



(12)发明专利

(10)授权公告号 CN 105471961 B

(45)授权公告日 2019.03.08

(21)申请号 201510765025.5

(22)申请日 2015.08.14

(65)同一申请的已公布的文献号
申请公布号 CN 105471961 A

(43)申请公布日 2016.04.06

(30)优先权数据
14/486270 2014.09.15 US

(73)专利权人 英特尔公司
地址 美国加利福尼亚州

(72)发明人 董耀祖 田坤

(74)专利代理机构 中国专利代理(香港)有限公司 72001
代理人 姜冰 张懿

(51)Int.Cl.

H04L 29/08(2006.01)

G06F 9/455(2006.01)

(56)对比文件

CN 101488960 A,2009.07.22,

US 2006010227 A1,2006.01.01,

CN 103155518 A,2013.06.12,

US 2012093150 A1,2012.04.19,

Wei Ye, Yaozu Dong,etc..“High Availability Through Output Continuity”.《IEEE International Conference on Cluster Computing Workshops》.2012,

审查员 高惠敏

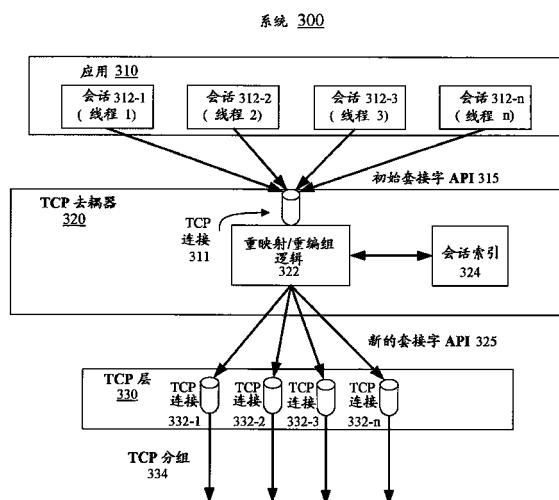
权利要求书3页 说明书17页 附图10页

(54)发明名称

用于重映射多线程应用的会话的方法和设备

(57)摘要

本公开的发明名称是“用于重映射多线程应用的会话的技术”。示例可以包括多线程应用的会话的重映射,其可以在经由多个传输控制协议(TCP)连接耦合到服务器的客户端或服务端执行。会话可以被重映射,使得多线程应用可以期望通过相同TCP连接来路由会话但是会话经由分开的TCP连接被实际输出。



1. 一种设备,包括:

在服务器的电路,所述服务器经由多个传输控制协议(TCP)连接耦合到客户端,所述服务器布置成主管能够单独地执行第一应用的主或次虚拟机(VM);

截取组件,用于由所述电路执行以截取从所述第一应用调用的第一和第二套接字应用接口(API),所述第一和第二套接字API布置成路由相应的第一和第二会话以用于通过来自所述多个TCP连接之中的相同TCP连接的期望输出,所述第一和第二会话由所述第一应用响应于从在所述客户端的第二应用接收的请求而生成;

索引组件,用于由所述电路执行以使用所述第一和第二会话的相应第一和第二标识符信息来生成会话索引;以及

重映射组件,用于由所述电路执行以重映射所述第一和第二会话以用于通过来自所述多个TCP连接之中的不同的TCP连接的实际输出,使得所述会话索引能够在所述客户端用来将所述第一和第二会话与所述相同TCP连接重组合以用于向所述第二应用的递送。

2. 如权利要求1所述的设备,包括:

所述截取组件用于通过替代由所述应用用来路由会话以用于通过所述多个TCP连接向所述客户端的期望输出的套接字API的动态链接库来截取所述第一和第二套接字API,用TCP去耦链接库来替代所述动态链接库;以及

所述重映射组件用于重映射所述第一和第二会话,包括使用所述TCP去耦链接库以促使所述第一和第二会话通过所述不同的TCP连接被实际输出到所述客户端。

3. 如权利要求1所述的设备,所述截取组件用于截取所述第一和第二套接字API包括所述截取组件用于基于与所述客户端的TCP数据交换或与所述客户端的TCP连接设置之一来模仿所述第一和第二套接字API。

4. 如权利要求3所述的设备,包括:

所述截取组件用于基于所述TCP连接设置来模仿所述第一和第二套接字API,所述截取组件用于创建所述第一和第二套接字API的多个操作系统(OS)套接字API并且指派所述多个OS套接字API到所述不同的TCP连接;以及

用于重映射所述第一和第二会话的所述重映射组件使用所述多个OS套接字API以促使所述第一和第二会话通过所述不同的TCP连接被实际输出到所述客户端。

5. 如权利要求4所述的设备,包括:

所述截取组件用于响应于与所述客户端的TCP拆除的文件结束(EOF)消息而移除所述第一和第二套接字API的所述多个OS套接字API。

6. 如权利要求3所述的设备,包括:

所述截取组件用于基于所述TCP数据交换来模仿所述第一和第二套接字API并且使用所述会话索引以寻址将所述第一和第二会话映射到指派到所述不同的TCP连接的多个操作系统(OS)套接字API的固定映射表;以及

所述重映射组件用于基于所述固定映射表来重映射所述第一和第二会话以促使所述第一和第二会话通过所述不同的TCP连接被实际输出到所述客户端。

7. 如权利要求1所述的设备,所述截取组件用于基于所述应用有选择地被选取用于截取而能够由所述主或次VM执行的另一个应用不被选取用于截取来截取所述第一和第二套接字API。

8. 如权利要求1所述的设备,所述第一和第二标识符信息包括指派给用来生成所述第一会话的第一线程的第一线程名或标识符以及指派给用来生成所述第二会话的第二线程的第二线程名或标识符。

9. 如权利要求1所述的设备,所述第一和第二标识符信息包括含有与所述第一会话的生成关联的第一因特网协议(IP)地址的第一协议栈模式以及含有与所述第二会话的生成关联的第二IP地址的第二协议栈模式。

10. 一种在布置成主管能够单独地执行应用的主或次虚拟机(VM)的服务器的电路实现的方法,所述方法包括:

截取从所述应用做出的第一和第二套接字应用接口(API)调用,所述第一和第二套接字API布置成路由相应的第一和第二会话以用于通过来自耦合到所述服务器的客户端的多个传输控制协议TCP连接之中的相同TCP连接的期望输出,所述第一和第二会话由所述应用响应于从所述客户端接收的数据而生成;

使用所述第一和第二会话的相应第一和第二标识符信息来生成会话索引;以及

重映射所述第一和第二会话以用于通过来自所述多个TCP连接之中的不同的TCP连接的实际输出,使得所述会话索引能够在所述客户端用来将所述第一和第二会话与所述相同TCP连接重组。

11. 如权利要求10所述的方法,包括:

截取所述第一和第二套接字API包括替代由所述应用用来路由会话以用于通过所述多个TCP连接向所述客户端的期望输出的套接字API的动态链接库,所述动态链接库用TCP去耦链接库来替代;以及

重映射所述第一和第二会话包括使用所述TCP去耦链接库以促使所述第一和第二会话通过所述不同的TCP连接被实际输出到所述客户端。

12. 如权利要求10所述的方法,截取所述第一和第二套接字API包括基于与所述客户端的TCP数据交换或与所述客户端的TCP连接设置之一来模仿所述第一和第二套接字API。

13. 如权利要求12所述的方法,基于所述TCP连接设置来模仿所述第一和第二套接字API包括:

创建所述第一和第二套接字API的多个操作系统(OS)套接字API,所述多个OS套接字API被指派到所述不同的TCP连接;以及

重映射所述第一和第二会话包括使用所述多个OS套接字API以促使所述第一和第二会话通过所述不同的TCP连接被实际输出到所述客户端。

14. 如权利要求13所述的方法,包括:

响应于与所述客户端的TCP拆除的文件结束(EOF)消息而移除所述第一和第二套接字API的所述多个OS套接字API。

15. 如权利要求14所述的方法,基于所述TCP数据交换来模仿所述第一和第二套接字API包括:

使用所述会话索引以寻址将所述第一和第二会话映射到指派到所述不同的TCP连接的多个操作系统(OS)套接字API的固定映射表;以及

重映射所述第一和第二会话包括使用所述固定映射表以促使所述第一和第二会话通过所述不同的TCP连接被实际输出到所述客户端。

16. 如权利要求10所述的方法,截取所述第一和第二套接字API包括基于所述应用有选择地被选取用于截取而能够由所述主或所述次VM执行的另一个应用不被选取用于截取来进行截取。

17. 至少一个机器可读媒体,包括多个指令,所述指令响应于被在服务器的系统来执行而促使所述系统实行根据权利要求10到16的任一项的方法。

18. 一种设备,包括用于执行如权利要求10到16的任一项所述的方法的部件。

19. 一种方法,包括:

在经由多个传输控制协议(TCP)连接耦合到服务器的客户端的电路经由来自所述多个TCP连接之中的不同的TCP连接接收第一和第二会话,所述第一和第二会话由所述服务器主管的主或次虚拟机执行的第一应用来生成并且经由所述不同的TCP连接从所述服务器输出,所述第一和第二会话是响应于在所述客户端的第二应用先前发送到所述第一应用的请求而生成的;

接收具有所述第一和第二会话的标识符信息的会话索引;

基于所述会话索引重编组所述第一和第二会话至来自所述多个TCP连接之中的相同TCP连接;以及

经由所述相同TCP连接转发所述第一和第二会话到相应的第一和第二套接字应用接口(API)以用于所述第二应用。

20. 如权利要求19所述的方法,所述标识符信息包括指派给用来生成所述第一应用的所述第一会话的第一线程的第一线程名或标识符以及指派给用来生成所述第一应用的所述第二会话的第二线程的第二线程名或标识符。

21. 如权利要求19所述的方法,重编组所述第一和第二会话到所述相同TCP连接包括使用如在所述服务器用来将所述第一和第二会话在所述服务器从所述相同TCP连接重映射到所述不同的TCP连接的相同循环策略。

22. 至少一个机器可读媒体,包括多个指令,所述指令响应于被在耦合到服务器的客户端的系统来执行而促使所述系统实行根据权利要求19到21的任一项的方法。

23. 一种设备,包括用于执行如权利要求19到21的任一项所述的方法的部件。

24. 一种在经由多个传输控制协议(TCP)连接耦合到服务器的客户端的设备,所述服务器布置成主管能够单独地执行第一应用的主或次虚拟机(VM),所述设备包括:

电路;

接收组件,用于由所述电路执行以经由来自所述多个TCP连接之中的不同的TCP连接接收所述第一应用生成的第一和第二会话,所述第一和第二会话是响应于在所述客户端的第二应用先前发送到所述第一应用的请求而生成的;

索引组件,用于由所述电路执行以接收具有所述第一和第二会话的标识符信息的会话索引;

重编组组件,用于由所述电路执行以基于所述会话索引将所述第一和第二会话重编组到来自所述多个TCP连接之中的相同TCP连接;以及

转发组件,用于由所述电路执行以经由所述相同TCP连接转发所述第一和第二会话到相应的第一和第二套接字应用接口(API)以用于所述第二应用。

用于重映射多线程应用的会话的方法和设备

技术领域

[0001] 本文所描述的示例一般涉及路由通过服务器和客户端之间的多线程程序生成的数据。

背景技术

[0002] 与客户端计算装置耦合的网络服务器日益地被布置成支持或主管(host)虚拟计算机(VM),其使多个操作系统和/或应用能够由单一计算平台来支持。同样地,当对于主管VM的服务器希望高有效性时,主VM(PVM)和次VM(SVM)可以各自地在分开的服务器或节点(例如在数据中心内)上被主管,并且它们的状态可以被复制。这种状态的复制可以提供环境无关、软件实现硬件故障容错解决方案用于“非停止服务”。在主管PVM的服务器经受硬件故障和/或PVM进入故障状态时,故障容错解决方案可以考虑SVM来接管(故障转移)。

[0003] 锁步是可以按指令复制VM状态的故障容错解决方案。例如,PVM和SVM并联执行用于确定性指令,但是锁步用于非确定性指令。然而,在处理多处理器(MP)实现时,锁步可以经受很大的开销,其中每个存储器存取可以是非确定性的。

[0004] 检查点是在以周期性时期复制PVM状态到SVM的另一种故障容错解决方案。对于检查点,为了保证成功的故障转移,全部的输出分组可以需要被缓冲直到成功的检查点已被完成。在VM环境中缓冲直到成功的检查点可以导致额外的网络等待时间和开销,归因于输出分组缓冲和频繁的检查点。

[0005] 粗粒度锁步(COLO)是仍有的另一个故障容错解决方案,其使用来自客户端的相同的请求/数据(输入)网络分组来使PVM和SVM都反馈。支持COLO的逻辑可以能够监视PVM和SVM的输出响应以及认为SVM的状态为PVM的状态的有效复本,只要由SVM生成的网络响应(输出)匹配PVM的网络响应。如果给定的网络响应不匹配,那么网络响应到客户端的传送被阻止直到PVM状态已经与SVM状态同步(强加新的检查点)。因此,COLO可以确保经由向SVM的故障转移故障容错系统是高度可用的。这个高可用性可以存在,即使非确定性可以表示SVM的内部状态不同于PVM的内部状态,SVM同等有效并且从外部观测者对于实现COLO的故障容错系统的观点来看保持一致。因此,COLO可以在纯锁步或检查点故障容错解决方案上具有优势。

[0006] COLO故障容错解决方案可以利用如与传输控制协议(TCP)栈相关的此类协议。TCP栈可以布置成具有按连接的状态并且可以能够从分组损耗和/或分组重排序中恢复。COLO可以包括按TCP的连接响应分组比较的使用。按TCP的连接响应分组比较可以认为SVM状态为有效复本,如果从PVM输出的每个TCP连接的响应分组匹配从SVM输出的每个TCP连接的响应分组。此匹配是不管跨TCP连接的可能的分组顺序。

发明内容

[0007] 本公开提供一种设备,包括:在服务器的电路,所述服务器经由多个传输控制协议(TCP)连接耦合到客户端,所述服务器布置成主管能够单独地执行第一应用的主或次虚拟

机 (VM) ; 截取组件, 用于由所述电路执行以截取从所述第一应用调用的第一和第二套接字应用接口 (API), 所述第一和第二套接字 API 布置成路由相应的第一和第二会话以用于通过来自所述多个 TCP 连接之中的相同 TCP 连接的期望输出, 所述第一和第二会话由所述第一应用响应于从在所述客户端的第二应用接收的请求而生成; 索引组件, 用于由所述电路执行以使用对于所述第一和第二会话的相应第一和第二标识符信息来生成会话索引; 以及重映射组件, 用于由所述电路执行以重映射所述第一和第二会话以用于通过来自所述多个 TCP 连接之中的不同的 TCP 连接的实际输出, 使得所述会话索引能够在所述客户端用来将所述第一和第二会话与所述相同 TCP 连接重组合以用于向所述第二应用的递送。

[0008] 本公开还提供一种在布置成主管能够单独执行应用的主或次虚拟机 (VM) 的服务器的电路实现的方法, 所述方法包括: 截取从所述应用做出的第一和第二套接字应用接口 (API) 调用, 所述第一和第二套接字 API 布置成路由相应的第一和第二会话以用于通过来自耦合到所述服务器的客户端的多个 TCP 连接之中的相同传输控制协议 (TCP) 连接的期望输出, 所述第一和第二会话由所述应用响应于从所述客户端接收的数据而生成; 使用对于所述第一和第二会话的相应第一和第二标识符信息来生成会话索引; 以及重映射所述第一和第二会话以用于通过来自所述多个 TCP 连接之中的不同的 TCP 连接的实际输出, 使得所述会话索引能够在所述客户端用来将所述第一和第二会话与所述相同 TCP 连接重组合。

[0009] 本公开还提供至少一个机器可读媒体, 包括多个指令, 所述指令响应于被在服务器的系统来执行而促使所述系统实行上述方法。

[0010] 本公开还提供一种设备, 包括用于执行上述方法的部件。

[0011] 本公开还提供一种方法, 包括: 在经由多个传输控制协议 (TCP) 连接耦合到服务器的客户端的电路经由来自所述多个 TCP 连接之中的不同的 TCP 连接接收第一和第二会话, 所述第一和第二会话由所述服务器主管的主或次虚拟机执行的第一应用来生成并且经由所述不同的 TCP 连接从所述服务器输出, 所述第一和第二会话是响应于在所述客户端的第二应用先前发送到所述第一应用的请求而生成的; 接收具有对于所述第一和第二会话的标识符信息的会话索引; 基于所述会话索引重编组所述第一和第二会话至来自所述多个 TCP 连接之中的相同 TCP 连接; 以及经由所述相同 TCP 连接转发所述第一和第二会话到相应的第一和第二套接字应用接口 (API) 以用于所述第二应用。

[0012] 本公开还提供一种设备, 包括用于执行前一段所述方法的部件。

[0013] 本公开还提供一种在经由多个传输控制协议 (TCP) 连接耦合到服务器的客户端的设备, 所述服务器布置成主管能够单独地执行第一应用的主或次虚拟机 (VM), 所述设备包括: 电路; 接收组件, 用于由所述电路执行以经由来自所述多个 TCP 连接之中的不同的 TCP 连接接收所述第一应用生成的第一和第二会话, 所述第一和第二会话是响应于在所述客户端的第二应用先前发送到所述第一应用的请求而生成的; 索引组件, 用于由所述电路执行以接收具有对于所述第一和第二会话的标识符信息的会话索引; 重编组组件, 用于由所述电路执行以基于所述会话索引将所述第一和第二会话重编组到来自所述多个 TCP 连接之中的相同 TCP 连接; 以及转发组件, 用于由所述电路执行以经由所述相同 TCP 连接转发所述第一和第二会话到相应的第一和第二套接字应用接口 (API) 以用于所述第二应用。

附图说明

- [0014] 图1图示示例第一系统。
- [0015] 图2图示示例编程模型。
- [0016] 图3图示示例第二系统。
- [0017] 图4图示示例第三系统。
- [0018] 图5图示示例TCP流程。
- [0019] 图6图示用于设备的示例框图。
- [0020] 图7图示第一逻辑流程的示例。
- [0021] 图8图示第二逻辑流程的示例。
- [0022] 图9图示第三逻辑流程的示例。
- [0023] 图10图示存储媒体的示例。
- [0024] 图11图示示例计算平台。

具体实施方式

[0025] 如在本公开中所预期的,COL0可以在纯锁步或检查点故障容错解决方案上具有优势。COL0可以经由使用主管PVM和SVM的服务器而大大地改善高度可用的故障容错系统的性能。然而,COL0的性能依靠每个TCP连接的输出类似性。例如,PVM和SVM可以生成的按TCP的连接的同样的输出分组的数量或PVM和SVM可以生成的持续时间匹配全部的TCP的连接的输出分组。

[0026] 输出类似性上的潜在地大的影响可以是通过由PVM或SVM执行的多线程应用来实现的编程模型。例如,更高级别层数的编程模型例如有1996年6月公布的国际标准化组织(ISO)公布ISO/IEC 7498-1所描述的开放式系统互连(OSI)模型的5、6或7层。如果专用线程用来生成按TCP连接的响应分组,输出将很可能在PVM和SVM之间匹配。这就是所谓的单一会话编程模型。另一方面,如果更高OSI层的编程模型使用结合到单一TCP连接的多个线程(多会话编程),那么输出类似性变得成问题的。输出类似性可以变得成问题,因为多会话编程中的每个线程的执行时间可以在PVM和SVM之间是不同的。关于这些挑战,本文所描述的示例是需要的。

[0027] 根据一些第一示例,用于重映射多线程应用的会话的技术可以包括在用于布置成主管能够单独执行应用的PVM或SVM的服务器的电路截取由应用做出的第一和第二套接字API调用。对于这些第一示例,第一和第二套接字API可以布置成通过来自耦合到服务器的客户端的多个TCP连接之中的相同的TCP连接来路由相应的第一和第二会话用于期望输出。第一和第二会话可以由应用响应于从客户端接收的数据来生成。技术也可以包括使用对于第一和第二会话相应的第一和第二标识符信息来生成会话索引,并且然后重映射第一和第二会话以用于通过来自所述多个TCP连接之中的不同的TCP连接的实际输出。第一和第二会话可以被重映射,使得会话索引能够在客户端用来将第一和第二会话与相同的TCP连接重组。

[0028] 在一些第二示例中,用于重映射多线程应用的会话的技术可以包括在经由多个TCP连接耦合到服务器的客户端的电路,经由来自所述多个TCP连接之中的不同的TCP连接来接收第一和第二会话。对于这些第二示例,第一和第二会话可以由服务器主管的PVM或SVM执行的第一应用来生成并且经由所述不同的TCP连接从服务器输出。第一和第二会话可

以是响应于先前由在客户端的第二应用发送到第一应用的请求而生成的。技术也可以包括接收具有对于第一和第二会话的标识符信息的会话索引并且基于该会话索引重编组第一和第二会话至来自所述多个TCP连接之中的相同TCP连接。技术也可以包括经由相同TCP连接转发第一和第二会话到相应的第一和第二套接字API用于第二应用。

[0029] 图1示出示例第一系统。在一些示例中,如图1中所示,第一示例系统包括系统100。系统100包括数据中心105,其具有经由内部网络130来与次节点/服务器120耦合的主节点/服务器110。同样地,如图1中所示,主节点110或次节点120可以经由通过网络140路由的网络(NW)通信信道142耦合到客户端150。

[0030] 根据一些示例,主节点110和次节点120可以被布置为故障容错系统的部分。对于这些示例,主节点110可以被布置来主管PVM 112以及次节点120可以被布置来主管SVM 122。PVM 112和SVM 122都可以能够分别地执行相同的应用101。在一些示例中,应用101可以利用多会话编程模型,其包括将多个会话路由到相同TCP连接。此类型的多会话编程模型在下面更详细地描述。

[0031] 在一些示例中,主节点110和次节点120可以各自地维持相应的心跳(heartbeat) 115和125,以传递对于PVM 112和SVM 122的健康状况。例如,心跳115可以转播对于PVM 112的健康状况信息以便使次节点120能够确定PVM是否已经故障或变得无响应,及从而要求SVM 122故障转移并且成为PVM来服务或处理来自客户端150的请求。类似地,心跳125可以转播对于SVM 122的健康状况信息以便使主节点110能够确定SVM是否已经故障或变得无响应,及从而要求另一个SVM被配置用于提供对于PVM 112的故障容错。

[0032] 根据一些实施例,PVM 112和SVM 122可以在相同网络域0内操作用于处理从客户端150处的应用152而接收的请求。对于这些示例,请求可以通过NW通信信道142而经由在主节点110处的外部NW接口118 来被接收。请求可以然后同时地被路由到在PVM 112和SVM 122被执行的应用101。主节点110处的COL0管理器114和COL0管理器124可以监视由应用101生成的输出/响应。COL0管理器114和124可以阻止输出/响应并且至少暂时存储被阻止的输出/响应于相应的存储116和126中(如果输出/响应不匹配的话)。COL0管理器114和124可以然后强加新的检查点(例如,VM检查点113和123)以促使SVM 122变得与SVM 112同步。

[0033] 在一些示例中,技术可以被执行,其可以减少由于多会话编程导致的在PVM 112和SVM122分开执行的应用101之间可能的输出不等。如下文更多地描述,这些技术可以包括逻辑和/或特征(未在图1中示出),其能够通过重映射那些会话到实际输出的不同的TCP会话,将经由相同的TCP连接的期望输出的会话去耦。对于重映射的TCP会话的实际输出/响应可以然后由COL0管理器114和124来比较以确定是否PVM 112和SVM122处于同步状态。通过COL0管理器114和124的该比较典型地依赖于按TCP的连接。在经由和应用101初始期望路由这些输出/响应相同TCP连接将输出/响应递送到客户端150处的应用152之前,客户端150处的类似去耦逻辑和/或特征可以能够重编组会话。因此,应用101和应用152都可以不知道此去耦。

[0034] 图1描绘一种类型的混合虚拟机管理器(VMM)模型,其中特权顾客操作系统(比如域0)可以被用来运行本地装置驱动器并管理其它顾客。在其它示例中,各类型的VMM模型(比如管理程序模型或基于主管的模型)可以在类似数据中心105的数据中心处被实现。对于这些其它示例,管理程序模型或基于主管的模型可以是II型VMM模型。同时在图1中所示

的混合VMM模型可以是I型VMM模型。示例不是专有地限制于I型或II型VMM模型。对于重映射用于多线程应用的会话的示例可以同等地应用于VMM模型的两种类型。

[0035] 图2示出示例编程模型200。在一些示例中,如在图2中所示,单一会话编程210可以包括用于应用层212、表示层214和会话层216的更高等级层栈处理的单一线程的使用。从此单一线程的输出可以经由或通过TCP层218处的TCP连接215,来在套接字应用接口(API)217处被路由。对于这些示例,TCP分组219可以然后被发送或输出到OSI协议栈(例如网络、数据链或物理层)的更低层,用于至请求源(其导致TCP分组219的生成)的最终递送。单一会话编程210的使用在由PVM或SVM执行的应用被执行时可以很可能导致TCP分组219的类似输出。因此COL0故障容错方案可以与这种类型的单一会话编程一起很好地工作。

[0036] 根据一些实施例,如在图2中所示,多会话编程220可以包括多线程T1、T2、T3和Tn的使用,其中“n”可以是比3大的任何正的完全整数(whole integer)。对于这些示例,线程T1到Tn可以用于应用层222、表示层224和会话层226的更高等级层栈处理。线程T1到Tn的输出可以然后经由或通过TCP层228处的TCP连接225在套接字API 227处被路由。TCP分组229可以然后被发送或输出到OSI协议栈的更低层用于至请求源(其导致TCP分组219的生成)的最终递送。与单一会话编程210相反,多会话编程220的使用在由PVM或SVM执行的应用执行时可以很可能导致TCP分组219的不同输出。如之前所提及,那个差异可以归因于每个线程之间的执行时间中的变化。当用于多会话编程220的线程数量增加时,这些变化将很可能增加。PVM或SVM之间的TCP分组219的输出的变化对于COL0故障容错方案的性能可以是成问题的。

[0037] 图3示出示例第二系统。如图3中所示,示例第二系统包括系统300。在一些示例中,系统300可以包括应用310,其可以由节点或服务器主管的PVM或者SVM所执行。节点或服务器可以位于,例如,在数据中心(比如在图1中示出的数据中心105)。应用310也可以在耦合到服务器的客户端(比如在图1中示出的客户端150)处执行。根据一些示例,应用310可以布置成使用多会话编程模型,其可以包括由相应的线程1、2、3和n正执行的会话312-1、312-2、312-3和312-n。对于这些示例,应用310可以调用初始套接字API 315以路由会话312-1到312-n用于通过相同TCP连接311的期望输出。输出可以相关联或相关于接收的请求(例如,从客户端到服务器或反之从服务器到客户端)。

[0038] 在一些示例中,如在图3中所示,系统300也可以包括TCP去耦器320,其位于对于应用310的会话输出和TCP层330之间。对于这些示例,TCP去耦器320可以包括重映射/重编组逻辑322。重映射/重编组逻辑322可以能够截取初始套接字API 315,其被布置成路由会话312-1到312-n以用于通过TCP连接311的输出。重映射/重编组逻辑322也可以能够得到对于会话312-1到312-n的标识符信息,以生成会话索引324,并且然后重映射会话312-1到312-n以用于通过TCP层330处不同的TCP连接332-1到332-n的实际输出。重映射/重编组逻辑322,例如,可以调用新的套接字API 325来路由会话至312-1到312-n,以用于通过TCP连接332-1到332-n的实际输出来促使TCP分组334通过这些不同的TCP连接来输出。

[0039] 根据一些示例,会话索引324可被传递到服务器或客户端,所述服务器或客户端发起了请求应用,其促使会话312-1到312-n从应用310输出。在那一点上,在至请求应用的递送之前,会话索引324可以能够用在服务器或客户端将会话312-1到312-n与TCP连接311重组。在一些示例中,标识符信息可以包括线程名或标识符,其可以已经被分开指派(例如,

由应用310)给线程1到n的每个。在一些其它示例中,标识符信息可以包括与会话312-1到312-n相关联的协议栈模式,比如与用于生成相应的会话312-1到312-n的执行协议栈模式关联的代码模式(例如包括网际协议(IP)地址)的组合。

[0040] 在一些示例中,应用310可以利用套接字API的动态链接库(未示出)以路由会话312-1到312-n以用于通过包括TCP连接311的多个TCP连接的期望输出。对于这些示例,通过用TCP去耦链接库(未示出)替代动态链接库,重映射/重编组逻辑322可以能够截取初始套接字API 315。所替代的TCP去耦链接库可以然后用来重映射会话312-1到312-n以用于通过TCP连接322-1到322-n的实际输出。去耦链接库可以连同会话索引124一起被传递,以使请求者客户端或服务端能够在递送请求应用之前将会话312-1到312-n与TCP连接311重组合。

[0041] 根据一些示例,虽然未在图3中示出,而不是在应用310和TCP层330之间配置TCP去耦器320。TCP去耦器320可以是嵌入的内部层4(TCP)软件,其可以被包括在TCP层330中。例如,嵌入在内核TCP/IP栈。

[0042] 根据一些示例,使用重编组策略来包括,但不限制于,循环(round-robin)策略,请求者客户端可以能够重编组会话312-1到312-n至TCP连接311。同样地,在归因于主节点故障(例如硬件故障)的故障转移的情形中,内部TCP错误控制机制可以被利用来经由SVM的使用而恢复故障的PVM的TCP状态,并且还在按TCP连接的基础(其可以然后根据重编组策略而重编组)上递送响应分组。

[0043] 在一些示例中,在正被监视的输出是对于正使用多会话编程模型的应用时,会话312-1到312-n的重映射可以对于在主节点或次节点被布置成主管PVM或SVM的COLO管理器,改善按TCP输出类似性。此外,在一些示例中,会话312-1到312-n可以被映射成TCP连接的集合,而不是如在图3中所示的1:1映射。例如,从TCP连接311重映射可以包括根据m:1的比率重映射,其中“m”是比1大的任何正的完全整数值并且出自会话312-1到312-n的m个会话可以被映射以共享出自TCP连接332-1到332-n的单一TCP连接。从TCP连接311重映射也可以包括根据m:o的比率的重映射,其中“o”是比m大的任何正的完全整数值。对于m:o的比率,出自会话312-1到312-n的m个会话可以被映射至来自TCP连接332-1到332-n之中的o TCP连接。1:1、m:1或m:o的这些比率可以基于可以利用这些重映射方案的系统的需要来灵活地建立。例如,TCP连接在给定系统中可以是被限制或富有的,并且也可以基于给定系统的需求来变动。因此,符合给定系统的需要的比率可以因此来调节。

[0044] 图4示出示例第三系统。如图4中所示,示例第三系统包括系统400。在一些示例中,如在图4中所示,系统400 包括客户端410,其经由NW通信信道430耦合到服务器420。对于这些示例,客户端410可以包括应用412、TCP去耦器414和操作系统(OS)TCP栈416。同样地,服务器420可以包括应用422、TCP去耦器424和OS TCP栈426。服务器420,例如,可以布置成主管PVM或SVM(未示出),所述PVM或SVM可以能够执行应用422。

[0045] 根据一些示例,应用412和422都可以包括至少一个应用,其利多会话编程模型,所述模型调用相应的APP_套接字411和421以通过相同的TCP连接路由多个会话。类似于对于图3的上述的TCP去耦器320,TCP去耦器414和424可以能够经由不同的TCP连接重映射会话用于向相应的OS_套接字415的实际输出。例如,TCP去耦器424的重映射/重编组逻辑423可以截取这些会话,得到标识符信息,生成会话索引405并且然后经由OS TCP栈426处的不同的TCP连接重映射会话用于向OS_套接字425的实际输出。OS TCP栈426可以然后通过NW通信

信道430来将响应TCP分组路由到客户端410。TCP去耦器414的重编组逻辑413可以然后促使重映射至不同TCP连接的会话通过应用422初始映射这些会话的相同TCP连接来被重编组。会话索引405可以由TCP去耦器414使用,以由相同TCP连接来重编组会话用于递送到应用412。

[0046] 在一些示例中,重映射/重编组逻辑423可以能够有选择地截取和重路由应用422生成的会话。例如,应用422可以包括执行单一会话编程模型的第一应用和执行多会话编程模型的第二应用。重映射/重编组逻辑423可以有选择地仅截取第二应用用于重映射。重映射/重编组逻辑423可以不选取第一应用,因为单一会话编程不经受很可能是使用多个线程的多会话编程的情形的可能的全异输出。因此,重映射可以不被需要用于单一会话编程以便改善由服务器420主管的PVM和SVM之间的输出类似性。

[0047] 在一些其它示例中,应用422可以使用多个TCP连接并且仅这些多个TCP的一部分可以由重映射/重编组逻辑423来截取和重映射。例如,如果一个TCP连接与单一会话编程模型一起来使用,而另一个TCP连接与多会话编程模型一起来使用。对于这些其它的示例,仅与多会话编程模型一起使用的其它TCP连接可以被截取。

[0048] 图5示出示例TCP流程500。在一些示例中,TCP流程500可以是TCP编程接口的范例并且可以描绘TCP客户端510和TCP服务器520之间通信的流程。如图5中所示,TCP流程500可以包括TCP编程的各种阶段或时期,以包括TCP连接设置505、数据交换515和TCP拆除525。根据一些示例,TCP去耦器(比如TCP去耦器320或TCP去耦器413/423)可以能够API模仿以便截取套接字API,其被布置成路由多个会话用于向相同TCP连接输出。基于各个阶段的哪个阶段包括该API模仿所针对的TCP连接设置505、数据交换515和TCP拆除525,该API模仿可以发生。

[0049] 根据一些示例,对于TCP连接设置505处的API模仿,TCP去耦器可以截取用于多个会话的套接字API(在图5中示出为套接字O)并且立即创建多个OS_socket API用于这些截取的套接字API。例如,os_socket_x(x=1、2、3...n),并且然后与app_socket_1一起返回应用。对于这些示例,app_socket_1可以仅在TCP去耦器中有效。应用也可以创建多个其它app_socket_x(x=2、3...n)。通过向os_socket_x(x=1、2、...、n)的OS_socket层的bind()、listen()、connect()和accept()的操作,相同策略可以来应用到app_socket_1的bind()、listen()、connect()和accept()API访问。

[0050] 在一些示例中,TCP连接设置505处的API模仿也可以包括去耦器以便使用内部软件状态纯粹地模仿socket()、bind()、listen()、connect()和accept()API,而没有立即涉及os_socket层访问。反而,一旦它检测到TCP连接设置505被完成(例如,TCP 3-way 信号交换被完成,使用os_socket层的socket()、bind()、listen()、connect()和accept()API,TCP去耦器开始设置os_socket层连接os_socket_x(x=1、2、3...n)并且然后与在socket_1(在socket()中返回)中返回的app_socket_1一起映射。在一些示例中,在TCP拆除525处,TCP去耦器可以然后移除映射到os_socket_x的多个会话,以响应于TCP服务器520读取的从TCP客户端510发送的文件结束(EOF)消息。

[0051] 根据一些示例,数据交换515处的API模仿可以包括(对于输出的数据会话)使用当前的[对于给定线程的标识符和/或对于给定线程的名和/或调用者功能IP地址]作为唯一会话索引的TCP去耦器。TCP去耦器可以使用会话索引来寻址固定的映射表(例如线性表)如

果os_socket_x连接ID,或者使用具有碰撞链的散列表以便从会话到os_socket映射。对于这些示例,顾客OS任务管理组件也可以被挂钩以提供对于每个线程的唯一标识符。一旦os_socket_x被选择,TCP去耦器可以使用os_socket_x的write O API来发送会话数据(例如,会话索引)。在另一方面,一旦使用read O从多个TCP连接接收会话数据,客户端510可以组合它们在一起并且递送重编组的会话到客户端510处的会话层。

[0052] 图6示出对于设备600的示例框图。如在图6中所示,第一设备包括设备600。虽然图6中所示的设备600在某一拓扑中具有元件的限制数量,但是可以领会,设备600在备选拓扑中可以包括如对于给定实现所希望的或多或少的元件。

[0053] 设备600可以由在服务器或客户端计算装置处维持的电路620来支持。电路620可以布置成执行一个或更多软件或固件实现模块或组件622-a。值得注意的是,如本文所使用的“a”“b”“c”和类似的符号旨在表示任何正整数的变量。因此,例如,如果实施方式设置a值=6,那么对于组件622-a的软件或固件的完整集合可以包括组件622-1、622-2、622-3、622-4、622-5或622-6。示出的示例不限制于这个上下文并且通篇使用的不同变量可以表示相同或不同的整数值。同样地,这些“组件”可以是存储在计算机可读媒体中的软件/固件,并且虽然组件在图6中示为离散框,但是这不限制这些组件到分离的计算机可读媒体组件(例如分开的存储器)中存储。

[0054] 根据一些示例,电路620可以包括处理器或处理器电路。电路620可以是在服务器或客户端(例如主节点/服务器110、次节点/服务器120、服务器420或客户端410)处可以包括处理核或元件的电路的一部分。包括一个或更多处理核的电路能够是任何各种商业上可以得到的处理器,包括不限制于AMD® Athlon®、Duron® 和Opteron® 处理器;ARM® 应用、嵌入和安全处理器;IBM® 和Motorola® DragonBall® 和PowerPC处理器;IBM和SONY® Cell处理器;Intel® Atom®、Celeron®、Core (2) Duo®、Core i3、Core i5、Core i7、Itanium®、Pentium®、Xeon®、Xeon Phi® 和XScale® 处理器,以及类似处理器。根据一些示例,电路620也可以包括专用集成电路(ASIC)和至少一些组件622-a可以作为ASIC的硬件元件来实现。

[0055] 根据一些第一示例,设备600可以在服务器经由多个TCP连接615与客户端耦合,服务器可以布置成主管能够单独地执行第一应用的PVM或SVM。对于这些第一示例,设备600可以包括截取组件622-1。截取组件622-1可以通过电路620来执行以便截取由第一应用做出的第一和第二套接字API调用。在一些示例中,第一和第二套接字API可以包括于套接字API 605中。套接字API 605可以布置成路由第一应用生成的会话至来自TCP连接605之中的相同TCP连接。会话是响应于从在客户端的第二应用接收的请求而生成的。

[0056] 根据一些第一示例,设备600也可以包括索引组件622-2。索引组件622-2可以由电路620来执行以便使用对于由第一应用生成的会话的标识符信息来生成会话索引。对于这些示例,索引组件622-2可以在包括会话索引624-a的数据结构(例如,查找表(LUT)中维持或存储所生成的会话索引。同样地,所生成的会话索引可以经由会话索引610被传递至客户端。

[0057] 在第一示例中,设备600也可以包括重映射组件622-3。重映射组件622-3可以由电路620来执行以便重映射由第一应用生成的会话以用于通过来自TCP连接615之中的不同的TCP连接的实际输出。对于这些示例,会话可以被重映射,使得会话索引610可以在客户端被

使用以将第一和第二会话与相同TCP连接重组合,用于向在客户端的第二应用的递送。在一些示例中,重映射组件622-3可以能够使用去耦链接库625-b(例如,在LUT中维持)以便重映射会话到所述不同的TCP连接。

[0058] 根据一些第二示例,设备600可以在客户端经由TCP连接615被耦合到服务器。类似于第一示例,服务器可以布置成主管能够单独地执行第一应用的PVM或SVM。对于这些第二示例,设备600也可以包括接收组件622-4。接收组件622-4可以由电路620来执行以便经由来自TCP连接615之中的不同的TCP连接接收由第一应用生成的第一和第二会话。第一和第二会话可以响应于由在客户端的第二应用先前发送到第一应用的请求而已被生成。

[0059] 根据一些第二示例,在客户端的设备600也可以包括如上提及的用于第一示例的索引组件622-2。索引组件622-2对于这些第二示例可以接收具有对于第一和第二会话的标识符信息的会话索引610。索引组件622-2,例如,可以在数据结构(比如,包括会话索引624-a的LUT)中维持或存储所接收的会话索引610。

[0060] 根据一些第二示例,设备600也可以包括重编组组件622-5。基于在会话索引624-a处存储或维持的会话索引,重编组组件622-5可以由电路620来执行以便重编组第一和第二会话至来自TCP连接615之中的相同TCP连接。

[0061] 根据一些第二示例,设备600也可以包括转发组件622-6。转发组件622-6可以由电路620来执行以便经由相同的TCP连接转发第一和第二会话到相应的第一和第二套接字API以用于在客户端的第二应用。

[0062] 本文所包括的是表示用于施行所公开构造的新颖方面的示例方法论的逻辑流程的集合。为了解释的简单的目的,当本文所示的一个或更多方法论示出和描述为一系列行为时,本领域技术人员将理解和领会,方法论不限制于行为的顺序。一些行为可以根据其以不同顺序发生和/或与来自本文所描述或图示的行为的其它行为同时发生。例如,本领域技术人员将理解和领会,方法论能备选地被表示为一系列相互关连的状态或事件,比如在状态框图中。而且,并不是方法论中图示的全部的行为可以被要求用于新颖的实施方式。

[0063] 逻辑流程可以以软件、固件和/或硬件来实现。在软件和固件实施例中,逻辑流程可以通过计算机可以执行指令来实现,计算机可以执行指令存储在至少一个非瞬时计算机可读媒体或机器可读媒体上,比如光、磁或半导体存储。实施例不限制于此上下文。

[0064] 图7示出第一逻辑流程的示例。如在图7中所示,第一逻辑流程包括逻辑流程700。逻辑流程700可以表示由本文所描述的一个或更多的逻辑、特征或装置(比如设备600)来执行的一些或全部操作。更特别的是,逻辑流程700可以由至少截取组件622-1、索引组件622-2或重映射组件622-3来实现。

[0065] 根据一些示例,在布置成主管能够单独执行应用的主或次VM的服务器的电路,框702处的逻辑流程700可以截取由应用做出的第一和第二套接字API调用。第一和第二套接字API布置成路由相应的第一和第二会话以用于通过来自耦合到服务器的客户端的多个TCP连接之中的相同TCP连接的期望输出。第一和第二会话由应用响应从客户端接收的数据而生成。对于这些示例,截取组件622-1可以截取第一和第二套接字API。

[0066] 在一些示例中,框图704处的逻辑流程700可以使用对于第一和第二会话的相应的第一和第二标识符信息来生成会话索引。对于这些示例,索引组件622-2可以生成会话索引。

[0067] 根据一些示例,框图706处的逻辑流程700可以重映射第一和第二会话用于通过来自所述多个TCP连接之中的不同的TCP连接的实际输出,使得会话索引能够在客户端用来将第一和第二会话与相同TCP连接重组合。对于这些示例,重映射组件622-3可以重映射第一和第二会话。

[0068] 图8示出第二逻辑流程的示例。如在图8中所示,第二逻辑流程包括逻辑流程800。逻辑流程800可以表示通过本文所描述的一个或更多的逻辑、特征或装置(比如设备600)来执行一些或全部操作。更特别的是,逻辑流程800可以通过至少接收组件622-4、索引组件622-2、重编组组件622-5或转发组件622-6来实现。

[0069] 根据一些示例,在布置成主管能够单独执行应用的主或次VM的服务器,框802处的逻辑流程800可以经由与客户端的不同的TCP连接接收第一和第二会话。第一和第二会话是在客户端的第二应用响应于先前从第一应用发送到第二应用的请求而生成的。对于这些示例,接收组件622-4可以接收第一和第二会话。

[0070] 在一些示例中,框图804处的逻辑流程800可以接收具有对于第一和第二会话的标识符信息的会话索引。对于这些示例,索引组件622-2可以接收会话索引。

[0071] 根据一些示例,框图806处的逻辑流程800可以基于会话索引重编组第一和第二会话到相同的TCP连接。对于这些示例,重编组组件622-5可以重编组第一和第二会话。

[0072] 在一些示例中,框图808处的逻辑流程800可以经由相同的TCP连接转发第一和第二会话到相应的第一和第二套接字API用于第一应用。对于这些示例,转发组件622-6可以转发第一和第二会话到在服务器的第一应用。

[0073] 图9示出第三逻辑流程的示例。如在图10中所示,第三逻辑流程包括逻辑流程900。逻辑流程900可以表示由本文所描述的一个或更多的逻辑、特征或装置(比如设备600)来执行的一些或全部操作。更特别的是,逻辑流程900可以由至少接收组件622-4、索引组件622-2、重编组组件622-5或转发组件622-6来实现。

[0074] 根据一些示例,在经由多个TCP连接耦合到服务器的客户端的电路,框图902处的逻辑流程900可以经由来自所述多个TCP连接之中的不同的TCP连接接收第一和第二会话。第一和第二会话由服务器主管的主或次虚拟机来执行的第一应用生成以及经由所述不同的TCP连接从服务器输出。第一和第二会话是响应先前由在客户端的第二应用发送到第一应用的请求而生成的。对于这些示例,接收组件622-4可以接收第一和第二会话。

[0075] 在一些示例中,框图904处的逻辑流程900可以接收具有对于第一和第二会话的标识符信息的会话索引。对于这些示例,索引组件622-2可以接收会话索引。

[0076] 根据一些示例,框图906处的逻辑流程900可以基于会话索引重编组第一和第二会话到相同的TCP连接。对于这些示例,重编组组件622-5可以重编组第一和第二会话。

[0077] 在一些示例中,框图908处的逻辑流程900可以经由相同的TCP连接转发第一和第二会话到相应的第一和第二套接字API用于第二应用。对于这些示例,转发组件622-6可以转发第一和第二会话到在客户端的第二应用。

[0078] 图10示出第一存储媒体的示例。如图11中所示,第一存储媒体包括存储媒体1000。存储媒体1000可以包括制造的物品。在一些示例中,存储媒体1000可以包括任何非瞬时计算机可读媒体或机器可读媒体,比如光,磁或半导体存储。存储媒体1000可以存储各种类型的计算机可执行指令,比如实现逻辑流程700、800或900的指令。计算机可读或机器可读存

储媒体的示例可以包括能够存储电子数据的媒体,包括易失存储器或非易失存储器、可移除或非可移除存储器、可擦除或不可擦除存储器、可写或可重写存储器等等的任何有形。计算机可以执行指令的示例可以包括任何适合类型的代码,比如源代码、编译代码、直译码、可执行代码、静态代码、动态代码、面向对象的代码、视码等等。示例不限制于此上下文。

[0079] 图11图示示例计算平台1100。在一些示例中,如在图11中所示,计算平台1100可以包括处理组件1140、其它平台组件1150或通信接口1160。根据一些示例,计算平台1100可以在能够经由NW通信信道耦合的服务器或客户端计算装置来实现。

[0080] 根据一些示例,处理组件1140可以执行处理操作或逻辑用于设备600和/或存储媒体1000。处理组件1140可以包括各种硬件元件、软件元件或两者的组合。硬件元件的示例可以包括装置、逻辑装置、组件、处理器、微处理器、电路、处理器电路、电路元件(例如晶体管、电阻器、电容器、电感器,等等)、集成电路、专用集成电路(ASIC)、可以编程逻辑电路(PLD)、数字信号处理器(DSP)、现场可编程门阵列(FPGA)、存储器单元、逻辑门、寄存器、半导体装置、芯片、微芯片、芯片集合,等等。软件元件的示例可以包括软件组件、程序、应用、计算机程序、应用程序、装置驱动器、系统程序、软件发展程序、机器程序、操作系统软件、中间件、固件、软件模块、例程、子程序、功能、方法、例程、软件接口、应用程序接口(API)。指令集合、计算代码、计算机代码、代码部、计算机代码部、字、值、符号或其任何组合。确定是否示例使用硬件元件和/或软件元件来实现可以根据任何数量的因子(比如希望的计算的速率、功率等级、热容错、处理循环预算、输入数据速率、输出数据速率、存储器资源、数据总线速度和其它设计或性能限制,如对于给定示例所希望的)而改变。

[0081] 在一些示例中,其它平台组件1150可以包括普通计算元件,比如一个或更多处理器、多核处理器、联合处理器、存储器单元、芯片集合、控制器、外围设备、接口、振荡器、定时装置、视频卡、音频卡、多媒体输入/输出(I/O)组件(例如,数字显示器)、电源,等等。存储器单元的示例可以包括不限制于以一个或更高速度存储器单元形式的各种类型的计算可读和机器可读存储媒体,比如只读存储器(ROM)、随机存取存储器(RAM)、动态RAM(DRAM)、双数据速率DRAM(DDRAM)、同步DRAM(SDRAM)、静态RAM(SRAM)、可编程ROM(PROM)、可擦除可编程ROM(EPROM)、电可擦除可程序ROM(EEPROM)、闪存存储器、聚合物存储器(比如铁电聚合物存储器)、双向存储器、相变或铁电存储器、硅氧化氮氧化硅(SONOS)存储器、磁或光卡、比如独立磁碟冗余阵列(RAID)驱动器的装置阵列、固态存储器装置(例如,USB存储器)固态驱动器(SSD)以及任何其它类型的适合存储信息的存储媒体。

[0082] 在一些示例中,通信接口1160可以包括逻辑和/或特征以支持通信接口。对于这些示例,通信接口1160可以包括一个或更多通信接口,其根据各种通信协议或标准来操作以通过直接或间接网络通信链路来通信。直接通信可以经由在一个或更多工业标准(包括蜕变产物和变型)(比如与PCIe规范相关联的那些)中所描述的通信协议或标准的使用而发生。网络通信可以经由通信协议或标准(比如由IEEE发布的一个或更多以太网标准中所描述的那些)的使用而发生。例如,一个这样的以太网标准可以包括IEEE 802.3。网络通信也可以根据一个或更多OpenFlow规范(比如OpenFlow 硬件抽象API规范)而发生。

[0083] 如上述,计算平台1100可以在服务器或客户端计算装置中实现。因此,本文所描述的计算机平台1100的功能和/或具体配置可以在计算平台1100的各种实施例中来包括或省略,如对于服务器或客户端计算装置的合适地希望。

[0084] 计算平台1100的组件和特征可以使用离散电路、专用集成电路(ASIC)、逻辑门或单一芯片构造的任何组合来实现。此外,计算平台1100的特征可以在适合地恰当之处使用微控制器、可编程逻辑阵列和/或微处理器或前述的任何组合来实现。注意的是,硬件、固件和/或软件元件可以共同地或个别地本文指为“逻辑”或“电路”。

[0085] 应领会,图11的框图中所示的示范性计算平台1100可以表示多个潜在的实施方式的一个功能上描述示例。因此,附图中描绘的块功能的分开、省略或包括不推断对于实现这些功能的硬件组件、电路、软件和/或元件将会必要地在实施例中被分开、省略或包括。

[0086] 至少一个示例的一个或更多方面可以通过存储在至少一个机器可读媒体上的代表指令来实现,该机器可读媒体表示处理器内的各种逻辑,当其由机器读取时,计算装置或系统促使机器、计算装置或系统制作逻辑以便执行本文所描述的技术。这种已知为“IP核”的表示可以存储在有形的、机器可读媒体上并且提供给各种顾客或制造设备以便装载到实际上制造逻辑或处理器的制作机器。

[0087] 各种示例可以使用硬件元件、软件元件或两者的组合来实现。在一些示例中,硬件元件可以包括装置、组件、处理器、微处理器、电路、电路元件(例如晶体管、电阻器、电容器、电感器,等等)、集成电路、专用集成电路(ASIC)、可以编程逻辑装置(PLD)、数字信号处理器(DSP)、现场可编程门阵列(FPGA)、存储器单元、逻辑门、寄存器、半导体装置、芯片、微芯片、芯片集合,等等。在一些示例中,软件元件可以包括软件组件、程序、应用、计算机程序、应用程序、系统程序、机器程序、操作系统软件、中间件、固件、软件模块、例程、子程序、功能、方法、例程、软件接口、应用程序接口(API)、指令集合、计算代码、计算机代码、代码部、计算机代码部、字、值、符号或其任何组合。确定示例是否使用硬件元件和/或软件元件来实现可以依照任何数量的因子(比如希望的计算的速率、功率等级、热容错、处理循环预算、输入数据速率、输出数据速率、存储器资源、数据总线速度和其它设计或性能限制,如对于给定示例所希望的)而改变。

[0088] 一些示例可以包括制造的物品或至少一个计算机可读媒体。计算机可读媒体可以包括存储逻辑的非瞬时存储媒体。在一些示例中,非瞬时存储媒体可以包括能够存储电子数据一个或更多类型的计算机可读存储媒体,其包括易失存储器或非易失存储器,可移除或非可移除存储器、可擦除或不可擦除存储器,可写或可重写存储器,等等。在一些示例中,逻辑可以包括各种软件元素,比如软件组件、程序、应用、计算机程序、应用程序、系统程序、机器程序、操作系统软件、中间件、固件、软件模块、例程、子程序、功能、方法、例程、软件接口、API、指令集合、计算代码、计算机代码、代码部、计算机代码部、字、值、符号或其任何组合。

[0089] 根据一些实施例,计算机可读媒体可以包括非瞬时存储媒体,以存储或维持指令,当其由机器执行时,计算装置或系统促使机器、计算装置或系统根据所描述的示例来执行方法和/或操作。指令可以包括任何适合类型的代码,比如源代码、编译代码、直译码、可执行代码、静态代码、动态代码等等。指令可以根据预定义的计算机语言、方式或语法来实现以用于指导机器、计算装置或系统执行某一功能。指令可以使用任何适合的高级、低级、面向对象、视觉、编译和/或直译编程语言来实现。

[0090] 一些示例可以使用表述“一个示例中”或“示例”连同它们的派生来描述。这些术语表示与示例有关联的所描述的特定特征、结构或特性包括于至少一个示例中。说明书中的

各位置中的短语“一个示例中”的出现不必全部指的是相同示例。

[0091] 一些示例可以使用表述“耦合”和“连接”连同它们的派生词来描述。这些术语对于彼此不必意为同义词。例如,使用术语“连接”和/或“耦合”的描述可以指示两个或更多元件处于彼此直接的物理或电接触。然而,术语“耦合”也可以表示两个或更多元件不处于彼此直接的接触,但是仍一直彼此合作或互相作用。

[0092] 下面示例属于本文所公开的技术的附加示例。

[0093] 示例1. 一种示例设备可以包括在经由多个TCP连接而耦合到客户端的服务器的电路。对于这些示例,服务器可以被布置成主管能够单独执行第一应用的主或次VM。设备也可以包括截取组件用于由电路执行以截取由第一应用做出的第一和第二套接字API调用。第一和第二套接字API可以布置成路由相应的第一和第二会话以用于通过来自所述多个TCP连接之中的相同TCP连接的期望输出。第一和第二会话可以由第一应用响应于从在客户端的第二应用接收的请求而生成。设备也可以包括索引组件,用于由电路执行以使用对于第一和第二会话的相应的第一和第二标识符信息来生成会话索引。设备也可以包括重映射组件,用于由电路执行以重映射第一和第二会话以用于通过来自所述多个TCP连接之中的不同的TCP连接的实际输出,使得会话索引能够在客户端用来将第一和第二会话与相同TCP连接重组以用于向第二应用的递送。

[0094] 示例2. 示例1的设备也可以包括截取组件通过替代由该应用用于路由会话以用于通过所述多个TCP连接向客户端的期望输出的套接字API的动态链接库来截取第一和第二套接字API。对于这些示例,动态链接库可以使用TCP去耦链接库来替代。设备也可以包括重映射组件通过使用TCP耦合链接库来重映射第一和第二会话以促使第一和第二会话通过所述不同的TCP连接被实际输出到客户端。

[0095] 示例3. 示例1的设备,截取组件可以通过截取组件基于与客户端的TCP连接设置或与客户端的TCP数据交换之一来模仿第一和第二套接字API,从而截取第一和第二套接字API。

[0096] 示例4. 示例3的设备也包括截取组件基于TCP连接设置来模仿第一和第二套接字API。对于这些示例,截取组件可以创建对于第一和第二套接字API的多个OS套接字API并且指派所述多个OS套接字API到所述不同的TCP连接。示例3的设备也可以包括重映射组件重映射第一和第二会话以使用多个OS套接字API以促使第一和第二会话通过所述不同的TCP连接被实际输出到客户端。

[0097] 示例5. 示例4的设备也可以包括截取组件响应于与客户端的TCP拆除的EOF消息而移除对于第一和第二套接字API的所述多个OS套接字API。

[0098] 示例6. 示例3的设备也可以包括截取组件基于TCP数据交换来模仿第一和第二套接字API,并且使用会话索引来寻址的固定映射表,该固定映射表将第一和第二会话映射到指派给所述不同的TCP连接的多个OS套接字API。示例3的设备也可以包括重映射组件基于固定的映射表重映射第一和第二会话以促使第一和第二会话通过所述不同的TCP连接被实际输出到客户端。

[0099] 示例7. 示例1的设备,截取组件可基于该应用有选择地被选取用于截取而能够由主或次VM执行的另一个应用没有被选取用于截取来截取第一和第二套接字API。

[0100] 示例8. 示例1的设备,第一和第二标识符信息可以包括指派给用来生成第一会话

的第一线程的第一线程名或标识符以及指派给用来生成第二会话的第二线程的第二线程名或标识符。

[0101] 示例9. 示例1的设备, 第一和第二标识符信息可以包括第一协议栈模式(其包括与第一会话的生成关联的第一IP地址)和第二协议栈模式(其包括与第二会话的生成关联的第二IP地址)。

[0102] 示例10. 示例1的设备也可以包括耦合到电路来呈现用户接口观看的数据显示器。

[0103] 示例11. 一种示例方法可以包括在布置成主管能够单独地执行应用的主或次VM的服务器的电路截取由应用做出的第一和第二套接字API调用。对于这些示例, 第一和第二套接字API可以布置成路由相应的第一和第二会话以用于通过来自耦合到服务器的客户端的多个TCP连接之中的相同TCP连接的期望输出。第一和第二会话可以由该应用响应于从客户端接收的数据来生成。方法也可以包括使用对于第一和第二会话相应的第一和第二标识符信息来生成会话索引。方法也可以包括重映射第一和第二会话以用于通过来自所述多个TCP连接之中的不同的TCP连接的实际输出, 使得会话索引能够在客户端用来将第一和第二会话与相同TCP连接重组。

[0104] 示例12. 截取第一和第二套接字API的示例11的方法可以包括替代由应用用于路由会话以用于通过所述多个TCP连接向客户端的期望输出的套接字API的动态链接库。对于这些示例, 动态链接库可以使用TCP去耦链接库来替代。同样地对于示例11的方法, 重映射第一和第二会话可以包括使用TCP耦合链接库来促使第一和第二会话通过所述不同的TCP连接被实际输出到客户端。

[0105] 示例13. 示例11的方法, 截取第一和第二套接字API可以包括基于与客户端的TCP数据交换或与客户端的TCP连接设置之一来模仿第一和第二套接字API。

[0106] 示例14. 示例13的方法, 基于一个TCP连接设置来模仿第一和第二套接字API可以包括创建对于第一和第二套接字API的多个OS套接字API。对于这些示例, 所述多个OS套接字API可以指派给所述不同的TCP连接。示例13的方法也可以包括重映射第一和第二会话, 使得使用多个OS套接字API可以促使第一和第二会话通过所述不同的TCP连接被实际输出到客户端。

[0107] 示例15. 示例14的方法可以包括响应于与客户端的TCP拆除的EOF消息而移除对于第一和第二套接字API的所述多个OS套接字API。

[0108] 示例16. 示例13的方法, 基于与TCP数据交换来模仿第一和第二套接字API可以包括使用会话索引来寻址固定映射表, 其将第一和第二会话映射到指派给所述不同的TCP连接的多个OS套接字API。对于示例13的方法, 重映射第一和第二会话可以包括使用固定映射表来促使第一和第二会话通过所述不同的TCP连接被实际输出到客户端。

[0109] 示例17. 示例11的方法, 截取第一和第二套接字API可以包括基于应用有选择地被选取用于截取而能够由主或次VM执行的另一个应用没有被选取用于截取来进行截取。

[0110] 示例18. 示例11的方法, 第一和第二标识符信息可以包括指派给用来生成第一会话的第一线程的第一线程名或标识符以及指派给用来生成第二会话的第二线程的第二线程名或标识符。

[0111] 示例19. 示例11的方法, 第一和第二标识符信息可以包括第一协议栈模式(其具有与第一会话的生成关联的第一IP地址)和第二协议栈模式(其包括与第二会话的生成关

联的第二IP地址)。

[0112] 示例20.至少一个机器可读媒体可以包括多个指令,所述指令响应于在服务器的系统执行而促使所述系统实行根据示例11到19的任何一个的方法。

[0113] 示例21.一种设备可以包括用于执行示例11到19的任何一个的方法的部件。

[0114] 示例22.示例至少一个机器可读媒体可以包括多个指令,所述指令其响应于在布置成主管能够单独执行第一应用的主或次VM的服务器的系统来执行而可促使所述系统经由与客户端的不同的TCP连接来接收第一和第二会话。第一和第二会话可以由在客户端的第二应用响应于从第一应用先前发送到第二应用的请求而生成。所述指令也可以促使所述系统接收具有对于第一和第二会话的标识符信息的会话索引。所述指令也可以促使所述系统基于会话索引重编组第一和第二会话到相同的TCP连接。所述指令也可以促使所述系统经由相同的TCP连接转发第一和第二会话到相应的第一和第二套接字API以用于第一应用。

[0115] 示例23.示例22的至少一个机器可读媒体,标识符信息可以包括指派给用来生成第一应用的第一会话的第一线程的第一线程名或标识符以及指派给用来生成第一应用的第二会话的第二线程的第二线程名或标识符。

[0116] 示例24. 示例22的至少一个机器可读媒体,标识符信息可以包括第一协议栈模式(其包括与第一会话的生成关联的第一IP地址)以及第二协议栈模式(其包括与第二会话的生成关联的第二IP地址)。

[0117] 示例25. 示例22的至少一个机器可读媒体,所述指令还可以促使所述系统使用如服务器用来在服务器将第一和第二会话从相同TCP连接重映射到所述不同的TCP连接的相同循环策略来重编组第一和第二会话到相同TCP连接。

[0118] 示例26.一种示例设备可以包括在经由多个TCP连接耦合到服务器的客户端的电路。对于这些示例,服务器可以布置成主管能够单独执行第一应用的主或次VM。设备也可以包括接收组件用于由电路执行以经由来自所述多个TCP之中的不同的TCP连接来接收由第一应用生成的第一和第二会话。第一和第二会话可响应于在客户端的第二应用先前发送到第一应用的请求而生成。设备也可以包括索引组件,用于由电路执行以接收具有对于第一和第二会话的标识符信息的会话索引。设备也可以包括重编组组件,用于由电路执行以基于会话索引重编组第一和第二会话至来自所述多个TCP连接之中相同TCP连接。设备也可以包括转发组件,用于由电路执行以经由该相同TCP连接转发第一和第二会话到相应的第一和第二套接字API以用于第二应用。

[0119] 示例28.示例26的设备,标识符信息可以包括指派给用来生成第一应用的第一会话的第一线程的第一线程名或标识符以及指派给用来生成第一应用的第二会话的第二线程的第二线程名或标识符。

[0120] 示例29. 示例26的设备,标识符信息可以包括第一协议栈模式(其包括与第一会话的生成关联的第一IP地址)和第二协议栈模式(其包括与第二会话的生成关联的第二IP地址)。

[0121] 示例30. 示例26的设备也可以包括耦合到处理电路用来呈现用户接口观看的数字显示器。

[0122] 示例31.一种示例方法可以包括在经由多个TCP连接耦合到服务器的客户端的电路,经由来自所述多个TCP连接之中的不同的TCP连接接收第一和第二会话。对于这些示例,

第一和第二会话可以由服务器主管的主或次虚拟机执行的第一应用来生成,并且经由所述不同的TCP连接从服务器输出。第一和第二会话可响应于在客户端的第二应用先前发送到第一应用的请求而生成。方法也可以包括接收具有对于第一和第二会话的标识符信息的会话索引。方法也可以包括基于会话索引重编组第一和第二会话至来自所述多个TCP连接之中相同的TCP连接。方法也可以包括经由相同的TCP连接转发第一和第二会话到相应的第一和第二套接字API以用于第二应用。

[0123] 示例32. 示例31的方法,标识符信息可以包括指派给用来生成第一应用的第一会话的第一线程的第一线程名或标识符以及指派给用来生成第一应用的第二会话的第二线程的第二线程名或标识符。

[0124] 示例33. 示例31的方法,标识符信息可以包括第一协议栈模式(其包括与第一会话的生成关联的第一IP地址)和第二协议栈模式(其包括与第二会话的生成关联的第二IP地址)。

[0125] 示例34. 示例31的方法,重编组第一和第二会话会话到该相同的TCP连接可以包括使用如服务器用于将第一和第二会话在服务器从该相同TCP连接重映射到所述不同的TCP连接的相同循环策略。

[0126] 示例35. 至少一个机器可读媒体,其包括多个指令,所述指令响应于在耦合到服务器的客户端的系统来执行而可促使所述系统实行根据示例31到34的任何一个的方法。

[0127] 示例36. 一种设备可以包括用于执行示例31到34的任何一个的方法的部件。

[0128] 示例37. 至少一个机器可读媒体可以包括多个指令,所述指令响应于在经由多个TCP连接耦合到服务器的客户端的系统来执行,服务器可以布置成主管能够单独执行第一应用的主或次VM。所述指令可促使所述系统经由来自所述多个TCP连接之中的不同的TCP连接来接收第一和第二会话。对于这些示例,第一和第二会话可以由第一应用来生成并且经由所述不同的TCP连接从服务器输出。第一和第二会话可响应于由在客户端的第二应用先前发送到第一应用的请求而生成。所述指令也可以促使所述系统接收具有对于第一和第二会话的标识符信息的会话索引。所述指令也可以促使所述系统基于会话索引来重编组第一和第二会话至来自所述多个TCP连接之中的相同TCP连接。所述指令也可以促使所述系统经由该相同TCP连接转发第一和第二会话到相应的第一和第二套接字API以用于第二应用。

[0129] 示例38. 示例37的至少一个机器可读媒体,标识符信息可以包括指派给用来生成第一应用的第一会话的第一线程的第一线程名或标识符以及指派给用来生成第一应用的第二会话的第二线程的第二线程名或标识符。

[0130] 示例39. 示例37的至少一个机器可读媒体,标识符信息可以包括第一协议栈模式(其包括与第一会话的生成关联的第一IP地址)和第二协议栈模式(其包括与第二会话的生成关联的第二IP地址)。

[0131] 示例40. 示例37的至少一个机器可读媒体,重编组到该相同TCP连接的第一和第二会话会话可以包括使用如服务器用于将第一和第二会话在服务器从该相同TCP连接重映射到所述不同的TCP连接的相同循环策略。

[0132] 强调地是,本公开的摘要被提供以遵守37.C.F.R节1.72(b),其要求将允许读者快速地确定技术公开的本性质的摘要。它具有它将不会用来解释或限制权利要求的范围或意思的理解而被提交。此外,在前述详细的描述中,能够看到,各种特征一起组合在单一示例

中以用于使本公开简化的目的。公开的此方法不解释为反映所主张的示例要求比每个权利要求中清楚地所引用的更多的特征的意图。相反,如所附权利要求反映,发明的主题物处于少于单一公开的示例的全部特征。因此所附权利要求因此结合于详细的说明中,其中每个权利要求独立作为单独示例。在所附权利要求中,术语“包括(include)”和“其中(in which)”分别用作相应的术语“包括(comprise)”和“其中(wherein)”的易懂的英语同等物。而且,术语“第一(first)”、“第二(second)”、“第三(third)”等等,仅用作标签,并且不意味着在它们的对象上强加数字要求。

[0133] 虽然在针对结构特征和/或方法论行为的语言中,主题物已经被描述,但是要理解,所附权利要求中限定的主题不必限制于上述具体特征或行为。相反,上述具体特征和行为作为实现权利要求的示例形式来公开。

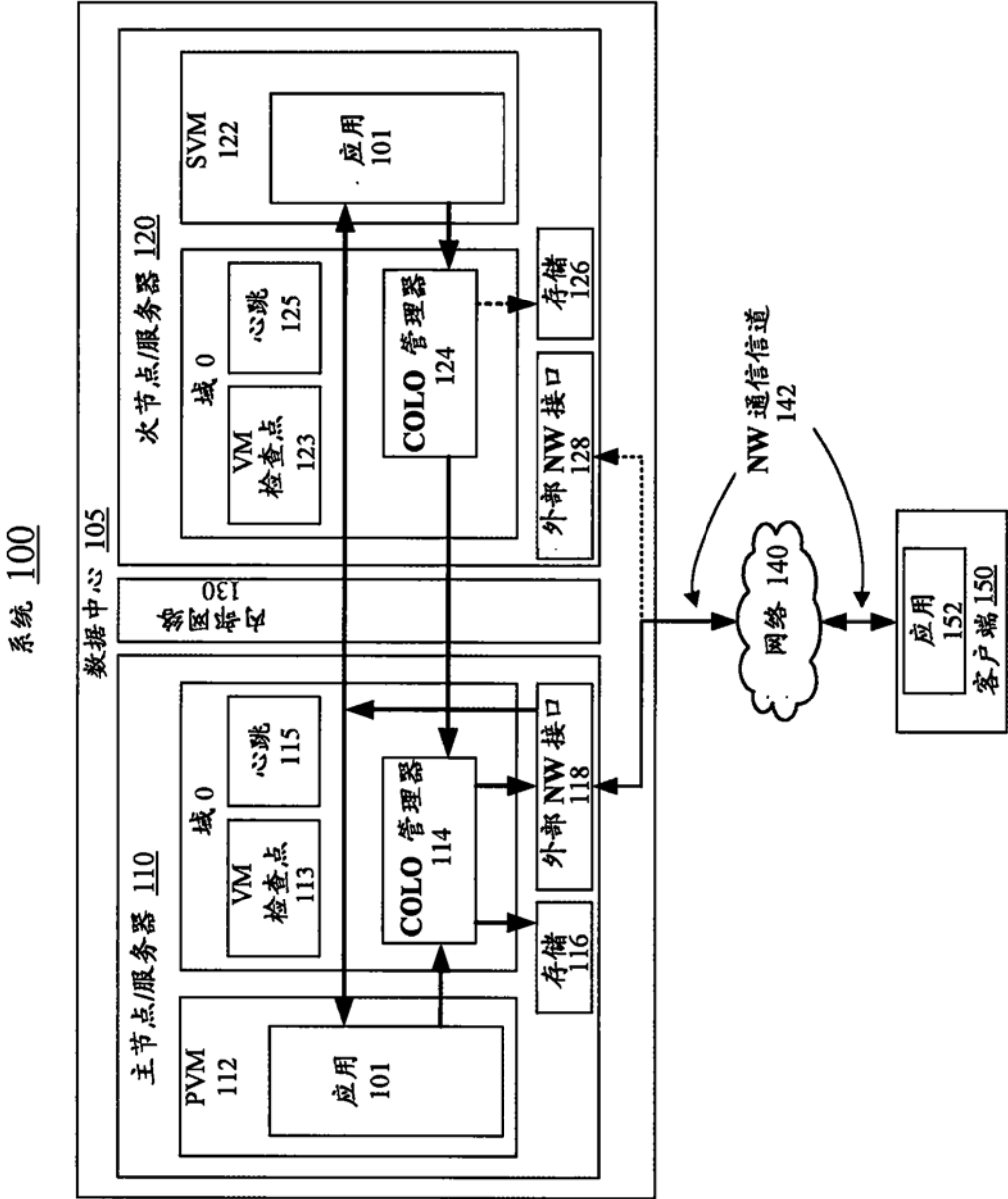


图1

编程模型 200

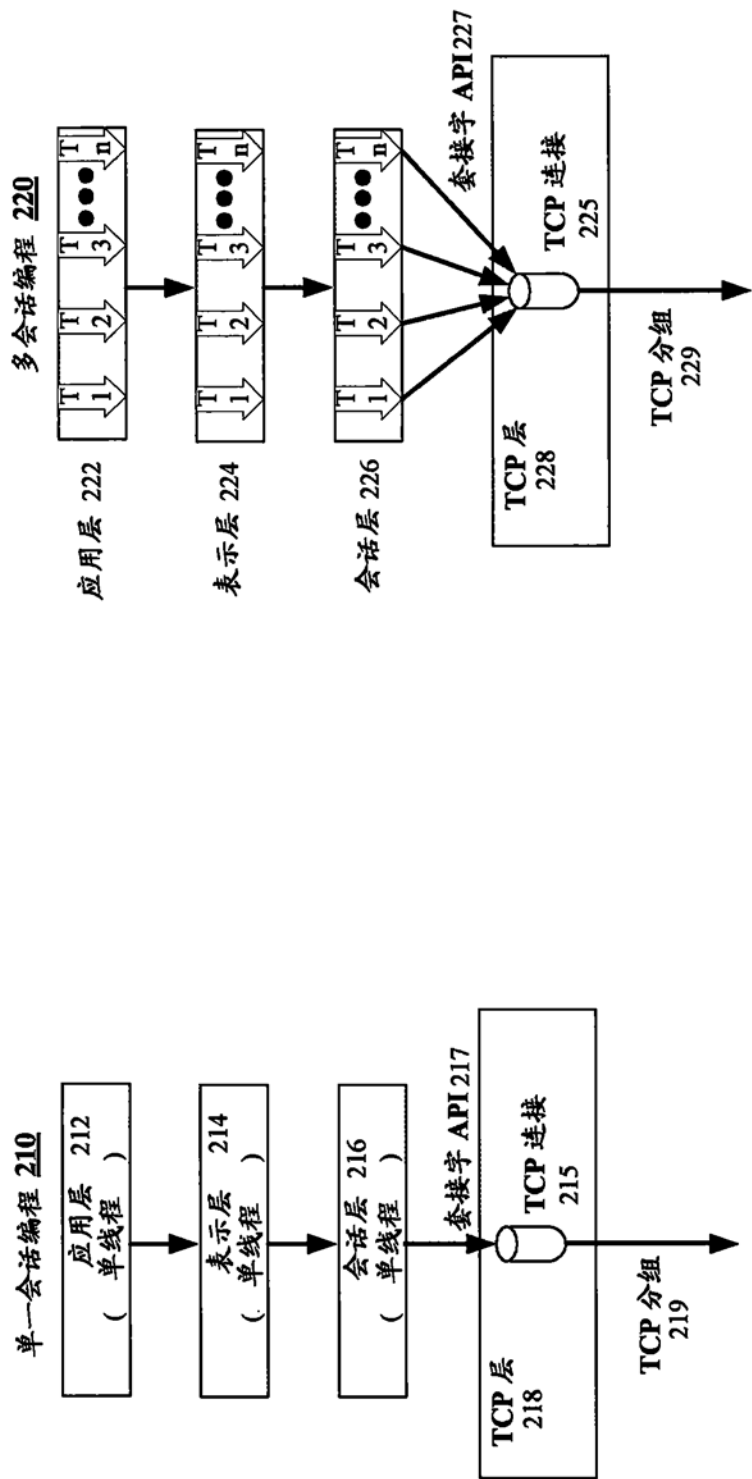


图2

系统 300

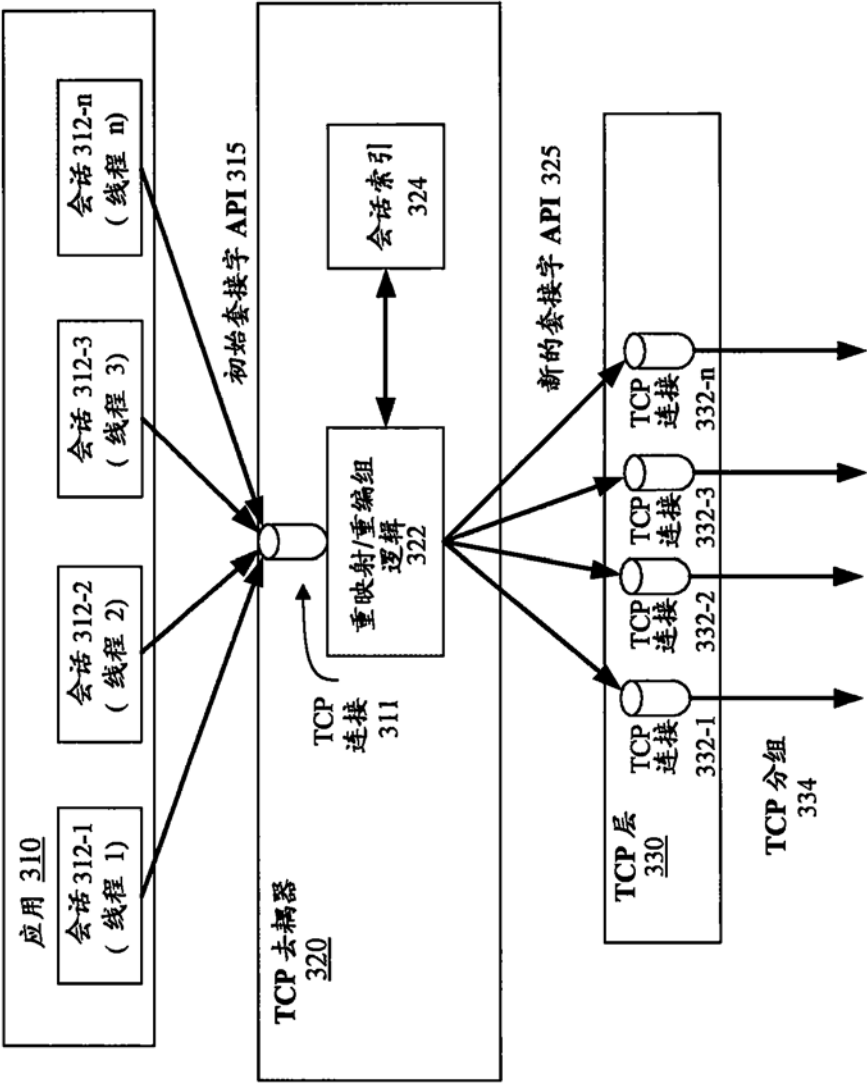


图3

系统 400

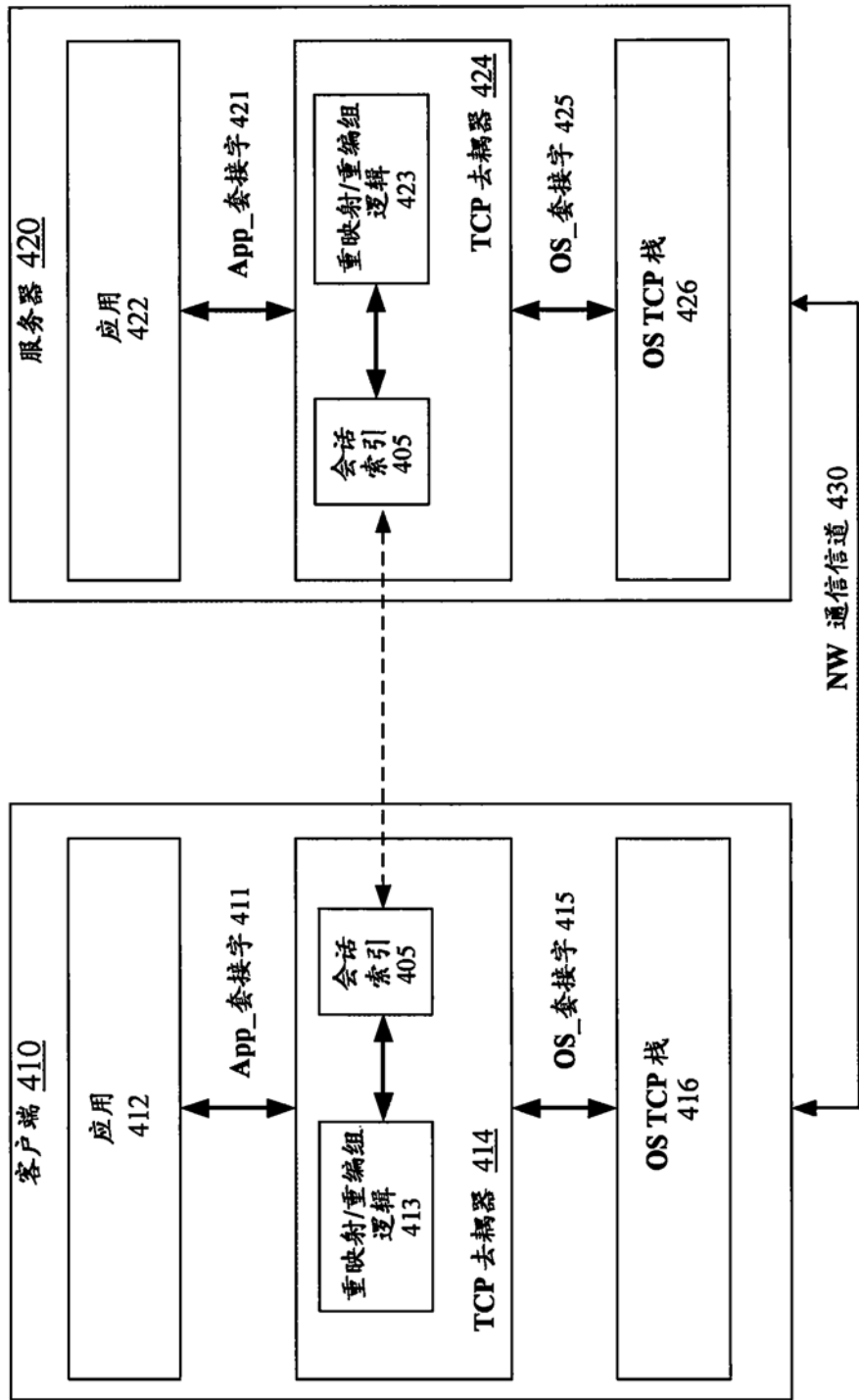


图4

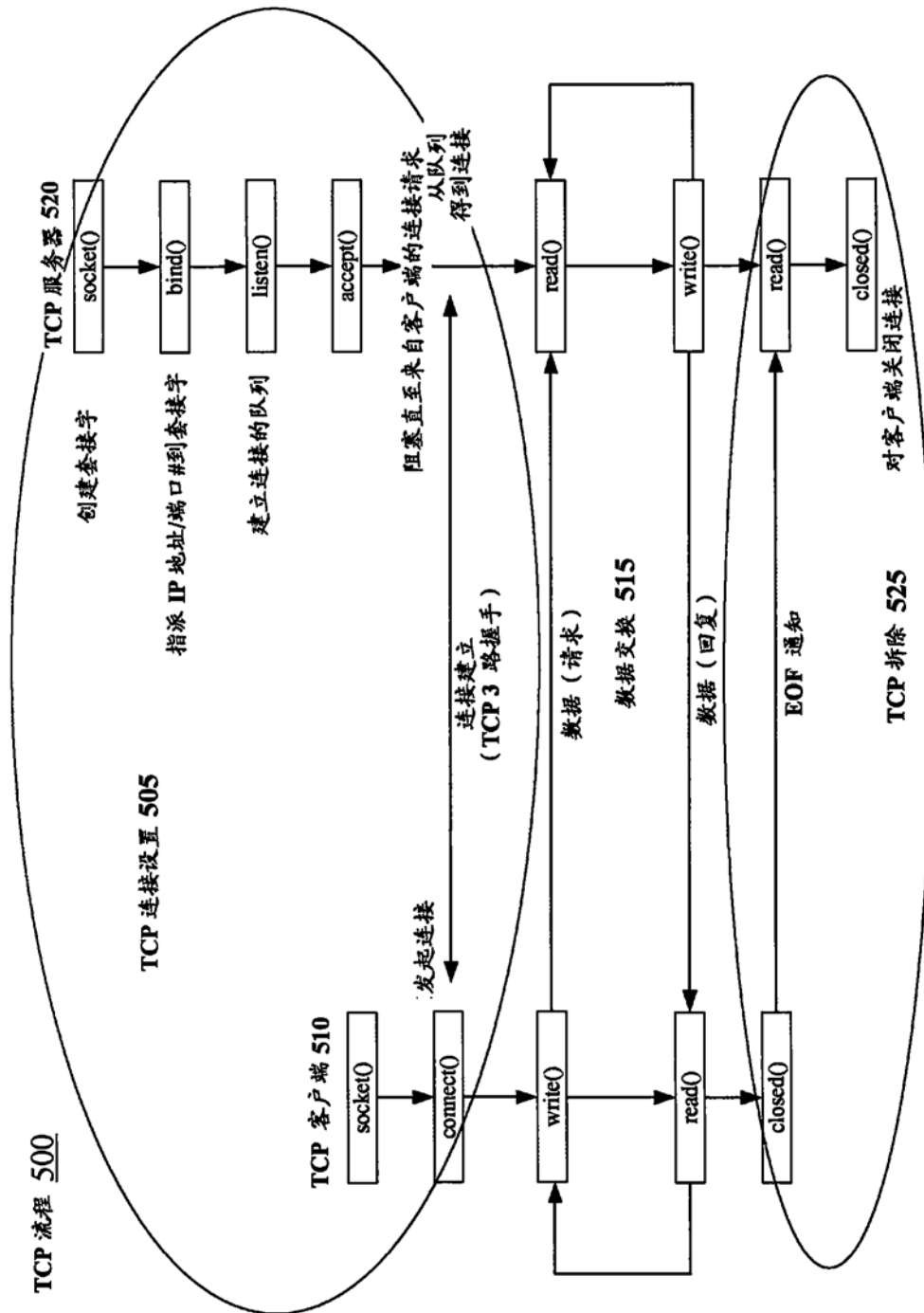


图5

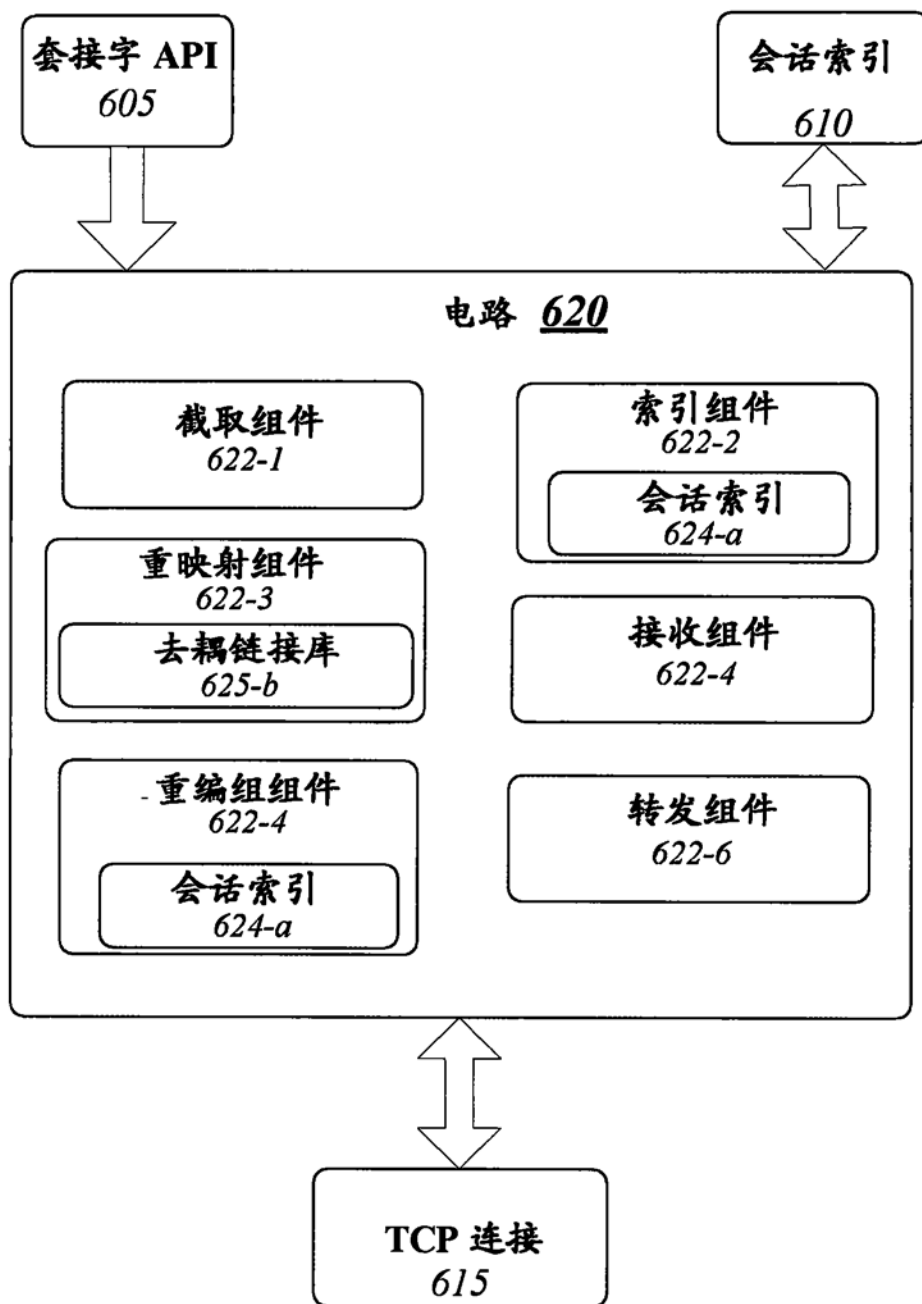
设备 600

图6

700

在布置成主管能够单独实现应用的主或次 VM 的服务器的电路，截取从该应用做出的第一和第二套接字 API 调用，第一和第二套接字 API 布置成路由相应的第一和第二会话以用于通过来自耦合到该服务器的客户端的多个 TCP 连接之中的相同 TCP 连接的期望输出，第一和第二会话由该应用响应于从客户端接收的数据而生成

702

使用对于第一和第二会话的相应第一和第二标识符信息来生成会话索引

704

重映射第一和第二会话以用于通过来自所述多个 TCP 连接之中分开的 TCP 连接的实际输出，使得会话索引能够在客户端用来将第一和第二会话与该相同 TCP 连接重组合

706

图7

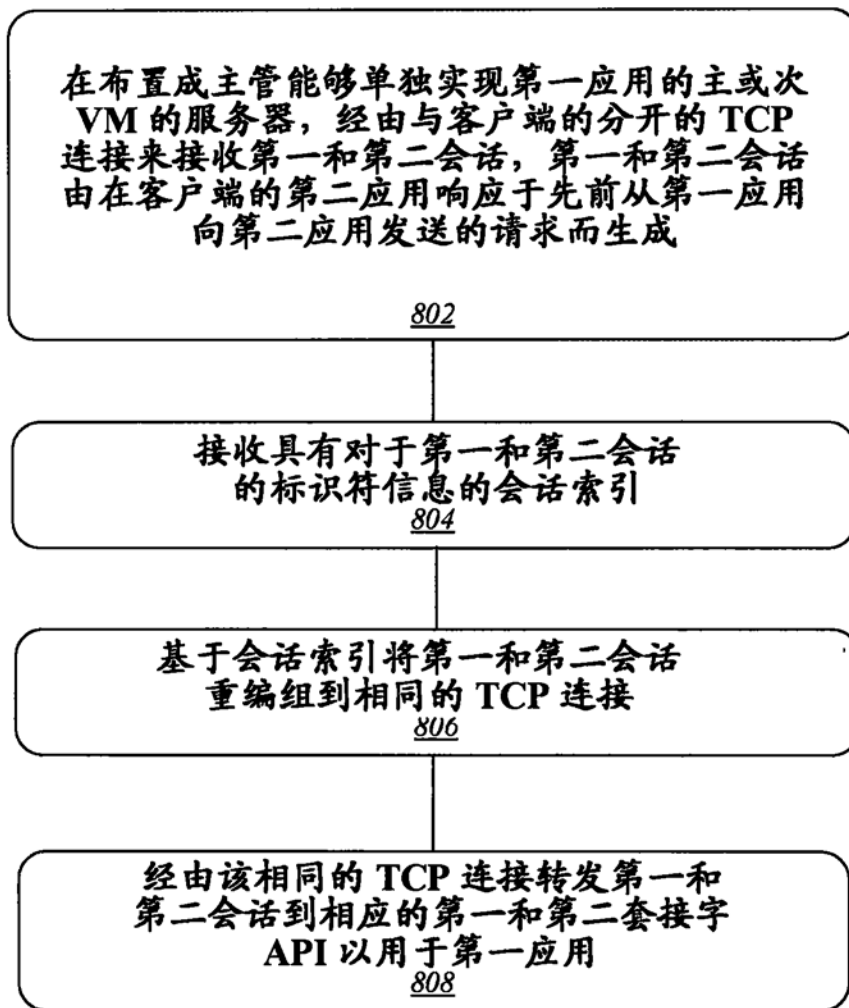
800

图8

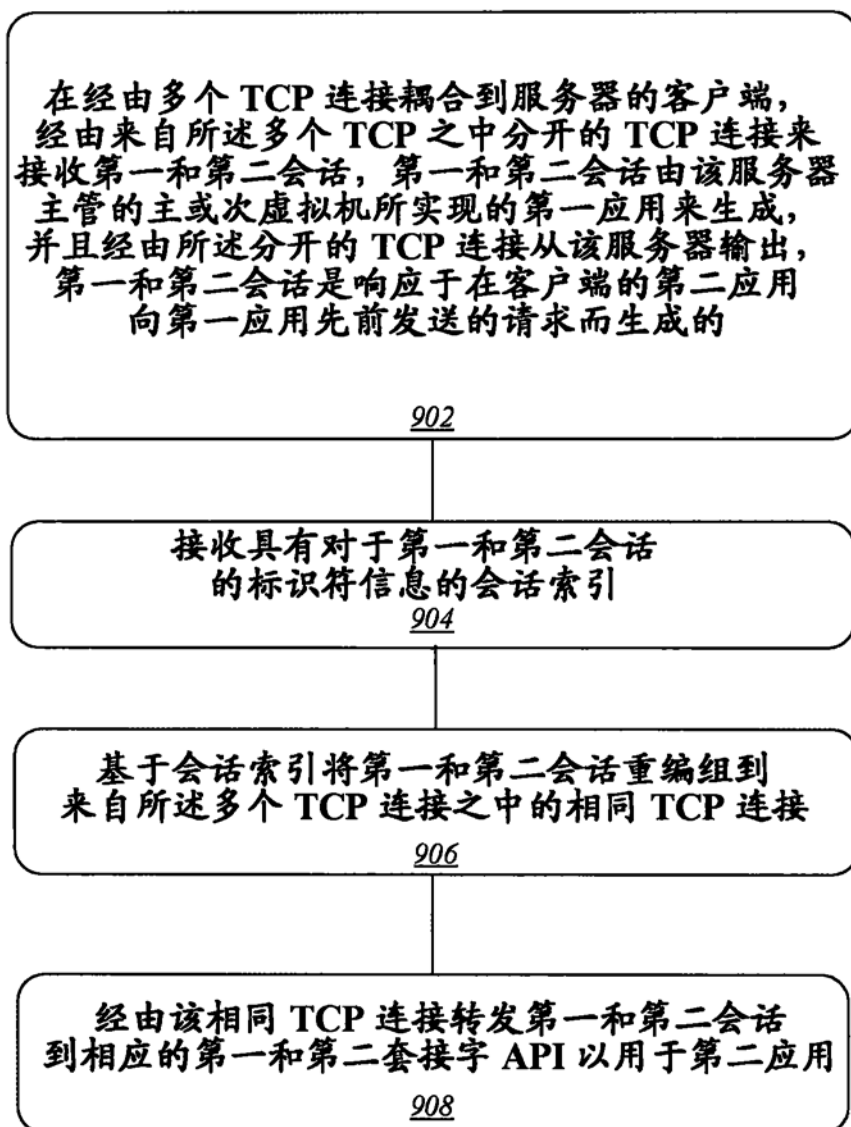
900

图9

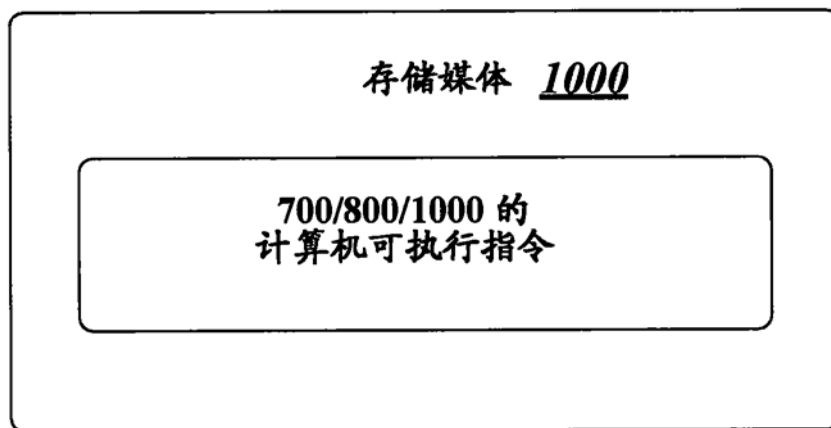


图10

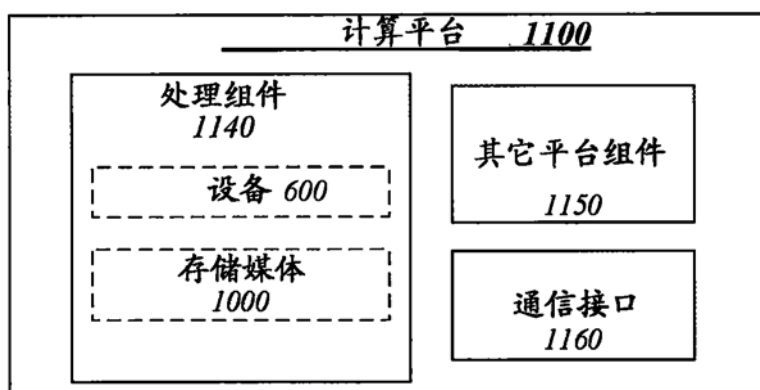


图11