(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 101984778 A
(43)申请公布日 2011.03.09

(21)申请号 200980109863.X

(22)申请日 2009.01.26

(30)优先权数据
61/023849 2008.01.26 US

(85)PCT申请进入国家阶段日
2010.09.25

(86)PCT申请的申请数据
PCT/US2009/032046 2009.01.26

(87)PCT申请的公布数据
WO2009/094657 EN 2009.07.30

(71)申请人 思杰系统有限公司
地址 美国佛罗里达

(72)发明人 P·阿加瓦 S·K·阿德雅
S·西如娜拉雅南 J·哈里斯

(74)专利代理机构 北京泛华伟业知识产权代理有限公司 11280
代理人 王勇

(51)Int.Cl.
H04L 29/06(2006.01)

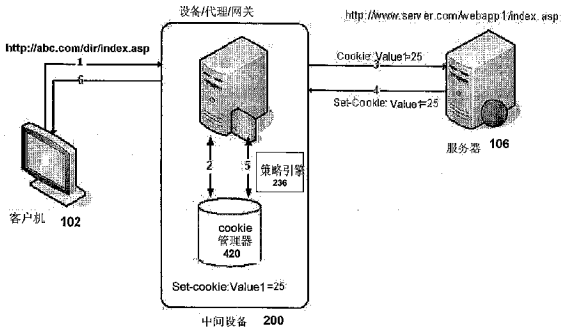
权利要求书 4 页 说明书 47 页 附图 14 页

(54)发明名称

用于细粒度策略驱动的COOKIE代理的系统和方法

(57)摘要

本发明使得企业可以基于和客户机、服务器相关的多种信息或者客户机和服务器之间的交互的细节和性质来配置不同的策略以处理业务量的多个子集。部署在客户机和服务器之间的中间设备可以在客户机和服务器之间建立SSL VPN会话。该中间设备可以经由免客户机(clientless)的SSL VPN会话从服务器接收对于客户机的请求的响应。该响应可以包括一个或者多个cookie。中间设备可以识别用于免客户机SSL VPN会话的访问配置文件。访问配置文件可以识别用于代理cookie的一个或者多个策略。中间设备可以响应于访问配置文件的一个或者多个策略来确定是否为客户机代理一个或者多个cookie或者将一个或者多个cookie的代理忽略。



CN 101984778 A

1. 一种用于通过中间设备对一个或者多个服务器和一个或者多个客户机之间的 cookie 进行配置驱动代理的方法，该中间设备在一个或者多个客户机和一个或者多个服务器之间建立 SSL VPN 会话，该方法包括：

(a) 该中间设备经由中间设备在服务器和客户机之间建立的免客户机 SSL VPN 会话从服务器接收对于客户机的请求的响应，该响应包括一个或者多个 cookie；

(b) 该中间设备识别用于免客户机 SSL VPN 会话的访问配置文件，该访问配置文件识别用于代理 cookie 的一个或者多个策略；和

(c) 该中间设备响应访问配置文件的一个或者多个策略来确定是否为客户机代理一个或者多个 cookie 或者忽略一个或者多个 cookie 的代理。

2. 权利要求 1 的方法，其中，步骤 (a) 还包括该中间设备经由该响应来接收一个或者多个 cookie 中的客户机消耗的 cookie，其中步骤 (b) 还包括该访问配置文件识别包括对于客户机消耗的 cookie 的 cookie 代理动作的策略，并且步骤 (c) 还包括中间设备响应于该策略来忽略客户机消耗的 cookie 的代理。

3. 权利要求 2 的方法，还包括该中间设备在转发给客户机的响应中保留客户机消耗的 cookie。

4. 权利要求 1 的方法，其中步骤 (a) 还包括该中间设备经由响应来接收一个或者多个 cookie 中的服务器 cookie，并且其中步骤 (c) 还包括中间设备代理该服务器 cookie。

5. 权利要求 4 的方法，还包括该中间设备从响应中移除服务器 cookie 并且将响应转发给客户机。

6. 权利要求 1 的方法，其中步骤 (c) 包括该中间设备响应于经由一个或者多个策略确定客户机不支持该一个或者多个 cookie 来代理响应中的一个或者多个 cookie。

7. 权利要求 1 的方法，其中步骤 (b) 包括该访问配置文件识别出用于限定对于指定域名的服务器消耗的 cookie 的 cookie 代理动作的策略，并且其中步骤 (c) 包括该中间设备如策略动作所指定对该响应进行修改。

8. 权利要求 1 的方法，还包括该中间设备基于应用类型的标识来经由请求或者响应识别访问配置文件。

9. 权利要求 1 的方法，还包括该访问配置文件基于用户或者用户组的标识来识别出包括忽略代理一个或者多个 cookie 的一个 cookie 的 cookie 代理动作的策略。

10. 权利要求 1 的方法，还包括该访问配置文件基于中间设备的虚拟服务器的标识来识别出忽略代理一个或者多个 cookie 的一个 cookie 的策略。

11. 权利要求 1 的方法，其中步骤 (c) 还包括除非访问配置文件的一个或者多个策略识别出一个或者多个 cookie 的一个 cookie 将被忽略，否则中间设备代理响应的一个或者多个 cookie。

12. 一种用于对一个或者多个服务器和一个或者多个客户机之间的 cookie 进行配置驱动代理的中间设备，该中间设备在一个或者多个服务器和一个或者多个客户机之间建立 SSL VPN 会话，该中间设备包括：

用于从服务器经由免客户机 SSL VPN 会话接收对于客户机请求的响应的包引擎，该免客户机 SSL VPN 会话由中间设备在服务器和客户机之间建立，该响应包括一个或者多个 cookie；

用于识别对于免客户机 SSL VPN 会话的访问配置文件的策略引擎，该访问配置文件识别用于代理 cookie 的一个或者多个策略；以及

其中，该中间设备响应于访问配置文件的一个或者多个策略来确定为客户机代理一个或者多个 cookie 或者忽略代理一个或者多个 cookie。

13. 权利要求 12 的中间设备，其中，该包引擎经由响应接收一个或者多个 cookie 中的客户机消耗的 cookie，该访问配置文件识别包括对于客户机消耗的 cookie 的 cookie 代理动作的策略，并且其中该中间设备还响应于该策略来确定忽略对客户机消耗的 cookie 的代理。

14. 权利要求 13 的中间设备，其中，该中间设备在转发给客户机的响应中保留客户机消耗的 cookie。

15. 权利要求 12 的中间设备，其中，该包引擎经由响应来接收一个或者多个 cookie 中的服务器 cookie，并且其中该中间设备响应于一个或者多个策略来代理该服务器 cookie。

16. 权利要求 15 的中间设备，其中，该中间设备从响应中移除服务器 cookie 并且将响应转发给客户机。

17. 权利要求 12 的中间设备，其中，该中间设备响应于经由一个或者多个策略确定客户机不支持一个或者多个 cookie 来代理响应中的一个或者多个 cookie。

18. 权利要求 12 的中间设备，其中，该策略引擎经由访问配置文件识别出对于指定域名的服务器消耗的 cookie 限定 cookie 代理动作的一个或者多个策略中的一个策略，并且其中该中间设备如策略动作所指定对响应进行修改。

19. 权利要求 12 的中间设备，其中，该策略引擎基于应用类型的标识来经由请求或者响应识别访问配置文件。

20. 权利要求 12 的中间设备，其中，该策略引擎基于用户或者用户组的标识来经由访问配置文件识别出包括忽略代理一个或者多个 cookie 的一个 cookie 的 cookie 代理动作的策略。

21. 一种通过中间设备为客户机管理 cookie 的方法，该方法包括：

中间设备从服务器接收对于客户机请求的响应，该响应包括 URL 和 cookie；

该中间设备通过将 cookie 从响应移除并且将唯一的客户机标识符插入 URL 来修改该响应；

该中间设备保存和唯一客户机标识符相关的所移除的 cookie；并且

该中间设备将修改的响应转发到客户机。

22. 权利要求 21 的方法，包括 cookie 管理器将与响应所包括的 cookie 相关联的一个或者多个值匹配到多个 cookie 中的一个 cookie。

23. 权利要求 22 的方法，其中，所述修改是响应于该匹配的。

24. 权利要求 21 的方法，还包括：

该中间设备从客户机接收请求，该请求包括请求 URL；和

Cookie 管理器将请求的 URL 和多个 cookie 中的所移除的 cookie 匹配。

25. 权利要求 21 的方法，还包括：

该中间设备通过增加已移除的 cookie 来修改请求的 URL；并且

该中间设备将修改的请求 URL 转发给服务器。

26. 权利要求 24 的方法，其中，多个 cookie 和请求 URL 所提供的至少一个域名或者至少一个路径相关联。

27. 权利要求 24 的方法，还包括该中间设备响应于该匹配从 cookie 管理器接收一个或者多个名称值对。

28. 权利要求 21 的方法，其中，该响应包括多个 cookie 并且修改的响应包括和多个 cookie 相关联的多个客户机标识符。

29. 权利要求 21 的方法，其中，该唯一客户机标识符是包括 cookie 的一部分的 cookie 代理会话 cookie。

30. 权利要求 21 的方法，还包括 ：

该中间设备从客户机接收请求，该请求包括 cookie 代理会话 cookie，cookie 代理会话 cookie 包括所述 cookie 的一部分 ；

该中间设备将 cookie 代理会话 cookie 匹配到所述 cookie ；

该中间设备通过将 cookie 代理会话 cookie 从请求移除并且将该 cookie 增加到请求来修改该请求 ；并且

该中间设备将该修改后的请求转发给服务器。

31. 一种用于使用网络上的代理来管理 cookie 的中间设备，该中间设备包括 ：

中间设备的包引擎从服务器接收对于客户机请求的响应，该响应包括 URL 和 cookie ；

中间设备的 cookie 代理，用于通过将 cookie 从响应中移除并且将唯一的客户机标识符插入 URL 来修改该响应 ；

中间设备的 cookie 管理器，用于保存和唯一客户机标识符相关的所移除的 cookie ；以及

该中间设备用于将修改的响应转发到客户机。

32. 权利要求 31 的中间设备，其中，该响应还包括和所述 cookie 相关联的一个或者多个值，并且其中该 cookie 管理器将与该 cookie 相关联的一个或者多个值匹配到来自多个 cookie 中的一个 cookie。

33. 权利要求 32 的中间设备，其中，该修改是响应于该匹配的。

34. 权利要求 31 的中间设备，其中，该包引擎从客户机接收请求，该请求包括请求 URL，并且 Cookie 管理器将请求 URL 匹配到多个 cookie 中的所移除的 cookie。

35. 权利要求 31 的中间设备，其中，该中间设备 ：

通过增加已移除的 cookie 来修改请求 URL ；并且

将修改的请求 URL 转发给服务器。

36. 权利要求 34 的中间设备，其中，该多个 cookie 和请求 URL 所提供的至少一个域名或者至少一个路径相关联。

37. 权利要求 34 的中间设备，其中，该中间设备响应于该匹配从 cookie 管理器接收一个或者多个名称值对。

38. 权利要求 31 的中间设备，其中，该响应包括多个 cookie 并且该修改的响应包括和多个 cookie 相关联的多个客户机标识符。

39. 权利要求 31 的中间设备，其中，该唯一客户机标识符是包括 cookie 所包括的一部分信息的 cookie 代理会话 cookie。

40. 权利要求 31 的中间设备，其中，该中间设备：

从客户机接收请求，该请求包括 cookie 代理会话 cookie，cookie 代理会话 cookie 包括 cookie 的一部分；

将 cookie 代理会话 cookie 匹配到该 cookie；

通过将 cookie 代理会话 cookie 从请求移除并且将该 cookie 增加到该请求来修改该请求；和

将该修改的请求转发给服务器。

用于细粒度策略驱动的 COOKIE 代理的系统和方法

[0001] 相关申请

[0002] 本申请要求 2008 年 1 月 26 日提交的美国临时申请 61/023849 号的优先权，其内容通过引用包含于此。

技术领域

[0003] 本发明总的涉及数据通信网络。更具体地，本发明涉及用于细粒度策略驱动的 cookie 代理管理、免客户机 cookie 管理的系统和方法以及用于 cookie 代理的技术。

背景技术

[0004] 企业可以通过网络来为多个不同客户提供不同的服务。一些客户可以经由更安全的连接和网络连接，而另一些客户可以使用安全性较差的网络连接。类似地，一些客户可以配置为使用 cookie 和服务器进行网络通信，而另一些客户不可以这样。在企业的服务器使用 cookie 来提供服务的例子中，不能使用 cookie 的客户难以连接服务并且使用服务。客户之间的网络配置和网络连接中的这样的变化可以对企业在提供服务到这些客户方面产生挑战。

发明内容

[0005] 在另一个方面中，提供一种技术方案，使得企业可以决定可以代理哪个 cookie 并且如何代理以及哪个 cookie 不可以代理。所提供的技术方案使得企业可以基于和客户机、服务器相关的多种信息或者客户机和服务器之间的交互的细节和性质来配置不同的策略以处理业务量的多个子集。例如，部署在客户机和服务器之间的中间设备可以在客户机和服务器之间建立 SSL VPN 会话。该中间设备可以经由免客户机 (clientless) 的 SSL VPN 会话从服务器接收对于客户机的请求的响应。该响应可以包括一个或者多个 cookie。中间设备可以识别用于免客户机 SSL VPN 会话的访问配置文件。访问配置文件可以识别用于代理 cookie 的一个或者多个策略。中间设备可以响应访问配置文件的一个或者多个策略来确定是否为客户机代理或者忽略代理一个或者多个 cookie。

[0006] 在一些方面，本发明涉及用于通过一个或者多个服务器和一个或者多个客户机之间的 cookie 的中间设备进行配置驱动的代理的方法。中间设备可以在一个或者多个客户机和一个或者多个服务器之间建立 SSL VPN 会话。该中间设备可以经由中间设备在服务器和客户机之间建立的免客户机的 SSL VPN 会话从服务器接收对于客户机的请求的响应。该响应可以包括一个或者多个 cookie。中间设备可以识别用于免客户机 SSL VPN 会话的访问配置文件。访问配置文件可以识别用于代理 cookie 的一个或者多个策略。中间设备可以响应访问配置文件的一个或者多个策略来确定对于客户机是否代理或者忽略一个或者多个 cookie 的代理。

[0007] 在一些实施例中，中间设备可以接收一个或者多个 cookie 中的客户机消耗的 cookie。访问配置文件可以识别包括对于客户机消耗的 cookie 的 cookie 代理动作的策

略。在另一个实施例中，中间设备还响应于该策略来忽略客户机消耗的 cookie 的代理。在另一个实施例中，中间设备在转发给客户机的响应中保留客户机消耗的 cookie。在又一个实施例中，中间设备经由响应来接收一个或者多个 cookie 的服务器 cookie 并且代理服务器 cookie。仍在另一个实施例中，中间设备从响应中移除服务器 cookie 并且将响应转发给客户机。在又一个实施例中，中间设备响应于经由一个或者多个策略确定客户机不支持一个或者多个 cookie 来代理响应的一个或者多个 cookie。仍在另一个实施例中，访问配置文件识别对于指定域名的服务器消耗的 cookie 限定 cookie 代理动作的策略并且中间设备如策略动作所指定的修改响应。

[0008] 在一些实施例中，中间设备基于应用类型的标识来识别使用请求或者响应的访问配置文件。在另一个实施例中，访问配置文件基于用户或者用户组的标识来识别包括 cookie 代理动作的策略，以忽略代理一个或者多个 cookie 的一个 cookie。仍然在另一个实施例中，访问配置文件基于中间设备的虚拟服务器的标识来识别策略，以忽略代理一个或者多个 cookie 的一个 cookie。在又一个实施例中，除非一个或者多个访问配置文件的策略识别要忽略的一个或者多个 cookie 的一个 cookie，否则中间设备代理响应的一个或者多个 cookie。

[0009] 在一些方面中，本发明涉及用于一个或者多个服务器和一个或者多个客户机之间的 cookie 的配置驱动代理的中间设备。中间设备可以在一个或者多个服务器和一个或者多个客户机之间建立 SSL VPN 会话。用于从服务器经由免客户机 SSL VPN 会话接收对于客户机请求的响应的包引擎可以通过服务器和客户机之间的中间设备来建立。响应可以包括一个或者多个 cookie。策略引擎用于识别对于免客户机 SSL VPN 会话的访问配置文件，访问配置文件识别用于代理 cookie 的一个或者多个策略。中间设备响应于访问配置文件的一个或者多个策略来确定是否为客户机代理一个或者多个 cookie 或者将代理忽略。

[0010] 在一些实施例中，包引擎经由响应接收一个或者多个 cookie 中的客户机消耗的 cookie。在一些实施例中，访问配置文件可以识别包括对于客户机消耗的 cookie 的 cookie 代理动作的策略。在另一个实施例中，中间设备还响应于该策略来确定将客户机消耗的 cookie 的代理忽略。在一些实施例中，中间设备在转发给客户机的响应中保留客户机消耗的 cookie。在又一个实施例中，包引擎经由响应来接收一个或者多个 cookie 的服务器 cookie 并且其中中间设备响应于一个或者多个策略来代理服务器 cookie。仍在另一个实施例中，中间设备从响应中移除服务器 cookie 并且将响应转发给客户机。在一些实施例中，中间设备响应于经由一个或者多个策略确定客户机不支持一个或者多个 cookie 来代理响应的一个或者多个 cookie。

[0011] 在一些实施例中，策略引擎经由访问配置文件识别对于指定域名的服务器消耗的 cookie 限定 cookie 代理动作的一个或者多个策略的一个策略。在另一个实施例中，中间设备如策略动作所指定的修改响应。在一些实施例中，策略引擎基于应用类型的标识来识别经由请求或者响应的访问配置文件。在一些实施例中，访问配置文件基于用户或者用户组的标识来经由访问配置文件识别包括 cookie 代理动作的策略来将代理一个或者多个 cookie 的一个 cookie 忽略。

[0012] 提供一种技术方案使得企业为客户机提供服务时不需要考虑它们的 cookie 或有关 cookie 的安全配置，并且在一些实施例中不需要牺牲传输的安全性和整体性。所提供的

技术方案使得没有配置为使用 cookie 的客户机访问使用 cookie 的服务器，该 cookie 用于与客户机通信。部署在客户机和服务器之间的中间设备拦截和修改客户机和服务器之间的传输来补偿客户机和服务器之间的 cooki e 配置的不匹配。

[0013] 在一些方面中，本申请涉及通过中间设备管理用于客户机的 cookie 的方法。中间设备从服务器接收对于客户机请求的响应。该响应可以包括统一资源定位符 (URL) 和 cookie。中间设备可以通过将 cookie 从响应移除并且将唯一的客户机标识符插入 URL 来修改该响应。中间设备可以保存和唯一客户机标识符相关的所移除的 cookie 并且将修改的响应转发到客户机。

[0014] 在一些实施例中，cookie 管理器将与响应所包括的 cookie 相关联的一个或者多个值匹配到来自多个 cookie 的一个 cookie。在另一个实施例中，中间设备响应于该匹配来修改所接收的响应。在一些实施例中，中间设备从客户机接收请求，该请求包括请求 URL。Cookie 管理器可以将请求 URL 匹配到来自多个 cookie 的所移除的 cookie。在一些实施例中，中间设备通过增加已移除的 cookie 来修改请求 URL 并且将修改的请求 URL 转发给服务器。在另一个实施例中，多个 cookie 和请求 URL 所提供的至少一个域名或者至少一个路径相关联。在又一个实施例中，中间设备响应于该匹配从 cookie 管理器接收一个或者多个名称值对。在一些实施例中，响应包括多个 cookie 并且修改的响应包括和多个 cookie 相关联的多个客户机标识符。在其他实施例中，唯一客户机标识符是包括 cookie 的一部分的 cookie 代理会话 cookie。

[0015] 在一些实施例中，中间设备从客户机接收请求，该请求包括 cooki e 代理会话 cookie，cookie 代理会话 cookie 包括 cookie 的一部分。中间设备可以将 cookie 代理会话 cookie 匹配到该 cookie。在一些实施例中，中间设备可以通过将 cookie 代理会话 cookie 从请求移除并且将该 cookie 增加到请求来修改该请求，并且将该请求转发给服务器。

[0016] 在一些方面中，提供用于使用网络上的代理来管理 cookie 的中间设备。中间设备的包引擎从服务器接收对于客户机请求的响应，该响应可以包括 URL 和 cookie。中间设备的 cookie 代理可以通过将 cookie 从响应移除并且将唯一的客户机标识符插入 URL 来修改该响应。中间设备的 cookie 管理器可以保存和唯一客户机标识符相关的所移除的 cookie。中间设备可以将修改的响应转发到客户机。

[0017] 在一些实施例中，响应还可以包括和 cookie 相关联的一个或者多个值。cookie 管理器可以将与 cookie 相关联的一个或者多个值匹配到来自多个 cookie 的一个 cookie。在一些实施例中，中间设备响应于该匹配来修改所接收的响应。在一些实施例中，包引擎从客户机接收请求。该请求可以包括请求 URL 并且 Cookie 管理器可以将请求 URL 匹配到来自多个 cookie 的所移除的 cookie。在一些实施例中，中间设备通过增加已移除的 cookie 来修改请求 URL 并且将修改的请求 URL 转发给服务器。在另一个实施例中，多个 cookie 和请求 URL 所提供的至少一个域名或者至少一个路径相关联。在又一个实施例中，中间设备响应于该匹配从 cookie 管理器接收一个或者多个名称值对。在一些实施例中，响应包括多个 cookie 并且修改的响应包括和多个 cookie 相关联的多个客户机标识符。在一些实施例中，唯一客户机标识符是包括 cookie 所包括的一部分信息的 cookie 代理会话 cookie。

[0018] 在一些实施例中，中间设备从容户机接收请求，该请求可以包括 cookie 代理会话 cookie。cookie 代理会话 cookie 可以包括 cookie 的一部分。中间设备可以将 cookie 代

理会话 cookie 匹配到该 cookie 并且通过将 cookie 代理会话 cookie 从请求移除并且将该 cookie 增加到请求来修改客户机的请求。中间设备可以将该修改后的请求转发给服务器。

附图说明

[0019] 参考结合附图的以下描述，本发明的前述和其他对象、方面、特征和优势将会变得更加明显和更好理解，其中：

[0020] 图 1A、1B 和 1C 是客户机经由一个或者多个设备来访问服务器的网络环境的实施例的框图；

[0021] 图 1D 是经由设备来将计算环境从服务器递送到客户机的环境的实施例的框图；

[0022] 图 1E 和 1F 是计算装置的实施例的框图；

[0023] 图 2A 是处理在客户机和服务器之间的通信的中间设备的实施例的框图；

[0024] 图 2B 是用于优化、加速、负载平衡和路由在客户机和服务器之间的通信的中间设备的另一个实施例的框图；

[0025] 图 3A 是经由中间设备访问服务器的免客户机虚拟专用网络的实施例的框图；

[0026] 图 3B 是经由中间设备访问服务器的免客户机虚拟专用网络的另一个实施例的框图；

[0027] 图 4A 是涉及 cookie 管理的多个实施例的框图；

[0028] 图 4B 是表示用在 cookie 管理中的多个管理序列图的框图；

[0029] 图 4C 是表示 cookie 代理数据流的多个实施例的框图，包括涉及 cookie 代理的方法；和

[0030] 图 5 是用于通过中间设备代理配置驱动 cookie 的方法的实施例的流程图。

[0031] 根据以下结合附图提出的详细描述，本发明的特征和优势将变得更加明显，其中相同的参考符号在全文中标示对应元件。在附图中，同样的附图标记通常指示相同的、功能类似的和 / 或结构类似的元件。

具体实施方式

[0032] A. 网络和计算环境

[0033] 在讨论设备和 / 或客户机的系统和方法的实施例的细节之前，讨论可以部署这样的实施例的网络和计算环境是有帮助的。现在参考图 1A，描述了网络环境的一个实施例。总的来说，网络环境包括经由一个或多个网络 104 和 104′（通常被称为网络 104）与一个或多个服务器 106a-106n（通常也被称为服务器 106 或远程机器 106）通信的一个或多个客户机 102a-102n（通常也被称为本地机器 102 或客户机 102）。在一些实施例中，客户机 102 经由设备 200 与服务器 106 通信。

[0034] 虽然图 1A 示出客户机 102 和服务器 106 之间的网络 104 和网络 104′，但客户机 102 和服务器 106 可以在同一个网络 104 上。网络 104 和 104′ 可以是相同类型的网络或不同类型的网络。网络 104 和 / 或网络 104′ 可以是像公司内联网的局域网 (LAN)、城域网 (MAN) 或者诸如因特网或万维网的广域网 (WAN)。在一个实施例中，网络 104′ 可以是专用网而网络 104 可以是公用网。在一些实施例中，网络 104 可以是专用网而网络 104′ 可以是公用网。在另一个实施例中，网络 104 和 104′ 可以都是专用网。在一些实施例中，客户

机 102 可以位于公司的分支机构，经由网络 104 上的 WAN 连接来与位于公司的数据中心的服务器 106 进行通信。

[0035] 网络 104 和 / 或 104′ 可以是任一类型和 / 或形式的网络，并且可以包括下列任意一种网络 ：点到点网络、广播网、广域网、局域网、电信网、数据通信网、计算机网络、ATM（异步传送模式）网络、SONET（同步光网络）网络、SDH（同步数字系列）网络、无线网络和有线网络。在一些实施例中，网络 104 可以包括诸如红外信道或卫星频带的无线链路。网络 104 和 / 或 104′ 的拓扑结构可以是总线型、星型或环型网络拓扑结构。网络 104 和 / 或 104′ 以及网络拓扑结构可以是能够支持此处描述的操作的本领域内普通技术人员所知的任一种这样的网络或网络拓扑结构。

[0036] 如图 1A 所示，在网络 104 和 104′ 之间示出也可以被称为接口单元 200 或网关 200 的设备 200。在一些实施例中，设备 200 可以位于网络 104 上。例如，公司的分支机构可以在所述分支机构处部署设备 200。在其它实施例中，设备 200 可以位于网络 104′ 上。例如，设备 200 可以位于公司的数据中心。在又一个实施例中，多个设备 200 可以部署在网络 104 上。在一些实施例中，多个设备 200 可以部署在网络 104′ 上。在一个实施例中，第一设备 200 与第二设备 200′ 相通信。在其它的实施例中，设备 200 可以是在与客户机 102 相同或不同的网络 104，104′ 上的任一客户机 102 或服务器 106 的一部分。一个或多个设备 200 可以位于在客户机 102 和服务器 106 之间的网络或网络通信路径中的任一点处。

[0037] 在一些实施例中，设备 200 包括被称为 Citrix NetScaler 装置的由位于 Ft. Lauderdale Florida 的 Citrix Systems 公司制造的任一网络装置。在其它实施例中，设备 200 包括由位于 Seattle，Washington 的 F5Networks 公司制造的被称为 WebAccelerator 和 BigIP 的任意一个产品实施例。在另一个实施例中，设备 205 包括由位于 Sunnyvale，California 的 Juniper Networks 公司制造的 DX 加速装置平台和 / 或诸如 SA700、SA2000、SA4000 和 SA6000 装置的 SSL VPN 系列装置中的任意一个。在又一个实施例中，设备 200 包括由位于 San Jose，California 的 CiscoSystems 公司制造的任一应用加速和 / 或安全相关的设备和 / 或软件，例如 Cisco ACE 应用控制引擎模块业务 (Application Control Engine Moduleservice) 软件和网络模块以及 Cisco AVS 系列应用速度系统 (Application Velocity System)。

[0038] 在一个实施例中，该系统可以包括多个、逻辑分组的服务器 106。在这些实施例中，服务器的逻辑分组可以被称为服务器群组 38。在这些实施例中的一些实施例中，服务器 106 可以是在地理上分散的。有时候，群组 38 可以被管理为单一的实体。在其它实施例中，服务器群组 38 包括多个服务器群组 38。在一个实施例中，服务器群组代表一个或多个客户机 102 来执行一个或多个应用。

[0039] 在每个群组 38 中的服务器 106 可以是不同种类的。一个或多个服务器 106 可以根据一种类型的操作系统平台（例如，由位于 Redmond，Washington 的微软公司出品的 WINDOWS NT）来进行操作，而一个或多个其它的服务器 106 可以根据另一种类型的操作系统平台（例如，Unix 或 Linux）来进行操作。每个群组 38 中的服务器 106 不需要与同一群组 38 中的另一个服务器 106 物理上接近。因此，逻辑上被分组为群组 38 的服务器 106 的分组可以使用广域网 (WAN) 连接或中域网 (medium-areanetwork，MAN) 连接来互连。例如，群组 38 可以包括在物理上位于不同的洲或位于一个洲、国家、州、城市、校园或房间的不同

区域的服务器 106。如果使用局域网 (LAN) 连接或一些形式的直接连接来连接服务器 106，则可以增加在群组 38 中的服务器 106 之间的数据传输速度。

[0040] 服务器 106 可以被称为文件服务器、应用服务器、web 服务器、代理服务器或网关服务器。在一些实施例中，服务器 106 可以有能力起到应用服务器或主应用服务器的作用。在一个实施例中，服务器 106 可以包括活动目录 (Active Directory)。客户机 102 也可以被称为客户机节点或端点。在一些实施例中，客户机 102 有能力起到寻求访问服务器上的应用的客户机节点以及作为对于其它的客户机 102a-102n 提供对寄载的应用的访问的应用服务器的作用。

[0041] 在一些实施例中，客户机 102 与服务器 106 进行通信。在一个实施例，客户机 102 直接与群组 38 中的服务器 106 的其中一个进行通信。在另一个实施例中，客户机 102 执行程序邻近应用以与群组 38 中的服务器 106 进行通信。在又一个实施例中，服务器 106 提供主节点的功能。在一些实施例中，客户机 102 通过网络 104 与群组 38 中的服务器 106 进行通信。例如，通过网络 104，客户机 102 可以请求执行由群组 38 中的服务器 106a-106n 寄载的多个应用，并接收应用执行的输出结果用于显示。在一些实施例中，只有主节点提供所要求的识别并提供与寄载被请求的应用的服务器 106′ 相关的地址信息的功能。

[0042] 在一个实施例中，服务器 106 提供 web 服务器的功能。在另一个实施例中，服务器 106a 从客户机 102 接收请求，将请求转发到第二服务器 106b，并使用来自于服务器 106b 的对请求的响应来对客户机 102 的请求进行响应。在又一个实施例中，服务器 106 获得客户机 102 可用的应用的列举以及与寄载由所述应用的列举所标识的应用的服务器 106 相关的地址信息。在又一个实施例中，服务器 106 使用 web 接口将对请求的响应提供给客户机 102。在一个实施例中，客户机 102 直接与服务器 106 进行通信以访问所标识的应用。在另一个实施例中，客户机 102 接收由执行服务器 106 上的标识的应用所生成的诸如显示数据的应用输出数据。

[0043] 现在参考图 1B，描述了部署多个设备 200 的网络环境的实施例。第一设备 200 可以部署在第一网络 104 上，而第二设备 200′ 部署在第二网络 104′ 上。例如，公司可以在分支机构部署第一设备 200，而在数据中心部署第二设备 200′ 。在另一个实施例中，第一设备 200 和第二设备 200′ 被部署在同一个网络 104 或网络 104′ 上。例如，第一设备 200 可以部署用于第一服务器群组 38，而第二设备 200′ 可以部署用于第二服务器群组 38′ 。在另一个实例中，第一设备 200 可以部署在第一分支机构，而第二设备 200′ 被部署在第二分支机构′ 。在一些实施例中，第一设备 200 和第二设备 200′ 彼此协同或联合工作，以加速客户机和服务器之间的网络业务量或应用和数据的递送。

[0044] 现在参考图 1C，描述了使用一个或多个其它类型的设备（例如在一个或多个 WAN 优化设备 205，205′ 之间的设备），来部署设备 200 的网络环境的另一个实施例。例如，第一 WAN 优化设备 205 显示在网络 104 和 104′ 之间，而第二 WAN 优化设备 205′ 可以部署在设备 200 和一个或多个服务器 106 之间。通过示例，公司可以在分支机构部署第一 WAN 优化设备 205，而在数据中心部署第二 WAN 优化设备 205′ 。在一些实施例中，设备 205 可以位于网络 104′ 上。在其它实施例中，设备 205′ 可以位于网络 104 上。在一些实施例中，设备 205′ 可以位于网络 104′ 或网络 104″ 上。在一个实施例中，设备 205 和 205′ 在同一个网络上。在另一个实施例中，设备 205 和 205′ 在不同的网络上。在另一个实例中，第

一 WAN 优化设备 205 可以部署用于第一服务器群组 38，而第二 WAN 优化设备 205′ 可以部署用于第二服务器群组 38′ 。

[0045] 在一个实施例中，设备 205 是用于加速、优化或者以其他方式改善诸如往和 / 或返于 WAN 连接的业务量的任一类型和形式的网络业务量的性能、操作或服务质量的装置。在一些实施例中，设备 205 是一个性能提高的代理。在其它实施例中，设备 205 是任一类型和形式的 WAN 优化或加速装置，有时也被称为 WAN 优化控制器。在一个实施例中，设备 205 是由位于 Ft. Lauderdale Florida 的 Citrix Systems 公司出品的被称为 WANScaler 的产品实施例中的任意一种。在其它实施例中，设备 205 包括由位于 Seattle，Washington 的 F5 Networks 公司出品的被称为 BIG-IP 链路控制器和 WANjet 的产品实施例中的任意一种。在另一个实施例中，设备 205 包括由位于 Sunnyvale，California 的 Juniper NetWorks 公司出品的 WX 和 WXC WAN 加速装置平台中的任意一种。在一些实施例中，设备 205 包括由 San Francisco，California 的 Riverbed Technology 公司出品的虹鳟 (steelhead) 系列 WAN 优化设备中的任意一种。在其它实施例中，设备 205 包括由位于 Roseland，NewJersey 的 Expand Networks 公司出品的 WAN 相关装置中的任意一种。在一个实施例中，设备 205 包括由位于 Cupertino，California 的 Packeteer 公司出品的任意一种 WAN 相关设备，例如由 Packeteer 提供的 PacketShaper、iShared 和 SkyX 产品实施例。在又一个实施例中，设备 205 包括由位于 San Jose，California 的 Cisco Systems 公司出品的任一 WAN 相关设备和 / 或软件，例如 Cisco 广域网应用服务软件和网络模块以及广域网引擎设备。

[0046] 在一个实施例中，设备 205 提供用于分支机构或远程办公室的应用和数据加速业务。在一个实施例中，设备 205 包括广域文件服务 (WAFS) 的优化。在另一个实施例中，设备 205 加速文件的递送，例如经由通用 Internet 文件系统 (CIFS) 协议。在其它实施例中，设备 205 在存储器和 / 或存储设备中提供高速缓存来加速应用和数据的递送。在一个实施例中，设备 205 提供在任一级别的网络堆栈或在任一的协议或网络层的网络业务量的压缩。在另一个实施例中，设备 205 提供传输层协议优化、流量控制、性能增强或修改和 / 或管理，以加速 WAN 连接上的应用和数据的递送。例如，在一个实施例中，设备 205 提供传输控制协议 (TCP) 优化。在其它实施例中，设备 205 提供对于任一会话或应用层协议的优化、流量控制、性能增强或修改和 / 或管理。

[0047] 在另一个实施例中，设备 205 将任一类型和形式的数据或信息编码成网络分组的定制或标准的 TCP 和 / 或 IP 报头字段或可选字段，以将存在、功能或能力通告给另一个设备 205′ 。在另一个实施例中，设备 205′ 可以使用在 TCP 和 / 或 IP 报头字段或选项中编码的数据来与另一个设备 205′ 进行通信。例如，设备可以使用 TCP 选项或 IP 报头字段或选项来传达在执行诸如 WAN 加速的功能时或者为了彼此联合工作而由设备 205，205′ 所使用的一个或多个参数。

[0048] 在一些实施例中，设备 200 保存在设备 205 和 205′ 之间传达的 TCP 和 / 或 IP 报头和 / 或可选字段中编码的任一信息。例如，设备 200 可以终止经过设备 200 的传输层连接，例如经过设备 205 和 205′ 的在客户机和服务器之间的一个传输层连接。在一个实施例中，设备 200 识别并保存在由第一设备 205 通过第一传输层连接发送的传输层分组中的任一编码信息，并经由第二传输层连接来将具有编码信息的传输层分组传达到第二设备 205′ 。

[0049] 现在参考图 1D，描述了用于递送和 / 或操作客户机 102 上的计算环境的网络环境。

在一些实施例中，服务器 106 包括用于将计算环境或应用和 / 或数据文件递送给一个或多个客户机 102 的应用递送系统 190。简单概述，客户机 10 经由网络 104、104′ 和设备 200 与服务器 106 通信。例如，客户机 102 可以驻留在例如分支机构的公司的远程办公室，而服务器 106 可以驻留在公司的数据中心。客户机 102 包括客户机代理 120 和计算环境 15。计算环境 15 可以执行或操作访问、处理或使用数据文件的应用。可以经由设备 200 和 / 或服务器 106 来递送计算环境 15、应用和 / 或数据文件。

[0050] 在一些实施例中，设备 200 加速将计算环境 15 或其任一部分递送给客户机 102。在一个实施例中，设备 200 通过应用递送系统 190 来加速计算环境 15 的递送。例如，此处描述的实施例可以用来将可由应用处理的流应用和数据文件加速从中央的公司数据中心递送到远程用户的位置，例如公司的分支机构。在另一个实施例中，设备 200 加速在客户机 102 和服务器 106 之间的传输层业务量。设备 200 可以提供用于加速从服务器 106 到客户机 102 的任一传输层有效载荷的加速技术，例如 :1) 传输层连接池，2) 传输层连接多路复用，3) 传输控制协议缓冲，4) 压缩和 5) 高速缓存。在一些实施例中，设备 200 提供响应于客户机 102 的请求的服务器 106 的负载平衡。在其它实施例中，设备 200 充当代理或访问服务器以提供对一个或多个服务器 106 的访问。在另一个实施例中，设备 200 提供从客户机 102 的第一网络 104 到服务器 106 的第二网络 104′ 的安全虚拟专用网连接，例如 SSL VPN 连接。在又一些实施例中，设备 200 提供在客户机 102 和服务器 106 之间的连接和通信的应用防火墙安全、控制和管理。

[0051] 在一些实施例中，应用递送管理系统 190 根据多个执行方法以及根据经由策略引擎 195 应用的任一验证和授权策略来提供将计算环境递送到远端或另外的用户的桌面的应用递送技术。使用这些技术，远程用户可以从任一网络连接装置 100 获取计算环境以及访问服务器存储的应用和数据文件。在一个实施例中，应用递送系统 190 可以驻留于服务器 106 或在服务器 106 上执行。在另一个实施例中，应用递送系统 190 可以驻留于多个服务器 106a-106n 上或在多个服务器 106a-106n 上执行。在一些实施例中，应用递送系统 190 可以在服务器群组 38 中执行。在一个实施例中，执行应用递送系统 190 的服务器 106 还可以存储或提供应用和数据文件。在另一个实施例中，第一组的一个或多个服务器 106 可以执行应用递送系统 190，而不同的服务器 106n 可以存储或提供应用和数据文件。在一些实施例中，应用递送系统 190、应用和数据文件中的每一个可以驻留或位于不同的服务器上。在又一个实施例中，应用递送系统 190 的任一部分可以驻留、执行或保存或被分配于设备 200 或多个设备。

[0052] 客户机 102 可以包括用于执行使用或处理数据文件的应用的计算环境 15。客户机 102 可以经由网络 104、104′ 和设备 200 来从服务器 106 请求应用和数据文件。在一个实施例中，设备 200 可以将来自于客户机 102 的请求转发到服务器 106。例如，客户机 102 可以不具有本地存储或可存取的应用和数据文件。响应于请求，应用递送系统 190 和 / 或服务器 106 可以递送应用和数据文件到客户机 102。例如，在一个实施例中，服务器 106 可以以应用流的形式发送应用，以在客户机 102 上的计算环境 15 中进行操作。

[0053] 在一些实施例中，应用递送系统 190 包括 Citrix Systems 公司的例如 MetaFrame 或 Citrix 表示 (Presentation) 服务器 ™ 的 Citrix 访问套件 ™ 的任一部分和 / 或由微软公司出品的任意一种微软® Windows 终端服务。在一个实施例中，应用递送系统 190 可以

通过远程显示协议或以其它方式通过基于远程或基于服务器的计算来递送一个或多个应用到客户机 102 或用户。在另一个实施例中，应用递送系统 190 可以通过应用的流式传输来递送一个或多个应用到客户机或用户。

[0054] 在一个实施例中，应用递送系统 190 包括用于控制和管理应用执行方法的访问、选择以及应用的递送的策略引擎 195。在一些实施例中，策略引擎 195 确定用户或客户机 102 可以访问的一个或多个应用。在另一个实施例中，策略引擎 195 确定应用应该如何被递送给用户或客户机 102，例如执行方法。在一些实施例中，应用递送系统 190 提供从中选择应用执行方法的多个递送技术，例如基于服务器的计算、本地流式传输或递送应用给客户机 120 以用于本地执行。

[0055] 在一个实施例中，客户机 102 请求执行应用而包括服务器 106 的应用递送系统 190 选择执行应用的方法。在一些实施例中，服务器 106 从客户机 102 接收证书。在另一个实施例中，服务器 106 从客户机 102 接收列举可用的应用的请求。在一个实施例中，响应于所述请求或收到的证书，应用递送系统 190 列举客户机 102 可用的多个应用。应用递送系统 190 接收请求以执行所列举的应用。应用递送系统 190 选择预定数目的方法中的一个来执行列举的应用，例如响应于策略引擎的策略。应用递送系统 190 可以选择一个执行应用的方法，使得客户机 102 可以接收通过在服务器 106 上执行应用而生成的应用输出数据。应用递送系统 190 可以选择执行应用的方法，使得本地机器 10 可以在检索包括应用的多个应用文件之后本地执行所述应用。在又一个实施例中，应用递送系统 190 可以选择执行应用的方法以经由网络 104 将应用流式传输到客户机 102。

[0056] 客户机 102 可以执行、操作或以其他方式提供应用，所述应用可以是任一类型和 / 或形式的软件、程序或可执行指令，例如任一类型和 / 或形式的 web 浏览器、基于 web 的客户机、客户机 - 服务器应用、瘦 - 客户机的计算客户机、ActiveX 控件、或 Java 小程序、或可以在客户机 102 上执行的任一其它类型和 / 或形式的可执行指令。在一些实施例中，应用可以是代表客户机 102 在服务器 106 上执行的基于服务器或基于远程的应用。在一个实施例中，服务器 106 可以使用任一瘦 - 客户机或远程显示协议来显示输出到客户机 102，所述远程显示协议例如由位于 Ft. Lauderdale，Florida 的 Citrix Systems 公司出品的独立计算架构 (ICA) 协议或由位于 Redmond，Washington 的微软公司出品的远程桌面协议 (RDP)。应用可以使用任一类型的协议，并且它可以是例如 HTTP 客户机、FTP 客户机、Oscar 客户机或 Telnet 客户机。在其它实施例中，应用包括与 VoIP 通信相关的任一类型的软件，例如软 IP 电话。在进一步的实施例中，应用包括与实时数据通信相关的任一应用，例如用于流式传输视频和 / 或音频的应用。

[0057] 在一些实施例中，服务器 106 或服务器群组 38 可以运行一个或多个应用，例如提供瘦 - 客户机计算的应用或远程显示表示应用的应用。在一个实施例中，服务器 106 或服务器群组 38 作为应用而执行 Citrix Systems 公司的例如 MetaFrame 或 Citrix 表示服务器™ 的 Citrix 访问套件 ™ 的任一部分和 / 或由微软公司出品的任意一种微软® Windows 终端服务。在一个实施例中，应用是由位于 Fort Lauderdale，Florida 的 Citrix Systems 公司开发的 ICA 客户机。在其它实施例中，应用包括由位于 Redmond，Washington 的微软公司开发的远程桌面 (RDP) 客户机。此外，服务器 106 可以运行应用，例如，所述服务器 106 可以是提供例如由位于 Redmond，Washington 的微软公司出品的微软 Exchange 的电子邮件服务的

应用服务器、web 或 Internet 服务器、或桌面共享服务器、或协作服务器。在一些实施例中，任意一种应用可以包括任一类型的寄载服务或产品，例如由 Santa Barbara，California 的 Citrix Online 部门提供的 GoToMeeting™、由位于 Santa Clara，California 的 WebEx 公司提供的 WebEx™、或由位于 Redmond，Washington 的微软公司提供的微软 OfficeLive Meeting。

[0058] 仍然参考图 1D，网络环境的一个实施例可以包括监控服务器 106A。监控服务器 106A 可以包括任一类型和形式的性能监控业务 198。性能监控业务 198 可以包括监控、测量和 / 或管理软件和 / 或硬件，包括数据收集、集合、分析、管理和报告。在一个实施例中，性能监控业务 198 包括一个或多个监控代理 197。监控代理 197 包括用于在诸如客户机 102、服务器 106 或设备 200 和 205 的装置上执行监控、测量和数据收集活动的任一软件、硬件或其组合。在一些实施例中，监控代理 197 包括诸如 Visual Basic 脚本或 Java 描述语言的任一类型和形式的脚本。在一个实施例中，监控代理 197 相对于装置的任一应用和 / 或用户透明地执行。在一些实施例中，监控代理 197 相对于应用或客户机不引人注目地被安装和操作。在又一个实施例中，监控代理 197 被安装和操作而不需要用于该应用或装置的任何设备 (instrumentation)。

[0059] 在一些实施例中，监控代理 197 以预定频率监控、测量和收集数据。在其它实施例中，监控代理 197 基于任一类型和形式的事件的检测来监控、测量和收集数据。例如，监控代理 197 可以在检测到对 web 页面的请求或收到 HTTP 响应时收集数据。在另一个实例中，监控代理 197 可以在检测到诸如鼠标点击的任一用户输入事件时收集数据。监控代理 197 可以报告或提供任一所监控、测量或收集的数据给监控业务 198。在一个实施例中，监控代理 197 根据调度或预定频率来发送信息给监控业务 198。在另一个实施例中，监控代理 197 在检测到事件时发送信息给监控业务 198。

[0060] 在一些实施例中，监控业务 198 和 / 或监控代理 197 执行诸如客户机、服务器、服务器群组、设备 200、设备 205 或网络连接的任一网络资源或网络基础结构元件的监控和性能测量。在一个实施例中，监控业务 198 和 / 或监控代理 197 执行诸如 TCP 或 UDP 连接的任一传输层连接的监控和性能测量。在另一个实施例中，监控业务 198 和 / 或监控代理 197 监控和测量网络等待时间。在又一个实施例中，监控业务 198 和 / 或监控代理 197 监控和测量带宽利用。

[0061] 在其它实施例中，监控业务 198 和 / 或监控代理 197 监控和测量终端用户响应时间。在一些实施例中，监控业务 198 执行应用的监控和性能测量。在另一个实施例中，监控业务 198 和 / 或监控代理 197 执行到应用的任一会话或连接的监控和性能测量。在一个实施例中，监控业务 198 和 / 或监控代理 197 监控和测量浏览器的性能。在另一个实施例中，监控业务 198 和 / 或监控代理 197 监控和测量基于 HTTP 的事务的性能。在一些实施例中，监控业务 198 和 / 或监控代理 197 监控和测量 IP 上语音 (VoIP) 应用或会话的性能。在其它实施例中，监控业务 198 和 / 或监控代理 197 监控和测量诸如 ICA 客户机或 RDP 客户机的远程显示协议应用的性能。在又一个实施例中，监控业务 198 和 / 或监控代理 197 监控和测量任一类型和形式的流媒体的性能。在进一步的实施例中，监控业务 198 和 / 或监控代理 197 监控和测量寄载应用或软件即服务 (Software-As-A-Service，SaaS) 递送模型的性能。

[0062] 在一些实施例中，监控业务 198 和 / 或监控代理 197 执行与应用相关的一个或多个事务、请求或响应的监控和性能测量。在其它实施例中，监控业务 198 和 / 或监控代理 197 监控和测量应用层堆栈的任一部分，例如任一 .NET 或 J2EE 调用。在一个实施例中，监控业务 198 和 / 或监控代理 197 监控和测量数据库或 SQL 事务。在又一个实施例中，监控业务 198 和 / 或监控代理 197 监控和测量任一方法、函数或应用编程接口 (API) 调用。

[0063] 在一个实施例中，监控业务 198 和 / 或监控代理 197 执行经由诸如设备 200 和 / 或设备 205 的一个或多个设备从服务器到客户机的应用和 / 或数据的递送的监控和性能测量。在一些实施例中，监控业务 198 和 / 或监控代理 197 监控和测量虚拟化应用的递送的性能。在其它实施例中，监控业务 198 和 / 或监控代理 197 监控和测量流式应用的递送的性能。在另一个实施例中，监控业务 198 和 / 或监控代理 197 监控和测量递送桌面应用到客户机和 / 或在客户机上执行桌面应用的性能。在另一个实施例中，监控业务 198 和 / 或监控代理 197 监控和测量客户机 / 服务器应用的性能。

[0064] 在一个实施例中，监控业务 198 和 / 或监控代理 197 被设计和构建为应用递送系统 190 提供应用性能管理。例如，监控业务 198 和 / 或监控代理 197 可以监控、测量和管理经由 Citrix 表示服务器递送应用的性能。在该实例中，监控业务 198 和 / 或监控代理 197 监控单独的 ICA 会话。监控业务 198 和 / 或监控代理 197 可以测量总的以及每次的会话系统资源使用，以及应用和连网性能。监控业务 198 和 / 或监控代理 197 可以对于给定用户和 / 或用户会话来标识有效服务器。在一些实施例中，监控业务 198 和 / 或监控代理 197 监控在应用递送系统 190 和应用和 / 或数据库服务器之间的后端连接。监控业务 198 和 / 或监控代理 197 可以测量每个用户会话或 ICA 会话的网络等待时间、延迟和容量。

[0065] 在一些实施例中，监控业务 198 和 / 或监控代理 197 测量和监控对于应用递送系统 190 的诸如总的存储器使用、每个用户会话和 / 或每个进程的存储器使用。在其它实施例中，监控业务 198 和 / 或监控代理 197 测量和监控诸如总的 CPU 使用、每个用户会话和 / 或每个进程的应用递送系统 190 的 CPU 使用。在另一个实施例中，监控业务 198 和 / 或监控代理 197 测量和监控登录到诸如 Citrix 表示服务器的应用、服务器或应用递送系统所需的时间。在一个实施例中，监控业务 198 和 / 或监控代理 197 测量和监控用户登录应用、服务器或应用递送系统 190 的持续时间。在一些实施例中，监控业务 198 和 / 或监控代理 197 测量和监控应用、服务器或应用递送系统会话的有效和无效的会话计数。在又一个实施例中，监控业务 198 和 / 或监控代理 197 测量和监控用户会话等待时间。

[0066] 在又一个进一步的实施例中，监控业务 198 和 / 或监控代理 197 测量和监控任一类型和形式的服务器规格 (metrics)。在一个实施例中，监控业务 198 和 / 或监控代理 197 测量和监控与系统存储器、CPU 使用和磁盘存储器有关的规格。在另一个实施例中，监控业务 198 和 / 或监控代理 197 测量和监控和页错误有关的规格，诸如每秒页错误。在其它实施例中，监控业务 198 和 / 或监控代理 197 测量和监控往返时间的规格。在又一个实施例中，监控业务 198 和 / 或监控代理 197 测量和监控与应用崩溃、错误和 / 或中止相关的规格。

[0067] 在一些实施例中，监控业务 198 和监控代理 198 包括由位于 Ft.Lauderdale，Florida 的 Citrix Systems 公司出品的被称为 EdgeSight 的任意一种产品实施例。在另一个实施例中，性能监控业务 198 和 / 或监控代理 198 包括由位于 Palo Alto，California 的 Symphoniq 公司出品的被称为 TrueView 产品套件的产品实施例的任一部分。在一个实

施例中，性能监控业务 198 和 / 或监控代理 198 包括由位于 SanFrancisco，California 的 TeaLeaf 技术公司出品的被称为 TeaLeafCX 产品套件的产品实施例的任一部分。在其它实施例中，性能监控业务 198 和 / 或监控代理 198 包括由位于 Houston，Texas 的 BMC 软件公司出品的诸如 BMC 性能管理器和巡逻产品 (BMC Performance Manager and Patrolproducts) 的商业业务管理产品的任一部分。

[0068] 客户机 102、服务器 106 和设备 200 可以被部署和 / 或执行在任一类型和形式的计算装置上，例如可以在任一类型和形式的网络上通信并执行此处描述的操作的计算机、网络装置或设备。图 1E 和 1F 描述了可用于实施客户机 102、服务器 106 或设备 200 的实施例的计算装置 100 的框图。如图 1E 和 1F 所示，每个计算装置 100 包括中央处理单元 101 和主存储器单元 122。如图 1E 所示，计算装置 100 可以包括可视显示装置 124、键盘 126 和 / 或诸如鼠标的点击装置 127。每个计算装置 100 也可以包括另外的可选元件，例如一个或多个输入 / 输出装置 130a-130b（通常使用附图标记 130 来指示）以及与中央处理单元 101 通信的高速缓存 140。

[0069] 中央处理单元 101 是响应并处理取自主存储器单元 122 的指令的任一逻辑电路。在许多实施例中，中央处理单元由微处理器单元提供，例如：由位于 Mountain View，California 的 Intel 公司出品的产品；由位于 Schaumburg，Illinois 的 Motorola 公司出品的产品；由位于 SantaClara，California 的 Transmeta 公司出品的产品；由位于 WhitePlains，New York 的国际商业机器公司出品的 RS/6000 处理器；或者由位于 Sunnyvale，California 的 Advanced Micro Devices 公司出品的产品。计算装置 100 可以基于任一的这些处理器、或者可以如此处所描述地操作的任一其它处理器。

[0070] 主存储器单元 122 可以是可以保存数据并允许由微处理器 101 直接访问的任一存储位置的一个或多个存储芯片，例如静态随机存取存储器 (SRAM)、突发式 SRAM 或同步突发式 SRAM(BSRAM)、动态随机存取存储器 (DRAM)、快速页面模式 DRAM(FPM DRAM)、增强型 DRAM(EDRAM)、扩展数据输出 RAM(EDO RAM)、扩展数据输出 DRAM(EDO DRAM)、突发式扩展数据输出 DRAM(BEDO DRAM)、增强型 DRAM(EDRAM)、同步 DRAM(SDRAM)、JEDEC SRAM、PC100SDRAM、双数据速率 SDRAM(DDR SDRAM)、增强型 SDRAM(ESDRAM)、同步链接 DRAM(SLDRAM)、直接 Rambus DRAM(DRDRAM)、或铁电 RAM(FRAM)。主存储器 122 可以基于任意一种上面描述的存储芯片、或者可以如此处所描述地操作的任一其它可用的存储芯片。在图 1E 中所示的实施例中，处理器 101 通过系统总线 150（在下面进行更详细的描述）与主存储器 122 进行通信。图 1E 描述了在其中处理器通过存储器端口 103 直接与主存储器 122 通信的计算装置 100 的实施例。例如，在图 1F 中，主存储器 122 可以是 DRDRAM。

[0071] 图 1F 描述了在其中主处理器 101 通过有时被称为背端总线的次级总线来直接与高速缓存 140 通信的实施例。在其它实施例中，主处理器 101 使用系统总线 150 与高速缓存 140 进行通信。高速缓存 140 典型地具有比主存储器 122 更快的响应时间，并且典型地通过 SRAM、BSRAM 或 EDRAM 来提供。在图 1E 中所示的实施例中，处理器 101 通过本地系统总线 150 与多个 I/O 装置 130 进行通信。多种总线可以用来将中央处理单元 101 连接到任意一种 I/O 装置 130，所述总线包括 VESA VL 总线、ISA 总线、EISA 总线、微通道架构 (MCA) 总线、PCI 总线、PCI-X 总线、PCI-Express 总线或 NuBus。对于 I/O 装置是视频显示器 124 的实施例，处理器 101 可以使用高级图形端口 (AGP) 来与显示器 124 进行通信。图 1F 描述

了在其中主处理器 101 通过 HyperTransport、快速 I/O 或 InfiniBand 来直接与 I/O 装置 130 通信的计算机 100 的一个实施例。图 1F 还描述了混合本地总线和直接通信的一个实施例：处理器 101 使用本地互连总线与 I/O 装置 130 进行通信，同时直接与 I/O 装置 130 进行通信。

[0072] 计算装置 100 可以支持任一适当的安装装置 116，例如用于接收像 3.5 英寸、5.25 英寸磁盘或 ZIP 磁盘这样的软盘的软盘驱动器、CD-ROM 驱动器、CD-R/Rw 驱动器、DVD-ROM 驱动器、多种格式的磁带驱动器、USB 装置、硬盘驱动器或适于安装像任一客户机代理 120 或其部分的软件和程序的任一其它装置。计算装置 100 还可以包括存储装置 128，例如一个或多个硬盘驱动器或独立磁盘的冗余阵列，用于保存操作系统及其它相关软件，以及用于保存诸如与客户机代理 120 相关的任一程序的应用软件程序。可选地，任意一种安装装置 116 还可以被用作存储装置 128。另外，操作系统和软件可以从可引导介质中运行，所述可引导介质例如像 KNOPPIX ®的可引导 CD，作为来自于 knoppix.net 可用作 GNU/Linux 分发的 GNU/Linux 的可引导 CD。

[0073] 进一步地，计算装置 100 可以包括通过多种连接联接到局域网 (LAN)、广域网 (WAN) 或因特网的网络接口 118，所述多种连接包括但不限于标准电话线、LAN 或 WAN 链路 (例如，802.11、T1、T3、56kb、X.25)、宽带连接 (例如，ISDN、帧中继、ATM)、无线连接或上述任一或所有连接的一些组合。网络接口 118 可以包括内置网络适配器、网络接口卡、PCMCIA 网卡、插件总线网络适配器、无线网络适配器、USB 网络适配器、调制解调器或适于将计算装置 100 连接到可以传达并执行此处所描述的操作的任一类型的网络的任一其它装置。各式各样的 I/O 装置 130a-130n 可以存在于计算装置 100 中。输入装置包括键盘、鼠标、轨道垫、轨道球、麦克风以及绘画板。输出装置包括视频显示器、扬声器、喷墨打印机、激光打印机和染料升华打印机。I/O 装置 130 可以由如图 1E 所示的 I/O 控制器 123 控制。I/O 控制器可以控制诸如键盘 126 和例如鼠标或光笔的点击装置 127 的一个或多个 I/O 装置。进一步地，I/O 装置还可以为计算装置 100 提供存储装置 128 和 / 或安装介质 116。还是在其它实施例中，计算装置 100 可以提供 USB 连接以接收诸如由位于 LosAlamitos，California 的 Twintech Industry 公司出品的 USB 闪存驱动器系列装置这样的便携 USB 存储装置。

[0074] 在一些实施例中，计算装置 100 可以包括或连接到多个显示装置 124a-124n，每个显示装置可以是相同或不同的类型和 / 或形式。因而，任意一种 I/O 装置 130a-130n 和 / 或 I/O 控制器 123 可以包括任一类型和 / 或形式的适当的硬件、软件或硬件和软件的组合，以支持、允许或提供通过计算装置 100 连接和使用多个显示装置 124a-124h。例如，计算装置 100 可以包括任一类型和 / 或形式的视频适配器、视频卡、驱动程序和 / 或库，以联系、通信、连接或以其他方式使用显示装置 124a-124n。在一个实施例中，视频适配器可以包括多个连接器以联接多个显示装置 124a-124n。在其它实施例中，计算装置 100 可以包括多个视频适配器，每个视频适配器连接到一个或多个显示装置 124a-124n。在一些实施例中，计算装置 100 的操作系统的任一部分可以被配置用于使用多个显示器 124a-124n。在其它实施例中，一个或多个显示装置 124a-124n 可以由一个或多个诸如例如通过网络连接到计算装置 100 的计算装置 100a 和 100b 的其它的计算装置来提供。这些实施例可以包括被设计和构建为将另一个计算机的显示装置用作计算装置 100 的第二显示装置 124a 的任一类型的软件。本领域普通技术人员将认识和理解计算装置 100 可以被配置为具有多个显示装置

124a-124n 的多个方法和实施例。

[0075] 在进一步的实施例中，I/O 装置 130 可以是在系统总线 150 和外部通信总线之间的网桥 170，所述外部通信总线例如 USB 总线、Apple Desktop 总线、RS-232 串行连接、SCSI 总线、FireWire 总线、FireWire 800 总线、以太网总线、AppleTalk 总线、吉比特以太网总线、异步传送模式总线、HIPPI 总线、超 HIPPI 总线、SerialPlus 总线、SCI/LAMP 总线、FibreChannel 总线或串行附加小型计算机系统接口总线。

[0076] 图 1E 和 1F 中描述类型的计算装置 100 典型地在控制任务的调度和对系统资源的访问的操作系统的控制下操作。计算装置 100 可以运行任一操作系统，例如任意一种版本的微软® Windows 操作系统、不同版本的 Unix 和 Linux 操作系统、用于 Macintosh 计算机的任一版本的 Mac OS ®、任一的嵌入式操作系统、任一的实时操作系统、任一的开放源操作系统、任一的专用操作系统、用于移动计算装置的任一操作系统、或者可以运行在计算装置上并执行此处所描述的操作的任一其它操作系统。典型的操作系统其中包括 :WINDOWS 3.x、WINDOWS 95、WINDOWS 98、WINDOWS 2000、WINDOWS NT 3.51、WINDOWS NT 4.0、WINDOWS CE 和 WINDOWS XP，所有这些均由位于 Redmond，Washington 的微软公司出品 ；由位于 Cupertino，California 的苹果计算机出品的 MacOS ；由位于 Armonk，NewYork 的国际商业机器公司出品的 OS/2 ；以及由位于 Salt Lake City，Utah 的 Caldera 公司发布的可免费使用的 Linux 操作系统或者任一类型和 / 或形式的 Unix 操作系统，以及其它。

[0077] 在其它实施例中，计算装置 100 可以具有和所述装置一致的不同的处理器、操作系统和输入装置。例如，在一个实施例中，计算机 100 是由 Palm 公司出品的 Treo180、270、1060、600 或 650 智能电话。在该实施例中，Treo 智能电话在 PalmOS 操作系统的控制下操作，并包括指示笔输入装置以及五向导航装置。此外，计算装置 100 可以是任一工作站、台式计算机、膝上型或笔记本计算机、服务器、便携计算机、移动电话、任一其它计算机、或者可以通信并具有执行此处所描述的操作的足够的处理器能力和存储容量的其它形式的计算或电信装置。

[0078] B. 设备架构

[0079] 图 2A 举例说明了设备 200 的一个示例实施例，其还可以称为中间设备 200、代理或者网络浏览器 (Netscaler)。提供图 2A 中的设备 200 的架构仅仅是为了说明，并不是意于进行限制。如图 2 所示，设备 200 包括硬件层 206 和被分为用户空间 202 和内核空间 204 的软件层。

[0080] 硬件层 206 提供在其上执行内核空间 204 和用户空间 202 中的程序和服务的硬件元件。硬件层 206 还提供允许内核空间 204 和用户空间 202 中的程序和服务关于设备 200 的向内和向外传递数据的结构和元件。如图 2 所示，硬件层 206 包括用于执行软件程序和服务的处理单元 262、用于保存软件和数据的存储器 264、用于在网络上发送和接收数据的网络端口 266 以及用于执行与在网络上发送和接收的数据的安全套接字层处理相关的功能的加密处理器 260。在一些实施例中，中央处理单元 262 可以在单个的处理器中执行加密处理器 260 的功能。另外，硬件层 206 可以包括用于每个处理单元 262 和加密处理器 260 的多个处理器。处理器 262 可以包括如上所述的与图 1E 和 1F 有关的任一处理器 101。在一些实施例中，中央处理单元 262 可以在单个的处理器中执行加密处理器 260 的功能。另外，硬件层 206 可以包括用于每个处理单元 262 和加密处理器 260 的多个处理器。例如，在

一个实施例中，设备 200 包括第一处理器 262 和第二处理器 262′ 。在其它实施例中，处理器 262 或 262′ 包括多核处理器。

[0081] 虽然通常所示设备 200 的硬件层 206 具有加密处理器 260，但处理器 260 可以是用于执行与诸如安全套接字层 (SSL) 或传输层安全 (TLS) 协议的任一加密协议相关的功能的处理器。在一些实施例中，处理器 260 可以是通用处理器 (GPP)，并且在进一步的实施例中，可以具有用于执行任一安全相关协议的处理的可执行指令。

[0082] 虽然在图 2 中用某些元件来说明设备 200 的硬件层 206，但设备 200 的硬件部分或部件可以包括计算装置的任一类型和形式的元件、硬件或软件，诸如此处结合图 1E 和 1F 来举例说明和讨论的计算装置 100。在一些实施例中，设备 200 可以包括服务器、网关、路由器、交换机、网桥或其它类型的计算或网络装置，并具有与此相关的任一硬件和 / 或软件元件。

[0083] 设备 200 的操作系统将可用的系统存储器分配、管理或者以其他方式分离成内核空间 204 和用户空间 204。在示例的软件架构 200 中，操作系统可以是任一类型和 / 或形式的 Unix 操作系统，尽管本发明并未这样限制。因而，设备 200 可以运行任一操作系统，例如任意一种版本的微软® Windows 操作系统、不同版本的 Unix 和 Linux 操作系统、用于 Macintosh 计算机的任一版本的 Mac OS ®、任一的嵌入式操作系统、任一的网络操作系统、任一的实时操作系统、任一的开放源操作系统、任一的专用操作系统、用于移动计算装置或网络装置的任一操作系统、或者可以运行在设备 200 上并执行此处所描述的操作的任一其它操作系统。

[0084] 内核空间 204 被保留用于运行内核 230，所述内核 230 包括任一设备驱动程序、内核扩展或其它内核相关软件。如本领域技术人员所知，内核 230 是操作系统的核心，并提供对应用 104 的资源和硬件相关的元件的访问、控制和管理。根据设备 200 的实施例，内核空间 204 还包括和有时还被称为集成高速缓存的高速缓存管理器 232 一起工作的多个网络服务或进程，此处进一步详细描述其有益之处。另外，内核 230 的实施例将依赖于由装置 200 所安装、配置或者以其他方式使用的操作系统的实施例。

[0085] 在一个实施例中，装置 200 包括诸如基于 TCP/IP 的堆栈的一个网络堆栈 267，用于与客户机 102 和 / 或服务器 106 进行通信。在一个实施例中，网络堆栈 267 用于与诸如网络 108 的第一网络以及第二网络 110 进行通信。在一些实施例中，装置 200 终止诸如客户机 102 的 TCP 连接的第一传输层连接，并建立由客户机 102 使用的到服务器 106 的第二传输层连接，例如，第二传输层连接在设备 200 和服务器 106 处终止。第一和第二传输层连接可以经由单个的网络堆栈 267 建立。在其它实施例中，装置 200 可以包括例如 267 和 267′ 的多个网络堆栈，并且第一传输层连接可以在一个网络堆栈 267 处建立或终止，而第二传输层连接在第二网络堆栈 267′ 上建立或终止。例如，一个网络堆栈可以用于在第一网络上接收和发送网络分组，而另一个网络堆栈用于在第二网络上接收和发送网络分组。在一个实施例中，网络堆栈 267 包括用于由设备 200 发送的一个或多个网络分组排队的缓冲器 243。

[0086] 如图 2 所示，内核空间 204 包括高速缓存管理器 232、高速层 2-7 集成分组引擎 240、加密引擎 234、策略引擎 236 和多协议压缩逻辑 238。在内核空间 204 或内核模式而不是用户空间 202 中单独以及组合地运行这些部件或进程 232、240、234、236 和 238 改进每一

个这些部件的性能。内核操作意味着这些部件或进程 232、240、234、236 和 238 运行在装置 200 的操作系统的核心地址空间中。例如，在内核模式中运行加密引擎 234 通过将加密与解密操作移到内核来改善加密性能，从而减少在内核模式中的存储空间或内核线程与用户模式中的存储空间或线程之间的转换的次数。例如，可以不需要将内核模式中获得的数据传递或复制到运行在用户模式中的进程或线程，例如从内核级的数据结构到用户级的数据结构。在另一个方面，还减少了在内核模式与用户模式之间的上下文转换的次数。另外，在内核空间 204 中可以更有效地执行在任意一个部件或进程 232、240、235、236 和 238 之间通信和通信的同步。

[0087] 在一些实施例中，部件 232、240、234、236 和 238 的任一部分可以运行或操作在内核空间 204 中，而这些部件 232、240、234、236 和 238 的其它部分可以运行或操作在用户空间 202 中。在一个实施例中，设备 200 使用提供对一个或多个网络分组的任一部分的访问的内核级数据结构，例如，网络分组包括来自于客户机 102 的请求或来自于服务器 106 的响应。在一些实施例中，可以由分组引擎 240 经由到网络堆栈 267 的传输层驱动程序接口或过滤器来获得内核级数据结构。内核级数据结构可以包括可经由与网络堆栈 267 相关的内核空间 204 存取的任一接口和 / 或数据、由网络堆栈 267 接收或传送的网络业务量或分组。在其它实施例中，可以由部件或进程 232、240、234、236 和 238 中的任意一个来使用内核级数据结构，以执行部件或进程的期望的操作。在一个实施例中，部件 232、240、234、236 和 238 在使用内核级数据结构时运行于内核模式 204 中，而在另一个实施例中，部件 232、240、234、236 和 238 在使用内核级数据结构时运行于用户模式中。在一些实施例中，可以将内核级数据结构复制或传递到第二内核级数据结构或任一期望的用户级数据结构。

[0088] 高速缓存管理器 232 可以包括软件、硬件或软件和硬件的任一组合，以提供对诸如由发信服务器 106 提供的对象或动态生成的对象的任一类型和形式的内容的高速缓存访问、控制和管理。由高速缓存管理器 232 处理和保存的数据、对象或内容可以包括诸如标记语言的或者通过任一协议传达的任一格式的数据。在一些实施例中，高速缓存管理器 232 复制存储在别处的原始数据或者以前计算、生成或发送的数据，其中原始数据也许需要相对于读取高速缓存元件来说更长的访问时间以取出、计算或者以其他方式获取。一旦数据被保存在高速缓存元件中，未来的使用可以通过访问高速缓存的拷贝而不是重新取回或再计算原始数据来进行，从而减少访问时间。在一些实施例中，高速缓存元件可以包括装置 200 的存储器 264 中的数据对象。在其它实施例中，高速缓存元件可以包括具有比存储器 264 更快的访问时间的存储器。在另一个实施例中，高速缓存元件可以包括诸如硬盘的一部分的装置 200 的任一类型和形式的存储元件。在一些实施例中，处理单元 262 可以提供由高速缓存管理器 232 使用的高速缓存。然而在进一步的实施例中，高速缓存管理器 232 可以使用存储器、存储装置或处理单元的任一部分和组合，以用于高速缓存数据、对象及其它内容。

[0089] 进一步地，高速缓存管理器 232 包括任一逻辑、功能、规则或操作，以执行此处所描述的设备 200 的技术的任一实施例。例如，高速缓存管理器 232 包括根据失效时间周期的期满或一旦从客户机 102 或服务器 106 接收到失效命令来使对象无效的逻辑或功能。在一些实施例中，高速缓存管理器 232 可以作为在内核空间 204 中执行的程序、服务、进程或任务来操作，而在其它实施例中是在用户空间 202 中操作。在一个实施例中，高速缓存管理

器 232 的第一部分在用户空间 202 中执行，而第二部分在内核空间 204 中执行。在一些实施例中，高速缓存管理器 232 可以包括任一类型的通用处理器 (GPP) 或者诸如现场可编程门阵列 (FPGA)、可编程逻辑器件 (PLD) 或应用专用集成电路 (ASIC) 的任一其它类型的集成电路。

[0090] 例如，策略引擎 236 可以包括智能统计引擎或者其它的可编程应用。在一个实施例中，策略引擎 236 提供配置机制以允许用户标识、指定、限定或配置高速缓存策略。在一些实施例中，策略引擎 236 还可以访问存储器以支持诸如查找表或哈希表的数据结构来启用用户选择的高速缓存策略决策。在其它实施例中，策略引擎 236 可以包括任一逻辑、规则、功能或操作，以便确定和提供除了由设备 200 执行的安全、网络业务量、网络访问、压缩或任一其它功能或操作的访问、控制和管理之外的由设备 200 高速缓存的对象、数据或内容的访问、控制和管理。此处进一步描述特定高速缓存策略的进一步的实例。

[0091] 加密引擎 234 包括用于操控诸如 SSL 或 TLS 的任一安全相关协议的处理的任一逻辑、商业规则、功能或操作，或者另外的任一相关功能。例如，加密引擎 234 加密并解密经由设备 200 传递的网络分组或者其中的任一部分。加密引擎 234 还可以为客户机 102a-102n、服务器 106a-106n 或设备 200 设置或建立 SSL 或 TLS 连接。因而，加密引擎 234 提供 SSL 处理的卸载和加速。在一个实施例中，加密引擎 234 使用隧道协议来在客户机 102a-102n 和服务器 106a-106n 之间提供虚拟专用网。在一些实施例中，加密引擎 234 与加密处理器 260 进行通信。在其它实施例中，加密引擎 234 包括运行在加密处理器 260 上的可执行指令。

[0092] 多协议压缩引擎 238 包括用于压缩诸如由装置 200 的网络堆栈 267 使用的任意一种协议的一个或多个协议的网络分组的任一逻辑、商业规则、功能或操作。在一个实施例中，多协议压缩引擎 238 双向地在客户机 102a-102n 和服务器 106a-106n 之间压缩任一的基于 TCP/IP 的协议，包括消息应用编程接口 (MAPI)（电子邮件）、文件传送协议 (FTP)、超文本传送协议 (HTTP)、通用 Internet 文件系统 (CIFS) 协议（文件传送）、独立计算架构 (ICA) 协议、远程桌面协议 (RDP)、无线应用协议 (WAP)、移动 IP 协议和 IP 上语音 (VoIP) 协议。在其它实施例中，多协议压缩引擎 238 提供基于超文本标记语言 (HTML) 的协议的压缩，并且在一些实施例中提供诸如可扩展标记语言 (XML) 的任一标记语言的压缩。在一个实施例中，多协议压缩引擎 238 提供诸如为设备 200 设计用于设备 200 通信的的任一协议的任一高性能协议的压缩。在另一个实施例中，多协议压缩引擎 238 使用修改的传输控制协议来压缩任一通信的任一有效载荷或任一通信，所述修改的传输控制协议诸如事务 TCP(T/TCP)、具有选择确认的 TCP(TCP-SACK)、具有大窗口的 TCP(TCP-LW)、诸如 TCP-Vegas 协议的拥塞预测协议以及 TCP 欺骗协议。

[0093] 因而，多协议压缩引擎 238 为经由桌面客户机以及甚至移动客户机访问应用的用户加速性能，所述桌面客户机例如微软 Outlook 以及诸如由诸如 Oracle、SAP 和 Siebel 的通用的企业应用所启动的任一客户机的非 web 瘦客户机，所述移动客户机例如掌上电脑。在一些实施例中，通过执行于内核模式 204 中以及与访问网络堆栈 267 的分组处理引擎 240 结合在一起，多协议压缩引擎 238 可以压缩诸如任一应用层协议的由 TCP/IP 协议所携带的任意一种协议。

[0094] 通常也被称为分组处理引擎或分组引擎的高速层 2-7 集成分组引擎 240 负责管理由设备 200 经由网络端口 266 接收和发送的分组的内核级处理。高速层 2-7 集成分组引擎

240 可以包括用于在例如接收网络分组或发送网络分组的处理期间排队一个或多个网络分组的缓冲器。另外，高速层 2-7 集成分组引擎 240 与一个或多个网络堆栈 267 通信以经由网络端口 266 发送和接收网络分组。高速层 2-7 集成分组引擎 240 和加密引擎 234、高速缓存管理器 232、策略引擎 236 和多协议压缩逻辑 238 一起工作。更具体地，加密引擎 234 被配置为执行分组的 SSL 处理，策略引擎 236 被配置为执行诸如请求级内容交换和请求级高速缓存重定向的与业务量管理相关的功能，而多协议压缩逻辑 238 被配置为执行与数据的压缩和解压缩相关的功能。

[0095] 高速层 2-7 集成分组引擎 240 包括分组处理定时器 242。在一个实施例中，分组处理定时器 242 提供一个或多个时间间隔以触发输入（即，接收）或输出（即，发送）网络分组的处理。在一些实施例中，高速层 2-7 集成分组引擎 240 响应于定时器 242 来处理网络分组。分组处理定时器 242 提供任一类型和形式的信号给分组引擎 240，以通知、触发或传达时间相关的事件、间隔或发生。在许多实施例中，分组处理定时器 242 以例如像 100 毫秒、50 毫秒或 25 毫秒这样的毫秒级来进行操作。例如，在一些实施例中，分组处理定时器 242 提供时间间隔或者以其他方式使高速层 2-7 集成分组引擎 240 以 10 毫秒的时间间隔来处理网络分组，而在其它实施例中按 5 毫秒的时间间隔，以及甚至在更进一步的实施例中短到 3、2 或 1 毫秒的时间间隔。在操作期间，高速层 2-7 集成分组引擎 240 可以与加密引擎 234、高速缓存管理器 232、策略引擎 236 和多协议压缩引擎 238 交互、集成或通信。因而，可以响应于分组处理定时器 242 和 / 或分组引擎 240 来执行加密引擎 234、高速缓存管理器 232、策略引擎 236 和多协议压缩逻辑 238 的任一逻辑、功能或操作。因此，可以以例如小于或等于 10 毫秒的时间间隔的通过分组处理定时器 242 提供的时间间隔的粒度来执行加密引擎 234、高速缓存管理器 232、策略引擎 236 和多协议压缩逻辑 238 的任一逻辑、功能或操作。例如，在一个实施例中，高速缓存管理器 232 可以响应于高速层 2-7 集成分组引擎 240 和 / 或分组处理定时器 242 来执行任一高速缓存对象的失效。在另一个实施例中，可以将高速缓存对象的满期或失效时间设置为与分组处理定时器 242 的时间间隔相同的粒度级，例如每 10 毫秒。

[0096] 与内核空间 204 不同，用户空间 202 是由用户模式应用或者以其他方式运行于用户模式的程序所使用的存储器区域或部分操作系统。用户模式应用可以不直接访问内核空间 204 而使用服务调用以访问内核服务。如图 2 所示，设备 200 的用户空间 202 包括图形用户界面 (GUI)210、命令行接口 (CLI)212、命令解释程序 (shell) 服务 214、健康监测程序 216 和守护服务 218。GUI 210 和 CLI 212 提供一个装置，通过所述装置，系统管理员或其它用户可以与设备 200 的操作相互作用并控制设备 200 的操作，例如通过设备 200 的操作系统，并且两者之一是用户空间 202 或内核空间 204。GUI 210 可以是任一类型和形式的图形用户界面，并且可以通过文本、图形或者以其他方式通过像浏览器的任一类型的程序或应用来呈现。CLI 212 可以是任一类型和形式的命令行或基于文本的接口，例如由操作系统提供的命令行。例如，CLI 212 可以包括命令解释程序，所述命令解释程序是允许用户与操作系统相互作用的工具。在一些实施例中，CLI 212 可以通过 bash、csh、tcsh 或 ksh 型命令解释程序来提供。命令解释程序服务 214 包括程序、服务、任务、进程或可执行指令以支持用户通过 GUI 210 和 / 或 CLI 212 与设备 200 或操作系统相互作用。

[0097] 健康监测程序 216 被用于监控、检查、报告和确保网络系统在正常工作以及用户

通过网络接收所请求的内容。健康监测程序 216 包括一个或多个程序、服务、任务、进程或可执行指令以提供用于监测设备 200 的任一活动的逻辑、规则、功能或操作。在一些实施例中，健康监测程序 216 拦截并检查经由设备 200 传递的任一网络业务量。在其它实施例中，健康监测程序 216 通过任一合适的方法和 / 或机制与一个或多个下列单元连接 :加密引擎 234、高速缓存管理器 232、策略引擎 236、多协议压缩逻辑 238、分组引擎 240、守护服务 218 和命令解释程序服务 214。因而，健康监测程序 216 可以调用任一应用编程接口 (API) 以确定设备 200 的任一部分的状态、状况或健康。例如，健康监测程序 216 可以周期性地查验或发送一个情况查询以检测程序、进程、服务或任务是否有效以及当前正在运行。在另一个实例中，健康监测程序 216 可以检查由任一程序、进程、服务或任务提供的任一状态、错误或历史记录，以确定设备 200 的任一部分的任一情况、状态或错误。

[0098] 守护服务 218 是连续或在后台运行并处理由设备 200 接收到的周期性服务请求的程序。在一些实施例中，守护服务可以将请求转发给其它程序或进程，例如酌情转发给另一个守护服务 218。如本领域技术人员所知，守护服务 218 可以无人监护地运行以执行诸如网路控制的连续的或周期性的全系统的功能或者执行任一期望的任务。在一些实施例中，一个或多个守护服务 218 运行在用户空间 202 中，而在其它实施例中，一个或多个守护服务 218 运行在内核空间中。

[0099] 现在参考图 2B，描述了设备 200 的另一个实施例。总的来说，设备 200 提供下列服务、功能或操作中的一个或多个 :用于一个或多个客户机 102 以及一个或多个服务器 106 之间的通信的 SSL VPN 连通性 280、交换 / 负载平衡 284、域名服务解析 286、加速 288 和应用防火墙 290。每个服务器 106 可以提供一个或多个网络相关的服务 270a-270n(称为服务 270)。例如，服务器 106 可以提供 HTTP 服务 270。设备 200 包括一个或多个虚拟服务器或虚拟网际协议服务器，其被称为 vServer、VIP 服务器或仅仅称为 VIP 275a-275n(此处也被称为 vServer 275)。vServer 275 根据设备 200 的配置和操作来接收、拦截或者以其他方式处理客户机 102 和服务器 106 之间的通信。

[0100] vServer 275 可以包括软件、硬件或软件和硬件的任一组合。vServer275 可以包括在设备 200 中的用户模式 202、内核模式 204 中或其任一组合中操作的任一类型和形式的程序、服务、任务、进程或可执行指令。vServer 275 包括任一逻辑、功能、规则或操作以执行此处所描述的技术的任一实施例，例如 SSL VPN 280、交换 / 负载平衡 284、域名服务解析 286、加速 288 和应用防火墙 290。在一些实施例中，vServer 275 建立到服务器 106 的服务 270 的连接。服务 275 可以包括可以连接和通信到设备 200、客户机 102 或 vServer 275 的任一程序、应用、进程、任务或可执行指令组。例如，服务 275 可以包括 web 服务器、HTTP 服务器、ftp、电子邮件或数据库服务器。在一些实施例中，服务 270 是用于监听、接收和 / 或发送用于诸如电子邮件、数据库或企业应用的应用的通信的守护进程或网络驱动程序。在一些实施例中，服务 270 可以在一个特定 IP 地址或 IP 地址和端口上进行通信。

[0101] 在一些实施例中，vServer 275 将策略引擎 236 的一个或多个策略应用到客户机 102 和服务器 106 之间的网络通信。在一个实施例中，策略与 VServer 275 有关。在另一个实施例中，策略基于一个用户或一组用户。在又一个实施例中，策略是全局的并且应用到一个或多个 vServers275a-275n 以及经由设备 200 通信的任一用户或用户组。在一些实施例中，策略引擎的策略有条件，在所述条件时根据诸如网际协议地址、端口、协议类型、报头或

分组中的字段的通信的任一内容或者诸如用户、用户组、vServer 275、传输层连接和/或客户机 102 或服务器 106 的标识或属性的通信上下文来应用策略。

[0102] 在其它实施例中，设备 200 与策略引擎 236 通信或连接以确定对远程用户或远程客户机 102 访问服务器 106 的计算环境 15、应用和/或数据文件的验证和/或授权。在另一个实施例中，设备 200 与策略引擎 236 通信或连接以确定对远程用户或远程客户机 102 的验证和/或授权，以使应用递送系统 190 递送计算环境 15、应用和/或数据文件的一个或多个。在又一个实施例中，设备 200 根据策略引擎 236 对远程用户或远程客户机 103 的验证和/或授权来建立 VPN 或 SSL VPN 连接。在一个实施例中，设备 102 根据策略引擎 236 的策略来控制网络业务量和通信会话的流量。例如，设备 200 可以根据策略引擎 236 来控制对计算环境 15、应用或数据文件的访问。

[0103] 在一些实施例中，vServer 275 建立诸如经由客户机代理 120 与客户机 102 的 TCP 或 UDP 连接的传输层连接。在一个实施例中，vServer 275 监听并接收来自于客户机 102 的通信。在其它实施例中，vServer 275 与客户服务器 106 建立诸如 TCP 或 UDP 连接的传输层连接。在一个实施例中，vServer 275 建立到运行在服务器 106 上的服务器 270 的网际协议地址和端口的传输层连接。在另一个实施例中，vServer 275 将到客户机 102 的第一传输层连接与到服务器 106 的第二传输层连接关联起来。在一些实施例中，vServer 275 建立到服务器 106 的传输层连接池并多路复用经由所述池化的传输层连接的客户机请求。

[0104] 在一些实施例中，设备 200 提供在客户机 102 和服务器 106 之间的 SSL VPN 连接 280。例如，第一网络 104 上的客户机 102 请求建立到第二网络 104′ 上的服务器 106 的连接。在一些实施例中，第二网络 104′ 是不可从第一网络 104 路由的。在其它实施例中，客户机 102 在公用网 104 上，而服务器 106 在诸如公司网的专用网 104′ 上。在一个实施例中，客户机代理 120 拦截第一网络 104 上的客户机 102 的通信，加密所述通信，并经由第一传输层连接发送所述通信到设备 200。设备 200 将第一网络 104 上的第一传输层连接关联到第二网络 104’上的到服务器 106 的第二传输层连接。设备 200 从客户机代理 102 接收被拦截的通信，解密所述通信，并经由第二传输层连接发送所述通信到第二网络 104 上的服务器 106。第二传输层连接可以是池化的传输层连接。因而，设备 200 提供在两个网络 104 和 104′ 之间用于客户机 102 的端到端安全传输层连接。

[0105] 虚拟专用网络 (VPN) 可以是使用诸如因特网的公用电信基础架构来为远程客户机、服务器或者其它通信装置提供诸如从公用网络到专用网络的访问或者连接的任一网络。虚拟专用网络 (VPN) 是使用诸如因特网的公用电信基础架构来为远程用户提供对企业或者专用网络的访问的方法。在一些实施例中，该访问经由加密或者隧穿是安全的。在一些实施例中，此处描述的中间设备提供从客户机的第一网络到服务器的第二网络的安全虚拟专用网络连接。

[0106] 安全套接字层 (SSL)VPN 可以使用 SSL 或者 TLS 或者任一其它类型和形式的安全协议来建立具有安全级的连接。在一些实施例中，SSL VPN 可以使用任意类型和形式的加密用于建立或者维持安全访问。SSL VPN 可以经由诸如使用 HTTPS（安全超文本传输协议）的浏览器来建立和/或访问。SSL VPN 可以通过支持 SSL 的浏览器或者应用来建立或者提供。

[0107] 可以通过使用基于客户机或者免客户机的方法来建立或者提供 SSLVPN 连接或者

会话。基于客户机的 SSL VPN 可以使用任一类型和形式客户机代理或者客户机 102 上任一软件相关的代理，来建立 SSL VPN 连接或者会话。例如，可以经由下载到客户机的 SSL VPN 客户机代理来提供基于客户机的 SSL VPN，诸如从设备下载的。客户机代理可以被设计并且配置为在客户机和设备或者服务器之间建立和提供 SSL VPN 功能性、连接和访问。

[0108] 免客户机 SSL VPN 可以是不使用下载并安装到客户机 102 的 SSL VPN 客户机代理、软件或者程序来建立 SSL VPN 连接或者会话的任一 SSL VPN。在一些实施例中，免客户机 SSL VPN 可以是不需要客户机 102 来安装或者执行被设计和构成为提供 SSL VPN 功能性的预定软件或者可执行文件以建立和另一个网络装置的 SSL VPN 连接的任一 SSL VPN。在一些实施例中，经由不下载或者不需要使用 VPN 或者 SSL VPN 客户机代理的 SSL 使能的浏览器来建立免客户机 SSL VPN。免客户机 SSL VPN 连接或者会话可以使用标准浏览器或者应用的协议和通信，诸如 SSL 使能的浏览器。免客户机 SSL VPN 连接或者会话可以通过此处描述的在第一网络和第二网络之间翻译、重写或者转换请求和响应的内容的中间设备或者应用来提供。

[0109] 在一个实施例中，设备 200 在虚拟专用网 104 上寄载客户机 102 的内联网网际协议或内联网 IP 282 地址。客户机 102 具有诸如第一网络 104 上的网际协议 (IP) 地址和 / 或主机名的本地网络标识符。当经由设备 200 连接到第二网络 104′ 时，设备 200 在第二网络 104′ 上为客户机 102 建立、分配或者以其他方式提供内联网 IP，其是诸如 IP 地址和 / 或主机名的网络标识符。使用客户机建立的内联网 IP 282，设备 200 在第二或专用网 104′ 上监听并接收指向客户机 102 的任一通信。在一个实施例中，设备 200 在第二专用网 104 上充当或代表客户机 102。例如，在另一个实施例中，vServer 275 监听并响应到客户机 102 的内联网 IP 282 的通信。在一些实施例中，如果第二网络 104′ 上的计算装置 100 发送请求，则设备 200 处理所述请求，就像它是客户机 102 一样。例如，设备 200 可以响应到客户机的内联网 IP 282 的查验。在另一个实例中，设备可以与第二网络 104 上的请求与客户机的内联网 IP 282 连接的计算装置 100 建立诸如 TCP 或 UDP 连接的连接。

[0110] 在一些实施例中，设备 200 为客户机 102 和服务器 106 之间的通信提供下列一个或多个加速技术 288 ：1) 压缩 ；2) 解压缩 ；3) 传输控制协议池 ；4) 传输控制协议多路复用 ；5) 传输控制协议缓冲 ；以及 6) 高速缓存。在一个实施例中，设备 200 通过打开与每个服务器 106 的一个或多个传输层连接并维持这些连接以允许客户机经由因特网的重复数据访问来减轻服务器 106 的由反复打开和关闭到客户机 102 的传输层连接所造成的大量处理负载。这个技术在这里被称为“连接池”。

[0111] 在一些实施例中，为了经由池化的传输层连接来无缝接合从客户机 102 到服务器 106 的通信，设备 200 通过在传输层协议级修改序号和确认号来转换或多路复用通信。这被称为“连接多路复用”。在一些实施例中，不需要应用层协议相互作用。例如，在到来分组（即，自客户机 102 接收的分组）的情况中，所述分组的源网络地址被改变为设备 200 的输出端口的网络地址，而目的网络地址被改变为目的服务器的网络地址。在发出分组（即，自服务器 106 接收的一个分组）的情况中，源网络地址被从服务器 106 的网络地址改变为设备 200 的输出端口的网络地址，而目的地址被从设备 200 的网络地址改变为请求的客户机 102 的网络地址。所述分组的序号和确认号也被转换为到客户机 102 的设备 200 的传输层连接上的客户机 102 所期待的序号和确认。在一些实施例中，传输层协议的分组校验和被

重新计算以解释这些转换。

[0112] 在另一个实施例中，设备 200 为客户机 102 和服务器 106 之间的通信提供交换或负载平衡功能 284。在一些实施例中，设备 200 根据层 4 或应用层请求数据来分配业务量并将客户机请求指向服务器 106。在一个实施例中，虽然网络分组的网络层或层 2 标识了目的服务器 106，但设备 200 通过作为传输层分组的有效载荷而携带的应用信息和数据来确定服务器 106 以分配网络分组。在一个实施例中，设备 200 的健康监测程序 216 监控服务器的健康以确定为其分配客户机的请求的服务器 106。在一些实施例中，如果设备 200 探测到服务器 106 不可用或具有超过预定阈值的负载，则设备 200 可以将客户机请求指向或分配到另一个服务器 106。

[0113] 在一些实施例中，设备 200 充当域名服务 (DNS) 解析器或者以其他方式提供对来自于客户机 102 的 DNS 请求的解析。在一些实施例中，设备拦截由客户机 102 发送的 DNS 请求。在一个实施例中，设备 200 响应具有设备 200 的 IP 地址或由设备 200 寄载的 IP 地址的客户机的 DNS 请求。在该实施例中，客户机 102 把给域名的网络通信发送到设备 200。在另一个实施例中，设备 200 响应具有第二设备 200′ 的 IP 地址或由第二设备 200′ 寄载的 IP 地址的客户机的 DNS 请求。在一些实施例中，设备 200 响应具有由设备 200 确定的服务器 106 的 IP 地址的客户机的 DNS 请求。

[0114] 在又一个实施例中，设备 200 为客户机 102 和服务器 106 之间的通信提供应用防火墙功能 290。在一个实施例中，策略引擎 236 提供用于检测和阻塞非法请求的规则。在一些实施例中，应用防火墙 290 防止拒绝服务 (DoS) 攻击。在其它实施例中，设备检查被拦截的请求的内容以识别和阻塞基于应用的攻击。在一些实施例中，规则 / 策略引擎 236 包括用于提供对多个种类和类型的基于 web 或因特网的脆弱点的保护的一个或多个应用防火墙或安全控制策略，例如下列的一个或多个 :1) 缓冲器溢出，2)CGI-BIN 参数操纵，3) 格式 / 隐藏字段操纵，4) 强制浏览，5)cookie 或会话中毒，6) 破译的访问控制表 (ACLs) 或弱的口令，7) 跨站点的脚本 (XSS)，8) 命令注入，9)SQL 注入，10) 错误触发敏感信息泄漏，11) 加密技术的不安全使用，12) 服务器误配置，13) 后门和调试选择，14)web 站点毁损，15) 平台或操作系统的脆弱点，以及 16) 零天攻击。在一个实施例中，对下列情况的一种或多种，应用防火墙 290 以检查或分析网络通信的形式来提供 HTML 格式字段的保护 :1) 返回所需的字段，2) 不允许附加字段，3) 只读和隐藏字段强制 (enforcement)，4) 下拉列表和单选按钮字段的一致，以及 5) 格式字段最大长度强制。在一些实施例中，应用防火墙 290 确保 cookies 不被修改。在其它实施例中，应用防火墙 290 通过强制实施合法 URL 来防止强制浏览。

[0115] 还是在其它实施例中，应用防火墙 290 保护在网络通信中包含的任一机密信息。应用防火墙 290 可以根据引擎 236 的规则或策略来检查或分析任一网络通信以识别网络分组的任一字段中的任一机密信息。在一些实施例中，应用防火墙 290 在网络通信中识别信用卡号、口令、社会保险号、姓名、病人代码、联系信息和年龄的一次或多次出现。网络通信的编码部分可以包括这些出现或机密信息。在一个实施例中，根据这些出现，应用防火墙 290 可以对网络通信采取策略行动，例如阻止网络通信的发送。在另一个实施例中，应用防火墙 290 可以重写、移除或者以其他方式掩盖这样识别出的出现或机密信息。

[0116] 仍然参考图 2B，设备 200 可以包括如上面结合图 1D 所讨论的性能监控代理 197。在一个实施例中，设备 200 从如图 1D 中所描述的监控业务 198 或监控服务器 106 中接收监

控代理 197。在一些实施例中，设备 200 在诸如磁盘的存储装置中保存监控代理 197，以用于递送给与设备 200 通信的任一客户机或服务器。例如，在一个实施例中，设备 200 在接收到建立传输层连接的请求时发送监控代理 197 给客户机。在其它实施例中，设备 200 在建立与客户机 102 的传输层连接时发送监控代理 197。在另一个实施例中，设备 200 在拦截或检测对 web 页面的请求时发送监控代理 197 给客户机。在又一个实施例中，设备 200 响应于监控服务器 198 的请求来发送监控代理 197 到客户机或服务器。在一个实施例中，设备 200 发送监控代理 197 到第二设备 200′ 或设备 205。

[0117] 在其它实施例中，设备 200 执行监控代理 197。在一个实施例中，监控代理 197 测量和监控在设备 200 上执行的任一应用、程序、进程、服务、任务或线程的性能。例如，监控代理 197 可以监控和测量 vServers275A-275N 的性能与操作。在另一个实施例中，监控代理 197 测量和监控设备 200 的任一传输层连接的性能。在一些实施例中，监控代理 197 测量和监控通过设备 200 的任一用户会话的性能。在一个实施例中，监控代理 197 测量和监控通过设备 200 的诸如 SSL VPN 会话的任一虚拟专用网连接和 / 或会话的性能。在进一步的实施例中，监控代理 197 测量和监控设备 200 的存储器、CPU 和磁盘使用以及性能。在又一个实施例中，监控代理 197 测量和监控诸如 SSL 卸载、连接池和多路复用、高速缓存以及压缩的由设备 200 执行的任一加速技术 288 的性能。在一些实施例中，监控代理 197 测量和监控由设备 200 执行的任一负载平衡和 / 或内容交换 284 的性能。在其它实施例中，监控代理 197 测量和监控由设备 200 执行的应用防火墙 290 保护和处理的性能。

[0118] C. 免客户机虚拟专用网络环境

[0119] 现在参考图 3A，描述了用于经由设备 200 或代理访问服务器的免客户机虚拟专用网络 (VPN) 环境的实施例。总的来说，客户机 102 操作在计算装置 100 上并且执行通过用户操作的浏览器。客户机 102 可以在第一网络 104 上，诸如公用网络。客户机 102 上的用户可以经由浏览器来请求对于第二网络 104’上的资源的访问，诸如企业的专用网络。设备 200 为用户提供对于所请求资源的免客户机 VPN 访问。客户机可以不安装、执行或者以其他方式执行被构建和 / 或设计来为网络 104’提供 VPN 连接性 (称为基于客户机的 VPN) 的代理、部件、程序、驱动器或者应用。而是，设备或者代理可以重写来自服务器的响应和来自客户机的请求以提供 VPN 功能，而不需要使用对在客户机上操作的的 VPN 代理。例如，设备可以重写客户机和服务器之间的统一资源定位符 (URL)，诸如通过服务器对任一内容服务器中的 URL 或者客户机传送的请求中的 URL 进行重写。设备 200 可以以对于客户机和服务器的任一者或者两者透明并且无缝的方式重写服务器和客户机之间的 URL。由此，客户机、浏览器或者服务器和服务器应用不需要知晓或者了解免客户机 SSL VPN 访问方案。

[0120] 设备 200 可以经由之前描述的 SSL VPN280 模块来提供用于访问资源的功能。在一个实施例中，设备 200 通过在用于和设备 200 通信的客户机 102 上提供、安装或者执行 SSL VPN 代理来提供基于客户机的对网络的访问。在一些实施例中，设备 200 提供对资源 (诸如 http/https/ 文件共享) 的免客户机 SSL VPN 访问，而不需要下载 SSL VPN 客户机或者代理到客户机 102。例如，用户可以期望从诸如处于机房的外部机器来访问公司内的资源，在该外部机器上用户并没有特权来安装客户机或者不期望经历客户机安装过程。当装置 (例如市场上的新的 PDA) 不支持 SSL VPN 客户机而装置运行的是 SSL 使能的浏览器，则免客户机 SSL VPN 特征也是有用的。在其他实施例中，设备 200 基于策略和任一策略规则、动作和 /

或条件来在对资源的基于客户机和免客户机 SSL VPN 访问之间选择用户。

[0121] 客户机可以包括任一类型和形式的用户代理，可以是浏览器、编辑器、网络爬虫程序 (web 穿越自动机) 或者任一其它终端用户工具或者程序。客户机 102 可以包括任一类型和形式的浏览器。在一个实施例中，浏览器是 Washingtong Redmond 的微软公司制备的任一版本的 InternetExplorer(IE)。在另一个实施例中，浏览器是网景通信公司制备的任一版本的网景浏览器。在其他实施例中，浏览器是称之为 Firefox 并且由 California 的 Mozilla Foundation 提供的并且在 www.mozilla.com 可以找到的任一版本的开放源浏览器。在又一个实施例中，浏览器是 NorwayOslo 的 Opera Software ASA 制备的称之为 Opera 的任一版本的浏览器。在一些实施例中，客户机 102 执行或者包括任一类型或者形式的应用或者程序，用于显示 web 页面、web 内容、HTML、XML、CSS(层叠式样式表)、Java 脚本或者 HTTP 内容。

[0122] 在图 3A 描述的实施例的操作中，用户登入设备 200 提供的 SSL VPN 站点，诸如通过设备 200 寄载的域名和 IP 地址。例如，用户经由客户机 102 的浏览器可以选择或者输入 URL 到 SSL VPN 站点。设备 200 可以验证用户并且还可以进一步确定用户访问设备 200 或者 SSL VPN站点的授权。在成功验证之后，设备为客户机提供入口页面来经由浏览器显示给用户。入口页面可以包括导航盒 (navigation box)，诸如一组一个或者多个用户接口元件用于用户来选择操作或者运行应用。入口页面可以包括到用户可访问的其它页面或者 URL 的链接。入口页面上的 URL 或者链接可以索引或者识别设备 200 提供的 SSL VPN 站点的主机名或者 IP 地址。

[0123] 用户经由入口页面可以例如通过点击有效超链接或者 URL 来选择一个或者多个 URL。随之，浏览器或者客户机将请求传送给设备 200 寄载的域。例如，如图 3A 中描述的，用户可以经由设备请求服务器 106 的应用 :“https://sslvpn.x.com/cvpn/http/server.x.com/app.cgi”。在一些实施例，用户发送另一个请求，诸如“https://proxy.x.com/cvpn/http/server.x.com/app.cgi”。设备 200 从客户机 102 接收请求并且重写该请求以发送给服务器。例如，如图 3A 中描述的，设备可以移除或者删除设备所寄载的域名诸如“sslvpn.x.com”或者“proxy.x.com”并且将请求的剩余部分转发给服务器 106。

[0124] 响应于该请求，服务器将内容发送给客户机。响应的内容或者体可以包括到服务器的其他页面或者到网络 104’ 上的其他服务器的嵌入式链接或者 URL，诸如到“http://server.x.com/app.cgi”的嵌入式链接。设备重写该首部和体来修改任一URL，以索引到SSL VPN 站点的域名或者 IP 地址，使得经由客户机浏览器的任一其它 URL 或者链接选择将请求发送给设备 200。设备发送修改后的内容给客户机 102。设备 200 诸如经由 AppFw 290(有时称之为 AppSecure 模块 290) 可以设计并且构建为基于策略引擎的策略来重写请求和响应的 URL。该页面和在此 SSL VPN 会话期间从服务器随后接收的其他页面中的链接 (URL) 由设备通过指向 SSL VPN 站点 (VPNVIP 275) 的链接和初始请求 URL(绝对或者相对) 编码在该请求 URL 中的方式来进行修改。

[0125] 现在参考图 3B，描述用于提供 VPN 访问以及 cookie 管理的 VPN 环境的另一个实施例。总的来说，设备 200 可以包括用于处理如此处描述的基于免客户机和 / 或客户机的任一 SSL VPN 功能性的 VPN 模块 280。设备和 / 或 VPN 模块 280 可以具有 AAA 模块来执行任一类型和形式的验证、授权和审核 (AAA) 和 / 或跟综和管理 VPN 会话信息。AAA 模块还可以

执行任一类型或者形式的 VPN 会话查询来确定用于任一客户机请求的 VPN 会话。VPN 模块还可以执行 URL 译码并且将 URL 转换为服务器格式，诸如用来提交给专用网络上的服务器。VPN 模块 280 还包括经由 VPN 处理器函数、逻辑或者运算的 DNS 查询功能性和授权。

[0126] 设备可以包括用于保存、跟踪和管理客户机和服务器之间的 cookie 的 cookie 代理或者 cookie 管理器。Cookie 可以包括用于增加或者插入 cookie 以及移除 cookie 的 cookie 存储装置，称之为 cookie 存储器 (jar)。Cookie 管理器或者代理可以包括在 cookie 存储器中通过请求和 / 或响应的 URL、域名或者其他信息来保存和查询 cookie 信息的功能、逻辑或者运算。在一些实施例中，设备 200 代表不支持 cookie、停用的 cookie 的客户机或者对于期望或优选不发送 cookie 给客户机的情况中管理 cookie。

[0127] 设备还可以包括 AppFW 280，其在 Citrix System 公司制备的设备的情况下称为 AppSecure。AppSecure 280 模块可以包括用于执行任一类型和形式的内容重写，诸如 URL 重写的逻辑、功能或运算。在一些实施例中，AppSecure 280 模块执行到客户机和服务器之间的请求和 / 或响应的任一类型和形式的内容注入。在一些实施例中，AppSecure 模块 280 将脚本插入到对客户机的响应中，诸如 Java 脚本，来执行任一类型和形式的期望的功能性。

[0128] 用于免客户机 SSL VPN 访问的设备 200 的任一部件可以响应于配置或者通过配置驱动，诸如经由策略引擎的任意一个或者多个策略。策略可以指导和确定通过 VPN 模块执行的 URL 编码和译码的类型和形式。在一些实施例中，策略可以指导和确定 cookie 代理如何并且何时管理和代理 cookie。在其他实施例中，策略可以指导并且确定 AppSecure 模块如何并且何时执行 URL 重写和 / 或内容注入。策略可以指导用户访问专用网络和专用网络上的应用的方式。策略可根据访问方案配置，该访问方案可以包括基于用户、客户机的类型和形式、网络的类型和形式、访问资源的类型、所使用应用的类型、暂时信息以及可以通过设备经由传输到的网络业务量来确定的任一信息的任一组合的访问。

[0129] 参考图 3B，讨论经由设备 200 用于免客户机 SSL VPN 访问的包流。响应于成功的登入请求，VPN 设备可以发送入口页面给登入请求的发送者。入口页面可以具有结合图 3A 描述的“vpn 编码格式”的一个或者多个链接。入口页面流经以下描述的响应码路径。当用户点击入口页面中的任一 URL 时，包流可以以多种方式并且使用多个步骤来执行。在一些实施例中，对于步骤 Q1 的请求路径，设备 200 可以接收 URL 请求并且查询 AAA 模块中的 VPN 会话。在步骤 Q2，设备可以将 VPN 编码的 URL 译码为期望的 URL 用于服务器或者网络 104’。设备还可以将请求的首部（诸如首部值）修改为服务器格式或者意于通过服务器 106 传输和使用的格式，诸如 HTTP 服务器。设备可以重新解析首部，使得设备的任一其它模块以服务器格式查看该请求。在步骤 Q3，在请求路径中，设备经由 cookie 管理器或者代理可以基于 URL 的域和路径来查看用于该请求的 cookie。在一些情况中，如果该请求包括 cookie，则该设备可以从 cookie 存储器插入 cookie。在步骤 Q4，设备可以经由设备的 DNS 查询功能 / 模块来将以 URL 中的服务器的域名解析为服务器的 IP 地址。设备可以基于 AAA 模块中的 DNS 查询来建立服务器信息。此外，可以评估授权策略来确定该请求是否可以传送给服务器。在步骤 Q5，设备可以将请求发送给服务器。在一些实施例中，只有在授权成功的情况下，设备才将该请求发送给服务器。

[0130] 在从服务器经由设备到客户机的响应路径中，在步骤 S1，设备可以接收来自服务

器的响应。VPN 模块 280 可以处理该响应。VPN 模块可以将该响应首部传递到 cookie 代理模块并且将响应的体传递到 AppSecure 模块。在步骤 S2，cookie 代理可以从响应的首部移除未被配置或者以其它方式识别为客户机所消耗 cookie 的 cookie 并且将它们保存在当前会话所使用的 cookie 存储器中。在步骤 S3，AppSecure 模块可以根据重写策略来重写“vpn 编码形式”的任一 URL。AppSecure 模块还可以将任一脚本插入响应体中，诸如要在客户机侧执行的 Java 脚本代码。在步骤 S4，设备可以发送修改后的响应给客户机。在许多实施例中，任一个 Q 或者 S 步骤以任一顺序或者以与此处所描述的任一其它步骤或者实施例的任一组合进行。

[0131] D、用于细粒度策略驱动的 cookie 代理的系统和方法

[0132] Cookie 可以用来维持网络上两个实体之间的前一事件、事务或者通信的系统或者存储器的状态。在一些情况中，cookie 可以用于多种类型的会话跟踪。Cookie 管理可以认为是网络装置的特征，诸如中间设备 200 的特征，并且可以提供一种在建立、利用或者控制在服务器 106 和客户机 102 之间传输的 cookie 方面管理服务器 106 或者客户机 102 的方法。此处描述的中间设备的实施例提供管理客户机和服务器的 cookie 的方法。在一些实施例中，中间设备在服务器侧管理服务器消耗的 cookie 而不发送 cookie 给客户机浏览器。

[0133] 图 4A、4B 和图 4C 示出用于 cookie 代理的系统和方法。图 4A 描述用于在免客户机 SSL VPN 环境的实施例中管理 cookie 的系统和方法的实施例。图 4B 示出用于包括 SSL VPN 免客户机访问方案的 cookie 管理的方法的实施例的时序图和步骤。图 4C 描写中间设备使用诸如唯一的客户机 ID 的唯一标识符来进行 cookie 代理的实施例。这些示出的实例可以涉及用于执行 cookie 管理的系统和方法的多个实施例，诸如免客户机 cookie 管理。

[0134] 免客户机 cookie 管理可以使得驻留在不安全网络中的 web 客户机访问寄载在安全网络之后的 web 应用，而不会损害安全网络的安全性。例如，免客户机 cookie 管理可以通过移除服务器消耗的 cookie 数据来改进所传输信息的安全性。免客户机 cookie 管理可以使得服务器消耗的 cookie 数据不传输到客户机并且禁止对可包括在 cookie 中的任一敏感信息的访问。此外，免客户机 cookie 管理可以使得不支持 cookie 的诸如 PDA 和 WAP 浏览器的 web 浏览器可以和要求 cookie 的服务器上的 web 应用一起运行。进一步，在所使用的 web 应用和 cookie 路径重写不兼容的例子中，免客户机 cookie 管理通过重写 cookie 路径可以提供允许这样的应用工作的服务。

[0135] 现在参考图 4A，示出经由中间设备 200 来执行 SSL VPN 免客户机 cookie 管理的系统和方法的实施例。图 4A 描述经由中间设备 200 和服务器 106 通信的客户机 102。中间设备 200 包括策略引擎 236 和 cookie 管理器 420。客户机 102 发送请求给中间设备 200，诸如意图发往服务器 106 的 HTTP（超文本传输协议）请求。该请求包括可以识别保存在服务器 106 上或者以其他方式从服务器 106 可用的服务或者资源的 URL（统一资源定位符）。Cookie 管理器 420 接收该请求并且使用策略引擎 236 产生和客户机 102 相关联的 cookie。所产生的 cookie 可以满足服务器 106 的任一优先设置或者配置，因此使得初始的无 cookie 的请求可被服务器 106 接受并且处理。Cookie 管理器 420 修改该请求来包括 cookie 并且将该修改后的请求转发给服务器 106。服务器 106 发出针对该请求的响应，并且将该响应使用通过 cookie 管理器 420 产生的 cookie 传送给中间设备 200。Cookie 管理器 420 通过移除该 cookie 来修改该响应。修改后的响应随后传输给客户机 102。服务器 106 和客户

机 102 还可以使用 cookie 来发送附加的请求和响应，因此允许无 cookie 的客户机 102 来存取 cookie 配置的服务器 106 上的资源。

[0136] Cookie 管理器 420 可以是产生、终止、修改或者管理 cookie 的任一装置、部件、单元、函数或者设备。Cookie 管理器 420 还可以修改客户机 102 的请求和服务器 106 的响应。Cookie 管理器 420 可以包括管理和控制 cookie 的硬件、软件或者硬件和软件部件的任一组合。Cookie 管理器 420 可以包括用来控制、管理或者修改客户机 102 和服务器 106 之间的传输信息的逻辑、控制函数、处理电路、软件程序、算法和脚本。在多个实施例中，cookie 管理器 420 包括在管理 cookie 和提供对客户机 102 和服务器 106 之间的通信的控制的过程期间所使用的策略。

[0137] 在一些实施例中，Cookie 管理器 420 可以唯一地识别网络上的用户，例如通过使用配置的策略来提供唯一的标识符，配置的策略诸如策略引擎 236 的引擎。唯一的标识符可以是唯一标识网络上客户机 102、服务器 106 或者设备 200 的任一数据、值或者数据集、数字集或者字符集。在一些实施例中，Cookie 管理器 420 可利用唯一标识符来将来自服务器 106 的接收的响应关联到响应发往的特定的客户机 102。Cookie 管理器 420 可以包括任一功能性来使用 cookie 和客户机标识符修改响应，使得客户机 102 以期望格式接收响应。类似地，Cookie 管理器 420 可以包括任一功能性来使用客户机唯一标识符和为客户机所产生的 cookie 修改特定客户机 102 的请求，以包括特定的 cookie 来访问服务器 106 上的资源。

[0138] Cookie 管理器 420 可以修改或者改变通过服务器 106 或客户机 102 发送的任一传输。在一些实施例中，Cookie 管理器 420 使用策略引擎 236 中的一个或者多个用来处理 cookie 的策略来修改客户机 102 和服务器 106 之间的传输。Cookie 管理器 420 可以修改传输以包括或者排除 cookie 和唯一的客户机标识符。在一些实施例中，Cookie 管理器 420 为请求访问任一服务器 106 的任意客户机 102 产生 cookie。Cookie 管理器 420 可以响应于策略为客户机 102 产生 cookie。在一些实施例中，Cookie 管理器可以响应于策略确定不应该允许特定的客户机有 cookie。在其他实施例中，Cookie 管理器 420 响应于策略确定特定的唯一客户机标识符应该和客户机 102 相关联。在进一步的实施例中，Cookie 管理器 420 响应于策略来确定为请求产生的 cookie 的类型和形式。在又一个实施例中，Cookie 管理器响应于策略来确定是否或者如何保存 cookie 以用于客户机的未来请求。

[0139] 在一些实施例中，Cookie 管理器从请求或者响应消除或者删除 cookie。Cookie 管理器可以分配或者重新分配 cookie 给客户机 102 或者服务器 106。仍在另一个实施例中，Cookie 管理器 420 改变、修改或者重写来自客户机 102 的请求或者来自服务器 106 的响应的 cookie。Cookie 管理器 420 可将唯一识别客户机的值、名称或者唯一客户机标识符匹配到和客户机、服务器或者中间设备 200 相关的 cookie 或者 cookie 的一部分。在一些实施例中，Cookie 管理器 420 可以将和唯一客户机标识符相关的 cookie 增加到 URL。在进一步的实施例中，Cookie 管理器 420 可以移除唯一的客户机标识符并且代以增加和唯一客户机标识符相关的 cookie。在多个实施例中，Cookie 管理器 420 可以使用唯一标识符代替 cookie，有时也称为唯一识别客户机 102 或者服务器 106 的唯一的 ID。

[0140] Cookie 管理器 420 可以使用解析器用来解析传输。Cookie 管理器 420 还可以使用内部映射表用于将涉及多个客户机、服务器或者设备 200 的多个唯一标识符匹配到涉及客户机、服务器或者设备的多个 cookie。例如，Cookie 管理器 420 可以使用包括和一个或

者多个 cookie 相关的唯一标识符的映射图，用于将通过唯一标识符唯一标识的客户机 102 和一个或者多个 cookie 匹配。在这样的情况中，Cookie 管理器 420 可以使用匹配到唯一客户机标识符的一个或者多个 cookie 来修改、改变或者编辑来自客户机的请求或者到客户机的响应。在需要遵从接收装置即接收传输的客户机 102 或者服务器 106 的配置或者优先设置时，Cookie 管理器 420 可以利用策略通过管理、增加或者移除来自或者去往客户机 102 和服务器 106 的传输的 cookie 和唯一客户机标识符来管理客户机 102 和服务器 106 之间的传输。

[0141] 分号定界的客户机列表 (semi-colon client delimited list) 可以是包括和与一个或者多个服务器 106 经由中间设备 200 通信的客户机 102 相关的信息的 cookie 列表的任一映射表、列表、数据库或者文件。此处分号定界的客户机列表还可以互换地称为内部映射表或者 cookie 列表或者映射表。在一些实施例中，分号定界的客户机列表包括可用于代替用于上行通信即发往服务器 106 的通信或者下行通信即发往客户机 102 的通信的 cookie 的 cookie 和值或者唯一标识符的名称值对。有时，Cookie 管理器 420 可以使用客户机消耗的 cookie 和 / 或客户机和服务器消耗的 cookie 的分号定界的列表来将一个或多个客户机 102 的值或者唯一标识符链接、匹配或者关联到和客户机相关的每一个 cookie。通过使用分号定界的列表，Cookie 管理器 420 可以确定哪一个 cookie 重新注入、增加或者包括到下行或者上行传输中。Cookie 管理器 420 可以使用 cookie 的映射表或者列表来将来自通过中间设备 200 接收的传输的 cookie 匹配到传输所要发送的客户机 102。Cookie 管理器 420 随后可以编辑或者修改传输来排除 cookie 并且代以包括和客户机 102 相关的任一其他信息。类似地，Cookie 管理器 420 可以使用 cookie 的映射表或者列表来将唯一客户机标识符匹配到要包括到传输中的 cookie。

[0142] 在一些实例中，通过开始服务器侧 cookie 管理并且不指定分号定界的列表，web 应用会可能不正确地运行。使用该分号定界的列表可以消除在 cookie 管理器 420 滤除服务器 106 处的客户机消耗的 cookie 时产生的错误。使用该分号定界的列表还可以消除在 web 应用尝试访问导致不期望行为的客户机 102 上的 cookie 值时产生的错误。在多个实施例中，定界的客户机 cookie 列表，或者列表可以包括用于服务器侧或者客户机侧 cookie 管理的多种开启 / 关断设置。在一些实施例中，定界的客户机 cookie 列表可以保存在任意个数据层中或者多个表中，该表包括用于该方法或者过程的每一步骤的任一变量的设置和配置。

[0143] Cookie 可以通过类型或者特征来分类。cookie 管理器 420 可以通过和 cookie 关联的唯一标识符来将 cookie 分类或者分选。服务器消耗的 cookie 可以是通过诸如服务器 106 的资源设置的 cookie，在响应上发布 Set-Cookie。服务器消耗的 cookie 值可以通过任一客户机侧代码检查或者可以不检查。在某些实施例中，cookie 管理器 420 可以将服务器消耗的 cookie 识别或者分类为服务器消耗的 cookie。在多个实施例中，cookie 管理器 420 可以从请求或者响应的下行传输中移除服务器消耗的 cookie 并且将它们重新注入和该请求或者响应相关的上行传输中。在其他实施例中，服务器消耗的 cookie 可以和有关会话的秘密或者敏感数据相关联。在一些实施例中，服务器消耗的 cookie 可以通过 cookie 管理器 420 来管理并且不可以发送到 web 浏览器。在某些实施例中，服务器消耗的 cookie 可以从发送到 web 浏览器的消息剥离并且可以保存在 cookie 管理器 420 或者和 cookie 管理器

420 相关的任一存储装置。

[0144] 客户机消耗的 Cookie 可以是通过客户机 102 的 web 浏览器经由诸如 Java 脚本的脚本设置在上行传输上的 Cookie。在一些实施例中，客户机消耗的 Cookie 是设置在从源起始的下行传输上的 cookie，其中该源可以是客户机 102、服务器 106 或设备 200。在多个实施例中，客户机消耗的 Cookie 可以通过客户机 102 或者中间设备 200 来检查或者修改。在一些实施例中，客户机消耗的 Cookie 不通过服务器 106 检查或者修改。服务器 106 可以仅按原样接收请求，并且依赖于中间设备 200 来对该请求提供修改。类似地，客户机 102 也不能修改响应，而是可以依赖于中间设备 200 来修改该响应。在一些实施例中，客户机消耗的 Cookie 通过服务器 106 来检查、编辑或者修改。在进一步的实施例中，客户机消耗的 Cookie 不通过 cookie 管理器 420 来管理，而是下行发送到客户机 102 的 web 浏览器。有时，客户机消耗的 Cookie 和服务器消耗的 cookie 通过客户机 102 和服务器 106 二者来读出、修改和产生。在一些实施例中，cookie 管理器 420 不能管理客户机或服务器消耗的 cookie，并且客户机和服务器消耗的 cookie 可以下行发送到客户机 102 的 web 浏览器。在一些实施例中，客户机 102 可以执行服务器 106 的功能性，并且相反亦然。在其他实施例中，客户机 102 可以互换使用来替代服务器 106，服务器 106 也可以互换使用来替代客户机 102。在一些实施例中，客户机或者服务器消耗的 cookie 可以被使用、修改、读、写或者传输来自或者去往客户机 102、服务器 106 或者中间设备 200 的任一个。在许多实施例中，服务器消耗的 cookie 包括客户机消耗的 cookie 的所有功能性，并且以相同方式和通过如任一客户机消耗的 cookie 一样的相同部件来处理、修改、控制或者以其他方式使用。有时，所有的 cookie 可以是服务器消耗的 cookie。服务器消耗的 cookie 可以通过服务器来使用、读出或者编辑。在进一步的实施例中，一些服务器消耗的 cookie 还可以是客户机消耗的 cookie。客户机消耗的 cookie 可以通过客户机使用、读出或者编辑以及通过服务器来使用、读出或者编辑。在一些实施例中，服务器和客户机消耗的 cookie 通过中间设备 200 来使用、编辑、读出、写入或者修改。有时在一些实施例中，一些 cookie 通过客户机而不通过服务器来使用。

[0145] 上行或者下行通信可以用来指示通信的方向。例如，有时上行请求通信或者上行方向可以和从客户机 102 到服务器 106 的通信或者事务相关。在一些实施例中，上行请求通信或者上行方向可以和从服务器 106 到客户机 102 的通信或者事务相关。在多个实施例中，下行请求通信或者下行方向在一些实施例中可以和从客户机 102 到服务器 106 的通信、事务或者方向相关。在多数实施例中，下行请求通信或者下行方向可以和从服务器 106 到客户机 102 的通信或者事务相关。有时，朝向服务器的事务或者通信可以称为上行事务或者上行通信并且朝向客户机的事务或者通信可以称为下行事务或者下行通信。

[0146] 进一步参考图 4A，示出用于执行免客户机 cookie 管理的实施例的方法的步骤。在步骤 1，客户机 102 通过中间设备 200 发送请求给服务器 106。该请求包括 URL 请求，诸如中间设备 200 提供的 URL，例如 http://abc.com/dir/index.asp。在步骤 2，中间设备 200 和 cookie 管理器 420 通信，该 cookie 管理器 420 比较 URL 提供的域名和路径的 cookie 内部映射图和输入请求 URL。如果 cookie 管理器检测到输入请求 URL 和内部映射图中可用的 cookie 之间的任一匹配，则从 cookie 管理器返回名称值对的阵列给中间设备。在步骤 3，中间设备将修改的请求转发给服务器 106。在一些实施例中，请求还可以称之为 HTTP 请求，并且修改后的请求可以称之为修改后的 HTTP 请求。该请求可以被修改为包括来自 cookie

管理器 420 的一个或者多个 cookie。如图 4A 中所示，在步骤 3 中发送的消息包括分配给 cookie 的等于 25 的值。在步骤 4，服务器以包括首部和体的响应来响应该请求，诸如 HTTP 首部和内容体。该响应可以包括通过 Set-cookie HTTP 首部限定的多个 cookie 定义。进一步的例子，来自服务器的响应包括设为 25 的 Set-cookie 值。中间设备可以在步骤 5 访问 cookie 管理器 420，而将来自响应的 cookie 值传递到 cookie 管理器。Cookie 管理器 420 检查所接收值是否是新的或者对于给定 URL 是否更新，并且在输入映射图上执行任一必要的更新。Cookie 管理器 420 还检查客户机消耗的 cookie 是否应该返回给中间设备，用于 web 浏览器消耗。在步骤 6，中间设备将来自服务器的响应发送给客户机 102 的 web 浏览器，服务器消耗的 cookie 从该响应的首部移除，增加客户机消耗的 cookie 到首部。

[0147] 在进一步的细节中，图 4A 描述客户机 102 通过发送第一请求给中间设备 200 来初始化具有发往服务器 106 的第一请求的通信。图 4A 中的箭头 1 标示发往中间设备 200 的第一请求 http://abc.com/dir/index.asp 的传输。在一些实施例中，第一请求通过服务器 106、中间设备 200 或者网络 104 上的任一其它装置来发送。第一请求可以包括或者可以不包括 cookie。在一些实施例中，第一请求包括 URL 或者 HTTP 请求。在进一步的实施例中，第一请求包括对保存在服务器 106 上的资源的路径。在进一步的实施例中，第一请求包括唯一识别网络上通信的所有装置之中的客户机 102 的唯一标识符。在又一个实施例中，第一请求包括唯一识别客户机 102 上的会话的唯一标识符。在进一步实施例中，第一请求包括唯一识别客户机 102 上用户的唯一标识符。在一些实施例中，传送第一请求的客户机 102 未配置 cookie 并且不传送 cookie。在其他实施例中，传送第一请求的客户机使用并不安全或不期望传输敏感信息的网络或者连接。

[0148] 如图 4A 的箭头 2 所示，中间设备 200 接收第一请求并且将其转发给 cookie 管理器 420。在一些实施例中，中间设备 200 响应于所接收的第一请求来初始化或激活 cookie 管理器 420。在其他实施例中，中间设备 200 响应于识别到客户机 102 满足用于激活 cookie 管理器 420 的一组预设条件来初始化或者激活 cookie 管理器。该组预设条件可以包括和客户机 102、客户机 102 上的会话或者客户机 102 上的用户相关的任一判定。该判定可以通过中间设备 200、客户机 102、服务器 106 或者网络 104 上的任一其它部件或者装置来做出。在一些实施例中，判定通过策略引擎 236 来做出。在一些实施例中，该组预设条件包括该请求来自不支持 cookie 的客户机 102 的判定。在进一步的实施例中，该组预设条件包括客户机 102 使用不安全的会话或者连接的判定。在又一个实施例中，该组预设条件包括客户机使用不安全网络的判定。在进一步的实施例中，该组预设条件包括客户机 102 可以访问通过第一请求来请求的服务器 106 上的所请求资源或者服务的判定。

[0149] 中间设备 200 可以激活或者初始化 cookie 管理器 420 用来根据 cookie 的内部映射表来检查或者匹配所接收的请求 URL。Cookie 的内部映射图也称为 cookie 的映射图、列表或者分号定界的列表。在一些实施例中，策略引擎 236 的策略根据 cookie 的内部映射图匹配第一请求的一部分。该映射图包括任一数量的 cookie，每一个 cookie 可以和任一数量的客户机 102、服务器 106 或者设备 200 相关联、连接或者配对。在一些实施例中，cookie 管理器 420 检查或者匹配所接收的请求 URL 到保存在映射图中的消息的路径或者域。在一些实施例中，策略引擎 236 的策略将涉及客户机 102 的信息匹配到保存在映射图中的消息的路径或者域。消息的路径或者域可以将第一请求的 URL 或者一部分或者任一部分匹配到

用于客户机 102 的一个或者多个 cookie。在一些实施例中，cookie 管理器 420 或者策略检测或者确定第一请求的一部分和与客户机 102 或者服务器 106 相关的一个或者多个 cookie 或者唯一标识符之间的匹配。在一些实施例中，cookie 管理器 420 或者策略引擎 236 的策略检测或确定第一请求的一部分和一个或者多个名称值对或者值名称对之间的匹配。

[0150] Cookie 管理器 420 可以响应于在所接收请求的一部分和 cookie 或者来自 cookie 的内部映射图的一个或者多个 cookie 之间做出的匹配来产生、提供或返回一个或者多个名称值对。可以通过策略引擎 236 的策略做出所接收的请求的一部分和 cookie 或者来自 cookie 的内部映射图的一个或者多个 cookie 之间的匹配。在一些实例中，cookie 管理器 420 响应于来自客户机 102 的请求或者服务器 106 对于该请求的响应匹配来自映射图的任一 cookie 或者唯一标识符的确定来返回一个或者多个名称值对。在一些实施例中，cookie 管理器 420 可以将 cookie 的一部份匹配到来自所接收请求或者对所接收请求的响应的 URL 的一部分。Cookie 管理器 420 可以将匹配的 cookie 分配给第一请求。在一些实施例中，cookie 管理器 420 可以修改第一请求来包括所匹配的 cookie。在一些实施例中，cookie 管理器 420 在未作出匹配时产生用于客户机 102 或者服务器 106 的 cookie。在一些实施例中，cookie 管理器为第一请求的客户机 102 产生 cookie 并且将该 cookie 分配给客户机 102。所产生的 cookie 可以包括 cookie 的值。Cookie 的值可以是将 cookie 唯一关联到 cookie 映射表中的客户机 102 的唯一值。Cookie 管理器可以使用 cookie 值来将输入响应和从服务器 106 到客户机 102 的第一请求关联。Cookie 管理器可以给客户机 102 分配 cookie 以用于从客户机 102 到服务器 106 的第一请求或者任一其它进一步请求。Cookie 管理器 420 或者中间设备 200 的任一其它部分可以重写、修改、格式化或者改变诸如第一请求的所接收的请求，来包括匹配或者产生的 cookie 或者满足通过服务器 106 接收的请求的任一格式或者内容需求。

[0151] 继续参考图 4A，箭头 3 表示设备 200 把通过 cookie 管理器 420 处理的第一请求发送到服务器 106 的步骤。所传送的第一请求可以被修改。在一些实施例中，该请求通过 cookie 管理器 420 或者设备 200 来修改。通过中间设备 200 传送的请求可以包括来自 cookie 管理器 420 的一个或者多个 cookie。在一些实施例中，所修改的请求包括 cookie 管理器 420 可用来关联该请求到客户机 102 的 cookie 的值。已经通过设备 200 或者 cookie 管理器 420 修改或者改变的请求可以称之为修改的请求。服务器 106 可以以用于通过服务器 106 处理的请求的优先设置或者配置相符的格式来接收修改的请求。服务器 106 可以接收修改的请求并且确定所接收的修改的请求是合法请求。

[0152] 箭头 4 示出服务器 106 传送或者发出对于修改的请求的响应的步骤。所发布的响应可以包括客户机 102 已经请求的任一信息、服务或者资源。在一些实施例中，发送的响应包括 web 页面。在其他实施例中，所发送的响应包括文件。在进一步的实施例中，所发送的响应包括应用或者计算机软件程序。在进一步的实施例中，所发送的响应包括用于建立与客户机 102 的会话的消息或者验证或者授权消息。来自服务器 106 的响应可以包括用来唯一识别响应所发往的客户机 102 的 cookie 的值。来自服务器的响应可以包括首部和内容体，其每一个可以包括以下任一个 :一个或者多个 cookie、一个或者多个 cookie 限定、一个或者多个 cookie 的组件或者部分、和涉及 cookie 或者和 cookie 关联的信息或者值。在一些实施例中，cookie 限定通过“Set-cookie”或者“Set-cookie2”HTTP 首部设置。

“Set-cookie”或者“Set-cookie2”HTTP 首部在此处可称为 Set-cookie。

[0153] 箭头 5 示出中间设备 200 和 cookie 管理器 420 通信并且修改对于第一请求的响应的步骤。在许多实施例中，中间设备 200 将来自服务器 106 的响应的一个或者多个 cookie 值或者唯一客户机标识符传送给 cookie 客户机 420。在许多实施例中，中间设备 200 将来自服务器的响应的一个或者多个 cookie 传送给 cookie 客户机 420。中间设备 200 可以激活或者初始化 cookie 管理器 420 用来根据 cookie 的映射表或者列表来检查或者匹配来自响应的 URL。中间设备 200 可以修改或者编辑服务器的响应来包括和响应的一部分匹配的 cookie。在一些实施例中，cookie 管理器 420 可以检查或者匹配 URL、响应的首部或者任一其它部分到保存在映射表中的消息的路径、域、cookie 或者 cookie 的一部分。在这样的实例中，如果响应的一部分的任一 cookie 或者唯一标识符和与客户机 102 相关联的保存信息相匹配，则中间设备 200 可以修改或者编辑该响应来如期望地在响应中包括客户机 102 的 cookie 或者和客户机 102 相关联的任一其它信息。在 cookie 管理器 420 检测到来自请求或者响应的 URL 之间的匹配的情况中，来自 cookie 列表或者内部映射表的一个或者多个 cookie、一个或者多个名称值对、或者值名称对可以返回到中间设备 200。在一些实施例中，策略引擎 236 的策略将响应的一部分和一个或者多个 cookie 或者唯一标识符相匹配。有时，cookie 管理器 420 响应于响应的一部分和 cookie 或者唯一标识符之间的匹配来返回名称值对的阵列。在一些实施例中，cookie 管理器 420 可以将 cookie 的一部分匹配到来自所接收请求的 URL 的一部分。在多个实施例中，cookie 管理器 420 将 cookie 的一部分匹配到所接收的请求的任一部分，诸如请求的 URL、体、或者首部。在一些实施例中，如果 cookie 管理器 420 未将对该请求的所接收响应的一部分匹配到任一 cookie 或者任一唯一标识符，cookie 管理器 420 产生新的 cookie 或者新的唯一标识符或二者并且将其一个 / 二者分配给客户机 102 或者服务器 106。Cookie 管理器 420 可以使用新产生的 cookie 来修改服务器 106 对第一请求的响应。用于客户机或者服务器的这样的新的 cookie 和唯一标识符可以用于相同客户机或者服务器 106 的任意未来的请求或者响应。在一些实施例中，cookie 管理器 420 确定有客户机 102 的请求或者服务器 106 的响应的新的或者更新的 cookie 值，并且进一步相应更新映射图或者数据库。此外，cookie 管理器 420 还可以确定客户机 102 所消耗的 cookie 是否应该返回到中间设备 200，用于 web 浏览器消耗。

[0154] 仍旧参考图 4A，箭头 6 示出中间设备 200 经由下行响应来发送或者转发修改的响应给客户机 102 的步骤。在一些实例中，下行响应可以称为从服务器 106 经由中间设备 200 朝向客户机 102 的传输。类似地，上行通信可以是从客户机 102 经由中间设备 200 朝向服务器 106 的任一通信。在一些实施例中，修改后的响应不包括任一 cookie。在进一步的实施例中，修改后的响应包括通过 cookie 管理器 420 用来将和 cookie 相关的一个或者多个客户机 102 关联到客户机 102 的唯一客户机标识符。在进一步的实施例中，修改的响应包括客户机 102 接受或者所期望的任一格式。从中间设备 200 到客户机 102 的修改的响应可以包括从响应移除的服务器消耗的 cookie。在一些实施例中，所转发的修改的响应可以包括重新增加到响应的首部或者任一其它部分的客户机消耗的 cookie。在许多实施例中，从中间设备 200 到客户机 102 的修改的响应包括来自服务器的响应，该响应被修改以排除 cookie 管理器 420 中和客户机 102 相关的 cookie。

[0155] Cookie 管理，诸如通过 cookie 管理器 420 的 cookie 管理，可以导致或者提供通过

客户机 102 使用的并且通过服务器 106 提供的服务、资源或者应用，来在相同的 cookie 域名空间中运行或者提供。中间设备 200 可以提供客户机侧 cookie 管理或者服务器侧 cookie 管理来增加在客户机 102 和服务器 106 之间传送的 cookie 的安全性。Cookie 管理，诸如客户机侧 cookie 管理，可以消除 HTTP 协议限制的约束，诸如允许来自每单个客户机的单个源的 cookie 的最大数量。例如，在仅允许客户机每个会话 20 个 cookie 的系统中，通过中间设备 200 的 cookie 管理器 420 的 cookie 管理可以使得客户机能够通过重新使用和客户机相关并且保存在 cookie 管理器 420 中的 cookie 来消除这样的限制。使用客户机侧 cookie 管理，在这样的例子中，可以使得客户机 102 即使在每会话 20 个 cookie 限制会影响提供给客户机的服务时能够继续和服务器 106 相通信。在此情况中，拦截并且转发客户机和服务器之间的通信的中间设备 200 可以管理、修改、重写或者编辑请求或者响应的部分并且使用和客户机或者服务器相关的 cookie，因此即使在超过 20 个的 cookie 被传输之后也可以进行通信。

[0156] Cookie 管理器 420 或者中间设备 200 执行的免客户机 cookie 管理可以包括重写发送给客户机 102 或者服务器 106 的下行 cookie 首部上的 cookie 路径。在一些实施例中，cookie 管理器 420 或者中间设备 200 将来自客户机 102 或者服务器 106 的响应或者请求转发给所期望的目标，而没有对其改变或者修改。在其他实施例中，通过 cookie 管理器 420 执行的免客户机 cookie 管理可以包括经由中间设备 200 的状态管理。Cookie 管理器 420 可以由中间设备激活，用来检查上行请求以发现应该注入到发往服务器 106 的流或者通信中的服务器 cookie。来自下行通信的响应可以保持在 cookie 管理器中，该响应意图被修改使得它们没有 cookie。

[0157] 现在参考图 4B，描述用于免客户机 cookie 管理的方法的步骤的实施例的序列图。总的来说，图 4B 示出经由设备 200 和 cookie 管理器 420 和服务器 106 的 web 应用通信的客户机 102 的浏览器。客户机 102 发送 HTTP 请求到设备 200，设备 200 在此处还被称为中间设备 200。中间设备 200 处理该请求并且使用 cookie 管理器 420 来检查 cookie。Cookie 管理器 420 将和该请求相关的 cookie 返回给中间设备 200。中间设备 200 修改 HTTP 请求的首部来包括这些 cookie 并且将修改的 HTTP 请求发送到服务器 106。服务器 106 返回对 HTTP 请求的 HTTP 响应。中间设备 200 发送来自 HTTP 响应的任一 cookie 给 cookie 管理器 420，以增加到用来将客户机 102 的所有 cookie 关联到客户机 102 的映射图。中间设备 200 修改该响应以将 cookie 从 HTTP 响应移除并且使用 cookie 管理器 420 来进一步将任一客户机 102 相关的 cookie 增加到 HTTP 响应。中间设备 200 将修改的 HTTP 响应传送到客户机 102。

[0158] 进一步总的来说，图 4B 示出客户机 102 发送请求给中间设备 200，此处也称为代理。该请求可以是访问由服务器 106 提供的任一资源或者任一服务的任一请求。在一些实施例中，该请求是访问 web 页面或者 web 站点相关的服务的 HTTP 请求。在其他实施例中，该请求是建立和服务器 106 的连接的请求。在进一步的实施例中，该请求是建立和服务器 106 的会话的请求。在进一步的实施例中，该请求是使用服务器 106 提供的应用的请求。在又一个实施例中，该请求是访问流文件的请求，诸如音频或者视频文件。在又一个实施例中，该访问是对安全文档的访问。该请求可以包括多个请求。

[0159] 中间设备 200 可以处理所接收的请求并且使用 cookie 管理器 420 来检查 cookie。

中间设备可以处理该请求并且建立客户机 102 所包括的 cookie（如果有的话）。处理该请求还可以包括建立唯一客户机标识符来识别和客户机 102 的任一其它通信。中间设备 200 可以处理该请求并且将其转发给 cookie 管理器 420。在一些实施例中，中间设备 200 将请求的一部分转发给 cookie 管理器 420。在一些实施例中，cookie 管理器 420 使用一个或者多个映射图来将诸如客户机的唯一标识符的客户机 102 关联信息关联或联系到用于由客户机 102 至服务器 106 的通信的 cookie。类似地，cookie 管理器 420 的映射图可以用来将任一客户机 102 侧 cookie 关联到服务器 106 侧 cookie，客户机 102 使用该客户机侧 cookie 用于和服务器 106 的通信。在一些实施例中，来自请求的唯一标识符被匹配到映射表中的名称值对或者 cookie。Cookie 可以经由 cookie 管理器 420 的映射图和客户机 102 相关联。如果用于客户机 102 的新的 cookie 在 cookie 管理器 420 的映射图中不是已经存在，则 Cookie 管理器 420 可以建立这样的 cookie。

[0160] 在一些实施例中，当中间设备 200 处理每个请求或者响应时，中间设备 200 检查所处理的传输中的 URL 是否是支持服务器侧 cookie 管理。中间设备 200 可以调用诸如 ProcessRequest() 的 cookie 管理器相关的函数来进行请求处理。Cookie 管理器 420 还可以检查内部映射图，其可以是 cookie 管理器映射图或者 cookie 管理器列表，用来帮助确定是否存在需要增加到上行请求的任一输入 cookie。Cookie 管理器 420 还可以在映射图内部记录在会话中第一次遇到的所有 cookie 用于未来的请求。Cookie 管理器还可以确保该系统不发送特定的 cookie，诸如设备 200 相关的 cookie，到寄载的 web 应用，诸如 NSC_AAAC。在一些实施例中，诸如 ProcessRequest() 函数的函数可以内部调用另一个函数，诸如 FilterCookies()。在一些实施例中，FilterCookies() 可以调用 ProcessRequest()。在一些实施例中，两组值可以称之为 Citrix.Fei.ClientCookies 和 Citrix.Fei.ServerCookies。cookie 管理器 420 可以使用 Citrix.Fei.ClientCookies 和 Citrix.Fei.ServerCookies 来建立分号定界的列表或者内部 cookie 管理器映射图，此处称为 cookie 管理器列表或者映射图。该映射图可用在 cookie 管理的决策制定过程期间使用。Cookie 管理器 420 可以将 cookie 管理器 420 中的 cookie 管理器映射图项目串行组织为两个会话值，Citrix.Fei.ClientCookies 和 Citrix.Fei.ServerCookies。cookie 管理器 420 还可以使用该两个值来继续该会话或者将该会话关联到客户机 102 或者服务器 106，用于任一未来的传输信息。如果 cookie 管理器 420 将客户机 102 的该请求匹配到映射图中的 cookie 或者多个 cookie，则 cookie 管理器 420 可以将该 cookie 或者该多个 cookie 传送到设备 200。

[0161] 中间设备 200 可以修改 HTTP 请求的首部并且将修改的 HTTP 请求传送给服务器 106。在一些实施例中，cookie 管理器 420 修改该请求来包括和客户机 102 相关联的 cookie 或者多个 cookie。在一些实施例中，cookei 管理器 420 修改请求的一部分，诸如 URL 或者首部，来包括识别 cookie 的值或者一组值或者字符。在一些实施例中，cookie 管理器 420 修改该请求来包括或者增加和客户机 102 相关联的 cookie 的一部分。在其他实施例中，cookie 管理器 420 修改该请求来包括或者增加和客户机 102 相关联的唯一标识符。在进一步的实施例中，cookie 管理器 420 修改该请求来从客户机的该请求来排除或者移除 cookie 或者唯一标识符。在一些实施例中，中间设备 200 的任一部件修改该请求来包括 cookie 管理器 420 提供的 cookie 或者多个 cookie。

[0162] 中间设备 200 可以执行针对该请求的任一数量的修改。在一些实施例中，中间设

备 200 使用一个或者多个 cookie 的一个或者多个部分来覆盖该请求的一部分。在其他实施例中，中间设备 200 将一个或者多个 cookie 的一部分或者一个或者多个 cookie 的多个部分增加到该请求。仍在进一步的实施例中，中间设备 200 修改该请求中的 cookie。中间设备 200 可以改变该 cookie 中的一个或者多个值或者字符，或者以其他方式修改该 cookie 以通过服务器 106 可以接受。在又一个实施例中，中间设备 200 对该请求的一部分进行解密。在进一步的实施例中，中间设备 200 修改该请求来包括服务器 106 的任一配置相关的优先设置或者需求。修改的请求可以被改变，使得其可以通过服务器 106 来处理。中间设备 200 可以将该修改的请求传送给服务器 106。

[0163] 服务器 106 可以处理该修改的请求，并且随之将对该请求的响应传送给中间设备 200。在一些实施例中，对该请求的响应包括 HTTP 传输。在进一步的实施例中，该响应包括 web 页面。在又一个实施例中，该响应包括客户机 102 所请求的文件。在进一步的实施例中，该响应包括验证服务器 106 上的客户机 102 的验证消息。在又一个实施例中，该响应包括用于启动或者打开客户机 102 和服务器 106 之间的会话或连接的传输。在进一步的实施例中，该响应包括可执行文件、程序、函数、数据、流文件或者服务器 106 所提供的任一其它资源或者服务。在一些实施例中，服务器 106 将对于该请求的多个响应传送给中间设备 200。

[0164] 中间设备 200 可以将来自 HTTP 响应的任一 cookie 发送给 cookie 管理器 420，以增加到 cookie 管理器 420 的映射图上。增加到 cookie 管理器的映射图的任一新的 cookie 可以用于在客户机 102 和服务器 106 之间的未来的传输。在一些实施例中，中间设备修改该回复以将 cookie 从回复排除。在其他实施例中，中间设备从回复中取出 cookie 并且将 cookie 传送给 cookie 管理器 420。Cookie 管理器 420 可以将所接收的 cookie 和已经保存在映射图中的 cookie 相比较。在一些实施例中，cookie 管理器 420 响应于确定所接收的 cookie 没有之前保存在和客户机 102 或者服务器 106 相关的映射图来保存所接收的 cookie。所保存的 cookie 可以之后用于同一客户机 102 的通信。在一些实施例中，到 cookie 管理器 420 的传输包括涉及客户机 102、服务器 106 或者客户机 102 和服务器 106 二者的任意数量个 cookie、或者一部分、cookie 或者唯一客户机标识符。

[0165] 中间设备 200 修改该响应来将 cookie 从 HTTP 响应移除。中间设备还可以使用 cookie 管理器 420 来增加任一客户机 102 相关的 cookie 到 HTTP 响应，如果这样的 cookie 存在或者是必要的。在一些实施例中，中间设备修改、编辑或者改变该响应来排除任一服务器 106cookie。在进一步的实施例中，中间设备 200 修改该响应来使用客户机 102cookie 来替代服务器 106cookie。在又一个实施例中，中间设备 200 修改该响应来满足客户机 102 的格式、配置或者优先设置，使得修改的响应可以由客户机 102 接受或者使用。当处理每个响应或者请求时，中间设备 200 可以确定服务器侧 cookie 管理或者客户机侧 cookie 管理是否可用。在一些实施例中，当服务器侧 cookie 管理可用时，中间设备 200 可以调用函数，诸如 cookie 管理器 420 函数 ProcessRequest()。在多个实施例中，当客户机侧 cookie 管理可用时，中间设备 200 可以调用 cookie 管理器 420 函数 ProcessRequest() 或者执行 cookie 或者唯一标识符管理或内部 cookie 映射图管理的另一个函数。Cookie 管理器 420 可以检查来自接收的响应或者请求的 cookie，查找服务器或者客户机消耗的 cookie。在一些实施例中，cookie 管理器将从请求或者响应接收的 cookie 和在映射图中保存或者列出的 cookie

匹配。Cookie 管理器 420 还可以增加名称值对，如果这样的 cookie 或者客户机标识符没有之前注册、列在或者分配到映射图中，还需要在映射图中注册或者分配新的 cookie 或者新的客户机标识符。Cookie 管理器 420 可以因此将新的客户机标识符或者新的 cookie 增加到映射图中，该新的客户机标识符或者新的 cookie 可以用在未来和与这样的 cookie 或者唯一标识符相关的客户机和服务器的通信。在多个实施例中，cookie 或者 Set-cookie 首部可以在发送返回中间设备 200 时从该响应移除。通过禁止服务器消耗的 cookie 发送到客户机，系统可以将给定域的多个 cookie 扩展到超过预定限制的数量。在将 cookie 数量限制到最大 20 的系统中，该特征可以用于使得客户机 102 能够持续使用服务器 106 上的服务，而不会达到 cookie 最大二十的限制。这样的实践还可以禁止重要的 cookie 数据在向外发送敏感信息的防火墙以外的网络空间被访问或者读取。

[0166] 中间设备 200 将修改的 HTTP 响应传送给客户机 102。在一些实施例中，修改的响应包括客户机 102cookie。在其他实施例中，修改的响应包括服务器 106 发送到中间设备 200 的原始请求。在进一步的实施例中，修改的响应包括根据用于处理修改的响应的客户机 102 或者客户机 102 应用或者函数的配置或者标准重新格式化的响应的一部分。中间设备 200 可以响应于所接收的修改的响应来将发往服务器 106 的另一个请求传送到中间设备 200。

[0167] 中间设备 200 可以应用访问配置文件、策略、规则和动作的任一个到通过中间设备 200 的网络业务量的任一级别粒度的部分或者子集。该粒度级别可以基于该配置从细到粗。此处描述的访问配置文件、规则和策略的规则条件、标准或者逻辑可以限定或者指定为应用到经由设备 200 传输的网络业务量或者传输的任意期望子集或者部分。在一个方面，粒度级别是指配置可以应用到的网络业务量的一部分的级别、测量值、细度或者粗糙度。在配置的十分宽或者粗糙的粒度中，访问配置文件、规则或者策略可应用到所有的网络业务量。在十分细的粒度配置中，访问配置文件或者策略可以应用到特定用户的网络业务量的指定子集，诸如特定用户的特定应用的业务量或者业务量的一部分。在一些粒度的配置中，访问配置文件、策略或者规则应用到发送请求到服务器的任一客户机 102。策略、规则或者访问配置文件可以限定为用于或者应用到任一客户机 102，并且可以基于客户机 102 的任一配置或者涉及客户机 102 的信息，诸如客户机 102 请求的一部分。类似地，策略、规则或者访问配置文件可以限定为用于或者应用到任一服务器 106，并且可以基于客户机 106 的任一配置或者涉及服务器 106 的信息，诸如服务器 106 响应的一部分。在一些粒度的配置中，访问配置文件、策略或者规则限定为应用到客户机 102 经由设备 200 用来连接到服务器 106 的指定会话或者连接。

[0168] 在进一步的实施例中，访问配置文件、策略或者规则限定为应用到经由 SSL VPN 会话或者连接来连接的任一客户机 102。在进一步的实施例中，访问配置文件、策略或者规则限定为应用到经由免客户机 SSL VPN 会话或者连接来连接的任一客户机 102。在进一步的实施例中，访问配置文件、策略或者规则限定为应用到经由基于客户机的 SSL VPN 会话或者连接来连接的任一客户机 102。在进一步的实施例中，访问配置文件、策略或者规则限定为应用到发送请求到特定服务器 106 的任一客户机 102 或者客户机会话。在又一个实施例中，访问配置文件、策略或者规则限定为应用到请求服务器上特定应用或者资源的任意客户机 102 或者客户机会话。在进一步的实施例中，访问配置文件、策略或者规则限定为基于例如

cookie 是否被启用或者停用的 cookie 配置应用到任一客户机 102 或者客户机会话。在进一步的实施例中，访问配置文件、策略或者规则限定为应用到发送包括特定 URL、或者特定 URL 的一部分的请求的任一客户机 102 或者客户机会话。在又一个实施例中，访问配置文件、策略或者规则限定为基于通过客户机 102 发送的请求的一部分和访问配置文件、策略或者规则的短语或者键值之间的匹配来应用到任一客户机 102 或者客户机会话。在一些实施例中，访问配置文件、策略或者规则限定为基于涉及访问服务器 106 的客户机 102 的信息来应用到任一服务器 106 或者服务器会话。这样的信息可以包括客户机 102 的请求的一部分或者特征、客户机 102 的设置或者配置或者任一其它客户机 102 相关的信息。在一些实施例中，访问配置文件、策略或者规则限定为基于服务器 106 的配置或者服务器 106 发送到客户机 102 的内容的特征来应用到任一服务器 106 或者服务器会话。

[0169] 现在参考图 4C，示出 cookie 代理数据流控制的实施例。总的来说，图 4C 描述客户机 102 经由管理客户机 102 和服务器 106 之间的 cookie 流的中间设备 200 和服务器 106 进行通信。客户机 102 经由设备发送请求给服务器 106，设备也称为中间设备 200。该请求包括诸如“GET/index.html HTTP/1.1”的 URL。中间设备 200 拦截通过客户机 102 发送的请求并且将该请求转发给服务器 106。服务器 106 响应于该请求发出包括 cookie 的响应，cookie 诸如“″ HTTP/1.1200OK\nSet-Cookie:name = value”。中间设备 200 剥离并且保存该 cookie 并且使用唯一识别客户机 102 的客户机 ID 来替换该 cookie。唯一客户机 ID 可以是合并的 cookie 并且可以和中间设备 200 中的 cookie 相关联用于未来的传输。中间设备 200 将包括唯一客户机 ID 的修改的响应转发给客户机 102。修改的服务器的响应可以包括诸如“HTTP/1.1200OK\nSet-cookie:NSC_AAAC = Unique client ID”的信息。客户机 102 使用唯一的客户机 ID 来发送第二请求，诸如“GET/foo.html HTTP/1.1\nCookie :NSC_AAAC = Unique client ID”。中间设备 200 接收第二请求并且使用该唯一客户机 ID 获取基于唯一客户机 ID 保存的 cookie。中间设备 200 修改该请求并且将之前保存的 cookie 插入到该请求中。修改的第二请求可以格式化为包括在之前传输中所使用的相同的或者类似的 cookie，诸如“GET/foo.html HTTP/1.1\nSet-Cookie:name = value”。中间设备 200 将修改的第二请求发送到服务器 106。

[0170] 进一步参考图 4C，客户机 102 可以通过发送请求到中间设备 200 初始化和服务器 106 的通信。在一些实施例中，该请求可以是 HTTP 请求，诸如“GET/index.html HTTP/1.1”。在一些实施例中，客户机 102 尝试和服务器 106 进行第一次通信。在进一步的实施例中，客户机 102 将意图发往服务器 106 的任一 HTTP 请求发送给中间设备 200。客户机 102 可以将任一请求传送到中间设备 200 来访问服务器 200 上的资源或者服务。

[0171] 中间设备 200 可以将该请求转发给服务器 106。在一些实施例中，中间设备 200 修改该请求并且将修改的请求转发给服务器 106。在其他实施例中，中间设备 200 不修改该请求。在进一步的实施例中，中间设备 200 将该请求转发给服务器 106，而不会修改请求的任一部分。在一些实施例中，中间设备将 HTTP 请求转发给中间设备 200，HTTP 请求诸如“GET/index.html HTTP/1.1”。

[0172] 服务器 106 可以发布针对该请求的包括 cookie 的响应。在一些实施例中，该响应可以是例如“HTTP/1.1200OK\nSet-Cookie:name = value”。Cookie 可以在响应的首部中或者在响应的任一其它部分中。在一些实施例中，cookie 可以包括在响应的 URL 中。该

cookie 可以是任一类型和形式的 cookie 并且可以包括在响应中的任一位置。

[0173] 中间设备 200 可以将 cookie 从响应剥离并且保存并使用唯一识别客户机 102 的客户机 ID 来替换 cookie。一旦服务器 106 的响应通过中间设备 200 接收，则中间设备可以建立 cookie 存储器，用于客户机的给定域。Cookie 存储器可以包括或者保存给定域和诸如客户机 102 的客户机的 cookie 的任一集合。在一些实施例中，cookie 存储器可以是文件、列表、数据库、阵列、数据结构或者文件夹，其包括任意数量的 cookie 或者通过 cookie 所包括的任意数量的信息。在一些实施例中，中间设备 200 可以从通过服务器 106 发送的响应首部剥离“Set-Cookie”首部，并且其可以将 Set-Cookie 首部保存在 cookie 存储器中。中间设备 200 还可以产生唯一的 cookie 代理会话 cookie。Cookie 代理会话 cookie 可以包括客户机 102 可以从服务器 106 发送的 cookie 接收的任一相关信息，而不会实际接收该 cookie。Cookie 代理会话 cookie 因此可以将所有来自服务器 106 发送的 cookie 的相关信息以客户机 102 或者客户机 102 的 web 浏览器可接受的方式传送给客户机 102。中间设备 200 可以将 Cookie 代理会话 cookie 插入到 cookie 存储器或者将 Cookie 代理会话 cookie 关联到服务器 106 发送的相关的和相对应的 cookie。在多个实施例中，中间设备 200 可以将 Cookie 代理响应 cookie 插入到从中间设备 200 发送到客户机 102 的消息的响应首部。在一些实施例中，中间设备 200 可以不改变域和路径，而在其它实施例中，中间设备可以改变域或者路径，或者域和路径二者。在一些实施例中，中间设备 200 将唯一客户机 ID 插入到从中间设备发送到客户机 102 的通信的任一部分。

[0174] 中间设备 200 可以将修改的响应转发给客户机 102。修改的响应可以包括唯一的客户机标识符，诸如唯一的客户机 ID。在一些实施例中，修改的响应可以包括客户机 102cookie。在进一步的实施例中，修改的响应修改为根据客户机 102 的配置。在一些实施例中，修改的响应包括服务器 106 的原始响应，而没有任一改变。修改的服务器的响应可以包括任一 HTTP 格式的信息，诸如“HTTP/1.12000K\nSet-Cookie:NSC_AAAC = Unique clientID”。

[0175] 客户机 102 使用唯一客户机 ID 发送第二请求。第二请求可以和第一请求相同、类似或者大体类似。在一些实施例中，第二请求包括和第一请求一样的形式。在进一步的实施例中，第二请求来自传输第一请求的相同应用。在一些实施例中，第二请求是 HTTP 请求，诸如“GET/foo.html HTTP/1.1\nCookie:NSC_AAAC = Unique client ID”。第二请求可以包括通过中间设备 200 发布的唯一客户机 ID 或者唯一客户机 ID 的一部分。

[0176] 中间设备 200 可以修改第二请求来包括和服务器 106 相关的 cookie。中间设备 200 可以使用和服务器 106 的 cookie 相关保存的唯一客户机 ID 来获取服务器 106cookie。中间设备 200 可以修改第二请求并且将之前保存的 cookie 插入到第二请求。中间设备 200 可以修改该请求并且将之前保存的 cookie 插入到该请求中。在一些实施例中，中间设备 200 修改该请求以将之前保存的 cookie 包括在第二请求中。中间设备 200 可以检查 cookie 代理会话 cookie 是否存在。在 cookie 代理会话 cookie 存在的情况下，中间设备 200 基于 cookie 代理会话 cookie 来搜索 cookie 存储器。中间设备 200 可以使用通过使用 cookie 代理会话 cookie 来获取的 cookie 来找到目的域和路径。中间设备 200 还可以将 cookie 或者多个 cookie 插入到请求代码路径中和 / 或剥离 cookie 代理会话 cookie。

[0177] 在一些实施例中，代理 200 为了标记响应路径可以保存从服务器侧 PCB 到 cookie

存储器的基准指针。在进一步的实施例中，代理 200 为了标记响应路径在 cookie 中保存从会话信息开始的基准指针。在一些实施例中，如果已经为包括指定客户机 102 和服务器 106 的特定会话建立 cookie 存储器，则中间设备 200 在同一会话中的第二组通信期间不能建立 cookie 存储器。相反，中间设备 200 可以使用和之前对于相同会话所使用的相同的 cookie 存储器。在一些实施例中，中间设备 200 可以已经具有对于客户机 102 和服务器 106 所建立 cookie 存储器的基准。在多个实施例中，对于 cookie 存储器的基准可以通过协议控制块或者 PCB、控制器、以及任何软件、数据库、阵列或者包括任一组值的结构来实现。在某个实施例中，如果客户机停用 cookie，则来自客户机的随后的请求不可以包括任一 cookie 代理会话 cookie。在一些实施例中，如果客户机停用 cookie，则来自客户机的随后的请求可以包括 cookie 代理会话 cookie。在一些实施例中，从服务器 106 到客户机 102 的随后的响应可以建立不再参考 cookie 存储器所用于的客户机 102 或者服务器 106 的 cookie 存储器。在进一步的实施例中，从服务器 106 到客户机 102 的随后的响应可以建立参考 cookie 存储器所用于的客户机 102 或者服务器 106 的 cookie 存储器。

[0178] 在多个实施例中，其中客户机 102 和服务器 106 在给定会话中超过一次通信，中间设备 200 在第一次通信之后不需要再发送 cookie 代理会话 cookie。中间设备可以使用唯一客户机识别方法来唯一识别客户机 102 或者服务器 106。在一些实施例中，唯一客户机识别方法也称为唯一客户机 ID 可以用来唯一识别和服务器 106 通信的客户机 102 或者和客户机 102 通信的服务器 106。在多个实施例中，唯一客户机 ID 可以用来唯一识别经由中间设备 200 和服务器 106 通信或者发送消息或者请求到服务器 106 的客户机 102。在多个实施例中，唯一客户机 ID 可用来唯一识别经由中间设备 200 进行通信或发送消息或者请求至服务器 106 的服务器 106。在一些实施例中，中间设备 200 使用唯一客户机 ID 来检测和确定是否代理客户机 102 所发送的通信。

[0179] 在多个实施例中，中间设备 200 可以执行清除 cookie，确定不必要的 cookie 或者终止不必要的 cookie。在多个实施例中，中间设备 200 可以使用从客户机 102 的 PCB 到为客户机 102 所建立的 cookie 存储器的基准指针。修改的第二请求可以格式化为包括在前一传输中所使用的相同或相似的 cookie。在一些实施例中，修改的第二请求是如修改的第二 HTTP 请求的 HTTP 请求，诸如“GET/foo.html HTTP/1.1\nSet-Cookie:name = value”。

[0180] 中间设备 200 可以将修改的第二请求传送到服务器 106。在一些实施例中，中间设备 200 将任一数量个修改的请求传送给服务器 106。修改的请求可以是任意类型、形式和格式。中间设备 200 因此可以利用 cookie 存储器来回传送来自客户机 102 的任一数量个请求和来自服务器 106 的响应。

[0181] Cookie 代理可以是任一模块控制、管理或者重新构形的 cookie 或者利用配置设置的 cookie 的传输。在一些实施例中，cookie 代理可以是 cookie 管理器 420。在多个实施例中，cookie 代理可以是 cookie 管理器 420 的一部分或者子部件。在多个实施例中，cookie 代理可以包括 cookie 管理器 420。在某个实施例中，cookie 代理可和 cookie 管理器 420 交换使用，并且可以包括 cookie 管理器 420 的任一和全部功能性和性能装置。在一些实施例中，cookie 管理器 420 可以称为 cookie 代理。在多个实施例中，cookie 代理可以独立于设备 200 或者 cookie 管理器 420。在某个实施例中，cookie 代理可以是能够独立于中间设备 200 或者 cookie 管理器 420 执行的或者和中间设备的 200 或者 cookie 管理器

420 一起执行的软件程序或者应用。

[0182] Cookie 代理可以包括基于策略或者策略动作的配置设置。在多个实施例中，用户或者管理员可以配置 cookie 代理来确定哪个 cookie 保存到 cookie 存储器中和哪个 cookie 不保存在 cookie 存储器中。在多个实施例中，用户或者管理员对于任一指定域，诸如 www.foo.com，可以确定中间设备 200 是否应该代理 cookie1 并且允许 cookie2、cookie3 流过，如同 cookie1、cookie2 和 cookie3 中的任一个可以或者不可以是客户机消耗的 cookie。

[0183] 在一些实施例中，cookie 代理配置可以使用命令行接口 (CLI) 语法，诸如：

[0184] add/delete/set/unset/show cookieproxy action<action-name>

[0185] <ALL[-EXCEPT<cookie-name>,[<cookie-name>,...]]\

[0186] <cookie-name>,[<cookie-name>,...]>

[0187] 此外，cookie 代理配置还可以包括其它语法，诸如：

[0188] add/delete/set/unset/show cookieproxypolicy<name><rule><jar name>

[0189] [-CookieProxyAction<action-name>][<undefAction>]

[0190] 在多个实施例中，如果用户或者管理员没有指定动作，则 cookie 代理的缺省行为可以来代理所有的 cookie，或者不代理任何 cookie，或者代理如通过和 cookie 代理相关的一组策略确定的 cookie 的一部分。

[0191] 汇集操作可以将诸如策略的配置和诸如用户或者资源的实体汇集、集合、联系在一起或相关联。汇集动作可以将配置置于应用到所分配实体的有效状态。在一些实施例中，汇集操作可以将一个实体和另一个实体相关联或者将模块的功能性应用到实体。汇集操作可以是通过 cookie 代理、或者通过 cookie 代理的策略所执行的操作。在一些实施例中，通过配置命令，cookie 代理可以汇集到虚拟服务器 275。在多个实施例中，cookie 代理策略可以汇集到负载平衡服务器，GSLB 服务器或者 VPN 服务器。Cookie 代理或者中间设备 200 所使用的策略规则可以是基于任一策略基础结构规则语言 (PIRL) 的。在一些实施例中，cookie 代理或者中间设备 200 所使用的策略可以在响应时间或者在中间设备或者 cookie 代理响应请求的时间或者二者期间来评估。在多个实施例中，cookie 代理或者中间设备 200 所使用的策略可以在响应时间期间来评估或者执行。

[0192] 在一些实施例中，cookie 代理方法可以使用分配给每一客户机的唯一客户机 ID。唯一客户机 ID 可以由中间设备 200 用来将和特定客户机 102 相关的 cookie 存储器映射到客户机 102。在多个实施例中，客户机也称为客户机 102，可以不随同事务一起发送任一唯一客户机 ID，该事务也称为通信或者请求。在多个实施例中，cookie 代理会话 cookie 可以用作缺省客户机识别机制或者唯一客户机 ID。在一些实施例中，客户机识别可以基于客户机因特网协议地址、请求或者 HTTP 通信的片段、客户机 102 或者服务器 106 所发送的通信的唯一部件、涉及会话的唯一特征、SSL VPN 会话 cookie 或者 SSL VPN 会话体。在多个实施例中，客户机识别是可配置的。在多个实施例中，客户机识别可以利用客户机 102 的因特网协议地址（也称为 IP 地址）来实现。

[0193] 在决定诸如 cookie 存储器的大小或者涉及 cookie 存储器的定时（例如 cookie 存储器的空闲定时）的多个参数过程中，可以完成 cookie 存储器清除或者 cookie 存储器消除。在一些实施例中，cookie 存储器清除方法可以使用空闲时间或者存储器阈值的超

时。在多个实施例中，cookie 存储器清除可以基于和 cookie 自身相关的配置或者会话超时来实现，会话超时可以导致属于该会话的所有 cookie 在超时后被清除。在一些实施例中，cookie 存储器清除方法可以基于自特定 cookie 或者 cookie 存储器最后被使用或者访问所经过的时间多少来确定哪个 cookie 被清除。

[0194] 在一些实施例中，CLI 语法可以用于存储器和客户机识别，诸如：

[0195] add/delete/set/unset/show cookieproxy jar<jar-name>

[0196] -clientidentification<default\request based PIXL expression>

[0197] -maxMem<Memory limit>

[0198] 在多个实施例中，缺省可以是，在不支持 cookie 时，使用由 cookie 代理模块插入到首部或者 URL 中的会话 cookie。在多个实施例中，策略规则可以用来找到任一唯一首部字段，例如用于 LB 负载平衡的会话 cookie 或者用于 SSL VPN 的 cookie。

[0199] 在多个实施例中，当任一 cookie 存储器达到最大存储器限制时，最长时间周期无效的会话可能需要被超时并且属于该会话的 cookie 可能需要被清除。在多个实施例中，时间戳可以关联到每个客户机 102、服务器 106、客户机 / 域组合、或者每个客户机 - 服务器会话、或者任一会话，用来实现基于 URL 的清除。

[0200] Cookie 代理或者中间设备 200 可以利用不同策略或者动作的运行时间的聚合。在多个实施例中，中间设备 200 或者 cookie 代理可以使用策略或者动作的运行时间的聚合来减少配置开销或者简化配置修改。在多个实施例中，中间设备 200 或者 cookie 代理可以配置为在第一匹配的策略处停止。在一些实施例中，管理员或者用户必须在配置时间期间聚合并且建立合适的策略和动作。在某些实施例中，cookie 代理动作的运行时间行为可以部分通过搜索策略或策略列表中的匹配并且在遇到匹配时在该策略处停止来规定。在一些实施例中，cookie 代理可以在限定或者包括要被代理的一组 cookie 或者涉及要被代理的一组 cookie 的信息的第一策略处停止，其可以减低运行时间的聚合。

[0201] Cookie 存储器基础结构可以用于 cookie 数据仓库，然而在一些实施例中，附加的 API 可以用于基于域或者路径来获取 cookie。在一些实施例中，基于哈希的搜索机制可以用来获取 cookie 代理会话。该方法可以类似于在其它中间设备 200 相关的应用中所用的 SSL VPN 会话哈希机制。在多个实施例中，哈希函数的键值可以依赖于客户机识别机制或者客户机识别协议。在多个实施例中，对于不同客户机识别机制可以利用不同的哈希函数。

[0202] Cookie 数据仓库管理可以依赖于性能或者资源。在一些实施例中，cookie 数据仓库通过关联每个域每个客户机的 cookie 存储器来管理。在多个实施例中，该组 cookie 名称包括反复在每个 cookie 存储器中保存的 cookie 名称。在多个实施例中，所使用的该组 cookie 名称在 cookie 存储器之间不同，而在其他实施例中，在多个 cookie 存储器之间所使用的该组 cookie 名称包括一些相同或者相似的名称。在一些实施例中，cookie 存储器组织成使得 cookie 存储器可以和特定客户机相关联并且和处理该传输的特定虚拟服务器相关联。在多个实施例中，cookie 存储器组织成使得 cookie 存储器和客户机、虚拟服务器和域相关联。

[0203] 在一些实施例中，cookie 存储器不需要保存和存储器中另一个 cookie 名称相同的 cookie。在某些实施例中，cookie 存储器可以包括具有和在诸如 HTTP 首部的首部中利用的方法类似的方法中的 cookie 名称相关的值的 cookie 名称。

[0204] Cookie 代理可以包括多个功能性。在一些实施例中，cookie 代理可以利用 cookie 代理或者此处讨论的任意数量个实施例来确定浏览器是否可以处理或者接受 cookie。在多个实施例中，cookie 代理可以确定客户机 102 或者服务器 106 发送的请求是否匹配某个标准，用来确定来自客户机或者服务器的浏览器是否接受或者处理 cookie。在一些实施例中，如果通过客户机 102 或者服务器 106 发送的请求由于能够接受或者处理 cookie 的标准而匹配标准，则中间设备 200 可以发送重定向消息给客户机，诸如：

[0205] http://incoming_host/incoming_url ? new_param_added = secure_client_id

[0206] along with a secure_client_id set-cookie.

[0207] 在一些实施例中，如果客户机返回请求 URL，诸如“http://incoming_host/incoming_url ? new_param_added = secure_client_id”，则中间设备可以验证唯一的客户机 ID 是否和 cookie 值相关联。如果中间设备检测到匹配，中间设备可以剥离已经增加的参数并且可以处理原始请求。此外，中间设备还可以将 cookie 代理会话标记为使用“cookie 代理会话 cookie”。在唯一客户机 ID 不匹配 cookie 值的情况中，中间设备可以利用诸如体重写的不同方法用于会话跟踪。在这样的方法中，响应体可以重写以在每个 HTTP 链路中包括会话信息。

[0208] 在一些实施例中，cookie 代理或者中间设备 200 还可以包括 cookie 代理 cookie 存储器，也称为 cookie 代理会话 cookie 存储器。Cookie 代理 cookie 存储器可以包括任一多个 cookie 代理 cookie，也称为 cookie 代理会话 cookie。在多个实施例中，Cookie 代理 cookie 存储器可以以类似于 cookie 存储器的方式来组织或者实现，并且可以包括 cookie 存储器的所有功能性。

[0209] 在某些实施例中，Cookie 代理 cookie 存储器可以和高可用性的应用和技术一起运行，高可用性的应用和技术也称为 HA 技术。在一些实施例中，包括一组 cookie 和唯一客户机 ID 的中间设备可以发送 cookie 和客户机 ID 到网络上的其他设备。在多个实施例中，包括与客户机 102 或者服务器 106 相关的 cookie 和唯一客户机 ID 的第一中间设备 200 可以和第二中间设备 200 或者多个设备 200 共享 cookie 或者唯一客户机 ID 相关的信息。在第一中间设备和第二设备共享 cookie 或者唯一客户机 ID 相关的信息的情况中，第二中间设备还可以使用 cookie 和唯一客户机 ID 来实现客户机和服务器之间的通信。

[0210] Cookie 代理、cookie 管理器 420 或者中间设备 200 可以包括以脚本或者软件实现的任一数量个软件应用程序或者函数来建立并且管理 cookie。在一些实施例中，cookie 代理、cookie 管理器 420 或者中间设备 200 可以包括用于管理 cookie 存储器的软件代码，诸如：

[0211] /* AppSecure Cookie-jar API. */

[0212] /* Create an empty cookie jar */

[0213] as_cookie_jar_t * as_cookie_jar_create(as_allocator_t * allocator);

[0214] /* Get value of a cookie, given name */

[0215] as_cookie_t * as_cookie_jar_get(as_cookie_jar_t * cookie_jar, astr_t * name);

[0216] /* Add a cookie to the jar. If nodup is set, and a previous cookie exists with the same

[0217] name,path,domain,then delete it before adding the new one * /

[0218] ns_status_t as_cookie_jar_add(as_cookie_jar_t * cookie_jar,as_cookie_t * cookie,int nodup) ;

[0219] / * Delete cookies with same name,value,path and domain as cookie * /

[0220] ns_status_t as_cookie_jar_delete(as_cookie_jar_t * cookie_jar,as_cookie_t * cookie) ;

[0221] / * Delete all name-value pairs given name * /

[0222] ns_status_t as cookie_jar_delete_by_name(as cookie_jar_t * cookie_jar,astr_t * name) ;

[0223] / * Destroy cookie-jar * /

[0224] void as_cookie_jar_destroy(as_cookie_jar_t * cookie_jar) ;

[0225] / * Parse an http Cookie header cookie string into multiple cookies and add * them to the cookie jar * /

[0226] ns_status_t as cookie_jar_parse_cookie(as_cookie_jar_t * cookie_jar,const astr_t * cookie_string) ;

[0227] / * Parse an http Set-Cookie header string into multiple cookies and add them to the cookie jar * /

[0228] ns_status_tas_cookie_jar_parse_set_cookie(as_cookie_jar_t * cookie_jar,const astr_t * cookie_string) ;

[0229] / * Stringify cookie jar to use as cookie value in an http request * /

[0230] astr_t * as_cookie_jar_to_cookie_string(as_allocator_t * allocator,as_component_t owner,as_cookie_jar_t

[0231] * cookie_jar) ;

[0232] / * Stringijy cookie jar to use as the set-cookie value in the http response * /

[0233] astr_t * as_cookie_jar_to_set_cookie string(as_allocator_t * allocator,as_component_t owner,as_cookie_jar_t

[0234] * cookie_jar) ;

[0235] /*Create an iterator*/

[0236] as_cookie_jar_iterator_t * as_cookie_jar_iterator_create(as _allocator_t *allocatoras_component_t owner_id,as_cookie_jar_t*cookie_jar) ;

[0237] int as_cookie_jar_iterator_init(as_allocator_t*allocator,as_component_t owner_id,

[0238] as_cookie_jar_t*cookie_jar,as_cookie_jar_iterator_t*iter) ;int

[0239] as_cookie_jar_iterate(as_cookie_jar_iterator_t*iter,as_cookie_t**cookie) ;

[0240] void as_cookie_jar_iterator_destroy(as_allocator_t*allocator,

[0241] as_cookie_jar_iterator_t*iter) ;

[0242] int as_cookie_jar_size(as_cookie_jar_t*cookie_jar) ;

[0243] as_cookie_jar_iterator_create(as_allocator_t *allocator, as_component_t owner_id, as_cookie_jart_t*cookie_jar) ;

[0244] cookie 管理器或者代理可以响应于策略引擎的一个或者多个策略（包括这些策略的任一规则、条件或者动作）确定何时、如何并且把哪个 cookie 来管理和 / 或保存到 cookie 存储器中，以及上述任一操作。任意策略和对应的 cookie 操作可以基于会话。在多个实施例中，策略引擎 236 提供策略或者规则，通过该策略或者规则来确定关于该 cookie 管理的动作。在一些实施例中，策略引擎 236 可以包括策略或规则的列表，为中间设备 200 或者 cookie 管理器 420 提供一种方法来确定关于要执行的 cookie 或者唯一客户机 ID 的动作。由此，通过配置和策略，中间设备可以提供用于 cookie 管理的细粒度控制，包括对于免客户机 SSL VPN 访问。

[0245] 在一个例子中，服务器可以提供对经由设备 200 访问不同应用的多个客户机的访问。两个这样的应用可以是应用 1 和应用 2。这两个应用可以使用 ASP.NETSESSIONID，其可以是不在客户机侧使用及写入的服务器消耗的 cookie。除了 ASP.NETSESSIONID，应用 1 还可以使用 cookieAppClientInfo，cookie AppClientInfo 可以通过访问或者使用应用 1 的第一客户机读和写，但不由访问同一应用的第二客户机读写。

[0246] 在这样的实施例中，设备 200 处理这样或类似情况的配置可以是 ：

[0247] add patclass app 1_clientconsumed_cookies

[0248] bind patclass app1_clientconsumed_cookies AppClientlnfo

[0249] set vpn clientlessAccessProfile app1_profile-ClientConsumedCookies

[0250] app1_clientconsumed_cookies

[0251] add vpn clientlessAccessPolicy app 1_access_pol

[0252] " http.req.url.path.get(1).eq(\" app1")" app1_profile

[0253] 在访问 web 应用 1 和应用 2 时产生的 URL 使用条目 :/app1 来识别。上例中表达的策略对以下实例或者情况可评估为真，即，存在所接收的 HTTP 请求的 URL 路径以“/app1/”开始的实例或者情况。这样的 HTTP 请求的一个例子是“GET/app1/display.asp”。因此，该请求（应用 1，或 app1）除了命名为 AppClientInfo 的 cookie 以外的所有 cookie 将被代理。

[0254] 在进一步的例子中，应用 2 使用可以用在客户机侧或者通过客户机使用的 App2ClientCookie1 和 App2ClientCookie2，但是由其所使用的其它 cookie 的剩余部分无需被表示。这样的配置可以是 ：

[0255] add patclass app2_clientconsumed_cookies

[0256] bind patclass app2_clientconsumed_cookies App2ClientCookie1

[0257] bind patclass app2_clientconsumed_cookies App2ClientCookie2

[0258] set vpn clientlessAccessProfile app2_profile-ClientConsumedCookies

[0259] app2_clientconsumed_cookies

[0260] add vpn clientlessAccessPolicy app2_access_pol

[0261] " http.req.hostname.set_text_mode(ignorecase).eq(\" app2\")" app2_profile

[0262] 在此配置中，应用 2（称之为 App2）可以寄载在主机名为 app2 的 web 服务器

上，因此，对于应用 2(App2)，除了 App2ClientCookie1 和 App2ClientCookie2 外的所有 cookie 将被代理。在这些以及类似例子中，管理员可以配置将为应用 2 代理的、具有名称 AppClientInfo 的相同 cookie，而不为应用 1 代理此 cookie。类似地，管理员可以基于这样或者类似的用于策略的配置而配置代理或者不代理具有任一名称或者和任一服务或者资源相关的 cookie、或者任一客户机 102 或者服务器 106。

[0263] 在又一个实施例中，配置可以为设置所有站点的所有 cookie 以免客户机 VPN 模式来代理。不应该代理的客户机消耗的 cookie 可以通过指定 patclass 命令或者指令中的 cookie 的名称来配置。例如，如果一些应用需要两个 cookie 出现在客户机侧，如 Cookie1 和 Cookie2，则配置可以标识为：

[0264] add patclass app_bypass_cookies

[0265] bind patclass app_bypass_cookies Cookie1

[0266] bind patclass app_bypass_cookies Cookie2

[0267] set vpn clientlessAccessprofile<app_profile>-ClientConsumedCookies

[0268] app_bypass_cookies

[0269] 用于该配置文件的该代码可以之后在 clientlessAccessPolicy 指令中使用，诸如：

[0270] add vpn clientlessAccessPolicy<policyName><rule><vpnclientlessAccessProfile>

[0271] 上述策略可以使用免客户机访问来选择免客户机访问配置文件，使得代理除了具有名称为 Cookie1 和 Cookie2 的 Cookie 之外的所有 Cookie。从而，对于通过策略规则识别的业务量的子集，不可以代理给定组的 Cookie。策略规则可以用来选择特定的 web 应用或者特定的服务器或者服务器上的目录。使用类似上述的配置，对于不同用户组、用户集或者 vpn 虚拟服务器可以代理不同组的 Cookie，其任一个可以依赖于策略的配置限定或者处理哪个实体。

[0272] 现在参考图 5，示出用于经由中间设备的配置驱动 Cookie 代理的方法的步骤的实施例。总的来说，在步骤 505，中间设备从服务器经由 SSLVPN 会话接收包括一个或者多个 Cookie 的响应。在步骤 510，中间设备识别具有用于代理一个或者多个 Cookie 的一个或者多个策略的一个或者多个访问配置文件。在步骤 515，中间设备响应于一个或者多个策略来确定代理一个或者多个 Cookie 或者将忽略一个或者多个 Cookie。在步骤 520，中间设备代理一个或者多个 Cookie 或者忽略代理一个或者多个 Cookie 并且将响应转发给客户机。

[0273] 更详细地，在步骤 505，中间设备 200 从服务器 106 经由任一会话或者连接来接收包括一个或者多个 Cookie 的响应。在一些实施例中，中间设备 200 从任意数量个服务器 106 接收任意数量个响应，诸如 HTTP 响应或者包括客户机 102 请求的内容或者服务的响应。在一些实施例中，中间设备 200 从服务器 106 经由 SSL VPN 会话接收响应。在进一步的实施例中，中间设备 200 从服务器 106 经由免客户机 SSL VPN 会话来接收响应。在进一步的实施例中，中间设备 200 从服务器 106 经由基于客户机的 SSL VPN 会话来接收该响应。在一些实施例中，该响应包括一个或者多个 Cookie。该响应可以包括服务器消耗的 Cookie，或者通过服务器 106 使用、读出、编辑或者写入的 Cookie。在进一步的实施例中，该响应包括客户机消耗的 Cookie 或者客户机侧 Cookie，或者通过客户机 102 使用、读出、编辑或者写入的 Cookie。在又一个实施例中，该响应包括通过服务器 106 和通过客户机 102 使用、

读出、编辑或者写入的 Cookie。在一些实施例中，该响应包括任一类型、形式和类别的多个 Cookie。该 Cookie 可以用来跟踪客户机会话、维持客户机相关的信息或者协助服务器 106 和客户机 102 来共享或者跟踪信息。在一些实施例中，该响应包括 URL。该 URL 可以包括 Cookie 或者部分 Cookie，诸如 Cookie 值。

[0274] 在步骤 510，中间设备识别具有用于代理一个或者多个 Cookie 的一个或者多个策略的一个或者多个访问配置文件。中间设备 200 可以使用中间设备 200 的任一子部件、单元、函数或者装置来识别一个或者多个访问配置文件。在一些实施例中，诸如策略引擎 236 的策略引擎可以包括用于识别访问配置文件的一个或者多个策略。访问配置文件可以基于来自服务器 106 的响应的一部分来识别。访问配置文件可以基于来自服务器 106 的 URL 或者 URL 的一部分来识别。访问配置文件可以基于响应的一部分内容和与响应的一部分内容相关联的访问配置文件的一个或者多个策略的配置来识别。在一些实施例中，访问配置文件可以基于来自服务器 106 的响应的一部分和访问配置文件的配置之间的匹配来识别，该访问配置文件响应于该匹配来选择访问配置文件的一个或者多个策略。在进一步的实施例中，访问配置文件可以基于来自服务器 106 的响应的一部分和访问配置文件的策略的配置之间的匹配来识别，该访问配置文件触发一个或者多个匹配的策略的一个或者多个动作。在一些实施例中，访问配置文件基于涉及客户机 102 会话或者连接的信息和访问配置文件的策略的配置之间的匹配来识别，该访问配置文件触发策略的动作。在进一步的实施例中，访问配置文件可以基于来自响应的基于客户机的 Cookie 和访问配置文件的策略的配置之间的匹配来识别，该访问配置文件基于和基于客户机的 Cookie 的匹配来触发策略的动作。在进一步的实施例中，访问配置文件可以基于来自响应的服务器侧的 Cookie 和访问配置文件的策略的配置之间的匹配来识别，该访问配置文件基于和服务器 Cookie 的匹配来触发策略的动作。在进一步的实施例中，访问配置文件可以基于客户机的特定组中的客户机 102 和访问配置文件的策略的配置之间的匹配来识别，该访问配置文件基于和来自客户机组的任一客户机的匹配来触发策略的动作。在进一步的实施例中，访问配置文件可以基于客户机 102 正访问或者使用的特定应用或者资源或者应用或资源的类型和访问配置文件的策略的配置之间的匹配来识别，该访问配置文件基于该匹配来触发策略的动作。访问配置文件的策略的配置可以基于关于客户机 102、服务器 106、客户机和服务器所使用的连接或者会话的类型、客户机和服务器的配置、客户机 102 上的用户、关于客户机 102 所使用的软件或者应用的信息或者设备 200 可用的任一其它类型和形式的信息。在一些实施例中，访问配置文件可以基于客户机 102 正使用的会话或者连接的类型（诸如基于客户机或者免客户机 SSL VPN 会话）和访问配置文件的策略的配置之间的匹配来识别，该访问配置文件基于该匹配来触发策略的动作。在进一步的实施例中，访问配置文件可以基于关于客户机 102 所请求的文件、web 页面、应用、资源或服务的信息和触发该动作的访问配置文件的策略的配置之间的匹配来识别。在一些实施例中，中间设备为客户机 102 和服务器 106 之间的任一传输识别访问配置文件和策略。在一些实施例中，为服务器 106 的每单个响应或者客户机 102 的每单个请求识别一个访问配置文件和一个策略。在其他实施例中，为服务器的单个响应或者客户机的单个请求识别任意数量个访问配置文件和访问配置文件的任意数量个策略。

[0275] 在步骤 515，中间设备响应于一个或者多个策略来确定要执行的代理动作。在一

些实施例中，中间设备响应于一个或者多个已识别访问配置文件的一个或者多个已识别策略来确定代理还是不代理响应的 Cookie。在进一步的实施例中，中间设备确定如何代理该 Cookie。通过策略的配置触发的策略的动作可以包括代理 Cookie 或者忽略代理 Cookie。代理 Cookie 可以包括 Cookie 的任一处理，诸如产生、终止、插入、移除、重写或者以其他方式编辑或者变换响应的任一 Cookie。在一些实施例中，代理 Cookie 可以包括将来自服务器 106 的响应或者来自客户机 102 的请求的 Cookie 重写、编辑、移除或者插入。代理 Cookie 还可以包括中间设备 200 产生、终止、修改、插入或者移除来往传输的 Cookie，诸如通过设备 200 的来自服务器 106 的响应和来自客户机 102 的请求。策略可以确定设备 200 应该忽略代理或者设备不应该代理来自传输的 cookie。在这样的实施例中，设备 200 简单地不修改或者改变或者以其他方式处理来自传输的 cooKie。根据策略的配置或者设置，cookie 可以以多种方式来处理。在一些实施例中，设备 200 确定为通过该设备 200 的响应或者请求来产生新的 cookie。在进一步的实施例中，设备 200 确定修改来自响应或者请求的 cookie 的值。在进一步的实施例中，设备 200 确定重写来自响应或者请求的 cookie 的一部分。在进一步的实施例中，设备 200 确定将和客户机 102 或者服务器 106 关联的 cookie 插入到客户机 102 和服务器 106 之间的传输中。在又一个实施例中，设备 200 确定从客户机 102 和服务器 106 之间的信息移除 cookie。在进一步的实施例中，设备确定终止用于在客户机 102 和服务器 106 之间的传输的 cookie。还可以通过可包括或者指向用于制定或实现这些和类似决定的任意数量个步骤或者指令的策略来指定这些和类似的动作。

[0276] 在步骤 520，中间设备可以实现通过访问配置文件的策略确定的任一动作。在一些实施例中，中间设备代理一个或者多个 Cookie 或者忽略代理一个或者多个 Cookie 并且将响应转发给客户机 102。在进一步的实施例中，中间设备代理一个或者多个 Cookie 或者忽略代理一个或者多个 Cookie 并且将响应转发给服务器 106。中间设备可以按照配置所识别或者匹配的策略的一个或者多个动作所指示或者指定的方式来代理 cookie。在一些实施例中，中间设备 200 产生 cookie。在其他实施例中，中间设备 200 终止 cookie。在一些实施例中，中间设备转发该响应或者请求，而不会对响应或者请求中的一个或多个 cookie 采取任何动作。在进一步的实施例中，中间设备按照由所识别的访问配置文件或由所识别的策略的动作所指示或者识别的方式来修改来自该响应或者请求的 cookie。中间设备 200 在所采用或者执行的动作之后可以将该请求或者响应转发给目的地。在一些实施例中，中间设备 200 将该请求转发给服务器 106。在其他实施例中，中间设备 200 将该响应转发给客户机 102。

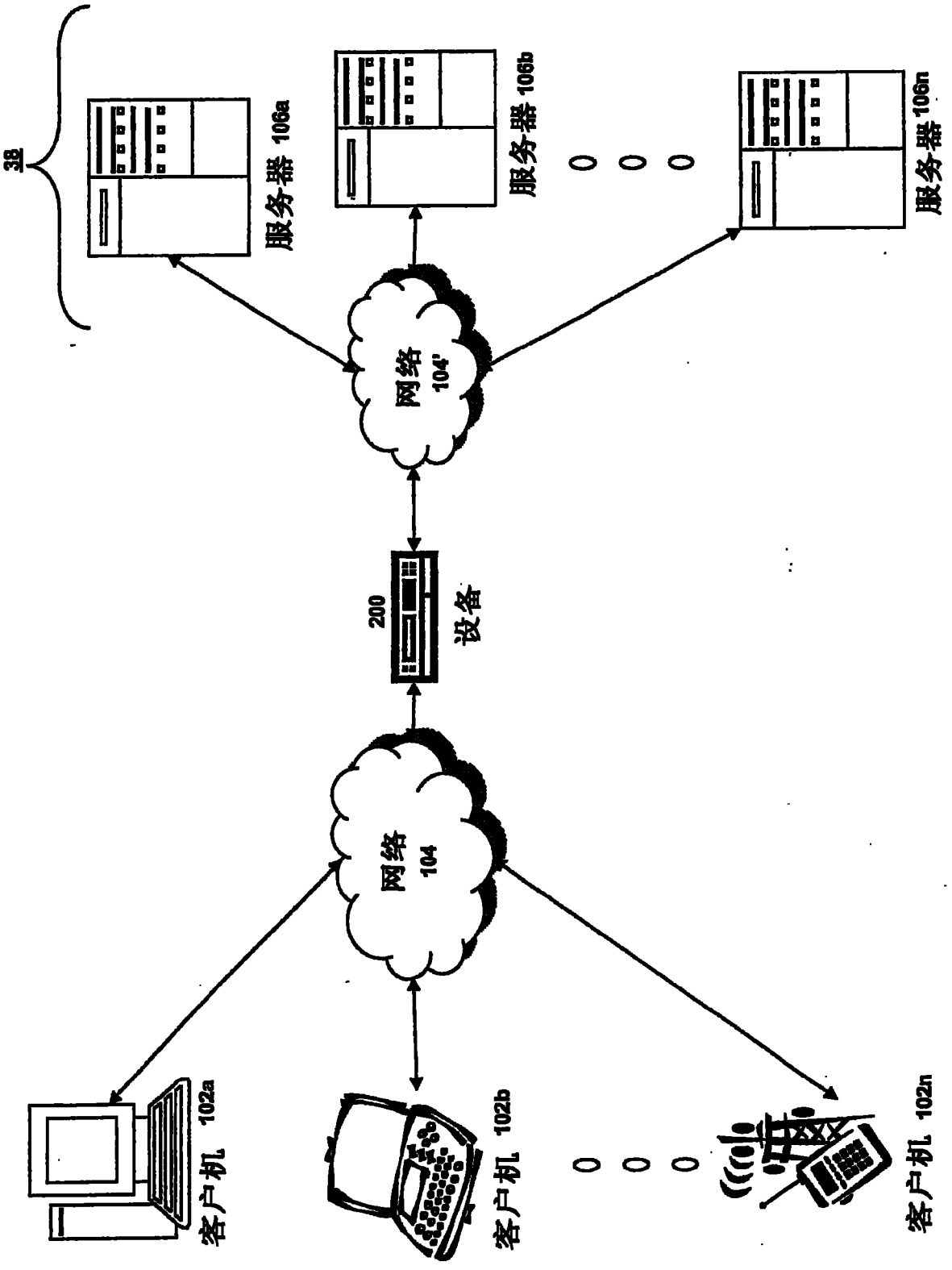

38
服务器 106a
服务器 106b
服务器 106n
网络 104'
200
设备
网络 104
客户机 102a
客户机 102b
客户机 102n

图 1A

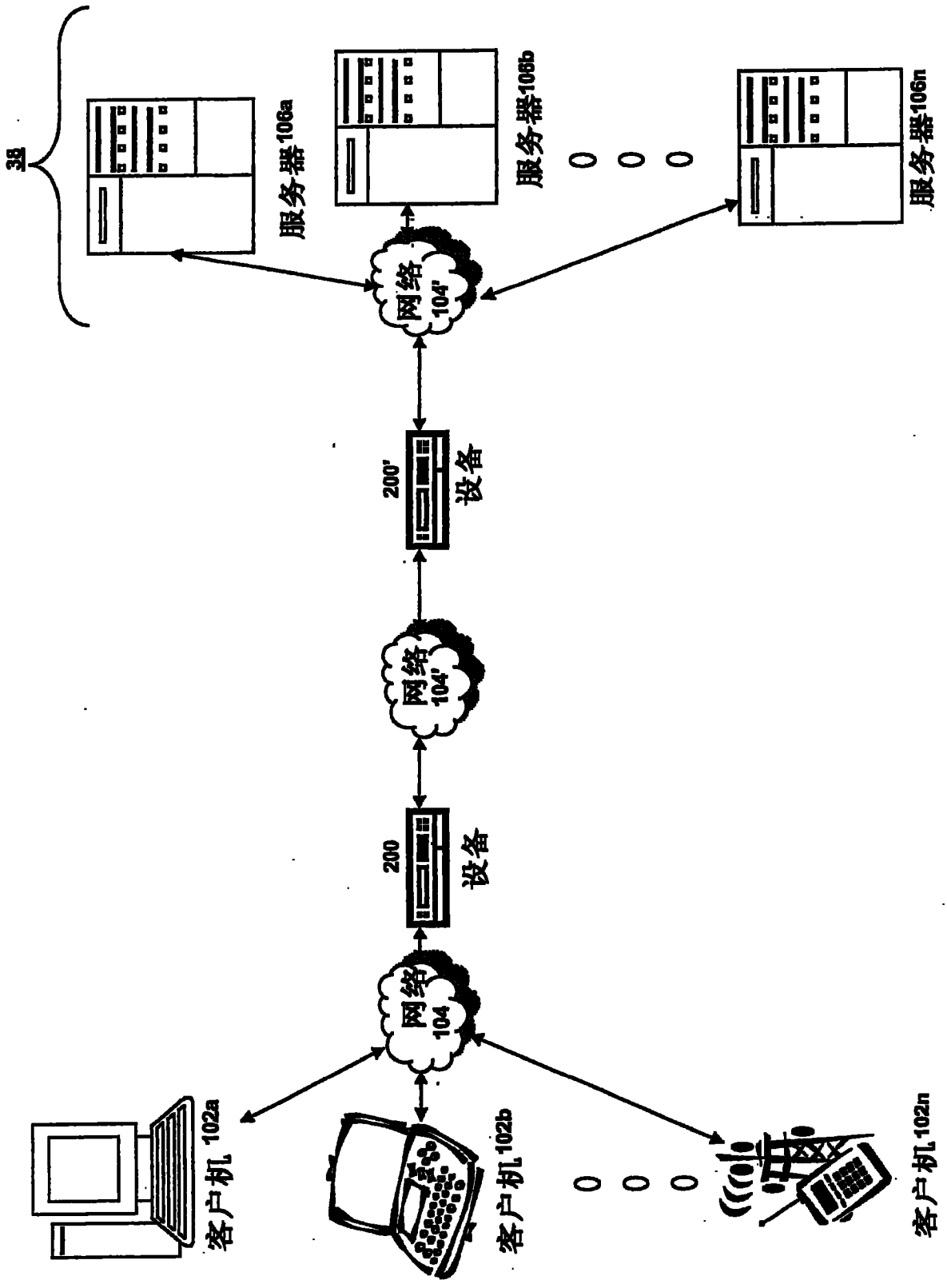

38
服务器106a
服务器106b
服务器106n
网络104'
200'
设备
网络104'
200
设备
网络104
客户机102a
客户机102b
客户机102n

图 1B

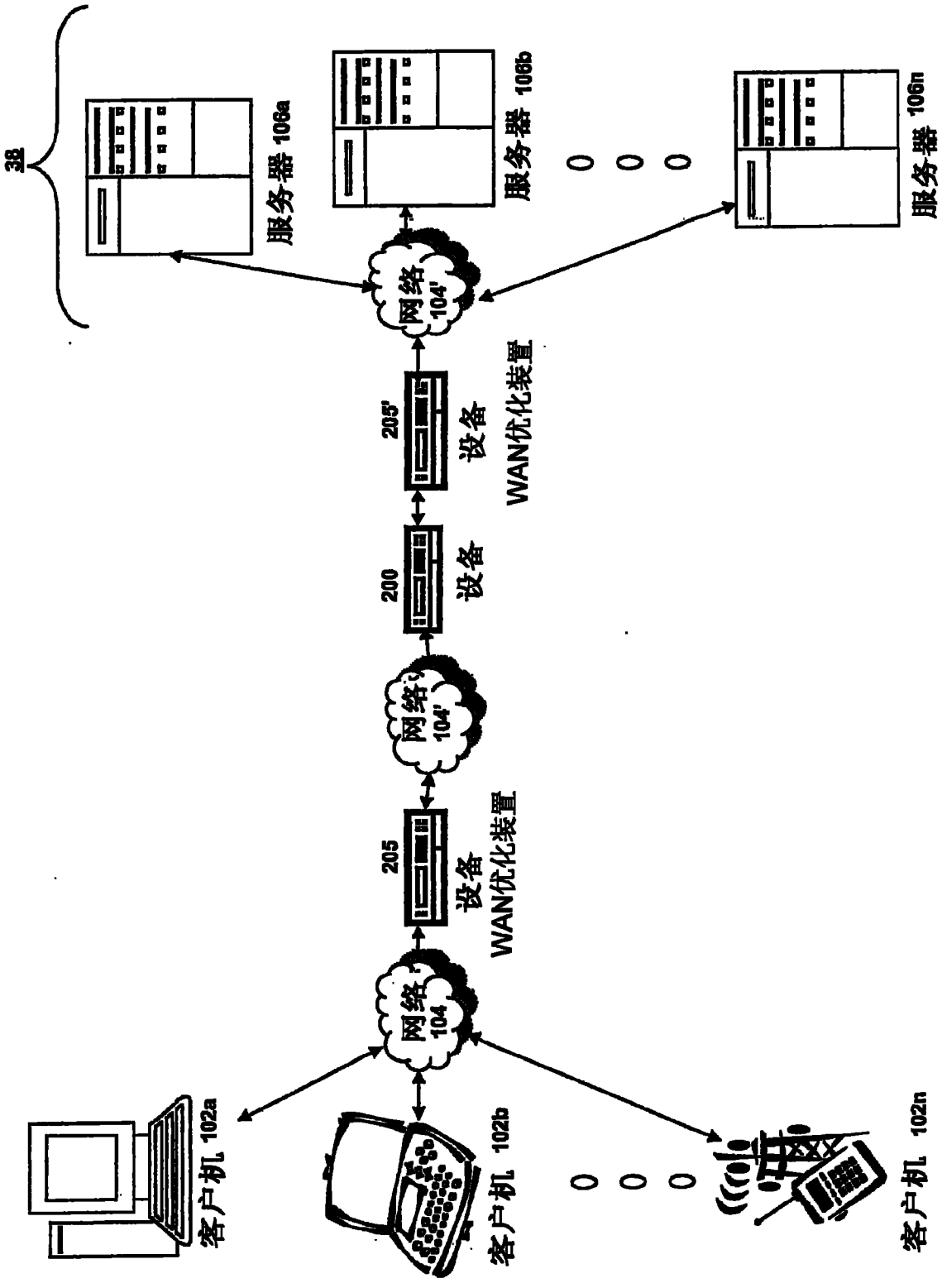

38
服务器 106a
服务器 106b
服务器 106n
网络 104'
205' 设备 WAN优化装置
200 设备
网络 104'
205 设备 WAN优化装置
网络 104
客户机 102a
客户机 102b
客户机 102n

图 1C

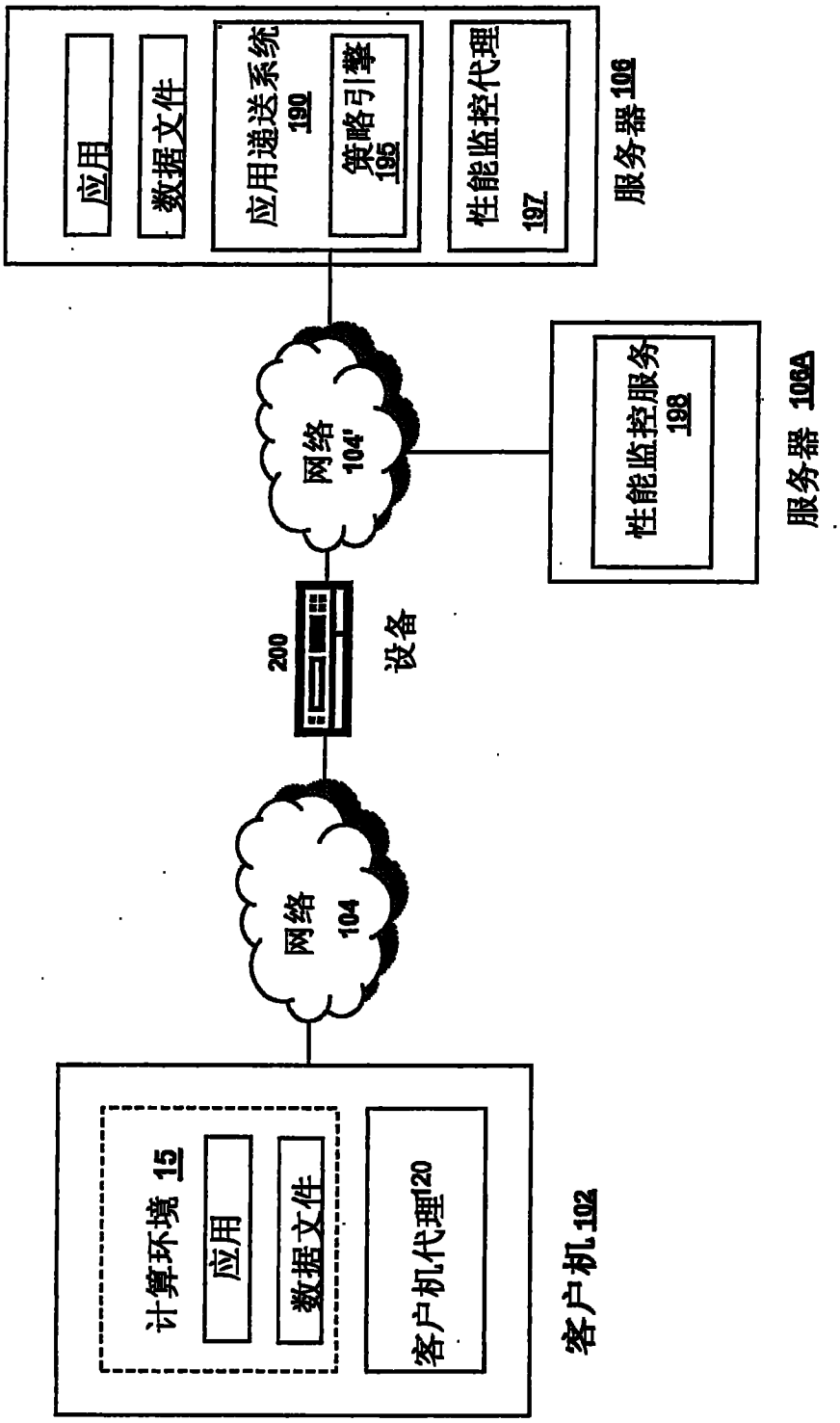

应用
数据文件
应用递送系统
190
策略引擎
195
性能监控代理
197
服务器 106
网络
104'
性能监控服务
198
服务器 106A
200
设备
网络
104
计算环境 15
应用
数据文件
客户机代理120
客户机 102

图 1D

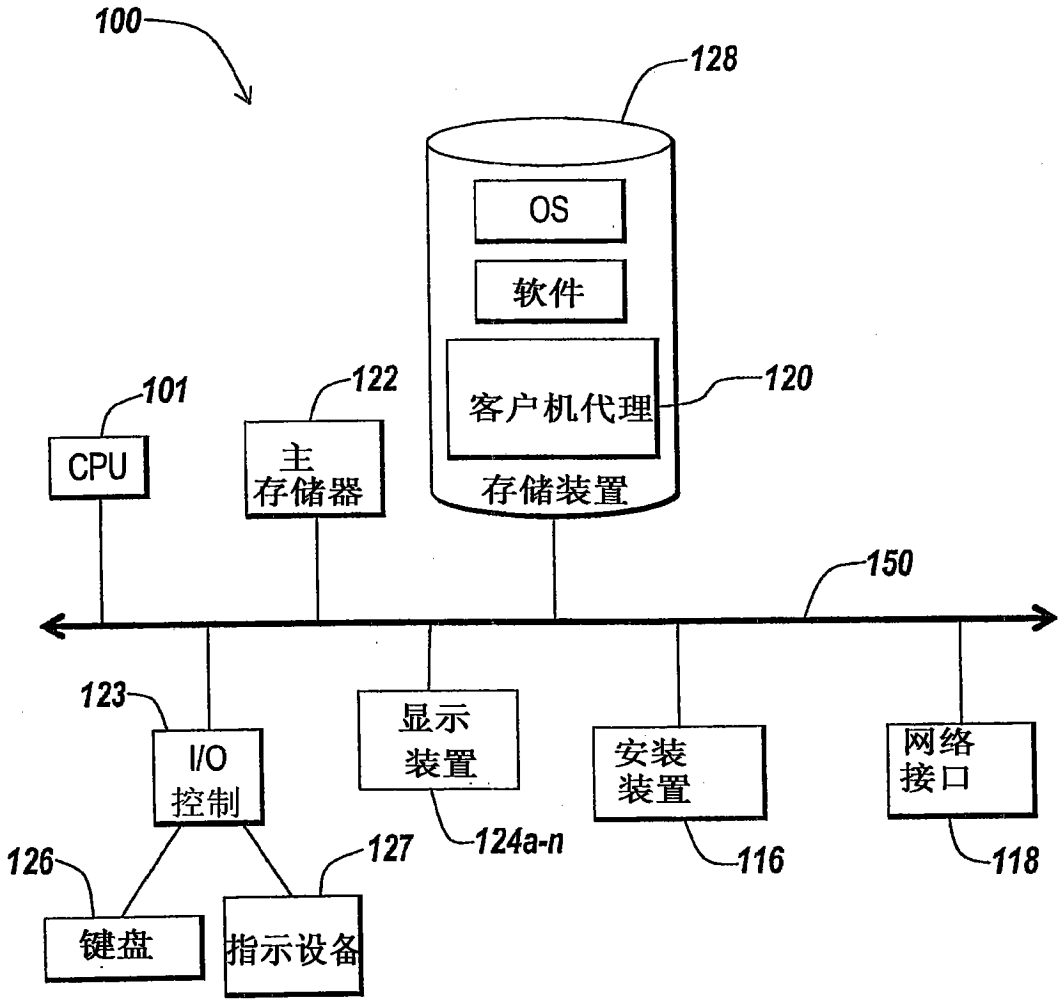

100
128
OS
软件
客户机代理
120
存储装置
101
CPU
122
主
存储器
150
123
I/O
控制
显示
装置
124a-n
安装
装置
116
网络
接口
118
126
键盘
127
指示设备

图 1E

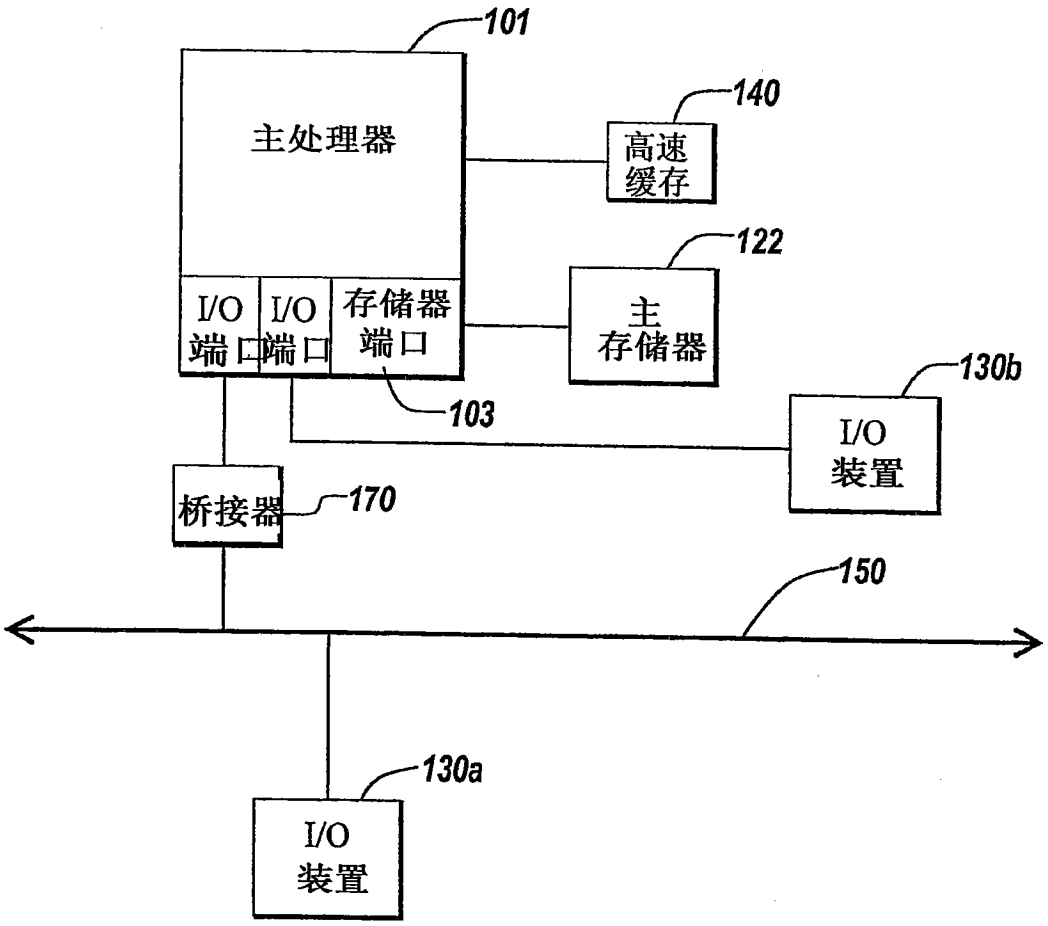

101
主处理器
140
高速缓存
122
主存储器
I/O端口
I/O端口
存储器端口
103
130b
I/O装置
桥接器
170
150
130a
I/O装置

图 1F

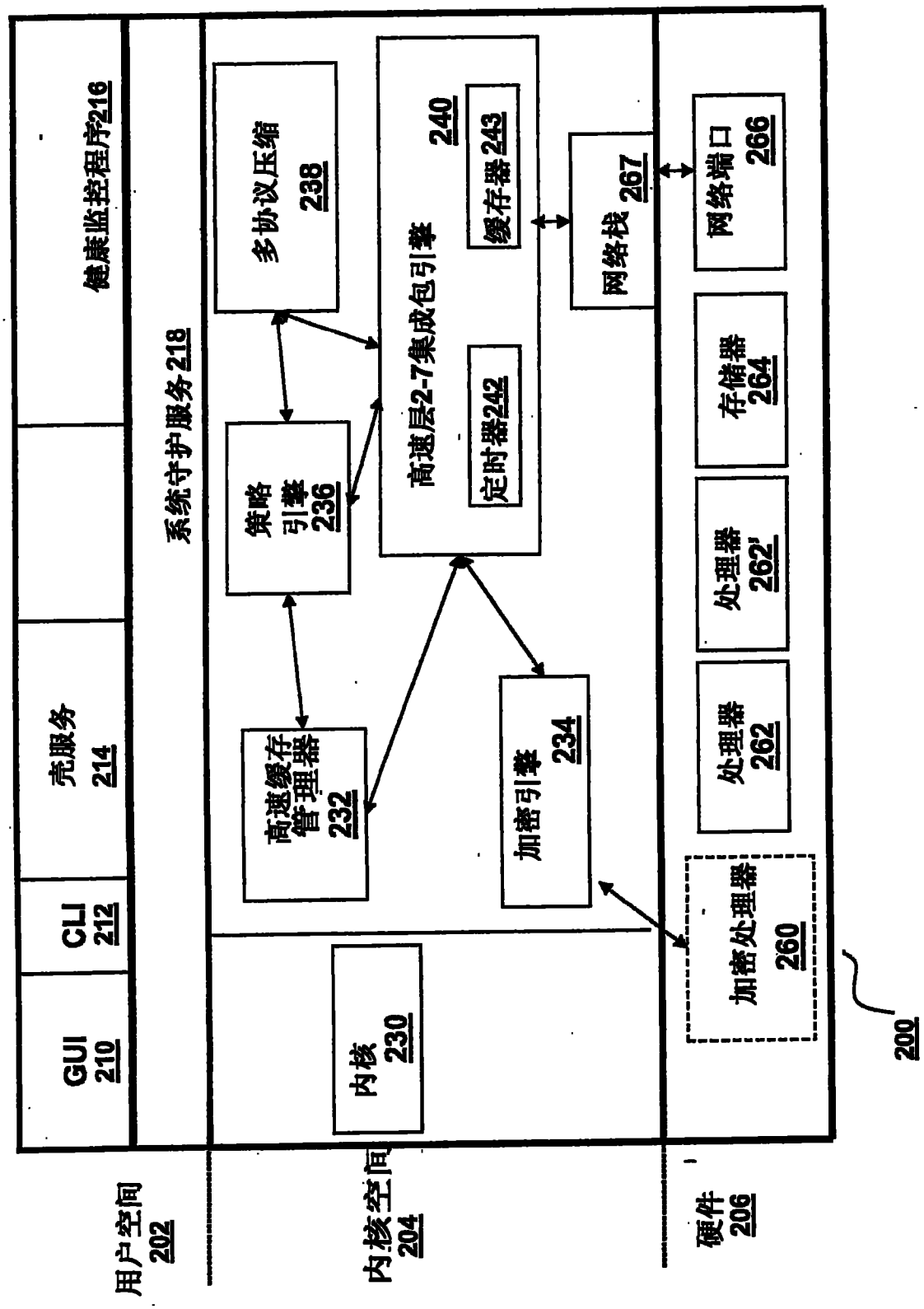

健康监控程序216
系统守护服务218
GUI 210
CLI 212
壳服务 214
多协议压缩 238
策略引擎 236
高速缓存管理器 232
内核 230
高速层2-7集成包引擎 240
缓存器243
定时器242
加密引擎 234
网络栈 267
网络端口 266
存储器 264
处理器 262'
处理器 262
加密处理器 260
200
用户空间 202
内核空间 204
硬件 206

图 2A

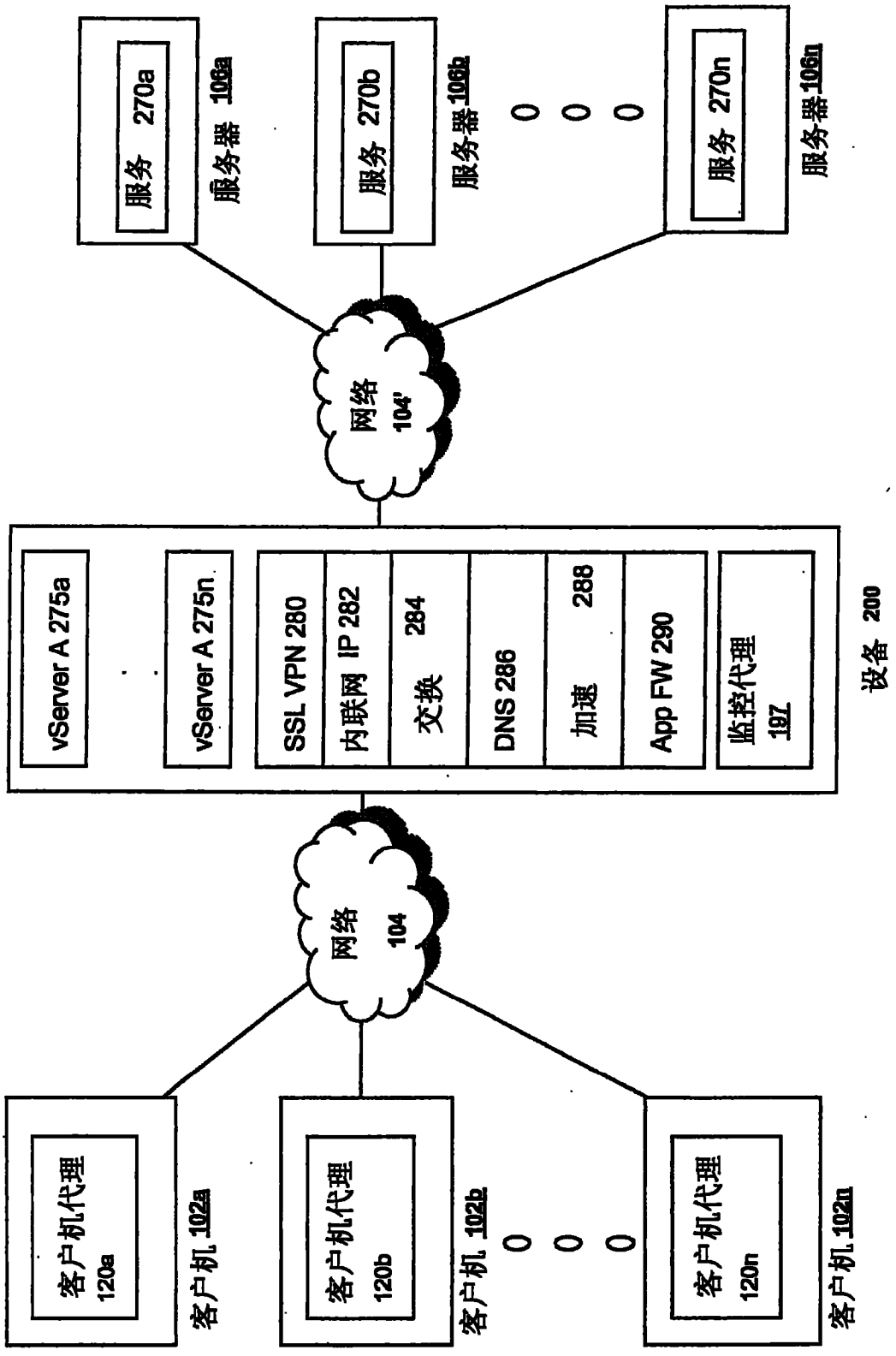

服务 270a
服务器 106a
服务 270b
服务器 106b
服务 270n
服务器 106n
网络 104'
vServer A 275a
vServer A 275n
SSL VPN 280
内联网 IP 282
交换 284
DNS 286
加速 288
App FW 290
监控代理 197
设备 200
网络 104
客户机代理 120a
客户机 102a
客户机代理 120b
客户机 102b
客户机代理 120n
客户机 102n

图 2B

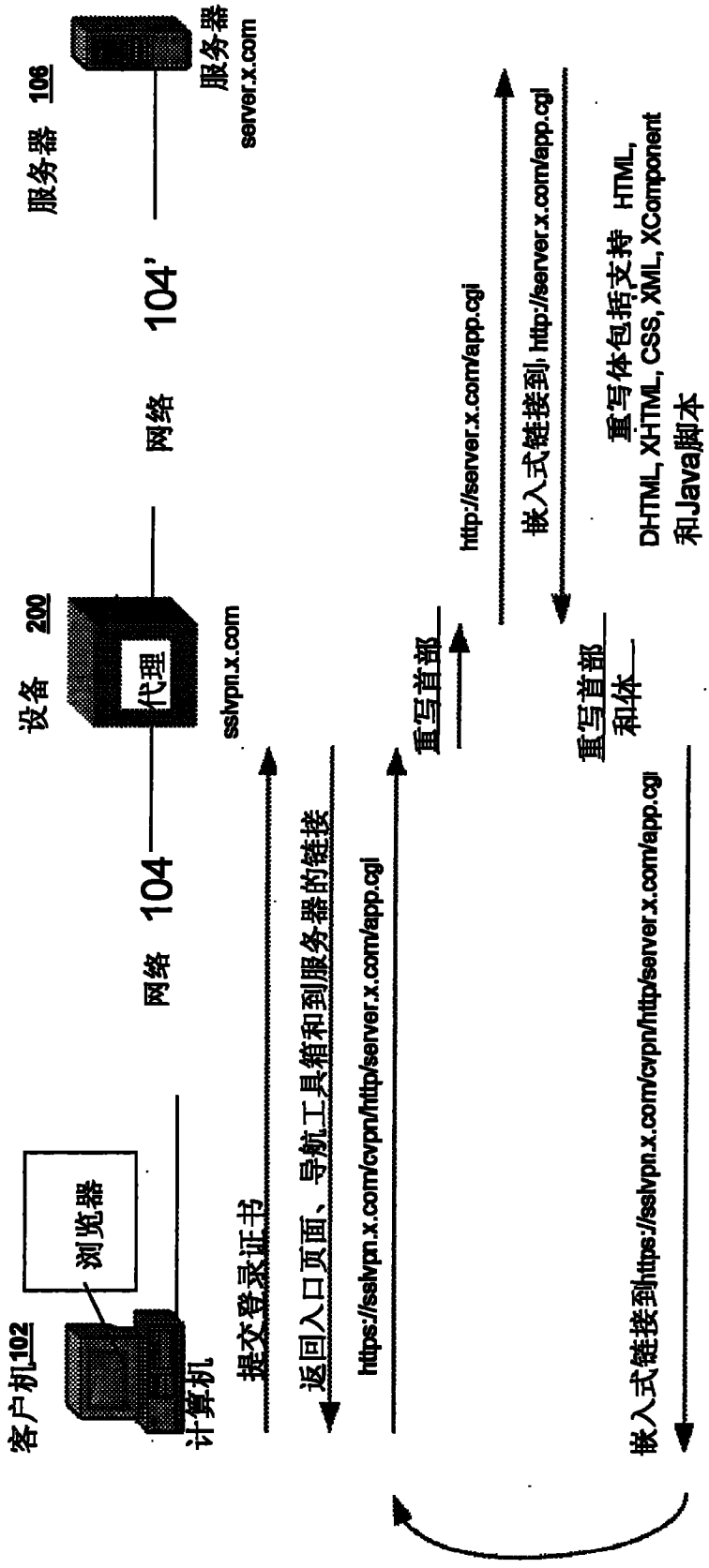

客户机102
浏览器
计算机
网络 104
设备 200
代理
sslvpn.x.com
网络 104'
服务器 106
服务器
server.x.com
提交登录证书
返回入口页面、导航工具箱和到服务器的链接
https://sslvpn.x.com/cvpn/http/server.x.com/app.cgi
重写首部
http://server.x.com/app.cgi
嵌入式链接到 http://server.x.com/app.cgi
重写首部
和体
重写体包括支持 HTML,
DHTML, XHTML, CSS, XML, XComponent
和Java脚本
嵌入式链接到https://sslvpn.x.com/cvpn/http/server.x.com/app.cgi

图 3A

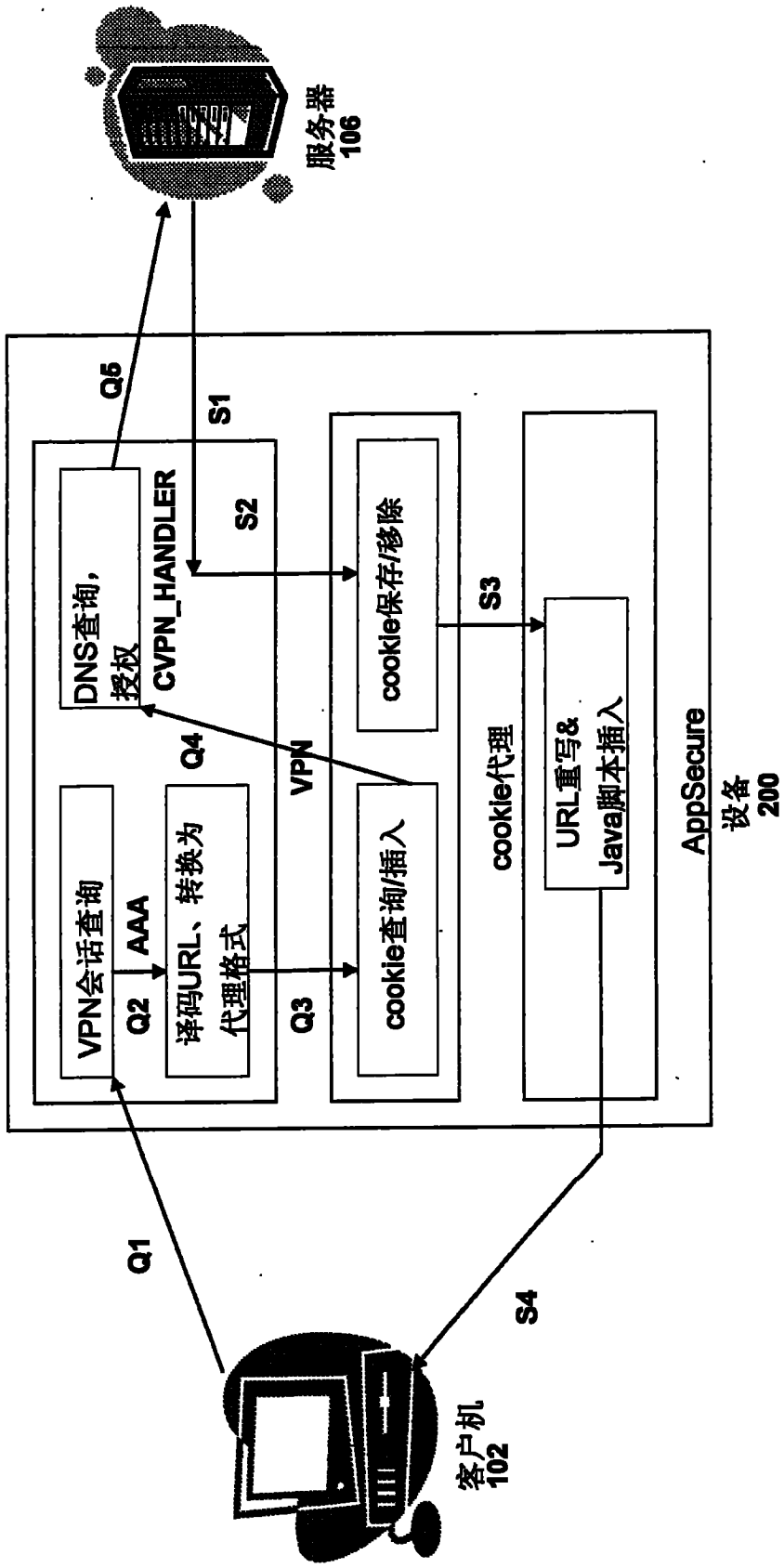

服务器
106
Q5
S1
S2
DNS查询，
授权
CVPN_HANDLER
cookie保存/移除
S3
Q4
VPN
VPN会话查询
AAA
Q2
译码URL、转换为
代理格式
Q3
cookie查询/插入
cookie代理
URL重写&
Java脚本插入
AppSecure
设备
200
Q1
S4
客户机
102

图 3B

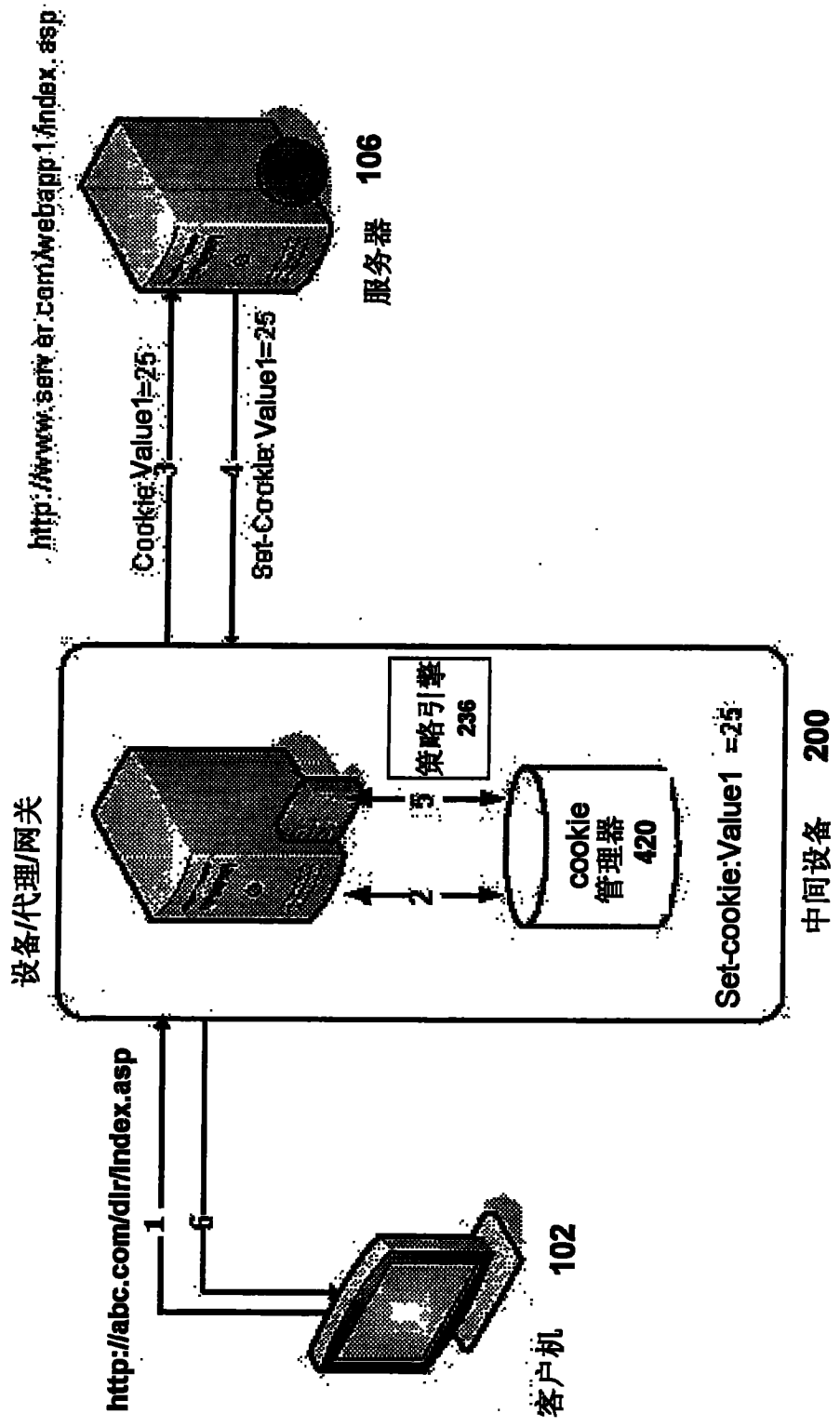

http://www.server.com/webapp1/index.asp
服务器 106
Cookie:Value1=25
3
4
Set-Cookie:Value1=25
设备/代理/网关
策略引擎 236
5
2
cookie 管理器 420
Set-cookie:Value1 =25
中间设备 200
http://abc.com/dir/index.asp
1
6
客户机 102

图 4A

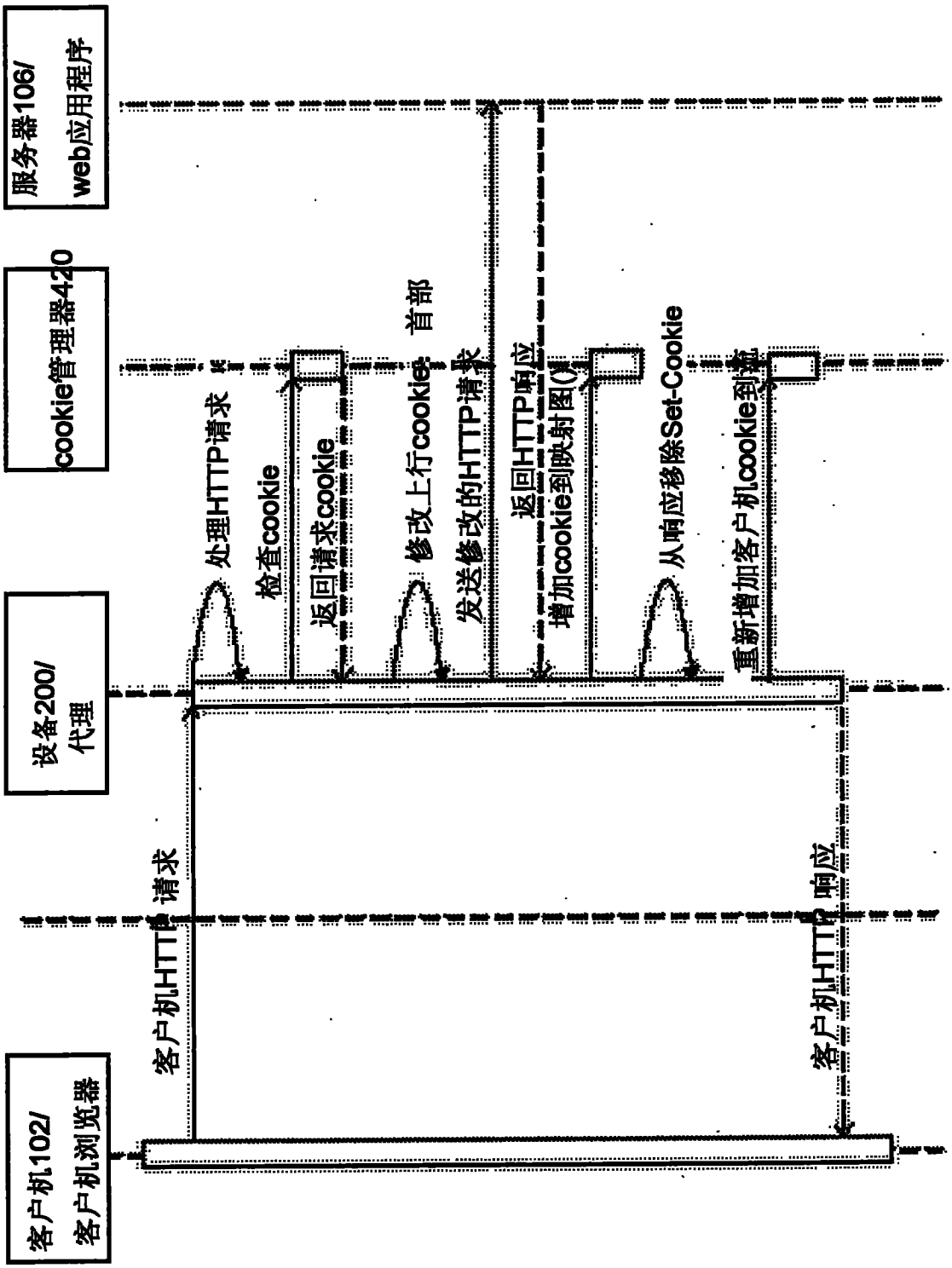

客户机102/
客户机浏览器
设备200/
代理
cookie管理器420
服务器106/
web应用程序
客户机HTTP请求
处理HTTP请求
检查cookie
返回请求cookie
修改上行cookie
首部
发送修改的HTTP请求
返回HTTP响应
增加cookie到映射图(
从响应移除Set-Cookie
重新增加客户机cookie到
客户机HTTP响应

图 4B

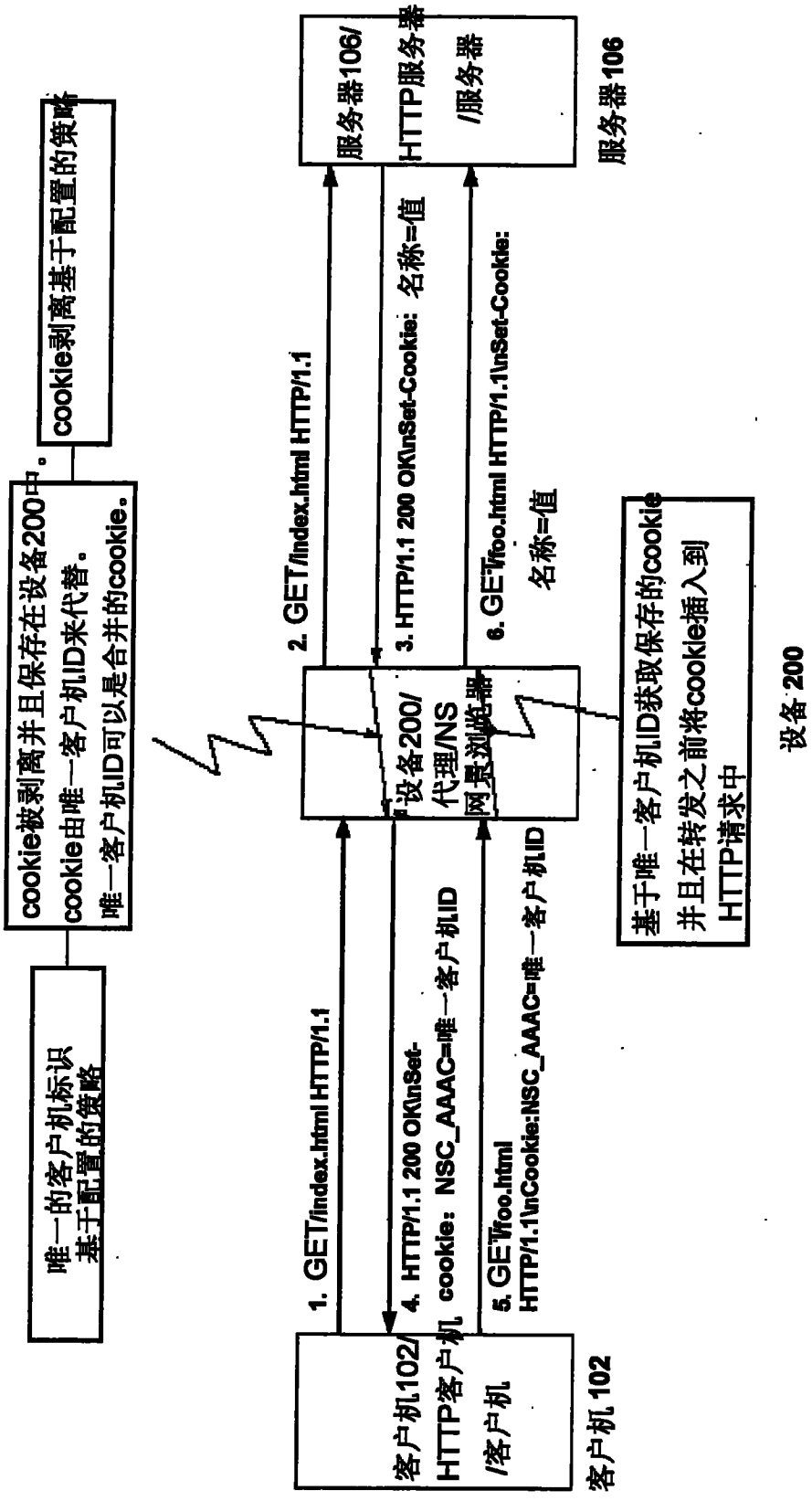

唯一的客户机标识
基于配置的策略
cookie被剥离并且保存在设备200中。
cookie由唯一客户机ID来代替。
唯一客户机ID可以是合并的cookie。
cookie剥离基于配置的策略
1. GET/index.html HTTP/1.1
2. GET/index.html HTTP/1.1
3. HTTP/1.1 200 OK\nSet-Cookie: 名称=值
4. HTTP/1.1 200 OK\nSet-Cookie: NSC_AAAC=唯一客户机ID
5. GET/foo.html HTTP/1.1\nCookie:NSC_AAAC=唯一客户机ID
6. GET/foo.html HTTP/1.1\nSet-Cookie:
名称=值
基于唯一客户机ID获取保存的cookie
并且在转发之前将cookie插入到
HTTP请求中
客户机102/
HTTP客户机
/客户机
客户机 102
设备200/
代理/NS
网景浏览器
设备 200
服务器106/
HTTP服务器
/服务器
服务器106

图 4C

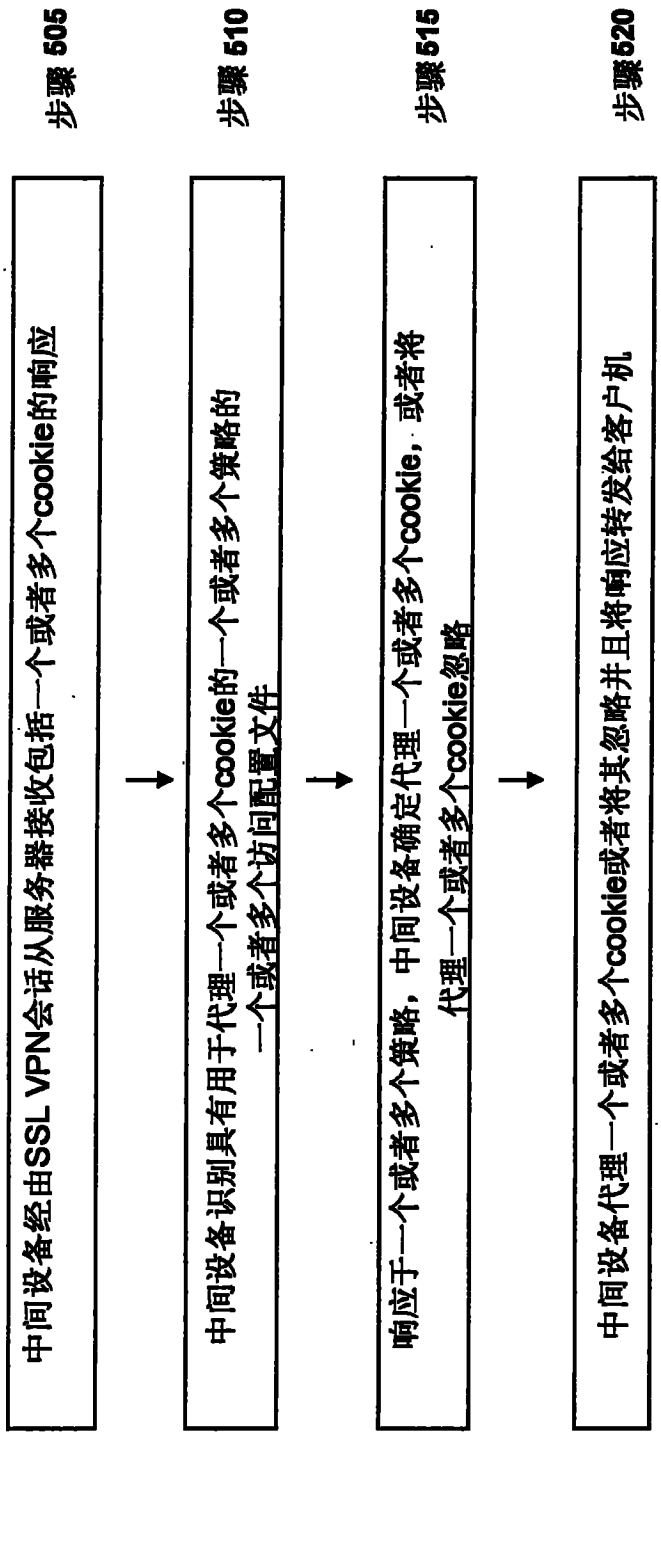

步骤 505
中间设备经由SSL VPN会话从服务器接收包括一个或者多个cookie的响应
步骤 510
中间设备识别具有用于代理一个或者多个cookie的一个或者多个策略的一个或者多个访问配置文件
步骤 515
响应于一个或者多个策略，中间设备确定代理一个或者多个cookie，或者将代理一个或者多个cookie忽略
步骤 520
中间设备代理一个或者多个cookie或者将其忽略并且将响应转发给客户机
500

图 5