图数据库系统研究综述

韩浩明 / 同济大学 软件学院

摘 要：针对图数据管理的新需求，呈现出许多面向特定应用的图数据库系统。本文针对图数据库系统的相关研究进行综述。首先介绍图数据的三种类型。然后根据图数据库功能的不同将图数据库分为查询类和分析类进行介绍。最后对文章进行总结。

关键词：图数据库；图查询；图分析

图是计算机科学中的重要数据结构。由于图的逻辑表现能力很强，图被广泛的应用于化学分子结构，生物网络，社会网络以及计算机辅助设计等领域中。目前，人们主要利用图数据库对图数据进行管理。因此，近年来，图数据库呈井喷式发展，不断有新的图数据库出现。虽然可供选择的图数据库很多，但其特性各不相同，适用的范围也有很大区别。人们对图数据库的总结资料还比较少，对研究人员给予的帮助有限。

针对上述问题，我们对图数据库进行了调研，并将调研结果进行整理。结果分为两部分：第一部分是简单的介绍图数据库的背景知识包括图数据的类型介绍。第二部分是根据图数据库的功能特性将其分成查询和分析两大类，并分别以Neo4j和GraphChi为例介绍两类数据库的特性。最后对文章进行总结。

1 图数据类型

在这一章节，我们根据图数据库对自身的数据类型在设计和实现方式上各不相同，归纳出三种图数据类型：基本类型G=（V，E）、超点和超图类型。

1.1 基本图数据类型

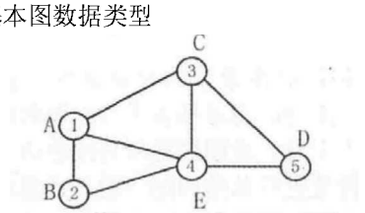

图1 图数据基本类型（无向图）

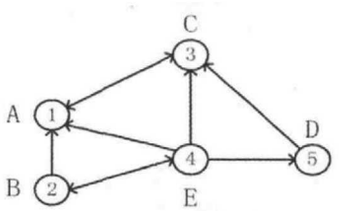

图2 图数据基本类型（有向图）

图数据的基本类型是G=（V，E）。V是图G的顶点集合，E⊆V×V是图G的边集合。节点和边都可能包含属性。图的边可以有方向，即当一张图是有向图时，对于任何的顶点u，v∈V，（u，v）≠（v，u）。早期的图数据库模型都是基于G=（V，E）这个基本图数据类型。

1.2 拓展图数据类型

虽然图的表现能力很强，但是一些复杂的事物还是不能由图的基本类型表示，所以在原来的基础上对图的基本类型进行了拓展，出现了超点和超图。

超点就是对图的节点进行功能扩充。它的概念理解为一张图中的节点表示另一种图结构，其特点就是嵌套图结构，利用这结构可以更加简单的建模属性复杂事物。超图则是对图的边进行扩充。由于普通图中限定每条边的关联结点为两个，限制了图的表达能力。现实世界中，广泛地存在着各种各样的多元联系，难以用普通图直观地表达。这时则可以用超图来建模。

其实超点和超图模型实际上是互补的，他们的出现都是为了弥补基本图数据结构的不足，更好的建模复杂对象。

2 图数据库综述

图数据库按实现的功能不同可分为查询类图数据库和分析类图数据库。两者的区别在于查询类数据库擅长根据查询条件快速输出满足该查询条件的子图信息。分析类数据库则擅长针对整个图的结构拓扑信息，分析其整体的拓扑结构特征信息，从而挖掘有用的信息。

这一章节我们将对这两类数据库进行综述。

2.1 查询类图数据库Neo4j[1-2]

随着信息时代的到来，半语义结构和面向网络的数据愈来愈庞大，以前的关系模型静态、刚性和不灵活的特点已经很难满足现在的业务需求。未解决此类问题，图数据库Neo4j出现。

Neo4j使用图相关的概念来描述数据模型，把数据保存为图中的节点以及节点之间的关系。数据主要由三部分构成：

（1）节点。节点表示对象实例，每个节点有唯一的ID区别其它节点，节点带有属性。

（2）关系。就是图里面的边，连接两个节点，另外这里的关系是有向的并带有属性。

（3）属性。key-value对，存在于节点和关系中，如图3所示。

Neo4j使用遍历操作进行查询。为了加速查询，Neo4j

中图分类号：TP311.13

会建立索引，并根据索引找到遍历用的起始节点。默认情况下，相关的索引是由Apache Lucene提供的。但也能使用其他索引实现来提供。用户可以创建任意数量的命名索引。每个索引控制节点或者关系，而每个索引都通过key/value/object三个参数来工作。其中object要么是一个节点，要么是一个关系，取决于索引类型。另外，Neo4j中有关于节点（关系）的索引，系统通过索引实现从属性到节点（关系）的映射。

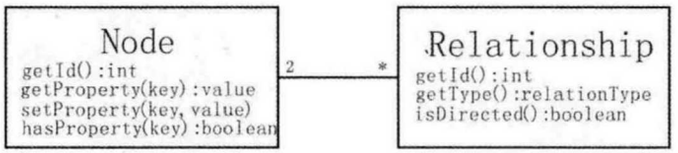

图3 节点、关系和属性三者的关系

依据内部建立的索引，Neo4j通过遍历API在图形中进行遍历，从而查找到对应结果。

系统通过设定访问条件比如，遍历的方向，使用深度优先或广度优先算法等条件对图进行遍历，从一个节点沿着关系到其他节点。另外，Neo4j可以快速的插入删除节点和关系，并更新节点和关系中的属性。

Neo4j提供了大规模可扩展性，在一台机器上可以处理数十亿节点/关系/属性的图，可以扩展到多台机器并行运行。Neo4j已经应用在高请求的24/7环境下超过3年了。它是成熟、健壮的，完全达到了部署的门槛。成为了目前最流行的图数据库。

2.2 分析类图数据库GraphChi[3]

单台PC机环境下对大规模的图数据进行处理有很多优点。比如，当对同一张图进行多个任务时，相对于分布系统复杂的扩展，单机处理则只需要简单增加PC机就可达到提高效率的功能。同时只单机系统更易于管理，对硬件的要求比较低等。但是，实现基于磁盘的单机图处理面临棘手的随机访问问题。即当系统要读取数据时，会从不同磁盘位置多次读取，所以每次读取都要消耗几毫秒，即使是现在的SSD（固态硬盘）中读取速度有了很大提升，但与内存相比仍有很大差距。因此，随机读取问题大大的降低了系统的效率。

为合理解决随机存储的问题，实现单机环境下对大规模图数据进行处理。Aapo Kyrola开发GraphChi使用PSW（Parallel Sliding Windows）方法，通过优化对外存的访问使普通计算机能对大规模图数据进行分析。PSW具体分为以下3步：

（1）加载子图。使用GraphChi中算法loadSubGraph（p）实现，将图中的顶点分成P份，加载到intervals里面，同时每个interval分别对应shard存入入边（指向顶点的边）。Shard的大小应满足可以完全加载到内存中。

（2）更新子图。GraphChi使用算法update-function对子图的边的值进行修改。

（3）将更新结果写回磁盘。GraphChi使用算法Parallel Sliding Windows每执行完一个interval后修改的边的值会被写回磁盘，代替以前的值。再到执行下一个interval时，PSW将从磁盘中读到新的值。PSW可有效的执行此操作：边从磁盘中加载到缓存中的数据块中，边作为指针指向数据块，所以修改边的同时直接修改了数据块的值。

以上面PSW为基础，GraphChi可在单机上对图进行分析操作，具体可以实现Pagerank，connected components，community detection等分析操作。下面以PageRank[4]为例。

GraphChi在预先设定好的迭代次数下对每个顶点执行更新操作完成PageRank。每次更新时，sum求和每个顶点的入边排名。最后带入公式0.15+0.85*sum求出顶点的PageRank。具体伪代码如下：

Algorithm 1：Pseudo-code of the vertex update function for weighted PageRank.

```
1 typedef: VertexType float
2 Update(vertex) begin
3   var sum 0
4   for e in vertex.inEdges() do
5     sum += e.weight*neighborRank(e)
6   end
7   vertex.setValue(0.15 + 0.85 *sum)
8   broadcast(vertex)
9 end
```

3 结束语

图数据库是针对图数据的存储系统，被业界广泛关注，各商家针对不同的需求推出了许多图数据库。本文从图数据库的分类出发，将图数据库分成查询类与分析类，对图数据库的特性进行总结归纳。随着计算机技术的不断发展，还有许多关于图数据库的挑战性工作值得我们深入研究。

参考文献：

[1]Eifrem, E.,Neo4j—the benefits of graph databases.no:sql(east),2009.
[2]Developers,N.J.,Neo4J.Graph NoSQL Database[OL].2012.
[3]Kyrola,A.,G.Blelloch,and C.Guestrin.GraphChi:Large-scale graph computation on just a PC. in Proceedings of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI).2012.
[4]Page,L.,et al.,The PageRank citation ranking:bringing order to the web.1999.

作者简介：韩浩明，男，江苏苏州人，软件工程硕士，研究方向：数据分析与挖掘。
作者单位：同济大学 软件学院，上海 200092