

Introducción a la Computación Paralela – Semestre 1 de 2022

Taller de repaso de punteros y operaciones con vectores

Objetivo: Conocer las formas para manejar memoria que ofrece C/C++ y distinguir usos posibles en el contexto de la computación científica.

Se requiere el uso de una función para contar el tiempo de ejecución de un conjunto de instrucciones de C/C++. Un ejemplo a continuación:

```
auto t1 = high_resolution_clock::now();
    long_operation();
    auto t2 = high_resolution_clock::now();
    /* Getting number of milliseconds as an integer. */
    auto ms_int = duration_cast<milliseconds>(t2 - t1);
```

Construir un programa que calcule la multiplicación de dos matrices cuadradas de $N \times N$ elementos. El programa debe leer el valor de N mediante argumentos de línea de comando (usar `argv` y `argc`). Las matrices deben tener valores aleatorios. Las matrices deben guardarse usando un vector de una dimensión siguiendo el orden de principal de fila.

Para una matriz de $N \times N$ con $N = 4$:

a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_{30}	a_{31}	a_{32}	a_{33}

La representación unidimensional sería:

a_{00}	a_{01}	a_{02}	a_{03}	a_{10}	a_{11}	a_{12}	a_{13}	a_{20}	a_{21}	a_{22}	a_{23}	a_{30}	a_{31}	a_{32}	a_{33}
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

Con la forma de calcular el índice correspondiente a cada elemento de la matriz en el vector (ni) dada por:

$$ni = fila * N + columna$$

1. Utilizar vectores generados utilizando la función `malloc()`
2. Utilizar vectores generados utilizando el operador `new()`
3. Utilizar vectores generados utilizando la clase `vector<>`
4. Utilizar vectores generados utilizando la clase `vector<vector<>>`
5. Organizar los conjuntos de operaciones por bloques y calcular el tiempo que demora cada bloque para comparar el desempeño de los diferentes mecanismos.
6. Presentar los resultados y concluir.
7. Responder las siguientes preguntas:

- Que es un operador de dereferenciación y que es un operador de dirección de memoria. ¿Cómo se relacionan?
- Si se usan estructuras y punteros ¿Cuál es la diferencia entre usar el operador . Y el operador -> ?

Entregar código fuente y un documento en PDF con las respuestas.