



(12) 发明专利申请

(10) 申请公布号 CN 102257489 A

(43) 申请公布日 2011. 11. 23

(21) 申请号 200980151830. 1

A · 阿加瓦尔 S · 索内 J · R · 凯然

(22) 申请日 2009. 11. 18

(74) 专利代理机构 上海专利商标事务所有限
公司 31100

(30) 优先权数据

12/337, 648 2008. 12. 18 US

代理人 陈斌

(85) PCT申请进入国家阶段日

2011. 06. 17

(51) Int. Cl.

G06F 17/00 (2006. 01)

G06F 3/14 (2006. 01)

(86) PCT申请的申请数据

PCT/US2009/064905 2009. 11. 18

(87) PCT申请的公布数据

W02010/080212 EN 2010. 07. 15

(71) 申请人 微软公司

地址 美国华盛顿州

(72) 发明人 E · A · 马丁内兹 S · 莱

R · R · 杰加德巴 A · N · 维什瓦纳坦

K · B · VS · 科拉萨拉 T · 巴蒂亚

R · 戈文德 N · 穆基加

权利要求书 1 页 说明书 11 页 附图 12 页

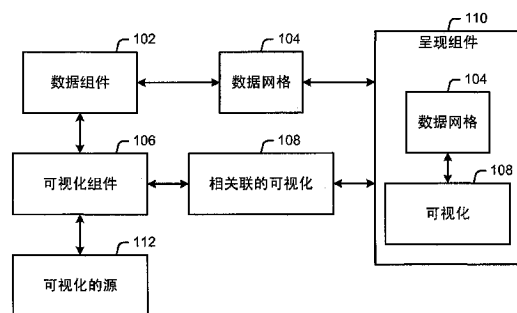
(54) 发明名称

网格和相关联的可视化的双向更新

(57) 摘要

一种便于将多个可视化类型（例如图表）附加到网格的数据上以及在该网格的旁边呈现该可视化的体系结构。该基础结构允许将每个数据视图与一组可视化相关联。这些可视化与网格（在分开的演示窗格中）一起呈现，这样使得用户无需切换上下文就能执行分析。可用的可视化包括示出聚集和 / 或非聚集数据的图表。在服务器农场的场景中，提供了诸如下钻等全部用户交互。可以使用 MHTML (MIME 超文本标记语言) 获取对这些可视化的服务器农场下钻。当在该网格上应用过过滤器时，可以自动地更新该可视化，并且当在该可视化上执行下钻时，自动地更新该网格。可以从例如基于 web 的浏览器客户端和 / 或厚客户端来消费该可视化。

100



1. 一种计算机实现的数据查看系统 (100), 包括
用于生成数据的网格的数据组件 (102);
用于将可视化与所述网格的数据进行关联的可视化组件 (106); 以及
用于结合所述网格数据呈现所述可视化的呈现组件 (110)。
2. 如权利要求 1 所述的系统, 其特征在于, 当将过滤器应用于所述网格数据时, 自动地更新所述可视化。
3. 如权利要求 1 所述的系统, 其特征在于, 响应于对所述可视化的交互式下钻, 自动地更新所述网格, 所述下钻是部分基于与选择下钻的图表的数据点的相关的图表参数以及编组依据参数执行的。
4. 如权利要求 1 所述的系统, 其特征在于, 在基于服务器的数据查看应用程序中, 所述可视化是紧接数据来呈现的。
5. 如权利要求 1 所述的系统, 其特征在于, 所述可视化是第三方插件。
6. 如权利要求 1 所述的系统, 其特征在于, 所述可视化是使用响应于单个下钻请求接收的图像和图像地图两者形成的。
7. 如权利要求 1 所述的系统, 其特征在于, 还包括用于相对于彼此操纵所述网格和所述可视化的合成控制组件。
8. 如权利要求 1 所述的系统, 其特征在于, 还包括用于响应于下钻过程来流传送图表图像的制图组件。
9. 一种计算机实现的查看数据的方法, 包括:
生成数据的网格 (800);
将一个或多个可视化与所述网格的数据相关联 (802);
紧接所述网格呈现原始可视化 (804); 以及
响应于对所述网格数据的操作或与所述原始可视化的交互, 选择不同的并且紧接所述网格呈现所述可视化 (806)。
10. 如权利要求 9 所述的方法, 其特征在于, 还包括响应于过滤所述数据来更新所述原始可视化。
11. 如权利要求 9 所述的方法, 其特征在于, 还包括基于对所述原始可视化的下钻来更新所述网格和网格数据。
12. 如权利要求 9 所述的方法, 其特征在于, 还包括构建将图像和图像地图嵌入到对可视化的下钻的单个响应中的多内容类型响应, 其中所述可视化是图表。
13. 如权利要求 9 所述的方法, 其特征在于, 还包括在网页中生成便于所述网格和可视化的相对的大小调整以及布局的合成控件。
14. 如权利要求 9 所述的方法, 其特征在于, 还包括检查可视化与所述网格数据的一致性。
15. 如权利要求 9 所述的方法, 其特征在于, 所述数据与顾客关系管理相关并且可视化将所述数据表示为图表。

网格和相关联的可视化的双向更新

[0001] 背景

[0002] 计算硬件和软件的进步促进了存储和检索大量信息的能力。网络使得该信息可以从实际上任何位置进行访问。由此,用户具有搜索和检索诸如网格中的数据等许多不同格式的数据的能力。然而,在许多情况下,所返回的数据是表格形式的,这样使得用户将不得不用智力分析关于数据的关键信息段,诸如趋势、高值和低值等。

[0003] 数据分析用户在很久以前就确定提供数据的图形表示可很大程度上协助更多地理解数据。例如,饼图、柱状图等就有关数据的特性提供了信息丰富的可视化。这种表示可以在允许用户选择数据集并且接着将该数据表示为例如图表的电子表格应用程序中找到。

[0004] 顾客管理系统是顾客数据的有价值的存储库。然而,尽管诸如在 SQL 服务器服务中提供的报告解决方案允许顾客关系管理用户得出一些见解,但是存在若干缺点。例如,某些缺点包括缺乏与网格的紧密集成、缺乏交互性,以及还需要将另一软件模块安装在系统上。此外,在没有有用的可视化的协助的情况下,用户难以获取否则可能在该数据中找到的见解。

[0005] 概述

[0006] 以下提出了简化概述以提供对此处所描述的一些新颖的实施例的基本理解。本概述并不是全面综述,也不旨在标识关键/重要的元素或描绘其范围。其唯一目的是以简化的形式来介绍一些概念,作为稍后呈现的更为详细的描述的序言。

[0007] 所公开的体系结构便于将多种可视化类型附加到(例如网格中的)数据上。该基础结构允许将每个数据视图与一组可视化相关联。可视化的一个示例与图表、制图模块以及可视化的其他应用方面相关联。

[0008] 可视化特征允许基于网格中的数据来呈现可视化。这些可视化与网格(在分开的演示窗格中)一起呈现,这样使得用户无需切换上下文就能执行分析。可用的可视化包括显示聚集和/或非聚集数据的图表。在服务器农场的场景中,提供了诸如下钻(drill-down)等全部用户交互。在 MHTML(MIME 超文本标记语言)使用的 MIME 格式(例如消息/rfc822)的帮助下仅使用单个页面请求-响应或服务器往返来获取服务器农场下钻。用于所有特征(包括下钻)的代码库在多服务器部署、单服务器部署以及可在因特网上使用的部署之间是一致的。换言之,该体系结构包括可以在单服务器模式、多服务器模式、面向因特网模式中运行并且也可以在浏览器客户端或诸如 PIM(个人信息管理器)应用程序(例如微软公司的 Outlook™)的插件(例如顾客关系管理)的独立客户端中使用的应用程序。

[0009] 此外,当将过滤器应用到网格上时,可自动地更新可视化。相反,当对可视化执行下钻时,就自动地更新网格。网格以及相关联的可视化的这一双向更新特征向用户提供了唯一且信息丰富的呈现。

[0010] 为达到上述和相关目的,此处结合以下描述和附图描述了某些说明性方面。这些方面指示了可以实现本文所公开的原理的各种方式,并且所有方面及其等效方面旨在落入所要求保护的的主题的范围内。结合附图阅读下面的详细描述,则其他优点和新颖特征将变

得清楚。

[0011] 附图简述

[0012] 图 1 示出了根据所公开的体系结构的计算机实现的数据查看系统。

[0013] 图 2 示出了替换性的数据查看系统。

[0014] 图 3 示出了描绘其中结合网格来呈现图表窗格的可视化的示例性用户界面。

[0015] 图 4 示出了示例性状态图以及附属的可视化。

[0016] 图 5 示出了用于向客户端呈现可视化的逻辑流程图。

[0017] 图 6 示出了概括的数据流程图。

[0018] 图 7 示出了详细的数据流程图的示例。

[0019] 图 8 示出了查看数据的方法。

[0020] 图 9 示出了基于对网格数据的过滤来更新可视化的方法。

[0021] 图 10 示出了基于对可视化的交互性改变来更新网格数据的方法。

[0022] 图 11 示出了根据所公开的体系结构的可用于执行网格和可视化组合的计算系统的框图。

[0023] 图 12 示出了用于基于 web 的网格 / 可视化呈现的计算环境的示意性框图。

[0024] 详细描述

[0025] 所公开的体系结构包括允许在网格旁边呈现可视化（例如图表）的可视化特征。可以用可插入的方式添加这些可视化的各种提供者。例如，可视化提供者可以是 .Net 图表模块，其呈现来自任何网格（例如客户关系管理（CRM）环境）的聚集和非聚集数据的图表。提供者是可用于为浏览器客户端或处理某些视图的个人信息管理器（PIM）（例如微软公司的 Outlook™）的插件客户端呈现可视化的应用程序。换言之，在 CRM 上下文中，PIM 客户端不仅是 PIM，而是以 PIM 插件形式可用的整个 CRM 应用程序。由此，PIM 内的可视化是 PIM 的 CRM 插件的。可视化在 PIM 离线以及正常模式中可用。

[0026] 在诸如客户关系管理（CRM）应用程序等基于服务器的应用程序的上下文中，可以在浏览器窗口内部示出网格。例如，也可以在个人信息管理器（PIM）CRM 客户端中示出网格。当在在线（例如因特网上）环境中部署 CRM 时，它支持多租户。当在因特网上的服务器环境中部署应用程序时，例如，该应用程序可以在多服务器环境中运行。

[0027] 诸如用网格视图使得其可用的图表等可视化提供了“平滑的”用户界面（UI）以允许用户访问和使用图表。每个网格提供 CRM 中数据的视图。每个视图基本上是从 CRM 数据库检索数据并在网格中示出该数据的查询。基于查询返回的数据集来呈现可视化。即使在若干个页面中示出网格中的数据，可视化也是从整个数据集（所有页面的数据）作出的。这允许用户从查看数据中获得见解，该见解否则不能在传统的网格页面中看出。

[0028] 这种解决方案比例如基于仪表板和报告来得出见解，而不能快速地在数据旁提供上下文内可视化的其他实现更加新颖。上下文内可视化会允许用户从执行实际工作的地方获得见解。

[0029] 例如，所公开的体系结构允许用户从描绘网格上的数据的聚集和非聚集的概要可视化来得出快速的见解，并允许可视化反映网格上的用户动作，反之亦然，即，使用过滤器更新网格数据会更新可视化并且可视化（图表）上的下钻会更新网格数据，并且允许在服务器农场的场景中使用。此外，允许多个可视化提供者。

[0030] 在一个实施例中,如在图 4 的状态图中所示,当用户手动选择可视化上的“更新可视化”按钮时,可更新可视化。以下是可以响应于这种更新选择发生的一系列动作。用户查看可视化和网格。用户在一列上应用过滤器。可视化现在变得陈旧,并且示出“更新可视化”按钮(例如,按钮图像淡入并且在背景中示出)。用户应用更多的过滤器以缩减数据集。用户接着点击“更新可视化”按钮以查看经更新的可视化。

[0031] 这种方法的优点是因为可视化不是自动更新的,所以无需在每次将过滤器应用到网格上时刷新可视化。由于应用每个过滤器时都会更新可视化,因此在应用每个过滤器时更新可视化会导致大量的性能开销。此外,由于当用户应用新的过滤器时,先前过滤器集的可视化可能会正在使用,因此自动更新对于用户而言会是可使用性的问题。这种行为会使用户感到混淆;因此,手动更新提供了更为可行的方式。然而,应该理解,这不应该被解释为限制性的,因为所公开的体系结构可按需采用自动更新。

[0032] 如此处所使用的,图表描绘了视图中的各列的聚集和非聚集数据。图表是一种具体类型的可视化。聚集函数可以包括例如求和、计数、最小值、最大值和平均值。可以按需添加其他聚集函数。也可以在离线模式中呈现可视化。离线模式是指即使用户不连接到服务器,PIM 插件也允许用户操作数据/网格/可视化的能力。该插件允许用户使用与 PIM 的框架内的浏览器客户端中可用的功能相同的功能。

[0033] 现在参考附图,所有附图中使用相同的参考标号来指代相同的元素。在以下描述中,为解释起见,阐明了众多具体细节以提供对本发明的全面理解。然而,很明显,各新颖实施例可以在没有这些具体细节的情况下实施。在其它情况下,以框图形式示出了公知的结构和设备以便于描述本发明。旨在涵盖所有这些落入所附权利要求书的精神和范围内的改变、等效方面和变动。

[0034] 图 1 示出了根据所公开的体系结构的计算机实现的数据查看系统 100。系统 100 包括用于生成数据的一个或多个网格 104 的数据组件 102、用于将一个或多个可视化 108 与网格 104 的数据进行关联的可视化组件 106 以及用于结合网格数据来呈现可视化 108 的呈现组件 110。可视化组件 106 可访问可视化的源 112。源 112 可以是例如作为辅助代码或“插件”来提供可视化的第三方供应商。

[0035] 此处所描述的其他方面包括呈现聚集数据和/或非聚集数据的网格 104,当将过滤器应用到网格数据上时可自动地更新可视化 108,以及可以响应于可视化上的交互式下钻来自动地更新网格。此外,在基于服务器的应用程序中可视化 108 被在紧接网格处呈现,并且可使用响应于单个下钻请求接收到的图像和图像地图两者来形成可视化。也可以实现可视化的“粘性”行为,这样使得即使用户导航离开网格并返回,网格也可继续看到相同的可视化。

[0036] 注意,尽管在系统 100 各实体之间使用双向箭头来描绘,但是这并不是必须的。例如,可视化的源 112 无需消费来自可视化组件 106 的信息。

[0037] 图 2 示出了替换性的数据查看系统 200。系统 200 包括用于生成数据的一个或多个网格 104 的数据组件 102、用于将一个或多个可视化 108 与网格 104 的数据进行关联的可视化组件 106 以及用于结合如由可视化的源 112 所访问和提供的网格 104 或网格数据来呈现可视化 108 的呈现组件 110。

[0038] 在该实施例中,系统 200 还示出了提供使用(例如,基于 .Net 的)制图库来显示

数据的逻辑的制图组件 202。制图组件 202 还包括将图表图像流传送到客户端的逻辑、对特定图表下钻、本地化、国际化、数据格式化、使用（例如基于 .Net 的）模板 XML 来配置图表以及应用一系列数据点到图表上。接着由呈现组件 110 在可视化窗格中示出制图组件 202 生成的图表。

[0039] 为了支持这一点，呈现组件 110 还包括定义用于所有 web UI 控件的界面的合成控制组件 204，这样使得可以将控制组件 204 包括在页面上。注意，所公开的体系结构可用于商业应用程序，诸如在客户关系管理（CRM）中用于客户信息的显示和可视化。相应地，合成控制组件 204 可以包括作为子控件的 CRM 网格控件以及可视化窗格控件。可以重新包装 CRM 网格控件以便包括在实体页面上。可视化窗格控件可以是 ASP.NET 用户控件以及用于可视化的容器。

[0040] 合成控制组件 204 可以被用作实体页面上的独立控件。此外，合成控制组件 204 允许用户为网格和窗格选择相关的尺寸和布局。可以隐藏可视化窗格或将其扩展为占据在其中呈现网格的整个空间，并且可以在与网格相关的位置中示出它，诸如例如在网格的右侧或顶部。因此，可以在页面上可放下网格控件的所有位置处放下合成控件。合成控件允许可视化窗格与网格一起示出。

[0041] 系统 200 还可以包括应用程序制图基础结构模块 206，该模块是使制图组件 202 和合成控制组件 204 能够运作的客户端例程 208 和服务端例程 210 的集合。该基础结构包括用于构建聚集获取请求、取得数据、通过扩展查询来启用下钻、将数据转换成可以消费的格式（例如 .Net）、基于可用的可操作区域计算图表的像素大小的方法等。当采用获取请求时，可使用依照在请求中指定的查询从数据库返回数据的 web 服务调用来执行该请求。模块 206 还可以包括在现有图表变得陈旧时使得现有图表淡出的逻辑。模块 206 也可以包括例如控制可视化和可视化的管理（例如移除 / 编辑 / 设为默认 / 设置）的菜单（例如功能区（ribbon））动作。

[0042] 合成控制组件 204 可便于相对于彼此操纵网格和可视化。制图组件例如响应于下钻过程将图表图像流传送到客户端。

[0043] 换而言之，数据查看系统 200 可以包括用于生成数据网格的数据组件 102、用于将一个或多个可视化与网格 104 的数据进行关联的可视化组件 106，以及用于在基于服务器的应用程序中在紧接网格 104 处呈现该一个或多个可视化的呈现组件 110，其中一个或多个可视化呈现网格上的数据的聚集和非聚集的概要。

[0044] 可视化组件 106 构建将图像数据和 HTML 文档嵌入到单个响应中的多内容类型响应。当将过滤器应用到网格数据上时，如上所述地自动更新可视化，并且响应于对可视化的交互式下钻来自动更新网格。该系统还可以包括用于相对于彼此操纵网格和可视化的合成控制组件 204 和用于响应于下钻过程来流传送图表图像的制图组件 202。

[0045] 也可采用所公开的体系结构以允许以下情况。可通过给予“*What If*”（假设分析）分析能力来实现双向通信的强大的益处之一。网格可以具有可操纵来确定（查看）对可视化的影响的上下文内的可编辑字段。

[0046] 此外，可视化窗格可以被用作分析白板，其中用户可以放下来自网格的任何列并且动态地查看相关联的可视化。这给予商业用户灵活性以及活动式（*on-the-go*）分析。

[0047] 此外，经由一个地方从来自多个源的数据得出见解以获取完整的图片是有价值的

益处。例如,企业操作各个集成系统并且具有从一个系统流入到另一系统的数据。具有从其他数据源拉出数据的能力可以给出更为丰富的分析体验。

[0048] 另一有益的情况涉及报告和协作。可通过将数据导出到电子表格或在电子邮件上将数据发出或将数据发送到打印机来获取共享详细的且综合的信息。

[0049] 对各实体的分析提供了从跨多个相关的实体的观点和参数来分析数据的丰富的方式。可以通过从嵌套的链接实体(例如,按区域分组的产品)中挖掘数据来得出深入的见解。可以选择帐户并且返回相关联的订单。订单通向产品和例如地图中的按照收益的产品地理分布。

[0050] 此外,可视化窗格可以用作网格中的所选记录的预览/历史窗格。这给予销售代理例如快速获取该代表正在与之通话的顾客の詳細资料的强大方式。预览窗格也可以示出对该记录的评论或粘贴笔记。

[0051] 注意,尽管本说明书的某些领域是在 CRM 实现的上下文中描述的,但是应该理解本发明不限于此。

[0052] 图 3 示出了描绘可视化 302 的示例性用户界面 300,其中结合网格 306 来呈现图表窗格。网格 306 列出了数据并且图表窗格 304 呈现对应于销售流水线的漏斗图表 308,用于与在网格 306 的页面上示出的所有行相关联的数据的快速且容的可视化。可视化 302 示出到机会所在的阶段为止机会的聚集。例如,可视化 302 示出“开放”阶段中的所有机会的聚集,所估算的收益之和为 \$415,000。同样,这可以是聚集或非聚集数据。还构想了基于在网格上选择的行集来启用可视化。例如,地图可视化可以突出显示机会所位于的地图区域。

[0053] 可以自动地为网格 306 中的所有行式项目生成不同或相同类型的图表(例如漏斗、柱、散点、行、列、饼等)。例如,可以自动地生成柱状图并且结合所选的行式项目来呈现柱状图。

[0054] 可以操纵可视化 302 以便可以使网格 306 的大小适合消费整个可视化覆盖区(footprint),可以调整图表窗格 304 的大小以消费整个可视化覆盖区,或者这两者当中的任何变型。

[0055] 下钻被表示为图像(实际或原始图表)和图像地图(图表中可钻区域)的组合。通过生成客户机侧图像地图在图表图像上启用下钻。为了创建可下钻的可视化,基于图像的服务器控件(例如.NET 图表)通常在请求页面期间生成图像,并且将图像与图像地图一起呈现为 HTML 中的 标签。然而,客户端浏览器通常必须发出附加的请求以下载在标签中引用的实际图像。因此,用于生成图像地图的图像必须与在第二请求中发送的图像一致。然而,在服务器农场的环境中,可能不能总是确保这样,因为不能在多个进程之间共享存储在存储器中的图像。此外,出于安全性的考虑,图像没有被持久保存在数据库或共享的文件系统中。

[0056] 对该问题的一个解决方案采用了多内容类型响应,该多内容类型响应将图像和 HTML 嵌入到单个响应中,由此避免多个请求。为了实现这一点,在对客户机请求的一个响应中,充分利用 MHTML(MIME HTML)格式来将 HTML 文档(包含图像地图)和图像作为一个多部分的/相关的文档发送,这也节省了服务器上的额外开销、通信时间以及线路带宽的消耗。这种解决方案确保在服务器农场的情况中的正确的下钻行为。

[0057] 图 4 示出了示例性状态图 400 和附属的可视化。第一状态 402 是其中用户界面示

出不带可视化的网格,换而言之,用户仅使用网格。第二状态 404 是其中用户界面示出带有到可视化的转移的网格,换而言之,呈现对应于网格上的数据的可视化。第三状态 406 是其中用户界面示出带有到可视化的转移的网格,并且可视化与网格数据相一致。换而言之,可视化是结合网格呈现的并且与网格数据相一致。第四状态 408 是其中呈现网格,但是可视化与网格数据不相一致。换而言之,在将一个或多个过滤器应用到网格上之后,可视化变得无效。第五状态 410 是其中呈现带有可视化下钻的网格,即对可视化执行了下钻。第六状态 412 是其中呈现网格,但是因为在下钻之后应用了一个或多个过滤器,所以可视化下钻不一致。

[0058] 状态图 400 还示出了用户界面提供在用于用户交互的图表窗格和用于用户交互的网格之间进行切换的按钮。示出了其他状态转移,诸如例如刷新不一致的状态、更新可视化。

[0059] 图 5 示出了用于向客户端 502 呈现可视化的逻辑流程图 500。第一客户端 502 可以是例如浏览器或者可以呈现信息的其他客户端。第二客户端 504 可以是脚本客户端。可视化应用程序 506 可以是诸如例如使用页面后的.aspx 文件以及代码以及服务器侧处的 web 服务方法的 Asp.net 等 web 应用程序框架。可视化平台 508 可以是支持诸如 CRUD(创建、检索、更新、删除)、导入、导出等对可视化的一般操作的各种 API 和调用。制图模块 510 是指专用于处理图表(例如取得数据并将数据供应给图表、验证可视化的 XML 等)的平台侧以及应用程序侧代码。

[0060] 最初,在 1) 处,第一客户端 502 向应用程序 506 请求带有图表的页面。在 2) 处,应用程序 506 从平台 508 请求关于该视图的所有相关联的可视化。平台 508 获取相关联的可视化并用这些可视化对应用程序 506 进行响应。在 4) 处,应用程序 506 将带有用于图表和所有相关联的可视化的占位符的页面发送给第一客户端 502。

[0061] 在 5) 处,第二客户端 504 包括制作将图表呈现给制图模块 510 的脚本的模块。在 5.1) 处,制图模块 510 将验证可视化定义的请求发送给平台 508。在 5.2) 处,平台 508 用验证定义结果(如,已验证)来对制图模块 510 进行响应。在 6) 处,制图模块 510 使用网格 XML 来查询数据。在 7) 处,平台 508 将聚集数据集返回给制图模块 510。在 8) 处,制图模块将图表的 HTML 返回给第二客户端 504。

[0062] 图 6 示出了概括的数据流程图 600。如此处所使用的,ViewID(视图 ID)标识可以运行以取得数据来在网格中进行示出的特定数据库查询。查询包括实体的名称以及要与过滤数据的条件一起取得的实体列。可视化 ID(vizID)标识特定的可视化。可视化是对作为对视图应用过滤器条件的结果得出的数据进行安排的方式。可视化定义要取得哪些列、要应用什么聚集函数(仅适用于聚集)以及如何在视觉上呈现数据(例如示出柱状图)。

[0063] 在一个实现中,取决于用户查看合成控件的模式(例如,仅网格、网格+可视化、或仅可视化),为网格上的给定页面取得一次数据并且为绘制可视化取得一次数据。注意,视图和可视化都在数据行的相同集合上操作,不同之处在于视图返回整个行集的子集,而可视化返回所有的数据行的经处理的版本(例如,总和)。

[0064] 在 1) 处,将网格视图 ID(viewID)、过滤器参数和 vizID(可视化 ID)从 web 客户端 502 传送到可视化应用程序 506。在 2) 处,将 vizID 从应用程序 506 处传送到平台 508。在 3) 处,平台 508 查询数据库 602。在 4) 处,数据库 602 将结果返回给平台 508。在 5) 处,应

用程序 508 以 XML 文件的形式将数据和呈现信息发送给平台 508。在 6) 处,应用程序 506 也以 XML 文件的形式将数据和呈现信息发送给可视化模块 604。在 7) 处,模块 604 将验证请求发送给取得 XML 的平台 508,而平台 508 将结果发送回模块 604,如 8) 处所示。在 9) 处,模块 604 将 HTML 信息发送给应用程序 506。在 10) 处,将可视化从应用程序 506 发送到 web 客户端 502。

[0065] 如先前所示,所公开的体系结构提供了呈现给用户的网格和相关联的可视化的双向更新特征。换言之,当在网格上应用过滤器时,可以自动地更新可视化,而当对可视化执行下钻时,自动地更新网格。

[0066] 在一个简化的示例中,假定用户正在查看示出销售代表的团队的机会的总和的图表。用户想要钻研具有最大份额的估计收益的特定销售代表 Rep A 的细节。通过选择图表,向用户呈现用户选择要在其上下钻的范围(例如州)和图表类型(例如饼图)的菜单。系统按照州呈现 Rep A 的机会,并且在可视化更新旁呈现网格以示出 Rep A 的机会。基于新的可视化,用户接着能够在视觉上推测 Rep A 在州 B 具有最大的机会。

[0067] 可以如下地计算下钻图表。通过在以上描述中的 Rep A 的基础上过滤视图,获取用于示出下钻图表的记录。当用户点击图像/图表上对应于 Rep A 的图像区域时,系统自动地理解要将 Rep A 的过滤准则应用到网格。如果在网格上已存在一些过滤器,则更新过滤器条件,这样使得条件将网格限制到基于现有的过滤器以及 Rep A 的记录上。用户可以按需添加/移除更多的过滤器条件并且可视化继续被更新(使用更新可视化按钮)。可以由最终用户更新/修改过滤器条件。

[0068] 通过用户从下钻查看依据(View By)选择器做出的选择来获取关于下钻图表的类别。用于下钻的测量和聚集函数与父图表(附加到视图上的可视化)相同。图表类型是从用户做出的选择(例如图表类型选择器)获取的。

[0069] 图 7 示出了采用下钻的详细的流程图 700 的示例。可以对附加在网格上的任何图表执行下钻。当用户点击图表的可下钻区域时,可以向用户呈现包含两个选择器的弹出式菜单,例如查看依据(View-by)选择器和图表类型选择器。在用户做出选择和点击 OK 按钮后执行下钻。由这两个选择器提供的值是下钻参数。查看依据选择器中的选项是在网格中显示的相同的/相关的实体的其他属性,在图表类型下钻中的选项是诸如例如饼、列、柱、漏斗和线等常见图表类型。换言之,当用户点击图表的特定组件(例如,柱状图中的柱、饼图中的一部分)以获取对组成该组件的数据的更精细的观察时,发生下钻。可以按照不同的参数对所得的数据进行编组。下钻参数包括至少两段信息:点击了图表的哪个组件(用户想要在哪个数据点上下钻)以及新的编组依据(group-by)的参数。

[0070] web 客户端 502 可以访问应用程序 web 服务 702 并且发送下钻参数和经刷新的网格数据。web 客户端 502 也可以通过将 viewID、vizID、下钻参数以及对 XML 的取得命令发送给使得能够将 HTML 文档嵌入到另一 HTML 文档内的 IFrame(内联帧)中的可视化页面 704 来请求可视化。接着将 vizID、所查看的、下钻参数和取得从页面 704 发送给可视化模块 604。模块 604 可以发送图表对象以生成图像 706,并且接着将该图像 706 流传送回模块 604。模块 604 可以发送图表对象以生成图像和下钻 HTML 708,并将 HTML 发送回模块 604。模块 604 发送图像流和 HTML 以构建多部分响应 710,接着将该响应 710 作为 MHTML 响应返回给模块 604。

[0071] 模块 604 也可以将 vizID、所查看的, 下钻参数和取得请求传递给数据 web 服务 712。web 服务 712 对可视化平台 508 执行取得, 且平台 508 将取得结果返回到服务 712。服务 712 将结果作为数据表传递给模块 604。模块 604 将 MHTML 返回给可视化页面 704, 接着将可视化页面 704 传递回 web 客户端 502 以便与网格一起呈现。

[0072] 此处包括的是代表用于执行所公开的体系结构的新颖方面的示例性方法的一组流程图。尽管出于简化解释的目的, 此处例如以流程图或流线图形式示出的一种或多种方法被可视化和描述为一系列的动作, 但应该理解和明白, 这些方法不受动作的顺序所限, 因为一些动作能够以与在此所示和所描述的所不同的顺序发生和 / 或与其他动作同时发生。例如, 本领域技术人员将会明白并理解, 方法可替换地被表示为一系列相互关联的状态或事件, 诸如以状态图的形式。此外, 对于新颖的实现可能并不需要在方法中所示的所有动作。

[0073] 图 8 示出了查看数据的方法。在 800 处, 生成数据的网格。在 802 处, 将一个或多个可视化与网格的数据相关联。在 804 处, 在紧接网格处呈现原始可视化。在 806 处, 响应于对网格数据的操作和与原始可视化的交互来选择并且在紧接网格处呈现不同的可视化。用户可以在各个状态之间导航, 诸如例如在 804 和 806 之间。

[0074] 该方法还可以包括构建将图像和图像地图嵌入到对可视化 (其中, 可视化是图表) 的下钻的单个响应中的多内容类型响应, 在网页中生成便于网格和可视化的相对的大小调整以及布局的合成控件, 以及检查可视化与网格数据的一致性。同样, 用户可以在这些状态之间来回导航。数据可以相关于顾客关系管理, 其中可视化将数据表示为图表。

[0075] 图 9 示出了基于对网格数据的过滤来更新可视化的方法。在 900 处, 生成并呈现数据的网格。在 902 处, 选择并在网格旁边呈现可视化。在 904 处, 将过滤器应用到网格数据。在 906 处, 基于经过滤的数据来自动更新可视化。

[0076] 图 10 示出了基于对可视化的交互式改变来更新网格数据的方法。在 1000 处, 生成并呈现数据的网格。在 1002 处, 选择并且在网格旁边呈现可视化。在 1004 处, 将交互式改变应用到可视化。在 1006 处, 基于对可视化的交互式改变来自动更新数据。

[0077] 尽管相对于作为屏幕截图的某些附图示出和描述了向用户显示信息的某些方式, 但是相关领域的技术人员会认识到可以采用各种其他替换。术语“屏幕”、“屏幕截图”、“网页”、“文档”和“页面”在本文中一般被可互换地使用。页面或屏幕作为显示描述、作为图形用户界面或通过描绘屏幕上 (例如, 无论是个人计算机、PDA、移动电话或其他适当的设备) 的信息的其他方法来存储和 / 或发送, 其中将要在页面上显示的布局和信息或内容被存储在存储器、数据库或另一存储设施中。

[0078] 如在本申请中所使用的, 术语“组件”和“系统”旨在表示计算机相关的实体, 其可以是硬件、硬件和软件的组合、软件、或者执行中的软件。例如, 组件可以是, 但不限于, 在处理器上运行的进程、处理器、硬盘驱动器、多个存储驱动器 (光学和 / 或磁存储介质)、对象、可执行件、运行的线程、程序, 和 / 或计算机。作为说明, 在服务器上运行的应用和该服务器两者都可以是组件。一个或多个组件可以驻留在进程和 / 或执行的线程内, 且组件可以位于一台计算机上和 / 或分布在两台或更多的计算机之间。词语“示例性”此处可用于表示用作示例、实例或说明。在此被描述为“示例性”的任何方面或设计并不一定要被解释为相比其他方面或设计更优选或有利。

[0079] 现在参考图 11, 示出了可用于执行根据所公开的体系结构的网格和可视化组合的计算系统 1100 的框图。为了提供用于其各方面的附加上下文, 图 11 及以下讨论旨在提供对其中可实现各方面的合适的计算系统 1100 的简要概括描述。尽管以上描述是在可在一个或多个计算机上运行的计算机可执行指令的一般上下文中进行的, 但是本领域的技术人员将认识到, 新颖实施例也可结合其他程序模块和 / 或作为硬件和软件的组合来实现。

[0080] 用于实现各方面的计算系统 1100 包括计算机 1102, 其具有处理单元 1104、系统存储器 1106、以及系统总线 1108。处理单元 1104 可以是各种市场上可买到的处理器中的任一种, 包括单处理器、多处理器、单核单元以及多核单元。此外, 本领域的技术人员可以理解, 各新颖方法可用其他计算机系统配置来实施, 包括小型机、大型计算机、以及个人计算机 (例如、台式、膝上型等)、手持式计算设备、基于微处理器的或可编程消费电子产品等, 其每一个都可在操作耦合到一个或多个相关联的设备。

[0081] 系统存储器 1106 可包括易失性 (VOL) 存储器 1110 (例如, 随机存取存储器 (RAM)) 和非易失性存储器 (NON-VOL) 1112 (如 ROM、EPROM、EEPROM 等)。基本输入 / 输出系统 (BIOS) 可被存储在非易失性存储器 1112 中, 并且包括诸如在启动期间便于在计算机 1102 内的组件之间传递数据和信号的基本例程。易失性存储器 1110 还可包括诸如静态 RAM 等高速 RAM 来用于高速缓存数据。

[0082] 系统总线 1108 提供了用于包括, 但不限于存储器子系统 1106 的系统组件到处理单元 1104 的接口。系统总线 1108 可以是若干种总线结构中的任一种, 这些总线结构还可使用各类可购买到的总线体系结构中的任一种互连到存储器总线 (带有或没有存储器控制器) 以及外围总线 (例如, PCI、PCIe、AGP、LPC 等)。

[0083] 计算机 1102 还包括存储子系统 1114 以及用于将存储子系统 1114 接口到系统总线 1108 和其他所需计算机组件的存储接口 1116。存储子系统 1114 可包括例如硬盘驱动器 (HDD)、磁软盘驱动器 (FDD) 和 / 或光盘存储驱动器 (例如, CD-ROM 驱动器、DVD 驱动器) 中的一种或多种。存储接口 1116 可包括诸如, 例如 EIDE、ATA、SATA 和 IEEE 1394 等接口技术。

[0084] 一个或多个程序和数据可被存储在存储器子系统 1106、可移动存储器子系统 1118 (例如, 闪存驱动器形状因子技术) 和 / 或存储子系统 1114, 包括操作系统 1120、一个或多个应用程序 1122、其他程序模块 1124 以及程序数据 1126。一般而言, 程序模块包括执行特定任务或实现特定抽象数据类型的例程、方法、数据结构、其他软件组件等等。一个或多个应用程序 1122、其他程序模块 112 以及程序数据 1126 可包括例如图 1 的系统 100、图 2 的系统 200、图 3 的 UI 300、图 4 的状态图 400、图 5 的流程图 500、图 6 的概括数据流程图 600、图 7 的更详细数据流程图 700、以及图 8-10 中表示的方法。

[0085] 操作系统 1120、应用程序 1122、模块 1124 和 / 或数据 1126 的全部或部分也可被高速缓存在诸如易失性存储器 1110 等存储器中。应该明白, 所公开的体系结构可以用各种市场上可购得的操作系统或操作系统的组合 (例如, 作为虚拟机) 来实施。

[0086] 存储子系统 1114 和存储器子系统 (1106 和 1118) 用作用于数据、数据结构、计算机可执行指令等的易失性和非易失性存储的计算机可读介质。计算机可读介质可以是可由计算机 1102 访问的任何可用介质, 且包括易失性和非易失性介质、可移动和不可移动介质。对于计算机 1102, 介质容纳适当的数字格式的数据的存储。本领域的技术人员应当理

解,可使用其他类型的计算机可读介质,如 zip 驱动器、磁带、闪存卡、磁带盒等来存储用于执行所公开的体系结构的新颖方法的计算机可执行指令。

[0087] 用户可以使用诸如键盘和鼠标等外部用户输入设备 1128 来与计算机 1102、程序和数据交互。其他外部用户输入设备 1128 可包括话筒、IR(红外)遥控器、操纵杆、游戏手柄、照相机识别系统、指示笔、触摸屏、姿势系统(例如,眼移动、头移动等)和/或类似物。在计算机 1102 是例如便携式计算机的情况下,用户可以使用诸如触摸垫、话筒、键盘等板载用户输入设备 1130 来与计算机 1102、程序和数据交互。这些和其他输入设备通过输入/输出(I/O)设备接口 1132 经由系统总线 1104 连接到处理单元 1108,但也可通过其他接口连接,如并行端口、IEEE 1394 串行端口、游戏端口、USB 端口、IR 接口等。I/O 设备接口 1132 也便于输出外围设备 1134 的使用,如打印机、音频设别、摄像设备等,如声卡和/或板载音频处理能力。

[0088] 一个或多个图形接口 1136(通常也称为图形处理单元(GPU))提供计算机 1102 和外部显示器 1138(例如,LCD、等离子)和/或板载显示器 1140(例如,对于便携式计算机)之间的图形和视频信号。图形接口 1136 也可作为计算机系统板的一部分来制造。

[0089] 计算机 1102 可以使用经由有线/无线通信子系统 1142 到一个或多个网络和/或其他计算机的逻辑连接在联网环境(例如,IP)中操作。其他计算机可包括工作站、服务器、路由器、个人计算机、基于微处理器的娱乐设备、对等设备或其他常见的网络节点,并且通常包括以上相对于计算机 1102 描述的许多或所有元件。逻辑连接可包括到局域网(LAN)、广域网(WAN)、热点等的有线/无线连接。LAN 和 WAN 联网环境常见于办公室和公司,并且方便了诸如内联网等企业范围计算机网络,所有这些都可连接到例如因特网等全球通信网络。

[0090] 当在联网环境中使用时,计算机 1102 经由有线/无线通信子系统 1142(例如,网络接口适配器、板载收发机子系统等)连接到网络来与有线/无线网络、有线/无线打印机、有线/无线输入设备 1144 等通信。计算机 1102 可包括用于通过网络建立通信的调制解调器或其他装置。在联网环境中,相对于计算机 1102 的程序和数据可被存储在远程存储器/存储设备中,如与分布式系统相关联。应该理解,所示网络连接是示例性的,并且可以使用在计算机之间建立通信链路的其他手段。

[0091] 计算机 1102 可用于使用诸如 IEEE 802.xx 标准家族等无线电技术来与有线/无线设备或实体通信,例如在操作上安置在与例如打印机、扫描仪、台式和/或便携式计算机、个人数字助理(PDA)、通信卫星、任何一件与无线可检测标签相关联的设备或位置(例如,电话亭、报亭、休息室)以及电话的无线通信(例如,IEEE 802.11 空中调制技术)中的无线设备。这至少包括用于热点的 Wi-Fi(即无线保真)、WiMax 和蓝牙™无线技术。由此,通信可以是如对于常规网络那样的预定义结构,或者仅仅是至少两个设备之间的自组织(ad hoc)通信。Wi-Fi 网络使用称为 IEEE 802.11x(a,b,g 等等)的无线电技术来提供安全、可靠、快速的无线连接。Wi-Fi 网络可用于将计算机彼此连接、连接到因特网以及连接到有线网络(使用 IEEE 802.3 相关介质和功能)。

[0092] 所示各方面也可以在其中某些任务由通过通信网络链接的远程处理设备来执行的分布式计算环境中实施。在分布式计算环境中,程序模块可以位于本地和/或远程存储和/或存储器系统中。

[0093] 现在参考图 12, 示出用于基于 web 的网络 / 可视化呈现的计算环境 1200 的示意性框图。环境 1200 包括一个或多个客户机 1202。客户机 1202 可以是硬件和 / 或软件 (例如, 线程、进程、计算设备)。例如, 客户机 1202 可容纳 cookie 和 / 或相关联的上下文信息。

[0094] 环境 1200 还包括一个或多个服务器 1204。服务器 1204 也可以是硬件和 / 或软件 (例如, 线程、进程、计算设备)。服务器 1204 可以例如通过使用本体系结构来容纳线程以执行变换。客户机 1202 和服务器 1204 之间的一种可能的通信可以是以适用于在两个或更多计算机进程之间传输的数据包的形式。数据包可以包括, 例如, cookie 和 / 或相关联的上下文信息。环境 1200 包括可以用来使客户机 1206 和服务器 1202 之间通信更容易的通信框架 1204 (例如, 诸如因特网等全球通信网络)。

[0095] 通信可经由有线 (包括光纤) 和 / 或无线技术来促进。客户机 1202 可操作地连接到一个或多个客户机数据存储 1208, 可以使用这些客户机数据存储来存储客户机 1202 本地的信息 (例如, cookie 和 / 或相关联的上下文信息)。同样地, 服务器 1204 可在操作上连接到可以用来存储对服务器 1210 本地的信息的一个或多个服务器数据存储 1204。

[0096] 上面描述的包括所公开的体系结构的各示例。当然, 描述每一个可以想到的组件和 / 或方法的组合是不可能的, 但本领域内的普通技术人员应该认识到, 许多其他组合和排列都是可能的。因此, 该新颖体系结构旨在涵盖所有这些落入所附权利要求书的精神和范围内的更改、修改和变化。此外, 就在说明书或权利要求书中使用术语“包括”而言, 这一术语旨在以与术语“包含”在被用作权利要求书中的过渡词时所解释的相似的方式为包含性的。

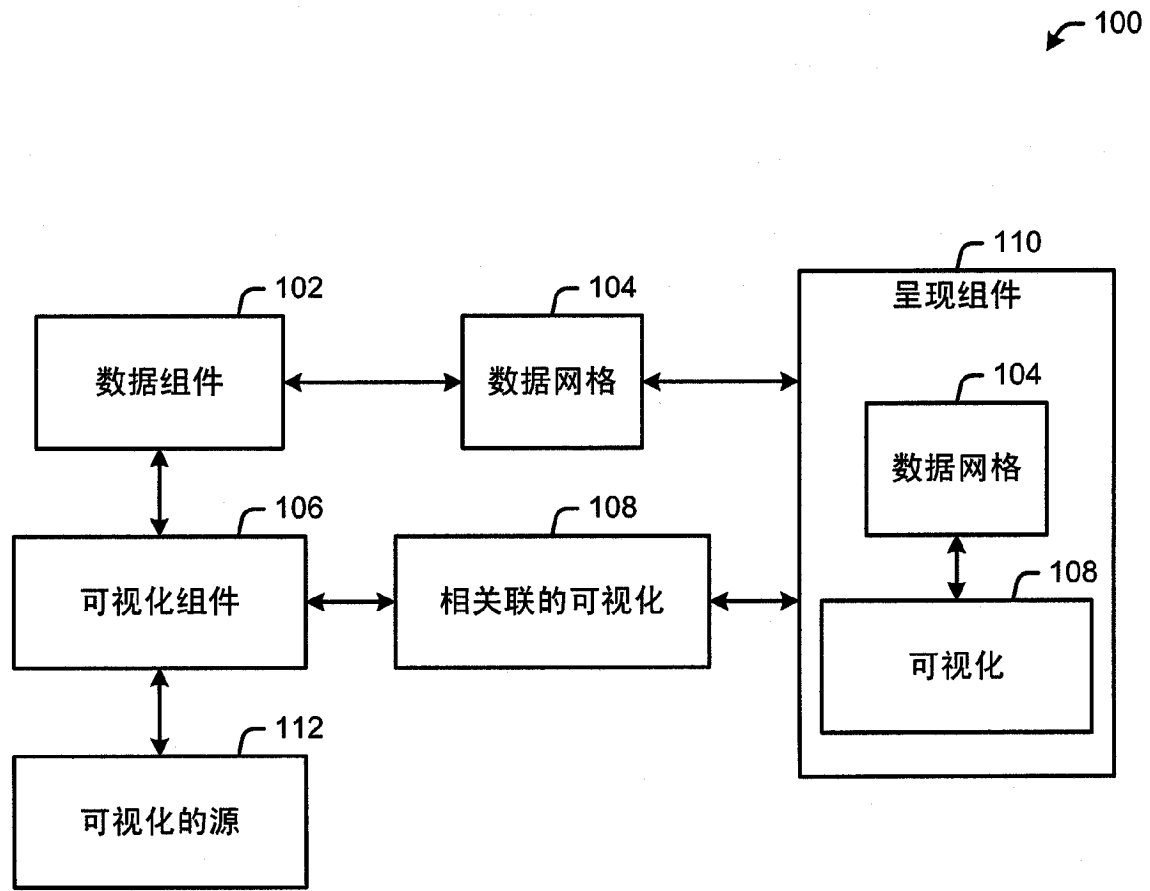


图 1

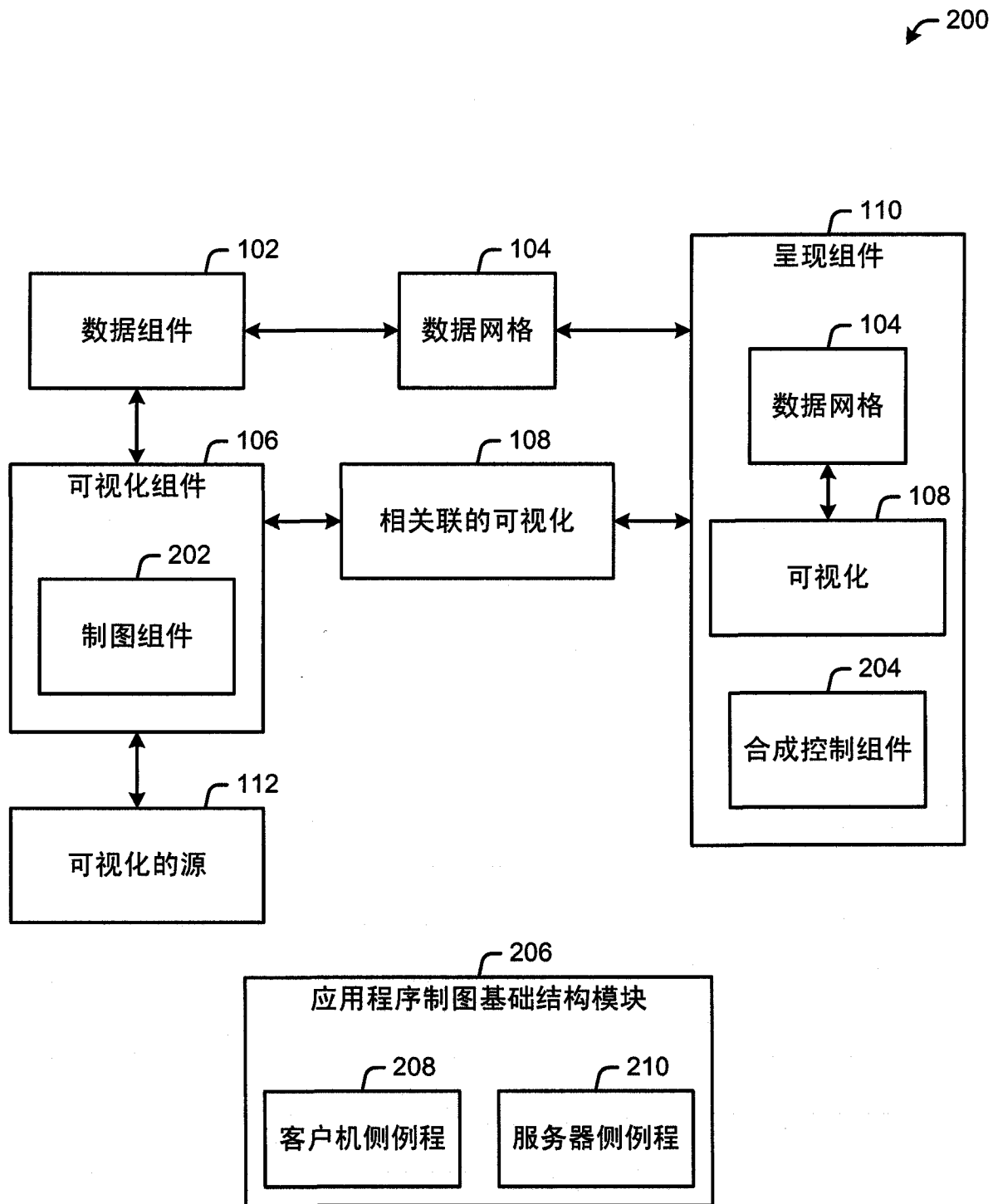


图 2

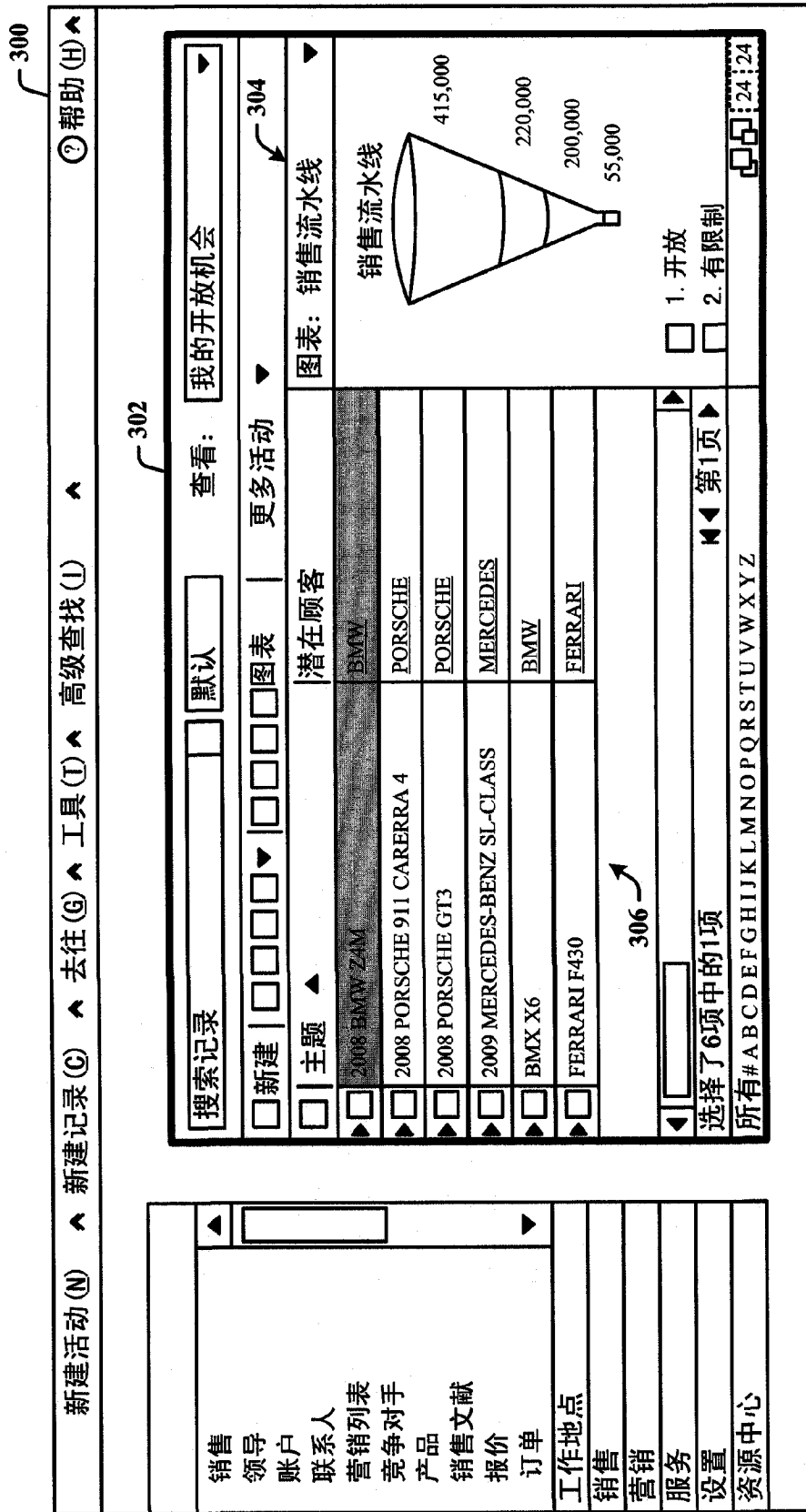


图 3

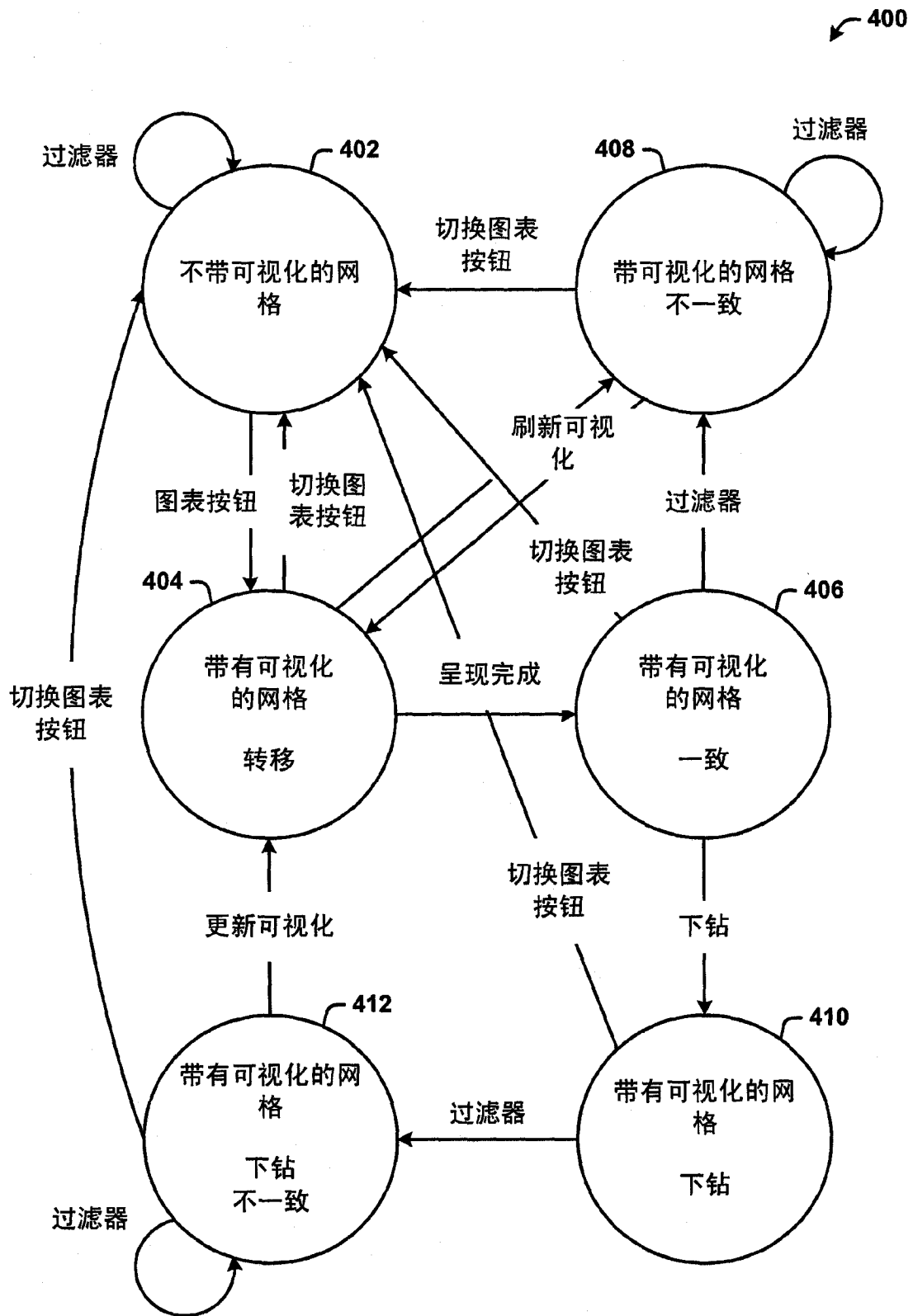


图 4

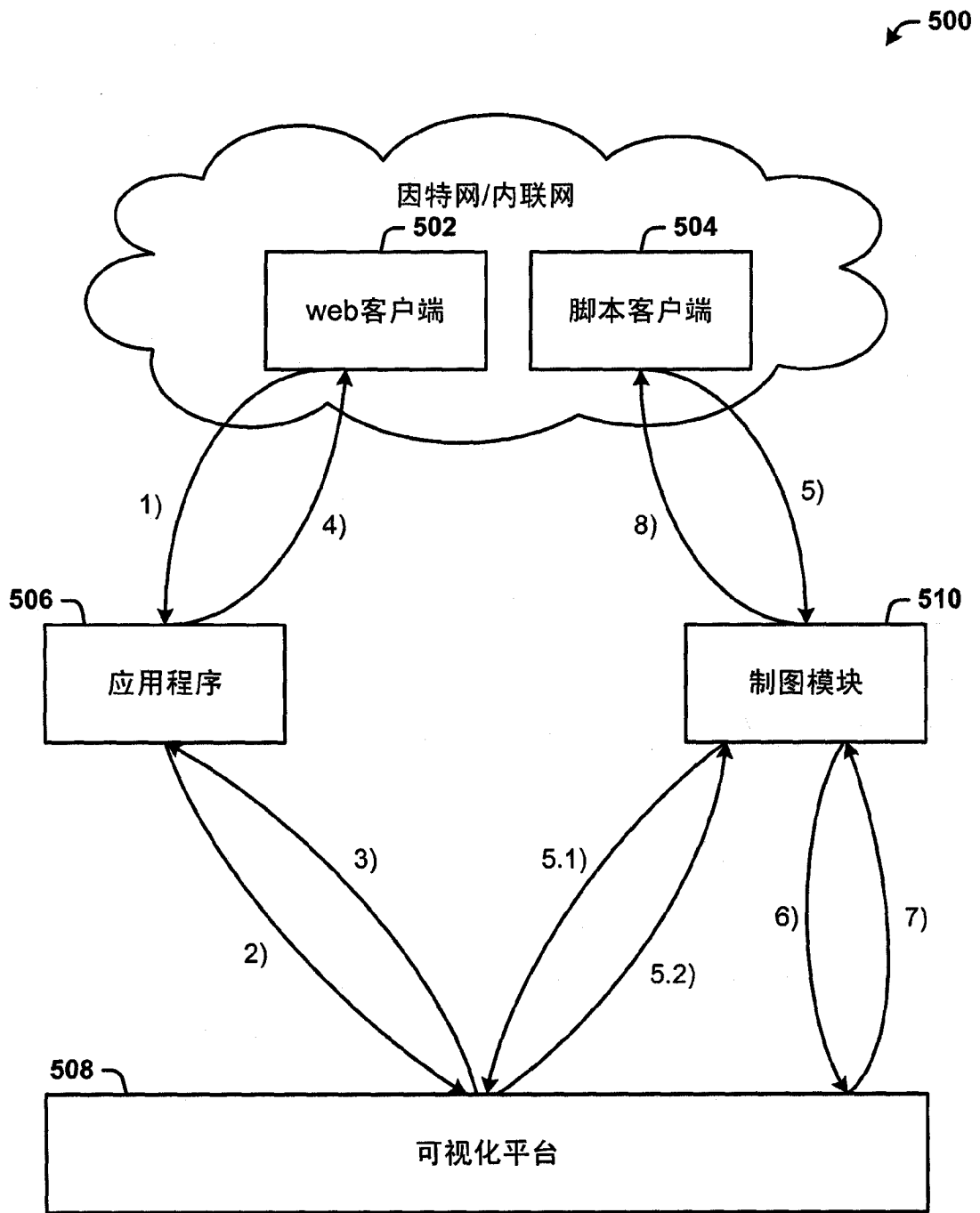


图 5

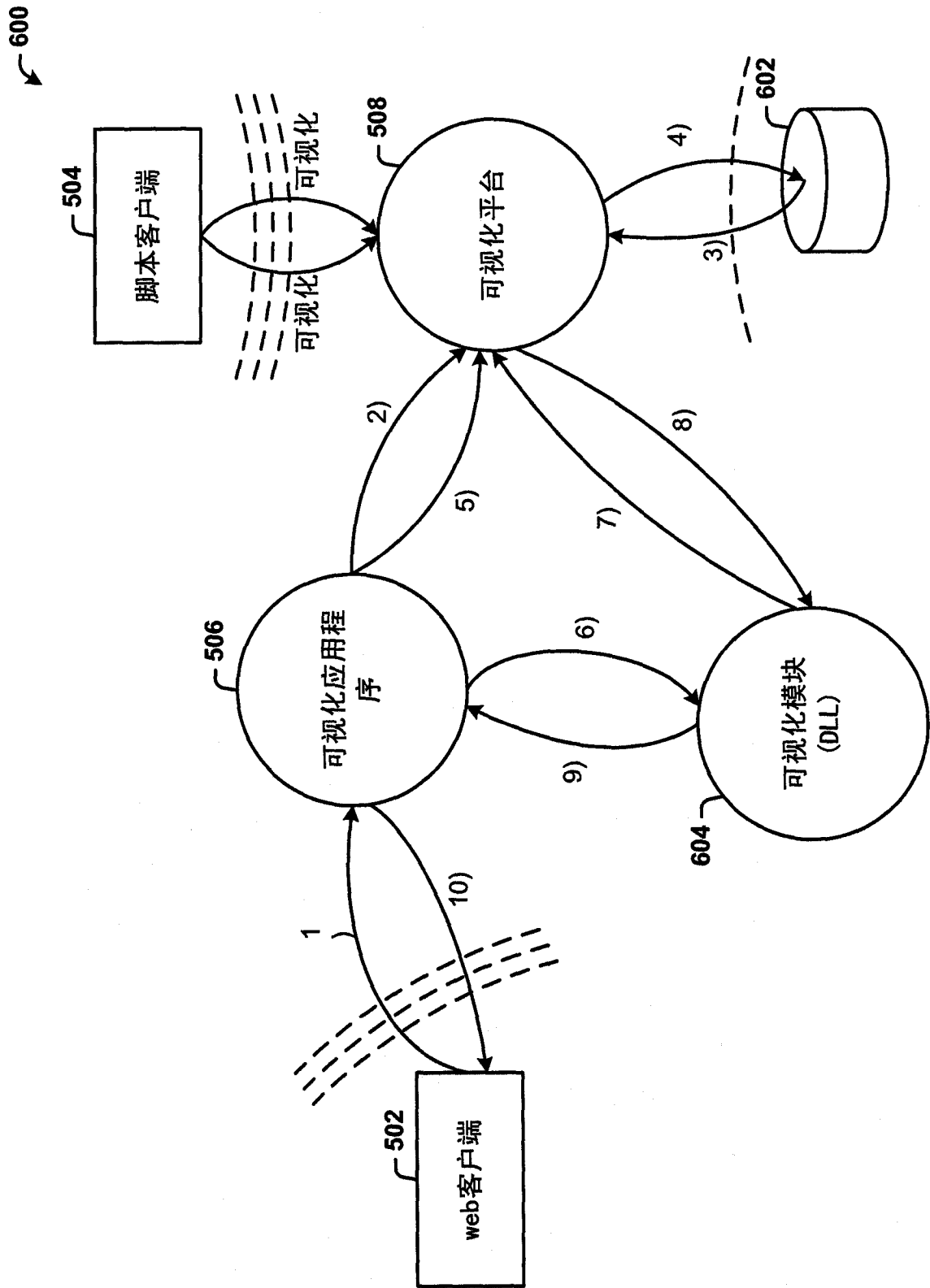


图 6

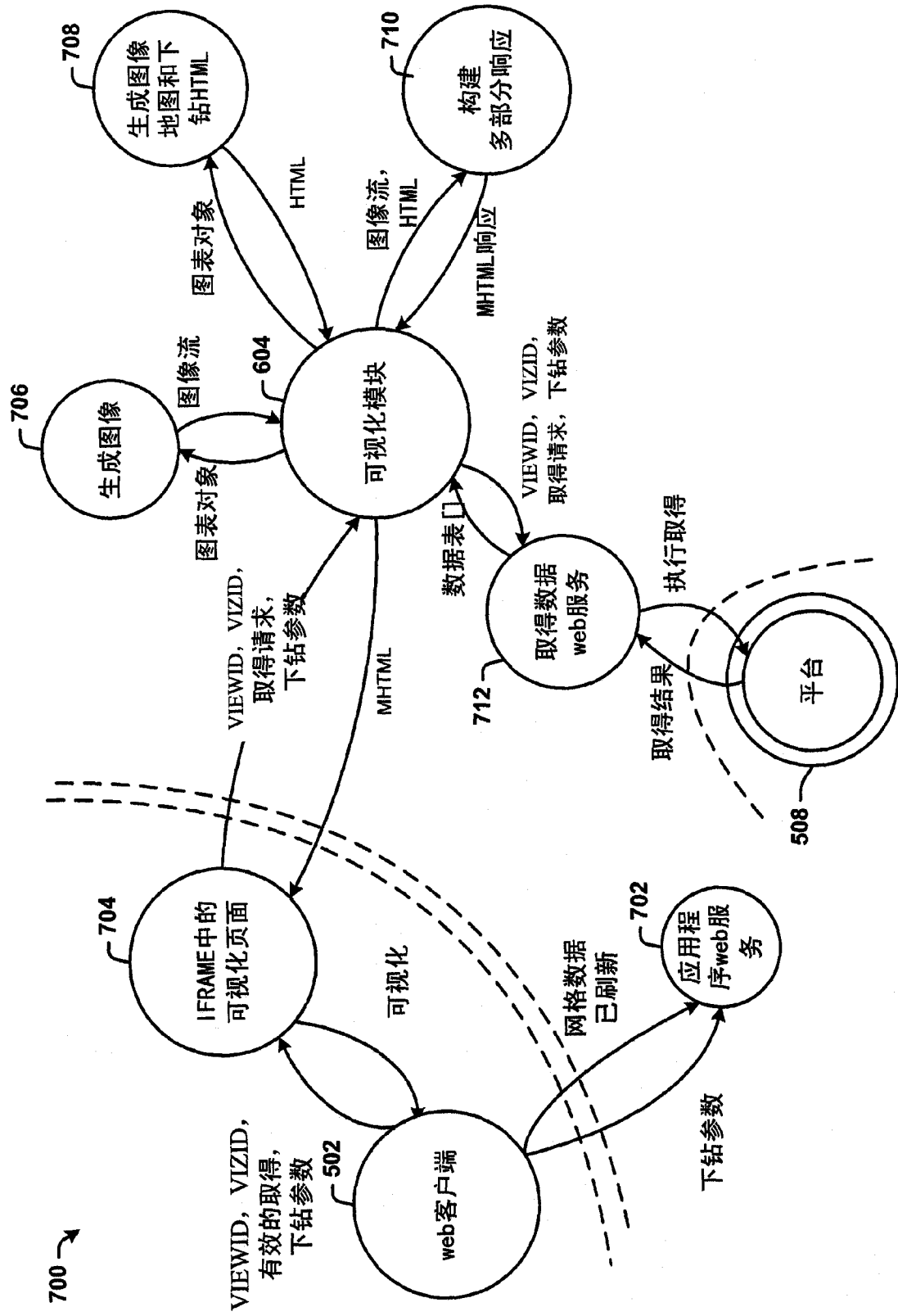


图 7

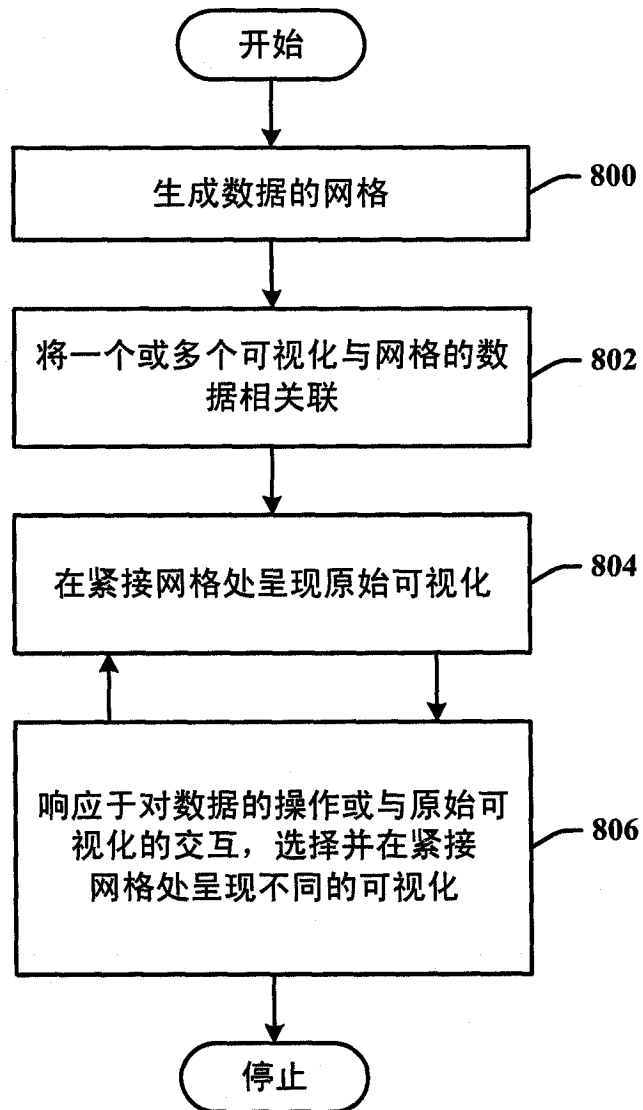


图 8

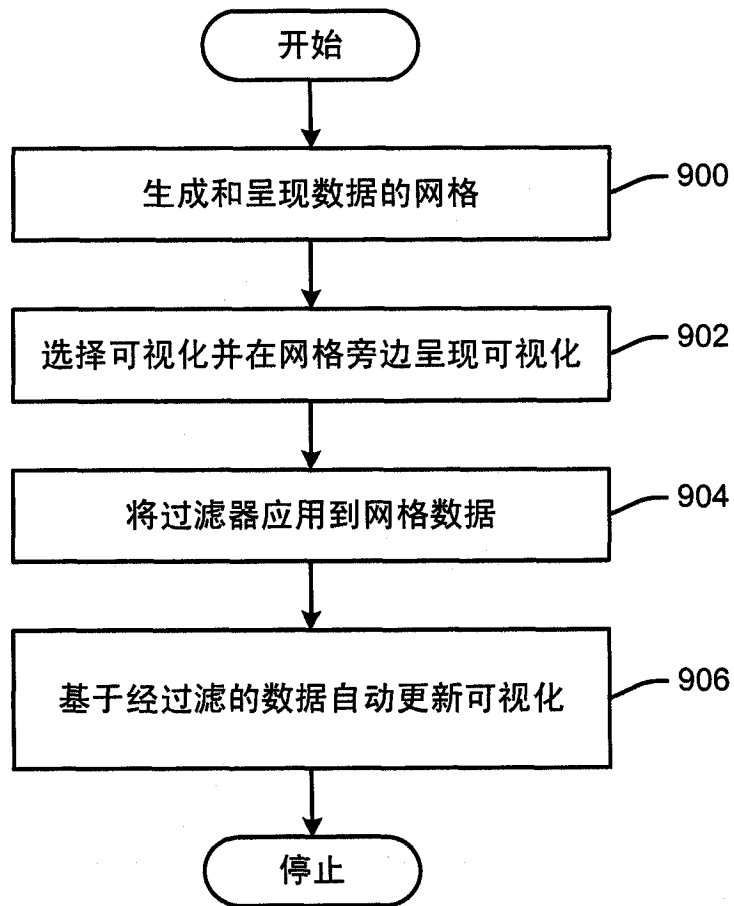


图 9

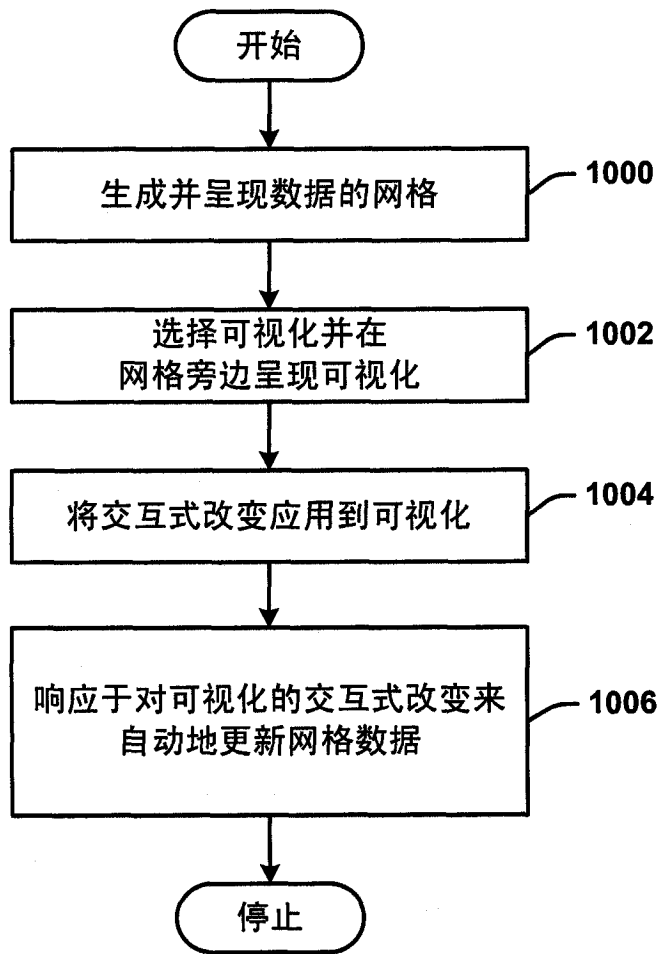


图 10

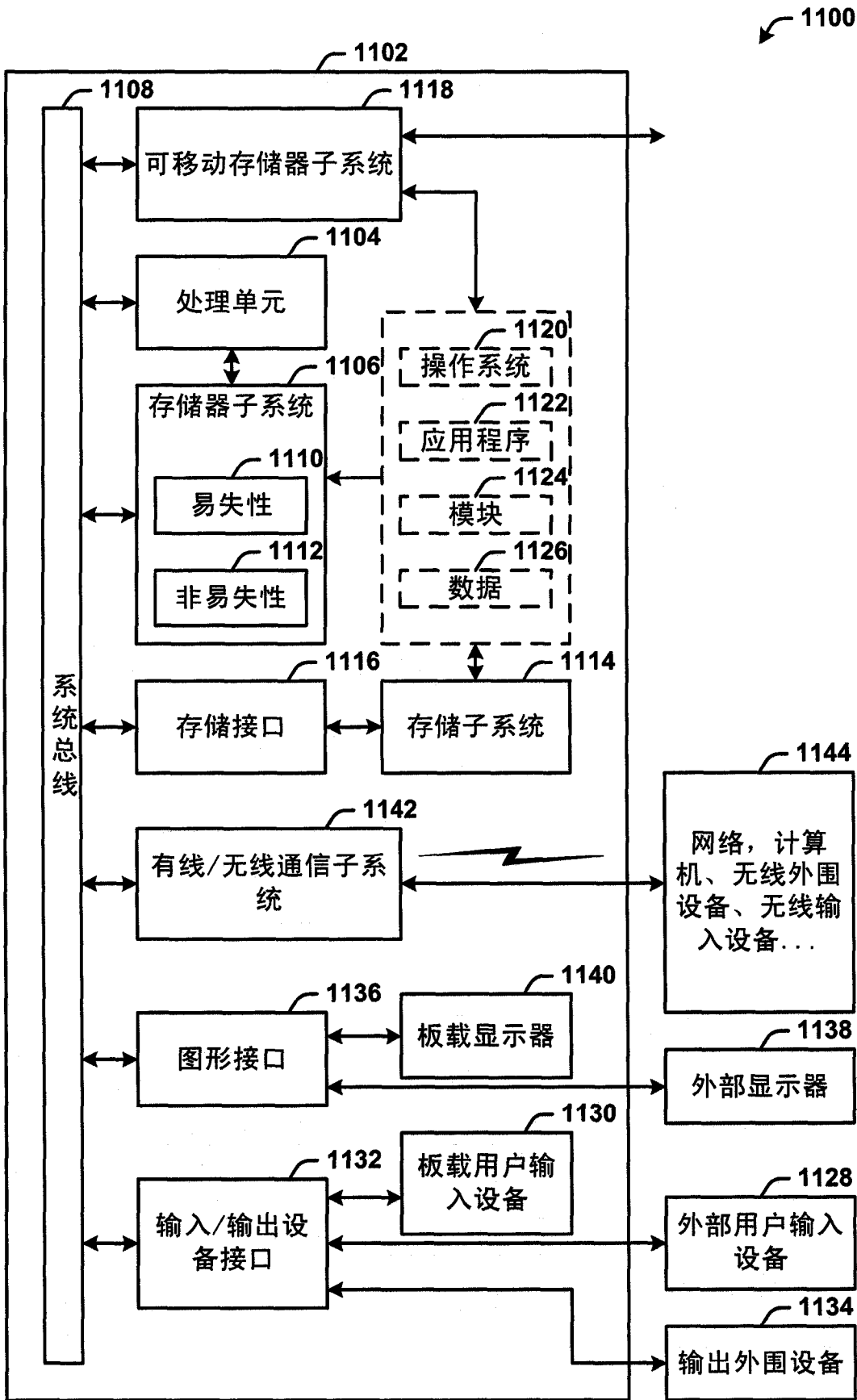


图 11

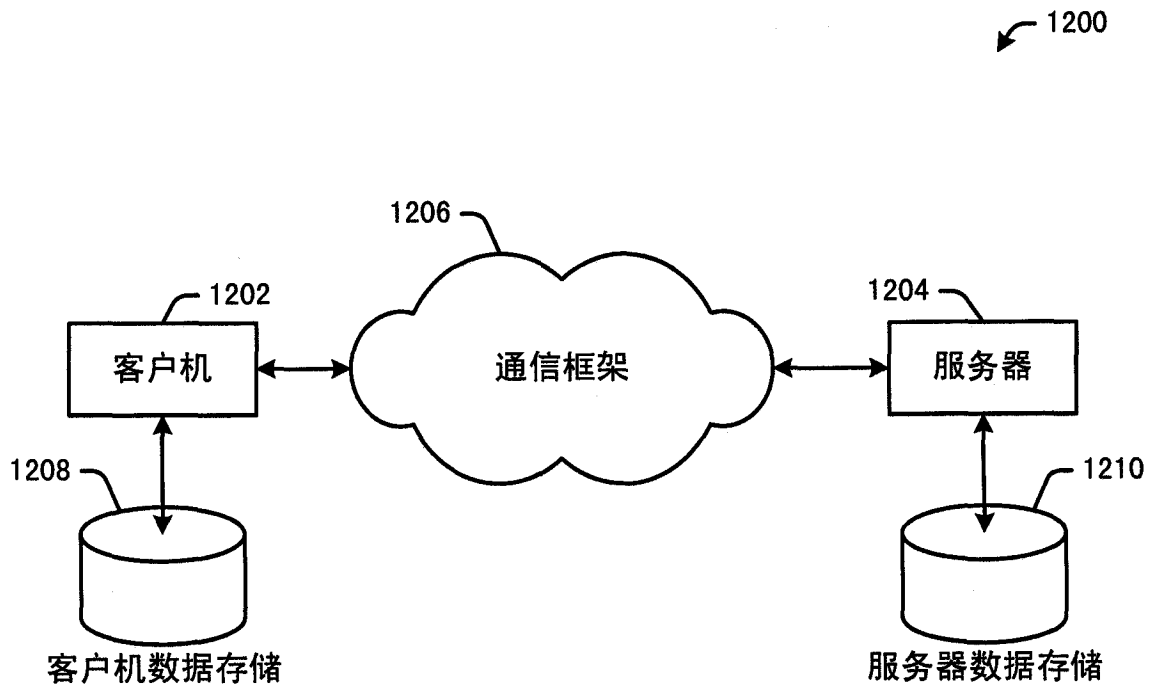


图 12