

Recherche de filtres de Bloom similaires

Application à la recherche par mots clés basée sur une DHT

NDOMBI TSHISUNGU Christian & **DOAN** Cao Sang
Encadrant: M. **MAKPANGOU** Mesaac, Regal

UPMC

2 Mai 2015

Table de contents

- 1 Présentation
- 2 Filtre de Bloom
 - Insertion
 - Recherche
- 3 Réalisation du travail
- 4 Algorithme des fonctions
 - CREATE_FILTER
 - PUT
 - SEARCH
- 5 Résultat de tests
 - Contexte de test
 - Recherche aléatoire
 - Recherche selective
- 6 Question

Enfin, le texte ! :)

Filtre de Bloom

Insertion dans le filtre de Bloom

IN : x objet à insérer dans le filtre de Bloom B

FUNCTION : $insert(x)$

OUT : \emptyset

for $i = 0 \dots k - 1$ **do**

$j \leftarrow h_i(x)$

if $B_j == 0$ **then**

$B_j \leftarrow 1$

end

end

Filtre de Bloom

Exemple d'insertion

Par exemple, supposons que nous souhaitions ajouter la clé "computer" dans la table B de taille 16 bits, que nous ayons 4 fonctions de hachage $h_i, 0 \leq i < 4$ et que $h_0(\text{"computer"}) = 3$, $h_1(\text{"computer"}) = 8$, $h_2(\text{"computer"}) = 15$, $h_3(\text{"computer"}) = 10$, $h_4(\text{"computer"}) = 11$. Donc, l'état de la table B après l'insertion sera :

15																0
1	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	

TABLE: Exemple filtre de Bloom

Filtre de Bloom

Recherche dans le filtre de Bloom

IN : x objet à tester dans le filtre de Bloom B

FUNCTION : $ismember(x)$

OUT : $bool$

$m \leftarrow true$

$i \leftarrow 0$

while $m \ \&\& \ i \leq k - 1$ **do**

$j \leftarrow h_i(x)$

if $B_j == 0$ **then**

$m \leftarrow false$

end

$i \leftarrow i + 1$

end return m

Filtre de Bloom

Exemple de recherche

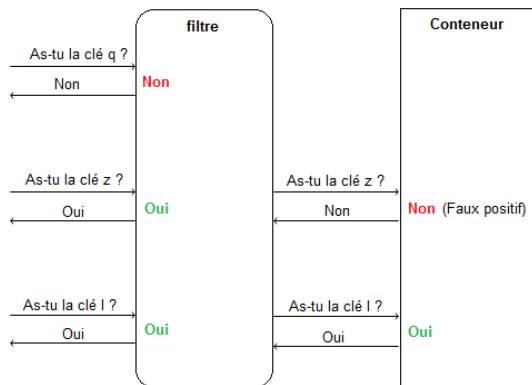


FIGURE: isMember ?

Algorithme des fonctions

Algorithme des fonctions

CREATE_FILTER

IN : $\sum desc$
FUNCTION : $create_filter(\sum desc)$
OUT : B^{512}

```
init( $B^{512}$ )  
 $x \leftarrow FIRST(\sum desc)$   
while  $x \neq \emptyset$  do  
     $i \leftarrow SHA\_256(x)$   
     $j \leftarrow i \bmod 512$   
     $B^{512}[j] \leftarrow 1$   
     $x \leftarrow NEXT(\sum desc)$   
end  
return  $B^{512}$ 
```

Algorithme des fonctions

PUT

IN : filtre de Bloom de taille 512 bits B^{512}

FUNCTION : $put(B^{512})$

OUT : \emptyset

$i \leftarrow \text{MAX_LEVEL}$

$\text{vector}_i \leftarrow \text{CREATE_VECTOR}(B^{512}, i)$

$x \leftarrow \text{FIRST}(VA_file)$

while $x \neq \emptyset$ **do**

if $\text{vector}_i = x$ **then**

 BREAK

end

$x \leftarrow \text{NEXT}(VA_file)$

end

Algorithme des fonctions

PUT(suite)

```
if  $vector_i \neq x$  then
   $VA\_file \leftarrow \text{ADD}(vector_i)$ 
end
for  $i = MAX\_LEVEL \dots 1$  do
  if  $i = 1$  then
     $vector_i \leftarrow \text{CREATE\_VECTOR}(B^{512}, i)$ 
     $\text{CREATE\_FILE}(vector_i, B^{512})$ 
  else
     $vector_i \leftarrow \text{CREATE\_VECTOR}(B^{512}, i)$ 
     $\text{CREATE\_FILE}(vector_i, \text{CREATE\_VECTOR}(B^{512}, i - 1))$ 
  end
end
return  $\emptyset$ 
```

Algorithme des fonctions

SEARCH

IN : filtre de Bloom de taille 512 bits B_{req}^{512}

FUNCTION : $search(B_{req}^{512})$

OUT : $\sum doc$

$i \leftarrow \text{MAX_LEVEL}$

$vector_i \leftarrow \text{CREATE_VECTOR}(B_{req}^{512}, i)$

$tmp \leftarrow \text{CREATE_FILE}(i)$

$x \leftarrow \text{FIRST}(VA_file)$

while $x \neq \emptyset$ **do**

if $vector_i \subseteq x$ **then**

$tmp \leftarrow \text{ADD}(x)$

end

$x \leftarrow \text{NEXT}(VA_file)$

end

Algorithme des fonctions

SEARCH(suite)

```
for  $i = MAX\_LEVEL - 1 \dots 1$  do  
   $vector_i \leftarrow CREATE\_VECTOR(B_{req}^{512}, i)$   
   $x \leftarrow FIRST(FILE(i + 1))$   
   $tmp \leftarrow CREATE\_FILE(i)$   
  while  $x \neq \emptyset$  do  
     $y \leftarrow FIRST(FILE(x))$   
    while  $y \neq \emptyset$  do  
      if  $vector_i \subseteq y$  then  
         $tmp \leftarrow ADD(y)$   
      end  
       $y \leftarrow NEXT(FILE(x))$   
    end  
     $x \leftarrow NEXT(FILE(i + 1))$   
  end  
end
```

Algorithme des fonctions

SEARCH(suite)

```
x ← FIRST(FILE(1))
while x ≠ ∅ do
  y ← FIRST(FILE(x))
  while y ≠ ∅ do
    if  $B_{req}^{512} \subseteq y$  then
       $\sum doc \leftarrow \text{ADD}(\text{FIRST}(\text{FILE}(y)))$ 
    end
    y ← NEXT(FILE(x))
  end
  x ← NEXT(FILE(1))
end
return  $\sum doc$ 
```

Résultat de test

Contexte de test

Résultat de tests

Recherche aléatoire

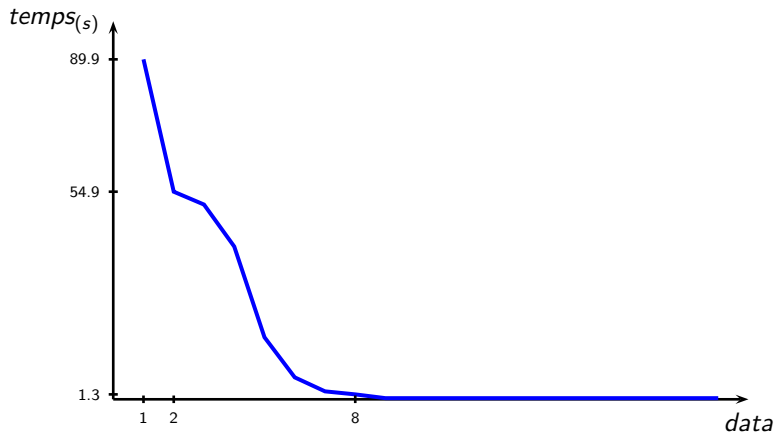


FIGURE: Recherche aléatoire

Résultat de tests

Recherche sélective

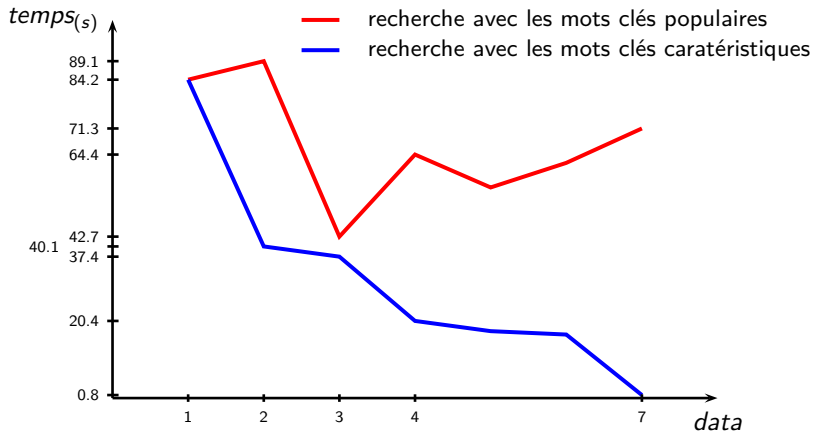


FIGURE: Recherche sélective

Merci de votre attention & Question