

基于 TCAM 的范围匹配方法——C-TCAM

朱国胜¹, 余少华^{1,2}

(1. 华中科技大学 计算机学院, 湖北 武汉 430074;

2. 武汉邮电科学研究院 新一代光纤通信技术和网络国家重点实验室, 湖北 武汉 430074)

摘要: 提出一种基于 TCAM 的范围匹配方法——C-TCAM (compressed TCAM)。空间方面, 通过二级压缩存储, C-TCAM 可以将 2 个扩展后的 TCAM 表项压缩成一个, 最坏情况下范围扩张因子为 $w-1$ 或者 $w-2$, 提高了空间利用率; 功耗方面, 通过一种新的 TCAM 查找算法来避免无效表项参与比较从而降低了功耗; 分析和仿真显示 C-TCAM 方法在实现性能分组分类的同时在空间利用率、功耗等方面具有优势。

关键词: 分组分类; TCAM; 范围匹配; 表项扩张; 压缩; 功耗

中图分类号: TP393.08

文献标识码: A

文章编号: 1000-436X(2012)01-0031-07

Range matching method based on TCAM: C-TCAM

ZHU Guo-sheng¹, YU Shao-hua^{1,2}

(1. School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China;

2. Wuhan Research Ins. of Posts and Telecoms, State Key Laboratory for New Optical Communication Technologies and Networks, Wuhan 430074, China)

Abstract: A new TCAM range matching method called C-TCAM (compressed TCAM) was proposed. Firstly C-TCAM could compress two expanded TCAM entries into one and the expansion factor of $w-1$ and $w-2$ respectively in the worst case. Secondly, a new TCAM matching algorithm to lower down the power dissipation was designed by avoiding matching of unnecessary TCAM entries. Analysis and simulation showed that C-TCAM gets advantages over other schemes in TCAM space efficiency and power dissipation.

Key words: packet classification; TCAM; range matching; expansion; compression; power dissipation

1 引言

互联网路由交换网络设备需要采用分组分类技术, 将到达分组的头部信息和分类规则库中的规则条件进行匹配, 并根据匹配结果实现规则动作, 包括转发、丢弃、队列排队等, 从而实现访问控制、安全过滤、带宽控制等。和 IP 路由查找一样, 分组分类功能处在网络设备的数据平面, 需要进行每分组处理, 对性能要求很高。

目前, 业界普遍采用三态内容寻址存储器 (TCAM, ternary content-addressable memory) 来实

现高速分组分类, Cisco 的 Catalyst 6500 高端交换机采用多片 TCAM 来分别实现 3 层 IP 路由、访问控制列表 ACL、分组分类以及 NETFLOW 等功能^[1]。TCAM 可配置成 36bit、72bit、144bit、288bit、576bit 等宽度, 用于存储固定宽度的查找条件。和普通内存通过地址寻找内容不同, TCAM 通过内容来定位地址, 它将查找关键字和所有 TCAM 表项进行并行比较来定位关键字匹配的存储地址, 根据得到的地址来得到存储在快速 SRAM 中的规则动作。TCAM 的并行特性使得分组分类查找可以在固定的时钟周期内完成, 目前 TCAM 的时钟周期可以做到 2ns,

收稿日期: 2010-07-02; 修回日期: 2010-11-10

基金项目: 国家高技术研究发展计划 (“863” 计划) 基金资助项目 (2005AA121410)

Foundation Item: The National High Technology Research and Development Program of China(863 Program)(2005AA121410)

可以实现 500MSPS 的查找速率^[2], TCAM 三态特性需要采用多达 16 个晶体管来实现一个比特的存储, 导致芯片面积较大而且价格较高, 而 TCAM 的并行比较特性使得功耗很高, 一个 18Mbit 的 TCAM 功耗高达 15W^[3]。因此在单板线卡设计中必须考虑 TCAM 的成本、功耗和体积, 并充分利用 TCAM 的每个比特。

有效利用 TCAM 来完成高速分组分类的主要障碍是 TCAM 不适合实现范围匹配, TCAM 的三态特性可以很容易地实现规则条件中的前缀匹配和精确匹配, 比如规则条件中的地址字段和协议字段匹配, 而规则条件中的范围字段比如端口范围则无法直接实现, 必须转换成前缀匹配和精确匹配, 一个包含范围字段的规则需要转换成多个规则才能存放在 TCAM 中, 称之为范围扩张 (range expansion), 范围扩张一方面降低了 TCAM 的空间使用效率, 另一方面 TCAM 的功耗和参与比较的表项的数目成正比, 因此范围扩张会导致 TCAM 功耗的增加。

如图 1 所示为 4bit 宽度的范围字段范围扩张的示例, 范围[1,14]需要扩张成 6 个前缀匹配或者精确匹配的条目, 在 TCAM 中进行存放并参与匹配比较, 范围[1,14]的范围扩张因子为 6, 每条携带范围[1,14]的匹配规则需要扩张成 6 条规则。

因此, 基于 TCAM 的范围匹配主要解决如下的 2 个问题: 如何减少范围扩张, 以有效利用 TCAM 的空间; 如何设计有效的 TCAM 匹配算法以降低功耗。

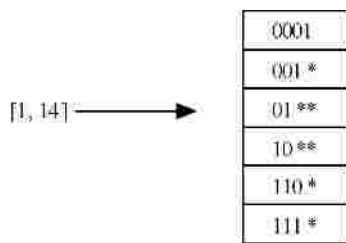


图 1 宽度为 4bit 的范围扩张示例

2 相关工作

2.1 规则库无关范围匹配方法

对每个范围单独进行转换, 无需考虑规则库里各种范围的分布以及和其他范围的关系。文献[4]提出了一种前缀扩展 (PE, prefix expansion) 方法, 该方法将范围转换成若干个前缀, 每个前缀对应一

个 TCAM 表项, 在最坏情况下一个范围需要转换成 $2(W-1)$ 个前缀, 其中, W 为范围字段的长度, 范围 $R_w=[1, 2^w-2]$ 就是这种情况(前面的引言的范围[1,14]是 4bit 示例); 对于长度为 16 的源端口和目的端口来说, 最坏情况下一条规则需要转换成 $(2 \times 16-2)^2=900$ 个 TCAM 表项, 对于真实分类规则库转换显示平均会导致 6 倍的表项扩张^[5]。文献[6]提出了利用 TCAM 额外比特对范围进行独立编码的 DIRPE(database independent range preencoding) 方法, 假定 TCAM 的额外比特为 2^w , 则每个范围只需要一个 TCAM 表项就可完成转换, 由于 TCAM 的额外比特数有限, 在配置为 144bit 宽度时, 额外比特数仅为 40, 为减少对 TCAM 额外比特的需求, 提出了对范围字段进行分段编码以及结合规则库相关的编码方法, 增加了每个范围需要的 TCAM 表项数目, 因此, 该方法的效率和范围字段的长度和 TCAM 的额外比特数量相关。文献[7]提出了基于格雷码 (Gray code) 的范围编码方法 SRGE, 该方法利用了格雷码的相邻编码之间只有一个比特不同从而可以合并成一个 TCAM 表项的特点(将该不同比特用通配符 ‘*’ 来表示), 在最坏情况下一个范围需要转换成 $2(W-2)$ 个前缀, 该方法无需利用 TCAM 的额外比特, 但仅对于短的范围比较有效。

2.2 规则库相关范围匹配方法

文献[8]提出了对每个单独的范围用一个额外的 TCAM 比特来表示的规则库相关范围匹配方法, 只需要一个 TCAM 表项就可以完成一个范围规则的表示。由于目前规则库中单独的范围在增加, 已经超过 300 个, 而且还在不断地增加, 而额外的 TCAM 比特数量非常有限, 同样, 作者也提出分段编码方法 (region-based encoding) 增加 TCAM 表项从而减少对 TCAM 额外比特的需求。为了进一步对范围进行有效地编码并减少编码的比特数, 文献[9]提出层次化编码方法, 在每个层次里面, 范围都是不相交的, 对每个层次里面的范围进行单独编码, 假定层 L_i 中的不相交范围个数为 N_i , 则需要的比特数位 $\text{lb } N_i$, 而前述方法中每个范围需要一个 TCAM 比特, 从而减少了每个范围需要的比特。

另外, 文献[10]提出了 E-TCAM 方法, 该方法直接修改 TCAM 硬件设计支持范围匹配, 可以降低 90% 的功耗, 由于需要对 TCAM 硬件进行修改, 因此会增加每比特成本, 存在兼容性和扩展性问题, 短期内无法大规模部署和实现。

相比而言,规则库无关范围匹配方法和具体规则库无关,在规则更新方面具有优势。

3 C-TCAM 方法

3.1 C-TCAM 算法思想

我们发现:一条规则经过前缀扩展转换或者格雷码转换成多条规则后,规则条件里面的非范围字段是一样的,仅范围字段不一样,可以将多条规则存放在一个 TCAM 表项中,而且,只需保存一份非范围字段的拷贝,比如对于 IPv4 分组分类来说,采用 144bit TCAM 位宽,非范围字段包括源地址、目的地址、协议字段共 72bit,范围字段包括源端口、目的端口共 32bit,还剩余 40bit 可以再保存一份源端口、目的端口字段,这样可以将二条规则压缩在一个 TCAM 表项里面,可以成倍地减少最坏情况下范围扩张,有效利用昂贵的 TCAM 空间。

另外,查找时,现有商用 TCAM 芯片都支持全局掩码寄存器(GMR, global mask register)和分区掩码寄存器(BMR, block mask register)^[2],GMR 寄存器可以从纵向确定 TCAM 芯片中参与比较的比特,BMR 寄存器用于支持在横向上将 TCAM 芯片进行分区,支持分区禁用,本方法利用了 TCAM 芯片的特性,在查找时通过 GMR 和 BMR 动态地选择参与比较的比特和参与比较的 TCAM 表项,在减少范围扩张的同时减少参与比较的 TCAM 表项数目,从而降低功耗。

3.2 问题定义和相关术语

分组分类的相关术语和定义沿用文献[6,7]的说法。分组的头部由若干个字段组成,每个字段包含若干个比特。查找关键字 KEY 是从分组头部提取的 K 个字段的组合。查找关键字需要和存储在分类规则库中的规则条件进行匹配。

规则包含规则条件和规则动作,规则条件由相应的 K 个字段组成,如果分组 P 的 K 个关键字字段和规则 R 的 K 个规则条件匹配,则认为分组 P 匹配规则 R ,相应会对分组 P 采取规则 R 规定的规则动作。每个规则字段 f 可以指定 3 种类型的匹配条件。

- 1) 精确匹配:关键字字段和规则条件字段指定的值相等。
- 2) 前缀匹配:规则条件字段是关键字字段的前缀。
- 3) 范围匹配:关键字字段在规则条件指定的范

围里面,比如规则条件指定的范围为 $[s, e]$, s 和 e 为整数, $s < e$, 对于 IP 端口号来说,该字段长度 $W=16$ 。

分类规则库可以存放在 TCAM 中以实现高速分组分类,规则在 TCAM 中按照优先级从高到低排序存放,高优先级存放在 TCAM 低地址区,TCAM 总是返回第一个匹配的结果(低地址区)。TCAM 的每个比特支持“0”、“1”和“*”3种状态,精确匹配规则条件和前缀匹配规则条件可以直接存放,而范围匹配则需要转换成相应的精确匹配和前缀匹配,因此一条规则 R 需要采用转换方法 E 转换成多个 TCAM 表项 $TCAM_{R[1, n_R]}$, 范围 R 的扩张因子 $R_E = n_R$, 假定 $n(D)$ 为分类规则库 D 的规则数量, $n_E(D)$ 采用转换方法 E 对分类规则库进行转换后所有 TCAM 表项的数目,则分类规则库 D 的扩张因子 D_E 表示为

$$D_E = n_E(D)/n(D) \quad (1)$$

由于规则库里存在不包含范围匹配的规则,为进一步考察范围规则转换的有效性,假定 $r(D)$ 表示规则库里面包含范围的规则数, $r_E(D)$ 表示对这些包含范围的规则进行转换后的 TCAM 表项数,规则库 D 的范围冗余因子 $FR_E(D)$ 表示为

$$FR_E(D) = (r_E(D) - r(D))/r(D) \quad (2)$$

而某条规则 R 的范围冗余因子 $FR_E(R)$, 也就是规则 R 需要的额外的 TCAM 表项的个数

$$FR_E(R) = r_E(R) - 1 \quad (3)$$

文章的第 5 节的仿真实验中将通过实验对不同算法的规则库扩张因子 D_E 和规则库冗余因子 $FR_E(D)$ 进行比较。

3.3 C-TCAM 体系结构

图 2 所示为 C-TCAM 的体系结构,对规则库中包含范围的规则进行前缀扩展或者格雷码扩展,扩展后的规则在 TCAM 中进行按照规则优先级排序,在每行 TCAM 中存放二条规则,这二条规则除了源端口和目的端口不同外,其他比如源地址、目的地址、协议和规则动作均相同,因此除了源端口和目的端口外的其他字段只需要保存一份拷贝。图 2 所示假定规则库 D 包含 $R1$ 到 $R7$ 7 条规则,并且按照优先级进行排序,经过前缀扩展编码或者格雷码编码转换后, $R1$ 需要 4 个 TCAM 表项(表示为 $R1_1$ 、 $R1_2$ 、 $R1_3$ 、 $R1_4$) $R3$ 和 $R6$ 需要一个 TCAM 表项,其他规则需要 2 个 TCAM 表项,转换后的规则存储

如图 2 所示, R_1 的 4 个 TCAM 表项占据 2 行 TCAM, 其他规则均占据一行 TCAM, 需要注意的是, 虽然 R_3 和 R_6 只需要一个 TCAM 表项, 但依然在 GMR3 对应的比特存放了 R_3 和 R_6 对应的源端口和目的端口, 而且和 GMR1 对应的源端口和目的端口相同, 这主要是为了保证匹配算法返回的结果正确, 具体原因在 3.4 节的 C-TCAM 查找部分进行说明。

图中的 GMR 用于从纵向上选择和查找关键字进行匹配的比特, 当选择 $GMR=GMR_0+GMR_1+GMR_2$, GMR3 和 GMR4 对应的比特不参与匹配比较; 当选择 $GMR=GMR_0+GMR_2+GMR_3$, GMR1 和 GMR4 对应的比特不参与匹配比较, 可以看出, GMR4 对应的比特为空闲比特, 在 2 种情况下都不参与比较, 对于 IPv4 分组分类, TCAM 配置为 144bit 宽度, GMR4 的宽度为 8bit, 某些规则条件还包括 TCP 标志字段, 这时这些空闲比特可以进一步被利用。

图中的 BMR 用于从横向上选择参与匹配比较的 TCAM 表项, 由于规则存在重叠关系, C-TCAM 方案需要经过二轮匹配才能确定匹配的规则, 比如 R_7 的规则条件包含 R_4 , 但是 R_7 定义的动作和 R_4 不同, 为保证结果正确 (选择优先级高的 R_4), C-TCAM 需要查找匹配需要进行二轮匹配, 判决逻辑用于对二次匹配的结果进行比较, 并选择低地址区的匹配结果作为最终分组分类结果, 对分组执行规则条件对应的动作。

3.4 C-TCAM 匹配算法

经过上述的扩展和压缩存储后, 减少了范围扩张数目, 有效利用 TCAM 的空间, 较好地解决了基

于 TCAM 的范围匹配的第一个问题; 对于第二个问题, 需要设计有效的 TCAM 匹配算法减少参与比较的表项数目以降低功耗。

可以看出, 表项通过压缩存储后, 为了返回正确的结果, 需要进行两轮查找, 并选择两轮查找的索引小者作为查找结果用于定位规则动作。

在上节曾提到, 虽然 R_3 和 R_6 只需要一个 TCAM 表项, 但依然在 GMR3 对应的比特存放了 R_3 和 R_6 对应的源端口和目的端口, 而且和 GMR1 对应的源端口和目的端口相同, 主要是因为如果 R_3 和 R_6 的 GMR3 对应的比特放置通配, 则在第二轮查找时 R_3 和 R_6 总是会返回匹配, 会导致结果的不正确。例如, 假定某关键字 KEY 应该匹配 R_7 , 在第二轮匹配时, R_3 、 R_6 和 R_7 都返回匹配, R_3 优先级最高, 判决逻辑会选择 R_3 作为匹配结果, 从而导致结果错误, 如果在 GMR3 的对应比特放置一份和 GMR1 相同的源端口和目的端口, 则可避免该问题的出现。

C-TCAM 匹配算法如图 3 所示, 该算法输入为查找关键字 KEY, 输出为匹配的规则在 TCAM 中的地址对应的索引, 用于定位规则动作。在获取 Index1 之后, 计算 Index1 所在的 Block, BMR 被赋值给 Index1 所在 Block 及其以上的 Block, 匹配得到 Index2, 将 Index1 和 Index2 中的较小者返回。原因是在第一轮匹配时, 对于存放位置大于 Index1 的表项的比较是没有意义的, 而且会增加功耗, 通过对大于 Index1 的 Block 进行禁用, 在第二轮只允许 Index1 所在的 Block 及其以上的 Block 参与比较, 从而降低功耗, 也不会影响结果的正确性。



图 2 C-TCAM 体系结构

```

C-TCAM-MATCH2(KEY)
{
    Index1 = Index2 = Max-Number; /* */
    GMR = GMRO + GMR1 + GMR2;
    BMR = All Blocks;
    Index1 = Match(KEY, TCAM);
    GMR = GMRO + GMR2 + GMR3;
    Blockcurrent = Index1 Block;
    BMR = Blockcurrent Block;
    Index2 = Match(KEY, TCAM);
    If Index1 < Index2 Then
        return Index1;
    else
        return Index2;
}
    
```

图 3 C-TCAM-MATCH2 算法

上述 C-TCAM 方法将 2 个表项进行压缩存储，称之为二级 C-TCAM。由于 TCAM 的位宽是可配的，如果经过二级 C-TCAM 压缩存储后还有空闲比特可用，可以进行多级压缩存储进一步提供空间利用率，查找算法经简单修改后依然适用。必须注意，多级扩展会导致查找性能的下降，因此，必须综合考虑空间、功耗和性能等因素。

4 算法比较和分析

表 1 所示为不同算法的关键指标对比，包括最坏情况下的范围转换扩张因子、对 TCAM 额外比特的占用需求、规则更新的代价、功耗以及匹配性能等。表中 F 为包含范围匹配的字段的个数（对于 IP 分组分类来说为 2：源端口字段和目的端口字段）， W 为包含范围匹配的字段的长度， N 为规则库规则条目总数。

最坏情况下扩张因子方面，二级 C-TCAM 方法比前缀扩展方法和格雷码方法降低一倍，优势比较明显。

指标	前缀扩展	格雷码	二级 C-TCAM
最坏情况下扩张因子	$(2(W-1))^F$	$(2(W-2))^F$	$(W-1)^F$ 或 $(W-2)^F$
TCAM 额外比特	0	0	$2W$
更新代价	$O(W^F)$	$O(W^F)$	$O(W^F)$
最坏情况下功耗	$N(2(W-1))^F$	$N(2(W-2))^F$	$(1+?)N(2(W-1))^F/2$ 或者 $(1+?)N(2(W-2))^F/2$
匹配性能	One Clock	One Clock	Two Clock

TCAM 额外比特方面，前缀扩展和格雷码无需额外比特，二级 C-TCAM 方法需要 $2W$ bit，在 IP 分组分类下为 $2 \times 16=32$ bit，C-TCAM 方法的额外比特是利用空闲的 TCAM bit，不但不会新增另外的 TCAM 空间，而且对空闲空间进行了充分的利用。

规则更新方面，C-TCAM 方法与前缀扩展和格雷码方法相同，一条规则的添加、删除和规则库里其他规则无关，支持规则的增量更新。

功耗方面：最坏情况下的功耗 P_{max} 是硬件设计的功耗预算计算和电源设计时必须考虑的内容。TCAM 功耗和参与比较的 TCAM 条目总数相关，前缀扩展方法和格雷码方法的最坏情况下功耗分别为

$$P_{max}(PE)=N(2(W-1))^F \quad (4)$$

$$P_{max}(SRGE)=N2(W-1))^F \quad (5)$$

二级 C-TCAM 方法的功耗为

$$P_{max}(C-TCAM)=(1+?) (P_{max}(PE))/2 \quad (6)$$

或者

$$P_{max}(C-TCAM)=(1+?) (P_{max}(SRGE))/2 \quad (7)$$

参数 $?$ 的取值和第一轮匹配时 Index1 所在的 Block 号相关，假定 Index1 所在 Block 号为 $B1$ ，所用到的 TCAM 的所有 Block 数量为 B ，则

$$? = (B+B1)/2B \quad (8)$$

$B1$ 总是小于等于 B ，因此 $?$ 总是小于等于 1，C-TCAM 方法功耗最低。

查找性能方面，其他方法需要一个时钟周期，二级 C-TCAM 需要 2 个时钟周期，但是可以看到，现有商用 TCAM 时钟周期可以做到 2ns，以最小以太网分组长 672bit(包括前导码、帧间隔)来计算，为保证 100Gbit/s 接口的线速转发，分组转发率要达到 148.8Mpacket/s，每分组处理时延小于 6.72ns，二级 C-TCAM 依然可以满足要求。

5 实验仿真

由于真实的分类规则包含隐私信息无法公开获取，采用 Classbench^[11]工具来生成分类规则库和测试用 Trace 文件。Classbench 是华盛顿大学圣路易斯分校 Taylor 等开发的开放源代码分组分类规则和测试用 Trace 文件生成工具，Classbench 通过分析真实的访问控制列表、防火墙和 Linux 的 IPChains，得到种子文件，这些种子文件包含用于生成分类规则库的相关参数，比如地址字段长度分布、协议字段分布、端口字段通配分布、端口范围和精确匹配分布等。Classbench 生成的 Trace 文件包含随机的头部信息，用于分类匹配算法的测试。

利用 Classbench 生成了 6 个规则库 D1 到 D6，规则库的规则条目以及经过前缀扩展转换、格雷码转换、C-TCAM 转换后的规则条目如表 2 所示。

规则库	D1	D2	D3	D4	D5	D6
规则条目数	675	802	1 024	1 556	2 198	3 350
范围条目数	453	603	945	1 104	1 901	2 703
前缀扩展	1 674	2 254	3 062	3 859	5 957	9 682
格雷码 SRGE	1 586	2 077	2 765	3 703	5 781	9 045
C-TCAM(PE)	959	1 227	1 577	2 163	3 121	5 193
C-TCAM(SRGE)	905	1 139	1 423	2 085	3 055	4 858

从表 2 可以看出 经过 C-TCAM 转换后 TCAM 规则条目大大减少。为便于比较,图 4 和图 5 给出了基于表 2 的不同算法的规则库扩张因子 D_E 对比和规则库冗余因子 $FR_E(D)$ 对比,从图中可以看出, C-TCAM 方法的扩张因子和冗余因子比前缀扩展方法和格雷码方法的要低。因此在 TCAM 空间方面具有优势。

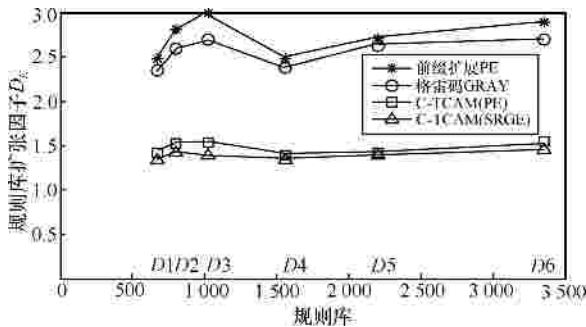


图 4 不同算法的规则库扩张因子 D_E 对比

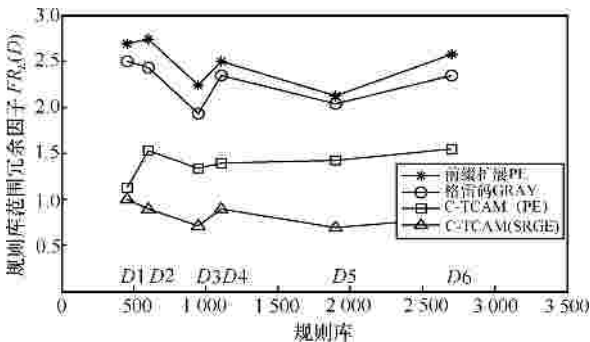


图 5 不同算法的规则库冗余因子 $FR_E(D)$ 对比

功耗方面,TCAM 的功耗和参与比较的 TCAM 的条目数相关。基于规则库 D_3 的 1024 条规则,选择 TCAM Block 大小为 128bit,TCAM 宽度为 144bit,利用 Classbench 生成测试用 Trace 文件,Trace 文件包含 2 000 个分组头信息。将 2 000 个分组输入不同的分类匹配方案进行仿真测试,记录每个分组头需要比较的 TCAM 条目的数 如图 6 所示。

图 6(a)为前缀扩展方法和 C-TCAM 的匹配条目

对比,图 6(b)为格雷码方法和 C-TCAM 的匹配条目对比。图中,横轴表示 2 000 次实验,纵轴表示每次实验需要比较的 TCAM 条目数。

从图 6(a)可以看出,前缀扩展方法每次需要比较的 TCAM 条目数固定为 3 062 表现为直线,C-TCAM (PE)方法需要比较的 TCAM 条目数在 1 577 和 3 154 之间波动,其均值为 2 457,小于前缀扩展方法的 3 062。

同样,从图 6(b)可以看出,格雷码方法每次需要比较的 TCAM 条目数固定为 2 765 表现为直线,C-TCAM (SRGE)方法需要比较的 TCAM 条目数在 1 423 和 2 826 之间波动,其均值为 2 258,小于格雷码方法的 2 765。

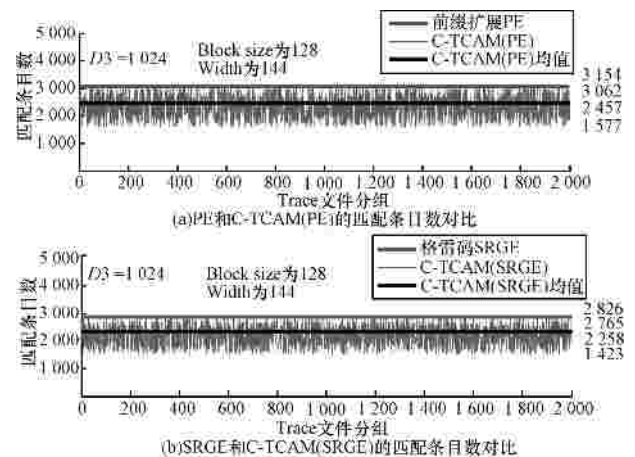


图 6 不同匹配算法需要比较的 TCAM 条目数 (规则库 D_3)

6 结束语

针对基于 TCAM 分组分类在实现范围匹配时存在的 TCAM 空间利用率低和功耗高的问题,本文提出了一种新的基于 TCAM 的范围匹配方法 C-TCAM:空间方面,通过二级压缩存储,C-TCAM 可以将 2 个扩展后的 TCAM 表项压缩成一个,最坏情况下表项扩张因子为 $W-1$ 或者 $W-2$,提高了空间利用率;功耗方面,通过查找算法来避免无效表项参与比较从而降低了功耗;分析和仿真显示 C-TCAM 方法在实现高速分组分类的同时在空间利用率、功耗等方面具有优势。

必须看到,虽然基于现有商用 TCAM,C-TCAM 可以支持 100Gbit/s 的线速分组分类,但是该方法需要经过多个 TCAM 时钟周期才能完成一次分组分类,因此,它是一种在性能、空间和功耗上做出选择的方案。

参考文献：

- [1] Suran de Silva. Cisco 6500 FIB forwarding capacities[EB/OL]. <http://www.nanog.org/mtg-0702/presentations/fib-desilva.pdf>, 2007.
- [2] Netlogic microsystems[EB/OL]. <http://www.netlogicmicro.com/>, 2010.
- [3] ZANE F, NARLIKAR G, BASU A. CoolCAMs: power-efficient TCAMs for forwarding engines[A]. Proceedings of the 22nd IEEE INFOCOM[C]. San Francisco, USA, 2003. 42-52.
- [4] TAYLOR D, SPITZNAGEL E, TURNER J. Packet classification using extended tcams[A]. ICNP '03 Proceedings of the 11th IEEE International Conference on Network Protocols[C]. 2003.120-131.
- [5] SRINIVASAN V, VARGHESE G, SURI S, *et al.* Fast and scalable layer four switching[A]. ACM SIGCOMM 98[C]. 1998.191-202.
- [6] TAYLOR D E. Survey and taxonomy of packet classification techniques[J]. ACM Computer Surveys, 2005,37(3): 238-275.
- [7] VENKATACHARY S, LAKSHMINARAYANAN K, RANGARAJAN A. Algorithms for advanced packet classification with ternary cams[J]. ACM SIGCOMM Computer Communication Review, 2005 35(4): 193-204.
- [8] BREMLER-BARR A, HENDLER D. Space-efficient TCAM-based classification using gray coding[A]. INFOCOM 2007, The 26th IEEE International Conference on Computer Communications[C]. 2007. 1388-1396.
- [9] LIU H. Efficient mapping of range classifier into ternary-cam[A]. High Performance Interconnects[C]. 2002.95-100.
- [10] BREMLER-BARR A, HAY D, HENDLER D, *et al.* Layered interval codes for tcam-based classification[A]. INFOCOM 2009, the 28th IEEE International Conference on Computer Communications[C]. 2009.1305-1313.
- [11] TAYLOR D E, TURNER J S. ClassBench: a packet classification benchmark[J]. IEEE/ACM Transactions on Networking, 2007, 15(3): 499-511.

作者简介：



朱国胜（1972-），男，湖北大冶人，华中科技大学博士生，主要研究方向为下一代互联网体系结构和高速网络系统设计。



余少华（1962-），男，湖北武汉人，华中科技大学教授、博士生导师，武汉邮电科学研究院总工程师，主要研究方向为光通信和 IP 网络技术。

（上接第 30 页）

- [14] 高飞, 童恒庆. 基于改进粒子群优化算法的混沌系统参数估计方法[J]. 物理学报, 2006, 55(2): 577-582.
GAO F, TONG H Q. Parameter estimation for system based on particle swarm optimization[J]. Acta Phys, 2006, 55(2):577-582.
- [15] LI L X, PENG H P, WANG X D, *et al.* An optimization method inspired by chaotic ant behavior[J]. International Journal of Bifurcation and Chaos, 2006, (16): 2351-2364.
- [16] SOLE R V, MIRAMONTES O, GOODWIN B C. Oscillations and chaos in ant societies[J]. Journal of Theoretical Biology, 1993, 161(3): 343-357.



郑康锋（1975-），男，山东烟台人，北京邮电大学讲师，主要研究方向为网络与信息安全。

李丹（1986-），女，甘肃陇西人，北京邮电大学博士生，主要研究方向为信息安全、人工智能算法。

武斌（1981-），男，山东泰安人，北京邮电大学讲师，主要研究方向为网络与信息安全。

作者简介：



胥小波（1985-），男，四川盐亭人，北京邮电大学博士生，主要研究方向为信息安全、人工智能算法。



杨义先（1961-），男，四川盐亭人，北京邮电大学教授、博士生导师，主要研究方向为信息安全、密码学。