

数据查询请求的处理方法、装置、设备及存储介质

本申请要求于2022年07月27日提交的申请号为202210891780.8、发明名称为“数据查询请求的处理方法、装置、设备及存储介质”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及数据库领域，特别涉及一种数据查询请求的处理方法、装置、设备及存储介质。

背景技术

基于图数据库的数据通常采用分布式数据存储系统进行存储。分布式数据存储系统通过多个节点设备将数据进行分片存储的方式，来实现支持大规模数据的存储。

存储有数据分片的每台节点设备，会使用节点设备中的多核心处理器的各个核心，来共同管理该节点设备上存储的所有数据。当节点设备接收到针对其存储的某个数据的数据查询请求时，会为该数据查询请求创建相应的线程，以通过创建的线程来处理数据查询请求从而获取查询结果。

当分布式数据存储系统接收到并行的数据查询请求，例如接收到用于查询图数据库中的起始顶点的二度邻居的数据查询请求时，分布式数据存储系统会对不同节点设备中存储的数据进行并发查询。此时接收到数据查询请求的各个节点设备需要为不同的数据查询请求分别创建其使用的线程，在需要创建大量线程的情况下，会导致数据查询消耗的时间增加。

发明内容

本申请提供了一种数据查询请求的处理方法、装置、设备及存储介质，所述技术方案如下。

根据本申请的一方面，提供了一种数据查询请求的处理方法，所述方法应用于第一节点设备，所述第一节点设备是基于图数据库的分布式数据存储系统中的至少两个节点设备中的一个，所述方法包括以下步骤。

所述第一节点设备通过所述第一节点设备中的第一处理模型，接收第二节点设备发送的数据查询请求，所述数据查询请求用于查询与所述图数据库中的第一顶点相关的数据，所述数据查询请求携带有起始顶点的标识，所述第一节点设备存储有所述起始顶点的数据。

所述第一节点设备通过所述第一处理模型，向所述第一节点设备中的第二处理模型发送所述数据查询请求，所述第二处理模型是所述第一节点设备中用于处理所述起始顶点的数据的处理模型。

所述第一节点设备通过所述第二处理模型，基于所述数据查询请求确定第三处理模型，所述第三处理模型用于处理所述第一顶点或中间顶点的数据，所述中间顶点是所述起始顶点和所述第一顶点之间的顶点。

所述第一节点设备通过所述第二处理模型向所述第三处理模型发送所述数据查询请求。

根据本申请的另一方面，提供了一种数据查询请求的处理装置，所述装置是基于图数据库的分布式数据存储系统中的至少两个节点设备中的一个，所述装置包括以下模块。

接收模块，用于通过所述装置中的第一处理模型接收第二节点设备发送的数据查询请求，所述数据查询请求用于查询与所述图数据库中的第一顶点相关的数据，所述数据查询请求携带有起始顶点的标识，所述装置存储有所述起始顶点的数据。

发送模块，用于通过所述第一处理模型，向所述装置中的第二处理模型发送所述数据查询请求，所述第二处理模型是所述装置中用于处理所述起始顶点的数据的处理模型。

确定模块，用于通过所述第二处理模型，基于所述数据查询请求确定第三处理模型，所述第三处理模型用于处理所述第一顶点或中间顶点的数据，所述中间顶点是所述起始顶点和

所述第一顶点之间的顶点。

所述发送模块，用于通过所述第二处理模型向所述第三处理模型发送所述数据查询请求。

根据本申请的另一方面，提供了一种计算机设备，所述计算机设备包括处理器和存储器，所述存储器中存储有至少一段程序，所述至少一段程序由所述处理器加载并执行以实现如上方面所述的数据查询请求的处理方法。

根据本申请的另一方面，提供了一种计算机可读存储介质，所述可读存储介质中存储有至少一段程序，所述至少一段程序由处理器加载并执行以实现如上方面所述的数据查询请求的处理方法。

根据本申请的另一方面，提供了一种计算机程序产品或计算机程序，该计算机程序产品或计算机程序包括计算机指令，该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令，处理器执行该计算机指令，使得该计算机设备执行上述方面的各种可选实现方式中提供的数据查询请求的处理方法。

本申请提供的技术方案带来的有益效果至少包括：

通过节点设备在接收到数据查询请求后，通过节点设备中的处理模型能够直接对数据查询请求进行处理，从而确定用于处理图数据库中与所述数据查询请求相关的顶点的数据的处理模型，进而可实现确定数据查询请求的处理结果。在处理数据查询请求的过程中，无需创建线程来处理请求，而是通过处理模型直接对数据查询请求进行处理，能够避免在处理数据查询请求时创建相应的线程，从而加快数据查询的速度。

附图说明

图1是本申请一个示例性实施例提供的图数据库的属性图的示意图；

图2是本申请一个示例性实施例提供的基于B树的数据结构的示意图；

图3是本申请一个示例性实施例提供的计算机系统的结构示意图；

图4是本申请一个示例性实施例提供的的数据查询请求的处理过程的示意图；

图5是本申请一个示例性实施例提供的一种数据查询请求的处理方法的流程示意图；

图6是本申请一个示例性实施例提供的另一种数据查询请求的处理方法的流程示意图；

图7是本申请一个示例性实施例提供的全局一致性哈希环的示意图；

图8是本申请一个示例性实施例提供的本地一致性哈希环的示意图；

图9是本申请一个示例性实施例提供的顶点在图数据库中的存储结构的示意图；

图10是本申请一个示例性实施例提供的的数据查询请求的处理装置的结构示意图；

图11是本申请一个示例性实施例提供的计算机设备的结构示意图。

具体实施方式

首先，对本申请涉及的名词进行介绍。

图数据库：是一个使用图结构进行语义查询的数据库。图数据库通过顶点、边和属性来表示和存储数据。示例地，图1是本申请一个示例性实施例提供的图数据库的属性图。如图1所示，图数据库的属性图由顶点（圆圈）、边（箭头）、属性组成。图1中三个顶点中每个顶点有一个学生的标签，学生具备的属性包括姓名，属性的值分别是用户1、用户2、用户3。顶点之间的边表示了顶点之间的关系，例如哥哥、弟弟、见过。如果两个顶点之间的边是双向的，说明两个顶点的数据之间有关联关系，如果两个顶点之间的边是单向的，说明两个顶点的数据有单向的关联关系。

B树（Balance tree）：是一种自平衡的树状的数据结构，能够保持数据有序。基于B树的数据结构能够让查找数据、顺序访问、插入数据及删除数据的动作，都在对数时间内完成，即随着数据量增加，处理数据消耗的时间呈现对数式增长而非指数式增长，具有较高的数据处理效率。

示例地，图2是本申请一个示例性实施例提供的基于B树的数据结构的示意图。如图2所示，该B树中包括7个节点，每个节点由数据的索引组成。其中节点（6）称为根节点，

节点 (4) 和 (11) 称为中间节点, 节点 (1、2、3)、(5)、(7、8、9)、(12、13) 称为叶子节点。

B+树是针对于 B 树的改进, 与 B 树的区别在于其叶子节点中包括了全部数据的索引。B+树这种数据结构具有较好的查询性能, 以及在图表示中具有清楚表示的特点, 因而广泛的被应用在图数据库中。然而对树 (B 树、B+树) 这种数据结构下的数据的并发写 (同时写入多个数据的数据索引) 的处理比较麻烦, 原因在于在写数据的过程中会涉及节点的分裂和合并, 因此一般在对 B 树/B+树进行并发写操作的时候, 会对其进行不同粒度的加锁处理来实现顺序单写 B 树/B+树。例如, 对全局整棵树进行加锁来实现写操作, 目前也能够通过更细粒度的加锁方式来应对高并发的写情况。

继续参照图 2, 在 B 树中写入索引 (10) 时, 会将节点 (7、8、9) 进行分裂, 同时将索引 (8) 合并入节点 (11), 形成节点 (8、11)。在并行写入多个数据索引的情况下, 需要对 B 树进行加锁, 从而采用顺序单写的方式依次处理写入请求。但这样会导致数据处理的效率较低。

参与者模型 (Actor model): 是一个并发计算的数学模型, 参与者模型可以针对需要执行的任务做出本地决策, 来作为其对接收到的消息 (任务) 的响应, 也能够创建更多参与者模型, 来发送更多消息, 并确定如何响应接收到的下一条消息。参与者模型之间直接发送消息, 不需要经过什么中介, 消息是异步发送和处理的。通过不同参与者模型之间的消息发送实现模块之间的通信和交互。参与者模型采用的哲学是一切都是参与者 (Actor), 这与一些面向对象编程语言应用的“任何事物都是一个对象”的哲学类似, 因此其消息发送更加符合面向对象的原始意图。参与者模型属于并发组件模型, 通过组件方式定义并发编程范式的高级阶段, 可以避免使用者直接接触多线程并发或线程池等基础概念。

云存储 (cloudstorage): 是在云计算概念上延伸和发展出来的一个新的概念, 分布式云存储系统 (以下简称存储系统) 是指通过集群应用、网格技术以及分布存储文件系统等功能, 将网络中大量各种不同类型的存储设备 (存储设备也称之为存储节点) 通过应用软件或应用接口集合起来协同工作, 共同对外提供数据存储和业务访问功能的一个存储系统。示例性的, 本申请实施例中的分布式数据存储系统为云存储系统。

目前, 存储系统的存储方法为: 创建逻辑卷, 在创建逻辑卷时, 就为每个逻辑卷分配物理存储空间, 该物理存储空间可能是某个存储设备或者某几个存储设备的磁盘组成。客户端在某一逻辑卷上存储数据, 也就是将数据存储于文件系统中, 文件系统将数据分成许多部分, 每一部分是一个对象, 对象不仅包含数据而且还包含数据标识 (IDentity, ID) 等额外的信息, 文件系统将每个对象分别写入该逻辑卷的物理存储空间, 且文件系统会记录每个对象的存储位置信息, 从而当客户端请求访问数据时, 文件系统能够根据每个对象的存储位置信息让客户端对数据进行访问。存储系统为逻辑卷分配物理存储空间的过程, 具体为: 按照对存储于逻辑卷的对象的容量估量 (该估量往往相对于实际要存储的对象的容量有很大余量) 和独立冗余磁盘阵列 (Redundant Array of Independent Disk, RAID) 的组别, 预先将物理存储空间划分成分条, 一个逻辑卷可以理解为一个分条, 从而为逻辑卷分配了物理存储空间。

数据库 (Database): 简而言之可视为电子化的文件柜——存储电子文件的处所, 用户可以对文件中的数据进行新增、查询、更新、删除等操作。所谓“数据库”是以一定方式储存在一起、能与多个用户共享、具有尽可能小的冗余度、与应用程序彼此独立的数据集合。示例性的, 本申请实施例中的数据库为图数据库, 数据结构采用 B 树或 B+树的结构。

数据库管理系统 (DataBase Management System, DBMS) 是为管理数据库而设计的电脑软件系统, 一般具有存储、截取、安全保障、备份等基础功能。数据库管理系统可以依据它所支持的数据库模型来作分类, 例如关系式、可扩展标记语言 (eXtensible Markup Language, XML); 或依据所支持的计算机类型来作分类, 例如服务器群集、移动电话; 或依据所用查询语言来作分类, 例如结构化查询语言 (Structured Query Language, SQL)、用于 XML 数据

查询的语言 (XML Query, X Query); 或依据性能冲量重点来作分类, 例如最大规模、最高运行速度; 亦或其他分类方式。不论使用哪种分类方式, 一些 DBMS 能够跨类别, 例如, 同时支持多种查询语言。

图 3 是本申请一个示例性实施例提供的计算机系统的结构示意图。如图 3 所示, 该计算机系统 300 中包括第一节点设备 301、第二节点设备 302 和第三节点设备 303。

第二节点设备 302 可以是一台服务器, 或者由若干台服务器组成的服务器集群, 或者是一个云计算服务中心中的虚拟服务器等。可选地, 该第二节点设备 302 是分布式数据存储系统中用于调度数据处理请求的节点设备。例如, 第二节点设备 302 接收外部的数据查询请求, 根据数据查询请求所要查询的数据, 将数据查询请求调度至存储有需要查询的数据的节点设备或存储有与需要查询的数据相关的数据的节点设备进行处理。第二节点设备 302 和第一节点设备 301、第三节点设备 303 之间可以通过有线网络或无线网络建立连接。

第一节点设备 301 和第三节点设备 303 可以是一台服务器, 或者由若干台服务器组成的服务器集群, 或者是一个云计算服务中心中的虚拟服务器等。第一节点设备 301 和第三节点设备 303 中存储有相同、不同或部分相同的数据分片, 从而实现数据的分布式存储。在不同节点设备中存储有相同数据的情况下, 即存储数据副本的情况下, 对于数据副本一致性的问题, 可以采用无冲突复制数据类型 (Conflict-free Replicated Data Type, CRDT) 将一致性的问题转换为计算问题, 通过该方式可减少在并发修改数据的过程中的加锁问题。第一节点设备 301 和第三节点设备 303 之间可以通过有线网络或无线网络建立连接。

需要说明的是, 上述计算机系统 300 中的节点设备的数量仅用作示意, 不作为对本申请实施例提供的计算机系统 300 的限制。

图 4 是本申请一个示例性实施例提供的数据查询请求的处理过程的示意图。如图 4 所示, 第二节点设备是基于图数据库的分布式数据存储系统中的节点设备, 第二节点设备在接收到数据查询请求 401 后, 会将数据查询请求 401 发送至第一节点设备 402 的第一处理模型 4021。其中, 该数据查询请求用于查询与图数据库中的第一顶点相关的数据, 该数据查询请求携带有起始顶点的标识, 第一节点设备 402 存储有该起始顶点的数据。第一顶点也称为目标顶点, 目标顶点是用于处理数据查询请求的顶点。还可以描述为, 目标顶点是用于提供数据查询请求的处理结果的顶点。

第一节点设备 402 通过第一处理模型 4021, 会向第一节点设备 402 中的第二处理模型 4022 发送该数据查询请求 401。该第二处理模型 4022 是第一节点设备 402 中用于处理起始顶点的数据的处理模型。第一节点设备 402 会通过第二处理模型 4022, 基于该数据查询请求 401 来确定第三处理模型。该第三处理模型用于处理第一顶点或中间顶点的数据, 该中间顶点是起始顶点和第一顶点之间的顶点。

可选地, 在与第一节点设备相连的顶点中存在第一顶点的情况下, 第一节点设备 402 通过第二处理模型 4022 能够直接确定第一顶点, 从而确定用于处理第一顶点的数据的第三处理模型。在与第一节点设备相连的顶点中不存在第一顶点的情况下, 第一节点设备 402 会通过第二处理模型 4022 确定与第一顶点相连的中间顶点, 从而确定用于处理中间顶点的数据的第三处理模型。在确定第三处理模型后, 第一节点设备 402 会通过第二处理模型 4022 向第三处理模型发送该数据查询请求。在第三处理模型用于处理第一顶点的数据的情况下, 第三处理模型用于处理数据查询请求并向第二节点设备发送数据查询请求的处理结果。在第三处理模型用于处理中间顶点的数据的情况下, 第三处理模型用于通过上述方法再继续确定第四处理模型, 以及继续发送数据查询请求直至得到数据查询请求的处理结果。

可选地, 该第三处理模型属于第一节点设备 402 或第三节点设备 403。在第三处理模型属于第一节点设备 402 的情况下, 第一节点设备 402 会通过第二处理模型 4022 直接向第三处

理模型发送数据查询请求。在第三处理模型属于第三节点设备 403 的情况下，第一节点设备 402 会通过第二处理模型 4022 向第三节点设备 403 中用于调度数据查询请求的处理模型发送数据查询请求，之后该处理模型会向第三处理模型发送该数据查询请求。

在通过上述方式确定用于处理第一顶点的数据的目标处理模型（比如第三处理模型）后，目标处理模型会处理数据查询请求，并向其所在节点设备中用于调度数据查询请求的处理模型发送数据查询请求的处理结果。之后该用于调度数据查询请求的处理模型会向第二节点设备发送数据查询请求的处理结果。在存在多个处理模型向第二节点设备发送数据查询请求的处理结果的情况下，第二节点设备会将全部处理模型发送的数据查询请求的处理结果进行汇总处理，从而最终得到数据查询请求的处理结果。可选地，上述处理模型为参与者模型（Actor model）。其中，上述“多个”即是“至少两个”，可以理解为两个及两个以上。

需要说明的是，本申请实施例提供的方法，可应用于涉及分布式数据存储系统的场景、涉及大数据处理的场景、涉及机器学习模型相关的训练数据的处理场景、涉及应用程序（包括社交、短视频、即时通讯、游戏、电商、金融等领域的应用程序）中的数据的处理场景、以及涉及数据搜索的场景（例如商品搜索场景）等。

示例地，在本申请实施例提供的方法应用于社交类应用程序的数据处理场景的情况下，图数据库中的顶点表示某一用户帐号相关的数据，顶点之间的边表示用户帐号之间存在关联关系，边的属性用于反映该关联关系的属性，例如好友关系、拉黑关系、屏蔽关系等。节点设备为社交类应用程序的后台服务器，处理模型为后台服务器中的参与者模型。示例地，起始节点为第一用户帐号对应的节点，目标节点为第一用户帐号的好友帐号中，与第一用户帐号建立好友关系时间最长的第二用户帐号对应的节点。通过本申请实施例提供的方法，能够实现在第一用户帐号的好友帐号中，查询满足建立好友关系时间最长这一条件的第二用户帐号，从而可查询到第二用户帐号相关的数据。

示例地，在本申请实施例提供的方法应用于短视频类应用程序的数据处理场景的情况下，图数据库中的顶点表示某一用户帐号相关的数据，顶点之间的边表示用户帐号之间存在关联关系，边的属性用于反映该关联关系的属性，例如关注关系、点赞关系、收藏关系等。节点设备为短视频类应用程序的后台服务器，处理模型为后台服务器中的参与者模型。示例地，起始节点为第三用户帐号对应的节点，目标节点为第三用户帐号关注的用户帐号中，同样也关注了第三用户帐号的第四用户帐号对应的节点。通过本申请实施例提供的方法，能够实现在第三用户帐号关注的用户帐号中，查询满足关注了第三用户帐号这一条件的第四用户帐号，从而可查询到第四用户帐号相关的数据。

示例地，在本申请实施例提供的方法应用于游戏类应用程序的数据处理场景的情况下，图数据库中的顶点表示某一用户帐号相关的数据，顶点之间的边表示用户帐号之间存在的交互行为，边的属性用于反映该交互行为的属性，例如反映用户帐号之间存在交易虚拟物品的行为。节点设备为游戏类应用程序的后台服务器，处理模型为后台服务器中的参与者模型。示例地，起始节点为第五用户帐号对应的节点，目标节点为与第五用户帐号存在交易虚拟物品行为的用户帐号中，交易的虚拟物品的价格最高的第六用户帐号。通过本申请实施例提供的方法，能够实现在与第五用户帐号存在交易虚拟物品行为的用户帐号中，查询满足交易的虚拟物品的价格最高这一条件的第六用户帐号，从而可查询到第六用户帐号相关的数据。

可选地，本申请实施例提供的方法还可应用于云技术、人工智能、车联网、智慧交通、辅助驾驶等各种场景，用于处理上述场景中涉及的相关数据。在应用于车联网相关的场景中时，上述节点设备还能够为车载设备，车载设备可以是车载终端。

节点设备在接收到数据查询请求后，通过节点设备中的处理模型能够直接对数据查询请求进行处理，包括确定用于处理（图数据库中数据查询请求相关的）顶点的数据的处理模型，或确定数据查询请求的处理结果。在处理数据查询请求的过程中，无需创建线程来处理请求，而是通过处理模型直接对数据查询请求进行处理，能够避免在处理数据查询请求时创

建相应的线程，从而加快数据查询的速度。

图 5 是本申请一个示例性实施例提供的一种数据查询请求的处理方法的流程示意图。该方法可应用于如图 3 所示的系统中的第一节点设备，第一节点设备是基于图数据库的分布式数据存储系统中的多个节点设备中的一个。如图 5 所示，该方法包括以下步骤。

步骤 502：第一节点设备通过第一节点设备中的第一处理模型接收第二节点设备发送的数据查询请求。

该第二节点设备是分布式数据存储系统中用于调度数据查询请求的节点设备，该第二节点设备还可用于调度数据写入请求、数据删除请求和数据修改请求等数据处理请求。

该数据查询请求用于查询与图数据库中的第一顶点相关的数据，该数据查询请求携带有起始顶点的标识。可选地，该数据查询请求用于基于起始顶点查询第一顶点、基于起始顶点查询第一顶点的数据、查询起始顶点至第一顶点之间的路径（图数据库中的边）、查询起始顶点与第一顶点之间的边或边的属性、查询图数据库中包括起始顶点以及第一顶点的子图。可选地，该数据查询请求还能够携带第一顶点的标识，例如数据查询请求用于查询起始顶点和第一顶点之间的路径。

示例性地，该数据查询请求用于查询起始顶点的二度邻居的数据，即查询与中间顶点相连的顶点的数据。其中，该中间顶点与起始顶点相连，且该中间顶点与第一顶点相连。在该情况下，存在一个或多个中间顶点，存在一个或多个第一顶点。

该第一节点设备存储有起始顶点的数据。第二节点设备在接收到数据查询请求后，会确定存储有起始顶点的数据的节点设备，并向该节点设备中的处理模型发送该数据查询请求。

需要说明的是，节点设备中至少包括以下两类处理模型：用于任务调度的处理模型，比如用于调度数据查询请求、数据处理请求的处理模型；用于管理数据分片的处理模型，一个处理模型可以管理一个或多个数据分片，比如第一节点设备中可以包括用于管理顶点 1 的数据分片的处理模型、以及管理顶点 2 和顶点 3 的数据分片的处理模型。

该第一处理模型是第一节点设备中用于调度数据查询请求（数据处理请求）的模型，即第一处理模型属于用于任务调度的处理模型。第一处理模型在接收到数据查询请求后，会向第一节点设备中用于处理起始顶点的数据的处理模型发送该数据查询请求。可选地，该第一处理模型不具有需要负责处理的顶点的数据，分布式数据存储系统中存储有顶点的数据的节点设备中，均存在与该第一处理模型功能相同的处理模型。可选地，第二节点设备中存储有不同节点设备的用于调度数据查询请求的处理模型的地址，第二节点设备在确定第一节点设备后，会向第一处理模型对应的地址发送该数据查询请求。

可选地，该起始顶点包括多个，在该情况下，第二节点设备会按照起始顶点对数据查询请求进行拆分，确定每个数据查询请求对应的节点设备，向确定的节点设备中相应的处理模型发送其对应的数据查询请求。

步骤 504：第一节点设备通过第一处理模型，向第一节点设备中的第二处理模型发送数据查询请求。

该第二处理模型是第一节点设备中用于处理起始顶点的数据的处理模型，即第二处理模型是属于用于管理数据分片的处理模型。可选地，第一节点设备存储的多个顶点的各自的数据分片均具有其对应的处理模型。第一处理模型在接收到数据查询请求后，会根据起始顶点的标识，在第一节点设备中确定用于处理起始顶点的数据的处理模型，从而确定出第二处理模型。

可选地，第一节点设备通过第一处理模型，能够根据本地一致性哈希环确定第二处理模型，之后向第二处理模型发送数据查询请求。该本地一致性哈希环用于反映第一节点设备存储的每个顶点的数据分片与第一节点设备中的处理模型之间的对应关系。可选地，在起始顶点为多个的情况下，第一节点设备也会通过第一处理模型对数据查询请求进行拆分之后再进

行发送。

可选地，第一处理模型的本地存储有包括第一节点设备中的各处理模型的地址的元数据，第一节点设备通过该第一处理模型，根据该元数据从而向确定的第二处理模型发送该数据查询请求。

步骤 506：第一节点设备通过第二处理模型，基于数据查询请求确定第三处理模型。

该第三处理模型用于处理第一顶点或中间顶点的数据，该中间顶点是起始顶点和第一顶点之间的顶点，即第三处理模型属于用于管理数据分片的处理模型。例如，在起始顶点和第一顶点之间相连且满足数据查询请求的查询条件的情况下，第三处理模型用于处理第一顶点的数据。在起始顶点和第一顶点之间还存在中间顶点的情况下，第三处理模型用于处理中间顶点的数据。可选地，该中间顶点与起始顶点相连（在图数据库中连接中间顶点和起始顶点的边）。

可选地，在第二处理模型处理数据查询请求，从而得到数据查询请求的部分查询结果的情况下，例如查询起始顶点至第一顶点的路径，第一节点设备会通过第二处理模型向第一处理模型发送其确定的查询结果，之后第一节点设备会通过第一处理模型向第二节点设备发送该查询结果，并由第二节点设备对接收到的数据查询请求的查询结果进行汇总处理。

该第三处理模型为第一节点设备中的处理模型，或者为第三节点设备中的处理模型。

步骤 508：第一节点设备通过第二处理模型向第三处理模型发送数据查询请求。

在起始顶点和第一顶点之间相连且满足数据查询请求的查询条件的情况下，即第三处理模型用于处理第一顶点的数据的情况下，第三处理模型用于处理数据查询请求并向第二节点设备发送数据查询请求的处理结果。在起始顶点和第一顶点之间存在中间顶点的情况下，第三处理模型用于按照上述方法继续确定第四处理模型，并向第四处理模型继续发送数据查询请求，之后第四处理模型会按照上述方法继续对数据查询请求进行处理，直至最终得到数据查询请求的处理结果。第四处理模型是中间顶点中用于调度数据查询请求（数据处理请求）的模型，即第四处理模型属于用于任务调度的处理模型。

可选地，第二处理模型的本地存储有包括用于处理与起始顶点相连的顶点的数据的处理模型的地址的元数据，第一节点设备通过该第二处理模型，根据该元数据从而向第三处理模型发送该数据查询请求。

综上所述，本实施例提供的方法，节点设备在接收到数据查询请求后，通过节点设备中的处理模型能够直接对数据查询请求进行处理，从而确定用于处理图数据库中数据查询请求相关的顶点的数据的处理模型，进而可实现确定数据查询请求的处理结果。在处理数据查询请求的过程中，无需创建线程来处理请求，而是通过处理模型直接对数据查询请求进行处理，能够避免在处理数据查询请求时创建相应的线程，从而加快数据查询的速度。

图 6 是本申请一个示例性实施例提供的另一种数据查询请求的处理方法的流程示意图。该方法可应用于如图 3 所示的系统中的第一节点设备，第一节点设备是基于图数据库的分布式数据存储系统中的多个节点设备中的一个。如图 6 所示，该方法包括以下步骤。

步骤 602：第一节点设备通过第一节点设备中的第一处理模型接收第二节点设备发送的数据查询请求。

该第二节点设备是分布式数据存储系统中用于调度数据查询请求的节点设备。该数据查询请求用于查询与图数据库中的第一顶点相关的数据，数据查询请求携带有起始顶点的标识。

第二节点设备在接收到数据查询请求后，会确定存储有起始顶点的数据的节点设备，并向该节点设备中的处理模型发送该数据查询请求。该第一节点设备存储有起始顶点的的数据。可选地，分布式数据存储系统的各节点设备存储有图数据库中的至少一个顶点的数据分片，节点设备存储的顶点的数据分片与节点设备之间的对应关系是通过全局一致性哈希环（Global Consistence Hash Ring）建立的。例如，节点设备 1 中存储有图数据库中的顶点 1 的

数据分片，节点设备2中存储有图数据库中的顶点2和顶点3的数据分片，节点设备3中存储有图数据库中的顶点4的数据分片。上述节点设备与图数据库中的顶点的对应关系可通过全局一致性哈希环建立。

示例性的，通过对节点设备的标识进行取模运算，从而能够得到一个固定的值。以及，通过对图数据库中的各顶点的标识进行取模运算，同样能够得到一个固定的值。例如，均使用 2^{32} 进行取模运算。根据节点设备的标识或顶点的标识对 2^{32} 进行取模运算的结果值，能够组织形成一个圆环，这个圆环可被称为全局一致性哈希环。类比如钟表，钟表的圆可以理解为由60个点组成的圆，而对于全局一致性哈希环，可将其想象成由 2^{32} 个点组成的圆。上述取模运算得到的数值，即为不同顶点或节点设备在全局一致性哈希环上的位置。之后，可基于顶点在全局一致性哈希环上的位置，按照顺时针方向在全局一致性哈希环上寻找节点设备，并将找到的第一个节点设备确定为用于存储该顶点的数据分片的节点设备。需要说明的是，取模运算使用的 2^{32} 仅用作示例，也可以使用其它数值。“ $\wedge$ ”表示乘方，比如“ 2^{32} ”即是2的23次方。

第二节点设备在接收到数据查询请求后，会根据起始顶点的标识在全局一致性哈希环上确定第二位置，之后根据第二位置以及分布式数据存储系统的各节点设备在该全局一致性哈希环上的位置，能够确定数据查询请求对应的该第一节点设备。可选地，第二节点设备中不存储图数据库中的顶点的数据。

示例地，图7是本申请一个示例性实施例提供的全局一致性哈希环的示意图。如图7所示，第二节点设备在接收到数据查询请求后，会根据起始顶点的标识计算哈希值，从而根据该哈希值在全局一致性哈希环上确定第二位置。例如，第二位置为顶点701的位置，之后在全局一致性哈希环上顺时针寻找第一个节点设备，即节点设备703，并将节点设备703确定为第一节点设备。节点设备在全局一致性哈希环上的位置是通过计算节点设备的标识的哈希值确定的。或者，第二位置为顶点702的位置，在该情况下，第二节点设备在全局一致性哈希环上顺时针寻找第一个节点设备，即节点设备704，此时第二节点设备会将节点设备704确定为第一节点设备。

该第一处理模型是第一节点设备中用于调度数据查询请求的模型，第一处理模型在接收到数据查询请求后，会向第一节点设备中用于处理起始顶点的数据的处理模型发送该数据查询请求。

步骤604：第一节点设备通过第一处理模型，确定第一节点设备中的第二处理模型。

第二处理模型是第一节点设备中用于处理起始顶点的数据的处理模型。可选地，第一节点设备存储的多个顶点的各自的数据分片均具有其对应的处理模型。第一处理模型在接收到数据查询请求后，会根据起始顶点的标识，在第一节点设备中确定用于处理起始顶点的数据的处理模型，从而确定出第二处理模型。

可选地，第一节点设备存储有图数据库中的多个顶点各自的数据分片，每个数据分片与第一节点设备中的处理模型之间的对应关系是通过本地一致性哈希环（Local Consistence Hash Ring）建立的。第一节点设备通过第一处理模型，根据起始顶点的标识能够在本地一致性哈希环上确定第一位置。之后第一节点设备通过第一处理模型，根据第一位置以及第一节点设备中的各处理模型在本地一致性哈希环上的位置，能够确定数据查询请求对应的第二处理模型。

示例地，图8是本申请一个示例性实施例提供的本地一致性哈希环的示意图。如图8所示，第一处理模型在接收到数据查询请求后，第一节点设备会通过第一处理模型根据起始顶点的标识计算哈希值，从而根据该哈希值在本地一致性哈希环上确定第一位置。例如，第一位置为顶点801的位置，之后在本地一致性哈希环上顺时针寻找第一个处理模型，即处理模型803，并将处理模型803确定为第二处理模型。处理模型在本地一致性哈希环上的位置是通过计算处理模型的标识的哈希值确定的。或者，第一位置为顶点802的位置，在该情况下，

第一节点设备通过第一处理模型在本地一致性哈希环上顺时针寻找第一个处理模型，即处理模型 804，此时第一节点设备会通过第一处理模型将处理模型 804 确定为第二处理模型。

步骤 606：第一节点设备通过第一处理模型，向第一节点设备中的第二处理模型发送数据查询请求。

可选地，第一节点设备通过第一处理模型，向第二处理模型的消息队列发送数据查询请求。其中，第二处理模型的消息队列用于存储第二处理模型的待处理任务。可选地，第二处理模型会对其消息队列中的待处理任务依次进行处理，直至处理完成全部的待处理任务。

步骤 608：第一节点设备通过第二处理模型，根据起始顶点在图数据库中的存储结构以及数据查询请求中的查询条件，确定用于处理第一顶点的数据的第三处理模型。

起始顶点在图数据库中的存储结构包括起始顶点的属性、与起始顶点具有连接关系的顶点的标识以及连接关系的属性。可选地，与起始顶点具有连接关系的顶点可分为与起始顶点连接，边指向起始顶点，以及与起始顶点连接，边指向与起始顶点连接的顶点。该连接关系的属性包括不同方向的边的属性。

可选地，每个处理模型均存储有与其对应的顶点相关的资源，即存储有其负责处理的顶点的存储结构。例如，处理模型 1 用于处理顶点 1 和顶点 4 的数据，处理模型 2 用于处理顶点 2 的数据，处理模型 3 用于处理顶点 3 的数据。处理模型 1 存储有顶点 1 的存储结构以及顶点 4 的存储结构，处理模型 2 存储有顶点 2 的存储结构，处理模型 3 存储有顶点 3 的存储结构。

示例地，图 9 是本申请一个示例性实施例提供的顶点在图数据库中的存储结构的示意图。如图 9 的 (a) 所示，在图数据库中，顶点 1 与顶点 2、顶点 3 和顶点 4 分别相连。其中，顶点 1 和顶点 2 之间的边指向顶点 1，顶点 1 和顶点 3 之间的边分别指向顶点 1 和顶点 3，顶点 1 和顶点 4 之间的边指向顶点 4。顶点 2 和顶点 3 相连，顶点 2 和顶点 3 之间的边指向顶点 2。顶点 2 和顶点 4 相连，顶点 2 和顶点 4 之间的边指向顶点 4。顶点 3 和顶点 4 相连，顶点 3 和顶点 4 之间的边指向顶点 3。

如图 9 的 (b) 所示，处理模型 1 用于处理顶点 1 和顶点 4 的数据，处理模型 2 用于处理顶点 2 的数据，处理模型 3 用于处理顶点 3 的数据。处理模型 1 存储有顶点 1 和顶点 4 的存储结构，其中 Vp1 表示顶点 1 (V1) 的属性，V1_in 包括与顶点 1 连接的顶点中边指向顶点 1 的顶点的标识，以及该顶点与顶点 1 之间的边的属性，例如 Ep3 表示顶点 1 和顶点 3 之间指向顶点 1 的边的属性，Ep2 表示顶点 1 和顶点 2 之间指向顶点 1 的边的属性。V1_out 包括与顶点 1 连接的顶点中边指向该连接顶点的顶点标识，以及该顶点与顶点 1 之间的边的属性。Vp4 表示顶点 4 (V4) 的属性，V4_in 包括与顶点 4 连接的顶点中边指向顶点 4 的顶点的标识，V4_out 包括与顶点 4 连接的顶点中边指向该连接顶点的顶点标识，以及该顶点与顶点 4 之间的边的属性。处理模型 2 存储有顶点 2 的存储结构，其中 Vp2 表示顶点 2 (V2) 的属性，V2_in 包括与顶点 2 连接的顶点中边指向顶点 2 的顶点的标识，V2_out 包括与顶点 2 连接的顶点中边指向该连接顶点的顶点标识，以及该顶点与顶点 2 之间的边的属性。处理模型 3 存储有顶点 3 的存储结构，其中 Vp3 表示顶点 3 (V3) 的属性，V3_in 包括与顶点 3 连接的顶点中边指向顶点 3 的顶点的标识，V3_out 包括与顶点 3 连接的顶点中边指向该连接顶点的顶点标识，以及该顶点与顶点 3 之间的边的属性。

第一节点设备通过第二处理模型，根据起始顶点的属性、与起始顶点具有连接关系的顶点以及连接关系的属性中的一个或多个，在与起始顶点具有连接关系的顶点中查询满足查询条件的顶点。响应于与起始顶点具有连接关系的顶点中存在满足查询条件的顶点，第一节点设备通过第二处理模型将满足查询条件的顶点确定为第一顶点。之后第一节点设备通过第二处理模型，能够确定用于处理第一顶点的数据的第三处理模型。可选地，第二处理模型存储有用于处理与其相连的顶点的数据的处理模型的信息，在确定与其相连的第一顶点后，可根据该数据确定第三处理模型。

示例地，继续参照图 9，该数据查询请求用于查询顶点 1 的一度邻居中，边指向顶点 1 且边的属性满足查询条件的顶点的数据。第一节点设备通过第二处理模型，根据顶点 1 的存储结构，确定顶点 2 为第一顶点，之后可确定用于处理顶点 2 的数据的第三处理模型。

步骤 610：第一节点设备通过第二处理模型，根据起始顶点在图数据库中的存储结构以及数据查询请求中的查询条件，确定用于处理中间顶点的数据的第三处理模型。

该中间顶点是起始顶点和第一顶点之间的顶点。第一节点设备通过第二处理模型，按照步骤 608 中的方法查询起始顶点的存储结构时，响应于与起始顶点具有连接关系的顶点中不存在满足查询条件的顶点，第一节点设备通过第二处理模型会将与起始顶点具有连接关系的顶点确定为中间顶点，例如将全部与起始顶点具有连接关系的顶点确定为中间顶点。之后第一节点设备通过第二处理模型，确定用于处理中间顶点的数据的第三处理模型。可选地，第二处理模型存储有用于处理与其相连的顶点的数据的处理模型的信息，在确定与其相连的中间顶点后，可根据该数据确定第三处理模型。

示例地，继续参照图 9，该数据查询请求用于查询顶点 1 的二度邻居。若顶点 1 和顶点 4 未直接连接，则第一节点设备通过第二处理模型根据顶点 1 存储结构只能够查询到相连的顶点 2 和顶点 3，此时会将顶点 2 和顶点 3 确定为中间顶点，并将用于处理顶点 2 的数据的处理模型和用于处理顶点 3 的数据的处理模型均确定为第三处理模型。

步骤 612：第一节点设备通过第二处理模型向第三处理模型发送数据查询请求。

在第三处理模型属于第一节点设备的情况下，第一节点设备通过第二处理模型，向第一节点设备中的第三处理模型发送数据查询请求。

在第三处理模型属于第三节点设备的情况下，第一节点设备通过第二处理模型向第三节点设备中的第四处理模型发送数据查询请求，该第四处理模型是第三节点设备中用于调度数据查询请求的处理模型。其中，第三节点设备用于通过第四处理模型向第三处理模型发送数据查询请求，从而使得第三处理模型能够接收到该数据查询请求。

可选地，在跨节点设备发送数据查询请求的情况下，例如第一节点设备通过第二处理模型向第三节点设备中的第四处理模型发送数据查询请求的情况下，会基于远程过程调用（Remote Procedure Call, RPC）向第四处理模型发送数据查询请求。第一节点设备通过第二处理模型，会向第三处理模型（第四处理模型）的消息队列发送该数据查询请求。

在起始顶点和第一顶点之间相连且满足数据查询请求的查询条件的情况下，即第三处理模型用于处理第一顶点的数据的情况下，第三处理模型用于处理数据查询请求并向第二节点设备发送数据查询请求的处理结果。在起始顶点和第一顶点之间存在中间顶点的情况下，即第三处理模型用于处理中间顶点的数据的情况下，第三处理模型用于按照上述方法继续确定第四处理模型，并向第四处理模型继续发送数据查询请求，之后第四处理模型会按照上述方法继续对数据查询请求进行处理，直至最终得到数据查询请求的处理结果。

示例性地，在第三处理模型确定数据查询请求的处理结果的情况下，即第三处理模型用于处理第一顶点的数据的情况下，第三处理模型所在的节点设备用于通过第三处理模型，向第三处理模型所在的节点设备中用于调度数据查询请求的处理模型发送数据查询请求的处理结果。之后第三处理模型所在的节点设备还用于通过用于调度数据查询请求的处理模型，将数据查询请求的处理结果发送至第二节点设备。

可选地，存在多个处理模型向第二节点设备发送数据查询请求的处理结果，例如查询起始顶点至第一顶点之间的路径，需要查询出多个边并反馈至第二节点设备。在该情况下，第二节点设备用于将全部处理模型发送的数据查询请求的处理结果进行汇总处理，从而最终得到数据查询请求的处理结果。

可选地，第一节点设备的各处理器核心的多个线程与第一节点设备中的多个处理模型一一绑定，从而能够实现由每个线程独立运行处理模型，避免处理模型之间的资源抢占。分布式数据存储系统中的处理模型均为参与者模型（Actor model），例如上述第一处理模型、第二

处理模型、第三处理模型以及第四处理模型为参与者模型。可选地，第一节点设备包括多个用于存储数据的硬盘，硬盘与第二节点设备的处理器核心具有绑定关系。通过上述绑定关系，可使得每个参与者模型具有自己的独立的存储和计算资源，可以做到数据处理过程中的无锁处理。

需要说明的是，本申请实施例提供的方法，除了用于处理数据查询请求，还可用于处理数据处理请求。在处理数据处理请求时，第二节点设备会根据数据处理请求携带的顶点的标识，确定存储该顶点的数据的节点设备，并向该节点设备中用于调度数据处理请求（数据查询请求）的处理模型发送该数据处理请求。之后该处理模型会向节点设备中用于处理该顶点的数据的处理模型发送该数据处理请求，并由负责处理该顶点的数据的处理模型对该数据处理请求进行处理。在该场景下，通过将图数据库中顶点的数据在节点设备中进行分片存储，并通过节点设备中的不同处理模型负责不同顶点的数据的数据处理。能够实现在对不同顶点的数据进行并发处理时，无需对数据库加锁，而是由不同处理模型并行地进行不同顶点的数据处理，从而提升了数据处理的效率。在使用基于B树或B+树的数据结构的情况下，能够避免对树进行不同粒度的加锁，从而提升了数据处理的效率。

针对传统的数据修改同步加锁的问题，本申请实施例提供的方法根据现在多核处理器的特点，将数据分片进行了调整，对于每个数据分片由独立的参与者模型来管理，并且没有将参与者模型绑定在指定的中央处理器（Central Processing Unit, CPU）核心上，这样做可将传统每台节点设备上的逻辑分片转为了真正的物理分片，更细粒度的对数据进行了管理，更能够适应高并发修改的场景，同时这种细粒度的管理方式也使得在扩展性上能够更加灵活，特别适用于云上的扩展需求。

综上所述，本实施例提供的方法，节点设备在接收到数据查询请求后，通过节点设备中的处理模型能够直接对数据查询请求进行处理，从而确定用于处理图数据库中数据查询请求相关的顶点的数据的处理模型，进而可实现确定数据查询请求的处理结果。在处理数据查询请求的过程中，无需创建线程来处理请求，而是通过处理模型直接对数据查询请求进行处理，能够避免在处理数据查询请求时创建相应的线程，从而加快数据查询的速度。

本实施例提供的方法，还通过根据起始顶点在图数据库中的存储结构来确定第三处理模型，提供了一种便捷的确定第三处理模型的方式。

本实施例提供的方法，还通过在根据储存结构确定第一顶点的情况下，确定用于处理第一顶点的第三处理模型来处理数据查询请求，可以实现快速将数据查询请求分配至相应的处理模型进行处理。

本实施例提供的方法，还通过在根据储存结构确定中间顶点的情况下，确定用于处理中间顶点的第三处理模型来处理数据查询请求，可以实现由第三处理模型根据数据查询请求继续对第一顶点进行查询。

本实施例提供的方法，还通过本地一致性哈希环来维护每个节点设备中每个处理模型负责的顶点，因此可以动态的增加或者减少每台节点设备上处理模型的数量，而不会影响处理模型和其对应的顶点的关系。

本实施例提供的方法，还通过向处理模型的消息队列发送数据查询请求，由于当处理模型的消息队列中有了消息后会马上处理请求，这样可以提高处理请求的效率。

本实施例提供的方法，还通过向不同节点设备中用于调度请求的处理模型发送数据查询请求，提供了一种调度数据查询请求的方式。

本实施例提供的方法，还通过向不同节点设备中的处理模型发送数据查询请求，提供了一种发送数据查询请求的方式。

本实施例提供的方法，还通过在跨节点设备传输时，通过RPC的方式进行传输，提供了一种跨节点设备传输数据查询请求的方式。

本实施例提供的方法，还通过同时在数据查询过程中，通过确定查询结果之后直接推送

(push) 查询结果的方式, 相比传统的逐级拉取 (pull) 查询结果的方式, 实现了高性能的查询。

本实施例提供的方法, 还通过将数据查询请求的处理结果进行汇总, 提供了一种高效确定数据查询结果的方式。

本实施例提供的方法, 还通过将处理模型与处理器核心的线程绑定, 通过增加或减少处理器核心, 可实现真正的处理器核心级别的扩缩处理模型的功能。

本实施例提供的方法, 还通过参与者模型处理数据查询请求, 提供了一种用于处理数据查询请求的模型。

需要说明的是, 本申请实施例提供的方法步骤的先后顺序可以进行适当调整, 步骤也可以根据情况进行相应增减, 任何熟悉本技术领域的人员在本申请揭露的技术范围内, 可轻易想到变化的方法, 都应涵盖在本申请的保护范围之内, 因此不再赘述。

在一个具体的例子中, 继续参照图 4, 数据查询请求用于查询顶点 m ($id=\$id$) 的二度邻居, 该数据查询请求可表示为 $Match(m)-[*2]->(n) \text{ where } id(m)=\$id \text{ return } n$ 。当用户将数据查询请求发送至分布式数据存储系统的查询层 (即第二节点设备) 的时候, 查询层会根据全局数据分片配置 (全局一致性哈希环) 将该数据查询请求发送到存储有顶点 m 的数据的节点设备上。当该节点设备收到数据查询请求的时候, 会通过用于调度的参与者模型根据本地的一个数据分片规则 (本地一致性哈希环) 将该数据查询请求发送到负责处理顶点 m 的数据的参与者模型的消息队列中, 该参与者模型接收到消息 (数据查询请求) 后会对消息进行解析, 从而会根据消息指令执行相应的指令。例如在该情况下, 该参与者模型会获取顶点 m 的所有一度邻居, 然后再通过调用调度顶点 (DispatchVertex) 函数将消息分派到一度邻居相关的参与者模型上, 同时将该参与者模型查到的涉及二度邻居的数据发送到该节点设备中负责查询分配的参与者模型的消息队列中, 以向查询层反馈部分查询结果。同时其它接收到消息的参与者模型在接收到消息后, 会马上会根据消息中的信息在本地进行查询, 并将查询结果返回给其所述在节点设备中用于查询分配的参与者模型, 以向查询层反馈部分查询结果。对于跨节点设备发送消息的情况, 消息会传给节点设备上负责查询分配的参与者模型, 其会执行上述分配操作, 并最终将数据查询结果返回至查询层。上述查询过程可以类比于: $Map(Actors[DestList()], GetDestList())$, 其语义是将获取目标列表 (GetDestList()) 方法, 同时应用到相关的参与者模型上, 然后对所有返回数据进行汇总处理, 从而最终得到数据查询请求的处理结果。

图 10 是本申请一个示例性实施例提供的数据查询请求的处理装置的结构示意图, 该装置是基于图数据库的分布式数据存储系统中的多个节点设备中的一个。如图 10 所示, 该装置包括以下模块。

接收模块 1001, 用于通过所述装置中的第一处理模型接收第二节点设备发送的数据查询请求, 所述数据查询请求用于查询与所述图数据库中的第一顶点相关的数据, 所述数据查询请求携带有起始顶点的标识, 所述装置存储有所述起始顶点的数据。

发送模块 1002, 用于通过所述第一处理模型, 向所述装置中的第二处理模型发送所述数据查询请求, 所述第二处理模型是所述装置中用于处理所述起始顶点的数据的处理模型。

确定模块 1003, 用于通过所述第二处理模型, 基于所述数据查询请求确定第三处理模型, 所述第三处理模型用于处理所述第一顶点或中间顶点的数据, 所述中间顶点是所述起始顶点和所述第一顶点之间的顶点。

所述发送模块 1002, 用于通过所述第二处理模型向所述第三处理模型发送所述数据查询请求。

在一个可选的设计中, 所述确定模块 1003, 用于:

通过所述第二处理模型, 根据所述起始顶点在所述图数据库中的存储结构以及所述数据

查询请求中的查询条件，确定所述第三处理模型。

在一个可选的设计中，所述起始顶点在所述图数据库中的存储结构包括所述起始顶点的属性、与所述起始顶点具有连接关系的顶点的标识以及所述连接关系的属性；所述确定模块 1003，用于：

通过所述第二处理模型，根据所述起始顶点的属性、与所述起始顶点具有所述连接关系的顶点以及所述连接关系的属性中的一个或多个，在与所述起始顶点具有所述连接关系的顶点中查询满足所述查询条件的顶点；

响应于与所述起始顶点具有所述连接关系的顶点中存在满足所述查询条件的顶点，通过所述第二处理模型将满足所述查询条件的顶点确定为所述第一顶点；

通过所述第二处理模型，确定用于处理所述第一顶点的数据的所述第三处理模型。

在一个可选的设计中，所述起始顶点在所述图数据库中的存储结构包括所述起始顶点的属性、与所述起始顶点具有连接关系的顶点的标识以及所述连接关系的属性；所述确定模块 1003，用于：

通过所述第二处理模型，根据所述起始顶点的属性、与所述起始顶点具有所述连接关系的顶点以及所述连接关系的属性中的一个或多个，在与所述起始顶点具有所述连接关系的顶点中查询满足所述查询条件的顶点；

响应于与所述起始顶点具有所述连接关系的顶点中不存在满足所述查询条件的顶点，通过所述第二处理模型将与所述起始顶点具有所述连接关系的顶点确定为所述中间顶点；

通过所述第二处理模型，确定用于处理所述中间顶点的数据的所述第三处理模型。

在一个可选的设计中，所述装置存储有所述图数据库中的多个顶点各自的数据分片，每个所述数据分片与所述装置中的处理模型之间的对应关系是通过本地一致性哈希环建立的；所述确定模块 1003，用于：

通过所述第一处理模型，根据所述起始顶点的标识在所述本地一致性哈希环上确定第一位置；

通过所述第一处理模型，根据所述第一位置以及所述装置中的各处理模型在所述本地一致性哈希环上的位置，确定所述数据查询请求对应的所述第二处理模型。

在一个可选的设计中，所述发送模块 1002，用于：

通过所述第一处理模型，向所述第二处理模型的消息队列发送所述数据查询请求；

其中，所述第二处理模型的消息队列用于存储所述第二处理模型的待处理任务。

在一个可选的设计中，所述第三处理模型属于所述装置；所述发送模块 1002，用于：

通过所述第二处理模型，向所述装置中的所述第三处理模型发送所述数据查询请求。

在一个可选的设计中，所述第三处理模型属于第三节点设备；所述发送模块 1002，用于：

通过所述第二处理模型向所述第三节点设备中的第四处理模型发送所述数据查询请求，所述第四处理模型是所述第三节点设备中用于调度所述数据查询请求的处理模型；

其中，所述第三节点设备用于通过所述第四处理模型向所述第三处理模型发送所述数据查询请求。

在一个可选的设计中，所述发送模块 1002，用于：

通过所述第二处理模型，基于 RPC 向所述第四处理模型发送所述数据查询请求。

在一个可选的设计中，在所述第三处理模型确定所述数据查询请求的处理结果的情况下，所述第三处理模型所在的节点设备用于通过所述第三处理模型，向所述第三处理模型所在的节点设备中用于调度所述数据查询请求的处理模型发送所述数据查询请求的处理结果；

所述第三处理模型所在的节点设备，还用于通过所述第三处理模型所在的节点设备中用于调度所述数据查询请求的处理模型，将所述数据查询请求的处理结果发送至所述第二节点设备。

在一个可选的设计中，所述第三处理模型确定所述数据查询请求的处理结果；

在所述第三处理模型属于所述第一节点设备的情况下，所述第一节点设备用于通过所述第三处理模型，向所述第一处理模型发送所述数据查询请求的处理结果，所述第一处理模型是所述第一节点设备中用于调度所述数据查询请求的处理模型；所述第一节点设备还用于通过所述第三处理模型，将所述数据查询请求的处理结果发送至所述第二节点设备；

在所述第三处理模型属于第三节点设备的情况下，所述第三节点设备用于通过所述第三处理模型，向第四处理模型发送所述数据查询请求的处理结果，所述第四处理模型是所述第三节点设备中用于调度所述数据查询请求的处理模型；所述第三节点设备还用于通过所述第四处理模型，将所述数据查询请求的处理结果发送至所述第二节点设备。

在一个可选的设计中，存在多个处理模型向所述第二节点设备发送所述数据查询请求的处理结果；

所述第二节点设备用于将所述至少两个处理模型发送的所述数据查询请求的处理结果进行汇总处理。

在一个可选的设计中，所述装置的各处理器核心的多个线程与所述装置中的多个处理模型一一绑定。

在一个可选的设计中，所述第一处理模型、所述第二处理模型以及所述第三处理模型为参与者模型。

本申请的实施例还提供了一种计算机设备，该计算机设备包括：处理器和存储器，存储器中存储有至少一段程序，至少一段程序由处理器加载并执行以实现上述各方法实施例提供的的数据查询请求的处理方法。

可选地，该计算机设备为服务器。示例地，图 11 是本申请一个示例性实施例提供的计算机设备的结构示意图。

所述计算机设备 1100 包括中央处理单元（Central Processing Unit, CPU）1101、包括随机存取存储器（Random Access Memory, RAM）1102 和只读存储器（Read-Only Memory, ROM）1103 的系统存储器 1104，以及连接系统存储器 1104 和中央处理单元 1101 的系统总线 1105。所述计算机设备 1100 还包括帮助计算机设备内的各个器件之间传输信息的基本输入/输出系统（Input/Output 系统，I/O 系统）1106，和用于存储操作系统 1113、应用程序 1114 和其他程序模块 1115 的大容量存储设备 1107。

所述基本输入/输出系统 1106 包括有用于显示信息的显示器 1108 和用于用户输入信息的诸如鼠标、键盘之类的输入设备 1109。其中所述显示器 1108 和输入设备 1109 都通过连接到系统总线 1105 的输入输出控制器 1110 连接到中央处理单元 1101。所述基本输入/输出系统 1106 还可以包括输入输出控制器 1110 以用于接收和处理来自键盘、鼠标、或电子触控笔等多个其他设备的输入。类似地，输入输出控制器 1110 还提供输出到显示屏、打印机或其他类型的输出设备。

所述大容量存储设备 1107 通过连接到系统总线 1105 的大容量存储控制器（未示出）连接到中央处理单元 1101。所述大容量存储设备 1107 及其相关联的计算机可读存储介质为计算机设备 1100 提供非易失性存储。也就是说，所述大容量存储设备 1107 可以包括诸如硬盘或者只读光盘（Compact Disc Read-Only Memory, CD-ROM）驱动器之类的计算机可读存储介质（未示出）。

不失一般性，所述计算机可读存储介质可以包括计算机存储介质和通信介质。计算机存储介质包括以用于存储诸如计算机可读存储指令、数据结构、程序模块或其他数据等信息的任何方法或技术实现的易失性和非易失性、可移动和不可移动介质。计算机存储介质包括 RAM、ROM、可擦除可编程只读存储器（Erasable Programmable Read Only Memory, EPROM）、电子抹除式可复写只读存储器（Electrically-Erasable Programmable Read-Only Memory, EEPROM）、闪存或其他固态存储设备、CD-ROM、数字多功能光盘（Digital Versatile Disc,

DVD) 或其他光学存储、磁带盒、磁带、磁盘存储或其他磁性存储设备。当然, 本领域技术人员可知所述计算机存储介质不局限于上述几种。上述的系统存储器 1104 和大容量存储设备 1107 可以统称为存储器。

存储器存储有一个或多个程序, 一个或多个程序被配置成由一个或多个中央处理单元 1101 执行, 一个或多个程序包含用于实现上述方法实施例的指令, 中央处理单元 1101 执行该一个或多个程序实现上述各个方法实施例提供的方法。

根据本申请的各种实施例, 所述计算机设备 1100 还可以通过诸如因特网等网络连接到网络上的远程计算机设备运行。也即计算机设备 1100 可以通过连接在所述系统总线 1105 上的网络接口单元 1111 连接到网络 1112, 或者说, 也可以使用网络接口单元 1111 来连接到其他类型的网络或远程计算机设备系统(未示出)。

所述存储器还包括一个或者一个以上的程序, 所述一个或者一个以上程序存储于存储器中, 所述一个或者一个以上程序包含用于进行本申请实施例提供的方法中由计算机设备所执行的步骤。

本申请实施例中还提供了一种计算机可读存储介质, 该可读存储介质中存储有至少一段程序, 当该至少一段程序由计算机设备的处理器加载并执行时, 实现上述各方法实施例提供的的数据查询请求的处理方法。

本申请还提供了一种计算机程序产品或计算机程序, 该计算机程序产品或计算机程序包括计算机指令, 该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令, 处理器执行该计算机指令, 使得该计算机设备执行上述各方法实施例提供的的数据查询请求的处理方法。

权利要求书

1.一种数据查询请求的处理方法，所述方法应用于第一节点设备，所述第一节点设备是基于图数据库的分布式数据存储系统中的至少两个节点设备中的一个，所述方法包括：

所述第一节点设备通过所述第一节点设备中的第一处理模型，接收第二节点设备发送的数据查询请求，所述数据查询请求用于查询与所述图数据库中的第一顶点相关的数据，所述数据查询请求携带有起始顶点的标识，所述第一节点设备存储有所述起始顶点的数据；

所述第一节点设备通过所述第一处理模型，向所述第一节点设备中的第二处理模型发送所述数据查询请求，所述第二处理模型是所述第一节点设备中用于处理所述起始顶点的数据的处理模型；

所述第一节点设备通过所述第二处理模型，基于所述数据查询请求确定第三处理模型，所述第三处理模型用于处理所述第一顶点或中间顶点的数据，所述中间顶点是所述起始顶点和所述第一顶点之间的顶点；

所述第一节点设备通过所述第二处理模型向所述第三处理模型发送所述数据查询请求。

2.根据权利要求1所述的方法，所述第一节点设备通过所述第二处理模型，基于所述数据查询请求确定第三处理模型，包括：

所述第一节点设备通过所述第二处理模型，根据所述起始顶点在所述图数据库中的存储结构以及所述数据查询请求中的查询条件，确定所述第三处理模型。

3.根据权利要求2所述的方法，所述起始顶点在所述图数据库中的存储结构包括所述起始顶点的属性、与所述起始顶点具有连接关系的顶点的标识以及所述连接关系的属性；

所述第一节点设备通过所述第二处理模型，根据所述起始顶点在所述图数据库中的存储结构以及所述数据查询请求中的查询条件，确定所述第三处理模型，包括：

所述第一节点设备通过所述第二处理模型，根据所述起始顶点的属性、与所述起始顶点具有所述连接关系的顶点以及所述连接关系的属性中的一个或至少两个，在与所述起始顶点具有所述连接关系的顶点中查询满足所述查询条件的顶点；

响应于与所述起始顶点具有所述连接关系的顶点中存在满足所述查询条件的顶点，所述第一节点设备通过所述第二处理模型将满足所述查询条件的顶点确定为所述第一顶点；

所述第一节点设备通过所述第二处理模型，确定用于处理所述第一顶点的数据的所述第三处理模型。

4.根据权利要求2所述的方法，所述起始顶点在所述图数据库中的存储结构包括所述起始顶点的属性、与所述起始顶点具有连接关系的顶点的标识以及所述连接关系的属性；

所述第一节点设备通过所述第二处理模型，根据所述起始顶点在所述图数据库中的存储结构以及所述数据查询请求中的查询条件，确定所述第三处理模型，包括：

所述第一节点设备通过所述第二处理模型，根据所述起始顶点的属性、与所述起始顶点具有所述连接关系的顶点以及所述连接关系的属性中的一个或至少两个，在与所述起始顶点具有所述连接关系的顶点中查询满足所述查询条件的顶点；

响应于与所述起始顶点具有所述连接关系的顶点中不存在满足所述查询条件的顶点，所述第一节点设备通过所述第二处理模型将与所述起始顶点具有所述连接关系的顶点确定为所述中间顶点；

所述第一节点设备通过所述第二处理模型，确定用于处理所述中间顶点的数据的所述第三处理模型。

5.根据权利要求1至4任一所述的方法，所述第一节点设备存储有所述图数据库中的至少两个顶点各自的数据分片，每个所述数据分片与所述第一节点设备中的处理模型之间的对应关系是通过本地一致性哈希环建立的；

所述方法还包括：

所述第一节点设备通过所述第一处理模型，根据所述起始顶点的标识在所述本地一致性哈希环上确定第一位置；

所述第一节点设备通过所述第一处理模型，根据所述第一位置以及所述第一节点设备中的各处理模型在所述本地一致性哈希环上的位置，确定所述数据查询请求对应的所述第二处理模型。

6.根据权利要求1至4任一所述的方法，所述第一节点设备通过所述第一处理模型，向所述第一节点设备中的第二处理模型发送所述数据查询请求，包括：

所述第一节点设备通过所述第一处理模型，向所述第二处理模型的消息队列发送所述数据查询请求；

其中，所述第二处理模型的消息队列用于存储所述第二处理模型的待处理任务。

7.根据权利要求1至4任一所述的方法，所述第三处理模型属于所述第一节点设备；

所述第一节点设备通过所述第二处理模型向所述第三处理模型发送所述数据查询请求，包括：

所述第一节点设备通过所述第二处理模型，向所述第一节点设备中的所述第三处理模型发送所述数据查询请求。

8.根据权利要求1至4任一所述的方法，所述第三处理模型属于第三节点设备；

所述第一节点设备通过所述第二处理模型向所述第三处理模型发送所述数据查询请求，包括：

所述第一节点设备通过所述第二处理模型向所述第三节点设备中的第四处理模型发送所述数据查询请求，所述第四处理模型是所述第三节点设备中用于调度所述数据查询请求的处理模型；

其中，所述第三节点设备用于通过所述第四处理模型向所述第三处理模型发送所述数据查询请求。

9.根据权利要求8所述的方法，所述第一节点设备通过所述第二处理模型向所述第三节点设备中的第四处理模型发送所述数据查询请求，包括：

所述第一节点设备通过所述第二处理模型，基于远程过程调用向所述第四处理模型发送所述数据查询请求。

10.根据权利要求1至4任一所述的方法，所述第三处理模型确定所述数据查询请求的处理结果；

在所述第三处理模型属于所述第一节点设备的情况下，所述第一节点设备用于通过所述第三处理模型，向所述第一处理模型发送所述数据查询请求的处理结果，所述第一处理模型是所述第一节点设备中用于调度所述数据查询请求的处理模型；所述第一节点设备还用于通过所述第一处理模型，将所述数据查询请求的处理结果发送至所述第二节点设备；

在所述第三处理模型属于第三节点设备的情况下，所述第三节点设备用于通过所述第三处理模型，向第四处理模型发送所述数据查询请求的处理结果，所述第四处理模型是所述第三节点设备中用于调度所述数据查询请求的处理模型；所述第三节点设备还用于通过所述第四处理模型，将所述数据查询请求的处理结果发送至所述第二节点设备。

11.根据权利要求10所述的方法，存在至少两个处理模型向所述第二节点设备发送所述数据查询请求的处理结果；

所述第二节点设备用于将所述至少两个处理模型发送的所述数据查询请求的处理结果进行汇总处理。

12.根据权利要求1至4任一所述的方法，所述第一节点设备的各处理器核心的至少两个线程与所述第一节点设备中的至少两个处理模型一一绑定。

13.根据权利要求1至4任一所述的方法，所述第一处理模型、所述第二处理模型以及所述第三处理模型为参与者模型。

14.一种数据查询请求的处理装置，所述装置是基于图数据库的分布式数据存储系统中的至少两个节点设备中的一个，所述装置包括：

接收模块，用于通过所述装置中的第一处理模型接收第二节点设备发送的数据查询请求，所述数据查询请求用于查询与所述图数据库中的第一顶点相关的数据，所述数据查询请求携带有起始顶点的标识，所述装置存储有所述起始顶点的数据；

发送模块，用于通过所述第一处理模型，向所述装置中的第二处理模型发送所述数据查询请求，所述第二处理模型是所述装置中用于处理所述起始顶点的数据的处理模型；

确定模块，用于通过所述第二处理模型，基于所述数据查询请求确定第三处理模型，所述第三处理模型用于处理所述第一顶点或中间顶点的数据，所述中间顶点是所述起始顶点和所述第一顶点之间的顶点；

所述发送模块，用于通过所述第二处理模型向所述第三处理模型发送所述数据查询请求。

15.根据权利要求14所述的装置，

所述确定模块，用于通过所述第二处理模型，根据所述起始顶点在所述图数据库中的存储结构以及所述数据查询请求中的查询条件，确定所述第三处理模型。

16.根据权利要求15所述的装置，所述起始顶点在所述图数据库中的存储结构包括所述起始顶点的属性、与所述起始顶点具有连接关系的顶点的标识以及所述连接关系的属性；

所述确定模块，用于：

通过所述第二处理模型，根据所述起始顶点的属性、与所述起始顶点具有所述连接关系的顶点以及所述连接关系的属性中的一个或至少两个，在与所述起始顶点具有所述连接关系的顶点中查询满足所述查询条件的顶点；

响应于与所述起始顶点具有所述连接关系的顶点中存在满足所述查询条件的顶点，通过所述第二处理模型将满足所述查询条件的顶点确定为所述第一顶点；

通过所述第二处理模型，确定用于处理所述第一顶点的数据的所述第三处理模型。

17.根据权利要求15所述的装置，所述起始顶点在所述图数据库中的存储结构包括所述起始顶点的属性、与所述起始顶点具有连接关系的顶点的标识以及所述连接关系的属性；

所述确定模块，用于：

通过所述第二处理模型，根据所述起始顶点的属性、与所述起始顶点具有所述连接关系的顶点以及所述连接关系的属性中的一个或至少两个，在与所述起始顶点具有所述连接关系的顶点中查询满足所述查询条件的顶点；

响应于与所述起始顶点具有所述连接关系的顶点中不存在满足所述查询条件的顶点，通过所述第二处理模型将与所述起始顶点具有所述连接关系的顶点确定为所述中间顶点；

通过所述第二处理模型，确定用于处理所述中间顶点的数据的所述第三处理模型。

18.一种计算机设备，所述计算机设备包括处理器和存储器，所述存储器中存储有至少一段程序，所述至少一段程序由所述处理器加载并执行以实现如权利要求1至13任一所述的数据查询请求的处理方法。

19.一种计算机可读存储介质，所述可读存储介质中存储有至少一段程序，所述至少一段程序由处理器加载并执行以实现如权利要求1至13任一所述的数据查询请求的处理方法。

20.一种计算机程序产品，所述计算机程序产品包括计算机指令，所述计算机指令存储在计算机可读存储介质中；计算机设备的处理器从所述计算机可读存储介质读取所述计算机指令，所述处理器执行所述计算机指令，使得所述计算机设备执行如权利要求1至13任一所述的数据查询请求的处理方法。