(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 112534412 A
(43)申请公布日 2021.03.19

(21)申请号 201980052298.1
(22)申请日 2019.03.20
(30)优先权数据
16/055,978 2018.08.06 US
(85)PCT国际申请进入国家阶段日
2021.02.05
(86)PCT国际申请的申请数据
PCT/US2019/023264 2019.03.20
(87)PCT国际申请的公布数据
WO2020/033012 EN 2020.02.13
(71)申请人 甲骨文国际公司
地址 美国加利福尼亚
(72)发明人 T·拉希里 J·R·洛埃扎 G·F·斯沃特 J·卡普 A·潘特 H·基穆拉
(74)专利代理机构 中国贸促会专利商标事务所有限公司 11038
代理人 刘玉洁
(51)Int.Cl.
G06F 11/20(2006.01)

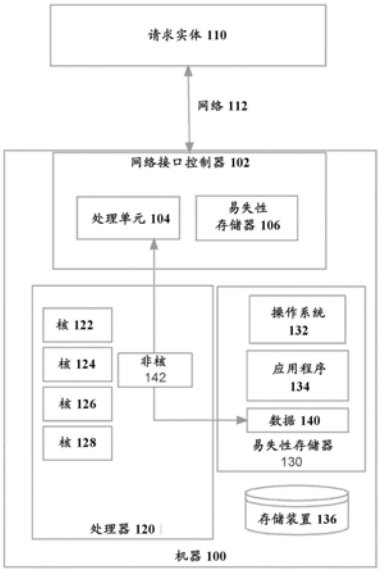
权利要求书3页 说明书9页 附图4页

(54)发明名称

单侧可靠的远程直接存储器操作

(57)摘要

提供了技术以允许不完全起作用的机器远程执行更复杂的操作。可以由已经经历硬件和/或软件错误的机器可靠地执行的操作在本文中被称为远程直接存储器操作或“RDMO”。与通常涉及非常简单的操作(诸如从远程机器的存储器中检索单个值)的RDMA不同,RDMO可能是任意复杂的。当应用程序与之交互的远程系统上存在软件故障或小差错时,本文所述的技术可以帮助应用程序运行而不中断。



CN 112534412 A

1.一种在远程计算设备上执行操作的方法,包括:

在所述远程计算设备上实现:

能够执行所述操作的第一执行候选者;以及

能够执行所述操作的第二执行候选者;

其中所述第一执行候选者和所述第二执行候选者能够访问所述远程计算设备的共享存储器;

其中所述操作需要访问所述共享存储器中的数据;

其中所述第二执行候选者与和所述第一执行候选者不同的可靠性域相关联;

从在相对于所述远程计算设备远程的特定计算设备上执行的请求实体接收执行所述操作的第一请求;

响应于所述第一请求,尝试使用所述第一执行候选者来执行所述操作;以及

响应于所述第一执行候选者未成功执行所述操作,尝试使用所述第二执行候选者来执行所述操作。

2.如权利要求1所述的方法,其中尝试使用所述第二执行候选者来执行所述操作包括尝试使用所述第二执行候选者执行操作,而不向所述请求实体通知所述第一执行候选者执行所述操作失败。

3.如权利要求1所述的方法,其中尝试使用所述第二执行候选者来执行所述操作包括:

向所述请求实体通知所述第一执行候选者执行所述操作失败;

从所述请求实体接收执行所述操作的第二请求;以及

响应于所述第二请求,尝试使用所述第二执行候选者来执行所述操作。

4.如权利要求1所述的方法,其中所述操作涉及在持久性存储设备上读取和/或写入数据,所述持久性存储设备能够被所述远程计算设备直接访问而不能被所述特定计算设备直接访问。

5.如权利要求1所述的方法,其中:

所述第一执行候选者是运行在所述远程计算设备的一个或多个处理器上的应用程序;以及

所述第二执行候选者在所述远程计算设备的网络接口控制器中实现。

6.如权利要求5所述的方法,其中所述第二执行候选者在所述网络接口控制器的固件中实现。

7.如权利要求5所述的方法,其中所述第二执行候选者是在所述网络接口控制器内的一个或多个处理器上执行的软件。

8.如权利要求5所述的方法,其中所述第二执行候选者是通过解释在提供给所述网络接口控制器的数据中指定的指令来执行所述操作的解释器。

9.如权利要求1所述的方法,其中所述第一执行候选者和所述第二执行候选者中的一个在所述远程计算设备上执行的操作系统内实现。

10.如权利要求1所述的方法,其中所述第一执行候选者和所述第二执行候选者中的一个在所述远程计算设备上的特权域内实现。

11.如权利要求1所述的方法,其中:

所述第一执行候选者在所述远程计算设备中的处理器的第一组一个或多个核上执行;

所述第二执行候选者在所述处理器的第二组一个或多个核上执行;以及

所述第二组一个或多个核的成员与所述第一组一个或多个核的成员不同。

12.如权利要求11所述的方法,其中所述第二组一个或多个核包括不属于所述第一组一个或多个核的至少一个核。

13.如权利要求1所述的方法,其中所述第一执行候选者和所述第二执行候选者中的一个包括解释器,所述解释器解释指令,所述指令在被解释时使得执行所述操作。

14.如权利要求13所述的方法,其中所述解释器在网络接口控制器上实现,所述远程计算设备通过网络通过所述网络接口控制器与所述特定计算设备进行通信。

15.一种在远程计算设备上执行操作的方法,包括:

在所述远程计算设备上执行:

能够执行所述操作的第一执行候选者;以及

能够执行所述操作的第二执行候选者;

其中所述第一执行候选者和所述第二执行候选者能访问所述远程计算设备的共享存储器;

其中所述操作需要多次访问所述共享存储器中的数据;

其中所述第二执行候选者与和所述第一执行候选者不同的可靠性域相关联;以及

同时请求所述第一执行候选者和所述第二执行候选者执行所述操作。

16.如权利要求15所述的方法,其中:

所述第一执行候选者在网络接口控制器上实现,所述远程计算设备通过网络通过所述网络接口控制器进行通信;以及

所述第二执行候选者由所述远程计算设备的一个或多个处理器执行。

17.一种在远程计算设备上执行操作的方法,包括:

在所述远程计算设备上执行能够执行所述操作的应用程序;

其中所述远程计算设备包括网络接口控制器,所述网络接口控制器具有能够解释指令的执行候选者,所述指令在被解释时使得执行所述操作;

其中所述应用程序和所述执行候选者能访问所述远程计算设备的共享存储器;

其中所述操作需要多次访问所述共享存储器中的数据;

基于一个或多个因素,在相对于所述远程计算设备远程的特定计算设备上执行的请求实体从所述应用程序和所述执行候选者中选择目标;

响应于所述应用程序被选择为所述目标,所述请求实体向所述应用程序发送执行所述操作的请求;以及

响应于所述执行候选者被选择为所述目标,所述请求实体使所述执行候选者解释指令,所述指令在由所述执行候选者解释时使得执行所述操作。

18.如权利要求17所述的方法,还包括从所述特定计算设备向所述远程计算设备发送指定所述指令的数据,所述指令在由所述执行候选者解释时使得执行所述操作。

19.如权利要求17所述的方法,其中所述解释器是JAVA虚拟机,并且所述指令是字节码。

20.一个或多个非暂时性计算机可读介质,所述一个或多个非暂时性计算机可读介质存储指令,所述指令在被执行时使得如权利要求1-19中的任一项所述的方法被执行。

21.一种系统，包括被配置为执行如权利要求1-19中的任一项所述的方法的一个或多个计算设备。

单侧可靠的远程直接存储器操作

技术领域

[0001] 本发明涉及计算机系统,并且更具体地,涉及用于增加远程访问功能的可用性的技术。

背景技术

[0002] 提高服务可用性的一种方式是以使得即使当服务的组件中的一个或多个组件出现故障时该服务仍继续正常起作用的方式来设计服务。例如,2017年5月26日提交的第15/606,322号美国专利申请(该美国专利申请通过此引用并入本文中)描述了用于实现如下操作的技术:使得请求实体能够从执行数据库服务器实例的远程服务器机器的易失性存储器中检索由数据库服务器实例管理的数据,而在检索操作中不涉及数据库服务器实例。

[0003] 因为检索不涉及数据库服务器实例,所以即使当数据库服务器实例(或主机服务器机器本身)已停滞(stall)或变得无响应时,检索操作也可以成功。除了增加可用性之外,直接检索数据将通常比通过与数据库服务器实例的常规交互来检索相同信息要更快且更高效。

[0004] 为了从远程机器中检索数据库命令中指定的“目标数据”而不涉及远程数据库服务器实例,请求实体首先使用远程直接存储器访问(RDMA)来访问关于目标数据驻留在服务器机器中的位置的信息。基于这样的目标位置信息,请求实体使用RDMA来从主机服务器机器中检索目标数据而不涉及数据库服务器实例。由请求实体发出的RDMA读取(数据检索操作)是单方的操作,并不需要主机服务器机器(RDBMS服务器)上的CPU中断或OS内核参与。

[0005] RDMA技术对于仅涉及从崩溃的服务器机器的易失性存储器中检索数据的操作很好地工作。然而,期望即使当故障的组件负责执行比仅存储器访问更复杂的操作时,也提供高可用性。为了满足此需求,一些系统提供了一组有限的“动词”,用于经由网络接口控制器执行远程操作,诸如存储器访问和原子操作(测试和设置、比较和交换)。只要系统上电并且NIC能访问主机存储器,这些操作就可以完成。然而,它们支持的操作类型是有限的。

[0006] 通常,通过对在远程机器上运行的应用程序进行远程过程调用(RPC),来对驻留在远程机器的存储器中的数据执行更复杂的操作。例如,期望存储在数据库中的一组数字的平均值的数据库客户端可以对管理数据库的远程数据库服务器实例进行RPC。响应于RPC,远程数据库服务器实例读取该组数字,计算平均值,然后将平均值发送回数据库客户端。

[0007] 在此示例中,如果远程数据库服务器发生故障,则平均值计算操作失败。然而,可以有可能使用RDMA来从远程服务器中检索组中的每个数字。一旦请求实体检索到该组数字中的每个数字,则请求实体可以执行平均值计算操作。然而,使用RDMA检索组中的每个数字并且然后在本地执行计算,比让远程服务器上的应用程序检索数据并执行平均值计算操作的效率低得多。因此,期望扩展当远程服务器不完全起作用(fully functional)时继续从远程服务器可用的操作的范围。

[0008] 本节中描述的方法是可以采用的方法,但不一定是先前已经设想或采用的方法。因此,除非另有说明,否则不应仅由于本节中所述的任何方法包括在本节中而将其作为现

有技术。

附图说明

[0009] 在图中：

[0010] 图1是根据实施例的与包括请求的操作的若干执行候选者的远程机器进行交互的请求实体的框图，其中每个执行候选者在不同的可靠性域中实现；

[0011] 图2是示出了根据实施例的后备执行候选者的使用的流程图；

[0012] 图3是示出了根据实施例的并行执行候选者的使用的流程图；以及

[0013] 图4是可用于实现本文描述的技术的计算机系统的框图。

具体实施方式

[0014] 在下面的描述中，出于解释的目的，许多具体细节被阐述以提供对本发明的透彻理解。然而，将明显的是，可以在没有这些具体细节的情况下实践本发明。在其他实例中，众所周知的结构和设备以框图形式示出，以避免不必要地模糊本发明。

[0015] 总体概述

[0016] 本文描述了用于允许不完全起作用的机器远程执行更复杂的操作的技术。可以由已经经历硬件和/或软件错误的机器可靠地执行的操作在本文中被称为远程直接存储器操作或“RDMO”。与通常涉及非常简单的操作(诸如从远程机器的存储器中检索单个值)的RDMA不同，RDMO可能是任意复杂的。例如，RDMO可以使远程机器计算一组数字的平均值，其中数字驻留在远程机器的存储器中。当应用程序与之交互的远程系统上存在软件故障或小差错(glitches)时，本文所述的技术可以帮助应用程序运行而不中断。

[0017] 执行候选者和可靠性域

[0018] 根据一个实施例，在单个机器内，提供了多个实体来执行相同的RDMO。在给定机器内能够执行特定RDMO的实体在本文中被称为RDMO的“执行候选者”。

[0019] 根据一个实施例，尽管在相同机器中，但是RDMO的执行候选者属于单独的可靠性域。执行候选者的“可靠性域”通常是指在机器上必须正确起作用以使执行候选者能够正确执行RDMO的软件/硬件。如果执行候选者中的一个能够正确执行RDMO，而软件和/或硬件错误已使另一个执行候选者无法正确执行RDMO，则这两个执行候选者属于不同的可靠性域。

[0020] 因为RDMO的多个执行候选者属于不同的可靠性域，所以执行候选者中的一个有可能在机器内的硬件/软件错误阻止机器中的其他执行候选者执行RDMO的时间段期间执行RDMO。多个执行候选者可用于特定RDMO的事实增加了RDMO在被未驻留在机器上的请求实体请求时将成功的可能性。

[0021] 增加RDMO的可用性

[0022] 当远程服务器正在执行用于特定的RDMO的多个执行候选者，并且这些执行候选者来自不同的可靠性域时，RDMO的可用性增加。例如，在一个实施例中，当请求实体请求特定的RDMO时，进行使用机器上的第一执行候选者来执行特定的RDMO的尝试。如果第一执行候选者无法执行RDMO，则进行使用机器上的第二执行候选者来执行特定的RDMO的尝试。此过程可以继续，直到特定的RDMO成功，或者已经尝试过所有执行候选者为止。

[0023] 在替代实施例中，当请求实体请求特定的RDMO时，可以由执行候选者中的两个或

更多个执行候选者同时尝试特定的RDMO。如果任何执行候选者成功，则向请求实体报告特定的RDMO成功。

[0024] 在一个实施例中，RDMO的执行候选者可以是已经被动态地编程以执行RDMO的实体。例如，机器的网络接口控制器中的计算单元可以执行解释器。响应于确定应该在网络控制器中而不是通过机器上运行的应用程序来执行特定的RDMO，包括用于执行特定的RDMO的指令的数据可以被发送到解释器。响应于在网络控制器处解释那些指令，即使应用程序本身可能已经崩溃，特定的RDMO也可以被执行。

[0025] 可靠性域

[0026] 如上所述，执行候选者的“可靠性域”通常是指必须正确起作用以使执行候选者成功执行RDMO的软件/硬件。图1是包括机器100的系统的框图，其中RDMO的执行候选者驻留在多个不同的可靠性域中。具体地，请求实体110可以通过经网络112向机器100发送请求来请求特定的RDMO。机器100包括具有一个或多个计算单元(被表示为处理单元104)的网络接口控制器102，该网络接口控制器102正在执行加载到其本地易失性存储器106中的固件和/或软件。

[0027] 机器100包括执行操作系统132和任何数量的应用程序(诸如应用程序134)的处理器120。操作系统132和应用程序134的代码可以存储在持久性存储装置136中，并根据需要加载到易失性存储器130中。处理器120可以是机器100内的任何数量的处理器中的一个。处理器120本身可以具有被示为核122、124、126和128的许多不同的计算单元。

[0028] 处理器120包括允许在网络接口控制器102中执行的实体访问机器100的易失性存储器130中的数据140而不涉及核122、124、126和128的电路(被示为非核(uncore)142)。

[0029] 在诸如机器100之类的远程服务器中，能够执行由请求实体110请求的RDMO的实体可以驻留在任意数量的可靠性域中，包括但不限于以下中的任何一个：

[0030] • 网络接口控制器102内的硬件

[0031] • 网络接口控制器102内的FPGA

[0032] • 存储在网络接口控制器102中的固件

[0033] • 处理单元104在网络接口控制器102内执行的软件

[0034] • 易失性存储器106中由解释器解释的指令，该解释器正在由处理单元104在网络接口控制器102内执行

[0035] • 加载到机器100的易失性存储器130中、由处理器120的一个或多个核(122、124、126和128)执行的操作系统132

[0036] • 加载到机器100的易失性存储器130中、由处理器120的一个或多个核(122、124、126和128)执行的应用程序134

[0037] • 在特权域(例如“dom0”)中执行的软件。特权域和dom0例如在en.wikipedia.org/wiki/Xen处被描述。

[0038] 此可靠性域列表仅是示例性的，并且本文描述的技术不限于来自这些可靠性域的执行候选者。上面给出的示例作为不同的可靠性域，因为这些域内的实体在不同条件下发生故障。例如，即使当应用程序134已经崩溃或以其他方式发生故障时，操作系统132也可以继续正常起作用。类似地，即使当处理器120正在执行的所有进程(包括操作系统132和应用程序134)都已崩溃时，在网络接口控制器102内运行的执行候选者也可以继续正常起作用。

[0039] 此外，由于处理器120内的每个核是可以独立于处理器120内的其他计算单元而发生故障的计算单元，因此，核122正在执行的执行候选者与核124正在执行的执行候选者处于不同的可靠性域中。

[0040] 添加对新RDMO的支持

[0041] 机器100可以在网络接口控制器102中或在其他地方包括专用硬件，以实现特定的RDMO。但是，硬件实现的执行候选者不能被轻松扩展以支持附加的RDMO。因此，根据一个实施例，提供了用于向其他类型的执行候选者添加对附加的RDMO的支持的机制。例如，假设NIC 102包括用于执行特定一组RDMO的固件。在这些条件下，可以通过常规固件更新技术向NIC 102添加对附加的RDMO的支持。

[0042] 另一方面，如果执行候选者在FPGA中实现，则可以通过对FPGA重新编程来添加对新RDMO的支持。可以例如通过在上电时向FPGA加载修改后的FPGA比特流来执行这样的重新编程。

[0043] 类似地，可以使用常规软件更新技术将对新RDMO的支持添加到软件实现的执行候选者(例如，NIC 102中的软件、操作系统132和应用程序134)中。在涉及在NIC 102内执行解释器的实施例中，可通过向NIC 102发送实现新的RDMO的代码来支持新的RDMO。NIC 102可将代码存储在易失性存储器106中，并通过解释代码来执行新的RDMO。例如，NIC 102可能正在执行Java虚拟机，并且请求实体110可以通过向NIC 102发送Java字节码来使NIC 102执行新的RDMO(例如，计算数字集合的平均值)，该Java字节码在被Java虚拟机解释时使目标数字集合从易失性存储器130被检索，并且其平均值被计算。

[0044] 示例RDMO

[0045] 在前面的讨论中，给出了示例，其中讨论中的RDMO计算一组值的平均值。这只是可以由来自多个不同可靠性域的执行候选者支持的RDMO的一个示例。如上所述，RDMO可能是任意复杂的。然而，RDMO越复杂，高效执行RDMO将需要的资源就越大，并且RDMO的执行候选者将遇到问题的可能性就越大。另外，复杂的RDMO在由来自具有有限资源的可靠性域的执行候选者执行时可能执行缓慢。例如，通常由使用处理器120的所有计算单元执行的应用程序134执行的复杂RDMO如果由NIC 102上的相对“轻量级”处理单元104执行则将花费显著更长的时间。

[0046] 可以由同一机器内来自不同可靠性域的多个执行候选者支持的RDMO的示例包括但不限于:

[0047] • 在远程服务器机器内设置一批标志，以指示远程服务器机器的易失性存储器的对应部分不可用或已损坏

[0048] • 在远程服务器机器内调整状态，以使远程服务器机器使用错误通知来响应某些类型的请求。当远程服务器机器无法正确处理某些请求时，这可以避免通知所有其他计算设备的需要。

[0049] • 对(存储在远程服务器机器处的易失性和/或持久性存储器中的)数据结构的辅助副本执行一批更改，以使辅助副本反映对(存储在另一台机器的易失性和/或持久性存储器中的)数据结构的主副本所做的更改。

[0050] • 涉及分支的任何算法(例如，被配置为测试一个或多个条件并基于测试结果确定用哪种方式进行分支(以及因此执行哪些动作)的逻辑)。

[0051] • 涉及对远程服务器的易失性存储器进行多次读取和/或多次写入的任何算法。因为每次读取和/或写入都不涉及机器之间的RDMA通信，所以响应于单个RDMO请求来执行多次读取和/或写入显著减少机器间通信开销。

[0052] • 从可操作地耦合到远程服务器机器的持久性存储设备(例如，存储装置136)中读取数据/将数据写入该持久性存储设备中。当持久性存储装置不能被请求实体直接访问时，使用NIC 102中的执行候选者来执行该操作的能力特别有用，使得即使当远程机器的计算单元(例如，核122、124、126和128)上运行的软件(潜在地包括操作系统132本身)可能已经崩溃时，持久性存储装置136上的数据也保持对请求实体可用。

[0053] 后备执行候选者

[0054] 如上所述，可通过使来自一个可靠性域的RDMO执行候选者充当来自另一可靠性域的RDMO执行候选者的后备，来增加RDMO的可用性。选择哪个执行候选者作为RDMO的主候选者以及哪个作为后备候选者可能基于候选者的特征和RDMO的性质而不同。

[0055] 例如，在RDMO用于相对简单的操作的情况下，运行在NIC102中的执行候选者可以比运行在处理器120上的应用程序134更高效地执行操作是可能的。因此，对于该特定的RDMO，可能期望让运行在NIC 102中的执行候选者成为RDMO的主执行候选者，而应用程序134成为后备执行候选者。另一方面，如果RDMO相对复杂，则将应用程序134用作RDMO的主执行候选者而NIC 102上的执行候选者充当后备执行候选者可能更为高效。

[0056] 图2是示出了根据实施例的后备执行候选者的使用的流程图。在步骤200处，请求实体通过网络向远程计算设备发送对RDMO的请求。在步骤202处，远程计算设备处的第一执行候选者尝试执行RDMO。如果RDMO被成功执行(在步骤204处确定)，则在步骤206处指示成功的响应(可能包含附加的结果数据)被发送回请求实体。否则，在步骤208处确定尚未被尝试的另一执行候选者是否可用于执行RDMO。如果没有其他执行候选者可用于RDMO，则在步骤210处错误指示被发送回请求实体。否则，尝试使用另一个执行候选者来执行RDMO。

[0057] 此过程可以继续，直到RDMO被成功执行或RDMO的所有执行候选者都已经失败为止。在一个实施例中，远程服务器处的实体负责遍历执行候选者。在这样的实施例中，进行使用后备执行候选者来执行RDMO的尝试而不向请求实体通知任何失败，直到RDMO的所有可能的执行候选者都已经失败为止。此实施例减少了在遍历过程期间生成的机器间消息流量。

[0058] 在替代实施例中，每次执行候选者执行RDMO失败时请求实体都被通知。响应于失败指示，请求实体确定接下来尝试哪个执行候选者(如果有的话)。响应于确定接下来应尝试特定的执行候选者，请求实体将另一个请求发送到远程计算设备。新请求指示然后应尝试执行RDMO的后备执行候选者。

[0059] 应当注意，执行候选者的失败可以隐式地指示，而不是显式地指示。例如，如果例如在特定时间量过去之后执行候选者尚未确认成功，则可以假定执行候选者已经失败。在这样的情况下，请求实体可以在接收表明先前选择的执行候选者已失败的任何显式指示之前发出由另一执行候选者执行RDMO的请求。

[0060] 尝试执行任何给定RDMO的实际执行候选者以及执行候选者进行尝试的顺序可能基于多种因素而不同。在一个实施例中，将以基于执行候选者的失败可能性的顺序来尝试执行候选者，其中首先尝试运行在远程计算设备上的应用程序(其可能是最有可能崩溃的

候选者)(因为它具有更多可用资源),最后尝试NIC中的硬件实现的候选者(其可能是最不可能崩溃的候选者)(因为它具有最少的可用资源)。

[0061] 作为另一示例,在看起来远程计算设备的处理器过载或崩溃的情况下,请求实体可以首先请求由在远程计算设备的网络接口控制器中实现的实体执行RDMO。另一方面,如果没有这样的指示,则请求实体可以首先请求由远程机器上运行的应用程序执行RDMO,该应用程序可以充分利用远程机器上可用的计算硬件。

[0062] 作为另一示例,对于相对简单的RDMO,请求设备可以首先请求由网络接口控制器中实现的相对“轻量级”的执行候选者来执行RDMO。另一方面,对于相对复杂的RDMO的请求可以首先被发送到远程计算设备上的应用程序,并且仅在应用程序执行RDMO失败时才被发送给轻量级的执行候选者。

[0063] 并行执行候选者

[0064] 同样如上所述,可以通过使来自不同可靠性域的多个执行候选者尝试并行执行RDMO来增加RDMO的可用性。例如,假定RDMO将确定一组数字的平均值。响应于来自请求实体110的单个请求,可以调用在NIC 102中实现的执行候选者和应用程序134两者以执行RDMO。在此示例中,应用程序134和NIC 102中的执行候选者两者都将从易失性存储器130中读取相同组的值(例如,数据140),对组中的数字进行计数,对组中的数字求和,然后将总和除以计数以获得平均值。如果两个执行候选者都没有失败,则两个执行候选者都可以将其响应提供给请求实体110,该请求实体110可以简单地丢弃重复的响应。

[0065] 在一个执行候选者失败并且一个执行候选者成功的情况下,未失败的执行候选者将响应返回给请求实体110。因此,尽管机器100内的一些东西未正确起作用,RDMO仍成功完成。

[0066] 在最初尝试RDMO的所有执行候选者都失败的情况下,来自与第一组执行候选者不同的可靠性域的第二组执行候选者可以尝试并行执行RDMO。在至少一个后备执行候选者成功的情况下,RDMO成功。此过程可以继续进行,直到RDMO的执行候选者中的一个成功,或者RDMO的所有执行候选者失败。

[0067] 机器100内的一些“协调实体”可以调用一组执行候选者以并行执行RDMO,而不是所有成功的执行候选者都将响应返回给请求实体100。如果多个候选者成功,则成功的候选者可以将响应提供回协调实体,然后该协调实体将单个响应返回给请求实体110。此技术简化了请求实体110的逻辑,使机器100上有多少执行候选者被要求执行RDMO以及这些执行候选者中的哪一个成功对请求实体110是透明的。

[0068] 图3是示出了根据实施例的并行执行候选者的使用的流程图。参考图3,在步骤300处,请求实体将对RDMO的请求发送到远程计算设备。在步骤302-1至302-N处,RDMO的N个执行候选者中的每一个同时尝试执行RDMO。在步骤304处,如果任何执行候选者成功,则控制传递到步骤306,在步骤306处,成功的指示(其可以包括附加的结果数据)被发送给请求实体。否则,如果全部失败,则在步骤310处失败的指示被发送到请求实体。

[0069] 作为基于NIC的执行候选者的解释器

[0070] 如上所述,RDMO的一种执行候选者形式是解释器。为了使解释器充当RDMO的执行候选者,向该解释器提供代码,该代码在被解释时执行RDMO所需的操作。这样的解释器可以例如由NIC 102内的处理单元104、由处理器120或由处理器120的核的子集来执行。

[0071] 根据一个实施例，向解释器注册用于特定RDMO的代码。一旦被注册，请求实体就可以(例如，通过远程过程调用)调用代码，从而使代码由解释器解释。在一个实施例中，解释器是Java虚拟机，并且代码是Java字节码。然而，本文使用的技术不限于任何特定类型的解释器或代码。尽管由NIC 102内的解释器执行的RDMO可能比由在处理器120上执行的已编译的应用程序(例如，应用程序134)执行相同的RDMO花费的时间长得多，但NIC 102内的解释器可以在一些错误阻止已编译的应用程序的操作的时间段内是可操作的。

[0072] 硬件概述

[0073] 根据一个实施例，本文描述的技术由一个或多个专用计算设备实现。专用计算设备可以被硬连线以执行技术，或者可以包括数字电子设备，诸如被持久编程以执行技术的一个或多个专用集成电路(ASIC)或现场可编程门阵列(FPGA)，或者可以包括被编程以根据固件、存储器、其他存储装置或其组合中的程序指令来执行技术的一个或多个通用硬件处理器。这样的专用计算设备还可以将定制的硬连线逻辑、ASIC或FPGA与定制的编程相结合以实现技术。专用计算设备可以是台式计算机系统、便携式计算机系统、手持式设备、网络设备或结合了硬连线和/或程序逻辑以实现技术的任何其他设备。

[0074] 例如，图4是示出了可以在其上实现本发明的实施例的计算机系统400的框图。计算机系统400包括用于传达信息的总线402或其他通信机制，以及与总线402耦合以用于处理信息的硬件处理器404。硬件处理器404可以是例如通用微处理器。

[0075] 计算机系统400还包括主存储器406(诸如随机存取存储器(RAM)或其他动态存储设备)，主存储器406耦合到总线402以用于存储信息和要由处理器404执行的指令。主存储器406还可以用于在执行要由处理器404执行的指令期间存储临时变量或其他中间信息。这样的指令当被存储在处理器404可以访问的非暂时性存储介质中时，使计算机系统400成为被定制以执行指令中指定的操作的专用机器。

[0076] 计算机系统400还包括耦合到总线402的只读存储器(ROM)408或其他静态存储设备，用于存储静态信息和用于处理器404的指令。存储设备410(诸如磁盘、光盘或固态驱动器)被提供并被耦合到总线402以用于存储信息和指令。

[0077] 计算机系统400可以经由总线402耦合到显示器412(诸如阴极射线管(CRT))，用于向计算机用户显示信息。包括字母数字键和其他键的输入设备414耦合到总线402，用于将信息和命令选择传达给处理器404。另一种类型的用户输入设备是光标控件416(诸如鼠标、轨迹球或光标方向键)，用于将方向信息和命令选择传达给处理器404，并用于控制显示器412上的光标移动。该输入设备通常具有允许设备指定平面中的位置的在两个轴(第一轴(例如，x)和第二轴(例如，y))上的两个自由度。

[0078] 计算机系统400可以使用与计算机系统结合使计算机系统400成为专用机器或将计算机系统400编程为专用机器的定制的硬连线逻辑、一个或多个ASIC或FPGA、固件和/或程序逻辑来实现本文所述的技术。根据一个实施例，本文的技术由计算机系统400响应于处理器404执行包含在主存储器406中的一个或多个指令的一个或多个序列来执行。这样的指令可从另一存储介质(诸如存储设备410)读入主存储器406中。主存储器406中包含的指令序列的执行使处理器404执行本文所述的处理步骤。在替代实施例中，硬连线电路可以代替软件指令被使用或与软件指令结合使用。

[0079] 如本文所使用的术语“存储介质”是指存储使机器以具体方式操作的数据和/或指

令的任何非暂时性介质。这样的存储介质可以包括非易失性介质和/或易失性介质。非易失性介质包括例如光盘、磁盘或固态驱动器，诸如存储设备410。易失性介质包括动态存储器，诸如主存储器406。存储介质的常见形式包括例如软盘、柔性盘、硬盘、固态驱动器、磁带或任何其他磁性数据存储介质、CD-ROM、任何其他光学数据存储介质、带孔图案的任何物理介质、RAM、PROM和EPROM、FLASH-EPROM、NVRAM、任何其他存储器芯片或盒式磁带。

[0080] 存储介质与传输介质不同，但可以与传输介质结合使用。传输介质参与存储介质之间的信息传输。例如，传输介质包括同轴线缆、铜线和光纤，包括包含总线402的线。传输介质还可以采用声波或光波的形式，诸如在无线电波和红外数据通信期间生成的声波或光波。

[0081] 将一个或多个指令的一个或多个序列携带到处理器404以用于执行可以涉及各种形式的介质。例如，指令最初可以被携带在远程计算机的磁盘或固态驱动器上。远程计算机可以将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指令。计算机系统400本地的调制解调器可以在电话线上接收数据，并使用红外发射器将数据转换为红外信号。红外检测器可以接收红外信号中携带的数据，并且适当的电路可以将数据放置在总线402上。总线402将数据携带到主存储器406，处理器404从该主存储器406中检索指令并执行指令。由主存储器406接收的指令可以可选地在由处理器404执行之前或之后被存储在存储设备410上。

[0082] 计算机系统400还包括耦合到总线402的通信接口418。通信接口418提供与连接到本地网络422的网络链路420耦合的双向数据通信。例如，通信接口418可以是集成服务数字网络(ISDN)卡、线缆调制解调器、卫星调制解调器或提供到对应类型的电话线的数据通信连接的调制解调器。作为另一个示例，通信接口418可以是局域网(LAN)卡，以提供到兼容LAN的数据通信连接。还可以实现无线链路。在任何这样的实现中，通信接口418发送和接收携带表示各种类型的信息的数字数据流的电信号、电磁信号或光信号。

[0083] 网络链路420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420可以通过本地网络422提供到主机计算机424或由因特网服务提供商(ISP)426操作的数据设备的连接。ISP426又通过现在通常被称为“因特网”428的全球分组数据通信网络提供数据通信服务。本地网络422和因特网428都使用携带数字数据流的电信号、电磁信号或光信号。携带去往和来自计算机系统400的数字数据的通过各种网络的信号和在网络链路420上并通过通信接口418的信号是传输介质的示例形式。

[0084] 计算机系统400可以通过(一个或多个)网络、网络链路420和通信接口418发送消息并接收数据，包括程序代码。在因特网示例中，服务器430可以通过因特网428、ISP 426、本地网络422和通信接口418发送应用程序的请求代码。

[0085] 接收到的代码可以在它被接收到时由处理器404执行，和/或存储在存储设备410或其他非易失性存储装置中，以用于以后执行。

[0086] 云计算

[0087] 本文中通常使用术语“云计算”来描述一种计算模型，该计算模型使得能够按需访问共享的计算资源池(诸如计算机网络、服务器、软件应用程序和服务)，并允许以最少的管理工作或服务提供商交互来快速提供和释放资源。

[0088] 可以以多种不同的方式来实现云计算环境(有时也被称为云环境或云)，以最佳地

适合不同的需求。例如，在公共云环境中，底层计算基础设施由使其云服务对其他组织或一般公众可用的组织所拥有。相比之下，私有云环境通常旨在仅用于由单个组织使用或在单个组织内使用。社区云旨在由社区内的若干组织共享；而混合云包括通过数据和应用程序可移植性绑定在一起的两种或更多种类型的云(例如，私有的、社区的或公共的)。

[0089] 通常，云计算模型可以使先前可能已经由组织自己的信息技术部门提供的那些职责中的一些指责能够替代地作为云环境内的服务层被交付，以供(根据云的公共/私有性质，在组织内或外部的)消费者使用。取决于特定的实现，由每个云服务层提供或在每个云服务层内的组件或特征的精确定义可能会有所不同，但是常见示例包括：软件即服务(SaaS)，其中消费者使用在云基础设施上运行的软件应用程序，而SaaS提供商管理或控制底层的云基础设施和应用程序。平台即服务(PaaS)，其中消费者可以使用由PaaS提供商支持的软件编程语言和开发工具来开发、部署和以其他方式控制其自己的应用程序，而PaaS提供商管理或控制云环境的其他方面(即，运行时执行环境以下的所有内容)。基础设施即服务(IaaS)，其中消费者可以部署和运行任意软件应用程序，和/或提供处理、存储、网络和其他基础计算资源，而IaaS提供商管理或控制底层物理云基础设施(即，操作系统层以下的所有内容)。数据库即服务(DBaaS)，其中消费者使用在云基础设施上运行的数据库服务器或数据库管理系统，而DbaaS提供商管理或控制底层云基础设施、应用程序和服务器，包括一个或多个数据库服务器。

[0090] 在前述说明书中，已经参考可能随实现方式而变化的许多具体细节描述了本发明的实施例。因此，说明书和附图应被认为是说明性的而不是限制性的。本发明范围的唯一且排他性的指示物以及申请人旨在作为本发明范围的内容是以权利要求发布的具体形式从本申请发布的一组权利要求的字面和等效范围，包括任何后续更正。

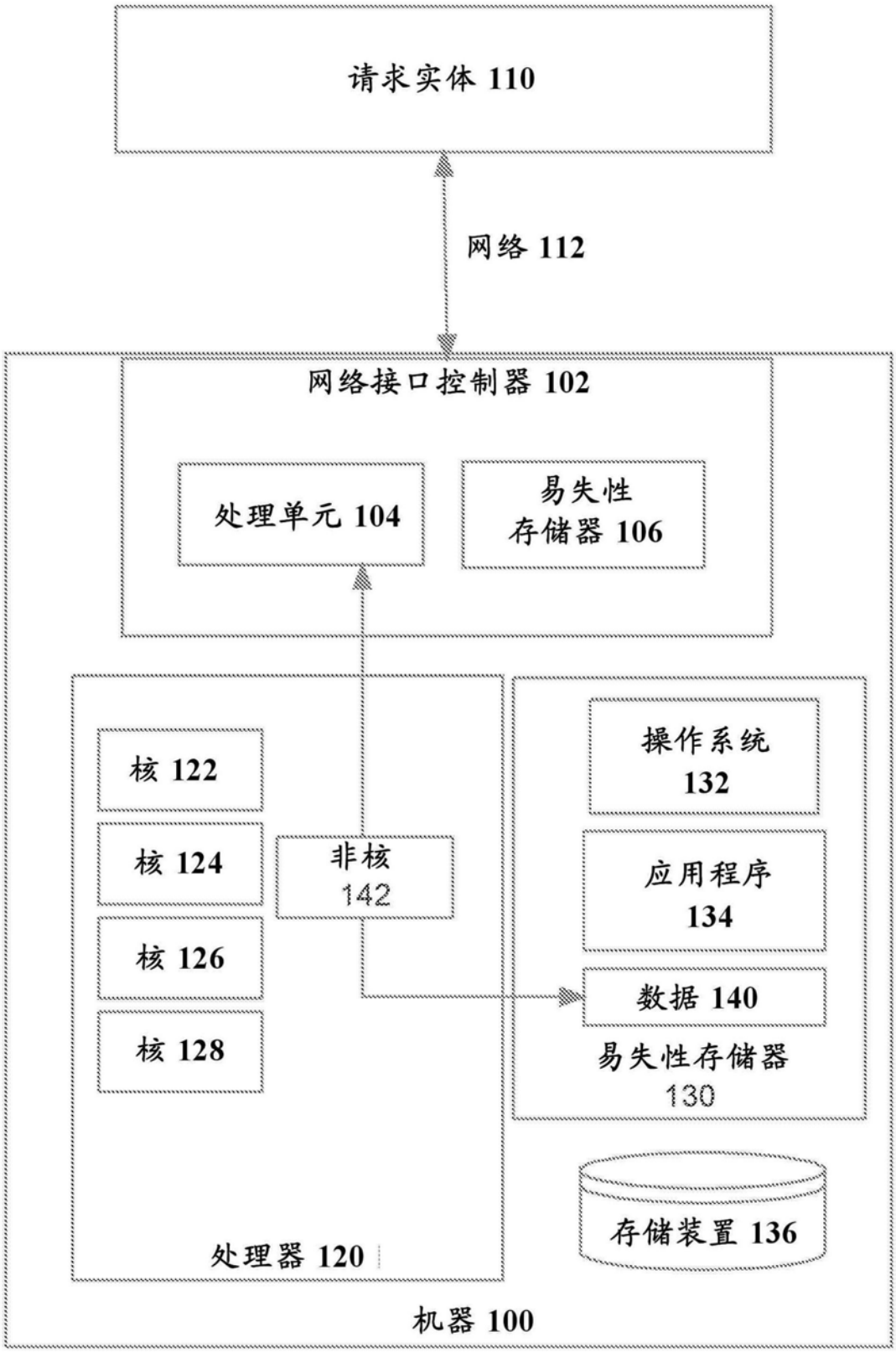

请求实体 110
网络 112
网络接口控制器 102
处理单元 104
易失性
存储器 106
核 122
核 124
核 126
核 128
非核
142
操作系统
132
应用程序
134
数据 140
易失性存储器
130
存储装置 136
处理器 120
机器 100

图1

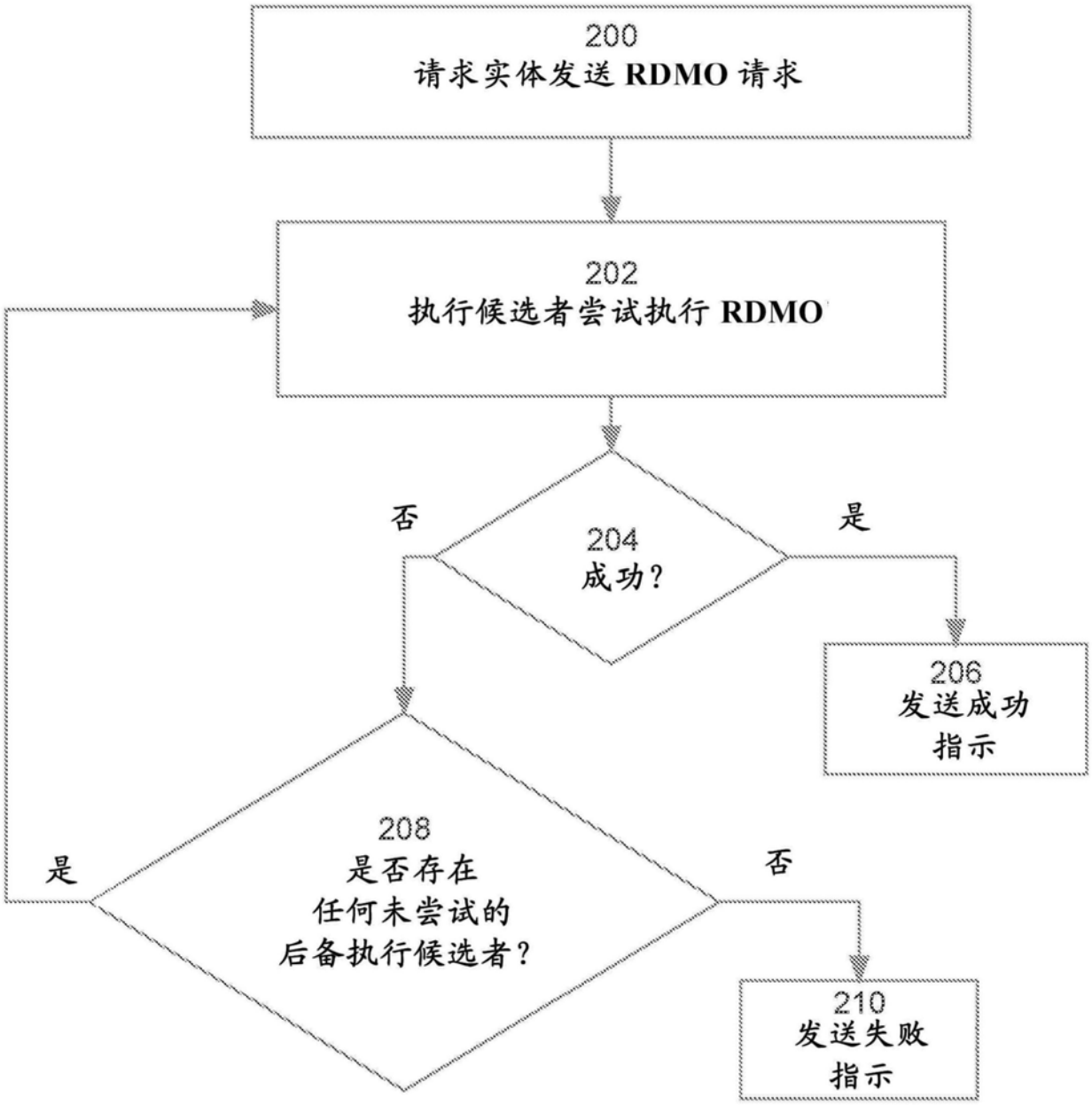

200
请求实体发送 RDMO 请求
202
执行候选者尝试执行 RDMO
204
成功？
否
是
206
发送成功
指示
208
是否存在
任何未尝试的
后备执行候选者？
是
否
210
发送失败
指示

图2

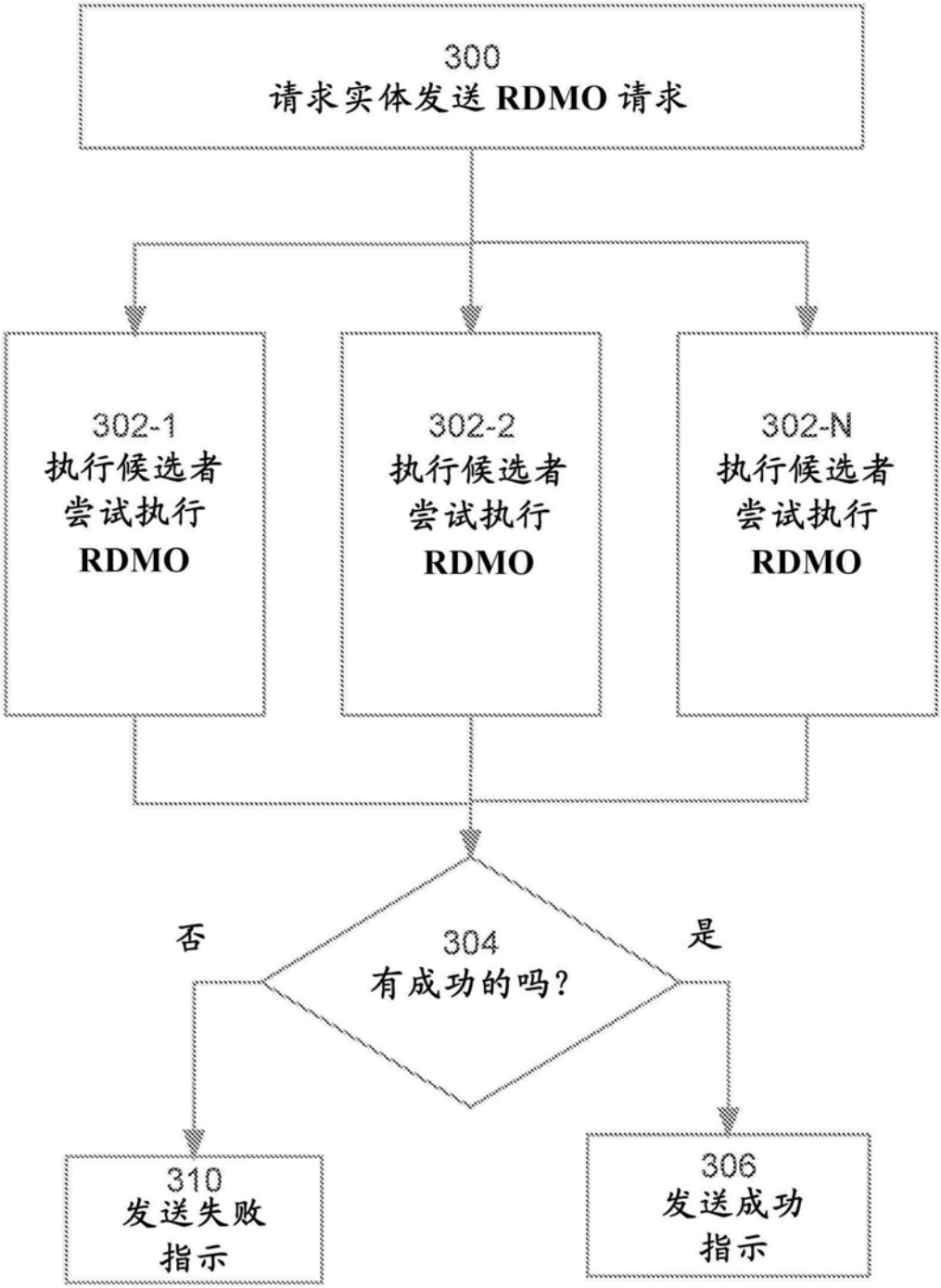

300
请求实体发送 RDMO 请求
302-1
执行候选者
尝试执行
RDMO
302-2
执行候选者
尝试执行
RDMO
302-N
执行候选者
尝试执行
RDMO
304
有成功的吗?
否
是
310
发送失败
指示
306
发送成功
指示

图3

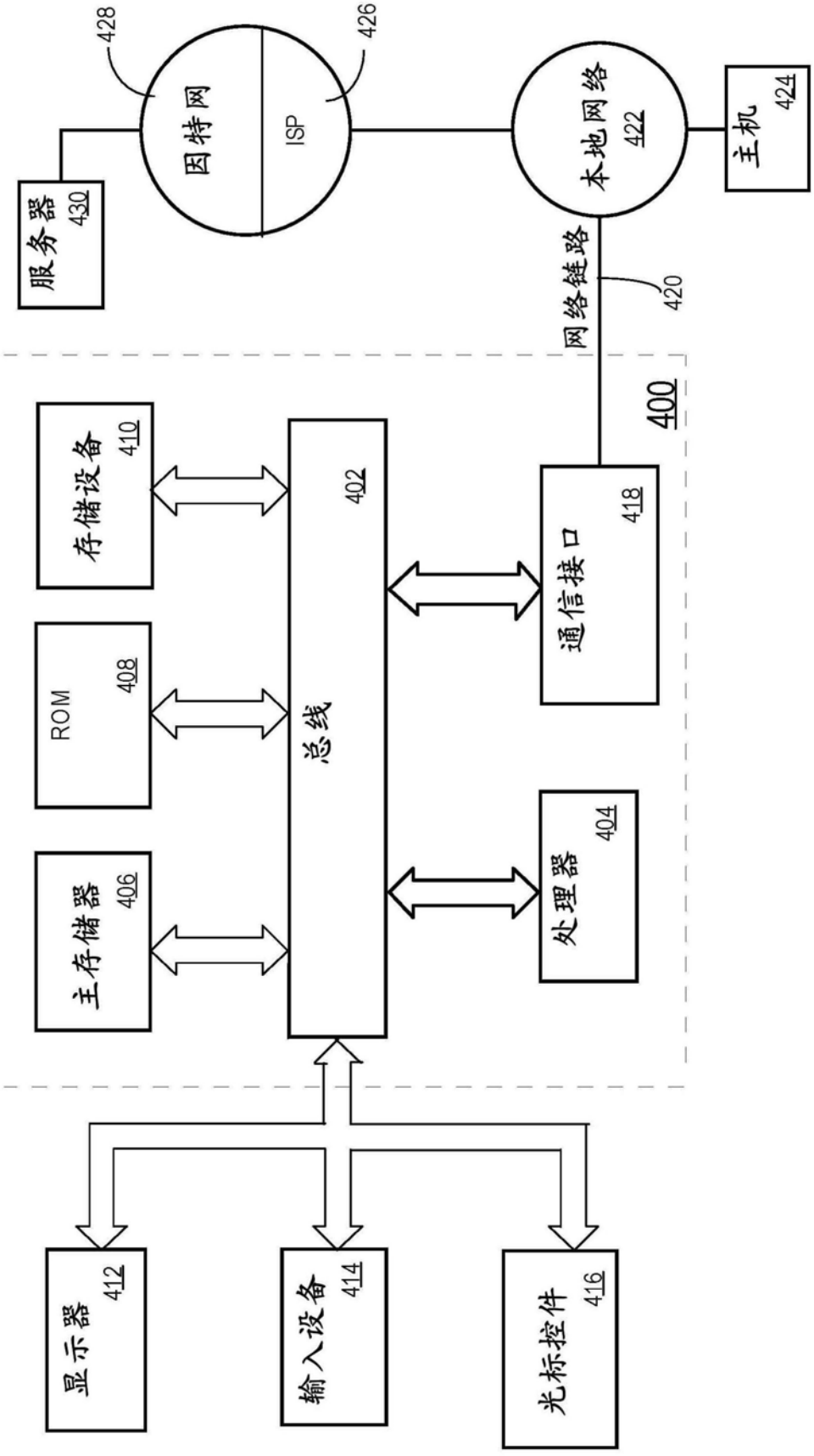

428
因特网
ISP
426
服务器
430
本地网络
422
主机
424
网络链路
420
400
存储设备
410
402
通信接口
418
ROM
408
总线
主存储器
406
处理器
404
显示器
412
输入设备
414
光标控件
416

图4