

基于位置信息的移动 SNS 数据动态划分复制算法

王青芸 程春玲

(南京邮电大学计算机学院 南京 210003)

摘 要 现有社交网络数据划分算法大多关注于好友关系和交互关系,忽略了位置信息,造成基于位置查询的响应时间较长。针对该问题,设计了一种移动社交网络双层社交图模型,该模型考虑了移动社交网络中用户交互行为的位置依赖性特点;并在此基础上提出了一种基于位置信息的移动社交网络数据动态划分复制算法 MSDPR,该算法采用改进的 K-Means 算法对位置信息进行聚类,再根据聚类结果对数据进行划分,并利用社交关系进行数据的复制。实验结果表明:MSDPR 算法在移动社交网络环境下能够有效地提高本地访问率,降低访问延迟,并且在动态加入数据时具有较好的适应性。

关键词 移动社交网,分布式存储,动态划分复制,位置信息,用户交互

中图分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.03.046

Mobile SNS Data Dynamic Partitioning and Replication Algorithm Based on Location Information

WANG Qing-yun CHENG Chun-ling

(College of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract The existing social network data partitioning algorithms focus on the social relationship and interaction, without considering location information, which results in the long response time of location-based queries. To solve this problem, we designed a two-layer graph model of mobile social network which takes the location dependency of the user interaction behavior into account. We proposed a mobile SNS data dynamic partitioning and replication algorithm based on location information—MSDPR. MSDPR divides data based on the clustered results generated by an improved K-Means clustering algorithm, and then replicates data by using the social relationships. Experiments reveal that MSDPR can effectively improve the efficiency of the local access and reduce the latency of access in the mobile social network. Moreover, it also has better adaptability when adding data dynamically.

Keywords Mobile social network, Distributed storage, Dynamic partitioning and replication, Location information, User interaction

1 引言

据统计,截至 2015 年 6 月,我国手机网民人数已达到 5.93 亿^[1],越来越多的网民使用移动终端设备进行社交。TalkingData 移动数据研究中心预计目前国内社交应用覆盖终端数已达 10.41 亿^[2],日益增长的数据量给后台支撑系统带来了巨大的压力。

系统扩展是每个应用发展到一定阶段必然会遇见的难题,一些社交网络由于未能合理地解决系统扩展的问题,导致了整个应用的瘫痪^[3]。后台支撑系统扩展的方法主要分为两大类:垂直扩展和水平扩展。垂直扩展是通过升级硬件配置来实现的,由于移动社交网络应用规模的不可确定性,采用垂直扩展不仅代价大,而且随着应用规模的进一步扩大,需不断地进行垂直扩展。水平扩展具有灵活、代价小的特点,只需添加新的机器,并将一部分数据及业务请求分配给新的机器即可

实现扩展。采用水平扩展的方法时需对数据进行分布式存储,数据划分复制算法的好坏直接影响着支撑系统的性能表现。

移动社交网络服务与传统社交网络服务的不同主要体现在其对位置信息的充分利用和数据时效性强两个方面。移动社交网络服务大量地加入基于位置信息的功能,如新浪微博的“周边”功能,通过它可以获取身边的人发出的微博信息;手机 QQ 的“附近”功能。使用它通过位置信息可以添加附近的人为好友。数据时效性强归功于移动设备的便携性,方便了人们随时访问,增加了访问频率。

本文针对移动社交网络服务的特点,考虑如何设计一种数据划分复制算法,以使划分复制结果能够较好地支持移动社交网络服务。本文的主要贡献如下:

(1)针对移动社交网络中用户交互行为对位置信息的依赖性,结合二分图思想,提出了一种基于位置的移动社交网络双层图模型;

到稿日期:2015-11-24 返修日期:2016-02-26

王青芸(1991—),女,硕士生,主要研究方向为数据管理,E-mail:wangqingyun2013@sina.com;程春玲(1972—),女,教授,CCF 会员,主要研究方向为数据管理、云计算中的资源管理和优化等。

(2)提出了一种基于位置信息的移动 SNS(Social Networking Services)数据动态划分复制算法 MSDPR(A mobile SNS data dynamic partitioning and replication algorithm based on location information),该算法采用改进的 K-Means 算法对用户位置信息层进行聚类,然后根据聚类结果实现数据划分,并通过计算用户的副本收益来选择副本存放的位置。

2 相关工作

在对数据进行分布式存储时,数据划分是负载均衡的关键,并且直接影响着数据查询的响应速度。数据划分主要分为两种:水平数据划分和垂直数据划分。垂直数据划分算法利用用户的查询操作只涉及部分属性的特征,将一个关系划分为多个属性集。国内外研究人员主要提出了以下几种算法:文献[4]提出了 BEA(Bond Energy Algorithm)算法,它通过计算属性间的亲密度,将关系中的属性进行划分,这种方法能够得到最优的划分结果,但需要耗费很长的计算时间;文献[5]提出了基于事务的垂直划分算法,这种算法会选择“重要的”事务进行优化。对于大数据量的移动社交网络,若采用垂直数据划分方法,各分片易出现数据量太大而仍需要水平划分的情况,因此本文采用水平数据划分方法。

水平数据划分算法中最为常见的是基于哈希的方法,即在确定分片数后,采用哈希函数对数据进行划分。由于基于哈希的方法十分简单,因此很多知名的社交网络采用了这种方法,如 Twitter 使用的 Gizzard 和 Facebook 使用的 Cassandra^[6]。虽然这种方法可以避免数据量的倾斜,但这种简单的机制不能捕获数据间的关系并且有可能导致通信代价过高等问题。

移动社交网络数据是以图为基础结构的数据,对于图数据这种数据间关系丰富的数据类型来说,采用水平数据划分方法时必需对数据间的关系进行划分,因此如何降低划分后子图间的连通性成为了关键。一些水平数据划分算法将社交网络特性作为数据划分的依据,如好友关系和交互关系。文献[8]提出的 SPAR 方法在划分数据时尽可能多地保留好友关系,复制时保证所有的单跳邻居数据存放在同一台服务器上,以保证数据的本地语义。这是首次利用社交网络特性来解决社交网络中数据的划分复制问题。文献[9]通过分析在线社交网络的数据集,发现社交网络中一般用户只与他们好友中的 22.03% 进行交互,这一点是符合帕累托分布属性的。因此 SPAR 方法会造成大量的数据冗余,在进行写操作时副本写操作的代价偏大。文献[10]在 SPAR 方法的基础上利用了在线社交网络中用户的社交连通性信息以及 Gossip 技术,有效地降低了副本拷贝代价。在社交网络中,交互响应的快慢直接反映了系统性能的好坏,文献[8]和文献[10]都采用了好友关系作为数据划分的依据,然而好友关系并不能直接地反映出交互关系,因此使用好友关系作为划分依据的数据划分算法,其划分结果极易出现负载不均、分片间交互频繁的现象。

为此,文献[9]提出了在线动态划分复制算法 WEPAR,其综合考虑了社交网络中的读操作、写操作以及副本拷贝代

价,但并没有考虑存储容量的限制。文献[11]提出的 COSI 方法根据查询记录来划分社交网络,可以有效地优化静态查询工作负载,但很难应用于动态查询工作负载中。近两年,将超图理论应用到水平数据划分中成为了一种趋势,文献[12]提出了一个关联数据放置方案,其改善了关联数据的协同定位和局部数据服务,并采用超图划分技术进行数据项的划分。文献[13-14]都采用超图对负载情况进行记录,并以此为依据进行数据划分和复制。但超图模型在构造时代价太大,且不易维护,并不适用于移动社交网络这种动态性较强的情景。

综上所述,上述的数据划分复制算法通常依据好友关系和交互关系,并不完全适用于移动社交网络环境下基于位置查询较多的情况。因此,本文设计了一个移动社交网络双层社交图模型,并在设计划分复制算法时将位置信息作为一个重要依据。

3 移动社交网络双层社交图模型

由于移动社交网络中存在大量的基于位置信息的服务,因此在对移动社交网络用户的社交关系进行建模时除了保留一般的社交关系外,还着重考虑如何表达用户的位置信息。本文以社交网络有向图模型为基础,采用二分图的思想对移动社交网络中用户位置的信息进行记录,从而对移动社交网络用户社交关系进行建模。

定义 1 社交网络有向图模型 $SG=[V,E]$,其中 $V=(v_1,v_2,\dots,v_n)$ 表示社交网络中所有的用户, n 为用户数, $E=(e_1,e_2,\dots,e_m)$ 表示社交网络中用户间的社交关系,其中 $e_q=(hI_q,eI_q)$ 表示社交网络中的第 eI_q 个用户是第 hI_q 个用户的好友。

社交网络有向图模型可以形象地描绘出社交网络中的社交关系,但用其对移动社交网络进行建模则会忽略对位置信息的描述,一种简单的解决方法是在社交网络有向图模型中的用户节点上增加位置属性,但是这种方式在进行划分时并不易处理。因此本文采用二分图的思想对社交网络有向图模型进行改进,以实现移动社交网络位置信息进行记录。

定义 2 二分图 $G_a=(V_a \cup V_b, E_a)$,其中 $V_a \cap V_b = \emptyset$, G_a 中的每一条边的两个顶点分别属于 V_a 和 V_b 。

定义 3 移动社交网络双层社交图 $G_m=(V_s \cup V_p, E_s \cup E_p)$ 。

(1) $V_s=\{v_{s1},v_{s2},\dots,v_{sf}\}$ 表示移动社交网络中所有的用户,处于移动社交网络双层社交图的下层, f 为用户数, $v_{sy}=(uData_{sy},uCopy_{sy},uTag_{sy})$ 表示第 y 个用户,其中 $uData_{sy}$ 表示第 y 个用户的数据量大小, $uCopy_{sy}$ 表示第 y 个用户的副本数, $uTag_{sy}=1$ 表示用户常驻位置发生变化,否则 $uTag_{sy}=0$ 。

(2) $V_p=\{v_{p1},v_{p2},\dots,v_{pm}\}$ 表示移动社交网络中所有的位置信息,处于移动社交网络双层社交图的上层, m 表示位置信息顶点数, $v_{pi}=(longitude_{pi},latitude_{pi},pData_{pi})$ 表示第 i 个位置顶点,其中 $longitude_{pi}$ 为经度, $latitude_{pi}$ 为纬度, $pData_{pi}$ 表示第 i 个位置顶点连接的用户的的数据量总和。

(3) $E_s=\{e_{s1},e_{s2},\dots,e_{sc}\}$ 为有向边集合,表示移动社交网络中用户间的社交关系,处于移动社交网络双层社交图的下

层内部,其中 c 为该类边的总数, $e_{sj} = (hI_{sj}, eI_{sj})$ 表示移动社交网络中的第 eI_{sj} 个用户是第 hI_{sj} 个用户的好友, $eI_{sj} \in V_s$, $hI_{sj} \in V_s, eI_{sj} \neq hI_{sj}$ 。

(4) $E_p = \{e_{p1}, e_{p2}, \dots, e_{pd}\}$ 为无向边集合, 表示移动社交网络中用户的常驻位置信息, 处于移动社交网络双层社交图的上下层之间, 其中 d 为该类边的总数, $e_{pz} = (hI_{pz}, eI_{pz})$ 表示移动社交网络中第 hI_{pz} 个用户的位置信息是 $v_{peI_{pz}}$ 。其中, 用户的常驻位置可以采用两种方法进行确定: 1) 移动社交应用用户进行手工设置; 2) 统计用户签到位置信息, 选择签到次数最多的位置作为常驻位置。本文以月为单位, 采用第二种方法确定、调整用户的常驻位置。

图 1 示出了一个简单的移动社交网络, 图中 6 个白色的顶点表示移动社交网络中有 6 个用户, 5 个灰色的顶点表示该移动社交网络中用户的位置信息。白色顶点间带箭头的边表示用户间的好友关系, 灰色顶点与白色顶点间的边表示用户所处的常驻位置。例如, 图 1 中的边 e_{s1} 表示用户 6 是用户 2 的好友, 边 e_{p1} 表示用户 1 的位置在 (118.78, 32.09)。

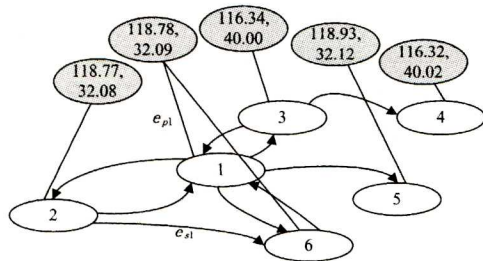


图 1 一个简单的移动社交网络双层社交图

4 基于位置信息的移动 SNS 数据动态划分复制算法

4.1 移动 SNS 数据动态划分复制算法的问题描述

数据划分将数据进行逻辑上的划分, 以使其能够进行分布式存储。对于图数据这类耦合性较强的数据, 数据划分的好坏直接影响着系统的处理效率, 必须尽量减少子图间的连通性, 且保证子图负载均衡, 避免出现大的偏斜。除此以外, 为了减少跨片访问次数, 需要对数据进行复制, 同时需要避免因副本数过多而给系统造成过重的负担。由于本文的研究背景为移动社交网络, 因此划分复制算法除了需使各分片间的连通程度最小化、各分片负载均衡外, 还希望使基于位置的查询代价最小化。问题描述如下:

对于一个移动社交网络双层社交图 G_m , 将点集 V_s 划分为 k 个子集, 表示为 $P = \{P_1, P_2, \dots, P_k\}$, 并使其满足各分片间的连通程度最小化、各分片负载均衡、基于位置的查询代价最小化。该问题是一个多目标优化问题, 其数学描述如下:

$$\min f_{connectivity}(P_1, P_2, \dots, P_k) = \sum_{j=1}^c \delta_j \quad (1)$$

$$\min f_{loadbalance}(P_1, P_2, \dots, P_k) = \Gamma^{interaction} = \frac{2}{(k-1)\Delta^{interaction}}$$

$$\sum_{pid=1}^k pid \times \lambda_{pid}^{interaction} - \frac{k+1}{k-1} \quad (2)$$

$$\min f_{location\ query}(P_1, P_2, \dots, P_k) = \sum_{i=1}^m \tau_i \quad (3)$$

$$s. t. \bigcup_{1 \leq pid \leq k} P_{pid} = V_s \quad (4)$$

$$\sum_{v_{sj} \in P_{pid}} uData_{sj} \leq LimV, 1 \leq pid \leq k \quad (5)$$

其中, 式(1)表示各分片间的连通程度, 值越小表示子图间的连通性越弱, 当 $\delta_j = 1$ 时, 表示第 hI_{sj} 个用户和第 eI_{sj} 个用户被划分到不同的分片中, 否则 $\delta_j = 0$; 式(2)采用基尼系数 $\Gamma^{interaction}$ 来表示移动社交网络划分后各分片负载均衡的程度, 值越小表示越均衡, $\Delta^{interaction}$ 表示所有分片负载之和, $\Delta^{interaction} = \sum_{pid=1}^k \lambda_{pid}^{interaction}$, 其中 $\lambda_{pid}^{interaction}$ 表示第 pid 个分片的负载, $\lambda_{pid}^{interaction} = \sum_{j=1}^c \omega(j, pid)$, 其中 $\omega(j, pid) = \begin{cases} 1, & eI_{sj} \in P_{pid} \\ 0, & eI_{sj} \notin P_{pid} \end{cases}$; 式(3)表示基于位置的查询代价, 值越小表示基于位置的查询代价越小, 其中 τ_i 表示位置信息顶点 v_{pi} 所连接的用户顶点覆盖的分片数目; 式(4)保证划分结果包含所有用户; 式(5)保证各分片的数据量不超过最大限制 $LimV$ 。

由于该问题是一个多目标优化问题, 并不存在唯一的全局最优解, 因此需要求解出一个可以应用到实际情形的最终解。本文的应用背景为移动社交网络, 在该环境下存在大量基于位置的查询操作, 因此本文选择式(3)作为最重要的优化目标来设计算法。

4.2 算法描述

定义 4(副本收益 $copy_lucre(y, pid)$) 表示第 y 个用户在第 pid 个分片上建立副本可获得的收益, 其值等于第 y 个用户在第 pid 个分片上的好友数目。

MSDPR 算法首先判断移动社交网络双层社交图是否存在, 若不存在, 则依据历史信息构建移动社交网络双层社交图; 若存在, 则直接进行划分复制操作, 过程如下:

(1) 采用改进后的 K-Means 算法对移动社交网络双层社交图的上层进行聚类, 设置移动社交网络双层社交图上层中顶点度排名前 k 的顶点作为初始中心点, 使用顶点间经纬度的欧氏距离作为聚类属性, 最终将上层顶点聚为 k 类。

(2) 根据聚类结果, 将移动社交网络双层社交图下层中 $uTag$ 属性等于 1 的顶点进行划分调整, 即将用户顶点划分到其所连接的位置顶点所属的分片中。

(3) 依次计算 $uTag$ 属性等于 1 的顶点, 若在其他划分中建立副本获得的副本收益大于设定值 $copy_lucre0$, 且该用户顶点副本数少于 3 时, 则进行副本拷贝。

由于移动社交网络是动态变化的, 因此需要对移动社交网络双层社交图进行更新, 并选择合适的时机进行移动社交网络数据划分调整。设置分片调整阈值 cha_0 , 在每一次数据划分结束时初始化 $cha = 0$, cha 用于判断是否需要移动社交网络数据分片的动态调整。 cha 取值更新规则如下:

(1) 新用户 $v_{s(f+1)}$ 加入时, 令其 $uData_{s(f+1)} = 0$, $uCopy_{y(s(f+1))} = 0$, $uTag_{sy} = 1$, 根据用户注册时的位置信息, 建立新边 $e_{p(d+1)}$, 此时 $cha = cha + \psi_{new}$, 其中 $\psi_{new} > 0$ 。

(2) 用户注销账号时, 将对应的顶点以及相关的边删除, cha 值不变。

(3) 用户常驻位置发生变化时, 令该用户的 $uTag_{sy} = 1$, $cha = cha + \psi_{position}$, 其中 $\psi_{position} > 0$ 。

当 $cha > cha_0$ 时, 采用 MSDPR 算法对 G_m 中 $uTag_{sy} = 1$

部分的图重新进行划分复制。

MSDPR 算法的伪代码如算法 1 所示。

算法 1 MSDPR

```

Input: mobile social network two-layer graph  $G_m$ ;
Setting the number of partitions  $k$ ;
Setting the maximum number of copies  $\max\_copies$ ;
while( $cha > cha_0$ ) {
    //选取前  $k$  个位置聚集点作为初始中心;
    list centriole_list = Select_init_centriole( $V_p, k$ );
    do{//对移动社交网络双层社交图的上层顶点进行聚类;
        centriole_list = new_centriole_list;
        for each ( $v$  in  $V_p$ ) {
            best_centriole->distance = MAX;
            best_centriole->class = Undefined;
            for each (centriole in centriole_list) {
                //计算欧氏距离
                double distance = Euclidean_distance( $v, centriole$ );
                if(best_centriole->distance > distance) {
                    best_centriole->distance = distance;
                    best_centriole->class = centriole->tag;
                }
            }
            category_list[best_centriole->class].Add( $v$ );
        }
        new_centriole_list = Relocate_centriole(category_list, centriole_list,  $k$ );
    } while(!Is_centriole_stable(new_centriole_list, centriole_list))
    //根据聚类结果对移动社交网络双层社交图的下层进行划分复制;
    for each ( $v$  in  $V_s$ ) {
        if( $v->uTag == 1$ ) {
            int partition_id = find_partition_id( $v, E_p$ );
             $P_{partition\_id}$ .Add( $v$ );
            //如果用户副本数小于最大限制,执行复制操作;
            If( $v->uCopy < \max\_copies$ ) {
                for each ( $p$  in  $P^-$ ){//逐一判断不含有  $v$  的分片;
                    int copy_lucre = Calculate_copy_lucre( $v, p$ );
                    if(copy_lucre > copy_lucre0) {
                         $P_p$ .Add( $v$ );
                         $v->uCopy++$ ;
                    }
                }
                If( $v->uCopy \geq \max\_copies$ ) {
                    break;
                }
            }
        }
         $v->uTag = 0$ ;
    }
    //根据更新规则动态更新  $cha$  值和  $G_m$  图;
    Dynamic_update( $cha, G_m$ );
}
Output: partition  $P = \{P_1, P_2, \dots, P_k\}$ ;
MSDPR 算法中对移动社交网络双层社交图的上层顶点

```

进行聚类的时间复杂度为 $O(mkt)$, 其中 m 为上层顶点数, k 为聚类数, t 为迭代次数, 此处 $k \leq m, t \leq m$; 根据聚类结果对移动社交网络双层社交图的下层进行划分复制的时间复杂度为 $O(fk)$, 其中 f 为用户数, $k \leq f$ 。由于 k 和 t 均可认为是常量, 因此 MSDPR 算法总的时间复杂度可简化为 $O(N)$ 。该算法的空间开销主要用于存储移动社交网络双层社交图, 因而 MSDPR 算法的空间复杂度为 $O(N^2)$ 。本算法能保证在一定的时间和空间限制下获得最终结果。

5 实验结果及分析

5.1 实验内容与设置

本文采用美国斯坦福大学网络分析项目提供的 Brightkite 数据集进行实验。该数据集由两部分组成, 即 Brightkite 用户的朋友关系数据和用户进行签到操作的时间位置数据。Brightkite 用户的朋友关系数据包含用户数 58228 个、好友关系有向边 428156 条; 用户进行签到操作的时间位置数据包含 4491143 条签到数据。本实验将数据集分为两部分, 将 2010 年 1 月 1 日前的数据作为历史数据, 用户数为 48971 个, 签到数据共计 3741979 条; 将 2010 年 1 月 1 日后的数据作为动态数据, 用于验证算法的适应性, 用户数增加至 58227 个, 期间新增的签到数据共计 1005308 条。Brightkite 是一个基于位置的社交网络服务, 它提供注册用户与他们现有的朋友的交互功能, 也提供通过查看周围的人寻找新好友的功能, 用户可以使用移动终端的 Brightkite 应用程序进行“签到”操作。

本文首先对 Brightkite 数据集进行统计分析, 以初步验证本文所提算法的合理性, 然后对 Hash 算法^[6]、SPAR 算法^[8]以及 MSDPR 算法进行对比实验。评价指标主要包括跨片访问次数、负载均衡程度、副本占比、基于位置的查询代价以及算法动态适应性。跨片访问次数采用式(1)进行计算以评价划分结果的连通程度; 负载均衡程度采用式(2)中的基尼系数进行计算以评价每次实验中各分片的负载均衡程度; 副本占比采用公式 $\sum_{y=1}^k uCopy_y / f$ 进行计算以评价每次实验的副本拷贝代价, 该方法在文献[8-9]中被使用; 基于位置的查询代价采用随机抽取位置点查询范围的方法, 计算每个查询需访问的分片数以评价每种划分方案在提供基于位置信息查询服务时所需查询代价的大小; 动态适应性采用二次划分复制时发生的用户迁移数进行度量。

实验的软硬件环境: 采用数据库 MySQL5.0 对数据集进行存储, 在 MyEclipse8.5 中对 Hash 算法、SPAR 算法以及 MSDPR 算法进行仿真; 实验采用了 1 台 PC 机, 其配置为 2.20GHz Inter(R) Core(TM)2 Duo CPU, 2GB 内存, 320GB 硬盘。

本实验采用精确到小数点后一位的经纬度来表示位置信息。首先根据用户 2010 年 1 月 1 日前的签到情况统计出用户的常驻位置, 采用的统计方法选取签到次数最多的位置作为常驻位置; 然后对 2010 年 1 月 1 日后的签到情况按月统计出每月签到次数最多的位置信息。每轮实验设置不同的分片数量: 2, 4, 8, 16, 32, 一共进行 5 轮, 然后对划分复制结果进行统计分析。

5.2 实验结果及其分析

5.2.1 数据集统计分析结果

本节对 Brightkite 数据集进行统计分析,首先分析是否存在用户聚集的位置点,根据每个位置点上的聚集人数将位置点分为 5 类:1~5 人、6~10 人、11~100 人、101~500 人及 500 以上,分别统计每类位置点的数目,分析结果的统计情况如图 2 所示。然后统计出每类位置点包含的用户数占总用户数的比例,如图 3 所示。

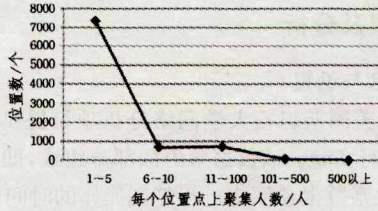


图 2 位置点上聚集人数与位置数的关系

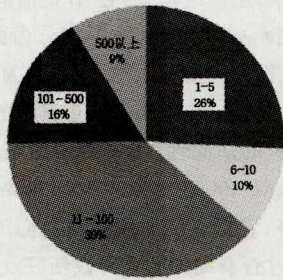


图 3 不同类型的位置点包含的用户数占总用户数的比例

由图 2 可知,聚集人数越多的位置点的数量越少,如聚集了 500 人以上的位置仅有 6 个,而聚集了 1~5 人的位置有 7359 个。由图 3 可知,聚集了 11~100 人的位置点所包含的用户数量是最多的,约占总人数的 39%,而聚集 100 人以上的位置点所包含的用户数量相对较少,尤其是聚集人数在 500 人以上的位置点所包含的人数只占总人数的 9%。因此在移动社交网络环境中,用户的分布呈现“小聚集”的特点,采用位置信息为依据对数据进行划分具有可行性。

5.2.2 跨片访问次数

本节首先采用 Hash, SPAR 和 MSDPR 算法在设置不同分片数的情况下对实验数据进行划分,然后通过计算跨片访问次数来比较每种算法划分结果的连通程度,跨片访问次数越小表示查询时需要进行跨片访问的次数越少。实验结果如图 4 所示。

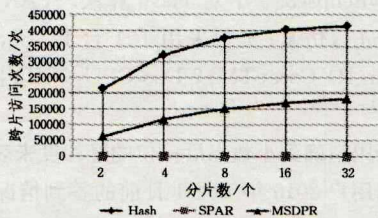


图 4 跨片访问次数与分片数的关系

由图 4 可知,SPAR 算法由于完全保留了好友的关系,因此跨片访问次数一直保持为 0;Hash 算法和 MSDPR 算法的跨片访问次数随着分片数的增加逐渐增加,但 Hash 算法的

跨片访问次数增加速率较快,且一直保持在 MSDPR 算法两倍的水平。这是由于 Hash 算法完全没有考虑社交关系,采用的是随机的划分策略;而 MSDPR 算法以用户位置信息作为划分依据,在移动社交网络环境中位置信息与社交关系之间具有很强的关联性,因此可以取得较好的划分效果。

5.2.3 负载均衡程度

本节采用 Hash, SPAR 和 MSDPR 算法在设置不同分片数的情况下对实验数据进行划分,然后通过计算各分片的基尼系数来比较每种算法划分结果的各分片的负载均衡程度,值越小表示越均衡。实验结果如图 5 所示。

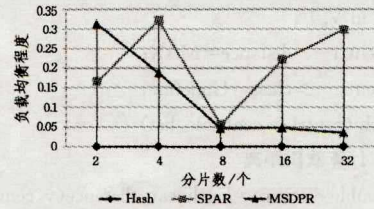


图 5 不同分片数下的负载均衡程度

由图 5 可知,Hash 算法由于采用的是完全随机划分的策略,因此实现了负载的均衡划分。SPAR 算法的负载均衡程度较不稳定,这是由于在划分过程中依次依据好友关系进行划分会存在一些用户顶点一开始需要随机划分的情况,从而造成负载情况不稳定。随着分片数的增加,采用 MSDPR 算法得到的划分结果的负载越来越均匀,这是由于该算法是基于位置信息聚类进行划分的,而移动社交网络中负载在各位置点上较为分散,因此可以得到负载较均衡的划分结果。

5.2.4 副本占比

本节采用 SPAR 和 MSDPR 算法在设置不同分片数的情况下对实验数据进行划分复制,然后通过计算副本占比来比较各算法划分结果的副本拷贝代价。本实验设置 $copy_lucres_0 = 10$,实验结果如图 6 所示。

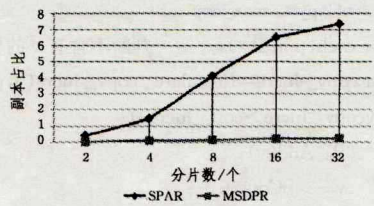


图 6 副本占比与分片数的关系

由图 6 可知,SPAR 算法为了能够完全在片内保留社交关系,其副本拷贝代价较高,当分片数设置为 32 时,SPAR 的副本占比达到 7.3753。而对于 MSDPR 算法,只有当副本建立收益达到一定数值时,才进行副本的建立,因此其副本拷贝代价相对较低。

5.2.5 基于位置的查询代价

在移动社交网络中,基于位置的查询是最具“移动”特色的功能,本节在 Hash, SPAR 和 MSDPR 算法对 Brightkite 数据集的划分复制结果(分片数为 16)上进行实验,以比较 3 种算法对基于位置的查询功能的支持能力。本节随机选取 10 个中心位置、5 种范围进行实验,实验结果如表 1 所列。

表 1 基于位置的查询代价

中心位置	范围/kM	Hash	SPAR	MSDPR
(40, 6, 23)	2	6	2	1
(-33, 9, 151. 2)	2	16	14	1
(59. 3, 18. 1)	5	16	16	1
(39, -84. 5)	5	16	16	4
(41, 28. 8)	10	16	16	1
(58. 4, 26. 7)	10	16	16	1
(35. 3, 137)	100	16	16	1
(28, -16. 7)	100	16	16	1
(26, -97. 5)	200	16	16	5
(41. 1, 28. 9)	200	16	16	1

由表 1 可知,在 Hash 和 SPAR 算法的划分结果上进行基于位置的查询操作的代价是十分高的,几乎每次查询都要访问所有的分片;而 MSDPR 算法在大多数情况下只需要访问 1 个数据片,因此 MSDPR 算法很适合用于基于位置查询操作较为频繁的应用场景。

5. 2. 6 动态适应性

由于移动社交网络是动态变化的,因此移动社交网络数据划分复制算法需要具有较好的动态适应能力。本节对 MSDPR 算法的动态适应性进行实验,分片数设置为 16,将 Brightkite 数据集中 2010 年 1 月 1 日后的数据逐月加入历史数据集(2010 年 1 月 1 日前)中,并统计每月产生的用户迁移数,实验结果如图 7 所示。

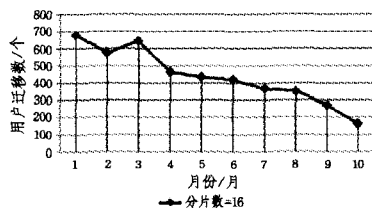


图 7 用户迁移数与月份的关系

由图 7 可知,随着每月用户签到数据的加入,采用 MSDPR 算法进行划分复制,其产生的用户迁移数保持在 700 以下的水平,由于总用户数为 58227,即迁移率保持在 1. 20% 以下,因此 MSDPR 算法在移动社交网络环境中具有较好的动态适应性。

结束语 本文在传统的社交网络图模型的基础上引入了二分图的理论,对移动社交网络用户社交关系进行建模,该模型将移动社交网络中用户的位置信息独立抽象出来,以方便以位置信息为依据进行一系列操作。在此基础上,本文提出的 MSDPR 算法实现了对移动社交网络数据的划分和复制。实验结果表明,与 Hash 和 SPAR 算法相比,采用 MSDPR 算法得到的划分结果具有较少的跨片访问次数,且其负载均衡程度也较好,最重要的是在其之上进行基于位置的查询操作时查询代价远小于其他划分复制算法。在今后的研究工作中,将进一步考察移动社交网络双层社交图模型和该数据划分复制算法的功效,并开发辅助工具以帮助用户选择最佳的划分数目和动态调整阈值;同时可以考虑结合移动社交网络用户的交互情况,采用数据挖掘的方法来确定用户的常驻位置,从而减轻 MSDPR 算法因容易出现的数据频繁迁移等问题而给后台支撑系统带来的压力。

参 考 文 献

- [1] 中国互联网络信息中心. 第 36 次中国互联网络发展状况统计报告[EB/OL]. [2015-07-22]. <http://www.cnnic.cn/hlwfzyj/hl-wxzb/hlwtjbg/201507/P020150723549500667087.pdf>.
- [2] TalkingData 移动数据研究中心. 2015 移动社交应用行业报告[EB/OL]. [2015-07-09]. <http://www.talkingdata.com/index/files/2015-07/1436432039347.pdf>.
- [3] HOFFT. Friendster lost lead because of a failure to scale. [EB/OL]. (2007-11-13) [2015-11-9]. <http://highscalability.com/blog/2007/11/13/friendster-lost-lead-because-of-a-failure-to-scale.html>.
- [4] TAMER M, PATRICK V. Principles of distributed database systems(Third Edition)[M]. New York: Springer, 2011.
- [5] JIN H S, KIM M H. An adaptable vertical partitioning method in distributed systems[J]. Journal of Systems & Software, 2004, 73(3): 551-561.
- [6] LAKSHMAN A, MALIK P. Cassandra: a decentralized structured storage system [J]. ACM SIGOPS Operating Systems Review, 2010, 44(2): 35-40.
- [7] YU G, GU Y, BAO Y B, et al. Large Scale Graph Data Processing on Cloud Computing Environments[J]. Chinese Journal of Computers, 2011, 34(10): 1753-1767. (in Chinese)
于戈, 谷峪, 鲍玉斌, 等. 云计算环境下的大规模图数据处理技术[J]. 计算机学报, 2011, 34(10): 1753-1767.
- [8] PJUOL J M, ERRAMILI V, et al. The little engine(s) that could: scaling online social networks [J]. IEEE/ACM Transactions on Networking (TON), 2012, 20(4): 1162-1175.
- [9] HUANG Y, DENG Q, ZHU Y. Differentiating your friends for scaling online social networks[C] // Proceedings of the 2012 IEEE International Conference on Cluster Computing. IEEE, 2012: 411-419.
- [10] NASIR M, RAHIMIAN F, GIRDZIJ AUSKAS S. Gossip-based partitioning and replication for Online Social Networks[C] // Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASO-NAM). IEEE, 2014: 33-42.
- [11] BROCHELER M, PUGLIESE A, SUBRAHMANIAN V S. Co-si: Cloud oriented subgraph identification in massive social networks[C] // Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining (ASO-NAM). IEEE, 2010: 248-255.
- [12] YU B, PAN J. Location-aware associated data placement for geo-distributed data-intensive applications[C] // 2015 IEEE Conference on Computer Communications (INFOCOM). IEEE, 2015: 603-611.
- [13] KUMAR K, QUAMAR A, DESHPANDE A, et al. SWORD: workload-aware data placement and replica selection for cloud data management systems [J]. The VLDB Journal-The International Journal on Very Large Data Bases, 2014, 23(6): 845-870.
- [14] TURK A, SELVITOPPI O, FERHATOSMANOGLU H, et al. Temporal workload-aware replicated partitioning for social networks [J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(11): 2832-2845.