

Instituto Politécnico Nacional
Escuela Superior de Cómputo



Desarrollo de Sistemas Distribuidos

Tarea 3. Multiplicación de matrices utilizando paso de mensajes

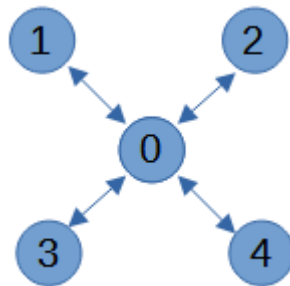
Nombre: Sampayo Hernández Mauro

Grupo: 4CV1

Profesor: Pineda Guerrero Carlos

Introducción:

Se realizará a continuación la demostración de la compilación y ejecución de un programa Cliente-Servidor que calcula la multiplicación de dos matrices, A y B mediante 5 nodos como se muestra en la siguiente figura:



Para este caso, el nodo 0 actuará como un servidor en el programa, mientras que los nodos 1, 2, 3 y 4 actuarán como el cliente.

Las matrices se inicializarán de la siguiente manera:

$$A[i][j] = 3 \cdot i + 2 \cdot j$$

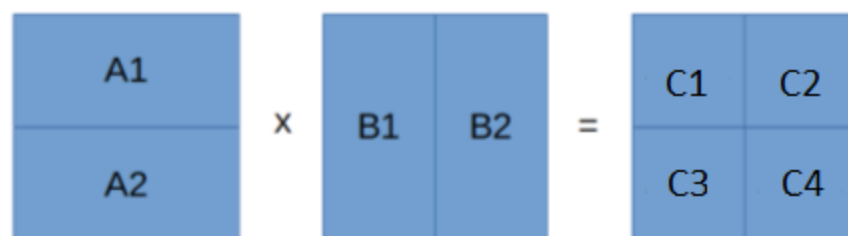
$$B[i][j] = 3 \cdot i - 2 \cdot j$$

La multiplicación de las matrices se realizará de la siguiente manera:

Suponga que divide la matriz A en las matrices A1 y A2. El tamaño de las matrices A1 y A2 es N/2 renglones y N columnas.

La matriz B se divide en las matrices B1 y B2. El tamaño de matrices B1 y B2 es N renglones y N/2 columnas.

Entonces la matriz $C = A \times B$ se compone de las matrices C1, C2, C3 y C4, tal como se muestra en la siguiente figura:



Donde:

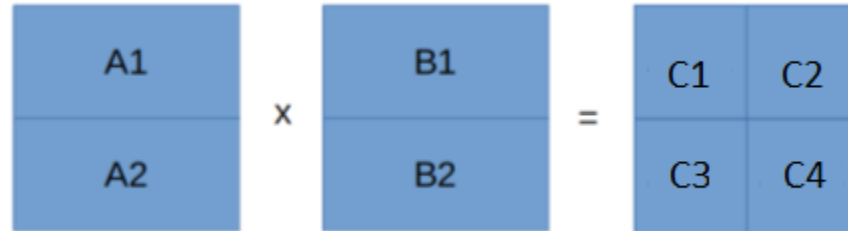
$$C1 = A1 \times B1$$

$$C2 = A1 \times B2$$

$$C3 = A2 \times B1$$

$$C4 = A2 \times B2$$

Debido a que las matrices se guardan en memoria por renglones, se realizará la transposición de la matriz B para que el proceso quede de la siguiente manera:



Compilación del programa:

Se realiza la compilación del programa por medio del comando “javac MultDisMatrices.java” en la terminal de comandos de Windows.

```

C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 3. Multiplicación distribuida de matrices utilizando paso de mensajes\src\MultiplicacionDistribuidaMatrices>javac MultDisMatrices.java
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 3. Multiplicación distribuida de matrices utilizando paso de mensajes\src\MultiplicacionDistribuidaMatrices>

```

Ejecución del programa:

El programa por defecto requerirá de un argumento numérico que indique el nodo que se ejecutará al correr el programa.

En caso de que el programa no reciba ningún argumento, se mostrará el siguiente mensaje:

```

C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 3. Multiplicación distribuida de matrices utilizando paso de mensajes\src\MultiplicacionDistribuidaMatrices>java MultDisMatrices
Uso:
java MultiplicacionDistribuidaMatrices <nodo>

```

En caso de que el programa reciba un argumento no válido (que el argumento no sea un número) este arrojará el siguiente error:

```

C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 3. Multiplicación distribuida de matrices utilizando paso de mensajes\src\MultiplicacionDistribuidaMatrices>java MultDisMatrices a
Exception in thread "main" java.lang.NumberFormatException: For input string: "a"
    at java.base/java.lang.NumberFormatException.forInputString(Unknown Source)
    at java.base/java.lang.Integer.parseInt(Unknown Source)
    at java.base/java.lang.Integer.valueOf(Unknown Source)
    at MultDisMatrices.main(MultDisMatrices.java:148)

```

Finalmente, en el caso de que el programa reciba un argumento válido, esté se ejecutará exitosamente.

Para iniciar la ejecución del servidor, pasamos un cero como argumento al programa, iniciando así la ejecución del nodo 0, el cuál quedará a la espera de que un cliente se conecte.

```
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 3. Multiplicación distribuida de matrices utilizando paso de mensajes\src\MultiplicacionDistribuidaMatrices>java MultDisMatrices 0
```

El servidor se encargará de inicializar las matrices A y B, y realizar la división de las matrices para poder obtener las matrices A1, B1, A2 y B2; y posteriormente enviarlas cada uno de los nodos cliente los cuáles le regresarán al servidor los el resultado de las multiplicaciones C1, C2, C3, y C4. Finalmente, el servidor juntará los 4 resultados obtenidos de cada cliente ara así obtener la matriz $C=A \times B$ y su respectivo checksum.

Ahora, procedemos a ejecutar los nodos que actuarán como clientes, los cuáles serán el nodo 1, 2, 3 y 4, en terminales diferentes, pasándoles como argumento el número correspondiente al nodo a ejecutar.

Una vez que alguno de los nodos cliente haya establecido la conexión con el servidor, recibirá dos de las 4 matrices en las que fueron divididas las matrices A y B. En el caso de el nodo 1 se reciben las matrices A1 y B1, para el nodo 2 las matrices A1 y B2, para el nodo 3 las matrices A2 y B1, y para el nodo 4 las matrices A2 y B2. Finalmente se realizará la multiplicación de ambas matrices en cada nodo y los resultados se le regresarán al servidor.

Se realiza la ejecución de los clientes (nodos 1, 2, 3 y 4).



The image displays four terminal windows, each titled 'Símbolo del sistema', showing the execution of the Java program for different nodes. The command used in each terminal is `java MultDisMatrices` followed by the node number (1, 2, 3, or 4). The paths and the program name are consistent across all terminals.

```
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 3. Multiplicación distribuida de matrices utilizando paso de mensajes\src\MultiplicacionDistribuidaMatrices>java MultDisMatrices 1
```

```
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 3. Multiplicación distribuida de matrices utilizando paso de mensajes\src\MultiplicacionDistribuidaMatrices>java MultDisMatrices 2
```

```
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 3. Multiplicación distribuida de matrices utilizando paso de mensajes\src\MultiplicacionDistribuidaMatrices>java MultDisMatrices 3
```

```
C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 3. Multiplicación distribuida de matrices utilizando paso de mensajes\src\MultiplicacionDistribuidaMatrices>java MultDisMatrices 4
```

Como el servidor muestra datos distintos dependiendo de si $N=4$ o $N=1000$. Se mostrarán a continuación las salidas que el servidor muestra en cada uno de los dos casos:

Cuando $N=4$:

Cuando $N=4$, el servidor imprimirá las matrices A y B, así como también el resultado de la multiplicación de ambas matrices expresado en la Matriz C. Finalmente el programa imprimirá el checksum de la matriz C, la cuál consta de la suma de todos elementos de la matriz.

```

C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 3. Multiplicación distribuida de matrices utilizando paso de mensajes\src\MultiplicacionDistribuidaMatrices>java MultDisMatrices 0
----Matriz A----
|0 2 4 6|
|3 5 7 9|
|6 8 10 12|
|9 11 13 15|
----Matriz B----
|0 -2 -4 -6|
|3 1 -1 -3|
|6 4 2 0|
|9 7 5 3|
----Matriz C----
|84 60 36 12|
|138 90 42 -6|
|192 120 48 -24|
|246 150 54 -42|
Checksum: 1200

```

Cuando $N=1000$:

Cuando $N=1000$, el servidor imprimirá únicamente el checksum de la matriz C. Cabe mencionar que la ejecución de cada cliente es algo lenta debido al tamaño de las matrices.

```

C:\Users\146261\Documents\IPN ESCOM\Desarrollo de Sistemas Distribuidos (Redes 3)\Tarea 3. Multiplicación distribuida de matrices utilizando paso de mensajes\src\MultiplicacionDistribuidaMatrices>java MultDisMatrices 0
Checksum: 158899721392000

```

Para finalizar, cabe mencionar que el programa al tratarse de un sistema distribuido, seguirá funcionando sin importar el orden en el que se ejecuten los nodos siempre y cuando ninguno de estos se repitan.

Conclusión:

El uso de nodos e hilos es bastante útil al momento de programar un algoritmo, pues, además de hacer dicho algoritmo más rápido y eficiente, nos ayuda a incrementar la localidad espacial y temporal de los datos utilizados durante la ejecución de dicho algoritmo.