

UNIVERSITÉ DE SHERBROOKE  
DÉPARTEMENT ET D'INFORMATIQUE

Devoir # 2

**IFT 630**

Processus concurrents et parallélisme

**Devoir à remettre au plus tard le 17 mars 2021**

1. Vous devez résoudre les problèmes suivants à l'aide du langage C, C++ ou Python et la bibliothèque de communication par messages MPI.

- (a) Implanter la multiplication de matrice tel que décrite dans l'énoncé du devoir 1.
- (b) Implanter un programme qui résout l'équation de Laplace à l'aide de la méthode appelée « itération de Jacobi ». Une solution à ce problème a été présentée dans le cours et est disponible dans le répertoire public. Vous pouvez vous inspirer de cette solution pour concevoir la vôtre. Votre solution doit avoir une structure maître/esclaves. Votre processus maître initialise la matrice, distribue les valeurs nécessaires aux calculs aux processus esclaves et affiche le résultat.

Pour résoudre ce problème, vous devez créer une matrice dont les « côtés » contiennent la valeur limite à atteindre et l'intérieur, les valeurs de départ à manipuler. La méthode de Jacobi itère sur les valeurs internes en les remplaçant par la moyenne des quatre voisins grâce à la formule :

$$new[i, j] = \frac{old[i - 1, j] + old[i + 1, j] + old[i, j - 1] + old[i, j + 1]}{4}$$

Les itérations terminent lorsque les valeurs internes sont assez près (epsilon) de la limite fixée.

Exemple de grille pour le calcul dont la limite à atteindre est -1 et les valeurs internes varient entre 1 et 4.

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| -1 | -1 | -1 | -1 | -1 | -1 |
| -1 | 1  | 1  | 1  | 1  | -1 |
| -1 | 2  | 2  | 2  | 2  | -1 |
| -1 | 3  | 3  | 3  | 3  | -1 |
| -1 | 4  | 4  | 4  | 4  | -1 |
| -1 | -1 | -1 | -1 | -1 | -1 |

2. Vous devez résoudre le problème suivant à l'aide du langage C ou C++ et les bibliothèques multi-fils et OpenCL.

Implanter un programme qui effectue une attaque brute sur un système d'encodage simple. Le système de codage en question est donné en annexe. Il prend en entrée une chaîne de caractères contenant exactement sept caractères constituée seulement de lettres minuscules, l'encode en se servant de la chaîne elle-même comme clé et produit un message codé de sept caractères.

Vous devez développer un programme qui effectue une attaque brute sur ce code. Votre programme devra s'exécuter en trois versions : une séquentielle, une parallèle sur les coeurs d'un ordinateur normal (multi-fils en C++ ou autre langage) et une autre en OpenCL qui permettra d'utiliser de multiples coeurs d'une carte graphique. Vous devez me produire des résultats sur les temps requis pour briser différents mots de passe avec la version séquentielle de votre programme et les versions parallèles. Expliquer les différences dans les temps de traitement.

#### Attention

Le système d'encodage peut produire des collisions. Vous n'avez pas à en tenir compte. Quand cela se produit essayer une autre chaîne de caractères.

## Programme pour l'encodage des mots

```
/******
   Cette fonction effectue une substitution de
   caracteres en additionnant une valeur
   *****/
string ADD(string mot, int cle)
{
    int taille;

    taille = mot.length();
    for (int i=0; i< taille; i++)
        mot[i] = mot[i]+cle*i;
    return mot;
}

/******
   Cette fonction effectue un ou exclusif
   entre la chaine de caracteres et la cle
   *****/
string XOR(string mot, string cle)
{
    int taille;
    taille = mot.length();

    for (int i=0; i< taille; i++)
        mot[i] = mot[i]^cle[i];
    return mot;
}

/******
   Cette fonction effectue un decalage circulaire
   vers la droite de "nb" caracteres
   *****/
string decale(string mot, int nb)
{
    char t[nb];
    int max = mot.length()-1;

    for(int i=0; i<nb; i++)
        t[i] = mot[max-i];
    for (int i = max; i>=nb; i--)
        mot[i] = mot[i-nb];
    for (int i=0; i<nb; i++)
        mot[i] = t[nb-i-1];
    return mot;
}

/******
   Cette fonction effectue une translation
   entre les caracteres du mot d'une distance
   deduite de la cle
   *****/
```

```

*****/
string echange(string mot, int cle)
{
    int taille = mot.length();

    for(int i=0; i+cle < taille; i++)
    {
        char temp;
        temp = mot[i];
        mot[i] = mot[i+cle];
        mot[i+cle] = temp;
    }
    return mot;
}

/*****/
Cette fonction genere une cle numerique
*****/
int getCle(string mot)
{
    int cle, temp;
    temp = 0;
    for (int i=0; i<mot.length() ; i++)
        temp = temp + (unsigned int) mot[i];
    cle = (temp % 4)+1;
    return abs(cle);
}

/*****/
Cette fonction encode le message
*****/
string encode(string mot)
{
    string copie;
    int cle;

    copie = mot;
    for (int j=0; j<3; j++)
    {
        cle = getCle(mot);
        mot = decale(mot, cle/2);
        mot = ADD(mot, cle);
        mot = echange(mot, cle);
        mot = XOR(mot, copie);
    }
    return mot;
}

```