

基于 TCAM 的 K 步长多模式匹配算法及硬件实现

赵海斌, 李训根, 王奇敏

(杭州电子科技大学 微电子 CAD 研究所, 浙江 杭州 310018)

摘 要: 对 K 步长状态机进行改进, 消除了各状态的失效链, 节省了存储资源, 提高了匹配效率; 根据 TCAM(Ternary Content Addressable Memory)的并行查找和精确查找的特性, 设计了一种用 FPGA 实现 TCAM 的方法, 并以此作为基本匹配单元; 最后结合改进的 K 步长状态机和基本匹配单元设计了一个快速多模式匹配系统. 接收网络数据流进行实验, 结果证明设计的匹配系统能达到数千兆位的吞吐率.

关键词: 多模式匹配; K 步长状态机; 三态内容寻址存储器; 现场可编程门阵列

中图分类号: TP393.08

文献标识码: A

文章编号: 1000-7180(2013)04-0046-04

TCAM—Based K—Step Multi—Pattern Matching Algorithm and Hardware Implementation Research

ZHAO Hai-bin, LI Xun-gen, WANG Qi-min

(IC CAD Research Center, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: Through the improvement of K—step state machine, failure chain of every state is removed which saves storage resources and improves the matching efficiency. Besides, combined with the characteristics of parallel searching and accurate searching of TCAM(Ternary Content Addressable Memory), FPGA is used to implement TCAM as the basic matching cell, which the matching speed is significantly improved. Finally a fast multi—pattern matching system is designed based on improved K—step state machine and the basic matching cell. Networkstream is received for experiment. The result of the experiment is proved that the matching system can achieve multi—gigabit throughput.

Key words: multi—pattern matching; K—step state machine; TCAM; FPGA

1 引言

随着网络的高速发展和普及, 网络安全问题日益突出. 模式匹配是解决网络信息安全的核心方法, 对它的研究不断深入, 从最初的单模式匹配如 KMP 算法、Commentz—Walter 算法和 BM 算法, 到后来的多模式匹配如 AC(Aho—Corasick)算法和 Wu—Manber 算法. 目前模式匹配系统大都是用软件实现的, 由于受处理器速度和软件串行处理特性的限制, 无法实现高速并行匹配, 而且由于描述攻击行为的特征数目急剧增加, 使匹配速度变慢, 传统的基于软件的模式匹配已成为入侵检测技术发展的瓶颈, 直接影响系统的实时性和安全性. 硬件由于处理速

度快以及可并行等优点, 近年来被广泛应用于模式匹配. 采用的硬件技术包括 ASIC 和可编程逻辑 FPGA, 而 FPGA 与 ASIC 相比, 设计更灵活, 可重构, 设计周期更短, 成本更低, 应用更广泛.

TCAM 是一种基于内容寻址的存储器, 可以在 1~2 个时钟周期内给出查找结果, 是快速多模式匹配研究中的热点, 但 TCAM 芯片存在以下缺点: (1) 容量小, 数据端口宽度有限, 配置不够灵活; (2) 作为一种片外存储器, 访问时延较大, 对他的频繁访问势必会造成系统整体性能的下降. 本文对 K 步长状态机^[1]进行了改进, 消除了各状态的失效链, 用 FPGA 实现 TCAM 的方法, 最后设计了一个快速高效的多模式匹配系统, 实验结果表明系统的有效性.

2 基本原理

2.1 K 步长多模式匹配算法的改进

AC 算法提出了 FSM(Finite State Machine)的思想,把所有的规则构造成一个有限自动机,整个匹配过程都在 FSM 中跳转,既解决了规则长度的问题,也解决了匹配过程中的回溯问题,匹配的速度和效率大大提高,但这种算法每个周期只能处理一个字符,且跳转太过频繁,需不停的访问外部存储器中的 AC 状态机。为此采用 K 步长状态机,每个周期可以处理 K 个字符^[1],以 $K=4$ 为例,根据规则 $s1 \sim s6$ 构造的状态机如图 1 所示。 $q0$ 到 $q6$ 表示状态, $s1$ 到 $s6$ 分别表示 6 条规则,“*”表示任意字符。由于规则长度不一定是 K 的整数倍,在构造 K 步长状态机时会在规则尾部剩下 $n(n < K)$ 个字符,补充 m 个“*”,使 $n+m=K$ 。图 1 中“e****”表示:第一个字符是“e”,后面是三个任意的字符“***”。以规则“technical”为例,由于 AC 算法每次跳转只能处理一个字符,共需 9 次跳转才能完成匹配;而用图 1 的 K 步长状态机只需 3 次跳转即可完成匹配,大大缩短了匹配的时间。

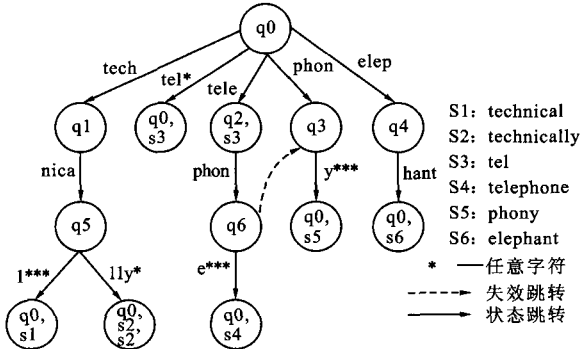


图 1 K 步长状态机

AC 算法中要计算各个状态的失效函数, K 步长状态机也存在失效状态的问题,图 1 中 $q6$ 的失效状态为 $q3$ (见虚线箭头),其他状态的失效状态都为 $q0$ (未标出),为此必须开辟一定的存储空间来存储每个状态的失效状态。另外,由图 1 得出 $q6$ 的失效链为 $q6 \rightarrow q3 \rightarrow q0$,即当前输入字符在状态 $q6$ 处匹配失败后,还要转到 $q3$ 处继续匹配,如在 $q3$ 处也不匹配,继续失效到 $q0$ 状态。除了要花费资源来存储每个状态的失效状态,失效链的遍历也降低了匹配的效率。对图 1 的 K 步长状态机进行改进。

输入 s 在状态 q 处匹配失败后,依次遍历 q 的失效链:

(1) 如果 s 在失效链中的 $qa (qa \neq q0)$ 处产生匹

配,能跳转到状态 qb 处,则在状态 q 处直接建立经 s 到状态 qb 的分支;

(2) 如果依次遍历了失效链中除了 $q0$ 之外的所有状态后,都不能产生匹配,则状态 q 的最终失效状态就是 $q0$ 。

改进后的 K 步长状态机如图 2 所示,图中所有状态的失效状态都为 $q0$,不用再存储失效状态,节省了存储资源;而且 $q6$ 的失效链已被消除,如果当前状态为 $q6$,当前输入为“yabc”,则 $q6$ 可直接经“y****”跳转到 $(q0, s5)$,不用再经过 $q3$ 。改进后省去了存储失效状态的资源开销,也提高了匹配效率,且失效链越长,效果越明显。

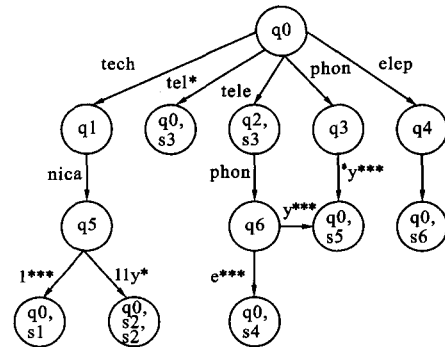


图 2 改进后的 K 步长状态机

2.2 TCAM 的设计与实现

为了实现 $TCAM256 \times 4bit$ (256 个地址, 4bit 的数据)单元,把 FPGA 内部的 M4K 配置成双端口网络^[2],如图 3 所示。写端口配置为 4096×1 ,用于 TCAM 的初始化;读端口配置为 16×256 ,用于 TCAM 的查找;掩码寄存器 $Mask[255:0]$ 用来实现三态存储。

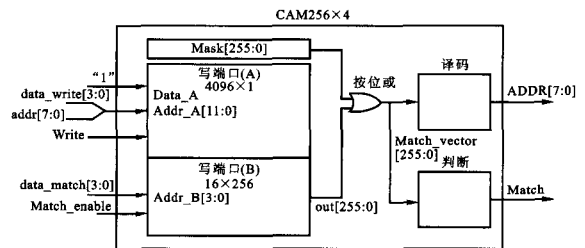


图 3 $TCAM256 \times 4bit$ 单元

写端口中的 $data_write[3:0]$ 是 $Addr_A$ 的高 4 位,用于寻址图 4 中的 16 行; $addr[7:0]$ 是 $Addr_A$ 的低 8 位,用于寻址 256 列,图 4 中 256 列代表 256 个地址。读端口中,待查找数据 $data_match$ 直接用来寻址,输出为 $out[255:0]$,每位代表一个地址。两个 $TCAM256 \times 4bit$ 单元可以扩展成一个 $TCAM256 \times 8bit$,一个负责高 4 位的查找,另一个负责低 4 位的查找。

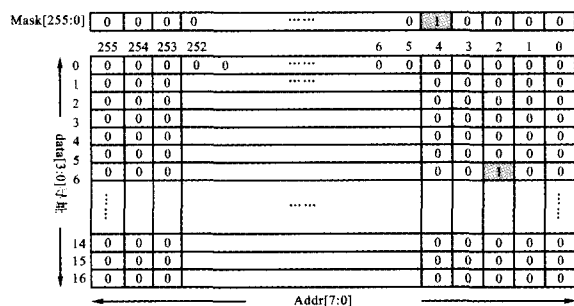


图4 TCAM256×4bit的内部结构

TCAM的初始化过程就是模式字符的写入过程,写入TCAM的数据除了一般的字符外,还有可能是任意字符“*”。下面分别举例说明:当 $data_write=0110$,写入地址2中,在Write信号和时钟信号的控制下,地址 $Addr_A=0110_0000_0010$ 中被写入1,见图4中第6行第2列的阴影部分;当 $data_write$ 为“*”,写入地址为4时,表示地址4中要写入任意字符,即地址4对应的那一列的所有16行中都要写入1,见图4,所以就需要16次的写操作,为了避免这一繁琐操作,设计一个掩码寄存器 $Mask[255:0]$,将地址4对应的 $Mask[4]$ 置为1,见图4里Mask中的阴影部分。

TCAM执行查找功能时,令 $Addr_B=data_match$,读端口输出为 $out[255:0]$,匹配向量 $Match_vector = Mask \mid out$,如果匹配向量中的某一位为1,说明在该位对应的地址中查找到了 $data_match$,把 $Match_vector$ 译码后,得到对应的地址 $ADDR[7:0]$ 和Match信号,再以 $ADDR$ 去寻址外部RAM。大规模的TCAM可以通过TCAM256×4bit单元进行扩展得到。

3 匹配系统设计

由于规则集中模式串的长度差异很大,而TCAM数据端口的宽度有限,如果针对每种长度分别构建TCAM是不现实的。把K步长状态机和TCAM结合起来就解决了上述问题,因为K步长状态机将所有模式串截成固定长度的小段,只需构建端口宽度固定的TCAM即可。

3.1 系统框图和匹配过程

根据K步长状态机,结合TCAM的特性,设计了一个模式匹配系统^[3-5]。其整体结构如图5所示。在FPGA内部实现网络接口模块、数据缓冲区FIFO、TCAM和匹配控制逻辑。网络上的数据流经过网络接口模块接收后,存放在FIFO中等待匹配;匹配控制逻辑用来更新当前状态,并读出FIFO中的待匹配

数据做为“当前输入”;TCAM用于并行查找。

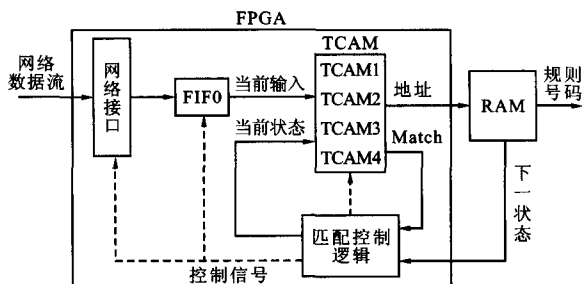


图5 系统框图

在进行匹配之前要初始化TCAM和外部RAM,根据图6的存储结构,在TCAM中写入当前状态和当前输入,然后在RAM的对应地址中写入下一状态和规则号码。匹配过程中,把“当前状态+当前输入”作为TCAM的输入,如果TCAM中存储了和输入相同的数据,就说明匹配,输出Match信号和该输入在TCAM中的地址,以这个地址去寻址RAM,读出“下一状态+规则号码”,然后匹配控制逻辑及时的更新当前状态和当前输入。比如当TCAM的输入为“ $q4+hant$ ”时,TCAM输出“地址1”,然后用“地址1”去读RAM,得到“ $q0+s6$ ”,如图6所示,然后匹配控制逻辑把当前状态更新为 $q0$,并读FIFO来更新当前输入。

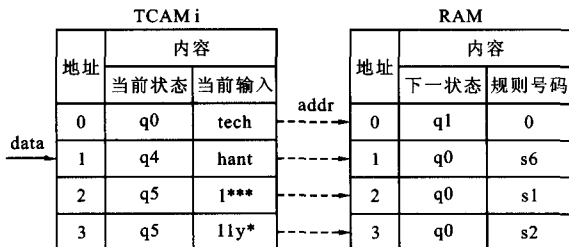


图6 K步长状态机的存储享

用K步长状态机进行匹配时,为了在匹配过程中不漏检,必须采用K路并行匹配,且相邻的两路要错开一个字符^[1,6],如图7所示。

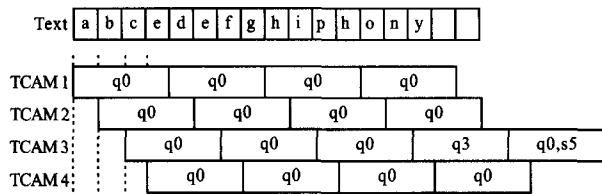


图7 四路并行匹配(K=4)

3.2 匹配控制模块——流水线

根据3.1中介绍的匹配过程,TCAM并行查找需要1个时钟周期,根据TCAM的查找结果读外部RAM需要2个时钟周期,更新当前状态和当前数据需要1个时钟周期,所以从数据进入匹配系统到

得到匹配结果,共需要 4 个时钟周期.而且在实际的应用中,绝大多数的网络数据是安全的,不会发生匹配.为了提高系统的吞吐率,在匹配控制模块中采用流水线技术,让待匹配数据流水式的进入匹配系统,不用等待 Match 信号,但要把待匹配数据逐级寄存(寄存的级数等于流水线的级数).一旦出现 Match 信号,流水线停止,然后逐个匹配寄存的待匹配数据,等寄存的待匹配数据都匹配完,再启动流水线.

4 实验分析

用 QuartusII9.0 完成了 RTL 级代码的编写,在 Modelsim 下进行了仿真,仿真结果正确.下载到 Altare FPGA 开发板进行测试,设定实验环境:系统的工作频率 $F=80\text{MHz}$,以 Snort2.0 中的 attack-responses.rules 作为规则库, p 为输入数据流中匹配的部分所占的比例.表 1 列出了 K 和 p 取不同值的时候,匹配系统吞吐率的变化.

表 1 基于 TCAM 的 K 步长多模式匹配系统吞吐率

K	$p=0(\text{Gbps})$	$p=3\%(\text{Gbps})$	$p=100\%(\text{Gbps})$
1	0.62	0.59	0.15
2	1.27	1.23	0.32
4	2.56	2.49	0.64
8	5.12	5.03	1.28

为了与其他实现方案作比较,在相同的实验环境下,对文献[1]中提出的基于 Bloom Filter 的 K 步长多模式匹配系统进行测试,结果如表 2 所示.

表 2 基于 Bloom Filter 的 K 步长多模式匹配系统吞吐率

K	$p=0(\text{Gbps})$	$p=3\%(\text{Gbps})$	$p=100\%(\text{Gbps})$
1	0.58	0.56	0.06
2	1.09	1.06	0.11
4	1.90	1.85	0.19
8	3.03	2.95	0.31

从测试结果来看,两个匹配系统的匹配速度都很高,在 p 值相同时, K 的值越大,基于 TCAM 的匹配系统性能更加优异.另外,随着 p 的增加,匹配速度降低,当 $p=100\%$ 时,基于 TCAM 的匹配系统的吞吐率是相同条件下基于 Bloom Filter 匹配系统的 2.5 倍以上,仍能保持百兆比特的匹配速度,所以基于 TCAM 的匹配系统抵御恶意攻击的能力更强.

5 结束语

该匹配系统的优点是:(1)匹配速度快.匹配速度与规则长度和数量无关,只与系统工作频率和外

部 RAM 的工作频率有关;(2)可重构.规则库更新时只需改写 TCAM 和外部 RAM 中的内容即可;(3)配置灵活.Snort2.0 中有 2259 条规则,总长度约为 30KB,基于这些规则构造的 K 步长状态机的规模并不大,TCAM 和 RAM 的容量不需太大,如果提高系统的工作频率或者做成专用芯片,匹配速度还会有更大提高.不足之处是受 FPGA 片内逻辑资源的限制,很难在其内部实现大规模的 TCAM.目前 FPGA 技术发展很快,片内的逻辑资源和集成度都大大提高,拓展了这种技术的应用范围.

参考文献:

- [1] Sarang Dharmapurikar, Lockwood John. Fast and scalable pattern matching for network intrusion detection systems[J]. IEEE, 2006, 24(10): 1781-1792.
- [2] Brelet Jean-Louis. Using block RAM for high performance read/write TCAMs [EB/OL]. Xilinx XAPP204[2012-4-19]. <http://www.xilinx.com>.
- [3] 冯安. 基于 Bloom Filter 的硬件字符串匹配设计与验证[J]. 图像·编码与软件, 2009, 22(12): 63-68.
- [4] Sourdis I. Pre-decoded TCAMs for efficient and high-speed NIDS pattern matching[C] // Proceedings of the 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'04). California: Napa Valley, 2004: 258-267.
- [5] Fang Yu, Katz R H, Lakshman T V. Gigabit rate packet pattern-matching using TTCAM[C] // Proceeding of 12th IEEE International Conference on Network Protocol. Berlin, Germany, 2004: 174-183.
- [6] Son H, Lockwood J. Multi-pattern signature matching of hardware network intrusion detection systems [C] // Global Telecommunications Conference. New York, 2005: 1686-1690.

作者简介:



赵海斌 (1984-), 硕士研究生. 研究方向为网络信息安全.

李训根 (1971-), 博士, 副教授, 硕士生导师. 研究方向为网络信息安全.

王奇敏 (1987-), 硕士研究生. 研究方向为网络信息安全.