连续不确定 XML 数据索引技术研究

张换香[1] 张晓琳[2] 刘立新[2]

[1](内蒙古科技大学工程训练中心 内蒙古 包头 014010)

[2](内蒙古科技大学信息工程学院 内蒙古 包头 014010)

摘 要 针对连续不确定 XML 数据的概率阈值查询,提出 CPTI(Continuous Probabilistic Threshold Index)索引技术,包括 CPTI 结构索引和 CPTI 值索引。CPTI 结构索引扩展了结构索引 F-index 支持连续不确定 XML 数据,通过 CPTI 结构索引查询 twig 小枝,并确定小枝的路径概率;CPTI 值索引是一个二维表,记录 cont 类节点的概率信息,通过 CPTI 值索引过滤与查询无关的元素以减少查询中需要处理的元素数目。实验表明,此索引技术可极大地提高查询处理的性能。

关键词 连续不确定 XML 索引 概率阈值查询

中图分类号 TP392 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2013.08.014

ON INDEXING TECHNOLOGY OF CONTINUOUS UNCERTAIN XML DATA

Zhang Huanxiang[1] Zhang Xiaolin[2] Liu Lixin[2]

[1](*Engineering and Training Center, Inner Mongolia University of Science and Technology, Baotou 014010, Inner Mongolia, China*)

[2](*School of Information Engineering, Inner Mongolia University of Science and Technology, Baotou 014010, Inner Mongolia, China*)

Abstract In light of the probabilistic threshold query of continuous uncertain XML data, we propose indexing technology of continuous probabilistic threshold index (CPTI), including CPTI structure indexing and CPTI value indexing. CPTI structure indexing extends the structure index F-index to support continuous probabilistic XML data; through CPTI structure indexing it is able to query twig and to determine the path probability of twig; CPTI value indexing is a two-dimensional table recoding the probability information of the nodes of cont class, through CPTI value indexing to filter the elements which is nothing to do with the query, it is able to reduce the number of element to be processed in query. Experiment shows that this index technology can greatly improve the performance of query processing.

Keywords Continuous uncertain XML Indexing Probabilistic threshold query

0 引 言

由于 XML 数据的灵活性,自描述性好及可扩展性高,成为当前主流的数据形式,并成为 Internet 中进行数据交换和表示的标准。由于客观世界的复杂性,不确定性是数据常见的内在属性,因此不确定的信息是普遍存在的。通常不确定信息以概率值的形式在 XML 文件中表示。如何在连续不确定 XML 中建立索引实现快速高效的查询成为了当务之急。

索引是提高查询效率的有效途径,DataGuides[1]、1-index[2]、A(k)-index[3]、D(k)-index[4],都是其中典型的代表。但是这些索引结构有个共同的特点就是仅支持简单路径查询,不支持分支路径查询。文献[5]提出一种扁平结构索引 F-index,能够快速过滤所有与查询无关的索引结点,进而过滤掉与查询无关的元素序列。在处理深度嵌套的复杂结构 XML 文档时具有很大的优势,但是这种索引结构仅适用于普通 XML 文档中的查询。文献[6]提到一种处理连续不确定数据的索引方法,这种方法通过对节点提前计算一些附加信息,在查询时通过这些信息过滤与查询无关的节点,最小化概率阈值查询中概率计算的次数。但是这种索引只适用连续不确定数据的查询处理,对于连续不确定 XML 文档没有实际应用。

在连续不确定 XML 中进行的查询,多数只需要知道取得某个值的概率是否超过了一个给定的阈值,即概率阈值查询。提出 CPTI 索引技术。首先扩展了结构索引 F-index,建立了概率 XML 数据的扁平结构链表,此链表在原有的普通 XML 数据扁平结构链表的基础上又添加了结点状态(普通结点和分布式结点)和相应的概率信息,查询可直接在链表里进行,这种结构可快速的返回 twig 小枝的查询结果,并且可以确定节点的路径概率值(即从根节点到本节点的路径概率);其次建立了值索引,此索引在服从连续分布的叶子结点,记录了结点概率信息,查询时先根据此概率信息过滤掉一些与查询无关的叶子节点,减少叶子节点概率的计算。

收稿日期:2012-07-06。国家自然科学基金项目(61163015);内蒙古自然科学基金重点项目(20080404Zd21)。**张换香**,讲师,主研领域:数据库技术。**张晓琳**,教授。**刘立新**,讲师。

1 CPTI 索引

1.1 建立模型

一个 PXML 文档可表示成一棵树，记作 $T=(Vp,rp,Ep,tag)$。其中：(1) Vp 是结点的集合。(2) $rp \subseteq Vp$ 是树的根结点。(3) Ep 是边的集合。(4) $tag:VA \rightarrow <name,value,valuetype>$，给每个结点赋予一个三元字符串组，分别表示该结点的节点名、值和值的类型，如图 1 所示是包含 mux 和 cont 类节点的 P-文档[7]。

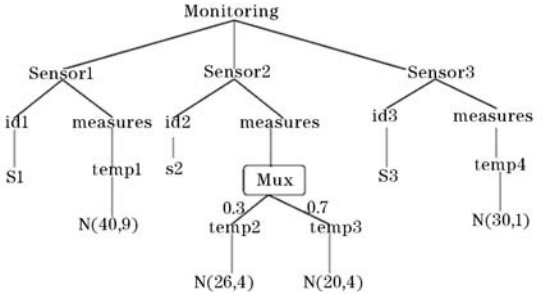

图 1 $PrXML^{\{mux,cont\}}$ 概率文档

1.2 CPTI 结构索引

CPTI 结构索引是一个链表，记录一个节点 A 和它的所有 tag 为 B 的后代节点的情况，链表结构如图 2 所示。图 2(a)是链表表头，其中 PC 表示父子关系，AD 表示祖先子孙关系，图 2(b)为链表元素，此链表是在结构索引 F-index 链表元素的基础上增加了三个元素，Flag 表示后代结点的类型，包含 F、Fi、和 Fm，F 表示普通节点，Fi 表示独立节点，Fm 表示互斥节点；P 表示后代节点的路径概率值，即从根节点到该节点的路径概率；CurrentNode 表示对应的后代结点。

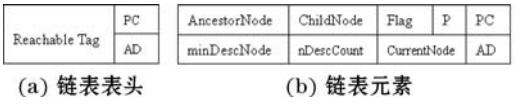

图 2 链表结构

记录图 1 中所有非叶节点和它后代的可达性信息，建立了如图 3 所示的 CPTI 结构索引。其中 Mo 表示 monitoring、Si 表示 sensori、Ms 表示 measures、Ti 表示 tempi。

图 3 CPTI 结构索引

1.3 CPTI 值索引

1.3.1 正态分布的概念及特征

若连续型随机变量 X 的概率密度为 $f(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$，$-\infty<x<\infty$ 其中 $\mu,\sigma(\sigma>0)$ 为常数，则 X 服从参数为 μ,σ^2 的正态分布或高斯分布，记为 $X\sim N(\mu,\sigma^2)$。正态分布的概率密度曲线如图 4 所示，曲线关于 $X=\mu$ 对称。要使得落在区间 (x_1,x_2) 上的概率为 p，则区间 (x_1,x_2) 存在三种可能情况：

$$\begin{cases} x_1 < X < x_2 \\ -\infty < X < x_R \\ x_L < X < +\infty \end{cases}$$

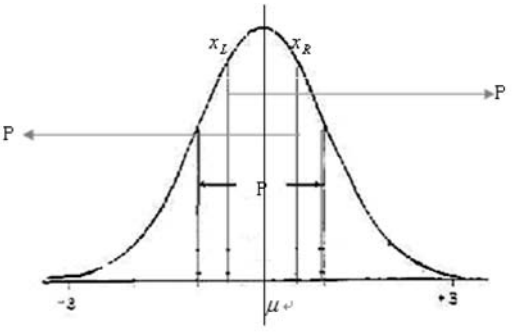

图 4 正态分布概率密度曲线

1.3.2 值索引

值索引是一个二维表结构，记录 cont 类节点概率值和对应区间关系的信息表，值索引结构如图 5 所示，P 表示用户给定的概率值，$0<P<1$；$|x_2-x_1|$ 表示图 4 中关于 $X=\mu$ 对称的、概率为 P 的最短区间长度；x_L、x_R 含义与图 4 中 x_L 和 x_R 含义相同。查询时，可根据此表信息过滤与查询无关的元素以减少处理元素的数目。

P	$\lvert x_2-x_1\rvert$	x_L	x_R

图 5 CPTI 值索引

图 1 中叶子结点 T 服从正态分布 $N(\mu,\delta^2)$，根据 T 的实际分布情况，计算得到一些信息，构成一个信息表，例如 T2 结点，设初值 0.1，步长 0.1，确定 P 值，并计算得到图 6 所示值索引。

p	$\lvert x_2-x_1\rvert$	x_L	x_R
0.1	0.52	28.58	23.42
0.2	1.04	27.7	24.3
0.3	1.56	27.06	24.94
0.4	21.2	26.52	25.48
0.5	2.72	26	26
0.6	3.38	25.48	26.52
0.7	4.14	24.94	27.06
0.8	5.14	24.3	27.7
0.9	6.58	23.42	28.58

图 6 CPTI 值索引实例

2 基于 CPTI 索引的查询处理过程

例如查询图 1 中温度在(28,31)范围内的概率 P 大于 0.6 的传感器 s，如图 7 所示。使用 CPTI 结构索引查询图 7 中的 Twig，利用 CPTI 值索引过滤不满足 *Temp* 在(28,31)的概率大于 0.6 这一条件的 Twig。

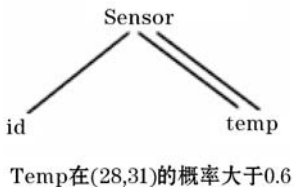

图 7　查询实例

2.1　CPTI 结构索引查询 Twig

CPTI 结构索引查询步骤:

(1) 通过 CPTI 结构索引找到 S-id 的 PC 指针和 S-T 的 AD 指针,两指针同时推进,比较两个指针所指链表中 AncestorNode 是否相同,如果相同,则找到符合条件的小枝。如果不同,则继续推进,直到 PC 或 AD 为空。

(2) 根据链表元素 P 确定 T 的节点类型和路径概率。如果 T 的路径概率低于查询概率阈值将被过滤掉,否则保留。

根据以上策略,最终找到如图 8 所示三个小枝。

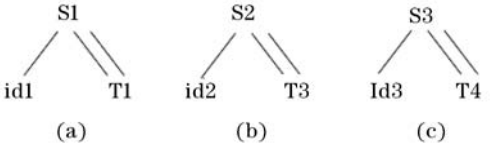

图 8　找到的三个小枝

2.2　CPTI 值索引过滤 Twig

CPTI 值索引过滤步骤:

(1) (a,b)是概率阈值查询的查询区间,P, x_1, x_2, x_L, x_R 含义和图 4 中表示的含义相同,查询的概率为 Pi = 阈值概率/路径概率。本例中 $a=28, b=31, Pt=0.6/(T$ 的路径概率)。

(2) 当 $|b-a| \leqslant |x_2-x_1|$ 时,T 被过滤掉。

(3) 当 $|b-a| > |x_2-x_1|$ 时,如果 $a < x_1 < x_2 < b$,则 T 满足条件;如果 $b \leqslant x_R$,则 T 被过滤掉;如果 $a \geqslant x_L$,T 也被过滤掉。

(4) 其余情况均利用 pdf 进行计算。

根据以上过滤策略,只有 T4 符合条件。所以符合图 7 查询的只有 S3[/id3]//T4。

3　实验分析

3.1　实验环境和数据集

本实验是在 Dell Optiplex 380(2.93GHz),RAM 2GB,300G 硬盘上运行,OS 是 Windows XP Professional SP-3。实验测试采用人工合成数据集。

3.2　测试及结果分析

本实验进行了两组测试。第一组测试中,数据集如表 1 所示,P 文档逐渐增大,分别测试了没有索引存在、只有结构索引存在、结构索引和值索引都存在时的运行时间。结果如图 9 所示,从图中可以看出通过索引进行查询,查询时间大幅度地减少,并且发现,通过 CPTI 结构索引处理查询时,P 文档越大,时间变化幅度越小,效率越高。

表 1　测试数据集

P-document	Pdoc2	Pdoc2	Pdoc3	Pdoc4	Pdoc5	Pdoc6	Pdoc7	Pdoc8	Pdoc9	Pdoc10
Size of P-document(MB)	5	11	19	27	36	44	56	62	70	83

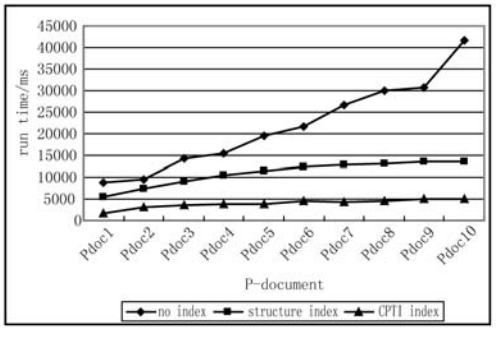

图 9　有无索引和 p-文档大小对查询时间的影响

第二组测试中,P 文档不变,44MB,只改变查询的概率值($P1 \sim P9$ 分别是 0.1 ~ 0.9,步长 0.1),测试了运行时间,如图 10 所示,从图中可以看出,概率值越大,运行时间越短,即概率值越大,通过 CPTI 索引查询的效率越高。

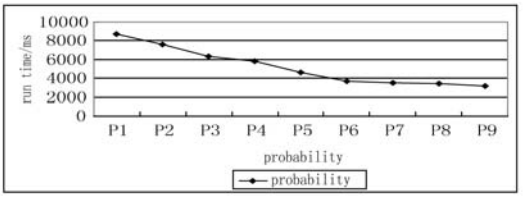

图 10　查询概率值对查询时间的影响

4　结　语

本文在已有 XML 索引方法的基础上提出了 CPTI 索引结构,可以实现连续不确定 XML 的概率阈值查询,使用 CPTI 结构索引加速了 Twig 查询,通过 CPTI 值索引过滤 Twig,进一步减少了查询时间。实验表明,效率较高。进一步的工作是对叶子节点服从任意分布的情况进行研究。

参　考　文　献

[1] Goldman R, Widom J. DataGuides: Enabling query formulation andoptimization in semistructured databases [C]//Proc. of the 23rdInt'l Conf. on Very Large Data Bases (VLDB), Athens: Morgan KaufmannPublishers, 1997:436 - 445.

[2] Milo T, Suciu D. Index structures for path expressions [C]//Proc. of the 7th Int'l Conf. on Database Theory (ICDT), LNCS 1540, Jerusalem: Springer-Verlag, 1999:277 - 295.

[3] Kaushik R, Sheony P, Bohannon P, et al. Exploiting localsimilarity for efficient indexing of paths in graph structured data[C]//Proc. of the 18th Int'lConf. on Data Engineering (ICDE), San Jose: IEEE Computer Society, 2002:129 - 140.

[4] Chen Q, Lim A, Ong K W. D(k)-index: An adaptive structuralsummary for graph-structured data [C]//Proc. of the 2003 ACMSIGMOD Int'l Confon Management of Data (SIGMOD), San Diego: ACM Press, 2003:134 - 144.

[5] He H, Yang J. Multiresolution indexing of XML for frequentqueries [C]//Proc. of the 20th Int'l Conf. on Data engineering(ICDE), Boston, IEEE Computer Society, 2004:683 - 694.

[6] 周军锋,孟小峰,蒋瑜,等. F-index:一种加速 Twig 查询处理的扁平结构索引[J]. 软件学报,2007,18(6):1429 - 1442.

[7] Kimelfeld B, Sagiv Y. Matching twigs in probabilistic XML[C]// VLDB07, Vienna, Austria, 2007.