计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(8):1−16 [doi: 10.15888/j.cnki.csa.008713]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

图数据库发展综述①

刘宇宁, 范冰冰

(华南师范大学 计算机学院, 广州 510631)
通信作者: 范冰冰, E-mail: fanbb1962@qq.com

摘　要: 随着数据关联关系的发现、管理和应用的深入, 图数据库快速发展. 归纳总结了图数据库概念、图模型、组成架构图和数据库的特点; 详细阐述了图数据库的关键技术; 分析比较了当前主流图数据库产品, 归纳了当前图数据库主要应用场景; 最后提出图数据库未来发展的趋势.

关键词: 图数据库; 图存储; 图查询语言

引用格式: 刘宇宁,范冰冰.图数据库发展综述.计算机系统应用,2022,31(8):1–16. http://www.c-s-a.org.cn/1003-3254/8713.html

Survey on Graph Database Development

LIU Yu-Ning, FAN Bing-Bing

(School of Computer Science, South China Normal University, Guangzhou 510631, China)

Abstract: With the discovery, management and application of data associations, graph databases develop rapidly. By analyzing the concept, model and structure of graph databases, this study summarizes the characteristics of graph databases. Regarding graph databases, the study expounds their three key technologies, compares existing products, and summarizes their current application scenarios. Finally, it proposes the trend of graph database development in the future.

Key words: graph database; graph storage; graph query language

数据库是计算机最广泛应用的技术, 传统的关系型数据库以“表格化结构”的方式, 对实际中的联系进行建模, 从上世纪起一直占据着数据库行业的主导地位[1]. 随着数据量的快速增长, 数据类型的进一步扩展, 部分研究指出, 非结构化数据占据了总数据量 85% 的比重[2]. 基于此, NoSQL 数据库凭借其无模式、可水平扩展的架构以及更为宽松的 BASE 原则等特点逐渐占据市场. 通常认为 NoSQL 数据库分为 4 种类型, 包括: key-value 数据库、列式数据库、文档数据库和图数据库. 随着各行业在数据分析及深度挖掘数据间的内在联系方面有了更多需求, 如金融行业尝试在海量的账目操作数据中进行欺诈行为检测、社交网络平台对用户之间关系进行深度关联分析等. 关系型数据库在处理关联数据时, 复杂关联情况会使得关系型数据库表格之间的“外键”连接操作增加, 这使得其查询复杂和高代价, 无法达到实际应用中快速响应的需求. 例如, 在探究社交网络多个用户之间关系时, SQL 语句的层级结构使用递归连接, 这导致了表格之间连接的高度复杂, 查询效率低下. 在社交关系领域, 社交网络可以看作是人与人之间密集关联的网状模型, 关系型数据库的表格模型或者 NoSQL 数据库的文档、key-value 存储方式都难以表达它的结构特点. 而采用图建模可以更好地表达其关系结构, 在进行实时关系查询时, 通过图算法能够更加高效. 以“图”为核心, 将“边”视为数据库的“一等公民”(即数据库中最基本、最核心的概念) 的图数据库在近十年的时间逐渐得到业界重视并快速发展[3].

根据中国信息通信研究院 2019 年 12 月发布的

① 基金项目: 广东省重大科技专项 (2016B030305003)
收稿时间: 2021-10-10; 修改时间: 2021-11-29, 2021-12-10, 2022-01-28; 采用时间: 2022-02-18; csa 在线出版时间: 2022-06-16

《图数据库白皮书》中信息, 图数据库的发展被划分为两个阶段[4]: Graph1.0 (2007–2010 年) 小规模原生图存储阶段, 图数据库主要以 Neo4j 的 1.0 版本为代表, 采用了原生图的底层存储方式, 在复杂关联数据的查询性能上相对于关系型数据库有了明显提高. 《图数据库》[5] 一书在 5 000 万点和边的数据规模下, 对比了 Neo4j 与关系型数据库在关联查询的时间对比, 随着关联关系深度的增加, 关系型数据库性能呈指数倍增长甚至无法运行. 该阶段图数据库产品在架构设计上支持了单机部署, 在产品性能和业务扩展能力方面都有限; Graph2.0 (2010 年至今) 分布式架构的发展, 使得更多的图数据库产品支持分布式大规模图存储. 如 JanusGraph 允许使用多种后端存储方式, OrientDB 采用了原生图存储并扩展了分布式存储模块. 通过支持硬件上的水平扩展, 分布式图数据库进一步提升了在海量数据中对关联数据的存储以及实时查询. 部分图数据库产品在全图计算分析等复杂场景下需要结合图处理引擎 (如 GraphX) 进行离线计算和分析[6]. 近年来, 发展势头迅速的 TigerGraph[7] 则将其产品定义为原生大规模并行图处理的第三代图数据库, 通过内置丰富的算法库和特有的数据存储结构, 进一步提升图数据库在复杂场景下的查询表现和可扩展性. 但目前业界对第三代图数据库产品还未有明确定义. 2019 年初, Gartner 数据与分析峰会上将图列为 2019 年十大数据和分析趋势之一, 并认为到 2022 年, 全球图处理及图数据库应用都将以每年 100% 的速度增长.

本文第 1 节对图数据库概念、组成架构、特点以及和关系型数据库对比 4 个部分进行介绍. 第 2 节对图查询语言、图计算以及图存储 3 种图数据库关键技术进行比较分析. 第 3 节对比当前图数据库主流产品间的特点, 并介绍了各领域应用情况. 第 4 节对全文进行了总结, 并探索了未来的研究方向.

1 图数据库概念以及特点

1.1 图数据库概念

图数据库管理系统 (以下简称图数据库) 是一种经过优化的用于存储、查询和更新图结构数据的数据库管理系统[5], 它支持对图数据模型的增、删、查、改 (CRUD) 方法.

图数据库以数学中的图论作为理论基础, 以图模型为特点进行数据存储, 主流的图模型有 3 种, 分别是属性图、RDF 和超图. 相对于 RDF 和超图属性图模型目前被图数据库业界广泛采用, 由图数据管理领域学术界和工业界成员共同组成的关联数据基准委员会 (linked data benchmark council, LDBC) 也正以属性图为基础对图数据模型和查询语言进行标准化, 本文所讨论的是以属性图为数据模型的图数据库研究.

属性图模型一般可以用以下四元组进行描述[6], 属性图表示为 G, 包含节点集合 V、属性集合 P、关系 (边) 集合 E, 标签集合 T 组成, $G=<V, E, P, T>$.

节点: 用于表示图中实体信息, 可以包含一个或多个属性, 节点之间使用关系建立连接.

关系: 关系用于连接节点, 可以有一个或多个属性 (存储为键值对 key-value), 节点之间可以有多个甚至递归的关系.

属性: 属性是命名值, 其中名称 (或键) 一般是字符串, 属性可以被索引和约束, 可以从多个属性创建复合索引.

标签: 标签用于将节点分组, 一般而言一个节点可以具有多个标签, 在图数据库中可以对标签进行索引, 用于提高查询效率.

1.2 图数据库组成架构

当前图数据库的组成架构如图 1 所示, 整体上采用了分层设计的模式[4–8], 由 3 层组成, 分别是: 接口层、计算层、存储层.

图 1 图数据库组成架构

(1) 接口层

位于架构中的顶层, 负责对外提供服务, 包含了用户使用图数据库所需要直接操作的接口及组件等, 提供了直观的数据展示方法和友好的交互模式, 有以下几种方式:

查询语言接口: 提供除该图数据库原有查询语言之外的, 如 Cypher[9]、Gremlin[9]、GSQL[10] 等主流图

查询语言接口.

API: 提供 ODBC、JDBC、RPC、RESTful 等接口与应用端进行交互.

SDK: 在 Python、Java、C++等主流编程语言中, 通过库函数的方式以调用图数据库的相关接口.

可视化组件: 通过图形化界面的形式展示数据模型并且实现和用户的交互操作.

(2) 计算层

作为中间层, 承担上下层数据传输的桥梁, 提供对操作的处理和计算. 一般而言, 计算层实现了以下内容:

语法解析: 对输入语法进行检查, 进行语法解析并转换成数据的具体操作内容.

查询引擎: 为语法解析后的内容提供查询等操作.

优化器: 对查询或者计算内容提供优化操作.

事务管理: 用于保证事务的原子性和可串行性.

任务调度: 对多项任务进行调度管理, 保证查询效率.

图算法: 图算法可能由图数据库产品自身实现, 也可能是提供图处理引擎接口实现.

(3) 存储层

图数据库以底层存储方式的不同分为原生和非原生两个类别, 存储层提供图数据结构、索引逻辑方面的管理.

1.3 图数据库特点

(1) 实际关系的建模方式

对于现实世界中的复杂实体关系, 图模型的存储和展示方式能够更加直接地进行表达, 这有利于使用者对数据有更直观的了解.

(2) 建模易扩展

图数据库提供了灵活的数据模式, 可以根据业务变化和场景需求, 对数据模型进行更改. 图数据库使用者无需在设计之初就把所有内容填充完毕, 在后续的使用中能够对数据模型进行扩展, 免去了冗余的标准化时间成本.

(3) 针对关联数据的快速查询

属性图模型提供了内置的索引数据结构和对应的图查询算法进行针对性优化, 它无需针对给定查询而加载或接触无关的数据, 防止局部数据查询引发的全局数据读取. 在处理深度关联数据时, 通过“点-边-点”的连接方式能够做到实时数据响应.

(4) 支持 ACID 事务完整性

图数据库在保证快速的数据访问更新同时实现了数据的一致性保持 , 支持 ACID 事务管理, 能够安全地进行数据管理.

1.4 图数据库对比关系型数据库

(1) 建模方式

关系型数据库以“表格化结构”的方式对实际中的联系建模, 对结构化数据的处理起了重要的作用. 但其严格的建模约束, 使其难以适应动态变化的数据结构关系, 在处理“联系”的具体问题上, 关系型数据库需要通过“外键”连接的方式对表格进行操作, 这使得其查询效率低下. 而图数据库采用“图建模”的方式, 能够更加贴切地表达现实世界中的实体及关系. 以社交图关系为例, 分别采用关系型数据库和图数据库建模方式进行建模, 如图 2, 图 3.

人员

编号	名称
1	用户 1
2	用户 2
3	用户 3
…	…
n	用户 *N*

关注

关注编号	被关注编号
1	2
2	3
3	5
…	…
n	*n*+1

图 2 关系型数据库社交关系建模

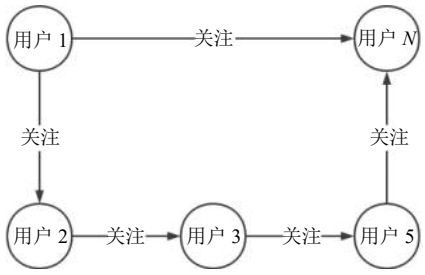

图 3 图数据库社交关系建模

可以发现, 针对于人与人密集关联的网状模型, 表格的方式并不能充分直观地表达其关联关系, 相反, 图建模则能够方便地表达这种网络结构.

(2) 查询效率

在查询效率方面, 图数据库和关系型数据库之间尚未形成统一的测试基准方法和数据集. Cheng 等人提出一个两者间的统一基准[11], 通过对图数据和关系型数据之间的转换方案, 来解决两者间不同数据模型所带来的影响, 将 TPC-H 和 LDBC 数据集进行扩展, 使用不同的存储引擎对比评估了关系型数据库和图数据库之间性能指标. 实验表明, 在主要由 group by、

sort、aggresgartion 等操作构成的查询中, 关系型数据库更具优势, 而在多表连接、路径识别等查询中, 图数据库有更优越的性能. 李金阳[1] 以 MySQL 和 Neo4j 分别作为代表, 对比了在社交关系数据上实现多度查询时的执行时间差异, 在面对大量且复杂的数据连接查询时, 图数据库的响应时间呈线性增长, 远低于关系型数据库的响应时间. Macák 等人[12] 将数据导入到分布式版本数据库中, 探索在扩展集群情况下查询性能的差异, 针对连接、查询过滤以及计算处理 3 类查询分别进行对比. 实验结果表明, 图数据库在前两类查询中更占优势, 而关系型数据库则在计算处理方面的查询响应时间更短.

2 图数据库关键技术

2.1 图查询语言

目前, 图数据库领域没有统一的图查询语言标准, 2019 年 6 月隶属 ISO/IEC 的 Joint Technical Cominttee1(JTC1, 联合技术委员会 1) 通过图查询语言 (graph query language) 的标准提案, 将在未来为期 48 个月的指定工作, 这项工作旨在为属性图标准化图查询语言, 其关注重点是语法的表达能力和语义方面. 而 G-core 研究、SQL/PGQ 项目[13–16] 也都在为图查询语言的标准化统一制定方案. 一般而言, 查询语言分为命令式和声明式两种.

(1) 命令式查询语言

命令式查询语言是一种描述计算机所需作出的行为的编程范式, 系统需要顺序依次执行用户命令, 这种方式有着较高的查询效率但需要用户具备一定的编程能力, 一般情况下命令式查询语言是在需要对查询等业务性能有着更高要求的情况下使用. 在图数据库技术中, Gremlin 和 Neo4j Java API 包含了命令式功能[3].

(2) 声明式查询语言

声明式查询语言允许用户表达检索哪些数据, 剩下的由系统优化完成执行步骤, 这意味着用户操作更加便捷. 声明式查询语言通常作为常规查询语言, 提高图数据的易用性. 在图数据库技术中, 主流的图查询语言, 如: Cypher、Gremlin、GSQL 都是声明式查询语言.

接下来将上述 3 种图查询语言进行对比分析, 参考表 1.

表 1 图查询语言对比

查询语言	Cypher	Gremlin	GSQL
提出者	Neo4j	Apache TinkerPop	TigerGraph
介绍	采用类SQL的语法, 开源版本为OpenCypher	采用类Scala语法	采用类SQL的语法, 支持Map-Ruduce模型
分析型查询	聚合函数	聚合函数、pageRank聚类、PeerPressure	聚合函数、PageRank等
数据定义语言 (DDL)	CRUD	无	CRUD
数据更新语言 (DML)	有	无	有
实现系统	Neo4j、Agens Graph等	TinkerPop等	TigerGraph
图灵完备性	SQL完备 (但结合Java使用可以图灵完备)	图灵完备	图灵完备
使用产品	Neo4j、AgensGraph、RedisGraph等	OrientDB、JanusGraph等	TigerGraph

图算法属于图分析工具, 是分析关联数据的有效方法. 基于图论的数学原理, 图算法利用节点之间的关系来推测复杂系统的组织形态和动态性. 图数据库的使用场景分为实时查询及离线数据分析.

(1) 实时查询

实时查询一般是对数据进行局部查询, 通过对图数据进行遍历搜索、过滤、迭代计算和统计等内容. 图数据库为实时查询提供了两种常用的图算法: 图搜索算法和路径发现算法.

图搜索算法也被称为图遍历算法, 指从一个点出发, 沿边搜索其他顶点过程, 是实现图查询、更新的基础. 这些算法目标是在图上找出路径, 但并不关心这些路径在计算上是否最优. 常见的图遍历算法包括深度优先搜索和广度优先搜索.

路径发现算法基于图搜索算法, 探索节点之间的路径, 从某个节点开始遍历关系, 直到目标节点, 这类算法往往在条件限定的情况下用来识别图中的最优路径, 常见的路径发现算法包括: Dijkstra 最短路径算法、A*算法和最小生成树等. 其中 Dijkstra 算法用于计算一对节点之间的最短 (加权) 路径, 该算法首先查找从起始节点到与其直接连接的节点的最小权重关系, 并追踪这些权重并移动至“最近”节点, 然后针对当前节点执行同一计算, 权重采用起始节点

起算的累计值. 算法重复上述操作, 直至目标节点. Dijkstra 算法的缺点是不支持负的权重, 且运行速度较慢.

为了加快查找速度, A*最短路径算法在前者基础上进行了改进, 在运行过程中, A*算法在主循环的每次迭代中都会判定要扩展哪些路径, 通过估计到达目标节点剩余路径的 (启发式) 代价来完成该过程. A*算法的路径选择是要使式 (1) 最小化:

$$f(n)=g(n)+h(n) \tag{1}$$

其中, $g(n)$ 是从起始节点到节点 n 的路径代价. $h(n)$ 是节点 n 到目标节点的路径代价估计, 通过启发式函数计算.

A*算法中对估计路径代价采用的启发式方法需要根据不同情况制定, 如果在启发式方法中高估路径代价, 则可能导致跳过本该被计算的实际较短路径, 导致错误结果. 此外, A*算法需要相对较高的内存存储其使用数据. 研究者们通过对启发式函数优化选择以及其获取方式的改进[17], 提出双向 A*算法、迭代 A*算法等, 进一步提升了执行效率.

最小生成树算法从给定节点开始, 查找其所有可达节点以及连接这些节点权重最小的关系集合. 它从任意已经过节点遍历到下一位访问节点的权重最小, 从而避免了环的出现. 对比 Dijkstra 最短路径算法, 它没有在每个关系结束时都最小化总路径长度, 而是将每个关系的长度分别最小化, 这使其可以计算带有负权重的关系图, 但该算法在图没有关系权重或者关系权重值相同情况下运行则没有意义.

(2) 离线分析

除了实时查询, 应用场景还经常需要对海量数据进行离线分析, 以便从中挖掘出有效信息. 相对而言, 离线分析需要更长的时间完成, 算法也更加复杂, 一般通过解决问题的目的不同分为图挖掘算法、中心性算法以及社区发现算法.

图挖掘算法是基于图的数据挖掘, 用来发现数据的模式, 可以帮助用户或者上层应用更好地挖掘数据中的潜在信息. 典型的图挖掘算法包括频繁子图、三角形计数等.

频繁子图算法用于枚举在图中所有出现次数超过设定阈值的子图, 一般采用自底向上 (即扩展图规模) 的挖掘策略, 包括基于 Apriori 的 Apriori-Max-Graph 算法、基于 FP-增长的 MARGIN 算法等. 该类算法缺点在于挖掘过程中需经过多次迭代及多次子图同构的判断, 且子图同构的判断属于 NP 完全问题, 此外基于 FP-增长的方法挖掘到的最大频繁子图集与频繁子图集相比只是减少了结果集的数量, 并不能降低挖掘难度. 研究者还提出基于深度优先搜索的 gSpan 算法, 能够有效减少冗余候选子图的生成, 提高挖掘效率.

三角形计数算法用于确定图中经过每个节点的三角形数量, 该算法往往和聚类系数紧密相关, 用于估计群组稳定性以及图中出现紧密相关的簇.

中心性算法用于理解图中特定节点的角色及其对网络的影响. 这些算法能够识别最重要的节点, 并帮助我们了解群体动态, 例如可信度、可访问性、事物传播的速度以及子图之间的连接点. 常见中心性算法包括度中心性算法、中介中心性算法、接近中性性算法和 PageRank 算法等.

度中心性算法用于度量节点拥有的关系数量, 可用来识别在线拍卖网站存在的普通用户和诈骗分子群体, 后者加权中心度往往有着明显的异常[18]. 接近中心性算法用于发现可通过子图高效传播信息的节点, 其衡量指标是该节点到其他各节点的平均距离 (反距离), 接近中心度得分高的节点与其他节点距离短. 其计算公式如下:

$$C(u)=\frac{1}{\sum_{v=1}^{n-1} d(u,v)} \tag{2}$$

其中, u 为节点, n 为图中节点总数, $d(u, v)$ 是另一个节点 v 和节点 u 之间最短距离. 通常会将该得分进行归一化处理, 以此表示最短路径的平均长度, 而非最短路径之和, 归一化后可以比较在不同规模图中节点的接近中心性. 接近中心性归一化公式如下:

$$C_{\text{norm}}(u)=\frac{n-1}{\sum_{v=1}^{n-1} d(u,v)} \tag{3}$$

Wasserman&Faust 算法在接近中心性算法基础上提出改进, 计算群组中可达节点数与到可达节点平均距离的比值, 对于检测一个节点在整个图 (而非子图) 中的重要性更加有效.

中介中心性算法计算连通图中没对接点之间的最

短 (加权路径), 每个节点的分值根据通过该节点的最短路径数量确定. 对所有最短路径, 将下面公式的结果相加以计算节点中介中心性得分:

$$B(u)=\sum_{s\neq t\neq u}\frac{p(u)}{p} \tag{4}$$

其中, u 为节点, p 为节点 s 和 t 之间最短路径总数, $p(u)$ 为 s 与 t 之间通过节点 u 的最短路径数量.

PageRank 算法度量节点的传递性 (或方向性) 的影响, 通过对节点输入关系的数量和质量, 评估该节点的重要性, 最初用于网页排序, 其算法公式如式 (5):

$$PR(u)=(1-d)+d\left(\frac{PR(T1)}{C(T1)}+\cdots+\frac{PR(Tn)}{C(Tn)}\right) \tag{5}$$

假设页面 u 中含有从第 $T1$ 页到 Tn 页的引用, d 是取值介于 0–1 的阻尼系数, 1–d 是不考虑任何关系直接到达节点的概率, $C(Tn)$ 定义为节点 T 的出度.

PageRank 算法的缺点是忽略了主题相关性, 导致查询结果与主题的相关性降低; 另外, PageRank 对新的链接不友好. 近年来, 研究者们针对 PageRank 算法在不同领域的使用进行了优化, Sayyadi 等人[17] 将时间因素整合到 PageRank 算法, 提出 FutureRank 算法. 华一雄等人[18] 提出基于文本相似度及入出比的改进方案, 将其应用于文献搜索领域. Yang 等人[19] 将时间反馈和主题相似度结合, 通过添加页面更新率因子、主题相关因子来进行 PageRank 算法改进. Zhong 等人[20] 则提出一种基于资源分配的改进方案, 能够在定向网络中是被影响力更高的节点.

社区发现算法是基于网络拓扑结构信息识别出的具有相似特征或起相同作用的节点的集合, 发现社团的一般性原则是社团成员在群组内部的关系要多于其与群组外部节点的关系. 近年来, 已有许多复杂网络社区挖掘方法被提出, 依据原理可以被分为基于划分、基于模块度优化、基于标签传播的方法等.

基于划分的社区去挖掘方法核心思想是先找出社区间的全部链接, 将其删除, 最后每个连通分支对应着一个社区. 基于上述思想, GN 算法[21] 被提出, 该算法采用的启发式规则是: 社区间链接的边介数应大于社区内链接的边介数, 其中每个链接的边介数被定义为“网络中经过该链接的任意两点间最短路径的条数”. GN 算法的缺点是计算速度慢, 针对其不足, 研究者们引入统计方法[22]、链接聚类系数[23]、结构相似度[24] 等关键技术进行改进.

模块度 Q 作为一种社区发现指标, 将图分割为更粗粒度的模块, 然后度量分组强度, 模块度采用矩阵形式表达为式 (6) 和式 (7):

$$Q=\frac{1}{2m}Tr(S^{\mathrm{T}}BS) \tag{6}$$

$$B_{ij}=A_{ij}-\frac{k_ik_j}{2m} \tag{7}$$

其中, k 代表的是节点 i 的度, A_{ij} 为邻接矩阵, S 为每个节点所属社区的 one-hot 表示, S_{ir}=1 表示第 i 个节点属于第 r 个社区.

2004 年 Newman 等人提出第一个基于模块度优化的社团发现算法 (FN 算法)[25]. 该算法去初始化时, 对候选解中每个社区仅包含一个节点, 迭代过程中, FN 算法选择使模块性函数 Q 增加最大 (或减小最少) 的社区进行合并, 当候选集只对应一个社区时算法结束. 基于模块度思想, 后续研究者结合谱图理论[26]、局部优化和多层次聚类[27] 技术等技术实现优化. 模块度算法缺点在于多个分割选项出现相似模块度时, 可能出现停滞, 形成局部极点阻碍进一步处理.

标签传播算法是一类启发式算法, 其启发式规则为“在具有社区结构的网络中, 任一节点都应该与大多数邻居在同一个社区”. 2007 年, Raghavan 等人提出著名的标签算法 (LPA 算法)[28]. 其流程是初始化时, 每个节点被赋予唯一标签, 在迭代过程中, 每个节点采用大多数邻居的标签来更新自身标签, 当所有节点的标签都与多数邻居标签相同时, 算法结束. 基于 LPA 算法, 研究者们通过对目标函数修正[28]、多步层次贪婪算法[29] 等进行性能提升.

针对社区发现的研究还处于发展阶段, 社区发现算法的评价标准还未统一, 对于同一个网络, 用不同社区发现算法会得到不同的社区结构, 不同评价标准也会得到不同的最优社区结构.

目前图算法的研究仍处于发展阶段, 不同领域针对同构、异构、动态网络的图特征挖掘有着不同的解决方案, 结合机器学习以及图神经网络技术的发展, 图算法的效率在被进一步提升[30,31].

2.2 图存储

图数据存储被依据其底层实现原理划分为原生图存储和非原生图存储. 原生图存储以图为原型对数据

进行组织管理, 是针对于图数据定制化管理方式. 非原生图存储则采用了外部存储引擎, 使用包括列式结构、key-value 结构等存储数据, 能够兼容不同类型数据格式.

Vertex	Inuse	NextRelId	NetPropId	Labels	Extra
Bytes	0	1	5	9	14

Edge (relation)	Inuse	First node	Seconde node	RelType	FirstPrev relId	FirstNext relId	SecPrev relId	SecNext relId	NextPropId	FiretlnChain-Marker
Bytes	0	1	5	9	13	17	21	25	29	33

图 4 Neo4j 中顶点和边的物理存储结构

2.2.1 原生图存储

原生图数据库的代表是 Neo4j 和 TigerGraph, 它提供原生的图数据存储、处理和检索, 其原生图存储层使用了“无索引邻接”[32], 该方法的特性是指, 每个顶点维护指向它的邻接顶点的直接引用[3], 每个顶点可以看作是它的邻接顶点的一个“局部索引”. 下面以 Neo4j 的物理存储结构为例展示无索引邻接的实现, Neo4j 将属性图的顶点、边、属性和标签保存在了不同的存储文件中, 通过分离存储方案, 提高了存储和访问效率. 图 4 给出了 Neo4j 2.2 版本的顶点和边的物理存储逻辑结构, 其中顶点占用 15 字节, 边记录则占用 34 个字节, 顶点和边记录中各字节的详细内容参考表 2、表 3.

表 2 Neo4j 顶点记录物理存储结构中各字节作用

顶点记录	字节位置	用处
issue	0	表示该节点是否在被使用中还是已经删除
NextRelId	1–4	关联到该节点的第一个关系的id (边记录)
NextPropId	5–8	关联到该节点的第一个属性的id
labels	9–12	指向顶点标签存储的指针
Extra	13	存储内部标志信息

表 3 Neo4j 边记录物理存储结构中各字节作用

顶点记录	字节位置	用处
issue	0	表示该节点是否在被使用中还是已经删除
firstNode	1–4	起始顶点id
secondNode	5–8	终止顶点id
relType	10–13	指向该边的关系类型的指针
firstPrevRelId	13–16	指向起始顶点上前一个边的指针
firstNextRelId	17–20	指向起始顶点上后一个边的指针
secPrevRelId	21–24	指向终止顶点上前一个边的指针
secNextRelId	25–28	指向终止顶点上后一个边的指针
nextPropId	29–32	边记录上第一个属性的id
firstinChainMarker	33	表示这条边是否是“关系链”第一条记录的标记位置

2.2.2 非原生图存储

非原生图数据库在底层数据存储的实现上没有直接采用图模型, 而是在此之上对图进行封装. 以 JanusGraph 为例, 其存储端采用了基于 Google Bigtable[33] 的 KCV (key-column-value) 的数据模式. 它的存储方案中包含了两种图切割方法: 按边切割 (edge cut) 和按节点切割 (vertex cut). 在默认模式下, JanusGraph 采用了按边切割的方式[3] 来进行图切割存储.

按边切割: 根据边进行切割, 以节点为中心, 每条边存储两次, 源节点的领接列表存储一次, 目标节点的领接链表存储一次.

按节点切割: 根据点进行切割, 每个边只存储一次, 节点对应的边便会多一份该节点的存储.

JanusGraph 基于使用 BigTable 模型的存储后端, 完成存储的逻辑结构, 如图 5 所示.

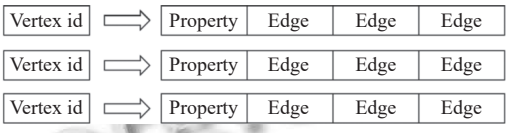

图 5 JanusGraph 的图存储整体结构

图的存储整体结构分为 3 个部分: vertex id、property、edge.

Vertex id: 对应节点的唯一 id, 以使用 HBase 为例, 则代表当前行的 Rowkey, 代表某个节点.

Property: 代表节点的属性.

Edge: 代表节点的对应的边.

图 6 则给出了 Vertex id 的存储结构, 进行序列化存储时, vertex id 共包含一个字节, 8 位, 64 bit, 分为 3 个部分: partition id、count、ID padding. 前 5 位为 partition id, partition 是 JanusGraph 抽象出的概念, 通过其数量计算可以最终使数据均匀分配到多台机器中. 中间的 count 是流水号, 最高位固定为 0, 其剩余位数足以生成 2 的 55 次幂 (约 30 000 兆) 个 id, 满足节点数量生成. 最后几位 bit 是 ID padding, 表示 Vertex 的类

型, 具体位数长度会随不同类型有所修改, 常用情况值为“000”.

图 7 给出了 Edge 和 Property 的逻辑结构, 均分别由 column 和 value 两部分组成.

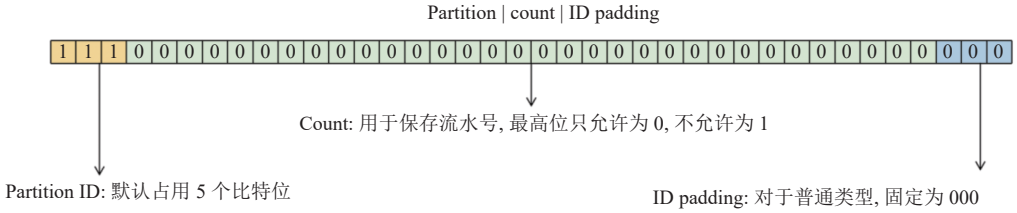

图 6 JanusGraph 中 Vertex id 的物理存储结构

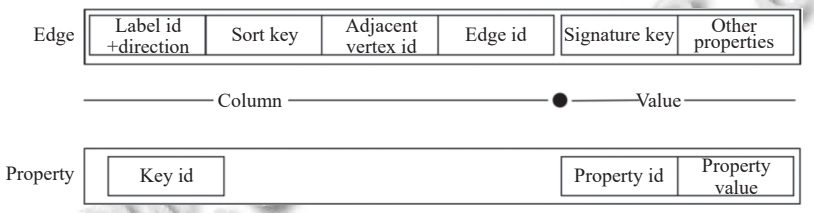

图 7 JanusGraph 的节点和边的物理存储结构

在 Edge 的 column 中, 包含了 label id、direction、sort key、adjacent vertex id 和 edge id. 其中, label id 是边类型代表的 id. Direction 是图的方向, 用 0 和 1 来分别代表出和入. Sort key 可以指定多个边的属性. Adjacent vertex id 是目标节点 id, 实际存储的是目标节点 id 和源节点 id 的差值. Edge id 则是边的全局唯一 id. Edge 的 value 由 signature key 和 other property 组成, 前者用于提升 edge 的属性的检索速度, 后者则是存储边的其他属性.

Property 的 column 包含 key id 和 property id, 前者用于存储属性 label 对应的 id 值, 后者则是指定属性的唯一 id. Value 中只有 property value 用于保存属性值.

基于上述整体的逻辑结构, 可以发现 JanusGraph 通过 vertex id 行保存了跟当前节点相关的所有边, 在边的逻辑结构中, 使用 adjacent vertex id 字段来获取目标节点, 形成了由源节点指向目标节点的图模式.

图存储方案作为图数据库底层数据管理基础, 在原生和非原生存储方案上各具优劣, 不同产品针对其优化也体现了各自优势. 当前, Neo4j、TigerGraph 和 JanusGraph 等主流图数据库研究人员针对其方案优化, 仍需进一步探索.

3 图数据库产品及应用

3.1 图数据库产品

从 DB-Engines 中获取的图数据库排名可知, 目前国内外对图数据库产品的研发投入越来越高, 各产品迭代更新也越来越快. 本节将选取 Neo4j、TigerGraph、JanusGraph、ArangoDB、OrientDB 这 5 款图数据库进行对比分析, 详情参照表 4.

3.1.1 原生图数据库

(1) Neo4j

Neo4J 是目前业界最受欢迎和使用的原生图数据库, 其底层存储采用了原生的属性图模型, 具有“无索引邻接”的特性. Neo4j 有着高扩展性支持存储实现上亿数量级的节点及关系内容, 提供了两种运行模式, 分别是嵌入式模式和服务器模式. 在嵌入式模式下, Neo4j 和应用程序运行于同一进程, 该模式的目标应用是硬件设备, 桌面应用程序和应用服务器的组件. 服务器模式是更为常用的方式, 每个服务器的核心是一个 Neo4j 的实例, 典型的访问 Neo4j 的服务器的方式是使用 REST API. Neo4j 支持使用名为 Cypher 图查询语言, 有着良好的表现力和较高的查询效率.

(2) TigerGraph

TigerGraph 是一款作为原生、实时和大规模并行处理 (MPP) 的图数据库, 其存储通过底层原生属性图设计, 避免了从虚拟图操作到物理存储操作的转换带来的开销. TigerGraph 为用户的快速访问提供了便捷, 用户可以在使用中设置参数来指定存储于内存的图大小, 如果图不能整个在内存中进行操作,

则多余的数据会放置在磁盘, 数据值会被压缩保存, 其采用 C++编写, 对内存进行细粒度管理, 内置压缩因子随着图结构和数据的不同而变化, 提高存储和查询效率.

表 4 图数据库产品对比

图数据库	Neo4j	JanusGraph	ArangoDB	TigerGraph	OrientDB
发布方	Neo Technology	Apache	triAGENS GmbG	TigerGraph	orienttechnologies
存储方案	原生图存储	非原生图存储, 可以使用Cassandra、HBase、Bigtable、Berkeley作为存储后端	非原生存储, 可以使用键值对 (key-value)、文档或者图形的数据模型进行存储.	原生图存储	原生图存储, 支持文档、图形、键值对 (key-value)和对象的数据模型
图查询语言	Cypher	Gremlin	AQL	GSQL	Gremlin、扩展的SQL
分布式	不支持	支持	支持	支持	支持
备份	社区版支持手工备份, 企业版支持热备份	未提供支持, 需用户编写	开发社区中存在大量问题	支持	支持
许可证	开源GPLv3/商业	开源Apche Licence 2.0	开源Apche Licence 2.0	商业闭源	开源Apche Licence 2.0/商业

3.1.2 非原生图数据库

(1) ArangoDB

ArangoDB 是一款多模型数据库, 可以使用键值对 (key-value)、文档或者图形的数据模型进行存储. ArangoDB 采用了试用于所有数据模型的统一内核和统一数据库查询语言. 因此, 用户可以在单次查询过程中混合使用多种模型, 这种混合多模型数据存储方式增加了数据存储的灵活性、简化了性能扩展、提高了容错能力以及降低了的存储成本.

(2) OrientDB

OrientDB 是一个多模型的开源数据库, 支持文档、图形、键值对 (key-value) 和对象的数据模型, , 支持事务性和分布式体系结构, 数据库的操作可以使用 Java、SQL 或者 Gremlin 来完成, 物理数据存储可以在内存和磁盘上完成. OrrientDB 使用文档数据库和面向对象功能来存储物理顶点. 它支持分布式体系结构, 支持多尺度安全验证和数据加密, 有着良好的数据安全性支持.

(3) JanusGraph

JanusGraph 是一个开源的分布式图数据库, 是个有着良好的扩展性, 通过多机集群可以支持存储和查询数百亿的顶点和边图数据. JanusGraph 为数据持久化, 数据索引和客户端访问提供了强大的模块化接口. 它可以适配多种数据库和索引, 数据库方面包括大数据处理中常见的 Apche Cassandra、Apache HBase、Google Bigtable、Oracle Berkeley. 索引方面, 为了加快并支持更复杂的查询, 它支持 Elasticsearch、Apache Solr、Apache Lucene 等. 这种模块化的设计能够更好地增强分布式图系统的功能, 提供了优化的磁盘表示形式, 以便更高效地存储和更快的访问速度. JanusGraph 还通过集成大数据平台, 如 Apache Spark、Apache Giraph、Apache Hadoop 等, 支持全图数据的分析、报表和 ETL.

3.2 图数据库基准测试

图数据库暂无像关系型数据库一般形成统一的基准测试标准, 国内外研究一般集中于数据加载、查询性能和查询可扩展性 3 个方面.

(1) 数据加载

Deutsch 等[7] 对多款数据库在数据加载的时间和占用磁盘容量两个角度分别进行了比对分析. 在数据加载时间上, 采用初始数据批量操作方式, 得出在使用时间上: TigerGraph<Neo4J<ArangoDB<JanusGraph 的结论. 而在磁盘占用容量上, TigerGraph 在测试中有明显的优势, 其内置的自动编码和压缩数据算法让它在导入后的数据比原始数据所占空间更小, 而其他几款产品加载后的数据空间占用程度都较于原始数据更高.

(2) 查询性能

针对基本查询、图遍历以及最短路径计算 3 个方面进行对比, JanusGraph 的效率均高于 Neo4j[30]. 而由 Fernandes 等人[32] 提出的测试对比中, 在查询响应时间方面 ArangoDB 更优于 OrientDB 和 JanusGraph, 结合其在用户体验上的特性, 在综合评分上 ArangoDB> JanusGraph>OrientDB. 针对 ArangoDB 和 Neo4j 在查询处理时间、内存利用率和 CPU 利用率上进行了对比, 与 Neo4j 相比, AranggoDB 查询时间上相对更占优

势且占用的 CPU 相对较少, 但主存耗费更多[11]. 针对 K 度查询、弱联通子图查询和图算法 PageRank 的查询实现进行对比测试中[15], 在 K 度路径顶点计数查询方面, 报告分 1、2、3、6 度 4 个梯度分别进行测试, 在一度和二度查询上, TigerGraph 在两个测试集上表现都为最佳, 而 Neo4j、JanusGraph 则有着相对接近的查询耗时, ArangoDB 在这项测试中表现明显低于其他几项数据库. 而在三度和多度这两项深度查询中, 除 TigerGraph 外其他几款数据库均出现超时或者内存耗尽的情况. 在弱联通子图和 PageRank 查询中, 仅有 TigerGraph 和 Neo4j 在规定时间内完成了查询.

(3) 查询可扩展性

在查询的可扩展性方面, TigerGraph 能够通过集群布置能够显著提升其查询性能, Neo4j 则因未提供对图的切片处理而无法测试[15]. 而 Cheng 等人[11] 通过使用多个节点部署的集群测试中, ArangoDB、JanusGraph、OrientDB 均在多节点的情况下用时高于单节点, 有着较好的扩展性.

3.3 图数据库各领域应用研究

当前而言, 图数据库应用的领域和场景越来越多样化, 许多著名的公司都开始使用图数据库来完成产业的发展和创新. Twitter、Facebook、Google 等公司更是在图数据库出现的早期就开始了对其在社交网络应用上的探索[34,35]. Neo4j 和 OrientDB 的客户包括安全公司、调查单位、媒体公司 (Sky、Comcast、Warner) 和贸易公司 (Ebay 或全球 500 强物流), 它们使用图形数据库为客户提供实时的产品路线和交付[36]. 此外, 使用图结构来表示生物医学领域分子、药物等模型能够较高程度还原其真实性, 越来越多生物医学行业研究采用图数据库来进行数据存储[37–58]. 本文于 2021 年 10 月使用 DBLP、中国知网数据库、万方数据知识服务平台, 以“Graph database”“图数据库”作为主题检索自 2018 年以来相关文献, 共获得以图数据库为基础的应用共 145 篇, 经人工统计, 应用分布比例如图 8. 知识图谱是图数据库关联最为紧密、应用范围最广的应用场景, 采用图数据库作为底层数据存储方式, 实现了医疗、生物、金融、公文政策等多个领域的知识图谱[59–66], 能够帮助行业提高决策效率. 在社交网络方面, 使用图数据库存储出行人员多维数据, 帮助挖掘可疑人员、密切接触者等重点群体[67–70], 针对 2019 年所爆发的新冠疫情, 能够快速找到确诊病例和疑似病例的密切接触者, 提高了分析效率. 图数据库运用在推荐系统领域, 包括的电影推荐、图书推荐、医生推荐等, 能够进一步提升推荐的准确率[71–76]. 电力系统及智能物联网领域通过图数据库对连接设备进行建模, 能够直观地反应电网拓扑结构、设备信息流通以及资源消耗情况, 帮助提高设备管理效率[77–91]. 金融行业通过图数据库, 利用多维交叉关联信息深度刻画申请和交易行为, 可以有效识别规模化、隐蔽性的欺诈网络和洗钱网络[92–100]. 使用图数据库产品以及丰富的图分析算法, 结合 NLP、CV 等人工智能热门领域技术, 情报科学领域在各类非结构化文档实体关系提取以及视频数据中人员关联分析等应用也进一步发展[100–102]. 表 5 选取了图数据库近年来在各领域的代表性应用研究, 可以发现目前 Neo4j 凭借良好的开源性社区支持, 成为了广大开发者使用首选.

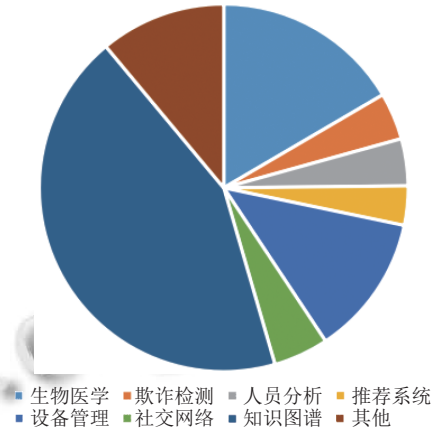

图 8 图数据库应用统计分布

4 总结展望

目前, 图数据库的各项理论、方法、技术和系统处于快速发展和开发完善阶段. 数据库学术界和产业界对图数据库研发投入的成本不断增加. 本文主要概述了图数据库基本特点, 并从图查询语言、图计算、图存储 3 个方面对图数据库关键技术进行论述分析, 最后比对了市面上主流的图数据库产品以及应用场景. 从目前研究情况来看, 图数据库相关研究仍存在较大发展空间, 未来的研究方向主要包括以下几点.

表 5 图数据库应用

应用领域	文献	时间	使用工具	目标	效果
知识图谱	[98]	2019	Neo4j	构建扶贫领域知识图谱	构建了半自动话生成扶贫领域知识图谱的应用, 对扶贫工作数据应用和公开有重大意义
	[52]	2020	Neo4j	构建电力项目知识图谱	借助电力项目知识图谱, 便于海外电力项目管理, 直观了解项目之间的关系, 便于进一步统筹规划
	[103]	2020	Neo4j	构建贵州省大数据政策知识图谱	可较好地实现基于政策/政令多粒度知识发现的公文间关系分析与推理, 本文所提方法为提升政策制定的系统性和科学性提供参考
	[42]	2019	Neo4j	建立疾病数据知识图谱	使用Neo4j构建疾病数据知识图谱, 通过直观的图像分析提供治疗建议
关联人员分析	[66]	2020	Neo4j	针对话单数据和出行数据构建多维关系网络, 完成人员关系挖掘分析	在保证准确率的情况下, 分析效率提高5倍
	[65]	2019	Neo4j	提出基于图数据库实现的调查性图检索技术	能够有效分析可疑群体信息, 提高犯罪预测成功率
	[99]	2018	Neo4j	提出了一个将DBLP书目建模为图数据库的系统, 用于执行基于图的查询和社会网络分析	基于Neo4j实现GraphDBLP系统对该社区进行社交网络分析
推荐系统	[101]	2017	Neo4j	建立电影推荐系统	设计了一种实体映射关系, 将数据从关系型数据库映射至图数据库以实现电影推荐系统
	[73]	2020	Neo4j	建立医生推荐系统	针对医患关系数据进行存取时间对比, 发现图数据库优于关系型数据库; 引入“信任因子”建立多层网络模型, 提高了该系统的推荐成功率
IT设备管理	[84]	2019	Neo4j	实现电力图拓扑分析引擎	能够提高电网状态初始化性能, 有着更好的性能提升潜力
	[81]	2020	TigerGraph	实现电网拓扑高性能分析原型系统	能直观对运行的电力设备实现存储查询、安全预警等实时管理
	[82]	2017	Neo4j	实现电力系统信息通信资产拓扑网络建模	使用图数据库实现信息通信资产管理可视化
欺诈检测	[93]	2020	Neo4j	实现信用卡反欺诈系统	能够较好地挖掘信用卡黑名单可疑用户群体
	[95]	2020	Neo4j	实现疾病保险行业反欺诈应用	通过图数据库建模、计算得到数据特征, 能够针对性开展反欺诈工作, 对案件进行尽早发现

(1) 统一的图查询语言和测试基准

目前图查询语言尚未有统一标准, 当前图查询语言 (GQL) 的标准还处于制定阶段, 市面上多种查询语言使用了不同的标准, 提高了用户的学习成本. 在具体业务上, 不同产品的查询语言表达方式使得在同样的应用场景下, 所获取的结果不一致. 在出现复杂闭环的图中, 不同产品提供了不同的解决方案[7], 影响了用户使用体验. 统一图查询语言标准的制定, 能够使 GQL 使用更加稳定和高效.

此外, 各公司所提供的实验及数据都没有形成统一的准则, 这让用户在不同产品之间的选择变得困难, Cheng 等人[11] 提出供业界参考的测试基准, 但相对于已经被成熟使用的 RDMS, 图数据库测试标准的制定是未来发展的一个主要方向.

(2) 深度融合图数据库和图处理引擎

目前, 不同图数据库产品在图算法的提供上参差不齐, 部分图数据库无法独立完成复杂的全图迭代计算, 需要使用外部图处理引擎完成任务. 这在一定程度上增加了额外的开销, 加重了系统负担. 而部分图数据库采取了分布式的设计方案, 这使得其处理的数据量规模得到了进一步提升, 同时通过优化提高了查询能力. 图数据库和图处理引擎的深度融合, 如采用内置图算法库以及图处理引擎, 从而为用户复杂的计算提供更简单的内在操作是业界的研发方向.

(3) 融合多模数据库发展

多模数据库对多种数据模型进行存储, 其内置图数据管理的方案存在差异, 建立适用于不同数据模型间转换的图数据接口能够进一步提高数据间的转换效率以及统一使用, 如 Neo4j 4.0 版本中提供了帮助用户将关系型数据定制化转换成图数据模型的工具, 鄂海红等也提出了从表格数据和 RDF 数据格式到图数据的转换方法[104,105]. 此外, 采取分布式架构实现底层数据存储, 使用索引等技术方案, 可以进一步提高使用效率.

(4) 软硬件一体化

图数据库非规则访问的特性对底层硬件的需求越发迫切, 将来可以通过软硬件协同设计方案[6], 比如采用 NVM 减少持久化存储的开销, 使用 RDMA 增强通信效率, 或者将事 务的部分要求交给硬件 (例如 HTM) 来控制、简化软件设计等将成为下一步研究方向.

(5) 支持实时决策和人工智能应用

实时深度关联分析使用户能够比以往更准确、更快速和更深入地探究、发现和预测关系. 使用图数据库对海量数据间关联进行深度实时分析, 可以帮助企业完成人工智能创新应用, 提高用户对海量数据的实时监控和挖掘分析.

随着图数据库技术的不断成熟, 其应用场景也将愈发丰富. 图数据库与图处理引擎融合的图系统带来的强大的图存储和分析能力, 将会进一步推动图数据库在金融风控、社交网络等典型应用场景的使用升级, 也为智能物联网等行业带来了新的应用发展方向.

参考文献

1 李金阳. 图数据库在图书馆的应用研究. 图书馆, 2020, (11): 109–115. [doi: 10.3969/j.issn.1002-1558.2020.11.017]

2 沈志宏, 赵子豪, 王海波. 以图为中心的新型大数据技术栈研究. 数据分析与知识发现, 2020, 4(7): 50–65.

3 王鑫, 邹磊, 王朝坤, 等. 知识图谱数据管理研究综述. 软件学报, 2019, 30(7): 2139–2174. [doi: 10.13328/j.cnki.jos.005841]

4 李俊逸, 魏凯, 姜春宇, 等. 图数据库白皮书. 中国信息通信研究院云计算与大数据研究所, 2019.

5 Robinson I, Webber J, Eifrem E. 图数据库. 刘璐, 梁越, 译. 2 版. 北京: 人民邮电出版社, 2016.

6 李俊逸, 王卓, 马鹏玮. 图数据库技术发展趋势研究. 信息通信技术与政策, 2021, 47(5): 67–72. [doi: 10.12267/j.issn.2096-5931.2021.05.013]

7 Deutsch A, Xu Y, Wu MX, *et al*. TigerGraph: A native MPP graph database. arXiv: 1901.08248, 2019.

8 Green A, Guagliardo P, Libkin L, *et al*. Updating graph databases with Cypher. Proceedings of the VLDB Endowment, 2019, 12(12): 2242–2254. [doi: 10.14778/3352063.3352139]

9 Angles R, Arenas M, Barceló P, *et al*. Foundations of modern query languages for graph databases. ACM Computing Surveys, 2018, 50(5): 68.

10 Deutsch A, Xu Y, Wu MX, *et al*. Aggregation support for modern graph analytics in TigerGraph. Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. Portland: ACM, 2020. 377–392.

11 Cheng YJ, Ding PJ, Wang TT, *et al*. Which category is better: Benchmarking relational and graph database management systems. Data Science and Engineering, 2019, 4(4): 309–322. [doi: 10.1007/s41019-019-00110-3]

12 Macák M, Stovcik M, Buhnova B. The suitability of graph databases for big data analysis: A benchmark. Proceedings of the 5th International Conference on Internet of Things, Big Data and Security. Prague: SciTePress, 2020. 213–220.

13 Chai B, Qiu HB, Liu SY, *et al*. A study of topology analysis engine based on graph databases in electric utilities. 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). Chengdu: IEEE, 2017. 585–588. [doi: 10.1109/ITNEC.2017.8284800]

14 Pacaci A, Zhou A, Lin J, *et al*. Do we need specialized graph databases: Benchmarking real-time social networking applications. Proceedings of the 5th International Workshop on Graph Data-management Experiences & Systems. Chicago: Association for Computing Machinery, 2017. 12.

15 Foead D, Ghifari A, Kusuma MB, *et al*. A systematic literature review of A* pathfinding. Procedia Computer Science, 2021, 179: 507–514. [doi: 10.1016/j.procs.2021.01.034]

16 Bangcharoensap P, Kobayashi H, Shimizu N, *et al*. Two step graph-based semi-supervised learning for online auction fraud detection. Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Porto: Springer, 2015. 165–179.

17 Sayyadi H, Getoor L. FutureRank: Ranking scientific articles by predicting their future PageRank. Proceedings of the 2009 SIAM International Conference on Data Mining. Sparks: SIAM, 2009. 533–544.

18 华一雄, 张执南. 基于文本相似度和入出比的改进 PageRank 科研文献搜索方法. 机械设计与研究, 2021, 37(1): 6–9. [doi: 10.13952/j.cnki.jofmdr.2021.0002]

19 Yang WJ, Zheng PH. An improved Pagerank algorithm based on time feedback and topic similarity. 7th IEEE International Conference on Software Engineering and Service Science. Beijing: IEEE, 2017. 534–537.

20 Zhong LF, Lv FM. An improved pagerank for identifying the influential nodes based on resource allocation in directed networks. 14th International Computer Conference on Wavelet Active Media Technology and Information Processing. Chengdu: IEEE, 2017. 42–45.

21 周万珍, 宋健, 许云峰. 异质网络社区发现方法研究综述.

河北科技大学学报, 2021, 42(3): 231–240. [doi: 10.7535/hbkd.2021yx03004]
22 刘大有, 金弟, 何东晓, 等. 复杂网络社区挖掘综述. 计算机研究与发展, 2013, 50(10): 2140–2154. [doi: 10.7544/issn1000-1239.2013.20120357]
23 Radicchi F, Castellano C, Cecconi F, *et al*. Defining and identifying communities in networks. Proceedings of the National Academy of Sciences of the United States of America, 2004, 101(9): 2658–2663. [doi: 10.1073/pnas.0400054101]
24 金弟, 刘杰, 贾正雪, 等. 基于 *k* 最近邻网络的数据聚类算法. 模式识别与人工智能, 2010, 23(4): 546–551. [doi: 10.3969/j.issn.1003-6059.2010.04.015]
25 Newman MEJ, Girvan M. Finding and evaluating community structure in networks. Physical Review E, 2004, 69(2): 026113. [doi: 10.1103/PhysRevE.69.026113]
26 Blondel VD, Guillaume JL, Lambiotte R, *et al*. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008, 2008: P10008. [doi: 10.1088/1742-5468/2008/10/P10008]
27 Shen HW, Cheng XQ, Cai K, *et al*. Detect overlapping and hierarchical community structure in networks. Physica A: Statistical Mechanics and its Applications, 2009, 388(8): 1706–1712. [doi: 10.1016/j.physa.2008.12.021]
28 Barber MJ, Clark JW. Detecting network communities by propagating labels under constraints. Physical Review E, 2009, 80(2): 026129. [doi: 10.1103/PhysRevE.80.026129]
29 Liu X, Murata T. Advanced modularity-specialized label propagation algorithm for detecting communities in networks. Physica A: Statistical Mechanics and its Applications, 2010, 389(7): 1493–1500. [doi: 10.1016/j.physa.2009.12.019]
30 葛唯益, 王振宇, 王羽, 等. 主流知识图谱存储系统试验对比. 指挥信息系统与技术, 2019, 10(5): 28–33, 75. [doi: 10.15908/j.cnki.cist.2019.05.006]
31 史晓丽. Bigtable 分布式存储系统的研究 [硕士学位论文]. 西安: 西安电子科技大学, 2014.
32 Fernandes D, Bernardino J. Graph databases comparison: AllegroGraph, ArangoDB, InfiniteGraph, Neo4J, and OrientDB. Proceedings of the 7th International Conference on Data Science, Technology and Applications. Porto: SciTePress, 2018. 373–380.
33 Neiheiser R, Rech L, Bravo M, *et al*. Fireplug: Efficient and robust geo-replication of graph databases. IEEE Transactions on Parallel and Distributed Systems, 2020, 31(8): 1942–1953. [doi: 10.1109/TPDS.2020.2981019]
34 D'Agostino D, Liò P, Aldinucci M, *et al*. Advantages of using graph databases to explore chromatin conformation capture experiments. BMC Bioinformatics, 2021, 22(2): 43.
35 Licheri N, Bonnici V, Beccuti M, *et al*. GRAPES-DD: Exploiting decision diagrams for index-driven search in biological graph databases. BMC Bioinformatics, 2021, 22(1): 209. [doi: 10.1186/s12859-021-04129-0]
36 崔斌, 高军, 童咏昕, 等. 新型数据管理系统研究进展与趋势. 软件学报, 2019, 30(1): 164–193. [doi: 10.13328/j.cnki.jos.005646]
37 Cardoso C, Sousa RT, Köhler S, *et al*. A collection of benchmark data sets for knowledge graph-based similarity in the biomedical domain. Database, 2020, 2020: baaa078. [doi: 10.1093/database/baaa078]
38 Mei SQ, Huang XW, Xie CS, *et al*. GREG—Studying transcriptional regulation using integrative graph databases. Database, 2020, 2020: baz162. [doi: 10.1093/database/baz162]
39 Simpson CM, Gnad F. Applying graph database technology for analyzing perturbed co-expression networks in cancer. Database, 2020, 2020: baaa110. [doi: 10.1093/database/baaa110]
40 Lose T, Van Heusden P, Christoffels A. COMBAT-TB-NeoDB: Fostering tuberculosis research through integrative analysis using graph database technologies. Bioinformatics, 2020, 36(3): 982–983.
41 Rickett CD, Maschhoff KJ, Sukumar SR. Massively parallel processing database for sequence and graph data structures applied to rapid-response drug repurposing. 2020 2020 IEEE International Conference on Big Data (Big Data). Atlanta: IEEE, 2020. 2967–2976.
42 Arias JF. The benefits of graph databases for the computation of clinical quality measures. 2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS). Rochester: IEEE, 2020. 433–436.
43 Pabón MC, Millán M, Roncancio C, *et al*. GraphTQL: A visual query system for graph databases. Journal of Computer Languages, 2019, 51: 97–111. [doi: 10.1016/j.cola.2018.12.006]
44 Aguilera-Mendoza L, Marrero-Ponce L, Beltran JA, *et al*. Graph-based data integration from bioactive peptide databases of pharmaceutical interest: Toward an organized collection enabling visual network analysis. Bioinformatics, 2019, 35(22): 4739–4747. [doi: 10.1093/bioinformatics/btz260]
45 El Helou S, Kobayashi S, Yamamoto G, *et al*. Graph

databases for openEHR clinical repositories. International Journal of Computational Science and Engineering, 2019, 20(3): 281–298. [doi: 10.1504/IJCSE.2019.103955]

46 Zhao J, Hong ZG, Shi MY. Analysis of disease data based on Neo4j graph database. 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS). Beijing: IEEE, 2019. 381–384.

47 Ueta K, Nakamoto Y, Xue XY, *et al*. Distributed graph database as base of smart world things. 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). San Francisco: IEEE, 2017. 1–5.

48 Yang C, de Baets B, Lachat C. From DIKW pyramid to graph database: A tool for machine processing of nutritional epidemiologic research data. 2019 IEEE International Conference on Big Data (Big Data). Los Angeles: IEEE, 2019. 5202–5205.

49 Algul E, Wilson RC. A database and evaluation for classification of RNA molecules using graph methods. Proceedings of the 12th International Workshop on Graph-Based Representations in Pattern Recognition. Tours: Springer, 2019. 78–87.

50 Thapa I, Ali H. A new graph database system for multi-omics data integration and mining complex biological information. Proceedings of the 9th International Conference on Computational Advances in Bio and Medical Sciences. Miami: Springer, 2019. 171–183.

51 Le KK, Whiteside MD, Hopkins JE, *et al*. Spfy: An integrated graph database for real-time prediction of bacterial phenotypes and downstream comparative analyses. Database, 2018, 2018: bay086.

52 Messina A, Fiannaca A, La Paglia L, *et al*. BioGraph: A web application and a graph database for querying and analyzing bioinformatics resources. BMC Systems Biology, 2018, 12(S5): 98. [doi: 10.1186/s12918-018-0616-4]

53 Shoshi A, Hofestädt R, Zolotareva O, *et al*. GenCoNet—A graph database for the analysis of comorbidities by gene networks. Journal of Integrative Bioinformatics, 2018, 15(4): 20180049.

54 Wiese L, Wangmo C, Steuernagel L, *et al*. Construction and visualization of dynamic biological networks: Benchmarking the Neo4j graph database. Proceedings of the 13th International Conference on Data Integration in the Life Sciences. Hannover: Springer, 2018. 33–43.

55 Da Silva WMC, Wercelens P, Walter MEMT, *et al*. Graph databases in molecular biology. Proceedings of the 11th Brazilian Symposium on Bioinformatics. Niterói: Springer, 2018. 50–57.

56 Liu H B, Jiang GY, Su LH, *et al*. Construction of power projects knowledge graph based on graph database Neo4j. 2020 International Conference on Computer, Information and Telecommunication Systems (CITS). Hangzhou: IEEE, 2020. 1–4.

57 Wu H, Wang YJ, Chen P, *et al*. Application of graph database for the storage of knowledge map of power grid model. 2020 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC). Macao: IEEE, 2020. 1–5.

58 Chatterjee J, Dethlefs N. XAI4Wind: A multimodal knowledge graph database for explainable decision support in operations & maintenance of wind turbines. arXiv: 2012.10489, 2020.

59 Vogt L, Baum R, Bhatty P, *et al*. SOCCOMAS: A FAIR Web content management system that uses knowledge graphs and that is based on semantic programming. Database, 2019, 2019: baz067. [doi: 10.1093/database/baz067]

60 Xia T, Gu YJ. Building terrorist knowledge graph from global terrorism database and Wikipedia. 2019 IEEE International Conference on Intelligence and Security Informatics (ISI). Shenzhen: IEEE, 2019. 194–196.

61 Sequeda JF, Briggs WJ, Miranker DP, *et al*. A pay-as-you-go methodology to design and build enterprise knowledge graphs from relational databases. Proceedings of the 18th International Semantic Web Conference. Auckland: Springer, 2019. 526–545.

62 Omasa A, Inoue U. Extracting related concepts from Wikipedia by using a graph database system. 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). Toyama: IEEE, 2019. 268–273.

63 Zaki N, Tennakoon C, Al Ashwal H, *et al*. Methods of creating knowledge graph by linking biological databases. International Conference on Practical Applications of Computational Biology & Bioinformatics. Toledo: Springer, 2018. 52–62.

64 李剑, 温海燕, 张恒, 等. 基于图数据库的传染病高风险人群防控系统研究. 中国信息化, 2021, (3): 79–82. [doi:

10.3969/j.issn.1672-5158.2021.03.031]
65 沈阳, 李洪磊, 陈杰. 图数据模型及其在疫情追溯领域的应用研究. 软件导刊, 2021, 20(2): 13–17. [doi: 10.11907/rjdk.211005]
66 丁洪丽. 基于 Neo4j 图数据库的人员关系挖掘. 电讯技术, 2020, 60(7): 771–777. [doi: 10.3969/j.issn.1001-893x.2020.07.006]
67 尹玉娇, 张伟. 一种基于图数据库的虚拟身份关系挖掘算法. 软件导刊, 2020, 19(1): 117–122.
68 屈新明, 郭鹏, 丘建栋, 等. 基于图数据库的公交出行行为分析. 智能城市, 2019, 5(13): 20–22.
69 陈喜春, 牛晓莉. 基于图数据库的装备信息联想式查询. 计算机系统应用, 2019, 28(5): 244–247. [doi: 10.15888/j.cnki.csa.006895]
70 高松. 基于知识图谱的合作者推荐系统设计与实现 [硕士学位论文]. 大连: 大连理工大学, 2019.
71 梁艳杰. 基于图数据库的个性化推荐系统研究与设计 [硕士学位论文]. 重庆: 重庆邮电大学, 2018.
72 Giabelli A, Malandri L, Mercorio F, *et al*. Skills2Job: A recommender system that encodes job offer embeddings on graph databases. Applied Soft Computing, 2021, 101: 107049. [doi: 10.1016/j.asoc.2020.107049]
73 Mondal S, Basu A, Mukherjee N. Building a trust-based doctor recommendation system on top of multilayer graph database. Journal of Biomedical Informatics, 2020, 110: 103549. [doi: 10.1016/j.jbi.2020.103549]
74 Candell R, Kashef M, Liu YK, *et al*. A graph database approach to wireless IIoT workcell performance evaluation. 2020 IEEE International Conference on Industrial Technology (ICIT). Buenos Aires: IEEE, 2020. 251–258.
75 Küçükkeçeci C, Yazıcı A. Big data model simulation on a graph database for surveillance in wireless multimedia sensor networks. Big Data Research, 2018, 11: 33–43. [doi: 10.1016/j.bdr.2017.09.003]
76 Zhou AH, Qiu HB, Pan S, *et al*. Research on power grid topology analysis based on graph database. 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS). Nanjing: IEEE, 2018. 841–844.
77 Jamkhedkar P, Johnson T, Kanza Y, *et al*. A graph database for a virtualized network infrastructure. Proceedings of the 2018 International Conference on Management of Data. Houston: ACM, 2018. 1393–1405.
78 吴新新. 面向电网拓扑管理的图数据库关键技术研究 [硕士学位论文]. 南京: 南京邮电大学, 2020.
79 李旻. 物联网全流程追溯中图数据库的应用. 电脑知识与技术, 2020, 16(32): 36–37, 53.
80 张钊棋, 宋辉, 崔国栋, 等. 基于图数据库的电力设备绝缘状态文本数据存储. 电气自动化, 2020, 42(5): 64–66. [doi: 10.3969/j.issn.1000-3886.2020.05.020]
81 刘广一, 戴仁昶, 路轶, 等. 基于图计算的能量管理系统实时网络分析应用研发. 电工技术学报, 2020, 35(11): 2339–2348.
82 汤亚宸, 方定江, 韩海韵, 等. 基于图数据库和知识图谱的电力设备质量综合管理系统研究. 供用电, 2019, 36(11): 35–40.
83 刘广一, 何明, 周建其, 等. 基于图计算的主动配电网综合能源服务技术支持系统. 供用电, 2019, 36(11): 3–11, 54.
84 谭俊, 张国芳, 刘广一, 等. 基于图计算的配电网建模与分析. 供用电, 2019, 36(11): 28–34, 54.
85 李文波, 贾嵘, 张怀春, 等. 基于配电网一张图的运营指挥系统研究. 供用电, 2019, 36(11): 48–54.
86 黄华, 戴江鹏, 王毅, 等. 基于图数据库的电网 CIM/E 模型构建及网络拓扑. 电力系统自动化, 2019, 43(22): 122–129. [doi: 10.7500/AEPS20190424001]
87 李广野, 李伟, 王丽霞, 等. 基于图数据库的电网拓扑搜索引擎研究. 信息技术, 2019, 43(5): 155–158.
88 张冰雪, 刘婷婷, 汤亚宸, 等. 基于图数据库的电力设备全生命周期管理技术研究. 电力信息与通信技术, 2019, 17(3): 1–7.
89 Marinho SSC, Filho JSC, Moreira LO, *et al*. Using a hybrid approach to data management in relational database and blockchain: A case study on the E-health domain. 2020 IEEE International Conference on Software Architecture Companion (ICSA-C). Salvador: IEEE, 2020. 114–121.
90 Gorawski M, Grochla K. Performance tests of smart city IoT data repositories for universal linear infrastructure data and graph databases. SN Computer Science, 2020, 1(1): 31. [doi: 10.1007/s42979-019-0031-y]
91 Lehotay-Kéry P, Kiss A. Process, analyze and visualize telecommunication network configuration data in graph database. Vietnam Journal of Computer Science, 2020, 7(1): 65–76. [doi: 10.1142/S2196888820500037]
92 倪思嘉. 图数据库在商业银行反欺诈审计领域的应用. 上海商业, 2020, (7): 53–55.
93 范倬瑜. 基于图数据库 Neo4j 的信用卡反欺诈系统的设计与实现 [硕士学位论文]. 杭州: 浙江工业大学, 2020.
94 奥渊博. 基于用户信息图谱的互联网金融虚假用户信息检测系统的设计与实现 [硕士学位论文]. 太原: 太原科技大学, 2018.
95 施朝浩. 基于图特征的欺诈检测方法研究与应用 [硕士学位论文]. 杭州: 浙江大学, 2019.

96 周晓楠, 黄磊, 王飞跃, 等. 图数据库在识别重大疾病保险团伙式欺诈中的应用研究. 保险研究, 2020, (9): 92–104.
97 史旭东. 图数据库技术在信用卡反套现领域的应用. 金融电子化, 2020, (9): 81–82.
98 胡欢, 云红艳, 贺英, 等. 半自动构建扶贫领域知识图谱工具的研究. 计算机与数字工程, 2019, 47(8): 1961–1965, 2055. [doi: 10.3969/j.issn.1672-9722.2019.08.024]
99 Mezzanzanica M, Mercorio F, Cesarini M, *et al*. GraphDBLP: A system for analysing networks of computer scientists through graph databases. Multimedia Tools and Applications, 2018, 77(14): 18657–18688. [doi: 10.1007/s11042-017-5503-2]
100 吕旭明, 郑善奇, 曹丽娜, 等. 图数据库技术在电力系统信息通信资产管理中的应用. 东北电力技术, 2017, 38(11): 27–30. [doi: 10.3969/j.issn.1004-7913.2017.11.007]
101 Roy-Hubara N, Rokach L, Shapira B, *et al*. Modeling graph database schema. IT Professional, 2017, 19(6): 34–43. [doi: 10.1109/MITP.2017.4241458]
102 Muramudalige SR, Hung BWK, Jayasumana AP, *et al*. Investigative graph search using graph database. 2019 1st International Conference on Graph Computing (GC). Laguna Hills: IEEE, 2019. 60–67. [doi: 10.1109/GC46384.2019.00017]
103 张维冲, 王芳, 黄毅. 基于图数据库的贵州省大数据政策知识建模研究. 数字图书馆论坛, 2020, (4): 30–38.
104 鄂海红, 韩鹏昊, 宋美娜. 关系型数据库向图数据库的转换方法. 计算机科学, 2021, 48(10): 140–144. [doi: 10.11896/jsjkx.201100073]
105 Angles R, Thakkar H, Tomaszuk D. Mapping RDF databases to property graph databases. IEEE Access, 2020, 8: 86091–86110. [doi: 10.1109/ACCESS.2020.2993117]

(校对责编: 牛欣悦)