维、哈、柯全文搜索引擎索引器的设计与实现*

Design and Implementation of the Uyghur, Kazak, Kyrgyz Full-text Search Engine Indexer

吐尔地·托合提　维尼拉·木沙江　艾斯卡尔·艾木都拉

(新疆大学信息科学与工程学院　乌鲁木齐　830046)

摘　要　介绍了一种基于倒排索引机制的索引器设计方案、技术及实现算法,它是维、哈、柯多语种全文搜索引擎系统中的查询模块,该模块用来对特定的用户查询进行高效的检索。针对维、哈、柯文的特点、网络信息量以及本系统所拥有的硬件资源,查询效率极高的 Hash 链表作为数据结构,在内存建立倒排索引表,整体建立和更新索引并支持更新时查询而不会影响查询效率。

关键词　多语种搜索引擎　倒排索引　Hash 链表　索引器　少数民族语言

中图分类号　TP391.3

随着 Internet 和 WWW 的迅速发展,Internet 上的资源越来越丰富,基于 Internet 的各类信息检索服务随之诞生并获得了迅速的发展。在中国,出现了非常优秀的中文、英文搜索引擎,如 Google、Baidu 等,但是这些搜索引擎没有解决少数民族语言文字特征方面的关键问题(民文在线输入及显示,标准字符编码,网页布局及书写方向,检索词预处理等),完全不能满足广大少数民族网络用户的信息检索需求。到目前为止,针对维吾尔文、哈萨克文、柯尔克孜文(以下简称维、哈、柯文)等少数民族语言的搜索引擎的研究还处在空白阶段,还没有一个比较成熟的搜索引擎。为了能够在网上快速检索以本民族语言发布的信息而开发一个多文种搜索引擎是新疆少数民族面临的一个急待解决的重要问题。以下是我们设计的维、哈、柯多文种全文搜索引擎的系统结构[1]。由数据采集器(Crawler)、文档分析器(Analyzer)、索引器(Indexer)、检索服务器(Retrieval Server)4 个服务模块组成,如图 1 所示。

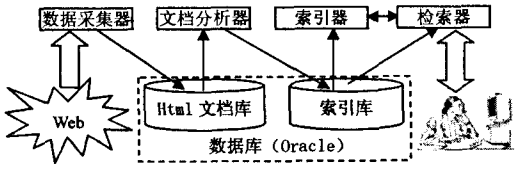

图 1　维、哈、柯多文种全文搜索引擎系统架构

本搜索引擎中,数据采集器(Crawler)的功能是下载维,哈,柯文网页并存储到 html 文档库中。文档分析器(Analyzer)的功能是从 html 文档库中读取已下载的页面数据并提取页面正文,将正文字符转换成标准维、哈、柯文 Unicode 字符,再经过分词(关键词)、去除停用词和常用词、词根切分等处理之后生成索引关键词序列并存储到索引库中[2]。索引器(Indexer)的功能是负责根据索引库的内容和 Page rank 值在内存中建立到排索引表,并根据查询关键词给检索器(Retrieval server)返回关键词有关索引信息。检索器(Retrieval Server)的功能是负责接受用户查询,对查询关键词进行预处理和进行数据库检索,并将结果展现给用户。

在搜索引擎中,索引器的设计直接决定着搜索引擎的查询速度,因此设计和实现过程中应该把快速性作为前提确定数据结构和实现算法。本搜索引擎的设计和实现过程中,针对维、哈、柯文的特点,网络信息量以及本系统所拥有的硬件资源,将 Hash 链表作为数据结构,使用倒排索引技术[3~5]及 Java 语言[6]实现了结构简单、快速而高效的索引器。

1　索引器总体结构设计

到目前为止,维、哈、柯文网页的总数目大约在 5 万个左右,相对于其他大语种网页来说,数目非常小。充分考虑维、哈、柯文网页所具有的实际情况以及构建维、哈、柯多语种搜索引擎所拥有的硬件资源,更重要的是尽量缩短索引表的建立与更新时间、提高查询效率,在设计维、哈、柯多语种搜索引擎时,将索引器跟索引库(Oracle 数据库)运行在同一台具有 8GB 内存的 Sun Sparc 服务器上,将索引全部建立在服务器的内存中。本索引器采用的是目前应用最广泛的倒排索引技术并选择了查询效率极高的数据结构——Hash 链表。虽然维护倒排索引比较困难,但是查询中倒排索引表现出来极强的优越性,由于一次查询可以查找出所有包含该单词的文档信息,所以缩短查询响应时间。本模块的主要功能是依照文档分析器(Analyzer)得到的数据(索引关键词序列),以及网页 Page Rank 值来建立和更新倒排索引,同时快速响应检索器(Retrieval server)的查询请求。维、哈、柯多语种搜索引擎索引器模块结构及工作流程如图 2 所示。

基金项目:新疆维吾尔自治区高技术研究与发展计划项目"维、哈、柯、汉多文程多项搜索引擎关键技术研究与开发"(项目编号:200612115);新疆维吾尔自治区高校科研计划重点资助项目"维、哈、柯、汉多文种信息多项搜索引擎开发"(项目编号:XJEDU2006113)。

作者简介:吐尔地·托合提,男,1975 年生,硕士,讲师,研究方向为自然语言处理及信息检索;维尼拉·木沙江,教授,硕士生导师;艾斯卡尔·艾木都拉,教授,博士生导师。

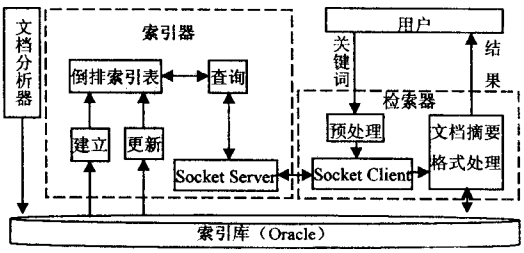

图 2 维、哈、柯多文种全文搜索引擎索引器模块架构

2 倒排索引表数据结构

索引器应该尽可能快地响应用户的查询请求，所以应该选择一种查询效率极高的数据结构。根据维、哈、柯文网页的数量以及硬件资源的情况，本搜索引擎选择了 Hash 链表作为建立倒排索引表的数据结构。Hash 链表的节点中存储每个关键词的倒排索引信息，包括关键词、包含此关键词的网页编号、该网页中的起始位置，出现次数，长度等信息。Hash 链表中某条链的所有节点的 key 值按照由大到小的顺序进行有序存放，这种设计可以将平均查找时间降低一半。倒排索引表的组织形式如图 3 所示。

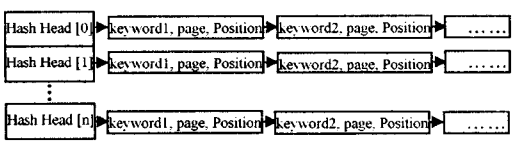

图 3 倒排索引表的组织形式

对于 Hash 链表而言，一个合适的 Hash 函数可以尽可能地减少聚集的发生。在本搜索引擎的具体实现时，采用数组来存储 Hash 链表的表头，对于每一个关键词，具体的 Hash 函数实现如下：

```
for(int i=0;i<keyWord.length();i++)
{key += ((int)keyWord.charAt(i) * Math.pow(2,i)) % Integer.MAX_VALUE;}
number = key % arraySize
```

首先利用循环计算出每个关键词的 key 值，计算 key 值的具体方法为通过循环依次进行如下操作：得到关键词字符串中相对应位置上的字符并将其转换为整型数据，再将转换后的整型数据同 2 的 i 次幂进行乘积运算，i 为此字符在字符串中所处的位置减 1，最后用乘积得到的数据对整型中的最大值进行取模运算并与之前的 key 值进行相加。用这种方式得到的 key 值可以尽量避免与其他 key 值的冲突，从而尽可能减少 Hash 链表中聚集的发生，再用此 key 值对数组长度进行取模运算，这时得到的 number 值就是相对应的数组下标，就可以将这个索引项连接到数组下标为 number 的 Hash 链中。

3 倒排索引表的建立与更新

本索引器使用到了三张数据表来建立和更新倒排索引表，分别为 FLAG 表、STABLE 表和 STABLE1 表。其中 FLAG 表的作用是指示最近更新过的网页数据存储在索引数据表 STABLE 还是 STABLE1 中。STABLE 表与 STABLE1 表具有完全相同的结构，用这两张表来存储经过 Analyzer 处理后的网页的数据。在这两张表中，包括以下属性：URLID，用来存储网页编号；URL，用来存储该网页的 URL 地址；KEYWORDS，用来存储经过处理的网页中的关键词；POSITION，用来存储每个关键词在该网页中的位置信息；PAGERANK，用来存储该网页的 PageRank 值；CONTENT，用来存储网页正文；TITLE，用来存储该网页的标题；UPDATETIME，用来存储该网页的更新时间。利用 STABLE 与 STABLE1 交替存储新的网页数据，其目的是为了保证在倒排索引表进行更新时处理用户的查询请求，搜索引擎程序仍然可以进行正常的工作。

3.1 **倒排索引表的建立** 倒排索引表的建立主要思想就是将 STABLE 或 STABLE1 中的以 URLID 为主键的正排表转换为以 keyWord 为主键的倒排表。为了将正排表转换为倒排表，需要建立三个数组，keyword、page 和 position。用它们分别来存放数据表中出现的所有关键词、每个关键词所对应的网页编号的集合以及每个关键词在其出现的每个网页中的位置信息集合。建立倒排索引表的具体算法如下：a. 从 FLAG 表中的 flag 字段得到数据表的标志信息，即指明最近更新的网页数据存储在哪张表中。b. 从 flag 字段所指示的数据表中读取数据，包括网页的编号 URLID，网页中包含的关键词 keywords，网页中关键词在该网页中的位置信息 position 并按照数据表中的 PageRank 值对查询结果进行降序排列。c. 将读取得到的网页信息按照关键词进行拆分，将关键词、网页编号、关键词的位置信息按照相同的顺序分别存储到 keyword、page 和 position 数组中。在存储时有两种情况，如果该关键词是第一次出现，直接将它添加到 keyword 的末尾存储单元，将它出现的网页编号信息及网页中的位置信息分别添加到 page 和 position 的末尾存储单元；如果该关键词在此前已经出现过，则在 keyword 中找到存储该关键词的位置，再从 page 和 position 中从刚才得到的位置处取出相对应的网页编号信息的字符串及网页中的位置信息的字符串，将新的网页编号信息和网页位置信息追加到相应字符串的末尾，再将更新后的字符串放回到 page 和 position 中相对应的位置。d. 直到最后一个网页中的所有关键词都处理完毕之后，就可以用 keyword、page 和 position 中相对应位置上的字符串和 flag 字段的值建立新的节点，并插入到建立好的 Hash 链表中，直到 keyword 中的最后一个关键词都插入进 Hash 链表后，倒排索引建立完毕。倒排索引表的建立流程如图 4 所示。

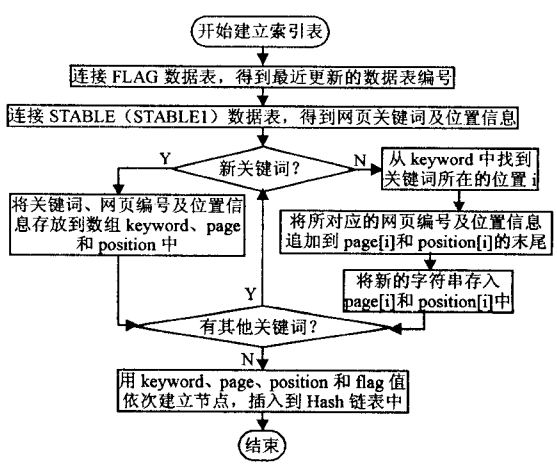

图 4 倒排索引表的建立流程

3.2 **倒排索引表的更新** 由于维、哈、柯文网页数量相对来说比较少，而且维、哈、柯文网站的更新速度相对也比较缓慢，所以反映到搜索引擎来说，第一，对所有网页做一次完整的更新时间不会太长；第二，索引表更新的频率不需要太快。因此，在对索引进行更新时，我们选择了整体更新方法，即在一次更新中读取所有的网页同时进行更新。具体实现方法如下：a. 按照建立索引的方法，得到 keyword、page 和 position 的值。b. 将负责处理用户查询的类中的更新标志设置为 true，即通知该类更新开始，需要对用户的查询进行特殊处理。c. 用 keyword、page、position 和 flag 建立新节点，接着在原 Hash 链表中查找是否存在与即将插入到 Hash 链表中节点具有相同关键词的节点。如果有，则先将此节点从 Hash 链表中删除，之后再将新的节点插入；如果没有，直接将新的节点插入到 Hash 链表中。d. 待全部节点插入到 Hash 链表中后，对 Hash 链表进行一次遍历，取出每个节点中的 tableFlag 变量的值，同 flag 字段的值进行比较，若相等，则不进行任何处理；若不相等，则将该节点从 Hash 链表中删除。这样做的目的就是删除在原来的网页中出现过的而在新一次搜集的网页中没有出现的关键词。区分这类节点的标志就是 Link 对象中保存的 tableFlag 字段，由于 Analyzer 模块是交替使用 STABLE 表和 STABLE1 表，那么假如上一次 FLAG 表中的 flag 字段的值为 0 的话，同时根据此 flag 值建立的索引表节点中的 tableFlag 的值也为 0，那么这一次 flag 字段的值一定是 1，同时节点中的 tableFlag 的值也为 1。由此，可以判断出那些存在于 Hash 链表中的 tableFlag 的值为 0 的节点，一定是已过期的索引对象，需要删除。e. 将负责处理用户查询的类中的更新标志设置为 false，通知该类更新结束。索引更新的过程如图 5 所示：

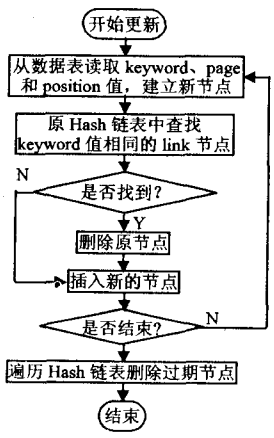

图 5 倒排索引表的更新流程

4 倒排索引表的查询

查询模块的主要功能就是与检索器进行交互，对检索器提交的查询请求尽快进行处理并返回查询结果。查询具体实现方法如下：a. Indexer 启动 Socket Server，等待检索器的查询请求。b. 当检索器有查询请求到来时，Indexer 查询模块建立一个线程并将更新标志设计为 true，表示更新时查询（动态查询）或 false，表示非更新时查询（静态查询），用该线程去处理检索器的查询请求并继续等待检索器的其他查询请求。c. 查询模块接收检索器提交的用户查询请求后检查更新标志，如为 false（静态查询），直接查找 Hash 链表并返回查询结果并立即断开与检索器之间的连接。如为 true（动态查询），首先检查关键词的个数，若关键词个数为 1，则直接查找 Hash 链表；若关键词个数大于 1，则首先查看每个关键词所在节点中的 tableFlag 值，如果所有查询关键词的 tableFlag 值都一样，则说明用户查询的关键词或者全部更新过，或者全部没有更新过，对于这种情况，直接查询就可以了。如果 tableFlag 值不同，说明用户输入的多个查询关键词中有一些已经更新过，而有一些未更新过，这时就将未更新过的关键词舍弃，只对已更新过的关键词进行查询。

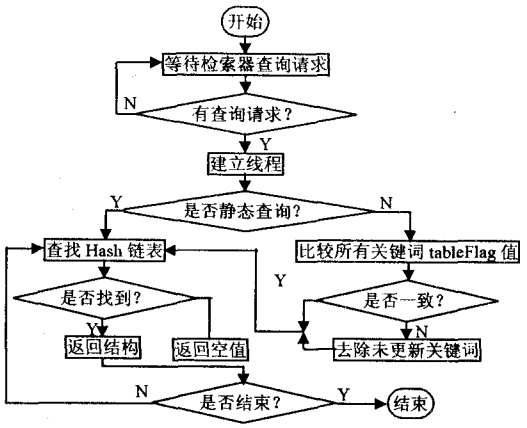

图 6 倒排索引表的查询流程

5 结束语

针对维、哈、柯文网络信息量以及本系统所拥有的硬件资源，将 Hash 链表作为数据结构，使用倒排索引技术及 Java 语言实现了结构简单，快速而高效的索引器，明显提高了维、哈、柯全文搜索引擎德查询效率。该搜索引擎已经以 www.tapkak.com 为域名开始提供维、哈、柯文信息检索服务，得到了广大维吾尔族、哈萨克族、柯尔克孜族网络用户的认可，成为了新疆地区唯一的少数民族语种的全文搜索引擎。

参 考 文 献

1 李晓明，闫宏飞，王继民. 搜索引擎——原理、技术与系统 [M]. 科学出版社，2005，4(1)：20－27

2 徐宝文，张卫丰. 搜索引擎与信息获取技术 [M]. 清华大学出版社，2003，4(1)：120－121

3 F Scholer，H E Williams，J Yiannis，et al. Compression of Inverted Indexes for Fast Query Evaluation [J]. 25th ACM－SIGIR International Conference on Research and Development in Information Retrieval. Finland，2002：222－229

4 A Tomasic，H Garcia－Molina，K A Shoens. Incremental Updates of Inverted Lists for Text Document Retrieval[J]. Proceedings of 1994 ACM SIGMOD International Conference on Management of Data. Minneapolis，1994，23(2)：289－300

5 贾 崇，陆玉昌，鲁明羽. 一种支持高效检索的即时更新倒排索引方法[J]. 计算机工程程与应用.，2003，29：198－201

6 Robert. Lafore. Data Structures & Algorithms in Java[M]. 中国电力出版社，2004，2(1)：427－428

（责编：白燕琼）