



(12) 发明专利

(10) 授权公告号 CN 1647482 B

(45) 授权公告日 2010.08.25

(21) 申请号 03807855.4

(22) 申请日 2003.03.21

(30) 优先权数据

10/118,203 2002.04.05 US

(85) PCT申请进入国家阶段日

2004.10.08

(86) PCT申请的申请数据

PCT/US2003/009134 2003.03.21

(87) PCT申请的公布数据

W02003/088621 EN 2003.10.23

(73) 专利权人 甲骨文国际公司

地址 美国加利福尼亚州

(72) 发明人 胡维民 胡安·洛艾萨

(74) 专利代理机构 北京康信知识产权代理有限

责任公司 11240

代理人 余刚

(51) Int. Cl.

H04L 29/06 (2006.01)

H04L 29/04 (2006.01)

(56) 对比文件

CN 1324555 A, 2001.11.28, 全文.

WO 98/26559 A1, 1998.06.18, 全文.

Tirthankar Lahiri, Amit Ganesh, Ron Weiss, Ashok Joshi. Fast-Start: Quick Fault Recovery in Oracle. 2001, 全文.

Oracle. Cluster-Aware ODBC and OCI Client Applications for Oracle Fail Safe Solutions. 2001, 全文.

Oracle. Oracle9i Application Server- The Best Application Server for the Oracle9i Database. 2002, 第 36-37 页.

Svend Frolund, Fernando Pedone, Jim Pruyne. Web e-transactions. 全文.

Microsoft Corporation. Planning for Reliability and High Availability. 2001, 全文.

审查员 冯玉学

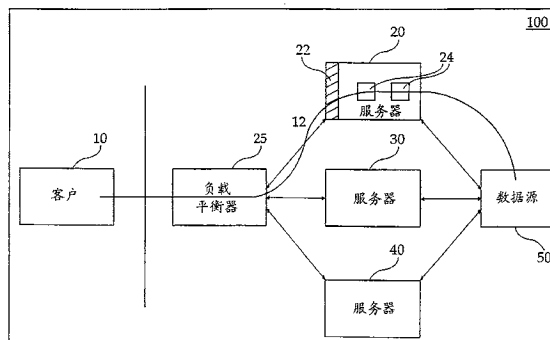
权利要求书 4 页 说明书 11 页 附图 7 页

(54) 发明名称

用于网络通信管理的系统和方法

(57) 摘要

本发明提供了一种对客户和数据源之间的重试通信进行管理的系统。所述系统包括多个逻辑部件,其按顺序结合以构成用于客户通信的通信路径。在所述通信路径中的所述部件中的至少一个被指定为重试部件。所述重试部件为所述通信路径中的所有部件重试失败的客户通信。



1. 一种用于对发送方和指定接收方之间的通信进行管理的系统,所述系统包括:
多个逻辑部件,其相互连接,以将发送方通信从所述发送方按顺序发送到所述指定接收方;

重试部件,为所述多个逻辑部件中的一个,如果所述发送方通信不能到达所述指定接收方,则所述重试部件被指定向所述指定接收方重试发送所述发送方通信;以及

多重部件,其在所述多个逻辑部件中,通过将所述发送方通信失败的指示发送到所述重试部件,来对所述发送方通信失败作出响应,而无需所述多重部件中的任何多重部件向所述指定接收方重试发送所述发送方通信。

2. 根据权利要求1所述的系统,其中,所述发送方为客户,所述接收方为数据源,以及所述发送方通信为对由所述数据源管理的数据的请求。

3. 根据权利要求1所述的系统,其中,所述重试部件在从所述发送方到所述指定接收方的通信路径上,位于所述多重部件之前。

4. 根据权利要求1所述的系统,其中,所述重试部件被设置为服务器的接口的一部分,所述接口被配置成对来自其他设备的请求进行处理。

5. 根据权利要求1所述的系统,其中,所述重试部件为服务器的逻辑服务器部件,其发送请求到所述指定接收方。

6. 根据权利要求1所述的系统,其中,所述重试部件位于负载平衡设备上,所述负载平衡设备从所述发送方接收所述发送方通信,并将所述发送方通信发送到服务器,所述服务器包括所述多个逻辑部件中的其他部件。

7. 根据权利要求1所述的系统,其中,至少所述多个逻辑部件中的多重部件位于第一服务器上,以及,其中所述重试部件被配置成使用第二服务器的部件,来重试将所述发送方通信发送到所述指定接收方。

8. 根据权利要求1所述的系统,其中,所述重试部件访问拓扑图,以确定哪个所述多重部件正确地运行。

9. 根据权利要求1所述的系统,还包括外部监视器,其位于所述多个逻辑部件的外部,所述外部监视器被配置成检测所述发送方通信是否没能到达所述指定接收方,并且当所述发送方通信未能到达所述指定接收方时,与所述重试部件通信。

10. 根据权利要求1所述的系统,其中,所述重试部件在重新发送所述发送方通信之前,确定所述发送方通信是否可以重试。

11. 根据权利要求1所述的系统,其中,如果所述多重逻辑部件中的一个或多个指出所述发送方通信未能到达所述指定接收方,所述重试部件是用于重试将所述发送方通信发送到所述指定接收方的惟一部件。

12. 根据权利要求10所述的系统,其中,所述重试部件至少部分地基于所述发送方通信的特征,确定所述发送方通信是否可以重试。

方的标识被指定为允许重新发送所述发送方通信,则重新发送所述发送方通信。

13. 根据权利要求10所述的系统,其中,所述重试部件和所述多重部件位于服务器上,以及其中,所述重试部件被配置成,如果任意所述多重部件指示所述发送方通信未能到达所述指定接收方,就与其他服务器联络以重新发送所述发送方通信。

14. 根据权利要求1所述的系统,其中,所述重试部件位于所述发送方上,并通过广域

网与所述多重部件通信。

15. 根据权利要求 10 所述的系统,其中,所述重试部件在确定所述发送方通信是否可以重试时,确定所述发送方通信是否安全。

16. 根据权利要求 10 所述的系统,其中,所述重试部件在确定所述发送方通信是否可以重试时,对所述发送方的标识进行确认。

17. 根据权利要求 10 所述的系统,其中,所述重试部件基于所述指定的接收方的标识来确定所述发送方通信是否可以重试。

18. 根据权利要求 1 所述的系统,其中,所述重试部件位于服务器内,以及其中,所述单个重试部件确定所述服务器中的至少一个其他的可选部件,并通过将所述发送方通信发送到所述至少一个其他的可选部件,来重试将所述发送方通信发送到所述指定接收方。

19. 根据权利要求 1 所述的系统,其中,在重试发送所述发送方通信之前,所述重试部件确定所述发送方通信之前试图重试的次数,然后仅当所确定次数小于最大重试阈值时,才重试发送所述发送方通信。

20. 根据权利要求 1 所述的系统,其中,在所述重试部件向所述指定接收方重新发送所述发送方通信之前,所述重试部件利用所述发送方正请求的资源的资源标识符来确定所述发送方通信是否是可重试的。

21. 根据权利要求 20 所述的系统,其中,所述重试部件利用统一资源定位符作为所述发送方正请求的资源的资源标识符。

22. 一种用于对网络通信进行管理的计算机实施方法,所述方法包括:

在发送方和指定接收方之间的第一通信路径上指定一个重试点,所述第一通信路径包括多个逻辑部件,其顺序地接收并将发送方通信从所述发送方发送到所述指定接收方,所述重试点构成所述第一通信路径的一部分;

如果所述发送方通信不能沿所述重试点之外的所述第一通信路径的一部分到达所述指定接收方,则所述多个逻辑部件中的一个逻辑部件向所述重试点发送关于所述发送方通信未能到达所述指定接收方的指示;然后

只从所述重试点重新发送所述发送方通信,以到达所述指定接收方。

23. 根据权利要求 22 所述的方法,其中,响应于检测到所述发送方通信未能到达所述指定接收方,而向所述重试点发送关于所述发送方通信未能到达所述指定接收方的指示。

24. 根据权利要求 22 所述的方法,其中,所述重试点的指定包括指定逻辑服务器部件以执行重试逻辑。

25. 根据权利要求 23 所述的方法,其中,检测所述发送方通信未能到达所述指定的接收方包括,检测所述指定接收方是不可用的。

26. 根据权利要求 23 所述的方法,其中,检测所述发送方通信未能到达所述指定的接收方包括,检测到所述第一通信路径中的逻辑部件失败。

27. 根据权利要求 22 所述的方法,还包括在检测到所述发送方通信失败后,选择新的发送方通信通信路径。

28. 根据权利要求 27 所述的方法,其中,选择新的通信路径包括,使用所述重试点处的逻辑来确定可用的通信路径,以到达所述指定接收方。

29. 根据权利要求 22 所述的方法,其中,指定所述重试点包括,选择构成所述第一通

信路径的所述逻辑部件的一部分作为所述重试点,所述逻辑部件部分包括至少两个逻辑部件。

30. 根据权利要求 29 所述的方法,其中,选择所述逻辑部件的一部分包括,从负载平衡设备选择所述逻辑部件部分,所述负载平衡设备在多个服务器之间分配负荷量,所述逻辑部件部分位于所述多个服务器中的一个上。

31. 根据权利要求 22 所述的方法,还包括确定所述发送方通信是否可以重试;以及只有在所述发送方通信可以重试的情况下才重新发送所述发送方通信。

32. 根据权利要求 31 所述的方法,其中,确定所述发送方通信是否可以重试包括,确定所述发送方的标识,以及如果所述发送方的标识被指定为允许重新发送所述发送方通信,则重新发送所述发送方通信。

33. 根据权利要求 31 所述的方法,其中,确定所述发送方通信是否可以重试包括,确定所述指定接收方的标识,以及如果所述指定接收方的标识被指定为允许重新发送所述发送方通信,则重新发送所述发送方通信。

34. 根据权利要求 31 所述的方法,其中,确定所述发送方通信是否可以重试包括,如果所述发送方通信是安全的,则确定所述发送方通信为不可重试。

35. 根据权利要求 31 所述的方法,其中,确定所述发送方通信是否可以重试包括,确定所述发送方通信之前试图重试的次数,然后,仅当所确定次数小于最大重试阈值时,才重试发送所述发送方通信。

36. 根据权利要求 31 所述的方法,其中,确定所述发送方通信是否可以重试包括,将所述发送方通信的特征与指示所述发送方通信是否可以重试的相应指示进行参考。

37. 根据权利要求 31 所述的方法,其中,在所述发送方和所述指定接收方之间的所述第一通信路径上指定所述重试点包括,选定与所述发送方相对应的逻辑服务器部件作为所述重试点。

38. 根据权利要求 31 所述的方法,其中,重新发送所述发送方通信以访问所述指定接收方包括使用不同的通信路径将所述发送方通信重新发送到所述指定接收方。

39. 根据权利要求 38 所述的方法,其中,使用所述不同的通信路径包括,访问位于所述第一通信路径之外的通信设备,以将所述发送方通信发送到所述指定接收方。

40. 根据权利要求 31 所述的方法,其中,向所述重试点发送关于所述发送方通信未能到达所述指定接收方的指示包括,在所述重试点和所述指定接收方之间的多个逻辑部件中的一个上,检测所述发送方通信是否未能到达所述指定接收方。

41. 根据权利要求 31 所述的方法,其中,仅从所述重试点重新发送所述发送方通信到所述指定接收方包括,选择一个新的通信路径,其至少部分地位于在所述重试点的对应设备之外的设备之上。

42. 根据权利要求 41 所述的方法,还包括为了确定所述发送方通信是否可以重试,确定所述新的通信路径是否要被使用。

43. 根据权利要求 42 所述的方法,还包括确定所述发送方通信是否安全,以及如果所述发送方通信是不安全的,则确定所述新的通信路径要被用来重新发送所述发送方通信。

44. 根据权利要求 31 所述的方法,其中,确定所述发送方通信是否可以重试包括,确定所述发送方通信是否安全,并且,确定是否重新发送所述发送方通信到所述指定接收方包

括,使用在所述第一通信路径外部的通信设备。

45. 一种用于对发送方和指定接收方之间的通信进行管理的系统,所述系统在包括多个逻辑部件的设备上得以实施,所述多个逻辑部件按顺序结合以将发送方通信发送到所述指定接收方,所述系统包括:

46. 根据权利要求 45 所述的系统,其中,所述存储在存储器中的数据基于所述发送方通信是否安全,来指示所述发送方通信是否可以重试。

47. 根据权利要求 45 所述的系统,其中,所述存储在存储器中的数据基于所述发送方通信的操作,来指示所述发送方通信是否可以重试。

48. 根据权利要求 45 所述的系统,其中,所述重试部件被配置为所述多个逻辑部件中唯一的部件,来重试未能到达所述指定接收方的发送方通信。

49. 根据权利要求 48 所述的系统,其中,所述重试部件被配置成从一组部件中选择部件组合,以作为传输路径来重试发送所述通信,所述一组部件包括所述设备的多个逻辑部件,其中,所述一组部件中的至少一些部件位于所述设备外部。

用于网络通信管理的系统和方法

技术领域

[0001] 本发明涉及对通过网络的通信进行管理。特别地,本发明的实施例属于多层网络通信系统的重试技术。

[0002] 背景技术

[0003] 处理客户-服务器型通信的网络应用变得越来越复杂。一种典型的客户-服务器通信涉及客户向服务器发送请求、以及服务器返回响应给客户。服务器通常由多个处理客户请求的逻辑部件组成。在响应于客户请求产生响应之前,通常由服务器的多个逻辑部件以某种方式对客户请求进行处理。

[0004] 在典型的应用中,服务器访问数据源,以响应于客户请求检索客户所需信息。在将客户请求发送到指定数据源或另一部件之前,服务器的每一个逻辑部件都针对客户请求行使职能。例如,服务器部件之一可能为 JAVA 程序,其执行某些运算、为选定数据查询数据源、执行更多运算,然后将结果返回给客户。服务器的其他部件可能按顺序对通信进行处理,以执行附加的运算或进行更多的数据库查询。

[0005] 一些服务器部件还可处理对客户请求的响应。例如,除了将客户请求发送到数据源以外,服务器部件还可将来自数据源的响应发回给客户。该部件还可执行运算,和/或响应于接收针对客户请求的响应而进行附加的查询。

[0006] 在一些结构中,处理客户通信的设备包括负载平衡设备和服务器。该服务器由负载平衡设备从一组服务器中选出。选定的服务器和负载平衡器被用来在客户和诸如数据库的数据源之间交换信息。服务器将客户通信从负载平衡设备发送到指定数据库。该数据库可相当于网络服务器或其他机器,其提供对关于客户通信的信息的访问。

[0007] 对客户通信进行处理的多个设备相结合提供了多个逻辑部件,其在客户通信被发送到数据库之前对该客户通信进行处理。例如,可以在服务器内把逻辑部件分层,以便对用于处理客户通信的过程进行分配。这些分层的部件以顺序的方式对客户通信进行处理。其结果是,该客户通信在被指定数据库接收之前,由位于不同服务器上的多个部件按顺序对其进行处理。

[0008] 客户通信经常不能到达其目的地。其不能到达的原因可能是因为在,将客户通信发信号通知给目的地时,目的地无法访问。其原因还可能是因为在,对客户通信进行处理的部件之一发生故障。

[0009] 许多客户-服务器系统中的部件被配置成在检测到失败的地方重试失败的客户通信。在多层结构中,失败检测传播到单独处理客户通信的所有逻辑部件。其结果是,在同一服务器中的多个部件分别尝试重试失败的客户通信。而且,在失败被报告回客户之前,这种多层结构使得一些部件多次重试发送失败的客户通信。大部分这些额外的重试失败的客户通信的尝试都是没有用处的。

[0010] 例如,如图 8 所示,假设服务器提供一个通信路径,其包括第一应用程序 830、第二应用程序 840、和服务器应用程序 820。该通信路径用于将客户 810 与目的地连接。在这个实例中,目的地为数据源 850。传统的不具有指定重试部件的系统将使用第二应用程序

840、第一应用程序 830、和服务器应用程序 820 来反复重试失败的客户通信。例如，如果在第二应用程序 840 试图联系数据源 850 时发生失败，那么传统的系统将使第二应用程序 840 重试客户通信，然后将失败指示给第一应用程序 830。然后，该第一应用程序重试失败的客户通信。如果数据源 850 对第二应用程序 840 仍然不可用，那么在第一应用程序 830 进行重试之后由第二应用程序 840 进行另一次重试。在第二应用程序 840 重试两次、第一应用程序 830 重试一次之后，将失败通知给服务器部件 820。如果服务器部件 820 进行了重试，那么第二应用程序 840 可重试两次，并且第一应用程序 830 可再次重试。其结果是，在通知客户 810 失败的客户通信之前，该客户将等待较长的时间。

[0011] 在一些现有系统中，当在具有冗余部件的环境中发生失败，则该环境将对自身重新配置，使得后续请求被发送到另一未失败的环境。这样，重试请求通常将取得成功，但需要另一机器和附加的资源。例如，如果数据库服务器崩溃，该环境将对自身重新配置，使得后续请求到达另一个服务器上。可由负载平衡器来选择另一个服务器，这通常在初始服务器的内部部件失败之后进行。

[0012] 综上所述，无疑需要提供用于处理通信失败的技术，而避免了由通信路径中的多个部件为响应在通信路径中发生的失败进行重试操作而造成的浪费。

发明内容

[0013] 本发明提供了用于对客户和数据源之间的失败通信进行重试的技术。在一个实施例中，服务器构成了客户和数据源之间的通信路径的一部分，其被配置成包括位于单个内部逻辑部件上的重试逻辑。服务器内的其他逻辑部件将失败的客户通信报告给指定用来尝试重试的部件。

[0014] 根据一个实施例，重试通信可由重试逻辑所驻留的部件进行管理。该重试逻辑可被配置成选择性地确定某些通信为可以重试，并确定其他通信为不可重试。该重试逻辑还可与为失败的客户通信选择新的通信路径的逻辑相结合。

[0015] 其他实施例包括重试技术和对来自客户通信路径中的另一设备的重试通信的管理。例如，可以用负载平衡设备或客户来实施重试技术。

附图说明

[0016] 本发明以举例的方式、而不是以限定的方式，通过附图中的图来进行说明，其中，相同的附图标记指代相同的元件，其中：

[0017] 图 1 示出了一种包括用于在客户和数据源之间传送通信的指定重试部件的系统；

[0018] 图 2 所示的方框图示出了被配置成包括有重试逻辑的服务器的操作；

[0019] 图 3 示出了在客户和数据源之间的通信路径中实施重试逻辑的流程图；

[0020] 图 4 示出了一种实施服务器的重试逻辑的方法，其中所述服务器用来将客户与数据源相连接；

[0021] 图 5 示出了一种用于在将客户与数据源相连的系统上实施重试逻辑的方法；

[0022] 图 6 示出了一种数据结构，其能以重试逻辑来实施，以确定某种失败通信是否可以重试；

[0023] 图 7 所示的方框图示出了一种通过其实施本发明实施例的计算机系统；以及

[0024] 图 8 示出了一种用于重试针对服务器的失败的客户通信的现有技术系统。

具体实施方式

[0025] 本发明描述了一种用于访问数据源的方法和装置。在以下的描述中,为了进行说明,为提供对本发明的详尽理解而列出了许多具体细节。然而,可以知道,本发明无需这些具体细节也可得以实施。在其他情况下,为了避免对本发明的不必要的混淆,以方框图的形式示出了众所周知的结构和设备。

[0026] A. 总述

[0027] 根据一个实施例,为构成客户和数据源之间的通信路径的顺序部件指定一个重试点。该指定的重试部件被设置在通信路径中的多个部件中的一个上。如果从客户到数据源的通信没能到达其目的地,那么该指定的重试部件就是重试把客户通信发送到目的地的惟一部件。其他的部件不重试把通信发送到目的地,而是将通信未能到达目的地通知给重试部件。

[0028] 当客户通信在通信路径中的一个部件上失败时,需要重试部件。此外,当通信路径中的一个部件漏掉、或未能处理客户通信时,也需要重试部件。

[0029] 目的地可以是数据源。数据源的例子包括数据库和目录。在一个实施例中,包括数据库服务器和数据库的数据库系统可包含数据源。

[0030] 数据源可包括网络服务器、诸如目录和数据库设备的数据存储器以及其他存储设备。在互联网应用中,数据源 50 通常为网络服务器、或网络激活数据存储器。

[0031] 逻辑部件包括程序、应用程序、软件、编码的固件或硬件,其被配置成处理客户通信以便执行特定任务。重试部件是被配置成向数据源重试发送客户通信的逻辑部件。重试部件通过把通信发送给构成到数据源的通信路径的一部分的中间部件,来向数据源发送通信。该重试部件可使用初始通信路径、或类似通信路径来重试发送客户通信。作为选择,该重试部件可被配备成选择由在重试部件外部的设备上的不同中间部件构成的另一通信路径。如果初始数据源在通信路径中失败,该重试部件还可尝试访问另一数据源。

[0032] 重试部件的一个具体实例为位于服务器上的逻辑服务器部件,其为服务器服务并分派输入请求。用作服务器接口的其他部件同样可适于实施重试逻辑。

[0033] 在一个实施例中,将多个逻辑部件按顺序连接,以将通信从客户发送到数据源。这些部件中的一个被指定为用于通信路径部件的重试部件。如果客户通信被检测为未能访问数据源,则该重试部件向数据源重试发送第一通信。通信路径中的其他部件被指定向重试部件发送通信失败的指示。当通信路径中的一个部件失败,前一个部件(亦即,将请求发送到失败部件的部件)向重试部件发送指示。任何不是重试部件但却位于通信路径中的部件接收指示,并且必须同样地将向重试部件发送指示。这样,每一个其他部件都将指示通知给重试部件,而不需要向数据源重新发送第一通信。

[0034] 通信路径包括逻辑部件,其按顺序结合以将客户通信发送到数据源。通信路径的逻辑部件可位于不同的设备上。重试点为沿通信路径的一个位置,逻辑位于该重试点处,以重试发送失败的客户通信。

[0035] 在尝试重新向数据源发送客户通信之前,重试逻辑可确定失败的客户通信是否可以重试。某些类型的通信或特定请求可被指定为不可重试。这些指定可结合到重试逻辑中。

[0036] 在重新发送失败客户通信之前,重试逻辑还可确定替换的通信路径。该替换的通信路径可位于前一通信路径的部件所在的设备之外。重试部件可沿任一路径发送请求,以绕过失败的通信。例如,重试逻辑可考虑是否需要将通信发送到发生失败的设备之外。在进行该决定的过程中,重试部件还可确定一个特殊请求是否可以重试。

[0037] 在另一实施例中,在客户和数据源之间的第一通信路径上指定重试点。该第一通信路径包括多个逻辑部件,其顺序地接收并发送客户通信到数据源。该重试点构成通信路径的一部分。如果客户通信路径中的一个部件失败,或者不能到达其目的地,则向该重试部件发出失败通知。

[0038] 例如,在一个实施例中,如果客户通信不能沿位于重试点和数据源之间的通信路径部分访问数据源,则通过该通信路径中的其它部件中的一个或多个来通知重试点,指示客户通信未能访问数据源。

[0039] 根据一个实施例,重试逻辑可确定客户通信是否可以重试。只有在客户请求被确定为可以重试的情况下,重试逻辑才可重新发送客户通信。

[0040] 利用本文中描述的技术,系统可使用单个的指定重试部件来执行最优次数的重试。在很多情况下,一次重试就足以确定数据源是否可以访问。与此相反,传统的系统无法被设定成执行最优次数的重试,尤其是仅一次重试,因为一旦检测到失败,通常每个部件将自动重试客户通信。因此,即使各个部件都被设置为执行一次重试,实际执行的重试次数可能至少与执行重试的部件的数目一样多。

[0041] 更进一步,本发明的实施例允许重试逻辑被配置成做出其他确定和决定。具体来说,重试部件可被配置成确定一个特殊的客户通信是否可以重试,以及如果需要的话确定一条新的通信路径。传统的系统包括太多的重试失败的客户通信的部件,使得对重试逻辑进行配置变得困难,而且效率低。

[0042] 在另一个实施例中,重试部件在重试发送客户通信之前进行确定。该确定可与之前试图重试发送客户通信的次数相对应。如果之前尝试的次数小于最大重试阈值,则重试部件可重试发送客户通信。

[0043] B. 重试技术的系统描述

[0044] 图 1 示出了一种包括用于在客户和数据源之间传送通信的指定重试部件的系统 100。系统 100 的选定部件按顺序结合,以构成在客户 10 和数据源 50 之间的一个或多个通信路径。重试部件在选定的通信路径上标示出指定位置,在这个位置处,可重新发送失败客户通信以访问数据源 50。

[0045] 为了进行说明,在客户 10 向数据源 50 请求信息的环境下说明系统 100。然而,系统 100 可用在更广泛的环境中,可对通信进行处理并将其从源计算机发送到目的计算机,以及对响应进行处理并将其从目的计算机发送到源计算机。因此,参照作为客户通信目的地的数据源 50 对一个应用进行说明,其中,目的地为例如数据库系统。

[0046] 系统 100 可在客户 10 和数据源 50 之间提供多个通信路径。每一个通信路径仅包括一个部件作为重试部件,或者作为选择,仅包括任一服务器上的构成通信路径一部分的选定部件作为重试部件,该重试部件被配置成重试把失败客户通信发回给数据源。各个通信路径的其他部件被配置成将失败的客户通信提交到重试部件。

[0047] 在其他的优点中,本发明的实施例避免了失败的客户通信的无用重试次数的增

加。通过选定一个用于进行重试的部件,使系统 100 进行重试的次数得以减少。其结果是,与其他不是特别地使用重试逻辑的系统相比,尤其在与其他系统的最坏情况,即多个部件停止发出重试请求相比较时,将失败报告回客户 10 的延迟可显著减少。

[0048] 由一个实施例所提供的另一优点在于,重试逻辑可通过较容易且很少的支持在系统中得以实施。例如,重试逻辑可在服务器的单个逻辑部件中得以实施。在重试逻辑已经处于适当位置的情况下,可将附加部件加入到环境中,而无需对环境进行实质性的配置。

[0049] 此外,将重试逻辑集中到一个部件中将使对通信重试的管理变得容易。重试逻辑可被配置成防止对先前已被指定为不可重试的某些客户通信进行重试。例如,安全的通信先前可被指定为不可重试。可以基于请求类型来确定请求是否可以重试。例如,那些无论什么时候都返回同样的答复的请求是可以重试的。

[0050] 与此相反,其他传统的系统允许通信路径的各个部件都能重新发送失败客户通信。这通常导致延迟延长,因为在客户被最终通知先前的通信已经失败之前,一个失败部件可使得在通信路径的上行的多个部件重试发送客户通信。

[0051] 在图 1 所示的结构中,系统 100 包括负载平衡设备 25、第一服务器 20、第二服务器 30 和第三服务器 40。这些服务器的组合可相当于一个服务器场 (farm),其用来在将客户通信发送到目的地之前对客户通信进行处理。负载平衡设备 25 选定这些服务器中的一个,以形成客户 10 和数据源 50 之间的通信路径。

[0052] 网络可相当于一个广域网,例如互联网。客户 10 可为一个终端,其被配置成通过网络进行通信。客户 10 和网络的一个典型实例包括一个带有访问互联网的网络浏览器的台式计算机。

[0053] 负载平衡设备 25 截取从客户 10 发送到数据源 50 的客户通信。客户通信可相当于一个数据请求。客户通信可选择性地请求在数据源 50 内写入数据、修改已有数据、或删除已有数据。响应于接收客户通信,负载平衡设备 25 根据各个服务器的可用资源来选定一个服务器以访问数据源 50。每个服务器 20、30、40 都包括多个逻辑部件,其按顺序对客户通信进行处理。在图 1 提供的实例中,负载平衡设备 25 选定第一服务器 20,从而形成一个由客户 10、负载平衡设备 25、第一服务器 20 和数据源 50 组成的通信路径 12。在第一服务器 20 中的多个逻辑部件 (参见图 2) 按顺序结合,从而形成通信路径 12 的一部分。

[0054] 根据一个实施例,服务器 20、30、40 中的每一个都配置有重试逻辑,其用于在该服务器内从指定逻辑部件重新发送失败客户通信。图 1 示出了第一服务器 20,其被配置成包括重试部件 22。在一个实施例中,重试部件 22 直接与负载平衡设备 25 通信。例如,重试部件 22 可在直接从平衡负载设备 25 分派客户通信的逻辑服务器部件上得以实施。第一服务器 20 的多个其他部件 24 沿通信路径 12 被设置在重试部件 22 之后。

[0055] 在该结构中,对顺序结合以形成通信路径 12 的第一服务器 20 的各个部件进行分布,使得序列中的第一部件成为重试失败客户通信的惟一部件。对于失败客户通信而言,第一服务器 20 中的其他部件 24 向重试部件 22 返回一个客户通信失败的指示。部件 24 不尝试重新发送失败客户通信。

[0056] 在其他的实施例中,重试逻辑可被设置在沿客户 10 和数据源 50 之间的通信路径之一的其它位置。在一个实施例中,负载平衡设备 25 包括用作重试部件 22 的逻辑。在另一个实施例中,客户 10 包括用作重试部件 22 的逻辑。通过将重试部件 22 从数据源 50 中

除去,减少了在通信路径上可以重试向数据源 50 发回失败客户通信的部件的数目。

[0057] 图 2 所示的方框图示出了被配置成包括重试逻辑的服务器的操作。该服务器包括多个逻辑部件,其以顺序的方式对发送到服务器的通信进行处理。

[0058] 在一个实施例中,第一服务器 120 包括使用多重的分层的部件处理客户通信 112。该部件按照顺序 114 排列,以处理客户通信 112。顺序 114 中的至少一个部件被指定为重试点 122。

[0059] 在一个实施例中,重试部件 122 对应于在序列 114 中最先接收客户通信 112 的部件。在序列 114 中使用的其它部件包括第一应用程序 124 和第二应用程序 126。第一应用程序 124 从重试点 122 接收客户通信 112。第二应用程序 126 从第一部件 124 接收客户通信 112。第二应用程序 126 把客户通信 112 发送给数据源 150。只有在重试点 122,服务器 120 才试图把失败客户通信 112 重新发回给数据源 150。

[0060] 在一个实施例中,重试部件 122 对应于被配置成包括重试逻辑的第一服务器 120 的服务器部件。第一应用程序 124 可相当于一个表示层,其用于表示从数据源 150 到客户 110 的信息。第二应用程序 126 可相当于特殊类型网络通信的业务逻辑或其他应用程序。例如,在 JAVA 环境中,第一应用程序 124 为 JAVA Servlet Engine(JSERVE) 应用程序,以表示来自数据源 150 的 JAVA 激活数据。第二应用程序 126 可相当于 EJB 服务器应用程序,其传送用于完成商业应用程序的逻辑。每一个应用程序都可独立地对客户通信 112 进行处理和修改。

[0061] 数据源 150 可利用相反的顺序 116 将响应 132 传回给客户。通过该相反的顺序 116,响应 132 被数据源 15 传回给第二应用程序 126,然后,第二应用程序 126 将该响应发送到第一应用程序 124,然后发送到重试点 122。接下来,响应 132 被发回给客户 110。

[0062] 有时,处理客户通信 112 的部件中的一个失败。该失败可由序列 114 中的任意部件引起,或者作为选择,可由数据源 150 引起。例如,数据源 150 可变得不可用或太过拥堵,而不能处理客户通信 112。然后,当第二应用程序 126 尝试传送客户通信 112 到数据源 150 时,将产生失败。位于失败客户通信点处的部件可利用相反的顺序 116 发送一个客户通信 112 失败的指示。重试点 122 响应于接收到客户通信 112 失败的指示,可重试将客户通信 112 发回给数据源 150。

[0063] 当重新发送失败客户通信 112 时,重试点 122 实施一个重试序列。在一个实施例中,第一重试序列 118 包括与序列 114 相同的部件和排列。在另一个实施例中,第二重试序列 119 包括另一服务器 130 的部件。例如,第二重试序列 119 可使得重试点 122 将客户通信 112 发送到第二服务器 130 的第一应用程序 134 上。第一应用程序 134 可将客户通信 112 发送到该服务器上的第二应用程序 136 上。第二服务器 130 的第一和第二部件 134 和 136 可执行与第一服务器 120 的第一和第二应用程序 124 和 126 相似的或相同的功能。

[0064] 更进一步,如果服务器 120 包括多个用于处理和发送客户通信的过程,其他的实施例可使得重试点 122 选定同一服务器 120 上的另一过程。

[0065] 在一个实施例中,每一个部件都在其后续层中保存部件的拓扑图。然后,每个部件都可选定可用的部件。重试点 122 可利用拓扑图来确定其他服务器、或相同服务器上的部件的状态。重试点 122 可配备有智能信息,以利用来自拓扑图的信息选择合适的路径来发送重试请求。

[0066] 此外,其他的实施例可提供其他的重试序列。例如,如果序列 114 中的一个部件被检测为已经失败,则当位于重试点 122 处的部件重试发送失败客户通信 112 时,可选择绕过该部件。

[0067] 服务器部件 122 可包括用以选定重试序列的逻辑。例如,如果客户通信 112 是安全的,服务器部件 122 可只使用第一服务器 120 内部的重试序列。作为选择,服务器部件 122 可根据性能选定重试部件。例如,服务器部件 122 可确定应该将重试保持在同一服务器上,以便使用具有用于特殊客户请求的有用数据集的高速缓存。

[0068] 图 3 示出了在客户 310 和数据源 350 之间的通信路径中实施重试逻辑的流程图。该通信路径包括负载平衡设备 315、服务器部件 320、第一应用程序 330、第二应用程序 340 和数据源 350。图 3 中假定服务器部件 320 包括重试逻辑。此外,图 3 中假定重试逻辑不选择一个新的通信路径来重新发送失败客户通信。

[0069] 如上所述,如果图 3 所示的系统使用传统的重试技术,那么将导致无用重试次数的增加。现在,将使用图 3 来说明一个系统的操作过程,该系统被配置成从指定部件重试失败客户通信,由此减小由传统的系统所产生的延迟。通过仅仅使用指定重试部件 320,本发明的实施例可将重试次数减少至一次尝试。

[0070] 利用在此所述的技术,客户通过网络把发出的客户通信 312 传送到负载平衡设备 315。该负载平衡设备 315 将客户通信 312 发送到服务器部件 320。该服务器部件 320 可位于由负载平衡设备 315 选定的服务器上。客户通信 312 由服务器部件 320、第一应用程序 330、和第二应用程序 340 发送到数据源 350。

[0071] 例如,如果第二应用程序 340 不能访问数据源 350,那么第二应用程序 340 可将失败通知 322 发送到第一应用程序 330。该第一应用程序 330 将通知 322 发送到服务器部件 320。

[0072] 响应于接收来自第一应用程序 330 的通知 322,服务器部件 320 可将重试 332 发回给第一应用程序 330。重试 332 包括客户通信 312 的内容。该重试 332 被发送到第一应用程序 330、第二应用程序 340,然后被发送到数据源 350。

[0073] 如果重试 332 成功,来自数据源 350 的响应 342 被返回到客户 310。可以利用第二应用程序 340、第一应用程序 330、服务器部件 320、负载平衡设备 315 和客户 310 这样的顺序,把响应 342 发送到客户 310。

[0074] C. 实施重试技术的方法

[0075] 图 4 示出了一种实施相对于服务器的重试逻辑的方法,其中该服务器用于使客户与数据源连接。如图 4 所描述的方法可在如图 2 所描述的部件和系统上得以实施。

[0076] 在步骤 410 中,对失败客户通信进行检测。该失败客户通信可在失败点处被检测到。例如,如果第二应用程序 126(图 2 中示出)不能访问数据源 150,则检测点在第二应用程序 126 处。

[0077] 在步骤 420 中,将失败客户通信报告给重试点。检测到失败的部件可按照通信路径的顺序,将失败报告回给相邻的部件。一个或多个中间部件相结合,以按顺序将客户通信从检测到失败的部件发送到重试点。作为选择,检测到失败客户通信的部件可直接将失败报告给重试点,从而绕过通信路径上的中间部件。

[0078] 在步骤 430 中,将失败客户通信从重试点重新发送到数据源。可响应于重试点接

收到客户通信失败的报告而自动执行该步骤。

[0079] 在步骤 440 中,确定重试是否成功。根据一个实施例,如果重试失败,则将失败报告给步骤 450 中的客户。因此,重试点可被配置成只尝试一次重试。

[0080] 如果重试成功,则步骤 460 将来自于数据源的响应报告回客户。例如,如果客户通信相当于一个读取操作,则发回给客户的响应包括数据和其他信息,使得客户的浏览器能够显示对应于客户请求的信息。

[0081] 与其他的重试技术相对比,如图 4 所描述的实施例所提供的优点在于,对失败客户通信进行检测并采取行动的点被定位在客户通信路径中的一个选定部件上。发生重试行为的点没有被客户通信的通信路径上的多个部件毫无区别地共享。当多个逻辑部件相结合而形成客户通信的通信路径时,一个实施例使得仅有一个部件对失败采取行动。这种方法保护了其中有数百或数千个客户通信同时被处理的系统的资源。

[0082] 此外,重试部件可以是通信路径中对失败进行检测的惟一位置。作为选择,可在客户通信路径中的另一个部件上检测失败点。更进一步地,重试部件可对外部监视器检测到的失败客户通信进行重试。

[0083] 而且,被配置成实施如图 4 所描述的方法的系统还可被配置成实施带有重试逻辑的其它特征。图 5 描述了另一种重试技术的实例。

[0084] 图 5 示出了一种用于在把客户与数据源相连的系统上实施重试逻辑的方法。图 2 中的附图标记用于说明一个实施例所用的示例性部件。如图 5 所描述的方法可在形成客户 110 和数据源 150 之间的通信路径的一个或多个逻辑部件上被实施。

[0085] 在步骤 510 中,由传送客户通信到数据源的逻辑部件之一对失败客户通信进行检测。该逻辑部件可相当于序列 114 中重试点 122 后的一个部件。如果逻辑部件之一不能访问序列 114 中的下一个中间部件,则可将客户通信检测为失败。如果序列 114 中的最后一个逻辑部件不能访问数据源 150,同样可将客户通信检测为失败。

[0086] 在步骤 520 中,将客户通信失败的指示传送到重试点。该指示可沿相反的顺序被发回给重试部件 122。作为选择,检测到失败客户通信的部件可将指示直接发送到重试点 122,从而绕过在客户通信最初从重试点 122 发送时对客户通信进行处理的中间部件。

[0087] 在步骤 530 中,确定失败客户通信是否可以重试。在一个实施例中,在重试点 122 处进行确定。例如,重试点 122 可相当于服务器 120 的服务器部件,其配置有包括用以确定失败客户通信是否可以重试的智能信息重试逻辑。

[0088] 为了确定客户通信是否可以重试,重试点 122 可确定与客户通信有关的信息。该信息可以在表或其他数据结构中被参考,以便指示出失败客户通信是否应该重试。重试点 122 可确定某些类型的通信是不可重试的。例如,安全通信可被确定为不可重试。或者,通信的类型可指示请求是否可以重试。例如,静态数据请求可被预先确定为可以重试。更进一步地,重试点 122 可被配置成将客户通信的某些源和 / 或目的地确定为不可重试。

[0089] 如果步骤 530 中的确定结果为失败客户通信是不可重试的,则步骤 540 中通知客户该客户通信失败。重试点 122 可将客户通信失败的指示发送给客户 110。

[0090] 在一个实施例中,如果步骤 530 中的确定结果为失败客户通信是可以重试的,则在步骤 550 中,重试点 122 选定一个通信路径以重试发送失败客户通信。例如,重试点 122 可访问另一服务器,并使用一个新的结合有其他服务器的部件的通信路径。或者,如果,例

如确定在数据源 150 处发生故障,则重试点 122 可确定使用先前的通信路径。

[0091] 一旦选定了通信路径,重试点 122 就将失败客户通信重新发回给数据源 150。重试点 122 通过将客户通信发送到所选定的通信路径的序列中的下一个后续部件,来发送客户通信。

[0092] 在例如基于网络的通信的应用中,请求的内容可指出请求是否可以重试。例如,URL(统一资源定位符)可包含一个变量或其他用语,以指示请求是否可以重试。

[0093] 图 6 示出了一种数据结构 600,其能以重试逻辑来实施,以确定某些失败客户通信是否可以重试。该数据结构 600 使从客户通信确定的一个或多个参数与相应值进行匹配,以确定客户通信是否可以重试。所述从客户通信确定的一个或多个参数属于一组参数特征 610。各个参数的相应值属于一组参数值 620。

[0094] 在所提供的实例中,参数特征 610 包括:安全参数 612,用以确定客户通信是否安全;操作参数 614,用以确定客户通信指定的为何种操作(读、写、删除等);一个或多个用户定义参数 616;和/或一个或多个厂家定义参数 618。用户定义参数可由客户端的用户进行定义或配置。厂家定义参数可由服务器供应商进行定义或配置。例如,在服务器场中,厂家在服务器上提供或配置一个或多个逻辑应用程序。

[0095] 参数值 620 包括以下值。安全参数值 622 可要求对非安全通信进行失败客户通信重试,但不对安全通信进行失败客户通信重试。操作参数值 624 可指定对指定了读操作的客户通信进行重试,但不对指定了写或删除操作的客户通信进行重试。值 626 可被配置成与用户定义参数 616 匹配。举例来说,如果客户通信的目的地为特定的网络地址 IP(a),那么值 626 可排除重试失败客户通信。值 628 可被配置成与厂家定义参数 618 匹配。例如,对于某些具有 ID(a)的客户端用户,厂家定义参数可排除重试失败客户通信。

[0096] 用于确定可以重试的客户通信的更为复杂的数据结构和技术也是可能的。在一个实施例中,可对数据结构 600 进行修改,以确定可以利用位于重试点外部的新的通信路径重试客户通信。例如,安全客户通信是可以重试的,但只有当其利用相同服务器的部件进行传送时才可以重试。

[0097] 在一个特定应用中,表 600 可相当于统一资源定位符(URL)的模式。例如,如果从网络浏览器发送的超文本传输协议(HTTP)请求中的 URL 模式包含特殊的模式,那么该模式可与特定的重试功能或指示相匹配。例如,指定或包括诸如 `http://website/documents/reads/*` 的模式的请求可被确定为可以重试。但指定诸如 `http://website/documents/updates/*` 的模式的请求可被确定为不可重试。在后一种情况下,该请求不得是可重试的,因为请求更新数据在本质上意味着,被访问的数据是动态的,而不是静态的。

[0098] D. 替换的配置

[0099] 虽然本文中描述的实施例仅仅描述了客户 10 和数据源 50 之间的通信路径的单个重试部件,但其他的实施例可提供多个重试部件。具体来说,多个重试部件可以位于客户 10 和数据源 50 之间的通信路径上的关键位置。例如,一个实施方案中可使服务器 20 和负载平衡设备 25 各自包括一个重试部件。服务器 20 上的部件被配置成,如果客户通信在服务器和数据源之间失败,则服务器 20 上的部件从服务器的重试部件重试发送失败客户通信。同样地,负载平衡设备 25 上的部件被配置成,如果客户通信在负载平衡设备的逻辑部件中的一个上失败,则负载平衡设备 25 上的部件从负载平衡设备重试发送客户通信。

[0100] 本文中描述的实施例集中在从客户到数据源的通信上。然而,本文中所描述的技术并不局限于此。相反,这些技术可应用于发送方发送信息到指定接受方的任意情况中,其中信息必须通过多个中间部件。例如,该信息可为由数据源响应于来自客户的请求而发送到客户的响应。请求还可从客户被发送到提供服务但不是被客户检索的数据源的服务器。

[0101] E. 硬件描述

[0102] 图 7 所述的方框图示出了一种通过其实施本发明实施例的计算机系统 700。计算机系统 700 包括总线 720 或其他的用于传递信息的通信机构,以及与总线 720 相连接并用于处理信息的处理器 704。计算机系统 700 还包括一个主存储器 706,例如随机存取存储器 (RAM) 或其他的动态存储器,其与总线 702 相连,用于存储信息和要被处理器 704 执行的指令。在要被处理器 704 执行的指令的执行过程中,主存储器 706 还可用来存储临时变量或其他的中间信息。计算机系统 700 还包括只读存储器 (ROM) 708 或其他的静态存储器,其与总线 702 相连,用于存储静态信息和用于处理器 704 的指令。提供了诸如磁盘或光盘的存储设备 710,其与总线 702 相连,用于存储信息和指令。

[0103] 计算机系统 700 可通过总线 702 与诸如阴极射线管 (CRT) 的显示器 712 相连,该显示器用于显示信息给计算机用户。包括字母数字和其他键的输入设备 714 与总线 702 相连,用于向处理器 704 传送信息和指令选择。另一种类型的用户输入设备为光标控制器 716,例如鼠标、跟踪球、或用于向处理器 704 传送方向信息和指令选择以及控制显示器 712 上的光标移动的光标方向键。这种输入设备通常在两个轴,即第一轴 (例如 X) 和第二轴 (例如 Y) 上具有两个自由度,这使得该设备可在平面内进行定位。

[0104] 本发明涉及用于实施本文所述技术的计算机系统 700 的使用。根据本发明的一个实施例,响应处理器 704 执行在主存储器 706 中包含的一个或多个指令的一个或多个序列,计算机系统 700 执行这些技术。这种指令可从诸如存储设备 710 的另一计算机可读介质被读入到主存储器 706 中。主存储器 706 中包含的指令序列的执行使得处理器 704 执行本文中所描述的过程步骤。在替换实施例中,硬件电路可用于替代软件指令或与软件指令结合使用,来实施本发明。因此,本发明的实施例不局限于任何具体的硬件电路和软件的结合。

[0105] 本文中使用的术语“计算机可读介质”指的是参与向处理器 704 提供要执行的指令的所有介质。这种介质可采取多种形式,包括但不限于非易失性介质、易失性介质和传输介质。例如,非易失性介质包括诸如存储设备 710 的光盘或磁盘。易失性介质包括如主存储器 706 的动态存储器。传输介质包括同轴电缆、铜线和光纤,其包括构成总线 702 的线。传输介质同样可采取声波或光波的形式,例如在无线电波和红外数据通信过程中产生的波。

[0106] 通常形式的计算机可读介质包括,例如,软盘、软磁盘、硬盘、磁带、或其他任何磁介质、CD-ROM、其他任何光介质、穿孔卡、纸带、具有孔图案的其它任何物理介质、RAM、PROM、和 EPROM、FLASH-EPROM、其它任何存储器芯片或磁带、下文所述的载波、或计算机可读的其它任何介质。

[0107] 各种形式的计算机可读介质可涉及用于向处理器 704 传送一个或多个指令的一个或多个序列以用于处理。例如,这些指令开始时可存储在远程计算机的磁盘上。该远程计算机可将这些指令加载到其动态存储器中,并使用调制解调器通过电话线发送这些指令。计算机系统 700 的本地调制解调器可通过电话线接收数据,并使用红外发送器将该数据转换为红外信号。红外检测器可接收在红外信号中携带的数据,并且合适的电路可将数据放

置在总线 702 上。总线 702 将数据传送到主存储器 706, 处理器 704 从该主存储器 706 检索并执行这些指令。在被处理器 704 执行之前或之后, 主存储器 706 收到的指令可选择性地存储到存储设备 710 上。

[0108] 计算机系统 700 还包括与总线 702 相连的通信接口 718。通信接口 718 提供了一个双向的数据通信, 其与连接到本地网络 722 的网络链路 720 相连。例如, 通信接口 718 可为综合服务数字网 (ISDN) 卡或调制解调器, 以提供到相应类型电话线的数据通信连接。另举例来说, 通信接口 718 可为局域网 (LAN) 卡, 以提供到兼容的 LAN 的数据通信连接。无线链路同样可以实施。在任意这样的实施例中, 通信接口 718 发送和接收电信号、电磁信号或光信号, 这些信号转载表征不同类型信息的数字数据流。

[0109] 网络链路 720 通常通过一个或多个网络向其他数据设备提供数据通信。例如, 网络链路 720 可通过本地网络 722 向主机 724 提供连接, 或向由互联网服务供应商 (ISP) 726 操作的数据设备提供连接。ISP 726 又通过目前通常被称为“互联网”728 的全球分组数据通信网络提供数据通信服务。本地网络 722 和互联网 728 都使用转载数字数据流的电信号、电磁信号或光信号。通过各种网络的信号和在网络链路 720 上并通过通信接口 718 的信号是传输信息的载波的典型形式, 其中, 这些信号从计算机系统 700 传送数字数据, 并向计算机系统 700 传送数字数据。

[0110] 计算机系统 700 可通过网络、网络链路 720 和通信接口 718 发送信息并接收包括程序代码的数据。在互联网情况下, 服务器 730 可通过互联网 728、ISP 726、本地网络 722 和通信接口 718 传输用于应用程序的请求代码。

[0111] 处理器 704 接收到代码后, 可执行收到的代码, 并 / 或将收到的代码存储在存储设备 710、或其他的非易失性存储器中, 以用于日后的执行。通过这种方式, 计算机 700 可获得载波形式的应用程序代码。

[0112] 在前面的说明中, 参照本发明的具体实施例对其进行了描述。然而, 很明显, 在不脱离本发明的精神和范围的情况下, 可对本发明进行各种修改和改变。因此, 说明书及附图应视作说明性的, 而不是进行限定。

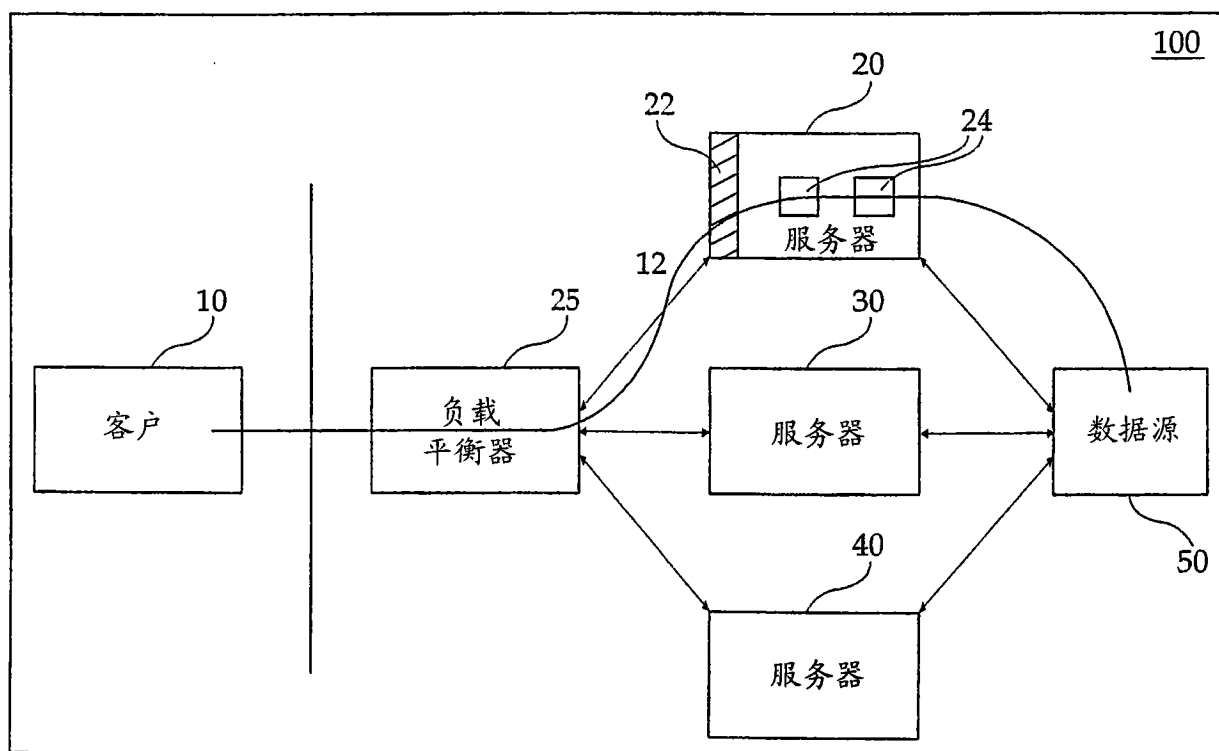


图 1

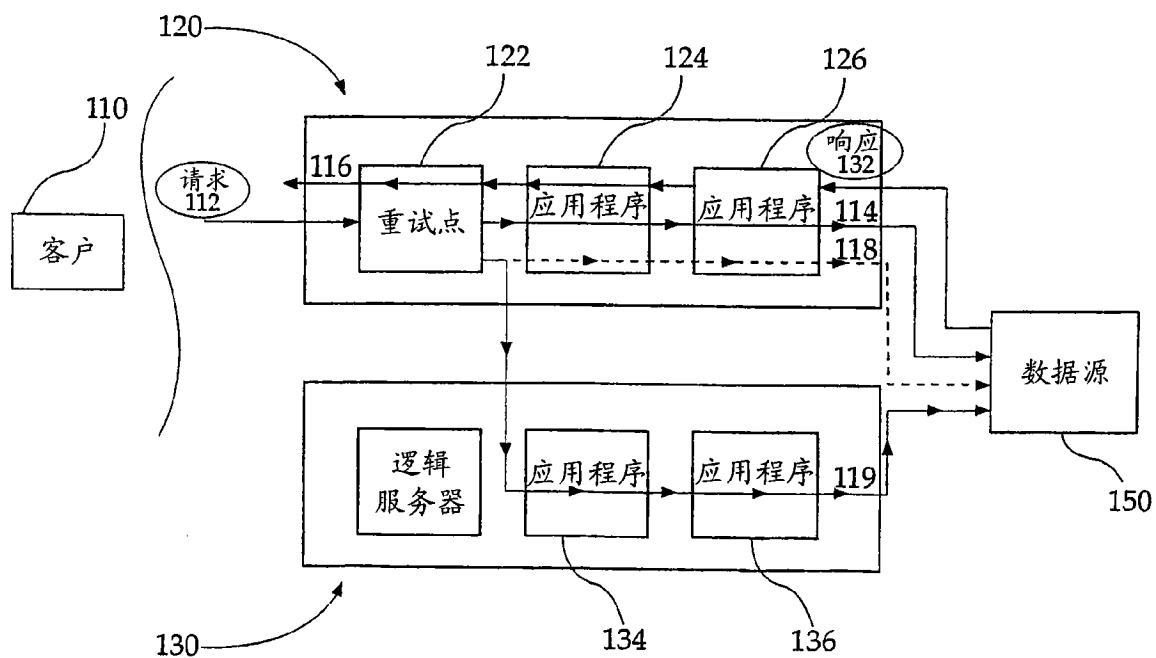
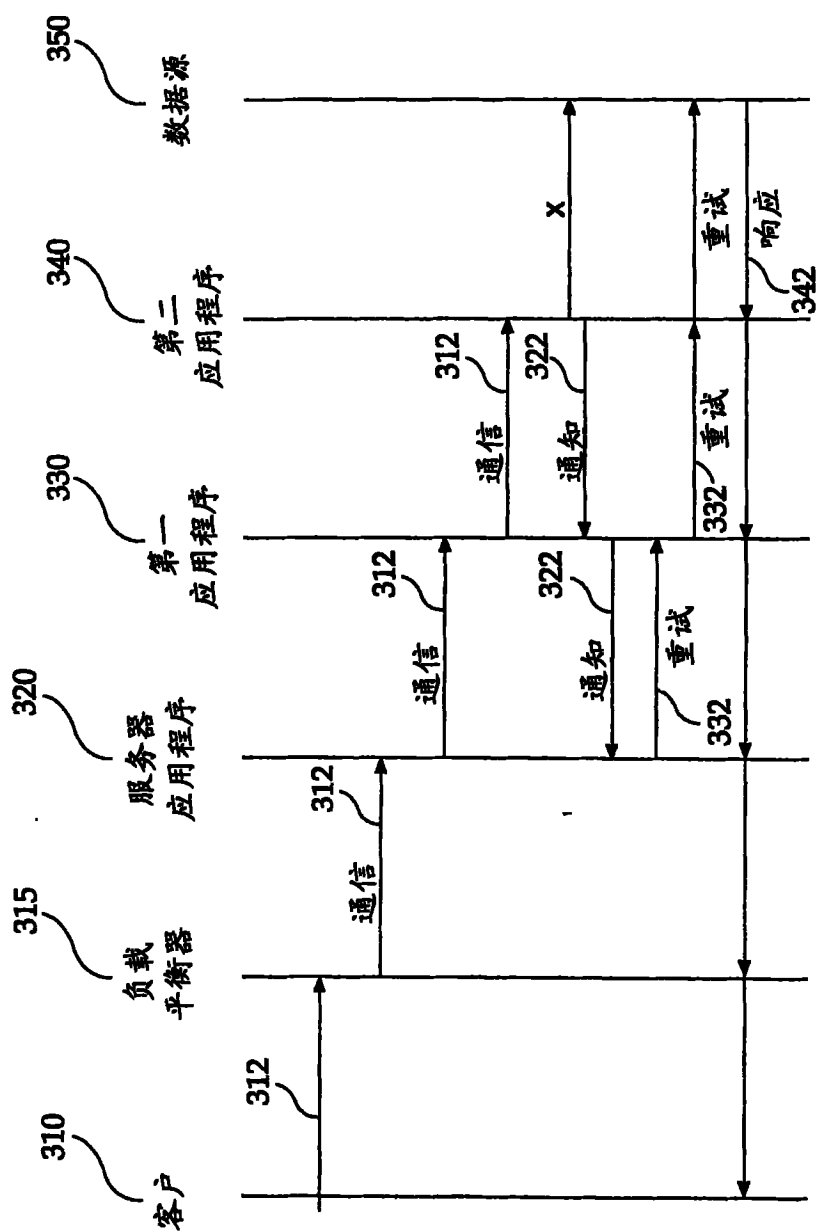


图 2



3
四

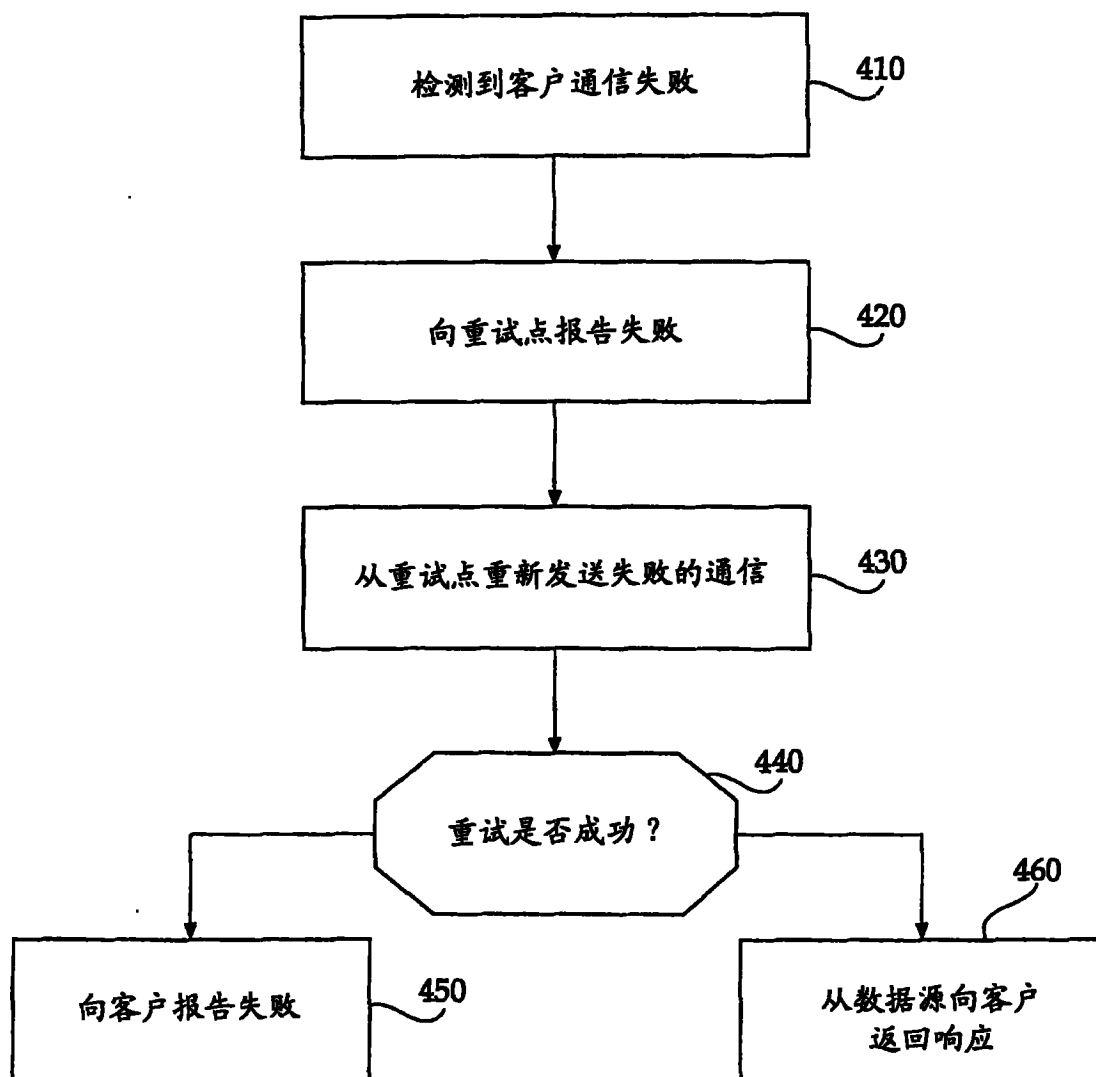


图 4

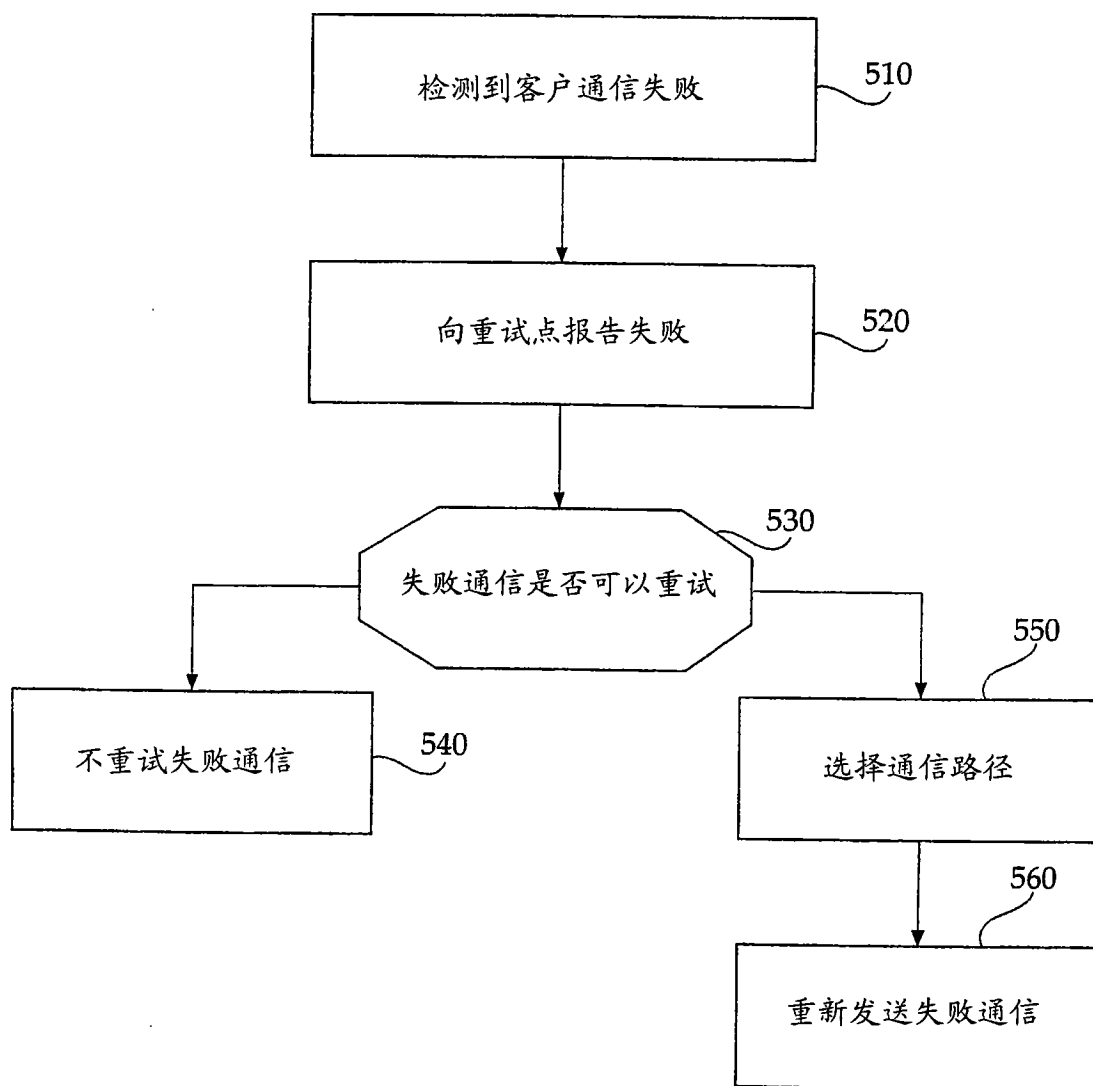


图 5

通信类型	重试配置
安全 / 不安全	重试不安全通信
操作	重试“读”操作
用户定义的 A	不向 IP _A 重试
厂家定义的 B	不从 IP _B 重试

600

图 6

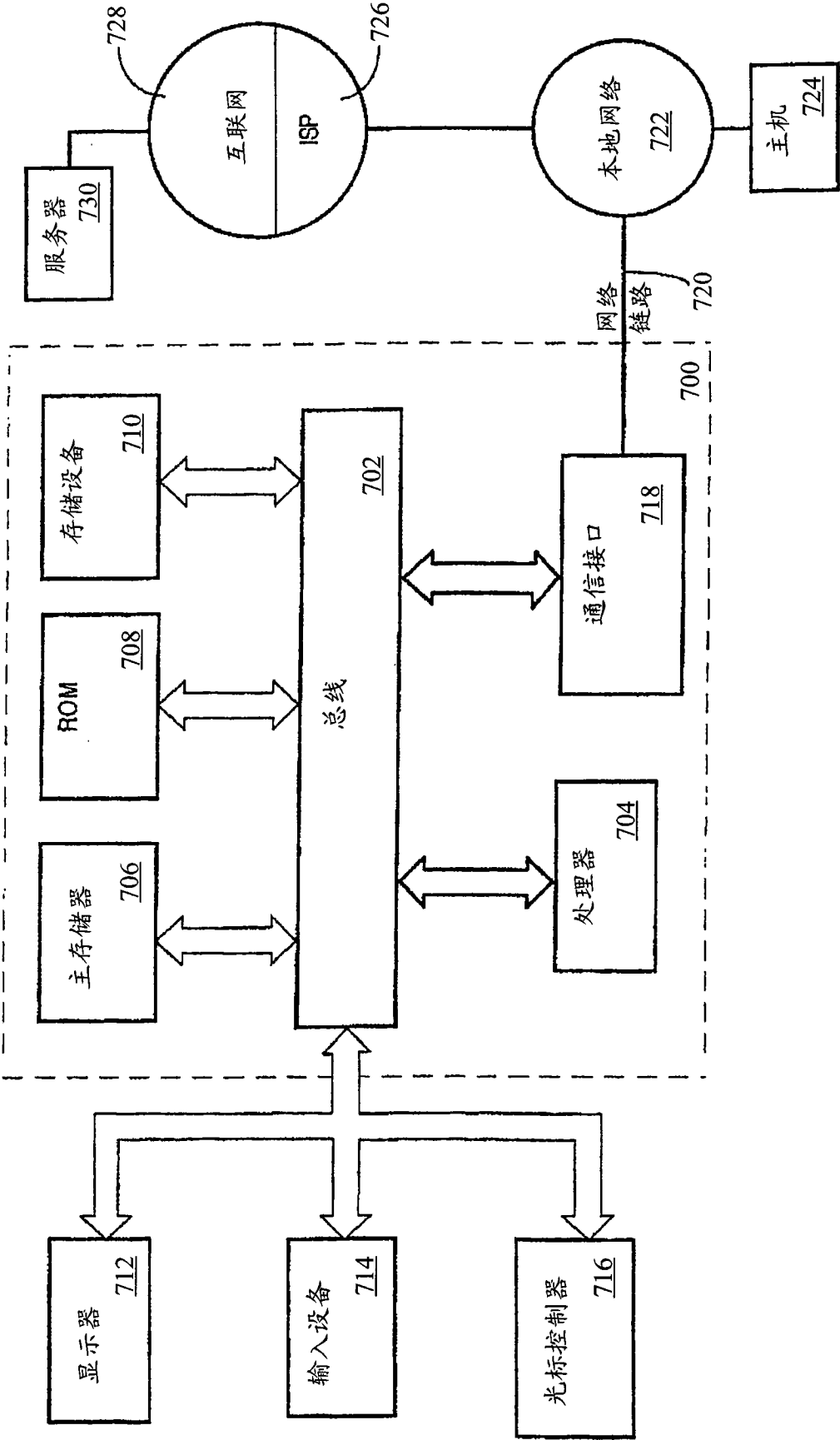


图 7

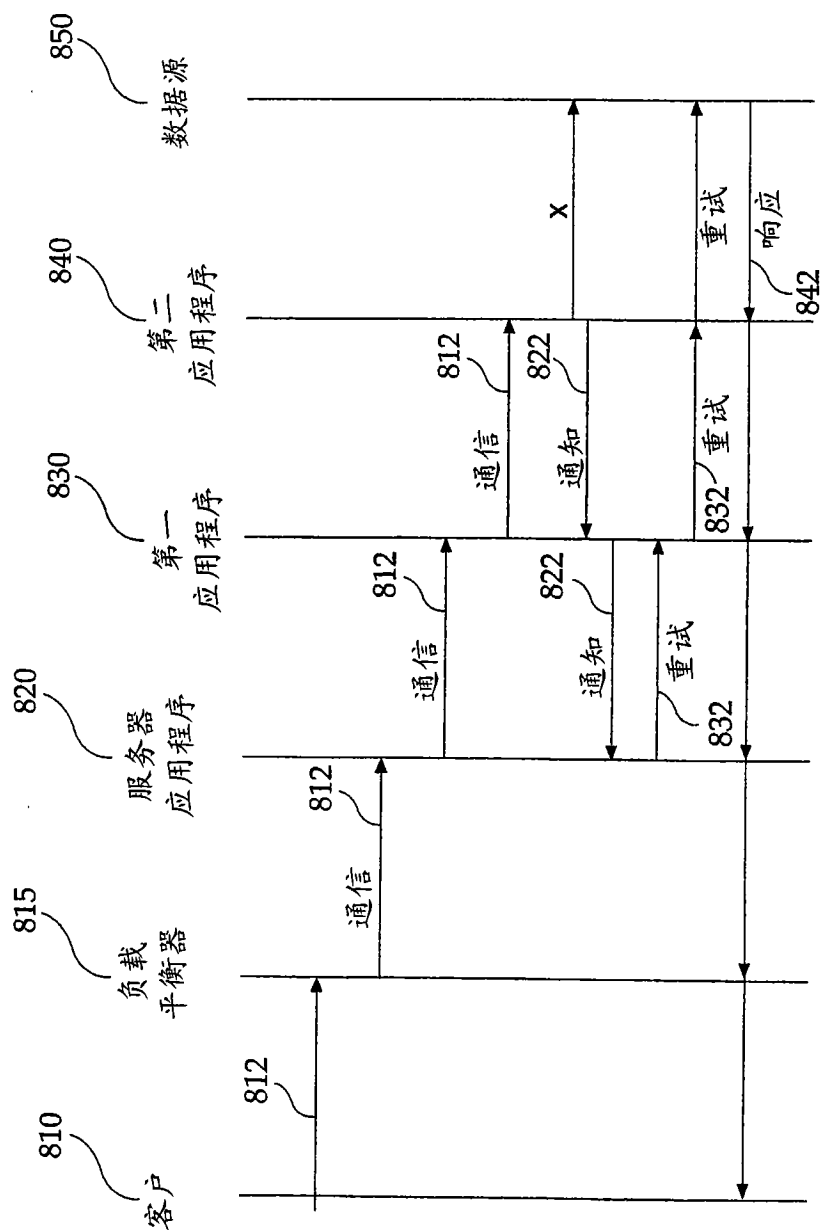


图 8
(现有技术)