HASH FILTER: MAKING BLOOM FILTER EVEN MORE COMPACT AND DELETABLE
Bin Fan, David G. Andersen (Carnegie Mellon University), Michael Kaminsky (Intel Labs Pittsburgh)

BLOOM FILTER: APPROXIMATE SET-MEMBERSHIP TEST

- Answers questions like “is foo present in this set?”
- Returns “no” with 100% certainty, “yes” with a false positive rate e (tunable parameter)

![Bloom Filter Diagram]

- 1.44 \log_2 \left( \frac{1}{e} \right) \text{ bits/key, 44% more than optimal}
- False positive rate vs. bits/key
- Hash filter is more space efficient when f.p.r. < 2%
- Can not delete or spend 4x space for a counting Bloom Filter.

HASH FILTER IN A NUTSHELL

- Essentially a hash table filled with signatures!

- An optimal hash filter to represent set \{A,..,H\}, with f.p.r. e

- A signature of \((\log_2(1/e)+3) \text{ bits}

- Cuckoo hash table

- Close to 100% full (e.g. 93%), but much cheaper to construct/update

- Delete supported!

MEM ACCESSES VS. BITS/KEY

- When f.p.r. < 1%, hash filter wins in both metrics!

\[ k \times \log_2(1/e) \text{ vs. } k \times \log_2(1/e) \]

\[ \text{Bits per entry} \]

\[ \text{False positive probability} \]

\[ \text{Crossover: 7.9 bits (2% false positive rate)} \]