

嵌入式空间索引策略

谢 忠, 凤 鸣, 马常杰

中国地质大学信息工程学院, 湖北武汉 430074

摘要: 嵌入式 GIS 的移动终端设备被有限的资源所束缚, 使得需要使用合适的策略去解决这些问题如有限存储与大数据量之间的矛盾以及适应实时的需求。分别总结了移动终端和 GIS 空间数据的特点, 如移动终端的效率性和空间数据的复杂性等。将嵌入式 GIS 应用划分为 3 类: 电子地图浏览、导航系统和野外数据采集。综合描述已有的索引算法: R 树、四叉树、Buddy 树, 同时提出了基于主题导向的空间索引策略。实践证明, 该索引策略对已有的索引算法扬长避短, 大大提高嵌入式 GIS 数据存取、内存使用和 CPU 性能方面的效率。

关键词: 嵌入式 GIS; 空间索引策略; 空间数据管理。

中图分类号: TP311

文章编号: 1000-2383(2006)05-0653-06

收稿日期: 2006-05-30

Index Strategies for Embedded-GIS Spatial Data Management

XIE Zhong, FENG Ming, MA Chang-jie

Faculty of Information Engineering, China University of Geosciences, Wuhan 430074, China

Abstract: Mobile terminals with embedded-GIS are constrained by limited resources. Proper strategies are required to solve problems such as limited memory conflicting with large quantities of data and the requirement of real-time processing. This paper firstly analyzes the characteristics of mobile terminals and GIS spatial data, such as the efficiency of mobile terminals and the complexity of spatial data. Then classifies the embedded-GIS applications into three types according to its different themes: electronic-map, navigation system and field data collection. Finally it describes the integration of existing index algorithms (packed R-tree, Quad-tree and Buddy-tree), and advances the application theme-oriented index strategy. The practice proves that this strategy takes advantage of these algorithms and overcomes its shortcoming to improve the efficiency of data access, memory utility and CPU performance.

Key words: embedded-GIS; spatial index strategies; spatial data management.

0 引言

随着移动技术的发展(移动电话、数字个人助理等), 越来越多的人开始感受到嵌入式 GIS 技术的好处, 并开始在日常生活中使用嵌入式 GIS。与桌面 GIS 类似, 空间数据管理同样适用于嵌入式 GIS 系统。它负责如何在文件或数据库中的数据的组织, 在程序启动时如何管理内存空间, 以及如何提高在大量数据中的检索效率。但是由于嵌入式 GIS 的终端设备资源的限制, 笔者更需要考虑合适的策略去解

决有限存储与大数据量之间的矛盾以及满足实时的需求。论文将重点解决上述的问题, 同时提出根据应用主题来“定制”的索引策略, 用以提高它们在数据存取、内存使用和 CPU 性能方面的效率。

1 嵌入式系统及开发特点

1.1 嵌入式系统概述

嵌入式系统一般指非 PC 系统, 区别于“通用”计算机被用来执行各种各样的普通任务, 嵌入式系

统执行单独一项定义明确的任务。它也由硬件和软件结合。硬件包括处理器/微处理器、存储器及外设器件和 I/O 端口、图形控制器等。软件部分包括操作系统软件(OS) (要求实时和多任务操作) 和应用程序编程。但是区别于通用计算机的是:通用计算机的硬件通常是预先定义好的,包括 CPU、监视器、键盘、鼠标等,并且不需要定制就能得到。对于嵌入式而言,硬件通常是为了系统的特殊目的而定制的,主要原因就是每个系统的要求的差别相当的大,从而导致系统中满足所需功能软件差别也非常的大。

1.2 嵌入式系统的要求

尽管嵌入式系统的类型有所不同,但是他们都具有一些共同的要求:(1)可靠性。嵌入式系统必须在不需要重启和复位的情况进行工作,而这些重启和复位对于很多的桌面操作系统来说是常事,这就要求系统的硬件和软件都非常可靠。可靠性在任何系统中都是很关键的,但有时因为操作系统和软件的不可预见性,习惯于重新启动桌面系统。然而,作为嵌入式软件开发人员必须将硬件和软件的可靠性当作最重要的考虑因素,因为系统并不准备让人去干预它。

(2)考虑成本和耗电。如果一个嵌入式系统被用于非常特殊的目的,例如用于外层空间探测,那成本当然不成问题。但是如果嵌入式系统用于大众市场,那成本就是要考虑的主要问题了。这种情况下,开发人员将开发 ASIC(application specific integrated circuit,专用集成电路)或 ASM(application specific microprocessor,专用微处理器)来减少硬件组件并因而降低成本(Dreamtech 软件研发组,2002)。

很多嵌入式系统是由电池供电,而不是由主电源供电,在这种情况下,耗电量应该减至最低以避免消耗电池。硬件设计人员必须重视这个问题,例如通过减少硬件组件的数目来降低耗电量。

(3)有效利用存储器。大部分嵌入式系统没有硬盘这类存储设备。可用于嵌入式系统的存储芯片只有保存程序的 ROM 和保存数据的 RAM。由于大部分嵌入式系统都没有次存储设备,因此“闪存”(flash memory)被用于存储包括操作系统在内的程序。这就要求嵌入式应用体积必须尽可能的小。

(4)适当的执行时间。大部分嵌入式系统都要求某项任务必须在指定时间被完成,也就是通常意义上所说的“实时操作系统”。这样的情况下,当超过指

定时间即使得出结果也和运行失败是一样的效果,对于应用是无用甚至是有害的。因此在进行嵌入式开发的时候,运行速度是必须考虑的。

1.3 嵌入式系统开发中的问题

由于上述的嵌入式系统的一系列的特点导致嵌入式开发人员在开发软件过程中需要考虑比在桌面操作系统上更多的问题。最典型的的就是代码的优化。在桌面系统中软件开发人员不必过多地考虑代码的优化,因为处理器功能强大,一般有足够的存储器可用并且在响应时间上几秒钟的差异不会带来显著的区别,但是在嵌入式操作系统上存储器和执行时间是主要的约束条件。

(1)更短、更快的处理过程:那些挑战数据管理性能的嵌入式系统应该具备非常高的数据处理速率,而单一处理过程的持续时间应该十分短暂(单一处理过程指的是对数据进行一次基本操作,包括只读、写入或读写)。

(2)降低功耗:软件代码优化是针对 ARM 嵌入式处理器而言的。代码的优化则可以产生高达 32.3% 的功耗节省。Simunic 等(张天骥等,2004)曾分别做过用各种针对 ARM 处理器的编译器进行的试验。此实验结果发现,风格比较好的代码产生的效果远比用 ARM 编译器优化的效果好。

2 嵌入式 GIS 数据特点以及应用范围

2.1 普通 GIS 数据的特点

GIS 数据库指的是某区域内关于一定地理要素特征的数据集合,主要涉及对图形和属性数据的管理和组织。针对普通数据库 GIS 数据具有以下特点:

(1)GIS 数据库不仅有与一般数据库数据性质相似的地理要素的属性数据,还有大量的空间数据,即描述地理要素空间分布位置的数据,且这 2 种数据之间具有不可分割的联系。

(2)地理系统是一个复杂的巨型系统,用数据来描述资源环境,数据量大,即使是一个极小的区域,数据的应用也相当广,如地理研究、环境保护、土地利用与规划、资源开发、生态环境、市政管理、道路建设等。

上述特点决定了建立 GIS 数据库时,一方面应该遵循和应用通用的数据库的原理和方法,另一方面还必须采取一些特殊的技术和方法来解决其他数据库所没有的管理空间数据的问题。由于 GIS 数据库具有明显的空间特性,所以称它为空间数据库。

2.2 嵌入式 GIS 数据特点

随着嵌入式技术和 GIS 技术的成熟和发展,二者相结合的产物——嵌入式 GIS 在普通生活中的应用也逐步扩大,以锐不可挡的气势迅猛发展。嵌入式 GIS 数据除了有 GIS 数据本身的数据量大、复杂的特点外,还必须要符合嵌入式本身的特点就是存储所占空间小,计算检索简单快捷。这就导致了嵌入式 GIS 数据发展中必须要解决的 2 大矛盾:

(1)大数据量与小存储空间的矛盾。复杂的空间数据相对于传统的关系型数据库的数据来说数据量更大,而且数据分布更不均匀。但是在嵌入式系统上,存储空间是非常宝贵的,这就要求用更有效的结构和方式使得 GIS 数据的存储空间更小,使用时所占内存更小。

(2)复杂的空间检索需求与嵌入式的计算速度的矛盾。对于 GIS 数据库来说,将传统数据库中的范围查询扩展到相交查询, GIS 数据库中查询条件是空间位置相关而不直接是属性数据这也是导致 GIS 数据库比传统数据库复杂得多的主要原因。而在嵌入式系统上更短、更快的处理要求,以及嵌入式本身处理器的处理速度,使得我们必须采用比桌面操作系统更为灵活的机制来满足嵌入式的要求。

以上 2 大矛盾也就是嵌入式 GIS 数据的特点。为解决这 2 大矛盾可以从 2 个方面来考虑:(1)采用合适的数据压缩技术;(2)定制合适的“数据索引”机制。

由于嵌入式本身存储和速度的特点使得索引机制不能像在桌面操作系统上一样大而全,并且在嵌入式 GIS 中对于数据的管理和需求也不像桌面操作系统一样比较全面,通常它只是运用到桌面 GIS 数据库的一部分功能。因此针对不同应用的数据特点就需要定制特定的“数据索引”机制。

2.3 嵌入式 GIS 的主要运用及数据特点

目前嵌入式 GIS 虽然运用的行业较为广泛,但就其数据组织和使用的根本来看可以分为 3 大类:

(1)电子地图浏览。这是嵌入式 GIS 应用的一个较大的方面。这种电子地图浏览的应用数据有以下 2 个特点:(a)数据分布较均匀。我们知道在 GIS 中数据通常都是分层的。因此这里的分布均匀有 2 层意思:一是每一层内部分布均匀指的是数据的位置大致分布均匀;二是层与层分布均匀,即每层的数据量大体相同。(b)数据不需要修改。在这类的应用中,仅仅只是对数据的浏览,涉及到的数据的操作仅仅也就是数据的查询。这里的 GIS 数据对于用户来

说是只读的。所以在建立索引的时候就不需要去考虑数据的插入、删除操作了。

(2)导航系统。这是目前嵌入式 GIS 应用最多的一个方面。这类应用和电子地图的最大区别就在于数据分布不均匀。对于导航系统,它的数据模型也是分层,但是它的数据量是逐层递增的,例如其中的城市数据模型,它的数据量是以 2 倍的速率递增。导航系统的数据与电子地图相同的地方就是它对数据也不进行修改操作。

(3)野外数据采集。这是嵌入式 GIS 应用中唯一可能涉及到对原有数据进行修改的应用。但是这里的修改不象在桌面操作系统上的大批量数据的修改,而是对于小范围的少量数据的修改、增加或者删除。

3 基于 PC 的空间索引及其在嵌入式上的限制

3.1 基于 PC 的基本空间索引

3.1.1 网格文件 网格文件在给定的数据集中以反映数据分布的方式将空间数据进行划分。这种划分保证任何的点查询(检索与被查询点相关信息的查询)最多只需要 2 次磁盘的存取。网格文件使用与坐标轴平行的线将空间划分为矩形区域,通过说明坐标轴上的分割点就可以描述网格文件的划分。如果 X 轴被分成 i 个部分, Y 轴被分成 j 个部分,那么总共有 $i \times j$ 个划分。网格目录是一个 $i \times j$ 的数组,每个划分是一个条目。这些信息存储在一个称为线性比例的数组中,每个坐标轴都有一个线性比例 (Raghu and Johannes, 2004)。

网络文件依赖于网络目录识别出包含所需要的点的数据页。当搜索一个点的时候,首先找到网格目录中相应条目(图 1)。

这种网格结构对于处理静态一致分布式数据(例如卫星影响)很有用 (Shekhar and Sanjay, 2004)。网格文件的搜索时间是很短的。对于精确匹配查询来说,只作一次数据访问。然而,网格文件本质上会造成目录非常松散,因而浪费主存缓冲区和二级存储。并且对于局部匹配和范围查询来说,需要扫描很多只有少量数据块的目录项。

3.1.2 R 树系列 R 树是最早基于对象进行划分的一种空间索引树。R 树是高度平衡树 (Guttman, 1984),它实际上是 B 树在 K 维上的自然扩展。R 树

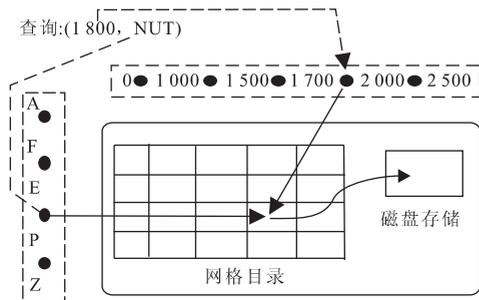


图 1 网格文件索引

Fig. 1 Grid file index

用对象的最小外包矩形(MBR)来表示对象。

R 树搜索的性能取决于 2 个参数:覆盖和交叠。

树的某一层的覆盖是指这一层所有结点的 MBR 所覆盖的全部区域,或者树覆盖的空白区。树的某一层的交叠是指在该层上被多个与结点关联的矩形所覆盖的全部空间区域。交叠使得查找一个对象时必须访问树的多个结点(Shekhar and Sanjay, 2004)。

因此若要得到一个高效的 R 树,覆盖和交叠都应该是最小而且交叠的最小化比覆盖的最小化还要关键。为了解决这个问题,产生其他基于 R 树的变种。例如 packed R 树(Roussopoulos and Leifker, 1985)、R* 树(Beckmann *et al.*, 1990)等。

packed R 树方法,假定数据是相对静止的,并且数据对象在构造树之前就已知。其基本思想是根据空间对象的最小边界矩形的一个角点的 X 坐标或 Y 坐标对空间对象进行排序,然后用这些有序的空间对象逐个压满树的叶结点,自下而上,一次一层,递归生成最终的压缩 R 树。由算法可知,在压缩 R 树中的每一层中,除了最后一个结点可能不满外,其他所有结点都是满的。因此,可以获得几乎 100% 的空间利用率。数据库第一次建立时,可以为将覆盖和交叠最小化而进行有效的组织。此后的插入和删除都遵循 Guttman(1984)的原始 R 树的结构。

3.2 嵌入式对空间索引使用的限制

一般的空间索引的研究都是在资源非常丰富的 PC 机或服务器这样的硬件环境下展开的:总体上资源比较丰富,相对而言,内存有限,外存很大,外存的读取速度与内存相比很慢,为了加快索引,要尽可能减少对外存的读取次数,因此,以前的大部分多维索引结构的研究都是针对所谓的外存索引展开的。而嵌入式的硬件环境与 PC 机有很大不同。与 PC 机或服务器相比,嵌入式与空间索引相关的硬件限制主要是 CPU 速度、存储量(内存/外存)、电量及屏

幕显示量 4 大限制。

3.2.1 存储量 以 iPAQ3700 为例,掌上电脑中用于存储与计算的只有一个 ROM、一个 RAM,其中 ROM 像一个小型的只读硬盘,用于存储全部的操作系统,以及与系统绑定的应用程序。ROM 上的程序被称为就地执行程序,即直接从 ROM 执行,而不是加载到 RAM 再运行,从而节省了 RAM 资源。而 RAM 又被分成 2 个部分:程序内存区与对象内存区,它们的大小相互之间是可调整的。应用程序一般存放在对象内存中,需要时被加载到程序内存执行。这个 RAM 一般来说比较小,通常程序内存与对象内存一共只有 32 M 或 64 M,而且 CPU 对这 2 部分的读取速度是一样的(除了从 RAM 中读取文件有一个由操作系统控制的解压过程)。因此,如果不考虑其他辅助存储设备,而只用掌上电脑内部的硬件来实现全部的系统,只能用主存存储结构。在 PC 机上,比较常用的主存索引的结构有:K-D 树、BSP 树、点四叉树、Cell 树等。

3.2.2 CPU 速度及电量 掌上电脑用电量的多少与 CPU 工作负荷有关,即如果程序的数据结构与算法设计得比较合理,使 CPU 的计算量相对较少,则耗电量也减少。

如果 R 树、四叉树、Buddy 树作为只描述线型数据的主存索引,则四叉树空间占用量最小,Buddy 树空间占用量最大(Ning *et al.*, 2000)。对于点查询,Buddy 树速度最快,电量消耗最少,R 树与四叉树在速度与耗电上都差不多。对于范围查询,Buddy 树与 R 树性能比较接近,速度差不多,耗电量也差不多,但四叉树无论在速度还是耗电上都明显差一些。而对于最邻近查询,R 树索引速度最快,耗电量最大,四叉树的速度一般,耗电量最小。可见,没有哪种结构有绝对的优势,采用哪一种应根据系统实际应用来确定。

3.2.3 屏幕显示 与 PC 机相比,掌上电脑的屏幕通常比较小(Palm 通常为 160×160 , Windows CE 通常为 320×240),如果仅从矢量地图的显示角度考虑,则采用分层的形式来表示空间数据最为合理。因此如何针对这种分层结构建立索引也是考虑的主要方面。

4 基于嵌入式 GIS 的索引策略

由上面分析的嵌入式 GIS 数据的特点,笔者就

需要针对不同的应用系统分别制定不同的数据索引策略. 数据出于显示的考虑, 已经全部都采用了分层存储结构.

4.1 总体策略

当数据量比较大时, 将在程序内存建立主索引, 而数据文件保存在对象存储 RAM 上, 通过建立内存映射来读取, 这属于主索引. 当空间数据量非常大时, 则可用内存映射来实现空间索引, 而数据文件保存在 CF 卡等外存上, 这就类似于磁盘索引的空间索引方式. 这里笔者主要讨论的是第一种情况.

由于这几类应用数据特点完全符合 packed R 树建立的前提假设条件, 静态(或对于野外数据采集由于其数据量非常小几乎也可以看作数据是静态的)所以这里建树采用的是 packed R 树(静态批量建树的方法).

影响检索速度的一个重要因素就是检索树的深度. 采用静态批量建树, 这时 R 树的存储利用率是最大的, 几乎达到 100%; 索引所占的空间是最小的深度也是最小的. 我们知道 packed R 树的一个最大的缺点就是在插入方面的缺陷. 当需要插入一个数据的时候, 基本上是要对每一层的结点都需要进行分裂. 但是在电子地图中是不存在结点的插入操作. 也就是说地图数据库对于用户来说是只读的静态的. 所以笔者建树时采用 packed R 树.

4.2 针对具体应用的索引的制定

4.2.1 一般的电子地图浏览 尽管这类应用数据量比较大, 但是由于数据是分层的, 所以采用的是仅读取只参与显示的数据, 也就是当前编辑层. 数据的载入笔者是采用动态加载.

(1) 多级索引. 对于整幅地图建立一个线性表, 这张表用来实现管理每层地图的索引与当前图层的对应关系. 对于每一层地图, 固定每一棵树的深度. 这样如果树的深度大于事先规定的则将该层地图利用网格技术划分. 具体操作如下: (a) 规定的树的最大深度为 N , 地物的总数量为 M , R 树的扇出(fan-out)为 FAN ; (b) 根据地物的数量估计需要树的棵数为 $S = M/FAN^N$; (c) 使得 $n = S^{(1/2)}$, 这样将整幅地图划分为 $n \times n$ 的网格; (d) 对每一个网格建立一个 R 树.

(2) 动态加载. 对于上述建立的多级索引, 笔者在数据分层的基础上采用动态加载, 具体操作如下: 当程序启动后将对应整幅地图的索引表读入内存, 并常驻内存; 当用户指定图层查询的时候根据索引

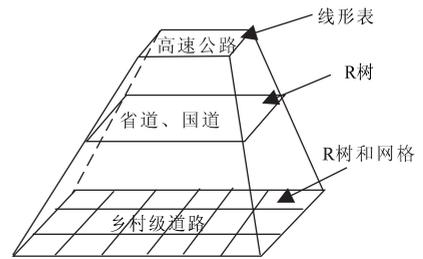


图 2 动态金字塔型的多级索引

Fig. 2 Dynamic pyramidal index

表查到对应的网格文件索引(如果没有则索引表中直接存储的就是该图层 R 树的索引)读入内存; 再根据查询的位置查找到对应的 R 树, 读入内存; 查询完毕释放相应内存; 仅保留整幅地图的索引表.

4.2.2 导游导航系统 对于导游导航系统, 笔者仍然是采用分级索引. 由于导游导航的分层特点, 将采用动态金字塔型的多级索引(图 2). 这里的 R 树仍然采用的是 packed R 树. 对于第 3 层的网格的划分, 与一般的电子地图浏览中的划分方法相同.

4.2.3 野外数据采集应用 对于这一类的应用, 总体思想是建立在电子地图浏览的基础之上的. 对于用户需要修改的数据, 采用建立用户文件. 也就是说对于地图采用电子地图的索引机制, 而用户修改增加的数据时建立用户索引文件. 对于这种野外采集, 用户添加的数量是有限的. 所以在搜索的时候首先搜索的是用户索引文件. 如果没有符合条件的则在系统的索引文件中查找. 这里查找的机制和电子地图的是一样的.

5 结论

由于移动终端有限存储和 GIS 空间数据的大数据量之间的矛盾, 分析了 2 者的特点, 同时根据嵌入式 GIS 应用的主体将其划分为电子地图浏览、导航系统和野外数据采集 3 大类. 结合已有的空间数据索引: R 树、四叉树、Buddy 树, 提出了基于主题导向的空间索引策略, 以提高在数据存取、内存使用和 CPU 性能方面的效率. 这些策略对于嵌入式系统都是可行的.

References

- Beckmann, N., Kerigel, H., Schneider, R., et al., 1990. The R*-Tree: An efficient and robust access method for points and rectangles. In: Proc. ACM SIGMOD conf.,

- Atlantic City, NJ, USA. 322—331.
- Dreamtech Soft Research Group, 2002. Programming for embedded systems. Electronics Industry Press, Beijing. 10—12(in Chinese).
- Guttman, A. , 1984. R-tree: A dynamic index structure for spatial searching. In: Proc. ACM SIGMOD conf. on the management of data. Waterloo, Canada. 47—57.
- Ning, A. , Sivasubramaniam, A. , Narayanan, V. , et al. , 2000. Analyzing energy behavior of spatial access methods for memory-resident data. *VLDB Journal* , 8: 410—420.
- Raghu, R. , Johannes, G. , 2004. Database management systems. Tsinghua University Press, Beijing. 719 (in Chinese).
- Roussopoulos, N. , Leifker, D. , 1985. Direct spatial search on pictorial data-bases using packed R-trees. In: Proc. ACM SIGMOD conf. , Austin, Texas. 17—31.
- Shekhar, S. , Sanjay, C. , 2004. Spatial databases: A tour. Electronics Industry Press, Beijing. 116—119.
- Zhang, T. Q. , Ling, X. K. , Yu, X. , 2004. The lower power design of SoC system. The Application of SCM and embedded system. <http://www.51eda.com>(in Chinese).

附中文参考文献

- Dreamtech 软件研发组, 2002. 嵌入式系统编程源代码解析. 北京: 电子工业出版社. 10—12.
- Raghu, R. , Johannes, G. , 2004. 数据库管理系统原理与设计. 北京: 清华大学出版社. 719.
- 张天骥, 林孝康, 余翔, 2004. SoC 系统的低功耗设计. 单片机及嵌入式系统应用. <http://www.51eda.com>.