8F: Compact Data Structures for SDNs

Muthukrishnan (Rutgers) and Rexford (Princeton)

Collaborators

Vibhaalakshmi Sivaraman
Srinivas Narayana
Jennifer Rexford
Bala Krishamurthy

Ori Rottenstreich
Yaron Kanza
Morteza Monemizadeh

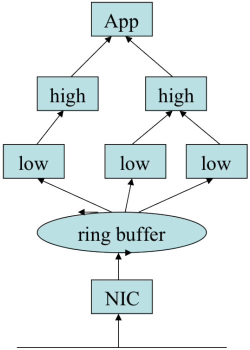
Brano Kveton

Backround: Streaming

- Say $a_1, \ldots, a_t$ arrive online, $a_i \in [1, U]$. Let $f_t(i)$ be the number of times i is seen.
- Compute frequency moments, $\sum_i (f_t(i))^j$ for $j = 0, 1, 2, \ldots, \infty$.
- Constraint: Use space $o(U, t)$, preferably $O(\log U)$.
- Following seminal work of Alon, Matias and Szegedy in 1996, lot of work in theory with different
  - models (inserts, deletes),
  - objects (graphs, matrices, geometric points, strings),
  - problems (clustering, matrix rank approximation, learning, signal processing).
- Compact data structures, CDS. Linear, composable, ...

Background: Motivation

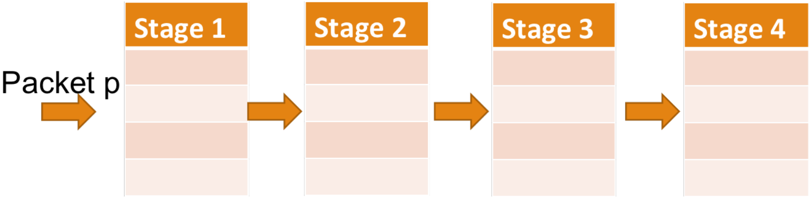
IP routers see packet (headers, contents) and need to analyze the traffic.



- We built two level arch: fast lightweight low level; high level expensive.
- Pure DSMS. SQL-like language.
- Parallelize by hashing on distinct groupbys, heartbeat mechanism, load shedding.
- http://www.corp.att.com/attlabs/docs/att_gigascope_factsheet_071405.pdf
- Linearity: $\mathrm{CDS}_{F+G} = \mathrm{CDS}_F + \mathrm{CDS}_G$.

Background: Software Defined Networks

- Centralized controller C, with programmable routers:
  - Stateful memory, supporting basic arithmetic.
  - Pipelined operations over multiple stages
  - State carried in packets across stages



- State of SDNs: in various stages of disrupting IP backbones to data centers, academic research and budget of IP service providers.

Constraints (Features?)

- Deterministic, small time budget for packet processing at each stage, few ns.
- Limited number of accesses to stateful memory per stage.
    - One read, modify, write.
- Limited amount of memory per stage.
    - Shared between forwarding rules and state for monitoring and collecting statistics.
- Feed-forward processing to avoid stalling and reduced throughput.
    - Multiple packets might be simultaneously processed by the switch pipeline, it is desirable to process it exactly once in orer to maintain throughput since packets will be stalled in the pipeline otherwise.

Example: Heavy hitters

- Return heavy hitters, all “flows” i with $f_t(i) \geq \frac{\sum_i f_t(i)}{k}$.
- Space-saving algorithm [ICDT 05]:
  - Maintains $O(k)$ flows with associated frequency counters; on each new IP packet, potentially find the minimum frequency and replace it.
- HashPipe [SOSR17]:
  - In each stage of counters, sample one location as surrogate for the min.
  - From stage to stage, carry the min from prior stage in the packet.
  - P4 implementation.
- Experts will observe that Count-Min Sketch works. Count-min is now in P4.

Open Problems

- Extreme Streaming.
  - Estimate the median in one pass with 1 unit of memory.
  - Imagine streaming over CDSs or sketches.
- Path Problems.
  - We can count the number of packets that went through routers i and j. Not doable in std distributed streaming.
  - What is the power of distributed streaming when items can carry $O(1)$ memory?
  - We can estimate the traffic matrix in IP N/Ws. [KKM17]
- Multidimensional CDSs.
  - With d dimensions, space/time becomes exponential in d.
  - Use a graphical model on IP flow dimensions, use count-min sketch on marginals. [KM+ ECML16]
  - Can we estimate graphical models via (extreme) streaming?

Far Open

- Multiple, continuous queries over multiple routers.
- Optimize per flow space, per packet time, per router communication.
- Novelty: Optimize EXECUTE packets and paths.
- AI on IP networks.: Why did this flow have high latency?