SUPPLEMENTAL MATERIAL

A MATHEMATICAL ANALYSIS

In this section, we analyze the expected AAE for frequency estimation of BM. Then we define the "loading rate" to represent the load status of the data structure at a certain time. Additionally, we define "error rate" as the probability of identifying the frequency of an item that has never appeared as non-zero.

A.1 Average absolute error (AAE)

The calculation formula of AAE is defined as $\frac{1}{|\Psi|}\sum_{(e_i\in\Psi)}|f_i-\tilde{f}_i|$, where f_i is the real frequency of item e_i, $\tilde{f}_i$ is the estimated frequency, and the Ψ is the query set.

Suppose that BM records a data stream S with N items, and E is the item set ($|E|=n$). Let e_i be the i^{th} item in E, whose actual frequency is f_i. Assume that there are B entries in each bucket of BM on average, and use $fp()$ to refer to the $fingerprint()$ function. The final frequency estimation of item e_i is

$$\hat{f}_i = f_i - X_i + Y_i \tag{2}$$

where X_i is the decrement brought by the replacement strategy (when an item cannot find a vacancy in the bucket, it will decrement the smallest entry by 1, which will only lead to underestimation). And Y_i is the increment due to fingerprint collisions (which only leads to overestimation). The two error bounds of BitMatcher–the lower bound and the upper bound–result from X_i and Y_i respectively. We will calculate them separately.

A.1.1 For $E(X_i)$. We assume that the replacement strategy is "direct decrease by 1", and no fingerprint collision occurs at this time, which means $Y_i = 0$. Since there are a total of N items in the data stream, and only $2w$ buckets are used to accommodate them, on average, each bucket will have $\frac{N}{2w}$ items inserted. And since there are B entries in each bucket on average to accommodate B types of items , and the frequency of these items increase sequentially in the bucket ($entry_1.cnt \le ... \le entry_B.cnt$), so the maximum number of decays for item e_i occurs in this case: $entry_2 \sim entry_B$ are already occupied by other items, e_i is always in $entry_1$, and all the items coming in later are used to decrease e_i. So it follows that: $E(X_i) \le max\ \#\ of\ decays = \min(f_i, \frac{N}{2w}\times\frac{1}{B+1})$.

A.1.2 For $E(Y_i)$. Assuming that e_i is already in $\mathcal{A}_1[\tilde{h}_1(e_i)]$ (same as $\mathcal{A}_2[\tilde{h}_2(e_i)]$). we introduce indicator variables $I_{i,j}, (1 \le i,j \le n)$ as:

$$I_{i,j} = \begin{cases} & (i\neq j)\wedge(fp(e_i)=fp(e_j))\wedge \\ 1, & [\tilde{h}_1(e_i)=\tilde{h}_1(e_j)\vee \\ & \tilde{h}_2(e_i)=\tilde{h}_2(e_j)] \\ 0, & else \end{cases} \tag{3}$$

The last two conditions in the brackets ensure that e_i and e_j are mapped to at least one same bucket, and its probability can be written as

$$\begin{aligned} &Pr[\tilde{h}_1(e_i)=\tilde{h}_1(e_j)\vee\tilde{h}_2(e_i)=\tilde{h}_2(e_j))] \\ &= Pr[\tilde{h}_1(e_i)=\tilde{h}_1(e_j)] + Pr[\tilde{h}_2(e_i)=\tilde{h}_2(e_j))] \\ &\quad - Pr[\tilde{h}_1(e_i)=\tilde{h}_1(e_j)\wedge\tilde{h}_2(e_i)=\tilde{h}_2(e_j))] \\ &\le \frac{1}{w}+\frac{1}{w}-\frac{1}{w^2} < \frac{2}{w} \end{aligned} \tag{4}$$

Here we assume $hash(e_i)$ is much larger than w. Then we have:

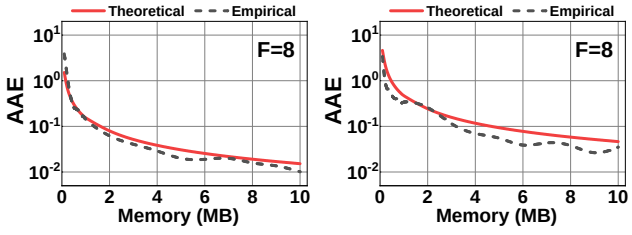
$$\begin{aligned} E(I_{i,j}) &< Pr[fp(e_i)=fp(e_j)]\times\frac{2}{w} \\ &\le \frac{1}{range(fingerprint)}\times\frac{2}{w} = \frac{1}{w2^{\mathcal{F}-1}} \end{aligned} \tag{5}$$

Here we assume the length of fingerprint is $\mathcal{F}$. And since given the two buckets mapped by e_i and e_j (and they have a shared bucket $\mathcal{A}$), e_i has a probability of $\frac{1}{2}$ being mapped into $\mathcal{A}$. So

$$E(Y_i) = E(\sum_{j=1}^{n}\frac{1}{2}I_{i,j}f_j) \approx \frac{1}{2}\sum_{j=1}^{n}f_jE(I_{i,j}) = \frac{N}{w2^{\mathcal{F}}} \tag{6}$$

A.1.3 Experimental Results. Next, we compare the theoretical and empirical AAE. We note that according to the definition of AAE and Eq. (2), we have AAE $= |\hat{f}_i - f_i| = |Y_i - X_i|$. For cold item, its $E(X_i) \le \min(f1, \frac{N}{2w}\times\frac{1}{B+1}) = f_i$ is a very small value. For hot item, the upper bound of $E(X_i)$ is actually too loose, because when a hot item enters a bucket, it will quickly grow into larger entry. It won't be affected by too many replacement strategies in the process. So $E(X_i)$ of hot item is much smaller than $\min(f1, \frac{N}{2w}\times\frac{1}{B+1})$. So to sum up, $E(X_i)$ is always a negligibly small value, and we can only consider the effect of Y_i in the experiment, that is, AAE $\approx |Y_i|$.

We use the CAIDA and IMC dataset. For detailed information, please refer to Section 5.1.2. The experimental results are shown in Fig. 19. The abscissa is memory, the range is 0.1 ~ 1 MB (interval is 0.1) and 1 ~ 10 MB (interval is 1). The black line is the empirical AAE, and the red line is the theoretical expectation of AAE. The fingerprint length $\mathcal{F}$ is 8 bits. We can find that the theoretical values fit fairly well.



(a) CAIDA (b) IMC

Figure 19: Empirical and theoretical AAE.

A.2 Loading rate

First we introduce the definition of loading rate.

Definition 2. *If the BM contains $2w$ buckets, of which x buckets are full (no empty entries), then the **loading rate** at this time is $\frac{x}{2w}$.*

Then we have the following theorem:

Theorem 3. *Assuming that the BM has $2w$ buckets, each bucket has an average of B entries, Each array has k candidate buckets ($k = 1$*

in this paper), and n types of items arrive at this time. Then for the loading rate LR at this time, we have:

$$LR \geq 1 - \sum_{l=0}^{n} \left\{ \binom{n}{l} (\frac{1}{2w})^l (1 - \frac{k}{w})^{n-l} \left[\sum_{i=0}^{\min(l,B-1)} \binom{l}{i} (2k-1)^{l-i} \right] \right\}$$

Proof. We connect BM's $array_1$ after $array_2$. There are $2k$ hash functions in total at this time (the range becomes $0 \sim (2w-1)$, and the insert operation is the same as before). We define the loading rate in this case as LR'. We first prove that $LR \geq LR'$.

We consider the following: when BM processes a data stream, if one hash method is easier to find empty buckets (with at least 1 empty entry in it) than another (assuming the total number of buckets is the same), then the loading rate will increase faster at this time. Because if the hash function maps to a full bucket, it will only perform the replacement strategy without any contribution to the loading rate. So we only need to compare the probability of finding empty buckets between these two hashing methods to know which one has a higher loading rate.

We define the probability that the original BM and the connected BM find an empty bucket at a certain moment as P_{ori}^{LR} and P_{new}^{LR}. Assume that there are x_1 full buckets in $array_1$ and x_2 full buckets in $array_2$ at this time. We have:

$$\begin{aligned} P_{ori}^{LR} &= 1 - P(all\ 2k\ mapped\ buckets\ are\ full) \\ &= 1 - (\frac{x_1}{w})^k \times (\frac{x_2}{w})^k \\ &= 1 - \frac{(x_1 x_2)^k}{w^{2k}} \end{aligned} \tag{7}$$

$$\begin{aligned} P_{new}^{LR} &= 1 - P(all\ 2k\ mapped\ buckets\ are\ full) \\ &= 1 - (\frac{x_1 + x_2}{2w})^{2k} \\ &= 1 - \frac{(\frac{x_1+x_2}{2})^{2k}}{w^{2k}} \end{aligned} \tag{8}$$

Since $\frac{x_1+x_2}{2} \geq (x_1 x_2)^{\frac{1}{2}}$, we have $P_{ori}^{LR} \geq P_{new}^{LR}$, so $LR \geq LR'$.

Next we calculate LR'. Note that the loading rate at a certain moment (n types of items have arrived) is actually the probability that a bucket is full at this time, which is $1 - P(it\ has\ a\ vacancy)$. Assuming that there is no fingerprint collision at this time, there are B entries in each bucket on average. When we look at a bucket, we consider the probability of the following event: the candidate bucket for l of n items is this bucket, and this bucket currently has i entries occupied. Let this probability be P_i^l, with a little knowledge of combinatorics, we have:

$$P_i^l = \binom{n}{l} (\frac{2k}{2w})^l (1 - \frac{2k}{2w})^{n-l} \times \binom{l}{i} (\frac{1}{2k})^i (\frac{2k-1}{2k})^{l-i} \tag{9}$$

Sum it over l and i to get: $P(it\ has\ a\ vacancy) = \sum_{l,i} P_i^l$. Therefore, we have $LR \geq LR' = 1 - \sum_{l,i} P_i^l$, and the result in Theorem 3 can be obtained by substituting Eq. 9 into it. □

Theorem 4. *Under the conditions of Theorem 3, and we consider the fingerprint effect. Assuming that the theoretical upper bound of the loading rate is LR_{opt}, we have:*

$$LR_{opt} = \frac{n}{2wB^2} \left[\sum_{i=0}^{B-1} \frac{2^F - i}{2^F} \right] \tag{10}$$

Proof. The proof is omitted. □

We use the CAIDA dataset to validate our conclusions, which contains approximately $165K$ different kinds of items. As shown in Fig. 20, we show the theoretical optimal value and theoretical lower bound of the loading rate of BM, and the empirical value is recorded every $5K$ items. The experimental results validate our theory, and the empirical value is relatively close to the ideal value.

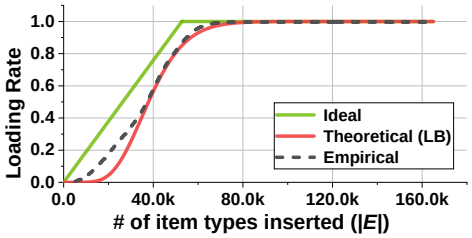


Figure 20: Empirical and theoretical LR.

A.3 Error rate

After processing a data stream, if we query the frequency of an item that has not yet appeared, we do not expect it to return non-zero results. Of course, for a probabilistic algorithm like sketch, this is inevitable when memory is tight. But we expect this probability to be as small as possible. Since there are a number of applications such as blacklist (when some items are blocked) or path verification which would be impacted by it. We return 0 when we do not find the corresponding fingerprint and there is an empty entry in two candidate buckets. And we return the minimum value of two buckets if there is no empty entry. This raises the question: will we identify more non-occurring items as non-zero compared to normal CM-Sketch?

First we introduce the definition of error rate.

Definition 3. *After processing a data stream, the probability that an algorithm identify the frequency of a non-occurring item as non-zero is defined as* ***error rate****.*

Our next step is to show that this error rate of BM is smaller than CM. Suppose d and w_{CM} are the depth and width of CM sketch respectively, w_{BM} is the width of BitMatcher, n represents the type of item in the data stream (i.e., $|E|$), M stands for memory usage (unit is MB).

A.3.1 For CM Sketch. After inserting all items in stream, we look at a counter in a row, which has probability $P_{1\_cnt\_0}$ to be zero. Here $P_{1\_cnt\_0}$=Pr[n kinds of item are mapped to the remaining $(w_{cm} - 1)$ counters]=$(1 - \frac{1}{w_{CM}})^n$. For a item that does not appear, we define P_{CM}^{ER} as the error rate of CM, we have:

$$\begin{cases} P_{CM}^{ER} = (1 - P_{1\_cnt\_0})^d = (1 - (1 - \frac{1}{w_{CM}})^n)^d \\ d \times w_{CM} \times 2 = M \times 1024 \times 1024 \end{cases} \tag{11}$$

Note that in the paper we take the counter size of CM-Sketch as 16 bits (2 bytes).

A.3.2 For BitMatcher. After inserting all items in stream, we also define P_{BM}^{ER} as the error rate of BM. Let's first look at a bucket in $array_1$, the probability of which is full is defined as P_{1_occ}, we have: (we assume there are no fingerprint collisions)

$$\begin{aligned} P_{1\_occ} &= 1 - \sum_{i=0}^{B-1} Pr[i \ entries \ occupied]) \\ &= 1 - \sum_{i=0}^{B-1} \binom{n}{i} (\frac{1}{w_{BM}})^i (1 - \frac{1}{w_{BM}})^{n-i} \end{aligned}$$

Since n and w_{cc} are quite large, we have:

$$P_{1\_occ} \approx 1 - (1 - \frac{1}{w_{BM}})^n \sum_{i=0}^{B-1} \frac{n^i}{i! w_{BM}^i} \tag{12}$$

Furthermore, we have:

$$P_{BM}^{ER} = Pr[A_1[h_1(e)] \ is \ full \ \wedge A_2[h_2(e)] \ is \ full] \tag{13}$$

We then want to compute an upper bound on P_{BM}^{ER}. We give an instance of BM that has been inserted, assuming that there are x_1 buckets in $array_1$ are full, and x_2 buckets in array$_2$ are full. At this time, according to our algorithm, $P_{BM}^{ER} = \frac{x_1 x_2}{w_{BM}^2}$. And, if we connect $array_2$ after $array_1$, and treat it as a whole array, we choose a bucket, and the probability that it is full is defined as $P'_{BM} = \frac{x_1+x_2}{2w_{BM}}$. Note that according to the definition of P'_{cc}, it is actually $P_{1\_occ}(2w_{BM})$. We then have:

$$\begin{aligned} P_{BM}^{ER} - P'_{BM} &= \frac{x_1 x_2}{w_{BM}^2} - \frac{x_1 + x_2}{2w_{BM}} \\ &= \frac{2x_1 x_2 - x_1 w_{BM} - x_2 w_{BM}}{2w_{BM}^2} \\ &< 0 \quad (since \ x_1, x_2 < w_{BM}) \end{aligned} \tag{14}$$

So $P'_{BM} = P_{1\_occ}(2w_{BM})$ is an upper bound of P_{BM}^{ER}:

$$\begin{cases} P_{BM}^{ER} < P_{1\_occ}(2w_{BM}) \\ \quad \approx 1 - (1 - \frac{1}{(2w_{BM})})^n \sum_{i=0}^{B-1} \frac{n^i}{i!(2w_{BM})^i} \\ 2 \times w_{BM} \times 8 = M \times 1024 \times 1024 \end{cases}$$

Note that in the paper we take the bucket size of BitMatcher as 64 bits (8 bytes).

A.3.3 Experimental Results. Below we draw the images of theoretical P_{CM}^{ER} and P_{BM}^{ER}'s upper bound, we take $d = 4$, $n = 0.1Million = 100000$, and the range of M is $0.1 \sim 2MB$.

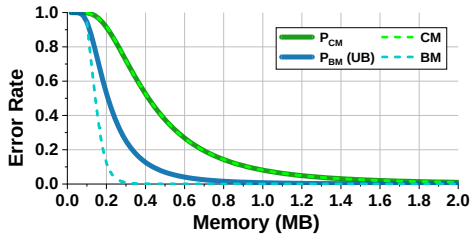


Figure 21: Empirical and theoretical ER.

Note that we also draw the results of P_{CM}^{ER} and P_{BM}^{ER} in the real experiment (Fig. 21: *green-solid* line represents theoretical P_{CM}^{ER}, *blue-solid* line represents theoretical upper bound of P_{BM}^{ER}, *green-dash* line represents empirical P_{CM}^{ER}, *blue-dash* line represents empirical P_{BM}^{ER}). We can find that: (1). The theoretical upper bound of P_{BM}^{ER} is much lower than the theoretical results of P_{CM}^{ER}, and we have also selected different n (e.g. 1 million) to plot, and the results are similar to the above Fig. 21. (2). The real P_{CM}^{ER} is very close to the theoretical value. (3). The real P_{BM}^{ER} is slightly lower than the theoretical upper bound.