Pairwise Independent Hashing

Alek Westover

February 28, 2023

Abstract

Hash-tables are one of the most fundamental data-structures in computer science. In hashing there is a trade-off between how simple your hash function is, and how random its output is. One measure of a hash function's performance is its "max-load": the number of values that hash to the most popular value. In this paper we present some known bounds on max-load achieved by pairwise independent hash functions. In particular, we present a proof that some pairwise independent hash functions have max-load $\Omega(\sqrt{n})$, and also present work due to Knudsen which shows a randomly chosen "linear" hash function achieves max-load $\tilde{\mathcal{O}}(n^{1/3})$. It remains an important open question to determine whether there are pairwise independent hash functions which achieve max-load $o(n^{1/3})$.

Problem Specification. We denote the set $\{1, 2, \dots, n\}$ by $[n]$. Throughout the paper we use the convention of "one-indexing". In particular, $n \bmod n = n$ rather than $n \mod n = 0$. We say that a set of points in $\mathbb{Z}_p$ is an ***interval*** if they are contiguous (wrap-around is allowed). We denote by $|x - y|$ the positive smallest distance between x, y with wrap-around allowed.

We have a universe $[p]$, where we take p to be prime for simplicity[1]. We will take an arbitrary subset $X \subset [p]$, of size $|X| = n$ for some $n < p$. The hashing problem is to give a function $h : [p] \to [n]$, selected randomly from a family of functions $\mathcal{H}$. The ***max-load*** for some set X is a random variable which counts the number of values that hash to the most popular value in $[n]$, i.e.

$$M_X(h) = \max_{i \in [n]} |\{x \in X \ : \ h(x) = i\}|.$$

Our goal is to minimize $\mathbb{E}[M_X]$.

We often use the analogy of throwing balls into bins to discuss hashing.

Complete Independence. The most randomness possible would be if we hash each number $[p]$ to a independently chosen random value in $[n]$.

Proposition 1. *If we use a completely independent hash function, then the expected max-load is* $\Theta\left(\frac{\log n}{\log\log n}\right)$.

Proof. Fix $X \subset [p]$ of size n. For each $i \in [n]$, let H_i be a random variable which counts how many $x \in X$ hash to i. Let $k = \frac{99 \log n}{\log\log n}$. We claim that $\Pr\left[\bigvee_i H_i > k\right]$ is very small. By a union bound and monotonicity we see that

$$\Pr\left[\bigvee_i H_i > k\right] \le n \Pr[H_1 > k] \le n^2 \Pr[H_1 = k]$$

H_1 has binomial distribution, so

$$\begin{aligned}
\Pr[H_1 > k] &= \binom{n}{k} \frac{1}{n^k} \left(1 - \frac{1}{n}\right)^{n-k} \\
&\le \frac{n^k}{k!} \frac{1}{n^k} \frac{1}{e} \left(1 - \frac{1}{n}\right)^{-k} \\
&\le k^{-(1/e - o(1))k} \\
&\le \left(\frac{\log n}{\log\log n}\right)^{-9\frac{\log n}{\log\log n}} \\
&\le \frac{1}{n^8}.
\end{aligned}$$

Now, we can easily bound max-load as follows:

$$\mathbb{E}[M_X] \le n \Pr\left[\bigvee_i H_i > k\right] + 1 \cdot k \le \frac{n}{n^8} + \mathcal{O}\left(\frac{\log n}{\log\log n}\right),$$

as desired. ∎

By analyzing the variance of these random variables it can be shown that this result is tight. [Stanford course notes]

However, there is a major problem: storing / computing a completely random hash function destroys the whole point of a hash-function. Thus, we need simpler hash functions; but would like to still have small max-load.

[1] If non-prime size u is desired, we can round u up to a prime in $(u, 2u)$

Pairwise Independence Lower Bound. One particularly simple class of hash-functions is ***pairwise independent hash functions***: i.e. $\mathcal{H}$ so that for randomly selected $h \in \mathcal{H}$ and any $x \neq y$, $h(x)$ and $h(y)$ are independent and uniformly random. For such functions the probability of any two specific elements colliding is $\frac{1}{n}$. However, it turns out that some pairwise independent hash functions have expected max-load as large as $\Omega(\sqrt{n})$. We note that no pairwise independent hash function can have expected max-load larger than $\mathcal{O}(\sqrt{n})$ (or else the probability of falling in that bin would be so large that we could not have pairwise independence). Peter Shor proposes the following simple construction for such a pairwise independent family:

- Ball n goes to a completely random spot, independent of all other balls.
- Some bin k is chosen, uniformly at random, to be the "crowded" bin.
- A permutation $\pi \in S_{n-1}$ is chosen uniformly at random.
- Each ball $i \in [n-1]$ is sent to bin k with probability $\frac{1}{\sqrt{n}}$, and sent to bin $(k + \pi_i) \mod n$.

Clearly this scheme results in $\sqrt{n}$ expected max-load.

Proposition 2. *In Shor's construction the bins that balls fall into are pairwise independent.*

Proof. Two balls collide in bin k with probability $\frac{1}{n}$. If we condition on this not happening, if we condition on either of the balls falling in bin k the other ball's position is completely random. If we condition on the balls falling in separate bins and neither falling in bin k they also are completely independent of each other. ☐

Linear Hashing is Better. However, Knudsen [1] shows that some methods of pairwise independent hashing, for instance "linear hashing" achieve smaller max-load than the worst case from the previous section.

Theorem 1. *Fix prime p, and $n < p$. Let $X \subset [p]$ be arbitrary. The family of linear hash functions*

$$h_{a,b}(x) = ((ax + b) \mod n) \mod p$$

has expected max-load $\mathbb{E}[M] = \tilde{\mathcal{O}}(n^{1/3})$.

We prove the theorem via a series of propositions.

Proposition 3. *WLOG $b = 0$.*

Proof. Shifting will not affect the max-load. ☐

Proposition 4. *Let $A = n^{-1}aX$ for a the randomly chosen coefficient in the linear hash function. There is an interval I_a such that $|I_a| < \frac{p}{n}$ and*

$$|I_a \cap A| \geq M.$$

Proof. This interval is basically the "pre-image" of the fullest bin which has M values hash to it. In paritcular, if h is the chosen hash function, then we know that h maps M elements of A to some value $i \in [n]$. Then $h^{-1}(i)$ consists of M elements which are the same $\mod n$. multiplying these values by n^{-1} serves to put them all in an interval of size at most $\frac{p}{n}$. ☐

Let $\alpha < \frac{n}{4}, \delta = \Pr[M > 4\alpha]$. We aim to show that δ is very small. Let h_a denote the hash function $x \mapsto ax \mod p \mod n$ and $M_A(h_a)$ be the max-load on A with hash function h_a. Let

$$\mathcal{A} = \{a_0 \ : \ M_A(h_{a_0}) > 4\alpha\}$$

be the set of "bad" a's for $\mathcal{A}$. Let $S = \text{PRIMES} \cap (\alpha, 2\alpha)$. Let $B = S \cap a^{-1}\mathcal{A}$, where a is a random variable, so B is as well. In particular, for some $s \in S$, we have that $\Pr[s \in B] = \delta$, because $a \cdot s$ is a uniform random variable. Note that

$$\mathbb{E}[|B|] = |S|\delta = \Omega\left(\frac{\alpha}{\log \alpha}\delta\right) \tag{1}$$

by linearity of expectation and the Prime Number Theorem.

Now we establish a key lemma which states that there must be many pairs of points in A which are close to each other.

Lemma 1. *There are $\alpha|B|$ ordered pairs $a, a' \in A$ satisfying*

$$|a - a'| < \frac{p}{n\alpha}.$$

Proof. $b^{-1}I_b = \bigcup_{j \in [b]} I_{b,j}$ with each $|I_{b,j}| < \frac{p}{bn}$ because you have to go forward b steps before looping back around.

For $b \neq c$, $b^{-1}I_b \cap c^{-1}I_c$ consists of at most an interval, by virtue of b, c being prime, and in particular of b, c, p all being coprime. In particular, this is for the following reason: if $bx = cy$ and we want $b(x + i) = c(y + j)$ we would need $i = b^{-1}cj$ where $j < \frac{p}{n}$, which corresponds exactly to an element in the same interval as x.

Let $\delta(b, j)$ denote the number as the number of elements c such that $I_{b,j} \cap c^{-1}I_c \neq \varnothing$. Note that of course $\delta(b, j) \geq 1$ because $c^{-1}I_c$ certainly intersects with $I_{b,j}$ for $c = b$. On the other hand, because each pair b, c has only one of the intervals of $b^{-1}I_b$ and $c^{-1}I_c$ intersecting, we have that

$$\sum_{j\in[b]} \delta(b,j) < |B| + b \leq 3\alpha. \tag{2}$$

If $a, a' \in A \cap I_{b,j}$ then

$$|a - a'| < \frac{p}{nb} < \frac{p}{n\alpha};$$

We say that such a pair a, a' are ***close***. Recall that our goal in this lemma is precisely to show that there are lots of pairs a, a' which are close. We define

$$\tau(b,j) = \max(0, |A \cap I_{b,j}| - 1).$$

The number of close pairs a, a' is at least

$$\sum_{b,j} \frac{\tau(b,j)^2}{\delta(b,j)} \tag{3}$$

because the $\delta(b,j)$ factor handles the over-counting. We use the Cauchy Shwarz Inequality on (3) as follows:

$$\sum_{j} \frac{\tau(b,j)^2}{\delta(b,j)} \geq \frac{\left(\sum \tau(b,j)\right)^2}{\sum \delta(b,j)}. \tag{4}$$

We can bound the numerator of (4) by

$$\left(\sum \tau(b,j)\right)^2 \geq (4\alpha - b)^2 \geq (2\alpha)^2$$

because $|I_b \cap bA| \geq 4\alpha$. We have already bounded the denominator of (4) in (2). Thus, the expression in (4) is at most α. Combined with (3), this implies that there are at least $\alpha|B|$ close pairs, as desired. □

As an immediate consequence of the lemma, we have that there are at least $\alpha\,\mathbb{E}[|B|]$ close pairs, in expectation.

Now, we analyze the number of close pairs with a different method.

Proposition 5. *The expected number of close pairs is $\Theta\left(\frac{n}{\alpha}\right)$.*

Proof. The probability that any particular pair is close is just $\frac{1}{n\alpha}$. There are $\Theta(n^2)$ pairs. By linearity of expectation we have the desired result. □

Comparing the two bounds on the number of close pairs gives the desired bound on $\delta = \Pr[M > 4\alpha]$, which will allow us to complete the proof of the theorem.

Proof of Theorem 1.

$$\alpha\,\mathbb{E}[|B|] \leq \frac{n}{\alpha}$$

By (1), we have

$$\frac{\alpha^2}{\log\alpha}\delta \leq \frac{n}{\alpha}.$$

Solving yields

$$\delta \leq \frac{n\log\alpha}{\alpha^3}.$$

Now, we use this to bound the expected max-load.

$$\begin{aligned}\mathbb{E}[M] &\leq \sum_{k>0} \Pr[M \geq k] \\ &\leq \sum_{k=1}^{(n\log n)^{1/3}} 1 + \mathcal{O}\left(\sum_{k>(n\log n)^{1/3}} \frac{n\log n}{k^3}\right) \\ &\leq (n\log n)^{1/3}.\end{aligned}$$

□

References

[1] Mathias Bæk Tejs Knudsen. Linear Hashing is Awesome, June 2017. arXiv:1706.02783 [cs].