

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 15/16 (2006.01)



[12] 发明专利说明书

专利号 ZL 00814155. X

[45] 授权公告日 2006 年 12 月 6 日

[11] 授权公告号 CN 1288575C

[22] 申请日 2000.9.22 [21] 申请号 00814155. X

[30] 优先权

[32] 1999. 9. 24 [33] US [31] 09/405,608

[86] 国际申请 PCT/US2000/026097 2000.9.22

[87] 国际公布 WO2001/022195 英 2001.3.29

[85] 进入国家阶段日期 2002.4.11

[73] 专利权人 阿坎巴公司

地址 美国加利福尼亚州

[72] 发明人 杰克·J·史密斯

理查德·T·伯赖特

W·斯潘塞·沃利第三

约翰·B·麦克唐奈

约翰·A·瓦斯塔诺

威廉·T·韦瑟福德

审查员 王学睿

[74] 专利代理机构 北京市柳沈律师事务所

代理人 马莹 邵亚丽

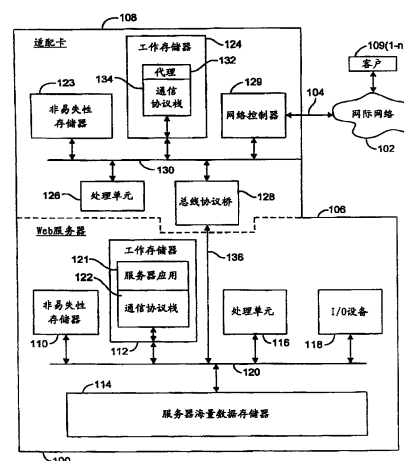
权利要求书 3 页 说明书 11 页 附图 8 页

[54] 发明名称

用于管理客户与服务器之间的连接的系统和
方法

[57] 摘要

一种用于管理客户(109)和网络服务器(106)之间的连接的适配卡(108), 将连接管理的负担从服务器(106)转移出来。适配卡(108)包括带嵌入式代理应用(132)和通信协议栈(134)的存储器(124)、用于执行应用代码的处理单元(126)、用于与网际网络(102)接口的网络控制器(129)和用于与网络服务器(106)的内部总线(120)接口的总线协议桥(128)。代理应用(132)通过相对慢且不可靠的网络连接代表服务器(106)接收客户请求, 并且通过快且可靠的总线连接将该请求提交给服务器。



1. 在可操作地连接到源服务器的内部总线的接口设备中,一种用于管理至少一个客户机和所述源服务器之间的连接的方法,所述方法包括如下步骤:

经由网络与所述客户机中的一个客户机建立网络连接;

经由所述网络连接从所述客户机接收通信;

经由所述源服务器的内部总线,与所述源服务器建立总线连接;以及

经由所述总线连接将所述客户机通信转发到所述源服务器上。

2. 如权利要求 1 所述的方法,其中,从所述客户机接收通信的所述步骤包括在缓冲区中存储所述通信。

3. 如权利要求 2 所述的方法,其中,在缓冲区中存储所述通信的所述步骤包括在所述缓冲区中累积一个或多个来自所述客户机的单独传输。

4. 如权利要求 3 所述的方法,其中,与所述服务器建立总线连接的所述步骤包括在与所述服务器建立所述总线连接之前,等待直到在所述缓冲区中累积出一个完整的客户机请求。

5. 如权利要求 4 所述的方法,进一步包括:

利用所述总线连接从所述服务器接收对所述客户机通信的响应;和

利用所述网络连接将所述响应转发给所述客户机。

6. 如权利要求 5 所述的方法,其中,从所述服务器接收所述响应的所述步骤包括在缓冲区中存储所述响应。

7. 如权利要求 6 所述的方法,其中,从所述服务器接收所述响应的所述步骤包括在接收到所述响应之后终止所述总线连接。

8. 如权利要求 1 所述的方法,进一步包括:

利用所述总线连接从所述服务器接收对所述客户机通信的响应;和

利用所述网络连接将所述响应转发给所述客户机。

9. 如权利要求 8 所述的方法,其中,从所述服务器接收所述响应的所述步骤包括在缓冲区中存储所述响应。

10. 如权利要求 9 所述的方法,其中,从所述服务器接收所述响应的所述步骤包括在接收到所述响应之后终止所述总线连接。

11. 如权利要求 8 所述的方法,其中,所述客户机通信包括 HTTP 请求。

12. 如权利要求 11 所述的方法,其中,来自所述服务器的所述响应包括

HTTP 页面。

13. 如权利要求 1 所述的方法, 其中, 所述与客户机建立网络连接的步骤包括利用所述网络与多个客户机中的每个建立单独的网络连接。

14. 如权利要求 13 所述的方法, 其中, 与所述服务器建立所述总线连接的所述步骤包括利用所述服务器的所述内部总线与所述服务器建立多个连接。

15. 如权利要求 14 所述的方法, 其中, 最大同时客户机连接数超过最大同时服务器连接数。

16. 如权利要求 1 所述的方法, 进一步包括在将所述客户机通信转发给所述服务器之前, 对所述客户机执行安全操作。

17. 如权利要求 1 所述的方法, 其中:
接收所述客户机通信的所述步骤包括从所述客户机通信辨别应用标识符; 和
将所述客户机通信转发给所述服务器的所述步骤包括根据所述应用标识符调用多个代理应用中的一个。

18. 如权利要求 17 所述的方法, 其中, 所述应用标识符为连接端口号。

19. 如权利要求 1 所述的方法, 其中, 接收所述客户机通信的所述步骤包括接收 HTTP 请求的至少一部分。

20. 一种可操作地连接到源服务器的内部总线的适配卡, 用于管理源服务器和网络的通信, 所述适配卡包括:

网络控制器, 用于在所述网络上与客户机进行通信;
存储器设备, 用于存储数据和代码, 所述代码包括反向代理应用;
处理单元, 连接到所述存储器设备, 用于执行所述代码; 和
协议适配器, 连接到所述处理单元, 并且用来连接到所述源服务器的所述内部总线, 以与所述源服务器进行通信。

21. 如权利要求 20 所述的适配卡, 其中, 所述代码进一步包括通信协议栈。

22. 如权利要求 21 所述的适配卡, 其中, 所述通信协议栈包括标准 TCP/IP 协议栈。

23. 如权利要求 20 所述的适配卡, 其中, 所述代理应用包括安全代理。

24. 如权利要求 20 所述的适配卡, 其中, 所述代理应用包括中转代理。

25. 如权利要求 20 所述的适配卡，其中，所述代理应用包括 HTTP 代理。

26. 如权利要求 20 所述的适配卡，进一步包括用于存储从所述客户机接收的数据的数据缓冲区。

27. 如权利要求 20 所述的适配卡，其中，所述代理应用包括一主进程模块，该模块对从所述客户机中的一个接收的连接请求进行响应，并且用来与所述客户机建立连接，并且启动新的客户机进程模块来维护所述所建立连接。

28. 如权利要求 27 所述的适配卡，其中，所述主进程模块进一步用来将所述所建立连接通知给所述代理应用。

用于管理客户与服务器的连接的系统和方法

发明背景

技术领域

本发明一般涉及网络服务器，特别涉及驻留大量客户连接的服务器，更加特别涉及驻留大量相对慢的客户连接的服务器(例如，因特网 web 服务器)。

背景技术

对于网络文件服务器，如因特网 web 服务器，驻留大量相对慢的客户连接，是常见的。只是管理所打开的连接，大量所打开连接就对服务器中央处理器(central processing unit, CPU)施加有沉重的负担。例如，管理在负载服务器上的所打开连接会消耗 CPU 操作量的 30-40% 或更多。该负担大大减少用来执行服务器主要功能，即提供数据给客户的 CPU 周期的百分率。

加在服务器 CPU 上的连接管理负担降低服务器软件例程的性能，并且减少可以同时打开的最大客户连接数。结果，web 驻留公司必须提供额外、冗余的服务器来为数目增加的客户服务。获取并维护额外 web 服务器的成本是相当大的。

代理服务器执行一些客户连接管理功能，并且在本领域内为人所知。然而，在本领域内为人熟知且常见的是，这种代理服务器在位置上必须与服务器进行分开，并且因此必须通过服务器必须管理的，相对慢、易出错的网络连接与服务器进行通信。例如，请参见 Ari Luotonen, Web 代理服务器(Prentice Hall, 1997)，在此加入以作参考。

因此，需要一种用于减轻服务器 CPU 的连接管理负担，从而让服务器更有效地驻留数目增加的客户的系统和方法。

概要

本发明通过提供一种用于管理多个客户和一个服务器之间的连接的系统和方法，克服现有技术相关的问题。本发明帮助将连接管理负担从主机 CPU 转移到介入网络和主机总线之间的适配卡。

适配卡包括网络控制器、存储器设备、处理单元和协议适配器。存储器

设备提供数据和代码的存储。代码包括通过网络控制器在网络上与客户进行通信，并且通过直接连接到服务器总线的协议适配器与服务器进行通信的代理应用。

当由处理单元进行执行时，代理应用通过利用网络建立代理应用与客户之间的网络连接，并且利用服务器总线建立代理应用与服务器之间的总线连接，管理客户连接。另外，存储器设备提供数据缓冲，这允许打开很多与客户的网络连接，而打开相对少的到服务器的总线连接。在具体实施例中，代理从大量慢的客户连接将客户数据累积在缓冲区中，然后通过快总线连接将客户数据提交给服务器。相反，代理通过快总线连接接收服务器数据，临时存储服务器数据，然后通过慢客户连接将服务器数据转发给客户。

在更为特定的实施例中，代码包括由应用代理使用以与客户和服务器进行通信的通信协议栈。在更加特定的实施例中，通信协议栈为传输控制协议/网际协议(Transmission Control Protocol/Internet Protocol, TCP/IP)栈。

在一个实施例中，只有在代理确定已接收完整的客户请求之后，才打开服务器连接。然后在代理从服务器接收对客户请求的响应之后关闭服务器连接。可选地，在系统启动时打开预定数目的持久性服务器连接，并且代理使用这些持久性连接，与服务器进行通信。

代理应用可选地包括很多应用特定代理，包括但不限于 HTTP 代理、安全代理，和/或中转(pass-through)代理。在特定实施例中，代理的主进程模块从客户数据辨别应用标识符(例如，为人熟知的端口号)，并且调用一个或多个与该标识符的值对应的应用特定代理。

附图简述

参照下面附图对本发明进行描述，其中相同的参考号表示相当类似的元素：

图 1 是本发明的服务器和适配卡的方框图；

图 2 是更加详细地示出代理模块的图 1 的适配卡的工作存储器的方框图；

图 3 是更加详细地示出图 2 的代理模块的方框图；

图 4 是示出存储在图 2 的数据缓冲区中的至少一些数据的示例性数据结构

图 5 是概述本发明的一种用于管理客户和服务端之间的连接的方法的流程图；

图 6 是概述一种用于执行图 5 的方法的第一步骤的方法的流程图；

图 7 是概述一种用于执行图 5 的方法的第二步骤的方法的流程图；

图 8 是概述一种用于执行图 5 的方法的第三步骤的方法的流程图；

图 9 是概述一种用于执行图 5 的方法的第四步骤的方法的流程图；

图 10 是概述一种用于执行图 5 的方法的第五步骤的方法的流程图；

图 11 是概述一种用于执行图 5 的方法的第六步骤的方法的流程图。

详细描述

本发明通过使用运行在不同处理单元上的代理应用将大量连接管理负担从服务器的主处理器转移出来，克服现有技术的问题。在下面描述中，对众多特定细节进行阐述(例如，特定通信协议、特定软件和数据结构等)，以彻底理解本发明。然而，本领域的技术人员将知道，本发明在实施时可以不同于这些特定细节。另外，省略为人熟知的网络组件和程序实现的细节(例如，利用通信协议栈建立连接)，以免不必要地使本发明变得不明了。

图 1 是示出利用物理网络介质 104 连接到网际网络 102 的系统 100 的方框图。在特定实现中，系统 100 是因特网 web 服务器，并且网际网络 102 是因特网，但是本领域的技术人员将知道，可以在任何类型的网络服务器中实现本发明。

系统 100 包括文件服务器(例如 HTTP web 服务器)106 和适配卡 108。文件服务器 106 利用适配卡 108，在网际网络 102 上向客户 109(1-n)提供数据，并且从其接收数据。适配卡 108 建立并维护客户 109(1-n)和适配卡 108 之间的网络连接，并且建立服务器 106 和适配卡 108 之间的总线连接。如此连接之后，适配卡 108 代表服务器 106 从客户 109(1-n)接收通信，并且将该通信转发给服务器 106，代表客户 109 从服务器 106 接收响应，并且将该响应转发给客户 109。

服务器 106 包括非易失性存储器 110、工作存储器 112、服务器海量数据存储器 114、处理单元 116，和一个或多个用户输入/输出(input/output, I/O)设备 118，所有这些组件均利用服务器总线 120(例如，PCI 总线)进行相互通信。非易失性存储器 110(例如，只读存储器和/或一个或多个硬盘驱动器)提

供甚至当服务器 106 断电时仍保留的数据和代码存储。工作存储器 112(例如, 随机存取存储器)为服务器 106 提供工作存储器, 并且包括在启动时载入到工作存储器 112 中的可执行代码(例如, 操作系统)。在其它程序中, 工作存储器 112 包括服务器应用 121 和通信协议栈 122。服务器应用 121 包括让服务器 106 用作网络服务器的网络软件应用(例如, FTP、HTTP 等)。通信协议栈 122 是帮助通过网际网络与其它机器进行通信的标准协议栈(例如, TCP/IP)。标准协议栈在本领域内为人熟知。例如, 参见 W. Richard Stevens, TCP/IP 阐述, 卷 1(Addison-Wesley, 1994), 在此加入以作参考。服务器海量数据存储器 114 为服务器向连接到网际网络 102 的客户 109(1-n)提供的数据提供数据存储(例如, 一个或多个硬盘驱动器)。处理单元 116 执行工作存储器 112 中的指令, 以使服务器 106 执行它的主要功能(例如, 向客户提供数据, 并且从其接收数据)。I/O 设备 118 典型地包括键盘、监视器和/或帮助与服务器 106 的用户交互的其它设备。在网络服务器如因特网 web 服务器中, 典型地提供有上述组件中的每个组件。

适配卡 108 包括非易失性存储器 123、工作存储器 124、处理单元 126、总线协议桥 128 和网络控制器 129, 所有这些组件均利用适配器总线 130 进行相互通信。非易失性存储器 123 提供甚至当适配器 108 断电时仍保留的数据和代码(例如, 引导代码)存储。处理单元 126 通过执行工作存储器 124 中的代码, 实现适配卡 108 的功能。总线协议桥 128 提供适配器总线 130 与服务器总线 120 之间的接口, 并且网络控制器 129 提供适配器总线 130 与网络介质 104 之间的接口。

工作存储器 124 为适配器 108 提供工作存储器, 并且包括代理应用 132 和通信协议栈 134。代理 132 和协议栈 134 在启动时从非易失性存储器 123 载入到工作存储器 134 中。可选地, 代理 132 和协议栈 134 可以从一个或多个其它源, 包括但并不限于服务器 106 的非易失性存储器 110 或服务器海量数据存储器 114, 进行载入。代理 132, 当由处理单元 126 执行时, 建立并管理上述适配器 108 与服务器 106, 和适配器 108 与客户 109 之间的连接。

在本发明的这个特定实施例中, 协议栈 122 和 134 是标准(例如, TCP/IP)协议栈。在适配器 108 中使用标准通信协议栈帮助使用在绝大多数网络服务器中已存在的标准通信软件(例如, 协议栈 122)。然而, 本领域的技术人员将知道, 该特定元素(和其它所述元素, 即使没有明确陈述)不是本发明的根

本元素。例如，本发明可以在服务器 106 和适配器 108 中使用可定制通信软件(例如，服务器应用 121 与协议栈 134 或代理 132 之间的直接通信)进行实现。而且，在本发明的特定实施例中，通过向代理 132 提供对服务器 106 的资源(例如，服务器海量数据存储器 114)进行直接访问，可以省略该元素。

适配卡 108 利用总线协议桥 128 和服务器总线 120 之间的总线连接 136 连接到服务器 106。在该特定实施例中，总线连接 136 是典型的总线扩展槽，例如 PCI 插槽。然而，本领域的技术人员将知道，本发明可以使用其它类型的总线连接，包括但不限于 ISA 插槽、USB 端口、串行端口或并行端口，进行实现。总线连接 136 帮助在代理 132 和服务器应用 121 之间进行高速、大包、相对不易出错(与网络连接相比较)的通信，大大减轻了加在服务器 106 的处理单元 116 上的连接管理负担。总而言之，代理 132(运行在处理单元 116 上)通过慢且易于出错的网络连接与客户 109 进行通信，然后通过高速总线连接 136 代表客户 109 与服务器应用 121 进行通信。

图 2 是更加详细地示出代理 132 和协议栈 134 的工作存储器 124 的方框图。本领域的技术人员将知道，虽然代理 132 的各种软件模块作为相互连接的功能块进行示出，但是这些软件模块实际上是当由处理单元 126(图 1)执行时可以进行相互通信的存储在工作存储器 124 中的可执行代码块。

代理 132 包括主进程模块 202、多个客户过程模块 204(1-n)、数据缓冲区 206 和应用代理模块 208。主进程模块 202 提供对代理 132 的各个模块的全面控制和协调。对来自网际网络 102 上的客户 109(图 1)的连接请求进行响应，主进程模块 202 接受连接请求，初始化数据缓冲区 206 中用于那个客户连接的数据结构，启动一个新的单独客户连接进程 204，以处理该连接，然后通知应用代理模块 208，已经建立特定客户连接。每个客户进程 204 处理一个这种客户连接。应用代理模块 208 建立并管理与服务器 106 的总线连接。数据缓冲区 206 提供对从客户 109 接收且目的地为服务器 106 的数据，从服务器 106 接收且目的地为客户 109 的数据，和与所建立客户和/或服务器连接相关的连接数据的存储。

通信协议栈 134 是 TCP/IP 栈，包括套接字层 210、TCP 层 212、IP 层 214 和包括网络驱动程序 216 和服务器总线驱动程序 218 的设备层。协议栈 134 各层的功能在本领域内为人熟知，因此在此不作详细的讨论。通过套接字层 210、TCP 层 212、IP 层 214 和服务器总线驱动程序 218 建立代理 132 的各

个模块与服务器应用 121 之间的连接。通过套接字层 210、TCP 层 212、IP 层 214 和网络驱动程序 216 建立代理 132 的各个模块与客户 109 之间的连接。

图 3 是示出包括多个应用特定代理 208(1-f)的应用代理模块 208 的方框图，其中应用特定代理 208 包括超文本传输协议(Hypertext Transfer Protocol, HTTP)代理 208(1)、中转代理 208(2)、安全代理 208(3)和“其它”代理 208(f)。通过配置一个或多个应用特定代理 208(1-f)来对客户连接进行服务，主进程 202 将所建立客户连接通知给应用代理 208。配置应用特定代理(例如，HTTP 代理 208(1))的一个方法是在应用特定代理的处理队列中输入客户进程标识符。

主进程 202 从包含在客户连接请求中的端口号确定为特定用户进程执行哪个应用特定代理。标准做法是使用为人所知的端口号标识特定网络应用和/或协议(例如，文件传输协议(File Transfer Protocol, FTP)、HTTP 等)。例如，端口号 80 对应于 HTTP 连接请求。主进程 202 因此将启动用来响应表示端口 80 的连接请求的所有客户进程 204 通知给 HTTP 代理 208(1)。

HTTP 代理 208(1)监控它被通知的每个客户进程。当 HTTP 代理 208(1)确定完整的 HTTP 请求由客户进程(例如 204(n))接收并存储在数据缓冲区 206 中时，HTTP 代理 208(1)打开到服务器的连接，将该请求传输到服务器，并且从服务器接收响应，并将该响应存储在数据缓冲区 206 中，然后关闭服务器连接。服务器响应然后由相关联的客户进程 204(n)传输到客户 109(n)。

当主进程 202 接收端口号不对应于任何一个其它应用特定代理的连接请求时，主进程 202 通知中转代理 208(2)。中转代理 208(2)简单地打开服务器连接，将从相关联的客户进程 204 接收的数据从数据缓冲区 206 传输到服务器 106，然后关闭服务器连接。

主进程 202 可以将所有客户连接通知给一些应用特定代理，而不管相关联的端口号。例如，安全代理 208(3)用来通过例如终止任何因响应缺少一些授权标记的连接请求而启动的客户进程，在执行一个其它应用特定代理之前，对所有客户连接请求进行过滤。

图 3 包括“其它”代理 208(f)，以示出应用代理 208 可以包括任何适合特定应用的目前所知或将来开发的代理，包括但不限于，缓存 HTTP 代理应用、电子邮件应用和文件传输应用。

图 4 示出在数据缓冲区 206 中实现以完成通过代理 132 的数据传输的客

户数据结构 402(1-n)和代理数据结构 404(1-f)的例子。主进程 202 为每个客户进程 204 创建并初始化一个客户数据结构 402，并且为应用代理 208 中的每个应用特定代理创建并初始化一个代理数据结构 404。

每个客户数据结构 402 包括客户套接字 406、服务器套接字 408、连接状态 410、输入队列 412、输出队列 414 和应用代理数据 416。对于每个客户连接(例如，连接(n))，客户套接字 406(n)和服务器套接字 408(n)每个分别包括客户 109(n)和服务器 106 的 IP 地址和端口号，从而唯一地将每个客户数据结构 402(n)与一个客户进程 204(n)相关联。连接状态 410(n)表示连接(n)的当前状态(例如，所接收的完整请求、所接收的响应等)。输入队列 412(n)用于存储和累积由与特定数据结构 402(n)相关联的客户进程 204(n)从客户 109(n)接收的数据。输出队列 414(n)用来存储由客户进程 204(n)转发到客户 109(n)的来自应用代理 208 的数据。应用代理数据 416(n)用来存储任何特定于特定应用代理的信息(例如，标志等)

每个代理数据结构(例如，404(f))包括客户队列 418(f)、客户就绪队列 420(f)和待读取(read pending)队列 422(f)。客户队列 418(f)包括与对应于代理数据结构 404(f)的特定应用代理(f)相关联的每个客户进程 204 的客户进程描述符(例如，指向相关客户数据结构 402 的指针)。客户就绪队列 420(f)包括在其输入队列 412 中有准备由相关联的应用代理(f)进行处理(例如传输到服务器(106))的数据的每个客户数据结构 402 的客户进程描述符。待读取队列 422(f)包括等待来自服务器 106 的响应的每个客户进程的客户进程描述符。

本领域的技术人员将理解，上述客户数据结构 402 和代理数据结构 404 在性质上是示例性的，并且本发明可以使用其它数据结构。这种替换数据结构的配置必须依赖于所采用特定应用特定代理的功能和结构。

图 5 是概述本发明的一种用于管理客户和服务器之间的连接的特定方法 500 的流程图。在第一步骤 502，代理 132 与客户 109 建立网络连接，然后在第二步骤 504，利用网络连接从客户 109 接收通信(例如，HTTP 请求)。下一步，在第三步骤 506，代理 132 与服务器 106 建立总线连接，然后在第四步骤 508，利用总线连接将所接收的客户通信转发给服务器 106。然后，第五步骤 510，代理 132 从服务器 106 接收对客户通信的响应(例如，HTML 数据)，并且在第六步骤 512，利用客户网络连接将响应传输到客户 109。最后，在第七步骤 514，代理 132 确定是否存在终止(例如，关闭)信号，并且如果

存在终止信号，那么方法 500 结束。如果在第七步骤 514 不存在终止信号，那么方法 500 返回到第一步骤 502，以与另一客户 109 建立网络连接。

图 6 是概述一种执行方法 500 的第一步骤 502(与客户建立网络连接)的特定方法 600 的流程图。在第一步骤，主进程 202 连接到网际网络 102。然后在第二步骤 604，主进程 202 监听网际网络 102 上的通信，并且确定是否存在来自客户 109 的连接请求。如果不存在客户连接请求，那么方法 600 结束。如果存在来自客户 109(n)的连接请求，那么在第三步骤 606，主进程 202 接受来自客户 109(n)的连接请求，启动客户进程 204(n)以处理该连接，并且初始化数据缓冲区 206 中的客户数据结构 402(n)。下一步，在第四步骤 608，主进程 202 从客户连接请求辨别代理应用标识符(例如，端口号)，并且通过将客户进程描述符(例如，指向客户数据结构 402(n)的指针)写入到各个代理数据结构 404 的客户队列 418 中，根据标识符的值通知一个或多个应用代理 208(1-f)。最后，在第五步骤 610，主进程 202 确定是否打开最大客户连接允许数。如果打开最大客户连接数，那么方法 600 结束。如果没有打开最大客户连接数，那么方法 600 返回到第二步骤 604，监听另一客户请求。

图 7 是概述一种执行方法 500 的第二步骤 504(从客户 109 接收通信)的方法 700 的流程图。在第一步骤 702，主进程 202 确定是否存在任何客户进程 204 要进行处理以接收数据。如果主进程 202 已经处理完所有客户进程 204(1-n)，那么方法 700 结束。否则，在第二步骤 704，主进程 202 调用第一客户进程 204(1)。然后，在第三步骤 706，客户进程 204(1)检查它的客户连接(例如，TCP 缓冲区)，确定是否存在来自客户 109(1)的任何输入数据。如果没有第一客户进程 204(1)的输入数据，那么方法 700 返回到第一步骤 702，以处理任何剩余的客户进程 204(2-n)。如果在第三步骤 706，客户进程 204(1)确定存在来自客户 109(1)的输入数据，那么在第四步骤 708，客户进程 204(1)检查客户数据结构 402(1)，确定输入队列 412(1)是否可用于接收客户数据。如果输入队列 412(1)不可用，那么方法 700 返回到第一步骤 702，以处理任何剩余的客户进程 204(2-n)。如果在第四步骤 708，客户进程 204(1)确定输入队列 412(1)可用于接收数据，然后在第五步骤 710，客户进程 204(1)将输入客户数据传输到输入队列 412(1)中。然后，在第六步骤 712，客户进程 204(1)确定累积在输入队列中的数据是否构成完整的请求(即，准备传输到服务器 106 的数据，例如完整的 HTTP 请求)。如果该数据不构成完整的

请求, 那么方法 700 返回到第一步骤 702, 以处理任何剩余的客户进程 204(2-n)。然而如果在第六步骤 712, 客户进程 204(1)确定输入队列 412(1)中的数据构成完整的请求, 那么, 在第七步骤 714, 客户进程例如通过设置相应的连接状态 410(1), 通知代理应用 208 存在一个完整的请求。然后, 方法 700 返回到第一步骤 702, 以确定是否存在任何更多要进行处理的客户进程 204(2-n)。

图 8 是概述一种执行方法 500 的第三步骤 506(与服务器 106 建立总线连接)的方法 800 的流程图。在第一步骤 802, 第一个应用代理 208(1)从它的代理数据结构 404(1)的客户队列 418(1)提取第一客户描述符。然后在第二步骤 804, 代理 208(1)检查由第一客户描述符标识的客户数据结构 402 的连接状态 412, 确定第一客户在它的输入队列 412 中是否存在完整的请求。如果连接状态 412 表示完整的请求, 然后在第三步骤 806, 代理 208(1)将该客户描述符加到它的客户就绪队列 420(1)。下一步, 在第四步骤 808, 代理 208(1)确定是否打开服务器 106 的最大连接数。如果已经打开最大服务器连接数, 那么方法 800 结束。如果尚未打开最大服务器连接数, 那么在第五步骤 810, 代理 208(1)打开与服务器 106 的总线连接, 并且将连接信息写入相关联的客户数据结构 402 的服务器套接字 408。下一步, 在第六步骤 812, 代理 208(1)确定是否它已检查它的客户队列 418(1)中的最后一个客户描述符。如果已检查最后一个描述符, 那么方法 800 结束。否则, 方法 800 返回到第一步骤 802, 以提取客户队列 418(1)中的下一个客户描述符。如果在第二步骤 804, 代理 208(1)确定尚未接收完整的客户请求, 那么方法 800 直接前进到第六步骤 812。一旦处理完代理数据结构 404(1)的客户队列 418(1)中的所有描述符, 对每个其它应用代理 208(2-f), 重复方法 800 或类似方法。

图 9 是概述一种执行方法 500 的第四步骤 508(将客户通信转发给服务器 106)的方法 900 的流程图。在第一步骤 902, 代理 208(1)从它的代理数据结构 404(1)的客户就绪队列 420(1)接收第一客户描述符。然后, 在第二步骤 904, 代理 208(1)检查第一客户的数据结构 402 的服务器套接字 408, 确定是否打开服务器连接。如果打开服务器连接, 那么在第三步骤 906, 代理 208(1)通过所打开的服务器连接将客户数据(例如, HTTP 请求)从客户输入队列 412 传输到服务器 106。下一步, 在第四步骤 908, 代理 208(1)将客户描述符从客户就绪队列 420(1)移到待读取队列 422(1)。然后在第五步骤 910, 代理 208(1)

确定是否已检查客户就绪队列 420(1)中的最后一个客户。如果没有,那么方法 900 返回到第一步骤 902,以检查客户就绪队列 420(1)中的下一客户。如果已检查最后一个客户,那么方法 900 结束。如果在第二步骤 904,代理 208(1)确定没有打开特定客户的服务器连接,那么方法 900 直接前进到第五步骤 910,以确定是否已检查客户就绪队列 420(1)中的最后一个客户。一旦处理完代理数据结构 404(1)的客户就绪队列 420(1)中的所有描述符,对每个其它应用代理 208(2-f),重复方法 900 或类似方法。

图 10 是概述一种执行方法 500 的第五步骤 510(从服务器 106 接收响应)的方法 1000 的流程图。在第一步骤 1002,代理 208(1)确定待读取队列 422(1)是否为空,并且如果为空,那么方法 1000 结束。如果待读取队列 422(1)不为空,那么在第二步骤 1004,代理 208(1)从待读取队列提取第一客户描述符。下一步,在第三步骤 1006,代理 208(1)检查由用第一客户描述符标识的客户数据结构 402 的服务器套接字 408 标识的所打开服务器连接,以确定在那个连接上是否存在任何输入服务器数据(即,对客户请求的响应)。如果在那个连接上不存在输入服务器数据,那么方法 1000 返回到第二步骤 1004,以检查待读取队列中的下一客户。如果存在输入服务器数据,那么在第四步骤 1008,代理 208(1)检查确定由客户描述符标识的客户数据结构 402 的输出队列 414 是否可用。如果输出队列 414 不可用,那么方法 1000 返回到第二步骤 1004,以检查待读取队列 422(1)中的下一客户描述符。如果输出队列 414 可用,那么在第五步骤 1010,代理 208(1)将输入服务器数据移到客户数据结构 402 的输出队列 414。下一步,在第六步骤 1012,代理 208(1)确定服务器数据是否包含“文件结束”指示符。如果否,那么在第七步骤 1014,代理 208(1)检查确定是否已处理待读取队列 422(1)中的最后一个客户描述符。如果是,方法 1000 结束。如果否,方法 1000 返回到第二步骤 1004,以从待读取队列 422(1)读取下一客户描述符。

如果,在第六步骤 1012,代理 208(1)确定服务器数据包含文件结束指示符,那么方法 1000 前进到第八步骤 1016,其中,代理 208(1)从待读取队列删除该客户描述符,然后在第九步骤 1018,关闭服务器连接。在第九步骤 1018 之后,方法 1000 返回到第七步骤 1014。一旦处理完代理数据结构 404(1)的待读取队列 422(1)中的所有描述符,对每个其它应用代理 208(2-f),重复方法 1000 或类似方法。

图 11 是概述一种执行方法 500 的第六步骤 512(将数据传输到客户 109)的方法 1100 的流程图。在第一步骤 1102, 主进程 202 调用第一客户进程 204(1)。然后在第二步骤 1104, 第一客户进程 204(1)确定在客户数据结构 402(1)的输出队列 414(1)中是否有任何数据。如果在输出队列 414(1)中没有任何数据, 那么方法 1100 返回到第一步骤 1102, 其中, 主进程 202 调用剩余客户进程 204(2-n)的下一个进程。然而, 如果在第二步骤 1104, 客户进程 204(1)确定在输出队列 414(1)中有数据, 那么在第三步骤 1106, 客户进程 204(1)确定客户网络连接是否就绪以接收数据。如果客户网络连接没有就绪, 那么方法 1100 返回到第一步骤 1102。如果客户网络连接已经就绪, 那么在第四步骤 1108, 客户进程 204(1)将输出队列 414(1)中的至少部分数据移到客户连接(例如, TCP 输出缓冲区)。下一步, 在第五步骤 1110, 主进程 202 确定是否已调用最后一个客户进程。如果是, 则方法 1100 结束。如果不是, 则方法 1100 返回到第一步骤 1102, 以调用剩余客户进程 204(2-n)的下一个进程。

现在已完成对本发明的特定实施例的描述。很多所述特性可以在不脱离本发明的范围的情况下进行替换、修改或省略。例如, 适配器 108 的工作组件(例如处理单元 126 或代理 132)可以直接合并到服务器中, 而不是在可卸载适配卡中进行提供。而且, 可以用其它数据结构代替所提供的示例性数据结构。另外, 在此公布的方法和例程的特定次序不认为是本发明的本质元素。作为另一个例子, 主进程 202 可以配置为在启动时打开预定数目的与服务器 106 的持久性总线连接, 并且管理应用代理 208(1-f)对这些连接的使用, 从而消除服务器 106 重复打开和关闭总线连接的需要。对于本领域的技术人员, 特别鉴于前述公布, 将理解这些和其它与所示特定实施例的偏差。

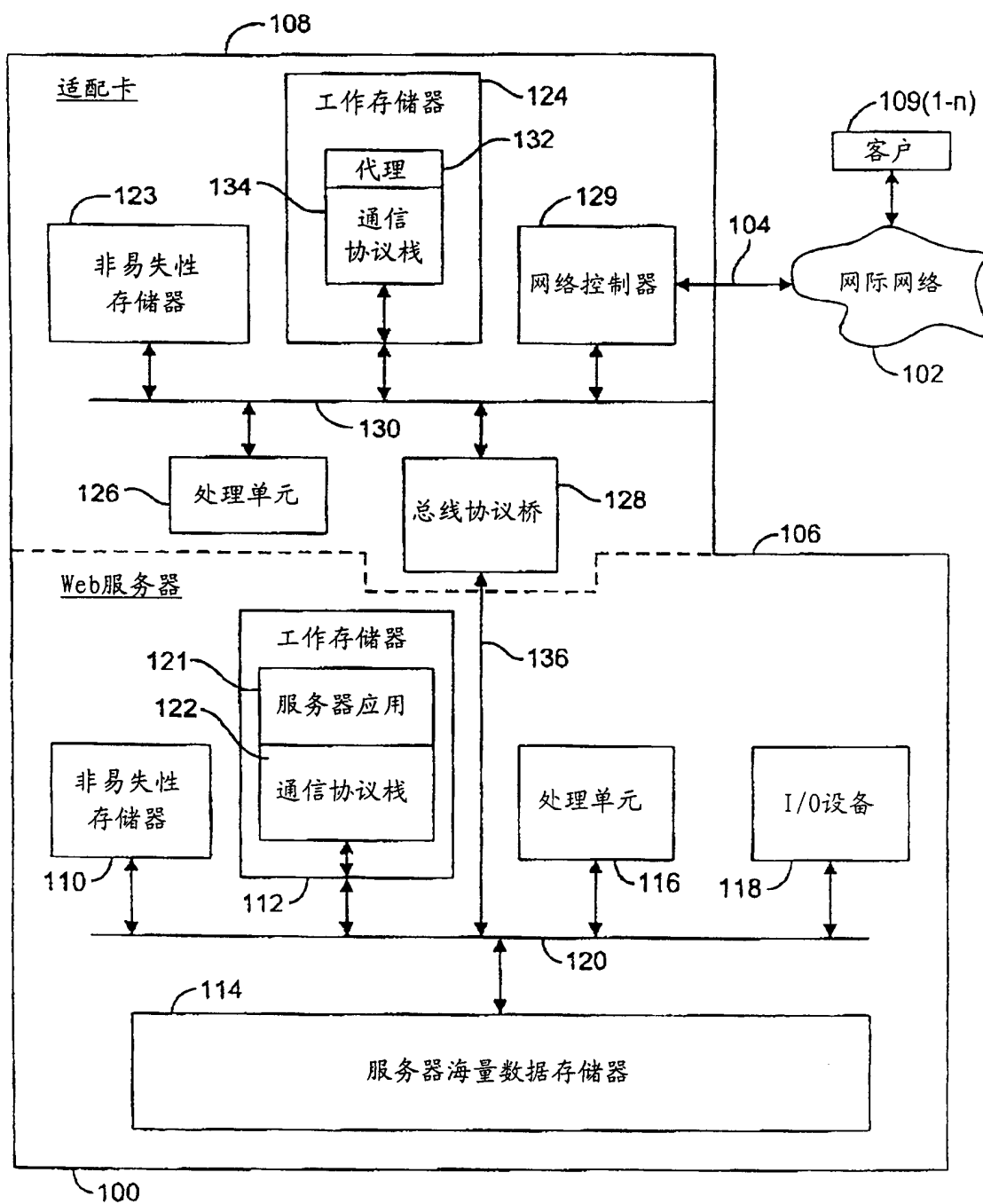


图 1

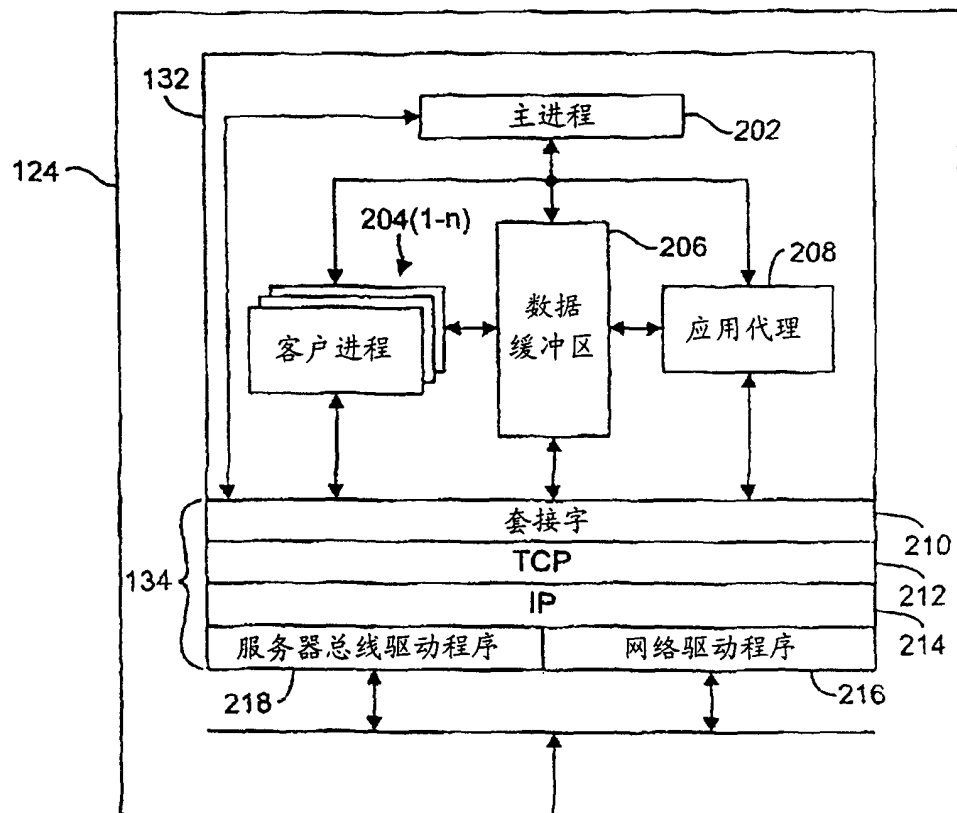


图 2

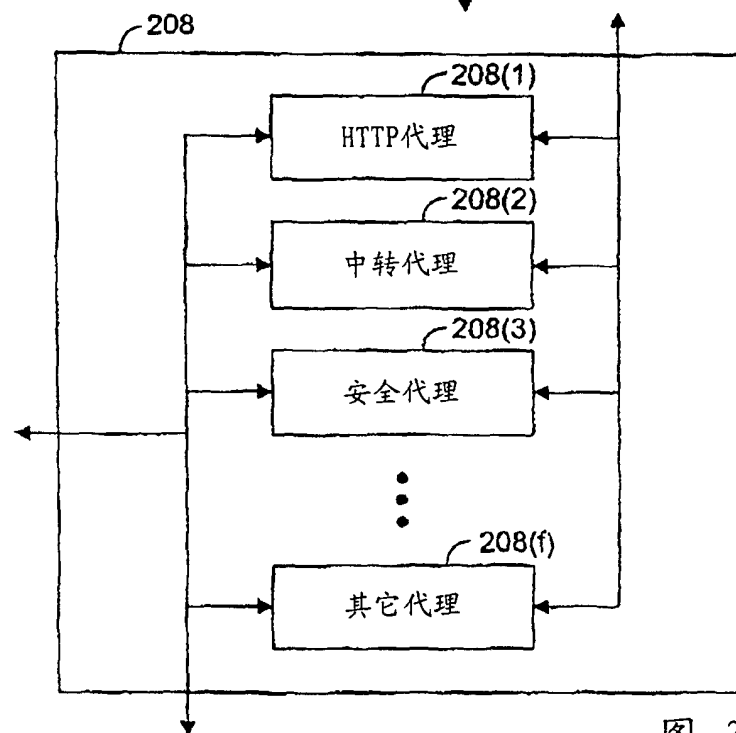


图 3

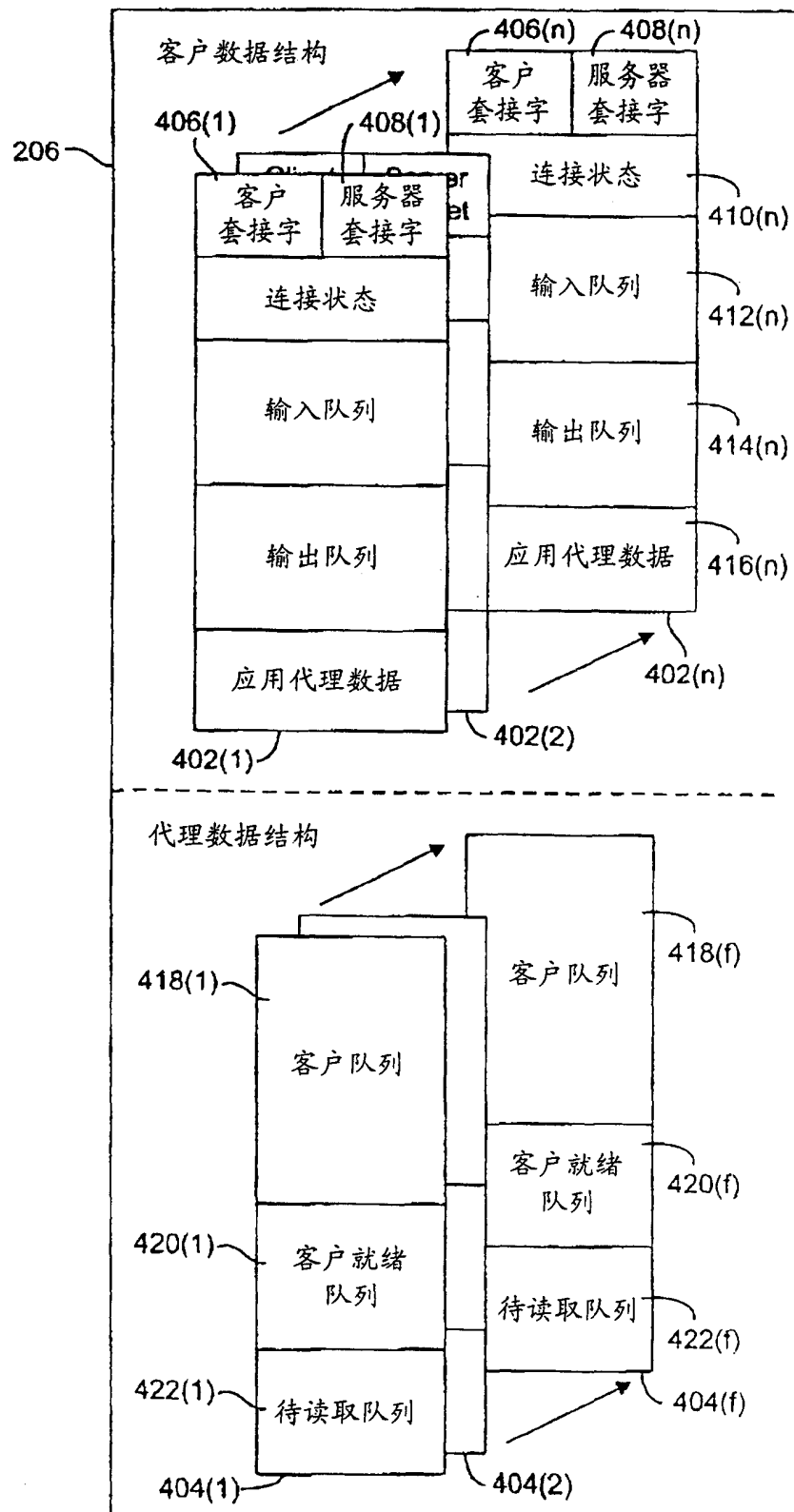


图 4

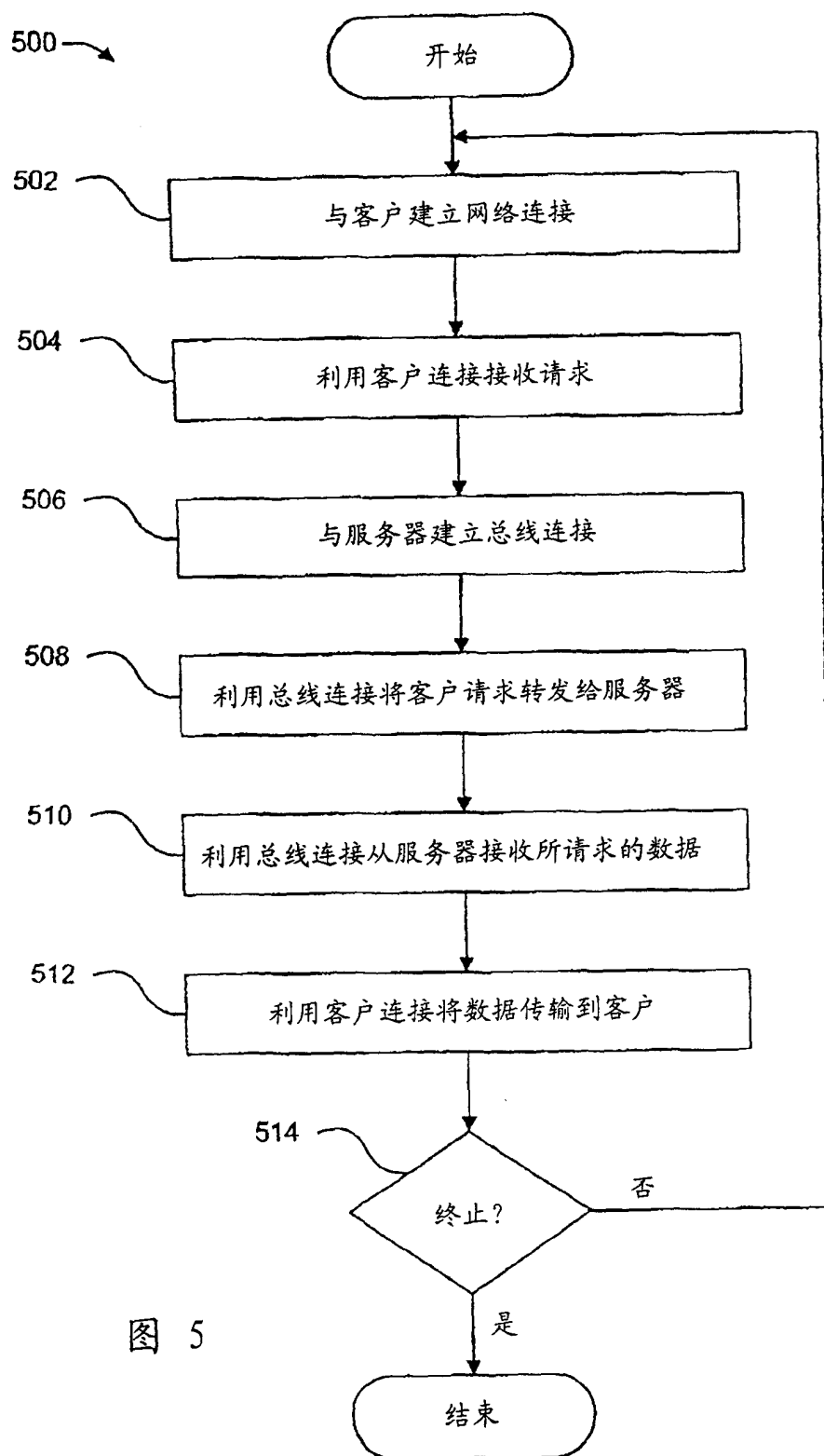


图 5

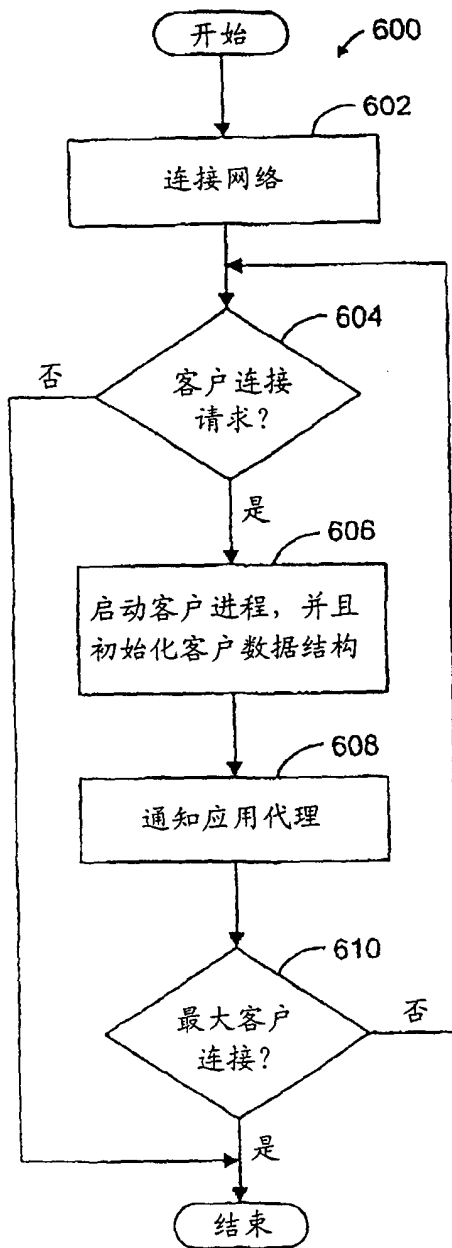


图 6

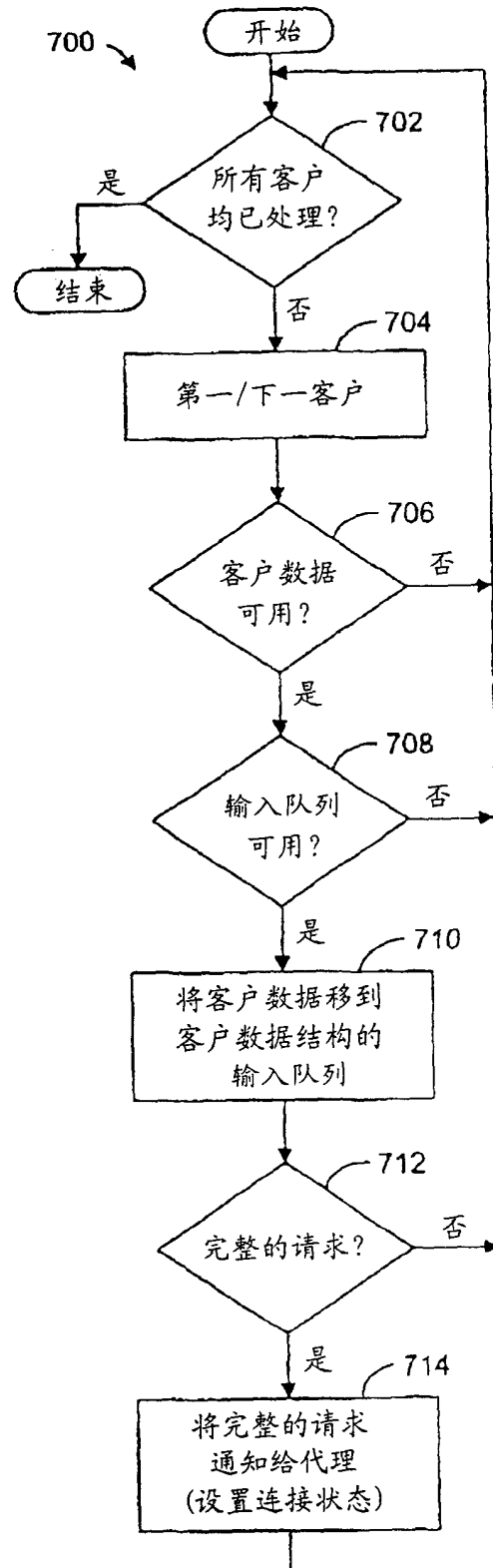


图 7

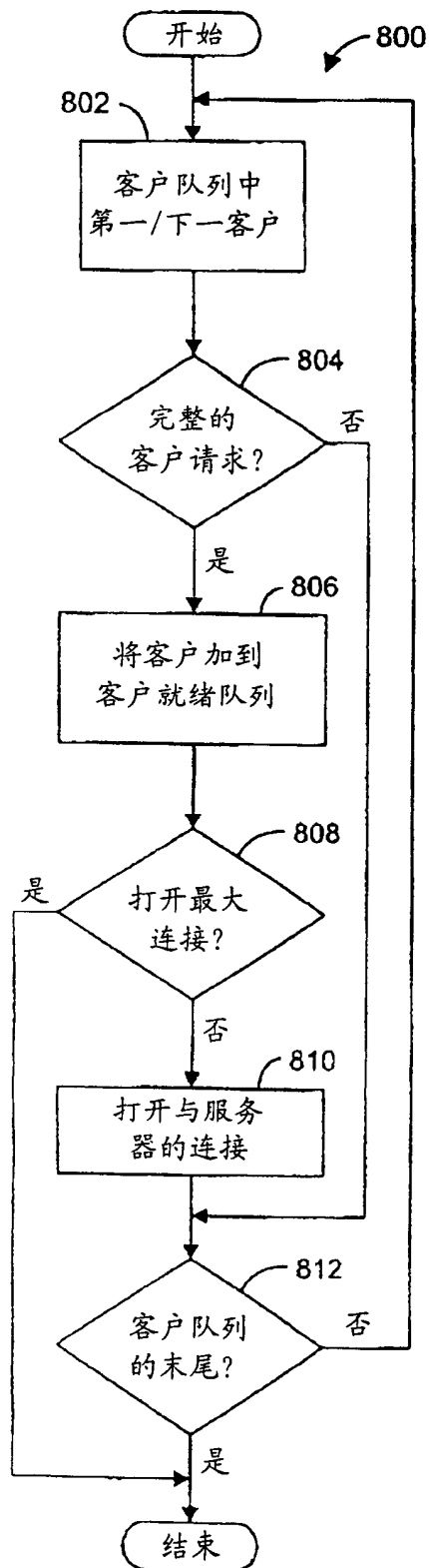


图 8

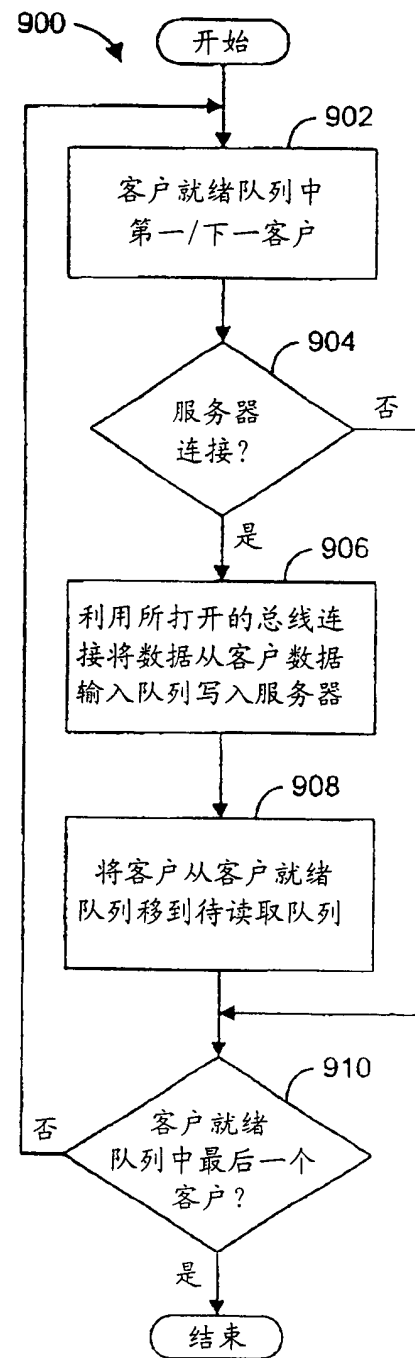
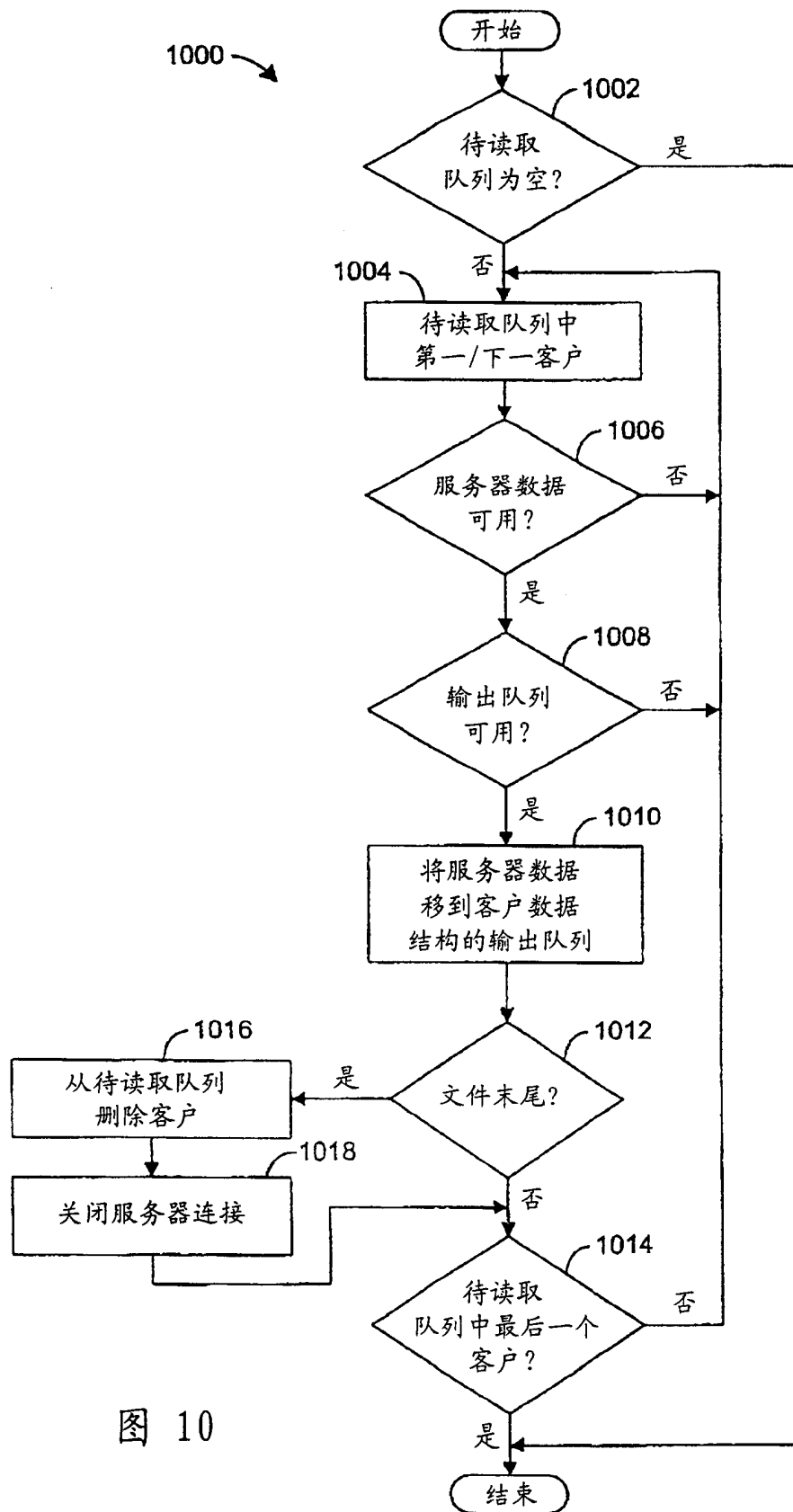


图 9



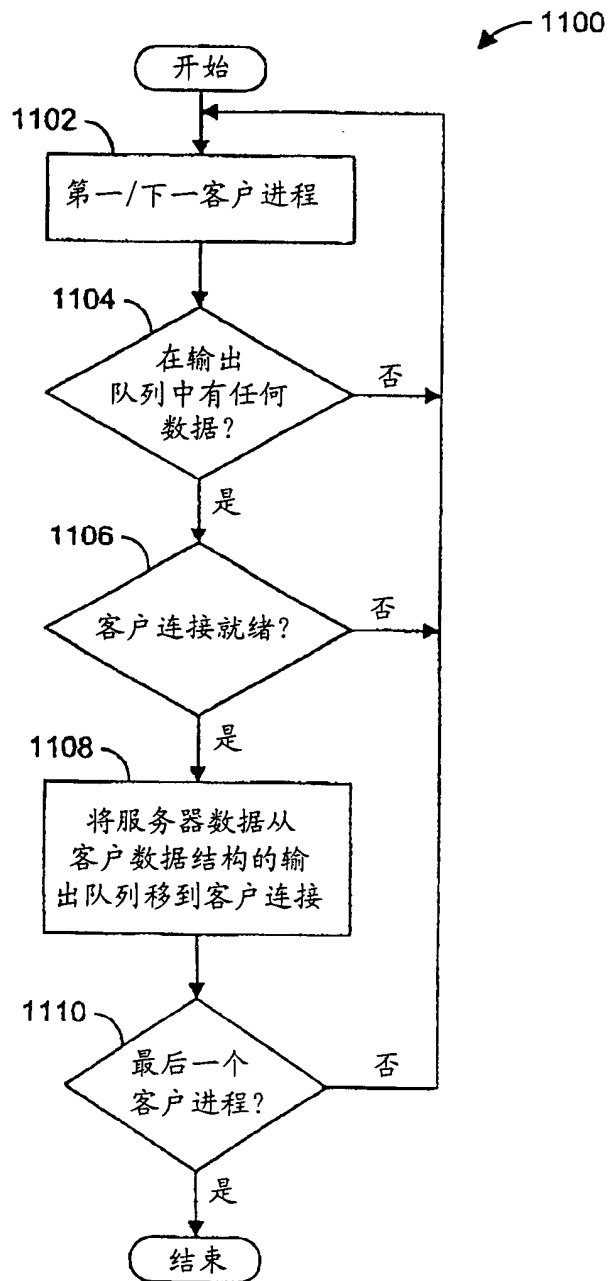


图 11