

带关系属性的空间关键词并行查询处理算法

徐 哲 刘 亮 秦小麟 秦伟萌

(南京航空航天大学计算机科学与技术学院 南京 210016)

摘 要 移动互联网、物联网的快速发展产生了大量带关系属性的空间文本对象数据。面向网页文本数据的搜索引擎仅支持文本关键词查询,无法处理包含地理位置信息、文本信息、关系属性的混合数据。现有面向空间关键字的查询处理技术未将关系属性作为过滤条件,且是基于单机实现的,无法满足查询性能的要求。为解决上述问题,提出了一种新颖的将关系属性、空间和关键字 3 种属性映射成文本数据的 Baseline 算法(Baseline Algorithm of Distributed Keywords and Location-aware with Relational Attributes Query, BADKLRQ),利用分布式倒排文本索引对转换后的文本数据进行并行索引。针对带关系属性、空间和关键字的查询请求,将查询请求转换成映射空间中的多个文本关键字,对转换后的文本数据进行查询,并提出基于 Baseline 算法的改进算法 MGDKLRQ,以改进空间属性转换成文本关键字的算法。实验结果表明,在索引时间和查询时间上,BADKLRQ 算法比现有算法提升了 10%~15%,MGDKLRQ 算法比现有算法提升了 20%~30%。

关键词 空间关键字,关系属性,范围查询,分布式索引

中图法分类号 TP311 **文献标识码** A

Distributed Spatial Keyword Query Processing Algorithm with Relational Attributes

XU Zhe LIU Liang QIN Xiao-lin QIN Wei-meng

(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract The rapid growth of the mobile internet and the internet of things generates a large amount data of spatial text object with relational attributes. Search engines for webpage text data can efficiently store and index textual data, but only support textual keyword queries. However mixed data including geographic location information, textual information, and relational attributes cannot be processed. Existing query-processing techniques for space-oriented keywords do not consider relation attributes as filter conditions. And those techniques are based on stand-alone implementation and cannot meet query performance requirements. In order to solve the above problems, this paper proposed a novel Baseline algorithm named BADKLRQ (Baseline Algorithm of Distributed Keywords and Location-aware with Relational Attributes Query) that maps attributes of relation attributes, space, and keywords into text data. The row text index indexes the converted text data. For query requests with relation attributes, space, and keywords, the query request is also converted into a plurality of text keywords in the mapping space, and the converted text data is queried. And an improved algorithm based on Baseline algorithm MGDKLRQ is proposed to improve the algorithm of converting spatial attributes into text keywords. Experiments show that the BADKLRQ algorithm improves by 10% to 15% and MGDKLRQ algorithm improves by 20% to 30% over the existing algorithm in terms of index time and query time.

Keywords Spatial keywords, Relation attributes, Range query, Distributed index

1 引言

移动电话的普及使得人们能够随时随地地发送和收取带有地理位置的信息,比如在最大的相片分享网站 Flickr 上,每个月新增数百万张带有地理位置信息的相片,用户分享照片时通常会将有定位的信息和自己的感受以文本的格式上传

到网络。而在社交软件微博上,每天新增带有地理位置和文本信息的博文数目高达数十万条。遇到爆炸性新闻或者春节等特殊节日,博文数目更可能高达数百万条。因此,如何高效地对这些包含地理位置、文本信息、关系属性的混合数据进行高效索引和查询是目前急需解决的问题。

图 1 给出了大众点评上的一个饭店的信息。大众点评显

本文受国家自然科学基金(61402225,61373015,41301407),江苏省自然科学基金(BK20140832),中国博士后基金(2013M540447),智能电网保护和运行控制国家重点实验室项目资助。

徐 哲(1993-),男,硕士生,CCF 学生会员,主要研究领域为大数据、空间查询,E-mail: xuzhe@nuaa.edu.cn; 刘 亮(1985-),男,博士,讲师,主要研究领域为传感器网络数据库、时空数据管理等,E-mail: LiangLiu@nuaa.edu.cn(通信作者); 秦小麟(1953-),男,教授,博士生导师,主要研究领域为空间与时空数据库、分布式数据管理与安全等; 秦伟萌(1996-),女,硕士,主要研究领域为空间查询。

示这个饭店的地理位置在南京市江宁区胜太西路19号(中国农业银行向西20m),分类是火锅,营业时间为周一到周日早上10点半到晚上12点整,评分分别是:口味8.8、环境9.0、服务8.8,并且有972条评论。大众点评上这个饭店的信息包含地理位置、文本信息、数值点属性和数值段属性。因为这些数据包含关系属性信息,相比传统空间文本数据更复杂,本文定义这些对象为带关系属性的空间文本对象。



图1 大众点评上某餐厅的所有信息

对于这些对象,普遍使用的空间文本关键词查询(LKQ)^[1-5]不足以让用户找到他们感兴趣的对象。比如,对于饭店信息查询,LKQ只能查询最近的相关饭店而无法查询距离近、有川菜且评论数大于500或者人均消费小于100元的饭店。目前,对于这类问题的相关研究仍处于起步阶段。

随着物联网、移动互联网的高速发展,包含关系属性的地理位置数据和文本信息呈现指数增长趋势,数据量已经达到了TB级甚至是PB级,现有针对空间文本数据的研究^[1-6]均为单机算法,目前尚未解决的问题是如何对包含关系、空间、文本的数据建立统一的分布式索引,从而实现关系、空间、文本3个维度的并行过滤和剪枝,支持面向关系、空间、文本大数据的高效分布式查询。

本文针对现有空间关键词查询算法剪枝效率低、不支持并行的缺点,设计了一种支持关系属性、空间和文本数据的统一索引机制,提出了一种新颖的将关系属性、空间和关键词这3种属性映射成文本数据的Baseline算法,利用分布式倒排文本索引对转换后的文本数据进行并行索引。针对带关系属性、空间和关键词的查询请求,将查询请求转换成映射空间中的多个文本关键词,对转换后的文本数据进行并行查询,提出基于Baseline算法的改进算法MGDKLRQ,改进了空间属性转换成文本关键词的算法。与现有LINQ^[6]算法在索引时间和查询时间上进行了比较,实验结果表明,本文提出的索引算法和查询算法在索引速度和查询速度上性能更优。

2 相关工作

对于给定的空间文本对象集合,带关系属性的空间关键词查询从中筛选出关系属性、空间属性、关键词均满足要求的结果集。现有空间索引方法包括基于R树的空间索引^[2,7-9]、基于网格的空间索引和基于空间填充曲线的空间索引^[10-11]3类,这些方法仅对空间属性进行了索引,支持在空间属性维度进行剪枝,对于带关系属性的空间关键词查询,无法进行文本关键词和关系属性的过滤,查询效率不高。

目前常用的关键词索引结构为倒排文件^[12-15]。倒排文件是关键词查询评估中最流行和高效的数据结构,一般用于文本信息的索引,能够快速定位某一关键词的位置,但是无法实现对空间信息和关系属性的直接索引。

空间关键字存储和处理算法在空间索引基础上构建关键字剪枝数据结构,提高空间关键字查询的效率,具有代表性的算法有IR-tree^[3]、LTKR^[2]和ECMTK^[16]。IR-tree^[3](布尔查询)算法分别指定不同叶子节点应有的位图,从而将文本索引和空间索引联系在一起。该算法的查询性能好,读写次数少,但是只能对文本数据进行布尔查询。LTKR^[2](排序查询)算法将文本的概要集成到空间索引的各个节点中,然后判断对象的文本是否包含查询中的关键字而选择是否进行剪枝,进而进行文本对象的top-k排序查询。ECMTK^[16](排序查询)算法的查询对象包含动态的地理位置与一组固定的关键字,查询结果为动态变化的总体相关度排名前k的对象,但是未考虑到关系属性的查询。文献[6]在IR-tree索引上加入了概要树(Synopses Tree)来索引关系属性的数值点属性,但是不能处理数值段属性。另外,以上算法均为单机算法,不能满足目前TB级甚至PB级数据量的查询性能要求。

3 带关系属性的空间关键词分布式查询算法

为了解决现有空间关键词查询算法剪枝效率低、不支持并行的问题,本文提出了一种新颖的支持关键词、地理位置信息和关系属性的分布式索引BADKLRQ(Baseline Algorithm of Distributed Keywords and Location-aware with Relational Attributes Query index),并基于该索引,提出了一种高效的带有关系属性的空间关键词查询处理算法,然后在BADKLRQ的基础上,优化空间位置信息的索引方式,提出了改进算法MGDKLRQ。该算法不仅可以同时索引这3类数据,且支持分布式存储,索引和查询处理速度显著提高。

3.1 问题描述

带关系属性的空间关键词对象 $O = \{K, G, P\}$ 。 $O, K = \{K_1, K_2, \dots, K_n\}$ 是对象的关键词集合, $O, G = (lon, lat)$ 是对象的经纬度坐标, $O, P = \{P', S\}$ 是关系属性的集合。 $O, P, P' = \{P_1', P_2', \dots, P_n'\}$ 是关系属性的数值点集合, $O, P, S = \{S_1, S_2, \dots, S_m\}$ 是关系属性的数值段属性集合。

带关系属性的空间关键词范围查询 $Q = \{K, R, P\}$ 。 $Q, K = \{QK_1, QK_2, \dots, QK_n\}$ 是查询的关键词集合, $Q, R = \{T, L, B, R\}$ 是查询的地理位置范围集合, $Q, R, T = (lon_t, lat_t)$, $Q, R, L = (lon_l, lat_l)$, $Q, R, B = (lon_b, lat_b)$ 和 $Q, R, R = (lon_r, lat_r)$ 分别代表该地理位置范围的左上角、左下角、右下角和右上角。 $Q, P = \{P', S\}$ 是查询的关系属性集合, $Q, P, P' = \{p_1', p_2', \dots, p_n'\}$ 是查询的关系属性数值点集合, $Q, P, S = \{s_1, s_2, \dots, s_m\}$ 是查询的关系属性数值段属性集合。

对象满足关键词查询的充分必要条件是:

$$O, K \supseteq Q, K \quad (1)$$

对象满足空间范围查询的充分必要条件是:

$$\begin{aligned} & ((O, G, lon < Q, R, T, lon_t) \cap (O, G, lat > Q, R, T, lat_t)) \cap \\ & ((O, G, lon > Q, R, L, lon_l) \cap (O, G, lat > Q, R, L, lat_l)) \cap \\ & ((O, G, lon > Q, R, B, lon_b) \cap (O, G, lat < Q, R, B, lat_b)) \cap \\ & ((O, G, lon < Q, R, R, lon_r) \cap (O, G, lat < Q, R, R, lat_r)) \end{aligned} \quad (2)$$

对象满足关系属性的充分必要条件是:

$$((O, P, P', P_1' = Q, P, P_1') \cap (O, P, P', P_2' = Q, P, P_2'))$$

$$\bigcap \cdots \bigcap (O.P.P', P_n' = Q.P.P_n') \bigcap (O.P.S.S_1 \supseteq Q.P.S.S_1) \bigcap (O.P.S.S_2 \supseteq Q.P.S.S_2) \bigcap \cdots \bigcap (O.P.S.S_m \supseteq Q.P.S.S_m) \quad (3)$$

综上,对象满足带关系属性的空间关键字范围查询的充分必要条件为:

$$(1) \bigcap (2) \bigcap (3)$$

即对象必须包含查询语句中的关键字,必须在查询的空间范围以内,数值点属性必须等于查询的数值点属性以及数值段属性必须在查询的数值段属性范围内。

3.2 Baseline 算法

3.2.1 索引结构设计

本文提出了一种支持关键字、地理位置信息和关系属性的分布式索引 BADKLRQ。不同于大部分现有索引技术采用的混合索引方式(如空间位置信息采用 R 树而文本信息采用倒排索引方式),BADKLRQ 将文本关键字信息、地理位置信息和关系属性信息变换后统一通过倒排文件索引,且支持索引数据的分布式存储和查询。

以某带关系属性的空间关键字对象 $o = \langle k, g, p \rangle$ 为例,说明 BADKLRQ 索引的构建过程。其中, $o.k = \{k_1, k_2, \dots, k_n\}$ 是该对象的关键字集合, $o.g = (lon, lat)$ 是该对象的经纬度坐标, $o.p = \langle p', s \rangle$ 是该对象关系属性的集合, $o.p.p' = \{p_1', p_2', \dots, p_n'\}$ 是该对象关系属性的数值点集合, $o.p.s = \{s_1, s_2, \dots, s_m\}$ 是关系属性的数值段属性集合, $o.p.s.s_i$ 为区间 $[s_{i\min}, s_{i\max}]$ 。

文本关键字信息 $o.k = \{k_1, k_2, \dots, k_n\}$ 经过分词器的分析,将文本信息分析成去重单词 k_1, k_2, \dots, k_n 并标注属于该对象 o , 然后对单词进行倒排索引,生成倒排文件。

地理位置信息 $o.g = (lon, lat)$ 包括对象的经纬度坐标,本文首先将整个空间划分成多个相同的单位矩形,每个对象的真实经纬度 (lon, lat) 都转换到单位矩形中,用该矩形的左下角坐标 (X_R, Y_R) 代表整个矩形中所有点的坐标,以便于倒排文件索引。然后,将地理位置信息标志字段 geo 与单位矩形的经纬度坐标 (X_R, Y_R) 通过下划线连接成如 $geo_X_R_Y_R$ 的字符串,并将其存储在倒排文件中。图 2 是部分对象经纬度坐标转换到单位矩形中的位置示意图,以图中 x 轴坐标范围 $[24, 42]$ 和 y 轴坐标范围 $[60, 78]$ 为例,将每一个点都用其所在矩形的左下角坐标代替,如点 A、点 B、点 C 的地理位置信息都分别转换成“ geo_36_69 ”的字符串存储在倒排文件中。

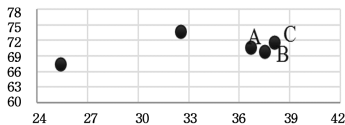


图 2 实验数据中的部分样本点的坐标位置图

关系属性信息 $o.p = \langle p', s \rangle$ 包括数值点属性 $o.p.p' = \{p_1', p_2', \dots, p_n'\}$ 和数值段属性 $o.p.s = \{s_1, s_2, \dots, s_m\}$, 数值点属性将字段名与数值点用下划线连接,以 $f_{1_p_1'}$, $f_{2_p_2'}$, \dots , $f_{n_p_n'}$ 的形式存储在倒排文件中。考虑数值段属性的连续性,将 s_1, s_2, \dots, s_m 转成离散点属性,即将区间 $[s_{i\min}, s_{i\max}]$ 转化为一列点与字段名,并以下划线连接,以 $f_{i_s_{i\min}}$, $f_{i_s_{i_2}}$, $f_{i_s_{i_3}}$, \dots , $f_{i_s_{i\max}}$ 的形式存储在倒排文件中。

图 3 是 BADKLRQ 算法的索引查询示意图,由于文本关键字信息、地理位置信息和关系属性信息互不干扰,该算法可实现 3 类信息的并行索引。

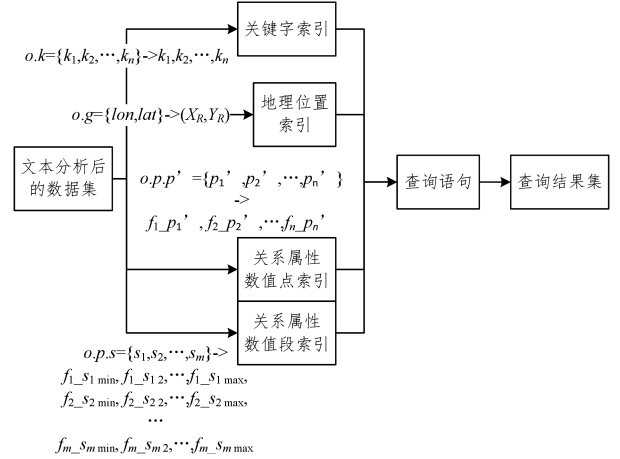


图 3 BADKLRQ 索引查询示意图

3.2.2 Baseline 查询算法

本文基于上述索引提出带关系属性的空间关键字范围查询算法,查询流程如下:

- 1) 用户输入查询语句;
- 2) 将查询语句拆分为文本关键字查询、空间位置信息查询和关系属性查询 3 个部分;
- 3) 将拆分完的 3 个子查询语句转换成 Query 格式语句;
- 4) 将 Query 格式语句构造成 Query 对象树进行查询;
- 5) 合并查询结果集并按照得分高低进行排序。

首先,加载构建完成的索引文件,用户输入格式化查询语句,本文规定用户查询语句的格式为:

```
keywords = Q.k1, Q.k2, ..., Q.kn
&.geopoint = lowlon, highlon, lowlat, highlat
&.attributes = Q.δ1 < ω1, Q.δ1 > ω1' ... Q.δn = ωn
```

其中, $Q.k$ 是该查询的所有关键字信息。lowlon, highlon, lowlat, highlat 是该查询的空间划分范围,将该范围规定为矩形。 $Q.\delta$ 代表关系属性中数值点属性和数值段属性的总称, ω 是关系属性的取值极限值,其中 $Q.\delta_i < \omega_i$ 和 $Q.\delta_i > \omega_i'$ 划分关系数值段属性的查询区间范围, $Q.\delta_k = \omega_k$ 指定了查询的关系数值点。

对于关键字查询,查询语句中的“ $keywords = Q.k_1, Q.k_2, \dots, Q.k_n$ ”通过文本切分器切分成 n 个关键字信息 $Q.k_1, Q.k_2, \dots, Q.k_n$, 然后生成以 AND 连接的关键字域查询语句 $Query: (Q.k_1 \text{ AND } Q.k_2 \text{ AND } \dots \text{ AND } Q.k_n)$, 通过倒排文件进行查询。

对于空间查询,查询语句中的“ $geopoint = lowlon, highlon, lowlat, highlat$ ”通过预处理程序转化为该范围内所有矩形左下角的坐标数组 $[[X_1, Y_1], [X_1, Y_2], \dots, [X_2, Y_1], \dots, [X_n, Y_n]]$, 再将该数组转化成包含地理位置信息的字符串数组 $[geo_X_1_Y_1, geo_X_1_Y_2, \dots, geo_X_2_Y_1, \dots, geo_X_n_Y_n]$, 然后生成以 OR 连接的文本查询语句 $Query: (geo_X_1_Y_1 \text{ OR } geo_X_1_Y_2 \text{ OR } \dots \text{ OR } geo_X_n_Y_n)$. 该语句通过判断索引中的地理位置信息是否满足该字符串数组中的任意元素来判断相应对象的地理位置是否在查询语句的划分范围内。

对于关系属性查询而言,“ $attributes = Q, \delta_1 < \omega_1, Q, \delta_1 > \omega_1', \dots, Q, \delta_n = \omega_n$ ”通过文本切分器切分成 n 个算式,针对数值段查询,生成以 AND 连接的关键字域查询语句 $Query$:“($Q, \delta_1 < \omega_1$ AND $Q, \delta_1 > \omega_1'$)”。针对数值点查询,生成相应关键字域查询语句 $Query$:“($Q, \delta_n = \omega_n$)”,最后将各查询语句以 AND 连接,得到 $Query$:“($Q, \delta_1 < \omega_1$ AND $Q, \delta_1 > \omega_1'$) AND \dots AND $Q, \delta_n = \omega_n$)”,通过倒排文件进行查询。

图 4 给出了用户查询语句转成 Query 格式语句的过程。

当 Query 语句转换成功后,依照 Query 语句、Query 对象树、权重对象树、得分对象树、总得分对象树的顺序进行并行查询,从而得到最终的结果集合和打分情况。

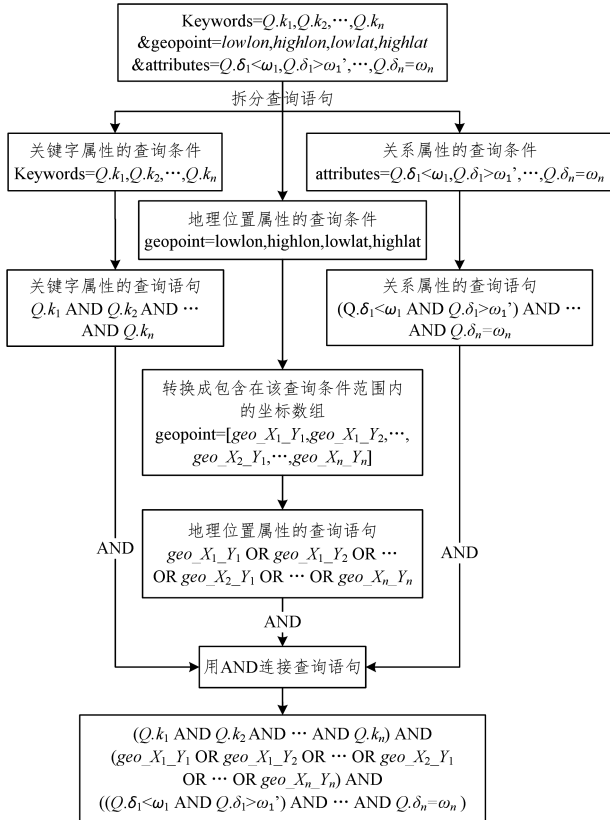


图 4 用户查询语句转成 Query 格式语句的过程

3.3 基于 Baseline 的改进算法 MGDKLRQ

Baseline 算法在空间查询范围较小时性能优异,但随着空间查询范围的扩大,索引生成的范围坐标数组容量随之增加,每次的空间索引查询时间更长,降低了查询效率。为了解决该问题,基于 3.2 节提出的 Baseline 算法,提出基于 Baseline 算法的改进算法 MGDKLRQ,采用结构体存储对象的经纬度信息,直接在经度和纬度上进行精确范围查询,避免了 Baseline 算法在粗粒度查找相应矩形后仍需过滤查询范围外数据的操作,查询效率提高了 15%。

3.3.1 索引结构设计

改进后的 MGDKLRQ 算法也采用单一倒排文件索引,文本关键字信息和关系属性信息的索引方式不变,地理位置信息中经纬度的存储方式由字符串改为结构体形式。图 5 给出了 MGDKLRQ 算法的经纬度坐标转换过程,将包含经纬度坐标的文本对象转换成包含地理位置信息的结构体索引 Geopoint。

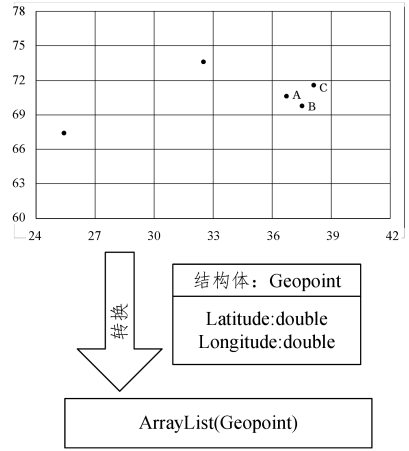


图 5 MGDKLRQ 算法的经纬度坐标转换过程

3.3.2 MGDKLRQ 查询算法

以 3.2 节中给定的用户查询语句为例,详细说明改进 MGDKLRQ 算法的查询过程。

对于关键字查询与关系属性查询,改进 MGDKLRQ 算法的查询过程与 BADKLRQ 算法相同。

对于空间范围查询而言,MGDKLRQ 算法将空间范围查询分成对纬度和经度的范围查询。该算法将空间划分范围坐标 ($lowlon, highlon, lowlat, highlat$) 转换为纬度和经度上的范围条件 ($lowlon \leq Q.R. lon \leq highlon$) \cap ($lowlat \leq Q.R. lat \leq highlat$)。用户查询语句中的“ $geopoint = lowlon, highlon, lowlat, highlat$ ”通过预处理程序被转化为 [$(lowlon, lowlat), (highlon, highlat)$],代表范围的两个坐标,再将这两个坐标转成两个包含不等式的范围查询语句 $Query$:“($Q.R. lon \geq lowlon$ AND $Q.R. lon \leq highlon$)”和 $Query$:“($Q.R. lat \geq lowlat$ AND $Q.R. lat \leq highlat$)”,然后通过 AND 将两个范围查询语句连接,从而得到最终的查询语句 $Query$:“($Q.R. lon \geq lowlon$ AND $Q.R. lon \leq highlon$ AND $Q.R. lat \geq lowlat$ AND $Q.R. lat \leq highlat$)”,以保证该查询语句返回的对象满足其经纬度分别在两个坐标范围内。

图 6 给出了 MGDKLRQ 算法将用户的查询语句转成 Query 格式语句的过程。

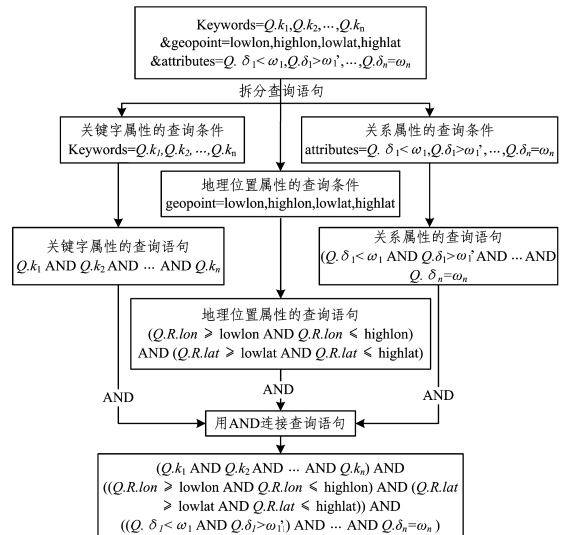


图 6 MGDKLRQ 算法将用户查询语句转成 Query 格式的过程

MGDKLRQ 算法在转化为 Query 格式语句后生成的对

象树与 Baseline 算法的对象树在地理位置信息分支上有所不同,但依然按照从 Query 语句、Query 对象树、权重对象树、得分对象树、总得分对象树的顺序进行并行查询,从而得到最终的结果集合和打分情况。

4 实验结果与分析

4.1 实验环境设置

在服务器集群上进行实验,配置如下:CPU 为 Intel(R) Xeon(R) CPU E5620@ 2.40 GHz,内存为 16 GB,硬盘为 500 GB,操作系统为 CentOS 7。实验所有算法均使用 JAVA 语言编写,JDK 版本为 1.8。开发环境为 IntelliJ IDEA ULTIMATE 2017.1.3。

实验的数据集来自 Yelp 网站上的开源商家信息。数据集以行为单位,每一行表示一个商家的所有信息,包括商户所在位置的经纬度信息、描述该商户的一系列关键字信息、营业时间(数值段信息)以及口味指数、评分大小等数值点属性,总数据量为 20000000 条。

文献[6]提出了单机环境下带有关系属性的空间关键字查询 LINQ 算法。本文在索引时间和查询时间上对 LINQ 和本文提出的 Baseline 算法(BADKLRQ)和改进后的 Baseline 算法(MGDKLRQ)进行比较。

实验中,构造了 100 条对关键字属性的查询,且保证了每条查询均能返回大于 1000 个商户信息;在空间属性上设置了 100 条查询语句,并将空间查询范围从 1% 提升至 13%,每次提升 3%。每个空间查询范围有 25 条查询语句且范围不相同。在关系属性上设置了 100 条查询语句,其中的 34,33,33 条语句分别查询一个、两个、三个关系属性。在服务器数目方面,运行环境从单个服务器依次增加到 3 台服务器,分别用于分析单机环境下本算法与现有算法的优劣,以及在分布式环境下本算法的表现。

4.2 实验结果

1) 服务器数量对算法索引效率的影响

图 7 给出了在服务器个数变化过程中,BADKLRQ, MGDKLRQ 和 LINQ 算法的索引时间。总体而言,BADKLRQ、MGDKLRQ 和 LINQ 算法所用时间在数据量增多时呈线性增长趋势。

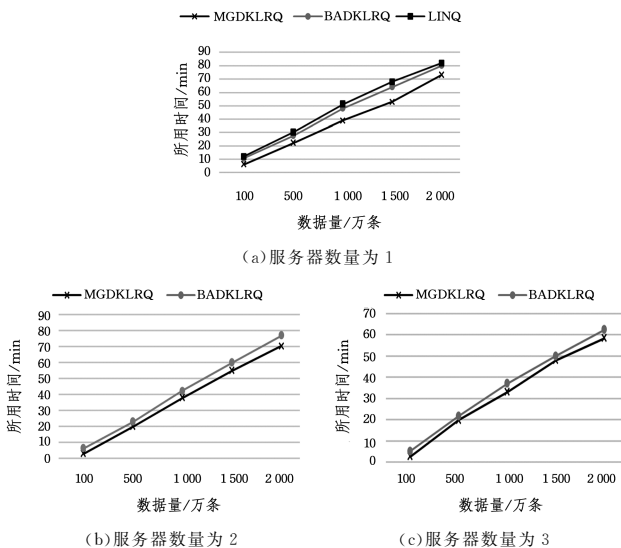


图 7 服务器数量对算法索引效率的影响

由图 7(a)可知,LINQ 算法比 BADKLRQ 算法耗费更多索引时间,而 BADKLRQ 算法比 MGDKLRQ 算法耗费更多索引时间。这是因为 LINQ 采用概要树对数据进行索引,构建过程较复杂。

由于 LINQ 是单机算法,因此无法进行分布式环境的索引时间比较实验。图 7(b)和图 7(c)显示了 BADKLRQ 和 MGDKLRQ 算法在分布式环境中的性能。随着服务器个数增加,两种算法的索引时间也相应缩短。MGDKLRQ 对将空间属性转换成文本进行倒排索引的策略进行了优化,因此其效率优于 BADKLRQ 算法。可见,本文提出的索引算法的优势在于:1)采用单一倒排文件索引能够实现分布式索引,有效降低索引时间;2)可以对数值段属性进行索引,而 LINQ 算法不支持数值段属性的索引。

(2) 查询语句变化下查询效率的比较

图 8 给出了在查询语句变化过程中,BADKLRQ, MGDKLRQ 和 LINQ 算法的查询时间对比。因为 LINQ 不能实现分布式的查询,所以 LINQ 取单机环境的实际数据。实验结果表明,对于不包含空间范围筛选条件的查询语句,Baseline 算法和 MGDKLRQ 算法的查询时间相当。当数据量增大时,BADKLRQ, MGDKLRQ 和 LINQ 3 种算法的查询时间都呈指数增长,BADKLRQ 和 MGDKLRQ 算法均优于 LINQ,且 MGDKLRQ 更优。

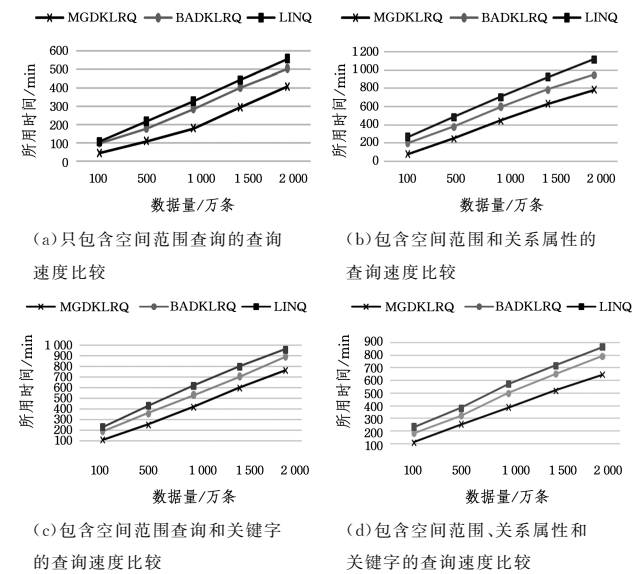


图 8 不同查询语句下查询速度的比较

结束语 针对现有空间关键字查询算法剪枝效率低、不支持并行的问题,设计了一种支持关系属性、空间和文本数据的统一索引机制,提出了一种新颖的将关系属性、空间和关键字这 3 种属性映射成文本数据的算法,利用分布式倒排文本索引对转换后的文本数据进行并行索引。实验结果表明,本文提出的索引算法和查询算法在索引速度和查询速度上性能更优。

参考文献

[1] WU D, CONG G, JENSEN C S. A framework for efficient spatial web object retrieval[M]. Springer-Verlag New York, Inc. 2012.