

## Algorithmique TD 5: Analyse des défauts de cache

### Exercice 1. Somme et somme des carrés

On considère un tableau  $T$  de  $n$  entiers machine (de la taille d'un mot machine). Le cache de la machine possède des lignes de  $L$  mots et a une taille totale  $Z$ .

On souhaite calculer la somme des éléments du tableau ainsi que la somme de leurs carrés.

a. En supposant que l'on fasse une boucle pour la somme puis une boucle pour le calcul de la somme des carrés, quel est le nombre de défauts de cache généré quand

1.  $n < Z$
2.  $n > Z$

b. Mêmes questions lorsque l'on ne fait qu'une boucle pour les deux calculs.

### Exercice 2. Parcours d'un tableau

On souhaite parcourir un tableau de taille  $n$  de trois façons :

1. incrémentalement du début à la fin par pas de 1
2. décrémentalement de la fin au début par pas de -1
3. incrémentalement avec un stride, par pas de  $s$ , puis en recommençant au début avec un décalage de 1.

a. Pour chacun de ces trois parcours, analyser les défauts de cache lorsque

1.  $n < Z$
2.  $n > Z$

### Exercice 3. Recherche Dichotomique

Quelle est le nombre de défauts de cache de l'algorithme de recherche dichotomique dans un tableau trié de  $n$  entiers ?

### Exercice 4. Parcours de matrices

En  $C$ , on stocke en général une matrice  $A$  de dimensions  $m \times n$  (un tableau bi-dimensionnel) par un tableau unidimensionnel en stockant les coefficients lignes par ligne. L'accès à l'élément  $(i, j)$  de la matrice se fait donc par  $T[i * n + j]$ .

On souhaite calculer dans un tableau  $S$  de taille  $m$  le maximum des éléments de chaque ligne de  $A$ ,  $S_i = \max_{j=1 \dots n}(a_{i,j})$ , avec l'algorithme suivant :

```
Pour i <- 0..m-1
  S[i] <- A[i*n]
  Pour j <- 1..n-1
    Si S[i] < A[i*n+j]
      S[i] <- A[i*n+j]
```

- a. Combien ce programme génère-t'il de défauts de cache? Est-il bon?
- b. On souhaite désormais calculer le minimum de élément de chaque colonne dans un tableau  $T$ . Proposer un algorithme qui effectue  $O(mn/L)$  défauts de cache.
- c. On souhaite désormais calculer les tableaux  $T$  et  $S$  simultanément. Proposer un algorithme. Combien de défauts de cache fait-il?
- d. Proposer un nouvel algorithme faisant  $(1 + \varepsilon)\frac{mn}{L} + O(\frac{m+n}{L})$  défauts où  $\varepsilon$  est petit quand  $Z \gg L^2$ .

### Exercice 5. Transposition de matrices

---

On s'intéresse à des algorithmes calculant la transposée d'une matrice :  $B \leftarrow A^T$  où  $A$  est  $m \times n$ .

- a. Combien de lignes de caches sont occupées par les entrées-sorties? En déduire une borne inférieure de la complexité cache pour cette opération.
- b. Proposer un algorithme (naïf) traitant la matrice  $A$  ligne à ligne.
- c. Donner sa complexité en défauts de cache selon que
  1. les deux matrices tiennent dans le cache :  $2n^2 < Z$ .
  2. une ligne de  $A$  et une colonne de  $B$  tiennent dans le cache :  $nL + n < Z$ .
  3. une colonne de  $B$  ne tient pas dans le cache :  $nL > Z$ .
- d. Proposer un algorithme Cache aware, basé sur une découpe en blocs  $K \times K$ . On supposera que  $K$  est suffisamment petit pour que 2 blocs  $K \times K$  tiennent dans le cache. Analyser sa complexité cache.
- e. Proposer une variante récursive cache-oblivious. Analyser sa complexité cache.

### Exercice 6. Produit de matrices

---

On s'intéresse ici à la complexité cache d'algorithmes de multiplication de matrices. On supposera qu'en entrée, la matrice  $C$  est initialisée à zéro. Par commodité, on notera  $A[i, j]$  l'accès au coefficient  $i, j$  de la matrice en tenant compte du stockage en ligne :  $A[i, j]$  signifie  $A[i*n+j]$ .

On considère les deux algorithmes suivants :

ALGO MMijk

Pour  $i \leftarrow 0..n-1$

  Pour  $j \leftarrow 0..n-1$

    Pour  $k \leftarrow 0..n-1$

$C[i, j] \leftarrow A[i, k] * B[k, j]$

ALGO MMikj

Pour  $i \leftarrow 0..n-1$

  Pour  $k \leftarrow 0..n-1$

    Pour  $j \leftarrow 0..n-1$

$C[i, j] \leftarrow A[i, k] * B[k, j]$

- a. Calculer la complexité en défauts de cache de chacun d'eux lorsque  $3n^2 \gg Z$ .
- b. Qu'en est-il avec l'algorithme de Strassen?