

提高数据库管理系统运行速度与效率的几种方法

南开大学计算机系 曹勇

摘要: 本文主要是从数据库管理系统的设计和软件编制方面,介绍了几种提高系统运行速度与效率的实用方法,并给出了若干具体实现方法。

微小型计算机上管理信息系统(MIS)的迅速发展,已使数据库管理系统(DBMS)在各行各业中都得到最广泛的应用。随着数据库规模的增大,这就使得对大量信息的管理变得十分困难,用户关心的一个主要问题即是:如何提高 DBMS 的运行效率和速度,这也是评价一个数据库应用系统成败的重要指标之一。虽然,当今计算机硬件技术已有了很大的发展,机器的运行速度可以达到很高(如:386、486 机器)。但是,在硬件环境、系统配置一定的前提下,提高系统的运行效率以及速度,对于 DBMS 的正常运行,特别对提高用户的工作效率有着重要的作用。当一种好的设计方法被采用时,往往能将 DBMS 的运行速度提高几倍,甚至几十倍,这将产生难以估量的社会、经济效益。因此,在系统设计、软件的编制上尽可能采纳优良的方法是十分必要的。

通过对以往实际工作中取得的经验进行总结,我们探索到一些提高 DBMS 运行效率与速度的方法。这些方法可以有效地使应用系统提高运行效率与速度,下面将着重从系统的分析与设计、软件设计技术等方面对如何提高 DBMS 的运行效率、速度进行阐述。

一、系统的分析与设计

数据库是进行数据管理的基础,各种操作都是围绕它进行的。因此,数据库设计的好坏,是能否得到较高的运行效率与速度的先决条件。数据库所包含的字段数、数据字段长度、存储结构的规范化以及是否采用代码等问题是影响系统运行效率与速度的重要因素。

1.在设计数据库时尽量压缩数据字段的长度,以减少数据冗余。

2.尽量不使数据库中的字段数过多

除非十分必要,应尽可能避免设计出包含八十个以上字段的大库。

3.遵循存储结构规范化理论的要求

1 NF(First Normal Form)、2 NF(Second Normal Form)、3 NF(Third Normal Form)对数据库中数据进行处理(见参考文献 2)。在一般情况下,经处理过的数据库不但能减少存储容量,而且会对提高运行效率大有益处。它尽可能避免了数据冗余;各数据字段间具有简明的逻辑关系;数据库存储结构简单,而且数据库规模相对较小。从而,数据能在较短的时间内被读出,响应操作要求。

4.数据代码化

所谓“代码化”,是指将原先以字符方式存储的数据,在机器中尽量以代码,即数字方式来表示。这在大多数情况下,可以减少数据的存储量;而且,对此种字段进行操作时速度也比较快。

那么怎样的数据可以进行代码化呢?首先,这种数据可能出现的内容数目应该是相对有限的,所谓“相对有限”即是指,数据项内容数目在一定时期内不会有较大的增删,可以满足正常的工作需要;并且,在系统设计之初,还应该能估计出其数目的大致范围,能满足以上条件的数据,可以说它具备了代码化的条件,可以进行代码化处理。例如,我们在为某公司开发网络 MIS 时,对“商品名”字段进行了分析,根据该公司现经营商品的情况得知,商品的数量将在一较长的时间内不会有大的变动与大幅度的增加;并且已知商品名 55 个,可以满足现在的日常工作需要;最后,我们根据该公司专业人员的预计,将商品名代码的范围定为 1~200,对商品名字段进行

代码化,用三位数(001~200)在机器内表示二百个不同的商品名,为了能在屏幕、报表上显示出商品的汉字名以方便于阅读,设计了一个代码库,每个商品代码在此库中对应一条记录,即代码的中文含义,这样,减少了数据的存储量,与此类似,对许多具有这种性质的字段进行了代码化处理,取得很好的效果。另外,值得一提的是,通过对部分字段进行代码化处理,还简便了此类字段的录入工作,取得了异想不到的效果。比如用户录入商品名时,可以设计将商品代码连同其中文含义一同显示于屏幕的某区域上,用户可以使用光标键、回车键或直接敲入其代码的方式输入。这种输入方式既简单、直观,又减少了击键次数,提高了录入的效率和速度,在实际应用中受到用户的普遍欢迎。据用户反映,在对数据进行代码化处理之后,每天录入的数据量是以前的5~10倍,而且现在数据处理起来的速度也比以前快多了。因此,代码化技术对于提高整个系统的运行效率与速度,将用户从繁重的录入工作中解放出来都起到了重要的作用。

总之,在系统分析与设计之初,就将提高系统的运行效率和速度作为一项重要的问题来考虑,尽可能设计出简单、高效的数据库。这对于从整体上提高DBMS的运行效率和速度有着至关重要的作用,并且也为以后从软件设计上提高系统的各项指标打下基础。

二、软件设计技术

软件的设计过程就是构造和描绘算法的过程。算法设计得好坏,将会以数量级的差别反映到运行系统的效率与速度上。由此,算法的分析和确定将是软件设计的关键。同时,还应对DBMS所用到的命令进行分析,权衡各相关命令的时间、空间开销与应用效果。

下面就几种重要的DBMS软件的设计方法加以讨论。

1.合理、有效地使用索引

数据库索引技术被广泛地应用于DBMS之中,建立索引可以减少系统对磁盘的I/O次数,同时它也是有效地加快系统的查询、统计速度的重要手段之一。但是,使用索引技术也常常带来一些副作用。例如,查询、统计的内容过多,条件组合的形式过多时,需采用多个索

引文件,且要多次打开、关闭它们,这是很费时间的,对大型数据库进行一次索引有时需要花费几个小时,索引文件在使用时往往还较呆板,通常只能是在系统开发时对某些数据库提前作好几个索引文件,并且它们都只能在固定位置使用,而如果用户要求随机对某数据库的任意个字段进行索引时,这在设计好了的应用系统中就无法实现了。针对以上在使用索引中遇到的问题,我们摸索出一种使用固定索引与随机索引相结合的方法。具体方法是:首先建立一个索引库,此库中的每条记录都对应一个索引文件的信息,系统中每个被用到的数据库与索引库中的一段记录区域相对应。其库结构如下:

	字段名	类型	长度	说明
1	I name	Character	10	索引文件名
2	Key s1	Character	80	用于显示的关键式串
3	Key s2	Character	80	用于建立索引的关键式串

对于应用系统中必定使用的固定关键式索引文件,由程序设计人员在系统外部建立,并由应用系统负责维护,保证索引文件与相应库中的数据一致。而索引库中的记录是用户在应用系统中任意建立的,并由用户主动进行维护为此我们用交互的方式让用户从屏幕上对某数据库的字段进行挑选。然后系统负责自动建立并命名该索引文件,将其信息填入索引库中,用户在需要查询、统计之前,可以用菜单的形式选择打开索引文件。日常维护这些随机索引文件的工作(重建、主动删除等),则交给用户完成,系统不与负责。为了不致使索引库的记录数过多而造成索引文件泛滥,在具体实现时,将每个数据库的随机索引文件数控制在40条以内。对这些索引文件采用最久未用算法(LRU, Lease Recently Used)进行管理。在一数据库的所有随机索引文件数超过40条时,将最长时间未使用的索引文件及其在索引库中的记录删除。每次使用或建立随机索引文件时,都要对索引库中的记录按LRU算法排序。随机索引文件的建立过程见以下流程图:

采用了以上技术之后,一方面,使用户可以更有效地使用索引文件,使DBMS的运行速度得以提高;另一方面,又使系统避免了对大量索引文件进行维护的时间开销。由此,索引技术能更有效地服务于DBMS。

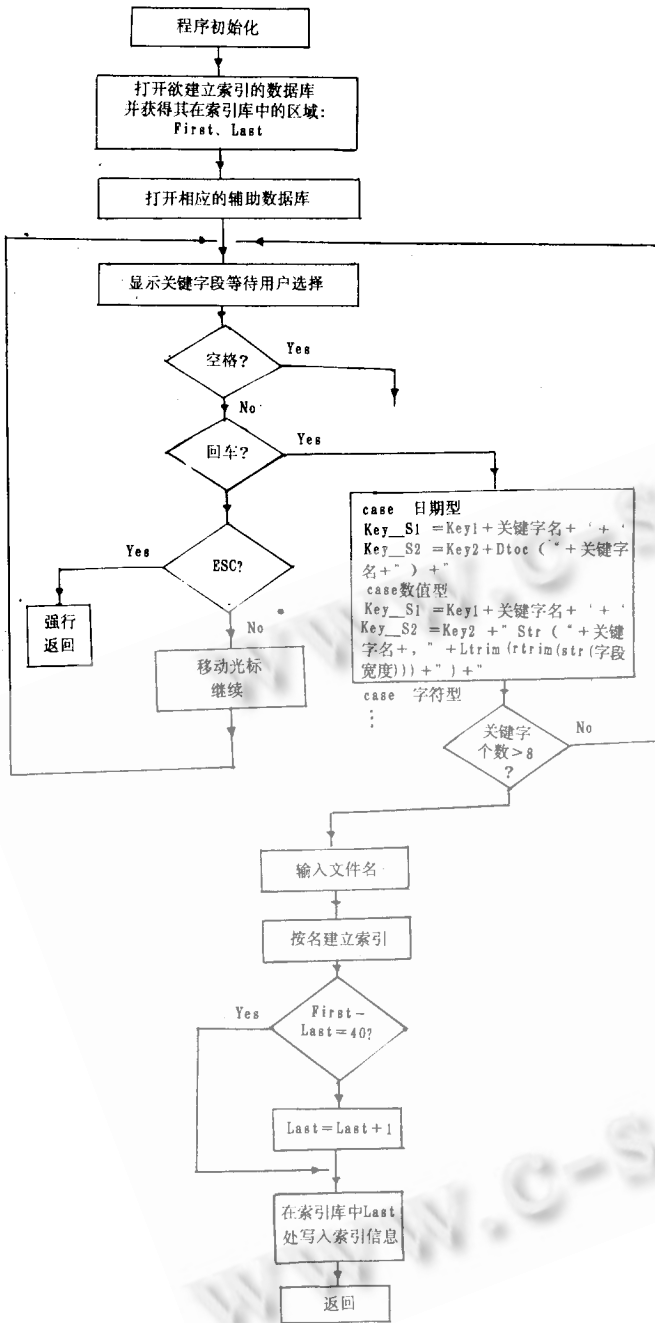


图 随机索引文件建立过程流程图

2.对文件进行归并,减少开/闭文件次数

一个完整的 DBMS 通常包含有众多的数据库与程序文件等。如果按常规的方法将它们以独立的文件方式分别存储,会造成系统在运行中频繁地打开/关闭文件

(包括数据文件、程序文件、索引文件等),而付出很大的时间开销;另外,这也不便于系统进行有效的管理。尽量减少开、关文件的数目与提高一次打开文件的效率是许多编程者一贯追求的目标。以往我们在编程中主要针对程序文件采取办法:尽量减少程序文件的数目,诸如:将多个子程序放在同一个过程文件中,一次将其全部打开;在 FoxBASE+中还可以用 FOXBIND 将多个过程组合成一个大过程等。

而实际工作中我们经常忽略数据库文件的效率问题。打开/关闭数据库文件是系统运行中频繁进行的工作,一个完整的 MIS 常需要对几十甚至上百个数据库文件进行操作。如果能把此种操作的效率提高,必将对 DBMS 的响应速度有很大的改善。在实际工作中我们发展,大量的 DBMS 数据库文件中,由于各种需要而建立的辅助管理和其它操作的数据库文件占绝大部分,比如,前边提到的索引库、代码库、结构库以及条件库(用于建立随机统计、查询条件)等。这些众多的数据库往往是一经系统投入运行就不再或很少变化的。同时还发现 <GO 记录号> 语句的速度比打开一个文件的速度快得多。由此,我们摸索出一条提高系统效率的方法,即同类数据库归并法。将同类型的辅助数据库合并到同一个数据文件之中,在使用时将其一次打开,然后根据预先标好的信息,用 GO 语句直接转到相应的位置,进行操作。例如,在对多个字段进行代码化之后,将与这些字段相对应的代码库合并,并把此库的不同记录区域分配给不同的代码化字段使用,为了能在使用时找到相应的区域,我们还在与含有此代码化字段的数据库所对应的结构库中增加了 c_head、c_len 字段,c_head 标识该字段是否为代码化的,等于-1 为百代码化字段,大于等于 1 为代码化字段,并以 c_head 此时的值指示其代码含义的记录在总代码库中的起始记录号;该字段如果是代码化的,那么 c_len 指出其现有的代码个数。用 c_head、c_lene 即可以在总代码库中标识某字段的代码区域。当然,考虑到代码有增加的可能,在各代码区域之间都适当留有空记录,以便代码的扩充。与此相似,可以将大部分的辅助数据库进行同类归并。这样,既提高了 DBMS 的运行效率,又大大地节省了时间,提高了系统的响应速度。在实际应用中,我们广泛采纳了这种技术,取得了令人鼓舞的效果。

3. 筛选与中间库相结合技术

所谓“筛选与中间库相结合”是指在对数据库作二维条件(即每个大类条件中还包含若干子条件)统计时,先将满足大类条件的所有记录从原数据库中挑选出来,拷贝到一中间库中;然后,用 set filter to <子条件> 语句对中间库作筛选处理,每次筛选后统计一次。对一个较大规模的数据库进行多次二维条件统计时,采用筛选与中间库相结合的方法可以比普通的对原库直接操作快几十倍甚至上百倍。现举例加以说明。设某数据库有记录 20000 条,要对满足 40X10 的二维条件进行统计,即大类条件 40 条,每类条件中有 10 条子条件,并设有 Num 条记录满足大类条件 Num 取不同值时,四种统计方式

的记录指针移动次数进行比较:

(1)直接对原库进行统计:

计算公式 $20000 \times 10 \times 40$;

(2)仅使用中间库技术:

计算公式 $20000 + 10 \times 40 \times \text{Num}$;

(3)对原库直接进行筛选:

计算公式 $20000 \times 10 + 40 \times (N_1 + N_2 + \dots + N_{10})$, N_1, N_2, \dots, N_{10} 为每大类中分别满足子条件的记录数。 $N_1 + N_2 + \dots + N_{10} = \text{Num}$;

(4)筛选与中间库相结合:

计算公式 $20000 + 10 \times \text{Num} + 40 \times \text{Num}$;

Table 1: 四种统计技术的比较

: A = 8,000,000

Num	B	A: B	C	A: C	D	A: D
1	20,400	1:392	20,040	1:399	20,050	1:399
1000	420,000	1:19	240,000	1:33	70,000	1:114
5000	2,020,000	1:4	400,000	1:20	270,000	1:30
10000	4,020,000	1:2	600,000	1:13	520,000	1:15
15000	6,020,000	1:1.3	800,000	1:10	770,000	1:10.4
20000	8,020,000	1:1	1,000,000	1:8	1,020,000	1:8

由上表可见,采用筛选与中间库相结合技术在对较大规模数据库进行多维条件统计时尤为适用,它对于大幅度地提高系统的运行速度有很显著的作用。

4. 有利于提高 DBMS 运行效率和速度的几个技巧

关于解决 DBMS 运行效率和速度问题的编程技巧有很多,方法也各不相同,只要深入分析所使用的数据库命令所要求的空间、时间开销是不难开发出好的编程方法的。这里简要介绍几种编程技巧。

(1)提高联结数据库的速度。我们有时需要将两个数据库(设分别有 A 条, B 条)联结成为第三个数据库,此数据库中应包含有 A、B 中的所有字段。若使用 join 语句,则记录指针作 $O(A \times B)$ 次操作,运行速度很慢,如改为先对两数据库按某相同字段排序,然后建立一个含有这两个数据库中所有字段的数据库,用 APPEND 语句将第一个数据库中的记录追加到此数据库中,最后用 update 替换第三个数据库中剩余的字段。这样,操作的次数是 $O(A+B)$,可以在一定程度上提高 DBMS 的运行速度。

(2)尽量使用数组,少用宏代换;尽量使用 find、seek 查找,少用 locate。

(3)将多次使用到的计算结果放在中间变量中,以免多次使用时重复计算。

(4)减少子程序嵌套,应尽可能将它们放在同一个程序中。

(5)及时释放使用过的变量,尤其是较大的数组。

(6)伪提高速度法。如果预计到系统在某项操作时,将运行较长时间,可以通过将系统运行完毕后所应在屏幕上显示的内容合理地分成几部分,分别在系统运行中逐次显示。或者采用比例色条,动态显示完成比例情况等手段给用户一种动态的视觉效果。虽然实际上运行时间并未减少,有时还略有增加,但用户却避免了面对凝滞画面时的难堪。从而觉得运行时间并不很长。

从上面的讨论可以看到,提高 DBMS 的运行效率与速度,即需要在系统的设计之初,就将提高系统的运行效率与速度作为一项重要的指标来考虑,设计出精巧的

(下转第 12 页)

(上接第5页)

系统;同时,还需要在软件的设计时,引进优良的算法和技术,巧妙地使用各种命令;最后,在系统环境、状态的配置上,还应该多方考虑,选择合理、正确的参数配置。实践证明,采取以上方法可以有效地提高 DBMS 的运行效率与速度。

本文在形成的过程中,得到了南开大学计算机与系统科学系吴功宜副教授的热心帮助与指导,在此表示由衷的感谢。

参考文献:

1.王成孝,“提高数据库应用系统运行速度的一些措施”,计算机世界月刊,1989,11。

2.王勇领,“系统分析与设计”,清华大学出版社,1991,10

3.唐建宇,“如何提高 DBMS 复合条件统计报表的运行速度”,新浪潮,1991,4

4.姜玉振,陶应全,李玉柱,“提高 FoxBASE+应用运行效率的一些途径”,计算机世界月刊,1991,7

5.张苏新,“ORACLE 索引的合理使用”,计算机世界月刊,1990,7

6.王云飞,沈雷鸣,潘南明,王德建,“新颖关系数据库管理系统——中西文 Fx BASE+”,同济大学出版社,1989,12