

振动外分类算法研究^{*}

董玉萍¹, 张志立^{2,3}, 齐德昱³

(1. 广州美院计算中心, 广东 广州 510261;
2. 许昌学院网络中心, 河南 许昌 461000;
3. 华南理工大学计算机学院, 广东 广州 510640)

摘 要: 给出了一种新的外分类算法: 振动外分类。它不必预先产生初始归并段, 即可得到预期的效果, 在特定的数据和硬件配置下, 性能上超过了二路平衡归并法和二路多步归并法。详细分析了其系统开销, 给出了算法正确性证明, 用 C 语言在 PC586 及以上机得以实现。

关键词: 振动外分类; 时间复杂度; 存储开销

中图分类号: TP301.6 **文献标识码:** A **文章编号:** 0529-6579 (2004) 04-0118-04

分类, 即排序, 它是计算机科学中一项复杂而又非常重要的技术。目前, 在内分类研究上已有了很大的进展, 除了已知的霍尔快速分类、堆分类等^[1-3] 优秀的分类算法外, 在特殊问题情况下, 线性时间复杂度的算法也被许多软件工作者提出^[4-8]; 但是在外分类研究上却没有多大的进展。特别是随着计算机技术渗透到我国的各行各业, 迫切需要对大量数据任意部分的快速存取, 因此一种优秀的外分类算法对于数据库系统的查询优化, 实现数据的快速存取, 具有重要的意义^[9]。

本文给出了一种新的外分类算法——振动外分类。它不必预先产生初始归并段, 即可得到预期的效果, 在特定的数据和硬件配置下, 性能上超过了已知的二路平衡归并法和二路多步归并法。

1 振动外分类的算法思想

振动外分类算法是在求一组数据的最小值和最大值基础上提出来的。它的基本思想是: 通过一趟分类工作, 将文件的最小块和最大块求出来, 然后再对去掉最小块和最大块的文件进行外分类, 直至内存能容纳下该文件未被分类的块时, 再进行最后一趟分类, 得到整个文件有序。

由于这种趟分类方式使记录组成的不同块序列按大小同时向文件 f 的两端移动, 如同机械振动机, 使被振动的所有大小不同的石块同时向上下两头跑一样, 故此方法称为振动外分类法。

当要分类的数据文件太长, 以致内存无法容纳时, 我们考虑的问题就显著增多了。即从编制一个功能优异的子程序问题改变为由于考虑 I/O 和 CPU

的均衡利用, 数据的块因子和缓冲技术以及各部分程序间的数据交换等来权衡整个程序的问题。外分类过程的效率必须用读、分类和写全部数据元素所需的时间总和来测定, 而一个排序算法本身仅能确定其中的一部分时间。即:

外分类所需总的时间 =

内分类时间($s * ut_s$) +

外存信息读/写时间($d * t_{IO}$)

其中 s 是分类趟数 (见定义 3); ut_s 是对 u 个记录进行内分类的时间; t_{IO} 是外存信息读/写时间; d 为读/写次数。所以提高外分类的效率应着眼于减少外存读/写的次数^[1, 2, 10]。

考虑到上述诸因素, 为了讨论方便, 本文中有如下约定, 并先介绍几个术语。

约定 1 一个文件 f , 其每一个记录中只有一个排序关键字; 一个块因子由一个或多个记录组成。

约定 2 在诸键上确定一个次序关系 “ $<$ ”, 使得对于任意 3 个键值 a, b, c , 下列条件成立:

① $a < b, a = b, b < a$ 3 个可能性中有一个可能成立;

② 如果 $a < b$ 而且 $b < c$ 则 $a < c$ 。

本文中分类后的文件 f 中所有记录关键字都按非递减次序排列。

约定 3 外分类时, 假定内存中一次可容纳文件 f 的 M 块。当然在内存中还需要留下一些额外块作计算用, 诸如在 M 块的值中找出最小、最大值以及存储输出等额外用途。

* 收稿日期: 2004-03-02

基金项目: 广东省重大科技专项资助项目 (20034310)

作者简介: 董玉萍 (1958 年生) 女, 讲师, 通讯联系人; 张志立, E-mail: zzl@xctc.edu.cn

定义 1 若文件 f 按块的序号增 1 的次序读入内存或写出, 则称顺序读入或写出。

定义 2 若文件 f 按块的序号减 1 的次序读入内存或写出, 则称逆序读入或写出。

定义 3 把整个文件 f 读入内存经处理后又写出的过程称为一趟外分类。

2 振动外分类的算法描述及实现

振动外分类算法 Skexsort 假定内存容量 $M=3$ 块, 分别对应内存的 I, II, III 三个缓冲区; j, k, l 分别为文件 f 的首、尾块和中间块指针。该算法原理见图 1。

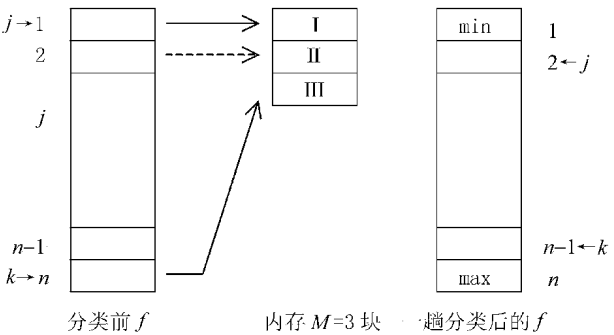


图 1 算法 S 原理图

Fig 1 Principle illustration of algorithm S

算法描述如下:

S1. [文件分块] 将文件 f 分成 n 块。

S2. [置块指针初值] $j \leftarrow 1, k \leftarrow n, l \leftarrow 3$ 。

S3. [对 n 进行循环] 若 $n > 3$ 则实施步骤 S4 ~ S11, 否则转 S12。

S4. [将 f 上的第 $j, j+1$ 块和第 k 块读入内存分类] 从 f 上把第 j 块、 $j+1$ 块顺序读入内存 I 区和 II 区, 第 k 块逆序读入内存 III 区, 并利用堆分类法将 3 块一起分类。

S5. [将内存中 II 区写到文件 f] 将分类后的 II 区顺序写出到 f 的第 $j+1$ 块上。

S6. [对 l 进行循环] 若 $l < k$ 则实施步骤 S7 ~ S9, 否则转 S10。

S7. [把文件 f 的第 l 块读入内存分类] 从 f 上顺序把第 1 块读入内存 II 区, 与 I、II、III 区一起分类 (内存中始终只有 2 块或 3 块)。

S8. [将分类后的内存中 II 区写到文件 f] 将分类后的 II 区顺序写出到 f 的第 1 块上。

S9. [改变循环指针] $l \leftarrow l + 1$

S10. [把内存中 I、III 区写到文件 f] 将分类后的内存 I 区和 III 区分别顺序和逆序写出到文件 f 的第 j 块和第 k 块上。

S11. [执行下一趟分类] $l \leftarrow j+3, j \leftarrow j+1, k \leftarrow k-2, n \leftarrow n-2$ 。

S12. [进行最后一趟分类, 分类结束] 把文件 f 上从第 j 块开始的 n 块顺序读入内存分类, 并把分类后的结果顺序写出到文件 f 上从第 j 块开始的位置上。至此整个算法终止。

该算法已用 C 语言在 PC/586 上得到了实现。
下面是该算法的一个例子 (见表 1—4)。

表 1 f 的初始状态
Tab 1 The f 's state of beginning

块 1	0	56	429	276	886	17	885	603	395	896
2	374	116	624	106	914	221	115	111	768	490
3	722	323	53	96	657	593	541	164	111	286
4	573	776	300	828	643	993	278	191	216	330
5	244	404	820	420	858	632	756	641	494	406
6	536	502	635	853	616	208	429	976	313	378
7	628	614	458	782	421	573	468	881	197	498
8	225	810	727	123	956	414	873	393	619	669

表 2 一趟分类结束, 求出了 f 的最小块和最大块, $n=6$

Tab 2 A trip sorting ends, and extract f 's minimum piece and maximum piece, $n=6$

块 1	0	17	53	56	96	106	111	111	115	116
2	276	374	393	395	414	429	490	603	619	624
3	123	164	221	225	286	323	541	593	657	669
4	191	216	278	300	330	573	643	722	727	768
5	244	404	406	420	494	632	641	756	776	810
6	208	313	378	429	502	536	616	635	820	828
7	197	421	458	468	498	573	614	628	782	853
8	858	873	881	885	886	896	914	956	976	993

表 3 最后一趟分类, 把块 4、5 顺序读入 I、II 区分类

Tab 3 The last trip sorting, piece 4, 5 in sequence readed into the I, II area to classify

I	404	406	414	420	421	429	429	458	468	490
II	494	498	502	536	541	573	573	593	603	614

表 4 分类后的 f 状态
Tab 4 The f 's state after sorting

块 1	0	17	53	56	96	106	111	111	115	116
2	123	164	191	197	208	216	221	225	244	276
3	278	286	300	313	323	330	374	378	393	395
4	404	406	414	120	421	429	429	458	468	490
5	494	498	502	536	541	573	573	593	603	614
6	616	619	624	628	632	635	641	643	657	669
7	722	727	756	768	776	782	810	820	828	853
8	858	873	881	885	886	896	914	956	976	993

表 1 中共有 80 个记录, 分成 8 块, 存于磁盘文件 f 上, 共进行 4 趟就完成了分类。

第 1 趟: 先把第 1、2 块顺序读入内存 I、II 区, 第 8 块逆序读入内存 II 区, 进行堆分类。分类结束后, 把 II 区顺序写出到文件 f 的第 2 块, 然后把第 3 块读入 II 区与 I、III 区一起分类, 并把分类后的 II 区顺序写出到 f 的第 3 块。就这样重复读/写, 直至文件 f 的第 7 块为止。该趟分类结束后, 确定了文件 f 的第 1 块和第 8 块, 共读/写磁盘 8 次 (见表 2)。第 2 趟: 确定了文件 f 的第 2 块和第 7 块, 共读/写磁盘 6 次。

第 3 趟: 确定了文件 f 的第 3 块和第 6 块, 共读/写磁盘 4 次。

第 4 趟: (最后一趟) 由于 f 中只剩下两块未被分类, 所以一次把两块读入内存 I、II 区分类。分类后再把 I、II 区写出到 f 的相应位置, 共读/写磁盘两次 (见表 3、4)。

在该例子中算法 S 共读/写外存 20 次, 而对同样块数和大小的文件 f 进行二路平衡归并需读/写外存 32 次。

可以得出, 该算法不必预先产生初始归并段, 即可得到预期的效果。性能上优于已知的二路平衡归并法和二路多步归并法。

3 算法正确性证明与分析

定义 4 一个文件 f 是按块有序的当且仅当相邻 2 块 B_p 和 B_q ($p < q$) 中的记录有如下关系:

(1) 对于 f 中任一块中的每一个记录关键字均不小于它的紧接前趋的记录关键字。

(2) 块 B_q 中的每一个记录关键字均不小于块 B_p 中的记录关键字。

定义 5 内分类是正确的当且仅当内存中的记录关键字分类后按非递减次序排列。

算法正确性证明

定理 1 算法 S 是正确的。

我们用归纳法来证明之。设文件 f 在外存上被分成 s 块 ($s \geq 2$), 即 $f = (B_1, B_2, \dots, B_s)$ 。

基始: 当 $s = 2$ 时, 依照算法 S 进行分类, 只需执行步骤 S12 即可, 由定义 4 和定义 5 可知, 分类后的 2 块 B_1 和 B_2 按块有序, 所以 $s = 2$ 时算法 S 是正确的。同理 $s = 3$ 时, 算法 S 也是正确的。

归纳: 假设当 $s \leq k$ ($k \geq 2$) 时, 算法 S 是正确的, 即 $f = (B_1, B_2, \dots, B_k)$ 分类后得到新的 $f = (B'_1, B'_2, \dots, B'_n)$ 按块有序, $B'_1 \leq \dots \leq B'_k$, 则当 $s = k + 1$ 时, 分两种情形讨论。若 k 是偶数, 则对 k 块分类, 进行到最后一趟时, 只有两块读入内存, 所

以当对 $k + 1$ 块分类时, 最后一趟只有 3 块读入内存, 而有归纳假设可知前面的几趟分类和最后一趟的 3 块分类都是正确的, 所以此时算法 S 是正确的; 若 k 是奇数, 则对 k 块分类进行到最后一趟时, 只有 3 块读入内存, 则对 $k + 1$ 块分类进行到最后一趟时只有 2 块读入内存, 而由归纳假设当 $s \leq k$ ($k \geq 2$) 时, 算法 S 是正确的, 所以当 $s = k + 1$ (k 是奇数) 时算法 S 也是正确的。

4 算法分析与结论

4.1 存贮开销 算法 S 在进行内分类时, 采用的是堆分类法, 所以只需占用 3 块存贮单元, 由于需要 1 块存贮单元作为读/写缓冲区, 因此算法 S 的存贮开销约为 4 块存贮单元。

4.2 读写/外存次数和时间复杂度 当文件 f 有 k 块时, 进行一趟分类确定 2 块, 则分类完毕后, 读/写外存的次数为

$$k + (k - 2) + \dots + 2 = \frac{(k/2)(k + 2)}{2} = \frac{k^2 + 2k}{4} \quad (k \text{ 为偶数})$$
$$k + (k - 2) + \dots + 3 = \frac{[(k - 1)/2](k + 3)}{2} = \frac{k^2 + 2k - 3}{4} \quad (k \text{ 为奇数})$$

由于外分类时, 算法的时间复杂度主要是由读/写外存的次数来支配的, 因此该算法 S 的时间复杂度为 $O(k^2)$ 。

表 5 两种外分类法比较

Tah 5 Comparison of two kinds of external sorting algorithm

块 k	4	5	6	7	8	9	15	16	17	22	23
算法 S	6	8	12	15	20	24	63	72	80	132	143
二路平衡归并	12	20	24	28	32	45	75	80	102	132	138

由表 5 可以看出, 算法 S 简单, 易实现, 而且不必预先产生初始归并段, 较二路平衡归并有效。其中二路平衡归并的读/写外存次数为 $k + k[\log_2 k]^{[2, 10]}$, 在实施时, 对待分类的文件都是随机存取的, 因此特别适合于磁盘分类。

参考文献:

[1] [美] 可努特 D E. 计算机程序设计技巧[M]. 管纪文, 苏运霖, 译. 北京: 国防工业出版社, 1984.
[2] 严蔚敏, 吴伟民. 数据结构[M]. 北京: 清华大学出版社, 1988.
[3] 王本颜, 方蕴昌. 数据结构技术[M]. 北京: 清华大学出版社, 1988.

[4] [美 洛林 H. 分类方法和分类系统[M] . 王书仁, 张鹤年, 石成孝, 译. 北京: 中国铁道出版社, 1982.

[5] 杨宪泽. 分级快速排序法研究[J] . 科学通报, 1989 (11) : 871— 873.

[6] 杨宪泽. 基于一类特殊问题的排序算法[J] . 计算机工程, 1992, 18(1) : 58— 62.

[7] 杨大顺, 陶明华. 一种新的插入排序和分档检索法[J] . 计算机学报, 1990(11) : 853— 859.

[8] 郭继展. 一次到位排序法[J] . 计算机应用研究, 1988, 5 (3) : 33— 34.

[9] ULLMAN J D. Principles of database and knowledge base system[M] . Computer Science Press, 1989.

[10] HOROWITZ E, SAHNI S. Fundamental of data structures [M] . Computer Science Press, 1976.

[11] 曹奇英. 均匀分布数据的一种排序算法[J] . 计算机应用与软件, 1992, 9(1) : 31— 33.

Research on Shaker External Sorting Algorithm

DONG Yu ping¹, ZHANG Zhi ti², QI De yu

(1. Computer Center, Guangzhou Academy of Fine Arts, Guangzhou 510261, China;

2. Center of Network, Xuchang University, Xuchang 461000, China;

3. College of Computer Science, South China University of Technology, Guangzhou 510640, China)

Abstract: A new external sorting algorithm (ESA) —— Shaker ESA are given. It need not in advance produce original merged segment, can immediately get the anticipant result. under the particular config with data and the hardware, the performance exceed two ways mergesort and two ways many steps mergesort. Its complexity is thoroughly analysed and its correctivity is proved. Shaker ESA. were implemented in C on PC /586 and above computers.

Key words: Shaker ESA; time complexity; memory overheads

(上接第 117 页)

The Multiple Dimension Lindstedt Poincaré Method for Nonlinear Vibration Analysis of Axially Moving Systems

HUANG Jian liang, CHEN Shu hui

(Department of Applied Mechanics and Engineering, Sun Yat sen University, Guangzhou 510275, China)

Abstract: The multiple dimension Lindstedt Poincaré (L—P) method is formulated for nonlinear vibration analysis of axially moving systems. The equations of motion of a axially moving strip are derived through Hamilton’s principle. The Galerkin method is used to discretize the governing equations. The forced response of an axially moving strip with internal resonance is studied and the complex internal resonance curves are obtained. The typical example shows that the multiple dimension L—P method is a good quantitative method for the analysis of axially moving systems.

Key words: nonlinear vibration; axially moving systems; multiple dimension L—P method; fundamental resonance