一种基于流索引的 XML 数据流的 Xpath 查询算法

An XML Data stream Xpath Query Algorithm Based on Stream Index

魏东平　张宏伟　王金凤　（中国石油大学 计算机与通信工程学院　257061）

摘要:XML 数据流查询过程中,核心操作是父子、祖孙节点的匹配问题。解决这一问题的有效途径是根据查询表达式构造非确定状态自动机,顺序处理解析后的节点,在这个过程中大量与查询无关的节点也参与了匹配。通过对 XML 数据流添加流索引,在执行查询时,直接跳过与查询不匹配的元素及其子树节点,提高了查询效率。

关键词:XML　数据流　流索引　查询

1　引言

由于 xml 的半结构化特征,xml 数据的处理必须经过解析器解析。在基于非确定状态自动机(NFA)的 XML 数据流查询中[1],算法需要完全扫描一遍所有的 XML 元素。事实上,如果当前的节点不能与查询匹配,那么该节点的所有子树节点也不能与查询匹配。如果我们能跳过这些对查询没有影响的无用节点,直接让解析器定位到下一个节点,就能够进一步提高查询速度。基于此,我们考虑引入索引机制快速定位 XML 流的元素,减少 XML 文档的解析量。

2　流索引的引入(Stream Index)

为 XML 文档建立索引是提高 XML 查询的有效手段,我们通过对 Numbing Schemes[2] 的扩展,提出了 XML 的流索引结构。即对 XML 节点树的每一个节点建立如下三元组:定义节点 Xi 的流索引是一个三元组(start ,sub_nodes ,level),记为 SI(Xi) 其中,start 是节点在数据流中的起始位置;sub_nodes 是 <Xi> 与 </Xi> 之间节点的个数,若 Xi 是文本节点,则 sub_nodes 的值为 1;level 是该节点在 XML 树中的层次信息。当访问的该节点的结束标志时(end element),确定该节点的 sub_nodes 信息.。一个 XML 流可以看作是对 XML 文档树的前序遍历,因此我们能够以流的方式实现对 XML 树的访问。

下面是引入流索引后一个查询匹配的例子:查询语句 Q = /A/C;XML 文档和相应的 NFA 如图 1 所示。

NFA 初始状态为 1,我们以数据流的形式读入 XML 文档。

(1) 解析节点 <A>,元素 A 驱动 NFA 进入状态 2;

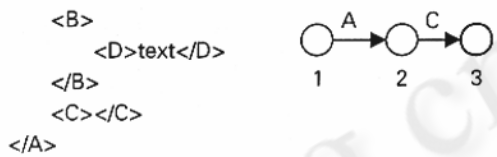

图 1

(2) 解析节点 <B>,元素不能驱动 NFA 产生任何状态转移,因此 <B> 及其子树节点对 Xpath 匹配结果不会产生影响;

(3) 通过流索引,得到 <B> 的 sub – nodes 值,直接到达节点 <C>

(4) 解析节点 <C>,驱动 NFA 到达终态 3,匹配成功

3　流索引的创建过程

下面介绍一个 XML 文档的流索引的创建过程,我们使用 SAX 解析 XML 流数据。并通过堆栈处理遍历过程中的节点信息。

(1) 对于元素 <Xi>,将其入栈保存,并保存与其对应的 start 和 level 信息;

(2) 解析器解析到文本节点时,将其入栈保存,由于该文本元素的 sub_nodes 是 1,因此可以直接得到该

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

文本节点的索引值;

(3) 解析器解析到 </Xi > 时,将栈中的元素出栈,直到元素 <Xi > 为止,并根据出栈元素的个数计算 <Xi > 的 sub_nodes 值;

(4) 当全体 XML 数据流解析结束后,所有节点的索引值也随之创建完毕。

4 基于流索引的 NFA

引入了 XML 流索引后,我们需要同时传输 XML 流数据及其流索引,并在 SI 的帮助下进行解析工作。注意到 SI 既是对流数据的索引,其自身也能以数据流的方式传输和表示,因此我们可以统一地对他们进行处理。我们将基于流索引的 XPATH 查询定义为如下问题。

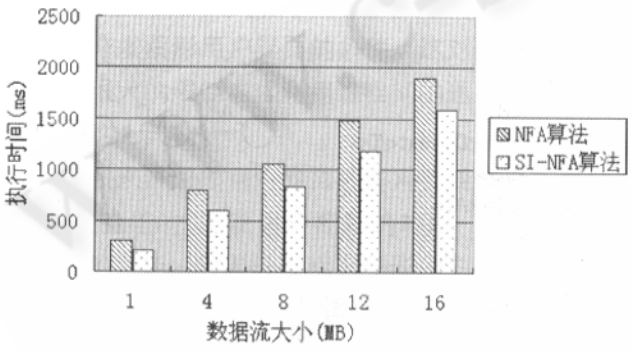

图 2

基于流索引的 XML 数据流的 XPATH 处理:已知一个 XML 数据流 SD,其树形结构的流表达为 ST = (X_0,X_1,X_2…X_n),该 XML 流的流索引为 SI = (S_0,S_1,S_2…S_n),给定一个路径查询表达式 Q,返回结果集 Result = (R_0,R_1,R_2…R_m,且 R_i(i = 1,2,…m) 是 SD 上对查询 Q 匹配的所有元素节点。

与一般的 NFA 驱动匹配算法相比,基于流索引的 NFA 的驱动算法最大的优点是:当一个元素节点不是 NFA 中任意一个当前活动状态的转移条件时(即该节点无法驱动 NFA 的状态转移),则表明该节点元素及其所有子树节点都不会对 XPATH 的查询结果产生影响。此时,可以利用索引信息,跳过 XML 流中的对应数据,从而最大程度地减少了 XML 文档的解析量,提高处理效率。下面是基于流索引的 XML 数据流的 Xpath 匹配算法,根据 Xpath 构造 NFA 的算法从略。

```
算法:SI – NFA 匹配
输入:XML 数据流和索引流
输出:匹配结果集合 Result
Algorithm SI – NFA
  Result = { } / * 查询结果集 * /
  SIndex = next (SI) / * 从流索引中得到下一个元素的索引 * /
  Repeat
    Element = SD. parse (SIndex. start) / * 根据 SIndex. start,定位 XML 流中元素的位置并解析 * /
    NFA. addInput (element) / * 将 element 作为 NFA 的输入 * /
    If (not NFA. activeStates. changed ( )) / * 如果该元素没有使 NFA 中的状态转移 * /
      SIndex = skipSIndex (SI,SIndex) / * 则跳过该元素及其子树节点 * /
    Else if (NFA. reachEndStarts ( ))
      R = R ∪ element; / * 如果到 NFA 的终态节点,则匹配成功 * /
    End if
    SIndex = next (SI) / * 读入下一个节点的索引 * /
  Until ( eof(SD)) / * 直到数据流结束 * /
  Return Result
End Algorithm
Function skipSIndex (SI, SIndex) / * 跳过该元素节点及其子树节点 * /
  endPosition = SIndex. statr + SIndex. sub_nodes
  Repeat
    SIndex = next (SI) / * 从流索引中读入下一个节点的索引 * /
  Until (SIndex. start > endPosition or eof (SI))
  Return SIndex
End Function
```

5 实验结果及分析

我们用 Java 和 Xerces – J2. 5 实现了基于非确定状态自动机(NFA)的 XML 数据流的 Xpath 查询算法和

(下转第 111 页)

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

(上接第 108 页)

基于流索引的 XML 数据流的 Xpath 查询算法，数据集是 XMark[4]，在 Pentium 4 2.40GHZ，256M 主存，Windows XP 平台下进行测试。使用相同的查询语句 Q ="/site//europe/ * /mail/date"，测试结果如图 2 所示。实验表明算法具有线性的时间复杂度而且在流索引的帮助下 SI－NFA 的查询效率高于基于 NFA 的查询。

参考文献

1 Al－Khalife, H. V. Jagadish, Nick Koudas, Jignesh M. Patel, Divesh Srivastava, Yuqing Wu. Structural Joins: A Primitive for Efficient XML Query Pattern Matching, In: Processing of International Conference on Data Engineering (ICDE), 2002.

2 Mehmet Altinel, Michael J. Franklin. Efficient Filtering of XML Document for Selective Dissemination of Information. Proceedings of the 26th VLDB Conference, Cairo, Egypt, 2000.

3 Alan Halverson, Josef Burger, Leonidas Galanis, Ameet Kini, Rajasekar Krishnamurthy, Mixed Mode XML Query Processing[C], VLDB 2003, pp. 225～236.

4 Schmidt A, Waas F, Kersten ML, Carey MJ, Mamolescu I, Busse R. XMark: A benchmark for XML data management. In: Bernstein PA, Loannidis YE, Ramakrishnan R, Papadias D, eds. Proceedings of the 28th International Conference on Very Large Data Bases. Hong Kong; Morgan Kaufmann Publishers. 2002. 974－985.

5 韩悦、蔡荣峰、岳丽华、龚育昌，一种高效的 XML 路径查询索引，计算机工程与科学，2005 年第 27 卷第 11 期.

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn