(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 103384989 A
(43)申请公布日 2013.11.06

(21)申请号 201180068455.1
(22)申请日 2011.12.27
(30)优先权数据
61/427692 2010.12.28 US
(85)PCT申请进入国家阶段日
2013.08.23
(86)PCT申请的申请数据
PCT/US2011/067372 2011.12.27
(87)PCT申请的公布数据
WO2012/092263 EN 2012.07.05
(71)申请人 思杰系统有限公司
地址 美国佛罗里达州
(72)发明人 D·格尔
(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 12/803(2013.01)
H04L 12/813(2013.01)
H04L 12/721(2013.01)

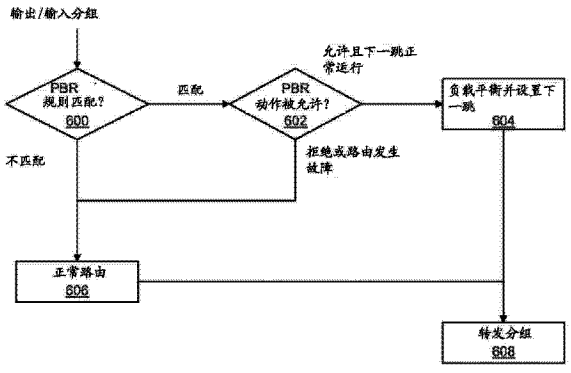
权利要求书2页 说明书48页 附图22页

(54)发明名称

用于对多个下一跳进行策略路由的系统和方法

(57)摘要

本申请涉及策略路由，用于经由多个下一跳进行智能流量管理。在一些实施例中，本文公开的系统和方法可以提供对跨越多个网络链路的输入和输出流量的管理，并且还可在链路故障情况下提供可靠性，以及响应于各种应用的延迟和带宽需求，提供对流量的平衡。相应地，该系统和方法可提供智能策略路由及网络和端口地址转换，其对应用流量类型、协议、源IP地址和端口、目的IP地址和端口、或上述的任一组合敏感，以及可以根据多种流量特征在多条可用路径之间平衡流量负载。可以在逐个分组、逐个事务或逐个会话基础上进行该路由，并且该系统和方法可以包括用于对可用网络路径进行应用觉察的健康监控的能力。



CN 103384989 A

1. 一种用于对多个下一跳进行策略路由的方法，所述方法包括：

(a) 由在多个装置与多个下一跳中间的装置将分组的一个或多个特征与策略的一个或多个参数相匹配，所述策略指定对所述多个下一跳中的一个下一跳的路由动作是否被允许；

(b) 由所述装置响应于所述匹配来确定所述路由动作被允许并且所述多个下一跳中的一个或多个下一跳是正常运行的；

(c) 由所述装置响应于负载平衡决策从所述多个下一跳中选择由所述策略识别的下一跳；以及

(d) 由所述装置将所述分组路由到所选择的下一跳。

2. 根据权利要求 1 所述的方法，其中，所述步骤(a)还包括将所述分组的一个或多个特征与所述策略的一个或多个参数中的相应参数相匹配，所述一个或多个特征包括下列中的一个或多个：源互联网协议地址、目的互联网协议地址、源端口、目的端口、协议、虚拟局域网(VLAN)和源机器访问控制(MAC)。

3. 根据权利要求 1 所述的方法，其中，所述步骤(a)还包括由过滤器将所述策略的一个或多个参数与包括所述装置的输入分组或输出分组的其中一个的分组相匹配。

4. 根据权利要求 1 所述的方法，其中，所述步骤(b)还包括由所述装置基于对所述多个下一跳中每一个的健康的监控来确定一个或多个下一跳是正常运行的。

5. 根据权利要求 1 所述的方法，其中，所述步骤(c)还包括由所述装置基于所述分组的协议类型来执行一种类型的负载平衡。

6. 根据权利要求 1 所述的方法，其中，所述步骤(c)还包括由所述装置执行基于请求的负载平衡以从所述多个下一跳中选择所述下一跳。

7. 根据权利要求 1 所述的方法，其中，所述步骤(c)还包括由所述装置对于每个新的传输层连接执行基于连接的负载平衡以从所述多个下一跳中选择所述下一跳。

8. 根据权利要求 1 所述的方法，其中，所述步骤(c)还包括由所述装置执行基于时间的负载平衡以从所述多个下一跳中选择所述下一跳。

9. 根据权利要求 1 所述的方法，其中，所述步骤(c)还包括由所述装置基于由目的互联网协议地址或源互联网协议地址的其中一个所确定的持久性来选择所述下一跳。

10. 根据权利要求 1 所述的方法，其中，所述步骤(d)还包括由所述装置基于所选择的下一跳来转换所述分组的互联网协议地址。

11. 一种用于对多个下一跳进行策略路由的系统，所述系统包括：

在多个装置与多个下一跳中间的装置；

所述装置的过滤器，用于将分组的一个或多个特征与策略的一个或多个参数相匹配，所述策略指定对所述多个下一跳中的一个下一跳的路由动作是否被允许；

所述装置的策略引擎，用于响应于所述匹配来确定所述路由动作被允许并且所述多个下一跳中的一个或多个下一跳是正常运行的；

所述装置的负载平衡器，用于经由负载平衡决策从所述多个下一跳中选择由所述策略识别的下一跳；并且

其中，所述装置将所述分组路由到所选择的下一跳。

12. 根据权利要求 11 所述的系统，其中，所述一个或多个特征包括下列中的一个或多

个：源互联网协议地址、目的互联网协议地址、源端口、目的端口、协议、虚拟局域网（VLAN）和源机器访问控制（MAC）。

13. 根据权利要求 11 所述的系统，其中，所述分组包括所述装置的输入分组或输出分组的其中一个。

14. 根据权利要求 11 所述的系统，还包括监控器，用于监控所述多个下一跳中每一个的健康。

15. 根据权利要求 11 所述的系统，其中，所述负载平衡器基于所述分组的协议类型来执行一种类型的负载平衡。

16. 根据权利要求 11 所述的系统，其中，所述负载平衡器执行基于请求的负载平衡来从所述多个下一跳中选择所述下一跳。

17. 根据权利要求 11 所述的系统，其中，所述负载平衡器对于每个新的传输层连接执行基于连接的负载平衡，以从所述多个下一跳中选择所述下一跳。

18. 根据权利要求 11 所述的系统，其中，所述负载平衡器执行基于时间的负载平衡来从所述多个下一跳中选择所述下一跳。

19. 根据权利要求 11 所述的系统，其中，所述负载平衡器基于由所述分组的目的互联网协议地址或源互联网协议地址的其中一个所确定的持久性来选择所述下一跳。

20. 根据权利要求 11 所述的系统，其中，所述装置基于所选择的下一跳来转换所述分组的互联网协议地址。

用于对多个下一跳进行策略路由的系统和方法

[0001] 相关申请

[0002] 本申请要求在2010年12月28日提交的、名称为“Systems And Methods For Policy Based Routing for Multiple Next Hops”、序列号为61/427692的美国临时申请的权益和优先权，通过引用将该美国临时申请全部包含于此。

技术领域

[0003] 本申请总的涉及数据通信网络。本申请尤其涉及用于对多个下一跳进行策略路由(policy based routing)的系统和方法。

背景技术

[0004] 为了提高网络的可靠性，许多组织通过使用多个网络链路来增加冗余。例如，一个组织可以具有到第一互联网服务提供者的链路和到第二互联网服务提供者的备份链路。如果其中一个提供者出现故障，另一个链路可以作为备份用来给该组织提供完整的服务。此外，当这两个链路都有效时，它们可以被用于平衡流量，以提高该组织的网络吞吐量。

[0005] 传统的通过多个网络链路路由流量的解决方案主要是简单的基于目的地址的转发。例如，在10.0.0.0和19.255.255.255之间目的互联网协议(IP)地址可以去往第一链路，而在20.0.0.0和29.255.255.255之间的目的IP地址可以去往第二链路。这些解决方案通常要求手工设置和频繁调整，以管理链路之间的平衡。而且，由于其仅关注目的IP地址，所以在一个IP收到比另一个IP更多的流量的情况下可造成不平衡的负载。例如，在第一目的IP地址10.0.0.0的第一服务器可能是邮件服务器并且仅处理间断的、时间不敏感的、短期突发的数据。在第二目的IP地址20.0.0.0的第二服务器可能是视频会议服务器，并且要求低延迟、高带宽的网络吞吐量。仅使用基于目的地址的转发，对从组织到这些服务器的多个链路的使用可能是不平衡的。此外，仅使用基于目的地址的转发可能不能管理返回的或输入的流量。

发明内容

[0006] 本申请涉及策略路由，用于经由多个下一跳进行智能流量管理。在一些实施例中，本文公开的系统和方法可以提供对跨越多个网络链路的输入和输出流量的管理。这些系统和方法还提供在链路故障情况下的可靠性，以及响应于各种应用的延迟和带宽需求，提供对流量的平衡。相应地，这些系统和方法可提供智能策略路由及网络和端口地址转换，其对应用流量类型或协议、源IP地址和端口、目的IP地址和端口、或上述的任一组合敏感，以及可以根据多种流量特征在多条可用路径之间平衡流量负载。在一些实施例中，可以逐个分组、逐个事务或逐个会话基础上进行该路由。在其他实施例中，该系统和方法可以包括用于对可用网络路径进行应用觉察的健康监控的能力。例如，流量管理系统可确定链路上的延迟对于第一应用太高，但对于第二应用是足够的，以及可以相应地重新平衡和路由流量。

[0007] 在一些方面，本解决方案涉及用于对多个下一跳进行策略路由的方法。该方法包

括由在多个装置与多个下一跳中间的装置将分组的一个或多个特征与策略的一个或多个参数相匹配。该策略指定对所述多个下一跳中的一个下一跳的路由动作是否被允许。该方法还包括由所述装置响应于所述匹配来确定所述路由动作被允许并且所述多个下一跳中的一个或多个下一跳是正常运行的；以及，由所述装置响应于负载平衡决策从所述多个下一跳中选择由所述策略识别的下一跳。所述装置将所述分组路由到所选择的下一跳。

[0008] 在一些实施例中，该方法包括将所述分组的一个或多个特征与所述策略的一个或多个参数中的相应参数相匹配。所述一个或多个特征可以包括下列中的一个或多个：源互联网协议地址、目的互联网协议地址、源端口、目的端口、协议、虚拟局域网(VLAN)和源机器访问控制(MAC)。在一些实施例中，该方法包括由过滤器将所述策略的一个或多个参数与包括所述装置的输入分组或输出分组的其中一个的分组相匹配。在一些实施例中，该方法包括由所述装置基于对所述多个下一跳中每一个的健康的监控确定一个或多个下一跳是正常运行的。

[0009] 在一些实施例中，该方法包括由所述装置基于所述分组的协议类型来执行一种类型的负载平衡。在一些实施例中，该方法包括由所述装置执行基于请求的负载平衡来从多个下一跳中选择下一跳。在一些实施例中，该方法包括由所述装置对于每个新的传输层连接执行基于连接的负载平衡，以从多个下一跳中选择下一跳。在一些实施例中，该方法包括由所述装置执行基于时间的负载平衡来从多个下一跳中选择下一跳。在一些实施例中，该方法包括由所述装置基于由目的互联网协议地址或源互联网协议地址的其中一个所确定的持久性来选择所述下一跳。在一些实施例中，该方法包括由所述装置基于所选择的下一跳来转换分组的互联网协议地址。

[0010] 在一些方面，本解决方案涉及用于对多个下一跳进行策略路由的系统。该系统包括在多个装置与多个下一跳中间的装置。该装置可包括过滤器，用于将分组的一个或多个特征与策略的一个或多个参数相匹配，该策略指定对所述多个下一跳中的一个下一跳的路由动作是否被允许。该装置还可包括策略引擎，用于响应于所述匹配来确定所述路由动作是被允许的并且所述多个下一跳中的一个或多个下一跳是正常运行的。该装置可包括负载平衡器，用于通过负载平衡决策从所述多个下一跳中选择由所述策略识别的下一跳。该装置将所述分组路由到所选择的下一跳。

[0011] 在一些实施例中，所述一个或多个特征包括下列中的一个或多个：源互联网协议地址、目的互联网协议地址、源端口、目的端口、协议、虚拟局域网(VLAN)和源机器访问控制(MAC)。在一些实施例中，所述分组包括所述装置的输入分组或输出分组的其中一个。在一些实施例中，该系统包括监控器，以监控所述多个下一跳中每一个的健康。

[0012] 在一些实施例中，负载平衡器基于分组的协议类型来执行一种类型的负载平衡。在一些实施例中，负载平衡器执行基于请求的负载平衡来从多个下一跳中选择下一跳。在一些实施例中，负载平衡器对于每个新的传输层连接执行基于连接的负载平衡，以从多个下一跳中选择下一跳。在一些实施例中，负载平衡器执行基于时间的负载平衡来从多个下一跳中选择下一跳。在一些实施例中，负载平衡器基于由分组的目的互联网协议地址或源互联网协议地址的其中一个所确定的持久性来选择下一跳。在一些实施例中，该装置基于所选择的下一跳来转换分组的互联网协议地址。

[0013] 在附图和下文描述中对本发明各种实施例的细节进行详细阐述。

附图说明

[0014] 通过参考下述结合附图的描述，本发明的前述和其它目的、方面、特征和优点将会更加明显并更易于理解，其中：

[0015] 图 1A 是客户机经由设备访问服务器的网络环境的实施例的框图；

[0016] 图 1B 是经由设备从服务器传送计算环境到客户机的环境的实施例的框图；

[0017] 图 1C 是经由设备从服务器传送计算环境到客户机的环境的又一个实施例的框图；

[0018] 图 1D 是经由设备从服务器传送计算环境到客户机的环境的又一个实施例的框图；

[0019] 图 1E 到 1H 是计算装置的实施例的框图；

[0020] 图 2A 是用于处理客户机和服务器之间的通信的设备的实施例的框图；

[0021] 图 2B 是用于优化、加速、负载平衡和路由客户机和服务器之间的通信的设备的又一个实施例的框图；

[0022] 图 3 是用于经由设备与服务器通信的客户机的实施例的框图；

[0023] 图 4A 是虚拟化环境的实施例的框图；

[0024] 图 4B 是虚拟化环境的又一个实施例的框图；

[0025] 图 4C 是虚拟设备的实施例的框图；

[0026] 图 5A 是在多核系统中实现并行机制的方式的实施例的框图；

[0027] 图 5B 是使用多核系统的系统实施例的框图；

[0028] 图 5C 是多核系统方面的另一实施例的框图；

[0029] 图 6A 是用于智能策略路由的系统的实施例的框图；

[0030] 图 6B 是智能策略路由的方法的实施例的流程图；

[0031] 图 7A 是使用基于端口的路由策略的系统的实施例的框图；

[0032] 图 7B 是在链路故障期间，使用基于端口的路由策略的系统的实施例的框图；

[0033] 图 7C 是使用智能的基于目的地的路由策略的系统的实施例的框图；

[0034] 图 7D 是在链路故障期间，使用智能的基于目的地的路由策略的系统的实施例的框图；

[0035] 图 7E 是使用到目的地的具有不同网络特征的多条路径的系统的实施例的框图。

[0036] 根据在下文结合附图详细阐述的更细节的描述，本发明的特征和优势将会更明显，其中同样的附图标记自始至终标识对应的元素。在附图中，同样的附图标记通常指示相同的、功能上相似的和 / 或结构上相似的元素。

具体实施方式

[0037] 为了阅读下文各种实施例的描述，下述对于说明书的部分以及它们各自内容的描述是有用的：

[0038] -A 部分描述可用于实施本文描述的实施例的网络环境和计算环境；

[0039] -B 部分描述用于将计算环境传送到远程用户的系统和方法的实施例；

[0040] -C 部分描述用于加速客户机和服务器之间的通信的系统和方法的实施例；

[0041] -D 部分描述用于对应用传送控制器进行虚拟化的系统和方法的实施例；

[0042] -E 部分描述用于提供多核架构和环境的系统和方法的实施例；

[0043] -F 部分描述用于对多个下一跳进行策略路由的系统和方法的实施例。A. 网络和计算环境

[0044] 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，讨论可在其中部署这些实施例的网络和计算环境是有帮助的。现在参见图 1A，描述了网络环境的实施例。概括来讲，网络环境包括经由一个或多个网络 104、104’（总的称为网络 104）与一个或多个服务器 106a-106n（同样总的称为服务器 106，或远程机器 106）通信的一个或多个客户机 102a-102n（同样总的称为本地机器 102，或客户机 102）。在一些实施例中，客户机 102 通过设备 200 与服务器 106 通信。

[0045] 虽然图 1A 示出了在客户机 102 和服务器 106 之间的网络 104 和网络 104’，客户机 102 和服务器 106 可以位于同一个的网络 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 104’可为局域网（LAN）例如公司内网，城域网（MAN），或者广域网（WAN）例如因特网或万维网。在一个实施例中，网络 104 可为专用网络并且网络 104’可为公网。在一些实施例中，网络 104 可为专用网并且网络 104’可为公网。在又一个实施例中，网络 104 和 104’可都为专用网。在一些实施例中，客户机 102 可位于公司企业的分支机构中，通过网络 104 上的 WAN 连接与位于公司数据中心的服务器 106 通信。

[0046] 网络 104 和 / 或 104’可以是任何类型和 / 或形式的网络，并且可包括任何下述网络：点对点网络，广播网络，广域网，局域网，电信网络，数据通信网络，计算机网络，ATM（异步传输模式）网络，SONET（同步光纤网络）网络，SDH（同步数字体系）网络，无线网络和有线网络。在一些实施例中，网络 104 可以包括无线链路，诸如红外信道或者卫星频带。网络 104 和 / 或 104’的拓扑可为总线型、星型或环型网络拓扑。网络 104 和 / 或 104’以及网络拓扑可以是对于本领域普通技术人员所熟知的、可以支持此处描述的操作的任何这样的网络或网络拓扑。

[0047] 如图 1A 所示，设备 200 被显示在网络 104 和 104’之间，设备 200 也可被称为接口单元 200 或者网关 200。在一些实施例中，设备 200 可位于网络 104 上。例如，公司的分支机构可在分支机构中部署设备 200。在其他实施例中，设备 200 可以位于网络 104’上。例如，设备 200 可位于公司的数据中心。在又一个实施例中，多个设备 200 可在网络 104 上部署。在一些实施例中，多个设备 200 可部署在网络 104’上。在一个实施例中，第一设备 200 与第二设备 200’通信。在其他实施例中，设备 200 可为位于与客户机 102 同一或不同网络 104、104’的任一客户机 102 或服务器 106 的一部分。一个或多个设备 200 可位于客户机 102 和服务器 106 之间的网络或网络通信路径中的任一点。

[0048] 在一些实施例中，设备 200 包括由位于佛罗里达州 Ft.Lauderdale 的 Citrix Systems 公司制造的被称为 Citrix NetScaler 设备的任何网络设备。在其他实施例中，设备 200 包括由位于华盛顿州西雅图的 F5Networks 公司制造的被称为 WebAccelerator 和 BigIP 的任何一个产品实施例。在又一个实施例中，设备 205 包括由位于加利福尼亚州 Sunnyvale 的 Juniper Networks 公司制造的 DX 加速设备平台和 / 或诸如 SA700、SA2000、SA4000 和 SA6000 的 SSLVPN 系列设备中的任何一个。在又一个实施例中，设备 200 包括由位于加利福尼亚州 San Jose 的 Cisco Systems 公司制造的任何应用加速和 / 或安全

相关的设备和/或软件，例如Cisco ACE应用控制引擎模块服务(Application Control Engine Module service)软件和网络模块以及Cisco AVS系列应用速度系统(Application Velocity System)。

[0049] 在一个实施例中，系统可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38。在其中一些实施例中，服务器106可为地理上分散的。在一些情况中，群38可以作为单个实体被管理。在其他实施例中，服务器群38包括多个服务器群38。在一个实施例中，服务器群代表一个或多个客户机102执行一个或多个应用程序。

[0050] 在每个群38中的服务器106可为不同种类。一个或多个服务器106可根据一种类型的操作系统平台(例如，由华盛顿州Redmond的Microsoft公司制造的WINDOWS NT)操作，而一个或多个其它服务器106可根据另一类型的操作系统平台(例如，Unix或Linux)操作。每个群38的服务器106不需要与同一群38内的另一个服务器106物理上接近。因此，被逻辑分组为群38的服务器106组可使用广域网(WAN)连接或城域网(MAN)连接互联。例如，群38可包括物理上位于不同大陆或大陆的不同区域、国家、州、城市、校园或房间的服务器106。如果使用局域网(LAN)连接或一些直连形式来连接服务器106，则可增加群38中的服务器106间的数据传送速度。

[0051] 服务器106可指文件服务器、应用服务器、web服务器、代理服务器或者网关服务器。在一些实施例中，服务器106可以有作为应用服务器或者作为主应用服务器工作的能力。在一个实施例中，服务器106可包括活动目录。客户机102也可称为客户端节点或端点。在一些实施例中，客户机102可以有作为客户机节点寻求访问服务器上的应用的能力，也可以有作为应用服务器为其它客户机102a-102n提供对寄载的应用的访问的能力。

[0052] 在一些实施例中，客户机102与服务器106通信。在一个实施例中，客户机102与群38中的服务器106的其中一个直接通信。在又一个实施例中，客户机102执行程序邻近应用(program neighborhood application)以与群38内的服务器106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38中的服务器106通信。通过网络104，客户机102例如可以请求执行群38中的服务器106a-106n寄载的各种应用，并接收应用执行结果的输出进行显示。在一些实施例中，只有主节点提供识别和提供与寄载所请求的应用的服务器106’相关的地址信息所需的功能。

[0053] 在一个实施例中，服务器106提供web服务器的功能。在又一个实施例中，服务器106a接收来自客户机102的请求，将该请求转发到第二服务器106b，并使用来自服务器106b对该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由该应用的列举所识别的应用的服务器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通信以访问所识别的应用。在又一个实施例中，客户机102接收由执行服务器106上所识别的应用而产生的诸如显示数据的应用输出数据。

[0054] 现参考图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’。在又一个实施例中，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 可以被部署用于第一服务器群 38，而第二设备 200 可以被部署用于第二服务器群 38’。在另一个实例中，第一设备 200 可以被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流量或应用和数据的传送。

[0055] 现参考图 1C，描述了网络环境的又一个实施例，在该网络环境中，将设备 200 和一个或多个其它类型的设备部署在一起，例如，部署在一个或多个 WAN 优化设备 205，205’之间。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。例如，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。在其他实施例中，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。在又一个实施例中，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 可以被部署用于第一服务器群 38，而第二 WAN 优化设备 205’可以被部署用于第二服务器群 38’。

[0056] 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任何类型和形式的网络流量（例如去往和 / 或来自 WAN 连接的流量）的性能、操作或服务质量的装置。在一些实施例中，设备 205 是一个性能增强代理。在其他实施例中，设备 205 是任何类型和形式的 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 是由位于佛罗里达州 Ft. Lauderdale 的 Citrix Systems 公司出品的被称为 WANScaler 的产品实施例中的任何一种。在其他实施例中，设备 205 包括由位于华盛顿州 Seattle 的 F5Networks 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何一种。在又一个实施例中，设备 205 包括由位于加利福尼亚州 Sunnyvale 的 Juniper NetWorks 公司出品的 WX 和 WXC WAN 加速装置平台中的任何一种。在一些实施例中，设备 205 包括由加利福尼亚州 San Francisco 的 Riverbed Technology 公司出品的虹鳟（steelhead）系列 WAN 优化设备中的任何一种。在其他实施例中，设备 205 包括由位于新泽西州 Roseland 的 Expand Networks 公司出品的 WAN 相关装置中的任何一种。在一个实施例中，设备 205 包括由位于加利福尼亚州 Cupertino 的 Packeteer 公司出品的任何一种 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 包括由位于加利福尼亚州 San Jose 的 Cisco Systems 公司出品的任何 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。

[0057] 在一个实施例中，设备 205 为分支机构或远程办公室提供应用和数据加速服务。在一个实施例中，设备 205 包括广域文件服务（WAFS）的优化。在又一个实施例中，设备 205 加速文件的传送，例如经由通用互联网文件系统（CIFS）协议。在其他实施例中，设备 205 在存储器和 / 或存储装置中提供高速缓存来加速应用和数据的传送。在一个实施例中，设备 205 在任何级别的网络堆栈或在任何的协议或网络层中提供网络流量的压缩。在又一个实施例中，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传送。例如，在一个实施例中，设备 205 提供传输控制协议（TCP）优化。在其他实施例中，设备 205 提供对于任何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。

[0058] 在又一个实施例中，设备 205 将任何类型和形式的数据或信息编码成网络分组的定制的或标准的 TCP 和 / 或 IP 的报头字段或可选字段，以将其存在、功能或能力通告给另一个设备 205’。在又一个实施例中，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。

[0059] 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何信息。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何编码信息，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’。

[0060] 现参考图 1D，描述了用于传送和 / 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 包括用于向一个或多个客户机 102 传送计算环境或应用和 / 或数据文件的应用传送系统 190。总的来说，客户机 10 通过网络 104、104’和设备 200 与服务器 106 通信。例如，客户机 102 可驻留在公司的远程办公室里，例如分支机构，并且服务器 106 可驻留在公司数据中心。客户机 102 包括客户机代理 120 以及计算环境 15。计算环境 15 可执行或操作用于访问、处理或使用数据文件的应用。可经由设备 200 和 / 或服务器 106 传送计算环境 15、应用和 / 或数据文件。

[0061] 在一些实施例中，设备 200 加速计算环境 15 或者其任何部分到客户机 102 的传送。在一个实施例中，设备 200 通过应用传送系统 190 加速计算环境 15 的传送。例如，可使用此处描述的实施例来加速从公司中央数据中心到远程用户位置（例如公司的分支机构）的流应用（streaming application）及该应用可处理的数据文件的传送。在又一个实施例中，设备 200 加速客户机 102 和服务器 106 之间的传输层流量。设备 200 可以提供用于加速从服务器 106 到客户机 102 的任何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 响应于来自客户机 102 的请求提供服务器 106 的负载平衡。在其他实施例中，设备 200 充当代理或者访问服务器来提供对一个或者多个服务器 106 的访问。在又一个实施例中，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’的安全虚拟专用网络连接，诸如 SSL VPN 连接。在又一些实施例中，设备 200 提供客户机 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。

[0062] 在一些实施例中，基于多个执行方法并且基于通过策略引擎 195 所应用的任一验证和授权策略，应用传送管理系统 190 提供将计算环境传送到远程的或者另外的用户的桌面的应用传送技术。使用这些技术，远程用户可以从任何网络连接装置 100 获取计算环境并且访问服务器所存储的应用和数据文件。在一个实施例中，应用传送系统 190 可驻留在服务器 106 上或在其上执行。在又一个实施例中，应用传送系统 190 可驻留在多个服务器 106a-106n 上或在其上执行。在一些实施例中，应用传送系统 190 可在服务器群 38 内执行。在一个实施例中，执行应用传送系统 190 的服务器 106 也可存储或提供应用和数据文件。在又一个实施例中，一个或多个服务器 106 的第一组可执行应用传送系统 190，而不同的服务

器 106n 可存储或提供应用和数据文件。在一些实施例中，应用传送系统 190、应用和数据文件中的每一个可驻留或位于不同的服务器。在又一个实施例中，应用传送系统 190 的任何部分可驻留、执行、或被存储于或分发到设备 200 或多个设备。

[0063] 客户机 102 可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 可通过网络 104、104' 和设备 200 请求来自服务器 106 的应用和数据文件。在一个实施例中，设备 200 可以将来自客户机 102 的请求转发到服务器 106。例如，客户机 102 可能不具有本地存储或者本地可访问的应用和数据文件。响应于请求，应用传送系统 190 和 / 或服务器 106 可以传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 可以把应用作为应用流来传输，以在客户机 102 上的计算环境 15 中操作。

[0064] 在一些实施例中，应用传送系统 190 包括 Citrix Systems 有限公司的 Citrix Access Suite™ 的任一部分 (例如 MetaFrame 或 Citrix Presentation Server™)，和 / 或微软公司开发的 Microsoft® Windows 终端服务中的任何一个。在一个实施例中，应用传送系统 190 可以通过远程显示协议或者以其它方式通过基于远程计算或者基于服务器计算来传送一个或者多个应用到客户机 102 或者用户。在又一个实施例中，应用传送系统 190 可以通过应用流来传送一个或者多个应用到客户机或者用户。

[0065] 在一个实施例中，应用传送系统 190 包括策略引擎 195，其用于控制和管理对应用的访问、应用执行方法的选择以及应用的传送。在一些实施例中，策略引擎 195 确定用户或者客户机 102 可以访问的一个或者多个应用。在又一个实施例中，策略引擎 195 确定应用应该如何被传送到用户或者客户机 102，例如执行方法。在一些实施例中，应用传送系统 190 提供多个传送技术，从中选择应用执行的方法，例如基于服务器的计算、本地流式传输或传送应用给客户机 120 以用于本地执行。

[0066] 在一个实施例中，客户机 102 请求应用程序的执行并且包括服务器 106 的应用传送系统 190 选择执行应用程序的方法。在一些实施例中，服务器 106 从客户机 102 接收证书。在又一个实施例中，服务器 106 从客户机 102 接收对于可用应用的列举的请求。在一个实施例中，响应该请求或者证书的接收，应用传送系统 190 列举对于客户机 102 可用的多个应用程序。应用传送系统 190 接收执行所列举的应用的请求。应用传送系统 190 选择预定数量的方法之一来执行所列举的应用，例如响应策略引擎的策略。应用传送系统 190 可以选择执行应用的方法，使得客户机 102 接收通过执行服务器 106 上的应用程序所产生的应用输出数据。应用传送系统 190 可以选择执行应用的方法，使得本地机器 10 在检索包括应用的多个应用文件之后本地执行应用程序。在又一个实施例中，应用传送系统 190 可以选择执行应用的方法，以通过网络 104 流式传输应用到客户机 102。

[0067] 客户机 102 可以执行、操作或者以其它方式提供应用，所述应用可为任何类型和 / 或形式的软件、程序或者可执行指令，例如任何类型和 / 或形式的 web 浏览器、基于 web 的客户机、客户机 - 服务器应用、瘦客户端计算客户机、ActiveX 控件、或者 Java 程序、或者可以在客户机 102 上执行的任何其它类型和 / 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者基于远程的应用。在一个实施例中，服务器 106 可以使用任何瘦 - 客户端或远程显示协议来显示输出到客户机 102，所述瘦 - 客户端或远程显示协议例如由位于佛罗里达州 Ft. Lauderdale 的 Citrix Systems 公司出品的独立计算架构 (ICA) 协议或由位于华盛顿州 Redmond 的微软公司出品的远程桌

面协议(RDP)。应用可使用任何类型的协议,并且它可为,例如,HTTP 客户机、FTP 客户机、Oscar 客户机或 Telnet 客户机。在其他实施例中,应用包括和 VoIP 通信相关的任何类型的软件,例如软 IP 电话。在进一步的实施例中,应用包括涉及到实时数据通信的任一应用,例如用于流式传输视频和 / 或音频的应用。

[0068] 在一些实施例中,服务器 106 或服务器群 38 可运行一个或多个应用,例如提供瘦客户端计算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 38 作为一个应用来执行 Citrix Systems 有限公司的 Citrix Access Suite™ 的任一部分(例如 MetaFrame 或 Citrix Presentation Server™),和 / 或微软公司开发的 Microsoft® Windows 终端服务中的任何一个。在一个实施例中,该应用是位于佛罗里达州 Fort Lauderdale 的 Citrix Systems 有限公司开发的 ICA 客户机。在其他实施例中,该应用包括由位于华盛顿州 Redmond 的 Microsoft 公司开发的远程桌面(RDP)客户机。另外,服务器 106 可以运行一个应用,例如,其可以是提供电子邮件服务的应用服务器,例如由位于华盛顿州 Redmond 的 Microsoft 公司制造的 Microsoft Exchange, web 或 Internet 服务器,或者桌面共享服务器,或者协作服务器。在一些实施例中,任一应用可以包括任一类型的所寄载的服务或产品,例如位于加利福尼亚州 Santa Barbara 的 Citrix Online Division 公司提供的 GoToMeeting™,位于加利福尼亚州 Santa Clara 的 WebEx 有限公司提供的 WebEx™,或者位于华盛顿州 Redmond 的 Microsoft 公司提供的 Microsoft Office Live Meeting。

[0069] 仍参考图 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 可以包括任何类型和形式的性能监控服务 198。性能监控服务 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,性能监控服务 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何软件、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 Javascript 任何类型和形式的脚本。在一个实施例中,监控代理 197 相对于装置的任何应用和 / 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显眼地被安装和操作。在又一个实施例中,监控代理 197 的安装和操作不需要用于该应用或装置的任何设备。

[0070] 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。在其他实施例中,监控代理 197 基于检测到任何类型和形式的事件来监控、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 可以报告或提供任何所监控、测量或收集的数据给监控服务 198。在一个实施例中,监控代理 197 根据时间安排或预定频率来发送信息给监控服务 198。在又一个实施例中,监控代理 197 在检测到事件时发送信息给监控服务 198。

[0071] 在一些实施例中,监控服务 198 和 / 或监控代理 197 对诸如客户机、服务器、服务器群、设备 200、设备 205 或网络连接的任何网络资源或网络基础结构元件的进行监控和性能测量。在一个实施例中,监控服务 198 和 / 或监控代理 197 执行诸如 TCP 或 UDP 连接的任何传输层连接的监控和性能测量。在又一个实施例中,监控服务 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,监控服务 198 和 / 或监控代理 197 监控和测量带宽利用。

[0072] 在其他实施例中，监控服务 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，监控服务 198 执行应用的监控和性能测量。在又一个实施例中，监控服务 198 和 / 或监控代理 197 执行到应用的任何会话或连接的监控和性能测量。在一个实施例中，监控服务 198 和 / 或监控代理 197 监控和测量浏览器的性能。在又一个实施例中，监控服务 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，监控服务 198 和 / 或监控代理 197 监控和测量 IP 电话（VoIP）应用或会话的性能。在其他实施例中，监控服务 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，监控服务 198 和 / 或监控代理 197 监控和测量任何类型和形式的流媒体的性能。在进一步的实施例中，监控服务 198 和 / 或监控代理 197 监控和测量所寄载的应用或软件即服务（Software-As-A-Service，SaaS）传送模型的性能。

[0073] 在一些实施例中，监控服务 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。在其他实施例中，监控服务 198 和 / 或监控代理 197 监控和测量应用层堆栈的任何部分，例如任何 .NET 或 J2EE 调用。在一个实施例中，监控服务 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，监控服务 198 和 / 或监控代理 197 监控和测量任何方法、函数或应用编程接口（API）调用。

[0074] 在一个实施例中，监控服务 198 和 / 或监控代理 197 对经由诸如设备 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 / 或数据的传送进行监控和性能测量。在一些实施例中，监控服务 198 和 / 或监控代理 197 监控和测量虚拟化应用的传送的性能。在其他实施例中，监控服务 198 和 / 或监控代理 197 监控和测量流式应用的传送的性能。在又一个实施例中，监控服务 198 和 / 或监控代理 197 监控和测量传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。在又一个实施例中，监控服务 198 和 / 或监控代理 197 监控和测量客户机 / 服务器应用的性能。

[0075] 在一个实施例中，监控服务 198 和 / 或监控代理 197 被设计和构建成为应用传送系统 190 提供应用性能管理。例如，监控服务 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（Citrix Presentation Server）传送应用的性能。在该实例中，监控服务 198 和 / 或监控代理 197 监控单独的 ICA 会话。监控服务 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。监控服务 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（active server）。在一些实施例中，监控服务 198 和 / 或监控代理 197 监控在应用传送系统 190 和应用和 / 或数据库服务器之间的后端连接。监控服务 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。

[0076] 在一些实施例中，监控服务 198 和 / 或监控代理 197 测量和监控对于应用传送系统 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。在其他实施例中，监控服务 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 / 或每个进程的应用传送系统 190 的 CPU 使用。在又一个实施例中，监控服务 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、服务器或应用传送系统所需的时间。在一个实施例中，监控服务 198 和 / 或监控代理 197 测量和监控用户登录应用、服务器或应用传送系统 190 的持续时间。在一些实施例中，监控服务 198 和 / 或监控代理 197

测量和监控应用、服务器或应用传送系统会话的有效和无效的会话计数。在又一个实施例中，监控服务 198 和 / 或监控代理 197 测量和监控用户会话等待时间。

[0077] 在另外的实施例中，监控服务 198 和 / 或监控代理 197 测量和监控任何类型和形式的服务器指标。在一个实施例中，监控服务 198 和 / 或监控代理 197 测量和监控与系统内存、CPU 使用和盘存储器有关的指标。在又一个实施例中，监控服务 198 和 / 或监控代理 197 测量和监控和页错误有关的指标，诸如每秒页错误。在其他实施例中，监控服务 198 和 / 或监控代理 197 测量和监控往返时间的指标。在又一个实施例中，监控服务 198 和 / 或监控代理 197 测量和监控与应用崩溃、错误和 / 或中止相关的指标。

[0078] 在一些实施例中，监控服务 198 和监控代理 198 包括由位于佛罗里达州 Ft.Lauderdale 的 Citrix Systems 公司出品的被称为 EdgeSight 的任何一种产品实施例。在又一个实施例中，性能监控服务 198 和 / 或监控代理 198 包括由位于加利福尼亚州 Palo Alto 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，性能监控服务 198 和 / 或监控代理 198 包括由位于加利福尼亚州 San Francisco 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何部分。在其他实施例中，性能监控服务 198 和 / 或监控代理 198 包括由位于德克萨斯州 Houston 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 Patrol products）的商业服务管理产品的任何部分。

[0079] 客户机 102、服务器 106 和设备 200 可以被部署为和 / 或执行在任何类型和形式的计算装置上，诸如能够在任何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、网络装置或者设备。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 / 或诸如鼠标的指示装置 127。每个计算装置 100 也可包括其它可选元件，例如一个或多个输入 / 输出装置 130a-130b（总的使用附图标记 130 表示），以及与中央处理单元 101 通信的高速缓存存储器 140。

[0080] 中央处理单元 101 是响应并处理从主存储器单元 122 取出的指令的任何逻辑电路。在许多实施例中，中央处理单元由微处理器单元提供，例如：由加利福尼亚州 Mountain View 的 Intel 公司制造的微处理器单元；由伊利诺伊州 Schaumburg 的 Motorola 公司制造的微处理器单元；由加利福尼亚州 Santa Clara 的 Transmeta 公司制造的微处理器单元；由纽约州 White Plains 的 International Business Machines 公司制造的 RS/6000 处理器；或者由加利福尼亚州 Sunnyvale 的 Advanced Micro Devices 公司制造的微处理器单元。计算装置 100 可以基于这些处理器中的任何一种，或者能够如此处所述方式运行的任何其它处理器。

[0081] 主存储器单元 122 可以是能够存储数据并允许微处理器 101 直接访问任何存储位置的一个或多个存储器芯片，例如静态随机存取存储器 (SRAM)、突发 SRAM 或同步突发 SRAM (BSRAM)、动态随机存取存储器 DRAM、快速页模式 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 SRAM（ESDRAM）、同步链路 DRAM（SLDRAM）、直接内存总线

DRAM（DRDRAM）或铁电 RAM（FRAM）。主存储器 122 可以基于上述存储芯片的任何一种，或者能够如此处所述方式运行的任何其它可用存储芯片。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。

[0082] 图 1F 描述了在其中主处理器 101 通过第二总线与高速缓存存储器 140 直接通信的实施例，第二总线有时也称为后端总线。其他实施例中，主处理器 101 使用系统总线 150 和高速缓存存储器 140 通信。高速缓存存储器 140 通常有比主存储器 122 更快的响应时间，并且通常由 SRAM、BSRAM 或 EDRAM 提供。在图 1F 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。可以使用各种不同的总线将中央处理单元 101 连接到任何 I/O 装置 130，所述总线包括 VESA VL 总线、ISA 总线、EISA 总线、微通道体系结构（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）与显示器 124 通信。图 1F 说明了主处理器 101 通过超传输（HyperTransport）、快速 I/O 或者 InfiniBand 直接与 I/O 装置 130 通信的计算机 100 的一个实施例。图 1F 还描述了在其中混合本地总线和直接通信的实施例：处理器 101 使用本地互连总线与 I/O 装置 130b 进行通信，同时直接与 I/O 装置 130a 进行通信。

[0083] 计算装置 100 可以支持任何适当的安装装置 116，例如用于接纳诸如 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、各种格式的磁带驱动器、USB 装置、硬盘驱动器或适于安装像任何客户机代理 120 或其部分的软件和程序的任何其它装置。计算装置 100 还可以包括存储装置 128，诸如一个或者多个硬盘驱动器或者独立磁盘冗余阵列，用于存储操作系统和其它相关软件，以及用于存储诸如涉及客户机代理 120 的任何程序的应用软件程序。或者，可以使用安装装置 116 的任何一种作为存储装置 128。此外，操作系统和软件可从例如可引导 CD 的可引导介质运行，诸如KNOPPIX®，一种用于 GNU/Linux 的可引导 CD，该可引导 CD 可自 knoppix.net 作为 GNU/Linux 一个分发版获得。

[0084] 此外，计算装置 100 可以包括通过多种连接接口到局域网（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路、LAN 或 WAN 链路（例如 802.11，T1，T3、56kb、X.25）、宽带连接（如 ISDN、帧中继、ATM）、无线连接、或上述任何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 网络卡、卡总线网络适配器、无线网络适配器、USB 网络适配器、调制解调器或适用于将计算装置 100 接口到能够通信并执行这里所说明的操作的任何类型的网络的任何其它设备。计算装置 100 中可以包括各种 I/O 装置 130a-130n。输入装置包括键盘、鼠标、触控板、轨迹球、麦克风和绘图板。输出装置包括视频显示器、扬声器、喷墨打印机、激光打印机和热升华打印机。如图 1E 所示，I/O 装置 130 可以由 I/O 控制器 123 控制。I/O 控制器可以控制一个或多个 I/O 装置，例如键盘 126 和指示装置 127（如鼠标或光笔）。此外，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。在其他实施例中，计算装置 100 可以提供 USB 连接以接纳手持 USB 存储装置，例如由位于美国加利福尼亚州 Los Alamitos 的 Twintech Industry 有限公司生产的 USB 闪存驱动系列装置。

[0085] 在一些实施例中，计算装置100可以包括多个显示装置124a-124n或与其相连，这些显示装置各自可以是相同或不同的类型和/或形式。因而，任何一种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-124n。例如，计算装置100可以包括任何类型和/或形式的视频适配器、视频卡、驱动器和/或库，以与显示装置124a-124n接口、通信、连接或以其他方式使用显示装置。在一个实施例中，视频适配器可以包括多个连接器以与多个显示装置124a-124n接口。在其他实施例中，计算装置100可以包括多个视频适配器，每个视频适配器与显示装置124a-124n中的一个或多个连接。在一些实施例中，计算装置100的操作系统的任一部分都可以被配置用于使用多个显示器124a-124n。在其他实施例中，显示装置124a-124n中的一个或多个可以由一个或多个其它计算装置提供，诸如例如通过网络与计算装置100连接的计算装置100a和100b。这些实施例可以包括被设计和构造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。本领域的普通技术人员应认识和理解可以将计算装置100配置成具有多个显示装置124a-124n的各种方法和实施例。

[0086] 在另外的实施例中，I/O装置130可以是系统总线150和外部通信总线之间的桥170，所述外部通信总线例如USB总线、Apple桌面总线、RS-232串行连接、SCSI总线、FireWire总线、FireWire800总线、以太网总线、AppleTalk总线、千兆位以太网总线、异步传输模式总线、HIPPI总线、超级HIPPI总线、SerialPlus总线、SCI/LAMP总线、光纤信道总线或串行SCSI总线。

[0087] 图1E和1F中描述的那类计算装置100通常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可以运行任何操作系统，如Microsoft® Windows操作系统，不同发行版本的Unix和Linux操作系统，用于Macintosh计算机的任何版本的MAC OS®，任何嵌入式操作系统，任何实时操作系统，任何开源操作系统，任何专有操作系统，任何用于移动计算装置的操作系统，或者任何其它能够在计算装置上运行并完成这里所述操作的操作系统。典型的操作系统包括：WINDOWS3.x、WINDOWS95、WINDOWS98、WINDOWS2000、WINDOWS NT3.51、WINDOWS NT4.0、WINDOWS CE和WINDOWS XP，所有这些均由位于华盛顿州Redmond的微软公司出品；由位于加利福尼亚州Cupertino的苹果计算机出品的MacOS；由位于纽约州Armonk的国际商业机器公司出品的OS/2；以及由位于犹他州Salt Lake City的Caldera公司发布的可免费使用的Linux操作系统或者任何类型和/或形式的Unix操作系统，以及其它。

[0088] 在其他的实施例中，计算装置100可以有符合该装置的不同的处理器、操作系统和输入设备。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可以是任何工作站、桌面计算机、膝上型或笔记本计算机、服务器、手持计算机、移动电话、任何其它计算机、或能够通信并有足够的处理器能力和存储容量以执行此处所述的操作的其它形式的计算或者电信装置。

[0089] 如图1G所示，计算装置100可以包括多个处理器，可以提供用于对不只一个数据片同时执行多个指令或者同时执行一个指令的功能。在一些实施例中，计算装置100可包

括具有一个或多个核的并行处理器。在这些实施例的一个中，计算装置 100 是共享内存并行设备，具有多个处理器和 / 或多个处理器核，将所有可用内存作为一个全局地址空间进行访问。在这些实施例的又一个中，计算装置 100 是分布式存储器并行设备，具有多个处理器，每个处理器访问本地存储器。在这些实施例的又一个中，计算装置 100 既有共享的存储器又有仅由特定处理器或处理器子集访问的存储器。在这些实施例的又一个中，如多核微处理器的计算装置 100 将两个或多个独立处理器组合在一个封装中，通常在一个集成电路 (IC) 中。在这些实施例的又一个中，计算装置 100 包括具有单元宽带引擎 (CELL BROADBAND ENGINE) 架构的芯片，并包括高能处理器单元以及多个协同处理单元，高能处理器单元和多个协同处理单元通过内部高速总线连接在一起，可以将内部高速总线称为单元互连总线。

[0090] 在一些实施例中，处理器提供用于对多个数据片同时执行单个指令 (SIMD) 的功能。其他实施例中，处理器提供用于对多个数据片同时执行多个指令 (MIMD) 的功能。又一个实施例中，处理器可以在单个装置中使用 SIMD 和 MIMD 核的任意组合。

[0091] 在一些实施例中，计算装置 100 可包括图像处理单元。图 1H 所示的在这些实施例的一个中，计算装置 100 包括至少一个中央处理单元 101 和至少一个图像处理单元。在这些实施例的又一个中，计算装置 100 包括至少一个并行处理单元和至少一个图像处理单元。在这些实施例的又一个中，计算装置 100 包括任意类型的多个处理单元，多个处理单元中的一个包括图像处理单元。

[0092] 在一些实施例中，第一计算装置 100a 代表客户计算装置 100b 的用户执行应用。又一个实施例中，计算装置 100 执行虚拟机，其提供执行会话，在该会话中，代表客户计算装置 100b 的用户执行应用。在这些实施例的一个中，执行会话是寄载的桌面会话。在这些实施例的又一个中，计算装置 100 执行终端服务会话。终端服务会话可以提供寄载的桌面环境。在这些实施例的又一个中，执行会话提供对计算环境的访问，该计算环境可包括以下的一个或多个 :应用、多个应用、桌面应用以及可执行一个或多个应用的桌面会话。

[0093] B. 设备架构

[0094] 图 2A 示出设备 200 的一个示例实施例。提供图 2A 的设备 200 架构仅用于示例，并不意于作为限制性的架构。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。

[0095] 硬件层 206 提供硬件元件，在内核空间 204 和用户空间 202 中的程序和服务在该硬件元件上被执行。硬件层 206 也提供结构和元件，就设备 200 而言，这些结构和元件允许在内核空间 204 和用户空间 202 内的程序和服务既在内部进行数据通信又与外部进行数据通信。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，用于存储软件和数据的存储器 264，用于通过网络传输和接收数据的网络端口 266，以及用于执行与安全套接字协议层相关的功能处理通过网络传输和接收的数据的加密处理器 260。在一些实施例中，中央处理单元 262 可在单独的处理器中执行加密处理器 260 的功能。另外，硬件层 206 可包括用于每个处理单元 262 和加密处理器 260 的多处理器。处理器 262 可以包括以上结合图 1E 和 1F 所述的任一处理器 101。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'。在其他实施例中，处理器 262 或者 262' 包括多核处理器。

[0096] 虽然示出的设备 200 的硬件层 206 通常带有加密处理器 260，但是处理器 260 可为执行涉及任何加密协议的功能的处理器，例如安全套接字协议层 (SSL) 或者传输层安全

(TLS)协议。在一些实施例中,处理器 260 可为通用处理器(GPP),并且在进一步的实施例中,可为用于执行任何安全相关协议处理的可执行指令。

[0097] 虽然图 2 中设备 200 的硬件层 206 包括了某些元件,但是设备 200 的硬件部分或组件可包括计算装置的任何类型和形式的元件、硬件或软件,例如此处结合图 1E 和 1F 示出和讨论的计算装置 100。在一些实施例中,设备 200 可包括服务器、网关、路由器、开关、桥接器或其它类型的计算或网络设备,并且拥有与此相关的任何硬件和 / 或软件元件。

[0098] 设备 200 的操作系统分配、管理或另外分离可用的系统存储器到内核空间 204 和用户空间 204。在示例的软件架构 200 中,操作系统可以是任何类型和 / 或形式的 Unix 操作系统,尽管本发明并未这样限制。这样,设备 200 可以运行任何操作系统,如任何版本的 Microsoft® Windows 操作系统、不同版本的 Unix 和 Linux 操作系统、用于 Macintosh 计算机的任何版本的 Mac OS®、任何的嵌入式操作系统、任何的网络操作系统、任何的实时操作系统、任何的开放源操作系统、任何的专用操作系统、用于移动计算装置或网络装置的任何操作系统、或者能够运行在设备 200 上并执行此处所描述的操作的任何其它操作系统。

[0099] 保留内核空间 204 用于运行内核 230,内核 230 包括任何设备驱动器,内核扩展或其他内核相关软件。就像本领域技术人员所知的,内核 230 是操作系统的核心,并提供对资源以及设备 104 的相关硬件元件的访问、控制和管理。根据设备 200 的实施例,内核空间 204 也包括与高速缓存管理器 232 协同工作的多个网络服务或进程,高速缓存管理器 232 有时也称为集成的高速缓存,其益处此处将进一步详细描述。另外,内核 230 的实施例将依赖于通过设备 200 安装、配置或其他使用的操作系统的实施例。

[0100] 在一个实施例中,设备 200 包括一个网络堆栈 267,例如基于 TCP/IP 的堆栈,用于与客户机 102 和 / 或服务器 106 通信。在一个实施例中,使用网络堆栈 267 与第一网络(例如网络 108)以及第二网络 110 通信。在一些实施例中,设备 200 终止第一传输层连接,例如客户机 102 的 TCP 连接,并建立客户机 102 使用的到服务器 106 的第二传输层连接,例如,终止在设备 200 和服务器 106 的第二传输层连接。可通过单独的网络堆栈 267 建立第一和第二传输层连接。在其他实施例中,设备 200 可包括多个网络堆栈,例如 267 或 267',并且在一个网络堆栈 267 可建立或终止第一传输层连接,在第二网络堆栈 267'上可建立或者终止第二传输层连接。例如,一个网络堆栈可用于在第一网络上接收和传输网络分组,并且另一个网络堆栈用于在第二网络上接收和传输网络分组。在一个实施例中,网络堆栈 267 包括用于为一个或多个网络分组进行排队的缓冲器 243,其中网络分组由设备 200 传输。

[0101] 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 以及多协议压缩逻辑 238。在内核空间 204 或内核模式而不是用户空间 202 中运行这些组件或进程 232、240、234、236 和 238 提高这些组件中的每个单独的和结合的性能。内核操作意味着这些组件或进程 232、240、234、236 和 238 在设备 200 的操作系统的核地址空间中运行。例如,在内核模式中运行加密引擎 234 通过移动加密和解密操作到内核可改进加密性能,从而可减少在内核模式中的存储空间或内核线程与在用户模式中的存储空间或线程之间的传输的数量。例如,在内核模式获得的数据可能不需要传输或拷贝到运行在用户模式的进程或线程,例如从内核级数据结构到用户级数据结构。在另一个方面,也可减少内核模式和用户模式之间的上下文切换的数量。另外,在任何组件或进程 232、240、235、236 和 238 间的同步和通信在内核空间 204 中可被执行的更有效

率。

[0102] 在一些实施例中，组件 232、240、234、236 和 238 的任何部分可在内核空间 204 中运行或操作，而这些组件 232、240、234、236 和 238 的其它部分可在用户空间 202 中运行或操作。在一个实施例中，设备 200 使用内核级数据结构来提供对一个或多个网络分组的任何部分的访问，例如，包括来自客户机 102 的请求或者来自服务器 106 的响应的网络分组。在一些实施例中，可以由分组引擎 240 通过到网络堆栈 267 的传输层驱动器接口或过滤器获得内核级数据结构。内核级数据结构可包括通过与网络堆栈 267 相关的内核空间 204 可访问的任何接口和 / 或数据、由网络堆栈 267 接收或发送的网络流量或分组。在其他实施例中，任何组件或进程 232、240、234、236 和 238 可使用内核级数据结构来执行组件或进程的需要的操作。在一个实例中，当使用内核级数据结构时，组件 232、240、234、236 和 238 在内核模式 204 中运行，而在又一个实施例中，当使用内核级数据结构时，组件 232、240、234、236 和 238 在用户模式中运行。在一些实施例中，内核级数据结构可被拷贝或传递到第二内核级数据结构，或任何期望的用户级数据结构。

[0103] 高速缓存管理器 232 可包括软件、硬件或软件和硬件的任何组合，以提供对任何类型和形式的内容的高速缓存访问、控制和管理，例如对象或由源服务器 106 提供服务的动态产生的对象。由高速缓存管理器 232 处理和存储的数据、对象或内容可包括任何格式 (例如标记语言)的数据，或者通过任何协议的通信的任何类型的数据。在一些实施例中，高速缓存管理器 232 复制存储在其他地方的原始数据或先前计算、产生或传输的数据，其中相对于读高速缓存存储器元件，需要更长的访问时间以取得、计算或以其他方式得到原始数据。一旦数据被存储在高速缓存存储元件中，通过访问高速缓存的副本而不是重新获得或重新计算原始数据即可进行后续操作，因此而减少了访问时间。在一些实施例中，高速缓存元件可以包括设备 200 的存储器 264 中的数据对象。在其他实施例中，高速缓存存储元件可包括有比存储器 264 更快的存取时间的存储器。在又一个实施例中，高速缓存元件可以包括设备 200 的任一类型和形式的存储元件，诸如硬盘的一部分。在一些实施例中，处理单元 262 可提供被高速缓存管理器 232 使用的高速缓存存储器。在又一个实施例中，高速缓存管理器 232 可使用存储器、存储区或处理单元的任何部分和组合来高速缓存数据、对象或其它内容。

[0104] 另外，高速缓存管理器 232 包括用于执行此处描述的设备 200 的技术的任一实施例的任何逻辑、功能、规则或操作。例如，高速缓存管理器 232 包括基于无效时间周期的终止，或者从客户机 102 或服务器 106 接收无效命令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 可作为在内核空间 204 中执行的程序、服务、进程或任务而操作，并且在其他实施例中，在用户空间 202 中执行。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 可包括任何类型的通用处理器 (GPP)，或任何其他类型的集成电路，例如现场可编程门阵列 (FPGA)，可编程逻辑设备 (PLD)，或者专用集成电路 (ASIC)。

[0105] 策略引擎 236 可包括例如智能统计引擎或其它可编程应用。在一个实施例中，策略引擎 236 提供配置机制以允许用户识别、指定、定义或配置高速缓存策略。策略引擎 236，在一些实施例中，也访问存储器以支持数据结构，例如备份表或 hash 表，以启用用户选择的高速缓存策略决定。在其他实施例中，除了对安全、网络流量、网络访问、压缩或其它任何

由设备 200 执行的功能或操作的访问、控制和管理之外，策略引擎 236 可包括任何逻辑、规则、功能或操作以确定和提供对设备 200 所高速缓存的对象、数据、或内容的访问、控制和管理。特定高速缓存策略的其他实施例此处进一步描述。

[0106] 加密引擎 234 包括用于操控诸如 SSL 或 TLS 的任何安全相关协议或其中涉及的任何功能的处理的任何逻辑、商业规则、功能或操作。例如，加密引擎 234 加密并解密通过设备 200 传输的网络分组，或其任何部分。加密引擎 234 也可代表客户机 102a-102n、服务器 106a-106n 或设备 200 来设置或建立 SSL 或 TLS 连接。因此，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 使用隧道协议来提供在客户机 102a-102n 和服务器 106a-106n 间的虚拟专用网络。在一些实施例中，加密引擎 234 与加密处理器 260 通信。在其他实施例中，加密引擎 234 包括运行在加密处理器 260 上的可执行指令。

[0107] 多协议压缩引擎 238 包括用于压缩一个或多个网络分组协议（例如被设备 200 的网络堆栈 267 使用的任何协议）的任何逻辑、商业规则、功能或操作。在一个实施例中，多协议压缩引擎 238 双向压缩在客户机 102a-102n 和服务器 106a-106n 间任一基于 TCP/IP 的协议，包括消息应用编程接口（MAPI）（电子邮件）、文件传输协议（FTP）、超文本传输协议（HTTP）、通用互联网文件系统（CIFS）协议（文件传输）、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 协议以及互联网协议电话（VoIP）协议。在其他实施例中，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，提供任何标记语言的压缩，例如可扩展标记语言（XML）。在一个实施例中，多协议压缩引擎 238 提供任何高性能协议的压缩，例如设计用于设备 200 到设备 200 通信的任何协议。在又一个实施例中，多协议压缩引擎 238 使用修改的传输控制协议来压缩任何通信的任何载荷或任何通信，例如事务 TCP（T/TCP）、带有选择确认的 TCP（TCP-SACK）、带有大窗口的 TCP（TCP-LW）、例如 TCP-Vegas 协议的拥塞预报协议以及 TCP 欺骗协议（TCP spoofing protocol）。

[0108] 同样的，多协议压缩引擎 238 为用户加速经由桌面客户机乃至移动客户机访问应用的性能，所述桌面客户机例如 Micosoft Outlook 和非 web 瘦客户机，诸如由像 Oracle、SAP 和 Siebel 的通用企业应用所启动的任何客户机，所述移动客户机例如掌上电脑。在一些实施例中，通过在内核模式 204 内部执行并与访问网络堆栈 267 的分组处理引擎 240 集成，多协议压缩引擎 238 可以压缩 TCP/IP 协议携带的任何协议，例如任何应用层协议。

[0109] 高速层 2-7 集成分组引擎 240，通常也称为分组处理引擎，或分组引擎，负责设备 200 通过网络端口 266 接收和发送的分组的内核级处理的管理。高速层 2-7 集成分组引擎 240 可包括用于在例如接收网络分组和传输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 通信以通过网络端口 266 发送和接收网络分组。高速层 2-7 集成分组引擎 240 与加密引擎 234、高速缓存管理器 232、策略引擎 236 和多协议压缩逻辑 238 协同工作。更具体地，配置加密引擎 234 以执行分组的 SSL 处理，配置策略引擎 236 以执行涉及流量管理的功能，例如请求级内容切换以及请求级高速缓存重定向，并配置多协议压缩逻辑 238 以执行涉及数据压缩和解压缩的功能。

[0110] 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处理，例如，接收或者输出（即传输）网络

分组。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 处理网络分组。分组处理定时器 242 向分组引擎 240 提供任何类型和形式的信号以通知、触发或传输时间相关的事件、间隔或发生。在许多实施例中，分组处理定时器 242 以毫秒级操作，例如 100ms、50ms、或 25ms。例如，在一些实例中，分组处理定时器 242 提供时间间隔或者以其它方式使得由高速层 2-7 集成分组引擎 240 以 10ms 时间间隔处理网络分组，而在其他实施例中，使高速层 2-7 集成分组引擎 240 以 5ms 时间间隔处理网络分组，并且在进一步的实施例中，短到 3、2 或 1ms 时间间隔。高速层 2-7 集成分组引擎 240 在操作期间可与加密引擎 234、高速缓存管理器 232、策略引擎 236 以及多协议压缩引擎 238 连接、集成或通信。因此，响应于分组处理定时器 242 和 / 或分组引擎 240，可执行加密引擎 234、高速缓存管理器 232、策略引擎 236 以及多协议压缩引擎 238 的任何逻辑、功能或操作。因此，在由分组处理定时器 242 提供的时间间隔粒度，可执行加密引擎 234、高速缓存管理器 232、策略引擎 236 以及多协议压缩引擎 238 的任何逻辑、功能或操作，例如，时间间隔少于或等于 10ms。例如，在一个实施例中，高速缓存管理器 232 可响应于高速层 2-7 集成分组引擎 240 和 / 或分组处理定时器 242 来执行任何高速缓存的对象的终止。在又一个实施例中，高速缓存的对象的终止或无效时间被设定为与分组处理定时器 242 的时间间隔相同的粒度级，例如每 10ms。

[0111] 与内核空间 204 不同，用户空间 202 是被用户模式应用或在用户模式运行的程序所使用的操作系统的存储区域或部分。用户模式应用不能直接访问内核空间 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 包括图形用户接口（GUI）210、命令行接口（CLI）212、壳服务（shell service）214、健康监控程序 216 以及守护（daemon）服务 218。GUI210 和 CLI212 提供系统管理员或其他用户可与之交互并控制设备 200 操作的装置，例如通过设备 200 的操作系统。GUI210 和 CLI212 可包括运行在用户空间 202 或内核框架 204 中的代码。GUI210 可以是任何类型或形式的图形用户接口，可以通过文本、图形或其他形式由任何类型的程序或应用（如浏览器）来呈现。CLI212 可为任何类型和形式的命令行或基于文本的接口，例如通过操作系统提供的命令行。例如，CLI212 可包括壳，该壳是使用户与操作系统相互作用的工具。在一些实施例中，可通过 bash、csh、tcsh 或者 ksh 类型的壳提供 CLI212。壳服务 214 包括程序、服务、任务、进程或可执行指令以支持由用户通过 GUI210 和 / 或 CLI212 的与设备 200 或者操作系统的交互。

[0112] 健康监控程序 216 用于监控、检查、报告并确保网络系统正常运行，以及用户正通过网络接收请求的内容。健康监控程序 216 包括一个或多个程序、服务、任务、进程或可执行指令，为监控设备 200 的任何行为提供逻辑、规则、功能或操作。在一些实施例中，健康监控程序 216 拦截并检查通过设备 200 传递的任何网络流量。在其他实施例中，健康监控程序 216 通过任何合适的方法和 / 或机制与一个或多个下述设备连接 ：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 以及壳服务 214。因此，健康监控程序 216 可调用任何应用编程接口（API）以确定设备 200 的任何部分的状态、情况或健康。例如，健康监控程序 216 可周期性地查验（ping）或发送状态查询以检查程序、进程、服务或任务是否活动并当前正在运行。在又一个实施例中，健康监控程序 216 可检查由任何程序、进程、服务或任务提供的任何状态、错误或历史日志以确定设备 200 任何部分的任何状况、状态或错误。

[0113] 守护服务 218 是连续运行或在背景中运行的程序，并且处理设备 200 接收的周期

性服务请求。在一些实施例中，守护服务可向其他程序或进程(例如合适的另一个守护服务218)转发请求。如本领域技术人员所公知的，守护服务218可无人监护的运行，以执行连续的或周期性的系统范围功能，例如网络控制，或者执行任何需要的任务。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，而在其他实施例中，一个或多个守护服务218运行在内核空间。

[0114] 现参考图2B，描述了设备200的又一个实施例。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。服务器106的每一个可以提供一个或者多个网络相关服务270a-270n(称为服务270)。例如，服务器106可以提供http服务270。设备200包括一个或者多个虚拟服务器或者虚拟互联网协议服务器，称为vServer275、vS275、VIP服务器或者仅是VIP275a-275n(此处也称为vServer275)。vServer275根据设备200的配置和操作来接收、拦截或者以其它方式处理客户机102和服务器106之间的通信。

[0115] vServer275可以包括软件、硬件或者软件和硬件的任何组合。vServer275可包括在设备200中的用户模式202、内核模式204或者其任何组合中运行的任何类型和形式的程序、服务、任务、进程或者可执行指令。vServer275包括任何逻辑、功能、规则或者操作，以执行此处所述技术的任何实施例，诸如SSL VPN280、转换/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，vServer275建立到服务器106的服务270的连接。服务275可以包括能够连接到设备200、客户机102或者vServer275并与之通信的任何程序、应用、进程、任务或者可执行指令集。例如，服务275可以包括web服务器、http服务器、ftp、电子邮件或者数据库服务器。在一些实施例中，服务270是守护进程或者网络驱动器，用于监听、接收和/或发送应用的通信，诸如电子邮件、数据库或者企业应用。在一些实施例中，服务270可以在特定的IP地址、或者IP地址和端口上通信。

[0116] 在一些实施例中，vServer275应用策略引擎236的一个或者多个策略到客户机102和服务器106之间的网络通信。在一个实施例中，该策略与vServer275相关。在又一个实施例中，该策略基于用户或者用户组。在又一个实施例中，策略为通用的并且应用到一个或者多个vServer275a-275n，和通过设备200通信的任何用户或者用户组。在一些实施例中，策略引擎的策略具有基于通信的任何内容应用该策略的条件，通信的内容诸如互联网协议地址、端口、协议类型、分组中的头部或者字段、或者通信的上下文，诸如用户、用户组、vServer275、传输层连接、和/或客户机102或者服务器106的标识或者属性。

[0117] 在其他实施例中，设备200与策略引擎236通信或接口，以便确定远程用户或远程客户机102的验证和/或授权，以访问来自服务器106的计算环境15、应用和/或数据文件。在又一个实施例中，设备200与策略引擎236通信或交互，以便确定远程用户或远程客户机102的验证和/或授权，使得应用传送系统190传送一个或多个计算环境15、应用和/或数据文件。在又一个实施例中，设备200基于策略引擎236对远程用户或远程客户机102的验证和/或授权建立VPN或SSL VPN连接。一个实施例中，设备200基于策略引擎236的策略控制网络流量以及通信会话。例如，基于策略引擎236，设备200可控制对计算环境15、应用或数据文件的访问。

[0118] 在一些实施例中，vServer275与客户机102经客户机代理120建立传输层连接，诸

如 TCP 或者 UDP 连接。在一个实施例中，vServer275 监听和接收来自客户机 102 的通信。在其他实施例中，vServer275 与客户机服务器 106 建立传输层连接，诸如 TCP 或者 UDP 连接。在一个实施例中，vServer275 建立到运行在服务器 106 上的服务器 270 的互联网协议地址和端口的传输层连接。在又一个实施例中，vServer275 将到客户机 102 的第一传输层连接与到服务器 106 的第二传输层连接相关联。在一些实施例中，vServer275 建立到服务器 106 的传输层连接池并经由所述池化（pooled）的传输层连接多路复用客户机的请求。

[0119] 在一些实施例中，设备 200 提供客户机 102 和服务器 106 之间的 SSL VPN 连接 280。例如，第一网络 102 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’是不能从第一网络 104 路由的。在其他实施例中，客户机 102 位于公用网络 104 上，并且服务器 106 位于专用网络 104’上，例如企业网。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，加密该通信，并且经第一传输层连接发送该通信到设备 200。设备 200 将第一网络 104 上的第一传输层连接与到第二网络 104 上的服务器 106 的第二传输层连接相关联。设备 200 接收来自客户机代理 102 的所拦截的通信，解密该通信，并且经第二传输层连接发送该通信到第二网络 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。同样的，设备 200 为两个网络 104、104’之间的客户机 102 提供端到端安全传输层连接。

[0120] 在一个实施例中，设备 200 寄载虚拟专用网络 104 上的客户机 102 的内部网互联网协议或者 IntranetIP282 地址。客户机 102 具有本地网络标识符，诸如第一网络 104 上的互联网协议（IP）地址和 / 或主机名称。当经设备 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、分配或者以其它方式提供 IntranetIP，其是诸如 IP 地址和 / 或主机名称的网络标识符。使用为客户机的所建立的 IntranetIP282，设备 200 在第二或专用网 104’上监听并接收指向该客户机 102 的任何通信。在一个实施例中，设备 200 在第二专用网络 104 上用作或者代表客户机 102。例如，在又一个实施例中，vServer275 监听和响应到客户机 102 的 IntranetIP282 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，设备 200 如同客户机 102 一样来处理该请求。例如，设备 200 可以响应对客户机 IntranetIP282 的查验。在又一个实施例中，设备可以与请求和客户机 IntranetIP282 连接的第二网络 104 上的计算装置 100 建立连接，诸如 TCP 或者 UDP 连接。

[0121] 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 通过开启与每一服务器 106 的一个或者多个传输层连接并且维持这些连接以允许由客户机经因特网的重复数据访问，来为服务器 106 缓解由重复开启和关闭到客户机 102 的传输层连接所造成的大量处理负载。该技术此处称为“连接池”。

[0122] 在一些实施例中，为了经池化的传输层连接无缝拼接从客户机 102 到服务器 106 的通信，设备 200 通过在传输层协议级修改序列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。分组的序列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 所期待的序列号和确认。在一些实施例中，传输层协议的分组校验和被重新计算以计及这些转换。

[0123] 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布流量并将客户机请求定向到服务器 106。在一个实施例中，尽管网络分组的网络层或者层 2 识别目的服务器 106，但设备 200 通过承载为传输层分组的有效载荷的数据和应用信息来确定服务器 106 以便分发网络分组。在一个实施例中，设备 200 的健康监控程序 216 监控服务器的健康来确定分发客户机请求到哪个服务器 106。在一些实施例中，如果设备 200 探测到某个服务器 106 不可用或者具有超过预定阈值的负载，设备 200 可以将客户机请求指向或者分发到另一个服务器 106。

[0124] 在一些实施例中，设备 200 用作域名服务（DNS）解析器或者以其它方式为来自客户机 102 的 DNS 请求提供解析。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 以设备 200 的 IP 地址或其所寄载的 IP 地址来响应客户机的 DNS 请求。在此实施例中，客户机 102 把用于域名的网络通信发送到设备 200。在又一个实施例中，设备 200 以第二设备 200’的或其所寄载的 IP 地址来响应客户机的 DNS 请求。在一些实施例中，设备 200 使用由设备 200 确定的服务器 106 的 IP 地址来响应客户机的 DNS 请求。

[0125] 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 提供用于探测和阻断非法请求的规则。在一些实施例中，应用防火墙 290 防御拒绝服务（DoS）攻击。在其他实施例中，设备检查所拦截的请求的内容，以识别和阻断基于应用的攻击。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脆弱点 :1）缓冲区泄出，2）CGI-BIN 参数操纵，3）表单 / 隐藏字段操纵，4）强制浏览，5）cookie 或会话中毒，6）被破坏的访问控制列表（ACLs）或弱密码，7）跨站脚本处理（XSS），8）命令注入，9）SQL 注入，10）错误触发敏感信息泄露，11）对加密的不安全使用，12）服务器错误配置，13）后门和调试选项，14）网站涂改，15）平台或操作系统弱点，和 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 cookie 不被修改。在其他实施例中，应用防火墙 290 通过执行合法的 URL 来防御强制浏览。

[0126] 在其他实施例中，应用防火墙 290 保护在网络通信中包含的任何机密信息。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在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。基于这些出现，在一个实施例中，应用防火墙 290 可以对网络通信采取策略行动，诸如阻止发送网络通信。在又一个实施例中，应用防火墙

290 可以重写、移动或者以其它方式掩盖该所识别的出现或者机密信息。

[0127] 仍参考图 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 中所描述的监控服务 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，以用于传送给与设备 200 通信的任何客户机或服务器。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。在其他实施例中，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。在又一个实施例中，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。

[0128] 在其他实施例中，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 上执行的任何应用、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。在又一个实施例中，监控代理 197 测量和监控设备 200 的任何传输层连接的性能。在一些实施例中，监控代理 197 测量和监控通过设备 200 的任何用户会话的性能。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 会话的任何虚拟专用网连接和 / 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 的内存、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 执行的任何加速技术 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。在其他实施例中，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。

[0129] C. 客户机代理

[0130] 现参考图 3，描述客户机代理 120 的实施例。客户机 102 包括客户机代理 120，用于经由网络 104 与设备 200 和 / 或服务器 106 来建立和交换通信。总的来说，客户机 102 在计算装置 100 上操作，该计算装置 100 拥有带有内核模式 302 以及用户模式 303 的操作系统，以及带有一个或多个层 310a-310b 的网络堆栈 310。客户机 102 可以已经安装和 / 或执行一个或多个应用。在一些实施例中，一个或多个应用可通过网络堆栈 310 与网络 104 通信。所述应用之一，诸如 web 浏览器，也可包括第一程序 322。例如，可在一些实施例中使用第一程序 322 来安装和 / 或执行客户机代理 120，或其中任何部分。客户机代理 120 包括拦截机制或者拦截器 350，用于从网络堆栈 310 拦截来自一个或者多个应用的网络通信。

[0131] 客户机 102 的网络堆栈 310 可包括任何类型和形式的软件、或硬件或其组合，用于提供与网络的连接和通信。在一个实施例中，网络堆栈 310 包括用于网络协议组的软件实现。网络堆栈 310 可包括一个或多个网络层，例如为本领域技术人员所公认和了解的开放式系统互联（OSI）通信模型的任何网络层。这样，网络堆栈 310 可包括用于任何以下 OSI 模型层的任何类型和形式的协议 ：1）物理链路层 ；2）数据链路层 ；3）网络层 ；4）传输层 ；5）会话层）；6）表示层，以及 7）应用层。在一个实施例中，网络堆栈 310 可包括在互联网协议（IP）的网络层协议上的传输控制协议（TCP），通常称为 TCP/IP。在一些实施例中，可在以太网协议上承载 TCP/IP 协议，以太网协议可包括 IEEE 广域网（WAN）或局域网（LAN）协议的任何族，例如被 IEEE802.3 覆盖的这些协议。在一些实施例中，网络堆栈 310 包括任何类型和

形式的无线协议，例如 IEEE802.11 和 / 或移动互联网协议。

[0132] 考虑基于 TCP/IP 的网络，可使用任何基于 TCP/IP 的协议，包括消息应用编程接口（MAPI）（email）、文件传输协议（FTP）、超文本传输协议（HTTP）、通用因特网文件系统（CIFS）协议（文件传输）、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，以及互联网协议电话（VoIP）协议。在又一个实施例中，网络堆栈 310 包括任何类型和形式的传输控制协议，诸如修改的传输控制协议，例如事务 TCP（T/TCP），带有选择确认的 TCP（TCP-SACK），带有大窗口的 TCP（TCP-LW），例如 TCP-Vegas 协议的拥塞预测协议，以及 TCP 欺骗协议。在其他实施例中，网络堆栈 310 可使用诸如基于 IP 的 UDP 的任何类型和形式的用户数据报协议（UDP），例如用于语音通信或实时数据通信。

[0133] 另外，网络堆栈 310 可包括支持一个或多个层的一个或多个网络驱动器，例如 TCP 驱动器或网络层驱动器。网络层驱动器可作为计算装置 100 的操作系统的一部分或者作为计算装置 100 的任何网络接口卡或其它网络访问组件的一部分被包括。在一些实施例中，网络堆栈 310 的任何网络驱动器可被定制、修改或调整以提供网络堆栈 310 的定制或修改部分，用来支持此处描述的任何技术。在其他实施例中，设计并构建加速程序 302 以与网络堆栈 310 协同操作或工作，上述网络堆栈 310 由客户机 102 的操作系统安装或以其它方式提供。

[0134] 网络堆栈 310 包括任何类型和形式的接口，用于接收、获得、提供或以其它方式访问涉及客户机 102 的网络通信的任何信息和数据。在一个实施例中，与网络堆栈 310 的接口包括应用编程接口（API）。接口也可包括任何函数调用、钩子或过滤机制，事件或回调机制、或任何类型的接口技术。网络堆栈 310 通过接口可接收或提供与网络堆栈 310 的功能或操作相关的任何类型和形式的数据结构，例如对象。例如，数据结构可以包括与网络分组相关的信息和数据或者一个或多个网络分组。在一些实施例中，数据结构包括在网络堆栈 310 的协议层处理的网络分组的一部分，例如传输层的网络分组。在一些实施例中，数据结构 325 包括内核级别数据结构，而在其他实施例中，数据结构 325 包括用户模式数据结构。内核级数据结构可以包括获得的或与在内核模式 302 中操作的网络堆栈 310 的一部分相关的数据结构、或者运行在内核模式 302 中的网络驱动程序或其它软件、或者由运行或操作在操作系统的内核模式的服务、进程、任务、线程或其它可执行指令获得或收到的任何数据结构。

[0135] 此外，网络堆栈 310 的一些部分可在内核模式 302 执行或操作，例如，数据链路或网络层，而其他部分在用户模式 303 执行或操作，例如网络堆栈 310 的应用层。例如，网络堆栈的第一部分 310a 可以给应用提供对网络堆栈 310 的用户模式访问，而网络堆栈 310 的第二部分 310a 提供对网络的访问。在一些实施例中，网络堆栈的第一部分 310a 可包括网络堆栈 310 的一个或多个更上层，例如层 5-7 的任何层。在其他实施例中，网络堆栈 310 的第二部分 310b 包括一个或多个较低的层，例如层 1-4 的任何层。网络堆栈 310 的每个第一部分 310a 和第二部分 310b 可包括网络堆栈 310 的任何部分，位于任何一个或多个网络层，处于用户模式 203、内核模式 202，或其组合，或在网络层的任何部分或者到网络层的接口点，或用户模式 203 和内核模式 202 的任何部分或到用户模式 203 和内核模式 202 的接口点。

[0136] 拦截器 350 可以包括软件、硬件、或者软件和硬件的任何组合。在一个实施例中，拦截器 350 在网络堆栈 310 的任一点拦截网络通信，并且重定向或者发送网络通信到由拦

截器 350 或者客户机代理 120 所期望的、管理的或者控制的目的地。例如，拦截器 350 可以拦截第一网络的网络堆栈 310 的网络通信并且发送该网络通信到设备 200，用于在第二网络 104 上发送。在一些实施例中，拦截器 350 包括含有诸如被构建和设计来与网络堆栈 310 对接并一同工作的网络驱动器的驱动器的任一类型的拦截器 350。在一些实施例中，客户机代理 120 和 / 或拦截器 350 操作在网络堆栈 310 的一个或者多个层，诸如在传输层。在一个实施例中，拦截器 350 包括过滤器驱动器、钩子机制、或者连接到网络堆栈的传输层的任一形式和类型的合适网络驱动器接口，诸如通过传输驱动器接口（TDI）。在一些实施例中，拦截器 350 连接到诸如传输层的第一协议层和诸如传输协议层之上的任何层的另一个协议层，例如，应用协议层。在一个实施例中，拦截器 350 可以包括遵守网络驱动器接口规范（NDIS）的驱动器，或者 NDIS 驱动器。在又一个实施例中，拦截器 350 可以包括微型过滤器或者微端口驱动器。在一个实施例中，拦截器 350 或其部分在内核模式 202 中操作。在又一个实施例中，拦截器 350 或其部分在用户模式 203 中操作。在一些实施例中，拦截器 350 的一部分在内核模式 202 中操作，而拦截器 350 的另一部分在用户模式 203 中操作。在其他实施例中，客户机代理 120 在用户模式 203 操作，但通过拦截器 350 连接到内核模式驱动器、进程、服务、任务或者操作系统的部分，诸如以获取内核级数据结构 225。在其他实施例中，拦截器 350 为用户模式应用或者程序，诸如应用。

[0137] 在一个实施例中，拦截器 350 拦截任何的传输层连接请求。在这些实施例中，拦截器 350 执行传输层应用编程接口（API）调用以设置目的地信息，诸如到期望位置的目的地 IP 地址和 / 或端口用于定位。以此方式，拦截器 350 拦截并重定向传输层连接到由拦截器 350 或客户机代理 120 控制或管理的 IP 地址和端口。在一个实施例中，拦截器 350 把连接的目的地信息设置为客户机代理 120 监听的客户机 102 的本地 IP 地址和端口。例如，客户机代理 120 可以包括为重定向的传输层通信监听本地 IP 地址和端口的代理服务。在一些实施例中，客户机代理 120 随后将重定向的传输层通信传送到设备 200。

[0138] 在一些实施例中，拦截器 350 拦截域名服务（DNS）请求。在一个实施例中，客户机代理 120 和 / 或拦截器 350 解析 DNS 请求。在又一个实施例中，拦截器发送所拦截的 DNS 请求到设备 200 以进行 DNS 解析。在一个实施例中，设备 200 解析 DNS 请求并且将 DNS 响应传送到客户机代理 120。在一些实施例中，设备 200 经另一个设备 200’或者 DNS 服务器 106 来解析 DNS 请求。

[0139] 在又一个实施例中，客户机代理 120 可以包括两个代理 120 和 120’。在一个实施例中，第一代理 120 可以包括在网络堆栈 310 的网络层操作的拦截器 350。在一些实施例中，第一代理 120 拦截网络层请求，诸如因特网控制消息协议（ICMP）请求（例如，查验和跟踪路由）。在其他实施例中，第二代理 120’可以在传输层操作并且拦截传输层通信。在一些实施例中，第一代理 120 在网络堆栈 210 的一层拦截通信并且与第二代理 120’连接或者将所拦截的通信传送到第二代理 120’。

[0140] 客户机代理 120 和 / 或拦截器 350 可以以对网络堆栈 310 的任何其它协议层透明的方式在协议层操作或与之对接。例如，在一个实施例中，拦截器 350 可以以对诸如网络层的传输层之下的任何协议层和诸如会话、表示或应用层协议的传输层之上的任何协议层透明的方式在网络堆栈 310 的传输层操作或与之对接。这允许网络堆栈 310 的其它协议层如所期望的进行操作并无需修改以使用拦截器 350。这样，客户机代理 120 和 / 或拦截器 350

可以与传输层连接以安全、优化、加速、路由或者负载平衡经由传输层承载的任一协议提供的任一通信，诸如 TCP/IP 上的任一应用层协议。

[0141] 此外，客户机代理 120 和 / 或拦截器可以以对任何应用、客户机 102 的用户和与客户机 102 通信的诸如服务器的任何其它计算装置透明的方式在网络堆栈 310 上操作或与之对接。客户机代理 120 和 / 或拦截器 350 可以以无需修改应用的方式被安装和 / 或执行在客户机 102 上。在一些实施例中，客户机 102 的用户或者与客户机 102 通信的计算装置未意识到客户机代理 120 和 / 或拦截器 350 的存在、执行或者操作。同样，在一些实施例中，相对于应用、客户机 102 的用户、诸如服务器的另一个计算装置、或者在由拦截器 350 连接的协议层之上和 / 或之下的任何协议层透明地来安装、执行和 / 或操作客户机代理 120 和 / 或拦截器 350。

[0142] 客户机代理 120 包括加速程序 302、流客户机 306、收集代理 304 和 / 或监控代理 197。在一个实施例中，客户机代理 120 包括由佛罗里达州 Fort Lauderdale 的 Citrix Systems Inc. 开发的独立计算架构（ICA）客户机或其任一部分，并且也指 ICA 客户机。在一些实施例中，客户机代理 120 包括应用流客户机 306，用于从服务器 106 流式传输应用到客户机 102。在一些实施例中，客户机代理 120 包括加速程序 302，用于加速客户机 102 和服务器 106 之间的通信。在又一个实施例中，客户机代理 120 包括收集代理 304，用于执行端点检测 / 扫描并且用于为设备 200 和 / 或服务器 106 收集端点信息。

[0143] 在一些实施例中，加速程序 302 包括用于执行一个或多个加速技术的客户机侧加速程序，以加速、增强或者以其他方式改善客户机与服务器 106 的通信和 / 或对服务器 106 的访问，诸如访问由服务器 106 提供的应用。加速程序 302 的可执行指令的逻辑、函数和 / 或操作可以执行一个或多个下列加速技术 :1）多协议压缩，2）传输控制协议池，3）传输控制协议多路复用，4）传输控制协议缓冲，以及 5）通过高速缓存管理器的高速缓存。另外，加速程序 302 可执行由客户机 102 接收和 / 或发送的任何通信的加密和 / 或解密。在一些实施例中，加速程序 302 以集成的方式或者格式执行一个或者多个加速技术。另外，加速程序 302 可以对作为传输层协议的网络分组的有效载荷所承载的任一协议或者多协议执行压缩。

[0144] 流客户机 306 包括应用、程序、进程、服务、任务或者可执行指令，所述应用、程序、进程、服务、任务或者可执行指令用于接收和执行从服务器 106 所流式传输的应用。服务器 106 可以流式传输一个或者多个应用数据文件到流客户机 306，用于播放、执行或者以其它方式引起客户机 102 上的应用被执行。在一些实施例中，服务器 106 发送一组压缩或者打包的应用数据文件到流客户机 306。在一些实施例中，多个应用文件被压缩并存储在文件服务器上档案文件中，例如 CAB、ZIP、SIT、TAR、JAR 或其它档案文件。在一个实施例中，服务器 106 解压缩、解包或者解档应用文件并且将该文件发送到客户机 102。在又一个实施例中，客户机 102 解压缩、解包或者解档应用文件。流客户机 306 动态安装应用或其部分，并且执行该应用。在一个实施例中，流客户机 306 可以为可执行程序。在一些实施例中，流客户机 306 可以能够启动另一个可执行程序。

[0145] 收集代理 304 包括应用、程序、进程、服务、任务或者可执行指令，用于识别、获取和 / 或收集关于客户机 102 的信息。在一些实施例中，设备 200 发送收集代理 304 到客户机 102 或者客户机代理 120。可以根据设备的策略引擎 236 的一个或多个策略来配置收集

代理 304。在其他实施例中，收集代理 304 发送在客户机 102 上收集的信息到设备 200。在一个实施例中，设备 200 的策略引擎 236 使用所收集的信息来确定和提供到网络 104 的客户机连接的访问、验证和授权控制。

[0146] 在一个实施例中，收集代理 304 包括端点检测和扫描机制，其识别并且确定客户机的一个或者多个属性或者特征。例如，收集代理 304 可以识别和确定任何一个或多个以下的客户机侧属性：1）操作系统和 / 或操作系统的版本，2）操作系统的服务包，3）运行的服务，4）运行的进程，和 5）文件。收集代理 304 还可以识别并确定客户机上任何一个或多个以下软件的存在或版本：1）防病毒软件；2）个人防火墙软件；3）防垃圾邮件软件，和 4）互联网安全软件。策略引擎 236 可以具有基于客户机或客户机侧属性的任何一个或多个属性或特性的一个或多个策略。

[0147] 在一些实施例中，客户机代理 120 包括如结合图 1D 和 2B 所讨论的监控代理 197。监控代理 197 可以是诸如 Visual Basic 或 Java 脚本的任何类型和形式的脚本。在一个实施例中，监控代理 197 监控和测量客户机代理 120 的任何部分的性能。例如，在一些实施例中，监控代理 197 监控和测量加速程序 302 的性能。在又一个实施例中，监控代理 197 监控和测量流客户机 306 的性能。在其他实施例中，监控代理 197 监控和测量收集代理 304 的性能。在又一个实施例中，监控代理 197 监控和测量拦截器 350 的性能。在一些实施例中，监控代理 197 监控和测量客户机 102 的诸如存储器、CPU 和磁盘的任何资源。

[0148] 监控代理 197 可以监控和测量客户机的任何应用的性能。在一个实施例中，监控代理 197 监控和测量客户机 102 上的浏览器的性能。在一些实施例中，监控代理 197 监控和测量经由客户机代理 120 传送的任何应用的性能。在其他实施例中，监控代理 197 测量和监控应用的最终用户响应时间，例如基于 web 的响应时间或 HTTP 响应时间。监控代理 197 可以监控和测量 ICA 或 RDP 客户机的性能。在又一个实施例中，监控代理 197 测量和监控用户会话或应用会话的指标。在一些实施例中，监控代理 197 测量和监控 ICA 或 RDP 会话。在一个实施例中，监控代理 197 测量和监控设备 200 在加速传送应用和 / 或数据到客户机 102 的过程中的性能。

[0149] 在一些实施例中，仍参考图 3，第一程序 322 可以用于自动地、静默地、透明地或者以其它方式安装和 / 或执行客户机代理 120 或其部分，诸如拦截器 350。在一个实施例中，第一程序 322 包括插件组件，例如 ActiveX 控件或 Java 控件或脚本，其加载到应用并由应用执行。例如，第一程序包括由 web 浏览器应用载入和运行的 ActiveX 控件，例如在存储器空间或应用的上下文中。在又一个实施例中，第一程序 322 包括可执行指令组，该可执行指令组被例如浏览器的应用载入并执行。在一个实施例中，第一程序 322 包括被设计和构造的程序以安装客户机代理 120。在一些实施例中，第一程序 322 通过网络从另一个计算装置获得、下载、或接收客户机代理 120。在又一个实施例中，第一程序 322 是用于在客户机 102 的操作系统上安装如网络驱动的程序的安装程序或即插即用管理器。

[0150] D. 用于提供虚拟化应用传送控制器的系统和方法

[0151] 现参考图 4A，该框图描述虚拟化环境 400 的一个实施例。总体而言，计算装置 100 包括管理程序层、虚拟化层和硬件层。管理程序层包括管理程序 401（也称为虚拟化管理器），其通过在虚拟化层中执行的至少一个虚拟机来分配和管理对硬件层中的多个物理资源（例如处理器 421 和盘 428）的访问。虚拟化层包括至少一个操作系统 410 和分配给至少

一个操作系统 410 的多个虚拟资源。虚拟资源可包括而不限于多个虚拟处理器 432a、432b、432c（总称为 432）和虚拟盘 442a、442b、442c（总称为 442），以及如虚拟存储器和虚拟网络接口的虚拟资源。可将多个虚拟资源和操作系统称为虚拟机 406。虚拟机 406 可包括控制操作系统 405，该控制操作系统 405 与管理程序 401 通信，并用于执行应用以管理并配置计算装置 100 上的其他虚拟机。

[0152] 具体而言，管理程序 401 可以以模拟可访问物理设备的操作系统的任何方式向操作系统提供虚拟资源。管理程序 401 可以向任何数量的客户操作系统 410a、410b（总称为 410）提供虚拟资源。一些实施例中，计算装置 100 执行一种或多种管理程序。这些实施例中，管理程序可用于模拟虚拟硬件、划分物理硬件、虚拟化物理硬件并执行提供对计算环境的访问的虚拟机。管理程序可包括由位于美国加州的 Palo Alto 的 VMWare 制造的这些程序；XEN 管理程序（一种开源产品，其开发由开源 Xen.org 协会监管）；由微软公司提供的 HyperV、VirtualServer 或虚拟 PC 管理程序，或其他。一些实施例中，计算装置 100 执行创建客户操作系统可在其上执行虚拟机平台的管理程序，该计算装置 100 被称为宿主服务器。在这些实施例的一个中，例如，计算装置 100 是由位于美国佛罗里达州 Fort Lauderdale 的 Citrix Systems 有限公司提供的 XEN SERVER。

[0153] 在一些实施例中，管理程序 401 在计算装置上执行的操作系统之内执行。在这些实施例的一个中，执行操作系统和管理程序 401 的计算装置可被视为具有宿主操作系统（执行在计算装置上的操作系统），和客户操作系统（在由管理程序 401 提供的计算资源分区内执行的操作系统）。其他实施例中，管理程序 401 和计算装置上的硬件直接交互而不是在宿主操作系统上执行。在这些实施例的一个中，管理程序 401 可被视为在“裸金属（bare metal）”上执行，所述“裸金属”指包括计算装置的硬件。

[0154] 在一些实施例中，管理程序 401 可以产生操作系统 410 在其中执行的虚拟机 406a-c（总称为 406）。在这些实施例的一个中，管理程序 401 加载虚拟机映像以创建虚拟机 406。在这些实施例的又一个中，管理程序 401 在虚拟机 406 内执行操作系统 410。仍在这些实施例的又一个中，虚拟机 406 执行操作系统 410。

[0155] 在一些实施例中，管理程序 401 控制在计算装置 100 上执行的虚拟机 406 的处理器调度和内存划分。在这些实施例的一个中，管理程序 401 控制至少一个虚拟机 406 的执行。在这些实施例的又一个中，管理程序 401 向至少一个虚拟机 406 呈现由计算装置 100 提供的至少一个硬件资源的抽象。其他实施例中，管理程序 401 控制是否以及如何将物理处理器能力呈现给虚拟机 406。

[0156] 控制操作系统 405 可以执行用于管理和配置客户操作系统的至少一个应用。一个实施例中，控制操作系统 405 可以执行管理应用，如包括如下用户接口的应用，该用户接口为管理员提供对用于管理虚拟机执行的功能的访问，这些功能包括用于执行虚拟机、中止虚拟机执行或者识别要分配给虚拟机的物理资源类型的功能。又一个实施例中，管理程序 401 在由管理程序 401 创建的虚拟机 406 内执行控制操作系统 405。又一个实施例中，控制操作系统 405 在被授权直接访问计算装置 100 上的物理资源的虚拟机 406 上执行。一些实施例中，计算装置 100a 上的控制操作系统 405a 可以通过管理程序 401a 和管理程序 401b 之间的通信与计算装置 100b 上的控制操作系统 405b 交换数据。这样，一个或多个计算装置 100 可以和一个或多个其他计算装置 100 交换有关处理器或资源池中可用的其他物理资

源的数据。在这些实施例的一个中，这种功能允许管理程序管理分布在多个物理计算装置上的资源池。在这些实施例的又一个中，多个管理程序管理在一个计算装置 100 上执行的一个或多个客户操作系统。

[0157] 在一个实施例中，控制操作系统 405 在被授权与至少一个客户操作系统 410 交互的虚拟机 406 上执行。又一个实施例中，客户操作系统 410 通过管理程序 401 和控制操作系统 405 通信，以请求访问盘或网络。仍在又一个实施例中，客户操作系统 410 和控制操作系统 405 可通过由管理程序 401 建立的通信信道通信，例如，通过由管理程序 401 提供的多个共享存储器页面。

[0158] 在一些实施例中，控制操作系统 405 包括用于直接与由计算装置 100 提供的网络硬件通信的网络后端驱动器。在这些实施例的一个中，网络后端驱动器处理来自至少一个客户操作系统 110 的至少一个虚拟机请求。其他实施例中，控制操作系统 405 包括用于与计算装置 100 上的存储元件通信的块后端驱动器。在这些实施例的一个中，块后端驱动器基于从客户操作系统 410 接收的至少一个请求从存储元件读写数据。

[0159] 在一个实施例，控制操作系统 405 包括工具堆栈 404。其他实施例中，工具堆栈 404 提供如下功能 :和管理程序 401 交互、和其他控制操作系统 405（例如位于第二计算装置 100b 上）通信，或者管理计算装置 100 上的虚拟机 406b、406c。又一个实施例中，工具堆栈 404 包括自定义应用，其用于向虚拟机群的管理员提供改进的管理功能。一些实施例中，工具堆栈 404 和控制操作系统 405 中的至少一个包括管理 API，其提供用于远程配置并控制计算装置 100 上运行的虚拟机 406 的接口。其他实施例中，控制操作系统 405 通过工具堆栈 404 和管理程序 401 通信。

[0160] 在一个实施例中，管理程序 401 在由管理程序 401 创建的虚拟机 406 内执行客户操作系统 410。又一个实施例中，客户操作系统 410 为计算装置 100 的用户提供对计算环境中的资源的访问。又一个实施例中，资源包括程序、应用、文档、文件、多个应用、多个文件、可执行程序文件、桌面环境、计算环境或对计算装置 100 的用户可用的其他资源。又一个实施例中，可通过多个访问方法将资源传送给计算装置 100，这些方法包括但不限于 :常规的直接在计算装置 100 上安装、通过应用流的方法传送给计算装置 100、将由在第二计算装置 100’上执行资源产生的并通过表示层协议传送给计算装置 100 的输出数据传送给计算装置 100、将通过在第二计算装置 100’上执行的虚拟机执行资源所产生的输出数据传送给计算装置 100、或者从连接到计算装置 100 的移动存储装置（例如 USB 设备）执行或者通过在计算装置 100 上执行的虚拟机执行并且产生输出数据。一些实施例中，计算装置 100 将执行资源所产生的输出数据传输给另一个计算装置 100’。

[0161] 在一个实施例中，客户操作系统 410 和该客户操作系统 410 在其上执行的虚拟机结合形成完全虚拟化虚拟机，该完全虚拟化虚拟机并不知道自己是虚拟机，这样的机器可称为“Domain U HVM（硬件虚拟机）虚拟机”。又一个实施例中，完全虚拟化机包括模拟基本输入 / 输出系统（BIOS）的软件以便在完全虚拟化机中执行操作系统。在又一个实施例中，完全虚拟化机可包括驱动器，其通过和管理程序 401 通信提供功能。这样的实施例中，驱动器可意识到自己在虚拟化环境中执行。又一个实施例中，客户操作系统 410 和该客户操作系统 410 在其上执行的虚拟机结合形成超虚拟化（paravirtualized）虚拟机，该超虚拟化虚拟机意识到自己是虚拟机，这样的机器可称为“Domain U PV 虚拟机”。又一个实施例中，

超虚拟化机包括完全虚拟化机不包括的额外驱动器。又一个实施例中，超虚拟化机包括如上所述的被包含在控制操作系统 405 中的网络后端驱动器和块后端驱动器。

[0162] 现参考图 4B，框图描述了系统中的多个联网计算装置的一个实施例，其中，至少一个物理主机执行虚拟机。总体而言，系统包括管理组件 404 和管理程序 401。系统包括多个计算装置 100、多个虚拟机 406、多个管理程序 401、多个管理组件（又称为工具堆栈 404 或者管理组件 404）以及物理资源 421、428。多个物理机器 100 的每一个可被提供为如上结合图 1E-1H 和图 4A 描述的计算装置 100。

[0163] 具体而言，物理盘 428 由计算装置 100 提供，存储至少一部分虚拟盘 442。一些实施例中，虚拟盘 442 和多个物理盘 428 相关联。在这些实施例的一个中，一个或多个计算装置 100 可以与一个或多个其他计算装置 100 交换有关处理器或资源池中可用的其他物理资源的数据，允许管理程序管理分布在多个物理计算装置上的资源池。一些实施例中，将虚拟机 406 在其上执行的计算装置 100 称为物理主机 100 或主机 100。

[0164] 管理程序在计算装置 100 上的处理器上执行。管理程序将对物理盘的访问量分配给虚拟盘。一个实施例中，管理程序 401 分配物理盘上的空间量。又一个实施例中，管理程序 401 分配物理盘上的多个页面。一些实施例中，管理程序提供虚拟盘 442 作为初始化和执行虚拟机 450 进程的一部分。

[0165] 一个实施例中，将管理组件 404a 称为池管理组件 404a。又一个实施例中，可以称为控制管理系统 405a 的管理操作系统 405a 包括管理组件。一些实施例中，将管理组件称为工具堆栈。在这些实施例的一个中，管理组件是上文结合图 4A 描述的工具堆栈 404。其他实施例中，管理组件 404 提供用户接口，用于从如管理员的用户接收要供应和 / 或执行的虚拟机 406 的标识。仍在其他实施例中，管理组件 404 提供用户接口，用于从如管理员的用户接收将虚拟机 406b 从一个物理机器 100 迁移到另一物理机器的请求。在进一步的实施例中，管理组件 404a 识别在其上执行所请求的虚拟机 406d 的计算装置 100b 并指示所识别的计算装置 100b 上的管理程序 401b 执行所识别的虚拟机，这样，可将管理组件称为池管理组件。

[0166] 现参考图 4C，描述了虚拟应用传送控制器或虚拟设备 450 的实施例。总体而言，上文结合图 2A 和 2B 描述的设备 200 的任何功能和 / 或实施例（例如应用传送控制器）可以部署在上文结合图 4A 和 4B 描述的虚拟化环境的任何实施例中。应用传送控制器的功能不是以设备 200 的形式部署，而是将该功能部署在诸如客户机 102、服务器 106 或设备 200 的任何计算装置 100 上的虚拟化环境 400 中。

[0167] 现在参考图 4C，描述了在服务器 106 的管理程序 401 上操作的虚拟设备 450 的实施例的框图。如图 2A 和 2B 的设备 200 一样，虚拟机 450 可以提供可用性、性能、卸载和安全的功能。对于可用性，虚拟设备可以执行网络第 4 层和第 7 层之间的负载平衡并执行智能服务健康监控。对于通过网络流量加速实现的性能增加，虚拟设备可以执行缓存和压缩。对于任何服务器的卸载处理，虚拟设备可以执行连接复用和连接池和 / 或 SSL 处理。对于安全，虚拟设备可以执行设备 200 的任何应用防火墙功能和 SSL VPN 功能。

[0168] 结合附图 2A 描述的设备 200 的任何模块可以虚拟化设备传送控制器 450 的形式被打包、组合、设计或构造，虚拟化设备传送控制器 450 可部署成在诸如流行的服务器这样的任何服务器上的虚拟化环境 300 或非虚拟化环境中执行的软件模块或组件。例如，可以

安装在计算装置上的安装包的形式提供虚拟设备。参考图 2A，可以将高速缓存管理器 232、策略引擎 236、压缩 238、加密引擎 234、分组引擎 240、GUI210、CLI212、壳服务 214 中的任一个设计和构成在计算装置和 / 或虚拟化环境 300 的任何操作系统上运行的组件或模块。虚拟化设备 400 不使用设备 200 的加密处理器 260、处理器 262、存储器 264 和网络堆栈 267，而是可使用虚拟化环境 400 提供的任何这些资源或者服务器 106 上以其他方式可用的这些资源。

[0169] 仍参考图 4C，简言之，任何一个或多个 vServer275A-275N 可以操作或执行在任意类型的计算装置 100（如服务器 106）的虚拟化环境 400 中。结合附图 2B 描述的设备 200 的任何模块和功能可以设计和构造成在服务器的虚拟化或非虚拟化环境中操作。可以将 vServer275、SSL VPN280、内网 UP282、交换装置 284、DNS286、加速装置 288、APP FW280 和监控代理中的任一个打包、组合、设计或构造成应用传送控制器 450 的形式，应用传送控制器 450 可部署成在装置和 / 或虚拟化环境 400 中执行的一个或多个软件模块或组件。

[0170] 一些实施例中，服务器可以在虚拟化环境中执行多个虚拟机 406a-406b，每个虚拟机运行虚拟应用传送控制器 450 的相同或不同实施例。一些实施例中，服务器可以在多核处理系统的一个核上执行一个或多个虚拟机上的一个或多个虚拟设备 450。一些实施例中，服务器可以在多处理器装置的每个处理器上执行一个或多个虚拟机上的一个或多个虚拟设备 450。

[0171] E. 提供多核架构的系统和方法

[0172] 根据摩尔定律，每两年集成电路上可安装的晶体管的数量会基本翻倍。然而，CPU 速度增加会达到一个稳定的水平（plateaus），例如，2005 年以来，CPU 速度在约 3.5-4GHz 的范围内。一些情况下，CPU 制造商可能不依靠 CPU 速度增加来获得额外的性能。一些 CPU 制造商会给处理器增加附加核以提供额外的性能。依靠 CPU 获得性能改善的如软件和网络供应商的产品可以通过利用这些多核 CPU 来改进他们的性能。可以重新设计和 / 或编写为单 CPU 设计和构造的软件以利用多线程、并行架构或多核架构。

[0173] 一些实施例中，称为 nCore 或多核技术的设备 200 的多核架构允许设备打破单核性能障碍并利用多核 CPU 的能力。前文结合图 2A 描述的架构中，运行单个网络或分组引擎。nCore 技术和架构的多核允许同时和 / 或并行地运行多个分组引擎。通过在每个核上运行分组引擎，设备架构利用附加核的处理能力。一些实施例中，这提供了高达七倍的性能改善和扩展性。

[0174] 图 5A 示出根据一类并行机制或并行计算方案（如功能并行机制、数据并行机制或基于流的数据并行机制）在一个或多个处理器核上分布的工作、任务、负载或网络流量的一些实施例。总体而言，图 5A 示出如具有 n 个核的设备 200’ 的多核系统的实施例，n 个核编号为 1 到 N。一个实施例中，工作、负载或网络流量可以分布在第一核 505A、第二核 505B、第三核 505C、第四核 505D、第五核 505E、第六核 505F、第七核 505G 等上，这样，分布位于所有 n 个核 505N（此后统称为核 505）或 n 个核中的两个或多个上。可以有多个 VIP275，每个运行在多个核中的相应的核上。可以有多个分组引擎 240，每个运行在多个核的相应的核。所使用任何方法可产生多个核中任一核上的不同的、变化的或类似的工作负载或性能级别 515。对于功能并行方法，每个核运行由分组引擎、VIP275 或设备 200 提供的多个功能的不同功能。在数据并行方法中，数据可基于接收数据的网络接口卡（NIC）或 VIP275 并行或分

布在核上。又一个数据并行方法中，可通过将数据流分布在每个核上而将处理分布在核上。

[0175] 图 5A 的进一步的细节中，一些实施例中，可以根据功能并行机制 500 将负载、工作或网络流量在多个核 505 间分布。功能并行机制可基于执行一个或多个相应功能的每个核。一些实施例中，第一核可执行第一功能，同时第二核执行第二功能。功能并行方法中，根据功能性将多核系统要执行的功能划分并分布到每个核。一些实施例中，可将功能并行机制称为任务并行机制，并且可在每个处理器或核对同一数据或不同数据执行不同进程或功能时实现。核或处理器可执行相同或不同的代码。一些情况下，不同的执行线程或代码可在工作时相互通信。可以进行通信以将数据作为工作流的一部分从一个线程传递给下一线程。

[0176] 一些实施例中，根据功能并行机制 500 将工作分布在核 505 上，可以包括根据特定功能分布网络流量，所述特定功能例如为网络输入 / 输出管理（NW I/O）510A、安全套接层（SSL）加密和解密 510B 和传输控制协议（TCP）功能 510C。这会产生基于所使用的功能量或功能级别的工作、性能或者计算负载 515。一些实施例中，根据数据并行机制 540 将工作分布在核 505 上可包括基于与特定的硬件或软件组件相关联的分布数据来分布工作量 515。一些实施例中，根据基于流的数据并行机制 520 将工作分布在核 505 上可包括基于上下文或流来分布数据，从而使得每个核上的工作量 515A-N 可以类似、基本相等或者相对平均分布。

[0177] 在功能并行方法的情况下，可以配置每个核来运行由设备的分组引擎或 VIP 提供的多个功能中的一个或多个功能。例如，核 1 可执行设备 200’ 的网络 I/O 处理，同时核 2 执行设备的 TCP 连接管理。类似地，核 3 可执行 SSL 卸载，同时核 4 可执行第 7 层或应用层处理和流量管理。每个核可执行相同或不同的功能。每个核可执行不只一个功能。任一核可运行结合附图 2A 和 2B 识别和 / 或描述的功能或其一部分。该方法中，核上的工作可以粗粒度或细粒度方式按功能划分。一些情况下，如图 5A 所示，按功能划分会使得不同核运行在不同的性能或负载级别 515。

[0178] 在功能并行方法的情况下，可以配置每个核来运行由设备的分组引擎提供的多个功能中的一个或多个功能。例如，核 1 可执行设备 200’ 的网络 I/O 处理，同时核 2 执行设备的 TCP 连接管理。类似地，核 3 可执行 SSL 卸载，同时核 4 可执行第 7 层或应用层处理和流量管理。每个核可执行相同或不同的功能。每个核可执行不只一个功能。任何核可运行结合附图 2A 和 2B 识别和 / 或描述的功能或其一部分。该方法中，核上的工作可以粗粒度或细粒度方式按功能划分。一些情况下，如图 5A 所示，按功能划分会使得不同核运行在不同的性能或负载级别。

[0179] 可以用任何结构或方案来分布功能或任务。例如，图 5B 示出用于处理与网络 I/O 功能 510A 相关联的应用和进程的第一核 Core1505A。一些实施例中，与网络 I/O 相关联的网络流量可以和特定的端口号相关联。因而，将具有与 NW I/O510A 相关联的端口目的地的发出和到来的分组导引给 Core1505A，该 Core1505A 专用于处理与 NW I/O 端口相关联的所有网络流量。类似的，Core2505B 专用于处理与 SSL 处理相关联的功能，Core4505D 可专用于处理所有 TCP 级处理和功能。

[0180] 虽然图 5A 示出如网络 I/O、SSL 和 TCP 的功能，也可将其他功能分配给核。这些其他功能可包括此处描述的任一或多个功能或操作。例如，结合图 2A 和 2B 描述的任何功能

可基于功能基础分布在核上。一些情况下，第一 VIP275A 可运行在第一核上，同时，具有不同配置的第二 VIP275B 可运行在第二核上。一些实施例中，每个核 505 可处理特定功能，这样每个核 505 可处理与该特定功能相关联的处理。例如，Core2505B 可处理 SSL 卸载，同时 Core4505D 可处理应用层处理和流量管理。

[0181] 其他实施例中，可根据任何类型或形式的数据并行机制 540 将工作、负载或网络流量分布在核 505 上。一些实施例中，可由每个核对分布式数据的不同片执行相同任务或功能来实现多核系统中的数据并行机制。一些实施例中，单个执行线程或代码控制对所有数据片的操作。其他实施例中，不同线程或指令控制操作，但是可执行相同代码。一些实施例中，从分组引擎、vServer(VIP)275A-C、网络接口卡(NIC)542D-E 和 / 或设备 200 上包括的或者与设备 200 相关联的任何其他网络硬件或软件的角度实现数据并行机制。例如，每个核可运行同样的分组引擎或 VIP 代码或配置但是在不同的分布式数据集上进行操作。每个网络硬件或软件结构可接收不同的、变化的或者基本相同量的数据，因而可以具有变化的、不同的或相对相同量的负载 515。

[0182] 在数据并行方法的情况下，可以基于 VIP、NIC 和 / 或 VIP 或 NIC 的数据流来划分和分布工作。在这些的方法的一个中，可通过使每个 VIP 在分布的数据集上工作来将多核系统的工作划分或者分布在 VIP 中。例如，可配置每个核运行一个或多个 VIP。网络流量可分布在处理流量的每个 VIP 的核上。在这些方法的又一个中，可基于哪个 NIC 接收网络流量来将设备的工作划分或分布在核上。例如，第一 NIC 的网络流量可被分布到第一核，同时第二 NIC 的网络流量可被分布给第二核。一些情况下，核可处理来自多个 NIC 的数据。

[0183] 虽然图 5A 示出了与单个核 505 相关联的单个 vServer，正如 VIP1275A、VIP2275B 和 VIP3275C 的情况。但是，一些实施例中，单个 vServer 可以与一个或者多个核 505 相关联。相反，一个或多个 vServer 可以与单个核 505 相关联。将 vServer 与核 505 关联可包括该核 505 处理与该特定 vServer 关联的所有功能。一些实施例中，每个核执行具有相同代码和配置的 VIP。其他实施例中，每个核执行具有相同代码但配置不同的 VIP。一些实施例中，每个核执行具有不同代码和相同或不同配置的 VIP。

[0184] 和 vServer 类似，NIC 也可以和特定的核 505 关联。许多实施例中，NIC 可以连接到一个或多个核 505，这样，当 NIC 接收或传输数据分组时，特定的核 505 处理涉及接收和传输数据分组的处理。一个实施例中，单个 NIC 可以与单个核 505 相关联，正如 NIC1542D 和 NIC2542E 的情况。其他实施例中，一个或多个 NIC 可以与单个核 505 相关联。但其他实施例中，单个 NIC 可以与一个或者多个核 505 相关联。这些实施例中，负载可以分布在一个或多个核 505 上，使得每个核 505 基本上处理类似的负载量。与 NIC 关联的核 505 可以处理与该特定 NIC 关联的所有功能和 / 或数据。

[0185] 虽然根据 VIP 或 NIC 的数据将工作分布在核上具有某种程度的独立性，但是，一些实施例中，这会造成如图 5A 的变化负载 515 所示的核的不平衡的使用。

[0186] 一些实施例中，可根据任何类型或形式的数据流将负载、工作或网络流量分布在核 505 上。在这些方法的又一个中，可基于数据流将工作划分或分布在多个核上。例如，客户机或服务器之间的经过设备的网络流量可以被分布到多个核中的一个核并且由其处理。一些情况下，最初建立会话或连接的核可以是该会话或连接的网络流量所分布的核。一些实施例中，数据流基于网络流量的任何单元或部分，如事务、请求 / 响应通信或来自客户机

上的应用的流量。这样，一些实施例中，客户机和服务器之间的经过设备 200’的数据流可以比其他方式分布的更均衡。

[0187] 在基于流的数据并行机制 520 中，数据分布和任何类型的数据流相关，例如请求 / 响应对、事务、会话、连接或应用通信。例如，客户机或服务器之间的经过设备的网络流量可以被分布到多个核中的一个核并且由其处理。一些情况下，最初建立会话或连接的核可以是该会话或连接的网络流量所分布的核。数据流的分布可以使得每个核 505 运行基本相等或相对均匀分布的负载量、数据量或网络流量。

[0188] 一些实施例中，数据流基于网络流量的任何单元或部分，如事务、请求 / 响应通信或源自客户机上的应用的流量。这样，一些实施例中，客户机和服务器之间的经过设备 200’的数据流可以比其他方式分布的更均衡。一个实施例中，可以基于事务或一系列事务分布数据量。一些实施例中，该事务可以是客户机和服务器之间的，其特征可以是 IP 地址或其他分组标识符。例如，核 1505A 可专用于特定客户机和特定服务器之间的事务，因此，核 1505A 上的负载 515A 可包括与特定客户机和服务器之间的事务相关联的网络流量。可通过将源自特定客户机或服务器的所有数据分组路由到核 1505A 来将网络流量分配给核 1505A。

[0189] 虽然可部分地基于事务将工作或负载分布到核，但是，其他实施例中，可基于每个分组的基础分配负载或工作。这些实施例中，设备 200 可拦截数据分组并将数据分组分配给负载量最小的核 505。例如，由于核 1 上的负载 515A 小于其他核 505B-N 上的负载 515B-N，所以设备 200 可将第一到来的数据分组分配给核 1505A。将第一数据分组分配给核 1505A 后，核 1505A 上的负载量 515A 与处理第一数据分组所需的处理资源量成比例增加。设备 200 拦截到第二数据分组时，设备 200 会将负载分配给核 4505D，这是由于核 4505D 具有第二少的负载量。一些实施例中，将数据分组分配给负载量最小的核可确保分布到每个核 505 的负载 515A-N 保持基本相等。

[0190] 其他实施例中，将一部分网络流量分配给特定核 505 的情况下，可以每单元为基础分配负载。上述示例说明以每分组为基础进行负载平衡。其他实施例中，可以基于分组数目分配负载，例如，将每 10 个、100 个或 1000 个分组分配给流量最少的核 505。分配给核 505 的分组数量可以是由应用、用户或管理员确定的数目，而且可以为大于零的任何数。仍在其他实施例中，基于时间指标分配负载，使得在预定时间段将分组分布到特定核 505。这些实施例中，可以在 5 毫秒内或者由用户、程序、系统、管理器或其他方式确定的任何时间段将分组分布到特定核 505。预定时间段过去后，在预定时间段内将时间分组传输给不同的核 505。

[0191] 用于将工作、负载或网络流量分布在一个或多个核 505 上的基于流的数据并行方法可包括上述实施例的任意组合。这些方法可以由设备 200 的任何部分执行，由在核 505 上执行的应用或者一组可执行指令执行，例如分组引擎，或者由在与设备 200 通信的计算装置上执行的任何应用、程序或代理执行。

[0192] 图 5A 所示的功能和数据并行机制计算方案可以任何方式组合，以产生混合并行机制或分布式处理方案，其包括功能并行机制 500、数据并行机制 540、基于流的数据并行机制 520 或者其任何部分。一些情况下，多核系统可使用任何类型或形式的负载平衡方案来将负载分布在一个或多个核 505 上。负载平衡方案可以和任何功能和数据平行方案或其

组合结合使用。

[0193] 图 5B 示出多核系统 545 的实施例，该系统可以是任何类型或形式的一个或多个系统、设备、装置或组件。一些实施例中，该系统 545 可被包括在具有一个或多个处理核 505A-N 的设备 200 内。系统 545 还可包括与存储器总线 556 通信的一个或多个分组引擎（PE）或分组处理引擎（PPE）548A-N。存储器总线可用于与一个或多个处理核 505A-N 通信。系统 545 还可包括一个或多个网络接口卡（NIC）552 和流分布器 550，流分布器还可与一个或多个处理核 505A-N 通信。流分布器 550 可包括接收侧调整器（Receiver Side Scaler-RSS）或接收侧调整（Receiver Side Scaling-RSS）模块 560。

[0194] 进一步参考图 5B，具体而言，一个实施例中，分组引擎 548A-N 可包括此处所述的设备 200 的任何部分，例如图 2A 和 2B 所述设备的任何部分。一些实施例中，分组引擎 548A-N 可包括任何下列的元件 ：分组引擎 240、网络堆栈 267、高速缓存管理器 232、策略引擎 236、压缩引擎 238、加密引擎 234、GUI210、CLI212、壳服务 214、监控程序 216 以及能够从数据总线 556 或一个或多个核 505A-N 中的任一个接收数据分组的其他任何软件和硬件元件。一些实施例中，分组引擎 548A-N 可包括一个或多个 vServer275A-N 或其任何部分。其他实施例中，分组引擎 548A-N 可提供以下功能的任意组合 ：SSL VPN280、内部网 IP282、交换 284、DNS286、分组加速 288、APPFW280、如由监控代理 197 提供的监控、和作为 TCP 堆栈关联的功能、负载平衡、SSL 卸载和处理、内容交换、策略评估、高速缓存、压缩、编码、解压缩、解码、应用防火墙功能、XML 处理和加速以及 SSL VPN 连接。

[0195] 一些实施例中，分组引擎 548A-N 可以与特定服务器、用户、客户或网络关联。分组引擎 548 与特定实体关联时，分组引擎 548 可处理与该实体关联的数据分组。例如，如果分组引擎 548 与第一用户关联，那么该分组引擎 548 将对由第一用户产生的分组或者目的地址与第一用户关联的分组进行处理和操作。类似地，分组引擎 548 可选择不与特定实体关联，使得分组引擎 548 可对不是由该实体产生的或目的是该实体的任何数据分组进行处理和以其他方式进行操作。

[0196] 一些实例中，可将分组引擎 548A-N 配置为执行图 5A 所示的任何功能和 / 或数据并行方案。这些实例中，分组引擎 548A-N 可将功能或数据分布在多个核 505A-N 上，从而使得分布是根据并行机制或分布方案的。一些实施例中，单个分组引擎 548A-N 执行负载平衡方案，其他实施例中，一个或多个分组引擎 548A-N 执行负载平衡方案。一个实施例中，每个核 505A-N 可以与特定分组引擎 548 关联，使得可以由分组引擎执行负载平衡。在该实施例中，负载平衡可要求与核 505 关联的每个分组引擎 548A-N 和与核关联的其他分组引擎通信，使得分组引擎 548A-N 可共同决定将负载分布在何处。该过程的一个实施例可包括从每个分组引擎接收对于负载的投票的仲裁器。仲裁器可部分地基于引擎投票的持续时间将负载分配给每个分组引擎 548A-N，一些情况下，还可基于与在引擎关联的核 505 上的当前负载量相关联的优先级值来将负载分配给每个分组引擎 548A-N。

[0197] 核上运行的任何分组引擎可以运行于用户模式、内核模式或其任意组合。一些实施例中，分组引擎作为在用户空间或应用空间中运行的应用或程序来操作。这些实施例中，分组引擎可使用任何类型或形式的接口来访问内核提供的任何功能。一些实施例中，分组引擎操作于内核模式中或作为内核的一部分来操作。一些实施例中，分组引擎的第一部分操作于用户模式中，分组引擎的第二部分操作于内核模式中。一些实施例中，第一核上的第

一分组引擎执行于内核模式中，同时，第二核上的第二分组引擎执行于用户模式中。一些实施例中，分组引擎或其任何部分对 NIC 或其任何驱动器进行操作或者与其联合操作。

[0198] 一些实施例中，存储器总线 556 可以是任何类型或形式的存储器或计算机总线。虽然在图 5B 中描述了单个存储器总线 556，但是系统 545 可包括任意数量的存储器总线 556。一个实施例中，每个分组引擎 548 可以和一个或者多个单独的存储器总线 556 相关联。

[0199] 一些实施例中，NIC552 可以是此处所述的任何网络接口卡或机制。NIC552 可具有任意数量的端口。NIC 可设计并构造成连接到任何类型和形式的网络 104。虽然示出单个 NIC552，但是，系统 545 可包括任意数量的 NIC552。一些实施例中，每个核 505A-N 可以与一个或多个单个 NIC552 关联。因而，每个核 505 可以与专用于特定核 505 的单个 NIC552 关联。核 505A-N 可包括此处所述的任何处理器。此外，可根据此处所述的任何核 505 配置来配置核 505A-N。另外，核 505A-N 可具有此处所述的任何核 505 功能。虽然图 5B 示出七个核 505A-G，但是系统 545 可包括任意数量的核 505。具体而言，系统 545 可包括 N 个核，其中 N 是大于零的整数。

[0200] 核可具有或使用被分配或指派用于该核的存储器。可将存储器视为该核的专有或本地存储器并且仅有该核可访问该存储器。核可具有或使用共享的或指派给多个核的存储器。该存储器可被视为由不只一个核可访问的公共或共享存储器。核可使用专有或公共存储器的任何组合。通过每个核的单独的地址空间，消除了使用同一地址空间的情况下的一些协调级别。利用单独的地址空间，核可以对核自己的地址空间中的信息和数据进行工作，而不用担心与其他核冲突。每个分组引擎可以具有用于 TCP 和 / 或 SSL 连接的单独存储器池。

[0201] 仍参考图 5B，上文结合图 5A 描述的核 505 的任何功能和 / 或实施例可以部署在上文结合图 4A 和 4B 描述的虚拟化环境的任何实施例中。不是以物理处理器 505 的形式部署核 505 的功能，而是将这些功能部署在诸如客户机 102、服务器 106 或设备 200 的任何计算装置 100 的虚拟化环境 400 内。其他实施例中，不是以设备或一个装置的形式部署核 505 的功能，而是将该功能部署在任何布置的多个装置上。例如，一个装置可包括两个或多个核，另一个装置可包括两个或多个核。例如，多核系统可包括计算装置的集群、服务器群或计算装置的网络。一些实施例中，不是以核的形式部署核 505 的功能，而是将该功能部署在多个处理器上，例如部署多个单核处理器上。

[0202] 一个实施例中，核 505 可以为任何形式或类型的处理器。一些实施例中，核的功能可以基本类似此处所述的任何处理器或中央处理单元。一些实施例中，核 505 可包括此处所述的任何处理器的任何部分。虽然图 5A 示出 7 个核，但是，设备 200 内可以有任意 N 个核，其中 N 是大于 1 的整数。一些实施例中，核 505 可以安装在公用设备 200 内，其他实施例中，核 505 可以安装在彼此通信连接的一个或多个设备 200 内。一些实施例中，核 505 包括图形处理软件，而其他实施例中，核 505 提供通用处理能力。核 505 可彼此物理靠近地安装和 / 或可彼此通信连接。可以用以物理方式和 / 或通信方式耦合到核的任何类型和形式的总线或子系统连接核，用于向核、从核和 / 或在核之间传输数据。

[0203] 尽管每个核 505 可包括用于与其他核通信的软件，一些实施例中，核管理器(未示出)可有助于每个核 505 之间的通信。一些实施例中，内核可提供核管理。核可以使用各种接口机制彼此接口或通信。一些实施例中，可以使用核到核的消息传送来在核之间通信，比

如，第一核通过连接到核的总线或子系统向第二核发送消息或数据。一些实施例中，核可通过任何种类或形式的共享存储器接口通信。一个实施例中，可以存在在所有核中共享的一个或多个存储器单元。一些实施例中，每个核可以具有和每个其他核共享的单独存储器单元。例如，第一核可具有与第二核的第一共享存储器，以及与第三核的第二共享存储器。一些实施例中，核可通过任何类型的编程或 API（如通过内核的函数调用）来通信。一些实施例中，操作系统可识别并支持多核装置，并提供用于核间通信的接口和 API。

[0204] 流分布器 550 可以是任何应用、程序、库、脚本、任务、服务、进程或在任何类型或形式的硬件上执行的任何类型和形式的可执行指令。一些实施例中，流分布器 550 可以是用于执行此处所述任何操作和功能的任何电路设计或结构。一些实施例中，流分布器分布、转发、路由、控制和 / 或管理多个核 505 上的数据和 / 或在核上运行的分组引擎或 VIP 的分布。一些实施例中，可将流分布器 550 称为接口主装置（interface master）。一个实施例中，流分布器 550 包括在设备 200 的核或处理器上执行的一组可执行指令。又一个实施例中，流分布器 550 包括在与设备 200 通信的计算机器上执行的一组可执行指令。一些实施例中，流分布器 550 包括在如固件的 NIC 上执行的一组可执行指令。其他实施例，流分布器 550 包括用于将数据分组分布在核或处理器上的软件和硬件的任何组合。一个实施例中，流分布器 550 在至少一个核 505A-N 上执行，而在其他实施例中，分配给每个核 505A-N 的单独的流分布器 550 在相关联的核 505A-N 上执行。流分布器可使用任何类型和形式的统计或概率算法或决策来平衡多个核上的流。可以将如 NIC 的设备硬件或内核设计或构造成支持 NIC 和 / 或核上的顺序操作。

[0205] 系统 545 包括一个或多个流分布器 550 的实施例中，每个流分布器 550 可以与处理器 505 或分组引擎 548 关联。流分布器 550 可包括允许每个流分布器 550 和在系统 545 内执行的其他流分布器 550 通信的接口机制。一个实例中，一个或多个流分布器 550 可通过彼此通信确定如何平衡负载。该过程的操作可以基本与上述过程类似，即将投票提交给仲裁器，然后仲裁器确定哪个流分布器 550 应该接收负载。其他实施例中，第一流分布器 550’可识别所关联的核上的负载并基于任何下列标准确定是否将第一数据分组转发到所关联的核：所关联的核上的负载大于预定阈值；所关联的核上的负载小于预定阈值；所关联的核上的负载小于其他核上的负载；或者可以用于部分基于处理器上的负载量来确定将数据分组转发到何处的任何其他指标。

[0206] 流分布器 550 可以根据如此处所述的分布、计算或负载平衡方法而将网络流量分布在核 505 上。一个实施例中，流分布器可基于功能并行机制分布方案 550、数据并行机制负载分布方案 540、基于流的数据并行机制分布方案 520 或这些分布方案的任意组合或用于将负载分布在多个处理器上的任何负载平衡方案来分布网络流量。因而，流分布器 550 可通过接收数据分组并根据操作的负载平衡或分布方案将数据分组分布在处理器上而充当负载分布器。一个实施例中，流分布器 550 可包括用于确定如何相应地分布分组、工作或负载的一个或多个操作、函数或逻辑。又一个实施例中，流分布器 550 可包括可识别与数据分组关联的源地址和目的地址并相应地分布分组的一个或多个子操作、函数或逻辑。

[0207] 一些实施例中，流分布器 550 可包括接收侧调整（RSS）网络驱动器模块 560 或将数据分组分布在一个或多个核 505 上的任何类型和形式的可执行指令。RSS 模块 560 可以包括硬件和软件的任意组合。一些实施例中，RSS 模块 560 和流分布器 550 协同工作以将数

据分组分布在核 505A-N 或多处理器网络中的多个处理器上。一些实施例中，RSS 模块 560 可在 NIC552 中执行，其他实施例中，可在核 505 的任何一个上执行。

[0208] 一些实施例中，RSS 模块 560 使用微软接收侧调整（RSS）方法。一个实施例中，RSS 是微软可扩展网络主动技术（Microsoft Scalable Networking initiative technology），其使得系统中的多个处理器上的接收处理是平衡的，同时保持数据的顺序传送。RSS 可使用任何类型或形式的哈希方案来确定用于处理网络分组的核或处理器。

[0209] RSS 模块 560 可应用任何类型或形式的哈希函数，如 Toeplitz 哈希函数。哈希函数可应用到哈希类型值或者任何值序列。哈希函数可以是任意安全级别的安全哈希或者是以其他方式加密。哈希函数可使用哈希关键字（hash key）。关键字的大小取决于哈希函数。对于 Toeplitz 哈希，用于 IPv6 的哈希关键字大小为 40 字节，用于 IPv4 的哈希关键字大小为 16 字节。

[0210] 可以基于任何一个或多个标准或设计目标设计或构造哈希函数。一些实施例中，可使用为不同的哈希输入和不同哈希类型提供均匀分布的哈希结果的哈希函数，所述不同哈希输入和不同哈希类型包括 TCP/IPv4、TCP/IPv6、IPv4 和 IPv6 头部。一些实施例中，可使用存在少量桶时（例如 2 个或 4 个）提供均匀分布的哈希结果的哈希函数。一些实施例中，可使用存在大量桶时（例如 64 个桶）提供随机分布的哈希结果的哈希函数。在一些实施例中，基于计算或资源使用水平来确定哈希函数。在一些实施例中，基于在硬件中实现哈希的难易度来确定哈希函数。在一些实施例中，基于用恶意的远程主机发送将全部哈希到同一桶中的分组的难易度来确定哈希函数。

[0211] RSS 可从任意类型和形式的输入来产生哈希，例如值序列。该值序列可包括网络分组的任何部分，如网络分组的任何头部、域或载荷或其一部分。一些实施例中，可将哈希输入称为哈希类型，哈希输入可包括与网络分组或数据流关联的任何信息元组，例如下面的类型：包括至少两个 IP 地址和两个端口的四元组、包括任意四组值的四元组、六元组、二元组和 / 或任何其他数字或值序列。以下是可由 RSS 使用的哈希类型示例：

[0212] - 源 TCP 端口、源 IP 版本 4（IPv4）地址、目的 TCP 端口和目的 IPv4 地址的四元组。

[0213] - 源 TCP 端口、源 IP 版本 6（IPv6）地址、目的 TCP 端口和目的 IPv6 地址的四元组。

[0214] - 源 IPv4 地址和目的 IPv4 地址的二元组。

[0215] - 源 IPv6 地址和目的 IPv6 地址的二元组。

[0216] - 源 IPv6 地址和目的 IPv6 地址的二元组，包括对解析 IPv6 扩展头部的支持。

[0217] 哈希结果或其任何部分可用于识别用于分布网络分组的核或实体，如分组引擎或 VIP。一些实施例中，可向哈希结果应用一个或者多个哈希位或掩码。哈希位或掩码可以是任何位数或字节数。NIC 可支持任意位，例如 7 位。网络堆栈可在初始化时设定要使用的实际位数。位数介于 1 和 7 之间，包括端值。

[0218] 可通过任意类型和形式的表用哈希结果来识别核或实体，例如通过桶表（bucket table）或间接表（indirection table）。一些实施例中，用哈希结果的位数来索引表。哈希掩码的范围可有效地限定间接表的大小。哈希结果的任何部分或哈希结果自身可用于索引间接表。表中的值可标识任何核或处理器，例如通过核或处理器标识符来标识。一些实

施例中，表中标识多核系统的所有核。其他实施例中，表中标识多核系统的一部分核。间接表可包括任意多个桶，例如 2 到 128 个桶，可以用哈希掩码索引这些桶。每个桶可包括标识核或处理器的索引值范围。一些实施例中，流控制器和 / 或 RSS 模块可通过改变间接表来重新平衡网络负载。

[0219] 一些实施例中，多核系统 575 不包括 RSS 驱动器或 RSS 模块 560。在这些实施例的一些中，软件操控模块（未示出）或系统内 RSS 模块的软件实施例可以和流分布器 550 共同操作或者作为流分布器 550 的一部分操作，以将分组引导到多核系统 575 中的核 505。

[0220] 一些实施例中，流分布器 550 在设备 200 上的任何模块或程序中执行，或者在多核系统 575 中包括的任何一个核 505 和任一装置或组件上执行。一些实施例中，流分布器 550’可在第一核 505A 上执行，而在其他实施例中，流分布器 550”可在 NIC552 上执行。其他实施例中，流分布器 550’的实例可在多核系统 575 中包括的每个核 505 上执行。该实施例中，流分布器 550’的每个实例可和流分布器 550’的其他实例通信以在核 505 之间来回转发分组。存在这样的状况，其中，对请求分组的响应不是由同一核处理的，即第一核处理请求，而第二核处理响应。这些情况下，流分布器 550’的实例可以拦截分组并将分组转发到期望的或正确的核 505，即流分布器 550’可将响应转发到第一核。流分布器 550’的多个实例可以在任意数量的核 505 或核 505 的任何组合上执行。

[0221] 流分布器可以响应于任一个或多个规则或策略而操作。规则可识别接收网络分组、数据或数据流的核或分组处理引擎。规则可识别和网络分组有关的任何类型和形式的元组信息，例如源和目的 IP 地址以及源和目的端口的四元组。基于所接收的匹配规则所指定的元组的分组，流分布器可将分组转发到核或分组引擎。一些实施例中，通过共享存储器和 / 或核到核的消息传输将分组转发到核。

[0222] 虽然图 5B 示出了在多核系统 575 中执行的流分布器 550，但是，一些实施例中，流分布器 550 可执行在位于远离多核系统 575 的计算装置或设备上。这样的实施例中，流分布器 550 可以和多核系统 575 通信以接收数据分组并将分组分布在一个或多个核 505 上。一个实施例中，流分布器 550 接收以设备 200 为目的地的数据分组，向所接收的数据分组应用分布方案并将数据分组分布到多核系统 575 的一个或多个核 505。一个实施例中，流分布器 550 可以被包括在路由器或其他设备中，这样路由器可以通过改变与每个分组关联的元数据而以特定核 505 为目的地，从而每个分组以多核系统 575 的子节点为目的地。这样的实施例中，可用 CISCO 的 vn-tag 机制来改变或标记具有适当元数据的每个分组。

[0223] 图 5C 示出包括一个或多个处理核 505A-N 的多核系统 575 的实施例。简言之，核 505 中的一个可被指定为控制核 505A 并可用作其他核 505 的控制平面 570。其他核可以是次级核，其工作于数据平面，而控制核提供控制平面。核 505A-N 共享全局高速缓存 580。控制核提供控制平面，多核系统中的其他核形成或提供数据平面。这些核对网络流量执行数据处理功能，而控制核提供对多核系统的初始化、配置和控制。

[0224] 仍参考图 5C，具体而言，核 505A-N 以及控制核 505A 可以是此处所述的任何处理器。此外，核 505A-N 和控制核 505A 可以是能在图 5C 所述系统中工作的任何处理器。另外，核 505A-N 可以是此处所述的任何核或核组。控制核可以是与其他核不同类型的核或处理器。一些实施例中，控制核可操作不同的分组引擎或者具有与其他核的分组引擎配置不同的分组引擎。

[0225] 每个核的存储器的任何部分可以被分配给或者用作核共享的全局高速缓存。简而言之，每个核的每个存储器的预定百分比或预定量可用作全局高速缓存。例如，每个核的每个存储器的50%可用作或分配给共享全局高速缓存。也就是说，所示实施例中，除了控制平面核或核1以外的每个核的2GB可用于形成28GB的共享全局高速缓存。例如通过配置服务而配置控制平面可确定用于共享全局高速缓存的存储量(the amount of memory)。一些实施例中，每个核可提供不同的存储量供全局高速缓存使用。其他实施例中，任一核可以不提供任何存储器或不使用全局高速缓存。一些实施例中，任何核也可具有未分配给全局共享存储器的存储器中的本地高速缓存。每个核可将网络流量的任意部分存储在全局共享高速缓存中。每个核可检查高速缓存来查找要在请求或响应中使用的任何内容。任何核可从全局共享高速缓存获得内容以在数据流、请求或响应中使用。

[0226] 全局高速缓存580可以是任意类型或形式的存储器或存储元件，例如此处所述的任何存储器或存储元件。一些实施例中，核505可访问预定的存储量(即32GB或者与系统575相当的任何其他存储量)。全局高速缓存580可以从预定的存储量分配而来，同时，其余的可用存储器可在核505之间分配。其他实施例中，每个核505可具有预定的存储量。全局高速缓存580可包括分配给每个核505的存储量。该存储量可以字节为单位来测量，或者可用分配给每个核505的存储器百分比来测量。因而，全局高速缓存580可包括来自与每个核505关联的存储器的1GB存储器，或者可包括和每个核505关联的存储器的20%或一半。一些实施例，只有一部分核505提供存储器给全局高速缓存580，而在其他实施例，全局高速缓存580可包括未分配给核505的存储器。

[0227] 每个核505可使用全局高速缓存580来存储网络流量或缓存数据。一些实施例中，核的分组引擎使用全局高速缓存来缓存并使用由多个分组引擎所存储的数据。例如，图2A的高速缓存管理器和图2B的高速缓存功能可使用全局高速缓存来共享数据以用于加速。例如，每个分组引擎可在全局高速缓存中存储例如HTML数据的响应。操作于核上的任何高速缓存管理器可访问全局高速缓存来将高速缓存响应提供给客户请求。

[0228] 一些实施例中，核505可使用全局高速缓存580来存储端口分配表，其可用于部分基于端口确定数据流。其他实施例中，核505可使用全局高速缓存580来存储地址查询表或任何其他表或列表，流分布器可使用这些表来确定将到来的数据分组和发出的数据分组导向何处。一些实施例中，核505可以读写高速缓存580，而其他实施例中，核505仅从高速缓存读或者仅向高速缓存写。核可使用全局高速缓存来执行核到核通信。

[0229] 可以将全局高速缓存580划分成各个存储器部分，其中每个部分可专用于特定核505。一个实施例中，控制核505A可接收大量的可用高速缓存，而其他核505可接收对全局高速缓存580的变化的访问量。

[0230] 一些实施例中，系统575可包括控制核505A。虽然图5C将核1505A示为控制核，但是，控制核可以是设备200或多核系统中的任何一个核。此外，虽然仅描述了单个控制核，但是，系统575可包括一个或多个控制核，每个控制核对系统有某种程度的控制。一些实施例中，一个或多个控制核可以各自控制系统575的特定方面。例如，一个核可控制决定使用哪种分布方案，而另一个核可确定全局高速缓存580的大小。

[0231] 多核系统的控制平面可以是将一个核指定并配置成专用的管理核或者作为主核。控制平面核可对多核系统中的多个核的操作和功能提供控制、管理和协调。控制平面核可

对多核系统中的多个核上存储器系统的分配和使用提供控制、管理和协调，这包括初始化和配置存储器系统。一些实施例中，控制平面包括流分布器，用于基于数据流控制数据流到核的分配以及网络分组到核的分配。一些实施例中，控制平面核运行分组引擎，其他实施例中，控制平面核专用于系统的其他核的控制和管理。

[0232] 控制核 505A 可对其他核 505 进行某种级别的控制，例如，确定将多少存储器分配给每个核 505，或者确定应该指派哪个核来处理特定功能或硬件 / 软件实体。一些实施例中，控制核 505A 可以对控制平面 570 中的这些核 505 进行控制。因而，控制平面 570 之外可存在不受控制核 505A 控制的处理器。确定控制平面 570 的边界可包括由控制核 505A 或系统 575 中执行的代理维护由控制核 505A 控制的核的列表。控制核 505A 可控制以下的任一个 :核初始化、确定核何时不可用、一个核出故障时将负载重新分配给其他核 505、决定实现哪个分布方案、决定哪个核应该接收网络流量、决定应该给每个核分配多少高速缓存、确定是否将特定功能或元件分布到特定核、确定是否允许核彼此通信、确定全局高速缓存 580 的大小以及对系统 575 内的核的功能、配置或操作的任何其他确定。

[0233] F. 用于对多个下一跳进行策略路由的系统和方法

[0234] 本公开的系统和方法的实施例涉及对多个下一跳进行策略路由。传统路由算法，即使是高度复杂的路由算法，对于输出流量可能从未实现真正的负载平衡。应用请求可能倾向一个服务器而非另一个服务器，这导致对一个目的 IP 的、比另一个目的 IP 更多的请求。使用对这些应用不可知的路由器，可能不会适当地或持续地平衡流量。

[0235] 因此，在本系统的一个实施例中，设备，例如本文讨论的设备 200 的任何实施例，可以识别产生请求的应用，并且相应地应用一个或多个策略来路由和平衡这些请求。在一些实施例中，该设备可以包括流量过滤器以识别流量的部分，例如事务或会话，并在更细粒度的基础来路由该流量。在许多实施例中，该设备可以包括用于在应用敏感基础上监控一个或多个链路的健康的功能。例如，该设备可以监控链路延迟并且相应地路由来自要求高延迟或低延迟链路的应用的请求。在许多实施例中，该设备可以提供网络地址转换（NAT）、端口地址转换（PAT）和逆向网络地址转换以提供透明的双向路由和转发。

[0236] 因此，本文公开的系统和方法的实施例可通过使用多个链路来方便地提供可靠性，其中每个链路提供对其他链路的备份，以用于 24 小时 /7 天可用性。这些系统和方法可以通过平衡对多个链路的并发使用来改善网络吞吐量、提供在网络基础设施中的更好的投资回报、通过消除流量激增和连通性故障时间来提供更有效的资源利用率以及通过在多个网络路径之间智能分布负载来改善应用的性能和服务质量。

[0237] 现参考图 6A，描述了用于智能策略路由（Policy-based routing，PBR）的系统的实施例。总的来说，设备 200 可以是在客户机 / 服务器与由多个 ISP700A-700N 提供的多个下一跳中间的装置。该设备可以包括策略引擎 236，分组过滤器或分类器 645 和负载平衡器 275。策略引擎可以响应于一个或多个策略路由（PBR）策略 636 而操作，所述策略指定用于匹配或识别由分组过滤器 645 为任何输入或输出分组所确定的分组特征 645 的一个或多个参数 647。作为分组特征与 PBR 规则的参数 647 的匹配结果，PBR 可以从多个下一跳中选择一个下一跳。该设备的负载平衡器 275 可以使用任何类型和形式的负载平衡算法或方案在多个下一跳之间负载平衡该选择。该设备可包括一个或多个监控器 650，用于监控多个下一跳或链路的状态或健康。

[0238] 该设备的实施例可以经由多个链路与互联网通信或连接到互联网，例如经由由互联网服务提供者（ISP A-N）700A-700N 提供的任何链路。下一跳是路由器或服务器，或其他网络装备和 / 或到这样的路由器、服务器或网络装备的链路。“下一跳”识别信息分组接下来可以或将会被发送到哪个中间装置，例如，到哪个路由器、服务器或网络装置。接着，该下一跳装置可以再次决定该信息将会被发送到哪里（例如，自先前的下一跳的下一跳）。该分组被从一个路由器、服务器或网络装置传递到下一个路由器、服务器或网络装置，直到该分组到达目的装置的物理网络。在一些实施例中，下一跳被认为是网络服务 270。ISP 可以是提供、支持或维护链路或对部分互联网的访问的任何类型和形式的互联网服务提供者。

[0239] 策略引擎 236 可以包括本文之前描述的策略引擎的任何实施例，包括策略引擎 195 的任何实施例。策略引擎可以被设计为、构建成和 / 或适合于配置或接收输入以配置一个或多个 PBR 策略 636。策略引擎可以被设计为、构建成和 / 或适合于响应于一个或多个 PBR 策略 636 来执行或运行。

[0240] PBR 策略可以指定要采取的路由动作。PBR 策略可以基于一个或多个参数 647 来指定要采取的路由动作。PBR 策略可以指定动作和规则。该规则可识别或指定将会采取该策略的特定动作的一组条件。该规则的这组条件可以基于将该策略的一个或多个参数与分组的相应特征 645 的匹配或评估。该策略可以使用表达式或策略语言来配置 PBR 策略的参数和 / 或以逻辑表达式评估参数。参数 647 可以包括但不限于语法或语言结构以识别对于下列特征的任何一个或多个参数：

[0241] • 源 IP（单独地或通过范围）

[0242] • 目的 IP（单独地或通过范围）

[0243] • 源端口（单独地或通过范围）

[0244] • 目的端口（单独地或通过范围）

[0245] • 应用层、会话层、传输层和 / 或互联网层协议

[0246] • 虚拟局域网（VLAN）

[0247] • 源介质访问控制（MAC）地址

[0248] • 接口（例如该设备的多个接口中的第一物理接口）

[0249] PBR 策略可以经由下一跳允许或拒绝分组访问。PBR 策略可以指定对于该路由动作的预定的下一跳。PBR 策略可以指定下一跳的枚举列表，该设备可以从该列表中选择一个以向其转发该分组。在一些实施例中，该枚举列表可以是按优先级排序的。在一些实施例中，PBR 策略可以指定预定的 ISP，通过其进行该路由动作。在一些实施例中，PBR 策略可以指定用于负载平衡多个下一跳（例如对于该路由动作的下一跳）的负载平衡方案、算法或类型。

[0250] 由设备 200 接收的、可对其应用 PBR 策略的分组可以包括输入分组。由设备 200 接收的、可对其应用 PBR 策略的分组可以包括输出分组。每个分组可具有与经由 PBR 策略的参数可配置或表示的参数对应的一个或多个特征。这些特征包括但不限于关于下列内容的数据和 / 或信息：

[0251] • 源 IP（单独地或通过范围）

[0252] • 目的 IP（单独地或通过范围）

[0253] • 源端口（单独地或通过范围）

[0254] • 目的端口(单独地或通过范围)

[0255] • 应用层、会话层、传输层和 / 或互联网层协议

[0256] • 虚拟局域网(VLAN)

[0257] • 源介质访问控制(MAC)地址

[0258] • 接口(例如该设备的多个接口的第一物理接口)

[0259] 过滤器 640 可包括在例如设备 200 的装置上执行的应用、程序、库、过程、服务、任务或任何类型和形式的可执行指令。过滤器可以是虚拟服务器的部分。过滤器可以是分组引擎的部分。过滤器可以是策略引擎的部分。过滤器可以是本文描述的设备的任何组件、模块或单元的部分。在一些实施例中,过滤器是该设备的拦截和处理网络分组的组件的部分。在一些实施例中,过滤器是网络分组的拦截器。

[0260] 过滤器可以检查、解析或以其他方式处理分组以识别或确定分组的一个或多个特征 645。过滤器可以从分组中任一层的任一头部或字段来识别或确定该分组的特征,例如该分组的传输层或网络层的任何头部或字段。过滤器可以从分组的任一协议的任一头部或字段来识别或确定该分组的特征。过滤器可以从分组的有效载荷中的任何数据来识别或确定该分组的特征。过滤器可以将每个分组的特征 645 存储在存储或内存单元中,例如数据结构或对象。

[0261] 过滤器可以包括分类器以根据一个或多个分组特征 645 和 / 或根据一个或多个 PBR 策略对网络流量中的分组进行分类。过滤器可以根据分组的特征 645 对这些分组进行分类。在一些实施例中,过滤器可以根据哪些特征匹配哪些策略的参数来对分组进行分类。一旦进行了分类或在进行分类之后,根据一个或多个策略,可以将流量转发到多个下一跳中指定的下一跳,或者转发到沿到目的地路径的中间节点。

[0262] 过滤器和策略引擎可以联合工作以根据分组的特征对该分组应用 PBR 策略。在一些实施例中,过滤器和策略引擎是同一组件或模块(例如该设备的分组引擎)的部分。过滤器可将分组的分组特征 645 提供或传送给策略引擎 236 以应用到 PBR 策略 636。过滤器可以将策略引擎的 PBR 策略应用到分组的特征 645。策略引擎可以确定任何分组特征是否匹配任一 PBR 的任何参数。策略引擎可以用来自相应分组特征的值替换 PBR 策略的表达式中参数的变量。策略引擎可以基于具有来自相应分组特征的值的参数来评估任何规则。策略引擎可以基于具有来自相应分组特征的值的参数来对策略或其规则的任何逻辑表达式进行求值。策略引擎可以确定对策略的表达式的求值结果,该结果例如可以是布尔值。

[0263] 策略引擎和 / 或 PBR 策略可以对监控器进行响应。在一些实施例中,策略引擎和 / 或 PBR 策略可以确定下一跳是否是健康的或可用的,并且基于该确定,该 PBR 策略可以采取路由动作。在一些实施例中,基于经由监控确定的健康或可用性,策略引擎可以相应地对与下一跳的健康或状态对应的 PBR 策略的任何规则的表达式进行求值。

[0264] 该设备可以包括一个或多个负载平衡器,例如负载平衡器虚拟服务器 275,包括本文所述的虚拟服务器的任何实施例。负载平衡器可以负载平衡多个下一跳。负载平衡器可以响应于一个或多个 PBR 策略和 / 或过滤来进行负载平衡。负载平衡器可以基于分组的协议类型来执行一种类型的负载平衡。在一些实施例中,负载平衡器执行基于请求的负载平衡来从多个下一跳中选择下一跳。在一些实施例中,负载平衡器对于每个新的传输层连接执行基于连接的负载平衡,以从多个下一跳中选择下一跳。在一些实施例中,负载平衡器执

行基于时间的负载平衡来从多个下一跳中选择下一跳。在一些实施例中，负载平衡器基于由分组的目的互联网协议地址或源互联网协议地址的其中一个所确定的持久性来选择下一跳。在一些实施例中，该设备或 vServer 基于所选择的下一跳来转换分组的互联网协议地址。

[0265] 监控器或监控代理可以包括报告服务（例如下一跳或 ISP700A-700N 的链路）的性能或操作特征的任何程序、脚本、守护进程或其他计算例程。监控代理可以与链路或下一跳通信或者以预定的频率进行通信。在一些实施例中，监控代理可以与链路或下一跳使用请求 / 回复消息传送机制或协议。在一些实施例中，该一个或多个监控代理确定该一个或多个下一跳或 ISP 用于对请求进行响应的响应时间。

[0266] 在一些实施例中，监控代理确定对 UDP 请求的网络服务的可用性。在这些实施例的一个中，该代理使用“UDP echo”命令来向下一跳发送数据报，接收来自下一跳响应的数据报，并且基于该数据报的往返时间确定响应时间。在这些实施例的又一个中，监控代理验证来自下一跳的响应包括了预期内容并且不包含错误。

[0267] 可以由设备 200 来给监控代理分配权重。权重可包括整数、小数或任何其他数字指示器。在一些实施例中，用户可以配置与给定监控代理对应的权重。在一些实施例中，可以给所有的监控代理分配相同的权重。在其他实施例中，可以给多个监控代理中的每一个分配不同的权重。可以基于任何指示相对重要性的标准来给监控器分配权重，所述标准包括但不限于所监控的服务的重要性、该监控机制的可靠性以及监控的频率。

[0268] 监控器可以基于任何统计或操作或性能特征来确定任一下一跳的健康或状态。监控器可以基于经由下一跳的带宽使用量来确定该下一跳的健康或状态。监控器可以基于经由下一跳的连接数量来确定该下一跳的健康或状态。监控器可以基于使用下一跳的用户数量来确定该下一跳的健康或状态。监控器可以基于经过下一跳的分组数量来确定该下一跳的健康或状态。监控器可以基于该设备和下一跳之间的往返时间来确定该下一跳的健康或状态。监控器可以基于该设备和下一跳之间的延迟来确定该下一跳的健康或状态。监控器可以基于正在监控的统计、操作或性能特征的任何相应阈值来确定健康或状态。

[0269] 在该系统运行的实施例的进一步细节中，对于每个基于策略的规则（PBR）可以配置多个下一跳。这样做可以实现互联网链路冗余，有效的负载平衡、应用特定的负载平衡和优先级排序。可以使用一个或多个负载平衡算法和策略在多个下一跳之间应用负载平衡。这些算法和策略（有时被称为负载平衡方法或 LB 方法）可以指示如何在多个链路之间分布流量负载。通过网络地址转换（NAT）和反向 NAT，这些方法可跨越多个互联网连接透明地平衡输入和输出流量。在一个实施例中，负载平衡算法可以定义该设备可用来选择下一跳以向其转发分组的标准或阈值。在另外的实施例中，当所选择的下一跳达到所配置的标准或阈值，该设备可以选择不同的下一跳。例如，在一个这样的实施例中，第一跳可以被配置有 45ms 的延迟阈值。该设备可以在这一跳上路由流量，直到延迟增加到超过 45ms，然后可以经由不同的下一跳来路由流量。这些特征可以被狭义或广义地限定，这取决于具体需求。例如，延迟阈值可以通过例如，低、中或高的范围来限定，或者通过特定值来限定。

[0270] 而且，在一些实施例中，可以以不同的粒度级别将负载平衡算法应用到单个分组、分组流、事务、请求、连接、会话或预定时间段中的一组分组。例如，在输入分组是 HTTP 或 HTTPS 请求的一个实施例中，可以在基于请求的粒度级别上应用负载平衡算法。可以对于每

个 HTTP 或 HTTPS 请求执行下一跳选择，而不考虑所使用的 TCP 连接的数量。在又一个实施例中，可以在基于连接的粒度级别上应用负载平衡算法。例如，可以对于每个新的 TCP 连接执行下一跳选择。在又一个实施例中，可以在基于时间或基于会话的粒度级别上应用负载平衡算法。例如，对于 UDP 或其他时间敏感的连接，可以在连接被开启时进行下一跳选择，并且可以在预定的时间段使用该下一跳选择。这在一些实施例中可以被称为会话。当这段时间到期时，该会话可以被删除或关闭，接着新的会话被创建并且再次执行下一跳选择。即使分组仍来自同一客户机也可以这样做。因此，可以以在简单的基于目的地的路由中不可能的方式来对 UDP 会话进行负载平衡。

[0271] 在一个实施例中，负载平衡算法或方法可以包括基于循环的转发，其中该设备可以将分组转发到每个链路而不考虑任何特定链路上的负载。在又一个实施例中，负载平衡算法或方法可包括基于最小响应的转发。在该算法中，该设备可选择具有最小平均响应时间的链路。在另外的实施例中，可以根据应用、根据服务或根据协议来确定最小平均响应时间。例如，该设备可包括一个或多个探测器或监控器以确定经由各种服务的不同链路的响应时间。在某些情况下，也许由于互联网服务提供者的流量整形或过滤，经由一个链路，HTTP 流量可能具有低响应时间，而经由同一链路，对等 (P2P) 或比特流（bit torrent，BT）流量具有高响应时间。类似地，另一个链路可能完全相反，对于对等流量具有低响应时间而对于 HTTP 流量具有高响应时间。通过基于每协议或每应用监控响应时间，该设备可以基于最小平均响应时间以及基于分组的特定协议或应用来智能地路由流量。

[0272] 在又一个实施例中，负载平衡算法或方法可包括基于哈希结果的转发。网络分组的头部和 / 或主体中的各种数据字段和串可以与哈希算法一起使用来确定与下一跳列表中的一个下一跳对应的索引值。哈希算法可能是高效的，而且相应地，可以高速缓存哈希值，这通过消除对于会话或事务中多个分组进行下一跳选择的必要而降低了设备的 CPU 使用率。例如，在使用基于目的地的哈希的一个这样的实施例中，该设备可以计算对属于特定子网或目的地址的请求的哈希并选择下一跳。可以高速缓存该哈希和所选择的下一跳，使得将来的具有相同目的地址（并且相应地，具有相同的哈希值）的分组可以被路由到所选择的下一跳而不需要另外的选择或负载平衡。在类似的实施例中，对于基于源的哈希，可以利用分组的源地址。在又一个实施例中，可以对源和目的地址都进行哈希，这导致更精细的基于哈希的路由。在许多实施例中，哈希分组或执行基于会话的路由可以允许会话持久性。会话持久性（其中会话的所有分组都经由相同的下一跳传输）可能是非常期望的，尤其是在高丢失环境中。重传和乱序分组如果经由各种不同的链路发送则可能导致不可预测的行为。因此，通过上文讨论的各种哈希方法，包括多个分组的单个会话可以经由相同的下一跳而被发送，这减少或消除了这种不可预测的行为。

[0273] 在又一个实施例中，负载平衡算法或方法可包括基于最少带宽的转发。可以在预定时间段内监控被转发到每个下一跳的流量，例如以每秒兆比特（Mbps）为单位，并且该设备可基于哪一跳具有最小当前流量带宽值来选择下一跳。在类似的实施例中，负载平衡算法或方法可包括基于最少或最小分组数的转发。该设备可以监控预定时间段内转发到每个下一跳的分组的数量，并且可以基于该时间段内哪一跳具有最少分组数来选择下一跳。在其他实施例中，该设备可以监控下一跳或每个下一跳的设备或计算装置的一个或多个状态，包括 CPU 使用率、内存可用性以及响应时间，并且响应于哪个装置具有这些值中一个或

多个的最小值来转发分组到下一跳。

[0274] 在一些实施例中，通过监控链路和 / 或下一链路的装置的网络参数，该设备可以基于链路健康来进行负载平衡和转发。因此，如果链路发生故障，该设备可以将流量转发到备份下一跳。与传统链路健康监控(其中，链路仅可以“正常运行(up)”或“发生故障(down)”)不同，在许多实施例中，该设备可以基于每应用来监控下一跳装置的健康。例如，该设备可以监控状态，包括延迟和响应时间、分组丢失、内存要求、CPU 使用率或下一跳装置的其他特征，其对于 TCP、HTTP、SSL、SIP、FTP、RADIUS、DNS、ICMP、POP3、LDAP、MySQL、SNMP、SMTP、RTSP 或任何其他协议或应用具有单独的数据。因此，例如，下一跳装置可能对于 HTTP 通信被认为“正常运行”而对于 FTP 通信被认为“发生故障”。相应地，可以基于每应用来做出智能路由和负载平衡决策。

[0275] 在许多实施例中，转发或路由分组可以包括网络地址转换。这在使用多个下一跳的系统中可能是复杂的，其中设备可以在主下一跳发生故障时将分组转发到备份下一跳。例如，在简单的基于目的地的路由系统中，设备可以连接到两个互联网服务提供者，第一互联网服务提供者具有地址 203.10.10.0，而第二互联网服务提供者具有地址 201.1.1.0。经由第一 ISP 从远程地址接收的流量可以被转换成 203.10.10.0 并由该设备转发给客户机。客户机相应地可以对 203.10.10.0 进行回复，并且该设备可以经由基于目的地的路由正确地将该流量路由到第一 ISP。然而，如果第一 ISP 发生故障，该设备可能不能够容易地经由第二 IP 来重新路由该流量，因为该链路可能没有合适的 NAT 表。通过使用本文所讨论的用于智能地址转换的更高级的下一跳选择和负载平衡算法，该设备可以避免该问题。

[0276] 现参考图 6B，示出了用于进行智能策略路由(PBR)的方法的实施例的流程图。总的来说，在步骤 600，由设备接收的分组可以被传递经过各种分组过滤器。这些过滤器可包括一组特征，使得可以根据匹配的特征组来处理和转发匹配的分组。在一个实施例中，基于策略的路由动作可以被标记为允许(ALLOW)或拒绝(DENY)。可以响应于分组匹配或未匹配指定的特征来执行该动作。例如，在一个这样的实施例中，如果分组匹配在任何所配置的被标记为允许的策略路由规则中的所有参数，则在步骤 608 该设备可以尝试使用在该策略中配置的下一跳来路由该分组。否则，正常转发该分组。在又一个这样的实施例中，在步骤 602，如果分组匹配任何所配置的被标记为拒绝的策略路由规则，则不对该分组进行策略路由而是正常转发该分组。在又一个这样的实施例中，如果在步骤 600，分组不匹配任何所配置的策略路由规则，或者在步骤 602，该策略中配置的下一跳发生故障，那么该分组可以被正常转发。在另外的实施例中，在步骤 606 正常转发分组可包括对于分组执行基于目的地的路由。

[0277] 在进一步的细节中，在步骤 600，由该设备接收输入或输出分组。可以从在该设备与下一跳或 ISP 中间的客户机接收该分组。可以从在该设备与下一跳或 ISP 中间的服务器接收该分组。可以从下一跳装置接收该分组，用于向客户机或服务器进行传输。该设备的过滤器可以识别或确定一个或多个分组特征。过滤器和 / 或策略引擎可以确定是否有任何 PBR 策略或规则具有匹配该一个或多个分组特征的参数。在一些实施例中，过滤器和 / 或策略引擎确定没有任何 PBR 策略匹配该一个或多个分组特征。在一些实施例中，过滤器和 / 或策略引擎确定存在单个 PBR 策略匹配该一个或多个分组特征。在一些实施例中，过滤器和 / 或策略引擎确定有多个 PBR 策略匹配该一个或多个分组特征。

[0278] 如果没有 PBR 策略或规则匹配分组或分组的特征，则该设备可以确定不对该分组应用任何策略路由。该设备可以对分组使用所谓的正常路由 606。在一些实施例中，经由正常路由，不采用 PBR 并且将分组转发到网络接口。在一些实施例中，经由正常路由，不采用 PBR 并且根据任何标准的或传统的下一跳路由或转发来对分组进行转发。

[0279] 在步骤 602，如果存在 PBR 策略或规则匹配分组或分组的特征，则该设备例如通过策略引擎来根据策略确定对该分组的动作。在多个匹配 PBR 策略的一些实施例中，该设备按顺序或优先级次序选择要应用的其中一个 PBR 策略。在多个匹配 PBR 策略的一些实施例中，该设备按顺序或优先级次序采用要应用的所有 PBR 策略。该动作可以是允许访问、使用或转发该分组到下一跳。该动作可以是根据下一跳的状态来允许访问、使用或转发该分组到该下一跳。该动作可以是根据策略路由拒绝访问、使用或转发该分组。该动作可以是拒绝访问、使用或转发该分组到特定的下一跳。该动作可以是拒绝访问、使用或转发该分组到下一跳。该动作可以是拒绝访问、使用或转发该分组到特定的下一跳。该动作可以是使用正常路由。如果该动作是拒绝并且 / 或者下一跳发生故障，则该设备可以转到或使用正常路由 606。

[0280] 在步骤 602，PBR 可以确定分组的下一跳。例如，如果该动作是允许并且下一跳正常，则在步骤 604，该设备可以对该分组设置下一跳。该设备可以例如经由分组引擎或 vServer 修改分组以将目的或路由 IP 地址设置为下一跳。可以通过 PBR 来指定或确定该下一跳。可以通过负载平衡器来指定或确定该下一跳。在一些实施例中，该设备可以负载平衡到该下一跳或提供该下一跳的 ISP 的多条链路。在一些实施例中，该设备可以负载平衡多个下一跳以选择正常运行的一个下一跳，并且允许分组访问、使用或被转发到该下一跳。

[0281] 在步骤 604，该设备可以基于分组的协议类型来执行一种类型的负载平衡。该设备可执行基于请求的负载平衡来从多个下一跳中选择下一跳。该设备可以对于每个新的传输层连接执行基于连接的负载平衡，以从多个下一跳中选择下一跳。该设备可执行基于时间的负载平衡来从多个下一跳中选择下一跳。该设备可基于由分组的任一 IP 地址（例如目的互联网协议地址和 / 或源互联网协议地址）所确定的持久性来选择下一跳。该设备可基于由分组的任一元组确定的持久性来选择下一跳。

[0282] 在步骤 608，该设备可以基于分组的下一跳设置经由链路向 ISP 传输或转发分组。在一些实施例中，可以根据基于策略和 / 或基于负载平衡选择的下一跳来传输分组。在一些实施例中，可以根据正常路由来传输分组，在正常路由中，不根据基于策略和 / 或基于负载平衡的选择来设置或改变分组的下一跳。该设备可以对源 IP 地址进行网络地址转换以经由下一跳链路或 ISP 正确地路由该分组。

[0283] 相应地，本文所述的方法和系统为智能策略路由提供这样的支持 :定义用于进行路由决策的复杂策略以及根据所配置的策略或规则过滤并解析流量以进行重定向。而且，这些方法和系统允许高级的下一跳负载平衡，其通过智能和高效的负载平衡算法，例如循环、最少分组数、最小带宽、基于哈希的、最小响应、定制负载或其他，来在多个互联网链路之间进行有效的链路负载平衡。此外，通过在每应用基础上监控下一跳健康，这些方法和系统可响应于单独的网络特征来提供有效的路由。该系统和方法也可以在主链路运行时有效地使用备份链路带宽，以及在主链路发生故障时提供即时故障转移（fail over），从而提供链路冗余和效率。最后，该系统和方法还使用基于源 IP、目的 IP 和组合的源 / 目的 IP 哈希

的持久性转发来提供基于持久性的路由。

[0284] 为了强调该系统的其中一些好处，提供了几个示例。提供这些示例仅用于解释说明的目的而非旨在进行任何限制。本领域技术人员可以容易地理解下列示例可以被扩展或应用到相似情景而不脱离本公开的保护范围。图 7A 所示的是使用基于端口的路由策略的系统的实施例的框图。如图所示，客户机 102 可以经由设备 200 连接到多个互联网服务提供者 700A-700B(总的称为 ISP700)。在一些实施例中，这些可以是到单个 ISP 的两个连接，例如两个有线调制解调器或 T1 链路；可以是不同类型的两个连接，例如 T1 线路和 ISDN 线路；或者可以是到两个不同服务提供者的连接。这样做可确保服务提供者发生故障情况下的可靠性，或者充分利用不同网络架构的优势。例如，如图 7A 所示，通过使用两个 ISP700，客户机 102a 经由设备 200 可以具有到服务器 106a 的两条独立网络路径。ISP A700A 经由第一网络 104 和第二网络 104”两者进行连接，并且 ISP B700B 经由第三网络 104’和第二网络 104”两者进行连接。尽管被称为不同网络，但在许多实施例中，这些可以表示网络分段或节点。到服务器的每条路径可以具有不同的网络特征，这取决于长度、地理位置、在 ISP 处的装备或其他中间节点、跳数等等。尽管传统的转发系统可以基于通用质量确定一条“最好”路径，但这对于不同类型的流量可能并不是最好的路径。例如，一条路径可能是高延迟，但高带宽，由此，该路径对于低延迟网络电话 (VoIP) 流量可能是较差的但对于 FTP 文件传输可能是非常好的。通过执行上文所述的基于策略的转发方法，设备 200 可以针对每种类型的流量充分利用最好的网络路径。

[0285] 例如，如图 7A 所示，客户机可以发送两个业务、请求或分组的流，第一个流定向到目的端口 i (实线)而第二个流定向到同一目的服务器 106a 处的目的端口 j (虚线)。设备 200 可以响应于应用到每个传输的策略，经由不同的下一跳定向这两个流。这样做可以分享负载并且还可以在其中一个服务提供者发生故障的情况下确保可靠性。

[0286] 现参考图 7B，示出了在链路故障期间，使用基于端口的路由策略的系统的实施例的框图。如图所示，如果 ISP700 (例如 ISP A700A) 失效或发生故障，则设备 200 可立即通过第二 ISP B700B 来路由去往 ISP A700A 的流量。在应用于该流量的策略中，ISP B 可以被配置作为将会发送到 ISP A 的分组的备份下一跳；反过来，ISP A 可以被配置作为将会发送到 ISP B 的分组的备份下一跳。

[0287] 现参考图 7C，示出了使用智能的基于目的地的路由策略的系统的实施例的框图。与上文讨论的基于端口的路由类似，客户机可能希望分别以目的 IP i 和目的 IP j 与两个目的服务器 106a-106b 通信。通过对该流量应用策略，设备 200 可以经由不同的 ISP 路由不同的流量以进行负载平衡并有效使用带宽。不同于传统的基于目的地的转发，该设备可以使用上文所讨论的负载平衡算法和基于哈希的转发来在任何时间动态地选择要使用哪个 ISP，而不必考虑最终的目的地。

[0288] 现参考图 7D，示出了在链路故障期间，使用智能的基于目的地的路由策略的系统的实施例的框图。与如图 7B 所示的类似，在应用于该流量的策略中，ISP B 可以被配置作为将会发送到 ISP A 的分组的备份下一跳；反过来，ISP A 可以被配置作为将会发送到 ISP B 的分组的备份下一跳。相应地，当服务提供者链路失效或发生故障时，可立即通过另一个服务提供者来路由网络流量。

[0289] 现参考图 7E，示出了使用到一个目的地的具有不同网络特征的多条路径的系统的

实施例的框图。如图所示，设备 200 可以连接到大量的服务提供者，该大量的服务提供者可以连接到不同的网络分段。因此，对于单个客户机到 y 条链路中的 x 条链路，其中 1<x<y，该设备可能需要提供反向网络地址转换。例如，一个国家可能有 5 个主要的互联网服务提供者是可用的。这些服务提供者的其中三个可能具有到第二个国家的直接链路，而它们中的其他两个可能仅有到第二国家的间接链路，该链路经过第三或第四国家。因此，设备 200 可确定对于要连接到服务器 106a 的客户机 102a，三个 ISP700A-700C 是快速的、低延迟链路，而 ISP700D-700n 是具有几个额外跳的高延迟链路。设备 200 可以使用这样的策略：为每个分组配置多个下一跳，并通过经由 ISP700A-700C 的快速链路进行负载平衡。每个快速链路可以被配置为其他链路的备份，并且在一些实施例中，慢速链路 700D-700n 可以被配置为在每个快速链路同时发生故障情况下进一步的备份。在类似的环境中，第二服务器 106b（未示出）可能位于第三或第四个国家，并且由此，对于这个特定的服务器，链路 700D-700n 可能比链路 700A-700C 更快。设备 200 可以具有多个有效的策略，这些策略经由对于特定流量最有效的路径来路由和负载平衡流量。

[0290] 应理解，上文所述的系统可提供这些组件的任意多个或每一个并且这些组件可以在独立机器上提供，或者在一些实施例中，可在分布式系统的多个机器上提供。可以使用编程和 / 或工程技术将上文所描述的系统和方法实现为方法、装置或产品以提供软件、固件、硬件或上述的任何组合。此外，上述系统和方法可作为在一件或多件产品上实现或在其中实现的一个或多个计算机可读程序而被提供。此处使用的术语“产品”旨在包括从一个或多个计算机可读的装置、固件、可编程逻辑、存储器装置（例如，EEPROM、ROM、PROM、RAM、SRAM 等）、硬件（例如，集成电路芯片、现场可编程门阵列（FPGA）、专用集成电路（ASIC）等）、电子装置、计算机可读的非易失存储单元（例如，CD-ROM、软盘、硬盘等）可访问的或嵌入其中的代码或逻辑。所述产品可以是从经由网络传输线、无线传输介质、通过空间传播的信号、无线电波、红外信号等提供对计算机可读程序的访问的文件服务器可访问的。所述产品可以是闪存卡或磁带。所述产品包括硬件逻辑以及嵌入在计算机可读介质中由处理器执行的软件或可编程代码。通常，计算机可读程序可以任何编程语言来实现，如 LISP、PERL、C、C++、C#、PROLOG，或者诸如 JAVA 的任何字节码语言。软件程序可以作为目标代码被存储在一件或多件产品上或其中。

[0291] 已经描述了对于多个下一跳提供策略路由的方法和系统的某些实施例，对本领域技术人员而言，显而易见可以使用包含本发明的概念的其他实施例。

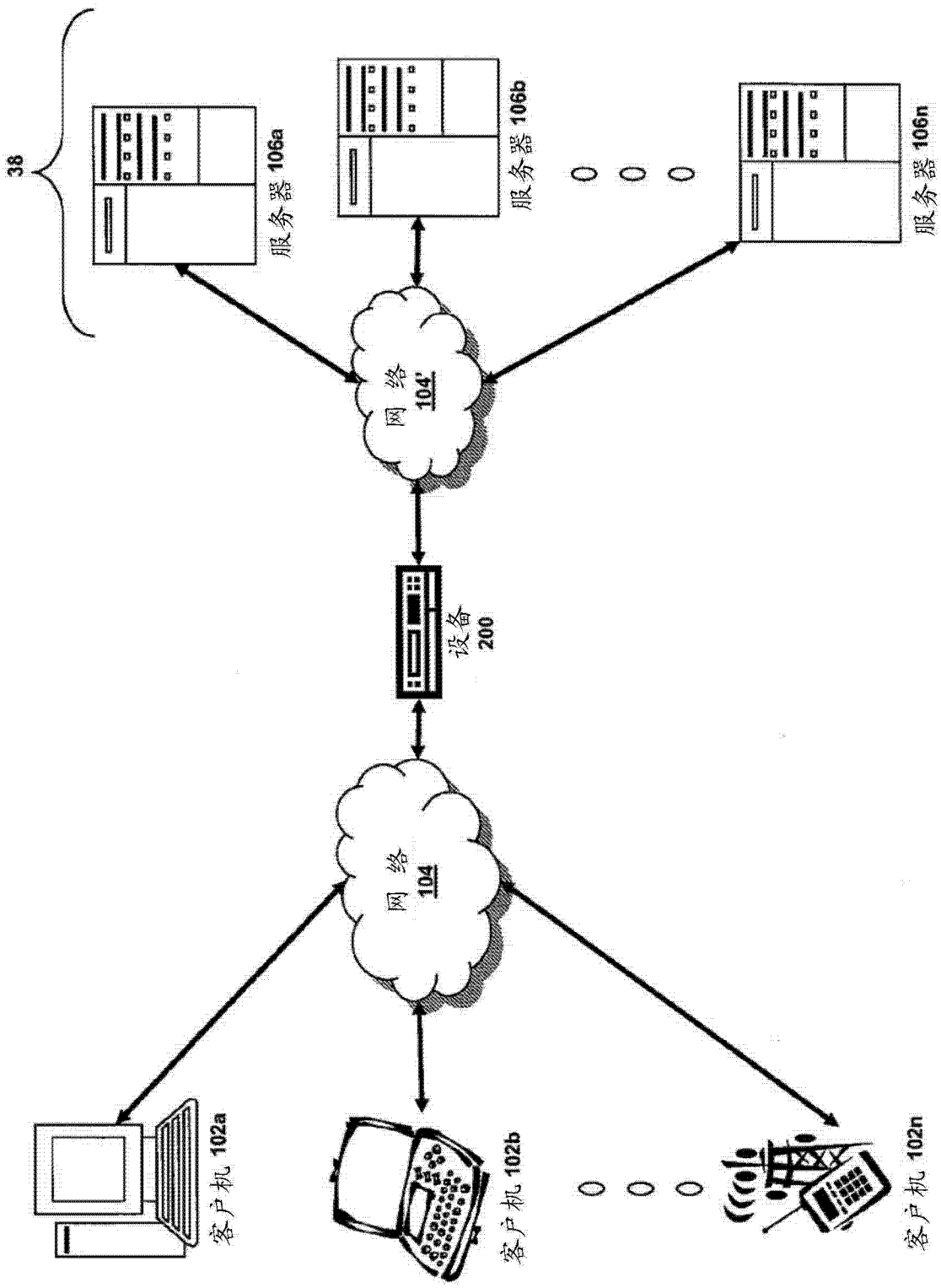

38
服务器 106a
服务器 106b
服务器 106n
网络 104'
设备 200
网络 104
客户机 102a
客户机 102b
客户机 102n

图 1A

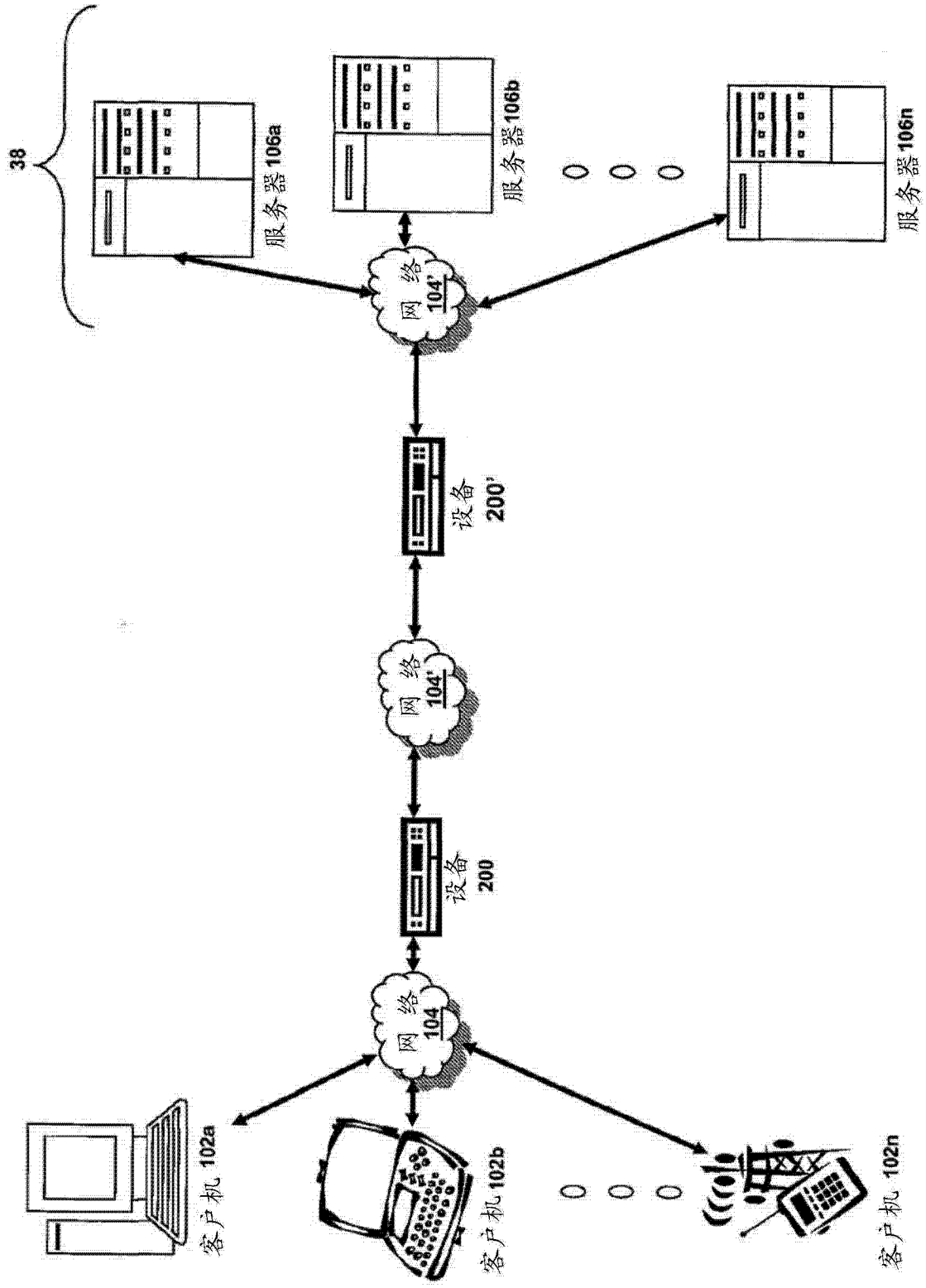

38
服务器106a
服务器106b
服务器106n
网络104'
设备200'
网络104'
设备200
网络104
客户机102a
客户机102b
客户机102n

图 1B

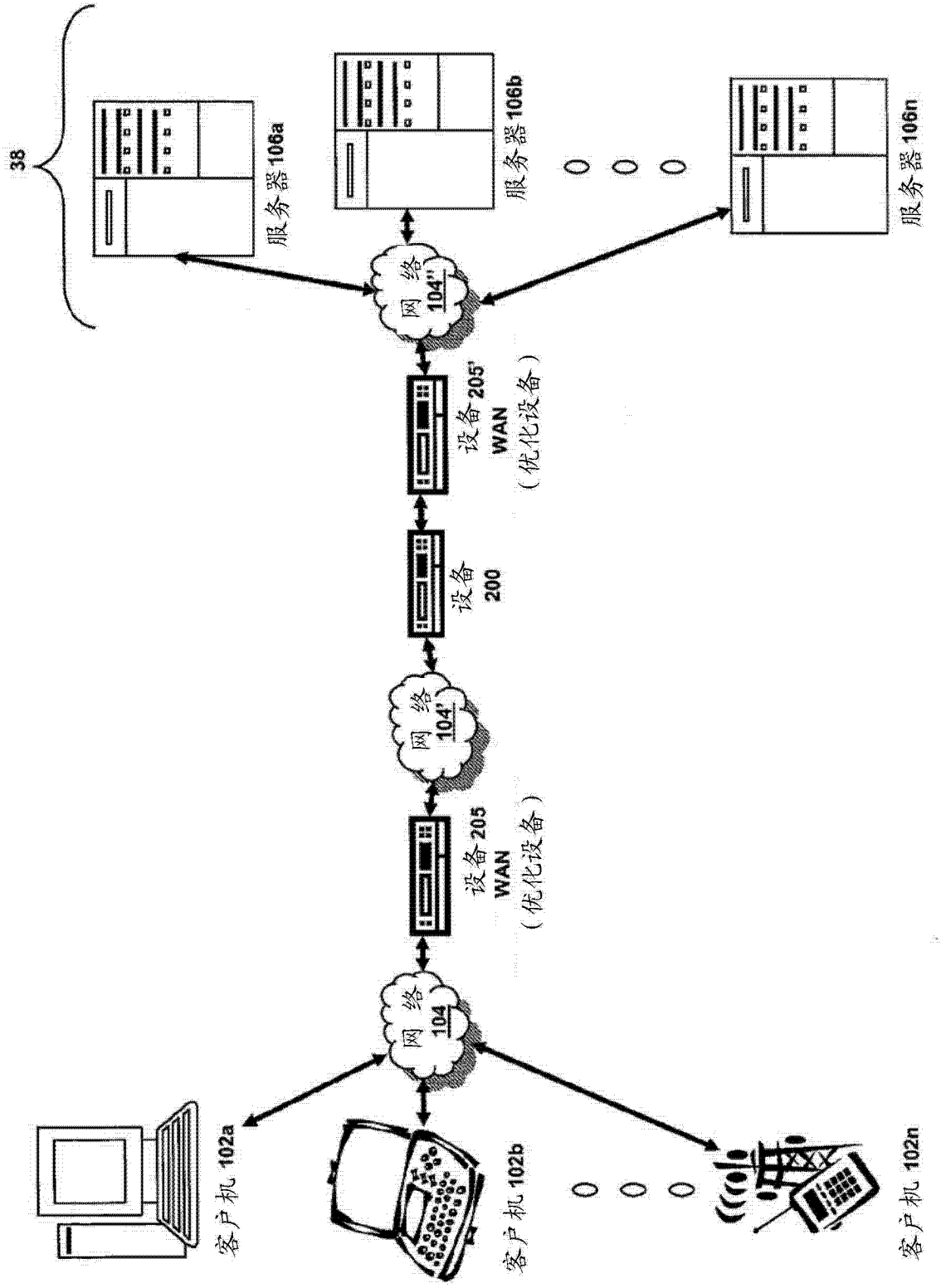

38
服务器106a
服务器106b
服务器106n
网络104''
设备205' WAN（优化设备）
设备200
网络104'
设备205 WAN（优化设备）
网络104
客户机102a
客户机102b
客户机102n

图 1C

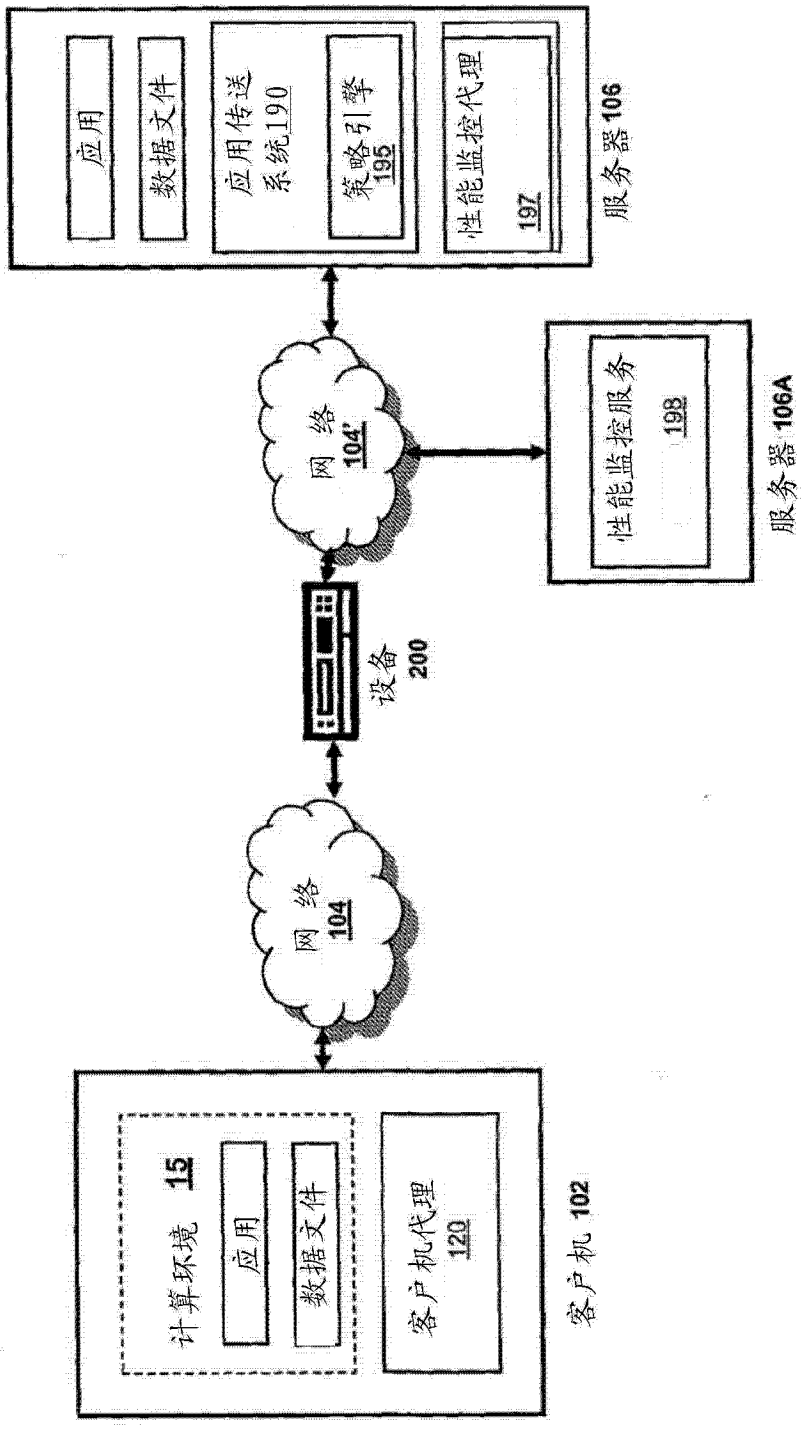

应用
数据文件
应用传送系统190
策略引擎195
性能监控代理197
服务器106
网络104'
性能监控服务198
服务器106A
设备200
网络104
计算环境15
应用
数据文件
客户机代理120
客户机102

图 1D

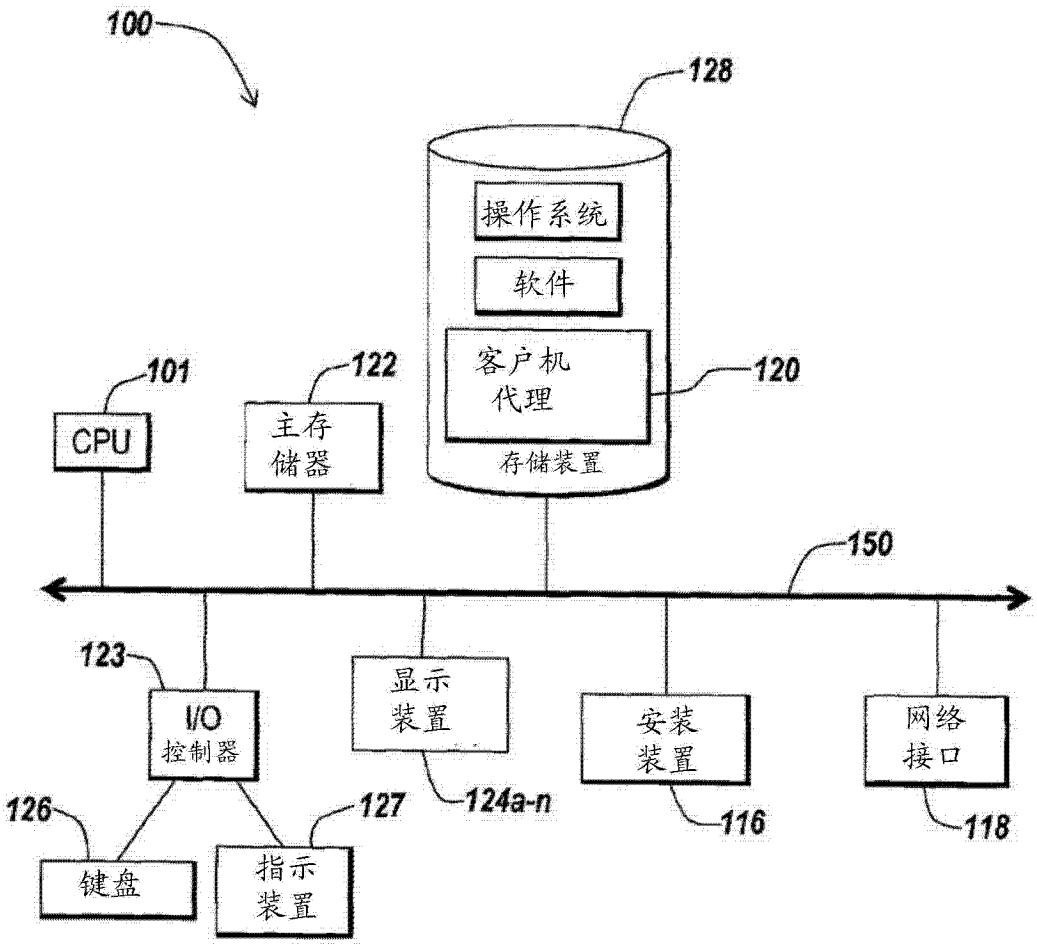

100
128
操作系统
软件
客户机
代理
120
存储装置
101
CPU
122
主存
储器
150
123
I/O
控制器
显示
装置
124a-n
安装
装置
116
网络
接口
118
126
键盘
127
指示
装置

图 1E

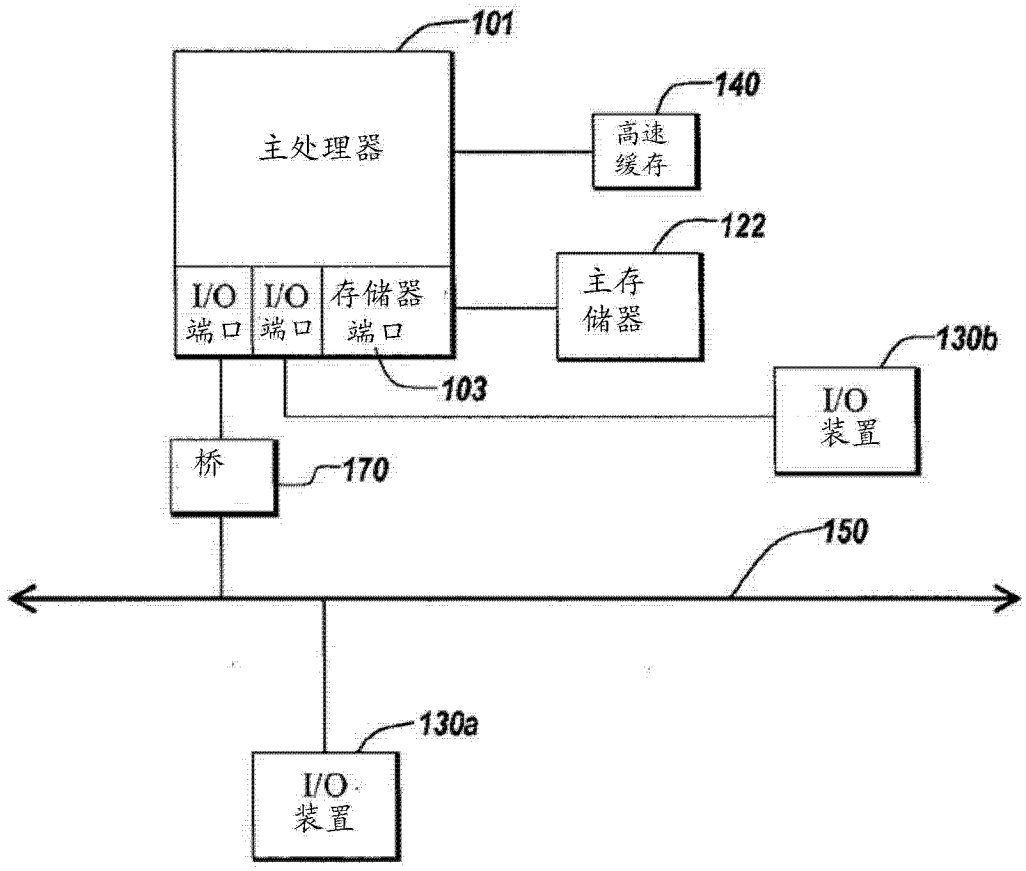

101
主处理器
140
高速缓存
122
主存储器
I/O 端口
I/O 端口
存储器端口
103
130b
I/O 装置
桥
170
150
130a
I/O 装置

图 1F

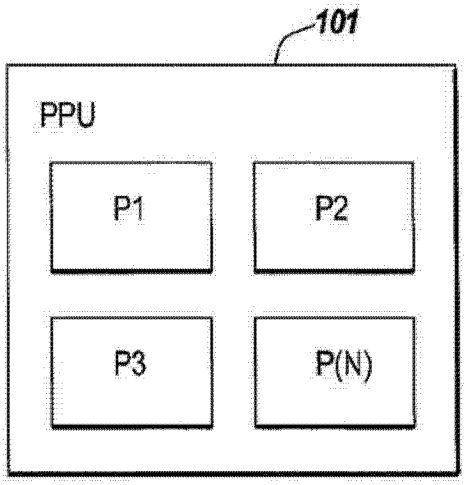

101
PPU
P1
P2
P3
P(N)

图 1G

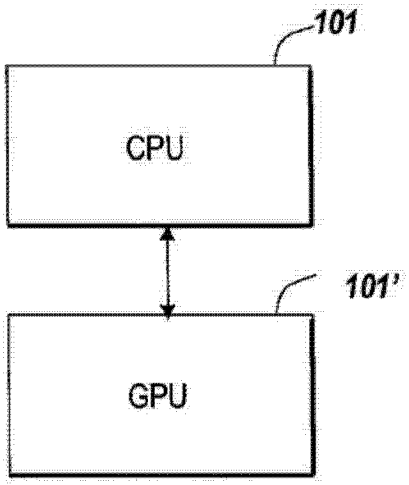

101
CPU
101'
GPU

图 1H

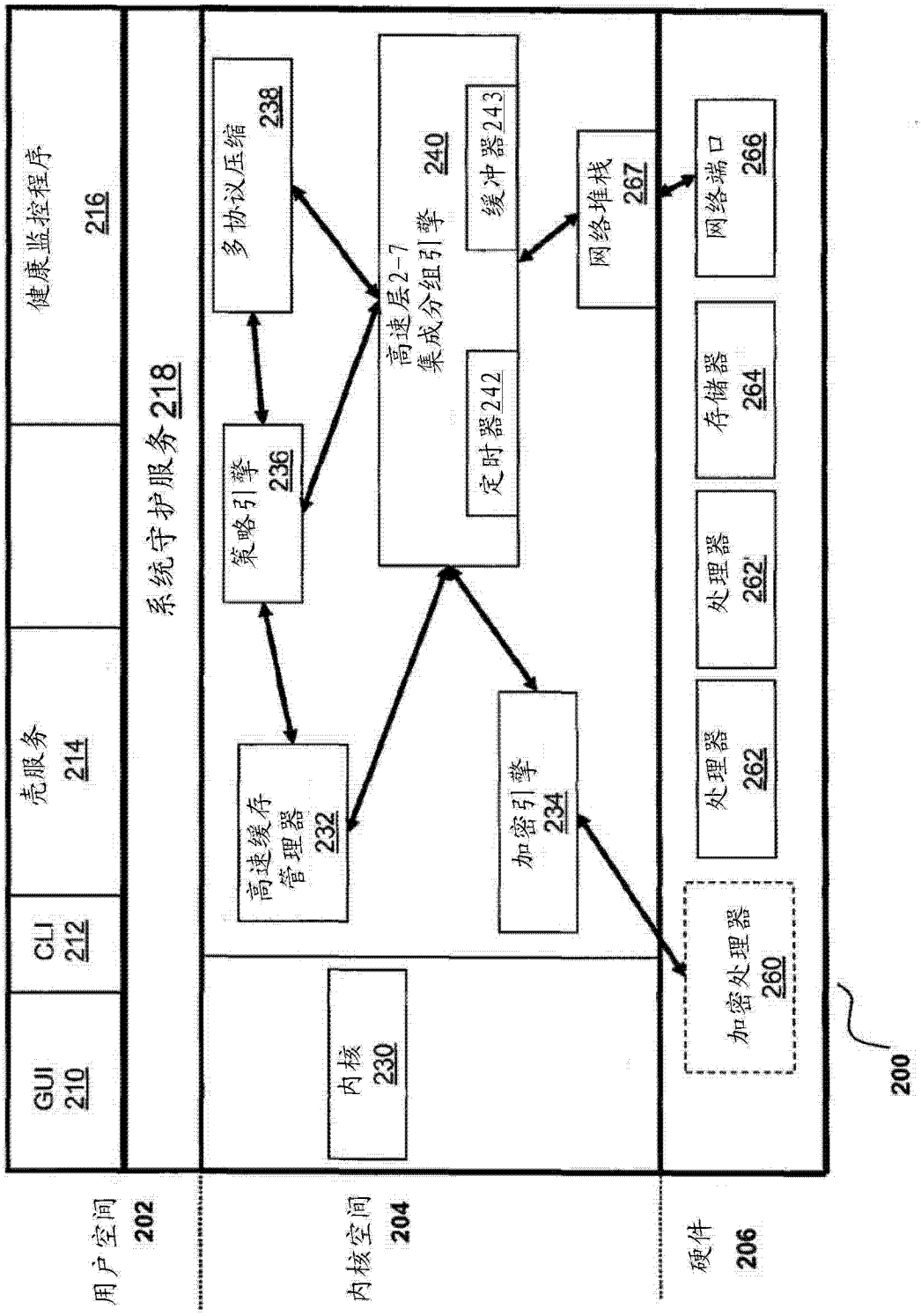

GUI 210
CLI 212
壳服务 214
健康监控程序 216
系统守护服务 218
多协议压缩 238
策略引擎 236
高速缓存管理器 232
内核 230
高速层2-7集成分组引擎 240
定时器242
缓冲器243
加密引擎 234
网络堆栈 267
网络端口 266
存储器 264
处理器 262'
处理器 262
加密处理器 260
200
用户空间 202
内核空间 204
硬件 206

图 2A

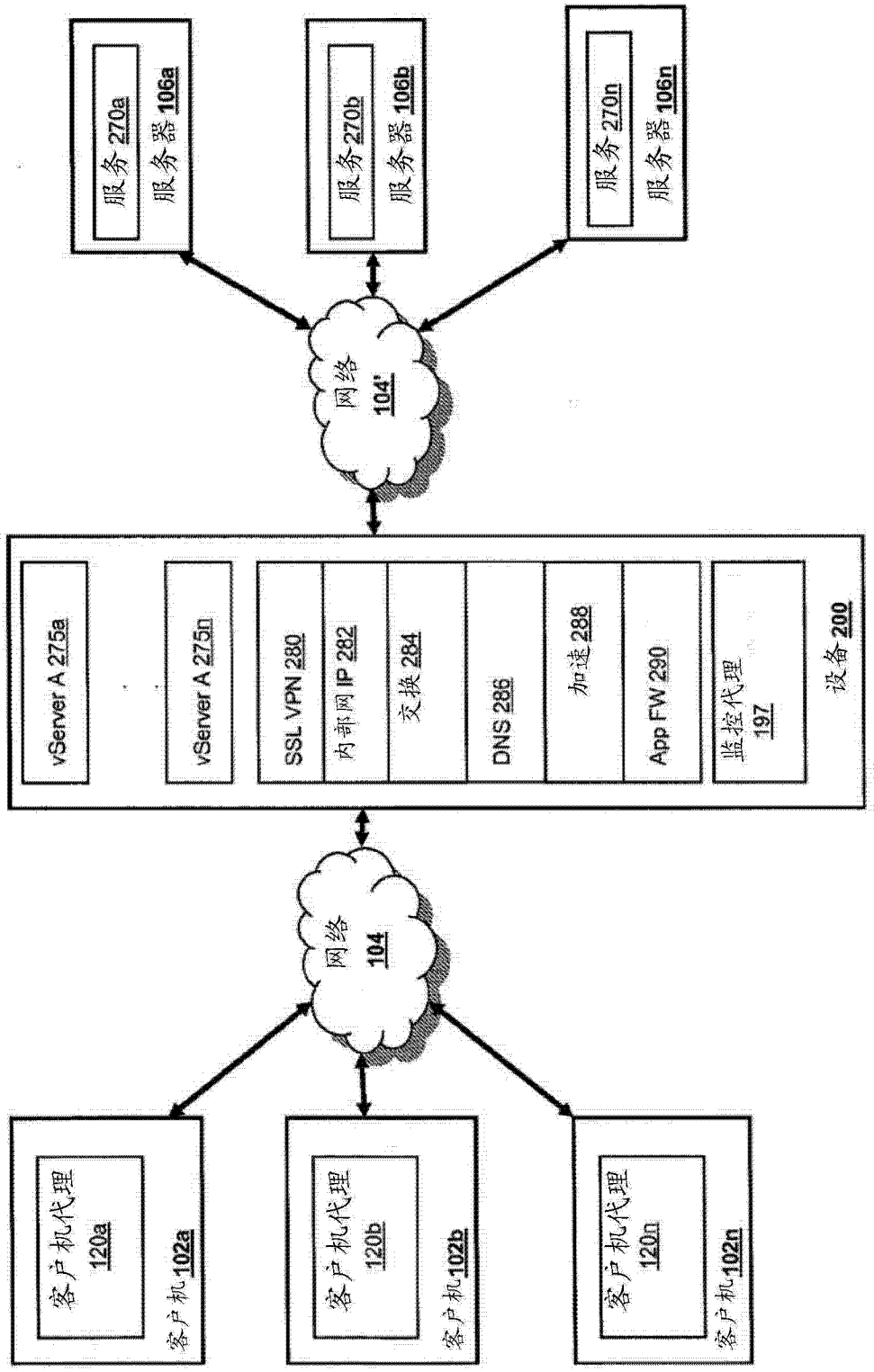

服务 270a
服务器 106a
服务 270b
服务器 106b
服务 270n
服务器 106n
网络 104'
vServer A 275a
vServer A 275n
SSL VPN 280
内部网 IP 282
交换 284
DNS 286
加速 288
App FW 290
监控代理 197
设备 200
网络 104
客户机代理 120a
客户机 102a
客户机代理 120b
客户机 102b
客户机代理 120n
客户机 102n

图 2B

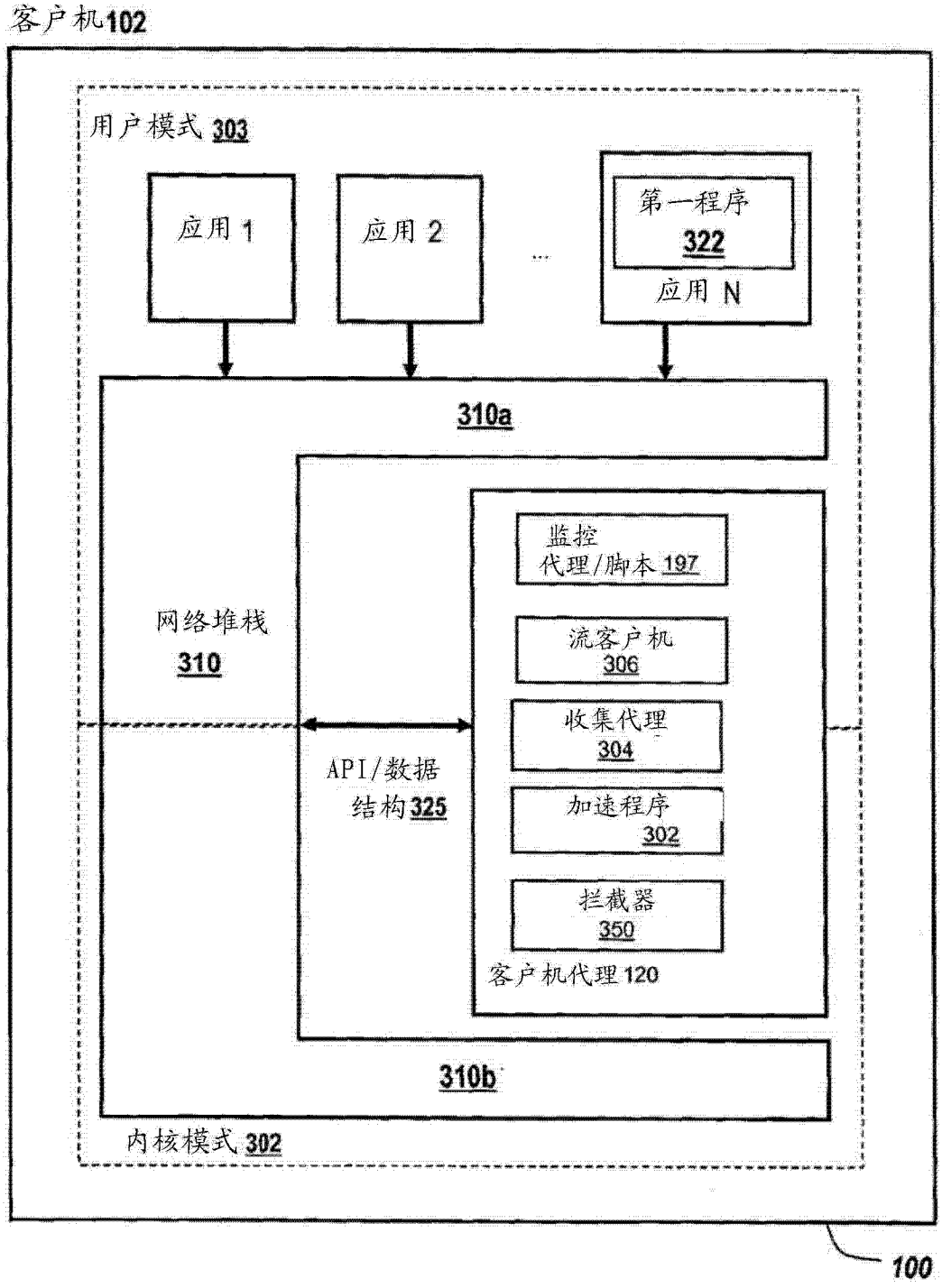

客户机102
用户模式 303
应用 1
应用 2
...
第一程序
322
应用 N
310a
网络堆栈
310
监控
代理/脚本 197
流客户机
306
收集代理
304
加速程序
302
拦截器
350
客户机代理 120
API/数据
结构325
310b
内核模式 302
100

图 3

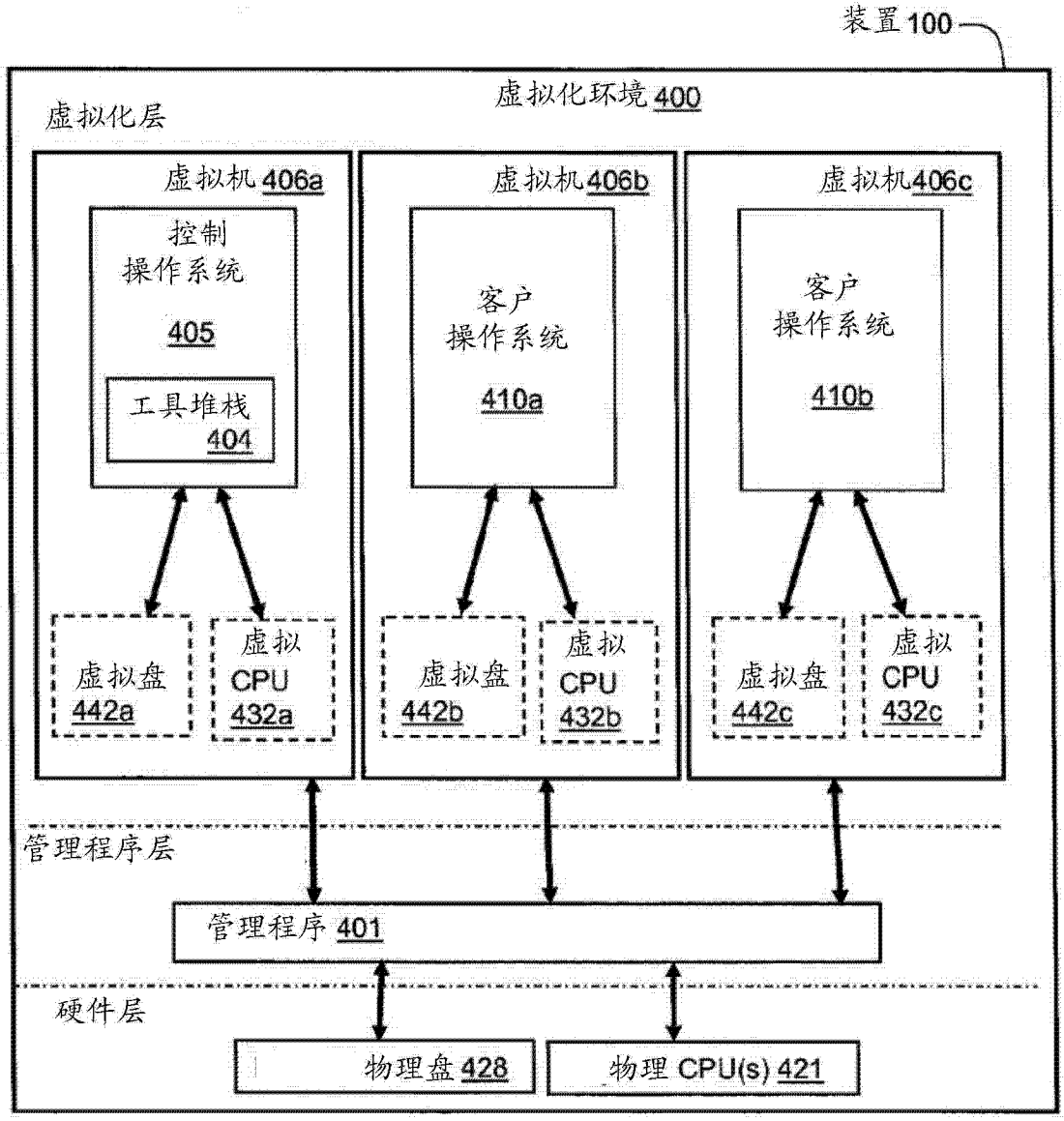

装置 100
虚拟化环境 400
虚拟化层
虚拟机 406a
虚拟机 406b
虚拟机 406c
控制
操作系统
405
工具堆栈
404
客户
操作系统
410a
客户
操作系统
410b
虚拟盘
442a
虚拟
CPU
432a
虚拟盘
442b
虚拟
CPU
432b
虚拟盘
442c
虚拟
CPU
432c
管理程序层
管理程序 401
硬件层
物理盘 428
物理 CPU(s) 421

图 4A

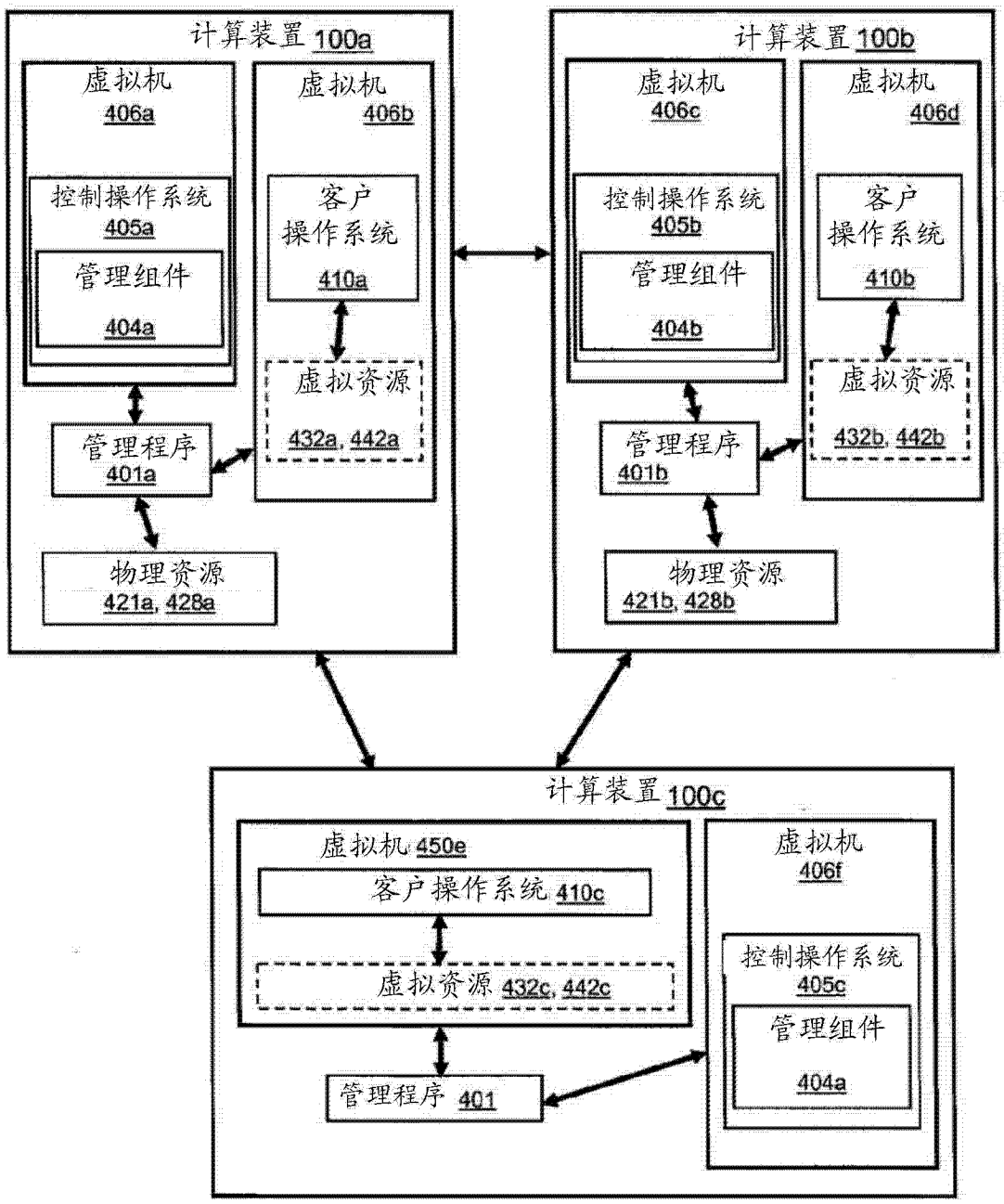

计算装置 100a
虚拟机 406a
控制操作系统 405a
管理组件 404a
虚拟机 406b
客户操作系统 410a
虚拟资源 432a, 442a
管理程序 401a
物理资源 421a, 428a
计算装置 100b
虚拟机 406c
控制操作系统 405b
管理组件 404b
虚拟机 406d
客户操作系统 410b
虚拟资源 432b, 442b
管理程序 401b
物理资源 421b, 428b
计算装置 100c
虚拟机 450e
客户操作系统 410c
虚拟资源 432c, 442c
管理程序 401
虚拟机 406f
控制操作系统 405c
管理组件 404a

图 4B

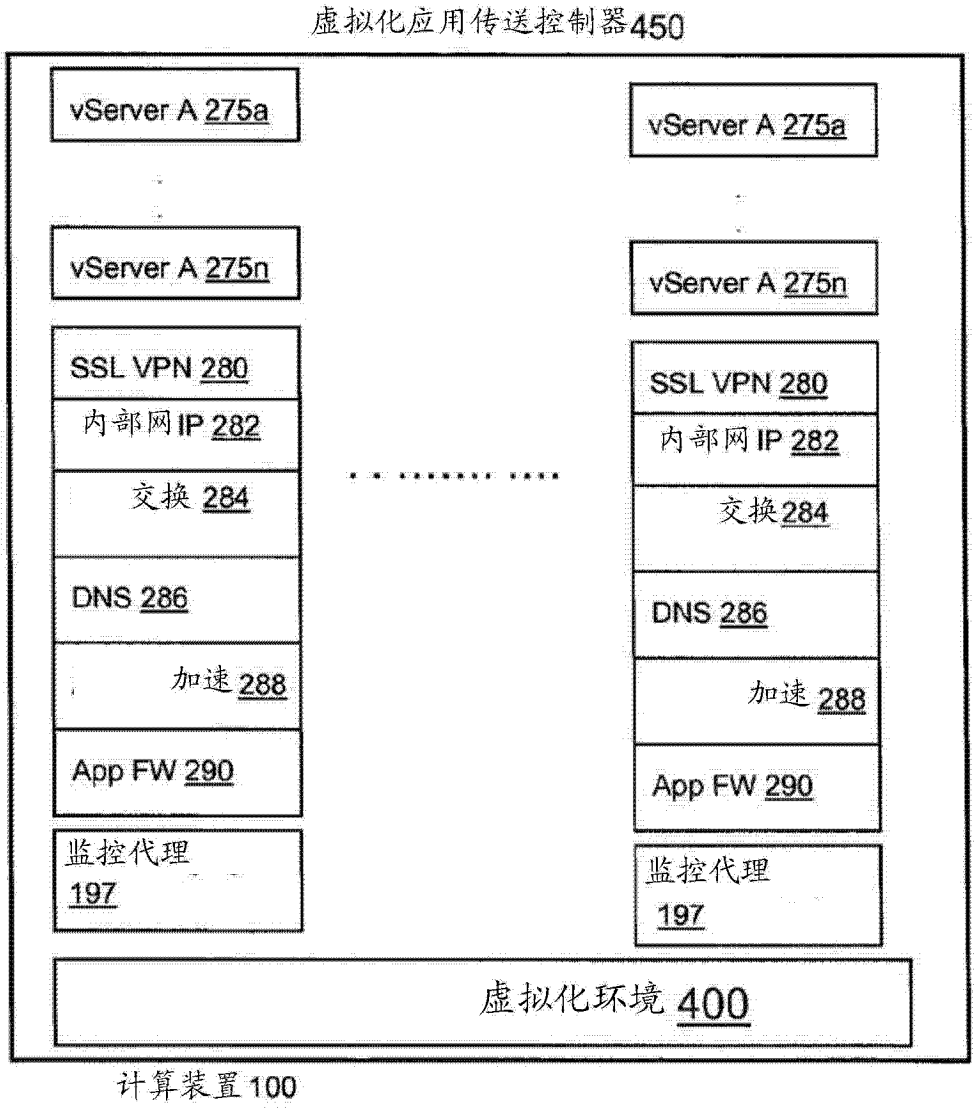

虚拟化应用传送控制器450
vServer A 275a
vServer A 275n
SSL VPN 280
内部网IP 282
交换 284
DNS 286
加速 288
App FW 290
监控代理 197
vServer A 275a
vServer A 275n
SSL VPN 280
内部网IP 282
交换284
DNS 286
加速 288
App FW 290
监控代理 197
虚拟化环境 400
计算装置100

图 4C

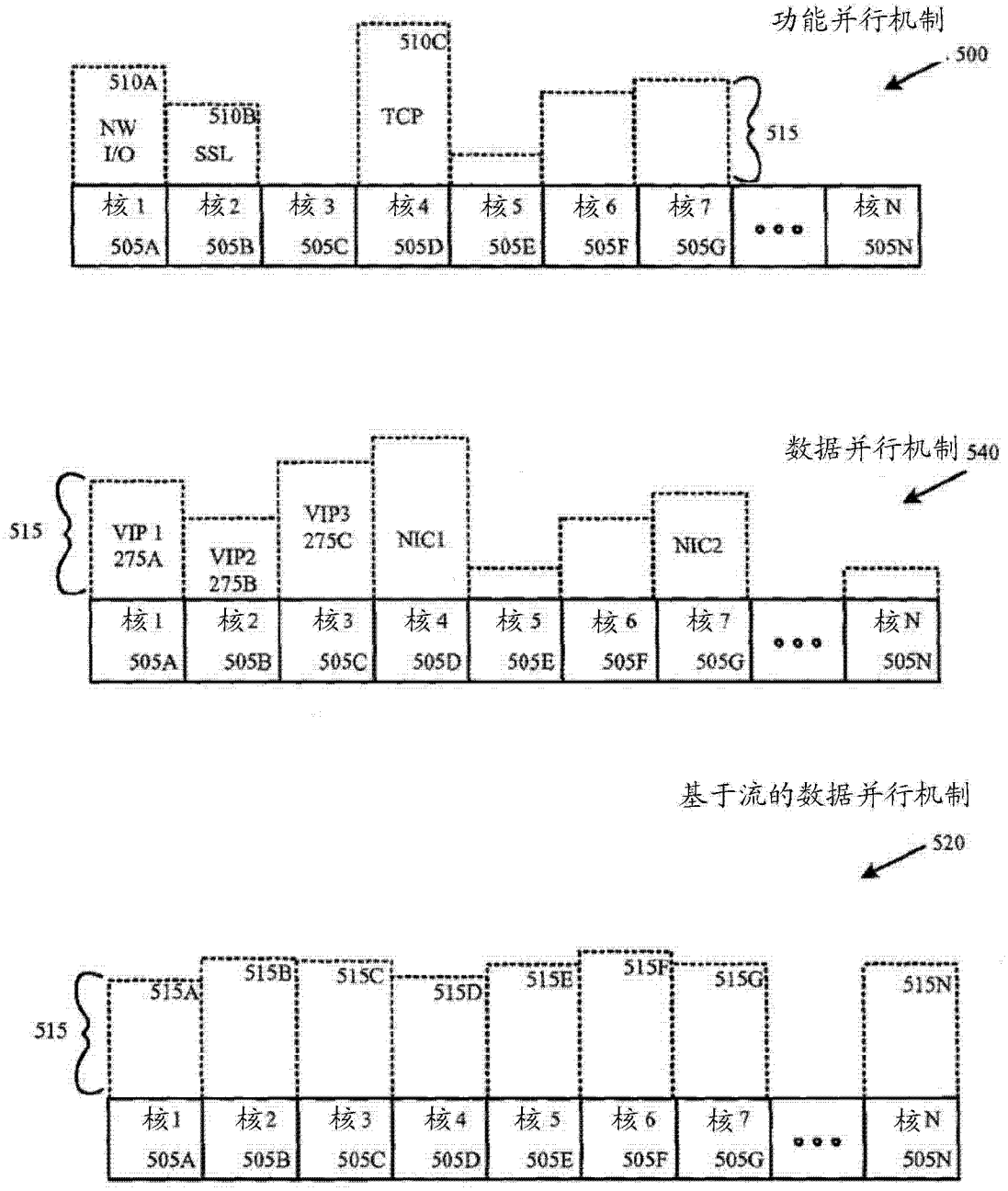

功能并行机制
500
510A
NW I/O
510B
SSL
510C
TCP
515
核 1 505A
核 2 505B
核 3 505C
核 4 505D
核 5 505E
核 6 505F
核 7 505G
核 N 505N
数据并行机制 540
515
VIP 1 275A
VIP2 275B
VIP3 275C
NIC1
NIC2
基于流的数据并行机制
520
515A
515B
515C
515D
515E
515F
515G
515N

图 5A

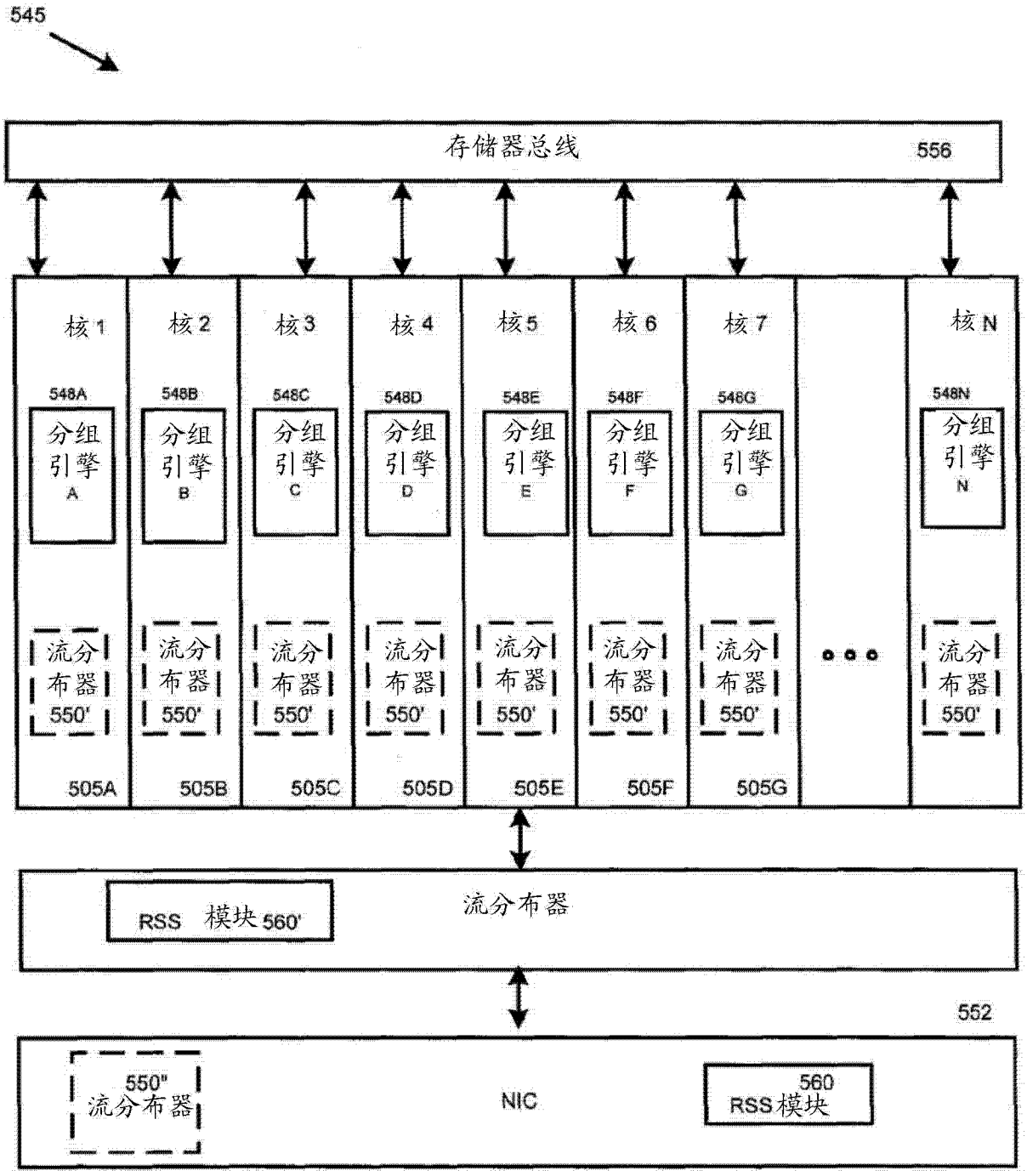

545
存储器总线
556
核 1
核 2
核 3
核 4
核 5
核 6
核 7
核 N
548A
548B
548C
548D
548E
548F
548G
548N
分组引擎 A
分组引擎 B
分组引擎 C
分组引擎 D
分组引擎 E
分组引擎 F
分组引擎 G
分组引擎 N
流分布器 550'
505A
505B
505C
505D
505E
505F
505G
RSS 模块 560'
流分布器
552
550" 流分布器
NIC
560 RSS 模块

图 5B

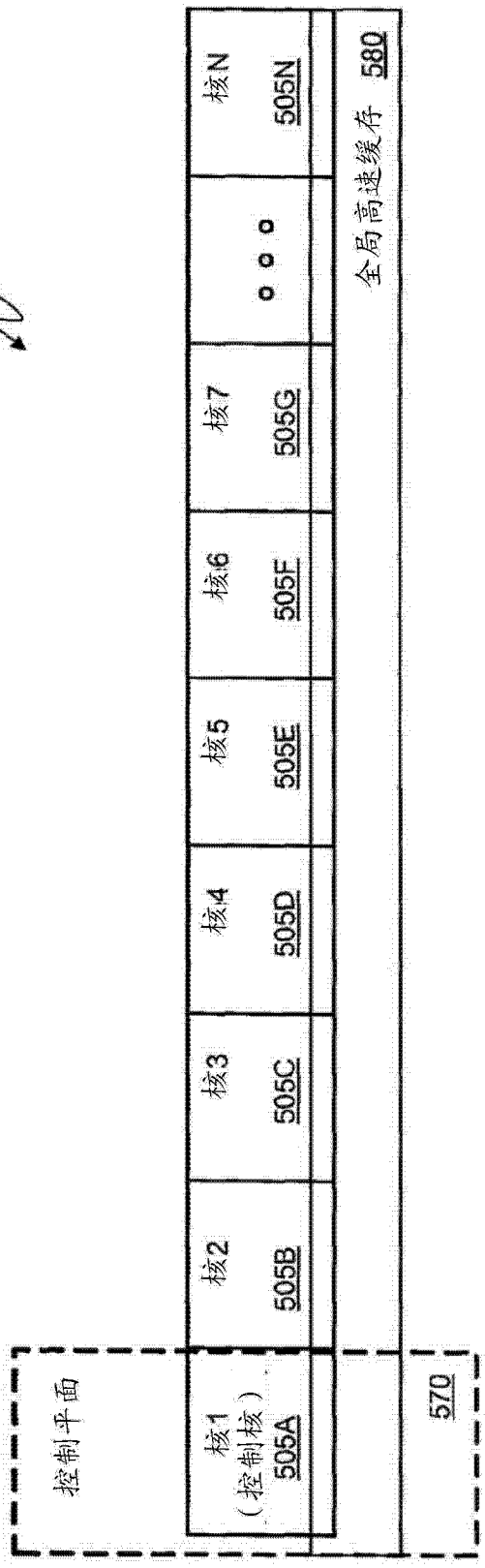

575
控制平面
核1
（控制核）
505A
核2
505B
核3
505C
核4
505D
核5
505E
核6
505F
核7
505G
核N
505N
全局高速缓存 580
570

图 5C

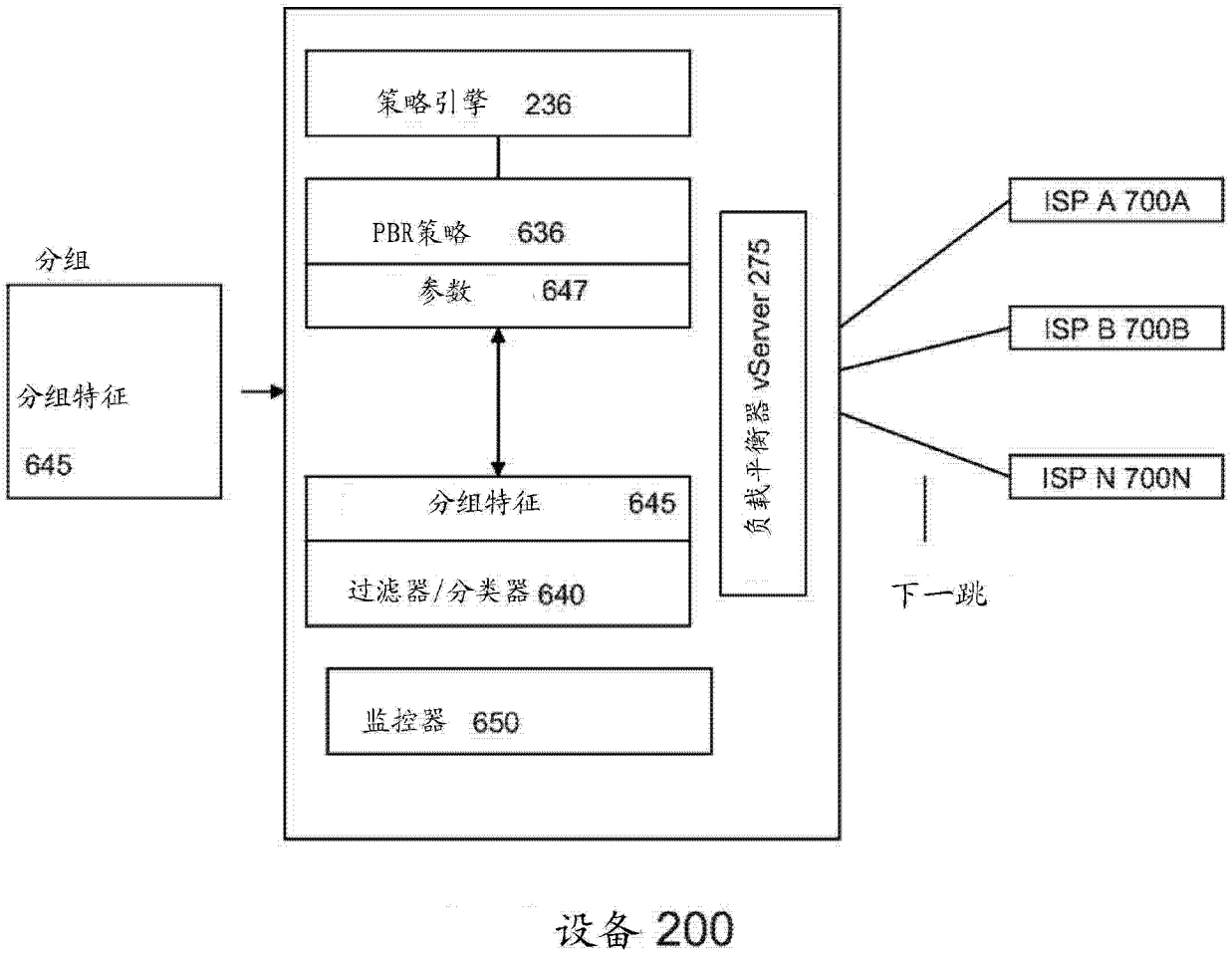

策略引擎 236
PBR策略 636
参数 647
分组特征 645
过滤器/分类器 640
监控器 650
负载平衡器 vServer 275
分组
分组特征
645
ISP A 700A
ISP B 700B
ISP N 700N
下一跳
设备 200

图 6A

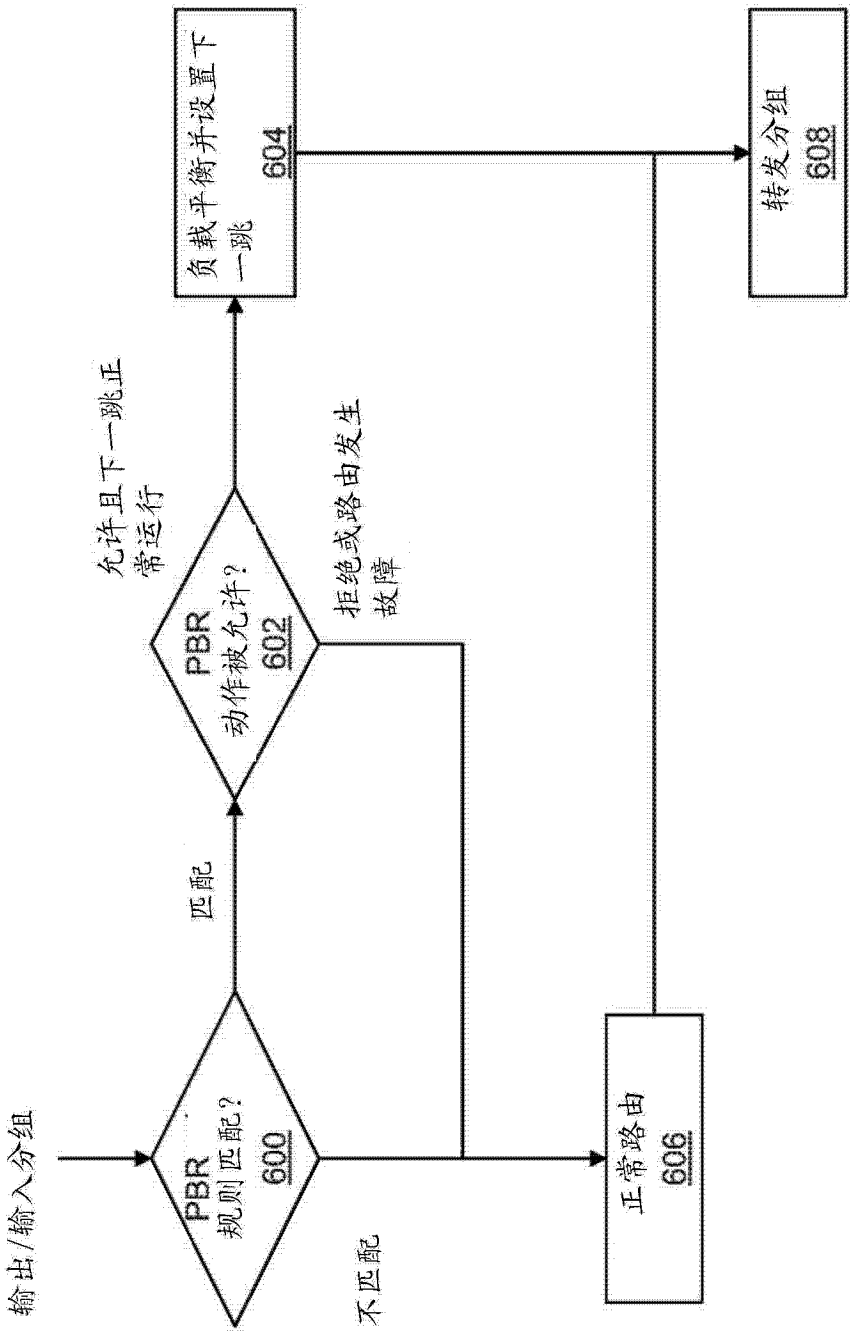

输出/输入分组
PBR
规则匹配?
600
匹配
不匹配
PBR
动作被允许?
602
允许且下一跳正常运行
拒绝或路由发生故障
负载平衡并设置下一跳
604
正常路由
606
转发分组
608

图 6B

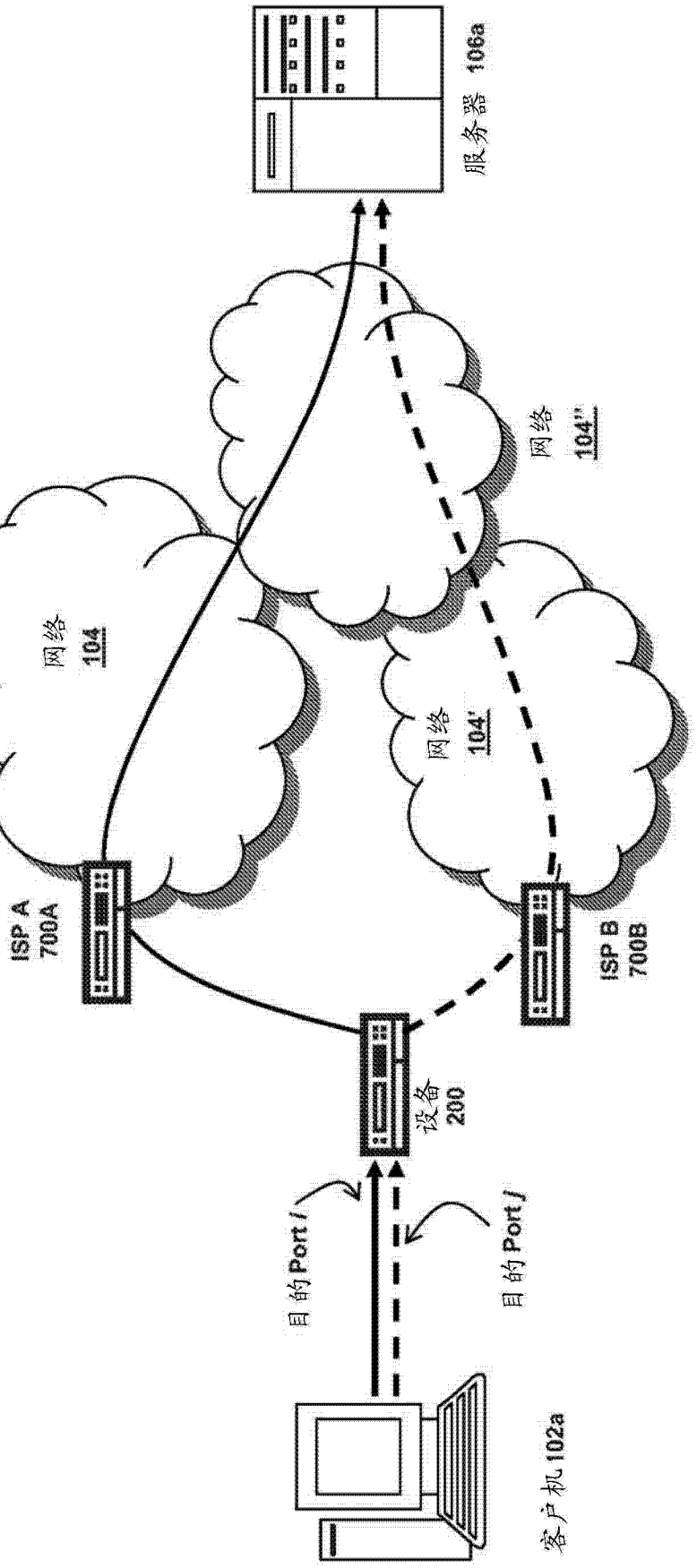

服务器 106a
网络 104''
网络 104
网络 104'
ISP A 700A
ISP B 700B
设备 200
目的 Port i
目的 Port j
客户机 102a

图 7A

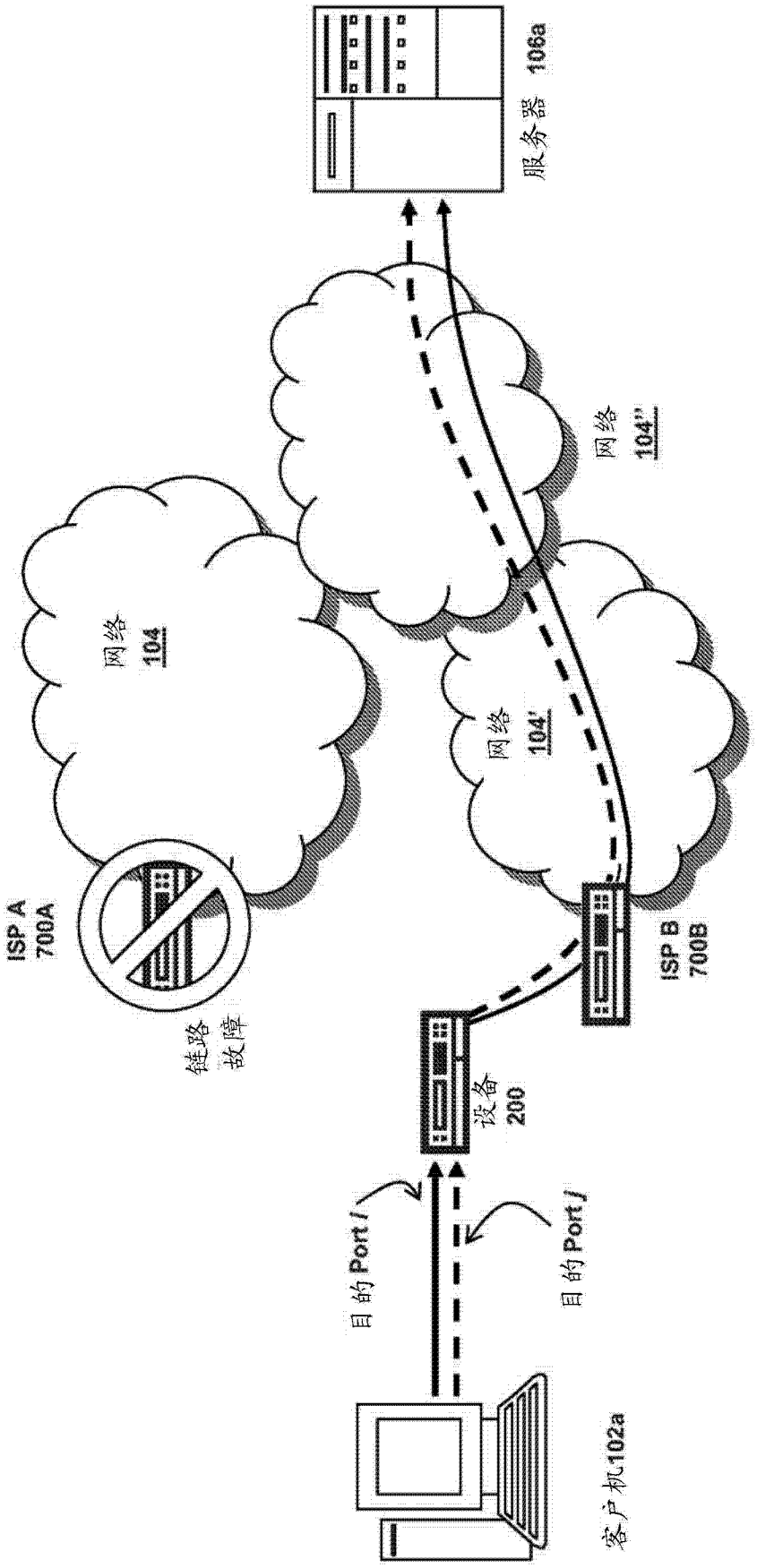

服务器 106a
网络 104''
网络 104
网络 104'
ISP A 700A
ISP B 700B
链路故障
设备 200
目的 Port I
目的 Port J
客户机102a

图 7B

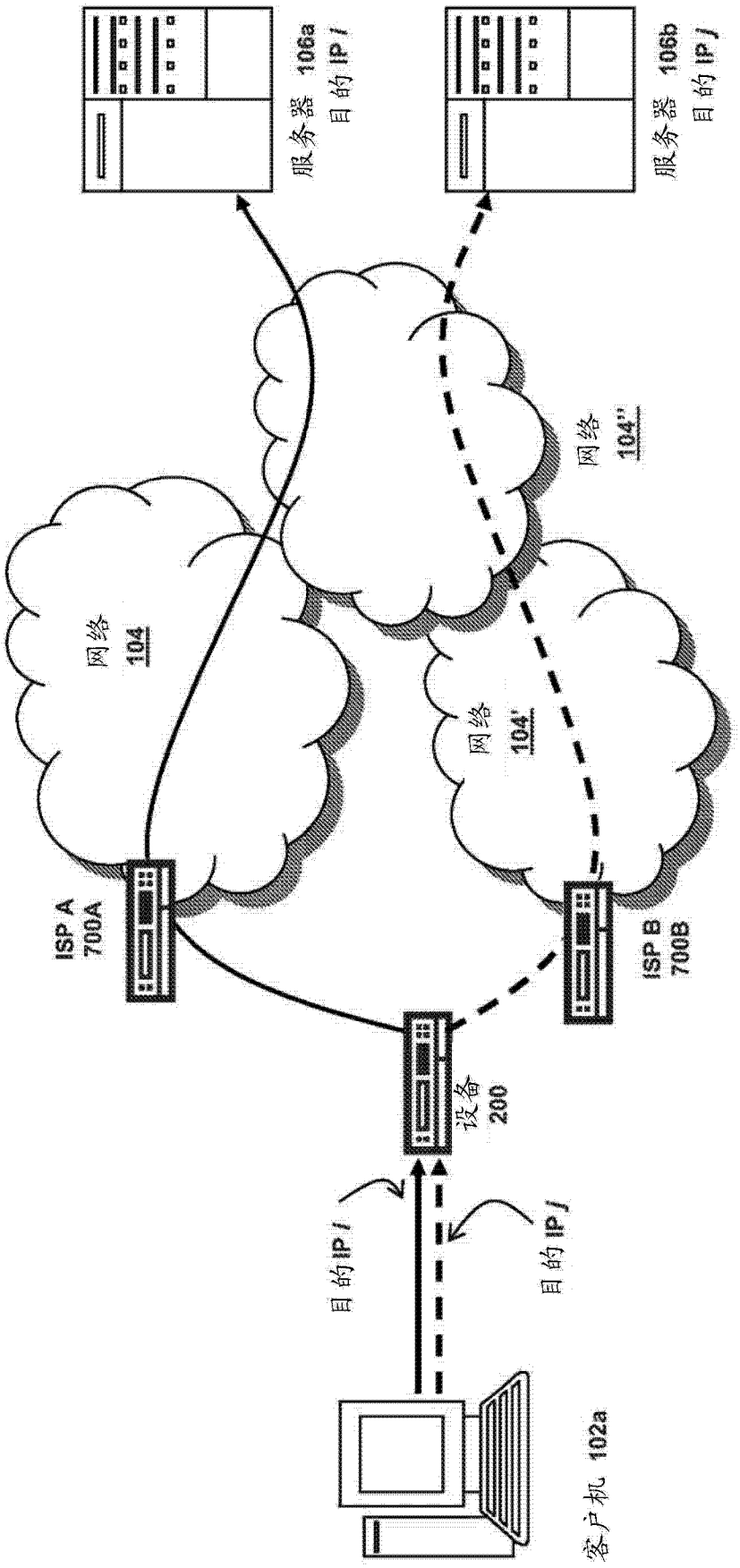

服务器 106a
目的 IP i
服务器 106b
目的 IP j
网络
104''
网络
104
网络
104'
ISP A
700A
ISP B
700B
设备
200
目的 IP i
目的 IP j
客户机 102a

图 7C

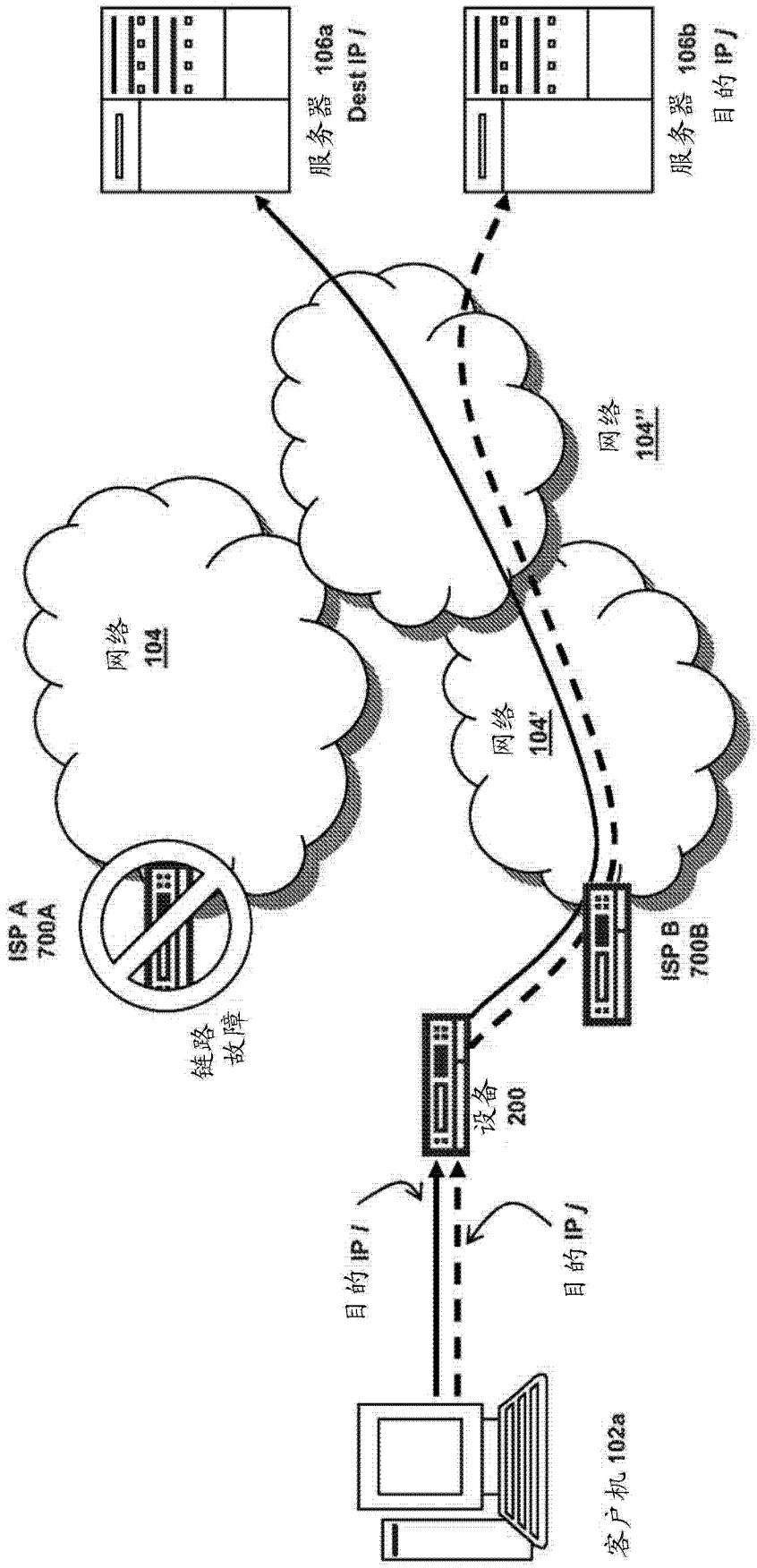

服务器 106a
Dest IP i
服务器 106b
目的 IP j
网络
104
网络
104'
网络
104''
ISP A
700A
ISP B
700B
链路
故障
设备
200
目的 IP i
目的 IP j
客户机 102a

图 7D

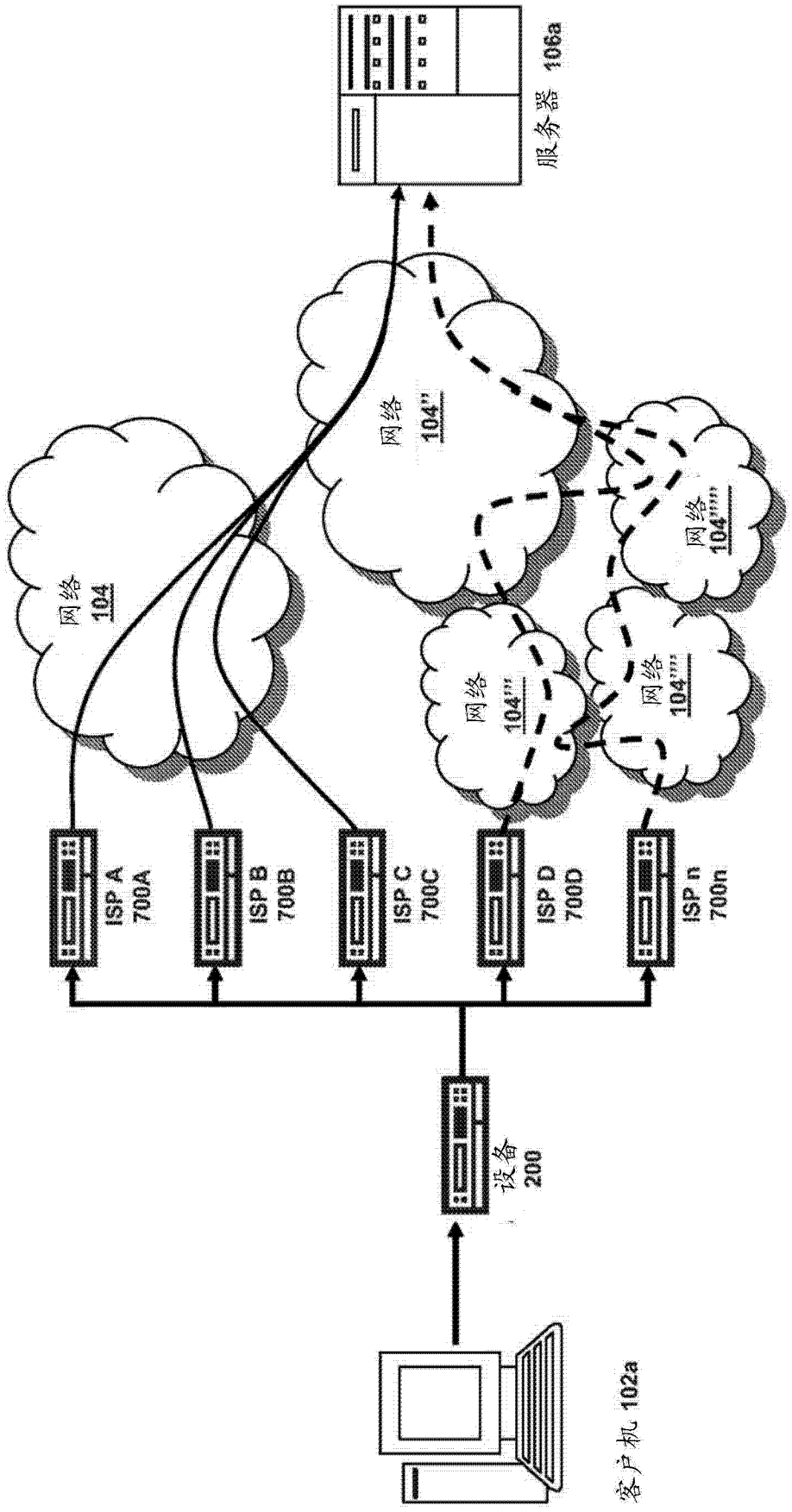

服务器 106a
网络 104''
网络 104
网络 104''''
网络 104'''
网络 104'''''
ISP A 700A
ISP B 700B
ISP C 700C
ISP D 700D
ISP n 700n
设备 200
客户机 102a

图 7E