

Introducción a la Programación Paralela - Actividad Java

Realizá un informe corto a partir de los ejercicios en máquina que figuran debajo. Es necesario también entregar el código implementado. Recordá entregar un código que sea fácilmente ejecutable en otra máquina (podés adicionar JARs, scripts auxiliares, o lo que consideres) y que sea legible (sin código de prueba comentado, limpio, con nombre de variables representativos, etc).

Dado el código siguiente, que implementa el algoritmo de paralelización clásico de matrices:

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 1024
#define SEED 6834723
int main() {
    int i, j, k;
    double A[SIZE][SIZE];
    double B[SIZE][SIZE];
    double C[SIZE][SIZE];
    srand48(SEED);
    for (i=0; i < SIZE; i++)
        for (j=0; j < SIZE; j++) {
            C[i][j] = 0;
            A[i][j] = drand48();
            B[i][j] = drand48();
        }
    for (i=0; i < SIZE; i++)
        for (j=0; j < SIZE; j++)
            for (k=0; k < SIZE; k=k+1)
                C[i][j] = C[i][j] + A[i][k] * B[k][j];
    printf("Fin: %lf\n", C[0][0]);
    return 0;
}
```

Luego, realizar los siguientes ejercicios:

- Generá una versión secuencial escrita en Java.
- Desarrollá una versión paralela de a) utilizando la API de concurrencia de Java, más específicamente el `ExecutorService`. Queda a tu criterio a qué nivel del bucle paralelizar. No utilices `virtual threads`.
- Desarrollá una versión paralela de b) utilizando el framework `ForkJoin` de Java visto en teoría.
- Realizá una comparación de tiempo de ejecución de las tres alternativas, variando la cantidad de hilos utilizados para las variantes paralelas.
- ¿Cómo reimplementarías el código de b) para utilizar `virtual threads`? Podés guiarte con el ejemplo de código subido al material de la materia.

- f) Realizá los puntos a)-e) sobre el código de Nqueens que trabajamos en la primer entrega.

Nota: Para obtener un tiempo estimado de ejecución, realizá varias ejecuciones de cada variante con un número determinado de hilos. Luego promediá estos tiempos para obtener el valor estimativo. Si el desvío estándar de los experimentos es aún grande, añadir más ejecuciones hasta tener un promedio estable. Recordá describir el CPU del hardware utilizado (cantidad de núcleos y velocidad en MhZ).