



综述

模式匹配在网络安全中的研究

徐东亮¹, 张宏莉¹, 张磊², 姚崇崇¹

(1. 哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001;

2. 河南大学数据与知识工程研究所 开封 475004)

摘要: 模式匹配模块是网络安全系统的核心模块, 其系统资源占用率最高时可达 70% 以上。针对模式匹配在当前网络安全系统中面临的问题和挑战, 以 2000 年为界将模式匹配方法分为传统模式匹配方法和新模式匹配方法, 通过分析不同模式匹配方法的原理、计算复杂度和新模式匹配方法的发展与演化过程, 得出不同方法的适用环境, 对模式匹配技术在网络安全系统中的作用进行了全面综述和评价, 最后对未来的研究方向进行了总结和展望。

关键词: 模式匹配; 入侵检测; 网络安全

doi: 10.11959/j.issn.1000-0801.2015068

Study on Pattern Matching in Network Security

Xu Dongliang¹, Zhang Hongli¹, Zhang Lei², Yao Chongchong¹

(1. School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China;

2. Institute of Data and Knowledge Engineering, Henan University, Kaifeng 475004, China)

Abstract: Pattern matching is the core module in network security system, it's the highest occupancy rate of system resources is more than 70%. According to problem and challenges of pattern matching in the network security system, taking 2000 as the boundary, the pattern matching was divided into the traditional pattern matching method and the new pattern matching method. Through the analysis of the principle and the computational complexity of different pattern matching method and the development and evolution of the new pattern matching method, applicable environment of different methods was concluded. A comprehensive review and evaluation of pattern matching in network security system was presented. Finally, a conclusion and some suggestions for future research were summarized and prospected.

Key words: pattern matching, intrusion detection, network security

收稿日期: 2015-01-07; 修回日期: 2015-03-04

基金项目: 国家重点基础研究发展计划 (“973” 计划) 基金资助项目 (No. 2011CB302605), 国家高技术研究发展计划 (“863” 计划) 基金资助项目 (No. 2011AA010705, No. 2012AA012502, No. 2012AA012506), 国家科技支撑计划基金资助项目 (No. 2012BAH37B01), 国家自然科学基金基金资助项目 (No. 61202457, No. 61402149)

Foundation Items: The National Basic Research Program of China (973 Program) (No. 2011CB302605), The National High Technology Research and Development Program of China (863 Program) (No. 2011AA010705, No. 2012AA012502, No. 2012AA012506), The National Key Technology R&D Program (No. 2012BAH37B01), The National Science Foundation of China (No. 61202457, No. 61402149)

论文引用格式: 徐东亮, 张宏莉, 张磊等. 模式匹配在网络安全中的研究. 电信科学, 2015068

Xu D L, Zhang H L, Zhang L, *et al.* Study on Pattern Matching in Network Security. Telecommunications Science, 2015068

1 引言

近年来,随着计算机、智能手机和互联网的迅速发展,新的应用软件和服务不断出现,计算机、智能手机和互联网已经深入人类生活、工作、科技和文化的各个领域。然而,在人们充分享受计算机网络带来的便捷的同时,也不得不面临一个无法回避的问题,即网络信息安全问题。

数据流管理是管理安全分析技术之一,是处理相对固定不变的大量查询和源源不断的流动数据的技术。数据流处理器实际上也可以看成一种数据管理系统,该系统的组织结构如图1所示。在数据流管理中,相对静止的是检索条件和操纵程序,相对易变的是数据,数据流管理的对象就是检索条件和操纵程序。

目前,网络安全方面的数据流管理系统主要有防火墙系统、病毒防护系统、恶意软件检测系统、入侵检测系统等。在这些网络安全系统中,数据通常以大量、快速、持续的数据流的形式到达。在网络安全领域,如何对这些数据进行快速有效的处理一直是一个具有挑战性的问题。从图1可以看出,数据流管理系统都需要一个核心模块——数据流查询(数据分组检测或分组过滤)。早先的数据分组检测主要是检测数据分组头的五元组,这种方法简单、高效,对应用层协议相对透明,但是这种方法无法检测数据分组的内容。为了解决这一问题,研究者们将注意力转移到基于内容的过滤方法——深度分组检测技术。与只扫描数据分组头五元组的传统方法不同,深度分组检测技术对数据分组的负载(payload)进行扫描分析,在扫描过程中将数据分组的内容与一组给定的规则进行比较,检测其中的

非法内容,以决定该数据是否应被进一步处理。该过程所依赖的最重要的技术就是模式匹配技术^[1]。

从20世纪70年代到21世纪的今天,模式匹配算法经历了从单模式匹配到多模式匹配、从精确匹配到近似匹配、从串行匹配到并行匹配、从软件算法到硬件实现的多个阶段。下面以2000年为分界线,对近半个世纪的模式匹配算法的发展进行简单介绍。

2 问题描述

模式匹配^[1],即字符串(特征、模式、字符串、规则这几个词在本文中意义相同)匹配,一个字符串是一个定义在有限字母表 Σ 上的字符序列。根据模式串和文本串是否预先给出,本文将模式匹配问题分为两大类:一是不对文本进行预处理的在线模式匹配(online string matching);二是文本预先给出,可以对文本建立索引的离线模式匹配。由于网络内容安全中涉及的大都是在线匹配问题,因此本文只关注第一类问题,即不对文本建立索引数据结构。

根据算法的功能可以分为单模式匹配算法、多模式匹配算法、近似匹配算法、基于硬件的并行模式匹配算法。

单模式匹配算法是指扫描一遍文本只匹配一个模式串的算法,经典的单模式匹配算法有Knuth-Morris-Pratt(KMP)算法^[2]和Boyer-Moore(BM)算法^[3]等。

多模式匹配算法是指只扫描一遍文本就可以查找出字符串集合 $P=\{p^1, p^2, \dots, p^r\}$ 中所有字符串出现的位置。大多数单模式字符串匹配算法经过一定的改进都可以很容易地扩展成多模式字符串匹配算法。例如,Aho-Corasick(AC)算法^[4]是由KMP算法扩展而来的,backward dawg matching(BDM)算法^[5]和Wu-Manber(WM)算法^[6]是由BM算法扩展而来的。

近似模式匹配就是允许在文本 T 中查找到的匹配与模式串 p 之间有 k 个不同。近似模式匹配问题中最经典的算法都是基于动态规划的匹配算法,除此之外还有基于自动机的方法、基于位并行的方法等^[7]。

基于硬件实现的算法是指将传统的模式匹配算法移植到具有更加高效计算能力的专用硬件后的算法,或者根据硬件本身的特性设计实现一种专门在该硬件上使用的模式匹配算法。多核处理器、众核处理器及专用处理芯片等硬件技术的发展使得并行的深层次细粒度的内容检测和分析被越来越多地用于处理模式匹配问题。

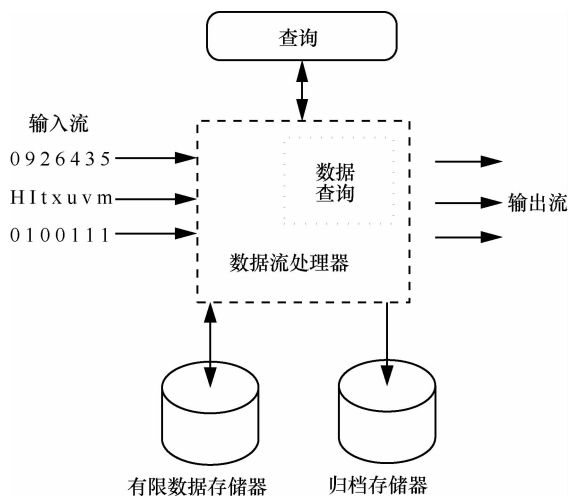


图1 数据流管理系统的组织结构



3 传统模式匹配算法

本文将 2000 年之前提出的模式匹配算法称为传统模式匹配算法,2000 年之后出现的算法称为新模式匹配算法。传统模式匹配算法大多都是纯软件算法,主要是精确模式匹配算法和近似模式匹配算法,采用的方法可以概括为前缀匹配、后缀匹配、子串匹配等几大类,其代表算法有 KMP 算法、Shift-And^[8]算法、Shift-Or^[9]算法、BM 算法、Horspool^[10]算法、backward nondeterministic DAWG matching (BNDM)算法^[11]、backward oracle matching (BOM)算法^[12]等,所用到的技术包括滑动窗口、位并行、自动机和后缀树等。

3.1 精确模式匹配算法

精确模式匹配算法主要分为单模式匹配算法和多模式匹配算法。在网络内容安全分析中,单模式匹配算法使用较少,现在只有理论研究价值,而多模式匹配算法被广泛使用。

3.1.1 单模式匹配算法

字符串匹配问题最坏计算复杂度的下界是 $O(n)$,第一个达到这个下界的算法是 1970 年由 Morris 和 Pratt 提出的^[13],之后 Knuth 等人在 1977 年将其改进成著名的单模式匹配算法——KMP 算法,它是第一个在线性时间解决模式匹配问题的算法,也是单模式匹配算法中前缀匹配滑动窗口技术的代表算法之一。

(1) 前缀匹配

KMP 算法的匹配过程如图 2 所示。在 $T = \text{"abcabcabcaddabba"}$ 中查找 $p = \text{"abcabd"}$,模式串 p 的长度为搜索窗口。在匹配时,窗口从左到右逐个字符比较。当字符发生不等时可以跳跃。当第一次搜索到 $T[5]$ 和 $p[5]$ 不匹配时,模式串 p 并不是向后移动 1 位,而是根据 p 中 $p[5] = \text{'d'}$ 的字符串函数值 ($\text{next}[5] = 2$) 检测 $T[5]$ 和 $p[2]$ 是否相等。如果相等, T 和 p 的下标同时增加,最终在 T 中匹配到 p 。

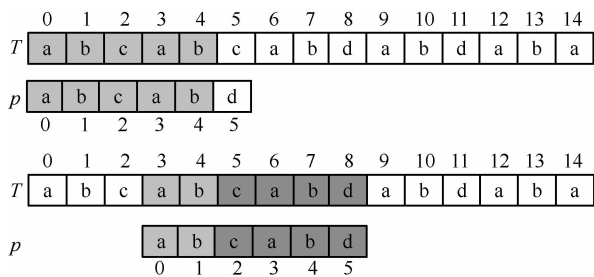


图 2 KMP 算法的匹配过程示意

(2) 后缀匹配

后缀匹配方法移动窗口时是从左到右的,而进行比较时窗口内则从右向左逐个比较字符,找出窗口内文本子串和模式串的最长公共后缀。如果到达窗口的起始位置,说明已经发现了一个匹配。在单模式匹配算法中,BM 算法是后缀匹配方法的代表算法之一。

BM 算法包含两个跳转方法:坏字符跳转和好后缀跳转。这两种跳转方法的目的是让模式串每次向右移动尽可能大的距离。如图 3 所示,好后缀跳转有两种情况:在情况 1 中,如果子串和好后缀在模式串中被匹配到,则最右边的子串将被移动到好后缀的位置,继续匹配;在情况 2 中,如果模式串中没有与好后缀相同的子串,则在模式串 p 和好后缀中找到一个最长前缀,并与之对齐。

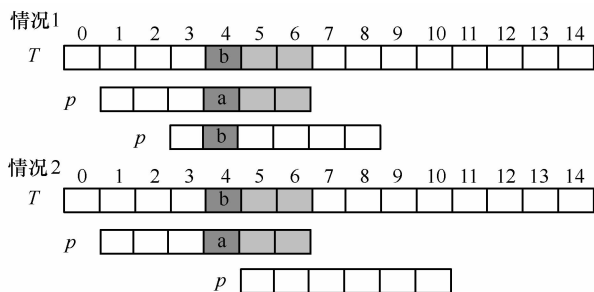


图 3 坏字符跳转示意

BM 算法每次向右移动模式串的距离是根据好后缀跳转和坏字符跳转得到的最大距离。BM 算法具有亚线性的平均时间复杂度,而搜索阶段的最坏时间复杂度为 $O(m \times n)$ 。BM 算法的很多变体都可以保证线性的最坏时间复杂度。

(3) 子串匹配

基于子串匹配的算法在字符等概率出现且互相独立的情况下,具有最优的平均时间复杂度。搜索窗口在基于子串匹配算法中的移动很优美,如图 4 所示。假设模式串的子串 u 已经在匹配窗口中从后向前倒序地被识别出来,且下一个字符“b”无法被继续识别,这意味着 p 的子串中不包含 bu ,文本中不可能存在覆盖 bu 的字符串。所以,这时匹配窗口可以被安全地移动到 u 之后。如何识别模式串

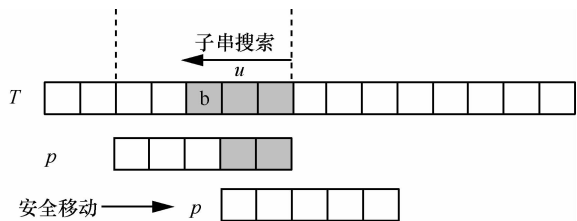


图 4 子串搜索示意

的所有子串是基于子串匹配算法的最大难点。

在单模式匹配算法中, BDM 算法^[5]是基于子串搜索的代表算法之一。该算法使用了后缀自动机这种数据结构, 虽然功能强大, 但结构复杂。当模式串的长度不超过机器字长时, 后缀自动机可以用位并行方法进行有效模拟。

位并行算法 BNDM^[11]比 BDM 算法更快、更易于实现, 且可用于扩展字符串匹配, 它与 BDM 算法具有相同的最坏时间复杂度 $O(m \times n)$, 也都具有最优的平均时间复杂度 $O(n \log m/m)$; 当模式串的长度较长时, BOM 算法^[12]能够获得与 BDM 算法相同的效果, 且具有相同的最坏时间复杂度, 但是 BOM 算法对基于子串的方法进行了修改, 采用了更简单的自动机——factor oracle。

3.1.2 多模式匹配算法

单模式匹配方法可以很自然地扩展到同时搜索一组字符串 $P = \{p^1, p^2, \dots, p^r\}$ 的方法, 这就是多模式匹配。其中, p^i 是在有限字母 Σ 表上定义的字符串 $p^i = p_1^i p_2^i \dots p_{m_i}^i$ 。设 P 中

所有字符串长度之和为 $|P|$, 即 $|P| = \sum_{i=1}^r |p^i| = \sum_{i=1}^r m_i$, P 中最长模式串和最短模式串的长度分别为 $l_{\max}$ 和 $l_{\min}$ 。 P 中的一些字符串或者其子串、前缀、后缀可能与集合中的其他字符串相同。在文本 T 中同时搜索 r 个模式串, 可以直接执行 r 遍某个单模式匹配算法, 这种简单直接的匹配方法的最坏时间复杂度为 $O(r \times n)$, 预处理的最坏时间复杂度为 $O(|P|)$ 。

单模式匹配中用到的 3 类主要搜索方法——前缀搜索、后缀搜索和子串搜索, 也是多模式匹配所用到的 3 类主要方法。

(1) 前缀匹配

前缀匹配是指在固定长度的窗口内逐一字符自左向右地扫描文本, 对于已读入的字符, 既是文本的后缀也是模式集中某个模式串的最长前缀, 如图 5 所示。

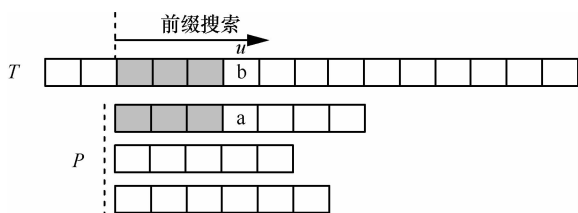


图 5 多模式前缀匹配示意

最经典的前缀匹配算法是著名的 AC 算法。该算法使用了一种被称为 Aho-Corasick (AC) 自动机的特殊自动机。该自动机由模式集 P 生成, 它是对 trie 树添加了失败转移

函数扩展而成的。

自动机在模式匹配的应用中性能稳定, 由于具有不依赖于模式集合和文本本身的特点, 所以一直受到应用者的欢迎。但自动机有一个缺点, 即它需要很大的内存空间, 尤其在正则表达式的匹配中, 正则表达式的通配符等引起的状态爆炸导致内存爆炸, 这使得自动机在应用中受到很大限制。

(2) 后缀匹配

后缀匹配是指在固定窗口内从右向左逐一字符匹配模式串的后缀。与单模式串的后缀匹配算法相似, 根据匹配到后缀的下一次出现位置来移动模式串。这种方法可以跳过一些不必要查找的字符, 如图 6 所示。

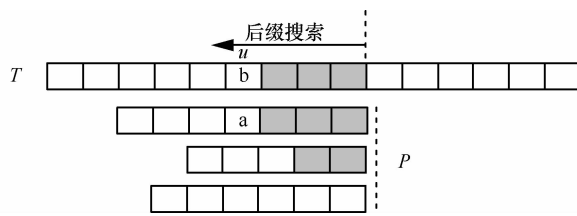


图 6 多模式后缀匹配示意

WM 算法是多模式后缀匹配方法的代表算法之一。散列和跳跃不可能匹配的字符是 WM 算法最重要的思想。Commentz-Walter 算法^[14]是第一个从单模式扩展到多模式的后缀匹配算法, 虽然在时间、空间、实用价值方面它都无法与经典的 AC 算法和 WM 算法等相比, 但它是第一个实现亚线性平均时间复杂度的多模式匹配算法, 在多模式匹配算法史上具有里程碑式的意义。

(3) 子串匹配

子串匹配是指搜索沿着文本从后向前进行, 在模式串长度为 $l_{\min}$ 的前缀中搜索子串, 以此决定当前文本位置的移动。这种方法也可以跳过不必要的字符, 如图 7 所示。

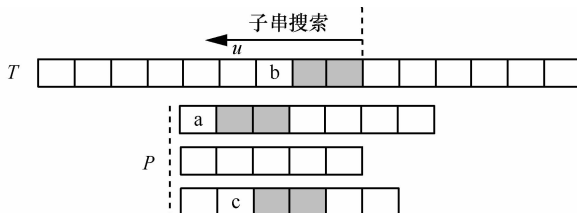


图 7 多模式子串匹配示意

一般基于子串的方法只要解决两个技术问题, 都可以直接扩展到一组模式串的情形: 如何安全地移动模式串, 以避免遗漏可能的成功匹配; 如何识别一组字符串的子



串。SBOM 算法^[15]和 SBDM 算法^[16]是多模式匹配方法中基于子串搜索的代表算法,它们分别是对 BOM 算法和 BDM 算法的扩展。

SBDM 算法在长度为 $l_{\min}$ 的文本窗口内使用以模式集建立的后缀自动机来移动窗口,移动时需要从后向前识别出模式串的所有子串。该算法在所有模式串长度为 $l_{\min}$ 的前缀窗口上建立反转自动机,从后向前识别文本在 $l_{\min}$ 的窗口中的最长后缀,该后缀同时也是模式集 P 中模式串长度为 $l_{\min}$ 的前缀子串。

SBOM 算法使用 factor oracle 结构为模式串集合 P 建立自动机,该自动机识别的是模式集合 P 中所有模式串的所有子串的超集。搜索时,在所有模式串的前缀 $l_{\min}$ 的窗口上建立反转自动机 factor oracle (时间复杂度为 $O(r \times l_{\min})$),在长度为 $l_{\min}$ 的文本窗口内从后向前匹配文本字符来移动窗口。如果无法继续识别文本字符,那么当前窗口可以安全地移动到该字符之后;如果完全识别了窗口内的字符,到达了窗口的起始位置,那么需要将 P 的一个子集与文本进行比较验证。SBOM 算法与 SBDM 算法的最坏时间复杂度相同,均为 $O(n \times |P|)$,平均时间复杂度也都是亚线性的。

3.2 近似模式匹配算法

近似串匹配又称“模糊匹配”或“允许误差的串匹配”,就是查找一个模式串 p 在文本 T 中出现的所有位置,并且允许有限的 k 个差异存在于模式串 p 和它在文本中的出现之间。近似模式匹配目前主要有如下 4 种方法。

(1)用来计算编辑距离的动态规划算法,其中 1980 年 Sellers 设计了一个非常经典的算法^[17],它的时间复杂度为 $O(m)$ 。如此看来,该算法的效率不是很高,但对它稍加改动就可以适用于更复杂的模型。

(2)使用位并行等各种方法模拟实现自动机进行匹配。这种自动机最初以其确定化的形式在参考文献[18]中被提出,后来在参考文献[19]中给出了更明确的阐述。当模式串较短时,这类算法效率不错。

(3)用位并行方法模拟其他方法,该方法是所有方法中最有效的。近似搜索中大量使用位并行方法,很多很好的结果都是使用位并行方法得到的。

(4)过滤方法,该方法先执行一个高效的过滤算法,这样文本中那些不存在成功匹配的很多区域就可以被直接跳过,不需要匹配,然后剩下的少量文本区域是否存在成功匹配,再利用一个普通的模式匹配算法验证即可。

4 新模式匹配算法

2000 年以前提出的经典模式匹配算法共 40 种左右,但在 2000 年至今的这 15 年里就提出了 50 种左右,但这 50 种算法大多数都是对以前经典算法的改进或者移植,且主要是面向并行或者基于硬件加速的模式匹配方法。本节介绍最近这 15 年提出的新模式匹配算法。

4.1 精确模式匹配算法

4.1.1 单模式匹配算法

BM 算法激发了后续单模式匹配算法的研究,近十几年出现的很多单模式匹配算法都是 BM 算法的变体。

fast-search^[20]算法是一个简单有效的 BM 算法的变体,模式匹配阶段在模式串的第 $m-1$ 个字符处开始,根据 horspool 算法的坏字符规则计算移动增量。在匹配结束阶段,采用好后缀规则进行跳转。随后,又产生了 fast-search 算法的两种变体:backward-fast-search^[21]和 forward-fast-search^[22],分别采用 backward good suffix 和 forward good suffix 规则。其中,规则表的构建需要 $O(m \times \max(m, \sigma))$ 时间, σ 表示字母表的大小。

算法 GRASPm^[23]是使用 horspool 坏字符的转移方法,通过散列函数在模式上计算 2-grams 的过滤方法。由此产生的算法最坏时间复杂度为 $O(m \times n)$,平均时间复杂度为亚线性,需要 $O(\sigma + m)$ 的空间计算散列表和坏字符转移表。

4.1.2 多模式匹配算法

近 15 年的多模匹配算法大部分是基于自动机原理的改进算法。

算法 extended-BOM^[24]和 simplified extended BOM^[25]都是 BOM 算法的变体,采用自动机的方法进行实现,这两个算法的区别在于计算模式串最右的两个字符转移是否在一步完成。forward-BOM^[23]算法进一步改进了 extended-BOM 算法,通过当前窗口后的字符计算转移混合了 extended-BOM 算法和 quick-search 算法。

Kumar 等人^[26]研究发现,自动机中很多失败转移指针指向相同的状态,这造成了大量状态转移边的冗余。为了解决这个问题,Kumar 等人介绍了一种新的自动机——delayed input DFA(D²FA)。D²FA 将 DFA(确定有限自动机)中所有指向相同状态的转移用一个默认转移代替,这样就大量减少了不同状态间转移的数量。算法 XFA^[27]使用有限暂存器扩展了有限状态自动机(finite state automata, FSA),并用指令来操作这个暂存器。这种方法的时间复杂

度接近 DFA,空间复杂度接近 NFA(非确定有限自动机)。

4.2 近似模式匹配算法

最近 15 年内的近似模式匹配方法发展依然缓慢,提出的算法不多。

Alba A 等人^[28]提出了针对近似模式匹配的新颖方法,该方法不依赖于编辑距离的计算,而是采用相似度指数来实现。在生物学领域,允许用户在短时间内找出相关的匹配。Kim Y 等人^[29]提出了针对近似串匹配的 Top-K 算法。采用 q -grams 跳跃字符的方式同样不需要计算字符串的编辑距离,通过真实的实验数据展示了算法的有效性和伸缩性。

Li C 等人^[30]研究了对于一个给定的模式串,如何快速找到它的一个相似模式串集合。在此基础上,他们提出了 ScanCount 算法、MergeSkip 算法和 DivideSkip 算法这 3 个近似模式匹配算法。Prasad 等人^[31]提出了两个多模式近似匹配算法:MASM1 和 MASM2。这两个算法可以处理长度超过一个机器字的模式串。Xu 等人^[32]在 Prasad 等人工作的基础上提出了一个改进的位并行算法——impMASM,该算法可以方便地移植到 GPU(graphic processing unit,图形处理器)上且可以处理不等长模式串。

4.3 并行模式匹配算法

近 15 年来,对纯软件并行算法的研究也不是很多。

马明^[33]采用 MPI 并行编程技术,使用 MPI 消息传递函数,在不同处理单元上进行并行处理,对经典的 KMP 算法和 BM 算法进行了并行实现,并给出了算法实现的核心代码。

独立并行压缩自动机 (independent parallel compact finite automata, PC-FA)^[34] 算法也是一个基于压缩自动机的并行模式匹配算法。该方法是将经典的单模式最长前缀匹配算法扩展到多模式匹配的算法。它同时使用了 k 个 PC-FA 处理输入流中的 k 个字符,该参考文献还从理论上证明了 DFA 与 PC-FA 是等价的。

4.4 基于硬件的模式匹配算法

进入 21 世纪,随着高性能硬件多核处理器、众核处理器、FPGA (field-programmable gate array, 现场可编程门阵列)、TCAM(ternary content addressable memory, 三态内容寻址存储器)和 GPU 等的兴起,模式匹配的研究另辟蹊径。多核、众核架构提供了一个低成本实现高速模式匹配的方法。Villa 等人^[35, 36]在 IBM cell broadband engine (CBE) 上实现了高速模式匹配系统,他们的工作表明将模式匹配算法经过优化,在多核架构上实现的性能可能超过专业的加速器。但是在 IBM CBE 上编程、优化非常复杂,因此谭光明

等人^[37]就开始在普通的多核架构下寻找解决模式搜索的方法。他们在普通的多核架构下设计实现了一个基于分解策略的高效多模式匹配算法,将模式集分解成多个子集,对每一个子集根据其模式特点选择合适的匹配算法,并将它们分配到每一个核上进行处理。

FPGA 具有高速计算能力和可重配能力,被广泛应用于网络安全过滤系统^[38-40]。基于 FPGA 的模式匹配技术^[41, 42]是利用片上资源把模式集编译成状态机,将该状态机存储于片内存储器内,然后再对输入数据进行过滤。纯 FPGA 技术由于本身存储器较小,很难满足大规模模式集的匹配,所以应用中的 FPGA 都与可重写 ROM 结合,采用比较的方法实现模式匹配算法。

TCAM 主要用于快速查找访问控制列表 (ACL)、路由表等,它是从 CAM 的基础上发展而来的。一般的 CAM 存储器中每个比特位的状态只有两个,即“0”或“1”,而 TCAM 中每个比特位有 3 种状态,除“0”和“1”之外,还有“don't care”状态,所以称为“三态”,它是通过掩码实现的。正是 TCAM 的第三种状态使其既能进行精确匹配查找^[43-45],又能进行模糊匹配查找^[46],而 CAM 没有第三种状态,所以只能进行精确查找^[47]。

将 GPU 用于图形处理以外的其他领域被称为基于 GPU 的通用计算 (general-purpose computing on graphics processing units, GPGPU)。CPU+GPU 的异构模式是 GPGPU 的通用模式,其中, GPU 负责并行处理密集型的海量数据, CPU 负责不适合并行计算的事务管理和复杂逻辑处理。该异构模式利用 GPU 的高带宽和高性能的并行计算能力弥补了 CPU 的计算能力不足的劣势。

GPU 是单指令多数据流 (simple instruction multiple data, SIMD) 体系结构,大量的计算部件使它很适合处理模式匹配算法中对比字符这种计算密集型操作。PixeSnort^[48]是第一种在入侵检测系统中将 GPU 用于模式匹配计算的方法。这种方法使用修改的 KMP 算法将数据分组内容匹配部分的工作从 CPU 移交给 GPU 处理,利用 GPU 强大的并行计算能力来提高流量处理能力。MIDeA^[49]架构利用了多个 NICs、多个 CPUs 和多个 GPUs 的并行性,将它们结合起来运用到实际网络中,将相应的操作映射到相应的设备中,构建了一个并行网络入侵检测系统。该系统没有顺序执行部件,无需同步、等待或者内容共享。大多数基于 GPU 实现的模式匹配方法都将 GPU 作为协处理器,帮助 CPU 分担主要的计算工作。而参考文献[50]实现了 GPU-to-GPU,



即输入和输出都在 GPU 的内存中进行,相对于 GPU 只作为 CPU 的协处理器,这种方法的匹配速度提高了 3~4 倍。

4.5 未来趋势

近两年来,对网络安全中模式匹配的研究主要集中在以下两个方面。

(1)利用专用硬件芯片,并行加速传统模式匹配算法或改进算法。例如,参考文献[51]基于 GPU 的并行体系结构实现了一个近似模式匹配算法;参考文献[52]利用二叉搜索树在 FPGA 上实现了一个内在高效的模式匹配算法。

(2)针对特定需求,改进现有算法或设计新算法。例如,参考文献[53]针对海量模式集造成的高冲突率导致算法效率下降的问题,提出了一种随机指纹模型,并将该模型用于 WM 算法,从而极大地降低了 WM 算法模式块散列的冲突率,提高了算法的效率;参考文献[54]对表达式的包含关系做了分类,提出了表达式冗余消除算法,然后在 BitCount 算法的基础上提出了一种时间复杂度为 $O(1)$ 的掩码验证算法 MaskVeri。除此之外,本文第 4 节中列出的近两年的参考文献也都是基于这两个方面做出的研究。

利用专用硬件芯片和针对特定环境需求,依然是未来网络安全中模式匹配最主要的两个研究方向。专用芯片有 GPU、CAM、FPGA、众核等;特定需求包括降低海量模式规则冲突率、高效内存利用率等。

5 结束语

从网络中快速通过的数据流中查找用户感兴趣的信息(海量模式集),是一项非常富有挑战性的工作。随着互联网的发展,网络流量日益增大,病毒、垃圾邮件等攻击手段越来越多样,安全系统中的模式集趋向海量发展。与之相反的是,安全系统中的核心模块——模式匹配算法的发展过于缓慢,传统的模式匹配算法性能已经趋于理论上限。模式匹配技术在实际应用中还面临以下两个问题。

- 到目前为止,还没有一种算法可以适用于任何网络环境。
- 目前,网络内容过滤方面的主要矛盾为纯软件的模式匹配算法的吞吐量已经达到或接近理论上限与网络流量指数级增长之间的矛盾。

本文针对目前模式匹配在网络安全系统中面临的挑

战,以 2000 年为分界线对传统模式匹配方法和新模式匹配方法进行了介绍。2000 年以前的传统模式匹配算法大都以纯软件算法为主,且部分算法性能已经接近或达到了理论上限。但是,即使性能达到理论上限的算法也并不是适用于所有环境的,这也是至今依然有很多研究者从事模式匹配问题研究的原因。2000 年以后,由于专用芯片等硬件的快速发展,众多的研究者开始改进大量的传统模式匹配算法,利用专用芯片的高计算性能来提高匹配效率。

不难看出,经过近半个世纪的发展,模式匹配技术已经从纯软件方法逐渐发展到借助于专用硬件来提升效率。因此,本文针对海量模式集和特定应用环境,着重介绍了利用专用芯片加速模式匹配算法,提升网络安全系统性能,这也是目前网络安全中模式匹配的重要方向。

参考文献

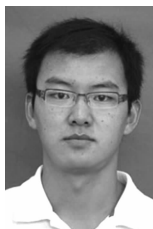
- 1 Navarro G, Raffinot M. Flexible Pattern Matching in Strings: Practical On-Line Search Algorithms for Texts and Biological Sequences. Cambridge: Cambridge University Press, 2002
- 2 Knuth D E, Morris J H, Pratt V R, *et al.* Fast pattern matching in strings. SIAM Journal on Computing, 1977, 6(2): 323~350
- 3 Boyer R S, Moore J S. A fast string searching algorithm. Communications of the ACM, 1977, 20(10): 762~772
- 4 Aho A V, Corasick M J. Efficient string matching: an aid to bibliographic search. Communications of the ACM, 1975, 18(6): 333~340
- 5 Crochemore M, Czumaj A, Gasieniec L, *et al.* Speeding up two string-matching algorithms. Algorithmica, 1994, 12(4~5): 247~267
- 6 Wu S, Manber U. A Fast Algorithm for Multi-Pattern Searching. Technical Report TR-94-17, University of Arizona, 1994
- 7 Ukkonen E. Finding approximate patterns in strings. Journal of Algorithms, 1985, 6(1~3): 132~137
- 8 Wu S, Manber U. Fast text searching: allowing errors. Communications of the ACM, 1992, 35(10): 83~91
- 9 Baeza-Yates R A, Gonnet G H. A new approach to text searching. Communications of the ACM, 1992, 35(10): 74~82
- 10 Horspool R N. Practical fast searching in strings. Software: Practice and Experience, 1980, 10(6): 501~506
- 11 Navarro G, Raffinot M. Fast and flexible string matching by combining bit-parallelism and suffix automata. ACM Journal of Experimental Algorithmics, 2000, 5(4): 1~36

- 12 Allauzen C, Crochemore M, Raffinot M. Efficient experimental string matching by weak factor recognition. *Proceedings of 17th Annual Symposium on Combinatorial Pattern Matching*, Barcelona, Spain, 2006: 51~72
- 13 Morris J J, Pratt V R. A Linear Pattern-Matching Algorithm. Technical Report 40, University of California, 1970
- 14 Commentz-Walter B. A String Matching Algorithm Fast on the Average. Berlin: Springer Berlin Heidelberg, 1979
- 15 Allauzen C, Raffinot M. Factor Oracle of a Set of Words. Institute Gaspard-Monge, University de Marne-la-Vallée, TR-99-11, 1999
- 16 Blumer A, Blumer J, Haussler D, *et al.* Complete inverted files for efficient text retrieval and analysis. *Journal of the ACM*, 1987, 34(3): 578~595
- 17 Sellers P H. On the theory and computation of evolutionary distances. *SIAM Journal on Applied Mathematics*, 1974, 26(4): 787~793
- 18 Ukkonen E. Finding approximate patterns in strings. *Journal of Algorithms*, 1985, 6(1): 132~137
- 19 Wu S, Manber U. Fast text searching: allowing errors. *Communications of the ACM*, 1992, 35(10): 83~91
- 20 Cantone D, Faro S. Fast-search: a new efficient variant of the boyer-moore string matching algorithm. *Proceedings of the 2nd International Workshop on Experimental and Efficient Algorithms*, Ascona, Switzerland, 2003: 247~267
- 21 Cantone D, Faro S. Forward-fast-search: another fast variant of the boyer-moore string matching algorithm. *Proceedings of the Prague Stringology Conference*, Prague, Czech Republic, 2003: 10~24
- 22 Cantone D, Faro S. Searching for a substring with constant extra-space complexity. *Proceedings of the 3rd International Conference on Fun with Algorithms*, Isola d'Elba, Italy, 2004: 118~131
- 23 Deusdado S, Carvalho P. GRASPm: an efficient algorithm for exact pattern-matching in genomic sequences. *International Journal of Bioinformatics Research and Applications*, 2009, 5(4): 385~401
- 24 Fan H, Yao N, Ma H. Fast variants of the backward-oracle-matching algorithm. *Proceedings of the 4th International Conference on Internet Computing for Science and Engineering*, Harbin, China, 2009: 56~59
- 25 Faro S, Lecroq T. Efficient variants of the backward-oracle-matching algorithm. *Proceedings of the Prague Stringology Conference*, Prague, Czech Republic, 2008: 146~160
- 26 Kumar S, Dharmapurikar S, Yu F, *et al.* Algorithms to accelerate multiple regular expressions matching for deep packet inspection. *Proceedings of SIGCOMM*, Pisa, Italy, 2006: 339~350
- 27 Smith R, Estan C, Jha S. XFA: Faster signature matching with extended automata. *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 2008: 187~201
- 28 Abela A, Rodriguez-Kessler M, Arce-Santana E R, *et al.* Approximate string matching using phase correlation. *Proceedings of the 34th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, San Diego, CA, USA, 2012: 6309~6312
- 29 Kim Y, Shim K. Efficient top-*k* algorithms for approximate substring matching. *Proceedings of ACM SIGMOD*, New York, USA, 2013: 385~396
- 30 Li C, Lu J H, Lu Y M. Efficient merging and filtering algorithms for approximate string searches. *Proceedings of IEEE 24th International Conference on Data Engineering*, Cancun, Mexico, 2008: 257~266
- 31 Prasad R, Sharma A K, Singh A, *et al.* Efficient bit-parallel multi-patterns approximate string matching algorithms. *Scientific Research and Essays*, 2011, 6(4): 876~881
- 32 Xu K, Cui W, Hu Y, *et al.* Bit-parallel multiple approximate string matching based on GPU. *Procedia Computer Science*, 2013, 17(5): 523~529
- 33 马明. 串匹配算法的并行实现策略 (硕士学位论文). 湖北大学, 2010
Ma M. The parallel implementation of string matching algorithm (master dissertation). Hubei University, 2010
- 34 Tang Y, Jiang J, Wang X, *et al.* Independent parallel compact finite automata for accelerating multi-string matching. *Proceedings of IEEE Global Telecommunications Conference*, Miami, FL, USA, 2010: 1~5
- 35 Villa O, Scarpazza D P, Petrini F. Accelerating real-time string searching with multicore processors. *Computer*, 2008, 41(4): 42~50
- 36 Scarpazza D P, Villa O, Petrini F. Peak-performance DFA-based string matching on the cell processor. *Proceedings of 21th International Parallel and Distributed Processing Symposium*, Long Beach, USA, 2007: 1~8
- 37 Tan G M, Liu P, Bu D B, *et al.* Revisiting multiple pattern matching algorithms for multi-core architecture. *Journal of Computer Science and Technology*, 2011, 26(5): 866~874
- 38 Song H, Lockwood J W. Efficient packet classification for network intrusion detection using FPGA. *Proceedings of the 13th International Symposium on Field-Programmable Gate Arrays*, Monterey, CA, USA, 2005: 238~245
- 39 Schuehler D V, Lockwood J W. A Modular System for FPGA-Based TCP Flow Processing in High-Speed Networks. Berlin: Springer Berlin Heidelberg, 2004: 301~310



- 40 Katashita T, Maeda A, Toda K, *et al.* Highly efficient string matching circuit for IDS with FPGA. Proceedings of the 14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, Napa, CA, USA, 2006: 285~286
- 41 Dandass Y S, Burgess S C, Lawrence M, *et al.* Accelerating string set matching in FPGA hardware for bioinformatics research. BMC Bioinformatics, 2008, 9(1)
- 42 Van Court T, Herbordt M C. Families of FPGA-based accelerators for approximate string matching. Microprocessors and Microsystems, 2007, 31(2): 135~145
- 43 Meiners C R, Patel J, Norige E, *et al.* Fast regular expression matching using small TCAMs for network intrusion detection and prevention systems. Proceedings of the 19th USENIX Conference on Security, Washington, USA, 2010
- 44 Hua N, Song H, Lakshman T V. Variable-stride multi-pattern matching for scalable deep packet inspection. Proceedings of IEEE INFOCOM, Rio de Janeiro, Brazil, 2009: 415~423
- 45 Bremner-Barr A, Hay D, Koral Y. CompactDFA: generic state machine compression for scalable pattern matching. Proceedings of IEEE INFOCOM, San Diego, CA, USA, 2010: 1~9
- 46 张小山, 赵国鸿, 王勇军等. 基于 TCAM 的深度报文过滤技术研究. 2005 全国网络与信息安全技术研讨会论文集, 2005: 143~148
- Zhang X S, Zhao G H, Wang Y J, *et al.* Research on deep packet filtering technology based on TCAM. 2005 National Conference on Network and Information Security Technology, 2005: 143~148
- 47 Gan C G. FPGA based CAM architecture string matching for network intrusion detection (Ph D dissertation). Universiti Teknologi Malaysia, Faculty of Electrical Engineering, 2012
- 48 Nigel J, Carla B. Offloading IDS computation to the GPU. Proceedings of the 22nd Computer Security Applications Conference, Miami Beach, FL, USA, 2006: 371~380
- 49 Vasiliadis G, Polychronakis M, Ioannidis S. MIDeA: a multi-parallel intrusion detection architecture. Proceedings of the 18th ACM Conference on Computer and Communications Security, Chicago, USA, 2011
- 50 Zha X, Sahni S. GPU-to-GPU and host-to-host multipattern string matching on a GPU. IEEE Transactions on Computers, 2013, 62(6): 1156~1169
- 51 Man D, Nakano K, Ito Y. The approximate string matching on the hierarchical memory machine, with performance evaluation. Proceedings of the 7th International Symposium on Embedded Multicore Socs, Tokyo, Japan, 2013: 79~84
- 52 Le H, Prasanna V K. A memory-efficient and modular approach for large-scale string pattern matching. IEEE Transactions on Computers, 2013, 62(5): 844~857
- 53 张宏莉, 徐东亮, 梁敏等. 海量模式高效匹配方法研究. 电子学报, 2014, 42(6): 1220~1224
- Zhang H L, Xu D L, Liang M, *et al.* Massive strings efficient matching method research. Acta Electronica Sinica, 2014, 42(6): 1220~1224
- 54 杨天龙, 张宏莉. 一种关键字表达式的匹配优化方法. 电信科学, 2013, 29(1): 39~45
- Yang T L, Zhang H L. Optimization of expression matching for string matching. Telecommunications Science, 2013, 29(1): 39~45

[作者简介]



徐东亮,男,哈尔滨工业大学计算机科学与技术学院博士生,主要研究方向为模式匹配、入侵检测和信息安全等。



张宏莉,女,博士,哈尔滨工业大学计算机科学与技术学院教授、博士生导师,主要研究方向为网络与信息安全、网络测量与建模、网络计算、并行处理等。



张磊,男,河南大学数据与知识工程研究所博士生,主要研究方向为概念格分析、数据挖掘和信息安全。



姚崇崇,男,哈尔滨工业大学计算机科学与技术学院硕士生,主要研究方向为字符串匹配、入侵检测。