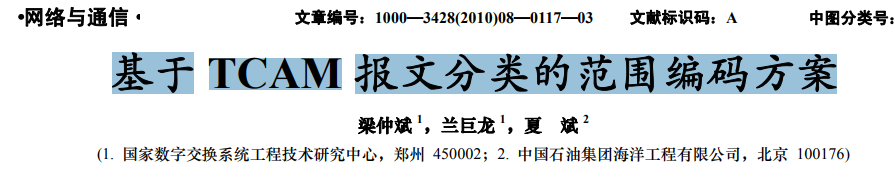
# DxR 路由查找

# 范围搜索 (Range Query)

* 基于 TCAM 的范围匹配方法——C-TCAM
* 一种基于 TCAM 的有效包分类方法



* 基于 TCAM 报文分类的范围编码方案



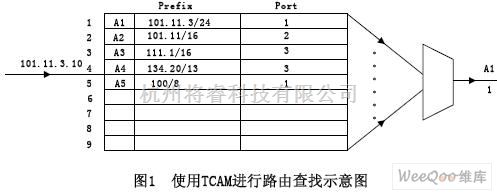
* **[针对范围对的高效查找算法设计（不准用数组）](https://www.cnblogs.com/lanxuezaipiao/p/3802699.html" \o "发布于 2014-06-22 19:52)**https://www.cnblogs.com/lanxuezaipiao/p/3802699.html
* **[使用二分查找判断某个数在某个区间中--如何判断某个IP地址所属的地区](https://www.cnblogs.com/hapjin/p/7252898.html" \o "发布于 2017-07-28 22:22)**

<https://www.cnblogs.com/hapjin/p/7252898.html>

# tcam更新效率低

由于TCAM仅简单地将地址最低的匹配表项的存储地址作为结果（索引）输出，要保障最长前缀匹配，表项的存储必须按前缀长度相对地址降序排列，即低地址存储前缀较长的表项，高地址存储前缀较短的表项。这种存储顺序关系使得TCAM的更新操作变得复杂。当加入一条新的前缀表项时，为了能仍然保持表项间的顺序关系，就需要移动一些前缀表项。

图1是使用TCAM进行路由查找的示意图。表项长度是按路由前缀的长度降序排列。假设 为目的地址101.11.3.10的ip报文查找转发路径。TCAM同时将它保存的所有表项与关键字101.11.3.10进行匹配查找，表项A1，A2都与关键字匹配，但是TCAM返回地址最小的表项， 即A1。

  
  
　　路由表是动态的，也就是说路由表项会随着网络拓扑结构的不断变化而相应的增加或者 删除。一般来说，在路由更新的同时，路由查找是不能够进行的，在这段时间内报文需要缓 存在报文缓冲区内等待路由更新的完成，因此慢的路由更新对系统报文缓冲区的容量有很大 的要求，同时也会延长报文转发的时间。所以要尽可能的减小路由更新的时间。

　　由于TCAM需要维持所有的路由表项按照前缀长度有序，所以对于路由的动态更新来说， 效率就会比较低。以图1为例，假设现在需要在转发表中增加新的表项101.11.128/18，按照 表项组织的方式，新的表项应该保存在表项101.11.3/24（A1）和表项101.11/16（A2）之间， 但是目前在这两个表项之间没有空闲的表项空间，所以需要通过移动其它表项为新表项腾出空间。下一小节我们给出一种较好的表项管理算法，可以有效降低表项更新的开销。

# 盛科tcam

* TCAM背景

当前商业芯片对OpenFlow的支持最欠缺的表现在:流表大小、流表数量、流表动作三方面。如严格按照OpenFlow标准，则流表只能用TCAM来实现。那我们简单讨论下什么是TCAM？为什么需要用TCAM来实现流表？

* 概念

TCAM是ternary(三重的) content addressable memory的缩写，中文是:三重内容可寻址内存。一般memory是根据地址来访问存储的内容，而TCAM恰好相反，它是根据内容去找到地址。一般的memory中每个bit都只能表示两个值:0或1,而TCAM每个bit可表示三个值:0，1和X，X表示don’t care。

* TCAM之所以一个bit能表示三个值，原因是它的一条entry其实在物理上由两条entry组成，一条存放data，另外一条存放相应的mask。由此可以实现0/1/X的表示。有了TCAM,就可支持各种LOOKUP KEY的任何组合查找，任意地MASK掉任何不关心的字段。只有TCAM可以做这样的事情，HASH不行。
* 如果有多条TCAM ENTRY都能匹配上，TCAM优先选择INDEX最小的那一条。TCAM的查找效率与深度无关，也就是说，无论表项多大，查找速度是一样的，这跟TCAM的硬件实现方式有关。
* TCAM可内置在芯片里面，也可外挂在芯片之外，芯片通过接口去访问。但是内置TCAM比较占芯片面积，一条TCAM ENTRY至少抵得上5条DRAM/SRAM ENTRY,如果芯片内部放了太大的TCAM会导致芯片成本和功耗直线上升。所以一般芯片都不会内置太大TCAM。
* 外挂TCAM的成本和功耗一样很大，甚至比ASIC芯片本身还要贵，而且如果一个报文要访问外部TCAM多次的话，受限于TCAM接口速率，会导致无法线速。所以交换机一般也都不会使用外挂TCAM。
* 普通交换机中的ACL功能必须使用TCAM，所以内置TCAM省不掉，而对于路由，如果也必须使用TCAM的话，路由表项做不大。正因为TCAM的高成本和高功耗，所以现在最新的商业芯片已经开始使用算法来支持路由了，盛科的GreatBelt芯片是最早使用算法来做路由的商业芯片。
* 如果OpenFlow交换机必须使用TCAM的话，成本和功耗会居高不下，自然会影响OF交换机的推广。盛科在ONS 2013上获得SDN IDOL的V350,就是针对这一点做了创新。 GreatBelt芯片采用了N-FLOW技术，TCAM+HASH相结合，HASH采用哪种KEY是可配的，HASH FLOW高达64K。
* TCAM entry size一般有多种，比如某著名厂家的有single, double, quad三种宽度，single只能放一点L2或者L3的常用字段，double可以放大部分L2或者L3或者常用的L2+L3字段，而quad则可以放基本上所有的字段。要完全满足Openflow的灵活需求，至少要quad，他们这个做法仅适用于ACL。
* 假设TCAM entry宽度全部配置成single可以支持N条的话，那么配制成double就可以支持N/2, 配制成quad就可以支持N/4.这就是为什么有的设备厂商宣称他们支持N条flow，但是实测只有N/2,或者N/4，因为他们都按照最理想情况宣称，但是实际测试的时候，人家都用最坏情况测试。
* 盛科芯片内置的使用IBM技术的TCAM，功耗可以做到很低。在entry宽度上也跟其它厂家稍微有所不同，它设计了四种宽度，分别是80/160/320/640bits。常用L2 or IPv4字段用80就可以，完整的L2 or IPv4用160，完整的L2+L3 or IPv6用320,完整的L2+IPv6用640,在字段组合上也跟别家不同，更注重灵活性。

# 基于TCAM的路由查找及表项管理

* 背景

为了取出RAM中的数据，操作系统必须给出存储数据的内存地址才能取出数据，CAM正相反， CAM通过对内容本身执行查询，CAM将此关键字与CAM中所有的表项同时进行匹配从而找到存储数据的地址，由于CAM的并行性使得它比RAM快很多。

而路由器查找有最长前缀匹配的特点，为了能够使用CAM来进行最长前缀路由的查找，我们可以为每一类可能的地址前缀长度使用一个CAM，每个CAM保存对应长度的所有前缀集合。对IPv4来说共需使用32个CAM，但是由于CAM仅提供两个结果：0或1，很明显，如果对地址前缀长度没有准确的了解，为了能够保证存储前N个前缀表，每个CAM都需要有N个表项的空间，因此CAM存储空间的利用率大大降低了。而为了克服该缺点，提出了TCAM。

* 什么是TCAM

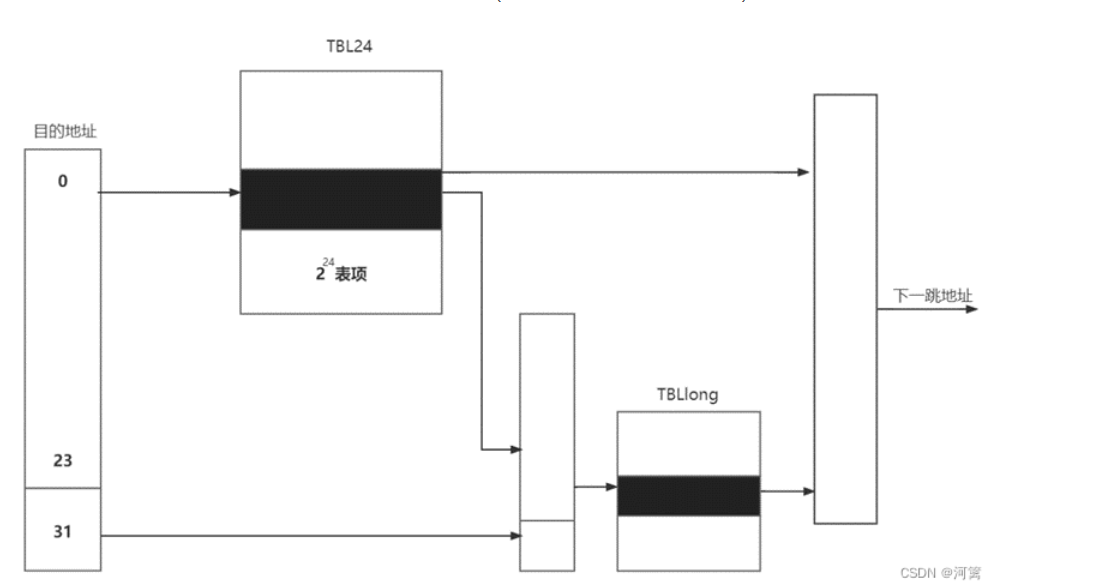
TCAM（ternary content-addressable memory）三态内容寻址存储器是一种专用的高速内存，能够在单个时钟周期内搜索出数据（IP地址）。

TCAM，ternary content-addressable memory 中ternary三态指的是内存利用三种不同的输入（0、1、X）来存储和查询数据，其中X表示“不在乎”状态（也称无关状态），使TCAM能够基于模式匹配进行更广泛的搜索：既能精确匹配查找，也能模糊匹配查找。而CAM仅仅用0和1来进行精确匹配。

* 基于TCAM的路由查找

1. 基于RAM的路由查找

最简单的基于RAM的路由查找方案是在RAM中为所有的IP地址都建立一个对应的转发表项，进行路由查找时，仅需要根据目的IP地址进行检索，一次访存就可以找到对应的路由信息，对IPv4来说，它所需要的转发表大小为232这会造成空间的极大浪费，因此Gupta等人在文献[1]中通过统计互联网中前缀长度的分布发现99.93%的前缀长度都分布在24比特或者小于24比特的范围内，因此他们提出了基于硬件的DIR-24-8-BASIC算法。DIR-24-8-BASIC算法结构如图1（DIR-24-8-BASIC算法结构图）：



DIR-24-8-BASIC算法原理：采用深度为2的多分支Trie树设计，其中第一层trie树步宽为24，对应的数据结构为TBL24表，主要保存的是路由地址前缀小于等于24的表项，因此一共含有16M个表项，第二层trie树步宽为24，对应的数据结构为TBLlong表，主要保存地址前缀大于24表项的下一跳的地址信息。路由查找时，先根据IP分组中目的地址前24个比特查找TBL24表，判断TBL24表输出的最高位，若为0，直接返回下一跳路由信息；若为1，则将其输出作为索引并和目的地址的后8位组成TBLlong的寻址地址，然后输出TBLlong相应单元内容，得到下一跳信息。

1. RAM的优点

算法简单，查表速度快，可用于高速路由器的设计实现。

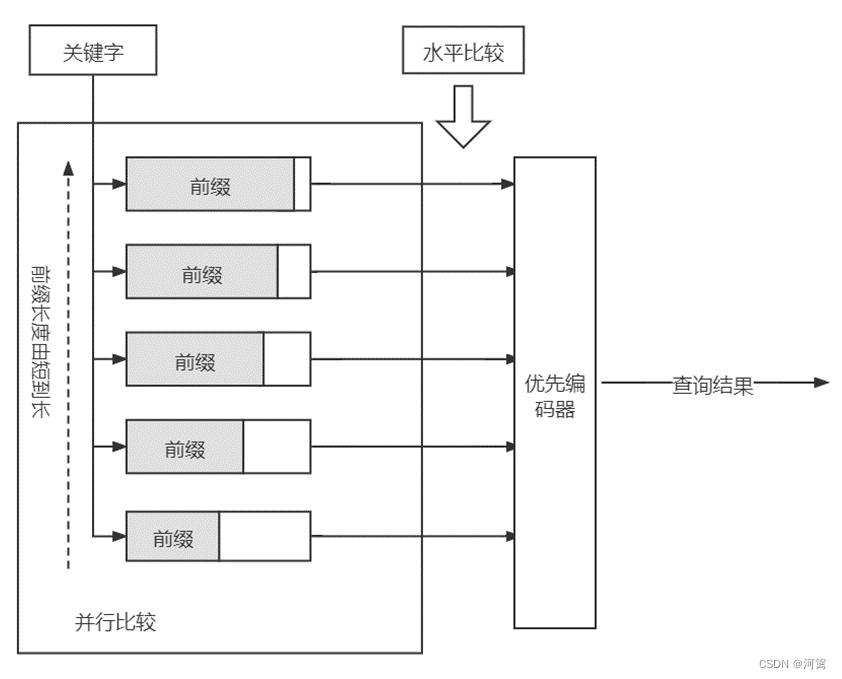
1. RAM的缺点

存储代价较高。

算法的更新复杂，路由改变时，表项更新单元数多，在最坏情况下，更新一个表项需要操作64K个存储单元。

* 基于TCAM的路由查找

TCAM芯片通过保存关键字掩码的方式保存任意长度的关键字表项，其中每一个表项都是以<地址，掩码>序偶的形式保存[3]：假设地址关键字的长度从1到N，那么地址和掩码分别占用N比特，对于关键字长度为Y（1≤Y≤N）的表项来说，它的地址后W-Y个比特可以是0或1；它的掩码的前Y个比特为1，剩下的N-Y个比特为0。如：假设N为5，那么地址前缀10\*就可以用<10000，11000>来表示。由于多个表项可能与输入的关键字匹配，因此TCAM规定在所有匹配的表项中选取地址最低的表项作为最后结果，而为了符合最长前缀路由的查找，在TCAM的低地址存储前缀较长的关键字，在高地址存储前缀较短的关键字。TCAM的查找结构如图2（TCAM查找结构图）。



1. TCAM的优点

查询速度快，实现简单，非常适合查找数据包或帧、决定发送数据包或帧的接口，常见于网络设备中，如高性能路由器和交换机，以提高路由查找、数据包分类、数据包转发等速度。

1. TCAM的缺点

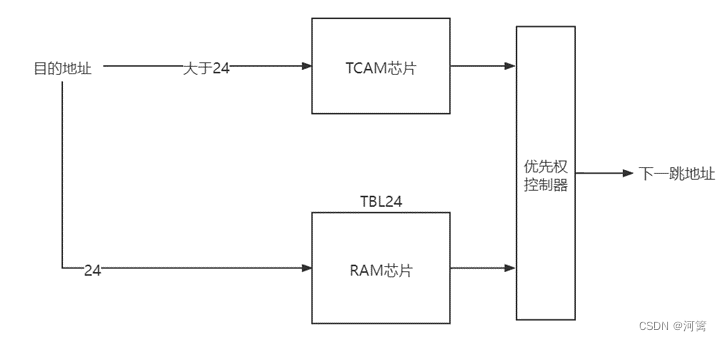
与RAM相比，TCAM更贵且存储芯片容量相对较小，制造成本高。

由于TCAM是并行匹配的方式，与所有的关键字表项都进行了比较，因此TCAM的功耗较大，散热性高。

TCAM的更新操作比较复杂，当插入一条新的表项时，为了能够保证最长前缀匹配，需要对前缀长度比新表项长的表项进行移动。

* 基于RAM和TCAM的路由查找

根据2.1和2.2的分析可知（即上文分析），基于RAM的查找和基于TCAM的查找速度都比较快且实现简单，而基于RAM的查找需要较高的存储代价，基于TCAM的查找仅适合小路由表的情况，因此，结合RAM和TCAM优点得到的高速路由查找算法[2]原理如图3（基于RAM和TCAM查找的算法原理图）：



TCAM芯片用于存储所有路由器地址前缀大于24的表项，TCAM的地址存储前缀较长的关键字表项，高地址区域存储前缀较短的关键字表项；同时为便于TCAM更新，将空闲的TCAM表项至于TCAM的中间，RAM芯片保存所有路由地址前缀小于等于24的表项，TCAM和RAM返回的信息都送到优先权控制器。目的IP地址到达后，检查TCAM中是否有匹配的目的地址，同时目的地址的前24位查找RAM中的TBL24表，返回TBL24对应的下一跳信息， 若TCAM和RAM同时有下一跳信息返回，则下一跳地址为TCAM返回的路由信息。

1. TCAM的表项管理

路由表会随着网络拓扑结构的变化而改变，一般来说，在路由更新的同时，路由查找是不能进行的，因此我们希望路由更新的时间越小越好，而不同的表项管理算法带来的更新操作开销是不同的。

* 顺序移动法

从TCAM的低地址开始顺序排序，将所有的空闲空间存放在TCAM的高地址，如果我们需要插入一个表项，最简单的方法就是将比插入地址高的表项依次向空闲空间移动一位，这种方法效率很低，最差情况下的算法复杂度为O(N)，N是目前TCAM中保存的表项数。

预留表项空间的顺序移动法

为了避免插入表项造成其他表项的大规模移动，可以为每个长度的前缀集合预留一些空闲的表项，当需要插入新表项时，如果对应前缀集合中包含空闲表项，则无需进行表项移动；如果不存在空闲表项，则需从相邻前缀集合中借用空闲表项。这种方法能够提高路由更新的平均效率，但是在最差情况下，更新算法的复杂度仍为O(N)。顺序移动法与预留表项空间的顺序移动法对比如图4所示（顺序移动法与预留表项空间的顺序移动法对比图）：

* 选择移动法

根据最长前缀匹配，TCAM要求所有的路由按照前缀长度降序排列，令Pj代表前缀长度为j的路由集合，令Pk代表前缀长度为k的路由集合，若j>k，则，Pj的路由表项应该在Pk之前，但TCAM对各个路由前缀内部的顺序关系没有严格规定，因此当需要在TCAM中加入长度为k(8≤k≤32)的路由前缀时，首先从长度为8的前缀块开始，将前缀块的第一项移动到最后一项，即TCAM的空闲表项区，然后再将长度为9的前缀块中的第一项移动到这一个空闲表项处，以此类推，直到新加入表项加入到分配处的空闲表项即可，这种算法复杂度为O(W)，W是路由前缀长度。

* 改进的选择移动法

将TCAM的空闲表项从TCAM的底部移动到TCAM的中间，算法的基本思想不变，此时算法的复杂度降低为O(W/2)。选择移动法与改进的选择移动法原理对比图如图5（选择移动法及其改进说明图）：