(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102422274 A
(43)申请公布日 2012.04.18

(21)申请号 201080020349.1

(22)申请日 2010.04.30

(30)优先权数据
12/436,233 2009.05.06 US

(85)PCT申请进入国家阶段日
2011.11.04

(86)PCT申请的申请数据
PCT/US2010/033255 2010.04.30

(87)PCT申请的公布数据
WO2010/129432 EN 2010.11.11

(71)申请人 微软公司
地址 美国华盛顿州

(72)发明人 C·P·贾泽斯基

(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 高见

(51)Int.Cl.
G06F 15/16(2006.01)
G06F 9/06(2006.01)
G06F 13/00(2006.01)

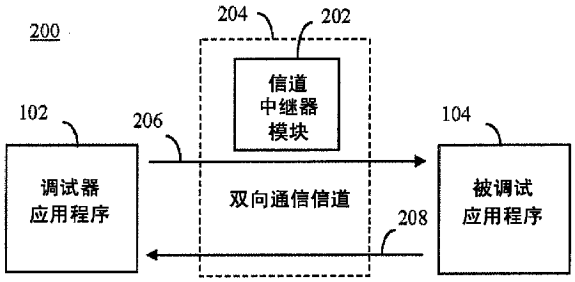
权利要求书 3 页 说明书 16 页 附图 14 页

(54)发明名称

低特权调试信道

(57)摘要

基于浏览器的调试器应用程序被配置成调试第二基于浏览器的应用程序。调用信道中继器模块以在调试器应用程序和第二应用程序之间建立双向通信信道。通过信道中继器模块从第二应用程序传输开始指示。第二应用程序进入被阻止等待状态。对第二应用程序执行调试操作。通过信道中继器模块，将调试请求消息传输到第二应用程序，以导致第二应用程序从被阻止等待状态转换到运行状态，并执行由调试请求消息指示的调试动作。通过信道中继器模块，从第二应用程序接收对调试请求消息的响应。第二应用程序从运行状态转换到被阻止等待状态。



CN 102422274 A

1. 一种第一基于浏览器的应用程序 (102,300) 中的方法 (500),包括 :

调用 (502) 信道中继器模块以在所述第一基于浏览器的应用程序和第二基于浏览器的应用程序之间建立双向通信信道,所述第一和第二基于浏览器的应用程序被允许通过使用由浏览器启用的通信协议,来通过所述双向通信信道进行通信,而无需提升特权 ;

通过所述信道中继器模块,从所述第二基于浏览器的应用程序接收 (504) 开始指示,所述第二基于浏览器的应用程序处于被阻止等待状态 ;以及

对所述第二基于浏览器的应用程序执行 (506) 调试操作,所述执行包括

通过所述信道中继器模块,将调试请求消息传输 (1302) 到所述第二基于浏览器的应用程序,以导致所述第二基于浏览器的应用程序从所述被阻止等待状态转换到运行状态,并执行由所述调试请求消息指示的调试动作,以及

通过所述信道中继器模块,从所述第二基于浏览器的应用程序接收 (1304) 对所述调试请求消息的响应,所述第二基于浏览器的应用程序被从所述运行状态转换到所述被阻止等待状态。

2. 在如权利要求 1 所述的方法,其特征在于,所述调用信道中继器模块以在所述第一基于浏览器的应用程序和第二基于浏览器的应用程序之间建立双向通信信道包括 :

调用远离主存所述第一基于浏览器的应用程序的客户机计算机的服务器上的所述信道中继器模块。

3. 在如权利要求 1 所述的方法,其特征在于,所述调用信道中继器模块以在所述第一基于浏览器的应用程序和第二基于浏览器的应用程序之间建立双向通信信道包括 :

调用主存所述第一基于浏览器的应用程序的客户机计算机上的所述信道中继器模块。

4. 如权利要求 1 所述的方法,其特征在于,所述通过信道中继器模块从所述第二基于浏览器的应用程序接收开始指示包括 :

作为由所述第二基于浏览器的应用程序向所述信道中继器模块传输开始指示的结果,从所述信道中继器模块异步地接收所述开始指示。

5. 如权利要求 1 所述的方法,其特征在于,还包括 :

向所述信道中继器模块传输等待指示 ;

将浏览器导航到所述第二基于浏览器的应用程序的地址 ;以及

等待来自所述信道中继器模块的对所述等待指示的响应 ;

其中所述通过所述信道中继器模块从所述第二基于浏览器的应用程序接收开始指示包括

响应于所述等待指示,并且作为由所述第二基于浏览器的应用程序向所述信道中继器模块传输开始指示的结果,从所述信道中继器模块接收所述开始指示。

6. 如权利要求 1 所述的方法,其特征在于,所述通过信道中继器模块将调试请求消息传输到所述第二基于浏览器的应用程序,以导致所述第二基于浏览器的应用程序从所述被阻止等待状态转换到运行状态,并执行由所述调试请求消息指示的调试动作包括 :

将所述调试请求消息传输到所述信道中继器模块,所述信道中继器模块被配置成将所述调试请求消息传输到所述第二基于浏览器的应用程序,作为对由所述第二基于浏览器的应用程序向所述信道中继器模块进行的先前的传输的响应 ;以及

其中所述通过所述信道中继器模块从所述第二基于浏览器的应用程序接收对所述调

试请求消息的响应包括

从所述信道中继器模块接收包括由所述第二基于浏览器的应用程序向所述信道中继器模块传输的调试已执行消息的对所述调试请求消息的响应。

7. 如权利要求 6 所述的方法，其特征在于，所述通过信道中继器模块将调试请求消息传输到所述第二基于浏览器的应用程序，以导致所述第二基于浏览器的应用程序从所述被阻止等待状态转换到运行状态，并执行由所述调试请求消息指示的调试动作还包括：

等待来自所述信道中继器模块的对所述调试请求消息的响应；

8. 如权利要求 1 所述的方法，其特征在于，所述调试请求消息包括设置断点指令、运行指令或步进指令中的至少一个，其中所述通过所述信道中继器模块从所述第二基于浏览器的应用程序接收对所述调试请求消息的响应包括：

接收已经到达所述断点或所述步进完成中的至少一项的指示。

9. 如权利要求 1 所述的方法，其特征在于，所述通信协议是超文本传输协议 (HTTP)，并且从所述第一基于浏览器的应用程序和从所述第二基于浏览器的应用程序向所述信道中继器模块进行的传输是根据 HTTP 来配置的。

10. 如权利要求 9 所述的方法，其特征在于，所述从信道中继器模块接收到的开始指示和所述对从信道中继器模块接收到的调试请求消息的响应是根据不是 HTTP 的第二通信协议来配置的。

11. 在如权利要求 1 所述的方法，其特征在于，所述调用信道中继器模块以在所述第一基于浏览器的应用程序和第二基于浏览器的应用程序之间建立双向通信信道包括：

将所述双向通信信道配置为低特权调试流水线。

12. 一种包括在其上记录有计算机程序逻辑的计算机可读介质的计算机程序产品，包括：

用于允许处理器执行权利要求 1-11 中的任一项权利要求的计算机程序逻辑装置 (2234)。

13. 一种基于浏览器的调试器应用程序 (102,300)，包括：

信道建立器模块 (302)，所述信道建立器模块被配置成调用信道中继器模块 (202) 以在所述基于浏览器的调试器应用程序 (102,300) 和第二基于浏览器的应用程序 (104,400) 之间建立双向通信信道 (204)，所述基于浏览器的调试器应用程序 (102,300) 和所述第二基于浏览器的应用程序 (104,400) 被允许使用由浏览器 (112) 启用的通信协议通过所述双向通信信道 (204) 进行通信，而无需提升特权；

调试启动模块 (304)，所述调试启动模块 (304) 被配置成通过所述信道中继器模块 (202) 从所述第二基于浏览器的应用程序 (104,400) 接收开始指示 (312)；以及

调试器模块 (306)，所述调试器模块被配置成对所述第二基于浏览器的应用程序 (104,400) 执行一个或多个调试操作，所述调试器模块 (306) 包括

调试指令传输模块 (308)，所述调试指令传输模块 (308) 被配置成通过所述信道中继器模块 (202)，将调试请求消息 (314) 传输到所述第二基于浏览器的应用程序 (104,400)，以导致所述第二基于浏览器的应用程序 (104,400) 从被阻止等待状态转换到运行状态，并执行由所述调试请求消息 (314) 指示的调试动作，以及

调试响应接收器模块 (310)，所述调试响应接收器模块 (310) 被配置成通过所述信道

中继器模块 (202) 从所述第二基于浏览器的应用程序 (104,400) 接收对所述调试请求消息 (314) 的响应 (316),所述第二基于浏览器的应用程序 (104,400) 从所述运行状态转换到所述被阻止等待状态。

14. 如权利要求 13 所述的基于浏览器的调试器应用程序,其特征在于,所述通信协议是超文本传输协议 (HTTP),而所述调试指令传输模块被配置成将调试请求消息作为超文本传输协议 (HTTP)post 传输到所述信道中继器模块。

15. 如权利要求 14 所述的基于浏览器的调试器应用程序,其特征在于,所述从信道中继器模块接收到的开始指示和所述对从所述信道中继器模块接收到的调试请求消息的响应是根据不是 HTTP 的第二通信协议来配置的。

低特权调试信道

[0001] 背景

[0002] 调试器是允许程序员监视程序的执行的工具。调试器可以停止正被调试的程序的执行、重新启动程序的执行、在程序中设置断点，和 / 或访问或改变存储器中的值。调试器可以允许程序被一步一步地运行（叫做“步进”），当遇到断点时停止在代码的特定行（中断），并且可以在程序处于断点或当程序正在运行时允许检查变量的值（检查）。当程序正在运行时，除观察和报告程序状态之外，一些调试器也可以修改程序状态。因此，调试器会非常有用的。

[0003] 通常为调试器进程提供了相对高级别的特权，以使得它们被允许以此方式对程序进行调试。然而，使得调试器对于查找错误有用的此相同功能还能使调试器成为有用的软件破解工具，其可以被用来违反安全性，包括检索敏感信息、逃避拷贝保护、规避数字权限管理等等。由此，必须仔细地管理向调试器进程提供的特权级别。

[0004] 例如，在某些情况下，可能需要使用在客户机计算机上运行的基于浏览器应用程序来调试服务器上的应用程序。然而，在客户机计算机上运行的应用程序通常没有足够的特权来调试服务器上的应用程序。如果在客户机上运行的应用程序具有足够的特权来调试服务器上的应用程序，则将会出现很大的安全性问题。为将基于客户机的基于浏览器的应用程序的特权级别提高到调试基于服务器的应用程序的足够的级别，通常必须在客户机计算机处安装启用软件 (enabling software)。例如，被配置成提高基于客户机计算机的应用程序的特权的调试 DLL（动态链接库），或其他形式的调试 API（应用程序编程接口），可能需要被安装在客户机计算机处。这样的 API 可能对于许多应用程序不可用。为获取对这样的 API 的访问，应用程序可能需要用户通过普通 .MSI(Microsoft ® Installer（安装器））或通过引用 ActiveX 控件来安装更具特权的调试堆栈（例如，Microsoft ® ICorDebug），或者可能需要执行其他复杂的进程。

[0005] 概述

[0006] 提供本发明内容以便以简化的形式介绍将在以下具体实施方式中进一步描述的一些概念。本发明内容并不旨在标识所要求保护主题的关键特征或必要特征，也不旨在用于限制所要求保护主题的范围。

[0007] 提供了用于使基于浏览器的调试器应用程序能调试基于浏览器的被调试应用程序而无需要求提升特权的方法、系统，以及计算机程序产品。

[0008] 在一种实现中，提供了基于浏览器的调试器应用程序调试第二基于浏览器的应用程序的方法。调用信道中继器模块以在第一基于浏览器的应用程序和第二基于浏览器的应用程序之间建立双向通信信道。通过信道中继器模块从第二基于浏览器的应用程序传输开始指示。第二基于浏览器的应用程序进入被阻止等待状态。可以对第二基于浏览器的应用程序执行一个或多个调试操作。例如，调试请求消息可以通过信道中继器模块传输到第二基于浏览器的应用程序以导致第二基于浏览器的应用程序从被阻止等待状态转换到运行状态，并执行由调试请求消息指示的调试动作。可以通过信道中继器模块，从第二基于浏览器的应用程序接收对调试请求消息的响应。第二基于浏览器的应用程序从运行状态转换到

被阻止等待状态。

[0009] 在另一种实现中，提供了基于浏览器的调试器应用程序。基于浏览器的调试器应用程序包括信道建立器模块、调试启动模块，以及调试器模块。信道建立器模块被配置成调用信道中继器模块以在基于浏览器的应用程序和第二基于浏览器的应用程序之间建立双向通信信道。调试启动模块被配置成通过信道中继器模块从第二基于浏览器的应用程序接收开始指示。调试器模块被配置成对第二基于浏览器的应用程序执行一个或多个调试操作。调试器模块包括调试指令传输模块和调试响应接收器模块。调试指令传输模块被配置成通过信道中继器模块，向第二基于浏览器的应用程序传输调试请求消息。调试请求消息被配置成导致第二基于浏览器的应用程序从被阻止等待状态转换到运行状态。第二基于浏览器的应用程序执行由调试请求消息指示的调试动作。调试响应接收器模块被配置成通过信道中继器模块，从第二基于浏览器的应用程序接收对调试请求消息的响应。第二基于浏览器的应用程序从运行状态转换到被阻止等待状态。

[0010] 此处还描述了供调试器应用程序调试被调试应用程序的计算机程序产品。

[0011] 此外，此处还描述了用于被调试应用程序、信道中继器模块，并用于如此处所描述的更进一步的实施例的方法、系统，以及计算机程序产品。

[0012] 下面将参考各个附图，详细描述本发明的进一步特点和优点，以及本发明的各实施例的结构和操作。值得注意的是，本发明不仅限于此处所描述的特定实施例。这样的实施例只是出于例示的目的。基于此处所包含的原理，另外的实施例对那些相关领域技术人员是显而易见的。

[0013] 附图简述

[0014] 结合到本说明书并构成本说明书的一部分的附图示出了本发明，且与描述一起，进一步用于说明本发明的原理，并允许那些精通相关的技术人员实施和使用本发明。

[0015] 图 1 示出了其中可以实现本发明的实施例的示例客户机 - 服务器通信系统的框图。

[0016] 图 2 示出了根据本发明的示例实施例的调试系统的框图。

[0017] 图 3 示出了根据示例实施例的调试器应用程序的框图。

[0018] 图 4 示出了根据示例实施例的被调试应用程序的框图。

[0019] 图 5 示出了根据示例实施例的提供用于执行调试的调试器应用程序的过程的流程图。

[0020] 图 6 和 7 示出了根据示例实施例的调试系统的框图。

[0021] 图 8 示出了根据示例实施例的调试系统的框图。

[0022] 图 9-12 示出了根据示例实施例的提供用于启动对被调试应用程序的调试的过程的流程图。

[0023] 图 13-16 示出了根据示例实施例的提供用于由调试器应用程序执行对被调试应用程序的调试的过程的流程图。

[0024] 图 17-19 示出了根据示例实施例的提供用于终止调试的过程的流程图。

[0025] 图 20 示出了根据示例实施例的用于配置低特权调试流水线的流程图。

[0026] 图 21A 和 21B 示出了根据示例实施例的用于客户机 - 客户机低特权调试的示例系统的框图。

[0027] 图 22 示出了可以被用来实现本发明的各实施例的示例计算机的框图。

[0028] 通过下面的结合附图对本发明进行的详细说明，本发明的特点和优点将变得更加显而易见，在附图形中，类似的附图标记在整个说明书中标识对应的元素。在附图中，相同的附图标记一般指示相同的、功能上类似的和 / 或在结构上类似的元素。元素首先在其中出现的附图由对应的附图标记中最左边的数字来指示。

[0029] 详细描述

[0030] I. 引言

[0031] 本说明书公开了包括本发明的特征的一个或多个实施例。所公开的实施例只例示了本发明。本发明的范围不仅限于所公开的实施例。本发明由所附的权利要求进行定义。

[0032] 说明书中对“一个实施例”、“实施例”、“示例实施例”等等的引用表示所描述的实施例可包括特定特征、结构或特性，但是，每一个实施例可以不一定包括该特定特征、结构，或特征。此外，这样的短语不一定是指同一实施例。此外，当结合实施例描述特定特征、结构或特性时，假定在本领域技术人员的知识范围内，可以与其他实施例一起实施这样的特征、结构或特性，无论是否被显式地描述。

[0033] II. 调试器和特权

[0034] 许多现代处理器体系结构具有允许操作系统以不同的特权级别运行的 CPU 模式。例如，访问资源的进程、资源和指令被用特权级别进行标记。当一个进程试图使用资源或试图执行有特权的指令时，处理器确定该进程是否具有执行此项操作的许可，并准许或者拒绝该操作。这可防止用户进程损坏操作系统或在机器上运行的其他进程。可能存在各种数目的特权级别。例如，某些处理器支持两个特权级别，如用户（低特权）和超级用户（高特权）。其他处理器具有四个级别，级别 #0 具有最大特权（高特权），级别 #3 具有最小的特权（级别 #2 和级别 #3 可以被视为低特权）。

[0035] 调试器是允许程序员监视程序的执行的工具。一般而言，调试器进程是高特权进程，而被调试程序（正在被调试的程序或调试目标）通常是低特权进程。当在本地调试时（例如，用户正在他自己的计算机上调试程序），两者（调试器或被调试程序）中的哪一个特权更大通常无关紧要，因为计算机上的所有信息通常都属于同一个用户。远程调试（调试在通过网络连接可访问的系统上运行的程序）更有问题。远程调试通常在客户机 - 服务器环境中执行（例如，在存在客户机 - 服务器关系的情况下）。

[0036] 客户机 - 服务器关系是在两个不同的计算机上运行的两个计算机程序之间的关系，其中，一个程序，客户端，作出对另一个程序（服务器）的服务请求。诸如电子邮件交换、web 访问以及数据库访问之类的典型网络功能都基于客户机 - 服务器模型。术语“服务器”一般而言是指在计算机上运行的服务于请求的进程和该服务在其上面运行的实际计算机两者。服务器通常专用于通过计算机网络——典型地通过请求 - 响应协议——提供一个或多个服务。服务通常由被设计成能处理多个并行请求的服务器应用程序提供。服务器应用程序的示例包括邮件服务器、文件服务器、web 服务器，以及代理服务器。服务器通常是高特权的，以使得它可以服务于请求并访问它服务于请求所需要访问的信息。

[0037] 客户机是通过网络访问服务器上的远程服务的应用程序或计算机。客户机 - 服务器模型广泛地用于因特网上，其中，用户可以通过因特网协议套件连接到在远程系统上操作的服务。web 浏览器是用户计算机上的可以访问网络中的任何 web 服务器上的信息并检

索网页供显示在用户计算机上的客户机程序（或进程，即，正在执行的程序是进程）。这样的客户机对于保护机密、有特权或敏感信息，以及对于防窜改目的，通常是低特权的。

[0038] 例如，在某些情况下，可能需要使用在客户机计算机上运行的基于浏览器的应用程序来调试服务器应用程序。然而，在客户机计算机上运行的基于浏览器的应用程序通常没有足够的特权来调试服务器上的应用程序。基于客户机的应用程序的特权级别相对于服务器被保持得较低，以使得基于客户机的应用程序不能损坏操作系统或在服务器上运行的其他进程。为将基于客户机的基于浏览器的应用程序的特权级别提高到调试基于服务器的应用程序的足够的级别，通常必须在客户机计算机处安装允许软件。例如，被配置成提高基于客户机计算机的应用程序的特权的调试DLL（动态链接库），或其他形式的调试API（应用程序编程接口），可能需要被安装在客户机计算机处。

[0039] 本发明的各实施例克服了用于在客户机－服务器环境中调试应用程序的传统技术的缺陷。下面几部分描述了这样的实施例的示例。

[0040] III. 示例低特权调试实施例

[0041] 此处所描述的示例实施例只是为说明性用途而提供，而不是限制性的。此外，通过此处的原理，附加的结构和操作实施例，包括修改／更改，对于相关领域的技术人员将变得显而易见。

[0042] 本发明的各实施例涉及对应用程序的调试。在各实施例中，在客户机计算机上运行的调试器应用程序被允许在不提升在客户机计算机处运行的基于浏览器的应用程序的特权级别的情况下调试基于服务器的应用程序。由于不必提高调试器应用程序的特权级别，因此，避免了服务器上的安全性问题。在一个实施例中，在调试器应用程序和被调试应用程序之间建立双向通信信道。与调试相关的消息被允许通过双向通信信道在调试器和被调试应用程序之间传输。在一个实施例中，消息未经修改地到达，并按由特定发送方（调试器或者被调试应用程序）发送的相同顺序。在一个实施例中，被调试应用程序被配置成能够阻止其自己的执行的线程，同时等待来自调试器应用程序的调试请求。

[0043] 在各实施例中，可以由调试器和被调试应用程序使用各种协议来通过双向通信信道进行通信。例如，在一个实施例中，可以根据HTTP（超文本传输协议）配置双向通信信道。以此方式，标准web安全机制可以由通信来使用。

[0044] 图1示出了其中可以实现本发明的实施例的示例客户机－服务器通信系统100的框图。如图1所示，系统100包括客户机计算机106、服务器计算机108，以及网络110。下面描述了系统100。

[0045] 客户机计算机106和服务器计算机108可以各自是此处所描述的任何合适类型或以其他方式已知的计算设备，包括台式计算机、移动计算机或计算设备，或其他类型的计算机。网络110被配置成可通信地耦合客户机计算机106和服务器计算机108。网络110可包括有线和／或无线的一个或多个通信链路和／或通信网络，如PAN（个人区域网络）、LAN（局域网）、WAN（广域网），或诸如因特网之类的网络的组合，并可包括有线或无线的一个或多个电信网络，如GSM（全球移动通信系统）网络、3G网络，和／或其他网络。

[0046] 服务器计算机108包括（例如，存储在存储器和／或相关联的存储器中）调试器应用程序102和被调试应用程序104。调试器应用程序102包括调试能力，并可包括更进一步的能力。例如，调试器应用程序102被配置成能够调试被调试应用程序104。调试器

和被调试应用程序 102 和 104 可以是相同类型的应用程序或进程，或者也可以是不同类型的应用程序或进程。如图 1 所示，客户机计算机 106 被配置成执行浏览器 112。在一个实施例中，调试器和被调试应用程序 102 和 104 可以是基于浏览器的应用程序，它们能够被由诸如浏览器 112 之类的浏览器加载和运行。例如，调试器和被调试应用程序 102 和 104 可以由浏览器 112 直接执行和 / 或由浏览器 112 的“插件”或“附加件”执行。这样的插件的示例包括由位于美国华盛顿州雷德蒙市的微软公司发布的 Silverlight™、Javascript™ 插件 (Javascript™ 是由位于加利福尼亚州圣克拉拉市的 Sun Microsystems 有限公司发布的)、由位于加利福尼亚州圣何塞市的 Adobe Systems 有限公司开发的 Adobe Flash 播放器，以及其他插件。由此，调试器和被调试应用程序 102 和 104 可以是 Silverlight™ 进程、Javascript™ 进程、Flash 进程 (例如，.SWF 文件)，以及其他应用程序 / 脚本。浏览器 112 可以是任何类型的 web 浏览器，包括 Microsoft Internet Explorer ®、Google Chrome、Mozilla Firefox 等等。

[0047] 如由图 1 中的虚线所指示的，浏览器 112 可以加载调试器应用程序 102。例如，浏览器 112 可以导航 (例如，通过用户与浏览器 112 的交互) 至由包括调试器应用程序 102 的服务器计算机 108 服务的网页。结果，调试器应用程序 102 被下载到客户机计算机 106 中。在客户机计算机 106 上，用户可以与调试器应用程序 102 进行交互 (例如，通过浏览器 112) 以调试被调试应用程序 104。根据一个实施例，在调试器应用程序 102 和被调试应用程序 104 之间建立双向通信信道，通过该信道，调试器应用程序 102 能够调试被调试应用程序 104。

[0048] 例如，图 2 示出了根据示例实施例的调试系统 200 的框图。如图 2 所示，系统 200 包括调试器应用程序 102、被调试应用程序 104、信道中继器模块 202，以及双向通信信道 204。如图 2 所示，调试器应用程序 102 通过双向通信信道 204 将调试相关消息 206 传输到被调试应用程序 104，而被调试应用程序 104 通过双向通信信道 204 将与调试相关的消息 208 传输到调试器应用程序 102。在一个实施例中，调试相关消息 206 和 208 分别以未经修改的形式并以与它们被传输的相同的顺序到达被调试应用程序 104 和调试器应用程序 102。在一个实施例中，调试器应用程序 102 和被调试应用程序 104 之间的消息 206 和 208 在双向通信信道 204 中流过信道中继器模块 202。信道中继器模块 202 允许调试器应用程序 102 通过双向通信信道 204 来调试被调试应用程序 104，而无需提升调试器应用程序 102 的特权级别。

[0049] 可以以各种方式来配置调试器应用程序 102 和被调试应用程序 104 以便能通过双向通信信道 204 进行调试。例如，图 3 示出了根据示例实施例的调试器应用程序 300 的框图。此外，图 4 还示出了根据示例实施例的被调试应用程序 400 的框图。调试器应用程序 300 是调试器应用程序 102 的示例，而被调试应用程序 400 是被调试应用程序 104 的示例。下面描述了调试器应用程序 300 和被调试应用程序 400。

[0050] 如图 3 所示，调试器应用程序 300 包括信道建立器模块 302、调试启动模块 304，以及调试器模块 306。在图 3 的示例中，调试器模块 306 包括调试指令传输模块 308 和调试响应接收器模块 310。信道建立器模块 302 被配置成调用信道中继器模块 202 以在调试器应用程序 102 和被调试应用程序 104 之间建立双向通信信道 204。调试启动模块 304 被配置成通过信道中继器模块 202 从图 4 的被调试应用程序 400 接收开始指示 312。调试器模块

306 被配置成对被调试应用程序 400 执行一个或多个调试操作。调试指令传输模块 308 被配置成通过信道中继器模块 202 将调试请求消息 314 传输到被调试应用程序 400。调试响应接收器模块 310 被配置成通过信道中继器模块 202 接收调试已执行消息 316，作为对来自被调试应用程序 400 的调试请求消息 314 的响应。

[0051] 如图 4 所示，被调试应用程序 400 包括调试控制模块 402 和应用程序执行模块 410。调试控制模块 402 被配置成执行对被调试应用程序 400 的调试，如由来自调试器应用程序 400 的通过信道中继器模块 202 的通信所控制的。应用程序执行模块 410 允许如由调试控制模块 402 所控制的，执行被调试应用程序 400 的应用程序代码。在图 4 的示例中，调试控制模块 402 包括调试开始模块 404、执行阻止模块 406，以及调试响应传输模块 408。调试开始模块 404 被配置成允许调试被调试应用程序 400。例如，如图 4 所示，调试开始模块 404 生成开始指示 412，用以指示对被调试应用程序 400 的调试可以进行。开始指示 412 由图 3 中的调试启动模块 304 通过信道中继器模块 202 接收到，来作为开始指示 312。

[0052] 如图 4 所示，执行阻止模块 406 从调试开始模块 404 接收阻止信号 414。调试开始模块 404 在传输开始指示 412 之后生成阻止信号 414。执行阻止模块 406 维持被调试应用程序 104 的状态，并被配置成在接收到阻止信号 414 之后进入被阻止等待状态。此外，执行阻止模块 406 还生成执行启用 / 禁用信号 418，该信号由应用程序执行模块 410 接收。在执行阻止模块 406 的被阻止等待状态，生成执行启用 / 禁用信号 418 以导致被调试应用程序 400 的代码在应用程序执行模块 410 中的执行被禁用（即，被停止或阻止）。

[0053] 如在图 4 中进一步示出的，执行阻止模块 406 可以从信道中继器模块 202 接收调试请求消息 416，该消息是由信道中继器模块 202 接收到的调试请求消息 314。调试请求消息 416 指示要对被调试应用程序 400 的代码执行的一个或多个调试动作 / 操作。在接收到调试请求消息 416 之后，执行阻止模块 406 变换到运行状态，并生成执行启用 / 禁用信号 418，以导致被调试应用程序 400 的代码应用程序执行模块 410 中的执行被启用（例如，执行启用 / 禁用信号 418 的值相对于被配置成禁用应用程序执行模块 410 的值而切换或改变）。由此，由应用程序执行模块 410 对被调试应用程序 400 的代码的执行可以进行。如图 4 所示，应用程序执行模块 410 生成执行输出 420，该输出由调试响应传输模块 408 接收。执行输出 420 指示正在由应用程序执行模块 410 执行的代码中的执行的当前点，一个或多个变量值，和 / 或关于被调试应用程序 104 的代码的执行的所感兴趣的更进一步的信息。

[0054] 此外，执行阻止模块 406 还生成调试指令 422。调试指令 422 是从调试请求消息 416（从图 3 的调试请求消息 314）中提取的调试指令。调试指令 422 可包括要对被调试应用程序 400 的代码执行的各种调试操作指令中的一个或多个，包括设置断点和运行，设置断点和步进的指令等等。调试指令 422 由调试响应传输模块 408 接收。调试响应传输模块 408 监视执行输出 420，并确定调试指令 422 何时完成，诸如到达断点，步进已完成等等。当调试响应传输模块 408 确定调试指令 422 已完成时，调试响应传输模块 408 生成由执行阻止模块 406 接收的阻止信号 426。执行阻止模块 406 被配置成在接收到阻止信号 426 之后转换到被阻止等待状态，并生成执行启用 / 禁用信号 418，以导致被调试应用程序 400 的的代码在应用程序执行模块 410 中的执行被禁用。此外，当调试响应传输模块 408 确定调试指令 422 已完成时，调试响应传输模块 408 还生成调试已执行消息 424。调试已执行消息 424 被传输到信道中继器模块 202，该模块 202 将调试已执行消息 424 作为调试已执行消息

316 传输到调试响应接收器模块 310。调试已执行消息 424 指示调试请求消息 314 已经被完成，并可以可任选地包括更进一步的信息，包括执行输出 420 的信息。

[0055] 回顾图 2，调试器应用程序 102 可以以各种方式执行对被调试应用程序 104 的调试。例如，图 5 示出了根据示例实施例的提供了调试器应用程序执行对被调试应用程序 104 的调试的过程的流程图 500。例如，流程图 500 可以由调试器应用程序 102 执行并由图 3 的调试器应用程序 300 执行。基于关于流程图 500 的讨论，其他结构和操作实施例对相关领域技术人员而言是显而易见的。在下面的子节描述了流程图 500。

[0056] A. 用于建立双向通信信道的示例实施例

[0057] 如图 5 所示，流程图 500 从步骤 502 开始。在步骤 502 中，调用信道中继器模块以在第一基于浏览器的应用程序和第二基于浏览器的应用程序之间建立双向通信信道。参考图 2，例如，可以调用信道中继器模块 202 以在调试器应用程序 102（第一基于浏览器的应用程序）和被调试应用程序 104（第二基于浏览器的应用程序）之间建立双向通信信道 204。例如，在一个实施例中，信道中继器模块 202 可以由图 3 所示出的信道建立器模块 302 来调用。当信道中继器模块 202 被调用时，信道中继器模块 202 变成被启用以从调试器应用程序 102 接收消息，并将它们传输到被调试应用程序 104，并且从被调试应用程序 104 接收消息，并将它们传输到调试器应用程序 102。调试器和被调试应用程序 102 和 104 被允许使用它们被加载到其中的浏览器（诸如图 1 所示出的浏览器 112）启用的通信协议来进行通信。例如，通信协议可以由浏览器直接启用（例如，HTTP）和 / 或由浏览器的附加件 / 插件启用。这允许调试器和被调试应用程序 102 和 104 通过双向通信信道进行通信，以调试被调试应用程序 104，而无需提升特权。

[0058] 在一个实施例中，信道中继器模块 202 可以被包括在调试器应用程序 300 中。在这样的实施例中，当对应用程序的调试将被执行时，信道建立器模块 302 可以调用调试器应用程序 300 中的信道中继器模块 202。在另一实施例中，信道中继器模块 202 可以是调试器应用程序 300 外部的，如被包括在与其中包括了调试器应用程序的同一计算机系统中，或被包括在不同的计算机系统中。在这样的实施例中，当对应用程序的调试将被执行时，信道建立器模块 302 可以通过将信道调用信号 318（图 3 所示出的）传输到信道中继器模块 202 来调用信道中继器模块 202。

[0059] 例如，图 6 示出了根据示例实施例的调试系统 600 的框图。调试系统 600 是图 2 所示出的调试系统 200 的示例。如图 6 所示，系统 600 包括调试器应用程序 102、被调试应用程序 104、信道中继器模块 202、双向通信信道 204、第一浏览器 112a、第二浏览器 112b，以及服务器计算机 108。在图 6 中，第一浏览器 112a 运行并为调试器应用程序 102 提供用户界面，而第二浏览器 112 运行并为被调试应用程序 104 提供用户界面。第一浏览器 112a 和第二浏览器 112b 可以被包括在同一浏览器中（例如，是图 1 中的浏览器 112 的分开的窗口或标签），或者也可以是分开的浏览器调用。在图 6 的实施例中，信道中继器模块 202 被包括在服务器计算机 108 中。由此，如上文所描述的，当对应用程序的调试将被执行时，图 3 所示出的信道建立器模块 302 可以通过将信道调用信号 318 传输到服务器计算机 108 上的信道中继器模块 202 来调用信道中继器模块 202。

[0060] 如图 6 所示，调试器应用程序 102 通过服务器计算机 108 上的信道中继器模块 202（例如，通过图 1 所示出的网络 110），将与调试相关的消息 206 传输到被调试应用程序

104。例如，调试器应用程序 102 将调试相关消息 206a（例如，图 3 的消息 314）传输到服务器计算机 108 上的信道中继器模块 202，该模块 202 又将相同的消息作为调试相关消息 206b（例如，图 4 的消息 416）传输到被调试应用程序 104。此外，被调试应用程序 104 还通过服务器计算机 108 上的信道中继器模块 202 将调试相关消息 208 传输到调试器应用程序 102。例如，被调试应用程序 104 将调试相关消息 208a（例如，图 4 的开始指示 412 或消息 424）传输到服务器计算机 108 上的信道中继器模块 202，该模块 202 又将相同的消息作为调试相关消息 208b（例如，图 3 的开始指示 312 或消息 316）传输到调试器应用程序 102。

[0061] 图 7 示出了根据示例实施例的调试系统 700 的框图。调试系统 700 是图 2 所示出的调试系统 200 的另一个示例。如图 7 所示，系统 700 包括调试器应用程序 102、被调试应用程序 104、信道中继器模块 202、双向通信信道 204、第一浏览器 112a、第二浏览器 112b，以及服务器计算机 108。在图 7 的实施例中，信道中继器模块 202 和双向通信信道 204 被包括在客户机计算机 106 中。当对应用程序的调试将被执行时，图 3 所示出的信道建立器模块 302 可以调用客户机计算机 106 中的信道中继器模块 202。如上文所描述的，虽然在图 6 中未示出，信道中继器模块可以被包括在调试器应用程序 102 中。

[0062] 如图 7 所示，调试器应用程序 102 通过客户机计算机 106 中的信道中继器模块 202 将调试相关消息 206 传输到被调试应用程序 104。例如，调试器应用程序 102 将调试相关消息 206a（例如，图 3 的消息 314）传输到客户机计算机 106 上的信道中继器模块 202，该模块 202 又将调试相关消息 206b（例如，图 4 的消息 416）传输到被调试应用程序 104。此外，被调试应用程序 104 还通过客户机计算机 106 上的信道中继器模块 202 将与调试相关的消息 208 传输到调试器应用程序 102。例如，被调试应用程序 104 将调试相关消息 208a（例如，图 4 的开始指示 412 或消息 424）传输到客户机计算机 106 上的信道中继器模块 202，该模块 202 又将调试相关消息 208b（例如，图 3 的开始指示 312 或消息 316）传输到调试器应用程序 102。

[0063] B. 用于启动调试的示例实施例

[0064] 在此子章节描述了用于启动调试的示例实施例。可以以各种方式启动对被调试应用程序 104 的调试。回顾图 5，在流程图 500 的步骤 504 中，通过信道中继器模块从第二基于浏览器的应用程序接收开始指示。参考图 2，例如，被调试应用程序 104 可以通过信道中继器模块 202 将开始指示传输到调试器应用程序 102 以启动对被调试应用程序 104 的调试。

[0065] 图 8 示出了根据示例实施例的调试系统 800 的框图。本章节参考图 8 以示出了用于启动对被调试应用程序 104 的调试的示例实施例。调试系统 800 类似于图 2 所示出的调试系统 200，并且还示出了在双向通信信道 204 中调试器应用程序 102 和被调试应用程序 104 之间的通过信道管理器模块的示例通信。参考示出了根据示例实施例的用于启动对被调试应用程序 104 的调试的相应的流程图的图 9-12 描述了图 8。图 9 示出了被调试应用程序中的用于启动调试的过程的流程图 900。图 10 和 11 示出了在启动调试过程中调试器应用程序的相应的流程图 1000 和 1100。图 12 示出了在启动调试过程中信道中继器模块的流程图 1200。基于关于流程图 900、1000、1100、1200，以及图 8 的系统 800 的讨论，其他结构和操作实施例对相关领域技术人员是显而易见的。

[0066] 如图 9 所示，流程图 900 始于步骤 902。在步骤 902 中，通过信道中继器模块，将

开始指示传输到第一基于浏览器的应用程序。例如，参考图 8，被调试应用程序 104 可以将开始指示 412 传输到信道管理器模块 202，而信道管理器模块 202 可以将开始指示 412 作为开始指示 312 传输到调试器应用程序 102。如上文参考图 4 所描述的，调试开始模块 404 可以生成开始指示 412，并且如图 3 所示，调试启动模块 304 可以接收开始指示 312。

[0067] 在步骤 904 中，第二基于浏览器的应用程序转换到被阻止等待状态。例如，在传输开始指示 412 之后，被调试应用程序 104 可以转换到被阻止等待状态。例如，如上文参考图 4 所描述的，执行阻止模块 406 可以生成执行启用 / 禁用信号 418，以导致被调试应用程序 400 的代码在应用程序执行模块 410 中的执行被禁用（即，被停止或阻止）。以此方式，被调试应用程序 104 的代码的执行可以被阻止，甚至在被调试应用程序 104 的第一代码行之前。

[0068] 关于调试器应用程序 102，可以以各种方式启动对被调试应用程序 104 的调试。流程图 1000 和 1100（图 10 和 11）示出了用于在调试器应用程序 102 中启动对被调试应用程序 104 的调试的示例实施例。例如，如图 10 所示，流程图 1000 包括步骤 1002。在步骤 1002 中，作为由第二基于浏览器的应用程序向所述信道中继器模块的传输开始指示的结果，从所述信道中继器模块异步地接收开始指示。例如，参考图 8，可以从信道管理器模块 202 异步地接收开始指示 312。在这样的实施例中，调试器应用程序 102 不必一定等待或提供调试器应用程序 102 正在等待开始指示 312 的指示。

[0069] 在图 11 的实施例中，流程图 1100 始于步骤 1102。在步骤 1102 中，将等待指示传输到信道中继器模块。例如，如图 8 所示，调试器应用程序 102 可以将等待指示 802 传输到信道管理器模块 202。例如，参考图 3，调试启动模块 304 可以被配置成传输等待指示 802。

[0070] 在步骤 1104 中，浏览器导航到第二基于浏览器的应用程序的地址。例如，如上文所描述的，调试器应用程序 102 可能先前已经被调用，如通过将浏览器 112 导航到包括调试器应用程序 102 的服务器计算机 108 上的网页。在步骤 1104 中，浏览器 112 可以导航到服务器计算机 108 上的被调试应用程序 104 的地址（例如，网页的统一资源定位符 (URL)）。以此方式，被调试应用程序 104 被加载，以供由调试器应用程序 102 进行调试。

[0071] 在步骤 1106 中，等待来自信道中继器模块的对张贴 (post) 的等待指示的响应。例如，参考图 8，在传输等待指示 802 之后，调试器应用程序 102 可以被配置成等待接收开始指示 312。步骤 1106 是可选的，而在其他实施例中，调试器应用程序 102 可以不一定等待开始指示 312（例如，调试器应用程序 102 可以同时执行其他功能）。

[0072] 在步骤 1108 中，响应于等待指示，且作为由第二基于浏览器的应用程序向信道中继器模块传输开始指示的结果，从信道中继器模块接收开始指示。例如，参考图 8，可以响应于等待指示 802 被传输到信道中继器模块 202，从信道管理器模块 202 接收开始指示 312。

[0073] 如上文所描述的，信道中继器模块 202 对于由调试器应用程序 102 和被调试应用程序 104 传输的信号执行中继器功能。参考图 12，流程图 1200 始于步骤 1202。在步骤 1202 中，从第一基于浏览器的应用程序接收等待指示。例如，如图 8 所示，等待指示 802 由信道中继器模块 202 从调试器应用程序 102 接收。步骤 1202 是可选的。

[0074] 在步骤 1204 中，从第二基于浏览器的应用程序接收开始指示。例如，如图 8 所示，开始指示 412 由信道中继器模块 202 从被调试应用程序 104 接收。

[0075] 在步骤 1206 中，向第一基于浏览器的应用程序传送开始指示。例如，如图 8 所示，

开始指示 412 由信道中继器模块 202 作为开始指示 312 传输到调试器应用程序 102。

[0076] C. 用于执行调试的示例实施例

[0077] 在此子章节描述了用于执行调试的示例实施例。回顾图 5，在流程图 500 的步骤 506 中，对第二基于浏览器的应用程序执行调试操作。参考图 2，例如，调试器应用程序 102 可以通过信道中继器模块 202 对被调试应用程序 104 执行一个或多个调试操作。

[0078] 可以以各种方式执行由调试器应用程序 102 对被调试应用程序 104 的调试。例如，图 13-16 示出了根据示例实施例的用于执行调试的流程图。图 13 和 14 示出了在调试器应用程序中执行用于调试的过程的流程图 1300 和 1400。图 15 示出了调试期间被调试应用程序的流程图 1500。图 16 示出了执行调试期间信道中继器模块的流程图 1600。为了说明，参考图 8 所示出的调试系统 800 描述了流程图 1300、1400、1500，以及 1600。基于关于流程图 1300、1400、1500、1600，以及图 8 的系统 800 的讨论，其他结构和操作实施例对相关领域技术人员是显而易见的。

[0079] 如图 13 所示，流程图 1300 始于步骤 1302。在步骤 1302 中，通过信道中继器模块将调试请求消息传输到第二基于浏览器的应用程序以导致第二基于浏览器的应用程序从被阻止等待状态转换到运行状态，并执行由调试请求消息指示的调试动作。例如，如图 8 所示，调试器应用程序 102 可以将调试请求消息 314 传输到信道管理器模块 202。例如，参考图 3，调试指令传输模块 308 可以被配置成传输调试请求消息 314。调试请求消息 314 可包括要对被调试应用程序 104 执行的一个或多个调试操作，如在别处所描述的。

[0080] 在步骤 1304 中，通过信道中继器模块，从第二基于浏览器的应用程序接收对调试请求消息的响应。例如，参考图 8，通过信道中继器模块 202，由调试器应用程序 102 接收调试已执行消息 316，作为对来自被调试应用程序 104 的调试请求消息 314 的响应。参考图 3，调试响应接收器模块 310 可以被配置成接收调试已执行消息 316。在接收到调试已执行消息 316 之后，被调试应用程序 104 的代码的执行被阻止。由此，用户（例如，正在与浏览器 112 进行交互）可以被允许在其当前执行状态执行对被调试应用程序 104 的调试分析。

[0081] 调试器应用程序 102 可以可任选地等待对调试请求消息 314 的响应。例如，在这样的实施例中，调试器应用程序 102 可以根据图 14 所示出的流程图 1400 来执行图 13 的流程图 1300。如图 14 所示，流程图 1400 始于步骤 1402。在步骤 1402 中，将调试请求消息传输到信道中继器模块。例如，如图 8 所示，调试器应用程序 102 可以将调试请求消息 314 传输到信道管理器模块 202。

[0082] 在步骤 1404 中，等待来自信道中继器模块的对调试请求消息的响应。例如，参考图 8，在传输调试请求消息 314 之后，调试器应用程序 102 可以被配置成等待接收调试已执行消息 316。步骤 1404 是可选的，而在其他实施例中，调试器应用程序 102 可以不一定等待调试已执行消息 316（例如，调试器应用程序 102 可以同时执行其他功能）。

[0083] 在步骤 1406 中，从信道中继器模块接收包括由第二基于浏览器的应用程序向信道中继器模块传输的调试已执行消息的对调试请求消息的响应。例如，参考图 8，通过信道中继器模块 202，由调试器应用程序 102 接收调试已执行消息 316——作为对来自被调试应用程序 104 的调试请求消息 314 的响应。

[0084] 关于被调试应用程序 104，可以以各种方式执行调试。例如，如图 15 所示，流程 1500 始于步骤 1502。在步骤 1502 中，响应于先前的传输，从信道中继器模块接收调试请求

消息。例如，参考图 8，由被调试应用程序 104 接收传输到信道中继器模块 202 的调试请求消息 314，作为调试请求消息 416。如上文参考图 4 所描述的，执行阻止模块 406 可以接收调试请求消息 416。在一个实施例中，由被调试应用程序 104 响应于先前到信道中继器模块 202 的传输，接收调试请求消息 416。例如，可以响应于被调试应用程序 104 向信道中继器模块 202 传输开始指示 412（例如，在启动调试时）或响应于向信道中继器模块 202 传输先前的调试已执行消息 424（例如，在紧邻的先前的调试周期期间），从信道中继器模块 202 接收调试请求消息 416。

[0085] 在步骤 1504 中，第二基于浏览器的应用程序从被阻止等待状态转换到运行状态。例如，在接收到调试请求消息 416 之后，被调试应用程序 104 可以从被阻止等待状态转换到运行状态。例如，如上文参考图 4 所描述的，执行阻止模块 406 可以生成执行启用 / 禁用信号 418，以导致被调试应用程序 400 的代码在应用程序执行模块 410 中的执行被启用。

[0086] 在步骤 1506 中，执行由调试请求消息指示的调试动作。被调试应用程序 104 执行由调试请求消息 416 指示的调试动作。例如，如图 4 所示，执行阻止模块 406 生成调试指令 422。调试指令 422 是从调试请求消息 416 中提取的调试指令。调试指令 422 可包括要对被调试应用程序 400 的代码执行的各种调试操作指令中的一个或多个，包括设置断点和运行，设置断点和步进的指令或此处所描述的或以其他方式已知的其他调试操作。

[0087] 在步骤 1508 中，将调试已执行消息传输到信道中继器模块。例如，参考图 8，调试已执行消息 424 由被调试应用程序 104 传输到信道中继器模块 202。如上文参考图 4 所描述的，调试响应传输模块 408 可以生成调试已执行消息 424。

[0088] 如上文所描述的，信道中继器模块 202 对于由调试器应用程序 102 和被调试应用程序 104 传输的信号执行中继器功能。参考图 16，流程图 1600 始于步骤 1602。在步骤 1602 中，从第一基于浏览器的应用程序接收调试请求消息。例如，如图 8 所示，调试请求消息 314 由信道中继器模块 202 从调试器应用程序 102 处接收。

[0089] 在步骤 1604 中，响应于从第二基于浏览器的应用程序到信道中继器模块的先前的传输，调试请求消息被传输到第二基于浏览器的应用程序。例如，如图 8 所示，调试请求消息 314 作为调试请求消息 416，被从信道管理器模块 202 传输到被调试应用程序 104。在一个实施例中，如上文所描述的，调试请求消息 314 可以作为对以前的传输的响应被从被调试应用程序 104 传输到信道中继器模块 202。

[0090] 在步骤 1606 中，从第二基于浏览器的应用程序接收调试已执行消息。例如，如图 8 所示，在信道管理器模块 202 处从被调试应用程序 104 接收调试已执行消息 424。

[0091] 在步骤 1608 中，响应于调试请求消息，调试已执行消息被传输到第一基于浏览器的应用程序。例如，如图 8 所示，调试已执行消息 424 被从信道管理器模块 202 作为调试已执行消息 424 传输到调试器应用程序 102。在一个实施例中，调试已执行消息 424 可以作为对从调试器应用程序 102 到信道中继器模块 202 的先前的传输（如调试请求消息 314）的响应，被传输到调试器应用程序 102。

[0092] 注意，如上文所描述的，可以以各种方式执行由调试器应用程序 102 通过信道中继器模块 202 对被调试应用程序 104 的调试。例如，在一个实施例中，可以以多线程方式调试被调试应用程序 104，其中，调试开始模块 404 在第一线程中执行，且应用程序执行模块 410 在第二线程中执行，并且第一和第二线程协作以如上文参考流程图 1500 所描述的类似

的方式执行调试。

[0093] 在另一实施例中，可以使用过程调用来执行流程图 1500，以调试被调试应用程序 104。例如，在这样的实施例中，执行阻止模块 406 可以被配置成将调用插入到被调试应用程序 400 的代码中。应用程序执行模块 410 被配置成在被由调试控制模块 402 接收到的调试请求消息 416（步骤 1502）启用（步骤 1504）之后执行代码（图 15 的步骤 1506）。应用程序执行模块 410 执行代码，直到到达代码中的被插入的调用。在这个点，应用程序执行模块 410 被作为方法调用而被发送到调试控制模块 402 的消息 420 阻止执行进一步的代码，而调试已执行消息 424 被从调试控制模块 402 传输（步骤 1508）。每当接收到另一个调试请求消息 416 时，应用程序执行模块 410 都从由消息 420 调用的方法返回，并执行代码，直到到达下一调用，在该点，代码的执行被消息 420 的下一方法调用阻止，并且调试已执行消息 424 被传送。如果需要，则可以使用单个线程执行以此方式使用过程调用的实施例。在这样的情况下，如上文所描述的，当到达执行代码中的调用时，线程可以能够阻止其本身。这可以例如通过使用 XMLHttpRequest（这是提供用于在客户机和服务器之间传输数据的脚本化客户机功能的 API）的发送方法的非异步形式来传输消息 412 和 424 来完成。发送方法返回将在随后充当接收消息 416。当发送方法被阻止时，应用程序执行模块 410 被阻止（根据需要）等待消息 420 返回，消息 420 又被阻止等待消息 416 到达。

[0094] D. 用于终止调试的示例实施例

[0095] 在一个实施例中，由调试器应用程序 102 对被调试应用程序 104 的调试可以终止。可以以各种方式实现这样的调试的结束。例如，图 17-19 示出了根据示例实施例的用于结束调试的流程图 1700、1800，以及 1900。为了说明，参考图 8 所示出的调试系统 800 描述了图 17-19。基于关于图 17-19，以及图 8 的系统 800 的讨论，其他结构和操作实施例对相关领域技术人员是显而易见的。

[0096] 例如，图 17 示出了根据示例实施例的可以由调试器应用程序 102 执行以终止调试的步骤 1702。在步骤 1702 中，将终止消息传输到信道管理器模块。例如，如图 8 所示，调试器应用程序 102 将终止消息 804 传输到信道管理器模块 202。参考图 3，例如，可以由调试指令传输模块 308 生成终止消息 804。例如，正在调试被调试应用程序 104 的用户可以确定调试已完成，并可以导致调试器应用程序 102（例如，通过与浏览器 112 上的调试器应用程序 102 进行交互）传输终止消息 804。可另选地，可以通过关闭被调试应用程序 104，通过关闭正在运行被调试应用程序 104 的浏览器，或以另一种方式，来传输终止消息 804。

[0097] 图 18 示出了根据示例实施例的在调试终止期间可以在信道管理器模块 202 中执行的流程图 1800。在步骤 1802 中，从第一基于浏览器的应用程序接收终止消息。例如，如图 8 所示，终止消息 804 由信道管理器模块 202 从调试器应用程序 102 接收。

[0098] 在步骤 1804 中，向第二基于浏览器的应用程序传输终止消息。例如，如图 8 所示，结束消息 804 被作为结束消息 806 从信道管理器模块 202 传输到被调试应用程序 104。

[0099] 图 19 示出了根据示例实施例的在调试终止期间可以在被调试应用程序 104 中执行的流程图 1900。在步骤 1902 中，从信道管理器模块接收终止消息。例如，如图 8 所示，被调试应用程序 104 接收终止消息 806。在一个实施例中，响应于被调试应用程序 104 向信道中继器模块 202 传输先前的调试已执行消息 424（例如，在紧邻的先前的调试周期期间），可以从信道中继器模块 202 接收终止消息 806。

[0100] 在步骤 1904 中，终止执行。在一个实施例中，在接收到终止消息 806 之后，被调试应用程序 104 终止。

[0101] E. 示例通信协议实施例

[0102] 注意，如上文所描述的，在各实施例中，可以由调试器应用程序 102 和被调试应用程序 104 使用各种协议来通过双向通信信道 204 进行通信。例如，在一个实施例中，双向通信信道 204 可以被根据 HTTP（超文本传输协议）来配置，由此，通信可以通过双向通信信道 204 使用标准 web 安全机制来执行。使用标准 HTTP 协议允许第一基于浏览器的应用程序调试基于另一个浏览器的应用程序。例如，第一 Silverlight™ 应用程序可以被允许调试第二 Silverlight™ 应用程序，而无需以此方式提升特权。

[0103] 在一个实施例中，可以使用 HTTP 服务器（例如，服务器计算机 108）和标准 HTTP 协议来实现双向通信信道 204。通过向 HTTP 服务器执行 HTTPpost，信道参与者可以通过双向通信信道 204 来发送和接收消息。例如，参考图 8，等待指示 802、开始指示 412、调试请求消息 314、调试已执行消息 424，以及终止消息 804 可以各自被作为 HTTP post 传输到信道管理器模块 202。开始指示 312、调试请求消息 416、调试已执行消息 316、，以及终止消息 806 可以各自由调试器应用程序 102 和被调试应用程序 104 作为对相对应的 HTTP post 的响应，从信道管理器模块 202 传输。在一个实施例中，可以通过使用 HTTPS（超文本传输协议安全）代替 HTTP 来保护双向通信信道 204 的安全。关于授权，在一个实施例中，可以使用标准 HTTP 授权技术，包括 SharedKey 等等。

[0104] 如果如上文所描述的，服务器计算机 108 是调试器应用程序 102 的起点的服务器，并且是被调试应用程序 104 的起点的域，则不需要提升特权。如果不是，则可以由调试器应用程序 102 使用标准跨站脚本技术。被调试应用程序 104 可以使用允许阻止执行的跨站脚本技术。

[0105] 服务器可以通过使用 URI 参数来指定哪一个信道正在被使用，来支持多个同时的信道。在一个实施例中，可以使用此技术，以及 ASP.NET 的 IHttpAsyncHandler 来实现请求处理程序以最小化在信道参与者正在等待消息时消耗的线程的数量。

[0106] 通过双向通信信道 204 传输的消息的格式可以是诸如 XML（可扩展标记语言）文档或其他消息格式类型之类的八位位组的任何序列。

[0107] 如上文所描述的，被调试应用程序 104 被配置成当到达断点或完成步进操作时阻止执行。例如，如上文所描述的，执行阻止模块 406 可以允许这样的阻止。

[0108] IV. 示例低特权调试流水线实施例

[0109] 在一个实施例中，双向通信信道 204 可被配置成低特权调试信道。例如，可以配置低特权调试流水线，该流水线使用双向通信信道 204 来在调试器应用程序 102 和被调试应用程序 104 之间执行通信。在一个实施例中，起点 HTTP 服务器（例如，服务器计算机 108）的域可以通过提供低特权调试流水线来促进调试器应用程序 102 和被调试应用程序之间的通信。

[0110] 例如，图 20 示出了根据示例实施例的可以由服务器计算机 108 执行的步骤 2002。在步骤 2002 中，双向通信信道被配置为低特权调试流水线。图 21A 和 21B 示出了根据此处所公开的主题的其他方面允许低特权调试的系统 2100 的示例。系统 2100 可包括下列各项中的一个或多个 :诸如服务器 2102 之类的一个或多个服务器计算机等等，和 / 或一个或多

个客户机计算机 2118、2120 等等。服务器 2102 是服务器计算机 108 的示例，而客户机计算机 2118 和 2120 是客户机计算机 106 的示例。

[0111] 可以通过将第一客户机进程（例如，浏览器客户机 1 2104）通过第一连接（例如，因特网协议连接 2122）连接到服务器（例如，服务器计算机 2102），将第二客户机进程（例如，浏览器客户机 2 2106）通过第二连接（例如，因特网协议连接 2124）连接到服务器计算机 2102，并在随后从通信信道中移除服务器，如图 21B 所示，允许客户机进程（例如，浏览器客户机 1 2104 和浏览器客户机 2 2106）通过低特权流水线（例如，低特权调试流水线 2114）直接进行通信——代替如在传统的调试模式中那样让第一客户机下载可调试第二客户机的具特权的组件——来建立客户机 - 客户机调试连接。

[0112] 例如，假设因特网用户通过该用户的浏览器（例如，浏览器客户机 12104）导航到服务器 2102 上的网页并下载 JAVA 小程序，某些 Microsoft Silverlight 代码，或通常在浏览器内的客户端上运行的其他应用程序，通常是低特权组件。假设应用程序（JAVA 小程序或 Silverlight 代码）打开第二浏览器（例如，浏览器客户机 2 2106），以使得现在客户机机器具有在它上运行的两个浏览器客户机。假设应用程序具有数行脚本代码在它内部运行。第一浏览器客户端（例如，浏览器客户端 2104）可以通过发布低特权调试流水线（例如，低特权调试流水线 2114）（通过使用低特权调试流水线发布者），来调试下载的代码。第一浏览器客户机通过低特权调试流水线 2114 向后与服务器 2102 进行通信。服务器 2102 可以将低特权调试流水线 2114 转发到第二浏览器客户端 2106。由此，服务器充当客户机计算机 2118 上的第一和第二浏览器之间的流水线管道。服务器 2102 可以在连接之外被优化，以便通信，不是从客户机上的一个浏览器到服务器并返回到客户机上的另一个浏览器，而是在客户机计算机 2118 上的客户机浏览器之间直接进行。即，通信的路由可以被优化，以便通过低特权调试流水线直接在在客户机上运行的两个低特权浏览器之间建立通信，从而建立客户机 - 客户机低特权调试会话。

[0113] 在一个实施例中，可以在调试流水线 2114 中实现双向通信信道 204（包括信道中继器模块 202），以使得调试器应用程序 102 和被调试应用程序 104 能够以低特权方式如此处所描述的进行通信。在 2008 年 12 月 19 日提交的题为“Low Privilege Debugging Pipeline（低特权调试流水线）”的共同受让的、共同待审的美国申请 No. 12/339,111 中描述了适用于本发明的各实施例的低特权调试流水线的进一步的描述和示例，该申请的全部内容通过援引纳入于此。

[0114] V. 其他示例实施例

[0115] 图 22 描绘了其中可以实现本发明的各实施例的计算机 2200 的示例性实现。例如，可以类似于计算机 2200 实现客户机计算机 106（图 1、6 和 7）、服务器计算机 108（图 1 和 6）、客户机计算机 2118（图 21A 和 21B），以及服务器计算机 2102（图 21A 和 21B），并可包括计算机 2200 的一个或多个特征和 / 或替换的特征。计算机 2200 可以是例如诸如个人计算机、移动计算机或工作站之类的常规计算机形式的通用计算设备，或者，计算机 2200 可以是特殊用途的计算设备。此处所提供的对计算机 2200 的描述只是为了说明，并不是限制性的。如相关领域的技术人员所知道的，本发明的各实施例可以在其他类型的计算机系统中实现。

[0116] 如图 22 所示，计算机 2200 包括处理单元 2202、系统存储器 2204，以及将包括系统

存储器 2204 的各种系统组件耦合到处理单元 2202 的总线 2206。系统总线 2206 表示若干类型的总线结构中的任何一种总线结构的一个或多个，包括存储器总线或存储器控制器、外围总线、加速图形端口，以及使用各种总线体系结构中的任何一种的处理器或局部总线。系统存储器 2204 包括只读存储器 (ROM) 2208 和随机存取存储器 (RAM) 2210。基本输入 / 输出系统 2212(BIOS) 存储在 ROM 2208 中。

[0117] 计算机 2200 还具有一个或多个以下驱动器 :用于读写硬盘的硬盘驱动器 2214、用于读或写可移动磁盘 2218 的磁盘驱动器 2216、以及用于读或写诸如 CD ROM、DVD ROM 或其他光介质之类的可移动光盘 2222 的光盘驱动器 2220。硬盘驱动器 2214、磁盘驱动器 2216，以及光驱动器 2220 分别通过硬盘驱动器接口 2224、磁盘驱动器接口 2226，以及光学驱动器接口 2228 连接到系统总线 2206。驱动器以及它们相关联的计算机可读介质为计算机提供了对计算机可读指令、数据结构、程序模块，及其他数据的非易失存储器。虽然描述了硬盘、可移动磁盘和可移动光盘，但是，也可以使用诸如闪存卡、数字视频盘、随机存取存储器 (RAM)、只读存储器 (ROM) 等等之类的其他类型的计算

[0118] 机可读介质来存储数据。

[0119] 数个程序模块可被存储在硬盘、磁盘、光盘、ROM，或 RAM 上。这些程序包括操作系统 2230、一个或多个应用程序 2232、其他程序模块 2234，以及程序数据 2236。应用程序 2232 或程序模块 2234 可包括，例如，用于实现浏览器 112、调试器应用程序 102、被调试应用程序 104、信道中继器模块 202、信道建立器模块 302、调试启动模块 304、调试器模块 306、调试指令传输模块 308、调试响应接收器模块 310、调试控制模块 402、调试开始模块 404、执行阻止模块 406、调试响应传输模块 408、应用程序执行模块 410、调试流水线 2114、浏览器客户机 2104、浏览器客户机 2106、流程图 500、900、1000、1100、1200、1300、1400、1500、1600、1700、1800、1900 和 / 或 2000(包括流程图 500、900、1000、1100、1200、1300、1400、1500、1600、1700、1800、1900 和 / 或 2000 的任何步骤)，和 / 或如上文所描述的任何进一步实施例的计算机程序逻辑 (和 / 或其他合适的装置)。

[0120] 用户可以通过诸如键盘 2238 和定点设备 2240 之类的输入设备向计算机 2200 中输入命令和信息。其他输入设备 (未示出) 可包括话筒、操纵杆、游戏手柄、圆盘式卫星天线、扫描仪等。这些及其他输入设备常常通过耦合到总线 2206 的串行端口接口 2242 连接到处理单元 2202，但是，也可以通过其他接口，诸如并行端口、游戏端口、通用串行总线 (USB) 端口，来进行连接。

[0121] 监视器 2244 或其他类型的显示设备也可以经由诸如视频适配器 2246 之类的接口来连接到系统总线 2206。除了监视器之外，计算机 2200 还可包括其他外围输出设备 (未示出)，如扬声器和打印机。

[0122] 计算机 2200 通过网络接口或适配器 2250、调制解调器 2252、或用于通过网络建立通信的其他装置连接到网络 2248(例如，因特网)。调制解调器 2252(可以是内置的或外置的)，通过串行端口接口 2242 连接到系统总线 2206。

[0123] 如此处所使用的，术语“计算机程序介质”和“计算机可读介质”被用来泛指诸如与硬盘驱动器 2214 相关联的硬盘、可移动磁盘 2218、可移动光盘 2222，以及诸如闪存卡、数字视频盘、随机存取存储器 (RAM)、只读存储器 (ROM) 等等之类的其他介质。

[0124] 如上文所指示的，计算机程序和模块 (包括应用程序 2232 及其他程序模块 2234)

可以存储在硬盘、磁盘、光盘、ROM 或 RAM 上。这样的计算机程序也可以通过网络接口 2250 或串行端口接口 2242 来接收。这样的计算机程序，当由应用程序执行或加载时，使得计算机 2200 能实现此处所讨论的本发明的特征。相应地，这样的计算机程序表示计算机 2200 的控制器。

[0125] 本发明还涉及包括存储在任何计算机可使用介质上的软件的计算机程序产品。这样的软件，当在一个或多个数据处理设备中执行时，使数据处理设备如此处所描述的那样操作。本发明的各实施例使用现在已知的或将来已知的任何计算机可使用或计算机可读介质。计算机可读介质的示例包括，但不仅限于，诸如 RAM、硬盘驱动器、软盘、CD ROM、DVD ROM、zip 磁盘、磁带、磁存储设备、光存储设备、MEM（存储器）、基于纳米技术的存储设备等等之类的存储设备。

[0126] VI. 结论

[0127] 尽管上文描述了本发明的各实施例，但是，应该理解，它们只是作为示例来呈现的，而不作为限制。那些精通有关技术的人员将理解，在不偏离如所附权利要求书所定义的本发明的精神和范围的情况下，可以在形式和细节方面进行各种修改。因此，本发明的范围不应该受到上述示例性实施例的任一个的限制，而只应根据下面的权利要求和它们的等效内容进行定义。

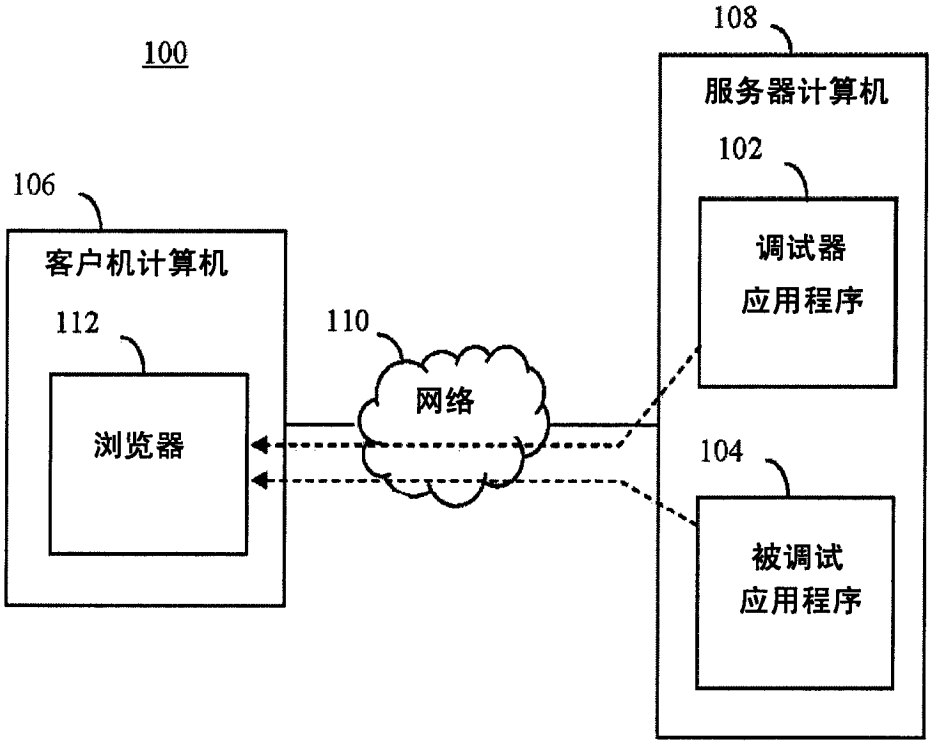

100
108
服务器计算机
102
调试器
应用程序
106
客户机计算机
112
浏览器
110
网络
104
被调试
应用程序

图 1

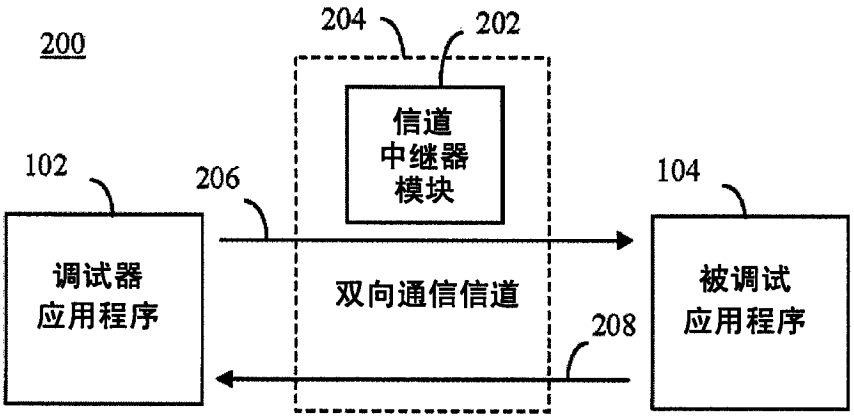

200
204
202
信道
中继器
模块
102
206
104
调试器
应用程序
双向通信信道
208
被调试
应用程序

图 2

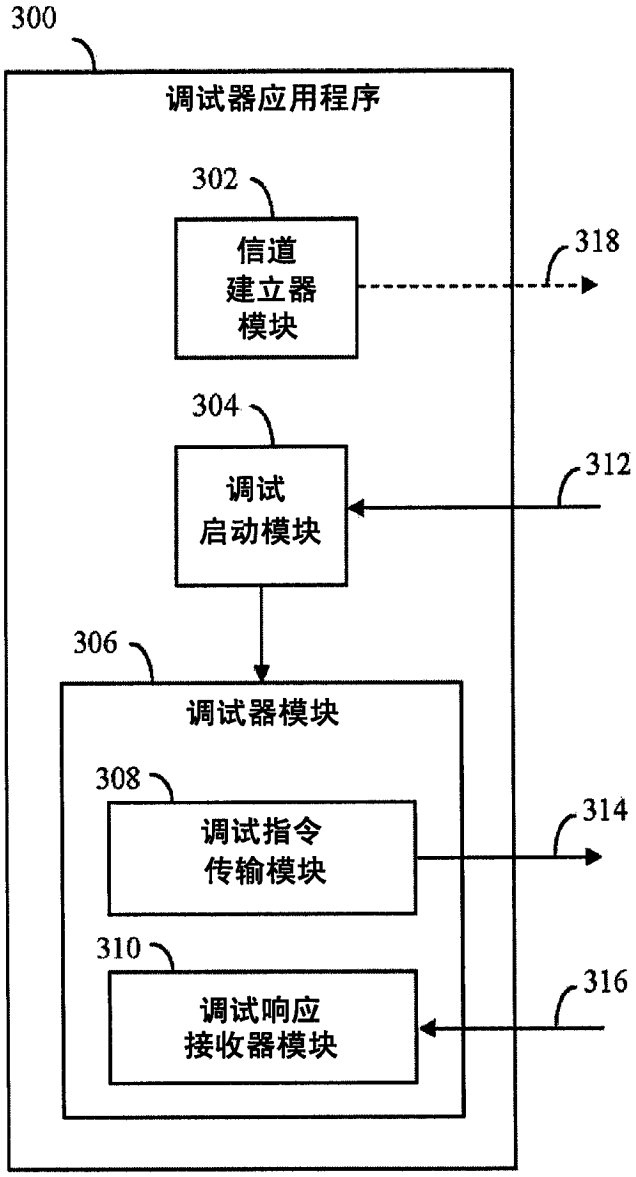

300
调试器应用程序
302
信道
建立器
模块
318
304
调试
启动模块
312
306
调试器模块
308
调试指令
传输模块
314
310
调试响应
接收器模块
316

图 3

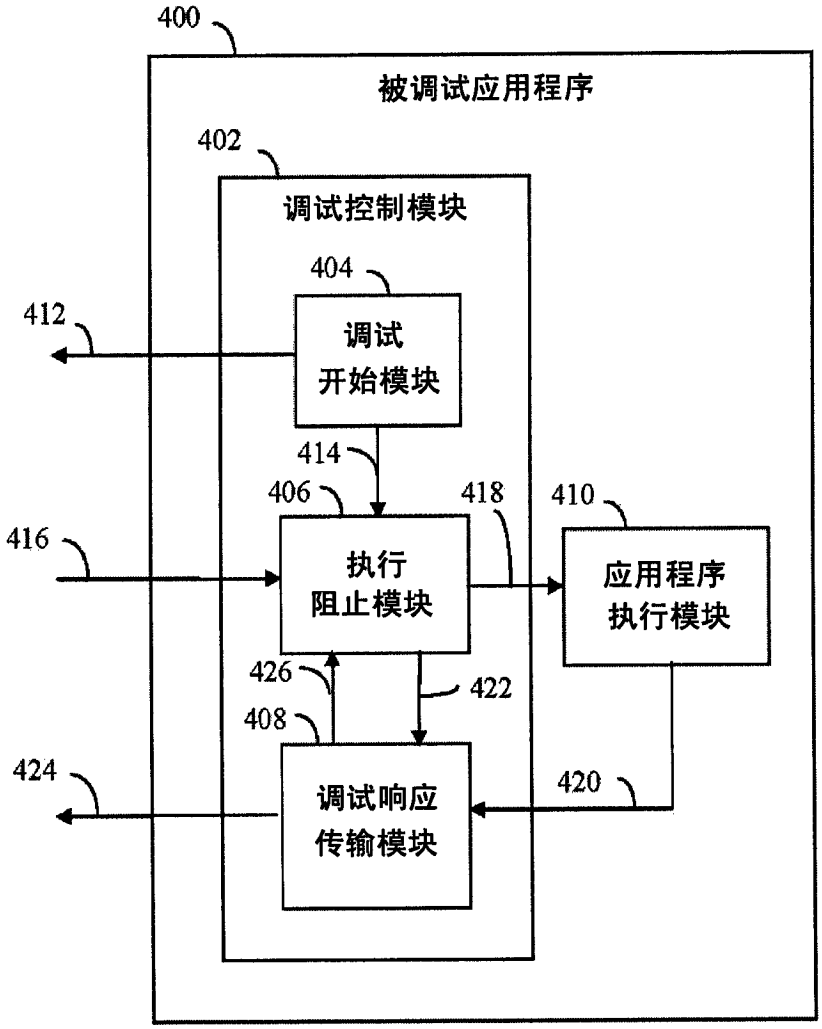

400
被调试应用程序
402
调试控制模块
404
412
调试
开始模块
414
406
418
410
416
执行
阻止模块
应用程序
执行模块
426
422
408
424
420
调试响应
传输模块

图 4

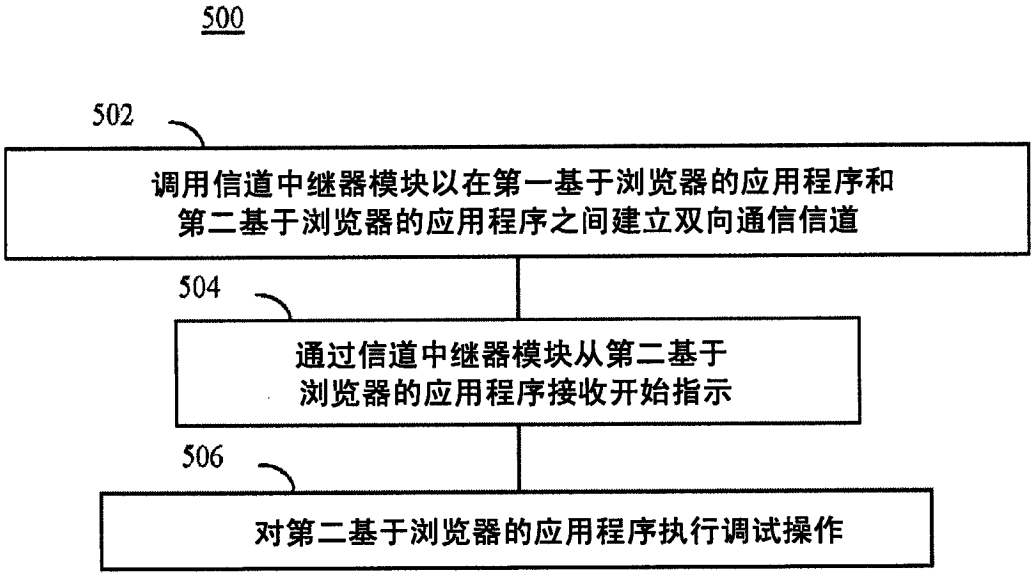

500
502
调用信道中继器模块以在第一基于浏览器的应用程序和第二基于浏览器的应用程序之间建立双向通信信道
504
通过信道中继器模块从第二基于浏览器的应用程序接收开始指示
506
对第二基于浏览器的应用程序执行调试操作

图 5

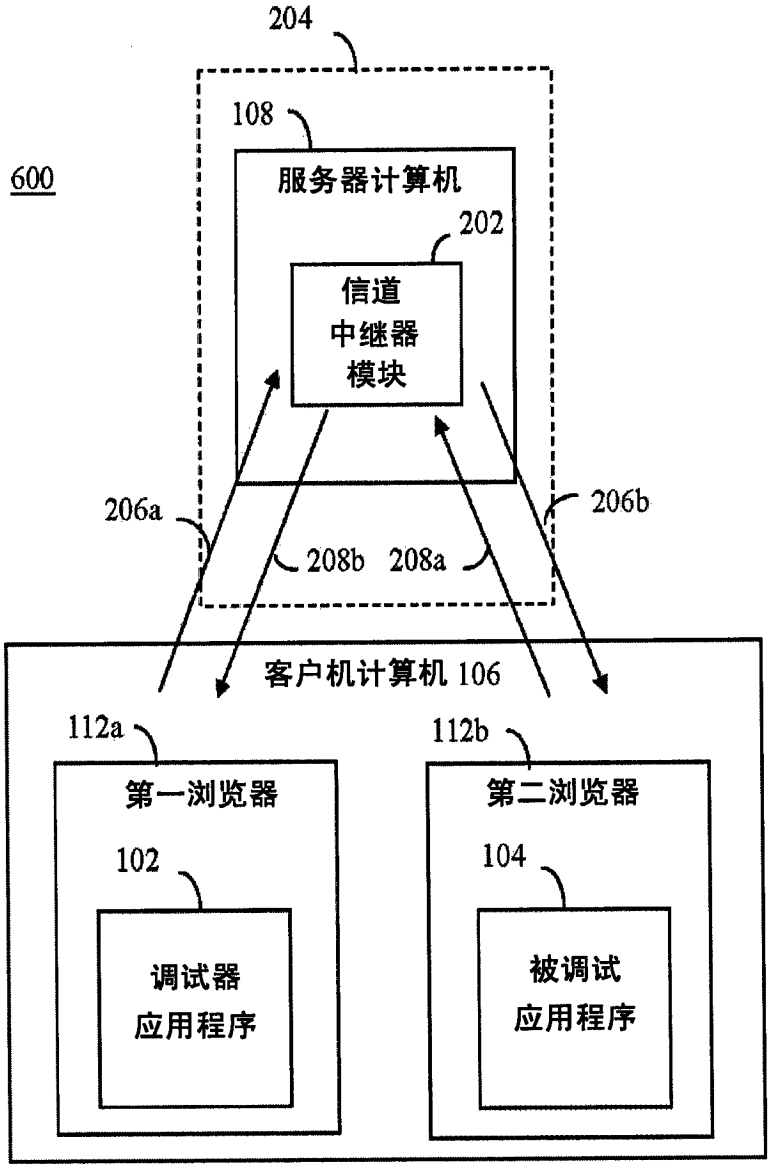

204
108
600
服务器计算机
202
信道
中继器
模块
206a
206b
208b
208a
客户机计算机 106
112a
112b
第一浏览器
第二浏览器
102
104
调试器
应用程序
被调试
应用程序

图 6

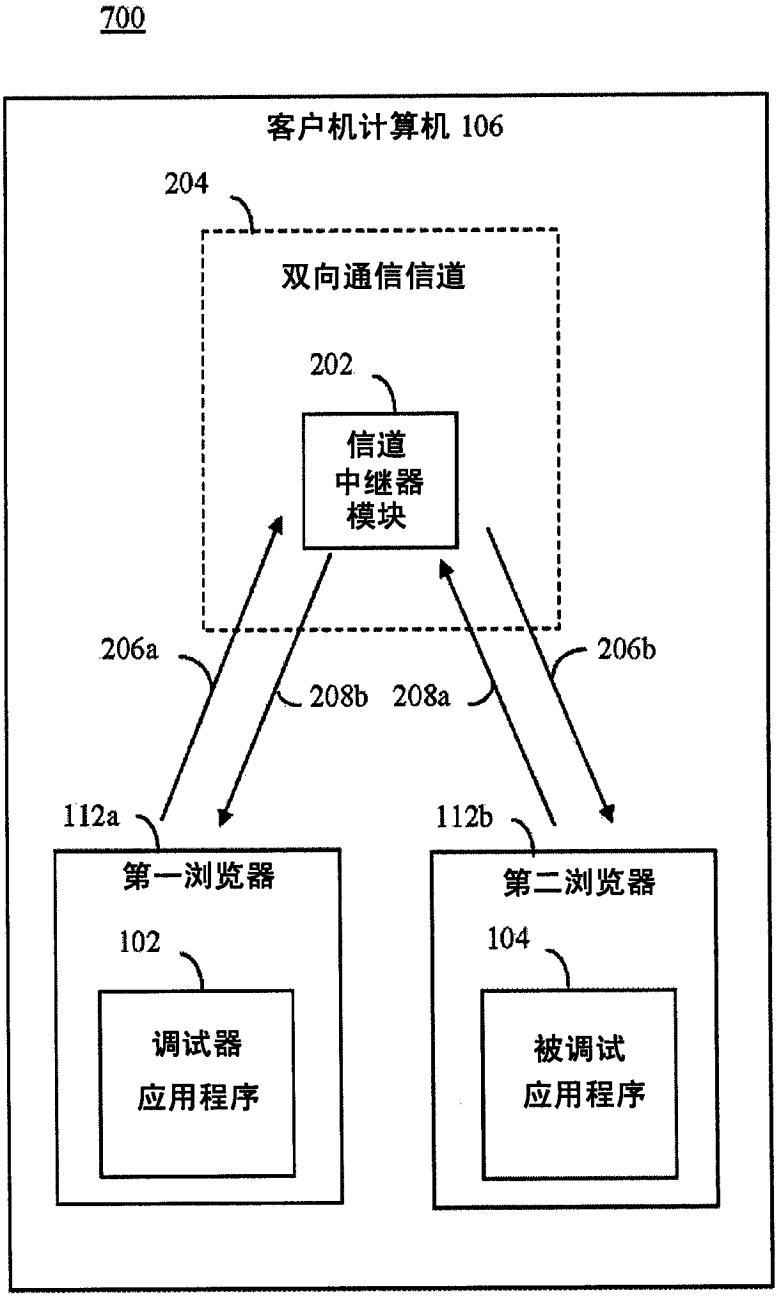

700
客户机计算机 106
204
双向通信信道
202
信道
中继器
模块
206a
208b
208a
206b
112a
112b
第一浏览器
102
调试器
应用程序
第二浏览器
104
被调试
应用程序

图 7

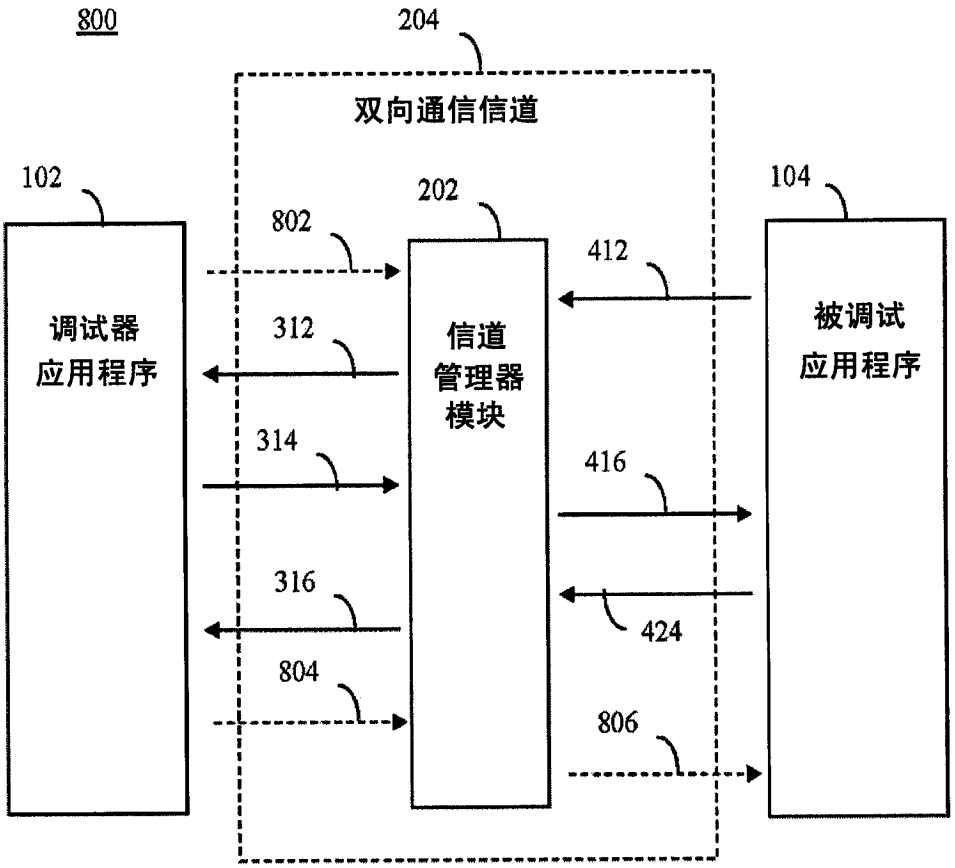

800
204
双向通信信道
102
202
104
802
412
调试器
应用程序
312
信道
管理器
模块
被调试
应用程序
314
416
316
424
804
806

图 8

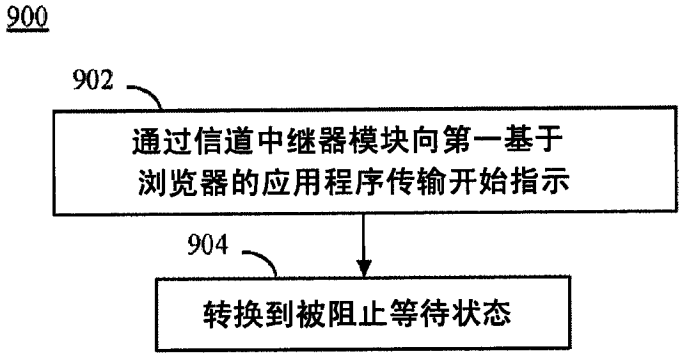

900
902
通过信道中继器模块向第一基于
浏览器的应用程序传输开始指示
904
转换到被阻止等待状态

图 9

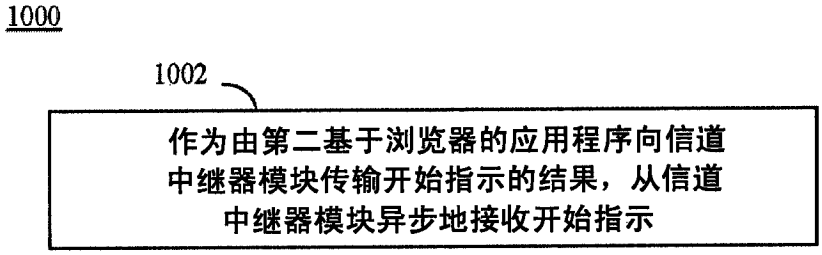

1000
1002
作为由第二基于浏览器的应用程序向信道中继器模块传输开始指示的结果，从信道中继器模块异步地接收开始指示

图 10

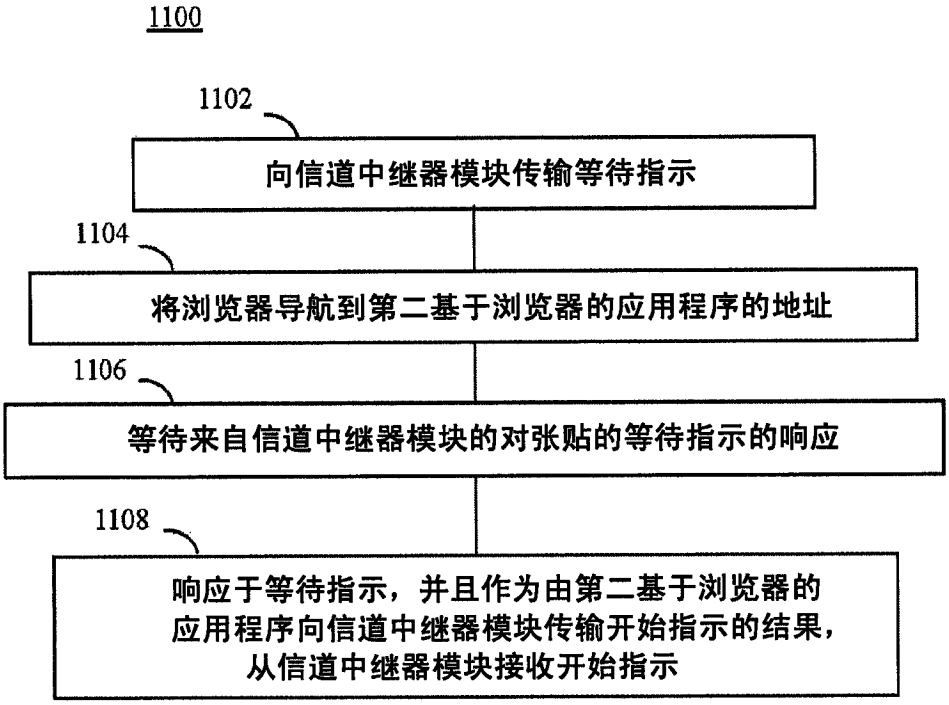

1100
1102
向信道中继器模块传输等待指示
1104
将浏览器导航到第二基于浏览器的应用程序的地址
1106
等待来自信道中继器模块的对张贴的等待指示的响应
1108
响应于等待指示，并且作为由第二基于浏览器的应用程序向信道中继器模块传输开始指示的结果，从信道中继器模块接收开始指示

图 11

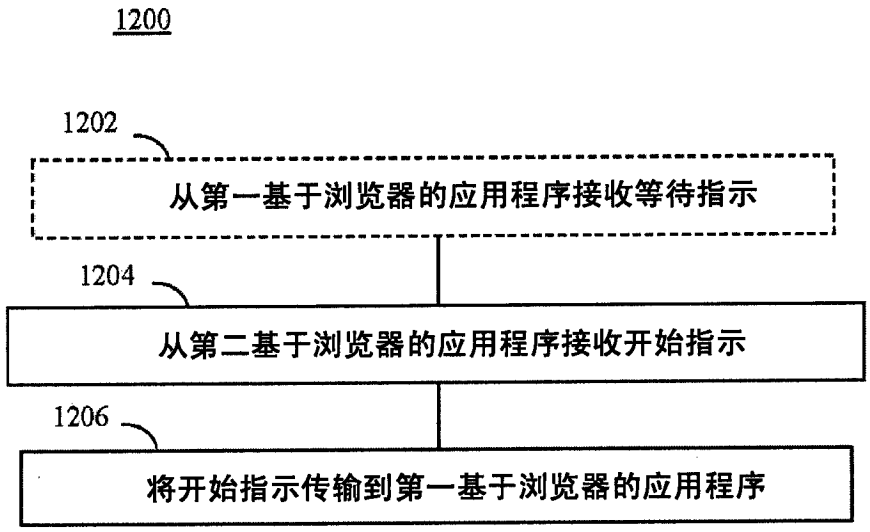

1200
1202
从第一基于浏览器的应用程序接收等待指示
1204
从第二基于浏览器的应用程序接收开始指示
1206
将开始指示传输到第一基于浏览器的应用程序

图 12

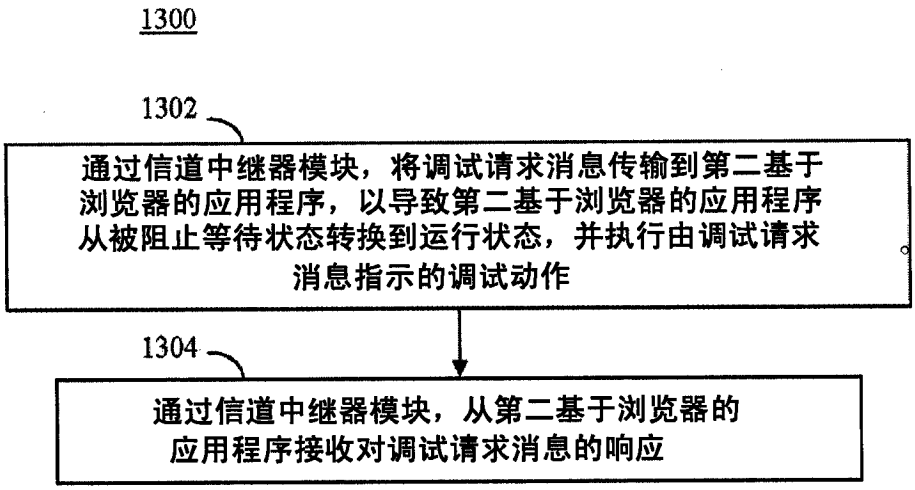

1300
1302
通过信道中继器模块，将调试请求消息传输到第二基于浏览器的应用程序，以导致第二基于浏览器的应用程序从被阻止等待状态转换到运行状态，并执行由调试请求消息指示的调试动作
1304
通过信道中继器模块，从第二基于浏览器的应用程序接收对调试请求消息的响应

图 13

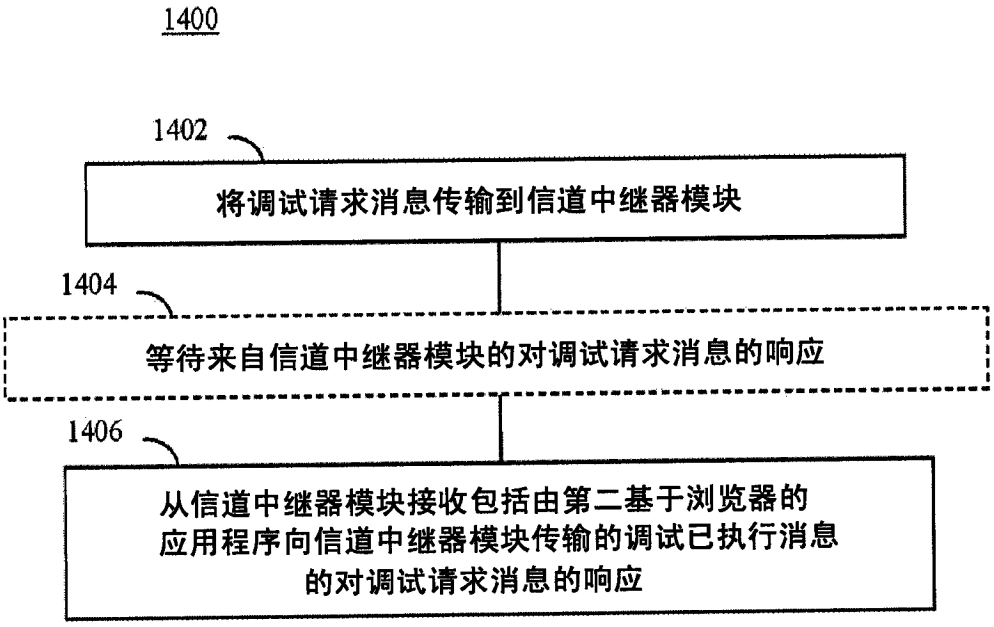

1400
1402
将调试请求消息传输到信道中继器模块
1404
等待来自信道中继器模块的对调试请求消息的响应
1406
从信道中继器模块接收包括由第二基于浏览器的应用程序向信道中继器模块传输的调试已执行消息的对调试请求消息的响应

图 14

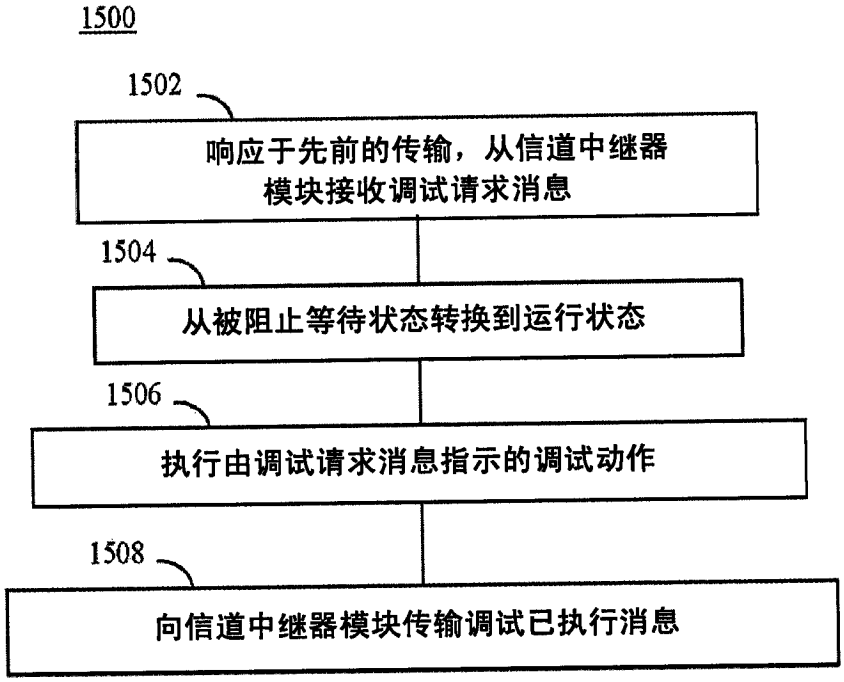

1500
1502
响应于先前的传输，从信道中继器模块接收调试请求消息
1504
从被阻止等待状态转换到运行状态
1506
执行由调试请求消息指示的调试动作
1508
向信道中继器模块传输调试已执行消息

图 15

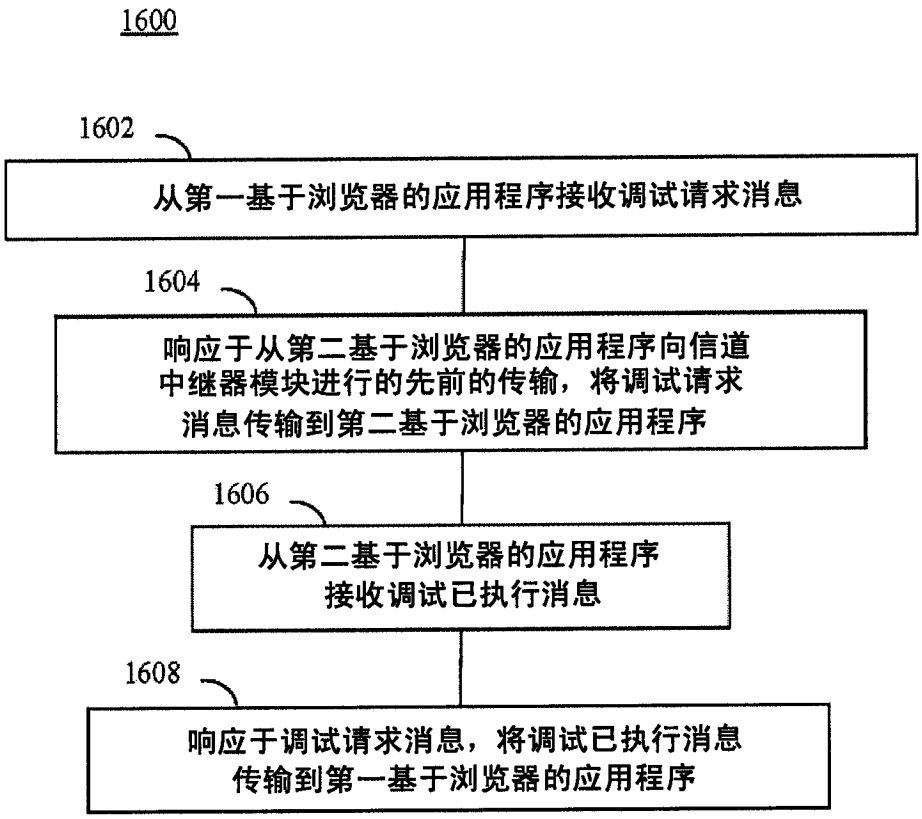

1600
1602
从第一基于浏览器的应用程序接收调试请求消息
1604
响应于从第二基于浏览器的应用程序向信道中继器模块进行的先前的传输，将调试请求消息传输到第二基于浏览器的应用程序
1606
从第二基于浏览器的应用程序接收调试已执行消息
1608
响应于调试请求消息，将调试已执行消息传输到第一基于浏览器的应用程序

图 16

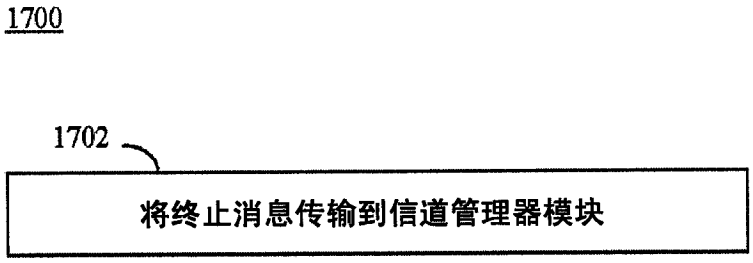

1700
1702
将终止消息传输到信道管理器模块

图 17

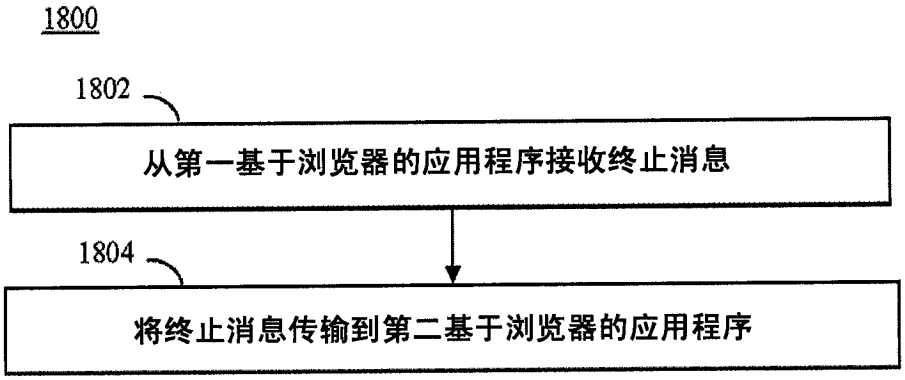

1800
1802
从第一基于浏览器的应用程序接收终止消息
1804
将终止消息传输到第二基于浏览器的应用程序

图 18

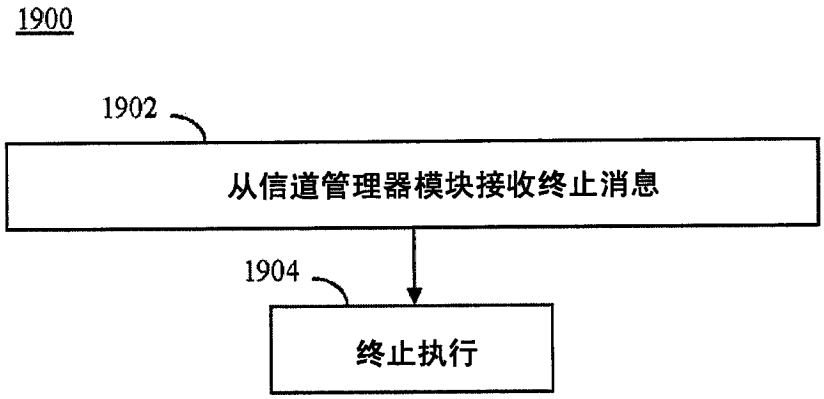

1900
1902
从信道管理器模块接收终止消息
1904
终止执行

图 19

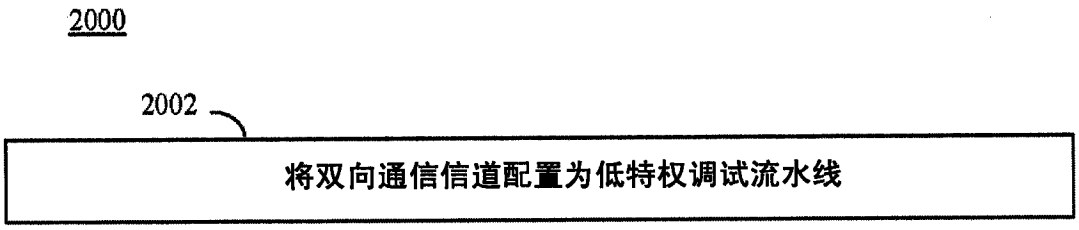

2000
2002
将双向通信信道配置为低特权调试流水线

图 20

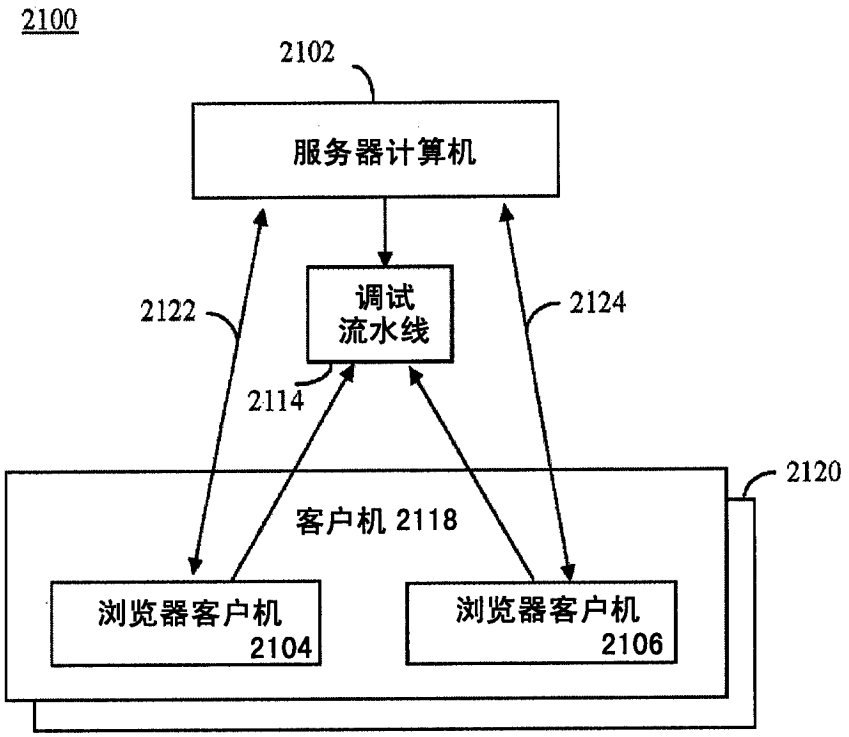

2100
2102
服务器计算机
调试
流水线
2122
2124
2114
2120
客户机 2118
浏览器客户机
2104
浏览器客户机
2106

图 21A

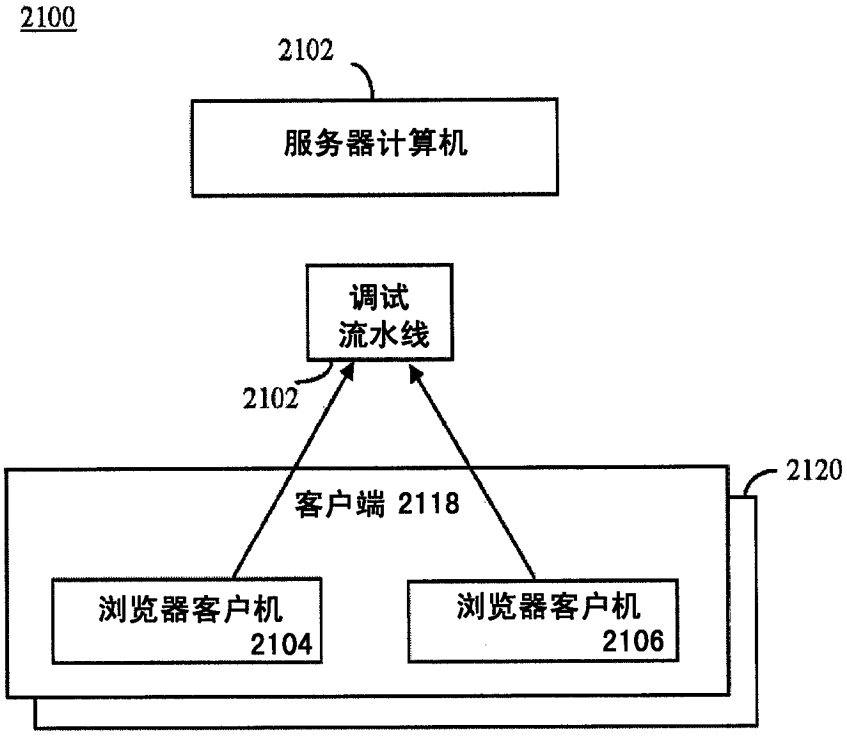

2100
2102
服务器计算机
调试
流水线
2102
2120
客户端 2118
浏览器客户机
2104
浏览器客户机
2106

图 21B

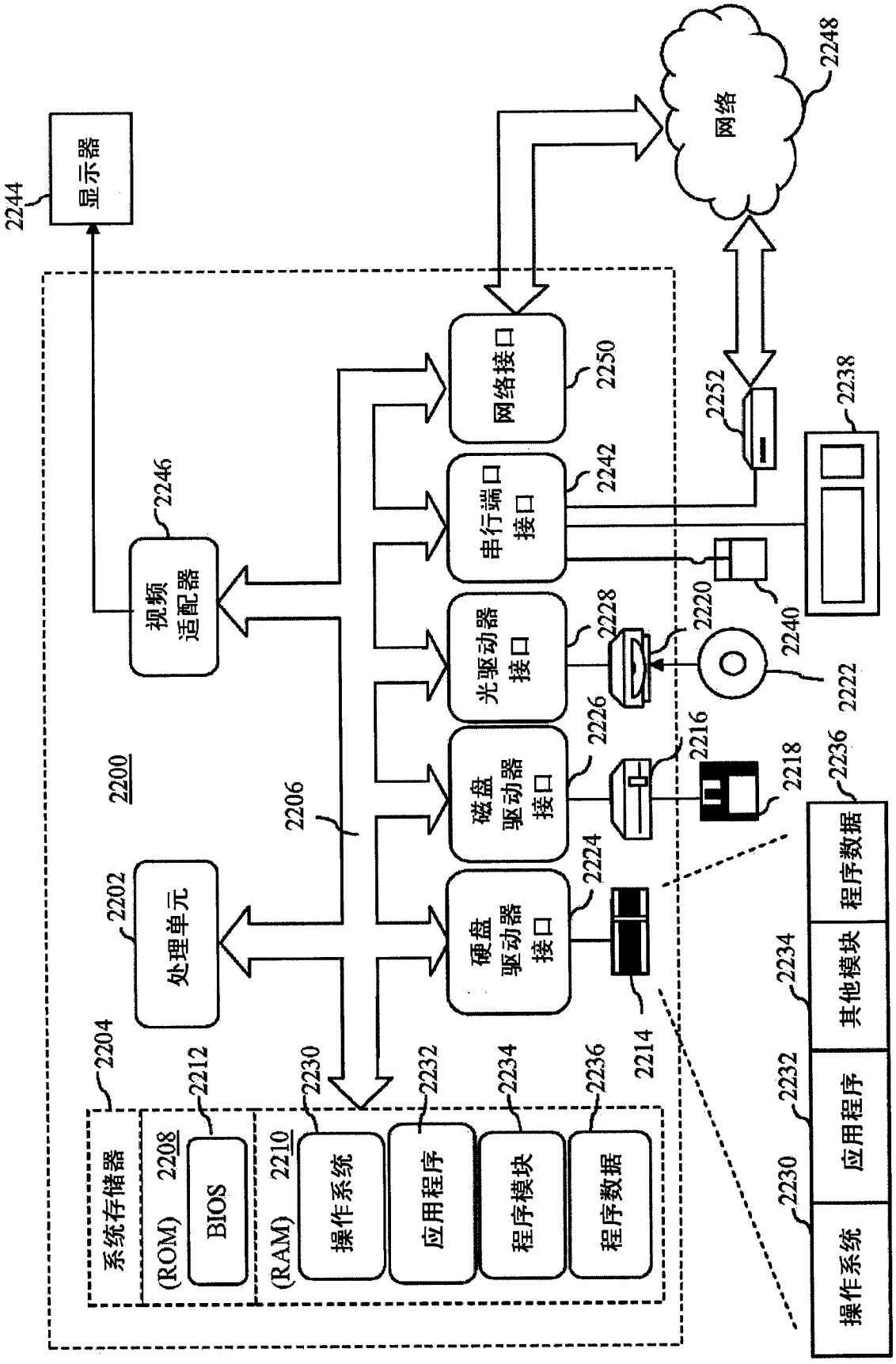

显示器
2244
网络
2248
网络接口
2250
串行端口接口
2242
光驱动器接口
2228
磁盘驱动器接口
2226
硬盘驱动器接口
2224
视频适配器
2246
处理单元
2202
2200
2206
2204
系统存储器
(ROM) 2208
BIOS
2212
(RAM) 2210
操作系统
2230
应用程序
2232
程序模块
2234
程序数据
2236
2214
2252
2238
2220
2240
2222
2216
2218
2236
操作系统
2230
应用程序
2232
其他模块
2234
程序数据

图 22