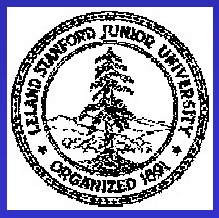
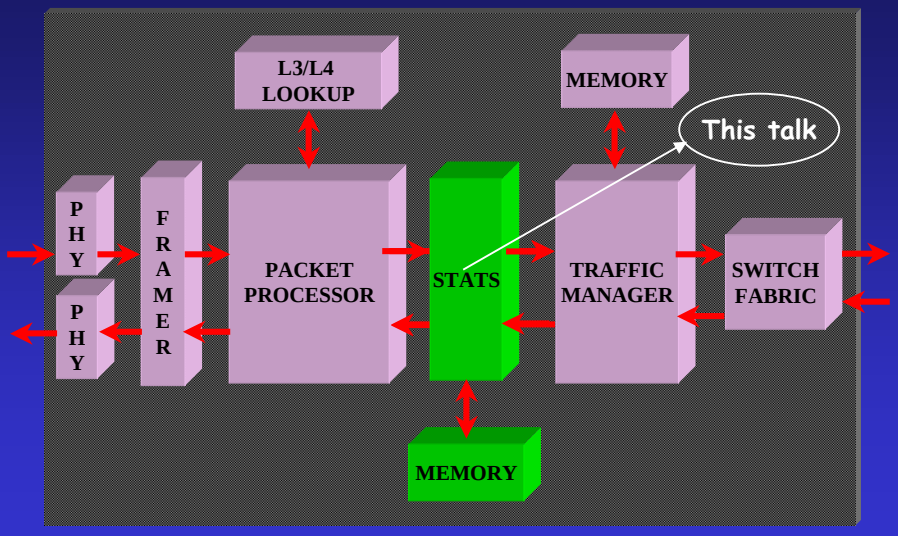
Analysis of a Statistics Counter Architecture



Devavrat Shah, Sundar Iyer,
Balaji Prabhakar & Nick McKeown
(devavrat, sundaes, balaji, nickm)@stanford.edu
Departments of Electrical Engineering &
Computer Science, Stanford University

Motivation:
Typical Line Card Architecture



What is a Statistics Counter?

- When a packet arrives, it is first classified to determine the type of the packet.
- Depending on the type(s), one or more counters corresponding to the criteria are updated/incremented

Examples of Statistics Counters

- Packet switches maintain counters for
  - Route prefixes, ACLs, TCP connections
  - SNMP MIBs
- Counters are required for
  - Traffic Engineering
    - Policing & Shaping
  - Intrusion detection
  - Performance Monitoring (RMON)
  - Network Tracing

Counter Requirements: A standard IP Router

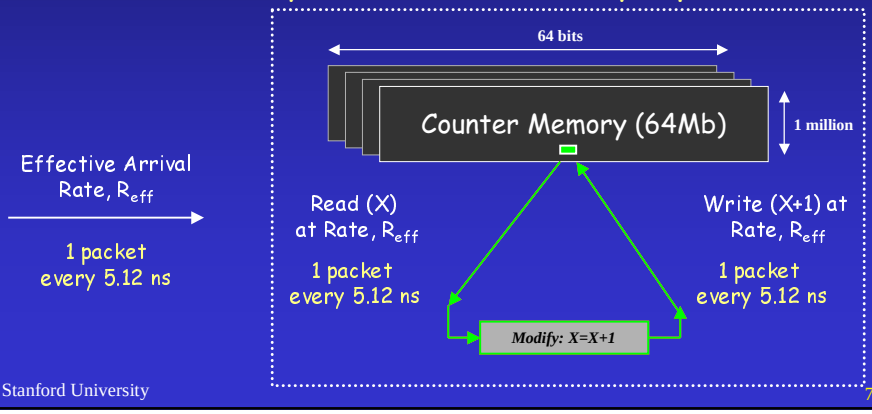
- Number of Counters
  - ~~1 Million~~ : per prefix counters
  - ~~128K~~ : policing counters
- Size of Counters
  - 32-64 bit counters

Motivation: How not to compromise

Determine and analyze techniques for building very high speed (>100Gb/s) statistics counters and support an extremely large number of counters.

Why is Implementing this a Hard Problem?

OC192c = 10Gb/s; Counters/pkt (C) = 10; R_{eff} = 100Gb/s;
Total Counters (N) = 1 million; Counter Size (M) = 64 bits;
Counter Memory = N*M = 64Mb; 64 byte packets



Why is Existing Memory Technology not Ideal?

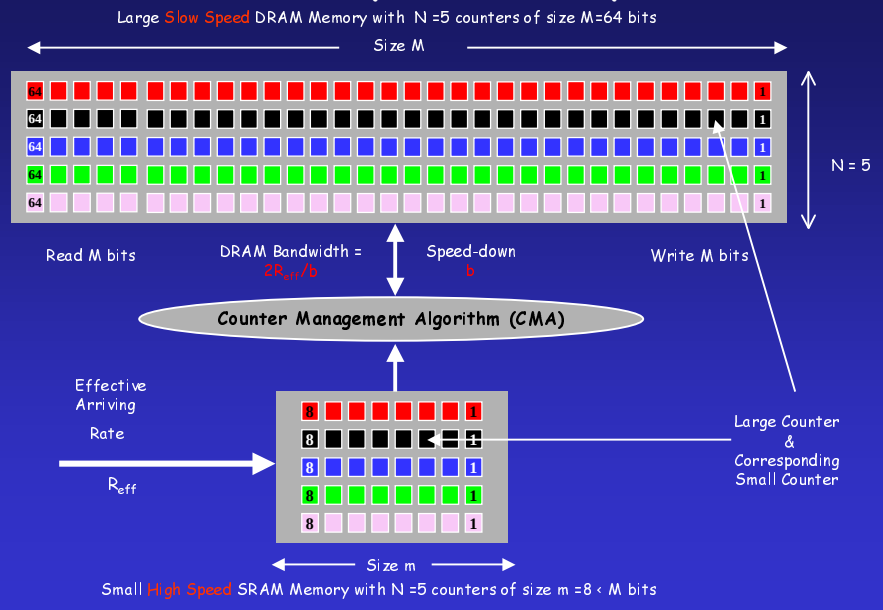
Use SRAM?

+ fast enough random access time, but
- too expensive, and
- too low density to store 64Mb of data.

Use DRAM?

+ high density means we can store data, but
- too slow (typically 50ns random access time)
- Read-modify-write penalty

Memory Hierarchy

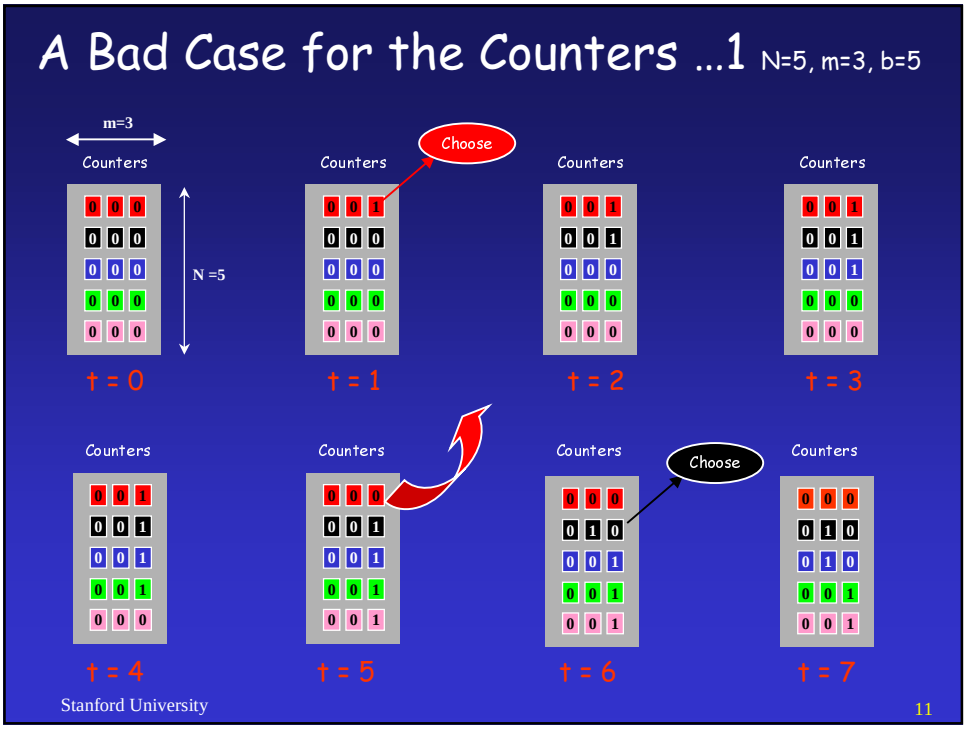


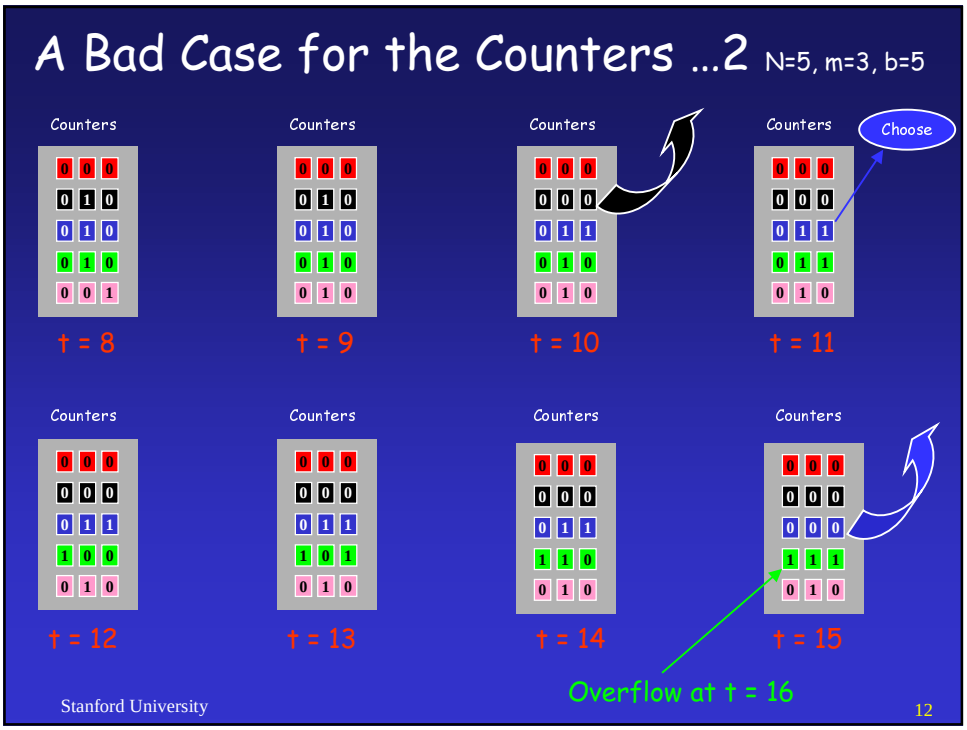
Questions

How large does the SRAM need to be:

- To deterministically guarantee that none of the counters in the SRAM overflow, irrespective of the arriving traffic pattern.

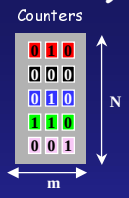
What Counter Management Algorithm (CMA) should we use?


A Bad Case for the Counters ...1 N=5, m=3, b=5
m=3
N =5
Counters
Choose
t = 0
t = 1
t = 2
t = 3
t = 4
t = 5
t = 6
t = 7
Stanford University
11


A Bad Case for the Counters ...2 N=5, m=3, b=5
Counters
Choose
t = 8
t = 9
t = 10
t = 11
t = 12
t = 13
t = 14
t = 15
Overflow at t = 16
Stanford University
12

Largest Counter First (LCF-CMA)

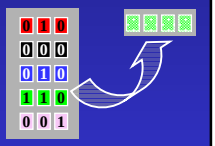
1. In SRAM: N Counters of size m



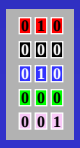
2. Compute: Find out the counter with the largest count every b timeslots.

green!

3. Update: Schedule update to counter in DRAM



4. Clear: Set value of the counter in SRAM to zero



Optimality of LCF-CMA

Theorem:

LCF-CMA, is optimal in the sense that it minimizes the size of the counter maintained in SRAM

Minimum Size of the SRAM Counter

Theorem: (speed-down factor b>1)

LCF-CMA, minimizes the size of counter (in bits) in the SRAM to:

$$f(b, N) = \log_2 \left(\frac{\ln bN}{\ln [b/(b-1)]} \right)$$

$$= \Theta (\log \log N)$$

Implementation Numbers

Example:

• OC192 Line Card	:R = 10Gb/sec.
• No. of counters per packet	:C = 10
• R_{eff}= R*C	:100Gb/s
• Cell Size	:64bytes
• T_{eff} = 5.12/2	:2.56ns
• DRAM Random Access Time	:51.2ns
• Speed-down Factor, b >= T/T_{eff}	:20
• Total Number of Counters	:1000, 1 million
• Size of Counters	:64 bits, 8 bits

Implementation Examples

OC192 Line Card

Values	f(b,N)	Brute Force	LCF-CMA
N=1000 M=8	8	SRAM=8Kb DRAM=0	SRAM=8Kb DRAM=8Kb
N=1000 M=64	8	SRAM=64Kb DRAM=0	SRAM=8Kb DRAM=64Kb
N=1000000 M=64	9	SRAM=64Mb DRAM=0	SRAM=9Mb DRAM=64Mb

Conclusion

- The SRAM size required by LCF-CMA is a very slow growing function of N
- There is a minimal tradeoff in SRAM memory size
- The LCF-CMA technique allows a designer to arbitrarily scale the speed of a statistics counter architecture using existing memory technology