

搜索引擎倒排文件的一种分块组织技术

彭 波, 李晓明

(北京大学计算机科学技术系, 北京 100871)

摘 要: 检索效率对大规模信息检索系统至关重要. 本文基于 Web 搜索应用背景, 针对用户查询的统计特性, 提出了一种分块组织倒排文件的方法. 通过建立检索性能模型, 进行分析和仿真实验, 结果表明这一方法下的检索算法可以有效的减少检索执行时间, 并得到这一组织方式中分块参数的优化选择方法.

关键词: 搜索引擎; 信息检索; 倒排文件; 检索效率

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2005) 02-0358-05

A Blocking Inverted File Structure for Search Engine

PENG Bo, LI Xiao-ming

(Department of Computer Science & Technology, Peking University, Beijing 100871, China)

Abstract: The efficiency of retrieval system is crucial for large-scale information retrieval systems. By analyzing the documents and the users' query logs of a real search engine, a blocking inverted file structure is proposed. Simulation results show that the retrieval algorithm under the new organization of the inverted file can decrease its execution time significantly, and the optimal parameter selection for this blocking organization is discussed.

Key words: search engine; information retrieval; inverted file; retrieval efficiency

1 引言

倒排文件是大规模信息检索系统常用的一种索引组织技术. 在 Web 检索环境下, 搜索引擎索引数据量和用户量都不断增加, 从而使得优化倒排文件的组织、改进检索算法, 不断提高系统检索效率成为一个被关注的研究课题. 早期研究主要从并行与分布式系统角度研究倒排文件数据分布方式对性能的影响^[1,2]. 对大规模文档数据的倒排文件索引, 检索算法执行中对磁盘上倒排表 (posting list) 的访问开销是影响性能的重要因素, 从而有大量的研究集中在倒排文件的压缩技术上^[3~6], 力图通过数据压缩, 减少检索执行中的磁盘访问开销, 提高系统整体效率. 研究表明, 有序整数序列通过游程编码可以高效压缩, 因此通常将倒排表数据项按文档编号增序组织, 以提高压缩效率. 另一方面, 在保证查询结果质量的条件下如何减少查询过程中所涉及的数据量也是一个重要研究内容. 文[7]提出在压缩索引中插入同步点, 使得检索算法可以跳过某些数据块, 减少数据处理时间. 文[8]提出了过滤向量空间模型以及相应的倒排文件优化组织方法, 通过把倒排表按索引词词频降序组织, 结合过滤向量空间模型的检索算法, 可以只读取和处理少量倒排表数据得到检索结果, 且不影响检索效果. 文[9]中描述了一种在搜索引擎系统常用的技

术, 即对文档中的标题、超链文本 (anchor text) 等内容单独建立索引, 检索时只有当这个小索引没有返回足够的结果, 才去访问完整文档数据索引. 搜索引擎和传统的信息检索应用相比, 用户查询模式差异很大, 用户输入的查询词通常较少, 这时传统的信息检索模型, 如文[8]并不适用. 搜索引擎一般考虑结合布尔检索与多种影响文档排序的权值计算方法, 比如查询词出现的位置, 以及查询词间的位置关系等.

综合上述相关研究, 本文提出一种倒排表数据的分块组织方法. 其基本思想是将倒排文件数据分块存储, 使块内数据按文档编号增序排列, 以利于高效数据压缩, 而块间保持文档权值降序组织, 以求只读取少量倒排表数据而得到检索结果, 总体上提高检索效率. 为此, 本文研究了如下问题:

倒排文件的检索性能模型.

分块组织方式下检索算法的执行时间分析.

不同分块组织方式对检索效率的影响.

2 倒排索引的检索性能模型

2.1 文档模型

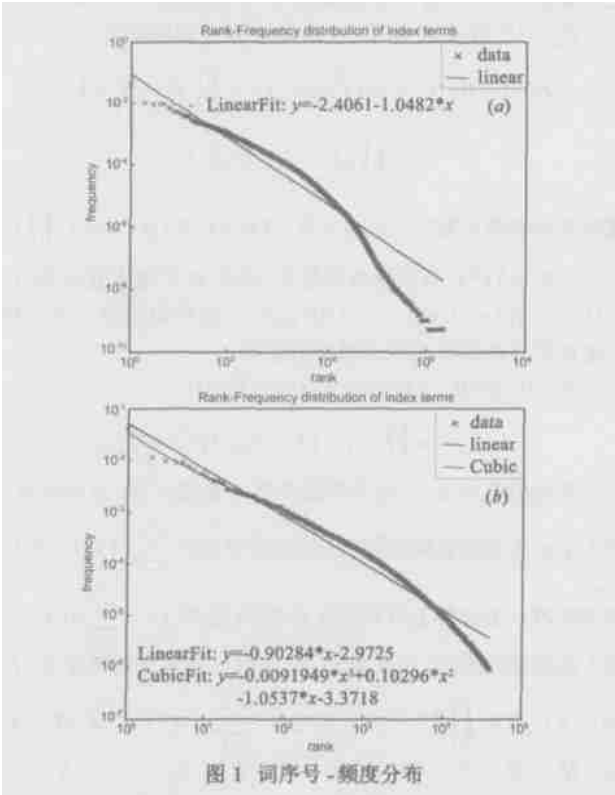
从天网搜索引擎收集的网页集合中随机抽取出一批文档, 建立倒排索引, 统计得到文档模型的参数如表 1:

表 1 文档模型参数

参数	数值	说明
D	2603035	文档总数
V	$ V = 2315493$	索引词(term)集合
W	720	平均每篇文档的索引词数(文档词数分布近似为对数正态的形式,此处简化取均值)
T	$1.9150e+009$	包含重复出现在内的索引词总数
	1.6	倒排项数据压缩系数. 每个文档编号和位置编号压缩后的平均字节数
$Z(j)$	$Z(j)$	序号为 j 的索引词出现的概率,即 $Pr(\text{term} = j)$
$L(j)$	$L(j)$	索引词 j 的倒排项数据长度

对大规模英文文本数据集,单词词频分布遵循 zipf's law,即单词按出现的频度 f 降序排列,赋予序号 r ,有 $f(r) = c/r$ (c 为常数, $c > 1$). 对中文文本的统计研究表明,词频分布不严格遵循 zipf's law,但大体近似 [10, 11]. 图 1 是对数坐标下索引词的序号-频度分布图,图 (a) 是全部索引词按总体出现频度(即在所有文档中出现的总次数除以 T) 降序排列下的频度分布,可以看到这个和 zipf's law 有较大差异. 图 (b) 是序号小于 40000 的高频词词频分布. 类似文 [10], 采用扩展 zipf's law 的高次多项式拟合,可以更精确描述索引词的分布情况. 本文采用 3 次多项式拟合, $Z(j)$ 为:

$$Z(j) = \exp(-0.0091949 * \log^3(j) - 3 + 0.10296 * \log^2(j) - 1.0537 * \log(j) - 3.3718)$$



$L(j)$ 是索引词 j 的倒排表数据长度. 倒排表数据由索引词出现的所有文档编号和位置编号(采用压缩编码), 以及每

个出现文档的权值(按单字节计) 构成. 假设文档中词语是独立且同分布的, 一篇文档则是在一个词典上随机进行词语选取的结果. 一个文档 d 包含索引词 j 的概率为 $Pr(j \in d) = 1 - (1 - Z(j))^W$, 则词 j 的文档频率 $DF(j) = D * Pr(j \in d) = D * (1 - ((1 - Z(j))^W))$, 词频 $TF(j) = T * Z(j)$. 于是词 j 的倒排表数据长度 $L(j) = (T * Z(j) + DF(j)) * 1.6$.

2.2 用户查询模型

用户查询模型包括用户查询(query)中的查询词(term) 个数分布以及索引词被查询的概率分布. 参数如表 2:

表 2 用户查询模型参数

参数	数值	说明
$Q(j)$		序号为 j 的索引词被查询的概率, 即 $Pr(\text{query term} = j)$
$Qn(i)$		查询词个数分布, 即 $Pr(\text{query term count} = i)$
$E(Qn)$	2.5948	查询中平均包含查询词的个数

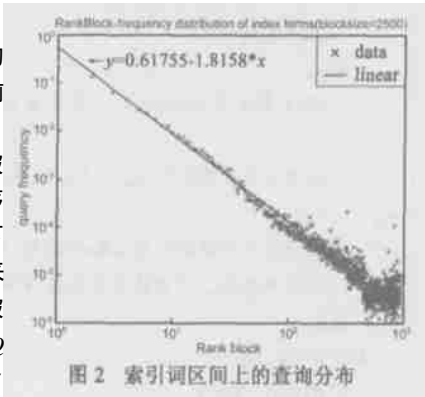
对搜索引擎用户查询日志的统计已经有很多研究结果. 文 [12] 的统计表明, 用户输入查询所含词的平均个数为 2.4, 包含多于 7 个查询词的查询不到 1.8%. 本文对天网搜索引擎用户查询日志 [15] (编号 YQ-QUERYLOG.021203) 进行了统计, 针对未被系统查询结果缓存命中的查询, 共 3,522,968 个记录, 统计结果如表 3, 基本与文 [12] 一致:

表 3 用户查询词个数分布

查询词个数	1	2	3	4	5	6	7
比例	0.2787	0.2877	0.2220	0.1071	0.0512	0.0252	0.0130

人们通常假设索引词独立且同分布, 但索引词被查询的概率分布没有一致的结论. 文 [2] 中使用 V 中前 $u|V|$ 个索引词上的均匀分布, ($0 < u < 1$), 在索引词序号 $u|V|$ 处截止, 忽略更低频的索引词被查询的可能性. 文 [1] 认为被查询概率与索引词本身出现的概率成正比, 也同样在索引词序号 $u|V|$ 处截止. 把索引词序号划分为同样宽度的词序号区间, 考察各个区间上的被查询频度分布, 结果如图 2 (区间大小为 2500).

$Pr(\text{block} > 10) = 0.10213$, 即 90% 的查询词都落在前 25000 个索引词上. 图 2, 索引词区间被查询的概率随着序号增大按负幂率下降, 可以按区间来粗略估计索引词被查询概率的分布 $Q(j) = Pr(\text{block} = (j/\text{blocksize}))$, 有:



$$Q(j) = e^{-0.61755 * (j/2500)^{1.8158}}$$

2.3 计算机系统性能参数

通过在天网索引节点上测量, 得到倒排文件检索算法相关的系统性能参数如表 4:

表 4 计算机性能参数

参数	数值	说明
	5.7ms	每一次 I/O 访问的准备时间
	0.017us	I/O 访问中每字节数据的传输时间
T_d	5.2e - 5ms	索引项压缩编码每字节解码时间
T_{ci}	1.0e - 6ms	比较两个整数的时间(倒排表合并处理每字节时间)
$T(L)$		磁盘系统读取长为 L 的一块连续数据需要的 I/O 时间.

根据磁盘的访问特性,有: $T(L) = T_{\text{准备}} + T_{\text{传输}} * L$. 其中 $T_{\text{准备}}$ 包括了文件系统管理开销,磁盘 I/O 命令的启动时间、磁盘的寻道时间和旋转延迟; $T_{\text{传输}}$ 包括了数据传输时间以及在内存中移动耗费的时间.

3 分块索引项的组织策略

分块组织的基本思想是结合按文档编号组织与按权值组织,把倒排表数据按每项的权值大小分为若干块,每一块(block)内按文档号增序压缩保存,块间按权值降序保存,即:若 $i < j$,则 block i 内文档权值 $>$ block j 内文档权值. 引入索引词 k 的分块因子 (k, j) ,表示词 k 的第 j 块数据块长度比例,有 $\text{blocksize}(k, j) = (k, j) * L(k)$; $\text{blockcnt}(k)$ 表示索引词 k 的数据块数,当 $j > \text{blockcnt}(k)$ 时,置 $(k, j) = 0$.

对查询 $Q = \{q_1, q_2, \dots, q_m\}$, q_i 按倒置文档频率(idf)降序排列,检索算法返回最相关的 K 篇文档. 算法描述如下图 3:

```

a. 初始化,结果集合  $R = \emptyset$ , 权值累加器  $A = 0$ 
b. for  $j$  from 1 to  $\text{blockcnt}(q_m)$  do
  begin
    b1. for each  $q_i$  in  $Q$  do
      begin
        读取  $q_i$  的第  $j$  块倒排表数据  $\text{readblock}(q_i, j)$ 
      end
    b2.  $R_j = \emptyset$ , 清空本轮结果集合
    for each  $q_i$  in  $Q$  do
      begin
        b2.1 解压缩  $\text{decode}(q_{ij})$ , 得到  $q_{ij}$  的文档集  $\{d_{qij}\}$ , 权值  $\{w_{qij}\}$ 
        b2.2 合并文档集合  $\{d_{qj}\} = \{d_{q1j}\} \cup \{d_{q2j}\} \cup \dots \cup \{d_{qmj}\}$ ,  $\{w_{qj}\} = \{w_{q1j}\} \cup \{w_{q2j}\} \cup \dots \cup \{w_{qmj}\}$ 
        b2.3 执行布尔查询  $R_j = \text{Boolean}(R_j, \{d_{qj}\})$ .
        b2.4 使用  $\{w_{qj}\}$  更新结果集合  $R_j$  中元素对应的权值  $A$ .
      end
    b3.  $R = R \cup R_j$ 
    If  $|R| \geq K$  goto c
  end
c. 根据累加器  $A$  从结果集合  $R$  中选取最大权值的  $K$  个结果 select  $(R, A, K)$ 

```

图 3 分块组织倒排文件的检索算法

在算法执行过程中,依次从前往后读入各个查询词的倒排表数据块,执行布尔查询得到结果集合,并为结果赋予相关

性权值,算法在 $|R| \geq K$ 时停止. 因为相关性权值由倒排表数据中的文档权值累加计算得到,在结果集中文档权值高的文档,相应相关性权值也高,其正确性可以保证.

若此算法在第 n 轮循环结束. 主要三个步骤的时间开销为:

$$\begin{aligned}
 T_{b1} &= \sum_{i=1}^m \left[n * T_{\text{准备}} + \sum_{j=1}^n L(q_i) * T_d * (q_i, j) \right] \\
 T_{b2} &= \sum_{i=1}^m \left[L(q_i) * T_d * \sum_{j=1}^n (q_i, j) \right] \\
 &\quad + \sum_{i=1}^m \left[L(q_i) * 2 * T_{ci} * \sum_{j=1}^n \sum_{k=1}^j (q_i, i) \right] \\
 T_c &= (|R| + 2) * |R| * \log(K) * T_{ci}
 \end{aligned}$$

其中, T_{b1} 为读取倒排表数据的时间. T_{b2} 为倒排表数据处理时间,包括解压缩和归并操作. $b2$ 步骤中,需要把解码的倒排表数据保留在临时内存中,避免后续访问时的重复解压缩开销,这样算法需要较大的临时内存来保存处理过的倒排数据项,这是一个缺点. T_c 为结果集中选取权值最大的 K 个元素,使用最小堆选择算法^[6],时间估计为 $|R| + 2 * |R| * \log(K)$. 检索算法执行时间 $T(Q) = T_{b1} + T_{b2} + T_c$.

4 实验设计

4.1 算法仿真

算法中的布尔检索使用搜索引擎一般默认的 AND 运算. 对查询 $Q = \{q_1, q_2, \dots, q_m\}$, 检索得到 $R = \{d | q_i \subseteq d\}$. 根据文档模型,一个文档 d 包含索引词 i 的概率为 $\text{Pr}(i \subseteq d) = 1 - (1 - Z(i))^W$. 在查询 Q 下,文档 d 在 R 中的概率为:

$$\begin{aligned}
 \text{Pr}(d \in R | Q) &= \text{Pr}\left(\bigcap_{i=1}^m q_i \subseteq d\right) = \prod_{i=1}^m \text{Pr}(q_i \subseteq d) \\
 &= \prod_{i=1}^m (1 - (1 - Z(q_i))^W)
 \end{aligned}$$

此时查询结果个数为: $|R|_Q = D * \text{Pr}(d \in R | Q) = D * \prod_{i=1}^m (1 - (1 - Z(q_i))^W)$. 再根据查询模型,查询 Q 的概率分布为: $\text{Pr}(Q) = Q(q_1) * Q(q_2) * \dots * Q(q_m)$. 于是可以得到一个文档 d 在 R 中与查询 Q 的联合概率分布为:

$$\begin{aligned}
 \text{Pr}(d \in R | Q) &= \text{Pr}(d \in R | Q) * \text{Pr}(Q) \\
 &= \prod_{i=1}^m ((1 - (1 - Z(q_i))^W) * Q(q_i))
 \end{aligned}$$

分块组织方式下,检索算法在第 n 轮循环处,对查询词 i 读入了 n 块倒排表数据块,占总数据长度的 $\sum_{j=1}^n (q_i, j)$,可以认为此时一个位置上出现词 i 的概率为 $Z(i) * \sum_{j=1}^n (q_i, j)$. 按上面同样的推导,在查询 Q 下,文档 d 在 R 中的概率为: $\text{Pr}(d \in R | Q) = \prod_{i=1}^m \left[1 - \left(1 - Z(q_i) * \sum_{j=1}^n (q_i, j) \right)^W \right]$. 此时查询结果个数为: $|R|_Q = D * \prod_{i=1}^m \left[1 - \left(1 - Z(q_i) * \sum_{j=1}^n (q_i, j) \right)^W \right]$,以此判断算法是否结束退出.

检索算法的性能以执行时间 $T(Q)$ 衡量,计算方法如节 3.

4.2 问题分析

分块组织的检索算法读取和解压缩的索引数据量较少,代价是更多的磁盘 I/O 访问次数和数据处理时间.其有效性归结为如下问题:在文档模型和查询模型条件下,是否有 $|R| \gg K$? 分块检索算法能否读取少量数据,而达到 $|R| \approx K$,尽快结束? 第一组实验研究 $|R|$ 与 K 的关系.

第二个问题:分块检索算法是否能减少执行时间? 第二组实验计算比较 $T(Q)$ 在两种索引组织方式下的差异.

第三个问题:分块因子 α 的选择对检索算法效率的影响.分块因子 α (i, j) 的选择有两个维度,一是对同一个索引词 i ,它的各个倒排项数据块长度如何划分.这由文档权值的分布决定.与文[8]中按文档内的词频分块组织不同的是,搜索引擎里的文档权值综合了词频、词的出现位置多种因素,其频度分布相对原始的文档内词频频度分布要平缓得多.统计表明平均情况下:文档权值最小的倒排项占总数的 $1/4 \sim 1/3$,而高权值(出现在标题、超链文字以及摘要文字)的倒排项比例基本相当.这说明对同一索引词可以使用相同块长度,且块数目不宜太多.的另一选择是对不同索引词的取值.本文讨论两种选择策略:规定 α 的一个最小值 α_{\min} ,给高频索引词使用,随着索引词序号增加, α 也相应增大,只到索引词倒排项长度很小, α 取 1 为止.一种策略是,在这个索引词序号区间上,按等宽的序号区间长度间隔增加,从 α_{\min} 到 1.另一策略考虑倒排项数据长度基本和 $Z(j)$ 一样,按负幂率下降,上述等间距增加 α 的方式可能会让过多短倒排项数据被分块.基于这一分析,设计按几何级数来设定区间宽度,增加 α 值.第三组实验计算比较 $T(Q)$ 在这两种分块策略以及不同 α_{\min} 的设定下的差异,选择最佳的分块方式.

5 仿真实验和结果

5.1 $|R|$ 与 k 比较

基于查询模型随机生成 100000 组查询,通过检索算法仿真,统计 $Pr(|R| > k)$ 的值.在不分块情况下 $|R|$ 与 k 比较结果如图 4.图中 CDF 线为按查询长度分布累计的 $Pr(|R| > k)$ 值.随着查询词长度增加,结果集合迅速减小.但由于查询词本身的比例也快速下降,从 CDF 线可以看到,长查询对总体的影响并不明显.本文模型不考虑查询词间的相关性,可以预计在实际运行环境中 $Pr(|R| > k)$ 值会更高,对分块检索更有利.图 4 比较了 k 值相差一个数量级的 $|R|$ 值情况, $Pr(|R|$

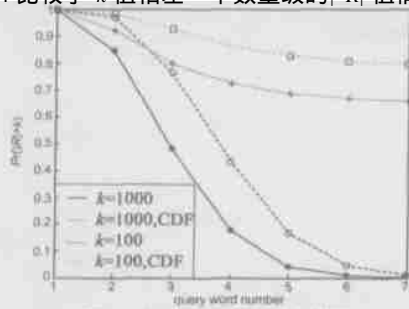


图 4 $|R|$ 与 k 比较的仿真结果

> 1000) 最低为 0.66,占实际 $Pr(|R| > 100) = 0.80$ 的 82%,这说明在不分块情况下确实会有相当数量的查询检索了过多的结果,分块策略的前提成立.

5.2 $T(Q)$ 以及分块因子 α 对 $T(Q)$ 的影响

基于查询模型随机生成 100000 组查询,通过检索算法仿真,统计平均执行时间.图 5 为等区间分配方式各查询长度下累计的平均执行时间图, α_{\min} 分别取 1, 1/2, 1/4, 1/8, 1/16 (取 1 时即普通非分块组织方式).图 6 为几何级数设定区间下的结果图.

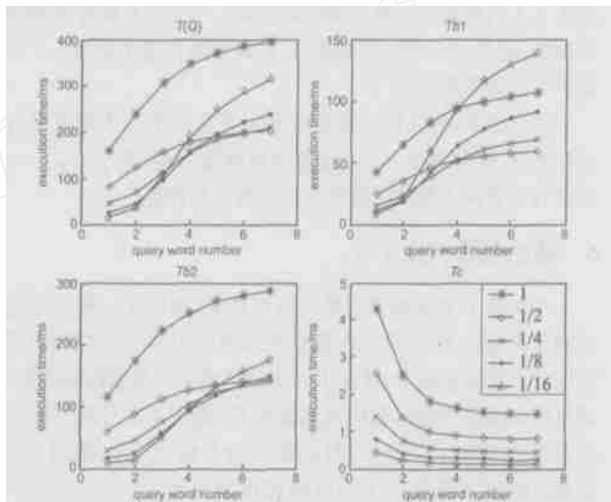


图 5 按词区间均匀分块下的 $T(Q)$

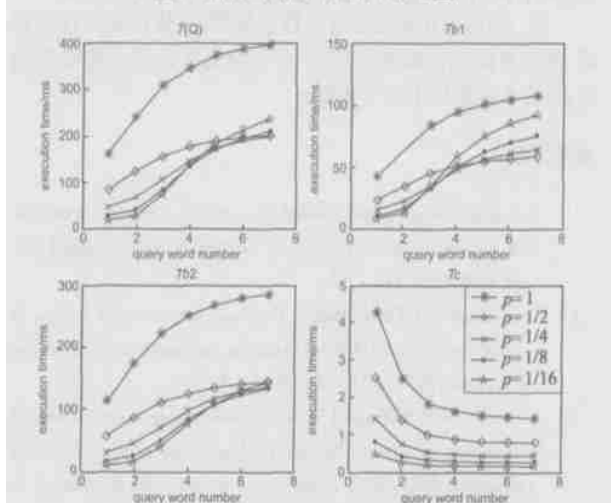


图 6 按几何级数区间分块下的 $T(Q)$

从图 5 和图 6 的 $T(Q)$ 图上看,分块组织检索算法的平均执行时间比标准组织方式小,最小能减少 50% 左右,这说明分块方式有效性.

分析图 5 的 $Tb1$ 图,当查询的词个数少 3 个时,分块方式可以有效减少磁盘访问时间,这时 α_{\min} 越小,分块越多,越有效.随着查询词个数增加, α_{\min} 小的分块方式带来更多的磁盘访问次数,因读取数据量减少带来的性能提升被逐步抵消, α_{\min} 越小,抵消的速度越快.最后系统平均 $Tb1$ 时间, $\alpha_{\min} = 1/16$ 时,比不分块还要差,而简单的 1/2 分块方式却得到了最好的性能. $Tb2$ 图,因为 $Td \gg Tci$, $Tb2$ 的时间主要由倒排项数据

解压缩决定。 \min 越小, 算法结束时读入的倒排表数据可能越少, 总时间 $Tb2$ 也越小。但随着查询词个数增加, 检索算法快速结束的可能性下降, 同时还会大量增加归并过程的次数, 综合起来, $\min = 1/16$ 最后比其他分块方式都差, 类似 $Tb1$, 简单的 $1/2$ 分块方式性能也很好。 Tc 图, \min 越小, Tc 就越小, 不过 Tc 相对 $Tb1$ 和 $Tb2$ 太小, 它的改进对整体性能影响不大。

图 6 与图 5 比较, 可以看到随着查询词个数增加, $Tb1$ 、 $Tb2$ 在 \min 小的情况下, 性能损失相对减少。比如在查询词长度 4 处, $\min < 1/2$ 时, 性能都比 $1/2$ 好。可以让检索算法对长度大于 4 的查询回退到基本检索方式, 一次读入所有倒排表数据块, 避免 $Tb1$ 中的性能损失, 可以预期获得比简单 $1/2$ 分块方式更好的性能。

综合起来, 按几何级数设定区间宽度比平均区间宽度方法性能好。同时过小的 \min 值会导致性能下降, 取 $\min = 1/4$, 再结合长查询下算法回退可以得到较好的性能。

6 结论与进一步工作

本文基于 Web 搜索引擎的应用背景, 提出了一种分块组织倒排文件的方法。首先在统计天网搜索引擎的文档和用户查询日志数据基础上, 建立了一个倒排索引的检索性能模型。通过对该模型的分析 and 仿真实验研究, 表明了分块方法下检索算法的执行时间可以明显减少。在分块设计上, 采用几何级数设定区间宽度比平均区间宽度效果要好, 当分块因子 \min 取 $1/4$, 结合长查询下算法回退可以得到较好的性能。

进一步的工作将完善这一性能模型, 继续研究在针对短语、邻近查询优化的索引结构^[13]下的倒排文件分块组织问题, 并在实际系统中进一步验证。

参考文献:

- [1] B-S Jeong, E Omiecinski. Inverted file partitioning schemes in multiple disk systems[J]. IEEE Transactions on Parallel and Distributed Systems, 1995, 6(2): 142 - 153.
- [2] A Tomasic, H Garcia-Molina. Performance of inverted indices in shared-nothing distributed text document information retrieval systems [A]. Proc PDIS Conf [C]. San Diego, CA, 1993.
- [3] F Scholer, H E Williams, J Yiannis, J Zobel. Compression of inverted indexes for fast query evaluation [A]. Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval [C]. Tampere, Finland, 2002. 222 - 229.
- [4] G Navarro, E Maura, M Neubert, N Ziviani, R Baeza-Yates. Adding compression to block addressing inverted indexes [J]. Kluwer Information Retrieval Journal, 2000. 3(1): 49 - 77.

- [5] Anh NgocVo, Alistair Moffat. Compressed inverted files with reduced decoding overheads [A]. Proceedings of the 21st International Conference on Research and Development in Information Retrieval [C]. New York City: ACM Press, August 1998. 290 - 297.
- [6] Witten I H, Moffat A, Bell T C. Managing Gbytes: Compressing and Indexing Documents and Images [M]. Van Nostrand Reinhold, New York, 1994.
- [7] A Moffat, J Zobel. Self-indexing inverted files for fast text retrieval [J]. ACM Transactions on Information Systems, 1996, 14(4): 349 - 379.
- [8] M Persin, J Zobel, R Sacks-Davis. Filtered document retrieval with frequency-sorted indexes [J]. Journal of the American Society for Information Science, 1996, 47(10): 749 - 764.
- [9] S Brin, L Page. The anatomy of a large-scale hypertextual Web search engine [A]. In Proceedings of the 7th WWW conference [C]. Computer Networks, Amsterdam, 1998.
- [10] Lua K T. Frequency-rank curves and entropy for Chinese characters and words [J]. Computer Processing of Chinese & Oriental Languages, 1994, 8(1): 37 - 52.
- [11] 关毅, 王晓龙, 张凯. 现代汉语计算语言模型中语言单位的频度-频级关系 [J]. 中文信息学报, 1998, 13(2): 8 - 15.
- [12] Amanda Spink, Dietmar Wolfram, B J Jansen, Tefko Saracevic. Searching the web: The public and their queries [J]. Journal of the American Society for Information Science, 2001, 53(2): 226 - 234.
- [13] D Bahle, H E Williams, J Zobel. Efficient phrase querying with an auxiliary index [A]. Proceedings of the ACM-SIGIR Conference on Research and Development in Information Retrieval [C]. Tampere, Finland, August 2002. 215 - 221.
- [14] 天网搜索引擎. <http://e.pku.edu.cn/Z/OL>.
- [15] 中国 WEB 信息博物馆. <http://www.infomall.cn/Z/OL>.

作者简介:



彭 波 男, 1975 年出生于新疆喀什市, 现为北京大学计算机系博士研究生, 研究方向为网络与分布式处理。E-mail: pb@net.pku.edu.cn.

李晓明 男, 1957 年出生于湖北省荆州地区, 现为北京大学教授, 博士生导师, 研究方向为并行与分布式处理、Internet 与 Web 技术。Email: lxm@pku.edu.cn.