Direct Telemetry Access

Jonatan Langlet
Queen Mary University of London

Ran Ben Basat
University College London

Sivaramakrishnan Ramanathan
University of Southern California

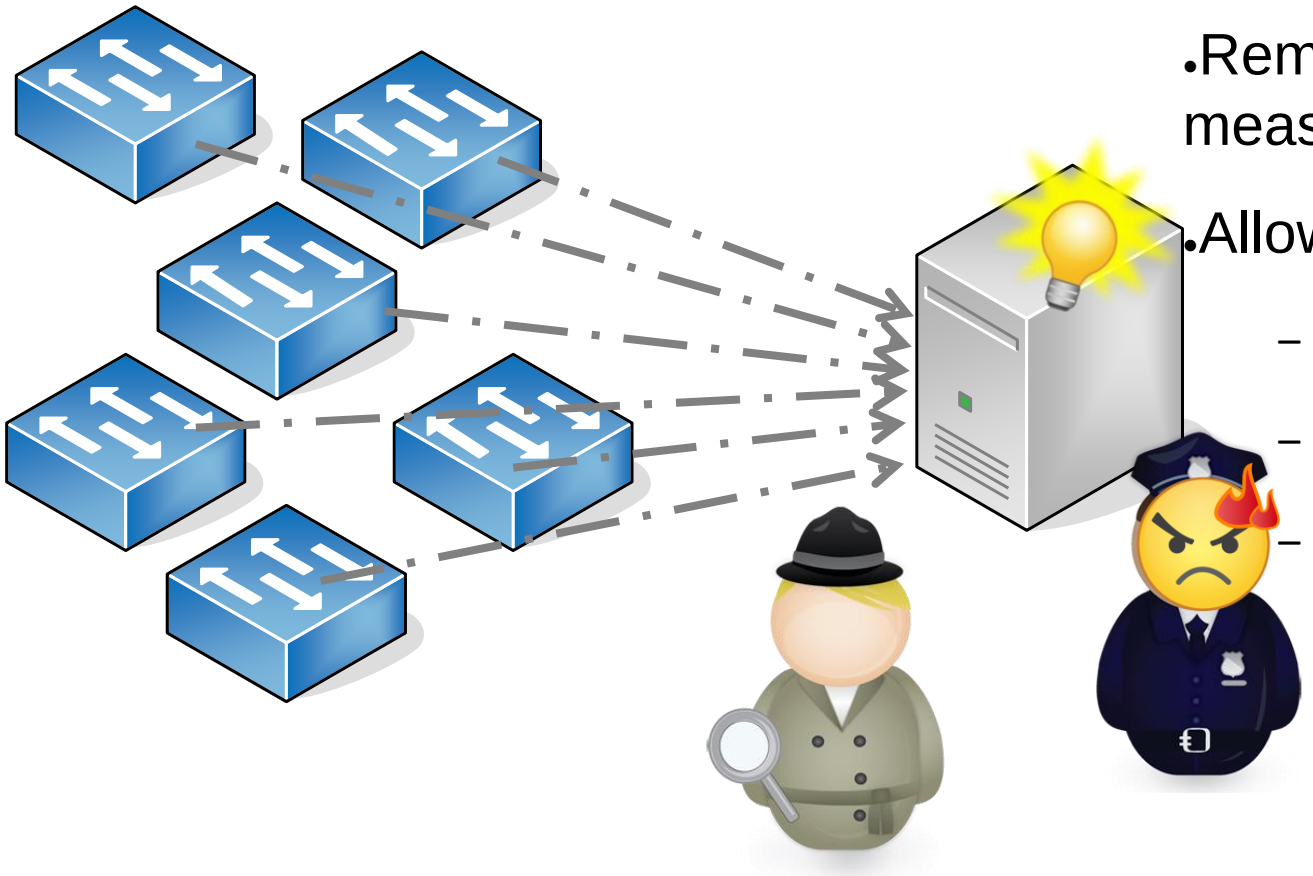
Gabriele Oliaro
Harvard University

Michael Mitzenmacher
Harvard University

Minlan Yu
Harvard University

Gianni Antichi
Queen Mary University of London

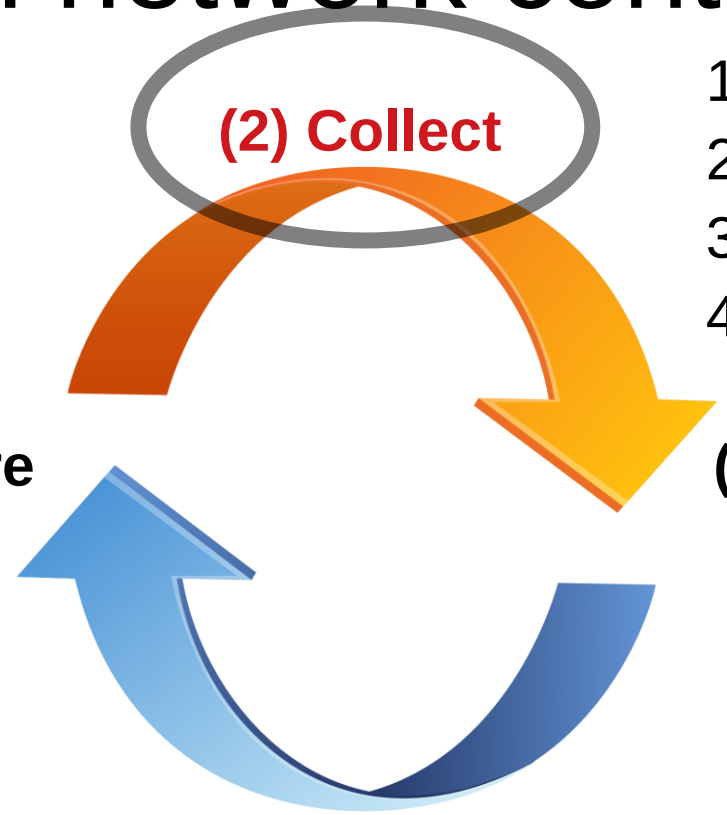
What is network telemetry?



- Remotely reading network measurements
- Allows deep network insight
  - Troubleshooting
  - SLA compliance
  - Network reactivity

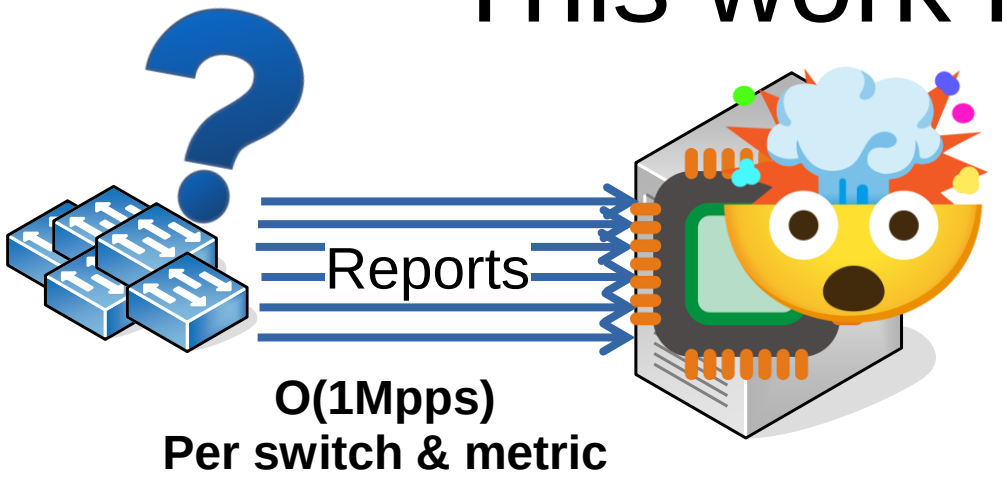
Modern network control loop

- Flow paths
- Queue depths
- Latency spikes
- Packet losses

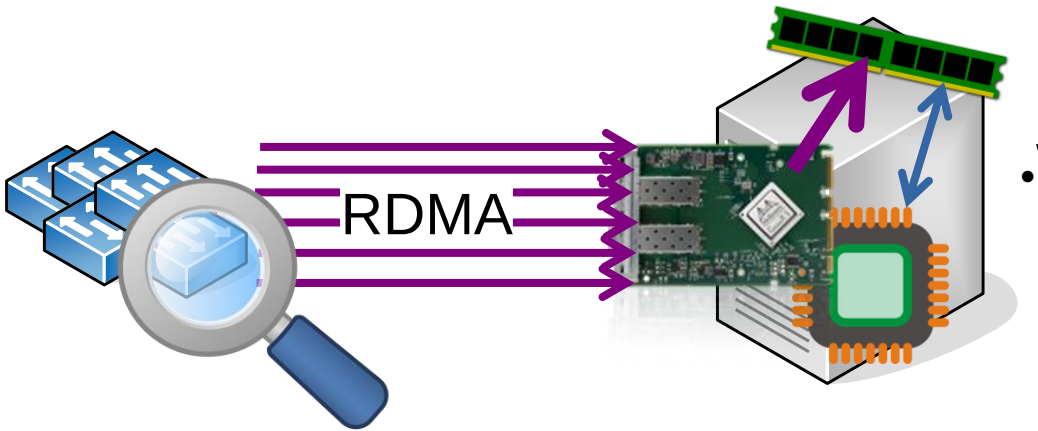


1)Continuously measure
2)Collecting for analysis
3)Adapting the network
4)Push changes

This work in one slide

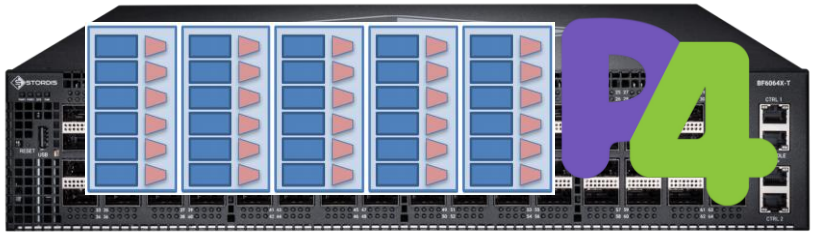


- Telemetry generates **tons** of data!
  - Overloads collector CPU
  - Reduced insight



- We redesign telemetry collection
  - Bypassing the CPU
  - Increased insight

Why is this now a problem?



- Programmable switches enable incredible traffic insight
  - Custom per-packet logic
- **Tons** of monitoring papers!

In-band Network Telemetry via Programmable Dataplanes

Changhoon Kim*, Anirudh Sivaraman**, Naga Katta***, Antonin Bas*, Advait Dixit*, Lawrence J Wobker*
*Barefoot Networks, **Massachusetts Institute of Technology, ***University
{chang, antonin, adixit, ljw}@barefootnetworks.com, anirudh@csail.mit

Flow Event Telemetry on Programmable Data Plane

Yu Zhou*†, Chen Sun*, Hongqiang Harry Liu*, Rui Miao*, Shi Bai*, Bo Li*, Zhilong Zheng*†, Lingjun Zhu*, Zhen Shen*, Yongqing Xi*, Pengcheng Zhang*, Dennis Cai*, Ming Zhang*, Mingwei Xu†
*Alibaba Group †INSC and BNRist, Tsinghua University

PINT: **Probabilistic In-band Network Telemetry**

Yuliang Li
Harvard University
yuliangli@g.harvard.edu

Michael Mitzenmacher
Harvard University
michaelm@eecs.harvard.edu

Sonata: Query-Driven Streaming Network Telemetry

Arpit Gupta
Princeton University

Rob Harrison
Princeton University

Marco Canini
KAUST

Nick Feamster
Princeton University

Jennif
Princet

Language-Directed Hardware Design for Network Performance Monitoring

PacketScope: Monitoring the Packet Lifecycle Inside a Switch

Ross Teixeira
Princeton University
rapt@cs.princeton.edu

Arpit Gupta
UC Santa Barbara
arpitgupta@cs.ucsb.edu

Rob Harrison
United States Military Academy

Vikram Nathan[1], Prateesh Goyal[1], umar Jeyakumar[3], Changhoon Kim[4]

One Sketch to Rule Them All: Rethinking Network Flow Monitoring with UnivMon

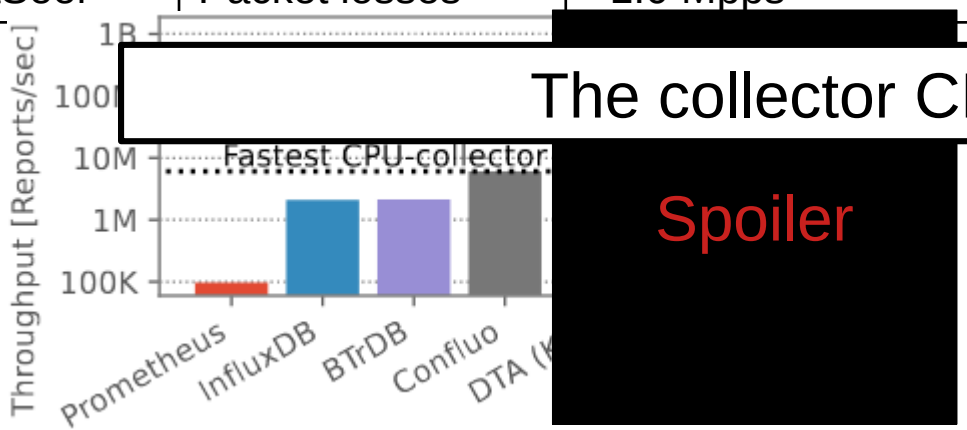
Zaoxing Liu†, Antonis Manousis*, Gregory Vorsanger†, Vyas Sekar*, Vladimir Braverman†
† Johns Hopkins University * Carnegie Mellon University

Telemetry is INTENSE

System	Metric	Generation Rate
INT 0.5%	Raw	~19 Mpps
Marple	TCP Out-of-Sequence	~6.7 Mpps
Marple	Flow counting	~4.3 Mpps
NetSeer	Packet losses	~1.0 Mpps

- Single monitoring query
  - O(1M) **per switch query**
- Collection performance
  - **Max** O(10M)

The collector CPU is bottlenecking!



Spoiler

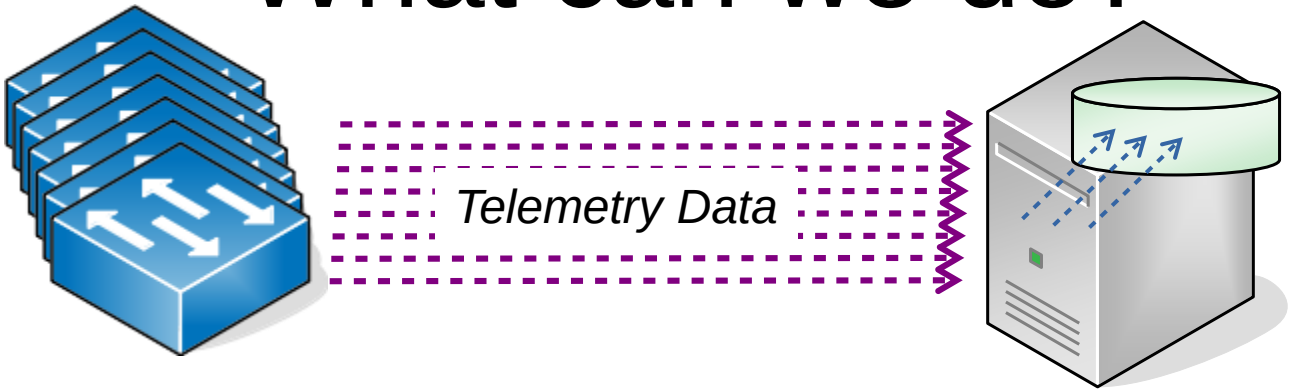
Fig: 16-core performance

Narayana, Srinivas, et al., "Language-Directed Hardware Design for Network Performance Monitoring", SIGCOMM'17

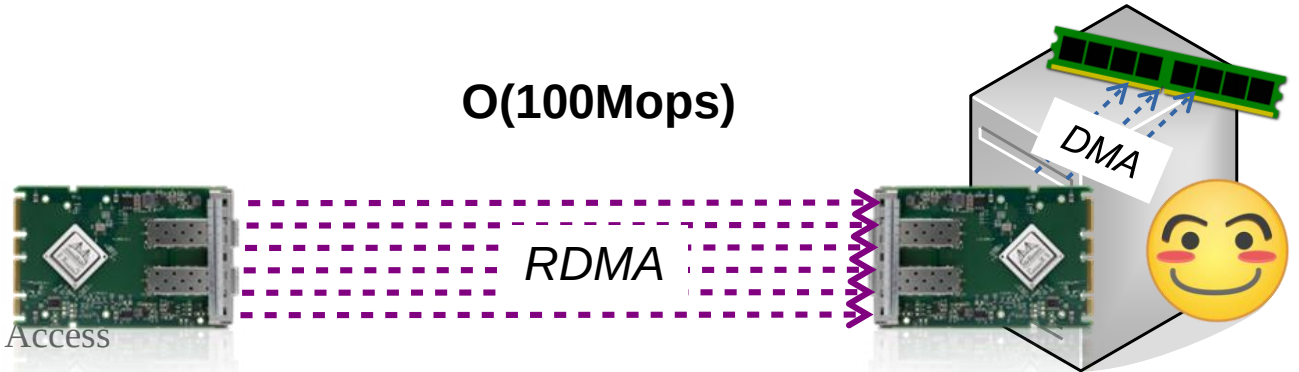
Zhou, Yu, et al., "Flow Event Telemetry on Programmable Data Plane", SIGCOMM'20

Khandelwal, Anurag, et al., "Confluo: Distributed Monitoring and Diagnosis Stack for High-speed Networks", NSDI'19

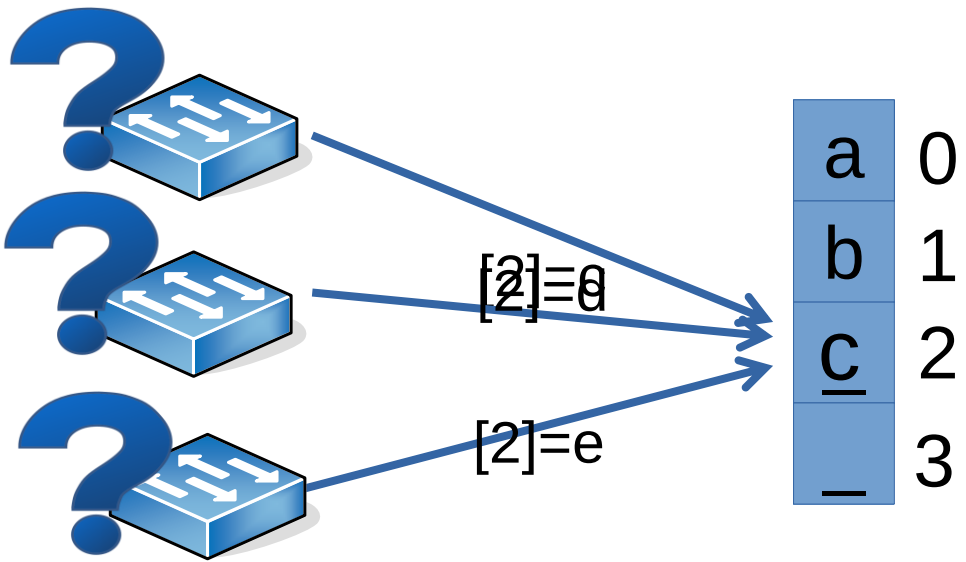
What can we do?



This is what **RDMA** is designed for!



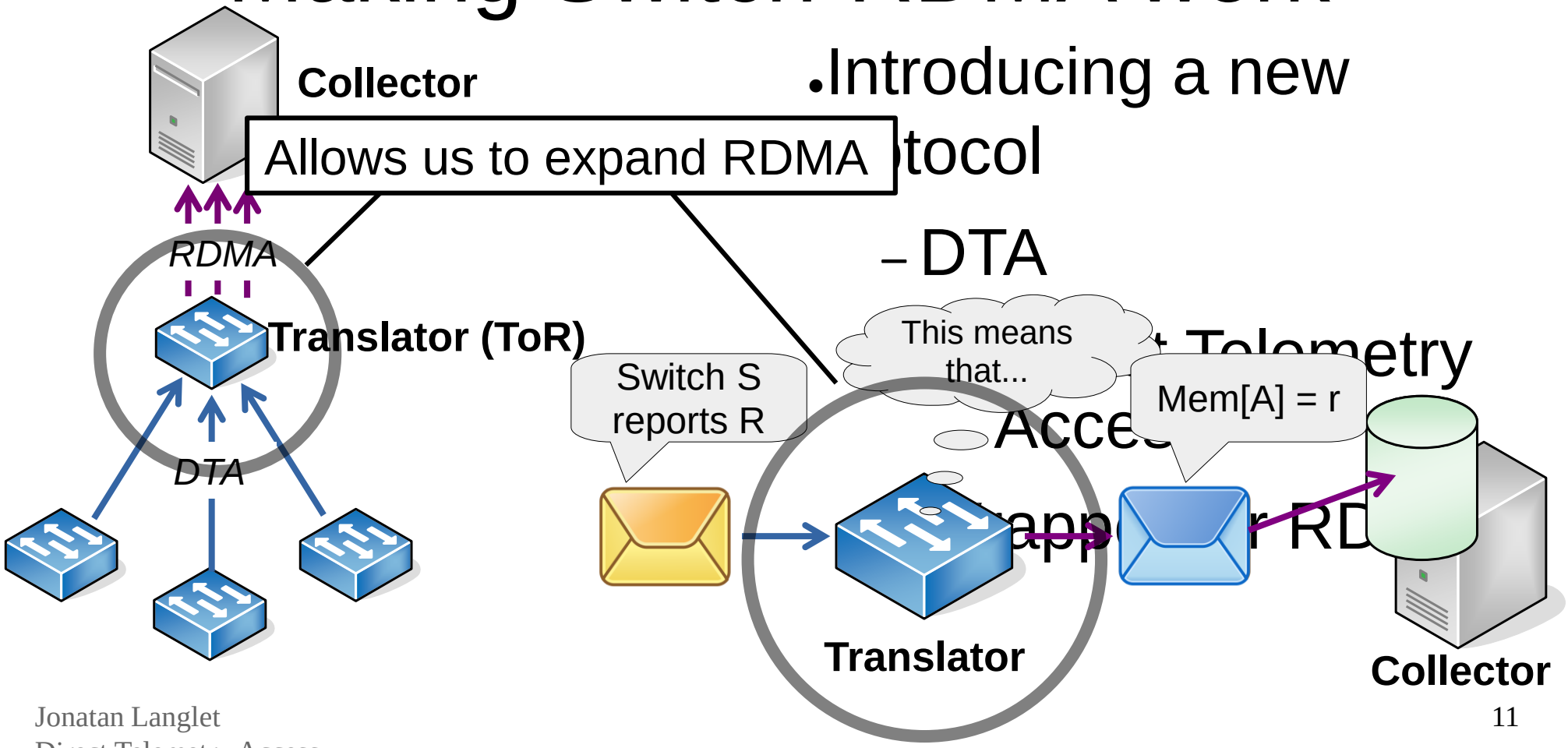
But RDMA is limited...



- Two high-speed operations
  - Read
  - Write
- High-speed RDMA is limited
  - Dumb memory operations
  - Explicit addresses

Making Switch-RDMA work

- Introducing a new
  tocol
  - DTA
  Telemetry
  Acce
  r RD



Fast and Generic

INT

HyperLogLog

Count Sketch

CM Sketch

Sonata (SIGCOMM'18)

Marple SIGCOMM'17)

AROMA (IFIP'20)

TurboFlow (EuroSys'18)

NetSeer (SIGCOMM'20)

PacketScope (SOSR'20)

PINT (SIGCOMM'20)

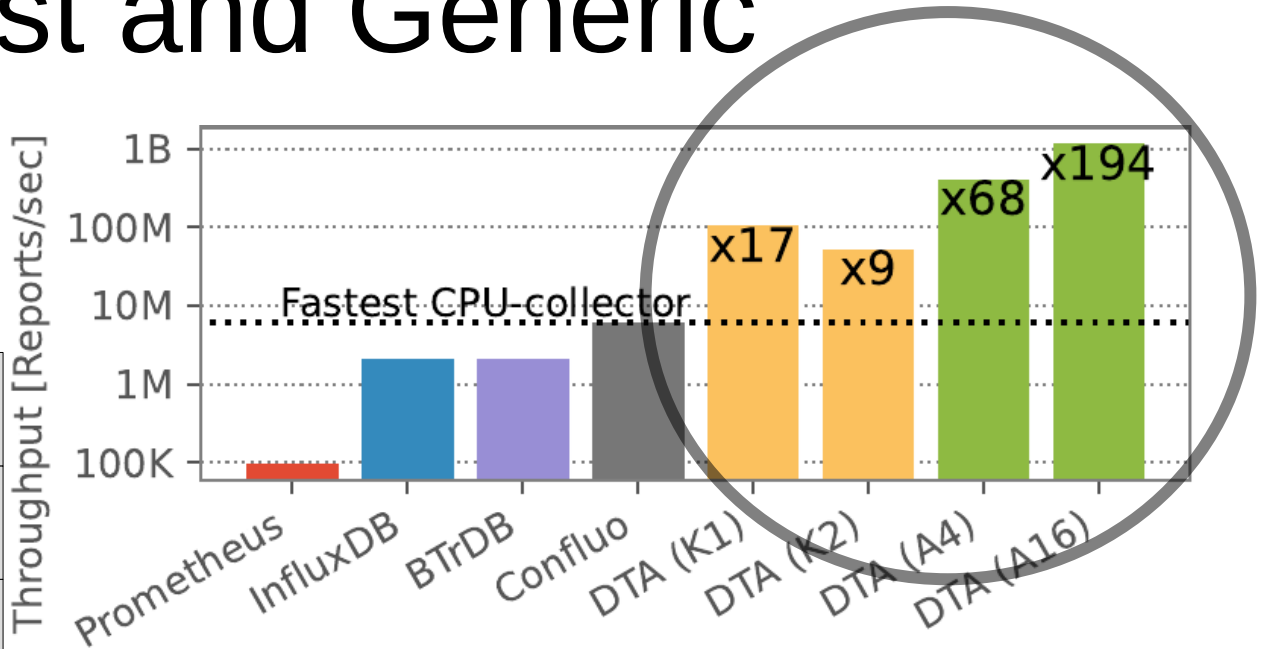
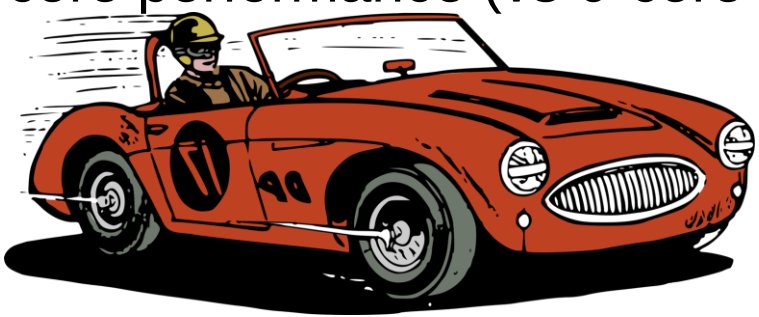
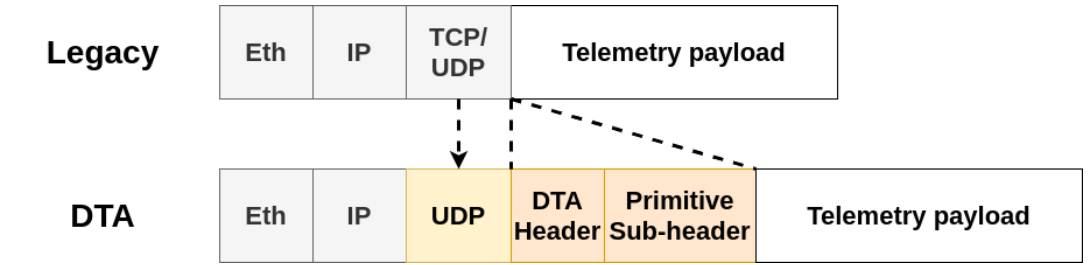


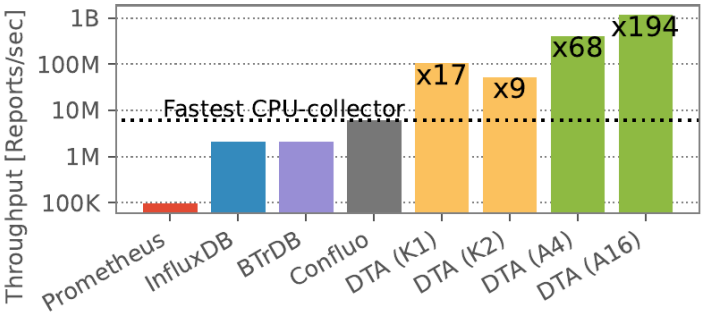
Fig: 16-core performance (vs 0-core DTA)

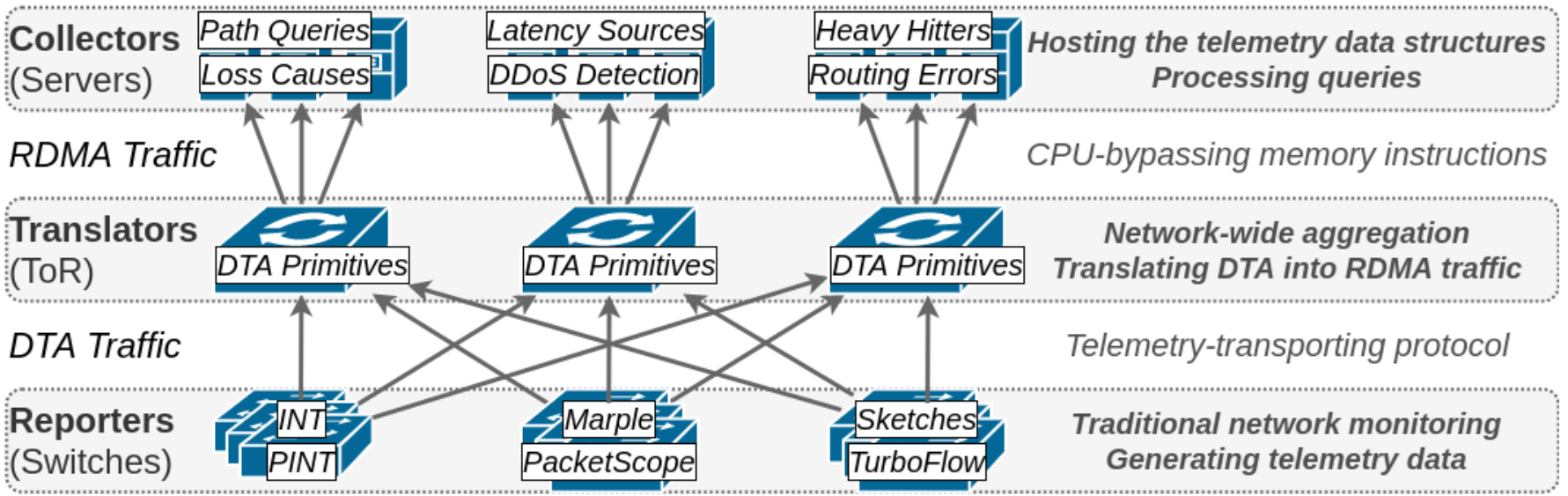


Backup slides

Overview


Legacy
Eth
IP
TCP/ UDP
Telemetry payload
DTA
Eth
IP
UDP
DTA Header
Primitive Sub-header
Telemetry payload


Throughput [Reports/sec]
1B
100M
10M
1M
100K
Fastest CPU-collector
x17
x9
x68
x194
Prometheus
InfluxDB
BTrDB
Confluo
DTA (K1)
DTA (K2)
DTA (A4)
DTA (A16)


Collectors
(Servers)
Path Queries
Loss Causes
Latency Sources
DDoS Detection
Heavy Hitters
Routing Errors
Hosting the telemetry data structures
Processing queries
RDMA Traffic
CPU-bypassing memory instructions
Translators
(ToR)
DTA Primitives
DTA Primitives
DTA Primitives
Network-wide aggregation
Translating DTA into RDMA traffic
DTA Traffic
Telemetry-transporting protocol
Reporters
(Switches)
INT
PINT
Marple
PacketScope
Sketches
TurboFlow
Traditional network monitoring
Generating telemetry data

Primitives

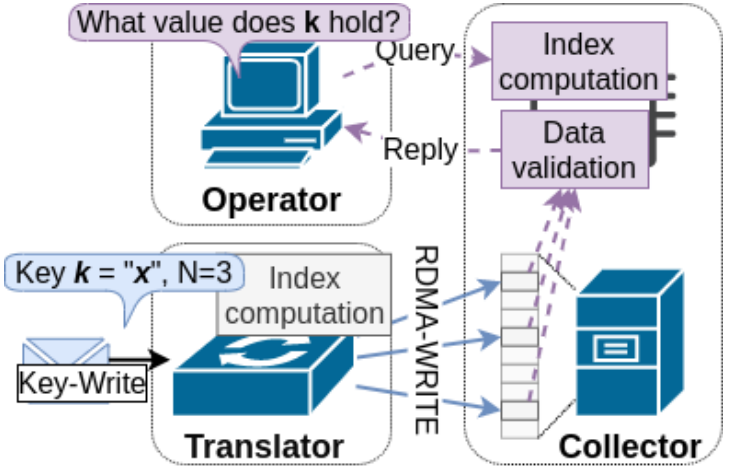


Fig: Key-Write

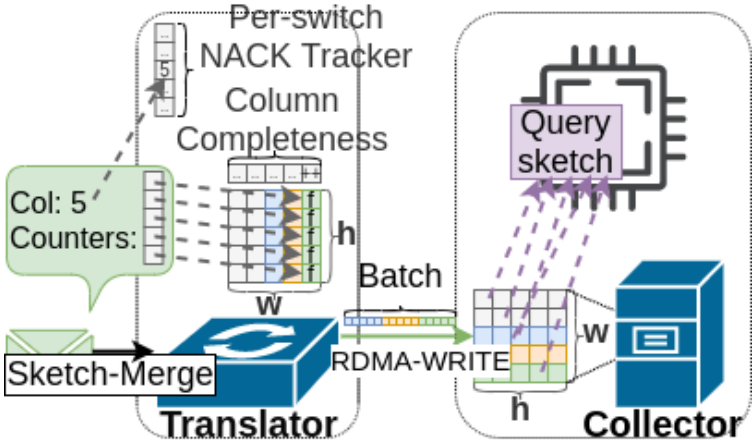


Fig: Sketch-Merge

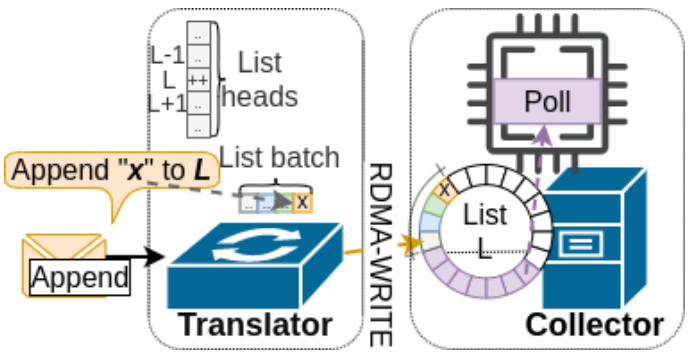


Fig: Append

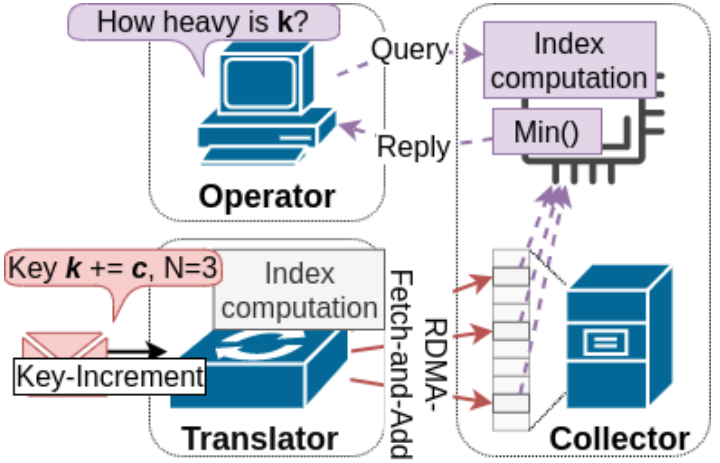
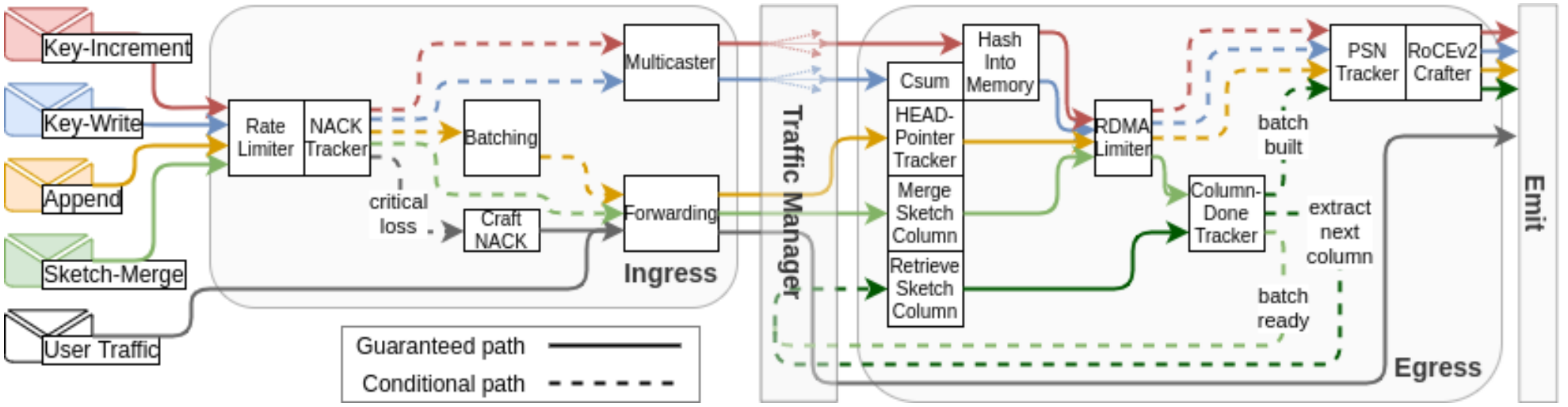


Fig: Key-Increment

ASIC Pipeline



Resource	Base footprint	Batching	Retransmission
SRAM	5.5%	+3.0%	+0.5%
Match Crossbar	5.9%	+9.0%	+0.2%
Table IDs	27.6%	+7.8%	+1.0%
Hash Dist Unit	13.9%	+20.8%	-
Ternary Bus	20.3%	+7.8%	+1.1%
Stateful ALU	10.4%	+31.3%	+2.1%

Table 3: Resource costs of the translator. Append batching creates batches of 16x4B data payloads, and retransmission supports tracking 65K reporter sequence numbers with 256 in-transit reports each.

Querying

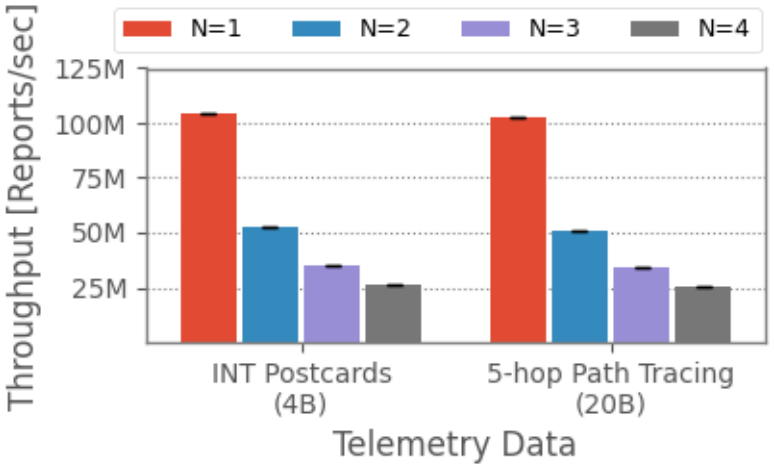
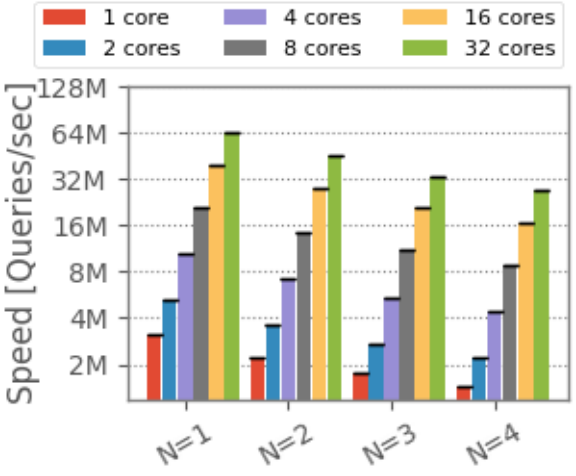


Fig: Key-Write speed



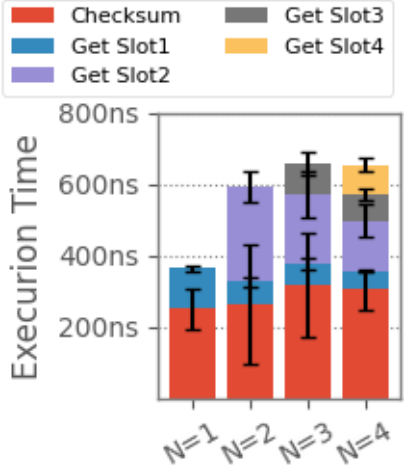


Fig: Key-Write breakdown

Fig: Key-Write Querying

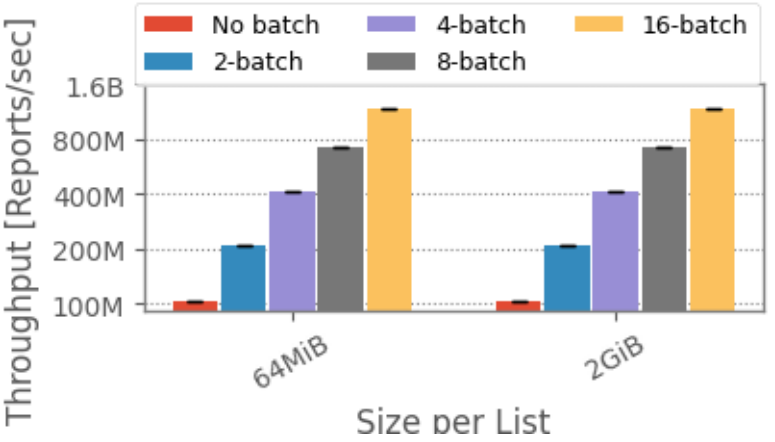


Fig: Append speed

Fig: Append querying

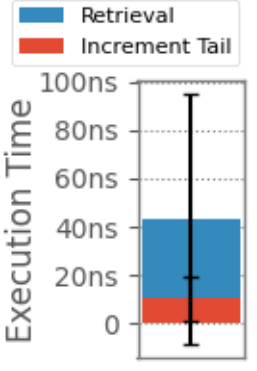
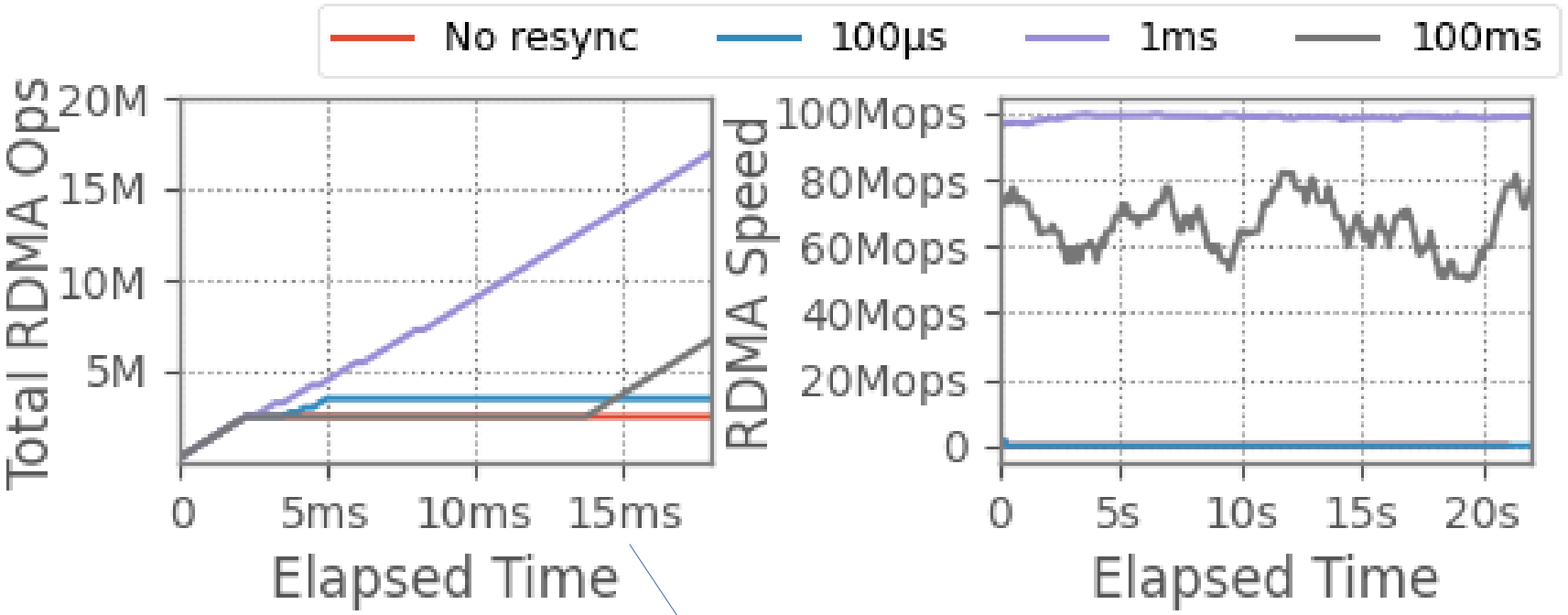


Fig: Append breakdown

Resync


No resync
100µs
1ms
100ms
Total RDMA Ops
20M
15M
10M
5M
0
5ms
10ms
15ms
Elapsed Time
RDMA Speed
100Mops
80Mops
60Mops
40Mops
20Mops
0
0
5s
10s
15s
20s
Elapsed Time

x10 :)