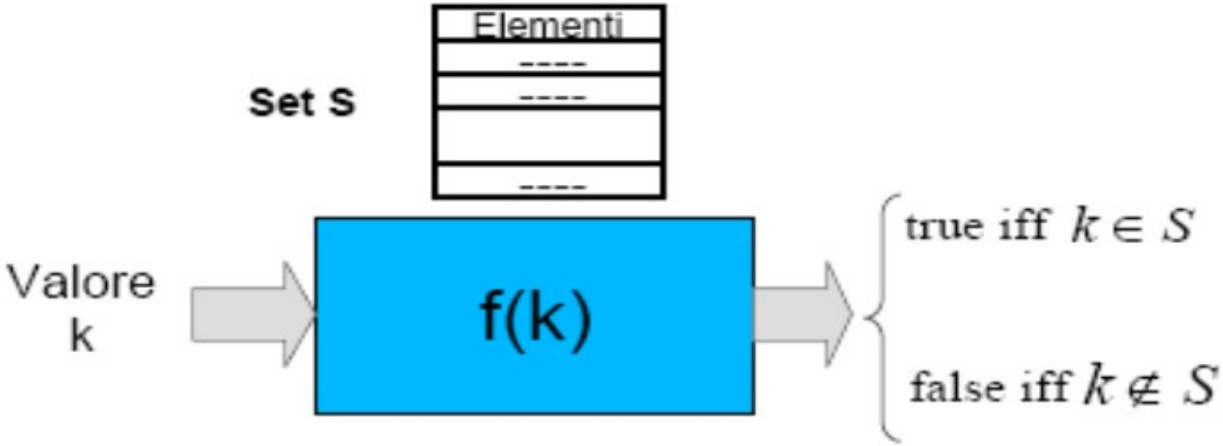
Lezione n.20
22/05/2008
BLOOM FILTERS
Materiale didattico sulla pagina del corso Laura Ricci

BLOOM FILTERS

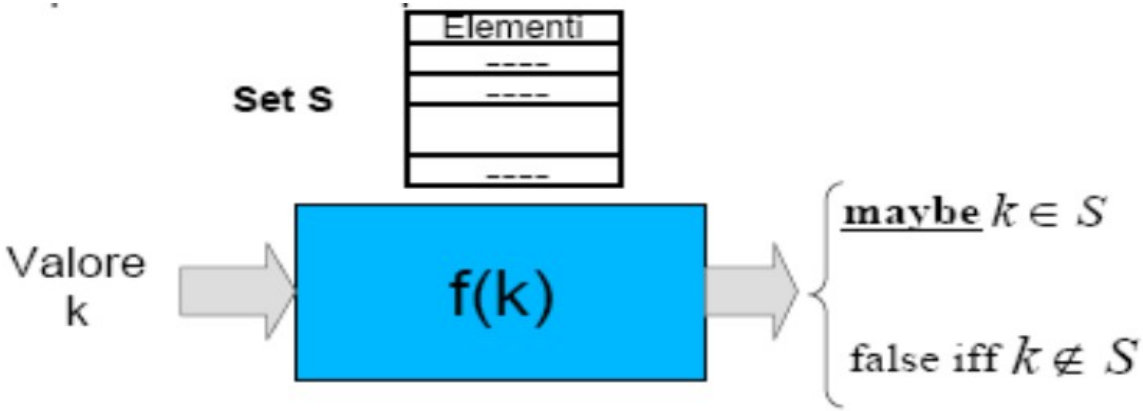
Set Memebership Lookup: Dato un generico insieme, determinare se un certo elemento appartiene all'insieme oppure no



La funzione f è booleana e restituisce vero o falso a seconda della presenza o meno di k nell'insieme dato

BLOOM FILTERS

Se si è disposti a rinunciare ad un certo grado di accuratezza nelle operazioni di look up, a favore dell'efficienza in occupazione di memoria della rappresentazione dell'insieme, una possibile soluzione è data dall'impiego di strutture dati particolarmente compatte: i Bloom Filters

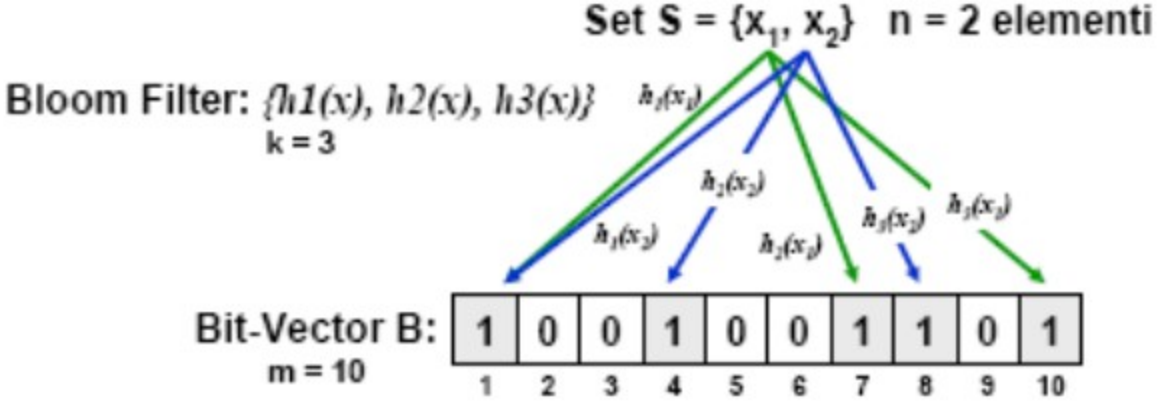


Il Bloom Filter èuna struttura dati molto compatta che fornisce informazioni certe solo sulla non appartenenza di un elemento all'insieme, mentre se da una risposta affermativa, questa non è detto che questa sia corretta (possibilità di falsi positivi)

BLOOM FILTERS

Inserimento di un Elemento in un Bloom Filter

- Un Bloom Filter consente di mappare un insieme $S\{s_1,s_2,\ldots,s_n\}$ di n elementi su un bit vector B di m elementi inizialmente posti a 0 (con m<<n)
- Il filtro è composto da k funzioni hash indipendenti $h_1(x)$, $\ldots h_k(x)$, che producono un valore nel range [1..m]
- L'inserimento nel Bloom Filter di un elemento x_i dell'insieme avviene applicando in sequenza le k funzioni hash $z_1=h_1(x_i),\ldots,z_k=h_k(x_i)$ e ponendo ad 1 gli elementi $B(z_k)$ del bit vector B
- Un bit in B può essere posto ad 1 più di una volta

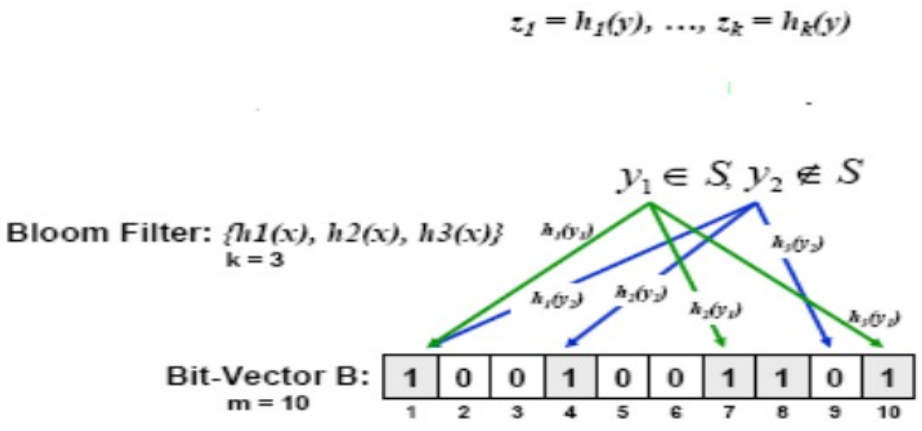


BLOOM FILTERS

LookUp in un Bloom Filter

Per verificare che un generico elemento y appartenga all'insieme S mappato sul Bloom Filter, le k funzioni hash vengono applicate ad y

- Se il risultato di almeno una delle funzioni hash è uguale a 0, l'elemento non appartiene all'insieme
- Se il risultato di tutte le funzioni hash è uguale ad 1, l'elemento appartiene all'insime (possibiltà di falsi positivi)



BLOOM FILTERS

Look Up in un Bloom Filters

- Se l'esito del look up è negativo, allora sicuramente l'elemento non appartiene all'insieme
- Se l'esito del look up è positivo, allora è possibile che l'elemento appartenga all'iniseme, con una certa probabilità di errore
- Possibilità falsi positivi
  - Le funzioni hash calcolate su valori diversi possono portare allo stesso risultato (collisione)
  - Nella fase di look up è possibile che una funzione hash restituisca l'indice di una posizione precedentemente impostata a 1 per un altro elemento dell'insieme

BLOOM FILTERS: FALSI POSITIVI

- Per calcolare la probabilità di un falso positivo
- Consideriamo un insieme di n elementi mappato su un vettore di m bits mediante k funzioni hash
- Dopo che tutti gli elementi sono stati mappati mediante le funzioni hash sul vettore, la probabilità che un specifico bit del vettore sia ancora uguale a 0, è la seguente

$$p' = \left(1 - \frac{1}{m}\right)^{kn} \approx e^{-kn/m}.$$

- Consideriamo ora la probabilità di ottenere un falso positivo. In questo caso ho applicato le k funzioni ed ho ottenuto sempre 1, mentre avrei dover ottenuto almeno una volta uno 0. Da cui si ottinene la probabilità f di falsi positivi

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

BLOOM FILTERS: FALSI POSITIVI

- Probabilità di avere un falso positivo

$$P = (1 - e^{kn/m})^k$$

- Fissando uno o più parametri nella funzione della probabilità di errore, è possibile ottimizzare altri parametri per ottenere le prestazioni desiderate
- Ad esempio, fissati n ed m, è possibile trovare k (numero di funzioni hash) che minimizzi la probabilità di errore

BLOOM FILTERS: OPERAZIONI

- Dati due Bloom Filters, B1 e B2 che rappresentano, rispettivamente gli insiemi S1 ed S2 mediante lo stesso numero di bits e lo stesso numero di funzioni hash, il Bloom filter che rappresenta $S1 \cup S2$ è ottenuto calcolando l'OR bit a bit di B1 e B2
- E' possibile dimezzare la dimensione di un Bloom Filter semplicemente calcolando l'OR della prima metà del vettore con la seconda metà e trascurando poi la seconda metà
- Eliminazione degli elementi dall'insieme: non è corretto azzerare tutti i bit corrispondenti all'applicazione delle k funzioni hash all'elemento eliminato
- Counting Bloom Filters: per ogni entrata del Bloom Filter un contatore, invece di un singolo bit.
  - utilizzati per agevolare la rimozione di elementi da un Bloom Filter
  - al momento dell'inserzione, aumento il contatore
  - al momento dell'eliminazione, decremento il contatore

BLOOM FILTERS: OPERAZIONI

- Dati due Bloom Filters, B1 e B2 che rappresentano, rispettivamente gli insiemi S1 ed S2 mediante lo stesso numero di bits e lo stesso numero di funzioni hash, il Bloom filter che rappresenta $S1 \cap S2$ è approssimato dal Bloom Filter ottenuto calcolando l'and bit a bit dei due Bloom Filters
- Infatti se un bit assume valore uguale ad 1 in entrambe i Bloom filters, questo può accadere perchè
  - quel bit corrisponde ad un elemento $\in S1 \cap S2$ e quindi viene settato ad 1 in entrambe i filtri
  - quel bit corrisponde sia ad un elemento $\in S1 - (S1 \cap S2)$ che ad un elemanto $\in S2 - (S1 \cap S2)$ e quindi non corrisponde ad un elemento nell'intersezione

BLOOM FILTERS E CONTENT DISTRIBUTION

- Utilizzati in Content Distribution Networks

- Set Reconciliation Problem: A possiede un insieme di dati S_A, B un insieme di dati S_B.
- B deve indicare ad A i dati che possiede, in modo che A possa inviargli i dati in S_A-S_B
- B invia ad A un Bloom filter calcolato a partire dai dati in suo possesso
- A scorre gli elementi in proprio possesso e controlla se questi elementi possono appartenere anche a B, mediante il Bloom filter di B
- A causa dei falsi positivi, alcuni elementi in S_A-S_B possono non essere spediti
- Questi elementi possono essere singolarmente richiesti da B ad A

BLOOM FILTERTS: APPLICAZIONI

- Alternativa alle DHT per reti P2P di dimensione modesta
  - Ogni nodo memorizza mediante un Bloom Filter gli elementi memorizzati negli altri nodi della rete

- Query Multi Attributo
  - pubblicazione di informazioni caratterizzati da più chiavi di ricerca (es: CPU, memoria, spazio disco per ricerca di risorse)
  - ogni chiave mappata indipendentemente su un nodo di una DHT
  - query su più attributi (es: CPU=3GHZ and mem=30M,.....)
  - una query (DHT get) per ogni attributo
  - è necessario calcolare l'intersezione dei risultati
  - i risulatati delle singole query rappresenate con bloom filters
  - calcolo intersezione delle query

BLOOM FILTERS: QUERY MULTI ATTRIBUTO

- Problema generale: due peer A, B con gli insiemi di elementi S_A, S_B, calcolare

$$S_A \cap S_B$$

- Il peer B invia ad A un Bloom Filter che codifica i propri elementi.
- Il peer A
  - applica ad ogni elemento del proprio insieme S_A le funzioni hash e controlla se l'elemento può appartenere anche ad S_B utilizzando il Bloom Filter ricevuto da B
  - invia i suoi elementi che possono appartenere anche S_B.
- Presenza di falsi positivi: invio a B di un numero di elementi maggiore di quello contenuto nell'intersezione

BLOOM FILTERS: APPLICAZIONI

Ricerca di Risorse in Reti P2P non strutturate

- Ogni nodo di un overlay P2P mantiene un Bloom filter per ogni arco che lo collega con un peer adiacente sull'overlay network
- Il Bloom Filter indica quali elementi possono essere trovati seguendo quel link, entro un numero limitato d di hops
- Per ogni valore di d , il Bloom Filter codifica le risorse che possono essere raggiunte seguendo quel link, mediante d hops.