

Práctica Nro. 2

Programación con Pthreads

Compilar en Linux gcc:

`gcc -pthread -o salidaEjecutable archivoFuente`

Calcular el speedup y la eficiencia del algoritmo paralelo respecto del algoritmo secuencial.

1. Paralelizar la multiplicación de matrices cuadradas de $N \times N$. Obtener el tiempo de ejecución para $N=512, 1024$ y 2048 . Ejecutar con 2 y 4 threads.
2. Paralelizar un algoritmo que cuente la cantidad de veces que un número X aparece dentro de un vector de tamaño N . Al finalizar, la cantidad de ocurrencias del número X debe quedar almacenada en una variable llamada *ocurrencias*. Ejecutar con 2 y 4 threads.
3. Paralelizar un algoritmo que calcule el valor promedio de los números almacenados en un vector de tamaño N . Ejecutar con 2 y 4 Threads.
4. Paralelizar la búsqueda del mínimo y el máximo valor en un vector de tamaño N . Ejecutar con 2 y 4 Threads.
5. Paralelizar un algoritmo que obtenga la intersección de dos conjuntos de números enteros. Ejecutar con 2 y 4 Threads. Tener en cuenta la siguiente propiedad:

$$A \cap B = A \cap (B_0 \cup B_1 \cup B_2 \dots \cup B_n) = (A \cap B_0) \cup (A \cap B_1) \cup (A \cap B_2) \dots \cup (A \cap B_n)$$

6. Paralelizar un algoritmo que ordene un vector de N elementos por mezcla. Ejecutar con 2 y 4 Threads.