



(12) 发明专利申请

(10) 申请公布号 CN 104205756 A

(43) 申请公布日 2014. 12. 10

(21) 申请号 201380014967. 9

(51) Int. Cl.

(22) 申请日 2013. 01. 10

H04L 12/917(2006. 01)

H04L 12/70(2006. 01)

(30) 优先权数据

13/353, 381 2012. 01. 19 US

(85) PCT国际申请进入国家阶段日

2014. 09. 18

(86) PCT国际申请的申请数据

PCT/US2013/020964 2013. 01. 10

(87) PCT国际申请的公布数据

W02013/109455 EN 2013. 07. 25

(71) 申请人 没有束缚软件有限公司

地址 美国威斯康星州

(72) 发明人 托德·莱勒·史密斯

马克·D·A·万古利克

(74) 专利代理机构 中原信达知识产权代理有限

责任公司 11219

代理人 周亚荣 安翔

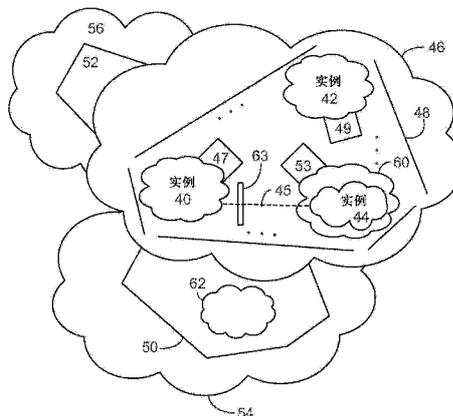
权利要求书4页 说明书29页 附图11页

(54) 发明名称

并发进程执行

(57) 摘要

除了其他以外,使得节点能够与其他节点一起参与形成并使用通信网络中的传输层特征,该传输层特征可扩展以支持在各参与节点上运行的应用之间或之中的一千万或更多同时可靠会话。



1. 一种方法,包括:

使得节点能够与其他节点一起参与形成并使用通信网络中的传输层特征,所述传输层特征能扩展以支持在相应参与节点上运行的应用之间或之中的一千万个或更多个并行的可靠的会话。

2. 根据权利要求1所述的方法,其中,所述会话基于以下中的至少一个是可靠的:可靠地传送通知、可靠地传送数据流以及不可靠地传送数据报。

3. 根据权利要求1所述的方法,其中,所述节点被使得能够在不考虑所述节点所运行于的平台的情况下进行参与。

4. 根据权利要求1所述的方法,其中,在所述通信网络的应用层提供所述传输层特征。

5. 根据权利要求1所述的方法,其中,所述参与节点及其他参与节点被自动地组织以提供可扩展传输层特征。

6. 根据权利要求1所述的方法,其中,所述会话基于以下中的至少一个是可靠的:(a) 可靠地传送通知,或者(b) 通过不可靠地传送数据报以及对不可靠数据报传送应用进程以保证流传送的可靠性,来可靠地传送数据流。

7. 一种方法,包括:

使得在通信网络的节点上的应用层中运行的用户应用能够协作以在所述通信网络上实现网络传输层特征并使用所实现的网络传输层特征。

8. 根据权利要求7所述的方法,其中,所述传输层特征包括TCP特征。

9. 根据权利要求8所述的方法,其中,所述TCP特征被用来可靠地载送通知。

10. 根据权利要求7所述的方法,其中,所述传输层特征包括UDP特征。

11. 根据权利要求10所述的方法,其中,所述UDP特征被用于节点的自动发现和节点拓扑的自动组织。

12. 一种方法,包括:

使得小的通信网络的节点能够形成并参与传输层特征,所述传输层特征提供能用于在所述节点上托管的应用之间的通信的多达数万亿个通信信道。

13. 根据权利要求12所述的方法,其中,所述小的通信网络包括少于因特网上的所有节点。

14. 根据权利要求12所述的方法,其中,所述通信信道中的每一个包括每个通过持久性服务句柄表示的两个通信端点。

15. 根据权利要求12所述的方法,其中,所述服务句柄由托管应用的节点保持,所述应用通过所述通信信道中的一个来提供或使用相关联的服务。

16. 根据权利要求12所述的方法,其中,由所述节点形成所述传输层特征包括管理与所述通信信道的端点相关联的服务句柄。

17. 根据权利要求16所述的方法,其中,所述节点协作以保持现有服务句柄的公共全局视图。

18. 根据权利要求12所述的方法,其中,所述网络传输特征包括TCP特征。

19. 根据权利要求12所述的方法,其中,所述网络传输特征包括UDP特征。

20. 一种方法,包括:

随着通信网络的配置改变,在所述网络的节点处动态地确定将被用于通过所述网络在

节点之间路由通信的表,所述动态地确定包括:

传播在相应节点处生成的近邻快照,以及
响应于所传播的近邻快照,迭代地延迟路由表的确定。

21. 根据权利要求 20 所述的方法,其中,一个节点在另一节点加入或离开其近邻时针对递增的稍后时间调度其路由表的重建。

22. 根据权利要求 21 所述的方法,其中,所述节点在又一节点加入或离开其近邻时针对又递增的稍后时间重新调度其路由表的重建。

23. 一种方法,包括:

在通信网络中的节点处,相对于由在所述节点上托管的应用或由在所述通信网络的其他节点上托管的应用提供或使用的服务而为在所述节点上托管的应用提供服务定位机构,所述服务定位机构保持服务与相应服务标识符之间的关联。

24. 根据权利要求 23 所述的方法,包括将所述关联的快照从所述节点传播至网络中的其他节点。

25. 根据权利要求 23 所述的方法,其中,所述关联被保持在服务目录中。

26. 根据权利要求 23 所述的方法,包括提供用于由应用使用所述服务目录来对感兴趣的服务进行定位的替换模式。

27. 根据权利要求 23 所述的方法,包括使用所述关联来提供任播特征。

28. 根据权利要求 23 所述的方法,包括使用所述关联来提供多播特征。

29. 根据权利要求 23 所述的方法,包括使用所述关联相对于所述通信网络的使用而提供负载平衡特征。

30. 根据权利要求 23 所述的方法,包括使用所述关联来提供接近路由特征。

31. 一种方法,包括:

在通信网络的节点中,使得能够保持通信端点以便在建立所述网络的所述节点和应用的会话时使用,在发生以下中的一个或多个时持久性地保持所述端点:(a) 会话被建立和终止,(b) 网络传输软件实例被关闭和重启,(c) 网络传输软件实例所运行于的节点被关闭和重启,(d) 整个网络传输层网格被关闭和重启,或者(e) 整个通信网络被关闭和重启。

32. 根据权利要求 31 所述的方法,包括基于所述端点的持久性来应用安全技术。

33. 根据权利要求 31 所述的方法,其中,持久地保持所述端点包括持久地保持相关联的服务句柄。

34. 根据权利要求 33 所述的方法,包括保持所述服务句柄的统计唯一的全局身份。

35. 根据权利要求 34 所述的方法,包括使得服务句柄能够被传输软件实例再用来表示会话的给定参与者。

36. 根据权利要求 31 所述的方法,包括使得所述通信网络的节点上的应用能够基于所述端点的持久性来秘密地在它们之间提供和使用服务。

37. 根据权利要求 31 所述的方法,包括将应用从所述网络的一个节点迁移至另一节点,并且基于所述端点的持久性来使得所迁移的应用能够相互提供和使用服务。

38. 根据权利要求 31 所述的方法,包括基于所述端点的持久性来分析静态程序正确性。

39. 根据权利要求 31 所述的方法,包括基于所述端点的持久性来在所述通信网络的故

障之后重建所述节点的会话。

40. 一种方法,包括:

在通信网络中,其中托管在所述网络的节点上的应用通过所述网络上的节点之间的通信来提供和使用服务,使得所述网络的节点能够在在一个节点处的故障影响来自托管在该节点上的应用的服务的可用性时进行协作以提供可靠的通知。

41. 根据权利要求 40 所述的方法,其中,所述故障包括软件重启。

42. 根据权利要求 40 所述的方法,其中,所述故障包括硬件重置。

43. 根据权利要求 40 所述的方法,其中,使得所述网络的节点能够协作以通过使用在这些节点上运行的传输层软件实例来提供可靠通知。

44. 根据权利要求 43 所述的方法,其中,所述故障包括所述实例中的一个或多个实例的操作丢失。

45. 根据权利要求 40 所述的方法,其中,所述节点包括在硬件上运行的操作系统软件。

46. 根据权利要求 45 所述的方法,其中,所述故障包括操作系统、硬件或两者的操作丢失。

47. 一种方法,包括:

在通信网络中,使得在所述网络的节点上托管的应用能够发布由所述应用提供的服务的可用性并预订由其他应用提供的服务,所述发布包括:

在以下情况下在一个模式下发布:一个服务由一个应用预订,而该应用被托管在与发布该服务的应用相同的节点上,以及

在以下情况下在一个不同的模式下发布:一个服务由一个应用预订,而该应用被托管在与托管了发布该服务的应用的节点不同的节点上。

48. 根据权利要求 47 所述的方法,包括使用所发布的服务可用性对应用的对服务的位置的请求进行响应。

49. 根据权利要求 48 所述的方法,其中,请求所述位置的应用不需要具有所述服务在本地节点上还是在远程节点上可用的先验知识。

50. 根据权利要求 48 所述的方法,包括无论所述服务在所述本地节点上还是在所述远程节点上可用,所述应用都使用单个位置中立的接口来请求所述位置。

51. 一种方法,包括:

在通信网络中,使得在网络的节点上托管的应用能够预订由网络上的应用发布的服务,所述预订包括:

在以下情况下在一个模式下预订:一个服务由一个应用发布,而该应用被托管在与预订该服务的应用相同的节点上,以及

在以下情况下在一个不同的模式下预订:一个服务由一个应用发布,而该应用被托管在与托管了预订该服务的应用的节点不同的节点上。

52. 根据权利要求 51 所述的方法,其中,在所述不同的模式下,由在与预订该服务的应用相同的节点上运行的传输层软件来本地地登记该预订。

53. 根据权利要求 52 所述的方法,其中,如果已在用于由在所述不同的节点上托管的应用发布的服务的相同节点处登记任何预订,则本地节点不需要向远程发布应用报告该新的订户。

54. 一种方法,包括:

当由网络的本地节点托管的第一客户端应用想要预订由在远程节点上运行的服务应用提供的服务时,所述本地节点上的本地传输层软件实例向所述远程节点发送预订管理消息以代表所述第一客户端应用预订所述服务,以及

所述本地传输层软件实例使得其他本地应用能够使用所述服务而不要求在所述网络上向另一节点发送任何其他预订管理消息。

55. 根据权利要求 54 所述的方法,包括:

所述本地传输层软件实例只有当没有本地客户端应用在使用所述服务时才发送另一预订管理消息。

56. 根据权利要求 54 所述的方法,包括以如下方式对针对服务的位置的请求进行响应:所述方式取决于应用所寻求的服务是否托管在与提供所述服务的应用相同的节点上。

57. 根据权利要求 56 所述的方法,其中,所述响应仅仅基于在所述一个模式下发布的服务或者在所述一个模式下以及在所述不同的模式下发布的服务。

58. 一种方法,包括:

使得能够通过被可靠地传送的通知和被不可靠地传送的数据报的组合来由在通信网络的节点上托管的应用进行通信。

59. 根据权利要求 58 所述的方法,包括使用启用的通信来可靠地传送流数据。

60. 根据权利要求 58 所述的方法,包括使用数据报来传送用户数据。

61. 一种方法,包括:

在如下的通信网络中:其中

所述网络的每个节点 (a) 能够代表在该节点上托管的应用参与与所述网络中的其他节点的通信以及 (b) 提供用于通信的物理传送和接收的 I/O 系统,以及

所述通信竞争使用 I/O 系统,

提供相对于竞争通信的所述 I/O 系统的完全无死锁的异步操作。

并发进程执行

背景技术

[0001] 本描述涉及并发进程执行。

[0002] 参考图 1, 多个进程 10(也称为应用或程序)能够由例如位于网络 16 的不同节点 14 处的相应处理器 12(例如, 计算机)运行。可以由遵守例如传输控制协议(TCP)的发送和接收网络数据分组 18 的进程来管理该并发执行。通过在每个分组中识别正在发送和接收数据分组的节点的网络上的源和目的地地址 20、22 以及已经被发送和接收进城预留用于将载送数据分组的连接的发送和接收节点处的源和目的地端口号 24 和 26 来促进 TCP 数据分组的正确传送。TCP 通过提供 16 位可寻址端口空间(0-65535)而允许在给定节点处预留有限数目的端口。

发明内容

[0003] 一般地, 在一方面, 使得节点能够与其他节点一起参与形成并使用通信网络中的传输层特征, 该传输层特征可扩展以支持在相应参与节点上运行的应用之间或之中的一千万或更多的同时可靠会话。

[0004] 实施方式可包括以下特征中的一个或多个。会话基于以下中的至少一个而是可靠的: 可靠地传送通知以及通过不可靠地传送数据报且对不可靠数据报传送应用进程以保证流传送的可靠性来可靠地传送数据流。使得节点能够在不考虑节点所运行于的平台的情况下参与。在通信网络的应用层级提供传输层特征。该参与节点及其他参与节点被自动地组织为提供可扩展传输层特征。该会话基于以下中的至少一个是可靠的 (a) 可靠地传送通知, 或者 (b) 通过不可靠地传送数据报且对不可靠数据报传送应用进程以保证流传送的可靠性来可靠地传送数据流。

[0005] 一般地, 在一方面, 使得在通信网络的节点上的应用层中运行的用户应用能够协作以在通信网络上实现网络传输层特征并使用所实现的网络传输层特征。

[0006] 实施方式可包括以下特征中的一个或多个。该传输层特征包括 TCP 特征。该 TCP 特征用来可靠地载送通知。传输层特征包括 UDP 特征。UDP 特征被用于节点的自动发现和节点拓扑的自动组织。

[0007] 一般地, 在一方面, 使得小的通信网络的节点能够形成并参与传输层特征, 其提供可用于在节点上托管的应用之间的通信的多达数万亿个通信信道。

[0008] 实施方式可包括以下特征中的一个或多个。该小的通信网络包括少于因特网上的所有节点。每个通信信道包括每个用持久性服务句柄来表示的两个通信端点。服务句柄由托管应用的节点保持, 该应用通过通信信道中的一个来提供或使用相关联的服务。由节点来形成传输层特征包括管理与通信信道的端点相关联的服务句柄。节点协作以保持现有服务句柄的公共全局视图。网络传输特征包括 TCP 特征。网络传输特征包括 UDP 特征。

[0009] 一般地, 在一方面, 随着通信网络的配置改变, 在网络的节点处动态地确定将被用于通过网络在节点之间路由通信的表。动态确定包括传播在相应节点处生成的近邻快照, 并且响应于传播的近邻快照而迭代地延迟路由表的确定。

[0010] 实施方式可包括以下特征中的一个或多个。节点在另一节点加入或离开其近邻时对于增量的随后时间调度其路由表的重建。节点在又另一节点加入或离开其近邻时对于又一增量的随后时间重新调度其路由表的重建。

[0011] 一般地,在一方面,在通信网络中的节点处,相对于由在节点上托管的应用或由在通信网络的其他节点上托管的应用提供或使用的服务而为在节点上托管的应用提供服务定位机构。该服务定位机构保持服务与相应服务标识符之间的关联。

[0012] 实施方式可包括以下特征中的一个或多个。关联的快照被从节点传播至网络中的其他节点。该关联被保持在服务目录中。提供了用于应用使用服务目录来对感兴趣服务进行定位的替换模式。该关联被用来提供任播特征。该关联被用来提供多播特征。该关联被用来提供相对于通信网络的使用的负载均衡特征。该关联被用来提供接近路由特征。

[0013] 一般地,在一方面,在通信网络的节点中,使得能够进行通信端点的保持以便在建立网络的节点和应用的会话时使用。在发生以下中的一个或多个时持久性地保持端点:(a) 建立和终止会话,(b) 网络传输软件实例被关闭和重启,(c) 网络传输软件实例正运行于的节点被关闭和重启,(d) 整个网络传输层网络被关闭和重启,或者(e) 整个通信网络被关闭和重启。

[0014] 实施方式可包括以下特征中的一个或多个。基于端点的持久性来应用安全技术。持久地保持端点包括持久地保持相关联的服务句柄。保持服务句柄的统计唯一全局身份。使得服务句柄能够被传输软件实例再使用以表示会话的给定参与者。使得通信网络的节点上的应用能够基于端点的持久性而在其之间秘密地提供并使用服务。使得应用能够从网络的一个节点迁移至另一节点,并且使得迁移的应用能够基于端点的持久性而相互提供和使用服务。基于端点的持久性来分析静态程序正确性。在通信网络的故障之后基于端点的持久性来重新建立节点的会话。

[0015] 一般地,在一方面,在其中在网络的节点上托管的应用通过网络上的节点之间的通信来提供和使用服务的通信网络中,使得网络的节点能够在节点处的故障影响来自在节点上托管的应用的服务可用性时进行协作以提供可靠的通知。

[0016] 实施方式可包括以下特征中的一个或多个。故障包括软件重启。故障包括硬件重置。使得网络的节点能够通过使用在节点上运行的传输层软件实例而协作以提供可靠通知。故障包括实例中的一个或多个的操作的丢失。节点包括在硬件上运行的操作系统软件。故障包括操作系统、硬件或两者的操作的丢失。

[0017] 一般地,在一方面,在通信网络中,使得在网络的节点上托管的应用能够发布由应用提供的服务的可用性并预订由其他应用提供的服务。该发布包括当由在与发布服务的应用相同的节点上托管的应用预订服务时在一个模式下发布,并且当由在与托管发布服务的应用的节点不同的节点上托管的应用预订服务时在不同模式下发布。

[0018] 实施方式可包括以下特征中的一个或多个。发布的服务可用性被用来对应用对服务的定位的请求进行响应。请求该定位的应用不需要具有服务在本地节点上还是远程节点上可用的先验知识。无论服务是在本地节点上还是在远程节点上可用,应用将单个位置中立接口用于请求该定位。

[0019] 一般地,在一方面,在通信网络中,使得在网络的节点上托管的应用能够预订由网络上的应用发布的服务。该预订包括:当由在与预订服务的应用相同的节点上托管的应用

来发布服务时在一个模式下预订,并且当由在与托管预订该服务的应用的节点不同的节点上托管的应用来发布服务时在不同的模式下预订。

[0020] 实施方式可包括以下特征中的一个或多个。在不同模式下,由在与预订服务的应用相同的节点上运行的传输层软件来本地地登记该预订。如果在相同节点处已对于由在不同节点上托管的应用发布的服务登记了任何预订,则本地节点不需要向远程发布应用报告新的订户。

[0021] 一般地,在一方面,当由网络的本地节点托管的第一客户端应用想要预订由在远程节点上提供的服务应用提供的服务时,本地节点上的本地传输层软件实例向远程节点发送预订管理消息以代表第一客户端应用预订服务。本地传输层软件实例使得其他本地应用能够使用该服务而不要求在网络上向所述另一节点发送任何其他预订管理消息。

[0022] 实施方式可包括以下特征中的一个或多个。本地传输层软件实例只有当没有本地客户端应用在使用该服务时才发送另一预订管理消息。以取决于应用所寻求的服务是否在与提供服务的应用相同的节点上托管的方式对用于服务定位的请求进行响应。该响应可以基于仅在一个模式下发布的服务或者在所述一个模式下和在所述不同模式下发布的服务。

[0023] 一般地,在一方面,由可靠地传送的通知与不可靠地传送的数据报的组合来启用在通信网络的节点上托管的应用的通信。

[0024] 实施方式可包括以下特征中的一个或多个。所启用的通信被用来可靠地传送流送数据。该数据报被用来传送用户数据。

[0025] 一般地,在一方面,在通信网络中,网络的每个节点 (a) 能够代表在节点上托管的应用参与与网络中的其他节点的通信,和 (b) 提供用于通信的物理传送和接收的 I/O 系统。该通信竞争对 I/O 系统的使用。相对于竞争通信而提供 I/O 系统的完全无死锁异步操作。可以将这些及其他方面、特征以及实施方式表达为方法、系统、设备、程序产品、经营商业的方法、用于执行功能的手段和步骤、部件以及其他方式。

[0026] 根据具体实施方式和附图以及根据权利要求,本发明的其他特征、目的和优点将变得显而易见。

具体实施方式

[0027] 图 1 至 12 是框图。

[0028] 虽然由 TCP 提供的 16 位可寻址端口空间对于许多用户应用及其之间的网络通信而言是足够的,但其对于超级计算集群和网格而言常常太小。例如,有限的端口空间可以使得不可能实现将执行大规模并发算法的数千个参与进程的互连集团之间的直接 TCP 分组通信。

[0029] 虽然 TCP 对其连接(即,在其连接空间上)仅施加了<源 IP、源端口、目的地 IP、目的地端口>的唯一性约束,但有时不能在伯克利软件分发中心(BSD)衍生套接字应用编程接口(API)的规范下完全分配连接空间。具体地,API 要求客户端进程在发起到服务器的连接之前分配唯一的本地 TCP 端口,并且客户端的节点被端口空间限于 2^{16} (65536) 个输出 TCP 连接。类似地,托管服务器进程的节点限于来自客户端的特定节点的 2^{16} 个输入连接。

[0030] 网际协议版本 6 (IPv6) 上的 TCP 通过大大地扩展网络源和目的地地址空间(而不是扩展端口空间)来处理这些规模限制,但是 IPv6 的典型实现的各方面约束可用于网格计

算应用中的并行程度,具体地在其中分布式软件有效地利用在特定节点处可用的处理器核心的系统中。

[0031] 作为示例,给定跨 120 个节点分布的网络应用,其中的每一个托管用于其 24 个处理器核心中的每一个的一个进程,使得每个进程希望统一地使用 TCP 来与每个其他参与进程通信,每个节点将需要使 69,096 个端口专用于在该节点上运行的网络应用进程的本地使用。端口的此数目比 TCP 端口空间可以支持的多了数千个。

[0032] 在这里,我们讨论新的平台中立网络传输层,其提供显著地缩放超过 TCP 16 位端口空间限制的连接空间机会。此新传输层还提供深的高效网络缓冲和鲁棒的服务架构,其支持任播和多播寻址、负载平衡、身份持久性以及事件的可靠通知。能够在不施加特殊硬件要求的情况下使用可用处理器、存储器及其他硬件来管理跨数千个应用和数百个节点分布的数千万个活动通信端点。可以为网络计算应用提供高水平的并行,具体是在分布式软件正在很好地利用在特定节点处可用的处理器核心的情况下。

[0033] 如图 2 中所示,在某些示例中,可以将此平台中立、大型连接空间网络传输层 30 实现为我们称为网络传输软件 32 的内容,其实例在网络的相应节点处运行。我们在非常宽泛的意义上使用短语网络传输软件来包括例如在网络节点上运行且提供在这里所述的新颖特征中的任何一个或多个或任何组合的软件的实例。网络传输软件的某些实施方式可以采取可从美国威斯康星州麦迪逊市的 MioSoft 公司获得的 Mioplexer™ 软件的实例的形式。在本描述中对 Mioplexer 的任何引用意图是对包括在本文档中所述种类的任何种类的此类网络传输软件的宽泛引用。

[0034] 网络传输软件作为高级网络传输层 29 在 TCP 34 和用户数据报协议 (UDP) 36 上操作。在某些实施方式中,网络传输软件支持网际协议版本 4 (IPv4) 和 (IPv6)。

[0035] 如图 3 中所示,网络传输软件实例 40 使用广播寻址来自动发现在相同网络 46 上操作的其他实例 42、44,以形成用于该网络的节点的大型连接空间网络传输网络 48。自动发现进程共享网络传输软件标识符,其出于自动发现的目的指定 TCP 侦听端口。网络传输软件 32 包括标识符分辨进程 33,其使用域名系统 (DNS) 35 来在将一致的十进制和十六进制数字标识符分别地视为 IPv4 和 IPv6 地址的同时分辨非数值标识符。

[0036] 如果广播寻址不可用或不够,则可用预配置目标的单播寻址来补充自动发现进程。此机制还可以用来将与不同网络 54、56 相关联的大型连接空间网络传输网络 50、52 结合在一起。在某些实施方式中,可以在合并了网络接口卡 (NIC) 的商业可获得的商用硬件上实现且在支持 Java 平台的任何操作系统上运行网络传输软件。

[0037] 如图 3 中所示,由网络传输软件形成的互连网络 48 包括跨网络、诸如 TCP/IP 网络的每个网络 46、54、56 中的许多网络节点 60、62 分布的网络传输软件的许多实例 40、42、44 的集合。在典型配置中,每个参与节点仅仅托管网络传输软件的单个实例。(有时,我们将网络中的托管网络传输软件的实例的节点简单地称为节点。有时我们可互换地使用术语节点和网络传输软件。请注意,虽然节点托管网络传输软件,但软件可在节点正在运行时关闭。并且,当节点关闭时,软件也关闭。)

[0038] 此配置类似于传统网络传输层的典型配置:节点处的操作系统实例提供将被所有用户应用共享的 TCP 堆栈的单个实施方式。在我们在这里描述的内容的某些实施方式中,网络中的单个节点可以托管网络传输软件的多个副本,其可以被用于本地地测试基础软件

和用户应用。

[0039] 在节点中运行的网络传输软件实例使用基于 UDP 的自动发现进程来将其本身组织成互连网络。在合理地稳定的网络环境中,在网络的各个节点上运行的用户应用 11、13(图 1)能够自动地利用预先建立网络来减少发起分布式算法的并发平行处理否则将需要的启动等待时间。

[0040] 网格内的相邻节点被使用 TCP 可靠地连接。网络传输软件实例将默认为 13697 的相同端口号用于关于输入和输出 UDP 自动发现相关消息的新 TCP 连接。自动发现进程贯穿网络传输软件的整个寿命保持活动,并且因此使由临时网络中断引起的丢失 TCP 连接的快速恢复自动化。假设网络链路并未由于拓扑重组而消失,则自动发现机制自动地修理网格中的长期违反。

[0041] 在不同节点上托管的网络传输软件实例 40、42、44(为了简单起见,我们有时将把网络传输软件的实例简单地称为网络传输软件)能够使用全双工 TCP 连接 45 而相互连接。一旦已经在两个网络传输软件实例之间建立了 TCP 连接(我们有时将传输软件的实例之间的这些连接称为网络传输软件连接),在客户端服务器模型示例中,客户端节点和服务器节点协商以商定例如 Mioplexer 协议版本。如果不能达成一致,则客户端必须从服务器断开连接。如果客户端在此事件中未能断开连接,则服务器必须在发生第一违反协议时将客户端断开连接。

[0042] 参考图 4,网格 59 支持希望在分离地址空间或网络节点 70、72 之间交换数据 68、提供或使用非本地服务 74、76 或协作以执行并行算法或者那些活动中的两个或更多的任何组合等的用户应用 64、66。

[0043] 希望将网格用于任何这些活动的用户应用首先建立到网格内的特定网络传输软件实例 78、80 的 TCP 连接 82、84。虽然用户应用可选择参与网络传输软件自动发现进程以对适当的目标实例进行定位,但用户应用常常将具有特定网络传输软件实例及其托管节点的身份和位置的先验知识。目标实例常常将在与用户应用相同的节点上运行。

[0044] 我们将网络传输软件实例与用户应用之间的 TCP 连接称为应用连接。当用户应用相对于由网络传输软件实例提供的服务充当客户端时,可将应用连接称为客户端连接。关于网络传输软件实例和应用所扮演的角色,任何网络传输软件实例都能够充当用于寻找服务的客户端应用的服务器。并且网络传输软件实例可以在期待设立到另一节点的输出连接时充当客户端。如果用户应用需要来自另一用户应用或来自节点或服务的服务以向另一用户应用提供服务,则其可以是客户端。在所有这些情况下,客户端需要服务且服务器提供该服务。

[0045] 在实例与应用之间存在两个层级的逻辑连接。较低层级是用户应用与网络传输软件实例之间的 TCP 连接。较高级别是两个用户应用之间的服务句柄(例如,信道)连接。逻辑连接通常建立客户端服务器关系的方向性。用户应用和网络传输软件实例两者都能够根据上下文而扮演角色(客户端或服务器)。

[0046] 在某些实施方式中,出于一切目的将应用连接视为全双工的。在建立应用连接之后,用户应用和网络传输软件进行协商以商定例如 Mioplexer 协议版本。如果不能达成一致,则用户应用将从网络传输软件断开连接。如果客户端在此事件中未能断开连接,则网络传输软件将在发生第一协议违反时将用户应用断开连接。如果另一方面协议版本协商导致

可行的应用连接,则例如作为客户端进行操作的用户应用沿着此连接发送控制消息、查询以及数据报,并且能够沿着同一连接从网络传输软件或其他用户应用接收控制消息确认、查询响应、数据报以及事件通知。客户端数据报可以从一个用户应用向另一个载送用户数据。

[0047] 如图 5 中所示,网络 89 还使得用户应用 90、92 能够通过开放所谓的服务句柄 94、96 且借助于该服务句柄来交换用户数据 98 而直接地相互通信。服务句柄是不透明纪念物,其通用且唯一地表示可以客户端数据报 100 的形式来发送或接收用户数据的持久性通信端点 93、95。客户端数据报交换协议是无连接的。服务句柄只需要开放以使得客户端能够发送或接收客户端数据报。任何两个开放服务句柄 94、96 限定信道 102,客户端数据报 100 可跨该信道 102 流动。

[0048] 虽然用户应用可具有例如在另一节点处促进特定服务的特定服务句柄的显式先验知识,但用户应用还可以使用以一般方式对所需服务进行命名的服务标识符 106 来查询其网络传输软件 104(例如,由与用户应用相同的节点托管的实例)。提供服务 91 的用户应用 90 可让其网络传输软件 108 将服务标识符 110 绑定 112 到促进该服务的每个服务句柄 94;此过程称为服务广告。

[0049] 一旦服务句柄被绑定到服务标识符,则其能够被用户应用发现。服务标识符不需要是唯一的。在某些实施方式中,许多服务句柄 114、116 广告相同的服务标识符。如果存在与特定服务标识符匹配的多个服务句柄,则网络传输软件可以应用由来自用户应用的查询 106 指定的附加过滤器 118 并用满足该查询的服务句柄进行答复。

[0050] 此布置允许网络传输软件提供按需负载平衡、接近路由、任播路由或其他高级路由能力或其中的两个或更多的任何组合,并且在满足用户应用的查询的过程中提供其他管理功能。在某些实施方式中,可以实现规则以确保服务客户端不发现不适当的服务提供者。例如,当且仅当两个服务句柄以相同的方式提供相同服务时,才允许这两个服务句柄绑定同一服务标识符。负责网络传输层网格的管理的组织可能希望建立命名权限和过程以防止网络传输软件网格中的全局服务标识符命名空间中的意外冲突。

[0051] 如图 6 中所示,用户应用 120 可向任何其他服务句柄 124 的事件流 126 预订 121 其开放服务句柄 122 中的任何一个,甚至是从未开放的一个。我们将前一服务句柄命名为订户 122 并将后者命名为发布者 124。当在发布者的生命周期中发生感兴趣事件 130 时,诸如其开放或关闭,其向所有订户发布此事件的通知 132。来自给定发布者的事件通知被按照发生顺序可靠地传送 134 至其所有订户。保证事件通知是唯一的;网络层软件实例仅发送事件的单个通知,并且没有订户曾接收到重复通知,即使在存在混乱或不稳定网络的情况下。

[0052] 基于最佳努力来传送应用(例如,客户端)数据报 136,并且将网格设计成甚至在系统性重负载下也很好地表现。然而,在某些实施方式中,网格的网络层软件实例可根据其自己的判断丢弃 138 客户端数据报。直接地使用该客户端数据报传输的用户应用必须接受客户端数据报的任意丢失的可能性,但是在实践中,软件实例仅丢弃与缓慢流动信道相关联的客户端数据报,并且只有当系统被极其繁重的业务全局地重压时。

[0053] 由于通过网格的路由可由于节点故障、网络中断和网格中的新网络层软件实例的自动发现而改变,所以客户端数据报可按照与其被从源服务句柄发送的顺序不同的顺序到达其目的地服务句柄。可以将网格配置成缓存客户端数据报并调谐至与环境的当前使用情

况匹配。该缓存可以包括适合于大多数业务模式的合理默认。

[0054] 虽然不可靠用户数据报 139 和可靠事件通知 134 的此组合对许多用户应用而言是足够有用的,但传输层还可以提供用户数据的可靠按顺序传送。网络层软件的用户能够在由网络层软件提供的平台中立网络传输层上设计传输层。在某些实施方式中,可以与网络传输层一起捆绑和部署较高级传输层 29(图 2)。该较高级传输层可包含生产质量客户端库 31,其实现利用大范围的网络传输软件能力的强大且鲁棒的面向连接可靠流送协议。

[0055] 返回自动发现,为了降低用户配置成本且使可靠性最大化,网络传输软件及其节点可使用连续自动发现进程来识别对端节点并建立且保持可行网格。自动发现进程涉及到触发 TCP 连接尝试的 UDP 消息的周期性互换。此进程还可帮助确保丢失的 TCP 连接在网络条件允许时尽可能快速地自动恢复。

[0056] 一旦节点上的网络传输软件正在运行,则其启动以具有例如 10,000ms(10s)的默认值的用户定义时段周期性地期满的定时器。此定时器定义问候心跳,并且控制由网络传输软件的该实例在 UDP 上广播自动发现消息的速率。给定软件实例处的初始心跳的时序被随机化成在由该时段建立的跨度内发生而引入在网格内协作的节点之间的心律不齐。该心律不齐降低了脉冲 UDP 广播的可能性和影响,其否则将由于同时地在许多节点上启动网络传输软件而发生。这种策略减少了网络硬件丢弃的 UDP 分组的数目(UDP 分组通常在其他分组之前被丢弃)。

[0057] 每个心跳一次,给定节点的网络传输软件通过 UDP 向每个目标网络广播请求问候消息。作为默认,节点的网络传输软件以节点参与其中的所有网络为目标。请求问候消息包括在其网格上唯一地识别发送器节点的网络传输软件标识符(47、49、53,图 3)。此标识符是<节点名、服务器端口号>,其中,节点名是尺寸前缀 UTF-8 串,其表示例如网络传输软件主机节点的 DNS 名、IPv4 地址或 IPv6 地址。

[0058] 当在节点上托管的网络传输软件接收到请求问候消息时,其在必要时将包含在消息中的网络传输软件标识符分辨成 IP 地址。如果到发送器的 TCP 连接已不存在,则接收者使用问候消息通过 UDP 上的单播进行应答。问候消息包括发送器的网络传输软件标识符。接收者然后发起到所指示的<IP 地址、服务器端口号>的 TCP 连接。如果到发送器的 TCP 连接已存在,则在没有进一步动作的情况下丢弃请求问候消息。

[0059] 在某些实施方式中,网格中的每个托管网络传输软件的两个节点可竞争以相互建立 TCP 连接。可几乎同时地启动在许多节点上托管的网络传输软件,并且期望的是在任何两个节点之间仅保持一个 TCP 连接以便实现网络资源的最高效使用。由于网络传输软件标识符在网格内是唯一的,所以其能够用来定义 TCP 连接的总顺序。在某些实施方式中,当在网格的两个节点之间建立 TCP 连接时,具有较低对照网络传输软件标识符的网络传输软件检查预先存在的 TCP 连接的存在。如果其发现此类连接,则其废除其发起的 TCP 连接并保留另一个。控制内部 TCP 连接管理数据结构的同步机制确保这两个连接中的一个必须严格地在另一个之前完成,因此该算法保证冗余连接是暂时的。被防火墙 63(图 3)分离且被网络地址转换(NAT)隔离的每个托管网络传输软件的网格中的两个节点因此能够可靠地相互通信;只要节点中的一个可从另一个到达,则可在其之间建立全双工连接。

[0060] 想要利用网络传输软件自动发现进程的用户应用可在适当的 UDP 端口上侦听请求问候消息。用户应用不用问候消息对请求问候消息进行响应,以免由请求问候消息的发

起者对另一网络传输软件实例混淆。在类似于网络的部署情形中,网络传输软件将与相应用户应用居于同处。因此,用户应用应通常在采取侦听请求问候消息之前尝试建立到同一节点的标准网络传输软件端口的 TCP 连接以便对可行网络传输软件实例进行定位。

[0061] 关于协议版本协商,在从任意用户(例如,客户端)应用向节点(例如,服务器节点)建立应用连接之后,协商网络传输软件协议版本以确保相互兼容性。每个一致的客户端应用认可一个列表的可接受服务器协议版本。作为服务器的每个网络传输软件实例认可一个列表的可接受客户端协议版本。在某些实施方式中,网络传输软件充当客户端,例如当建立到网格中的另一节点的输出 TCP 连接时,并且充当服务器,例如当接受 TCP 连接时。此方案确保网络传输软件实施方式之间的向后和向前兼容性的滑动窗口。

[0062] 协议版本协商必须在可发出任何请求、给出响应或交换用户数据之前成功地完成。为了减少用于用户(例如,客户端)应用和网格开发者两者的实现负担,可在协议协商之前或期间交换活动性消息。

[0063] 当客户端应用已成功地建立应用连接时,客户端发射客户端版本消息,其封装唯一地识别客户端的优选网络传输软件协议版本的尺寸前缀 UTF-8 串。网络传输软件协议串的内容可以专有地由单个控制源(诸如 MioSoft 公司)规定。在某些实施方式中,实际网络传输软件协议串按照惯例可以符合格式“MUX YYYY.MM.DD”,其中,YYYY 是四位格里历年,MM 是基于 1 的两位月序数,并且 DD 是基于 1 的两位天序数。该数据可以对应于网络传输软件协议的设计日期。

[0064] 当服务器接收到此客户端版本消息时,其针对其可接受客户端协议版本的列表中的成员资格而检查嵌入协议版本,以查看其是否能够保证协议版本兼容。服务器用包含其自己的优选网络传输软件协议版本和协议版本兼容声明的服务器版本消息进行响应。此声明是作为成员资格测试的结果的布尔值。真的值指示服务器保证与客户端的协议兼容;假的值放弃任何此类保证。

[0065] 当客户端接收到此服务器版本消息时,其检查协议版本兼容声明。如果该声明是真,则协议版本协商已成功地完成。如果声明是假,则客户端针对其可接受的服务器协议版本的列表中的成员资格而检查嵌入协议版本。如果成员资格测试是肯定的,则协议版本协商已成功地完成。

[0066] 如果 1) 兼容声明是假且 2) 客户端侧成员资格测试是否定的,则协议版本协商已失败:客户端和服务端不具有共有的协议版本,并且因此是不兼容的。不可发送请求,不可接收响应,并且不可交换用户数据。当客户端已检测到此状况时,其在不发射任何附加消息的情况下从服务器断开连接。

[0067] 如果协议版本协商成功地完成,则客户端可发射服务请求和用户数据,期望服务器理解输入消息并将适当地进行反应。

[0068] 如图 7 中所示,关于路由,节点 142 处的网络传输软件 140 负责传送沿着其输入 TCP 网络连接 148 到达的任何用户(例如,客户端)数据报 144(其中包装了用户数据)。客户端数据报消息(我们常常将其简单地称为客户端数据报)在特定源服务句柄 150 处发起并跨网格而行进至其目的地服务句柄 152。当客户端数据报消息到达负责目的地服务句柄的网络传输软件时,网络传输软件检查目的地服务句柄的状态。如果服务句柄是开放的,则网络传输软件向在适当 TCP 客户端连接 158 的另一端 156 处的用户应用 154 传送客户端数

据报消息。如果服务句柄不是开放的,则网络传输软件丢弃 160 客户端数据报。

[0069] 如果用户应用 154 向与托管两个用户应用的同一节点 142 直接相关联的另一用户应用 166 发送客户端数据报消息 164,则网络传输软件 140 简单地在其自己的内部数据结构内进行导航以传送消息。在某些实施方式中,用户应用 175、154 相互远离且常驻于不同的网络节点 142 上。在这种情况下,网络传输软件 178 跨其活动网络间传输软件间 TCP 网络连接 182 中的一个将输入客户端数据报消息 180 朝着预定接收者 154 路由。网络传输软件通过参与合作动态路由协议来实现这一点。

[0070] 网格中的每个节点上的网络传输软件保持其自己的路由表 184 以便仅使用本地可用信息来指引输入消息。路由表是<目的地网络传输软件标识符、相邻网络传输软件标识符>的集合。每个此类三元组使目的地与去往目的地的消息应被转送到邻居相关联。邻居是与之存在活动输出 TCP 连接的节点。节点的近邻包括所有其邻居。

[0071] 网格中的节点可通过相应的 TCP 网络连接向任何邻居可靠地发送消息。此类消息到达邻居处或者导致 TCP 网络连接上的 TCP 错误。为了尽快地检测到连接中断,节点周期性地跨其所有 TCP 连接发送活动性消息,包括其应用连接 181 及其 TCP 网络连接 182。这些消息的频率是可配置的。

[0072] 节点处的网络传输软件每当运行网络传输软件实例的另一节点加入或离开其近邻时调度其路由表的重建。在节点等待重建其路由表的同时,对其近邻的任何其他改变触发整个调度量子的更新。因此,近邻中的递增变化导致此延期的递增延长。用于参与大型网格的节点的路由表的重建要求在网格尺寸方面为线性的努力,并且此延期减少了例如当快速连续地启动或停止许多节点上的网络传输软件时可存在的高网格通量的时段期间的中间路由表的不必要计算(和近邻快照的传输)。

[0073] 作为任何近邻变化的结果,节点保存新的近邻快照,将其网络传输软件标识符、单调增加的快照版本号以及近邻的新成员资格组合。某些实施方式使用自从 Unix 时代(1970-01-01T00:00:00Z[ISO 8601])以来所经历的纳秒作为快照版本号。节点不仅保存其自己的近邻快照,而且保存描述其他节点的许多近邻快照的集合。与路由表的不完全重建一致,网络传输软件发送包含其自己的近邻快照和接收者列表的近邻快照消息。接收者列表与当前邻居相同。该消息被发送到所有接收者。

[0074] 当网络传输软件接收到近邻快照消息时,当且仅当 1) 其从未从相关联的节点接收到近邻快照,或者 2) 其快照版本号超过与相应的保存近邻快照相关联的一个时,其保存所包含的近邻快照。在其他情况下,网络传输软件丢弃该消息且不采取关于它的进一步动作。这防止被长的路线或罕见网格拓扑任意地延迟的旧近邻快照退回节点的关于远程近邻的知识。假设网络传输软件保存近邻快照,则其然后计算其自己的邻居与包含消息的接收者之间的集合差。如果该差不是空集,则网络传输软件构造新的近邻快照消息,其包含外来快照和原始接收者的与先前计算的差并集。网络传输软件然后将该新消息发送到该差的所有成员。相应地,将不会循环地路由近邻快照消息;算法终止。无论是否实际上发送了任何新消息,网络传输软件调度其路由表的重建(或者更新显著延迟的重建的调度量子)。

[0075] 重建路由表的算法接受所有已保存的近邻快照(包括节点自己的)作为输入,并产生路由表作为输出。保存的近邻快照隐含地定义网格的连接性图表。该路由算法用执行节点的直接邻居来播种工作队列和新的路由表。其然后消耗该工作队列,添加仅用于尚未

被路由的目的地的新路线和工作队列项目。这组成连接性图表的宽度优先遍历,从而确保当首先遇到新的网络传输软件标识符时,所建立的路线将是可能的最短的。该算法具有线性空间和时间要求。具体地,其要求 $O(N)$ 空间,其中, n 是参与所考虑的网格的节点的数目,以及 $O(e)$ 时间,其中, e 是在这些节点之间存在的邻居关系的数目。

[0076] 近邻快照传播和路由表构造算法允许参与网格的所有节点并行地会聚以具有网格连接性的统一视图,并且每个节点将具有为其自己在图内的位置优化的路由表。当需要进行路由判定时,例如因为客户端数据报消息刚刚到达节点处,可使用仅本地可用信息进行判定。稳定网格的使用提供优点。例如,一旦网格关于节点成员资格和连接性静寂,则可在不要求更多控制消息业务开销的情况下进行网格中的所有路由判定。

[0077] 在某些实施方式中,其中网格可能不是稳定的,可以在不使用诸如 TCP 的生存周期 (TTL) 之类的机制的情况下防止客户端数据报消息的循环路由,该机制促使处理分组的每个路由器在重传之前将嵌入计数器递减,并且如果值达到零的话,则丢弃该分组。在某些实施方式中,平台中立网络传输层使用邮戳系统。当节点接收到客户端数据报消息且既不是其源也不是其目的地节点时,其在重传消息之前将其自己的网络传输软件标识符附加于邮戳列表。由源和目的地服务句柄编码的源和目的地网络传输软件标识符被自动地视为邮戳,因此要使源和目的地显式地附加其标识符将是冗余的。

[0078] 如果节点在目的地是某个其他节点的输入客户端数据报消息上发现其自己的邮戳,则其丢弃该消息以缩减无界循环路由。相应地,允许以每个客户端数据报更大开销为代价的任意长的路线。大多数环境可预期将建立其中每个节点使所有其他节点作为其邻居的网格集团。在此类集团中,开销限于必须的源和目的地网络传输软件标识符。

[0079] 对于大多数用户应用而言,实际网格的成员资格和连接性的知识是不必要的。这些应用分别简单地使用和提供服务作为客户端或服务器。希望提供服务的用户应用获取服务句柄并绑定适当的服务标识符。希望使用服务的用户应用采用静态已知服务标识符或静态已知服务句柄来对服务进行定位和接触。

[0080] 在某些实施方式中,某些用户应用监视网格健康并报告状态。为了支持此类用户应用,网络传输软件提供应用可预订以接收路由事件的通知的服务 240。具体地,每当一组节点的可到达性改变时,所有节点向每个感兴趣用户应用发送路由通知消息,其包含可到达性状态 { 可到达、不可到达 } 和表示其可到达性已改变的节点的网络传输软件标识符的列表。用户应用通过向其网络传输软件发送包括应开始接收路由通知的服务句柄的路由预定消息来登记对路由通知的兴趣。如果用户应用不再希望接收路由通知,则其可发送包含先前预订的服务句柄的路由不可预订消息。

[0081] 如图 12 中所示,在典型实施方式中,利用 (使用) 网格的用户应用具有两个特性中的至少一个或两个:其是提供特征集或服务 201 的服务提供者 200,或者其是请求并使用那些特征集或服务的服务客户端 202。此类布置可以遵守分布式计算的客户端服务器模型。不排除用户应用之间的端对端关系。还可以实现客户端服务器和端对端布置的组合。

[0082] 一旦用户应用已建立与在节点上托管的网络传输软件 206 的 TCP 连接 204,则用户应用获取其用来与其他用户应用 (位于本地或远程节点处) 通信的一个或多个服务句柄 208 的所有权。这些其他用户应用可以是联系服务句柄 208 以请求服务的客户端。其还可以是通过其自己的服务句柄来提供服务的服务器,在这种情况下,拥有服务句柄 208 的用

户应用可联系这些服务句柄以请求服务。符合的用户应用将服务句柄视为不透明原子值。然而,从节点的角度来看,服务句柄不是不透明的,而是<网络传输软件标识符,UUID>,其中,UUID是128位Leach-Salz变体4通用唯一标识符[RFC 4122]。

[0083] 为了获得用于其用作服务消费者、服务提供者或两者的服务句柄,用户应用向其网络传输软件发送包含新的会话标识符的请求服务句柄消息。会话标识符可以是例如64位整数值,其唯一地识别用户应用与其网络传输软件之间的请求响应交易。在接收到请求服务句柄消息时,网络传输软件用新服务句柄消息进行响应,其包含相同的对话标识符和新分配、统计上唯一的服务句柄。嵌入此服务句柄中的网络传输软件标识符表示为其分配的网络传输软件,其允许消息的正确路由。

[0084] 在这里,网络传输软件已在服务句柄的巨大全局空间210中创建新值。在用户应用能够使用新服务句柄之前,其向其网络传输软件发送开放服务句柄消息。此消息包含新对话标识符和新分配的服务句柄。当网络传输软件接收到此消息时,其向发送者登记该服务句柄,从而促使该服务句柄进入开放状态,并且用包括请求的对话标识符和ok的确认代码的客户端确认消息来答复。

[0085] 如果服务句柄向用户应用登记,则其是开放的;如果其未向用户应用登记,则其被关闭。所有服务句柄在关闭状态下开始。另外,每个未分配的服务句柄被网络传输软件视为关闭,使得关闭状态独立于服务句柄的存在。服务句柄的完整集合是{开放、关闭、不可到达}。(不可到达状态是被服务句柄通知机制用来指示到远程发布者的所有路线都已丢失的伪状态,如下面进一步讨论的。)

[0086] 想要充当服务提供者的应用通常将开放一个或多个服务句柄以侦听输入服务请求。不同于作为<IP地址、端口号>的暂时绑定的因特网套接字,服务句柄是持久性实体。从巨大的空间提取服务句柄,并且如果其跨服务提供者的所有实例在概念上描述相同的通信端点,则可以再使用该服务句柄。在某些实施方式中,服务客户端还持久性地使用服务句柄。服务句柄的此持久性及其使用允许创建并保持网格内的用户应用的专用网络。例如,如果服务提供者应用及其客户端应用实现先协定,则其可使用未广告的服务句柄进行通信,从而通过排除其他用户应用能够发现参与的服务句柄并向其发送客户端数据报的可能性而有效地使其通信专有化。

[0087] 在某些情况下,服务客户端将不知道其应与之通信以使用服务的确切服务句柄。为了更灵活地且匿名地支持服务客户端,服务提供者可发布绑定服务标识符消息,其包含新会话标识符和<服务标识符、开放服务句柄>的服务绑定214。服务标识符212是以服务提供者的客户端所预期的方式对服务进行命名的尺寸前缀UTF-8串。在接收到时,网络传输软件向服务目录276中输入服务绑定。该服务目录是所有服务绑定的集合。由于每个服务句柄还识别负责它的节点,即拥有用户应用被附着到的那个,服务目录指示在哪里能够联系所有服务。最后,网络传输软件用包含请求的会话标识符和ok的确认代码的客户端确认消息进行答复。服务提供者自由地将超过一个服务标识符绑定到开放服务句柄,例如通过针对每个期望绑定发送一个绑定服务标识符消息。

[0088] 当发生本地服务提供的变化时,本地节点的网络传输软件保存将其网络传输软件标识符、单调递增快照版本号以及本地服务绑定的新集合组合的新服务目录快照277。某些实施方式可使用自从Unix时代(1970-01-01T00:00:00Z[ISO 8601])以来所经历的纳秒作

为快照版本号。节点不仅保存其自己的服务目录快照，而且保存描述由被附着于其他节点的用户应用提供的服务的服务目录快照的集合。每当节点保存其自己的本地服务提供的服务目录快照时，作为服务绑定的建立或解除的结果，其调度将发送包含此服务目录快照和接收者列表的服务目录快照消息的任务。接收者列表与当前邻居相同。该消息被发送到所有接收者。

[0089] 在节点等待发送的同时，对其本地服务提供的任何其他改变触发整个调度量子的更新。因此，递增更新导致此延期的递增延长。此递增延长避免了诸如当快速连续地开始和停止许多节点时占优势的高服务通量的时段期间的服务目录快照的不必要传输。

[0090] 当节点接收到服务目录快照消息时，当且仅当 1) 其从未从相关联的节点接收到服务目录快照或者 2) 其快照版本号超过与相应的保存服务目录快照相关联的那个时，其才保存所包含的服务目录快照。在其他情况下，节点丢弃该消息，并且不采取关于该消息的进一步动作。因此防止被长的路线或罕见网络拓扑任意地延迟的旧服务目录快照退回节点的关于远程服务提供的知识。

[0091] 假设节点保存服务目录快照，其通过比较旧服务目录快照和新服务目录快照来计算两个集合。第一集合包括要添加到服务目录的绑定，并且体现存在于新快照而不在于旧快照中的绑定。第二集合包括要从服务目录去除的绑定，并且体现存在于旧快照而不在于新快照中的绑定。第一集合的内容立即被添加到服务目录；第二集合的内容立即被从服务目录去除。网络传输软件然后计算其自己的邻居与包含消息的接收者之间的集合差。如果该差不是空集，则网络传输软件构造新服务目录快照消息，其包含外来快照和原始接收者与先前计算的差的并集。网络传输软件然后将该新消息发送到该差的所有成员。将不会循环地路由服务目录快照消息，并且算法终止。

[0092] 服务目录快照传播和服务目录构造算法允许参与网格的所有节点并行地会聚以具有服务可用性的统一视图（公文）298。当服务查询到达时，可使用仅本地可用信息对其进行分辨。稳定的服务公文可以提供优点。例如，一旦稳定服务公文实质化，则可在不要求更多控制消息业务开销的情况下实现所有服务分辨判定。

[0093] 为了找到服务，用户应用向其节点发送定位服务消息。此消息包括新对话标识符、服务标识符匹配模式、期望匹配模式、期望定位模式以及作为毫秒的 64 位编码的响应超时。服务标识符匹配模式是尺寸前缀 UTF-8 串，其语义由所选匹配模式确定，但是是服务标识符或意图与一个或多个服务标识符匹配的 Java 正规表达式（由例如 `java.util.regex.Pattern` circa 1.6.0_19 定义）。在某些实施方式中，匹配模式可以是 {精确、模式}，其中，精确意指匹配模式将被针对当前服务绑定逐字地匹配，并且模式意指将使用正则表达式匹配引擎来应用匹配模式。在某些实施方式中，定位模式是 {所有、任何}，其中，所有意指网络传输软件应用每个匹配服务绑定进行答复，并且任何意指网络传输软件应任意地用任何匹配服务绑定进行答复。

[0094] 当节点接收到定位服务消息时，其尝试针对其完整服务目录的指定查找。如果发现匹配，则节点立即用包括相同会话标识符和适当数目和种类的匹配服务绑定的服务列表消息进行答复。提供完整绑定，使得请求者可访问精确服务标识符以及其绑定服务句柄；这对使用模式匹配模式的客户端而言特别有用。如果未发现匹配，则节点向一组待决请求添加该请求，并且调度当在定位服务消息中指定的响应超时期满时将启动的定时器。

[0095] 每当由于处理绑定服务标识符消息或服务目录快照消息而建立新的服务绑定时，节点针对新的服务绑定检查每个待决请求。任何匹配导致从待决请求集合的立即去除、定时器的禁用已经适当服务列表消息的传输。如果定时在相应的请求与任何服务绑定匹配之前期满，则节点从待决请求集合去除请求，并发送不包含服务绑定的服务列表消息。

[0096] 由于服务列表消息可包含多个服务绑定，所以其被布置成希望联系特定服务的服务客户端将判定要选择哪个服务句柄。相等的服务标识符将指定相等的服务，因此希望用特定服务标识符来联系服务的用户应用可任意地从所检索的绑定中选择被绑定到该服务标识符的任何服务句柄。一般地，用户应用将不能针对相等的服务标识符在服务句柄之间智能地进行判定，因此将只能进行任意判定。可操作负责网格的组织，从而向不同的服务分配不同的名称并向相同的服务分配相同的名称。虽然相等的服务标识符将表示相等的服务（即，以相同的方式做相同的事的服务），但通常用户应用不能在嵌入相等的服务标识符的服务绑定之间智能地进行判定。可存在最佳判定，例如由查询进行的所有服务应答中的压力最小或最不远的，但是用户应用通常在得出明智判定的错误有利点处。网络传输软件有时能够代表服务客户端而进行更好的判定，例如当在定位服务消息中指定适当的定位模式时。未来定位模式可以直接地支持服务提供者接近和负载平衡。

[0097] 服务提供者可将先前针对其开放服务句柄中的一个建立的任何服务绑定解除绑定，例如通过向其网络传输软件实例发送包含新的会话标识符和服务绑定的解除绑定服务标识符消息。接收到此类消息的节点从其本地服务提供中去除该服务绑定，保存新的服务目录快照，并且调度服务目录快照消息的传输，如上文详细地描述的。在本地更新完成之后，网络传输软件用包括请求的会话标识符和 ok 的确认代码的客户端确认消息进行答复。

[0098] 如图 8 中所示，两个开放服务句柄 302、304 可交换客户端数据报 306。在某些实施方式中，以这种方式（亦即，使用数据报）在用户应用之间传输所有用户数据。由于由网络传输软件提供的此基本通信协议根本上是无连接的，所以重要的是用户应用知道其对端何时可用于发送和接收数据报。在某些实施方式中，用户应用 310 预订开放服务句柄以接收由另一服务句柄 312 发射的事件通知 308。前一服务句柄是订户且后者是发布者。为了向发布者预订服务句柄，用户应用向其网络传输软件发送服务句柄预订消息，其包含新的对话标识符、订户以及发布者。在本地地登记客户端的兴趣之后，网络传输软件用包括请求的会话标识符和 ok 的确认代码的客户端确认消息进行答复。

[0099] 预订的服务句柄可偶尔地接收关于其发布者的服务句柄通知消息。服务句柄通知消息体现被登记到接收客户端的订户、发布者以及发布者的状态大约消息创建时间。在某些实施方式中，当且仅当发布者改变状态时，才创建并发送此类消息。没有重复通知被节点发送或被客户端接收。发布者状态改变的所有通知因此是真实的，并且可在不需要复杂的客户端侧状态跟踪逻辑的情况下由客户端相应地作出反应。

[0100] 在某些实施方式中，客户端使用这些通知作为数据阀。

[0101] 发布者开放的通知指示客户端可开始向发布者发送客户端数据报，并且可根据通信的风格预期从发布者接收消息。

[0102] 发布者关闭的通知指示客户端不应向发布者发送新的客户端数据报。由于在网格中可存在许多路径，所以某些客户端数据报可在发送关闭通知之后到达发布者。从关闭的服务句柄到达的此类客户端数据报可被丢弃。在某些实施方式中，特定应用域应驱动判定

是否丢弃此类客户端数据报的此策略判定。

[0103] 发布者不可到达的通知指示客户端和发布者的网络传输软件实例之间的最后的路线已经消失。虽然发布者是不可到达的,但其可经历其订户未被告知的状态改变。由于所有节点间链路都是全双工的,所以节点的可到达性(因此的不可到达性)都是对称的。如在上述情况下一样,此类不可用通知可与去往订户的客户端数据报竞争。在某些实施方式中,忽视由在不可到达发布者处发起的节点接收到的任何通知,即其未被向前转送至订户。开放或关闭的发布者状态的后续接收意味着本地和远程节点再次地相互可到达;报告的状态是两个节点之间的路线的大约重建。

[0104] 有时,客户端可能不再希望在特定订户处从特定发布者接收通知。该客户端可发送服务句柄退订消息,包含新的会话标识符、订户以及发布者。在接收到时,网络传输软件撤销订户对发布者的兴趣并用客户端确认消息进行答复,其包括请求的会话标识符和 ok 的确认代码。

[0105] 节点 330 中的传输层软件实例 331 采用服务句柄预订管理器 332 来跟踪其客户端的服务句柄预订。该预订管理器出于管理预订和服务句柄状态转移的目的而保持数据结构的多个集合。在某些实施方式中,第一集合包括以下:

[0106] 1. 客户端订户映射,映射 {发布者→本地发布者},其中,发布者是服务句柄,并且本地订户是预订键的本地登记服务句柄的集合。此映射支持通知的高效传送。

[0107] 2. 客户端发布者映射,映射 {本地订户→发布者},其中,本地订户是本地登记的服务句柄,并且发布者是键预订的服务句柄的集合。此映射支持服务句柄关闭时的高效清理,例如当服务句柄被显式关闭时或者当失去客户端连接时。

[0108] 3. 由网络传输软件实例映射的发布者,映射 {网络传输软件标识符→发布者},其中,网络传输软件标识符表示参与网格的任何节点,并且发布者是向键的指示物登记的服务句柄的集合。此映射支持对节点上的网络传输软件的可到达性中的改变的高效反应。

[0109] 当节点接收到服务句柄预订消息时,其服务句柄预订管理器前后紧接地更新这些映射。结果:客户端订户映射现在列出其发布者的订户集合中的订户;客户端发布者映射现在列出订户的发布者集合中的发布者;由网络传输软件实例映射的发布者现在列出其网络传输软件标识符的已登记发布者的集合中的发布者。该本地网络传输软件注意这是否是初始预订,亦即第一次其已登记服务句柄中的一个向指定发布者预订。

[0110] 当节点接收到服务句柄退订消息时,其服务句柄预订管理器也前后紧接地更新这些映射。结果:客户端订户映射不再列出其发布者的订户集合中的订户;客户端发布者映射不再列出订户的发布者集合中的发布者;由网络传输软件实例映射的发布者不再列出其网络传输软件标识符的已登记发布者的集合中的发布者。该本地网络传输软件注意这是否是最后退订,亦即不再存在向指定发布者预订的任何已登记服务句柄。

[0111] 该服务句柄预订管理器将双层机制用于管理服务句柄预订。

[0112] 第一层使用上述数据结构使开放订户与发布者相关联。当客户端向登记到已被附着于同一节点的另一客户端的发布者预定其服务句柄中的一个时,只需要第一层以管理该预订并正确地传送服务句柄状态通知。由于仅涉及到一个节点,所以每当发布者变得开放或关闭时,节点可通过到相应客户端的全双工应用连接来直接地通知所有本地订户。类似地,节点不需要告知本地订户本地发布者是不可到达的。为了从特定本地发布者传送通知,

节点从客户端订户映射获取与发布者相关联的集合。网络传输软件在此集合范围内进行迭代,并向用于每个已登记订户的每个客户端发送一个服务句柄通知消息。在某些实施方式中,节点每当检测到本地发布者的状态的改变时(例如由于处理开放服务句柄消息)这样做。

[0113] 第二层使得具有开放订户的节点与远程发布者相关联。为了支持此第二层,服务句柄预订管理器保持数据结构的第二集合。数据结构的第二集合的示例包括:

[0114] 1. 网络传输软件订户映射,映射{本地发布者→网络传输软件标识符},其中,本地发布者是本地登记的服务句柄,并且网络传输软件标识符是表示具有对键的订户的远程节点的一组网络传输软件标识符。此映射支持通知的高效传输。

[0115] 2. 网络传输软件发布者映射,映射{网络传输软件标识符→本地发布者},其中,网络传输软件标识符表示远程节点且本地发布者是对于其而言键具有订户的一组发布者。此映射支持该机制的高效实施方式,其在网络传输软件循环之后传播服务句柄状态。

[0116] 3. 网络传输软件预订会话映射,映射{网络传输软件服务句柄预订键→预订会话}。网络传输软件服务句柄预订键是<发布者、网络传输软件标识符>,其中,发布者是本地登记的服务句柄且网络传输软件标识符描述具有对此发布者的订户的节点。预订会话是<会话标识符、收割机状态数>,其中,会话标识符描述嵌入最近接收的第二层预订控制消息内的会话标识符。收割机状态数对应于负责清理已关闭会话(还在下面讨论)的接收者任务的特定性能。此映射提供预订会话的信息单调性。

[0117] 用于第二层预订的控制消息的示例包括:节点服务句柄预订、节点服务句柄退订、节点请求服务句柄通知、节点服务句柄通知。可通过中间节点将这些消息中的任何一个在途中路由到其目的地。

[0118] 在网格中可存在许多可用路线(或者导致重传的丢弃网络帧),并且控制消息不按顺序到达是可能的。在某些实施方式中,忽视并非新的控制消息以防止预订会话的退回。如果 1) 不存在关于预订键的会话,或者 2) 嵌入消息中的会话标识符比在进行中的会话中记录的那个更新,则认为第二层预订控制消息是新的。如果第二层预订控制消息被确定为是新的,则接收消息的节点更新网络传输软件预订会话映射,使得适当的预订键随后绑定新的会话,其包括嵌入消息中的会话标识符和下一收割机状态数。在接收到第二层预订控制消息之后不久,接收者用可路由节点确定消息进行不可靠地答复,该可路由节点确认消息包含请求的会话标识符和 ok 的确认代码。主处理可以在发送此确认之后发生。

[0119] 对远程发布者的每次初始预订促使本地网络传输软件通过可靠地将节点服务句柄预订消息路由到发布者的节点而使其本身向发布者预订。此消息包括新的会话标识符和适当的网络传输软件服务句柄预订键,其指定发布者和预订节点。当节点接收到此类消息时,其提取预订键并在网络传输软件预订会话映射中查找与之相关联的会话。如果消息是新的,则接收者在锁定步骤中更新另一第二层映射。结果:网络传输软件订户映表现在在其发布者的订户集合中列出预订节点;网络传输软件发布者映射现在在预订节点的发布者集合中列出发布者。最后,接收者可靠地向预订节点发送节点服务句柄通知消息,其包括新会话标识符、订户的网络传输软件标识符、发布者以及发布者的状态大约消息创建时间。当在启动节点上的网络传输软件之后不久发送关于关闭的发布者的通知时出现附加的复杂性;下面更详细地对这些进行描述。

[0120] 所述节点可偶尔地接收到关于其发布者的节点服务句柄通知消息,例如当发布者改变状态时,因为其网络传输软件处理相应的开放服务句柄消息。如果节点服务句柄通知消息是新的,则接收者从客户端订户映射获取与所述发布者相关联的集合。接收节点在此集合内进行迭代并向用于每个已登记订户的每个客户端发送一个服务句柄通知消息。

[0121] 在从远程发布者接收到最后退订时,本地节点通过将节点服务句柄退订消息可靠地路由到发布者的节点而将其本身从发布者退订。此消息包括新的会话标识符和适当的网络传输软件服务句柄预订键,其指定发布者和退订节点。当节点接收到此类消息时,其在网络传输软件预订会话映射中查找与指定的预订键相关联的会话。如果消息是新的,则接收者在锁定步骤中更新另一第二层映射。结果:网络传输软件订户映射不再在其发布者的订户集合中列出退订节点;网络传输软件发布者映射不再在退订节点的发布者集合中列出发布者。

[0122] 第二层预订控制消息在传送中可能丢失。在某些实施方式中,可靠的传送是必需的,例如用于服务句柄预订机制的良好性能。在某些实施方式中,当发送这些控制消息时,副本被存储在重传列表上。另外,调度用以每个完整量子循环地执行一次的任务。可以基于系统或用户的需要来配置此量子,即重传速率,并且其具有 5,000ms (5s) 的默认值。此任务在执行时将控制消息的副本发送到其目的地。当节点接收到节点确认消息时,其从重传列表去除副本消息,该副本消息的会话标识符匹配,并取消其相应的重传任务。不要求可靠地发射节点确认消息,因为其不能出现导致相关联的控制消息的反射性重传。

[0123] 有时,节点处的网络传输软件实例可终止,例如由于处理重启消息或关机消息、节点的操作系统进程的用户或系统发起的终止或者软件错误。在这种情况下,网络传输软件实例及其邻居与客户端之间的 TCP 网络连接和应用连接在没有更多控制消息的传输的情况下自发地放弃。在关机事件之后,认为节点是参与网格的其他节点不可到达的。同样地,由其客户端登记的任何服务句柄也被认为是不可到达的。每当节点确定参与节点网格的某些节点已变得不可到达时,其使用不可到达节点的网络传输软件标识符作为键通过网络传输软件实例映射迭代地查询发布者。网络传输软件然后计算所有结果得到的集合的并集,以确定现在其订户不可到达的发布者的完整集合。网络传输软件在此集合范围内进行迭代,并向用于每个已登记订户的每个客户端发送一个服务句柄通知消息。

[0124] 当已关闭的节点和 / 或其网络传输软件重启时,许多客户端将尝试自动地重新连接到新的网络传输软件实例并重建其服务句柄、服务绑定以及预订。以免当已重启节点的存在被其他节点检测到时这些客户端的服务句柄被认为是关闭的,重启节点遵守服务重建宽限时段。此宽限时段的持续时间可被用户配置且具有 30,000ms (30s) 的默认值。

[0125] 在宽限时段期间,节点将不发送报告用于其包含的发布者的关闭状态的服务句柄通知消息或节点服务句柄通知消息。网络传输软件替代地将该消息排在服务重建宽限队列上以用于在宽限时段期满时传输。如果发布者的状态在此时间期间转换,例如网络传输软件接收到适当的开放服务句柄消息,则排队的消息被丢弃,并发送替换消息以报告用于其发布者的开放状态。当宽限时段期满时,仍在宽限队列上的所有消息被发送到其相应目的地。

[0126] 从客户端的角度来看,任何不可到达的发布者可能在其节点或网络传输软件的停机期间正在任意地改变状态。如果不可到达的网络传输软件实例未循环,但某个其他条件

已中断通信,则情况可能的确如此。未插接网络线缆可具有此效果。另外,可以允许本地订户从不可到达的发布者退订,即使发布者的网络传输软件本身根据定义是不可到达的。

[0127] 为了解决这种情况,两个节点必须在可到达性的相互确定时协调其预订和服务句柄状态。每个节点通过在其再次变得可到达时向其远程伙伴发送节点请求服务句柄通知消息来实现此效果。此消息包含新的对话标识符、通过网络传输软件实例映射针对发布者中的目的地节点记录的发布者的完整集合以及预订网络传输软件实例的网络传输软件标识符。

[0128] 当网络传输软件接收到节点请求服务句柄通知消息时,其首先使用预订节点的网络传输软件标识符和请求通知 UUID、即从被网络传输软件预留供其内部使用的范围统计分配的 UUID 来计算特殊网络传输软件服务句柄预订键。此预订键被具体地用来将特殊会话内的节点请求服务句柄通知消息排序。在某些实施方式中,在接收网络传输软件本地的发布者的完整集合被嵌入消息中,其在消息创建时间是瞬间正确的。在此类实施方式中,特殊预订键的使用防止关于第二层预订的知识的聚合退回。如果消息是新的,则接收者计算三个集合:

[0129] 1. 忘记的发布者。这是不再存在于预订节点的预订列表中的发布者的集合。为了计算此集合,首先查询具有预订网络传输软件的网络传输软件标识符的网络传输软件发布者映射。这些是最后已知的发布者。提取封装在节点请求服务句柄通知消息中的发布者。这些是当前发布者。期望的结果是最后已知的发布者与当前发布者之间的集合差。

[0130] 2. 新的发布者。这是自从两个节点相互可到达的最后时间以来对于预订节点的预订列表而言是新的发布者的集合。期望结果是当前发布者与最后已知的发布者之间的集合差。

[0131] 3. 保持的发布者。这是在停机之前和之后存在于预订节点的预订列表中的发布者的集合。这是当前发布者和最后已知发布者的集合交集。

[0132] 出于更新相关联的预订会话和第二层映射的目的,已忘记的发布者的集合中的每个发布者被视为如同其是单独节点服务句柄退订消息的目标一样。同样地,出于相同的目的,新发布者的集合中的每个发布者被视为如同其为单独节点服务句柄预订消息的目标一样。保持的发布者的集合中的每个发布者被视为如同其为单独冗余节点服务句柄预订消息的目标一样,因此只有相关联的预订会话被更新。另外,构造并发送所有适当的节点服务句柄通知消息,根据需要而遵守服务重建宽限时段。

[0133] 接收第二层预订控制消息序列的效果与其被接收到的顺序无关,这是预订机制的本质方面且允许对发布者状态的改变的可靠通知。两层机制与一层机制相比可以减少网络业务且可以减少通知等待时间。具体地,当托管网络传输软件的节点被部署在大型格栅状网格中时,预订架构至少缩放至向数百个或数千个发布者不同地预订的数百万个服务句柄。

[0134] 网络传输软件预订会话映射不丢弃任何会话。在某些实施方式中,大多数服务句柄被动态地分配以满足用户应用的通信要求。此类服务句柄因此在其有限的寿命内仅仅是可行的发布者;一旦被关闭,一般地可预期其再次变得开放。在这些情况下,网络传输软件预订会话映射 400(图 9) 将累积关于永久关闭服务句柄的会话。

[0135] 在某些实施方式中,为了防止由于累积会话而引起的无界存储器增长,收割机任

务 404 以可配置的间隔周期性地执行。作为默认,收割机时段为三个小时。当收割机任务执行时,其收集满足至少以下准则的每个会话,即 1) 对于其网络传输软件服务句柄预订键 406 而言没有预订是现存的,以及 2) 其收割机状态数 408 小于当前收割机状态数。然后,收割机任务事务地从会话映射去除所有此类会话。最后,收割机任务还递增收割机状态数。在某些实施方式中,相对长的默认收割机时段足以针对上述大规模部署情形而保持 1GB 堆极限。

[0136] 在服务句柄 401 变得开放之后的任何时间,其已登记的用户应用 403 可通过向其网络传输软件实例 410 发送包含新会话标识符 412 和服务句柄的关闭服务句柄消息而放弃所有权。此消息被网络传输软件处理促使服务句柄被撤销,从而促使服务句柄进入关闭状态。与服务句柄相关联的任何服务标识符 420 和预定 422 然后被忘记,如同应用适当的解除绑定服务标识符和服务句柄退订消息。到达已关闭服务的客户端数据报在目的地网络传输软件处被丢弃。一旦该消息被完全处理,则网络传输软件用包括请求的会话标识符和 ok 的确认代码的客户端确认消息进行答复。如果用户应用突然地从其网络传输软件断开连接,则网络传输软件自动地关闭登记到用户应用的所有开放服务句柄。这如同用户应用已首先发送用于其开放服务句柄中的每一个的关闭服务句柄消息一样发生。

[0137] 在某些情况下,网络传输软件可能不能成功地处理控制消息。在接收到任何控制消息时,网络传输软件在判定允许相应的操作进行之前针对其内部状态检查该消息。例如,用户应用不能单独地或通过另一用户应用来开放已登记为开放的服务句柄。同样地,用户应用不能关闭被另一用户应用登记为开放的服务句柄。这些错误条件可意味着无意义的操作,如同关闭已关闭的服务句柄,或者特权的违背,如同废除用于由与请求者不同的用户应用拥有的服务句柄的服务绑定。此类操作产生客户端确认消息,其确认代码不同于 ok。在某些实施方式中,客户端检查结果得到的确认代码以相应地进行,并且不进行控制消息的处理是成功的假设。

[0138] 我们现在考虑网络传输软件 500 的输入 / 输出 (I/O) 系统 502 (图 10) 的操作。在某些实施方式中,节点的 I/O 子系统缩放为管理数以万计的同时 TCP 连接的数百个线程。理论极限较高,只是节点的连接性以 TCP 的限制为界。在节点及其外部邻居与内部客户端之间可存在不超过 2^{16} 个 TCP 连接。这是由 TCP 施加的设计极限,并且其对应于 TCP 端口号的完整空间。当在节点上运行的其他进程也消耗 TCP 端口号时,实际极限可较低。

[0139] 网络传输软件通过提供虚拟信道 504 来克服这些限制,其中的许多可在单个共享 TCP 连接 505 上对数据进行复用。在某些实施方式中,在任何两个相邻节点之间存在精确地一个 TCP 网络连接 505,并且在节点与客户端 508 之间存在精确地一个应用连接 506。在某些实施方式中,这些方之间的所有网络业务必须跨这些单数 TCP 连接流动。客户端登记的每个服务句柄建立活动通信端点;可以存在特定客户端登记的非常多的服务句柄。每个其他服务句柄是潜在的通信端点。任何两个服务句柄可以定义信道 504,并且任何两个开放服务句柄 511、512 定义活动信道。节点的内部数据结构缩放为管理分散在无数的客户端上的数百万个开放服务句柄。

[0140] 使用以下示例来举例说明信道的可缩放性及其他优点。使 $M(N)$ 为用于客户端 N 的本地网络传输软件实例。使 $S(N)$ 为向客户端 N 登记的服务句柄的集合。给定两个客户端 A 和 B ,假设在 A 和 $M(A)$ 之间存在精确地一个应用连接,对于 B 和 $M(B)$ 而言是同样地,并且

在 $M(A)$ 与 $M(B)$ 之间存在精确地一个 TCP 网络连接。于是,只需要 3 个 TCP 连接以支持笛卡尔乘积 $S(A) \times S(B)$ 。给定 $S(A)$ 和 $S(B)$ 中的每一个可以是包含 1 百万个开放服务句柄的集合,活动连接的数目可超过 1 万亿。信道相比于专用 TCP 连接提供巨大的可缩放性优点。

[0141] 为了使得网络传输软件能够缩放至任意大型部署方案,其 I/O 机制需要独立于网络负载正确地操作。可缩放 I/O 算法展示与业务量成反比的性能和相对于业务量不变的正确性。可缩放系统可经受死锁条件。

[0142] 网络传输软件的 I/O 子系统的至少某些实施方式的重要方面在于所有规模没有死锁。此自由是理论和实践两方面的。在某些实施方式中,为了获得免于死锁,设定了要满足的至少以下准则:1) 通过系统调用提供的所有 I/O 操作是异步的,并且 2) 到保护内部数据结构的关键区段的进入条件不阻止正在执行的线程达任意的时间量。在某些实施方式中,为了满足 2),需要适当地调度等待访问关键区段的线程。

[0143] 网络传输软件通过仅使用异步的平台 I/O API 来满足第一条件。来自 TCP 连接的所有读取、到 TCP 连接的写入、新 TCP 连接的发起以及 TCP 连接的建立是异步地执行的,只有当可在不会不确定地阻止正在执行的线程的情况下完成操作时消耗资源。具体地,在某些实施方式中,当发起新的连接时,仅使用异步 DNS 方案。用于 DNS 方案的平台 API 是典型地同步的,尤其是在 UNIX® 变体和衍生物上。在某些实施方式中,通过使用异步自定义 API,网络传输软件仍避免在所有情况下和针对所有支持的平台的同步 DNS 方案。

[0144] 第二条件的满足使用如下架构支持。

[0145] 如图 11 中所示,在某些实施方式中,网络传输软件的 I/O 子系统 502 包括至少三种实体:单个协调器 522,具有管理线程并缓存具体化和串行化消息的职责;一个或多个(例如四个)代理 524,其每一个管理不同种类的 TCP I/O 事件;以及一个或多个(例如)许多管道 526,其每一个丰富单个基于套接字的 TCP 连接 505。

[0146] 协调器提供两个任务执行器,其每一个得到不同线程池的支持。写任务执行器 528 被预留给执行其独有功能是向套接字写入单个串行化消息的任务之用。一般任务执行器 530 可用于执行所有其他任务,但是主要被用于执行其独有功能分别地是从套接字读取单个串行化消息或完成异步 TCP 连接的任务。两个任务执行器的分离通过减少写与其他活动、特别是读之间的竞争来改善性能,但并不是算法正确所必需的。经验证据显示此分工导致改善的吞吐量,并且此改善足以批准增加的复杂性。

[0147] 希望利用这些线程池 532、534 中的一个的线程通过向相应任务执行器的无界任务提交队列 537、539 来这样做。每当任务执行器具有空闲线程时,其使在任务提交队列的首位的任务出列且被布置成用于使空闲线程将其执行。任务执行因此相对于任务提交而言是异步的。任务执行器的主要客户端是四个代理。

[0148] 协调器还跟踪有待于传输的所有消息的聚合存储器利用并实行缓冲阈值。该缓冲阈值是可配置参数,并表示节点将缓存的字节的大约数目。缓冲标签 540 是当前被缓存的字节数的协调器估计。消息的尺寸是其完整存储器覆盖区,包括诸如其对象报头之类的“不可见”系统开销。每个消息还知道其串行化形式的尺寸。出于说明聚合存储器利用的目的,协调器将消息视为如同其本征典型要求是两个覆盖区中的较大的一个一样。这简化并加快了计数。

[0149] 存在四个代理,每个基本种类的 TCP 事件一个。读代理 536 管理异步读取。当操

作系统的 TCP 实施方式指示用于特定套接字 527 的数据已到达,读代理在一般任务执行器上将在被执行时将读取如可从相关联的网络缓冲获得的那样多的字节并将其附加于负责套接字的管道所拥有的消息汇编缓冲的任务排队。特定读取可能不导致使来自消息汇编缓冲的完整消息具体化的能力。消息的串行化形式具有足以允许高效逐步存储和汇编的内部结构。当读取导致完整消息的汇编和具体化时,其被同步地处理。

[0150] 连接代理 538 和接受代理 540 分别地负责建立输出和输入 TCP 连接。当操作系统指示连接已完成时,适当的代理在一般任务执行器上将在被执行时将创建并配置提取新套接字的管道的任务排队。已被延期直至连接建立完成的任何动作将被同时地执行。

[0151] 写代理 542 管理异步写。当操作系统指示可将数据写到特定套接字时,写代理在写任务执行器上将在被执行时促使负责套接字的管道将当前传输窗口可用性所允许的尽可能多的待决消息串行化并发送的任务排队。特定写可不导致整个消息的传输。一般地,管道在将附加消息串行化并发送之前完成部分发送消息的传输。

[0152] 网络传输软件使用管道与邻居和客户端通信。管道 526 封装套接字 528 并使对其访问抽象化 551。管道以允许其客户端对消息的串行化施加细粒度控制的方式提供异步读和写能力。客户端通过让协调器发起或接受 TCP 连接而获得管道。当相对于连接发起而异步地建立 TCP 连接时,客户端指定将在 TCP 连接建立时执行的配置动作。

[0153] 在使用中,配置动作将翻译链绑定到管道。翻译链 548 包含模块化、可插接翻译器 550 的有序序列。翻译器用于在消息的串行表示之间双向地迁移。翻译器具有写转换器和读转换器。每个转换器将数据的缓冲接受为输入并产生数据的缓冲作为输出。写转换器接受朝着套接字流动的数据的缓冲;读转换器接受从套接字流动的数据的缓冲。可在写方向上应用翻译链,并且翻译链然后接受具体化的消息并按照客户端指定的顺序通过其翻译器的写转换器将其传递,以产生将被写到导管的套接字的最后串行形式。相反地,当在读方向上应用翻译链时,其接受来自导管的套接字的最后串行形式,按照相反的顺序应用其翻译器的读转换器,并产生具体化的消息。

[0154] 翻译链可被用于各种目的,例如实行协议要求、压缩流、将流加密等。翻译器可以有状态的,从而允许翻译链改变消息的事务边界;最小翻译量子可包含多个协议消息。

[0155] 配置动作还使读动作与管道相关联。此动作在管道的翻译链产生具体化的消息时执行。此动作与配置动作异步地且与从套接字网络读缓冲进行的数据的实际读取同步地执行。该动作在由一般任务执行器管理的线程中运行。为了允许网络传输软件没有死锁,读动作不执行可阻止达任意时间量的任何操作。此约束具体地适用于直接 I/O 操作。然而,读动作在不担心死锁的情况下将消息排队以便在任何管道上传输。每当管道被告知已在其套接字上接收到数据时,其通过其翻译链在读方向上传递此数据。一旦足够的数据已穿过翻译链,使得一个或多个具体化的消息可用,则按顺序一次一个地对其中的每一个执行读动作。

[0156] 客户端可向管道写入消息。在某些实施方式中,这在任何时间且在任何背景下都是允许的。被写入管道的消息不立即被串行化并使用底层套接字发送。首先,从单调递增计数器为其分配消息号码。然后在管道的两个传输队列中的一个上将其排队:控制队列 500,被预留类似于开放服务句柄和绑定服务标识符之类的高优先级控制消息;以及写队列 562,被用于类似于活动性的客户端数据报消息和低优先级控制消息。管道将对任一队列

的任何写入告知协调器,从而允许协调器将缓冲标签递增新排队消息的尺寸。网络传输软件保证在管道的控制队列上排队的消息最后将被串行化和发射。

[0157] 如果对管道的写入促使缓冲标签超过缓冲阈值,则可丢弃在管道的写队列上排队的消息。协调器保持管道的优先级队列,称为受害者队列 562,按照在每个管道的写队列上排队的最旧消息的消息号码排序。在某些实施方式中,当且仅当管道具有在其写队列上排队的的一个或多个消息时,其才出现在此优先级队列中。当对管道的写入促使缓冲标签超过缓冲阈值时,协调器丢弃消息直至缓冲标签不再超过缓冲阈值为止。

[0158] 具体地,协调器去除受害者队列的头部,去除并丢弃其写队列的头部,将缓冲标签递减丢弃消息的尺寸,将管道重新插入受害者队列,并重复该过程直至缓冲标签小于缓冲阈值为止。最缓慢流动的管道首先被处罚,从而允许沿着其他管道的业务继续前进。在某些实施方式中,网络传输软件客户端采用高级流协议 29 来相互通信,并且丢弃被最快重传的消息。

[0159] 在某些情况下,可设想只有高优先级控制消息在管道上排队,但是缓冲标签由于大量的控制消息而以某种方式超过缓冲阈值。在这种情况下,协调器可以继续不确定地缓存消息,并且不遵守缓冲阈值。

[0160] 当管道变得适合于向其套接字写入数据时,其首先发送尽可能多的当前完全翻译的缓冲。如果管道成功地消耗并发送此缓冲,其在平常情况下可能已是空的,则其使消息出列。如果存在在管道的控制队列上排队的消息,则排队消息中的最旧的一个出列;否则,管道使写队列上的最旧消息出列。这样,算法优选将高优先级控制消息串行化并发送。不仅此类消息在其处理中更有可能显示出时间敏感性,而且其对网络传输软件施加较高压力,因为即使在重负载下,网络也不能自由地将其丢弃。

[0161] 已使消息出列,管道命令协调器将其缓冲标签递减该消息的尺寸。然后,管道在写方向上通过翻译链来传递该消息以产生串行化缓冲。如果未产生缓冲,则管道将翻译链排序以冲洗。如果未产生缓冲,则管道放弃该传输过程并等待新消息的排队。假设已经获得缓冲。该管道命令协调器将其缓冲标签递增缓冲的尺寸,可能促使在一个或多个管道的写队列上排队的旧消息被丢弃。然后,管道发送如套接字的传输窗口可用性允许的那样多的产生的缓冲并适当地将缓冲标签递减。

[0162] 在某些实施方式中,为每个管道、代理以及协调器配备控制对其数据结构的访问的再进入锁。管道的使用能够驱动锁获取。例如,希望获取用于<管道、协调器、代理>的特定三元组的锁的线程按照在元组中指定的顺序获取锁以避免死锁的可能性。网络传输软件例如使用将确保实施方式的正确性并尽快地检测与正确锁定顺序的偏差的技术来实现、例如严格地实现锁定顺序。在某些实施方式中,所获取的锁在短时间段内被管道拥有,例如小于 1ms,允许高吞吐量。

[0163] 相对于启动、停止以及重启,网络传输软件已被设计成高度可配置且提供用于设置可配置参数的机制。例如,为了支持各种部署方案,可使用以下来指定这些参数,1) 特定于平台的命令行,2) XML 配置文档,其最外元素是 <configuration>,或者 3) Java 系统性质,或者那些中的两个或更多的某个组合。如果通过这些机制来多重地指定特定参数,则网络传输软件将不会在针对参数给定的所有值在语义上匹配之前开始。否则,网络传输软件发出描述所检测的不相干性的错误消息以允许终端用户以直接的方式检查运行网络传输软

件的设置。终端用户不必记住配置源的优先规则,并且可以使用从错误消息获得的信息来确定其源不一致的参数的实际运行值。

[0164] 在某些实施方式中,通过命令行选项,仅使得几个配置参数可用。这些包括最常用且最重要的选项。其充当用于终端用户的有用的语义文档,该终端用户通过特定于平台的应用或实用工具来检查节点的运行过程,诸如 WindowsTask Manager(Microsoft **Windows**®)、Activity Monitor(Mac OS **X**®) 和 psortop(**UNIX**®变体),其以显示应用的命令行的模式为特征。

[0165] 可配置参数的完整集合的示例如下。用正则表达式来描述某些配置模式,特别是用以解释可选或重复元素。

[0166] • 网络传输软件标识符。实例的网络传输软件标识符可以包括以下参数。

[0167] 命令行 `--myId = (host:)? port`

[0168] XML 元素 `<myId>(host:)? port</myId>`

[0169] 系统性质 `:com.miosoft.mioplexer.myId = (host:)? port`

[0170] Default:`<autodetected DNS hostname, 13697>`

[0171] ◦ host 是节点的 DNS 主机名且 port 是在范围 [0,65535] 内的可用 TCP 端口号。host 是可选的且默认为自动检测主机名。其可以通过查询操作系统来确定,如果未指定的话。如果此自动检测过程未能确定用于节点的唯一主机名,则选择主机名“本地主机”。未能正确地建立网络传输软件标识符可导致示例的不可到达性。

[0172] • 问候端口号。实例的问候端口号可以包括以下参数。这是网络传输软件自动发现反射所使用的 UDP 端口号。

[0173] 命令行 `--greeterPort = port`

[0174] XML 元素 `<greeterPort>port</greeterPort>`

[0175] 系统性质 `:com.miosoft.mioplexer.greeting.greeterPort = port`

[0176] 默认:网络传输软件标识符的 TCP 端口号

[0177] ◦ port 是在范围 [0,65535] 内的可用 UDP 端口号。未能正确地建立问候端口号可导致实例不能参与网络传输软件自动发现机制。

[0178] • 问候目标。自动发现进程将尝试联系< DNA 主机名、UDP 端口号>的完整集合。可能必须明确地指定这些以确保被防火墙分隔的节点能够进行通信。

[0179] 命令行 `--greeterTargets = (host:)? port(, (host:)? port)*`

[0180] XML 元素 `<greeterTargets>(<greeterTarget>(host:)? port</greeterTarget>)*</greeterTargets>`

[0181] 系统性质 `:com.miosoft.mioplexer.greeting.greeterTargets = (host:)? port(, (host:)? port)*`

[0182] - 默认:所有对的集合<广播地址、问候端口号>,其中,广播地址是节点的网络适配器中的一个的广播地址。

[0183] ◦ host 是节点的 DNS 主机名且 port 是在范围 [0,65535] 内的 TCP 端口号。host 是可选的且默认为自动检测主机名。其可以通过查询操作系统来确定,如果未指定的话。如果此自动检测程序未能确定用于节点的唯一主机名,则选择主机名“本地主机”。未能正确地建立此列表可导致意外且罕见的网状拓扑结构。

[0184] • 问候心跳。问候心跳是网络传输软件用来向所有问好目标发送请求问候消息的频率的平均水平。以毫秒为单位来指定参数。

[0185] XML 元素 :<greeterHeartbeatMillis>rate</greeterHeartbeatMillis>

[0186] 系统性质 :com.miosoft.mioplexer.greeting.greeterHeartbeatMillis = rate

[0187] ◦ 网络传输软件将以每几额定毫秒一次的频率向所有问候目标发送请求问候消息。

[0188] • 活动性探测速率。此速率是用来跨所建立的 TCP 连接发送活动性消息的频率的倒数。以毫秒为单位来指定参数。

[0189] XML 元素 :<livenessProbeRateMillis>rate</livenessProbeRateMillis>

[0190] 系统性质 :com.miosoft.mioplexer.routing.livenessProbeRateMillis = rate

[0191] 默认 :30,000

[0192] ◦ 网络传输软件将以每几额定毫秒一次的频率向每个所建立的 TCP 连接发送活动性消息,无论是客户端还是邻居。可将活动性探测速率设置成低的以减少网络业务,或者设置成高的以快速地检测低业务连接上的故障。

[0193] • 路由推迟量子。量子推迟路由任务,诸如路由表构造和近邻快照传播。以毫秒为单位来指定参数。此量子在发生将促使延迟计算产生不同的答复的更新时被更新。这允许延迟的递增延长。

[0194] XML 元素 :<routingPostponementMillis>quantum</routingPostponementMillis>

[0195] 系统性质 :com.miosoft.mioplexer.routing.postponementMillis = quantum

[0196] 默认 :5

[0197] ◦ 量子是将延迟路由任务的以毫秒为单位的时间量。未能明智地设置路由推迟量子可导致不良性能。

[0198] • 重传速率。用来重传网络间传输软件控制消息的频率的平均水平。以毫秒为单位来指定参数。

[0199] XML 元素 :<retransmissionRateMillis>rate</retransmissionRateMillis>

[0200] 系统性质 :com.miosoft.mioplexer.services.retransmissionRateMillis = rate

[0201] 默认 :5,000

[0202] ◦ 网络传输软件将以每几额定毫秒一次的频率重传在重传列表上的消息。未能明智地设置重传速率将导致增加的网络业务或用于服务请求的增加的等待时间。

[0203] • 服务重建宽限时段。此时段是在网络传输软件应发送报告关闭的服务句柄状态的服务句柄通知或节点服务句柄通知消息之前在节点上的网络传输软件启动之后必须流逝的时间量。以毫秒为单位指定。

[0204] XML 元素 :<gracePeriodMillis>quantum</gracePeriodMillis>

[0205] 系统性质 :com.miosoft.mioplexer.services.gracePeriodMillis = quantum

[0206] 默认 :30,000

[0207] ◦ 网络传输软件将把受影响的通知的传输延迟量子毫秒。未能明智地设置服务重建宽限时段将在网络传输软件实例循环时导致通信中的增加的中断或增加的等待时间。

[0208] • 登记推迟量子。该量子与登记任务的推迟有关,诸如服务目录快照传播。以毫秒为单位来指定参数。此量子在发生将促使延迟计算产生不同的答复的更新时被更新。这允许延迟的递增延长。

[0209] XML 元素 :`<registrarPostponementMillis>quantum</registrarPostponementMillis>`

[0210] 系统性质 :`com.miosoft.mioplexer.services.postponementMillis = quantum`

[0211] 默认 :5

[0212] ◦ 量子是将延迟登记任务的以毫秒为单位的时间量。未能明智地设置路由推迟量子可导致不良性能。

[0213] • 收割机时段。此时段是收割机任务执行的频率的倒数。以毫秒为单位来指定参数。

[0214] XML 元素 :`<reaperPeriodMillis>rate</reaperPeriodMillis>`

[0215] 系统性质 :`com.miosoft.mioplexer.services.reaperPeriodMillis = rate`

[0216] 默认 :10,800,000

[0217] ◦ 收割机任务将以每几额定毫秒一次的频率执行。可以将收割机时段设置成防止第二层预订会话或过度存储器生长的退回。

[0218] • 缓冲阈值。该阈值设置网络传输软件在丢弃合格消息之前缓存的字节的大约数目。单个消息或缓冲可越过此阈值且是以任意量。以字节为单位来指定该参数。

[0219] 命令行 :`--bufferThreshold = threshold`

[0220] XML 元素 :`<bufferThreshold>threshold</bufferThreshold>`

[0221] 默认 :200,000,000

[0222] ◦ 网络传输软件将缓存消息和缓冲的阈值字节加单个消息或缓冲。未能明智地设置缓冲阈值可导致不良性能。

[0223] • 线程池尺寸。此尺寸指定将被分配给每个网络传输软件的线程池的线程的最大数目。

[0224] XML 元素 :`<threadPoolSize>size</threadPoolSize>`

[0225] - 默认 :处理器核的数目的两倍。

[0226] ◦ 网络传输软件将用至多此许多操作系统核可调度的线程来填充每个线程池。未能明智地设置线程池尺寸可导致不良性能。

[0227] 在启动期间,网络传输软件向其标准输出写入信息性宣布,如果有的话。此宣布可以包括构建版本、优选服务器协议版本、所支持的服务器协议版本、所支持的客户端协议版本、与宣布的生成密切相关的详细时间戳以及版权标记。可访问此宣布的终端用户能够在解决问题时容易地确定开发者和支持人员所需种类的许多重要事实。

[0228] 网络传输软件是在没有特殊关闭要求的情况下设计和实现的。具有对网络传输软件的进程的逻辑访问的终端用户可使用平台的工具来终止该进程。网络传输软件不要求清洁的关闭过程,因此这是停止实例的可接受手段。节点能够在没有用于参与网格的其他节点或用于实例的替换化身的任何例外后果的情况下完全关闭或崩溃。

[0229] 在许多环境中,网格管理员可能并非可访问参与网格的所有节点或实例的进程。为了在实际上执行整个网格的管理,网格管理员可使用管理客户端来停止或重启节点上的

网络传输软件。为了停止网络传输软件，客户端向其本地网络传输软件发送请求关闭消息。此消息封装新会话标识符、目标网络传输软件的网络传输软件标识符、目标在退出之前应延迟的时间量（以毫秒为单位）以及操作系统进程应用来退出的状态代码。

[0230] 当节点接收到请求关闭消息时，其创建可路由的关闭消息并使用与针对第二层预订控制消息相同的机制可靠地将其发送到目的地。此消息包含相同的目的地网络传输软件标识符、超时以及状态代码，加上其自己的网络传输软件标识符和新的会话标识符。只有在接收到包含此会话标识符的节点确认消息时，网络传输软件才借助于包含原始会话标识符和 ok 的确认代码的客户端确认消息来确认发起客户端。

[0231] 当网络传输软件接收到关闭消息时，其立即用包含相同的会话标识符和 ok 的确认代码的节点确认消息进行答复。其然后延迟指定时间量。最后，网络传输软件用所发送的状态代码退出操作系统进程。

[0232] 为了重启节点上的网络传输软件，客户端向其本地节点发送请求重启消息。此消息封装新会话标识符、目标网络传输软件的网络传输软件标识符、目标在重启之前应延迟的时间量（以毫秒为单位）以及可选的替换网络传输软件二进制码。

[0233] 当节点接收到请求重启消息时，其创建可路由的重启消息并将其可靠地发送到目的地。此消息包含相同的目的地网络传输软件标识符、超时以及替换二进制码，加上其自己的网络传输软件标识符和新的会话标识符。当其最后接收到包含此会话标识符的节点确认消息时，其用包含原始会话标识符和 ok 的确认代码的客户端确认消息向客户端答复。

[0234] 当网络传输软件接收到重启消息时，其立即用包含相同的会话标识符和 ok 的确认代码的节点确认消息进行答复。其然后延迟指定量子。一旦该量子期满，则网络传输软件准备重启。如果并未指定替换网络传输软件二进制码，则网络传输软件启动特殊网络传输软件重启器应用并退出。网络传输软件重启器延迟直至其父进程已终止为止。其然后启动网络传输软件并最后退出。

[0235] 如果已经指定替换网络传输软件二进制码，则网络传输软件实例安全地将其写入临时文件。网络传输软件实例然后启动网络传输软件重启器，指定替换网络传输软件二进制码的位置。网络传输软件现在退出。网络传输软件重启器延迟直至其父进程已终止为止。其然后用临时文件的内容来覆写原始网络传输软件二进制码。最后，其启动新的网络传输软件二进制码并退出。网络传输软件和重启器在二进制码中被绑定在一起，因此重启器本身被同时地更新，并提供两个应用之间的语义兼容性。经良好管理客户端的促进，网络管理员因此可实现单个节点或整个网络的容易升级。

[0236] 对关闭或重启消息进行答复的节点确认消息在途中丢失是有可能的。当目标节点由于已退出而变得从客户端的网络传输软件不可到达时，客户端的网络传输软件取消负责可靠地发送关闭或重启消息的重传任务。在没有此预防措施的情况下，节点上的新启动的网络传输软件可能接收到意图用于其先前实例的关闭或重启消息，并不适当地退出。此错误可以通过实例的许多迭代而级联，只要竞态条件继续以相同方式解决其本身即可。

[0237] 相对于用户访问、诊断以及记录，网络传输软件本质上作为守护进程来运行。虽然进程可控制终端会话，例如当从平台命令行启动网络传输软件时，此进程不向程序供应输入。此类会话被用来向用户显示信息，诸如宣布、高优先级信息性消息以及在发生值得注意的异常条件时得到的堆栈跟踪。

[0238] 某些实施方式使用提供有 Java 运行时环境 (JRE) 以提供终端用户可自定义记录的 Java 记录 API。此框架允许可使用壳或台式计算机对节点上的网络传输软件进行逻辑访问的终端用户判定要使用哪些途径 (终端、文件系统、套接字等) 和如何用其本征优先级对消息进行过滤。在某些实施方式中, 可使用以下 Java 系统性质来设置用于各种网络传输软件子系统的记录优先级过滤器:

[0239] • `com.miosoft.io.Coordinator.level`。这设置 I/O 和缓冲管理子系统的冗长。这在记录优先级过滤器被设置为低于推荐值时可能是噪声严重的, 因为其提供与连接维护和消息业务相关联的丰富调试信息。此附加信息的生成和输出可使性能下降。推荐值是 INFO。

[0240] • `com.miosoft.mioplexer.Mioplexer.level`。这确定是否将记录伪造或未识别的消息。推荐值是 WARNING。

[0241] • `com.miosoft.mioplexer.MioplexerConfiguration.level`。这设置配置处理器的冗长。如此, 其提供关于可配置参数的通知, 诸如其最终值和尝试对其解析或获得默认值时遇到的问题。推荐值是 WARNING。

[0242] • `com.miosoft.mioplexer.greeting.Greeter.level`。这设置自动发现反射的冗长。当记录优先级过滤器被设置为非常低时, 这可能是略微有噪声的, 因为其提供关于请求问候和问候消息的传输的调试信息。推荐值是 WARNING。

[0243] • `com.miosoft.mioplexer.routing.Router.level`。这设置路由器的冗长。这可能是周期性地有噪声的, 特别是当网络正在经历通量时, 但一般地是静寂的。推荐值是 INFO; 其冲击性能与报告之间的良好平衡。

[0244] • `com.miosoft.mioplexer.services.Registrar.level`。这设置登记的冗长。这可能是周期性地有噪声的, 特别是当网络正在经历蜂拥的客户端活动时, 但一般地是静寂的。推荐值是 INFO。基于此设置, 在接收时记录最令人感兴趣的信息, 诸如开放服务句柄、关闭服务句柄、请求重启以及请求关闭。

[0245] 日志使得网络管理员能够被动地监视网络健康并执行此后的调查。有时针对运行系统运行活动查询是有价值的。例如, 希望检查运行的网络传输软件实例的内部状态的客户端可发送适合于其特定兴趣集的请求诊断消息。此消息包括新会话标识符、目的地节点的网络传输软件标识符以及一组诊断请求标识符。每个诊断请求标识符唯一地指定特定类型的诊断信息, 并且应将该集合聚合地理解成表示事务完整兴趣集。

[0246] 当节点的网络传输软件接收到请求诊断消息时, 其向目的地网络传输软件发送节点请求诊断消息。此消息包括新的会话标识符、其创建者的网络传输软件标识符以及同一组的诊断请求标识符。网络传输软件关于第二层预订控制消息及关闭和重启消息使用相同的机制而可靠地将其发送。

[0247] 当节点接收到节点请求诊断消息时, 其检查该组诊断请求标识符并计算适当的诊断信息。可以在概念上提供的诊断的种类是相当宽泛的。在某些实施方式中, 仅在写入时指定并实现句柄。这些是:

[0248] • 构建版本。这是目标网络传输软件的当前构建版本。这帮助网络管理员保持所有软件是当前的。

[0249] • 近邻。这是目标网络传输软件的当前近邻, 被指定为一组网络传输软件标识符。

[0250] •可到达网络传输软件实例。这是从目标网络传输软件可到达的节点的完整集合。在健康环境中,一旦网络稳定下来,这应会聚成参与网络的节点的完整集合。

[0251] •近邻对。这是对目标网络传输软件已知的近邻对<源、邻居>的完整集合,其中,源是发起证明该关系的近邻快照的节点的网络传输软件标识符,并且邻居是源节点的近邻中的邻居的网络传输软件标识符。

[0252] •路由对。这是对目标网络传输软件已知的路由对<目标、下一跳>的完整集合,其中,目标是可到达节点的网络传输软件标识符,并且下一跳是业务应被路由到以到达目标网络传输软件的节点的网络传输软件标识符。

[0253] •本地服务目录。存在目标网络传输软件的本地服务提供,被指定为一组服务绑定。

[0254] •服务目录。这是对目标网络传输软件已知的服务提供的完整集合,被指定为一组服务绑定。

[0255] •开放服务句柄。这是向目标网络传输软件的客户端登记的开放服务的完整集合。

[0256] •活动服务句柄预订对。这是活动服务句柄预订对<订户、发布者>的完整集合,其中,订户是登记到目标网络传输软件的客户端的开放服务句柄,并且发布者是本地或远程的任何发布者。

[0257] •活动路由预订。这是路由预订的完整集合,被指定为向目标网络传输软件的客户端登记的一组开放服务句柄。

[0258] 在某些实施方式中,网络传输软件将能够提供用于更多变化的诊断的支持。具体地,网络传输软件可以能够报告所有可配置参数的值。另外,网络传输软件可以能够报告关于其节点的信息,类似于CPU、磁盘以及网络活动水平。一旦已计算了所有诊断,则网络传输软件将其封装成具有会话标识符的诊断信息,该会话标识符与在节点请求诊断消息内载送的那个匹配。该诊断消息还包括与其具体化的时间紧密地相对应的时间戳。当客户端的附着网络传输软件接收到诊断消息时,其从重传列表中去除复制的节点请求诊断消息,以便防止诊断信息到客户端的冗余传送(由于输入诊断消息与缓慢的输出节点请求诊断信息竞争)。网络传输软件然后提取诊断和时间戳并创建新的诊断消息,其包含此信息和客户端的原始会话标识符。最后,其将诊断消息传送至客户端。

[0259] 相对于确认代码,当客户端向其连接的网络传输软件实例发送服务控制消息时,诸如开放服务句柄消息或关闭服务句柄消息,该网络传输软件用客户端确认消息进行答复。当节点向另一节点发送第二层预订控制消息时,远程节点用节点确认消息可靠地进行答复。两种确认消息都包括描述尝试指定操作的结果的确认代码。由于所请求的操作通常是在没有错误的情况下完成的,所以此确认代码通常将是ok。可以有其他确认代码,并且有时是不良客户端行为的结果。

[0260] 下面列出确认代码的示例。括弧中的值是确认代码的数字表示,如出现在例如串行化确认消息中。意图列表是可引出传送确认代码的响应的消息。

[0261] •ok(0)。网络传输软件在未遇到任何异常情况的情况下满足指定请求。当接收到消息时可应用:

[0262] 开放服务句柄

[0263] 关闭服务句柄

- [0264] 绑定服务标识符
- [0265] 解除绑定服务标识符
- [0266] 服务句柄预订
- [0267] 服务句柄退订
- [0268] 节点服务句柄预订
- [0269] 节点服务句柄退订
- [0270] 节点请求服务句柄通知
- [0271] 路由预订
- [0272] 路由退订
- [0273] 请求重启
- [0274] 请求关闭
- [0275] 重启
- [0276] 关闭
- [0277] • `error_service_handle_allocated_by_another_node(-1)`。节点拒绝满足该请求,因为目标服务句柄是由不同节点分配的。
- [0278] 开放服务句柄
- [0279] • `error_service_handle_registered_to_another_client(-2)`。节点拒绝满足该请求,因为目标服务句柄被登记到不同的客户端。
- [0280] 开放服务句柄
- [0281] 关闭服务句柄
- [0282] 绑定服务标识符
- [0283] 解除绑定服务标识符
- [0284] 服务句柄预订
- [0285] 服务句柄退订
- [0286] 路由预订
- [0287] 路由退订
- [0288] • `error_service_handle_already_open(-3)`。节点拒绝满足该请求,因为目标服务句柄已经开放。
- [0289] 开放服务句柄
- [0290] • `error_service_handle_not_open(-4)`。节点拒绝满足该请求,因为目标服务句柄不是开放的。
- [0291] 关闭服务句柄
- [0292] 绑定服务标识符
- [0293] 解除绑定服务标识符
- [0294] 服务句柄预订
- [0295] 服务句柄退订
- [0296] 路由预订
- [0297] 路由退订
- [0298] • `error_service_binding_already_established(-5)`。节点拒绝满足该请求,因

为已建立目标服务绑定。

[0299] 绑定服务标识符

[0300] • `error_service_binding_not_established(-6)`。节点拒绝满足该请求,因为并未建立目标服务绑定。

[0301] 解除绑定服务标识符

[0302] • `error_service_handle_already_subscribed(-7)`。节点拒绝满足该请求,因为目标预订已存在。

[0303] 服务句柄预订

[0304] 路由预订

[0305] • `error_service_handle_not_subscribed(-8)`。节点拒绝满足该请求,因为目标服务句柄预订不存在。

[0306] 服务句柄退订

[0307] 路由退订

[0308] • `error_special_service_handle(-9)`。节点拒绝满足请求,因为嵌入服务句柄包含落在被预留用于内部使用的范围内的 UUID。此范围是 `[0x00000000000000000000000000000000, 0x000000000000000000000000000003E8]`,即前 1,000 个顺序 UUID。

[0309] 开放服务句柄

[0310] 服务句柄预订

[0311] 在某些实施方式中,需要检查在客户端确认消息内传送的确认代码以确保算法的正确性,并且应使用合理的编程实践。

[0312] 在这里所述的技术可以在大范围的领域中和大范围的应用中使用,例如要求在运行于网络的节点上的应用之间的非常大量通信路径或专用于建立并保持网络中的通信路径的相对低的开销量或两者的应用或网络。

[0313] 可以在计算机硬件、路由器、网关、布线、光纤及其他联网硬件、操作系统、应用软件、固件、联网、无线通信、用户接口等领域中的多种商业可获得平台上实现这里所述的给水。

[0314] 其他实施方式在所附权利要求的范围内。

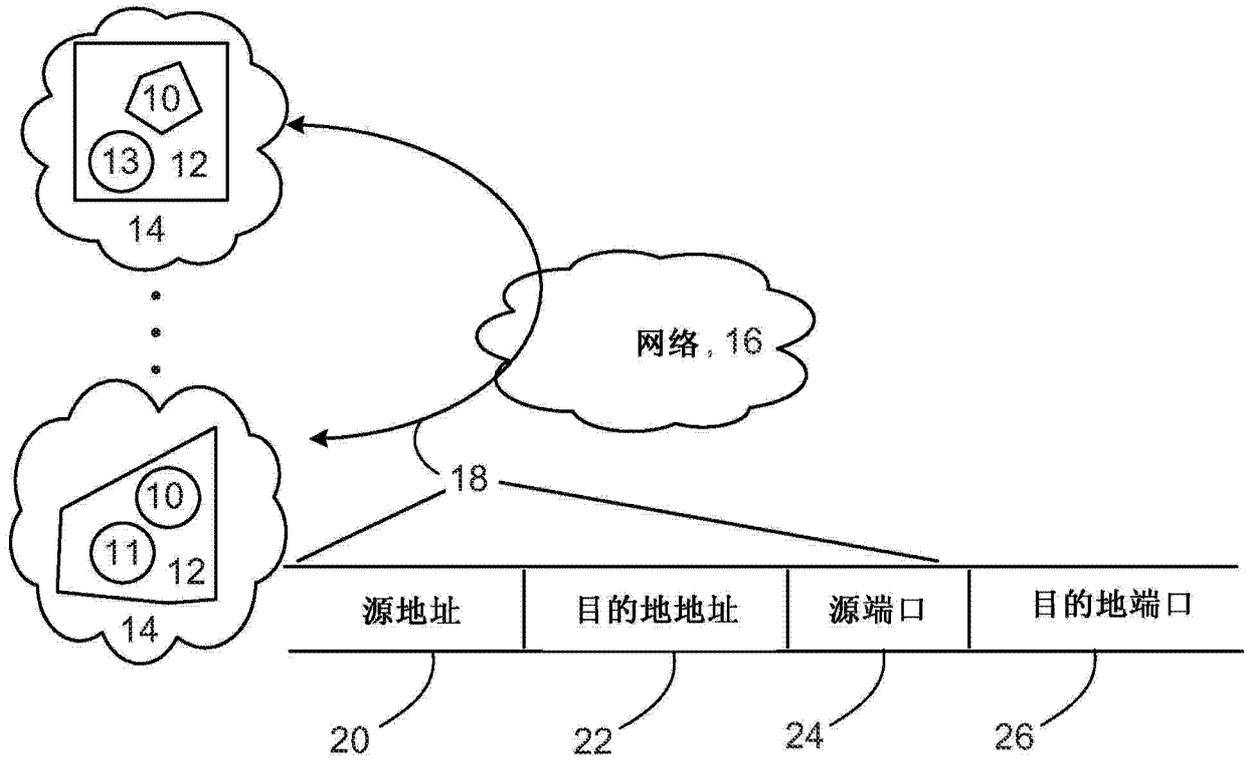


图 1

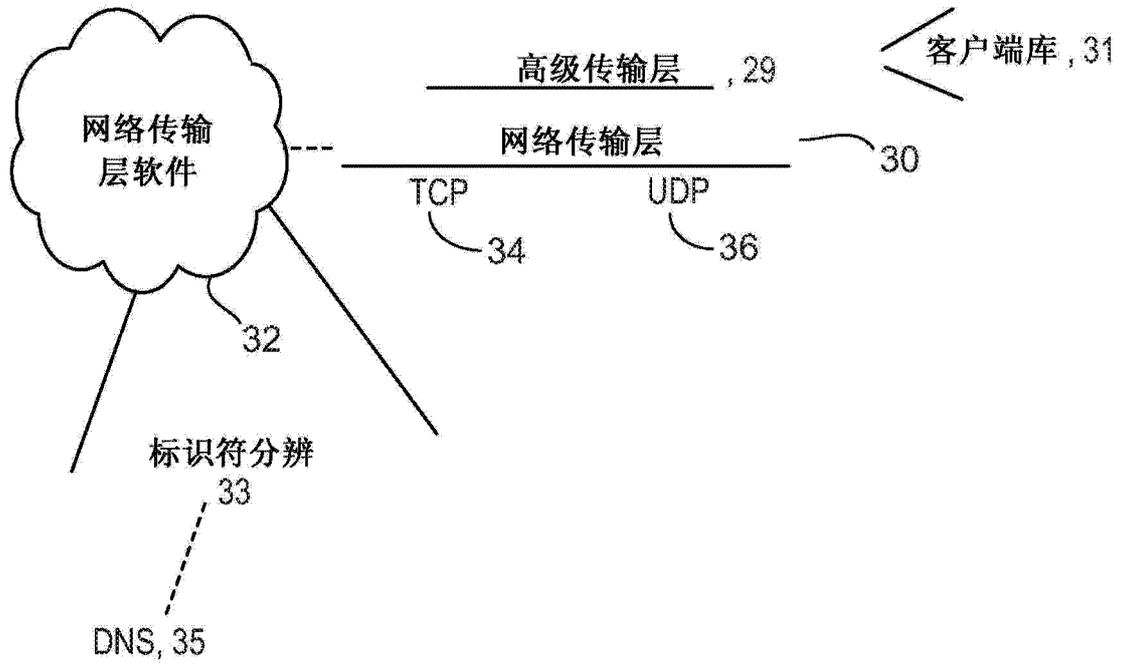


图 2

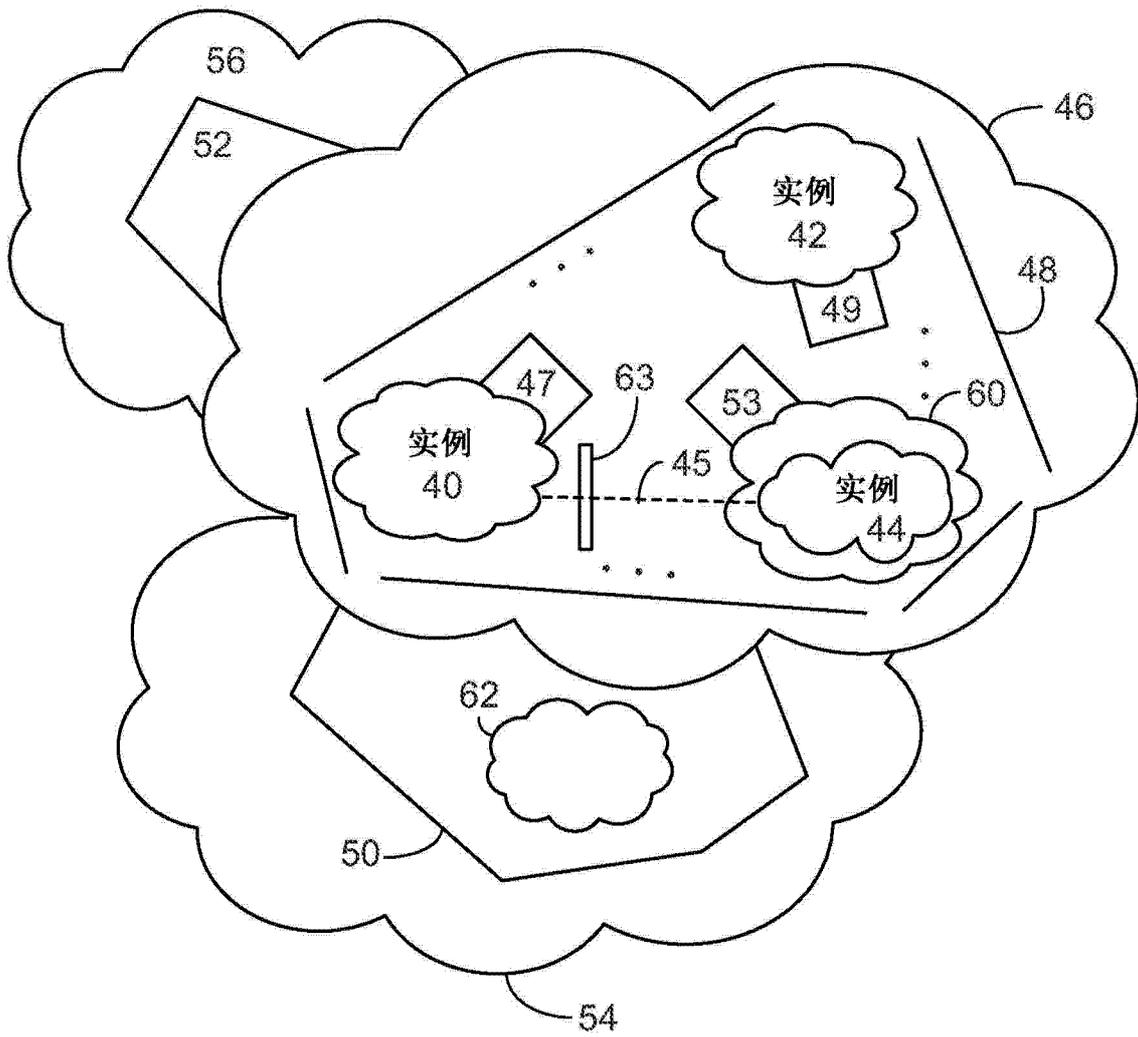


图 3

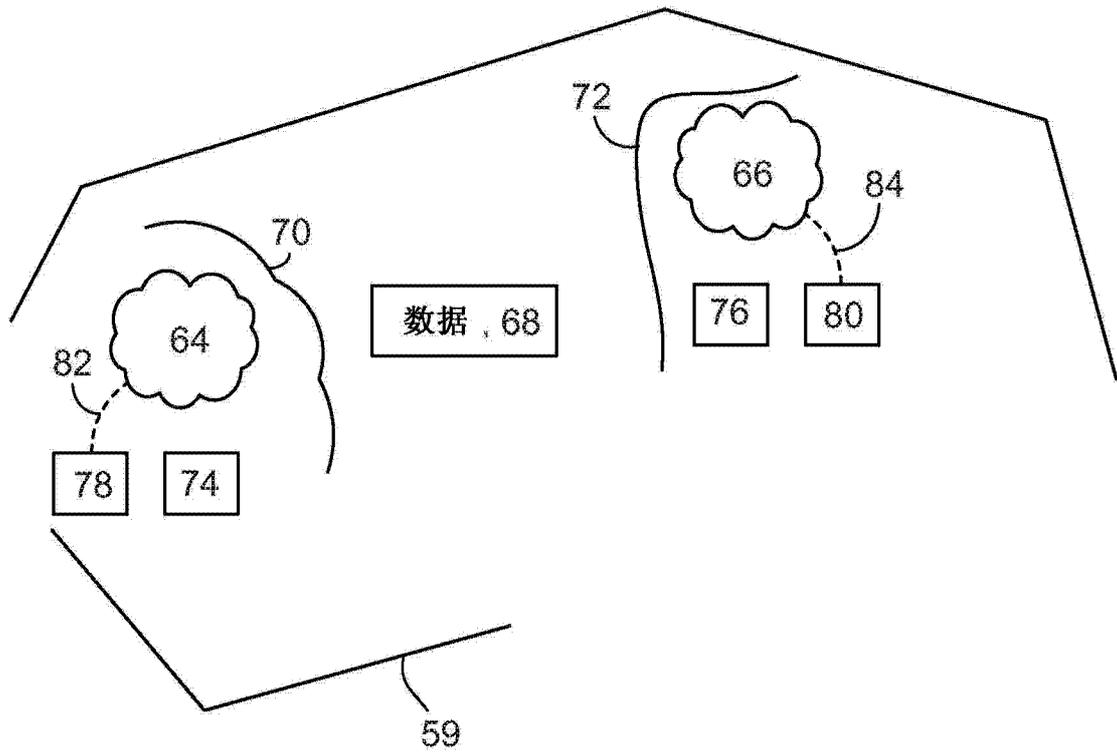


图 4

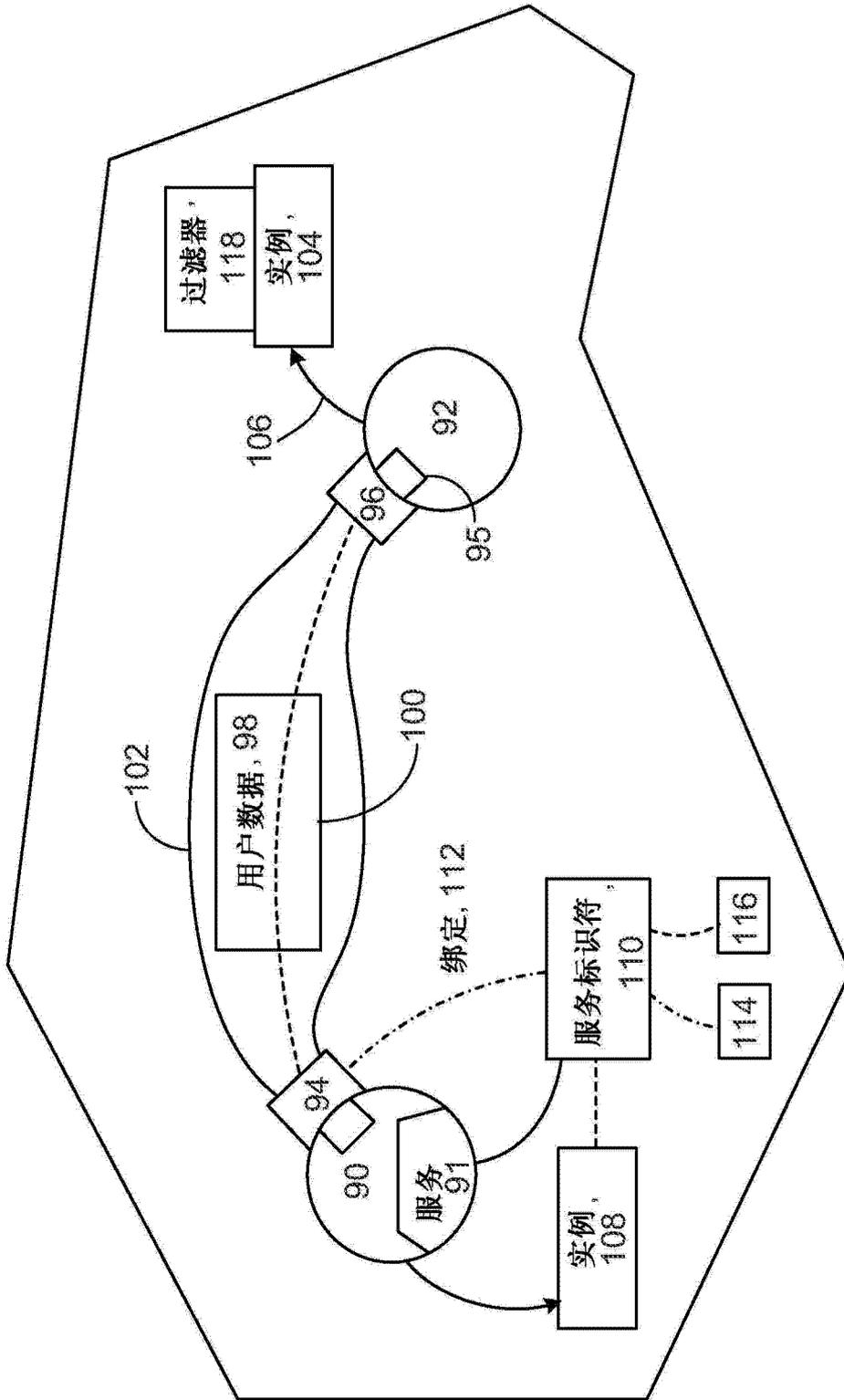


图 5

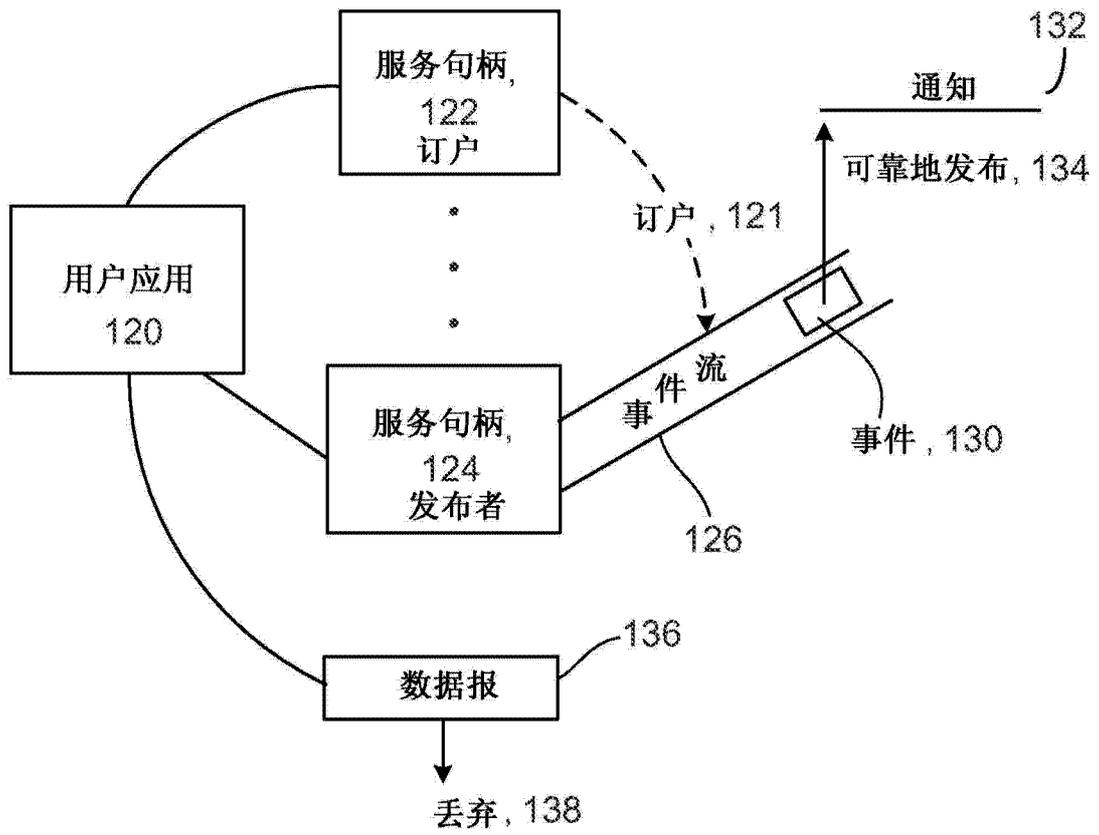


图 6

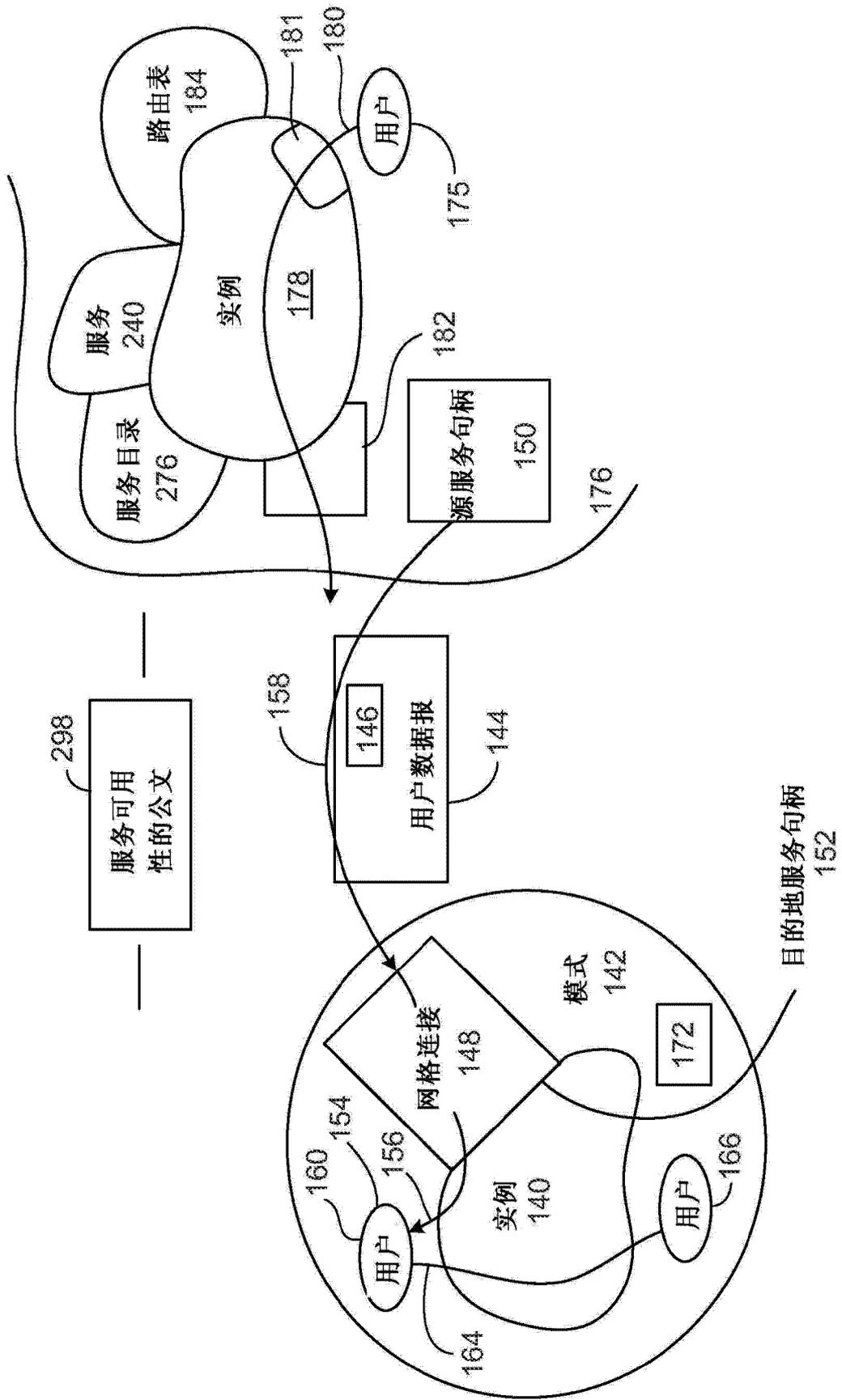


图 7

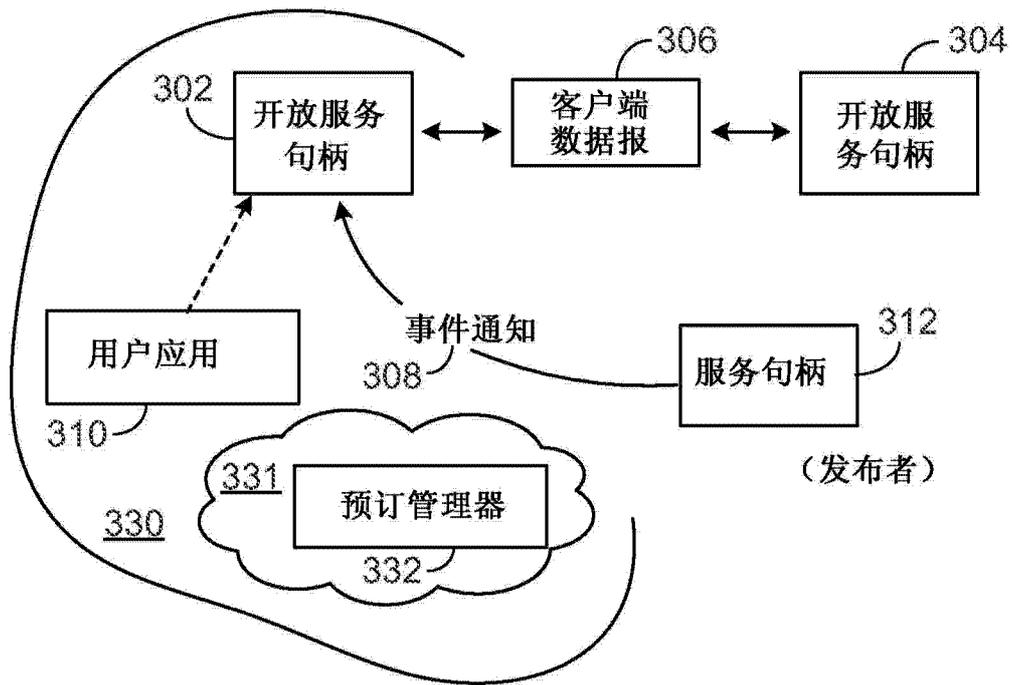


图 8

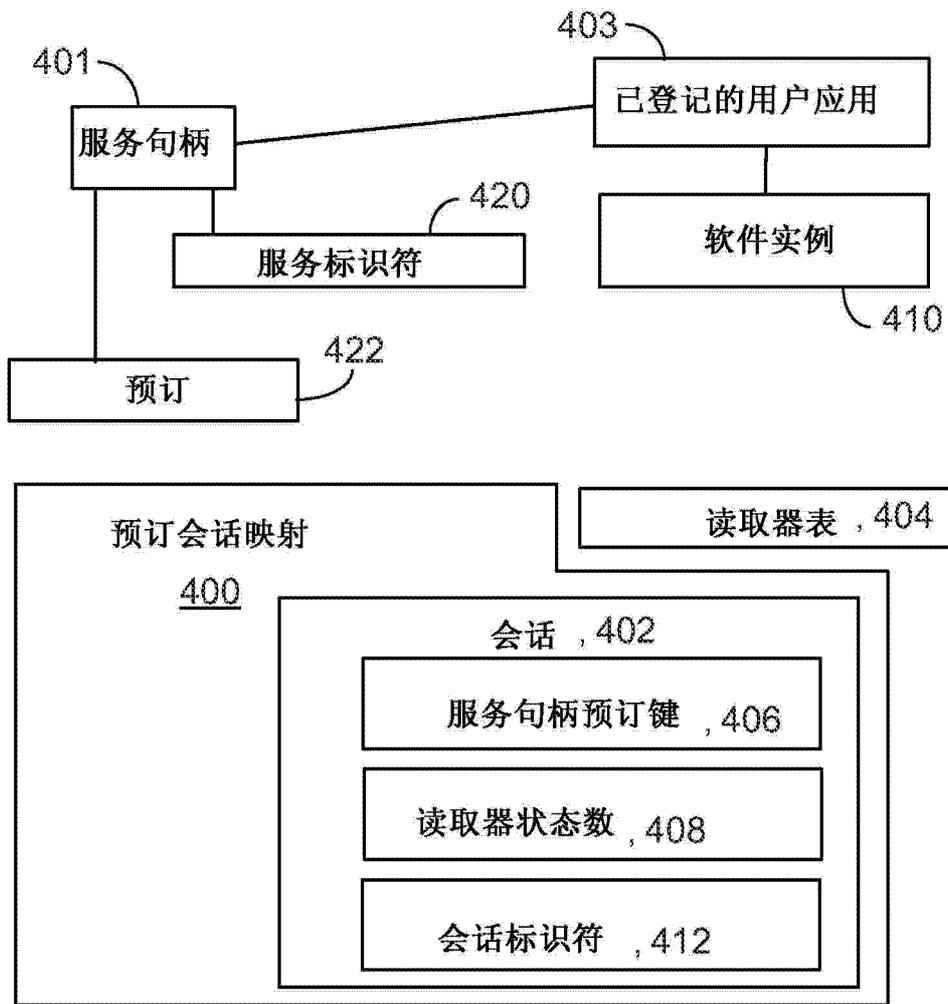


图 9

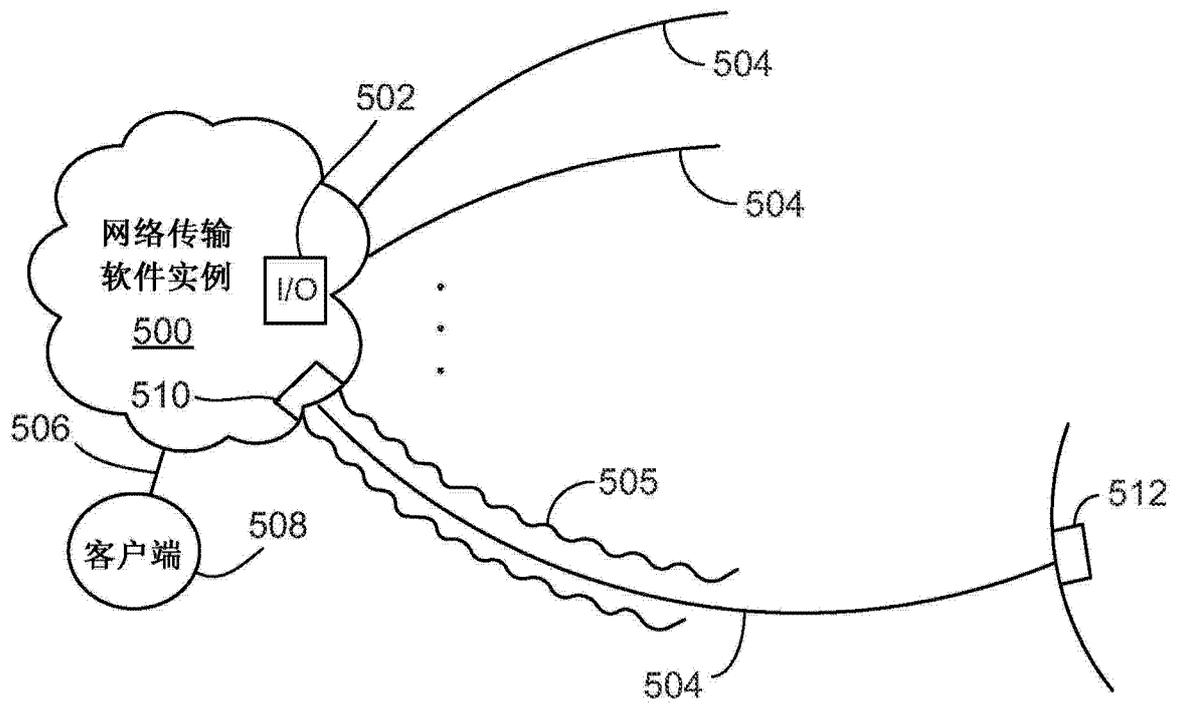


图 10

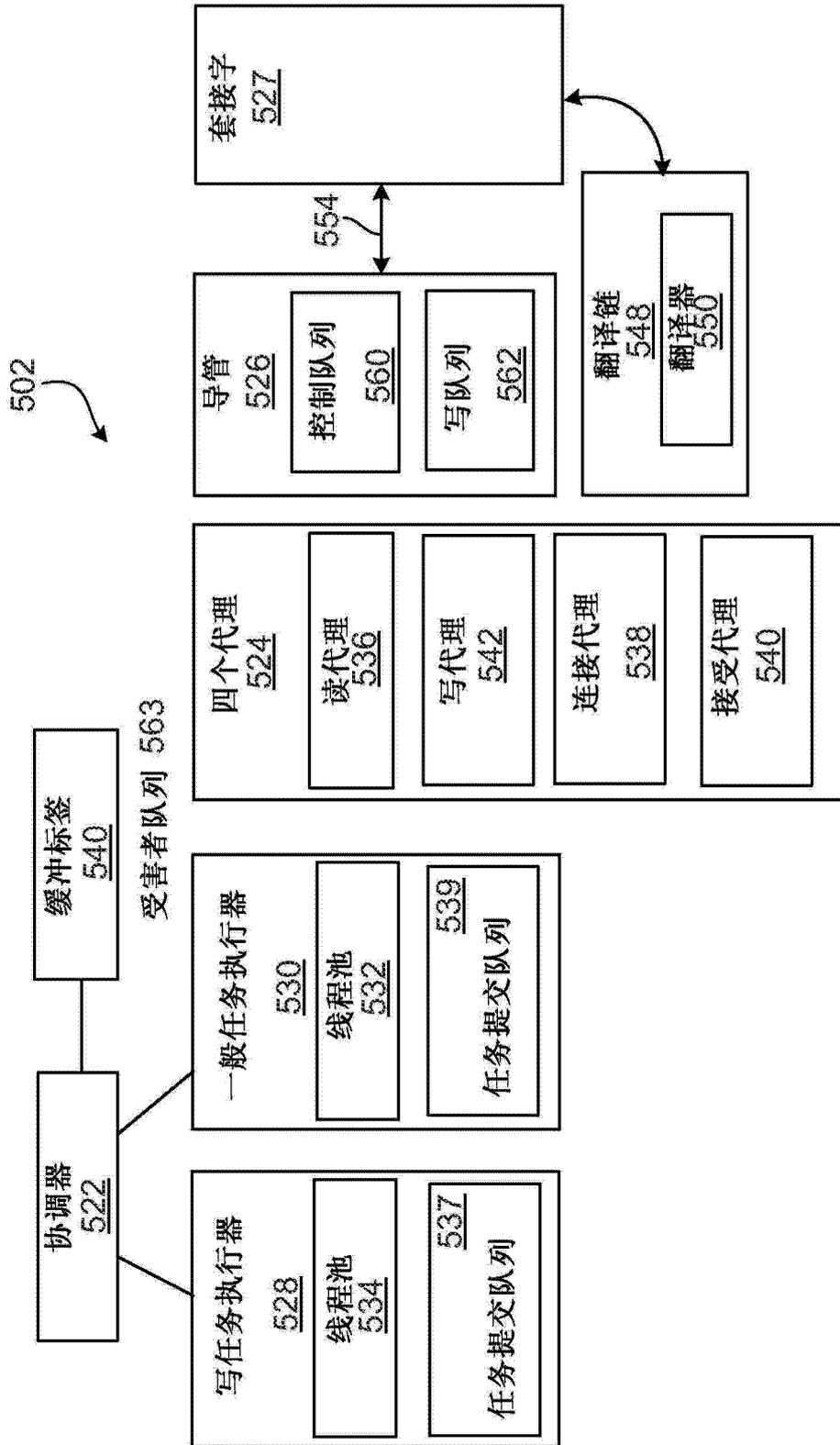


图 11

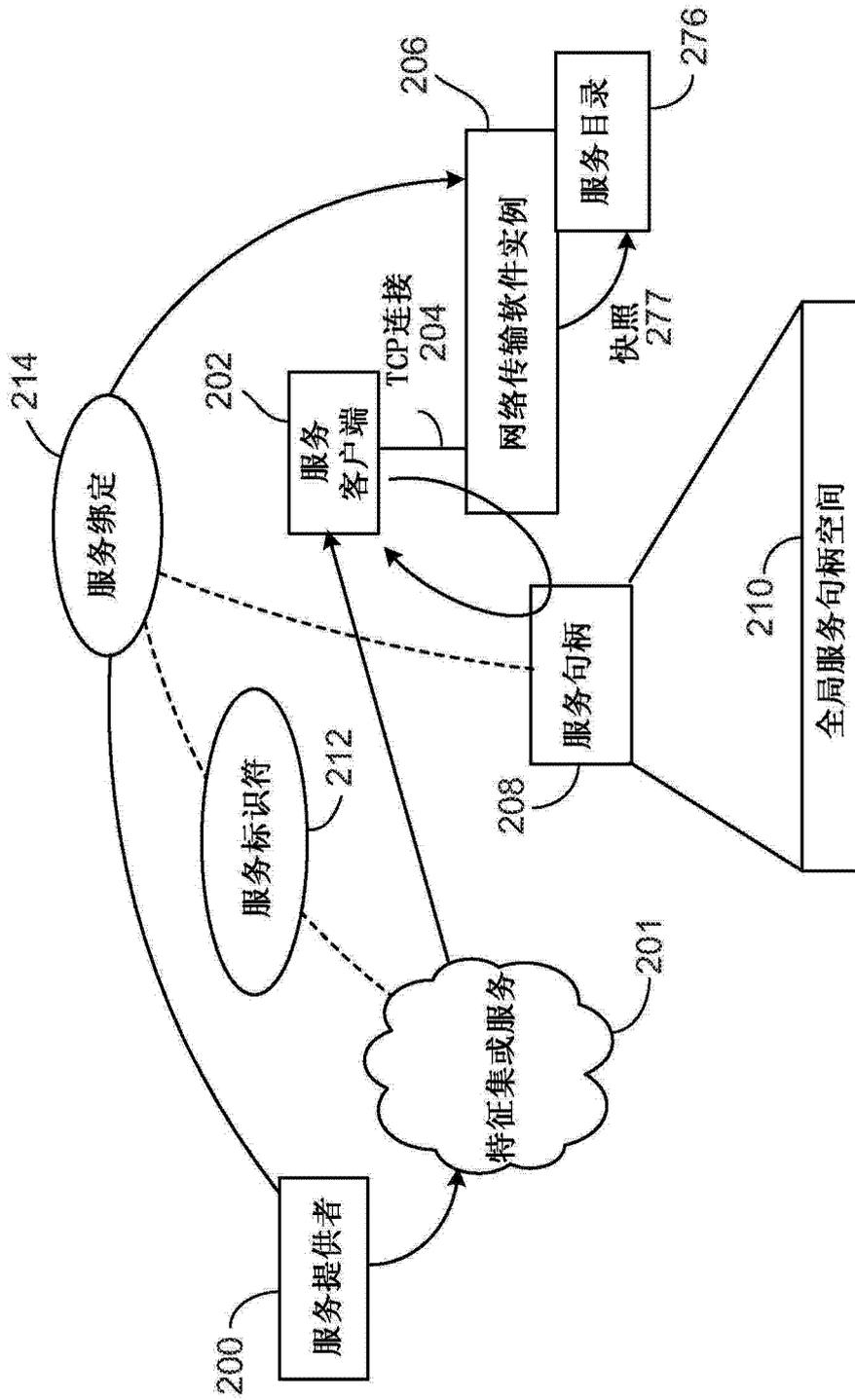


图 12