Contrôle n°1 (Mardi 03 mai 2022) - Programmation Parallèle - L3 MIAGE

Exercice 1
Écrire un programme permettant de générer 1 million de nombres aléatoires et de calculer la moyenne de ces nombres. Vous pouvez utiliser des tableaux. Vous afficherez le résultat de la moyenne à la fin du programme, ainsi que le temps de calcul total (en secondes). Vous écrirez une version du programme en mode séquentiel (sans paralléliser) et une autre version du programme en utilisant la parallélisation. L'utilisateur pourra choisir le nombre de processeurs max dans la ligne de commande pour lancer le programme. Vous pouvez utiliser le langage de programmation que vous voulez (de préférence Perl).

Exercice 2
Ecrire un autre programme permettant de paralléliser le téléchargement de 2 fichiers:
http://ftp.sra.ebi.ac.uk/vol1/fastq/SRR130/005/SRR13015805/SRR13015805_1.fastq.gz
http://ftp.sra.ebi.ac.uk/vol1/fastq/SRR130/005/SRR13015805/SRR13015805_2.fastq.gz
Afficher le temps total de calcul.

Notez qu'il y a plusieurs façons de télécharger des fichiers à partir d'un programme. Par exemple, en langage Perl, on peut faire appel à des commandes systèmes (wget). Vous pouvez regarder sur internet comment utiliser la commande wget (https://doc.ubuntu-fr.org/wget), et la fonction system de Perl (https://blog.udemy.com/perl-system-command/).

Exercice 3
Écrire un programme avec un compteur global "compt" (initialisé à 20), et qui crée deux threads :
- Le premier thread itère l'opération suivante : on incrémente un deuxième compteur "compt1" et on attend un temps aléatoire entre 1 et 8 secondes, avant d'afficher la valeur de ce deuxième compteur.
- Le deuxième thread affiche toutes les trois secondes la valeur d'un troisième compteur "compt2" (incrémenté avec la valeur de compteur global +1).
Les deux threads se terminent lorsque les compteurs "compt1" et "compt2" ne dépassent pas valeur limite de 100.

Note: les fonctions wait() ou sleep() existent dans beaucoup de langages de programmation.

Exercice 4
Créer un programme qui a en variable globale un tableau de N double, avec N=100. Dans le "main", le tableau sera initialisé avec des valeurs réelles aléatoires entre 0 et 100, sauf les valeurs tableau[0] et tableau[99] qui valent 0.
Le programme crée alors deux threads :
- Le premier thread remplace chaque valeur tableau[i], avec i = 1,2,...,98 par la moyenne (tableau[i-1]+tableau[i]+tableau[i+1])/3. Il attend ensuite un temps aléatoire entre 1 et 3 secondes ;
- Le deuxième thread affiche le tableau toutes les 4 secondes.

Exercice 5
Créer un programme contenant un tableau à deux dimensions de taille 100 (exemple: tab[100][100]), initialisé avec des valeurs réelles aléatoires entre 1 et 90. Le programme permettra d'optimiser le calcul de la moyenne des moyennes de chaque colonne (exemple de colonne pour j compris entre 1 et 100: tab[0][i]). Le programme pourrait également afficher le temps total d'exécution.