

一种无回溯的最长前缀匹配搜索算法

张飞飞^{1,2,3}, 李华伟^{1,2}, 韩银和^{1,2}

(1. 中国科学院计算机系统结构重点实验室, 北京 100080; 2. 中国科学院计算技术研究所, 北京 100080;

3. 中国科学院研究生院, 北京 100039)

摘 要: 研究网络处理器中的搜索算法, 提出一种基于 Patricia 树的无回溯搜索算法, 并进行仿真和评估分析。该算法被用于中科院计算所的网络处理器的搜索引擎的设计中, 该搜索引擎可以运行在 155.9 MHz 的 XC2VP30 FPGA 上, 占用 421 个 LUT, 当频率为 100 MHz 时, 每秒可以执行约 7 000 000 次搜索操作, 实现了资源消耗和性能的折中。

关键词: 搜索算法; 最长前缀匹配; Patricia 树; 搜索引擎

Non-backtracking Longest Prefix Match Search Algorithm

ZHANG Fei-fei^{1,2,3}, LI Hua-wei^{1,2}, HAN Yin-he^{1,2}

(1. Key Laboratory of Computer System and Architecture, Chinese Academy of Sciences, Beijing 100080; 2. Institute of Computing Technology,

Chinese Academy of Sciences, Beijing 100080; 3. Graduate University of Chinese Academy of Sciences, Beijing 100039)

【Abstract】 This paper studies search algorithms designed for network processor, and presents a new non-backtracking longest prefix match algorithm based on Patricia tree. The algorithm is simulated and evaluated, which is used in the searching engine design of a network processor designed by Institute of Computing Technology, Chinese Academy of Sciences. The searching engine can run at a speed of 155.9 MHz, and occupies 421 LUTs when implemented in a XC2VP30 FPGA. It can finish 7 000 000 search operations when running at a speed of 100 MHz.

【Key words】 search algorithm; longest prefix match; Patricia tree; searching engine

网络处理器是一种专为网络应用而设计的可编程器件, 将通用处理器的可编程能力和 ASIC 的高性能有机地结合在一起, 在满足不断增长的网络处理要求的同时快速地适应新的服务要求。它的出现为网络设备的设计提供了新的解决方案, 已经成为新一代路由交换设备的核心。

网络处理器利用大量可编程的处理引擎、专门协处理器的并行处理操作, 来提供较高的吞吐率。搜索是网络处理中非常重要的一种操作, 是实现快速分组转发的关键, 在网络处理器设计中通常采用专用搜索引擎来实现, 搜索算法的选择直接影响到网络处理器的分组处理性能。

笔者研究了网络处理器中的搜索算法, 提出了一种用于网络处理器搜索引擎的无回溯最长前缀匹配搜索算法。

1 网络处理中常用搜索算法介绍

评价搜索算法的主要指标包括搜索速度、搜索算法的存储开销、查找表的更新时间、算法的可扩展性以及算法的可实现性等方面。根据数据结构和查找机制的不同, 将现有的搜索算法分为 5 种类型^[1]: (1)基本二叉树及路径压缩二叉树机制, 如 Patricia 树^[2]。(2)多叉树机制, 如 Lulea 算法^[3]。(3)在树结构层间二叉查找而在层内采用哈希查找的算法, 如文献[4]的算法。(4)将每个前缀看作一个地址空间的范围, 依此将整个地址空间划分为不同的段, 在各个段上分别进行一个两路或多路搜索, 如文献[5]的算法。(5)基于 CAM(Content-Addressable Memory)存储器的搜索机制, 如基于 TCAM 的算法。

表 1 对前述各类算法在最坏情况下的存储空间需求、搜索算法的时间复杂度和更新速度进行了总结^[1], 表中“-”表示相应的项是与特定的实现相关的。

表 1 搜索机制最坏情况的复杂度

搜索机制	搜索复杂度	存储空间需求	更新复杂度
基二叉树及路径压缩二叉树	$O(W)$	$O(NW)$	$O(W)$
多叉树	$O(W/K)$	$O(2^K \times NW/K)$	$O(W/K + 2^K K)$
层内二分查找	$O(W/K)$	$O(2^K \times NW/K)$	$O(W/K + 2^K K)$
多路区间搜索	$O(\log N)$	$O(N)$	$O(N)$
TCAM	$O(1)$	$O(N)$	-

在这 5 类算法中, 基于二叉树的搜索方法搜索和更新的复杂度都比较适中, 存储空间消耗较小, 且这种算法具有很好的灵活性和扩展性, 适合网络处理器的搜索引擎实现。

根据前述分析和应用需求, 本文提出了一种基于 Patricia 树的搜索算法, 将其作为网络处理器搜索引擎的算法基础。该算法具有扩展性好、简单灵活的特点, 且搜索和更新复杂度都较小, 所需的额外空间复杂度为 $O(1)$, 搜索结构的存储空间消耗也比较适中, 很适合用于网络处理器中搜索引擎的设计和实现。

2 搜索算法设计

该搜索算法以一种路径压缩的二叉树——Patricia 树^[2]为基础。基于 Patricia 树的最长前缀匹配搜索算法一般采用递归的方法来实现^[6], 搜索不成功时, 需要沿搜索路径回溯以找到正确的结果。递归的方法在搜索和维护对应搜索结构时, 所需要的额外空间很大, 且随叶子节点的增加而增加, 这在硬件实现时是不可承受的。

基金项目: 国家自然科学基金资助项目(60606008)

作者简介: 张飞飞(1982-), 男, 硕士研究生, 主研方向: 网络处理, 网络处理器; 李华伟, 副研究员、博士、博士生导师; 韩银和, 助理研究员、博士

收稿日期: 2007-07-15

E-mail: zhangfeifei@ict.ac.cn

所以,必须设计一种新的非递归的、无回溯的、搜索和维护树结构时所需的额外存储空间固定的算法,以便于在搜索引擎中用硬件实现。本文设计了一种搜索和维护时所需的额外的空间复杂度为 $O(1)$ 的算法。

2.1 基础数据结构定义

该算法采用树结构来存储信息,对树的检索、插入、删除等操作都是根据关键词的值来进行的,树的叶子节点存放了作为索引的关键词值以及对应的用户信息。

2.1.1 模式查找控制块

模式查找控制块(PSCB)是本文算法的一个重要数据结构,构成树的中间节点。

当读入树的根节点非空时,就需要遍历一个或多个 PSCB,找到一个合适的叶子节点。PSCB 代表了树的叶子节点或者 2 个分支的起始位置,在 PSCB 中存储了一个叫下一待测位(NBT)的域,表明了下一步要测试的位的位置。本文的算法只在叶节点中的关键词不同的地方才能插入 PSCB,这使查找算法的性能主要取决于 PSCB 的层次亦即树中叶子的个数,而与关键词的长度无关。

2.1.2 叶节点控制块

叶节点控制块(LCB)是本文搜索算法的另外一个重要数据结构,用于构成树的叶子节点。

LCB 至少包含关键词、前缀长度和用户信息 3 个部分。关键词存放该叶子节点代表的关键词,前缀长度用于指明树中叶子节点的所代表的前缀长度。当查找到一个叶子节点时,搜索算法通过将输入的关键词与该值进行比较,以确定查找是否成功。用户信息的格式由用户定义,其中存放了与该叶子节点相关的信息,如 IP 路由转发表中的下一跳信息等。

2.1.3 搜索树结构

搜索树的每个内部节点包含了两个完全相同的部分,每一部分称作模式查找控制块行(PSCBLine)。每个 PSCBLine 包含下一个 PSCB 地址(NPA)、叶子控制块地址(LCBA)、下一待测试位(NBT)和格式(Format) 4 个域。搜索树的内部节点结构如图 1 所示。

Format0	NPA0	LCBA0	NBT0
Format1	NPA1	LCBA1	NBT1

图 1 搜索树内部节点结构

其中,NPA 和 LCBA 域分别存放该中间节点上连接的模式查找控制块和叶节点控制块的地址;NBT 域则用于指明搜索时下一步要测试关键词的哪一位;以确定下一步的动作,即究竟遍历 2 个 PSCBLine 中的哪一个;格式域(Format)定义了对应的 PSCBLine 的类型,其定义如表 2 所示,其中,格式域的值是用二进制表示的。

表 2 搜索树节点格式域的定义

Format 值	定义	NPA	LCBA	NBT
00	空节点	无效	无效	无效
01	包含指向 PSCB 的指针	NPA	无效	NBT
10	包含指向 LCB 的指针	无效	LCBA	无效
11	包含指向 PSCB 的指针和包含指向 LCB 的指针	NPA	LCBA	NBT

2.2 搜索与维护算法流程

在本文设计的最长前缀匹配算法中,树的中间节点没有父节点指针,因此,在查找过程中,要暂存搜索的结果可以用于比较。对搜索树的搜索就是对它进行遍历的过程,流程

见图 2。

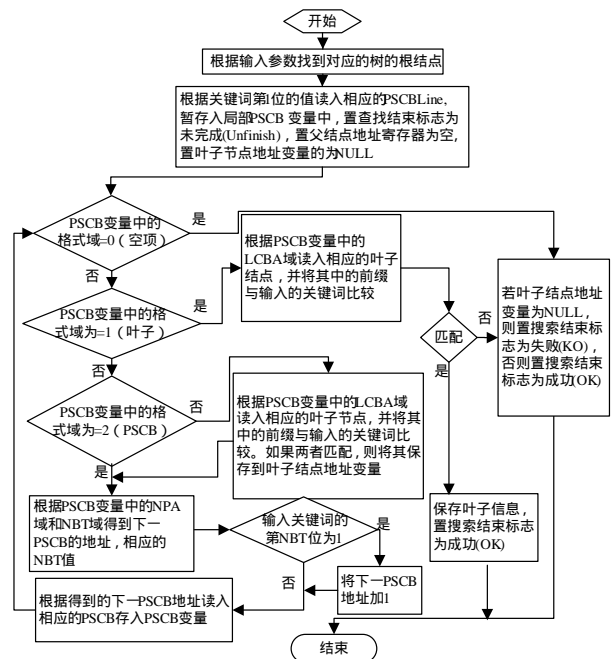


图 2 搜索树搜索流程

对搜索树的维护操作主要包括建立、插入和删除等,建立一棵搜索树可以通过向一个空树中连续插入各个叶节点来实现,因此,只要设计树的插入和删除算法即可。

搜索树插入流程见图 3。

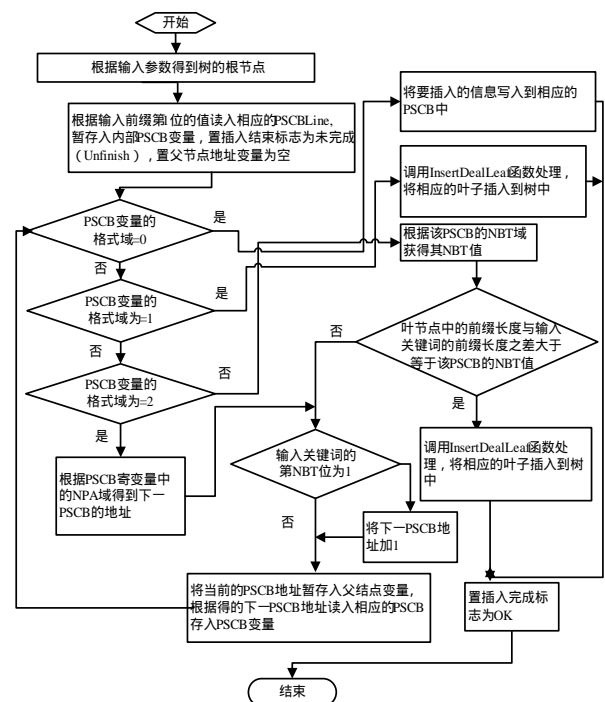


图 3 搜索树插入流程

对于搜索树的插入算法,因为插入的是前缀,所以不可以使用搜索的算法查找插入点,且在插入前缀的过程中还有许多种情况需要处理,这都使插入操作变得复杂。图 3 中插入算法使用的 InsertDealLeaf 函数的处理流程如图 4 所示。搜索树的删除操作的难点在于如何处理删除时空 PSCB 及如何保证将树中所有节点均被删除,本文的搜索算法删除流程如图 5 所示。

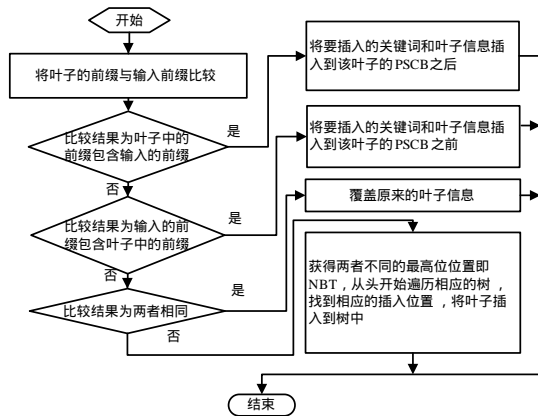


图4 InsertDealLeaf 函数处理流程

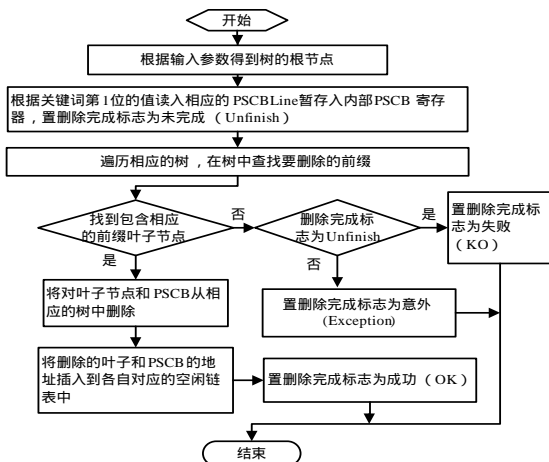


图5 搜索树的删除流程

3 算法验证与评估分析

验证是确保设计正确性的重要环节,验证方法主要包括模拟验证和形式化验证两种。前者通过对设计施加特定或者随机的激励,并通过模拟执行得到响应,从而检验设计的正确性;而后者通常采用数学的手段来证明设计的正确性,主要包括:模型检验,等价检验,定理证明等。其中,形式化验证的处理能力具有局限性,使用相对困难;模拟验证则难于产生完备的测试集,不能确保测试到各种边界条件。目前,模拟验证有了比较成熟的流程、工具以及方法指导,因此,笔者采用模拟仿真的方法来验证该设计。

本文采用C++语言对搜索算法进行建模,使用一个激励生成器控制模拟器进行树结构的建立、添加、删除和搜索等操作,并收集执行中的信息,对算法的正确性和性能进行分析,排除了设计当中的大量错误,保证了设计的正确性。在验证过程中,本文采用测试数据主要包括手工编写的针对各种边界条件的测试数据序列、经过整理的来自亚太互联网信息中心(APNIC)的一个包括1万多条路由记录的路由表和包含数千条记录组成的随机生成MAC表等。经过这些数据的测试,验证了本文设计的搜索维护算法的正确性及其用于网络处理器的搜索引擎设计的可行性。

在搜索算法设计过程中,对设计方案的性能和开销进行仔细分析和评估是非常重要的。通过性能分析和评估可以确定设计所作的各种权衡取舍是否得当,找出系统性能存在的瓶颈,从而加以改进。可以从2个方面评价该实现方案:(1)方案复杂度和灵活性,即算法用硬件实现面积开销、运行

速度和功耗等,这些都极大地影响了最后产品的成本。(2)方案可以提供的处理能力,即搜索算法的搜索和更新速度,数据结构的空间开销及其用于搜索引擎实现时效率等。

本文在搜索算法的C++模型上运行了大量的测试数据,并对运行结果进行了评估和分析。根据运行结果,当表项数目为5000时,查找树的平均深度为15左右。若采用的PSCBLine的宽度为32位,在一个总线宽度为32位的系统上,平均进行一次搜索要进行15次存储器访问,进行一次更新操作所需的存储器访问次数在20次左右。当表项个数增加时,可以采用哈希的方法将所有表项分配到若干棵树中,以提高搜索和更新的速度。

4 算法应用

本文设计的搜索算法已经被应用于中科院计算所的一款用于边缘网络的网络处理器——TinyNP的搜索引擎协处理器的设计中。TinyNP采用并发多处理结构,具有良好的扩展性和灵活性,可根据特定的网络应用需求配置处理引擎的个数和外围接口,主要用于边缘网络的协议转换、转发及远程控制等各种领域的应用。该系统能够支持包括uC/OS嵌入式操作系统、U-Boot Boot Loader以及LWIP协议栈在内的多种基础软件。在Xilinx公司的XC2VP30 FPGA上4处理引擎TinyNP系统稳定运行于80 MHz,满足在4个10/100 Mb/s全双工以太网口的线速分组处理要求。

在网络处理中,对表格的搜索操作远多于维护操作,而对树结构的维护操作又比搜索操作复杂得多。因此,TinyNP的搜索引擎采用硬件搜索、软件维护的策略,将最频繁的搜索操作作用硬件实现,而不常用但却比较复杂的操作则通过运行在处理引擎上的微码程序来完成,以最小的硬件开销实现最大的性能改善。在XC2VP30 FPGA上,该搜索引擎单独运行的最高频率为155.9 MHz,占用421个4输入的LUT(该FPGA的总LUT数的1%)。当运行在100 MHz的系统时,该搜索引擎平均每秒可以进行约7000000次搜索操作,具有很好的性能和较小的资源消耗。

5 结束语

本文设计的搜索算法可扩展性良好、简单灵活。搜索和更新复杂度都较小,搜索和更新所需的额外空间消耗的复杂度为 $O(1)$,搜索结构的空间消耗也适中,可用于网络处理器的搜索引擎的硬件实现。该算法已经被应用于一款专用于边缘网络的网络处理器搜索引擎的设计中,以较少的资源消耗取得较好的处理性能。

参考文献

- [1] Jonathan H C. Next Generation Routers[J]. Proceedings of the IEEE, 2002, 90(9): 1518-1558.
- [2] Morrison D R. PATRICIA-practical Algorithm to Retrieve Information Coded in Alpha-Numeric[J]. Journal of the ACM, 1968, 15(4): 514-534.
- [3] Degermark M. Small Forwarding Tables for Fast Routing Lookups[C]//Proc. of ACM SIGCOMM'97. [S. l.]: ACM Press, 1997.
- [4] Waldvogel M. Scalable High-speed IP Routing Lookups[C]//Proc. of ACM SIGCOMM'97. [S. l.]: ACM Press. 1997-09: 25-36.
- [5] Lampson B. IP Lookups Using Multiway and Multicolumn Search[C]//Proc. IEEE INFOCOM'98. [S. l.]: IEEE Press, 1998.
- [6] Lorion S. Patricia Trie Implementation[EB/OL]. (2003-05-02). <http://www.codeproject.com/csharp/patriciatrie.asp>.