

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

H04L 12/56 (2006.01)

G06F 17/30 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200810088707.7

[43] 公开日 2009年11月4日

[11] 公开号 CN 101572647A

[22] 申请日 2008.4.30

[21] 申请号 200810088707.7

[71] 申请人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为
总部办公楼

[72] 发明人 龚 钧 詹 翀 沈士军 赵鸿翔

[74] 专利代理机构 北京集佳知识产权代理有限公司

代理人 梁明升 逯长明

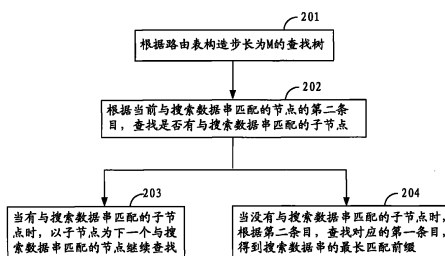
权利要求书 6 页 说明书 15 页 附图 4 页

[54] 发明名称

一种数据查找的方法及装置

[57] 摘要

本发明公开了一种数据查找的方法，该方法包括：根据当前与搜索数据串匹配的节点的第二条目，查找是否有与所述搜索数据串匹配的子节点；当有与所述搜索数据串匹配的子节点时，以所述子节点为下一个与搜索数据串匹配的节点继续查找；当没有与所述搜索数据串匹配的子节点时，根据所述第二条目，查找对应的第一条目，得到所述搜索数据串的最长匹配前缀。及一种数据查找的装置，根据每个节点中的项数不同，采用一个或多个长度为 L 的数据结构表示每个节点，通过这种灵活的方式，提高内存的利用率，进一步的提高网络容量。



1、一种数据查找的方法，用于查找最长匹配前缀，系统硬件支持的最大步长为 N (N 为自然数)，步长 N 对应的数据结构长度为 L ；其特征在于，

根据路由表构造步长为 M (M 为大于 N 的自然数) 的查找树，所述查找树的每个节点包括第一条目和第二条目，所述第一条目用于表示所述数组的前缀节点，所述第二条目用于表示所述数组的子节点，所述第二条目包括指向所述第一条目的指针；

当所述节点的前缀节点中的项数不小于 $2^N/M$ 时，所述第一条目采用二的 $M-N$ 次幂个长度为 L 的数据结构表示，对应的第一条目的数据结构类型为第一类型；当所述节点的前缀节点中的项数小于 $2^N/M$ 时，所述第一条目采用一个长度为 L 的数据结构表示，对应的第一条目的数据结构类型为第二类型；

当所述节点的子节点中的项数不小于 $2^N/M$ 时，所述第二条目采用二的 $M-N$ 次幂个长度为 L 的数据结构表示，对应的第二条目的数据结构类型为第一类型；当所述节点的子节点中的项数小于 $2^N/M$ 时，所述第二条目采用一个长度为 L 的数据结构表示，对应的第二条目的数据结构类型为第二类型；

该方法包括：

根据当前与搜索数据串匹配的节点的第二条目，查找是否有与所述搜索数据串匹配的子节点；

当有与所述搜索数据串匹配的子节点时，以所述子节点为下一个与搜索数据串匹配的节点继续查找；

当没有与所述搜索数据串匹配的子节点时，根据所述第二条目，查找对应的第一条目，得到所述搜索数据串的最长匹配前缀。

2、根据权利要求 1 所述的方法，其特征在于，所述根据当前与搜索数据串匹配的节点的第二条目，查找是否有与所述搜索数据串匹配的子节点包括：

当所述第二条目对应的数据结构类型为第一类型时，根据所述搜索数据串中对应的索引字段，找到所述第二条目的二的 $M-N$ 次幂个长度为 L 的数据结构中与所述搜索数据串对应的长度为 L 的数据结构；

在所述对应的长度为 L 的数据结构中，根据所述搜索数据串查找子节点位图；

当所述子节点位图指示存在子树时，则有与所述搜索数据串匹配的子节

点；当所述子节点位图指示不存在子树时，则没有与所述搜索数据串匹配的子节点。

3、根据权利要求1所述的方法，其特征在于，所述根据当前与搜索数据串匹配的节点的第二条目，查找是否有与所述搜索数据串匹配的子节点包括：

当所述第二条目对应的数据结构类型为第二类型时，在第二条目对应的长度为L的数据结构中，根据所述搜索数据串查找子节点位图；

当所述子节点位图指示存在子树时，则有与所述搜索数据串匹配的子节点；当所述子节点位图指示不存在子树时，则没有与所述搜索数据串匹配的子节点。

4、根据权利要求2或3所述的方法，其特征在于，所述当有与所述搜索数据串匹配的子节点时，以所述子节点为下一个与搜索数据串匹配的节点继续查找包括：

根据所述子节点位图，读取对应的子节点，以所述对应的子节点为下一个与搜索数据串匹配的节点继续查找。

5、根据权利要求1所述的方法，其特征在于，所述根据所述第二条目，查找对应的第一条目，得到所述搜索数据串的最长匹配前缀包括：

根据所述第二条目中指向所述第一条目的指针，读取对应的第一条目，得到所述搜索数据串的最长匹配前缀。

6、根据权利要求5所述的方法，其特征在于，所述读取对应的第一条目，得到所述搜索数据串的最长匹配前缀包括：

当所述第一条目对应的数据结构类型为第一类型时，根据所述搜索数据串中对应的索引字段，找到所述第一条目的二的M-N次幂个长度为L的数据结构中与所述搜索数据串对应的长度为L的数据结构；

在所述对应的长度为L的数据结构中，根据所述搜索数据串查找前缀节点位图；

当所述前缀节点位图指示存在前缀分布时，根据所述前缀分布得到所述搜索数据串的最长匹配前缀；当所述前缀节点位图指示不存在前缀分布时，以缺省下一跳索引作为所述搜索数据串的最长匹配前缀。

7、根据权利要求5所述的方法，其特征在于，所述读取对应的第一条目，

得到所述搜索数据串的最长匹配前缀包括:

当所述第一条目对应的数据结构类型为第二类型时,在第一条目对应的长度为 L 的数据结构中,根据所述搜索数据串查找前缀节点位图;

当所述前缀节点位图指示存在前缀分布时,根据所述前缀分布得到所述搜索数据串的最长匹配前缀;当所述前缀节点位图指示不存在前缀分布时,以缺省下一跳索引作为所述搜索数据串的最长匹配前缀。

8、根据权利要求 7 所述的方法,其特征在于,所述前缀节点位图用于表示前缀的地址范围,所述根据所述前缀分布得到所述搜索数据串的最长匹配前缀包括:

根据所述前缀节点位图记录的前缀的地址范围,计算偏移地址,以结果数组指针为基址,得到所述搜索数据串的最长匹配前缀。

9、根据权利要求 1 所述的方法,其特征在于,当所述步长为 M 的查找树中存在单一路径时,在所述单一路径的第一级节点的第二条目中设置跳转标志位和跳转字段,所述跳转标志位用于表示存在单一路径,所述跳转字段用于表示所述单一路径的路径信息;该方法还包括:

当所述当前与搜索数据串匹配的节点的第二条目中包括跳转标志位时,根据所述跳转字段,读取所述单一路径的最后一级节点,以所述单一路径的最后一级节点作为与所述搜索数据串匹配的子节点。

10、一种数据查找的装置,用于查找最长匹配前缀,该装置硬件支持的最大步长为 N (N 为自然数),步长 N 对应的数据结构长度为 L;其特征在于,该装置包括:

存储单元,用于存储根据路由表构造的步长为 M (M 为大于 N 的自然数)的查找树,所述查找树的每个节点包括第一条目和第二条目,所述第一条目用于表示所述数组的前缀节点,所述第二条目用于表示所述数组的子节点,所述第二条目包括指向所述第一条目的指针;

当所述节点的前缀节点中的项数不小于 $2^N/M$ 时,所述第一条目采用二的 M-N 次幂个长度为 L 的数据结构表示,对应的第一条目的数据结构类型为第一类型;当所述节点的前缀节点中的项数小于 $2^N/M$ 时,所述第一条目采用一个长度为 L 的数据结构表示,对应的第一条目的数据结构类型为第二类型;

当所述节点的子节点中的项数不小于 $2^N/M$ 时, 所述第二条目采用二的 $M-N$ 次幂个长度为 L 的数据结构表示, 对应的第二条目的数据结构类型为第一类型; 当所述节点的子节点中的项数小于 $2^N/M$ 时, 所述第二条目采用一个长度为 L 的数据结构表示, 对应的第二条目的数据结构类型为第二类型;

查找单元, 用于根据当前与搜索数据串匹配的节点的第二条目, 查找所述存储单元中是否有与所述搜索数据串匹配的子节点; 当有与所述搜索数据串匹配的子节点时, 以所述子节点为下一个与搜索数据串匹配的节点继续查找; 当没有与所述搜索数据串匹配的子节点时, 根据所述第二条目, 查找对应的第一条目, 得到所述搜索数据串的最长匹配前缀。

11、根据权利要求 10 所述的装置, 其特征在于, 当所述第二条目对应的数据结构类型为第一类型时, 所述查找单元包括:

第一子单元, 用于根据所述搜索数据串中对应的索引字段, 在所述存储单元中找到所述第二条目的二的 $M-N$ 次幂个长度为 L 的数据结构中与所述搜索数据串对应的长度为 L 的数据结构;

第二子单元, 用于在所述第一子单元找到的对应的长度为 L 的数据结构中, 根据所述搜索数据串查找子节点位图; 当所述子节点位图指示存在子树时, 则有与所述搜索数据串匹配的子节点; 当所述子节点位图指示不存在子树时, 则没有与所述搜索数据串匹配的子节点。

12、根据权利要求 10 所述的装置, 其特征在于, 当所述第二条目对应的数据结构类型为第二类型时, 所述查找单元包括:

第三子单元, 用于在所述存储单元存储的第二条目对应的长度为 L 的数据结构中, 根据所述搜索数据串查找子节点位图; 当所述子节点位图指示存在子树时, 则有与所述搜索数据串匹配的子节点; 当所述子节点位图指示不存在子树时, 则没有与所述搜索数据串匹配的子节点。

13、根据权利要求 10 所述的装置, 其特征在于, 当没有与所述搜索数据串匹配的子节点时, 所述查找单元包括:

第四子单元, 用于根据所述存储单元存储的第二条目中指向所述第一条目的指针, 读取对应的第一条目, 得到所述搜索数据串的最长匹配前缀。

14、根据权利要求 13 所述的装置, 其特征在于, 当所述第一条目对应的

数据结构类型为第一类型时, 所述第四子单元包括:

第一模块, 用于根据所述搜索数据串中对应的索引字段, 在所述存储单元中找到所述第一条目的二的 $M-N$ 次幂个长度为 L 的数据结构中与所述搜索数据串对应的长度为 L 的数据结构;

第二模块, 用于在所述第一模块找到的对应的长度为 L 的数据结构中, 根据所述搜索数据串查找前缀节点位图; 当所述前缀节点位图指示存在前缀分布时, 根据所述前缀分布得到所述搜索数据串的最长匹配前缀; 当所述前缀节点位图指示不存在前缀分布时, 以缺省下一跳索引作为所述搜索数据串的最长匹配前缀。

15、根据权利要求 14 所述的装置, 其特征在于, 当所述存储单元存储的路由表中对应的当前与搜索数据串匹配的节点的前缀节点中前缀连续时, 所述前缀节点位图仅记录连续前缀的起始地址和结束地址, 所述第二模块包括:

第一子模块, 用于根据所述第一模块找到的对应的长度为 L 的数据结构中的前缀节点位图记录的前缀地址和/或连续前缀的起始地址和结束地址, 计算偏移地址;

第二子模块, 用于以所述第一子模块计算得到的偏移地址, 并且以所述第一模块找到的对应的长度为 L 的数据结构中的结果数组指针为基址, 得到所述搜索数据串的最长匹配前缀。

16、根据权利要求 13 所述的装置, 其特征在于, 当所述第一条目对应的数据结构类型为第二类型时, 所述第四子单元包括:

第三模块, 用于在所述存储单元存储的第一条目对应的长度为 L 的数据结构中, 根据所述搜索数据串查找前缀节点位图;

第四模块, 用于当所述第三模块查找到的前缀节点位图指示存在前缀分布时, 根据所述前缀分布得到所述搜索数据串的最长匹配前缀; 当所述前缀节点位图指示不存在前缀分布时, 以缺省下一跳索引作为所述搜索数据串的最长匹配前缀。

17、根据权利要求 16 所述的装置, 其特征在于, 所述前缀节点位图用于表示前缀的地址范围, 所述第四模块包括:

第三子模块, 用于根据所述前缀节点位图记录的前缀的地址范围, 计算偏

移地址;

第四子模块,用于以所述第三子模块计算得到的偏移地址,并且以所述第三模块找到的对应的长度为L的数据结构中的结果数组指针为基址,得到所述搜索数据串的最长匹配前缀。

18、根据权利要求10所述的装置,其特征在于,当所述存储单元存储的步长为M的查找树中存在单一路径时,在所述单一路径的第一级节点的第二条目中设置跳转标志位和跳转字段,所述跳转标志位用于表示存在单一路径,所述跳转字段用于表示所述单一路径的路径信息;该装置还包括:

跳转单元,用于当所述存储单元存储的当前与搜索数据串匹配的节点的第二条目中包括跳转标志位时,控制所述查找单元根据所述跳转字段,读取所述单一路径的最后一级节点,以所述单一路径的最后一级节点作为与所述搜索数据串匹配的子节点。

一种数据查找的方法及装置

技术领域

本发明涉及通信领域，特别涉及一种数据查找的方法及装置。

背景技术

随着通信技术的发展和用户需求的增长，对网络的速度和容量的要求越来越高。为了提高网络的速度和容量，普遍方法是采用包交换技术，包交换技术主要是对包进行处理，在网络设备对包进行处理的过程中，需要对包的路由进行查找，对包的路由的查找速度直接影响到网络的速度和容量。路由查表进行最长前缀匹配（The Longest Prefix Match），即路由器在转发网际协议（Internet Protocol, IP）报文时，根据IP地址找出最长匹配的前缀后，根据该前缀所对应的下一跳信息转发IP报文。

路由表通常存储在一个层状数据结构中，搜索沿层次向下进行。一般采用的层状数据结构是二叉树。基本的二叉树搜索一步检查一个比特，若相应的地址前缀最长为M，则树的深度为M。如果搜索一步检查K个比特，则树的深度可减少到M/K，这样树的内部节点包含的匹配项增加为二的K次幂，这样的树被称为二的K次幂树，进行路由查表时，在每个节点处检查的比特数为K，K被成为树的步宽。

图1所示为多比特树（Multi-Bit Trie）的结构示意图。在图1中，匹配项为P1=*, P2=1*, P3=00*, P4=101*, P5=111*, P6=1000*, P7=11101*, P8=111001*, P9=1000011*。该树的步长为三，对于不足三比特的匹配项，将该匹配项展开至三比特，例如，P2=1*，该匹配项不足三比特，则将该匹配项展开至三比特（100, 101, 110, 111）。由于步长是三，因此每个节点包含二的三次幂项，也就是八项。在这八项中，每项包含两个内容，一个是该项的匹配项，另一个是该项的下一个节点的指针。

在实现本发明的过程中，发明人发现现有技术中至少存在如下问题：

- 1、查找的步长受到系统硬件的限制，从而限制了查找的速度。
- 2、每个节点都要申请固定的大小，即使该节点只包含一个匹配项或不包

含匹配项，这样在进行路由查表时，需要查找每个节点的每项，降低了查找速度，进而影响网络的速度；同时，每个节点都申请固定的大小，对内存资源造成浪费，并影响网络容量。

发明内容

本发明实施例在于提供数据查找的方法及装置，提高数据查找的速度，提高内存的利用率。

本发明实施例提供了一种数据查找的方法，用于查找最长匹配前缀，系统硬件支持的最大步长为 N (N 为自然数)，步长 N 对应的数据结构长度为 L ；

根据路由表构造步长为 M (M 为大于 N 的自然数) 的查找树，所述查找树的每个节点包括第一条目和第二条目，所述第一条目用于表示所述数组的前缀节点，所述第二条目用于表示所述数组的子节点，所述第二条目包括指向所述第一条目的指针；

当所述节点的前缀节点中的项数不小于 $2^N/M$ 时，所述第一条目采用二的 $M-N$ 次幂个长度为 L 的数据结构表示，对应的第一条目的数据结构类型为第一类型；当所述节点的前缀节点中的项数小于 $2^N/M$ 时，所述第一条目采用一个长度为 L 的数据结构表示，对应的第一条目的数据结构类型为第二类型；

当所述节点的子节点中的项数不小于 $2^N/M$ 时，所述第二条目采用二的 $M-N$ 次幂个长度为 L 的数据结构表示，对应的第二条目的数据结构类型为第一类型；当所述节点的子节点中的项数小于 $2^N/M$ 时，所述第二条目采用一个长度为 L 的数据结构表示，对应的第二条目的数据结构类型为第二类型；

该方法包括：

根据当前与搜索数据串匹配的节点的第二条目，查找是否有与所述搜索数据串匹配的子节点；

当有与所述搜索数据串匹配的子节点时，以所述子节点为下一个与搜索数据串匹配的节点继续查找；

当没有与所述搜索数据串匹配的子节点时，根据所述第二条目，查找对应的第一条目，得到所述搜索数据串的最长匹配前缀。

本发明实施例还提供了一种数据查找的装置，用于查找最长匹配前缀，该

装置硬件支持的最大步长为 N (N 为自然数), 步长 N 对应的数据结构长度为 L ; 其特征在于, 该装置包括:

存储单元, 用于存储根据路由表构造的步长为 M (M 为大于 N 的自然数) 的查找树, 所述查找树的每个节点包括第一条目和第二条目, 所述第一条目用于表示所述数组的前缀节点, 所述第二条目用于表示所述数组的子节点, 所述第二条目包括指向所述第一条目的指针;

当所述节点的前缀节点中的项数不小于 $2^N/M$ 时, 所述第一条目采用二的 $M-N$ 次幂个长度为 L 的数据结构表示, 对应的第一条目的数据结构类型为第一类型; 当所述节点的前缀节点中的项数小于 $2^N/M$ 时, 所述第一条目采用一个长度为 L 的数据结构表示, 对应的第一条目的数据结构类型为第二类型;

当所述节点的子节点中的项数不小于 $2^N/M$ 时, 所述第二条目采用二的 $M-N$ 次幂个长度为 L 的数据结构表示, 对应的第二条目的数据结构类型为第一类型; 当所述节点的子节点中的项数小于 $2^N/M$ 时, 所述第二条目采用一个长度为 L 的数据结构表示, 对应的第二条目的数据结构类型为第二类型;

查找单元, 用于根据当前与搜索数据串匹配的节点的第二条目, 查找所述存储单元中是否有与所述搜索数据串匹配的子节点; 当有与所述搜索数据串匹配的子节点时, 以所述子节点为下一个与搜索数据串匹配的节点继续查找; 当没有与所述搜索数据串匹配的子节点时, 根据所述第二条目, 查找对应的第一条目, 得到所述搜索数据串的最长匹配前缀。

采用本发明实施例的技术方案, 在不改变系统硬件的前提下, 采用现有技术中最大步长对应长度的数据结构表示更大步长的查找树, 从而减少数据查找的级数, 提高数据查找的速度。

现有技术采用固定长度数据结构表示每个节点, 即使该节点只包含一个匹配项或不包含匹配项, 本发明实施例根据每个节点中的项数不同, 采用一个或多个长度为 L 的数据结构表示每个节点, 通过这种灵活的方式, 提高内存的利用率, 进一步的提高网络容量。

附图说明

图 1 所示为多比特树的结构示意图;

图 2 所示为本发明实施例一中数据查找的方法的流程示意图；
图 3 所示为本发明实施例二中 Child Node 的结构示意图；
图 4 所示为本发明实施例二中 Prefix Node 的结构示意图；
图 5 所示为本发明实施例二中直接查找表的结构示意图；
图 6 所示为本发明实施例二中数据查找的方法的流程示意图；
图 7 所示为本发明实施例二中另一种 Child Node 的结构示意图；
图 8 所示为本发明实施例三中数据查找的装置的结构示意图。

具体实施方式

图 2 所示为本发明实施例一中数据查找的方法的流程示意图。

一种数据查找的方法，用于查找最长匹配前缀，系统硬件支持的最大步长为 N (N 为自然数)，步长 N 对应的数据结构长度为 L 。

201、根据路由表构造步长为 M (M 为大于 N 的自然数) 的查找树。

构造的查找树的每个节点包括第一条目和第二条目，第一条目用于表示数组的前缀节点，第二条目用于表示数组的子节点，第二条目包括指向第一条目的指针；在本实施例中，约定第一条目的名称为 Prefix Node，第二条目的名称为 Child Node，此约定仅为了下文中描述技术方案方便起见，不能认为是对第一条目以及第二条目的限定。

当节点的前缀节点中的项数不小于 $2^N/M$ 时，第一条目采用二的 $M-N$ 次幂个长度为 L 的数据结构表示，对应的第一条目的数据结构类型为第一类型；当节点的前缀节点中的项数小于 $2^N/M$ 时，第一条目采用一个长度为 L 的数据结构表示，对应的第一条目的数据结构类型为第二类型；在本实施例中，约定第一类型的名称为 Segment，第二类型的名称为 Compact，此约定仅为了下文中描述技术方案方便起见，不能认为是对第一类型以及第二类型的限定。

当节点的子节点中的项数不小于 $2^N/M$ 时，第二条目采用二的 $M-N$ 次幂个长度为 L 的数据结构表示，对应的第二条目的数据结构类型为第一类型；当节点的子节点中的项数小于 $2^N/M$ 时，第二条目采用一个长度为 L 的数据结构表示，对应的第二条目的数据结构类型为第二类型；

202、根据当前与搜索数据串匹配的节点的第二条目，查找是否有与搜索

数据串匹配的子节点。

搜索数据串为用于搜索最长匹配前缀的数据串，包括但不限于网络协议（IP）地址或者其他可用于查找路由的数据串。

对应当前与搜索数据串匹配的节点中的 Child Node 的两种类型，有如下两种方式查找：

类型为 Segment：

1、根据搜索数据串中对应的索引字段，找到第二条目的二的 M-N 次幂个长度为 L 的数据结构中，与搜索数据串对应的长度为 L 的数据结构。

类型为 Segment 的 Child Node，包括二的 M-N 次幂个长度为 L 的数据结构，根据搜索数据串中对应的索引字段直接从中找到一个与搜索数据串对应的长度为 L 的数据结构，并针对该对应的长度的 L 的数据结构进行后续的查找操作，从而提高查找的速度。

2、在对应的长度为 L 的数据结构中，根据搜索数据串查找子节点位图。

类型为 Segment 的 Child Node 的子节点位图包括位置字段和类型字段，其中，约定位置字段的名称为 Child Bitmap，类型字段的名称为 Child Type，此约定仅为了下文中描述技术方案方便起见，不能认为是对类型为 Segment 的 Child Node 中位置字段以及类型字段的限定。

Child Bitmap 用于表示对应的位置的子节点是否存在，Child Type 用于表示对应的位置的子节点存在时，子节点的第二条目的数据结构类型是第一类型还是第二类型。

3、当子节点位图指示存在子树时，则有与搜索数据串匹配的子节点；当子节点位图指示不存在子树时，则没有与搜索数据串匹配的子节点。

类型为 Compact：

1、在第二条目对应的长度为 L 的数据结构中，根据搜索数据串查找子节点位图。

类型为 Compact 的 Child Node 的子节点位图包括位置字段和类型字段，其中，约定位置字段的名称为 Child i (i 是子节点的个数)，类型字段的名称为 Type，此约定仅为了下文中描述技术方案方便起见，不能认为是对类型为 Compact 的 Child Node 中位置字段以及类型字段的限定。

Child i 用于记录存在子节点的位置, Type 用于表示子节点的第二条目的数据结构类型是第一类型还是第二类型。

2、当子节点位图指示存在子树时, 则有与搜索数据串匹配的子节点; 当子节点位图指示不存在子树时, 则没有与搜索数据串匹配的子节点。

203、当有与搜索数据串匹配的子节点时, 以子节点为下一个与搜索数据串匹配的节点继续查找。

对应当前与搜索数据串匹配的节点中的 Child Node 的两种类型, 以子节点为下一个与搜索数据串匹配的节点继续查找有如下两种方式:

类型为 Segment:

根据子节点位图, 读取对应的子节点, 以对应的子节点为下一个与搜索数据串匹配的节点继续查找。

具体的可以包括: 根据 Child Bitmap 和 Child Type 计算偏移地址, 以 Child Node 中的子节点数组指针 (Child Array Pointer) 作为基址, 用基址+偏移地址的方式读取对应的子节点。

类型为 Compact:

根据子节点位图, 读取对应的子节点, 以对应的子节点为下一个与搜索数据串匹配的节点继续查找。

具体的可以包括: 将搜索数据串与 Child i 比较, 如果存在相同项, 则根据该相同项之前 Child i 的个数和 相应的 Type 得到偏移地址, 以 Child Node 中的 Child Array Pointer 作为基址, 用基址+偏移地址的方式读取对应的子节点。

204、当没有与搜索数据串匹配的子节点时, 根据第二条目, 查找对应的第一条目, 得到搜索数据串的最长匹配前缀。

根据第二条目, 查找对应的第一条目的方式可以是: 根据第二条目中指向第一条目的指针, 读取对应的第一条目。约定第二条目中指向第一条目的指针名称为 Prefix Type, Prefix Type 用于表示对应的第一条目的数据结构类型为第一类型还是第二类型; 此约定仅为了下文中描述技术方案方便起见, 不能认为是对第二条目中指向第一条目的指针的限定。

具体的, 对应当前与搜索数据串匹配的节点中的 Child Node 的两种类型, 读取对应的第一条目可以有如下两种方式:

Child Node 类型为 Segment:

根据 Child Bitmap 和 Child Type 计算偏移地址，以 Child Node 中的 Child Array Pointer 作为基址，根据 Prefix Type，用基址+偏移地址的方式读取对应的第一条目。

Child Node 类型为 Compact:

根据 Prefix Type，用 Child Node 中的 Child Array Pointer 作为基址，读取对应的第一条目。

对应当前与搜索数据串匹配的节点中的 Prefix Node 的两种类型，有如下两种方式得到搜索数据串的最长匹配前缀:

类型为 Segment:

1、根据搜索数据串中对应的索引字段，找到第一条目的二的 M-N 次幂个长度为 L 的数据结构中，与搜索数据串对应的长度为 L 的数据结构。

类型为 Segment 的 Prefix Node，包括二的 M-N 次幂个长度为 L 的数据结构，根据搜索数据串中对应的索引字段直接从中找到一个与搜索数据串对应的长度为 L 的数据结构，并针对该对应的长度的 L 的数据结构进行后续的查找操作，从而提高查找的速度。

2、在对应的长度为 L 的数据结构中，根据搜索数据串查找前缀节点位图。

约定类型为 Segment 和 Compact 的 Prefix Node 中的前缀节点位图的名称为 Prefix Code，Prefix Code 用于表示前缀节点内前缀的编码；此约定仅为了下文中描述技术方案方便起见，不能认为是对前缀节点位图的限定。

3、当前缀节点位图指示存在前缀分布时，根据前缀分布得到搜索数据串的最长匹配前缀；当前缀节点位图指示不存在前缀分布时，以缺省下一跳索引作为搜索数据串的最长匹配前缀。

根据搜索数据串查找 Prefix Code，如果存在前缀分布，则根据 Prefix Code 计算偏移地址，以 Prefix Node 中的结果数组指针 (Result Array Pointer) 作为基址，得到下一跳地址作为搜索数据串的最长匹配前缀。

如果不存在前缀分布，以缺省下一跳索引作为搜索数据串的最长匹配前缀。其中，缺省下一跳索引可以是 Prefix Node 中携带的，也可以是系统缺省的。

类型为 Compact:

1、在第一条目对应的长度为 L 的数据结构中，根据搜索数据串查找前缀节点位图。

类型为 Compact 的 Prefix Node 中 Prefix Code 用于表示每一个前缀的地址范围，具体的，可以记录不同前缀的起始地址，并采用标志位字段表示该地址是起始还是结束，当路由表中对应的当前与搜索数据串匹配的节点的前缀节点中前缀连续时，前缀节点位图仅记录连续前缀的起始地址和结束地址，查找时无需读取中间前缀，进一步提高查找的速度。

2、当前缀节点位图指示存在前缀分布时，根据前缀分布得到搜索数据串的最长匹配前缀；当前缀节点位图指示不存在前缀分布时，以缺省下一跳索引作为搜索数据串的最长匹配前缀。

具体的可以包括：将搜索数据串与 Prefix Code 比较，看是否落在 Prefix Code 区间段之内，如果是，则表示存在前缀分布，根据 Prefix Code 计算偏移地址，以 Prefix Node 中的 Result Array Pointer 作为基址，得到下一跳地址作为搜索数据串的最长匹配前缀；如果不是，则表示不存在前缀分布，以缺省下一跳索引作为搜索数据串的最长匹配前缀。其中，缺省下一跳索引可以是 Prefix Node 中携带的，也可以是系统缺省的。

Prefix Code 用于记录每一个前缀的地址范围，在 Compact 的数据结构中，记录的是不同前缀的起始地址，并包括标志位字段表示该地址是起始还是结束。根据所述前缀节点位图记录的前缀的地址范围，计算偏移地址，以结果数组指针为基址，得到所述搜索数据串的最长匹配前缀。当存在连续前缀时，Prefix Code 记录该连续前缀的第一个前缀的起始地址和最后一个前缀的结束地址，而无需记录中间前缀的地址范围，节省了存储空间的同时，也提高了查找速度。

采用本实施例的技术方案，在不改变系统硬件的前提下，采用现有技术中最大步长对应长度的数据结构表示更大步长的查找树，从而减少数据查找的级数，提高数据查找的速度。

现有技术采用固定长度数据结构表示每个节点，即使该节点只包含一个匹配项或不包含匹配项，本发明实施例根据每个节点中的项数不同，采用一个或

多个长度为 L 的数据结构表示每个节点，通过这种灵活表示方式，提高内存的利用率，进一步的提高网络容量。

进一步的，当存在单一路径时，为了减少查找级数，提高查找速度，设置跳转标志位和跳转字段。当步长为 M 的查找树中存在单一路径时，在单一路径的第一级节点的第二条目中设置跳转标志位和跳转字段，跳转标志位用于表示存在单一路径，跳转字段用于表示单一路径的路径信息；该方法还可以包括：

当当前与搜索数据串匹配的节点的第二条目中包括跳转标志位时，根据跳转字段，读取单一路径的最后一节点，以单一路径的最后一节点作为与搜索数据串匹配的子节点。

实施例二是实施例一中数据查找的方法的具体应用。为便于技术方案的描述，本实施例沿用实施例一中的约定。

Child Node 和 Prefix Node 均有 Segment 和 Compact 两种表示方式。因为有这两种表示方式，所以在查找过程当中就需要告知下一节点是采用何种表示方式，于是在相应的数据结构中设置相关表示字段。

图 3 所示为本发明实施例二中 Child Node 的结构示意图。

Segment 的表示方式下，图 3 中示出的仅为一个长度为 L 的数据结构的结构示意图，其余长度为 L 的数据结构的结构相同。其中前缀类型 (Prefix Type) 表示本节点前缀是 Segment 还是 Compact 表示方式；继承最长前缀匹配 (Parent Longest Prefix Match) 表示上一级中到本节点的最长匹配前缀，其余长度为 L 的数据结构中，这个字段应该是相同的；子节点数组指针 (Child Array Pointer) 指向本节点的 Prefix Node，Prefix Node 后紧跟 Child Node，连续存放的；Child Bitmap 和 Child Type 表示的是对应位置的 Child Node 是否存在以及 Segment/Compact 表示方式。

Compact 的表示方式下，与 Segment 的表示方式中的不同之处在于 Child 字段直接记录存在子节点的位置，Type 字段为子节点的表示方式。

图 4 所示为本发明实施例二中 Prefix Node 的结构示意图。Segment 和 Compact 两种表示方式下，每个长度为 L 的数据结构的结构相同。其中，缺省下一跳索引 (Default Next Hop Index) 表示从上一级 Push 下来的前缀，在

Segment 表示中,也有可能是同一级前几个长度为 L 的数据结构中下压的前缀;结果数组指针 (Result Array Pointer) 指向本节点内前缀对应下一跳存放的起始地址,偏移地址通过对 Prefix Code 解码得到。Prefix Code 是对节点内前缀的编码,在 Compact 表示下,记录的是不同前缀的起始地址,另外还有标志位字段表示该地址起始还是结束。

图 5 所示为本发明实施例二中直接查找 (DT) 表的结构示意图。用于表示根节点的数据结构,包括子节点位图、子节点类型、下一跳索引和子节点指针,根节点不属于本发明实施例中限定的节点。

图 6 所示为本发明实施例二中数据查找的方法的流程示意图。其中,三角形表示的是查找树中的节点。

在本实施例中,搜索数据串为 IP 地址;系统硬件支持的最大步长为 5 (即 N 等于 5), M 等于 8, Segment 方式下长度为 L 的数据结构的个数为 8; IP 地址中用于找到 8 个长度为 L 的数据结构中 & IP 地址对应的长度为 L 的数据结构的索引字段长度为 3bit。

601、根据 Table ID 和待查 IP 的前 9 位数据在 DT 表中查找,可以得到 Next Hop Index 以及 Child Node 的表示类型和入口地址,如果不存在 Child Node 直接返回 Next Hop Index,查找结束;否则将 Next Hop Index 记录为缺省值,进入下一步。

602、取待查 IP 地址接下来的 8bit,查找 Child Type,根据表示方式读取相应的 Child Node。

603、查找 Parent Longest Prefix Match,如果存在修改缺省值为 Parent Longest Prefix Match。

604、查找是否有子节点,如果没有子节点,直接跳到 606;否则进入下一步。

605、查找 Child Node 节点,针对 Segment 和 Compact 表示有两种读取方式:

a) Compact: 把 8bit 数据依次与 Child i (本实施例中 i 等于 8) 比较,如果存在相同项则统计之前 Child Node 的个数以及相应的表示方式得到偏移地址,再以 Child Array Pointer 作为基址,根据相应的 Child Type 读取子节点中的 Child

Node, 跳到 603; 如果不存在相同项, 则根据 Prefix Type 字段, 再以 Child Array Pointer 为基址, 读取 Prefix Node, 跳到 606。

b) Segment: 以 8bit 数据中的前 3 位作为索引字段, 从 8 个长度为 L 的数据结构中找到与 IP 地址对应的一个长度为 L 的数据结构; 以 8bit 数据中的后 5 位来检查该对应的长度为 L 的数据结构中 Child Bitmap 是否存在子树, 如果不存在, 依据 Child Bitmap 和 Child Type 来计算偏移地址, 再以 Child Array Pointer 作为基址, 读取相应的 Prefix Node, 跳到 606; 如果存在, 则根据之前的 Child Bitmap 和 Child Type, 同样采取基址+偏移地址的方式读取 Child Node, 跳到 603。

606、查找 Prefix Node, 也有两种读取方式: Segment 和 Compact。

a) Compact: 将 8bit 数据依次与 Prefix Code 中记录的 Prefix 1~8 比较, 看是否落在该区间段之内, 如果是计算偏移地址, 以 Result Array Pointer 为基址, 返回 Next Hop 地址, 查找结束; 否则进入下一步。

b) Segment: 将 8bit 数据在 Prefix Code 查找, 看是否存在前缀分布, 如果存在则计算偏移地址, 以 Result Array Pointer 为基址, 返回 Next Hop 地址, 查找结束; 否则进入下一步。

607、查找是否存在 Default Next Hop Index, 如果存在直接返回, 查找结束; 否则返回缺省值, 查找结束。

图 7 所示为本发明实施例二中另一种 Child Node 的结构示意图。

为了在单一路径时进一步减少查找级数, 提高查找速度, 还可以在 Child Node 中加入跳转标志位和跳转字段, 跳转标志位用于表示存在单一路径, 跳转字段用于表示所述单一路径的路径信息; 在本实施例中约定跳转标志位的名称为 Skip Flag, 跳转字段的名称为 Skipped, 此约定仅为了下文中描述技术方案方便起见, 不能认为是对跳转标志位以及跳转字段的限定。

在以上算法中, 在 605 之前多一个 Skip 节点的判断:

先检查 Skip Flag, 如果存在单一路径的情况, 根据 Skip Flag 的值, 取待查 IP 数据接下来的 8bit, 将其与 Skipped 进行比较, 如果不相同, 则返回缺省值, 查找结束; 如果相同, 再进入下一步。

图8所示为本发明实施例三中数据查找的装置的结构示意图。该数据查找的装置是用于执行实施例一中数据查找的方法的设备，可以独立设置，也可以设置在网络设备中。

一种数据查找的装置，用于查找最长匹配前缀，该装置硬件支持的最大步长为 N (N 为自然数)，步长 N 对应的数据结构长度为 L ；该装置可以包括：

存储单元801，用于存储根据路由表构造的步长为 M (M 为大于 N 的自然数)的查找树，所述查找树的每个节点包括第一条目和第二条目，所述第一条目用于表示所述数组的前缀节点，所述第二条目用于表示所述数组的子节点，所述第二条目包括指向所述第一条目的指针；

当所述节点的前缀节点中的项数不小于 $2^N/M$ 时，所述第一条目采用二的 $M-N$ 次幂个长度为 L 的数据结构表示，对应的第一条目的数据结构类型为第一类型；当所述节点的前缀节点中的项数小于 $2^N/M$ 时，所述第一条目采用一个长度为 L 的数据结构表示，对应的第一条目的数据结构类型为第二类型；

当所述节点的子节点中的项数不小于 $2^N/M$ 时，所述第二条目采用二的 $M-N$ 次幂个长度为 L 的数据结构表示，对应的第二条目的数据结构类型为第一类型；当所述节点的子节点中的项数小于 $2^N/M$ 时，所述第二条目采用一个长度为 L 的数据结构表示，对应的第二条目的数据结构类型为第二类型；

查找单元802，用于根据当前与搜索数据串匹配的节点的第二条目，查找所述存储单元中是否有与所述搜索数据串匹配的子节点；当有与所述搜索数据串匹配的子节点时，以所述子节点为下一个与搜索数据串匹配的节点继续查找；当没有与所述搜索数据串匹配的子节点时，根据所述第二条目，查找对应的第一条目，得到所述搜索数据串的最长匹配前缀。

进一步的，当所述第二条目对应的数据结构类型为第一类型时，所述查找单元可以包括：

第一子单元，用于根据所述搜索数据串中对应的索引字段，在所述存储单元中找到所述第二条目的二的 $M-N$ 次幂个长度为 L 的数据结构中与所述搜索数据串对应的长度为 L 的数据结构；

第二子单元，用于在所述第一子单元找到的对应的长度为 L 的数据结构中，根据所述搜索数据串查找子节点位图；当所述子节点位图指示存在子树时，

则有与所述搜索数据串匹配的子节点；当所述子节点位图指示不存在子树时，则没有与所述搜索数据串匹配的子节点。

进一步的，当所述第二条目对应的数据结构类型为第二类型时，所述查找单元可以包括：

第三子单元，用于在所述存储单元存储的第二条目对应的长度为L的数据结构中，根据所述搜索数据串查找子节点位图；当所述子节点位图指示存在子树时，则有与所述搜索数据串匹配的子节点；当所述子节点位图指示不存在子树时，则没有与所述搜索数据串匹配的子节点。

进一步的，当没有与所述搜索数据串匹配的子节点时，所述查找单元可以包括：

第四子单元，用于根据所述存储单元存储的第二条目中指向所述第一条目的指针，读取对应的第一条目，得到所述搜索数据串的最长匹配前缀。

进一步的，当所述第一条目对应的数据结构类型为第一类型时，所述第四子单元可以包括：

第一模块，用于根据所述搜索数据串中对应的索引字段，在所述存储单元中找到所述第一条目的二的M-N次幂个长度为L的数据结构中与所述搜索数据串对应的长度为L的数据结构；

第二模块，用于在所述第一模块找到的对应的长度为L的数据结构中，根据所述搜索数据串查找前缀节点位图；当所述前缀节点位图指示存在前缀分布时，根据所述前缀分布得到所述搜索数据串的最长匹配前缀；当所述前缀节点位图指示不存在前缀分布时，以缺省下一跳索引作为所述搜索数据串的最长匹配前缀。

进一步的，当所述存储单元存储的路由表中对应的当前与搜索数据串匹配的节点的前缀节点中前缀连续时，所述前缀节点位图仅记录连续前缀的起始地址和结束地址，所述第二模块可以包括：

第一子模块，用于根据所述第一模块找到的对应的长度为L的数据结构中的前缀节点位图记录的前缀地址和/或连续前缀的起始地址和结束地址，计算偏移地址；

第二子模块，用于以所述第一子模块计算得到的偏移地址，并且以所述第

一模块找到的对应的长度为 L 的数据结构中的结果数组指针为基址, 得到所述搜索数据串的最长匹配前缀。

进一步的, 当所述第一条目对应的数据结构类型为第二类型时, 所述第四子单元可以包括:

第三模块, 用于在所述存储单元存储的第一条目对应的长度为 L 的数据结构中, 根据所述搜索数据串查找前缀节点位图;

第四模块, 用于当所述第三模块查找到的前缀节点位图指示存在前缀分布时, 根据所述前缀分布得到所述搜索数据串的最长匹配前缀; 当所述前缀节点位图指示不存在前缀分布时, 以缺省下一跳索引作为所述搜索数据串的最长匹配前缀。

进一步的, 所述前缀节点位图用于表示前缀的地址范围, 所述第四模块可以包括:

第三子模块, 用于根据所述前缀节点位图记录的前缀的地址范围, 计算偏移地址;

第四子模块, 用于以所述第三子模块计算得到的偏移地址, 并且以所述第三模块找到的对应的长度为 L 的数据结构中的结果数组指针为基址, 得到所述搜索数据串的最长匹配前缀。

进一步的, 当所述存储单元存储的步长为 M 的查找树中存在单一路径时, 在所述单一路径的第一级节点的第二条目中设置跳转标志位和跳转字段, 所述跳转标志位用于表示存在单一路径, 所述跳转字段用于表示所述单一路径的路径信息; 该装置还可以包括:

跳转单元, 用于当所述存储单元存储的当前与搜索数据串匹配的节点的第二条目中包括跳转标志位时, 控制所述查找单元根据所述跳转字段, 读取所述单一路径的最后一级节点, 以所述单一路径的最后一级节点作为与所述搜索数据串匹配的子节点。

通过以上的实施方式的描述, 本领域的技术人员可以清楚地了解到本发明可借助软件加必需的硬件平台的方式来实现, 当然也可以全部通过硬件来实施。基于这样的理解, 本发明的技术方案对背景技术做出贡献的全部或者部分可以以软件产品的形式体现出来, 该计算机软件产品可以存储在存储介质中,

如 ROM/RAM、磁碟、光盘等，包括若干指令用以使得一台计算机设备（可以是个人计算机，服务器，或者网络设备等等）执行本发明各个实施例或者实施例的某些部分所述的方法。

以上所述仅是本发明的具体实施方式，应当指出，对于本技术领域的普通技术人员来说，在不脱离本发明原理的前提下，还可以作出若干改进和润饰，这些改进和润饰也应视为本发明的保护范围。

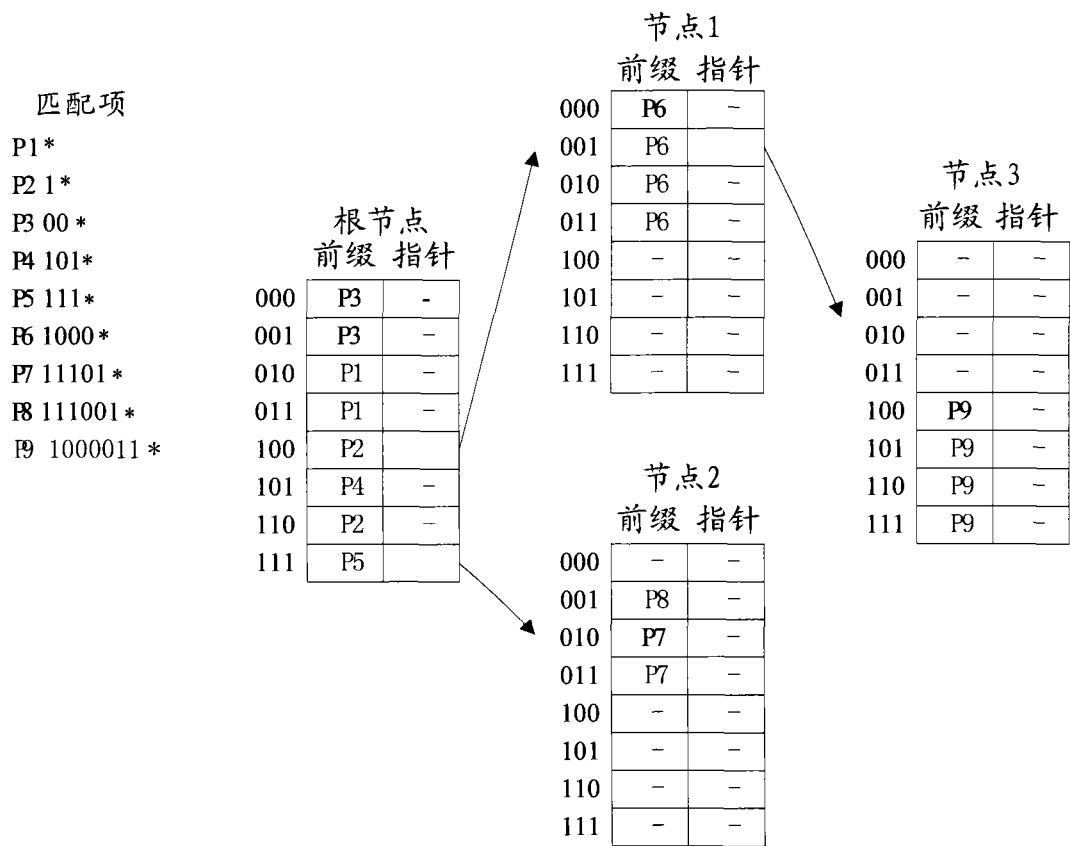


图 1

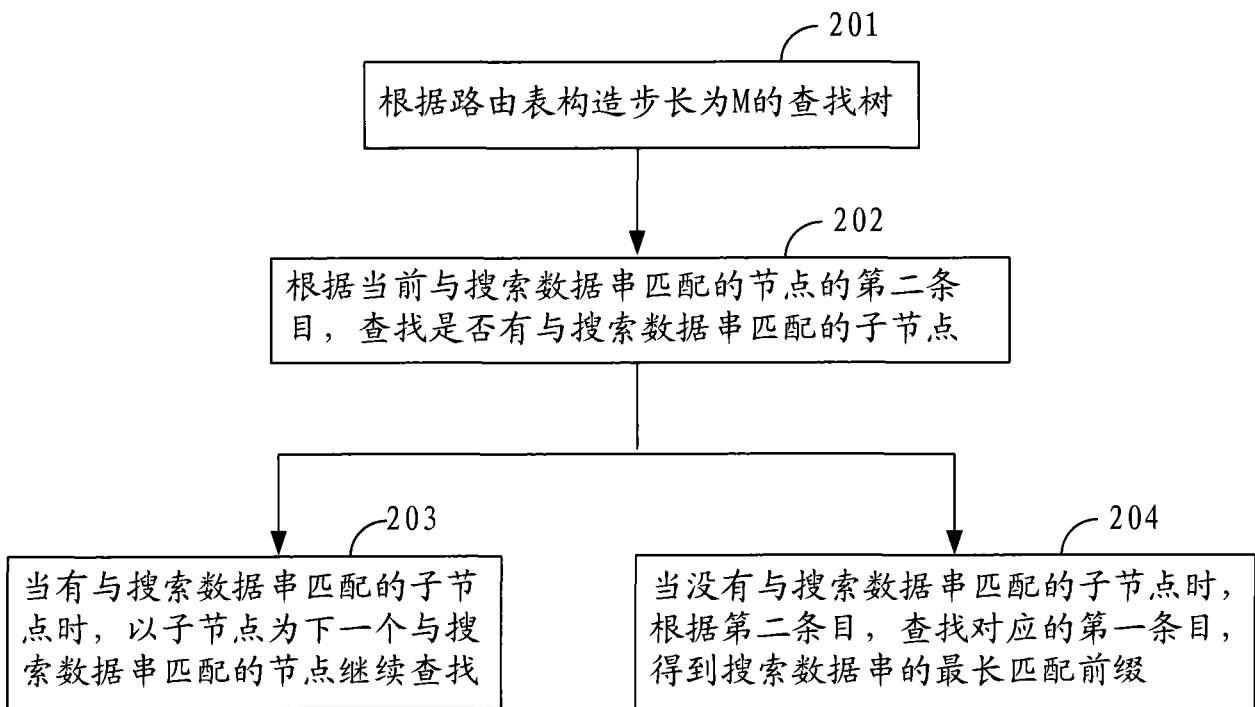


图 2

第一类型	前缀类型	继承最长前缀匹配	子节点数组指针	位置	类型			
第二类型	前缀类型	继承最长前缀匹配	子节点数组指针	位置	类型	位置	类型

图 3

缺省下一跳索引	结果数组指针	前缀节点位图
---------	--------	--------

图 4

子节点位图	子节点类型	下一跳索引	子节点指针
-------	-------	-------	-------

图 5

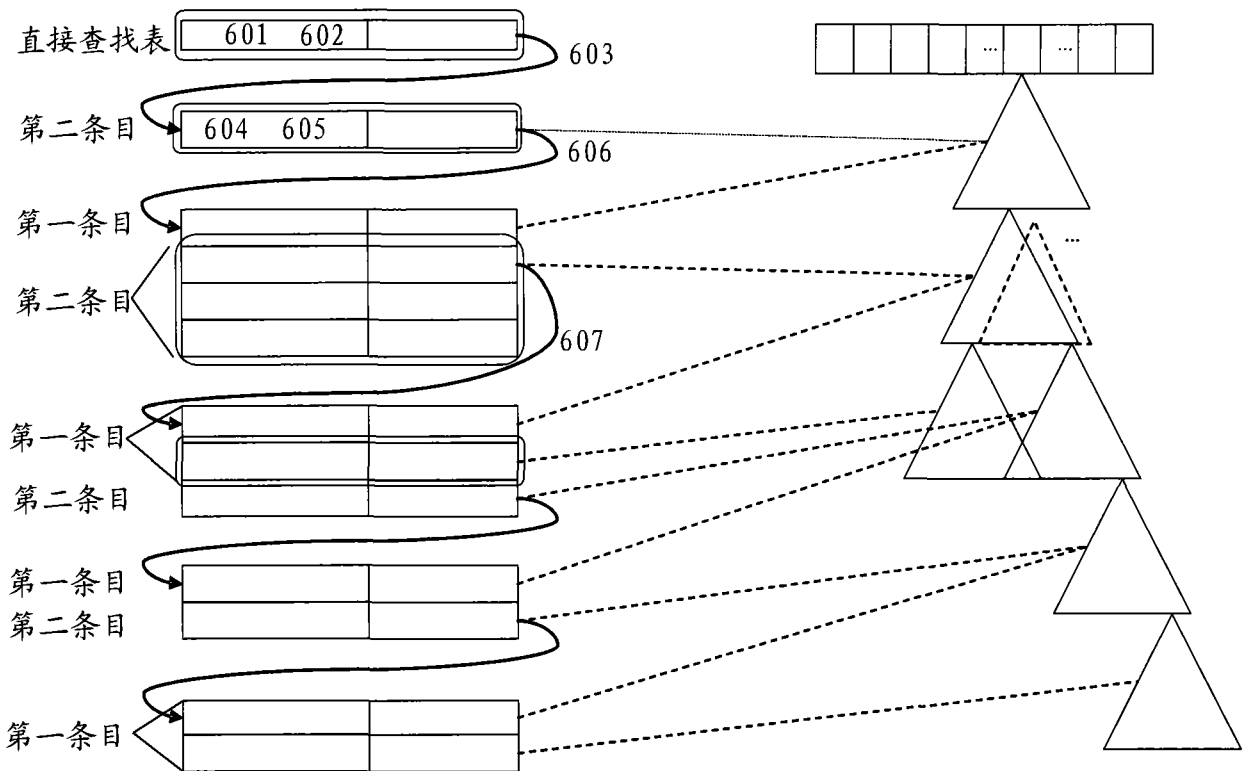


图 6

第一类型	前缀类型	继承最长前缀匹配	子节点数组指针	位置	类型	跳转标志位	跳转字段			
第二类型	前缀类型	继承最长前缀匹配	子节点数组指针	位置	类型	位置	类型	跳转标志位	跳转字段

图 7

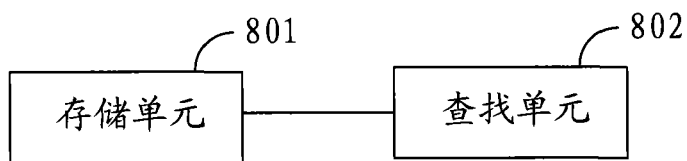


图 8