

INF8541: Devoir #1

(À remettre le 18 septembre)

1. Parallélisme récursif

Écrivez, dans la notation d'Andrews, des procédures qui, conjointement, vont permettre de calculer le produit scalaire de deux vecteurs de longueur n en utilisant une approche avec parallélisme récursif (approche diviser-pour-régner bien équilibrée).

L'interface de la procédure principale sera la suivante:

```
int produit_scalaire( int a[], int b[], int n );
```

Rappel:

$$\text{produit_scalaire}(a, b, n) = \sum_{i=0}^{n-1} a[i] \times b[i]$$

2. Parallélisme itératif

Soit les fonctions suivantes:

```
int inf( int i, int n ){ return( i      * (n / P)      ); }
int sup( int i, int n ){ return( (i+1) * (n / P) - 1 ); }

void somme_vecteurs_seq( int a[], int b[], int c[], int n )
{
    int i;
    for [i = 0 to n-1] {
        c[i] = a[i] + b[i];
    }
}
```

Utilisez ces procédures pour écrire une fonction ayant l'interface suivante et qui effectuera la somme de deux vecteurs de façon parallèle en générant P processus itératifs, où chacun traitera un sous-intervalle des tableaux:

```
void somme_vecteurs(int a[], int b[], int c[], int n)
```

3. Parallélisme itératif avec variable globale

Soit la solution présentée plus bas pour le deuxième exercice de la série d'exercices #1 (fait en classe), où il fallait calculer la somme de 1 à n à l'aide d'une approche avec parallélisme itératif. Cette solution est-elle correcte? Expliquez brièvement votre réponse. (C'est votre justification qui m'intéresse...)

```
int grand_total;

void somme( int i, int j )
{
    int k, tot = 0;

    for [k = i to j]
        tot += k;

    grand_total = grand_total + tot;
}

int somme_1_a_n( int n )
{
    int i;

    grand_total = 0;
    /* On cree les P processus iteratifs. */
    co [i = 0 to P-1] {
        somme( inf(i, n), sup(i, n) );
    }
    return( grand_total );
}
```