第 38 卷 第 4 期
Vol.38 No.4

计 算 机 工 程
Computer Engineering

2012 年 2 月
February 2012

·开发研究与设计技术· 文章编号：1000—3428(2012)04—0269—03 文献标识码：A 中图分类号：TP393.08

基于 TCAM 的高效浮动关键词匹配算法

李鲲鹏，兰巨龙

(国家数字交换系统工程技术研究中心，郑州 450002)

摘 要：针对传统浮动关键词匹配算法功耗高和速率低的问题，提出一种基于三态内容寻址寄存器(TCAM)的高效匹配算法。该算法应用关键词分类数据结构，将关键词存储在不同的 TCAM 模块中，并只将疑似关键词送入 TCAM 中查找匹配，从而减少每次访问 TCAM 查找的表项数目，提高一个查询周期内待匹配报文的移动速度。仿真结果表明，与传统算法相比，该算法功耗较低、匹配速度较快。

关键词：浮动关键词；三态内容寻址寄存器；关键词分类数据结构；分配器

Efficient Unfixed Keywords Matching Algorithm Based on TCAM

LI Kun-peng, LAN Ju-long

(National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China)

【Abstract】 To deal with the high power consumption and low speed of unfixed keywords matching based on Ternary Content Addressable Memory(TCAM), a low power consumption and high speed unfixed keywords matching algorithm is proposed. With the help of keyword classification data structure, keywords are stored in different blocks of TCAM and only the suspected keywords are put into TCAM to match. Therefore, the algorithm decreases the entries of TCAM and increases the average step length of data packets for each query cycle, so it can effectively reduce the power consumption of TCAM and improve the matching speed. Simulation result shows that the algorithm has lower power consumption and faster matching speed than traditional algorithms.

【Key words】 unfixed keywords; Ternary Content Addressable Memory(TCAM); keywords classification data structure; allotter

DOI: 10.3969/j.issn.1000-3428.2012.04.088

1 概述

许多病毒、业务流等的特征信息都隐藏在 IP 包的载荷部分并且位置不固定。因此，要实现对 IP 数据包的有效管控，必须实现对数据报文载荷部分浮动位置的任意关键词的搜索，即解决浮动关键词匹配问题。浮动关键词匹配按实现方式可分为基于软件的实现方式[1-2]和基于硬件的实现方式[3-4]。基于软件实现方式的算法主要包括 Wu-Manber 算法、Multiple BNDM 算法，SBOM 算法等。这些算法的匹配速度很难达到 1 Gb/s 的速度。随着互联网的不断发展，骨干网链路速率已经达到 OC-768，甚至更高。基于软件的实现方式很难满足线速处理的要求，因此，基于硬件的实现方式成为当前研究的重点。

在基于硬件的实现方式中，由于三态内容寻址寄存器(Ternary Content Addressable Memory, TCAM)器件具有高度并行的查找机制，查找时间复杂度为 $O(1)$，受到广泛研究和应用。文献[3]提出了一种实现方案，将短关键词直接存储在 TCAM 中，将长关键词切分成多个短关键词的切片。这种方法可以实现对任意长度关键词的匹配，然而，在一个 TCAM 查询周期内，待匹配的报文只能移动一个字节，如果 TCAM 的查询周期为 4 ns，该算法的匹配速度只能达到 8 bit/4 ns= 2 Gb/s。文献[4]提出了一种灵活的移位匹配算法，每个 TCAM 查询周期都能使待匹配报文移动 w 个字节。但这种方法即使在最好的情况下一个关键词也需要占用 w 条 TCAM 表项，这对于本来就稀缺的 TCAM 存储空间造成了巨大的浪费。另外，在这 2 种算法中，TCAM 每次查询时都需要对整个表项进行匹配，而 TCAM 的功耗巨大，在处理数据的过程中平均每比特要耗费的功率是 SRAM 的 150 倍[5]，巨大的功耗不仅造成了资源的浪费还产生了大量的热量，带来了散热的问题。文献[6-7]分别应用不同的方式去除 TCAM 表项中的冗余来减小 TCAM 的功耗，但是这 2 种算法并不能提高 TCAM 的匹配速度。

本文提出一种低功耗高速浮动关键词匹配算法(LPCHS 算法)，该算法以消耗少量现场可编程门阵列(Field Programmable Gata Array, FPGA)资源为代价，有效地降低了 TCAM 的功耗，并且在表项为 10 000 条的情况下，不增加 TCAM 表项时，实现了一个 TCAM 查找周期内报文平均移动 2.8 Byte 的匹配速度。

2 LPCHS 算法

2.1 关键词分类数据结构

TCAM 的功耗与所查找表项的数量呈正比，每次访问 TCAM 查找的表项越多，功耗也就越大；它还与访问的次数有关，次数越多，功耗也越大。为减小功耗，TCAM 支持分块查找的功能，例如 NL3300 可以将有 256 k 表项的 TCAM 分成 32 个模块，每个模块分配 8 000 条表项。这样就可以选择其中的某一个或某几个模块进行查找操作，节省 TCAM 的功耗。为达到这个目的，本文设计了关键词分类数据结构。这种数据结构的主要功能有：(1)能够将关键词写入不同的 TCAM 模块中，查询时能够将待匹配模式串送入正确的

基金项目：国家“863”计划基金资助项目“高可信网络管控系统”(2009AA01A346)

作者简介：李鲲鹏(1985－)，男，硕士研究生，主研方向：深度数据包检测；兰巨龙，教授、博士生导师

收稿日期：2011-08-17 **E-mail：**csuli1022@163.com

TCAM 模块查询；(2)支持关键词的更新，能够很好地写入新的关键词和删除不需要的关键词；(3)支持待匹配报文的快速移位，以达到高速匹配的目的。

关键词分类数据结构逻辑上包括 Hash 向量、计数向量和编码向量 3 个部分，如图 1 所示。

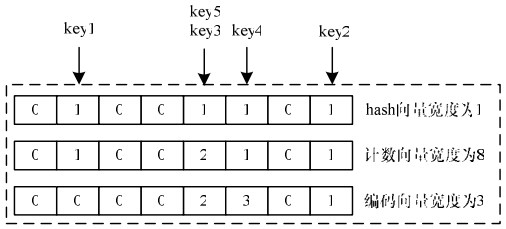
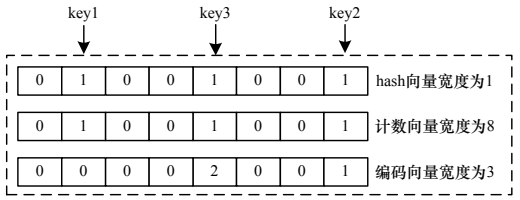


图 1 关键词分类数据结构

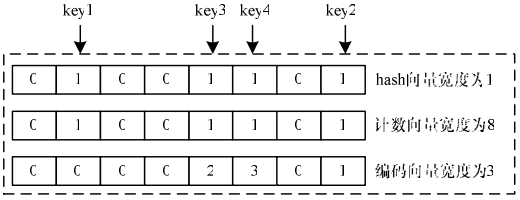
Hash 向量是关键词集的 Hash 函数对应的结果，例如关键词 key1 的 Hash 函数值对应 Hash 向量里的第 2 位，则就将该位置 1，若已经为 1，则不做任何操作。计数向量是 Hash 向量对应的计数器，记录 Hash 向量每一位中的关键词数。初始化时为全 0，当关键词的 Hash 函数值对应到该位时就加 1。图 1 中 Hash 向量的第 2 位对应一个关键词，则计数向量对应的位就为 1，Hash 向量的第 5 位对应 2 个关键词，则计数向量该位就为 2。编码向量对应该关键词所存储的 TCAM 模块，宽度由 TCAM 的模块数决定，如 TCAM 分成 8 个模块查找，则编码向量宽度为 3 即可。000 表示第 1 个 TCAM 模块，111 表示第 8 个模块。

该数据结构的工作过程可分为关键词写入和删除、报文查找 2 个阶段。

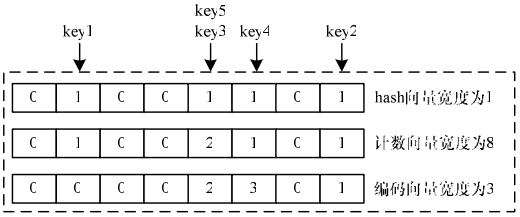
关键词的写入最重要的是将关键词尽量平均分配到 TCAM 的模块中，因为关键词在 TCAM 模块中的分布越均匀越有利于以后新表项的添加。具体过程如图 2 所示。



(a)已有表项



(b)关键词 key4 的添加过程



(c)关键词 key5 的添加过程

图 2 关键词添加过程

由于关键词集的规模较大，选用的 Hash 函数又能保证关键词集间的冲突概率较小，因此本文直接采用比较简单的方式分配关键词到 TCAM 模块中。这样虽然不能不能保证绝对平均，但却能够支持关键词集的实时更新。写入新关键词时首先计算其 Hash 函数值，如果该 Hash 函数值对应 Hash 向量里的值已经为 1，则查询该位对应的 TCAM 模块并将关键词写入该模块，对应的计数向量位加 1。如果 Hash 函数值对应的 Hash 向量里的值为 0，则根据计数向量计算每个 TCAM 模块里存储关键词的个数，将该关键词写入关键词个数最少的 TCAM 模块中，并设置相应的 Hash 向量、计数向量和编码向量。

关键词删除和报文查找阶段相对比较简单，删除时在清除 TCAM 里的表项时只需要根据计数向量对应的位修改 Hash 向量及编码向量即可。若计数向量为 1，则将 Hash 向量对应的位置 0，并将计数向量和编码向量对应位置 0，若大于 1，计数向量对应位减 1，其他不变。查询阶段若命中，只需根据编码向量的对应位将待匹配关键词送入相应的 TCAM 模块做进一步匹配。

2.2 算法设计

LPCHS 算法的整体结构如图 3 所示。

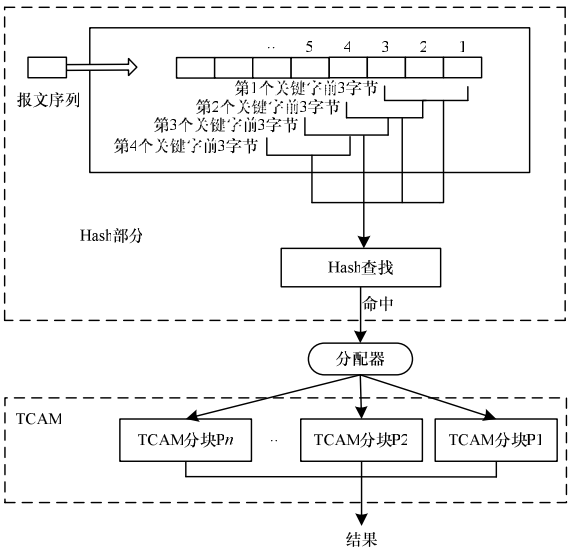


图 3 LPCHS 算法结构

该算法可分为 3 个部分：Hash 函数，分配器和 TCAM。Hash 函数部分主要实现关键词分类数据结构，分配器部分根据 Hash 函数的结果将待匹配的关键词送入相应的 TCAM 模块进行匹配，TCAM 部分主要完成匹配的工作。分配器和 TCAM 部分都非常容易实现，下面主要介绍 Hash 函数部分的实现。

Hash 函数部分应用 FPGA 提供的 Block Ram 资源实现，其中 Ram 存储关键词分类数据结构，Hash 函数值为 Ram 访问的地址。本文中采用 CRC16 作为 Hash 函数，但是 Hash 函数只能对精确关键词进行变换，不能对 TCAM 表项里的无关项进行变换。但是 TCAM 表项里的浮动关键词开始的字节都是精确的关键词。假定该系统所要存储的浮动关键词集的前 3 个字节都是精确的关键词，在写入关键词时，提取其前 3 个字节作为配置关键词分类数据结构的依据，其中，关键词分类数据结构的 Hash 向量的长度为 64 kb，计数向量的宽度为 8 bit，TCAM 分为 4 个模块，所以编码向量宽度为 2。匹配过程首先提取待匹配关键词的前 3 个字节计算 Hash 值并与 Hash 向量比较，若命中则将关键词与编码向量一起发送到分配器的缓存里等待送往 TCAM 模块匹配，若没有命中，则该数据报文肯定不匹配关键词集。匹配过程中为了增加报

文的移位速度，设置了 4 个 Hash 单元进行并行工作。

3　算法性能分析

3.1　功耗分析

TCAM 消耗的功率与每次查表的数目有关，还与访问 TCAM 的次数有关，用每次访问 TCAM 所查表项总数与待匹配报文每移动一个字节平均需要查表次数的乘积作为 TCAM 功耗的度量值。文献[2]的灵活移位浮动关键词算法的移位宽度设为 4，则在最好的情况下其 TCAM 中的表项就是原始表项的 4 倍，待匹配报文每移动一个字节平均需要的查表次数为 1/4。在文献[3]提出的实现方案中，TCAM 的表项数目与原始表项数目相同，待匹配报文每移动一个字节需要的查表次数为 1。本文提出的 LPCHS 算法 TCAM 的表项数目与原始表项数目相同，但是每次查表时查找的表项数目只与某个 TCAM 模块里的表项数目有关，而报文每移动一个字节所需的查表次数与 Hash 部分送往 TCAM 模块匹配的关键词数有关，推导如下：

假设关键词集的数目为 m，这些关键词中前 3 个字节相同的关键词数目为 t，则在对报文匹配时，只要遇到与关键词集里前 3 个字节相同的待匹配关键词就应该送入 TCAM 模块，如果组成关键词集的有效字符有 36 个(26 个字母和 10 个数字)，则应该送入 TCAM 的关键词概率为：

$$p_{\text{positive}} = \frac{m-t}{36^3}$$

由于 Hash 函数会产生误匹配，即将前 3 个字节不匹配的关键词也送入 TCAM，其概率由文献[8]可得：

$$p_{\text{false}} \approx 1-\mathrm{e}^{-m/n}$$

其中，n 为 Hash 向量的长度，即 64 kb。

因此，关键词需送入 TCAM 匹配的概率为：

$$p = p_{\text{positive}} + p_{\text{false}}$$

当关键词里的前 3 个字节都不相同时，即 t=0 时，p 的值最大，此时 p 随关键词集大小的变化如图 4 所示。

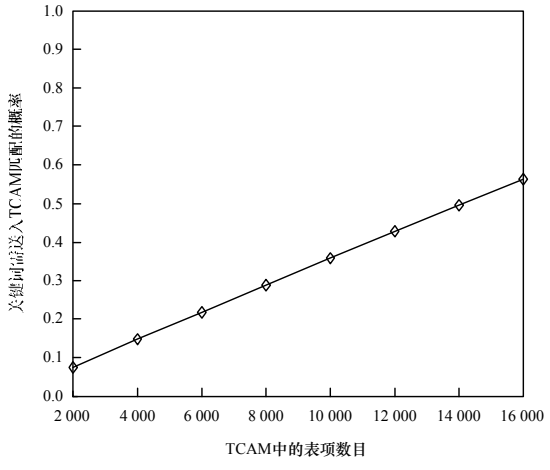


图 4　关键词需送入 TCAM 中匹配的概率

可以看出，在前 3 个字节都不同时，需要送入 TCAM 的待匹配关键词数最多。笔者设置了 4 个 Hash 函数部分，则每次待匹配报文向前移动 4 个字节，送入 TCAM 的关键词为 $4\times p$。每移动一个字节所需要的查表次数为 $4\times p/4$，即 p 次。

TCAM 分为 32 个模块，每个模块 4 000 条表项。3 种算法的比较如表 1 所示，其中，算法 1 为文献[3]方案；算法 2 为文献[4]方案。

表 1　TCAM 功耗比较

原始表项数目	查询 TCAM 的表项数目			功耗			较算法 1 节省量/(%)	较算法 2 节省量/(%)
	算法 1	算法 2	本文算法	算法 1	算法 2	本文算法		
6 000	8 000	24 000	4 000	8 000	6 000	1 000	87.5	83.33
10 000	12 000	40 000	4 000	12 000	10 000	1 436	88.03	85.64
15 000	16 000	60 000	4 000	16 000	15 000	2 122	86.74	85.85

3.2　匹配速度分析

因为 TCAM 的查表周期是固定的，所以在一个查表周期内待匹配报文的移动位置越多，匹配速度越快，本文用一个 TCAM 查表周期内待匹配报文移动的字节数表示算法的速度。

文献[3]算法每个查表周期只能移动 1 Byte，文献[2]算法一个查表周期可以实现 4 Byte(移动窗口大小)的移动，但是 TCAM 中的表项为原始表项的 4 倍。本文提出的算法每个查表周期移动的字节数与送往 TCAM 的查表关键词数目密切相关，平均情况下为 $4/(4\times p)$，其中，p 为每个 Hash 部分送往 TCAM 查表的概率。3 种算法的比较如图 5 所示。

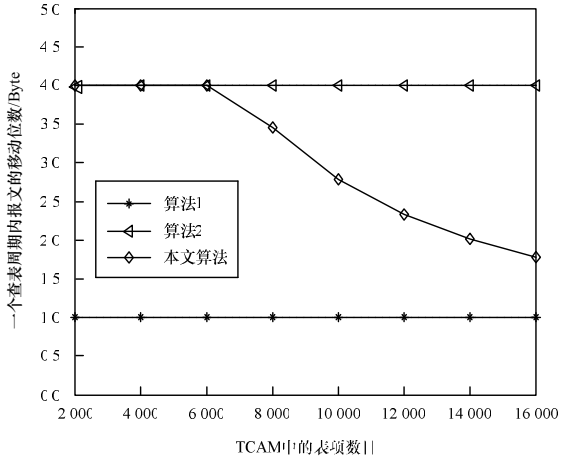


图 5　3 种算法的匹配速度比较

可以看出，文献[4]提出的灵活移位方法匹配的速度最高，但是这种算法增加了每个关键词在 TCAM 中表项的数目，每个关键词在 TCAM 中要占 4 条表项，这种方式不能适应关键词集的增加。而文献[3]算法每次只能移动一个字节，满足不了骨干网链路的速率要求。本文提出的算法利用了 FPGA 提供的资源，构造了 Hash 函数部分，从而在不需要增加 TCAM 表项的基础上增加了移动位数，在关键词集不大于 6 000 条的情况下，每个查表周期能够移动 4 Byte，即使关键词集达到 10 000 条，每次也能移动 2.8 Byte，提高了匹配速度。

4　结束语

本文提出了一种基于 TCAM 的高效浮动关键词匹配算法(LPCHS 算法)，以一定的 FPGA 资源为代价，有效降低了 TCAM 的功耗并且提高了匹配的速度。该算法要求关键词的前 3 个字节必须是确定关键词，但这并不影响它的应用，可以应用于数据包深度检测，敏感词过滤及 URL 过滤等各个领域。对关键词分类时算法使用了关键词的前 3 个字节，如果能够充分利用关键词里的所有确定关键词就可以进一步减少送入 TCAM 中查找的待匹配的关键词数目，从而更加有效地提高匹配速度和减低功耗，这也是下一步研究的方向。

(下转第 274 页)