



Instituto Politécnico Nacional

Escuela Superior de Cómputo

Sistemas Distribuidos

Pineda Guerrero Carlos

Tarea 2

4CV13

Humberto Alejandro Ortega Alcocer (2016630495)

24 de Marzo del 2023

Índice

Escuela Superior de Cómputo	0
Sistemas Distribuidos	0
Pineda Guerrero Carlos	0
Índice	1
Objetivo	2
Desarrollo	5
Creación de Máquinas Virtuales	5
Creación de Código (ChatGPT)	15
Conversación con ChatGPT	15
Ejecución del Programa	23
Conclusión	25

Objetivo

En esta tarea se deberá desarrollar un solo programa en Java el cual calculará el producto de dos matrices cuadradas en forma distribuida utilizando tres máquinas virtuales de Azure.

Se deberá escribir el programa en su totalidad utilizando ChatGPT, o bien, se podrá escribir partes del programa utilizando ChatGPT y después integrar las partes manualmente.

Sean A, B y C matrices cuadradas con elementos de tipo double, la dimensión de las matrices es N. C=AxB

Se deberá ejecutar dos casos:

1. N=12, desplegar las matrices A, B y C y el checksum de la matriz C.
2. N=3000, desplegar el checksum de la matriz C.

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:

$$\begin{aligned} A[i][j] &= 2*i+j \\ B[i][j] &= 3*i-j \end{aligned}$$

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

El programa deberá ejecutar en una computadora local (nodo 0) con Windows, Linux o MacOS y en tres máquinas virtuales con Ubuntu (nodos 1, 2 y 3).

El nombre de cada máquina virtual deberá ser: "T2-" concatenando el número de boleta del alumno o alumna, un guión y el número de nodo, por ejemplo, si el número de boleta es 12345678, entonces el nodo 1 deberá llamarse: T2-12345678-1, el nodo 2 deberá llamarse T2-12345678-2, y el nodo 3 deberá llamarse T2-12345678-3. No se admitirá la tarea si los nodos no se nombran como se indicó anteriormente.

Recuerden que deben eliminar las máquinas virtuales cuando no las usen, con la finalidad de ahorrar el saldo de sus cuentas de Azure.

¿Cómo realizar la multiplicación de matrices en forma distribuida?

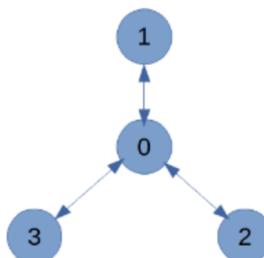
Sea BT la transpuesta de la matriz B. Se deberá dividir las matrices A y BT en tres partes de igual tamaño, por tanto la matriz C estará dividida en 9 partes tal como se muestra en la siguiente figura:

$$\begin{array}{|c|} \hline A1 \\ \hline A2 \\ \hline A3 \\ \hline \end{array} \times \begin{array}{|c|} \hline B^T1 \\ \hline B^T2 \\ \hline B^T3 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline C1 & C2 & C3 \\ \hline C4 & C5 & C6 \\ \hline C7 & C8 & C9 \\ \hline \end{array}$$

Debido a que las matrices se guardan por renglones en la memoria, para aprovechar la localidad espacial el producto se realizará renglón por renglón tal como vimos en clase. Entonces las matrices Ci se obtienen de la siguiente manera:

$$\begin{aligned} C1 &= A1 \times BT1 \\ C2 &= A1 \times BT2 \\ C3 &= A1 \times BT3 \\ C4 &= A2 \times BT1 \\ C5 &= A2 \times BT2 \\ C6 &= A2 \times BT3 \\ C7 &= A3 \times BT1 \\ C8 &= A3 \times BT2 \\ C9 &= A3 \times BT3 \end{aligned}$$

Considere la siguiente topología:



El nodo 0 será la computadora local. El nodo 0 inicializará las matrices A y B, obtendrá la transpuesta de la matriz B y enviará las matrices Ai y BTj a los nodos 1, 2 y 3.

El nodo 1 calculará los productos C1, C2 y C3; el nodo 2 calculará los productos C4, C5 y C6; el nodo 3 calculará los productos C7, C8 y C9.

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

Se deberá implementar los siguientes algoritmos:

Nodo 0:

1. Inicializar las matrices A y B.
2. Transponer la matriz B. Dejar la transpuesta en la misma matriz B.
3. Enviar la matriz A1 al nodo 1.

4. Enviará las matrices BT1, BT2 y BT3 al nodo 1.
5. Enviará la matriz A2 al nodo 2.
6. Enviará las matrices BT1, BT2 y BT3 al nodo 2.
7. Enviará la matriz A3 al nodo 3.
8. Enviará las matrices BT1, BT2 y BT3 al nodo 3.
9. Recibirá las matrices C1, C2 y C3 del nodo 1.
10. Recibirá las matrices C4, C5 y C6 del nodo 2.
11. Recibirá las matrices C7, C8 y C9 del nodo 3.
12. Calcular el checksum de la matriz C.
13. Desplegar el checksum de la matriz C.
14. Si N=12 entonces desplegar las matrices A, B y C.

Nodo 1:

1. Recibirá la matriz A1 del nodo 0.
2. Recibirá las matrices BT1, BT2 y BT3 del nodo 0.
3. Calcular las matrices C1, C2 y C3.
4. Enviará las matrices C1, C2 y C3 al nodo 0.

Nodo 2:

1. Recibirá la matriz A2 del nodo 0.
2. Recibirá las matrices BT1, BT2 y BT3 del nodo 0.
3. Calcular las matrices C4, C5 y C6.
4. Enviará las matrices C4, C5 y C6 al nodo 0.

Nodo 3:

1. Recibirá la matriz A3 del nodo 0.
2. Recibirá las matrices BT1, BT2 y BT3 del nodo 0.
3. Calcular las matrices C7, C8 y C9.
4. Enviará las matrices C7, C8 y C9 al nodo 0.

Desarrollo

Para desarrollar esta tarea, se descompondrá en tres etapas: creación de máquinas virtuales, creación de código (usando ChatGPT) y pruebas de ejecución. Esto con el fin de profundizar en cada sección de forma adecuada.

Creación de Máquinas Virtuales

Para esta materia, estamos utilizando la plataforma de nube de Microsoft: Azure. Para poderse registrar y utilizar esta plataforma, lo primero será darse de alta con el correo institucional y reclamar los \$100 USD de regalo para comenzar a utilizar Azure. A continuación se muestra la pantalla principal mostrando los créditos abonados a la cuenta:

The screenshot shows the Microsoft Azure portal's 'Education | Overview' page. On the left, there's a sidebar with 'Overview' selected under 'Learning resources'. The main area has tabs for 'Get started' and 'Overview'. Under 'Student offer details', it shows 'Available credits: \$100 out of \$100' and 'Days until credits expire: 366'. Below that, 'March costs' are listed as '\$0.00'. There are links to 'View cost details' and 'Explore all'. To the right, there are sections for 'Popular solutions' (Deploy a Docker container, Create your first Node.js app, Create and train a Machine Learning model, Build and deploy your first website) and 'Free Services' (Azure Virtual Machines – Windows, Azure Blob Storage, Computer Vision, Azure App Service). At the bottom, there are sections for 'Free software' (SQL Server 2019 Developer, Machine Learning Server 9.4.7 for Windows, Microsoft R Client 9.4.7, Agents for Visual Studio 2019 (version 16.0)..., Agents for Visual Studio 2019 (version 16.0)...), 'Free learning paths' (Data Scientist, AI Engineer, Developer, DataEngineer), and 'Resources' (Get started guide for Azure developers, Pricing calculator, Optimize your cloud investment with cost ..., Explore student hub).

Dado que debemos crear *Máquinas Virtuales*, nos dirigimos al menú lateral y seleccionamos la opción de crear recurso nuevo:

The screenshot shows the Azure portal's "Overview" page. On the left, a sidebar lists various service categories like All services, Favorites, and Storage accounts. The main area displays "Student offer details" with information such as Available credits (\$100 out of \$100), Days until credits expire (366, expires on 03/24/2024), and March costs (\$0.00). To the right, a section titled "Popular solutions" lists options like Deploy a Docker container, Create your first Node.js app, Create and train a Machine Learning model, and Build and deploy your first website.

Ahora seleccionaremos la opción que dice: “Virtual Machine”, de entre las opciones que nos muestra la plataforma:

The screenshot shows the "Create a resource" page in the Azure portal. The left sidebar has "Get Started" selected. The main content area shows "Popular Azure services" and "Popular Marketplace products". Under "Popular Azure services", the "Virtual machine" option is highlighted with a blue border. Other services listed include Web App, SQL Database, Function App, Key Vault, Data Factory, Template deployment, Logic App, and Automation. Under "Popular Marketplace products", options like Windows Server 2019 Datacenter, Windows 10 Pro, Windows 11 Pro, Ubuntu Server, and Red Hat Enterprise Linux are shown.

En la siguiente pantalla lo primero será verificar que la suscripción que se está utilizando sea la de *Azure for Students* y, posteriormente, crearemos un *Resource Group* con el nombre *SistemasDistribuidosResourceGroup* dónde colocaremos todos los recursos asociados con la materia durante el semestre.

This subscription may not be eligible to deploy VMs of certain sizes in certain regions.

Project details
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Azure for Students

Resource group * (New) Resource group Create new

Instance details

Virtual machine name * SistemasDistribuidosVM

Region * East US

Availability options

Security type

Image * Ubuntu Server 20.04 LTS - x64 Gen2

VM architecture Arm64 x64

Review + create < Previous Next : Disks > Give feedback

Seleccionamos la región en el este de los Estados Unidos, colocamos el nombre de la máquina virtual siguiendo las especificaciones de la tarea, seleccionamos una máquina con 1vCPU y 1GB de RAM, en arquitectura x86 de 64 bits. Estoy utilizando Ubuntu 20.04 en vez de Ubuntu 18.04 porque, para fines de la tarea, es indistinta la versión de la distribución que se quiera utilizar pues no caemos en las deprecaciones de servicios ni paquetes que se requieren.

Create a virtual machine

Subscription * ⓘ Azure for Students

Resource group * ⓘ (New) SistemasDistribuidosResourceGroup Create new

Instance details

Virtual machine name * ⓘ T2-2016630495-1

Region * ⓘ (US) East US

Availability options ⓘ No infrastructure redundancy required

Security type ⓘ Trusted launch virtual machines Configure security features

Image * ⓘ Ubuntu Server 20.04 LTS - x64 Gen2 See all images | Configure VM generation

VM architecture ⓘ x64 Arm64

Run with Azure Spot discount ⓘ

Size * ⓘ Standard_B1s - 1 vcpu, 1 GiB memory (\$7.59/month) See all sizes

Para poder acceder a la máquina tendremos que tener autenticación por SSH, para esto crearemos nuevas claves con el usuario humbertoody (lo uso en otros lados y me simplifica la gestión de llaves en mi computadora local) y el nombre del nuevo par de llaves como humbertoody-azure, de la siguiente forma:

Administrator account

Authentication type: SSH public key Password

Azure now automatically generates an SSH key pair for you and allows you to store it for future use. It is a fast, simple, and secure way to connect to your virtual machine.

Username *: humbertowoody

SSH public key source: Generate new key pair

Key pair name *: humbertowood-azure

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports *: None Allow selected ports

Select inbound ports *: SSH (22)

This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

Review + create **< Previous** **Next : Disks >** **Give feedback**

Ahora asignamos la configuración de Discos, la cual dejaremos en su configuración estándar:

Basics **Disks** Networking Management Monitoring Advanced Tags Review + create

Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. [Learn more](#)

VM disk encryption

Azure disk storage encryption automatically encrypts your data stored on Azure managed disks (OS and data disks) at rest by default when persisting it to the cloud.

Encryption at host:

Encryption at host is not registered for the selected subscription. [Learn more about enabling this feature](#)

OS disk

OS disk size *: Default size (30 GiB)

OS disk type *: Premium SSD (locally-redundant storage)

Delete with VM:

Key management: Platform-managed key

Enable Ultra Disk compatibility:
Ultra disk is not supported with selected security type.

Data disks for T2-2016630495-1

You can add and configure additional data disks for your virtual machine or attach existing disks. This VM also comes with a temporary disk.

LUN	Name	Size (GiB)	Disk type	Host caching	Delete with VM

Review + create **< Previous** **Next : Networking >** **Give feedback**

La configuración de Red, Administración y demás se dejará con los valores por defecto, siendo el resumen de nuestra nueva máquina virtual el siguiente:

Validation passed

Basics Disks Networking Management Monitoring Advanced Tags **Review + create**

Price
1 X Standard B1s by Microsoft **0.0104 USD/hr** Subscription credits apply
[Terms of use](#) | [Privacy policy](#)

TERMS
By clicking "Create", 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: HUMBERTO ALEJANDRO ORTEGA ALCOCER
Preferred e-mail address: hortegaa1500@alumno.ipn.mx
Preferred phone number: (+52) 55 57296166

Create < Previous Next > Download a template for automation Give feedback

Validation passed

Basics

Subscription	Azure for Students
Resource group	(new) SistemasDistribuidosResourceGroup
Virtual machine name	T2-2016630495-1
Region	East US
Availability options	No infrastructure redundancy required
Security type	Trusted launch virtual machines
Enable secure boot	Yes
Enable vTPM	Yes
Integrity monitoring	Yes
Image	Ubuntu Server 20.04 LTS - Gen2
VM architecture	x64
Size	Standard B1s (1 vcpu, 1 GiB memory)
Authentication type	SSH public key
Username	humbertowoody
Key pair name	humbertowoody-azure
Public inbound ports	SSH
Azure Spot	No

Create < Previous Next > Download a template for automation Give feedback

Validation passed

Disks

OS disk size	Default size (30 GiB)
OS disk type	Premium SSD LRS
Use managed disks	Yes
Delete OS disk with VM	Enabled
Ephemeral OS disk	No

Networking

Virtual network	(new) T2-2016630495-1-vnet
Subnet	(new) default (10.0.0.0/24)
Public IP	(new) T2-2016630495-1-ip
Accelerated networking	Off
Place this virtual machine behind an existing load balancing solution?	No
Delete public IP and NIC when VM is deleted	Disabled

Management

Microsoft Defender for Cloud	Basic (free)
System assigned managed identity	Off
Login with Azure AD	Off
Auto-shutdown	Off
Backup	Disabled

Create < Previous Next > Download a template for automation Give feedback

Validation passed

Management

Microsoft Defender for Cloud	Basic (free)
System assigned managed identity	Off
Login with Azure AD	Off
Auto-shutdown	Off
Backup	Disabled
Enable hotpatch	Off
Patch orchestration options	Image Default

Monitoring

Alerts	Off
Boot diagnostics	On
Enable OS guest diagnostics	Off

Advanced

Extensions	None
VM applications	None
Cloud init	No
User data	No
Disk controller type	SCSI
Proximity placement group	None
Capacity reservation group	None

Create < Previous Next > Download a template for automation Give feedback

Al hacer click en siguiente, Azure descargará la nueva llave de SSH para la máquina virtual a nuestro equipo local y comenzará la creación de la máquina virtual de la siguiente manera:

The screenshot shows the Microsoft Azure portal interface. At the top, the URL is portal.azure.com/?Microsoft_Azure_Education_correlationid=165.... The page title is "CreateVm-canonical.0001-com-ubuntu-server-focal-2-20230324144352 | Overview". The main content area shows a deployment status message: "Deployment is in progress". Below this, there's a table titled "Deployment details" with columns: Resource, Type, Status, and Operation details. A note says "No results.". On the left, there's a navigation sidebar with "Overview", "Inputs", "Outputs", and "Template". On the right, there's a sidebar with sections for "Microsoft Defender for Cloud", "Free Microsoft tutorials", and "Work with an expert". At the bottom, there's a "Show All" button.

Una vez finalizado el despliegue, tendremos la siguiente confirmación:

The screenshot shows the Microsoft Azure portal interface. The URL is portal.azure.com/?Microsoft_Azure_Education_correlationid=165.... The page title is "CreateVm-canonical.0001-com-ubuntu-server-focal-2-20230324144352 | Overview". The main content area shows a deployment status message: "Your deployment is complete". Below this, there's a table titled "Deployment details" with columns: Resource, Type, Status, and Operation details. A note says "No results.". On the left, there's a navigation sidebar with "Overview", "Inputs", "Outputs", and "Template". On the right, there's a "Notifications" sidebar with a message: "Deployment succeeded" and "Deployment 'CreateVm-canonical.0001-com-ubuntu-server-focal-2-20230324144352' to resource group 'SistemasDistribuidosResourceGroup' was successful." There are "Go to resource" and "Pin to dashboard" buttons. At the bottom, there's a "Show All" button.

Así, el recurso ya creado se visualiza de la siguiente manera:

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with icons for search, refresh, and account information. Below it, the main header says "Microsoft Azure" and "Search resources, services, and docs (G+)".

The main content area displays the properties of a virtual machine named "T2-2016630495-1". On the left, there's a sidebar with various tabs like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Networking, Connect, Disks, Size, Microsoft Defender for Cloud, Advisor recommendations, Extensions + applications, Continuous delivery, Availability + scaling, Configuration, Identity, and Properties. The "Properties" tab is currently selected.

The main panel is divided into sections: "Essentials", "Virtual machine", "Networking", and "Size".

Essentials:

Resource group	(move) : SistemasDistribuidosResourceGroup	Operating system	: Linux (ubuntu 20.04)
Status	: Running	Size	: Standard B1s (1 vcpu, 1 GiB memory)
Location	: East US	Public IP address	: 74.235.12.118
Subscription	(move) : Azure for Students	Virtual network/subnet	: T2-2016630495-1-vnet/default
Subscription ID	: a5d9d675-3ae1-4e2b-95b5-02150d67b8d2	DNS name	: Not configured
Tags (edit)	: Click here to add tags		

Virtual machine:

Computer name	T2-2016630495-1
Health state	-
Operating system	Linux (ubuntu 20.04)
Publisher	canonical
Offer	0001-com-ubuntu-server-focal
Plan	20_04-lts-gen2
VM generation	V2
VM architecture	x64
Agent status	Ready
Agent version	2.9.0.4

Networking:

Public IP address	74.235.12.118 (Network interface t2-2016630495-194)
Public IP address (IPv6)	-
Private IP address	10.0.0.4
Private IP address (IPv6)	-
Virtual network/subnet	T2-2016630495-1-vnet/default
DNS name	Configure

Size:

Size	Standard B1s
vCPUs	1

Lo siguiente será establecer conexión con la máquina virtual, para esto, lo primero será cambiar los permisos del archivo `*.pem` que acabamos de descargar y posteriormente realizar la conexión por SSH:

```
> ls -lha
Permissions Size User          Date Modified Name
.rw-r--r--@ 2.5k humbertowoody 24 Mar 15:19  humbertowoody-azure.pem
> chmod 600 humbertowoody-azure.pem
> ls -lha
Permissions Size User          Date Modified Name
.rw-----@ 2.5k humbertowoody 24 Mar 15:19  humbertowoody-azure.pem
> ssh -i humbertowoody-azure.pem humbertowoody@74.235.12.118
The authenticity of host '74.235.12.118 (74.235.12.118)' can't be established.
ED25519 key fingerprint is SHA256:uHeRdLR0hcYcyJd0LhPBjS1buQrlnYJeF145ycltlf0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '74.235.12.118' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1034-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information as of Fri Mar 24 21:24:27 UTC 2023

 System load:  0.11           Processes:      109
 Usage of /:   5.2% of 28.89GB  Users logged in:    0
 Memory usage: 33%           IPv4 address for eth0: 10.0.0.4
 Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

humbertowoody@T2-2016630495-1:~$
```

Y ¡listo! 🎉

Durante la práctica se necesita determinar los puertos requeridos para la ejecución de los servidores, los puertos que usaremos serán:

- Nodo 0: usará el puerto 7999 en mi computadora local.
- Nodo 1: usará el puerto 8080 en T2-2016630495-1.
- Nodo 2: usará el puerto 8081 en T2-2016630495-2.
- Nodo 3: usará el puerto 8082 en T2-2016630495-3.

Así, crearemos una nueva política de Puerto Entrante que incluya el rango 8080-8082 bajo el protocolo TCP y la colocaremos a cada máquina virtual de la siguiente manera:

The screenshot shows the Azure portal interface for managing a virtual machine named 'T2-2016630495-1'. On the left, the 'Networking' section is selected under 'Settings'. In the main pane, the 'Inbound port rules' table is displayed, showing five existing rules. A new rule is being configured in a modal dialog titled 'Add inbound security rule'. The 'Source' is set to 'Any', 'Destination' is 'Any', 'Service' is 'Custom', and 'Protocol' is 'TCP'. The 'Destination port ranges' field contains '8080-8082'. The 'Action' is set to 'Allow'. The 'Priority' is set to 310. At the bottom of the dialog are 'Add' and 'Cancel' buttons, and a 'Give feedback' link.

Finalmente, repetiremos estos pasos para las máquinas virtuales correspondientes a los nodos dos y tres. Así, las tres máquinas funcionando se ven de la siguiente manera en la página de Azure:

Showing 1 to 3 of 3 records.

Name	Type	Subscription	Resource group	Location	Status	Operating system	Size	Public IP address
T2-2016630495-1	Virtual machine	Azure for Students	SistemasDistribuidosR...	East US	Running	Linux	Standard_B1s	74.235.12.118
T2-2016630495-2	Virtual machine	Azure for Students	SistemasDistribuidosR...	East US	Running	Linux	Standard_B1s	20.25.1.241
T2-2016630495-3	Virtual machine	Azure for Students	SistemasDistribuidosR...	East US	Running	Linux	Standard_B1s	172.173.139.232

< Previous Page 1 of 1 Next >

Give feedback

Y, estableciendo la conexión con SSH a cada uno de los nodos:

```
~/Proyectos/IPN/sistemas-distribuidos-escom/tareas/tarea-2 main* https://ubuntu.com/azure/pro
>
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Mar 24 21:24:28 2023 from 148.204.56.40
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

humberto@T2-2016630495-1:~$
```

```
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

humberto@T2-2016630495-2:~$
```

```
humberto@T2-2016630495-2:~$ ESCOM 1 general 2 distribuidos+ ALCOCER 53% Fri 24/03 09:39 PM CST
```

Creación de Código (ChatGPT)

Uno de los principales retos de esta tarea es utilizar ChatGPT para generar o ayudarnos para generar el código correspondiente a la tarea.

El código final se anexa directamente en la entrega de la tarea por lo que no se hará un recorrido sobre el mismo, sin embargo en la sección de *Ejecución del Programa* se podrán ver detalles y datos de funcionamiento finales.

Conversación con ChatGPT

A continuación se anexa la conversación sostenida con ChatGPT con el fin de entender, generar y mejorar el código para la tarea actual.

Lo primero fue pedir un resumen de la tarea, así como detalles que nos ayuden a ver qué tanto puede comprender el modelo a partir

The screenshot shows a dark-themed ChatGPT interface. At the top, it says "Model: GPT-4". The conversation starts with a user message (HU) asking for a summary and specific steps to calculate the product of two square matrices using three Azure virtual machines. The AI responds by asking for more information about the matrices and the desired output checksum. It then provides instructions for the Java program, mentioning matrix dimensions N and the formula C = AxB. The AI also asks to execute two cases and provides a checksum formula. The interface includes standard ChatGPT controls like a send button and a regenerate response button.

Hola! Porfavor, quiero un resumen y las partes importantes para poder realizar la siguiente tarea, quisiera que me dieras los pasos de forma granular y específica, así como los algoritmos involucrados de forma corta y concisa, quiero que me expliques todo a manera de que sea lo más claro posible

¡Hola! Me encantaría ayudarte, pero necesito más información sobre la tarea o el tema que deseas que explique. Por favor, proporcioname más detalles para poder darte una respuesta adecuada y clara.

En esta tarea se deberá desarrollar un solo programa en Java el cual calculará el producto de dos matrices cuadradas en forma distribuida utilizando tres máquinas virtuales de Azure.

Se deberá escribir el programa en su totalidad utilizando ChatGPT, o bien, se podrá escribir partes del programa utilizando ChatGPT y después integrar las partes manualmente.

Sean A, B y C matrices cuadradas con elementos de tipo double, la dimensión de las matrices es N. C=AxB

Se deberá ejecutar dos casos:

N=12, desplegar las matrices A, B y C y el checksum de la matriz C.
N=3000, desplegar el checksum | Regenerate response

El checksum de la matriz C se calculará como la suma de todos elementos de la matriz. El

Una vez proporcionada la tarea, el bot nos responde con la lista de pasos a realizar para llevar a cabo la tarea, a partir de aquí podemos entender si el modelo está comprendiendo correctamente lo que queremos lograr, en este caso así es.

Valor de la tarea: 25% (2 puntos de la primera evaluación parcial)

Para completar esta tarea, primero se debe dividir el problema en componentes más pequeños. Te proporcionaré un resumen de los pasos y algoritmos a implementar.

1. Crear las máquinas virtuales en Azure con Ubuntu y nombrarlas correctamente según las instrucciones.
2. Configurar la comunicación entre el nodo 0 (computadora local) y los nodos 1, 2 y 3 en Azure. Puedes usar sockets para comunicarte entre los nodos. Investiga cómo crear y utilizar sockets en Java.
3. Crear las funciones para inicializar las matrices A y B, y para transponer la matriz B.
4. Crear las funciones para dividir las matrices A y BT en partes iguales y calcular las matrices C.
5. Implementar los algoritmos de Nodo 0, Nodo 1, Nodo 2 y Nodo 3 descritos en el enunciado, utilizando las funciones creadas en los pasos 3 y 4.
6. Ejecutar el programa con N=12 y N=3000. Asegurarse de que los checksums son correctos.
7. Documentar el proceso en un archivo PDF con capturas de pantalla y descripciones, incluyendo la creación de las máquinas virtuales, la comunicación entre nodos, la compilación y la ejecución del programa.

Recuerda que como ChatGPT, puedo ayudarte a desarrollar partes del programa o responder preguntas específicas que puedas tener durante el proceso.

[Regenerate response](#)

De aquí en adelante, se le pide al modelo que genere el código en Java que solucione la problemática planteada por la tarea, dado que GPT-4 es consciente de su límite de token, nos ofrece generar un “machote” de la estructura general requerida para solucionar la tarea, de la siguiente forma:

Recuerda que como ChatGPT, puedo ayudarte a desarrollar partes del programa o responder preguntas específicas que puedas tener durante el proceso.

HU Podrías generar únicamente el código en Java que satisface los requisitos planteados en la tarea? me gustaría que fuera el código más simple y menos complejo que puedes generar

Entiendo que estás buscando un código simple y menos complejo. A continuación, te proporciono un esquema básico de las clases y métodos que podrías implementar en Java. Este esquema no es un código completo y funcional, pero te servirá como guía para comenzar a desarrollar la solución.

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

public class MatrixMultiplication {

    public static void main(String[] args) {
        int node = Integer.parseInt(args[0]);

        if (node == 0) {
            nodo0();
        } else if (node == 1) {
            nodo1();
        } else if (node == 2) {
            nodo2();
        } else if (node == 3) {
            nodo3();
        }
    }

    private static void nodo0() {
        // Implementación para el nodo 0
    }

    private static void nodo1() {
        // Implementación para el nodo 1
    }

    private static void nodo2() {
        // Implementación para el nodo 2
    }

    private static void nodo3() {
        // Implementación para el nodo 3
    }
}
```

[Regenerate response](#)

```
nodo1();
} else if (node == 2) {
    nodo2();
} else if (node == 3) {
    nodo3();
}

public static void nodo0 {
    // Inicializar matrices A y B
    // Transponer matriz B
    // Dividir matrices A y BT en partes iguales
    // Enviar y recibir datos entre nodos utilizando sockets
    // Calcular checksum de matriz C
    // Mostrar resultados
}

public static void nodo1 {
    // Recibir matrices A1, BT1, BT2, BT3 de nodo 0 utilizando sockets
    // Calcular matrices C1, C2, C3
    // Enviar matrices C1, C2, C3 al nodo 0 utilizando sockets
}

public static void nodo2 {
    // Recibir matrices A2, BT1, BT2, BT3 de nodo 0 utilizando sockets
    // Calcular matrices C4, C5, C6
    // Enviar matrices C4, C5, C6 al nodo 0 utilizando sockets
}

public static void nodo3 {
    // Recibir matrices A3, BT1, BT2, BT3 de nodo 0 utilizando sockets
    // Calcular matrices C7, C8, C9
    // Enviar matrices C7, C8, C9 al nodo 0 utilizando sockets
}

// Métodos adicionales para operaciones con matrices (inicialización, tr
}



Regenerate response


```

Ahora trataremos de que el bot nos proporcione las definiciones de las funciones más importantes para la realización de la tarea.

Este esquema te proporciona una estructura básica para comenzar a implementar la solución en Java. Deberás completar los métodos correspondientes a cada nodo y agregar métodos adicionales para las operaciones con matrices (inicialización, transposición, división y multiplicación). Además, deberás investigar cómo utilizar sockets en Java para comunicarte entre nodos.

HU: no entiendo particularmente lo que se trata de lograr o cuál es la diferencia con la metodología estándar para multiplicación de matrices vs. lo que se propone en la tarea, podrías explicarme por qué es importante y una implementación simple en Java?

AI: La metodología estándar para la multiplicación de matrices implica calcular el producto de dos matrices de manera secuencial en una sola máquina o computadora. En la tarea que te han asignado, se te pide realizar la multiplicación de matrices de manera distribuida utilizando múltiples máquinas (nodos) para dividir y paralelizar la carga de trabajo. El objetivo es mejorar el rendimiento y la eficiencia en el cálculo de la multiplicación de matrices grandes.

La diferencia principal en la metodología propuesta en la tarea es que se divide la matriz A y la transpuesta de la matriz B (BT) en tres partes iguales, y cada nodo calcula una parte de la matriz C. Al dividir el cálculo en partes más pequeñas y distribuirlo entre varios nodos, puedes aprovechar la potencia de cómputo de múltiples máquinas y completar el proceso de manera más eficiente.

Aquí tienes una implementación sencilla para que puedas compararla con la metodología estándar (secuencial):

[Regenerate response](#)

nodos, puedes aprovechar la potencia de cómputo de múltiples máquinas y completar el proceso de manera más eficiente.

Aquí tienes una implementación simple en Java que ilustra cómo realizar la multiplicación de matrices utilizando la metodología estándar (secuencial) para que puedas compararla con la metodología distribuida propuesta en la tarea:

```
java
public class MatrixMultiplicationSequential {
    public static void main(String[] args) {
        int N = 12;

        double[][] A = initMatrixA(N);
        double[][] B = initMatrixB(N);
        double[][] C = multiplyMatrices(A, B, N);

        // Calcular el checksum y mostrar resultados
    }

    public static double[][] initMatrixA(int N) {
        double[][] A = new double[N][N];
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                A[i][j] = 2 * i + j;
            }
        }
        return A;
    }

    public static double[][] initMatrixB(int N) {
        double[][] B = new double[N][N];
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                B[i][j] = 3 * i - j;
            }
        }
        return B;
    }

    public static double[][] multiplyMatrices(double[][] A, double[][] B, int N) {
        double[][] C = new double[N][N];
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                for (int k = 0; k < N; k++) {
                    C[i][j] += A[i][k] * B[k][j];
                }
            }
        }
        return C;
    }
}
```

Esta implementación simple calcula el producto de las matrices A y B de manera secuencial en una sola máquina. Para realizar la multiplicación de matrices de manera distribuida, como se describe en la tarea, deberás implementar la comunicación entre múltiples máquinas (nodos) y dividir las matrices A y B en partes iguales para distribuir el cálculo entre los nodos.

HU Podrías generar únicamente el código en Java que realizaría el cálculo final en la versión de la tarea?

💡 En la versión de la tarea, el cálculo final se realiza en el nodo 0, donde se suman todas las submatrices C para formar la matriz C completa y luego se calcula el checksum. Aquí tienes un código simple en Java que muestra cómo realizar el cálculo final en el nodo 0, asumiendo que ya se han recibido las submatrices C del resto de los nodos:

```
java
public class MatrixCalculation {
    public static void main(String[] args) {
        // Implementación del cálculo final en el nodo 0
        // Aquí se suman las submatrices recibidas y se calcula el checksum
    }
}
```

En la versión de la tarea, el cálculo final se realiza en el nodo 0, donde se suman todas las submatrices C para formar la matriz C completa y luego se calcula el checksum. Aquí tienes un código simple en Java que muestra cómo realizar el cálculo final en el nodo 0, asumiendo que ya se han recibido las submatrices C del resto de los nodos:

```
java
Copy code

public class MatrixCalculationFinal {
    public static void main(String[] args) {
        int N = 12;
        double[][] submatricesC = new double[3][N / 3][N];

        // Suponer que las submatrices C ya han sido recibidas de los nodos
        // y están almacenadas en el arreglo 'submatricesC'

        double[][] C = combineSubmatrices(submatricesC, N);
        double checksum = calculateChecksum(C);

        // Mostrar el checksum y, si N = 12, las matrices A, B y C
    }

    public static double[][] combineSubmatrices(double[][][] submatricesC, int N) {
        double[][] C = new double[N][N];

        for (int n = 0; n < 9; n++) {
            int rowOffset = (n / 3) * (N / 3);
            int colOffset = (n % 3) * (N / 3);

            for (int i = 0; i < N / 3; i++) {
                for (int j = 0; j < N / 3; j++) {
                    C[rowOffset + i][colOffset + j] = submatricesC[n][i][j];
                }
            }
        }

        return C;
    }

    public static double calculateChecksum(double[][] C) {
        double checksum = 0.0;

        for (int i = 0; i < C.length; i++) {
            for (int j = 0; j < C[i].length; j++) {
                checksum += C[i][j];
            }
        }

        return checksum;
    }
}
```

Regenerate response

```
        C[rowOffset + i][colOffset + j] = submatricesC[n][i][j];
    }
}
return C;
}

public static double calculateChecksum(double[][] C) {
    double checksum = 0.0;

    for (int i = 0; i < C.length; i++) {
        for (int j = 0; j < C[i].length; j++) {
            checksum += C[i][j];
        }
    }

    return checksum;
}
}
```

Este código asume que las submatrices C han sido recibidas de los nodos 1, 2 y 3 y están almacenadas en el arreglo `submatricesC`. La función `combineSubmatrices()` combina estas submatrices en la matriz C completa, y la función `calculateChecksum()` calcula el checksum de la matriz C.

H0 Gracias, ahora me podrías generar el código necesario para el cálculo de cada submatriz C en alguno de los otros nodos? asume que tienes los datos necesarios

Claro, aquí tienes el código en Java para calcular las submatrices C en uno de los nodos (1, 2 o 3) asumiendo que ya se han recibido las matrices A y las matrices BT transpuestas del nodo 0:

```
java
Copy code

public class MatrixCalculationFinal {
    public static void main(String[] args) {
        int N = 12;
        double[][] submatricesC = new double[3][N / 3][N];

        // Suponer que las submatrices C ya han sido recibidas de los nodos
        // y están almacenadas en el arreglo 'submatricesC'

        double[][] C = combineSubmatrices(submatricesC, N);
        double checksum = calculateChecksum(C);

        // Mostrar el checksum y, si N = 12, las matrices A, B y C
    }

    public static double[][] combineSubmatrices(double[][][] submatricesC, int N) {
        double[][] C = new double[N][N];

        for (int n = 0; n < 9; n++) {
            int rowOffset = (n / 3) * (N / 3);
            int colOffset = (n % 3) * (N / 3);

            for (int i = 0; i < N / 3; i++) {
                for (int j = 0; j < N / 3; j++) {
                    C[rowOffset + i][colOffset + j] = submatricesC[n][i][j];
                }
            }
        }

        return C;
    }

    public static double calculateChecksum(double[][] C) {
        double checksum = 0.0;

        for (int i = 0; i < C.length; i++) {
            for (int j = 0; j < C[i].length; j++) {
                checksum += C[i][j];
            }
        }

        return checksum;
    }
}
```

Regenerate response

Claro, aquí tienes el código en Java para calcular las submatrices C en uno de los nodos (1, 2 o 3) asumiendo que ya se han recibido las matrices A y las matrices BT transpuestas del nodo 0:

```
java
public class SubmatrixCalculationNode {
    public static void main(String[] args) {
        int N = 12;

        // Asumir que las matrices A y BT ya han sido recibidas del nodo 0
        double[][] A = new double[N / 3][N];
        double[][] BT = new double[3][N / 3][N];

        double[][] submatricesC = calculateSubmatricesC(A, BT, N);

        // Enviar las submatrices C calculadas al nodo 0
    }

    public static double[][] calculateSubmatricesC(double[][] A, double[][] BT) {
        double[][] submatricesC = new double[3][N / 3][N];

        for (int n = 0; n < 3; n++) {
            for (int i = 0; i < N / 3; i++) {
                for (int j = 0; j < N / 3; j++) {
                    for (int k = 0; k < N; k++) {
                        submatricesC[n][i][j] += A[i][k] * BT[n][j][k];
                    }
                }
            }
        }

        return submatricesC;
    }
}
```

Este código asume que las matrices A y BT han sido recibidas del nodo 0. La función `calculateSubmatricesC` calcula las submatrices C correspondientes para el nodo actual, utilizando la matriz A y las matrices BT transpuestas recibidas. Después de calcular las submatrices C, deberás implementar la comunicación para enviar estas submatrices al nodo 0.

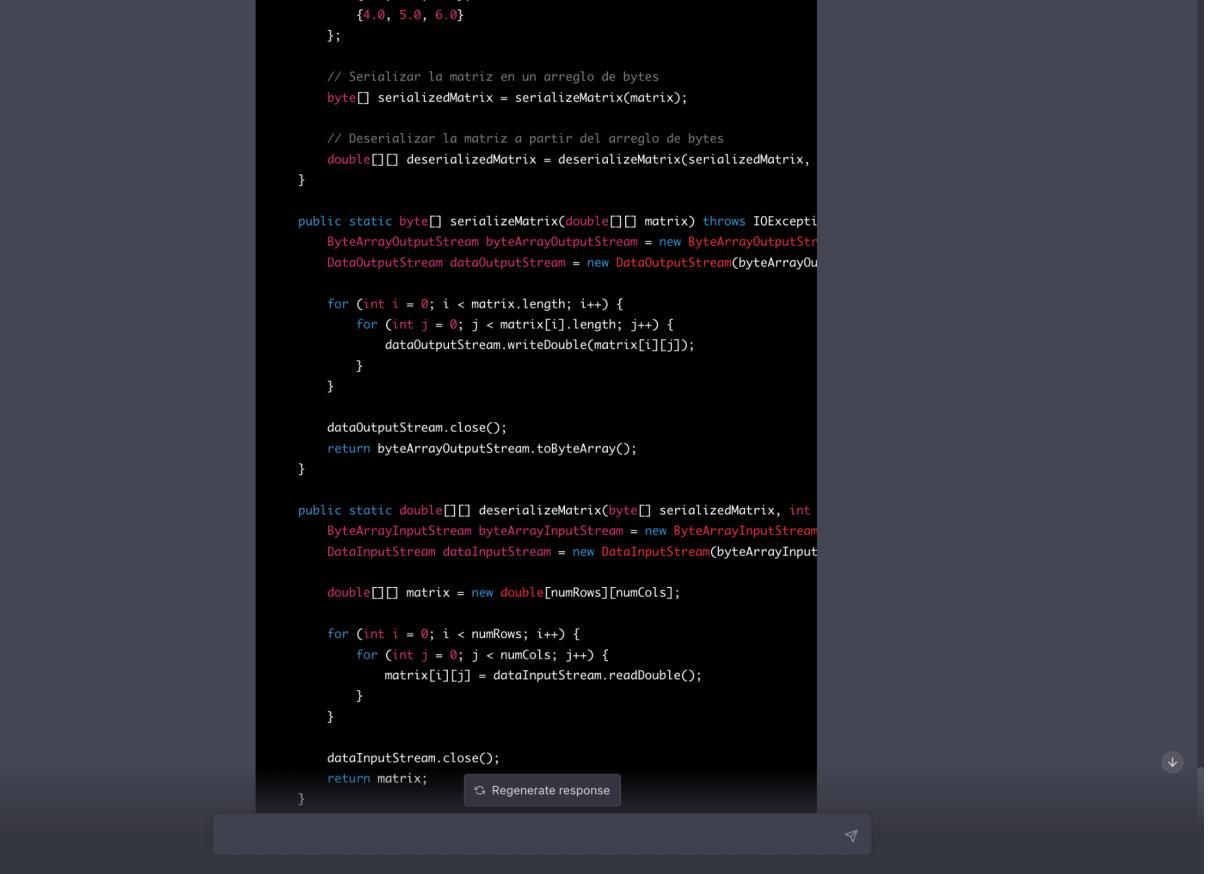
💡 cuál sería la estrategia de serialización más eficaz para comunicar esta cantidad de información entre los nodos usando sockets en Java?

La estrategia de serialización más eficaz para comunicar esta cantidad de información entre los nodos usando sockets en Java sería utilizar flujos de datos binarios y serializar las matrices como secuencias de números de tipo `double`.

Aquí tienes un ejemplo de cómo puedes serializar y deserializar matrices utilizando `DataOutputStream` y `DataInputStream` en Java:

```
java
import java.io.*;

public class MatrixSerialization {
    public static void main(String[] args) throws IOException {
        // Ejemplo de matriz a serializar
        double[][] matrix = {
            {1.0, 2.0, 3.0},
            {4.0, 5.0, 6.0}
        };
        // Serializar la matriz
        DataOutputStream dos = new DataOutputStream(new FileOutputStream("matrix.ser"));
        dos.writeDouble(matrix.length);
        dos.writeDouble(matrix[0].length);
        for (double[] row : matrix) {
            for (double value : row) {
                dos.writeDouble(value);
            }
        }
        dos.close();
    }
}
```



```
    {4.0, 5.0, 6.0}
};

// Serializar la matriz en un arreglo de bytes
byte[] serializedMatrix = serializeMatrix(matrix);

// Deserializar la matriz a partir del arreglo de bytes
double[][] deserializedMatrix = deserializeMatrix(serializedMatrix,
}

public static byte[] serializeMatrix(double[][] matrix) throws IOException {
    ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
    DataOutputStream dataOutputStream = new DataOutputStream(byteArrayOutputStream);

    for (int i = 0; i < matrix.length; i++) {
        for (int j = 0; j < matrix[i].length; j++) {
            dataOutputStream.writeDouble(matrix[i][j]);
        }
    }

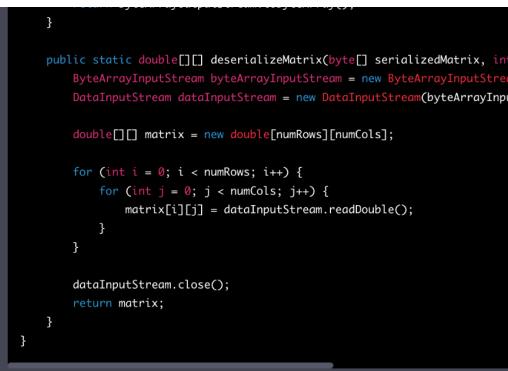
    dataOutputStream.close();
    return byteArrayOutputStream.toByteArray();
}

public static double[][] deserializeMatrix(byte[] serializedMatrix, int numBytes) throws IOException {
    ByteArrayInputStream byteArrayInputStream = new ByteArrayInputStream(serializedMatrix);
    DataInputStream dataInputStream = new DataInputStream(byteArrayInputStream);

    double[][] matrix = new double[numBytes][numBytes];

    for (int i = 0; i < numBytes; i++) {
        for (int j = 0; j < numBytes; j++) {
            matrix[i][j] = dataInputStream.readDouble();
        }
    }

    dataInputStream.close();
    return matrix;
}
```



```
}
```

En este ejemplo, las funciones `serializeMatrix()` y `deserializeMatrix()` manejan la serialización y deserialización de matrices, respectivamente. Puedes utilizar estas funciones para enviar y recibir matrices a través de sockets en Java. Cuando envíes las matrices a través de sockets, simplemente escribe el arreglo de bytes serializado en el `OutputStream` del socket y lee los datos del `InputStream` del socket en el nodo receptor.

Ten en cuenta que este enfoque de serialización no incluye información sobre las dimensiones de la matriz. Por lo tanto, es necesario que el nodo receptor tenga conocimiento de las dimensiones de la matriz que va a recibir o que se envíe esta información por separado antes de transmitir la matriz serializada.

Con esta información es más que suficiente para poder realizar el programa final con el cual se podrá ejecutar la tarea.

Ejecución del Programa

Lo primero será compilar el programa, de la siguiente manera:

```
> javac Tarea2.java
~/Proyectos/IPN/sistemas-distribuidos-escom/tareas/tarea-2 main*
> 
```

Para ejecutar el programa, el primer argumento será un número entero indicando el número de nodo, si es el 0, también se pedirá que se ingrese el valor de N. De la siguiente manera:

```
~/Proyectos/IPN/sistemas-distribuidos-escom/tareas/tarea-2 main*
> java Tarea2 0 12
> 
```

```
> java Tarea2 3
Nodo #3 esperando conexión en el puerto 5002 ...
```

```
> java Tarea2 1
Nodo #1 esperando conexión en el puerto 5000 ...
```

```
> java Tarea2 2
Nodo #2 esperando conexión en el puerto 5001 ...
```

Una vez que se inicie el nodo 0 (arriba a la izquierda), se ejecutará el programa y veremos el resultado:

```

-74.0 64.0 202.0 340.0 478.0 616.0 754.0 892.0 1030.0 1168.0 1306.0 1444 -26.0 16.0 58.0 100.0
.0 -38.0 28.0 94.0 160.0
-86.0 76.0 238.0 400.0 562.0 724.0 886.0 1048.0 1210.0 1372.0 1534.0 169 -50.0 40.0 130.0 220.0
6.0
-98.0 88.0 274.0 460.0 646.0 832.0 1018.0 1204.0 1390.0 1576.0 1762.0 19 Matriz C2:
48.0 58.0 76.0 94.0 112.0
-110.0 100.0 310.0 520.0 730.0 940.0 1150.0 1360.0 1570.0 1780.0 1990.0 142.0 184.0 226.0 268.0
2200.0 226.0 292.0 358.0 424.0
-122.0 112.0 346.0 580.0 814.0 1048.0 1282.0 1516.0 1750.0 1984.0 2218.0 310.0 400.0 490.0 580.0
2452.0
-134.0 124.0 382.0 640.0 898.0 1156.0 1414.0 1672.0 1930.0 2188.0 2446.0 Matriz C3:
2704.0 130.0 148.0 166.0 184.0
-146.0 136.0 418.0 700.0 982.0 1264.0 1546.0 1828.0 2110.0 2392.0 2674.0 310.0 352.0 394.0 436.0
2956.0 490.0 556.0 622.0 688.0
670.0 760.0 850.0 940.0

Checksum: 375408.0
~/Proyectos/IPN/sistemas-distribuidos-escom/tareas/tarea-2 main*
>

-122.0 112.0 346.0 580.0 -74.0 64.0 202.0 340.0
-134.0 124.0 382.0 640.0 -86.0 76.0 238.0 400.0
-146.0 136.0 418.0 700.0 -98.0 88.0 274.0 460.0

Matriz C2:
730.0 940.0 1150.0 1360.0 394.0 508.0 622.0 736.0
814.0 1048.0 1282.0 1516.0 478.0 616.0 754.0 892.0
898.0 1156.0 1414.0 1672.0 562.0 724.0 886.0 1048.0
982.0 1264.0 1546.0 1828.0 646.0 832.0 1018.0 1204.0

Matriz C3:
1570.0 1780.0 1990.0 2200.0 850.0 964.0 1078.0 1192.0
1750.0 1984.0 2218.0 2452.0 1030.0 1168.0 1306.0 1444.0
1930.0 2188.0 2446.0 2704.0 1210.0 1372.0 1534.0 1696.0
2110.0 2392.0 2674.0 2956.0 1390.0 1576.0 1762.0 1948.0

~/Proyectos/IPN/sistemas-distribuidos-escom/tareas/tarea-2 main*
> ~~/Proyectos/IPN/sistemas-distribuidos-escom/tareas/tarea-2 main*
> ALCOCER ♥ 55% Fri 24/03 09:30 PM CST

```

ESCOM ➔ 1 general ➔ 2 distribuidos*

Esto fue con N=12, ahora probaremos con N=900:

```

1.3494124E9 1.35116515E9 1.3529179E9 1.35467065E9 1.3564234E9 1.3581761 5.0915215E8 5.098249E8 5.1049765E8 5.111704E8 5.1184315E8 5.125159E8 5
5E9 1.3599289E9 1.36168165E9 1.3634344E9 1.36518715E9 1.3669399E9 1.36861 1318865E8 5.138614E8 5.1453415E8 5.152069E8 5.1587965E8 5.165524E8 5.1
9265E9 1.3704454E9 1.37219815E9 1.3739509E9 1.37570365E9 1.3774564E9 1.3 722515E8 5.178979E8 5.1857065E8 5.192434E8 5.1991615E8 5.205889E8 5.212
7920915E9 1.3809619E9 1.38271465E9 1.3844674E9 1.38622015E9 1.3879729E9 6165E8 5.219344E8 5.2260715E8 5.232799E8 5.2395265E8 5.246254E8 5.25298
1.38972565E9 1.3914784E9 1.39323115E9 1.3949839E9 1.39673665E9 1.3984894 15E8 5.259709E8 5.2664365E8 5.273164E8 5.2798915E8 5.286619E8 5.2933465
E9 1.40024215E9 1.401994E9 1.40374765E9 1.4055004E9 1.40725315E9 1.4090 E8 5.300074E8 5.3068015E8 5.313529E8 5.3202565E8 5.326984E8 5.3337115E8
059E9 1.41075865E9 1.4125114E9 1.41426415E9 1.4160169E9 1.41776965E9 1.4 5.340439E8 5.3471665E8 5.353894E8 5.3606215E8 5.367349E8 5.3740765E8 5
195224E9 1.42127515E9 1.4230279E9 1.42478065E9 1.4265334E9 1.42828615E9 380804E8 5.3875315E8 5.394259E8 5.4009865E8 5.407714E8 5.4144415E8 5.4
1.4300389E9 1.43179165E9 1.4333444E9 1.43529715E9 1.4370499E9 1.43880265 21169E8 5.427965E8 5.434624E8 5.4413515E8 5.448079E8 5.4548065E8 5.461
E9 1.4405554E9 1.44230815E9 1.4440609E9 1.44581365E9 1.4475664E9 1.44931 534E8 5.4682615E8 5.474989E8 5.4817165E8 5.488444E8 5.4951715E8 5.50189
915E9 1.4510719E9 1.45282465E9 1.4545774E9 1.45633015E9 1.4580829E9 1.45 9E8 5.5086265E8 5.515354E8 5.5220815E8 5.528809E8 5.5355365E8 5.542264E
98365E9 1.4615884E9 1.46334115E9 1.4650939E9 1.46684665E9 1.4685994E9 1 8 5.5489915E8 5.555719E8 5.5624465E8 5.569174E8 5.5759015E8 5.582629E8
.47035215E9 1.4721049E9 1.47385765E9 1.4756104E9 1.47736315E9 1.4791159E 5.589356E8 5.596084E8 5.6028115E8 5.609539E8 5.6162665E8 5.622994E8 5.
9 1.48086865E9 1.4826214E9 1.48437415E9 1.4861269E9 6297215E8 5.636449E8 5.6431765E8 5.649904E8 5.6566315E8 5.663359E8 5.67
00865E8 5.676814E8 5.6835415E8 5.690269E8

Checksum: 1.03139011125E15
~/Proyectos/IPN/sistemas-distribuidos-escom/tareas/tarea-2 main*
>

9 1.3388959E9 1.34064865E9 1.3424014E9 1.34415415E9 1.3459069E9 1.347659 9.1964215E8 9.208549E8 9.2206765E8 9.232804E8 9.2449315E8 9.257059E8 9
65E9 1.3494124E9 1.35116515E9 1.3529179E9 1.35467065E9 1.3564234E9 1.358 2691865E8 9.281314E8 9.2934415E8 9.305569E8 9.3176965E8 9.329242E8 9.3
17615E9 1.3599289E9 1.36168165E9 1.3634344E9 1.36518715E9 1.3669399E9 1.3 419515E8 9.354079E8 9.3662065E8 9.378334E8 9.3904615E8 9.402589E8 9.414
36869265E9 1.3704454E9 1.37219815E9 1.3739509E9 1.37570365E9 1.3774564E9 7165E8 9.426844E8 9.4389715E8 9.451099E8 9.4632265E8 9.473554E8 9.48748
1.37920915E9 1.3809619E9 1.38271465E9 1.3844674E9 1.38622015E9 1.387972 15E8 9.499609E8 9.5117365E8 9.523864E8 9.5359915E8 9.548119E8 9.5602465
9E9 1.38972565E9 1.3914784E9 1.39323115E9 1.3949839E9 1.39673665E9 1.3984894 1.39E8 9.572374E8 9.5845015E8 9.596629E8 9.6087565E8 9.620884E8 9.6330115E8
4894E9 1.40024215E9 1.401994E9 1.40374765E9 1.4055004E9 1.40725315E9 1. 9.645139E8 9.6572665E8 9.669394E8 9.6815215E8 9.693649E8 9.7057765E8 9
4090059E9 1.41075865E9 1.4125114E9 1.41426415E9 1.4160169E9 1.41776965E9 717904E8 9.7300315E8 9.742159E8 9.7542865E8 9.766414E8 9.7785415E8 9.7
1.4195224E9 1.42127515E9 1.4230279E9 1.42478065E9 1.4265334E9 1.4282861 90669E8 9.8027965E8 9.814924E8 9.8270515E8 9.839179E8 9.8513065E8 9.863
5E9 1.4300389E9 1.43179165E9 1.4333444E9 1.43529715E9 1.4370499E9 1.43880265 434E8 9.87555615E8 9.887689E8 9.8989165E8 9.911944E8 9.9240715E8 9.93619
0265E9 1.4405554E9 1.44230815E9 1.4440609E9 1.44581365E9 1.4475664E9 1.49E8 9.9483265E8 9.960454E8 9.9725815E8 9.984709E8 9.9968365E8 1.0008964
4931915E9 1.4510719E9 1.45282465E9 1.4545774E9 1.45633015E9 1.4580829E9 9. E9 1.00210915E9 1.0033219E9 1.00453465E9 1.0057474E9 1.00696015E9 1.008
1.4598365E9 1.4615884E9 1.46334115E9 1.4650939E9 1.46684665E9 1.4685994 1729E9 1.00938565E9 1.0105894E9 1.01181115E9 1.0130239E9 1.01423665E9 1
E9 1.47035215E9 1.4721049E9 1.47385765E9 1.4756104E9 1.47736315E9 1.4791159 0154494E9 1.01666215E9 1.0178749E9 1.01908765E9 1.0203004E9 1.02151315
159E9 1.48086865E9 1.4826214E9 1.48437415E9 1.4861269E9 1.49E9 1.0227259E9 1.02393865E9 1.0251514E9 1.02636415E9 1.0275769E9

~/Proyectos/IPN/sistemas-distribuidos-escom/tareas/tarea-2 main*
> ~~/Proyectos/IPN/sistemas-distribuidos-escom/tareas/tarea-2 main*
> ALCOCER ♥ 55% Fri 24/03 09:32 PM CST

```

ESCOM ➔ 1 general ➔ 2 distribuidos*

Para ejecutarlo en la nube, primero deberemos confirmar que podemos conectarnos con nuestras máquinas virtuales:

```
~/Proyectos/IPN/sistemas-distribuidos-escom/tareas/tarea-2 main* 
> 

https://ubuntu.com/azure/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Mar 24 21:24:28 2023 from 148.204.56.40
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

humbertoody@T2-2016630495-1:~$ 

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

humbertoody@T2-2016630495-3:~$ █
ESCOM 1 general 2 distribuidos* █
humbertoody@T2-2016630495-2:~$ █
ALCOCER 53% Fri 24/03 09:39 PM CST
```

Ahora realizaremos la copia de los archivos a las tres máquinas usando SCP:

```
~/Proyectos/IPN/sistemas-distribuidos-escom/tareas/tarea-2 main* 
> █

> scp -i humbertoody-azure.pem -r ..../tareas/tarea-2/Tarea2.java humbe
rtowoodoy@24.235.12.118:/home/humbertowood
Tarea2.java          100% 8977   135.9KB/s  00:00
~/Proyectos/IPN/sistemas-distribuidos-escom/azure main*      21:44:20
> 

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

humbertoody@T2-2016630495-2:~$ logout
Connection to 20.25.1.241 closed.
> ssh -i humbertoody-azure.pem humbertoody@20.25.1.241
> scp -i humbertoody-azure.pem -r ..../tareas/tarea-2/Tarea2.java humbe
rtowoodoy@20.25.1.241:/home/humbertowood
Tarea2.java          100% 8977   124.5KB/s  00:00
~/Proyectos/IPN/sistemas-distribuidos-escom/azure main*      21:45:14
> 

ESCOM 1 general 2 distribuidos* █
humbertoody@T2-2016630495-2:~$ █
ALCOCER 52% Fri 24/03 09:45 PM CST
```

Ahora verificaremos que tengamos Java instalado en todos los nodos:

The screenshot shows a terminal window with four panes. The top-left pane shows the Java and javac versions for node 1 (T2-2016630495-1). The top-right pane shows the same for node 3 (T2-2016630495-3). The bottom-left pane shows the versions for node 2 (T2-2016630495-2). The bottom-right pane shows the Java and javac versions for node 1 again. A status bar at the bottom indicates 'ALCOCER' with a battery level of 51% and the date 'Fri 24/03 09:51 PM CST'.

```
~/Proyectos/IPN/sistemas-distribuidos-escom/tareas/tarea-2 main*  
>  
  
humberto@T2-2016630495-1:~$ java --version && javac --version  
openjdk 11.0.18 2023-01-17  
OpenJDK Runtime Environment (build 11.0.18+10-post-Ubuntu-0ubuntu120.04  
.1)  
OpenJDK 64-Bit Server VM (build 11.0.18+10-post-Ubuntu-0ubuntu120.04.1,  
mixed mode, sharing)  
javac 11.0.18  
humberto@T2-2016630495-1:~$  
  
humberto@T2-2016630495-3:~$ java --version && javac --version  
openjdk 11.0.18 2023-01-17  
OpenJDK Runtime Environment (build 11.0.18+10-post-Ubuntu-0ubuntu120.04  
.1)  
OpenJDK 64-Bit Server VM (build 11.0.18+10-post-Ubuntu-0ubuntu120.04.1,  
mixed mode, sharing)  
javac 11.0.18  
humberto@T2-2016630495-3:~$  
  
humberto@T2-2016630495-2:~$ java --version && javac --version  
openjdk 11.0.18 2023-01-17  
OpenJDK Runtime Environment (build 11.0.18+10-post-Ubuntu-0ubuntu120.04  
.1)  
OpenJDK 64-Bit Server VM (build 11.0.18+10-post-Ubuntu-0ubuntu120.04.1,  
mixed mode, sharing)  
javac 11.0.18  
humberto@T2-2016630495-2:~$  
  
humberto@T2-2016630495-1:~$  
  
ESCOM ➤ 1 general ➤ 2 distribuidos* ➤ ALCOCER 🔋 51% Fri 24/03 09:51 PM CST
```

Finalmente, realizaremos la ejecución del programa usando un N=900.

Primero compilamos y ejecutamos los servidores en los tres nodos en la nube:

The screenshot shows a terminal window with four panes. The top-left pane shows the compilation and execution of Tarea2.java on node 1 (T2-2016630495-1). The top-right pane shows the output for node 3 (T2-2016630495-3). The bottom-left pane shows the output for node 2 (T2-2016630495-2). The bottom-right pane shows the output for node 1 again. A status bar at the bottom indicates 'ALCOCER' with a battery level of 49% and the date 'Fri 24/03 10:00 PM CST'.

```
~/Proyectos/IPN/sistemas-distribuidos-escom/tareas/tarea-2 main*  
> javac Tarea2.java && java Tarea2 0 900  
  
humberto@T2-2016630495-3:~$ javac Tarea2.java && java Tarea2 3  
Nodo #3 esperando conexión en el puerto 8082 ...  
  
humberto@T2-2016630495-1:~$ javac Tarea2.java && java Tarea2 1  
Nodo #1 esperando conexión en el puerto 8080 ...  
  
humberto@T2-2016630495-2:~$ javac Tarea2.java && java Tarea2 2  
Nodo #2 esperando conexión en el puerto 8081 ...  
  
humberto@T2-2016630495-1:~$  
  
ESCOM ➤ 1 general ➤ 2 distribuidos* ➤ ALCOCER 🔋 49% Fri 24/03 10:00 PM CST
```

Posterioremente ejecutamos nuestro programa localmente (arriba izquierda) y veremos como toma un tiempo (alrededor de 4s) en enviarse la información a cada nodo:

```

895E8 -2.108832E8 -2.1146745E8 -2.120517E8 -2.1263595E8 -2.132202E8 -2.1
5E8 -2.062092E8 -2.0679345E8 -2.073777E8 -2.0796195E8 -2.085462E8 -2.09
380445E8 -2.143887E8 -2.1497295E8 -2.155572E8 -2.1614145E8 -2.167257E8 -1
3045E8 -2.097147E8 -2.1029895E8 -2.108832E8 -2.1146745E8 -2.120517E8 -
2.1730995E8 -2.178942E8 -2.1847845E8 -2.190627E8 -2.1964695E8 -2.202312E
2.1263595E8 -2.132202E8 -2.1380445E8 -2.143887E8 -2.1497295E8 -2.155572
8 -2.2081545E8 -2.213997E8 -2.2198395E8 -2.225682E8 -2.2315245E8 -2.2373
E8 -2.1614145E8 -2.167257E8 -2.1730995E8 -2.178942E8 -2.1847845E8 -2.19
67E8 -2.2432095E8 -2.249052E8 -2.2548945E8 -2.260737E8 -2.2665795E8 -2.2
0627E8 -2.1964695E8 -2.202312E8 -2.2081545E8 -2.213997E8 -2.2198395E8 -
72422E8 -2.2782645E8 -2.284107E8 -2.2899495E8 -2.295792E8 -2.3016345E8 -2
2.225682E8 -2.2315245E8 -2.237367E8 -2.2432095E8 -2.249052E8 -2.2548945
2.307477E8 -2.3133195E8 -2.319162E8 -2.3250045E8 -2.330847E8 -2.3366895E
E8 -2.260737E8 -2.2665795E8 -2.272422E8 -2.2782645E8 -2.284107E8 -2.289
8 -2.342532E8 -2.3483745E8 -2.354217E8 -2.3600595E8 -2.365902E8 -2.37174
9495E8 -2.295792E8 -2.3016345E8 -2.307477E8 -2.3133195E8 -2.319162E8 -
45E8 -2.377587E8 -2.3834295E8 -2.389272E8 -2.3951145E8 -2.400957E8 -2.40
.3250045E8 -2.330847E8 -2.3366895E8 -2.342532E8 -2.3483745E8 -2.354217E
67995E8 -2.412642E8 -2.4184845E8 -2.424327E8 -2.4301695E8 -2.436012E8 -2
B -2.3600595E8 -2.365902E8 -2.3717445E8 -2.377587E8 -2.3834295E8 -2.389
.4418545E8 -2.447697E8 -2.4535395E8 -2.459382E8 -2.4652245E8 -2.471067E8
272E8 -2.3951145E8 -2.400957E8 -2.4067995E8 -2.412642E8 -2.4184845E8 -2
-2.4769095E8 -2.482752E8 -2.4885945E8 -2.494437E8 -2.5002795E8 -2.50612
.424327E8 -2.4301695E8 -2.436012E8 -2.4418545E8 -2.447697E8 -2.4535395E
2E8 -2.5119645E8 -2.517807E8 -2.5236495E8 -2.529492E8 -2.5353345E8 -2.54
1177E8 -2.5470195E8 -2.552862E8 -2.5587045E8 -2.564547E8

Checksum: 1.03139011125E15
~/Proyectos/IPN/sistemas-distribuidos-escom/tareas/tarea-2 main*
>

humberto@T2-2016630495-3:~$
```

```

07395E7 -7.22982E7 -7.252245E7 -7.27467E7 -7.297095E7 -7.31952E7 -7.3419
992E8 -1.4100345E8 -1.414077E8 -1.4181195E8 -1.422162E8 -1.4262045E8 -1
45E7 -7.36437E7 -7.386795E7 -7.40922E7 -7.431645E7 -7.45407E7 -7.476495E
.430247E8 -1.4342895E8 -1.438332E8 -1.4423745E8 -1.446417E8 -1.4504595E
7 -7.49892E7 -7.521345E7 -7.54377E7 -7.566195E7 -7.58862E7 -7.611045E7 -8
-8.1454502E8 -1.4585445E8 -1.462587E8 -1.4666295E8 -1.470672E8 -1.4747
7.63347E7 -7.655895E7 -7.67832E7 -7.700745E7 -7.72317E7 -7.745595E7 -7.7
145E8 -1.478757E8 -1.4827995E8 -1.486842E8 -1.4908845E8 -1.494927E8 -1.
6802E7 -7.790445E7 -7.81287E7 -7.835295E7 -7.85772E7 -7.880145E7 -7.9025
4989695E8 -1.503012E8 -1.5070545E8 -1.511097E8 -1.5151395E8 -1.519182E8
7E7 -7.924995E7 -7.94742E7 -7.969845E7 -7.99227E7 -8.014695E7 -8.03712E7
-1.5232245E8 -1.527267E8 -1.5313095E8 -1.533532E8 -1.5393945E8 -1.5434
-8.059545E7 -8.08197E7 -8.104395E7 -8.12682E7 -8.149245E7 -8.17167E7 -8
37E8 -1.5474795E8 -1.551522E8 -1.5555645E8 -1.559607E8 -1.5636495E8 -1.
194095E7 -8.21652E7 -8.238945E7 -8.26137E7 -8.283795E7 -8.30622E7 -8.32
567692E8 -1.5717345E8 -1.575777E8 -1.5798195E8 -1.583862E8 -1.5879045E8
8645E7 -8.35107E7 -8.373495E7 -8.39592E7 -8.418345E7 -8.44077E7 -8.46319
-1.591947E8 -1.5959895E8 -1.600032E8 -1.6040745E8 -1.608117E8 -1.61215
5E7 -8.48562E7 -8.508045E7 -8.53047E7 -8.552895E7 -8.57532E7 -8.597745E7
95E8 -1.616202E8 -1.6202445E8 -1.624287E8 -1.6283295E8 -1.632372E8 -1.
6 -8.62017E7 -8.642595E7 -8.66502E7 -8.687445E7 -8.70987E7 -8.732295E7 -8
364145E8 -1.640457E8 -1.644995E8 -1.648542E8 -1.6525845E8 -1.656627E8
.75472E7 -8.777145E7 -8.79957E7 -8.821995E7 -8.84442E7 -8.866845E7 -8.88
-1.6606695E8 -1.664712E8 -1.6687545E8 -1.672797E8 -1.6768395E8 -1.68088
927E7 -8.911695E7 -8.93412E7 -8.956545E7 -8.97897E7 -9.001395E7 -9.02382
2E8 -1.6849245E8 -1.688967E8 -1.6930095E8 -1.697052E8 -1.7010945E8 -1.
E7 -9.046245E7 -9.06867E7 -9.091095E7 -9.11352E7 -9.135945E7 -9.15837E7
05137E8 -1.7091795E8 -1.713222E8 -1.7172645E8 -1.721307E8 -1.7253495E8
-9.180795E7 -9.20322E7 -9.225645E7 -9.24807E7 -9.270495E7 -9.29292E7 -9.
-1.729392E8 -1.7334345E8 -1.737477E8 -1.7415195E8 -1.745562E8 -1.749604
315345E7 -9.33777E7 -9.360195E7 -9.38262E7 -9.405045E7 -9.42747E7
5E8 -1.753647E8

humberto@T2-2016630495-2:~$
```

ALCOCER ♥ 50% Fri 24/03 09:59 PM CST

Con esto queda finalizado el desarrollo de la práctica.

Algunos detalles:

- Si quito todas las impresiones, funciona más rápido el programa.
- El programa no funciona con N=3000, asumo que son muchos datos.

Conclusión

La tarea abordó el problema de la multiplicación de matrices cuadradas de manera distribuida utilizando tres máquinas virtuales de Azure. Los temas cubiertos incluyeron la inicialización de matrices, el cálculo del producto de matrices, la implementación de algoritmos distribuidos y la comunicación entre nodos a través de sockets en Java.

La complejidad de la tarea provino de la necesidad de dividir las matrices A y B en partes más pequeñas para ser procesadas en paralelo por diferentes nodos, así como de la comunicación entre los nodos para enviar y recibir las matrices necesarias y los resultados parciales. Además, se requirió el cálculo del checksum de la matriz C para validar la corrección del programa.

Se hizo uso de la transpuesta de la matriz B (BT) para facilitar el cálculo de las submatrices de C y aprovechar la localidad espacial. Esto resultó en un enfoque más eficiente para el cálculo del producto de matrices.

Uno de los aspectos interesantes de la tarea fue la implementación de la comunicación entre los nodos utilizando sockets en Java. La serialización y deserialización de matrices se realizó utilizando flujos de datos binarios para lograr una comunicación eficiente entre los nodos.

En resumen, la tarea proporcionó una oportunidad para aplicar conceptos de programación distribuida y comunicación entre nodos a un problema práctico de álgebra lineal, lo que permitió explorar técnicas eficientes para dividir y procesar datos en paralelo en múltiples máquinas virtuales.

– Humberto Alejandro Ortega Alcocer.