Universidad Católica
San Pablo

Ciencia de la Computación

Computación Paralela y Distribuida

Informe Tarea 01

Entregado el 24/08/2022

Moises Alberto Salazar Machaca

Semestre VIII

2022-2

"El alumno declara haber realizado el presente trabajo de acuerdo a las normas de la Universidad Católica San Pablo"

_____

Introducción:

Se realizará un experimento para dos fragmentos de algoritmos, los cuales recorren una matriz de tamaño N.

```
// Algoritmo 1

for (int j = 0; j < MAX; j++)
{
    for (int i = 0; i < MAX; i++)
    {
        y[i] += A[i][j] * x[j];
    }
}

// Algoritmo 2

for (int j = 0; j < MAX; j++)
{
    for (int i = 0; i < MAX; i++)
    {
        y[i] += A[i][j] * x[j];
    }
}
```

Estos son los dos algoritmos que se van a evaluar en el experimento.

Experimento:

El experimento constara de evaluar cual de estos es mas rápido, pero sabiendo que su comportamiento es el mismo, y complejidad de n^2, pero veremos la diferencia en el acceso a memoria, pero refiriéndonos a la memoria cache.

Entonces la métrica de medida será el tiempo de ejecución de cada fragmento de código del algoritmo, y viendo quien hace más:

- Cache Hit: Cuando encuentra la variable en la memoria cache.
- Cache Miss: Cuando no encuentra la variable en la memoria cache y tiene que buscarlo en la memoria física.

Realizando el experimento tenemos los siguientes datos:

Conclusiones:

Siguiente estas métricas se puede concluir que el algoritmo 1 hace menos cache miss, ya que cuando accede a la memoria de A[i][j] traerá todos los consecutivos hasta que se llene la memoria cache, o hasta donde se ocupe.

El algoritmo 1 es mas eficiente ya que recorre la matriz de forma secuencial ósea en otras palabras fila por fila.

El algoritmo 2 lo realiza columna por columna.

Por este motivo el algoritmo 2 hace mas cache miss, porque en un momento se va llenar la memoria cache, y cuando va querer la fila N, que aun no esta cargada realizara la cache miss, y esta diferencia se ve cuando tenemos una matriz de gran tamaño.

0 0	0 1	0 2
1 0	1 1	1 2

Para ser mas especifico, supongamos que tenemos una matriz de 3x4.

Para el algoritmo 1, solo hará 2 cache miss, porque en el primero traerá la matid de la matriz y luego cuando quiera acceder a [2 0] hará el segundo

cache miss, pero traerá la otra parte de la matriz de igual manera secuencialmente.

2 0	2 1	2 2
3 0	3 1	3 2

El algoritmo 2 hará igual si primer cache miss donde tendrá, [0 0] y [1 0]

0 0	0 1	0 2
1 0	1 1	1 2

Para acceder [2 0], no lo encontró y hará otro cache miss, y obtendrá [2 0], [3 0]

2 0	2 1	2 2
3 0	3 1	3 2

Lugo para [1, 1], no lo encontrara y hará otra vez cache miss, y siguiendo esta lógica, en total hará 6 cache miss en comparación del algoritmo 1 que solo hizo 2 cache miss.