

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/30 (2006.01)



[12] 发明专利申请公开说明书

[21] 申请号 200510116656.0

[43] 公开日 2006年6月7日

[11] 公开号 CN 1783079A

[22] 申请日 2005.10.26

[21] 申请号 200510116656.0

[30] 优先权

[32] 2004.11.30 [33] EP [31] 04292833.3

[71] 申请人 阿尔卡特公司

地址 法国巴黎市

[72] 发明人 塞尔日·巴库 马蒂厄·科德巴尔德

[74] 专利代理机构 北京市金杜律师事务所

代理人 鄧 迅

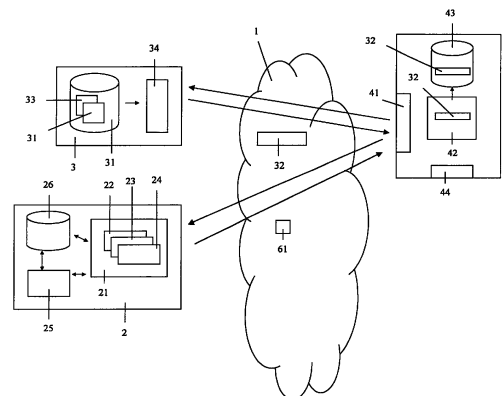
权利要求书 2 页 说明书 12 页 附图 3 页

[54] 发明名称

客户端计算机显示数据的方法

[57] 摘要

本发明涉及用于在客户端计算机上显示数据的方法，以及用于在客户端计算机上显示数据的计算机可执行软件代码和客户端计算机。客户端计算机获取包括图形用户界面的静态数据和至少一个用于包括网页动态数据的嵌入的网页，其中所述嵌入指向位于网络服务器上并且适于输出脚本的程序。客户端计算机将网页存储在高速缓存中。将包括由程序所产生的动态数据的脚本从网络服务器传送到客户端计算机。客户端计算机的浏览器通过包括在所获取的脚本中的动态数据将动态部分包括到网页的静态部分中。



1. 一种在客户端计算机上显示数据的方法，该方法包括以下步骤：
步骤：

5 从 web 服务器发送网页特别是 HTML 页面到该客户端计算机，该网页包含图形用户界面的静态数据和至少一个用于包括该网页动态部分的嵌入，其中所述嵌入指向位于网络服务器的程序并且适于输出脚本；

将该网页存储在该客户端计算机的高速缓存中；

10 将该脚本从该网络服务器传送到该客户端计算机，该脚本包含由该程序产生的动态数据；

通过所述动态数据将该动态部分包括到该网页的静态部分中；以及

通过在该客户端计算机上执行的浏览器显示该网页。

15 2. 根据权利要求 1 所述的方法，其中：

该客户端计算机执行以后可能需要的文件的后台预先装载机制，其中该预先装载机制选择、下载可能需要的文件并将其存储到客户端计算机的高速缓存中。

3. 根据权利要求 2 所述的方法，其中：

20 该方法还包括以下步骤：建立指示可能需要的后续网页的图形，该图形代表访问这些网页的概率；优选具有最高概率的分支对该图形进行游历；并且预先装载与在该游历过程中所选择的网页的相关联文件。

4. 根据权利要求 2 或者 3 所述的方法，其中

25 该方法还包括以下步骤：在该网页中包括用于隐藏框的脚本，该脚本适于执行网页特定的预先装载机制；用该脚本下载和高速缓存该网页；以及在该浏览器的该隐藏框内在后台执行该预先装载机制。

5. 根据权利要求 1 所述的方法，其中：

该方法还包括以下步骤：在该网页中包括脚本，该脚本适于执行

自动填充过程；并且通过该自动填充过程，使用由该嵌入提供的对应的动态数据填充网页中的域。

6. 根据权利要求1所述的方法，其中：

该方法还包括以下步骤：在该网络服务器上包括一个或者多个代表该脚本中动态数据的 javascript 变量；将该网页的 HTML 表单域命名为和 javascript 变量相同的名字；并且自动地在该客户端计算机上由包含在所获取的脚本中的对应 javascript 变量来填充 HTML 表单域。

7. 根据权利要求1所述的方法，其中：

该客户端计算机执行函数，该函数读取当前的查询串，将文件路径与该获取的查询串进行链接，以及动态地用该链接的结果构建用于包括该网页动态部分的嵌入。

8. 根据权利要求1所述的方法，其中：

该客户端计算机执行函数，该函数自动产生随机数字并且将该随机数字添加到该嵌入的查询串中。

9. 用于在客户端计算机上显示数据的计算机可执行软件代码，该代码包括：将从 web 服务器发送到客户端计算机的代码，该代码指定一个网页，特别是 HTML 页面，其中该网页包含图形用户界面的静态数据和至少一个用于包括该页面的动态部分的嵌入，其中所述嵌入指向位于网络服务器上并且适于输出脚本的程序；以及用于通过由该程序产生的并且通过该脚本从该网络服务器传送到该客户端计算机的动态数据将该动态部分包括到该网页的静态部分中的代码。

10. 一种客户端计算机，包括：用于从 web 服务器获取网页的通信单元；用于存放网页特别是 HTML 页面的存储器，该网页包括图形用户界面的静态数据以及至少一个用于包括该网页动态部分的嵌入，其中所述嵌入指向位于网络服务器上并且适于输出脚本的程序，该脚本包含由该程序产生的动态数据；以及浏览器，其适于将从该 web 服务器获取的该网页存储在高速缓存中，通过由该脚本传送的所述动态数据将该动态部分包括到该网页的静态部分中，并且显示该网页。

客户端计算机显示数据的方法

5 技术领域

本发明涉及在客户端计算机上显示数据的方法，用于在客户端计算机上显示数据的计算机可执行软件代码以及客户端计算机。本发明基于在先申请 EP 04 292 833.3，通过参考将其引入于此。

10 背景技术

因特网上的 web 服务器提供通常以 HTML（超文本标记语言）编码的具有图形网页的网站，其中连接到因特网的或者通过本地网络，例如局域网（LAN）与 web 服务器连接的客户端计算机可以访问这些网站。客户端计算机使用允许用户下载网页以及在客户端计算机上显示网页的浏览器。

基于 web 的用户界面通常是缓慢的。当 web 服务器和客户端计算机之间的因特网连接很弱（例如 64kB/s），用户数目巨大，提供页面的服务器配有较小的内存和/或 CPU 资源或者客户端设备很小，例如是 PDA（个人数字助理）、商务电话、移动电话或者智能电话时，该界面就更加缓慢了。

为了加快基于 web 的用户界面的响应时间，已知在客户端侧（浏览器高速缓存）或者中间代理（客户端和服务端之间的代理）上使用高速缓存技术。但是，高速缓存仅仅在已经下载了相同的页面时加速页面的下载。

该网页可以具有保持不变的静态数据和在网页上变化的动态数据。

例如，JSP 页面（JSP=Java 服务器端页面）使将网页的动态部分从静态 HTML 分离成为可能。人们简单地以通常的方式编写常规的 HTML 并且在特定的标签中包括用于动态数据的代码。人们所写的通

常看起来更像常规的 HTML 文件而不是小服务程序 (servlet)，然后在后台，JSP 页面才转换成通常的小服务程序。这通常在该页面被第一次请求时完成。小服务程序输出与小服务程序服务方法相关联的静态 HTML。这种页面不总是相同的，使得人们在客户端侧或者代理上都无法对其进行高速缓存。利用 JSP 技术，JSP 页面无法高速缓存并且该页面由小服务程序容器进行分析，这占用服务器的时间和资源。这种问题同样存在于其它种类的动态网页上，例如 ASP、PHP 或者 SSI 页面 (ASP = 动态服务器页面；PHP = 超文本预处理器；SSI = 服务器端包含)。

10 当网页包含动态数据时，网页必须周期性地刷新或者重新载入，以正确地反映任何变动的数据。浏览器必须持续地下载网页数据并更新已经下载的网页。一些浏览器试图通过在将网页下载到浏览器时使用高速缓存方案来存储 HTML 网页以提高效率。当网页必须被刷新时，浏览器重新下载整个网页，接受那些在已经在高速缓存中的在网页上使用的文件。如果请求刷新动态数据，则传统的浏览器简单地重新载入整个网页，但是该整个的重新载入是缓慢的并且占用大量系统资源。另一个在浏览器上刷新网页的方法是在网页上包括刷新小

15 程序，其指示浏览器以预先确定的时间间隔更新网页。

但是，所有这些方法在更新下载到浏览器上的网页时都不区分网站上的静态和动态数据。

20

发明内容

本发明的目标是加速基于 web 的界面的响应时间。

本发明的目标通过在客户端计算机上显示数据的方法来实现，该方法包括以下步骤：从 web 服务器发送网页特别是 HTML 页面到客户端计算机，该网页包括图形用户界面的静态数据以及至少一个用于包括网页动态部分的嵌入，其中所述嵌入指向位于网络服务器上并且适于输出脚本的程序；将网页存储在客户端计算机的高速缓存中；将脚本从网络服务器传送到客户端计算机，该脚本包括由该程序产生的

25

动态数据；通过动态数据将动态部分包括到网页的静态部分中；以及通过在客户端计算机上执行的浏览器显示该网页。本发明的目标进一步由用于在客户端计算机上显示数据的计算机可执行软件代码来实现，该代码包括：该代码包括：将从 web 服务器发送到客户端计算机的代码，该代码指定了一个网页，特别是 HTML 页面，其中该网页包含图形用户界面的静态数据和至少一个用于包括该网页的动态部分的嵌入，其中所述嵌入指向位于网络服务器上并且适于输出脚本的程序；以及用于通过由该程序产生的并且通过该脚本从该网络服务器传送到该客户端计算机的动态数据将该动态部分包括到该网页的静态部分中的代码。本发明的目标进一步通过客户端计算机来实现，其包括：用于从 web 服务器获取网页的通信单元；用于存放网页特别是 HTML 页面的存储器，该网页包括图形用户界面的静态数据以及至少一个用于包括该网页动态部分的嵌入，其中所述嵌入指向位于网络服务器上并且适于输出脚本的程序，该脚本包含由该程序产生的动态数据；以及浏览器，其适于将从该 web 服务器获取的该网页存储在高速缓存中，通过由该脚本传送的所述动态数据将该动态部分包括到该网页的静态部分中，并且显示该网页。

本发明使得将静态部分，即静态 GUI（图形用户界面）从动态部分（动态数据）中完全分离出来成为可能，并且允许最大化的使用经典的高速缓存技术以减少将在客户端和服务器之间传送的数据量。其还从用户体验的角度提高了速度感。

在因特网上通常的数据库驱动的/数据集中的网页应用程序中，50%到 90%的网页内容是静态信息。结合最大化使用经典的高速缓存技术获得的优势，本发明所达到的时间响应优势在 35%到 75%之间。本发明提高了响应时间、高速缓存的利用，并且允许静态 GUI 和动态数据之间的分离，从而引入了可以由不同开发者完成工作的决策模型。

进一步，本发明使得将图形用户界面的静态数据和动态数据托管在因特网上的不同服务器上成为可能。优选地，静态数据，即 HTML

网页，和动态数据，即服务器端程序，例如小服务程序，由不同类型的适用于各自数据的特定要求的服务器基础设施来处理。例如，通过 web 服务基础设施的不同地理分布服务器托管 HTML 网页，而通过专用的应用程序服务器托管小服务程序。这种方法提高了系统的速度
5 和可靠性。

进一步的优点由附属权利要求指示的本发明的实施例来实现。

嵌入由一个标准嵌入来完成，例如，标准 javascript 嵌入，但是 URL（统一资源定位符）不是一个 javascript 文件而是将脚本输出到其标准输出上的程序（ASP，JSP，PHP，小服务程序，C/C++程序或者任何语言编写的程序）
10

优选地，网页是指定图形用户界面（GUI）的静态部分并且具有一个 javascript 嵌入集合的 HTML 页面。HTML 页面编码了包括窗口小部件、文本、动态行为（表示“如果...那么将禁用该按钮”的函数）的图形用户界面。如果该 GUI 被设计得很好，则其可以完全独立于
15 传到该 GUI 的数据。因此，可以说该 GUI 是“静态的”。动态数据包括可以实例化 GUI 以填充域的信息。基于网页的 javascript 嵌入的嵌入机制将动态部分包括到静态部分中而不影响高速缓存（代理或者客户端侧）机制。

根据本发明的优选实施例，客户端计算机执行自动使用由嵌入提供的数据以填充域的自动数据填充机制。下载到客户端计算机的网页
20 包含适于执行该自动填充过程的脚本。客户端计算机使用该自动数据填充过程来填充网页域，即利用由嵌入所提供的对应动态数据填充 HTML 域。

动态部分可以总结为变量赋值，例如，javascript 变量集合。优选地，存在一个命名惯例要求 HTML 表单域和那些 javascript 变量取
25 相同的名字。网络服务器包括一个或者多个代表 javascript 中动态数据的 javascript 变量。客户端计算机使用包含在获取的脚本中的对应 javascript 变量自动地填充具有与这些 javascript 变量相同名字的 HTML 表单域。这个过程提供了强大的机制将动态部分包括进静态部

分而不影响高速缓存机制。

根据本发明的优选实施例，客户端计算机执行函数，该函数读取当前的查询串，用获取的查询串计算文件路径并且用链接的结果动态地构建用于包括网页动态部分的嵌入。

- 5 该方案解决了查询串发布问题。例如，人们想要描述一个列出用户的 GUI。为了访问第五个用户，URL 可以是：

`http://myServer/user.html?id=5`

考虑 `getUser` 是为给定用户提供数据的程序。`user.html` 文件是包括使用下列指令的动态数据的 javascript 嵌入的静态 HTML 文件。

- 10 `<script language = "javascript" src = "/servlets/getUser"></script>`

这里的问题是 `getUser` 程序不知道需要显示哪个用户。必须告诉 `getUser` 程序想要第五个用户的数据。换句话说，想要将原始的查询串（在原始 URL 中的“？”字符之后的所有字符）告知在 javascript 中调用的程序。这必须通过使用称为 `insertJSFile()` 的 javascript 函数而不是使用 `<script>` 标签来完成，如下例所示：

- 15 `<script>insertJSFile ("/servlets/getUser");</script>`

`insertJSFile(filePath)` 函数读取当前的查询串，将该 `filePath` 和获取的查询串链接起来，并且用链接的结果来编写 `<script>` 标签。

例如，该函数将动态的构建下列标签：

- 20 `<script language = "javascript" src= "/servlets/getUser?id=5"></script>`

根据本发明的另一个优选的实施例，客户端计算机在其上执行能够自动产生随机数字并且将其加入到该嵌入的查询串中的函数。该方法防止对程序的脚本输出进行高速缓存。当参照程序使用 `<script>` 标签时，一些浏览器将不会正确地工作。例如，Internet Explorer 会高速缓存程序的 javascript 输出，这是内存空间的一种浪费，因为它们是不必被高速缓存的动态数据。因为高速缓存了动态数据，浏览器将总是会返回相同的数据（高速缓存中的信息），即使数据已经改变了。函数自动地产生随机化的数字并且将其加到查询串的末端，使得浏览器将永远不能高速缓存该动态数据，因为浏览器使用完整的 URL 和

25

查询串一起来识别要高速缓存的文件。

根据本发明的一个优选实施例，客户端计算机执行对下一个文件的后台预先装载。客户端计算机执行对于以后可能需要的文件的后台预先装载机制。预先装载机制选择、下载可能需要的文件并将其存储到客户端计算机的高速缓存中。利用这种后台预先装载，用户甚至在低带宽下也会体验到非常小的响应时间。当装载一个网页时，用户通常5 需要思考一会儿，然后才使用他的输入设备（键盘、鼠标、PDA 笔等）执行一些动作。用户的思考时间和对键盘、鼠标或者任何输入设备的使用被用来在后台下载用户可能使用的下一个文件。优选地，使用隐藏框在后台完成预先装载，使得在需要之前用户将不会看见该10 网页。优选地，从 web 服务器下载的网页具有用于隐藏框的脚本，并且该脚本适于执行网页特定的预先装载机制，并且客户端计算机在后台在浏览器的隐藏框内执行该网页特定的预先装载机制。

优选地，下一个文件利用下面的算法进行预先装载：

15 建立用来指示可能需要的后续网页的图形，其中该图形代表访问这些页面的概率。游历过程优选那些具有最高概率的分支对图形进行游历。游历过程选择一些网页。在隐藏框中，在后台下载每一个选择的网页。在下载期间，相关联的文件（例如，图像，CSS = 层叠样式20 表单，javascripts）也通过使用网页中的超链接引用自动地下载。所有文件放置在浏览器的高速缓存中，使得当用户传送后续的页面时，所有静态的东西都会从高速缓存中下载，即使用户第一次看见该页面。只有动态数据将会在网络上下载。

附图说明

25 通过结合附图阅读下面的当前优选示例性实施例的详细描述将会更好地理解本发明的这些和其它特征和优点，其中：

图 1 是通过通信网络互连的服务器和客户端计算机的框图，框图说明了对网页的第一次访问。

图 2 是根据图 1 的服务器和客户端计算机的框图，其中框图说明

了对该网页的第二次访问。

图 3 是举例说明根据本发明的执行方法的功能性视图。

具体实施方式

5 图 1 表示了通信网络 1、两个服务器 2 和 3 以及客户端计算机 4。

通信网络 1 是使用 IP 协议 (IP = 因特网协议) 作为公共第三通信层的数据网络。通信网络 1 可以由各种不同的物理网络, 例如, ATM 网络、MPLS 网络或者以太网 (ATM = 异步传输模式, MPLS = 多协议标签交换) 组成。进一步, 客户端计算机 4 可以通过无线接入网络, 10 例如 GSM 网络、UMTS 网络或者 WLAN 网络 (GSM = 全球移动通信系统, UMTS = 通用移动通信系统) 与服务器 2 和服务器 3 连接。

优选地, 通信网络 1 是因特网。

服务器 3 是在通信网络 1 中提供网页的 web 服务器。从功能的角度来看, 服务器 3 具有用于存储多个网页, 例如网页 32 和网页 33 15 的数据库 31, 以及适于将这种网页下载到请求的客户端计算机的控制单元 34。

优选地, 服务器 3 代表了具有各种本地分布的 web 服务器的服务器基础设施, 其中 web 服务器在各自的专用的区域提供存储在数据库 31 中的网页的内容 (即根据获取各自网页的客户端计算机的位置将各自网页的 URL (统一资源定位符) 映射到不同的 web 服务器)。20 这种服务系统加速了服务器 3 上托管的网页的下载。

服务器 2 是服务特定的基于 web 的应用程序的网络服务器, 例如, 因特网支付服务。从功能的角度来看, 服务器 2 包括提供基于 web 的应用程序之动态数据的程序容器 21、数据库 26 以及, 可选地, 25 向容器 21 的程序提供服务的另一些程序的集合 25。将容器 21 的程序编码成 ASP、JSP、PHP、小程序、C/C++ 程序或者其它任何语言的程序 (ASP = 动态服务器页面; JSP = Java 服务器页面; PHP = PHP 超文本预处理器)。通过通信网络 1 经由分配的 URL 对这些程序进行寻址并可进行访问。程序输出包含基于 web 的应用程序之动

态数据的脚本作为标准输出。容器 21 的程序访问由数据库 26 提供的
数据源和程序 25 的服务，以产生基于 web 的应用程序的动态数据。
进一步，容器 21 的程序可以作为代理对象的集合，例如，java-beans，
来与由程序 25 提供的服务逻辑进行接口连接。

- 5 优选地，容器 21 是小服务程序容器并且容器 21 的程序是 java
小服务程序。

进一步，可以由相同的物理服务器或者服务基础设施提供服务器
2 和服务器 3 的功能，即网页和容器 21 的程序托管于同一个服务器。

- 10 客户端计算机 4 是个人计算机、PDA(个人数字助理)或者电话，
例如智能电话或者其它任何种类的增强蜂窝电话。

- 15 客户端计算机 4 包括具有至少一个微处理器和数据存储的电路，
由至少一个微处理器执行的软件代码，用于通过通信网络 1 通信的通
信设备以及多种输入/输出设备，例如液晶显示器、小键盘、麦克风、
扩音器或者鼠标板。客户端计算机 4 的功能是通过基于由客户端计算
机 4 的组件提供的硬件平台执行软件代码来实现的。从功能的角度来
看，客户端计算机 4 包括通信单元 41、高速缓存 43、浏览器 42 和输
出单元 44。

- 20 通信单元 41 包括通过通信网络 1 进行通信所需要的客户端计算
机 4 的物理和逻辑组件。例如，通信单元 41 包含处理 TCP/IP 协议栈
(TCP = 传输控制协议，IP = 因特网协议)的必要功能。浏览器 4 是
标准的浏览器应用程序，例如，Internet Explorer。输出单元 44 代表
了浏览器 42 使用的输出设备，例如，液晶显示器或者监视器和扩音
器。

- 25 当生成基于 web 的应用程序时，用户界面被编码成两个独立的
部分：

第一，静态图形用户界面 (GUI)，其包括网页的集合，其中网
页包括窗口小部件、文本、动态行为 (例如表示“如果...那么将禁用
该按钮”的函数)。例如，网页 32 和网页 33 指定了 GUI 的静态部分。

第二，动态部分，其包括用来实例化 GUI 的动态数据。动态部

分通过容器 21 的程序来编码，例如，通过程序 22、23 和 24。

将网页 32 和网页 33 编码成可以被浏览器 42 高速缓存的标准的 HTML 页面。这些 HTML 页面包含嵌入，即将文件装载进网页的函数，用来包括网页的动态部分。嵌入的 URL 不是一个文件而是在其
5 标准输出输出脚本的程序 22 到 24 中的一个。优选地，嵌入是标准的 javascript 嵌入，而 URL 不是 javascript 文件而是程序 22 到 24 中的一个。例如，网页 32 具有下面的嵌入指令：

```
<script language = "javascript" src = "/servlets/myServlet"></script>
```

在此例中，程序 “myServlet” 会显示下面的 javascript 指令：

```
10 <script>  
    var firstname = "John"  
    var lastname = "Woo"  
    var authoring = true;  
</script>
```

15 用户第一次请求网页 32 时，浏览器 42 通过通信网络 1 发送相应的请求消息。该请求消息路由到 web 服务器 3。web 服务器 3 通过通信网络 1 将所请求的网页 32 下载到浏览器 42，其中网页 32 例如是 DHML 页面（DHTML = 动态超文本标记语言）。如先前示例，网页 32 包含图形用户界面的静态数据以及至少一个用于包括网页的动态
20 部分的嵌入。该至少一个嵌入指向位于网络服务器 2 的程序并且适于输出脚本。浏览器 42 将网页 32 存储在高速缓存 32 中。进一步，其执行网页 32 的嵌入，联系服务器 2 中的相应的程序 22 到 24，并且从这些包含了将要在网页 32 中包括的动态数据的程序中获取脚本。例如，浏览器 42 获取 javascript 61，其包含了程序 “myServlet” 的上述指定的输出。现在，网页的静态部分必须使用动态部分，并且浏览器
25 必须将接收的动态数据包括进网页 32。这是通过所谓的“自动数据填充”机制来完成的。这个机制自动地将包括在获取的脚本中的动态数据填充到相应的 HTML 表单域中，例如，textInput、checkbox、droplist 等。例如，脚本 61 中的值 “firstname”，“lastname” 和 “authoring”

填充在网页 32 的相应 HTML 表单域中。

优选地，存在一个命名惯例，其中网页 32 和网页 33 的 HTML 表单域和那些由程序 22、23 和 24 输出的 javascript 变量取相同的名字。泛型代码（generic code）存储在 javascript 库中，其中 javascript 库包括在网页 32 和网页 33 的所有文件中。该泛型代码用来自动地将所获取的脚本的 javascript 变量填充到网页 32 和网页 33 的合适的域中。在网页 32 的下载期间，通过使用网页 32 中的超链接引用，相关联的文件（图像、CSS、javascript 等）也被自动下载（从服务器 3）并且存储在高速缓存 43 中。

10 网页 32 的第二个请求中，所有静态的东西已经在本地可用并且从高速缓存 43 中下载。只有动态数据必须通过通信网络 1 下载。这在下面通过图 2 的辅助来举例说明。

图 2 示出了通信网络 1、服务器 2 和服务器 3 以及客户端计算机 4。网页 32 和它们相关联的文件存储在客户端计算机 4 的高速缓存 15 43 中。当用户或者客户端计算机 4 的刷新函数或者用户请求访问网页 32 时，网页 32 和其相关联的文件被自动地从高速缓存 43 中下载到浏览器 42。只要浏览器对导向服务器 2 的程序的网页 32 的嵌入进行处理，浏览器就联系该程序并且获取脚本，例如脚本 62，其包含有将要包括进静态 GUI 的更新的动态数据。

20 下面，通过图 3 的辅助来举例说明本发明进一步的细节。

图 3 示出了具有网页 32 和网页 33 的数据库 31、浏览器 42、高速缓存 43 以及具有程序 22、23 和 24 的容器 21。

用户第一次请求网页 33 时，服务器 3 将网页 33 下载到浏览器 42。网页 33 是 DHTML 页面，具有多个指定静态数据的 HTML 标签和指向容器 21 中的程序的嵌入。例如，网页 33 包含标签 72 和标签 25 74 以及嵌入 71 和嵌入 73，其中标签 72 和标签 74 包含这种静态数据，而嵌入 71 和嵌入 73 指向容器 21 中的程序。

进一步，网页 33 包括具有程序代码的脚本 73，由浏览器 42 执行该脚本时，嵌入 71 和嵌入 73 提供的动态数据自动地填充标签 72

和标签 74 的 HTML 域。

进一步，网页 33 包括用于网页 33 中的隐藏框的脚本 76。脚本 76 包含用于网页特定的预先装载机制的数据和代码，其中网页特定的预先装载机制在由浏览器 42 执行时会选择、下载可能需要的文件并将其存储到客户端计算机 4 的高速缓存 43 中。

因此，网页 33 包含指定图形用户界面的静态数据即文本 72 和文本 74 的代码；用于包括网页的动态部分的嵌入 71 和嵌入 73；以及代码，即脚本 75，用于通过动态数据将动态部分包括进网页的静态部分，其中动态数据由程序 22 到程序 24 产生并且通过脚本从网络服务器 2 传送到客户端计算机 4。

当第一次下载网页 33 到浏览器 42 时，浏览器 42 将网页 33 和由网页 33 中的超链接引用包括的相关联文件存储到高速缓存 43 中。进一步，浏览器在后台在隐藏框 45 之内执行脚本 76 中指定的预先装载机制。预先装载机制预先装载以后可能需要的文件。将可能的下一个网页放置在一个图形中，其中访问该页面的概率与该图形的每一个分支相关联。游历过程优选那些具有最高概率的分支对图形进行游历。例如，在登录页面之后的下一个页面 90% 是主页并且 10% 是错误页面。进一步指定用户首选项的用户特定的配置文件可以用来建立或者修改图形。进一步，游历过程选择地考虑用户输入或者其它种类的外部数据并且其会影响由后台预先装载机制完成的选择。

预先装载在后台在隐藏框 45 之内完成。在后台下载由游历过程选择的每一个网页，并且因此自动地将其存储在高速缓存 43 中。例如，下载网页 81、82 和 83 并且将其存储到高速缓存 43 中。在下载期间，还自动地下载通过超链接引用包括在所选择的网页中的相关联文件（图像、CSS 和 javascript），并将其存储在高速缓存 43 中。当可能需要的文本文件的所有或者大多数相关部分已经存储到高速缓存中时，很有可能当用户点击进入下一个页面时，所有静态的东西会从高速缓存中下载，即使用户是第一次见该页面。

在可见框 46 中，当显示网页 33 时浏览器 42 执行网页 33 的嵌入

71 和嵌入 73，联系容器 21 中的相关联程序并且获取脚本 91 和脚本 92，其中脚本 91 和脚本 92 包括必须要被包括进 GUI 静态部分中的 GUI 的动态数据。网页 33 的脚本 75 提供自动将包括在脚本 91 和脚本 92 中的动态数据填充到标签 72 和标签 74 的合适 HTML 表单域中的泛型代码。

5 优选地，脚本 75 包含具有额外函数的代码，其读取当前查询串，将文件路径和获取的查询串链接起来，并且动态地用链接的结果构建用于包括网页动态部分的嵌入。优选地，该函数会进一步自动地产生随机化数字，并且将其添加到查询串，使得浏览器 42 将不会高速缓存动态数据。

10 例如，如果人们想要描述用于列出用户的 GUI，可访问第五个用户并且 getUser 会是为给定用户提供数据的程序，insertJSFile() 函数将为“/servlets/getUser”构建下面的 HTML 指令：

```
<script language = “ javascript”  
15 src = “/servlets/getUser?id=5&rand=345101615109701”></script>。
```

由函数 insertJSFile() 创建的该 HTML 指令在由浏览器 42 执行时告诉 getUser 程序需要第五个用户的数据，并且包含用来防止高速缓存动态数据的随机数字。

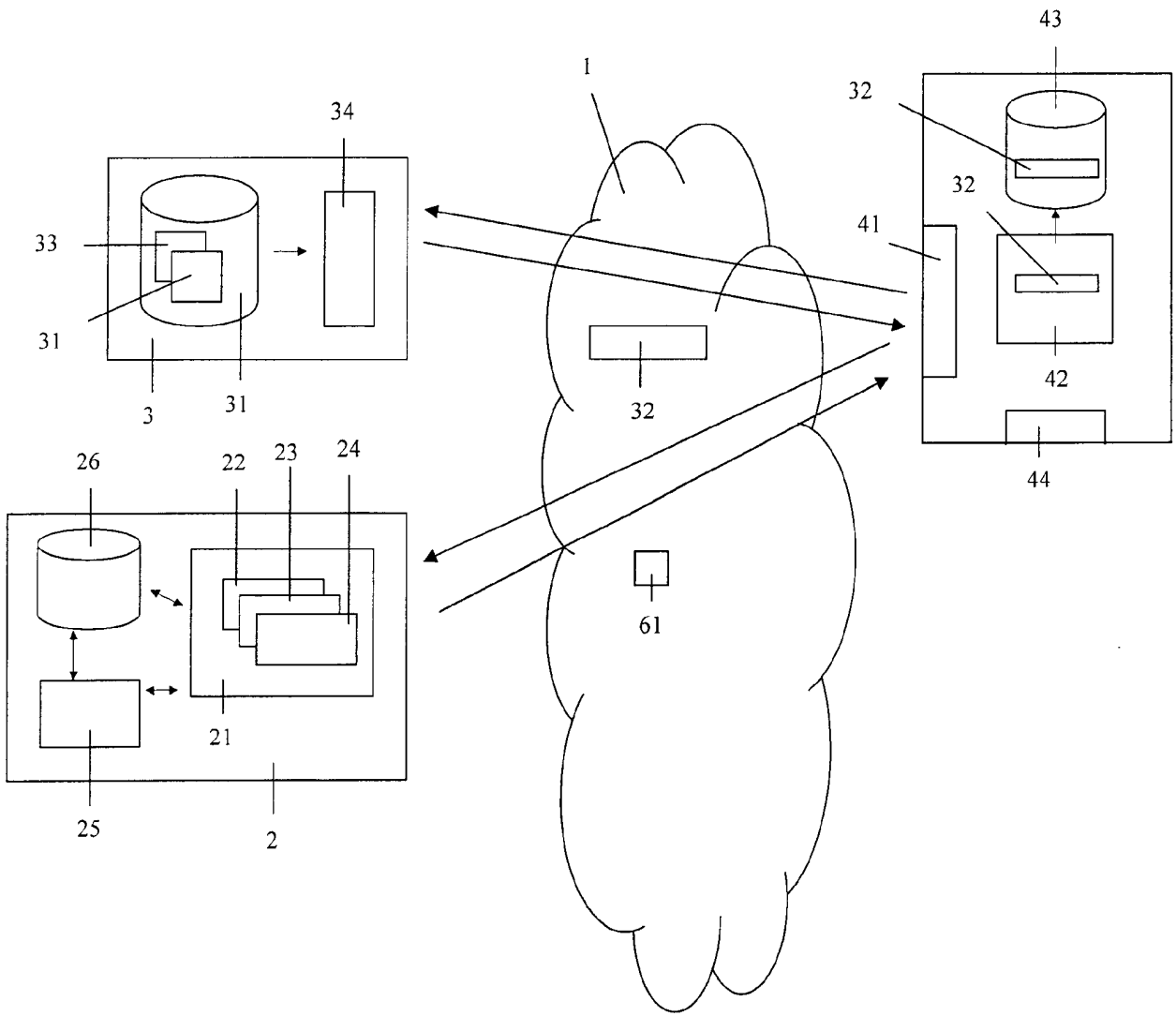


图 1

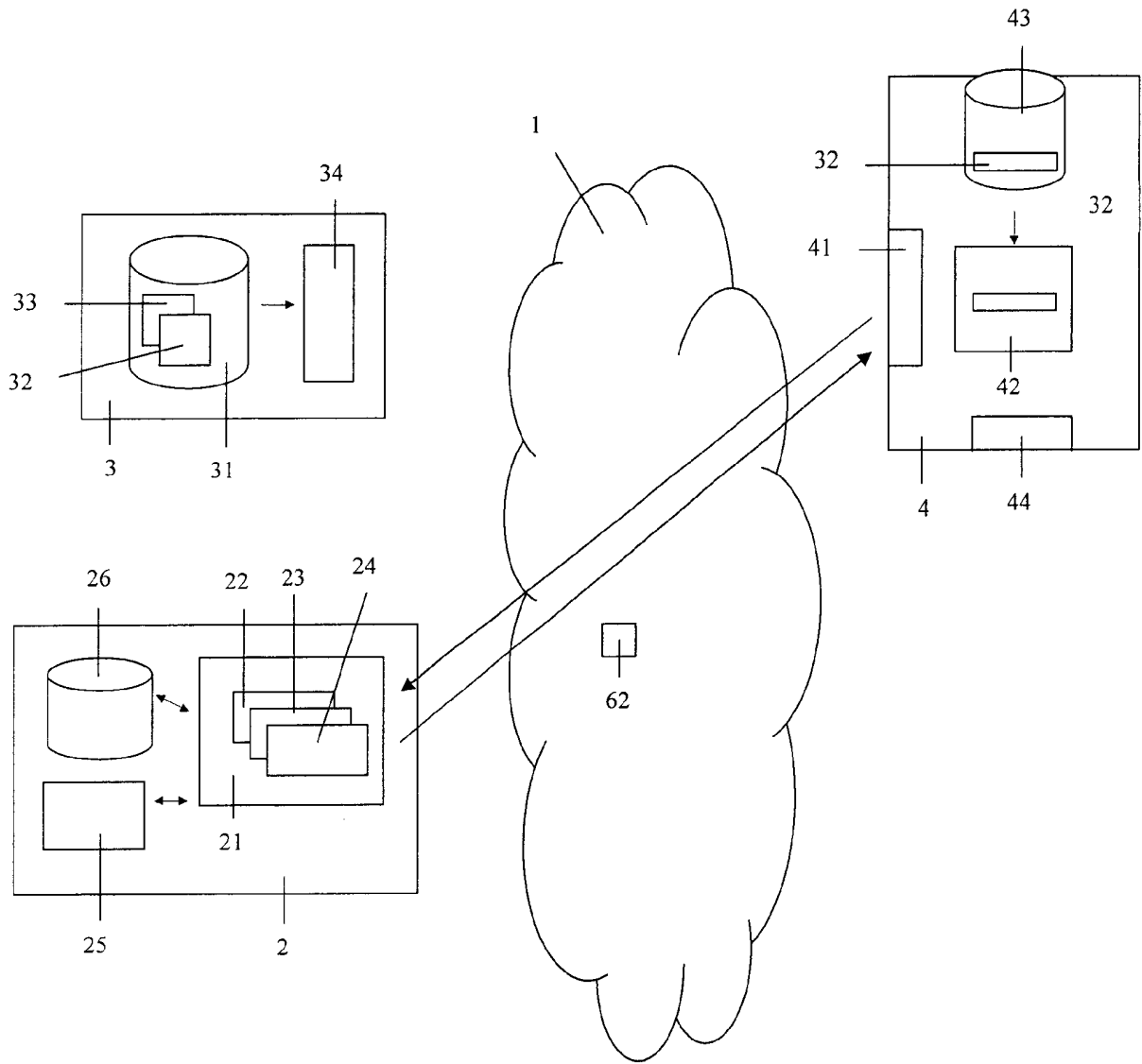


图 2

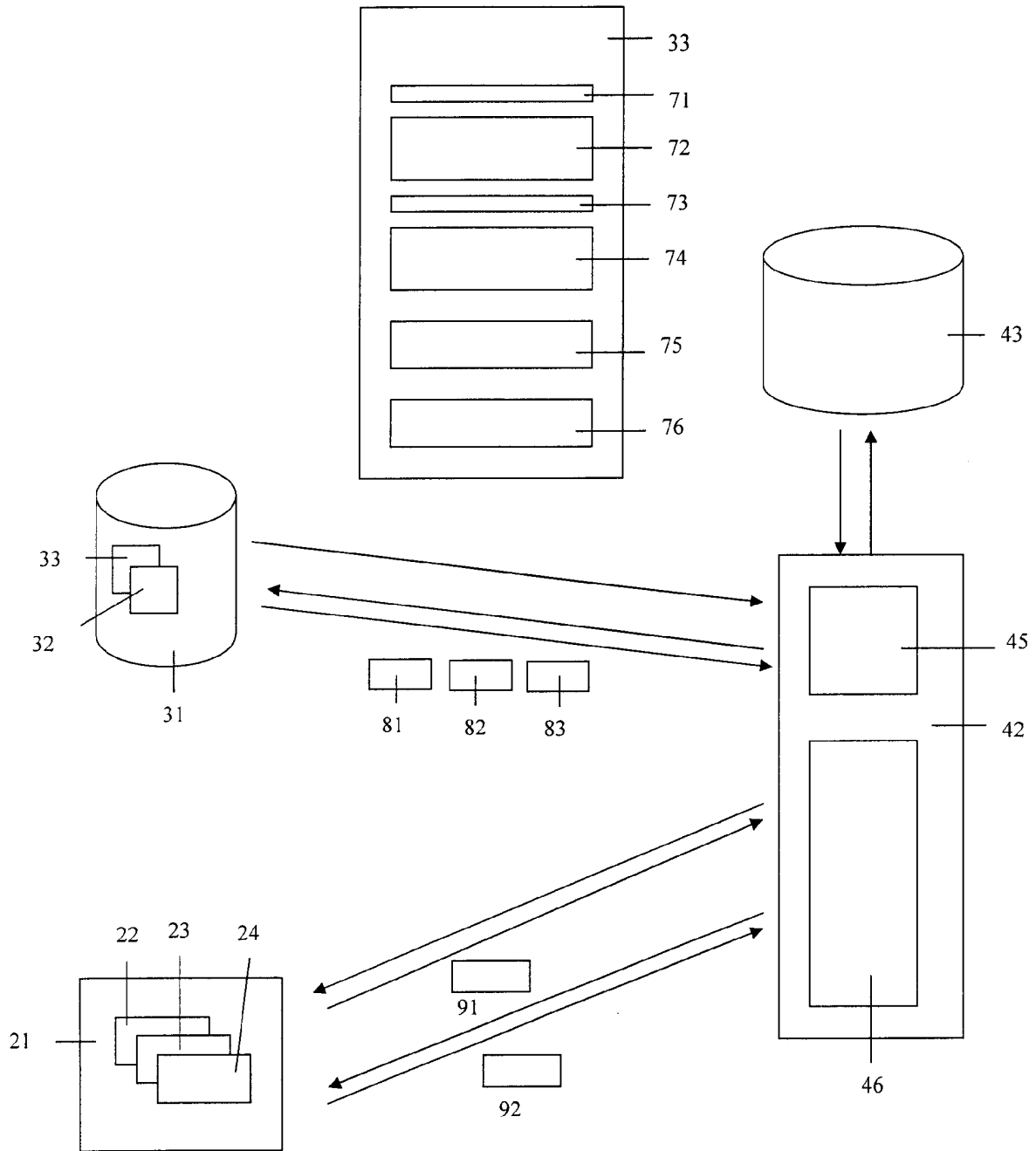


图 3