

TD 2 Algorithmique Parallèle

Master Method

Approximer si possible $T(n)$ pour

1. $T(n) = 4T(n/2) + n$
2. $T(n) = 4T(n/2) + n^2$
3. $T(n) = 4T(n/2) + n^3$
4. $T(n) = 4T(n/2) + n^2/\log n$

Merge Sort parallèle

1. Le programme de la fig. 1 donne le schéma d'un premier MergeSort parallèle. MergeSort trie le tableau $A[n]$ dans le tableau $B[n]$.
La fonction $Merge(int *C, int *A, int *B, int a, int b)$ effectue la fusion des tableaux triés $A[a]$ et $B[b]$ dans le tableau $C[a + b]$. On admettra que son temps de calcul séquentiel est $n = a + b$.
Peut-on supprimer l'instruction sync ?
Vérifier que le travail du MergeSort $S_1(n)$ est $\Theta(n \log n)$. Calculer le span $S_\infty(n)$, et l'indice de parallélisme. Que peut-on en conclure ?
2. Pour paralléliser l'étape de fusion (merge), on utilise le programme de la fig. 2 : si A est le plus grand tableau, on sélectionne d'abord l'élément médian de A , puis on le cherche dans B (*BinarySearch*). Ensuite, on fusionne les sous tableaux inférieurs d'une part, et supérieurs d'autre part. On utilisera le fait (vérification facile) que dans le pire cas, le nombre d'éléments dans le plus grand des deux P_Merge est $3n/4$, où $n = a + b$. Calculer le span $T_\infty^M(n)$ pour ce programme.
3. Calculer le span $T_\infty(n)$ pour un programme MergeSort parallèle qui exploite la fusion parallèle.

```
cilk void MergeSort(int *B, int *A, int n) {
    if (n == 1) {
        B[0] = A[0];
    } else {
        int *C;
        C = (int *)allocate(n*sizeof(int));
        spawn MergeSort(C, A, n/2);
        spawn MergeSort(C+n/2, A+n/2, n-n/2);
        sync;
        Merge(B, C, C+n/2, n/2, n-n/2);
    }
}
```

FIGURE 1 – Programme simple

```
cilk void P_Merge(int *C, int *A, int *B, int a, int b) {
    if(a < b) {
        spawn P_Merge(C, B, A, b, a);
    } else if (a==1) {
        < Cas de base >
    } else {
        int ma = a/2;
        int mb = BinarySearch(A[ma], B, b);
        spawn P_Merge(C, A, B, ma, mb);
        spawn P_Merge(C+ma+mb, A+ma, B+mb, a-ma, b-mb);
        sync;
    }
}
```

FIGURE 2 – Fusion parallèle