An Open Hash Method Using Predictors

Seiichi Nishihara* and Hiroshi Hagiwara**

Abstract

In the scatter storage technique, many methods of resolving collisions have been proposed. Those are classified into two main methods, i.e. the open hash method and the chaining method. A measure of efficiency for a table search is the average number E of probes necessary to retrieve a key in the table. The average number E of the open hash method is always greater than that of the chaining method.

In this paper, it is shown that the predictor method, which uses a several bit field reserved in each cell and is applicable to the open hash method, significantly reduces the average probe number E. The efficiency of the predictor method is estimated theoretically and verified experimentally. A comparison with the chaining method is also made with respect to the memory usage.

1. Introduction

Hash addressing has been found to be usually an efficient way to reduce the number of probes required to enter or retrieve a key in a table. Especially it is remarkable that the average number of probes depends just on the fraction α of the table that is occupied but not on the total number of keys. The fundamental idea of hash addressing is the usage of key to determine the address of the cell in a table in which the desired information is stored. It is therefore important to choose a good hash function that maps keys to addresses as uniformly as possible.

Since any key-to-address transformation generally corresponds to a many-to-one mapping, it will probably happen that more than one distinct keys have the same address. Such an occurrence is called a collision. Many techniques for resolving collisions have been proposed[1-6]. They are classified mainly into two methods: the open hash method and the chaining method. Furthermore, open hash techniques are divided into two classes according to whether or not they eliminate secondary clustering[3], which occurs when different keys hashed initially to the same location proceed to trace through the same sequence of locations.

Assuming equal usage of cells, the theoretical approximation of the average number E of probes necessary to retrieve a key has been given for each method: e.g.

$1+\alpha/2$ (chaining method),

$-(1/\alpha)\log(1-\alpha)$ (open hash method eliminating primary and secondary clusterings),

where α is the load factor of the table.

It is known that the average number of probes needed in the open hash method is

This paper first appeared in Japanese in Joho—Shori (Journal of the Information Processing Society of Japan), Vol. 15, No. 7 (1974), pp. 510~515.

* Institute of Electronics and Information Science, University of Tsukuba
** Department of Information Science, Faculty of Engineering, Kyoto University

greater than that needed in the chaining method. In this paper, however, it is shown that the predictor method, which is applicable to the open hash method, significantly reduces the average number E of probes. First the new method is introduced, and then the efficiency of the predictor method is estimated theoretically and verified experimentally. Finally a comparison with the chaining method is also made with respect to the memory usage.

2. The Predictor Method

2.1 Definition of Terms

Before describing the predictor method, we shall define the terms necessary for the algorithm. A hash table of size M is a set of M successive cells, N of which are occupied($N \leqq M$). The load factor α is defined as N/M. Each cell includes a key field. The search operation is performed on the table by using a series of functions h_i, i= 0,1,2,..., where $0 \leqq h_i(K) \leqq M-1$ for any i and key K. The first address $h_0(K)$ is called the hash address of K. Synonyms are the keys that are transformed to the same hash address. An algorithm for the open hash method takes the following form:

Step 1. Set $a=h_0(K)$, i=0;
Step 2. If the a-th cell is empty or contains K, then the search is concluded;
Step 3. Otherwise, set i=i+1, $a=h_i(K)$ and repeat step 2.

2.2 The Method Using Predictors

In this method, each cell contains not only a key field but also a j-bit field as a predictor. We consider just the open hashing in which the synonyms always produce a clustering, i.e. the secondary clustering may occur. The predictor is used for the purpose of searching only synonyms, i.e. keys in the same cluster.

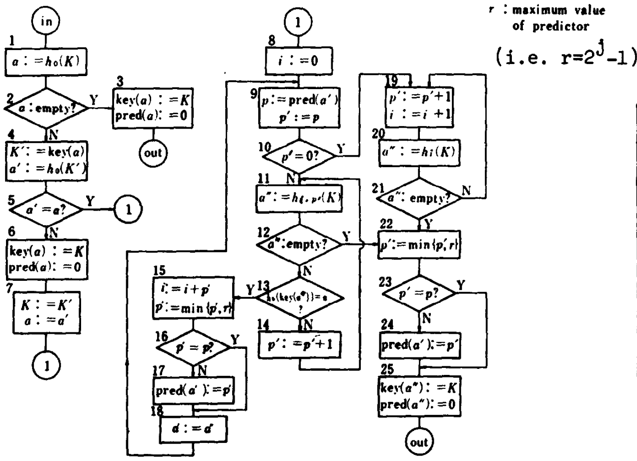
Assume that the search for key K is now at the address $h_i(K)$, i.e. none of the addresses $h_0(K),\dots,h_i(K)$ contains the key K. In the usual open hashing, the next search address is $h_{i+1}(K)$. However, in the case where the key in the $h_{i+1}(K)$-th location is not a synonym of K, there is no purpose in checking that location. In such cases, the value p of the predictor is used to tell the number of probes needed until an address containing a synonym is encountered, where $0 \leqq p \leqq 2^j-1$. In other words, another synonym is found in the $h_{i+p}(K)$-th location.

Especially, the predictor of the last cell of a cluster is always kept zero. It means that no more synonyms exist in the table, which is effective to reduce the reject time[5]. It may also happen that more probes than 2^j-1 are needed to find another synonym. In that case, after checking the $h_{i+2^j-1}(K)$-th location, we must repeat probing operations one by one using the series of functions h_i. This phenomenon is the only reason that still makes the average number E of probes greater than that of the direct chaining method(i.e. $1+\alpha/2$). The additional cost of this phenomenon is estimated in Section 3.

Here we give algorithms to enter or retrieve a key K. In the following, key(a) and pred(a) denote the key and the value of the predictor in the a-th location.

The Entering Algorithm (Figure 1)

The entering algorithm contains a moving operation of a key that has already been entered, as the chaining method does. The algorithm, given in Figure 1, consists of three main parts, i.e. steps 1-7, steps 8-18 and steps 19-25. First, the effect of steps 1-7 is to check the hash address to examine if a collision happens or if a key moving operation(steps 6 and 7) is necessary. Next, the operations of steps 8-18 enable to trace through the cells of a cluster until the last cell is encountered, and furthermore update(step 17) the predictors which have been disturbed by the key moving operation. Finally, the entering operation is executed by the operations of steps 19-25, in which steps 19-21 form a loop to find an empty cell.



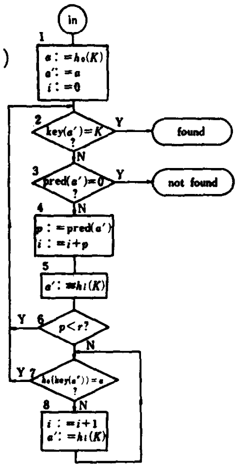


Fig.1 The algorithm to enter a key K.

Fig.2 The algorithm to retrieve a key K.

The Retrieving Algorithm (Figure 2)

The retrieving algorithm, given in Figure 2, is far simpler compared with the entering algorithm. This algorithm works correctly even if the key is not in the table. Actually, the absence of a key is proved in step 3. Probing occurs in step 2. In step 4 and 5, the predictor is used to calculate the next probing address. Execution of step 7 is needed only if p<r is not true(i.e. p=r) in step 6, where $r=2^j-1$. However, if the length of the predictor field is chosen to be more than 4 or 5 bits, such cases may rarely occur.

3. Efficiency of the Method

Let j and x be the bit length of the predictor field and the load factor respectively. Then the maximum value r of a predictor is 2^j-1. Assume that each cell in the table is hit as frequently as any other. Then, using the Poisson approximation, we can estimate that the probability P(i,x) of a cluster of length i is $e^{-x} \cdot x^i / i!$.

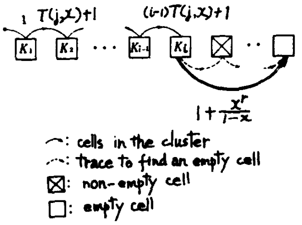


Fig. 3 Storing process when loading factor is x.

Figure 3 shows an entering process of key K, when the length of the cluster is i, i.e. all the hash addresses of keys $K_1, \ldots, K_i$ and K are the same. Let S(j,x) denote the average number of probes needed to retrieve the key which have been entered when the load factor is x. Then S(j,x) is given as the sum of the cost C_f of scanning the cluster and the cost C_e of finding an empty cell. We do not consider the effect of key moving operations.

First we estimate the cost C_e. Starting from the last cell of a cluster, the probability that just k probes are needed to find an empty cell is $x^{k-1} \cdot (1-x)$. While the number k does not exceed the maximum value r, the number of probes needed to access the same key is reduced to one by using the predictor. But if k>r, then the number of probes in case of accessing becomes 1+k-r. Therefore, the cost C_e is estimated as

$$\sum_{k=0}^{r} x^k(1-x) + \sum_{k=r+1}^{\infty}(1+k-r)x^k(1-x) \qquad \left(= 1 + \frac{x^r}{1-x}\right). \tag{1}$$

Next, let T(j,x) be the average number of probes between two cells adjoining each other in a cluster when the load factor is x. It always holds that T(j,x)>1. Then the cost C_f is given as

$$\sum_{i=1}^{\infty}((i-1)T(j,x)+1) \cdot P(i,x). \tag{2}$$

Therefore, from the results (1) and (2) it follows that

$$S(j,x) = 1 + \frac{x^r}{1-x} + \sum_{i=1}^{\infty}((i-1)T(j,x)+1) \cdot P(i,x).$$

Let E(j,α) denote the average number of probes needed to retrieve a key in a table when the load factor is α. Then E(j,α) is given by integrating and averaging S(j,x) as

$$E(j,\alpha) = \frac{1}{\alpha}\int_0^{\alpha} S(j,x)dx . \tag{3}$$

Now to get an approximation assume T(j,x)=1. Then equation (3) is rewritten as

$$\begin{aligned} E(j,\alpha) &= \frac{1}{\alpha}\int_0^{\alpha}\left[1 + \frac{x^r}{1-x} + \sum_{i=1}^{\infty} i \cdot P(i,x)\right]dx \\ &= 1 + \frac{\alpha}{2} - \frac{\log(1-\alpha)}{\alpha} - \sum_{i=1}^{r}\frac{\alpha^{i-1}}{i} , \end{aligned} \tag{4}$$

where $r=2^j-1$. Figure 4 shows the average number E of probes necessary to retrieve a key for our method(i.e. E(j,α)), the quadratic search method, and the direct chaining method.

4. Experimental Verification

Applying our method to the quadratic search method of Hopgood and Davenport[4], we repeated a set of experiments 40 times. The results achieved for a table of length 2048 using pseudorandom keys are compared with the theoretical values i.e. E(j,α) in Table 1. It is seen that the experiments give results very close to the expected values.

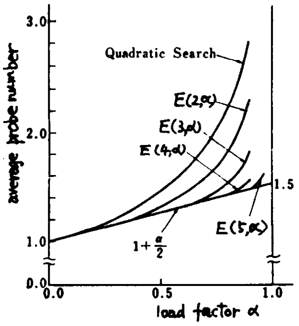


Fig. 4 Average number of probes

Table 1 Theoretical values $E(j, \alpha)$ and experimental values of the average probe number,

α	E(2,α)	observed E j=2	E(3,α)	observed E j=3	E(4,α)	observed E j=4	E(5,α)	observed E j=5
0.1	1.050	1.043	1.050	1.043	1.050	1.043	1.050	1.042
0.2	1.102	1.102	1.100	1.101	1.100	1.101	1.100	1.101
0.3	1.159	1.156	1.150	1.151	1.150	1.151	1.150	1.151
0.4	1.224	1.221	1.200	1.205	1.200	1.204	1.200	1.201
0.5	1.303	1.293	1.252	1.254	1.250	1.252	1.250	1.251
0.6	1.407	1.393	1.308	1.312	1.300	1.302	1.300	1.302
0.7	1.557	1.546	1.378	1.386	1.351	1.352	1.350	1.350
0.8	1.796	1.796	1.496	1.511	1.409	1.410	1.400	1.402
0.9	2.288	2.318	1.801	1.839	1.541	1.556	1.460	1.462

5. Comparison with the Chaining Method

The greater the bit length j of the predictor field is chosen, the closer the value of $E(j,\alpha)$ becomes to $1+\alpha/2$. In the chaining method, the length of a pointer field must be at least $\log_2 M$ bits, where M is the table size. In general, the size of a cell of the predictor method is less than that of the chaining method.

Let q and M be the key field length and the table size in the chaining method respectively. Then the total memory for the table is $M(\log_2 M+q)$. Now assume that the same number of bits are used for the table of the predictor method, then the available table size M' is given by

$$M' = M(\log_2 M+q)/(j+q) \quad > M \; , \qquad (5)$$

where j is the bit length of the predictor field.

Let $f(j,\alpha)$ be the load factor which satisfies the following

$$E(j,f(j,\alpha)) = 1+\alpha/2 \; .$$

Since $E(j,\alpha)$ is always greater than $1+\alpha/2$, it holds that $f(j,\alpha)<\alpha$. Then, with respect to the memory usage, the condition for the average number of probes of the predictor method to be less than that of the chaining method is given by

$$M'\cdot f(j,\alpha) > M\cdot\alpha \; .$$

By using equation (5), this condition can be rewritten as

$$\frac{f(j,\alpha)}{f} > \frac{j+q}{\log_2 M+q} \; .$$

In Figure 5, the two methods are compared for various values of j and M when α is 0.8. It is seen that if the size of the predictor field is chosen to be more than 4 or 5 bits, the predictor method is always preferable to the other.

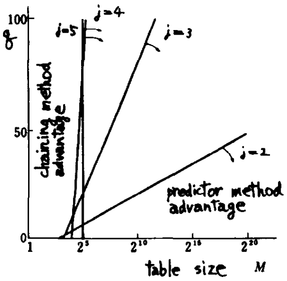


Fig. 5 Comparison of the predictor method and the direct chaining method when α is 0.8

6. Conclusion

We have proposed a method to reduce the average number of probes necessary to retrieve a key in a hash table.

The present method can be combined together with Brent's idea[6]. We have made some experiments of this combination and got good results, e.g. $E(j,\alpha)=1.505$ where $j=5$ and $\alpha=0.99$.

References

1) Johnson,L.R. An indirect chaining method for addressing on secondary keys. Comm. ACM,Vol.4,No.5(1961),pp.218-222.

2) Morris,R. Scatter storage techniques. Comm.ACM,Vol.11,No.1(1968),pp.38-44.

3) Bell,J.R. The quadratic quotient method: a hash code eliminating secondary clustering. Comm.ACM,Vol.13,No.2(1970),pp.107-109.

4) Hopgood,F.R.A. and Davenport,J. The quadratic hash method when the table size is a power of 2. Computer Journal,Vol.15,No.4(1972),pp.314-315.

5) Furukawa,K. Hash addressing with conflict flag. J. Information Processing Society of Japan,Vol.13,No.8(1972),pp.533-539.

6) Brent,R.P. Reducing the retrieval time of scatter storage techniques. Comm.ACM, Vol.16,No.2(1973),pp.105-109.