

# 线性散列在全文检索中的应用研究

束文杰, 时亚南, 于国欣

(新疆维吾尔自治区特种设备检验研究院, 新疆 乌鲁木齐 830011)

**摘要:**散列表是一种常见的数据结构,理论上它能以常数级时间复杂度 $O(1)$ 执行查询操作,因而在计算机技术中具有广泛的应用。在大规模用户并发向全文检索系统请求数据的情况下,系统会出现响应速度慢以及检索效率低等问题。为解决上述问题,引入了动态散列技术—线性散列,结合全文检索系统的实际需要,提出了一种分块式线性散列倒排索引的构建方法,并详细阐述了该线性散列索引的索引结构、存储方式、设计思路 and 实现细节。经大量实验测试,基于线性散列的倒排索引具有极快的响应速度,明显提高了全文检索的查询性能。

**关键词:**散列表;全文检索系统;线性散列;倒排索引

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2015)06-0197-05

doi:10.3969/j.issn.1673-629X.2015.06.044

## Research on Application of Linear Hash in Full-text Retrieval

SHU Wen-jie, SHI Ya-nan, YU Guo-xin

(Xinjiang Uygur Autonomous Region Inspection Institute of Special Equipment,  
Urumqi 830011, China)

**Abstract:** Hash table is a common data structure, and theoretically it can execute the query operation in a constant level time complexity  $O(1)$ , so it has a wide application in the computer technology. Under the circumstances that large-scale concurrent users try to request data from the full-text retrieval system, the system will be slow to respond and retrieve in low efficiency. In order to solve these problems, introduce a dynamic hashing technique—linear hash. Combined with the full-text retrieval system's actual needs, propose a method of block inverted index built on linear hash, and elaborate the linear hash index's index structure, storage pattern, design ideas and implementation details. After a large number of experimental tests, the inverted index based on linear hash has an extremely fast response speed, and significantly improves the full-text retrieval's query performance.

**Key words:** hash table; full-text retrieval system; linear hash; inverted index

## 0 引言

互联网的出现及迅速普及,深刻地影响和改变了人们的生活方式,并从根本上改变了人们获取信息的方式,网站成为人们获取日常信息的主要来源之一。但同时也带来了一个极具挑战的现实问题,那就是面对如此庞大的信息量,人们如何快速有效地从这些海量信息中获取自己想要的信息。搜索引擎便是在这种应用背景下被催生出来的一大新兴技术。从广义上讲,搜索引擎实际上就是全文检索引擎,其实质是全文索引,它对网页、文本、电子文档、视频和图像等非结构化信息提供强大的管理功能,从而能有效地解决上述问题,让人们能真正利用好互联网的这种海量信息特性。因此,对全文检索技术进行深入研究,对快捷、高

效、方便地检索网站信息具有十分重要的理论研究价值和实际价值。

全文检索系统主要是通过对非结构化数据建立全文索引的方式实现为最终用户提供对非结构化数据的透明检索和访问的功能。全文检索系统通常是将抓取的互联网信息存储在本地服务器,这些信息包括网页本身的信息、关键词库以及索引等内容。这样做的好处是可以避免边抓取边检索制约系统检索性能,提高系统响应速度,并提升用户体验。因此,如何存储和检索这些抓取的信息成为研究全文检索系统的关键所在。

目前国内对全文检索的研究方面达到一个高潮。在索引模型选取方面,复旦大学胡运发等对索引模型

收稿日期:2014-08-07

修回日期:2014-11-10

网络出版时间:2015-05-06

基金项目:新疆维吾尔自治区科技攻关项目(200931103)

作者简介:束文杰(1970-),女,主要从事计算机应用方面的研究。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20150506.1648.045.html>

选择的评价标准进行了总结,提出了时间复杂度、空间复杂度、查询完备性、适应性和动态性 5 个评价标准,并按照此标准对倒排表模型(Inverted Index List)、署名文件(Signature Files)、位图(BitMap)、Pat 树、互关联后继树等索引模型进行了对比研究,认为最广泛应用的倒排表模型膨胀比小,创建和检索速度都比较快<sup>[1-2]</sup>。在索引结构方面,邓攀,刘功申<sup>[3]</sup>提出了一种高效的倒排索引结构,即大倒排表应该尽可能存储在连续的一块磁盘上,以减少寻址的次数,而小倒排表应当共享一块连续的磁盘空间,尽可能减少磁盘空间的浪费。王冬等<sup>[4]</sup>提出了将倒排索引每个词项的记录表以链接块方式存储在倒排文件中的索引设计方法。文献[5-7]则重点讨论了索引器的设计结构及性能评估。文献[8-11]着重阐述了全文检索系统的设计及应用,而文献[12-14]则认为散列是一种高效的查询算法,并对几种动态散列技术进行了对比研究。

文中在上述研究的基础上,选择纯文件的存储模式以及基于倒排表模型的倒排索引结构分别作为全文检索系统的物理结构和逻辑结构,结合线性散列技术构建了倒排索引的二级线性散列索引,取得了良好的应用效果。

1 索引存储结构选择

存储结构是全文检索系统中的最关键部分之一,

一个好的存储结构是确保全文检索系统正常、高效、稳定工作的核心。目前索引主要有三种存储方式:纯数据库存储模式、纯文件存储模式、两种混合存储模式。

纯数据库存储指的是把抓取过来的网页信息、所有单词、倒排索引等内容全部存入数据库中,由数据库统一管理。该存储结构需要两张表,一张单词表和一张 url 表。单词表存储词本身 word、词 id 编号 word\_id、该词的倒排索引 urls、包含该词的 URL 的数目 url-count、所有 URLs 中该词出现的总次数 totalcount 等信息,而 url 表则存储 url 的 id 编号 url\_id、站点的 id 编号 site\_id、该 URL 本身 url、该网页的标题 title、该网页的长度 length、该网页的内容 content 等信息。

这种存储结构的优点是将所有的存储都交给数据库,所以程序会相对简单,代码量较少。缺点就是当网页数量较大时,检索速度会下降。

纯文件存储是指将抓取回来的网页信息、所有单词、倒排索引等内容全部存入文件中,由程序的存储模块统一管理。这种方式的实现流程见图 1。

这种模式的优点是将所有的存储都交给程序的存储模块,所以检索速度会较快。缺点是程序相对复杂,实现代码量大。

两种方式混合存储是指将抓取回来的网页信息、所有单词等内容存入数据库中,倒排索引存入文件中。该方式的实现流程见图 2。

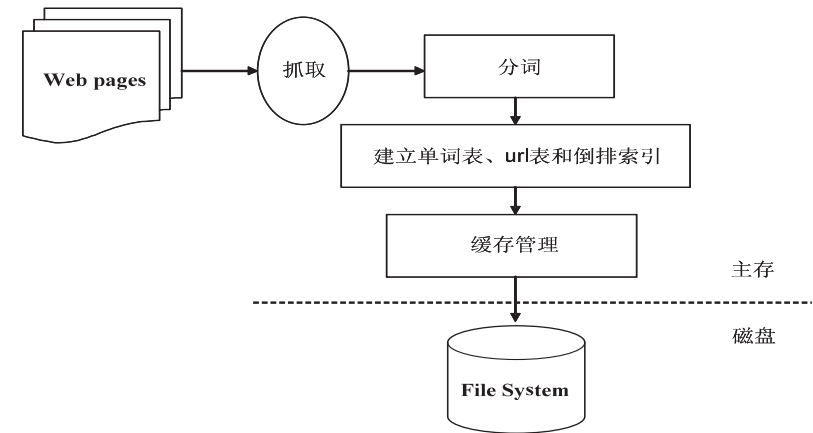


图 1 纯文件存储实现流程图

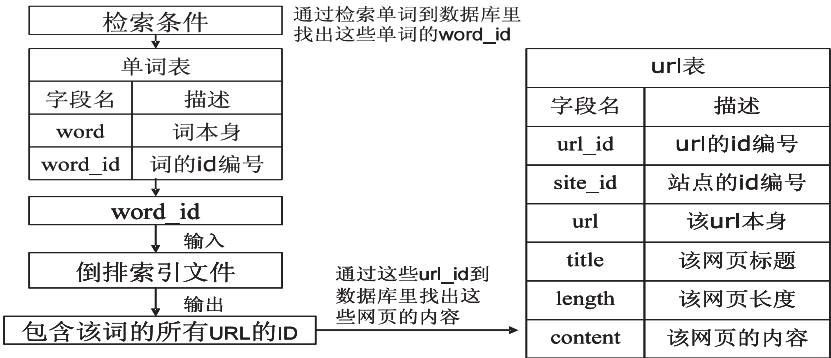


图 2 两种方式混合存储实现流程图

这种存储方式的优点是检索速度保持在较快的水平,程序的代码也相对较少。缺点是需要兼顾倒排文件和数据库,比较繁琐。

文中综合以上三种存储模式的优缺点,结合文中研究的目标,最后确定选用纯文件存储的方式作为文中索引的存储结构。

## 2 文档集来源

文中在 Fedora 14 Linux 上搭建好农业垂直搜索引擎 AgriRoom,仅使用该搜索引擎提供的网页抓取功能,对全国百强农业网站进行定向抓取,并将抓取下来的网页源码作为后续实现索引模块和检索模块的基础。该搜索引擎将抓取的网页源码压缩后存储在 MySQL 数据库中,因此,在网页抓取过程结束以后,还需要将其从 MySQL 数据库中解压出来,分别读取到每个单独的文本文档中,并按网页编号 url\_id 命名文本文档,这样每个文本文档中其实存储的是从互联网上抓取的网页的源码。文中实际抓取了 230 多万张网页,从中随机抽取 100 万张网页作为样本集进行测试研究。

## 3 基于线性散列的倒排索引构建

### 3.1 线性散列技术介绍

线性散列与可扩展散列都是动态散列技术。线性散列常用于索引文件结构组织,尤其是数据库的主文件组织,相对于可扩展散列来说,它的好处就是不需要维护一个单独的目录。可扩展散列需要一个目录来管理,其目录的增长或缩小取决于数据分布,它没有溢出桶,而线性散列没有目录,分裂桶按线性顺序排列,它有溢出桶。特别的,当要插入的数据分布为倾斜直线及两侧分布的情况时,溢出链会导致它的性能很差,实际上在这种情况下,它的性能比可扩展散列还要差,但是这种极端情况只在极个别情况下发生。

用作主存数据结构的散列表中有一个散列函数,它以查找键(也可以称为散列键)为参数并计算出一个介于 0 到  $B-1$  的整数,其中  $B$  是磁盘块(或称为页、桶)的数目,块数组,即一个序号从 0 到  $B-1$  的数组中包含  $B$  个链表的头,每一个对应于数组中的一个磁盘块。如果记录的查找键为 Key,那么通过将该记录链接到块号为  $h(\text{Key})$  的块列表中来存储它,其中  $h$  是散列函数。

而在基于外存的线性散列中,散列表本身存储在不同 pageNo 的磁盘块中,并可根据实际需要实时动态新增新的磁盘块,而桶链(即包含溢出桶)中实际存储真正的记录 RID。无论是散列表 HashTable,还是桶链 Bucket Chains,均由堆文件管理器将其逻辑地组织在

索引文件中。

### 3.2 线性散列索引的实现

下面依次分别给出基于外存的线性散列的插入算法、查询算法和删除算法。

#### 3.2.1 线性散列索引的插入算法

该算法的输入为词条编号 term\_id 和数据记录的标识 rid,算法的输出为布尔值(插入成功为 true,否则为 false)。该算法背后的思想是当一个查找键为 Key 的新记录被插入时,计算  $h(\text{Key})$ ,如果页号为  $h(\text{Key})$  的页还有空间,就把该记录存放到此页的存储块中,或在其存储块没有空间时存储到块链上的某个溢出块中,如果页中的所有存储块都没有空间,就增加一个新的溢出块到该页的链上,并把新记录存入该块。具体算法描述如下:

①调用查找函数 search(term\_id)检查要插入的 term\_id 是否在散列表中已存在,如果已存在的话,直接返回 false;如果不存在,执行步骤②至⑧;

②通过散列函数 hash(term\_id, level)计算出 term\_id 散列到散列表中的地址 index,如果计算出的 index 值小于准备分裂的桶号 next(即分裂点),则 level+1 进行再散列 rehash,重新计算该词条编号在散列表中的地址。这里 level 表示当前散列函数的级别(即分裂轮数);

③根据第②步计算出的散列表中的位置 index,得到该 term\_id 所处的页;

④如果该页未滿,直接将记录 term\_id 和 rid 插入到该页中;

⑤如果该页已滿,则需要进行分裂,执行步骤⑥至⑧;

⑥分配一个新页,并解订该新页,按 level+1 对分裂点 next 所指向页中的记录(如果有溢出页,溢出页中的记录也要处理)进行重新散列,并将所有散列出的值 newIndex 不等于 next 的记录插入到新页中,并从原有页中将这些记录删除,而散列出的值 newIndex 等于 next 的记录仍然保留在原有页中;

⑦如果新页中有记录,不为空,则将其加到散列表并移动分裂点 next,将 next 值加 1;

⑧如果分裂点 next 的值大于  $2^{\text{level}} * N - 1$  ( $N$  表示初始轮数的级别),则将分裂轮数 level 加 1,分裂点 next 值置为 0(表示分裂点回到初始位置,分裂点移动了一轮)。

#### 3.2.2 线性散列索引的查询算法

该算法的输入为词条编号 term\_id,算法的输出为数据记录的标识 rid,具体算法描述如下:

①首先根据词条编号 term\_id 及所处级别 level 计算出该词条编号在散列表中的地址,如果计算出地址



的值小于准备分裂的页号 next(即分裂点),则说明该桶已经分裂,需要将 level 加 1,然后和词条编号 term\_id 重新计算该词条编号在散列表中的地址;

②根据计算出的位置从散列表中获取该位置所指向的页链的首页 firstPage;

③根据该词条编号 term\_id 及首页 firstPage,在该页链中找到需要返回的记录。查找的具体方法是,先从当前页中找,如果当前页没有,再从溢出页中找,最后返回要查找的记录标识 rid。

### 3.2.3 线性散列索引的删除算法

删除算法的输入值为记录 record(LHRecord 类型,包含 term\_id 和 rid),输出值为布尔值 true 或 false,具体算法如下:

①首先根据记录 record,得到词条编号 termID,然后调用查找函数,看看要删除的记录是否存在,如果存在,执行步骤②,否则,返回 false;

②根据词条编号 termID 计算出该词条在散列表中的地址 index,然后根据 index 值得到散列表中该 index 所指向的页链的首页 firstPage,最后根据该记录 record 和首页 firstPage 将该记录从此页链中移除,并计算此时散列表的大小,如果该大小等于默认的分裂轮数,则返回 true,否则,执行步骤③;

③为提高空间利用率及查询效率,根据该页得到当前页下一页的页号,循环读页,对页进行解订和转换,移除没有记录的空页;

④记录下散列表中最后一个页的下标 lastIndex,计算出该 lastIndex 所指向的页链的首页,如果最后一个首页为空,并且其溢出页也全为空,则将该页链从散列表中移除。如果此时其分裂点 next 为 0,则将散列表的大小变为当前散列表大小的一半减 1,并将分裂的轮数减 1,否则只将分裂的轮数减 1。

## 4 系统检索模型构建

针对用户的查询,一篇文档要么满足查询的要求,要么不满足查询的要求。在文档集规模很大的情况下,满足查询的结果文档数可能会非常多,往往会大大超过用户能够浏览的文档的数目。因此,针对全文检索系统来说,构建检索模型至关重要。文中采用对文档进行评分计算与排序的方式对检索模型进行构建。因此,对于给定的查询,全文检索系统会计算每个匹配文档的得分。

另外,系统将源文档中出现的所有基本元素信息(比如次数和位置等)记录到索引库中,即保存至自己设计的文件库中,这样就避免了直接访问数据库,从而摆脱了对数据库的依赖,减轻了系统负载压力。而当用户进行数据查询时,检索程序会从构建好的索引库

中进行查找,并将检索的结果直接反馈给用户。该系统处理用户查询的详细过程是:首先用户输入检索条件,对用户输入的检索条件进行预处理(分词、词干抽取等),并将分词结果转换为相应的 term\_id,然后根据 term\_id 从二级索引线性散列索引中检索包含该检索条件的页链,最后从倒排索引库中将查找到的结果,经过文档评分、结果集排序、分组等操作将文档输出给终端用户。

## 5 实验测试

文中的测试环境为:Centos 6.4 操作系统,2G 内存,ext3 文件系统。测试结果为:查询每万条数据平均耗时 22.28 ms,插入每万条倒排索引数据平均耗时 43.58 s,删除每万条倒排索引数据平均耗时 37.43 s。其中,线性散列索引的插入删除过程平均耗时见图 3。

由此可见,基于线性散列的倒排索引查询速度极快,在 20 多个毫秒内即可完成 1 万条数据的检索,但插入删除过程却相对耗时,完成 1 万条数据的插入删除平均耗时在 40 s 左右。究其原因在于插入删除过程涉及倒排索引的多路归并等大量耗时操作。由图 3 可知,在极个别情况下线性散列倒排索引的耗时极大,这是由于插入的数据处于倾斜直线及两侧分布情况,溢出链会导致它的性能很差。总体来讲,基于线性散列的倒排索引查询性能较优,能很好地满足大规模用户并发检索的情况。

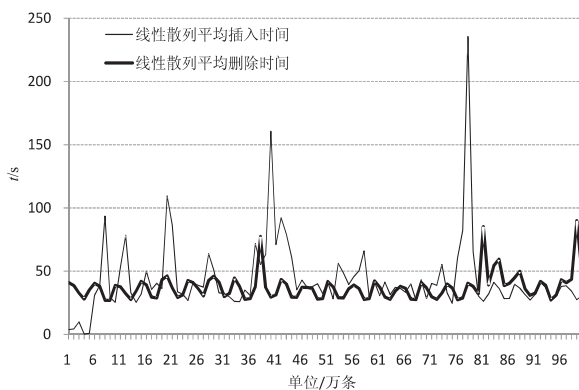


图3 线性散列插入删除平均时间

## 6 结束语

文中首先对纯数据库存储、纯文件存储以及两种方式混合存储这三种索引存储结构从实现原理以及优劣等方面进行了比较,最终确定了以纯文件存储作为实际索引存储模式,然后对线性散列的倒排索引的构建对索引模型的构建流程进行详细地介绍,最后对系统检索模型的实现原理进行了说明。

通过测试可以发现,I/O 仍然是制约检索性能的一个重要瓶颈。因此如何进一步降低系统 I/O 成为笔

者下一步需要着重解决的重点内容。在下一步的工作中,笔者希望增加对以下方面的研究:

(1)研究分布式检索机制,解决单服务器下服务器负载均衡较重的问题;

(2)加强对文档检索处理机制的研究。对文档和查询构建检索模型,提升系统检索处理能力,从而促进和改善在文本信息检索和模型构建方面的研究。

参考文献:

[1] 申展,江宝林,陈伟,等.全文检索模型综述[J].计算机科学,2004,31(5):61-64.

[2] 曾海泉,刘永丹,宋扬,等.基于互关联后继树的多时间序列关联模式挖掘[J].计算机研究与发展,2003,40(7):934-940.

[3] 邓攀,刘功申.一种高效的倒排索引存储结构[J].计算机工程与应用,2008,44(31):149-152.

[4] 王冬,左万利,赫枫龄,等.一种增量倒排索引结构的设计与实现[J].吉林大学学报:理学版,2007,45(6):953-958.

[5] 吐尔洪·吾司曼,维尼拉·木沙江.维、哈、柯多语种搜索

(上接第192页)

参考文献:

[1] 薛军,纪敦,李猛,等.飞机结构应变信号的采集与预处理系统[J].数据采集与处理,2009,24(S):315-318.

[2] 周鸣争,楚宁,周涛,等.一种基于能量约束的传感器网络动态数据融合算法[J].仪器仪表学报,2007,28(1):172-175.

[3] 刘慧,唐胜武,简荣坤.基于软件补偿算法的温度压力场测试系统设计[J].仪表技术,2011(6):39-42.

[4] 张有凤,王钦若,张慧.基于压力检测的高精度数据采集系统[J].陕西理工学院学报:自然科学版,2006,22(3):55-58.

[5] 凌振宝,王君,张瑞鹏.基于非晶态合金感应式传感器补偿电路的设计[J].传感技术学报,2003,16(2):207-209.

[6] 朱旭,张世中,胡哲.感应式磁传感器的补偿电路[J].物探与化探,2012,36(6):970-974.

[7] 张宏涛,薛军晓.FPGA在温度补偿气压测量系统设计中的应用[J].电子技术应用,2013,39(4):65-67.

[8] 曹建荣,刘辉.人工神经网络在电容式压力传感器设计

引擎中索引器的研究[J].新疆大学学报:自然科学版,2011,28(2):132-135.

[6] 李晶皎,何敬禹,郑牧野,等.文件系统索引结构的研究[J].东北大学学报:自然科学版,2004,25(4):318-321.

[7] 陈立.全文检索引擎的设计研究[J].现代情报,2007,27(10):223-225.

[8] 杨安生.基于倒排表的中文全文检索研究[J].情报探索,2009(7):77-80.

[9] 郑榕增,林世平.基于Lucene的中文倒排索引技术的研究[J].计算机技术与发展,2010,20(3):80-83.

[10] 苏潭英,郭宪勇,金鑫.一种基于Lucene的中文全文检索系统[J].计算机工程,2007,33(23):94-96.

[11] 熊回香,夏立新.基于词索引的中文全文检索关键技术及其发展方向[J].中国图书馆学报,2007,33(4):45-49.

[12] 陈慧杰,李建伟.动态散列目录扩展算法的研究[J].太原科技大学学报,2013,34(5):321-324.

[13] 李蔚,陈亚峰,王艳军.动态散列算法及其改进[J].郑州轻工业学院学报:自然科学版,2011,26(3):92-95.

[14] 郑德舜.一种高效的散列查询算法[J].南京邮电大学学报:自然科学版,2006,26(2):92-96.

上的应用[J].自动化仪表,2002,23(8):14-16.

[9] Lefeuve E,Badel A,Richard C,et al. A comparison between several vibration-powered piezoelectric generators for standalone systems[J].Sensors and Actuators A,2006,126:405-416.

[10] Fang H B,Liu J Q,Xu Z Y,et al. Fabrication and performance of MEMS-based piezoelectric power generator for vibration energy harvesting[J].Microelectronics Journal,2006,37(11):1280-1284.


[11] 郑国良.传感器非线性处理方法[J].传感器技术,1991(1):40-43.

[12] 孙慧卿,郭志友.压力传感器及误差补偿[J].传感器世界,2002,8(3):14-16.

[13] 徐军.用单片机软件实现传感器温度误差补偿[J].现代电子技术,2002(10):97-99.

[14] Williams S,Thompson H,Hufford M,et al. An improved CMOS ring oscillator PLL with less than 4ps accumulated jitter[C]//Proceedings of IEEE custom integrated circuits conference. [s.l.]:IEEE,2004:151-154.

线性散列在全文检索中的应用研究

作者：[束文杰](#)，[时亚南](#)，[于国欣](#)，[SHU Wen-jie](#)，[SHI Ya-nan](#)，[YU Guo-xin](#)  
作者单位：[新疆维吾尔自治区特种设备检验研究院, 新疆 乌鲁木齐, 830011](#)  
刊名：[计算机技术与发展](#)  
英文刊名：[Computer Technology and Development](#)  
年，卷(期)：2015(6)

引用本文格式：[束文杰](#).[时亚南](#).[于国欣](#).[SHU Wen-jie](#).[SHI Ya-nan](#).[YU Guo-xin](#) [线性散列在全文检索中的应用研究](#)

[期刊论文]-[计算机技术与发展](#) 2015(6)