Improving Memory Efficiency and Accuracy in High-Speed Data Stream Processing with Probabilistic Data Structures - Formal Analysis

Anonymous Author(s)

1 ERROR BOUND OF NINO-SKETCH

In this section, we derive the error bounds for our Nino-Sketch solution. Let e_i denote the i^{th} largest item with a real frequency of f_i, and let $\hat{f}_i$ represent the estimated frequency of e_i. G represents the number of items, and N denotes the total number of packets in all items. The length of the fingerprint in Stage 1 and Stage 2 is denoted by l_1 and l_2. Our analysis focuses on hot items.

Lemma 1.1. *Assuming the disregard of the order of arrival for each item and a uniform distribution of the probability for each item to be hashed in any bucket, let Y_i represent the increment from fingerprint collision (resulting only in the overestimation of f_i). Let ηG represent the number of cold items, and γN denote the number of packets from cold items. We have*

$$E(Y_i) < \frac{N}{g_1 2^{l_1}} \cdot Pr_{SB1} + \frac{(1-\gamma)N}{g_2 2^{l_2}} \cdot Pr_{SB2}, \tag{1}$$

where

$$Pr_{SB1} < e^{-\frac{G}{g_1 d_1}} \frac{(\frac{G}{g_1 d_1})^2}{2!} \cdot (1 - \frac{1}{2^{l_1}}), \tag{2}$$

$$Pr_{SB2} < e^{-\frac{(1-\eta)G}{g_2 d_2}} \frac{(\frac{(1-\eta)G}{g_2 d_2})^2}{2!} \cdot (1 - \frac{1}{2^{l_2}}). \tag{3}$$

Lemma 1.2. *Following the same assumptions outlined in Lemma 1.1 and setting X_i to be the decrement from the probability-based replacement (which will only result in the underestimation of f_i), then we have*

$$E(X_i) < Pr_{weakest} \cdot Pr_{SB2} \tag{4}$$

where

$$Pr_{weakest} \leq e^{-\frac{i-1}{g_2}} \frac{\left(\frac{i-1}{g_2}\right)^{\lfloor \frac{i-1}{g_2} \rfloor}}{\frac{i-1}{g_2}!}. \tag{5}$$

Theorem 1.3. *Based on Lemma 1.1 and Lemma 1.2, given a very small positive value ϵ and a hot item e_i, we obtain the error bound of Nino-Sketch as*

$$Pr\{|f_i - \hat{f}_i| \geq \epsilon N\} < \frac{1}{\epsilon N} \cdot |Pr_{weakest} \cdot Pr_{SB2} - E(Y_i)|. \tag{6}$$

2 ERROR BOUND PROOF

Let e_i denote the i^{th} largest item with a real frequency of f_i, and let $\hat{f}_i$ be the estimated frequency of e_i. It is important to note that we specifically focus on analyzing hot items in this derivation. Considering the final time point when all items have been processed, we can express the estimation as follows:

$$\hat{f}_i = f_i - X_i + Y_i, \tag{7}$$

where X_i represents the decrement from probability-based replacement (resulting in the underestimation of f_i), and Y_i is the increment from fingerprint collision (resulting in the overestimation of f_i).

2.1 Derivation of Lemma 1.1

Proof. To establish Lemma 1.1, we need to compute the expected count of fingerprint collisions encountered by the target hot item e_i throughout the entire process, as this contributes to *overestimation*. We make the assumption that hot items will not be evicted upon entering *Stage* 1. As fingerprint collisions can potentially happen for hot items in both *Stage* 1 and *Stage* 2, and considering the algorithm's differences in item handling between the two stages, it is crucial to examine the cases independently for each stage.

We represent the length of a fingerprint by l. The total number of items is denoted by G, and N represents the total number of packets included in all items. We use ηG to represent the number of cold items that will not enter *Stage* 2, where η is a parameter ranging from 0 to 1. Similarly, γN represents the total quantity of packets included in the cold items that remain in *Stage* 1, where γ is a parameter ranging from 0 to 1.

Case 1: When a new incoming item e_j cannot find a fingerprint match in the bucket of *Stage* 2, it is inserted into the same bucket as e_i in *Stage* 1 with matching fingerprints, leading to an increment of one in the frequency f_i of the target item. To formalize this process, we define indicator variables $I_{i,j,1} (1 \leq i, j \leq G \text{ and } k_1 \in 1, 2, ..., d_1)$ to represent the occurrence of a fingerprint collision in *Stage* 1:

$$I_{i,j,1} = \begin{cases} 1, & (i \neq j) \wedge (fp(e_i) = fp(e_j) \wedge (\exists! k_1 : (h_{k_1}(e_i) = h_{k_1}(e_j))) \\ 0, & otherwise \end{cases} \tag{8}$$

where $I_{i,j,1} = 1$ if and only if a fingerprint collision occurs between e_i and e_j in *Stage* 1.

The probability of the event $I_{i,j,1} = 1$ is equivalent to the existence of a positive integer $k_1 \in 1, 2, ..., d_1$ such that e_j (where $j \neq i$) is placed in the same bucket as e_i with matching fingerprints during the k^{th} hashing in *Stage* 1. Mathematically, this can be expressed as:

$$\begin{aligned} & Pr\{I_{i,j,1} = 1\} \\ &= \sum_{k=1}^{d_1} Pr\{fp(e_i) = fp(e_j)\} \cdot Pr\{h_{k_1}(e_i) = h_{k_1}(e_j), fpc(k_1)\} \\ &= \sum_{k=1}^{d_1} \frac{1}{2^{l_1}} (\frac{1}{g_1} (Pr_{sb1,k-1} \frac{1}{d_1})) \\ &= \frac{1}{g_1 d_1 2^{l_1}} \sum_{k=1}^{d_1} Pr_{sb1,k-1} \end{aligned} \tag{9}$$

Here, we assume that the probability of any item being hashed to each bucket is uniform (i.e., $\frac{1}{g_1 d_1}$). Additionally, the probability

$$Pr_{sb1,k_1} = \left[\binom{\frac{G}{d_1}}{2} \left(\frac{1}{g_1}\right)^2 \left(1-\frac{1}{g_1}\right)^{\frac{G}{d_1}-2} \cdot \left(1-\frac{1}{2^{l_1}}\right) \right]^{k_1} \tag{10}$$

represents the likelihood that an arbitrary incoming item during the k_1^{th} hash operation in *Stage* 1 ($k_1 \in \{1, 2, ..., d_1\}$) is hashed to a bucket where there are no empty slots, and none of the fingerprints of the items in the bucket match with the fingerprint of the incoming item. Note that the term $\binom{\frac{G}{d_1}}{2} \left(\frac{1}{g_1}\right)^2 \left(1-\frac{1}{g_1}\right)^{\frac{G}{d_1}-2}$ follows the Binomial distribution $\mathrm{B}\left(\frac{G}{d_1}, \frac{1}{g_1}\right)$, and we can approximate this binomial distribution by using a Poisson distribution $\mathrm{Pois}(\lambda_1 = \frac{G}{g_1 d_1})$, which is

$$\begin{aligned} Pr_{sb1,k_1} &\approx [e^{-\lambda_1} \frac{(\lambda_1)^2}{(2)!} \cdot (1-\frac{1}{2^{l_1}})]^{k_1} \\ &\leq e^{-\lambda_1} \frac{(\lambda_1)^2}{2!} \cdot (1-\frac{1}{2^{l_1}}) \\ &= Pr_{SB1}. \end{aligned} \tag{11}$$

Therefore, the expectation of $I_{i,j,1}$ is

$$\begin{aligned} E(I_{i,j,1}) &= 1 \cdot Pr(I_{i,j,1}=1) + 0 \cdot Pr(I_{i,j,1}=0) \\ &= Pr\{I_{i,j,1}=1\} \\ &= \frac{1}{g_1 d_1 2^{l_1}} \sum_{k=1}^{d_1} Pr_{sb1,k-1} \\ &< \frac{1}{g_1 2^{l_1}} \cdot Pr_{SB1}. \end{aligned} \tag{12}$$

Assume that e_i is the first item to enter *Stage* 1, and all other items, except e_i, have the potential to experience a fingerprint collision with it. Then the expectation of the total increment from the fingerprint collision in *Stage* 1 is

$$\begin{aligned} E(Y_{i,1}) &= E(\sum_{j \in \{1,2,...,G\}/\{i\}} I_{i,j,1} \cdot f_j) \\ &< N \cdot E(I_{i,j,1}) \\ &< \frac{N}{g_1 2^{l_1}} \cdot Pr_{SB1} \end{aligned} \tag{13}$$

Case 2: When a new incoming item e_j discovers a matching fingerprint in a bucket in *Stage* 2, it is inserted and placed in the same bucket as e_i with the same fingerprint. Consequently, the frequency f_i of the target item will be incorrectly incremented by one.

Similarly, we define new indicator variables $I_{i,j,2}$ (with $1 \leq i, j \leq (1-\eta)G$ and $k_2 \in \{1, 2, ..., d_2\}$), analogous to $I_{i,j,1}$, to denote the occurrence of a fingerprint collision in *Stage* 2. Thus, we have

$$I_{i,j,2} = \begin{cases} 1, & (i \neq j) \wedge (fp(e_i) = fp(e_j) \wedge (\exists! k_2 : (h_{k_2}(e_i) = h_{k_2}(e_j))) \\ 0, & otherwise \end{cases} \tag{14}$$

We assume that the probability of any hot item being hashed to each bucket in *Stage* 2 is uniform (i.e., $\frac{1}{g_2 d_2}$).

$$Pr\{I_{i,j,2}=1\} = \frac{1}{g_2 d_2 2^{l_2}} \sum_{k=1}^{d_2} Pr_{sb2,k-1}, \tag{15}$$

where

$$Pr_{sb2,k_2} = \left[\binom{\frac{G}{d_2}}{2} \left(\frac{1}{g_2}\right)^2 \left(1-\frac{1}{g_2}\right)^{\frac{G}{d_2}-2} \cdot \left(1-\frac{1}{2^{l_2}}\right) \right]^{k_2}, \tag{16}$$

The term $\binom{\frac{G}{d_2}}{2}(\frac{1}{g_2})^2(1-\frac{1}{g_2})^{\frac{G}{d_2}-2}$ follows the Binomial distribution $\mathrm{B}(\frac{G}{d_2}, \frac{1}{g_2})$ and can be further approximated by the Poisson distribution $\mathrm{Pois}(\lambda_2 = \frac{(1-\eta)G}{g_2 d_2})$. Therefore,

$$\begin{aligned} Pr_{sb2,k_2} &= [e^{-\lambda_2} \frac{(\lambda_2)^2}{(2)!} \cdot (1-\frac{1}{2^{l_2}})]^{k_2} \\ &< e^{-\lambda_2} \frac{(\lambda_2)^2}{2!} \cdot (1-\frac{1}{2^{l_2}}) \\ &= Pr_{SB2}. \end{aligned} \tag{17}$$

Then we obtain the expectation of $I_{i,j,2}$ is

$$\begin{aligned} E(I_{i,j,2}) &= Pr\{I_{i,j,2}=1\} \\ &= \frac{1}{g_2 d_2 2^{l_2}} \sum_{k=1}^{d_2} Pr_{sb2,k-1} \\ &< \frac{1}{g_2 2^{l_2}} \cdot Pr_{SB2}. \end{aligned} \tag{18}$$

Assuming that e_i is the first item to enter *Stage* 2, and all other hot items, except e_i, have the potential to experience a fingerprint collision with it, the expectation of the total increment from the fingerprint collision in *Stage* 2 is

$$\begin{aligned} E(Y_{i,2}) &= E(\sum_{j \in \{hot\ items\}/\{i\}}^{(1-\eta)G} I_{i,j,2} \cdot f_j) \\ &< ((1-\gamma)N) \cdot E(I_{i,j,2}) \\ &< \frac{(1-\gamma)N}{g_2 2^{l_2}} Pr_{SB2}. \end{aligned} \tag{19}$$

Considering both **Case 1** and **Case 2** as independent events, we can calculate the expected increase in the frequency of e_i throughout the entire process due to fingerprint collisions. Then we obtain

$$\begin{aligned} E(Y_i) &= E(Y_{i,1}) + E(Y_{i,2}) \\ &< \frac{N}{g_1 2^{l_1}} \cdot Pr_{SB1} + \frac{(1-\gamma)N}{g_2 2^{l_2}} \cdot Pr_{SB2}. \end{aligned} \tag{20}$$

Therefore, Lemma 1.1 holds. □

2.2 Derivation of Lemma 1.2

Proof. Disregard the arrival order for items and assume that probability-based replacement operations are only performed on hot items in *Stage* 2. Additionally, we assume that the analyzed e_i is subject to at most one execution of probability-based replacement.

The occurrence of probability-based replacement for item e_i can only happen in the following scenario: during the insertion process of a newly arrived item e_j, after g_2 hash operations, all the

buckets found are non-empty, and the fingerprint of e_j does not match the fingerprints of any existing items. Simultaneously, e_i is the weakest item among all the buckets encountered during the insertion process of e_j. The probability of this event, denoted as $Pr_{weakest}$, can be estimated using the Binomial distribution B($i-1$, $\frac{1}{g_2}$).

$$Pr_{weakest} = \binom{i-1}{S-1}(\frac{1}{g_2})^{S-1}(1-\frac{1}{g_2})^{i-1}, \tag{21}$$

where $S = \sum_{i=1}^{g_2} s_i$ denotes the sum of the number of slots in each bucket encountered during the insertion process of e_j, where s_i represents the number of slots in the bucket found during the i^{th} hashing operation.

When the number of hot items $(i-1)$ is large, and the number of buckets in each array (g_2) is small, the Binomial distribution can be approximated by a Poisson distribution Pois($\lambda_w = \frac{i-1}{g_2}$) with the probability density function

$$\begin{aligned} Pr_{weakest} &= e^{-\lambda_w}\frac{(\lambda_1)^{S-1}}{(S-1)!} \\ &\leq e^{-\lambda_w}\frac{(\lambda_w)^{\lfloor\lambda_w\rfloor}}{\lambda_w!}, \end{aligned} \tag{22}$$

Let X_i represent the total loss in frequency for e_1 due to probability-based replacement. Considering each frequency during the process of incrementing e_i from 1 to $E(\hat{f}_i)$ as a state s, we assume that the probability of successfully executing e_i through probability-based replacement is the same at each stage, following a uniform distribution in the range $[1, E(\hat{f}_i)]$. Furthermore, when the frequency of e_i exceeds $E(\hat{f}_i) - 2^\alpha$, the probability of e_i being replaced approaches zero. Therefore, the expected value of X_i can be obtained by calculating the product of the frequency at each stage and the probability of occurrence and successful execution of e_i under that frequency and then summing up the results. Mathematically, this can be expressed as follows:

$$\begin{aligned} E(X_i) &\leq \sum_{t=2^\alpha}^{E(\hat{f}_i)-2^\alpha}(t-2^\alpha)\cdot\frac{Pr_{weakest}\cdot Pr_{SB2}}{E(\hat{f}_i)-2^{\alpha+1}+1}\cdot\frac{1}{t} \\ &< \sum_{t=2^\alpha}^{E(\hat{f}_i)-2^\alpha}\frac{Pr_{weakest}\cdot Pr_{SB2}}{E(\hat{f}_i)-2^{\alpha+1}+1} \\ &= Pr_{weakest}\cdot Pr_{SB2}, \end{aligned} \tag{23}$$

where the term $(t-2^\alpha)$ represents the frequency loss incurred by the occurrence of e_i through probability-based replacement at state t. Here, α is the length of each bucket in *Stage* 1 (in bits), making 2^α the maximum value that each bucket in Stage 1 can represent plus one. It's essential to note that, as we assume that hot items do not undergo probability-based replacement in *Stage* 1, if e_i is replaced through probability-based replacement in *Stage* 2 and subsequently re-enters *Stage* 2 from *Stage* 1, the frequency of e_i is reset to 2^α. Therefore, the frequency loss is given by $(t-2^\alpha)$. The term $\frac{Pr_{weakest}\cdot Pr_{SB2}}{E(\hat{f}_i)-2^{\alpha+1}+1}$ denotes the probability that the newly arrived item, after undergoing d_2 unsuccessful hashing attempts without finding an available slot, successfully performs a probability-based replacement on the weakest item e_i, which is currently in state t, during the insertion. Furthermore, $\frac{1}{t}$ stands for the probability of successfully executing a probability-based replacement on an item at state t. Subsequently, we further bound and simplify the right-hand side of the inequality, ultimately obtaining the final result for equation 23.

Thus, Lemma 1.2 holds. □

2.3 Derivation of Theorem 1.3

Proof. Based on Lemma 1.1 and 1.2, $E(\hat{f}_i)$ can be written as

$$\begin{aligned} E(\hat{f}_i) &= f_i - E(X_i) + E(Y_i) \\ &> f_i - Pr_{weakest}\cdot Pr_{SB2} + E(Y_i), \end{aligned} \tag{24}$$

According to the Markov inequality, we attain the error bound of Nino-Sketch as

$$\begin{aligned} Pr\{|fi-\hat{f}_i| \geq \epsilon N\} &\leq \frac{E(|f_i-\hat{f}_i|)}{\epsilon N} \\ &= \frac{1}{\epsilon N}\cdot|f_i - E(\hat{f}_i)| \\ &< \frac{1}{\epsilon N}\cdot|f_i - (f_i - Pr_{weakest}\cdot Pr_{SB2} + E(Y_i))| \\ &= \frac{1}{\epsilon N}\cdot|Pr_{weakest}\cdot Pr_{SB2} - E(Y_i)|. \end{aligned} \tag{25}$$

Hence, Theorem 1.3 is valid. □