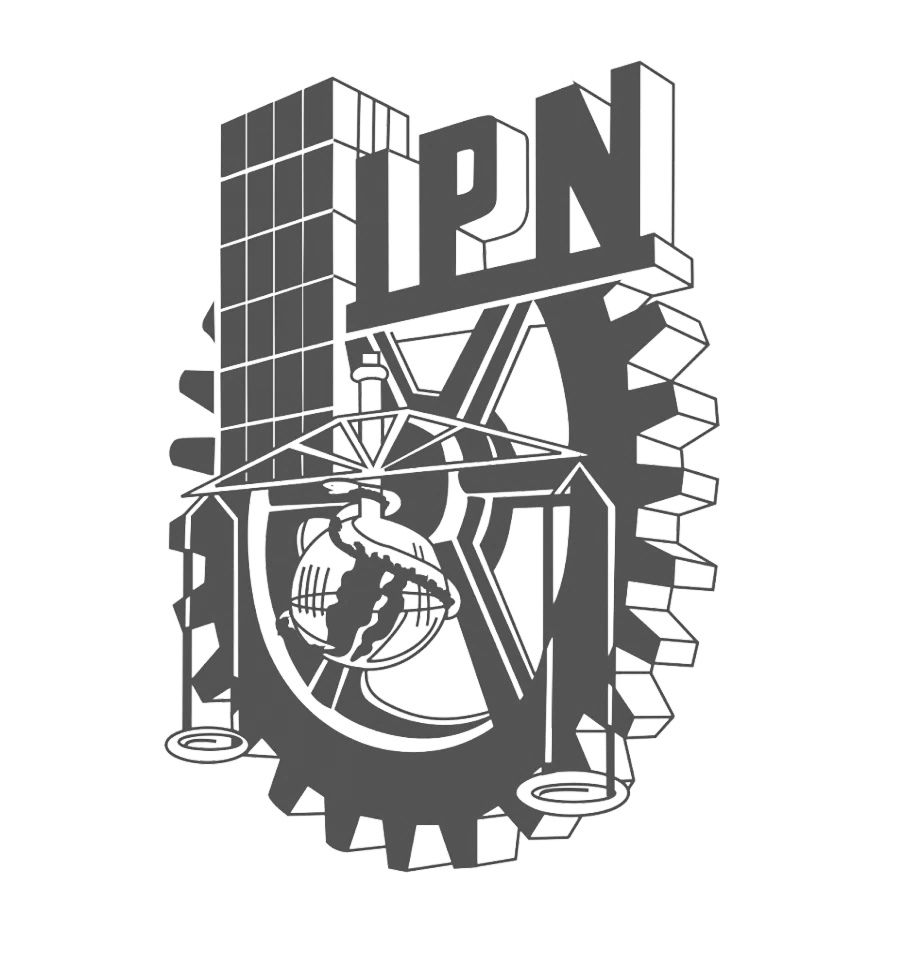


**UNIDAD DE APRENDIZAJE**

**“Desarrollo de sistemas Distribuidos”**

“Tarea 3. multiplicación de matrices”



**ALUMNO:**

**Moreno Hurtado Alejandro**

**GRUPO 4CV13**

**PROFESOR:**

**Pineda Guerrero Carlos**

# Desarrollo de la practica

La multiplicación de matrices es un problema que dependiendo del tamaño de estas pueden llevar un gran tiempo para un equipo de cómputo común. Aun con un equipo de computo potente si el tamaño de la matriz es muy grande podría tardar u ocasionar algún problema con la memoria de la máquina.

Para la solución de este problema podemos optar con dividir la matriz en pequeñas partes y que estas sean resultas en un equipo aparte.

Teniendo el cazo en que tenemos 2 maquinas aparte de nuestro cliente, vamos a dividir la matriz de tal forma que la primera maquina resuelva la parte superior de la matriz y la segunda la parte inferior.

Tabla

Descripción generada automáticamente

Ilustración : Diagrama de referencia

De tal forma:

-Servidor 1: C1, …, C8.

-Servidor 2: C9, … , C16.

Para la creación de nuestras máquinas virtuales se hará en la plataforma de Azure con una cuenta para estudiantes.

Buscaremos el recurso que queremos crear y lo seleccionaremos.

Una captura de pantalla de una computadora

Descripción generada automáticamente

Ilustración : Creación de máquina virtual

Después de esto llenaremos los datos que nos piden (Grupo de recursos, Nombre de la máquina virtual, Imagen de SO, Tamaño de la maquina) , para la autenticación usaremos una por usuario y contraseña para una conexión SSH.

Captura de pantalla de computadora

Descripción generada automáticamente

Ilustración : Parámetros de la máquina virtual.

Para el tipo de almacenamiento usaremos un HDD estándar.

Captura de pantalla de computadora

Descripción generada automáticamente

Ilustración : Discos de MV

Nos aseguramos que el puerto SSH(22) este seleccionado ya que eso nos dará acceso a la máquina.

Una captura de pantalla de una computadora

Descripción generada automáticamente

Ilustración : Redes de MV

En monitoreo vamos a deshabilitar el diagnostico de arranque y daremos procedemos a crear la maquina

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

Ilustración : Monitoreo

Revisemos que todos los parámetros relacionas sean correctos antes de crear nuestra máquina virtual.

Una captura de pantalla de una computadora

Descripción generada automáticamente

Ilustración : Creación de la máquina virtual

Una vez creada nuestra maquina virtual podemos proceder a utilizarla, pero para esta practica primero demos configurar el puerto por el cual se conectara nuestro programa para multiplicar matrices.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración : Máquina Virtual creada

Para abrir el puerto que queremos utilizar iremos al recurso de la maquina virtuar>Redes y seleccionaremos la opción de “Agregar regla de puerto de entrada”

Una captura de pantalla de una computadora

Descripción generada automáticamente

Ilustración : Agregar regla de puerto de entrada

Seleccionaremos el puerto o el rango de puerto a abrir para después agregar la regla.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración : Creando la regla de puerto

Una vez creada la máquina virtual y con la configuración necesaria para que sea nuestro servidor, empezaremos a instalar el JDK para la ejecución de nuestro programa.

Nos conectaremos a nuestra maquina por SSH de la siguiente manera: ssh *usuario@ip*

Texto

Descripción generada automáticamente

Ilustración : Conexión SSH

Una vez ingresemos la contraseña tendremos acceso a nuestra maquina virtual por medio de SSH.

Para la instalacion de JDK primero tendremos que actualizar las dependencias de nuestra maquina recién instalada con el comando:

sudo apt update

Imagen que contiene Texto

Descripción generada automáticamente

Ilustración : Actualización de dependencias

Una vez con las dependencias actualizadas procedemos a la instalacion de java con los comandos de:

*sudo apt install default-jre*

*sudo apt install default-jdk*

Texto

Descripción generada automáticamente

Ilustración : Instalación de java

Una vez terminada la instalación nos deberá reconocer las variables de java

Una vez terminada la instalacion de java procedemos a enviar el código fuente de nuestro programa a nuestra maquina vía sftp. Para eso haremos la conexión con el comando:

sftp *usuario@ip*

Texto

Descripción generada automáticamente

Ilustración : Conexión sftp

Al igual que SSH ingresaremos nuestra contraseña y nos dará conexión a nuestra máquina virtual.

Para enviar un archivo de nuestro equipo local a nuestro servidor remoto usaremos el comando de put seguido de la dirección de nuestro archivo en nuestra maquina local. Si no estamos seguros de en que parte de nuestra maquina virtual nos encontramos podemos ejecutar el comando *lpwd* y para saber a que carpeta de nuestro servidor nos conectamos podemos usar el comando *pwd*

Texto

Descripción generada automáticamente con confianza baja

Ilustración : Comando para saber ruta

Buscamos nuestro archivo en nuestra maquina local con el comando *lls* y con el comando *put* lo enviamos a nuestro servidor remoto.

Para verificar esto solo debemos ejecutar el comando *ls* para mostrar los archivos en nuestro directorio de conexión.

Texto

Descripción generada automáticamente

Ilustración : Subida de archivo por sftp

Procedemos a la ejecución de nuestro programa con una matriz de 12.

Imagen que contiene Texto

Descripción generada automáticamente

Ilustración : Multiplicacion de Matrices de 12

Procedemos a la ejecución de nuestro programa con una matriz de 4000.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

# Conclusión

El crear un solo programa para cliente y servidor es una técnica muy interesante para el computo distribuido ya que solo necesitas de un parámetro para diferencias que función se ejecutara en la máquina.

Un sistema distribuido para resolver el problema de la multiplicación de matrices para una mejor velocidad de cálculo es interesante aunque tambien dependerá de la conexión que se tenga con los servidores , la potencia que cuentan estos servidores y tambien las funciones que tenga el cliente ya que este pueden recibir los datos pero aparte debe procesarlos.