HeteroSketch

Coordinating Network-wide Monitoring in Heterogeneous & Dynamic Networks

Anup Agarwal (CMU), Zaoxing (Alan) Liu (BU),
Srinivasan Seshan (CMU)

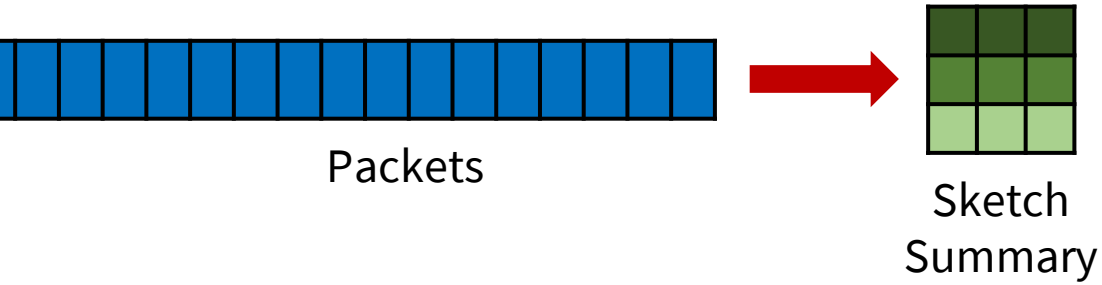
Carnegie Mellon University

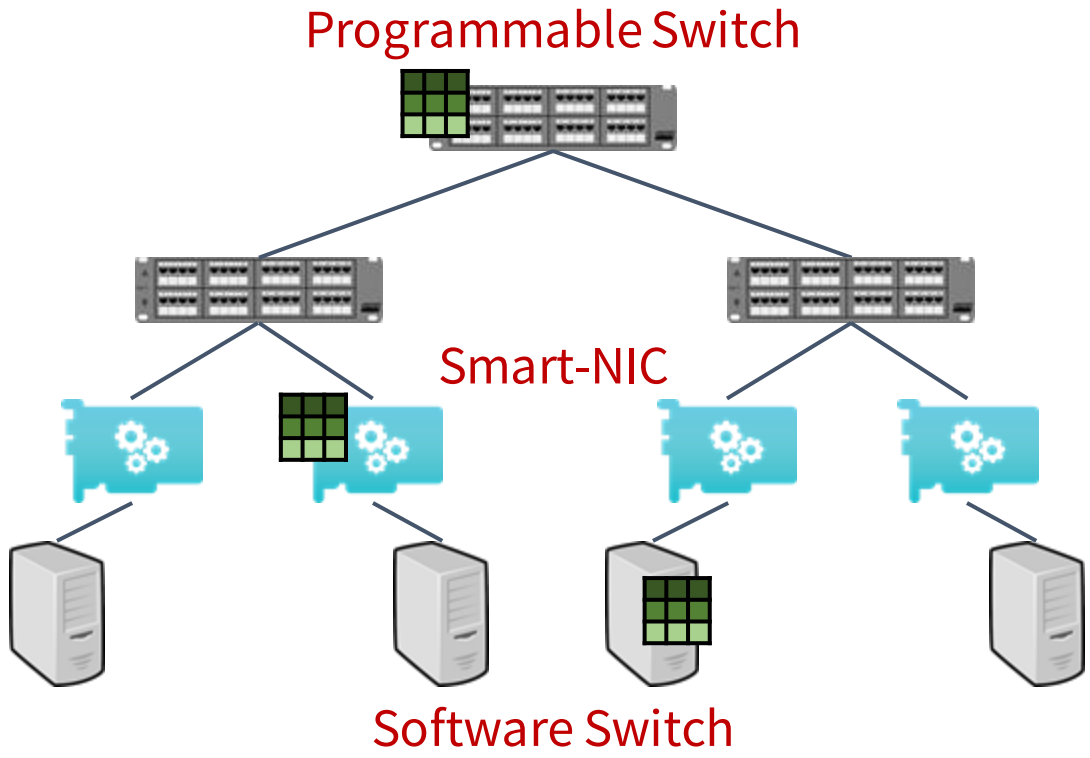
BOSTON UNIVERSITY

Advances: sketches & programmability

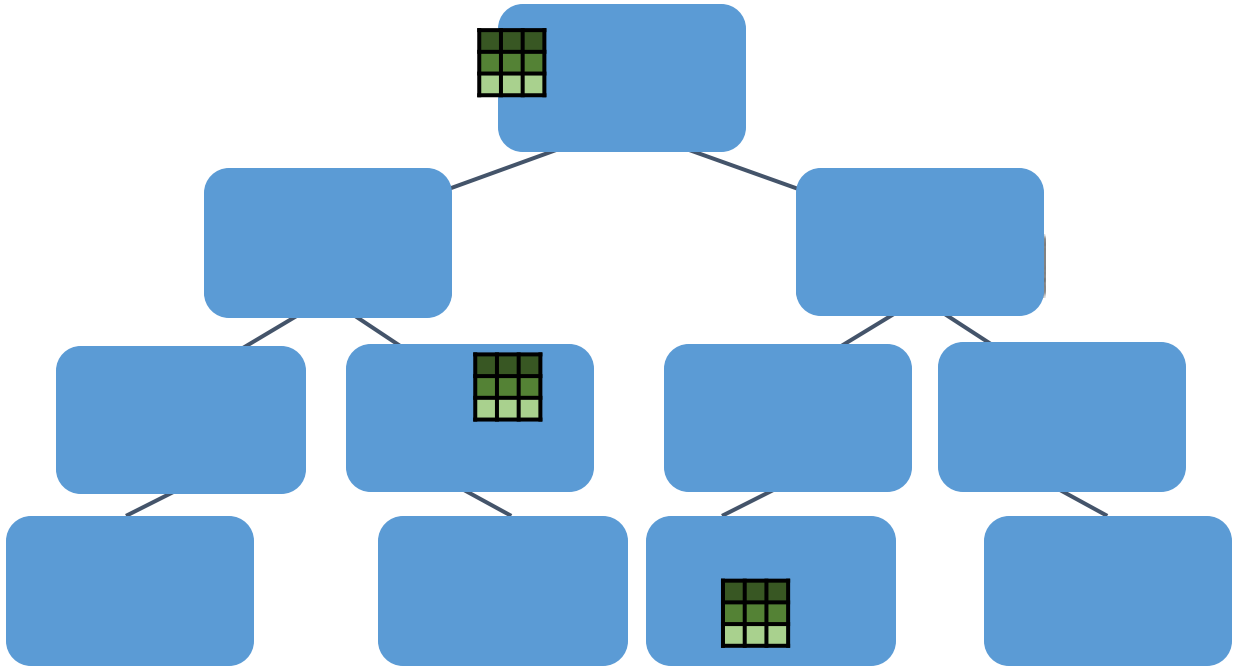
Sketches (data structures)	Deploy sketches anywhere

- Small (sub-linear) memory
- Accurate, Robust to workloads
- Variety of queries (Heavy-hitters, entropy, changes…)





Network-wide monitoring

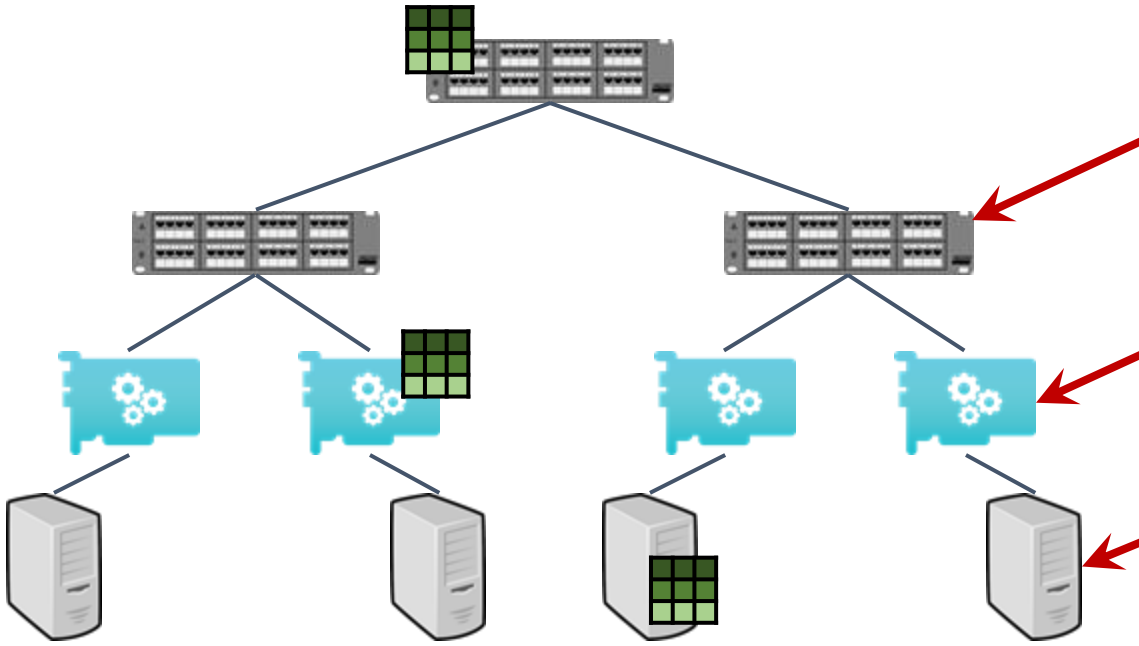


Coordinate *vantage points*:

- Cover traffic
- Ensure accuracy
- Meet memory constraints

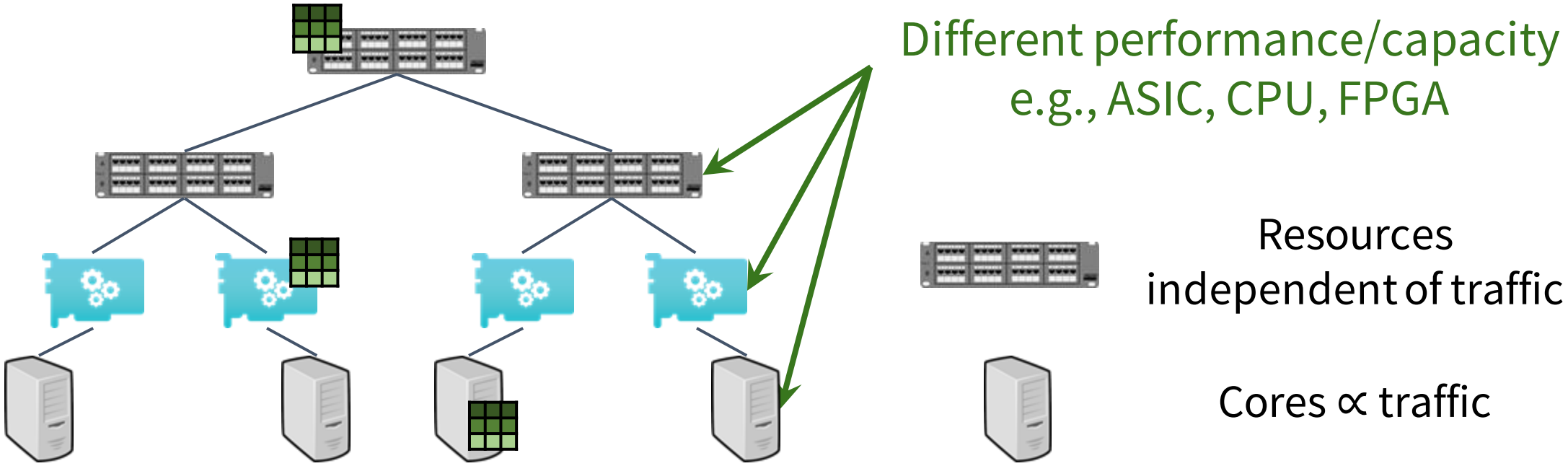
Assume network of homogeneous devices

Increasing trend towards heterogeneity



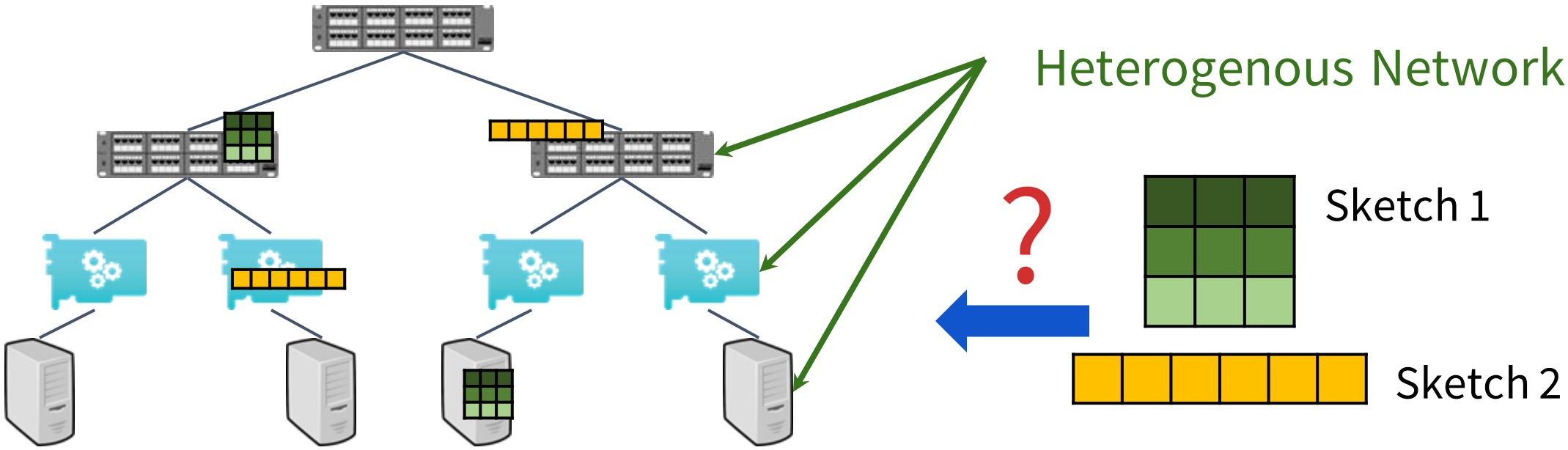
- Cetus. Alibaba [NSDI 22]
- Large-scale SmartNIC deployments. Azure [NSDI 18]
- Virtual software switches, VFP. Azure [NSDI 17]
- Push from vendors Nvidia Mellanox, Intel Barefoot, Netronome…

Overlooking heterogeneity is costly



- Existing works use **excessive resources** or **hamper performance**
- From evaluation: 1 extra core per server ⇒ **100k cores for 100k servers**

Place sketches & allocate resources?



System requirement:
Performance & resource efficiency, prompt responses (secs to mins)

HeteroSketch in a nutshell

Formulate as constrained optimization problem

Key Insights

1. Structure of sketches simplifies profiling ⇒ Cost/benefit analysis
2. Patterns in traffic and monitoring requirements ⇒ Quick responses

HeteroSketch Outline

1. Automated Profiler
2. Fast Optimizer

HeteroSketch in a nutshell

Formulate as constrained optimization problem

Key Insights

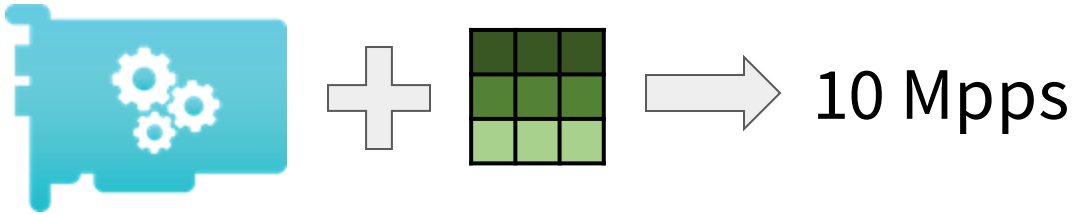
1. Structure of sketches simplifies profiling ⇒ Cost/benefit analysis
2. Patterns in traffic and monitoring requirements ⇒ Quick responses

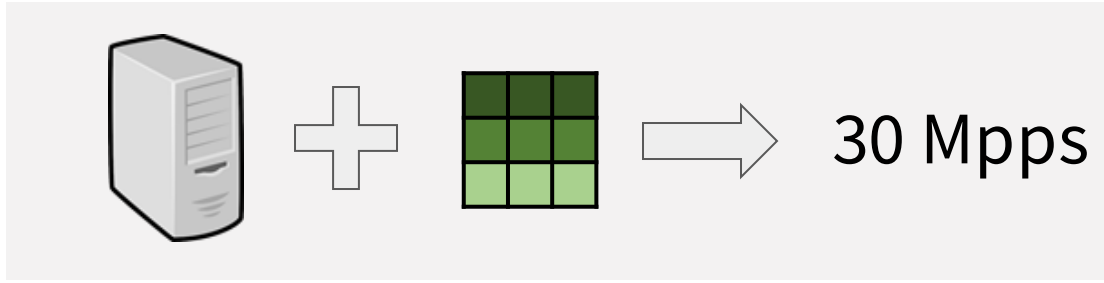
HeteroSketch Outline

1. **Automated Profiler**
2. Fast Optimizer

Profiler: Goal & Challenge

Hardware (Architecture, Resources) + Sketches (Params, Implementation) → Performance (Time per packet)





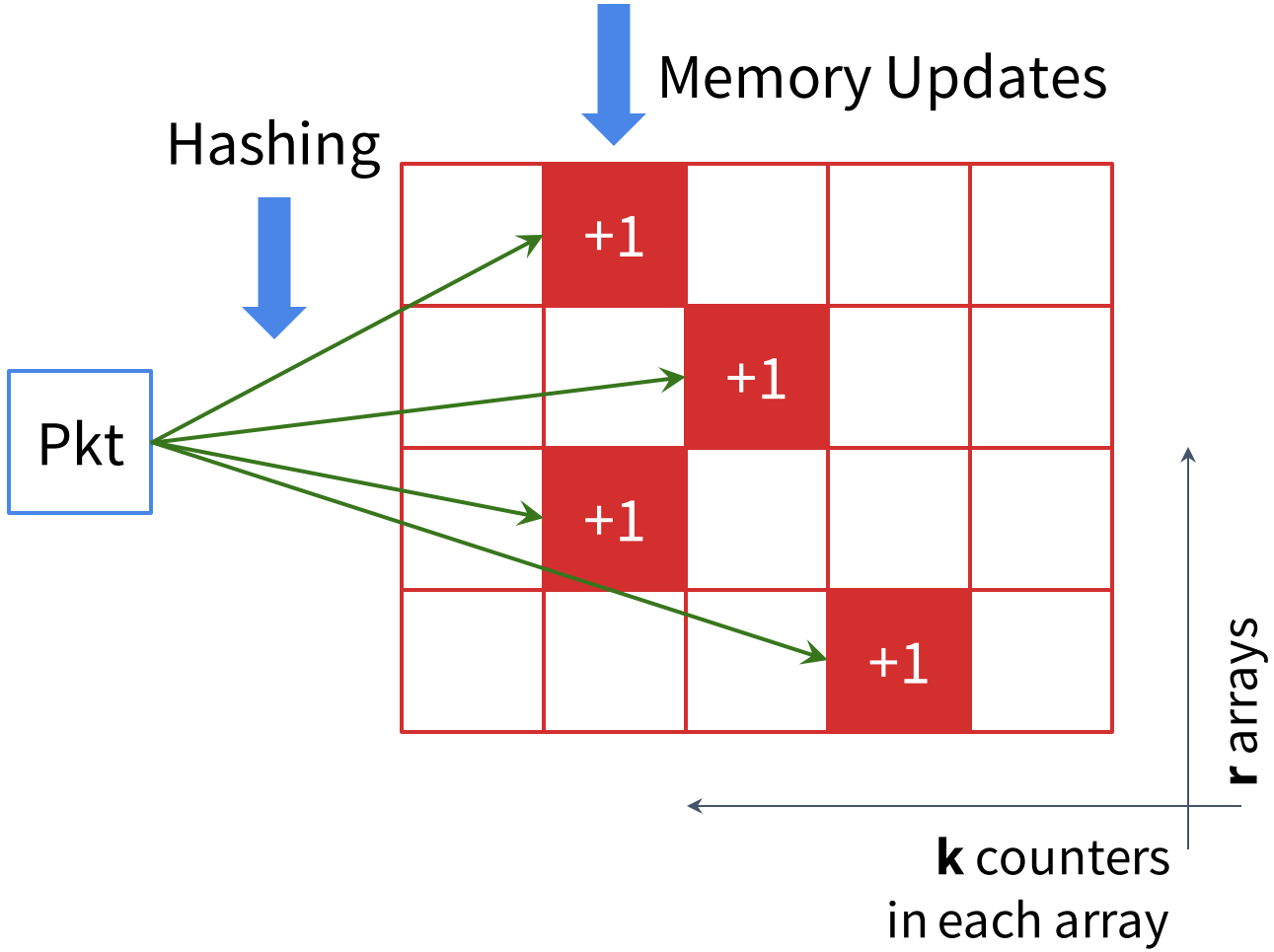
Performance prediction is hard in general

MonoTasks [SOSP17], Clara [HotNets20], SLOMO [SIGCOMM20]…

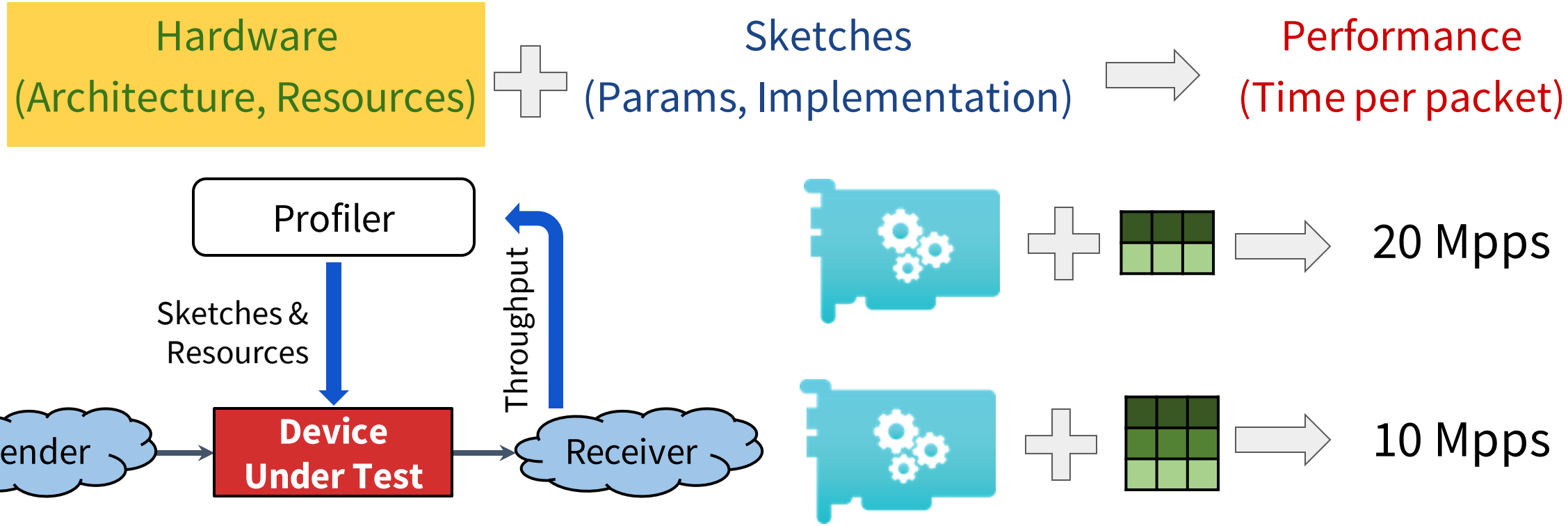
Sketch structure simplifies profiling

1) Many Independent primitive operations
2) Limited control flow and data dependencies.
3) Fixed Memory

⇒ Performance largely governed by number of primitive ops

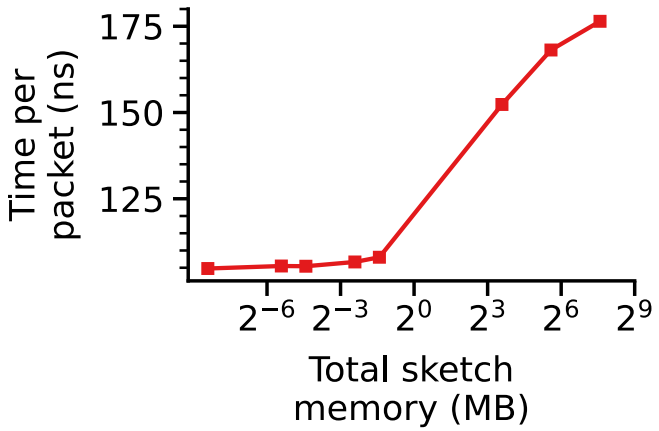


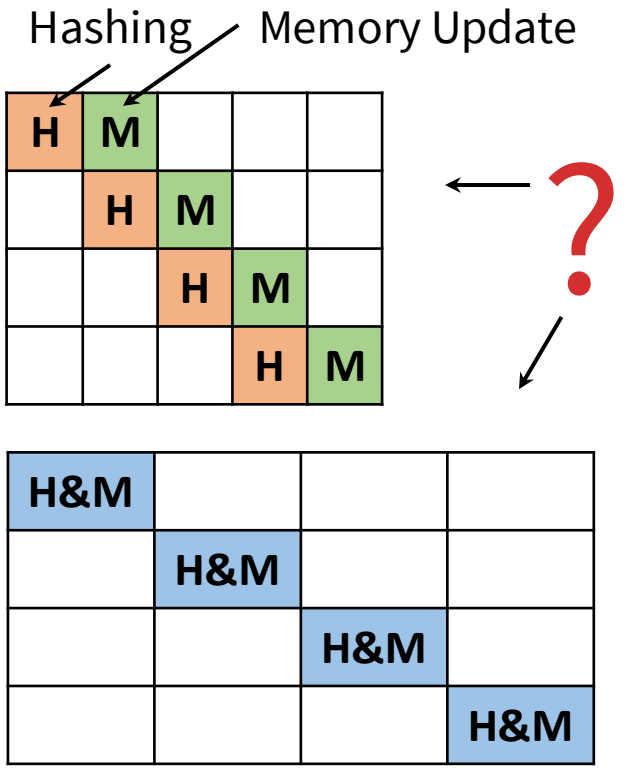
Micro-benchmarks: Device complexity

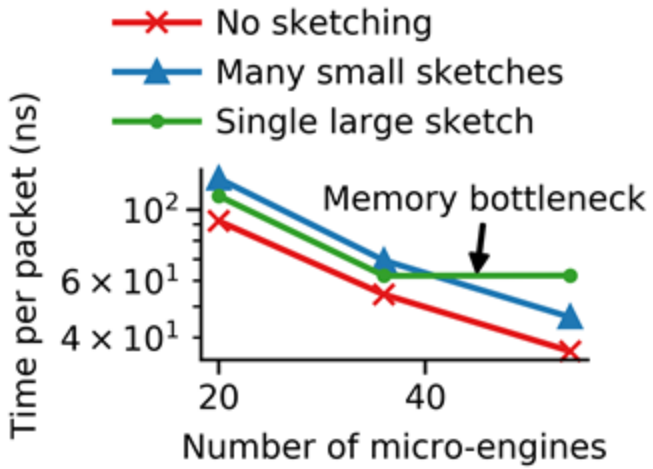


Device Agnostic. No intrusive measurements.

3 Phases: Micro-benchmark







1. Primitive Ops

Data-level parallelism

2. Compositions

Instruction-level parallelism

3. Resource Allocations

Thread-level parallelism

HeteroSketch in a nutshell

Formulate as constrained optimization problem

Key Insights

1. Structure of sketches simplifies profiling ⇒ Cost/benefit analysis
2. Patterns in traffic and monitoring requirements ⇒ Quick responses

HeteroSketch Outline

1. Automated Profiler
2. **Fast Optimizer**

Optimizer

O1: resources $\quad \text{Minimize} \sum_{d \in \mathcal{D}} (res_d + mem_d), \quad \textbf{s.t.} \qquad (1)$

C1: coverage $\quad \sum_{d \in p_\pi} b_{(d,s)} \geq 1 \quad \forall p \in \mathcal{P}, \forall s \in p_s$

C2: accuracy $\quad mem_{(d,s)} \geq s_{mem} \cdot b_{(d,s)} \quad \forall s \in \mathcal{S}, \forall d \in \mathcal{D}$

C3: capacity $\quad \sum_{s \in \mathcal{S}} b_{(d,s)} \cdot s_{rows} \leq d_{rows}, \quad \text{and}$

$$mem_d = \sum_{s \in \mathcal{S}} mem_{(d,s)} \leq d_{mem} \quad \forall d \in \mathcal{D}$$

C4: profiles $\quad \forall d \in \mathcal{D}:$

performance

$$time_d = d_{time}(res_d, \mathcal{P}_d, \{(mem_{(d,s)}, b_{(d,s)}) | s \in \mathcal{S}\})$$

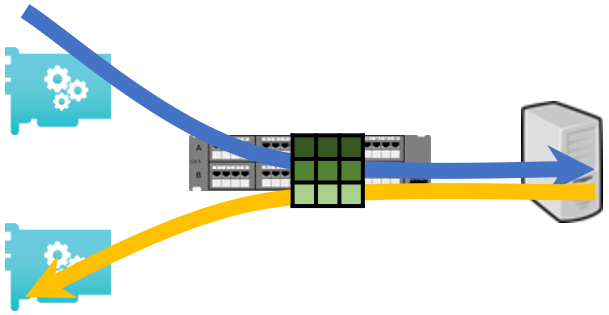
C5: traffic $\quad time_d \leq \frac{1}{d_{\text{traffic}}} \quad \forall d \in \mathcal{D}, \quad \text{where}$

$$d_{\text{traffic}} = \sum_{p \in \mathcal{P}_d} p_t, \quad \mathcal{P}_d = \{p | d \in p_\pi, p \in \mathcal{P}\}$$

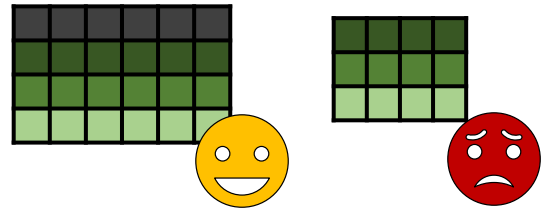
Placement and resource allocation
→ Mixed-Integer Program

Minimizes total resource usage

coverage



accuracy



capacity

H hashes
M memory

Fit?

performance
(from Profiler)

⇒ 10 Mpps

Challenge: Scalability & Dynamics

- Large and complex (non-convex) NP-Hard optimization problem.

  ⇒ Take hours to solve for a typical network with 1000s nodes.

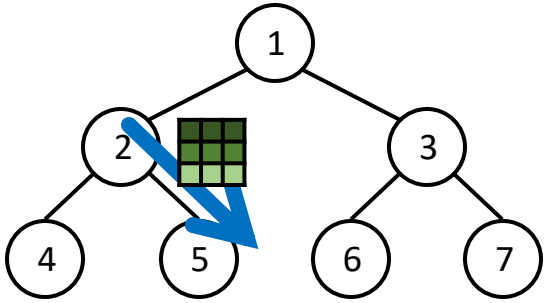
Hierarchically cluster devices to partition optimization into independent sub-problems

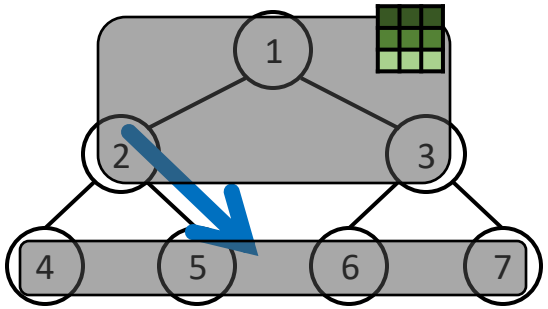
- Cluster devices
- Optimizer assigns sketches to clusters
- Optimizer (in-parallel) places sketches to devices inside the cluster

Clustering strategy matters

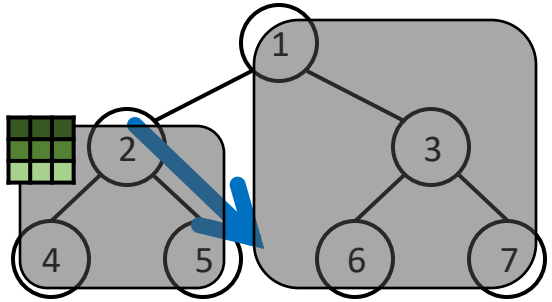
- Independent sub-problems can't share information

  ⇒ Infeasibility or sub-optimality





Strategy 1



Strategy 2

Keep devices that see the same traffic together

Evaluation: Profiler

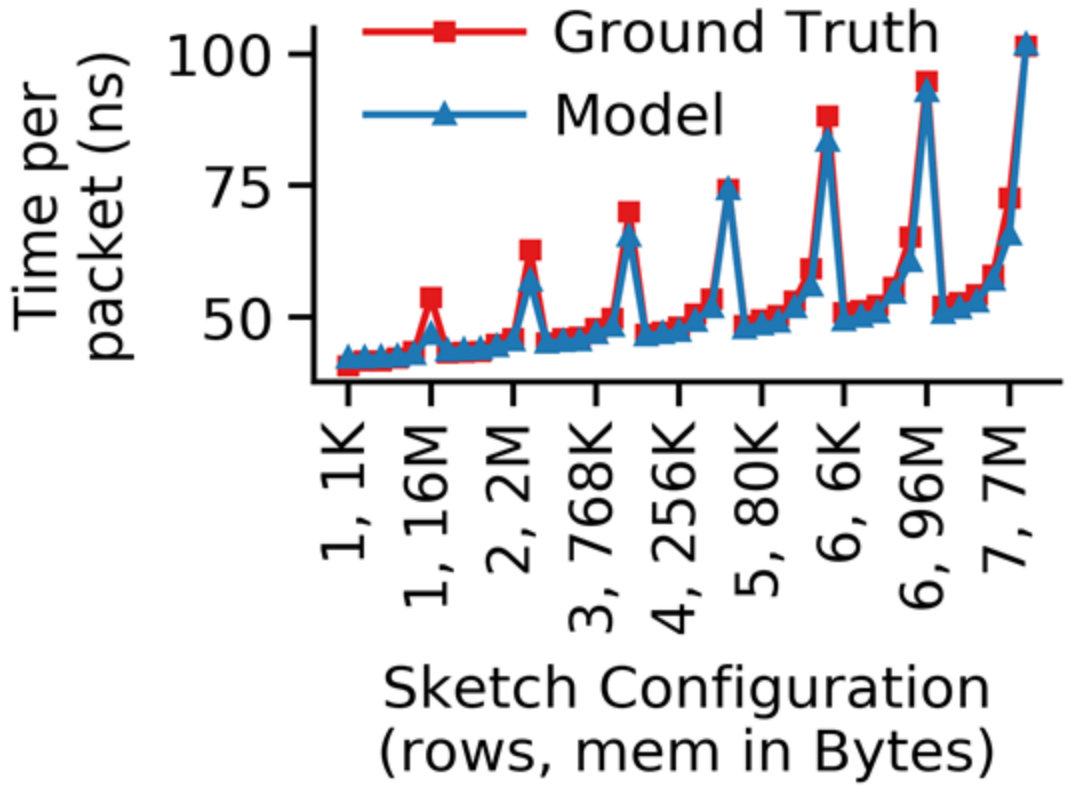
Devices:

1. Programmable switch (Barefoot Tofino)
2. FPGA Smart-NIC (Xilinx AU280)
3. SoC Smart-NIC (Netronome Agilio)
4. Software switch (OVS x86-based)

Sketches:

1. Count-Min Sketch
2. Count Sketch
3. UnivMon [SIGCOMM 16]

Within **~5% relative error**



Evaluation: Optimizer

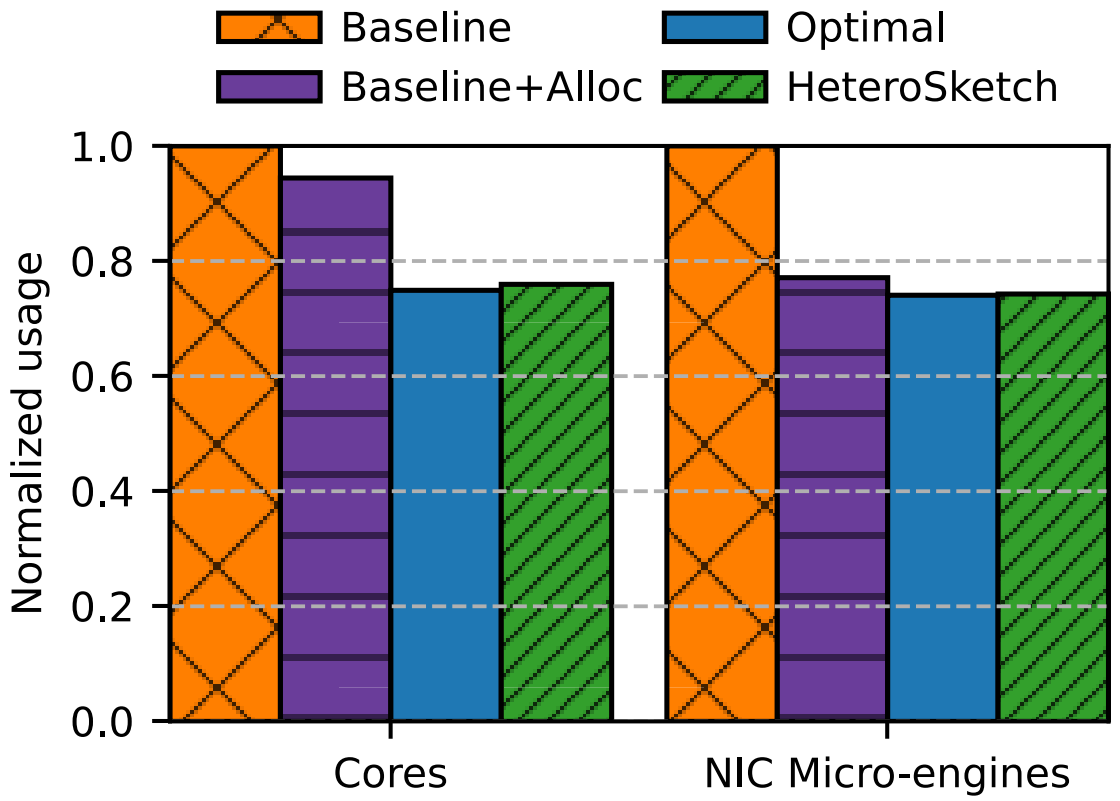
Metrics:

1. **Resource overhead**
2. Time to compute placement & resource allocation

Baselines:

1. Baseline: Memory only from UnivMon [SIGCOMM 16]
2. Baseline+Alloc: profile aware resource allocation post placement
3. Optimal: joint placement and allocation
4. … (others in paper)

Saves 20-30% resources. Close to optimal

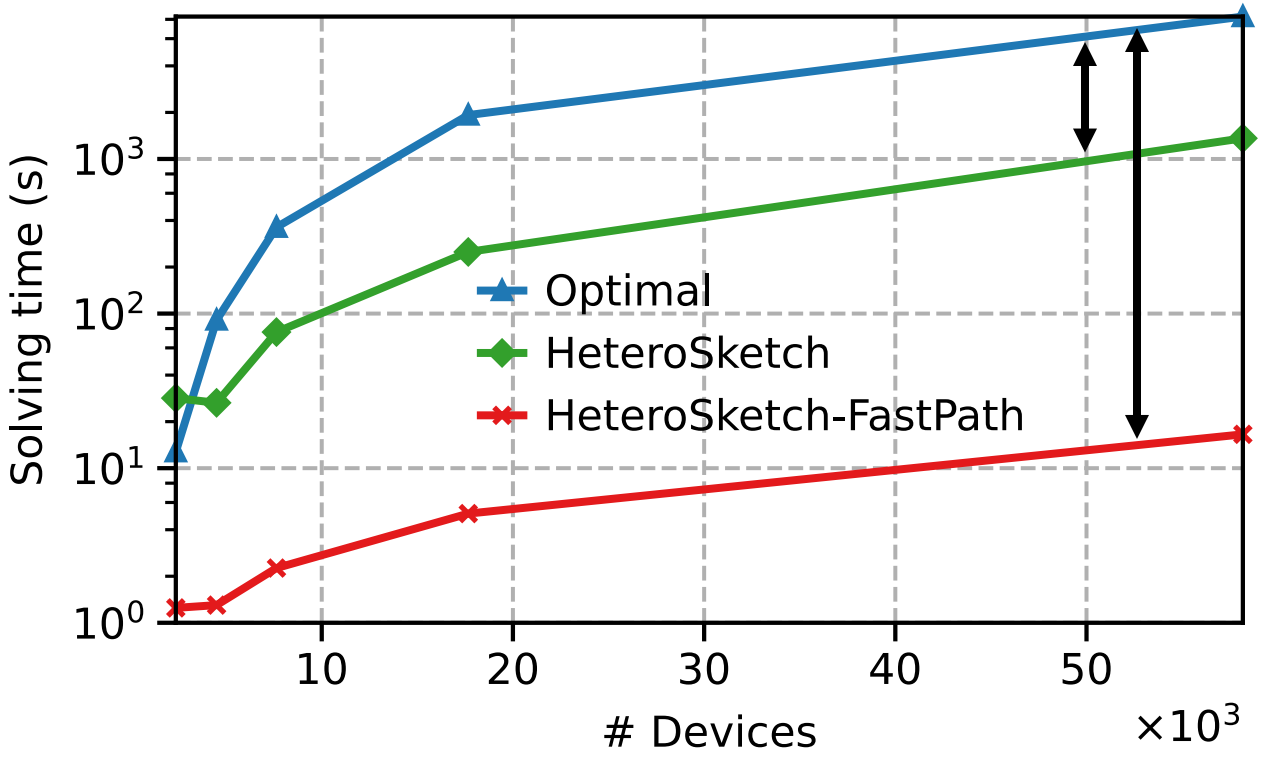


Evaluation: Optimizer scalability

Metrics:

1. Resource overhead
2. **Time to compute placement & resource allocation**

10x to 1000x quicker responses.
Scales to > 40k devices



* HeteroSketch+FastPath – additional optimization above clustering heuristic. See paper for details.

Summary & Future work

Existing solutions overlook heterogeneity and dynamics

HeteroSketch: Manage heterogeneity & dynamics
Automated Profiler, Fast Optimizer

✓ Reduce resource overhead by 20-30%.
✓ Prompt Responses. Scale to 40,000 devices.

→ Extended for general packet processing programs?

→ Clustering for other network optimization?
(NCFlow [NSDI21])

anupa@cmu.edu
Anup Agarwal

Zaoxing (Alan) Liu

Srinivasan Seshan

Thanks for your time! – Backup Slides

[NSDI22] HeteroSketch: Coordinating Network-wide Monitoring in Heterogeneous and Dynamic Networks

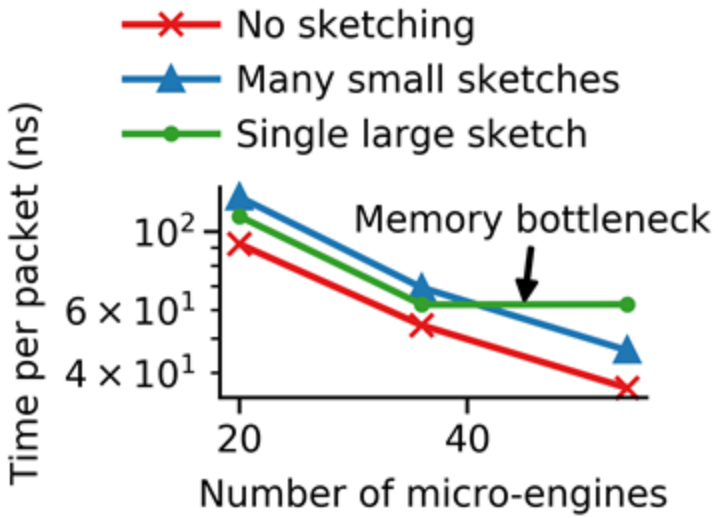
Anup Agarwal

Zaoxing (Alan) Liu

Srinivasan Seshan

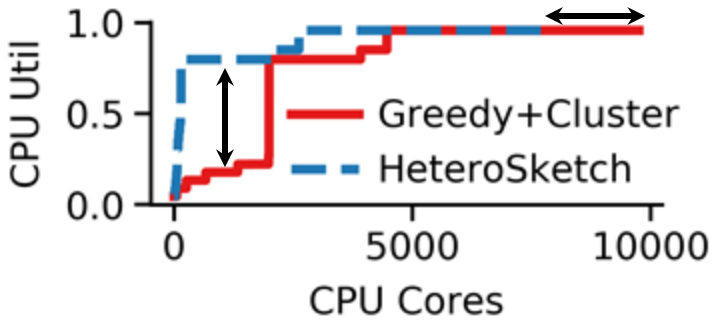
How? ⇒ Key sources of benefits

- Sketch-device affinity
- Bottleneck Awareness



- Able to trade-off resources
- Resource usage aware packing

Allocates 20% fewer cores, but each core > 80% utilized.



v/s OmniMon (SIGCOMM 20)

- General way to handle device heterogeneity. OmniMon mostly leverages location in topology and considers memory capacity. How would OmniMon differentiate between an ASIC vs FPGA SmartNIC?
- Memory / storage overheads scale with number of flows/packets.
- We consider dynamics w.r.t. changing requirements, traffic matrices, resources, topology.
- Sketch based v/s per-flow counters.
- Handling multiple flow definitions? 5-tuple, src/dest.

Evaluation: Dynamics (Fast Path)