



**Instituto Politécnico Nacional
Escuela Superior de Cómputo**



Tarea 3: Multiplicación de matrices distribuida utilizando paso de mensajes

**Olay Silis Jose Eduardo
4CV12
Prof. Carlos Pineda Guerrero**

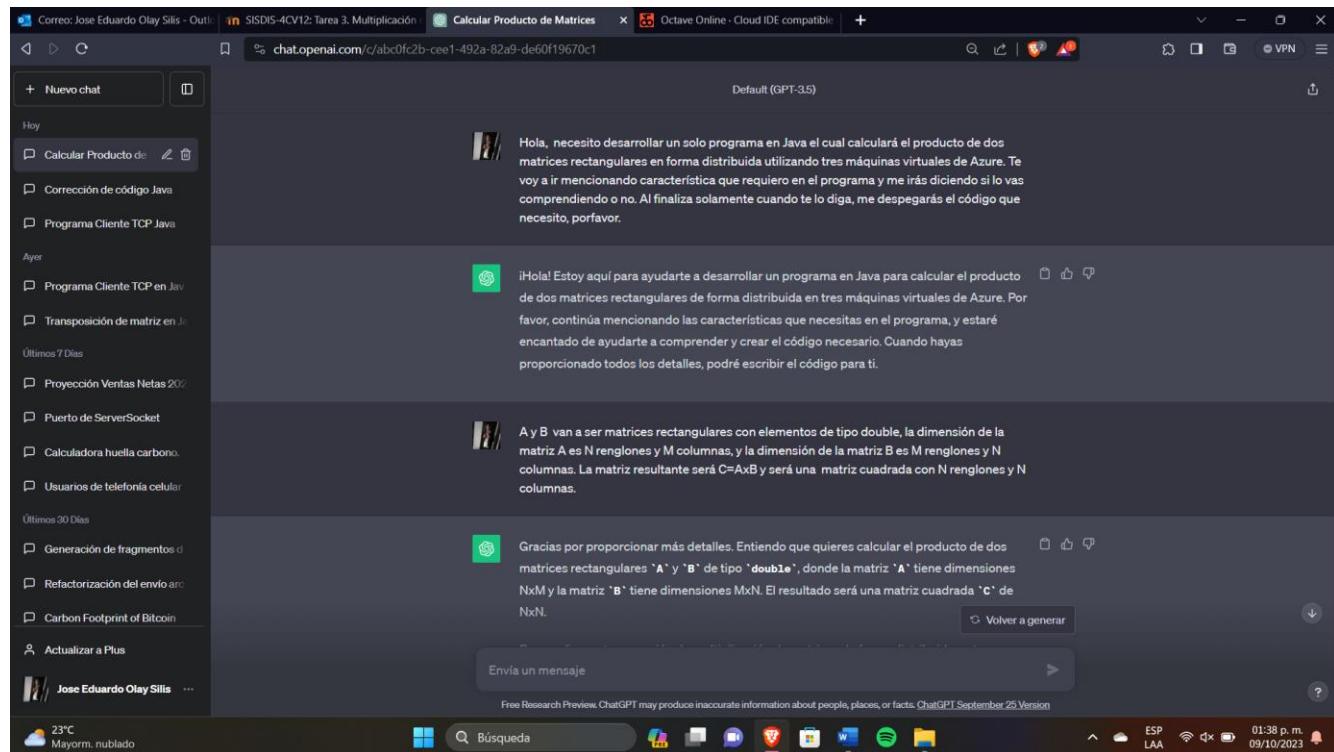
Introducción

Durante el desarrollo de esta práctica se implementó un programa TCP en Java para calcular el producto de matrices rectangulares en un entorno distribuido. La práctica involucra la división de tareas entre tres nodos virtuales en Azure. El Nodo 1 inicializa y coordina, mientras que los Nodos 2 y 3 realizan cálculos intermedios para obtener la matriz C. Utilizando sockets para la comunicación entre los nodos, y se calcula un checksum de C para verificar la integridad de los cálculos.

Desarrollo

Fase 1

En primera instancia comencé a realizar las indicaciones al chat GPT para que realizara la implementación del programa mencionándole en un inicio todas las características que necesitaba para el programa y posteriormente me generara la implementación de dichas instrucciones.



Correo: Jose Eduardo Olay Silis - Outlook | SISDIS-4CV12: Tarea 3. Multiplicación | Calcular Producto de Matrices | Octave Online - Cloud IDE compatible | +

chat.openai.com/c/abc0fc2b-cee1-492a-82a9-de60f19670c1

+ Nuevo chat

Hoy

Calcular Producto de

Corrección de código Java

Programa Cliente TCP Java

Ayer

Programa Cliente TCP en Java

Transposición de matriz en Java

Últimos 7 Días

Proyección Ventas Netas 2023

Puerto de ServerSocket

Calculadora huella carbono.

Usuarios de telefonía celular

Últimos 30 Días

Generación de fragmentos de código

Refactorización del envío de correo electrónico

Carbon Footprint of Bitcoin

Actualizar a Plus

Jose Eduardo Olay Silis ...

23°C Mayorm, nublado

Búsqueda

Gracias por proporcionar más detalles. Entiendo que quieres calcular el producto de dos matrices rectangulares 'A' y 'B' de tipo 'double', donde la matriz 'A' tiene dimensiones NxM y la matriz 'B' tiene dimensiones MxN. El resultado será una matriz cuadrada 'C' de NxN.

Para realizar esta operación de multiplicación de matrices de forma distribuida en tres máquinas virtuales de Azure, necesitarás utilizar algún marco de trabajo de procesamiento distribuido, como Apache Hadoop o Apache Spark, y configurar tus máquinas virtuales para que funcionen como nodos de procesamiento. El proceso implica dividir las matrices en bloques y distribuir los cálculos en los nodos para luego combinar los resultados.

Sin embargo, desarrollar una aplicación completa de este tipo es un proyecto bastante extenso y complejo. Puedo proporcionarte un esqueleto general del código y algunos consejos para que comiences, pero ten en cuenta que necesitarás más detalles y recursos específicos de Azure para implementar completamente esta solución.

Si estás listo para comenzar con el esqueleto del código y los consejos, por favor, avísame.

< 3/3 > Siguiente característica:

El checksum de la matriz C se calculará como la suma de todos elementos de la matriz. El checksum deberá ser de tipo double.

Se deberá inicializar las matrices de la siguiente manera:

A[0][0]
B[0][0]

Volver a generar

Envía un mensaje

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

01:38 p. m. 09/10/2023

Correo: Jose Eduardo Olay Silis - Outlook | SISDIS-4CV12: Tarea 3. Multiplicación | Calcular Producto de Matrices | Octave Online - Cloud IDE compatible | +

chat.openai.com/c/abc0fc2b-cee1-492a-82a9-de60f19670c1

+ Nuevo chat

Hoy

Calcular Producto de

Corrección de código Java

Programa Cliente TCP Java

Ayer

Programa Cliente TCP en Java

Transposición de matriz en Java

Últimos 7 Días

Proyección Ventas Netas 2023

Puerto de ServerSocket

Calculadora huella carbono.

Usuarios de telefonía celular

Últimos 30 Días

Generación de fragmentos de código

Refactorización del envío de correo electrónico

Carbon Footprint of Bitcoin

Actualizar a Plus

Jose Eduardo Olay Silis ...

23°C Mayorm, nublado

Búsqueda

Donde A[i][j] y B[i][j] son los elementos Ai,j y Bi,j respectivamente.

Aún no generes código solo ve agregando las características que te voy mencionando a tu planeación para el código ok? yo te avisaré cuando proporciones el código

Entendido. Añadiré la característica de calcular el checksum de la matriz C como la suma de todos los elementos de la matriz y también mencionaré la inicialización de las matrices A y B como se indica. Cuando estés listo para que proporcione el código, por favor avísame. Continuaré esperando tus instrucciones.

El programa deberá ejecutar en tres máquinas virtuales con Ubuntu (nodos 1, 2 y 3). De manera que cada nodo ejecute una instancia del programa de la siguiente manera:

El nodo 1 realizará lo siguiente:

Inicializará las matrices A y B.
Obtendrá la transpuesta de la matriz B.
Enviarán las matrices A y BT al nodo 2 (solo enviarán las matrices necesarias para realizar los productos C19 al C36).
Enviarán las matrices A y BT al nodo 3 (solo enviarán las matrices necesarias para realizar los productos C19 al C36).
Recibirá del nodo 2 las matrices C1 a la C18.
Recibirá del nodo 3 las matrices C19 a la C36.
Obtendrá la matriz C a partir de las matrices C1 a la C36.
Calculará y desplegará el checksum de la matriz C.
Desplegará las matrices A, B y C para el caso N=6, M=5.

Volver a generar

Envía un mensaje

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

01:38 p. m. 09/10/2023

El nodo 2 realizará lo siguiente:

Recibirá las matrices A_i y B_{Tj} del nodo 1 (solo recibirá las matrices necesarias para realizar los productos $C1$ al $C18$).
Calculará los productos $C1$ al $C18$.
Enviarán las matrices $C1$ a la $C18$ al nodo 1.

El nodo 3 realizará lo siguiente:

Recibirá las matrices A_i y B_{Tj} del nodo 1 (solo recibirá las matrices necesarias para realizar los productos $C19$ al $C36$).
Calculará los productos $C19$ al $C36$.
Enviarán las matrices $C19$ a la $C36$ al nodo 1.

Se deberá pasar el número de nodo como parámetro al programa.

Gracias por proporcionar detalles adicionales sobre la ejecución distribuida del programa en tres máquinas virtuales. Entiendo la estructura de la ejecución y cómo se comunican los nodos. A continuación, mencionaré esta estructura en la planificación del código:

Planificación del Programa:

- Inicialización de Matrices (Nodo 1):**
 - Inicializar las matrices ' A ' y ' B ' con valores aleatorios de acuerdo a las dimensiones $N \times M$ y $M \times N$ respectivamente.
- Obtención de la Transpuesta de B (Nodo 1):**
 - Calcular la transpuesta de la matriz ' B ', es decir, obtener ' B^T '.
- Envío de Matrices a Nodos 2 y 3 (Nodo 1):**
 - Volver a generar

Envía un mensaje ➤

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

23°C Mayorm. nublado

Búsqueda

Jose Eduardo Olay Silis ...

Correo: Jose Eduardo Olay Silis - Outlook SISDIS-4CV12: Tarea 3. Multiplicación Calcular Producto de Matrices Octave Online - Cloud IDE compatible

Hoy
Calcular Producto de Matrices
Corrección de código Java
Programa Cliente TCP Java
Ayer
Programa Cliente TCP en Java
Transposición de matriz en Java
Últimos 7 Días
Proyección Ventas Netas 2023
Puerto de ServerSocket
Calculadora huella carbono.
Usuarios de telefonía celular
Últimos 30 Días
Generación de fragmentos d
Refactorización del envío arc
Carbon Footprint of Bitcoin
Actualizar a Plus
Jose Eduardo Olay Silis ...

23°C Mayorm. nublado

Búsqueda

ESP LAA 01:39 p. m. 09/10/2023

Default (GPT-3.5)

Gracias por proporcionar detalles adicionales sobre la ejecución distribuida del programa en tres máquinas virtuales. Entiendo la estructura de la ejecución y cómo se comunican los nodos. A continuación, mencionaré esta estructura en la planificación del código:

Planificación del Programa:

- Inicialización de Matrices (Nodo 1):**
 - Inicializar las matrices ' A ' y ' B ' con valores aleatorios de acuerdo a las dimensiones $N \times M$ y $M \times N$ respectivamente.
- Obtención de la Transpuesta de B (Nodo 1):**
 - Calcular la transpuesta de la matriz ' B ', es decir, obtener ' B^T '.
- Envío de Matrices a Nodos 2 y 3 (Nodo 1):**
 - Enviar las matrices ' A_1 ' (porciones de ' A ') y ' B_{Tj} ' (porciones de ' B^T ') al nodo 2 y nodo 3, respectivamente, para realizar los productos parciales ' $C1$ ' al ' $C36$ '.
- Cálculo de Productos Parciales (Nodo 2 y Nodo 3):**
 - En cada nodo (2 y 3), recibir las matrices ' A_1 ' y ' B_{Tj} ' desde el nodo 1.
 - Calcular los productos parciales ' $C1$ ' al ' $C18$ ' en el nodo 2 y ' $C19$ ' al ' $C36$ ' en el nodo 3.
 - Enviar los productos parciales de regreso al nodo 1.
- Recepción de Productos Parciales (Nodo 1):**
 - Recibir las matrices ' $C1$ ' al ' $C18$ ' del nodo 2 y ' $C19$ ' al ' $C36$ ' del nodo 3.
- Combinación de Matrices para Obtener C (Nodo 1):**
 - Combinar las matrices ' $C1$ ' al ' $C36$ ' para obtener la matriz ' C '.
- Cálculo del Checksum (Nodo 1):**
 - Calcular el checksum de la matriz ' C ' como la suma de todos sus elementos.
- Despliegue de Resultados (Nodo 1):**
 - Mostrar en pantalla las matrices ' A ', ' B ' y ' C ' para el caso $N=6$ y $M=5$.

Volver a generar

Envía un mensaje ➤

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

23°C Mayorm. nublado

Búsqueda

Jose Eduardo Olay Silis ...

Correo: Jose Eduardo Olay Silis - Outlook SISDIS-4CV12: Tarea 3. Multiplicación Calcular Producto de Matrices Octave Online - Cloud IDE compatible

Hoy
Calculadora huella carbono.
Usuarios de telefonía celular
Últimos 30 Días
Generación de fragmentos d
Refactorización del envío arc
Carbon Footprint of Bitcoin
Actualizar a Plus
Jose Eduardo Olay Silis ...

23°C Mayorm. nublado

Búsqueda

ESP LAA 01:40 p. m. 09/10/2023

Perfecto, proporcioname el código de favor

Desarrollar un programa de este tipo es un proyecto complejo que requiere configuraciones y dependencias específicas, además de la implementación de la comunicación entre nodos. Aquí te proporcionaré un esqueleto de código en Java que puedes utilizar como punto de partida. Debes tener en cuenta que necesitarás configurar y ejecutar este código en las máquinas virtuales de Azure, y también se requieren bibliotecas adicionales para la comunicación entre nodos. Te recomiendo consultar la documentación de Azure y las bibliotecas de comunicación distribuida, como Apache Hadoop o Apache Spark, para obtener más detalles y ayuda en la implementación real. Aquí tienes un esqueleto básico en Java:

```
java
import java.util.Random;

public class MatrixMultiplication {
    public static void main(String[] args) {
        if (args.length != 1) {
            System.err.println("Debe proporcionar el número de nodo (1, 2 o 3)");
            System.exit(1);
        }

        int nodeNumber = Integer.parseInt(args[0]);
        if (nodeNumber == 1) {
            // Nodo 1
        }
    }
}
```

Volver a generar

Envía un mensaje

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

23°C Mayorm. nublado

```
int nodeNumber = Integer.parseInt(args[0]);

if (nodeNumber == 1) {
    // Nodo 1
    // Inicialización de matrices A y B
    double[][] A = initializeMatrix();
    double[][] B = initializeMatrix();

    // Cálculo de la transpuesta de B (BT)

    // Dividir A y BT en porciones y enviar a nodos 2 y 3

    // Recibir C1 al C18 de nodo 2 y C19 al C36 de nodo 3

    // Combinar matrices C1 al C36 para obtener C

    // Calcular y mostrar el checksum de C

    // Mostrar las matrices A, B y C
} else if (nodeNumber == 2) {
    // Nodo 2
    // Recibir porciones de A y BT de nodo 1

    // Calcular productos C1 al C18

    // Enviar C1 al C18 a nodo 1
} else if (nodeNumber == 3) {
    // Nodo 3
    // Recibir porciones de A y BT de nodo 1

    // Calcular productos C19 al C36
}
```

Volver a generar

Envía un mensaje

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

23°C Mayorm. nublado

The screenshot shows a Microsoft Edge browser window with the following details:

- Address Bar:** chat.openai.com/c/abc0fc2b-cee1-492a-82a9-de60f19670c1
- Left Sidebar (Messages):** Shows a list of messages from "Hoy" and "Ayer".
- Main Content Area:** Displays a Java code snippet for matrix multiplication. The code includes logic for sending matrix portions to node 1 or node 3, and handling invalid node numbers.
- Text Below Code:** A note explaining the skeleton code provides a basic structure but lacks real logic for matrix calculations and communication between nodes.
- Bottom Buttons:** Includes "Volver a generar" (Generate again) and a message input field.
- System Tray:** Shows weather (23°C, Mayorm, nublado), battery level (ESP LAA), and system status (01:41 p.m., 09/10/2023).

De primera instancia el chat proporciono un esqueleto de programa de manera local por lo que le pedí que hiciera el mismo esqueleto del código, pero utilizando sockets

The screenshot shows a Microsoft Edge browser window with the following details:

- Address Bar:** chat.openai.com/c/abc0fc2b-cee1-492a-82a9-de60f19670c1
- Main Content Area:** Shows a message from the AI stating it needs to implement socket communication for receiving information from nodes.
- Code Snippet:** A Java code snippet for matrix multiplication using sockets. It includes imports for java.io, java.net, and java.util.Random, and defines a MatrixMultiplication class with a main method for node 1.
- Text Below Code:** A note indicating the code is updated to include socket communication.
- Bottom Buttons:** Includes "Copy code", "Volver a generar" (Generate again), and a message input field.
- System Tray:** Shows weather (23°C, Mayorm, nublado), battery level (ESP LAA), and system status (01:42 p.m., 09/10/2023).

The screenshot shows a web-based chat interface with a sidebar on the left containing a list of recent conversations and a main panel on the right displaying Java code. The sidebar includes sections for 'Hoy', 'Ayer', and 'Últimos 7 Días'. The main panel displays the following Java code:

```
ServerSocket serverSocket = new ServerSocket(12345);

// Esperar conexiones entrantes de nodo 2 y nodo 3
Socket socket2 = serverSocket.accept();
Socket socket3 = serverSocket.accept();

// Inicialización de matrices A y B
double[][] A = initializeMatrix();
double[][] B = initializeMatrix();

// Cálculo de la transpuesta de B (BT)

// Dividir A y BT en porciones y enviar a nodos 2 y 3

// Recibir C1 al C18 de nodo 2 y C19 al C36 de nodo 3

// Combinar matrices C1 al C36 para obtener C

// Calcular y mostrar el checksum de C

// Mostrar las matrices A, B y C

// Cerrar sockets y limpiar recursos
socket2.close();
socket3.close();
serverSocket.close();
} catch (IOException e) {
    e.printStackTrace();
}
} else if (nodeNumber == 2) {
    // Nodo 2
```

The screenshot shows a web-based chat interface with a sidebar on the left containing a list of recent conversations and a main panel on the right displaying Java code. The sidebar includes sections for 'Hoy', 'Ayer', and 'Últimos 7 Días'. The main panel displays the following Java code:

```
// NODO 2
try {
    Socket socket1 = new Socket("NODO1_IP", 12345); // Reemplaza con la IP del nodo 1

    // Recibir porciones de A y BT de nodo 1

    // Calcular productos C1 al C18

    // Enviar C1 al C18 a nodo 1

    // Cerrar socket y limpiar recursos
    socket1.close();
} catch (IOException e) {
    e.printStackTrace();
}
} else if (nodeNumber == 3) {
    // Nodo 3
    try {
        Socket socket1 = new Socket("NODO1_IP", 12345); // Reemplaza con la IP del nodo 1

        // Recibir porciones de A y BT de nodo 1

        // Calcular productos C19 al C36

        // Enviar C19 al C36 a nodo 1

        // Cerrar socket y limpiar recursos
        socket1.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

The screenshot shows a web browser window with multiple tabs open. The active tab is a chat interface on chat.openai.com. The sidebar on the left lists various recent conversations. The main area displays a Java code snippet for matrix multiplication:

```
// Enviar C19 al C36 a nodo 1  
  
// Cerrar socket y limpiar recursos  
socket1.close();  
} catch (IOException e) {  
    e.printStackTrace();  
}  
} else {  
    System.err.println("Número de nodo no válido. Debe ser 1, 2 o 3.");  
    System.exit(1);  
}  
  
private static double[][] initializeMatrix() {  
    // Implementa la inicialización de una matriz NxM con valores aleatorios  
    // Retorna la matriz inicializada  
}
```

Below the code, a message from ChatGPT is displayed:

En este código, he agregado la creación de sockets y la recepción/envío de datos entre los nodos. Debes reemplazar `"**NODE1_IP**"` con la dirección IP del Nodo 1 en los nodos 2 y 3. Asegúrate de tener las conexiones de red configuradas adecuadamente en tus máquinas virtuales de Azure y que los puertos estén abiertos para la comunicación. Además, necesitarás implementar la lógica real para dividir las matrices y realizar los cálculos en cada nodo según tu diseño específico.

At the bottom right, there is a "Volver a generar" (Generate again) button.

Una vez dado el esqueleto del código comencé a pedirle que me fuera dando cada una de las partes faltantes de el esqueleto que me proporcionó de manera general.

The screenshot shows the same web browser window and chat interface as the previous one. The sidebar and Java code snippet remain the same. The message from ChatGPT has been updated:

ok completa el código del Nodo 1 para calcular la transpuesta de B

Claro, aquí tienes el código completo para calcular la transpuesta de la matriz "B" en el Nodo 1:

```
java  
  
import java.io.*;  
import java.net.*;  
import java.util.Random;  
  
public class MatrixMultiplication {  
    public static void main(String[] args) {  
        // ... (código anterior)  
  
        if (nodeNumber == 1) {  
            try {  
                ServerSocket serverSocket = new ServerSocket(12345);  
  
                // Esperar conexiones entrantes de nodo 2 y nodo 3  
                Socket socket2 = serverSocket.accept();  
                Socket socket3 = serverSocket.accept();  
  
                // Inicialización de matrices A y B  
                double[][] A = initializeMatrix();  
                double[][] B = initializeMatrix();  
            } catch (IOException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

At the bottom right, there is a "Volver a generar" (Generate again) button.

Correo: Jose Eduardo Olay Silis - Outl... SISDIS-4CV12: Tarea 3. Multiplicación Calcular Producto de Matrices Octave Online - Cloud IDE compatible

+ Nuevo chat

Hoy

- Calcular Producto de ...
- Corrección de código Java
- Programa Cliente TCP Java

Ayer

- Programa Cliente TCP en Jav...
- Transposición de matriz en J...

Últimos 7 Días

- Proyección Ventas Netas 202...
- Puerto de ServerSocket
- Calculadora huella carbono.
- Usuarios de telefonía celular

Últimos 30 Días

- Generación de fragmentos d...
- Refactorización del envío arc...
- Carbon Footprint of Bitcoin
- Actualizar a Plus

Jose Eduardo Olay Silis ...

Envía un mensaje Volver a generar

```

        Socket socket3 = serverSocket.accept();

        // Inicialización de matrices A y B
        double[][] A = initializeMatrix();
        double[][] B = initializeMatrix();

        // Cálculo de la transpuesta de B (BT)
        double[][] BT = new double[B[0].length][B.length];
        for (int i = 0; i < B.length; i++) {
            for (int j = 0; j < B[0].length; j++) {
                BT[j][i] = B[i][j];
            }
        }

        // Dividir A y BT en porciones y enviar a nodos 2 y 3
        int numRowsPerNode = A.length / 2; // Dividir A en 2 partes
        int numColsPerNode = BT[0].length / 2; // Dividir BT en 2 pa...

        ObjectOutputStream out2 = new ObjectOutputStream(socket2.get...
        ObjectOutputStream out3 = new ObjectOutputStream(socket3.get...

        // Enviar la parte correspondiente de A a nodo 2
        for (int i = 0; i < numRowsPerNode; i++) {
            out2.writeObject(A[i]);
        }

        // Enviar la parte correspondiente de BT a nodo 3
        for (int j = 0; j < numColsPerNode; j++) {
            double[] col = new double[BT.length];
            for (int i = 0; i < BT.length; i++) {
                col[i] = BT[i][j];
            }
        }
    }
}

```

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

23°C Mayorm, nublado

Búsqueda

ESP LAA 01:46 p. m. 09/10/2023

Correo: Jose Eduardo Olay Silis - Outl... SISDIS-4CV12: Tarea 3. Multiplicación Calcular Producto de Matrices Octave Online - Cloud IDE compatible

+ Nuevo chat

Hoy

- Calcular Producto de ...
- Corrección de código Java
- Programa Cliente TCP Java

Ayer

- Programa Cliente TCP en Jav...
- Transposición de matriz en J...

Últimos 7 Días

- Proyección Ventas Netas 202...
- Puerto de ServerSocket
- Calculadora huella carbono.
- Usuarios de telefonía celular

Últimos 30 Días

- Generación de fragmentos d...
- Refactorización del envío arc...
- Carbon Footprint of Bitcoin
- Actualizar a Plus

Jose Eduardo Olay Silis ...

Proporcioname como sería la implementación para dividir A en en 6 partes de igual tamaño [N/3][M] y que se almacenen en las variables A1, A2, A3, A4, A5 y A6

Para dividir la matriz A en 6 partes de igual tamaño [N/6][M] y almacenarlas en las variables A1, A2, A3, A4, A5 y A6, puedes hacerlo de la siguiente manera:

```

java
Copy code

int numRowsPerNode = N / 6; // Número de filas por cada una de las 6 partes

double[][] A1 = new double[numRowsPerNode][M];
double[][] A2 = new double[numRowsPerNode][M];
double[][] A3 = new double[numRowsPerNode][M];
double[][] A4 = new double[numRowsPerNode][M];
double[][] A5 = new double[numRowsPerNode][M];
double[][] A6 = new double[numRowsPerNode][M];

// Dividir la matriz A en las 6 partes
System.arraycopy(A, 0, A1, 0, numRowsPerNode);
System.arraycopy(A, numRowsPerNode, A2, 0, numRowsPerNode);
System.arraycopy(A, 2 * numRowsPerNode, A3, 0, numRowsPerNode);
System.arraycopy(A, 3 * numRowsPerNode, A4, 0, numRowsPerNode);
System.arraycopy(A, 4 * numRowsPerNode, A5, 0, numRowsPerNode);
System.arraycopy(A, 5 * numRowsPerNode, A6, 0, numRowsPerNode);

```

Asegúrate de agregar este código en el Nodo 1 después de la inicialización de la ... Volver a generar

antes de enviar las partes correspondientes de A a los Nodos 2 y 3. Esto dividirá la matriz A

Envía un mensaje Volver a generar

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

23°C Mayorm, nublado

Búsqueda

ESP LAA 01:48 p. m. 09/10/2023

Correo: Jose Eduardo Olay Silis - Outlook | SISDIS-4CV12: Tarea 3. Multiplicación | Calcular Producto de Matrices | Octave Online - Cloud IDE compatible | +

chat.openai.com/c/abc0fc2b-cee1-492a-82a9-de60f19670c1

+ Nuevo chat

Hoy

- Calcular Producto de
- Corrección de código Java
- Programa Cliente TCP Java

Ayer

- Programa Cliente TCP en Jav
- Transposición de matriz en Jav

Últimos 7 Días

- Proyección Ventas Netas 202
- Puerto de ServerSocket
- Calculadora huella carbono.
- Usuarios de telefonía celular

Últimos 30 Días

- Generación de fragmentos d
- Refactorización del envío arc
- Carbon Footprint of Bitcoin

Actualizar a Plus

Jose Eduardo Olay Silis ...

Proporcioname como sería la implementación para dividir BT en en 6 partes de igual tamaño [N/6][M] y que se almacenen en las variables BT1,BT2,BT3,BT4,BT5 y BT6

Para dividir la matriz BT en 6 partes de igual tamaño [N/6][M] y almacenarlas en las variables BT1, BT2, BT3, BT4, BT5 y BT6, puedes hacerlo de manera similar a la división de la matriz A. Aquí tienes la implementación:

```
java
```

```
int numRowsPerNode = N / 6; // Número de filas por cada una de las 6 partes

double[][] BT1 = new double[numRowsPerNode][M];
double[][] BT2 = new double[numRowsPerNode][M];
double[][] BT3 = new double[numRowsPerNode][M];
double[][] BT4 = new double[numRowsPerNode][M];
double[][] BT5 = new double[numRowsPerNode][M];
double[][] BT6 = new double[numRowsPerNode][M];

// Dividir la matriz BT en las 6 partes
System.arraycopy(BT, 0, BT1, 0, numRowsPerNode);
System.arraycopy(BT, 2 * numRowsPerNode, BT2, 0, numRowsPerNode);
System.arraycopy(BT, 3 * numRowsPerNode, BT3, 0, numRowsPerNode);
System.arraycopy(BT, 4 * numRowsPerNode, BT4, 0, numRowsPerNode);
System.arraycopy(BT, 5 * numRowsPerNode, BT5, 0, numRowsPerNode);
System.arraycopy(BT, 6 * numRowsPerNode, BT6, 0, numRowsPerNode);
```

Volver a generar

Envía un mensaje

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

23°C Mayorm. nublado

Búsqueda

ESP LAA 01:49 p. m. 09/10/2023

Correo: Jose Eduardo Olay Silis - Outlook | SISDIS-4CV12: Tarea 3. Multiplicación | Calcular Producto de Matrices | Octave Online - Cloud IDE compatible | +

chat.openai.com/c/abc0fc2b-cee1-492a-82a9-de60f19670c1

+ Nuevo chat

Hoy

- Calcular Producto de
- Corrección de código Java
- Programa Cliente TCP Java

Ayer

- Programa Cliente TCP en Jav
- Transposición de matriz en Jav

Últimos 7 Días

- Proyección Ventas Netas 202
- Puerto de ServerSocket
- Calculadora huella carbono.
- Usuarios de telefonía celular

Últimos 30 Días

- Generación de fragmentos d
- Refactorización del envío arc
- Carbon Footprint of Bitcoin

Actualizar a Plus

Jose Eduardo Olay Silis ...

Ahora proporcioname el código para enviar A1,A2,A3,BT1,BT2,BT3,BT4,BT5 y BT6 al nodo 2

Aquí tienes el código para enviar las matrices A1, A2, A3, BT1, BT2, BT3, BT4, BT5 y BT6 al Nodo 2:

```
java
```

```
ObjectOutputStream out2 = new ObjectOutputStream(socket2.getOutputStream());

// Enviar las partes correspondientes de A y BT a nodo 2
out2.writeObject(A1);
out2.writeObject(A2);
out2.writeObject(A3);
out2.writeObject(BT1);
out2.writeObject(BT2);
out2.writeObject(BT3);
out2.writeObject(BT4);
out2.writeObject(BT5);
out2.writeObject(BT6);
```

Agrega este código en el Nodo 1 después de haber dividido las matrices A y BT en las partes correspondientes (A1, A2, A3, BT1, BT2, BT3, BT4, BT5 y BT6). Esto enviará estas partes al Nodo 2 para su procesamiento.

Volver a generar

Envía un mensaje

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

23°C Mayorm. nublado

Búsqueda

ESP LAA 01:49 p. m. 09/10/2023

Correo: Jose Eduardo Olay Silis - Outlook SISDIS-4CV12: Tarea 3. Multiplicación Calcular Producto de Matrices Octave Online - Cloud IDE compatible +

chat.openai.com/c/abc0fc2b-cee1-492a-82a9-de60f19670c1 Default (GPT-3.5)

+ Nuevo chat

Hoy

Calcular Producto de Matrices

Corrección de código Java

Programa Cliente TCP Java

Ayer

Programa Cliente TCP en Java

Transposición de matriz en Java

Últimos 7 Días

Proyección Ventas Netas 2022

Puerto de ServerSocket

Calculadora huella carbono.

Usuarios de telefonía celular

Últimos 30 Días

Generación de fragmentos d

Refactorización del envío arc

Carbon Footprint of Bitcoin

Actualizar a Plus

Jose Eduardo Olay Silis ...

23°C Mayorm. nublado

Una vez que enviamos las partes correspondientes de A y BT a nodo 2

```
out2.writeObject(A1);
out2.writeObject(A2);
out2.writeObject(A3);
out2.writeObject(BT1);
out2.writeObject(BT2);
out2.writeObject(BT3);
out2.writeObject(BT4);
out2.writeObject(BT5);
out2.writeObject(BT6);
```

proporciona el código del nodo 2 que reciba dichas matrices

Claro, aquí tienes el código del Nodo 2 que recibe las matrices enviadas por el Nodo 1:

java

```
public class Nodo2 {
    public static void main(String[] args) {
        try {
            Socket socket1 = new Socket("localhost", 8080); // Reemplaza "localhost" por la dirección IP o nombre del nodo 1
            ObjectInputStream in1 = new ObjectInputStream(socket1.getInputStream());

            // Crear arreglos para almacenar las matrices recibidas
            double[][] A1 = (double[][]) in1.readObject();
            double[][] A2 = (double[][]) in1.readObject();
            double[][] A3 = (double[][]) in1.readObject();

            // Realizar los cálculos con las matrices recibidas (C1 al C18)
            double[][] C1 = multiplyMatrices(A1, BT1);
            double[][] C2 = multiplyMatrices(A1, BT2);
            double[][] C3 = multiplyMatrices(A1, BT3);
            double[][] C4 = multiplyMatrices(A1, BT4);
            double[][] C5 = multiplyMatrices(A1, BT5);
            double[][] C6 = multiplyMatrices(A1, BT6);
            double[][] C7 = multiplyMatrices(A2, BT1);
            double[][] C8 = multiplyMatrices(A2, BT2);
            double[][] C9 = multiplyMatrices(A2, BT3);
            double[][] C10 = multiplyMatrices(A2, BT4);
            double[][] C11 = multiplyMatrices(A2, BT5);
            double[][] C12 = multiplyMatrices(A2, BT6);
            double[][] C13 = multiplyMatrices(A3, BT1);
            double[][] C14 = multiplyMatrices(A3, BT2);
            double[][] C15 = multiplyMatrices(A3, BT3);
            double[][] C16 = multiplyMatrices(A3, BT4);
            double[][] C17 = multiplyMatrices(A3, BT5);
            double[][] C18 = multiplyMatrices(A3, BT6);

            // Enviar las matrices calculadas (C1 al C18) de vuelta al Nodo 1
            ObjectOutputStream out1 = new ObjectOutputStream(socket1.getOutputStream());
            out1.writeObject(C1);
            out1.writeObject(C2);
            out1.writeObject(C3);
            out1.writeObject(C4);
            out1.writeObject(C5);
            out1.writeObject(C6);
            out1.writeObject(C7);
            out1.writeObject(C8);
            out1.writeObject(C9);
            out1.writeObject(C10);
            out1.writeObject(C11);
            out1.writeObject(C12);
            out1.writeObject(C13);
            out1.writeObject(C14);
            out1.writeObject(C15);
            out1.writeObject(C16);
            out1.writeObject(C17);
            out1.writeObject(C18);
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

Volver a generar

Envía un mensaje

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

23°C Mayorm. nublado

Búsqueda

ESP LAA 01:49 p. m. 09/10/2023

Correo: Jose Eduardo Olay Silis - Outlook SISDIS-4CV12: Tarea 3. Multiplicación Calcular Producto de Matrices Octave Online - Cloud IDE compatible +

chat.openai.com/c/abc0fc2b-cee1-492a-82a9-de60f19670c1 Default (GPT-3.5)

+ Nuevo chat

Hoy

Calcular Producto de Matrices

Corrección de código Java

Programa Cliente TCP Java

Ayer

Programa Cliente TCP en Java

Transposición de matriz en Java

Últimos 7 Días

Proyección Ventas Netas 2022

Puerto de ServerSocket

Calculadora huella carbono.

Usuarios de telefonía celular

Últimos 30 Días

Generación de fragmentos d

Refactorización del envío arc

Carbon Footprint of Bitcoin

Actualizar a Plus

Jose Eduardo Olay Silis ...

23°C Mayorm. nublado

double[][] A3 = (double[][]) in1.readObject();
double[][] BT1 = (double[][]) in1.readObject();
double[][] BT2 = (double[][]) in1.readObject();
double[][] BT3 = (double[][]) in1.readObject();
double[][] BT4 = (double[][]) in1.readObject();
double[][] BT5 = (double[][]) in1.readObject();
double[][] BT6 = (double[][]) in1.readObject();

// Realizar los cálculos con las matrices recibidas (C1 al C18)
double[][] C1 = multiplyMatrices(A1, BT1);
double[][] C2 = multiplyMatrices(A1, BT2);
double[][] C3 = multiplyMatrices(A1, BT3);
double[][] C4 = multiplyMatrices(A1, BT4);
double[][] C5 = multiplyMatrices(A1, BT5);
double[][] C6 = multiplyMatrices(A1, BT6);
double[][] C7 = multiplyMatrices(A2, BT1);
double[][] C8 = multiplyMatrices(A2, BT2);
double[][] C9 = multiplyMatrices(A2, BT3);
double[][] C10 = multiplyMatrices(A2, BT4);
double[][] C11 = multiplyMatrices(A2, BT5);
double[][] C12 = multiplyMatrices(A2, BT6);
double[][] C13 = multiplyMatrices(A3, BT1);
double[][] C14 = multiplyMatrices(A3, BT2);
double[][] C15 = multiplyMatrices(A3, BT3);
double[][] C16 = multiplyMatrices(A3, BT4);
double[][] C17 = multiplyMatrices(A3, BT5);
double[][] C18 = multiplyMatrices(A3, BT6);

// Enviar las matrices calculadas (C1 al C18) de vuelta al Nodo 1
ObjectOutputStream out1 = new ObjectOutputStream(socket1.getOutputStream());
out1.writeObject(C1);
out1.writeObject(C2);
out1.writeObject(C3);
out1.writeObject(C4);
out1.writeObject(C5);
out1.writeObject(C6);
out1.writeObject(C7);
out1.writeObject(C8);
out1.writeObject(C9);
out1.writeObject(C10);
out1.writeObject(C11);
out1.writeObject(C12);
out1.writeObject(C13);
out1.writeObject(C14);
out1.writeObject(C15);
out1.writeObject(C16);
out1.writeObject(C17);
out1.writeObject(C18);

Volver a generar

Envía un mensaje

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

23°C Mayorm. nublado

Búsqueda

ESP LAA 01:50 p. m. 09/10/2023

Correo: Jose Eduardo Olay Silis - Outl SISDIS-4CV12: Tarea 3. Multiplicación Calcular Producto de Matrices Octave Online - Cloud IDE compatible chat.openai.com/c/abc0fc2b-cee1-492a-82a9-de60f19670c1

+ Nuevo chat

Hoy

Calcular Producto de

Corrección de código Java

Programa Cliente TCP Java

Ayer

Programa Cliente TCP en Jav

Transposición de matriz en J

Últimos 7 Días

Proyección Ventas Netas 202

Puerto de ServerSocket

Calculadora huella carbono.

Usuarios de telefonía celular

Últimos 30 Días

Generación de fragmentos d

Refactorización del envío arc

Carbon Footprint of Bitcoin

Actualizar a Plus

Jose Eduardo Olay Silis ...

23°C Mayorm. nublado

Búsqueda

```
double[][] C15 = multiplyMatrices(A3, BT3);
double[][] C16 = multiplyMatrices(A3, BT4);
double[][] C17 = multiplyMatrices(A3, BT5);
double[][] C18 = multiplyMatrices(A3, BT6);

// Enviar las matrices calculadas (C1 al C18) de vuelta al Nodo
ObjectOutputStream out1 = new ObjectOutputStream(socket1.getOutputStream());
out1.writeObject(C1);
out1.writeObject(C2);
out1.writeObject(C3);
out1.writeObject(C4);
out1.writeObject(C5);
out1.writeObject(C6);
out1.writeObject(C7);
out1.writeObject(C8);
out1.writeObject(C9);
out1.writeObject(C10);
out1.writeObject(C11);
out1.writeObject(C12);
out1.writeObject(C13);
out1.writeObject(C14);
out1.writeObject(C15);
out1.writeObject(C16);
out1.writeObject(C17);
out1.writeObject(C18);

// Cerrar el socket y liberar recursos
socket1.close();
} catch (IOException | ClassNotFoundException e) {
    e.printStackTrace();
}
```

Volver a generar

Envía un mensaje

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

01:50 p. m. 09/10/2023

Correo: Jose Eduardo Olay Silis - Outl SISDIS-4CV12: Tarea 3. Multiplicación Calcular Producto de Matrices Octave Online - Cloud IDE compatible chat.openai.com/c/abc0fc2b-cee1-492a-82a9-de60f19670c1

+ Nuevo chat

Hoy

Calcular Producto de

Corrección de código Java

Programa Cliente TCP Java

Ayer

Programa Cliente TCP en Jav

Transposición de matriz en J

Últimos 7 Días

Proyección Ventas Netas 202

Puerto de ServerSocket

Calculadora huella carbono.

Usuarios de telefonía celular

Últimos 30 Días

Generación de fragmentos d

Refactorización del envío arc

Carbon Footprint of Bitcoin

Actualizar a Plus

Jose Eduardo Olay Silis ...

23°C Mayorm. nublado

Búsqueda

```
socket1.close();
} catch (IOException | ClassNotFoundException e) {
    e.printStackTrace();
}

// Función para multiplicar dos matrices
private static double[][] multiplyMatrices(double[][] A, double[][] B) {
    int numRowsA = A.length;
    int numColsA = A[0].length;
    int numRowsB = B.length;
    int numColsB = B[0].length;

    if (numColsA != numRowsB) {
        System.err.println("No se pueden multiplicar las matrices. Las d");
        System.exit(1);
    }

    double[][] C = new double[numRowsA][numColsB];

    for (int i = 0; i < numRowsA; i++) {
        for (int j = 0; j < numColsB; j++) {
            for (int k = 0; k < numColsA; k++) {
                C[i][j] += A[i][k] * B[k][j];
            }
        }
    }

    return C;
}
```

Volver a generar

Envía un mensaje

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

01:50 p. m. 09/10/2023

Correo: Jose Eduardo Olay Silis - Outlook | SISDIS-4CV12: Tarea 3. Multiplicación | Calcular Producto de Matrices | Octave Online - Cloud IDE compatible | +

chat.openai.com/c/abc0fc2b-cee1-492a-82a9-de60f19670c1

+ Nuevo chat

Hoy

- Calcular Producto de
- Corrección de código Java
- Programa Cliente TCP Java

Ayer

- Programa Cliente TCP en Java
- Transposición de matriz en Java

Últimos 7 Días

- Proyección Ventas Netas 2023
- Puerto de ServerSocket
- Calculadora huella carbono.
- Usuarios de telefonía celular

Últimos 30 Días

- Generación de fragmentos d
- Refactorización del envío arc
- Carbon Footprint of Bitcoin
- Actualizar a Plus

Jose Eduardo Olay Silis ...

Proorcione el código del nodo1 para Recibir C1 al C36 de nodo 2 y 3

El código del Nodo 1 para recibir las matrices C1 al C36 enviadas por el Nodo 2 y el Nodo 3 sería el siguiente:

```
java
```

```
public class Nodo1 {
    public static void main(String[] args) {
        try {
            ServerSocket serverSocket = new ServerSocket(8080); // Puerto en el que escucha las conexiones entrantes

            // Esperar conexiones entrantes de Nodo 2 y Nodo 3
            Socket socket2 = serverSocket.accept();
            Socket socket3 = serverSocket.accept();

            // ... (Código para inicializar matrices A y B, enviar A1, A2, A3, B1, B2, B3)

            // Recibir C1 al C18 de Nodo 2
            ObjectInputStream in2 = new ObjectInputStream(socket2.getInputStream());
            double[][] C1 = (double[][]) in2.readObject();
            double[][] C2 = (double[][]) in2.readObject();
            double[][] C3 = (double[][]) in2.readObject();
            double[][] C4 = (double[][]) in2.readObject();
            double[][] C5 = (double[][]) in2.readObject();
            double[][] C6 = (double[][]) in2.readObject();
            double[][] C7 = (double[][]) in2.readObject();
            double[][] C8 = (double[][]) in2.readObject();
            double[][] C9 = (double[][]) in2.readObject();
            double[][] C10 = (double[][]) in2.readObject();
            double[][] C11 = (double[][]) in2.readObject();
            double[][] C12 = (double[][]) in2.readObject();
            double[][] C13 = (double[][]) in2.readObject();
            double[][] C14 = (double[][]) in2.readObject();
            double[][] C15 = (double[][]) in2.readObject();
            double[][] C16 = (double[][]) in2.readObject();
            double[][] C17 = (double[][]) in2.readObject();
            double[][] C18 = (double[][]) in2.readObject();

            // Recibir C19 al C36 de Nodo 3
            ObjectInputStream in3 = new ObjectInputStream(socket3.getInputStream());
            double[][] C19 = (double[][]) in3.readObject();
            double[][] C20 = (double[][]) in3.readObject();
            double[][] C21 = (double[][]) in3.readObject();
            double[][] C22 = (double[][]) in3.readObject();
            double[][] C23 = (double[][]) in3.readObject();
            double[][] C24 = (double[][]) in3.readObject();
            double[][] C25 = (double[][]) in3.readObject();
            double[][] C26 = (double[][]) in3.readObject();
            double[][] C27 = (double[][]) in3.readObject();
            double[][] C28 = (double[][]) in3.readObject();
            double[][] C29 = (double[][]) in3.readObject();
            double[][] C30 = (double[][]) in3.readObject();
            double[][] C31 = (double[][]) in3.readObject();
            double[][] C32 = (double[][]) in3.readObject();
            double[][] C33 = (double[][]) in3.readObject();
            double[][] C34 = (double[][]) in3.readObject();
            double[][] C35 = (double[][]) in3.readObject();
            double[][] C36 = (double[][]) in3.readObject();

            // Combinar matrices C1 al C36 para obtener la matriz final C
            // ... (Código para combinar matrices y calcular el checksum)

            // Mostrar las matrices A, B y C, y el checksum
            // Cerrar sockets y liberar recursos
            socket2.close();
            socket3.close();
        } catch (IOException | ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

Volver a generar

Envía un mensaje

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

23°C UV muy alto

Búsqueda

ESP LAA 01:51 p. m. 09/10/2023

Correo: Jose Eduardo Olay Silis - Outlook | SISDIS-4CV12: Tarea 3. Multiplicación | Calcular Producto de Matrices | Octave Online - Cloud IDE compatible | +

chat.openai.com/c/abc0fc2b-cee1-492a-82a9-de60f19670c1

+ Nuevo chat

Hoy

- Calcular Producto de
- Corrección de código Java
- Programa Cliente TCP Java

Ayer

- Programa Cliente TCP en Java
- Transposición de matriz en Java

Últimos 7 Días

- Proyección Ventas Netas 2023
- Puerto de ServerSocket
- Calculadora huella carbono.
- Usuarios de telefonía celular

Últimos 30 Días

- Generación de fragmentos d
- Refactorización del envío arc
- Carbon Footprint of Bitcoin
- Actualizar a Plus

Jose Eduardo Olay Silis ...

```
// Recibir C19 al C36 de Nodo 3
ObjectInputStream in3 = new ObjectInputStream(socket3.getInputStream());
double[][] C19 = (double[][]) in3.readObject();
double[][] C20 = (double[][]) in3.readObject();
double[][] C21 = (double[][]) in3.readObject();
double[][] C22 = (double[][]) in3.readObject();
double[][] C23 = (double[][]) in3.readObject();
double[][] C24 = (double[][]) in3.readObject();
double[][] C25 = (double[][]) in3.readObject();
double[][] C26 = (double[][]) in3.readObject();
double[][] C27 = (double[][]) in3.readObject();
double[][] C28 = (double[][]) in3.readObject();
double[][] C29 = (double[][]) in3.readObject();
double[][] C30 = (double[][]) in3.readObject();
double[][] C31 = (double[][]) in3.readObject();
double[][] C32 = (double[][]) in3.readObject();
double[][] C33 = (double[][]) in3.readObject();
double[][] C34 = (double[][]) in3.readObject();
double[][] C35 = (double[][]) in3.readObject();
double[][] C36 = (double[][]) in3.readObject();

// Combinar matrices C1 al C36 para obtener la matriz final C
// ... (Código para combinar matrices y calcular el checksum)

// Mostrar las matrices A, B y C, y el checksum
// Cerrar sockets y liberar recursos
socket2.close();
socket3.close();
```

Volver a generar

Envía un mensaje

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

23°C UV muy alto

Búsqueda

ESP LAA 01:51 p. m. 09/10/2023

Correo: Jose Eduardo Olay Silis - Outl... SISDIS-4CV12: Tarea 3. Multiplicación Calcular Producto de Matrices Octave Online - Cloud IDE compatible chat.openai.com/c/abc0fc2b-cee1-492a-82a9-de60f19670c1

+ Nuevo chat

Hoy

- Calcular Producto de
- Corrección de código Java
- Programa Cliente TCP Java

Ayer

- Programa Cliente TCP en Jav...
- Transposición de matriz en J...

Últimos 7 Días

- Proyección Ventas Netas 20...
- Puerto de ServerSocket
- Calculadora huella carbono.
- Usuarios de telefonía celular

Últimos 30 Días

- Generación de fragmentos d...
- Refactorización del envío arc...
- Carbon Footprint of Bitcoin
- Actualizar a Plus

Jose Eduardo Olay Silis ...

Una vez recibidas las matrices C1 a C18 y de C19 a C36

```
// Recibir C1 al C18 de nodo 2
// Crear un ObjectInputStream para recibir los datos
ObjectInputStream in2 = new ObjectInputStream(socket2.getInputStream());

// Crear un arreglo de matrices para almacenar C1 al C18
double[][] matricesCitoC18 = new double[18][];
for (int i = 0; i < 18; i++) {
    matricesCitoC18[i] = (double[][]) in2.readObject();
}

// Recibir las matrices C1 al C18
for (int i = 0; i < 18; i++) {
    matricesCitoC18[i] = (double[][]) in2.readObject();
}

// Recibir C19 al C36 de nodo 3
ObjectInputStream in3 = new ObjectInputStream(socket3.getInputStream());
// Crear un arreglo para almacenar las matrices C19 a C36
double[][] C19toC36 = new double[18][];
for (int i = 0; i < 18; i++) {
    C19toC36[i] = (double[][]) in3.readObject();
}

// Recibir las matrices C19 a C36 del Nodo 3
for (int i = 0; i < 18; i++) {
    C19toC36[i] = (double[][]) in3.readObject();
}
```

Proporciona el código para combinar dichas matrices recibidas tanto del nodo 2 como del nodo 3 para formar la matriz final C

Para combinar las matrices C1 al C18 y C19 al C36 que has recibido del Nodo 2 y respectivamente, y formar la matriz final C, puedes utilizar el siguiente código:

Envía un mensaje

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

23°C UV muy alto

Búsqueda

ESP LAA 01:51 p. m. 09/10/2023

Correo: Jose Eduardo Olay Silis - Outl... SISDIS-4CV12: Tarea 3. Multiplicación Calcular Producto de Matrices Octave Online - Cloud IDE compatible chat.openai.com/c/abc0fc2b-cee1-492a-82a9-de60f19670c1

+ Nuevo chat

Hoy

- Calcular Producto de
- Corrección de código Java
- Programa Cliente TCP Java

Ayer

- Programa Cliente TCP en Jav...
- Transposición de matriz en J...

Últimos 7 Días

- Proyección Ventas Netas 20...
- Puerto de ServerSocket
- Calculadora huella carbono.
- Usuarios de telefonía celular

Últimos 30 Días

- Generación de fragmentos d...
- Refactorización del envío arc...
- Carbon Footprint of Bitcoin
- Actualizar a Plus

Jose Eduardo Olay Silis ...

Para combinar las matrices C1 al C18 y C19 al C36 que has recibido del Nodo 2 y Nodo 3, respectivamente, y formar la matriz final C, puedes utilizar el siguiente código:

```
java
// Combinar matrices C1 al C18 y C19 al C36 para obtener la matriz final C
double[][][] C = new double[6][6][]; // Matriz C dividida en 6x6 bloques de 6x6

int blockIndex = 0;

for (int i = 0; i < 6; i++) {
    for (int j = 0; j < 6; j++) {
        if (blockIndex < 18) {
            C[i][j] = matricesCitoC18[blockIndex];
        } else {
            C[i][j] = C19toC36[blockIndex - 18];
        }
        blockIndex++;
    }
}

// Ahora, la matriz C es un arreglo 6x6 de bloques de N/6 x N/6, y luego
```

Este código crea una matriz 'C' que es un arreglo 6x6 de bloques de N/6 x N/6, y luego asigna las matrices C1 al C18 del Nodo 2 y C19 al C36 del Nodo 3 a las ubicaciones correspondientes en la matriz 'C'. Esto te dará la matriz final C dividida en los 36 bloques.

Envía un mensaje

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

23°C UV muy alto

Búsqueda

ESP LAA 01:52 p. m. 09/10/2023

Correo: Jose Eduardo Olay Silis - Outbox SISDIS-4CV12: Tarea 3. Multiplicación Calcular Producto de Matrices Octave Online - Cloud IDE compatible chat.openai.com/c/abc0fc2b-cee1-492a-82a9-de60f19670c1

Y proporciona el código para calcular el checksum de esa matriz C y de igual manera para imprimir la matriz C.

Por supuesto, aquí está el código para calcular el checksum de la matriz C y para imprimir la matriz C:

```
java
// Calcular el checksum de C
double checksum = 0.0;
for (int i = 0; i < 6; i++) {
    for (int j = 0; j < 6; j++) {
        double[][] block = C[i][j];
        for (int row = 0; row < block.length; row++) {
            for (int col = 0; col < block[0].length; col++) {
                checksum += block[row][col];
            }
        }
    }
}

// Imprimir la matriz C
System.out.println("Matriz C:");
for (int i = 0; i < 6; i++) {
    for (int j = 0; j < 6; j++) {
        double[][] block = C[i][j];
        for (int row = 0; row < block.length; row++) {
            for (int col = 0; col < block[0].length; col++) {
                System.out.print(block[row][col] + " ");
            }
        }
        System.out.println();
    }
}

System.out.println("Checksum de C: " + checksum);
```

Envía un mensaje Volver a generar

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

23°C UV muy alto Búsqueda ESP LAA 01:52 p. m. 09/10/2023

Correo: Jose Eduardo Olay Silis - Outbox SISDIS-4CV12: Tarea 3. Multiplicación Calcular Producto de Matrices Octave Online - Cloud IDE compatible chat.openai.com/c/abc0fc2b-cee1-492a-82a9-de60f19670c1

Default (GPT-3.5)

```
for (int row = 0; row < block.length; row++) {
    checksum += block[row][col];
}

// Imprimir la matriz C
System.out.println("Matriz C:");
for (int i = 0; i < 6; i++) {
    for (int j = 0; j < 6; j++) {
        double[][] block = C[i][j];
        for (int row = 0; row < block.length; row++) {
            for (int col = 0; col < block[0].length; col++) {
                System.out.print(block[row][col] + " ");
            }
        }
        System.out.println();
    }
}

System.out.println("Checksum de C: " + checksum);
```

Este código calcula el checksum sumando todos los elementos de la matriz C y luego imprime la matriz C y el checksum resultante. Debería agregarlo al final de su código en el nodo 1 después de combinar las matrices C1 al C36 para obtener la matriz final C.

Envía un mensaje Volver a generar

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

23°C UV muy alto Búsqueda ESP LAA 01:52 p. m. 09/10/2023

Cada parte que el Chat iba proporcionando la fui colocando de acuerdo al esqueleto inicial que había propuesto, al compilarlo simplemente tuve que meter mano en algunas cuestiones que el Chat no pudo resolverme, por ejemplo, la división de la matriz en subpartes, dicha implementación la realicé por mi cuenta.

```
// Implementación Realizada por Jose Eduardo
```

```

// Dividir A y BT en porciones y enviar a nodos 2 y 3
int numRowsPerNode = N / 6; // Dividir A en 6 partes
double[][] A1 = new double[numRowsPerNode][M];
double[][] A2 = new double[numRowsPerNode][M];
double[][] A3 = new double[numRowsPerNode][M];
double[][] A4 = new double[numRowsPerNode][M];
double[][] A5 = new double[numRowsPerNode][M];
double[][] A6 = new double[numRowsPerNode][M];

// Dividir la matriz A en las 6 partes
for (int i = 0; i < N; i++) {
    for (int j = 0; j < M; j++) {
        int part = i / numRowsPerNode;
        int rowInPart = i % numRowsPerNode;

        switch (part) {
            case 0:
                A1[rowInPart][j] = A[i][j];
                break;
            case 1:
                A2[rowInPart][j] = A[i][j];
                break;
            case 2:
                A3[rowInPart][j] = A[i][j];
                break;
            case 3:
                A4[rowInPart][j] = A[i][j];
                break;
            case 4:
                A5[rowInPart][j] = A[i][j];
                break;
            case 5:
                A6[rowInPart][j] = A[i][j];
                break;
        }
    }
}

```

De igual manera me di cuenta de que en la mención general de características no coloque la inicialización de las matrices por lo que le pedí al chat que las realizara

Haber de acuerdo al código que ya me proporcionaste solo necesito que lo modifiques y adecues de manera que tengas encuentra la siguiente estructura:
recuerda que la matriz C que es la del resultado del producto de la Matriz A y B se divide en 36 matrices cuadradas: de N/6 renglones y N/6 columnas de la siguiente manera: C= ((C1,C2,C3,C4,C5,C6),(C7,C8,C9,C10,C11,C12),...,(C31,C32,C33,C34,C35,C36)).

Ahora, el Nodo 1 Recuerda que inicializa las matrices A y B, despues obtiene la matriz transpuesta de B, despues envia las matrices A1, A2, A3, BT1, BT2, BT3, BT4, BT5 y BT6 al nodo 2, asi como las matrices A4, A5, A6, BT1, BT2, BT3, BT4, BT5 y BT6 al nodo 3. Para que en el nodo 2 se hagan las operaciones correspondientes que y esta envié al nodo 1 las matrices de C1 a la C18 y en el nodo 3 las operaciones correspondientes para que envíe al nodo 1 las matrices de la C19 a la C36 y finalmente en el nodo 1 se obtenga la matriz C a partir de las matrices c1 a la C36.

Realiza los cambios necesarios en el código que me proporcionaste para que se puedan implementar dichas características

Entendido, aquí tienes el código modificado para que se ajuste a la estructura que mencionaste:

Nodo 1:

```
java
import java.io.*;
import java.net.*;
import java.util.Random;

public class MatrixMultiplication {
    ...
    serverSocket.close();
} catch (IOException | ClassNotFoundException e) {
    e.printStackTrace();
}
// ... (código restante)
}

private static double[][] initializeMatrixA(int N, int M) {
    double[][] A = new double[N][M];
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            A[i][j] = 5 * i + 2 * j;
        }
    }
    return A;
}

private static double[][] initializeMatrixB(int N, int M) {
    double[][] B = new double[N][M];
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            B[i][j] = 6 * i + 3 * j;
        }
    }
    return B;
}
```

Envía un mensaje

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

Jose Eduardo Olay Silis

23°C Mayorm. nublado

Default (GPT-3.5)

```
...
serverSocket.close();
} catch (IOException | ClassNotFoundException e) {
    e.printStackTrace();
}
// ... (código restante)
}

private static double[][] initializeMatrixA(int N, int M) {
    double[][] A = new double[N][M];
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            A[i][j] = 5 * i + 2 * j;
        }
    }
    return A;
}

private static double[][] initializeMatrixB(int N, int M) {
    double[][] B = new double[N][M];
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            B[i][j] = 6 * i + 3 * j;
        }
    }
    return B;
}
```

Envía un mensaje

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT September 25 Version

Jose Eduardo Olay Silis

23°C Mayorm. nublado

Se puede observar incluso en la misma pantalla que cometí nuevamente un error en el prompt debido a que la matriz B es de [M] [N] y en la implementación las genero de [N] [N] por lo que esos cambios y algunas impresiones de pantalla en general para que se vea como se va realizando el proceso distribuido las agregue por mi cuenta quedando finalmente el código de la siguiente manera:

```
import java.io.*;
import java.net.*;

public class multiMatriz {
    public static void main(String[] args) {
        if (args.length != 3) {
            System.err.println("Debes proporcionar el número de nodo (1, 2 o 3), N y M como argumentos.");
            System.exit(1);
        }

        int nodeNumber = Integer.parseInt(args[0]);
        int N = Integer.parseInt(args[1]);
        int M = Integer.parseInt(args[2]);

        if (nodeNumber == 1) {
            // Nodo 1
            try {
                ServerSocket serverSocket = new ServerSocket(8080);
                System.out.println("Esperando conexión de nodo 2 y 3....");
                // Esperar conexiones entrantes de nodo 2 y nodo 3
                Socket socket2 = serverSocket.accept();
                Socket socket3 = serverSocket.accept();
                System.out.println("\nConexión realizada");

                // Inicialización de matrices A y B
                double[][] A = initializeMatrixA(N, M);
                double[][] B = initializeMatrixB(N, M);

                // Cálculo de la transpuesta de B (BT)
                double[][] BT = new double[N][M];
                for (int i = 0; i < N; i++) {
                    for (int j = 0; j < M; j++) {
                        BT[i][j] = B[j][i];
                    }
                }
            }
            //Implementación Realizada por Jose Eduardo

            // Dividir A y BT en porciones y enviar a nodos 2 y 3
            int numRowsPerNode = N / 6; // Dividir A en 6 partes
            double[][] A1 = new double[numRowsPerNode][M];
            double[][] A2 = new double[numRowsPerNode][M];
            double[][] A3 = new double[numRowsPerNode][M];
            double[][] A4 = new double[numRowsPerNode][M];
            double[][] A5 = new double[numRowsPerNode][M];
            double[][] A6 = new double[numRowsPerNode][M];
        }
    }
}
```

```

// Dividir la matriz A en las 6 partes
for (int i = 0; i < N; i++) {
    for (int j = 0; j < M; j++) {
        int part = i / numRowsPerNode;
        int rowInPart = i % numRowsPerNode;

        switch (part) {
            case 0:
                A1[rowInPart][j] = A[i][j];
                break;
            case 1:
                A2[rowInPart][j] = A[i][j];
                break;
            case 2:
                A3[rowInPart][j] = A[i][j];
                break;
            case 3:
                A4[rowInPart][j] = A[i][j];
                break;
            case 4:
                A5[rowInPart][j] = A[i][j];
                break;
            case 5:
                A6[rowInPart][j] = A[i][j];
                break;
        }
    }
}

int numRowsPerNodeB = N / 6; // Dividir A en 6 partes
double[][] BT1 = new double[numRowsPerNode][M];
double[][] BT2 = new double[numRowsPerNode][M];
double[][] BT3 = new double[numRowsPerNode][M];
double[][] BT4 = new double[numRowsPerNode][M];
double[][] BT5 = new double[numRowsPerNode][M];
double[][] BT6 = new double[numRowsPerNode][M];

// Dividir la matriz BT en las 6 partes
for (int i = 0; i < N; i++) {
    for (int j = 0; j < M; j++) {
        int part = i / numRowsPerNodeB;
        int rowInPart = i % numRowsPerNodeB;

        switch (part) {
            case 0:
                BT1[rowInPart][j] = BT[i][j];
                break;
            case 1:

```

```

                BT2[rowInPart][j] = BT[i][j];
                break;
            case 2:
                BT3[rowInPart][j] = BT[i][j];
                break;
            case 3:
                BT4[rowInPart][j] = BT[i][j];
                break;
            case 4:
                BT5[rowInPart][j] = BT[i][j];
                break;
            case 5:
                BT6[rowInPart][j] = BT[i][j];
                break;
        }
    }
}

ObjectOutputStream out2 = new
ObjectOutputStream(socket2.getOutputStream());
ObjectOutputStream out3 = new
ObjectOutputStream(socket3.getOutputStream());

// Enviar las partes correspondientes de A y BT a nodo 2
System.out.println("\nEnviando matrices a Nodo 2");
out2.writeObject(A1);
out2.writeObject(A2);
out2.writeObject(A3);
out2.writeObject(BT1);
out2.writeObject(BT2);
out2.writeObject(BT3);
out2.writeObject(BT4);
out2.writeObject(BT5);
out2.writeObject(BT6);

// Enviar las partes correspondientes de A y BT a nodo 3
System.out.println("\nEnviando matrices a Nodo 3");
out3.writeObject(A4);
out3.writeObject(A5);
out3.writeObject(A6);
out3.writeObject(BT1);
out3.writeObject(BT2);
out3.writeObject(BT3);
out3.writeObject(BT4);
out3.writeObject(BT5);
out3.writeObject(BT6);

```

```

        // Recibir C1 al C18 de nodo 2
        // Crear un ObjectInputStream para recibir los datos
        ObjectInputStream in2 = new
ObjectInputStream(socket2.getInputStream());

        // Crear un arreglo de matrices para almacenar C1 al C18
        double[][][] matricesC1toC18 = new double[18][][];
}

        // Recibir las matrices C1 al C18
        for (int i = 0; i < 18; i++) {
            matricesC1toC18[i] = (double[][]) in2.readObject();
}
System.out.println("\nMatrices recibidas del Nodo2");

        // Recibir C19 al C36 de nodo 3
        ObjectInputStream in3 = new
ObjectInputStream(socket3.getInputStream());
        // Crear un arreglo para almacenar las matrices C19 a C36
        double[][][] C19toC36 = new double[18][][];
}

        // Recibir las matrices C19 a C36 del Nodo 3
        for (int i = 0; i < 18; i++) {
            C19toC36[i] = (double[][]) in3.readObject();
}
System.out.println("\nMatrices recibidas del Nodo3");

        // Combinar matrices C1 al C18 y C19 al C36 para obtener la matriz
final C
        double[][][][] C = new double[6][6][][];
// Matriz C dividida en 6x6
bloques de N/6 x N/6

        int blockIndex = 0;

        for (int i = 0; i < 6; i++) {
            for (int j = 0; j < 6; j++) {
                if (blockIndex < 18) {
                    C[i][j] = matricesC1toC18[blockIndex];
                } else {
                    C[i][j] = C19toC36[blockIndex - 18];
                }
                blockIndex++;
            }
}
// Ahora, la matriz C es un arreglo 6x6 de bloques de N/6 x N/6
System.out.println("\ncalculando checksum de C...");
// Calcular el checksum de C

```

```

        double checksum = 0.0;
        for (int i = 0; i < 6; i++) {
            for (int j = 0; j < 6; j++) {
                double[][] block = C[i][j];
                for (int row = 0; row < block.length; row++) {
                    for (int col = 0; col < block[0].length; col++) {
                        checksum += block[row][col];
                    }
                }
            }
        }

        if(N==6 || M==5){
            // Mostrar las matrices A, B y C
            System.out.println("Matriz A:");
            printMatrix(A);
            System.out.println("\nMatriz B:");
            printMatrix(B);
            // Imprimir la matriz C
            System.out.println("\nMatriz C:");
            for (int i = 0; i < 6; i++) {
                for (int j = 0; j < 6; j++) {
                    double[][] block = C[i][j];
                    for (int row = 0; row < block.length; row++) {
                        for (int col = 0; col < block[0].length; col++) {
                            System.out.print(block[row][col] + "   ");
                        }
                    }
                    System.out.println();
                }
            }
        }

        System.out.println("Checksum de C: " + checksum);

        // Cerrar sockets y limpiar recursos
        socket2.close();
        socket3.close();
        serverSocket.close();
    } catch (IOException | ClassNotFoundException e) {
        e.printStackTrace();
    }
} else if (nodeNumber == 2) {
    // Nodo 2
    try {
        Socket socket1 = new Socket("localhost", 8080); // Reemplaza
        "NODO1_IP" con la IP del Nodo 1
    }
}

```

```

        // Recibir las partes correspondientes de A y BT del Nodo 1
        ObjectInputStream in1 = new
ObjectInputStream(socket1.getInputStream());

        double[][] A1 = (double[][][]) in1.readObject();
        double[][] A2 = (double[][][]) in1.readObject();
        double[][] A3 = (double[][][]) in1.readObject();
        double[][] BT1 = (double[][][]) in1.readObject();
        double[][] BT2 = (double[][][]) in1.readObject();
        double[][] BT3 = (double[][][]) in1.readObject();
        double[][] BT4 = (double[][][]) in1.readObject();
        double[][] BT5 = (double[][][]) in1.readObject();
        double[][] BT6 = (double[][][]) in1.readObject();

        System.out.println("Matrices recibidas");
        System.out.println("\nRealizando cálculos para obtener
submatrices....");
        // Realizar los cálculos para obtener C1 al C18
        double[][] C1 = multiplyMatrices(A1, BT1);
        double[][] C2 = multiplyMatrices(A1, BT2);
        double[][] C3 = multiplyMatrices(A1, BT3);
        double[][] C4 = multiplyMatrices(A1, BT4);
        double[][] C5 = multiplyMatrices(A1, BT5);
        double[][] C6 = multiplyMatrices(A1, BT6);
        double[][] C7 = multiplyMatrices(A2, BT1);
        double[][] C8 = multiplyMatrices(A2, BT2);
        double[][] C9 = multiplyMatrices(A2, BT3);
        double[][] C10 = multiplyMatrices(A2, BT4);
        double[][] C11 = multiplyMatrices(A2, BT5);
        double[][] C12 = multiplyMatrices(A2, BT6);
        double[][] C13 = multiplyMatrices(A3, BT1);
        double[][] C14 = multiplyMatrices(A3, BT2);
        double[][] C15 = multiplyMatrices(A3, BT3);
        double[][] C16 = multiplyMatrices(A3, BT4);
        double[][] C17 = multiplyMatrices(A3, BT5);
        double[][] C18 = multiplyMatrices(A3, BT6);

        // Enviar C1 al C18 a nodo 1
        ObjectOutputStream out = new
ObjectOutputStream(socket1.getOutputStream());

        // Crear un arreglo de matrices C1 al C18
        double[][][] matricesC = {C1, C2, C3, C4, C5, C6, C7, C8, C9, C10,
C11, C12, C13, C14, C15, C16, C17, C18};
        // Enviar las matrices C1 al C18
        for (int i = 0; i < matricesC.length; i++) {

```

```

        out.writeObject(matricesC[i]);
    }
    System.out.println("Matrices enviadas");

    // Cerrar socket y limpiar recursos
    socket1.close();
} catch (IOException | ClassNotFoundException e) {
    e.printStackTrace();
}
} else if (nodeNumber == 3) {
    // Nodo 3
    try {
        Socket socket1 = new Socket("localhost", 8080); // Reemplaza
"NODO1_IP" con la IP del Nodo 1

        // Recibir las partes correspondientes de A y BT del Nodo 1
        ObjectInputStream in1 = new
ObjectInputStream(socket1.getInputStream());
        // Crear variables para almacenar las matrices recibidas
        double[][] A4 = (double[][]) in1.readObject();
        double[][] A5 = (double[][]) in1.readObject();
        double[][] A6 = (double[][]) in1.readObject();
        double[][] BT1 = (double[][]) in1.readObject();
        double[][] BT2 = (double[][]) in1.readObject();
        double[][] BT3 = (double[][]) in1.readObject();
        double[][] BT4 = (double[][]) in1.readObject();
        double[][] BT5 = (double[][]) in1.readObject();
        double[][] BT6 = (double[][]) in1.readObject();

        // Realizar los cálculos para obtener C19 al C36
        System.out.println("Matrices recibidas");
        System.out.println("\nRealizando cálculos para obtener
submatrices....");
        double[][] C19 = multiplyMatrices(A4, BT1);
        double[][] C20 = multiplyMatrices(A4, BT2);
        double[][] C21 = multiplyMatrices(A4, BT3);
        double[][] C22 = multiplyMatrices(A4, BT4);
        double[][] C23 = multiplyMatrices(A4, BT5);
        double[][] C24 = multiplyMatrices(A4, BT6);
        double[][] C25 = multiplyMatrices(A5, BT1);
        double[][] C26 = multiplyMatrices(A5, BT2);
        double[][] C27 = multiplyMatrices(A5, BT3);
        double[][] C28 = multiplyMatrices(A5, BT4);
        double[][] C29 = multiplyMatrices(A5, BT5);
        double[][] C30 = multiplyMatrices(A5, BT6);
        double[][] C31 = multiplyMatrices(A6, BT1);
        double[][] C32 = multiplyMatrices(A6, BT2);
    }
}

```

```

        double[][] C33 = multiplyMatrices(A6, BT3);
        double[][] C34 = multiplyMatrices(A6, BT4);
        double[][] C35 = multiplyMatrices(A6, BT5);
        double[][] C36 = multiplyMatrices(A6, BT6);

        // Enviar (C19 al C36) a nodo 1
        ObjectOutputStream out = new
ObjectOutputStream(socket1.getOutputStream());

        // Crear un arreglo de matrices C19 al C36
        double[][][] matricesC = {C19, C20, C21, C22, C23, C24, C25, C26,
C27, C28, C29, C30, C31, C32, C33, C34, C35, C36};
        // Enviar las matrices C19 al C36
        for (int i = 0; i < matricesC.length; i++) {
            out.writeObject(matricesC[i]);
        }

        System.out.println("Matrices enviadas");
        // Cerrar socket y limpiar recursos
        socket1.close();
    } catch (IOException | ClassNotFoundException e) {
        e.printStackTrace();
    }
} else {
    System.err.println("Número de nodo no válido. Debe ser 1, 2 o 3.");
    System.exit(1);
}
}

private static double[][] initializeMatrixA(int N, int M) {
    double[][] A = new double[N][M];
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            A[i][j] = 5 * i - 2 * j;
        }
    }
    return A;
}

private static double[][] initializeMatrixB(int N, int M) {
    double[][] B = new double[M][N];
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < M; j++) {
            B[j][i] = 6 * i + 3 * j;
        }
    }
}

```

```

        return B;
    }

    // Función para multiplicar dos matrices
    private static double[][] multiplyMatrices(double[][] A, double[][] B) {

        int numRows=A.length;
        int numCols=A[0].length;

        if(numRows != B.length || numCols != B[0].length){
            System.out.println("No se pueden multiplicar las matrices. Las dimensiones no coinciden.");
            System.exit(1);
        }

        double[][] C = new double[numRows][numCols];

        for (int i = 0; i < numRows; i++) {
            for (int j = 0; j < numCols; j++) {
                C[i][j] = A[i][j] * B[i][j];
            }
        }

        return C;
    }

    private static void printMatrix(double[][] matrix) {
        int numRows = matrix.length;
        int numCols = matrix[0].length;

        for (int i = 0; i < numRows; i++) {
            for (int j = 0; j < numCols; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println(" ");
        }
    }
}

```

Fase 2

Una vez realizado el código comencé a crear las maquinas virtuales que simbolizarán cada uno de los nodos. A continuación, se describirá el proceso de creación del Nodo 1.

Comenzamos colocando el nombre de la máquina virtual así como la imagen.

Detalles del proyecto
Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción * (Azure for Students)
Grupo de recursos * (Nuevo T3-2020630347-1.group)

Detalles de instancia
Nombre de máquina virtual * (T3-2020630347-1)
Región * (US) East US 2
Opciones de disponibilidad (Zona de disponibilidad)
Zona de disponibilidad * (Zonas 1)
Ahora puede seleccionar varias zonas. Si selecciona varias zonas, se creará una VM por zona. [Más información](#)

Revisar y crear < Anterior Siguiente: Discos > Envíe comentarios

Seleccionamos el tamaño de memoria, el Usuario y la contraseña para la máquina y habilitamos el puerto 22 para utilizar SSH.

Cuenta de administrador
Tipo de autenticación (Clave pública SSH, Contraseña, Contraseña)* (Contraseña)
Nombre de usuario * (Ubuntu)
Contraseña * (*****)
Confirmar contraseña * (*****)

Reglas de puerto de entrada
Seleccione los puertos de red de máquina virtual que son accesibles desde la red Internet pública. Puede especificar acceso de red más limitado o granular en la pestaña Red.

Puertos de entrada públicos * (Ninguno, Permitir los puertos seleccionados) (Permitir los puertos seleccionados)
Seleccionar puertos de entrada * (SSH (22))
Se bloquea todo el tráfico de Internet de forma predeterminada. Puede cambiar las reglas del puerto de entrada en la página VM > Redes.

Revisar y crear < Anterior Siguiente: Discos > Envíe comentarios

Seleccionamos el tipo de disco de sistema operativo, en este caso HDD estándar.

Crear una máquina virtual

Tamaño del disco del SO: Valor predeterminado de la imagen (30 GiB)

Tipo de disco del sistema operativo: HDD estándar (almacenamiento con redundancia local)

Eliminar con VM:

Administración de claves: Clave administrada por la plataforma

Habilitar compatibilidad con Ultra Disks: No se admite el disco Ultra con el tipo de seguridad seleccionado.

Discos de datos para T3-2020630347-1

Crear y adjuntar un nuevo disco Asociar un disco existente

Opciones avanzadas

Revisar y crear < Anterior Siguiente: Redes > Enviar comentarios

Se crea la red virtual y la IP pública de la máquina.

Crear una máquina virtual

Interfaz de red

Red virtual: (nuevo) T3-2020630347-1-vnet

Subred: (nuevo) default (10.1.0.0/24)

IP pública: (nuevo) T3-2020630347-1-ip

Grupo de seguridad de red de NIC: Básico

Puertos de entrada públicos: Permitir los puertos seleccionados

Seleccionar puertos de entrada: SSH (22)

Este permitirá que todas las direcciones IP accedan a la máquina virtual. Esto solo se recomienda para los puertos. Use los controles avanzados de la pestaña Redes a fin de crear reglas para limitar el tráfico entrante a las direcciones IP conocidas.

Revisar y crear < Anterior Siguiente: Administración > Enviar comentarios

Se deshabilita el diagnóstico de arranque.

Correo: Jose Eduardo Olay Silis - Outl... | SISDIS-4CV12: Tarea 3. Multiplicación | Calcular Producto de Matrices | Crear una máquina virtual - Micr... | Curso: Desarrollo de Sistemas Distribu... +

Microsoft Azure Buscar recursos, servicios y documentos (G+/)

Inicio > Máquinas virtuales > Crear una máquina virtual ...

Datos básicos Discos Redes Administración Supervisión Opciones avanzadas Etiquetas Revisar y crear

Configure las opciones de supervisión de la máquina virtual.

Alertas

Habilitar reglas de alerta recomendadas

Diagnóstico

Diagnósticos de arranque Habilitar con la cuenta de almacenamiento administrada (recomendado) Habilitar con la cuenta de almacenamiento personalizada Deshabilitar

Habilitar diagnósticos del SO invitado

Revisar y crear < Anterior Siguiente: Opciones avanzadas > Enviar comentarios

Se revisa la máquina virtual para posteriormente pasar al proceso de creación de la misma.

Correo: Jose Eduardo Olay Silis - Outl... | SISDIS-4CV12: Tarea 3. Multiplicación | Calcular Producto de Matrices | Crear una máquina virtual - Micr... | Curso: Desarrollo de Sistemas Distribu... +

Microsoft Azure Buscar recursos, servicios y documentos (G+/)

Inicio > Máquinas virtuales > Crear una máquina virtual ...

✓ Validación superada

Datos básicos Discos Redes Administración Supervisión Opciones avanzadas Etiquetas Revisar y crear

El costo que se indica a continuación es una estimación y no el precio final. Use [Calculadora de precios](#) para todas sus necesidades de precios.

Price

1 X Standard B1s by Microsoft Subscription credits apply 0,0104 USD/hr Pricing for other VM sizes

TERMS

By clicking "Crear", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

Name JOSE EDUARDO OLAY SILIS

Crear < Anterior Siguiente > Descargar una plantilla para la automatización Enviar comentarios

Una vez creada esa máquina virtual procedemos a crear las otras 2 que simbolizarán los nodos restantes quedando de la siguiente manera.

Nombre	Tipo	Suscripción	Grupo de recursos	Ubicación	Estado	Sistema operativo	Tamaño	Dirección IP pública	Discos
T3-2020630347-1	Máquina virtual	Azure for Students	T3-2020630347-1_gro...	East US 2	En ejecución	Linux	Standard_B1s	172.200.169.25	1
T3-2020630347-2	Máquina virtual	Azure for Students	T3-2020630347-2_gro...	East US 2	En ejecución	Linux	Standard_B1s	172.200.169.113	1
T3-2020630347-3	Máquina virtual	Azure for Students	T3-2020630347-3_gro...	East US 2	En ejecución	Linux	Standard_B1s	-	1

Windows Taskbar showing weather (23°C), search, and system status (03:11 p.m., 09/10/2023).

Fase 3

Una vez creadas las máquinas virtuales comenzamos a realizar su configuración. En esta sección se realizará la configuración de la máquina virtual 1 y dicho proceso se repetirá en las máquinas virtuales que representan el nodo 2 y 3.

Una vez realizada la conexión a la máquina virtual mediante los siguientes comandos se instala el jdk a las máquinas virtuales:

- sudo add-apt-repository ppa:openjdk-r/ppa
- sudo apt-get update
- sudo apt install openjdk-8-jdk-headless

Una vez aplicados dichos comandos a las 3 máquinas podemos observar que efectivamente se ha instalado el jdk en las 3.

```

Ubuntu@T3-2020630347-1:~$ java -version
openjdk version "1.8.0_382"
OpenJDK Runtime Environment (build 1.8.0_382-8u382-ga-1~20.04.1-b05)
OpenJDK 64-Bit Server VM (build 25.382-b05, mixed mode)
Ubuntu@T3-2020630347-1:~$

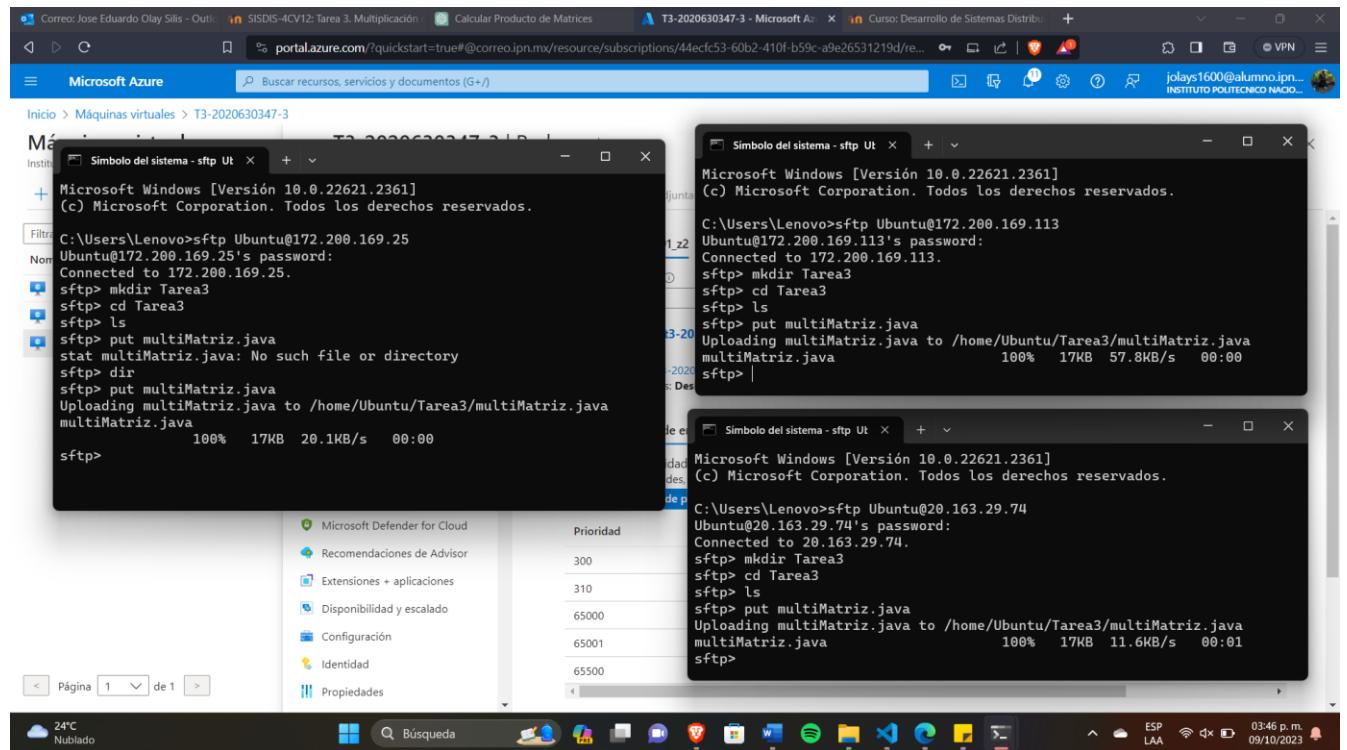
Ubuntu@T3-2020630347-2:~$ java -version
openjdk version "1.8.0_382"
OpenJDK Runtime Environment (build 1.8.0_382-8u382-ga-1~20.04.1-b05)
OpenJDK 64-Bit Server VM (build 25.382-b05, mixed mode)
Ubuntu@T3-2020630347-2:~$

Ubuntu@T3-2020630347-3:~$ java -version
openjdk version "1.8.0_382"
OpenJDK Runtime Environment (build 1.8.0_382-8u382-ga-1~20.04.1-b05)
OpenJDK 64-Bit Server VM (build 25.382-b05, mixed mode)
Ubuntu@T3-2020630347-3:~|

```

Windows Taskbar showing weather (24°C), search, and system status (03:36 p.m., 09/10/2023).

Ahora mediante el protocolo SFTP se mandará el archivo para que se puedan compilar y ejecutar en cada una de las máquinas.



Fase 4

Por último, ya una vez puestos los archivos dentro de las máquinas virtuales procedemos a realizar la ejecución del programa mostrando los resultados a continuación.

Caso 1:

N=6 M=5

```

Ubuntu@T3-2020630347-1:~/Tarea3$ java multiMatriz 1 6 5
Esperando conexión de nodo 2 y 3...
Conexión realizada
Enviando matrices a Nodo 2
Enviando matrices a Nodo 3
Matrices recibidas del Nodo2
Matrices recibidas del Nodo3
Calculando checksum de C...
Matriz A:
0.0 -2.0 -4.0 -6.0 -8.0
5.0 3.0 1.0 -1.0 -3.0
10.0 8.0 6.0 4.0 2.0
15.0 13.0 11.0 9.0 7.0
20.0 18.0 16.0 14.0 12.0
25.0 23.0 21.0 19.0 17.0

Matriz B:
0.0 6.0 12.0 18.0 24.0 30.0
3.0 9.0 15.0 21.0 27.0 33.0
6.0 12.0 18.0 24.0 30.0 36.0
9.0 15.0 21.0 27.0 33.0 39.0
12.0 18.0 24.0 30.0 36.0 42.0

Matriz C:
0.0 -6.0 -24.0 -54.0 -96.0
0.0 -18.0 -48.0 -90.0 -144.0
0.0 -30.0 -72.0 -126.0 -192.0
0.0 -42.0 -96.0 -162.0 -240.0
0.0 -54.0 -120.0 -198.0 -288.0
0.0 -66.0 -144.0 -234.0 -336.0
0.0 9.0 6.0 -9.0 -36.0
30.0 27.0 12.0 -15.0 -54.0
60.0 45.0 18.0 -21.0 -72.0

Ubuntu@T3-2020630347-2:~/Tarea3$ java multiMatriz 2 6 5
Matrices recibidas
Realizando cálculos para obtener submatrices...
Matrices enviadas
Ubuntu@T3-2020630347-2:~/Tarea3$

Ubuntu@T3-2020630347-3:~/Tarea3$ java multiMatriz 3 6 5
Matrices recibidas
Realizando cálculos para obtener submatrices...
Matrices enviadas
Ubuntu@T3-2020630347-3:~/Tarea3$

```

```

Ubuntu@T3-2020630347-1:~ % java multiMatriz 2 6 5
Matrices recibidas

Realizando cálculos para obtener submatrices...
Matrices enviadas
Ubuntu@T3-2020630347-2:~/Tarea3$ |
```

Adjunta: -130_z2

Adjunta: IP (principal)

Adjunta: t3-20

Adjunta: T3-2020630347-1-vnet/default IP pública de NIC: 172.200.169.25 IP privada de NIC: 10.1.0.4

Adjunta: Deshabilitado

```

Ubuntu@T3-2020630347-3:~ % java multiMatriz 3 6 5
Matrices recibidas

Realizando cálculos para obtener submatrices...
Matrices enviadas
Ubuntu@T3-2020630347-3:~/Tarea3$ |
```

Caso 2:

N=3000 M=2000

```

Ubuntu@T3-2020630347-1:~/Tarea3$ java multiMatriz 1 3000 2000
Esperando conexión de nodo 2 y 3.....
Conexión realizada
Enviando matrices a Nodo 2
Enviando matrices a Nodo 3
Matrices recibidas del Nodo2
Matrices recibidas del Nodo3

calculando checksum de C...
Checksum de C: 1.281878973E15
Ubuntu@T3-2020630347-1:~/Tarea3$ |
```

```

Ubuntu@T3-2020630347-2:~/Tarea3$ java multiMatriz 2 3000 2000
Matrices recibidas
= A[1]
Realizando cálculos para obtener submatrices...
= A[1]
Matrices enviadas
Ubuntu@T3-2020630347-2:~/Tarea3$ |
```

```

Ubuntu@T3-2020630347-3:~/Tarea3$ java multiMatriz 3 3000 2000
Matrices recibidas
part
[ Node
  = B
] = B
] = B
```

Conclusión

Durante el desarrollo de esta práctica, pude abordar de nueva cuenta varios conocimientos vistos durante el curso en la implementación de este programa en Java; logrado diseñar una solución eficiente y escalable para el cálculo del producto de matrices rectangulares, dividiendo las tareas entre tres nodos virtuales en Azure.

La implementación realizada mediante sockets entre los nodos me pudo hacer darme cuenta sobre la importancia de la planificación y la coordinación en sistemas distribuidos. Además, pude adquirir experiencia en la transmisión y recepción de datos entre nodos y en la manipulación de matrices para realizar cálculos complejos.

De igual manera pude darme cuenta de que con la verificación del resultado mediante el cálculo del checksum de la matriz C se logra resaltar la necesidad de garantizar la integridad de los datos en entornos distribuidos.

Durante esta práctica pude ampliar y reforzar mis conocimientos en programación distribuida y poder aplicar conceptos teóricos en un entorno práctico. Dándome cuenta que la capacidad de dividir tareas y coordinar la ejecución entre múltiples nodos es una habilidad valiosa en el mundo de la informática distribuida.