OctoSketch: Enabling Real-Time, Continuous Network Monitoring over Multiple Cores

Yinda Zhang, Peiqing Chen, Alan Zaoxing Liu

UNIVERSITY OF MARYLAND

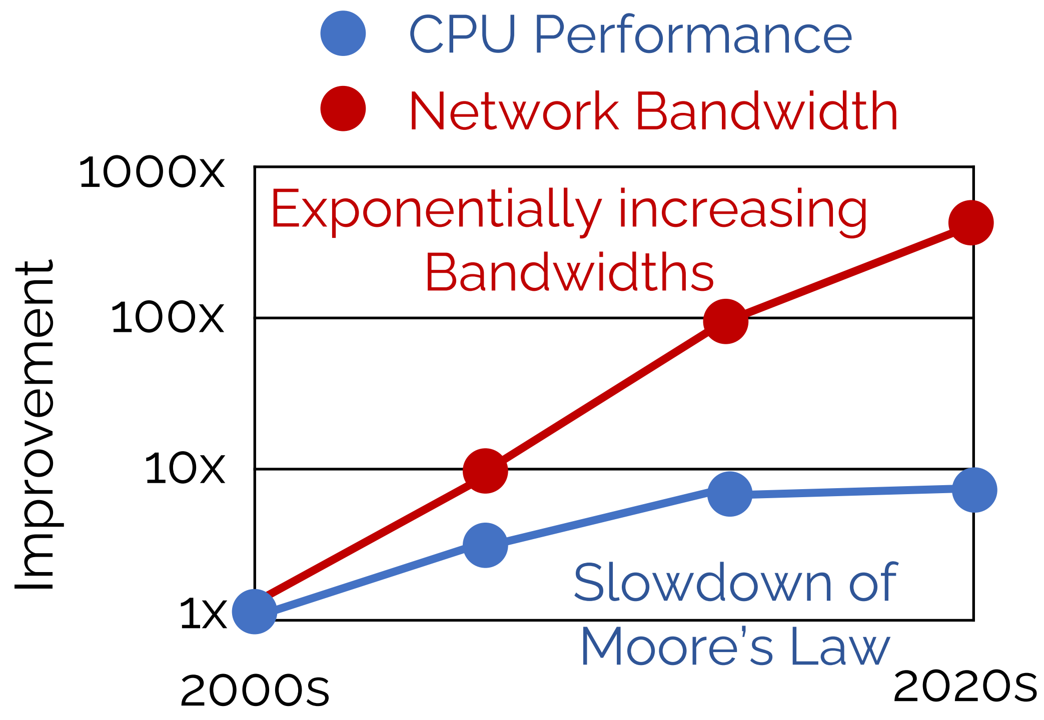
Penn
UNIVERSITY of PENNSYLVANIA

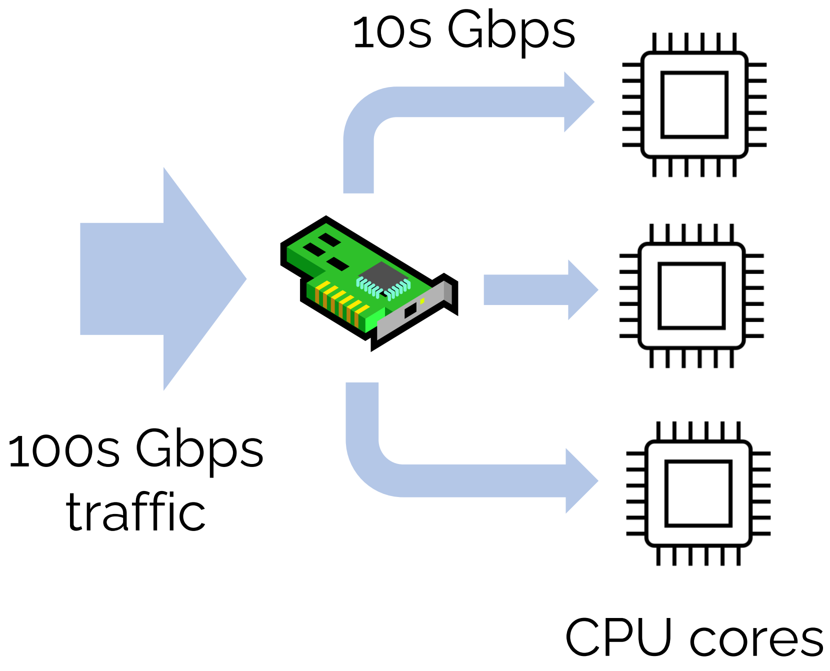
Packet processing needs multiple CPU cores

OvS Open vSwitch

fd.io The Universal Dataplane

BESS



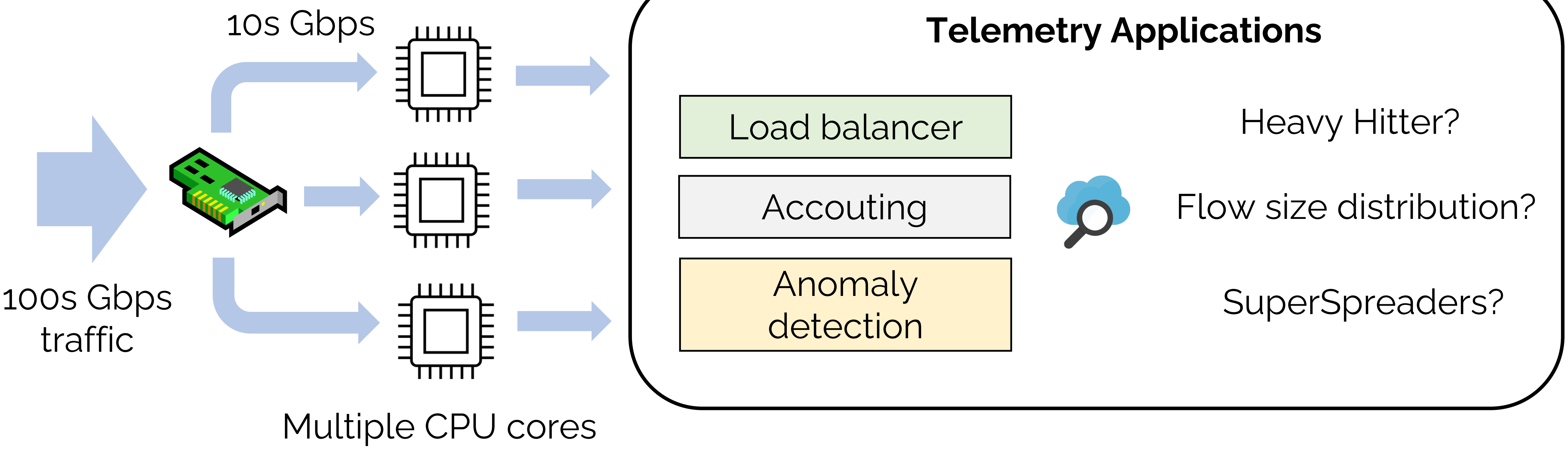


Need for network telemetry over multiple cores

OvS Open vSwitch

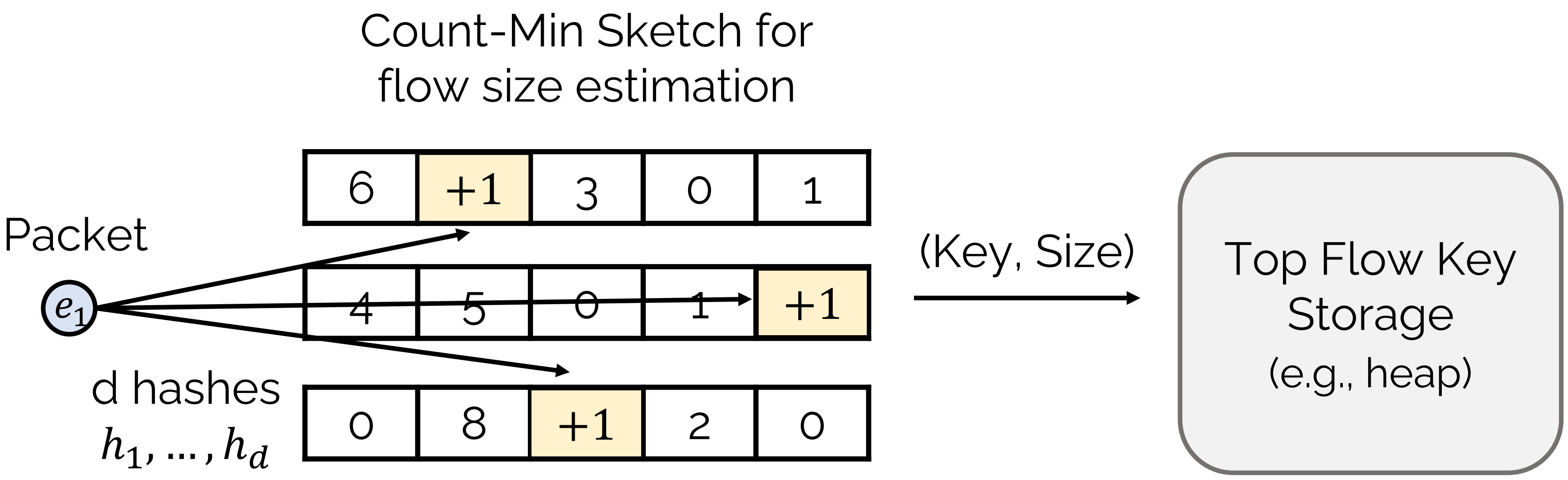
fd.io The Universal Dataplane

BESS



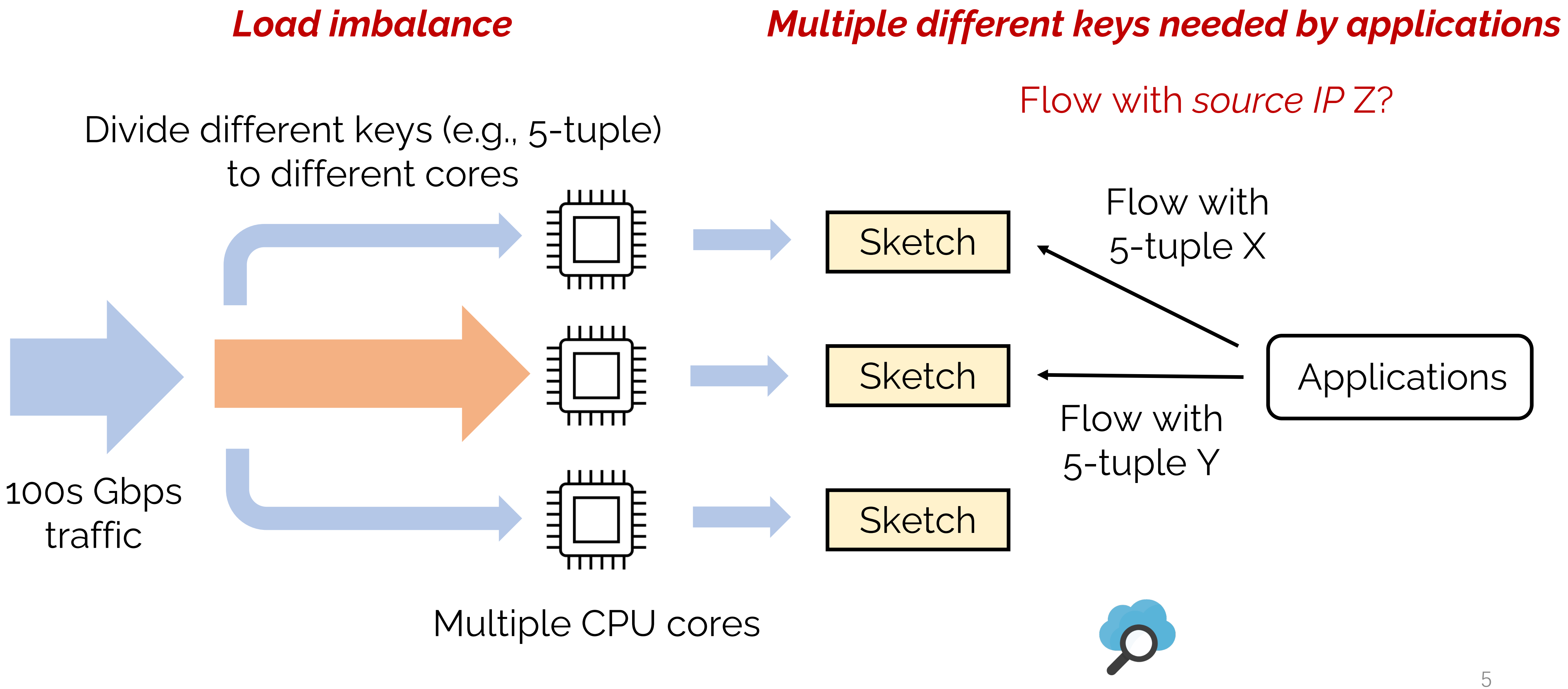
Sketches: promising solutions for *single-core telemetry*

(UnivMon [SIGCOMM'16], NitroSketch [SIGCOMM'19], CocoSketch [SIGCOMM'21])



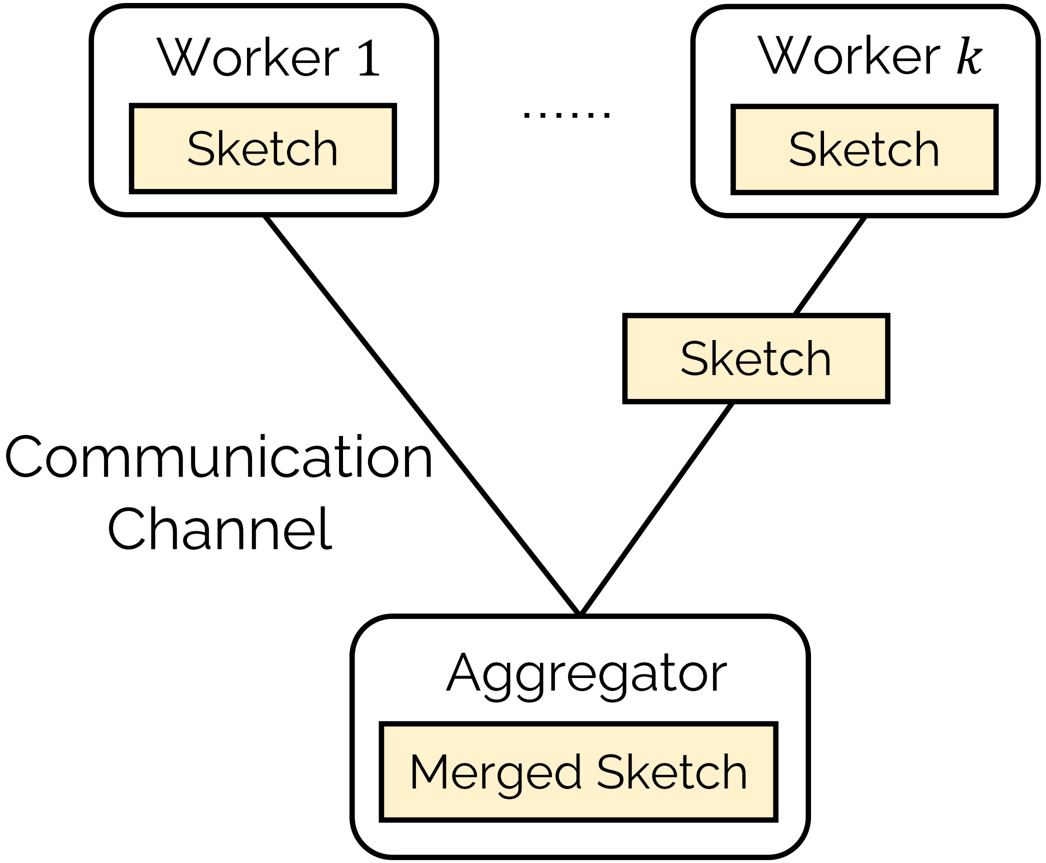
(1) **high** throughput (2) **small** memory usage (3) **bounded** error rates

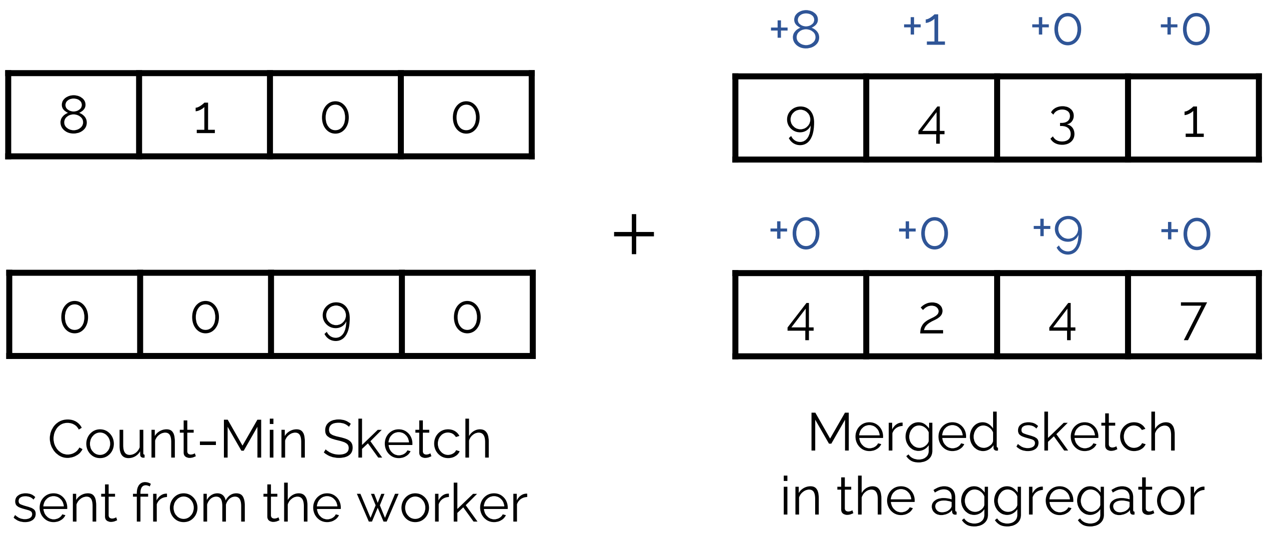
Strawman solution for *multi-core telemetry*: *Key-based Partition*



Existing solution for *multi-core telemetry*: *Entire-sketch-merge*

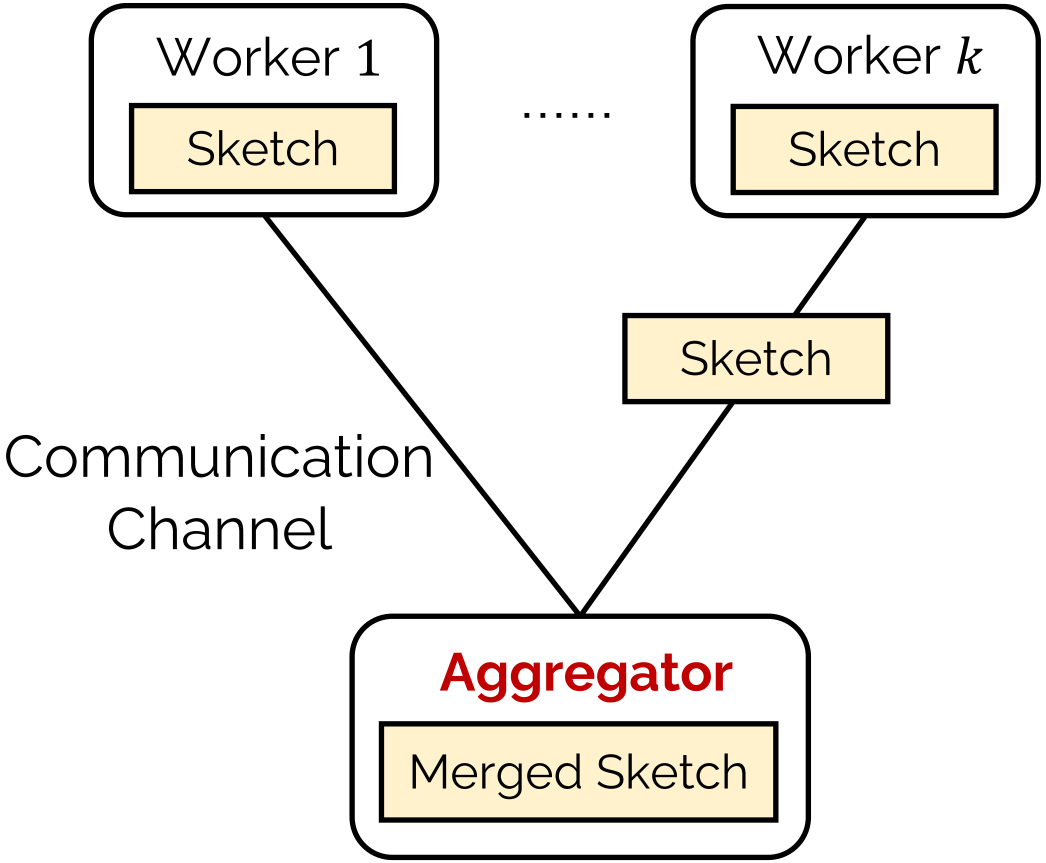
(UnivMon [SIGCOMM'16], Elastic Sketch [SIGCOMM'18], HeteroSketch [NSDI'22])



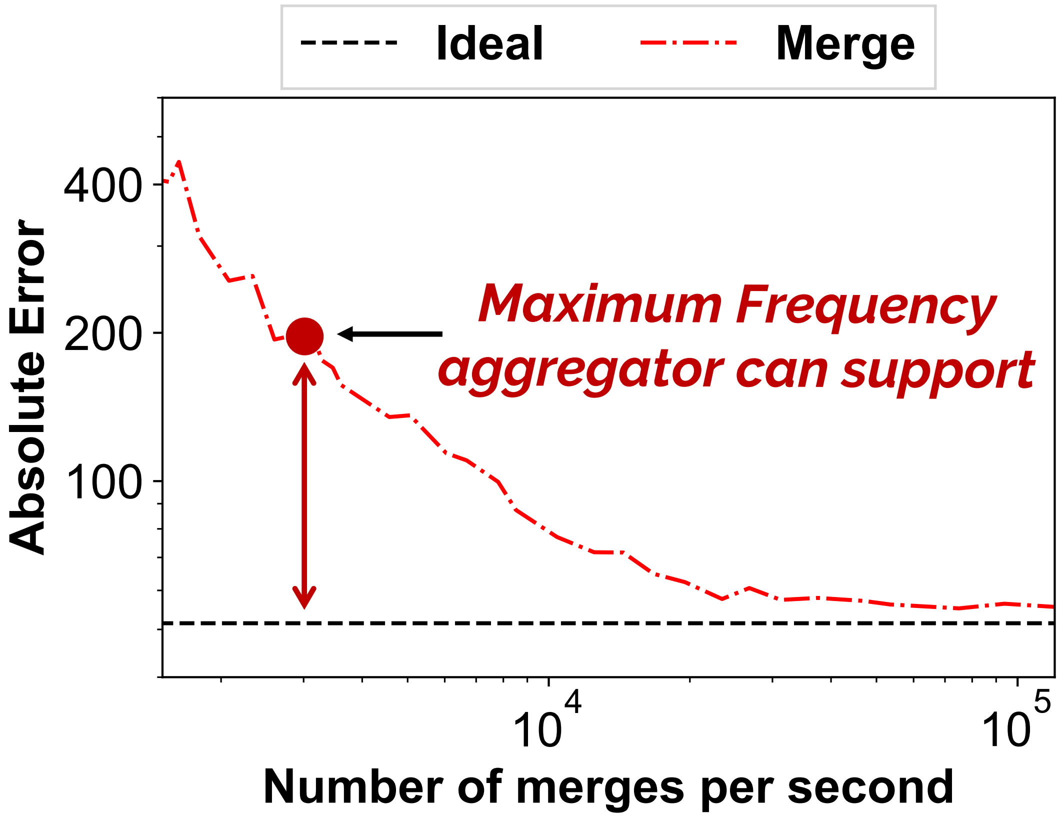


Merge process

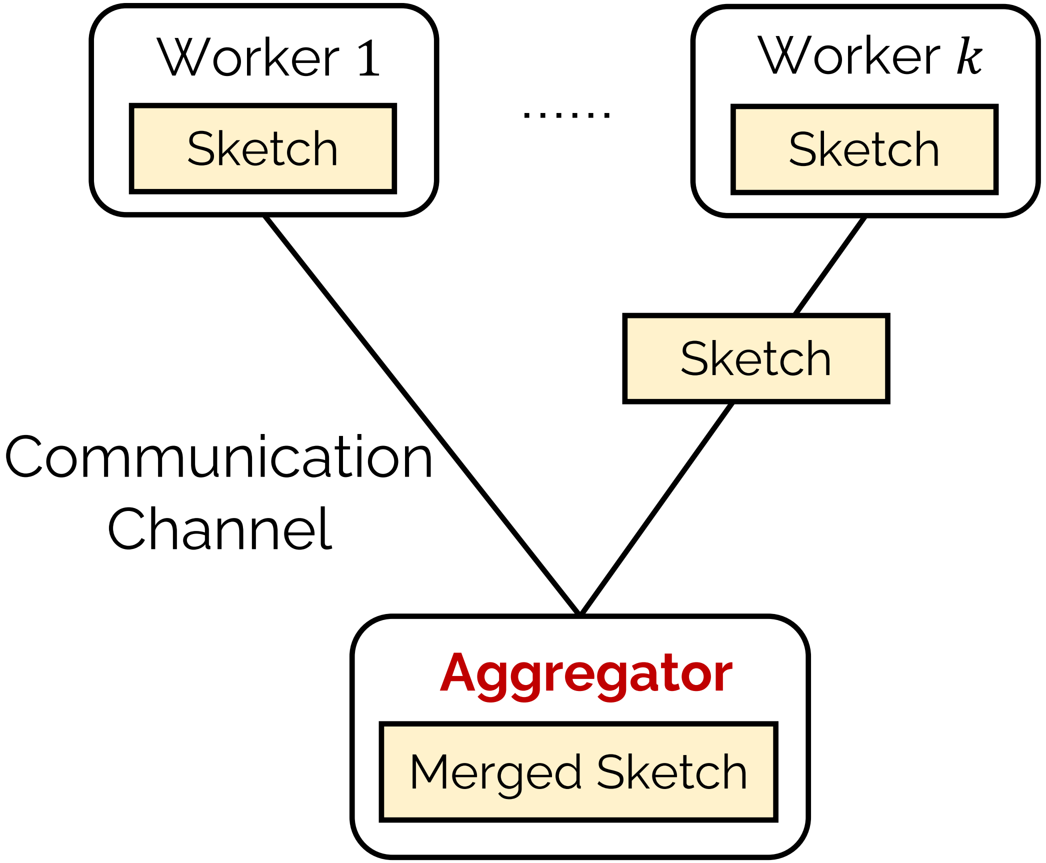
Bottleneck In the aggregator



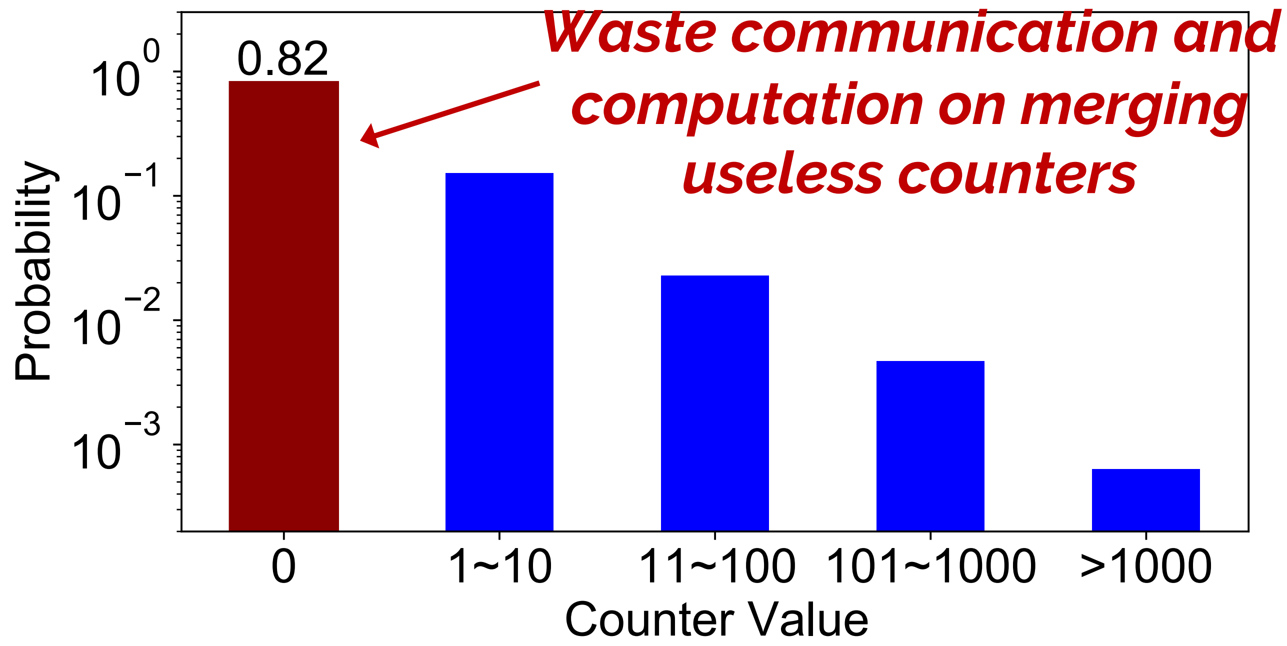
Ideal accuracy: the accuracy of sketch that works in a single core and measures the whole traffic



Bottleneck In the aggregator



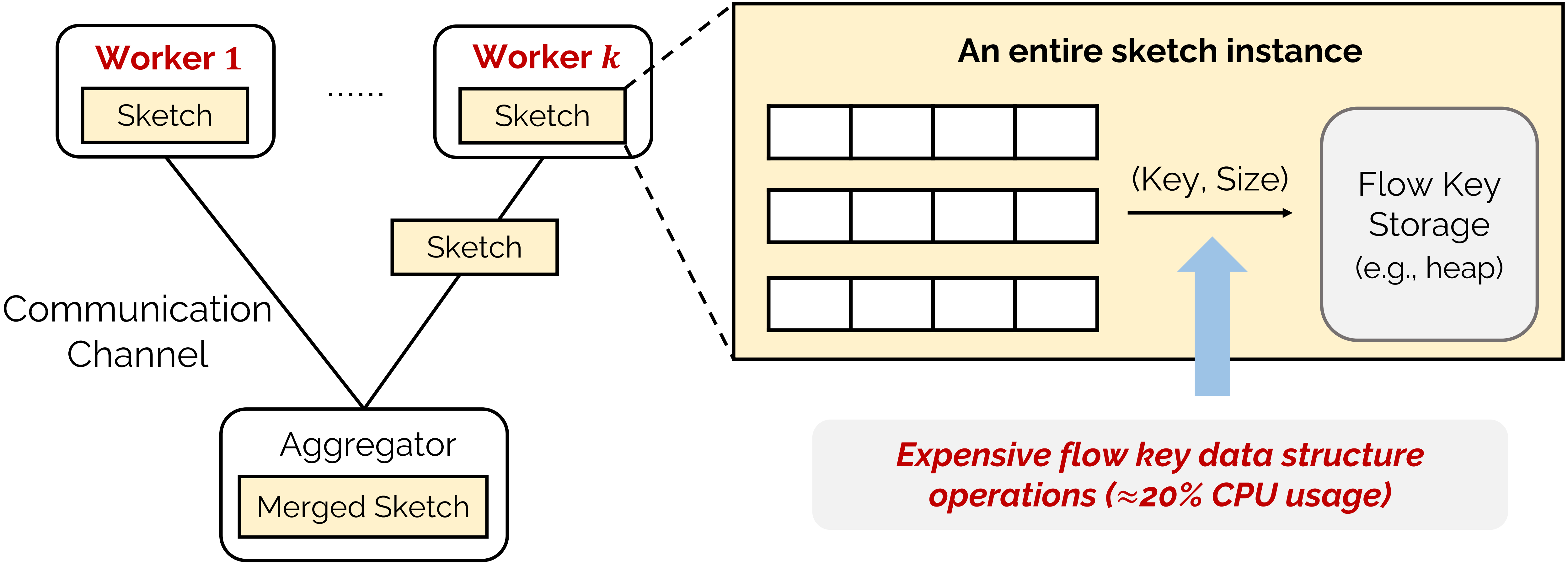
Counter value distribution when sketch-merge



Heavy-tailed network traffic leads to **heavy-tailed counter values** in the sketch

Bottleneck in the workers

(Elastic Sketch [SIGCOMM'18], NitroSketch [SIGCOMM'19], CocoSketch [SIGCOMM'21])



Expensive flow key data structure operations (≈20% CPU usage)

OctoSketch: Real-Time multi-core monitoring

A sketching framework for multicore monitoring that simultaneously has:

- **Online accuracy**: accuracy guarantees at any query time
  - Idea 1: Only send “sufficiently changed” counters
- **Adaptive**: adaptive to packet arrival rate and system objectives
  - Idea 2: Dynamic resource allocation based on queue length
- **Performance**: line-rate (e.g., 100G) with minimal CPU and memory
  - Idea 3: Remove redundant data structures

OctoSketch: Real-Time multi-core monitoring

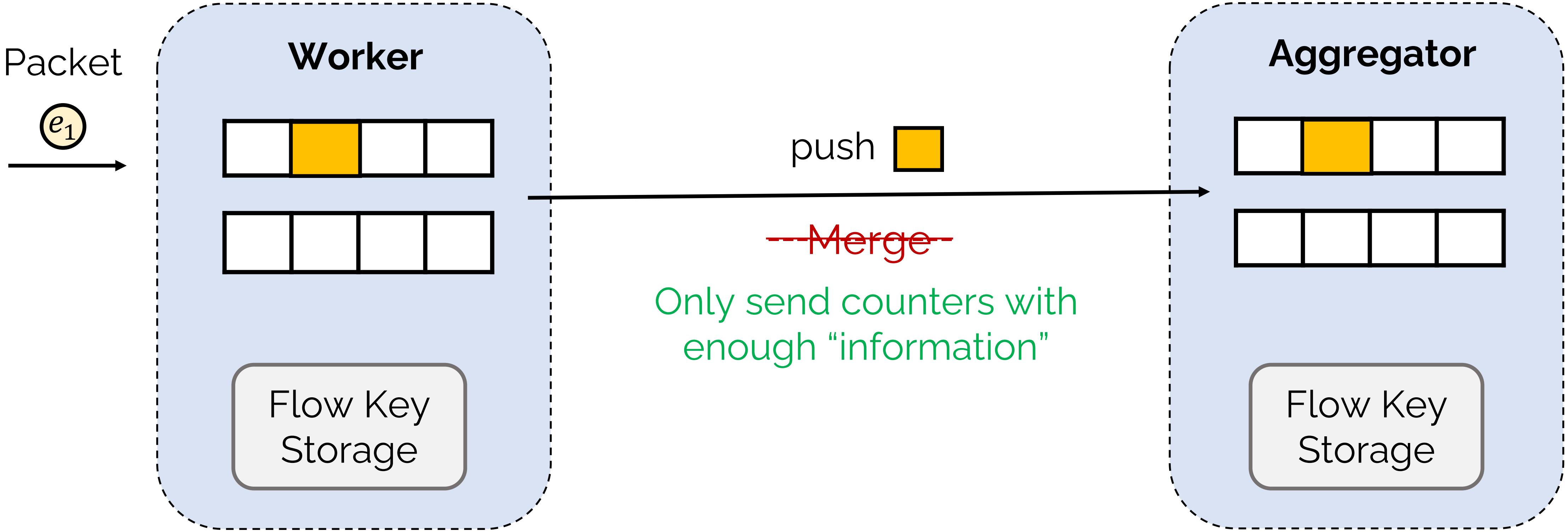
A sketching framework for multicore monitoring that simultaneously has:

- **Online accuracy**: accuracy guarantees at any query time
  - Idea 1: Only send "sufficiently changed" counters
- **Adaptive**: adaptive to packet arrival rate and system objectives
  - Idea 2: Dynamic resource allocation based on queue length
- **Performance**: line-rate (e.g., 100G) with minimal CPU and memory
  - Idea 3: Remove redundant data structures

Key Idea: Only send “sufficiently changed” counters

~~A large sketch merging operation~~

A series of small counter change notification



General to different sketches

- **Applied to 9 representative sketches over 6 different tasks**
- Cardinality-related sketches
  - e.g., HyperLogLog
- Counters with flow keys
  - e.g., CocoSketch
- Negative counter values
  - e.g., UnivMon

Check out our paper for more details!

Benefit of the continuous, change-based mechanism

- Retains the **same asymptotic error bounds as in the ideal case** in which traffic is not distributed **at any query time**.
- Offers accuracy guarantees for a variety of measurement tasks.
  - e.g., finding heavy hitters, estimating cardinality

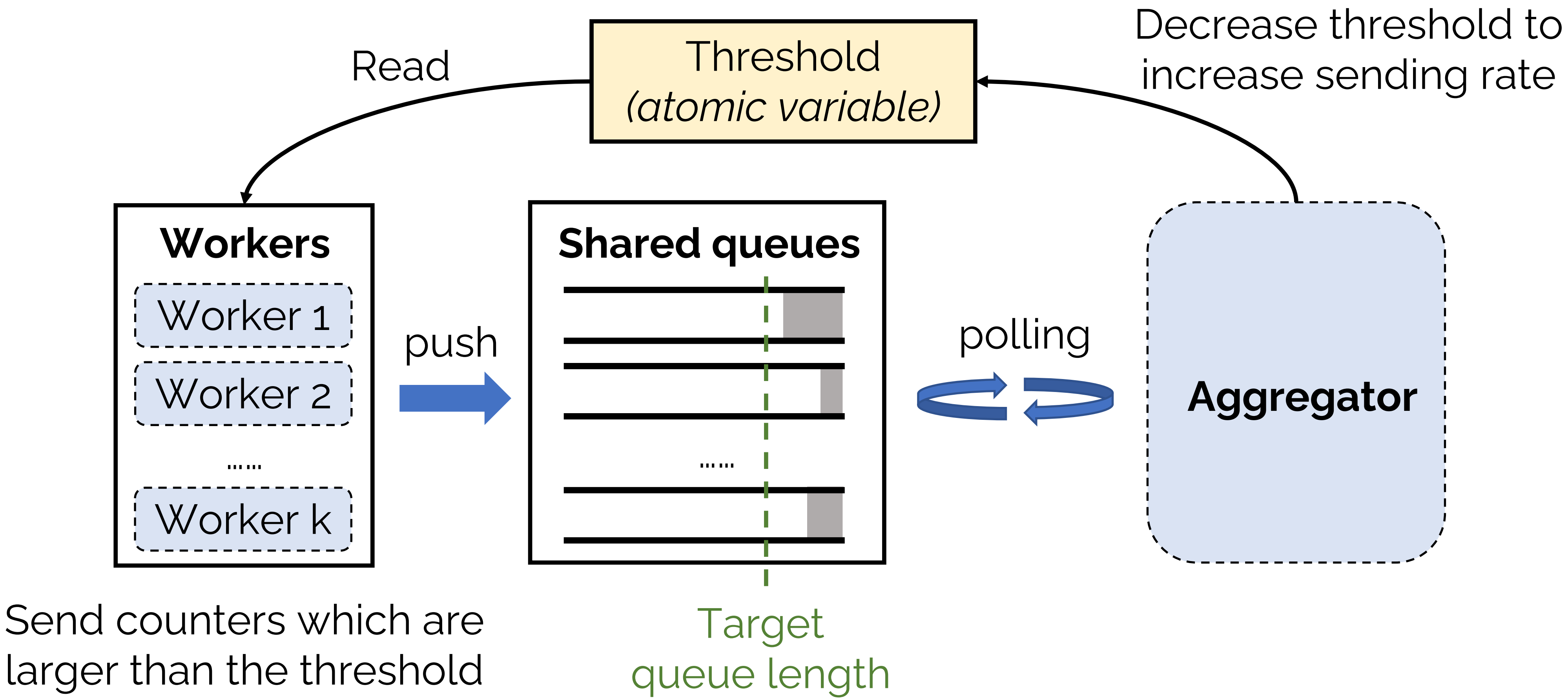
OctoSketch: Real-Time multi-core monitoring

A sketching framework for multicore monitoring that simultaneously has:

- **Online accuracy**: accuracy guarantees at any query time
  - Idea 1: Only send "sufficiently changed" counters
- **Adaptive**: adaptive to packet arrival rate and system objectives
  - Idea 2: Dynamic resource allocation based on queue length
- **Performance**: line-rate (e.g., 100G) with minimal CPU and memory
  - Idea 3: Remove redundant data structures

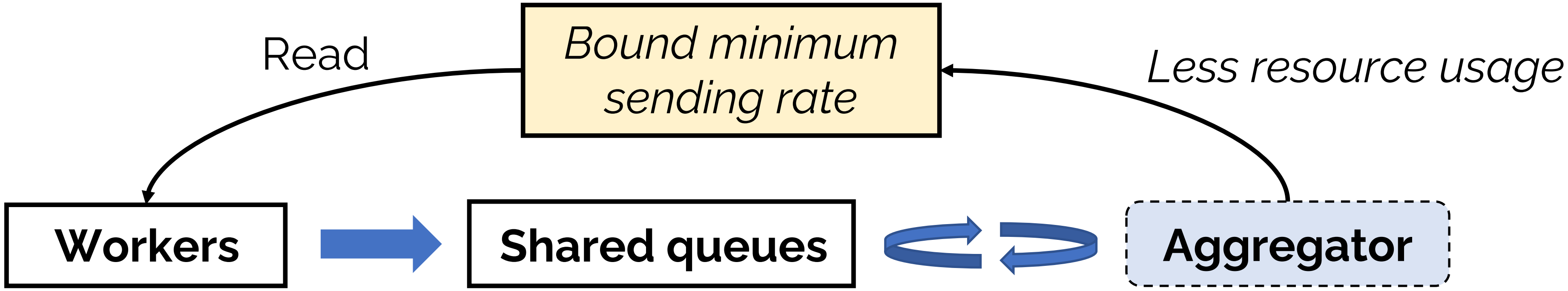
Idea 2: Adaptive sending rate

- Packet rate is low ⇒ Send changes frequently (high online accuracy)



Policies to meet various objectives

- Scenario 1: Given resource budget, achieve best possible online accuracy
  - Given *70% CPU usage* for the aggregator, modify the threshold to *optimize the online accuracy*
- Scenario 2: Given accuracy target, achieve minimum resource usage
  - Given a *99% accuracy target*, bound the minimum sending rate to *free up extra computation resources*



OctoSketch: Real-Time multi-core monitoring

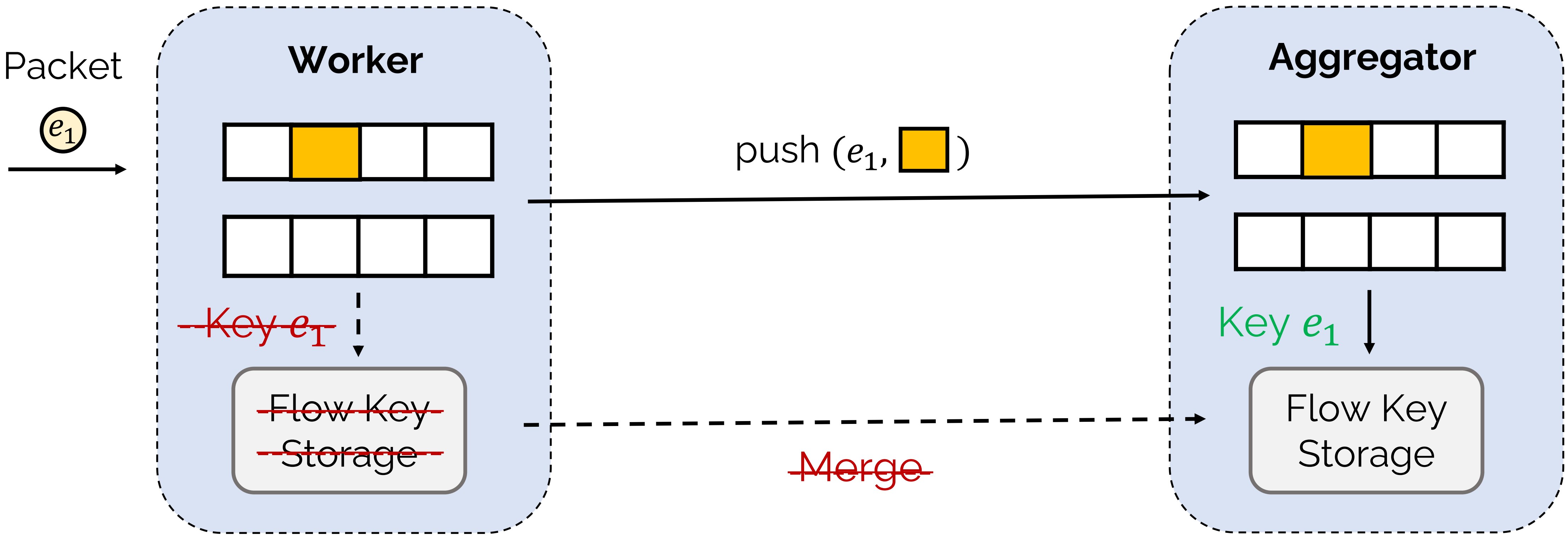
A sketching framework for multicore monitoring that simultaneously has:

- **Online accuracy**: accuracy guarantees at any query time
  - Idea 1: Only send “sufficiently changed” counters
- **Adaptive**: adaptive to packet arrival rate and system objectives
  - Idea 2: Dynamic resource allocation based on queue length
- **Performance**: line-rate (e.g., 100G) with minimal CPU and memory
  - Idea 3: Remove redundant data structures

Idea 3: Remove redundant data structures

~~Expensive flow key data structure operations~~

Remove flow key storage in workers

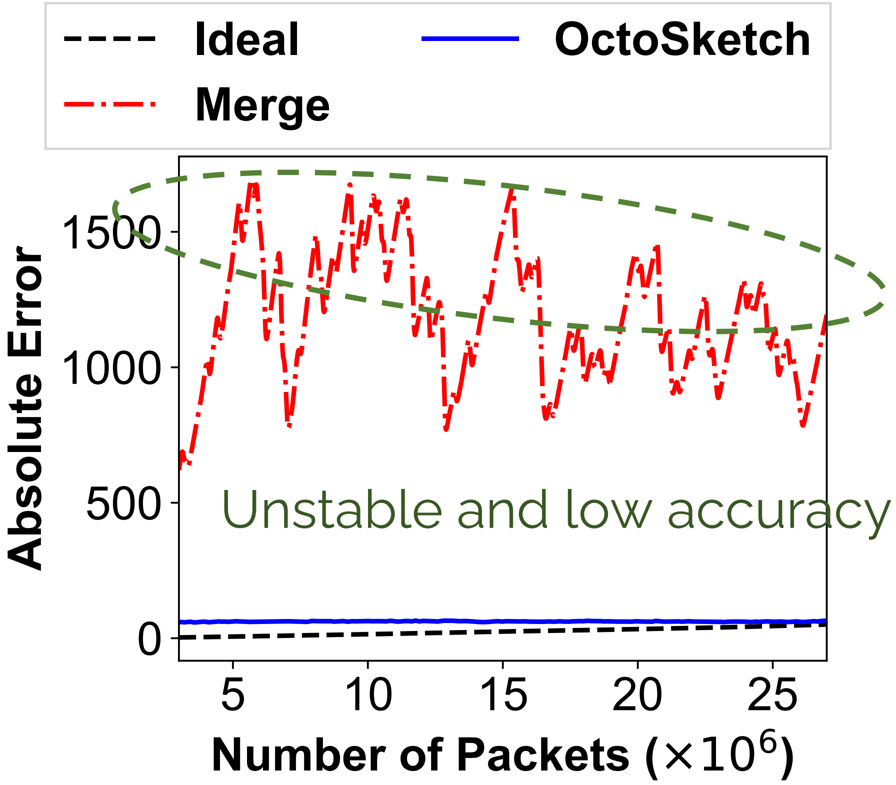


Evaluation setup

- **Use cases:** load balancer, key-value cache
- **Platforms:** CPU, DPDK, eBPF XDP
  - 2-16 workers' CPU + 1 aggregator CPU
- **Baselines**
  - **Entire-sketch-merge:** operating at maximum frequency for merging
  - **Ideal accuracy:** the accuracy of the sketch that works in a single core and measures the whole traffic

OctoSketch can achieve high online accuracy

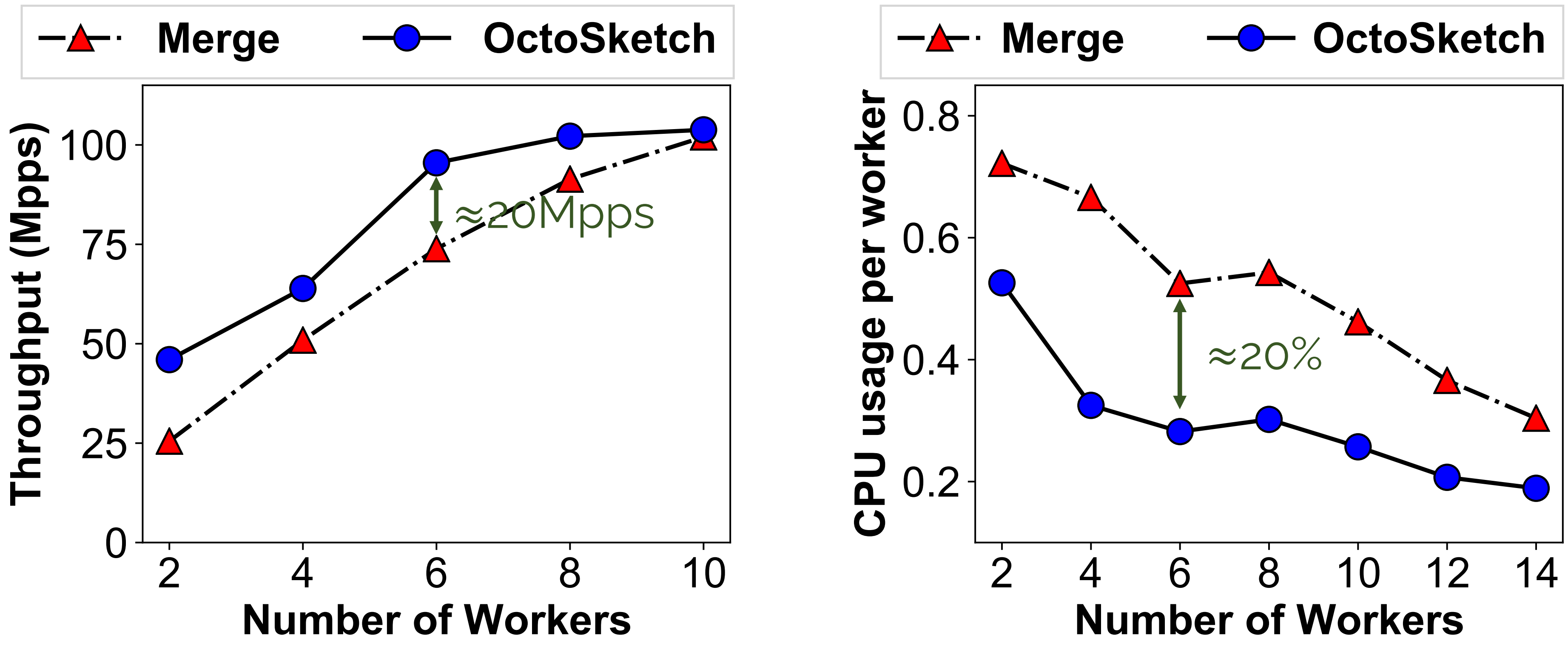
OctoSketch for the Count-Min sketch (Finding heavy hitters)



OctoSketch's *continuous mechanism* helps it maintain **accuracy close to the ideal one**

OctoSketch can achieve good CPU performance

OctoSketch for the Count-Min sketch in DPDK



Removing redundant data structures helps OctoSketch achieve both **high throughput** and **low CPU usage on workers**

OctoSketch is general to different sketches

Query Type	Sketch	Accuracy	Throughput
Flow Size	Count-Min	**9.32x**	**3.85x**
	Count Sketch	**9.04x**	**3.22x**
Cardinality	LogLog	**54.93x**	**1.29x**
	HyperLogLog	**38.97x**	**1.29x**
Super-Spreader	Locher Sketch	**4.06x**	**4.51x**
Quantile	DDSketch	**4.29x**	0.92x
Multi-Key	CocoSketch	**37.25x**	**1.01x**
Genreal	UnivMon	**13.55x**	**2.63x**
	ElasticSketch	**14.03x**	0.93x

Conclusions

- Multicore monitoring is needed
- Sketch-merge over multiple cores is impractical
- OctoSketch key ideas:
  - Continuous, change-based mechanism
  - Adaptive resource allocation
  - Remove redundant data structures
- OctoSketch achieves about 15.6x higher online accuracy and up to 4.5x higher throughput while retaining the generality

Source code: https://github.com/Froot-NetSys/OctoSketch