

Recherche de mot en parallèle

Thierry Lecroq

Thierry.Lecroq@univ-rouen.fr

Laboratoire d'Informatique, Traitement de l'Information, Systèmes
Université de Rouen

M2 ITA



Plan

- 1 Introduction et définitions
- 2 Recherche de mot

Plan

- 1 Introduction et définitions
- 2 Recherche de mot

PRAM

- nombre illimité de processeurs élémentaires (PE)
- chaque PE
 - possède une mémoire locale de taille constante
 - peut accéder à une grande mémoire partagée
 - est indexé
- en un seul cycle un PE peut
 - accéder à un emplacement de la mémoire partagée
 - ou exécuter une opération de base sur sa propre mémoire

PRAM

Les PE exécutent le même programme sur des données différentes : SIMD

Différentes règles existent pour restreindre les accès à la mémoire partagée :

EREW : Exclude Read Exclude Write

CREW : Concurrent Read Exclude Write

CRCW : Concurrent Read Concurrent Write
en cas d'accès concurrent en écriture

commun : tous les PE écrivent la même valeur

arbitraire : un PE écrit une valeur

priorité : le PE de plus haute priorité écrit une valeur

Mesures de complexités

- temps
- espace
- travail (temps \times nombre de processeurs)

Plan

- 1 Introduction et définitions
- 2 Recherche de mot

Recherche de mot

Théorème

Le recherche des occurrences d'un mot de longueur m dans un texte de longueur n peut s'effectuer en temps constant avec mn processeurs.

Recherche de mot

Lemme

Il ne peut y avoir deux occurrences d'un mot x dans un texte y commençant aux positions i et j si $|j - i| < per(x)$.

Témoin

Soit $0 \leq v \leq m$ un entier qui n'est pas une période de x .
Par définition de la période il doit y avoir un entier k tel que

$$x[k] \neq x[k - v]$$

On dit que k est un **témoin** du fait que v n'est pas une période de x
et on note $W[v] = k$.

Témoin

Lemme

Les valeurs $W[i]$ pour $0 < i < per(x)$ peuvent être calculées à l'aide des $2per(x) - 1$ premières lettres de x .

Preuve

Si $W[i] \geq 2per(x)$ alors soit $w = W[i] \bmod per(x)$

Si $w < i$ alors

$$W[i] \leftarrow w + per(x)$$

sinon

$$W[i] \leftarrow w$$

Phase de recherche

On organise des **duels** entre positions à l'aide de témoins.

PARALLELSM(x, m, y, n)

▷ Réduction du nombre d'occurrences potentielles à $2n/m$

1 **for** $j \leftarrow 1$ **to** $\log m - 1$ **do**

2 **for** $i \leftarrow 1$ **to** $n/2^j$ **do**

3 En parallèle : processeur i réduit le nombre d'occurrences

 ▷ dans $y[(i - 1)2^j .. i2^j - 1]$ à 1

 ▷ Test naïf des candidats restants

4 **for** $i \leftarrow 0$ **to** $n - 1$ **do**

5 En parallèle ; processeur $i = k \bmod m/2$ teste l'occurrence

 ▷ potentielle dans $y[k \times (m/2) .. (k + 1) \times (m/2) - 1]$

Complexité

- temps : $O(\log m)$
- processeurs : $O(n)$
- travail : $O(n \log m)$ (non optimal)

Il est possible de rechercher toutes les occurrences d'un mot de longueur m dans un texte de longueur n en temps constant avec $O(n)$ processeurs.

La phase de prétraitement est en temps $O(\log \log m)$ avec $(m/(\log \log m))$ processeurs.

(basé sur la notion d'échantillon déterministe)