基于 HBase 的面向语义单元的室内移动对象索引

谢传节1，张得群1

1.中国科学院地理科学与资源研究所 资源与环境信息系统国家重点实验室,北京100101

Semantic Cell Oriented Indoor Moving Objects Index based on HBase

Xie chuanjie1, Zhang dequn1

1. *State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, CAS, Beijing* 100101, *China*

**Abstract**: With the development of indoor positioning technique, index of IMOs (Indoor Moving Objects), as the key technique of Indoor LBS (Location Based Service), has become a hot research spot. But MOI(Moving Objects Index) existed now are almost for outdoor MOs which are in Euclidean Space and can’t supply sematic info, so we need design a new index structure of IMOs. SCoII (Semantic Cell Oriented Indoor Moving objects Index) in this paper is semantic cell oriented and for indexing history data of IMOs, it performs well in answering semantic space range query and semantic trajectory of IMO. It also can be adapt to large scale concurrent updating and querying since it implemented based on HBase. SCoII provides the base of management, analysis of IMO.

**Key words:** indoor; moving objects; index; semantic; HBase

**摘要：**随着室内定位技术的广泛应用，室内移动对象索引技术作为支持室内位置服务的核心技术也成为研究的热点。已有的移动对象索引多是针对室外欧氏空间的移动对象，并且忽略了室内环境的语义信息，不能有效地支持室内移动对象的管理和分析。本文设计了SCoII (Semantic Cell Oriented Indoor moving objects Index)索引结构，对室内移动对象的历史数据进行索引，索引直接面向室内语义单元，能够有效支持室内移动对象的语义轨迹查询、语义空间范围查询、时间范围查询。索引基于HBase实现，能够适应大规模的并发更新与查询，具有良好的规模扩展性。该索引的实现为室内移动对象的管理、分析与室内基于位置的服务提供了基础。

**关键词：**室内；移动对象；索引；语义；HBase

# 1引言

随着室内定位技术的不断发展，获取室内用户的位置数据已经成为可能。这些海量的移动对象数据背后往往蕴含着丰富的与用户行为相关的重要信息，对这些信息进行数据挖据已经成为一个热点问题。

在移动对象的轨迹数据中，相比于地理位置的坐标信息，位置的语义信息具有更加重要的利用价值，可以用来挖掘用户的移动模式、兴趣偏好等用户行为信息[1]。2007年Alvares等[2]提出语义轨迹概念，认为语义轨迹上的点包含除经纬度坐标、时间以外的语义信息，且这些语义信息隐含了在轨迹中可能发生的用户活动信息[3]。HWANG J R等[4]将时间语义信息集成到轨迹中，进行的路网的时空相似度匹配。[5]将语义标签添加到轨迹的位置用，用于计算移动对象的相似度， [6]对大量的交通轨迹数据进行语义相关性挖掘的谱聚类，从而更好地支持交通管理优化， [7]利用GPS的语义轨迹停留点进行位置匹配和数据挖掘等工作。可以看出，语义轨迹对移动对象的分析起到至关重要的作用。

移动对象数据库(Moving Objects Databases, MOD)[8]是对移动对象的位置及相关信息进行表示和管理并提供对移动对象查询的数据库。移动对象索引技术是提高移动对象数据库查询处理性能的关键，相比传统数据库，移动对象数据库查询更加复杂，寻求合适的移动对象索引方法减少搜索空间、加快查询响应速度是移动对象数据库的重要研究部分[9]。但现有的移动对象索引技术多是针对室外环境，并且仅仅关注于对象的位置、范围、时空查询，忽略了环境以及移动对象的语义信息，当处理基于语义约束的查询时，现有索引技术效率低下[10]，不能快速响应甚至不能响应查询。本研究在总结了已有的移动对象索引工作的基础上，设计了基于HBase的面向室内语义单元的移动对象索引，该索引能够有效支持语义空间范围查询、室内移动对象的语义轨迹查询，为用户行为特征提取、室内移动对象的管理与分析等工作建立了基础。

# 2 研究背景

## 2.1移动对象索引

移动对象位置信息的表达是基于时间和空间构成的多维空间，目前建立的移动对象索引主要是针对室外移动对象，在传统空间索引技术如R树，四叉树，网格索引，K-D树等结构的基础上添加时间维形成的。但不同于室外移动对象，室内移动对象除了移动对象所有复杂性、随机性、相关性、不精确性、不确定性[8]外还有一些独特的特点[11] [12]：

（1）、室内空间的独特的拓扑结构使得欧式空间模型和空间网络模型都无法直接使用，室内空间通常表示为一组具有语义的实体，这些实体使得室内空间的轨迹比欧氏空间的轨迹受限制，但又比空间网络模型中的轨迹自由。因此，为了有效的管理室内移动对象，需要使用新的模型去对室内空间建模，以适应轨迹的表达。

（2）、目前室内移动对象数据模型的研究主要着眼于描述移动对象的位置信息[13]，较少考虑语义信息，而语义信息在移动对象分析中起到重要作用，需要提出新的模型，有效支持语义分析。

（3）、不同于室外GPS等定位手段，现有的室内定位主要采用WI-FI定位、RFID定位等手段，精度较差。

另一方面，随着采集数据能力的提高，传感器获得的数据呈爆炸式地增长，传统依托在单机节点的空间数据管理方法已经很难满足大规模的数据存储、更新、检索需求，规模庞大且动态增长的海量信息造成的系统效率下降已经成为很多移动对象数据库的瓶颈问题。以HBase为代表的应大数据时代而生的分布式数据库给移动对象数据库带来了新的解决方案，HBase是根据Google公司Change等人发表的论文“BigTable: A Distributed Storage System for Structured Data”的一个开源实现[14]，其设计旨在能够提供从大规模数据集中随机和实时的高性能读写访问。空间数据既可以被传统的关系型数据库存储，又可以被 NoSQL(Not Only SQL)数据库存储[15]，为了充分利用分布式数据库的特性来提高移动对象数据库的效率，可根据需求构建基于HBase的针对室内移动对象的时空索引结构。

针对以上问题，需要我们建立适应室内空间的移动对象表达、存储模型，设计合理的时空数据库索引。本文所设计的移动对象数据库采用直接存储空间位置的方法，针对离散的历史数据进行索引，索引层直接面向带有语义的室内单元，极大地方便了移动对象的语义分析。

## 2.1研究现状

Jensen等人[16]提出了两种基于R-tree的室内移动轨迹索引结构—RTR-tree和TP2R-tree。RTR-tree将室内移动对象的轨迹表示为若干水平线段，用来支持空间范围查询，TP2R-tree负责把轨迹表示为一个带有时间参数的点，来实现针对特定移动对象的轨迹查询。但由于索引没有考虑符号化表达的室内空间要素的连通性和距离关系，因此无法支持室内空间关系查询和室内空间距离相关查询，并且RTR-tree中移动对象位置更新代价较高。甘早斌等人[17]在RTR-tree轨迹索引基础上增加一棵R-tree进行对象索引，形成了DR-tree (Dual R-tree)这一索引的改进有助于提高室内轨迹查询的性能，但空间开销较大[18]。

Shin等人[19]提出的 ACII (Adaptive Cell-based Index for Indoor moving objects)索引使用双层结构MC(Memory Cell)和 MEMO (Memory) 分别对当前时刻和历史时间区间两种时态的室内移动对象数据进行索引：MC结构将室内单元划分成固定大小的Cell，多个Cell组成楼道、房间等室内空间[19]，在此基础建立空间索引树，并以室内单元为标识对当前时刻所含有的移动对象建立Hash索引，MEMO结构以移动对象为主键，管理移动对象在历史区间内的轨迹数据。该索引支持全时态的移动对象查询，但没有考虑室内单元的距离关系和连通性，也无法支持室内空间关系查询和室内距离查询[17]。ACII可以实现较高的数据更新频率，适用于实时应用的场景，由于ACII是基于图模型的，空间单元大小不固定，因此可解决于不同时间段或者不同空间区域移动对象过于集中数据分布产生倾斜的问题，实验表明，ACII最多使用R-tree等其变种树30%的空间，轨迹查询效率却要高于R-tree[19]。

贲婷婷等人[10, 20]提出了MQII ( Multiple queries indoor index)索引结构，该索引分别建立了Hash表、对象链表、桶链表来存储移动对象历史轨迹信息和当前位置信息。对象链表提供针对移动对象的索引，存放移动对象信息和移动对象的桶链表指针；Hash表存储室内单元信息和该室内单元对应的桶链表地址，每一个单元对应一个通链表，用来存储该室内单元中对象的信息，包括进出时间等。该方法有效支持对象查询、位置查询、范围查询、时间片查询，实验效率高于ACII索引，但其消耗空间较大，针对大数据量的存储与检索有待进一步改进。

以上的索引结构是建立在传统的关系型数据库上的，虽然能够实现是移动对象的管理与分析，但通常要受限于数据量、更新频率、查询频率等，存在不同的瓶颈。另一方面，分布式数据库如BigTable, HBase等虽然能够应对大规模的并发场景，但并不直接支持时空索引和属性索引，不满足移动对象存储与分析的要求，如果没有合适的二级索引，即使基于MapReduce 的并行计算框架，也要扫描全部数据才能够实现数据检索，造成了计算资源、时间的极大浪费，降低了效率[21]。如何基于分布式数据库的特点针对移动对象数据设计合理数据结构与索引结构，许多学者也做了很多的研究工作。

Nishimura等人[22, 23]设计的MD-HBase时空索引结构通过利用 Z-ordering 空间填充曲线将三维时空数据降至一维，使其能够直接利用HBase的行健索引。MD-HBase首先通过KD-tree和Quad-tree建立空间索引，之后再通过Z-ordering空间填充曲线将其降至一维，Z-ordering的编码规则类似于PHT (Prefix Hash Tree)[24],是一种分布式的 Hash编码规则，其特点是如果区域 A包含区域 B,那么A的Hash编码是B的 Hash编码的一个前缀，保证了灵活的范围检索功能。MD-HBase能够实现较高效率的范围查询，但不支持对象查询，虽然MD-HBase也可以在空间索引树的结点上添加语义信息，但进行语义查询时需要先进行空间索引树搜索，再转换为Z-ordering编码，最后进行实际数据检索，造成了效率极大的下降，并且不能有效支持大范围的空间查询[21]。另外该方法只在HBase的Meta-table级别建立了索引，没有针对HRegion内部数据建立索引，因此当其结点较少，HRegion存储量较大时，数据读取效率有限[25]。

MD-HBase在对原始数据排序分割的基础上利用HBase 进行范围扫描实现的索引，没有解决数据写入热点、大范围查询等问题，LiShen等人[21]在HDFS上构建出的PyroDB，利用Moore空间填充曲线对时空数据进行降维，在检索过程汇总、聚集一些可能的小范围查询为一个大范围查询，以此提高扫描速度；该系统还保证了在空间区域划分的时候的实现数据本地化，节省了服务期间通信、传输数据的资源，提高了查询速度。Pyro提出了 Adaptive Aggregation Algorithm(A3) 算法，在进行数据检索前过滤误判的范围，进一步加快了数据检索效率。PyroDB还通过改进HBase集群Split的过程，使得数据能够及时分布到新节点上，解决了数据热点问题。

GeoMesa[26]是一个开源的、分布式的时空数据库引擎，可基于 Accumulo, HBase, Cassandra, Kafka等数据库构建，GeoMesa提供了NoSQL数据库的快速时空数据检索，其角色如同PostGIS对PostgreSQL一样[27]。GeoMesa索引移动对象数据的方法是将二维空间和时间的构成的三维空间均匀分割成时空三维立方体单元(3D Cube)，通过空间填充曲线Z-ordering的值作为HBase行健。GeoMesa通过建立多个索引数据表来支持非时空属性查询，每一个索引表实际上都包含了一个完整的数据备份，查询效率取决于三维立方体单元的大小，粒度越小，查询效率越高，但所耗费空间越大[28]，另外多个索引表中有个数据备份，极大地浪费了存储空间。

Chen等人[25]利用了HBase的内部机制建立了两层索引STEHIX(Spatio-Temporal HBase Index)，首先通过Hilbert空间填充曲线将空间数据的划分成等粒度的单元，根据单元的编码值将数据分布在不同的HRegion上，在HRegion内部对单元空间进行四叉分割细化并建立四叉树索引，对时间进行分段索引，进行时空约束查询时只需要求出时间索引和空间索引的交集部分即可。STEHIX通过这两级索引实现时空范围查询和kNN(k-nearest neighbors)查询，与只设计行健的方式相比，该方法提高了HBase的读取速度，实验证明其效率要高于MD-HBase。

从以上的索引结构不难看出，基于分布式数据库的移动对象索引设计主要思想是通过一定的方法将多维空间降至一维，多采用空间填充曲线的方法或者利用GeoHash进行，查询时一般只能实现范围查询，针对移动对象的轨迹查询、对象查询则较难处理。此外，这些索引方式都不支持移动对象的语义分析，需要根据应用需求改进。

表1 移动对象索引总结

Tab.1 List of Moving Objects Index

|  |  |  |  |
| --- | --- | --- | --- |
| 索引分类 | 索引名称 | 索引结构 | 范围查询 |
| 基于传统数据库的移动对象索引 | RTR-TP2R-tree | R-tree | 室内时空范围查询 |
| DR-tree | R-tree | 室内对象轨迹查询 |
| ACII | 先 R-tree后 Hash | 全时态移动对象查询 |
| MQII | Hash后使用链表指针 | 对象查询、范围查询 |
| 基于分布式数据库的移动对象索引 | MD-HBase | KD-tree/Quad-tree + Z-ordering | 时空范围查询 |
| Pyro | Moore | 时空范围查询 |
| GeoMesa | Z-ordering | 对象查询、时空范围查询、属性查询 |
| STEHIX | Hilbert + 四叉树 + 分段索引 | 范围查询，kNN |

# 3索引设计

与室外空间不同，室内空间多数是由多个相互独立的功能区组成，如商场内的一个商铺、走廊、扶梯等，我们为这些独立的功能区域定义一个唯一的ID，作为最小的语义单元( Semantic Cell)，当用户进入这些语义单元且被记录位置，就可以将该单元的语义信息关联到位置记录上，获得移动对象的语义位置，由语义位置的便可以计算该移动对象的语义轨迹。每一个室内单元可以用定义表示为：

(1)

其中，cid表示该单元的Id值，name表示该单元的语义名称，geom表示该单元的几何信息，用来判断移动对象与其的位置关系—即移动对象是否落在语义单元内，category表示该单元的分类，如在商场里可以为“女装”、“女鞋”等，tags是该店铺拥有的语义标签，如“时尚”、“休闲”等，也可以再附加其他需要的语义信息。

通常情况下一个室内移动对象的位置记录可以由一个四元组表示：

(2)

而一个移动对象的语义位置则是由位置记录关联语义信息，即本文中的语义单元Id构成，可以表示为：

(3)

语义停留点与语义位置不同，语义停留点为移动对象长时间处于同一语义单元，是抽象概念上的一个点，而语义位置是一次定位记录附加语义信息得到的，所以一个语义停留点可能是由多个语义位置记录组成，我们只需要记录该停留点的语义信息、开始时间和结束时间即可。简单用一个四元组表示，cid为其所停留的语义单元id，表示停留的开始时间，表示结束时间：

(4)

然后，一个室内移动对象的语义轨迹就可以表示为语义停留点的一个序列，表示为：

(5)

本文提出的面向室内语义单元的移动对象索引结构共包括Hash Table、语义空间表、语义轨迹表三部分。Hash Table用来存储和索引室内语义单元的信息，语义空间表用来存储该语义单元内包含的移动对象及原始数据记录，语义轨迹链表用来存储某移动对象所经过的室内单元，整体结构如下图所示。

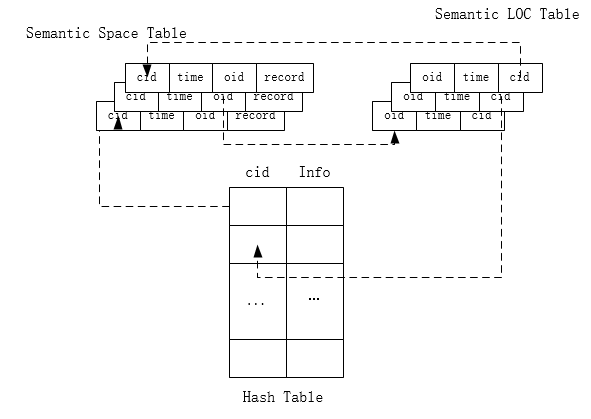


图1 面向室内语义单元的移动对象索引结构

Fig.1 Structure of SCoII

1. Hash Table

Hash Table 用来存储室内单元的语义信息，完成从cid到Cell的映射，可以根据cid快速查询该室内单元语义信息。

1. 语义空间表

语义空间表用来存储移动对象的原始数据记录，存储每个语义单元所包含的所有移动对象，实现基于语义单元的查询。

1. 对象语义位置表

对象语义位置表存储移动对象在每个定位时间戳是对应的语义位置信息，通过对语义位置构建时间正序的序列，便可以获得该对象在特定时间范围内的语义轨迹。

由于HBase不支持行健索引外的属性索引，必须针对时空信息进行降维，通过合理的规则将两者结合起来，共同组成行健，才能利用HBase的行健索引来达到建立移动对象索引的目的。已有的基于HBase的移动对象索引多是针对室外空间，采用空间填充曲线的方法进行，只能支持范围查询，不能实现移动轨迹查询和语义查询，因此需要根据目的设计特殊的行健组成。

根据语义位置的定义可以看出，移动对象的语义位置可以使用其所处的室内语义单元的cid进行关联，cid也代表空间信息，在此之上结合时间属性，就可以做到语义上的空间与时间共同索引，这是本文索引的思想基础。由于同一时间同一Cell内会有多个移动对象，为了有效区别开来，再结合每个对象的ID就可以唯一确定一个不重复的RowKey,作为一个语义位置的行健，并有效利用RowKey的索引，实现快速查询，语义空间表的行健构成如下所示：

(6)

其中，Cell用Cell-Id唯一表示，需要保证全局唯一性时间统一采用UNIX时间戳来表示，精确到秒，为十位整数；Object这里用移动对象设备的MAC地址来唯一标示。

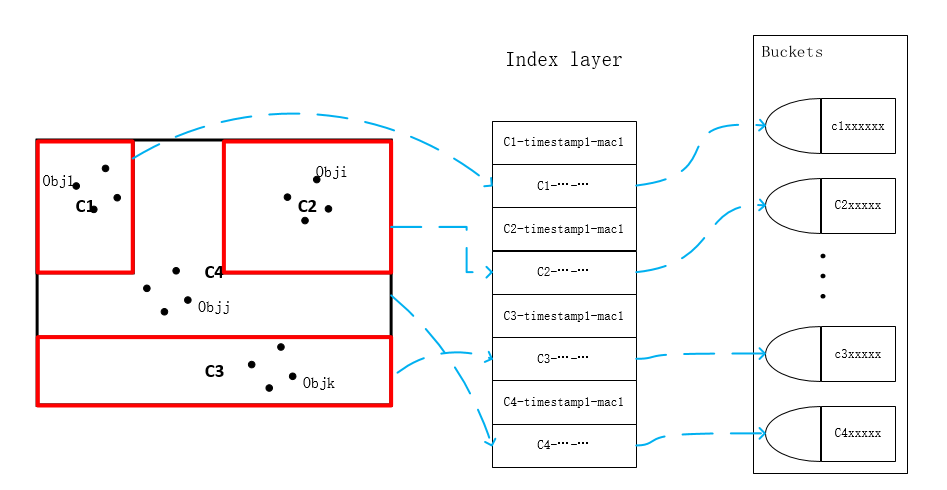


图2 语义空间表结构

Fig.2 Structure of Semantic Space Table

由于HBase不支持除RowKey外的属性索引，对于移动对象的轨迹查询，我们需要另外建立一个索引表，其RowKey组成与原始数据的组成部分一致，但将Object部分与Cell部分对换，这样给定移动对象和时间范围就可以查询出该对象的语义轨迹，如果要查询具体的位置信息，则可以针对查询出来的轨迹RowKey进行变换成语义空间表的RowKey，继而获得详细数据。需要指出的是该索引是稠密索引，并且只存在行键，并不存储实际数据。语义轨迹索引的RowKey设计如下：

(7)

综上所述，面向语义单元的室内移动对象索引的存储结构如图3所示。

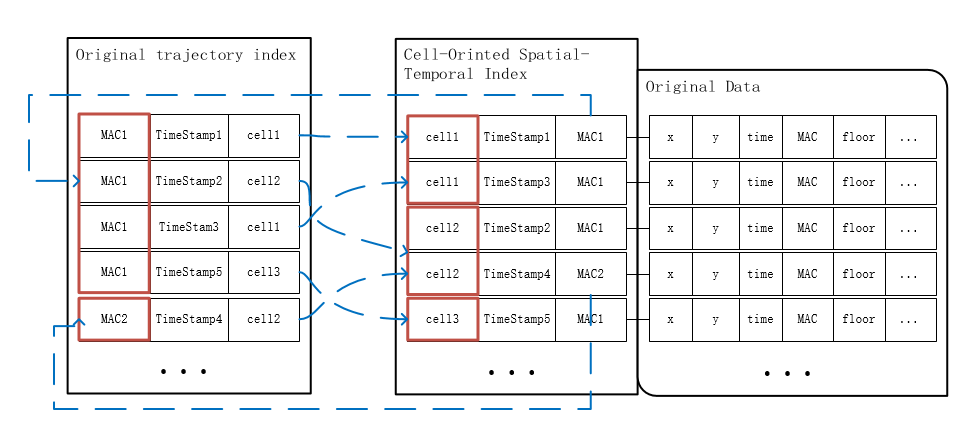


图3 SCoII存储结构

Fig3. Store Structure of SCoII

## 3.2语义单元定位

由于室内空间单元的不规则形，使用传统的R-tree及其变种树建立空间索引会引起较多的MBR(Minimum bounding rectangle)重合，降低了索引的效率，考虑到室内定位精度的影响，本文提出GridCell多对一的映射关系，利用定位点落入Grid判断的高效性来提高数据的更新速度，判断规则如下。

假设与均相交，相交的面积分比为，给定规则如果满足

（8）

以近似认为落入到中的定位点就在内，如果不满足条件的话，需要对 进行四等分，直至条件成立。假设，如图4所示，与均重合，但任何一部分面积都不满足假定规则，所以对进行四叉划分，分别为，，，，计算得到,，由规则可以认为落入这三个区域的移动对象都在内，由于，所以需要对G(12)再次进行四叉划分，直至递归结束。

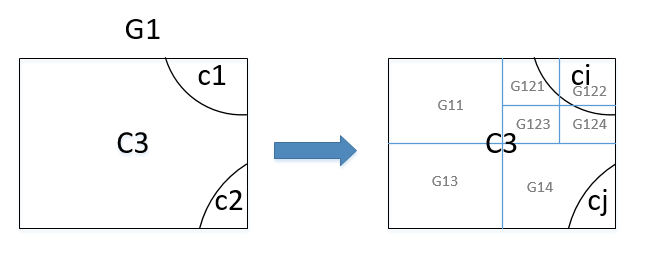


图4 Grid索引建立过程

Fig4. Procedure of Grid Index Building

图5中，室内空间建立Grid索引后，编号为的网格完全落入单元的内，所以当移动对象落入时可以完全确定其肯定位于内部，虽然只有部分位于内，但其重合部分面积的比例，我们可以近似认为落入到中的定位点位于中，对于单元格，则其判断结果是位于外部而不是内部。映射结果如图所示，可以根据定位点的坐标快速判断移动对象所处的Cell。虽然此判断方法具有一定的误判，但可以极大地提高效率，考虑到目前室内定位精度的影响，可以接受这种误差。

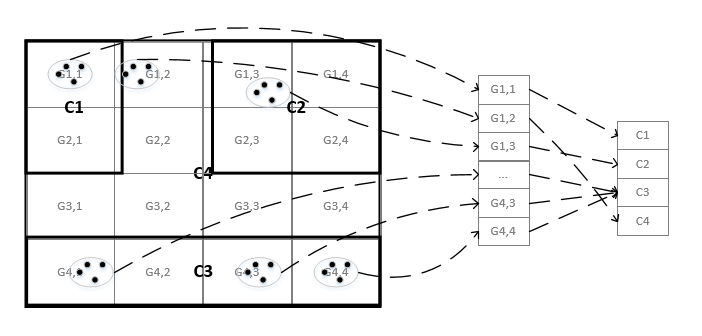


图5 利用Grid索引快速判断移动对象所处的语义单元

Fig5. Utilize Grid Index determine cell which an IMO dropped in.

## 3.3 Grid索引生成算法描述

算法1描述了Grid索引建立及完成GridCell映射的过程，对于给定的室内空间，第一步计算整个楼层的外界矩形，作为最大Grid，将此Grid进入队列，当队列不为空循环以下过程：队首出队，计算所有与相交的室内单元，如果满足条件(8)，则将此Grid与和其重合面积最大的室内单元添加到Map中，所有落入的移动点都视为落入中，否则将四等分得到SubGrids，并将四个子Grid进入队列，队列为空时整个过程结束。

## 3.4数据更新算法

给定数据记录,首先根据得到该定位记录落入的Grid,根的GridCell的映射关系可以得出该记录所处的室内单元，然后可根据公式(6)、公式(7)即可计算出该对象RowKey和对应的轨迹索引MacIdxRowkey作为HBase行健更新数据，过程如算法2所述。

## 3.5语义单元内移动对象统计算法

语义单元内的统计可以得出该单元的总访问量、平均访问量、移动对象密度等信息，可以用来预计客流量、人员控制等。室内单元统计需要输入三个参数，分别为语义单元，开始时间和结束时间，再根据过滤规则进行数据检索，得到检索结果后进行集合操作，最后返回该集合，整个过程如算法3所述。

## 3.5语义轨迹查询算法

因为在语义轨迹索引表中已经包含了语义单元的Id，所以只需要查询该表和Hash表就可以得出移动对象的语义轨迹，如果需要实际的位置轨迹，根据语义轨迹扫描原始记录表就可获得。算法4描述了具体过程：根据移动对象Id和开始结束时间确定扫描范围，仍需要同时满足两个前缀过滤，再根据语义轨迹中的语义单元Id从室内Hash表中检索出实际的语义信息，根据时间顺序排列进行集合操作便可以得到语义轨迹。

# 4系统实现

本文在3台物理计算机上部署了Hadoop 2.6.3、HBase 1.1.2集群环境，其中一个节点作为Master节点,另外两个作为Slave节点。硬件参数为：CPU(intel(R) Xeon(R) E5-2609 @2.40GHz)、RAM(16GB)、Disk(1 TB),操作系统为CentOS 6.4。

实验数据为北京某商场的定位数据，其中每条的记录如表2所示。

表2 定位记录字段组成

Tab.2 Fields of a positioning record

|  |  |
| --- | --- |
| 字段 | 含义 |
| x | X坐标值 |
| y | Y坐标值 |
| floor | 定位点所在的楼层，20040表示F4，10020表示B2 |
| time | 数据采集的时间，精确到秒 |
| mac | 定位对象设备的物理地址 |

## 4.1 Grid效率与STR索引效率对比

首先研究对比了利用映射与直接利用空间索引判断移动对象落入语义单元的效率。空间索引的结构为STR树(Sort Tile Recursive tree)，基于JTS (Java Topology Suite)开源框架实现。对比了不同数据量下两者判断相同定位点数花费的时间，从图中6可以看出，利用映射的时间远小于直接利用空间索引树的时间，效率比约为200:1，极大地提高了寻找语义单元的效率。

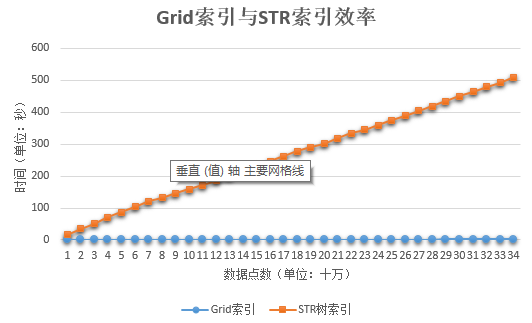


图6 Grid索引与STR判断语义单元效率对比

Fig.6 Efficiency comparison between Grid Index and STR-tree

## 4.2数据更新效率

HBase作为分布式数据库，具有较高的数据更新效率是其特点之一。实验对移动对象数据的更新速度进行了统计，如图7所示，针对实验数据，每万条记录入库的时间为0.35秒左右。实验还选择了PostgreSQL作为关系型数据库的代表，与HBase数据库进行了数据更新速度对比。PostgreSQL是加州大学伯克利计算机系研发的开源数据库，已在可靠性、稳定性、数据一致性等获得了业内极高的声誉，它也在所能管理的大数据量和所允许的大用户量并发访问时间具有完全的高伸缩性，并在9.4版本引入新的类型JSONB，成为混合型的(SQL + NoSQL)数据就，有实验证明其效率已经高于分布式数据库MongoDB[29]。但从图8中可以看出当数据量增加时，HBase的更新速度明显要快于PostgreSQL, 而且数据量越大，效率差越大。说明在海量的室内移动对象管理场景下，选择分布式数据库要优于传统的关系型数据库。

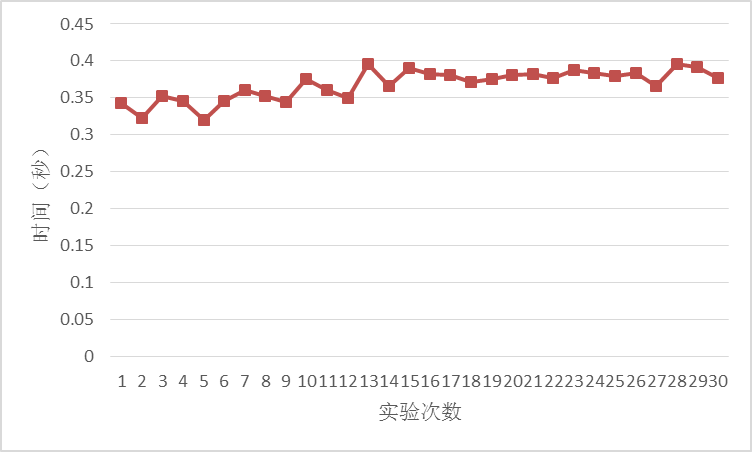


图7 HBase数据更新效率(秒/万)

Fig.7 Time needed of updating every ten thousand points on HBase

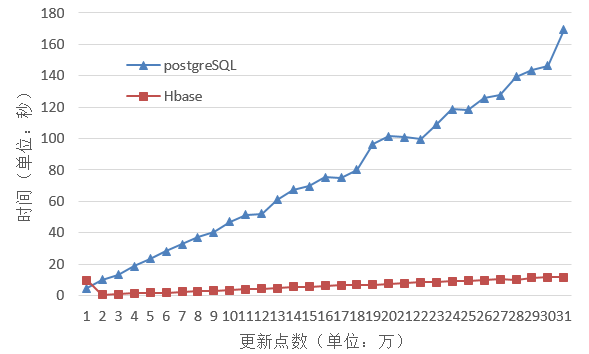


图8 HBase与PostgreSQL更新效率对比

Fig.8 Efficiency comparison between HBase and PostgreSQL

## 4.3基于语义轨迹的用户特征提取

以其中一个MAC为例，从数据库中查询其从2014-04-01 08:00 至2014-04-01 18:00语义轨迹，耗时50毫秒，共得到60个语义节点，经过对语义轨迹进行整理，具有可用信息的结果如表3所示。

表3 语义轨迹查询结果示例

Tab.3 A sample result of Semantic Trajectory query

|  |  |  |  |
| --- | --- | --- | --- |
| Cell名称 | 进入时间 | 离开时间 | 停留时长（单位：秒） |
| Rouge Diamant | 12:21:24 | 12:21:26 | 2 |
| 过道 | 12:21:26 | 12:21:28 | 2 |
| Rouge Diamant | 12:21:28 | 12:21:31 | 3 |
| Bcuthentique | 12:21:31 | 12:22:29 | 58 |
| FIVE PLUS+ | 12:23:14 | 12:24:01 | 47 |
| TRENDIANO | 12:27:51 | 12:50:04 | 1333 |
| Levi's ladies | 12:50:04 | 13:04:53 | 889 |
| Jack jones | 13:27:16 | 13:28:16 | 60 |
| TEENIE&WEENIE | 13:28:58 | 13:30:56 | 118 |
| TEENIE&WEENIE | 13:41:22 | 17:43:14 | 14512 |
| H&M | 17:43:14 | 17:57:07 | 833 |

查询结果中的前三行，用户存在从Rouge Diamant出去再回到Rouge Diamant且时间都比较短的不正常情况，根据实际情况分析，用户在过道中停留2秒有两种可能，一是本身定位误差导致，二是在从的映射引来的误差，也就是说该用户有可能一直停留在Rouge Diamant中，粗略地分析，在过道中的2秒应该归属于Rouge Diamant中。结合了语义信息后，对定位误差的修正也具有一定指导意义。

其他节点的均为整理后的数据，用户停留时间较长（能达到一分钟或超过一分钟）的室内单元有Bcuthentique, FIVE PLUS+，TRENDIANO, TEENIE WEENIE, Levi's ladies, H&M等，其中在Levi's ladies停留约有15分钟，在TRENDIANO停留时间超过半个小时，在TEENIE WEENIE停留超过了4个小时。这些单元的品牌所共有的标签为“女性 时尚 年轻 欧式”，由此可以初步判断，此用户应该是一位年轻时尚的女性顾客，欧式可能是其偏好的风格，并且根据常理猜测，该用户极有可能在停留时间最长的TEENIE WEENIE购买了商品，进一步也可以推断出其消费水平等更多信息。

# 5总结

本文建立了基于HBase的面向室内语义单元的移动对象索引，该索引能够支持语义轨迹查询和语义空间、时间范围查询，实验证明具有较好的更新、查询性能。支持语义轨迹查询是该索引的特点，能够更方便地进行室内用户轨迹分析、行为识别等，为更好的室内位置服务建立了基础。研究最后通过一条语义轨迹的简单分析，完成了粗略的用户画像，在此基础上，可以利用更多的分析手段描述更精细的用户画像，对其行为进行更精准的识别与预测。

虽然该索引是针对历史数据建立的，但其特性更适合索引移动对象的实时，良好的更新性能解决了海量数据更新的瓶颈。针对历史轨迹，更好的方法是建立历史轨迹数据库，将语义轨迹本地化，这样能够保证更快速的检索，为后续研究做准备。考虑到数据的海量性，如何利用Hadoop、Spark等并行计算环境加快数据的分析速度，也成为日后移动对象管理分析的研究方向。

参考文献（References）:

[1] 王倩. 室内移动对象轨迹相似性度量与应用 [D]; 中国科学技术大学, 2015.

[2] ALVARES L O, BOGORNY V, KUIJPERS B. Towards Semantic Trajectory Knowledge Discovery [J]. 2007,

[3] YING J C, LEE W C, WENG T C, et al. Semantic trajectory mining for location prediction; proceedings of the ACM Sigspatial International Symposium on Advances in Geographic Information Systems, Acm-Gis 2011, November 1-4, 2011, Chicago, Il, Usa, Proceedings, F, 2011 [C].

[4] HWANG J R, KANG H Y, LI K J. Spatio-temporal Similarity Analysis Between Trajectories on Road Networks [M]. Springer Berlin Heidelberg, 2005.

[5] YING J C, LU H C, LEE W C, et al. Mining user similarity from semantic trajectories, F, 2010 [C].

[6] 廖律超, 蒋新华, 邹复民, et al. 一种支持轨迹大数据潜在语义相关性挖掘的谱聚类方法 [J]. 电子学报, 2015, 05): 956-64.

[7] 齐凌艳, 陈荣国, 温馨. 基于语义轨迹停留点的位置服务匹配与应用研究 [J]. 地球信息科学学报, 2014, 5): 720-6.

[8] 樊守德. 移动对象轨迹模型、索引结构与查询研究 [D]; 哈尔滨理工大学, 2008.

[9] 方颖. 移动对象数据库中移动对象索引方法研究 [D]; 武汉大学, 2010.

[10] 贲婷婷, 秦小麟, 王丽. 基于语义和访问权限的室内移动对象索引 [J]. 计算机科学, 2015, v.42(03): 178-84.

[11] 杨彬. 室内移动对象的数据管理 [D]; 复旦大学, 2010.

[12] LU H, YANG B, JENSEN C S. Spatio-temporal joins on symbolic indoor tracking data; proceedings of the Data Engineering (ICDE), 2011 IEEE 27th International Conference on, F, 2011 [C]. IEEE.

[13] 汪娜. 面向室内空间的时空数据管理关键技术研究 [D]; 中国科学技术大学, 2014.

[14] 冯晓普. HBase存储的研究与应用 [D]; 北京邮电大学, 2014.

[15] ZHANG N, ZHENG G, CHEN H, et al. Hbasespatial: A scalable spatial data storage based on hbase; proceedings of the Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on, F, 2014 [C]. IEEE.

[16] JENSEN C S, LU H, YANG B. Indexing the trajectories of moving objects in symbolic indoor space [M]. Advances in Spatial and Temporal Databases. Springer. 2009: 208-27 %@ 3642029817.

[17] 甘早斌, 袁永光, 赵贻竹, et al. 基于 DR-tree 的室内移动对象索引研究 [J]. 计算机科学, 2012, 39(10): 177-81.

[18] 金培权, 汪娜, 张晓翔, et al. 面向室内空间的移动对象数据管理 [J]. 计算机学报, 2015, 09): 1777-95.

[19] SHIN S, KIM G, BAE H. Adaptive cell-based index for moving objects in indoor [J]. KSII Transactions on Internet and Information Systems (TIIS), 2012, 6(7): 1815-30 %@ 976-7277.

[20] 贲婷婷, 秦小麟, 许建秋. 支持多种查询的室内移动对象索引 [J]. 计算机研究与发展, 2015, v.52(09): 2002-13.

[21] LI S, HU S, GANTI R, et al. Pyro: a spatial-temporal big-data storage system; proceedings of the 2015 USENIX Annual Technical Conference (USENIX ATC 15), F, 2015 [C].

[22] NISHIMURA S, DAS S, AGRAWAL D, et al. MD-HBase: design and implementation of an elastic data infrastructure for cloud-scale location services [J]. Distributed and Parallel Databases, 2013, 31(2): 289-319.

[23] NISHIMURA S, DAS S, AGRAWAL D, et al. MD-HBase: a scalable multi-dimensional data infrastructure for location aware services; proceedings of the Mobile Data Management (MDM), 2011 12th IEEE International Conference on, F, 2011 [C]. IEEE.

[24] RAMABHADRAN S, HELLERSTEIN J M. Prefix hash tree: An indexing data structure over distributed hash tables; proceedings of the Symposium on Principles of Distributed Computing, F, 2004 [C].

[25] CHEN X, ZHANG C, GE B, et al. Spatio-temporal queries in HBase [J]. 2015, 1929-37.

[26] HUGHES J N, ANNEX A, EICHELBERGER C N, et al. GeoMesa: a distributed architecture for spatio-temporal fusion; proceedings of the SPIE Defense+ Security, F, 2015 [C]. International Society for Optics and Photonics.

[27] GeoMesa.<http://www.geomesa.org/>

[28] WHITMAN R T, PARK M B, AMBROSE S M, et al. Spatial indexing and analytics on Hadoop; proceedings of the ACM Sigspatial International Conference on Advances in Geographic Information Systems, F, 2014 [C]. ACM.

[29] Postgres Outperforms MongoDB and Ushers in New Developer Real.<http://www.enterprisedb.com/postgres-plus-edb-blog/marc-linster/postgres-outperforms-mongodb-and-ushers-new-developer-reality>