Appendices

A DETAILED ANALYSES

A.1 Derivation of Lemma 1

We use Indicator random variables $X_{i,j}$ to determine whether the join of $i \in R, j \in S$ is made.

$$X_{i,j} = \begin{cases} 0 & i, j \text{ not joined in } \Psi. \\ 1 & i, j \text{ joined in } \Psi. \end{cases}$$

The tuples probes as long as they are stored.

$$E[X_{i,j}] = I[i.k = j.k] \cdot p(p_R q_s) = \frac{\varepsilon_r \varepsilon_s}{p} I[i.k = j.k].$$

By the linearity of expectation, the complete derivation is given by:

$$E[|\Psi|] = \sum E[X_{i,j}] = \sum_{i.k=j.k} E[X_{i,j}] = \frac{\varepsilon_R \varepsilon_S}{p} \gamma_{1,1} = \frac{\varepsilon_R \varepsilon_S}{p} J.$$

A.2 Derivation of Lemma 2

Obviously,

$$\mathrm{Var}[\hat{J}] = \frac{p^2}{(\varepsilon_R \varepsilon_S)^2} \mathrm{Var}[|\Psi|],$$

and we can obtain the result by calculating $\mathrm{Var}[|\Psi|]$.

$$\begin{aligned} \mathrm{Var}[|\Psi|] &= \sum \mathrm{Cov}(X_{r,s}, X_{r',s'}) \\ &= \sum E[X_{r,s}, X_{r',s'}] - E[X_{r,s}]E[X_{r',s'}]. \end{aligned}$$

There is a case where there are two tuples a, b with the same key in R, S respectively. a and b are joined if and only if a is sampled by σ_R and b is sampled by σ_S.

$$\begin{aligned} & E[X_{r,s}, X_{r',s'}](r \neq r', s \neq s', r.k \neq r'.k) \\ & = E[X_{r,s}]E[X_{r',s'}] \\ & E[X_{r,s}, X_{r',s'}](r \neq r', s \neq s', r.k = r'.k) \\ & = pq_R^2 q_S^2 I[r.k = s.k = r'.k = s'.k] \\ & E[X_{r,s}, X_{r,s'}](s \neq s') \\ & = pq_R q_S^2 I[r.k = s.k = s'.k] \\ & E[X_{r,s}, X_{r',s}](r \neq r') \\ & = pq_R^2 q_S I[r.k = s.k = r'.k] \\ & E[X_{r,s}, X_{r,s}] = E[X_{r,s}] \end{aligned}$$

Except for the components above, other items in the covariance form of $Var[|\Psi|]$ can be resolved to 0, so the final variance is given by:

$$\begin{aligned} & \mathrm{Var}[\hat{J}] \\ & = \frac{p^2}{(\varepsilon_R \varepsilon_S)^2} \mathrm{Var}[|\Psi|] \\ & = \frac{p^2}{(\varepsilon_R \varepsilon_S)^2} \Big((\gamma_{2,2} - \gamma_{2,1} - \gamma_{1,2} + \gamma_{1,1})(pq_R^2 q_S^2 - p^2 p_R^2 q_s^2) \\ & \quad + \gamma_{1,1}(pp_R q_s - p^2 p_R^2 q_s^2) \quad + (\gamma_{1,2} - \gamma_{1,1})(pq_R q_S^2 - p^2 p_R^2 q_s^2) \\ & \quad + (\gamma_{2,1} - \gamma_{1,1})(pq_R^2 q_S - p^2 p_R^2 q_s^2) \Big) \\ & = \frac{1-p}{p} \gamma_{2,2} + \frac{p - \epsilon_S}{p\epsilon_S} \gamma_{2,1} + \frac{p - \epsilon_R}{p\epsilon_R} \gamma_{1,2} + \frac{(p - \epsilon_S)(p - \epsilon_R)}{p\epsilon_S \epsilon_R} \gamma_{1,1}. \end{aligned}$$

A.3 Derivation of Lemma 3

In the random assumption, there is an equal chance for the two arrival orders of i, j.

$$E[X_{i,j}] = \frac{p(q_R + q_S)}{2} I[i.k = j.k] = \frac{\varepsilon_R + \varepsilon_S}{2} I[i.k = j.k].$$

In particular, it is important to note that according to Definition 2, stream joins need to take into account timestamps, we cannot assume that a join can be generated as long as one of i, j is sampled and it is wrong to use the principle of inclusion-exclusion getting the following result.

$$E[X_{i,j}] = p(q_R + q_S - q_R q_S))I[i.k = j.k].$$

With the equal chance of r goes before s and s goes before r, the complete derivation is given by:

$$E[|\Psi|] = \sum E[X_{i,j}] = \sum_{i.k=j.k} E[X_{i,j}] = \frac{\varepsilon_R + \varepsilon_S}{2} \gamma_{1,1} = \frac{\varepsilon_R + \varepsilon_S}{2} J.$$

A.4 Derivation of Lemma 4

According to Appendix A.3,

$$\mathrm{Var}[\hat{J}] = 4\mathrm{Var}[|\Psi|]/(\varepsilon_R + \varepsilon_S)^2,$$

and we can obtain the result by calculating $\mathrm{Var}[|\Psi|]$.

$$\begin{aligned} \mathrm{Var}[|\Psi|] &= \sum \mathrm{Cov}(X_{r,s}, X_{r',s'}) \\ &= \sum E[X_{r,s}, X_{r',s'}] - E[X_{r,s}]E[X_{r',s'}]. \end{aligned}$$

There is a case where there are two tuples a, b with the same key in R, S respectively. For $\lambda = 1$, when a is sampled by σ_R and b is not sampled by σ_S, if a comes before b, then a join is generated. So, we should discuss $E[X_{r,s}, X_{r',s'}]$ with taking timestamps into account, which means the permutation of the arrival.

$$\begin{aligned} & E[X_{r,s}, X_{r',s'}](r \neq r', s \neq s', r.k \neq r'.k) \\ & \quad = E[X_{r,s}]E[X_{r',s'}] \\ & E[X_{r,s}, X_{r',s'}](r \neq r', s \neq s', r.k = r'.k) \\ & \quad = \frac{1}{4} p(q_R + q_S)^2 I[r.k = s.k = r'.k = s'.k] \\ & E[X_{r,s}, X_{r,s'}](s \neq s') \\ & \quad = \frac{1}{3} p(q_R + q_S q_R + q_S^2) I[r.k = s.k = s'.k] \\ & E[X_{r,s}, X_{r',s}](r \neq r') \\ & \quad = \frac{1}{3} p(q_R^2 + q_S q_R + q_S) I[r.k = s.k = r'.k] \\ & E[X_{r,s}, X_{r,s}] = E[X_{r,s}] \end{aligned}$$

So the final variance is given by:

$$\begin{aligned} & \mathrm{Var}[\hat{J}] \\ & \quad = \quad 4\mathrm{Var}[|\Psi|]/(\varepsilon_R + \varepsilon_S)^2 \\ & \quad = \quad \frac{4}{(\varepsilon_R + \varepsilon_S)^2} \\ & \qquad \Big((\gamma_{2,2} - \gamma_{2,1} - \gamma_{1,2} + \gamma_{1,1}) \end{aligned}$$

$$
\begin{aligned}
& \times\left(\frac{1}{4}p(q_R+q_S)^2-\frac{1}{4}p^2(q_R+q_S)^2\right) \\
& +(\gamma_{1,2}-\gamma_{1,1})\left(\frac{1}{3}p(q_R+q_Sq_R+q_S^2)-\frac{1}{4}p^2(q_R+q_S)^2\right) \\
& +(\gamma_{2,1}-\gamma_{1,1})\left(\frac{1}{3}p(q_R^2+q_Sq_R+q_S)-\frac{1}{4}p^2(q_R+q_S)^2\right) \\
& +\gamma_{1,1}\left(\frac{1}{2}p(q_R+q_S)-\frac{1}{4}p^2(q_R+q_S)^2\right)\Bigg) \\
= & \left(\frac{1-p}{p}\right)\gamma_{2,2}+\frac{(\varepsilon_S-3\varepsilon_R)(\varepsilon_S+\varepsilon_R)+4p\varepsilon_R}{3p(\varepsilon_R+\varepsilon_S)^2}\gamma_{2,1} \\
& +\frac{(\varepsilon_R-3\varepsilon_S)(\varepsilon_R+\varepsilon_S)+4p\varepsilon_S}{3p(\varepsilon_R+\varepsilon_S)^2}\gamma_{1,2}+\frac{-\varepsilon_R-\varepsilon_S-2p}{3p(\varepsilon_R+\varepsilon_S)}\gamma_{1,1}.
\end{aligned}
$$

A.5 Detailed Form of $Var[\hat{J}]$ in General Cases

Besides the notations $\xi, \mu, f(x_1, x_2)$ used in Theorem 1 and Theorem 2, we need to introduce extra tools. Similar to $\xi_{r,s}$, me define $xi_{r,s,r'}$ to be the indicator variable $I[r$ arrives before s before r' or r' arrives before s before $r]$. In $\xi_{r,s,r'}$, we do not distinguish r and r', since their relative order does not affect their contribution, and we have $\xi_{s,r,r'} + \xi_{r,s,r'} + \xi_{r,r',s} = 1$ With ξs using three subscripts, we define two more auxiliary functions:

$$
\begin{aligned}
f_1(x_1,x_2,x_3) = p\Bigg(& \frac{\epsilon_R^2\left(\lambda_S\left(1-\frac{\epsilon_S}{p}\right)+\frac{\epsilon_S}{p}\right)x_1}{p^2} \\
& +\frac{\epsilon_S\epsilon_R\left(\lambda_R\left(1-\frac{\epsilon_R}{p}\right)+\frac{\epsilon_R}{p}\right)x_2}{p^2} \\
& +\frac{\epsilon_S\left(\lambda_R\left(1-\frac{\epsilon_R}{p}\right)+\frac{\epsilon_R}{p}\right)^2x_3}{p}\Bigg), \\
f_2(x_1,x_2,x_3) = p\Bigg(& \frac{\epsilon_S^2\left(\lambda_R\left(1-\frac{\epsilon_R}{p}\right)+\frac{\epsilon_R}{p}\right)x_1}{p^2} \\
& +\frac{\epsilon_R\epsilon_S\left(\lambda_S\left(1-\frac{\epsilon_S}{p}\right)+\frac{\epsilon_S}{p}\right)x_2}{p^2} \\
& +\frac{\epsilon_R\left(\lambda_S\left(1-\frac{\epsilon_S}{p}\right)+\frac{\epsilon_S}{p}\right)^2x_3}{p}\Bigg).
\end{aligned}
$$

With the same technique in Appendix A.2 and Appendix A.4, we calculate:

$$
\begin{aligned}
Var[\hat{J}] = \quad & \frac{1-p}{p}\sum_{r,s,r',s'} I[r.k=s.k=r'.k=s'.k \wedge r\neq r' \wedge s\neq s'] \\
& \times\frac{f(\xi_{s,r},\xi_{r,s})}{f(\mu_{S,R},\mu_{R,S})}\frac{f(\xi_{s',r'},\xi_{r',s'})}{f(\mu_{S,R},\mu_{R,S})} \\
& +\sum_{r,r',s} I[r.k=r'.k=s.k \wedge r\neq r'] \\
& \times\frac{f_1(\xi_{r,r',s},\xi_{r,s,r'},\xi_{s,r,r'})-f(\xi_{s,r},\xi_{r,s})f(\xi_{s,r'},\xi_{r',s})}{f(\mu_{S,R},\mu_{R,S})^2} \\
& +\sum_{r,s,s'} I[s.k=s'.k=r.k \wedge s\neq s'] \\
& \times\frac{f_2(\xi_{s,s',r},\xi_{s,r,s'},\xi_{r,s,s'})-f(\xi_{s,r},\xi_{r,s})f(\xi_{s',r},\xi_{r,s'})}{f(\mu_{S,R},\mu_{R,S})^2} \\
& +\sum_{r,s} I[s.k=r.k]\times\frac{(1-f(\xi_{s,r},\xi_{r,s}))f(\xi_{s,r},\xi_{r,s})}{f(\mu_{S,R},\mu_{R,S})^2}.
\end{aligned}
$$

A.6 Detailed Analysis of Scaling on SUM

Let Ψ_{SUM} represent the direct SUM of FreeSam's output data, and $\Xi^{(s)}_{r,s} = \sum_{r,s} I[r.k = s.k] \cdot \xi_{r,s} \cdot r.v$, $\Xi^{(s)}_{s,r} = \sum_{r,s} I[r.k = s.k] \cdot \xi_{s,r} \cdot r.v$. Hence, given the property $J_{\mathsf{SUM}} = \Xi^{(s)}_{r,s} + \Xi^{(s)}_{s,r}$, we derive the following:

$$
E[\Psi_{\mathsf{SUM}}] = \Xi^{(s)}_{S,R}(\epsilon_S - \frac{\epsilon_R\epsilon_S}{p})\lambda_R + \Xi^{(s)}_{R,S}(\epsilon_R - \frac{\epsilon_R\epsilon_S}{p})\lambda_S + \frac{\epsilon_R\epsilon_S}{p}.
$$

By defining $\Xi^{(s)}_{s,r}/J_{\mathsf{SUM}}$ and $\Xi^{(s)}_{r,s}/J_{\mathsf{SUM}}$ as $\mu^{(s)}_{R,S}$ and $\mu^{(s)}_{S,R}$ respectively, we can present the unbiased estimator as:

$$
\hat{J}_{\mathsf{SUM}} = \Psi_{\mathsf{SUM}}/(\mu^{(s)}_{S,R}(\epsilon_S - \frac{\epsilon_R\epsilon_S}{p})\lambda_R + \mu^{(s)}_{R,S}(\epsilon_R - \frac{\epsilon_R\epsilon_S}{p})\lambda_S + \frac{\epsilon_R\epsilon_S}{p}).
$$

Applying techniques similar to those used in solving for output size, the final variance can be computed as follows:

$$
\begin{aligned}
Var[\hat{J}] = \quad & \frac{1-p}{p}\sum_{r,s,r',s'} I[r.k=s.k=r'.k=s'.k \wedge r\neq r' \wedge s\neq s'] \\
& \times\frac{r.vf(\xi_{s,r},\xi_{r,s})}{f(\mu^{(s)}_{S,R},\mu^{(s)}_{R,S})}\frac{r'.vf(\xi_{s',r'},\xi_{r',s'})}{f(\mu^{(s)}_{S,R},\mu^{(s)}_{R,S})} \\
& +\sum_{r,r',s} I[r.k=r'.k=s.k \wedge r\neq r'] \\
& \times\frac{r.v\,r'.vf_1(\xi_{r,r',s},\xi_{r,s,r'},\xi_{s,r,r'})-r.vf(\xi_{s,r},\xi_{r,s})r'.vf(\xi_{s,r'},\xi_{r',s})}{f(\mu^{(s)}_{S,R},\mu^{(s)}_{R,S})^2} \\
& +\sum_{r,s,s'} I[s.k=s'.k=r.k \wedge s\neq s'] \\
& \times\frac{r.v\,r.vf_2(\xi_{s,s',r},\xi_{s,r,s'},\xi_{r,s,s'})-r.vf(\xi_{s,r},\xi_{r,s})r.vf(\xi_{s',r},\xi_{r,s'})}{f(\mu^{(s)}_{S,R},\mu^{(s)}_{R,S})^2} \\
& +\sum_{r,s} I[s.k=r.k]\times\frac{(1-r.vf(\xi_{s,r},\xi_{r,s}))r.vf(\xi_{s,r},\xi_{r,s})}{f(\mu^{(s)}_{S,R},\mu^{(s)}_{R,S})^2}.
\end{aligned}
$$

A.7 Parameter Setting Analysis

There are two parameters unknown (p and λ). As foregoing statements, p majorly affects variance, while λ majorly affects output size. The adjustment between these metrics formulates the adaption of FreeSam.

More precisely, the adaption is achieved by using p to maintain the variance level and modifying λ to adjust the output size. Therefore, λ is in major charge of the adaption, and we prefer λ to be an arbitrarily settable value. Our major concern with parameter setting is p. Thus, the focus is to study the impact of p on variance, and the means is to take the partial derivative of the variance for p. Then, we figure out the coefficient of p. Through a mindset of verifying the boundary conditions, we fill $\lambda = 0$ and $\lambda = 1$ to explore the basic behavior of p first. Notably, $\lambda = 0$ and $\lambda = 1$ both make $\frac{f(\xi_{s,r},\xi_{r,s})}{f(\mu_{S,R},\mu_{R,S})}\frac{f(\xi_{s',r'},\xi_{r',s'})}{f(\mu_{S,R},\mu_{R,S})}$ become 1, and the rest parts are only $\frac{f_1(\xi_{r,r',s},\xi_{r,s,r'},\xi_{s,r,r'})}{f(\mu_{S,R},\mu_{R,S})}$, $\frac{f_2(\xi_{s,s',r},\xi_{s,r,s'},\xi_{r,s,s'})}{f(\mu_{S,R},\mu_{R,S})}$, and $\frac{f(\xi_{s,r},\xi_{r,s})}{f(\mu_{S,R},\mu_{R,S})^2}$. Since f_1 and f_2 also have a strong symmetry, we focus on:

$$\frac{f_1(\xi_{r,r',s},\xi_{r,s,r'},\xi_{s,r,r'})}{f(\mu_{S,R},\mu_{R,S})}. \tag{7}$$

When setting $\lambda = 0$, $\frac{f(\xi_{s,r},\xi_{r,s})}{f(\mu_{S,R},\mu_{R,S})^2} = \frac{p}{\epsilon_R \epsilon_S}$ and (7)= $1/\epsilon_S$. When setting $\lambda = 1$, $\frac{f(\xi_{s,r},\xi_{r,s})}{f(\mu_{S,R},\mu_{R,S})^2} = \frac{\epsilon_S \xi_{s,r} + \epsilon_R \xi_{r,s}}{(\epsilon_S \mu_{S,R} + \epsilon_R \mu_{R,S})^2}$ and

$$(7) = \frac{\frac{1}{p}(\epsilon_R^2 \xi_{r,r',s} + \epsilon_S \epsilon_R \xi_{r,r',s}) + \epsilon_S \xi_{s,r,r'}}{(\mu_{S,R} + \mu_{R,S})^2}. \tag{8}$$

Using the principle of inclusion-exclusion like Appendix A.2 and Appendix A.4, we can refill these constants back to the coefficient of p. It roughly conforms to the format $\frac{1}{p}(\gamma_{2,2}C_0 - \gamma_{1,2}C_1 - \gamma_{2,1}C_2 + \gamma_{1,1}C_3) + p\gamma_{1,1}C_4$, where $C_0, ...C_4$ are constants. Let $\lambda = 0$, it makes no impact on variance, $C_0, .., C_3$ all become 1, and the solution of p is :

$$\sqrt{\frac{\epsilon_R \epsilon_S(\gamma_{2,2} - \gamma_{1,2} - \gamma_{2,1} + \gamma_{1,1})}{\gamma_{1,1}}}.$$

When λ takes effect, these constants fluctuate in a relatively small range, since even the $rho = 1$ case does not change the level of p's coefficient. Not to mention the unpredictability of streaming data, the parameter setting has heuristic features at its root. Thus, by Occam's Razor, we cut the fluctuation and choose the direct way to set p by making $\lambda = 0$.

A.8 Hardware Conscious Implementation

By implementing the sampling-based *IaWJ* on multicore systems, we utilize both multicore parallelism and SIMD for acceleration.

Utilizing multicores. Multicore systems offer the capability to accelerate sampling through parallelism. In this part, we elaborate on the design strategies that effectively exploit multicore systems.

The schemes for partitioning on multicore systems significantly influence the efficiency of the sampling process. We employ the Join-Matrix (JM) strategy to facilitate the eager procedure. JM conceptualizes the potential join as a matrix, essentially the Cartesian product of the two relations, and allocates one stream to each kernel, while dividing the other stream into disjoint, equal-sized subsets. Specifically, one stream (R) is assigned to each kernel, while the other stream (S) is divided into disjoint and equal-sized subsets. Regardless of the sampling behavior, the tuples are assigned to cores by the aforementioned strategy.

Besides shuffling data for parallelism, a key reason for introducing partitioning is to optimize the use of spatial locality. We adopt this idea and manage to avoid direct sampling swaps between cores, maintaining a sampling stratum within each core to minimize storage switching. Once a tuple is assigned to a core through the partitioning scheme, sampling is initiated immediately followed by the subsequent build and probe phases within the hash table structure. Consequently, with the Join-Matrix (JM) approach, we ensure the individual sampling occurs for the stream distributed across all cores. For the stream copied to all cores, it is resampled independently within each core, but according to the expectation, this does not change the estimation. Regarding the estimator and its variance, we treat each partition as a case of stratified sampling. Besides, the independence of sampling across different strata (cores) allows us to straightforwardly aggregate the estimators and their variances to compute the overall result. This method demonstrates relatively stable performance due to JM's balanced allocation of tuples across the cores.

Utilizing SIMD. To harness the capabilities of new hardware features in our sampling process, we develop an AVX-accelerated pseudo-random number generator (PRNG). Given that AVX operations are performed in SIMD mode, our design includes an AVX-accelerated PRNG within a circular queue buffer. In this setup, we do not store all newly generated random numbers in the buffer permanently. Instead, we refill the buffer with AVX-accelerated PRNG and transfer the pointer back to the head of the queue only when it touches the boundary of the buffer. Moreover, to prevent shared memory access between the multicores and the locked synchronous consumption associated with the queue pointers, we keep the PRNGs separate in each core.

A.9 Impact of Modern Hardware Optimization

Multicore scalability. We now examine multicore scalability in terms of the average normalized 95th latency in Figure 15. Accordingly, we use one, two, four, and eight threads in all trials. We set both the sampling rate and the probe utilization to 0.6 for a more visible decline of the latency. As expected, the latency of FreeSam decreases as the number of cores increases. When more cores are used, FreeSam is expected to achieve better performance.

PRNG buffer size. We then validate the AVX-accelerated PRNG design mentioned in Section 6. It is necessary to determine how to set the size of the circular queue, i.e., the size of the PRNG buffer, to obtain optimal performance. We fix the sampling rate and probe utilization to 0.6 to better manifest the performance difference. The results in Figure 16 show that each dataset exhibits a downward-convex pattern. The optimal performance is achieved by setting the buffer size at around 26KB, close to the L1 cache size (32KB) of our Intel Core i9-10900X CPU. This suggests that an ideal buffer size can be achieved at the L1 cache level.

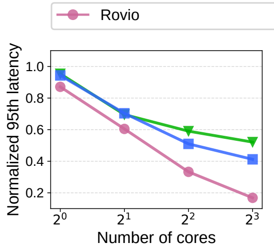

Figure 15: Average normalized latency with varying number of cores.

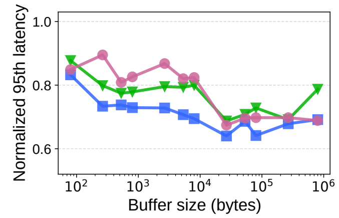

Figure 16: Average normalized latency with varying PRNG buffer sizes.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009