

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 13/00 (2006.01)

H04L 29/06 (2006.01)



[12] 发明专利说明书

专利号 ZL 200510131674.6

[45] 授权公告日 2009 年 9 月 23 日

[11] 授权公告号 CN 100543707C

[22] 申请日 2005.12.14

[21] 申请号 200510131674.6

[30] 优先权

[32] 2005.1.14 [33] US [31] 11/036,893

[73] 专利权人 微软公司

地址 美国华盛顿州

[72] 发明人 C·钟 J·C·邓恩 J·奇克

M·V·帕特瓦德翰

V·M·玛姆塔尼

[56] 参考文献

CN1549088A 2004.11.24

CN1266515A 2000.9.13

US2003/0142661A1 2003.7.31

US2002/0128065A1 2002.9.12

审查员 王 亮

[74] 专利代理机构 上海专利商标事务所有限公司

代理人 张政权

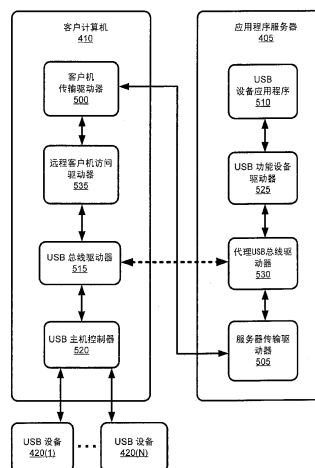
权利要求书 4 页 说明书 15 页 附图 8 页

[54] 发明名称

应用程序服务器环境中的 USB 设备

[57] 摘要

在客户计算机处连接一个或多个通用串行总线 (USB) 设备。由客户计算机识别 USB 设备，并作出关于应用程序服务器计算机是否为该 USB 设备中的一个或多个提供功能的判断。如果功能是由应用程序服务器提供的，那么提供自该 USB 设备至该服务器计算机的专用通信路径。



1. 一种为远程客户机执行应用程序的应用程序服务器，所述应用程序与客户机 USB 设备交互，所述应用程序服务器含有服务器驱动器栈，所述栈包括：

USB 功能驱动器，它具有供所述应用程序用来与所述客户机 USB 设备交互的接口；

代理 USB 总线驱动器，它被配置成与所述远程客户机通信；

其中，所述 USB 功能驱动器被配置成与所述代理 USB 总线驱动器连接，以与所述客户机 USB 设备交互。

2. 如权利要求 1 所述的应用程序服务器，其特征在于，所述代理 USB 功能驱动器被配置成与所述远程客户机处的实际 USB 总线驱动器通信，以与所述客户机 USB 设备交互。

3. 如权利要求 1 所述的应用程序服务器，其特征在于，所述服务器驱动器栈还包括被配置成提供所述代理 USB 总线驱动器和所述远程客户机之间的通信的服务器传输驱动器。

4. 一种客户机服务系统，包括：

如权利要求 1 所述的应用程序服务器；

远程客户机，其中，所述远程客户机含有被配置成与所述客户机 USB 设备交互的客户机 USB 总线驱动器；

其中，所述代理 USB 总线驱动器与所述客户机 USB 总线驱动器通信，以与所述客户机 USB 设备交互。

5. 一种客户机服务系统，包括：

如权利要求 1 所述的应用程序服务器，所述服务器驱动器栈还包括被配置成提供所述代理 USB 总线驱动器与所述远程客户机之间的通信的服务器传输驱动器；

远程客户机，其中，所述远程客户机含有客户机驱动器栈，所述客户机驱动器栈包括：

客户机 USB 总线驱动器，它被配置成与所述客户机 USB 设备交互；

以及

客户机传输驱动器，它被配置成提供所述客户机 USB 总线驱动器与
所述应用程序服务器之间的通信。

6. 一种客户机服务系统，包括：

如权利要求 1 所述的应用程序服务器，所述服务器驱动器栈还包括被配置
成提供所述代理 USB 总线驱动器与所述远程客户机之间的通信的服务器传输
驱动器；

远程客户机，其中，所述远程客户机含有被配置成与所述客户机 USB 设
备交互的客户机 USB 总线驱动器。

7. 一种客户机/服务器远程服务系统，包括：

如权利要求 1 所述的应用程序服务器；

远程客户机，其中，所述远程客户机含有客户机驱动器栈，所述栈包括：

客户机 USB 总线驱动器，它被配置成与所述客户机 USB 设备交互；
以及

USB 主机控制器驱动器，通过所述 USB 主机控制器驱动器，所述客
户机 USB 总线驱动器与所述客户机 USB 设备交互。

8. 一种为由应用程序服务器执行的第一应用程序提供用户界面的客户机，
包括：

第一 USB 设备；

USB 驱动器栈，包括：

USB 主机控制器驱动器，它与所述第一 USB 设备通信；

USB 总线驱动器，它通过所述 USB 主机控制器与所述 USB 设备交
互，所述 USB 总线驱动器含有被设计成与 USB 功能驱动器通信的接口，
所述 USB 功能驱动器包括供所述应用程序用来与所述第一 USB 设备交互
的接口；以及

客户机传输驱动器，它使用所述 USB 总线驱动器的接口，通过所述
USB 驱动器栈来提供由所述应用程序服务器执行的第一应用程序与所述
第一 USB 设备之间的通信。

9. 如权利要求 8 所述的客户机，其特征在于，所述客户机传输驱动器与
由所述应用程序服务器实现的服务器 USB 功能驱动器通信。

10. 如权利要求 8 所述的客户机，其特征在于，还包括：

第二 USB 设备；

其中，所述 USB 主机控制器驱动器与所述第二 USB 设备通信；

由所述客户机执行的第二应用程序；

客户机 USB 功能驱动器，它使用所述 USB 总线驱动器的接口，通过所述 USB 驱动器栈来提供所述第二应用程序与所述第二 USB 设备之间的通信。

11. 如权利要求 8 所述的客户机，其特征在于，所述客户机传输驱动器与由所述应用程序服务器实现的代理 USB 总线驱动器通信。

12. 如权利要求 8 所述的客户机，其特征在于，还包括：

第二 USB 设备；

其中，所述 USB 主机控制器驱动器与所述第二 USB 设备通信；

由所述客户机执行的第二应用程序；

客户机 USB 功能驱动器，它使用所述 USB 总线驱动器的接口，通过所述 USB 驱动器栈来提供所述第二应用程序与所述第二 USB 设备之间的通信；

其中，所述客户机传输驱动器与由所述应用程序服务器实现的服务器 USB 功能驱动器通信。

13. 一种与位于客户计算机处的第一和第二 USB 设备交互的方法，包括：

为由应用程序服务器执行的主宿的应用程序，在所述客户计算机处提供用户界面；

由所述客户计算机执行本地应用程序；

在所述客户计算机处提供 USB 驱动器栈，所述 USB 驱动器栈包括一连串栈，至少包括 USB 功能驱动器和 USB 总线驱动器，所述 USB 功能驱动器包括供所述应用程序用来与所述第一 USB 设备交互的接口，所述 USB 总线驱动器与所述 USB 设备交互且含有被设计成与 USB 功能驱动器通信的接口；

通过所述 USB 驱动器栈，在所述主宿的应用程序与所述第一 USB 设备之间通信；

通过所述 USB 驱动器栈，在所述本地应用程序与所述第二 USB 设备之间通信。

14. 如权利要求 13 所述的方法, 其特征在于, 所述 USB 驱动器栈还至少包括 USB 主机控制器驱动器。

15. 如权利要求 13 所述的方法, 其特征在于, 还包括在所述 USB 驱动器栈与由所述应用程序服务器实现的 USB 功能驱动器之间通信。

16. 一种与位于客户计算机处的第一 USB 设备和第二 USB 设备交互的设备, 包括:

用于为由应用程序服务器执行的主宿的应用程序, 在所述客户计算机处提供用户界面的装置;

用于由所述客户计算机执行本地应用程序的装置;

用于在所述客户计算机处提供 USB 驱动器栈的装置, 所述 USB 驱动器栈包括一连串栈, 至少包括 USB 功能驱动器和 USB 总线驱动器, 所述 USB 功能驱动器包括供所述应用程序用来与所述第一 USB 设备交互的接口, 所述 USB 总线驱动器与所述 USB 设备交互且含有被设计成与 USB 功能驱动器通信的接口;

用于通过所述 USB 驱动器栈, 在所述主宿的应用程序与所述第一 USB 设备之间通信的装置;

用于通过所述 USB 驱动器栈, 在所述本地应用程序与所述第二 USB 设备之间通信的装置。

17. 如权利要求 16 所述的设备, 其特征在于, 所述 USB 驱动器栈还至少包括 USB 主机控制器驱动器。

18. 如权利要求 16 所述的设备, 其特征在于, 还包括在所述 USB 驱动器栈与由所述应用程序服务器实现的 USB 功能驱动器之间通信的装置。

应用程序服务器环境中的 USB 设备

技术领域

本发明涉及将对通用串行总线（USB）设备的访问和控制定向至诸如服务器计算机等计算机。

背景技术

远程客户机访问平台和系统，诸如由微软公司提供的 Terminal Service™，允许计算机远程地访问主宿于并驻留在应用程序服务器中的应用程序。在远程客户机访问系统中，客户计算机一般依赖于服务器计算机通过驻留在服务器计算机上的应用程序来提供计算功能。应用程序的示例包括文字处理、多媒体和数据管理程序。

应用程序服务器计算机可以被称为主机计算机或终端服务器。客户计算机可以被称为远程终端、远程客户机或瘦客户机。客户计算机要被用于用户界面：与用户和设备 I/O 交互。客户计算机上的软件一般是通用的，即不是应用程序专用的，它一般包括操作系统和通用软件——包括支持远程客户机访问环境的软件。服务器计算机上的软件一般包括提供特定功能的专用应用程序软件，诸如数据库访问、文字处理、制图以及多种其它类型的应用程序。在客户计算机和应用程序服务器之间传递的数据通常包括关于用户界面的命令和数据，诸如图形数据、击键、鼠标移动等，以及关于位于客户计算机上的硬件设备的命令和数据。

应用程序服务器和客户机一般使用诸如由微软公司定义的远程桌面协议（RDP）等预定义的通信协议彼此通信或传递信息。也涉及诸如 TCP/IP 等较低层网络协议。

远程客户机访问系统的好处是，客户计算机可以是相对较低能力的，因为大多数功能和计算是发生在服务器计算机上的。尽管应用程序服务器通常比典型的台式计算机更昂贵，但是一台应用程序服务器计算机可以服务众多较不昂贵的客户机。

某些系统中的另一优点是，数据可以驻留在服务器计算机的物理位置上，且可以在该位置上由应用程序来处理，而不必通过相对缓慢的通信链路传送给客户计

算机——仅用户界面是在客户计算机的物理位置上实现的。

客户计算机通常具有连接外围设备的 USB（通用串行总线）端口。这样的设备通常涉及用户界面，正如 USB 键盘的情形中。在众多情形中，在服务器计算机上执行的应用程序需要访问这样的 USB 设备并与之交互。USB 设备的其它示例包括，数码照相机、文档扫描仪、外置磁盘驱动器以及介质读出器。由应用程序服务器主存的各种应用程序可能需要与这些硬件设备交互。

在典型的基于 Windows®的桌面环境中，本地应用程序通过被称为驱动器栈的一连串驱动器与 USB 设备通信。为组织和可重用性起见，在该栈的组件之间划分不同的责任。例如，某些驱动器对于所有或某些通用种类 USB 设备是通用的，且可以被使用或重复使用来与众多不同的设备通信。其它的驱动器实现对某些设备专用的功能，且通常被设计来专门伴随某些硬件。

图 1 示出了在作为独立计算机而不是作为应用程序服务器的客户机的计算机内实现的典型 USB 体系结构。图 1 显示了硬件和软件组件之间的逻辑通信。

图 1 的系统包括计算机 100 和 USB 设备 105。USB 设备 105 通过物理 USB 端口（未示出）被连接至计算机 100。应用程序 110 在计算机上执行，且通过驱动器栈 115 与 USB 设备 105 交互。驱动器栈 115 在该示例中拥有三个 USB 驱动器。

在驱动器栈 115 的最低层中，USB 主机控制器驱动器 120 直接与计算机中的 USB 硬件（未示出）通信，且通过该硬件与 USB 设备 105 通信。在其上方，低层 USB 总线驱动器 125（也被称为集线器或集线器驱动器）与 USB 主机控制器 120 通信，并管理 USB 设备的电源、枚举以及各种 USB 事务。这些设备都是 Windows®操作系统的一部分，且对所有 USB 设备通用；这些驱动器不必根据连接至计算机 100 的 USB 设备的类型来替换或修改。

驱动器栈 115 也包括 USB 功能驱动器 130。USB 功能驱动器 130 为特定的一个或一类设备所定制。结果，取决于实际使用的 USB 设备，装载不同的功能驱动器。USB 功能驱动器也被称为 USB 设备功能驱动器、类驱动器或自定义驱动器。

尽管图 1 仅示出单个应用程序 110，但计算机 110 一般含有多个应用程序，其中的任何一个或多个可以被配置成通过所示的单个驱动器栈 115 与 USB 设备 105 交互。应用程序 110 可以是文字处理程序、游戏程序、或是各种其它类型的程序中的任何一种。

图 2 示出了类似的示例，其中计算机 200 含有多个应用程序 205 和 210，以及多个 USB 设备 215、220 和 225。在该示例中，应用程序 205 与 USB 设备 215 交

互，而应用程序 210 与 USB 设备 220 和 225 交互。

如前述示例中，通信是由 USB 驱动器栈 230 实现的。USB 驱动器栈 230 包括与 USB 设备 215、220 和 225 中的每一个通信的 USB 主机控制器 235 和 USB 总线驱动器 240。USB 驱动器栈 230 还包括对应于 USB 设备 215、220、225 中的每一个的多个 USB 功能驱动器 245、250 和 255。如已述，取决于 USB 设备的特定类型或种类，选取并装载这些 USB 功能驱动器 245、250 和 255 中的每一个。

如示，应用程序可以与多于单个的 USB 设备交互。如果安装了多个 USB 设备，那么它们一般使用同一主机控制器驱动器和总线驱动器，尽管单个计算机含有一个以上 USB 驱动器栈是可能的。如果使用不同种类的 USB 端口，那么可以装载不同的端口和“小型端口”驱动器；然而，一般而言，总的 USB 驱动器栈保持不变。

图 3 示出了在现有技术的远程客户机/服务器体系结构中的 USB 设备的使用。在该示例中，客户计算机 300 具有连接至其 USB 端口的一个或多个 USB 设备 305(1)-305(N)。它也含有操作系统和对这样的 USB 设备的其它底层支持。具体地，远程客户计算机 300 包括与 USB 设备 305 连接的 USB 主机控制器 310。

客户计算机 300 响应于其自身与服务器计算机 315 之间传送的命令和数据来执行用户交互。一个或多个应用程序 320 驻留在服务器 315 上，并在其上执行。为了支持位于远处的 USB 设备 305，服务器计算机 315 的操作系统或其它支持软件包括 USB 支持。具体地，服务器计算机 315 包括 USB 功能设备驱动器 325(1)-325(N)、USB 总线驱动器 330 以及 USB 主机控制器 335。

USB 功能设备驱动器 325 执行与上述 USB 功能设备驱动程序 130 类似的功能。USB 功能设备驱动器 325 与 USB 总线驱动器 330 通信。同样，USB 总线驱动器 330 执行与上述的 USB 总线驱动器 125 类似的功能。服务器计算机 315 还包括连接至客户计算机 300 的 USB 主机控制器 310 或与之通信的 USB 主机控制器 335。USB 主机控制器 310 与 USB 主机控制器 335 之间的通信可以通过直接物理连接，或包括连接双方的中间网络。USB 主机控制器 335 通过与 USB 主机控制器 310 的通信，与 USB 设备 305 通信并对其访问。在该示例中，客户计算机 300 完全依赖于服务器计算机 315 通过应用程序 320 来为 USB 设备 305 提供所有所需的功能。

对比图 1 中所示的独立计算机的操作，在图 2 的客户机服务器方案中，应用程序 320 形式的所有功能在服务器 315 处提供。换言之，客户计算机 300 依赖于服务器计算机 315 来提供 USB 设备 305 的所有功能。在该方案中，客户计算机 300

不提供或不被允许来向 USB 设备 305 中的任何一个提供任何功能。

发明内容

在客户计算机处连接一个或多个通用串行总线（USB）设备。向服务器计算机提供对特定 USB 设备的访问和控制，且在客户计算机处提供从该特定 USB 设备至该服务器计算机的专用通信路径。接收来自服务器计算机的命令，并从该特定 USB 设备向该服务器计算机发送数据。

服务器计算机识别连接至客户计算机的特定 USB 设备，且在服务器计算机处提供对该特定 USB 设备的专用通信路径，服务器计算机向 USB 设备发送命令，并通过服务器计算机处的通信路径从 USB 设备接收数据。

附图说明

参考附图描述了详细描述。附图中，参考标号的最左端的数字标识其中该参考标号首次出现的附图。不同附图中的同一参考标号的使用指示类似或完全相同的项。

图 1 是在计算机上实现的现有技术 USB 体系结构的框图。

图 2 是使用多个应用程序和多个 USB 设备的计算机上实现的现有技术 USB 体系结构的框图。

图 3 是对一个或多个 USB 设备的远程访问的现有技术体系结构的框图。

图 4 是其中一个或多个 USB 设备或者由客户计算机或者由应用程序服务器计算机控制的远程客户机访问系统的图示。

图 5 是其中 USB 设备的功能在应用程序服务器计算机处提供的客户计算机和应用程序服务器计算机的框图。

图 6 是访问和控制 USB 设备的客户机计算机和应用程序服务器计算机的框图。

图 7 是示出将客户计算机 USB 设备的控制定向或重定向至客户计算机或应用程序服务器计算机的流程图。

图 8 是客户计算机或应用程序服务器计算机的实现的框图。

具体实施方式

以下揭示描述了将对特定 USB 设备的访问和控制定向或重定向至应用程序服

务器计算机。USB 设备被连接至与该应用程序服务器计算机通信的客户计算机处。

客户机-服务器计算机系统

图 4 示出了远程客户机访问系统 400。系统 400 包括应用程序服务器计算机 405 和一台或多台客户计算机。对在客户计算机处连接的通用串行总线 (USB) 的访问和控制可以被定向或重定向至应用程序服务器计算机 405。具体地, 由应用程序服务器计算机 405 选择性地访问和控制客户计算机处的 USB 设备。系统 400 可以是如微软公司提供或定义的 Terminal Service™系统, 其中多台客户计算机依赖于应用程序服务器计算机 405 的提供功能, 尤其是对一个或多个 USB 设备的访问和控制的全部或某些应用程序。在该示例中, 示出客户计算机 410, 它是其它客户计算机的示例。

网络 415 连接应用程序服务器计算机 405 和客户计算机 410。网络 415 可以用多种方式来实现, 以支持这样的连网环境, 包括基于有线的技术或无线技术。本发明的各方面不限于一种特定的网络体系结构或网络技术。系统 400 是不同体系结构的代表, 这些体系结构包括通过调制解调器的直接拨号、企业 LAN (局域网)、WAN (广域网) 以及因特网。网络 415 将服务器计算机 405 连接至一台或多台客户计算机 (例如, 客户计算机 410)。而且, 在服务器 405 与客户计算机之间的网络 415 的连接可以实现诸如通过互联网协议的传输控制协议 (TCP/IP) 的传输协议。

服务器计算机 405 使用诸如由微软公司提供的 Windows®服务器 2003 操作系统的操作系统来实现。服务器计算机 405 和客户计算机 410 可以实现诸如由微软公司定义的远程数据协议 (RDP) 等通信协议, 来彼此传递数据或信息 (即, 通信)。这样的通信协议, 尤其是 RDP 的使用, 可以在诸如 Terminal Service™系统等的远程客户机访问系统的环境中实现。

客户计算机 410 可以是通用 PC (个人计算机)、膝上型 PC、图形输入板 PC 等, 且实现诸如来自微软公司的 Windows®操作系统等操作系统。客户计算机 410 是主要连接至服务器计算机 305 来访问不是本地存储在客户计算机 410 处的文件或其它信息 (例如, 驻留在应用程序服务器计算机 405 处的应用程序) 的独立计算机。

客户计算机 410 配备支持现有传统以及未来的 USB 标准, 诸如 USB 1.0、1.1 和 2.0 的一个或多个 USB 端口。USB 标准允许端口 (计算机接口) 分裂以及 USB 设备成为 “菊花链”。换言之, 客户计算机 410 上的端口可以被 “分裂” 成多个端口, 从而允许多个设备使用同一端口。USB 端口可以被连接至客户计算机 410 中

的特定 USB 集线器（即，USB 1.1 设备被连接至 USB 1.1 集线器；USB 2.0 设备被连接至 USB 2.0 集线器）。

USB 集线器被连接至诸如 USB 1.1 主机控制器或 USB 2.0 主机控制器等的主机控制器。在某些情形中，主机控制器可以支持多个 USB 标准（例如，支持 USB 1.1 和 2.0 的 USB 主机控制器）。主机控制器被连接至诸如客户计算机 410 的 PCI（外围设备互连）总线等总线。PCI 总线由客户计算机 410 的中央处理单元（CPU）或处理器访问。客户计算机 410 的硬件体系结构将在下文中参考图 8 进一步描述。

客户计算机 410 被连接至一个或多个 USB 设备 420(1)、420(2)和 420(N)。USB 设备 420 包括，但不限于，数码照相机、摄影机、硬盘存储设备、数字媒体记录器、打印机和扫描仪。诸如 USB 设备 420 等 USB 设备由特定的设备类定义。换言之，存在用于数码照相机、摄影机、硬盘存储设备、数字媒体记录器、打印机、扫描仪等的特定设备类。客户计算机 410 和 USB 设备 420 特别地由 USB 连接 425(1)、425(2) 和 425(N)分别连接。USB 连接 425 被连接至上述客户计算机 410 处的一个或多个 USB 端口。

客户机和服务器计算机体系结构

图 5 示出了应用程序服务器计算机 405 和客户计算机 410 处的顶层体系结构，其中 USB 设备 420 的功能是在应用程序服务器计算机 405 处提供的。

客户计算机 410 和应用程序服务器计算机 405 通过它们各自的客户机传输驱动器 500 和服务器传输驱动器 505 彼此通信。如以上在图 1 中描述的，诸如网络 415 等网络可以连接服务器计算机 405 和客户机计算机 410。而且，客户计算机 410、服务器计算机 405 以及网络（例如，网络 415）可以是诸如系统 400 等更大的远程客户机访问系统的一部分。

在该示例中，客户计算机 410 依赖于应用程序服务器计算机 405 来为 USB 设备 420 提供功能。驻留在应用程序服务器 405 处的应用程序 510 特别为 USB 设备 420 中的一个或多个提供功能。这样的 USB 设备中的一个或多个的功能可以由支持各种远程客户机访问系统情形的应用程序服务器计算机 405 提供。

一个示例性的情形是当客户计算机 410 不具有支持 USB 设备的功能的时候（即，没有应用程序来支持）。这是当客户计算机 410 是瘦客户机且依赖于由服务器计算机 405 提供所有功能时的典型情形。

另一情形是当客户计算机 410 含有有限的功能或选择数量的应用程序来支持

USB 设备的时候。然而，驻留在客户计算机 410 上的有限数量的应用程序不包括特别地支持一个或多个 USB 设备 420 的应用程序。服务器计算机 405 包括提供一个或多个 USB 设备 420 所需的功能的特定应用程序 510。对一个或多个 USB 设备 420 的控制可以通过客户计算机 410 体系结构的即插即用（PNP）特征来确定。特别地，USB 总线驱动器 515 识别一个或多个 USB 设备 420，并将一个或多个 USB 设备 420 的存在传递至其它组件（例如，客户计算机的 CPU）、驱动器或软件模块。USB 主机控制器 520 与 USB 总线驱动器通过接口连接，并提供至 USB 设备 420 的物理或逻辑连接。如果由其它组件、驱动器或软件模块确定支持一个或多个 USB 设备 420 的可接受应用程序不驻留在客户计算机 410 上，那么对该一个或多个 USB 设备 420 的控制被定向至应用程序服务器计算机 405。

在其它示例情形中，客户计算机 410 可以包括支持一个或多个 USB 设备 420 的常驻应用程序；然而为众多原因中的一个起见，客户计算机 410 处的用户可以决定将该一个或多个 USB 设备 420 的支持和功能定向至应用程序服务器计算机 405。该用户启动的操作向应用程序服务器计算机 405 提供对该一个或多个 USB 设备 420 的访问和控制。

应用程序 510 与 USB 功能设备驱动器 525 通信。USB 功能设备驱动器 525 是特定 USB 设备的设备类专用的。USB 功能设备驱动器 525 创建表示 USB 设备类的特定功能设备对象（FDO）。该特定 FDO 是表示 USB 设备类的对象，并向应用程序 510 展示或提供接口。作为常驻于 USB 功能设备驱动器 525 上的对象，FDO 或由 FDO 展示的接口是由应用程序 510 特别地识别的。

USB 功能设备驱动器 525 与代理 USB 总线驱动器 530 通信。代理 USB 总线驱动器 530 是设备无关的，并支持多个包括本地 USB 设备以及诸如 USB 设备 420 等远程 USB 设备的 USB 设备。USB 功能设备驱动器 525 创建物理设备对象（PDO），它是表示 USB 设备 420 中的一个并供其专用的数据对象。PDO 数据对象表示 USB 设备 420 中的一个，并供该特定 USB 设备专用。唯一或专用的 PDO 允许服务器计算机 405 通过使用代理 USB 总线驱动器 530 来区分特定的 USB 设备。而且，通过各自的中间接口 500 和 505，代理 USB 总线驱动器 530 与远程客户机访问驱动器 535 通信。远程客户机访问驱动器 535 执行传递源自服务器计算机 405 的命令（即，来自应用程序 510 的命令）的特定功能。代理 USB 总线驱动器 530 逻辑上与 USB 总线驱动器 515 通信。

与应用程序 510 往来的通信可以包括 I/O 请求包（IRP），尤其是被称为 I/O

控制（IOCTL）码的一组专用的 IRP。IRP 是源自应用程序 510 的通信数据，包括对 USB 设备 420 中的一个或多个执行动作的请求，诸如将数据传送（发送）至应用程序 510 的请求。IOCTL 码可以由应用程序 510 使用来与 USB 功能设备驱动器 525 通信。IOCTL 码是由应用程序 510 传递至 USB 设备 420 中的一个或多个来执行特定动作的专用命令或指令。USB 功能设备驱动器 525 从应用程序 510 接收 IOCTL 码，并将该 IOCTL 码转换为传递给 USB 设备 420 中的一个或多个的 IRP。同样，从 USB 设备 420 中的一个或多个发送的 IRP 被转换为可以由应用程序 510 读取或使用的 IOCTL。IOCTL 特别地用于来自应用程序 510 的命令和/或动作（即，请求）与诸如 USB 功能设备驱动器 525 等设备或类专用驱动器的通信。USB 功能设备驱动器 525 确认定义命令或动作的 IOCTL，尤其是 IOCTL 参数。

图 6 示出了应用程序服务器计算机 405 和客户计算机 410 处的示例顶层体系结构，其中功能是由应用程序服务器 405 或者客户计算机 410 中的任一个提供的。USB 设备 420 中的一个或多个可以由客户计算机 410 或应用程序服务器计算机 405 中的任一个来访问和控制。特别地，位于应用程序服务器计算机 405 或客户计算机 410 中的任一个的应用程序向特定 USB 设备 420 提供功能。

在远程客户机访问系统实现中，客户计算机 410 被用来主要提供输入和输出（I/O），诸如由用户通过如键盘或鼠标等输入设备输入的数据。在该示例中，I/O 数据是由 USB 设备 420 提供的。常驻于客户计算机 410 处的应用程序 605 可以用于为 USB 设备 420 中的一个或多个提供功能，且特别地允许接收 I/O 数据，并将其发送至 USB 设备 420 中的一个或多个。具体地，由 USB 设备 420 中的一个或多个提供的数据或信息可由客户计算机 410 使用应用程序 604 来访问。应用程序 605 也可以向特定 USB 设备 420 提供数据或信息（即，控制）。在某些情形中，可能需要将对 USB 设备 420 中的一个或多个的访问和控制定向或重定向至驻留在服务器计算机 405 处的应用程序，而无论是否有诸如应用程序 605 的应用程序驻留在客户计算机 410 中。应用程序 605 的示例包括文字处理程序、多媒体程序以及数据管理程序。

应用程序 605 与 USB 功能设备驱动器 610 通信。USB 功能设备驱动器 610 是对其中组合 USB 设备 420 中的一个的设备类专用的。如上所述，特定 USB 功能设备驱动器支持特定的 USB 设备。例如，USB 功能设备驱动器是数码照相机、打印机、磁盘存储单元、扫描仪等专用的。USB 功能设备驱动器 610 创建表示 USB 设备 420 中的一个或多个的 FDO，被组合在同一类中的 USB 设备由同一 FDO 表示。

该特定的 FDO 是表示其中组合了一个或多个 USB 设备 420 的设备类的数据对象，并向应用程序 605 展示或提供接口。如果 USB 设备 420 是数码照相机，那么创建的 FDO（数据对象）表示一般的数码照相机而不是一个特定的数码照相机。作为驻留在 USB 功能设备驱动器 610 上的数据对象，FDO 以及由 FDO 展示的接口由与该 USB 功能设备驱动器 610 通信的应用程序 605、USB 总线驱动器 515 以及其它驱动器、模块或组件识别。

USB 功能设备驱动器 610 与 USB 总线驱动器 515 通信，USB 总线驱动器 515 也被称为“集线器驱动器”。USB 总线驱动器 515 是设备无关的，并支持多个诸如 USB 设备 420 等 USB 设备。USB 总线驱动器 515 创建 PDO，PDO 是表示 USB 设备 420 中的一个特定 USB 设备的数据对象。PDO 数据对象表示是 USB 设备 420 中的一个特定 USB 设备专用的。例如，如果 USB 设备 420 是连接的若干数码照相机中的一个，那么由 USB 集线器驱动器 515 创建标识特定 USB 设备的专用 PDO。该 PDO 被特别地传递至 USB 主机控制器 520。而且，由 USB 总线驱动器 515 创建的 PDO 也可以被用来标识供与 USB 集线器驱动器 515 通信的其它驱动器、模块或组件专用的 USB 设备。

USB 主机控制器 520 可以是对一个或多个 USB 设备 420 的物理接口，且它与 USB 总线驱动器 515 通信。USB 主机控制器 520 可以被连接至 USB 端口，该端口连接至 USB 设备 420。只要一个或多个 USB 设备 420 被连接至 USB 主机控制器 520，USB 主机控制器 520 就向 USB 总线驱动器 515 传递驻留在已连接的一个或多个 USB 设备 420 中的标识符或标识数据。USB 总线驱动器 515 创建标识 USB 设备 420 的 PDO，且向其它驱动器、模块或组件呈现该 PDO。通过标识 USB 设备 420 中的特定 USB 设备，客户计算机 410 安装（即，客户计算机 410 的 CPU 或处理器安装）访问和使用该特定 USB 设备所需的软件。这支持免去用户安装合适的软件来支持该特定 USB 设备的需求的 PNP 特征。

与应用程序 605 往来的通信可以包括 IRP 和 IOCTL 码。IRP 是源自应用程序 605 的通信数据，包括对 USB 设备 420 中的一个或多个执行动作的请求，诸如将数据传送（发送）至应用程序 605 的请求。IOCTL 码可以由应用程序 605 用来与 USB 功能设备驱动器 610 通信。IOCTL 码是由应用程序 605 传递至一个或多个 USB 设备 420 来执行特定动作的专用命令或指令。USB 功能设备驱动器 610 从应用程序 605 接收 IOCTL 码，并将该 IOCTL 码转换为传递给 USB 设备 420 中的一个或多个的 IRP。同样，从 USB 设备 420 中的一个或多个发送的 IRP 被转换为可以由

应用程序 605 读取或使用的 IOCTL。IOCTL 特别用于将来自应用程序 605 的命令和/或动作（即，请求）与诸如 USB 功能设备驱动器 610 的设备或类专用驱动器的通信。USB 功能设备驱动器 610 确认定义命令或动作的 IOCTL，尤其是 IOCTL 参数。

在客户计算机 410 处提供客户机传输驱动器 500，允许远程客户机访问与应用程序服务器计算机 405 通信。通信可以实现诸如 RDP 通信协议等通信协议，尤其实现编码发送的数据以及解码由客户机传输驱动器 500 接收的命令的通信协议。而且，客户机传输驱动器 500 可以在跨诸如因特网等网络的数据通信中实现诸如 TCP/IP 等传输协议。

在某些情况下，支持该一个或多个 USB 设备 420 的功能是由服务器计算机 405 提供的。当这样的功能由应用程序服务器计算机 405 提供时，装载远程客户机访问驱动器 535。远程客户机访问驱动器 535 执行由 USB 功能设备驱动器 610 提供的功能，包括传递源自应用程序服务器计算机 405 的命令。通过远程客户机访问驱动器 535，至 USB 设备 420 的通信从应用程序 605 和 USB 功能设备驱动 610 重定向。命令可以是 IRP 的形式，且被传递给 USB 总线驱动器 515。而且，远程客户机访问驱动器 535 识别由 USB 总线驱动器 515 创建的特定 PDO。远程客户机访问驱动器 535 特别地识别表示该一个或多个 USB 设备 420 的特定 PDO 以及由该特定 PDO 展示的接口。

应用程序服务器计算机 405 提供向诸如客户计算机 410 等一个或多个客户计算机提供特定功能的专用应用程序软件。在该示例中，应用程序服务器计算机 405 包括支持一个或多个 USB 设备 420 或向其提供功能的应用程序 510。应用程序 510 的示例包括文字处理程序、多媒体程序以及数据管理程序。

应用程序 510 与 USB 功能设备驱动器 525 通信。USB 功能设备驱动器 525 是供其中组合特定 USB 设备 420 的设备类专用的。USB 功能设备驱动器 525 创建表示 USB 设备的特定类的特定 PDO。如上所述，该特定 PDO 是表示设备类的数据对象，并且向应用程序 510 展示或提供接口。作为由 USB 功能设备驱动器 525 提供的数据对象，FDO 或由该 FDO 展示的接口特别地由应用程序 510 识别。

USB 功能设备驱动器 525 与代理 USB 总线驱动器 530 通信。代理 USB 总线驱动器 530 是设备无关的，并支持多个本地和诸如 USB 设备 420 等远程 USB 设备。代理 USB 总线驱动器 530 创建 PDO，PDO 是表示特定 USB 设备 420 并供该设备专用的数据对象。唯一的或专用的 PDO 允许服务器计算机 405 通过使用代理 USB

总线驱动器 530 来区分特定的 USB 设备。

与应用程序 510 往来的通信可以包括 IRP 和 IOCTL 码。IOCTL 码可以特别地由应用程序 510 用来与 USB 功能设备驱动器 525 通信。IOCTL 码是由应用程序 510 传递的专用命令或指令。USB 功能设备驱动器 525 从应用程序 510 接收 IOCTL 码, 并将该 IOCTL 码转换为传递给代理 USB 总线驱动器 530 的 IRP。由代理 USB 总线驱动器 530 接收的 IRP 由服务器传输驱动器 505 接收。服务器传输驱动器 505 也可以将 IRP 传递给代理 USB 总线驱动器 530, 其中这样的 IRP 表示来自诸如 USB 设备 420 等远程 USB 设备的数据和请求。

图 7 示出了将对 USB 设备的访问和控制定向或重定向至诸如应用程序服务器计算机 405 等远程计算机的进程 700。进程 700 被示为逻辑流程图中的框的集合, 这些框表示可以用硬件、软件、固件或其组合实现的操作的序列。在软件的语境中, 框表示计算机指令, 当由一个或多个处理器执行这些指令时, 执行所述的操作。进程 700 参考图 4、5 和 6 中描述的应用程序服务器计算机 405 和客户计算机 410 来描述, 其中应用程序服务器计算机 405 和客户计算机 410 提供执行特定进程的装置。尽管被描述为流程图, 但构想到该进程可以并发地发生。

在框 705 处, USB 设备被连接至客户计算机。该设备可以被物理地连接至客户计算机上的 USB 端口。该 USB 端口物理地与 USB 控制器连接。USB 总线驱动器与该 USB 控制器连接。该 USB 总线驱动器识别并标识该特定的 USB 设备, 还标识组合该 USB 设备的类。通过识别进程, 提供了 USB 设备的即插即用 (PNP), 在该示例中识别进程是特别地由 USB 总线驱动器执行的。换言之, 识别 USB 设备, 如果适用, 那么无需用户干涉就安装合适的软件。由该 USB 总线驱动器特别地提供被称为物理设备对象 (PDO) 的数据对象以及供该 USB 设备专用的接口。该 PDO 以及接口被传递至连接至该 USB 总线驱动器的组件、模块和驱动器。

在某些情况下, 期望客户计算机向已连接的 USB 设备提供功能, 或对其访问和控制 (即, 沿着框 710 的“否”分支)。在框 715 处, 当客户计算机处的应用程序为 USB 设备提供功能, 安装或装载专用的 USB 功能设备驱动器。USB 功能设备驱动器对该 USB 设备的设备类专用。USB 功能设备驱动器特别地提供供该已连接的 USB 设备专用的功能设备对象 (FDO) 和接口。该 FDO 和接口被传递至连接至 USB 功能驱动器的组件、模块以及驱动器。提供从应用程序到 USB 设备的通信路径。

在框 720 处, 执行 USB 设备与应用程序之间的通信。应用程序可以发送描述

对数据的请求的 I/O 控制 (IOCTL) 码。应用程序也可以接收 IOCTL 码。IOCTL 码在从应用程序发送之后, 可以被转换为 I/O 请求包 (IRP)。而且, 由应用程序接收的 IOCTL 码可以作为 IRP 源自 USB 设备。

在某些其它情况下, 期望应用程序服务器计算机向已连接的 USB 设备提供功能, 或对该设备访问和控制 (即, 沿着框 710 的“是”分支)。示例性的情况包括当客户计算机含有有限的应用程序或不含有应用程序来向 USB 设备提供足够的功能的时候。在其它情况下, 尽管应用程序可以驻留在客户计算机处来为 USB 设备提供功能, 但是客户计算机处的用户可能期望使得应用程序服务器计算机为 USB 设备提供功能。

在框 725 处, 在客户计算机处安装或装载远程客户机访问驱动器。来自应用程序服务器计算机, 尤其是服务器计算机处的应用程序的通信通过该远程客户机访问驱动器来定向, 这实质上提供了从应用程序服务器计算机 (即, 应用程序服务器计算机处的应用程序) 至 USB 设备的通信路径。可以在客户计算机处提供允许与应用程序服务器计算机通信的客户机传输驱动器。应用程序服务器计算机可以包括从客户计算机的客户机传输驱动器中接收通信并向其发送通信的服务器传输驱动器。

在框 730 处, 在应用程序服务器计算机处识别 USB 设备。具体地, 在应用程序服务器计算机处实现的代理 USB 总线驱动器识别 USB 设备以及该 USB 设备的特定类。

在框 735 处, 远程客户机访问总线驱动器装载或安装对该 USB 设备类专用的 USB 功能设备驱动器。USB 功能设备驱动器是供该 USB 设备所属的设备类或组专用的。

在框 740 处, 执行 USB 设备和驻留在应用程序服务器计算机处的应用程序之间的通信。应用程序可以发送描述对转化为由 USB 设备接收的 IRP 的数据的请求的 IOCTL。USB 功能设备驱动器可以创建并提供 FDO (设备专用数据对象)。FDO 向应用程序表示 USB 设备。

示例性计算机

图 8 示出了适合作为用于实现本发明的各个方面的环境的示例性计算设备或计算机 800, 例如作为客户计算机 410 或应用程序服务器计算机 405。计算机 800 的组件可以包括, 但不限于, 处理单元 820、系统存储器 830 和将包括系统存储器

830 在内的各种系统组件耦合至处理单元 820 的系统总线 821。系统总线 821 可以是若干类型的总线结构中的任一种，包括存储器总线或存储器控制器、外围总线和使用各种总线体系结构中的任一种的局部总线。作为示例，而非限制，这样的体系结构包括工业标准体系结构（ISA）总线、微通道体系结构（MCA）总线、扩展的 ISA（EISA）总线、视频电子技术标准协会（VESA）局部总线和外围部件互连（PCI）总线（也被称为 Mezzanine 总线）。

示例性计算机 800 通常包括各种计算机可读介质。计算机可读介质可以是能够被计算机 800 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。作为示例，而非限制，计算机可读介质可以包括计算机存储介质和通信介质。计算机存储介质包括以任何方法或技术实现的用于存储诸如计算机可读指令、数据结构、程序模块或其它数据等信息的易失性和非易失性、可移动和不可移动介质。计算机存储介质包括，但不限于，RAM、ROM、EEPROM、闪存或其它存储器技术、CD-ROM、数字多功能盘（DVD）或其它光盘存储、磁带盒、磁带、磁盘存储或其它磁性存储设备、或能用于存储所需信息且可以由计算机 800 访问的任何其它介质。通信介质通常具体化为诸如载波或其它传输机制等已调制数据信号中的计算机可读指令、数据结构、程序模块或其它数据，且包含任何信息传递介质。术语“已调制数据信号”指的是这样一种信号，其一个或多个特征以在信号中编码信息的方式被设定或更改。作为示例，而非限制，通信介质包括有线介质，诸如有线网络或直接线连接，以及无线介质，诸如声学、RF、红外线和其它无线介质。上述中任一个的组合也应包括在计算机可读介质的范围之内。

系统存储器 830 包括易失性或非易失性存储器形式的计算机存储介质，诸如只读存储器（ROM）831 和随机存取存储器（RAM）832。基本输入/输出系统 833（BIOS）包含有助于诸如启动时在计算机 800 中元件之间传递信息的基本例程，它通常存储在 ROM 831 中。RAM 832 通常包含处理单元 820 可以立即访问和/或目前正在操作的数据和/或程序模块。作为示例，而非限制，图 8 示出了操作系统 834、应用程序 835、其它程序模块 836 和程序数据 837。

示例性计算机 800 也可以包括其它可移动/不可移动、易失性/非易失性计算机存储介质。仅作为示例，图 8 示出了从不可移动、非易失性磁介质中读取或向其写入的硬盘驱动器 841，从可移动、非易失性磁盘 852 中读取或向其写入的磁盘驱动器 851，以及从诸如 CD ROM 或其它光学介质等可移动、非易失性光盘 856 中读取或向其写入的光盘驱动器 855。可以在示例性操作环境中使用的其它可移动/不可

移动、易失性/非易失性计算机存储介质包括，但不限于，盒式磁带、闪存卡、数字多功能盘、数字录像带、固态 RAM、固态 ROM 等。硬盘驱动器 841 通常由不可移动存储器接口，诸如接口 840 连接至系统总线 821，磁盘驱动器 851 和光盘驱动器 855 通常由可移动存储器接口，诸如接口 850 连接至系统总线 821。

以上描述和在图 8 中示出的驱动器及其相关联的计算机存储介质为计算机 800 提供了对计算机可读指令、数据结构、程序模块和其它数据的存储。例如，在图 8 中，硬盘驱动器 841 被示为存储操作系统 844、应用程序 845、其它程序模块 846 和程序数据 847。注意，这些组件可以与操作系统 834、应用程序 835、其它程序模块 836 和程序数据 837 相同或不同。操作系统 844、应用程序 845、其它程序模块 846 和程序数据 847 在这里被标注了不同的标号是为了说明至少它们是不同的副本。用户可以通过输入设备，诸如键盘 848 和定点设备 861（通常指鼠标、跟踪球或触摸垫）向示例性计算机 800 输入命令和信息。其它输入设备（未示出）可以包括麦克风、操纵杆、游戏垫、圆盘式卫星天线、扫描仪等。这些和其它输入设备通常由耦合至系统总线的用户输入接口 860 连接至处理单元 820，但也可以由其它接口或总线结构，诸如并行端口、游戏端口或通用串行总线（USB）连接。监视器 862 或其它类型的显示设备也经由接口，诸如视频接口 890 连接至系统总线 821。除监视器 862 以外，计算设备也可以包括其它外围输出设备，诸如扬声器 897 和打印机 896，它们可以通过输出外围接口 895 连接。

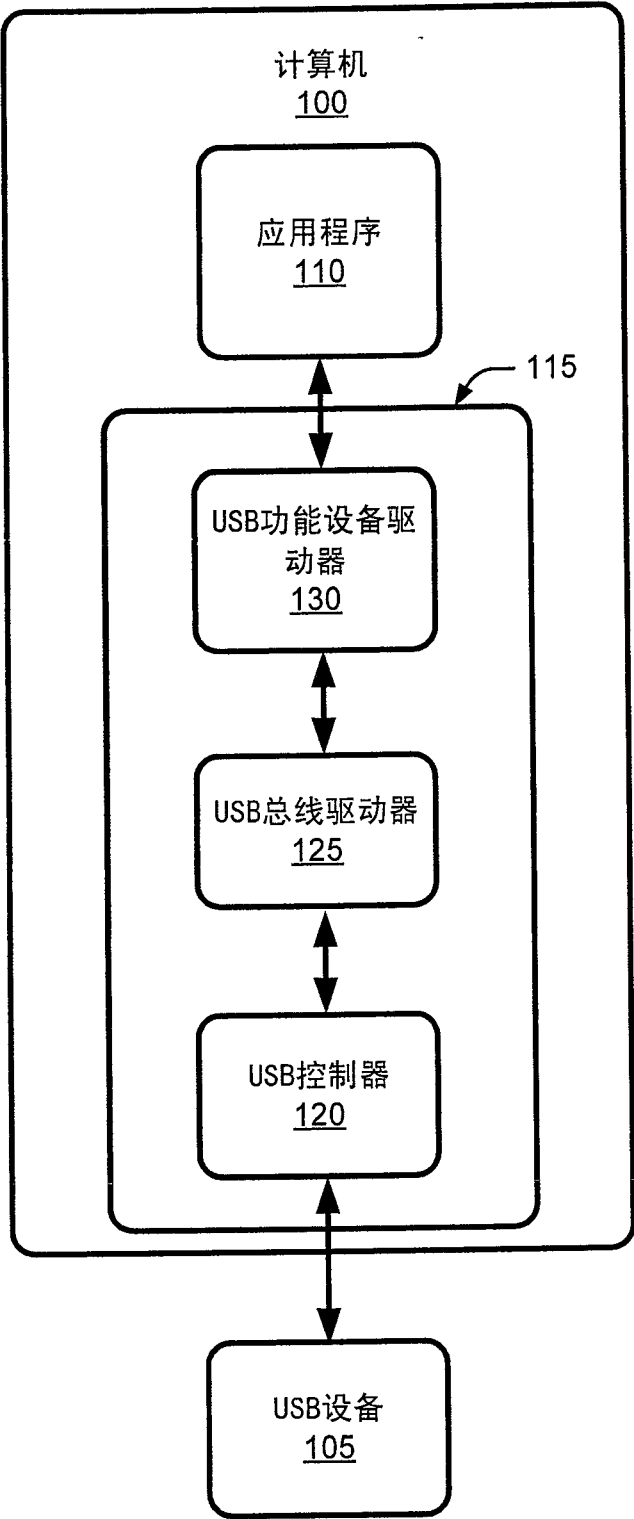
示例性计算机 800 可使用至一个或多个远程计算机，诸如远程计算设备 880 的逻辑连接在网络化环境中操作。远程计算设备 880 可以是个人计算设备、服务器、路由器、网络 PC、对等设备或其它常见网络节点，且通常包括上文相对于计算机 800 描述的许多或所有元件，尽管在图 8 中只示出存储器存储设备 881。图 8 中所示逻辑连接包括局域网（LAN）871 和广域网（WAN）873，但也可以包括其它网络。这样的网络环境在办公室、企业范围计算机网络、内联网和因特网中是常见的。

当在 LAN 网络环境中使用时，示例性计算机 800 通过网络接口或适配器 884 连接至 LAN 871。当在 WAN 网络环境中使用时，示例性计算机 800 通常包括调制解调器 872 或用于通过诸如因特网等 WAN 873 建立通信的其它装置。调制解调器 872 可以是内置或外置的，它可以通过用户输入接口 860 或其它合适的机制连接至系统总线 821。在网络化环境中，相对于示例性计算机 800 所描述的程序模块或其部分可以存储在远程存储器存储设备中。作为示例，而非限制，图 8 示出了远程应用程序 885 驻留在存储器设备 881 上。可以理解，所示的网络连接是示例性的，且

可以使用在计算机之间建立通信链路的其它手段。

结论

上文描述了将对连接至客户计算机的 USB 设备的访问和控制选择性地定向或重定向至服务器计算机。尽管本发明是以对结构特征和/或方法性动作专用的语言来描述的，但是可以理解，定义在所附权利要求书中的本发明不必限于所述的具体特征或动作。相反，将该具体特征和动作揭示为实现所要求保护的本发明的示例性形式。



现有技术

图 1

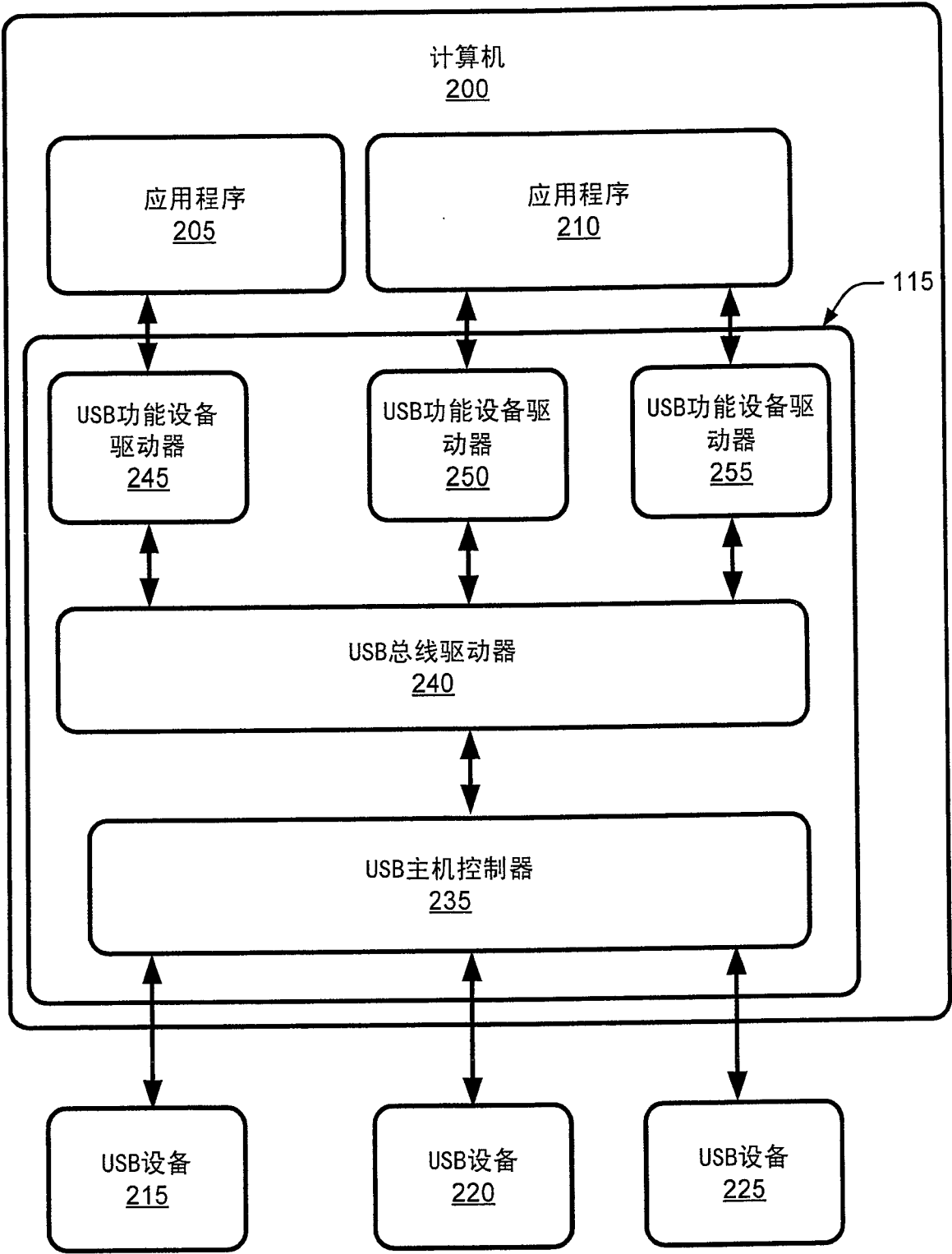


图 2
(现有技术)

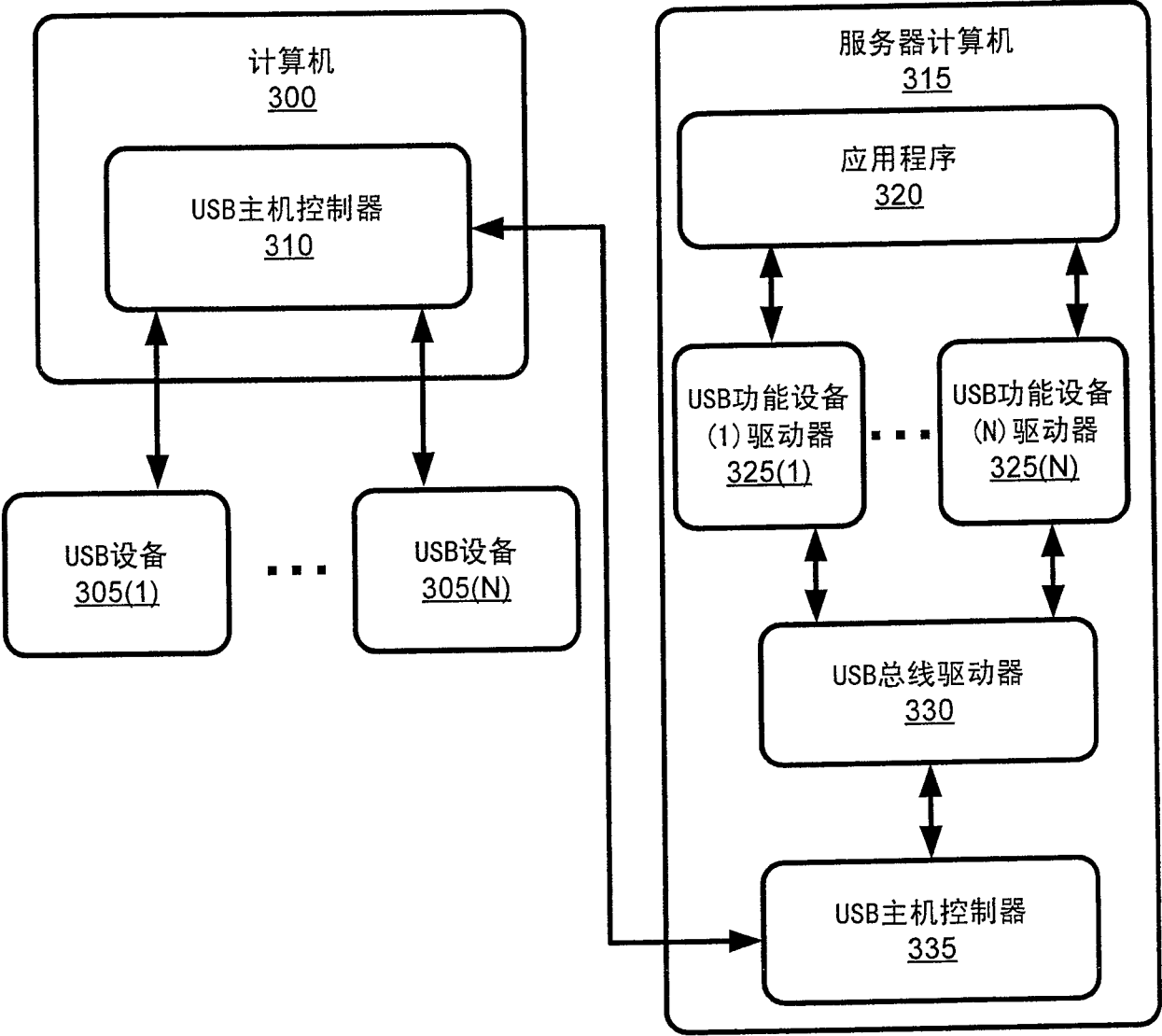


图 3
(现有技术)

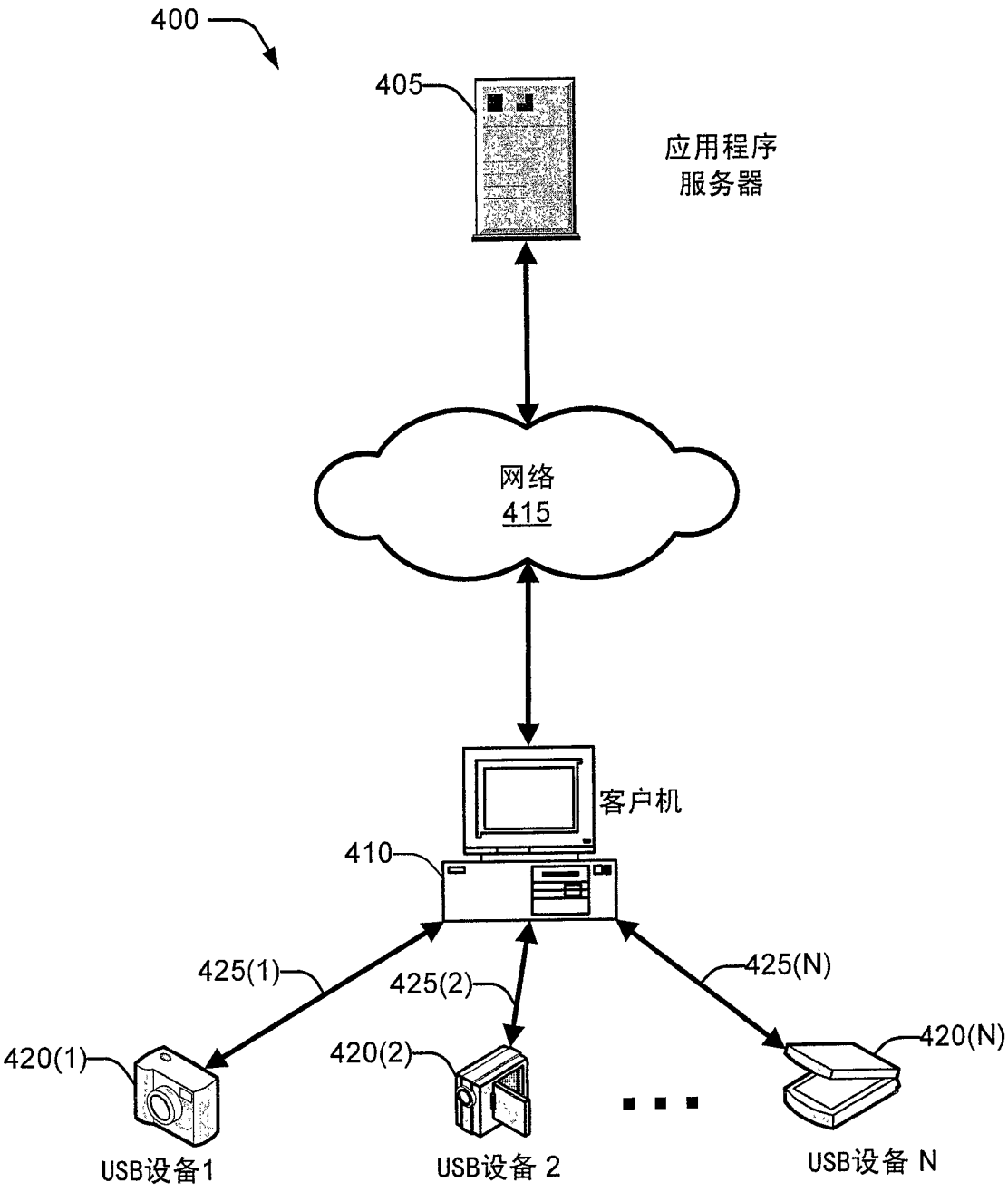


图 4

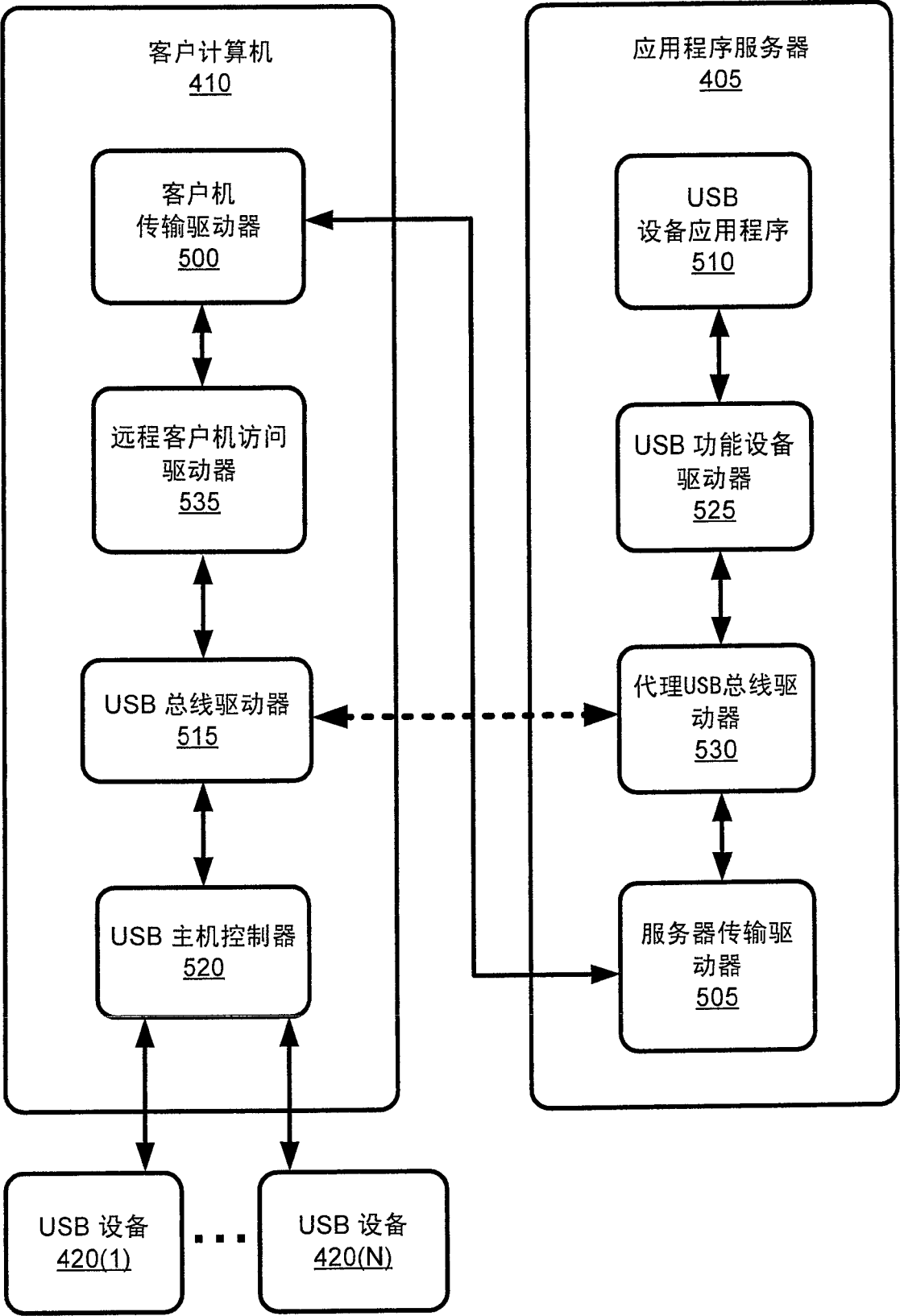


图 5

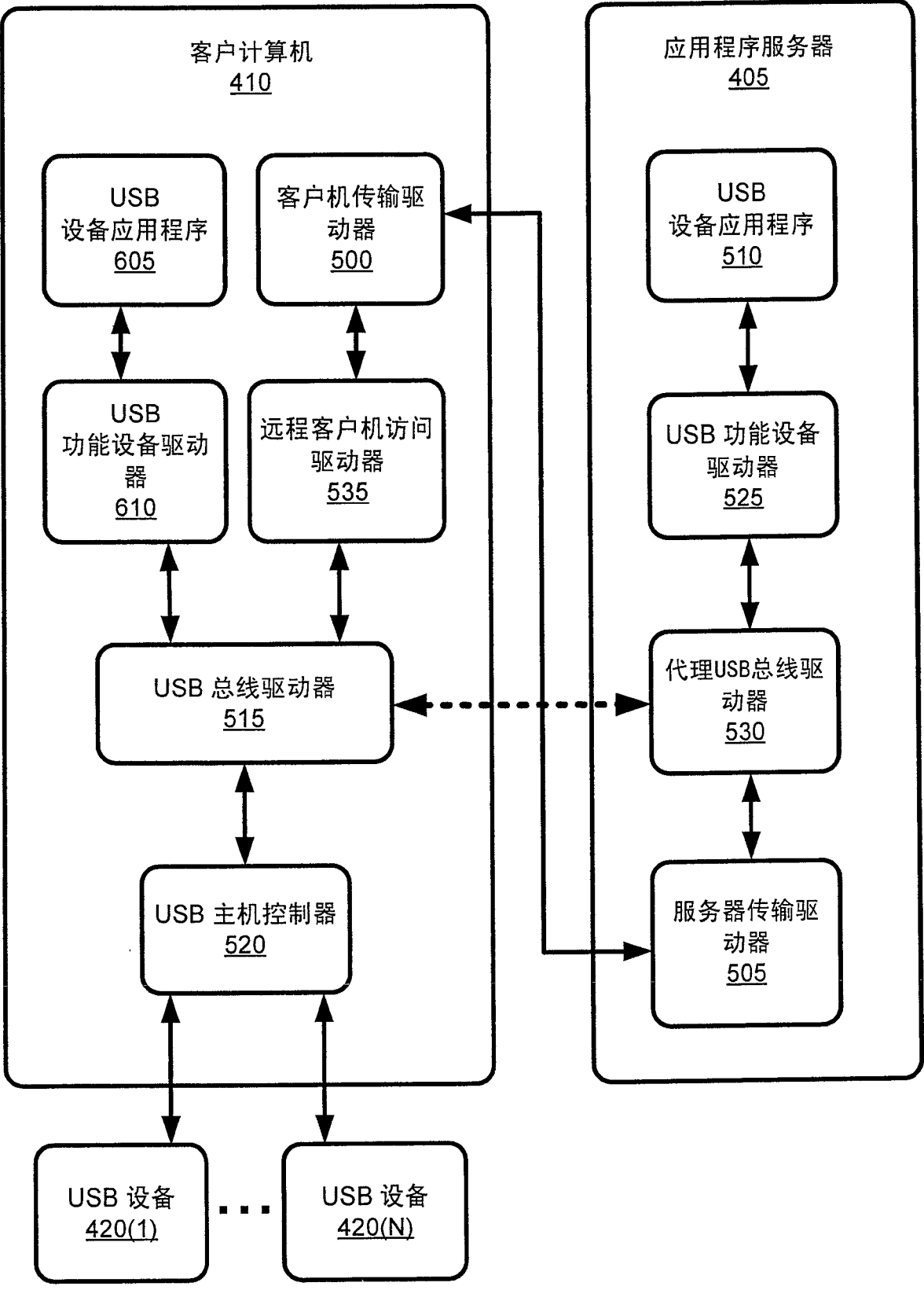


图 6

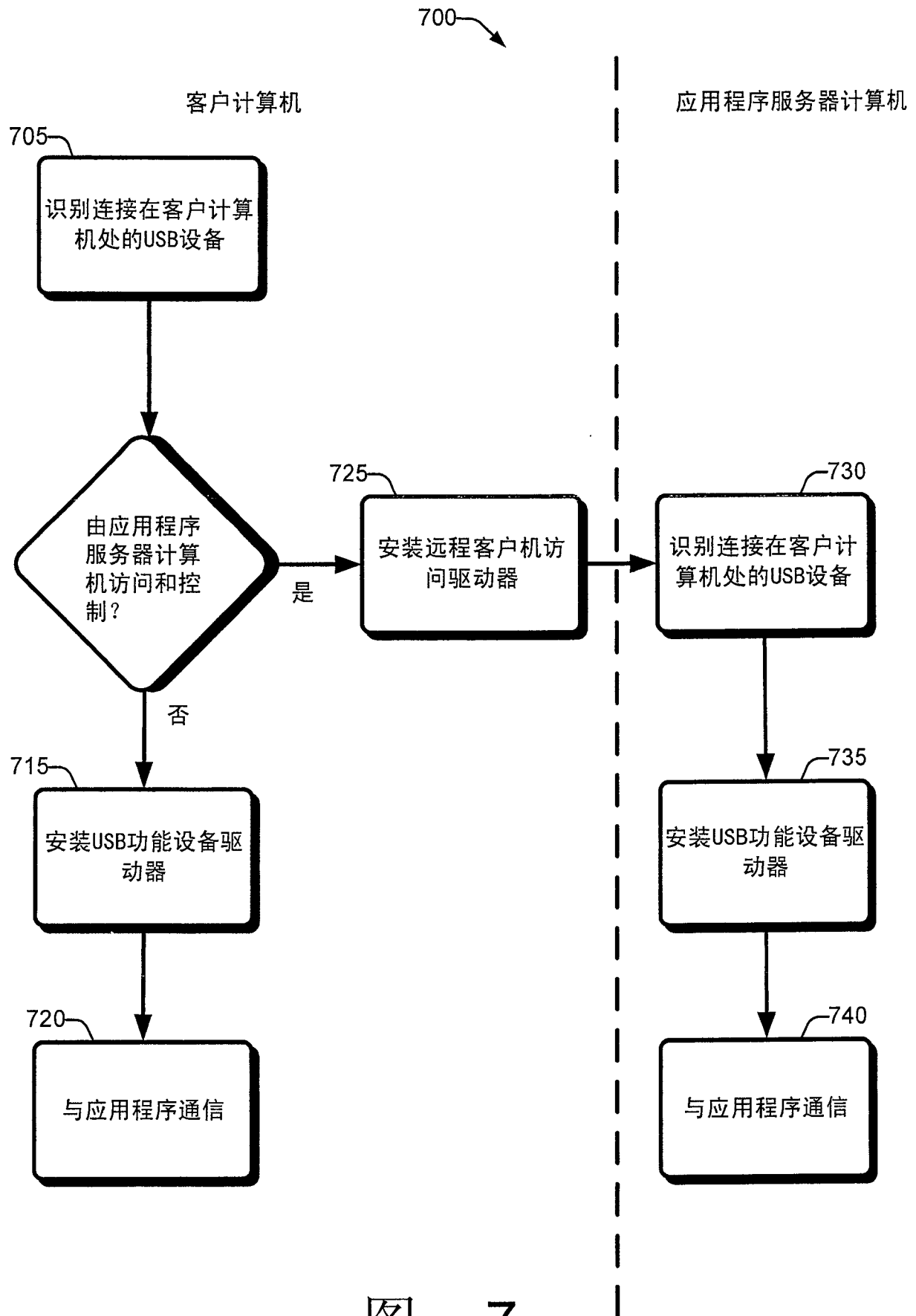


图 7

