Supplementary Materials of RingSketch

1 Mathematical Proofs

1.1 Proof for Theorem 4.1

Theorem 4.1. *Let $\mathcal{E}$ be the average absolute error (AAE) for estimating the freshness of all items that have appeared within recent $\mathcal{T}$ time units. Under the assumption of no hash collision, we have*

$$\mathbb{E}(\mathcal{E}) \approx \frac{\mathcal{T}_0}{3d}$$

.

Proof. Under the assumption that there are no hash collision in RingSketch, the estimation error is no more than $\frac{\mathcal{T}_0}{d}$.

Consider a certain item e_i. Suppose RingSketch estimates its pointer position as the midpoint between $\mathcal{A}_j[h_j(e_i)]$ and $\mathcal{A}_{j+1}[h_{j+1}(e_i)]$ where $h_j(e_i), h_{j+1}(e_i) \in [0, m)$. Denote $h_j(e_i)$ as x_j and $h_{j+1}(e_i)$ as x_{j+1}. The probability that the pointer locates between $\mathcal{A}_j[h_j(e_i)]$ and $\mathcal{A}_{j+1}[h_{j+1}(e_i)]$ at e_i's last arrival is

$$\mathcal{P}_j = \frac{x_j - x_{j+1} + m}{md}$$

.

The average estimation error of e_i is

$$\overline{\Delta F_i} = \frac{\mathcal{T}_0 \cdot (x_j - x_{j+1} + m)}{4md}$$

.

We derive $\mathbb{E}(\mathcal{E})$ by integrating over x_j and x_{j+1} as

$$\mathbb{E}(\mathcal{E}) \approx \frac{1}{d} \sum_{j=1}^{d} \frac{1}{m^2} \int_0^m \int_0^m \mathcal{P}_j \cdot \overline{\Delta F_i} dx_j dx_{j+1} = \int_{-1}^{0} \int_0^1 (y - x)^2 \cdot \frac{\mathcal{T}_0}{4d} dy dx = \frac{\mathcal{T}_0}{3d}$$

. □

1.2 Proof for Theorem 4.2

Theorem 4.2. *Consider a certain item e_i that has appeared within recent $\mathcal{T}$ time units. Suppose there are w distinct items in the data stream between the last arrival time of e_i and the present moment. Let $\Delta F_i = \left| F_i - \hat{F}_i \right|$ be the estimated error of RingSketch for the freshness of e_i. We have*

$$\mathbb{E}(\Delta F_i) \leqslant \frac{\mathcal{T}_0}{3d} + (1 - \mathcal{P}) \cdot \frac{5\mathcal{T}_0}{12d} + (1 - \mathcal{P})^d \cdot \frac{\mathcal{T}}{2}$$

where $\mathcal{P} = \left(1 - \frac{1}{m}\right)^w \approx e^{-\frac{w}{m}}$ is the probability that e_i does not collide with other item in any of the d parts, and $\mathcal{T}_0 = \frac{dm}{\mathcal{V}}$ is the time for the pointer to complete one cycle.

Proof. We discuss the estimation error of RingSketch in the following three cases.

Case 1: e_i doesn't collide with other items in all the d parts. In such case, the estimation error is resulted from the difference between the estimated pointer position at e_i's last arrival time and the true pointer position, which is no more than $\frac{1}{d}$ cycle. According to the conclusion of Theorem 4.1, the average estimation error is $\frac{\mathcal{T}_0}{3d}$.

Case 2: e_i collides with other items in some of the d parts. In such case, the estimation error is also resulted from the estimation difference of the pointer position. Suppose e_i has hash collisions in j parts ($1 \leqslant j < d$). The estimation difference is no more than $\frac{(j+2)\mathcal{T}_0}{2d}$, whose average is $\frac{(j+2)\mathcal{T}_0}{4d}$.

Case 3: e_i collides with other items in all the d parts. In such case, we cannot accurately infer the number of rotations the pointer has made since e_i's last arrival. Therefore, the estimation error can be up to $\mathcal{T}$, whose average is $\frac{\mathcal{T}}{2}$.

Now we consider the probability $\mathcal{P}$ that e_i does not collide with other item in any of the d parts. For a certain item e_j, the probability that it does not collide with e_i at $\mathcal{A}_1[h_1(e_i)]$ is $1 - \frac{1}{m}$, where m is the number of counters in $\mathcal{A}_1$. As there are w distinct items between e_i's last arrival to the present, we have $\mathcal{P} = (1 - \frac{1}{m})^w \approx e^{-\frac{w}{m}}$.

As the d parts of RingSketch are independent from each other, the number of collisions j obeys the binomial distribution $\mathcal{B}(d, \mathcal{P})$. Therefore, the expected error upper bound can be calculated as

$$\begin{aligned}
\mathbb{E}\left(\Delta F_i\right) &\leqslant \frac{\mathcal{T}_0}{3d} \cdot \mathcal{P}^d + \sum_{j=1}^{d-1} \binom{d}{j} \mathcal{P}^{d-j} (1-\mathcal{P})^j \cdot \frac{(j+2)\mathcal{T}_0}{4d} + (1-\mathcal{P})^d \cdot \frac{\mathcal{T}}{2} \\
&\leqslant \frac{\mathcal{T}_0}{3d} + \frac{5(1-\mathcal{P})\mathcal{T}_0}{12d} + (1-\mathcal{P})^d \cdot \left(\frac{\mathcal{T}}{2} - \frac{(d+2)\mathcal{T}_0}{4d}\right) \\
&\leqslant \frac{\mathcal{T}_0}{3d} + (1-\mathcal{P}) \cdot \frac{5\mathcal{T}_0}{12d} + (1-\mathcal{P})^d \cdot \frac{\mathcal{T}}{2}
\end{aligned}$$

□

1.3 Proof for Lemma 4.1

Lemma 4.1. *Consider a data stream following Zipf distribution with N items derived from n distinct items and parameter α. Let e_k be the k_{th} most frequent item in the Zipf distribution, and w_k be the number of distinct item in the data stream since e_k's last arrival. We have*

$$\mathbb{E}\left(w_k\right) = \sum_{j=1}^{n} \frac{j^{-\alpha}}{j^{-\alpha} + k^{-\alpha}} - \frac{1}{2}$$

In particular, let $N_{\mathcal{T}}$ be the number of items that have appeared within recent $\mathcal{T}$ time units.

When $\alpha = 1.0$, we have

$$\mathbb{E}\left(w_k\right) \leqslant \min\left(k \cdot \ln\left(\frac{n}{k} + 1\right) - \frac{1}{2}, \frac{N_{\mathcal{T}}}{2}\right)$$

When $\alpha = 1.5$, we have

$$\mathbb{E}\left(w_k\right) \leqslant \min\left(2.42k - \frac{1}{2}, \frac{N_{\mathcal{T}}}{2}\right)$$

Proof. Consider the j_{th} most frequent item e_j. In a Zipf distribution, the frequencies of e_j and e_k are $\frac{N}{j^\alpha \zeta(\alpha)}$ and $\frac{N}{k^\alpha \zeta(\alpha)}$ respectively. Therefore, the probability that e_j appears after the last appearance of e_k is

$$p_j = \frac{j^{-\alpha}}{j^{-\alpha} + k^{-\alpha}}$$

.

We can calculate $\mathbb{E}\left(w_k\right)$ by summing up p_j as

$$\mathbb{E}\left(w_k\right) = \sum_{j=1 \wedge j \neq k}^{n} p_j = \sum_{j=1}^{n} \frac{j^{-\alpha}}{j^{-\alpha} + k^{-\alpha}} - \frac{1}{2}$$

Therefore, we have

$$\mathbb{E}\left(w_k\right) = \sum_{j=1}^{n} \frac{1}{1 + \left(\frac{j}{k}\right)^\alpha} - \frac{1}{2} \leqslant \int_0^{\frac{n}{k}} \frac{k}{1+t^\alpha} dt - \frac{1}{2}$$

When $\alpha = 1$, we have

$$\mathbb{E}\left(w_k\right) \leqslant k \ln\left(\frac{1 + \frac{n}{k}}{1 + 0}\right) - \frac{1}{2} = k \ln\left(\frac{n}{k} + 1\right) - \frac{1}{2}$$

When $\alpha = 1.5$, we have $\int_0^{\frac{n}{k}} \frac{1}{1+t^\alpha} dt \approx 2.42$, meaning that

$$\mathbb{E}\left(w_k\right) \leqslant 2.42k - \frac{1}{2}$$

Notice that as there are $N_{\mathcal{T}}$ items that have appeared within recent $\mathcal{T}$ time units, the expectation of w_k cannot be larger than $N_{\mathcal{T}}$ on average. Therefore, we also have

$$\mathbb{E}\left(w_k\right) \leqslant \frac{N_{\mathcal{T}}}{2}$$

□

1.4 Proof for Theorem 4.3

Lemma 4.3. *Consider a data stream following Zipf distribution with N items derived from n distinct items and parameter α. Let $\mathcal{E}$ be the average absolute error (AAE) of RingSketch for estimating the freshness of all items that have appeared within recent $\mathcal{T}$ time units. The upper bound of $\mathbb{E}(\mathcal{E})$ satisfies*

$$\mathbb{E}(\mathcal{E}) \leqslant \frac{\mathcal{T}_0}{3d} + \sum_{k=1}^{n} \frac{k^{-\alpha}}{\zeta(\alpha)} \cdot \left(\frac{5\mathcal{T}_0 w_k}{12md} + \left(\frac{w_k}{m}\right)^d \cdot \frac{\mathcal{T}}{2} \right)$$

where w_k is the number of distinct item in data stream since e_k's last arrival, which can be substituted by the expectation (or upper bound) in Lemma 4.1 to attain an upper bound.

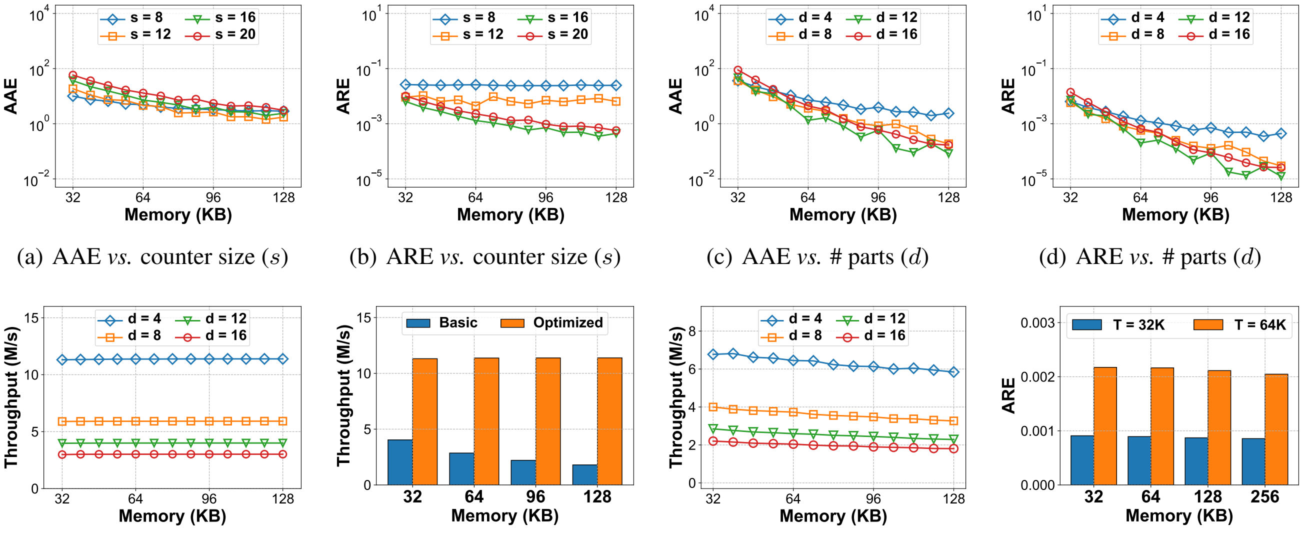
Proof. For each item in a data stream following Zipf distribution, the probability that it is the k_{th} most frequent item e_k is

$$q_k = \frac{1}{k^{\alpha}\zeta(\alpha)}$$

According to Theorem 4.2, we can derive the average error upper bound for all items as

$$\begin{aligned}
\mathbb{E}(\mathcal{E}) &\leqslant \frac{\mathcal{T}_0}{3d} + \sum_{k=1}^{n} q_k \cdot \left(\frac{5\mathcal{T}_0(1-\mathcal{P})}{12d} + \frac{(1-\mathcal{P})^d \cdot \mathcal{T}}{2} \right) \\
&= \frac{\mathcal{T}_0}{3d} + \sum_{k=1}^{n} \frac{k^{-\alpha}}{\zeta(\alpha)} \cdot \left(\frac{5\mathcal{T}_0}{12d} \left(1 - e^{\frac{-w_k}{m}}\right) + \frac{\mathcal{T}}{2}(1 - e^{\frac{-w_k}{m}})^d \right) \\
&\leqslant \frac{\mathcal{T}_0}{3d} + \sum_{k=1}^{n} \frac{k^{-\alpha}}{\zeta(\alpha)} \cdot \left(\frac{5\mathcal{T}_0 w_k}{12md} + \left(\frac{w_k}{m}\right)^d \cdot \frac{\mathcal{T}}{2} \right)
\end{aligned}$$

□



(a) AAE *vs.* counter size (s)

(b) ARE *vs.* counter size (s)

(c) AAE *vs.* # parts (d)

(d) ARE *vs.* # parts (d)

(e) Update throughput *vs.* d

(f) Update throughput acceleration

(g) Query throughput *vs.* d

(h) ARE on large-scale dataset

Figure 1: Impact of RingSketch parameters (CAIDA).

2 Additional Experimental Results

2.1 Impact of RingSketch Parameters

We first evaluate the impact of RingSketch parameters and provide a recommended setup for RingSketch. By default, we set $s = 16$, $d = 4$, and enable the multi-threading acceleration and SIMD acceleration. The experiments are conducted on CAIDA dataset.

Impact of counter size (s) on accuracy (Figure 1(a)-1(b)): We find that the RingSketch using $s = 16$ bit counters can achieve satisfactory accuracy. The results show that larger counter size goes with smaller relative error of RingSketch, and the absolute error is not significantly affected by the counter size. When using 128KB memory, the RingSketch using $s = 16$ bit counters achieves 2.40

AAE and 4.4×10^{-4} ARE, which is very accurate. In practice, we recommend to set $s = 16$, so that RingSketch can achieve high accuracy while being friendly to SIMD instructions.

Impact of hash number (d) on accuracy (Figure 1(c)-1(d)): We find that the RingSketch using $d = 4$ parts (or hash functions) can achieve satisfactory accuracy. The results show that in general, larger d goes with smaller AAE and ARE. But when $d = 4$, RingSketch can already achieve quite small error. When using 128KB memory, the RingSketch using $d = 4$ parts achieves 2.40 AAE and 4.4×10^{-4} ARE, which is very accurate. In practice, we recommend to set $d = 4$ to simultaneously achieve high accuracy and fast speed.

Impact of hash number (d) on update throughput (Figure 1(e)): We find that smaller d goes with higher update throughput, and the update throughput is nearly not affected by memory usage. The update throughput of the RingSketch using $d = 4, 8, 12, 16$ parts is about $11.4, 5.9, 4.0, 3.0$ M/s, respectively.

Impact of parallel optimization on update throughput (Figure 1(f)): We find that the parallel optimization technique using SIMD instructions and multi-threading can significantly improve the update throughput. When using 128KB memory, the basic version of RingSketch only has $1.80 M/s$ update throughput, while that of the optimized RingSketch is $11.4 M/s$, which is $6.33\times$ faster.

Impact of hash number (d) on query throughput (Figure 1(g)): We find that smaller d goes with higher query throughput, and the query throughput is nearly not affected by memory usage. The query throughput of the RingSketch using $d = 4, 8, 12, 16$ parts is about $6.7, 4.0, 2.9, 2.2$ M/s, respectively.

Accuracy on large-scale dataset (Figure 1(h)): We evaluate the accuracy of RingSketch on large-scale 1-hour CAIDA dataset containing about 1.5G items, where we enlarge the measurement window $\mathcal{T}$ from 8K to 16K and 32K. The results show that RingSketch also achieves small ARE, demonstrating its scalability to large-scale dataset. When using 32KB memory, the ARE for $\mathcal{T} = 32K$ and $\mathcal{T} = 64K$ are 9.1×10^{-4} and 2.1×10^{-3} respectively.