基于规则模板的正则表达式分组算法

邵翔宇,刘勤让,谭力波
(国家数字交换系统工程技术研究中心,河南郑州 450002)

摘 要: 采用规则分组的方法解决确定型有限自动机(Deterministic Finite Automata,DFA)状态爆炸问题,随着分组数目的增加,匹配效率大大降低.本文提出正则表达式的输入驱动特性理论,并基于此提出了基于规则模板的分组算法——模板有限自动机.模板有限自动机算法基于规则模板对规则集进行分组,各分组分别构建匹配引擎.理论分析和实验表明,与典型的 DFA 改进算法相比,预处理时间和存储空间有 2 ~ 3 个数量级别的缩减,且匹配效率没有明显降低.

关键词: 正则表达式;确定型有限自动机;分组自动机;扩展有限自动机;多维有限自动机;规则模板
中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2016)01-0236-05
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2016.01.036

A Regular Expression Grouping Algorithm Based on Signature Templates

SHAO Xiang-yu, LIU Qin-rang, TAN Li-bo
(*National Digital Switching System Engineering & Technological R&D Center, Zhengzhou, Henan* 450002, *China*)

Abstract: As the group number grows, the classical signature grouping algorithm solves the DFA state explosion problem with a big decrease on matching efficiency. This paper presents a regular expression(Regex) input drive theory. According to such theory, a grouping algorithm based on signature templates, templates based finite automata(TFA), is proposed. TFA divides Regex set based on signature templates and constructs matching engines in each set. Experiment results show that the preprocessing time and storage are reduced by 2 ~ 3 orders of magnitude compared with classical DFA improved algorithms, and TFA brings no obvious decrease on matching efficiency.

Key words: regular expression; deterministic finite automata; multiple DFAs; extended finite automata; multi-dimensional finite automata; signature templates

1 引言

在网络信息安全领域,入侵检测系统(Intrusion Detection Systems,IDS)扮演着重要的角色,它采用深度包检测(Deep Packet Inspection,DPI)方法进行病毒检测、入侵识别等.随着攻击模式的多样化,最早的基于精确字符串匹配方式已经无法满足要求,正则表达式以其强大的、灵活的表达能力而得到广泛应用.但随着网络带宽逐年增加、规则数目的快速增长以及正则表达式表达功能的强大,DPI 应用中的正则表达式匹配(Regular Expression Matching,REM)面临严峻挑战,其中最为急迫的是如何满足高速网络数据包处理的要求.有限自动机分为非确定性有限自动机(Non-deterministic Finite Automata,NFA)和确定性有限自动机(Deterministic Finite Automata,DFA)两种,其中 DFA 与 NFA 相比在任意时刻只有一个活跃状态、处理一个字符只需要一次迁移,具有线速的匹配性能,适用于高速数据链路中的 REM.

将多条正则表达式编译成一个 DFA 时,由于各条规则之间的相互重叠和影响,会导致状态爆炸问题.YU 等人[1]分析了这种爆炸问题,提出对正则表达式规则进行分组的 mDFA(Multiple DFAs)算法.mDFA 是基于规则之间的膨胀情况而进行的分组,各分组采用 DFA 进行匹配,这种分组无法进行彻底的存储压缩,且会造

收稿日期:2014-07-31;修回日期:2015-01-31;责任编辑:李勇锋
基金项目:国家 973 重点基础研究发展计划(No.2013CB329104)

成分割成的组数目过多,降低分组算法匹配效率.

本文提出了正则表达式匹配引擎的输入驱动特性理论,针对当前单一算法无法针对整个规则集进行有效的存储压缩问题,提出了基于规则模板的正则表达式分组算法—模板自动机(Templates based Finite Automata,TFA).TFA 算法将规则模板作为正则表达式分组的依据,各分组采用的特定改进算法进行匹配.

2　正则表达式输入驱动特性

2.1　状态爆炸问题

正则表达式规则集在编译成一个 DFA 时,产生状态爆炸的规则主要有两类:计数器型爆炸和克林闭包型爆炸,这两种爆炸类型严重制约着 DFA 在 IDS 中的应用.

在 IDS 中 DFA 的构造[2]中,状态爆炸通常在 NFA 子集构造时产生,按照构造角度来分,解决 DFA 状态爆炸问题方案有以下 4 个方向:分组算法[1,3,4],不确定度调节算法[5,6],状态转移表(State Transition Table,STT)压缩算法[7~9],自动机结构改进算法[10~12].以上四类算法可以很大程度压缩存储空间,但无法针对整个规则集进行有效的存储压缩.在自动机结构改进算法中,针对这两类爆炸类型各自有比较有效的压缩算法:针对计数器型爆炸改进的扩展有限自动机(eXtended Finite Automata,XFA)算法[10],针对克林闭包型爆炸改进的多维立方体自动机(Multi-dimensional Finite Automata,MFA)算法[11].

XFA 算法通过增加计数器型标记避免计数器部分状态的重复描述,达到对计数器部分状态的对数级别缩减.XFA 存在一定的规则受限性,对其他规则类型的爆炸类型的压缩效果不明显.

MFA 算法解决克林闭包型规则联合编译造成的爆炸问题.MFA 通过在多维空间构造联合 DFA,利用联合状态转移图的对称性达到压缩存储空间的目的.MFA 算法构建的多维模型针对克林闭包型爆炸进行冗余状态压缩,但存在一定的规则受限性,无法解决计数器等类型的爆炸问题.

2.2　正则表达式输入驱动特性

规则和文本是正则表达式匹配引擎的两个输入,采用有限自动机实现的正则表达式匹配引擎的存储空间和带宽受输入的影响,如图 1 所示.NFA 接收一个字符时带宽主要决定于激活状态集合内的状态数目,而激活的状态数目与字符所达到的状态位置有关,其匹配时间最差为 $O(n^2)$,DFA 在任何文本输入下处理一个字符的时间都为常数 $O(1)$,因此 NFA 的匹配时间受输入文本影响.NFA 状态数目与规则长度 n 呈线性关系,DFA 不同类型的规则编译产生的状态数是不同的,因此 DFA 状态数目受输入规则影响.

基于以上所述 NFA 和 DFA 的特性,本文给出正则表达式匹配引擎输入驱动特性的定义.正则表达式匹配引擎的带宽(匹配时间)受文本驱动,造成处理一个字符的时间变化,则称该正则表达式匹配引擎为文本驱动(text directed)引擎.正则表达式匹配引擎的空间复杂度(状态数目和存储空间)受规则类型影响,而产生不同的状态数目,则称该正则表达式匹配引擎为规则驱动(signature directed)引擎.

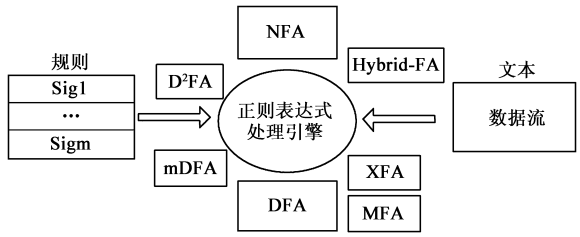

图1　正则匹配引擎的输入

表 1 列出了当前几类代表性算法的输入驱动特性.自动机结构改进算法中 XFA 执行的操作符受输入文本影响,造成不同输入文本的匹配时间有所不同;MFA 所处状态的受输入文本影响,造成匹配时间有所不同,但是两者最差匹配时间复杂度为常数级别,可以认为属于弱文本驱动.

单一的算法是无法针对所有规则同时消除其规则驱动和文本驱动特性,无法同时达到存储空间和匹配时间的下界.基于匹配引擎的驱动特性,有两种提高自动机匹配效率的改进思路:(1)输入文本控制:通常输入文本是需要待匹配的数据流,不受匹配引擎的控制,但是可以在文本进入匹配引擎之前进行文本预筛选,从而减小输入文本对匹配时间的影响;(2)输入规则控制:通常输入规则用于匹配病毒或恶意代码,输入规则的减少会严重影响引擎匹配的准确性,可以运行多个匹配引擎,保证规则的完整性.

3　基于规则模板的分组算法

分组算法是对输入规则控制的一种算法.通常为了尽可能最大程度隔离规则,需要较多的分组和较多的 DFA 引擎来完成对全部规则的覆盖,这也不可避免降低匹配引擎的匹配速度.本文对分组算法中 DFA 引擎进行改进,采用当前算法中 DFA 改进算法进行匹配引擎构造,改变分组算法的分组依据、降低分组数目,在保持分组算法规则高覆盖率的同时,提高分组算法匹配效率.由于 XFA、MFA 在各自适用的规则内能消除

规则驱动特性,且不存在严重的文本驱动特性,将 XFA 和 MFA 两种匹配算法同时应用在分组算法的匹配引擎中,达到更好的存储压缩效果.

表 1 当前算法的输入驱动特性

	DFA	NFA	mDFA	Hybrid-FA	D^2FA	XFA(Counter)	MFA
存储空间	规则驱动	- - -	规则驱动	规则驱动	规则驱动	在适用规则内不受规则驱动	在适用规则内不受规则驱动
匹配时间	- - -	文本驱动	- - -	文本驱动	文本驱动	弱文本驱动	弱文本驱动

3.1 基于规则模板的分组算法

针对 DFA 中最严重两种类型的爆炸问题,本文通过设定模板(template),按照 XFA、MFA 的适用规则特点进行规则分组,然后将各分组规则按照 XFA、MFA、DFA 构造自动机,形成模板自动机 TFA 匹配引擎. 图 2 给出了 TFA 算法的模型,其中包含:规则分组、匹配引擎构建、匹配过程.

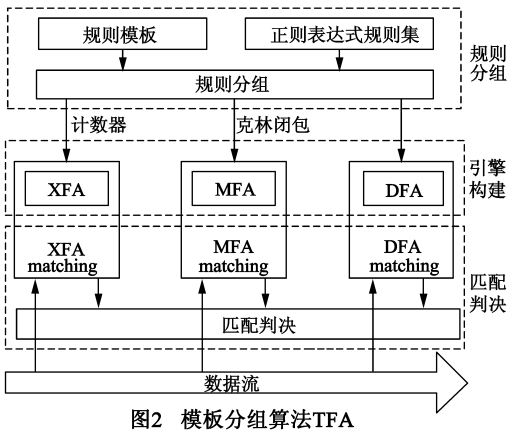

图2 模板分组算法TFA

规则分组

TFA 规则分组就是根据规则模板对规则集进行文本处理,从规则集中查找出符合规则模板的各条规则,形成三个规则子集的过程. 规则模板是一组正则表达式规则,用于查找包含特定类型的规则. 由于引擎中 XFA 和 MFA 算法的特点,用于规则分组模板的设定将直接影响 TFA 算法的效率,TFA 的规则模板设定为计数器型模板和克林闭包型模板.

通过计数器模板筛选出的都是包含计数器的正则表达式规则,可以通过 XFA 实现联合编译;通过克林闭包模板筛选出的包含克林闭包型规则,可以采用 MFA 算法进行联合编译;其他不膨胀规则可以简单采用 DFA 来实现.

引擎构造

将三组预处理规则分别按照 XFA、MFA 和联合 DFA 构造自动机,形成三个正则表达式匹配引擎. 其中 XFA 的构造方法,按照文献[10]中的方案. 而 MFA 的构造方法是:以克林闭包规则的一个 DFA(single-DFA)为一个维度,以每个规则的“. *”对应的状态为交点,构造一个在多维空间的跳转结构. DFA 构造方法参照基本流程构建联合 DFA.

匹配过程

待匹配数据送入系统之后,匹配判决会将相同的数据送入三个匹配引擎中,三个匹配引擎的进行相对独立的匹配过程. 由于各个引擎匹配结果产生的时间会有所不同,需要匹配判决模块根据系统的实际要求进行判决.

3.2 性能分析

衡量正则表达式算法性能主要有三个方面:预处理时间、存储空间和匹配时间. TFA 预处理时间即规则分组,是用正则表达式处理引擎规则文本的处理过程,处理时间较短. 因此,本文从 TFA 存储(状态)空间和 TFA 匹配时间来分析 TFA 算法复杂度.

存储空间(状态)复杂度

自动机的储存主要用于记录状态及其跳转信息,存储占用与状态数目直接相关,存储压缩最有效的方式是减少状态数目.

在包含 l 条计数器的规则中,XFA 针对计数部分进行的改进,其他部分与 DFA 相同. XFA 需要辅助的计数器空间,计数器重复次数为 x,则占用 $\log_2 x$,因此存储空间复杂度 S_{XFA_S1}

$$S_{XFA\_S1} = O(lr) + \log_2 x = O(lr) \tag{1}$$

在包含 m 条克林闭包的规则中,MFA 需要存储 m 个 DFA、m 条坐标轴信息和 2 个动态状态,因此 MFA 存储空间 S_{MFA_S2}

$$S_{MFA\_S2} = 2mr + 2 = O(2mr) \tag{2}$$

$S3$ 中 n 条 DFA 之间不会产生爆炸,因此 S_{DFA_S3} 与 Q_{DFA_S3} 一致,因此 TFA 存储空间复杂度 S_{TFA} 为 S_{XFA_S1}、S_{MFA_S2}、S_{DFA_S3} 之和

$$\begin{aligned} S_{TFA} &= S_{XFA\_S1} + S_{TFA\_S2} + S_{TFA\_S3} \\ &= O(lr) + O(2mr) + O(nr) \\ &= O(lr + 2mr + nr) \end{aligned} \tag{3}$$

因为计数器型改写后的 DFA 不产生爆炸,采用计数器对限定部分描述会造成空间节省,TFA 算法存储空间与规则数呈线性关系.

匹配时间复杂度

XFA 的匹配时间取决于当前状态的标志位操作,

XFA 中赋值、复位和计数的处理时间为常数,因此 XFA 匹配时间复杂度为常数级别 $O(1)$. MFA 的匹配时间长短取决于当前状态所处位置,交点与非交点处理时间不一致,但交点最差处理时间为常数,因此 MFA 匹配时间复杂度为 $O(1)$. DFA 匹配时间复杂度为常数 $O(1)$. 而 TFA 的匹配时间三者的匹配时间之和,因此其最差匹配时间复杂度应为常数级别 $O(1)+O(1)+O(1)$.

表 2 列出了 TFA 算法的时空复杂度,并将 DFA 和 NFA 作为对比. 由表可见,TFA 算法与 NFA 的存储空间复杂度类似,属于非规则驱动引擎;同时与 DFA 匹配时间复杂度一致,与输入有较弱的相关性,因此,TFA 算法在消除规则驱动特性同时并未引起严重的文本驱动特性,可以达到较好存储压缩效果.

表 2　TFA 算法的时空复杂度

	存储空间复杂度	匹配时间复杂度
TFA	$O(lr+2mr+nr)$	$O(1)+O(1)+O(1)$
DFA	- - -	$O(1)$
NFA	$O[l(r+x)+mr+nr]$	- - -

4　实验结果

仿真实验主要分为两部分:实验一主要是与 mDFA 算法对比,验证 TFA 分组算法的分组有效性;实验二是与经典改进算法对比,验证 TFA 算法的效率. 本文提出的规则模板分组算法是一种朴素的分组算法,参照实际 IDS 中各类规则比例,从实际 IDS 中选取 40 条规则作为测试规则集,包含 20 条精确型规则、10 条克林闭包型规则和 10 条计数器型规则.

本文采用 Becchi 提供的软件 RegEx Processor[13],分别实现了 NFA、DFA、mDFA、Hybrid-FA、D^2FA 以及本文的 TFA 算法. 利用 RegEx Processor 自带的 tracegen 工具生成 1GB 的测试数据,用于匹配时间的测试.

实验一　TFA 与 mDFA 算法对比

对比 TFA 与 mDFA 两种分组算法. 通过更改阈值设定不同的 mDFA 算法分组数目. 如表 3 所示,mDFA 算法的预处理时间和状态数目随着分组数目的增加而减少;但随着分组数目的增加,匹配时间与分组数目基本呈线性关系. 如图 3 所示的算法匹配时间和状态数目二维图上,TFA 相比于 mDFA 更加靠近坐标轴原点,说明 TFA 可以在不降低匹配效率的情况下缩减状态数目,可以避免 mDFA 分组数目过多造成的效率下降问题,TFA 分组算法比 mDFA 更有优势.

实验二　TFA 与经典算法对比

对比的算法包含 mDFA($m=6$)、Hybrid-FA、D^2FA 以及 NFA 和 DFA. 如表 4 所示,在预处理时间方面,TFA 与 mDFA 时间相当,比 Hybrid-FA、D^2FA 算法有几个数量级的缩减,实时更新能力最强;存储空间方面,TFA 比 Hybrid-FA、D^2FA 算法降低了 2 ~ 3 个数量级;匹配时间较基于 DFA 的算法 DFA、Hybrid-FA 无明显增加. 如图 4 所示的算法匹配时间和状态数目二维图上,相比于其他 DFA 改进算法 TFA 达到了存储空间和匹配速率更好的折中,算法匹配效率更好.

表 3　TFA 与 mDFA 对比

	分组数 m	预处理(分组)时间	状态数	mDFA 阈值	匹配时间(s/GB)
mDFA	2	30.1s	13368	10000	36.1
	3	10.3s	5246	2000	68.5
	4	9.8s	2073	6000	100.2
	5	9.6s	1520	500	126.5
	6	9.4s	1294	300	160.8
TFA	3	3.6s	861	NA	48.6

表 4　TFA 与经典算法对比

	NFA	DFA	mDFA	Hybrid-FA	D^2FA	TFA
预处理时间	<1s	285min	9.4s	410s	2311min	3.6s
存储空间	95K	285M	207K	21.3M	12.8M	186K
匹配时间(s/GB)	678.3	10.9	160.8	15.5	85.6	48.6

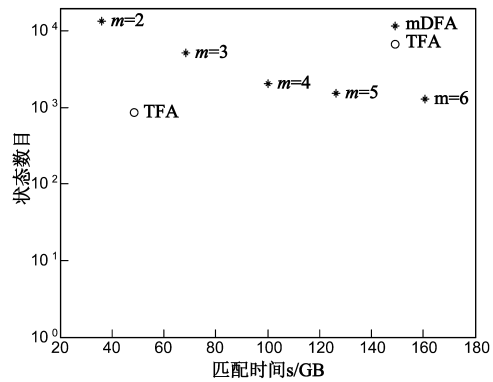

图3　TFA与mDFA算法对比

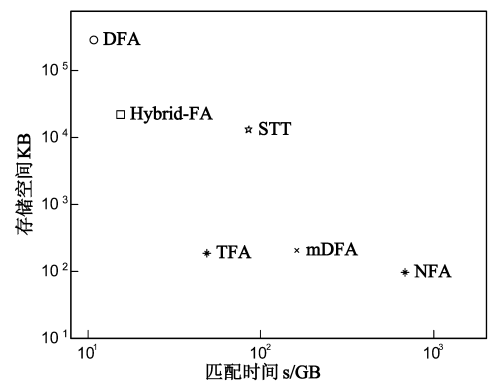

图4　TFA与经典算法对比

5　结束语

本文通过对算法的驱动特性进行分析,提出了基

于规则模板的分组算法 TFA,TFA 算法基于规则模板进行分组,各分组的实现采用 DFA 及改进算法. 理论分析和实验结果表明,TFA 能在不降低匹配效率的情况下,达到 mDFA 的空间压缩效果,同时避免了分组数目过多造成的效率下降问题. 与经典的改进算法相比,TFA 预处理时间和存储空间比 Hybrid-FA、D^2FA 算法降低了 2 ~ 3 个数量级;匹配时间与基于 DFA 的算法 DFA、Hybrid-FA 数量级相当.

本文提出的分组方法是一种基于 DFA 的朴素正则表达式匹配算法,而对于既是计数型也是闭包型的规则,基于 DFA 的分组方案无法进行有效的状态数目压缩. 因此,对 TFA 内部算法引擎的优化,对所有规则类型进行更加有效的存储压缩,是下一步的工作方向.

参考文献

[1] Yu F,Chen Z,Diao Y,et al. Fast and memory-efficient regular expression matching for deep packet inspection[A]. Proc of the IEEE/ACM Symp on Architectures for Networking and Communications Systems[C]. IEEE, 2006. 93 – 102.

[2] Hopcroft J E. Introduction to Automata Theory,Languages, and Computation[M]. Pearson Addison Wesley,2007.

[3] 徐乾,鄂跃鹏,葛敬国. 深度包检测中一种高效的正则表达式压缩算法[J]. 软件学报,2009,20(8):2214 – 2226.
Xu Q,E Y,Ge J. Efficient regular expression compression algorithm for deep packet inspection[J]. Journal of Software,2009,20(8):2214 – 2226. (in Chinese)

[4] 乔登科,王卿,柳厅文,等. 基于状态分组的高效 i – DFA 构造技术[J]. 通信学报,2013,34(8):102 – 109.
Qiao D,Wang Q,Liu T,et al. Efficient i-DFA construction algorithm based on state grouping[J]. Journal on Communications,2013,34(8):102 – 109. (in Chinese)

[5] Becchi M,Crowley P. A hybrid finite automaton for practical deep packet inspection[A]. Proceedings of the 2007 ACM CoNEXT Conference[C]. ACM,2007. 1.

[6] 张树壮,罗浩,方滨兴. 面向网络安全的高性能特征匹配技术研究[J]. 软件学报,2011,22(8):1838 – 1854.
Zhang S,Luo H,Fang B. Regular expressions matching for network security[J]. Journal of Software, 2011, 22(8): 1838 – 1854. (in Chinese)

[7] Kumar S,Dharmapurikar S,Yu F,et al. Algorithms to accelerate multiple regular expressions matching for deep packet inspection[J]. ACM SIGCOMM Computer Communication Review,2006,36(4):339 – 350.

[8] Ficara D,Giordano S,Procissi G,et al. An improved DFA for fast regular expression matching[J]. ACM SIGCOMM Computer Communication Review,2008,38(5):29 – 40.

[9] Liu T,Yang Y,Liu Y,et al. An efficient regular expressions compression algorithm from a new perspective[A]. Proceedings of IEEE INFOCOM 2001[C]. IEEE,2011. 2129 – 2137.

[10] Smith R,Estan C,Jha S,et al. Deflating the big bang:fast and scalable deep packet inspection with extended finite automata[J]. ACM SIGCOMM Computer Communication Review,2008,38(4):207 – 218.

[11] 宫阳阳,刘勤让,邵翔宇,等. 基于多维有限自动机的 DFA 改进算法[J]. 电子学报,2014,42(9):1818 – 1822.
Gong Y,Liu Q,Shao X,et al. A regular expression matching algorithm based on multi-dimensional cube[J]. Acta Electronica Sinica,2014,42(9):1818 – 1822. (in Chinese)

[12] Liu C,Pan Y,Chen A,et al. A DFA with extended character-set for fast deep packet inspection[J]. IEEE Transactions on Computers,2014,63(8):1527 – 1937.

[13] Regular Expression Processor[EB/OL]. http://regex.wustl.edu/index.php/Main_Page,2014.

作者简介

邵翔宇 男,1992 年 1 月出生,河南清丰人. 2012 年毕业于电子科技大学,现为国家数字交换系统工程技术研究中心在读研究生,主要研究方向为宽带信息网络及芯片设计.
E-mail:sxyslf@163.com

刘勤让 男,1975 年 11 月出生,河南睢县人. 现为国家数字交换系统工程技术研究中心研究员,硕士生导师. 主要研究方向为宽带信息网络及芯片设计.
E-mail:qinrangliu@sina.com

谭力波 男,1981 年 11 月出生,内蒙古赤峰人. 2004 年毕业于武汉理工大学,现为国家数字交换系统工程技术研究中心工程师,主要研究方向为芯片设计技术.