

Introduccion al HPC

Practica 2: Recorrido de matrices

*Merino Ortega Angel Nahum

**Parra Grimaldi Christopher Omar

Abstract— Arrays are stored in memory contiguously, meaning that the elements of the array are stored one after another in adjacent memory locations. This allows fast access to array elements using consecutive memory addresses. To improve performance when working with arrays, it is important to take full advantage of the memory hierarchy and minimize accesses to main memory, which is slower compared to the cache. Therefore, in this work we will analyze the combination of conditionals for filling matrices. In this exercise we will develop desktop tests to demonstrate the route used in different combination for filling matrices.

Palabras Clave- Matrices, Combinación, Multiplicación, Memoria, prueba de escritorio.

I. INTRODUCCION

Una prueba de escritorio, también conocida como "prueba de escritorio" o "evaluación de escritorio", es un método de evaluación utilizado en diversos campos, como la informática, la programación, la ingeniería, las matemáticas y otras disciplinas técnicas y científicas. El objetivo principal de una prueba de escritorio es verificar la corrección y eficacia de un algoritmo, un programa, una solución matemática o cualquier proceso lógico sin necesidad de implementarlo en un entorno de ejecución real.

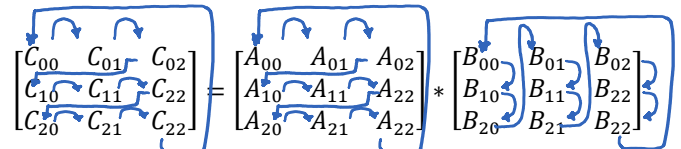
II. EJERCICIO DE EJEMPLO

Durante la clase del 13 de septiembre de 2023 se mostró un ejemplo de cómo hacer la prueba de escritorio. Este ejemplo será tomado de base para la posterior realización del ejercicio.

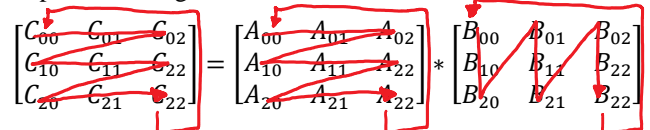
El ejercicio fue basado en el siguiente código

```
for(i=0;i<n,i++){
    for(j=0;j<n,j++){
        for(k=0;k<n,k++){
            C[i][j] = A[i][j]*B[k][j]
        }
    }
}
```

El desplazamiento por cada iteración de i, j y k es el siguiente



El desplazamiento general de cada matriz es el siguiente



III. DEMOSTRACION CON LOS DOS ORDENES MAS LENTOS

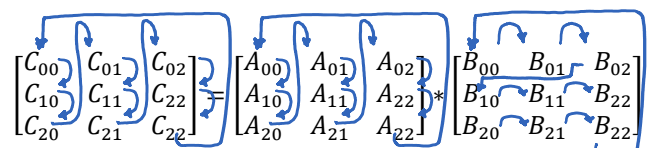
Durante la clase del 13 de septiembre de 2023 se mostró un ejemplo de cómo hacer la prueba de escritorio. Este ejemplo será tomado de base para la realización de la demostración.

Además de esto se usarán los datos de la primera practica donde se obtuvo que los 2 órdenes más lentos fueron: kji y kij

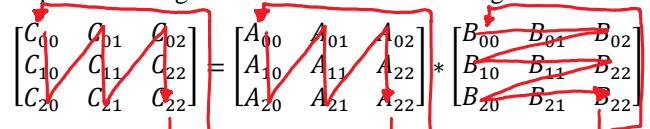
a) Para kji

El ejercicio fue basado en el siguiente código

```
for(k=0;k<n,k++){
    for(j=0;j<n,j++){
        for(i=0;i<n,i++){
            C[i][j] = A[i][j]*B[k][j]
        }
    }
}
```



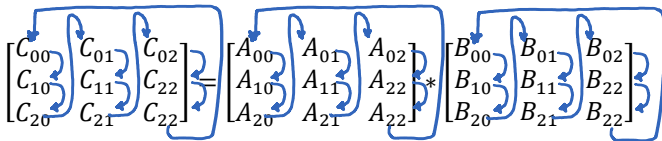
El desplazamiento general de cada matriz es el siguiente



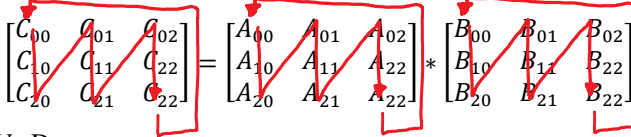
b) Para jki

El ejercicio fue basado en el siguiente código

```
for(j=0;j<n,j++){
    for(k=0,k<n,k++){
        for(i=0,i<n,i++){
            C[i][j] = A[i][j]*B[k][j]
        }
    }
}
```



El desplazamiento general de cada matriz es el siguiente



IV. DEMOSTRACION CON LOS DOS ORDENES MAS RAPIDOS

Durante la clase del 13 de septiembre de 2023 se mostró un ejemplo de cómo hacer la prueba de escritorio. Este ejemplo será tomado de base para la realización de la demostración.

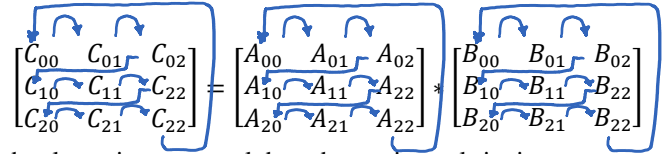
Durante la clase del 13 de septiembre de 2023 se mostró un ejemplo de cómo hacer la prueba de escritorio. Este ejemplo será tomado de base para la realización de la demostración.

Además de esto se usarán los datos de la primera practica donde se obtuvo que los 2 órdenes más lentos fueron: ikj y jik

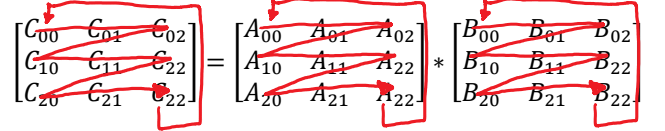
a) Para ikj

El ejercicio fue basado en el siguiente código

```
for(i=0,i<n,i++){
    for(k=0,k<n,k++){
        for(j=0,j<n,j++){
            C[i][j] = A[i][j]*B[k][j]
        }
    }
}
```



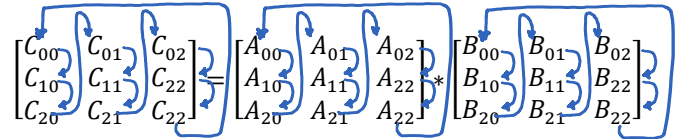
El desplazamiento general de cada matriz es el siguiente



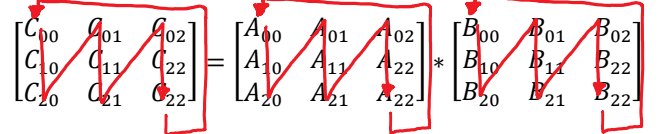
b) Para jik

El ejercicio fue basado en el siguiente código

```
for(j=0;j<n,j++){
    for(i=0,i<n,i++){
        for(k=0,k<n,k++){
            C[i][j] = A[i][j]*B[k][j]
        }
    }
}
```



El desplazamiento general de cada matriz es el siguiente



CONCLUSION

En esta actividad observamos de forma práctica por que el algunos de los mejores métodos fueron los que seguían el direccionamiento de memoria tal y como era en C. Por otro lado, algunos de los resultados no fueron tan acertados y no seguían el direccionamiento de C, esto se cree que pudo ser debido a que para la actividad se usaron dos computadoras diferentes causando así diferentes resultados. Aun así, la mayoría de estos fueron correctos siendo el caso de ikj el mejor resultado y de jki uno de los peores resultados debido al direccionamiento.

REFERENCIAS