基于 HBase 的面向室内语义单元的移动对象索引

**摘要：**随着室内定位技术的广泛应用，室内位置服务快速发展，移动对象索引技术作为支持位置服务的核心技术也成为研究的热点。然而已有的多数移动对象索引都是基于室外环境，或者不支持语义查询，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 (Semantics 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，能够有效支持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。文中使用的从Grid到Cell的映射算法极大地提高了判断定位点所处语义单元的效率，为点与面的非严格的空间关系判断提供了一定的借鉴意义。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，可应用于多种类型的室内空间。

**关键词：**室内 移动对象 索引 语义轨迹 HBase

# 1引言

在用户轨迹数据中，相比于地理位置所表示的坐标信息，位置的语义信息具有更加重要的利用价值，比如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]。

随着室内定位技术的不断发展，获取移动对象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，而如何对这些信息进行数据挖据已经成为一个热点问题[1]。在轨迹数据中，相比于地理位置的坐标信息，人们更关注的往往是这些坐标位置背后所隐含的人们行为及兴趣偏好相关的信息，如位置的语义信息等，这些信息对于挖掘用户的行为模式、兴趣偏好有着巨大的作用，从而有了语义位置的概念。Roth认为语义位置是含有相关关系的位置[2]，2007年Alvares等提出语义轨迹概念，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[3]，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[4]。文献[5]将时间语义信息集成到轨迹模型中，更好地支持了相关应用。文献[6]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，文献[7]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，文献[8]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。

移动对象数据库(Moving Objects Databases, MOD)[9]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[10]。但现有的移动对象索引技术多是基于室外环境，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，现有技术效率低下[11]。然而，语义约束查询在现实生活中大量存在，现有的索引技术不能高效处理这些查询，使得本文的研究更加富有实际意义。本文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，实验证明其能够充分利用分布式数据库的高并发性能，为移动对象的索引设计提供了新的思路，为室内移动对象的管理与分析提供了基础。

# 2 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借鉴空间索引技术如R树，四叉树，网格索引，K-D树等，在其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[9]外还有一些独特的特点[12] [13]：

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[14]，较少地考虑语义信息，尽管不同的移动对象有不同的坐标，但他们可能属于同一个空间单元(Cell)，拥有相同的语义信息，这需要提出新的模型，有效进行语义分析。

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，一方面加大了数据使用的难度，另一方面也可以降低分析精度，加快速度。

针对以上问题，需要我们建立适应室内空间的表达模型、移动对象存储模型，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轨迹查询。

## 2.1研究现状

Jensen等人[15]提出了两种基于的室内移动轨迹索引结构— 和。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，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 中移动对象位置更新代价较高。甘早斌等人[16]在 轨迹索引基础上增加一棵 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，但增加了空间开销[17]。

Shin等人[18]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[18]，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[16]。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[18]。

贲婷婷等人[11, 19]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 (Bucket List)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，存放对象信息和对应的桶链表指针；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虽然能够应对大规模的并发场景，但并不直接支持多维数据的索引，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，也要进行全表扫描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[20]。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[21, 22]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的主键索引。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)[23],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能够时间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[20]。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[24]。

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，但没有解决数据写入热点、大范围查询等问题，LiShen等人在 HDFS 上构建出的PyroDB，利用 Moore空间填充曲线将时空数据进行降维处理，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，在进行数据获取前前过滤误判的范围，进一步加快了数据检索效率。 HBase集群Split 的子节点只保留了指向父节点的指针，在负载均衡的时候才发生实质性的的文件移动复制，虽然保证了操作的简单方便，但不能有效利用新添子节点的服务资源，PyroDB通过改进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[20]。

GeoMesa[25]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 Postgres一样[26]。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主键。 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，最后 GeoMesa 还支持栅格数据的存储。GeoMesa的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[27]，另外多个索引表中有个数据备份，极大地浪费了存储空间。

以上除 MD-HBASE 外的各种方法都是通过设计 HBase 的主键方式来支持时空查询的，Chen等人[24]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范围查询和 kNN(k-th Nearest Neighbor)查询，与设计主键的方式相比，该方法提高了 HBase的读取速度，实验证明其效率要高于MD-HBase。

综上所述，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。另外，这些索引方式都不支持移动对象的语义分析。

表1 已有移动对象索引对比

Tab.1 Main data types and sources

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 | 差 | 低 |
| DR-tree | R-tree | 室内对象轨迹查询 | 差 | 低 |
| ACII | 先 R-tree后 Hash | 全时态移动对象查询 | 优 | 高 |
| MQII | Hash后使用链表指针 | 对象查询、范围查询 | 差 | 高 |
| 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 | 高 | 低 |
| Pyro | Moore | 时空范围查询 | 高 | 高 |
| GeoMesa | Z-ordering | 对象查询、时空范围查询、属性查询 | 高 | 高 |
| STEHIX | Hilbert + 四叉树 | 范围查询，kNN | 低 | 高 |
| EDMI | KD-tree + ZPR-tree | 范围查询 | 低 | 高 |

# 3索引设计

一个室内移动对象最基本的位置记录可以由一个四元组表示，定义为：

一个室内的移动对象轨迹是其位置记录的一个序列，表示为：

一个室内移动对象的语义位置可以简单用一个三元组表示，cid为其所停留的单元id，表示停留的开始时间，表示结束时间，在室内表示进入和离开语义单元的时间。

一个室内移动对象的语义轨迹可以表示为语义停留点的一个序列，表示为：

s

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，语义位置的序列便是该移动对象的语义轨迹，利用语义轨迹对用户分类和行为特征挖掘更加方便、直观[28]。

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,、语义单元表、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，语义单元表用来存储该单元内特定时间所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，用来提取语义轨迹。整体结构如下图所示。

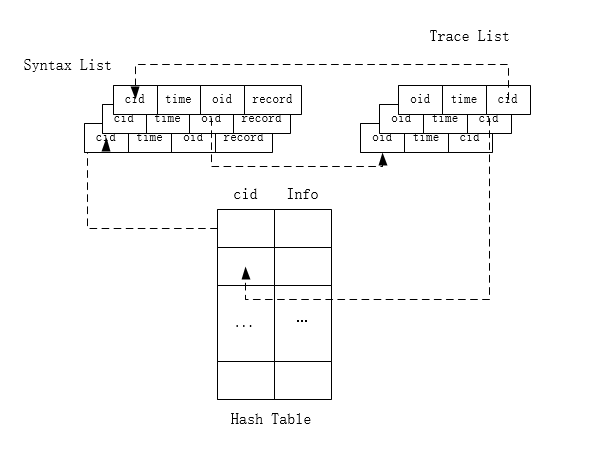


图 面向室内语义单元的移动对象索引结构

Fig. Structure of SCoII

1. Hash Table

Hash Table 用来存储室内单元的语义信息，每一个室内单元可以用定义表示：

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（包含、不包含），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，也可以再附加其他语义信息。Hash Table 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. 语义单元表

语义单元表用来存储移动对象的原始数据记录，以语义单元为准则，存储该单元在特定时间戳时所包含的所用移动对象，实现基于语义单元的查询和统计查询。

1. 语义轨迹表

语义轨迹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戳范围内的语义轨迹，从而进一步进行用户模式和行为识别。

由于HBase不支持主键索引外的属性索引，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，共同组成主键，才能利用HBase的主键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采用空间填充曲线的方法进行，但将时空检索的限制转化为序列值是存在一定的误判区间,一定程度上降低了精确查询的效率，并且不支持语义查询。本文通过提出新的RowKey设计方式，能够实现精确的语义时空查询，方便后续的分析。

如何设计出高效、能够同时包含空间、时间两种信息的RowKey是索引的关键，上述的GridCell的映射可以快速判断移动对象所处的空间单元，是空间信息的一个语义表达，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,并有效利用RowKey的索引，实现快速查询。如下面所示：

(2)

其中，Cell用Cell-Id唯一表示，需要保证全局唯一性，为了保证能够对时间快速查询，需要固定Cell-id的长度，本文的实验中为四位，不足的前缀补零，这样才能利用HBase的PrefixFilter快速定位数据位置；时间统一采用UNIX时间戳来表示，精确到秒，为十位整数；Object这里用移动设备的MAC地址来唯一标示。

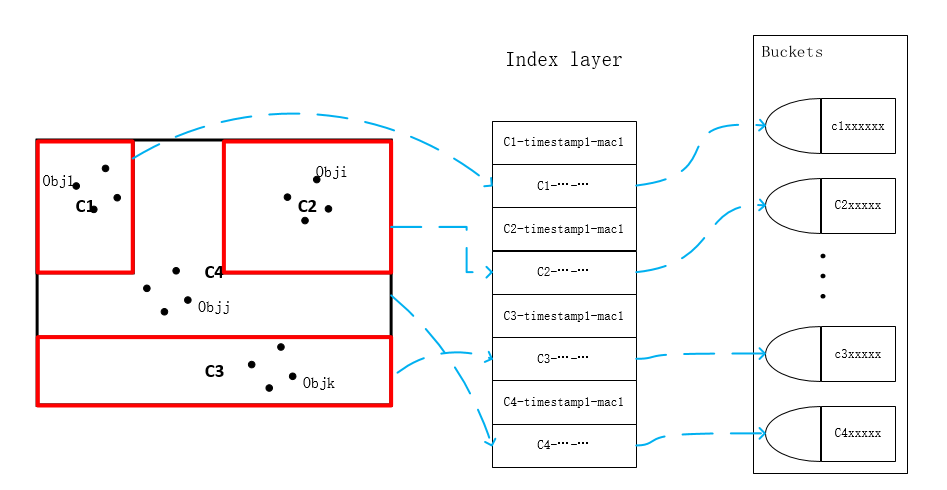


图 语义单元存储示意图

由于HBase的特性，对于移动对象的轨迹索引，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，不同的是将其MAC部分与Cell部分对换，这样给定MAC和Time就可以查询出该对象的语义轨迹，也就是路过的Cell,如果要查询详细的信息，则可以针对查询出来的轨迹RowKey进行变换成原始数据的RowKey,再查询出详细数据.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。轨迹索引的RowKey设计如下：

(3)

语义单元索引与轨迹索引的关系如下图所示

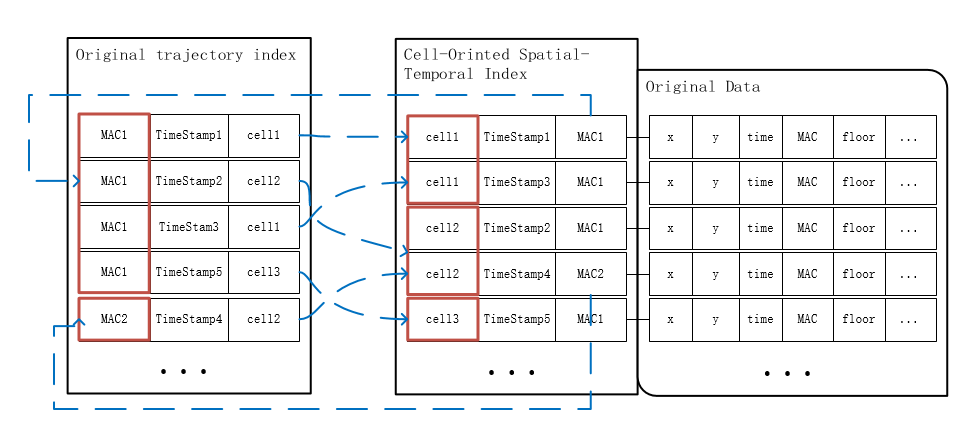


图 语义轨迹与语义单元关系

### 数据更新算法

给定数据记录record(mac,time,x,y,…),首先根据(x,y)得到该定位记录落入的grid,根据算法1得到的gridcell的映射可以得出该记录所处的室内单元，然后可根据公式（1）（2）生成该对象RowKey和对应的轨迹索引MacIdxRowkey，之后进行数据插入操作即可，过程如算法2所示。

### 单元内移动对象统计算法

室内单元统计算法需要输入参数name, beginTimeStamp, endTimeStamp,根据三个参数就得到从开始时间到结束时间的过滤规则和，需要同时满足这两个规则。需要指出的是，HBase可以根据RowKey的索引快速扫描到数据，因此具有较高的查询效率。得到扫描结果后对结果进行集合操作，最后返回该集合。

|  |
| --- |
|  |
|  |

语义轨迹是指该移动对象在一定时间段内经过的语义单元序列，每个语义轨迹的结点表示为(Cell, EntryTime, ExitTime)，而不是指实际的(X,Y,T)序列。因为在语义轨迹索引表中已经包含了语义单元的信息，所以只需要查询该表就可以得出语义轨迹，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。该算法的具体过程为：根据MAC地址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，从得到的MacIdxRowKey中提取出语义单元Id，从室内单元Hash表中检索出对应的室内单元后根据时间顺序排列进行集合操作，就是查询结果。

|  |
| --- |
|  |
|  |

# 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 建立空间索引会引起较多的 MBR 重合，降低了索引的效率，考虑到室内定位精度的影响，建立 grid cell多对一的映射关系，利用点落入 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

（1）

则可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 进行四叉树划分，直至满足条件，如图所示。

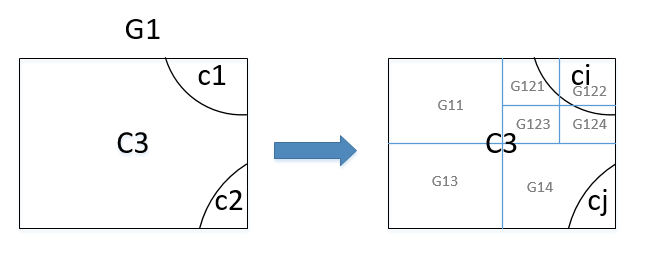


图 Grid生成示意图

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,得到,,根据规则，落入这三个区域的移动对象都认为其在内，由于，所以需要对再次进行四叉划分，直至递归结束。

如图，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。

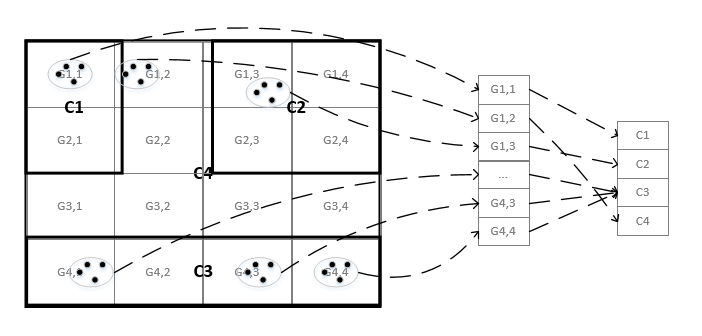


图 利用Grid判断移动对象落入的室内单元示意图

## Grid索引生成算法描述

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，如果满足公式1，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，如果不满足公式1，则将四等分得到，并将四个子Grid进入队列。循环直至队列为空，grid索引的建立和完成到cell的映射关系过程结果。

|  |
| --- |
|  |
|  |

# 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。其中硬件参数为：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一周的定位数据，约有1600万\*7条记录，其中每条的记录包含以下内容

|  |  |
| --- | --- |
| 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 | 记录的时间，精确到秒 |
| Mac | 定位对象的设备物理地址 |

1)Grid效率与STR索引效率对比

首先对比了利用映射与直接空间索引判断移动对象落入语义单元的效率，实现空间索引的结构为STR树，基于JTS (Java Topology Suite)开源框架实现，在对比了不同数据量下两者判断相同定位点数花费的时间，从图中可以看出，利用映射的速度约是直接利用空间索引的200倍，极大地提高了寻找语义单元的效率。

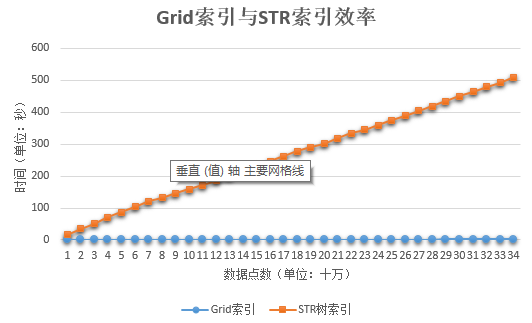
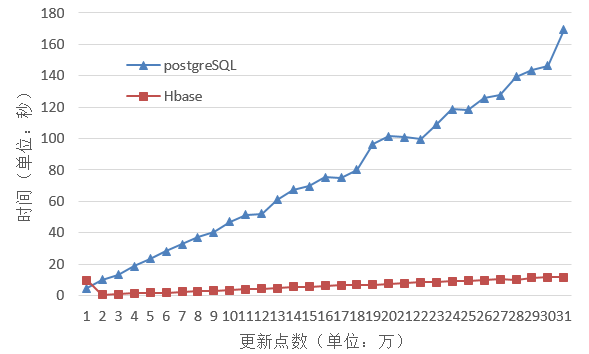


图 Grid与STR判断语义单元效率对比

2)大批量数据写入的效率对比

HBase作为分布式数据库的一个优点就是能够适应大数据量的更新查询等需求，因此我们选择与传统关系型数据库进行对比，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，实验证明其效率已经高于MongoDB,性能稳定。这里我们选择PostgreSQL作为对比对象[29]，测试在相同环境下HBase与PostgreSQL的数据更新效率，作为分布式数据库与传统关系型数据库在大数据引用中的表现代表。



HBase与PostgreSQL更新效率对比

从图中可以看出，当数据量比较少是，PostgreSQL的更新速度要高于HBase,原因是HBase需要与数据节点消耗较多的通信时间，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。

3)基于语义轨迹的用户特征提取

以其中一个MAC为例，从数据库中提取其从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，其所在单元名称及停留时间部分结果如下表所示

|  |  |
| --- | --- |
| Cell名称 | 停留时间（单位：秒） |
| Rouge Diamant | 2 |
| 过道 | 2 |
| Rouge Diamant | 3 |
| Bcuthentique | 58 |
| 过道 | 45 |
| FIVE PLUS | 47 |
| … | … |

根据实际情况分析，其中一个节点在一个过道中停留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，用户不会无理由地进进出出，所以粗略地分析，在过道中的2秒应该归属于Rouge Diamant中。

从其余节点的数据分析，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B+AB, Rouge Diamant, Bcuthentique, FIVE PLUS+，TRENDIANO, TEENIE WEENIE, Levi's ladies, H&M等，其中在TRENDIANO停留时间最长约有一个小时，在Levi's ladies停留约有20分钟，在TEENIE WEENIE停留的也超过了15分钟。经过分析，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MAC所代表的用户应该是一位年轻时尚的年轻女性顾客，虽然也在一些日式品牌店有活动记录，但重复次数更多的欧式应该是其偏好的风格。

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] ROTH J. Flexible positioning for location-based services [J]. Iadis International Journal on Www/internet, 2003, 1(1): 18-32.

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

[4] 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].

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

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 杨彬. 室内移动对象的数据管理 [D]; 复旦大学, 2010.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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].

[21] 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.

[22] 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.

[23] 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].

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

[25] 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.

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

[27] 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.

[28] 张心悦, 王光霞, 吴月, et al. 室内用户语义位置模式挖掘研究——以商场为例 [J]. 测绘与空间地理信息, 2016, v.39;No.202(02): 12-6.

[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>