A New Hashing Function : Statistical Behaviour and Algorithm

Zhiyu Tian, Shibai Tong & Shiyuan Yang
Department of Automation
Tsinghua University
Beijing 100084
People's Republic of China

abstract: *Existing hashing functions have various limitations. In this paper a new hashing function is proposed, which divides the range of the key–values into some equal segments, and maps the key–values in each segment linearly into the whole range of the address. The paper analyzes the statistical behavior of the function, and points out that, theoretically, by increasing the number of segments, the distribution of the resulting hash values can always approach uniform, if the key–values can be regarded as continuous. Two methods for obtaining the number of segments, the deterministic and the probabilistic, along with the algorithm, are also proposed.*

Key Words and Phrases: *Hashing function, direct access file, data structure, key–value, hash value*

1. Introduction

There are three factors affecting the efficiency of direct access files [Wu 87]:

1.-The hashing function.

2. The bucket size.

3. Collision resolution

In the literature on hashing techniques, most papers spent much time on the third factor and rarely discussed any particular hashing function [Hill 78] [Wu 87]. As a matter of fact, no matter how clever the collision resolution might be, the efficiency of the direct files will be reduced as long as overflow occurs. Therefore, the best way for improving efficiency is to avoid overflow by selecting a proper hashing function which can yield hash values as uniformly distributed as possible, disregarding the distribution of the key–values.

The existing hashing functions [Cichelli 80] [Fox 92] [Knott 75] [Pearson 90] [Wu 87] provide a variety of selection. However, each of them has certain problems and is thus limited to certain applications. For example, a function proposed in [Knott 75] is:

$$H(x) = [nF_x(x)] \tag{1}$$

where $F_X(x)$ is the cumulative distribution function (c.d.f.) of the key–value x, $\{0,\cdots,n-1\}$ is the range of the bucket address, and $[nF_X(x)]$ is the largest integer smaller than $nF_X(x)$.

According to [Knott 75], this function can definitely yield uniformly distributed hash values, but it is applicable only when $F_X(x)$ is known and easily computable. This requirement is not easy to meet.

Some other hashing functions, such as extraction, multiplication, exclusive oring, addition, division, radix conversion, etc., are easy to compute and do not need knowledge about the key–value distribution, but each of them can only be used for key–values following certain probability laws. Take division function as an example. If the key–value x, whose range is [a,b], can be regarded as a continuous real number, then the function can be expressed as:

$$y = H(x) = x \;\; mod \;\; m = x - im \qquad x \in [im,(i+1)m] \tag{2}$$
$$i \in \left[\left[\frac{a}{m} \right], \left[\frac{b}{m} \right] \right]$$

If the probability density function (p.d.f.) of x is f(x), then according to [Tian 93], the p.d.f. of y is

$$\psi(y) = \begin{cases} \sum\limits_{i=[\frac{a}{m}]}^{[\frac{b}{m}]} f(y+im) & y \in [0,m) \\ 0 & otherwise \end{cases} \tag{3}$$

Equation (3) means that the p.d.f. of y is the sum of all the segments of that of x. If the p.d.f. in the segments compensate for one another, H(x) may approach uniform. However, if this is not satisfied, the distribution will be uncertain or worse.

The minimal perfect hush functions proposed in [Cichelli 80] [Fox 92] can eliminate the problems of wasted space and collision, and provide uniformly distributed hush values. However, the applications of these functions are limited to transform static collections of keys which, once stored, are frequently read and rarely updated. In addition, to use these functions, the keys must be known in advance.

In the following section a new hushing function is proposed, which, although cannot completely solve the problems of wasted space and collision, can yield very uniformly distributed hush values. The function is suitable for keys which are frequently stored and read (this is required in applications such as real–time control and simulation), and requires very few knowledge about the keys and their distribution.

2. The hashing function

Let [a,b] be the range of variable x (representing the key–value), where a and b are finite real numbers, and [0,m] the range of y (the hash value), where m is also a finite real number. We divide [a,b] into n equal segments and get the following hashing function:

$$y = H(x) = \begin{cases} \frac{mn}{b-a}\left(x - a - i\frac{b-a}{n}\right), & a + i\frac{b-a}{n} \leqslant x < a + (i+1)\frac{b-a}{n} \\ & i = 0,\cdots,n-1 \\ 0 & otherwise \end{cases} \tag{4}$$

This function is illustrated in Fig. 1. The following theorem reveals an important property of the function:

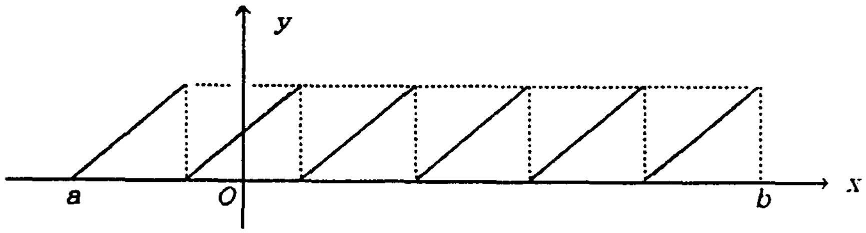

Fig. 1. The proposed hashing function

[Theorem 1] If the p.d.f. of x, f(x), is continuous and piecewise derivable, and if the maximum absolute value of $\frac{df(x)}{dx}$ is a finite value M, then the distribution of y will approach uniform with the increase of n.

[Proof] The c.d.f. of y is:

if 0 < y < m, then:

$$F_Y(y) = P\{Y \leqslant y\} = P\left\{\bigcup_{i=0}^{n-1}\left(a + i\frac{b-a}{n} \leqslant x < a + i\frac{b-a}{n} + \frac{b-a}{mn}y\right)\right\}$$
$$= \sum_{i=0}^{n-1}\int_{a+i\frac{b-a}{n}}^{a+i\frac{b-a}{n}+\frac{b-a}{mn}y} f(x)dx \tag{5}$$

if y < 0, then

$$F_Y = 0 \tag{6}$$

if y > m, then

$$F_Y(y) = 1 \tag{7}$$

The p.d.f. of y can, therefore, be expressed as:

$$\psi(y) = \frac{dF_Y(y)}{dy} = \begin{cases} \frac{b-a}{mn} \sum_{i=0}^{n-1} f\left(a + i\frac{b-a}{n} + \frac{b-a}{mn}y\right) & y \in [0,m] \\ 0 & otherwise \end{cases} \tag{8}$$

The absolute value of the derivative of ψ(y) is:

$$\begin{aligned} \left|\frac{d\psi(y)}{dy}\right| &= \left(\frac{b-a}{mn}\right)^2 \left|\sum_{i=0}^{n-1} f'\left(a + i\frac{b-a}{n} + \frac{b-a}{mn}y\right)\right| \\ &\le \left(\frac{b-a}{mn}\right)^2 \sum_{i=0}^{n-1} \left|f'\left(a + i\frac{b-a}{n} + \frac{b-a}{mn}y\right)\right| \\ &\le \left(\frac{b-a}{mn}\right)^2 \frac{M}{n} \end{aligned} \tag{9}$$

From (9) we learn that $\left|\frac{d\psi(y)}{dy}\right|$ decreases with the increase of n, that is

$$\lim_{n \to \infty} \left|\frac{d\psi(y)}{dy}\right| = 0 \tag{10}$$

Thus, ψ(y) approaches constant when n tends to infinity.

QED

3. Estimating M

According to *Theorem 1*, the p.d.f. of the result of function (4) gets increasingly flat in [0,m] with the number of segments, n. However, if n becomes too large (meaning that the length of each segment becomes too small), the assumption that the key–value, x, is continuous is no longer applicable, and the distribution of H(x) may no longer tend to be uniform.

We used the function proposed in this paper to transform two groups of mutually unequal numbers (i.e., in each group every number appears at most once), and performed χ^2 test [Richard 86] on the resulting hash values. The result of the χ^2 test are shown in Table 1, with different n in ascending order.

From Table 1 we can see that when n is small, χ^2 decreases with the increase of n, meaning that the distribution of the transformed result values gets more and more uniform. This tallies with *Theorem 1*. When n becomes too large (in table 1, when n is larger than 16 and 32, respectively, in the two groups), however, χ^2 becomes uncertain. The reason for this is that, when n is too large, in each segment there are too few key–values, so the key–values can no longer be considered as continuous, and the conditions of *Theorem 1* are no longer satisfied.

Table 1. The result of χ^2 test

n	χ^2(group 1)	χ^2(group 2)
1	3953.17	4158.13
2	1674.69	2246.66
4	1105.53	1349.81
8	1034.10	1151.86
16	936.58	1042.43
32	1107.35	990.90
64	1143.97	1065.17
128	1038.75	1081.23
256	987.77	1105.18
512	1015.49	1054.86

Therefore, it is necessary to select a proper number of segments. Before this, we should first get the usually unknown maximum absolute value of $\frac{df(x)}{dx}$, M. By making some approximations, the following method is proposed:

Divide the range of x into s equal segments, the length of each being $l = \frac{b-a}{s}$. If r is the total number of records whose key–values do not equal one another and r_i the number of key–values in the ith segment, let the value of f(x) in the middle of each segment be $f(x_i) = \frac{r_i}{rl}$ and $f(a) = \frac{r_1}{rl}$, $f(b) = \frac{r_s}{rl}$. Then by linking these mid–segment points the approximate p.d.f. can be obtained (Fig. 2).

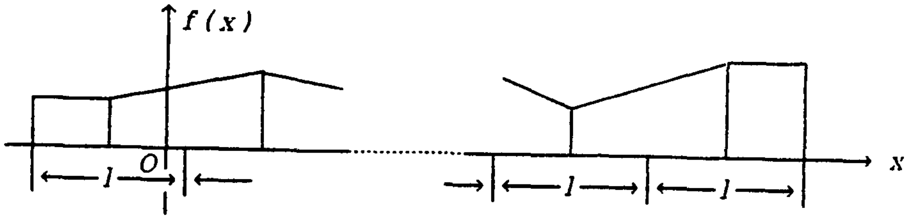

Fig. 2. The approximation of the p.d.f. of x

The area below f(x) between the ith and the (i+1)th mid–segment points (i = 1,⋯,s–1) is

$$\frac{1}{2}\cdot[f(x_i)+f(x_{i+1})]\cdot l=\frac{r_i+r_{i+1}}{2r}$$

The area between a and x_1 is

$$\frac{l}{2}\cdot\frac{r_1}{rl}=\frac{r_1}{2r}$$

Similarly, the area between x_s and b is $\frac{r_s}{2r}$. Therefore, the total area below f(x) is

$$\sum_{i=1}^{s-1}\frac{r_i+r_{i+1}}{2r}+\frac{r_1}{2r}+\frac{r_s}{2r}=\sum_{i=1}^{s}\frac{r_i}{r}=1$$

The last equation is based on the fact that $\sum_{i=1}^{s} r_i = r$. Thus we know that f(x) can be regarded as a p.d.f..

By means of the above approximation method, to get M, the following method is proposed:

Let Δx denote the minimum possible interval between the key–values (for instance, the minimum possible interval between integer key–values is 1, Δx for a real number with n–digit decimal precision is 10^{-n}), and divide [a,b] into r equal segments. The length of each segment is $l=\frac{b-a}{r}$, and the maximum number of key–values a segment can contain is $\left[\frac{b-a}{r\Delta x}\right]+1$. If the distribution of the key–values is uniform, in every segment there should be a key–value, then M = 0. On the other hand, if the distribution is not uniform, M > 0.

If the distribution of the key–values is unknown, we have to estimate the worst–case M. There are two ways the maximum possible value of M occurs:

1). if $r\leqslant\sqrt{\frac{b-a}{\Delta x}}$, then $r\leqslant\frac{b-a}{r\Delta x}$, meaning that in the most uneven case, all the r key–values can be contained in one single segment. In this segment $f(x)=\frac{r}{lr}=\frac{1}{l}=\frac{r}{b-a}$, whereas in the neighboring segments f(x) = 0. Therefore,

$$M=\frac{r/(b-a)}{(b-a)/r}=\left(\frac{r}{b-a}\right)^2 \tag{11}$$

2). if $r>\sqrt{\frac{b-a}{\Delta x}}$, a single segment cannot contain all the key–values. In this case, f(x) is most uneven when one of the segments contains the maximum number of key–values it can contain, $\left[\frac{b-a}{r\Delta x}\right]+1$, and at least one of its neighboring segments does not contain any key–value. Then in the whole segment

$$f(x)=\frac{\dfrac{\left[\frac{b-a}{r\Delta x}\right]+1}{r}}{\dfrac{b-a}{r}}=\frac{\left[\frac{b-a}{r\Delta x}\right]+1}{b-a} \tag{12}$$

and in the neighboring segment f(x) = 0, then

$$M=\frac{r\left\{\left[\frac{b-a}{r\Delta x}\right]+1\right\}}{(b-a)^2} \tag{13}$$

4. Calculating the number of segments

In [Tian 93] a deterministic formula was proposed according to (9) and the Mean–value Principle, which is expressed as:

$$n \geqslant \left[\frac{M}{m\varepsilon}(b-a)^2\right] \tag{14}$$

where ε is the required maximum value of maxψ(y)–minψ(y). Usually we can choose

$$n=n_1=\left[\frac{M}{m\varepsilon}(b-a)^2\right]+1 \tag{15}$$

In the derivation of (14) [Tian 93] (9) was used, in which the estimation was too conservative, causing the obtained n to be too large. By using probabilistic estimation, the obtained value of n will usually be much smaller and can achieve good hashing results. The approach is based on the following theorem:

[Theorem 2] if the assumptions in Section 2 are true, the p.d.f. of x, f(x), is obtained by means of the approximate method proposed in Section 3, and if at each point x ∈ [a, b], $\frac{df(x)}{dx}$ is independent of that at other points, then when n tends to infinity, $\frac{d\psi(y)}{dy}$ in [0,m] approximately follows normal distribution $N\left(0,\ \left(\frac{b-a}{m}\right)^4\frac{t+2rt+2t^2}{n^3r^2}M^2\right)$, where $t=\left[\frac{r^2\Delta x}{b-a}\right]+1$.

[Proof] In the worst case, there are at most $\left[\frac{b-a}{r\Delta x}\right]$ key–values in each of the $t = \left[\frac{r^2\Delta x}{b-a}\right] + 1$ segments, and in the other segments there are no key–values.

a. when $x \in \left[a,\ a+\frac{b-a}{2r}\right]$,

$$P\{f'(x)=M\} = \frac{\binom{r-1}{t-1}}{\binom{r}{t}} = \frac{t}{r}$$

$$P\{f'(x)= -M\} = 0$$

$$P\{f'(x)=0\} = \frac{r-t}{r}$$

b. when $x \in \left[b-\frac{b-a}{2r},\ b\right]$,

$$P\{f'(x)=M\} = 0$$

$$P\{f'(x)= -M\} = \frac{t}{r}$$

$$P\{f'(x)=0\} = \frac{r-t}{r}$$

c. when $x \in \left[a+\frac{b-a}{2r},\ b-\frac{b-a}{2r}\right]$,

$$P\{f'(x)=M\} = \frac{\binom{r-2}{t-1}}{\binom{r}{t}} = \frac{t(r-t)}{r(r-1)}$$

$$P\{f'(x)= -M\} = \frac{t(r-t)}{r(r-1)}$$

$$P\{f'(x)=0\} = 1 - P\{f'(x)=M\} - P\{f'(x)= -M\} = \frac{r^2-2rt+2t^2-r}{r(r-1)}$$

Comprehensively, when x ∈ [a,b],

$$P\{f'(x)=M\} = \frac{1}{2r}\cdot\frac{t}{r} + \frac{r-1}{r}\cdot\frac{t(r-t)}{r(r-1)} = \frac{t+2rt-2t^2}{2r^2}$$

$$P\{f'(x)= -M\} = \frac{t+2rt-2t^2}{2r^2}$$

$$P\{f'(x)=0\} = 2\cdot\frac{1}{2r}\cdot\frac{r-t}{r} + \frac{r-1}{r}\cdot\frac{r^2-2rt+2t^2-r}{r} = \frac{r^2-2rt+2t^2-t}{r^2}$$

The mean value of f'(x) in [a,b] is

$$E[f'(x)] = M \cdot P\{f'(x) = M\} + (-M)P\{f'(x) = -M\} = 0 \tag{16}$$

The variance of f′(x) in [a,b] is

$$D[f'(x)] = E[(f'(x))^2] = \frac{t + 2rt - 2t^2}{r^2} M^2 \tag{17}$$

According to (8)

$$\frac{d\psi(y)}{dy} = \left(\frac{b-a}{mn}\right)^2 \sum_{i=0}^{n-1} f'\left(a + i\frac{b-a}{n} + \frac{b-a}{mn}y\right) \tag{18}$$

$$E[\psi'(y)] = 0 \tag{19}$$

$$\begin{aligned} D[\psi'(y)] &= \left(\frac{b-a}{mn}\right)^4 \cdot nD[f'(x)] = \left(\frac{b-a}{m}\right)^4 \cdot \frac{D[f'(x)]}{n^3} \\ &= \left(\frac{b-a}{m}\right)^4 \cdot \frac{t + 2rt - 2t^2}{n^3 r^2} M^2 \end{aligned} \tag{20}$$

According to the Central Limit Theorem [Larsen 86], $\dfrac{\psi'(y) - E[\psi'(y)]}{\sqrt{D[\psi'(y)]}}$ follows the distribution of N(0,1), that is, ψ'(y) follows distribution $N\left(0,\ \left(\frac{b-a}{m}\right)^4 \frac{t + 2rt - 2t^2}{n^3 r^2} M^2\right)$.

QED

Because

$$max\psi(y) - min\psi(y) \leqslant m \cdot max\frac{d\psi(y)}{dy}$$

then

$$P(max\psi(y) - min\psi(y) \leqslant \varepsilon) \geqslant P\left(\left|\frac{d\psi(y)}{dy}\right| \leqslant \frac{\varepsilon}{m}\right)$$

If we desire P(maxψ(y)−minψ(y)⩽ε)>1−2[1−φ(z)] (where $\varphi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} du$), then

$$z\sqrt{D[\psi'(y)]} \leqslant \frac{\varepsilon}{m}$$

that is

$$n \geqslant \sqrt[3]{z^2\left(\frac{M}{rm\varepsilon}\right)^2 \cdot (b-a)^4 (t + 2rt - 2t^2)} \tag{21}$$

Usually we take

$$n = n_2 = \sqrt[3]{z^2\left(\frac{M}{rm\varepsilon}\right)^2 \cdot (b-a)^4(t+2rt-2t^2)+1} \approx \sqrt[3]{\left(\frac{z}{r}\right)^2(t+2rt-2t^2)n_1^2} \qquad (22)$$

n_2 is usually much smaller than n_1. For example, if m = 100, b–a = 40000, r = 10000, Δx = 1, then from (14), $n_1 > 500$, whereas from (21) $n_2 > 114.47$.

5. Algorithm and numerical examples

Here we only give the algorithm for establishing the hashing function, which is as follows:

a). In accordance with the type of the keys, select a mapping g: **K**→**R** (here **K** denotes the space of keys, and **R** the real axis) so as to convert each key into a unique numerical value (for example, if the key is a character string, it can be encoded into a string of ASCII codes, thus, "Tian" can be converted into $5469616E_H = 1416192366_D$. In fact, [Fox 92] has mentioned some existing approaches for mapping character strings to integers);

b). Calculate n_1 and n_2, and obtain n according to $n = min\left(n_1,\ n_2,\ \frac{b-a}{50\Delta x}\right)$. This is because only when the possible positions of key–values in a segment are more than about 50, the key–value can be regarded as continuous.

c). Form the function according to (4).

The hashing function proposed in this paper has been applied to two groups of key–values, and the χ^2 test [Larsen 86] has been performed on the resulting hash values. Table 2 shows the result of the χ^2 test on both this function and the division function.

Table 2. χ^2 test result

	group 1
	a = 170, b = 10000
	Δx = 1, r = 5279
	m = 1000, ε = 0.02
			group 2
			a = 0, b = 9788
			Δx = 1, r = 4684
			m = 1000, ε = 0.02	

	n	χ^2	n	χ^2
deterministic approach	756	10.4469	724	1.76003
probabilistic approach	126	2.01723	407	1.70453
division function		9.96950		25.2602

6. Conclusion and Discussion

In this paper a new hashing function is given, which, according to the analysis of the paper, can yield uniformly distributed hash values even if the key–values are unevenly distributed. Two methods for estimating n, a parameter of the function, are proposed. The function is easy to compute for computers possessing float point computation capability. Numerical examples are given and the results are good.

According to the analysis in section 2, if x is continuous, the distribution of H(x) can always approach uniform when n increases. However, as the effective bit length of practical computers is limited, the key–values can only take discrete values. Therefore, the χ^2 values in the χ^2 test will cease diminishing when n exceeds a certain value. But, an n corresponding to the smallest χ^2 exists, and this smallest χ^2 is usually smaller than that of the hash values obtained by means of other distribution–independent hashing functions. If the key set is static, we can find this n in advance.

Acknowledgment

The authors wish to thank the referees of this paper for many valuable suggestions on the improvement of the paper.

References

[1]. R. J. Cichelli, Minimal perfect hash functions made simple, Communications of the ACM, Vol.23, No.1, pp17–19, January 1980

[2]. E. A. Fox, et al, Practical minimal perfect hush functions for large databases, Communications of ACM, Vol.35, No.1, pp105–121, January 1992

[3]. E. Hill, Jr., A comparative study on very large databases, Springer–Verlag, 1978

[4]. G. D. Knott, Hashing functions, The Computer Journal, vol.18, No.1, pp265–278, 1975

[5]. R. J. Larsen and M. L. Marx, An introduction to mathematical statistics and its applications, Prentice Hall, England Cliffs, New Jersey, 1986

[6]. P. K. Pearson, Fast hashing of variable–length text strings, Communications of ACM, Vol.33, No.6, pp679–680, June 1990

[7]. Z. Y. Tian and S. B. Tong, An easy–to–compute general purpose hashing function, Advances in Modelling Analysis, B, Vol. 25, No. 1, pp11–18, 1993

[8]. H. L. Wu, Database principle and design (in Chinese), Chinese National Defence Industry Press, 1987