

空间数据仓库模型和空间 Cube 计算方法^{*})

Spatial Data Warehouse Model and Spatial Data Cube Computation Methods

童云海 谢昆青 唐世渭

(北京大学信息科学中心 视觉和听觉信息处理国家重点实验室 北京 100871)

Abstract A huge amount of spatial data has been stored in corporate computing environments. Spatial data warehouse is the result of importing the data warehouse and online analysis processing (OLAP) technology into the spatial information area. Spatial cube construction and computation are the core problem of the spatial data warehouse and spatial OLAP. In this paper, we introduce the concepts and attributes of the spatial data warehouse model and spatial data cube, and provide three new approximate calculation methods for spatial data cube computation methods according to the spatial aggregate characteristics. Experiments show that these methods can effectively reduce the amount of spatial data and improve the efficiency of spatial data cube construction.

Keywords Spatial data warehouse, Spatial data cube, Spatial aggregate

1 引言

据有关资料统计,目前人类拥有的各种信息 90%以上与空间位置相关。随着各种计算机化的数据收集工具的广泛应用,目前人们积累了大量的地理空间数据。例如,各种数字化的专题地图:岩石类型分布地图、降水量地图、人口分布地图、行政区划地图等。同时,随着各类地理信息系统(GIS)软件的广泛应用,这些空间数据大多得到了有效的管理。

但是,随着空间数据量不断增加、业务范围不断扩大,人们对信息的范围、信息的质量以及信息处理方法提出了更高的要求。特别是对于管理决策人员来说,他们所要求的信息是总揽全局的。目前的各种 GIS 软件难以满足这样的需求。一方面,这是由于现有的 GIS 软件产品尚未形成一套成熟的领域标准,各种 GIS 应用软件的数据结构及其操作彼此不一致,导致同类型的空间数据在不同 GIS 软件中的表示不一致。在这种情况下,高层决策人员很难获得支持分析所需全局的、一致的数据。另一方面,目前的各个 GIS 系统往往只适用于某些特定的业务领域,数据范围比较狭窄,信息量相对来说比较少。但是,政府部门或企业领导的决策通常需要在多个系统的数据的基础上,从时间、地域等不同层次进行横向、纵向的综合对比分析,才能进行有效的决策。

将数据仓库和联机分析处理技术应用到空间数据领域是解决空间信息共享和支持分析决策的一种很好方法,近年来,受到地学界和数据库界的高度重视。目前,作为决策支持系统关键技术的数据仓库和联机分析处理技术得到了很大发展,已经比较成熟,并有了广泛的应用^[1,2]。通过在空间数据领域引入数据仓库和联机分析处理技术,可以有效集成各类不同的空间数据源,建设空间数据仓库和空间数据的联机分析处理环境,从而有效地支持空间决策。

与数据仓库的概念类似,空间数据仓库(Spatial Data Warehouse)是集成的、面向主题的、相对稳定的、反映时间变化和地理空间变化的空间数据存储,为管理人员对空间数据的分析和决策服务^[4]。

空间数据仓库是 GIS、空间数据库与数据仓库技术相结合的产物,它在普通数据仓库基础上,引入空间数据,根据主题从不同的空间数据源中截取从瞬态到区段直至全球系统的不同规模时空尺度上的信息,从而为地学研究以及有关资源环境政策的制定等空间决策支持过程提供最好的信息服务。空间数据仓库的构造与普通数据仓库相比,难度更大。它所面临的挑战有:

- 空间数据的集成。空间数据仓库所要集成的数据分布于各个企业或各个政府部门,这些数据源通常具有不同的数据格式;此外,空间数据的表示、存储和访问方式目前还没有统一的标准,空间数据的语义也往往比较复杂。这就给空间数据的集成和共享带来了极大的困难。

- 空间 Cube 的构造和维护。空间 Cube 是空间数据 OLAP 的主要机制。OLAP 通常要求快速和灵活。但是,由于空间数据仓库中包含的空间数据的数据量通常比较庞大,而且各种空间操作比较复杂,因此,如何有效地构造和维护空间 Cube 是空间数据仓库面临的又一难题。

2 空间数据仓库模型

空间数据面向主题的集成的一个关键步骤是空间数据仓库的建模。空间数据仓库的逻辑模型与普通数据仓库模型一样,通常采用类似的多维模型。与普通数据仓库模型相比,空间数据仓库要管理复杂的空间数据类型,其维和度量中不仅可以包含简单数据类型的数据,同时也可以包含空间对象。由于空间对象占用的存储空间较大,其操作也比较复杂,这就需要扩充维和度量的定义,以便更有效地处理这样的维和度量。

2.1 空间数据仓库中的维

在空间数据仓库中,根据维在各层次的取值中是否包含空间对象,可以将维分为三种:

- 1) 非空间维 非空间维是仅包含非空间数据的维。例如时间维,它在每个层次上(日期、月份、年等)的取值都只包含非空间的数据。

- 2) 空间-非空间维 空间-非空间维在低层次上的取值为

^{*}) 本文研究得到国家重点基础研究发展规划(973)资助,编号 G1999032705。童云海 博士,研究方向为数据仓库、空间数据库与信息系统。谢昆青 副教授,研究方向为空间信息系统。唐世渭 教授,博士生导师,研究方向为数据库与信息系统。

空间数据,而通过概括,在某个层次以上的取值都是非空间数据的维。

3)空间-空间维 空间-空间维是在所有层次上的取值都是空间数据的维。

显然,一个空间属性(例如,行政省)能够以多种方式概括到较高层次的概念,并且这样概括出的概念既可以是空间的(例如国家),也可以是非空间的(例如字符串“东北地区”)。

2.2 空间数据仓库中的度量

与维类似,空间数据仓库中的度量包括数值度量和空间度量。

1)数值度量 数值度量是仅包含数值数据的度量。例如,在空间数据仓库中,某个地区(如:县)的月财政收入可以作为一个度量。

2)空间度量 空间度量是包含空间数据的度量。例如,在空间数据仓库中,具有相同降水量范围的行政县可以作为一个度量,它由一组行政县空间对象构成。

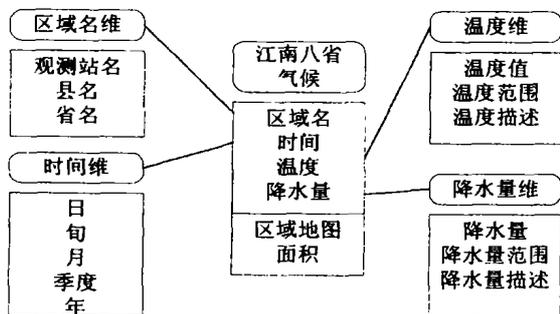
空间数据仓库在物理存储上有 MOLAP 和 ROLAP 两种选择,这两种实现方法的优缺点与数据仓库的情况类似。由于空间数据的存储量大,因此通常采用 ROLAP 方式。

ROLAP 常用模式有星型模式和雪花模式。一个星型模式包括一个大型的核心表和一组在逻辑上围绕这个核心表的维表。核心表包含汇总了的业务事实(即度量)和与之相关的多个维码,称为事实表;而维表就象星星的角一样,分布在事实表的外围,描述了业务的各个维。多维分析是面向主题的,事实表描述了主题的数据,而维表是从不同的角度描述了对主题的分析尺度。

雪花模式是星型模式的一种扩展。它对部分维表进行了规范化,将其进一步分解成更多的表。由于雪花模式对维进行了一定程度的规范化处理,因此,在设计中采用雪花模式能够节省数据存储空间。

空间数据仓库的维和度量都是可以包含空间对象的,这就使得空间数据仓库的星型模式和雪花模式在存储和管理时有其特殊之处。例如,由于空间对象往往较大,多次存储它们会带来非常惊人的存储开销,因此一般用指向空间对象的指针或指针集合来表示空间维和空间度量。

假设已有江南八省各个地市最近 10 年的日降水量、温度等气象数据,以及省、县、观测站的空间数据。现在要进行该地区的气候分析,希望能够观察地图上各区域在时间、温度、降水量等因素影响下的分布情况。为此,可以构建如图 1 的星型模式。



在此星型模式中,区域名、时间、温度和降水量是维,区域地图及其面积是度量。其中,区域地图是空间度量。

3 空间 Cube 的定义和实现

3.1 空间 Cube 的定义及其特点

数据 Cube 是指按多维模型组织的数据,如果将各维正交地放置,就会形成类似于立方体的结构。在多维数据的每一维中选定一个层次以后,所得到的将是一个由度量在相应维层次上所有取值构成的小立方体,它对应于 Cube 中的一个结点,称之为 Cuboid。在选定维层次后,如果再选定维成员,那么得到的将是度量的一个具体取值,它对应于 Cuboid 的一个结点,称之为 Cell。空间 Cube 与非空间 Cube 类似,但是在空间 Cube 中,维和度量都可以包含空间对象。空间维与空间度量采用空间对象的指针或指针集表示。

空间 Cube 中的所有 Cuboid 可以形成一个格(Lattice)。其中,最低层次的 Cuboid(称为基点)代表所有维的最低的抽象层次;最高层次的 Cuboid(称为顶点)则在最高的抽象层次上汇总了所有维。这样,在这个格中上升/下降就对应于 roll-up/drill-down 操作。图 2 所示为一个 3 维 Cube 中的 Cuboid 构成的格。其中,A、B 和 C 代表维名,下标表示概念层次级别(例如 A 表示行政区划维,其中 A₀ 表示县,A₁ 表示省;B 表示温度维;C 表示降水量维)。注意,roll-up 到一个维的最高层次“ALL”等同于去除这个维。

在空间 Cube 中,roll-up、drill-down 和维消减将产生不同的 Cuboid。如果空间 Cube 中只包含空间维而不包含空间度量,那么这些 OLAP 操作可以用类似于非空间 Cube 的方式实现。但是,如果空间 Cube 中包含空间度量,则这些空间度量的计算和存储就需要进行特殊的考虑。这正是空间 Cube 实现的难点。

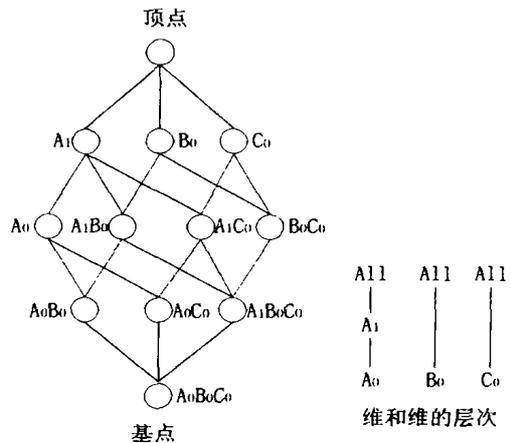


图 2 Cuboid 的格

空间度量与数值度量的差异主要表现在以下几个方面。

首先,数值度量的聚集(例如 SUM、AVG 等)结果是新的数值,而空间度量的聚集是空间对象指针的汇集,当且仅当这些空间指针所指向的空间对象是连通的时候,将合并为一个新的空间对象。否则,它们表示的就是一组分离的区域。

其次,一个数值度量通常只占用 2-8 字节的存储空间,只需要相对较小的计算时间。作为空间对象,一个空间度量可能占用上千或上兆字节的存储空间。因此,在空间 Cube 中应尽量减少物化的空间 Cuboid,以减少存储空间。

第三,空间操作的计算相当复杂,计算一组空间对象的合并或覆盖结果比计算数据度量代价高得多。当联机分析处理时,即时计算空间度量的聚集结果开销非常大。为了加速 O-

LAP 操作,需要尽可能地预先计算一些高层次的聚集视图(Cuboid)并将其作为物化的聚集视图(已计算的 Cuboid)保存起来,而不是即时计算它们。

在实现空间 Cube 时,必须在联机计算空间度量的时间代价和存储预计算空间度量结果所需的存储开销上做出权衡。这是一件困难的事情。在空间 Cube 计算过程中,一方面在不影响总体结果或在误差允许的范围内,需要考虑空间数据的表示方法,同时在计算过程中需要对 Cuboid 有选择地物化。

3.2 基于近似的空间 Cube 的计算方法

在选定需要物化的 Cuboid 后,就需要对这些 Cuboid 中的度量进行预计算。对于非空间度量来说,由于其计算相对简单,并且存放计算结果所需空间较少,通常执行产生该度量的聚集运算并存储计算结果。然而,对于空间度量来说,空间对象的“聚集”计算需要消耗大量时间,且保存计算结果(聚集产生的新的空间对象)需要大量空间。以空间区域合并为例(下文都将以此操作为例,其它空间操作与之类似),计算一个空间度量并保存其结果意味着将参与聚集的所有空间对象根据连通性分组,将每一组连通的空间对象合并为一个新的空间对象,并用该对象指针替代对应空间对象指针组。

因此,计算并存储所有空间度量是不现实的,尤其在考虑到许多由聚集产生的空间对象不会被精确使用或很少被精确使用时,更没有必要计算这些对象所付出的时间代价和为存储它们所付出的空间代价。

在某些情形下,作为即席空间分析并不需要显示或获得精确的数据值,仅需要粗略地估计空间度量或分析某种趋势,因此在大体保证空间对象地理位置及形状的前提下,利用近似计算方法来模拟一个复杂的空间对象,不仅可以大大减少空间对象的存储空间,同时也大大降低了计算复杂性,从而为即席分析决策提供快速响应。下面我们着重介绍基于近似方法的空间 Cube 的构建。

3.2.1 空间近似方法分类和空间近似度 对于空间近似计算的方法,大体上可以分为以下三类:

- 保守近似方法:当且仅当原地理对象的轮廓线包含于近似对象中,即近似对象包含的面积一定大于原对象包含的面积(图 3b);

- 激进近似方法:与保守近似相反,原地理对象的轮廓线一定包含近似对象,即近似对象的面积一定小于原对象包含的面积(图 3c);

- 综合近似方法:通过简化地理对象的轮廓(如:减少顶点个数等),来模拟空间对象。通常情况下,综合近似对象与原地理对象之间是相交关系(图 3d)。

该三类方法中原地理对象与近似对象之间的关系如图 3 所示。



(a)地理对象 (b)保守近似 (c)激进近似 (d)综合近似

图 3 空间近似方法分类

为了能够描述近似值与精确值之间的差距,我们给出近似度的概念,以描述近似质量。

$$G_{Appr} = \frac{A(O) + A(O \setminus Appr(O))}{A(O)} \times 100\%$$

其中: $A(O)$:表示地理对象的精确大小; $Appr(O)$:表示

地理对象的近似值; \setminus :表示两者之间的差距。

当 $G_{Appr} = 100\%$ 时,表明近似地理对象与原地理对象完全一致;

当 $G_{Appr} < 100\%$ 时,表明近似地理对象大于原地理对象;

当 $G_{Appr} > 100\%$ 时,表明近似地理对象小于原地理对象;

目前近似方法的研究主要集中在基于最小外接矩形(MBR, Minimum Bounding Rectangle)的方法。最小外接矩形表示与地理对象最为接近的矩形,它用地理对象的左下角和右上角的坐标参数表示 $(x1, y1, x2, y2)$ 来近似模拟地理对象,我们可以通过简单线性算法求得该两个点的坐标,但 MBR 方法的致命缺点在于精度低,当一个图层中包含多个地理对象时,存在较大的误差。

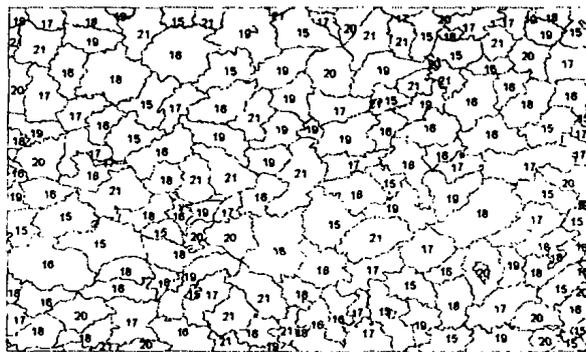


图 4 江南八省县市图及各地的气温状况

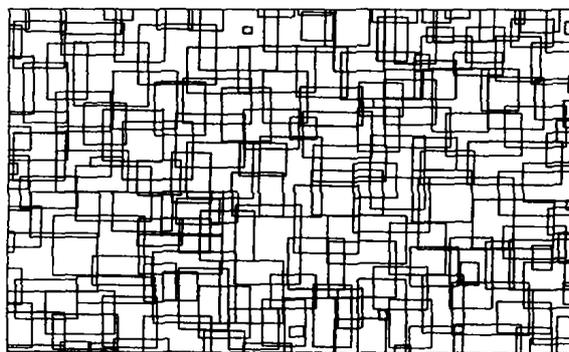


图 5 江南八省县市图的 MBR 表示

我们针对江南八省县图计算全部面积时,得到的 MBR 图形,如图 5 所示。

计算该近似面积为 45.019149,与其精确面积 24.000003 相比,求得其 G_{Appr} :

$$G_{Appr} = 12.4\%$$

3.2.2 基于旋转最小外接矩形的方法 针对最小外接矩形近似度低的缺点,我们抛弃仅对地理对象在 X 轴、Y 轴方向进行拓展的作法,而是用最小外接矩形进行旋转求得旋转最小外接矩形(RMBR),以四个点 $(P1, P2, P3, P4)$ 来表示最小旋转外接矩形可以有效提高近似度。两者的区别可以参见图 6。

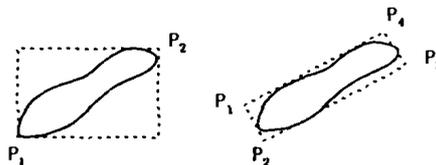


图 6 最小外接矩形和旋转最小外接矩形

```

求最小外接矩形算法如下:
for 图层中的每个 Polygoni
{
  for j=1 to 90 (旋转角度)
  {
    旋转 Polygoni j 度;
    计算旋转后的外接矩形;
    if A(j)<A(min)then min=j;
  }
  记录每个 Polygoni 的旋转最小外接矩形
}
    
```

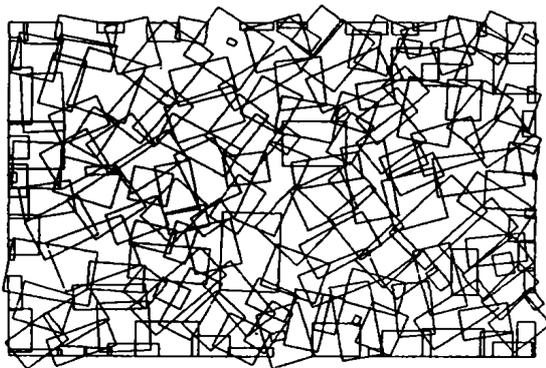


图7 江南八省县市图的 RMBR 表示

针对江南八省县市图,可以得到各个地理对象的旋转最小外接矩形如图7。

在 PIII, 256M 内存, 1 个 CPU 的环境下进行测试, 结果参见表1。

表1 利用 MBR 与 RMBR 近似计算的测试结果

	数据量(K)	面积值	近似度	计算时间(ms)
原图层	1029	24.00000	100%	100.125
RMBR	24.8	39.59618	35.0%	48.625
MBR	24.8	45.01915	12.4%	19.531

3.2.3 基于多级抽点的方法 由于空间信息中一个地理对象(如 Polygon)通常可能由成千或上万个点(Point)对象组成, 针对这种情况, 根据其要求的精度不同, 可以通过滤去地理对象中的若干点, 达到位置和形态的近似。具体的抽点方法包括:

- 随机抽点法: 从一个地理对象中随机抽取若干点。
- 距离去点法: 计算地理对象中任何两个相邻点的距离, 如果两点之间的距离小于某个阈值, 则抽去其中一点。
- 宽带面积去点法: 通过计算相隔两点所形成的面积, 并形成一个一定阈值的矩形范围, 如果中间点位于该矩形范围内, 则去掉该点。

上述每种抽点法都必须满足地理对象的完整性条件, 即抽去点后形成的地理对象仍然是一个完整有效的地理对象, 如多折线对象包含的点数 ≥ 2 , Polygon 包含的点数 ≥ 3 且首尾必须重合等。

下面将介绍随机抽点方法和宽带面积去点方法的相应算法及实验结果。

3.2.3.1 随机抽取点。针对面图层 Polygon 的随机抽取点算法如下:

```

#define NUM N//每 N 点抽取一个点
for 图层中的每个 Polygon i
{
  for Polygon i 中的每个 ring j
  {
    if (ring j 上的点数 >= 2 + 2 * NUM)
      //保证抽取后每个 ring 上的点数 >= 3
    }
  }
}
    
```

• 4 •

```

{
  for ring j 上的每个 point k
  if (point k 不是 ring j 的首点或尾点) and (k % NUM != 1)
    删除 point k;
  }
  else
  不作抽取;
}
    
```

我们针对江南八省县市图分别从 3 个点、5 个点和 10 个点中抽取一个点, 其中 10 个点中抽取一个点得到的结果如图 8 所示。

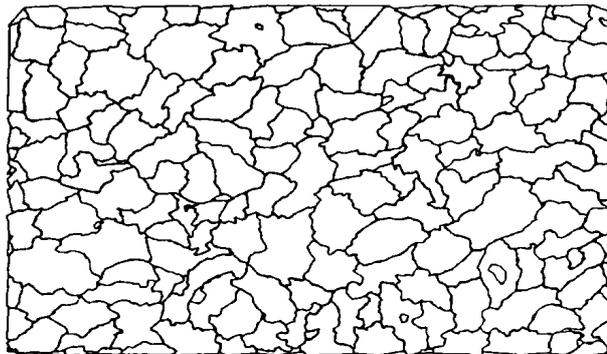


图8 10 个点中抽取 1 个点的测试结果

从测试结果表 2 中可以看出抽点方法可以大大改进近似度。

表2 随机抽点方法测试结果

	数据量(K)	面积值	近似度	计算时间(s)
原图层	1029	24.00000	100%	100.125
抽点: 3-1	353	23.77943	101%	66.321
抽点: 5-1	218	23.17465	104%	60.234
抽点: 10-1	117	22.64901	106%	56.254

3.2.3.2 宽带面积去点法。宽带面积去点法的算法如下:

```

#define AREA // 设定面积常量
Point Pts [3]; // 记录 ring 上连续的 3 个点
for 图层中的每个 Polygon i
{
  for Polygon i 中的每个 ring j
  {
    int pnum = ring j 上的点数;
    Pts [0] = Polygon [i].ring [j].points [pnum-1];
    Pts [1] = Polygon [i].ring [j].points [pnum-2];
    for (int k = pnum-3; k >= 0; k--)
    {
      if 当前 ring j 上的点数 < 4
        // 保证抽取后每个 ring 上的点数 >= 3
        break;
      Pts [2] = Polygon [i].ring [j].points [k];
      if (以 Pts [0] 和 Pts [2] 为左上角和右下角的矩形的面积 < AREA) and (Pts [1] 在该矩形内)
      {
        删除 ring j 上点 k+1;
        Pts [1] = Pts [2];
      }
      else
      {
        Pts [0] = Pts [1];
        Pts [1] = Pts [2];
      }
    }
  }
}
    
```

我们针对江南八省县市图, 分别取面积的阈值为 0.001、0.01 和 0.1 情况, 其中面积阈值为 0.1 得到的结果如图 9 所示。

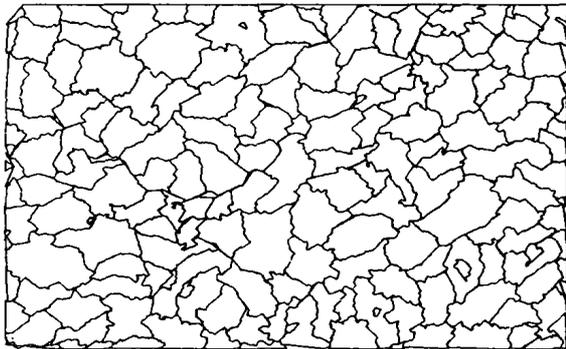


图9 面积阈值为0.1的情况

测试结果比较情况参见表3。

表3 宽带面积去值的测试结果

	数据量(K)	计算值	近似度	计算时间(s)
原图层	1029	24.00000	100%	100.125
宽带面积:0.001	208	23.84722	100%	60.34
宽带面积:0.01	169	22.44723	106%	58.23
宽带面积:0.1	166	21.37943	110%	58.00

3.2.4 基于数据精度转换的方法 在空间图层中,通常用浮点数据形式来表示一个位置,由于浮点运算数占用空间多,计算比较复杂。基于精度转换的方法,将原有的图层数据用整型数据来表示空间中的点。可以达到减少数据存储空间,提高计算速度的目的。

利用该方法对上述数据进行测试,结果如表4所示。

表4 基于精度转换方法的测试结果

	数据量(K)	计算值	近似度	计算时间(s)
原图层	1029	24.00000	100%	100.125
整型表示	256	23.8812	100%	64.38

总结 空间数据仓库和空间 Cube 的计算作为有效支持

空间数据分析的一种手段,目前越来越受到学术界和工业界的高度重视。本文在介绍空间数据仓库和空间 Cube 概念的基础上,针对空间信息数据量大、空间聚集操作复杂的特点,给空间 Cube 的高效计算带来了一些问题,提出了扩充的 MBR、无损精度压缩以及多级抽点方法研究空间 Cube 三种近似计算方法,在减少空间数据的存储空间的同时,提高了空间 Cube 计算的精度和速度。

参考文献

- 1 Inmon W H. Building the Data Warehouse. Second Edition. John Wiley & Sons, March 1996
- 2 Chaudhuri S, Dayal U. An Overview of Data Warehousing and OLAP Technology. ACM SIGMOD Record, 1997, 26(1): 65~74
- 3 Güting R H. An Introduction to Spatial Database Systems. VLDB Journal, 1994, 3(4): 357~399
- 4 Han J, Stefanovic N, Koperski K. Selective Materialization: An Efficient Method for Spatial Data Cube Construction. In: Proc. 1998 Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'98), Melbourne, Australia, April 1998. 144~158
- 5 Clement G, et al. OGD1: Toward Interoperability among Geospatial Databases. SIGMOD Record, 1997, 26(3): 18~23
- 6 http://www.opengis.org, 2000
- 7 唐世渭, 杨冬青, 徐其钧, 杨继国, 谢昆青, 裴健, 陈伟毅. 支持数字地球信息集成与共享的空间数据仓库体系结构. 中国图像图形学报, 1999, 4(A版增刊): 64~68
- 8 裴健, 杨冬青, 唐世渭. 基于数字地图的空间联机分析处理和空间数据挖掘. 中国图像图形学报, 1999, 4(A版增刊): 59~63
- 9 Barclay T, Gray J, Slutz D. Microsoft TerraServer: A Spatial Data Warehouse. [MSR-TR-99-29]. Accepted by ACM SIGMOD 2000
- 10 Han J, Koperski K, Stefanovic N. GeoMiner: A System Prototype for Spatial Data Mining. In: Proc. 1997 ACM-SIGMOD Int'l Conf. on Management of Data (SIGMOD'97), Tucson, Arizona, May 1997, (System prototype demonstration)
- 11 Gray J, et al. Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-tab and Sub-totals. Data Mining and Knowledge Discovery, 1997, 1(1): 29~54
- 12 Harinarayan V, Rajaraman A, Ullman J D. Implementing Data Cubes Efficiently. SIGMOD'96, Montreal, Canada, June 1996. 205~216
- 13 Agarwal S, et al. On the Computation of Multidimensional Aggregates. In: Proc. 1996 Int. Conf. Very Large Data Bases, Bombay, India, Sept. 1996. 506~521

(上接第24页)

程序设计语言的软件机制。其基本思想是旨在建立程序执行者对程序本身而非程序所有者的信任,使宿主系统能藉此判定外来代码可否在本机上安全执行。宿主系统预先制定安全策略,访问者据此赋予自己的代码一个证明,以说明其遵守安全守则;再把该证明提交给宿主系统带的证明确认器;证明被确认为对的,则说明真的符合安全需求,方可将代码交付执行。它能克服传统基于密码学的安全机制的一些缺陷,可适用于语言互操作、操作系统内核、尤其是移动代码安全等领域。本文简要介绍了它的概念、动因及实现,并探寻其在 Mogent 系统安全模块中的应用,分别从保护主机站点与保护移动 Agent 两方面进行了分析与设计。

PCC 有一些自然的优点:主机端工作简单,因为证明由外来访问者做,主机无须关心如何做而只需快速自动地确认;防篡改,一旦代码变动后就要重新证明;先检查后执行,避免产生不良后果,也无运行时刻的开销;检查一次通过可执行多次,有利于工程化设计;自含且兼容,不必借助密码学或第三方,但也不排斥同时与之联合使用。然而,在实现中也存在一些未决的限制与难点,诸如证明的自动生成、公理化、规约逻辑

辑、实用化甚至哲学问题,这些都将是未来研究努力的方向和突破口。

参考文献

- 1 Loureiro S. Mobile Code Protection. Thèse présentée pour obtenir le grade de docteur, de l'Ecole Nationale Supérieure, des Télécommunications, Jan 26, 2001
- 2 Jansen W, Karygiannis T. Mobile Agent Security. In: NIST special publication 800-19
- 3 Necular G C, Lee P. Proof-Carrying Code: [Technical Report CMU-CS-96-165]. School of Computer Science, Carnegie Mellon University, Sep. 1996
- 4 Lee P, Necular G. Research on Proof-Carrying Code for Mobile-Code Security. A position paper in DARPA Workshop on Foundations for Secure Mobile Code, March 1997. 26~28
- 5 Feigenbaum J, Lee P. Trust Management and Proof-Carrying Code in Secure Mobile-Code Applications. A position paper in DARPA Workshop on Foundations for Secure Mobile Code, March 1997. 26~28
- 6 李新, 吕建, 曹春, 冯新宇, 陶先平. 移动 Agent 的安全性研究. 软件学报, 2002, 13(4)