

XML 路由算法分类研究 *

郑祥毅 翟磊 陈继明 江曼 潘金贵

(南京大学计算机软件新技术国家重点实验室 南京大学计算机科学与技术系 南京 210093)

摘要 目前,由于网络上存在海量的 XML 信息,针对 XML 格式数据的过滤及信息的选择性分发成为当前网络通信研究的重点。如何利用大量 XPath 查询表达式来构造具有高效性和扩展性的 XML 过滤器,以匹配持续到来的 XML 文件流,是 XML 路由算法研究的核心。本文首先介绍 XML 路由模型,然后对 XML 路由算法进行分类比较,在此基础上对典型 XML 路由算法进行归纳总结,最后展望 XML 路由算法的应用前景。

关键词 XML 路由模型,XML 路由算法,XML 过滤器,匹配

Taxonomy Study of XML Routing Arithmetic

ZHENG Xiang-Yi ZHAI Lei CHEN Ji-Ming, JIANG Man PAN Jin-Gui

(State Key Lab For Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

Abstract Recently, XML has emerged as a standard information exchange mechanism on the Internet, and these lead to an increased interest in how to filter XML pattern data and process selective dissemination of information in the network communications field. The core technical challenge about these is processing a large collection of XPath queries on an incoming XML stream, and both filtering efficiency and system scalability are of paramount concern. In this paper, the model of XML routing will be introduced at first, and then, the XML routing arithmetic will be classified, and the characteristic of the different arithmetic will be summarized. Finally, the application of XML routing arithmetic is discussed.

Keywords XML routing model, XML routing arithmetic, XML filter, Matching

1 引言

由于互联网的迅猛发展,可获得的信息越来越丰富,如何在海量的信息中获得感兴趣的那部分内容,成为目前大家关注的焦点。以前,用户一般通过关键字查询来获取感兴趣的信息,但这种过滤技术因为不能有效地表达用户的意图和兴趣而多有不足。随着 XML^[1] (Extensible Markup Language) 成为网络信息交换的标准,越来越多的信息保存为 XML 格式,或以 XML 格式进行通信。针对这种情况,在 XML 的路由和选择性信息分发技术方面,需要寻找更有效的信息过滤技术和算法。目前在这方面已经做了大量的研究工作,比如 XFilter^[2], YFilter^[3,4], Index-Filter^[5], Bloom Filter^[6] 等,它们关注的共同点是面对持续到来的 XML 文件流,如何有效地处理大量的查询表达式(如 XPath^[7]),以进行匹配过滤。

XML 路由算法研究的核心是利用表达用户兴趣的大量 XPath 查询表达式构造出 XML 过滤器,从而对持续到来的 XML 文件流进行匹配。由于查询表达式和用户可能非常多,因此信息处理的高效性和扩展性就成为评价 XML 路由算法优劣的标准。目前有一些研究正在持续进行当中,也有一些 XML 路由算法已应用于股票、交通信息系统、即时娱乐信息速递等领域^[1]。本文主要对 XML 路由算法进行分类比较,在此基础上对典型 XML 路由算法进行归纳总结,以利于 XML 路由算法的进一步研究及应用。

2 XML 路由模型

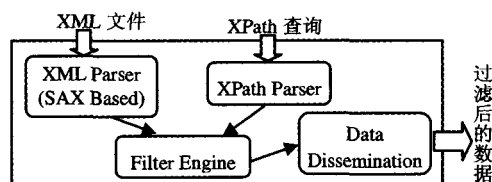


图 1 XML 路由模型架构图

设计 XML 路由算法一般是先对 XPath 表达式表示的大量 XML 路径查询进行转换,使之产生符合算法需要的模型结构,进而构造出合适的过滤器,以过滤由 XML 解析器解析的持续到来的 XML 文件流,最后把过滤后的信息发送给相应的用户。其中,由于表达用户兴趣的 XPath 查询表达式数量可能非常庞大,如何构造出具有高效性和扩展性的过滤器成为算法研究的重点^[8~15]。这里要介绍的 XML 路由算法都是构建在这样的 XML 路由模型之上,即给定一个 XML 输入文件 D 和一个路径查询集 $Q = \{q_1, \dots, q_n\}$,寻找 Q 的一个子集 $R = \{R_1, \dots, R_n\}$,其中 R_i 是 q_i 与 D 所能匹配的所有结果。XML 路由模型架构如图 1 所示,其中的核心模块是由 XPath 解析器(XPath Parser)构建而成的过滤器引擎(Filter Engine),它处理 XML 解析器(XML Parser)产生的事件,并把匹配后的结果通过数据分发模块(Data Dissemination)发送

* 本课题得到国家自然科学基金项目(60473113)、国家自然科学基金重点项目(60533080)资助。郑祥毅 翟磊 江曼 硕士研究生,主要研究方向为分布式虚拟环境与网络通信技术;陈继明 博士研究生,主要研究方向为分布式虚拟环境与 Agent 技术;潘金贵 教授,博士生导师,主要研究方向为中间件、Agent 技术、多媒体远程教育。

给相应的用户。

目前利用 XML 路由算法进行过滤时,一般使用基于事件的 SAX 解析器来解析 XML 文件流。因为这样可以不必占用大量内存,以存储需要过滤的 XML 文件,从而使系统更具扩展性。在查询语言方面,本文为了介绍方便,只使用 XPath,但在实际应用中可能使用 XQuery 等类 XPath 语言。

3 XML 路由算法

由于 XML 在信息交换中占据越来越重要的地位,并且在很多情况下,XML 数据的生产者和消费者在地理上是分散并通过网络互联的,因而如何对这些 XML 数据进行有效的过滤、分发、路由,成为当前研究的热点之一。目前不仅各大主要研究机构致力于 XML 路由算法的研究,一些厂商也纷纷进行 XML 路由算法的相关应用开发,并且当前已有 XML 路由器问世^[16]。在研究 XML 路由算法时,面临的共同问题是针对大量的 XPath 查询表达式,如何与持续到来的 XML 文件流进行匹配。

目前业界已经提出了许多有价值的算法,比如 XFilter, YFilter, Index-Filter, Bloom Filter 等。由于这些算法是为了解决 XML 路由的某一个或几个方面的问题而提出或加以改进,没有哪一种算法解决了 XML 路由所碰到的所有问题,因而它们有各自的特点及不同的适用场合。为了能在应用这些算法时更有针对性,也为了能更好地进行下一步的研究和改进,有必要对这些算法进行分类比较、归纳总结。由此,我们把这些算法分为 3 大类,分别是基于自动机的 XML 路由算法(Automation-based)^[2~4,17~22]、基于索引的 XML 路由算法(Index-based)^[5,23]和基于 Bloom Filter 的 XML 路由算法(Bloom Filter-based)^[6]。下面将分别介绍之。

3.1 基于自动机的 XML 路由算法

目前基于自动机的 XML 路由算法较多,且各有特点,但基于自动机导航是它们的共同点,即面对持续到来的 XML 文件流时,算法通过在 XPath 查询表达式所构建的自动机上导航来匹配解析得到的 XML 标签。当与自动机边上的符号匹配时,就进行相应的状态转换。该类算法在匹配 XML 文件时,同一时刻只处理一个标签。当基于事件的解析器(如 SAX 解析器)解析一个标签后,它就在相应的自动机上导航匹配。其中基于自动机的 XML 路由算法又可以根据所使用的自动机类型分为基于 NFA(比如 XFilter^[2]、YFilter^[3,4]、XTrie^[17])、基于 DFA(比如 LazyDFA^[18,19])和基于 PDA(比如 XPush Machine^[20,21])XML 路由算法 3 种。下面分别介绍这些具有代表性的基于自动机的 XML 路由算法。

XFilter 通过把路径查询表达式表示成多个有限状态机(FSM)的集合来构造过滤器,它是第一个基于 XML 的路由算法。在 XFilter 中,每个 XPath 查询表达式被分解成一个路径节点集,在路径节点集的基础上构建 FSM。其中路径节点就作为 FSM 的状态,然后通过一个查询索引(Query Index)把所有 FSM 状态连接起来。其中查询索引是由所有查询表达式去除了 * 及重复标签后的所有元素名称构建而成的一个哈希表。哈希表上的每个元素名称链接两个列表:候选列表(Candidate List)和等待列表(Wait List)。路径节点根据所属标签被添加到哈希表中相应元素名称所链接的候选列表或等待列表。进行过滤时,基于事件的解析器解析 XML 文件。遇到元素的开始标签时,就在查询索引中该元素的候选列表上检查所有的路径节点,根据每个路径节点上附带的层次信息和谓词信息分别进行层次检查和属性过滤器检查。如果二者都通过检查,并且当前正在检查的节点是当前查询表达式的最后一个路径节点,则匹配成功,否则就把查询表达式的下

一个节点从等待列表移到候选列表并准备下一步的匹配。可以使用列表平衡优化技术对 XFilter 进行优化,从而使算法性能提高数倍。这是因为原来建立的查询索引中的某些候选列表可能空出,可以经过调整使得每个候选列表都不空,从而达到平衡列表大小、提高匹配的效率和目的。

针对 XFilter 没有共享 XPath 查询表达式公共前缀而存在不足的问题,提出了一种改进算法 YFilter。YFilter 是一种典型的基于 NFA 的 XML 路由算法,它首先把多个 XPath 查询表达式表示成前缀树,然后把前缀树扩展成非确定性有限自动机(NFA)。YFilter 对查询表达式建立索引,把它构造成哈希树表示的 NFA,即为每个状态构建用于状态转换的哈希表。系统通过在 NFA 上导航进行匹配。执行 NFA 时,因为是非确定性自动机,可得到多个活跃状态,所以需要回溯。YFilter 使用一个运行时刻栈来实现此回溯。在条件相同的情况下,查询表达式数量比较大时,YFilter 的执行速度比 XFilter 要快数倍,这主要是因为实现了查询分组、共享前缀和建立了比 XFilter 更有效的索引结构。值得一提的是,YFilter 主要针对线性 XPath 表达式,对于复杂的嵌套 XPath 查询,会先将其转换成若干线性 XPath 查询的逻辑运算表达式。

对 XFilter 进行的另一种形式的改进是把每个查询表达式划分成只包含父子轴("/")的子串,并通过基于 Trie(Trie-based)数据结构来对所有的子串建立索引,从而提出了一种新的 XML 路由算法 XTrie。系统通过在子串构建的 XTrie 树上导航以进行匹配。在运行时,系统维持一个额外的数据结构用于跟踪子串之间的交互,因此公共子串的处理能够被查询所共享。XTrie 通过匹配 XPath 查询表达式的子串而不是单个元素来提高性能。与前两个算法相比,XTrie 主要特点体现在支持更复杂的 XPath 查询表达式过滤,同时支持有序和无序 XML 数据匹配和减少不必要的索引上。由于 XTrie 的存储空间由子串的数量决定,而子串的数量要小于元素的数量,因此空间复杂度要比 XFilter 好很多;同时,由于对子串的访问与对单个元素的访问相比,其次数要少得多,因此时间复杂度上也要比 XFilter 好很多。一般来说,在条件相同的情况下,XTrie 的运行效率要比 XFilter 高数倍。

上面提到的 3 种典型算法都是基于 NFA 的。而基于自动机的 XML 路由算法,除了基于 NFA 外,还有基于 DFA 和基于 PDA 两种类型。下面分别介绍这两种类型的 XML 路由算法。

基于 DFA 的 XML 路由算法针对 NFA 匹配产生回溯的问题,将 NFA 转化为 DFA。当匹配一个 XML 元素时,它无须转变到多个活跃状态,而是确定性地转变到一个新的状态,从而减小匹配的复杂性,但 NFA 的确定化也带来了时间和空间上的代价。目前已有 EagerDFA 和 LazyDFA 两种基于 DFA 算法^[18,19],但 EagerDFA 由于 XPath 表达式中子孙轴("/")的存在,会使得 DFA 的状态数量呈指数级增长,因此无法应用到实际中。相比之下,LazyDFA 的状态数量是可控的,它能够维持一个相对稳定的吞吐量,而与 XPath 查询表达式的数量无关,这使得算法在实际应用中具有良好的扩展性。这是因为增加 XPath 查询表达式时,DFA 是延迟构造的,很多不匹配的多余状态已经被预先排除,因此 DFA 状态变化不大;而且由于状态转换表是用哈希表实现的,其操作效率较高,因此增加 XPath 查询表达式时对 LazyDFA 的整体性能影响较小,而不像基于 NFA 的 XML 路由算法(比如 XFilter、YFilter、XTrie)增加 XPath 查询表达式时性能呈直线下降。

基于 PDA 路由算法 XPush Machine 增强了 NFA 自动机

的表达能力,这一扩展主要利用了支持查询表达式谓词之间 AND/OR 关系的 AFA(Alternating Finite Automation) 自动机,而 AFA 自动机利用扩展的状态来保存不同路径的执行情况。基于 NFA 和基于 DFA 的 XML 路由算法一般注重通过消减 XPath 查询表达式查询导航部分的公共子表达式来提高扩展性,却没有考虑消减 XPath 查询表达式中的公共谓词部分以提高扩展性。但当查询表达式中包含多个谓词时,谓词的处理效率就起决定作用,而这在 XML 消息系统中很

典型。基于 PDA 的 XML 路由算法既处理查询导航公共部分,也处理查询谓词公共部分,以消减二者各自的公共子表达式,使得在 XPath 查询表达式的数量和每个查询表达式的谓词数量上都具有扩展性。XPush Machine 算法首先把每个 XPath 查询表达式转化为 AFA,然后把所有的 AFAs 转化为一个单一的 XPush Machine。为了避免状态呈指数级增长,使用延迟构造方法,这在运行时需要相对较高的开销。但因为生成的状态可以重用,补偿了这种开销。

表 1 典型 XML 路由算法分类比较

分类	典型算法	主要特征	优化技术	效率比较	适用场合	
基于自动机	基于 NFA	XFilter	用 FSM 表示,单路径查询,没有共享多查询路径的公共前缀	平衡列表;预处理	第一个基于 XML 路由算法,是其他算法的基础	XPath 查询表达式数量比较少,差异比较大
		YFilter	典型的 NFA 算法,共享多查询路径的公共前缀	划分哈希表以支持并行计算	在 XFilter 基础上共享查询表达式的公共前缀;查询表达式比较多时,执行效率要比 XFilter 高数倍	用户的兴趣点比较集中,即 XPath 查询表达式公共前缀较多;查询表达式较多、XML 文件的层次较少
		XTrie	共享多查询路径中只包含父子关系的公共子串	优化为 Lazy XTrie,延迟探查子串表	对字符串的索引和查找要快于对元素的操作,同时跳过了多余匹配,因此执行效率要比 XFilter 高数倍	XPath 查询表达式的公共子串较多
	基于 DFA	Lazy-DFA	在 YFilter 基础上,把 NFA 转化为 DFA,减少自动机状态数	状态剪枝;聚合成 n-DFA;共享 NFA 状态表	过滤时比 YFilter 更有效,但构造自动机所需花费时间更多;当查询表达式数量增大时,受到影响较小,扩展性优于 NFA 算法	较规则的查询和没有深层嵌套的查询处理,以及查询中包含较少祖孙关系轴(/ /)
	基于 PDA	XPush Machine	既处理查询导航公共部分也处理谓词公共部分	自顶向下剪枝;次序优化;提前通知优化;训练自动机	考虑了其他算法没有考虑到的查询谓词公共部分;因此处理具有大量谓词的查询表达式时,要优于其他算法	XPath 查询表达式谓词包含 AND/OR 关系,具有较多公共部分的复杂谓词
基于索引	基于索引	Index-Filter	对要匹配的 XML 文件上的标签建立索引,然后在查询路径构造出的前缀树节点上附加这些索引信息以进行匹配	建立索引时使用 B-tree,减少内存需求;基于其他导航技术构造	当查询数量比较小,索引已实现的情况下,Index Filter 要比 YFilter 好。当查询数量增加时,情况相反。在实时建立索引的情况下,二者差别缩小。一般来说,查询处理数量比较少,XML 文件比较大的情况,Index Filter 更有效	XPath 查询表达式比较少,XML 文件比较大,或者静态 xml 文件比较多
基于 Bloom Filter	基于 Bloom Filter	Bloom Filter	将 XML 路径查询视为字符串,使用位向量组成的哈希函数高效地构造 Bloom Filter,由多个 Bloom Filter 生成路由表,结果会有误差	引入前缀过滤,减少候选路径数量	当查询表达式较多,XML 文件层次较小时,效率要高于其他算法;但其他条件不变,XML 文件的层次不断增加时,则 YFilter 更优	XPath 查询表达式较多,XML 文件层次较小;允许存在一定的概率误差

3.2 基于索引的 XML 路由算法

基于索引的 XML 路由算法是通过在 XML 文件的标签上建立索引来处理多路径查询的一类新颖的算法。通过充分利用 XML 文件上索引这种额外信息,该类算法能够避免处理 XML 文件中的一些特定标签,因为这些标签已经确认不会得到匹配。基于索引的 XML 路由算法为每个标签建立型为(L,R,D)的索引,其中 L 和 R 表示在先序遍历中节点的顺序号,D 表示在 XML 树中的层次。可以利用此索引信息确定树节点的相互关系(比如祖孙关系、父子关系等)。通过索引确认祖孙关系将和确认父子关系一样简单,这样就可以在没有中间节点信息的情况下获知祖孙关系。

PathStack 是一种基于索引的 XML 路由算法^[23],它在

XML 文件标签上引入索引,但只匹配单个查询表达式,作为多路径查询来说这不是很有效的办法。在 PathStack 基础上提出了一种改进算法 Index-Filter^[10],它把 PathStack 所使用的 XML 文件索引结构和 YFilter 所使用的可共享计算能力的前缀树结构结合起来,从而更有效地处理在 XML 文件上的多路径查询。

与 YFilter 算法相比,Index-Filter 扩展了前缀树结构。在执行算法前,前缀树中的每个节点都附加了 XML 文件中相应标签的索引信息。这样,XML 文件中的索引已经得到充分利用,在执行查询处理时,就只需要直接处理节点中附加的这些信息,而无须再处理 XML 文件。构建 Index-Filter 算法时,先对 XML 文件建立索引,然后把索引信息附加到 XPath

查询表达式生成的前缀树节点上,接着对前缀树进行处理。对于静态或成批到来的 XML 文件,索引建立后可以多次利用,从而提高了执行效率;对于动态到来的 XML 文件流,和其他算法一样,每个节点平均起来只被处理一次,执行效率也不低于其它算法。对于动态变化的前缀树,只需要为变化的那部分节点附加索引信息,没有变化的节点信息不变,所以平均起来,也只为前缀树的每个节点附加一次索引信息。对于 XML 文件流和前缀树都动态变化的情况,由于索引的使用可以避免匹配一些不必要匹配的节点,比如通过索引可获知有些节点不是当前节点的子孙,就可以不用匹配,这也同样提高了系统的整体性能。

一般来说,对于查询表达式数量比较少、XML 文件比较大的情况,已经预处理了 XML 文件,执行过滤时就可以直接使用索引信息,所以 Index-Filter 更有效;而查询表达式数量比较多、XML 文件比较小情况,因为 YFilter 用到的哈希表更具扩展性,所以 YFilter 更有效。

3.3 基于 Bloom Filter 的 XML 路由算法

基于 Bloom Filter 算法把每个 XML 路径查询表达式当作一个查询字符串,一个用户的所有查询字符串通过哈希函数映射到一个 Bloom Filter(具体表现为位向量),多个 Bloom Filter 组成一个路由表。进来的每个 XML 数据包,被解析产生一个候选路径集,每个候选路径通过使用相同的哈希函数映射得到一位向量,将该位向量与路由表中的位向量进行比较。如果与路由表中某用户的位向量匹配,则认为此 XML 数据包满足该用户的查询。

Bloom Filter 是一种高效存储的数据结构,它在上世纪 70 年代提出来,并在许多领域得到广泛应用。Bloom Filter 实际上是由位向量集组成的一个哈希表,在执行 XML 路由算法时,通过使用相同的哈希函数对 XML 文件进行映射而完成匹配工作。该匹配方法存在概率误差,称为假阳性(False Positive),即存在实际不匹配却被认为匹配的情况,但这个误差很小,在许多场合是可接受的。针对无法在 Bloom Filter 上删除元素的问题,提出计数 Bloom Filter(Counting Bloom Filter),即在每个位上维护一个计数器,从而达到灵活删除元素但不破坏 Bloom Filter 结构的目的。

为了解决因当前节点路径层次加深而导致的候选路径数量急剧增大的问题,提出了前缀过滤(Prefix Filters)技术,以减少候选路径的数量。即如果一个长度为 1 的候选路径与所有用户查询表达式长度为 1 的前缀字符串不匹配,那么所有以其开头的候选路径将不会与任何用户的查询表达式匹配。这样就可以直接过滤掉一些候选路径,从而达到减少冗余匹配的目的。

当查询表达式数量比较多、XML 文件层次比较小时,基于 Bloom Filter 的算法需要较少的路由表建立时间和过滤时间,因此提高了系统的性能。前面介绍的 YFilter 也比较适用于查询表达式数量比较多、XML 文件层次比较小的情况,但由于构造由位向量组成的哈希表比构造自动机要容易得多,而且在位向量上查询比在自动机上查询要高效得多,因此在容许一定误差的情况下,Bloom Filter 要更具扩展性。但伴随着 XML 文件层次的加深,Bloom Filter 的候选路径数量将迅速增大,因此算法的执行效率下降得比 YFilter 快,这时 YFilter 要更具扩展性。因为 Bloom Filter 算法过滤会产生一定的概率误差,因此不适用于需要精确过滤的场合。

3.4 典型算法的评价

基于自动机的 XML 路由算法把 XML 路径查询转化为自动机,持续到来的 XML 文件标签在自动机上导航,以匹配

查询。基于索引的 XML 路由算法把 XML 路径查询合并成一棵前缀树,并对持续到来的 XML 文件的每个元素建立索引,前缀树基于索引进行计算以匹配查询。基于 Bloom Filter 的 XML 路由算法将 XML 路径查询视为字符串,使用基于 Bloom Filter 的路由表来表示 XML 路径查询。一个用户的所有查询字符串通过哈希函数映射到一个 Bloom Filter,多个 Bloom Filters 组成一个路由表。

表 1 从主要特征、优化技术、效率比较和适用场合等几个方面对这些 XML 路由算法进行了系统的比较。

总结及展望 XML 格式数据固有的优点以及现实应用对于高效网络通信的需求使得 XML 成为网络数据交换的标准,针对 XML 格式数据的过滤、分发、路由也因此成为当前网络通信研究的重点。目前在诸如股票、球票、交通信息系统、个性化的电子新闻和娱乐信息速递等领域已有 XML 路由算法的实际应用。已有的研究以及在大规模信息分发系统已有的应用显示 XML 路由算法具有良好的应用前景,随着研究的深入以及技术的成熟,将有更多领域加入进来。

在分布式虚拟环境(DVE)^[24]领域,目前最受关注的是兴趣管理技术。兴趣管理本质上是一种信息过滤方法,它根据用户的兴趣来订阅分布式虚拟环境上发布的内容,这正好与上面讨论的 XML 路由技术处理模型一致。用户的订阅可以用 XPath 查询表达式表示,而分布式虚拟环境上发布的内容可以用 XML 文件表示。那么,对分布式虚拟环境进行兴趣管理的模型可以转化为 XML 路由模型,从而可以利用现成的一些 XML 路由算法和技术来提升兴趣管理的表达力和通用性,并且借助 XML 路由器较快的路由速度来提高兴趣管理性能。我们下一步研究的重点就是准备把 XML 路由技术引入到分布式虚拟环境,充分利用 XML 路由技术的优点,以提升分布式虚拟环境兴趣管理的性能。XML 的广泛应用以及 XML 路由技术的诸多优点显示,XML 路由技术在包括分布式虚拟环境的多个领域里具有良好的应用前景。

参考文献

- Extensible Markup Language (XML) 1.0 (Third Edition). W3C Recommendation 04, February 2004. <http://www.w3.org/TR/REC-XML/>
- altinel M, Franklin M J. Efficient Filtering of XML Documents for Selective Dissemination of Information. In: Proceedings of VLDB, 2000. 53~64
- Diao Yanlei, Zhang Hao, Fanklin M J. NFA-based Filtering for Efficient and Scalable XML Routing. [Technical Report]. Computer Science Division, University of California, Berkeley, 2002
- Diao Yanlei, Fischer P, Fanklin M J, et al. Yfilter: Efficient and Scalable Filtering of XML Documents. In: Proceedings of ICDE, 2002. 341
- Bruno N, Gravano L, Koudas N, et al. Navigation-vs. Index-based XML Multi-Query Processing. In: Proceedings of the 19th International Conference on Data Engineering (ICED'03), IEEE Computer Society, 2003. 139~150
- Gong Xueqing, Qian Weining, Yan Ying, et al. Bloom Filter-based XML Packets Filtering for Millions of Path Queries. In: 21st International Conf. on Data Engineering (ICDE'05), 2005
- XML Path Language (XPath) Version 1.0. W3C Recommendation 16 November 1999. <http://www.w3.org/TR/XPath>
- Felber P, Chan C-Y, Garofalakis M, et al. Scalable Filtering of XML Data for Web Services. IEEE Computer Society 1-2, 2003
- Yuriyama M, Nakamura H. Filtering Contents by Efficient Evaluation of XPath Expressions. In: Proceedings of the 2003 Symposium on Applications and the Internet (SAINT'03), IEEE Computer Society, 2003
- Murata M, Tozawa A, Kudo M. XML Access Control Using Static Analysis. In: CCS'03, Washington, DC, USA, 2003
- Gupta A K, Halevy A Y, Suciu D. View Selection for Stream Processing. In: WebDB Workshop. In: association with the International Conference on Management of Data, Madison, Wisconsin 2002
- Gupta A K, Suciu D, Halevy A Y. The View Selection Problem for XML Content Based Routing. PODS 2003, San Diego, CA, 2003
- Al-Khalifa S, Jagadish H V, Koudas N, et al. Structural Joins: A Primitive for Efficient XML Query Pattern Matching. In: Proceed-

- ings of the 2002 International Conference on Data Engineering, 2002
- 14 Ives Z, Levy A, Weld D. Efficient Evaluation of Regular Path Expressions on Streaming XML Data; [Technical report]. University of Washington, 2000
 - 15 Lakshmanan L V S, Parthasarathy S. On Efficient Matching of Streaming XML Documents and Queries. In: Proceedings of ED-BT, 2002
 - 16 Sarvega's XML Content Router. Sarvega, INC. 2005, <http://www.sarvega.com/xml-routing.html>
 - 17 Chan Chee-Yong, Felber P, Garofalakis M, et al. Efficient Filtering of XML Documents with XPath Expressions. In: Proceedings of the 18th International Conference on Data Engineering (ICDE'02), 2002
 - 18 Green T J, Miklau Miklau, Onizuka M, et al. Processing XML Streams with Deterministic Automata. In: Processing of ICDT, 2003. 173~189

- 19 Onizuka M. Light-weight XPath Processing of XML Stream with Deterministic Automata. In: CIKM'03, New Orleans, Louisiana, USA, November 2003
- 20 Gupta A K, D Gupta. Stream Processing of XPath Queries with Predicates. In: SIGMOD 2003, San Diego, CA, June 2003
- 21 Qlteanu D, Kiesling T, Bry F. An Evaluation of Regular Path Expressions with Qualifiers against XML Streams. In: Proceedings of the 19th International Conference on Data Engineering (ICED'03), IEEE Computer Society, 2003
- 22 高军, 杨冬青, 唐世渭, 等. 基于树自动机的 XPath 在 XML 数据流上的高效执行. 软件学报, 2005, 16(2): 223~232
- 23 Bruno N, Koudas N, Srivastava D. Holistic Twig Joins: Optimal XML Pattern Matching. In: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, 2002
- 24 潘志庚, 姜晓红, 等. 分布式虚拟环境综述. 软件学报, 2000, 11(4): 461~467

(上接第 88 页)

的随机变量, 则叠加大量的 ON/OFF 源就可以导致网络流量呈现出自相似的特性。

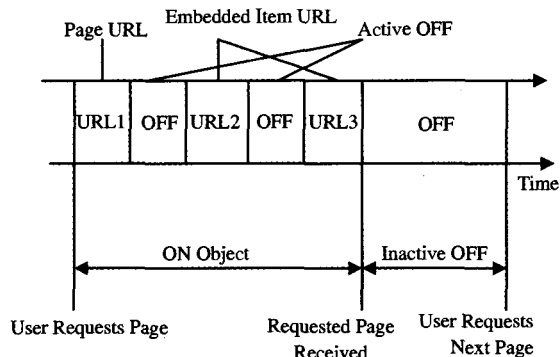


图 4 SURGE 中使用的 ON/OFF 模型

SURGE 就是使用大量的 ON/OFF 源叠加, 生产具有自相似特征的 HTTP 请求序列。图 4^[4] 是 SURGE 中使用的 ON/OFF 模型。

4 现有 Web 服务器测试系统存在的问题

以 WebStone、SPECWeb 和 SURGE 为代表的一类方法, 基本上是在前一个请求完成之后才发起新的请求, 这就使 Client 和 SUT (System Under Test, SUT 被测系统) 步伐一致, 所有 Client 的连接速率之和等于 SUT 的吞吐量。当 Client 的增加已使 SUT 满负荷时, 再增加 Client 只能使 SUT 的接收队列变长而增加排队延迟, 进而降低了 Client 的请求速率。所以, 生成的请求最多能达到 SUT 的承受能力, 但不会使 SUT 过载。

S-Client 可以产生过量的负载, 但其功能过于简单, 且每次都请求相同的静态文档。该文档会被缓存到内存中, 访问可能省去磁盘 I/O 时间, 与 Web 服务器的典型行为不符。

总结 本文分析了 Web 服务器的性能测试原理, 研究并阐述了网络流量的自相似特征。这些特征是测试工具要考虑的。对 Web 服务器性能进行评价, 可以发现存在的问题, 以便进行分析和改进。所以, 掌握 Web 服务器的性能测试方法很有必要。

本文所论述的 Web 服务器测试工具都有各自的优点, 但也存在一定的局限性。今后的主要工作可以放在具有请求序列高负载、自相似, 且服务器上文件的分布符合实际情况的 Web 服务器测试工具研究上, 这样才能够测量出 Web 服务器在真实环境下的性能。

参考文献

- 1 Leland W E, Taqqu M S, Willinger W, et al. On the Self-Similar Nature of Ethernet Traffic. IEEE/ACM Transactions on Networking, 1994, 2(1): 1~15
- 2 Crovella M E, Bestavros A. Self-similarity in word wide web traffic: Evidence and possible cause. In: Proceedings of the 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, 1996
- 3 <http://www.mindcraft.com/webstone/>
- 4 Barford P, Crovella M. Generating Representative Web Workloads for Network and Server Performance Evaluation. In: Proceedings of Performance '98/ACM SIGMETRICS '98, 1998
- 5 Hu Yiming, Nanda A, Yang Qing. Measurement, Analysis and Performance Improvement of the Apache Web Server. In: 18th IEEE International Performance Computing and Communications Conference, Phoenix, 1999
- 6 Standard Performance Evaluation Corporation. <http://www.spec.org/osg/web99/>
- 7 欧灿辉, 李晓明. Web 服务器性能评测. 计算机研究与发展, 2002, 39(5): 540~547
- 8 Banga G, Druschel P. Measuring the Capacity of a Web Server. In: Proceedings of the USENIX Symposium on Internet Technologies and Systems, California, 1997
- 9 Padmanabhan, V N, Qui Lili. The content and access dynamics of a busy Web site: Findings and implications. In: ACM SIGCOMM Symposium on Communications Architectures and Protocols, 2000. 111~123
- 10 Stallings W. High-Speed Networks TCP/IP and ATM Design Principles. Englewood Cliffs, NJ: Prentice Hall, 1998
- 11 Willinger W, Paxson V, Taqqu M S. Self-Similarity and Heavy Tails: Structural Modeling of Network Traffic [M]. In: A Practical Guide to Heavy Tails: Statistical Techniques and Applications, Birkhauser, Boston; Chapman & Hall, 1998. 27~53
- 12 Paxson V, Floyd S. Wide area traffic: the failure of Poisson modeling. IEEE/ACM Transactions on Networking (TON), 1995; 226~244
- 13 Trent G, Sake M. WebSTONE: The First Generation in HTTP Server Benchmarking. <http://www.mindcraft.com/webstone/paper.html>, 1995
- 14 Nahum E. Deconstructing SPECWeb99. In: Proceedings of the 7th International Workshop on Web Content Caching and Distribution, 2002
- 15 Aron M, Druschel P. TCP implementation enhancements for improving Web server performance: [Technical Report TR99-335]. Computer Science Dept, Rice University, 1999
- 16 Amza C, Cox A, Zwaenepoel W. Scaling and availability for dynamic content web sites: [Technical Report TR02-395]. Rice University, 2002
- 17 Challenger J, Iyengar A, Witting K et al. A publishing system for efficiently creating dynamic Web content. In: Proceedings of the Conference on Compute Communications (IEEE Infocom), Tel-Aviv, Israel, March 2000
- 18 Iyengar A, Challenger J. Improving Web server performance by caching dynamic data. In: Proceedings of the USENIX Symposium on Internet Technologies and Systems, Monterey, CA, 1997
- 19 Arlitt, M F, Williamson C L. Internet Web servers: Workload characterization and performance implications. IEEE/ACM Transactions on Networking, 1997, 5(5): 631~646
- 20 Barford P, Crovella M. A performance evaluation of hyper text transfer protocols. In: Proceedings of the ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems, Atlanta, GA, 1999
- 21 Banga G, Druschel P. Measuring the capacity of a web server under realistic loads. World Wide Web Journal (Special Issue on World Wide Web Characterization and Performance Evaluation), 1999
- 22 Banga G, Mogul J C. Scalable kernel performance for internet servers under realistic loads. In: USENIX Annu Tech Conf., New Orleans, LA, 1998