(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 114586012 B
(45)授权公告日 2025.07.11

(21)申请号 202080070094.3
(22)申请日 2020.09.30
(65)同一申请的已公布的文献号
申请公布号 CN 114586012 A
(43)申请公布日 2022.06.03
(30)优先权数据
16/594,391 2019.10.07 US
(85)PCT国际申请进入国家阶段日
2022.04.06
(86)PCT国际申请的申请数据
PCT/IB2020/059141 2020.09.30
(87)PCT国际申请的公布数据
WO2021/070013 EN 2021.04.15

(73)专利权人 国际商业机器公司
地址 美国纽约阿芒克
(72)发明人 A.夸尔马 F.奥兹坎 R.阿罗泰比雷川
(74)专利代理机构 北京市柳沈律师事务所 11105
专利代理师 陈金林

(51)Int.Cl.
*G06F 9/50*(2006.01)
(56)对比文件
CN 108984308 A,2018.12.11
US 2015347480 A1,2015.12.03
审查员 李晨

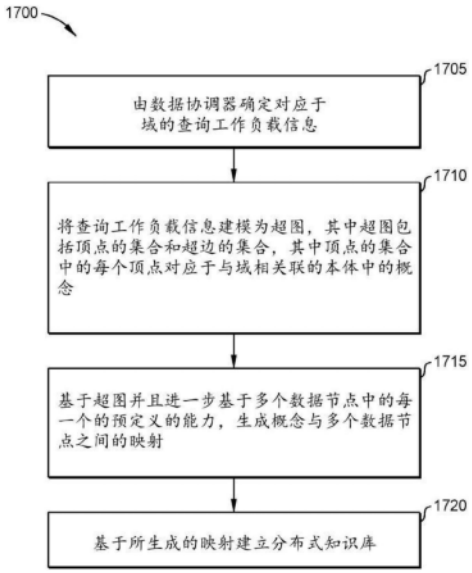
权利要求书4页 说明书26页 附图19页

(54)发明名称

# 用于分布式知识库的基于本体的数据存储

(57)摘要

提供了用于分布式数据放置的技术。由数据协调器确定对应于域的查询工作负载信息,并且将查询工作负载信息建模为超图,其中超图包括顶点的集合和超边的集合,其中顶点的集合中的每个顶点对应于与域相关联的本体中的概念。基于超图并进一步基于多个数据节点中的每一个的预定义的能力,生成概念和多个数据节点之间的映射。基于所生成的映射建立分布式知识库。




CN 114586012 B

1.一种方法,包括:

由数据协调器确定对应于域的查询工作负载信息,所述查询工作负载信息包括指定与所述域相关联的本体中的相应概念的本体查询以及由所述本体查询执行的相应操作,所述本体查询包括具有被访问的对应匹配概念的第一组,所述第一组执行操作的聚合集合;

将所述查询工作负载信息建模为超图,其中所述超图包括多个顶点和多个超边,其中所述多个顶点中的每个相应顶点对应于所述本体中的相应概念,并且所述多个超边中的每个相应超边指示被应用于与该相应超边相关联的概念的操作的相应集合,其中所述多个超边中的第一超边用操作的聚合集合来标记并且连接表示匹配概念的第一顶点;

基于所述超图并进一步基于多个数据存储中的每一个的预定义的能力,生成所述本体中的概念和所述多个数据存储之间的映射,其中所述映射对于所述本体中的每个相应概念指示在其上将存储该相应概念的所述多个数据存储的相应子集;以及

基于所生成的映射建立分布式知识库。

2.根据权利要求1所述的方法,其中确定所述查询工作负载信息包括:

接收先前本体查询的集合;

生成由先前本体查询的集合中的第一查询访问的概念的第一集合;

生成由所述第一查询执行的操作的第一集合;以及

通过以下步骤生成第一概括的查询:

从先前本体查询的集合标识具有对应的匹配第一集合的第一查询组;

基于所述第一查询组中的每个查询的操作的对应集合,确定操作的聚合集合;以及

将第一概括的查询与操作的聚合集合和在对应的匹配第一集合中反映的概念相关联。

3.根据权利要求2所述的方法,其中将所述查询工作负载信息建模为超图包括:

为所述本体中的每个概念创建顶点;

为第一概括的查询创建所述第一超边,其中所述第一超边连接所述超图中的顶点的第一集合,其中顶点的第一集合对应于在所述匹配第一集合中反映的概念;以及

用操作的聚合集合标记所述第一超边。

4.根据权利要求1所述的方法,其中生成所述映射包括:

为包括在所述超图中的第一操作创建第一集群;

标识由所述超图中的第一超边连接的概念的第一集合;

标识由所述第一超边指示的操作的第一集合;以及

在确定操作的第一集合包括所述第一操作时,将概念的第一集合分配给所述第一集群。

5.根据权利要求4所述的方法,其中生成所述映射还包括通过以下步骤将概念的第一集合映射到一个或多个数据存储:

标识能够执行所述第一操作的数据存储的集合;以及

将概念的第一集合中的每个概念映射到所标识的数据存储的集合中的每个数据存储。

6.根据权利要求1所述的方法,其中生成所述映射包括:

标识由所述超图中的第一超边连接的概念的第一集合;

标识由所述第一超边指示的操作的第一集合;

确定能够共同执行操作的第一集合的数据存储的最小集合;

生成包括概念的第一集合的集群;以及

用数据存储的最小集合来标记所述集群。

7.根据权利要求6所述的方法,其中生成所述映射还包括将概念的第一集合中的每个概念映射到数据存储的最小集合中的每个数据存储。

8.根据权利要求1所述的方法,其中建立所述分布式知识库包括,对于所述本体中的每个相应概念:

标识由所述映射指示的相应数据存储;

标识对应于所述相应概念的数据;以及

促进在所述相应数据存储中存储所标识的数据。

9.一种包含计算机程序代码的计算机可读存储介质,当由一个或多个计算机处理器的操作来执行时,所述计算机程序代码执行包括以下步骤的操作:

由数据协调器确定对应于域的查询工作负载信息,所述查询工作负载信息包括指定与所述域相关联的本体中的相应概念的本体查询以及由所述本体查询执行的相应操作,所述本体查询包括具有被访问的对应匹配概念的第一组,所述第一组执行操作的聚合集合;

将所述查询工作负载信息建模为超图,其中所述超图包括多个顶点和多个超边,其中所述多个顶点中的每个相应顶点对应于所述本体中的相应概念,并且所述多个超边中的每个相应超边指示被应用于与该相应超边相关联的概念的操作的相应集合,其中所述多个超边中的第一超边用操作的聚合集合来标记并且连接表示匹配概念的第一顶点;

基于所述超图并进一步基于多个数据存储中的每一个的预定义的能力,生成所述本体中的概念和所述多个数据存储之间的映射,所述映射对于所述本体中的每个相应概念指示在其上将存储该相应概念的所述多个数据存储的相应子集;以及

基于所生成的映射建立分布式知识库。

10.根据权利要求9所述的计算机可读存储介质,其中确定所述查询工作负载信息包括:

接收先前本体查询的集合;

生成由先前本体查询的集合中的第一查询访问的概念的第一集合;

生成由所述第一查询执行的操作的第一集合;以及

通过以下步骤生成第一概括的查询:

从先前本体查询的集合标识具有对应的匹配第一集合的第一查询组;

基于所述第一查询组中的每个查询的操作的对应集合,确定操作的聚合集合;以及

将第一概括的查询与操作的聚合集合和在对应的匹配第一集合中反映的概念相关联。

11.根据权利要求10所述的计算机可读存储介质,其中将所述查询工作负载信息建模为超图包括:

为所述本体中的每个概念创建顶点;

为第一概括的查询创建所述第一超边,其中所述第一超边连接所述超图中的顶点的第一集合,其中顶点的第一集合对应于在所述匹配第一集合中反映的概念;以及

用操作的聚合集合标记所述第一超边。

12.根据权利要求9所述的计算机可读存储介质,其中生成所述映射包括:

为包括在所述超图中的第一操作创建第一集群;

标识由所述超图中的第一超边连接的概念的第一集合;

标识由所述第一超边指示的操作的第一集合;以及

在确定操作的第一集合包括所述第一操作时,将概念的第一集合分配给所述第一集群。

13.根据权利要求12所述的计算机可读存储介质,其中生成所述映射还包括通过以下步骤将概念的第一集合映射到一个或多个数据存储:

标识能够执行所述第一操作的数据存储的集合;以及

将概念的第一集合中的每个概念映射到所标识的数据存储的集合中的每个数据存储。

14.根据权利要求9所述的计算机可读存储介质,其中生成所述映射包括:

标识由所述超图中的第一超边连接的概念的第一集合;

标识由所述第一超边指示的操作的第一集合;

确定能够共同执行操作的第一集合的数据存储的最小集合;

生成包括概念的第一集合的集群;以及

用数据存储的最小集合来标记所述集群。

15.根据权利要求9所述的计算机可读存储介质,其中建立所述分布式知识库包括,对于所述本体中的每个相应概念:

标识由所述映射指示的相应数据存储;

标识对应于所述相应概念的数据;以及

促进在所述相应数据存储中存储所标识的数据。

16.一种系统,包括:

一个或多个计算机处理器;以及

包含程序的存储器,所述程序在由一个或多个计算机处理器执行时执行操作,所述操作包括:

由数据协调器确定对应于域的查询工作负载信息,所述查询工作负载信息包括指定与所述域相关联的本体中的相应概念的本体查询以及由所述本体查询执行的相应操作,所述本体查询包括具有被访问的对应匹配概念的第一组,所述第一组执行操作的聚合集合;

将所述查询工作负载信息建模为超图,其中所述超图包括多个顶点和多个超边,其中所述多个顶点中的每个相应顶点对应于所述本体中的相应概念,并且所述多个超边中的每个相应超边指示被应用于与该相应超边相关联的概念的操作的相应集合,其中所述多个超边中的第一超边用操作的聚合集合来标记并且连接表示匹配概念的第一顶点;

基于所述超图并进一步基于多个数据存储中的每一个的预定义的能力,生成所述本体中的概念和所述多个数据存储之间的映射,其中所述映射对于所述本体中的每个相应概念指示在其上将存储该相应概念的所述多个数据存储的相应子集;以及

基于所生成的映射建立分布式知识库。

17.根据权利要求16所述的系统,其中确定所述查询工作负载信息包括:

接收先前本体查询的集合;

生成由先前本体查询的集合中的第一查询访问的概念的第一集合;

生成由所述第一查询执行的操作的第一集合;以及

通过以下步骤生成第一概括的查询:

从先前本体查询的集合标识具有对应的匹配第一集合的第一查询组;

基于所述第一查询组中的每个查询的操作的对应集合,确定操作的聚合集合;以及

将第一概括的查询与操作的聚合集合和在对应的匹配第一集合中反映的概念相关联。

18.根据权利要求17所述的系统,其中将所述查询工作负载信息建模为超图包括:

为所述本体中的每个概念创建顶点;

为第一概括的查询创建所述第一超边,其中所述第一超边连接所述超图中的顶点的第一集合,其中顶点的第一集合对应于在所述匹配第一集合中反映的概念;以及

用操作的聚合集合标记所述第一超边。

19.根据权利要求16所述的系统,其中生成所述映射包括:

为包括在所述超图中的第一操作创建第一集群;

标识由所述超图中的第一超边连接的概念的第一集合;

标识由所述第一超边指示的操作的第一集合;以及

在确定操作的第一集合包括所述第一操作时,将概念的第一集合分配给所述第一集群。

20.根据权利要求16所述的系统,其中生成所述映射包括:

标识由所述超图中的第一超边连接的概念的第一集合;

标识由所述第一超边指示的操作的第一集合;

确定能够共同执行操作的第一集合的数据存储的最小集合;

生成包括概念的第一集合的集群;以及

用数据存储的最小集合来标记所述集群。

# 用于分布式知识库的基于本体的数据存储

## 技术领域

[0001] 本公开涉及知识库,并且更具体地说,涉及使用本体在分布式知识库中进行有效的数据放置。

## 背景技术

[0002] 越来越多的企业正在利用知识库(KB)来增强其分析并改善其系统的决策做出、效率和有效性。通常,在企业的域内,KB相对专用。例如,金融机构依赖于具有重要金融知识的KB,诸如与金融市场的政府规章有关的数据。相反,健康护理企业可以维护具有从医学文献收集的相当大量的数据的KB。存在对管理这些KB的深度域专门化、有效系统和有效技术的实质需要。不知道提供域模式的实体中心视图的域本体的现有系统,不能有效地管理和路由对KB的查询。

[0003] 另外,KB可以分布在具有不同能力和成本的多个数据站点上,以便改善操作。诸如联合数据库的现有架构依赖于集中式中介器来聚合来自每个这样的源的数据。这是低效的并且缩放性差。另外,现有系统不理解每个数据源的底层本体和能力,因此不能有效地路由查询。这降低了这种系统的效率,需要大量的计算资源来响应典型的查询。

## 发明内容

[0004] 根据本发明的一个方面,提供了一种方法,包括:由数据协调器确定对应于域的查询工作负载信息。该方法还包括将查询工作负载信息建模为超图,其中超图包括顶点的集合和超边的集合,其中顶点的集合中的每个顶点对应于与域相关联的本体中的概念。另外,该方法包括基于基于超图并进一步基于多个数据节点中的每一个的预定义的能力,生成概念和多个数据节点之间的映射,并且基于所生成的映射建立分布式知识库。有利地,该方法使得数据协调器能够基于现有工作负载和每个数据节点的能力在分布式环境中有效地放置和存储数据。这减少了存储数据所需的计算开销,并且通过提供有效的数据映射进一步改善了系统响应性。

[0005] 根据本公开的另一实施例,确定查询工作负载信息包括接收先前本体查询的集合。根据该实施例的方法包括生成由先前本体查询的集合中的第一查询访问的概念的第一集合,以及生成由第一查询执行的操作的第二集合。通过以下步骤生成第一概括查询:从先前本体查询的集合标识具有对应的匹配第一集合的查询组并且基于所标识的查询组中的每个查询的对应的第二集合,确定操作的聚合集合。然后将第一概括的查询与操作的聚合集合和在对应的匹配第一集合中反映的概念相关联。在这样的实施例中,数据协调器通过有效地概括先前查询以确定数据的良好存储计划来改进现有系统,以便满足预期的需要。这再次提高了效率并减少了运行时的计算浪费。

[0006] 根据本公开的又一实施例,将查询工作负载信息建模为超图包括为本体中的每个概念创建顶点并且为第一概括的查询创建第一超边,其中第一超边连接超图中的顶点的第一集合,其中顶点的第一集合对应于在匹配第一集合中反映的概念。在一个这样的实施例

中,该方法还包括用操作的聚合集合标记第一超边。有利地,这样的实施例使得数据协调器能够有效地以图形式表示工作负载,这允许协调器更好且更有效地评估数据以驱动改进的放置决策。这显著地提高了运行时间性能。

[0007] 根据本公开的又一实施方式,生成映射包括为包括在超图中的第一操作创建第一集群。该实施例然后包括标识由超图中的第一超边连接的概念的第一集合,以及标识由第一超边指示的操作的第一集合。该方法包括在确定操作的第一集合包括第一操作时,将概念的第一集合分配给第一集群。这种实施例的一个优点在于,其使得能够高效地评估超图,并且导致在运行时需要最小移动的高度可靠的数据放置。

[0008] 根据本公开的另一实施例,生成映射还包括通过以下步骤将概念的第一集合映射到一个或多个数据节点:标识能够执行第一操作的数据节点的集合,以及将概念的第一集合中的每个概念映射到所标识的数据节点的集合中的每个数据节点。有利地,这使得系统能够基于节点的能力生成数据映射,同时考虑先前工作负载。这提高了系统将被充分地定位以便以最小的等待时间和资源消耗来响应将来的查询的可能性。

[0009] 根据本公开的又一实施方式,其中生成映射包括标识由超图中的第一超边连接的概念的第一集合,标识由第一超边指示的操作的第一集合,以及确定能够共同执行操作的第一集合的数据节点的最小集合。该方法然后包括生成包括概念的第一集合的集群,以及用数据节点的最小集合来标记集群。这种实施例通过最小化系统中数据的复制而改进了现有的解决方案,这降低了存储成本,并且另外降低了由系统中的数据移动引起的延迟和资源消耗。

[0010] 根据本公开的另一实施例,生成映射还包括将概念的第一集合中的每个概念映射到数据节点的最小集合中的每个数据节点。这类似地减少了系统的存储成本和等待时间。

[0011] 根据本公开的又一实施方式,建立分布式知识库包括,对于本体中的每个相应概念,标识由映射指示的相应数据节点,标识对应于相应概念的数据,以及促进在相应数据节点中存储所标识的数据。有利地,这样的实施例使得协调器能够以高效的方式有效地生成映射并将数据放置在适当的节点中,这减少了在运行时之前和期间的计算。

[0012] 根据本发明的不同实施例,上述实施例中的任一者可由计算机可读存储媒体实施。计算机可读存储介质包含计算机程序代码,当由一个或多个计算机处理器的操作执行时,所述计算机程序代码执行操作。在实施例中,所执行的操作可以对应于上述方法和实施例的任何组合。

[0013] 根据本公开的又一不同实施例,上述实施例中的任一个可由系统实现。该系统包括一个或多个计算机处理器和包含程序的存储器,当由一个或多个计算机处理器执行时,该程序执行操作。在实施例中,所执行的操作可以对应于上述方法和实施例的任何组合。

## 附图说明

[0014] 图1示出了根据本文公开的一个实施例的被配置为放置知识库的数据并路由本体查询的架构。

[0015] 图2示出了根据本文公开的一个实施例的用于处理和路由本体查询的工作流。

[0016] 图3A和图3B描绘了根据本文公开的一个实施例的可以用于评估和路由查询的示例本体。

[0017] 图4A和图4B示出了根据本文公开的一个实施例的用于解析和路由示例本体查询的工作流。

[0018] 图5是示出根据本文公开的一个实施例的被配置为路由本体查询的查询处理协调器的框图。

[0019] 图6是示出根据本文公开的一个实施例的用于处理和路由本体查询的方法的流程图。

[0020] 图7是示出根据本文公开的一个实施例的用于处理本体查询以高效地路由查询块的方法的流程图。

[0021] 图8是示出根据本文公开的一个实施例的用于评估潜在查询路由计划以便高效地路由本体查询的方法的流程图。

[0022] 图9是示出根据本文公开的一个实施例的用于路由本体查询的方法的流程图。

[0023] 图10示出了根据本文公开的一个实施例的用于评估工作负载和存储知识库数据的工作流。

[0024] 图11描绘了根据本文公开的一个实施例的用于评估工作负载和存储知识库数据的示例超图。

[0025] 图12是示出根据本文公开的一个实施例的被配置为评估工作负载并存储数据的数据放置协调器的框图。

[0026] 图13是示出根据本文公开的一个实施例的用于评估和概括查询工作负载以通知数据放置决策的方法的流程图。

[0027] 图14是示出根据本文公开的一个实施例的用于对本体工作负载建模以通知数据放置决策的方法的流程图。

[0028] 图15是示出根据本文公开的一个实施例的用于评估超图以驱动数据放置决策的方法的流程图。

[0029] 图16是示出根据本文公开的一个实施例的用于评估超图以驱动数据放置决策的方法的流程图。

[0030] 图17是示出根据本文公开的一个实施例的用于将本体概念映射到存储节点的方法的流程图。

## 具体实施方式

[0031] 本公开的实施例提供了一种本体驱动的架构,其使用多个数据存储或节点来提供对各种查询类型的支持,每个数据存储或节点可以具有不同的能力。有利地,本公开的实施例使得能够基于现有本体将查询高效地处理和路由到多样化数据存储库(store),这改进了系统的延迟并且减少了标识和返回相关数据所需的计算资源。在实施例中,本文描述的技术可以用于支持各种查询应用,包括自然语言和会话接口。在此描述的系统可以经由抽象本体查询语言(OQL)提供对底层后端存储库的透明访问,该OQL允许用户针对域本体表达他们的信息需求。

[0032] 在一个实施例中,为了为不同的查询类型提供改进的性能,系统首先通过根据其能力将数据的子集放置在适当的后端存储库中,来优化将KB数据放置到不同的存储库中。在一些实施例中,在运行时,系统解析和编译OQL查询,并将其翻译成不同后端数据节点的

查询语言和/或API。在一个实施例中,本文描述的系统还采用查询协调器,该查询协调器根据相关数据的放置将查询路由到单个或多个后端存储库。这种有效的路由减少了生成结果并将结果返回给请求实体所需的延迟。

[0033] 在实施例中,要查询的KB数据可以包括任何信息,包括结构化、非结构化和/或半结构化数据。为了支持深度域专门化,本文公开的实施例利用域本体。值得注意的是,在一个实施例中,域本体仅在元数据级别定义实体及其关系,而不提供实例级别信息。即,在一个实施例中,域本体提供域模式,而实例级别数据存储在各种后端数据存储库(也称为数据站点和数据节点)中。在实施例中,数据节点可以包括任何存储库架构,包括(一个或多个)关系数据库、(一个或多个)倒排索引文档存储库、(一个或多个)JavaScript对象符号(JSON)存储库、(一个或多个)图数据库等。

[0034] 在一些实施例中,所述系统可以在任何后端中移动、存储以及索引KB数据,后端提供了支持查询类型所需的能力。换句话说,系统不需要跨现有操作数据存储联合数据。这与联合数据库和基于中介器的方法明显不同。在一些实施例中,系统执行基于能力的数据放置步骤,其中符合给定本体的知识库数据被存储在各种后端资源中。在至少一个实施例中,多存储库架构被配置为最小化不同数据存储库之间的数据移动,因为数据移动不仅导致数据传送成本,而且导致昂贵的数据转换成本。最小化数据移动成本的一个解决方案是在所有后端节点中复制数据,从而确保查询可以由单个存储库来应答而无需任何数据移动。然而,这也需要浪费的复制,并且使使用多存储库架构来平衡多个数据存储器的不同能力的目的失效。在本发明的实施例中,KB数据可以被存储在任意数量的后端资源中,并且系统智能地路由查询,以选择(一个或多个)最合适的数据存储库,并最小化数据移动成本。

[0035] 在本文所述的一些实施例中,系统架构提供适当的抽象以查询数据,而不知道数据如何在多个数据存储库中存储和索引。在一个实施例中,为了实现该目的,引入本体查询语言(OQL)。OQL针对企业的域本体来表达。系统的用户只需知道定义实体及其关系的域本体。在实施例中,系统理解各种本体概念到后端数据存储库的映射以及它们的对应模式,并且提供从OQL到底层系统的目标查询语言的查询翻译器。

[0036] 在实施例中,对于任何给定查询,可以涉及一个或多个数据节点。本公开的实施例提供了用于标识所涉及的存储库并且生成适当的子查询以计算最终查询结果的技术。在一些实施例中,系统不使用中介器方法,其中全局查询被分成多个子查询并且最终结果在中介器中被组装。相反,数据存储库中的一个或多个被用作中介器,并且(一个或多个)后端数据存储库最终化查询回答。例如,在一个实施例中,关系数据库被用来最终化查询响应,因为关系存储库很可能能够快速完成连接操作(join operation)。

[0037] 图1示出了根据本文公开的一个实施例的被配置为路由本体查询的架构100。在所示的实施例中,查询处理协调器115接收使用OQL构造的本体查询105,其具有对应的本体模式(Ontology Schema)110。本体查询105由查询处理协调器115处理以标识并返回相关数据。在实施例中,查询处理协调器115将查询路由到包括各种数据节点130A-N的后端125。在所示的实施例中,该路由至少部分地基于概念映射140和能力135的集合来执行。

[0038] 在所示的实施例中,基于OQL来格式化本体查询105,该OQL用于表示对本体模式110中的概念和关系的集合进行操作的查询。在一个实施例中,OQL可以表达包括聚合、联并(union)、嵌套子查询等的查询。在一些实施例中,OQL还可以表达全文和字段搜索谓词,以

及路径查询。OQL查询通常包括单个查询块或多个OQL查询块的联并组成,其中每个OQL查询块包括SELECT子句和FROM子句组成。在一些实施例中,OQL块还可以包含WHERE子句、GROUP BY子句、ORDER BY子句、FETCH FIRST子句和/或HAVING子句。在实施例中,OQL查询对由FROM子句中所引用的概念的笛卡尔积所构造的元组的集合进行操作。

[0039] 在实施例中,本体模式110在语义级别上以KB描述实体及其关系,而不管数据实际上是如何存储在后端资源中的。在一个实施例中,本体模式110描述与域相关的实体、潜在与各种实体相关联的属性以及不同实体之间的潜在关系。本体模式110可以提供丰富和表达性数据模型,其捕获实体之间的各种真实世界关系,诸如泛函、继承、联并等。在一些实施方式中,本体模式110不包括实例数据。也就是说,本体模式110定义实体和关系,但是不包括与KB中的特定实体或关系相关的数据。例如,本体模式110可以定义“Company(公司)”实体可以具有诸如“名称(name)”和“地址(address)”的若干属性,但是本体模式110不包括用于特定实例的数据,诸如具有名称“Main Street Grocer”和地址“123Main Street”的公司。相反,这个实例数据在后端125中单独维护。

[0040] 在实施例中,概念映射140指示本体模式110所表示的逻辑模式与后端125中的底层数据节点130的物理模式之间的对应关系。例如,假设本体模式110被定义为$\mathcal{O}(C,R,P)$,其中C＝{$c_n$|1≤n≤N}表示概念(也称为实体)的集合,R＝{$r_k$|1≤k≤K}表示概念/实体之间的关系的集合,并且P＝{$p_m$|1≤m≤M}是数据属性的集合。在一个实施例中,每个关系是在两个或多个概念之间,而每个数据属性对应于概念的特性。在所示实施例中,概念映射140指示(在本体模式110中定义的)概念、关系和属性与底层数据节点130之间的映射。也就是说,在一个这样的实施例中,概念映射140指示存储给定概念(例如,实体)、关系和/或属性的每个数据节点130。例如,概念映射140可指示“Company”实体的实例存储在数据节点130A和130M中,而与“Company”实体之间的特定类型的关系有关的数据存储在数据节点130B中。

[0041] 在一个实施例中,能力135指示由每个数据节点130提供的操作和能力。在一些实施例中,数据节点130的能力被表达为列举了可由数据存储库处理/回答的所有可能的查询(视图定义)的视图。这种方法虽然灵活,但由于视图定义的数量可能非常大,因而不是可缩放的,潜在地导致使用无限数量的视图进行查询重写的问题。为了提高可缩放性,本公开的一些实施例根据后端数据节点130支持的操作(例如,连接(join)、分组、聚合(aggregation)、模糊(fuzzy)文本匹配、路径表达式等)而非列举数据存储库可以回答的所有可能查询来描述它们的能力。另外,本公开的至少一个实施例通过利用表达任何相关联的限制的机制来提供每个所支持的操作的更细粒度的描述。例如,在一个这样的实施例中,类型MAX的聚合函数可能仅在数字类型上得到支持。

[0042] 因此,在一个实施例中,能力135为每个给定数据节点130指示节点可以完成的操作的集合(以及在一些实施例中,对该能力的相关限制)。在实施例中,查询处理协调器115评估能力135和概念映射140,以选择所接收的本体查询105应当被路由到的一个或多个数据节点130。为此,在一个实施例中,查询处理协调器115标识包含所需数据(例如,基于概念映射140)、能够完成(一个或多个)所需操作(例如,基于能力135)或两者的数据节点130。查询处理协调器115然后可为每个所选择的数据节点130生成子查询。

[0043] 在实施例中,查询处理协调器115使用翻译器的集合120A-N来按需翻译子查询。在

所示实施例中,每种类型的数据节点130具有对应的翻译器120。在一些实施例中,每个翻译器120A-N接收OQL查询的全部或部分,并且以对应的数据节点130A-N的语言和/或语法生成等效的查询。例如,翻译器120A可以生成用于包含在数据节点130A中的关系数据库的SQL查询,而翻译器120B输出用于包含在数据节点130B中的图存储库的图查询。

[0044] 在一个实施例中,翻译器120依赖于将域本体中表示的概念和关系映射到目标物理模式中的适当模式对象的模式映射。例如,对于关系型后端数据节点130,模式映射可以提供(1)本体中的概念和关系模式中的表之间的对应关系;(2)本体中的概念和物理模式中的表列的数据属性或特质;和/或(3)本体中的概念和对应于数据库中的概念的表之间的主键-外键(primary key-foreign key)约束之间的关系。类似地,对于JSON文档存储库,模式映射可以将本体中表示的概念、数据属性和关系映射到JSON文档中适当的字段路径。

[0045] 在一些实施例中,翻译器120还处理可以在本体中表示的特殊概念和关系,诸如通常表示本体概念之间的连接条件的概念之间的联并、继承和遍历。取决于物理数据布局,这些被翻译成由后端125数据节点130支持的适当操作。

[0046] 在实施例中,每个数据节点130接收子查询,并且生成用于查询处理协调器115的响应。如果数据节点130在本地具有所有必要的数据,并且能够完成(一个或多个)所指示的操作,则该节点执行查询并返回结果。在一些实施例中,子查询可以指示数据节点130应当向一个或多个其他数据节点130发送查询,以检索完成该查询所需的数据。例如,在一些实施例中,优选地,由关系数据节点执行连接操作,因为关系系统倾向于具有连接操作的低延迟。另外,一些操作仅在某些节点上是可能的。例如,假设数据节点130A是可以完成连接操作的唯一节点,但是要连接的数据仅存在于数据节点130B上。在一个实施例中,数据节点130A接收指示其连接相关数据的子查询,以及一个或多个要被转发到数据节点130B以检索数据的其它子查询。

[0047] 也就是说,在一个这样的实施例中,查询处理协调器115准备用于数据节点130B的翻译的子查询,并将它们发送到数据节点130A。这允许数据节点130A简单地将它们转发到数据节点130B。然后,数据节点130B将数据返回给数据节点130A,其完成操作并将结果返回给查询处理协调器115。在另一个实施例中,查询处理协调器115可以将子查询发送到数据节点130B,并将所得到的数据转发到数据节点130A以进行处理。在又一实施例中,查询处理协调器115在本地执行连接。

[0048] 在所示的实施例中,架构100还包括数据放置协调器150,其确定哪个(哪些)数据节点130A-N应当被用于将数据存储在知识库中。如图所示,数据放置协调器150接收数据节点130的能力155的类似指示,以及查询工作负载160的指示。在一个实施例中,查询工作负载160指示针对知识库的平均或预期查询集。例如,在一个实施例中,查询工作负载160是通过观察用户随时间与知识库的交互来生成的。系统然后可以聚合这些交互以确定系统的平均、预期或典型工作负载。在一些实施例中,查询工作负载160包括对一起查询哪个(哪些)概念、对每个概念应用哪个(哪些)操作等的指示。

[0049] 在实施例中,基于节点能力155和已知的查询工作负载160,数据放置协调器150为本体模式110中的每个概念/实体确定哪个(哪些)数据节点130应当存储该概念的实例级别数据。例如,基于给定数据节点130A能够执行的操作并且进一步基于查询工作负载160,数据放置协调器150可以确定数据节点130A应当存储“company”实体和“public metric”实体

的所有数据,因为它们经常被一起查询。类似地,数据放置协调器150可以基于确定查询频繁地包括对“document(文档)”数据执行模糊匹配操作,并且数据节点130B支持模糊匹配,来确定将“document”概念的实例放置在数据节点130B中。

[0050] 这种智能数据放置可以减少运行时期间的后续数据传送。在实施例中,可以在开始时(例如,提供初始放置)和/或在运行时间期间周期性地(例如,基于查询工作负载160如何随着时间的过去而发展来重新形成放置决策)利用数据放置协调器150。尽管为了概念清楚而被描绘为分立的组件,但是在实施例中,查询处理协调器115和数据放置协调器150的操作可以被组合或分布在任意数量的组件上。

[0051] 如图所示,数据放置协调器150将放置决策的集合165输出到后端125,和/或输出到一个或多个中间服务,诸如提取、变换和加载(ETL)服务。然后,基于这些选择,将知识库中的实例级别数据存储在适当的数据节点130中。概念映射140反映数据的当前放置。在实施例中,如果数据放置协调器150被用来基于更新的查询工作负载160来修改数据放置,则概念映射140被类似地更新。在下文中,图2到图9更详细地讨论查询处理协调器115,并且假设数据已经被放置。参考图10到图17更详细地讨论了数据放置协调器150和各种技术以确保有效的数据放置。

[0052] 图2示出了根据本文公开的一个实施例的用于处理和路由本体查询的工作流200。如图所示,工作流200在接收到OQL查询205时开始。OQL查询205被提供给OQL解析器210,其解析查询以确定查询的含义。如箭头215所示,该解析的OQL查询然后被提供给查询图模型(QGM)构造器220。在一实施例中,QGM构造器200以查询图模型的形式生成查询的逻辑表示。这降低了查询编译和优化的复杂性。下面参考图4A和图4B更详细地讨论示例查询图模型。在实施例中,QGM使用诸如SELECT、GROUP BY、SETOP等操作符盒(operator box)来捕获查询中的数据流和依赖性。在实施例中,盒内的操作可以在它们之间自由地重新排序,但是当生成查询执行计划时遵守盒边界。也就是说,查询执行计划必须遵循查询盒的顺序。

[0053] 在一个实施例中,使用量词来表示查询块之间的数据流。这种格式允许系统推理查询等同性,并应用重写优化。在实施例中,查询的QGM表示使得系统能够集中于在查询执行期间优化不同数据存储库之间的数据流,而实际物理执行计划的选择被推迟到负责执行查询盒或片段的底层数据节点130。在实施例中,使用QGM 225来产生优化的多存储执行计划,其最小化跨不同后端的数据移动和变换。

[0054] 在一些实施例中,QGM 225包括在底部的量词的集合,其向查询提供输入概念集合,并且QGM中的每个查询块具有头部和主体。每个块的主体包括描述要对输入概念集合(诸如连接)执行的集合操作的谓词的集合,并且头部表达式描述应当如何计算结果概念的输出属性。换句话说,在实施例中,每个盒的主体包含谓词(也称为操作)的集合,每个谓词将被应用于该盒的输入量词。在一些实施例中,引用单个量词的谓词/操作被分类为本地谓词,而引用多个量词的谓词/操作表达连接谓词。

[0055] 在所示工作流200中,将QGM 225传递到继续查询路由过程的操作符放置组件230。如图所示,操作符放置组件230还接收概念映射235和节点能力240的集合,并且基于能力和概念映射来注释QGM 225中的查询块。在一个实施例中,操作符放置组件230将QGM从底部移到顶部,并且用可以执行操作的可能存储库的集合来注释查询中的每个操作。在一些实施例中,对于本地谓词和连接谓词不同地生成注释。回想在一些实施例中,与单个量词相关联

的所有头部表达式以与本地谓词相同的方式来处理。

[0056] 在一个实施例中,如果查询块包括本地谓词/操作(例如,单个量词),则操作符放置组件230确定量词是否是基本概念。如果是,则用(i)包含概念和(ii)具有执行谓词的能力的数据节点130的指示来注释谓词。在实施例中,如果量词来自另一QGM块(即,它由另一块计算),则操作符放置组件230用数据存储库的集合注释谓词,该数据存储库的集合(i)完成该量词的QGM盒,以及(ii)能够执行谓词。在一些实施例中,如果包含到该谓词的输入的数据节点130中没有一个也具有执行谓词的能力,则操作符放置组件230用能够执行谓词的存储库集合来注释它。

[0057] 例如,如果谓词包含模糊搜索,但数据仅存储在关系型后端数据节点130中,则操作符放置组件230可以用能够计算该模糊搜索的文档存储库来注释谓词,即使数据没有存储在那里。注意,在查询执行期间,在这种情况下将需要移动数据。

[0058] 在一些实施例中,如果查询块包括连接谓词/操作,则操作符放置组件230检查连接类型和连接谓词。在实施例中,每个连接谓词与两个或更多个量词相关联:每个连接输入一个。在实施例中,操作符放置组件230基于所标识的量词是计算的输入还是基本概念以及量词来自何处(例如,它是计算的和/或本地存储的,还是将从另一节点接收的)来标识这些量词的进行。

[0059] 在所有量词覆盖基本概念的情况下,操作符放置组件230评估任何基本概念所驻留的数据节点130的集合,并且为每个这样的存储库确定该存储库是否支持连接操作。操作符放置组件230然后用包含一个或多个所需概念并且支持连接操作的类型的存储库的指示来注释连接操作。在一些实施例中,如果存储库中的一个是关系节点,则操作符放置组件230用该关系数据节点130而不是其余节点的指示来注释操作。

[0060] 在一些实施例中,如果两个量词都由其他QGM查询块产生,则操作符放置组件230用支持连接操作类型(在一些实施例中,仅包括关系存储库)的数据节点130注释连接操作。在实施例中,如果产生量词的数据存储库能够执行连接(JOIN),则操作符放置组件230还用它们的指示来注释操作。

[0061] 在另一实施例中,如果量词之一是基本概念,而另一个是从另一QGM查询块计算的,则当两个量词都由其它QGM块计算时,连接操作放置决策与上述讨论类似。在实施例中,操作符放置组件230用支持连接类型并且对于连接输入至少之一是本地的数据存储库的集合注释连接操作。

[0062] 如图所示,该注释的QGM 245然后被传递到块放置组件250。在一个实施例中,块放置组件250利用由操作符放置组件230生成的注释来确定查询中的查询盒的可能放置选项。

[0063] 在一个实施例中,确定用于“select(选择)”查询盒的放置选项被建模为确定最小集合覆盖的问题。即,块放置组件250确定满足在给定选择盒内放置所有操作符所需的最小数量的数据节点130。在一些实施例中,块放置组件250利用贪婪启发式算法来执行谓词-存储库分组,其保证每个谓词被放置在单个存储库中,并且查询盒所跨越的存储库的总数被最小化。一旦每个谓词用它将被放置的适当存储库来注释,块放置组件250就确定查询块是否需要被拆分。

[0064] 在实施例中,如果选择查询盒中的所有谓词被放置在相同数据节点130中,则查询盒被注释以被放置在该存储库中。相反,如果选择查询盒中的谓词被放置在多于一个数据

节点130中(指示查询盒需要由多个存储库处理),则块放置组件250确定该块必须被拆分成多个查询盒。在这种情况下,块放置组件250然后对块进行划分,使得每个结果(子)块包括被分配给单个数据节点的谓词。

[0065] 在一些实施例中,对于“分组依据(group by)”查询盒,块放置组件250基于支持聚合操作的类型的存储库以及处理馈送分组依据盒(group by box)的查询盒的先前查询块的存储库来确定放置。在实施例中,如果馈送这些输入的存储库也支持“group by”和“聚合(aggregation)”函数,则块放置组件250将所选择的“group by”盒放置在同一存储库中。然而,如果该存储库不支持这些操作,则块放置组件250用可以执行分组依据操作(group by operation)的数据节点130的列表来注释该盒。注意,这种放置将需要从馈送存储库到可以处理该盒的存储库的数据移动。

[0066] 一旦块放置组件250已经用每个查询块的(一个或多个)所有可能放置(例如,可以执行块的所有数据节点130)注释了每个查询块,则成本组件255确定放置的每个组合的成本。在一个实施例中,用于查询路由的成本模型仅关注于数据传送和数据转换成本,以便在替代查询执行计划之间进行选择,因为数据移动和变换的成本可能支配多存储库环境中的任何执行计划的总成本。因此,在至少一些实施例中,给定执行计划执行的成本是通过聚合查询执行计划中的每个源和目标数据存储库对的数据移动的成本来确定的。注意,在一些实施例中,诸如连接和图操作的不同操作可以在不同数据节点130上以非常不同的性能来执行。例如,尽管连接操作可能被各种数据存储库(诸如JSON存储库)支持,但是关系数据库通常为连接操作提供最佳性能。

[0067] 在一些实施例中,成本模型还考虑不同存储库的操作的执行成本的这种变化。在其他实施例中,成本模型不包括这些因素。在一个这样的实施例中,使用表达底层存储库的能力的声明性机制来处理这样的情况。例如,为了避免在JSON存储库上执行连接操作,系统可以在能力描述中完全屏蔽JSON存储库的连接能力,或者添加限制以限制其对特定数据类型的适用性。

[0068] 在实施例中,成本组件255列举所有查询块上的所有可能放置组合,并且通过将某些查询盒分组在一起成为组来生成每个这样的组合的执行计划。在一个实施例中,查询计划中的每个组包括查询盒,其既是(i)QGM中的结果(例如,由单跳分开)又是(ii)可由同一数据节点130处理。然后,成本组件255基于QGM流联结这些块的组,定义表示数据节点130之间的数据流的边。在实施例中,成本组件255还基于所生成的计划中的组之间的边,为每个执行计划生成数据移动描述符的集合。

[0069] 在一些实施例中,使用用于每个计划的这些移动描述符,成本组件255利用上述数据移动成本模型来找到用于每个执行计划的成本,并且挑选具有最小成本的一个。例如,在一个实施例中,成本组件255为每个可能的执行计划确定对应的移动描述符中的每一个的成本。这可以包括延迟、计算成本等。然后,可以在每个执行计划内聚合这些值,以便确定哪个(哪些)执行计划具有最低成本。如图所示,该最小成本计划然后被选择并用于生成数据节点130的一个或多个翻译的查询260。

[0070] 图3A和图3B描绘了根据本文公开的一个实施例的可以用于评估和路由查询的示例本体300。在所示实施例中,使用椭圆来描绘每个概念305A-I,而使用圆角矩形来描绘每个属性310A-N。此外,箭头用于描绘概念305和属性310之间的关联,而粗体箭头描绘概念

305之间的关系。另外,基于关系的类型来标记每个关系箭头。例如,如图所示,“Company”概念305B是“公开公司(PublicCompany)”概念305D的子类。此外,公司概念305B与若干属性310相关联,包括“名称(name)”属性310E和标识符属性310D,两者都是字符串(string)。

[0071] 如上所述,在实施例中,知识库中的数据符合本体300。换句话说,本体300定义知识库中的概念和实体,以及实体和与每个概念相关联的属性之间的关系。值得注意的是,本体300不包括任何实例数据(例如,关于特定公司的数据),而是定义数据的结构。在一些实施例中,概念305、属性310和/或关系可分布在任意数量的数据节点130上。在一个实施例中,至少部分基于数据的类型将数据放置在数据节点130中。

[0072] 在一个这样的实施例中,如果概念映射指示给定概念305存储在给定数据节点130中,则对应于该概念305的所有实例数据存储在数据节点130中。例如,如果映射指示数据节点130A包括“PublicMetric”概念305F,则查询处理协调器115可从数据节点130A检索关于“PublicMetric”的任何实例(例如,对于任何公司的任何度量)的数据。因此,如果接收到的查询将需要访问“PublicMetric”数据,则查询处理协调器115将把查询的至少一部分路由到数据节点130A(或路由到也服务于概念305F的另一节点)。

[0073] 图4A和图4B示出了根据本文公开的一个实施例的用于解析和路由示例本体查询的工作流400。在所示实施例中,接收并评估输入405以从数据节点130检索相关数据。在所示实施例中,输入405是自然语言文本(例如,来自用户)。然而,在各种实施例中,输入405可以包括查询或其他数据。此外,输入405可以从任何数量的源接收,包括自动化应用、面向用户的应用、直接来自用户等。在一些实施例中,输入405作为聊天机器人或允许用户搜索和探索知识库的其他交互式应用的一部分被接收。

[0074] 在所示的示例中,输入405是短语“Show me the total revenue of all companies that filed technology patents in the last 5years(请告诉我过去5年中申请技术专利的所有公司的总收入)”。在所示工作流400中,使用诸如语义分析、关键字搜索、情感分析、意图分析等的一种或多种自然语言处理(NLP)技术来解析和评估输入405。这使得系统能够基于输入405生成OQL查询410。当然,在一些实施例中,输入405本身是OQL查询。

[0075] 如图所示,针对输入405的相应OQL查询410包括“select(选择)”操作和“group by(分组按照)”操作,以及关于相关的表或概念的指示,以及指示对查询的限制的“where”子句。然后,该OQL查询410被查询处理协调器115解析以生成OGM 435A,其是查询的逻辑表示,并且包括多个查询块。在所示实施例中,OGM包括“select”查询块415B和“group by”查询块415A。

[0076] 如图所示,QGM 435A包括位于底部的量词的集合430A-E,其向查询提供输入概念集合。在所示的实施例中,这些包括“PublicMetricData”、“PublicMetric”、“PublicCompany”、“Document”和“CompanyInfo”。在实施例中,QGM 435A中的每个查询块415包括头部420和主体425。主体425一般描述要对输入概念集合(诸如连接)执行的集操作,而头部420表达式描述应当如何计算结果概念的输出属性。例如,在所示的实施例中,“select”查询块415B的头部420B指定输出属性“oPMD.value”、“oPMD.year_calendar”和“oCl.id”。主体425B包含要被应用于输入量词430A-E的谓词的集合。如上所述,在实施例中,引用单个量词的谓词是本地谓词,而引用多个量词的谓词表达连接谓词。

[0077] 如图4B所示,因为查询处理协调器115认识到“select”查询块415B包括必须由不同的数据节点130执行的谓词。具体地,尽管查询块415B中的大多数谓词可以由关系数据存储库执行,但是查询处理协调器115已经确定谓词“oD->companylnfo＝oCI”和“oD.selfMATCH('Tech Patent Filed')”对应于使用关系数据库节点不支持的弹性搜索执行的操作。因此,查询处理协调器115通过将这些谓词分离到新的查询块415C中来拆分查询块415B。如图所示,该查询块415C充当查询块415B的输入。

[0078] 在所示实施例中,通过拆分查询块415B,查询处理协调器115已经确保了每个块可以完全在单个数据节点130内执行。例如,查询块415A和415B都可以在关系数据节点130中执行,而查询块415C将由支持弹性搜索的数据节点130执行。因此,在实施例中,查询处理协调器115标识或生成与查询块415A和415B相对应的子查询,将该子查询翻译成关系节点的适当语言和/或格式,并且将翻译的子查询发送到关系节点。

[0079] 在所示的实施例中,查询处理协调器115将进一步标识或生成子查询以完成查询块415C,并将其翻译成弹性搜索节点所支持的语言和/或格式。在一个实施例中,查询处理协调器115另外将该子查询发送到关系数据存储库,其将充当聚合器(aggregator)。然后,关系节点可以将子查询转发到弹性节点,并且使用由弹性节点返回的结果来完成其自己的子查询。

[0080] 图5是示出根据本文公开的一个实施例的被配置为路由本体查询的查询处理协调器115的框图。尽管被描绘为物理设备,但是在实施例中,查询处理协调器115可以使用(一个或多个)虚拟设备和/或跨多个设备(例如,在云环境中)来实现。如图所示,查询处理协调器115包括处理器510、存储器515、存储装置520、网络接口525以及一个或多个I/O接口530。在所示的实施例中,处理器510检索并执行存储在存储器515中的编程指令,以及存储和检索驻留在存储装置520中的应用数据。处理器510一般表示单个CPU和/或GPU、多个CPU和/或GPU、具有多个处理核的单个CPU和/或GPU等。存储器515通常被包括以表示随机存取存储器。存储装置520可以是盘驱动器、基于闪存的存储设备等的任何组合,并且可以包括固定和/或可移动存储设备,诸如固定盘驱动器、可移动存储器卡、高速缓存、光存储装置、网络附接存储装置(NAS)或存储区域网络(SAN)。

[0081] 在一些实施例中,输入和输出设备(诸如键盘、监视器等)经由(一个或多个)I/O接口530联结。此外,经由网络接口525,查询处理协调器115可以与一个或多个其他设备和组件通信地耦合(例如,经由网络580,其可以包括因特网、(一个或多个)本地网络等)。如图所示,处理器510、存储器515、存储装置520、(一个或多个)网络接口525和(一个或多个)I/O接口530通过一个或多个总线575通信地耦合。另外,查询处理协调器115经由网络580通信地耦合到多个数据节点130A-N。当然,在实施例中,数据节点130A-N可直接联结到查询处理协调器115、经由本地网络可访问、集成到查询处理协调器115中等。尽管未包括在所示的实施例中,但是在一些实施例中,查询处理协调器115还与数据放置协调器150通信地耦合。

[0082] 在所示实施例中,存储装置520包括本体560、能力数据565和数据映射570。尽管被描绘为驻留在存储装置520中,但是在实施例中,本体560、能力数据565和数据映射570可以以任何合适的位置和方式存储。在实施例中,如上所述,本体560指示与查询处理协调器115在其中操作的域相关的实体或概念,以及每个概念/实体的潜在属性和实体/概念之间的关系。在至少一个实施例中,本体560不包括实例级别数据。相反,KB中的实际数据被存储在数

据节点130A-N中。

[0083] 在实施例中,如上所述,能力数据565指示每个数据节点130A-N的能力,这可以包括例如关于每个数据节点130A-N支持哪个(哪些)操作的指示,以及关于该支持的任何对应的限制。例如,能力数据565可指示数据节点130A支持“连接”操作,但仅支持整数数据类型。在实施例中,如上所述,数据映射570指示存储本体560中定义的每个实体/概念的实例数据的(一个或多个)数据节点130。例如,数据映射570可指示“Company”概念的所有实例都存储在数据节点130A和130B中,而“Document”概念的所有实例都存储在数据节点130B和130N中。

[0084] 在所示的实施例中,存储器515包括查询应用535。尽管被描绘为驻留在存储器515中的软件,但是在实施例中,查询应用535可以使用硬件、软件或硬件和软件的组合来实现。如图所示,查询应用535包括解析组件540、路由组件545和(一个或多个)翻译器的集合120。尽管为了概念清楚而被描绘为离散组件,但在实施例中,解析组件540、路由组件545和(一个或多个)翻译器120的操作可被组合或分布在任何数量的组件上。

[0085] 在实施例中,解析组件540接收OQL查询,并解析它们以确定它们的含义,并生成查询的逻辑表示(诸如QGM)。如上所述,在一些实施例中,逻辑表示包括查询块的集合,其中每个查询块指定定义要对块的输入量词执行的操作的一个或多个谓词。在实施例中,这些量词可以是存储在一个或多个数据节点130中的(一个或多个)基本概念和/或由另一查询块计算的数据。在一个实施例中,该逻辑表示使得查询应用535能够理解和推理查询的结构以及数据如何在块之间流动。这使得路由组件545能够有效地路由查询(或来自查询的子查询)。

[0086] 在所示实施例中,路由组件545从解析组件540接收逻辑表示(例如,查询图模型),并对其进行评估以标识查询应当被路由到的一个或多个数据节点130。在一些实施例中,这包括分析逻辑表示中的每个查询块中包括的谓词。在一个实施例中,路由组件545基于可以完成谓词的(一个或多个)数据节点130来标记或注释给定查询块中的每个谓词。在一些实施例中,这包括包含(一个或多个)相关量词和/或能够执行(一个或多个)所指示的操作的(一个或多个)数据节点130。此外,在至少一个实施例中,一旦块中的每个谓词已经被处理,路由组件545可以评估注释以确定块的放置位置。也就是说,路由组件545确定查询块是否可以被放置于单个数据节点130中。如果是,则在一个实施例中,路由组件545将该块分配给该存储库。此外,在实施例中,如果该块不能被单个数据节点130处理,则路由组件545将该块拆分成两个或更多个查询块,并且针对每个查询块重复该路由过程。

[0087] 在实施例中,一旦每个查询块已经被分配给单个数据节点130,(一个或多个)翻译器120就生成一个或多个对应的翻译的查询。在实施例中,每个翻译器120对应于特定的数据节点130架构,并且被配置为将OQL查询(或子查询)翻译为用于对应的数据存储装置架构的适当查询。例如,在一个这样的实施例中,系统可以包括将OQL翻译为关系存储库(例如SQL)的查询的第一翻译器120、将OQL翻译为JSON查询的第二翻译器120、翻译为文档搜索的查询的第三翻译器120、以及将OQL翻译为图查询的第四翻译器120。

[0088] 在实施例中,基于将执行查询或子查询的数据节点130,将查询(或子查询)路由到适当的翻译器120。在一些实施例中,每个查询(或子查询)然后被直接发送到那个数据节点130(例如,经由应用编程接口或API)。在一些实施例中,如果完成查询将需要在数据节点

130之间移动数据(例如,以连接来自两个或更多个存储库的数据),则查询应用535可以基于所生成的逻辑表示来确定数据流,并且适当地发送查询。即,查询应用535使用QGM来确定哪个(哪些)存储库将需要从其他(哪些)存储库接收数据,并将所需的查询发送到这些连接存储库。例如,如果数据节点130A要从数据节点130N接收数据并完成对其的一个或多个操作,则查询应用535可向数据节点130A发送第一查询以从该节点检索数据,以及为数据节点130N配置的第二查询。然后,数据节点130A可以自己使用所提供的查询来查询数据节点130N。

[0089] 图6是示出根据本文公开的一个实施例的用于处理和路由本体查询的方法600的流程图。方法600开始于框605,其中查询处理协调器115接收本体查询以便针对知识库执行。在实施例中,相对于预定义的域本体来表达本体查询,其中本体指定与域相关的概念、属性和关系。在框610,查询处理协调器115解析所接收的查询并生成其逻辑表示。在一些实施例中,查询处理协调器115生成查询的查询图模式表示。在实施例中,逻辑表示包括表示查询所暗示的(一个或多个)基本概念、要对数据执行的(一个或多个)操作以及数据流的一个或多个查询块。

[0090] 方法600然后继续至框615,其中查询处理协调器115基于预定义的概念映射和/或数据存储能力将逻辑表示中的每个查询块映射到一个或多个数据存储库。例如,在一个实施例中,查询处理协调器115为每个查询块确定什么概念是相关的(例如,查询块将访问什么数据)。使用概念映射,查询处理协调器115然后可标识能够提供所需数据的(一个或多个)数据存储库。另外,在一个实施例中,查询处理协调器115为每个查询块确定将需要什么操作。使用预定义的存储能力,查询处理协调器115然后可标识哪个(些)数据节点能够执行(一个或多个)所需的操作。查询处理协调器115然后将(一个或多个)查询块映射到(一个或多个)数据节点,努力最小化存储库之间的数据移动。

[0091] 在框620,查询处理协调器115选择所映射的查询块之一。方法600然后前进至框625,其中查询处理协调器115基于所映射的数据存储库为查询块生成对应的翻译的查询。在一个实施例中,这包括从所接收的查询中标识或生成子查询,以执行所选择的查询块。查询处理协调器115然后确定将执行查询块的数据节点的配置。也就是说,在一个实施例中,查询处理协调器115确定节点被配置为处理的查询的语言和/或格式。在另一实施例中,查询处理协调器115标识对应于所映射的数据存储库的翻译器。查询处理协调器115然后使用该翻译器来生成用于该存储库的适当查询。

[0092] 方法600然后继续到框630,其中查询处理协调器115确定是否存在尚未被处理的至少一个附加查询块。如果是,则方法600返回到框620。否则,方法600继续到框635,其中查询处理协调器115将一个或多个翻译的查询发送到一个或多个数据节点。在一个实施例中,查询处理协调器115将每个查询发送到对应的节点。在另一实施例中,查询处理协调器115基于逻辑表示中的数据流来发送查询。例如,查询处理协调器115可将多个查询发送到单个节点,使得该节点可将查询转发到(一个或多个)适当的存储以检索数据。然后,节点可以充当中介器以最终化确定结果。

[0093] 有利地,通过采用数据节点之一作为中介器,查询处理协调器115可以降低其计算开销并扩展系统的可缩放性。在框640,查询处理协调器115从充当中介器的数据存储库(或者如果所接收的查询可以在单个后端资源中执行,则从查询被传送到的唯一数据存储库)

接收最终化的查询结果。查询处理协调器115然后将结果返回给请求实体。

[0094] 图7是示出根据本文公开的一个实施例的用于处理本体查询以高效地路由查询块的方法700的流程图。在一些实施例中,方法700提供了用于路由查询块的附加细节。方法700开始于框705,其中查询处理协调器115生成一个或多个查询块以表示所接收的查询,如上所述。在一个实施例中,每个查询块指定指示该块的输入数据的一个或多个量词。这些量词可以对应于基本概念(例如,知识库中的实例数据)和/或中间计算数据(例如,从存储库检索并以某种方式处理和/或变换的数据)。另外,在实施例中,每个查询块指定要对量词执行的操作。

[0095] 在框710,查询处理协调器115选择所生成的块之一。此外,在框715处,查询处理协调器115标识能够满足由选择的块定义的(一个或多个)操作的数据节点的集合。在一个实施例中,查询处理协调器115通过访问定义每个数据存储库可以完成的(一个或多个)操作的集合的预定义的能力定义以及关于该能力的(一个或多个)任何对应的限制来这样做。方法700然后进行到框720,其中查询处理协调器115标识能够满足查询块中列出的(一个或多个)量词的节点的集合。即,对于与基本概念相对应的每个量词,查询处理协调器115标识存储基本概念的节点。

[0096] 在一个实施例中,对于由另一查询块计算的每个量词,查询处理协调器115标识已被分配给该查询块的(一个或多个)数据节点。回想一下,在实施例中,查询处理协调器115通过从QGM的底部向上走而将查询块映射到节点。因此,如果第一块依赖于在第二块中计算的数据,则在查询处理协调器115开始评估第一查询块之前,第二块将必然已经被评估并被分配一个或多个数据节点。

[0097] 方法700然后继续到框725,其中查询处理协调器115确定是否存在能够提供量词和执行操作两者的至少一个数据节点。在一个实施例中,这包括确定在所标识的集合之间是否存在重叠。例如,如果量词都是基本概念,则两个集合之间的重叠指示能够执行所有所需操作并存储所有概念的节点的集合。如果量词包括由其它查询块计算的项目,则重叠集合指示可以执行操作并且将(潜在地)执行(一个或多个)其它块的节点。

[0098] 在实施例中,如果在集合之间存在重叠,则方法700继续到框730,其中查询处理协调器115用重叠中的所标识的节点来注释所选择的查询块。该注释指示指定的节点具有执行查询块的潜力,但不是将块最终分配给(一个或多个)节点。在实施例中,如果任何查询块具有包括在注释中的多个节点,则查询处理协调器115执行成本分析以评估每个替代以试图最小化存储库之间的数据传送,如以下更详细讨论的。方法700然后进行到框740。

[0099] 在所示的实施例中,如果在集合之间不存在重叠,则查询处理协调器115确定不存在可能完成查询块的单个数据节点。在一些实施例中,如果所有操作都可由单个存储库执行,则查询处理协调器115用可能执行块的操作的存储库来注释块。然后可以分配(一个或多个)其它存储库来提供量词。在所示的实施例中,方法700然后继续到框735,其中查询处理协调器115将所选择的查询块拆分成两个或更多个块以创建能够完全在单个节点内执行的查询块。在一个实施例中,拆分所选择的查询块包括标识可由单个存储执行的(一个或多个)量词和/或(一个或多个)操作的(一个或多个)子集。例如,查询处理协调器115可以标识指定的谓词的子集,这些子集共享注释(例如,可以由相同的存储库执行的谓词)。然后,可以通过基于谓词所属的子集将谓词分离到对应的盒中来拆分查询块。例如,如果查询块需

要传统的关系数据库操作以及模糊匹配操作,则查询处理协调器115可确定(一个或多个)模糊匹配应被拆分成单独的盒。这可以使得查询处理协调器115能够将每个拆分块分配到单个数据存储库中(例如,关系操作可以被分配到关系存储库,并且模糊匹配操作可以被分配到能够执行它的不同后端)。

[0100] 在所示的实施例中,这些新生成的查询块被放置于队列中以便被评估,如同现有块一样。方法700然后继续到框740。在框740,查询处理协调器115确定是否存在至少一个未被评估的附加查询块。如果是,则方法700返回到框710以迭代通过这些块。即,查询处理协调器115继续迭代通过查询块,如果必要的话拆分盒,直到所有查询块都用至少一个数据节点注释为止。

[0101] 如果所有查询块都已被注释,则方法700继续到块745,其中查询处理协调器115执行查询。在一些实施例中,这包括评估分配的替代组合以便最小化数据传送,如下文更详细地论述。在一些实施例中,如果查询块中的一个或多个的注释指示单个数据节点,则查询处理协调器115简单地将该块分配给所指示的节点,因为不存在可被考虑的替代。

[0102] 图8是示出根据本文公开的一个实施例的用于评估潜在查询路由计划以便高效地路由本体查询的方法800的流程图。在所示的实施例中,方法800在查询块已经用它们的潜在分配注释(例如,用能够执行整个块并提供所有所需量词的节点)之后开始。方法800开始于框805,其中查询处理协调器115选择节点分配的可能组合中的一个。即,在实施例中,查询处理协调器115基于对应的注释生成块的存储库的所有可能组合。为此,查询处理协调器115可以迭代地选择每个块的不同选项,直到已经生成了所有可能选择。

[0103] 例如,假设第一块被标注为“节点A”和“节点B”,而第二块被标注为“节点A”。在实施例中,查询处理协调器115将确定可能的路由计划包括将第一块和第二块分配给“节点A”,或者将第一块分配给“节点B”并将第二块分配给“节点A”。在框805,查询处理协调器115选择所标识的组合之一以进行评估。方法800然后继续到块810。

[0104] 在框810,查询处理协调器115标识在所选择的计划下将需要的(一个或多个)数据移动。在各实施例中,基于计划中的存储库选择和/或查询图模型来确定移动。继续以上示例,如果查询处理协调器115可确定将两个块分配给“节点A”的计划将不需要移动,因为两个查询块都可被分组到单个节点中。即,因为块在模型中是结果(例如,由单跳直接联结/分离,其间没有其他块)并且被分配相同的节点,所以它们被分组在一起并且不需要数据移动。相反,将第一块分配给“节点B”将需要在“节点A”和“节点B”之间传送数据,因为QGM指示数据从第二块流到第一块,并且这些块被分配给不同的存储库。

[0105] 方法800然后继续到框815,其中查询处理协调器115选择所选择的查询计划所需的所标识的数据传送之一。在框820,查询处理协调器115确定所选择的移动的计算成本。在实施例中,基于预定义的成本模型来做出该确定。在一些实施例中,模型可以针对每个有序数据节点对来指示从第一节点向第二节点传送数据的计算成本。成本可以包括例如实际传送数据和/或按需变换数据以允许目的地节点对其进行操作所引入的延迟、传送/变换的处理时间和/或存储器要求等。

[0106] 在框825,查询处理协调器115确定所选择的组合是否需要任何附加数据传送。如果是,则方法800返回到框815。否则,方法800继续到框830,其中查询处理协调器115通过聚合每个移动的单独成本来计算所选择的计划的总成本。在框835,查询处理协调器115确定

是否存在至少一个尚未被评估的替代计划。如果是,则方法800返回到框805。否则,方法800进行到框840,其中查询处理协调器115基于计划的聚合成本对计划进行排名。在实施例中,查询处理协调器115选择具有最低确定成本的计划。查询处理协调器115然后通过基于最小成本计划的节点分配来翻译和路由子查询来执行计划。

[0107] 图9是示出根据本文公开的一个实施例的用于路由本体查询的方法900的流程图。方法900开始于框905,其中查询处理协调器115接收本体查询。在框910,查询处理协调器115基于本体查询生成一个或多个查询块,每个查询块指示一个或多个操作和表示查询块之间的数据流的一个或多个量词。方法900然后进行到框915,其中查询处理协调器115基于一个或多个量词和一个或多个操作为一个或多个查询块中的每一个标识至少一个数据节点。此外,在框920处,查询处理协调器115基于预定义的成本标准选择所标识的数据节点中的一个或多个。另外,在框925,查询处理协调器115然后将一个或多个子查询发送到所选择的一个或多个数据节点。

[0108] 图10示出了根据本文公开的一个实施例的用于评估工作负载和存储知识库数据的工作流1000。如上所述,企业应用通常需要根据其查询工作负载支持不同的查询类型。为了支持这些不同的查询类型,本公开的实施例利用多个后端存储库,诸如关系数据库、文档存储库、图存储库等。在本公开的实施例中,系统具有在任何后端中移动、存储和索引知识库数据的能力,所述后端提供所支持的查询类型的所需能力。利用组织数据的这种灵活性,跨多个后端存储库的初始数据放置可以在高效查询执行中扮演关键角色。

[0109] 为了以最小的复制开销实现这种高效的运行时执行,本公开的一些实施例提供了离线数据准备和加载阶段,其包括智能数据放置。通常,数据摄取、放置和加载到多个后端存储库涉及一系列操作。最初,从包括结构化、半结构化和非结构化数据的各种不同源摄取域特定知识库的数据。在一些实施例中,来自这些数据源的数据摄取的第一阶段是数据丰富/策展(curation)过程,其包括信息提取、实体解析、数据集成和变换。然后,将由该步骤产生的符合域本体的输出数据馈送到数据放置协调器150,以便为多个后端数据存储库生成适当的数据放置。在一些实施例中,数据已经被管理,并且在应用数据放置协调器150之前,可能已经在响应查询中使用。一旦确定了满意的数据放置,数据加载模块就根据数据放置计划将实例数据放置在适当的数据存储库中。

[0110] 在一些实施例中,通过跨所有数据节点复制整个数据的集合,可以在查询执行期间避免数据移动。然而,这种解决方案导致巨大的复制开销并且显著地增加了存储空间要求。此外,在许多实施例中,不是所有的存储库都提供查询所需的所有必要能力,并且甚至完全复制也不能完全消除数据移动。为了最小化不必要的存储成本和数据移动,本公开的一些实施例提供基于能力的数据放置技术,该算法将数据分配给数据存储库,同时考虑预期工作负载和后端数据存储库的能力(例如,在它们可以对所存储的数据执行的操作方面)。

[0111] 在实施例中,数据放置协调器150在表示存储在知识库中的数据的模式的域本体的概念上的查询操作的级别上对数据放置进行推理。在一些实施例中,协调器基于针对知识库的给定工作负载以及底层存储库的能力来标识本体的不同且潜在重叠子集,并且输出数据的所标识的子集与数据应当存储在其中的目标数据存储库之间的映射。

[0112] 在所示实施例中,工作流1000在接收到OQL查询的集合1005时开始。在一些实施例

中,OQL查询1005是针对知识库的先前提交的查询。例如,在一些实施例中,知识库是用户和应用可以查询和探索的预先存在的数据语料库。在这样的实施例中,OQL查询1005可以对应于用户、应用和其他实体在与知识库交互时先前提交的查询。

[0113] 在实施例中,OQL查询1005一般表示知识库的平均、典型、预期和/或历史工作负载。换句话说,OQL查询1005通常表示知识库在运行时操作期间接收(或预期接收)的查询,并且可以用于标识经常一起查询的概念的集合、通常对每个概念执行的操作等。如图所示,这些代表性OQL查询1005被提供给OQL查询分析器1010,其分析和评估它们以生成概括的工作负载1015。

[0114] 在实施例中,OQL查询分析器1010将OQL查询1005表达为具有针对它们执行的对应的操作的概念和关系的集合。基于此,分析器生成概括的工作负载1015。换句话说,概括的工作负载1015反映所提供的OQL查询1005的概括,这使得能够进行更深入的分析以标识数据中的样式。在一个实施例中,概括的工作负载1015包括基于OQL查询1005生成的概括的查询的集合。值得注意的是,在一些实施例中,概括的查询不是可以针对知识库执行的完整格式化的查询。相反,概括的查询指示可能在运行时期间被一起查询的概念的集群以及可能被应用于概念的每个集群的操作的集合。

[0115] 在一些实施例中,OQL查询分析器1010将针对域本体表达的OQL查询的集合1005作为输入,并且对于每个查询,创建两个集合。在一个这样的实施例中,第一个是查询访问的概念的集合,而第二个是查询对那些概念执行的操作(例如,连接、聚合等)的集合。在至少一个实施例中,为了生成给定OQL查询1005的概括的工作负载1015表示,OQL查询分析器1010将访问相同概念的集合的查询分组到组中,然后创建组合该组中的每个查询的相关联的操作的集合。这将在下面更详细地讨论。

[0116] 在所示实施例中,概括的工作负载1015然后被提供给超图建模器1020,其评估所提供的概括的查询以生成超图1025。在实施例中,超图1025是包括顶点的集合和边(也称为超边)的集合的图,其中每个边可联结到任何数量的顶点。在一些实施例中,超图1025中的每个顶点对应于来自域本体的概念,并且每个超边对应于来自概括的工作负载1015的概括的查询。例如,每个超边跨越由对应的概括的查询指示的概念的集合。在一个实施例中,每个超边还利用由对应的概括的查询指示的操作的集合的指示来注释或者标记。

[0117] 如图所示,超图1025由数据放置组件1030连同本体模式110和节点能力155一起评估。在一个实施例中,数据放置组件1030基于查询操作将超图1025中的概念和关系分组为潜在重叠子集。然后,可以基于它们所支持的操作将对应于这些子集的数据放置在各个后端数据节点上。在一些实施例中,以所标识的本体子集的粒度做出数据放置决策,将任何本体概念的所有数据全部放置。换句话说,放置决策165不在不同存储库之间水平分割概念。例如,如果数据放置组件1030将“company”概念放置在第一数据节点中,则对应于“company”概念的所有实例级别数据被放置在第一数据节点中。

[0118] 在一些实施方式中,基于本体的数据放置组件1030遵循用于数据放置的两步方法。首先,数据放置组件1030在表示概括的工作负载1015的超图1025上运行图分析算法,以便基于对这些概念执行的操作的相似性来对域本体模式110中的概念进行分组。接下来,与这些标识的组或子集相对应的数据根据它们各自的能力映射到底层数据存储库,同时最小化所需的复制量。在实施例中,所得到的基于能力的数据放置在查询处理时间针对给定工

作负载最小化数据移动(和数据变换),从而极大地增强了多存储库环境中的查询处理的效率。如图所示,数据放置组件1030的最终输出是将本体概念映射到适当数据节点的概念到存储库(concept-to-store)映射(例如,放置决策165)。

[0119] 尽管在所示工作流1000中未描绘,但是在实施例中,系统然后利用放置决策165来将数据存储在各种后端资源中。在一个实施例中,系统通过调用提取、变换和加载(ETL)服务来这样做,该ETL服务执行允许数据被存储在相关数据节点130中所必需的任何变换或转换。

[0120] 在一些实施例中,工作流1000被用于周期性地重新评估和细化数据放置以便维持系统的效率。例如,在非高峰时间(off-peak time)期间(例如,在非营业时间期间),系统可以基于更新的OQL查询的集合1005(例如,包括在最后的数据放置决策之后接收的查询)来调用工作流1000,以便确定是否应当更新放置以反映演进工作负载。这可以通过防止数据位置变得陈旧来提高系统的功效。

[0121] 图11描绘了根据本文公开的一个实施例的用于评估工作负载和存储知识库数据的示例超图1100。在所描述的实施例中,每个概念1105A-G被描述为椭圆,并且每个超边1110A-D被描述为包围其对应的概念1105的虚线。例如,超边1110A连接概念1105A(“Company”)、1105B(“PublicCompany”)、1105C(“PublicMetric”)和1105D(“PublicMetricData”)。如图所示,每个超边1110可以包括概念1105的不连接的子集(例如,超边1110A和1110C不重叠)以及重叠子集(例如,超边1110A和1110B相对于“Company”概念1105A重叠)。

[0122] 另外,如图所示,每个超边1110用该边的相关操作1115A-C来标记。如上所述,在一个实施例中,标签指示可以或已经应用于由超边1110联结的概念1105的操作的集合。例如,在所描绘的示例中,超边1110A与操作1115A(“连接”)、1115B(“聚合”)和1115C(“模糊”匹配)相关联。值得注意的是,每个操作1115可以与任何数量的超边1110和/或概念1105相关联。在所描述的实施例中,“连接”操作1115A与超边1110A和1110B相关联,“聚合”操作1115B与超边1110A和1110D相关联,并且“模糊”操作1115C与超边1110A、1110D和1110C相关联。

[0123] 图12是示出了根据本文公开的一个实施例的被配置为评估工作负载并存储数据的数据放置协调器150的框图。尽管被描绘为物理设备,但是在实施例中,数据放置协调器150可以使用(一个或多个)虚拟设备和/或跨多个设备(例如,在云环境中)来实现。如图所示,数据放置协调器150包括处理器1210、存储器1215、存储装置1220、网络接口1225和一个或多个I/O接口1230。在所示实施例中,处理器1210检索并执行存储在存储器1215中的编程指令,以及存储和检索驻留在存储装置1120中的应用数据。处理器1210通常表示单个CPU和/或GPU、多个CPU和/或GPU、具有多个处理核的单个CPU和/或GPU等。存储器1215通常被包括以表示随机存取存储器。存储装置1220可以是盘驱动器、基于闪存的存储设备等的任意组合,并且可以包括固定和/或可移动存储设备,诸如固定盘驱动器、可移动存储器卡、高速缓存、光存储装置、网络附接存储装置(NAS)或存储区域网络(SAN)。

[0124] 在一些实施例中,输入和输出设备(诸如键盘、监视器等)经由(一个或多个)I/O接口1230联结。此外,经由网络接口1225,数据放置协调器150可以与一个或多个其它设备和组件通信地耦合(例如,经由网络1280,其可以包括因特网、(一个或多个)本地网络等等)。如图所示,处理器1210、存储器1215、存储装置1220、(一个或多个)网络接口1225和(一个或

多个)I/O接口1230通过一个或多个总线1275通信地耦合。另外,数据放置协调器150经由网络1280通信地耦合到数据节点130A-N。当然,在实施例中,数据节点130A-N可以直接联结到数据放置协调器150,经由本地网络可访问,集成到数据放置协调器150中,等等。尽管未包括在所示的实施例中,但是在一些实施例中,数据放置协调器150还与查询处理协调器115通信地耦合。

[0125] 在所示的实施例中,存储装置1220包括域本体1260、能力数据1265和数据映射1270的副本。尽管被描绘为驻留在存储装置1220中,但是在实施例中,本体1260、能力数据1265和数据映射1270可以以任何合适的位置和方式存储。在一个实施例中,本体1260、能力数据1265和数据映射1270对应于上面参考查询处理协调器115讨论的本体560、能力数据565和数据映射570。

[0126] 例如,如上所述,本体1260可以指示与域相关的实体或概念,以及每个概念/实体的潜在属性和实体/概念之间的关系,而不包括实例级别数据。类似地,如上所述,能力数据1265指示每个数据节点130A-N的能力,这可以包括例如关于每个数据节点130A-N支持哪个(哪些)操作的指示,以及关于该支持的(一个或多个)任何对应的限制。

[0127] 在实施例中,数据映射1270由放置应用1235生成,并且指示针对本体1260中定义的每个实体/概念存储实例数据的(一个或多个)数据节点130。例如,数据映射1270可指示“Company”概念的所有实例都存储在数据节点130A和130B中,而“Document”概念的所有实例都存储在数据节点130B和130N中。

[0128] 在所示实施例中,存储器1215包括放置应用1235。尽管被描绘为驻留在存储器1215中的软件,但是在实施例中,可以使用硬件、软件或硬件和软件的组合来实现放置应用1235。如图所示,放置应用1235包括概括组件1240、超图组件1245和放置组件1250。尽管为了概念清楚而被描绘为离散组件,但在各实施例中,概括组件1240、超图组件1245和放置组件1250的操作可跨任何数量的组件来组合或分布。

[0129] 在实施例中,概述组件1240接收反映系统上的先前、当前、预期、典型、平均、预期和/或一般工作负载的先前OQL查询(或样本查询)。然后,概括组件1240评估所提供的查询以概括所概括的查询形式的工作负载。在实施例中,概括的查询一般指示一起查询的概念以及在概念上执行的操作的集合,但是不对应于可以执行的实际查询。概述组件1240的功能在下面参考图13更详细地讨论。

[0130] 在实施例中,超图组件1245从概括组件1240接收概括的工作负载信息,并解析它以生成表示查询工作负载的超图。如上所述,在一些实施例中,超图包括本体1260中的每个概念的顶点。另外,在实施例中,概念顶点由表示概括的查询工作负载的超边联结。超图组件1245的操作将在下面参考图14更详细地讨论。

[0131] 在一个实施例中,放置组件1250评估超图以便生成数据放置决策(例如,数据映射1270)。然后,这些决策可以用于将数据路由到适当的节点。例如,在一个实施例中,系统迭代通过知识库以放置数据。对于每个数据项,系统可以确定其对应的(一个或多个)概念,查找对应的数据节点130,并将数据存储在(一个或多个)所指示的节点中。在一些实施例中,系统还执行适合于目的地存储库的任何变换或转换。放置组件1250的功能在下面参考图15和图16更详细地讨论。

[0132] 图13是示出根据本文公开的一个实施例的用于评估和概括查询工作负载以通知

数据放置决策的方法1300的流程图。方法1300开始于框1305,其中数据放置协调器150接收表示知识库工作负载的一个或多个查询。在框1310,数据放置协调器150选择所接收的查询之一用于评估。方法1300然后进行到框1315,其中数据放置协调器150标识由所选择的查询涉及的(一个或多个)概念。类似地,在框1320,数据放置协调器150标识由查询调用的(一个或多个)操作。

[0133] 在一个实施例中,数据放置协调器150将所确定的由查询指定的概念的集合与由查询调用的(一个或多个)操作的集合相关联。在一些实施例中,操作-概念配对是在粒度概念/操作级别基础上确定的。在一些其它实施例中,数据放置协调器150不确定哪些操作被链接到哪些概念,而是将操作的集合作为整体链接到概念的集合。也就是说,数据放置协调器150忽略查询的实际执行细节/期望结果,并且基于所涉及的数据/操作而不是变换的特定应用来定义相关概念和操作的集合。

[0134] 方法1300然后进行到框1325,其中数据放置协调器150确定是否还有至少一个附加查询要被评估。如果是,则方法1300返回到框1310。否则,方法1300进行到框1330。在组1330,数据放置协调器150基于针对每个的对应的(一个或多个)概念的集合来对所接收的查询进行分组。在一个实施例中,这包括对指定匹配的概念的集合的所有查询进行分组或聚类。例如,数据放置协调器150可以将指定“Company”概念和“PublicMetric”概念两者的所有查询分组到第一组中。值得注意的是,在这样的实施例中,仅指定“Company”概念或“PublicMetric”概念的查询将被放置于不同的组中。类似地,指定“Company”和“PublicMetric”但还包括“Document”概念的查询将不被放在第一组中。

[0135] 也就是说,在实施例中,数据放置协调器150对指定精确匹配的概念的集合的查询进行分组。指定附加概念或更少概念的查询被放置于其他组中。在一个实施例中,与每个相应分组相对应的概念的集合用于形成相应概括的查询。也就是说,在所示的实施例中,数据放置协调器150通过将访问相同概念的查询分组到集群中来聚合所接收的先前查询,而不管每个查询执行的操作。

[0136] 然后,方法1300进行到框1335,其中数据放置协调器150选择这些定义的查询组中的一个。在框1340,数据放置协调器150选择与所选择的组相关联的查询之一。另外,在框1345,数据放置协调器150将由选择的查询指定的对应的操作与表示选择的查询组的概括的查询相关联。在一个实施例中,如果操作已经反映在概括的查询中,则数据放置协调器150不再次添加它。也就是说,在实施例中,与概括的查询关联的操作集是二元值,指示给定操作的存在或者不存在。

[0137] 在实施例中,与概括的查询相关联的操作可以是任何粒度水平。例如,在一些实施例中,数据放置协调器150在操作级别(例如,“连接”)定义概括的查询,而不考虑操作的特定类型(例如,内部连接)、对操作的限制或者操作的细节(例如,数据的类型,诸如字符串连接或者整数连接)。在其它实施例中,这些细节被包括在概括的查询描述中,以便提供用于后续处理的更丰富的细节。

[0138] 方法1300然后继续到框1350,其中数据放置协调器150确定在所选择的组中是否存在至少一个附加查询。如果是,则方法1300返回到框1340。如果没有,则在框1355,数据放置协调器150确定是否还有至少一个附加查询组还没有被评估。如果是,则方法1300返回到框1335。否则,方法1300继续到框1360。在框1360,数据放置协调器150存储概括的工作负

载,使得其可以被使用和评估以生成超图。

[0139] 图14是示出根据本文公开的一个实施例的用于对本体工作负载建模以通知数据放置决策的方法1400的流程图。方法1400在框1405开始,其中数据放置协调器150选择本体的概念之一。在框1410,数据放置协调器150为该选择的概念生成超图顶点。方法1400然后继续到框1415,其中数据放置协调器150确定本体中是否剩余在超图中还没有顶点的任何附加概念。如果是,则方法1400返回到框1405。否则,方法1400进行到框1420。

[0140] 在框1420处,数据放置协调器150选择概括的查询之一。如上文讨论的那样,在实施例中,每个概括的查询指示概念的集合和已经应用于先前工作负载中的概念的操作的对应的集合。在框1425处,数据放置协调器150生成链接由所选择的概括的查询指示的(一个或多个)概念中的每一个的超边。方法1400然后进行到框1430,其中数据放置协调器150利用由选择的概括的查询指示的操作的指示来标记新生成的超边。以这种方式,数据放置协调器150可以随后评估超图以标识知识库的关系和使用的样式。

[0141] 方法1400然后继续到框1435,其中数据放置协调器150确定是否存在要评估并且合并到超图中的至少一个附加概括查询。如果是,则方法1400返回到框1420。否则,方法1400继续到框1440,其中数据放置协调器150存储所生成的超图以供后续使用。

[0142] 图15和图16描绘了示出根据本文公开的一个实施例的用于评估超图以驱动数据放置决策的方法的流程图。参考图15讨论的方法1500示出了用于生成概念映射的基于操作的聚类技术的一个实施例,而下面参考图16讨论的方法1600示出了最小覆盖技术的一个实施例。

[0143] 在一个实施例中,基于操作的聚类技术基于概念所经受的操作来对概念进行分组。在一个这样的实施例中,对于超图中的每个操作描述,数据放置协调器150创建相应集群。然后,数据布置协调器150在与每个超边相关联的操作描述组上迭代,并且对于每个这样的操作描述,将超边所跨越的所有概念分配给对应的操作的集群。在实施例中,一旦概念被聚类到一起,数据放置协调器150就将每个概念集群分配给数据节点的集合,使得每个节点具有与集群的操作描述相匹配的能力描述(例如,能够执行集群的对应的操作)。最后,在基于操作的系统中,数据放置协调器150生成映射,该映射将每个集群中的每个概念映射到对应的标识的数据存储库的集合。

[0144] 作为基于操作的聚类技术的一个实施例的示例,考虑图11中提供的超图1100。最初,数据放置协调器150为每个操作1115A-C(例如,$C_{Join}$、$C_{Agg}$和$C_{Fuzzy}$)生成集群。然后,系统为每个超边确定与其相关联的操作的集合。对于每个指示的操作,系统将由超边1110指定的概念分配给对应的集群。继续上面的示例,$C_{Join}$将包含来自超边1110A的概念1105A、1105B、110C和1105D,以及来自超边1110B的概念1105E和1105F。此外,$C_{Agg}$将包含来自超边1110A的概念1105A、1105B、1105C和1105D,以及来自超边1110D的概念1105G。最后,$C_{Fuzzy}$将包含来自超边1110A的概念1105A、1105B、1105C和1105D,以及来自超边1110C的概念1105H。

[0145] 在许多实施例中,这些基于操作的集群具有显著的重叠。例如,注意,概念1105A、1105B、1105C和1105D被包括在每个集群中。为了最终化映射,在一个实施例中,数据放置协调器150为每个集群标识能够执行对应的操作的所有数据节点130。然后,数据放置协调器150将集群中的所有概念1105映射到所有标识的数据节点130。在一些实施例中,尽管基于操作的技术可通过将数据放置到支持对应的操作的所有存储中来最小化或减少查询处理

时的数据移动,但它确实引入了一些复制开销,因为相同的概念的集群可被放置在多个存储库处,如果它们具有满足该集群的操作的能力的话。

[0146] 在一些实现中,为了进一步减少复制开销,利用最小覆盖技术的实施例。在一个实施例中,最小覆盖实施例通过进一步最小化数据复制的量,同时仍最小化查询处理时的数据移动,来改进基于操作的技术。在实施例中,该技术利用最小集合覆盖算法来找到支持查询工作负载超图中的每个超边所需的完整的操作的集合所需的最小数量的数据存储库。在一些实施例中,最小覆盖技术最小化满足超边所需的操作的集合的数据存储库的集合上的每个超边的跨度。

[0147] 在最小覆盖技术的一个示例实施例中,数据放置协调器150对于超图中的每个超边,找到覆盖所有指示的操作的最小数量的数据节点。例如,如果所有操作都可以由单个数据节点130A完成,则最小集合仅包括该节点。如果数据节点130A不能完成一个或多个操作,则将一个或多个其它数据节点130B-N添加到最小集合,直到满足所有操作为止。一旦为超边确定了最小集,超边中的每个概念就被映射到对应的最小集合内的每个节点。

[0148] 作为最小覆盖聚类技术的一个实施例的示例,考虑图11中提供的超图1100,假设数据节点130包括被配置为支持操作1115A和1115B的第一数据节点130A、被配置为支持操作1115B和1115C的第二数据节点130B、以及仅支持操作1115B的第三数据节点130C。对于超边1110A,数据放置协调器150可以确定没有节点单独可以支持所有三个所指示的操作,但是数据节点130A和130B的集合可以支持所有三个所指示的操作。因为这两个节点可以支持整个超边,所以不需要将数据节点130C添加到集合中。

[0149] 类似地,超边1110B将被分配给数据节点130A(被配置为提供连接操作的唯一节点),而超边1110C将被分配给数据节点130B(被配置为提供模糊匹配的唯一节点)。最后,超边1110D可以被分配给数据节点130A和130B,或者数据节点130B和130C。在一些实施例中,数据放置协调器150基于其它标准,诸如每个的延迟或计算资源、预定义的偏好等等,在这些另外等效的替代之间进行选择。然后,数据放置协调器150将每个超边1110的概念1105映射到(一个或多个)所分配的数据节点130。

[0150] 图15是示出根据本文公开的一个实施例的用于评估超图以驱动数据放置决策的基于操作符的方法1500的流程图。方法1500在框1505处开始,其中数据放置协调器150选择由超图指示的操作之一。在框1510,数据放置协调器150生成用于所选择的操作的集群。方法1500然后进行到框1515,其中数据放置协调器150确定是否存在在超图中反映的至少一个附加操作,其还没有与其相关联的集群。如果是,则方法1500返回到框1505。否则,方法1500继续到框1520。

[0151] 在框1520,数据放置协调器150选择超图中的超边之一以用于分析。在框1525,数据放置协调器150标识与所选择的边相关联的(一个或多个)概念和(一个或多个)操作。然后,方法1500继续到框1530,其中数据放置协调器150选择这些指示的操作之一。另外,在框1535,数据放置协调器150标识用于所选择的操作的对应的集群,并且将由所选择的超边表示的所有概念添加到该集群。方法1500继续到框1540,其中数据放置协调器150确定所选择的边是否指示将要处理的至少一个附加操作。如果是,则方法1500返回到框1530。

[0152] 如果没有附加操作与所选择的超边相关联,则方法1500继续到框1545,其中数据放置协调器150确定超图是否包括尚未被评估的至少一个附加边。如果是,则方法1500返回

到框1520。否则,方法1500进行到框1550,其中数据放置协调器150将(一个或多个)操作集群映射到被配置为处理每个操作的(一个或多个)对应的数据节点。例如,在一个实施例中,数据放置协调器150为每个集群标识能够执行对应的操作的数据节点的集合。在实施例中,数据放置协调器150然后把集群中的每个概念映射到(一个或多个)所标识的数据节点的集合中的每个节点。数据放置协调器150可以使用这些映射来在各种数据节点之间分布知识库中的数据。

[0153] 图16是示出根据本文公开的一个实施例的用于评估超图以驱动数据放置决策的基于最小覆盖的方法1600的流程图。方法1600在框1605开始,其中数据放置协调器150选择超图中的超边之一。在框1610,数据放置协调器150标识与所选择的边相关联的对应的概念和操作。此外,在框1615,数据放置协调器150确定能够共同满足所有指示的操作的数据节点的最小集合。

[0154] 在一个实施例中,数据放置协调器150通过迭代地评估数据节点的每个组合以确定该组合是否满足所指示的操作来这样做。即,由所选择的边指示的每个操作是否可以由组合中的至少一个数据节点执行。如果不是,则可以丢弃该组合,并且可以选择另一个节点或组合进行测试(或者可以将另一个节点添加到当前组合中)。数据放置协调器150可以确定每个组合是否是完整的和令人满意的,因为它可以执行所有需要的操作,并且标识具有最少数量的数据节点的(一个或多个)组合,因为这些组合将可能导致在运行时间期间最少的数据移动。在一个实施例中,如果两个或多个组合同样小,则数据放置协调器150可以利用预定义的标准或偏好来选择最佳组合。例如,预定义的规则可指示包括至少一个关系数据存储库的组合优先于没有关系数据存储库的组合。作为另一个示例,规则可以指示(一个或多个)特定类型的存储库和/或(一个或多个)特定数据节点130的权重或优先级。在这样的实施例中,数据放置协调器150可以为每个组合聚合或以其他方式聚合这些权重,以确定使用哪些存储库。然后,用所确定的数据节点的集合的指示来标记所选择的边。

[0155] 一旦确定了数据节点的最小集合,方法1600就进行到框1620,其中数据放置协调器150确定超图中是否存在至少一个附加超边。如果是,则方法1600返回到框1605。否则,方法1600进行到框1625。在框1625,数据放置协调器150选择系统中的可用数据节点中的一个。在框1630,数据放置协调器150标识超图中具有包括所选择的数据节点的标签的所有超边。然后,数据放置协调器150将这些超边(或者每个包括的概念)分组或聚类在一起,以形成将被存储在所选择的节点中的概念的组/集群。然后,方法1600进行到框1635,其中数据放置协调器150确定在系统中是否存在还没有被分配组/集群的至少一个附加数据节点。如果是,则方法1600返回到框1625。

[0156] 否则,方法1600进行到框1640,其中,数据放置协调器150对于每个数据节点将包括在对应的集群中的所有概念映射到存储库。数据放置协调器150随后可以使用这些映射来在知识库中跨越各种数据节点分布数据。

[0157] 图17是示出根据本文公开的一个实施例的用于将本体概念映射到存储节点的方法1700的流程图。方法1700开始于框1705,其中数据放置协调器150确定对应于域的查询工作负载信息。在框1710,数据放置协调器150将查询工作负载信息建模为超图,其中超图包括顶点的集合和超边的集合,其中顶点的集合中的每个顶点对应于与域相关联的本体中的概念。方法1700然后进行到框1715,其中,数据放置协调器150基于超图并且进一步基于多

个数据节点中的每一个的预定义的能力,生成概念与多个数据节点之间的映射。另外,在框1720,数据放置协调器150基于所生成的映射建立分布式知识库。

[0158] 已经出于说明的目的呈现了对本公开的各种实施例的描述,但是其并非旨在是穷举的或限于所公开的实施例。在不背离所描述的实施例的范围的情况下,许多修改和变化对于本领域的普通技术人员将是显而易见的。选择本文所使用的术语以最好地解释实施例的原理、实际应用或对市场上存在的技术改进,或使本领域的其他普通技术人员能够理解本文所公开的实施例。

[0159] 在前述内容和/或以下内容中,参考本公开中呈现的实施例。然而,本公开的范围不限于具体描述的实施例。相反,无论是否涉及不同的实施例,前述和/或以下特征和元件的任何组合都被预期用于实现和实践预期的实施例。此外,尽管本文公开的实施例可以实现优于其他可能的解决方案或现有技术的优点,但是给定实施例是否实现特定优点不限制本公开的范围。因此,前述和/或以下方面、特征、实施例和优点仅是说明性的,并且不被认为是所附权利要求的元素或限制,除非在权利要求中明确地陈述。同样,对“本发明”的引用不应被解释为对本文所公开的任何发明主题的概括,并且不应被认为是所附权利要求的元素或限制,除非在权利要求中明确记载。

[0160] 本公开的各方面可以采取完全硬件实施例、完全软件实施例(包括固件、驻留软件、微代码等)或组合软件和硬件方面的实施例的形式,其在本文可以全部统称为“电路”、“模块”或“系统”。

[0161] 本发明可以是系统、方法和/或计算机程序产品。计算机程序产品可以包括其上具有计算机可读程序指令的计算机可读存储介质(或多个介质),所述计算机可读程序指令用于使处理器执行本发明的各方面。

[0162] 计算机可读存储介质可以是能够保留和存储由指令执行设备使用的指令的有形设备。计算机可读存储介质可以是例如但不限于电子存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备或前述的任何合适的组合。计算机可读存储介质的更具体示例的非穷举列表包括以下:便携式计算机磁盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦除可编程只读存储器(EPROM或闪存)、静态随机存取存储器(SRAM)、便携式光盘只读存储器(CD-ROM)、数字多功能盘(DVD)、记忆棒、软盘、诸如上面记录有指令的打孔卡或凹槽中的凸起结构的机械编码装置,以及上述的任何适当组合。如本文所使用的计算机可读存储介质不应被解释为暂时性信号本身,诸如无线电波或其他自由传播的电磁波、通过波导或其他传输介质传播的电磁波(例如,通过光纤线缆的光脉冲)、或通过导线发送的电信号。

[0163] 本文描述的计算机可读程序指令可以从计算机可读存储介质下载到相应计算/处理设备,或者经由网络,例如因特网、局域网、广域网和/或无线网络,下载到外部计算机或外部存储设备。网络可以包括铜传输电缆、光传输光纤、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配卡或网络接口从网络接收计算机可读程序指令,并转发计算机可读程序指令以存储在相应计算/处理设备内的计算机可读存储介质中。

[0164] 用于执行本发明的操作的计算机可读程序指令可以是汇编指令、指令集架构(ISA)指令、机器相关指令、微代码、固件指令、状态设置数据,或者以一种或多种编程语言

的任意组合编写的源代码或目标代码,所述编程语言包括面向对象的编程语言(例如Smalltalk、C++等)以及常规的过程式编程语言(例如“C”编程语言或类似的编程语言)。计算机可读程序指令可以完全在用户的计算机上执行,部分在用户的计算机上执行,作为独立的软件包执行,部分在用户的计算机上并且部分在远程计算机上执行,或者完全在远程计算机或服务器上执行。在后一种情况下,远程计算机可以通过任何类型的网络联结到用户的计算机,包括局域网(LAN)或广域网(WAN),或者可以联结到外部计算机(例如,使用因特网服务提供商通过因特网)。在一些实施例中,为了执行本发明的各方面,包括例如可编程逻辑电路、现场可编程门阵列(FPGA)或可编程逻辑阵列(PLA)的电子电路可以通过利用计算机可读程序指令的状态信息来执行计算机可读程序指令以使电子电路个性化。

[0165] 在此参考根据本发明实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图描述本发明的各方面。将理解,流程图和/或框图的每个框以及流程图和/或框图中的框的组合可以由计算机可读程序指令来实现。

[0166] 这些计算机可读程序指令可以被提供给通用计算机、专用计算机或其他可编程数据处理装置的处理器以产生机器,使得经由计算机或其他可编程数据处理装置的处理器执行的指令创建用于实现流程图和/或框图的一个或多个框中指定的功能/动作的装置。这些计算机可读程序指令还可以存储在计算机可读存储介质中,其可以引导计算机、可编程数据处理装置和/或其他设备以特定方式工作,使得其中存储有指令的计算机可读存储介质包括制品,该制品包括实现流程图和/或框图的一个或多个框中指定的功能/动作的各方面的指令。

[0167] 计算机可读程序指令还可以被加载到计算机、其他可编程数据处理装置或其他设备上,以使得在计算机、其他可编程装置或其他设备上执行一系列操作步骤,以产生计算机实现的过程,使得在计算机、其他可编程装置或其他设备上执行的指令实现流程图和/或框图的一个或多个框中指定的功能/动作。

[0168] 附图中的流程图和框图示出了根据本发明的各种实施例的系统、方法和计算机程序产品的可能实现的架构、功能和操作。在这点上,流程图或框图中的每个框可以表示指令的模块、段或部分,其包括用于实现(一个或多个)指定的逻辑功能的一个或多个可执行指令。在一些替代实施例中,框中所提及的功能可不按图中所提及的次序发生。例如,连续示出的两个框实际上可以基本上同时执行,或者这些框有时可以以相反的顺序执行,这取决于所涉及的功能。还将注意,框图和/或流程图图示的每个框以及框图和/或流程图图示中的框的组合可以由执行指定功能或动作或执行专用硬件和计算机指令的组合的专用的基于硬件的系统来实现。

[0169] 本发明的实施例可以通过云计算基础设施提供给终端用户。云计算通常是指通过网络提供可缩放的计算资源作为服务。更正式地,云计算可以被定义为提供计算资源与其底层技术架构(例如,服务器、存储装置、网络)之间的抽象的计算能力,从而实现对可配置计算资源的共享池的方便的按需网络访问,所述可配置计算资源可以以最小的管理努力或服务提供商交互被快速供应和释放。因此,云计算允许用户访问“云”中的虚拟计算资源(例如,存储、数据、应用,甚至完整的虚拟化计算系统),而不考虑用于提供计算资源的底层物理系统(或那些系统的位置)。

[0170] 通常,云计算资源在按使用付费的基础上被提供给用户,其中用户仅针对实际使

用的计算资源(例如,用户消耗的存储空间量或用户实例化的虚拟化系统的数量)被收费。用户可以在任何时间以及从因特网上的任何地方访问驻留在云中的任何资源。在本发明的上下文中,用户可以访问云中可用的应用(例如,查询处理协调器115)或相关数据。例如,查询处理协调器115可以在云中的计算系统上执行,并且评估查询和后端资源。在这种情况下,查询处理协调器115可以路由查询并将后端资源和/或能力配置存储在云中的存储位置处。这样做允许用户从附接到与云联结的网络(例如,因特网)的任何计算系统访问该信息。

[0171] 虽然前述内容涉及本发明的实施例,但是在不偏离本发明的基本范围的情况下,可以设计本发明的其他和进一步的实施例,并且本发明的范围由所附权利要求确定。

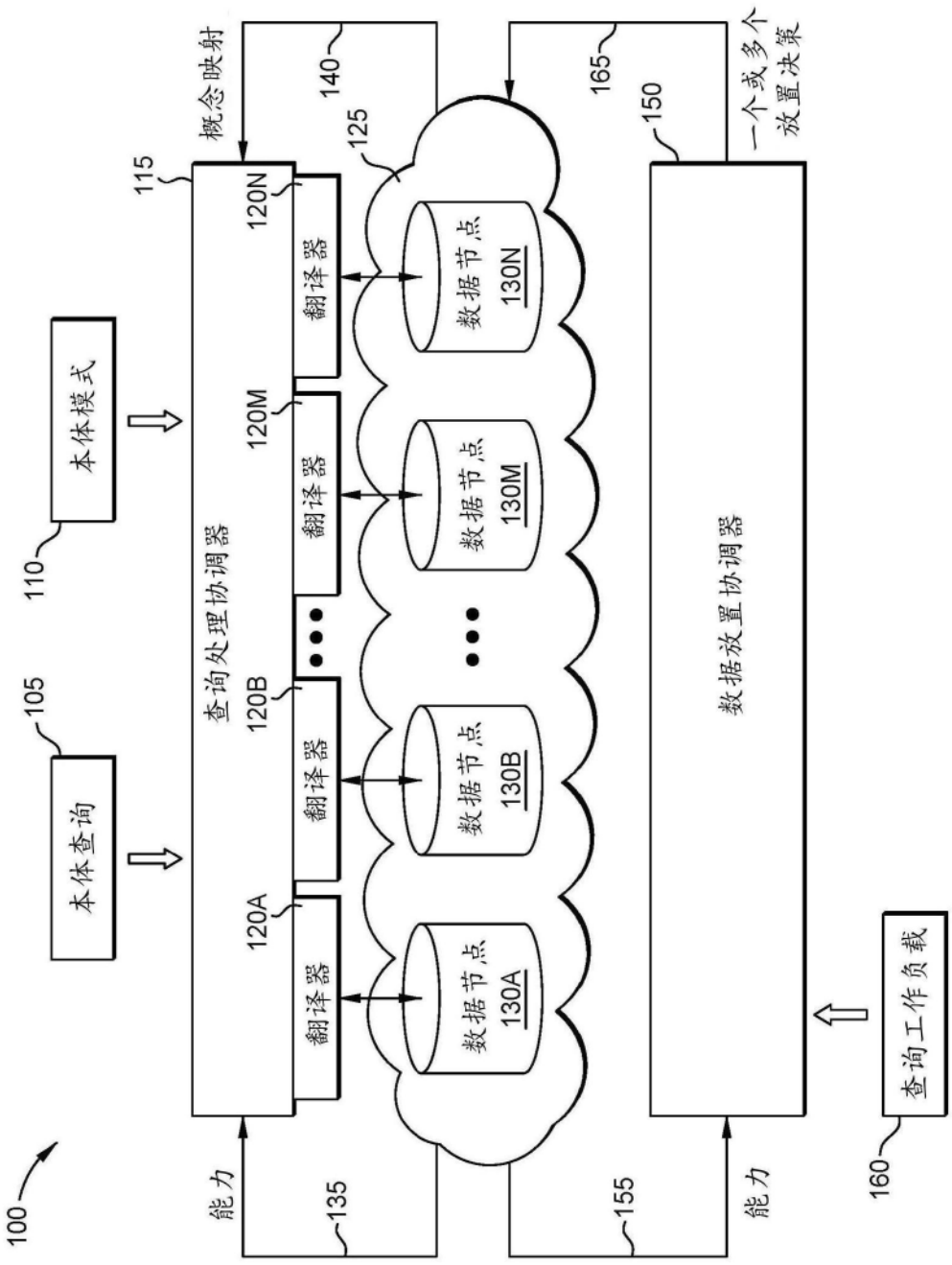

100
本体查询
105
本体模式
110
查询处理协调器
115
概念映射
140
翻译器
120A
翻译器
120B
翻译器
120M
翻译器
120N
数据节点
130A
数据节点
130B
数据节点
130M
数据节点
130N
125
能力
135
数据放置协调器
150
一个或多个放置决策
165
能力
155
查询工作负载
160

图1

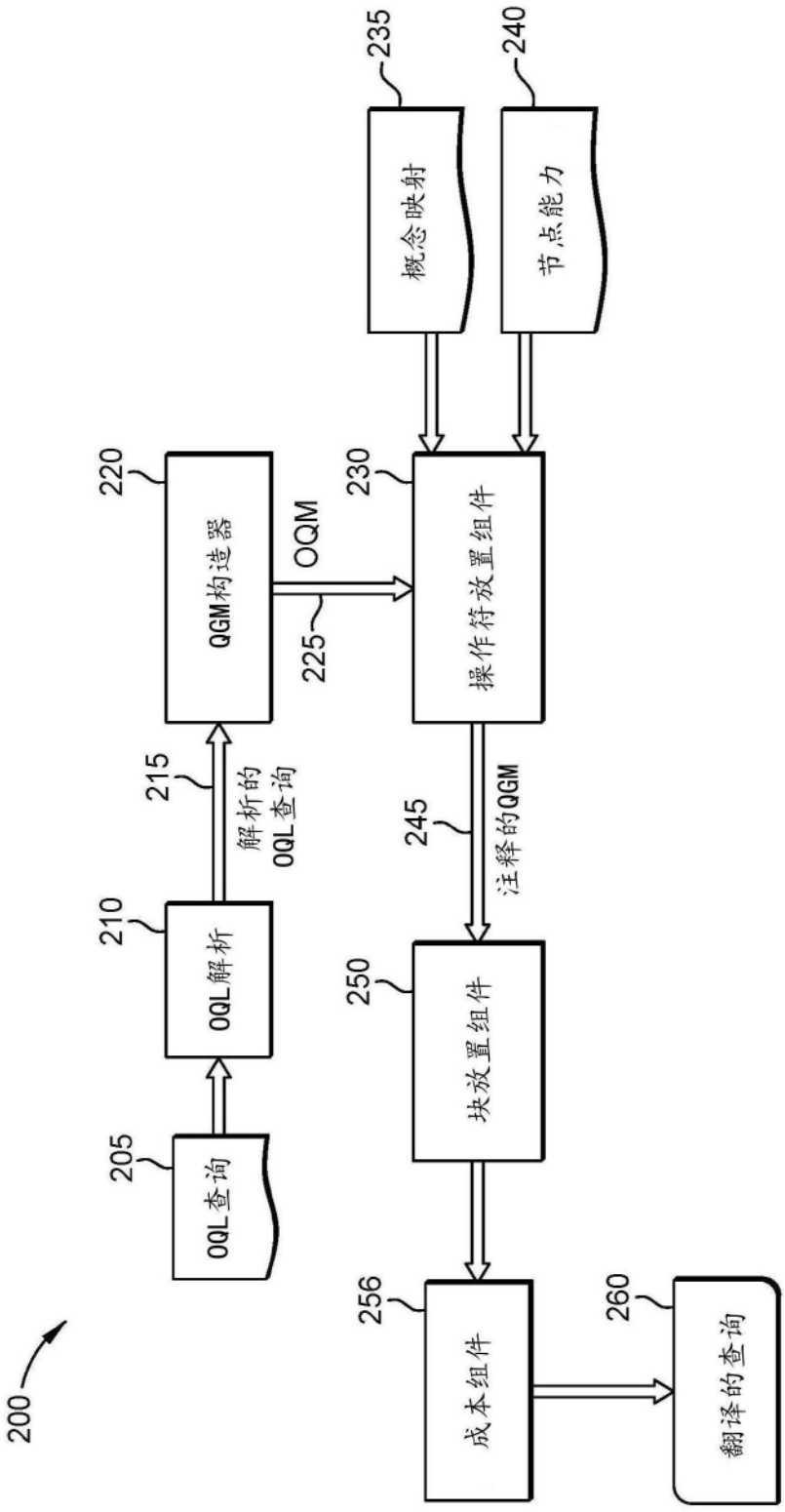

200
205
OQL查询
210
OQL解析
215
解析的OQL查询
220
QGM构造器
OQM
225
230
操作符放置组件
235
概念映射
240
节点能力
245
注释的QGM
250
块放置组件
256
成本组件
260
翻译的查询

图2

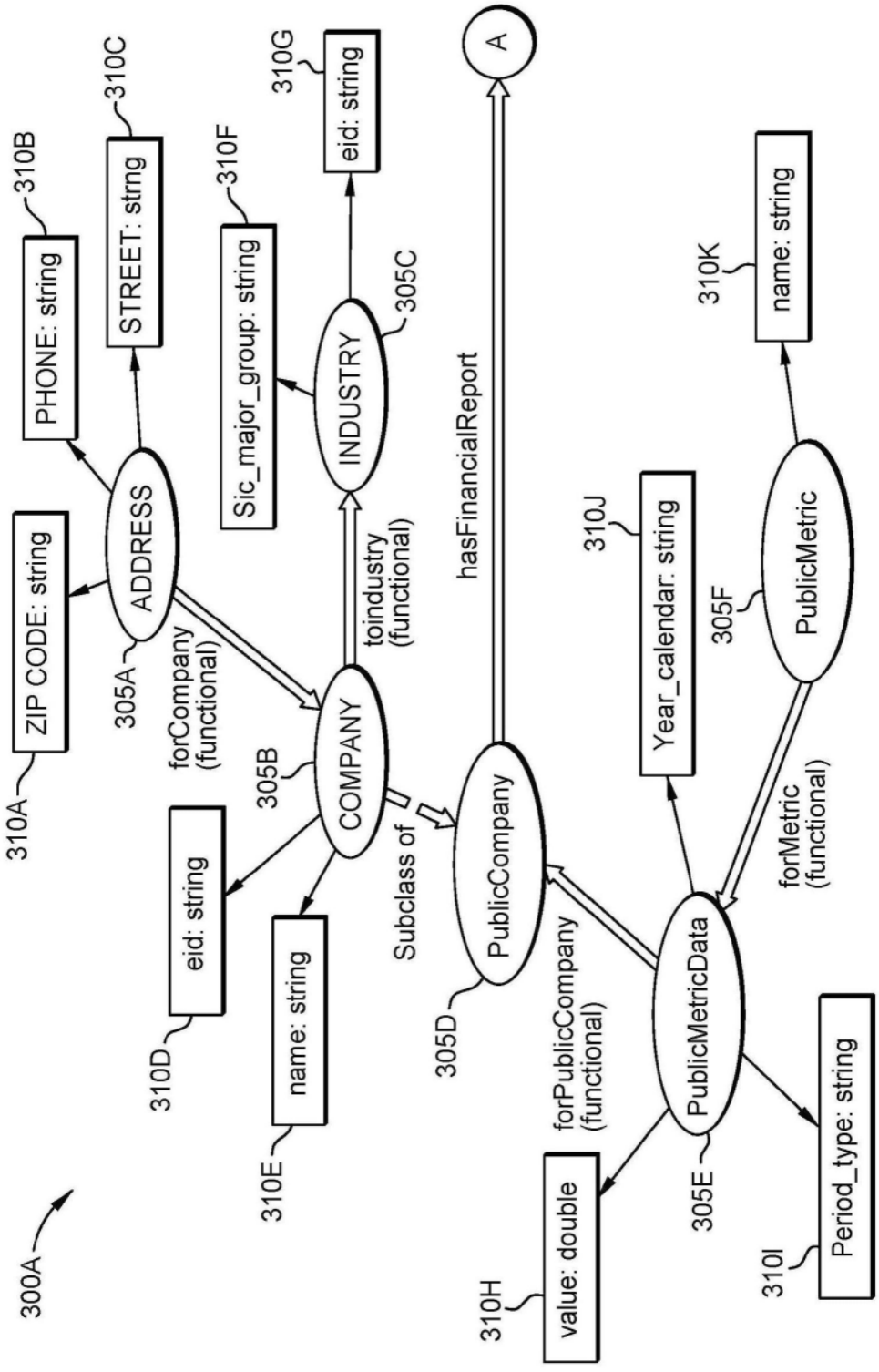

300A
310A
ZIP CODE: string
310B
PHONE: string
310C
STREET: strng
ADDRESS
305A
forCompany
(functional)
310D
eid: string
COMPANY
305B
310E
name: string
Subclass of
310F
Sic_major_group: string
310G
eid: string
INDUSTRY
305C
toindustry
(functional)
A
hasFinancialReport
PublicCompany
305D
forPublicCompany
(functional)
310H
value: double
310J
Year_calendar: string
PublicMetricData
305E
310K
name: string
305F
PublicMetric
forMetric
(functional)
310I
Period_type: string

图3A

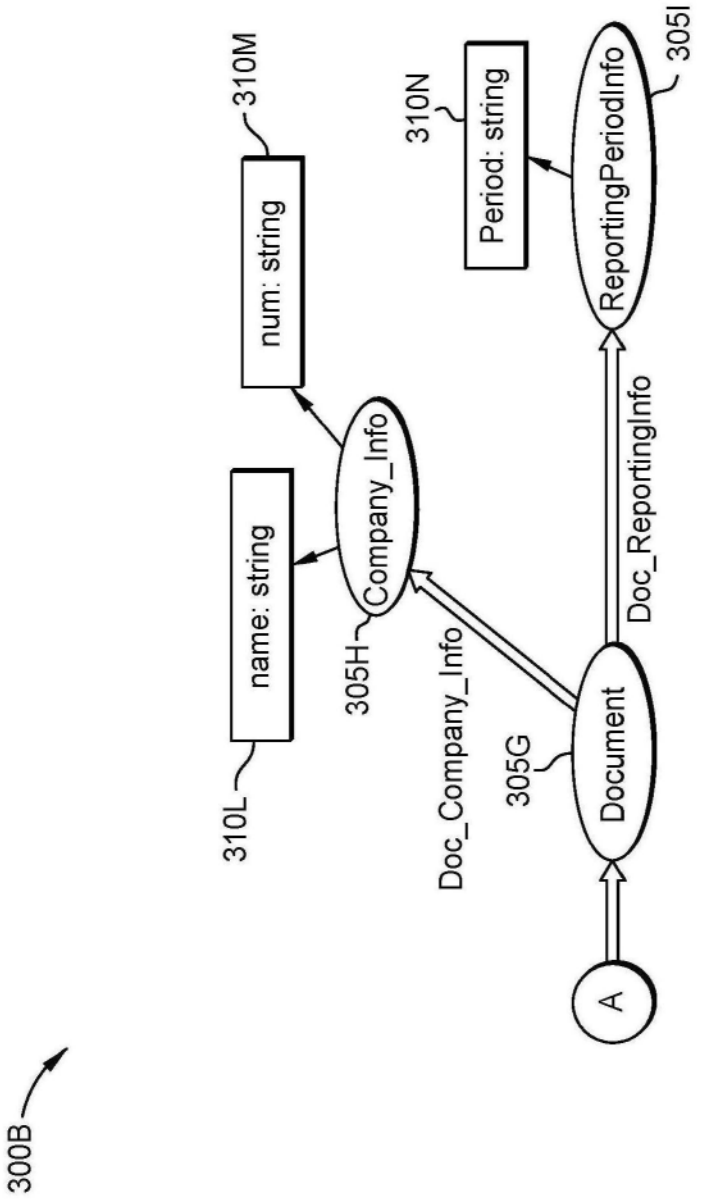

300B
310M
num: string
310N
Period: string
305I
ReportingPeriodInfo
Company_Info
305H
Doc_ReportingInfo
name: string
310L
Doc_Company_Info
305G
Document
A

图3B

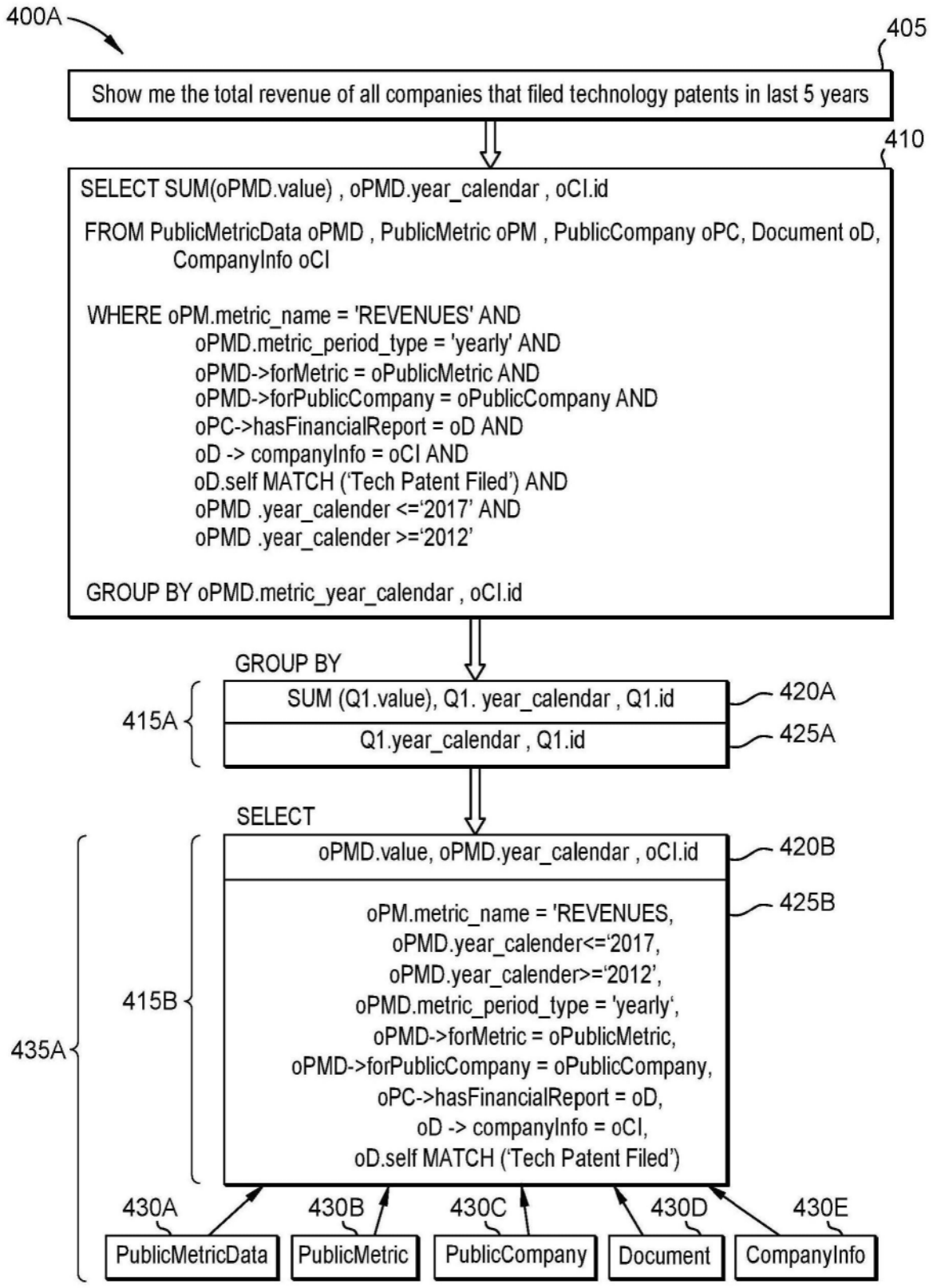

400A
405
Show me the total revenue of all companies that filed technology patents in last 5 years
410
SELECT SUM(oPMD.value) , oPMD.year_calendar , oCI.id
FROM PublicMetricData oPMD , PublicMetric oPM , PublicCompany oPC, Document oD, CompanyInfo oCI
WHERE oPM.metric_name = 'REVENUES' AND
oPMD.metric_period_type = 'yearly' AND
oPMD->forMetric = oPublicMetric AND
oPMD->forPublicCompany = oPublicCompany AND
oPC->hasFinancialReport = oD AND
oD -> companyInfo = oCI AND
oD.self MATCH ('Tech Patent Filed') AND
oPMD .year_calender <='2017' AND
oPMD .year_calender >='2012'
GROUP BY oPMD.metric_year_calendar , oCI.id
GROUP BY
415A
SUM (Q1.value), Q1. year_calendar , Q1.id
Q1.year_calendar , Q1.id
420A
425A
SELECT
oPMD.value, oPMD.year_calendar , oCI.id
oPM.metric_name = 'REVENUES,
oPMD.year_calender<='2017,
oPMD.year_calender>='2012',
oPMD.metric_period_type = 'yearly',
oPMD->forMetric = oPublicMetric,
oPMD->forPublicCompany = oPublicCompany,
oPC->hasFinancialReport = oD,
oD -> companyInfo = oCI,
oD.self MATCH ('Tech Patent Filed')
420B
425B
415B
435A
430A
430B
430C
430D
430E
PublicMetricData
PublicMetric
PublicCompany
Document
CompanyInfo

图4A

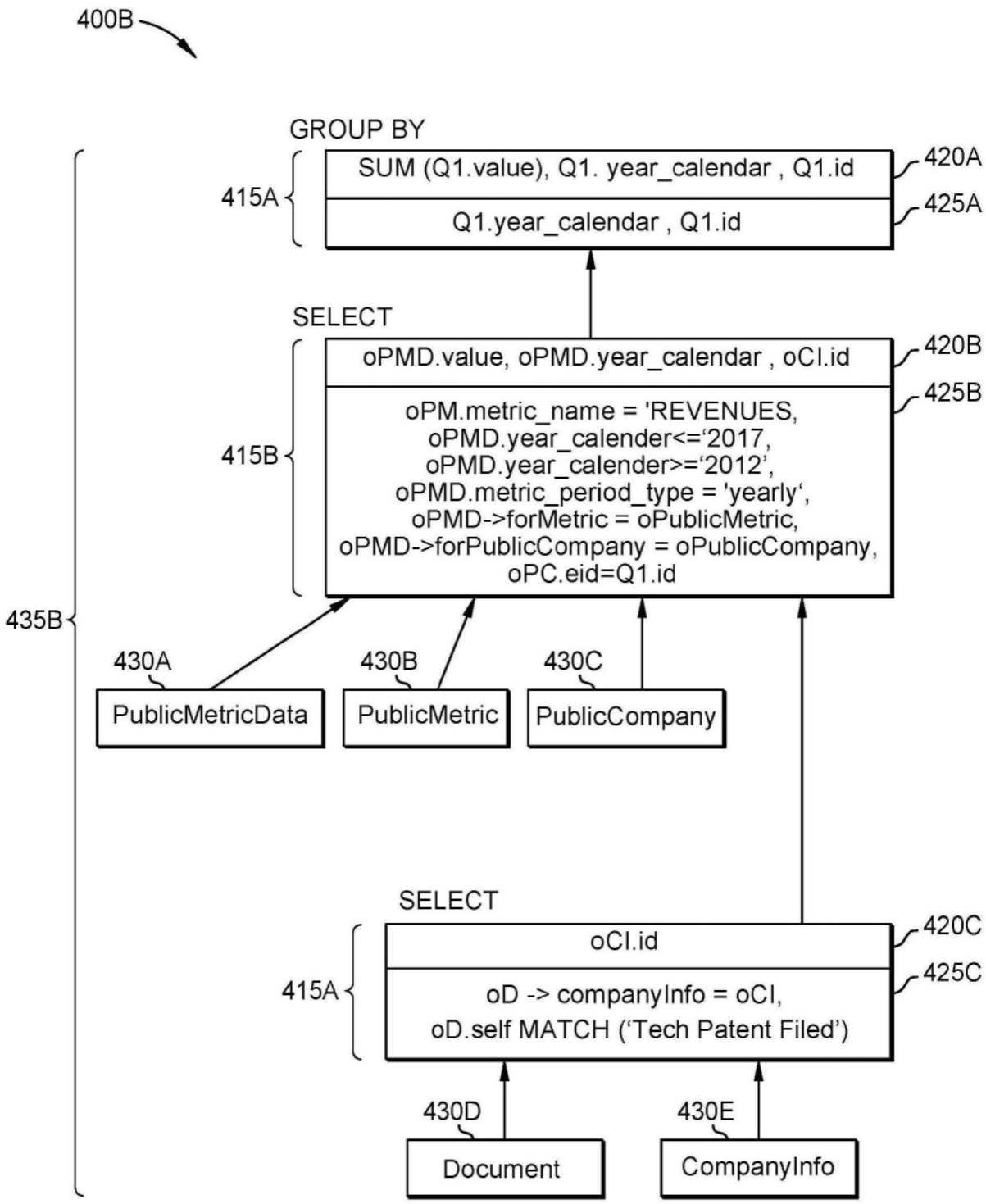

400B
GROUP BY
SUM (Q1.value), Q1. year_calendar , Q1.id
Q1.year_calendar , Q1.id
415A
420A
425A
SELECT
oPMD.value, oPMD.year_calendar , oCI.id
oPM.metric_name = 'REVENUES,
oPMD.year_calender<=‘2017,
oPMD.year_calender>=‘2012’,
oPMD.metric_period_type = 'yearly‘,
oPMD->forMetric = oPublicMetric,
oPMD->forPublicCompany = oPublicCompany,
oPC.eid=Q1.id
415B
420B
425B
435B
430A
PublicMetricData
430B
PublicMetric
430C
PublicCompany
SELECT
oCI.id
oD -> companyInfo = oCI,
oD.self MATCH (‘Tech Patent Filed’)
415A
420C
425C
430D
Document
430E
CompanyInfo

图4B

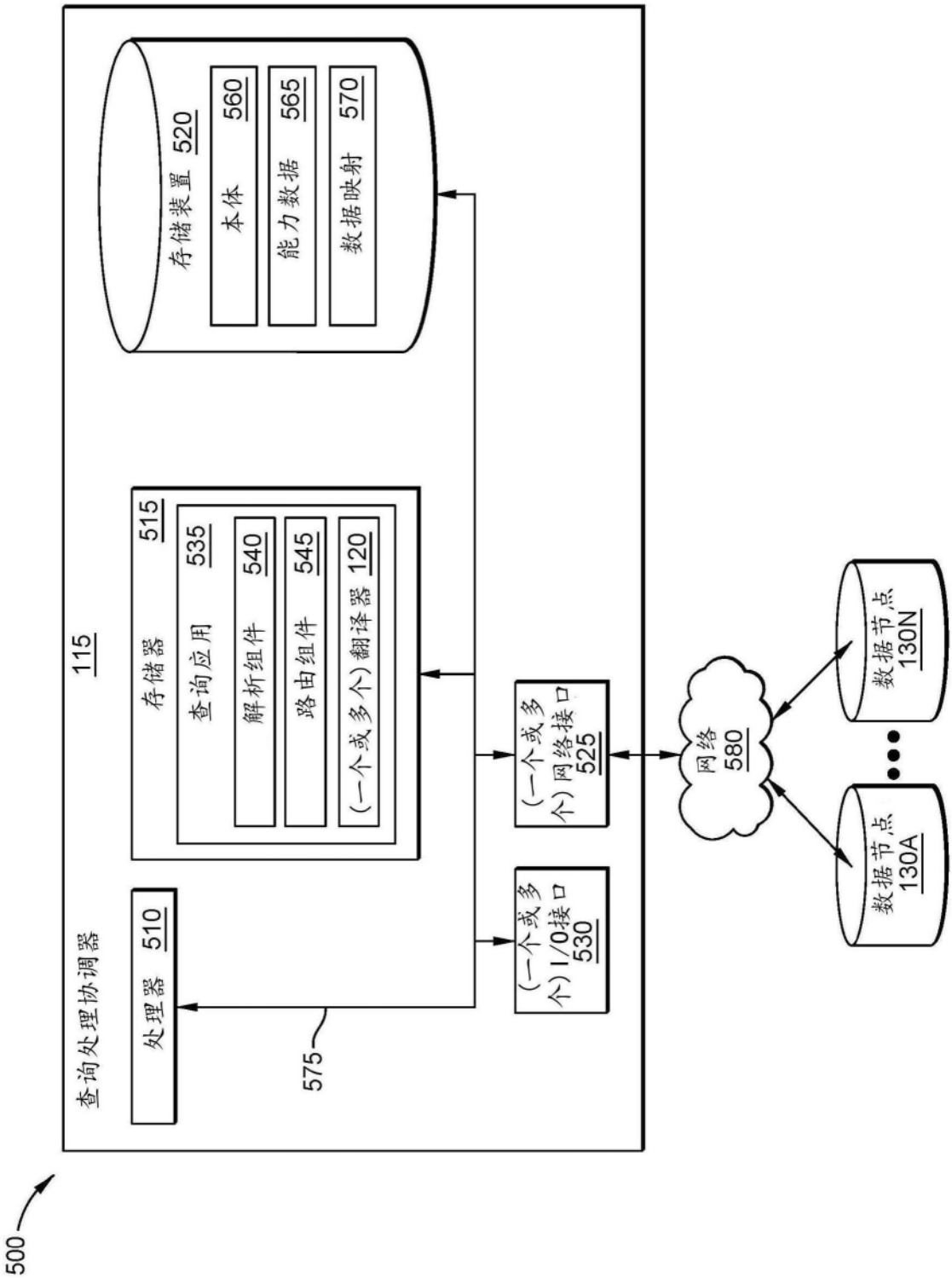

500
查询处理协调器
115
处理器 510
存储器 515
查询应用 535
解析组件 540
路由组件 545
(一个或多个)翻译器 120
存储装置 520
本体 560
能力数据 565
数据映射 570
575
(一个或多个)I/O接口 530
(一个或多个)网络接口 525
网络 580
数据节点 130A
数据节点 130N

图5

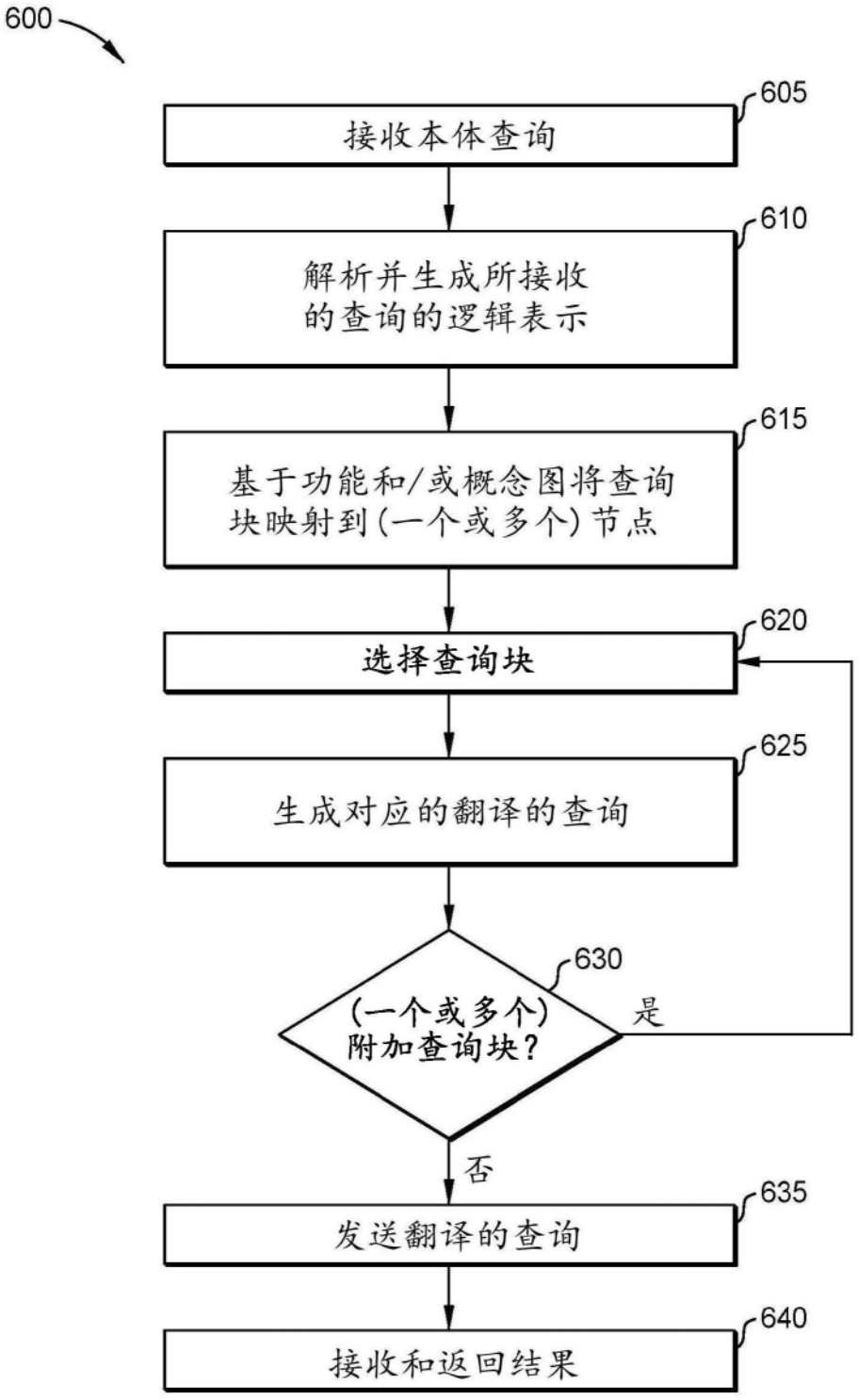

600
605
接收本体查询
610
解析并生成所接收的查询的逻辑表示
615
基于功能和/或概念图将查询块映射到(一个或多个)节点
620
选择查询块
625
生成对应的翻译的查询
630
(一个或多个)附加查询块?
是
否
635
发送翻译的查询
640
接收和返回结果

图6

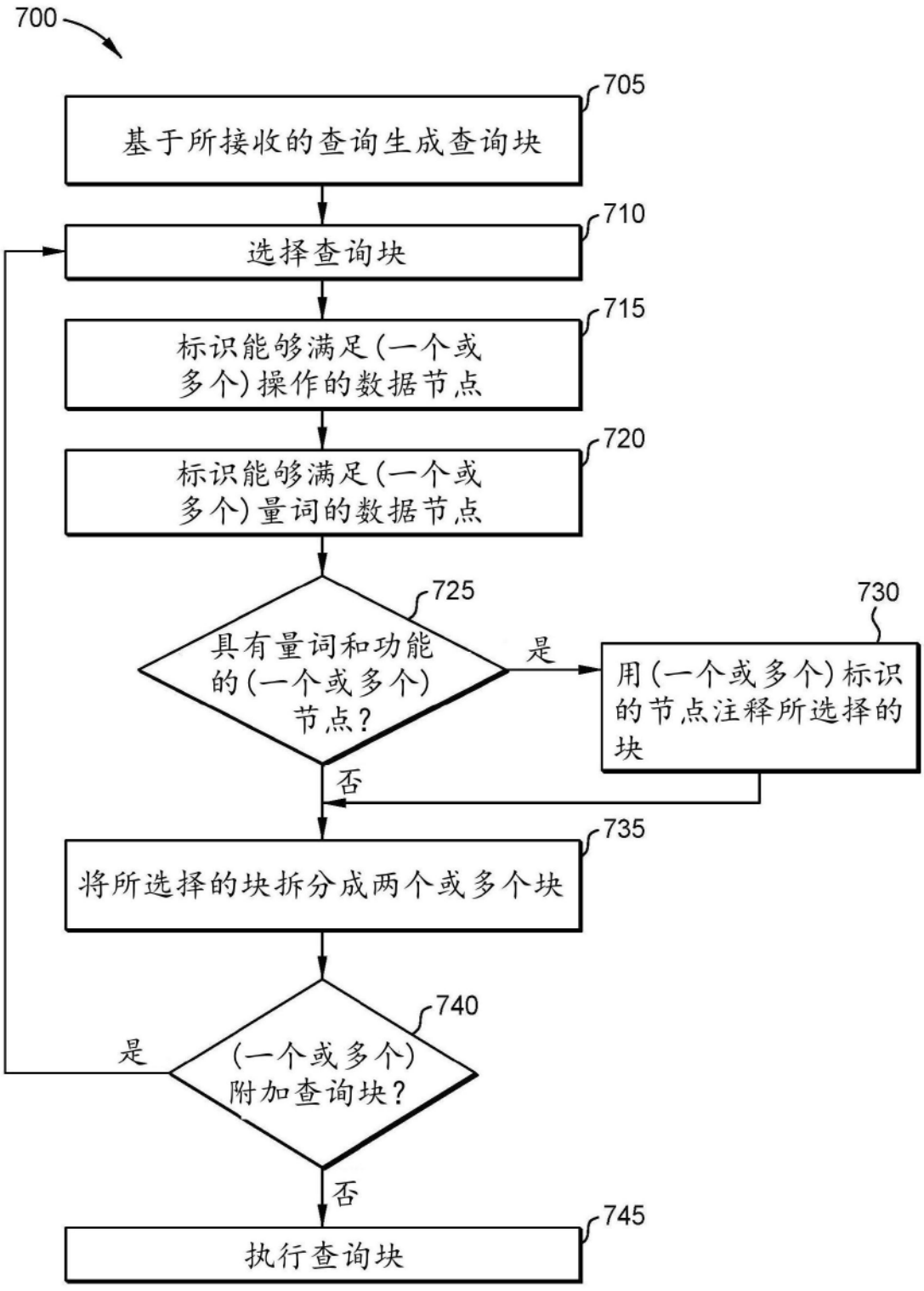

700
705
基于所接收的查询生成查询块
710
选择查询块
715
标识能够满足(一个或多个)操作的数据节点
720
标识能够满足(一个或多个)量词的数据节点
725
具有量词和功能的(一个或多个)节点?
是
730
用(一个或多个)标识的节点注释所选择的块
否
735
将所选择的块拆分成两个或多个块
740
(一个或多个)附加查询块?
是
否
745
执行查询块

图7

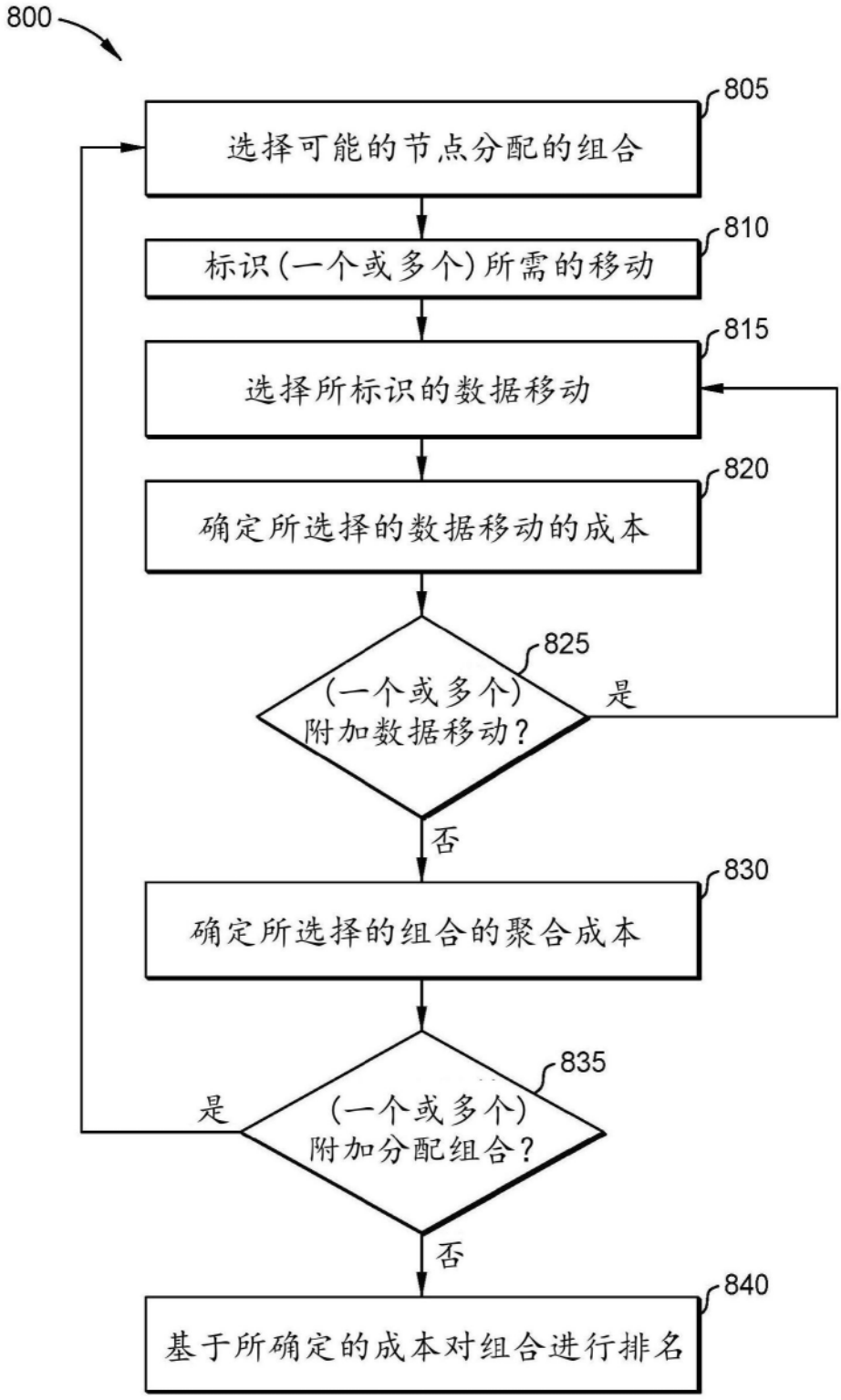

800
805
选择可能的节点分配的组合
810
标识(一个或多个)所需的移动
815
选择所标识的数据移动
820
确定所选择的数据移动的成本
825
(一个或多个)
附加数据移动?
是
否
830
确定所选择的组合的聚合成本
835
是
(一个或多个)
附加分配组合?
否
840
基于所确定的成本对组合进行排名

图8

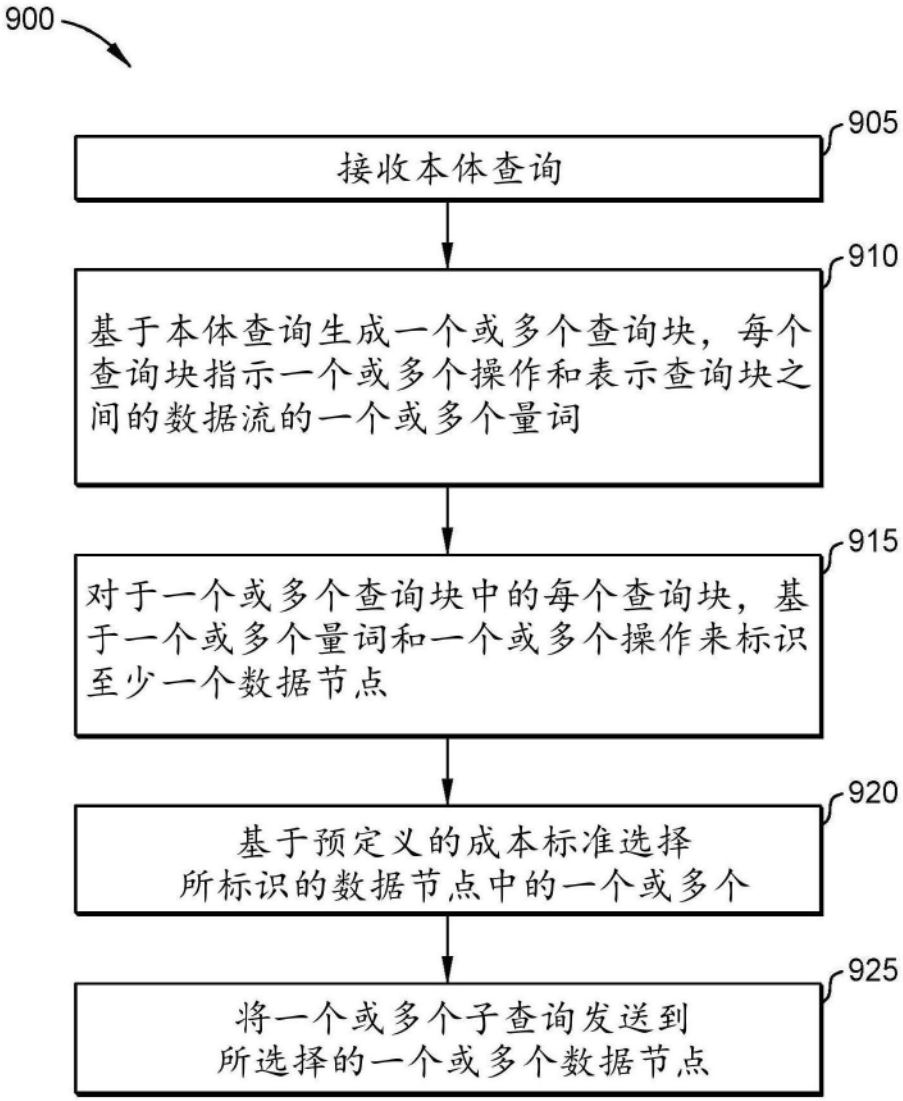

900
905
接收本体查询
910
基于本体查询生成一个或多个查询块，每个查询块指示一个或多个操作和表示查询块之间的数据流的一个或多个量词
915
对于一个或多个查询块中的每个查询块，基于一个或多个量词和一个或多个操作来标识至少一个数据节点
920
基于预定义的成本标准选择所标识的数据节点中的一个或多个
925
将一个或多个子查询发送到所选择的一个或多个数据节点

图9

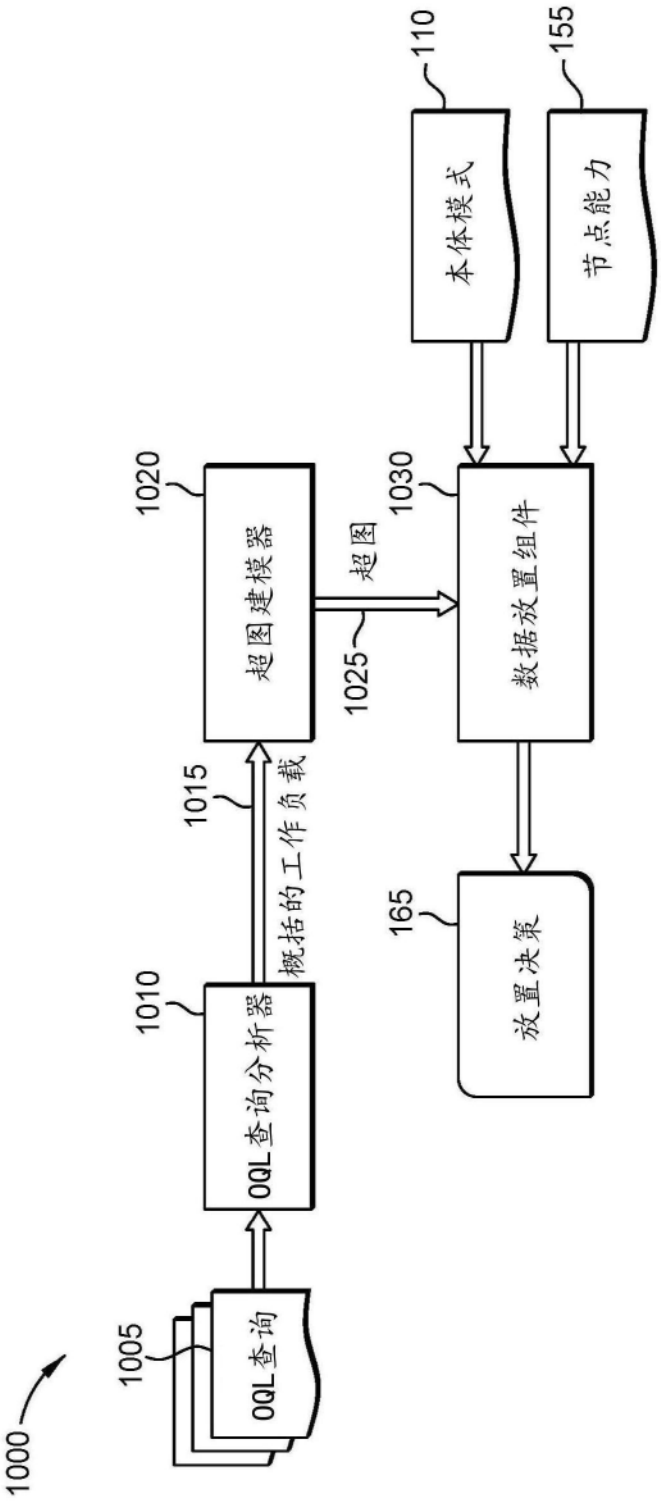

1000
1005
OQL查询
1010
OQL查询分析器
1015
概括的工作负载
1020
超图建模器
1025
超图
1030
数据放置组件
110
本体模式
155
节点能力
165
放置决策

图10

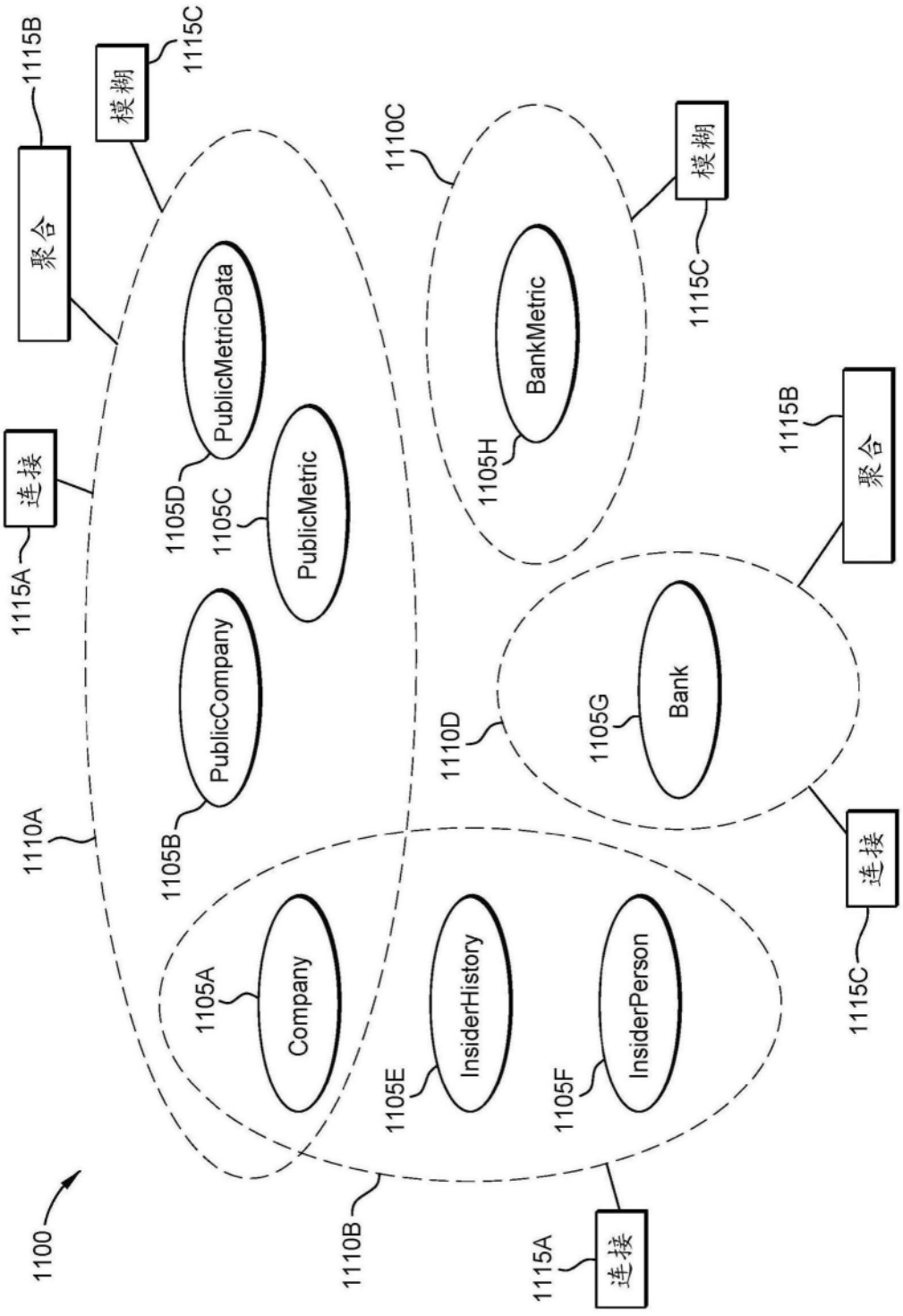

1100
1110A
1110B
1110C
1110D
1115A
连接
1115B
聚合
1115C
模糊
1105A
Company
1105B
PublicCompany
1105C
PublicMetric
1105D
PublicMetricData
1105E
InsiderHistory
1105F
InsiderPerson
1105G
Bank
1105H
BankMetric

图11

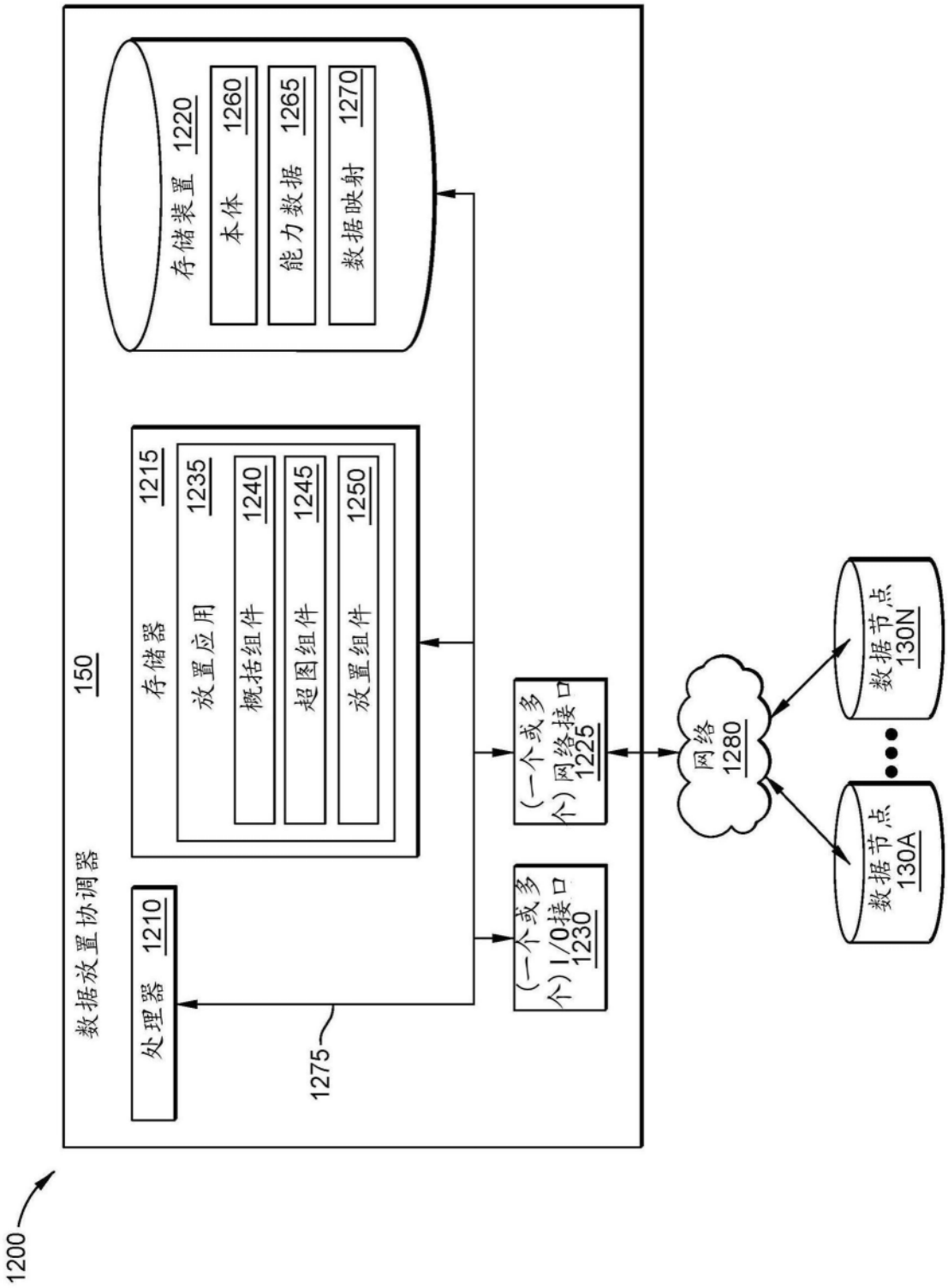

1200
数据放置协调器 150
处理器 1210
存储器 1215
放置应用 1235
概括组件 1240
超图组件 1245
放置组件 1250
存储装置 1220
本体 1260
能力数据 1265
数据映射 1270
1275
(一个或多个) I/O接口 1230
(一个或多个) 网络接口 1225
网络 1280
数据节点 130A
数据节点 130N

图12

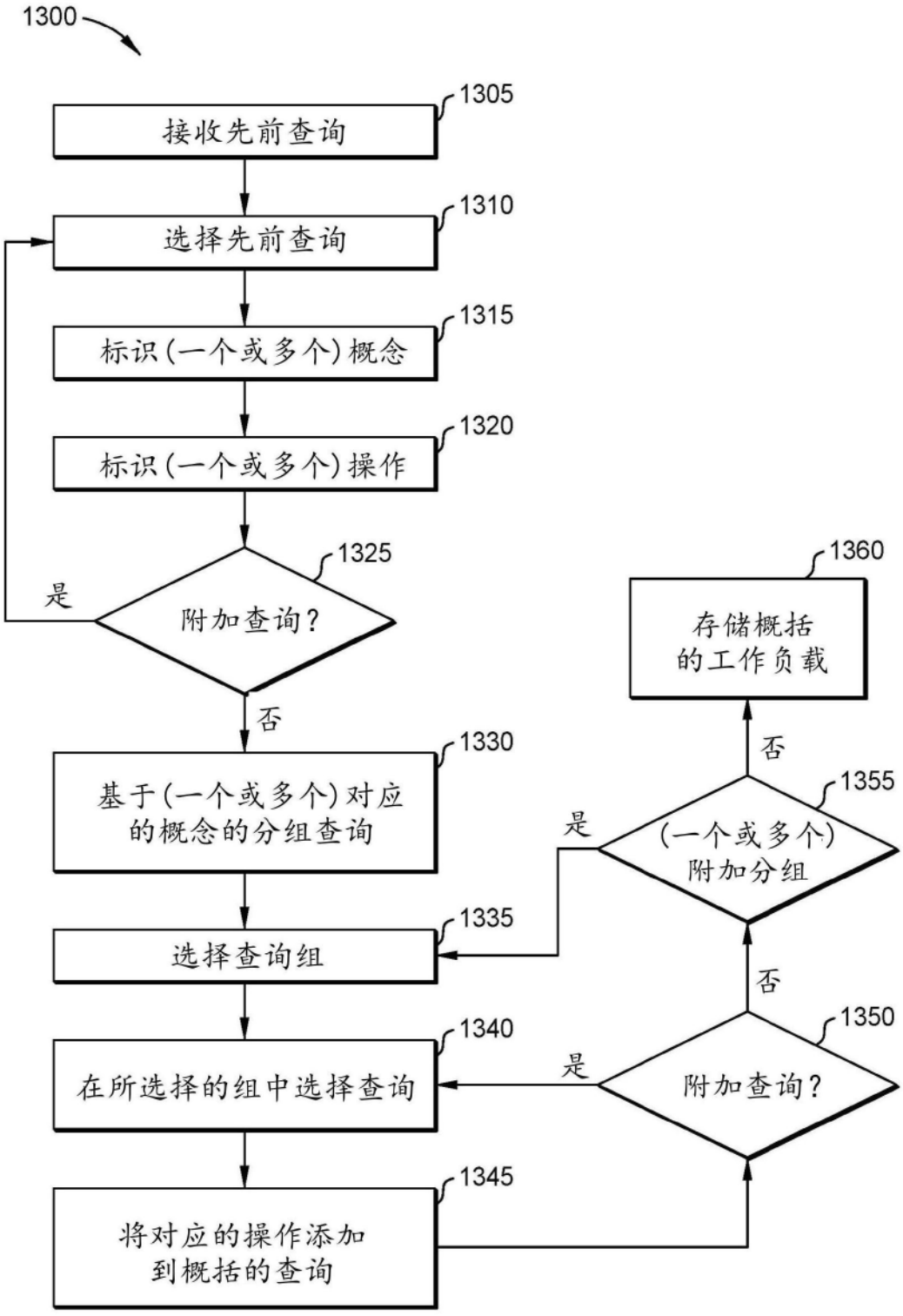

1300
1305
接收先前查询
1310
选择先前查询
1315
标识(一个或多个)概念
1320
标识(一个或多个)操作
1325
附加查询?
是
否
1330
基于(一个或多个)对应的概念的分组查询
1335
选择查询组
1340
在所选择的组中选择查询
1345
将对应的操作添加到概括的查询
1350
附加查询?
是
否
1355
(一个或多个)附加分组
是
否
1360
存储概括的工作负载

图13

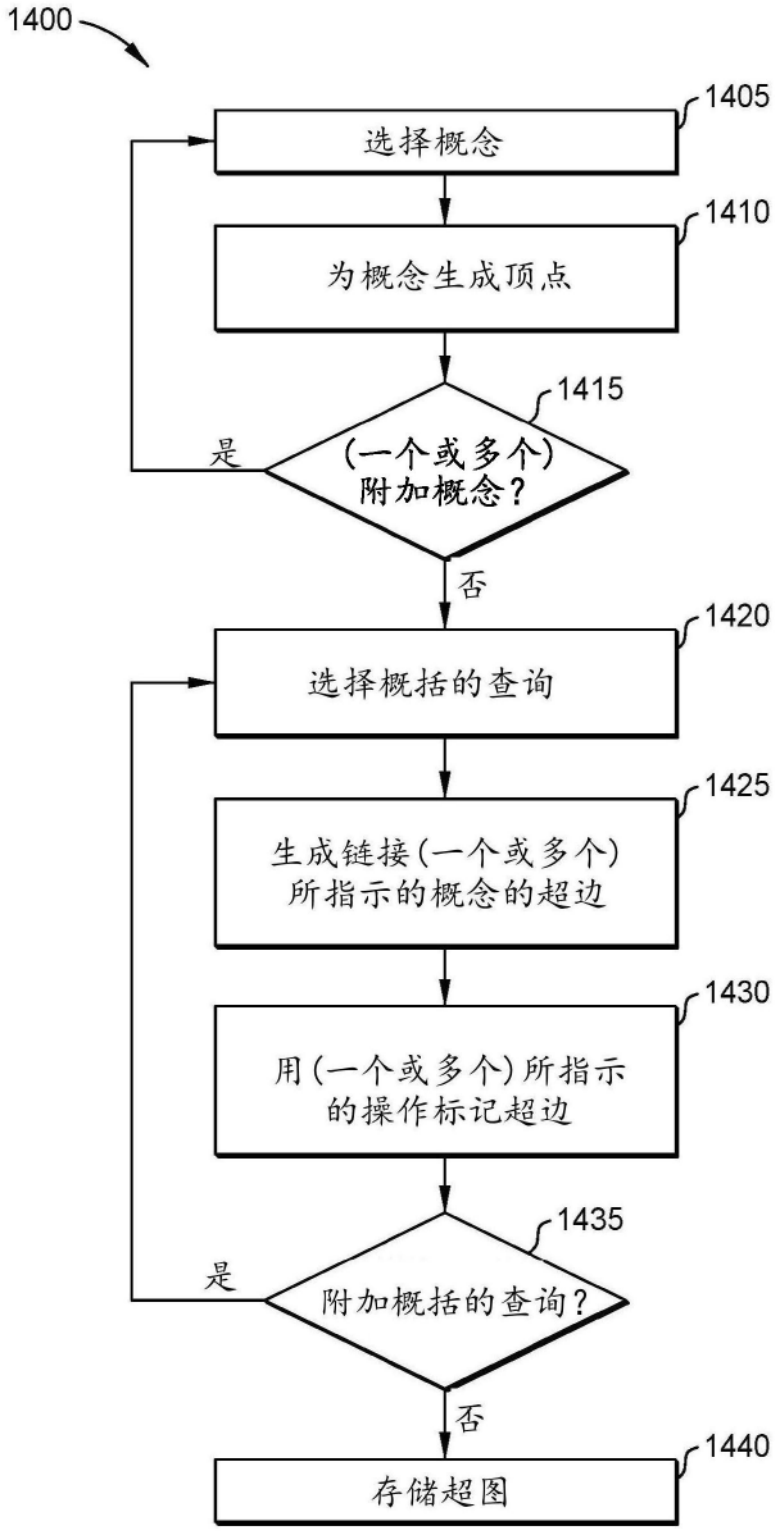

1400
1405
选择概念
1410
为概念生成顶点
1415
(一个或多个)
附加概念?
是
否
1420
选择概括的查询
1425
生成链接(一个或多个)
所指示的概念的超边
1430
用(一个或多个)所指示
的操作标记超边
1435
附加概括的查询?
是
否
1440
存储超图

图14

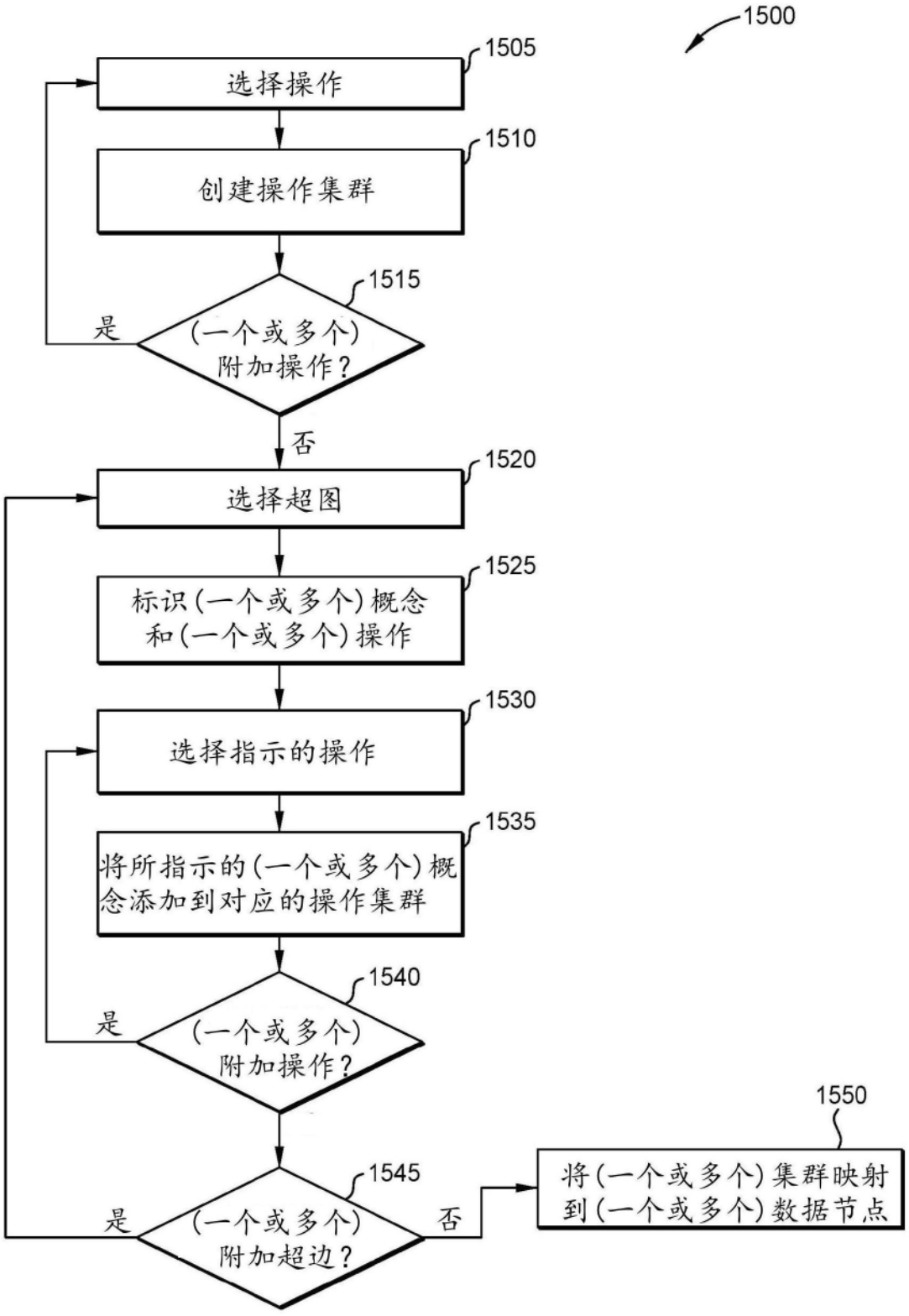

1500
1505
选择操作
1510
创建操作集群
1515
(一个或多个)附加操作?
是
否
1520
选择超图
1525
标识(一个或多个)概念和(一个或多个)操作
1530
选择指示的操作
1535
将所指示的(一个或多个)概念添加到对应的操作集群
1540
(一个或多个)附加操作?
是
1545
(一个或多个)附加超边?
是
否
1550
将(一个或多个)集群映射到(一个或多个)数据节点

图15

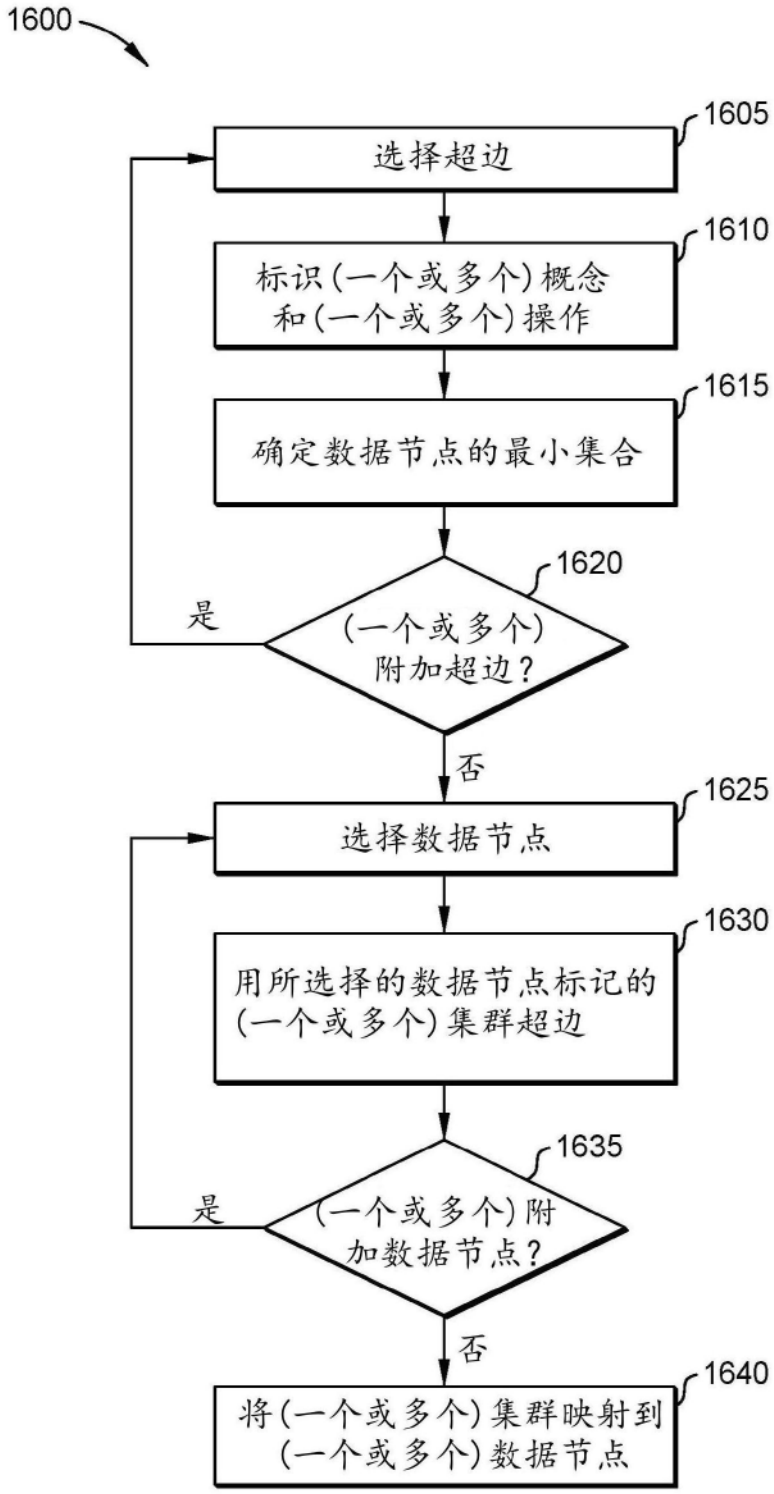

1600
选择超边
1605
标识(一个或多个)概念
和(一个或多个)操作
1610
确定数据节点的最小集合
1615
(一个或多个)
附加超边?
1620
是
否
选择数据节点
1625
用所选择的数据节点标记的
(一个或多个)集群超边
1630
(一个或多个)附
加数据节点?
1635
是
否
将(一个或多个)集群映射到
(一个或多个)数据节点
1640

图16

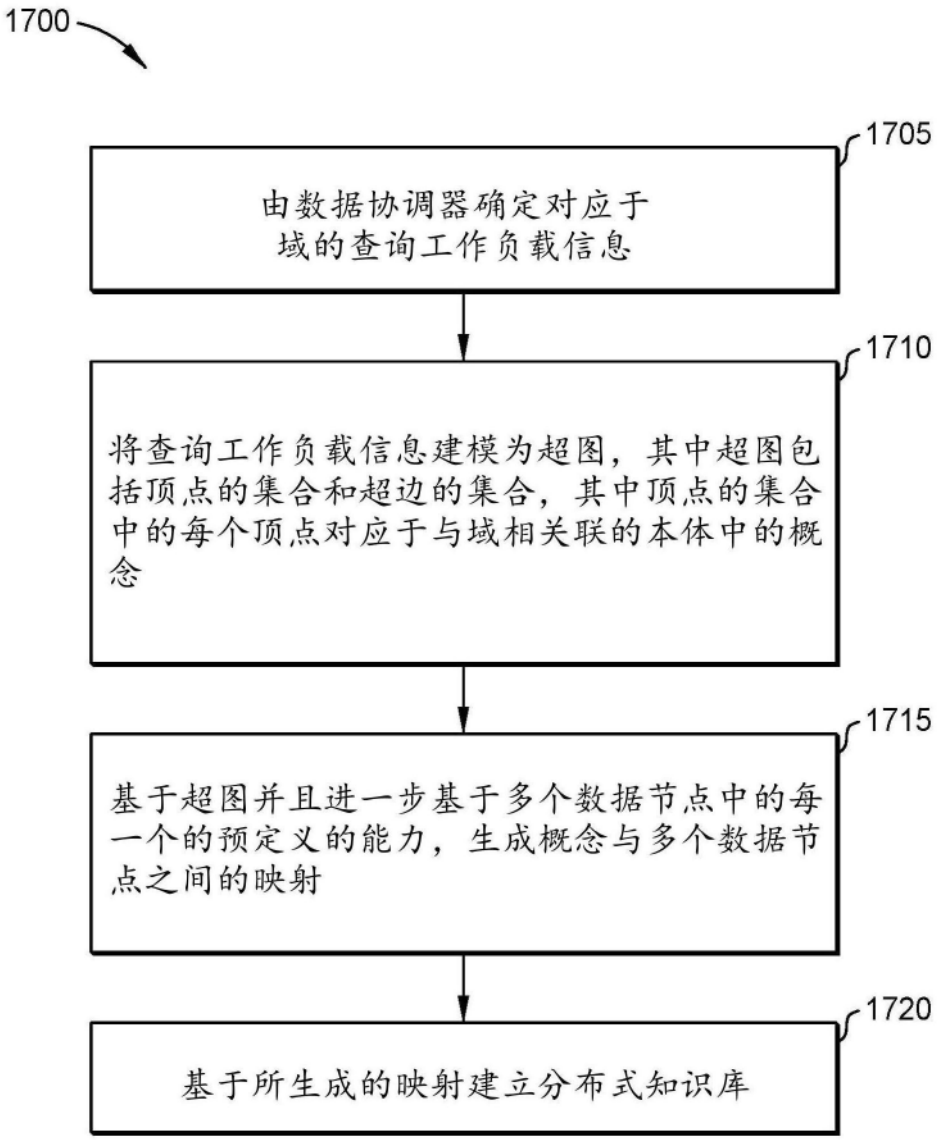

1700
1705
由数据协调器确定对应于域的查询工作负载信息
1710
将查询工作负载信息建模为超图，其中超图包括顶点的集合和超边的集合，其中顶点的集合中的每个顶点对应于与域相关联的本体中的概念
1715
基于超图并且进一步基于多个数据节点中的每一个的预定义的能力，生成概念与多个数据节点之间的映射
1720
基于所生成的映射建立分布式知识库

图17