(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 103201724 A
(43)申请公布日 2013.07.10

(21)申请号 201180035090.2
(22)申请日 2011.07.29
(30)优先权数据
12/847,030 2010.07.30 US
(85)PCT申请进入国家阶段日
2013.01.16
(86)PCT申请的申请数据
PCT/US2011/045951 2011.07.29
(87)PCT申请的公布数据
WO2012/016175 EN 2012.02.02
(71)申请人 赛门铁克公司
地址 美国加利福尼亚州
(72)发明人 乔格·罗希特·维贾伊
萨林·苏米特·曼莫汉
(74)专利代理机构 北京纪凯知识产权代理有限公司 11245
代理人 赵蓉民
(51)Int.Cl.
G06F 11/07(2006.01)

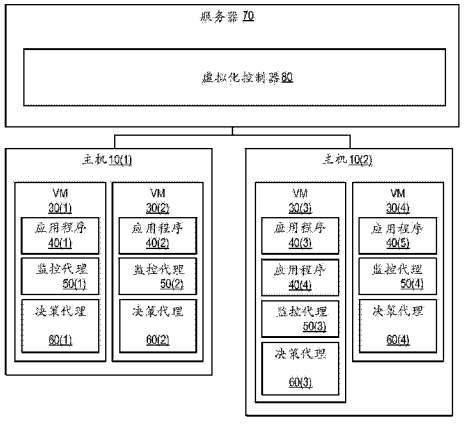
权利要求书2页 说明书11页 附图6页

(54)发明名称

在高可用性虚拟机环境中提供高可用性应用程序

(57)摘要

不同的系统和方法可以提供在高可用性虚拟机环境下执行的应用程序的高可用性。一种方法包括从在虚拟机中执行的监控代理接收指示该虚拟机中执行的应用程序的状态的信息。响应于接收到该信息，该方法涉及确定是否应该重启该虚拟机。根据该确定，该方法然后确定监控代理是否应该在超时间隔到期之前向虚拟化控制器发出心跳信息。如果该虚拟机在超时间隔到期之前没有发出心跳信息，配置该虚拟化控制器以重新启动虚拟机。



CN 103201724 A

1. 一种方法，包括：

从在一个虚拟机中执行的一个监控代理接收指示在该虚拟机中执行的一个应用程序的状态的信息；

响应于该接收，确定是否应该重启该虚拟机；并且

基于确定是否应该重启该虚拟机，确定该监控代理是否应该在一个超时间隔到期之前向一个虚拟化控制器发出一条心跳信息，其中该虚拟化控制器被配置为如果该虚拟机在该超时间隔到期之前没有发出该心跳信息就重启该虚拟机，并且其中确定是否发出该心跳信息由一个计算装置执行。

2. 根据权利要求 1 所述方法，其中该信息识别在该虚拟机中执行的多个应用程序中的每一个应用程序的运行状态，并且其中该确定是基于该多个应用程序中的每一个应用程序的运行状态和优先级。

3. 根据权利要求 2 所述方法，进一步包括响应于识别出该多个应用程序中一个较高优先级的应用程序正确执行并且该多个应用程序中一个较低优先级的应用程序发生故障的信息，确定不应该重启该虚拟机。

4. 根据权利要求 2 所述方法，进一步包括响应于识别出该多个应用程序中一个较高优先级的应用程序发生故障并且该多个应用程序中一个较低优先级的应用程序正确执行的信息，确定应该重启该虚拟机。

5. 根据权利要求 2 所述方法，其中该确定过程由一个决策代理执行，其中该决策代理不在该虚拟机中执行。

6. 根据权利要求 5 所述方法，进一步包括响应于识别出该应用程序未正确运行的信息，决定该监控代理应该尝试重启该应用程序，其中尝试重启该应用程序在确定是否应该重启该虚拟机之前执行。

7. 根据权利要求 6 所述的方法，进一步包括响应于识别出该应用程序已经有意脱机的信息，决定注销该监控代理以避免向该虚拟化控制器提供心跳信息。

8. 根据权利要求 7 所述方法，进一步包括在该监控代理注销后，该监控代理继续监控该应用程序。

9. 根据权利要求 7 所述的方法，进一步包括响应于识别出该应用程序重新在线的消息，重新注册该监控代理以便向该虚拟化控制器提供心跳信息。

10. 根据权利要求 1 所述方法，进一步包括从一个管理员接收信息，其中该信息指示该应用程序是否已经有意脱机和该监控代理是否应该继续监控该应用程序中的至少一项。

11. 一种系统，包括：

一个或多个处理器；以及

存储器，该存储器连接至该一个或多个处理器并存储可由该一个或多个处理器执行的程序指令，以便：

从在一个虚拟机中执行的一个监控代理接收指示在该虚拟机中执行的一个应用程序的状态的信息；

响应于接收到该信息，确定是否应该重启该虚拟机；并且

基于确定是否应该重启该虚拟机，确定该监控代理是否应该在一个超时间隔到期之前向一个虚拟化控制器发出一条心跳信息，其中该虚拟化控制器被配置为如果该虚拟机在该

超时间隔到期之前没有发出该心跳信息就重启该虚拟机。

12. 根据权利要求 11 所述的系统，其中该信息识别在该虚拟机中执行的多个应用程序中的每一个应用程序的运行状态，并且其中该确定是基于该多个应用程序中的每一个应用程序的运行状态和优先级。

13. 根据权利要求 12 所述的系统，其中该程序指令进一步可由该一个或多个处理器执行，以便响应于识别出该多个应用程序中一个较高优先级的应用程序正确执行并且该多个应用程序中一个较低优先级的应用程序发生故障的信息，确定不应该重启该虚拟机。

14. 根据权利要求 12 所述的系统，其中该程序指令进一步可由该一个或多个处理器执行，以便响应于识别出该多个应用程序中一个较高优先级的应用程序发生故障并且该多个应用程序中一个较低优先级的应用程序正确执行的信息，确定应该重启该虚拟机。

15. 根据权利要求 12 所述的系统，程序指令实施一个决策代理，并且其中该决策代理不在该虚拟机中执行。

16. 根据权利要求 11 所述的系统，其中该程序指令进一步可由该一个或多个处理器执行，以便响应于识别出该应用程序已经有意脱机的信息，决定注销该监控代理以避免向该虚拟化控制器提供心跳信息。

17. 一种计算机可读存储介质，包括由一个或多个处理器可执行的程序指令以便：

从在一个虚拟机中执行的一个监控代理接收指示在该虚拟机中执行的一个应用程序的状态的信息；

响应于接收到该信息，确定是否应该重启该虚拟机；并且

基于确定是否应该重启该虚拟机，确定该监控代理是否应该在一个超时间隔到期之前向一个虚拟化控制器发出一条心跳信息，其中该虚拟化控制器被配置为如果该虚拟机在该超时间隔到期之前没有发出该心跳信息就重启该虚拟机。

18. 根据权利要求 17 所述的系统，其中该信息识别在该虚拟机中执行的多个应用程序中的每一个应用程序的运行状态，并且其中该确定是基于该多个应用程序中的每一个应用程序的运行状态和优先级。

19. 根据权利要求 18 所述的系统，其中该程序指令进一步可由该一个或多个处理器执行，以便响应于识别出该多个应用程序中一个较高优先级的应用程序正确执行并且该多个应用程序中一个较低优先级的应用程序发生故障的信息，确定不应该重启该虚拟机。

20. 根据权利要求 18 所述的系统，其中该程序指令进一步可由该一个或多个处理器执行，以便响应于识别出该多个应用程序中一个较高优先级的应用程序发生故障并且该多个应用程序中一个较低优先级的应用程序正确执行的信息，确定应该重启该虚拟机。

21. 根据权利要求 18 所述的系统，程序指令实施一个决策代理，并且其中该决策代理不在该虚拟机中执行。

22. 根据权利要求 17 所述的系统，其中该程序指令进一步可由该一个或多个处理器执行，以便响应于识别出该应用程序已经有意脱机的信息，决定注销该监控代理以避免向该虚拟化控制器提供心跳信息。

在高可用性虚拟机环境中提供高可用性应用程序

[0001] 乔格·罗希特·维贾伊(Jog Rohit Vijay)萨林·苏米特·曼莫汉(Sarin Sumit Manmohan)

技术领域

[0002] 本发明涉及高可用性,更具体地是涉及在虚拟机中提供高可用性应用程序。

背景技术

[0003] 虚拟化系统允许多个操作系统(实际上可以是相同类型操作系统的单独实例)在同一时段内在同一硬件上执行。每个执行的操作系统充当一个独立的“虚拟机”,它们可以交互并且可以采用基本相同的方式用作在独立硬件上执行的独立操作系统。通过将一个硬件计算装置有效地转变成若干虚拟机,虚拟机允许提高硬件资源的使用率。

[0004] 一些虚拟化系统提供虚拟化控制器,该虚拟化控制器可以管理在一个或多个计算装置上实施的一个或多虚拟机。此类虚拟化控制器可以与虚拟机通信并且控制这些虚拟机的运行。在某些环境下,虚拟化控制器甚至可以采用提供虚拟机的高可用性的方式来管理虚拟机,这样如果一个特定虚拟机发生故障,虚拟化控制器可以在另一个计算装置上重启发生故障的虚拟机。不幸的是,在提供高可用性虚拟机的传统虚拟化系统中,同样不可能有效地使虚拟机内执行的应用程序具有高可用性。

发明内容

[0005] 在此披露了各种提供在高可用性虚拟机环境下执行的应用程序的高可用性的系统和方法。例如,一种方法包括从在虚拟机中执行的监控代理接收指示该虚拟机中执行的应用程序的状态的信息。响应于接收到该信息,该方法确定是否该重启该虚拟机。基于该确定,该方法然后确定监控代理是否在超时间隔到期之前向虚拟化控制器发出心跳信息。如果虚拟机在超时间隔到期之前没有发出心跳信息,配置虚拟化控制器以重新启动虚拟机。确定是否发出心跳信息的操作由计算装置执行。在一些实施方案中,由实施决策代理的计算装置来执行该确定操作,该决策代理并不在该虚拟机中执行。

[0006] 接收到的信息可以识别在虚拟机中执行的多个应用程序中的每一个应用程序的运行状态。确定是否重新启动该虚拟机可以基于该多个应用程序中的每一个应用程序的运行状态和优先级。

[0007] 例如,响应于识别出该多个应用程序中一个较高优先级的应用程序正确执行并且该多个应用程序中一个较低优先级的应用程序发生故障的信息,该方法可以确定不应该重启该虚拟机。或者,响应于识别出该多个应用程序中一个较高优先级的应用程序发生故障并且该多个应用程序中一个较低优先级的应用程序正确执行的信息,该方法可以确定应该重启该虚拟机。在一个实施方案中,响应于识别出该应用程序不正确运行的信息,该方法可以确定监控代理首先尝试重启该应用程序,其中在确定虚拟机是否应该重启之前尝试重启该应用程序。

[0008] 响应于识别出该应用程序已经有意脱机的信息，该方法可以决定注销监控代理以避免向虚拟化控制器提供心跳信息。监控代理被注销之后可以继续监控该应用程序。响应于识别出该应用程序重新上线的信息，该方法可以决定重新注册监控代理以便向虚拟化控制器提供心跳信息。

[0009] 该方法可以包括从管理员接收信息。例如，该信息可以指示是否应用程序已经脱机和 / 或是否监控代理应该继续监控应用程序。

[0010] 系统的一个实例包括一个或多个处理器和连接至该一个或多个处理器的存储器。该存储器存储程序指令，这些程序指令可执行用于实施类似上述方法的一种方法。同样，这些程序指令可以存储在计算机可读存储介质上。

[0011] 前述内容是概述，因此必然包含细节的简化、概括和省略；因此本领域的普通技术人员将会认识到此概述只是示例性的并且不用于任何形式的限制。仅由权利要求定义的本发明的其他方面、创新性特征以及优点在以下的非限定性详细描述中将变得明显。

[0012] 附图简述

[0013] 通过参考附图，本发明可以得到更好的理解，并且其众多目标、特征以及优点对本领域的普通技术人员是明显的。

[0014] 图 1 是根据本发明的一个实施方案的一种在高可用性虚拟机环境下提供应用程序的高可用性的系统的框图。

[0015] 图 2 是根据本发明的另一个实施方案的另一种在高可用性虚拟机环境下提供应用程序的高可用性系统的框图。

[0016] 图 3 是根据本发明的一个实施方案的一种在高可用性虚拟机环境下提供应用程序的高可用性的方法的流程图。

[0017] 图 4 是根据本发明的一个实施方案的一种确定是否允许心跳信号从执行多个监控应用程序的虚拟机发送到虚拟化控制器的方法的流程图。

[0018] 图 5 是根据本发明的一个实施方案的一种计算装置的框图，该框图展示了监控代理和决策代理两者如何在软件中实施。

[0019] 图 6 是根据本发明的一个实施方案的一种网络系统的框图，该框图展示了各种计算装置如何通过网络进行通信。

[0020] 尽管很容易对本发明做出各种修改和替代形式，但在附图和详细的说明中仍以实例提供本发明的具体实施方案。应当理解，附图和详细说明并无意将本发明限制于所披露的具体形式。相反，目的是覆盖所有落入由所附权利要求所限定的本发明的精神和范围内的修改形式、等同形式和替代形式。

[0021] 实施本发明的一种或多种模式

[0022] 图 1 是一个虚拟化系统的框图。如图所示，虚拟化系统包括两台主机 10(1) 和 10(2)。在此实例中，每台主机实施两个虚拟机（VMs）。如图所示，主机 10(1) 实施 VM30(1) 和 VM302(2). 同样，主机 10(2) 实施 VM30(3) 和 VM30(4)。应当注意的是替代实施方案可以每台主机实施与此实例中所示的主机不同数量的 VM，并且相同的系统可以包括一个或多个主机，其中每台主机可以和在同一系统的其他主机实现不同数量的 VM。

[0023] 虚拟机 VM30(1)-VM30(4) 各自提供一个执行操作系统的自容式实例。可以利用相同或不同的操作系统实施在同一主机上执行的多个 VM。与 VM 交互的客户端通常将以完全

相同的方式(客户端与独立硬件上运行的独立操作系统进行交互)进行交互。可以利用虚拟化软件(例如美国加利福尼亚州的 Palo Alto 公司提供的 VMware)来实施虚拟机。

[0024] 应用程序在虚拟机的每一个的内部执行。这种应用程序可以是数据库应用程序、邮件服务器等。这里,应用程序 40(1) 在 VM30(1) 中执行,应用程序 40(2) 在 VM30(2) 中执行,应用程序 40(3) 和 40(4) 在 VM30(3) 中执行,而应用程序 40(5) 在 VM30(4) 中执行如图所示,多个应用程序可以在一个单一 VM 内执行。在一个给定的 VM 上执行的应用程序的类型和在同一个虚拟化系统内的另一个 VM 上执行的应用程序的类型可以相同或不同。

[0025] 监控代理(一种特殊类型的应用程序)也在每个 VM 内执行。每个监控代理被配置为监控在相同 VM 内执行的一个或多个应用程序的状态并且将所监控的一个或多个程序的状态报告给决策代理。如图所示,监控代理 50(1) 在 VM30(1) 中执行并且监控应用程序 40(1) 的状态。监控代理 50(2) 在 VM30(2) 中执行并且监控应用程序 40(2) 的状态。监控代理 50(3) 在 VM30(3) 中执行并且监控应用程序 40(3) 和 40(4) 中每一个的状态。监控代理 50(4) 在 VM30(4) 中执行并且监控应用程序 40(5) 的状态。尽管图 1 展示了一个单一的监控代理可以监控多个应用程序的实施方案,但替代实施方案可以在每个 VM 上使用多个监控代理,这样在被监控的应用程序和监控代理之间存在一一对应关系。应当注意的是监控代理可以被配置为监控比给定 VM 中执行的所有应用程序更少的应用程序。

[0026] 在一些实施方案中,监控代理被实施为 VERITAS 集群服务器(VCS)代理,该代理可从美国加利福尼亚州的库比蒂诺的 Symantec 公司购得。在这种实施方案中,每个 VM 可以被配置为一个独立的单节点群集。监控代理可以被配置为该单节点群集的监控服务组。在 VM 中受到监控的每个应用程序还可以被配置为该单节点群集的服务组。该监控服务组监控相同单节点集群内的其他服务组的状态。

[0027] 决策代理(另一种特殊类型的应用程序)在每一个 VM 内执行。决策代理被配置用于基于一个或多个应用程序的状态(由一个或多个相应的监控代理来识别)来确定应当采取什么动作(如果存在的话)。将在以下进行更为详细地描述决策代理的操作。如图所示,决策代理 60(1) 在 VM30(1) 中执行,决策代理 60(2) 在 VM30(2) 中执行,决策代理 60(3) 在 VM30(3) 中执行,而决策代理 60(4) 在 VM30(4) 中执行。在一些实施方案中,监控代理和决策代理的功能可以合并为在每一个 VM 内执行的一个单一代理。

[0028] 主机 10(1) 和 10(2) 相连接以便与服务器 70 进行通信(例如,通过网络),该服务器执行虚拟化控制器 80。虚拟化控制器 80 控制在主机 10(1) 和 10(2) 上实施的多个 VM。因此,虚拟化控制器 80 监控每个 VM 的状态以识别每个 VM 是否正确执行。响应于检测到 VM 出现故障,虚拟化控制器 80 可以采取行动来纠正这种情况。此类行动可以包括在相同的或另一台主机上重启故障的 VM 或尝试纠正导致故障的问题。

[0029] 在一些实施方案中,虚拟化控制器 80 可以作为 VMware 的 vCenter 服务器 (TM) 来实施。在此类实施方案中,虚拟化控制器 80 可以通过 VMware Tools (TM)提供的心跳通道与 VM 进行通信,该 TM 可以安装在每一个 VM 中并在其中执行。在此类实施方案中,用于在另一个主机上重新启动 VM 的机构可以是 vMotion(TM)。可以利用虚拟化软件(例如由美国加利福尼亚州的 Palo Alto 公司的 VMware 所提供的虚拟化软件)来实现这些特征。

[0030] 这里,虚拟化控制器 80 依赖于每个 VM 的心跳信息来确定该 VM 的状态。在每个 VM 上的应用程序(例如,以下更为详细描述的监控代理)向虚拟化控制器 80 注册,以指示注册

的应用程序将发送心跳给虚拟化控制器80。而当处于从一个特定VM接收心跳的状态时，虚拟化控制器80预计从该特定VM每周期(也称作超时间隔)接收一次心跳。如果在一个给定的周期内没有接收到一个心跳(或者如果在连续周期上没有接收到一系列心跳)，虚拟化控制器80将确定该VM出现故障并且采取纠正行动(例如，重启另一台主机上的VM)。

[0031] 为了提供在图1的虚拟环境中执行的应用程序的高可用性，每个监控代理将向虚拟化控制器80注册，以为它的VM提供心跳。因此，监控代理50(1)注册以便为VM30(1)提供心跳，监控代理50(2)注册以便为VM30(2)提供心跳，监控代理50(3)注册以便为VM30(3)提供心跳，监控代理50(4)注册以便为VM30(4)提供心跳。

[0032] 如上所述，监控代理50(1)-50(4)将监控应用程序并且将这些应用程序的状态报告给决策代理60(1)-60(4)中的一个对应的决策代理。根据在特定VM上监控的程序的状态，该VM中的决策代理将决定是否应该重启该应用程序，该VM是否应该继续由虚拟化控制器进行监控，并且是否应该发送该VM的心跳信息。然后决策代理将其决策报告给监控代理，监控代理会采取所决定的行动。

[0033] 对于作为监控代理的相同VM内的每个受监控应用程序，该监控代理可以检测该应用程序是否正确执行。在至少一些实施方案中，监控代理还可以在一个非执行的应用程序是由于故障还是由于有意脱机而不执行之间进行辨别。例如，监控代理可以提供接口(例如，图形用户接口、命令行接口等)，管理员可以通过该接口通知该监控代理一个应用程序已经脱机，作为响应，该监控代理可以更新与该应用程序相关的状态信息以指示该应用程序是有意脱机的。应当注意的是管理员可以在任何时候使应用程序脱机或重新联线(并且通过接口识别此动作)，不管当前的操作状态是否由监控代理所检测。

[0034] 当监控代理检测到应用程序未执行，监控代理可以进行检查(例如，通过访问与该应用程序相关联的存储信息)以查看是否在该应用程序停止执行之前管理员就指示了该应用程序正在脱机。当有意脱机之后应用程序重新启动时，管理员可以再次(例如，通过接口)通知监控代理应用程序的状态变化。可替代地，监控代理可以简单检测到该应用程序已经在下个心跳周期重新启动并且清除以前与该应用程序相关的任何信息。

[0035] 因此，监控代理识别每个被监控的应用程序是否正确执行。如果未正确执行，至少在某些实施方案中，监控代理将进一步分辨应用程序没有执行是由于故障或由于有意脱机。监控代理可以产生描述应用程序及其检测到的执行状态的信息，并且将这种信息提供给决策代理。例如，监控代理50(1)可以检测到应用程序40(1)发生故障，并且可以向决策代理60(1)发送指示应用程序40(1)发生故障的信息。

[0036] 当从监控代理接收描述每个应用程序的状态的信息(例如，在线、故障、或有意脱机)时，作为响应，决策代理决定采取何种动作。如果在VM中只有一个单一的受监控的应用程序，而且该应用程序正在正常执行，决策代理将决定发送心跳信息(确保该虚拟化控制器不重启该VM)。如果该应用程序发生故障，决策代理将决定禁止发送心跳信息，这样将使虚拟化控制器重启VM(在该VM中发生故障应用程序正在执行)，这将在已重启的VM内有效地重启发生故障的应用程序。

[0037] 如果应用程序已经有意脱机(例如，如果应用程序不执行但是并未出现故障)，决策代理可以决定注销理应该撤销向虚拟化控制器提供心跳。监控代理通过向虚拟化控制器发送一个注销请求来实现注销。注销处理有效地从为高可用性而进行监控的VM组中删除

已注销的 VM。虚拟化控制器不再期望从 VM 接收到定期的心跳信息，并且当没有心跳从 VM 接收到时不会重启 VM。

[0038] 接口(例如，用于允许管理员指示何时应用程序已经有意脱机)还可以允许管理员指定监控代理何时应该重新注册。例如，当应用程序重新联机时，管理员可以指定是否监控代理应该注册心跳信息。例如，如果应用程序脱机进行升级时，管理员可以指示该注销应该是暂时的。相反如果该应用程序是因为不再用于提供服务而脱机的，管理员可以指示该注销应该是永久性的。

[0039] 监控代理可以获取并报告每个被监控的应用程序每一次心跳周期的状态。同样，决策代理可以接收该信息并基于该信息在每一次心跳周期上产生一个决定。在不注册监控代理以发送心跳给虚拟化控制器期间，监控代理仍然可以继续监控一个或多个应用程序的状态并且向决策代理报告应用程序状态。同样，决策代理可以每周期一次地的从监控代理接收信息并且利用该信息决定采取何种动作(例如，发送心跳，不发送心跳，或注销)。在注销的同时，通过继续监控应用程序，监控代理可以检测何时应用程序重新联机。当检测到该动作时，作为响应，决策代理可以使监控代理重新注册以便向虚拟化控制器发送心跳信息。

[0040] 如果多个应用程序被监控时，监控代理将向适当的决策代理报告每个被监控应用程序的状态，然后根据状态和每个应用程序的优先级决定采用何种操作。应用程序的优先级可以通过管理员进行配置(例如，通过命令行接口或图形用户接口输入信息)并由决策代理存储。

[0041] 通常在做决定时，相比于较低优先级应用程序，决策代理将给予高优先级应用程序的状态更高的权重。考虑应用程序优先级的各种不同算法中的任意一种都可用于实现此过程。例如，一个简单的算法可以简单评估最高优先级应用程序的状态并且根据该应用程序的状态做出决定。如果该最高优先级应用程序正确执行时，决策代理可以决定该监控代理应该发送心跳，而不管任何较低优先级应用程序的状态如何。相似地，即使所有较低优先级应用程序正确运行，如果最高优先级应用程序发生故障，决策代理可以决定监控代理不应当发送心跳信息。在此实例中，所有权重都给予最高优先级应用程序的状态。

[0042] 其他算法可以比上述例子分配更多权重给较低优先级的应用程序，上述例子中除了最高优先级的应用程序外，不分配任何权重给任意应用程序。例如，一个算法可以将 VM 内正在被监控的发生故障的应用程序的优先级总和与 VM 内正在被监控的正确执行的应用程序优先级总和进行比较。如果对应于发生故障的应用程序总和更大，决策代理可以决定停止发送心跳。相似地，如果对应于正确执行的应用程序总和更大，决策代理可以决定继续发送心跳。如果总和相等，在一个实施方案中，决策代理可以决定停止发送心跳。另一个算法可以将正确执行和有意脱机的应用程序的优先级总和与发生故障的应用程序优先级总和进行比较。如果前者的总和更大，决策代理将使监控代理继续发送心跳。如果后者总和更大，决策代理将阻止心跳的发送。

[0043] 通过将多个应用程序的状态压缩到一个单一心跳通信信道，决策代理可以有效地在一个单一通信信道上将多个应用程序状态多路复用到一个单一心跳信息。0}？这允许系统基于配置的优先级提供多个应用程序的高可用性，尽管只有一个单一心跳通信信道。

[0044] 在以上实例中，如果应用程序发生故障，标准的响应是停止发送心跳信息，从而使 VM (其中发生故障的应用程序执行)重启。在另一个实施方案中，代替自动使包含故障应用

程序的 VM 重启的是，决策代理相反可以首先决定监控代理应该尝试重启故障应用程序。决策代理可以被配置（例如，管理员通过接口输入信息）为具有最大数量的重试，以便允许在决策代理决定 VM 应该重启之前可以控制这种决定到达故障应用程序的次数。为了给应用程序重新启动的时间，在监控代理尝试重新启动应用程序的同时，决策代理可以指示监控代理向虚拟化控制器注销。

[0045] 如上所提及，用于控制每个决策代理的不同信息可以由管理员通过接口输入信息进行配置。这些信息在 VM 之间和应用程序之间可以不同。因此，在一个 VM 中，管理员可以选择使应用程序 A 比分配到另一个 VM 中的应用程序 A 的实例具有更高的优先权。

[0046] 图 2 展示了虚拟化系统的另一个例子。该实例说明了监控代理和决策代理的功能可以如何分离，这样决策代理在不同于监控代理的计算装置上执行。此外，该实例说明了在监控代理和决策代理之间不需要一一对应的关系。

[0047] 在图 2 的实例中，服务器 70 以及主机 10(1) 和 10(2) 以类似于图 1 所示的方式进行配置。然而，在每个 VM 内执行的不是一个独立的决策代理，而是一个单独的决策代理 60 在计算装置 90 上执行，该装置与主机 10(1) 和 10(2) 相连接以进行通信。在此实例中，每个监控代理 50(1)-50(4) 被配置用于将监控代理产生的信息发送到单一的决策代理 60，然后决策代理利用类似上述的技术处理该信息并产生决定。接着决策代理 60 将描述其决定的信息返回给合适的监控代理。例如，响应于监控代理 50(3) 接收到的指示该应用程序 40(4)（具有比应用程序 40(3) 更高的优先级）发生故障的信息，决策代理 60 可以决定监控代理 50(3) 应该停止向虚拟化控制器 70 发送心跳信息。响应于接收到的识别该决定的信息，监控代理 50(3) 将终止向虚拟化控制器 80 发送心跳信息，这将继而使虚拟化控制器 80 重启不同主机（例如，主机 10(1)）上的 VM30(3)。

[0048] 如图 1 所示的实例，管理员可以将决策代理 60 配置为在每个 VM 上不同的优先级用于不同应用程序。管理员可以为每个应用程序和每个 VM 提供不同的信息。决策代理可以选择使用哪个优先级和其他信息（例如，在决定对故障应用程序在其中执行的 VM 进行故障转移之前，尝试重启应用程序的次数）来处理从监控代理接收的一组给定信息（基于哪个监控代理发送了该信息）。因此，响应于从监控代理 50(1) 接收信息，决策代理 60 可以选择使用与 VM30(1) 相关的信息以处理接收的信息。

[0049] 图 3 展示了一种在虚拟环境中提供高可用性应用程序的方法。该方法可以通过结合决策代理操作的监控代理来实现。如上图所示，这些组件可以或不可以在与彼此相同的计算装置上执行。

[0050] 当在虚拟机内执行的监控代理检测到在该虚拟机内执行的应用程序状态时，该方法开始，以 300 示出。操作 300 的执行可可以包括监控另一个服务组（包括被监控的应用程序）的状态的监控服务组代理。描述操作 300 的结果的信息可以从监控代理发送至决策代理，决策代理可以或可以不与监控代理相集成。

[0051] 如果被监控的应用程序正确执行，如在 305 所确定的，可以针对是否当前注册了监控代理以向监控虚拟机的虚拟化控制器提供心跳信息而做出决定，以 310 示出。操作 305 和 310 可以通过决策代理来实现，决策代理处理由监控代理产生的信息。

[0052] 如果当前并未注册监控代理目来提供心跳信息，监控代理将注册以提供心跳信息，以 315 示出，然后在当前周期发送心跳。如果当前注册了监控代理以提供心跳，监控代

理将当前周期的心跳信息发送至虚拟化控制器，以 320 示出。操作 315 和 320 的执行可以包括决策代理决定采取的动作并将该动作传输给监控代理，然后监控代理采取决策代理所选择的动作。

[0053] 如果应用程序未正确执行，可以针对是否应用程序发生故障或已经有意脱机来做出决定，以 325 示出。操作 325 的执行包括监控代理检测应用程序的状态（例如，在应用程序已经有意脱机的情况下利用管理员输入的信息）以及决策代理处理该信息。

[0054] 如果应用程序发生故障，监控代理将注册以提供心跳（如果尚未注册的话），以 340 和 345 示出。那么监控代理将不向虚拟化控制器发送心跳信息，以 330 示出。

[0055] 相反，如果应用程序已经有意脱机，监控代理将注销发送心跳信息给虚拟化控制器，以 335 示出。操作 330 和 335 的执行包括决策代理决定采取的动作并将该决定传递给监控代理，然后监控代理采取决策代理所选择的动作。

[0056] 图 3 的方法（以及以下描述的图 4 的方法同样）可以每一次心跳周期重复一次。因此，例如在一个周期，可以检测到应用程序正在正常运行，而在下个周期，可以检测到相同的应用程序已经有意脱机了。作为响应，监控代理可以注销。此后的几个周期，可以再次检测到应用程序在线并且监控代理可以注册以再次提供心跳。此后一段时间，监控代理可以检测应用程序发生故障，并且监控代理可以通过停止发送心跳使应用程序重启（通过使应用程序在其中执行的虚拟机重新启动）。

[0057] 图 4 是一种在虚拟环境下提供多个应用程序的高可用性的方法的实例。类似图 3 的方法，该方法可以通过与决策代理相结合操作的监控代理来实现。

[0058] 图 3 所示的方法在 400 开始，此时监控代理检测在与检测代理相同的虚拟机内执行的若干被监控应用程序中每一个的运行状态（例如，正确执行，故障，或有意脱机）。决策代理可以检测那些被监控应用程序中每一个的优先级（例如，通过访问管理员输入的配置信息）。

[0059] 在 405，决策代理使用识别应用程序的运行状态的信息和优先级来确定所希望的一个或多个应用程序组是否正确执行。决策代理可以使用上述算法之一做出该决定，或使用考虑了应用程序优先级的任何其他算法。

[0060] 如果所需的应用程序正在执行并且当前注册了监控代理以向虚拟化控制器提供用于虚拟机的心跳，决策代理决定监控代理应该发送当前周期的心跳信息，以 420 示出。如果监控代理当前未注册，监控代理将注册（415），然后发送心跳信息（420）。

[0061] 如果所需的应用程序未正确执行，决策代理确定（例如，通过寻找最高优先级应用程序的运行状态，通过比较发生故障和有意脱机的应用程序的优先级总和等）是否所需应用程序发生故障或已经有意脱机，以 425 示出。如果所需的应用程序发生故障，监控代理将注册以提供心跳（如果未注册的话），以 440 和 445 示出。决策代理可以使监控代理不发送当前周期（430）的心跳信息，这将继而导致虚拟化控制器重启 VM，在该 VM 中故障应用程序在另一个主机被执行重启。相反如果所需的应用程序已经有意脱机，决策代理可以使监控代理向虚拟化控制器注销，以 435 示出。

[0062] 图 5 是能够实现如上所述的监控代理和 / 或决策代理的计算系统 510 的框图。计算系统 510 广义上代表能够执行计算机可读指令的任何单处理器或多处理器的计算装置或系统。计算系统 510 的实例包括（但不限于）各种装置中的任意一个或多个，这些装置包

括工作站、个人计算机、膝上型计算机、客户端侧终端、服务器、分布式计算系统、手持式装置(例如,个人数字助理以及移动电话)、网络设备、存储控制器(例如,阵列控制器、磁带驱动控制器、或硬盘驱动控制器)等。在其最基本的配置中,计算系统 510 可以包括至少一个处理器 514 和一个系统内存 516。通过执行实现监控代理和 / 或决策代理的软件,计算系统 510 成为一个专用计算装置,该装置被配置为在虚拟环境中提供一个或多个应用程序的高可用性。

[0063] 处理器 514 总体上代表能够处理数据或解释并执行多个指令的任何类型或形式的处理单元。在某些实施方案中,处理器 514 可以从一个软件应用程序或模块中接收指令。这些指令可以使处理器 514 执行在此所说明和 / 或展示的这些示例性实施方案中的一个或多个的功能。例如,处理器 514 可以执行和 / 或作为一种手段用于执行此处所描述的操作。处理器 514 还可以执行和 / 或作为一种手段来执行在此说明和 / 或展示的任何其他操作、方法、或过程。

[0064] 系统内存 516 总体上代表能够存储数据和 / 或其他计算机可读指令的任何类型或形式的易失性或非易失性存储装置或媒质。系统内存 516 的多个实例包括(但不限于)随机存取存储器(RAM)、只读存储器(ROM)、闪存、或任何其他适当的存储装置。尽管未作要求,在某些实施方案中计算系统 510 可以既包括一个易失性内存单元(例如像系统内存 516)又包括一个非易失性存储装置(例如像以下详细说明的主存储装置 532)。在一个实例中,监控代理 50 中的一个或多个(例如,图 5 和 2 的监控代理 50(1)-50(4) 中的一个)或决策代理 60 (例如,图 1 的决策代理 60(1)-60(4) 中的一个或图 2 的决策代理 60)可以被加载到系统内存 516 中。

[0065] 在某些实施方案中,除了处理器 514 和系统内存 516 外,计算系统 510 还可以包括一个或多个组件或元件。例如,如图 5 所示,计算系统 510 可以包括内存控制器 518、输入 / 输出(I/O)控制器 520、以及通信接口 522,它们中的每一个可以通过通信基础设施 512 相互连接。通信基础设施 512 总体上代表能够帮助在计算装置的一个或多个组件之间进行通信的任意类型或形式的基础设施。通信基础设施 512 的实例包括但不限于通信总线(例如工业标准体系结构(ISA)、外围组件互联(PCI)、第三代总线标准(PCIe)、或类似总线)和一个网络。

[0066] 内存控制器 518 总体上代表在计算系统 510 的一个或多个组件之间操作内存或数据或者控制通信的任意类型或形式的装置。例如,在某些实施方案中,内存控制器 518 可以通过通信基础设施 512 控制处理器 514、系统内存 516 以及 I/O 控制器 520 之间的通信。在某些实施方案中,内存控制器 518 可以独立地或与其他元件相结合地执行和 / 或作为一种手段执行在此描述和 / 或展示的多个步骤或特征中的一个或多个。

[0067] I/O 控制器 520 总体上代表能够协调和 / 或控制一种计算装置的输入和输出功能的任何类型或形式的模块。例如,在一些实施方案中 I/O 控制器 520 可以控制或协助在计算系统 510 的一个或多个元件(如处理器 514、系统内存 516、通信接口 522、显示适配器 526、输入接口 530、以及存储接口 534)之间的数据传送。

[0068] 通信接口 522 广义地代表能够协助计算系统 510 与一个或多个附加装置之间进行通信的任何类型或形式的通信装置或适配器。例如,在某些实施方案中,通信接口 522 可以协助计算系统 510 与包括多个附加的计算系统的私人或公共网络之间的通信。通信接口

522 的实例包括而不限于有线网络接口(例如网络接口卡)、无线网络接口(例如无线网络接口卡)、调制解调器、以及任何其他适当的接口。在至少一个实施方案中,通信接口 522 可通过到网络(如互联网)的直接链接来提供到一台远程服务器的直接连接。通信接口 522 还可以间接地提供这种连接,例如通过局域网(如以太网)、个人局域网、电话或缆线网、蜂窝电话连接、卫星数据连接、或任何其他适当的连接。

[0069] 在某些实施方案中,通信接口 522 还可以代表一种主机适配器,该主机适配器被配置为用于通过一条外部总线或通信信道协助计算系统 510 与一个或多个附加网络或存储装置之间的通信。主机适配器的实例包括,但不限于,小型计算机系统接口(SCSI)主机适配器、通用串行总线(USB)主机设配器、电气和电子工程学会(IEEE)1394 主机适配器、串行高级技术附件(SATA)和外部 SATA (eSATA)主机适配器、高级技术附件(ATA)和并行 ATA (PATA)主机适配器、光纤通道接口适配器、以太网适配器等。

[0070] 通信接口 522 还可以允许计算系统 510 参与分布式计算或远程计算。例如,通信接口 522 可以从一个远程装置接收指令或向一个远程装置发送指令用于执行。

[0071] 如图 5 所示,计算系统 510 还可以包括通过显示适配器 526 连接至通信基础设施 512 的至少一个显示装置 524。显示装置 524 总体上代表能够可视地呈现显示适配器 526 所转发的显示信息的任意类型或形式的装置。相似地,显示适配器 526 总体上代表任意类型或形式的装置,这些装置被配置用于从通信基础设施 512 (或从本领域已知的帧缓冲器)转发图形、文本以及其他数据以便显示在显示装置 524 上。

[0072] 如图 5 所示,计算系统 510 还可以包括通过输入接口 530 连接至通信基础设施 512 的至少一个输入装置 528。输入装置 528 总体上代表能够向示例性计算系统 510 提供由计算机或人员生成的输入的任意类型或形式的输入装置。输入装置 528 的实例包括但不限于键盘、定位装置、语音识别装置或任意其他输入装置。

[0073] 如图 5 所示,计算系统 510 还包括通过存储接口 534 连接至通信基础设施 512 的一个主存储装置 532 和一个备份存储装置 533。存储装置 532 和 533 总体上代表能够存储数据和 / 或其他计算机可读指令的任意类型或形式的存储装置或介质。例如,存储装置 532 与 533 可以是磁盘驱动器(例如,所谓的硬盘驱动器)、软盘驱动器、磁带驱动器、光盘驱动器、闪存驱动器、或者类似装置。存储接口 534 总体上代表用于在存储装置 532 和 533 和计算装置 510 的其他组件之间传输数据的任意类型或形式的接口或装置。类似于主存储装置 532 的存储装置可以存储信息,如配置信息 590 (例如,如上文所述,配置信息表示应用程序的优先级和每个应用程序重新尝试的次数)。

[0074] 在某些实施方案中,存储装置 532 和 533 可以被配置为用于读取自和 / 或写入到一个可移动存储单元,该可移动存储单元被配置为用于存储计算机软件、数据、或其他计算机可读信息。适合的可移动存储单元的实例包括但不限于软盘、磁带、光盘、闪存装置等等。存储装置 532 和 533 还可以包括其他类似的结构或装置,以允许计算机软件、数据或其他计算机可读指令下载到计算系统 510 中。例如,存储装置 532 和 533 可以被配置用于读或写软件、数据或其他计算机可读信息。存储装置 532 和 533 还可以作为计算系统 510 的一部分或可以是通过其他接口系统访问的一个分离的装置。

[0075] 很多其他装置或子系统可以连接至计算系统 510 上。相反地,为了实施在此描述和 / 展示的实施方案,不需要图 5 中所示的所有组件和装置。以上提到的这些装置和子系

统还能够以不同于图 5 中所示的方式进行相互连接。

[0076] 计算系统 510 还可使用任何数目的软件、固件、和 / 或硬件的配置。例如，在此披露的示例性实施方案中的一个或多个可以被编码为一种计算机可读媒质上的计算机程序（也称为计算机软件、软件应用程序、计算机可读指令、或计算机控制逻辑）。计算机可读存储介质的实例包括磁存储介质（例如硬盘驱动器和软盘）、光存储介质（例如，CD 或 DVD-ROM）、电存储介质（例如，固态驱动器和闪存）等。此类计算程序也可以通过网络（例如因特网或载体介质）传输到计算系统 510 以存储在内存中。

[0077] 包含计算机程序的计算机可读媒质可以载入到计算系统 510 中。存储在计算机可读媒质上的所有或部分计算机程序然后可以存储在系统内存 516 和 / 或存储装置 532 和 533 的不同部分上。当由处理器 514 执行时，载入到计算系统 510 中的计算机程序可以使处理器 514 执行和 / 或作为一种手段执行在此描述和 / 或展示的示例性实施方案中的一个或多个的功能。另外或可替代地，在此所说明和 / 或展示的示例性实施方案中的一个或多个可以在固件和 / 或硬件中实施。例如，可以将计算系统 510 配置为一种专用集成电路（ASIC），该电路被适配为用于实施在此所披露的这些示例性实施方案中的一个或多个。

[0078] 图 6 是网络体系结构 600 的框图，在该网络体系结构中客户端系统 610、620 和 630 以及服务器 640 和 645 可以连接至网络 650。客户端系统 610、620 和 630 总体上代表任意类型或形式的计算装置或系统，例如图 5 中的计算系统 510。

[0079] 类似地，服务器 640 和 645 总体上代表被配置为用于提供不同的数据库服务和 / 或运行某种软件应用程序的计算装置或系统，如应用程序服务器或数据库服务器。网络 650 总体上代表任何电信或计算机网络，例如包括：内部网、广域网（WAN）、局域网（LAN）、个人区域网（PAN）、或互联网。在一个实例中，客户端系统 610、620 和 / 或 630 和 / 或服务器 640 和 / 或 645 可以包括如图 1 和 2 所示的监控代理和 / 或决策代理。

[0080] 如图 6 所示，一个或多个存储装置 660(1)-(N) 可以直接附接至服务器 640。类似地，一个或多个存储装置 670(1)-(N) 可以直接附接至服务器 645。存储装置 660(1)-(N) 和存储装置 670(1)-(N) 总体上代表能够存储数据和 / 或其他计算机可读指令的任意类型或形式的存储装置或媒质。在某些实施方案中，存储装置 660(1)-(N) 和存储装置 670(1)-(N) 可以代表网络附联存储（NAS）装置，这些装置被配置为利用不同协议（例如网络文件系统（NFS）、服务器消息块（SMB）、或公共互联网文件系统（CIFS））与服务器 640 和 645 进行通信。

[0081] 服务器 640 和 645 还可以连接至存储区域网络（SAN）结构 680。SAN 结构 680 总体上代表能够协助多个存储装置之间通信的任意类型或形式的计算机网络或体系结构。SAN 结构 680 可以协助服务器 640 和 645 与多个存储装置 690(1)-(N) 和 / 或智能存储器阵列 695 之间的通信。SAN 结构 680 还可以通过网络 650 和服务器 640 和 645 协助客户端系统 610、620 和 630 与存储装置 690(1)-(N) 和 / 或智能存储器阵列 695 之间的通信，其方式为装置 690(1)-(N) 以及阵列 695 对客户端系统 610、620 和 630 呈现为本地附接的装置。与存储装置 660(1)-(N) 和存储装置 670(1)-(N) 一样，存储装置 690(1)-(N) 和存储阵列 695 总体上代表能够存储数据和 / 或其他计算机可读指令的任意类型或形式的存储装置或媒质。

[0082] 在某些实施方案中，参考图 5 的计算系统 510，通信接口（例如图 5 中的通信接口 522）可以用来在每个客户端系统 610、620 和 630 以及网络 650 之间提供连接。客户端系

统 610、620 和 630 能够利用例如网络浏览器或其他客户端软件访问服务器 640 或 645 的信息。这种软件可以允许客户端系统 610、620 和 630 访问由服务器 640、服务器 645、存储装置 660(1)-(N)、存储装置 670(1)-(N)、存储装置 690(1)-(N) 或智能存储器阵列 695 管理的数据。尽管图 6 描绘了使用网络(例如互联网)来交换数据,但在此描述和 / 或展示的实施方案不限于互联网或任意具体的基于网络的环境。

[0083] 在至少一个实施方案中,在此披露的示例性实施方案中的一个或多个的全部或一部分可被编码为一种计算机程序并且由服务器 640、服务器 645、存储装置 660(1)-(N)、存储装置 670(1)-(N)、存储装置 690(1)-(N)、智能存储阵列 695、或它们中的任意组合加载并执行。在此披露的示例性实施方案中的一个或多个的全部或一部分还可以被编码成为一种计算机程序,它存储在服务器 640 中、由服务器 645 来运行、并在网络 650 上分发给客户端系统 610、620、和 630。

[0084] 在一些实例中,图 1 中示例性系统 100 的全部或一部分可以代表云计算的或基于网络的环境的多个部分。云计算环境可以通过互联网提供各种服务和应用程序。这些基于云的服务(例如,软件即服务、平台即服务、基础设施即服务等)可以通过网络浏览器或其他的远程接口进行访问。在此所述的不同功能可以通过远程桌面环境或任何其他的基于云的计算环境来提供。

[0085] 另外,在此所述的这些模块中的一个或多个可以将数据、物理装置、和 / 或物理装置的表示从一种形式转换到另一种形式。例如,结合监控代理进行操作的决策代理可以通过控制心跳信息的传输来改变虚拟化系统的配置,方式为使虚拟机在另一台主机上重新启动。

[0086] 尽管已经结合一些实施方案描述了本发明,但无意将本发明限制于本文阐述的具体形式。相反的是,本发明意在涵盖可以合理地包含在所附权利要求定义的本发明范围内的这些替代形式、修改形式以及等效形式。

[0087] 工业适用性

[0088] 本发明适用于计算装置和网络计算装置领域。

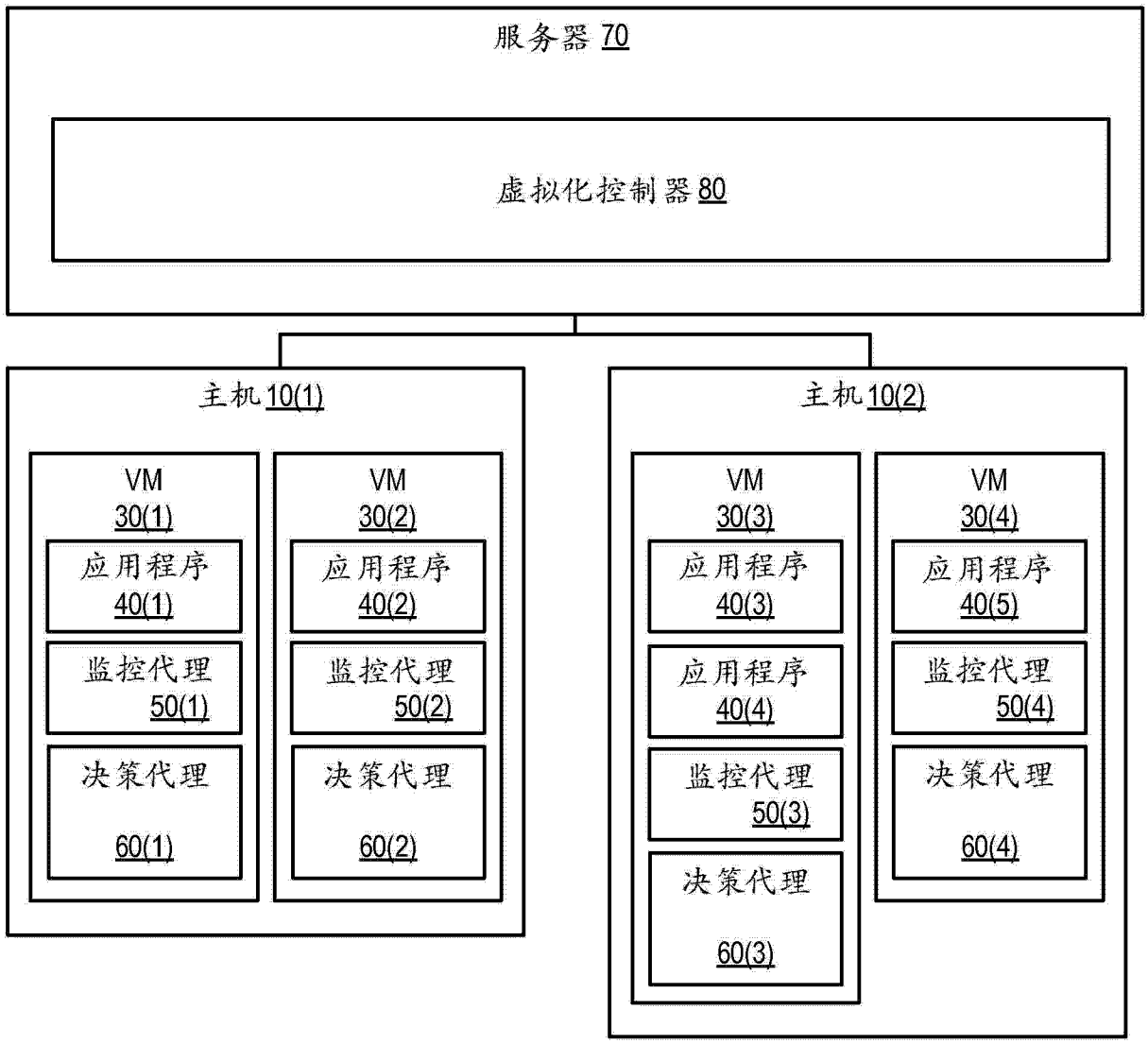

服务器 70
虚拟化控制器 80
主机 10(1)
VM 30(1)
应用程序 40(1)
监控代理 50(1)
决策代理 60(1)
VM 30(2)
应用程序 40(2)
监控代理 50(2)
决策代理 60(2)
主机 10(2)
VM 30(3)
应用程序 40(3)
应用程序 40(4)
监控代理 50(3)
决策代理 60(3)
VM 30(4)
应用程序 40(5)
监控代理 50(4)
决策代理 60(4)

图 1

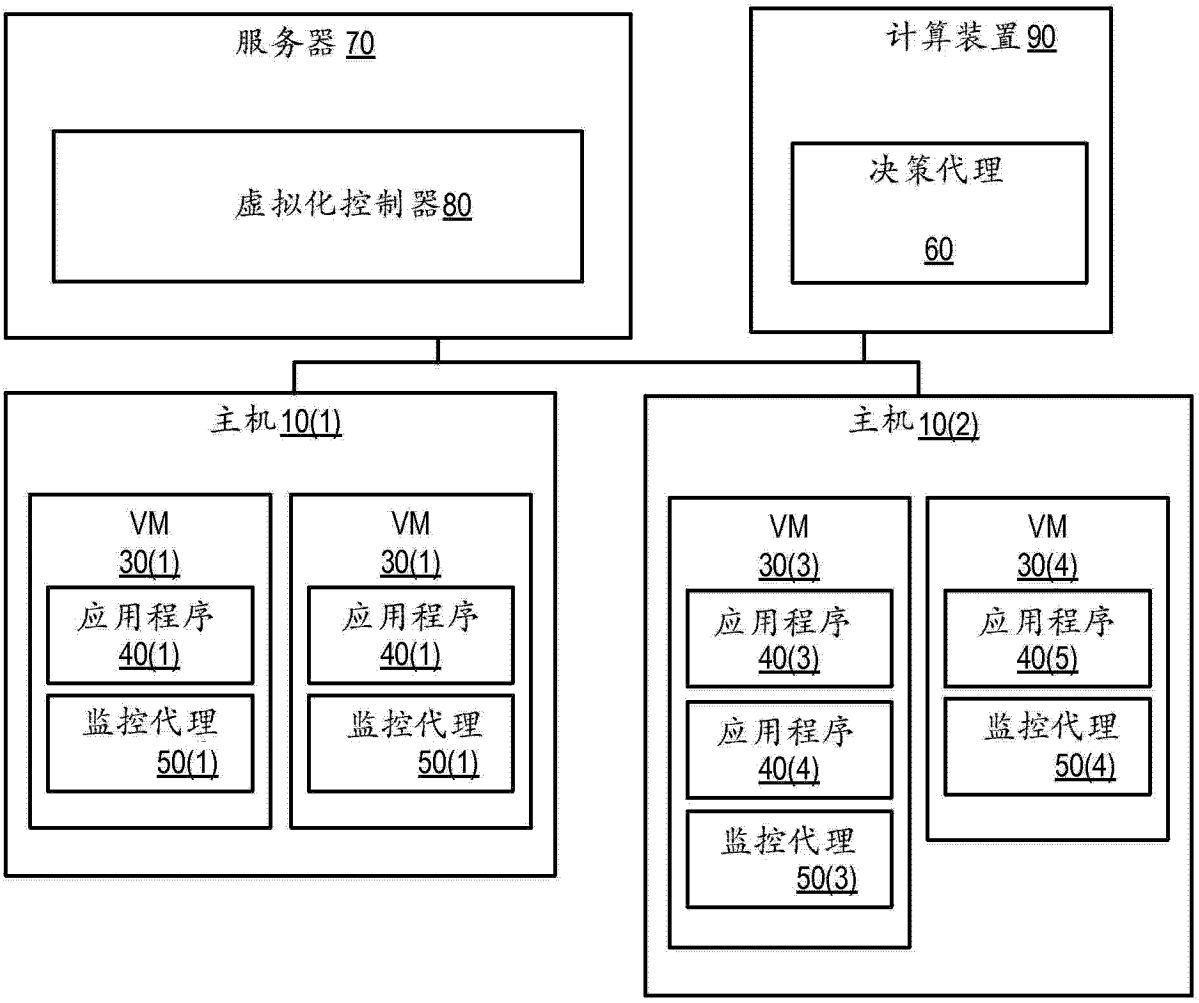

服务器 70
虚拟化控制器80
计算装置90
决策代理
60
主机10(1)
VM
30(1)
应用程序
40(1)
监控代理
50(1)
VM
30(1)
应用程序
40(1)
监控代理
50(1)
主机10(2)
VM
30(3)
应用程序
40(3)
应用程序
40(4)
监控代理
50(3)
VM
30(4)
应用程序
40(5)
监控代理
50(4)

图 2

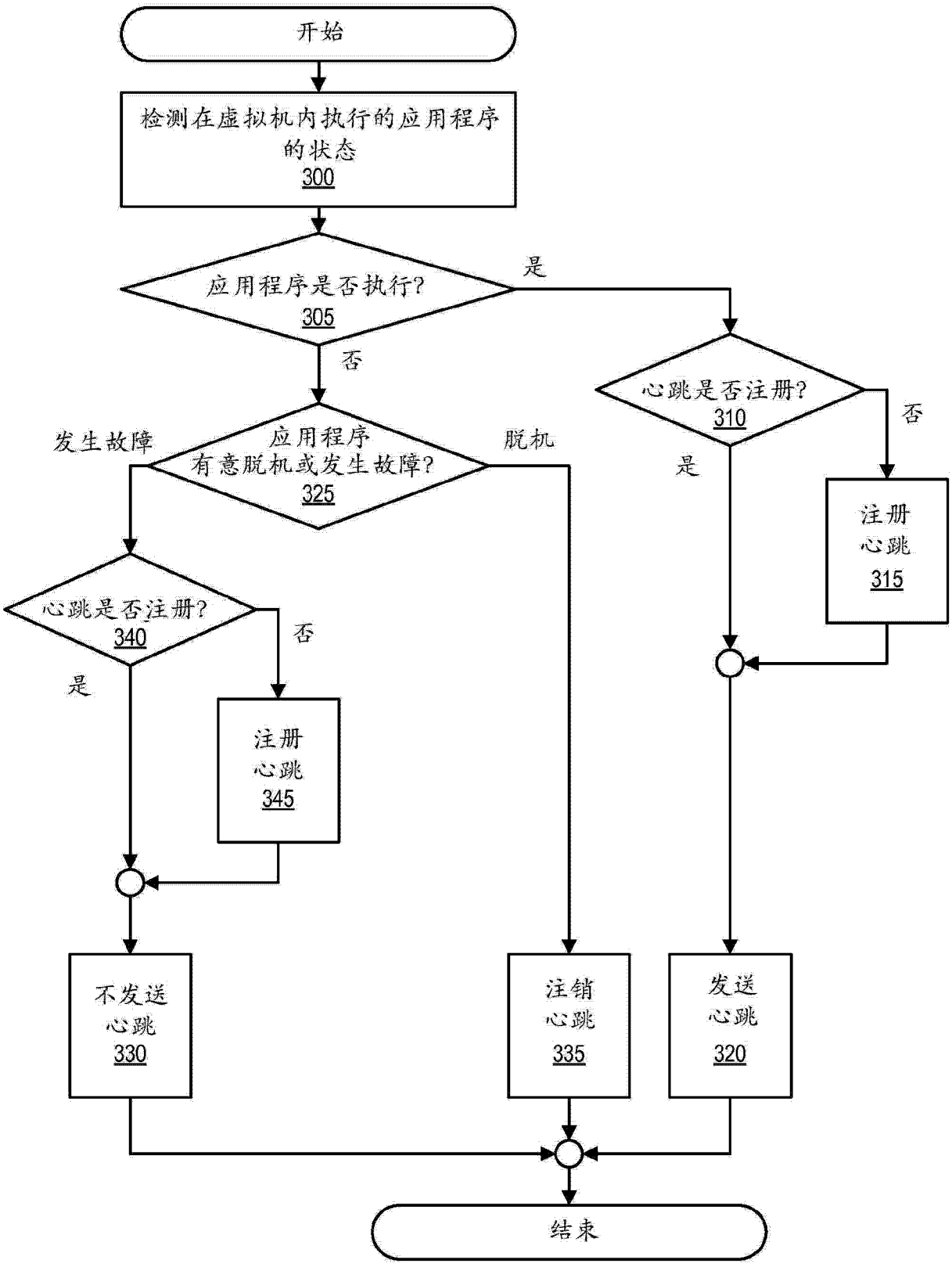

开始
检测在虚拟机内执行的应用程序的状态
300
应用程序是否执行?
305
是
否
心跳是否注册?
310
否
是
注册心跳
315
发生故障
应用程序有意脱机或发生故障?
325
脱机
心跳是否注册?
340
否
是
注册心跳
345
不发送心跳
330
注销心跳
335
发送心跳
320
结束

图 3

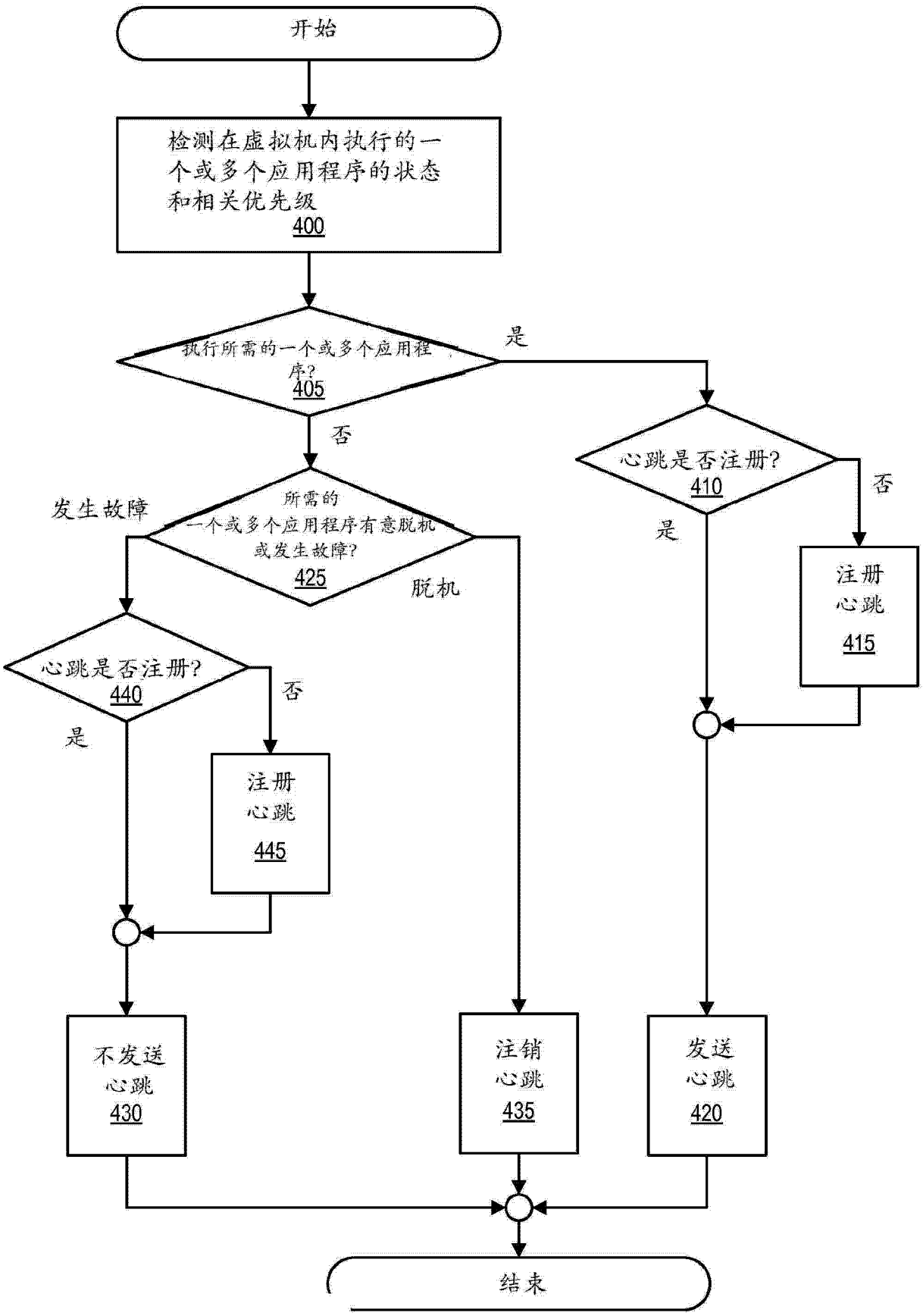

开始
检测在虚拟机内执行的一个或多个应用程序的状态和相关优先级
400
执行所需的一个或多个应用程序?
405
是
否
心跳是否注册?
410
否
是
注册心跳
415
发生故障
所需的一个或多个应用程序有意脱机或发生故障?
425
脱机
心跳是否注册?
440
否
是
注册心跳
445
不发送心跳
430
注销心跳
435
发送心跳
420
结束

图 4

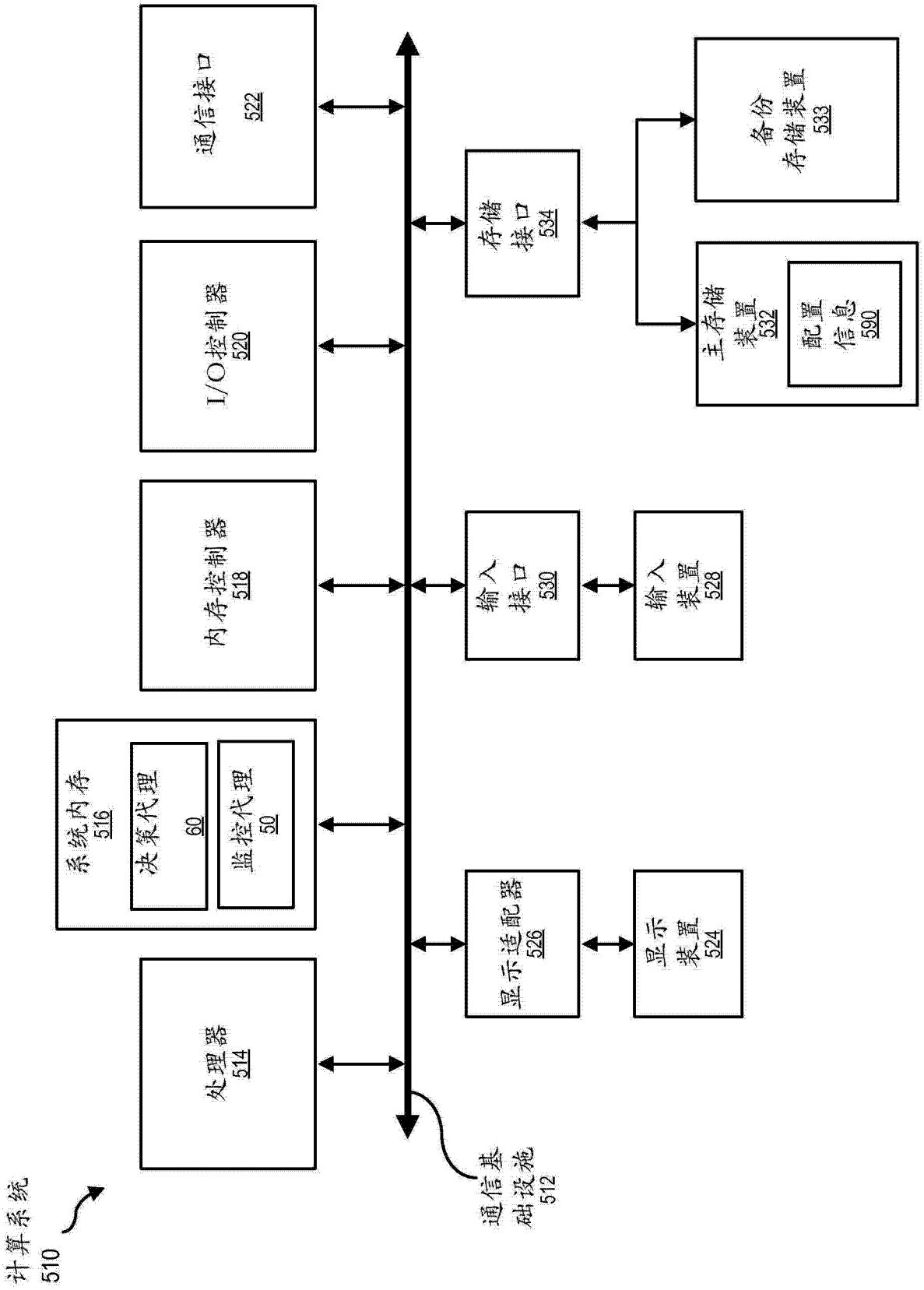

计算系统
510
处理器
514
系统内存
516
决策代理
60
监控代理
50
内存控制器
518
I/O控制器
520
通信接口
522
通信基础设施
512
显示适配器
526
显示装置
524
输入接口
530
输入装置
528
存储接口
534
主存储装置
532
配置信息
590
备份存储装置
533

图 5

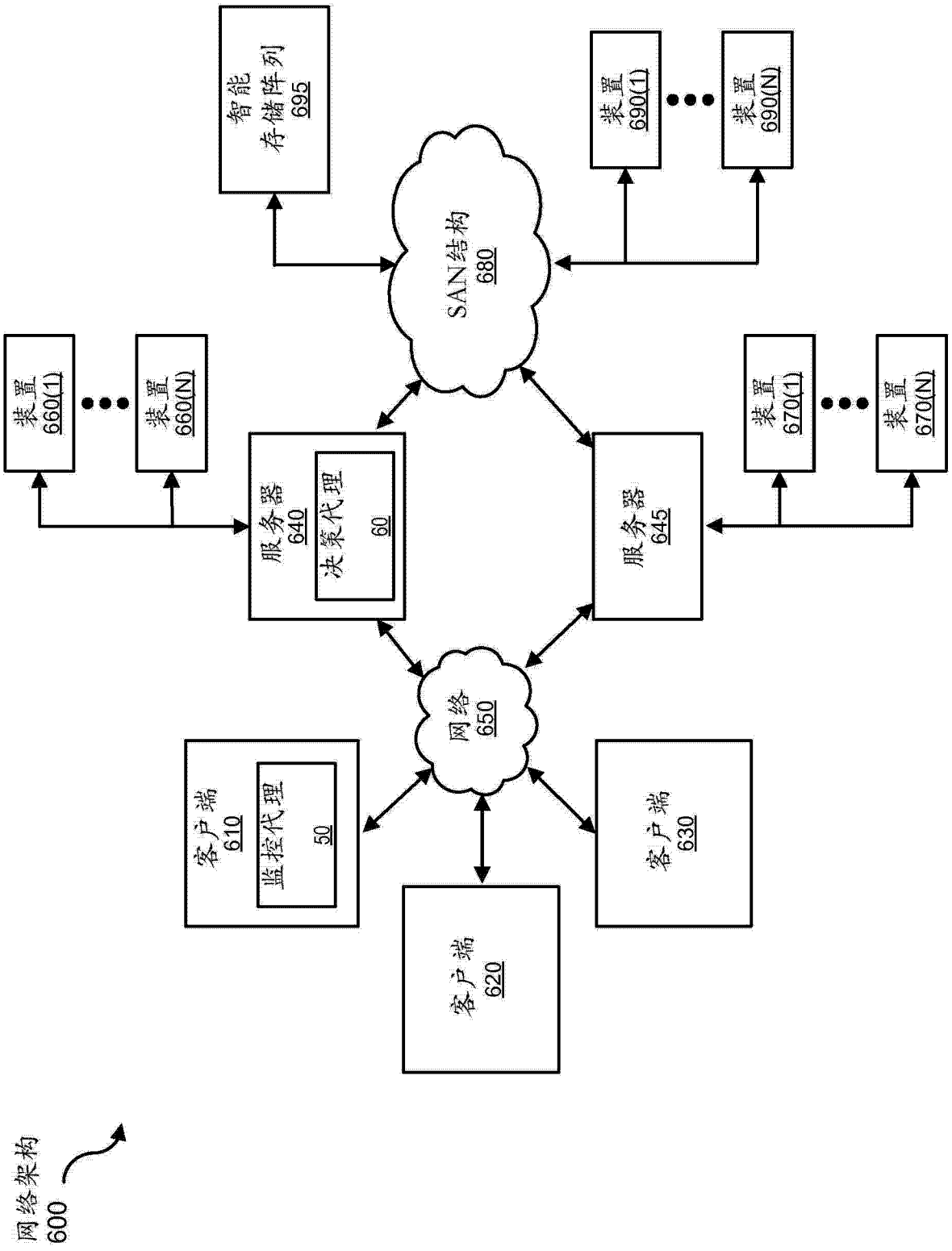

智能存储阵列 695
装置 690(1)
装置 690(N)
SAN结构 680
装置 660(1)
装置 660(N)
服务器 640
决策代理 60
服务器 645
装置 670(1)
装置 670(N)
网络 650
客户端 610
监控代理 50
客户端 630
客户端 620
网络架构 600

图 6