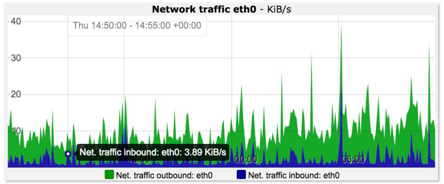
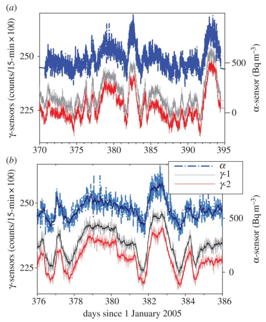
# Making Data Sketches Accurate and Fast by Filtering the Cold and Aggregating Items

**Yang Zhou**[1,2], Tong Yang[2], Jie Jiang[2], Bin Cui[2], Omid Alipoufard[3], Minlan Yu[1], Xiaoming Li[2], Steve Uhlig[4]

Harvard University[1], Peking University[2], Yale University[3], Queen Mary University of London[4]

# Data Streams are Pervasive


Network traffic eth0 - KiB/s
Thu 14:50:00 - 14:55:00 +00:00
Net. traffic inbound: eth0: 3.89 KiB/s
Net. traffic outbound: eth0
Net. traffic inbound: eth0



Network traffic     Video streaming     Sensor data     Web click data     (etc.)

In many applications, some statistical information is needed !

Applications: Network measurement, DBMS optimization, Search engine design, Security, etc.
Information required: flow size, heavy hitters, heavy changes, quantiles, etc.

## Accurate and Fast Data Stream Analysis is Challenging

Challenges:

1. Memory constraint
   - Fit into cache to boost speed
   - Hardware on-chip memory limited
2. Single-pass requirement
   - Data is of huge volume and fast speed: Dumping into disk is hard
   - Some applications need online analysis

Exact statistics (e.g., by using hash tables)
are difficult to obtain (and often unnecessary) !

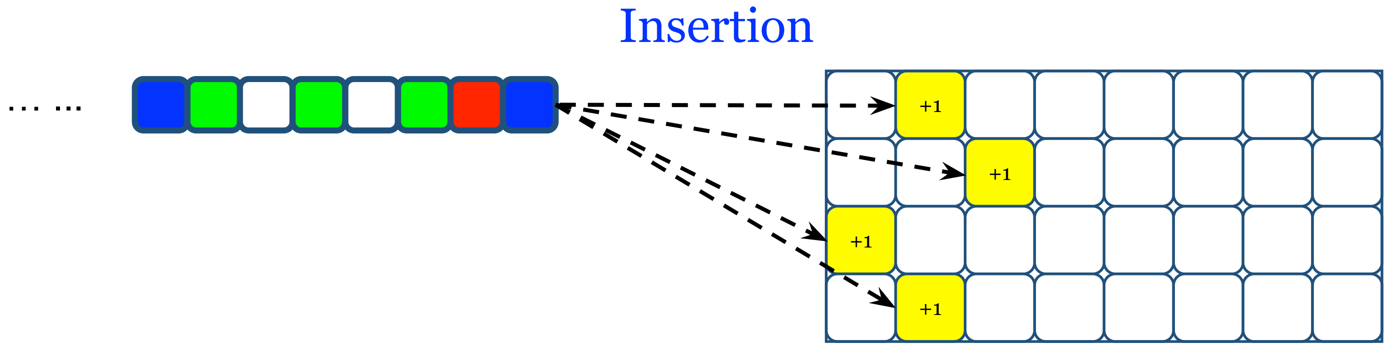
# Data Sketches can Help

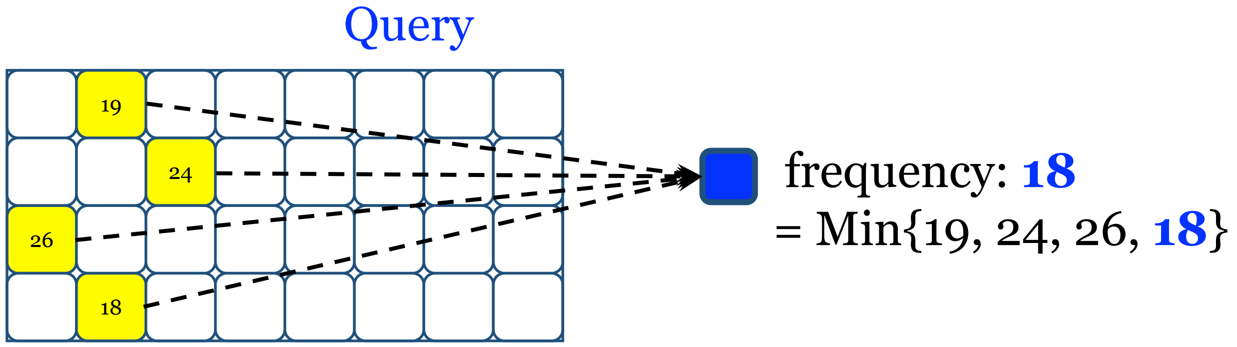
| Tasks | Data Sketch Algorithms |
|---|---|
| Frequency estimation | Count-Min, CM-CU, Count, ASketch |
| Top-k Hot items | Count-Min, CM-CU, Space-Saving ASketch, FlowRadar, UnivMon |
| Heavy changes | RevSketch, FlowRadar, UnivMon, Space-Saving |
| Superspreader /DDoS detection | TwoLevel |
| Frequency distribution | MRAC, FlowRadar |
| Cardinality | FM, LC, UnivMon |
| Entropy | FlowRadar, UnivMon |

# Data Sketches can Help

| Tasks | Data Sketch Algorithms |
|---|---|
| Frequency estimation | Count-Min, CM-CU, Count, ASketch |
| Top-k Hot items | Count-Min, CM-CU, Space-Saving ASketch, FlowRadar, UnivMon |
| Heavy changes | RevSketch, FlowRadar, UnivMon, Space-Saving |
| Superspreader /DDoS detection | TwoLevel |
| Frequency distribution | MRAC, FlowRadar |
| Cardinality | FM, LC, UnivMon |
| Entropy | FlowRadar, UnivMon |

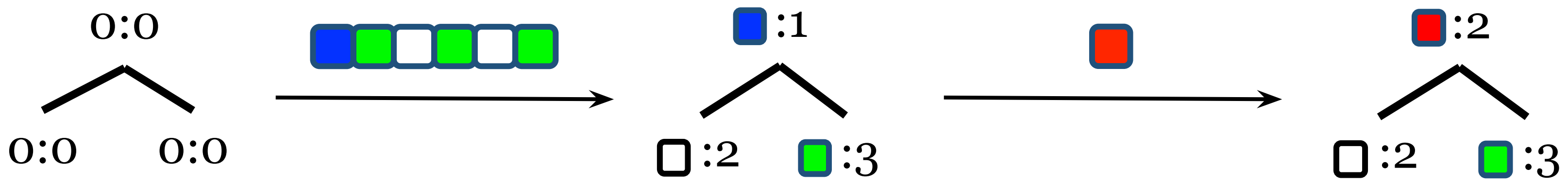
# Count-Min Sketch — Estimating Frequencies





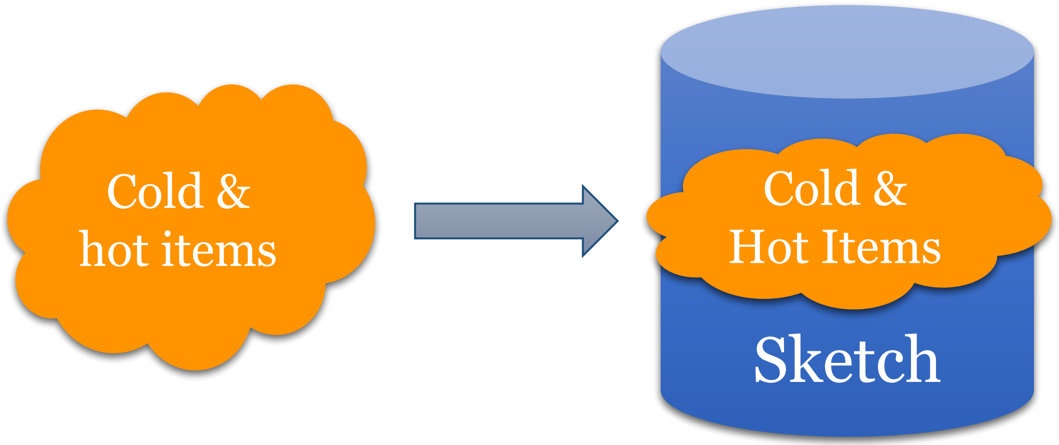
frequency: **18**
= Min{19, 24, 26, **18**}

# Space-Saving — Finding Top-k Hot Items



- Maintaining a heap-like data structure.
- If Space-Saving is full, the smallest item will be replaced by the new item, whose frequency is initialized to be $f_{min}$+1

# Limitations of Conventional Data Sketches



Real Data Streams :

Highly skewed

-> Majority: Cold items

-> Minority: Hot items

Count-Min :

All items use large counters

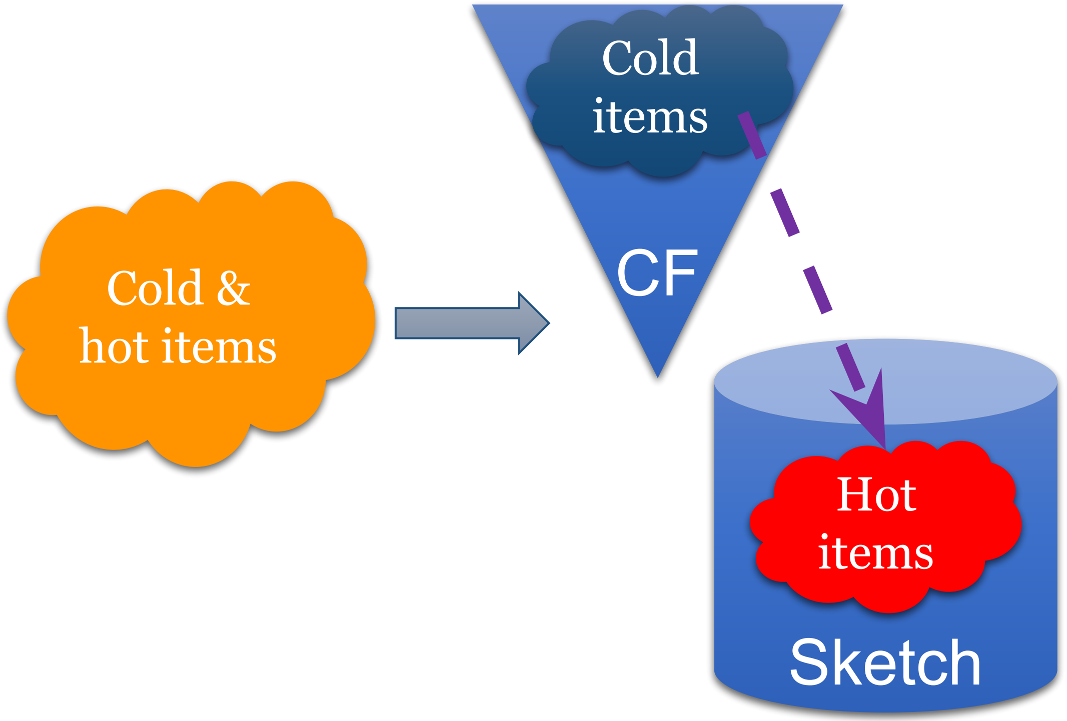
-> A waste of memory

Space-Saving :

A great many of replacements caused by cold items are unnecessary

-> poor accuracy

# Methodology of Cold Filter*



Count-Min:

Use small counters in CF

-> record cold items

Use large counters in sketch

-> record hot items

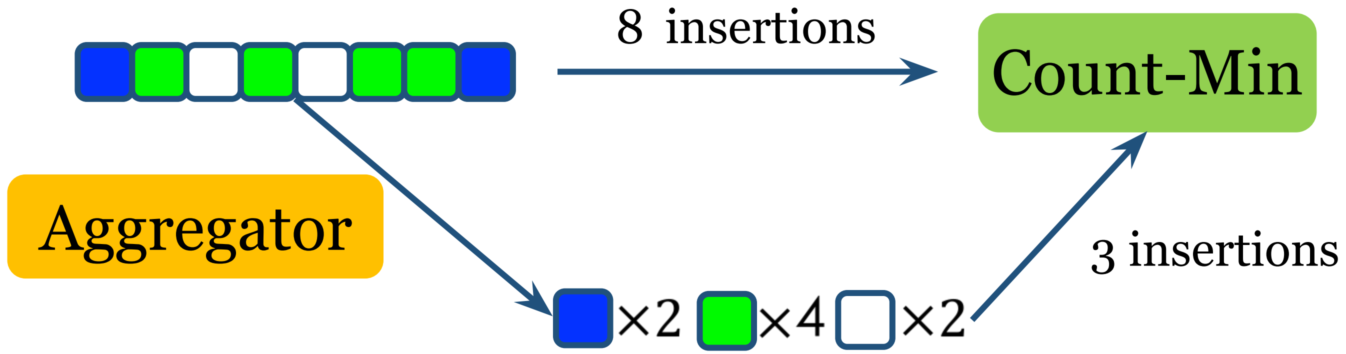
Space-Saving:

CF filters many cold items

-> reduce # unnecessary replacements

*Cold Filter: A Meta-Framework for Faster and More Accurate Stream Processing.* Yang Zhou, Tong Yang, Jie Jiang, Bin Cui, Minlan Yu, Xiaoming Li and Steve Uhlig. SIGMOD. Jun. 2018

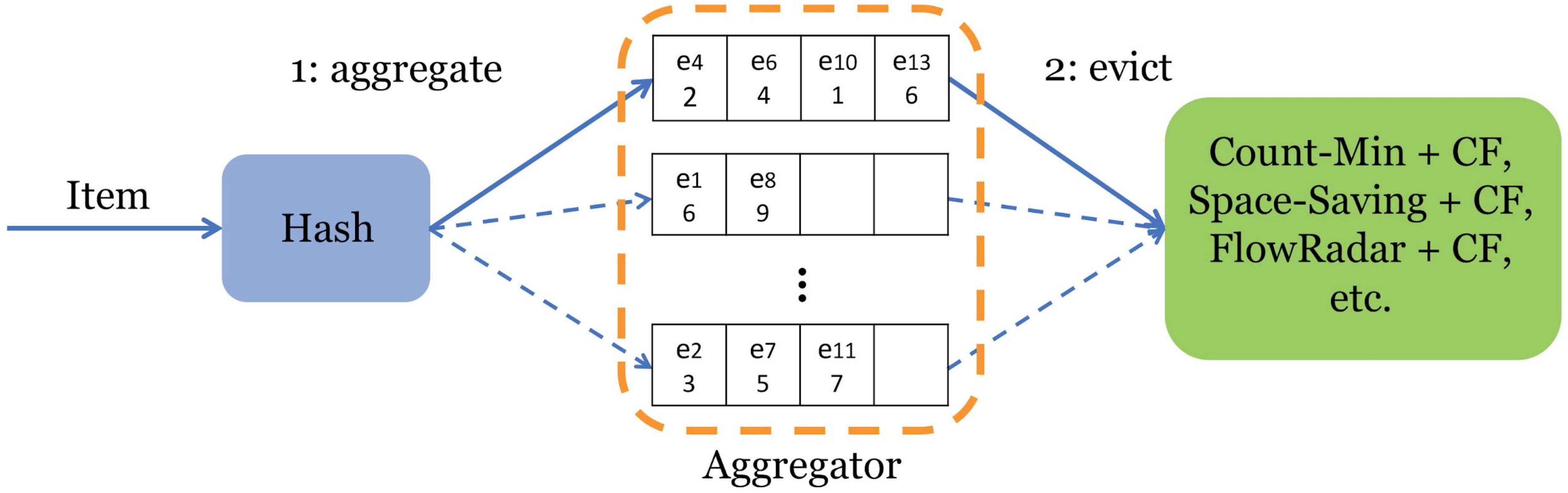
# Agg-Evict: Optimizing Speed



Ideally, 8/3=2.67 speed-up

-> How to design an efficient Aggregator ?

## Design of Agg-Evict[*]



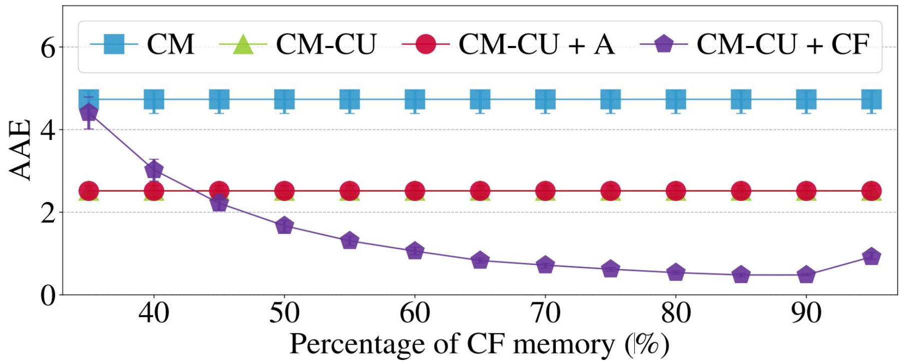
1.Using SIMD to query continuous cells in a K-V pair array

2.Using Random Eviction for simplicity and speed

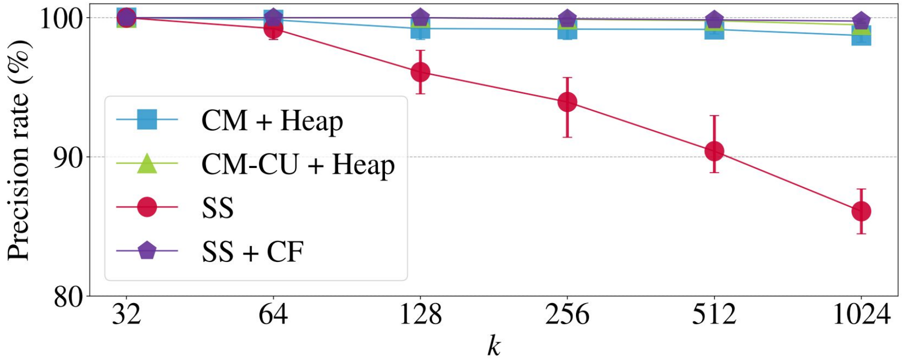
[*] *Accelerating Network Measurement in Software*. Yang Zhou, Omid Alipoufard, Minlan Yu and Tong Yang. ACM SIGCOMM Computer Communication Review. 2018

# Accuracy Improvement

Average Absolute Error



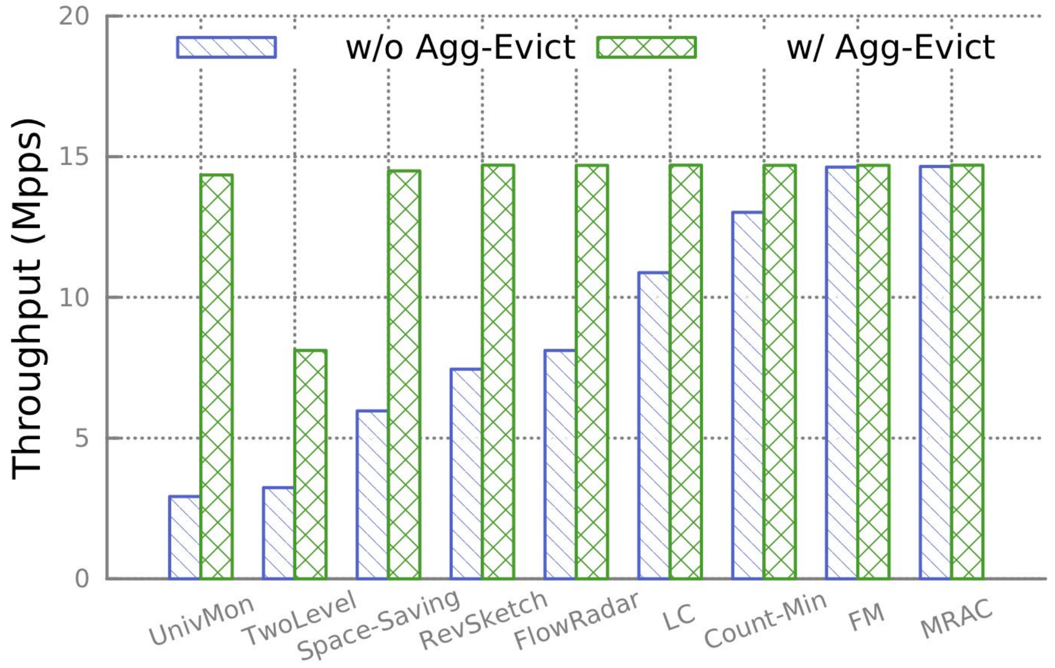
Frequency estimation: Varying the CF size



Finding Top-k hot items: Varying k

All algorithms use the same memory size

## Speed Improvement



## Conclusion

| | |
|---|---|
| Cold Filter | Improving accuracy by filtering the cold |
| Agg-Evict | Improving speed by aggregating items |
| Generic | Applicable to many different data sketches |

# Thanks!

Source Code: https://github.com/zhouyangpkuer/ColdFilter,

https://github.com/zhouyangpkuer/Agg-Evict.