



(12) 发明专利

(10) 授权公告号 CN 114448977 B

(45) 授权公告日 2024. 10. 15

(21) 申请号 202111294754.9
(22) 申请日 2021.11.03
(65) 同一申请的已公布的文献号
申请公布号 CN 114448977 A
(43) 申请公布日 2022.05.06
(30) 优先权数据
17/088,490 2020.11.03 US
(73) 专利权人 辉达公司
地址 美国加利福尼亚州
(72) 发明人 R·莱巴雷迪安 M·卡斯
B·哈里斯 A·舒利任科
(74) 专利代理机构 北京市磐华律师事务所
11336
专利代理师 高伟

(51) Int.Cl.
H04L 67/10 (2022.01)
H04L 67/131 (2022.01)
H04L 61/3015 (2022.01)
A63F 13/33 (2014.01)
(56) 对比文件
US 2018307794 A1, 2018.10.25
US 2014267237 A1, 2014.09.18
审查员 姜佳轮

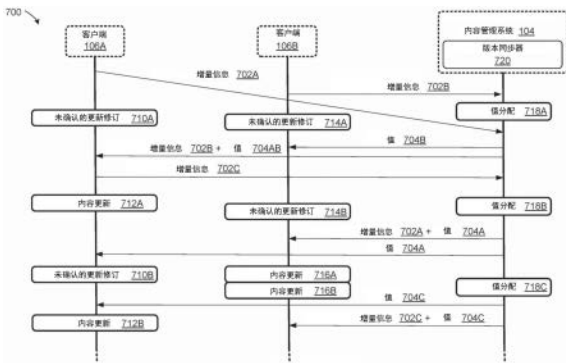
权利要求书3页 说明书39页 附图19页

(54) 发明名称

以云为中心的协作和连接平台中的增量传播

(57) 摘要

公开了以云为中心的协作和连接平台中的增量传播。内容管理系统可以使用场景图中的元素之间的层次关系来维护表示3D世界的场景描述。客户端可以在正在编辑和/或在客户端之间共享的内容版本之间交换增量信息。每个增量信息集可以被分配值序列中的值,该值定义了应用增量信息集以产生场景图的同步版本的顺序。客户端可以遵循冲突解决规则以一致地解决增量信息集之间的冲突。可以程序性地表示对内容的结构化元素的改变以保持客户端之间的结构一致性,同时可以声明性地表示对非结构化元素的改变以减小数据大小。为了存储和管理内容,可以使用节点标识符来引用结构化元素,并且可以将非结构化元素作为字段-值对分配给节点标识符。



1. 一种以云为中心的协作和连接平台中的增量传播方法,包括:

由内容管理系统的客户端传输表示三维虚拟环境的场景图的两个或更多个版本之间的一个或更多个差异的增量信息;

由所述客户端接收指示分配给所述增量信息的值的数据,所述值是被分配给对应增量信息集的值序列中的至少一个值,所述序列对顺序的次序进行编码以应用所述对应增量信息集;

由所述客户端根据所述顺序的次序并使用所述序列和所述值,将所述增量信息集应用于表示所述两个或更多个版本的第一版本的第一图,以修改所述第一图来产生表示所述两个或更多个版本的第二版本的第二图,所述第二版本是所述场景图的同步版本;以及

由所述客户端并使用所述第二图生成与所述场景图的同步版本相应的一个或更多个图像。

2. 如权利要求1所述的方法,其中所述应用包括:在所述场景图的先前同步版本上执行由所述增量信息指定的所述第一图的程序性更新,所述程序性更新包括将一个或更多个节点添加到所述第一图的有序命令列表以产生包括所述一个或更多个节点的所述第二图。

3. 如权利要求1所述的方法,其中所述应用包括:在所述场景图的先前同步版本上执行由所述增量信息指定的所述第一图的声明性更新,所述声明性更新定义对所述第一图的节点的字段值的至少一次分配以产生包括新分配给所述节点的所述字段值的第二图。

4. 如权利要求1所述的方法,其中所述增量信息指定对所述第一图的一个或更多个结构化元素的程序性更新和对所述第一图的一个或更多个非结构化元素的声明性更新,以产生所述第二图。

5. 如权利要求1所述的方法,还包括:

在传送所述增量信息后,由所述客户端接收不同的增量信息和分配给所述不同的增量信息的多个值中的不同值;以及

由所述客户端至少基于对应于所述序列中比所述值更早的位置的所述不同值,生成表示比所述场景图的同步版本更早的所述场景图的版本的第三图。

6. 如权利要求1所述的方法,其中所述应用是针对所述场景图的先前同步版本,并且所述应用至少基于确定分配给所述增量信息的值遵循分配给所述先前同步版本的所述序列的先前值来执行。

7. 如权利要求1所述的方法,其中所述应用是针对所述场景图的先前同步版本,所述增量信息指定与所述先前同步版本有冲突的至少一个命令,并且将所述增量信息应用于所述场景图使用冲突解决规则来解决所述冲突。

8. 如权利要求1所述的方法,其中所述增量信息集中的每个集定义所述3D虚拟环境的相应版本的相应图,所述相应图对应于所述场景图的相应同步版本和所述序列中的相应值。

9. 如权利要求1所述的方法,其中所述增量信息的应用是响应于所述客户端接收分配给所述增量信息的所述值,所述应用包括所述客户端基于所述值在所述序列中的相对位置使用所述值来确定所述增量信息在所述顺序的次序中的相对位置。

10. 如权利要求1所述的方法,其中所述增量信息使用表示所述场景图中的结构化元素的节点的节点标识符来定义所述场景图的非结构化元素到结构化元素的分配。

11.一种以云为中心的协作和连接平台中的增量传播方法,包括:

从内容管理系统的第一客户端接收表示三维虚拟环境的场景图的一个或多个版本之间的一个或多个差异的增量信息;

向所述增量信息分配值,所述值是值序列中的至少一个值,所述序列对顺序的次序进行编码以应用对应增量信息集;以及

将指示所述值的数据传送给所述第一客户端,所述传送使所述第一客户端根据所述顺序的次序并使用所述序列和所述值,将所述增量信息集应用于表示所述一个或多个版本的第一版本的第一图来修改所述第一图,以产生表示所述一个或多个版本的第二版本的第二图,所述第二版本是所述场景图的一个或多个同步版本。

12.如权利要求11所述的方法,其中所述值是所述场景图的版本的版本标识,所述内容管理系统的服务器至少基于该服务器从一个或多个客户端接收所述增量信息集的次序,使用所述序列来将所述值分配给所述增量信息集。

13.如权利要求11所述的方法,还包括:

从第二客户端接收所述场景图的版本之间的不同的增量信息;

由服务器至少基于数学公式递增所述值序列的值来生成所述值序列的不同值,所述数学公式输出所述序列中的值;

由所述服务器将所述值序列中的不同值分配给所述不同的增量信息;以及

由所述服务器向所述第一客户端传送指示所述不同值的数据,所述传送使所述第一客户端使用所述顺序的次序将所述不同的增量信息应用于所述场景图。

14.如权利要求11所述的方法,其中服务器至少基于由该服务器从所述内容管理系统的客户端接收所述增量信息集的次序,将所述序列中的值分配给所述增量信息集,来定义应用所述增量信息集的所述顺序的次序。

15.一种以云为中心的协作和连接平台中的增量传播系统,包括:

至少一个处理单元,用于实现:

通信管理器,用于建立与协作内容创建应用程序的客户端的双向通信信道,所述双向通信信道用于从所述客户端接收表示三维虚拟环境的场景图的不同版本之间的一个或多个差异的增量信息集,用于至少基于序列中的多个值与所述增量信息集之间的分配来同步所述场景图的不同版本,以获取一个或多个所述场景图的同步版本,以及用于将所述场景图的一个或多个同步版本传播到所述客户端;

其中,所述序列用于生成所述分配,所述序列对次序进行编码以将所述增量信息集应用于表示所述版本的第一版本的第一图来修改所述第一图,以产生表示所述版本的第二版本的第二图,所述第二版本是所述场景图的一个或多个同步版本。

16.如权利要求15所述的系统,其中所述系统包括数据存储,所述数据存储包括所述场景图的所述一个或多个同步版本中的至少一个同步版本的记录,并且所述记录表示所述场景图的一个或多个同步版本之间的增量。

17.如权利要求15所述的系统,其中所述多个值中的至少一个值标识存储在数据存储中的所述场景图的一个或多个同步版本中的至少一个同步版本。

18.如权利要求15所述的系统,其中节点标识符引用存储在数据存储中的所述场景图的所述一个或多个同步版本中的至少一个同步版本的结构化元素。

19. 如权利要求15所述的系统,其中应用所述增量信息集的次序基于所述增量信息集被所述通信管理器接收的次序。

20. 如权利要求15所述的系统,其中所述场景图属于场景图层中的层,所述场景图层使用所述层的等级构成以生成定义所述三维虚拟环境的复合场景图。

以云为中心的协作和连接平台中的增量传播

[0001] 相关申请的交叉引用

[0002] 本申请与2020年3月22日提交的题为“用于在3D虚拟环境中进行协作和连接的以云为中心的平台 (Cloud-Centric Platform for Collaboration and Connectivity on 3D Virtual Environments)”的美国非临时申请第16/826,296号和2019年8月12日提交题为“协作生成内容的平台和方法 (Platform and Method for Collaborative generation of Content)”的美国非临时申请第16/538,594号相关。这些申请中的每一个都通过引用整体并入本文。

背景技术

[0003] 游戏引擎——例如Unreal Engine、Unity和CryEngine——已经被用于使用户能够在游戏环境内以内容创建的基本形式进行协作。然而，传统游戏引擎并不特别适合协作创作三维(3D)世界的高质量内容。例如，游戏引擎通常旨在优化保真度和一致性的快速复制。因此，每个客户端都可以接收到足够准确以共享和传达场景或体验的共享3D环境的估计。然而，高质量的协作3D内容创作可能需要每个参与者查看共享3D环境的忠实且一致的表示。此外——为了促进快速复制——游戏引擎为客户端提供了一个简单的3D世界原子级描述，其可包括对象几何和变换。然而，创作高质量的3D世界可能需要交换世界的丰富描述，以支持现代内容创作工具所需的保真度和特征。

[0004] 通用场景描述(USD)框架允许使用场景图中元素之间的复杂层次关系对3D世界进行丰富的描述。USD是为非交互式娱乐的3D电影的离线开发而开发和设计的。在内容创建管线中，作者轮流单独开发内容，完成后可以通过手动传输和组合包含场景描述部分的大文件来合并。在支持并发协作和连接性的系统中使用如此丰富的描述对以保真度和一致性复制和存储场景元素提出了重大挑战。

发明内容

[0005] 本公开涉及用于在3D虚拟环境上进行协作和连接的以云为中心的平台的方法。本公开的方面提供在用于协作和连接的以云为中心的平台中的增量传播。

[0006] 内容管理系统可以使用场景图中的元素之间的层次关系来维护表示3D世界的场景描述。在某些方面，客户端可以在正在编辑和/或在客户端之间共享的内容的版本之间交换增量信息。每个增量信息集可以被分配值序列中的值，该值序列定义了将增量信息集应用于场景图以产生场景图的同步版本的顺序。客户端可以各自遵循冲突解决规则以一致地解决增量信息集之间的冲突。

[0007] 增量信息集可以包括对内容的结构化元素的改变和对内容的非结构化元素的改变。对结构化元素的改变可以程序式表示以保持跨客户端的内容的结构一致性，而对非结构化元素的改变可以声明式表示以减少数据大小。为了存储和管理内容，可以使用节点标识符(ID)来引用内容的结构化元素(节点)，并且可以将非结构化元素作为字段-值对分配给节点ID，从而允许识别适当的节点，即使该节点被重定父级或重命名。在实施例中，可以

使用对象版本的层次结构来存储内容,并且可以通过将子版本和父版本之间的改变存储到子(child)来减少存储空间。本公开的进一步方面涉及高速缓存对象的版本以有效地向客户端提供内容。

附图说明

[0008] 下面参照所附附图详细描述用于在以云为中心的平台中进行增量传播以进行协作和连接的本系统和方法,其中:

[0009] 图1是示出根据本公开的一些实施例的可用于协作创作共享内容的操作环境的示例的图;

[0010] 图2A示出了根据本公开的一些实施例的可以如何定义3D虚拟环境的资产的属性和价值的示例;

[0011] 图2B示出了根据本公开的一些实施例的可以如何解决图2A的属性和价值的示例;

[0012] 图2C是示出根据本公开的一些实施例的使用数据存储来创建多个虚拟环境的示例的框图;

[0013] 图2D是示出根据本公开的一些实施例的用于虚拟环境分叉的数据存储的使用的示例的框图;

[0014] 图3A示出了根据本公开的一些实施例的使用场景描述表示的3D虚拟环境的图形表示的显示的示例;

[0015] 图3B示出了根据本公开的一些实施例的使用图3A的场景描述表示的3D虚拟环境的图形表示的动画编辑器中的显示的示例;

[0016] 图3C示出了根据本公开的一些实施例的使用图3A的场景描述表示的3D虚拟环境的图形表示的游戏引擎编辑器中的显示的示例;

[0017] 图3D示出了根据本公开的一些实施例的使用图3A的场景描述表示的3D虚拟环境的图形表示的光栅图形编辑器中的显示的示例;

[0018] 图4A示出了说明根据本公开的一些实施例的在传输基础设施上实现发布/订阅模型的操作环境的组件的示例的框图;

[0019] 图4B示出了说明根据本公开的一些实施例的在包括网络的传输基础设施上实现发布/订阅模型的操作环境的组件的示例的框图;

[0020] 图5是示出根据本公开的一些实施例的内容管理系统和客户端之间的信息流的示例的框图;

[0021] 图6是示出根据本公开的一些实施例的包括多个内容管理系统的操作环境的示例的图;

[0022] 图7是示出根据本公开的一些实施例的用于同步3D虚拟环境的内容的版本的过程的示例的数据流程图;

[0023] 图8是示出根据本公开的一些实施例的客户端可用于更新内容的同步版本的方法的示例的流程图;

[0024] 图9是示出根据本公开的一些实施例的服务器可用于更新内容的同步版本的方法的示例的流程图;

[0025] 图10是示出根据本公开的一些实施例的系统可用于更新内容的同步版本的方法

的示例的流程图；

[0026] 图11是示出根据本公开的一些实施例的可以由数据存储使用以捕获表示分层元素的对象的结构示例的图；

[0027] 图12A是示出根据本公开的一些实施例的对象的版本的示例的图；

[0028] 图12B是示出根据本公开的一些实施例的用于对象的版本的数据存储的示例的图；

[0029] 图13是适用于实现本公开的一些实施例的示例计算设备的框图；以及

[0030] 图14是适合用于实现本公开的一些实施例的示例数据中心的框图。

具体实施方式

[0031] 本公开涉及用于在3D虚拟环境上进行协作和连接的以云为中心的平台的方法。本公开的方面提供在用于协作和连接的以云为中心的平台中的增量传播。

[0032] 内容管理系统可以使用场景图中的元素之间的层次关系来维护表示3D世界的场景描述。在某些方面，客户端可以在正在编辑和/或在客户端之间共享的内容的版本之间交换增量信息。每个增量信息集可以被分配值序列中的值，该值序列定义了将增量信息集应用于场景图以产生场景图的同步版本的顺序。当客户端向服务器发送增量信息时，客户端可以等待服务器提供该值，然后在收到该值后根据该值应用增量信息。在等待该值时，可以根据该顺序接收和应用增量信息集。客户端可以各自遵循冲突解决规则以一致地解决增量信息集之间的冲突。使用公开的方法，客户端无需等待来自服务器的增量信息集已被接受的确认。此外，由于在内容的错误版本之间创建增量信息，客户端不需要重新创建增量信息集。

[0033] 本公开的另外的方面提供用于创建捕获对包括分层元素（例如，场景描述）的内容的改变的增量信息集。增量信息集可以包括定义对场景描述的一个或多个结构化元素的一个或多个改变的部分和定义对场景描述的一个或多个非结构化元素的一个或多个改变的部分。结构化元素可以对应于场景图的图节点，以及节点之间所示的互连。非结构化元素可以指分配给节点和/或结构化元素的属性和/或值（例如，字段-值对）。非结构化元素通常不会影响场景图的结构，而结构化元素可以定义场景图的结构。内容的结构化元素（例如，定义节点和/或节点之间的关系）可以程序性地表示，例如使用可以对内容版本执行的一个或多个命令以生成内容的更新版本。这可以保持跨客户端的内容的结构一致性。内容的非结构化元素（例如，定义结构化元素的字段和值）可以声明性地表示。这可以减少数据大小，因为不需要记录版本之间的中间状态。

[0034] 本公开还提供了用于存储和管理包括分层元素的内容的方法。在至少一个实施例中，内容的每个节点可以具有唯一标识符（ID）。节点的唯一ID可以在创建节点时分配给节点（例如，在创建命令中）。唯一ID可用于节点的整个生命周期，无论是重命名、删除还是重定父级。可以使用节点的唯一ID指定节点的结构化改变和/或节点的属性-值对（例如，字段和/或字段值）的改变和/或分配。在一些实施例中，唯一ID可由创建节点的客户端106生成和/或分配给该节点。例如，节点的唯一ID（可以更一般地称为节点ID）可以是随机生成的64位或128位数字。因此，为了改变节点的字段的字段值，增量信息集可以包括节点ID、字段ID和字段值。

[0035] 根据本公开的一些方面,数据存储可以使用节点ID来存储和引用场景描述的结构化元素(节点),并且可以将非结构化元素分配给节点ID作为字段-值对。字段-值对可以用作键-值对,除了数据存储中的单个键-值对之外,键-值对可以是每节点ID或节点。例如,节点可以存储在与数据存储中的键-值对分开的结构或表中。当客户端引用节点时,客户端可以引用节点ID和一个或多个具有节点ID的相关字段-值对,即使该节点被重定父级或重命名,也允许识别正确的节点。

[0036] 在本公开的进一步方面,可以使用对象版本的层次结构来存储内容,并且可以通过将子版本和父版本之间的改变存储到子来减少存储空间。本公开的进一步方面涉及高速缓存对象的版本以有效地向客户端提供内容。

[0037] 虽然该描述主要提供对应于虚拟环境和三维(3D)内容的内容的示例,但所公开的方法可以应用于各种内容类型(例如,分层和/或树或基于图的内容)。

[0038] 参考图1,图1是示出根据本公开的一些实施例的可用于协作创作共享内容的操作环境100的示例的图。应当理解,本文描述的这种布置和其他布置仅作为示例阐述。除了所示的那些之外或代替所示的那些,可以使用其他布置和元件(例如,机器、接口、功能、顺序、功能分组等),并且可以完全省略一些元件。此外,本文描述的许多元件是功能实体,它们可以被实现为离散或分布式组件或与其他组件结合,并且以任何合适的组合和位置来实现。本文描述为由实体执行的各种功能可以由硬件、固件和/或软件来执行。例如,各种功能可以由执行存储在存储器中的指令的处理器来执行。举例来说,操作环境100可以在图13的计算设备1300的一个或多个实例上实现。

[0039] 操作环境100可以包括任意数量的客户端,例如客户端106A和106B到106N(也称为“客户端106”)和内容管理系统104。这些组件可以经由网络120彼此通信,其可以是有线、无线或两者。网络120可以包括多个网络或网络中的网络,但是以简单的形式示出以免混淆本公开的方面。举例来说,网络120可以包括一个或多个广域网(WAN)、一个或多个局域网(LAN)、一个或多个公共网络(例如因特网)和/或一个或多个专用网络。在网络120包括无线电信网的情况下,诸如基站、通信塔或甚至接入点(以及其他组件)之类的组件可以提供无线连接。

[0040] 每个客户端106可以对应于一个或多个应用程序、软件工具和/或服务,其可以在一个或多个计算设备上执行或使用一个或多个计算设备,例如客户端设备102A和102B到102N(也称为“客户端设备102”)。客户端设备102可以包括不同类型的设备;也就是说,它们可具有不同的计算和显示能力以及不同的操作系统。取决于硬件和软件能力,客户端设备102可用于将客户端106实现为胖客户端或瘦客户端。

[0041] 每个客户端设备102可以包括本文关于图13描述的示例计算设备1300的至少一些组件、特征和功能。作为示例而非限制,任何客户端设备102可以体现为个人计算机(PC)、膝上型计算机、移动设备、智能电话、平板计算机、智能手表、可穿戴计算机、个人数字助理(PDA)、媒体播放器、全球定位系统(GPS)或设备、视频播放器、服务器设备、手持通信设备、游戏设备或系统、娱乐系统、车载计算机系统、遥控器、电器、消费电子设备、工作站、这些描述设备的任何组合,或任何其他合适的设备。

[0042] 每个客户端设备102可以包括一个或多个处理器,以及一个或多个计算机可读介质。计算机可读介质可包括可由一个或多个处理器执行的计算机可读指令。这些指

令在由一个或更多个处理器执行时,可以使一个或更多个处理器执行本文描述的方法的任何组合和/或部分和/或实现图1的操作环境100的功能的任何部分(例如,以实现客户端106)。

[0043] 内容管理系统104包括数据存储114、数据存储管理器108和通信管理器110,它们可以在例如一个或更多个服务器上实现,例如服务器112。每个服务器112可以包括一个或更多个处理器,以及一个或更多个计算机可读介质。计算机可读介质可包括可由一个或更多个处理器执行的计算机可读指令。这些指令在由一个或更多个处理器执行时,可以使一个或更多个处理器执行本文描述的方法的任何组合和/或部分和/或实现图1的操作环境100的功能的任何部分(例如,以实现数据存储管理器108和/或通信管理器110)。在至少一个实施例中,内容管理系统104可以至少部分地在图14的数据中心1400中实现。

[0044] 数据存储114可以包括一个或更多个计算机可读介质。例如,数据存储114可以指一个或更多个数据库。数据存储114(或计算机数据存储装置)被描绘为单个组件,但可以体现为一个或更多个数据存储(例如,数据库)并且可能至少部分在云中。例如,数据存储114可以包括在一个或更多个计算系统(例如,数据中心)上实现和存储的多个数据存储和/或数据库。

[0045] 操作环境100可以被实现为以云为中心的平台。例如,操作环境100可以是基于网络的平台,其可以使用经由网络120(例如,因特网)连接并协同工作的一个或更多个设备来实现。然而,虽然操作环境100主要是根据客户端-服务器架构来描述的,但是考虑到不同的网络架构,例如对等网络或混合网络类型,可以考虑不同的布置。尽管被描绘在服务器112内,但是数据存储114可以至少部分地体现在服务器112、客户端设备102和/或一个或更多个其他服务器或设备的任何组合上。因此,应当理解,数据存储114中的信息可以以任何合适的方式分布在一个或更多个数据存储上以供存储(其中一些可以在外部托管)。类似地,数据存储管理器108、通信管理器110和/或客户端106的功能可以至少部分地体现在服务器112、客户端设备102和/或在一个或更多个其他服务器或设备的任何组合上。

[0046] 作为概述,内容管理系统104的数据存储114可以被配置为存储表示资产和元数据的数据,用于定义一个或更多个3D环境,例如一个或更多个3D场景和/或3D世界。内容管理系统104的数据存储管理器108可以被配置为管理数据存储114中的资产和元数据,包括解析3D虚拟环境的属性和/或值。内容管理系统104的通信管理器110可以被配置为管理由内容管理系统104提供或提供给内容管理系统104的通信,例如通过网络120,和/或内容管理系统104内的通信。

[0047] 在至少一个实施例中,内容管理系统104的通信管理器110可以被配置为建立和维护与一个或更多个客户端106的一个或更多个通信信道。例如,通信管理器110可以向每个客户端106提供相应的双向通信信道。在各种实施例中,双向通信信道包括一个或更多个网络套接字(例如,WebSocket)和/或一个或更多个端口。在实施例中,一个或更多个客户端106通过端口或套接字连接到服务器112,并使用实现双向通信的公共应用程序编程接口(API)与服务器112通信,其通过双向通信信道实现双向通信(例如,WebSocket API)。根据公开的实施例,可以在场景描述中定义虚拟环境的资产,场景描述可以是包括属性和值的场景图的形式,和/或根据一个或更多个模式描述属性和值的语言的形式(以文本形式)。在服务器112处对场景描述的部分(例如,文本描述)的改变可以通过信道复制到客户端106,

反之亦然。

[0048] 客户端106可以包括一种或更多种类型的应用、软件和/或服务,例如但不限于:物理模拟应用、人工智能(AI)应用、全局照明(GI)应用程序、游戏引擎、计算机图形应用程序、渲染器、图形编辑器、虚拟现实(VR)应用程序、增强现实应用程序或脚本应用程序。在应用程序或服务彼此不同的实施例中,客户端106可以被称为“异构客户端”。

[0049] 如上所述,内容管理系统104的数据存储114可以被配置为存储表示资产和元数据的数据,用于定义3D环境的一个或更多个元素,例如一个或更多个3D场景和/或3D世界。内容项可以指可单独识别和/或可寻址(例如,通过URI和/或其他标识符)资产或资产的元素(和/或其版本),例如一个或更多个属性或属性-值对。资产的元素可以包括结构化和/或非结构化元素,如本文所述。内容项的元数据(例如,以JSON格式)可以描述底层数据所在的位置,访问允许用户查看和/或修改内容项、时间戳、锁定和解锁状态、数据类型信息和/或其他服务信息的控制列表(ACL)。与底层数据相反,对数据存储114中的数据的许多改变可以对元数据进行操作。例如,复制操作可能不深,因为它可以通过复制元数据信息并创建到相同底层数据的链接例如以分叉内容来完成,如本文所述。

[0050] 元数据和底层数据可以分开存储在数据存储114中,因为它们的缩放方式不同。内存中键值数据库可以与元数据数据库和数据数据库一起使用。可以提供多个数据库实例(例如,在任意数量的机器上)用于扩展,并且可以包括一个或更多个读取从属,以通过复制主实例来更好地扩展读取性能。数据存储管理器108可以通过统一资源标识符(URI)在数据存储114中引用和定位内容项和相关联的元数据。在一些实施例中,数据存储管理器108可以散列URI以确定位置信息并选择适当的数据库实例来访问。在非限制性示例中,实例可以是单线程的,每个实例都运行每CPU内核。

[0051] 数据存储管理器108可以操作一个或更多个增量服务器(例如,每个元数据实例一个)。增量服务器可以将一系列增量变化(例如,对场景描述的)合并或坍塌成新版本的内容,如本文所述。例如,改变可以从特定客户端106接收并且可以被坍塌成与其他客户端106共享的关键帧版本,使得新传入客户端106可以接收反映了变化的内容的相对紧凑版本。

[0052] 资产示例

[0053] 资产可以对应于可以与其他资产一起使用以组成3D虚拟环境的数据(例如,3D数据)。“虚拟环境”可以指虚拟场景、世界或宇宙。虚拟场景可以组合起来形成虚拟世界或宇宙。每个资产可以根据一个或更多个属性、一个或更多个属性的一个或更多个值(例如,具有作为键的属性的键-值对)和/或一个或更多个其他资产和/或内容项来定义(例如,通过属性和值和/或语法)。资产的示例包括层、对象(例如,模型和/或模型组)、阶段(顶层或根场景图)、场景、基元、类和/或其组合。虚拟环境的资产可以在场景描述中定义,场景描述可以是包括属性和值的场景图的形式。此外,在各种实施例中,一些资产的内容项可以跨多个其他资产和/或跨多个文件(例如,场景描述的)和/或数据结构来描述和定义。

[0054] 属性和/或属性值的非限制性示例是可以指定和/或定义几何形状、着色器、纹理、几何变化、阴影变化、细节等级(LoD)、资产引用或标识符、动画、特殊效果、时间信息、模型装配信息、虚拟相机信息、照明信息、堆肥信息、对其的引用(例如,在下面关于引用资产的引用)和/或其实例化(例如,下面参考实例化资产引用的)的一个或更多个部分的那些。在各种示例中,资产的属性和/或属性的值可以随时间变化,例如通过由脚本和/或函数定义

的。

[0055] 可以根据一种或更多种模式、一种或更多种领域特定模式和/或一种或更多种场景描述语言来定义、指定、格式化和/或接口资产。在非限制性示例中,模式、格式、语言和/或接口(例如,API)可以根据通用场景描述(USD)框架。数据存储管理器108和/或客户端106(和/或本文描述的内容管理器410、渲染器414、服务412)可以分析场景描述的资产定义,以便解析3D虚拟环境的资产的属性和价值。模式可以赋予场景描述的属性和值的含义(例如,使用场景描述语言以文本形式书写),例如(例如但不限于)以下的任意或组合:几何、光、物理(例如,刚体、柔性材料、流体和气体)、材料、钻机以及它们的属性随时间变化的方式。可以包括物理参数以指定物理属性,例如质量、惯性张量、摩擦系数和恢复系数,以及关节、铰链和其他刚体约束的规范。用户可以通过添加嵌入在新模式中的自定义属性来扩展场景图。

[0056] 在各种示例中,场景描述的资产定义可以在其中指定和/或定义一个或更多个其他资产和/或其中其他资产(例如,在一个层)的一个或更多个部分(例如,属性和/或价值)。在这样的示例中,资产可以被称为包含资产或其他资产的容器,并且其他资产可以被称为相对于包含资产的嵌套资产。例如,层可以包括至少部分地定义在其中的一个或更多个对象。在实施例中,本文描述的各种资产类型中的任一种可以是包含资产和/或相对于另一资产的嵌套资产。此外,包含资产可以是任意数量的其他包含资产的嵌套资产和/或可以包括任意数量的嵌套资产,其中任何一个本身可以是一个或更多个其他资产的包含资产。

[0057] 同样在各种示例中,资产可以在场景描述中被指定和/或定义为一个或更多个其他资产和/或(例如,某类的)其他资产的一个或更多个部分(例如,属性和/或值)的实例化。在这样的示例中,资产可以被称为实例化资产或其他资产的实例,并且其他资产可以被称为相对于实例资产的源资产。在实施例中,本文描述的各种资产类型中的任一种可以是源资产和/或相对于另一资产的实例化资产。例如,对象可以是类的实例化。此外,实例化资产可以是任意数量的其他实例化资产的源资产和/或可以包括任意数量的源资产,其中任何一个本身可以是一个或更多个其他资产的实例化资产。在各种实施例中,实例化资产可以从任何数量的源资产(例如,类)继承。多重继承可以指实例化资产从多于一个源资产继承。例如,对象或类可以从多于一个父对象或父类继承属性和/或值。此外,与其他资产类型一样,父对象或父类可以跨任何数量的层来定义和解析,如本文所述。

[0058] 此外,资产的一个或更多个属性和/或值可以在场景描述中通过对一个或更多个其他资产的一个或更多个引用和/或一个或更多个其他资产的一个或更多个实例化来定义(例如,通过属性和值)。资产可以包括对另一资产的引用(例如,标识符)或指针,该另一资产将另一资产的一个或更多个部分合并到该资产中。在这样的示例中,该资产可以被称为参考资产,而另一资产可以被称为相对于参考资产的合并资产。在实施例中,本文描述的各种资产类型中的任一种可以是参考资产和/或相对于另一资产的合并资产。此外,参考资产可以是任意数量的其他参考资产的合并资产和/或可以包括任意数量的合并资产,其中任何资产本身可以是一个或更多个其他资产的参考资产。

[0059] 可以在场景描述中使用包含资产、嵌套资产、实例化资产、源资产、参考资产和/或合并资产的各种组合来共同定义3D虚拟环境的资产的属性和对应值。根据一个或更多个模式,这些关系可以通过属性和值明确定义或指定和/或从场景描述的结构中隐式定义或指

定。例如,指定和/或定义为实例化资产的资产可能会导致资产从源资产继承一个或多个属性和/或值。此外,被指定和/或定义为参考资产的合并资产的资产可以导致参考资产从合并资产继承一个或多个属性和/或值。

[0060] 此外,在至少一个实施例中,可以在场景描述中定义和/或指定从一个或多个其他资产继承的资产的一个或多个属性,其中对一个或多个属性的覆盖来自另一资产。例如,对属性的覆盖可以用不同的值和/或属性替换或取代属性的值和/或属性。可以根据资产描述的语法或模式(例如,在资产定义中)使用属性和值明确声明或指定资产的覆盖,和/或可以从语法或模式中隐含(例如,根据资产的声明位置)。例如,将值分配给资产中的属性可用作对该属性从另一资产继承的值的显式覆盖。

[0061] 在至少一个实施例中,可以在3D虚拟环境的场景描述中提供层。层可以包含或分组零个或多个其他资产类型,例如对象和类,这些资产类型又可以描述这些和/或其他资产的属性值。在一些示例中,每个层可以包括标识符,该标识符可以用于构造从其他层到该层的引用。在一些实施例中,每一层对应于用于表示数据存储114内的层的相应文件(例如,场景描述的相应文件)。

[0062] 每一层可被分配(例如,由客户端、用户和/或数据存储管理器108)相对于3D虚拟环境的其他层的排名。数据存储管理器108和/或客户端106可以使用排名来解析3D虚拟环境的资产的一个或多个属性和/或价值。例如,数据存储管理器108可以通过根据排名组合场景描述的资产定义来将属性和值确定为一个或多个层中的资产的合并视图。在一个或多个实施例中,层可以表达或定义关于组合的3D场景的资产的属性和/或值的“意见”,并且数据存储管理器108可以在组合或合并多个层的场景描述时使用最强或最高排名的层的意见。在至少一个实施例中,层的强度可以由层在有序列表或层堆栈中的位置来定义。例如,列表或堆栈可以从最强层到最弱层排序。层可用于修改场景描述中现有资产的属性和/或值,而无需修改其源,以便通过在更强的层中覆盖它来改变几乎任何方面。

[0063] 在至少一个实施例中,虚拟环境的场景描述可以被解析为变换层次结构(例如,场景图)的树结构。层之间的关系可以用于通过影响将3D虚拟环境的资产的一个或多个方面组合或解析为树结构(例如,根据排名)的方式,来改变转换层次结构中任何地方的资产的属性和/或值。例如,层内的对象或其他资产可以包括在转换层次结构的不同叶中。层的使用可以允许跨层(或组)中的对象或其他资产的属性和值被改变。例如,汽车的发动机和车门可以表示为转换层次结构中的不同对象。然而,引擎和门都可以包括螺钉,并且可以使用层来允许螺钉的属性发生改变,无论螺钉出现在转换层次结构中的什么位置。

[0064] 因此,可以在场景描述的资产定义的一个或多个层次结构中定义和描述场景的资产,其可以共同定义3D场景的资产或元素的属性和值。层次结构的非限制性示例包括模型层次结构、转换层次结构、层层次结构、类层次结构和/或对象层次结构,其中一个或多个可以嵌入到另一个层次结构和/或层次结构类型中。

[0065] 在各种示例中,数据存储管理器108可以分析场景描述的资产定义、元数据和/或由资产定义(根据层次结构)指定的相关联的属性和/或值,以便解析与3D虚拟环境的一项或更多项特定资产或元素相关联的一项或更多项属性和/或值。这可以包括,例如,遍历一个或多个层次结构、数据结构和/或其部分,以解析属性和值。例如,数据存储管理器108可以访问对由场景描述定义的资产和/或其实例化的指定引用,以便遍历层次结构。

[0066] 现在参考图2A和图2B,图2A和图2B示出了根据本公开的一些实施例的可以如何定义和解析3D虚拟环境的资产的属性和价值的示例。图2A的元素或资产可以称为场景描述的未解析元素或资产,以及图2B的元素或资产可被称为场景描述的解析的或组合的元素或资产。图2A示出了可以根据3D虚拟环境的场景描述来定义的层202和层204,图2B示出了3D虚拟环境的解析视图206。3D虚拟环境的场景描述可以包括附加资产,例如附加层,其未在图2A和图2B中示出。层202可以包括资产210、212、214、216、218、220、216、218、220、222和250的定义,并且层204可以包括资产230、216和222的定义。

[0067] 在所示的示例中,资产216、218和220可以在场景描述中各自被定义为层204的资产230的参考资产,其可以是关于资产216、218和220的合并资产。因此,资产216、218和220可以各自从资产230继承属性和/或值。资产230的场景描述可以包括将颜色属性分配给绿色的属性-值对236。然而,资产230可以被定义为资产222的实例化资产,其是关于资产230的源资产(例如,类别)。因此,资产230可以从将颜色属性分配给蓝色的资产222继承属性-值对228。层202可以被排序为比层204更强的层。因此,属性-值对228可以覆盖资产230的属性-值对236。因此,资产216、218和220每个都可以从资产230继承属性-值对228。然而,资产220的场景描述可以包括可以覆盖属性-值对228的属性-值对226。因此,数据存储管理器108可以将资产216解析为具有属性-值对228,将资产218解析为具有属性-值对228,并且将资产220解析为具有属性-值对226,如解析视图206所示。

[0068] 此外,资产220可以被定义为资产250的实例化资产,其是相对于资产220(例如,类别)的源资产。因此,资产220可以从资产250继承属性-值对252和254并且从资产222继承属性-值对228(在该示例中被覆盖),其提供多重继承的示例,其中实例化资产可以具有多个源资产。例如,资产220是多个类的实例化。另一资产(未示出)也可以从可包括或不包括资产250和/或资产222的一组不同的类别继承。例如,资产220可以表示飞机的螺旋桨,资产220和表示机场机库的资产两者可以从资产250继承,因此它们每个都包括闪亮金属表面的属性。因此,在各种实施例中,属性继承可以沿着转换层次结构以及从多个类操作。

[0069] 层202和204可以按照场景图由场景描述来定义,其解析为解析视图206的场景图,如图所示(例如,通过根据解析规则合并场景图)。解析视图可以由任意数量的层和/或组成场景图组成。场景图的一些属性和值可以通过声明场景图的对象或节点和/或节点或对象之间的关系来定义或声明场景图的结构。这些属性和值可以称为场景描述的结构化元素。定义或声明关系的结构化元素的示例包括声明或定义类或其他资产的实例化的结构化元素、对另一个对象或资产的引用、场景元素或对象的变体,和/或对象或资产之间的继承关系。一般来说,在图2A中,视觉描绘的图节点以及节点之间示出的互连可以各自对应于结构化元素。结构化元素的示例是图2A的层202中的资产222的声明。结构化元素的进一步示例可以是图2A中指示的资产216和230之间的引用关系的声明,以及资产250和资产220之间以及资产230和资产222之间的继承关系的声明。

[0070] 其他属性和值可以定义或声明属于场景图的对象或节点的字段和值。这些属性和值可以称为场景描述的非结构化元素。非结构化元素的示例是图2A的层202中的资产222的属性-值对228的声明。一般而言,在图2A中,附加到视觉描绘的图形节点的元素可以各自对应于非结构化元素。

[0071] 虽然图2B的解析视图206示出了由图2A的层202和204中描绘的每个未解析元素产

生的解析元素——例如资产(或对象)和对应的属性-值对,客户端106、内容管理系统104和/或其他组件可以根据需要或根据期望(通过解析和/或遍历场景描述的一个或多个部分或子集)来确定解析元素,并且可能不一定从未解析的场景描述中解析每个元素。通常,解析视图或场景描述可以指从场景描述显现或组成的3D虚拟环境的状态。解析视图的一个或多个元素可以是3D虚拟环境渲染和/或呈现的内容。

[0072] 在实施例中,客户端106和/或操作环境100的其他组件可以解析场景描述的可用的和/或活动的用于合成的部分。例如,客户端106可以解析客户端106订阅的场景描述的部分或内容项并且可以不使用未订阅的部分或内容项来解析或合成解析视图的一个或多个部分。这可导致不同客户端106使用相同共享场景描述的不同解析视图。例如,如果客户端106A被订阅到层202和204,则客户端106A可以使用图2B的解析视图206。然而,如果客户端106B被订阅到层202而不是层204,则客户端106B使用的解析视图可能不同。例如,资产216、218和220可能不再从资产222继承,从而资产216和218的颜色属性不再解析为蓝色,如图2B所示。为了进一步举例,客户端106B可以订阅到另一层(未示出),该层为资产230提供与层204不同的定义,导致资产216、218和220的不同属性和值。另外,另一层也可被客户端106A订阅,但未显示在解析视图206中,因为它具有比层204低的排名。由于层204对于客户端106B不可用和/或不活动,先前从另一层覆盖的一个或多个元素现在可以在客户端106B的解析视图中显现。

[0073] 现在参考图2C,图2C是示出根据本公开的一些实施例的使用数据存储来创建多个虚拟环境的示例的框图。在图2C的示例中,在数据存储114中描述的资产240A、240B和240C(或更一般地,内容项)可以被不同虚拟环境242A、242B和242C的场景描述引用。例如,资产240A可以在虚拟环境242A和242B两者中使用。作为示例,资产240B可以在虚拟环境242B的至少一些场景描述中定义并且由虚拟环境242A的至少一些场景描述引用或实例化,如本文所述。例如,层的场景描述可以在多个虚拟环境的场景描述之间共享。

[0074] 现在参考图2D,图2D是示出根据本公开的一些实施例的用于虚拟环境分叉的数据存储114的使用的示例的框图。例如,虚拟环境244可以被分叉以创建虚拟环境244A。将虚拟环境分叉成多个副本可能是相对便宜的(计算上)操作。例如,可以通过在版本控制系统中创建新的源控制分支来实现虚拟环境的分叉。对数据存储114中的一个或多个资产版本的引用可以从虚拟环境244复制到虚拟环境244A,如图2D所示。因此,为了从虚拟环境244中分叉虚拟环境244A,虚拟环境244A的对应资产名称可以被配置为指向虚拟环境244的资产版本260、262和264。在一些实施例中,可以采用写时复制(CoW)资源管理方案,使得被复制的资产版本最初在虚拟环境244和虚拟环境244A之间共享,如图2D所示。一旦分叉,虚拟环境244和/或244A的场景描述可以被修改以区分虚拟环境,例如通过覆盖、附加资产定义和/或对资产版本做出的改变。对虚拟环境244做出的一个或多个改变可以在不影响虚拟环境244A的情况下进行,反之亦然。例如,如果用户修改与虚拟环境244A中的资产版本264相对应的资产,则虚拟环境244A的资产名称可以被更新以指向新的资产版本264A,同时保留虚拟环境244的资产版本264,如图2D所示。如果用户向虚拟环境244添加新资产,则虚拟环境244的资产名称可以被创建并且可以指向对应的资产版本266,如图2D所示。尽管未示出,但如果在虚拟环境244A和244之间具有共享资产版本的资产中声明新资产,则对资产的改变也可能导致该资产的新资产版本(如虚拟环境244A和244每个都可以使用许多相互关联

的资产和/或文件来表示)。在一些实施例中,这些资产版本中的任何一个都可以进行合并,如本文所述。一个或多个客户端106可以请求(例如,在用户或算法的指导下)将3D虚拟环境的版本和/或其一个或多个特定内容项持久地存储在内容管理系统上104以保证可恢复性。

[0075] 现在参考图3A-3D,图3A-3D示出了根据本公开的一些实施例的3D虚拟环境的图形表示的显示的示例。根据本公开的实施例,图3A-3D中的显示300A、300B、300C和300D可以由图1的客户端106和/或客户端设备102的任何组合呈现。作为示例,所有显示300A、300B、300C和300D可以由相同的客户端106和/或相同的客户端设备102呈现(例如,在不同的窗口中和/或在不同的监视器上)。作为进一步的示例,显示300A、300B、300C和300D可以各自由相应的客户端106和/或相应的客户端设备102呈现。

[0076] 图3A-3D中的显示300A、300B、300C和300D是3D虚拟环境的相同场景描述的渲染。特别地,显示300A、300B、300C和300D可以各自对应于由客户端106经由内容管理系统104共享的3D虚拟环境的相同场景定义或描述和版本。然而,由于各种可能的原因,3D虚拟环境的图形表示可能在每个客户端中出现不同。例如,客户端106和/或数据存储管理器108可以在3D虚拟环境的场景描述中停用和/或激活资产和/或其部分的一个或多个描述。作为另一示例,由于缺乏客户端和/或用户的许可,3D虚拟环境的场景描述中的资产和/或其部分的一个或多个描述可能无法用于资产解析。当解析3D虚拟环境的资产时,数据存储管理器108和/或客户端106(和/或内容管理器410)可以排除场景描述的不可用和/或不活动部分(例如,当遍历由场景描述定义的层次结构时)。这可能会导致在图形表示中反映的不同的属性和值分辨率。

[0077] 为了说明前述,图3A-3D的3D虚拟环境的场景描述可以对应于图2A的场景描述,其包括层202和204以及一个或多个附加层的定义。一个或多个附加层(未在图2A中指示)可以包括附加资产的资产定义的附加至少部分,例如对应于地面的资产304和显示300C中表示的其他环境资产。对于显示300D和/或显示300B,对应于层的场景描述的一部分可能不可用和/或不活动,因此相应的属性和值可能不会在显示300D和/或显示300B中表示。对于显示300A,与3D虚拟环境相关联的所有层的场景描述可以是活动的。在一些示例中,显示300A、300B、300C或300D的任何组合可以对应于来自内容管理系统104的渲染器414的视频流,如关于图4A和图4B所描述的,或者可以对应于至少部分地由对应的客户端106渲染的帧。

[0078] 在场景描述中使用包含资产、嵌套资产、实例化资产、源资产、参考资产、合并资产和/或覆盖可以使内容管理系统104能够提供复杂场景的丰富描述,复杂场景能够支持现代内容创作工具所需的保真度和特征。例如,可以提供3D虚拟环境的单个表示,其可以捕获可由各种客户端106中的任何一个消费的所有各种场景信息,即使各个客户端106仅能够使用该信息的特定子集和/或格式。可以提供在客户端106之间传送数据的丰富方式,例如通过启用客户端106对数据的非破坏性编辑(例如,通过内容项的覆盖和激活/去激活),以及启用对资产编辑,以通过场景描述层次结构和引用传播到其他资产。另外,资产的表示可以通过允许底层数据的重用而在数据存储114处的存储器中是紧凑的。

[0079] 然而,3D虚拟环境的这种丰富表示可能对解析属性和值所需的网络带宽和计算施加显著限制。例如,支持3D虚拟环境的丰富表示的传统软件和系统(例如USD)是为非交互式

娱乐的3D电影的离线开发而开发和设计的。内容作者通常轮流单独开发内容的各个方面,完成后可以通过手动传输和组合包括场景描述的各部分的大文件来合并。最后,合成场景描述可以通过管线运行,以解析属性和值并将3D内容渲染为视频以供观看。

[0080] 在这种情况下,对于3D虚拟环境的丰富表示,跨设备和系统的动态3D虚拟环境的协作编辑、交互和/或查看以前是不可能的,也不是预期的。例如,当合并场景描述的各部分通常大得令人望而却步以导致使实时或接近实时的应用程序不可能或不切实际的传输时间时,数据的大小通常会被传输。此外,在解析资产时通常分析的场景描述的复杂性通常高得导致处理时间进一步使实时或接近实时应用程序在组合场景描述部分以形成3D虚拟环境时变得不可能或不切实际。

[0081] 发布和订阅模型以及对内容的增量更新

[0082] 根据本公开的方面,发布/订阅模型可由数据存储管理器108(一个或多个数据库服务器)操作,以向客户端106提供3D虚拟环境的场景描述的一个或多个部分。通过内容管理系统104的同步可以是递增的,其中仅对正在发布给订阅者的场景描述进行改变。递增更新可以允许客户端106的和/或内容管理系统104内的内容创建工具、渲染器、增强现实和虚拟现实软件 and/或高级模拟软件的实时互操作。在实施例中,客户端可以发布和订阅他们具有合适许可的任何内容片段(例如,内容项)。当多个客户端106发布和/或订阅相同或重叠的一组内容时,共享的虚拟环境可以具有来自客户端106中的任何一个的以交互速度反映到其他客户端的更新。

[0083] 用例包括但不限于:产品设计和架构的设计评审;场景生成;科学可视化(SciVis);汽车模拟(例如,AutoSIM);游戏的云版本;虚拟场景制作;以及具有用户生成内容和精心设计的世界的社交VR或AR。例如,图形编辑器(例如,Photoshop®)可以连接到内容管理系统104以向虚拟场景中的对象添加纹理,并且计算机图形应用程序或动画工具(例如,Autodesk Maya®)可以连接到内容管理系统104以动画化虚拟场景中的那个对象(或不同的对象)。

[0084] 如本文所述,对内容的订阅可以指对描述内容的场景描述的一部分的订阅。内容的改变或增量可以是关于那个场景描述部分的。例如,表示在操作环境100内交换的内容的数据可以采用场景描述的形式——例如通过文本形式的场景描述语言,和/或通过相应的数据结构和/或场景图组件——和/或以可用于从其版本重建修改的场景描述部分的差异数据的形式。

[0085] 每个客户端106和/或用户可以向内容管理系统104提供对3D虚拟环境的一个或多个识别的资产和/或其一个或多个识别的部分(例如,“内容”或“内容项”)的订阅的请求。基于该请求,内容管理系统104可以向客户端106发布对订阅内容的更新。客户端106对一个或多个资产和/或其一个或多个部分的订阅可以用作至少在未来被通知在内容管理系统104处对于相应内容的改变可用的请求。例如,基于订阅的发布可以包括改变可用于相应内容的通知和/或可以包括表示相应内容的一个或多个部分的数据。在通知识别出对应内容的改变可用的情况下,客户端106可以基于该通知请求表示对应内容和/或对应内容的一个或多个部分的数据。响应于该请求,客户端106可以接收所请求的数据。

[0086] 通常,响应于被提供对内容项的改变,客户端106和/或内容管理器410可以对该内

容项进行另一改变,并且更新共享描述以包括另一改变;对另一个内容项进行改变,并更新共享描述以包括对另一个内容项的改变;使用内容项,其包括在某种类型的操作中不会导致内容项的另一项改变的任何改变;渲染内容项/资产;显示内容项/资产;和/或更新对应于内容项/资产的图形表示。

[0087] 为了对场景描述的已解析属性和/或值的改变采取任何行动,客户端106和/或内容管理器410(以及类似的服务412或渲染器414)可能需要执行属性的一个或更多个部分和/或此处描述的值解析以说明对场景描述所做的任何改变。例如,对一个内容项的场景描述的一部分的改变可以通过本文描述的各种关系传播到任何数量的其他内容项(例如,在其他层中),例如覆盖、继承、引用、实例化等。对于不同的客户端106(或服务),该解析可能不同,这取决于哪些内容项是活动的和/或可用于该客户端106处的属性和值解析。

[0088] 使用本文描述的方法,当一个或更多个客户端106对3D虚拟环境的场景描述的一部分进行改变时,其他客户端106可以仅接收内容和/或对由那些客户端106订阅的场景描述的部分的改变的通知。因此,场景描述的内容及其改变可以根据需要或根据期望提供,从而减少需要跨操作环境100传输以用于协作编辑和/或可能通过网络120发生的客户端106的其他体验的数据量。同样在一些实施例中,而不是完全重新运行客户端106处的场景描述的属性和值解析,内容管理器410可以仅针对更新的内容项和/或对内容项的改变来更新属性和值解析。例如,可以识别差异并且如果这些差异涉及与另一个内容项的关系和/或覆盖,则可以对属性和值解析数据进行相应的更新。但是,可以保留和重用未受影响的属性和值,而不必解析场景图的整个本地版本。

[0089] 在本公开的进一步方面,对从客户端106接收和/或提供给客户端106的内容的更新可以包括与内容(例如,请求和/或订阅的内容)相对应的场景描述部分的版本之间的更改或差异。例如,不是将3D虚拟环境的资产和/或文件的完整描述传输到内容管理系统104,而是每个客户端106可以确定表示内容版本之间差异的数据(例如,描述添加的、删除的和/或修改的属性和/或值),并将该数据提供给内容管理系统104。可以确定差异数据,使得数据存储管理器108和/或其他客户端106能够构建来自差异数据的内容的更新版本(例如,这可以基于使用客户端106进行的编辑)。因此,使用公开的方法,不是在场景描述发生变化时传输场景描述资产的整个副本,而是可以仅传输实现这些改变所需的信息,从而减少需要跨操作环境100传输的数据量,用于可能发生在网络120上的客户端106的协作编辑和/或其他体验。

[0090] 现在参考图4A,图4A示出了说明根据本公开的一些实施例的在传输基础设施420上实现发布/订阅模型的操作环境100的组件的示例的框图。在图4A中,内容管理系统104的通信管理器110包括订阅管理器402、通知器404和API层406。内容管理系统104的数据存储管理器108包括差异确定器408。内容管理系统104还可以包括一个或更多个服务412(其可以包括或涉及一个或更多个微服务),以及一个或更多个渲染器414。在一些实施例中,一个或更多个渲染器414和/或一个或更多个服务412可以是客户端106。因此,客户端106的讨论可以类似地适用于渲染器414和/或服务412。

[0091] 在至少一个实施例中,客户端106、服务412和/或渲染器414可以各自通过API层406(例如,包括诸如Websocket之类的套接字)在传输基础设施420上与内容管理系统104接口。传输基础设施420可以包括图1的网络120的任何组合和/或一个或更多个服务器和/或

客户端设备的进程间通信。例如,在一些实施例中,传输基础设施420包括客户端设备102A、客户端设备102B、客户端设备102C中的一个或多个,一个或多个服务器112和/或一个或多个其他服务器和/或未示出的客户端设备中的一个或多个的进程间通信。

[0092] 在任何示例中,API层406、内容管理系统104的任何其他部分、一个或多个客户端106、一个或多个服务412和/或一个或多个渲染器414可以是至少部分地在这些设备中的一个或多个上实现。传输基础设施420可以根据这些配置而变化。例如,客户端设备102A可以托管内容管理系统104和客户端106A(并且在一些情况下多个客户端106)。在这样的示例中,本地客户端106A使用的传输基础设施420的一部分可以包括客户端设备102A的进程间通信。如果操作环境100中还包括非本地客户端106,则非本地客户端106使用的传输基础设施420的另一部分可以包括网络120的至少一部分。

[0093] 作为进一步的示例,图4B示出了说明根据本公开的一些实施例的在包括网络120的传输基础设施420上实现发布/订阅模型的操作环境的组件的示例的框图。在该示例中,服务412A和服务412B可以对应于图4A的服务412,渲染器414A和渲染器414B可以对应于图4A的渲染器414。服务412A和渲染器414A可以在一个或多个客户端和/或服务器设备上并且通过网络120与内容管理系统104通信。服务412B和渲染器414B可以与内容管理系统104共享客户端和/或服务器设备,并通过进程间通信与内容管理系统104通信。类似地,客户端106A和客户端106B可以在一个或多个客户端和/或服务器设备上并且通过网络120与内容管理系统104通信。客户端106N可以与内容管理系统104共享客户端和/或服务器设备并通过进程间通信与内容管理系统104通信。

[0094] 客户端(或服务或渲染器)可以使用API层406来例如查询和/或修改数据存储114、订阅3D虚拟环境的内容、取消订阅3D虚拟环境的内容和/或接收或提供3D虚拟环境内容的更新或其通知。订阅管理器402可以被配置为管理客户端106对内容的订阅。通知器404可以被配置为向客户端106提供对3D虚拟环境的内容的更新和/或其通知(例如,使用订阅管理器402)。差异确定器408可以被配置为确定内容版本之间的差异,例如内容的当前或基本版本与内容的更新版本之间的差异。在各种实施例中,这可以类似于或不同于由内容管理器410执行的操作,通知器404可以或不将这些差异转发给任何订阅客户端106。

[0095] 服务412可以针对一个或多个3D虚拟环境执行物理模拟、全局照明、光线追踪、人工智能操作和/或其他功能,其可以包括与视图无关的模拟或其他功能。在各种示例中,服务412可以通过使用数据存储管理器108操作和/或更新3D虚拟环境的场景描述来执行这些功能的任何组合。例如,属性和值可以由一个或多个服务412分析和/或更新以实现物理操作、全局照明、光线追踪效果、人工智能等。由服务412做出的改变可以是在客户端106之间共享的场景描述,并且可以或不通过发布/订阅模型进行操作。

[0096] 每个渲染器414可以针对一个或多个客户端106执行渲染存储在数据存储114中的3D虚拟环境的一个或多个方面。渲染的数据可以包括例如3D虚拟环境的帧,其可以流式传输到客户端106以在其上查看。在各种实施例中,渲染器414可以为作为瘦客户端的客户端106(例如移动设备)执行云渲染。在客户端106是VR客户端和/或AR客户端的情况下,渲染器414可以渲染比相机的视野更宽的视频流(例如,RGB-D),并且还可以传送补充深度和来自附近视点的孔填充数据。在客户端106具有陈旧数据的时段期间,客户端106可以使用深度和孔填充数据从新视点重新投影陈旧数据以创建适当的视差。

[0097] 渲染器414和/或集成到客户端106中的渲染器中的一个或更多个可以利用GPU的硬件加速的光线追踪特征。独立通道可用于镜面反射、漫反射、环境光遮挡等。此外,可支持交互式全路径跟踪以获得更准确的结果。渲染器可在单个节点上使用多个GPU以及多个节点协同工作。对于多节点渲染,每个节点可以通过订阅管理器402订阅相同的3D虚拟环境和/或其内容项并渲染适当的图块。控制节点可用于对结果进行定时和合成。可以使用内容管理系统104的消息传递服务来实现节点之间的同步。

[0098] 在图4A和图4B中,每个客户端106被示为包括各自的内容管理器410。例如,客户端106A包括内容管理器410A,客户端106B包括内容管理器410B,并且客户端106N包括内容管理器410N。内容管理器410A、410B和410N在本文中也称为“内容管理器410”。虽然每个客户端106被示为包括内容管理器410,但在一些示例中,一个或更多个客户端106可不包括内容管理器410。例如,在客户端106是瘦客户端(和/或是不在本地处理描述数据的客户端)的情况下,客户端106可以不包括内容管理器410。作为进一步的示例,不同的内容管理器410可以包括本文描述的功能的不同子集或组合。

[0099] 订阅管理器402可以被配置为管理客户端106对一个或更多个3D虚拟环境的内容的订阅。为了订阅一个或更多个内容项,客户端106可以向内容管理系统104的通信管理器110提供识别该内容(例如,经由API层406)的请求(例如,API调用)。例如,客户端106可以提供每项内容的标识符以请求对该内容的订阅。

[0100] 在一些实施例中,客户端106对内容项(例如,层或其他资产类型)的订阅可以对应于对数据存储114中的3D虚拟环境的场景描述的特定文件和/或资源(例如,特定场景描述部分)的订阅。例如,内容的标识符可以包括文件标识符和/或文件或资源的文件路径。在一些示例中,内容项和/或其资源可以在操作环境100内使用URI来识别,URI可以是文本串的形式(例如统一资源定位符(URL))也可以被称为网址。另一个示例包括统一资源名称(URN)。

[0101] 客户端106和内容管理系统104之间的通信可以使用以JavaScript对象表示法(JSON)格式编码的协议,但是也可以使用其他合适的格式。可以支持客户端106的命令(例如,到API层406)以认证、创建文件和/或资产、上传文件和/或资产的内容、读取文件和/或资产、接收目录和/或资产(或资源或内容项)内容的列表,并改变对文件、资源和/或内容项的许可(包括锁定和解锁写入)。内容管理系统104的通信管理器110还可以支持命令来实现消息传递机制,用于连接的客户端106和/或服务412之间所需的任何附加通信。

[0102] 在至少一个实施例中,读取内容项的请求可以用作对内容项的订阅请求。例如,当读取文件和/或资源(例如,场景描述部分)时,客户端106可以有订阅未来变化的选项。响应于客户端106的请求,订阅管理器402可以注册对所标识内容的订阅,并且数据存储管理器108可以将内容提供给客户端106。在内容被提供给客户端106之后,客户端106可以接收以增量的形式发布到内容的所有更新。在一些情况下,向客户端106提供内容可以包括提供所标识内容的所有场景描述。在其他示例中,提供内容可以包括在客户端106和数据存储管理器108之间同步表示内容描述的一个或更多个部分的数据。在客户端106已经包括对应于内容的数据(例如,在本地高速缓存中)的情况下,可以使用同步,所述内容例如内容的较旧版本和/或内容的一部分(例如,来自先前会话)。在此类示例中,差异确定器408可用于确定将内容的哪些部分发送到客户端106和/或确定一个或更多个内容项的客户端和服务端版本

之间的差异数据。在任何示例中,对读取请求的响应可以向客户端106提供在客户端106之间共享的内容的当代或最新版本。

[0103] 订阅请求的非限制性示例可以包括: {‘command’: ‘read,’ ‘uri’:

[0104] ‘/project/asset.usdc’, ‘etag’: -1 ‘id’: 12}。在该示例中,内容的标识符可以包括URI值“/project/asset.usdc”。请求的标识符可以包括id值12。此外,etag值-1可以指示可用于客户端106之间协作的内容的最新版本。在其他示例中,etag值可以用作内容的唯一版本标识符(例如,对于其他消息类型)。对订阅请求的响应的非限制性示例可以包括: {‘status’: ‘LATEST,’ ‘id’: 12}+<资产内容>。在该示例中,<资产内容>可以是表示所请求内容的一个或更多个部分的数据(例如,场景描述和/或差异数据)。其他请求和响应可遵循类似的格式。

[0105] 客户端106可以创建、删除和/或修改3D虚拟环境的内容。更新文件和/或资源可以通过客户端106为内容提供增量或差异来递增地完成。例如,这可针对内容的本地副本或版本而发生。例如,在客户端106从内容管理系统104接收一个或更多个内容项(例如,与一个或更多个订阅相关联)的情况下,客户端106处的内容管理器410可以跟踪对内容进行的此类编辑(例如,场景描述部分)。改变的示例包括向场景描述添加任何元素、从场景描述删除任何元素和/或修改场景描述的任何元素,例如其中的属性和/或值。例如,编辑可以改变内容中的属性值、向内容添加新属性和/或值等。此类编辑可以创建、删除或修改包含资产、嵌套资产、实例化资产、源资产、引用资产、合并资产、覆盖和/或定义用于共同定义3D虚拟环境的属性和相应值的此类关系。例如,用户可以向层和/或其他资产定义中的属性添加或改变覆盖值,并且该改变可以在属性值解析中传播到任何受影响的资产(例如,通过覆盖另一个资产或层中的值,甚至在客户端106未订阅该其他内容的情况下)。

[0106] 客户端106的内容管理器410可以跟踪客户端106对给定内容项和/或资源做出的所有改变。例如,内容管理器410可以跟踪由用户和/或用软件使用客户端106执行的多个编辑操作。基于这些改变,内容管理器410可以构造一个或更多个消息以发送到内容管理系统104,内容管理系统104包括表示这些改变的数据。在各种示例中,内容管理器410确定从内容管理系统104接收的内容项的版本与包括编辑或改变的内容项的版本(例如,带有时间戳的改变列表)之间的差异。表示这些差异的数据可以包括在消息中而不是整个内容项中。

[0107] 在一些示例中,差异数据可以例如使用可以对资产的一个或更多个版本执行的一个或更多个命令,例如关于场景描述的一个或多个属性-值对(例如,一个或多个结构化元素和/或非结构化元素)的创建命令、删除命令、修改命令、重命名命令和/或重定父级命令(这些命令可以按顺序执行以构建资产的更新版本),来程序性表示资产的更新版本的一个或更多个属性-值对。差异数据还可以表示和/或指示命令将被执行的顺序(例如,通过时间戳或按顺序列出它们)。在各种示例中,一个或更多个命令可以是或可以包括由提供差异信息的客户端106和/或客户端设备的用户执行以本地修改内容的相同命令。此外,该顺序可以对应于和/或是与命令由客户端106执行和/或由客户端设备的用户输入的顺序相同的顺序。

[0108] 附加地或替代地,差异数据可以声明性地表示资产的经更新版本的一个或更多个属性-值对,例如使用经更新的属性-值对、新的属性-值对和/或删除版本和经更新版本之间的属性-值对。在各种示例中,经更新版本的一个或更多个属性-值对可以相对于资产的

先前版本程序性地定义,而经更新版本的一个或更多个其他属性-值对可以声明性地定义。例如,场景图的结构化元素(例如,定义节点和/或节点之间的关系)可以程序性地表示,而场景图的非结构化元素(例如,定义字段和值)可以声明性地表示。

[0109] 例如,根据需要,内容管理器410可以为每个内容项(例如,层)构造描述自相应本地表示最后与外部表示同步以来所做的任何改变的增量(diff)文件。在示例中,用户可以拖动对象,创建对象的位置值的一系列改变。内容管理器410可以仅向内容管理服务器104发送消息以反映内容的一些状态,或者可以发送所有的改变。在任一情况下,可以周期性地或在可用时发送消息,例如以针对到客户端106的内容更新(在一些实施例中,单个消息可以描述对内容的改变的多个状态或版本)实现预定的帧或更新速率(例如,对于每秒30帧,大约每30毫秒)。客户端106的内容管理器410可以生成、传送和应用增量文件到外部源和从外部源(例如,内容管理系统104)生成、传送和应用增量文件,例如以使内容的本地表示与远程和共享的表示对应。

[0110] 从客户端106到内容管理系统104的消息(其编辑或修改内容项(例如,层))可以识别为更新命令。来自内容管理系统104的对来自客户端106的更新命令或读取命令的响应可以包括唯一版本标识符(例如,etag值)。由客户端106的内容管理器410确定的增量或差异可以与特定版本标识符(其可以包括在更新消息中)有关。如果增量到达内容管理系统104并且它与不再是当前的版本标识符相关,则内容管理服务器104可以拒绝更新。这可以被认为错误状况,并且为了让客户端106从这个错误状况中恢复,客户端106可以将内容项的内部表示更新到最当前版本(例如,通过同步)或可以接收最当前版本。内容管理器410然后可以构造相对于该最新版本(例如,etag)的新增量。然后可以提供包括相对于最新版本的差异的更新命令。

[0111] 在至少一个实施例中,为了避免与试图更新相同内容项的其他进程发生竞争条件的可能性,客户端106可以请求使用锁定命令锁定内容(例如,资产和/或相应的文件或资源)。在持有锁时,客户端106可以将更新流式传输到内容管理系统104而无需等待任何确认。在一些实施例中,该锁可以用作在更新之间没有其他进程可以修改内容的保证。客户端106还可以使用解锁命令来解锁内容。在一些示例中,来自不同客户端106的冲突更新可以被数据存储管理器108接受并解决。

[0112] 当内容管理系统104的通信管理器110接收到客户端106的递增更新时,它可以使用订阅管理器402直接将更新(例如,消息和/或差异数据)转发给订阅相应内容的所有其他客户端106(以及在一些实施例中的服务412或渲染器414)。使用这种方法,在分发之前不需要修改更新消息。这可以减少等待时间并且允许内容管理系统104支持大量客户端106并且具有快速更新速率。

[0113] 数据存储管理器108可以跟踪列表中每个内容项(例如,文件或资源)的所有更新。差异确定器408可以周期性地将内容的基础或原始版本和来自一个或更多个客户端106的一系列增量更新合并为内容的新版本。例如,差异确定器408可以使用来自客户端106的数据来本地重构内容项的一个或更多个版本。可以从多个客户端106接收相同内容项的差异,并且可以在内容管理系统104处与内容项的先前共享版本进行组合以确定和/或创建内容项的新版本(例如,共享版本)。如果客户端106对尚未合并的内容执行读取,则它可以接收内容的基本版本和(由一个或更多个服务412和/或客户端106创建的)一系列增量,客户端

106可以将该一系列增量应用于基础内容以重构最新版本。差异确定器408可以以比跟踪内容更新的数据存储管理器108的进程更低的优先级运行——在它可以时使用空闲周期来合并。

[0114] 在各种示例中,创建内容项的新版本可以包括合并对内容项做出的差异或改变的历史。合并的数据可以存储在表示内容项和/或3D虚拟环境的版本的文件和/或资源中。然而,确定内容项和/或3D虚拟环境的新版本可能不一定包括合并差异的历史。例如,在一些实施例中,差异确定器408可以从差异数据的分析(例如,相对于内容的特定版本)导出或识别内容项的特定版本和/或其属性或值(例如,最新共享版本)。

[0115] 合并差异的历史(例如,使用对应的时间戳)可以周期性地发生并且用于持久地存储和访问内容的版本,以及减少存储大小。可以丢弃差异数据以节省存储空间。在一些实施例中,一个或多个客户端106可以请求(例如,在用户或算法的指导下)将3D虚拟环境的版本和/或一个或多个特定内容项持久地存储在内容管理系统104上。

[0116] 在至少一个实施例中,内容管理器410的功能可以被构建到一个或多个客户端106的插件中。然而,内容管理器410的功能的一个或多个方面还可以至少部分地本地集成到一个或多个客户端106和/或主机操作系统或服务,或可以在客户端106外部的其他本地或基于云的软件中。至少部分作为客户端106的插件的内容管理器410适合于将各种各样的游戏引擎、3D建模和动画包、绘画程序和AR/VR库集成到操作环境100中,而不必修改本地代码。例如,这些插件可以用于允许软件使用通过充当集线器的内容管理系统104来回传递的实况更新来相互操作。

[0117] 在各种示例中,内容管理器410可以启用未专门开发用于与共享场景描述格式、API和/或内容管理系统104一起使用的遗留内容创建工具。参照图5来描述示例,其是示出根据本公开的一些实施例的内容管理系统和客户端之间的信息流的示例的框图。

[0118] 在示例中,与客户端106A相关联的内容管理器410A可以在客户端106A处的通用表示502A与内容管理系统104的数据存储114中的相应通用表示502之间建立镜像关系(例如,以便它们表示的内容是同步的)。在通用表示502与客户端106A不兼容的实施例中,内容管理器410A可以另外同步可由客户端106A使用的本地表示506。例如,本地(native)表示506可以是客户端106A的本地内部表示,通用表示502A包括可以在其他客户端106和/或内容管理系统104之间共享的对应描述格式或场景描述语言(例如,USD场景描述)。与客户端106B相关联的内容管理器410B还可在客户端106B处的通用表示502B与内容管理系统104的数据存储114中的对应通用表示502之间建立镜像关系。在该示例中,客户端106B可能够本地使用通用表示502B。

[0119] 对于该示例,假设图3B的显示300B对应于客户端106B并且图3C的显示300C或图3D的显示300D对应于客户端106A。如果用户在对应于显示300B的客户端106B处执行改变场景描述的操作,则内容管理器410B可以对本地共享通用表示502B进行相应的修改。如果启用实况更新,则内容管理器410B可以将增量发布到内容管理服务器104(例如,通过API层406)。如果订阅管理器402确定客户端106A订阅了相同的内容,则内容管理器410A可以接收增量。内容管理器410A可以对共享通用表示502A的本地版本做出相应的改变,并将该改变镜像或传播到客户端106A的本地表示506。因此,客户端106A和106B的用户都可基于客户端106B的用户所做的改变,实况看到关于显示器300B和300C或300D的场景更新。在实施例中,

内容管理器410可以在来自其他用户和/或服务412的更新发生时、以预定间隔或速率、和/或根据需要或指定接收和/或显示它们。

[0120] 虽然该特定示例可能涉及不同客户端设备102上的不同用户,但在其他示例中,一个或更多个客户端106可以在同一机器上使用。以此方式,用户可以根据其能力、优势和/或用户的偏好来使用每个客户端106。在多个客户端106在操作环境100内的公共客户端设备上操作的情况下,在一些实施例中,客户端106可以在与内容管理系统104兼容的内容的公共表示(例如,通用表示502A)上操作,而不是每个都保留和管理单独的副本。类似的概念可以跨本地网络上的机器应用。设想了各种实施例,例如内容管理器410充当管理多个客户端106与内容管理系统104的通信(或管理本地表示)的主机,或者其中每个内容管理器410与内容管理系统104和其他内容管理器410通信。

[0121] 另外,客户端106的一个或更多个用户可能不会主动参与内容创作或者可能不会以传统意义参与。在客户端106是AR客户端或VR客户端的示例中,客户端106和/或相关联的客户端设备102可以基于客户端设备102的取向确定相机变换,并将具有该变换的相机描述发布(例如,由内容管理器410)到由内容管理系统104管理的3D虚拟环境的共享描述。在示例用例中,另一个客户端106(例如,在具有全功能GPU的台式计算机或设备上)和/或渲染器414可以订阅相机并渲染可由相机观看或以其他方式基于订阅内容的场景。所得到的渲染然后可以被流式传输(例如,通过本地WiFi网络)到AR或VR客户端106并且在客户端设备102上显示(和/或到一个或更多个其他客户端106)。使用本文描述的方法,使用任意数量的设备或客户端106的任意数量的用户可以同时观看与移动设备或其他低功率设备的共享虚拟世界,而不受任何单独设备上的受限渲染功率的限制。

[0122] 类似于相机示例,对于VR应用程序,可以基于VR头戴式显示器和/或控制器的位置来摆化身的姿势。内容管理系统104和内容管理器410可以提供双向复制,使得VR用户的化身和/或视图被反映给所有订阅者、AR、VR和非AR或VR(例如,跨异构客户端106)。此外,所公开的实施例使得为特定客户端106开发的工具(例如,程序工具)能够作为代理或服务来操作,这些代理或服务影响具有在不受支持的客户端上反映的变化的共享3D虚拟环境。例如,游戏引擎可以包括可视化脚本工具。一旦支持该工具的客户端106订阅了共享3D虚拟环境,就可以向订阅受影响内容的所有连接的客户端106提供服务。例如,当特定对象进入给定的包围盒或满足某些其他条件时,可以触发可视化脚本工具。该条件可以通过由与托管工具的客户端106不同的客户端106引起的对共享3D虚拟环境的改变来满足。例如,其他客户端106的用户或算法可以将对象移动到该包围盒内,该移动可以被发布到内容管理系统104,并且可以被广播到托管该工具的客户端106,从而触发脚本。工具因此可以对场景进行改变,将它们发布到内容管理系统104,并且效果可以以交互速度显现给所有订阅客户端106。因此工具的执行引擎看起来是本地集成到每个订阅客户端106中的。

[0123] 可以成为代理或服务的工具的另一个示例是约束满足工具。约束满足工具可以提供理解和强制执行门、窗、墙和/或其他对象之间的关系的约束引擎。如果包括该工具的客户端106订阅了共享3D虚拟环境,则可以为所有订阅的客户端106提供约束满足。如果一个客户端106移动了墙,则包括该工具的客户端106可以识别任何约束违反,并且例如,可以对窗、门和/或其他对象的位置进行并发布由此产生的改变。

[0124] 虽然内容管理系统104使用的场景描述可以支持高水平的通用性,但这可能对跨

客户端106的更新的性能带来挑战。例如,对内容的改变可能通过包含资产、嵌套资产、实例化资产、源资产、引用资产、合并资产和/或覆盖影响其他内容。因此,属性和值解析可能会对该过程施加重大负担。根据本公开的实施例,客户端106(和/或内容管理系统104)的内容管理器410可以标记或指定一个或更多个内容项(例如,层、资产、属性、文件,资源)用于快速更新。来自客户端106的这样的指定可以充当内容项将不包括影响属性值解析的一个或更多个方面的改变和/或可以限制内容项包括这样的改变的承诺。通过确定一个或更多个更新满足这些标准(例如,更新仅针对一个或更多个现有属性值),数据存储管理器108可以做出类似的指定。

[0125] 在实施例中,这种受限的改变可以包括对3D虚拟环境的场景描述的结构改变(例如,资产之间的分层关系),其示例可以包括创建或删除内容项中的基元或关系。其他要求可能是内容项(例如,层)对于在属性值解析中定义这些属性是最强大的(例如,最高优先级),和/或内容项仅包含一组固定类型的固定属性的值。通过限制一个或更多个内容项的改变和/或特性,在跨操作环境100传播对内容项的改变时可以避免和/或简化属性值解析。例如,属性值可以使用预先分配的存储直接更新。这种方法在各种场景中可能是有用的,例如对于可以从专门的物理应用程序或服务(例如,服务412和/或内容管理器410)更新变换的物理模拟。

[0126] 懒加载

[0127] 在至少一个实施例中,由客户端106(例如,订阅内容项)接收的内容项的场景描述的一部分可以包括对场景描述的一个或更多个其他部分的引用,以用于并入到内容项中(除了内容项的属性和值)。这些被引用的部分可以对应于其他内容项并且可以被称为有效载荷。如本文所述,有效载荷可以是合并资产,但在一些实施例中,并非所有合并资产都可以是有效载荷。例如,有效载荷可以是一种合并资产并且在一些示例中可以定义或指定为场景描述中的有效载荷。在实施例中,客户端106的内容管理器410可以分析内容项的接收到的场景描述部分,识别对有效载荷的一个或更多个引用,并确定是否使用引用从内容管理系统104请求内容的相应部分。例如,内容管理器410可以确定是否阅读和/或订阅所引用的内容,其本身可以包括附加引用。这可以用于例如通过减少传输到客户端106的数据量来减少带宽需求,管理场景的内存占用,使其在客户端106不会变得太大,和/或仅加载对于期望的显示和/或内容的使用所必需的表示。在一些实施例中,不是有效载荷的其他类型的合并资产可以由于在订阅参考资产中被引用而被自动提供给客户端106,或者可以在客户端106标识参考资产的内容中的引用时被客户端106自动请求和/或订阅。

[0128] 在一些情况下,内容项可以包括一个或更多个引用的元数据,并且内容管理器410可以分析元数据以确定是否请求或订阅附加内容。元数据的示例包括有效载荷(例如,相应对象)在3D虚拟环境中的位置、有效载荷中包括的数据类型(例如,内容项和/或资产)、有效载荷的存储大小或3D虚拟环境中对象的大小,与有效载荷相关联的细节级别、与有效载荷相关联的场景元素或对象的变体等。在一些示例中,元数据可以包括内容项描述中与有效载荷相关联的属性和/或值。

[0129] 作为示例,引用可以对应于在图3C的显示300C上呈现的3D虚拟环境的3D对象。内容管理器410可以分析对应于显示300C的包围盒以确定3D对象对于相机是否可见。当3D对象在包围盒之外时,内容管理器410可以确定不从内容管理系统104请求该有效载荷。另外

或替代地,内容管理器410可以确定在虚拟环境中3D对象离相机足够远,从而它不需要被加载和/或显示。作为进一步的示例,有效载荷的元数据可以识别包括在有效载荷中的内容的类型,并且内容管理器410可以确定客户端106不能显示该类型的内容或对显示该类型的内容不感兴趣。使用这种方法,内容项的各部分可以按需由客户端106接收和加载。例如,该方法不仅可以用于由客户端106接收的内容的初始版本,而且还可以用于对内容项的更新。例如,内容管理器410可以确定不请求对某些有效载荷的更新。

[0130] 元网络实现

[0131] 现在参考图6,图6是示出根据本公开的一些实施例的包括多个内容管理系统的操作环境的示例的图。在图6的示例中,操作环境100包括任意数量的内容管理系统604A和604B到604N(也称为“内容管理系统”)。一个或更多个内容管理系统604可以对应于内容管理系统104。在示例中,一个或更多个内容管理系统604可以在一个或更多个方面彼此不同,例如仅允许客户端106要读取的3D虚拟环境的场景描述部分。

[0132] 如图6所示,一个或更多个内容管理系统604可以包括状态管理器612和/或URI管理器614,如内容管理系统604A中所示。在一些实施例中,使用状态管理器612和/或URI管理器614,内容管理系统604可以作为类似网络的服务运行,例如以存储、生成和向客户端106提供内容。

[0133] 每个客户端106可以通过由通信管理器110管理的标准端口连接到相应的内容管理系统604。数据存储114内的每个内容项(例如,文件或资源)或其部分可以具有相关联的URI,例如,在操作环境100内的URL。客户端106可以使用URI来引用到内容管理系统604的消息中的相应场景描述部分(例如,在读取请求、订阅请求、更新请求中、在其他命令中等等)。URI管理器614可以识别场景描述中对应于URI的部分,并相应地响应来自客户端106的消息,例如通过在响应中包括表示所请求内容的一个或更多个部分的数据,更新相应的内容等。在至少一个实施例中,提供给客户端106并在数据存储114中维护的场景描述可以包括3D虚拟环境内的任何可访问内容项(例如,有效载荷、合并资产,等等)的引用中的URI。

[0134] 在各种示例中,表示所请求内容的一个或更多个部分的数据可以存储在与接收该请求的系统不同的内容管理系统604和/或外部数据存储系统中。URI管理器614可以查找和检索与该其他内容管理系统604和/或外部数据存储系统相关联的URI并在响应中提供该URI。然后客户端106可以使用该URI来从适当的系统检索表示所请求内容的一个或更多个部分的数据。因此,一些客户端请求的内容可以由接收该请求的系统存储,而其他客户端请求的内容可以由不同的系统存储,其中向客户端提供检索该内容的手段(例如,URI)。又例如,接收对内容的请求的系统可以使用URI从另一个系统检索该内容并在响应中将该内容提供给客户端106。作为附加示例,接收对内容的请求的系统可以使用URI将该请求通知给其他系统,并且该其他系统可以在响应中将内容提供给客户端106。

[0135] 同样在各种示例中,一个或更多个内容管理系统604可以使用可以实现高速缓存服务的内容递送网络(CDN)。高速缓存服务可以拦截一个或更多个请求并向客户端106提供内容而不必查询后端服务器。

[0136] 特定内容项内的URI可以对应于存储在任何数量的内容管理系统604和/或其他系统中的内容。客户端106和/或内容管理器410可以使用名称解析系统(例如域名系统(DNS))将URI解析为地址(例如互联网协议(IP)地址),使得相应的消息通过网络120被路由到适当

的内容管理系统604和/或服务器。

[0137] 在至少一个实施例中,URI管理器614包括超文本标记语言(HTML)服务器并且URI包括URL。URL可以在内容项(例如,场景描述文件)内的超链接内。客户端106可以用URL交换内容的适当部分,类似于HTTP服务器允许客户端用URL交换HTML的方式。例如,DNS服务器可用于将URL解析为包括相应内容的适当内容管理系统604的地址。

[0138] 在各种实现中,与HTTP不同,操作环境100实现基本递增的、基于差异的协议。结果,每个内容管理系统604可以包括状态管理器612以维持与客户端106和/或网络会话的状态。为此,状态管理器612可以实现WebSocket服务器、表述性状态转移(REST)Hooks服务器、WebHooks服务器、Pub-Sub服务器或其他基于状态的管理解决方案的功能。在实施例中,可以使用双向有状态协议。例如,客户端106和内容管理系统604之间的会话可以通过持久的WebSocket连接来实现。由状态管理器612维护(例如,记录和跟踪)的用于连接到内容管理系统604的状态可以包括身份验证,以及发布/订阅模型的订阅集及其相应的版本标识符(例如,etags)。状态管理器612可以跨一个或更多个服务器112实现并且可以将作业或任务移交和/或分配给相同或不同内容管理系统604内的各种服务器和/或实例(例如,用于负载均衡目的)。这可以包括状态管理器612将与作业相关联的各种状态数据中的任何一个传递给那些服务器。

[0139] 本文描述的方法可用于实现高性能和实用的真正3D因特网。传统因特网基本上是二维(2D)和无状态的。当有关网页的某些内容发生变化时,该页面将完全重新加载。这是有效的,因为2D网页通常尺寸较小并且本质上并不复杂。然而,3D虚拟环境可能非常复杂和庞大。将此类内容集成到用于2D网页的传统互联网架构中可能会导致动态3D内容的加载时间过长,文件传输和处理时间过长。

[0140] 几十年来,计算机图形社区一直试图将3D内容集成到用于2D网页的传统互联网架构中。早期的尝试包括1994年的虚拟现实建模语言(VRML)和1997年的Web3D联盟。最近的例子包括Khronos Group标准,如WebGL、WebVR和GL传输格式(gLTF)。经过所有这些时间和共同努力,3D Web技术仍然很少被采用。这可能是由于这些解决方案的性能有限以及视觉质量低,部分原因是3D内容的图元表示。

[0141] 然而,根据所公开的实施例,通过使用到内容管理系统604的有状态连接,结合对内容、名称解析和3D虚拟环境的丰富描述的递增更新,可以实现真正的3D互联特网的高性能和实用基础。此外,在各种实施例中,可以跨不同系统和3D虚拟环境以及跨不同交互引擎来促进用户和客户端之间的交互体验,这些交互引擎可以使用非常不同且可能不兼容的规则和软件来促进用户与3D内容的交互。例如,内容和交互可以在游戏引擎和3D虚拟环境以及其他非面向游戏的引擎之间共享。内容项的场景描述部分中的超链接可以引用整个3D虚拟环境(例如,对3D虚拟环境的所有场景描述的顶层引用),例如USD舞台和/或场景图,它们可以由不同的内容管理系统604托管。软件可以基于在场景描述中指定链接的方式(例如,通过元数据、指令、指示符、上下文等)来处理链接和/或相应的内容。

[0142] 作为进一步的示例,链接可以指由不同的内容管理系统604托管并且嵌入在另一个3D虚拟环境中(例如,用于同时显示和/或互操作性)的内容项或3D虚拟环境。此外,此类链接可由客户端106和/或外部应用程序或服务用于在客户端106内加载3D虚拟环境的一个或更多个部分。例如,用户可点击网页浏览器内容内的链接、电子邮件、文件系统的显示或

在另一个应用程序或服务中,并且作为响应该软件可以导致在该软件或另一个应用程序或服务中加载和/或显示3D内容。

[0143] 分层元素的增量传播

[0144] 如本文所述,在至少一个实施例中,由客户端106的内容管理器410确定的增量或差异可以与特定版本标识符(其可以包括在更新消息中)有关。如果增量文件到达内容管理系统104并且它与不再是当前的版本标识符相关,则内容管理系统104可以拒绝更新。这可以被认为是错误状况,并且为了让客户端106从这个错误状况中恢复,客户端106可以将内容项的内部表示更新到最新版本(例如,通过同步)或可以接收最新版本。内容管理器410然后可以构造相对于该最新版本(例如,etag)的新增量。然后可以提供包括相对于最新版本的差异的更新命令。

[0145] 虽然这种方法在许多实施例中可能是足够的,但在某些情况下它可能引入延迟,因为在向内容管理系统104发送更新之后,在安全地发送后续更新之前,客户端106可能需要等待更新被应用的确认。此外,由于客户端106的内容管理器410可以在版本标识符不再是当前的时候构造新的增量,用于构造和传送旧的增量的计算资源可能被浪费。如果客户端106具有到内容管理系统104的高延迟连接,则这些问题可能会加剧。通过允许内容管理器410在接收关于先前更新的确认之前发送更新,可以使用锁定来缓解这些问题。然而,锁定可以使用客户端106的内容管理器410的手动锁定/解锁,从而在实现锁定机制时引入复杂性。此外,锁定可防止多于一个客户端同时更新内容项(例如,场景)。当一个客户端106正在修改内容并且其他客户端106处于“仅观看”模式时,这可能是合适的。

[0146] 根据本公开的至少一个实施例,可以向来自客户端106的内容管理器410的内容的更新(例如,增量信息集)分配定义客户端106要应用更新的顺序的值。使用这些值,客户端106的内容管理器410可以各自导出应用任何特定更新的相同顺序,使得内容的同步版本(例如,内容项、场景图和/或层)可以在客户端106处生成。使用所公开的方法,更新不需要被内容管理系统104拒绝,并且客户端106可以向内容管理系统104发送任意数量的更新而无需等待确认之前的更新。

[0147] 现在参考图7,图7是示出根据本公开的一些实施例的用于同步3D虚拟环境的内容的版本的示例的数据流程图。在各种示例中,版本同步器720可以用于将值分配给更新。版本同步器720可以在内容管理系统104和/或客户端设备102中的一个或多个上实现(例如,在对等实施例中或者服务器托管在客户端设备上的情况下)。

[0148] 由版本同步器720分配的值可以形成值序列,该值序列定义将增量信息集应用于内容(例如,场景描述,例如场景图)以产生内容的同步版本的顺序。此外,每个值可以对应于内容的版本。在实施例中,给定内容的版本及其在序列中的值,任何版本的内容可以由各种内容管理器410中的任何一个通过按照来自序列的相应值所指示的顺序应用插入的增量信息集来生成。

[0149] 在一些实施例中,由版本同步器720分配的值可以包括数字(例如,整数或浮点)和/或字母,版本同步器720针对增量信息集到值的每次分配而递增。例如,可以依次为第一增量信息集分配值1,可以向第二增量信息分配值2,等等。具有第一和第二增量信息集的客户端106的内容管理器410然后可以基于这些值来确定应用更新的顺序(例如,在具有较晚值的更新之前应用具有序列中较早值的更新,以及应用更新而不跳过序列中的任何值)。在

一些示例中,可以使用公式或其他算法来导出序列中的值和/或使用这些值应用增量信息集的顺序。

[0150] 在一些实施例中,版本同步器720可以至少基于接收更新的顺序来将值分配给更新。在一些示例中,分配可以按照接收更新的顺序。在进一步的示例中,当多个更新已被接收但尚未分配值和/或分配尚未提供给客户端106时,可以以与接收更新的顺序不同的顺序向更新分配值(例如,由于某些更新的处理延迟、并行处理、乱序处理等)。

[0151] 在各种示例中,客户端106可以在更新请求中传送(例如,通过传输基础设施420)增量信息集。客户端106可以标记或以其他方式存储(例如,本地地)更新未确认或未决的指示。客户端106在一个或更多个后续更新请求中发送额外的增量信息集之前不需要等待响应,并且可以类似地存储那些更新未确认或未决的指示。

[0152] 当版本同步器720从客户端106接收更新请求时,版本同步器720可以将序列中的值分配给更新。在客户端106已发送多个更新请求的情况下,版本同步器720可能不会按照客户端106发送更新请求的顺序来处理它们。响应于该请求,可以提供更新的值(例如,通过传输基础设施420传送)到客户端106。在一个或更多个其他客户端106订阅该内容或以其他方式与该内容相关联或对该内容进行更新的实施例中,该增量信息集和/或值可以被提供(例如,推送)给每个其他客户端106。

[0153] 根据实现方式,增量信息集和/或值可以由接收信息的另一个客户端106提供和/或可以由内容管理系统104的服务器直接提供给每个客户端106。在图7的示例中,内容管理系统104可以向每个其他客户端106提供增量信息和值。在其他示例中,做出更新请求的客户端106可以向一个或更多个其他客户端106提供增量信息(例如,在接收到来自版本同步器720的确认或值之前或之后)。

[0154] 如本文所述,每个客户端106的内容管理器410可以维护用于跟踪和/或确定未确认的增量信息集(例如,增量信息集没有关联的值)的列表、记录或其他数据。当内容管理器410接收到增量信息集的值时,内容管理器410可以更新数据以反映该确认。此外,由于各种潜在原因,内容管理器410可以或可以不将相应更新应用于内容的本地副本。例如,根据值序列,内容管理器410可能还没有要以在经确认更新之前的顺序应用的值和/或相应的更新。一旦接收到中间(intervening)信息,内容管理器410就可以按顺序应用每个更新。此外,即使在每个中间更新已被接收和确认的情况下,内容管理器410也可以延迟应用更新。例如,内容管理器410可以响应于应用更新的用户命令,使用批处理,和/或基于可能引入延迟的其他因素(这可能在客户端106之间变化)周期性地应用更新。在客户端106未被配置为或不能够对内容贡献更新的实施例中,客户端106的内容管理器410可不包括用于跟踪未确认更新的能力,但仍可包括确保按顺序更新应用的功能。

[0155] 将过程700描述为示例,在过程700开始时,客户端106A和客户端106B可各自具有内容版本和与该内容版本相关联的值序列中的值(例如,内容版本的值)。在其他示例中,客户端106A和客户端106B可以具有不同版本的内容。同样作为示例,客户端106A和106B没有任何未确认或未应用的内容的增量信息集。然而,在其他示例中,客户端106中的一者或两者可具有由客户端106接收或从客户端106传输至内容管理系统104的一个或更多个未确认或未应用的增量信息集。

[0156] 如图7所示,客户端106A的内容管理器410A可以生成增量信息702A并将其发送到

内容管理系统104。客户端106A的内容管理器410A可以至少基于发送增量信息702A来执行未确认的更新修订710A。未确认的更新修订710A可以是内容管理器410A可以用来跟踪和/或确定对内容管理器410A和/或客户端设备而言是本地但尚未被内容管理系统104确认的一个或更多个增量信息集的列表、记录或其他数据。例如,未确认的更新修订710A可以记录增量信息702A尚未被内容管理系统104确认。虽然在发送增量信息702A之后示出了未确认的更新修订710A,但在其他示例中,未确认的更新修订710A可以在发送增量信息702A之前发生。

[0157] 还如图7所示,在从客户端106A发送增量信息702A之后,客户端106B的内容管理器410B可以将增量信息702B发送到内容管理系统104。客户端106B的内容管理器410B可以至少基于发送增量信息702B来执行未确认的更新修订714A,其可以类似于未确认的更新修订710A。例如,未确认的更新修订714A可以是列表、记录或其他数据,内容管理器410B可以使用这些列表、记录或其他数据来跟踪和/或确定对于内容管理器410B和/或客户端设备是本地但尚未由内容管理系统104确认的一个或更多个增量信息集。

[0158] 在该示例中,尽管已经在增量信息702A之后发送,但是内容管理系统104可以在增量信息702A之前接收增量信息702B。版本同步器720可以被配置为至少基于内容管理系统104接收增量信息集的顺序(例如,接收集的顺序)为增量信息集分配值。例如,响应于接收到增量信息702B,版本同步器720可以执行值704B到增量信息702B的值分配718A。在至少一个实施例中,这可以包括将序列的先前值递增到序列中的值704B(或者该值可以是先前增加的,例如,当将先前值分配给增量信息集时)。同样响应于接收到增量信息702B,内容管理系统104可以将值704B发送到与增量信息702B相关联的客户端106B(提供更新的客户端)。

[0159] 在至少一个实施例中,客户端106B的内容管理器410B可以基于接收到值704B,执行未确认的更新修订714B以记录增量信息702B已经被确认。同样至少基于接收到值704B,客户端106B的内容管理器410B可以使用增量信息702B对客户端106B上的内容版本执行内容更新716A。内容更新716A可以至少基于在对应于序列中内容的版本的值之后(例如,紧随其后)的值704B来执行。此外,在一些示例中,未确认的更新修订714B可以在内容更新716A和/或内容更新716B(例如,使用批或定期更新)之后发生和/或作为其一部分而发生。

[0160] 在至少一个实施例中,还响应于接收到增量信息702B,内容管理系统104可以将值704B和增量信息702B发送到一个或更多个其他客户端106。例如,内容管理系统104可以将值704B和增量信息702B发送到订阅内容的每个客户端106。因此,如图所示,客户端106A可以接收增量信息702B和值704B。客户端106A的内容管理器410A可以使用增量信息702B在客户端106A上执行内容版本的内容更新712A。

[0161] 同样在过程700中示出,客户端106A的内容管理器410A可以在接收增量信息702A的值和/或确认之前将增量信息702C发送到内容管理系统104。版本同步器720可以对增量信息702A执行值704A的值分配718B。在至少一个实施例中,这可以包括递增值704B,该值可以是序列中的先前值。同样基于接收到增量信息702A,内容管理系统104可以将值704A传送到与增量信息702A相关联的客户端106A。此外,内容管理系统104可以将值704A和增量信息702A传送到任何其他客户端106(例如,所示示例中的客户端106B),例如订阅内容的客户端。

[0162] 客户端106A的内容管理器410A可以至少基于接收到值704A,执行未确认的更新修

订710B以指示增量信息702A已经被确认,和/或记录值704A。客户端106A的内容管理器410A还可以使用增量信息702A来执行内容更新712B。此外,客户端106B的内容管理器410B可以至少基于接收到值704A和增量信息702A,使用增量信息702B执行内容更新716A和使用增量信息702A执行内容更新716B。版本同步器720可以执行值704C到增量信息702C的值分配718C,将值704C提供给客户端106A,并且将值704C和增量信息702C提供给客户端106B。

[0163] 过程700可以继续,因为附加的增量信息集被传送到内容管理系统104。虽然过程700可以涉及单个内容项,但是内容管理系统104可以执行具有任意数量内容项的类似过程(例如,与图7的内容项同时地)。这些过程可以包括与过程700相同和/或不同的客户端106中的一个或更多个,并且具有用于对增量信息的应用进行排序的单独的值序列。例如,不同的客户端106可以订阅不同的内容项,这些内容项都可以是公共虚拟环境和/或场景描述的一部分。此外,一些客户端106可能正在与其他不同的虚拟环境上操作(例如,协作和/或观看),或者可能没有订阅虚拟环境的所有内容。

[0164] 公开的方法在客户端106发送、生成和应用对内容的更新的方式、顺序和定时方面提供显著的灵活性,同时提供内容的版本之间的同步。图7示出了一些示例,但不旨在限制所示示例。

[0165] 如本文所述,增量信息集可程序性表示资产的更新版本的一个或更多个属性-值对,例如使用可对资产的一个或更多个版本执行的一个或更多个命令,例如关于场景描述的一个或更多个属性-值对(例如,一个或更多个结构化元素和/或非结构化元素)的创建命令、删除命令、修改命令、重命名命令和/或重定父级命令,可以按顺序执行这些命令以构建资产的更新版本。差异数据还可以表示和/或指示命令将被执行的顺序(例如,通过时间戳或按顺序列出它们)。在各种示例中,一个或更多个命令可以是或可以包括由提供增量信息集的客户端106和/或客户端设备的用户执行以本地修改内容的相同命令中的至少一些。此外,该序列对应于由客户端106执行和/或由客户端设备的用户输入命令的序列和/或是与该序列相同的序列。在其他示例中,可以修改或优化这些命令以捕获等效结果。

[0166] 现在参考图8-10,方法800、900和1000的每个框以及本文描述的其他方法包括可以使用硬件、固件和/或软件的任何组合来执行的计算过程。例如,各种功能可以由执行存储在存储器中的指令的处理器来执行。该方法还可以体现为存储在计算机存储介质上的计算机可用指令。这些方法可以由独立应用程序、服务或托管服务(独立地或与另一托管服务组合)或另一产品的插件提供,仅举几例。此外,通过示例的方式,关于操作环境100和图7描述了这些方法。然而,这些方法可以附加地或替代地由任何一个系统或系统的任何组合来执行,包括但不限于本文所述的那些。

[0167] 图8是示出根据本公开的一些实施例的客户端可以用于更新内容的同步版本的方法800的流程图。在框B802,方法800包括在3D虚拟环境的场景图的版本之间传送增量信息。例如,客户端106A可以将三维(3D)虚拟环境的场景图的版本之间的增量信息702A传送到内容管理系统104。

[0168] 在框B804,方法800包括接收分配给增量信息的值,该值定义将增量信息集应用于场景图以产生场景图的同步版本的顺序。例如,客户端106A可以接收指示分配给增量信息702A的值704A的数据。如本文所述,值704A可以属于值序列,该值序列定义将增量信息集应用于场景图以产生场景图的同步版本的顺序。

[0169] 在框B806,方法800包括至少基于该值生成场景图的同步版本。例如,客户端106A可以执行内容更新712B以至少基于使用值704A按顺序将增量信息702A应用于场景图来生成场景图的同步版本中的同步版本。

[0170] 现在参考图9,图9是示出根据本公开的一些实施例的服务器可用于更新内容的同步版本的方法900的流程图。在框B902,方法900包括接收3D虚拟环境的场景图的版本之间的增量信息。例如,内容管理系统104可以从客户端106A接收3D虚拟环境的场景图的版本之间的增量信息702A。

[0171] 在框B904,方法900包括向增量信息分配值,该值定义将增量信息集应用于场景图以产生场景图的同步版本的顺序。例如,版本同步器720可以将值704A分配给增量信息702A。如本文所述,值704A可以属于值序列,该值序列定义将增量信息集应用于场景图以产生场景图的同步版本的顺序。

[0172] 在框B906,方法900包括传送该值,该传送使客户端使用该顺序将增量信息应用于场景图。例如,内容管理系统104可以向客户端106A发送指示值704A的数据。该传送可以使客户端106A使用内容更新712B中的顺序将增量信息702A应用于场景图。

[0173] 现在参考图10,图10是示出根据本公开的一些实施例的用于管理内容版本的同步的方法1000的流程图。在框B1002,方法1000包括存储表示3D虚拟环境的数据。例如,内容管理系统104可以将表示3D虚拟环境的场景图的数据存储在数据存储中。

[0174] 在框B1004,方法1000包括建立与一个或多个客户端的双向通信信道。例如,内容管理系统104可以与客户端106A和106B建立双向通信信道。双向通信信道可用于从客户端106A和106B接收场景图版本之间的增量信息集。双向通信信道还可以向客户端106A和106B提供值序列中的值和增量信息集之间的分配,以将场景图的同步版本传播到客户端106A和106B。

[0175] 在框B1006,方法1000包括通过双向通信信道接收场景图的版本之间的增量信息集。例如,内容管理系统104可以从客户端106A和106B接收场景图的版本之间的增量信息集。

[0176] 在框B1008,方法1000包括向客户端提供值序列中的值和增量信息集之间的分配以将场景图的同步版本传播到客户端。例如,内容管理系统104可以向客户端106A和106B提供由版本同步器720在值序列中的值和增量信息集之间做出的分配,以将场景图的同步版本传播到客户端106A和106B。

[0177] 增量格式的示例

[0178] 在至少一个实施例中,增量信息集可以包括定义对场景描述的一个或多个结构化元素的一个或多个改变的部分和定义对场景描述的一个或多个非结构化元素的一个或多个改变的部分。如本文所述,结构化元素可对应于场景图的图节点,以及节点之间所示的互连。如本文所述,非结构化元素可以指分配给节点和/或结构化元素的属性和/或值(例如,字段-值对)。非结构化元素通常不会影响场景图的结构,而结构化元素可以定义场景图的结构。在图2A和图2B中,资产可以是结构化元素并且属性-值对可以是非结构化元素。

[0179] 在各种实施例中,可以程序性地定义或指定对增量信息集的结构化元素的一个或多个改变。例如,一个或多个创建命令、删除命令、修改命令、重命名命令和/或重定父

级命令可以相对于一个或更多个结构化元素被顺序地定义。由于可以在增量信息集中程序性地定义对结构化元素的更新,内容管理器410和/或内容管理系统104可以按照增量信息集中定义的顺序应用更新,从而跨操作环境100的不同组件提供场景描述的一致结构化配置。例如,给定结构化元素A和B,A被重命名为C,B被重命名为A,然后C被重命名为B,结果是结构化元素彼此被重命名,如果它们没有由所有组件按该特定顺序应用,则会有不同和不一致的结果。

[0180] 在一些实施例中,由于客户端106可以同时修改场景描述的相同同步版本,然后生成并发送相应的增量信息集,所以可能会出现冲突。例如,一个客户端106可以生成删除结构化元素的第一增量信息集,而另一个客户端106可以生成将非结构化字段-值对分配给结构化元素的第二增量信息集。如果由版本同步器720分配给第一和第二增量信息集的值定义了在第一集之后应用第二集的顺序,则该命令可以对不存在的元素进行操作。为了解决这个问题,应用增量信息集的操作环境100的每个组件可以被配置为在应用增量信息集时应用一组公共的冲突解决规则。在本示例中,每个组件可以忽略或丢弃用于解决冲突的不存在元素的命令。

[0181] 在至少一个实施例中,可以声明性地定义或指定对增量信息集的非结构化元素的一个或更多个改变。在至少一个实施例中,增量信息集中改变的每个非结构化元素可以用它的最终值一次性指定。例如,如果客户端106在编辑场景描述的本地副本时多次改变来自场景描述的当前版本的字段的值,则内容管理器410可以在对应的增量信息集中仅包括字段描述的最近、最后或最近值。声明性地指定非结构化元素可以减小增量信息集的大小,同时仍然导致操作环境100的组件之间的一致结果。然而,如本文所述,对于结构化元素,客户端106可以包括程序性按照它们发生的顺序对场景描述做出的所有改变。虽然在一些情况下,内容管理器410可以压缩或优化程序改变,但是发送所有改变可以通过减少处理来允许更快的传输时间。

[0182] 在至少一个实施例中,场景描述(例如,场景图)的每个节点可以具有唯一标识符(ID)。在一些实施例中,节点的唯一ID可以在创建节点时(例如,在创建命令中)分配给节点。唯一ID可用于节点的整个生命周期,无论是重命名、删除还是重定父级。可以使用节点的唯一ID指定节点的结构改变和/或属性-值对(例如,字段和/或字段值)到节点的改变和/或分配。在一些实施例中,唯一ID可由创建节点的客户端106生成和/或分配给该节点。例如,节点的唯一ID(可以更一般地称为节点ID)可以是随机生成的64或128位数字。因此,为了改变节点的字段的字段值,增量信息集可以包括节点ID、字段ID和字段值。

[0183] 包括分层元素的内容的数据存储的示例

[0184] 数据存储114可以使用各种可能的格式和方法来存储场景描述。在一些示例中,可以使用键-值结构来捕获对场景描述的任何改变。例如,在场景描述包括分层元素的情况下,它们可以被坍塌(collapse)成键-值对,键-值对可以存储在数据存储114中。为了说明上述内容,桌/碗/颜色=蓝色可以表示分配给桌子的颜色分配给碗的值的分配。但是,当允许的改变包括场景描述中分层元素的节点的重命名和/或重定父级时,这种方法可能会很复杂。例如,如果一个客户端106将碗重定父级计数器,那么键将是计数器/碗/颜色。然而,还不知道该改变的另一个客户端106可以更新旧键。重命名可能会出现类似的问题。公开的方法允许在避免这些潜在问题的同时重命名和/或重定父级。

[0185] 根据本公开的一些方面,数据存储114可以使用节点ID来存储和引用场景描述的结构化元素(节点),并且可以将非结构化元素作为字段-值对分配给节点ID。字段-值对可以用作键-值对,除了数据存储中的单个键-值对之外,键-值对可以是每节点ID或每节点。例如,节点可以存储在与数据存储库114中的键-值对分开的结构或表中。当客户端106引用节点时,客户端106可以引用节点ID和一个或多个相关字段-值对,其中节点ID允许识别正确的节点,即使该节点被重定父级或重命名。

[0186] 现在参考图11,图11是示出根据本公开的一些实施例的结构1100的示例的图,该结构1100可以被数据存储使用以捕获表示分层元素的对象1102。

[0187] 在各种示例中,每个对象(例如,对象1102)可以表示场景图、分层数据结构的根、文件、场景描述、层和/或3D虚拟环境或部分或其版本。例如,对象1102的每个版本本身可以是包括图11中所示的元素的对象。如图所示,对象1102可以包括版本标识符1104、父标识符1106、版本名称1108、当前版本1110、创建的版本1112和一个或多个指向节点1114的指针,其示例包括节点1104A。

[0188] 如本文所述,每个节点(例如,节点1114A)可包括节点标识符1116,其可由客户端106使用以引用该节点。可以包括在节点中的数据的其他示例是父标识符1118、节点名称1120、节点类型1122、节点顺序1124、第一版本标识符1126、最新版本标识符1128、指向一个或多个字段1130的一个或多个指针,以及指向一个或多个时间样本1132的一个或多个指针。

[0189] 节点名称1120可以包括节点的名称。由于节点名称1120与节点ID1116分开,因此可以重命名节点,同时保留节点ID 1116,如本文所述。父标识符1118可以包括节点的父节点的节点ID 1116。节点类型1122可以指定节点的类型(例如,是否是结构化元素的类型,其示例在本文描述)。节点顺序1124可以指定节点的顺序。在一些示例中,节点顺序1124可以在遍历场景图时由内容管理器410使用并且可以指定或定义遍历子节点的顺序。节点顺序1124可用于说明多个客户端106同时修改结构(例如,添加、移除或重新排序节点)以确保所有客户端106以相同顺序应用节点的情况。第一版本标识符1126可以指定节点出现在其中的对象1102的第一版本。最新版本标识符1128可以指定在其中更新节点的对象1102的最后版本(节点的任何字段)。如果节点的最新版本已经存在(例如,基于确定当前版本标识符>=最新版本标识符1128),则最新版本标识符1128可用于跳过节点的处理。

[0190] 字段1130A是可以分配给节点1114A的字段1130之一的示例。每个字段(例如,字段1130A)可以包括字段名称1140、字段值1142和版本标识符1144。

[0191] 时间样本1132A是可分配给节点1114A的时间样本1132之一的示例。每个时间样本(例如,时间样本1132A)可以包括时间1150、值1152和版本标识符1154。

[0192] 图11的结构1100是支持对象版本的实现的示例。如本文所述,对象的版本可以持久地存储在内容管理系统104上,例如,响应于在用户或算法的指导下的请求。对象1102的版本标识符1104可以唯一地标识对象1102的版本。在各种示例中,数据存储管理器108可以将版本值分配给数据存储114中的对象,其相对于特定对象是按顺序的。用于在数据存储114中存储和引用对象的版本值可以与分配给增量信息集的值不同或相同,并且可以出于各种原因创建对象1102的新版本。

[0193] 现在参考图12A,图12A是示出根据本公开的一些实施例的对象1102的版本的示例

的图。例如,具有版本标识符(ID)1104的对象1102可以是图12中所示的其他对象的父对象。在各种实施例中,对象1102的版本可以分支,其中一个对象可能只有一个父级,但可以有多个子级。例如,对象1102具有包括对象1206和对象1208的子级。对象1208还包括包含对象1210和对象1212的子级。

[0194] 数据存储管理器108可以支持快速分支切换,其中如果存在同一父级的多个子级,则将客户端106从一个子级转换到另一个子级可以通过生成和提供增量信息集来实现,该增量信息集可以由客户端106应用于子级以产生另一个子级。例如,为了从对象1206切换到对象1208、1210或1212,数据存储管理器108可以在对象1102的版本之间生成增量信息集。增量信息集可以类似于或不同于用于跨客户端同步场景描述的各版本的增量信息。在一些示例中,对结构化元素和非结构化元素的改变可以分别被声明性地捕获,或者结构性改变可以被程序性地捕获。使用快速分支切换,内容管理系统104不需要从父版本发送任何数据。

[0195] 如本文所述,数据存储管理器108可以将版本值分配给数据存储114中的对象,其相对于特定对象是顺序的。在实施例中,数据存储管理器108可以分配版本值,使得子级相对于子级的父级的每一个在序列中具有较晚的值(例如,更大的版本号)。但是,遍历特定分支可能存在间隙,即使每次分配版本值都会递增序列。例如,在图12A中,从版本标识符1104为92的对象1102的分支开始,沿着该分支,对象1208的版本标识符1104可以是93并且对象1212的版本标识符1104可以是94。然而,沿着另一个分支从版本标识符1104为92的对象1102来看,可能存在对象1206的版本标识符1104为96的序列中的间隙。这可表明对象1206是在创建了图12A中的其他版本之后从对象1102分支出来的。尽管示出和描述了特定的排序方案,但可以采用其他方案,例如从每个父节点和/或分支开始新的子序列。例如,可以使用各种方法,包括足以用于唯一标识版本、父级和子级之间的关系和/或版本之间的时间关系(例如创建顺序)的方法。

[0196] 根据至少一些实施例,当数据存储管理器108创建对象1102的新版本,例如对象1208时,新对象1102的父标识符1106可以被设置为对象的先前版本。与存储字段1130和时间样本1132的所有数据不同,新对象可以仅存储从先前版本改变的内容,其余内容可以使用包括在结构1100的元素中的指针来捕获。

[0197] 对象的版本名称1108可用于引用该对象。例如,客户端106的内容管理器410可以使用对象的版本名称1108来引用该对象。在一些示例中,数据存储管理器108仅允许叶对象具有名称,并且仅叶对象可以由内容管理器410编辑。当内容管理器410和/或数据存储管理器108复制对象时,它可以创建名称到对象ID(其可以指版本标识符1104)映射的第二条目,其中两个映射都指向相同的现有对象。当内容管理器410和/或数据存储管理器108更新副本之一时,可以创建新对象以捕获任何改变,其中现有对象被设置为其父级。

[0198] 现在参考图12B,图12B是示出根据本公开的一些实施例的对象1102的版本的示例的图。在各种实施例中,数据存储管理器108可以存储节点和/或值,使得来自对象的不存在或缺失的节点和/或字段可以向数据存储管理器108指示来自父级(直接或间接)的字段或时间样本的对应字段值将用于该对象版本。作为示例,在对象1102中,为节点1114A的字段1130B(“字段B”)存储“5”的字段值1142。然而,在对象1208中,没有为节点1114A的字段1130B(“字段B”)存储数据,表明字段1130B应该被包括在对象1208的节点

1114A中。特别地,字段值1142和字段1130B将从包括字段1130B的数据(在这种情况下是对象1102)的最近的父级中检索并由其定义。

[0199] 类似地,对于对象1208的节点1114B,具有字段值的字段1130A和1130B将由对象1102的节点1114B包括,因为对象1208的节点1114B不包括字段1130A和1130B的数据。节点名称的处理方式类似于节点和/或字段名称。因此,如图12B中所示,对于对象1208,节点1114A的节点名称1120变为“Chair1 (椅子1)”。此外,对于对象1102和对象1208,节点1114B的节点名称1120都是“table (桌子)”。从父级添加的节点和/或字段值也可以存储在子级中。例如,对象1208的节点1114C添加字段1130C。从父节点中删除的节点和/或字段值可以在子节点中明确标记为已删除或通过存在但为空白来指示。例如,对象1208的节点1114E中的字段1130B存在,但没有值(为空)以向数据存储管理器108指示字段1130B在对象1208中已被删除并且不被包括在对象1208的节点1114E中(或其子节点中,除非重新声明)。

[0200] 使用所公开的方法,可以显著减少不同版本对象的存储大小。例如,磁盘上的对象可能只有几个字段,但它可指向具有更多字段要包括在对象中的父对象,而该对象本身可指向具有附加字段的另一个对象,一直沿版本链向上。当客户端106的内容管理器410连接到内容管理系统104以接收对象的版本时,如果客户端106没有对象的另一个版本(这可以由客户端106指示),则数据存储管理器108可以合并对象版本以生成表示对象版本的基本数据,并且可以将基本数据传送到客户端106。

[0201] 在一些示例中,可以至少部分地在客户端106连接(例如,参与协作编辑和/或查看动态场景)到内容管理系统104和/或连接到对象的版本以在客户端106连接时或如果连接时减少等待时间之前,生成基础数据。此外,基础数据可以定期更新和/或响应客户端106连接请求以传送到一个或更多个客户端106。如果客户端106确实具有对象的另一版本(其可以由客户端106指示或指定),数据存储管理器108可以生成表示客户端106处的对象版本与对象的期望版本之间的差异的差异数据。例如,差异数据可以捕获将客户端版本转换为对象的期望版本所需的最小命令集。因此,内容管理系统104不需要发送在协作创建对象的期望版本时在客户端106之间交换的所有增量。

[0202] 版本高速缓存

[0203] 所公开的方法还可以为跨服务器和/或边缘设备的数据高速缓存对象提供益处,这些服务器和/或边缘设备可能彼此远离。例如,如果内容管理系统104的主服务器或核心服务器在洛杉矶并且内容管理系统104的边缘或高速缓存服务器在莫斯科,则将数据快速传输到高速缓存服务器以用于对象的本地托管可能具有挑战性。根据一些实施例,对象的版本可以在客户端106连接或对对象的特定版本的请求之前被高速缓存在高速缓存服务器中。如果客户端106请求未高速缓存的版本,则核心服务器可以发送从高速缓存版本获得对象的请求版本所需的差异数据。

[0204] 在各种实施例中,当读取请求从客户端106到达时,高速缓存服务器可以首先检查该高速缓存以查看该请求是否可以由高速缓存直接服务。如果是,服务器可立即用到大文件传输(LFT)的重定向进行响应。LFT可以指服务器告诉客户端106通过向带外高速缓存服务器提供URL(例如,其中高速缓存服务器可以是HTTP高速缓存服务器)而通过带外高速缓存服务器读取数据的方法。小文件(例如,小于4KB或其他LFT阈值)可以直接带内返回(例如,通过WebSockets或其他形式的直接传输),而不是通过LFT过程。

[0205] 例如,如果高速缓存中没有直接答案,则高速缓存服务器可以查看高速缓存中的版本,并估计从这些版本到最新版本的增量大小,以及从客户端106具有的版本到最新版本的大小。所有这些信息都可以用于向客户端106(例如,在单个差异文件中)递送最优的增量序列(例如,最小的总大小)。例如,如果客户端106具有版本0,高速缓存具有版本0- $\rightarrow$ X,并且最新版本是Y,则高速缓存服务器可以要求从版本0- $\rightarrow$ Y和版本X- $\rightarrow$ Y的估计。高速缓存服务器也可以从高速缓存中知道版本0- $\rightarrow$ X的大小。如果版本0- $\rightarrow$ Y的大小小于版本大小(0- $\rightarrow$ X+X- $\rightarrow$ Y)的一半,则版本0- $\rightarrow$ Y可被写入高速缓存并返回。如果版本X- $\rightarrow$ Y的大小小于LFT阈值,则版本0- $\rightarrow$ X可以通过WebSocket或其他直接传输方法作为重定向递送到LFT和版本X- $\rightarrow$ Y。如果版本X- $\rightarrow$ Y的大小大于LFT阈值,则版本0- $\rightarrow$ X和版本X- $\rightarrow$ Y都可以作为重定向被递送到LFT。

[0206] 作为进一步的示例,假设客户端106具有版本15,高速缓存具有版本0- $\rightarrow$ 10、版本10- $\rightarrow$ 20和版本20- $\rightarrow$ 30,并且最新的是版本40。在一种方法中,可以向客户端106提供版本15- $\rightarrow$ 40的新增量。在另一种方法中,可以向客户端106提供版本15- $\rightarrow$ 20的新增量、版本20- $\rightarrow$ 30的现有增量和版本30- $\rightarrow$ 40的新增量。高速缓存服务器可以估计这些方法两者的大小。第一种方法总是更小,但如果它不是小很多,那么第二种方法可能更好,因为它导致另一个客户端106稍后可以使用的增量30- $\rightarrow$ 40。在一些实施例,高速缓存服务器可以基于确定第一种方法和第二种方法之间的大小比小于阈值(例如,0.5)来选择第一种方法。新的增量可以通过LFT或通过WebSocket或其他直接传输发送,具体取决于大小。

[0207] 高速缓存服务器可以包括垃圾收集过程,其周期性地从高速缓存中清除不再使用的旧增量。例如,可以通过大小阈值(例如,当高速缓存增长超过某个大小时)触发该垃圾收集,并且可以清除最近最少使用的增量。为此,高速缓存可以为每个增量包含从高速缓存中提供该增量的最后时间。垃圾收集过程可以被配置为基于在阈值时间内尚未被提供来删除增量。例如,垃圾收集过程可被配置为从不删除最近使用超过1小时(或配置的LFT超时值)的项目。

[0208] 来自客户端106的对对象的读取请求可以返回对象版本之间的差异。版本之一可以是客户端106已经具有的版本,而另一个可以是核心服务器上的最新版本。在根据一个或更多个实施例的示例场景中,客户端106最初可能没有对象,这可能被认为是版本0。客户端106的内容管理器410可以向核心服务器发送读取请求,其在客户端106处将版本0指定为最新版本。如果核心服务器上的当前版本是版本1,则核心服务器可能会生成版本0和1之间的差异数据,这些差异数据可以写入具有某个唯一文件ID和大小为size1的文件。核心服务器可以生成一些带有File_Id(文件_Id)和范围[0,size1)的Content_Id(内容_Id),然后将其返回给客户端106。客户端106可以接收这个Content_Id并发起从高速缓存服务器的下载。假设高速缓存服务器还没有具有File_Id的高速缓存的文件,则高速缓存服务器可以从[0,size1)范围内的File_Id的核心服务器发起下载。下载完成后,可能会有带有File_Id、大小为size1的高速缓存文件。客户端106可以读取高速缓存文件并应用差异数据以将对象更新到版本1。

[0209] 为了继续以上示例,现在假设核心服务器具有新的版本2,并且某个其他客户端106以版本0发起读取。核心服务器已经在来自先前请求的具有File_Id的文件中具有版本0- $\rightarrow$ 1的差异数据。因此,核心服务器可只生成版本1- $\rightarrow$ 2的差异数据,并将这些差异数据附加到带有File_Id的文件末尾,导致为size2的文件大小。可以生成具有File_Id和范围[0,

size2)的Content_Id2,然后将其返回给客户端106。假设[0,size1)范围已经在用于File_Id的高速缓存中,则可能只需要从核心服务器下载[size1,size2)。

[0210] 如果核心服务器处的当前版本是版本3,并且具有版本1的第一客户端106发起另一个读取请求,由于核心服务器在具有File_Id的文件中已经具有版本0->2的差异数据,它可以仅生成版本2->3的差异数据,并将此差异数据附加到具有File_Id的文件末尾,结果文件大小为size3。可以生成具有File_Id和范围[size1,size3)的新Content_Id3,并且可以将其返回给客户端106。由于[0,size2)范围已经在用于File_Id的高速缓存中,因此可能只需要从核心服务器下载[size2,size3)。

[0211] 如果服务器/高速缓存文件仅具有版本0->10、10->20和20->30的差异数据,并且客户端106发起读取,该读取指示其具有版本15,则至少可以重用版本20->30的差异数据,版本15->20可以作为新文件生成,或者版本15->30可以作为新文件生成。例如,这可能发生在连接丢失或客户端106从离线切换到在线时。虽然上述描述了一个大文件和返回范围,但在其他示例中,可以使用单独的文件进行存储并且可以返回这些文件而不是范围(使用对应的File_Id)。

[0212] 对象的文件可能总是在增长,因此在某些时候可能需要重置文件并重新开始,同时丢弃引用该文件的所有内容ID。可能并不总是可以重置和重用相同的文件,例如在存在文件的活动中下载中。因此,可以启动新文件,并且一旦完成旧文件的所有下载,就可以删除旧文件。

[0213] 如果在一个文件中存在具有相同大值变化的多个差异,则可以通过在所有差异中仅搜索最新变化来优化读取和应用该相同大值。例如,假设diff版本0->10的value1有很大的变化,diff版本10->20的value1和value2有很大的变化,而diff版本20->30的value2有很大的变化。当在客户端106上处理版本0->30时,只能采用版本10->20中的value1和版本20->30中的value2,忽略版本0->10中的value1和版本10->20中的value2。

[0214] 示例数据库格式

[0215] 在各种实施例中,可以使用多个表来存储对象和对象版本。一个OBJECT_ID(对象_ID)表可以使用对象ID作为键,值可以是最新创建的对象ID。PATH_TO_OBJECT_ID(路径到_对象_ID)表可以捕获对象名称(例如,由内容管理器410用来引用对象)和对象ID之间的映射。

[0216] OBJECT_REFCOUNT表可以使用对象ID作为键。该值可以表示有多少对象或路径引用该对象。在一些实施例中,如果数据存储管理器108确定存在对来自表的对象的引用,则数据存储管理器108将不删除该对象。例如,在一个示例场景中,其中有一树的对象都相互引用,对象A分支到对象B以及对象C和D,不应删除对象A,因为它被其他对象引用。然而,一旦子对象被删除并且没有引用对象剩余,父对象也将被删除。

[0217] OBJECT_HEADER(对象_报头)表可以使用对象ID作为键。该值可以表示具有版本信息和对象的父对象的打包结构。这可以用于如果对象被删除并且对象用相同名称被重新创建的场景,则客户端106可以确定它是不同的对象。

[0218] OBJECT_NODE(对象_节点)表可以使用对象ID\节点ID作为键。该值可以表示具有子列表中子节点的节点ID的结构。

[0219] OBJECT_FIELD_VERSION(对象_字段_版本)表可以用作键对象ID/节点ID/字段名

称。该值可以是具有节点信息的结构,例如图11中描述的。

[0220] OBJECT_FIELD_DATA (对象_字段_数据) 表可以用作键对象ID/节点ID/字段名称。该值可以表示字段的字段值。

[0221] OBJECT_TIME_SAMPLE_VERSION (对象_时间_样本_版本) 表可以用作键对象ID/节点ID/时间。该值可以表示节点中每个时间样本的版本以及该节点中每个时间样本的名称。

[0222] OBJECT_TIME_SAMPLE_DATA (对象_时间_样本_数据) 表可以用作键对象ID/节点ID/时间。该值可以表示字段的字段值。该值可以表示时间样本的时间时间值。

[0223] 附加示例

[0224] 在至少一个实施例中,系统包括处理单元和存储器,该存储器耦合到处理单元并在其中存储了数据存储,用于存储表示三维(3D)环境的对象的数据,其中这些对象中的对象包括跨3D环境的场景描述的内容项定义的属性和值的集合。该系统还包括通信管理器,该通信管理器耦合到存储器并且可操作用于与客户端建立双向通信信道以访问3D环境的一个或更多个内容项。表示由客户端中的第一客户端通过第一双向通信信道贡献的内容项中的内容项的对象的属性和值的集合的一个或更多个改变的增量信息被保存到数据存储并且基于第二客户端对内容项的订阅,通过第二双向通信信道提供给客户端中的至少第二客户端。内容项可以是场景描述的各层中的层,并且对象的属性和值的集合可以通过层的排名来解析。

[0225] 示例计算设备

[0226] 图13为适合用于实现本公开一些实施例的示例计算设备1300的框图。计算设备1300可以包括直接或间接耦合下列设备的互连系统1302:内存1304,一个或更多个中央处理单元(CPU) 1306,一个或更多个图形处理单元(GPU) 1308,通信接口1310,输入/输出(I/O) 端口1312,输入/输出组件1314,电源1316,一个或更多个呈现组件1318(例如显示器)和一个或更多个逻辑单元1320。

[0227] 在至少一个实施例中,计算设备1300可以包括一个或更多个虚拟机(VM),和/或其任何组件可以包括虚拟组件(例如,虚拟硬件组件)。例如,一个或更多个GPU 1308可以包括一个或更多个vGPU,一个或更多个CPU 1306可以包括一个或更多个vCPU,和/或一个或更多个逻辑单元1320可以包括一个或更多个虚拟逻辑单元。

[0228] 尽管图13的各个框被示为经由具有线路的互连系统1302连接,但是这并不意图是限制性的,并且仅仅为了清楚起见。例如,在一些实施例中,诸如显示设备之类的呈现组件1318可以被认为I/O组件1314(例如,如果显示器为触摸屏)。作为另一个示例,CPU 1306和/或GPU 1308可以包括内存(例如,内存1304可以表示除了GPU 1308、CPU 1306和/或其他组件的内存以外的存储设备)。换言之,图13的计算设备仅仅是说明性的。在诸如“工作站”、“服务器”、“膝上型电脑”、“台式机”、“平板电脑”、“客户端设备”、“移动设备”、“手持式设备”、“游戏控制台”、“电子控制单元(ECU)”、“虚拟现实系统”和/或其他设备或系统类型之类的类别之间不进行区分,因为所有这些都被考虑在图13的计算设备的范围内。

[0229] 互连系统1302可以表示一条或更多条链路或总线,例如地址总线、数据总线、控制总线或者其组合。互连系统1302可以包括一种或更多种链路或总线类型,例如行业标准架构(ISA)总线、扩展行业标准架构(EISA)总线、视频电子标准协会(VESA)总线、外围组件互连(PCI)总线、外围组件互连快速(PCIe)总线,和/或另一种类型的总线或链路。在一些实施

例中,组件之间存在直接连接。作为示例,CPU 1306可以直接连接到存储器1304。此外,CPU 1306可以直接连接到GPU 1308。在组件之间存在直接或点对点连接的情况下,互连系统1302可以包括PCIe链路来执行该连接。在这些示例中,计算设备1300中不需要包括PCI总线。

[0230] 内存1304可以包括各种各样的计算机可读介质中的任何计算机可读介质。计算机可读介质可以是可由计算设备1300访问的任何可用介质。计算机可读介质可以包括易失性和非易失性介质以及可移除和不可移除介质。举例而言且非限制性地,计算机可读介质可以包括计算机存储介质和通信介质。

[0231] 计算机存储介质可以包括易失性和非易失性介质和/或可移除和不可移除介质,其以用于存储诸如计算机可读指令、数据结构、程序模块和/或其他数据类型之类的信息的任何方法或技术实现。例如,内存1304可以存储计算机可读指令(例如其表示程序和/或程序元素,例如操作系统)。计算机存储介质可以包括但不限于RAM、ROM、EEPROM、闪存或者其他存储技术、CD-ROM、数字多功能盘(DVD)或其他光盘存储装置、磁带盒、磁带、磁盘存储装置或其他磁存储设备,或者可以用来存储期望的信息且可以由计算设备1300访问的任何其他介质。当在本文使用时,计算机存储介质并不包括信号本身。

[0232] 计算机存储介质可以在诸如载波之类的调制数据信号或其他传输机制中包含计算机可读指令、数据结构、程序模块和/或其他数据类型,并且包括任何信息输送介质。术语“调制数据信号”可以指这样的信号,该信号使它的特性中的一个或更多个以这样的将信息编码到该信号中的方式设置或改变。举例而言且非限制性地,计算机存储介质可以包括诸如有线网络或直接有线连接之类的有线介质,以及诸如声音、RF、红外和其他无线介质之类的无线介质。任何以上所述的组合也应当包含在计算机可读介质的范围内。

[0233] CPU 1306可以被配置为执行计算机可读指令中的至少一些,以便控制计算设备1300的一个或更多个组件执行本文描述的方法和/或过程中的一个或更多个。CPU 1306中的每一个可以包括能够同时处理大量软件线程的一个或更多个核(例如一个、两个、四个、八个、二十八个、七十二个等等)。CPU 1306可以包括任何类型的处理器,并且可以包括不同类型的处理器,这取决于实现的计算设备1300的类型(例如具有用于移动设备的较少核的处理器以及具有用于服务器的更多核的处理器)。例如,取决于计算设备1300的类型,处理器可以是使用精简指令集计算(RISC)实现的高级RISC机制(ARM)处理器或者使用复杂指令集计算(CISC)实现的x86处理器。除了一个或更多个微处理器或者诸如数学协处理器之类的补充协处理器之外,计算设备1300还可以包括一个或更多个CPU 1306。

[0234] 除了或替代CPU 1306,GPU 1308还可以被配置为执行至少一些计算机可读指令,以控制计算设备1300的一个或更多个组件执行一个或更多个本文所述的方法和/或过程。一个或更多个GPU 1308可以是集成GPU(例如,具有一个或更多个CPU 1306)和/或一个或更多个GPU 1308可以是离散GPU。在实施例中,一个或更多个GPU 1308可以是一个或更多个CPU 1306的协处理器。计算设备1300可以使用GPU 1308来渲染图形(例如,3D图形)或执行通用计算。例如,GPU 1308可用于GPU上的通用计算(GPGPU)。GPU 1308可以包括能够同时处理数百或数千个软件线程的数百或数千个核。GPU 1308可以响应于渲染命令(例如经由主机接口接收的来自CPU 1306的渲染命令)而生成用于输出图像的像素数据。GPU 1308可以包括诸如显示内存之类的用于存储像素数据或任何其他合适的数据(例如GPGPU数据)的图

形内存。显示内存可以作为内存1304的部分而被包括。GPU 1308可以包括(例如经由链路)并行操作的两个或更多GPU。链路可以直接连接GPU(例如,使用NVLINK),也可以通过交换机(例如,使用NVSwitch)连接GPU。当组合在一起时,每个GPU 1308可以生成用于输出的不同部分或者用于不同输出的像素数据或GPGPU数据(例如,第一GPU用于第一图像,第二GPU用于第二图像)。每个GPU可以包括它自己的内存,或者可以与其他GPU共享内存。

[0235] 除了或替代CPU 1306和/或GPU 1308,逻辑单元1320可以被配置为执行至少一些计算机可读指令,以控制计算设备1300的一个或更多个组件来执行本文描述的一种或更多种方法和/或过程。在实施例中,CPU 1306、GPU 1308和/或逻辑单元1320可以分立地或联合地执行方法、过程和/或其部分的任何组合。一个或更多个逻辑单元1320可以是一个或更多个CPU 1306和/或一个或更多个GPU 1308的一部分和/或集成在其中,和/或一个或更多个逻辑单元1320可以是CPU 1306和/或GPU 1308的分立组件或以其他方式在其外部。在实施例中,一个或更多个逻辑单元1320可以是一个或更多个CPU 1306和/或一个或更多个GPU 1308的处理器。

[0236] 逻辑单元1320的示例包括一个或更多个处理核心和/或其组件,例如张量核心(TC)、张量处理单元(TPU)、像素视觉核心(PVC)、视觉处理单元(VPU)、图形处理集群(GPC)、纹理处理集群(TPC)、流式多处理器(SM)、树遍历单元(TTU)、人工智能加速器(AIA)、深度学习加速器(DLA)、算术逻辑单元(ALU)、专用集成电路(ASIC)、浮点单元(FPU)、输入/输出(I/O)元件、外围组件互连(PCI)或外围组件互连快速(PCIe)元件等。

[0237] 通信接口1310可以包括一个或更多个接收器、发送器和/或收发器,其使得计算设备1300能够经由电子通信网络与其他计算设备通信,包括有线和/或无线通信。通信接口1310可以包括使能通过若干不同网络中的任何网络进行通信的组件和功能,所述网络例如无线网络(例如Wi-Fi、Z波、蓝牙、蓝牙LE、ZigBee等等)、有线网络(例如通过以太网或InfiniBand通信)、低功率广域网(例如LoRaWAN、SigFox等等)和/或因特网。

[0238] I/O端口1312可以使得计算设备1300能够逻辑地耦合到包括I/O组件1314、呈现组件1318和/或其他组件在内的其他设备,其中一些可以内置到(例如集成到)计算设备1300中。说明性I/O组件1314包括麦克风、鼠标、键盘、操纵杆、游戏垫、游戏控制器、碟形卫星天线、浏览仪、打印机、无线设备等等。I/O组件1314可以提供处理用户生成的空中手势、语音或其他生理输入的自然用户接口(NUI)。在一些实例中,输入可以传输至适当的网络元件以便进一步处理。NUI可以实现语音识别、手写笔识别、面部识别、生物特征识别、屏幕上和邻近屏幕的手势识别、空中手势、头部和眼睛跟踪以及与计算设备1300的显示器关联的触摸识别(如下文更详细地描述的)的任意组合。计算设备1300可以包括诸如立体相机系统之类的深度相机、红外相机系统、RGB相机系统、触摸屏技术以及这些的组合,以用于手势检测和识别。此外,计算设备1300可以包括使能运动检测的加速度计或陀螺仪(例如作为惯性测量单元(IMU)的部分)。在一些示例中,加速度计或陀螺仪的输出可以由计算设备1300用来渲染沉浸式增强现实或者虚拟现实。

[0239] 电源1316可以包括硬接线电源、电池电源或者其组合。电源1316可以向计算设备1300供电以使得计算设备1300的组件能够操作。

[0240] 呈现组件1318可以包括显示器(例如监视器、触摸屏、电视屏幕、平视显示器(HUD)、其他显示器类型或者其组合)、扬声器和/或其他呈现组件。呈现组件1318可以接收

来自其他组件 (例如GPU 1308、CPU 1306等等) 的数据,并且输出该数据 (例如作为图像、视频、声音等等)。

[0241] 示例网络环境

[0242] 适用于实现本公开的实施例的网络环境可以包括一个或更多个客户端设备、服务器、网络附加存储 (NAS)、其他后端设备和/或其他设备类型。客户端设备、服务器和/或其他设备类型 (例如,每个设备) 可以在图13的计算设备1300的一个或更多个实例上实现一例如,每个设备可以包括计算设备1300的类似组件、特征和/或功能。

[0243] 网络环境的组件可以通过网络彼此通信,网络可以是有线、无线或两者。网络可以包括多个网络,或者多个网络中的网络。举例来说,网络可以包括一个或更多个广域网 (WAN)、一个或更多个局域网 (LAN)、一个或更多个公共网络 (例如因特网和/或公共交换电话网 (PSTN))、和/或一个或更多个专用网络。在网络包括无线电信网的情况下,诸如基站、通信塔或甚至接入点 (以及其他组件) 之类的组件可以提供无线连接。

[0244] 兼容的网络环境可以包括一个或更多个对等网络环境 (在这种情况下服务器可能不包括在网络环境中),以及一个或更多个客户端-服务器网络环境 (在这种情况下一个或更多个服务器可以包含在网络环境中)。在对等网络环境中,本文描述的关于服务器的功能可以在任意数量的客户端设备上实现。

[0245] 在至少一个实施例中,网络环境可以包括一个或更多个基于云的网络环境、分布式计算环境、它们的组合等。基于云的网络环境可以包括框架层、作业调度器、资源管理器,以及在一个或更多个服务器上实现的分布式文件系统,这些服务器可以包括一个或更多个核心网络服务器和/或边缘服务器。框架层可以包括用于支持软件层的软件和/或应用层的一个或更多个应用程序的框架。软件或应用程序可分别包括基于网络的服务软件或应用程序。在实施例中,一个或更多个客户端设备可以使用基于网络的服务软件或应用程序 (例如,通过经由一个或更多个应用程序编程接口 (API) 访问服务软件和/或应用程序)。框架层可以是但不限于一种类型的免费且开源软件网络应用程序框架,例如可以使用分布式文件系统进行大规模数据处理 (例如,“大数据”)。

[0246] 基于云的网络环境可以提供执行本文描述的计算和/或数据存储功能 (或其一个或更多个部分) 的任何组合的云计算和/或云存储。这些各种功能中的任何一个都可以分布在来自 (例如,可以分布在州、地区、国家、全球等地的一个或更多个数据中心的) 中央或核心服务器的多个位置。如果到用户 (例如,客户端设备) 的连接相对靠近边缘服务器,则核心服务器可以将功能的至少一部分指定给边缘服务器。基于云的网络环境可以是私有的 (例如,仅限于单个组织),可以是公共的 (例如,对许多组织可用),和/或其组合 (例如,混合云环境)。

[0247] 客户端设备可以包括本文关于图13描述的示例计算设备1300的至少一些组件、特征和功能。作为示例而非限制,客户端设备可以体现为个人计算机 (PC)、膝上型计算机、移动设备、智能手机、平板计算机、智能手表、可穿戴计算机、个人数码助理 (PDA)、MP3播放器、虚拟现实头戴式显示器、全球定位系统 (GPS) 或设备、视频播放器、摄像机、监视设备或系统、车辆、船只、飞行器、虚拟机器、无人机、机器人、手持通信设备、医院设备、游戏设备或系统、娱乐系统、车载计算机系统、嵌入式系统控制器、遥控器、电器、消费电子设备、工作站、边缘设备、这些描述设备的任意组合或任何其他合适的设备。

[0248] 示例数据中心

[0249] 图14示出了示例数据中心1400,其中可以使用至少一个实施例。在至少一个实施例中,数据中心1400包括数据中心基础设施层1410、框架层1420、软件层1430和应用层1440。

[0250] 在至少一个实施例中,如图14所示,数据中心基础设施层1410可以包括资源协调器1412、分组的计算资源1414和节点计算资源(“节点C.R.”)1416(1)-1416(N),其中“N”代表任何完整的正整数。在至少一个实施例中,节点C.R.1416(1)-1416(N)可以包括但不限于任何数量的中央处理单元(“CPU”)或其他处理器(包括加速器、现场可编程门阵列(FPGA)、图形处理器等),内存设备(例如动态只读内存),存储设备(例如固态硬盘或磁盘驱动器),网络输入/输出(“NW I/O”)设备,网络交换机,虚拟机(“VM”),电源模块和冷却模块等。在至少一个实施例中,节点C.R.1416(1)-1416(N)中的一个或多个节点C.R.可以是具有一个或多个上述计算资源的服务器。

[0251] 在至少一个实施例中,分组的计算资源1414可以包括容纳在一个或多个机架内的节点C.R.的单独分组(未示出),或者容纳在各个地理位置的数据中心内的许多机架(也未示出)。分组的计算资源1414内的节点C.R.的单独分组可以包括可以被配置或分配为支持一个或多个工作负载的分组的计算、网络、内存或存储资源。在至少一个实施例中,可以将包括CPU或处理器的几个节点C.R.分组在一个或多个机架内,以提供计算资源来支持一个或多个工作负载。在至少一个实施例中,一个或多个机架还可以包括任何数量的电源模块、冷却模块和网络交换机,以任意组合。

[0252] 在至少一个实施例中,资源协调器1422可以配置或以其他方式控制一个或多个节点C.R.1416(1)-1416(N)和/或分组的计算资源1414。在至少一个实施例中,资源协调器1422可以包括用于数据中心1400的软件设计基础结构(“SDI”)管理实体。在至少一个实施例中,资源协调器可以包括硬件、软件或其某种组合。

[0253] 在至少一个实施例中,如图14所示,框架层1420包括作业调度器1432、配置管理器1434、资源管理器1436和分布式文件系统1438。在至少一个实施例中,框架层1420可以包括支持软件层1430的软件1444和/或应用程序层1440的一个或多个应用程序1442的框架。在至少一个实施例中,软件1444或应用程序1442可以分别包括基于Web的服务软件或应用程序,例如由Amazon Web Services,Google Cloud和Microsoft Azure提供的服务或应用程序。在至少一个实施例中,框架层1420可以是但不限于一种免费和开放源软件网络应用框架,例如可以利用分布式文件系统1438来进行大范围数据处理(例如“大数据”)的Apache Spark[™](以下称为“Spark”)。在至少一个实施例中,作业调度器1432可以包括Spark驱动器,以促进对数据中心1400的各个层所支持的工作负载进行调度。在至少一个实施例中,配置管理器1434可以能够配置不同的层,例如软件层1430和包括Spark和用于支持大规模数据处理的分布式文件系统1438的框架层1420。在至少一个实施例中,资源管理器1436能够管理映射到或分配用于支持分布式文件系统1438和作业调度器1432的集群或分组计算资源。在至少一个实施例中,集群或分组计算资源可以包括数据中心基础设施层1410上的分组的计算资源1414。在至少一个实施例中,资源管理器1436可以与资源协调器1412协调以管理这些映射的或分配的计算资源。

[0254] 在至少一个实施例中,包括在软件层1430中的软件1444可以包括由节点C.R.1416

(1) - 1416(N) 的至少一部分, 分组计算资源1414和/或框架层1420的分布式文件系统1438使用的软件。一种或更多种类型的软件可以包括但不限于Internet网页搜索软件、电子邮件病毒扫描软件、数据库软件和流视频内容软件。

[0255] 在至少一个实施例中, 应用层1440中包括的一个或更多个应用程序1442可以包括由节点C.R. 1416(1) - 1416(N) 的至少一部分、分组的计算资源1414和/或框架层1420的分布式文件系统1438使用的一种或更多种类型的应用程序。一种或更多种类型的应用程序可以包括但不限于任何数量的基因组学应用程序, 认知计算和机器学习应用程序, 包括训练或推理软件, 机器学习框架软件 (例如PyTorch、TensorFlow、Caffe等) 或其他与一个或更多个实施例结合使用的机器学习应用程序。

[0256] 在至少一个实施例中, 配置管理器1434、资源管理器1436和资源协调器1412中的任何一个可以基于以任何技术上可行的方式获取的任何数量和类型的数据来实现任何数量和类型的自我修改动作。在至少一个实施例中, 自我修改动作可以减轻数据中心1400的数据中心操作员做出可能不好的配置决定并且可以避免数据中心的未充分利用和/或执行差的部分。

[0257] 在至少一个实施例中, 数据中心1400可以包括工具、服务、软件或其他资源, 以根据本文所述的一个或更多个实施例来训练一个或更多个机器学习模型或者使用一个或更多个机器学习模型来预测或推理信息。例如, 在至少一个实施例中, 可以通过使用上文关于数据中心1400描述的软件和计算资源, 根据神经网络架构通过计算权重参数来训练机器学习模型。在至少一个实施例中, 通过使用通过本文所述的一种或更多种训练技术计算出的权重参数, 可以使用上面与关于数据中心1400所描述的资源, 使用对应于一个或更多个神经网络的经训练的机器学习模型来推理或预测信息。

[0258] 本公开可以在由计算机或者诸如个人数字助理或其他手持式设备之类的其他机器执行的、包括诸如程序模块之类的计算机可执行指令的机器可使用指令或者计算机代码的一般背景下进行描述。通常, 包括例程、程序、对象、组件、数据结构等等的程序模块指的是执行特定任务或者实现特定抽象数据类型的代码。本公开可以在各种各样的系统配置中实践, 这些配置包括手持式设备、消费电子器件、通用计算机、更专业的计算设备等等。本公开也可以在其中任务由通过通信网络链接的远程处理设备执行的分布式计算环境中实践。

[0259] 如在本文中使用的, “和/或” 关于两个或更多元素的叙述应当解释为仅指一个元素或者元素组合。例如, “元素A、元素B和/或元素C” 可以包括仅仅元素A, 仅仅元素B, 仅仅元素C, 元素A和元素B, 元素A和元素C, 元素B和元素C, 或者元素A、B和C。此外, “元素A或元素B中的至少一个” 可以包括元素A中的至少一个, 元素B中的至少一个, 或者元素A中的至少一个和元素B中的至少一个。进一步, “元素A和元素B中的至少一个” 可以包括元素A中的至少一个, 元素B中的至少一个, 或者元素A中的至少一个和元素B中的至少一个。

[0260] 这里详细地描述了本公开的主题以满足法定要求。然而, 描述本身并非意在限制本公开的范围。相反地, 本发明人已经设想到, 要求保护的主题也可以以其他的方式具体化, 以包括与本文中结合其他当前或未来技术描述的步骤不同的步骤或者相似的步骤的组合。而且, 尽管术语“步骤” 和/或“块” 在本文中可以用来隐含采用的方法的不同元素, 但是这些术语不应当被解释为暗示本文公开的各个步骤之中或之间的任何特定顺序, 除非明确描述了各步骤的顺序。

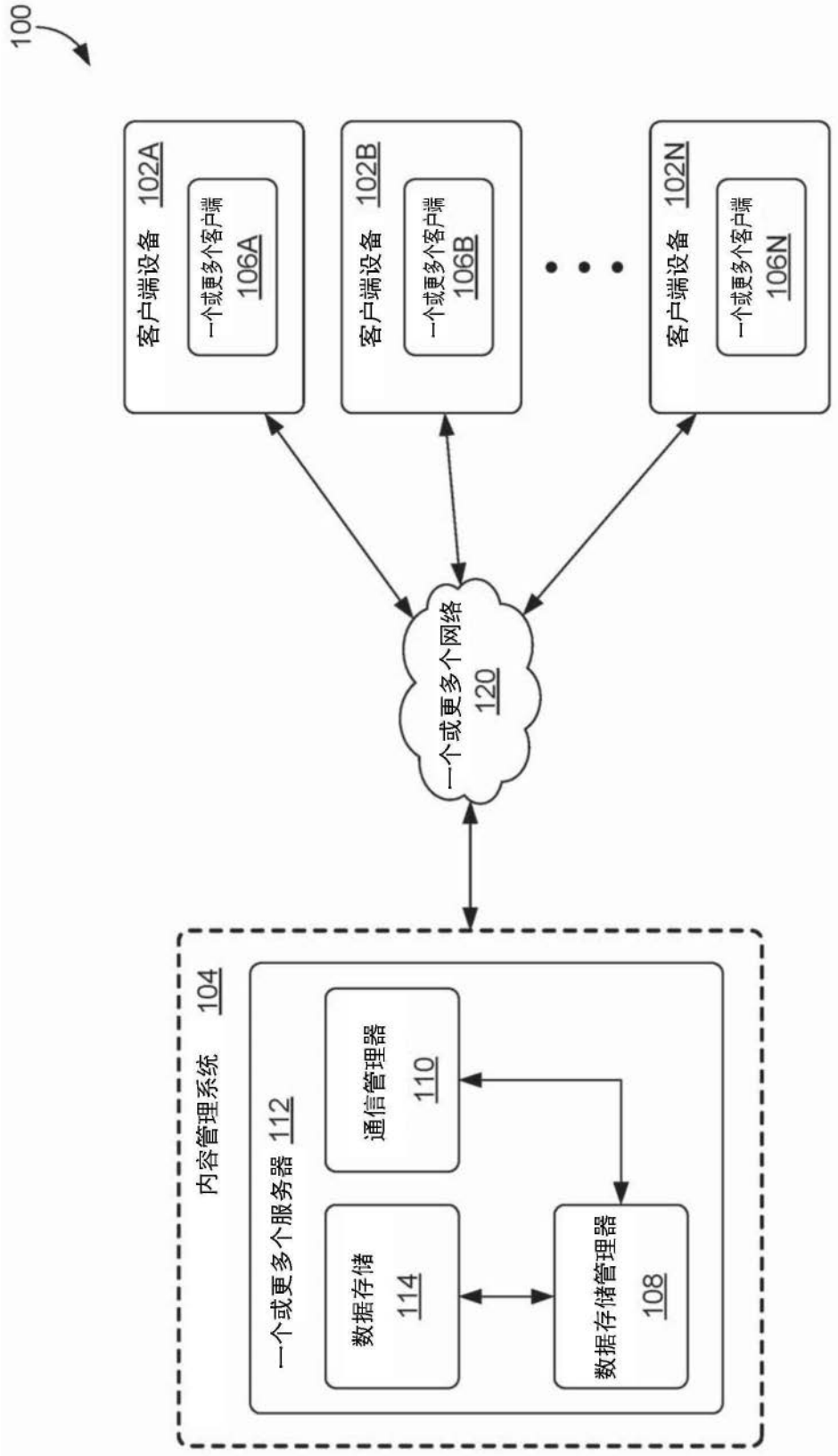


图1

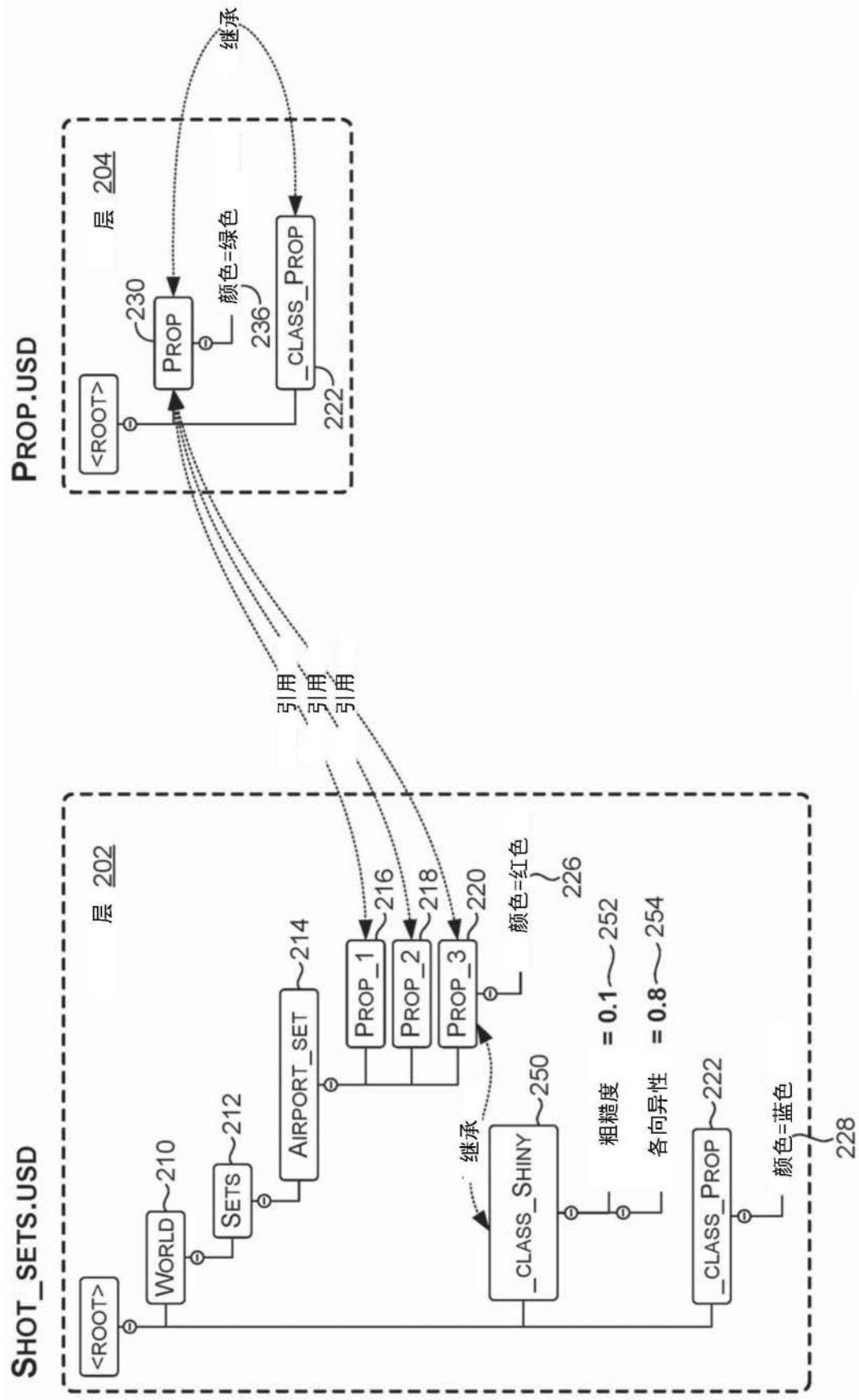


图2A

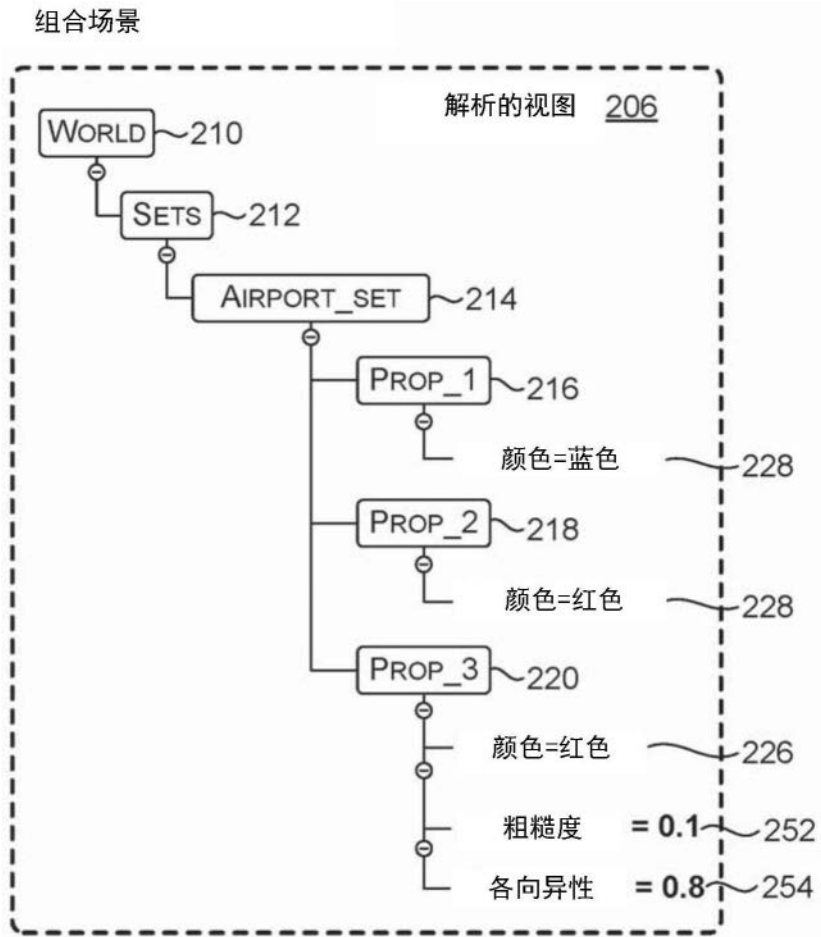


图2B

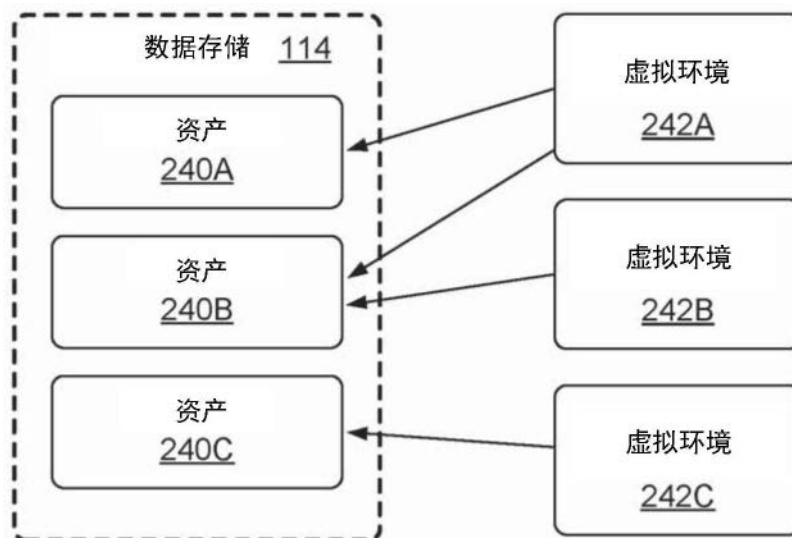


图2C

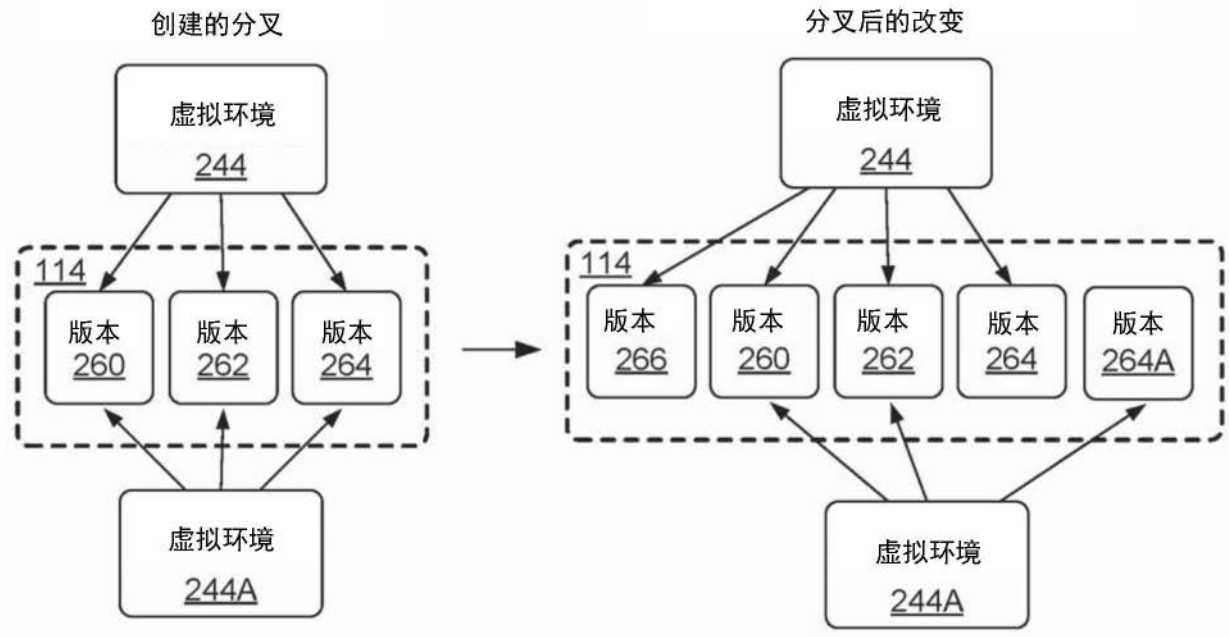


图2D

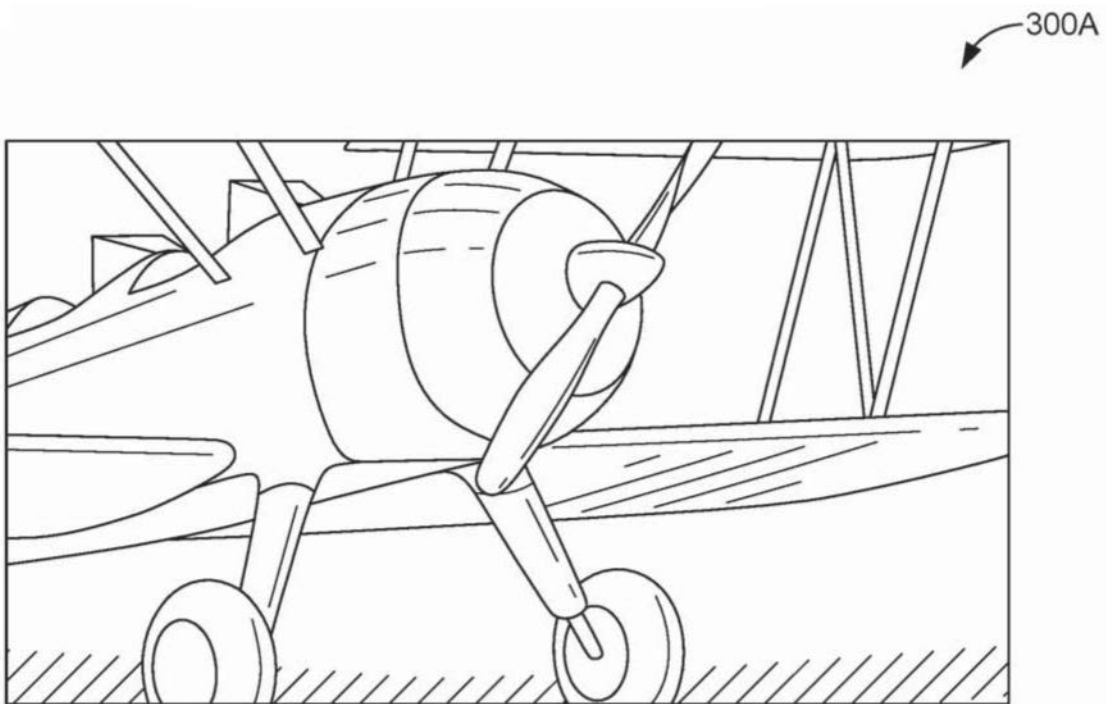


图3A

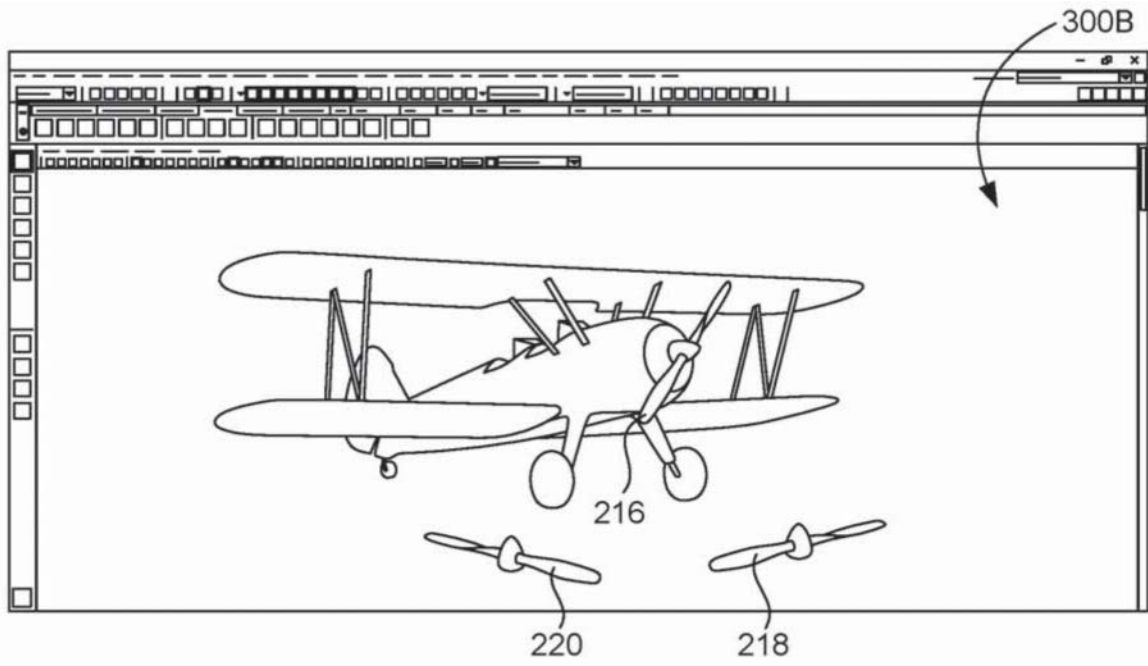


图3B

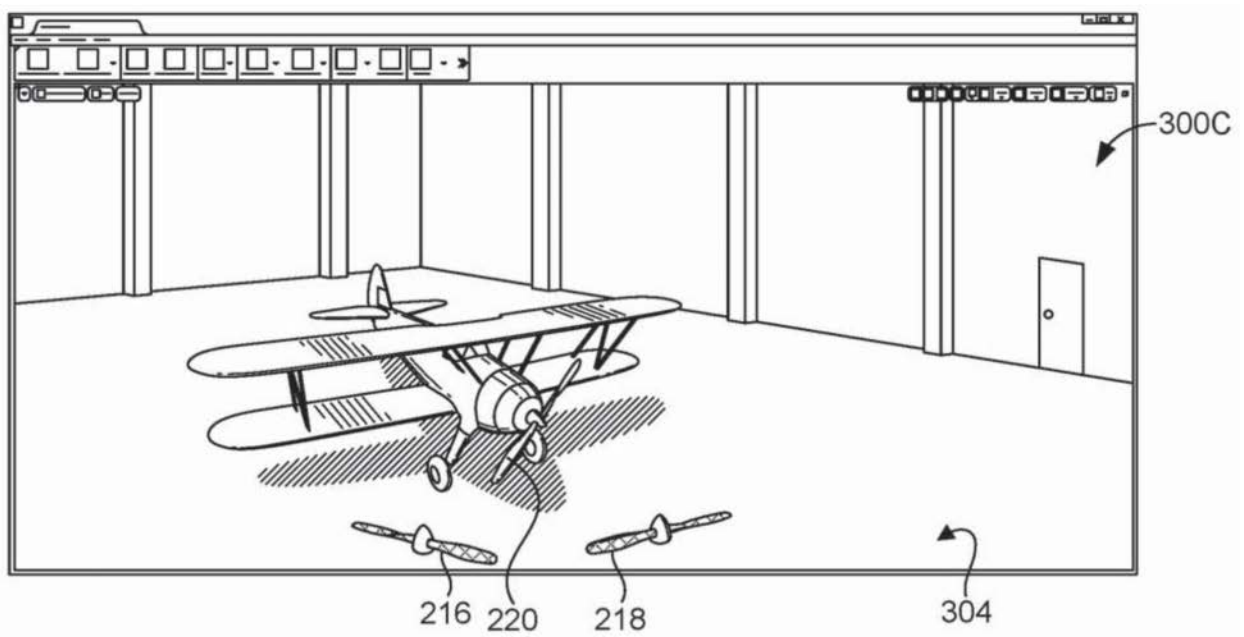


图3C

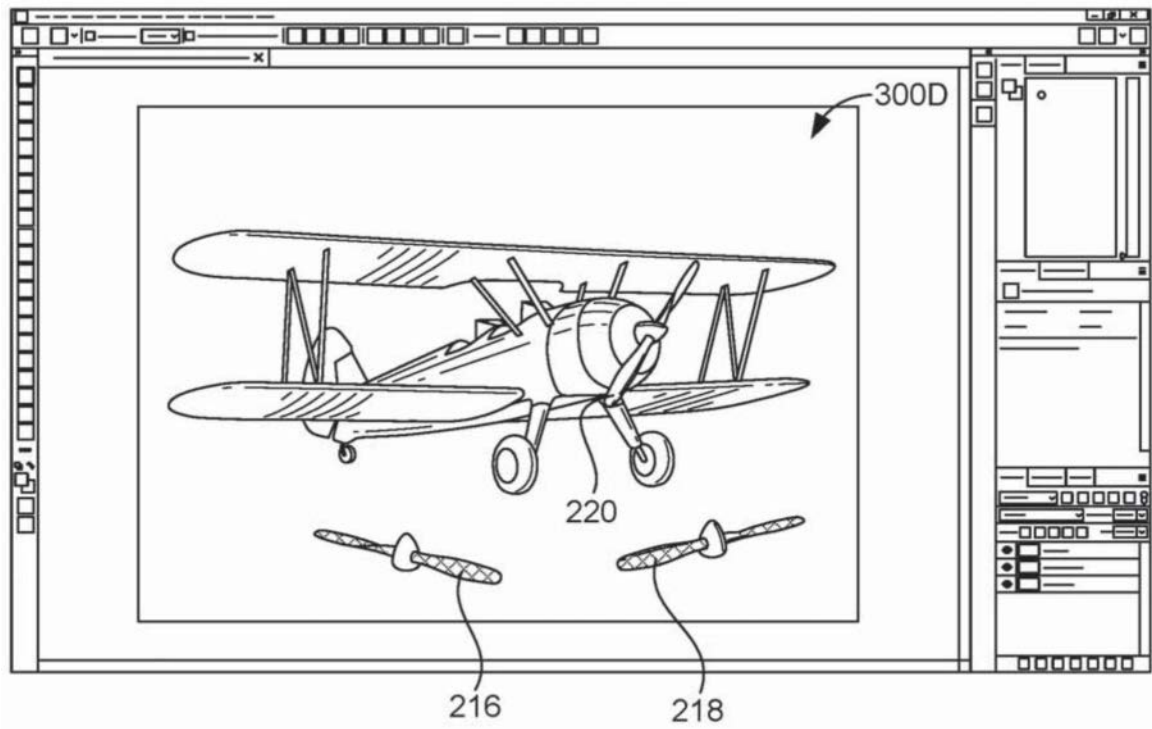


图3D

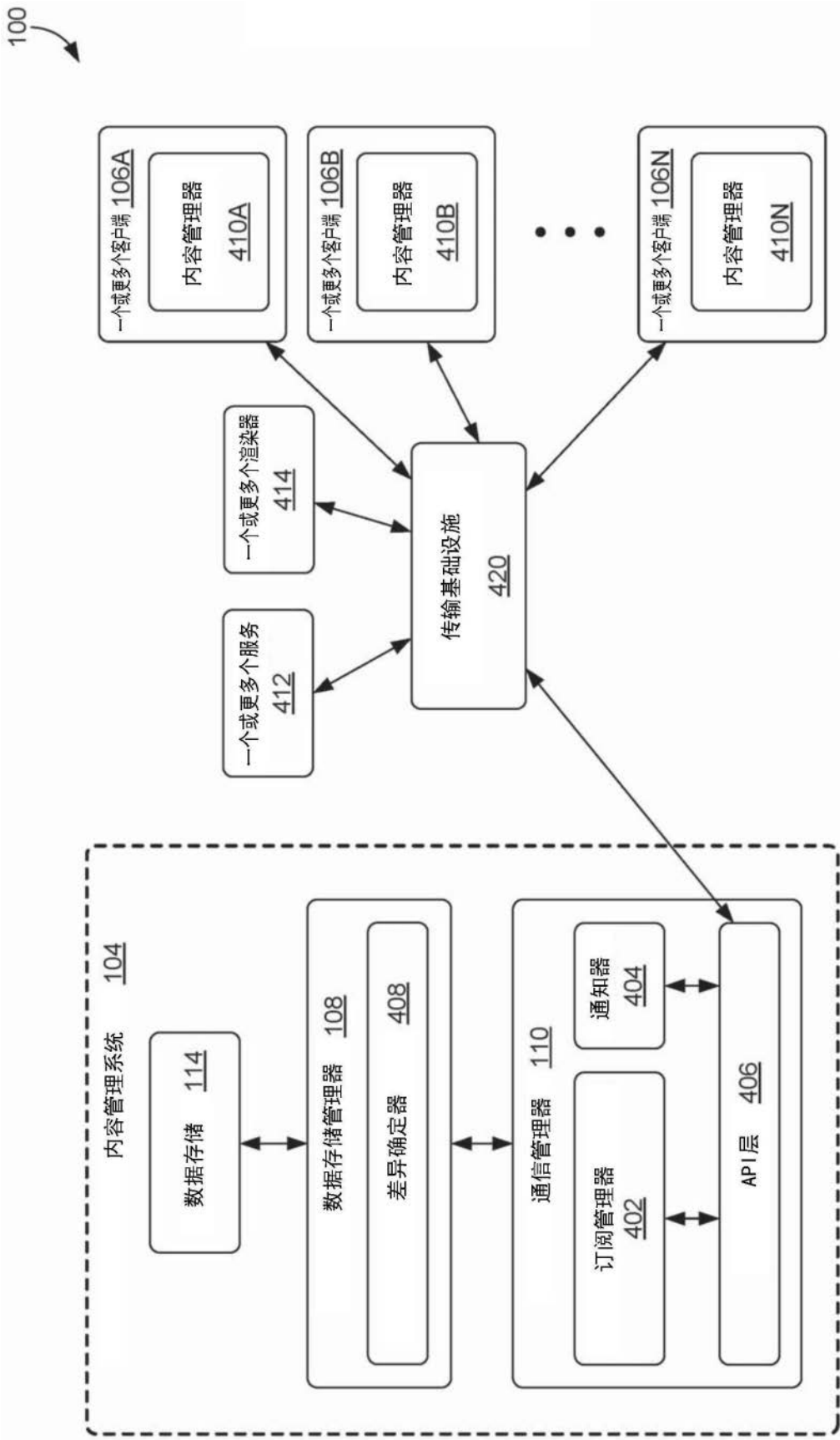


图4A

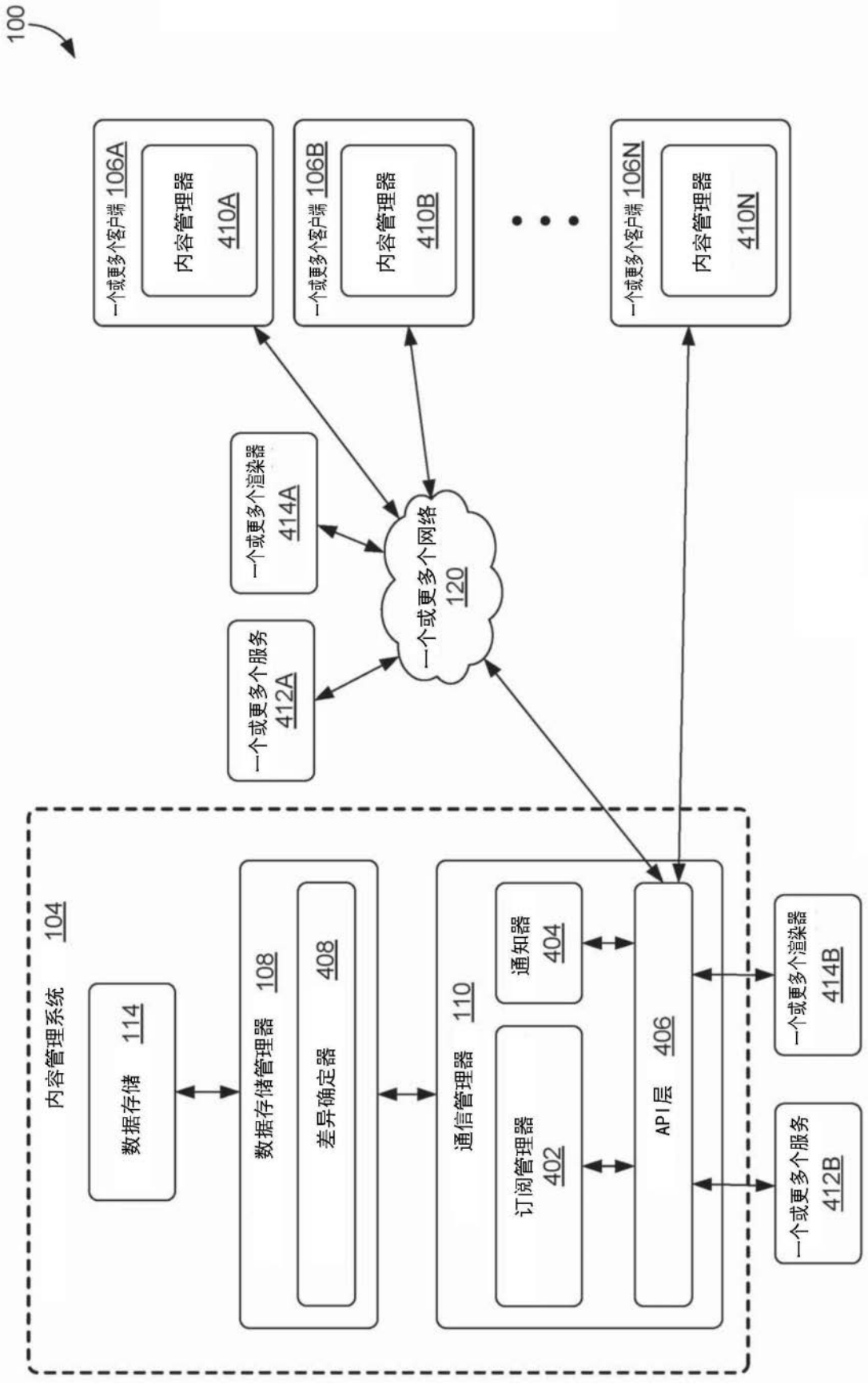


图4B

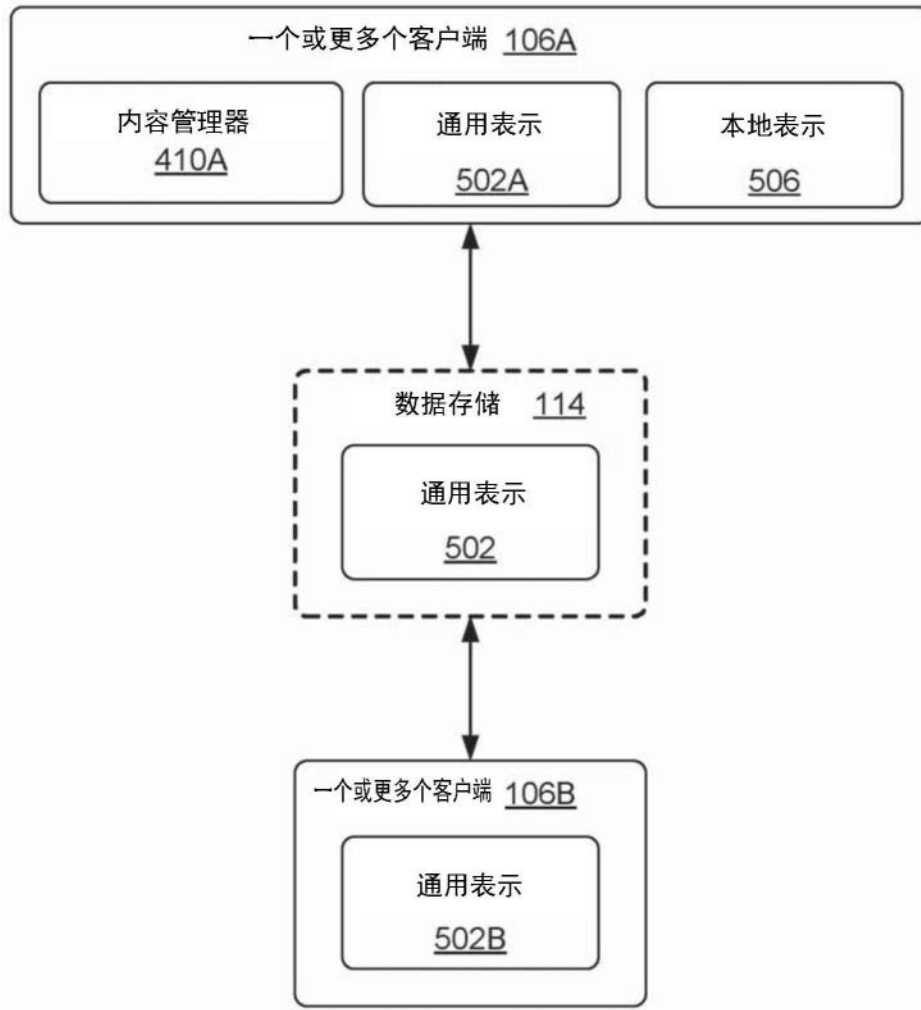


图5

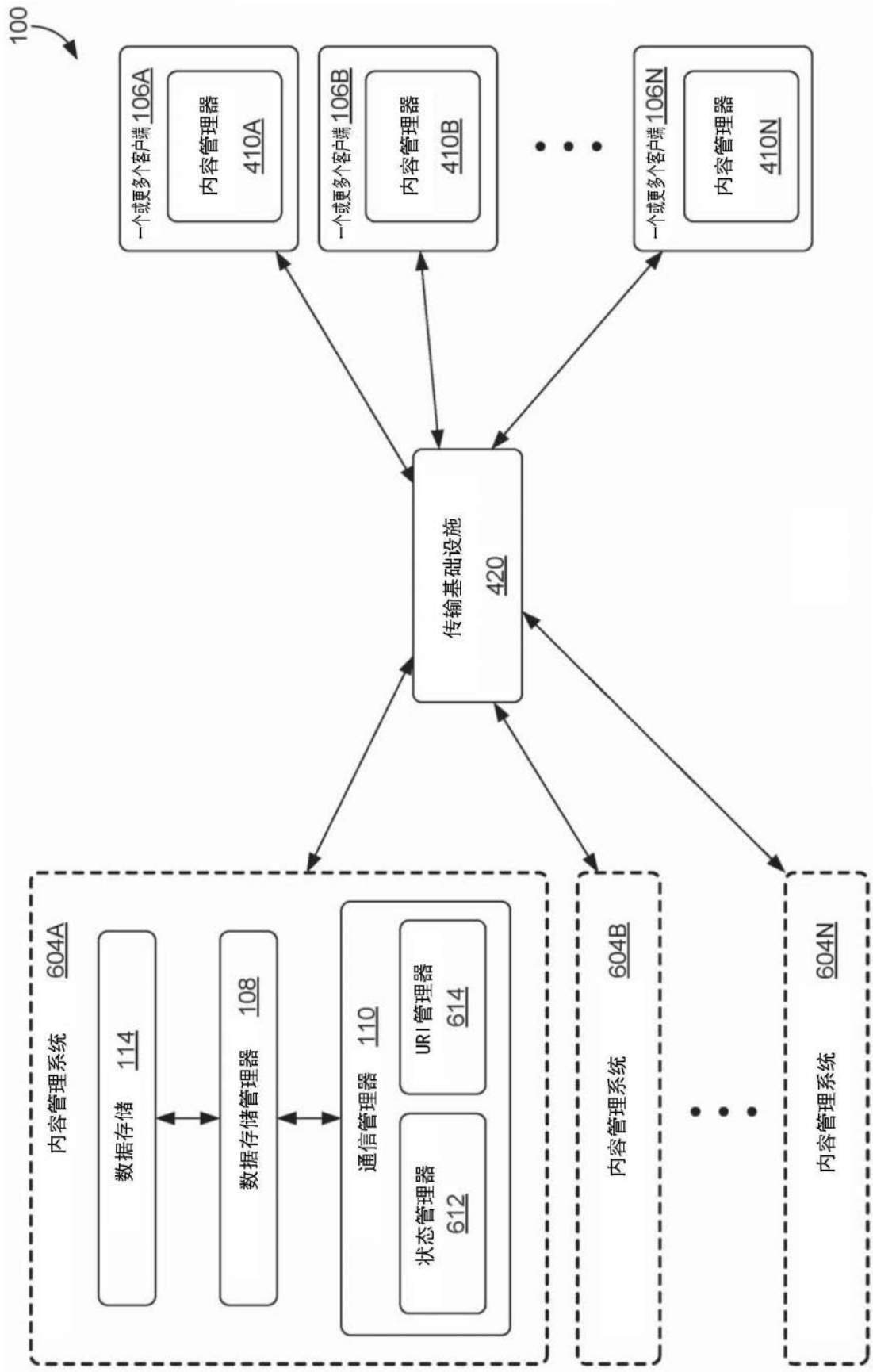


图6

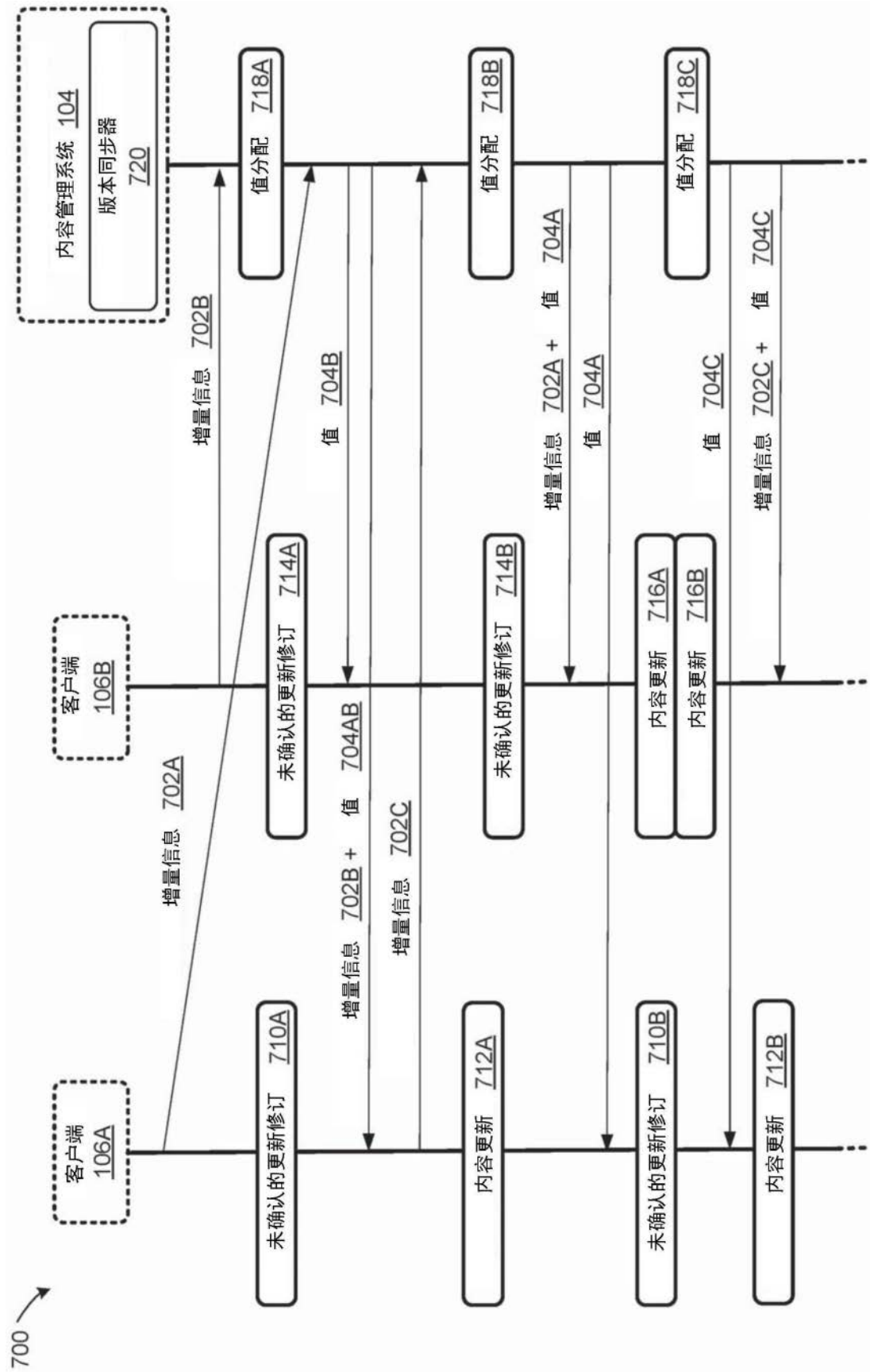


图7

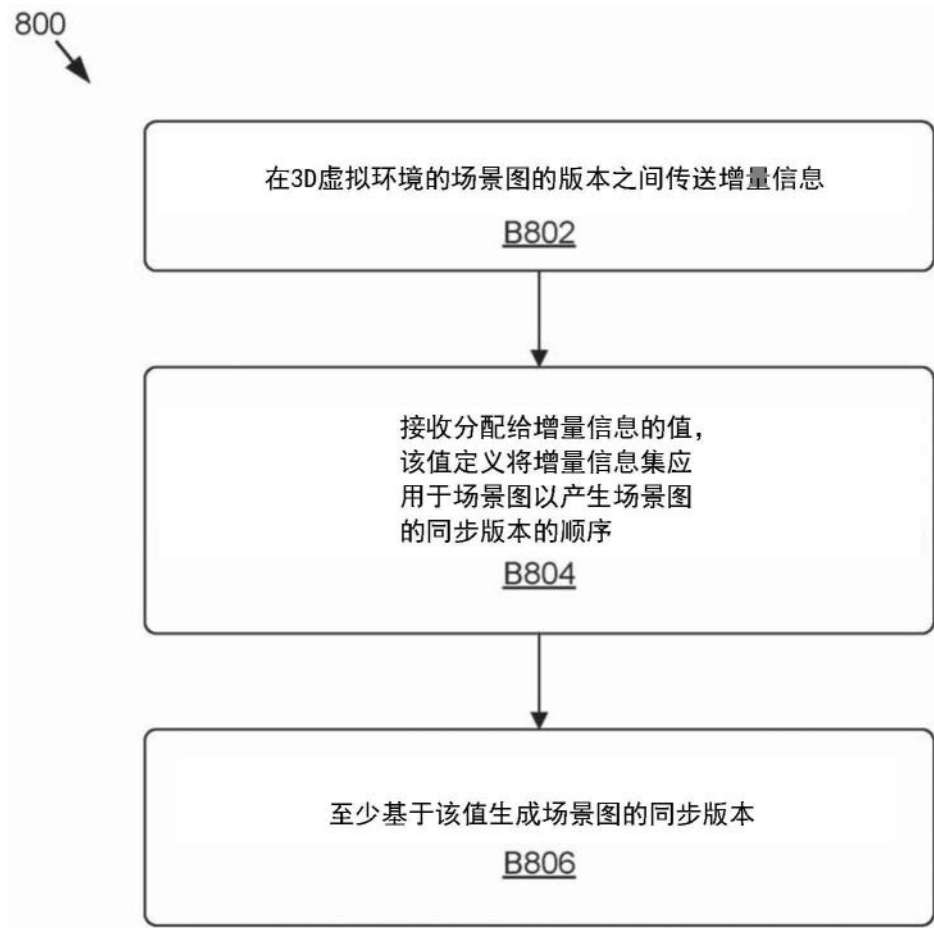


图8

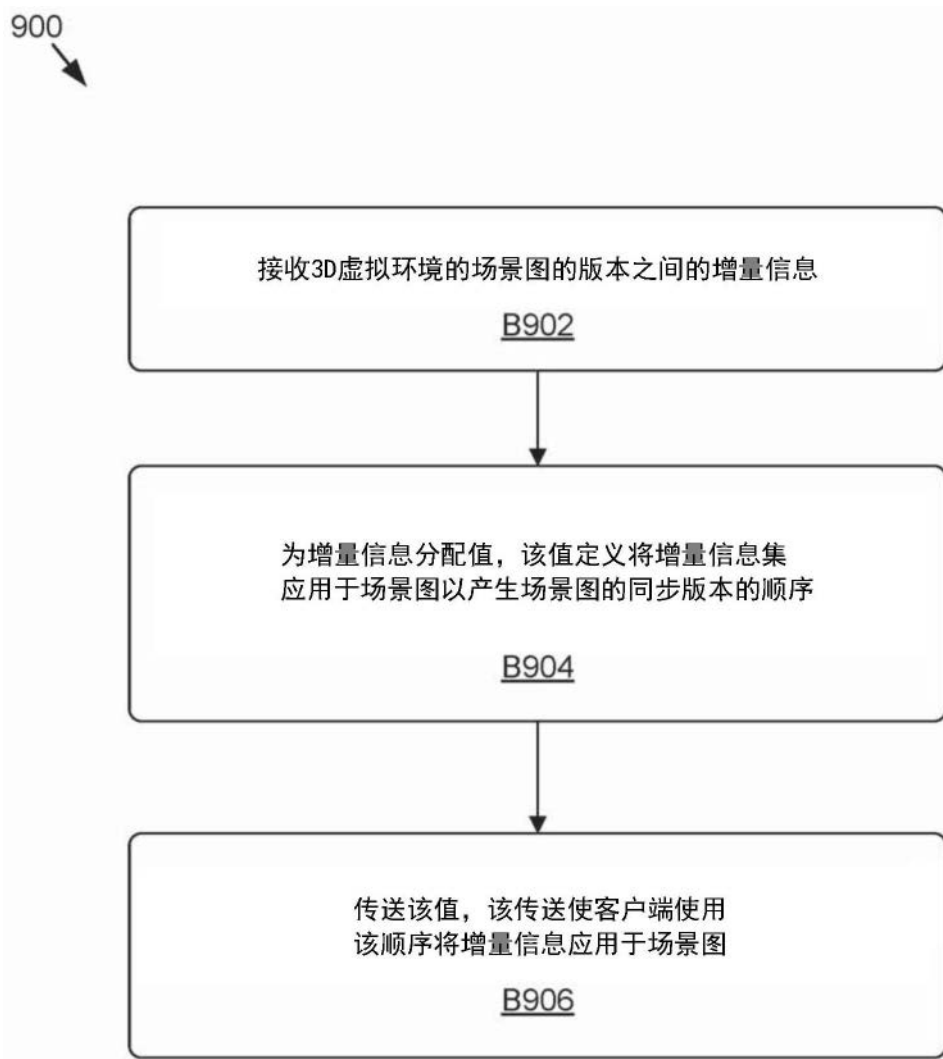


图9

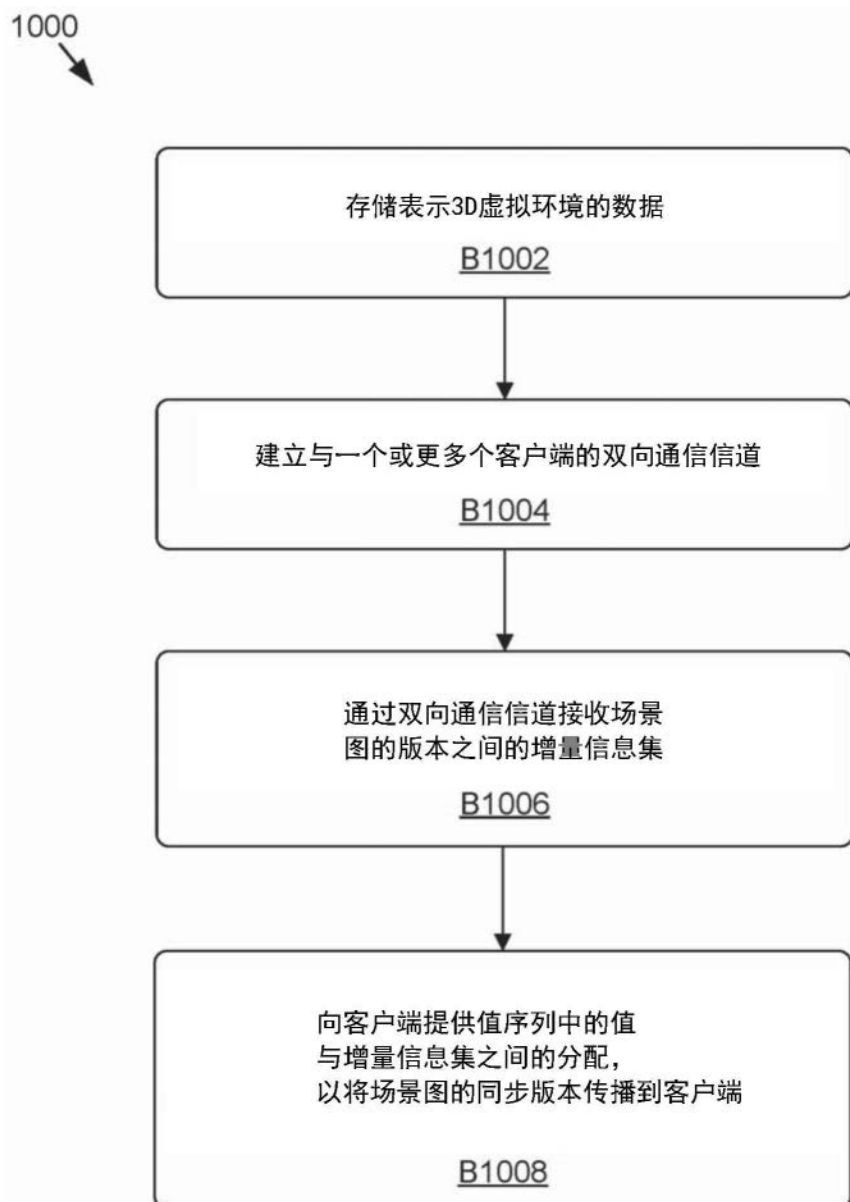


图10

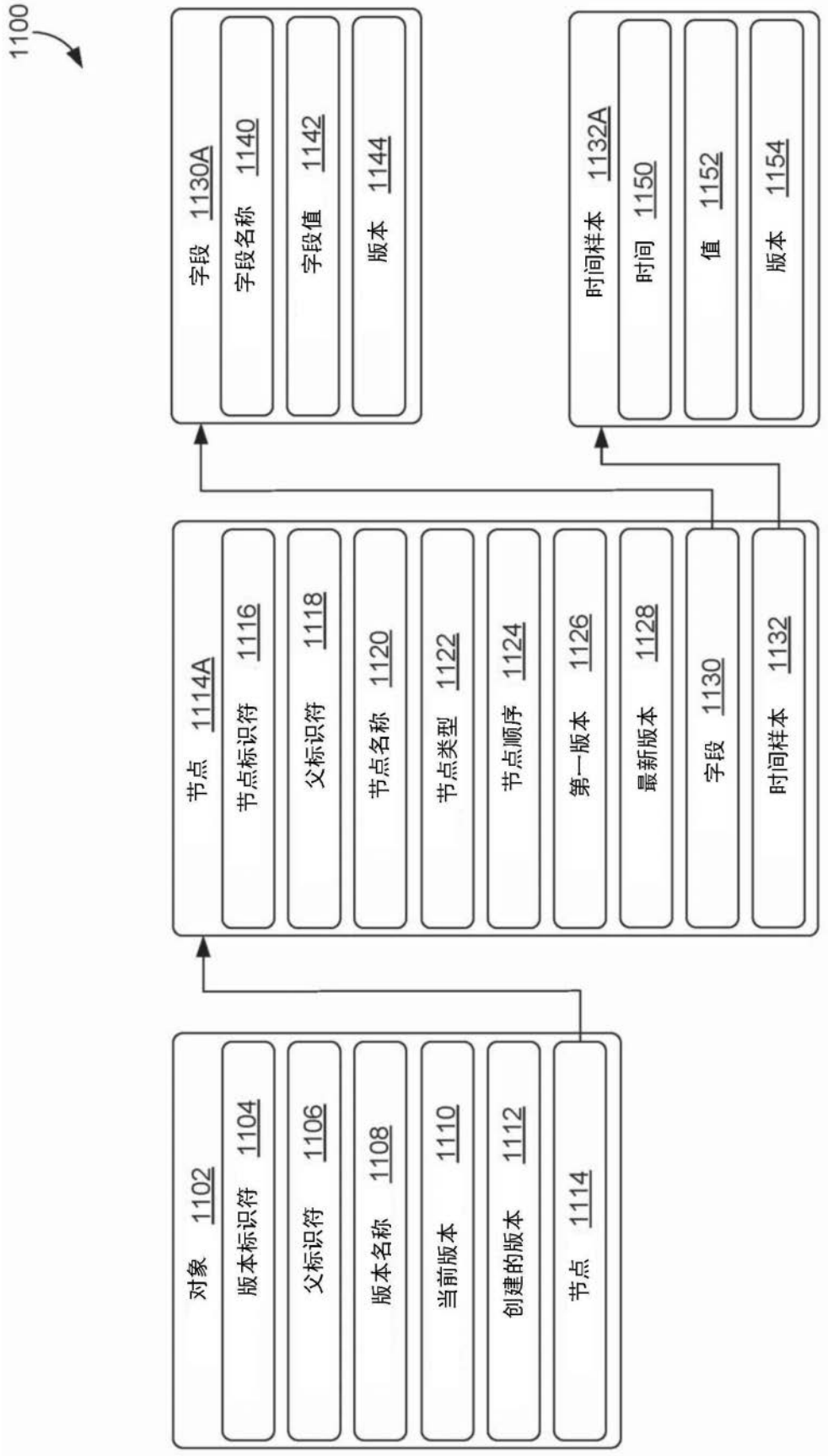


图11

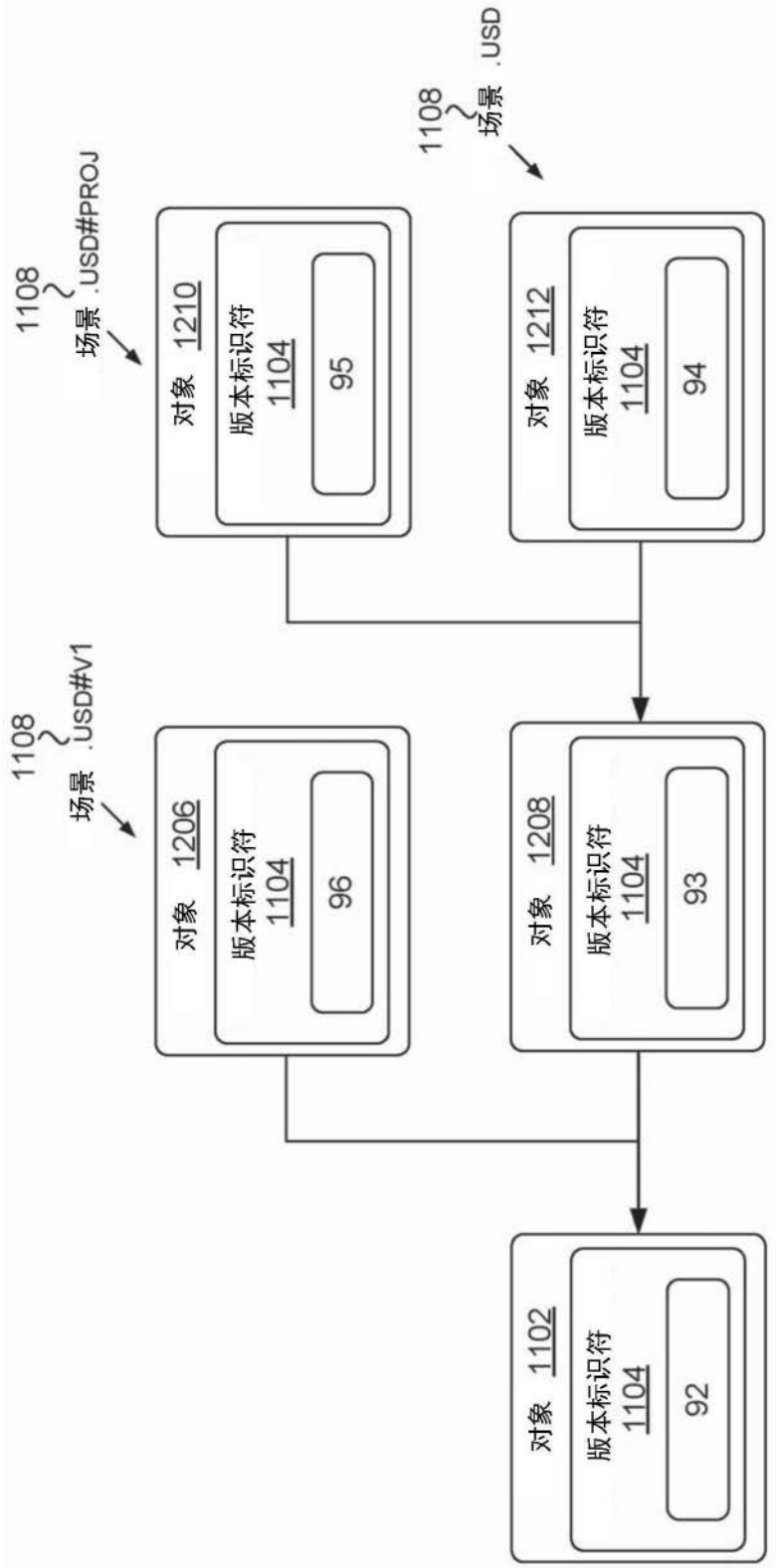


图12A

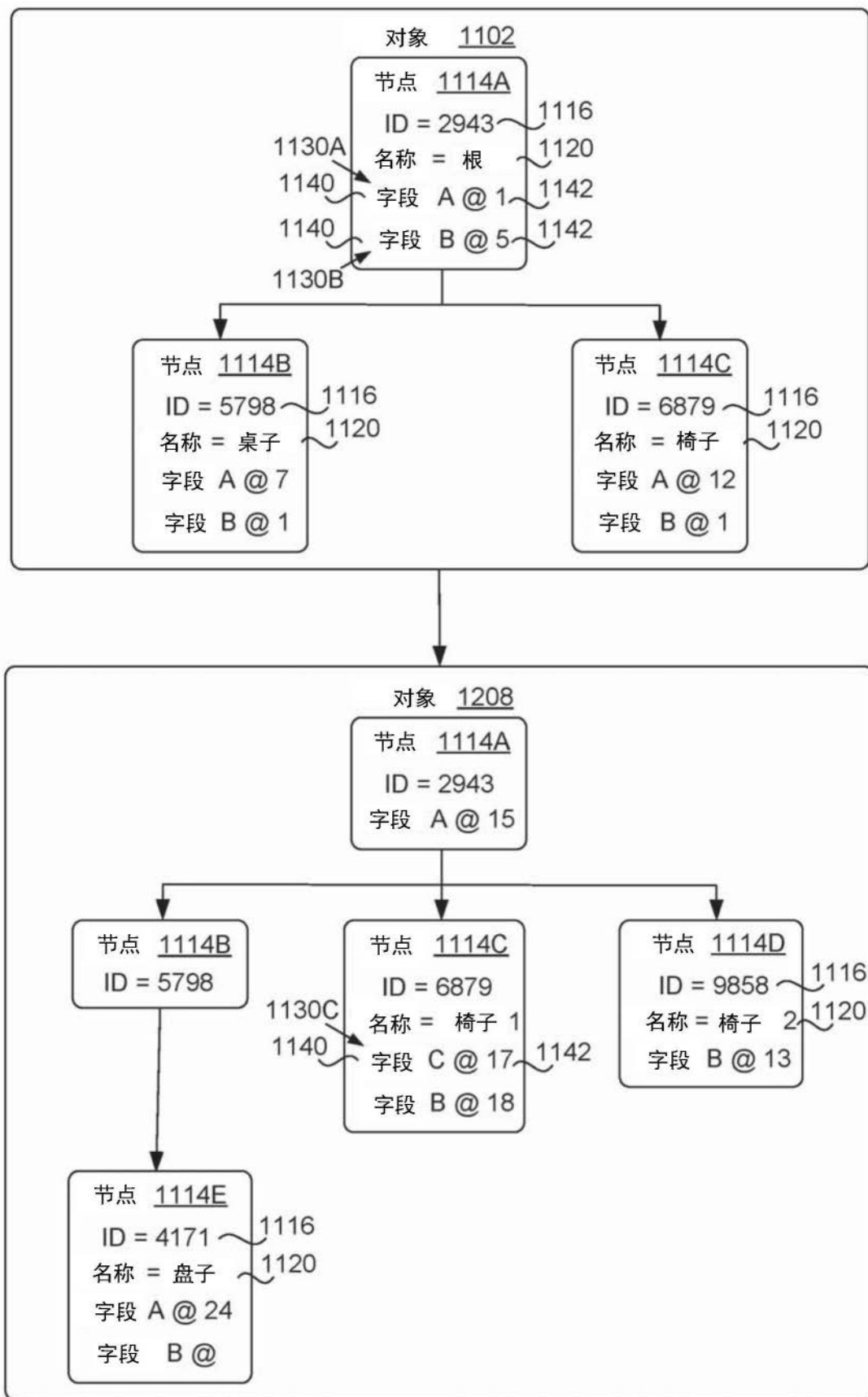


图12B

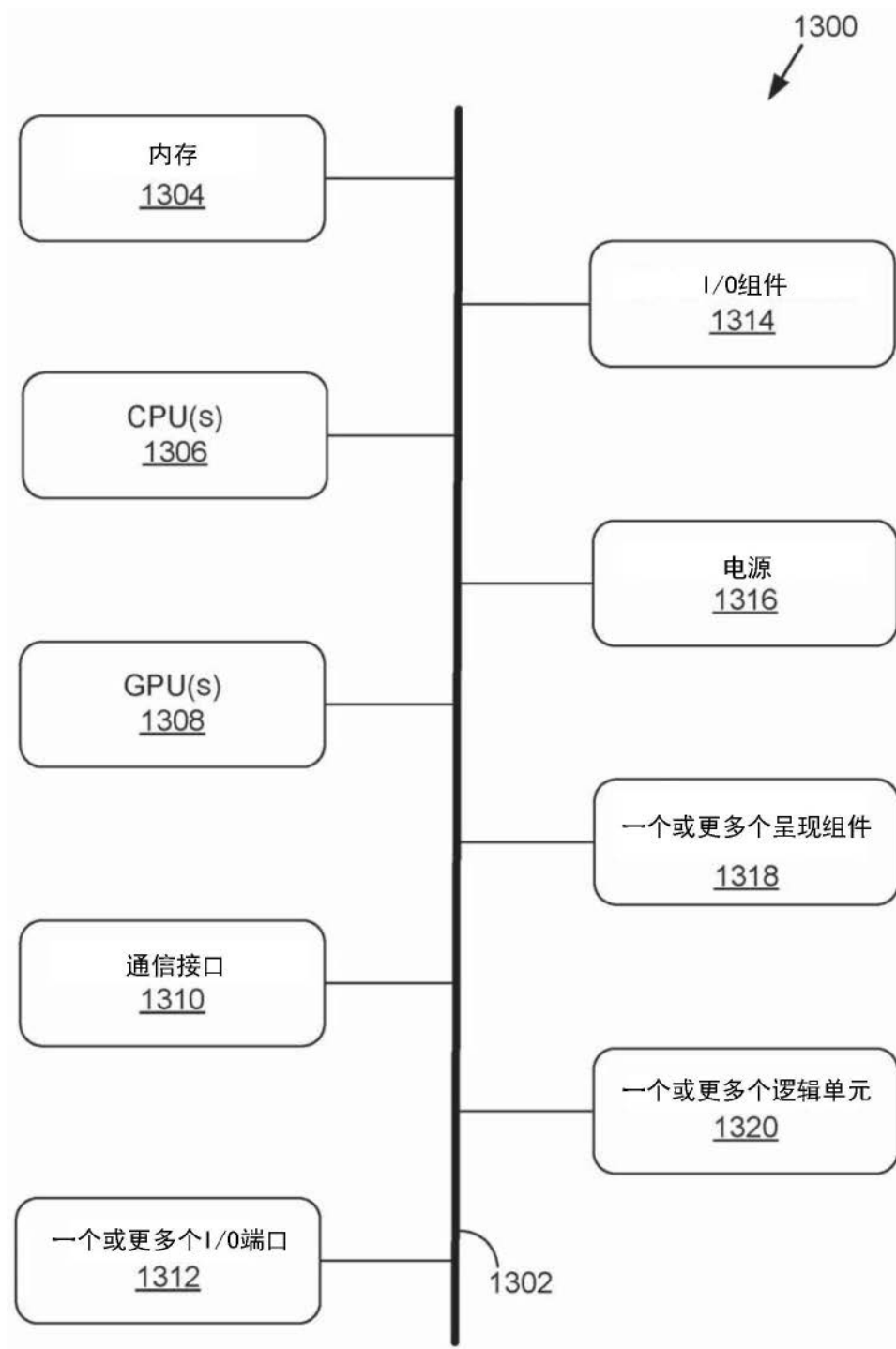


图13



图14