



Instituto Politécnico Nacional
Escuela Superior de Cómputo
ESCOM



DISEÑO DE SISTEMAS DISTRIBUIDOS

Tarea 3: Multiplicación distribuida de matrices utilizando paso de mensajes

Maya Martínez Alonso Rubén

Fecha de realización

19 de marzo del 2021

Profesor

Pineda Guerrero Carlos

Grupo

4CV1

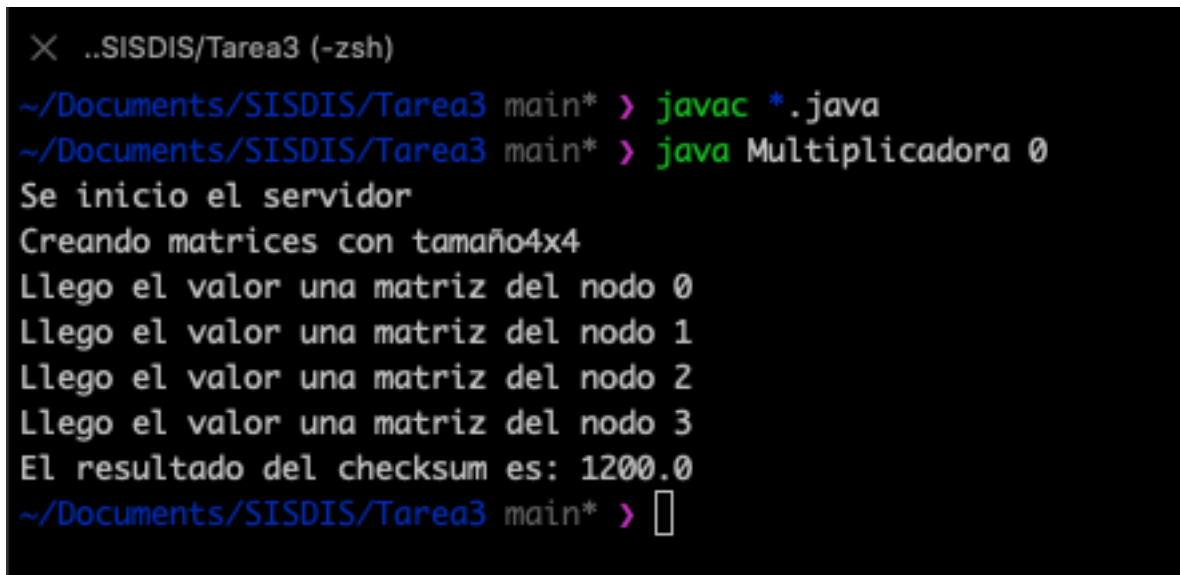
Detalles de la practica.

En esta practica se hará uso del paso de mensajes a través de sockets para hacer distribuido el calculo de la multiplicación de dos matrices utilizando el método de trasponer primero la matriz para así usar filas en vez de columnas y poder aprovechar lo localidad espacial y la cache del procesador

Ejecución.

Con $n = 4$

Como vemos en la imagen 1 se ejecuta el nodo cero que será el servidor, es decir, quien reciba todos los resultados y los una en una sola matriz y calcule el checksum. Podemos ver que se crea una matriz del tamaño 4 para hacer los cálculos. Podemos ver que como resultado obtenemos un checksum de **1200**



```
X ..SISDIS/Tarea3 (-zsh)
~/Documents/SISDIS/Tarea3 main* > javac *.java
~/Documents/SISDIS/Tarea3 main* > java Multiplicadora 0
Se inicio el servidor
Creando matrices con tamaño4x4
Llego el valor una matriz del nodo 0
Llego el valor una matriz del nodo 1
Llego el valor una matriz del nodo 2
Llego el valor una matriz del nodo 3
El resultado del checksum es: 1200.0
~/Documents/SISDIS/Tarea3 main* > █
```

Imagen 1: Compilación y ejecución del nodo 0 con $n = 4$

En la imagen dos observamos la ejecución de los nodos, aunque todos se ejecutaron en “orden” no importa realmente que nodo acabe primero y mande su resultado, el servidor es capaz de ordenar los resultados de manera correcta

```
× ..SISDIS/Tarea3 (-zsh)
~/Documents/SISDIS/Tarea3 main* > java Multiplicadora 1
Se inicio el client 1
Se conecto el client 1
~/Documents/SISDIS/Tarea3 main* >

× ..SISDIS/Tarea3 (-zsh)
~/Documents/SISDIS/Tarea3 main* > java Multiplicadora 2
Se inicio el client 2
Se conecto el client 2
~/Documents/SISDIS/Tarea3 main* >

× ..SISDIS/Tarea3 (-zsh)
~/Documents/SISDIS/Tarea3 main* > java Multiplicadora 3
Se inicio el client 3
Se conecto el client 3
~/Documents/SISDIS/Tarea3 main* >

× ..SISDIS/Tarea3 (-zsh)
~/Documents/SISDIS/Tarea3 main* > java Multiplicadora 4
Se inicio el client 4
Se conecto el client 4
~/Documents/SISDIS/Tarea3 main* >
```

Imagen 2: Ejecución de los nodos 1,2,3 y 4

Con $n = 1000$

La ejecución en este caso es exactamente la misma, en el código se cambia el tamaño, pero funciona exactamente igual tanto el nodo servidor como los clientes, así que solamente se muestra la compilación y ejecución del nodo 0 en la Imagen 3. Vemos que como resultado del checksum obtenemos **$1.74750075 \times 10^{15}$**

```
~/Documents/SISDIS/Tarea3 main* > javac *.java
~/Documents/SISDIS/Tarea3 main* > java Multiplicadora 0
Se inicio el servidor
Creando matrices con tamaño1000x1000
Llego el valor una matriz del nodo 0
Llego el valor una matriz del nodo 1
Llego el valor una matriz del nodo 2
Llego el valor una matriz del nodo 3
El resultado del checksum es: 1.74750075E15
```

Imagen 1: Compilación y ejecución del nodo 0 con $n = 4$

Conclusiones.

Con esta practica pudimos desarrollar el algoritmo visto en clases anteriores y aprovechar la memoria cache de un sistema distribuido. No solo se utiliza mejor la localidad espacial del cache del procesador, sino también aunque sea enorme la matriz la complejidad del algoritmo no crece mucho en cada nodo, ya que el tamaño de la matriz siempre se va a dividir en 4, con esto también podemos hacer operaciones enormes con computadoras que quizá no cuenten con la RAM suficiente para almacenar este tipo de datos.