

分布式空间拓扑连接查询优化处理算法

杨典华

(首都师范大学三维信息获取与应用教育部重点实验室, 北京 100048)

摘要: 针对传统分布式数据库查询应用于分布式空间数据库查询带来的传输和处理代价高的问题, 本文结合已有分布式跨边界片段连接优化方法, 深入研究了分布式空间拓扑连接查询处理, 提出跨边界连接优化的空间查询优化算法, 丰富了传统的分布式查询的关系代数等价变换规则。同时, 针对不同片段连接类型的分布式空间查询全局优化策略, 实现了分布式空间查询分解与数据本地化, 从而优化分布式查询中的数据传输所付出的高昂代价。最后, 提出了结点归并、连接归并树、执行结点、执行计划树等分布式查询优化方法, 利用相应归并和优化算法将全局空间查询转化为各个场地局部空间数据库的具体执行计划, 消除分布式查询中的冗余计算, 优化查询计算策略, 从而解决分布式空间查询中的处理代价高的问题。通过分布式空间查询实验表明, 本文的算法能够较好地提高分布式空间查询的性能。

关键词: 分布式空间数据库; 查询优化; 空间数据查询; 空间拓扑连接

DOI: 10.3724/SP.J.1047.2013.00643

1 引言

传统的分布式数据库查询的基本流程为查询分解、数据本地化、查询全局优化, 以及局部优化^[1-3]。分布式空间数据库系统(Distribute Spatial Database System)中查询处理的步骤与分布式数据库系统(Distribute Database System)基本相同^[3], 但相比传统数据库, 分布式空间数据查询处理更加复杂, 其查询流程中具体每一环节都不一样。这主要由于空间数据结构复杂、空间关系多样、数据量大, 导致空间查询处理涉及时间长, 而在分布式空间查询处理过程中, 传输代价和处理代价两者都不容忽视。因此, 分布式关系数据库系统的解决方案并不能直接应用于分布式空间数据库。

目前, 分布式空间查询处理中的空间拓扑连接操作主要有3种策略: (1)全连接策略(或称天真策略 NS(Naive Strategy)), 将一个场地参与连接的一个关系记录全部传到另外一个场地, 并利用集中式空间数据库的空间连接算法进行处理, 这种策略的主要问题是传输和执行代价都较高^[4]。(2)半连接策略, Kian-Lee Tan等人基于空间半连接(spatial semi-join)概念, 研究和设计了分布式空间连接算法, 以

减少空间数据传输量, 同时减少连接执行时间^[5-6]。

(3)MR2(Multiple step with remote indices, version 2)策略, 虽然效率与前2种策略相比有较大的提高, 但是, 该策略不适用于异构空间数据库的情形, 因此, MR2策略的适应性受到较大的限制^[7]。此外, 文献[8]在考虑到区域分片形式特殊性的情况下, 提出了分布式环境下的跨边界片段连接优化转化规则和相应的连接策略, 通过尽量去除不必要的片段连接和过滤空间片段中冗余的空间对象, 实现了在传输代价和局部处理代价两个方面的优化。

为了将已有的片段连接优化方法与传统的分布式查询处理过程有机的结合, 本文首先在已有的关系代数转换规则上, 提出了增加以跨边界连接优化的空间查询优化原则; 其次, 针对分布式空间数据库查询分解及跨边界片段连接优化与半连接优化结合的策略, 提出用连接归并树和执行计划树技术来实现分布式空间查询处理算法。

2 分布式空间拓扑连接查询优化处理

分布式空间查询优化处理主要包括3个部分:

(1)跨边界片段连接去除规则和优化转化, 其中包

收稿日期: 2013-05-15; 修回日期: 2013-08-24.

基金项目: 国家自然科学基金项目(40971232); 国家“863”计划项目(2012AA12A403)。

作者简介: 杨典华(1977-), 男, 博士生, 主要研究方向为高性能并行GIS。E-mail: yangdh@lreis.ac.cn

含了片段连接去除等3个规则;(2)在该规则的基础上实现分布式空间查询分解与数据本地化;(3)分割分片的分布式空间查询全局优化策略。

2.1 跨边界片段连接去除规则和优化转化规则

在传统的分布式数据库中,水平分片片段之间的连接存在许多多余的连接;在分布式空间数据库中,分割分片实际上是空间数据库的一种特殊的水平分片,而且水平分片的数据本地化方法可以用于分割分片。在文献[8]的基础上,本文提出的基于区域分片的片段连接优化规则如下:

片段连接去除规则:如果参与连接的2个空间片段的边界的(Minimum Bounding Rectangle)不相交,则片段间第1类连接可以去除。第1类连接是指片段间的拓扑相交连接,判断方法是从空间全局数据字典中得到的片段是否相邻或计算片段MBR的空间关系等。

片段间第1类连接优化规则:利用公共边界的MBR对2个片段的空间对象分别求交,再对2个交的结果集进行空间连接,鉴此,将2种情况过滤的矩形统一成为过滤MBR,即FMBR(Filter Minimum Bounding Rectangle)。2个跨边界片段的第1类空间连接可以转化为先用FMBR对2个片段的空间对象分别进行过滤,再对2个过滤后的结果集进行空间连接。其中,FMBR表示参与空间连接的片段边界MBR的交矩形。

片段间第2类连接优化策略:如果参与连接的2个片段的分割边界的MBR不相交,则片段间第2类连接可以转化为2个片段的笛卡尔积。第2类连接优化策略指片段间拓扑相离连接,如查询谓词disjoint。

2.2 分布式空间拓扑连接查询分解与数据本地化

空间拓扑连接查询的优化方法中,查询分解与数据本地化是实现分布式片段透明访问的基础。查询分解先将一般查询转换成关系代数表达式,分析得到的查询语法树,查询树构造的目的在于建立语法结构良好的语句成分树,即将完整的GSQL语句(Geographic SQL)分解成由多级语句成分所构成的树状结构^[9]。按关系代数等价变换规则实现查询重构^[10],改写后查询效率将会得到很大提高,总的说来可以概括为以下几点:

(1)施加于同一数据对象上的若干简单谓词可

以合并为一个复合谓词。

(2)选择运算应尽早执行,有时它能使执行时间呈数量级减少。

(3)投影应尽量早执行,但不先于选择。

(4)非空间操作尽量在空间操作(过滤,buffer)前做。

分布式数据本地化主要作用是利用数据分布信息定位查询数据,确定查询中涉及哪个片段,把一个全局关系上的查询落实到片段上的查询。将全局关系上的关系代数表达式,变换为在相应片段上的关系表达式。数据的本地化把全局查询的叶结点用分片模式去代替。水平分片的全局关系由多个片段的并组成,对于查询树上全局关系结点,用片段的并集去代替,将并操作上移,并根据相关规则进行简化。在已有研究成果的基础上,我们提出一些空间片段拓扑连接优化的关系代数转换原则,并结合传统分布式数据库的关系代数等价转换规则,实现分布式空间数据库查询的全局优化,其主要过程包括:

(1)去除不必要的跨边界片段连接,按空间片段连接的去掉规则去除多余的跨边界片段连接;

(2)尽量优化跨边界片段连接,按空间片段连接的过滤优化转化规则优化跨边界片段连接;对于不能优化跨边界连接,用空间半连接方法优化。如果是第2类片段连接,且2个片段的分割边界的MBR相交,则采用空间半连接进行优化。

2.3 分割分片的分布式空间查询全局优化策略

将全局查询按尽可能大的原则分解成片段查询后,片段查询主要由局部空间数据库进行优化和执行。全局优化主要是确定连接的顺序和选择连接执行的场地。在传统的分布式数据库中,对于连接操作的查询优化,究竟使用直接连接(全连接)方案还是用半连接方案,取决于对数据传输和局部处理的相对代价。采用半连接会导致通信次数增加和本地处理时间的增加。一般地,如果认为传输费用是主要的,那么,采用半连接方案处理策略比较有利(如SDD-1)。如果认为局部处理费用是主要的,则采用直接连接方案处理策略比较有利(如System R*)^[2]。在全局查询事务的优化策略问题上,通常采用的优化策略有2种:一是静态优化,即在事务执行前优化;二是动态优化,即根据实际的中间执行结果边执行边优化。动态优化的优化质

量好于静态优化,但是效率不如静态优化。通常采用动态优化和静态优化相结合的策略以改善系统性能,因全局优化在分布式中要比局部优化更为重要^[10]。根据跨边界片段连接优化准则,按并操作动态执行方式进行空间查询处理。在处理过程中,需要得到相关的全局信息,并确定片段连接操作的场地,以及并操作的执行场地。选择参与连接的其中1个片段所在的场地作为连接场地,也就是说不考虑把2个连接片段送到第3个场地上去执行连接。通过计算参与连接的2个片段之间的传输代价,选择传输代价较小的片段进行传输。

将片段连接类型与片段的MBR相交与否结合,可得到跨边界片段连接优化分类,如表1所示。

表1 片段连接的判断类型
Tab.1 The judgment of fragment join

	连接片段边界MBR不相交	连接片段边界MBR相交
第1类拓扑	类型1:	类型2:
相交连接	ONE_N_INSECT	ONE_INSECT
第2类拓扑	类型3:	类型4:
相交连接	TWO_N_INSECT	TWO_INSECT

针对不同的片段连接类型,选取不同的优化策略来实现查询总代价的最小化。其中,对于类型1、2、3可直接采用已提出的跨边界片段连接优化策略^[8]。对于类型4,相关研究表明,跨边界空间半连接比空间直接连接(天真策略NS)有效,因此,采用空间半连接策略。我们将分割分片下的分布式查询全局优化策略称为跨边界片段连接优化与半连接优化结合的优化策略。

3 分布式空间拓扑连接查询优化处理实现机制

针对上节提出的分布式空间查询优化处理规则,本节将讨论具体实现机制:首先,将空间查询转化为空间全局查询树,对该树进行预处理变换为连接归并树;然后,在连接归并树基础之上再进行查询分解和数据本地化,生成执行计划树;最后,按照执行计划树执行相关操作,得到查询结果。

3.1 连接归并树及其生成

全局查询树中的结点是关系代数中各种操作结点,同时包括空间操作结点。数据本地化主要考

虑的是连接结点,可以将连接结点提出来。但在优化过程中,其他结点同样要参与优化,因此,本文提出结点归并和连接归并树概念。

结点归并:对全局查询树上的某个连接结点(或者是叶结点)到其父连接结点(不包括父连接结点)之间的所有结点逻辑上合并为一个结点的过程,称之为该结点的归并化。为了简明扼要,本文将归并连接结点也简称连接结点,归并叶结点也简称叶结点。归并连接结点和归并叶结点同称为归并结点。

连接归并树:在连接结点的归并化过程中,全局查询树上连接结点的左右子连接结点(或者是叶结点)所对应的归并结点,仍然是该连接结点对应的归并结点的左右子结点。这样就形成了一个由全局查询树上所有的连接结点(或叶结点)所对应的归并结点组成的二叉树,称为连接归并树。这样,一个全局查询可以表示成一个连接归并树,反映全局查询中全局关系、中间结果,以及它们之间的全局连接关系。

如图1(a)表示一个全局查询,虚线部分表示全局查询树中有其他非连接结点,如图1(b)是对应于图1(a)的连接归并树。前序遍历全局查询树,对连接结点和叶结点进行归并化处理,可以生成连接归并树。

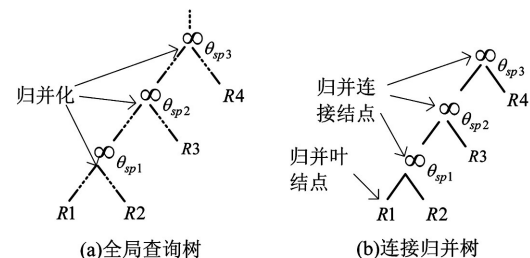


图1 结点归并与连接归并树

Fig.1 Node merging and join-merged tree

生成连接归并树的大致过程为:

- (1) 在前序遍历过程中,如果当前结点既不为叶结点,也不为连接结点,将该结点放入全局栈中;
- (2) 在前序遍历过程中,如果当前结点为连接结点,先将其与全局栈中的所有结点进行归并化,得到归并连接结点,并且清空全局栈;然后分别对左右子树进行归并化处理,将左右子树的归并化结点作为本归并化结点的左右子结点;
- (3) 在前序遍历过程中,如果当前结点为叶结点,则将其与全局栈中的所有结点进行归并化,得

到归并化叶结点,并且清空全局栈。

3.2 数据本地化与优化

在连接归并树上进行查询分解、数据本地化和优化。数据本地化与优化按后续遍历连接归并树完成,过程如下:

(1)在后序遍历过程中,如果当前遍历结点的左子结点不为空,而且结点是归并连接结点,则对左子树进行处理,得到子树Q1;

(2)在后序遍历过程中,如果当前遍历结点的右子结点不为空,并且结点是归并连接结点,则对右子树进行处理,得到子树Q2;

(3)处理当前连接结点:

①如果当前结点的左右子结点存在一个归并连接结点或两个都是归并叶结点,则对当前结点进行数据本地化处理,并生成一个数据本地化并优化处理后的查询子树,这个子树是一个根结点为并(U)操作的子树(称为并操作子树)。按分割分片的分布式空间查询优化策略对以当前结点为根结点的子树进行跨边界片段连接优化处理:去掉冗余的片段连接,并按不同的连接类型进行优化转换;生成查询执行计划;

②如果当前结点的左右子结点均为归并连接结点(即均为中间结果),结合Q1和Q2,生成查询执行计划;

③将当前结点生成查询执行计划的执行结果记为Q3,返回Q3。

3.3 执行计划生成查询结果

通过执行结点和执行计划树两个概念来阐述执行计划,并通过执行计划树来表达。

执行结点是全局查询分解和数据本地化后,按照尽可能大的策略分解的片段查询及其执行场地信息。这个片段查询在指定的执行场地上执行。执行场地是根据连接顺序,以及执行代价来确定的。因此,执行结点包括执行场地、片段查询等相关属性信息。全局调度程序将执行结点信息发送到执行场地。执行场地根据多级安全数据库系统LSDBMS (Levels Security Database Manage System)类型,决定执行的方式。

执行计划树是在连接归并树的基础上,将归并连接结点进行数据本地化、优化后,按执行结点来组织的一个全局查询树。通过对连接归并树进行后序遍历,将归并连接结点转化成以并操作为根结

点的子树或以2个中间结果进行连接的子树,生成执行计划树。实际生成时,执行计划树在数据本地化与优化的处理(当前)连接结点的步骤中完成,对连接归并树的遍历已隐含在数据本地化与优化的处理过程中。执行计划树具体生成过程如下:

(1)按尽可能大的片段查询分解策略将子树分解成片段查询,这步实际上在归并化过程中已完成;

(2)针对各片段查询,生成执行结点,确定片段查询的执行场地,得到当前归并结点的查询计划子树。

在得到执行计划树后,调度服务器根据执行结点的执行场地信息,将每个执行结点发送到其对应的执行场地执行。执行代理根据片段连接类型(表1),LSDBMS的能力及执行结点中的执行信息决定片段连接的执行方式,最后得到整个执行计划树的查询结果。

4 算法验证

为了验证算法的有效性,本文以我国1:20万图的矢量数据为例,查询城市与河流对象之间的空间拓扑关系。实验数据分别存储在两台服务器上,两台服务器采用的是四核Intel Xeon5405*2的中央处理器,内存大小2GB,服务器采用的操作系统为Windows 2003企业版,并安装了Oracle Spatial 9i空间数据库,并通过Internet相连。设空间关系Town、River按分割集 $S=\{s1,s2\}$ 进行分割(这里设s1、s2相邻),分割得到片段为:

$$PFS(Town) = Town1 \cup Town2$$

$$PFS(River) = River1 \cup River2$$

且数据分布为:

场地1(Site1): Town1, River1

场地2(Site2): Town2, River2

现有查询Q4.1

Q4.1: SELECT Town.Name

FROM Town, River

WHERE River.Name = "L" AND Intersect
(Town.Shape, River.Shape) = true

通过访问全局字典和判断片段MBR的空间关系,可判断片段连接类型为ONE_INSECT,即片段边界MBR相交的第1类拓扑相交连接,则采用跨边界片段连接优化策略执行分布式空间查询。图2表示查询Q4.1由全局查询树到局部片段查询(q1,q2,

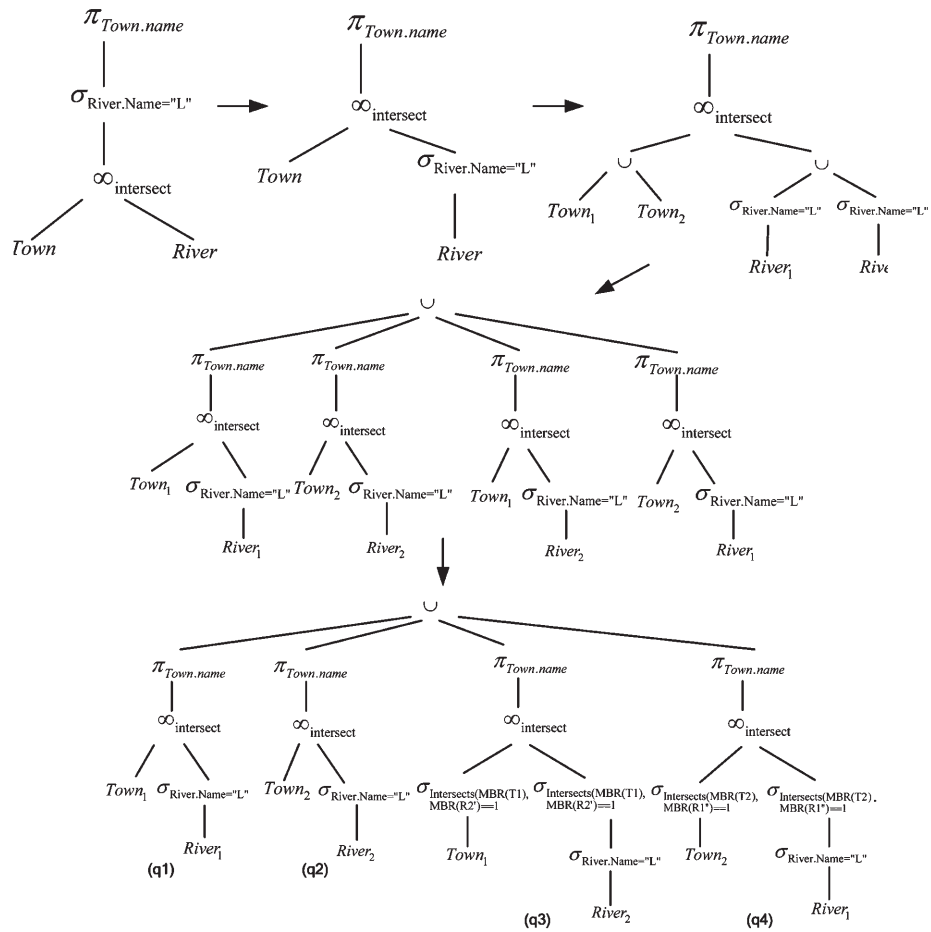


图2 基于空间拓扑连接优化的关系代数转换过程

Fig.2 The space topology optimization-based conversion of relational algebra

q3, q4) 的转换过程, 在该过程中利用本文提出的空间片段拓扑连接的关系代数转换原则对跨边界片段查询 q3 和 q4 做了进一步优化。

以实验数据为例, 我们利用本文提出的查询优化策略(PFJS)与半连接策略(SJS)进行查询代价比较, 前提条件是数据和计算环境均一致。如图3显示, 在整体查询时间上PFJS共花费了93.22s, 而SJS共花费了128.56s。2种查询方法虽然在算法复杂度上均为 $n\log(n)$ (n 数据记录数), 但相比于半连接策略, 本文的优化主要体现在片段查询q3与q4增加了一个交矩形过滤步骤, 减少片段间空间连接数和传输的数据量, 从而在总体上提高了查询效率约28.5%。

5 结论

在传统的分布式数据库查询处理的基础上, 结

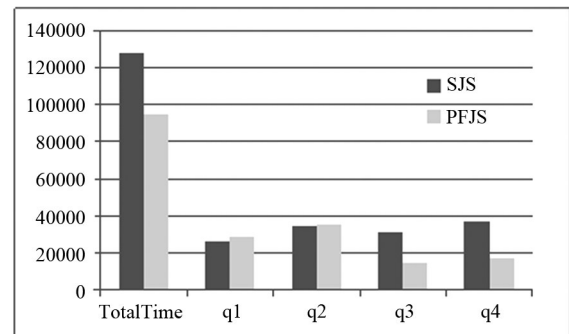


图3 查询代价对比(ms)

Fig.3 The cost comparison of query

合已有的片段连接优化规则, 本文深入研究并实现了分布式空间拓扑连接查询处理, 主要提出了跨边界连接优化的空间查询优化算法, 进一步丰富了传统的分布式查询的关系代数等价变换规则, 讨论了针对不同片段连接类型的分布式空间查询全局优化策略。最后提出了结点归并、连接归并树、执行

结点、执行计划树等概念。基于这些概念,经过查询分解与数据本地化后的空间查询树可以转化为执行计划树。连接归并树中的一个结点实质上是一个最大化的片段查询,生成执行计划树后,转化为执行计划树中的结点,这样全局空间查询就可以转化为各个场地局部空间数据库的具体执行计划。

参考文献:

- [1] 邵佩英. 分布式数据库系统及其应用[M]. 北京: 科学出版社, 2003.
- [2] Özsu M T, Valduriez P. Principles of distributed database systems[M]. 2nd ed.. New York: Prentice-Hall Inc., 1999.
- [3] 周玉科, 马廷, 周成虎, 等. MySQL 集群与 MPI 的并行空间分析系统设计与实验[J]. 地球信息科学学报, 2012, 14(4): 448-453.
- [4] Ramirez M R, de Souza J M. Distributed processing of spatial join[J/OL]. <http://cronos.cos.ufrj.br/publicacoes/reltes58902.pdf>, 2001.
- [5] Abel D J, Ooi B C, Tan K L, et al. Spatial join strategies in distributed spatial DBMS[C]. // Egenhofer M J, Herring J R (Eds.). Proc. of the 4th Int'l Symp. Advances in Spatial Databases. London: Springer-Verlag, 1995: 348-367.
- [6] Kian-Lee Tan, Beng Chin Ooi. Exploiting spatial indexes for semijoin-based join processing in distributed spatial databases[J]. IEEE Transactions on Knowledge and Data Engineering, November/December, 2000, 12(6): 920-937.
- [7] Ramirez M R, de Souza J M. Distributed processing of spatial join[C]. // Proc. of the Anais do III Workshop Brasileiro de GeoInformática-GeoInfo, 2001: 1-8.
- [8] 朱欣焰, 周春辉, 吴维, 等. 分布式空间数据分片与跨边界拓扑连接优化方法[J]. 软件学报, 2011, 22(2): 269-284.
- [9] 黄舟, 彭霞, 张珂, 等. 分布式计算环境中的空间查询语言全局解析机制[J]. 地理与地理信息科学, 2006, 22(3): 18-21.
- [10] 陈波, 高秀娥, 陈来杰. 基于等价变换的分布式查询优化方法研究[J]. 计算机工程与设计, 2006, 27(3): 390-392.
- [11] 刘书, 李正凡. 基于分布式数据库系统的一种查询优化算法[J]. 北京联合大学学报(自然科学版), 2005, 19(1): 51-54.
- [12] Clematis A, Mineter M, Marciano R. High performance computing with geographical data[J]. Parallel Computing, 2003(29): 1275-1279.
- [13] Hawick K A, Coddington P D, James H A. Distributed frameworks and parallel algorithms for processing large-scale geographic data[J]. Parallel Computing, 2003, 29(10): 1297-1333.
- [14] Armstrong M P, Marciano R J. Local interpolation using a distributed parallel supercomputer[J]. International Journal of Geographical Information Systems, 1996, 10(6): 713-729.
- [15] Ren Y C, Shen L, Yang C J, et al. Parallel map rendering system for massive geospatial data in distributed environment[C]. Proceedings of International Conference on GeoInformatics, 2009.
- [16] Sorokine A, Daniel J, Liu C. Parallel visualization for GIS applications. <http://www.geocomputation.org/2005/Abstracts/Sorokine.pdf>, 2005.
- [17] Hoel E G, Samet H. Data-parallel polygonization[J]. Parallel Computing, 2003(29): 1381-1401.
- [18] Lanthier M, Nussbaum D, Sack J-R. Parallel implementation of geometric shortest path algorithms[J]. Parallel Computing, 2003(29): 1445-1479.
- [19] Armstrong M P, Densham P J. Domain decomposition for parallel processing of spatial problems[C]. Computers, Environment, and Urban Systems, 1992(16): 497-513.
- [20] Ware A, Kinder D. Parallel implementation of the Delaunay Triangulation within a transputer environment[C]. Proceedings of the European Geographic Information Systems (EGIS'91) Conference (Utrecht/L EGIS), 1991, 1199-1209.

Research on the Distributed Spatial Topological Query Optimization Algorithm

YANG Dianhua*

(Key Laboratory of 3D Information Acquisition and Application, Ministry of Education, Capital Normal University, Beijing 100048, China)

Abstract: Due to complex data structure, complicated spatial relationship and massive data volume, distributed spatial query is a time-consuming processing, which will cause high transmission and processing cost. Query pro-

(下转第 679 页)

and the subdivided smooth virtual boreholes, 3D geological models of survey regions can be constructed, matching geostatistics patterns, being C^1 geometric continued, and showing favorable visual effect. Differing from the traditional modeling methods that regard virtual boreholes as complements to expert knowledge, the proposed method introduces virtual boreholes into each step of the construction of three-dimensional geologic models as the intermediate elements of the core algorithm, simplifying the realization process of the modeling algorithm effectively, and ensuring the stability and efficiency of the algorithm.

Key words: 3D geological modeling; virtual boreholes; spatial interpolation; pinch-out; subdivision algorithm

***Corresponding author:** ZHOU Liangchen, E-mail: zhouhch@hotmail.com

(上接第 648 页)

cessing method in traditional distributed database cannot satisfy the demands of query in distributed geospatial database. Therefore, new query methods in distributed geospatial database need to be studied. In this paper, the distributed spatial join query processing is deeply studied based on the existing optimizing methods of the conventional query processing in traditional distributed database, and a series of transformation rules of relational algebra expression based on cross-border topological join optimization rules are proposed. The processed query tree is optimized by equivalent transformation after data localization. The global optimized method of distributed spatial join query for different fragments is studied. The global spatial query can be transformed into some local fragments joins effectively. The spatial join query is processed in the local area, avoiding the data transmission of spatial data among data nodes during the processing of query, so that the query performance can be improved. To improve the efficiency of the method, some new concepts were put forward, including query merged tree and execution plan tree, which can optimize the executing path of query plan. For example, by adjusting the executing order, some processes with low cost execute first, and the time-consuming processes execute based on the result set generated by the previous processes so as to reduce the process of time-consuming parts and resolve the problem of high cost of query processing to improve the performance of distributed spatial query. The experiment based on the vector data of China shows our methods can reduce the cost of the spatial join and data transmission among the nodes, and the performance improve 28.5%, which demonstrates that our methods outperform the traditional methods in terms of both algorithm complexity and the running time.

Key words: distributed spatial database; query optimization; spatial data query; spatial topological join

***Corresponding author:** YANG Dianhua, E-mail: yangdh@lreis.ac.cn