Formulas are derived for the average number of record references required to retrieve a record from a file (a) in case the records are loaded without regard for relative frequency of reference and (b) in case each set of records with a common home address is arranged in order of decreasing frequency of reference.

The formulas are first derived under the assumption that the mapping from keys to addresses is "random." Finally, an informal argument is given which suggests the formulas will also hold under a familiar "pseudo-random" mapping based on the use of division, provided the keys have a certain property commonly encountered in practice.

Note on random addressing techniques

by W. P. Heising

The organization of a file in random access storage for efficient retrieval of records is an important and recurring problem in systems design. The usual form of the retrieval problem is: to find a particular record, given the content of one predetermined field (the *key*) of that record.

From this point of view, the most satisfactory method of file organization is to make the key and machine address of each record identical (e.g., account number equals address). However, since the method of key assignment is normally outside of the control of the systems engineer, such organization is often not possible.[1]

In this paper, some computations based on the postulated use of a randomly selected addressing function are reviewed and the effect of utilizing knowledge of relative transaction frequency such as contained in rules of the familiar "80–20" type is shown. A particular method of defining "pseudo-random" functions—namely, the use of division with the divisor relatively prime to the radix of the number system in which the keys are expressed—is also reviewed. Finally, an argument is advanced which makes plausible the expectation that addressing functions defined in the latter manner will have statistical behavior comparable to the randomly selected function considered in the first part of the note.

random mapping of keys

When the key assignment method is not a design parameter, it is usually necessary to find an *addressing function*—a method of mapping a sparse set of keys into a dense set of addresses.

Given R records to map into M possible addresses, there are M^R different addressing functions of which only $M!/(M - R)!$ map no more than one record into each address. The likelihood that a randomly chosen function will have this desirable property is therefore $M!/(M^R(M - R)!)$. To illustrate, if $R = 4000$ and $M = 5000$ then the likelihood becomes 10^{-12000}, which is sufficiently small to eliminate trial and error methods for finding an ideal mapping function.[2] It should be noted that the alternative of using an index table (cross reference file) does not circumvent the problem—the same problem is present with the associated index file lookup.[3]

Suppose then that a mapping, G, is selected at random from the set of all functions which map a file of R records into M addresses; thus each record will have a probability of $1/M$ of being assigned to any address. Then $p(n)$, the probability that a given address will correspond to exactly n records, is given by the binomial distribution:

$$p(n) = \frac{R!}{n!(R - n)!}\left(\frac{1}{M}\right)^n\left(1 - \frac{1}{M}\right)^{R-n}. \tag{1}$$

If $R \gg 1$, $M \gg 1$ and R/M is small, $p(n)$ approaches the Poisson distribution, $P(n)$:

$$p(n) \to P(n) = e^{-f}f^n/n! \qquad \text{(where } f = R/M\text{)}. \tag{2}$$

When several records map to the same address, those records in excess of the capacity of the addressed area (assumed here to be capable of holding one record) are sometimes chained to a separate overflow area (or alternatively back into interstices of the main area). If j records have the same "home address" the total number of record references to retrieve each of the j records once is $1 + 2 + \cdots + j = j(j + 1)/2$. Thus an expression for S, the expected number of references to retrieve every record at a given address, may be found as follows. We have,

$$S = \sum_{j=0}^{\infty} [j(j + 1)/2]P(j) = \sum_{j=0}^{\infty} [j(j + 1)/2]e^{-f}f^j/j!$$

and, if the relation $\sum_{j=0}^{\infty} f^j/j! = e^f$ is used in conjunction with the latter, simplification will yield

$$S = f(1 + f/2). \tag{3}$$

Since there is an average of f records per address, s, the average number of references to find a "typical record," may be written as

$$s = S/f = 1 + f/2. \tag{4}$$

The latter formula is meaningful even if $R/M = f > 1$. This indicates a large overflow area. At the "nominal capacity" of 100% ($R = M$), the expected number of references per retrieval ($s = 1 + 1/2$) is 1.5.

Continuing to use the mapping G, a reduction in the average search time can be obtained by utilizing knowledge of the relative

relative transaction frequency

transaction frequency such as contained in the familiar "80–20" rule of thumb that holds approximately for many commercial files. This rule states that 80 percent of the file transactions deal with the 20 percent most frequently used records in the file. Furthermore this rule also applies to the 20 percent of the file—i.e., that 64 percent of the transactions deal with 4 percent of the file, and so forth.

Let $\phi(n)$ be the relative frequency of reference to nth most frequently used record and assume the choice $\phi(1) = 1$ has been made. Then $F(N) = \sum_{n=1}^{N} \phi(n)$ gives the relative frequency of file references to the N most frequently used records.

Now the general form of the 80–20 rule can be expressed in terms of an equation involving F:

$$F(\alpha N)/F(N) = \beta, \tag{5}$$

where $\alpha = 5$ and $\beta = 5/4$ for the 80–20 case.

A continuous solution of equation (5) is given by

$$F(N) = N^{\gamma} \qquad (\text{where} \quad \gamma = \log \beta/\log \alpha). \tag{6}$$

To derive (6) as a solution of (5) we may proceed recursively starting from (5) to obtain $F(\alpha N) = \beta F(N)$, $F(\alpha^2 N) = \beta^2 F(N)$, $\cdots$ and in general, $F(\alpha^i N) = \beta^i F(N)$. Since the latter expression holds, in particular, for $N = 1$ and since $F(1) = \phi(1)$, we have

$$F(\alpha^i) = \beta^i. \tag{7}$$

One solution of (5) may be obtained by assuming that (7) holds for all real $j > 0$. Then by selecting $j = \log N/\log \alpha$ and observing consequently that $N = \alpha^i$, we obtain from (7):

$$\begin{aligned} F(N) &= \beta^{\log N/\log \alpha} = (e^{\log \beta})^{\log N/\log \alpha} \\ &= (e^{\log N})^{\log \beta/\log \alpha} = N^{\log \beta/\log \alpha} \\ &= N^{\gamma} \end{aligned}$$

as required.

If it is assumed that each set of records with a common home address is loaded in order of decreasing frequency of reference, the following expression for s, the average number of references to retrieve an individual record, may be derived:

$$s = 1 + f\gamma/(1 + \gamma) \qquad (\text{for large} \quad M, R). \tag{8}$$

To obtain (8), note that for the first k records loaded, the expected number of references required to retrieve each record once is MS where S is evaluated from (3) by replacing f by k/M. We have:

$$MS = M[k/M + k^2/(2M^2)] = k + k^2/(2M).$$

Therefore, $\delta(k)$, the expected number of references required for each retrieval of the kth item loaded may be found by forming the difference between MS evaluated with $f = k/M$ and $f = (k - 1)/M$, respectively. Thus:

$$\delta(k) = 1 + k/M - 1/(2M)$$
$$\cong 1 + k/M \qquad \text{(for large } M\text{)}. \tag{9}$$

Using the definition of ϕ, we may write

$$\phi(k) = F(k) - F(k-1). \tag{10}$$

Since the file is loaded in order of decreasing frequency of reference, the weighted average of the number of references per record is:

$$s(R/M) = \left(\sum_{k=1}^{R} \delta(k)\phi(k)\right) \Big/ \left(\sum_{k=1}^{R} \phi(k)\right),$$

which after elimination of δ and ϕ, with use of (9) and (10) becomes,

$$s(f) = 1 + f - \frac{1}{MR^{\gamma}} \sum_{k=1}^{R-1} k^{\gamma},$$

and since

$$\sum_{k=1}^{R-1} k^{\gamma} \cong \int_{1}^{R} k^{\gamma}\, dk \cong \frac{R^{\gamma+1}}{\gamma+1},$$

equation (8) results.

For $\gamma = 1$ (all records referred to with equal frequency) equation (8) reduces, in agreement with earlier computation, to equation (4).

To illustrate, if the 80–20 rule is applicable and file arranged by decreasing frequency of reference, equation (8) with $f = 1$ and $\gamma = \log(5/4)/\log 5$ indicates $s = 1.12$ for a full file as compared to 1.5 obtained from equation (4) for a full file with equal frequency of reference.

mapping defined using division

If we use the familiar "division" technique to obtain a mapping H of keys into addresses, H will satisfy the following definition:

Definition of H. Assume the set of M addresses is numbered consecutively from 0 to $M - 1$ and that M (adjusted slightly if necessary) is relatively prime to r, the radix of the number system in which the R keys are expressed. Then for each key x, $H(x)$ is defined: $H(x) = x \pmod{M}$.

If the following assumption is made about key set, an informal argument can be given which suggests that equations (4) and (8) hold under the "pseudo-random" mapping H.

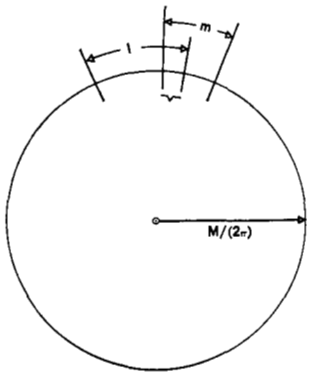
Assumption concerning key set. There are likely to be many groups of keys differing in only one or a few digit positions in an actual file. There is a very much higher clustering than random. This arises from normal methods of assigning keys which have huge gaps and then sequences of consecutive numbers beginning in various digit positions. A further reasonable assumption is that the starting points of the sequences are randomly distributed.

Under the above assumption, the keys may be expressed as members of some sequence, S_n:

$$S_n = S_0 + nr^k \qquad (0 \leq n),$$

where S_0 is the first key of a set numbered consecutively beginning in the "kth digit position."

Figure 1 Interference between sequences



Now under the mapping H, the home addresses are given by

$$H(S_n) = S_n(\text{mod}M) = S_0 + nr^k(\text{mod}M).$$

Since M is relatively prime to r, M is also relatively prime to r^k so that if $n - m \pmod{M} \neq 0$ then $H(S_n) \neq H(S_m)$, and thus no two home addresses within a sequence are the same.

We have yet to consider conflict between home addresses of keys from different sequences. Since under our assumption the starting points of the sequences are randomly distributed, a measure of the average interference between sequences of length l and m is given by the length of the arc formed by the overlapping of arcs of length l and m randomly selected on a circle of circumference M as indicated in Figure 1 (in view of the mod$M$ arithmetic).

Thus, coincidence can occur only between records from different sequences and such coincidence will be random. It follows that the probability of each record being assigned to any address by H is $1/M$, as was the case under the randomly selected mapping G considered earlier. A repetition of the same arguments leads to the conclusion that equations (4) and (8) will also hold for H.

CITED REFERENCES AND FOOTNOTES

1. The underlying reasons are covered in a paper by Werner Buchholz, *File Organization and Addressing* which appears in this issue. The reader's attention is also drawn to the comprehensive bibliography which appears in the latter paper.
2. An analytical approach to this problem is contained in a paper by G. Schay and N. Raver, "A Method for Key-to-Address Transformation," *IBM Journal of Research and Development*, Volume 7, Number 2 (April, 1963).
3. Peterson, W. W., "Addressing for Random Access Storage," *IBM Journal of Research and Development*, Volume 1, Number 2 (April, 1957).