# 基于 Neo4j 的社交网络平台设计与实现 *

张 琳 熊斯攀

(大连海事大学航运经济与管理学院 辽宁大连 116026)

**摘 要**:[目的/意义]应用 Neo4j 探讨对大规模社交网络数据进行有效的存储和管理。[方法/过程]Neo4j 是一个高性能的 NoSQL 数据库,支持图结构数据的存储和复杂查询。以 Neo4j 图形数据库为基础,采用 B/S 三层架构,结合 PHP、H5 等网络编程技术,设计并实现了 Linked-US 社交网络平台。[结果/结论]该平台可以更好的管理和更新社交网络数据及其内在关系,支持类似"好友的好友"等多层复杂查询。

**关键词**:Neo4j;NoSQL;图形数据库;社交网络数据

**中图分类号**:TP315 **文献标识码**:A **Adoi**:10.3969/j.issn.1005-8095.2018.08.013

## Design and Implementation of Social Network Platform Based on Neo4j

Zhang Lin Xiong Sipan

(School of Maritime Economics and Management of Dalian Maritime University, Dalian Liaoning 116026)

**Abstract:** [**Purpose/significance**] The paper is to apply Neo4j to explore effective storage and management for the large-scale social network data. [**Method/process**] Neo4j, which is a high performance of NoSQL database, supports storage and complex query for graph-structured data. Based on Neo4j, this paper designs and implements a social platform named Linked-US by using B/S three-tier architecture combined with some network programming technologies, such as PHP and H5. [**Result/conclusion**] Linked-US can effectively manage and update social network data, and support multi-layer complex queries like "friends of friends".

**Keywords:** Neo4j; NoSQL; graph database; social network data

社交网络数据是一种关于实体和实体之间关系的数据,多为半结构化或非结构化,所以与传统的数据类型相比,社交网络数据更加复杂。随着互联网和移动互联网的快速发展,社交网络数据变化迅速,需要频繁的查询,社交图谱也变得异常庞大,这导致传统的关系型数据库 RDBMS 越来越难以满足人们对大规模数据存储、并发读写及处理方面的要求。Neo4j 图形数据库善于处理大量复杂、动态、互连接和低结构化的数据[1],为解决这些问题提供了一个新的方向。它将数据作为图形处理,能够保存数据的自然图形结构,同时具有强大的图形遍历功能,支持复杂的查询,能够极大提高查询的效率[2]。

Neo4j 图形数据库在社交网络中有着广泛的应用。文献[3]基于 Neo4j 对社交网络进行了分析,并将 SKIP LIST 索引算法运用到 Neo4j 中以缩短检索时间,提升社交网络数据分析的效率。文献[4]通过实验从性能、代码可读性、代码量等方面比较了Neo4j 的图查询语言 Cypher、Gremlin 和 Native Acess 及 MySQL 使用的 JPA,结果表明 Neo4j 可以作为关系数据库的高性能替代品,特别是在处理社交网站上大量高度互联的数据时。文献[5]为了选择合适的数据库管理系统来满足移动社交网络对大量复杂数据的处理需求,选取了近百万级的数据对 MySQL 和 Neo4j 进行了性能测试,得到了与文献[4]类似的结论,即对于大型数据集,Neo4j 有着关系数据库无法比拟的性能优势。

因此,为了更好地适应社交网络数据的动态变化和关系复杂的特点[6],支持类似"好友的好友"这样多层复杂查询,本文采用 Neo4j 图形数据库来存储和展示社交网络中的海量社交数据。

**收稿日期**:2018-04-02

* 本文系中国博士后科学基金资助项目"大数据环境下基于异构图的文本聚类在自动文摘中的应用"(项目编号:2015M571292)成果之一。

**作者简介**:张琳(1984—),女,博士,讲师,主要研究方向为文本挖掘、自动文摘;熊斯攀(1994—),男,本科,主要研究方向为电子商务。

## 1　图形数据库 Neo4j

### 1.1　Neo4j 图存储结构

Neo4j 是一个高性能的 NoSQL 图形数据库,也是一个基于磁盘的、嵌入式的、支持海量数据的、具备完整 ACID 特性和迅速图查询特点的 Java 持久化引擎[7-8]。

Neo4j 的基本单元是节点、关系和属性。每个节点代表一个实体,可以有 0 个或多个属性,每个属性都以 key-value 键值对的形式存在[8]。在 2.0 版本中,Neo4j 引进了节点标签的概念,用于识别不同类型的节点[2]。关系表明 2 个节点之间的关联,在 Neo4j 中,每 1 个关系表现为 1 条带箭头的线,由起始节点指向终止节点。与节点一样,关系也可以有属性和标签。图 1 是社交网络中的 Neo4j 图形存储模型示例,其中,"Olivia""Lizzy"和"Mason"表示 3 个节点,节点标签都是 Person。以"Olivia"节点为例说明,该节点拥有 4 个属性,每个属性都是以 key-value 键值对的形式存在的,如"Olivia"节点的 gender 属性的属性值是 female,age 属性的属性值是 58,name 属性的属性值是 Olivia,id 属性的属性值是 8。

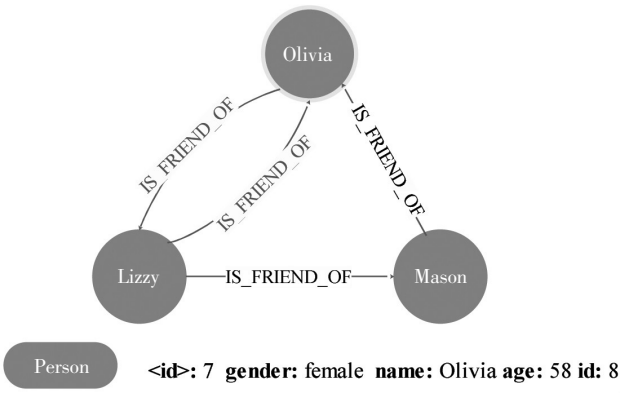


图 1　Neo4j 网络关系图存储示例

### 1.2　Neo4j 与 RDBMS

在关系型数据库系统中实体与实体之间的关系有一对一、一对多和多对多 3 种[9]。由于在对多对多数据模型查询时会产生大量的表连接,而 SQL 的连接操作(Join Operations)不仅费时且复杂,会极大影响 RDBMS 的性能,所以 RDBMS 并不擅长处理多对多数据模型,尤其是在处理大规模数据集时[2,10]。Neo4j 重点解决了这个问题。与 RDBMS 不同,Neo4j 不使用表、列和外键,而是围绕图形进行数据建模。在查询数据时,Neo4j 会以相同的速度遍历节点与边,且遍历速度与构成图形的数据量没有任何关系[1,7]。

在可扩展性方面,RDBMS 严格的模式约束使得对已有数据库的扩展变得非常困难,而 Neo4j 可以动态增加节点和节点之间的关系,并可轻易扩展至上亿级别的节点和关系,而且不需要重构数据库,因此,相比于 RDBMS,Neo4j 具有良好的可扩展性[6,11]。

此外,Neo4j 是无固定模式的,这使得它可以拓展到多台服务器上并行运行。

## 2　基于 Neo4j 的社交网络平台需求分析及设计

### 2.1　需求与角色分析

(1)功能需求

基于人们对社交的需求,Linked-US 平台提供了"览主页""荐好友""推动态""写动态""关于我"和"联系"6 个模块。其中,"览主页"模块为用户注册和登录平台提供接口;"荐好友"模块聚焦用户共同爱好,通过"好友的好友""好友的好友的好友"等多层复杂查询为用户推荐可能认识的朋友;"推动态"模块可基于位置、时间、关系层等为用户推送好友的动态;通过"写动态"模块可进行个人动态发布;通过"关于我"模块可对个人基本信息、个人好友、个人动态等信息进行管理;"联系" 模块则为用户提交关于平台建设等相关反馈信息提供接口。

借助于 Neo4j,可以对用户及其好友、动态等信息进行管理,为以后进行精准广告推广[12]和在线社交影响力分析[13]等社交网络挖掘提供基础。

(2)数据需求

在 Linked-US 平台上,设置了 4 种类型的节点和 5 种类型的关系。4 种节点类型分别是:user(普通用户)、admin(管理员)、moment(动态)和 feedback(反馈意见)。4 种关系类型分别是:BACK_FEED(反馈意见)、SHARE_MOMENT(发布动态)、IS_FRIEND_OF(好友)、COMMENT(点赞/评论动态)和 MAINTAIN(维护)。

节点之间的关系如图 2 所示。

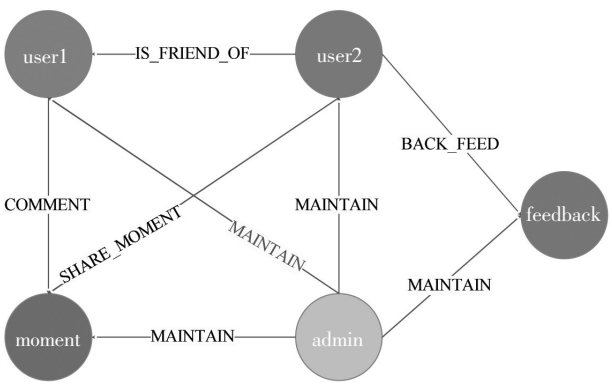


图 2　节点之间的关系

(3)角色分析

Linked-US 平台的使用者主要有 2 类：普通用户和管理员。借助于平台,普通用户可以进行个人展示、写动态、对平台建设提出反馈意见等。管理员分为普通管理员和超级管理员。普通管理员主要负责对用户发布的动态的内容进行筛查和监督，对用户提交的反馈信息进行筛选，以及对非法用户和越权访问进行封堵。超级管理员则主要负责对普通管理员的权限进行管理。

## 2.2　系统总体架构

Linked-US 平台的设计思路为：以提高用户体验度、用户粘性为基本点,以个性化服务为主题,充分利用 PHP、H5 等网络编程技术，结合 Neo4j 图形数据库,为用户提供一个便捷的在线交友平台。

Linked-US 平台采用 B/S 结构,总体架构如图 3 所示。

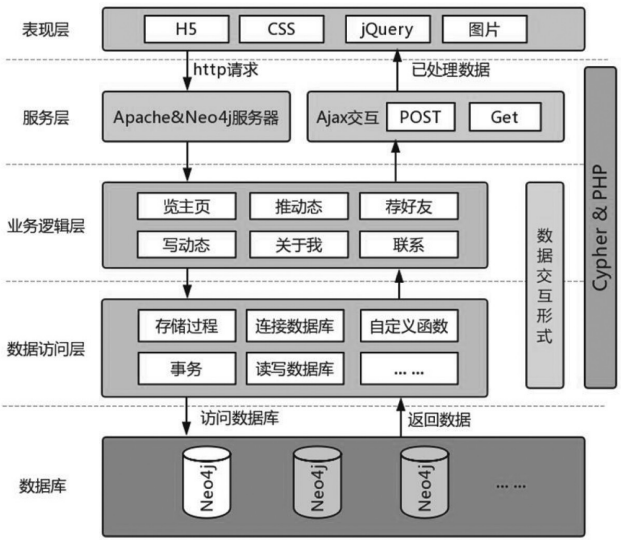


图 3　Linked-US 系统架构

# 3　Linked-US 社交网络平台实现

## 3.1　表现层实现

用户在注册时需要提供 3 方面信息:账户相关信息 ACCOUNT SETUP(如账号、密码),社交资料信息 SOCIAL PROFILES(如 QQ、邮箱、手机号等),以及个人信息 PERSONAL DETAILS(如真实名字、个性签名等),如图 4 所示。

## 3.2　逻辑层实现

(1)用户类实现

在推荐好友实现方面,本文通过构建 Cypher 查询,对当前用户的社交关系即 user->user->user 2 层关系(即当前用户好友的好友)、user->user->user->user 3 层关系(即认识当前用户好友的好友的用户)

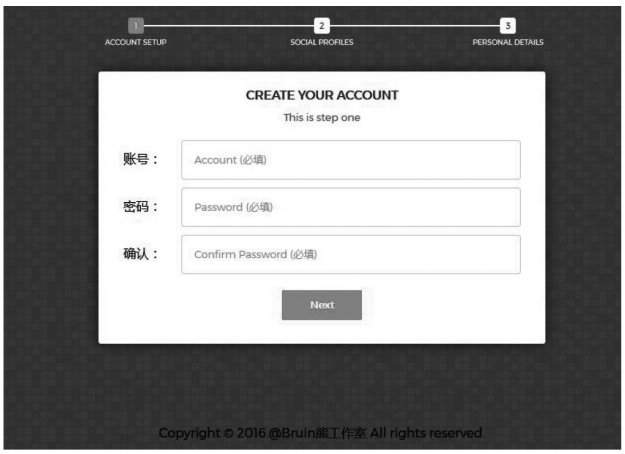


图 4　Linked-US 注册页

以及社交网络中的关键人物(即被关注数多的用户)进行遍历，并将所得结果分成“好友好友”“认识好友”和“关键人物”3 类好友进行推荐,如图 5 所示。

具体来说,本文以当前用户 user1 为起点,通过构建如下 Cypher 查询获得 user1 好友的好友 user2,然后从 user2 中过滤掉已与 user1 建立好友关系的人,剩下的 user2 即是图 5“好友好友”选项卡为 user1 推荐的好友。

```
MATCH(u:user)　WHERE u.account=user1
Match(u-[:IS_FRIEND_OF]->()-[:IS_FRIEND_OF]-> user2)　RETURN user2
```

同理可基于 3 层关系查询获得 user1 的潜在好友为其推荐,即图 5“认识好友”选项卡为 user1 推荐的好友。

图 5“关键人物”选项卡则是通过下边 Cypher 查询获得社交网络中 IS_FRIEND_OF 关系最多的用户,在过滤掉已经是 user1 好友的用户外,选取 top10 个用户作为关键人物推荐给 user1。

```
Match r=(n:user)-[:IS_FRIEND_OF]->(u:user)　RETURN count(r),u
ORDER BY count(r) DESC
```

(2)动态类 moment 的实现

在动态发布实现方面，本文根据用户信息及用户发布动态的时间、地理位置和内容等创建动态 moment 节点，并建立 user 节点和 moment 节点之间的 SHARE_MOMENT 关系。

在动态推送实现方面,本文根据 user->user1 层关系(即好友关系)、user->user->user 2 层关系(即好友的好友关系)以及动态发布的时间、位置、浏览量五种情况进行动态查询,并将所得结果分成“好友动态”“好友好友动态”“今日最佳”“附近动态”和“自

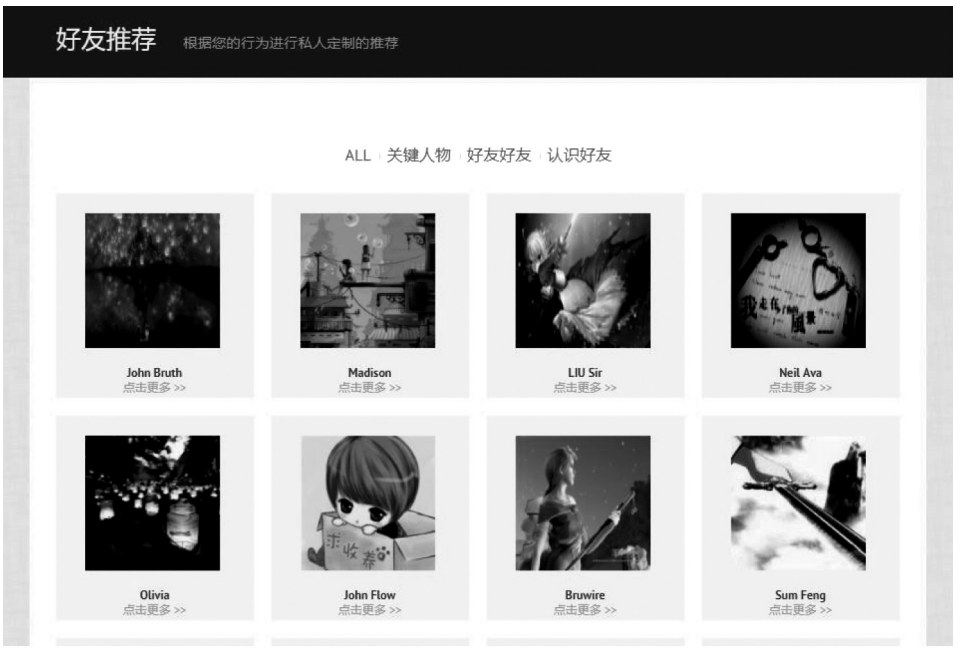


图 5　好友推荐页

己动态”6 类进行推送,如图 6 所示。其中,ALL 是将 5 种动态合并按照动态发布的时间降序输出。对于推送的动态,用户可以进行浏览、点赞、评论等操作。


图 6　推送动态页

下边以“今日最佳”为例来说明在动态推送方面相关实现。

“今日最佳”是基于动态评论量、浏览量进行的推荐。通过构建如下 Cypher 查询语句获得社交网络中浏览量最多的动态推送给用户。

```
MATCH ()-[rel:ACT_COMMENT]->(act:moment)
RETURN act.subject as subject,act.location as location,act.message as message,act.activeImg as activeImg,act.date as date,act.viewtimes as viewtimes,count(rel) as comms,id(act) as id
ORDER BY count(rel),act.viewtimes,act.date DESC
LIMIT 10
```

(3)**反馈类** feedback **的实现**

**根据用户提交的反馈信息创建** feedback **节点,并将** user **节点与该** feedback **节点关联;如果用户是匿名提交则将空属性的** user **节点指向** feedback **节点。**

### 3.3　数据层实现

(1)**加载** neoclient **组件**

**新建** composer.json **文件,通过命令行窗口输入** composer update **命令加载必需的文件框架,然后在** composer.json **文件中写入如下代码,并再次运行** composer update **命令加载所需要的包,完成** neoclient **的安装。**

```
{
    "require":{
        "graphaware/neo4j-php-client":"^4.0"
    }
}
```

(2)PHP **连接** Neo4j

**通过如下语句连接** Neo4j **数据库,以便进行社交网络数据的存储和构建** Cypher **语句进行社交关系的查询。**

```
ClientBuilder::create()->addConnection('default','http://neo4j:username@server_ip:port')->build();
```

**基于** Neo4j **的** Linked-US **社交网络平台的节点和关系存储图如图** 7 **所示。**

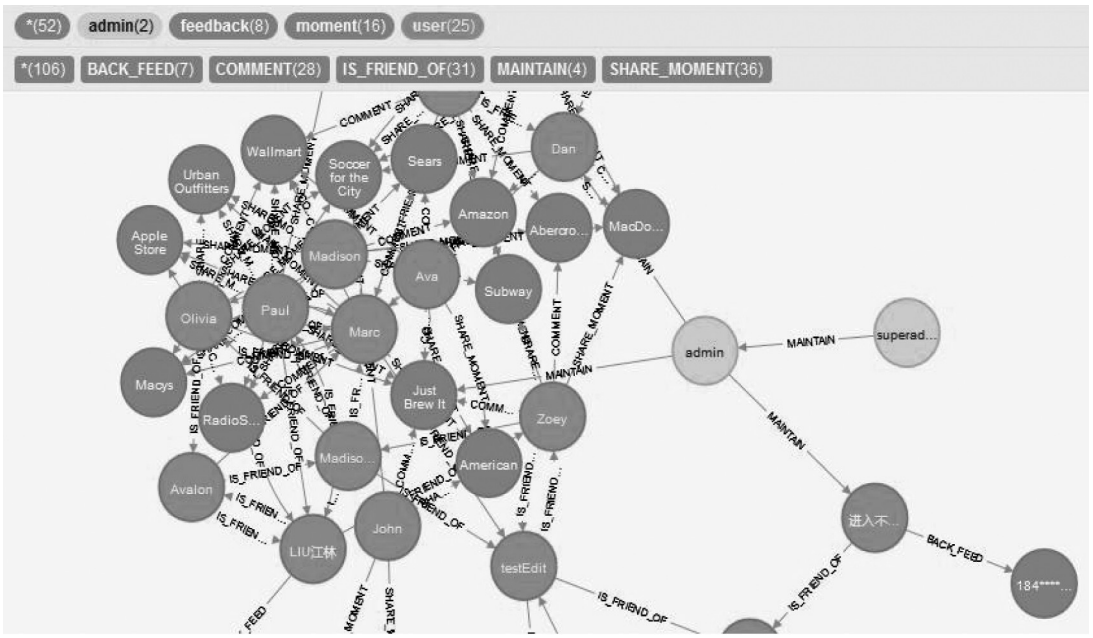


**图** 7　Linked-US **平台节点关系存储图**

## 4　结论

**针对社交网络数据间关系复杂,且动态变化,需频繁查询的问题,本文基于** Neo4j **图形数据库,结合** PHP、H5 **等技术,通过需求与角色分析,架构与数据库设计,表现层、逻辑层与数据层的交互,设计并实现了** Linked-US **社交网络平台,以实现海量社交网络数据的存储和多层复杂查询,为社交网络分析提供基础。在以后的研究中,将对** Linked-US **平台进行以下几方面的完善。**

(1)**后期在将平台转移到多服务器时,需要开发一个基于** B/S **架构的后台管理系统,并能随** Linked-US **项目进行迁移。**

(2)**对防止** SQL **注入等攻击进行改进。**

(3)**建立完善社交网路中确定关键人物的算法加权机制。**

**参考文献**

[1]　**王余蓝. 图形数据库** Neo4j **的内嵌式应用研究**[J]. **现代电子技术**,2012,35(22):36-38.

[2]　VUKOTIC A,WATT N,ABEDRABBO T,et al. Neo4j **实战**[M]. **张秉森,孔倩,张晨策,译. 北京:机械工业出版社**,2016:3-17,43-46.

[3]　**张凤军. 基于** Neo4j **图数据库的社交网络数据的研究与应用**[D]. **长沙:湖南大学**,2016.

[4]　HOLZSCHUHER F,PEINL R. Performance of Graph Query Languages:Comparison of Cypher,Gremlin and Native Access in Neo4j[C]New York:ACM,2013:195-204.

[5]　**尤惠芬. 移动社交网络中的数据库应用**[J]. **山西青**

年管理干部学院学报,2013,26(3):106-108.

[6] 王余蓝. 图形数据库 NEO4J 与关系据库的比较研究[J]. 现代电子技术,2012,35(20):77-79.

[7] 陆嘉恒. 大数据挑战与 NoSQL 数据库技术[M]. 北京:电子工业出版社,2013:54-55.

[8] 王红,张青青,蔡伟伟,等. 基于 Neo4j 的领域本体存储方法研究[J]. 计算机应用研究,2017,34(8):1-6.

[9] 陈锐. 网络结构与数据库模式在陕西科技信息网中的应用[J]. 情报杂志,2004,(10):57-58,61.

[10] 廖理. 基于 Neo4j 图数据库的时空数据存储[J]. 信息安全与技术,2015,6(8):43-44,56.

[11] 李桃陶,周斌,王忠振. 基于社交网络的图数据挖掘应用研究[J]. 计算机技术与发展,2014 24(10):6-11.

[12] 吴保来. 基于互联网的社交网络研究 [D]. 北京:中共中央党校,2013.

[13] 吴信东,李毅,李磊. 在线社交网络影响力分析[J]. 计算机学报,2014,37(4):735-752.