Network Applications of Bloom Filters: A Survey
By Andrei Broder and Michael Mitzenmacher

Objective

- “Simple space-efficient randomized data structure for representing a set in order to support membership queries”
  - False positives
  - But “the probability of an error [can be] made sufficiently low”
- Testing the membership is efficient
- Space saving
- Easily parallelizable
Network Applications of Bloom Filters: A Survey
Andrei Broder and Michael Mitzenmacher

Mechanics – Add

![Diagram showing the addition of objects to a Bloom filter.]

Mechanics – Check

![Diagram showing the check of objects and identifying a false positive.]

---

Pedro Brandão 2

2008/10/14 Reading Group - Pedro Brandão
### Parameters

<table>
<thead>
<tr>
<th></th>
<th>Decrease</th>
<th>Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num. of Hash functions $k$</td>
<td>• Less hash functions (computation)</td>
<td>• More hash functions (computation)</td>
</tr>
<tr>
<td></td>
<td>• Increase in “zeros” ratios $\Rightarrow$</td>
<td>• Higher change to find 0 bits for non members</td>
</tr>
<tr>
<td></td>
<td>higher false positive</td>
<td></td>
</tr>
<tr>
<td>Size of filter $m$</td>
<td>• Less memory needed</td>
<td>• More memory needed</td>
</tr>
<tr>
<td></td>
<td>• Decrease zero bit ratio hole $\Rightarrow$</td>
<td>• Increase zero &quot; ratio $\Rightarrow$ lower false positive</td>
</tr>
<tr>
<td></td>
<td>higher false positive</td>
<td></td>
</tr>
<tr>
<td>Num. of elements in set $n$</td>
<td>• Increase zero ratio $\Rightarrow$ lower false positive</td>
<td>• Decrease zero ratio $\Rightarrow$ higher false positive</td>
</tr>
</tbody>
</table>

### Theory I

- Probability of a bit being zero:
\[
p' = \left(1 - \frac{1}{m}\right)^{kn} \approx e^{-kn/m}.
\]

- Probability of false positive (approximation):
\[
f' = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k = (1 - p')^k
\]

- 2nd Approx. for false positive:
\[
f = \left(1 - e^{-kn/m}\right)^k = (1 - p)^k
\]
Theory II

- If the array is split in \( \frac{m}{k} \) for each hash function. A specific bit has zero with probability:

\[
\left(1 - \frac{k}{m}\right)^n \leq \left(1 - \frac{1}{m}\right)^{kn}
\]

\[
\approx e^{-kn/m}
\]

- But difference is negligible in practice

Minimizing I

\[
f = \exp(\ln(k) \ln(1 - e^{-kn/m}))
\]

\[
g = k \ln(1 - e^{-kn/m}).
\]

Minimize \( f \) is to minimize \( g \)

Find derivative to \( k \)

\[
\frac{\partial g}{\partial k} = \ln \left(1 - e^{-\frac{kn}{m}}\right) + \frac{kn}{m} \frac{e^{-\frac{kn}{m}}}{1 - e^{-\frac{kn}{m}}}
\]

For derivative = 0

\[
k = \ln 2 \cdot (m/n)
\]

Optimal \( k \) (using aprox of slide 6)

But this minimum does not depend on this aprox. and holds even without it
Minimizing II

- Thus for the optimal \( k \)
  \[ k = \ln 2 \cdot (m/n) \]

- The false positive rate is:
  \[ f = (1/2)^{k} \geq (1/2)^{m/n \ln 2} \approx (0.6185)^{m/n} \]

- The \( m \) for an \( \varepsilon \) false positive rate is:
  \[ m \geq n \log_2(1/\varepsilon) \] (see paper)

- Making \( f \leq \varepsilon \) and for the optimal \( k \) requires
  \[ m \geq n \log_2 e \cdot \log_2(1/\varepsilon). \]
  \( \approx 1.44 \)

Tricks

- Union \( \Rightarrow \) OR

- Intersection \( \Rightarrow \) AND

- Halving size
Variants

- Counter bloom filter
  - More than one bit per entry to enable increase and decrease of values
- Compressed bloom filters
  - Enable compression of transmitted filter by increasing $m$ (this allows to decrease $k$)
    - Compression is done using regular compression functions

Applications types

- Collaborating in overlay and peer-to-peer networks
- Resource routing
- Packet routing
- Measurement
Examples

- Dictionaries: Hyphenation, Dicts, bad passwds
- Databases: joins, estimate size semi-joins, differential files

Examples: Cache

- Proxy-cache
  - Exchange bloom filters between proxies "with" cache contents
  - False positive ➔ access to proxy without content
  - Use of counting bloom filters for cache changes
  - Compressed bloom filters enhance the transmission savings
Examples: P2P

- Moderated size nets
- Use bloom filters to know the location of objects (instead of the full id of the object)
  - False positive ➔ extra requests and need to have alternative location method
- Set reconciliation
  - False positives ➔ not all members of the (Sa-Sb) are sent
  - Could be used in large file distribution

Network Routing

- Finding resources (simple):
  - Bloom filter to denote where resources are found (using ORing to unify)
  - False positive ➔ extra path traversal and backtracking or alternative routing method needed
- Finding resources (P2P):
  - Bloom filters per edge per distance (on an edge, there exists a bloom filter per distance reachable through that edge)
  - False positive ➔ extra path traversal and alternative algorithm needed
Network Routing II

- Geographic routing:
  - Region is divided into areas with one responsible node
  - Each node has a bloom filter for reachabilities from itself and siblings
  - False positive $\Rightarrow$ extra path traversal and alternative...

Packet Routing

- Detecting loops (unicast, multicast)
  - Use a bloom filter in the packet to mark nodes traversed
  - False positive $\Rightarrow$ packet discarded?

- Queue management:
  - Flow behaviour detection
  - Use of counting bloom filter that increase/decrease based on the current queue for the flow
  - False positive $\Rightarrow$ well behaved flows punish
    - $\Rightarrow$ change hash functions periodically
Packet Routing II

- Multicast:
  - Detect itf where to send packet
  - Use bloom filter to associate addresses to itfs where the packet should be sent
  - False positive ➔ eventual loop

Measurement I

- Detect heavy flows:
  - Counting bloom filters that count bytes per flow
  - Heavy flows have value > threshold
  - False positive ➔ light flow marked
  - Conservative update

\[ y_i - 4 \]

\[ 1 2 0 3 5 9 0 \]

\[ 3 8 5 7 6 1 0 3 5 9 0 \]
Measurement II

- IP traceback:
  - Register a packets path (routers it has passed by)
  - Each router has bloom filter of packets received
  - False positive \(\rightarrow\) branches in paths

Thank you

Content is based on:
Network Applications of Bloom Filters: A Survey
By Andrei Broder and Michael Mitzenmacher