Timely Reporting of Heavy Hitters using External Memory

Prashant Pandey*, Shikha Singh*, Michael A. Bender, Jonathan W. Berry, Martin Farach-Colton, Rob Johnson, Thomas M. Kroeger, Cynthia A. Phillips

Sandia National Laboratories
Rutgers
Stony Brook University
Williams
vmware
Carnegie Mellon University

*Both authors contributed equally to this research.

Open problem in streaming

- A **high-speed stream** of key-value pairs arriving over time
- **Goal:** report every key **as soon as** it appears **24 times** without missing any
- Firehose benchmark (Sandia National Lab) simulates the stream
  https://firehose.sandia.gov/

Why should we care about this problem

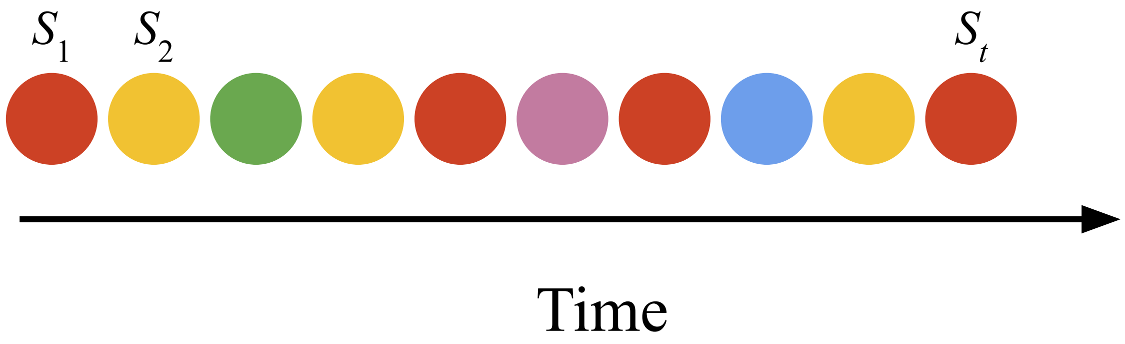
Defense systems for cyber security monitor high-speed streams for malicious traffic ➡ **Catch all malicious events**

Malicious traffic forms a small portion of the stream ➡ **Small reporting threshold**

Automated systems take defensive actions for every reported event ➡ **Minimize false positives**

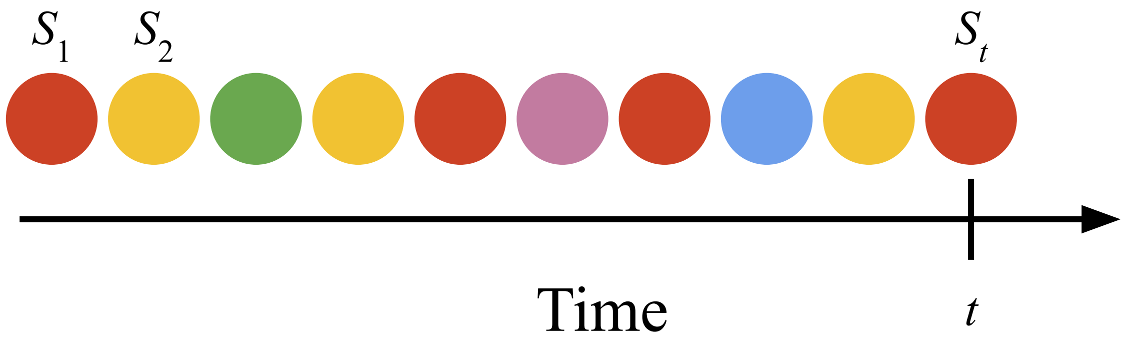
Timely event detection problem

- Stream of elements arrive over time



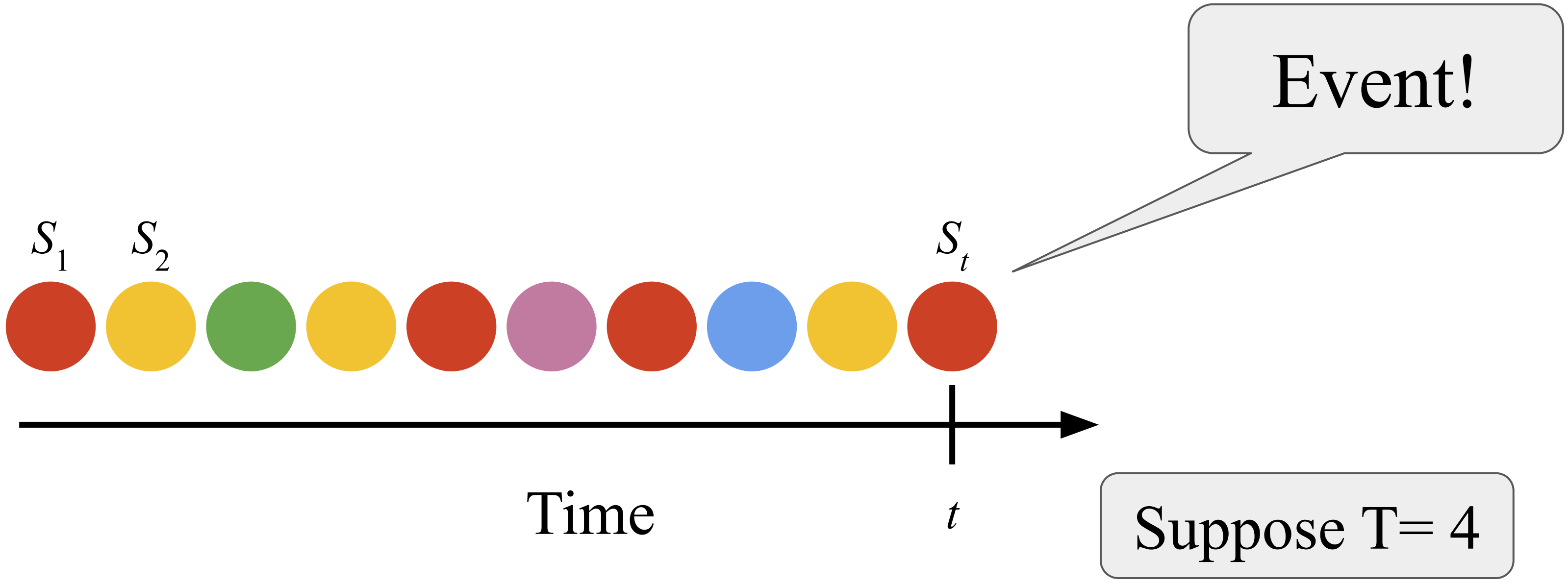
Timely event detection problem

- Stream of elements arrive over time
- An **event** occurs at time t if S_t occurs exactly T times in $(s_1, s_2 \ldots . s_t)$



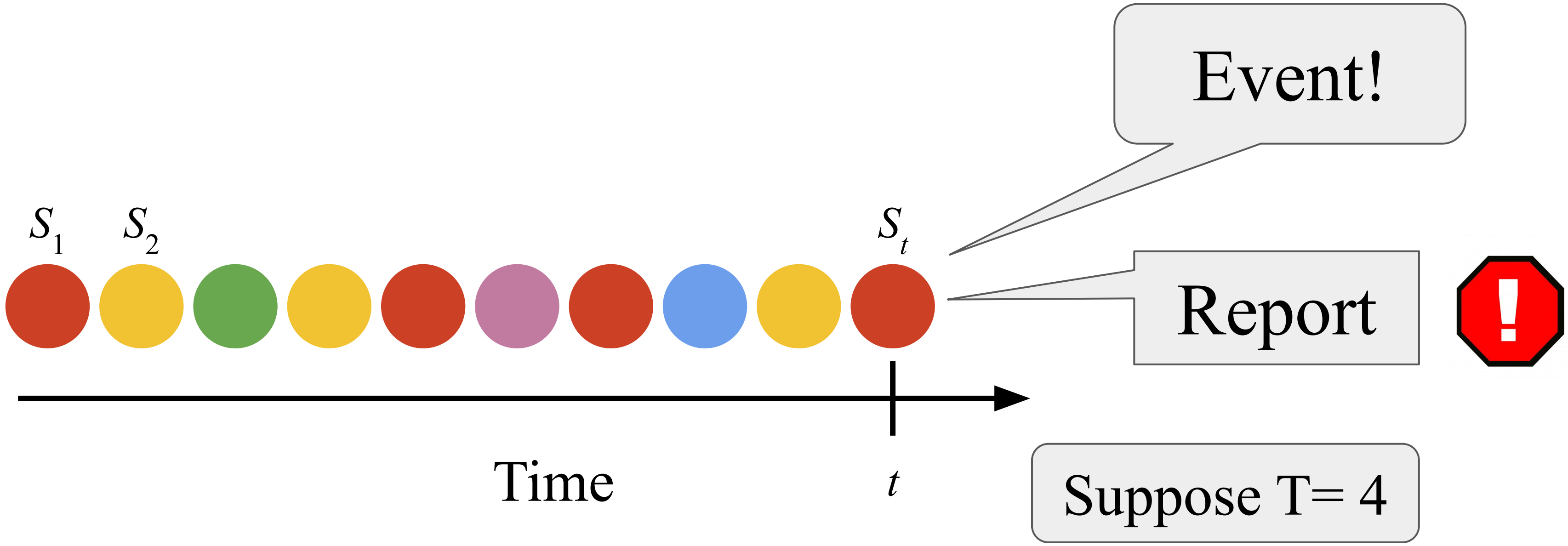
Timely event detection problem

- Stream of elements arrive over time
- An **event** occurs at time t if S_t occurs exactly T times in $(s_1, s_2 \ldots s_t)$



Timely event detection problem

- Stream of elements arrive over time
- An **event** occurs at time t if S_t occurs exactly T times in $(s_1, s_2 \ldots s_t)$
- In **timely event-detection problem (TED)**, we want to report all events shortly after they occur.



Features we need in the solution

- Stream is large (e.g., terabytes) and high-speed (millions/sec)

High throughput ingestion

Features we need in the solution

- Stream is large (e.g., terabytes) and high-speed (millions/sec)

High throughput ingestion

- Events are high-consequence real-life events

No false-negatives; few false-positives

Timely reporting (real-time)

Features we need in the solution

- Stream is large (e.g., terabytes) and high-speed (millions/sec)

High throughput ingestion

- Events are high-consequence real-life events

No false-negatives; few false-positives

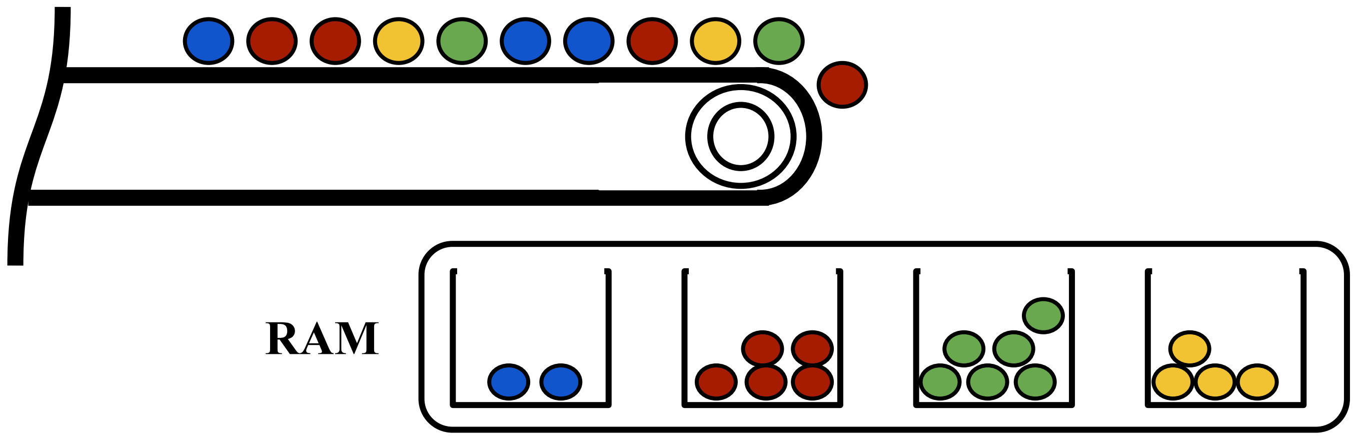
Timely reporting (real-time)

- Very small reporting threshold $T << N$ (stream size)

Very small reporting thresholds

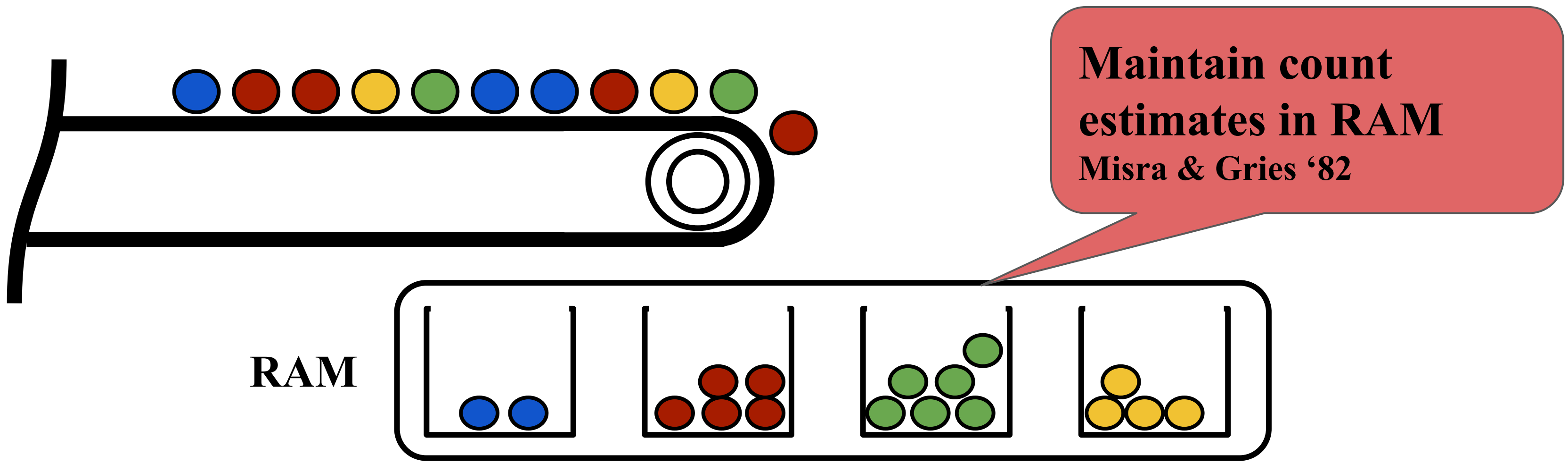
One-pass streaming has errors

- **Heavy hitter problem:** report items whose frequency $\geq \varphi N$
- Exact one-pass solution solution requires $\mathbf{\Omega}(N)$ space



One-pass streaming has errors

- **Approximate solution**: report all items with count $\geq \varphi N$, none with $< (\varphi - \varepsilon)N$ [Alon et al. 96, Berinde et al. 10, Bhattacharyya et al. 16, Bose et al. 03, Braverman et al. 16, Charikar et al. 02, Cormode et al. 05, Demaine et al. 02, Dimitropoulos et al. 08, Larsen et al. 16, Manku et al. 02.]
- Approximate solutions requires: $\Omega(1/\varepsilon)$

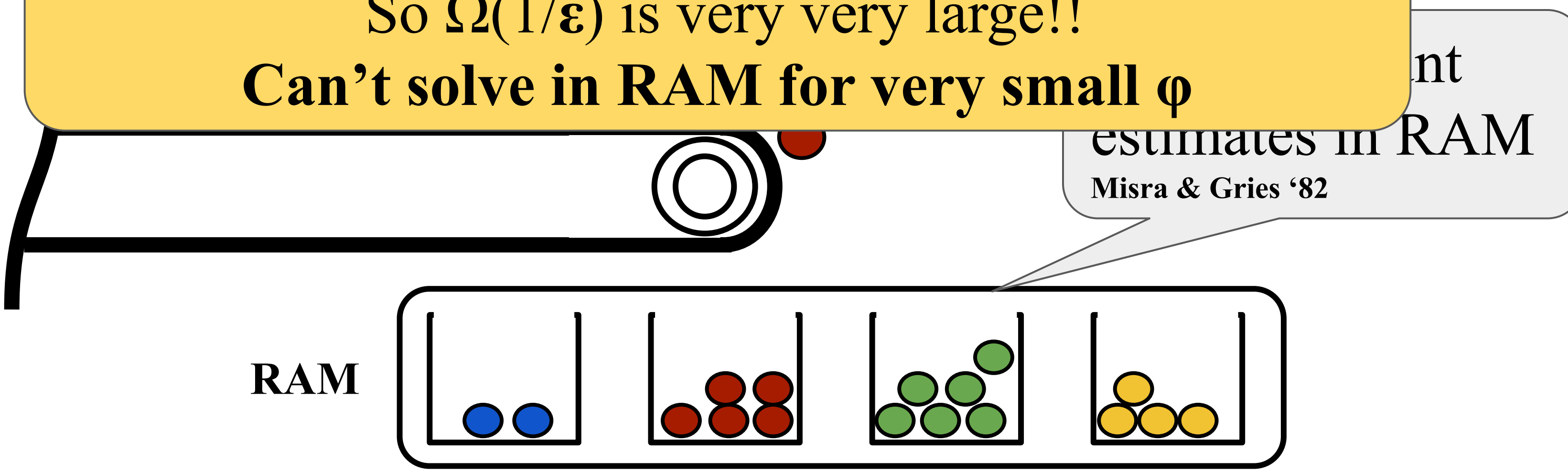


Real time with false-positives!

One-pass streaming has errors

- **Approximate solution**: report all items with count $\geq \varphi N$, none with $< (\varphi - \varepsilon)N$ [Alon et al. 96, Berinde et al. 10, Bhattacharyya et al. 16, Bose et al. 03, Braverman et al. 16, Charikar et al. 02, Cormode et al. 05, Demaine et al. 02, Dimitropoulos et al. 08, Larsen et al. 16, Manku et al. 02.]
- Approximate solutions requires: $\Omega(1/\varepsilon)$

For Sandia, φN is a small constant (24),
So $\Omega(1/\varepsilon)$ is very very large!!
Can't solve in RAM for very small φ



Real time with false-positives!

One-pass solution has:

- Stream is large (e.g., terabytes) and high-speed (millions/sec)

High throughput ingestion ✓

- Events are high-consequence real-life events

No false-negatives; few false-positives ✓

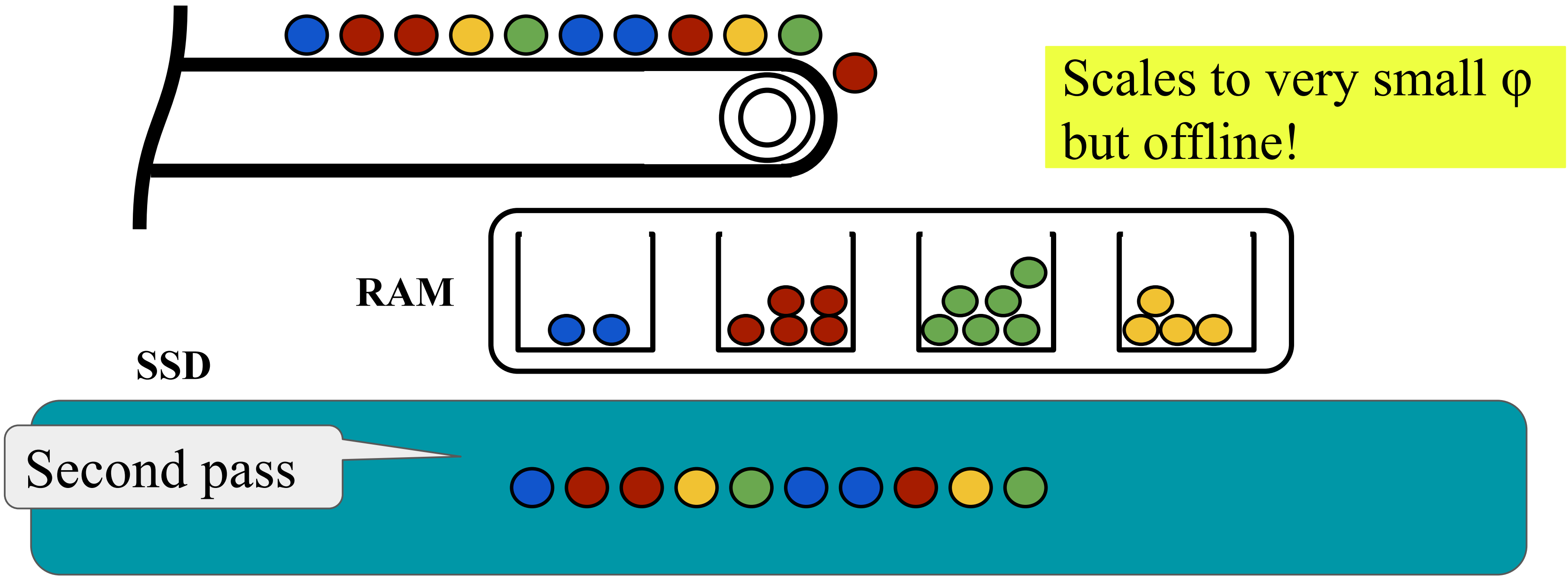
Timely reporting (real-time) ✓

- Very small reporting threshold $T << N$ (stream size)

Very small reporting thresholds ✗

Two-pass streaming isn't real-time

- A second pass over the stream can get rid of errors
- Store the stream on SSD and access it later



Two-pass solution has:

- Stream is large (e.g., terabytes) and high-speed (millions/sec)

  High throughput ingestion 

- Events are high-consequence real-life events

  No false-negatives; few false-positives ✓

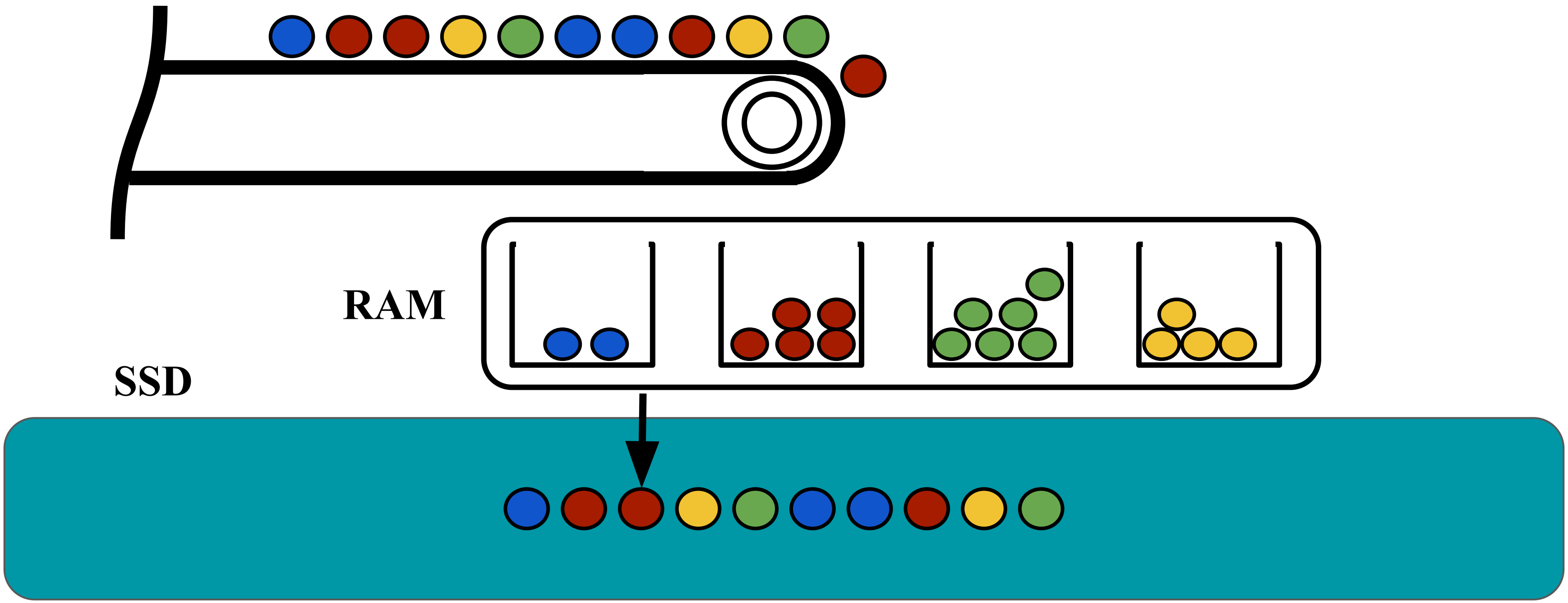
  Timely reporting (real-time) ✗

- Very small reporting threshold $T << N$ (stream size)

  Very small reporting thresholds ✓

If data is stored: why not access it?

Why wait for second pass?



Our contribution

Streaming model

External memory algorithms

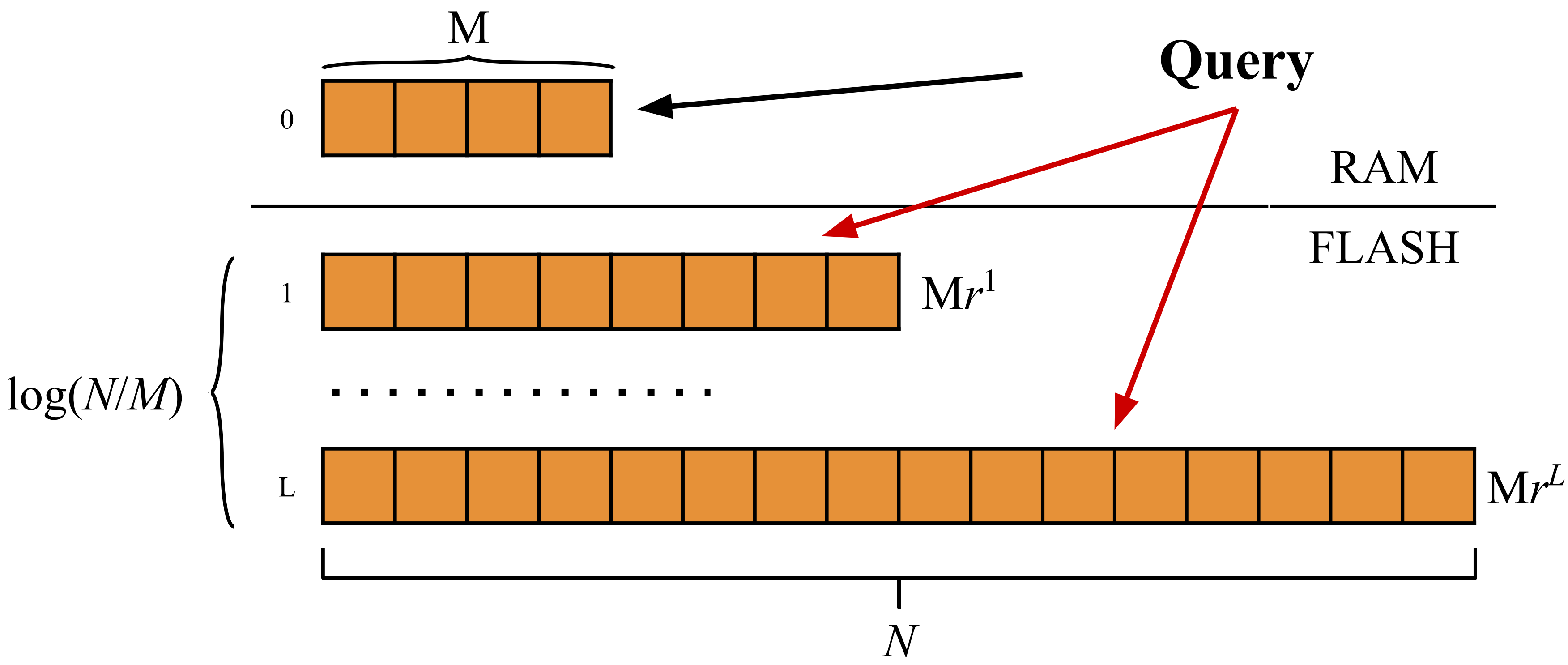
Combine streaming and EM algorithms to solve real-time event detection problem

Idea

Use an efficient external-memory counting data structure to scale Misra Gries algorithm to SSDs

Cascade filter: write-optimized quotient filter

Bender et al. '12, Pandey et al. '17



- The Cascade filter efficiently scales out-of-RAM
- It accelerates insertions at some cost to queries

Cascade filter operations

Insert	Query
$O\left(\frac{1}{B}\log\frac{N}{M}\right)$	$O\left(\log\frac{N}{M}\right)$

Cascade filter operations

Insert	Query
$O\left(\frac{1}{B}\log\frac{N}{M}\right)$	$O\left(\log\frac{N}{M}\right)$

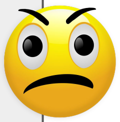
$<$ 1 I/O per observation

Cascade filter operations

Insert	Query
$O\left(\frac{1}{B}\log\frac{N}{M}\right)$	$O\left(\log\frac{N}{M}\right)$

< 1 I/O per observation

> 1 I/O per observation

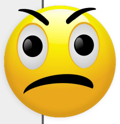


Cascade filter doesn't have real-time reporting

But every insert is also a query in real-time reporting!

Insert	Query
$O\left(\frac{1}{B}\log\frac{N}{M}\right)$	$O\left(\log\frac{N}{M}\right)$

< 1 I/O per observation 

> 1 I/O per observation 

Cascade filter doesn't have real-time reporting

But every insert is also a query in real-time reporting!

Insert	Query
$O\left(\frac{1}{B}\log\frac{N}{M}\right)$	$O\left(\log\frac{N}{M}\right)$

Traditional cascade filter doesn't solve the problem!

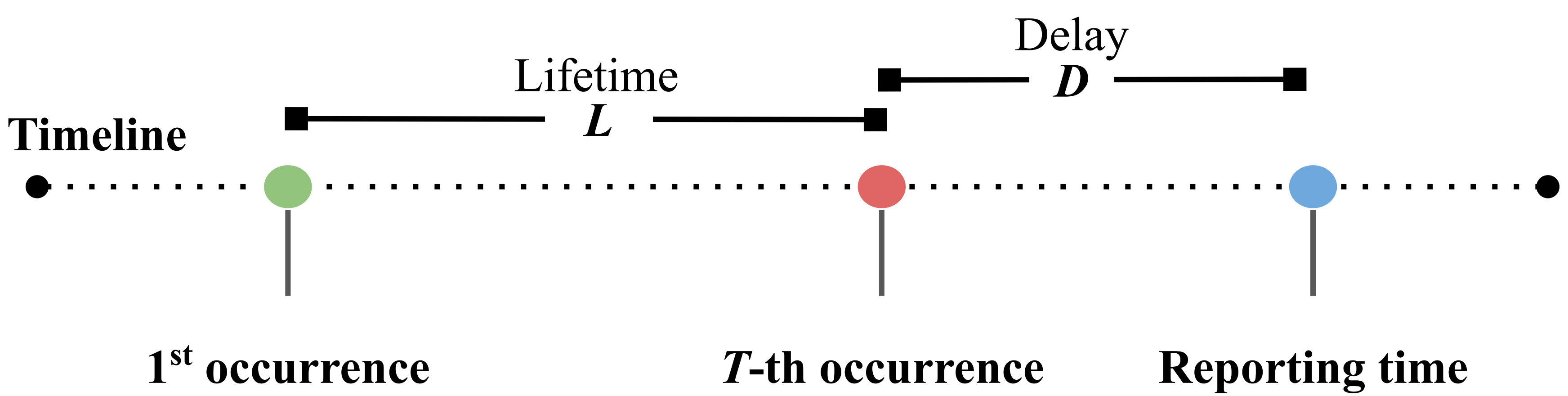
observation

observation

Idea: reporting with bounded delay

We define the time stretch of a report to be

$$\text{Time stretch} = 1 + \alpha = 1 + \frac{\text{Delay}}{\text{Lifetime}}$$



This paper: Leveled External-Memory Reporting Table (LERT)

- Given a stream of size N and $\varphi N > \boldsymbol{\Omega(N/M)}$ the amortized cost of solving real-time event detection is

$$O\left(\left(\frac{1}{B} + \frac{1}{(\phi - 1/M)N}\right) \log \frac{N}{M}\right)$$

- For a **constant** $\boldsymbol{\alpha}$, can support arbitrarily small thresholds φ with amortized cost

$$O\left(\frac{1}{B} \log \frac{N}{M}\right)$$

Takeaway: Online reporting comes at the cost of throughput but almost online reporting is essentially free!

This paper: Leveled External-Memory Reporting Table (LERT)

- Given a stream of size N and $\varphi N > \mathbf{\Omega(N/M)}$ the amortized cost of solving real-time event detection is

Can achieve timely reporting at effectively the optimal insert cost; no query cost

- with amortized cost

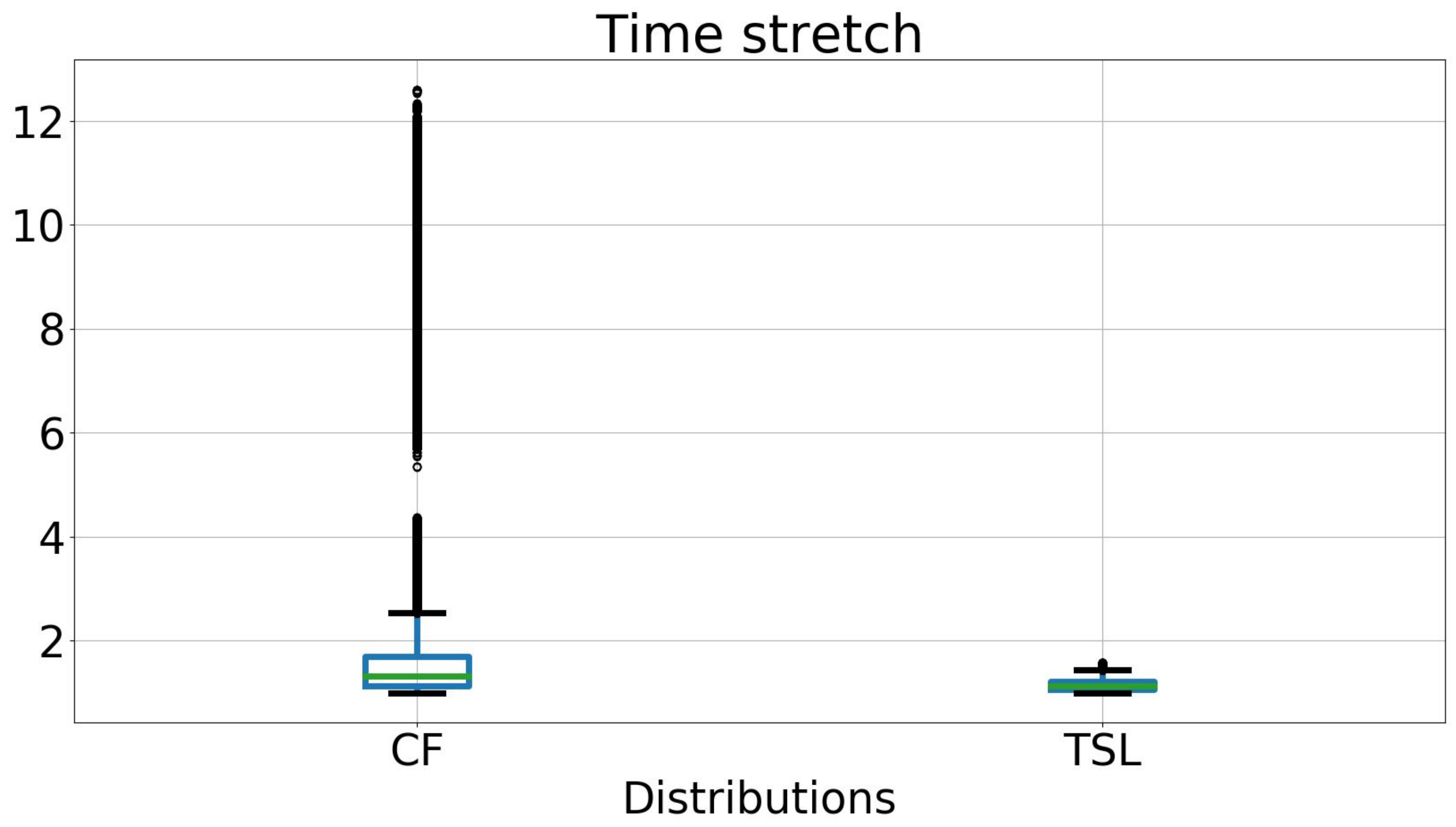
$$O\left(\frac{1}{B}\log\frac{N}{M}\right)$$

Takeaway: Online reporting comes at the cost of throughput but almost online reporting is essentially free!

Evaluation

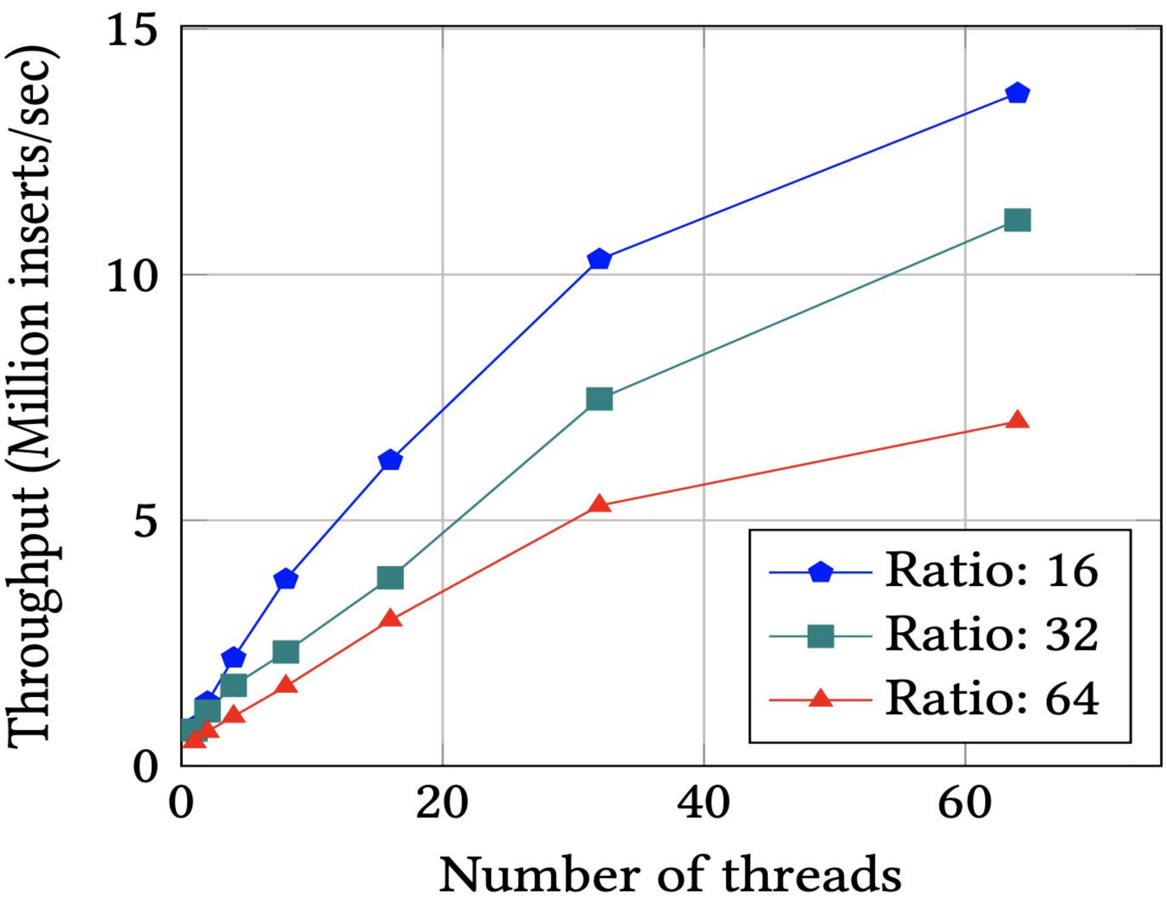
- Empirical timeliness
- High-throughput ingestion

Evaluation: empirical time stretch



Average time stretch is 43% smaller than theoretical upper bound.

Evaluation: scalability



The insertion throughput increases as we add more threads.

We can achieve > 13M insertions/sec.

LERT: supports scalable and real-time reporting

- Stream is large (e.g., terabytes) and high-speed (millions/sec)

High throughput ingestion ✓

- Events are high-consequence real-life events

No false-negatives; few false-positives ✓

Timely reporting (real-time) ✓

- Very small reporting threshold $T << N$ (stream size)

Very small reporting thresholds ✓

Conclusion

- **We can solve timely event detection problem at a level of precision that is not possible in the streaming model.**

- **This work suggests new research opportunities:**
  - What other streaming problems can be solved in external memory at comparable speed?
  - What is the right model for streaming in modern external memory?