

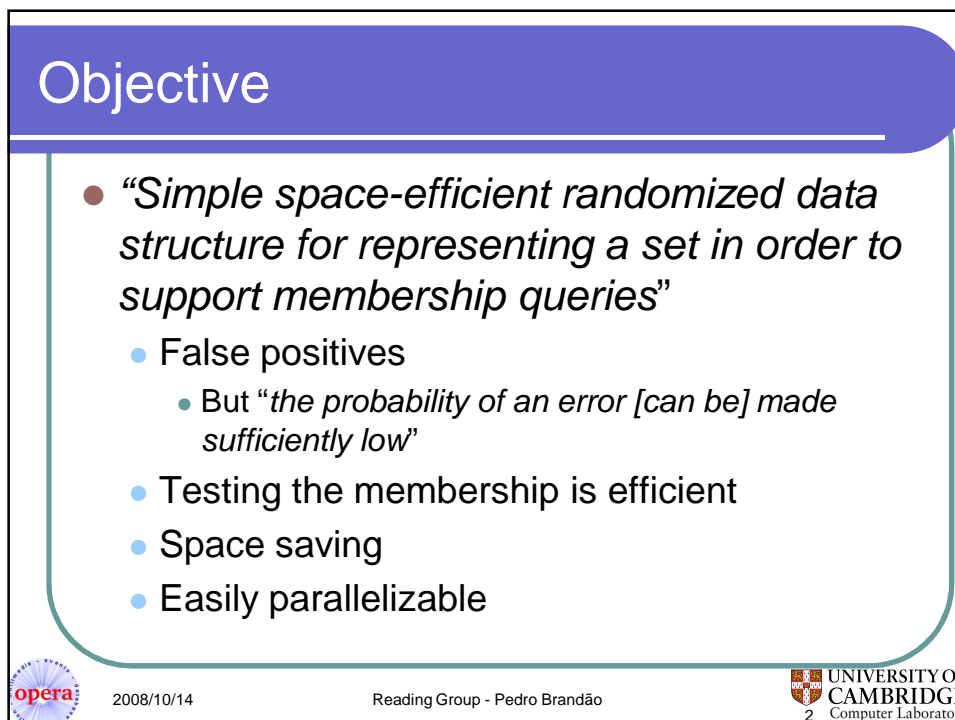


Network Applications of
Bloom Filters: A Survey
By Andrei Broder and Michael Mitzenmacher

Opera Reading Group
Pedro Brandão




UNIVERSITY OF
CAMBRIDGE
Computer Laboratory




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



2008/10/14

Reading Group - Pedro Brandão



UNIVERSITY OF
CAMBRIDGE
2 Computer Laboratory

Mechanics – Add

k	2
n	6
m	16
<i>ratio</i>	2.667
<i>Opt k</i>	1.848
<i>error opt</i>	0.289
<i>error</i>	0.291

2008/10/14
Reading Group - Pedro Brandão

Mechanics – Check



k	2
n	6
m	16
<i>ratio</i>	2.667
<i>Opt k</i>	1.848
<i>error opt</i>	0.289
<i>error</i>	0.291

Only Obj_a and Obj_b inserted

2008/10/14
Reading Group - Pedro Brandão

Parameters

		Decrease	Increase
Num. of Hash functions	k	<ul style="list-style-type: none"> • Less hash functions (computation) • Increase in “zeros” ratios → higher false positive 	<ul style="list-style-type: none"> • More hash functions (computation) • Higher change to find 0 bits for non members
Size of filter	m	<ul style="list-style-type: none"> • Less memory needed • Decrease zero bit ratio hole → higher false positive 	<ul style="list-style-type: none"> • More memory needed • Increase zero ” ratio → lower false positive
Num. of elements in set	n	<ul style="list-style-type: none"> • Increase zero ratio → lower false positive 	<ul style="list-style-type: none"> • Decrease zero ratio → higher false positive


2008/10/14
Reading Group - Pedro Brandão




UNIVERSITY OF
CAMBRIDGE
 5 Computer Laboratory

Theory I

$$p' = \left(1 - \frac{1}{m}\right)^{kn} \approx e^{-kn/m} \quad \text{Probability of a bit being zero}$$

$$f' = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k = (1 - p')^k \quad \text{Probability of false positive (approximation)}$$

$$f = \left(1 - e^{-kn/m}\right)^k = (1 - p)^k \quad \text{2nd Aprox. for false positive}$$


2008/10/14
Reading Group - Pedro Brandão


UNIVERSITY OF
CAMBRIDGE
 6 Computer Laboratory

Theory II

- If the array is split in 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



2008/10/14

Reading Group - Pedro Brandão



Minimizing I

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

Minimize f is to minimize g

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

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



2008/10/14

Reading Group - Pedro Brandão



Minimizing II

- Thus for the optimal k

$$k = \ln 2 \cdot (m/n)$$

- The false positive rate is:

$$f = (1/2)^k \approx (1/2)^{m/n \ln 2} \approx (0.6185)^{m/n}$$

- The m for an ϵ false positive rate is:

$$m \geq n \log_2(1/\epsilon) \quad (\text{see paper})$$

- Making $f \leq \epsilon$ and for the optimal k requires

$$m \geq n \log_2 e \cdot \log_2(1/\epsilon).$$

≈ 1.44



2008/10/14

Reading Group - Pedro Brandão



Tricks

- Union \rightarrow OR
- Intersection \rightarrow AND
- Halving size



2008/10/14

Reading Group - Pedro Brandão



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



2008/10/14

Reading Group - Pedro Brandão



Applications types

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



2008/10/14

Reading Group - Pedro Brandão



Examples

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



2008/10/14

Reading Group - Pedro Brandão



13 Computer Laboratory

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



2008/10/14

Reading Group - Pedro Brandão



14 Computer Laboratory

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



2008/10/14

Reading Group - Pedro Brandão



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



2008/10/14

Reading Group - Pedro Brandão



Network Routing II

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



2008/10/14

Reading Group - Pedro Brandão



Packet Routing

- Detecting loops (unicast, multicast)
 - Use a bloom filter in the packet to mark nodes traversed
 - False positive → 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 → well behaved flows punish
 - → change hash functions periodically



2008/10/14

Reading Group - Pedro Brandão



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



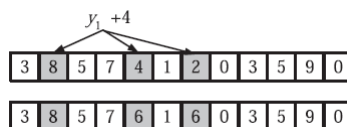
2008/10/14

Reading Group - Pedro Brandão



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



2008/10/14

Reading Group - Pedro Brandão



Measurement II

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



2008/10/14

Reading Group - Pedro Brandão



Thank you

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

