



[12] 发明专利说明书

[21] ZL 专利号 01814158.7

[45] 授权公告日 2005 年 2 月 16 日

[11] 授权公告号 CN 1189817C

[22] 申请日 2001.7.23 [21] 申请号 01814158.7

[30] 优先权

[32] 2000. 8. 14 [33] CA [31] 2,316,003

[86] 国际申请 PCT/GB2001/003315 2001.7.23

[87] 国际公布 WO2002/015006 英 2002.2.21

[85] 进入国家阶段日期 2003.2.14

[71] 专利权人 国际商业机器公司

地址 美国纽约州

[72] 发明人 萨蒂什·冈加比索恩

审查员 赵晓春

[74] 专利代理机构 北京市柳沈律师事务所

代理人 王志森 黄小临

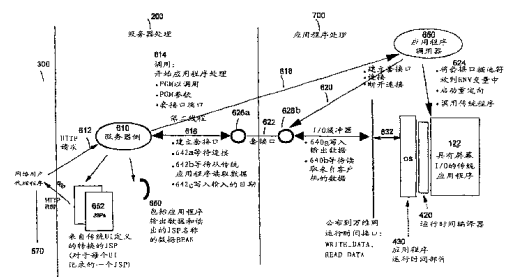
权利要求书 2 页 说明书 13 页 附图 6 页

[54] 发明名称 从因特网访问传统应用程序

[57] 摘要

一种方法和装置，能够从诸如因特网的网络运行交互传统应用程序而不需要在应用程序中改变任何代码。因此，所述应用程序不知道新的网络环境并继续就象在它的本机环境中运行。可以通过利用可以连接到所述计算机或与其结合的网络服务器从几个客户机装置中的任何一个访问传统应用程序，其中在所述计算机上正在执行所述应用程序。一般，这些传统应用程序对企业是重要的，在所述计算机上是自含式的，具有混合的商业和用户接口逻辑，并在分布式计算的软件设计原则出现之前写入。在传统应用程序的情况下，万维网应用程序结构所要求的将商业逻辑与用户接口逻辑分离是不实用的。诸如精简的客户机的客户机具有一个诸如万维网浏览器的网络用户代理程序，它可以访问连接到所述计算机的一个网络服务器。本发明的方法提供了一个环境使得当从所述网络用户代理程序调用一个应用程序的时候，一个运行时间数据收发器截取所述应用程序的原始数据并将数据发送到所述网络服务器，所述网络服务器随后在动态更新诸如 Java 服务器网页的的相关的应用程序网页之后通

过网络将所述数据用于所述网络用户代理程序，所述相关的应用程序网页通过转换所述传统应用程序的专有显示屏幕而产生。用户通过所述网络用户代理程序输入的数据被经由同样的运行时间截取而被返回发送到所述应用程序。以这种方式，在所述应用程序现在能够利用许多因特网和其他网络能力的优点的同时客户机和网络环境对于所述应用程序是透明的。



1. 一种执行在计算机上安装的一个计算机应用程序(122)的方法, 所述方法包括步骤:
- 5 a) 在连接到第一网络上的所述计算机的一个服务器(200)中建立一个小程序实例(610);
- b) 在所述计算机上运行所述应用程序(122)以产生应用程序输出数据; 其特征在于:
- c) 在所述计算机上截取所述应用程序输出数据和将其改发(640a)到一个网络公布部件(550);
- 10 d) 将所述应用程序输出数据从所述网络公布部件(550)向所述小程序实例(610)发送; 和
- e) 在所述服务器中建立数据对象(650)和向所述数据对象提供所述应用程序输出数据。
- 15 2. 按照权利要求1的方法, 还包括步骤:
从通过第二网络连接到所述服务器的一个客户机请求所述应用程序(122);
用所述应用程序输出数据更新至少一个网页; 和
向所述客户机发送被更新的网页。
- 20 3. 按照权利要求2的方法, 还包括步骤:
将所述应用程序的显示文件转换为能够显示应用程序输出数据的网页。
4. 按照权利要求2的方法, 其中所述网页被存储在所述服务器上。
5. 按照权利要求1-4的任何一个的方法, 还包括步骤:
在计算机中建立一个用于应用程序输出数据的输入/输出缓冲器。
- 25 6. 按照权利要求1-4中的任何一个的方法, 其中还包括步骤:
建立在所述小程序实例和所述网络公布部件之间的第一端点连接;
其中所述端点连接是下列之一: 一个套接口, 一个数据队列对象或一个消息队列。
7. 一种计算机系统, 用于执行一个应用程序, 包括:
- 30 a) 一个中央处理单元;
- b) 一个主存储器, 通过一条通信总线与所述中央处理单元连接, 所述应

- 用程序存储在所述主存储器中，能够在所述中央处理单元上被执行；
- c) 一个数据存储单元，与一个数据存储接口连接，所述数据存储接口与
所述通信总线连接；
 - d) 至少一个输入/输出装置，连接到所述通信总线，并与到一个外部计算
5 机网络的一个网络接口相连接；
 - e) 一个网络公布部件；
- 其特征在于：
- f) 一个数据改发器，用于将所述应用程序输出数据改发到所述网络公布
部件；
 - 10 g) 一个输入/输出接口，用于存储所述被改发的应用程序输出数据。
8. 一种计算机服务器，用于访问在一个计算机上存储和执行的一个应用
程序，包括：
- a) 一个中央处理单元；
 - b) 一个网络接口，通过一个网络连接到至少一个客户机；
 - 15 c) 一个小服务程序处理单元，用于从所述至少一个客户机接收一个访问
所述应用程序的请求，并将所述请求发送到所述计算机，所述小服务程序处
理单元通过一个服务器端点连接部，向所述计算机发送实时数据和从所述计
算机接收实时数据，所述应用程序在所述计算机上被执行；其中所述小服务
程序处理单元从所述应用程序接收实时数据并向所述服务器中的多个数据对
20 象提供输入/输出数据。
9. 按照权利要求8的计算机服务器，还包括：
- 网页显示单元，用于显示多个网络显示网页，每个网络显示网页对所述
应用程序的每个输入/输出屏幕定义是唯一的，
- 其中所述小服务程序处理单元更新所述网络显示网页以通过所述网络向
25 所述客户机发送。

从因特网访问传统应用程序

5 技术领域

本申请一般地涉及计算机软件应用领域，具体涉及通过诸如因特网的计算机网络来访问传统的应用程序。

背景技术

10 象所有的其他事物一样，计算机的使用这些年来已经改变。在早些年里，大型计算机主宰着产业，直到出现独立的个人计算机。现在，许多企业和家庭具有至少一台独立的个人计算机，即一台 PC。但是已经出现了一种新的计算范例：以网络为中心的计算或分布式计算，其中至少两个、但是更可能是更多的计算机——称为客户机和服务器——被通过网络互相连接，其中由客户机使用的软件应用程序驻留在服务器上。因此，服务器可以通过网络提供
15 由最终用户或由多个其他服务器和客户机使用的应用程序或客户程序或数据库。

在许多情况下并且越来越如此，连接客户机和服务器的网络是因特网。因特网指的是互相连接的计算机网络的集合，其中所述计算机网络使用因特
20 网协议，如 TCP/IP、UDP 等。软件应用程序被以独立于服务器或客户机的操作系统的编程语言来编写，以便访问和使用在因特网上的无数应用程序。而且，描述在因特网上的数据的数据的多种语言包括各种可扩充标记语言 (XML)，如超文本标记语言 (HTML)，无线标记语言 (WML) 等。万维网指的是一种软件管理方案，它利用超文本链接以各种用户代理程序访问因特网。

25 分布式计算已经从根本上改变了软件开发人员编写这些应用程序的代码的方法。软件开发人员不是编写一个具有几千或上百万行代码的巨大的自含式应用程序，而是将多个称作部件 (component) 的代码小部分选择和编织在一起，每个部件与一个特殊的功能(function)相关联，所述特殊功能可以被重新使用于在万维网上的多个应用程序。各部件是标准软件，它们可以被软件开发人员
30 从服务器拉下和并入到新的应用程序中。例如，一个日历部件可以被用于诸如调度应用程序、表示应用程序和/或数据库应用程序的几个应用程序以计

算雇员的假期和工资等。而且，用于因特网和万维网应用程序的当前的产业架构要求商业逻辑（business logic）与表示逻辑（presentation logic）分离，以便商业逻辑单独负责提供商业数据，而表示逻辑单独负责驱动表示或用户接口。

- 5 但是，具有几千或上百万行代码的巨大自含式应用程序可能仍然对于企业和组织很重要。实际上，一些自含式程序可能包括一个企业的整体存货清单/客户清单/数据库或所有上述内容。术语“传统应用程序（legacy application）”指的是旧的应用程序，它们仍然在企业和组织中被运行，但是，是在分布式计算和结构化程序设计以及它的模块化、耦合、内聚(cohesion)技术出现之前被开发的。被编写为一个大自含式程序的传统应用程序不容易适应于改变。但是这些传统应用程序经常对企业是如此的重要以至这些企业在关于这些应用程序的未来和开发新网络和因特网技术来用于远程和分布式访问这些重要应用程序的能力上面临困难的抉择。具有混合的商业和表示逻辑的交互式传统应用程序必须被重新构造或重新设计以便能够作为万维网应用程序。但是将商业逻辑与用户接口逻辑分离在以最少的干扰和测试将应用程序使用到万维网中方面不总是可行的。

- 20 在因特网和万维网上执行传统应用程序的其他方案可以使用屏幕剪贴（screen-scraping）技术，该技术将在运行时间的传统显示数据流转换为基于万维网的语言，如 HTML 或其他语言。经 HTML 转换的屏幕随后被显示在浏览器上。但是在运行时间转换传统的显示数据流较慢，牺牲了应用程序的性能，并不向用户提供提供将传统的应用程序与新的万维网或其他网络应用程序无缝地结合或桥接的能力。而且仅仅传统数据的转换也不提供当应用程序在因特网上运行的时候增进应用程序的用户接口的有效性的能力。

- 25 US 5754830 公开了一种服务器和万维网浏览器终端仿真器，用于永久连接到传统主机系统，并公开了一种基于屏幕剪贴技术的操作方法。使用屏幕剪贴技术的一种产品是 IBM®的 WebSphere®主机公布器。IBM 和 WebSphere 是国际商用机器公司的商标。

发明内容

- 30 因此在产业中需要从因特网、万维网或其他计算机通信网络来访问传统的应用程序而不必重新构造应用程序或对程序改变代码。

在产业中也需要将传统的应用程序于其他网络应用程序无缝地交织。

为此，本发明提供一种执行在计算机上安装的一个计算机应用程序的方法，所述方法包括步骤：在连接到第一网络上的所述计算机的一个服务器中建立一个小程序实例；在所述计算机上运行所述应用程序以产生应用程序输出数据；其中在所述计算机上截取所述应用程序输出数据和将其改发到一个网络公布部件；将所述应用程序输出数据从所述网络公布部件向所述小程序实例发送；和在所述服务器中建立数据对象和向所述数据对象提供所述应用程序输出数据。

上述方法，还包括步骤：从通过第二网络连接到所述服务器的一个客户机请求所述应用程序；用所述应用程序输出数据更新至少一个网页；和向所述客户机发送被更新的网页。

上述方法还包括步骤：将所述应用程序的显示文件转换为能够显示应用程序输出数据的网页。

按照上述方法，所述网页被存储在所述服务器上。

上述方法还包括步骤：在计算机中建立一个用于应用程序输出数据的输入/输出缓冲器。

上述方法还包括步骤，建立在所述小程序实例和所述网络公布部件之间的第一端点连接；其中所述端点连接是下列之一：一个套接口，一个数据队列对象或一个消息队列。

本发明提供一种计算机系统，用于执行一个应用程序，包括：一个中央处理单元；一个主存储器，通过一条通信总线与所述中央处理单元连接，所述应用程序存储在所述主存储器中，能够在所述中央处理单元上被执行；一个数据存储单元，与一个数据存储接口连接，所述数据存储接口与所述通信总线连接；至少一个输入/输出装置，连接到所述通信总线，并与到一个外部计算机网络的一个网络接口相连接；一个网络公布部件；其中一个数据改发器，用于将所述应用程序输出数据改发到所述网络公布部件；一个输入/输出接口，用于存储所述被改发的应用程序输出数据。

本发明还提供一种计算机服务器，用于访问在一个计算机上存储和执行的一个应用程序，包括：一个中央处理单元；一个网络接口，通过一个网络连接到至少一个客户机；一个小服务程序处理单元，用于从所述至少一个客户机接收一个访问所述应用程序的请求，并将所述请求发送到所述计算机，

所述小服务程序处理单元通过一个服务器端点连接部，向所述计算机发送实时数据和从所述计算机接收实时数据，所述应用程序在所述计算机上被执行；其中所述小服务程序处理单元从所述应用程序接收实时数据并向所述服务器中的多个数据对象提供输入/输出数据。

- 5 上述计算机服务器，还包括：网页显示单元，用于显示多个网络显示网页，每个网络显示网页对所述应用程序的每个输入/输出屏幕定义是唯一的，其中所述小服务程序处理单元更新所述网络显示网页以通过所述网络向所述客户机发送。

10 附图说明

图 1 是一种能够执行本发明的优选实施例的计算机系统的高级方框图。

图 2 是按照本发明的原理的一种服务器系统的简化方框图，所述服务器系统可以访问存储在所述计算机系统上的传统应用程序。

- 15 图 3 是能够利用本发明的优点的客户机和服务器的一个计算机网络的简化表示图。

图 4 是现有技术计算机传统应用程序环境的简化处理图和方框图，示出了要显示的数据的流动。

图 5 是如何按照本发明的原理将数据向一个网络和一个传统应用程序改发和从所述网络和所述传统应用程序改发的简化处理图。

- 20 图 6 是在因特网上如何利用万维网技术从一个传统应用程序改发数据的简化处理图和方框图。

具体实施方式

- 25 参见附图，其中在几个图中同样的数字表示同样的部件，图 1 示出了按照本发明的一个实施例的一种计算机系统 100 的高级方框图。计算机系统 100 可以包括中央处理单元 (CPU) 101、主存储器 102、终端接口 103、数据存储器 104 和诸如因特网的网络的接口 105。各种装置经由内部通信总线 110 互相通信。CPU 101 是一种通用可编程处理器，执行存储在存储器 102 中的指令；虽然图 1 示出了单个的 CPU，应当明白可以使用具有多个 CPU 的计算机系统。存储器 102 是一个随机存取半导体存储器，用于存储数据和程序；
30 存储器被在概念上示出为单个的整体实体，但是公知的是，存储器经常被配

置为多个高速缓冲存储器和其他存储装置的一个分层结构。操作系统 120 和应用程序 122 驻留在存储器 102 中。其中尤其是操作系统 120 提供诸如装置接口、存储网页管理、多任务管理等的功能，这是在本领域内公知的。应用程序 122 可以包括传统应用程序，如果它包括一个服务器软件应用程序，则
5 网络接口 105 可以与所述服务器软件应用程序 122 交互以使得计算机系统 100 能够作为网络服务器。

终端接口 103 可以支持连接单个或多个终端，可以被实现为一个或多个电路卡或其他单元。数据存储器 104 优选地包括一个或多个旋转磁性硬盘驱动单元，但是也可以使用其他类型的数据存储器。网络接口 105 提供一个物
10 理连接，用于到网络和自网络的数据传输。在本优选实施例中，网络是因特网，但是网络也可以是任何更小的自含式网络，如内联网、WAN、LAN 或其他内部或外部网络，它们使用例如电话传输线路、卫星、光纤、T1 线路等和任何不同的可以获得的技术。通信总线 110 支持在不同装置之间传送数据、命令和其他信息；虽然被以简化形式示出为单个的总线，但是它一般被构造
15 成多总线，并且可以被配置为多层的形式。

图 1 所示的计算机系统打算作为一个简化的表示图，可以明白，除了在此具体所述的那些之外，在系统配置中可能有许多变化形式。虽然系统 100 可以是一个个人计算机系统，但是计算机 100 也可以是更大的计算机系统，如 IBM 企业系统或 IBM AS/400 系统。虽然在此描述了一种特定的硬件配置
20 和各种替代形式，但是所述的方法一般可以利用允许从客户机通过计算机网络访问传统应用程序的任何硬件配置来实现。CPU 101 被适当地编程以通过具有传统应用程序和网络接口来实现本优选实施例，所述网络接口可以被其他连接的计算机使用来访问那个传统应用程序。

计算机系统 100 和它的部件在图 1 中被示出和描述为一个或多或少的单个自含式计算机系统。另外可能的是使用多计算机系统，尤其是共享单个大数据库的多系统。每个具有一个特殊的任务。例如，一个或多个计算机系统
25 100 可以专用于一个或多个访问数据库的传统应用程序，而一个或多个其他计算机系统 100 可以专用于为从客户机接收的请求服务或访问因特网。在此对计算机的描述应当被理解为包括或者单个计算机或者一个计算机系统集
30 合，它们提供对传统应用程序和对网络的访问，通过所述网络来连接一个客户机系统。

图 2 是一个网络服务器 200 的示例，它可以访问存储在计算机 100 上的传统应用程序。网络服务器 200 可以是一个计算机系统，包括 CPU 201、主存储器 202、各种装置适配器和接口 203-208、通信总线 210。CPU 201 是一个通用可编程处理器，执行存储在存储器 202 中的指令；虽然图 2 示出单个 CPU，但是应当明白可以使用具有多个 CPU 的计算机系统。存储器 202 是随机存取半导体存储器，用于存储数据和程序；存储器被从概念上示出为单个的单片实体，可以明白存储器 202 经常被配置为多个高速缓冲存储器和其他存储装置的一个分层结构。通信总线 210 支持在不同装置之间的数据、命令和其他信息的传送；虽然在附图中被以简化形式示出为单个总线，但是它也可以被构造为多总线，并且可以被配置为多层形式。显示适配器 203 支持视频显示器 211，它可以是阴极射线管显示器、平面显示器或触摸屏，但是也可以使用其他显示技术。键盘/指示器适配器 204 支持键盘 212 和被描述为鼠标的指示装置 213，可以明白也可以使用其他形式的输入装置。存储器适配器 205 支持一个或多个数据存储装置 214，它可以是旋转磁性硬盘驱动器或 CD-ROM 驱动器，但是也可以使用其他数据存储装置。打印机适配器 206 支持打印机 215。适配器 207 可以支持多种附加装置中的任何一个，如音频装置等。网络接口 208 向诸如因特网的网络提供物理接口。这个接口可以包括一个连接到一条电话线的调制解调器，通过所述电话线可以达到因特网服务提供商或在线服务提供商，其他更高带宽的接口越来越多地被使用。例如，网络服务器 200 可以经由一个使用以太网、令牌环(Token Ring)或其他协议的局域网连接到另一个网络服务器，而第二个网络服务器又连接到因特网。或者，可以通过有线电视、光纤、卫星、无线方式或其他连接来提供网络接口 208。图 2 表示的是打算作为高端服务器的示范简化表示，可以明白，在其他网络服务器 200 中，除了在此所述的那些，可能具有在系统配置中的许多变化形式。如果计算机系统 100 具有一个服务器软件应用程序作为它的应用程序 122 之一，则网络服务器 200 和计算机 100 可以被合并到同一系统中，在这种情况下，在网络服务器 200 和计算机 100 之间的网络是一条内部通信总线。

参见图 3，每个客户机系统 300 是与计算机 100 分离的装置，它可以访问在计算机 100 上驻留和运行的传统应用程序。客户机系统可以是一个个人计算机系统或一个诸如服务器的大计算机系统或诸如笔记本电脑或膝上型电

脑的小计算机系统。最后，客户机系统 300 根本不必是计算机，而是最好是更简单的用具似的具有更少的存储量的客户装置，如网络终端、精简客户机、终端式装置、语音响应单元等。计算、电信和消费电子装置的结合正在引起作为客户机 300 的普遍的移动装置的数量和种类上的惊人增长。这个移动结构使得具有多种客户机 300，包括膝上型电脑、子笔记本电脑 (sub-notebook)、诸如个人数字助理和伴随装置 (companion device) 的手持计算机、诸如智能电话 (smartphone)、寻呼机、简单消息接发装置、便携装置的移动设备。因此当客户机系统 300 是一个移动装置的时候，显示适配器和网络接口具有一个网络用户代理程序，并支持多种多模接口，其中包括传统的键盘和鼠标接口、小文本屏幕、输入笔、触摸屏、语音识别装置、文本到语音转换装置、诸如便携装置的其他正在出现的技术。网络用户代理程序使得计算机的应用程序能够在它的各个客户机 300 上使用。优选的打算是客户机系统 300 包括任何电子装置，所述电子装置可以通过诸如万维网浏览器的网络用户代理程序来与网络服务器 200 交互以访问在计算机系统 100 上驻留的传统或其他应用程序。这种特殊用途的用于访问万维网的装置例如电视机的因特网访问盒或便携无线万维网访问装置，它们可以作为用于调用和执行应用程序的用户代理程序来实现，它们也要被包括在客户机系统 300 的范围中。网络用户代理程序可以以控制电路通过使用逻辑门、可编程逻辑装置或其他硬件部件来实现，以取代基于处理器的系统。

下面将详细说明，优选实施例的各个方面与在计算机 100 或网络服务器 200 上的可以执行的具体的方法步骤有关。在一个替代实施例中，本发明可以被实现为一种计算机程序产品，用于网络服务器 200 和客户机 300 之一或两者。可以经由多种信号承载介质来向计算机 100 和/或网络服务器 200 提供限定优选实施例的功能的程序，包括但不限于：(a) 永久存储在不可写入的存储介质中的信息，所述介质例如在任何计算机中的只读存储装置，如由 CD-ROM 驱动器 214 可以读出的 CD-ROM 盘；(b) 存储在可写入存储介质，例如在软盘驱动器中的软盘或硬盘驱动器如图 2 中示出为 214 的驱动器上的可改变信息；或 (c) 通过电话或有线媒体网络、包括无线通信而传输到计算机的信息。这样的信号承载介质，当载有指示本发明的功能的计算机可读的指令的时候，是本发明的替代实施例。

图 3 是计算机网络的简化表示图。所述计算机网络表示因特网，它可以

被描述为基于在此说明的客户机-服务器模型的公知的计算机网络。在概念上，因特网包括多个网络服务器 200 的一个大网络，这些网络服务器 200 可以由客户机系统 300 通过专用因特网接入提供商 303 或在线服务提供商 304 来访问。在本优选实施例中，每个客户机系统 300 可以运行各自的诸如浏览器的网络用户代理程序以经由接入提供商来访问网络服务器 200。每个网络服务器 200 可以具有它自己的传统应用程序，或者可以通过因特网或其他到客户机 300 的连接部分连接到其他具有要被访问的传统应用程序的计算机 100。而且，传统应用程序不必被仅仅存储在一个计算机 100 上；而是单一应用程序的各个任务可以被存储在多个计算机 100 上，所述计算机 100 通过诸如因特网的网络连接到网络服务器 200 上。通过通用资源定位符 (URL) 来识别到服务器 200 的因特网网络路径，所述 URL 具有限定网络连接的公知句法。虽然示出了各种相对直接的路径，但是应当明白，图 3 仅仅是一个概念表示，诸如因特网的计算机网络实际上可以具有远远更为复杂的结构。也应当明白计算机网络也可以是内联网或其他内部计算机网络，如 WAN、LAN 等。虽然本实施例不打算如此限定，但它的优选实施例在因特网上尤其有用。

本实施例提供了一个环境，所述环境使得交互传统应用程序不用重新构造或重新设计应用程序代码而允许这些传统应用程序如同原来——在它们的本机环境中继续运行，并同时允许多个客户机利用万维网浏览器或其他网络技术来访问传统应用程序。本实施例也向交互传统应用程序提供了这样的能力，即产生所要求的商业数据和驱动按照由产业标准万维网应用结构要求的各个表示人工产物。本实施例令人感兴趣的是在“有状态(stateful)”传统应用程序和“无状态(stateless)”因特网事务之间的交互。交互传统应用程序的有状态在于，应用程序继续执行直到它到达一个 I/O 指令，在此执行被暂停并且应用程序的状态被保留在它的本机环境中直到 I/O 行为结束。另一方面，因特网事务使用诸如 HTTP 的无状态协议，在此，在因特网上的一个事务被完成之后，到因特网的连接被破坏并且事务的状态被丢失，除非另外的处理发生来将这个状态存储在网络服务器上。本实施例提供了一个机制和一个环境，从此使传统应用程序的状态在它转换到无状态万维网环境的同时被自动保留在它的本机环境中。应用程序的状态的自动保留是本发明的结构所固有的，本发明的结构提供了由交互传统应用程序所要求的必要连续性。

参见图 4，在此示出了现有技术计算机应用程序环境的简化处理图和方

框图。每个传统应用程序 122 具有数据 422，数据 422 按照程序 I/O 代码 410 通过运行时间编译器 420 被输入到应用程序运行时间操作系统 430 和从到应用程序运行时间操作系统 430 被输出。输出数据 422 可以被发送到数据管理器代码 432，其为应用程序运行时间操作系统 430 的一部分。在所述应用程序运行时间操作系统 430 中的工作站管理器 434 或等同的功能部件将输出数据 422 与来自传统应用程序的用户接口定义文件 440 的对应显示记录相结合以产生要显示在专有显示装置 211 上的数据流 442。当数据被插入到数据流中和被发送到显示装置以显示的时候，在屏幕定义本身中定义应用程序数据的显示格式，并由操作系统的运行时间操作系统 430 将格式代码施

5 加到数据。在应用程序运行时间之前，这些用户接口定义文件 440 被编译和插入到一个或多个在运行时间引用的显示对象中。输入的数据沿着反向路径流动：应用程序运行时间操作系统 430 从输入的数据流中提取输入数据，格式化数据，然后所述数据被提交到应用程序作为用户输入数据。在一些系统上，显示装置 211 仅仅在输入的数据流中建立屏幕的动态部分，即输入字段。

10

15 图 5 是一种按照本发明的原理的方法的简化处理图，所述方法使得传统应用程序能够通过网络被使用，所述网络优选是利用万维网技术的因特网，虽然其他网络和其他技术也被认为包括在本发明的范围内。假定本机应用程序屏幕的专有用户接口 (UI) 定义——在图 4 中为屏幕定义 440——被分别存储在屏幕定义文件中。首先，甚至在运行时间之前，在步骤 510 对专有屏

20 幕定义 440 进行语法分析和转换为一种格式，所述格式可以由诸如万维网浏览器的任何普遍计算机因特网用户代理程序 570 提供；这个格式最好基于诸如 HTML、WML 或 *ML 的 XML，这依赖于针对用于产生所谓的网页的因特网用户代理程序。在本优选实施例中，用户接口元件被机械地映射到支持用户接口元件的基于 XML 的语言中。

25 转换处理的定制可以以几种方式来实现，并且如果最终用户公司对于应用程序表示具有良好限定的规则，则转换处理的定制是所期望的。转换处理可以识别由所述公司使用的任何用户接口设计模式，因此使得用户可以定制模版来映射这些设计模式。转换算法随后使用这些格式来识别实际的应用程序屏幕并因此转换。如果转换的格式被保留在各个文件中，则由用户执行的

30 转换后改进是可能的。在作为 Java 服务器网页的网页的实施例中，通过将显示文件记录 440 转换为包括 DHTML 的 Java 服务器网页 520 来产生这些网页。

DHTML 是具有客户使用的级联类型表和 JavaScript 的 HTML, 用以执行输入字段的本地确认, 并便利在转换后的改进以针对用户的表示格式来定制。否则, 可以在观看屏幕的时候通过分析传统应用程序的屏幕定义或显示数据流来发现公司的屏幕设计规则。从这些分析获得的知识库可以随后在转换处理
5 前按照需要被改进。

在步骤 530, 在向在客户机 300 上的网络用户代理程序 570 发送被转换的用户接口页 520 之前, 利用现有的网页服务技术在网络服务器 200 上动态更新被转换的用户接口页 520。在步骤 540, 应用程序的原始输出数据被应用程序运行时间操作系统 430 的数据管理器 432 改发, 以便用于显示的输出数据
10 不被转换为图 4 所述的传统专有显示数据流, 而是被发送到标有“公布到万维网”的新的网络公布部件 550。网络公布部件 550 重新格式化应用程序数据以公布到网络, 因为其中有传统应用程序以不适合于在网络上显示的格式产生和编码应用程序数据。从网络用户代理程序 570 获得的的用户输入数据也被作为网络公布部件的公布到万维网的运行时间部件 550 处理, 并被重
15 新格式化为提交到应用程序运行时间操作系统 430 的应用程序的输入数据。网络服务器 200 和公布到万维网运行时间部件 550 的一部分可以被安装到计算机 100 上, 而不是其中安装了传统应用程序 122 和应用程序运行时间操作系统 430, 但是也可能与具有传统应用程序 122 的计算机 100 合并并作为其一部分。而且, 公布到万维网运行时间部件 550 被设计使得传统应用程序 122
20 不知道在它的本机环境中的任何改变, 因此不要求改变对应用程序的代码。

在作为网络的因特网和万维网的一个具体实施例中, JAVA 使得 Java 服务器网页作为网页以在所述网页被用于网络用户代理程序之前向 HTML 或 XML 页动态插入应用程序数据, 这个处理称作动态产生的万维网内容。由 Java 服务器网页读取的数据被存储在数据对象中, 即 JavaBean 中, 它们被已
25 经接收数据以公布的小服务程序提供数据。本实施例的特性是, 一旦所述传统应用程序数据已经被存储在网络服务器中的这些数据对象中, 则它可以用于任何目的, 包括从另一个连接到服务器的计算机的访问、用于操纵、用于传输... .., 在万维网服务器上的这个传统动态数据的用途是无限的。JavaBean 的数据字段定义具有与由 Java 服务器网页容纳的 HTML 或 XML 网页的可变
30 数据字段的一对一映射, 并且 JavaBean 的数据字段定义在转换处理期间产生作为 Java 类定义, 因为专有应用程序屏幕的定义包括屏幕的 I/O 字段。这样

的映射是本领域内公知的,还存在诸如 Active Server Pages(动态服务器网页)的其他产品,它们能够从服务器向客户机提供动态内容。在任一实施例中,小服务程序实例创建这些数据对象并向它们提供改发的传统应用程序的动态数据,通常是 I/O 数据。因此,优选的是,数据对象具有必要的方法来格式化数据以用于显示和用于将数据从应用程序的专有代码转换为专有语言的网络码页。

仅仅通过举例,图 6 示出了基于使用 Java 服务器网页的公布到万维网运行时间部件的结构。公布到万维网运行时间部件的实体有:小服务程序实例 610,它作为万维网网关,用于请求传统应用程序 122——612,和从传统应用程序 122 回答——660。一般,小服务程序是响应于客户请求而在服务器上运行的一个 JAVA 程序。对于输出的数据,所述小服务程序实例 610 从套接口接受一个数据缓冲器,查找数据的记录识别符,创建诸如 JavaBean 的相关数据对象和向它们提供数据,并启动相关的 Java 服务器网页以向网络用户代理程序提供数据供显示。对于提供的数据,小服务程序实例 610 经由所述套接口和公布到万维网运行时间部件将数据封装到一个数据缓冲器中并将其提交到应用程序。小服务程序实例 610 也建立应用程序处理和和应用程序处理和网络服务器处理之间的通信路径。所述通信路径可以包括套接口 626 作为一个通信端点。应当注意,套接口和其他通信端点——它们在等待输入的数据的同时暂停应用程序的执行——将固有地保留应用程序的状态和在本发明的环境中发生作用。应用程序调用器 660 保存套接口描述符以供 Read_data 和 Write_data 方法以后使用,启动数据的改发和调用传统应用程序。应用程序运行时间操作系统 430 调用 Read_data 和 Write_data 方法来在应用程序和因特网之间改发数据流。网络用户网页——在此为 Java 服务器网页——在用户接口转换处理期间产生。数据对象——如在这个情况下的 JavaBean——与每个 Java 服务器网页相关,并包括经由 Java 服务器网页公布到万维网的应用程序数据。

按照本发明的原理的处理的控制流进行如下:在客户机 300 上的用户通过在步骤 612 首先从在网络服务器 200 上的小服务程序实例 610 请求一个调用页而从网络用户代理程序 570 启动传统应用程序 122——优选的是一个万维网浏览器。在步骤 614,客户机 300——这可能是另一个计算机或个人——完成一组用于要调用的应用程序的名称和参数的输入字段并提交它。在步骤

616, 小服务程序实例 610 在接收到用于调用的 XML 或 HTTP 请求的时候建立一个套接口 626a 并向计算机 100 产生一个第二线程 618 以开始调用传统应用程序的应用程序处理。套接口 626a 随后在步骤 642a 等待连接。

第二线程 618 将套接端口号码和程序信息传送到应用程序调用器 660, 5 它建立一个套接口 626b 并在步骤 620 在网络服务器处理中插入到所述套接口的连接。在步骤 622 在两个套接口 626a 和 626b 之间建立连接之后, 在网络服务器处理中套接口 626b 等待来自传统程序的数据。应用程序调用器 660 向一个可以获得的环境写入应用程序侧的套接口 626b 的描述符, 通过应用程序编程接口启动数据改发, 并调用传统应用程序 122, 如步骤 624 所示。当传统应用程序 122 达到一个 I/O 指令的时候, 在步骤 632, 输出数据被发送到计
10 算机的操作系统的的应用程序运行时间操作系统 430, 它在 640a 调用 Write_Data 方法以将数据改发到应用程序侧套接口 626b。在步骤 642b 在网络服务器处理中套接口 626a 读取数据, 在步骤 650, 小服务程序实例 610 建立一个数据对象以存储数据。在步骤 652, 小服务程序实例 610 调用与这个输出记录相
15 关的网页, 如 Java 服务器网页。在步骤 652, 所述网页被从所述数据对象提供数据, 然后在步骤 660 用于网络用户代理程序 570 例如作为 HTTP 响应。

在网络用户代理程序 570 上的客户机接口和应用程序 122 之间的后续交互流经套接口连接 626a 和 626b。当客户机提交一个网页的时候, 在网络服务器上的小服务程序实例 610 从先前存储的会话对象——它具有对一个套接口
20 或数据队列对象或消息队列对象的引用——还原会话数据。输入的数据然后在步骤 642c 被改发到套接口或队列 626a, 改发到其他应用程序套接口或队列 626b 和 I/O 缓冲器——如果有的话, 和改发到应用程序运行时间操作系统 430, 并最后改发到正在等待 Read_Data 方法 640b 的传统应用程序 122。传统应用程序 122 随后在计算机 100 上在它的本机环境中继续它的执行直到下一个 I/O 交换。当应用程序结束的时候, 控制返回应用程序调用器 660, 它在步
25 骤 620 关闭端点的连接。

在图 6 中, 一个报告程序发生器 (RPG) 程序被用作传统应用程序的一个示例, 但是应当明白所述结构也可以支持以 COBOL 和其他编程语言所编写的传统应用程序。套接口被用作在分布的处理之间的通信机制, 但是也可以
30 使用服务器可以获得任何通信端点机制, 如用于增进性能的数据队列对象——如果服务器和传统应用程序在一个计算机中——或用于增进的可靠性

和安全性的消息队列。在预先开始的工作中的一个初始程序可以向一个可以改变的环境写入应用程序侧套接口的描述符、启动改发、调用传统应用程序。当所述程序遇到一个 I/O 指令——如一个写入指令，其后跟随读取指令，在 RPG 中的 EXFMT——的时候，输出数据被发送到一个数据管理器，它调用
5 公布到万维网运行时间接口，Write_Data，以向套接口写入。通过利用公布到万维网的运行时间接口 Read_Data 也可以由套接口 626b 读取数据。

因此，本发明使得用户可以整体的对商业重要的传统应用程序配置于因特网上以利用现代技术的优点，而不用必须重新构造程序或对程序进行代码改变。而且，传统应用程序正常地被执行。传统应用程序数据变得通过网络
10 可以获得来用于任何目的，因为它被存储在网络服务器上的数据对象中。而且，本发明使得用户可以改进诸如 Java 服务器网页的网页以在网络服务器中插入到新的基于万维网的应用程序的链接。这个能力允许在传统和其他基于网络的应用程序之间的交互，包括在网络服务器中将传统应用程序数据与来自其他网络应用程序的数据结合。允许远程调用程序的计算机和客户机系统
15 能够使用上述的结构来在分离的计算机系统中运行万维网服务器和传统应用程序。而且，本发明使得用户可以按照公司的用于应用程序表示的规则和原则来定制 Java 服务器网页。在此提供的方案还使得 I/O 数据可以被转换成适合于除了浏览器之外的显示类型的其他格式。一个有力的应用是将 I/O 数据和显示记录转换为一般的基于 XML 的用户接口。XML 用户接口可以随后被
20 提供到不同类型的用户接口再现器(renderer)。这些再现器可以是基于 Java-Swing 的、基于语音的和基于 PDA 的。因此本发明打开了从多种类型的未来装置对现有的传统应用程序的访问。

虽然已经在上面说明了本发明的不同实施例，应当明白，它们的提供是示例性的而不是限定性的，并且各种变化是可能的。因此本发明的宽度和范
25 围不应当被上述的示范性实施例中的任何一个所限定，而是应当按照所附的权利要求和它们的等价物来限定。

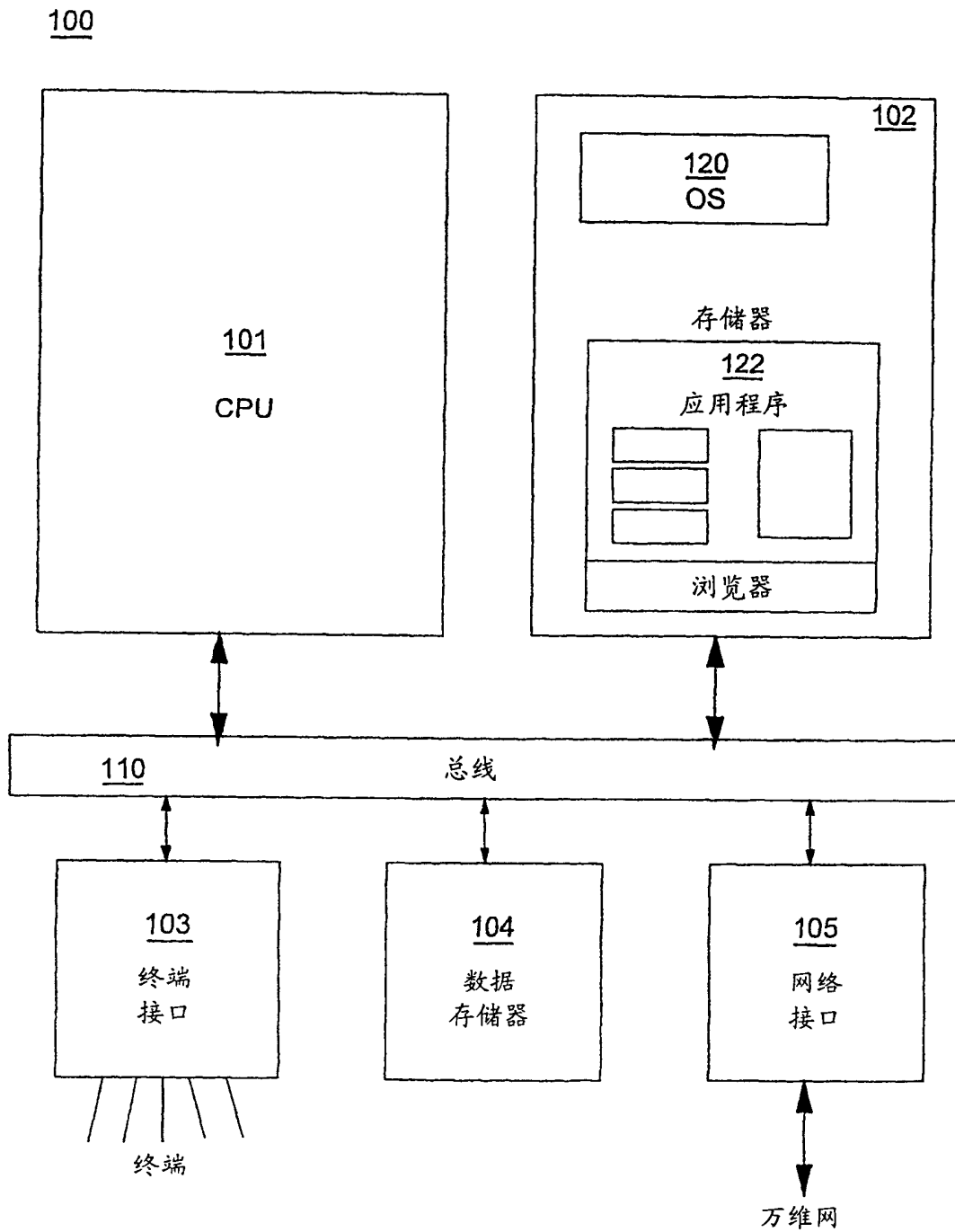


图 1

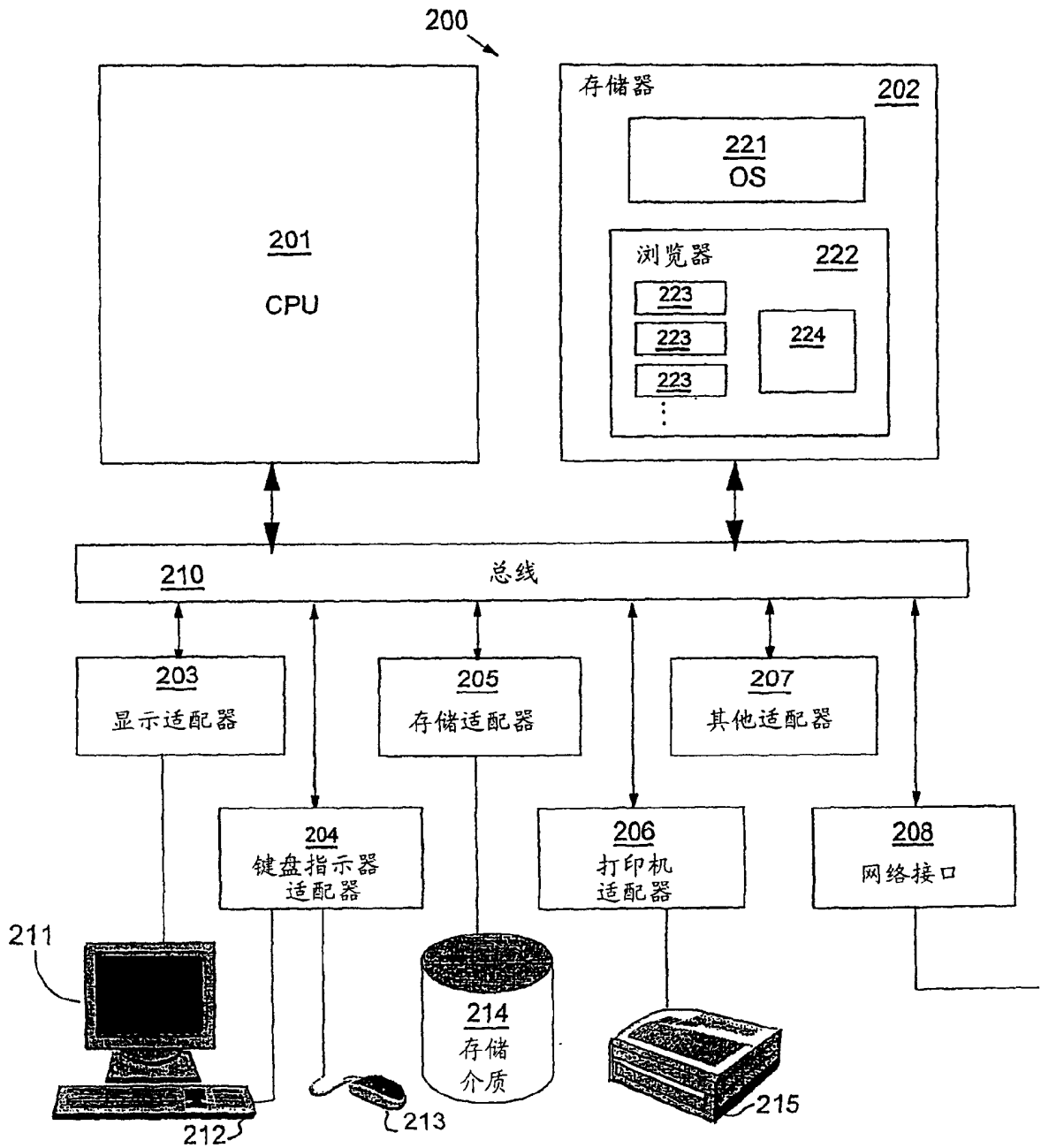


图 2

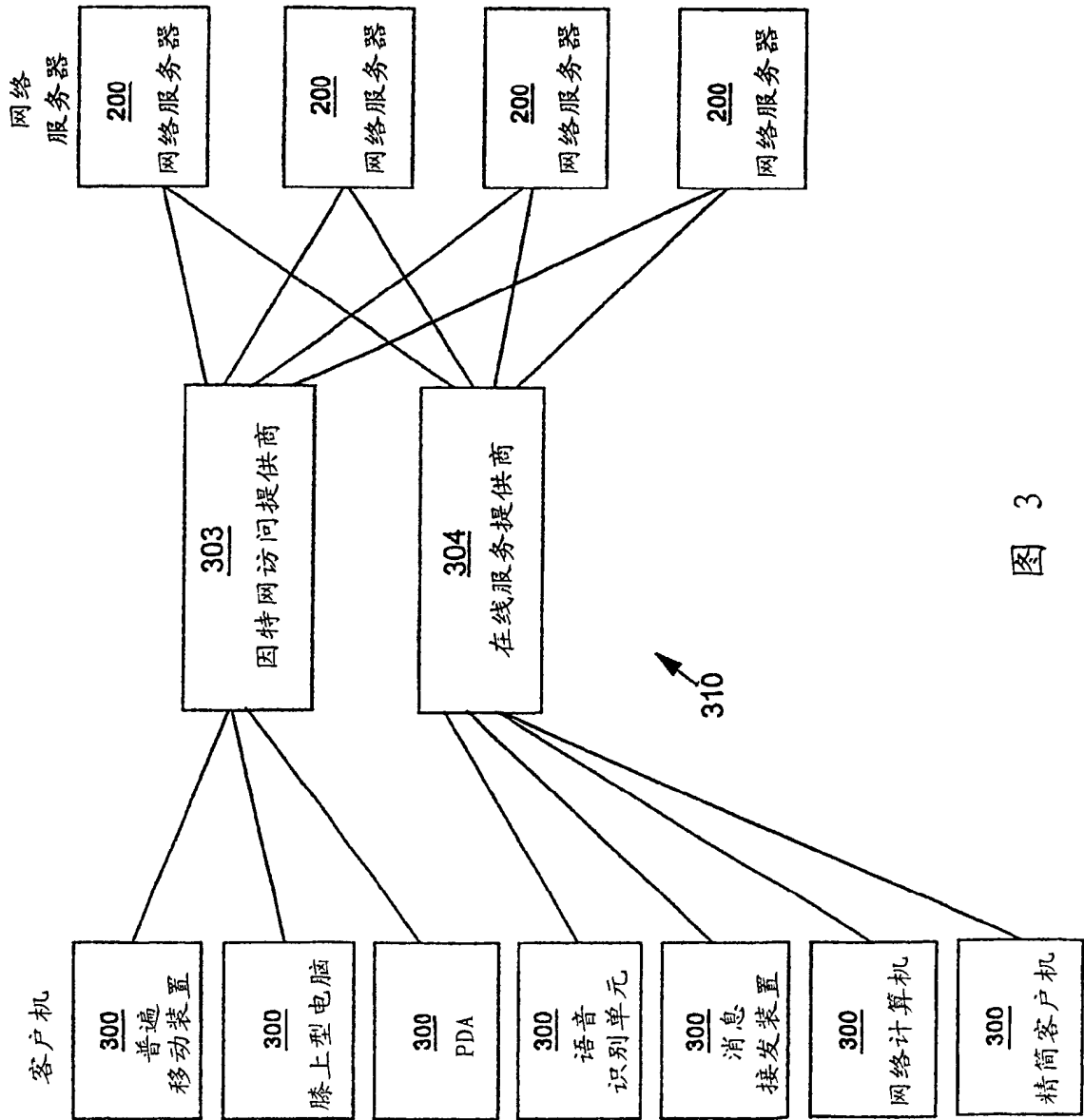


图 3

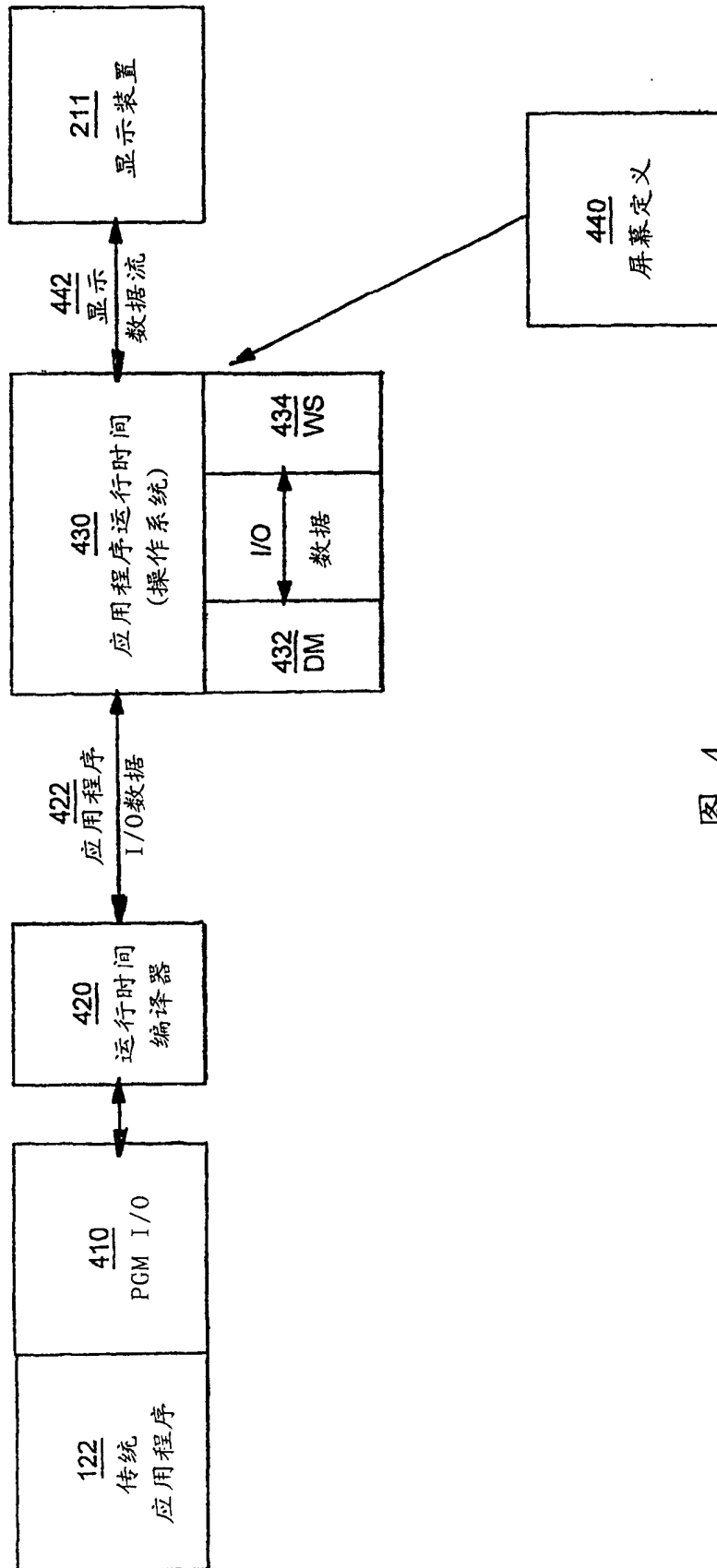


图 4

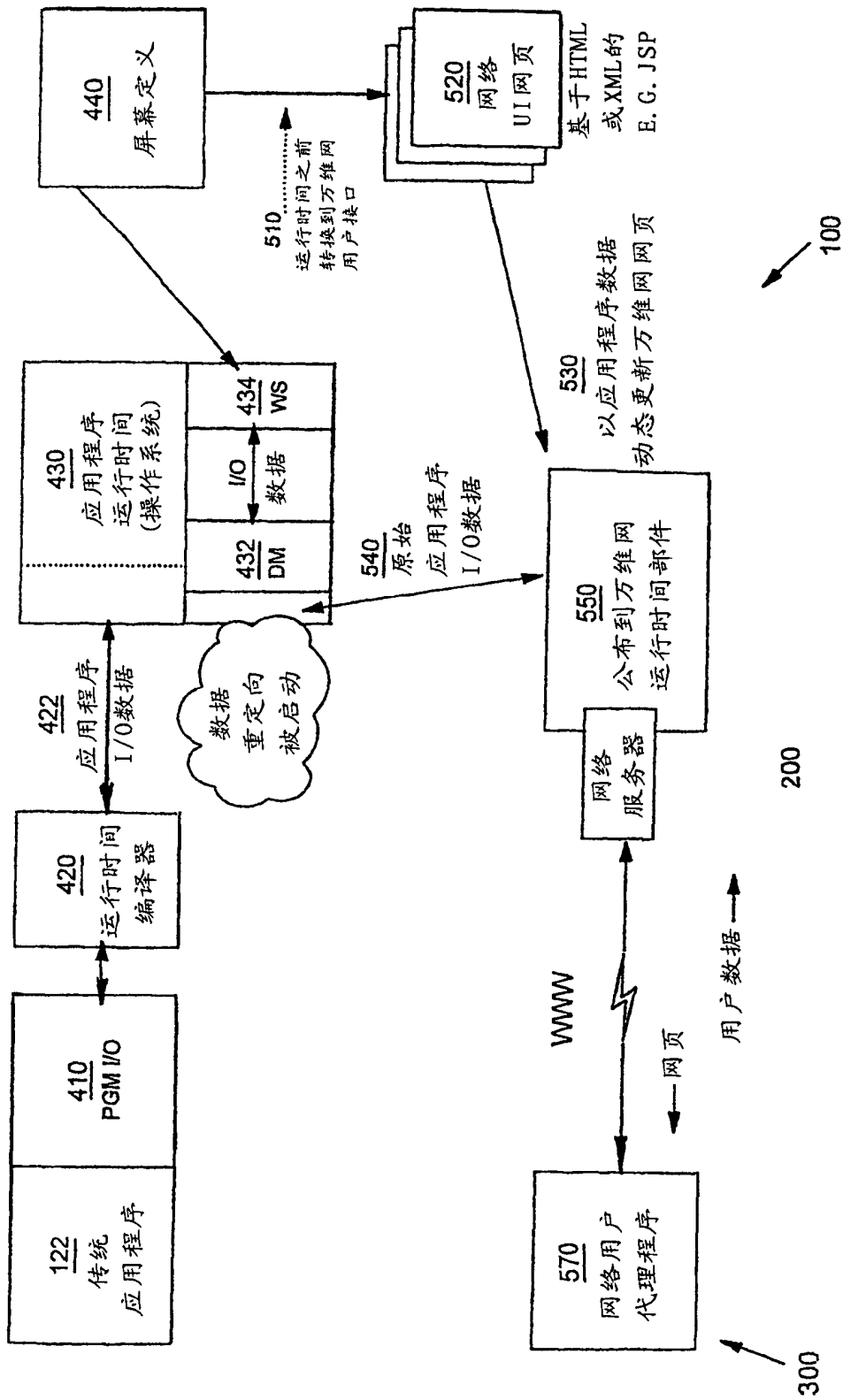


图 5

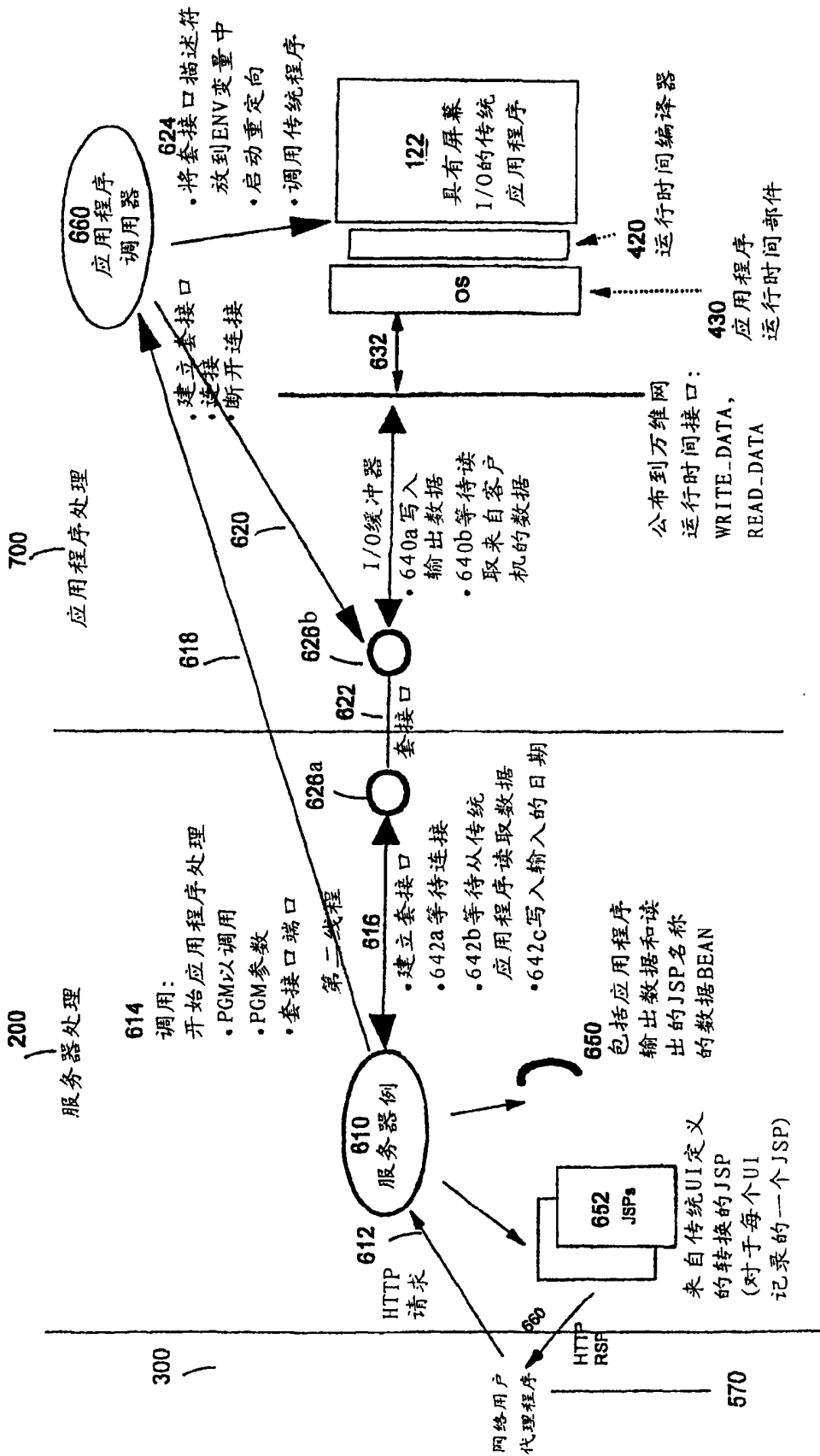


图 6