包括托管的资源的修改的副本的消息。托管式存储系统还被配置用于确定自从托管的资源的副本被创建起托管的资源尚未被修改。托管式存储系统还被配置用于响应于该确定并且在托管式存储系统中用托管的资源的修改的副本替换托管的资源。该系统还包括：请求器，被配置用于向托管式存储系统提供请求。该请求器还被配置用于从托管式存储系统接收副本消息。该请求器还被配置用于修改托管的资源的副本。该请求器还被配置用于向托管式存储系统提供消息。来自请求器的请求、副本消息和来自请求器的用于访问托管的资源的消息是超文本传送协议消息。

[0009] 实现方式可以包括以下特征中的任何特征、所有特征或者不包括以下特征中的任何特征。提供副本消息包括向请求器提供托管的资源的版本指示符以用于修改。接收托管的资源的修改的副本包括接收提供的版本指示符；确定包括比较提供的版本指示符与托管的资源的当前版本指示符。替换包括递增版本指示符。版本指示符包括多个值，每个值与托管的资源的版本或者关联于托管的资源的元数据值关联。比较提供的版本指示符与托管的资源的当前版本指示符包括比较提供的版本指示符的多个值的子集与当前版本指示符的多个值的子集。托管式存储系统包括：数据存储库，被配置用于存储托管的资源；元数据存储库，被配置用于存储与托管的资源有关的元数据，元数据包括访问控制列表；以及接口，被配置用于从请求器接收对于访问托管的资源的请求、向请求器提供副本消息并且从请求器接收包括托管的资源的修改的副本的消息。作为单个操作的部分执行确定和替换。托管式存储系统还被配置用于：在托管式存储系统并且在替换之前从第二请求器接收对于访问托管的资源的第二请求；创建托管的资源的第二副本；向第二请求器提供托管的资源的副

本以用于修改；接收托管的资源的第二修改的副本；确定自从托管的资源的第二副本被创建起托管的资源已经被修改；并且向第二请求器提供修改拒绝消息。该系统还包括：第二请求器，被配置用于向托管式存储系统提供第二请求；从托管式存储系统接收托管的资源的副本；修改托管的资源的副本；并且向托管式存储系统提供托管的资源的第二修改的副本；并且接收修改拒绝消息。托管式存储系统还被配置用于修改托管的资源的一部分；并且其中修改拒绝消息指示托管的资源的修改的一部分。

[0010] 在附图和以下描述中阐述一个或者多个实现方式的细节。其它特征和优点将从描述和附图中以及从权利要求中变得清楚。

[0011] 这里描述的主题内容的各种实现方式可以提供以下优点中的一个或者多个优点。例如，在存储数据之前检查版本指示符可以防止丢失对数据的改变。使用版本指示符的托管式存储系统可以在维护数据对象的状态之时向许多客户端提供数据对象的并发访问。跟踪多个版本值可以允许系统锁定数据对象的部分而保持数据对象的其它部分未锁定。

附图说明

[0012] 图1是示出用于向托管式存储装置提供版本锁定的系统的示例的示意图。

[0013] 图2是示出用于提供托管式存储并且从客户端设备访问托管式存储装置的系统的示例的框图。

[0014] 图3是示出用于版本锁定托管的数据对象的过程的示例的泳道图。

[0015] 图4是示出用于在托管式存储服务中存储数据的过程的示例的流程图。

[0016] 图5是示出用于在托管式存储服务中提供数据的过程的示例。

[0017] 图6是示出访问控制列表的示例生命周期的流程图

[0018] 图7示出了计算设备和移动计算设备的示例。

具体实施方式

[0019] 托管式存储系统可以接收、存储和向客户端供应数据对象。在客户端请求数据对象时，与版本指示符一起返回对象。客户端可以修改对象并且与相同版本指示符一起向托管式存储系统发送回修改的对象。如果更新的数据对象的版本指示符与在托管式存储系统中存储的数据对象的版本指示符匹配，则托管式存储系统可以用上传(和修改)的数据对象替换存储的数据对象。如果版本指示符未匹配(例如，由于已经在中间时期修改了在托管式存储系统的对象)，则托管式存储系统可以拒绝上传的数据对象并且向客户端报告拒绝。

[0020] 图1是示出用于向托管式存储装置提供版本锁定的系统100的示例的示意图。系统100包括由网络108(比如因特网)连接的一个或者多个请求器104、106和托管式存储系统102。请求器可以是例如由关于图7描述的计算设备700或者移动设备750实施的客户端设备。托管式存储系统102可以使用一个或者多个服务器系统和存储系统而被实施并且可以被实施于数据中心中。托管式存储系统102的实现方式的示例是图2中所示的托管式存储服务220。可以例如使用关于图7描述的计算设备700或者移动设备750来实施托管式存储系统102。

[0021] 在系统100中，托管式存储系统102可以存储数据对象并且向应用提供对那些存储的对象的访问。应用可以在地理上相互分离的计算设备上运行。存储的数据对象可以与在

改变存储的数据对象(和/或与对象关联的元数据)时递增或者以其它方式改变的版本指示符。

[0022] 请求器104和106可以将托管式存储系统102用于数据对象存储。在一些实现方式中,请求器104和106可以具有用于读取和写入托管式存储系统中的相同数据对象的权限。例如,请求器104和106可以由用户使用,这些用户是相同用户组的部分。作为相同用户组的成员,可以授权两个用户访问托管式存储系统102中的共享的数据对象。请求器104和106可以通过网络108(比如局域网、广域网或者因特网)与托管式存储系统102通信。

[0023] 请求器104和106可以分别发送和接收消息105和107。消息105和107可以包括用于请求数据对象、接收数据对象的副本和上传数据对象的修改的副本的消息。消息105和107可以例如用来修改数据对象。在一些实现方式中,数据对象可以是托管式存储系统102中的不透明二进制大型对象(blob),并且对数据对象的编辑可以仅可能由请求器104和406进行。请求器104和106对数据对象的修改可以包括但不限于对数据对象的编辑、对对象的元数据的改变、对象的删除和用不同内容替换数据对象的内容。

[0024] 托管式存储装置102可以维护用于托管式存储装置102中的每个数据对象的版本指示符。这一版本指示符可以例如包括在修改托管式存储装置102中的数据对象时递增的数值。托管式存储装置102可以在向请求器104和106发送数据对象时与数据对象一起包括版本指示符。这一版本指示符可以由托管式存储装置102跟踪和更新并且向请求数据对象的副本的任何客户端提供。由托管式存储装置102向任何客户端提供的数据对象的任何副本可以包括数据对象的当前版本指示符。在上传消息105和107中,更新的数据对象可以包括相同版本指示符。在这一示例中,版本指示符被示出为整数值(1),但是其它类型的版本指示符是可能的。

[0025] 请求器104和106可以对来自托管式存储装置102的相同数据对象独立和异步地操作。这些操作与消息105和107一起不是瞬时的。这可以创建竞态条件,在该竞态条件中,例如请求器106在请求器104对数据对象的本地副本操作之时访问数据对象。在这一示例中,请求器104可以对请求器106无权访问的数据对象进行改变。

[0026] 托管式存储系统102可以通过仅接收数据对象的上传来控制数据对象的状态,其中上传的数据对象版本信息与存储的数据对象版本指示符匹配。在这一示例中,可以接受来自请求器104的上传的数据对象,因为它包含与存储的数据对象版本指示符(1)相同的版本指示符(1)。为了接受对象,托管式存储系统102可以用上传的数据对象替换托管的数据对象。作为相同接受的一部分,可以递增数据对象版本指示符(至2)。随后,在托管式存储装置102从请求器106接收上传消息107时,托管式存储系统102可以拒绝上传的数据对象,因为上传的数据对象的版本指示符(1)不同于托管的数据对象的版本指示符(2)。作为拒绝来自请求器106的上传的数据对象的部分,托管式存储系统102可以向请求器106发送拒绝消息(未示出)、将上传的数据对象存储为不同对象、仅接收上传的数据对象的一部分(例如,仅元数据)和/或采取另一动作。

[0027] 图2是示出用于提供托管式存储并且从客户端设备202访问托管式存储装置的系统200的示例的框图。在一些实现方式中,托管式存储服务220可以提供由在地理上相互分离的计算设备上运行的应用对存储的数据的访问、提供站点外数据备份和恢复功能、向具有有限存储能力的计算设备提供数据存储装置和/或提供未在计算设备上实施的存储功

能。

[0028] 系统200可以提供用于存储数据对象的可伸缩存储库。客户端设备202可以向托管式存储服务220上传数据对象并且控制对上传的数据对象的访问。访问控制可以包括从保持数据秘密地保密到发布它而无限制的安全级范围。可以保护在托管式存储服务220中存储的数据免于未授权访问。托管式存储服务220可以使用简单和一致应用编程接口或者API，该API可以允许任意数量的结构化或者无结构数据保持私密或者在个人、组织之间或者与全世界共享。客户端设备202可以在托管式存储服务220中存储数据以用于相互业务原因(例如，提交由托管式存储服务220的所有者订购的工作产品)，或者用于在其它服务的数据处理中使用(例如，上传的图像用来自动和动态地创建相册网页)。

[0029] 可以使用计算设备(比如关于图6描述的计算设备600或者移动设备650)来实施客户端设备202。客户端设备202可以经由网络204(比如因特网)与托管式存储服务220通信。客户端设备202可以使用通信协议(如比如传输控制协议/网际协议(TCP/IP)、超文本传送协议(HTTP)、超文本传送协议安全(HTTPS)、安全外壳远程协议(SSH)或者应用程序接口(API)中的一项或者多项跨越网络通信。尽管仅示出了单个客户端设备202，但是可以有跨越网络204与托管式存储服务220和/或其它服务和设备通信的多个客户端设备。

[0030] 可以实施托管式存储服务220以使得客户端应用(比如客户端应用203)可以存储、取回或者以其它方式操纵托管式存储服务220中的数据对象。托管式存储服务220可以由一个或者多个服务器设备实施，该一个或者多个服务器设备可以使用计算设备(比如关于图6描述的计算设备600或者移动设备650)而被实施。例如，托管式存储服务220可以由在相同或者不同数据中心中操作的多个服务器设备实施。

[0031] 托管式存储服务220一般包括接口前端206、接口后端208、存储装置后端210、用于在存储装置后端210中存储的对象的元数据216和密钥存储库209。一般而言，接口前端206可以从客户端设备202接收请求并且向客户端设备202发送响应。例如，托管式存储服务220可以被实施为具有对应Web服务应用编程接口(API)集合的Web服务。Web服务API可以例如被实施为基于表示状态传送(REST)的HTTP接口或者基于简单对象访问协议(SOAP)的接口。

[0032] 接口前端206可以从客户端202接收消息并且将请求解析成由托管式存储服务220可使用的格式，比如对接口后端208的远程过程调用(RPC)。接口前端206编写由托管式存储服务220生成的响应以用于向客户端202传输。在一些实现方式中，多个接口前端206被实施以例如支持多个访问协议。

[0033] 接口前端206可以包括例如用于在用于数据访问的web浏览器上显示的图形前端。接口前端206可以包括用于支持大型文件的管理的上传和下载(例如，用于功能(比如暂停、继续和从超时恢复))的子系统。接口前端206可以监视负荷信息并且更新日志以例如跟踪和防范服务拒绝(DOS)攻击。

[0034] 如以上描述的那样，Web服务API可以是基于REST的HTTP接口。在基于REST的接口中，数据对象被访问作为使用URI来唯一命名的资源，并且客户端应用203和服务220使用定义的操作集合来交换资源状态的表示。例如，请求的动作可以被表示为动词，比如由HTTP GET、PUT、POST、HEAD和DELETE动词表示。GET动词可以用来取回对象，而HEAD动词可以用来取回关于对象的信息而未取回对象本身。DELETE动词可以用来从托管式存储服务220删除对象。PUT和POST动词可以用来向服务220上传对象。PUT请求可以来自客户端202并且在头

部(比如HTTP头部)中包含认证和授权证书和对象元数据。可以在客户端202想要从web浏览器表单上传时接收POST请求。用于托管式存储服务220的表单POST上传协议可以涉及到用于提供认证、授权和对象元数据的多个所需表单字段。更一般而言,API请求中的任何API请求可以例如在请求的头部中包括用于认证和授权的证书。例如,可以在REST请求中包括授权头部,该授权头部包括用于标识发送请求的实体的访问密钥。

[0035] 备选地或者附加地,可以基于在向API请求追加的浏览器cookie中存储的证书认证用户。如果不存在有效cookie,则可以生成向认证前端的重定向,并且认证前端可以用来生成浏览器cookie。认证前端除了托管式存储服务220之外还可以由系统和服务使用(例如,如果操作托管式存储服务220的组织也操作其它web服务,比如电子邮件服务)。可以附加地或者备选地基于来自外部证书发放服务或者包括证书发放功能的外部服务的认证证书认证用户。可以根据外部服务的证书信息计算用户或者组标识符信息。可以翻译并且向外部服务转发由客户端202向接口前端206发送的请求以用于认证。

[0036] 一般而言,在托管式存储服务220中存储的对象可以由对象标识符引用。托管式存储服务220可以定义有效对象标识符必须符合的名称空间。例如,名称空间可以要求对象标识符是Unicode字符序列,该序列的UTF-8编码至多为1024字节长度。作为另一示例,名称空间可以要求对象标识符是全局唯一标识符(GUID),这些GUID可以是128位整数。

[0037] 可以在托管式存储服务220中在桶(bucket)中存储对象。在一些示例中,在托管式存储服务220中唯一地命名每个桶,在桶中唯一地命名每个对象,并且每个桶和对象组合是唯一的。对象可以由包括桶名称和对象名称并且标识托管式存储服务220的URI唯一地标识。例如,可以使用URI样式(比如,http://s.hostedstorage system.com/music/long/song.mp3或者http://music.s.hostedstorage system.com/long/song.mp3来指定在被命名为“music”的桶中的被命名为“long/song.mp3”的对象。备选地,客户端202的用户可以创建被命名为www.music.org的桶、发布将这重定向到http://music.s.hostedstoragesystem.com的CNAME别名并且将对象寻址为http://www.music.org/long/song.mp3。在一些示例中,桶未嵌套。

[0038] 接口后端208可以处理请求认证和授权、可以管理数据和元数据并且可以跟踪活动以比如用于开账单。接口后端208可以提供用于独立前端/后端缩放以求在局部化繁重负荷之下资源利用和响应性的功能。可以在接口后端208中封装数据管理而可以在接口前端206中封装通信服务。接口后端208可以从面向客户端的接口前端206隔离安全机制。托管的资源的版本锁定可以由接口后端207实现。

[0039] 接口后端208可以暴露可由接口前端206和其它系统二者使用的接口。在一些示例中,接口后端208的一些特征仅可由托管式存储服务220的所有者(内部用户)使用的接口前端(未示出)访问。这样的特征可以包括管理任务所需要的特征(例如,将对象引用分解成低级盘地址)。接口后端208可以处理请求认证(例如,保证用户的证书有效)和授权(例如,查证请求的操作是允许的)。接口后端也可以提供加密和解密服务以防止甚至内部用户对数据的未授权访问。

[0040] 密钥存储库209可以存储用于数据对象的加密密钥,其中加密密钥本身已经由接口后端208加密。加密的密钥可以携带在密码上限于密钥本身的关联元数据并且这里被称为打包密钥。从接口后端208的观点来看,打包密钥可以是不透明对象。为了获得打包密钥

的明文密钥以用于使用(例如用来加密或者解密数据对象)，接口后端208可以向密钥库209提供打包密钥和客户端认证证书。密钥库209可以部分基于打包密钥的元数据查证提供的认证证书足以授权释放密钥，并且如果是这样则可以向接口后端208返回未打包密钥。接口后端208可以使用密钥以加密或者解密数据对象并且然后可以丢弃密钥。

[0041] 在一些示例中，加密密钥是可以用来既加密又解密对象的对称密钥。在一些示例中，打包密钥可以具有关联元数据，该元数据指示被授权访问明文密钥的多个用户或者组。在一些示例中，密钥存储库209可以复制打包密钥并且重新打包密钥以用于不同主体(例如包含不同元数据)。

[0042] 接口后端208可以例如在结构化数据格式(比如数据库(例如，MySQL或者BigTable))中管理与数据对象关联的元数据216。可以完全在元数据216内定义标注桶的、用户指定的名称，并且对象元数据216可以将资源名称映射到存储资源的一个或者多个数据共享212。元数据216也可以包含桶和对象创建时间、对象大小、哈希以及用于桶和对象二者的访问控制列表218(ACL 218)。接口后端208可以记录活动并且跟踪存储消耗以支持用于开账单和拒付的记账。在一些示例中，这包括在对客户收费的每个维度(例如，读取、写入、网络传送、使用的总存储)中的配额监视。

[0043] ACL 218定义谁被授权对对应桶或者对象执行动作以及允许的动作的性质。ACL 218可以是{范围，作用}对的无需列表加上布尔标志。范围可以定义用户或者用户组，并且作用可以定义用于用户或者组的访问权限。在一些示例中，所有{范围，作用}对的并集可以定义访问权。在一些示例中，更具体{范围，作用}对超越更一般{范围，作用}对。下面的表1：桶作用示出可以在用于桶的ACL 218中包括的示例作用列表。下面的表2：对象作用示出可以在用于数据对象的ACL 218中包括的示例作用列表。

[0044]

作用	能力
READ	可以列举桶的内容。不能创建或者删除对象。
WRITE	读取（READ）能力加上用于创建和删除桶中的对象的能力。
FULL_CONTROL	写入(WRITE)能力加上用于读取和写入桶 ACL 的能力。

[0045] 表1：桶作用

[0046]

作用	能力
READ	可以读取对象。

[0047]

FULL_CONTROL	读取器（READER）能力加上用于读取和写入对象 ACL 的能力。

[0048] 表2:对象作用

[0049] 可以向单个用户或者用户组定义范围。在一个实现方式中,具有FULL_CONTROL作用(并且因此能够修改用于给定的桶或者对象的ACL)的那些用户可以定义用户组,并且然后提供用于组的作用。例如,用户组可以由托管式存储服务220(或者更一般地由提供托管式存储服务220的服务提供者)管理以用于除了存储权限之外的原因(例如,在groups.google.com的Google Groups),并且那些组可以由与用户组关联的单个用户名或者其它标识符、与用户组关联的电子邮件地址(可以或者可以未也对应于组的标识符)或者与组关联的域名标识。这可以允许用户指定由服务提供者管理的已经由标识符、电子邮件地址或者域名定义的预先存在的组。相似地,用户可以能够指定用户组(例如,通过用户id或者电子邮件地址)并且将访问密钥与组关联。这可以允许形成用于管理存储权限的自组织组而不是服务提供者已经管理的组。

[0050] 以这一方式,可以简单地通过管理用户组的作用向组给予特定作用。相似地,如果ACL与包含多个对象的桶关联或者ACL以其它方式与多个对象关联,则可以通过简单地改变组的作用来容易地改变关于那些对象的作用。

[0051] 下面的表3:范围示出可以在用于桶和/或数据对象的ACL 218中包括的示例范围列表。

[0052]

名称	描述
服务 ID	由用户名指定的单个认证的用户。
电子邮件地址	由电子邮件地址指定的单个用户。
服务组 ID	由托管式存储服务 220 管理并且由关联的标识符指定的用户组。

[0053]

邀请权标	一个或者多个用户有权访问一次性使用数字权标。
组限制 密钥	一个或者多个用户有权访问持久使用数字密钥。
所有服务用户	托管式存储服务 220 的所有认证的用户。
所有用户	所有用户、无认证。可以是匿名或者半匿名。

[0054] 表3:范围

[0055] FULL_CONTROL作用可以代表所有可能的能力,比如向连接到财务负责方的对象或者桶所有者指派的能力。桶所有者可以被配置用于总是具有用于桶的FULL_CONTROL。一般而言,桶和/或对象所有者可以在对应ACL中创建或者修改范围和作用,但是在一些实现方式中,可以防止从桶ACL 218(或者对象ACL)去除对{桶所有者,FULL_CONTROL}。为了创建对象,用户可以具有可以由WRITE和FULL_CONTROL授予的对桶的写入权限。对桶的WRITE权限可以意味着允许删除或者改写桶中的对象。附加约束可以不允许对ACL 218的某些修改。在一些示例中,有可能创建桶所有者不能读取的对象。

[0056] 存储装置后端210可以包含多个数据存储库212a-212c。虽然示出了三个数据存储库212,但是更多或者更少是可能的。数据存储库212a-212c中的每个数据存储库可以用特定格式存储数据对象214a-214c。例如,数据存储库212a可以将数据对象214a存储为二进制大型对象(BLOB),数据存储库212b可以在分布式文件系统(例如,网络文件系统、Google文件系统)中存储数据对象214b,并且数据存储库212c可以在结构化数据格式(比如数据库(例如,MySQL、BigTable))中存储数据对象214c。

[0057] 接口后端208可以通过强制实行资源的版本锁定来保证对托管的资源(比如数据对象214、元数据216和ACL 218)的安全修改。版本锁定可以适用于由在托管式存储系统220内的单元(比如接口后端208)或者外部单元(比如客户端应用203)对托管的资源的访问。

[0058] 接口后端可以在元数据216中存储用于每个数据对象214的版本指示符。这一版本指示符可以反映数据对象214的当前版本并且可以在改变数据对象214或者它的关联的元数据216或者ACL 218时由接口后端208管理。在单元(比如客户端应用202)请求数据对象214(例如,用GET HTTPS请求)时,接口后端208可以作为响应而提供版本指示符以及数据对象214。

[0059] 在对于向数据对象214写入的请求由接口后端208接收(例如,PUT HTTPS请求)时,接口后端208可以比较请求中的版本指示符与在元数据216中存储的版本指示符。如果版本指示符匹配,则接口后端208可以处理写入请求。如果它们未匹配,则接口后端208可以拒绝请求并且向单元发送具有用于拒绝的原因的消息。

[0060] 用来确定版本指示符匹配的过程可以基于版本指示符的格式。版本指示符的一个格式可以是任何时间修改数据对象214或者关联的元数据216都被递增的单个整数。在这一情况下,可以比较两个整数。版本指示符的另一格式可以包括整数值的矢量。矢量的每个元素可以对应于数据对象214或者单个元数据216字段。在接收对于写入数据对象214或者元数据216的请求时,接口后端208可以仅比较版本指示符的与请求的数据对象214或者元数据216字段对应的矢量元素。在其它备选中,版本指示符可以包含非整数值,比如字节串或者ascii值。

[0061] 例如,托管式存储系统102可以具有用于对象大小、对象哈希、ACL 218和mime型字段的元数据216字段。关联的版本指示符可以是五维矢量:[对象版本,大小版本,哈希版本,ACL版本,mime型版本]。接口后端208可以接收对于仅向mime类型元数据216字段写入的包括示例版本指示符的写入请求。在这一情况下,接口后端208可以仅比较接收的版本指示符的“mime型版本”维度与存储的版本指示符的“mime型版本”维度。如果那两个维度匹配,则即使其它维度未匹配,接口后端208仍然可以接受和处理接收的写入请求。

[0062] 图3是示出了用于版本锁定托管的数据对象的过程300的示例的泳道图。过程300可以例如由托管式存储系统220以及请求器301和302执行，并且为了呈现清楚，以下描述使用系统200作为用于描述过程的基础。另一系统或者系统组合可以用来执行过程400。在这一示例中，请求器104是尝试改变数据对象的内容的外部客户端202，并且请求器106是托管式存储系统220的尝试改变相同数据对象的ACL 218的内部部件。在其它示例中，请求器104和106可以均在内部或者二者可以是外部客户端。

[0063] 请求器301生成对于托管的资源的请求(303)。例如，请求器301可以在用户指引下标识待访问以用于编辑的数据对象214，比如托管式文档。请求器301可以创建请求数据对象214的副本的消息(比如HTTPS GET或者HEAD消息)，并且该请求可以包括用户的认证信息。托管式存储系统220从请求器301接收对于访问托管的资源的请求(304)。托管式存储系统220向耦合到托管式存储系统的多个客户端提供托管式存储。例如，托管式存储系统220可以接收和认证请求。

[0064] 托管式存储系统220创建托管的资源的副本。例如，托管式存储系统220可以从数据存储库212读取出数据对象214的副本并且解密数据对象214。托管式存储系统220向请求器301提供托管的资源的副本。例如，托管式存储系统220可以向请求器301发送具有数据对象214的副本的消息。托管式存储系统220向请求器301提供具有托管的资源的版本指示符。例如，可以与数据对象214关联地在元数据216中存储数据对象214的版本指示符。可以在向请求器301的消息中包括这一版本指示符。

[0065] 在一些实现方式中，向请求器301的消息可以是对请求数据对象214的GET或者HEAD消息的HTTPS响应。HTTPS响应可以包括在这一示例中为x-sequence-header的定制HTTP头部，该定制HTTP头部包括版本指示符，以下示出该定制HTTP头部的示例。

[0066] HTTP/1.1 200 OK

[0067] Content-Type:application/xml;charset=UTF-8

[0068] Date:Fri,22Jul 2011 16:27:56GMT

[0069] Expires:Fri,22Jul 2011 16:27:56GMT

[0070] Cache-Control:private,max-age=0

[0071] x-sequence-header:2

[0072] X-Frame-Options:SAMEORIGIN

[0073] header:Content-Length:2475

[0074] 托管式存储系统220在请求器301修改托管的资源的副本之时维护托管的资源(306)。例如，托管式存储系统220可以无需在复制和发送数据对象214或者版本指示符的过程中修改、移动或者删除数据对象214或者版本指示符。然而，在一些实现方式中，可以执行一些修改、移动或者删除。

[0075] 请求器301从托管式存储系统220接收托管的资源并且编辑托管的资源(308)。例如，请求器301可以接收数据对象214、解析并且向用户显示数据对象214为文档。用户可以编辑文档，并且指令请求器301保存文档。

[0076] 请求器302生成对于访问托管的资源的第二请求(310)。例如，在用户编辑文档之时，请求器302可以创建对于改变用于数据对象214(包括由用户编辑的数据对象214)的组的ACL 218的请求。从第二请求器接收对于访问托管的资源的第二请求(312)。

[0077] 托管式存储系统220创建托管的资源的副本，并且托管式存储系统220向请求器302提供托管的资源的副本。例如，托管式存储系统220可以从数据存储库212读取出数据对象214的副本并且解密数据对象214。托管式存储系统220然后可以向请求器301发送具有数据对象214的副本的消息。托管式存储系统220向请求器302提供具有托管的资源版本指示符。例如，可以与数据对象214关联地在元数据216中存储数据对象214的版本指示符并且可以取回并且在向请求器302的消息中包括这一版本指示符。在一些实现方式中，这可以有效地创建与在304中发送的消息相同的消息。

[0078] 托管式存储系统220从请求器301接收托管的资源的修改的副本(314)。例如，请求器301可以响应于用户的用于保存文档的指令与请求器301在308中接收的版本指示符一起向托管式存储系统220发送具有修改的数据对象214的消息。托管式存储系统220从请求器301接收托管的资源的修改的副本的版本指示符。例如，托管式存储系统220可以解析在来自请求器301的消息中的版本指示符。托管式存储系统220确定自从托管的资源的副本被创建起托管的资源是否已经被修改。例如，托管式存储系统220可以比较来自请求器301的版本指示符与在用于数据对象214的元数据216中的版本指示符。如果版本指示符匹配，则托管式存储系统220可以确定自从数据对象214的副本被创建起数据对象214的托管的副本尚未被修改。

[0079] 在一些实现方式中，来自请求器301的消息可以是HTTPS PUT、POST或者DELETE消息。这一消息可以包括在这一示例中为x-if-sequence-number-match的定制HTTP头部，该定制HTTP头部包括来自请求器301的版本指示符，以下示出该定制HTTP头部的示例。

[0080] PUT/sampleput/xxx HTTP/1.1

[0081] Host:datarepository.example.com

[0082] Accept-Encoding:identity

[0083] Content-MD5:filbWD4p/PTGCwYwaCa+Mg＝＝

[0084] Content-Length:15

[0085] x-api-version:2

[0086] Content-Type:application/octet-stream

[0087] x-if-sequence-match:3

[0088] User-Agent:Agentname

[0089] 版本指示符指示多个值，每个值与托管的资源的版本或者关联于托管的资源的元数据值关联。例如，版本指示符可以包括整数或者其它值的数组。在数组中的每个条目可以对应于数据对象214的不同特征、方面或者部分。下面的表4:版本数组示出了可以在托管式存储系统220中使用的示例版本数组。第一行列举描述数据对象214的方面的标签，并且第二行输出示例当前版本。

[0090]

不透明对象	ACL	元数据字段1	元数据字段2
12	3	128	4

[0091] 表4:版本数组

[0092] 托管式存储系统220比较托管的资源的修改的副本的版本指示符与托管的资源的托管的副本的版本指示符。例如，如果版本指示符是整数或者整数数组，则托管式存储系统

220可以比较整数或者整数数组以确定它们是否具有相同值。比较版本指示符包括比较提供的版本指示符的多个值的子集与当前版本指示符的多个值的子集。例如，x-if-sequence-number-match头部字段可以指示仅修改数据对象214的一部分(例如，在文档示例中的Opaque对象)，并且仅可以比较版本数组的该部分。托管式存储系统220可以忽略版本数组的其它部分。也就是说，如果元数据字段1已经改变，但是Opaque对象字段尚未改变，则托管式存储系统220可以确定出于上传的数据对象214的目的而尚未修改存储的数据对象214。

[0093] 托管式存储系统220用托管的资源的修改的副本替换托管的资源(316)。例如，托管式存储系统220可以在数据存储库212中存储上传的数据对象214，并且数据对象214的全限定名称(例如，桶/名称)现在可以分解成上传的数据对象214。在一些实现方式中，可以删除、存档或者重新命名数据对象214的更旧副本。例如，托管式存储系统220可以重新命名旧数据对象214，并且可以在相同桶中保存它。在这一情况下，全限定名称的名称部分可以不同。在另一示例中，旧数据对象214可以移向不同桶。在这一情况下，全限定名称的桶部分可以不同。

[0094] 托管式存储系统220递增托管的资源的版本指示符。例如，对于作为单个整数的版本指示符，整数值被增加1。在其中版本指示符包括整数矢量的另一示例中，托管式存储系统220根据改变的数据对象214的特征、方面或者部分递增一个或者多个整数。

[0095] 托管式存储系统220作为单个操作的部分执行确定和替换。例如，一旦托管式存储系统220确定数据对象214尚未被修改，则阻止该数据对象214除了替换之外的任何类型的修改或者访问直至替换已经被完成。可以从请求器301和302的观点将确定和替换描述为原子性的。

[0096] 托管式存储系统220接收和编辑托管的资源(318)。例如，请求器302可以从托管式存储系统220接收具有与数据对象214关联的ACL 218的消息。请求器302可以编辑ACL 218，并且向托管式存储系统220发送回ACL 218和版本指示符。这可以例如在314和/或316期间或者之后发生。

[0097] 托管式存储系统220接收托管的资源的第二修改的副本(320)。例如，托管式存储系统220可以从请求器302接收具有修改的ACL 218的消息。托管式存储系统220确定自从托管的资源的副本被创建起托管的资源已经被修改。例如，托管式存储系统220比较来自请求器302的版本指示符或者版本指示符的一部分与在元数据216中存储的版本指示符，并且托管式存储系统220确定版本指示符不同。

[0098] 托管式存储系统220修改托管的资源的一部分(322)。例如，如果来自请求器302的版本指示符和元数据216的版本指示符仅一个字段(例如，Opaque字段)不同，但是来自请求器302的消息指示请求的改变仅涉及到数据对象214的不同特征(例如，ACL)，则可以改变指示的部分。在这一情况下，可以如请求器302请求的那样修改ACL 218。在其中来自请求器302的消息指示对数据对象214的两个特征(例如，Opaque对象和ACL)的改变但是那些字段之一不同于存储的版本指示符(例如，Opaque对象)的另一示例中，仅可以改变数据对象214的不同的特征(例如，ACL)。

[0099] 托管式存储系统220向请求器302提供拒绝消息。例如，托管式存储系统220可以生成指示数据对象214的当前版本的拒绝消息并且向请求器302发送该消息。拒绝消息指示托

管的资源的修改的版本。例如，如果在322中改变了数据对象214的一部分并且一部分被拒绝，则可以在消息中包括改变或者拒绝的部分的指示。请求器302接收拒绝消息(324)。

[0100] 图4是示出了用于在托管式存储服务中存储数据对象的过程400的示例的流程图。过程400可以例如由接口前端206和接口后端210执行，并且为了呈现清楚，以下描述使用系统200作为用于描述过程的基础。然而，另一系统或者系统的组合可以用来执行过程400。

[0101] 由接口前端206从客户端应用203接收对于存储对象的请求(402)。请求可以包括HTTP PUT或者POST请求、认证进行请求的主体(实体)的认证证书、数据对象以及由桶和数据对象名称构成的用于对象的目标。在一些示例中，认证证书可以包括可互操作认证头部、权标或者cookie。接口前端可以向后端208产生包括请求头部的RPC。

[0102] 接口后端208可以检查和验证认证证书(404)。例如，可以使用托管式存储服务220的内部验证特征来验证本机证书(例如，用户ID、权标)。可以向关联的外部系统发送外部证书(例如，用于另一系统的用户名和口令)以用于验证。

[0103] 接口后端208可以查询请求的目标桶的ACL 218以确定是否允许认证的主体创建桶中的对象(406)。例如，主体或者主体为其成员的组可以具有在桶的ACL 218中指派的WRITE或者FULL_CONTROL作用，该作用将允许主体创建桶中的对象。如果未授权主体创建对象，则请求被拒绝。

[0104] 否则，接口后端208向数据存储库212的具有目标数据对象名称的目标桶上传对象(408)。在一些示例中，每个桶仅与单个数据存储库212关联，并且指定目标桶指定数据存储库212。在一些示例中，接口后端208可以检查数据对象或者使用来自接口前端206的RPC中的参数以确定在哪个数据存储库212中存储对象，其中关联的元数据216指示对象的位置(也就是在其中存储对象的特定数据存储库和对象在该数据存储库中的位置)。

[0105] 接口后端208重新验证主体的认证和授权(410)。为了支持长久运行上传，可以暂时忽略认证证书的到期时间。

[0106] 接口后端208向密钥存储库209生成对于用于新上传的对象的打包密钥的新文档密钥请求(412)。新打包密钥被绑定到请求的主体的认证证书。密钥存储库209也可以供应用于加密打包密钥的加密密钥。接口后端208加密打包密钥(414)。

[0107] 接口后端208创建代表用于新创建的对象的访问控制列表的ACL 218(416)。在一些示例中，使用默认ACL 218或者可以按照来自客户端202的请求来指定ACL 218。

[0108] 接口后端208向密钥存储库209生成对于ACLK 218中的每个主体(用户或者组)的打包密钥的新文档密钥请求，该主体具有用于读取对象或者修改对象的ACL 218的权限。每个新打包密钥被绑定到单个主体。密钥存储库209也可以供应用于加密打包密钥的加密密钥，或者接口后端208可以重用来自步骤208的加密密钥。

[0109] 接口后端208在对象的元数据216中存储对象的ACL 218和打包密钥。接口后端208可以丢弃一个或者多个加密密钥。

[0110] 图5是示出了用于在托管式存储服务中提供数据对象的过程的示例的流程图。过程500可以例如由接口前端206和接口后端210执行，并且为了呈现清楚，以下描述使用系统200作为用于描述该过程的基础。然而，另一系统或者系统组合可以用来执行过程500。

[0111] 由接口前端206从客户端应用203接收对于下载对象的请求(502)。请求可以包括HTTP GET请求、认证进行请求的主体(实体)的认证证书和由桶(和可选地由数据对象)名称

构成的目标。在一些示例中，认证证书可以包括可互操作认证头部、权标或者cookie。接口前端可以向后端208产生包括请求头部的RPC。

[0112] 接口后端208检查和验证在请求中包括的认证证书(504)。例如，可以使用托管式存储服务220的内部验证特征来验证本机证书(例如，用户ID、权标)。可以向关联的外部系统发送外部证书(例如，用于另一系统的用户名和口令)以用于验证。

[0113] 接口后端208查询请求的桶或者对象ACL 218以确定是否允许认证的主体读取目标(506)。例如，主体或者主体为其成员的组可以具有指派的READ、WRITE或者FULL_CONTROL作用，该作用将允许主体读取或者以其它方式访问目标。如果未授权主体读取或者访问对象，则请求被拒绝。

[0114] 否则，接口后端208确定请求是否对于桶或者对于对象(508)。如果请求是对于桶，则接口后端208查询桶的内容列表(510)，并且向客户端应用203返回列表(512)。

[0115] 如果请求是对于对象，则接口后端208从对象的元数据216查找用于给定的认证的请求器的适当打包密钥(514)。接口后端208向密钥存储库209发送打包密钥和认证证书，密钥存储库209可以向接口后端208返回解密的对象加密密钥(516)。接口后端208可以取读和解密目标对象(518)以向客户端应用203返回(520)。

[0116] 图6是示出了ACL 218的示例生命周期600的流程图。虽然生命周期600的步骤示出了步骤顺序，但是并未暗示每个步骤直接通向另一步骤。所示的步骤是可以按照大致时间顺序对ACL 218执行的可能的步骤的列表。步骤的实际顺序、数目和种类将依赖于托管式存储系统320的实施细节和使用。

[0117] 由后端接口208基于来自客户端应用203的请求来创建桶或者对象(602)。客户端请求可以包括桶名称、对象名称和/或ACL 218。认证请求新桶或者对象的主体并且使该主体成为桶或者对象的所有者。

[0118] 如果在请求中指定了ACL 218(604)，则指定的ACL 218与桶或者对象关联。如果指定的ACL 218并不包括指定具有FULL_CONTROL的主体的{范围，作用}对，则可以向ACL 218添加一个这样的{范围，作用}对。在一个实现方式中，可以通过枚举将在ACL中包括的每个范围和作用对在请求中指定ACL，或者可以通过引用预先指定的或者“固定的(canned)”ACL的名称来指定ACL。在下面的表5固定ACL中示出了预先指定的或者‘固定的’ACL 218的列表。固定的ACL 218可以由托管式存储系统220编目录并且由客户端应用203按名称引用，而不是要求请求枚举每个范围和作用对。

[0119]

固定的 ACL 名称	{范围，权限}
private	{创建用户或者桶所有者， FULL_CONTROL}
public-read	{所有用户，READ} {桶所有者，FULL_CONTROL}
public-read-write	{所有用户，WRITE} {桶所有者，FULL_CONTROL}
authenticated-read	{所有认证的用户，READ} {桶所有者，FULL_CONTROL}
bucket-owner-read [for objects only]	{桶所有者，READ} {对象所有者，FULL_CONTROL}
bucket-owner-full-contr ol [for objects only]	{桶所有者，FULL_CONTROL} {对象所有者，FULL_ CONTROL}

[0120] 表5固定的ACL

[0121] 如果在请求中未指定ACL 218(604)，则可以使用默认ACL 218(608)。例如，桶和对象创建可以默认为用于认证的用户的‘私有(private)’固定的ACL 218。对于未授权用户的对象创建(比如对于在“public-read-write”桶中创建的新对象)，可以使用默认“bucket-owner-full-control”。

[0122] 可以例如向具有关联的桶或者对象的READ、WRITE或者FULL_CONTROL的主体供应ACL 218(610)。例如，客户端应用203可以用查询串?acl向目标的URI执行HTTP GET以取回与目标关联的ACL。ACL 218可以被串行化并且向客户端应用203返回。

[0123] 串行化可以例如由以下扩展的Backus-Naur形式定义。非终结符在斜体sans serif中出现，终结符在等宽粗体中出现，{}表示零个或者更多个重复，[]包围可选实体，|分离备选，并且()表示分组。在以下英文中定义终结符号canonical-id、email-address和domain：

[0124] access-control-list:<AccessControlList>owner entries</AccessControlList>

[0125] owner:<Owner>id</Owner>

[0126] entries:<Entries>entry{entry}</Entries>

[0127] entry:<Entry>(permission scope|scope permission)</Entry>

[0128] permission:<Permission>(READ|WRITE|FULL_CONTROL)</Permission>

[0129] scope:<Scope type＝UserById>id</Scope>

[0130] |<Scope type＝UserByEmail>email</scope>

[0131] |<Scope type＝GroupById>id</Scope>

[0132] |<Scope type＝GroupByEmail>email</Scope>

[0133] |<Scope type＝GroupByDomain><Domain>domain</Domain></Scope>

[0134] |<Scope type＝AllUsers/>

[0135] |<Scope type＝AllAuthenticatedUsers/>

[0136] id:<ID>canonical-id</ID>[<Name>text</Name>]

[0137] |[<Name>text</Name>]<ID>canonical-id</ID>

[0138] email:<EmailAddress>email-address</EmailAddress>[<Name>text</Name>]

[0139] |[<Name>text</Name>]<EmailAddress>canonical-id</EmailAddress>

[0140] text:{printable character excluding<and>}

[0141] canonical-id:64hex digits

[0142] email-address:standard RFC 822 email address

[0143] domain:standard RFC 822 domain specification

[0144] canonical-id或者email-address可以标识用户或者组。canonical-id是用于用户或者组的加密的服务id。电子邮件地址便于指定规范(canonical)id。在一些实现方式中，从系统返回的ACL总是包含规范id。<Name>text</Name>元素可以仅用于信息目的，从而使得它被系统以其它方式忽略，并且如果没有与用户或者组关联的公共名称，则系统未返回它。

[0145] 以下示出示例串行化ACL 218。

[0146] <AccessControlList>

[0147] <Owner>

[0148] <ID>a9a7b886d6fd24a52fe8ca5bef65f89a64e0193f23000e241bf9b1c61be666e9</ID>

[0149] <Name>chriscustomer</Name>

[0150] </Owner>

[0151] <Entries>

[0152] <Entry><Permission>FULL_CONTROL</Permission>

[0153] <Scope type＝UserById>

[0154] <ID>a9a7b886d6fd24a52fe8ca5bef65f89a64e0193f23000e241bf9b1c61be666e9</ID>

[0155] <Name>chriscustomer</Name>

[0156] </Scope>

[0157] </Entry>

[0158] <Entry><Permission>FULL_CONTROL</Permission>

[0159] <Scope type＝UserById>

[0160] <ID>79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be</ID>

[0161] <Name>Frank</Name>

[0162] </Scope>

[0163] </Entry>

[0164] <Entry><Permission>FULL_CONTROL</Permission>

[0165] <Scope type＝UserById>

[0166] <ID>de019164ebb0724ff67188e243eae9ccbebdde523717cc312255d9a82498e394a</ID>

[0167] <Name>Jose</Name>

[0168] </Scope>

[0169] </Entry>

[0170] <Entry><Permission>READ</Permission><Scope type＝AllUsers></Entry>

[0171] </Entries>

[0172] </AccessControlList>

[0173] ACL 218可以例如由具有关联的桶或者对象的WRITE或者FULL_CONTROL的主体更新(612)。在一些示例中，客户端必须读取、修改和写入ACL 218以便更新ACL 218。在这一示例中，ACL 218被供应(610)作为修改的部分(612)。在一些实现方式中，客户端应用203可以向托管式存储系统220发送ACL更新请求。

[0174] 图7示出可以用来实施这里描述的技术的计算设备700和移动计算设备750。计算设备700旨在于代表各种形式的数字计算机，比如膝上型计算机、台式计算机、工作站、个人数字助理、服务器、刀片服务器、大型机和其它适当计算机。移动计算设备750旨在于代表各种形式的移动设备，比如个人数字助理、蜂窝电话、智能电话和其它相似计算设备。这里所示的部件、它们的连接和关系及其功能意味着仅为示例而不意味着限制。

[0175] 计算设备700包括处理器702、存储器704、存储设备706、连接到存储器704和多个高速扩展端口710的高速接口708以及连接到低速扩展端口714和存储设备706的低速接口712。使用各种总线来互连并且可以在公共母板上装配或者在适当时以其它方式装配处理器702、存储器704、存储设备706、高速接口708、高速扩展端口710和低速接口712中的每个部件。处理器702可以处理用于在计算设备700内执行的指令，这些指令包括存储器704中或者在存储设备706上存储的用于在外部输入/输出设备(比如耦合到高速接口708的显示器716)上显示用于GUI的图形信息的指令。在其它实现方式中，多个处理器和/或多个总线可以在适当时与多个存储器和存储器类型一起使用。也可以连接多个计算设备，其中每个设备提供必需操作的部分(例如，作为服务器组、成组刀片服务器或者多处理器系统)。

[0176] 存储器704存储计算设备700内的信息。在一些实现方式中，存储器704是一个或者多个易失性存储器单元。在一些实现方式中，存储器704是一个或者多个非易失性存储器单元。存储器704也可以是另一形式的计算机可读介质，比如磁盘或者光盘。

[0177] 存储设备706能够提供用于计算设备700的海量存储装置。在一些实现方式中，存储设备706可以是或者包含计算机可读介质，比如软盘设备、硬盘设备、光盘设备或者带设备、闪存或者其它相似固态存储器设备或者如下设备的阵列，这些设备包括存储区域网络或者其它配置中的设备。可以在信息载体中存储指令。指令在由一个或者多个处理设备(例如，处理器702)执行时执行一种或者多种方法，比如以上描述的方法。指令也可以由一个或

者多个存储设备(比如计算机或者机器可读介质(例如,存储器704、存储设备706或者在处理器702上的存储器))存储。

[0178] 高速接口708管理用于计算设备700的带宽密集操作,而低速接口712管理带宽密集度更低的操作。这样的功能分配仅为示例。在一些实现方式中,高速接口708耦合到存储器704、显示器716(例如,通过图形处理器或者加速器)并且耦合到可以接受各种扩展卡(未示出)的高速扩展端口710。在该实现方式中,低速接口712耦合到存储设备706和低速扩展端口714。可以包括各种通信端口(例如,USB、蓝牙、以太网、无线以太网)的低速扩展端口714可以例如通过网络适配器耦合到一个或者多个输入/输出设备(比如键盘、指点设备、扫描仪)或者联网设备(比如交换机或者路由器)。

[0179] 如该图中所示,可以在多个不同形式中实施计算设备700。例如,它可以被实施为标准服务器720或者多次实施于成组这样的服务器中。此外,可以在个人计算机(比如膝上型计算机722)中实施它。它也可以被实施为架式服务器系统724的部分。备选地,来自计算设备700的部件可以与移动设备(未示出,比如移动计算设备750)中的其它部件组合。这样的设备中的每个设备可以包含计算设备700和移动计算设备750中的一个或者多个设备,并且整个系统可以由相互通信的多个计算设备组成。

[0180] 移动计算设备750包括处理器752、存储器764、输入/输出设备(比如显示器754)、通信接口766和收发器768以及其它部件移动计算设备750也可以具有用于提供附加存储的存储设备,比如微驱动或者其它设备。使用各种总线来互连部件处理器752、存储器764、显示器754、通信接口766和收发器768中的每个部件,并且可以在公共母板上或者在适当时以其它方式装配部件中的若干部件。

[0181] 处理器752可以执行移动计算设备750内的指令,这些指令包括在存储器764中存储的指令。处理器752可以被实施为包括分离以及多个模拟和数字处理器的芯片的芯片集。处理器752可以例如提供协调移动计算设备750的其它部件(比如控制用户界面)、由移动计算设备750运行的应用和移动计算设备750的无线通信。

[0182] 处理器752可以通过控制接口758和耦合到显示器754的显示接口756与用户通信。显示器754可以例如是TFT LCD(薄膜晶体管液晶显示器)显示器或者OLED(有机发光二极管)显示器或者其它适当显示器技术。显示接口756可以包括用于驱动显示器754以向用户呈现图形和其它信息的适当电路。控制接口758可以从用户接收命令并且转换它们以用于向处理器752提交。此外,外部接口762可以提供与处理器752通信以便支持移动计算设备750与其它设备的近区域通信。外部接口762可以例如在一些实现方式中提供有线通信或者在其它实现方式中提供无线通信,并且也可以使用多个接口。

[0183] 存储器764存储移动计算设备750内的信息。存储器764可以被实施为一个或者多个计算机可读介质、一个或者多个易失性存储器单元或者一个或者多个非易失性存储器单元中的一项或者多项。也可以提供扩展存储器774并且扩展存储器774可以通过可以例如包括SIMM(单列直插存储器模块)卡接口的扩展接口772连接到移动计算设备750。扩展存储器774可以提供用于移动计算设备750的额外存储空间或者也可以存储用于移动计算设备750的应用或者其它信息。具体而言,扩展存储器774可以包括用于执行或者补充以上描述的过程的指令并且也可以包括安全信息。因此,例如,扩展存储器774可以被提供作为用于移动计算设备750的安全模块并且可以用允许安全使用移动计算设备750的指令对扩展存储器

774进行编程。此外，可以经由SIMM卡将安全应用与附加信息一起提供，比如以不可黑客的方式将标识信息放置于SIMM卡上。

[0184] 如以下讨论的那样，存储器可以例如包括闪存和/或NVRAM存储器(非易失性随机存取存储器)。在一些实现方式中，在信息载体中存储指令。指令在由一个或者多个处理设备(例如，处理器752)执行时执行一种或者多种方法，比如以上描述的方法。指令也可以由一个或者多个存储设备(比如一个或者多个计算机或者机器可读介质(例如，存储器764、扩展存储器774或者在处理器752上的存储器))存储。在一些实现方式中，可以例如通过收发器768或者外部接口762在传播的信号中接收指令。

[0185] 移动计算设备750可以通过可以在必要时包括数字信号处理电路的通信接口766无线通信。通信接口766可以在各种模式或者协议(比如GSM语音呼叫(全球移动通信系统)、SMS(短消息服务)、EMS(增强型消息服务)或者MMS消息接发(多媒体消息接发服务)、CDMA(码分多址)、TDMA(时分多址)、PDC(个人数字蜂窝)、WCDMA(宽带码分多址)、CDMA2000或者GPRS(通用分组无线电服务)以及其它模式或者协议)之下提供通信。可以例如通过使用射频的射频收发器768发生这样的通信。此外，可以比如使用蓝牙、WiFi或者其它这样的收发器(未示出)发生近程通信。此外，GPS(全球定位系统)接收器模块770可以向移动计算设备750提供附加的与导航和位置有关的无线数据，该数据可在适当时由在移动计算设备750上运行的应用使用。

[0186] 移动计算设备750也可以使用音频编码解码器760来可听通信，音频编码解码器760可以从用户接收口述信息并且将它转换成可用数字信息。音频编码解码器760可以类似地比如通过例如在移动计算设备750的头戴式受话器中的扬声器生成用于用户的可听声音。这样的声音可以包括来自语音电话呼叫的声音、可以包括记录的声音(例如，语音消息、音乐文件等)并且也可以包括由在移动计算设备750上操作的应用所生成的声音。

[0187] 如该图中所示，可以用多个不同形式实施移动计算设备750。例如，它可以被实施为蜂窝电话780。它也可以被实施为智能电话782、个人数字助理或者其它相似移动设备的部分。

[0188] 可以在数字电子电路、集成电路、特殊设计的ASIC(专用集成电路)、计算机硬件、固件、软件和/或其组合中实现这里描述的系统和技术的各种实现方式。这些各种实现方式可以包括在可编程系统上可执行和/或可解译的一个或者多个计算机程序中的实现方式，该可编程系统包括可以是特殊或者通用的至少一个可编程处理器、至少一个输入设备和至少一个输出设备，该可编程处理器被耦合用于从存储系统接收数据和指令以及向存储系统传输数据和指令。

[0189] 这些计算机程序(也被称为程序、软件、软件应用或者代码)包括用于可编程处理器的机器指令并且可以用高级过程和/或面向对象的编程语言和/或用汇编/机器语言来实施。如这里所用，术语机器可读介质和计算机可读介质指代用来向可编程处理器提供机器指令和/或数据的任何计算机程序产品、装置和/或设备(例如，磁盘、光盘、存储器、可编程逻辑器件(PLD))，包括接收机器指令为机器可读信号的机器可读介质。术语机器可读信号指代用来向可编程处理器提供机器指令和/或数据的任何信号。

[0190] 为了提供与用户的交互，可以在计算机上实施这里描述的系统和技术，该计算机具有用于向用户显示信息的显示设备(例如，CRT(阴极射线管)或者LCD(液晶显示器)监视

器)以及用户可以用来向计算机提供输入的键盘和指点设备(例如,鼠标或者跟踪球)。其它种类的设备也可以用来提供与用户的交互;例如,向用户提供的反馈可以是任何形式的感官反馈(例如,可视反馈、听觉反馈或者触觉反馈);并且可以用包括声音、话音或者触觉输入的任何形式接收来自用户的输入。

[0191] 可以在计算系统中实施这里描述的系统和技术,该计算系统包括后端部件(例如,作为数据服务器)或者包括中间件部件(例如,应用服务器)或者包括前端部件(例如,具有图形用户界面或者Web浏览器的客户端计算机,用户可以通过该图形用户界面或者Web浏览器与这里描述的系统和技术的实现方式交互)或者这样的后端、中间件或者前端部件的任何组合。系统的部件可以由任何数字数据通信形式或者介质(例如,通信网络)互连。通信网络的示例包括局域网("LAN")、广域网("WAN")和因特网。

[0192] 计算系统可以包括客户端和服务器。客户端和服务器一般相互远离并且通常通过通信网络交互。借助在相应计算机上运行并且相互具有客户端-服务器关系的计算机程序产生客户端和服务器关系。

[0193] 虽然以上已经具体描述了少数实现方式,但是其它修改是可能的。例如,尽管客户端应用被描述为访问委托,但是在其它实现方式中,委托可以由一个或者多个处理器实施的其它应用(比如在一个或者多个服务器上执行的应用)运用。此外,在各图中描绘的逻辑流程可以无需所示特定顺序或者依次顺序以实现希望的结果。此外,可以提供其它动作或者可以从描述的流程消除动作,并且可以向描述的系统添加或者从描述的系统去除其它部件。因而,其它实现方式在所附权利要求的范围内。

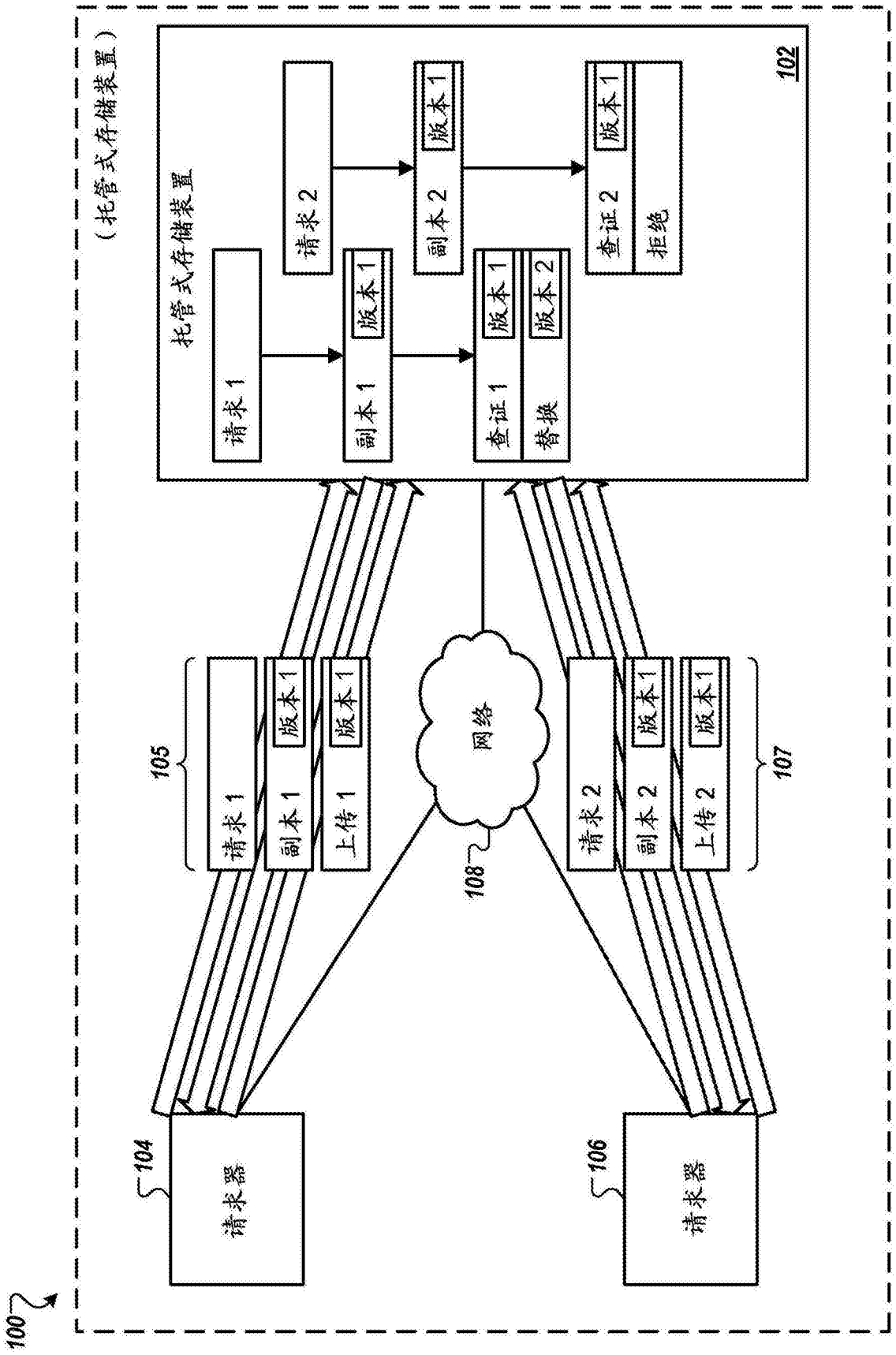

100
（托管式存储装置）
托管式存储装置
102
请求 1
副本 1
版本 1
查证 1
替换
版本 1
版本 2
请求 2
副本 2
版本 1
查证 2
拒绝
版本 1
105
请求 1
副本 1
版本 1
上传 1
版本 1
网络
108
请求 2
副本 2
版本 1
上传 2
版本 1
107
104
请求器
106
请求器

图1

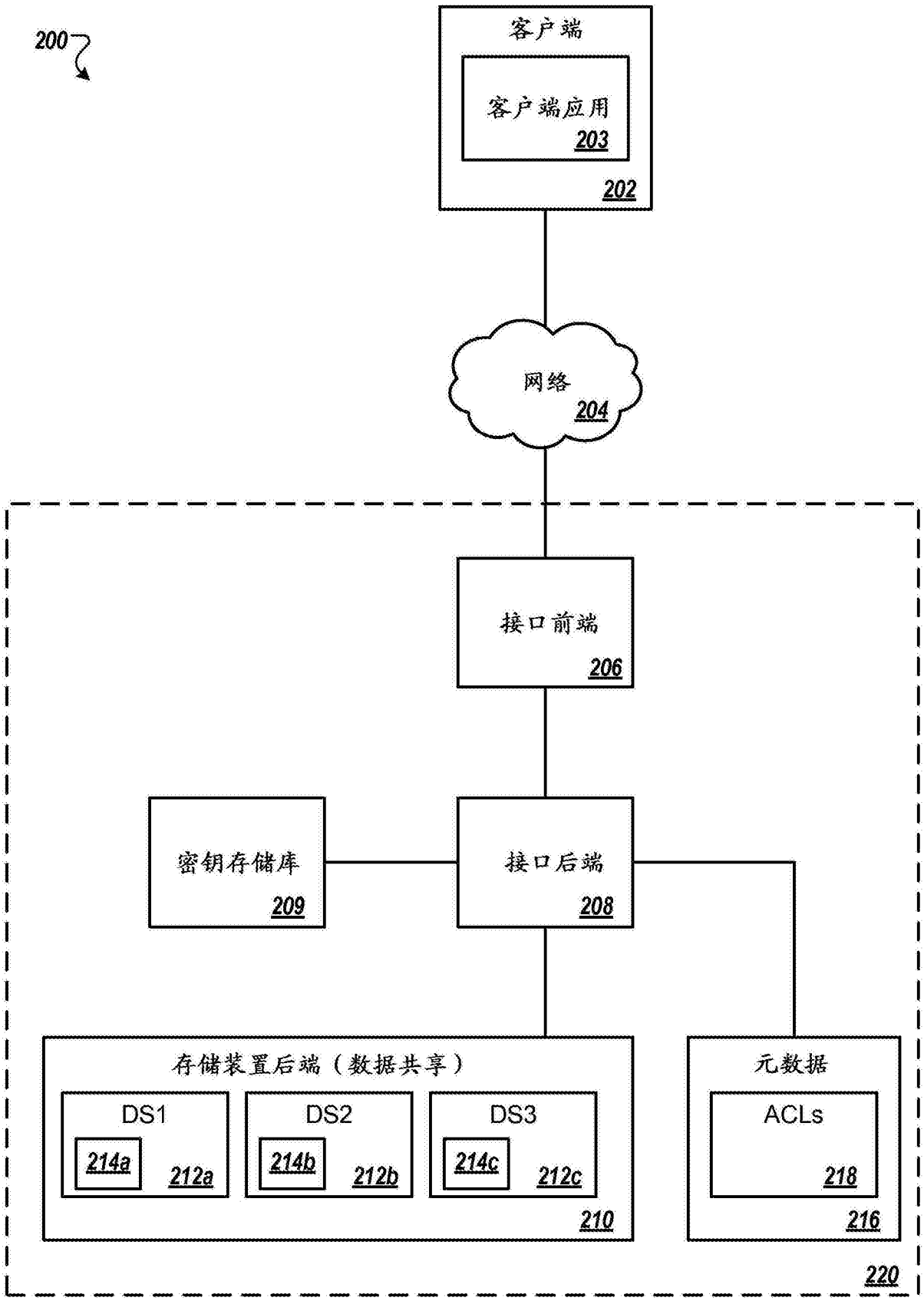

200
客户端
客户端应用
203
202
网络
204
接口前端
206
密钥存储库
209
接口后端
208
存储装置后端（数据共享）
DS1
214a
212a
DS2
214b
212b
DS3
214c
212c
210
元数据
ACLs
218
216
220

图2

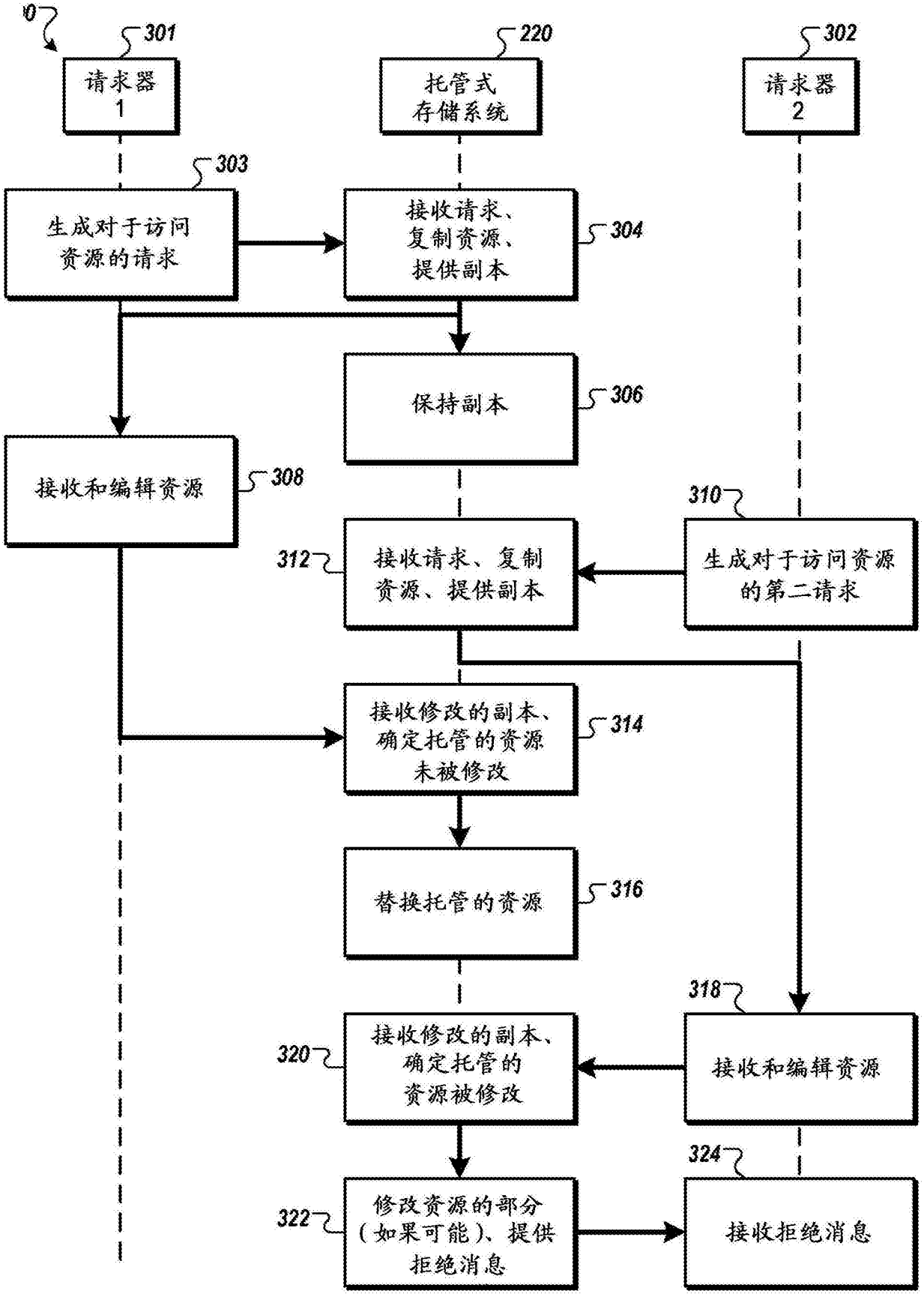

0
301
请求器 1
220
托管式存储系统
302
请求器 2
303
生成对于访问资源的请求
接收请求、复制资源、提供副本
304
保持副本
306
接收和编辑资源
308
310
生成对于访问资源的第二请求
312
接收请求、复制资源、提供副本
接收修改的副本、确定托管的资源未被修改
314
替换托管的资源
316
318
接收和编辑资源
320
接收修改的副本、确定托管的资源被修改
324
接收拒绝消息
322
修改资源的部分（如果可能）、提供拒绝消息

图3

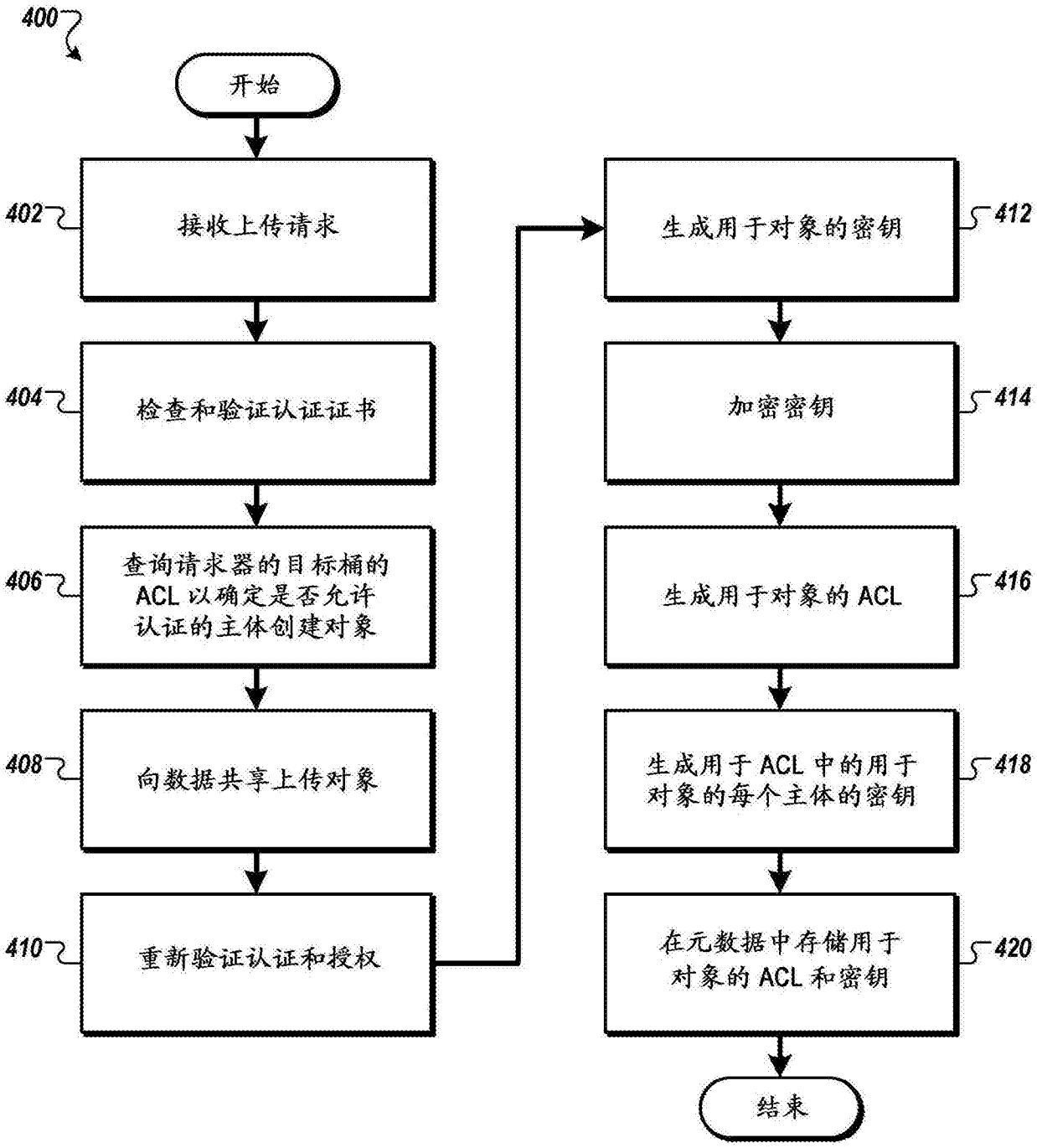

400
开始
402
接收上传请求
404
检查和验证认证证书
406
查询请求器的目标桶的ACL以确定是否允许认证的主体创建对象
408
向数据共享上传对象
410
重新验证认证和授权
412
生成用于对象的密钥
414
加密密钥
416
生成用于对象的ACL
418
生成用于ACL中的用于对象的每个主体的密钥
420
在元数据中存储用于对象的ACL和密钥
结束

图4

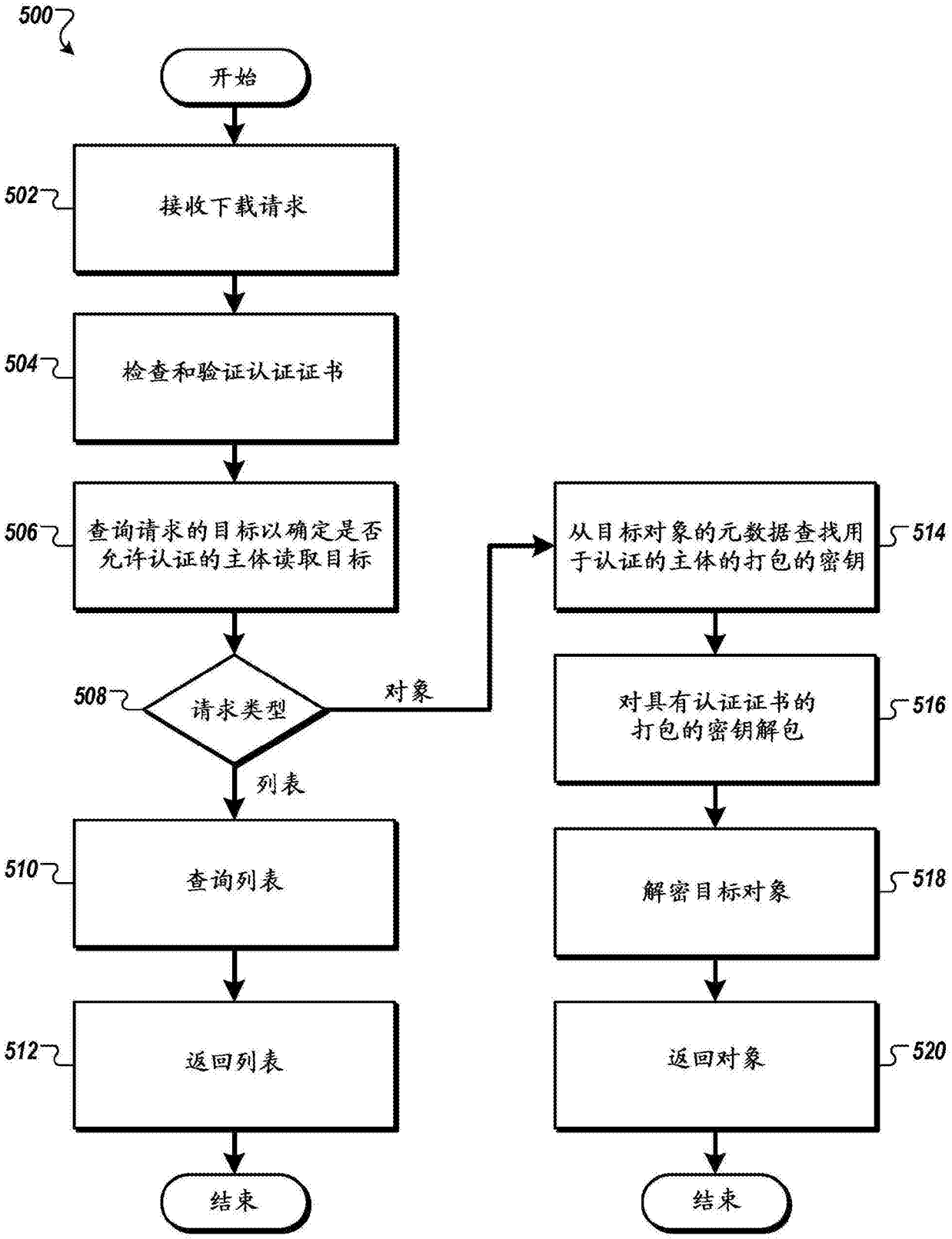

500
开始
502
接收下载请求
504
检查和验证认证证书
506
查询请求的目标以确定是否允许认证的主体读取目标
508
请求类型
对象
列表
510
查询列表
512
返回列表
结束
514
从目标对象的元数据查找用于认证的主体的打包的密钥
516
对具有认证证书的打包的密钥解包
518
解密目标对象
520
返回对象
结束

图5

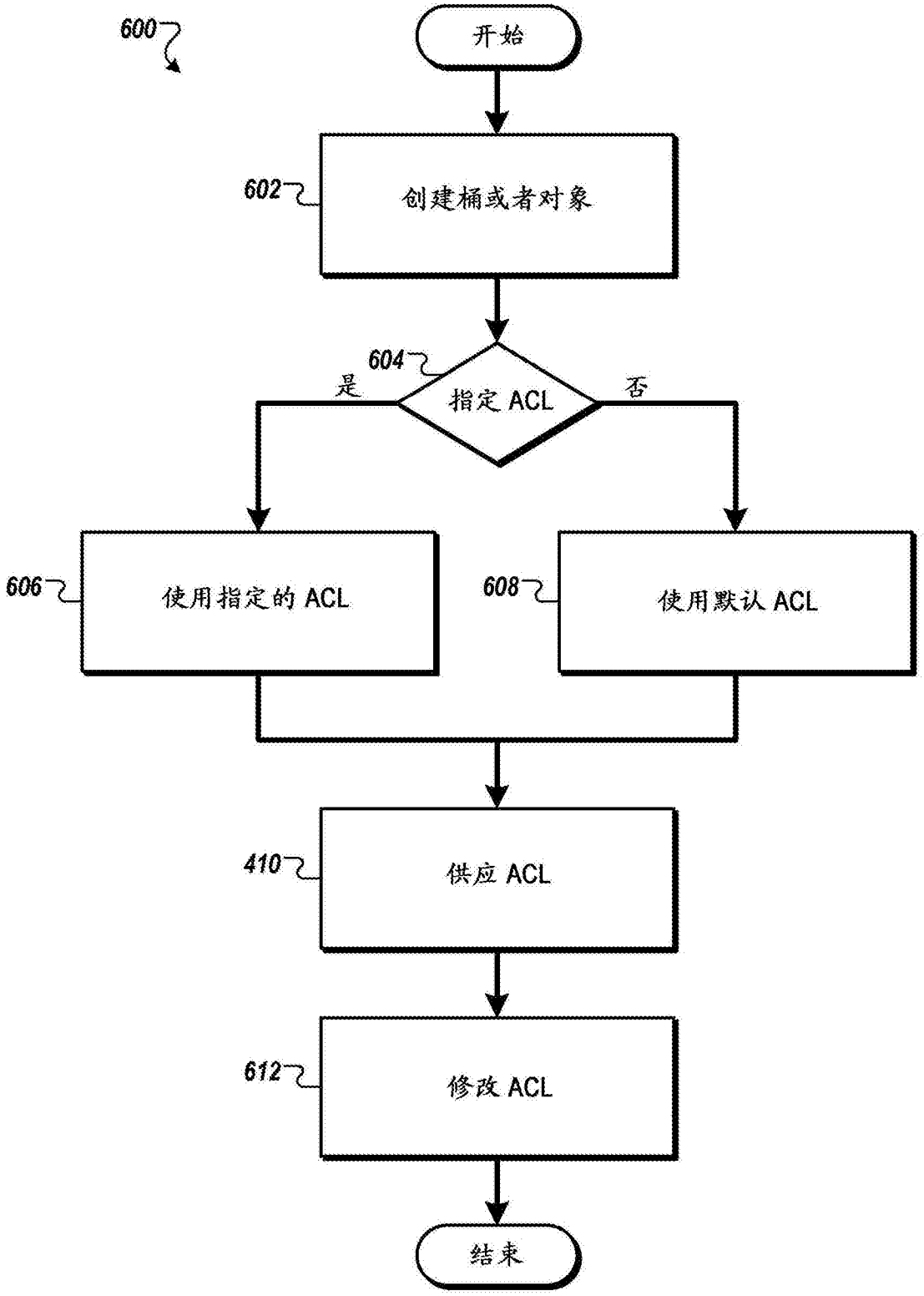

600
开始
602
创建桶或者对象
604
指定 ACL
是
否
606
使用指定的 ACL
608
使用默认 ACL
410
供应 ACL
612
修改 ACL
结束

图6

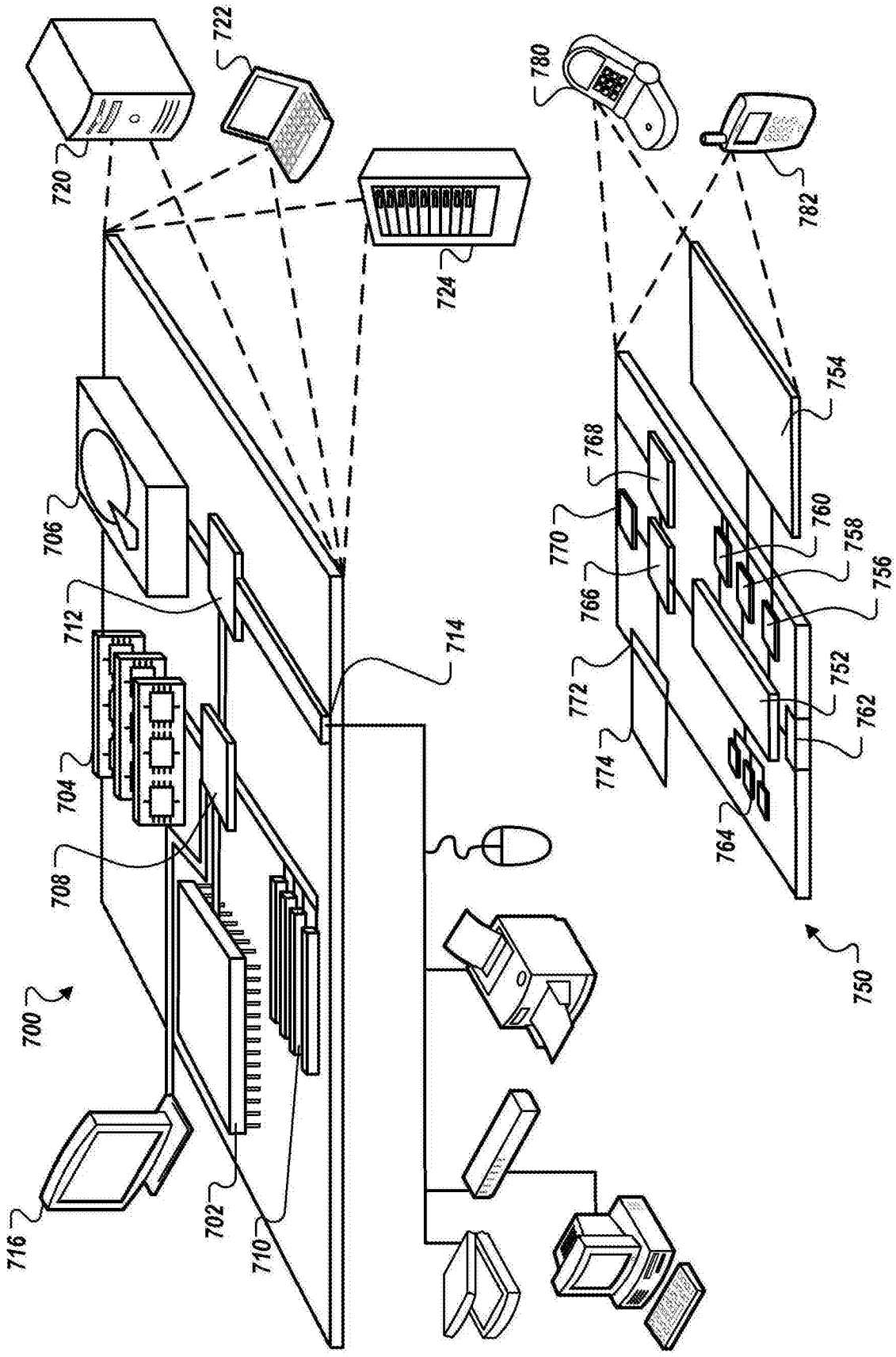

722
720
724
780
782
754
768
760
770
758
766
756
772
752
762
774
764
750
706
712
714
704
708
700
716
702
710

图7