

## 基于 TCAM 的低能耗正则表达式匹配算法

丁麟轩<sup>1</sup>, 黄昆<sup>2</sup>, 张大方<sup>1</sup>

(1. 湖南大学 信息科学与工程学院, 湖南 长沙 410082; 2. 中国科学院 计算技术研究所, 北京 100190)

**摘 要:** 提出一种基于字符索引的正则表达式匹配算法, 对确定型有限自动机 (DFA, deterministic finite automaton) 的字母表和状态进行分离存储, 构建字符索引, 减少匹配时激活的 TCAM 块数, 显著降低 TCAM 能耗。实验结果表明: 与 DFA 相比, 基于字符索引的 DFA (CIDFA, character-indexed DFA) 在能耗上平均减少了 92.7%, 在存储空间开销上平均减少了 32.0%, 在吞吐量上平均提高了 57.9%。

**关键词:** 正则表达式匹配; 字符索引; 分块存储; 低能耗

中图分类号: TP393

文献标识码: A

文章编号: 1000-436X(2014)08-0162-07

## Low-power TCAM for regular expression matching

DING Lin-xuan<sup>1</sup>, HUANG Kun<sup>2</sup>, ZHANG Da-fang<sup>1</sup>

(1. College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China;

2. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China)

**Abstract:** A character-indexed regular expression matching algorithm was presented to address the limitations of TCAM power consumption. This algorithm used the idea of separating the alphabet table from the states in deterministic finite automaton (DFA) for building a character index, in order to reduce the number of activated TCAM blocks, which in turn translated low TCAM power. Experimental results showed that proposed algorithm reduces the TCAM power by 92.7% on average as well as the TCAM space usage by 32.0% on average, and improves the matching throughput by 57.9% on average compared to previous solutions based on DFA.

**Key words:** regular expression matching; character index; block-based storage; low power

### 1 引言

近年来, 随着网络应用的日益增多, 网络安全面临着越来越严峻的挑战。网络入侵检测与防御系统 (NIDPS, network intrusion detection/prevention system) 是网络安全防御的主要手段, 它通过实时监测网络流量, 检查和阻止网络攻击<sup>[1]</sup>。深度分组检测<sup>[2]</sup> (DPI, deep packet inspection) 是 NIDPS、防火墙和应用层交换机的核心部件, 不仅检查数据分组的头部信息, 而且检查数据分组的有效载荷。由于正则表达式具有丰富和灵活的表达能力, 目前主流的 NIDPS, 例如 Snort<sup>[3]</sup>、Bro<sup>[4]</sup>、TippingPoint<sup>[5]</sup>、

Cisco<sup>[6]</sup>广泛采用正则表达式匹配算法进行 DPI 操作。

三态内容可寻址存储器 (TCAM, ternary content addressable memory) 是一种快速并行查找的专门硬件。TCAM 在一个时钟周期内并行查找所有表项, 返回与关键字匹配的第一条表项的地址。TCAM 具备三态表示特性, 即 TCAM 表项可存储 0、1 和 “\*”, 其中 “\*” 表示无关位, 可以匹配 0 或 1。在网络处理应用中, TCAM 通常与静态随机存储器 (SRAM, static random access memory) 结合在一起使用, 在关键字查找匹配成功后, TCAM 将匹配的表项地址发送到 SRAM, 输出 SRAM 中保存的相应信息。与 TCAM 相比, SRAM 的存储空间大且能耗低。

收稿日期: 2013-05-24; 修回日期: 2013-12-20

基金项目: 国家重点基础研究发展规划 (“973” 计划) 基金资助项目 (2012CB315805); 国家自然科学基金资助项目 (61173167, 61100171)

Foundation Items: The National Basic Research Program of China (973 Program) (2012CB315805); The National Natural Science Foundation of China (61173167, 61100171)

近年来,研究者提出了多种基于 TCAM 的正则表达式匹配算法<sup>[7-9]</sup>,实现线速 DPI。基于 TCAM 的正则表达式匹配算法采用 TCAM 表示确定型有限自动机 (DFA, deterministic finite automaton),即将状态迁移表的每条状态迁移边存储在一条 TCAM 表项中,每条表项由 TCAM 和 SRAM 两部分组成:源状态和输入字符存储在 TCAM 部分,目的状态及其他相关信息存储在与 TCAM 部分对应的 SRAM 部分。基于 TCAM 的正则表达式匹配算法的查找过程如下:首先,将当前状态和输入字符组合成一个关键字,在 TCAM 中查找与该关键字匹配的表项;然后,根据匹配的表项地址,访问相应的 SRAM 部分,获取迁移的目的状态;最后,将目的状态作为当前状态,读入下一个输入字符,重复执行相似的 TCAM 查找,直至输入字符结束。

已有基于 TCAM 的正则表达式匹配算法主要关注减少 TCAM 存储开销,很少关注减少 TCAM 能耗。TCAM 虽然匹配查找速率快,但是存在能耗高等缺点。在正则表达式匹配中,导致 TCAM 高能耗的原因是:给定一个查找的关键字(即当前状态和输入字符),TCAM 将该关键字与 TCAM 中存储的所有迁移边表项进行并行比较。例如,存储空间大小为 18 MB 的 TCAM 一次查找操作所需能耗高达 15 W<sup>[10]</sup>,这将导致嵌入 TCAM 芯片的路由器、交换机或者其他硬件设备能耗过载。同时,随着网络应用的发展,网络攻击和应用日益复杂化,特征规则集条数不断增多且描述越来越复杂,存储 DFA 所需的 TCAM 存储空间也越来越大,从而导致 TCAM 能耗开销进一步提高。

本文提出一种基于字符索引的正则表达式匹配算法,减少 TCAM 的能耗。该算法的核心思想是:将 TCAM 分为索引块和存储块;DFA 的字母表存储在 TCAM 索引块中,每个字符的索引范围存储在对应的 SRAM 中;状态迁移边的源状态存储在 TCAM 存储块中,目的状态存储在对应的 SRAM 中。该算法的匹配查找过程如下:首先,在 TCAM 索引块中查找当前的输入字符,在对应的 SRAM 中获取相应的索引范围;然后,根据索引范围启动对应的 TCAM 存储块,匹配当前状态,在 SRAM 中获取迁移的目的状态。

本文的主要贡献是:1)提出了基于字符索引的 DFA (CIDFA, character-indexed DFA),减少匹配

时激活的 TCAM 块数,降低 TCAM 能耗;2)针对 Snort 和 Bro 特征规则集开展实验,结果表明,与 DFA 相比,CIDFA 在 TCAM 能耗上平均减少了 92.7%,在 TCAM 存储空间开销上平均减少了 32.0%,在吞吐量上平均提高了 57.9%。

## 2 相关工作

DPI 技术的核心是正则表达式匹配算法,基于硬件的正则表达式匹配算法主要从压缩存储空间等方面开展研究。基于 FPGA 的算法<sup>[11-14]</sup>实现了非确定性有限自动机 (NFA, nondeterministic finite automata),基于 ASIC 的算法<sup>[15-18]</sup>则实现了 DFA。Kumar 等人<sup>[15, 16]</sup>提出了基于 D<sup>2</sup>FA 的正则表达式匹配算法,压缩了 DFA 的迁移边,Smith 等人<sup>[19, 20]</sup>提出了基于 XFA 的正则表达式匹配算法,压缩了 DFA 的状态。在基于 TCAM 的算法方面,Meiners 等人<sup>[8]</sup>提出了共享迁移边、融合状态迁移表等优化算法压缩了 DFA 的存储空间,将 DFA 实现在较小的 TCAM 片上,Peng 等人<sup>[9]</sup>利用 NFA 和 DFA 结构上的关联,提出了链式 DFA 压缩算法,保证匹配速度的同时,压缩了 DFA 的状态空间。在数据分组分类领域,Meiners 等人<sup>[21-23]</sup>提出了多种 TCAM 表项压缩算法,降低 TCAM 的存储空间开销。

TCAM 具有查找速度快、存储空间小等特性,且能耗与存储空间成正比,在部署大规模设备的网络环境中,如何降低 TCAM 能耗成为研究热点。在 IP 查找和数据分组分类领域,研究者提出了多种低能耗的 TCAM 方案,Zane 等人<sup>[24]</sup>提出了适用于 IP 查找的 CoolCAM,将路由表拆分成多个子表,通过将 IP 地址映射到路由表的一个子表,缩小查找范围。Spitznagel 等人<sup>[25]</sup>改进了 CoolCAM 的结构,提出了适用于数据分组分类的 Extended TCAM,将一个索引过滤器与多个 TCAM 存储块相连,限制 TCAM 查找过程中激活的 TCAM 块数。Ma 等人<sup>[10]</sup>提出了 SmartPC,将源 IP 地址和目的 IP 地址组成的地址对作为预分类器,数据分类器中的规则按照地址对的范围,启发式地存储在不同的 TCAM 块中,且每条规则仅存储在一个 TCAM 块中。SmartPC 仅需激活预分类器和部分 TCAM 块,即可完成规则匹配。SmartPC 在能耗上平均减少了 88%,索引开销小于 TCAM 存储空间的 4%,是目前所知最优的低能耗分组分类算法。

与上述低能耗 TCAM 方案不同,本文第一次提

出基于 TCAM 的低能耗正则表达式匹配算法。本文的方法将 TCAM 分为索引块和存储块，DFA 的字母表存储在 TCAM 索引块中，源状态存储在 TCAM 存储块中。匹配查找过程时，首先在 TCAM 索引块中查找当前读入的字符，根据 SRAM 中返回的索引范围，启动对应的 TCAM 存储块，完成查找。本文的方法不仅降低 TCAM 能耗，而且减少 DFA 所需的存储空间，同时提高匹配吞吐量。

### 3 基于字符索引的正则表达式匹配算法

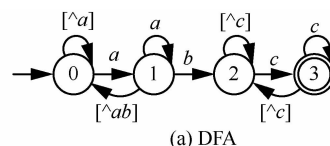
#### 3.1 基于 TCAM 的正则表达式匹配

对于给定的特征规则集，在 TCAM 上实现正则表达式匹配的步骤如下：首先，构造特征规则集的 DFA；然后，构建 DFA 的 TCAM 表，即将 DFA 的每条状态迁移边存储在一条 TCAM 表项中。每条表项由 TCAM 和 SRAM 两部分组成：源状态和输入字符存储在 TCAM 部分，目的状态存储在与 TCAM 部分对应的 SRAM 部分。该算法的匹配查找过程如下：首先，将当前状态和输入字符组合成一个关键字，启动 TCAM 的全部表项查找该关键字，返回与该关键字匹配的第一条表项的地址；然后，根据匹配表项的地址，获取 SRAM 中存储的目的状态；最后，将目的状态作为当前状态，读入下一个输入字符，重复执行相似的 TCAM 查找，直至输入字符结束。

图 1 给出了特征规则集  $\{.*ab.*c\}$  的 DFA 和 TCAM 表，图 1(a)中，初始状态为 0，匹配状态为 3，“ $[\wedge]$ ”表示非字符子集， $[\wedge ab]$ 表示除了  $a$  和  $b$  以外的其他输入字符。图 1(b)中，TCAM 表的表项总数与 DFA 的迁移边条数相同，源状态和目的状态对应状态迁移表中状态的二进制编码，输入字符为三态编码，特征规则集中出现的输入字符为该字符的二进制 ASCII 码，其他字符用“\*\*\*\*\*”表示，“\*\*\*\*\*”可匹配除特征规则集中字符之外的任意输入字符。

#### 3.2 TCAM 分块存储和字符索引

TCAM 是分块存储的，每个 TCAM 块中包含了固定数目的 TCAM 表项。TCAM 提供部分激活的机制，即查找过程中可以仅启动 TCAM 中的部分 TCAM 块，而不必启动整个 TCAM。利用 TCAM 部分激活机制的前提是：必须保证在激活的少数 TCAM 块中进行准确无误的查找。本文通过字符索引达到这一目的。



TCAM		SRAM
源状态	输入字符	目的状态
00	01100001	01
00	01100010	00
00	01100011	00
00	*****	00
01	01100001	01
01	01100010	10
01	01100011	00
01	*****	00
10	01100001	10
10	01100010	10
10	01100011	11
10	*****	10
11	01100001	10
11	01100010	10
11	01100011	11
11	*****	10

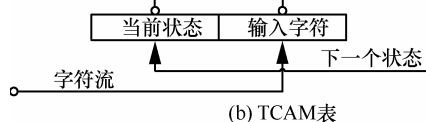


图 1 特征规则集  $\{.*ab.*c\}$  的 DFA 和 TCAM 表

算法 1 给出了 TCAM 分块存储和构建字符索引的伪代码。其中，*TransitionTable* 表示特征规则集的状态迁移表，*TCAMTable* 表示特征规则集对应的 TCAM 表，*StorBlock* 表示存储块，*IndexBlock* 表示索引块，*blockSize* 表示每个 TCAM 块的大小。第 1)~3)行代码是由状态迁移表构建 TCAM 表的步骤，即将状态迁移表的每条状态迁移边保存在 TCAM 表中，每条迁移边由三元组成：源状态 *srcID*、输入字符 *input* 和目的状态 *destID*。第 4)~10)行代码为构建存储块的步骤，首先，将 TCAM 表中的表项按照输入字符升序排列，计算输入字符相同的 TCAM 表项数，赋值给变量 *minStorEntryNum*，根据 TCAM 块的大小，计算每个输入字符所需的存储块数，赋值给变量 *storBlockNum*；然后，将 *srcID* 和 *destID* 组合为存储块表项，按照 TCAM 块的大小构建存储块。由于 *storBlockNum* 存储块中包含的表项总数大于或等于 *minStorEntryNum*，故在最后一个存储块中可能存在未使用的 TCAM 表项，本文采用无关位（“\*”）填满这些表项，保证 TCAM 中没有空白的表项。第 11)~15)行代码表示构建索引块的步骤，首先，将 *storBlockNum* 个存储块的编号 *storBlockID* 存储在每个输入字符 *input* 的索引范围 *indexRange* 中；然后，按照 TCAM 块的大小构建索引块。

**算法 1** BlockOrganized(*TransitionTable*)

```

1) for each transition in TransitionTable do
2)   TCAMTable PushBack (srcID, input, destID);
3) end for
4) for each entry in TCAMTable do
5)   Sort(entry in ascending order of input);
6)   minStorEntryNum = Count(entry with same
   input);
7)   storBlockNum = Ceil(minStorEntryNum / block-
   Size);
8)   StorEntry PushBack(srcID, destID);
9) end for
10) StorBlock = Partition(StorEntry srcID, block-
   Size);
11) for each input in alphabetTable do
12)   indexRange Insert(storBlockID according to
   storBlockNum);
13) IndexEntry PushBack(input, indexRange);
14) end for
15) IndexBlock = Partition (IndexEntry input,
   blockSize)

```

图 2 给出了图 1 应用本文算法后的结构示意图，其中，TCAM 块的大小为 4，TCAM 索引块保存 TCAM 存储块编号的二进制编码，用于指示每个字符对应的存储块范围；TCAM 存储块保存源状态的二进制编码，对应的 SRAM 部分保存目的状态的二进制编码。

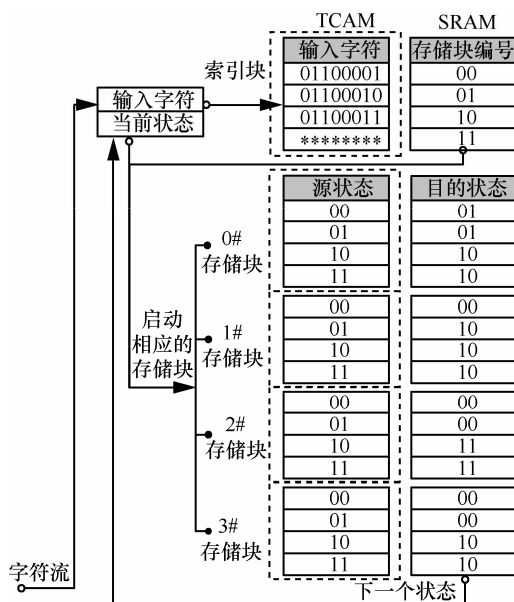


图 2 特征规则集{.\*ab.\*c}的分块存储和字符索引

基于字符索引的正则表达式匹配算法的查找过程如下：首先，在 TCAM 索引块中查找当前的输入字符，返回 SRAM 中的存储块编号；然后，根据存储块编号启动对应的 TCAM 存储块，匹配当前状态，在 SRAM 中获取迁移的目的状态；最后，将目的状态作为当前状态，读入下一个输入字符，重复执行相似的 TCAM 查找，直至输入字符结束。

比较图 2 和图 1 可知，本文的算法将基于 TCAM 的正则表达式匹配分为 2 个阶段：第 1 个阶段查找 TCAM 索引块，第 2 个阶段查找 TCAM 存储块。本文的算法在能耗、存储空间、吞吐量等重要指标上的变化将在下一节汇总详细讨论。

**4 实验评估**

在正则表达式处理软件<sup>[26]</sup>的基础上，本文采用 C/C++ 设计实现了 DFA 和 CIDFA，并运行在 CPU 为 Intel Xeon X5506、内存为 16 GB 的计算机上。在软件模拟实验中，本文采用 TCAM 仿真工具<sup>[27]</sup>模拟 DFA 和 CIDFA 的 TCAM 能耗、存储空间开销和吞吐量等性能指标。

**4.1 实验方法**

本文处理的特征规则集来自于 Snort 和 Bro 特征库，其中，R1~R9 取自 Snort 特征库，R10 取自 Bro 特征库。特征规则集的参数如表 1 所示，其中，通配符表达式的主要形式为{\*子表达式}\*{子表达式}，“.\*”表示匹配任意数目的输入字符。计数器表达式的主要形式为{[c<sub>1</sub>...c<sub>n</sub>]{n}}，“[c<sub>1</sub>...c<sub>n</sub>]”表示匹配输入字符为“c<sub>1</sub>”至“c<sub>n</sub>”中的任意一个，“{n}”表示重复出现 n 次。由特征规则集生成 DFA 时，通配符表达式容易引起 DFA 状态空间急剧增大<sup>[20]</sup>，导致存储自动机所需的 TCAM 空间开销增加，能耗也随之增大。

表 1 特征规则集的参数统计

特征规则集	规则条数	字母表大小	通配符表达式条数
R1	31	88	7
R2	23	81	2
R3	33	78	1
R4	33	78	14
R5	33	78	33
R6	31	88	15
R7	23	81	11
R8	23	81	23
R9	31	88	31
R10	217	110	18

本文采用的特征规则集中，通符合表达式条数占规则总数的 1.3%~42.3%，DFA 状态的规模为 5 000~100 000 个，存储状态迁移表的 TCAM 包含 400 000~8 000 000 条表项，能够模拟不同规模的 DFA 在本文算法下性能的改变。

本文采用线性模型计算 TCAM 能耗，即 TCAM 能耗为启动的 TCAM 块数与每个 TCAM 块能耗的乘积。影响 TCAM 能耗的一个重要参数是 TCAM 块的大小，这与硬件结构有关，本文采用 32、64、128、256、512 和 1 024 等 6 种不同大小的 TCAM 块，模拟 TCAM 能耗。给定一个包含  $N$  条状态迁移表的 DFA 和大小为  $R$  的 TCAM 块，DFA 需要启动全部的 TCAM 块，启动的 TCAM 块数为  $X = \lceil \frac{N}{R} \rceil$ ，DFA 的 TCAM 能耗为  $X$  个 TCAM 块的能耗之和。对于 CIDFA，假定索引块包含  $C$  条表项，与索引对应的存储块包含  $M$  条表项，索引块需要的 TCAM 块数为  $Y_1 = \lceil C/R \rceil$ ，存储块需要的 TCAM 块为  $Y_2 = \lceil M/R \rceil$ ，CIDFA 的 TCAM 能耗为  $(Y_1 + Y_2)$  个 TCAM 块的能耗之和。

TCAM 存储空间的大小为 TCAM 宽度与表项数的乘积，其中，表项数由 TCAM 块的大小与 TCAM 块数的乘积决定。DFA 占用的 TCAM 宽度为源状态和输入字符占用的比特数之和。本文采用的特征规则集的字母表为 ASCII 码，输入字符占用的比特数为 8 bit，对于状态数为  $S$  的 DFA，源状态占用的比特数为，故 DFA 占用的 TCAM 宽度为  $(W+8)$ bit，其 TCAM 存储空间开销为  $W = \lceil \lg S \rceil$ 。对于 CIDFA，索引块的 TCAM 宽度为输入字符的比特数，即 8 bit，存储块的 TCAM 宽度为源状态占用的比特数，即  $W$  bit，CIDFA 的 TCAM 存储空间开销  $S_2 = 8RY_1 + WRY_2$ 。

本文以吞吐量衡量算法的匹配速率。吞吐量为每次读入字符的比特数与 TCAM 查找时延之商，对于单步长正则表达式匹配算法，每次读入一个 ASCII 字符，为 8 bit。假定上述  $X$ 、 $Y_1$  和  $Y_2$  个 TCAM 块的查找时延分别为  $D_x$ 、 $D_{y1}$  和  $D_{y2}$ ，DFA 的吞吐量为  $8/D_x$ ，CIDFA 的吞吐量为  $8/(D_{y1} + D_{y2})$ 。

#### 4.2 能耗

本文采用 32、64、128、256、512 和 1 024 等 6 种不同大小的 TCAM 块，模拟 DFA 和 CIDFA 的 TCAM 能耗，并计算 CIDFA 在 TCAM 能耗上减少的比例。假定 DFA 的 TCAM 能耗为  $P_1$ ，CIDFA 的

TCAM 能耗为  $P_2$ ，CIDFA 在 TCAM 能耗上减少的比例为  $\frac{P_1 - P_2}{P_1} \times 100\%$ 。

图 3 给出了 10 个特征规则集在 TCAM 能耗上减少的比例。不同 TCAM 块大小下，CIDFA 在 TCAM 能耗上减少的比例为 78.8%~98.5%，平均减少比例为 92.7%。当 TCAM 块大小为 32 时，CIDFA 在 TCAM 能耗上减少的比例最小，为 8.8%~84.8%，平均减少比例为 80.3%。随着 TCAM 块大小的增加，CIDFA 在 TCAM 能耗上减少的比例不断提高，当 TCAM 块大小为 1 024 时，CIDFA 在 TCAM 能耗上减少的比例达到最大，为 98.0%~98.5%，平均减少比例为 98.2%。

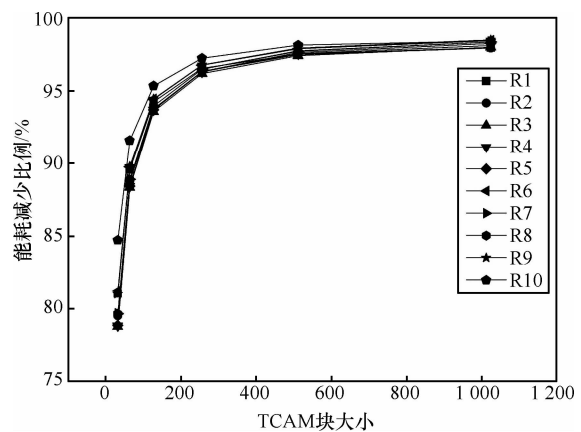


图 3 能耗减少比例

图 4 给出了当 TCAM 块大小为 64、256 和 1 024 时，DFA 与 CIDFA 的 TCAM 能耗比较。不同 TCAM 块大小下，同一特征规则集对应的 DFA 的 TCAM 能耗基本一致；随着 TCAM 块大小的增加，同一特征规则集对应的 CIDFA 的 TCAM 能耗不断减少。

#### 4.3 存储空间开销和吞吐量

本文采用 4.1 节中介绍的方法计算 DFA 和 CIDFA 的 TCAM 存储空间开销和吞吐量。

在存储空间方面，DFA 的 TCAM 片上保存状态迁移边的源状态和输入字符。CIDFA 额外开销了一部分 TCAM 空间作为索引块，但是，TCAM 片上仅保存状态迁移边的源状态。

表 2 给出了 DFA 和 CIDFA 的 TCAM 存储空间开销比较。与 DFA 相比，CIDFA 在 TCAM 存储空间开销上平均减少了 32.0%。对于不同大小的 TCAM 块，同一特征规则集对应的 DFA 和 CIDFA 的 TCAM 存储空间开销分别相同。

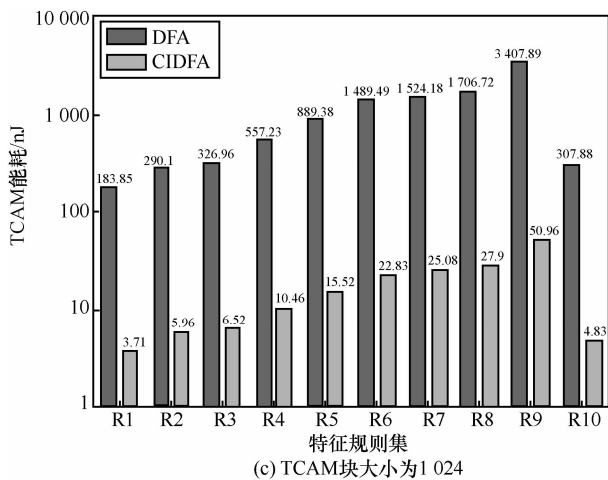
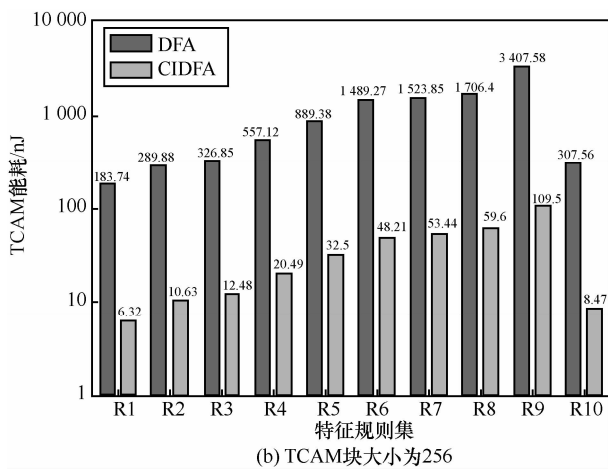
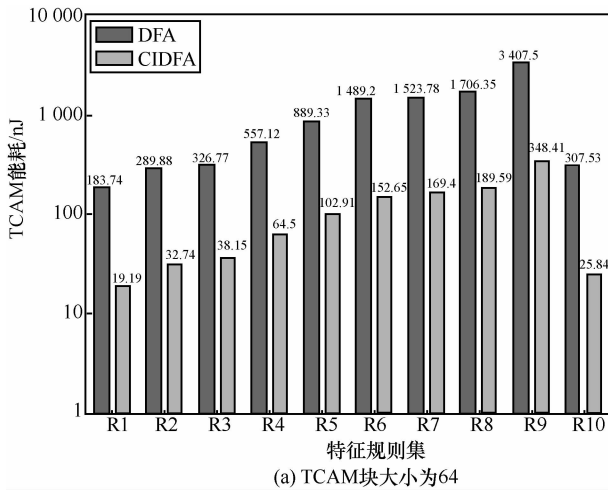


图 4 不同 TCAM 块大小下，DFA 与 CIDFA 的 TCAM 能耗比较

吞吐量决定了算法的匹配速率，由于 DFA 和 CIDFA 每次匹配操作处理的比特数相同，故算法的吞吐量由匹配时的查找时延决定。DFA 的查找时延为启动全部 TCAM 块完成查找操作的时间，CIDFA 的查找时延为启动 TCAM 索引块和 TCAM 存储块完成查找的时间之和。

表 2 DFA 和 CIDFA 的 TCAM 存储空间开销比较

特征规则集	DFA/MB	CIDFA/MB
R1	10.32	7.02
R2	16.30	11.09
R3	18.38	12.50
R4	31.34	21.32
R5	50.05	34.03
R6	83.83	57.00
R7	85.77	58.33
R8	96.05	65.32
R9	191.84	130.45
R10	17.29	11.76

表 3 给出了 DFA 和 CIDFA 的吞吐量比较。与 DFA 相比，CIDFA 在吞吐量上平均提高了 57.9%。对于不同大小的 TCAM 块，同一特征规则集对应的 DFA 吞吐量相同；随着 TCAM 块大小的增加，同一特征规则集对应的 CIDFA 吞吐量不断下降。当 TCAM 块大小为 32，CIDFA 的吞吐量达到最大，比 DFA 提高 50.3%~99.5%；当 TCAM 块大小为 1024 时，CIDFA 的吞吐量最低，比 DFA 提高 36.6%~70.9%。

表 3 DFA 和 CIDFA 的吞吐量比较

特征规则集	DFA/(Gbit·s <sup>-1</sup> )	CIDFA/(Gbit·s <sup>-1</sup> )	
		最小	最大
R1	1.98	3.11	3.63
R2	1.82	2.72	3.11
R3	1.82	2.72	3.11
R4	1.68	2.42	2.72
R5	1.56	2.42	2.72
R6	1.56	2.18	2.42
R7	1.56	2.18	2.42
R8	1.56	2.18	2.42
R9	1.45	1.98	2.18
R10	1.82	3.11	3.63

## 5 结束语

本文提出了一种基于字符索引的正则表达式匹配算法，该算法将 TCAM 分为索引块和存储块，其中，索引块中保存 DFA 的字母表，存储块中保存 DFA 的源状态。匹配时，首先在索引块中查找当前读入的字符，然后，仅启动与读入字符对应的部分存储块，即可完成查找操作。该算法能够避免启动

全部的存储块,从而达到降低能耗的目的。

针对实际特征规则集的实验表明:与 DFA 相比, CIDFA 在能耗上平均减少了 92.7%,在存储空间平均开销上平均减少了 32.0%,在吞吐量上平均提高了 57.9%。因此, CIDFA 是一种低能耗的高效 TCAM 方案,不仅确保查找速率,而且显著降低 TCAM 能耗和存储空间。

本文的算法可应用于数据中心网络中大规模流量识别技术,有助于减少数据中心网络中成千上万台交换机的能耗,下一步的研究方向是实现基于 TCAM 的多步长正则表达式匹配算法。

### 参考文献:

- [1] PAXSON V, ASANOVIC K, DHARMAPURIKAR S, *et al.* Rethinking hardware support for network analysis and intrusion prevention[A]. Proceeding of USENIX Workshop on Hot Topics in Security[C]. Vancouver, Canada, 2006.
- [2] SOMMER R, PAXSON V. Enhancing bytelevel network intrusion detection signatures with context[A]. Proceeding of ACM Conference on Computer and Communications security[C]. Washington, DC, USA, 2003, 262-271.
- [3] Snort::rules[EB/OL]. <http://www.snort.org/start/rules>, 2013-12-01.
- [4] The bro network security monitor[EB/OL]. <http://bro-ids.org>, 2013.
- [5] HP TippingPoint S1200N IPS A7500 module[EB/OL]. [http://h17007.www1.hp.com/us/en/products/network-security/HP\\_TippingPoint\\_S1200N\\_IPS\\_A7500\\_Module/index.aspx?jumpid=reg\\_r1002\\_usen](http://h17007.www1.hp.com/us/en/products/network-security/HP_TippingPoint_S1200N_IPS_A7500_Module/index.aspx?jumpid=reg_r1002_usen), 2013-12-01.
- [6] Intrusion prevention system(IPS)-Cisco systems[EB/OL]. [http://www.cisco.com/en/US/products/ps5729/Products\\_Sub\\_Category\\_Home.html](http://www.cisco.com/en/US/products/ps5729/Products_Sub_Category_Home.html), 2013-12-05.
- [7] YU F, KATZ R, LAKSHMAN T V. Gigabit rate packet pattern-matching using TCAM[A]. Proceeding of IEEE International Conference on Network Protocols[C]. Berlin, Germany, 2004.
- [8] MEINERS C R, PATEL J, NORIGE E, *et al.* Fast regular expression matching using small TCAMs for network intrusion detection and prevention systems[A]. Proceeding of the 19th USENIX Security Symposium[C]. Washington, DC, USA, 2010, 8.
- [9] PENG K, TANG S, CHEN M, *et al.* Chain-based DFA deflation for fast and scalable regular expression matching using TCAM[A]. Proceeding of ACM/IEEE Symposium on Architectures for Networking and Communications Systems[C]. Brooklyn, NY, USA, 2011.
- [10] MA Y, BANERJEE S. A smart pre-classifier to reduce power consumption of TCAMS for multi-dimensional packet classification[A]. Proceeding of ACM Conference of the Special Interest Group on Data Communication[C]. Helsinki, Finland, 2012.
- [11] SIDHU R, PRASANNA V K. Fast regular expression matching using FPGA[A]. Proceeding of IEEE International Symposium on Field-Programmable Custom Computing Machines[C]. Rohnert Park, CA, USA, 2001.
- [12] CLARK C R, SCHIMMEL D E. Scalable pattern matching on high-speed networks[A]. Proceeding of IEEE International Symposium on Field-Programmable Custom Computing Machines[C]. Napa Valley, CA, USA, 2004.
- [13] SOURDIS I, PNEVMATIKATOS D. Pre-decoded CAMs for efficient and high-speed NIDS pattern matching[A]. Proceeding of IEEE International Symposium on Field-Programmable Custom Computing Machines[C]. Napa Valley, CA, USA, 2004.
- [14] MOSCOLA J, LOCKWOOD J, LOUI R P, *et al.* Implementation of a content-scanning module for an internet firewall[A]. Proceeding of IEEE International Symposium on Field-Programmable Custom Computing Machines[C]. Napa Valley, CA, USA, 2003.
- [15] KUMAR S, DHARMAPURIKAR S, YU F, *et al.* Algorithms to accelerate multiple regular expressions matching for deep packet inspection[A]. Proceeding of ACM Conference of the Special Interest Group on Data Communication[C]. Pisa, Italy, 2006.
- [16] KUMAR S, TURNER J, WILLIAMS J. Advanced algorithms for fast and scalable deep packet inspection[A]. Proceeding of ACM/IEEE Symposium on Architectures for Networking and Communications Systems[C]. San Jose, CA, USA, 2006.
- [17] BECCHI M, CADAMI S. Memory-efficient regular expression search using stae merging[A]. Proceeding of IEEE International Conference on Computer Communications[C]. Anchorage, Alaska, USA, 2007.
- [18] BECCHI M, CROWLEY P. An improved algorithm to accelerate regular expression evaluation[A]. Proceeding of ACM/IEEE Symposium on Architectures for Networking and Communications Systems[C]. San Jose, CA, USA, 2008.
- [19] SMITH R, ESTAN C, JHA S. XFA: faster signature matching with extended automata[A]. Proceeding of IEEE Symposium on Security and Privacy[C]. Oakland, CA, USA, 2008.
- [20] SMITH R, ESTAN C, JHA S, *et al.* Deflating the big bang: fast and scalable deep packet inspection with extended finite automata[A]. Proceeding of ACM Conference of the Special Interest Group on Data Communication[C]. Seattle, WA, USA, 2008.
- [21] MEINERS C R, LIU A X, TORNG E. TCAM Razor: a systematic approach towards minimizing packet classifiers in TCAMs[A]. Proceeding of IEEE International Conference on Network Protocols[C]. Beijing, China, 2007.
- [22] LIU A X, MEINERS C R, ZHOU Y. All-match based complete redundancy removal for packet classifiers in TCAMs[A]. Proceeding of IEEE International Conference on Computer Communications[C]. Phoenix, AZ, USA, 2008.
- [23] MEINERS C R, LIU A X, TORNG E. Bit weaving: a non-prefix approach to compressing packet classifiers in TCAMs[A]. Proceeding of IEEE International Conference on Network Protocols[C]. Princeton, NJ, USA, 2009.
- [24] ZANE F, NARLIKAR G, BASU A. CoolCAMs: power-efficient TCAMs for forwarding engines[A]. Proceeding of IEEE International Conference on Computer Communications[C]. San Francisco, USA, 2003.
- [25] SPITZNAGEL E, TAYLOR D, TURNER J. Packet classification using extended TCAMs[A]. Proceeding of IEEE International Conference on Network Protocols[C]. Atlanta, Georgia, USA, 2003.

IEEE, 2011,99(3):490-512.

- [5] BARROS J, COSTA R A, MUNUARETTO D, *et al.* Effective delay control in online network coding[A]. The 28th Conference on Computer Communications, IEEE INFOCOM[C]. Brazil, 2009. 208-216.
- [6] BRAMKO L S, O'MALLEY S W, PETERSON L L. TCP vegas: new techniques for congestion detection and avoidance[A]. Proceedings of the SIGCOMM '94[C]. Symposium, 1994.
- [7] LUN D S, MEDARD M, EFFROS M. On coding for reliable communication over packet networks[J]. Physical Communication, 2008,1(1): 3-20.
- [8] XIA Y, SUBRAMANIAR L, STOICA I, *et al.* One more bit is enough[J]. IEEE Transaction in IEEE/ACM Trans Networking, 2008, 16(6):1281-1294.
- [9] QAZI I A, ZUAI T. On the design of load factor based congestion control protocols for next-generation networks[A]. The 27th Conference on Computer Communications[C]. Phoenix, AZ, 2008. 96-100.
- [10] CHEN J, TAN W, LIU L X, *et al.* Towards zero loss for TCP in wireless networks[A]. Performance Computing and Communications Conference (IPCCC)[C]. Scottsdale, AZ, 2009. 65-70.
- [11] NS-2 Network Simulator[EB/OL]. <http://www.isi.edu/nsnam/ns/>.

作者简介:



陈静 (1983-), 女, 江西定南人, 博士, 科学技术部信息中心助理研究员, 主要研究方向为计算机网络、电子政务。



宋学鹏 (1977-), 男, 河北邢台人, 博士, 科学技术部信息中心高级工程师, 主要研究方向为电子信息科学与技术。



刘芳 (1982-), 女, 内蒙古鄂尔多斯人, 博士, 中国科学院副研究员, 主要研究方向为基于GPU的通用计算和真实感绘制算法。

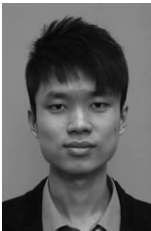
.....  
(上接第 168 页)

- [26] Regular expression processor[EB/OL]. [http://regex.wustl.edu/index.php/Main\\_Page](http://regex.wustl.edu/index.php/Main_Page), 2013-12-01.
- [27] AGRAWAL B, SHERWOOD T. Modeling TCAM power for next generation network devices[A]. Proceeding of IEEE International Symposium on Performance Analysis of Systems and Software[C]. Austin, TX, USA, 2006.



黄昆 (1978-), 男, 江西永丰人, 博士, 中国科学院助理研究员, 主要研究方向为未来互联网、网络安全。

作者简介:



丁麟轩 (1988-), 男, 山东烟台人, 湖南大学硕士生, 主要研究方向为网络安全。



张大方 [通信作者] (1959-), 男, 上海人, 博士, 湖南大学教授、博士生导师, 主要研究方向为可信系统与网络、网络安全、软件工程。E-mail:dfzhang@hnu.edu.cn。