Independence of k multiple H3 hash function

Tao Heng

March 28, 2017

1 The class of functions H3(from initial paper)

Each hash function in H_3 class is a linear transformation $B^T = Q_{r\times w}A^T$ that maps a w-bit binary string $A = a_1a_2\cdots a_w$ to an r-bit binary string $B = b_1b_2\cdots b_r$ as follows:

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_r \end{pmatrix} = \begin{pmatrix} q_{11} & q_{12} & \cdots & q_{1w} \\ q_{21} & q_{22} & \cdots & q_{2w} \\ \cdots & \cdots & \cdots & \cdots \\ q_{r1} & q_{r2} & \cdots & q_{rw} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_w \end{pmatrix} \tag{1}$$

where A and B are the input key (index) and its hash value, and the hash generation matrix $Q_{r\times w}$ is an $r\times w$ matrix defined over $GF(2) = \{0,1\}$ with each H_3 hash function uniquely corresponding to such a $Q_{r\times w}$. The hash function of $Q_{r\times w}$ can map the key ranged in $\{0, 2^w-1\}$ to a hash value ranged in $\{0, 2^r-1\}$.

The multiplication and addition in $GF(2)$ is Boolean AND($\bullet$) and XOR($\oplus$), respectively. According to (1), each bit of B is calculated as follows:

$$b_i = (a_1 \bullet q_{i1}) \oplus (a_2 \bullet q_{i2}) \oplus \cdots \oplus (a_w \bullet q_{iw})\ (i = 1, 2, \cdots, r) \tag{2}$$

We take two examples to illustrate the H_3 class hash function. In the first example, the hash generation matrix is

$$Q_{2\times 8} = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \tag{3}$$

where $w = 8$, $r = 2$, and the hash function is used to map the input key (index) to its hash value: $\{0, \dots, 2^8 - 1 = 255\} \to \{0, \dots, 2^2 - 1 = 3\}$. Under this hash function, the hash value for index 69 can be calculated by Eq.(1), expressed as follows

$$\begin{aligned} & h_1(69) = h_1(01000101) \\ & = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{aligned} \tag{4}$$

where $\begin{pmatrix} 1 \\ 0 \end{pmatrix}^T = \begin{pmatrix} 1 & 0 \end{pmatrix} = 2\,(decimal)$, so the hash value of 69 under the hash function h_1 is $h_1(69) = 2$.

2 How to construct k independent H_3 hash function

First hash function $Q^1_{r\times w}$ is random obtained[1]. we obtain boolean matrix $T_{w\times w}$ of row w and column w randomly, Second hash function is $Q^2_{r\times w} = Q^1_{r\times w}T_{w\times w}$. and so on, the kth hash function is $Q^k_{r\times w} = Q^1_{r\times w}T^k_{w\times w}, (T^k_{w\times w} = \underbrace{T^1_{w\times w}\cdots T^1_{w\times w}}_{k})$. k independent hash function of SFBF is $\{Q^1_{r\times w}, Q^1_{r\times w}T^2_{w\times w}, \cdots, Q^1_{r\times w}T^k_{w\times w}\}$

3 Proof of independent of the k H_3 hash function

3.1 Universal hashing

Let H be a finite collection of hash functions that map a given universe U of keys into the range $0, 1, \ldots, m-1$ Such a collection is said to be universal if for each pair of distinct keys $k, l \in U$, the number of hash functions $h \in$ H for which $h(k) = h(l)$ is at most $|\mathrm{H}|/m$. In other words, with a hash function randomly chosen from H, the chance of a collision between distinct keys k and l is no more than the chance $1/m$ of a collision if h(k) and h(l) were randomly and independently chosen from the set $0, 1, \ldots, m-1$.

3.2 Field $(R, \oplus, \bigwedge)$ is a set $R = 0, 1$ is vector space

let $(R, \oplus), R = \{0, 1\}$ is algebraic system, R = {0,1}.

$(R, \oplus)$ is abel group. proof is below.

- closure: $\forall x_1, x_2 \in R, x_1 \oplus x_2 \in R$
- identity: $\exists e = 0, \forall x \in R, x \oplus e = x$
- inverse: inverse of element 0 is 1,inverse of element 1 is 0.
- associativity: $\exists x, y, z \in R, x \oplus (y \oplus z) = (x \oplus y) \oplus z$
- commutativity: $\exists x, y \in R, x \oplus y = y \oplus x$

$(R, \bigwedge), R = \{0, 1\}$ is monoid. proof is below.

- closure: $\forall x_1, x_2 \in R, x_1 \bigwedge x_2 \in R$
- identity: $\exists e = 0, \forall x \in R, x \bigwedge e = x$
- associativity: $\exists x, y, z \in R, x \bigwedge (y \bigwedge z) = (x \bigwedge y) \bigwedge z$

$(R, \oplus, \bigwedge)$ is ring. proof is below.

- $(R, \oplus)$ is abel group
- $(R, \bigwedge), R = \{0, 1\}$ is monoid

- $\forall x, y, z \in R, a \bigwedge (b \oplus c) = a \bigwedge ((\neg b \bigwedge c) \bigvee (b \bigwedge \neg c)) = (a \bigwedge b) \oplus (a \bigwedge c)$

$(R, \oplus, \bigwedge)$ is field. proof is below.

- commutativity: $\exists x, y \in R, x \bigwedge y = y \bigwedge x$
- additative $\oplus$ identitity 0 is not equal to multiplicative $\bigwedge$ identity 1.

A vector space over a field $(R, \oplus, \bigwedge)$ is a set $R = 0, 1$.

3.3 Proof

Firstly, Independence of $Q^1_{r\times w}$ and $Q^2_{r\times w}$ is equality to $(\exists x \in A, Q^1_{r\times w}(x) = Q^2_{r\times w}(x)\ \textit{is at most}\ \lceil n/|B| \rceil,\ B = 2^r)$, n is the number of insert elements.

As we all know, H_3 hash function is universal hash function. $Q^2_{r\times w}(x) = (Q^1_{r\times w} T^1_{r\times w})(x) = Q^1_{r\times w}(T^1_{r\times w}(x))$(subsec.3.2).

Suppose $\exists y \in B, y = Q^1_{r\times w}(x) = Q^1_{r\times w}(T^1_{w\times w}(x)), x \in M$. How much is most probably largest cardinality of set M so that $y = Q^1_{r\times w}(x) = Q^1_{r\times w}(T^1_{w\times w}(x))$.

$T^1_{w\times w}$ is universal hash function,that is, $\exists t, t = T^1_{r\times w}(x)$ every value t of $T^1_{w\times w}$ is mapped by at most $\lceil n/2^w \rceil$. then $\exists y, y = Q^1_{r\times w}(x) = Q^1_{r\times w}(t)$, $Q^1_{r\times w}$ is universal hash function, every value y of $Q^1_{r\times w}$ is at most $\lceil n/2^w \rceil \times \lceil n/2^r \rceil$.

$$\lceil n/2^w \rceil \times \lceil n/2^r \rceil \leq \lceil n/2^r \rceil \tag{5}$$

in my paper, number of insert elements n is less than 2^w, because insert elements is represented through w-bit binary string. that is, $\lceil n/2^w \rceil = 1$.

et cetera, $Q^i_{r\times w}$ and $Q^j_{r\times w}(i \neq j, 1 < i, j \leq k)$is independent.

References

[1] M. Ramakrishna, E. Fu, and E. Bahcekapili, "A performance study of hashing functions for hardware applications," *terminology*, vol. 5, p. 8, 1994.