XML文档快速解析技术研究

陈 娟，李 晖，鱼 雷
（西安电子科技大学 计算机网络与信息安全教育部重点实验室，陕西 西安 710071）

摘 要：介绍了XML文档数据的快速解析理论及其解析的必要性，并在分析现有解析方式的优劣的基础上，着重研究了新的VTD格式的解析技术。这种方式能在性能和速度上大大提高XML的解析能力，它不仅解决了SAX和DOM的各种问题，还带来了非提取性能的其它好处，比如快速的解析与遍历、XPath的支持、增量更新(Incremental Update)等等。

关键词：XML；VTD；解析；非提取
中图分类号：TP312 **文献标识码**：A **文章编号**：1673－629X(2007)10－0040－03

Study of Technology for Parsing XML Document

CHEN Juan, LI Hui, YU Lei
(Ministry of Edu. Key Lab. of Computer Networks and Information Security, Xidian Univ., Xi'an 710071, China)

Abstract: Introduced the theory of fast parsing of XML document, and analyzed the advantages and disadvantages of two parsing methods available. On the basic of that, focused on a new parsing method－VTD. This method increases the speed and improves the performance of XML parsing. VTD doesn't only resolve several problems brought by SAX or DOM, but also contains other benefits of non－extractive, such as fast parsing and traversal, the support of XPath, incremental update. etc.

Key words: XML; VTD; parsing; non－extractive

0 引 言

XML(eXtensible Markup Language)——可扩展标记性语言，是W3C组织认可的文档数据格式标准，是SGML(Standard Generalized Markup Language，标准通用标记语言)的子集，它保留了SGML主要使用功能的同时大大缩减了SGML的复杂性。它独立于任何语言和体系结构，是公认的下一代网络标记语言。

由于XML文档能应用在不同平台上，实现数据的协同工作，它现在已成为不可缺少的企业技术的一部分。比如，XML增加了在商场中的电子商务和交流沟通，以及公司内部多样数据的综合。XML的使用因此而快速增长，分析家Ron Schmelzer预测，到2006年，XML将由2003年网络流量的3%上涨至24%，并且到2008年将至少上涨至40%[1]。

然而，对不断增长的XML文档的执行引起了一个关键的问题：由于文档中每个元素都包含了相当大的元数据，所以XML文档就包含了大量的数据。这便造成了程序处理的低效，并且给公司网络、处理器和存储结构都带来了很大的负载压力，这就导致了XML的两个关键问题：冗长和性能。

1)冗长：以XML格式储存的数据要比以其他数据库格式储存占有的空间要大得多。

2)性能：由于XML固有的冗余特性，对XML数据的管理将增大应用服务器的负担。

1 XML解析的必要性

XML解析器是XML应用的基础。XML本身只是以纯文本对数据进行编码的一种格式，要想利用XML，或者说利用XML文件中所编码的数据，必须先将数据从纯文本中解析出来，因此，要求必须有一个能够识别XML文档信息的文本文件阅读器(即XML解析器)，用来解析XML文档并提取其中的内容。显然，XML解析器在XML应用程序中有着重要的地位。XML解析器根据是否验证合法性，可分为验证性和非验证性解析器；而根据解析方式的不同，又可分为DOM解析器和SAX解析器等[2]。

2 XML解析器的工作原理

① 负载均衡器接收来自Internet的入站XML流量。

收稿日期：2006－12－27
作者简介：陈 娟(1982－)，女，陕西人，硕士研究生，研究方向为密码学；李 晖，教授，研究方向为网络安全。

② 负载均衡器直接将 XML 流量定向到 XML 解析器。

③ XML 解析器解压缩、解析、解释和确认 XML 流量,然后将它路由给适当的应用服务器。

④ XML 解析器对应从应用服务器接收到的 XML 流量进行转换、确认、加密和压缩。

⑤ XML 解析器将对 XML 数据送给负载均衡器,随后这些数据被送上 Internet。

如图 1 所示,XML 解析器把原来由应用服务器承担的许多大量耗费 CPU 时间的任务卸载到网络上,这使得企业获得线速性能的花费大大降低了。与通过扩展应用服务器来获得相同性能相比,利用 XML 解析器实现相同性能目标的花费仅为前者的 1/10。

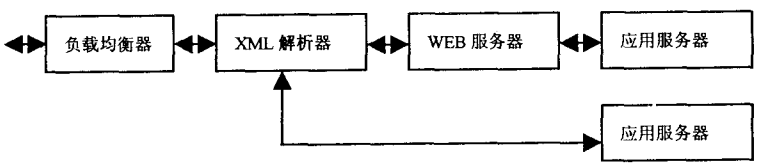

图 1　XML 解析器的工作模式

3　现行的两种 XML 解析方式:DOM 和 SAX

3.1　DOM

DOM 是以层次结构组织的节点或信息片断的集合。这个层次结构允许开发人员在树中寻找特定信息。分析该结构通常需要加载整个文档和构造层次结构,然后才能继续工作。由于它是基于信息层次的,因而 DOM 被认为是基于树或基于对象的。DOM 以及广义的基于树的处理具有几个优点:首先,由于树在内存中是持久的,因此可以修改它以便应用程序能对数据和结构做出更改;它还可以在任何时候在树中上下导航,而不是像 SAX 那样是一次性的处理。DOM 使用起来也要简单得多[3]。

另一方面,对于特别大的文档,解析和加载整个文档可能很慢且很耗资源,这也是 DOM 应用的局限性,因此使用其他手段来处理这样的数据会更好。

3.2　SAX

这种处理的优点非常类似于流媒体的优点。分析能够立即开始,而不是等待所有的数据被处理。而且,由于应用程序只是在读取数据时检查数据,因此不需要将数据存储在内存中。这对于大型文档来说是个巨大的优点。事实上,应用程序甚至不必解析整个文档;它可以在某个条件得到满足时停止解析。一般来说,SAX 还比它的替代者 DOM 快许多。但是 SAX 却由于没有结构信息,而具有无法遍历的缺点。

表 1 将这两种解析方式作了简单明了的对比。

表 1　两种解析方式的比较

优缺点 / 解析方式	优点	缺点
DOM	易用性强,遍历简单,支持 XPath	解析速度太慢,内存占用过高(原文件的 5x～10x),对于大文件来说几乎不可能使用
SAX	解析速度快,内存占用不与 XML 大小相联系(可以做到 XML 涨,内存不涨)	易用性差,因为没有结构信息,无法遍历,不支持 XPath,可维护性差

4　新的解析方式 VTD－XML

4.1　VTD—XML

VTD(Virtual Token Descriptor)——虚拟令牌描述符,是一种类似于 DOM 和 SAX 的处理模式,但是它具有这两种解析方式都不具备的优点。前面讲了,DOM 和 SAX,两者都需要构建一个复杂的内存中的数据结构,以及在这些模型中任意向前浏览或向后浏览的能力。它们还使用动态数据结构,这种结构会随着时间不断变化,并且截取任意大小的数据块有时候会非常大。而 VTD 处理技术就是在充分考虑了以上弊端的基础上产生的,并且发展迅速。

VTD－XML 是一种新型的、开源的、非提取性的 XML 处理方式。不同于现行的 XML 处理技术,VTD－XML 能够实现数据的随机存取,而不会导致过多的资源浪费。该格式的一个关键优化性能就在于它采用了非提取性的令牌环结构。VTD－XML 在内存中完整保存了未经编码的 XML 信息,令牌代表的是使用了专用起始偏移量和长度的令牌。该令牌是基于虚拟令牌描述符的二进制编码规范。一条 VTD 格式的信息大小是 64 比特的整数倍,其中包含有 XML 令牌长、起始偏移量、类型和嵌套深度的编码(如图 2 所示)。

为了实现非提取(non－extractive)这个目的,VTD 将原 XML 文件原封不动地以二进制的方式读进内存,连解码都不做,然后在这个字节(byte)数组上解析每个元素(element)的位置并把一些信息记录下来,之后的遍历操作便在这些保存下来的记录(record)上进行,如果需要提取 XML 内容就利用记录中的位置等信息在原始字节数组上进行解码并返回字符串[4]。

注意 VTD 是固定长度的(官方决定用 64bits),这样做的目的就是为了提高性能,因为长度固定,在读取、查询等操作的时候就具有显著的高效性,也就是可以用数组这种高效的结构来组织 VTD,这就大大减少了因为大量使用对象而产生的性能问题。VTD 的能力就在于它能够将 XML 这种树形的数据结构简单地变换成对一个字节数组的操作,任何需要的对字节数

组的操作都可以应用在 XML 上。这是因为读取进来的 XML 是二进制的(byte 数组),而 VTD 则记录了每个元素的位置等访问信息,当找到要操作的 VTD 的时候,只要用起始位置与长度等信息就可以对原始字节数组进行任何操作,或者可以直接对 VTD 进行操作。举例来说,要在一个大 XML 中找出一个元素并删除它,那么只需要找到这个元素的 VTD(遍历方法稍候再讲),将这个 VTD 从 VTD 数组中删除,然后再利用所有的 VTD 写出到另一个字节数组中就可以了,因为删除的 VTD 标明了要删除的元素的位置,所以在新写入的字节数组中就不会出现这段元素了,用 VTD 写入新的字节数组实际上就是一个字节数组的拷贝,其效率相当地高,这就是所谓的增量更新(incremental update)。

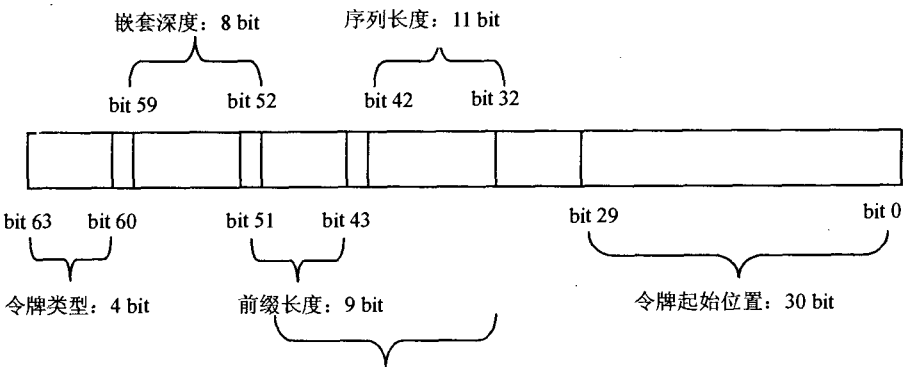

图 2 一条 VTD 记录的比特层格式

关于 VTD－XML 的遍历方式,它采用了 LC (Location Cache)结构,简单地说就是将 VTD 以其深度作为标准构建的一个树形的表结构。LC 的条目(entry)也是 64bits 长的数值类型,前 32bits 代表一个 VTD 的索引(index),后 32bits 代表了这个 VTD 的第一个子(child)的索引。利用这些信息就可以计算出任何想要到达的位置。基于这种遍历方式的 VTD－XML 有与 DOM 不同的操作接口,并且,VTD－XML 的这种遍历方式可以在最少的几步内查询到所需的数据,遍历的性能十分突出。

4.2 VTD—XML 的性能优点

首先,由于文档保存的完整性,使得应用程序能够精确地向一个字节队列的 XML 文档内容进行插入和移动操作;其次,该处理方式和加密/解密以及包数据的分割/组装操作类似,已经被广泛地认可并应用到硅芯片之中。

这里有一组数据,取自于 VTD－XML 的官方网站[5]:

VTD－XML 的解析速度是 SAX(with NULL content handler)的 1.5x～2.0x。With NULL content handler 的意思就是指 SAX 解析中没有插入任何额外的处理逻辑,即 SAX 的最高速度。

VTD－XML 的内存占用是原 XML 的 1.3x～1.5x(其中 1.0x 的部分是原 XML,0.3x～0.5x 是 VTD－XML 占用的部分),而 DOM 的内存占用则是原 XML 的 5x～10x。举一个例子,如果一个 XML 的大小是 50MB,那么用 VTD－XML 读取进来内存占用会在 65MB～75MB 之间,而 DOM 的内存占用则会在 250M～500MB 之间。基于这个数据用 DOM 处理大的 XML 文件几乎是不可能的。

5 分类处理 XML 数据方案

在对 XML 解析器进行深入的研究后,针对以上提出的 XML 数据处理存在的问题,提出一种分类处理 XML 数据方案,即可以根据网络上传输的不同数据流量,将其分发到不同的 XML 处理器中,这种方案既可以解决 DOM 对大文件解析速度慢的缺点,又可以克服 SAX 没有结构信息,无法遍历的问题,基本方案结构如图 3 所示。

① 在这个方案中,XML 解析模块将根据不同的 XML 数据,采用不同的解析方式(DOM,SAX,VTD)来进行解析。

② 管理模块对各个解析器进行统一管理,将不同的数据智能分发到不同的解析器中。

③ 维护模块可对网络设备和应用服务器进行统一的设备维护。

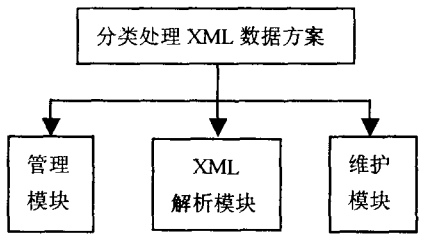

图 3 分类处理 XML 数据方案

6 结束语

随着 XML 更广泛的应用,它的速度和性能受到了越来越多的重视。介绍了 XML 解析的必要性和常用的解析方法,并分析了 SAX 和 DOM 这两种解析方式,在比较了它们优劣的基础上,介绍了新的解析方式

(下转第 49 页)

$->i+s(\mathrm{mod}\ N)$,任意两结点之间的最短路径。

4 总结和展望

大量的文献是通过数学方法以及与其等价的 L 形瓦来进行研究双环网络的,文中根据双环网络 $G(N;r,s)$ 的性质,采用生成树来研究紧优双环网络及其性质,并运用 C# 编程语言实现了双环网络 $G(N;r,s)$ 的最小生成树的算法,利用该程序,对任意给定的 N,可以计算双环网络 $G(N;r,s)$ 的紧优解,并研究其紧优点的分布特点和相应的最小生成树特性。

寻找紧优和几乎紧优是双环网络的研究重点,如何在双环网络的最小生成树上进一步研究紧优双环网的特性是今后值得研究的方向。

参考文献:

[1] 徐俊明.计算机互连双环网络的最优设计[J].中国科学:E辑,1999,29(3):272-278.

[2] Wong G K, Coppersmith D. A combinatorial problem related to multimodule memory organization[J]. J Assoc for Comput Mach,1974,21:392-401.

[3] Fiol M A, Yebra J L, Alegre I, et al. A discrete optimization problem in local networks and data alignment[J]. IEEE Trans Comput,1987,36:702-713.

[4] 李　乔,徐俊明,张忠良.最优双环网络的无限族[J].中国科学:A辑,1993,23(9):979-992.

[5] 徐俊明.不含紧优和几乎紧优双环网络无限族[J].科学通报,1999,44(5):486-490.

[6] 徐俊明,刘　琦.一类4紧优双环网无限族[J].中国科学:A辑,2003,33(1):71-74.

[7] 刘焕平,杨义先,胡铭曾.最优双环网的构造[J].系统工程理论与实践,2001,21(12):72-75.

[8] 刘焕平,杨义先,杨放春.双环网 G(N;S1,S2)的直径[J].系统工程理论与实践,1999,19(2):58-61.

[9] Chen C Y, Hwang F K. Equivalent nondegenerate L-shapes of double-loop networks[J]. Networks, 2000,36:118-125.

[10] Chen C Y, Hwang F K. Equivalent L-shapes of double-loop networks for the degenerate case[J]. Journal of Interconnection Networks, 2000,1:47-60.

[11] 方木云.双环网络 G(N;r,s)的L形瓦仿真算法[J].系统仿真学报,2005(4):914-916.

[12] 方木云,赵保华,屈玉贵.非单位步长双环网络 G(N;r,s)的L形瓦仿真算法[J].系统仿真学报,2006(10):2963-2965.

(上接第42页)

VTD解析,它的速度和性能都明显优于前两者。在此基础上,提出了一种分类处理 XML 数据的方案。

参考文献:

[1] Geer D. Will Binary XML Speed Network Traffic[J]. IEEE Computer Society,2005,38(4):16-18.

[2] Skonnard A,Gudgin M. XML 精要——快速参考手册[M].牛　韬,英　宇译.北京:人民邮电出版社,2002.

[3] Morrison M. XML 揭秘——入门、应用、精通[M].陆新年,陆新宇译.北京:清华大学出版社,2001.

[4] Zhang J. Better, Faster XML Processing with VTD-XML [EB/OL]. 2004. http://www.devx.com/xml/Article/22219.

[5] Zhang Jimmy. XML on a Chip[EB/OL]. 2005. http://www.xml.com/pub/a/2005/03/09/chip.html.

(上接第45页)

果质量有了较大的提高。通过对实验结果的分析,注意到启发式系统对结构依赖较大,当两个本体的结构相似时,规则的使用能较好地提高匹配质量。

4 结束语

介绍一个新的半自动的启发式本体匹配框架的设计和实现。已获得的实验数据证明系统是成功的。在今后的工作中,考虑:1)引进机器学习的方法来确定启发式规则的权值参数;2)提高系统的通用性和效率。

参考文献:

[1] Castano, Silvana, Antonellis, et al. Global viewing of heterogeneous data sources[J]. IEEE Transactions on Knowledge and Data Engineering,2001,13(2):277-297.

[2] Tang Jie, Liang Bangyong, Li Juanzi. Multiple Strategies Detection in Ontology Mapping[C]// In:Proceedings of the 14th International World Wide Web Conference (WWW'2005). Chiba,Japan:[s.n.],2005:992-993.

[3] Jian N, Hu W, Cheng G, et al. Falcon-AO: Aligning Ontologies with Falcon[C]// K-Cap 2005 Workshop on Integrating Ontologies. Banff, Canada:[s.n.], 2005.

[4] Avesani P, Giunchiglia F, Yatskevich M. A Large Scale Taxonomy Mapping Evaluation[C]// In: ISWC 2005, LNCS 3729. Berlin,Heidelberg:Springer-Verlag,2005:67-81.

[5] Fellbaum C. WordNet: An electronic lexical database[M]. Cambridge, MA:MIT Press,1998.