

(72)发明人 T·班多帕德里亚 B·迪格

(74) 专利代理机构 北京纪凯知识产权代理有限公司 11245

专利代理师 徐东升

(51) Int.Cl.

G06F 11/14 (2006.01)

G06F 11/20 (2006.01)

(56) 对比文件

US 7386752 B1, 2008.06.10

CN 104052789 A, 2014.09.17

US 2012066487 A1, 2012.03.15

US 2014074973 A1, 2014.03.13

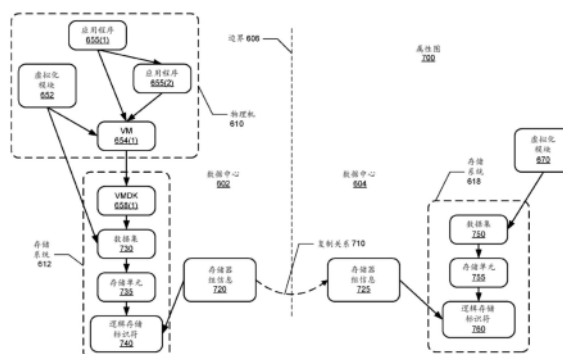
WO 2016121802 A1, 2016.08.04

审查员 余芳芳

权利要求书5页 说明书25页 附图18页

用于编排的工作负载的依赖性分析的方法和系统

本发明公开了提供实体及其关系的表示和分析的方法、计算机程序产品、计算机系统等。该方法例如包括选择属性图中的多个节点中的节点,分析与该节点相关联的实体信息,并且响应于该分析,将实体信息存储在依赖性信息中。多个节点中的每个节点表示多个实体中的实体,并且通过属性图的多个链路中的一个或多个链路耦接到多个节点中的至少一个其他节点。一个或多个链路中的每个链路表示每个节点与至少一个其他节点之间的关系。实体信息是关于实体的信息。依赖性信息被配置为便于多个实体中的一个或多个实体的编排。



1. 一种计算机实现的方法,包括:

分析属性图,包括

在所述属性图中选择多个节点中的第一节点,其中

所述多个节点中的每个节点表示多个实体的实体,

所述多个节点包括所述第一节点,

所述第一节点表示第一数据中心基础设施中的第一实体,

所述多个节点中的每个节点通过所述属性图的多个链路中的一个或多个链路耦接到所述多个节点中的至少一个其他节点,

通过使每个链路耦接到所述多个节点中的一些节点,所述多个链路中的每个链路表示所述多个实体中的一些实体之间的一个或多个依赖性关系中的相应依赖性关系,并且

如果一个实体被包含在所述第一数据中心基础设施中并且另一实体被包含在第二数据中心基础设施中,则所述一个实体和所述另一实体之间的依赖性关系表示所述第一数据中心基础设施和所述第二数据中心基础设施之间的接触点;

确定所述第一节点和表示所述第二数据中心基础设施中的第二实体的第二节点之间的第一依赖性关系是否表示所述第一数据中心基础设施和所述第二数据中心基础设施之间的第一接触点,其中

所述确定包括

分析与所述多个节点相关联的实体信息,其中

所述实体信息是关于所述第一实体的信息,以及

分析所述实体信息包括

确定所述第一节点和所述第二节点之间的链路是否跨越所述第一数据中心基础设施和所述第二数据中心基础设施之间的边界,以及

响应于确定所述链路跨越所述边界,将所述第一实体和/或所述第二实体的所述实体信息存储在表示所述第一实体和所述第二实体之间的第一依赖性关系的依赖性信息中,其中

所述存储包括

将接触点记录在所述依赖性信息中,并且

所述依赖性信息被配置为便于编排操作;以及

响应于所述第一依赖性关系表示所述第一数据中心基础设施和所述第二数据中心基础设施之间的接触点,

对所述第一实体执行所述编排操作,并且根据所述第一依赖性关系,对所述第二实体执行所述编排操作。

2. 根据权利要求1所述的方法,其中:

所述第一依赖性关系是以下之一:

低级复制关系,

高级副本关系,或者

高级同步关系。

3. 根据权利要求1所述的方法,还包括:

通过遍历所述属性图从所述第一节点到所述多个节点中的另一个节点的链路识别所

述多个链路中跨越所述边界的其他链路,其中

所述链路是所述多个链路中的一个链路;

确定与跨越所述边界的所述其他链路中的另一链路相关联的节点是否应该被包含在所述依赖性信息中;以及

响应于确定与跨越所述边界的所述另一链路相关联的所述第一节点应该被包含在所述依赖性信息中,

将针对具有跨越所述边界的所述链路的所述第一节点的另一接触点记录在所述依赖性信息中。

4. 根据权利要求3所述的方法,其中

选择所述第一节点作为所述属性图中的起点。

5. 根据权利要求4所述的方法,其中:

所述第一依赖性关系是以下之一:

低级复制关系,

高级副本关系,或者

高级同步关系。

6. 根据权利要求3所述的方法,其中所述遍历包括:

确定所述另一个节点是否应该包括在所述依赖性信息中;

响应于确定所述另一个节点应该包括在所述依赖性信息中,执行所述遍历;以及

响应于确定所述另一个节点不应该包括在所述依赖性信息中,防止执行所述遍历。

7. 根据权利要求6所述的方法,还包括:

确定所述第一节点与所述多个节点中的后续节点之间是否存在另一个链路;以及

响应于确定所述后续节点应该包括在所述依赖性信息中,递归地执行所述遍历。

8. 根据权利要求1所述的方法,其中

所述多个实体中的每个实体为以下中的一者:

虚拟计算部件,

虚拟存储部件,

物理计算部件,或者

物理存储部件。

9. 一种计算机系统,包括:

一个或多个处理器;

分析属性图的分析模块,包括

选择模块,其选择所述属性图中的多个节点中的第一节点,其中

所述多个节点中的每个节点表示多个实体中的实体,

所述多个节点包括所述第一节点,

所述第一节点表示第一数据中心基础设施中的第一实体,

所述多个节点中的每个节点通过所述属性图的多个链路中的一个或多个链路耦接到所述多个节点中的至少一个其他节点,

通过使每个链路耦接到所述多个节点中的一些节点,所述多个链路中的每个链路表示所述多个实体中的一些实体之间的一个或多个依赖性关系中的相应依赖性关系,并且

如果一个实体被包含在所述第一数据中心基础设施中并且另一实体被包含在第二数据中心基础设施中,则所述一个实体和所述另一实体之间的依赖性关系表示所述第一数据中心基础设施和所述第二数据中心基础设施之间的接触点;

确定模块,其确定所述第一节点和表示所述第二数据中心基础设施中的第二实体的第二节点之间的第一依赖性关系是否表示所述第一数据中心基础设施和所述第二数据中心基础设施之间的第一接触点,其中所述确定模块包括

分析模块,其分析与所述多个节点相关联的实体信息,其中

所述实体信息是关于所述第一实体的信息,并且

通过以下方式分析所述实体信息:

确定所述第一节点和所述第二节点之间的链路是否跨越所述第一数据中心基础设施和所述第二数据中心基础设施之间的边界,以及

存储模块,其响应于确定所述链路跨越所述边界将所述第一实体和/或所述第二实体的所述实体信息存储在依赖性信息中,其中

通过以下方式存储所述实体信息:

将接触点记录在所述依赖性信息中,并且

所述依赖性信息被配置为便于编排操作;

执行模块,其响应于所述第一依赖性关系表示所述第一数据中心基础设施和所述第二数据中心基础设施之间的接触点,对所述第一实体执行所述编排操作,并且根据所述第一依赖性关系,对所述第二实体执行所述编排操作;以及

非暂态计算机可读存储介质,其中所述非暂态计算机可读存储介质耦接到所述一个或多个处理器。

10. 根据权利要求9所述的计算机系统,其中:

所述第一依赖性关系是以下之一:

低级复制关系,

高级副本关系,或者

高级同步关系。

11. 根据权利要求9所述的计算机系统,其中所述计算机系统还包括:

识别模块,其识别所述多个链路中跨越所述边界的其他链路,所述识别模块包括遍历模块,所述遍历模块遍历所述属性图从所述第一节点到所述多个节点中的另一个节点的链路,其中

所述链路是所述多个链路中的一个链路;

确定模块,其确定与跨越所述边界的所述其他链路中的另一链路相关联的节点是否应该被包含在所述依赖性信息中;以及

记录模块,其响应于确定与跨越所述边界的所述另一链路相关联的所述第一节点应该被包含在所述依赖性信息中,将针对具有跨越所述边界的所述链路的所述第一节点的另一接触点记录在所述依赖性信息中。

12. 根据权利要求11所述的计算机系统,其中:

选择所述第一节点作为所述属性图中的起点。

13. 根据权利要求12所述的计算机系统,其中:

所述第一依赖性关系是以下之一：

低级复制关系，

高级副本关系，或者

高级同步关系。

14. 根据权利要求11所述的计算机系统，其中所述计算机系统还包括：

第一确定模块，其确定所述另一个节点是否应该包括在所述依赖性信息中；

执行模块，其响应于确定所述另一个节点应该包括在所述依赖性信息中而执行所述遍历；和

防止模块，其响应于确定所述另一个节点不应该包括在所述依赖性信息中而防止执行所述遍历。

15. 根据权利要求14所述的计算机系统，其中所述计算机系统还包括：

第二确定模块，其确定所述第一节点与所述多个节点中的后续节点之间是否存在另一个链路；和

递归执行模块，其响应于确定所述后续节点应该包括在所述依赖性信息中而递归地执行所述遍历。

16. 一种计算机系统，包括：

一个或多个处理器；

计算机可读存储介质，所述计算机可读存储介质耦接到所述一个或多个处理器；和

多个指令，所述多个指令在所述计算机可读存储介质中被编码并且被配置为使得所述一个或多个处理器：

通过以下方式分析属性图：

在所述属性图中选择多个节点中的第一节点，其中

所述多个节点中的每个节点表示多个实体中的实体，

所述多个节点包括所述第一节点，

所述第一节点表示第一数据中心基础设施中的第一实体，

所述多个节点中的每个节点通过所述属性图的多个链路中的一个或多个链路耦接到所述多个节点中的至少一个其他节点，

通过使每个链路耦接到所述多个节点中的一些节点，所述多个链路中的每个链路表示所述多个实体中的一些实体之间的一个或多个依赖性关系中的相应依赖性关系，并且

如果一个实体被包含在所述第一数据中心基础设施中并且另一实体被包含在第二数据中心基础设施中，则所述一个实体和所述另一实体之间的依赖性关系表示所述第一数据中心基础设施和所述第二数据中心基础设施之间的接触点；

确定所述第一节点和表示所述第二数据中心基础设施中的第二实体的第二节点之间的第一依赖性关系是否表示所述第一数据中心基础设施和所述第二数据中心基础设施之间的第一接触点，其中

通过以下方式确定所述第一依赖性关系表示所述第一接触点：

分析与所述多个节点相关联的实体信息，其中

所述实体信息是关于所述第一实体的信息，并且

分析所述实体信息包括

确定所述第一节点和所述第二节点之间的链路是否跨越所述第一数据中心基础设施和所述第二数据中心基础设施之间的边界,以及

响应于确定所述链路跨越所述边界,将所述第一实体和/或所述第二实体的所述实体信息存储在表示所述第一实体和所述第二实体之间的第一依赖性关系的依赖性信息中,其中

所述存储包括

将接触点记录在所述依赖性信息中,并且

所述依赖性信息被配置为便于编排操作;以及

响应于所述第一依赖性关系表示所述第一数据中心基础设施和所述第二数据中心基础设施之间的接触点,对所述第一实体执行所述编排操作,并且根据所述第一依赖性关系,对所述第二实体执行所述编排操作。

17. 根据权利要求16所述的计算机系统,其中所述多个指令还被配置为使得所述一个或多个处理器:

通过遍历所述属性图从所述第一节点到所述多个节点中的另一个节点的链路识别所述多个链路中跨越所述边界的其他链路,其中

所述链路是所述多个链路中的一个链路;

确定与跨越所述边界的所述其他链路中的另一链路相关联的节点是否应该被包含在所述依赖性信息中;以及

响应于确定与跨越所述边界的所述另一链路相关联的所述第一节点应该被包含在所述依赖性信息中,

将针对具有跨越所述边界的所述链路的所述第一节点的另一接触点记录在所述依赖性信息中。

18. 根据权利要求17所述的计算机系统,其中:

所述第一依赖性关系是以下之一:

低级复制关系,

高级副本关系,或者

高级同步关系。

19. 根据权利要求17所述的计算机系统,其中被配置为使得所述一个或多个处理器遍历所述链路的所述多个指令还被配置为使得所述一个或多个处理器:

确定所述另一个节点是否应该包括在所述依赖性信息中;

响应于确定所述另一个节点应该包括在所述依赖性信息中,执行被配置为使得所述一个或多个处理器遍历所述链路的所述多个指令;以及

响应于确定所述另一个节点不应该包括在所述依赖性信息中,防止执行被配置为使得所述一个或多个处理器遍历所述链路的所述多个指令。

用于编排的工作负载的依赖性分析的方法和系统

技术领域

[0001] 本发明涉及计算系统中的工作负载的管理,并且尤其涉及用于编排目的的工作负载的依赖性分析的方法和系统。

背景技术

[0002] 当今的组织经常广泛依赖在线维护的数据。此类经常访问的、不断变化的数据对于此类组织的持续操作至关重要。因此,计算资源不仅包括数据存储,还包括计算系统(使用物理部件和虚拟部件两者实现)。阻止此类数据和系统可用性的计划外事件会严重影响业务操作。另外,任何来自自然灾害或其他事件的永久性数据丢失都可能会对业务的持续可行性产生严重的负面影响。因此,当灾难发生时,组织必须准备好消除或最小化数据丢失和停机时间,并使用可用的数据和功能系统快速恢复。数据备份和其他灾难恢复(DR)技术可用于防止在发生此类灾难时丢失数据和功能性。例如,数据备份过程可用于创建原始数据的副本。这些副本可用于在数据丢失事件后恢复原始数据。在各种其他技术中,可以使用各种技术来生成此类备份,诸如完全备份、增量备份或差异备份以及各种类型的数据复制。如果发生故障,则此类在线数据处理系统应该提供快速、简单、有效的功能性,以便从此类灾难中恢复。类似地,可采用灾难恢复架构来确保服务的连续性。

发明内容

[0003] 本公开描述了使用表示实体和它们彼此之间的一个或多个关系的一个或多个属性图(或类似构造)来支持计算环境分析的方法、计算机程序产品、计算机系统等。该方法例如包括选择属性图中的多个节点中的节点,分析与节点相关联的实体信息,并且响应于分析,将实体信息存储在依赖性信息中。多个节点中的每个节点表示多个实体的实体。多个节点中的每个节点通过属性图的多个链路中的一个或多个链路耦接到多个节点中的至少一个其他节点。一个或多个链路中的每个链路表示每个节点与至少一个其他节点之间的关系。实体信息是关于实体的信息。依赖性信息被配置为便于多个实体中的一个或多个实体的编排。

[0004] 在某些实施方案中,此类方法、计算机程序产品、计算机系统等可以包括确定一个或多个链路是否跨越边界,并且响应于一个或多个链路中的一个链路跨越边界,将接触点记录在依赖性信息中。

[0005] 此外,在某些实施方案中,此类方法、计算机程序产品、计算机系统等可以包括遍历属性图从节点到多个节点中的另一个节点的链路,其中链路是多个链路中的一个链路。此外,可以执行包括以下的操作:确定链路是否跨越边界,并且响应于确定链路跨越边界,将接触点记录在依赖性信息中。可能的情况是选择节点作为属性图中的起点,并且/或者节点和另一个节点彼此具有依赖性关系。

[0006] 此外,在某些实施方案中,此类方法、计算机程序产品、计算机系统等可以包括确定链路是否表示接触点,并且响应于确定链路表示接触点,将接触点记录在依赖性信息中。

该实施方案和其他实施方案可以包括确定是否应该将另一个节点包括在依赖性信息中，(响应于确定应该将另一节点包括在依赖性信息中)执行遍历，并且响应于确定另一个节点不应该包括在依赖性信息中，防止遍历。

[0007] 在某些实施方案中，此类方法、计算机程序产品、计算机系统等可以包括确定节点与后续节点之间是否存在另一个链路，并且响应于确定后续节点应该包括在依赖性信息中，递归地执行遍历。在这些实施方案和其他实施方案中，多个实体中的每个实体可以是虚拟计算部件、虚拟存储部件、物理计算部件或物理存储部件中的一者。

附图说明

[0008] 通过参考附图，可以更好地理解本发明，并且其多个目的、特征和优点对于本领域的技术人员而言是显而易见的。

[0009] 图1是示出根据一个实施方案的包括管理功能性的计算环境的示例的框图。

[0010] 图2是示出根据一个实施方案的支持多个虚拟机 (VM) 和其他功能性的物理机的示例的框图。

[0011] 图3是示出根据一个实施方案的计算环境分析架构的示例的框图。

[0012] 图4是示出根据一个实施方案的计算环境的示例的框图。

[0013] 图5是示出根据一个实施方案的具有多个接触点的计算环境的示例的框图。

[0014] 图6是示出根据一个实施方案的计算环境中的硬件实体和软件实体的示例的框图。

[0015] 图7是示出根据一个实施方案的表示图6的计算环境的部分的属性图的表示的示例的框图。

[0016] 图8是描绘根据一个实施方案的用于实现属性图的示例过程的流程图。

[0017] 图9是描绘根据一个实施方案的用于属性图创建的示例过程的流程图。

[0018] 图10是描绘根据一个实施方案的用于遍历实体及其关系的示例过程的流程图，以便根据一个实施方案创建属性图。

[0019] 图11是描绘根据一个实施方案的用于分析一个或多个属性图的示例过程的流程图。

[0020] 图12是描绘根据一个实施方案的用于遍历属性图作为分析属性图的一部分的示例过程的流程图。

[0021] 图13是示出根据一个实施方案的属性图的表示的示例的框图，其中实现低级复制。

[0022] 图14是示出根据一个实施方案的属性图的表示的示例的框图，其中实现高级副本。

[0023] 图15是示出根据一个实施方案的属性图的表示的示例的框图，其中实现高级同步物理复制。

[0024] 图16是根据一个实施方案的编排架构的示例的框图。

[0025] 图17是描绘适合于实现根据诸如本文所公开的系统的实施方案的系统的各方面的计算机系统的框图。

[0026] 图18是描绘适合于实现根据诸如本文所公开的系统的实施方案的系统的各方面

的网络架构的框图。

具体实施方式

[0027] 以下内容旨在提供本发明的示例的详细描述,而不应视为限制本发明本身。相反,任何数量的变化都会落入本发明的范围内,本发明的范围在说明书之后的权利要求书中限定。

[0028] 虽然本发明容许各种修改和替代形式,但是在附图和详细描述中作为示例提供了本发明的特定实施方案。应当理解,附图和具体实施方式并非旨在将本发明限制于所公开的特定形式。相反,其目的在于覆盖落入由所附权利要求书限定的本发明的实质和范围内的所有修改形式、等同物和替代物。

[0029] 介绍

[0030] 诸如本文所述的那些的方法和系统通过以能够生成适合于编排系统使用的一个或多个结果的方式支持工作负载的依赖性分析来提供对计算环境中的硬件资源和软件资源的管理。此类方法和系统通过使用其一个或多个属性图(或类似的构造)来便于此类结果的产生。

[0031] 应当理解,识别工作负载的依赖性映射和基础设施接触点(接触点)是构建用于应用程序和其他数据中心部件的灾难恢复和迁移的稳健工作流的基本部分。通常,数据中心基础设施与正在编排的一个或多个工作负载之间存在多个接触点,其中任何(或全部)可能有助于支持给定工作负载所需的依赖性。

[0032] 编排操作(诸如迁移和灾难恢复方案)中面临的挑战包括物理上分离的接触点、此类接触点的隔离以及受此类编排操作影响的接触点的识别。方法和系统诸如本文所述的那些以高效且有效的方式解决了这些挑战,如下文和本文其他地方所述。

[0033] 例如,就基础设施单元之间(诸如数据中心之间)的地理分布的接触点而言。鉴于工作负载基础设施的多样性、不同的数据迁移方法和目标类型(托管数据中心或云),识别依赖性链具有挑战性,该依赖性链可用于将工作负载的迁移以及相关数据编排到另一个数据中心。就复制数据库而言,在依赖性评估时,足够的迁移接触点可以是数据库本身。这避免了编排数据库基础设施(如DR和迁移的文件系统)的需要。就使用低级别(例如,逻辑单元号(LUN)级别复制复制VM而言,在依赖性评估时,从VM到数据存储到LUN,迁移接触点将是所考虑的LUN以及驻存于其上的每个VM。另一种场景是在支持它们的一个或多个系统之外复制VM图像(带外复制)。在这种情况下,在依赖性评估时,迁移接触点是备份策略和所生成的备份图像。同样,这避免了编排VM基础设施诸如数据存储、LUN等的需要。

[0034] 另一个示例是隔离接触点的能力。例如,对于具有多种数据迁移方法的更复杂的恢复配置,应基于每个数据迁移方法隔离依赖性。这可能发生在实现低级别复制和带外复制的情况下(例如,使用带外复制复制VM图像,但VM的存储也在LUN级别复制)。

[0035] 另一个示例是识别受编排操作影响的(潜在地)接触点。例如,关于维护操作,可能需要找到将受此类操作影响的实体。此处的依赖性应该反映那些应该被停止或禁用的实体,以便能够执行所考虑的维护。此类情况可能会发生,例如在需要对存储设备执行维护的情况下,导致使用该设备的工作负载的需要应该被正常停止,并且使用该设备的文件系统将被卸载。如根据本公开内容应当理解的,通过确定它们与所考虑的一个或多个设备的关

系来快速且有效地确定受此类操作影响的实体的能力是有利的。

[0036] 方法和系统诸如本文所述的那些通过使用属性图或类似构造提供了此类优点。在此类构造中,硬件部件和软件部件(本文也称为“实体”)及其关系分别由节点(顶点)和链路(边缘)表示。此类结构的相关特征是通常能够提供计算、网络 and 存储资源的表示,并且能够以可以将此类信息呈现给一个或多个编排系统的方式进行分析,为了便于和支持这样表示的实体的编排。因此,描述了计算环境管理架构的各种创新元件的示例,特别强调以适合于编排(例如,用于灾难恢复的目的)的方式创建和分析计算环境。

[0037] 目前,没有解决方案能够提供解决诸如上述挑战的通用系统,这些挑战扩展到不依赖于手动生成的信息的灵活方法(显然,在具有潜在数百或数千个部件的大型系统中是不可逾越的任务,并且其间的关系数量呈指数级增长)。为了解决快速且有效地确定数据和应用程序迁移/DR操作的一个或多个依赖性映射以及其中涉及的接触点的挑战,并且以通用方式这样做,以下特征将被认为是有利的。

[0038] ○通过属性图表示连接域的实体,以实现所需程度的连接路径的自主计算/分析。这将便于识别依赖性和迁移接触点,而无需明确了解实体类型及其之间的关系。

[0039] ○用于识别技术中的映射和迁移接触点的表示的用例和技术不可知分析因此使得它不太可能需要修改,因为在环境中引入了新的依赖性、用例和技术。

[0040] ○以明确限定的标准格式表示依赖性映射,该格式可由其他系统使用,如用于应用程序的工作流创建系统以及VM DR和迁移自动化。

[0041] 为了解决上述需要和所述挑战,方法和系统诸如本文所述的那些提供了元件诸如下文描述的那些。

[0042] 1. 计算环境中各种实体的表示

[0043] 2. 实体的注释

[0044] 3. 依赖性映射的标识

[0045] 4. 依赖性映射的标准接口

[0046] 在提供此类特征时,根据方法和系统诸如本文所述的计算环境管理架构为实体诸如虚拟机 (VM) 以及各种应用程序(诸如可能在台式计算机平台上执行的应用程序)提供灾难恢复和工作负载管理能力。在这样做时,此类架构为系统提供了用于以各种形式管理用于众多弹性能力的信息技术(IT)应用程序,此类解决方案提供例如灾难恢复和工作负载管理。为此,方法和系统诸如本文所述的那些可以采用构造诸如属性图来表示计算环境中的各种实体,以及它们之间可能存在的关系。此类表示提供了多种益处。

[0047] 编排架构的示例特征

[0048] 根据方法和系统诸如本文所述的那些的计算环境还提供了虚拟机 (VM) 和其他此类实体(硬件部件和/或软件部件)的管理。此类实体与将在物理机(计算系统)上供应(并因此被托管)的VM以及已经由此类节点托管的VM相关联。就这一点而言,VM配置可以包括一个或多个资源要求以及一个或多个附加要求。资源要求指示用于托管给定VM的节点中所需的一个或多个资源。附加要求指示用于托管给定VM的节点中所需的一个或多个操作元件。将该VM配置与各种集群的配置进行比较。集群配置可以指示该集群的一个或多个节点的配置信息。此类比较可以由编排系统实现,以选择包括可以满足资源的节点的集群以及托管给定VM的附加要求。可以基于与该VM相关联的服务级别协议(SLA)来供应或迁移VM(例如,在

发生灾难的情况下)。SLA可以包括各种资源要求,诸如供应该VM的节点的物理要求。这些资源要求可以包括CPU、存储器和其他硬件要求,以及对其他实体的依赖性(可以是硬件或软件或其组合)。因此,将VM供应到(或迁移到)服务器,该服务器具有由与该VM相关联的SLA的资源要求指定的资源。

[0049] 如前所述,虚拟机(VM)是计算实体,诸如软件构造,其提供通常与物理计算设备相关联的功能。此类功能性包括,例如,运行应用程序的能力诸如文字处理、数据库和网络浏览程序。虚拟机可以执行操作系统,诸如**Windows[®]**、Linux、Mac OS **X[®]**等。虚拟机在物理机上执行,称为主机。可以在单个物理机上实现或供应多个虚拟机。给定的虚拟机在逻辑上与在物理机(或其他物理机)上运行的其他虚拟机隔离。

[0050] 当将在多个物理机可用的环境中配置新虚拟机时,选择将用于新虚拟机的物理机是一项重要决策。在将现有虚拟机从一个物理机移动到另一个物理机的情况下,例如,如果供应虚拟机的物理机正在离线以进行维护或者已经失败,则选择将现有虚拟机迁移到哪台物理机是一个同样重要的决定。

[0051] 在某些情况下,此类供应由SLA控制,SLA限定了具体实施的要求(如上所述),并且可以包括VM配置,诸如资源要求和附加要求。可以供应资源要求诸如该VM的节点的物理要求。这些资源要求可以包括CPU、存储器和其他硬件要求。一个或多个资源要求指示用于托管VM的节点中所需的一种或多种资源。例如,这些资源包括用于托管VM的节点的各种硬件要求。因此,可以将VM供应到(或迁移到)节点,该节点具有由与该VM相关联的SLA的资源要求指定的资源。

[0052] 附加要求指示用于托管VM的节点中所需的一个或多个操作元件。例如,这些附加要求可以包括可用性要求、数据保护要求和安全要求等。可用性要求可以限定一个或多个VM的可用性要求。一个或多个数据保护要求可以指示用于托管VM的潜在节点中所需的数据保护元件。一个或多个安全要求可以指示用于托管VM的潜在节点中的安全元件。一个或多个操作元件可以指示用于托管VM的潜在节点中所需的数据保护元件。

[0053] VM配置信息可以包括一个或多个资源要求和一个或多个附加要求。可以在集群的任何地方支持、实现、访问和/或存储此类VM配置信息(例如,此类配置信息可以在集群的节点和/或其存储之间分布)。VM配置信息也可以存储在集群外部,诸如存储在中央存储库中,并且例如可以经由网络访问。在一个实施方案中,VM配置包括在VM的服务级别协议(SLA)中。在一个实施方案中,VM配置与尚未托管的VM相关联。例如,可以根据VM配置接收命令,请求创建新VM(例如,在节点处供应)。在另一个实施方案中,VM配置与已经托管在集群的一个节点上的VM相关联。例如,可以接收命令,请求根据VM配置在新节点处供应VM。

[0054] 在一个实施方案中,集群还可以具有相关联的集群配置。集群配置可以包括集群的配置信息和/或一个或多个节点的配置信息。例如,此类集群信息可以包括给定节点的配置信息。需注意,可以在给定集群中的任何地方支持、实现、访问和/或存储集群配置,例如,包括分布在集群的节点之间,集群存储器中或集群外的存储器中,诸如在中央存储库中,并且经由网络访问。

[0055] 虚拟机(新虚拟机或从一台物理机迁移到另一台物理机的现有虚拟机)和相关联应用程序的性能可能会取决于选择托管虚拟机的物理机而异。也就是说,虚拟机的性能(例如,速度、响应性、稳定性)可以取决于实现虚拟机的物理机而异。在给定物理计算机上运行

虚拟机还会影响物理机和物理机上运行的其他进程的性能。虚拟机可以消耗物理机的资源,导致可用于由物理机执行的其他进程的较少资源,例如,在同一物理机上实现的其他虚拟机。更有效地使用物理机的资源的一种技术是在物理机上实现的虚拟机共享由多个虚拟机使用的存储页面。也就是说,可以对物理机的存储器进行去重。

[0056] 在多个物理机可用于托管虚拟机的计算环境中,选择托管虚拟机的物理机的一种方法是比较可用资源,例如,每个物理机的网络传输带宽、存储器、处理器周期,并选择具有最多可用资源的物理机。从多个可用物理机中选择物理机的另一种方法是循环法,其中可用的物理机轮流托管正在新供应或从另一台物理机迁移的虚拟机。满足协商服务水平协议中规定的条款有时是放置虚拟机的重要考虑因素。

[0057] 除了前述资源考虑之外,一个或多个资源要求信息通常可以限定VM的各种资源要求(以及用于最佳和/或可接受操作的“偏好”)。资源要求可包括CPU、存储器、网络、平台、启动盘图像等。这些通常是用于托管VM的节点(例如,服务器)的硬件要求。一个或多个附加要求可以包括可用性要求、数据保护要求和安全要求,以及对其他部件(例如,硬件部件和/或软件部件)的依赖性。

[0058] 可用性要求可以限定一个或多个VM的可用性要求,例如,虚拟机需要能够容忍的物理主机故障的数量,为虚拟机的灾难恢复配置的远程站点的数量等。例如,要求可能包括VM需要容忍N个物理主机故障。具有至少N+1个能够满足一个或多个VM资源要求的物理主机的集群可以满足该要求。在另一个示例中,要求可以包括容忍一个站点故障的VM,即,它需要具有灾难恢复设置。只有配置了灾难恢复(DR)的数据中心中的集群才能满足该要求,诸如,存在集群可用于激活集群上活动的虚拟机(例如,在远程站点)。例如,纽约的集群可能在伦敦有对应的DR集群设置。如果纽约集群中的主机发生故障,则可以通过伦敦的DR集群使在纽约集群上处于活动状态的虚拟机可用。

[0059] 安全要求限定虚拟机的安全区域,以限制虚拟机将访问的网络、恶意软件扫描的策略和频率等。例如,要求可能是每N小时扫描一次VM磁盘图像以查找病毒。可以通过在VM和/或托管VM的节点内设置病毒扫描软件来满足该要求。另选地,可以从可访问VM图像的任何地方(例如,放置VM图像的位置)或可访问虚拟机的磁盘图像的另一主机离线完成该要求。又如,要求可以是使虚拟基础设施符合某些准则。通过检查管理程序、网络、存储是否符合合规性评估系统,可以满足该要求。另一个要求示例是需要主机上供应VM,该主机配置为扫描所有出站流量的DLP网关。具有DLP网关的主机可以满足该SLA,该网关可以配置为扫描VM的所有出站流量。

[0060] 数据保护要求限定备份的频率、介质和方法/类型或虚拟机的快照。例如,要求可能是需要使用特定类型的备份软件每H小时备份一次图像。另一个要求可能需要使用某种类型的备份,诸如增量备份或完全备份。具有适当备份软件设置并可访问备份媒体的主机可满足该要求。

[0061] 在这方面,如根据本公开应当理解的,可以备份虚拟机和与其相关联的虚拟机数据(通常被组织为虚拟磁盘)以为虚拟机和虚拟磁盘提供数据保护。可以恢复备份的虚拟机和虚拟磁盘以还原或开始虚拟机操作。例如,在灾难恢复场景中,由于托管虚拟机的节点崩溃而无法操作的虚拟机可以从备份恢复到另一个主机,并且可以还原操作。在另一个实施方案中,可以从备份恢复虚拟机和虚拟磁盘到主机,并且可以开始操作,从而创建同一虚拟

机的多个实例。由于虚拟机本身包括相对少量的数据(例如,大约千字节或兆字节),因此通常可以相对快速地执行恢复虚拟机。恢复虚拟磁盘通常需要更长的时间,因为虚拟磁盘通常包括相对大量的数据(例如,大约为千兆字节)。

[0062] 示例编排架构

[0063] 图1是计算环境100的框图。如图所示,计算环境100包括若干物理机110(1)-110(N)(计算系统)。每个物理机110托管若干虚拟机115(1)-115(i)。需注意,另选实施方案可以实现不同数量的物理机,每个物理机实现不同数量的虚拟机,并且相同系统可以包括一个或多个物理机,每个物理机实现与相同环境内的其他物理机不同数量的虚拟机。

[0064] 需注意变量标识符,诸如上文使用的变量标识符(“N”和“i”)用于更简单地指定一系列相关或类似元件(例如,物理机)的最终元件(例如,物理机110(N))。重复使用此类变量标识符并不意味着暗示此类元件序列的大小之间的相关性,尽管可能存在此类相关性。使用此类变量标识符不要求每个元件系列具有与由相同变量标识符界定的另一系列相同数量的元件。相反,在每个使用实例中,所识别的变量可以保持与相同变量标识符的其他实例相同或不同的值。

[0065] 物理机110通过网络130耦接到管理服务器120和多个存储设备150(1)-150(4)。网络130可以包括WAN(广域网),诸如因特网、一个或多个LAN(局域网)和/或一个或多个SAN(存储区域网络)以及其他此类构造和系统。

[0066] 物理机110是计算设备或计算设备的部分,可以是任何类型的计算设备,诸如个人计算机、膝上型计算机、服务器、个人数字助理、蜂窝电话等。在一些实施方案中,此类物理机包括在云计算环境中,其中物理计算资源(诸如存储器和处理器周期)根据需求和/或按照合同分配。物理机110的资源由在物理机上实现的任何虚拟机共享。物理机110各自包括存储器112。在一个实施方案中,存储器112是随机存取存储器(RAM)。存储器112被组织成多个页面。页面通常是存储器的固定大小部分,例如4千字节(Kb)的存储器部分。

[0067] 虚拟机115可用于向例如与组织相关联的用户提供计算资源。当用户例如,组织中的信息技术(IT)专业人员请求附加资源时,云供应商处的管理员(例如,IT专业人员)可以供应一个或多个附加虚拟机,以满足对附加资源的请求。在实施方案中,可以采用供应模块121来检测用户在计算资源请求中指定的要求。供应模块121使用供应数据122来执行其功能性,可以响应于该请求自动创建或定位虚拟机,并使虚拟机在物理机110中的一个上供应。可以看出,供应模块121(及其相关联的供应数据122)以及属性图创建模块125和属性图分析模块126(以及相关联的属性图数据127)是由管理服务器120支持的模块之一(作为示例,其在图1中被描绘为管理模块129)。随后描述了系统的操作诸如属性图创建模块125和属性图分析模块126,以及与图3至图16结合并且与它们相关。

[0068] 管理员还可以将一个或多个虚拟机从第一物理机移动到另选物理机。例如,响应于检测到物理机发生故障、已经发生故障、过载,或者在与当前托管一个或多个虚拟机的物理机不同的物理机上可用的附加资源,管理员可以将一个或多个虚拟机移动(迁移)到不同的(目标)物理机。在一个实施方案中,供应模块121可以自动检测指示应该迁移虚拟机的条件,诸如检测到物理机发生故障、已经发生故障、过载,或者在与当前托管一个或多个虚拟机的物理机不同的物理机上可用的附加资源。供应模块121可以响应于检测到此类条件而自动迁移一个或多个虚拟机,作为提供编排服务的一部分。

[0069] 管理服务器120可以是例如计算设备或计算设备的一部分,诸如个人计算机、膝上型计算机、服务器、个人数字助理、蜂窝电话等。管理服务器120可以选择目标物理机来托管新的或迁移的虚拟机。管理服务器120包括供应模块121和供应数据122。

[0070] 供应模块121可以基于新虚拟机(或从一个物理机迁移到另一个物理机的虚拟机)将更有效地利用目标物理机的资源(例如,存储器)的可能性来选择目标物理机(例如,借助于与目标物理机托管或将由其托管的一个或多个虚拟机共享更多数量的存储器页面)。此外,可以使用呈现用于管理服务器120的管理的图形用户界面(GUI)以直观、有效的方式执行此类操作。

[0071] 供应模块121可以使用供应数据122收集和维护关于供应的信息,并确定供应一个或多个虚拟机、其相关联的存储部件(虚拟机磁盘(VMDK))、相关联功能性等以适当(或简单的可接受)使用可用资源的方式。这可以是,例如,接受给定VM的确定(例如,指示哪个物理机可能与给定虚拟机共享最大数量的共享存储器页面并将该信息存储在供应数据122中)。供应模块121可以分析供应数据122以选择哪个物理机将托管虚拟机。供应模块121可以从各种源处收集各种形式的供应数据122。供应模块121还可以跟踪每个物理机的存储器中维护的数据。在实施方案中,每个物理机周期性地关于存储在物理机的存储器中的页面的信息传输到供应模块121。另选地,所考虑的物理机可以响应于向存储器添加新页面、响应于在物理机上提供的新虚拟机、并且/或者响应于供应模块121对条目列表的请求而传输该列表。在实施方案中,每个条目包括地址,例如页面的地址。在一个实施方案中,此类信息可以包括标识物理机托管的一个或多个虚拟机的特征的标签,诸如由物理机托管的虚拟机使用的操作系统或应用程序。

[0072] 在虚拟机在物理机上运行并且正在迁移到另一物理机的实施方案中,供应模块121可以使用来自正在运行虚拟机的物理机的资源信息(例如,存储器信息)来确定虚拟机应该迁移到多个可用物理机(目标物理机)中的哪一个。使用本示例,供应模块121可以将关于虚拟机的信息(如从托管虚拟机的物理机接收的)与每个目标物理机的信息进行比较,以便识别可能迁移给定虚拟机的另一个候选。

[0073] 当虚拟机尚未在物理机上运行时,诸如在供应新虚拟机时,管理服务器120可以执行供应虚拟机的操作,确定供应的虚拟机所需的计算资源。在实施方案中,供应模块121从存储设备(诸如存储设备150中的一个)处接收虚拟机信息,代替和/或除了从物理机接收之外。存储设备150提供持久数据存储,使得即使在存储设备断电之后,存储在此类存储设备上的数据也将保持存储。此类存储设备可以是例如硬盘、光盘(CD)、数字通用光盘(DVD)或其他大容量存储设备,或者包括此类存储设备阵列的存储系统(例如,独立磁盘冗余阵列(RAID)系统或光存储自动点唱机)。此类存储设备也可以是在此类物理存储设备和/或存储系统上实现的虚拟或逻辑存储设备。例如,此类存储设备可以是在RAID存储系统上实现的逻辑卷。另外,此类存储设备可以包括一个或多个存储设备。存储设备还可以包括一种或多种类型的存储介质,包括固态介质(例如,闪存驱动器)、光学介质(例如,CD和DVD)以及磁介质(例如,硬盘或磁带)。在一些实施方案中,可以使用云存储来实现此类存储设备,其中存储设备是逻辑存储设备,一个或多个物理存储设备根据需要和/或按照合同分配到该逻辑存储设备。

[0074] 许多虚拟机(例如,VM 156(1)-(3)、VM 166(1)-(3)和VM 176(1)-(3))可以存储在

一个或多个存储设备150(例如,存储设备150(1)-(3))上,一个或多个VMDK(例如,存储在存储设备150(4)上的VMDK 180(1)-(N))也可以存储。供应模块121可以获得关于这些虚拟机中的一个或多个的信息并供应所考虑的一个或多个虚拟机。供应模块121还可以获得关于一个或多个虚拟机115的信息,以便存储所考虑的一个或多个虚拟机。如图1所示,存储设备150(1)实现文件系统152,其包括去重模块154。文件系统152支持存储在其中的数据去重,包括虚拟机156。在该示例中,虚拟机156(1)-156(3)被去重。也就是说,每个虚拟机156由一组数据形成。在某些实施方案中,虚拟机156包括与虚拟机156中的另一个相同的数据。这可能是在两个或更多个虚拟机156包括相同的应用程序和/或操作系统的情况下。当多个虚拟机156包括相同数据时,存储设备150(1)仅存储相同数据的单个副本。因此,去重数据158包括识别构成虚拟机156的数据的哪些部分在虚拟机之间共享的信息,例如,在两个或更多个虚拟机156之间相同。供应模块121可以使用去重数据158来选择托管虚拟机156中的一个托管虚拟机的物理机。

[0075] 此类存储设备还可以包括一个或多个指纹模块,其可以计算存储的虚拟机的指纹并将如此生成的指纹存储在一个或多个存储设备150上的指纹数据中。计算虚拟机的指纹涉及将虚拟机图像和与虚拟机相关的任何数据划分为块,并计算每个块的指纹。块是数据的固定长度或可变长度部分。指纹模块162可以利用各种函数来为每个块生成签名或指纹。此类函数可以包括例如安全散列算法(SHA)、消息摘要算法5(MD5)、Rabin散列、循环冗余校验(CRC)等中的一者或多者。例如,签名可以是由散列函数(诸如SHA-1)生成的散列,散列函数处理特定块并且作为响应计算散列(例如,SIS级签名)。

[0076] 在此类场景下,供应模块121从存储设备150处接收关于在所得的指纹数据中存储何种指纹的信息。存储设备150中的一个或多个可以响应于新指纹的添加(例如,如果新虚拟机存储在存储设备150之一中),并且/或者响应于供应模块121的请求,周期性地将关于指纹的信息传输到供应模块121。此类请求可以指定应该将指纹数据传输到供应模块121的一个或多个虚拟机。

[0077] 在环境诸如计算环境100中,方法和系统诸如本文所述的那些包括,如所指出的,属性图创建模块125和属性图分析模块126、以及创建/分析的属性图数据(例如,属性图数据127),以支持由构成管理模块120的一个或多个管理模块提供的编排功能。如根据本公开应当理解的,属性图创建模块125提供便于创建属性图数据诸如属性图数据127的功能性,而属性图分析模块126提供便于分析此类属性图数据的功能性,以及生成易于由一个或多个编排系统使用的形式的信息。例如,结合图8、图9和图10(关于属性图创建模块125)和图8、图11和图12(关于属性图分析模块126和利用其输出的编排系统)给出此类功能性的示例。

[0078] 图2是物理机110的框图。物理机110包括存储器200和处理器202。物理机110还包括若干虚拟机115(1)-115(3)。虽然示出了三个虚拟机,但是物理机110可以支持更多或更少的虚拟机。

[0079] 存储器200包括操作系统205。操作系统205控制物理机110的硬件,并向在客户端物理机110上执行的应用程序和虚拟机提供各种服务。例如,当虚拟机115请求将页面带入存储器200时(经由虚拟化模块诸如虚拟化模块225(例如,管理程序等)),操作系统205可以将请求中继到文件系统210,该文件系统可以请求例如从存储设备150中的一个存储设备中

的文件数据读取页面。文件系统210可以接收页面并将页面写入存储器200,例如写入虚拟机数据240。文件系统210可以将数据诸如存储设备150中的一个存储设备中的文件数据组织成文件和目录。文件系统210可以维护一个或多个存储设备150中的文件和数据的物理位置的列表。

[0080] 存储器200包括去重模块230。去重模块230可以执行与存储器200相关的去重操作,包括虚拟机数据240。当要将指定的数据单元(例如,页面)加载到存储器200中时,去重模块230确定包含相同数据的数据单元是否已经存在于存储器200中,例如,该数据可能已由托管在物理机器110上的另一个虚拟机请求,并允许对其进行访问。

[0081] 在实施方案中,文件系统210向去重模块230通知页面请求。为了确定页面是否已经存在于存储器中,去重模块230计算页面的签名(例如,指纹)并将指纹与存储在虚拟机数据240中的指纹列表进行比较。如果指纹存在于列表中,则页面是冗余的,并且去重模块230可以通知文件系统210取消加载。如果未找到指纹,则页面尚未存储在虚拟机数据240中,并且去重模块230将指纹添加到指纹列表并允许加载继续前进。虚拟机数据240包括虚拟机115使用的数据,以及数据的指纹列表。

[0082] 存储器200包括虚拟化模块225。虚拟化模块225执行与虚拟机的创建和管理以及虚拟机的资源相关的操作。虚拟化模块225被示为独立模块,但也可以包括在操作系统205中。虚拟化模块225为物理机110提供同时支持一个或多个虚拟机(诸如虚拟机115)的能力。虚拟化模块225通过协调计算机资源(例如,处理能力、存储器等)到虚拟机的分配来提供这种能力,使得虚拟机与物理计算机类似地操作。虚拟化模块225可以被设计为通过协调处理器资源来支持虚拟机以支持代表虚拟机执行指令,并执行存储器管理以帮助确保虚拟机有效地共享主计算机的系统存储器。此外,虚拟化模块225的某些实施方案不需要主机操作系统(通常称为“裸机管理程序”等)。

[0083] 存储器200包括集群模块235。在实施方案中,可能希望一次为多个虚拟机做出放置或迁移决定。在此类实施方案中,放置服务器可以形成类似位置的虚拟机的集群。集群的大小可以取决于各种因素,诸如每个物理机上的可用资源、虚拟机的资源要求等。各种算法可用于聚类。

[0084] 物理机110还包括虚拟机115(1)-115(3)。一般来讲,虚拟机115是充当物理计算机系统的软件构造。例如,虚拟机115可以包括并执行应用程序、提供服务和处理命令。虚拟机具有其自己的操作系统,诸如Microsoft Windows[®]、Unix等,并且可以与主机上的虚拟化软件接口,诸如在物理机110上提供的虚拟化模块225。

[0085] 架构特征及其表征结构的示例

[0086] 图3是示出根据一个实施方案的计算环境分析架构的示例的框图。如上所述,方法和系统诸如本文所述的那些提供了元件诸如下文描述的那些。

[0087] 1. 计算环境中各种实体的表示

[0088] 2. 实体的注释

[0089] 3. 依赖性映射的标识

[0090] 4. 依赖性映射的标准接口

[0091] 关于实体和关系表示,方法和系统诸如本文所述的那些表示由一个或多个抽取/转换/加载(ETL)进程馈送到表示构造诸如属性图中的实体记录。这导致实体的图形抽取及

其彼此的关系,其中实体被表示为此类属性图的顶点,并且它们的关系被表示为此类属性图的边缘。

[0092] 此类方法和系统以及表示构造提供了实体的注释。可以对可编排的实体(例如,支持操作界面的实体,诸如,例如上-下、启用-禁用等)进行注释。此类实体的示例是应用程序、虚拟机、文件系统、数据存储、阵列一致性组等。

[0093] 此类注释可包括,例如:

[0094] `y_ecosystem(_e)`:通过可能的操作对实体进行分组。示例包括计算、存储、网络、复制、云等。

[0095] `y_technology(_t)`:启用生态系统的技术名称。示例包括VM实体、复制实体、云实体等。

[0096] 在某些系统操作期间,某些实体可能不适合进行编排、某些编排操作或编排,或者可能永远不适合编排(例如,物理机无法迁移,除非通过物理地移动该单元)。因此,此类实体可能不需要(或甚至可能不能)编排(例如,物理服务器、机柜、管理程序等)。

[0097] 因此,配置用于数据传输(复制/副本)的实体可以用属性注释。

[0098] `y_replicated`:例如,复制实体的注释是`_replicated`。此类实体的示例是复制的LUN,是使用带外复制来复制的备份图像,复制的数据存储等。

[0099] `y_copy`:快照和备份副本另外使用`_copy`进行注释。

[0100] 方法和系统诸如本文所述的那些还支持依赖性映射的标识。所涉及的依赖性映射的标识是识别给定工作负载的实体和关系的最佳有序集合的迭代过程。

[0101] `y`系统从源数据中心上的选定工作负载开始遍历实体分级结构。

[0102] `y`它记录了使用`_technology`注释的实体。

[0103] `y`用`_replicated`注释的实体被记录为源数据中心的迁移接触点。

[0104] `y`对于此类实体,系统通过询问关系并选择也使用`_replicated`注释的路径并在目标数据中心上将其记录为迁移接触点来查找目标数据中心上的相关实体。

[0105] `y`取决于复制/副本的配置方式,可以存在多于一个的远程目标。

[0106] `y`系统遍历在目标数据中心上开始迁移接触点的实体分级结构。

[0107] `y`它记录了使用`_technology`注释的实体。

[0108] 方法和系统诸如本文所述的那些也提供标准表示。此过程在源和目标数据中心上提供了有序的实体列表。然后,系统合并生态系统和技術组中的实体,以提供可以编排的各种层的视图。

[0109] 简单地说,当使用属性图作为表示构造时,此类属性图,其中顶点和边缘都具有属性(键值对,其中键是属性名称),可用于维护有关实体及其关系的信息,如上所述。此处的优点是因为此类表示构造也具有边缘属性,遍历也可以考虑路径/关系的特征,遍历者可以将边缘/关系上的属性标记为永久或临时标记,可以在分析期间随后使用。此种表示构造可包含的信息类型(以及特征)的示例如下所示:

[0110] `Vertex1`:

```
    {
        _type: "host"
        id: "12345231"
[0111]    _technology: "VMware"
        _environment: "virtualization"
        platform: "ESXi"
        ncpu: 16
    }
[0112] Vertex2:
    {
        _type: "disk"
        id: "98345231"
[0113]    _environment: "virtualization"
        size: "16GB"
        type: "SSD"
    }
[0114] Vertex3:
    {
        _type: "disk"
        id: "98345232"
[0115]    _environment: "virtualization"
        size: "16GB"
        type: "SSD"
    }
[0116] Edge between vertex1 and vertex2:
    {
[0117]    label: "has"
        mode: "READ-WRITE"
    }
[0118] Edge between vertex1 and vertex3:
```

```
{  
    label: "has"  
[0119]    mode: "READ-ONLY"  
}
```

[0120] 从上面可以看出,某些特征(_type、_environment等)可用于确定给定实体的适当管理,并确定它们之间的依赖性关系。

[0121] 鉴于前述内容,图3描绘了计算环境分析架构300。计算环境分析架构300包括多个软件和/或硬件层,其可以实现为例如单独的模块,如根据本公开应当理解的。此外,图3中描绘的一个或多个层可以与其中描绘的其他层组合、移除、通过其他层增强等。如图3所示,计算环境分析架构300的层包括迁移层310、工作流创建层320和实体表示层(编排)330、接触点识别层340、依赖性识别层350、实体表示层360、实体注释层370和ETL(抽取/转换/加载)处理层380。作为在给定计算环境的一个或多个实体上执行编排的先决条件,构造诸如本文所述的属性图(或类似的此类构造)表示给定计算环境的各种虚拟和物理硬件和软件单元在此类构造中表示为实体及其关系。例如,这些实体和关系表示计算环境诸如数据中心内的各种硬件和软件单元。此类构造可以用给定实体的特征来注释,然后此类信息可以由编排系统用来编排这些实体。此类实体的注释可以包括识别实体的生态系统、技术、是否复制实体、是否有实体的副本以及其他此类特征的信息。通过使用此类构造,方法和系统诸如本文所述的那些提供了基于给定工作负载的接触点来识别依赖性映射和受影响实体的迁移。在这种情况下,可以在源计算环境(例如,数据中心)内遍历此类构造,从而允许记录关于此类实体及其关系的信息。例如,标记有给定标记(例如,技术)的实体允许分析过程(例如,属性图分析过程诸如本文所述的那个)记录关于实体的信息,这些实体随后可以作为给定编排系统操作的一部分进行迁移。此类遍历还可以基于被标记为被复制或副本的实体切换到目标计算环境(例如,数据中心)。在这样做时,此类遍历可以遍历数据中心之间的边界。

[0122] 因此,在此类实例中,属性图可以遍历到目标数据中心,从而允许记录目标数据中心中也被标记为与给定技术相关联的实体(例如,可以迁移的虚拟设备)。此外,可以定制遍历路径(例如,通过呈现给管理员的用户界面)以考虑外部可插拔规则,以及用于分析系统的其他手动输入的指导。通过使用工作流友好表示诸如属性图,分析系统诸如本文所述的那个能够在例如源数据中心和目标数据中心上提供有序的实体列表。此外,此类表示将实体合并到生态系统和技术组中,以提供可由给定编排系统编排的各种层的视图。

[0123] 图4是示出根据一个实施方案的计算环境的示例的框图。因此,图4描绘了计算环境400,其继而又包括由边界415分开的源系统405和目标系统410。边界诸如边界415的示例可以是逻辑的(例如,不同的虚拟网络、不同的网络域等)或物理的(例如,物理上由距离分开、物理上不同的网络等)。在这种情况下,分析计算环境诸如计算环境400中的依赖性所面临的挑战之一包括需要确定跨越边界诸如边界415的依赖性。

[0124] 如图4所示,计算环境400包括被描绘为工作负载420的实体。工作负载420与实体422和424以及实体430具有关系。源系统405还包括实体432、434、440、442和444。如图所示,实体430与实体432、434和440具有关系。实体440继而又与实体442和444具有关系。实体440

还与目标系统410中的实体具有关系(在图4中描绘为实体450)。实体450不仅与目标系统410的实体452和454具有关系,而且还与实体460有关,实体460继而又与图4中描绘的实体作为资源470相关。图4中描绘的计算环境400的表示允许遍历其中描绘的实体和关系。确定最佳依赖性映射和基础设施接触点对于构建支持编排操作(诸如灾难恢复和应用程序和数据中心迁移)的工作流非常重要。此类依赖性在图4中由用于某些实体和关系的较粗的线宽示出。例如,工作负载420取决于实体430,该实体继而又取决于实体440。源系统405中的实体440取决于目标系统410中的实体450。目标系统410中的资源470、实体460与实体450之间也存在依赖性关系。因此,通过遍历表示这些实体及其关系的构造,根据本文所述的方法和系统的依赖性分析系统能够产生所考虑的实体的有序表示,并从而支持对由此表示的硬件和软件单元的编排操作。

[0125] 图5是示出根据一个实施方案的具有多个接触点的计算环境的示例的框图。工作负载520与实体522和524以及实体530具有关系。源系统505还包括实体532、534、540、542和544。如图所示,实体530与实体532、534和540具有关系。实体540继而又与实体542和544具有关系。实体540还与目标系统510中的实体具有关系(在图5中描绘为实体550)。实体550不仅与目标系统510的实体552和554具有关系,而且还与实体560有关,实体560继而又与图5中描绘的实体作为资源570相关。

[0126] 图5中描绘的计算环境500的表示允许以前面提到的方式遍历实体和其中描绘的关系。确定最佳依赖性映射和基础设施接触点对于构建支持编排操作(诸如灾难恢复和应用程序和数据中心迁移)的工作流非常重要。此类依赖性在图5中由用于某些实体和关系的较粗的线宽示出。例如,工作负载520取决于实体530,该实体继而又取决于实体540。源系统505中的实体540取决于目标系统510中的实体550。目标系统510中的资源570、实体560与实体550之间也存在依赖性关系。因此,通过遍历表示这些实体及其关系的构造,根据本文所述的方法和系统的依赖性分析系统能够产生所考虑的实体的有序表示,从而支持对由此表示的硬件和软件单元的编排操作。图5中还描绘了实体524与资源580之间的关系。如根据本公开应当理解的,实体524与资源580之间的关系跨越边界515。该信息以实体540与实体550之间的关系的方式与实体524与资源580之间的边缘相关联。

[0127] 图6是示出根据一个实施方案的计算环境中的硬件实体和软件实体的示例的框图。如图所示,图6的计算环境包括由边界606分开的数据中心602和数据中心604。如前所述,边界诸如边界606可以由数据中心602与604之间的逻辑分离、物理分离和其他此类情况产生。应当理解,作为灾难恢复计划等的结果,由边界606表示的分离通常是有意的。数据中心602包括例如物理机610和存储系统612。类似地,数据中心604被描绘为包括物理机616和存储系统618。数据中心602和604通过广域网620彼此通信地耦接。数据中心602内的实体能够经由企业网络622彼此通信,该企业网络继而又耦接到广域网620。以类似的方式,数据中心604包括企业网络624,该企业网络将数据中心604的实体(例如,物理机616和存储系统618)彼此耦接。数据中心604还通过企业网络624通信地耦接到广域网620。

[0128] 图6中描绘的物理机(物理机610和616)支持各种硬件和软件单元。例如,物理机610包括处理器630、网络接口632和存储器634。如图6所示,各种软件模块存储在存储器634中(并由其执行)。此类软件模块可以包括例如操作系统650、虚拟化模块652和一个或多个虚拟机(在图6中示为虚拟机(VM)654(1) - (N))。虚拟化模块652可以实现为例如管理程序或

类似构造,以支持虚拟机654(1)-(N),继而虚拟机654(1)-(N)支持一个或多个应用程序的执行(在图6中示为应用程序655(1)-(6))。存储系统612包括虚拟机磁盘(VMDK)658(1)-(N),其为一个或多个相关联的虚拟机(例如,虚拟机654(1)-(N))提供存储。

[0129] 以类似的方式,物理机616提供部件诸如处理器660、网络接口662和存储器664。以存储器634的方式,存储器664存储并允许执行软件模块诸如操作系统670、虚拟化模块672和一些数量的虚拟机(在图6中示为虚拟机674(1)-(N))。继而,虚拟机674(1)-(N)可以支持一些数量的应用程序(在图6中示为应用程序675(1)-(4))。存储系统618包括虚拟机磁盘(VMDK)678(1)-(N),其为一个或多个相关联的虚拟机(例如,虚拟机654(1)-(N))提供存储。

[0130] 图7是示出根据一个实施方案的表示图6的计算环境的部分的属性图的表示的示例的框图。在图7中,计算环境诸如图6的计算环境600由属性图700(或类似构造)表示。如前所述,数据中心602和数据中心604由边界606分开。在属性图700中,如图7所示,数据中心602和数据中心604中的实体经由存储信息组720与存储信息组725之间的复制关系710彼此具有关系。可以看出,这些和其他实体虽然未在图6中示出,但是在物理上或逻辑上存在,因此在属性图700中表示。以所指出的方式,属性图700还表示出现在图6中的实体之间的关系。例如,应用程序655(1)取决于应用程序655(2)。应用程序655(1)和655(2)继而又取决于虚拟机654(1)。虚拟机654(1)取决于虚拟化模块652并且还取决于虚拟机磁盘中的一个(虚拟机磁盘658(1))。应用程序655(1)和655(2)也取决于存储在存储单元735上的数据集730,该存储单元由逻辑存储标识符740标识,该逻辑存储标识符均存在于存储系统612中。存储系统612的存储单元(例如,由逻辑存储标识符740标识的存储单元735)被存储组信息720识别为存储组的成员,这导致经由其中的存储组信息725与数据中心604中的存储复制关系(复制关系710)。以类似于数据中心602的方式,数据中心604通过在存储系统618内包括例如存储在由逻辑存储标识符760标识的存储单元755中的数据集750来提供对复制关系710的支持。

[0131] 表示构造的创建和分析的示例过程

[0132] 图8是描绘根据一个实施方案的用于实现属性图的示例过程的流程图。图8描绘了属性图实现过程800。属性图实现过程800开始于在810处创建属性图。一旦创建了必需的一个或多个属性图,就可以执行由此表示的硬件和软件部件的分析。因此,在步骤820处执行属性图分析。已经使用方法和系统诸如其中描述的那些分析了所考虑的计算环境的硬件和软件部件,现在可以执行那些硬件和软件部件的编排。然后在步骤830处完成基于如此执行的属性图分析的一个或多个结果的编排。

[0133] 图9是描绘根据一个实施方案的用于属性图创建的示例过程的流程图。可以通过属性图创建过程900来完成属性图(或一个或多个属性图)的创建,如图9所示。属性图创建过程900开始于将实体(例如,数据中心内的硬件和/或软件部件)识别为用于创建属性图(或其一部分)的起点并且在910处选择它。对于所选实体,确定所选实体是否已经收集了其信息(920)。如果尚未收集所选实体的信息,则收集所选实体的信息,并将其与表示所选实体的属性图中的节点相关联地存储(930)。一旦收集并存储了必要的信息(或确定已经执行了此类收集和存储,因此没有必要),就确定所选实体是否是需要在属性图中创建节点的最后一个实体(940)。如果所选节点确实是最后一个实体,则该过程结束。

[0134] 另选地,如果剩余附加实体,则管理软件遍历所选实体可能与其他实体的关系,以便将这些实体添加到属性图(950)。结合下文图10中描绘的过程描述了用于遍历此类关系的过程的示例。管理系统已经从所选实体遍历到下一个实体,下一个实体被选择作为下一个选择的实体用于分析(960)。然后,该过程返回到关于是否已经为(现在)所选实体收集实体信息的确定(920)。该过程继续,直到所考虑的实体的实体信息已被收集并与属性图中的那些实体相关联地存储。

[0135] 图10是描绘根据一个实施方案的用于遍历实体及其关系的示例过程的流程图,以便根据一个实施方案创建属性图。如上所述,管理软件将遍历硬件和软件部件及其关系,以便构建表示此类部件及其中之关系的构造(例如,属性图)。为此,管理过程将通过它们的关系遍历此类部件,如图10中由属性图创建遍历过程1000所描绘的。属性图创建遍历过程1000开始于识别当前实体与其他实体的一个或多个关系(即,所考虑的部件与给定数据中心或其他数据中心内的其他部件有关)(1010)。接下来,选择这些实体关系中的一者(1020)。然后,收集关于所选实体关系的关系信息(即,所讨论的部件之间的关系),并将其与表示该关系的属性图的边缘相关联地存储(1030)。然后确定当前实体(部件)与其他此类实体(部件)之间的其他实体关系是否仍然要添加到属性图(1040)。如果不存在进一步的关系,并且因此需要创建,则该过程前进到与当前实体具有关系的下一个实体将被选择的点(1050)。然后该过程结束。另选地,在仍然要将附加实体关系添加到属性图的情况下(1040),属性图创建遍历过程1000进行到图9中描绘的属性图创建过程的执行(属性图创建过程900),以允许创建表示一种或多种剩余实体关系(S)的必需链路(SM),并指示起始实体是当前选择的实体(1060)。在这样做时,并且根据本公开应当理解,在属性图创建遍历过程1000(以及属性图创建过程900)中创建一个或多个剩余链路的过程本质上是递归的,并且因此根据诸如本文所述的方法和系统,以提供图形创建的方式探索部件及其关系。因此,作为给定部件可以与其他部件具有的多个关系,可以在结果属性图中创建边缘(链路),从而提供部件及其在计算环境中的关系的准确表示。

[0136] 图11是描绘根据一个实施方案的用于分析一个或多个属性图的示例过程的流程图。因此,可以通过例如属性图分析过程1100来执行属性图分析。属性图分析过程1100开始于识别给定属性图中的节点,该节点表示待编排的工作负载(1110)。在这样做时,属性图分析过程1100将该节点识别为依赖性分析的起点。接下来,分析与属性图中的节点相关联的实体信息(1120)。已经确定与所选节点相关联的实体信息是否指示实体(节点)应该是正在生成的依赖性链的一部分(1130)。如果实体信息指示实体不需要成为依赖性链的一部分,则确定所选节点是否是针对给定依赖性链分析的最后一个此类节点(1140)。如果所考虑的节点是最后一个此类节点,则该过程结束。另选地,如果实体信息指示该节点应该包括在依赖性链中,则收集此类实体信息并将其存储在正被记录的依赖性信息中(1150)。如前所述,然后确定所选节点是否是依赖性链中的最后一个节点(1140)。如果所选节点是依赖性链中的最后一个节点,则该过程结束。另选地,如果仍有待分析的其他节点,则该过程前进到遍历一个或多个适当的链路和属性图以到达可能包括在依赖性链中的下一个节点(1160)。下文结合图12详细讨论可以在遍历到下一节点时执行的操作的示例。遍历到待分析的属性图中的下一个节点后,属性图分析过程1100选择所考虑的节点,并进行其分析(1170)。因此,该过程循环到该所选节点的分析(1120),并且属性图分析过程1100的过程继续。

[0137] 图12是描绘根据一个实施方案的用于遍历属性图作为分析属性图的一部分的示例过程的流程图。因此描绘了属性图分析遍历过程1200,并且开始于从属性图中的当前节点到属性图中的其他节点的一个或多个边缘的标识(1210)。此时,属性图分析遍历过程1200从当前节点中选择表示该实体与属性图中表示的另一个实体的关系的边缘(1220)。然后确定与边缘(关系)有关的边缘信息是否指示边缘通向将包括在依赖性链中的一个或多个节点(1230)。如果给定边缘不通向此类节点(因此可以得出结论,不需要通过属性图在该路径上进行进一步分析),则指示沿此类路径的遍历不需要进行(1240)。然后该过程结束。另选地,如果所考虑的边缘信息指示边缘通向应该是依赖性链的一部分的一个或多个节点,则确定是否需要遍历来自属性图中的当前节点的其他边缘(1250)。如果不需要遍历来自当前节点的其他边缘,则确定边缘是否表示接触的点(接触点),并且如果是,则将接触点信息记录在依赖性数据中(1260)。然后进行指示,遍历可以进行到如此识别的下一个节点(1270)。然后该过程结束。

[0138] 另选地,如果仍然要遍历属性图中的当前节点的其他边缘(1250),则属性图分析遍历过程1200对剩余边缘执行属性图分析(属性图分析遍历过程1100),指示用于此类分析的起始节点是当前节点(1280)。以类似于前面讨论的方式,应当理解,该过程本质上是递归的,因此,以确保所有必需节点(以及实体)都包含在有序依赖性信息中的方式遍历属性图问题,编排软件可以使用这些信息来执行灾难恢复等管理任务。如前所述,如果当前边缘表示接触点,则关于该接触点的信息被记录在依赖性数据中,供编排软件使用(1260)。同样如前所述,然后进行指示,即所考虑的属性图的遍历可以进行到如此识别的下一个节点(1270)。然后该过程结束。

[0139] 表示构造的示例

[0140] 图13是示出根据一个实施方案的属性图的表示的示例的框图,其中实现低级复制。在图13的示例中,虚拟机受LUN级复制保护(虚拟机受到保护,因为它驻存于存储阵列级复制的LUN上的数据存储上)。因此,在图13中描绘了属性图1300,并且如前所述,包括数据中心1302和数据中心1304中的实体的表示,其由边界1306分开。由图13中描绘的实体(节点)表示的数据中心1302中的部件包括虚拟机1310、虚拟机磁盘1315和数据集1317。虚拟化模块1320支持虚拟机1310、虚拟机磁盘1315和数据集1317。数据集1317存储在机柜1335中的存储单元1330上。存储单元1330由逻辑单元号1340标识。存储在存储单元1330中的信息在数据中心1302中的一致性组对象1350与数据中心1304中的一致性组对象1355之间复制。复制操作由低级复制关系1360表示。执行数据集1317的复制,因此数据集1317中的数据被复制到数据集1316,该数据集存储在机柜1375内的存储单元1370中,并由逻辑单元号1380标识。数据集1360由虚拟化模块1390支持。

[0141] 如图13所示,可以迁移由属性图1300的节点表示的各种实体(例如,在灾难恢复的情况下),并且它是实体的标识和某些其他特征的确定,使编排系统能够执行此类迁移。在这种情况下,图13中描绘的具有较重线宽的元件表示在属性图分析过程诸如图11的属性图分析过程1100期间识别的实体及其关系。例如,VM 1310被视为与VMDK 1315具有关系,VMDK 1315与数据集1317具有关系。存储单元1330和机柜1335(以及存储单元1330的标识符(逻辑单元号1340))不适合用于迁移,并且因此不以较粗的线宽示出。通过低级复制关系1360彼此相关联的一致性组对象1350和1355允许此类属性图分析过程关于数据中心1304中的实

体进行确定。在这种情况下,此类过程将数据集1360识别为(潜在地)受到迁移。在该示例中,存储单元1370(由逻辑单元号1380标识)和机柜1375以及虚拟化模块1390被认为不适合迁移。

[0142] 通过遍历图13中描绘的实体,可以组装用于属性图的信息:

```

DC1 {
    Ecosystems: {
        Compute: [
            {Tech: vmware, Objects: {VM1}}
        ],
        Storage: [
            {Tech: vmware, Objects: {vmdk1, DS1}}
[0143]     ],
        Replication: [
            {Tech: srdf, Objects: {CG1}}
        ]
    }
},
DC2 {
    Ecosystems: {
        Replication: [
            {Tech: srdf, Objects: {CG2}}
        ],
        Storage: [
[0144]     {Tech: vmware, Objects: {DS2}}
        ],
    }
}

```

[0145] 图14是示出根据一个实施方案的属性图的表示的示例的框图,其中实现高级副本。在图14所示的示例中,使用带外复制技术来保护虚拟机。因此,在图14中描绘了属性图1400,并且再次包括公共元件诸如数据中心1402、数据中心1404和它们之间的边界(在图14中描绘为边界1406)。在图14中描绘的场景中,VM 1410、VMDK 1415和数据集1417彼此相关地描绘,如机柜1425中的存储单元1420并且由逻辑单元号1430标识。如前所述,虚拟化模块1440支持存储在存储单元1420内的VMDK 1415中的VM 1410和数据集1417。数据集1417的副

本存在于数据中心1404中,作为存储在机柜1465中的存储单元1460中并由逻辑单元号1467标识的数据集1450,并由虚拟化模块1470支持。数据集1450是数据集1470的副本,并且由复制过程1482与复制过程1484之间的高级复制关系1480创建。复制过程1482在主过程1490的控制下,而复制过程1484在主过程1492的控制下。如在图13中呈现的示例中,图14中描绘的示例采用粗线宽来表示属性图1400中的节点,其在给定示例中受到编排。在这种情况下,数据中心1402中的VM 1410、VMDK 1415和数据集1417受到此类编排。然而,通过遍历属性图1400,可以确定数据集1450也受到此类编排。如根据本公开应当理解的,迁移可以包括所有这些实体或其某些部分。例如,在数据中心1402中的实体主题编排可以迁移到另一个数据中心,同时在例如灾难恢复场景中保持与数据中心1404(例如,数据集1450)内的实体的关系。

[0146] 通过遍历图14中描绘的实体,可以组装用于属性图的信息:

```
DC1 {
[0147]   Ecosystems: {
           Compute: [
               {Tech: vmware, Objects: {VM1}}
           ],
           Copy: [
               {Tech: nbu, Objects: {vmlcopy}}
           ]
       }
[0148] },
DC2 {
       Ecosystems: {
           copy [
               {Tech: nbu, Objects: {vmlcopy}}
           ]
       }
}
```

[0149] 图15是示出根据一个实施方案的属性图的表示的示例的框图,其中实现高级同步物理复制。因此,图15描绘了属性图1500,其包括由边界1506分开的数据中心1502和数据中心1504。在图15中呈现的场景中,数据中心1502支持数据库1510,该数据库包括一个或多个文件1515,这些文件系统存储在机柜1530中的存储单元1525上的文件系统1520中,并由逻辑单元号1535标识。主机计算系统1540支持文件系统1520中的数据库1510及其一个或多个文件1515。数据库1510被复制到数据中心1504并作为数据库1560出现在数据中心1504中,数据库1560包括在机柜1580中的存储单元1575上的文件系统1570中的一个或多个文件

1565,并由逻辑单元号1585标识。以类似于数据中心1502的方式,数据中心1504支持一个或多个主机计算系统(在图15中描绘为影响同步关系的主机计算系统1585(在图15中描绘为高级同步关系1590))上的数据库1560、一个或多个文件1565和文件系统1570,数据中心1502和1504中的部件包括同步物理复制模块1592和同步物理复制模块1594。同步物理复制模块1592和1594支持高级同步关系1590,其在属性图1500中表示并且可以通过属性图分析过程诸如图11的属性图分析过程1100来分析。为此,图15中描绘的较粗线宽的实体是受到这种属性图分析的那些实体,并且因此出现在编排信息中,该编排系统可以使用该编排信息来执行例如此类实体的迁移(例如,在灾难恢复场景的情况下)。因此,数据库1510,其一个或多个文件(一个或多个文件1515)和文件系统1520可以迁移到另一个数据中心以便于从灾难中恢复。在这种情况下,可以通过迁移同步物理复制模块1592或在新数据中心中正确配置此类模块来在保持高级同步关系1590的同时执行这些实体的迁移。

[0150] 通过遍历图13中描绘的实体,可以组装用于属性图的信息:

```

DC1 {
    Ecosystems: {
        App: [
            {Tech: oracle, Objects: {Oradb1}}
        ],
        Replication: [
            {Tech: dataguard, Objects: {DGuard2}}
        ]
    }
},
[0151] DC2 {
    Ecosystems: {
        Replication: [
            {Tech: dataguard, Objects: {DGuard2}}
        ],
        App: [
            {Tech: oracle, Objects: {Oradb2}}
        ],
    }
}

```

[0152] 使用表示构造的示例编排架构

[0153] 图16是根据方法和系统诸如本文所述的那些管理和编排架构的框图。如上所述,

ITRP架构诸如图16中所示(并且在其中描绘为ITRP架构1600)可以实现为集中式架构(例如,如图1所示(ITRP服务器120及其模块)),实现为分布式架构(例如,如此类架构诸如本文所述的那些的许多特征所指出的),或者实现为它们的一些组合。在这种情况下,图16中的ITRP架构1600的描绘是其中示出的元件的逻辑表示,它实现可以以适合于当前情况的任何方式分布。

[0154] 从图16中可以看出,ITRP架构1600包括ITRP系统1605,其继而又包括弹性管理器1610和基础设施管理服务1615。根据方法和系统诸如本文所述的那些,基础设施管理服务1615提供分布式管理功能性,在某些实施方案中,该分布式管理功能性还可以包括两层服务。基础设施管理服务1615继而又包括一个或多个层。如图16所示,基础设施管理服务1615包括两个层,示为聚合器层1620和代理层1630。然而,根据本公开,应当理解,基础设施管理服务1615可以配置有多于两层,并且可以实现任何数量的此类层,这可能适合于遇到的情况。

[0155] 聚合器层1620用作“顶”层,并且提供为所支持的硬件和软件元件提供所发现资产的统一概要、统一操作层和持久状态的能力。代理层1630,在其支持下,用作“底”层,并提供发现、监视和操作所支持的硬件元件(示例-存储阵列)和软件元件(示例-数据库(DB)和/或数据库管理系统(DBMS)实例)的能力。

[0156] 如上所述,基础设施管理服务1615还包括弹性管理器1610。弹性管理器1610可以被配置为提供集中式管理功能性,在某些实施方案中,该集中式管理功能包括两层服务。然而,与基础设施管理服务1615一样,弹性管理器1610可以配置有多于两层,并且可以实现任何数量的此类层,这可能适合于遇到的情况。如图16所示,弹性管理器1610包括业务层1640。业务层1640用作“顶”层,并且继而又可以包括多个模块,其示例在图16中被描绘为工作负载管理服务模块1642、恢复自动化服务模块1644和报告服务模块1646。

[0157] 弹性管理器1610还包括核心层1650。支持业务层1640的核心层1650用作“底”层,并且通过包括模块诸如供应模块1660、ER服务模块1661、数据库(DB)服务模块1662、消息服务模块1663、通信和数据外观服务模块1664、认证和授权服务模块1665、日志和审计服务模块1666、许可服务模块1667和工作流服务模块1668来提供多种服务。如根据本公开应当理解的,此类模块(以及它们提供的服务)仅仅是其示例,并且是可以提供以支持业务层1640的许多其他此类功能和服务。

[0158] 计算和网络环境的示例

[0159] 如上所示,本文所述的系统可以使用各种计算机系统和网络来实现。下文是参考图17和图18描述此类计算和网络环境的示例。

[0160] 图17描绘了适用于实现本文所述的系统的各方面的计算机系统1710的框图等。计算机系统1710包括总线1712,该总线将计算机系统1710中的主子系统互连,这些主子系统诸如中央处理器1714、系统存储器1717(通常是RAM,但也可以包括ROM、闪存RAM等)、输入/输出控制器1718、外部音频设备(诸如经由音频输出接口1722的扬声器系统1720)、外部设备(诸如经由显示器适配器1726的显示器屏幕1724)、串行端口1722和1730、键盘1732(与键盘控制器1733对接)、存储接口1734、可操作以接收软盘1738的软盘驱动器1737、可操作以与光纤信道网络1790连接的主机总线适配器(HBA)接口卡1735A、可操作以连接SCSI总线1739的主机总线适配器(HBA)接口卡1735B、以及可操作以接收光盘1742的光盘驱动器

1740。该计算机系统还包括鼠标1746 (或经由串行端口1722耦接到总线1712的其他即指即点(point-and-click)设备)、调制解调器1747 (经由串行端口1730耦接到总线1712) 和网络接口1748 (直接耦接到总线1712)。

[0161] 总线1712允许在中央处理器1714与系统存储器1717之间进行数据通信,该系统存储器可包括只读存储器(ROM)或闪存存储器(均未示出)、以及随机存取存储器(RAM) (未示出),如先前所指出。RAM通常是在其中加载操作系统和应用程序的主存储器。除其他代码外,ROM或闪存存储器可包含基本输入输出系统(BIOS),BIOS控制基本硬件操作,如与外围部件的交互。驻存于计算机系统1710的应用程序通常存储在计算机可读存储介质上并从其访问,诸如硬盘驱动器(例如,固定盘1744)、光驱(例如,光驱1740)、软盘单元1737或其他计算机可读存储介质。

[0162] 存储接口1734 (正如计算机系统1710的其他存储接口)可连接到用于存储和/或检索信息的标准计算机可读介质,如固定磁盘驱动器1744。固定磁盘驱动器1744可以是计算机系统1710的一部分,或者可为独立的并通过其他接口系统进行访问。调制解调器1747可以提供经由电话链路到远程服务器或经由互联网服务提供商(ISP)到互联网的直接连接。网络接口1748可以提供经由直接网络链路到远程服务器或经由POP (入网点)到互联网的直接连接。网络接口1748可使用无线技术来提供这种连接,包括数字蜂窝电话连接、蜂窝数字分组数据(CDPD)连接、数字卫星数据连接等等。

[0163] 许多其他设备或子系统(未示出)可通过类似方式连接(例如,文档扫描器、数字相机等等)。相反,图17中所示的所有设备不需要存在以实践本文所述的系统。这些设备和子系统可通过不同于图17所示的方式来互连。诸如图17所示的计算机系统的操作在本领域中是公知的,并且在本申请中不再详细讨论。用于实现本文所述的系统的模块的代码可以存储在计算机可读存储介质中,诸如存储在系统存储器1717、固定磁盘1744、光盘1742或软盘1738中的一者或多者中。计算机系统1710上提供的操作系统可以是MS-WINDOWS®、UNIX®、Linux®或其他操作系统。

[0164] 此外,关于本文所述的信号,本领域的技术人员将认识到,可以将信号从第一块直接传输到第二块,或者可以在块之间修改(例如,放大、衰减、延迟、锁存、缓冲、反转、滤波或以其他方式修改)信号。尽管上述实施方案的信号的特征在于从一个块传输到下一个块,但是其他实施方案可以包括所修改的信号来代替此类直接传输的信号,只要信号的信息和/或功能方面在块之间传输即可。在某种程度上,由于所涉及的电路的物理限制(例如,将不可避免地存在一些衰减和延迟),在第二块处输入的信号可被概念化为从第一块输出的第一信号导出的第二信号。因此,如本文所用,从第一信号导出的第二信号包括第一信号或对第一信号的任何修改,无论是由于电路限制还是由于通过不改变第一信号的信息和/或最终功能方面的其他电路元件。

[0165] 此外,并且根据本公开应当理解,本文所述的每个操作可以由模块(例如,软件模块)或模块的一部分或计算机系统用户来执行。因此,上述方法、其操作及其模块可以在被配置为执行该方法的操作的计算机系统上执行,并且/或者可以从计算机可读存储介质处执行。该方法可以体现在机器可读和/或计算机可读存储介质中,用于配置计算机系统以执行该方法。因此,软件模块可以存储在计算机系统存储器内和/或传输到计算机系统存储器,以配置计算机系统来执行模块的功能。

[0166] 本文所述的软件模块可以由计算机系统接收,例如,从计算机可读存储介质处接收。此类计算机可读存储介质可以永久地、可移除地或远程地耦接到计算机系统。计算机可读存储介质可以非排他地包括例如以下任何数量的:磁存储介质(包括磁盘和磁带存储介质);光学存储介质,诸如光盘介质(例如,CD ROM、CD R等)和数字视频盘存储介质;非易失性存储器,包括基于半导体的存储器单元,诸如FLASH存储器、EEPROM、EPROM、ROM或专用集成电路;以及易失性存储介质(包括寄存器、缓冲区或高速缓存、主存储器、RAM等)。在基于UNIX的实施方案中,软件模块可以体现在文件中,该文件可以是设备、终端、本地或远程文件、套接字或其他此类元件。其他新的和各种类型的计算机可读存储介质还可用于存储本文讨论的软件模块。

[0167] 许多其他设备或子系统(未示出)可通过类似方式连接(例如,条形码读取器、文档扫描器、数字相机等等)。相反,图17中所示的所有设备都不必用于实施本发明。这些设备和子系统可通过不同于图17所示的方式来互连。诸如图17所示的计算机系统的操作在本领域中是公知的,并且在本申请中不再详细讨论。可将用以实施本发明的代码存储在计算机可读存储介质中,诸如存储在系统存储器1716、固定磁盘1744、CD-ROM 1742或软盘1738中的一者或多者中。另外,计算机系统1710可以是任何类型的计算设备,因此包括个人数据助理(PDA)、网络装置、X窗口终端或其他此类计算设备。计算机系统1710上提供的操作系统可以是MS-DOS[®]、MS-WINDOWS[®]、UNIX[®]、Linux[®]或其他已知的操作系统。计算机系统1710还支持许多因特网访问工具,包括例如具有JavaScript解释器的HTTP兼容网络浏览器,诸如Netscape Navigator[®]、Microsoft Internet Explorer[®]等。

[0168] 此外,关于本文所述的信号,本领域的技术人员将认识到,可以将信号从第一块直接传输到第二块,或者可以在块之间修改(例如,放大、衰减、延迟、锁存、缓冲、反转、滤波或以其他方式修改)信号。尽管上述实施方案的信号的特征在于从一个块传输到下一个块,但是本发明的其他实施方案可以包括所修改的信号来代替此类直接传输的信号,只要信号的信息和/或功能方面在块之间传输即可。在某种程度上,由于所涉及的电路的物理限制(例如,将不可避免地存在一些衰减和延迟),在第二块处输入的信号可被概念化为从第一块输出的第一信号导出的第二信号。因此,如本文所用,从第一信号导出的第二信号包括第一信号或对第一信号的任何修改,无论是由于电路限制还是由于通过不改变第一信号的信息和/或最终功能方面的其他电路元件。

[0169] 图18是描绘网络架构1800的框图,其中客户端系统1810、1820和1830以及存储服务器1840A和1840B(其中任一个可以使用计算机系统1810实现)耦接到网络1850。存储服务器1840A还被描绘为具有直接附接的存储设备1860A(1)-(N),并且存储服务器1840B被描绘为直接附接存储设备1860B(1)-(N)。存储服务器1840A和1840B也连接到SAN构造1870,但是操作不需要连接到存储区域网络。SAN构造1870支持由存储服务器1840A和1840B访问存储设备1820(1)-(N),并且由客户端系统1810、1820和1830经由网络1850访问。智能存储阵列1890还被示为可经由SAN构造1870访问的特定存储设备的示例。

[0170] 相对于计算机系统1710、调制解调器1747、网络接口1748或一些其他方法可用于提供从客户计算机系统1810、1820和1830中的每一者到网络1850的连接。客户端系统1810、1820和1830能够使用例如Web浏览器或其他客户端软件(未示出)访问存储服务器1840A或

1840B上的信息。此类客户端允许客户端系统1810、1820和1830访问由存储服务器1840A或1840B或存储设备1860A(1) - (N)、1860B(1) - (N)、1820(1) - (N)或智能存储阵列1890中的一者托管的数据。图18描绘了使用网络诸如因特网来交换数据,但是本文描述的系统不限于因特网或任何特定的基于网络的环境。

[0171] 前述实施方案,其中不同部件包含在不同的其他部件内(例如,随后讨论的作为计算机系统1710的部件示出的各种元件)。应当理解,此类描绘的架构仅仅是示例,并且实际上可以实现许多其他架构,其实现相同的功能性。在抽象但仍然明确的意义上,实现相同功能性的任何部件布置被有效地“相关联”,使得实现期望的功能性。因此,本文组合以实现特定功能性的任何两个部件可以被视为彼此“相关联”,使得实现期望的功能性,而不管架构或中间部件。同样,如此相关联的任何两个部件也可以被视为彼此“可操作地连接”或“可操作地耦接”以实现期望的功能性。

[0172] 其他实施方案

[0173] 本文所述的系统非常适合于获得所提及的优点以及其中固有的其他优点。虽然已经通过参考特定描述描绘,描述和定义了此类系统,但是此类参考并不意味着对权利要求书的限制,并且不应推断出此类限制。本文所述的系统能够在形式和功能上进行相当大的修改、改变和等同,如在考虑本公开时相关领域的普通技术人员将想到的。所描绘和描述的实施方案仅是示例,并且绝不是穷举权利要求书的范围。

[0174] 前述的具体实施方式经由使用框图、流程图和示例阐述了本文所述的系统的各种实施方案。本领域技术人员应当理解,通过使用示例说明的每个框图部件、流程图步骤、操作和/或部件可以通过各种硬件、软件、固件或其任何组合来实现(单独地和/或共同地)。

[0175] 已经在全功能计算机系统的背景下描述了本文所述的系统;然而,本领域技术人员应当理解,本文所述的系统能够以各种形式作为程序产品分布,并且本文所述的系统同样适用,而不管用于实际执行分布的特定类型的计算机可读介质。计算机可读介质的示例包括计算机可读存储介质,以及将来开发的介质存储和分布系统。

[0176] 上述讨论的实施方案可以由执行与实施方案相关联的一个或多个任务的软件模块来实现。本文讨论的软件模块可以包括脚本、批处理或其他可执行文件。软件模块可以存储在机器可读或计算机可读存储介质上,诸如磁性软盘、硬盘、半导体存储器(例如,RAM、ROM和闪存类型的媒体)、光盘(例如,CD-ROM、CD-R和DVD)或其他类型的存储器模块。根据实施方案的用于存储固件或硬件模块的存储设备还可以包括基于半导体的存储器,其可以永久地、可移除地或远程地耦接到微处理器/存储器系统。因此,模块可以存储在计算机系统存储器内,以配置计算机系统来执行模块的功能。其他新的和各种类型的计算机可读存储介质可用于存储本文讨论的模块。

[0177] 以上描述旨在例示而不应视为限制。如根据本公开应当理解的,其他实施方案是可能的。本领域的技术人员将容易地实现提供本文公开的结构和方法所必需的步骤,并且应当理解,工艺参数和步骤顺序仅作为示例给出,并且可以改变以实现期望的结构以及在权利要求书范围内的修改。在不脱离权利要求书的范围的情况下,可以基于本文阐述的描述做出对本文公开的实施方案的变型和修改,从而在所有方面充分认识到其等同物。

[0178] 尽管已经结合若干实施方案描述了本文所述的系统,但是这些实施方案及其描述并不旨在限于本文阐述的特定形式。相反,此类实施方案旨在解决可以合理地包括在所附

权利要求书的范围内的此类另选形式、修改和等同物。

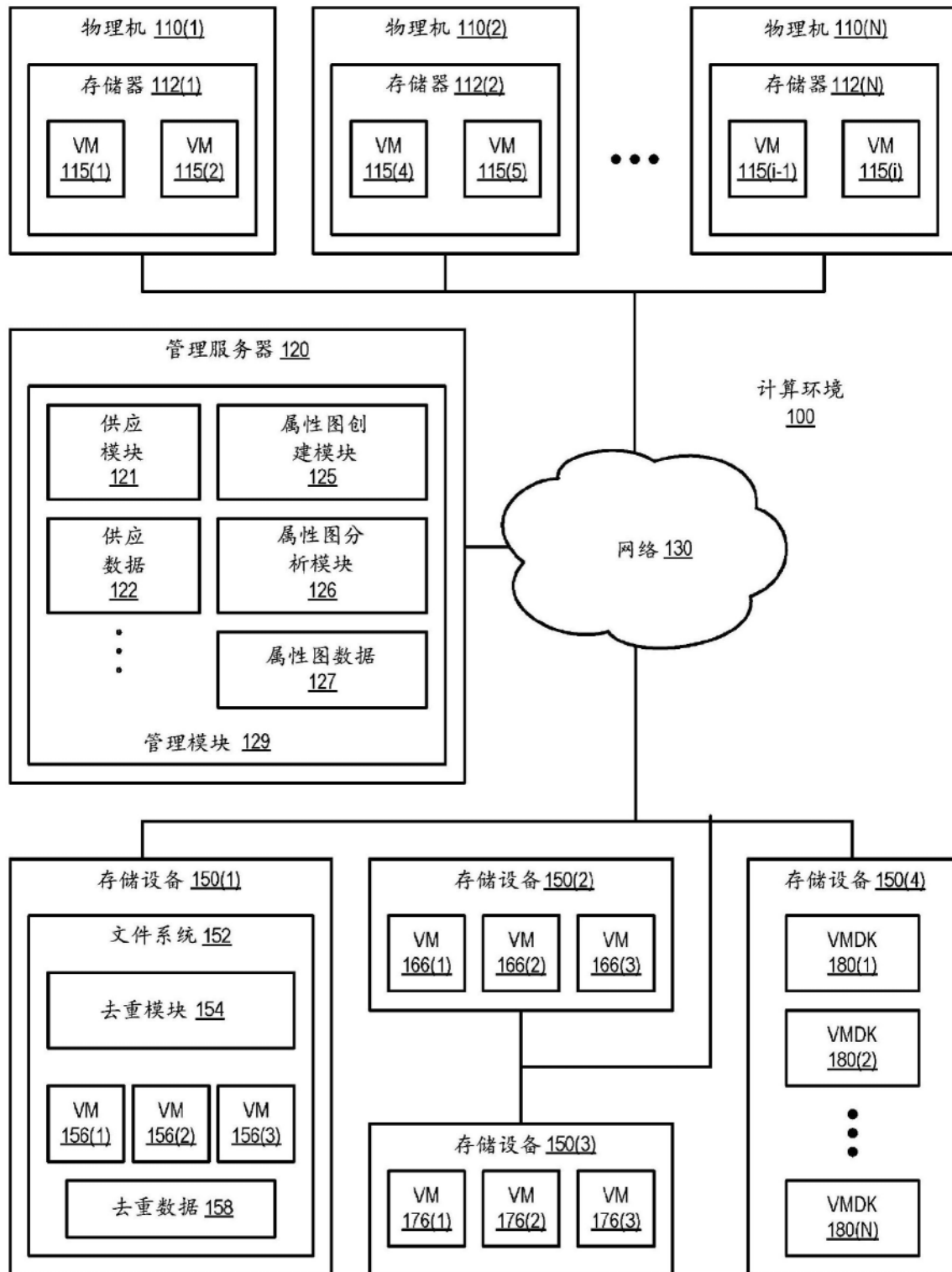


图1

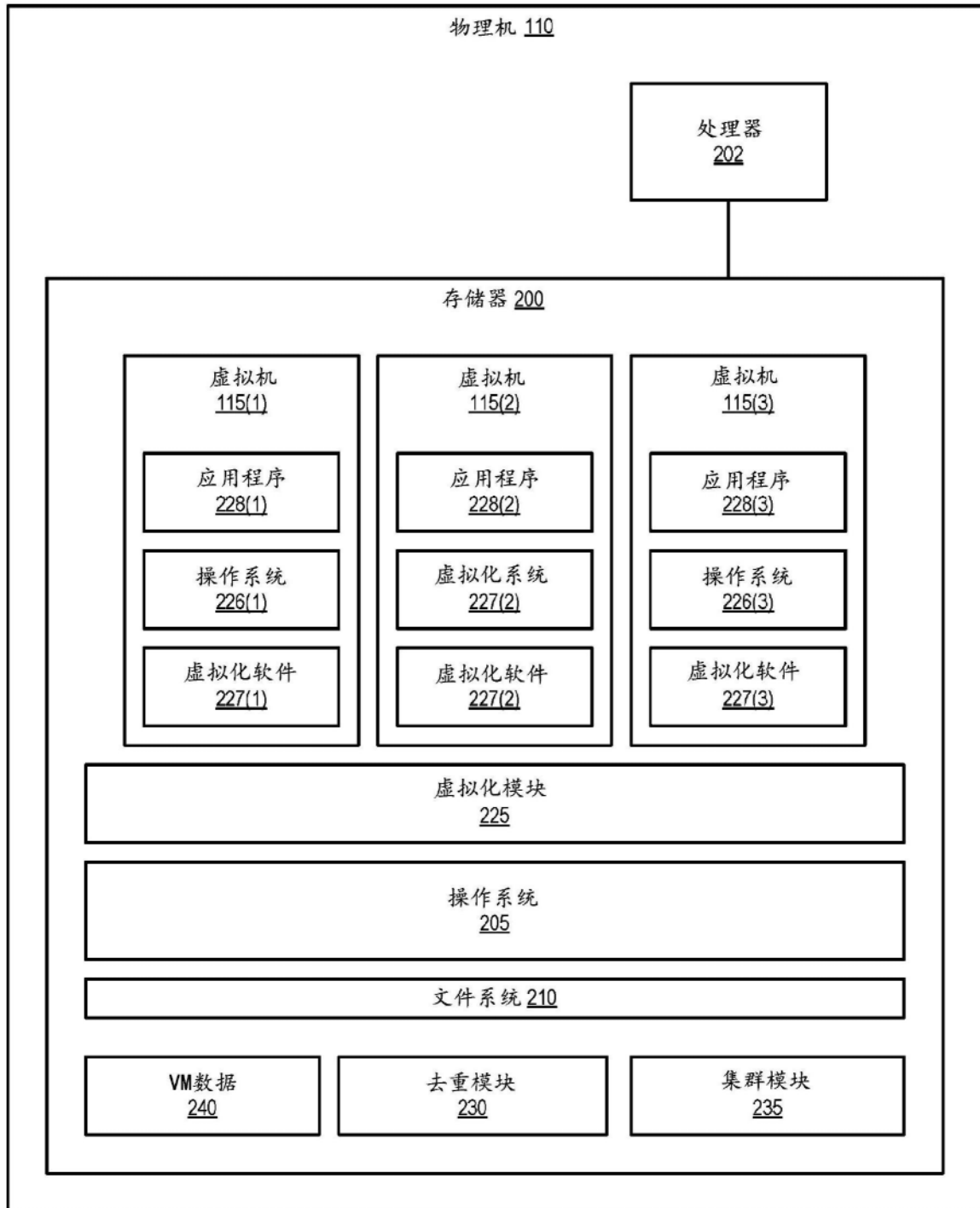


图2

计算环境
分析架构
300

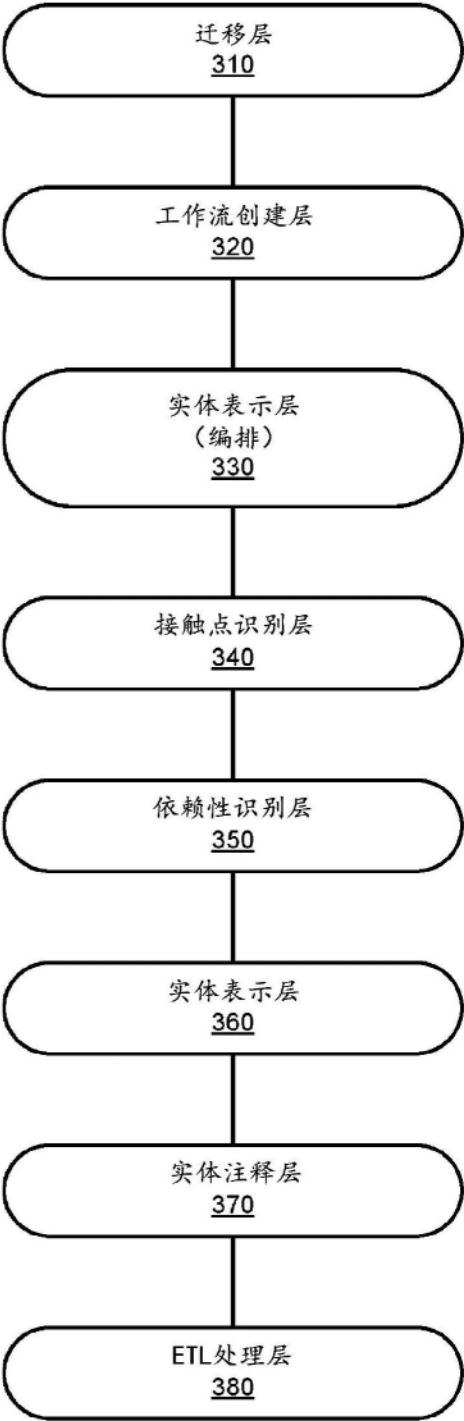


图3

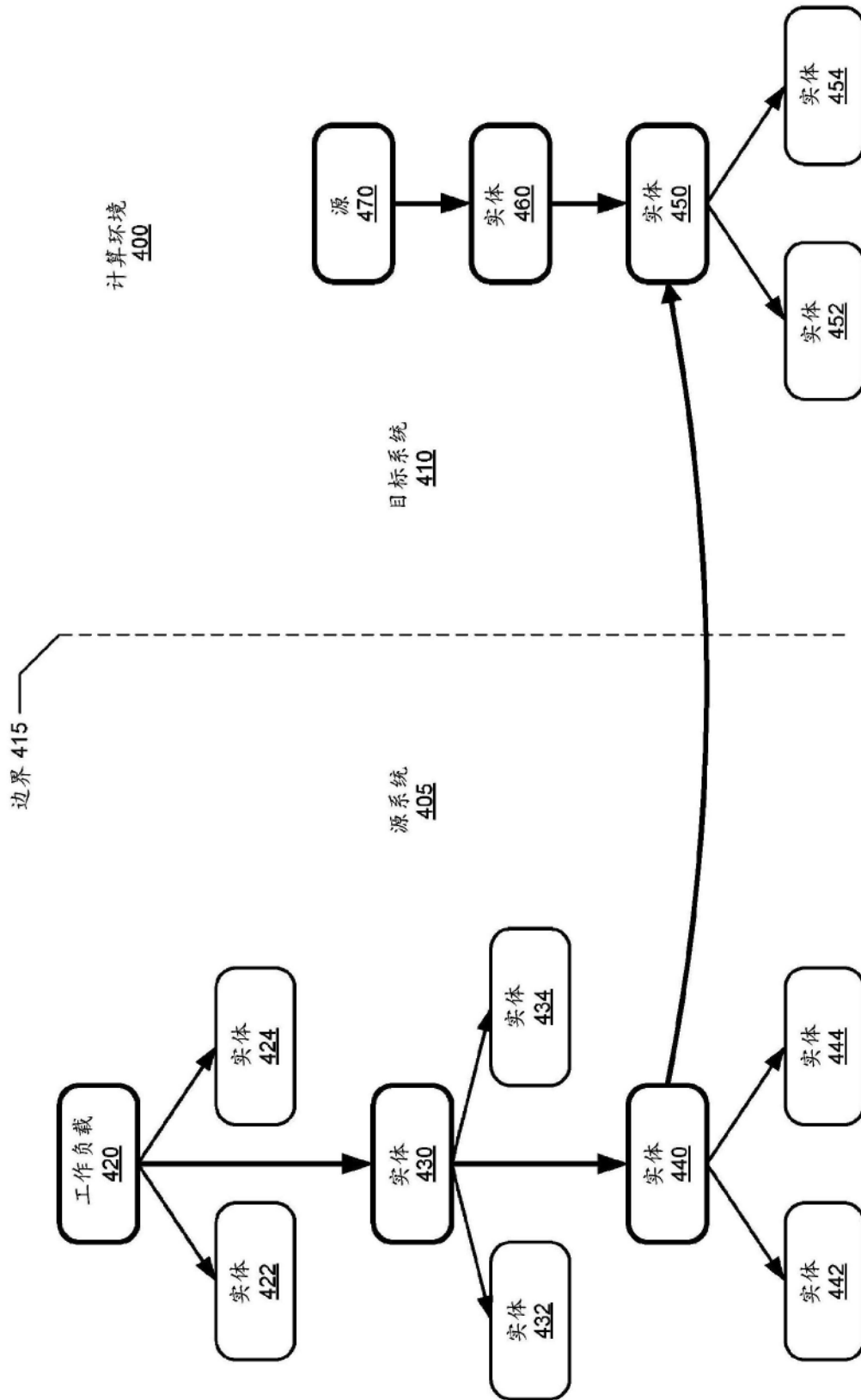


图4

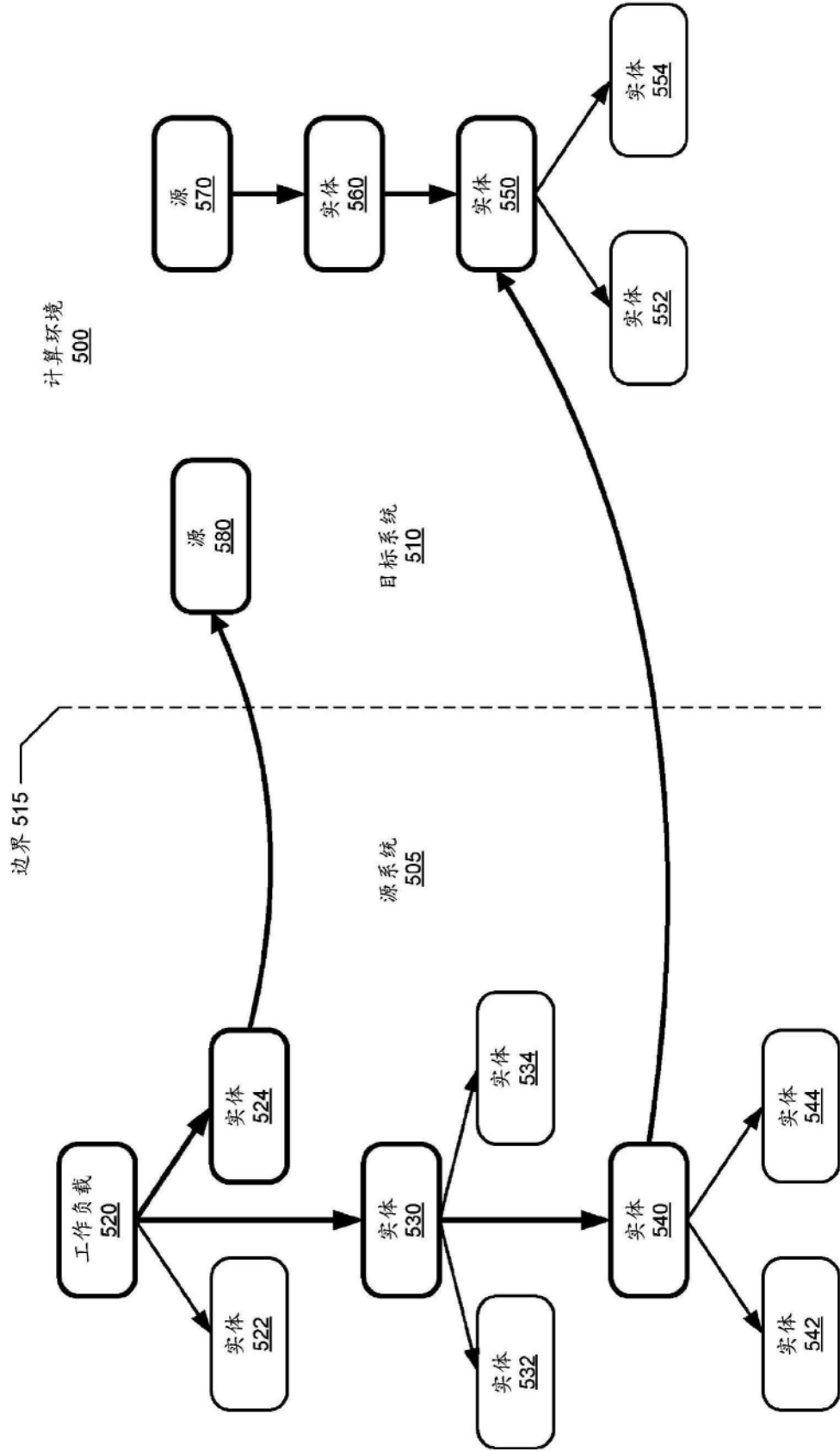


图5

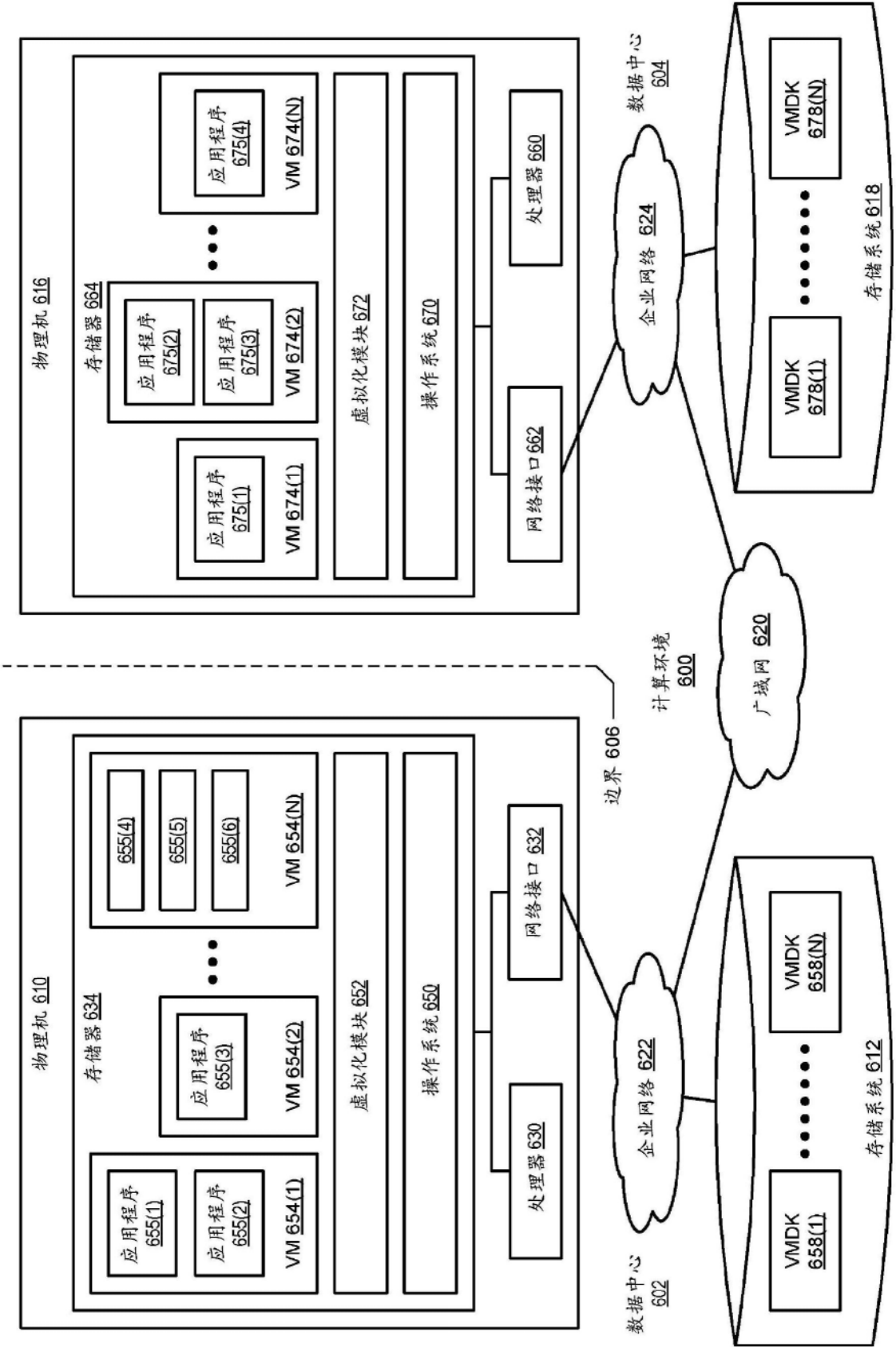


图6

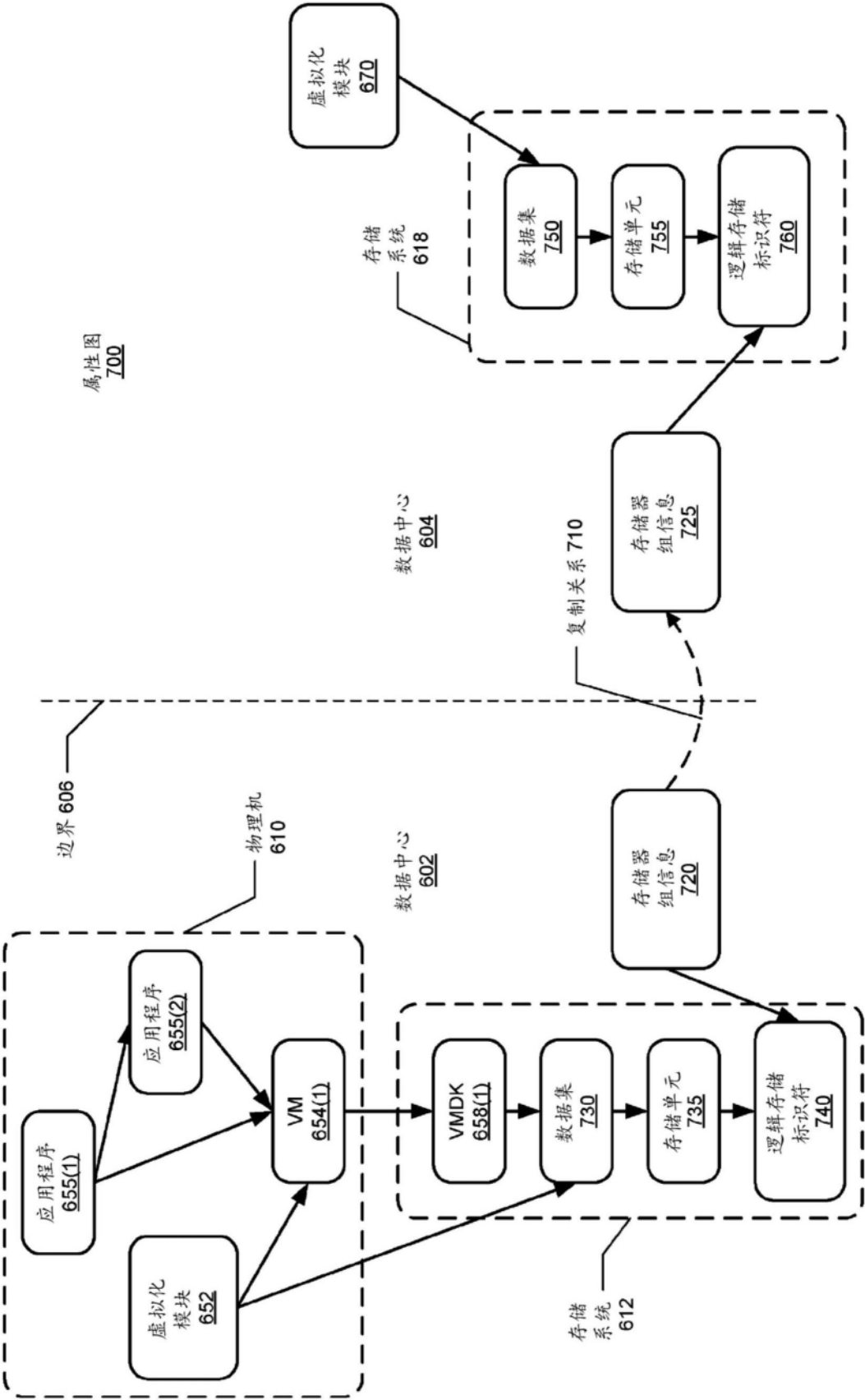


图7

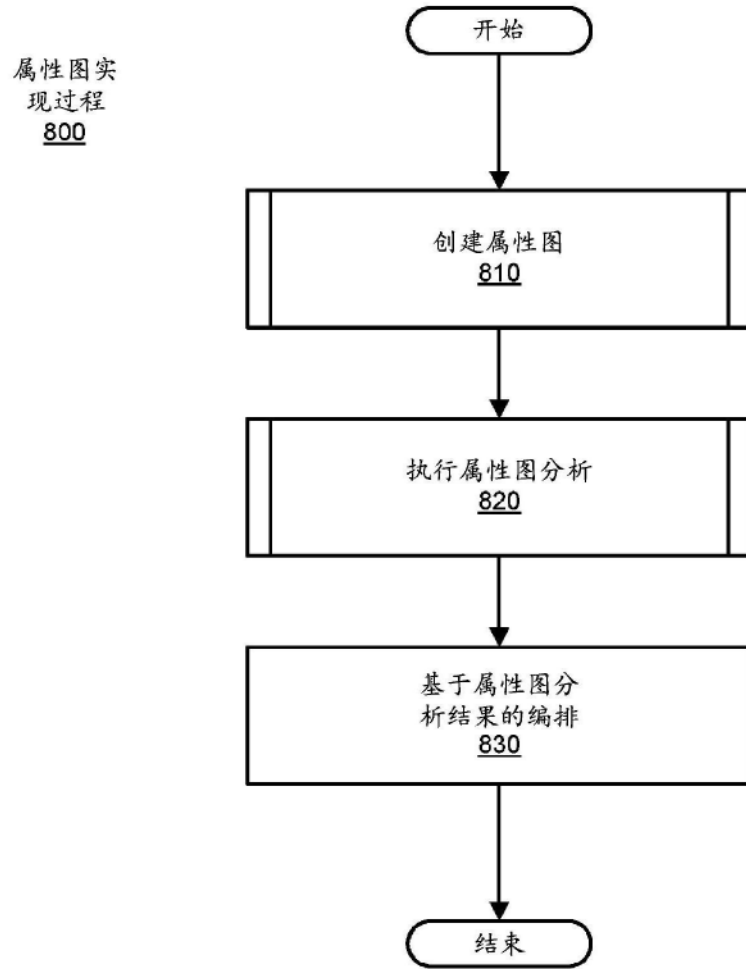


图8

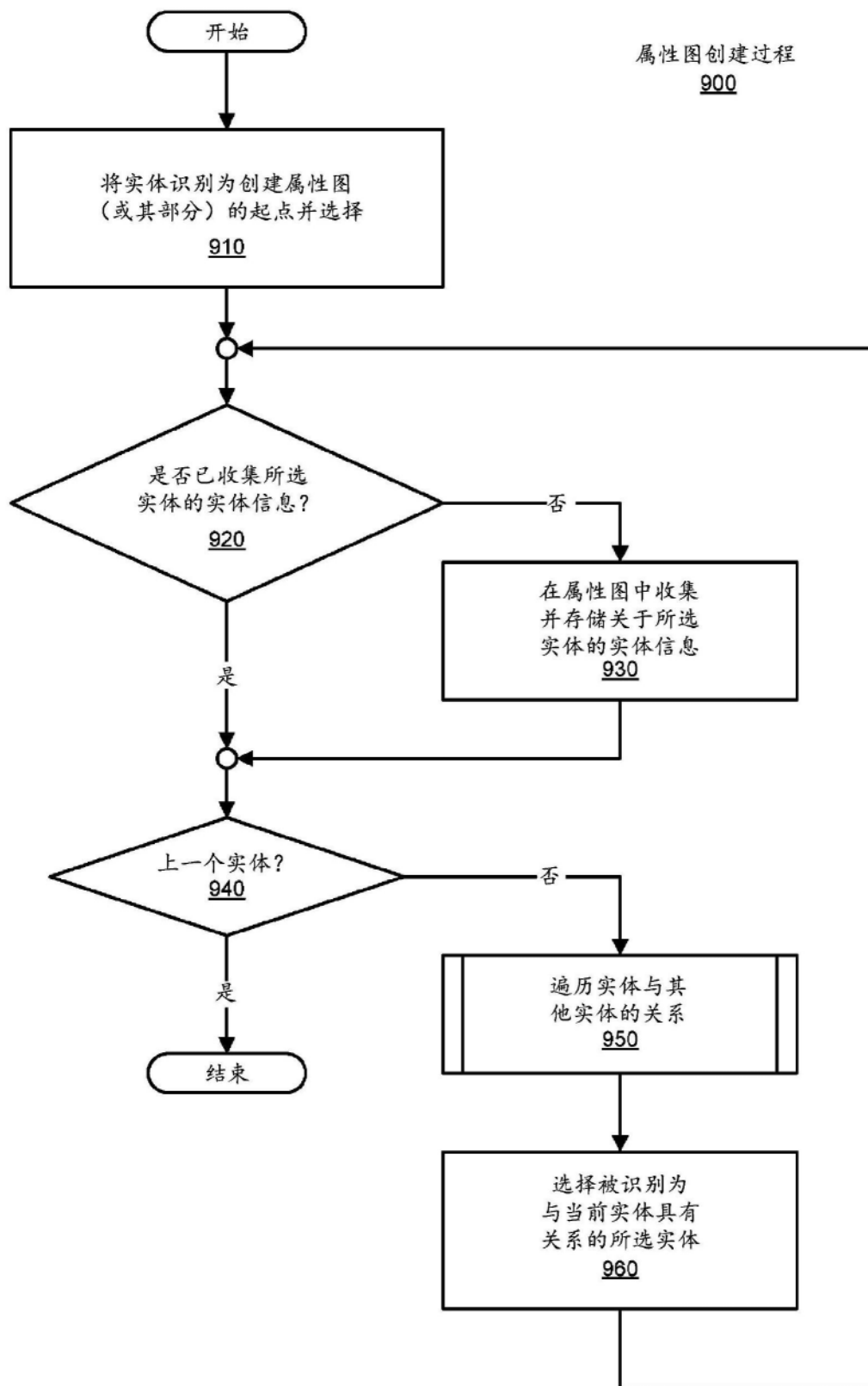


图9

属性图创建
遍历过程
1000

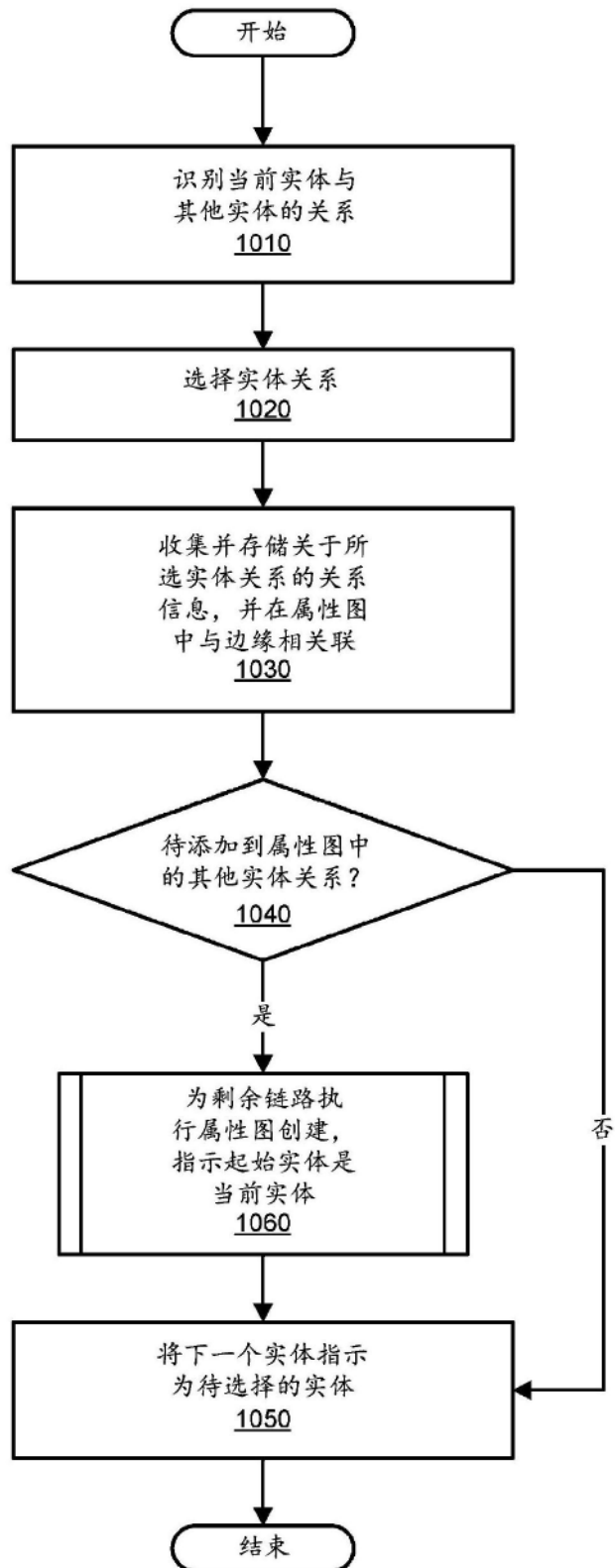


图10

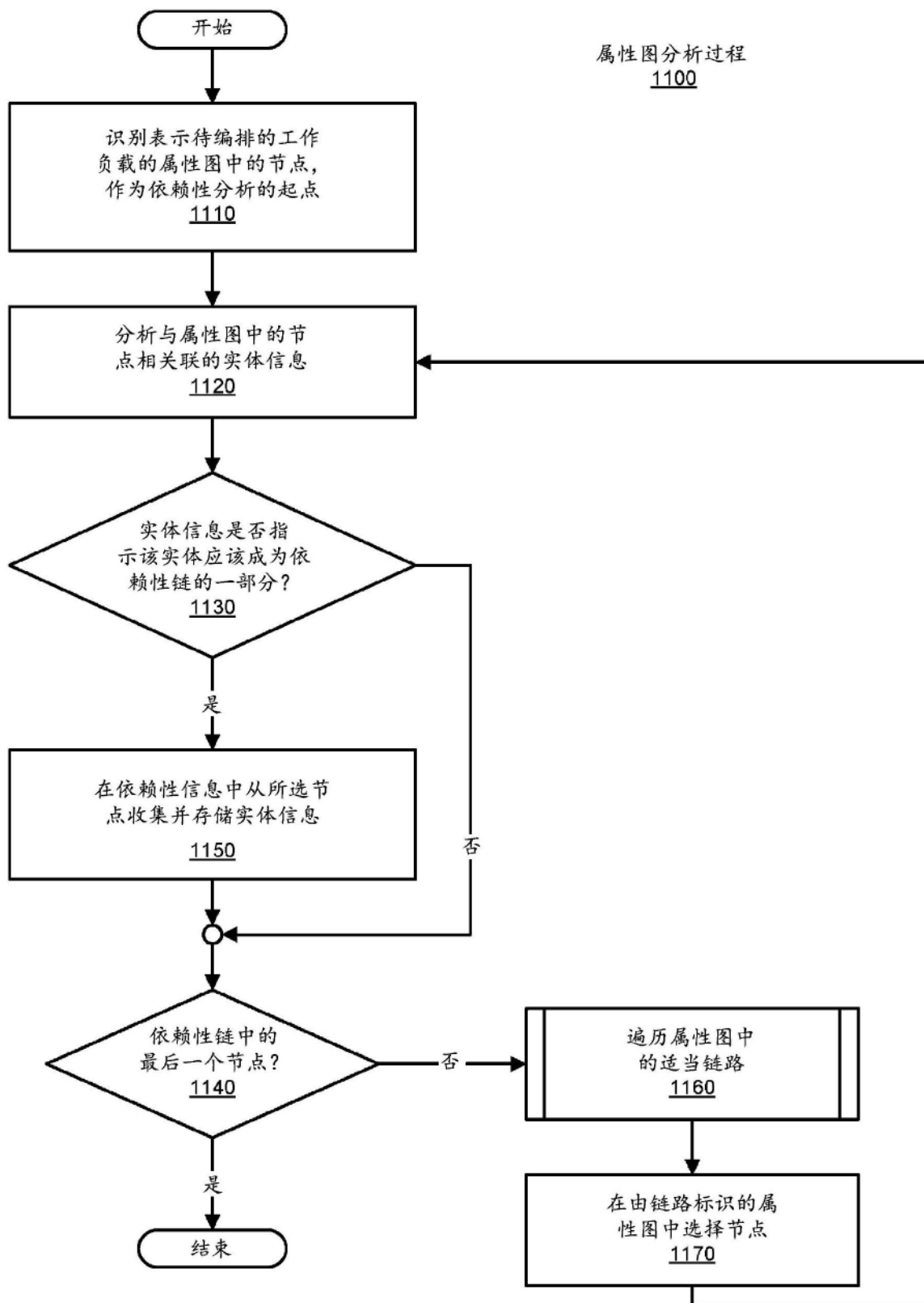


图11

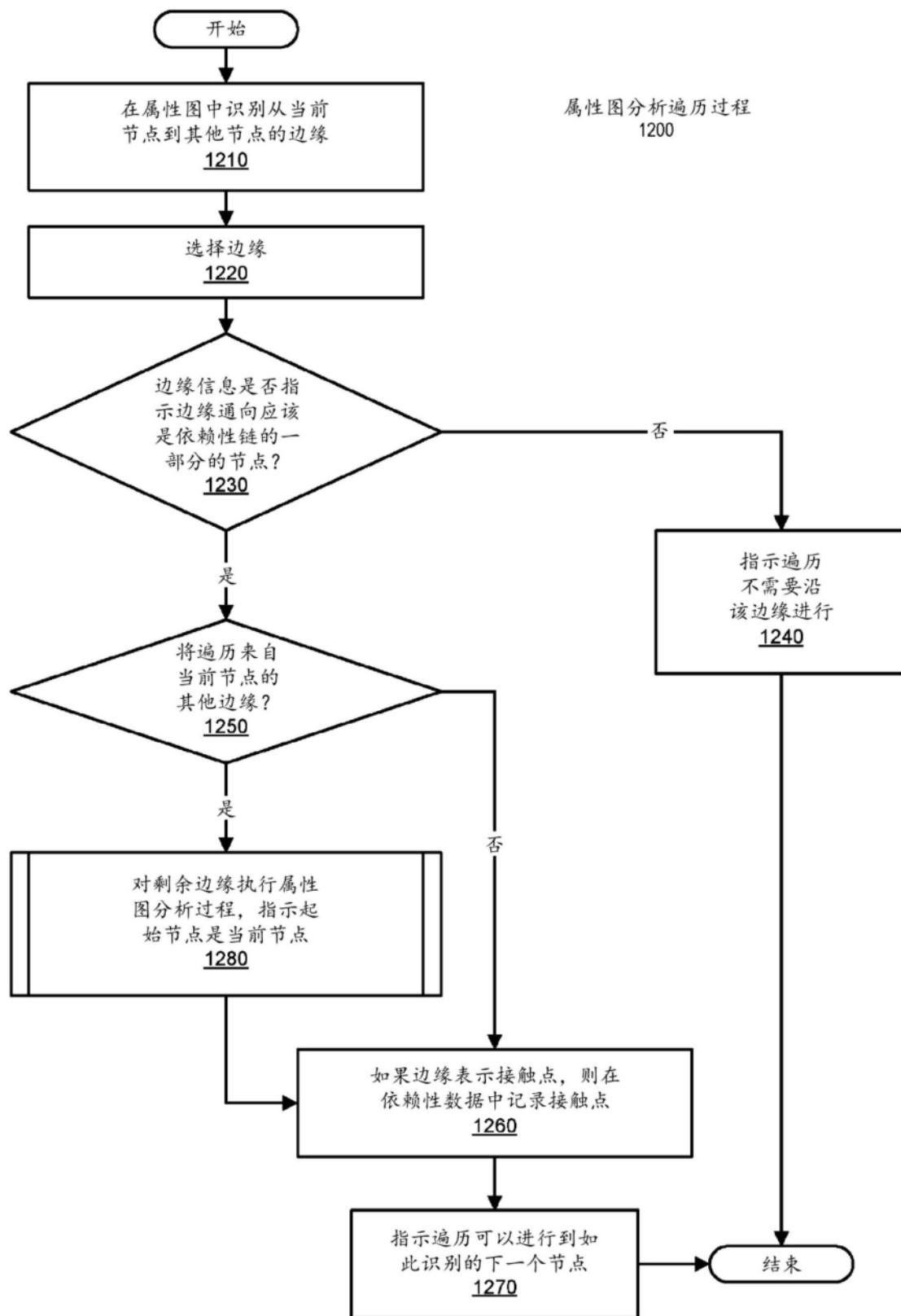


图12

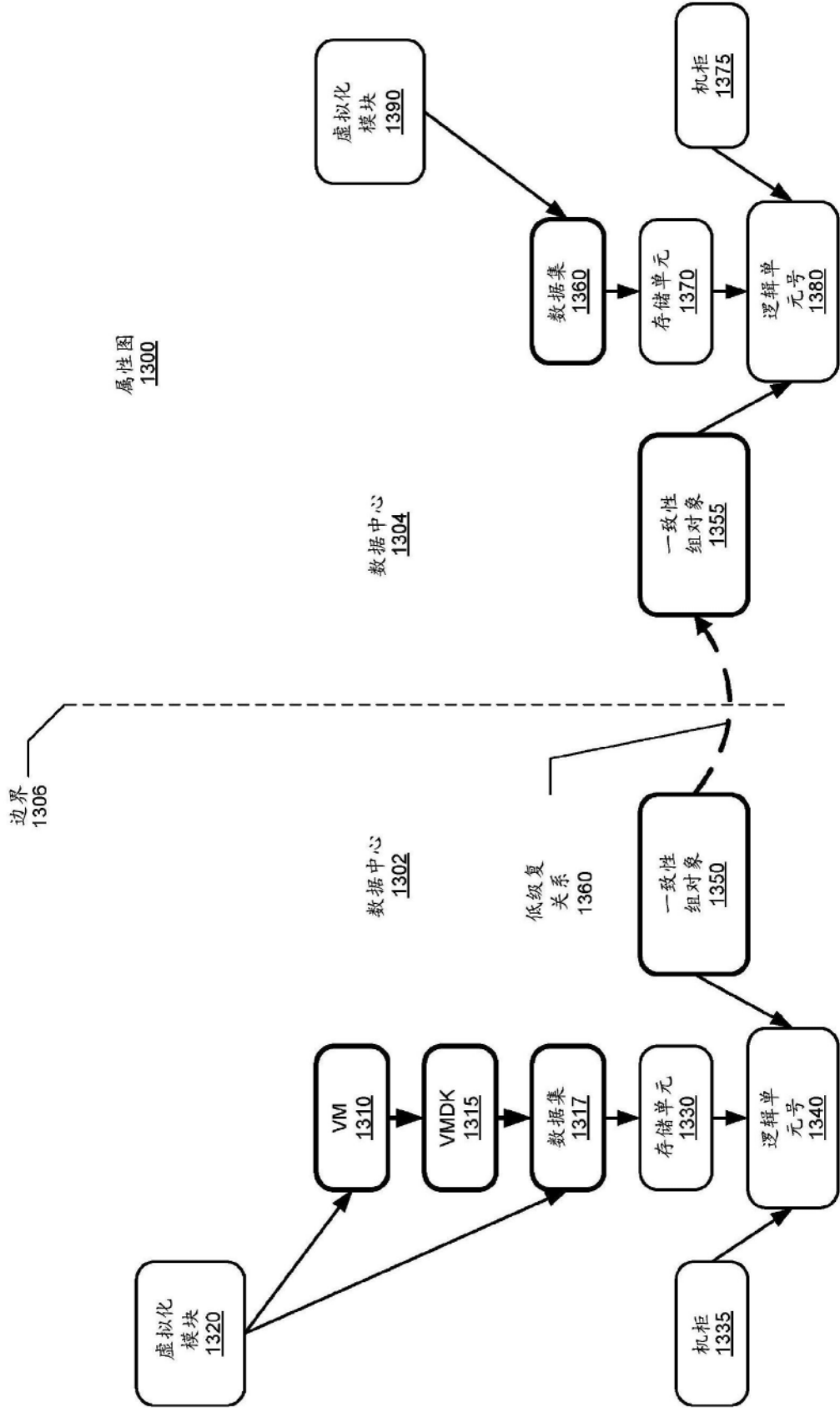


图13

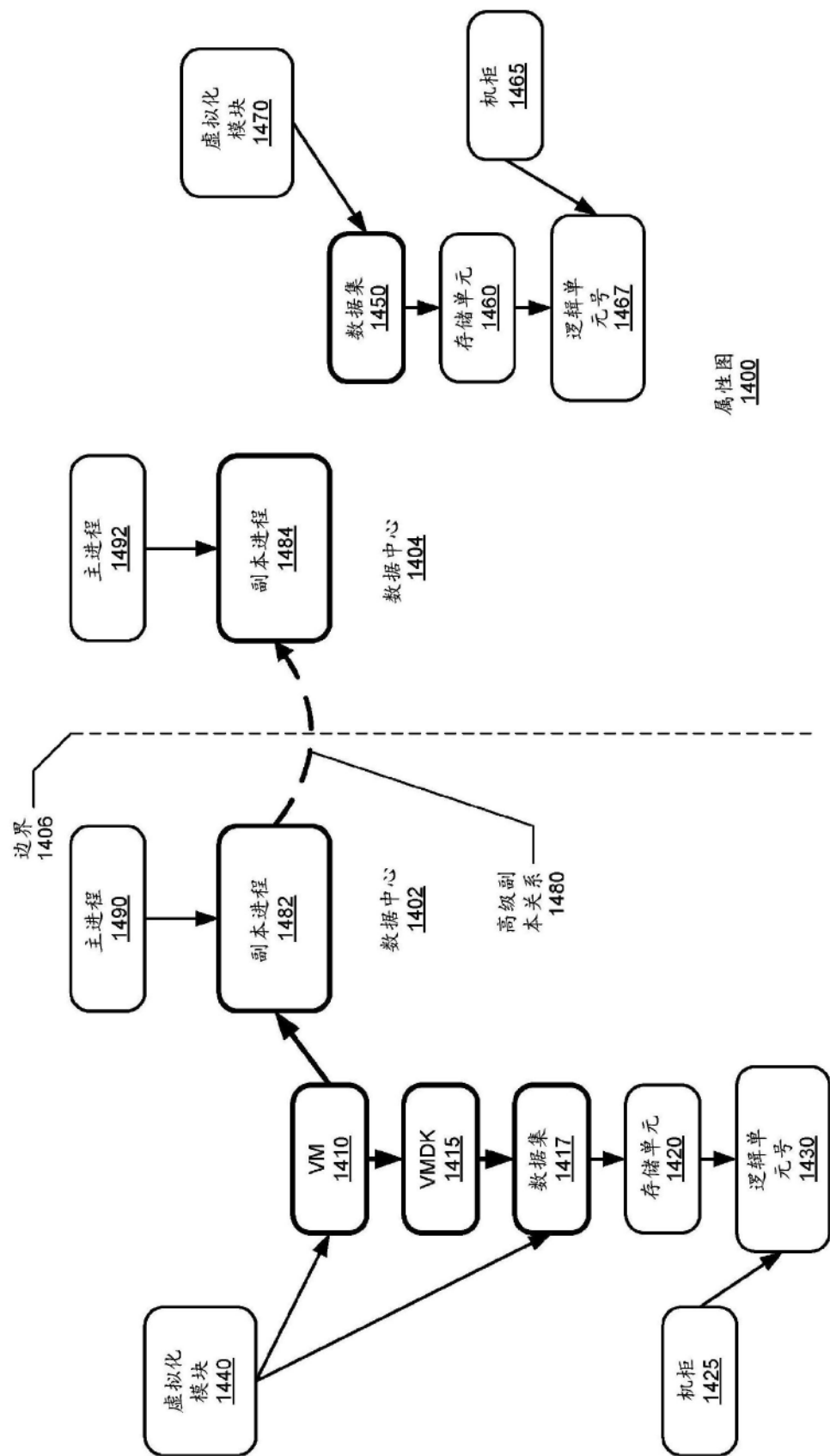


图14

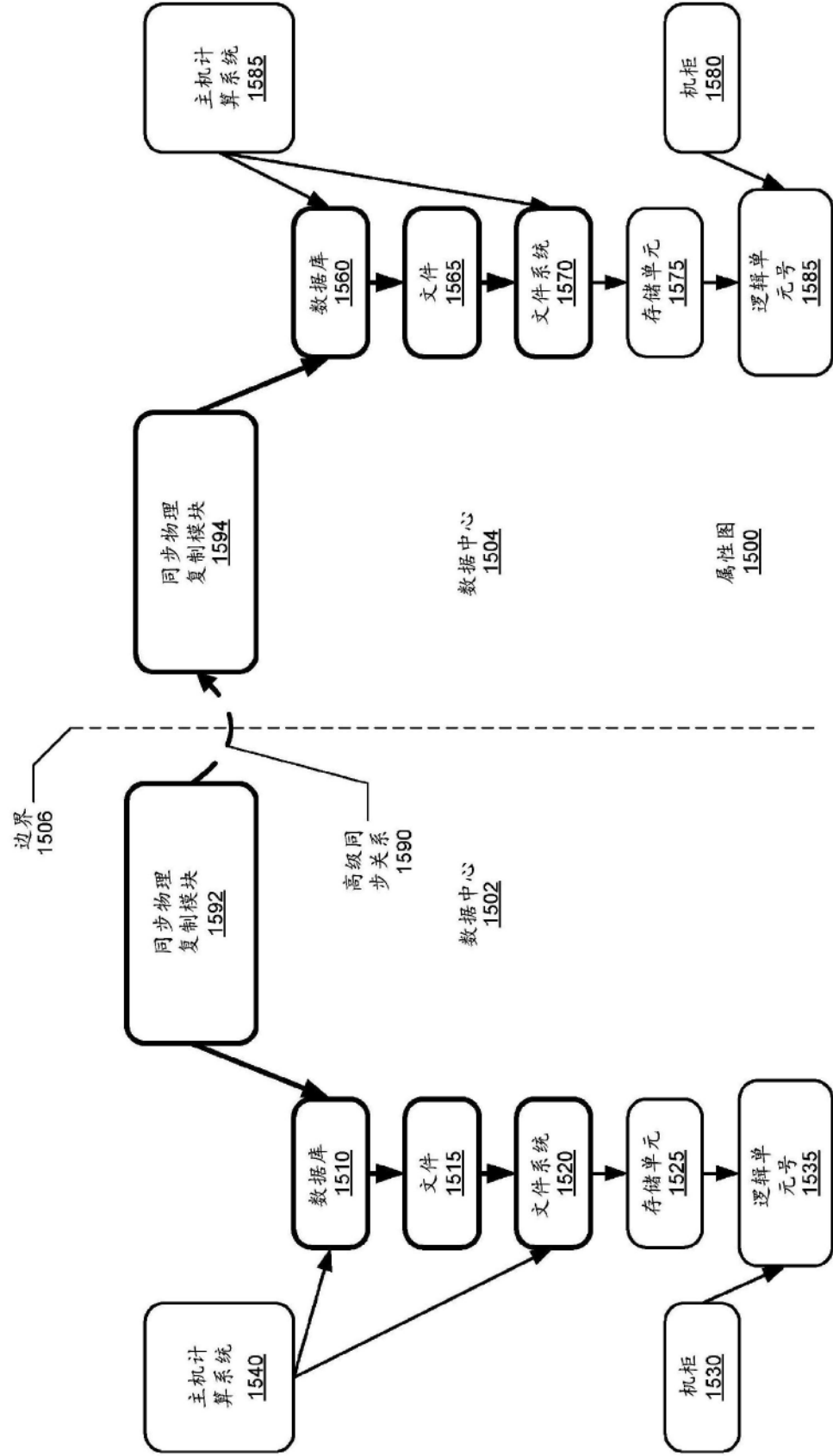


图15

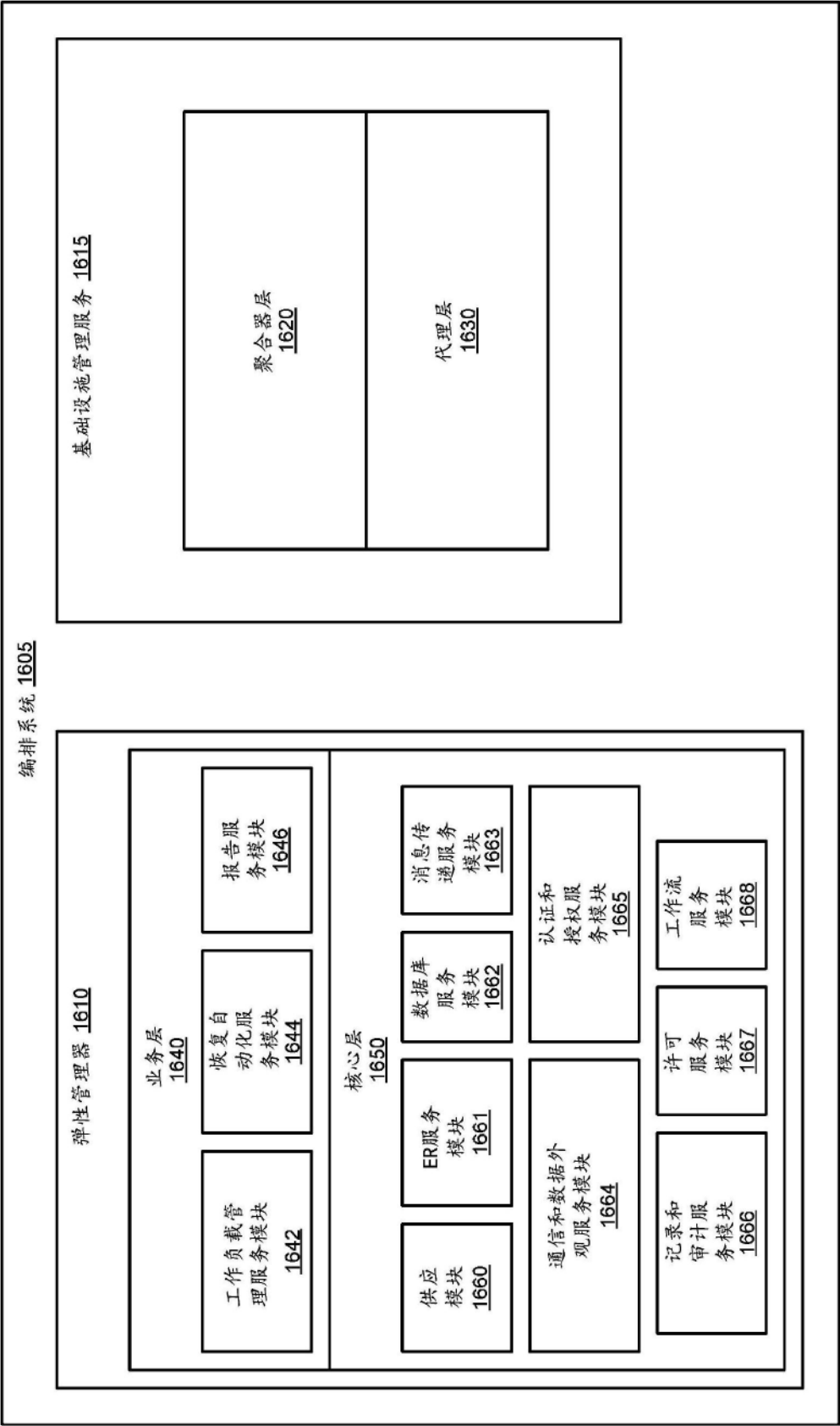


图16

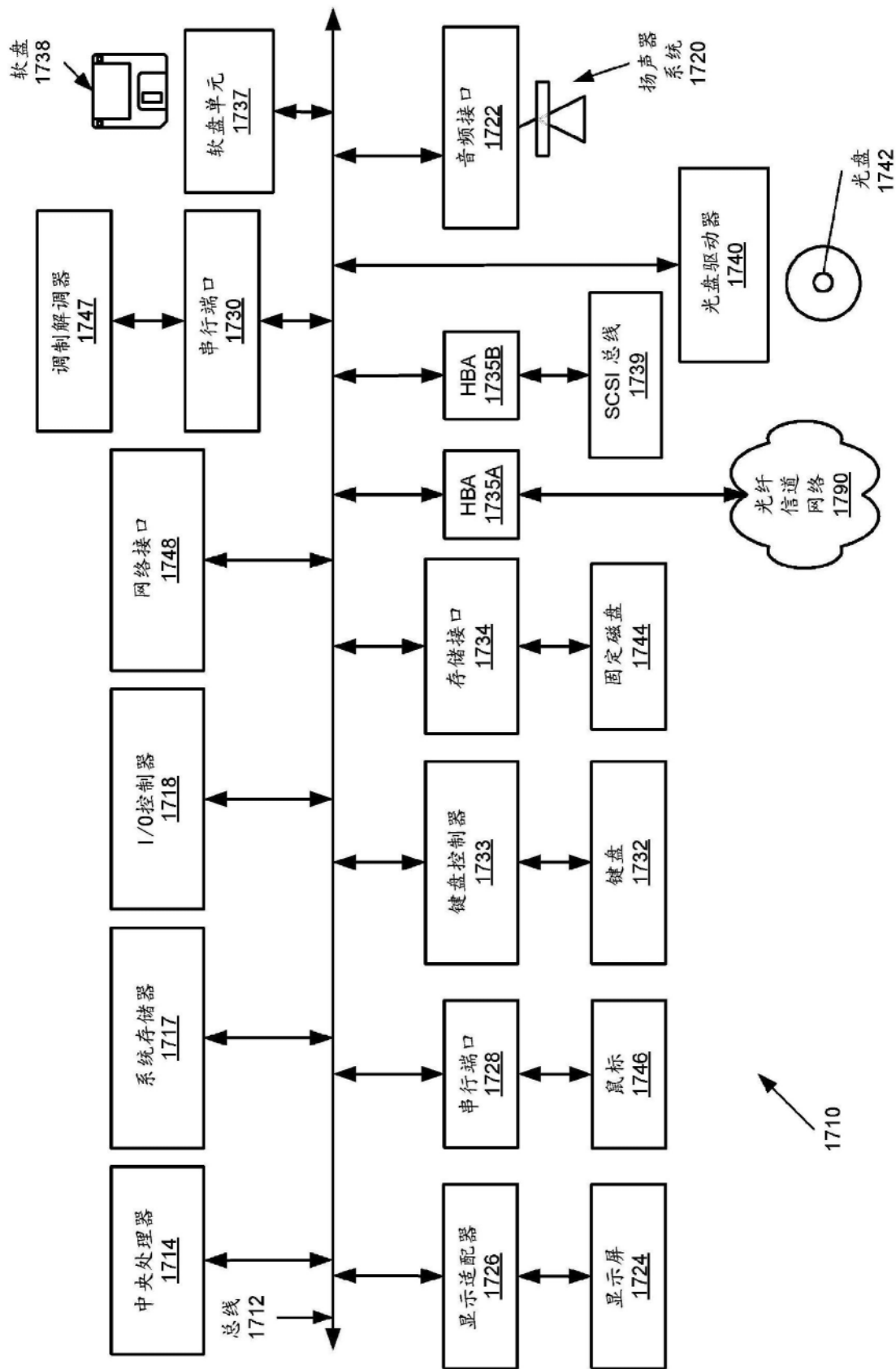


图17

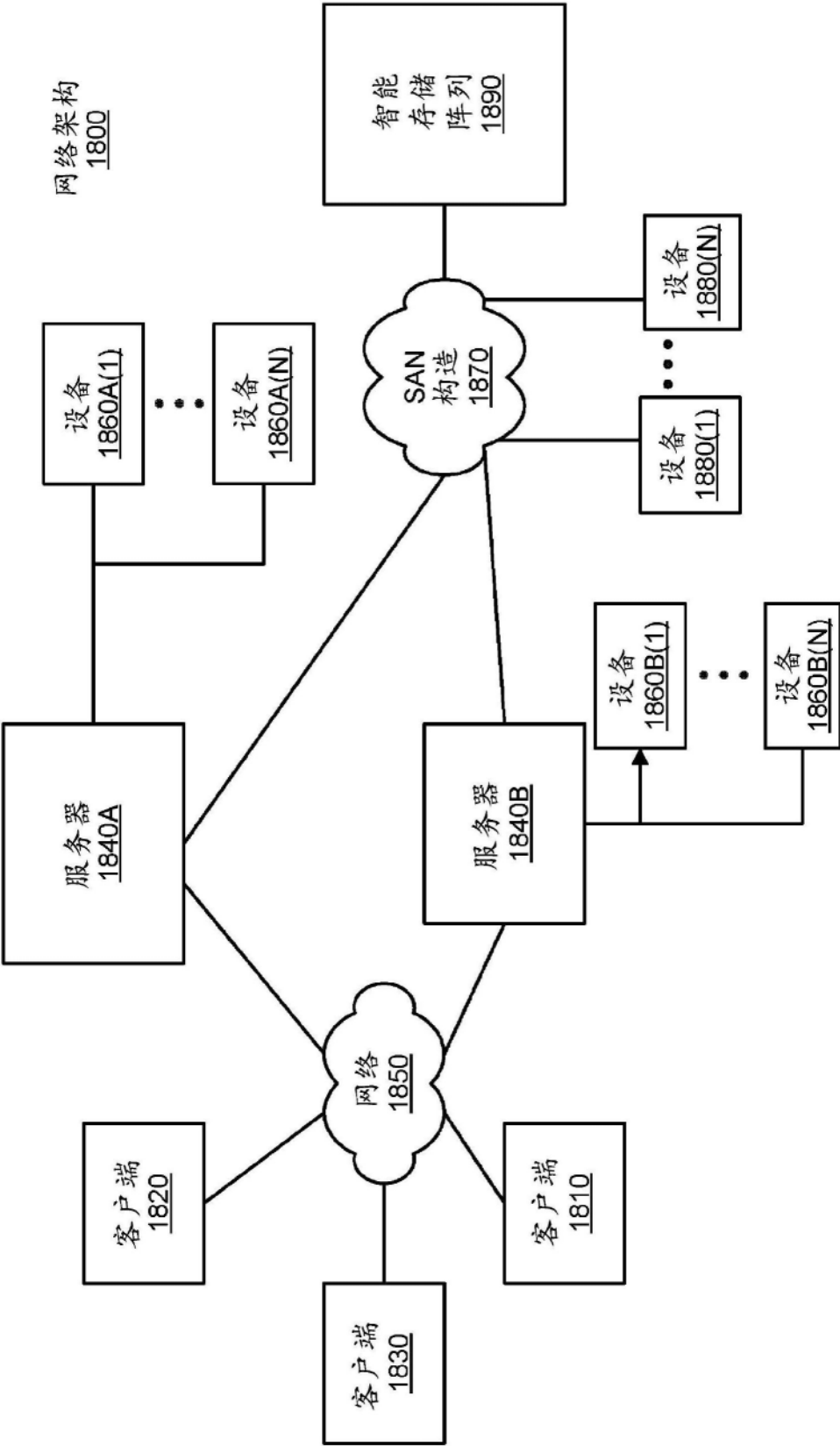


图18