Instituto

Politécnico

Nacional

Escuela Superior de Cómputo

Multiplicación de matrices utilizando objetos distribuidos

TAREA 5

Materia:

Desarrollo De Sistemas Distribuidos

Grupo:

4CV14

Profesor:

Pineda Guerrero Carlos

Alumno:

Castro Cruces Jorge Eduardo

Boleta:

2015080213

Fecha:

Viernes, 22 de octubre de 2021

# Desarrollo del programa

Cada alumno deberá desarrollar un sistema que calcule el producto de dos matrices cuadradas utilizando Java RMI, tal como se explicó en clase.

Se deberá ejecutar dos casos:

* N=9, se deberá desplegar las matrices A, B y C y el checksum de la matriz C.
* N=3000, deberá desplegar el checksum de la matriz C.

Los elementos de las matrices A, B y C serán de tipo double y el checksum será de tipo double.

Se deberá inicializar las matrices A y B de la siguiente manera:  
  
A[i][j]= 3\*i + j  
B[i][j] = i - 4 \* j  
  
Se deberá dividir las matrices A y B en tres partes, por tanto la matriz C estará dividida en 9 partes:

Gráfico

Descripción generada automáticamente con confianza baja

El cliente RMI ejecutará en una máquina virtual con Ubuntu en Azure (nodo 0). El servidor RMI ejecutará en tres máquinas virtuales (nodo 1, nodo 2 y nodo 3) con Ubuntu en Azure. El programa rmiregistry ejecutará en cada nodo donde ejecute el servidor RMI. El nodo 1 calculará los productos C1, C2 y C3, el nodo 2 calculará los productos C4, C5 y C6, y el nodo 3 calculará los productos C7, C8 y C9.

Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza media

El cliente RMI inicializará las matrices A y B, obtendrá la transpuesta de la matriz B, invocará el método remoto multiplica\_matrices(), calculará el checksum de la matriz C, y en su caso (N=9) desplegará las matrices A, B y C.

Se deberá adaptar las funciones: separa\_matriz(), multiplica\_matrices() y acomoda\_matriz() que vimos en clase.

Se **deberá**subir a la plataforma los archivos texto correspondientes al código fuente del sistema desarrollado y un reporte de la tarea en formato PDF con portada, desarrollo y conclusiones como mínimo. El archivo PDF deberá incluir las capturas de pantalla de la compilación y ejecución del programa, se **deberá incluir** la captura de pantalla correspondiente a **cada paso** de la creación de la primea máquina virtual (nodo 0). **No se admitirá la tarea** si no incluye las pantallas correspondientes a cada paso del procedimiento de creación de la máquina virtual. Cada captura de pantalla deberá incluir una descripción.

El nombre de cada máquina virtual deberá iniciar con "JR" seguido del número de boleta del alumno, un guion y el número de nodo, por ejemplo, si el número de boleta del alumno es 12345678, entonces el nodo 0 deberá llamarse: JR12345678-0, el nodo 1 deberá llamarse JR12345678-1, y así sucesivamente. **No se admitirá la tarea** si los nodos no se nombran como se indicó anteriormente.

Valor de la tarea: 30% (2.1 puntos de la segunda evaluación parcial)

# Pruebas de escritorio

Ingresar al portal de Azure en la siguiente URL:

[**https://azure.microsoft.com/es-mx/features/azure-portal/**](https://azure.microsoft.com/es-mx/features/azure-portal/)  
  
1. Dar click al botón "Iniciar sesión".  
Una captura de pantalla de una computadora

Descripción generada automáticamente  
2. En el portal de Azure seleccionar "Máquinas virtuales".  
Una captura de pantalla de una computadora

Descripción generada automáticamente

3. Seleccionar la opción "+Crear".  
  
4. Seleccionar la opción "+Virtual machine"  
Una captura de pantalla de una computadora

Descripción generada automáticamente  
5. Seleccionar el grupo de recursos o crear uno nuevo. Un grupo de recursos es similar a una carpeta dónde se pueden colocar los diferentes recursos de nube que se crean en Azure.  
  
6. Ingresar el nombre de la máquina virtual.  
  
7. Seleccionar la región dónde se creará la máquina virtual. Notar que el costo de la máquina virtual depende de la región.  
  
8. Seleccionar la imagen, en este caso vamos a seleccionar Ubuntu Server 18.04 LTS.  
  
9. Dar click en "Seleccionar tamaño" de la máquina virtual, en este caso vamos a seleccionar una máquina virtual con 1 GB de memoria RAM. Dar click en el botón "Seleccionar".  
  
10. En tipo de autenticación seleccionamos "Contraseña".  
  
11. Ingresamos el nombre del usuario, por ejemplo: ubuntu  
  
12. Ingresamos la contraseña y confirmamos la contraseña. La contraseña debe tener al menos 12 caracteres, debe al menos una letra minúscula, una letra mayúscula, un dígito y un carácter especial.  
  
13. En las "Reglas de puerto de entrada" se deberá dejar abierto el puerto 22 para utilizar SSH (la terminal de secure shell).

Captura de pantalla de computadora

Descripción generada automáticamente  
14. Dar click en el botón "Siguiente: Discos>"  
  
15. Seleccionar el tipo de disco de sistema operativo, en este caso vamos a seleccionar HDD estándar.  
Una captura de pantalla de una computadora

Descripción generada automáticamente

16. Dar click en el botón "Siguiente: Redes>"  
  
17. Dar click en el botón "Siguiente: Administración>"  
  
18. En el campo "Diagnóstico de arranque" seleccionar "Desactivado".  
Una captura de pantalla de una computadora

Descripción generada automáticamente  
19. Dar click en el botón "Revisar y crear".  
  
20. Dar click en el botón "Crear".  
Una captura de pantalla de una computadora

Descripción generada automáticamente

21. Dar click a la campana de notificaciones (barra superior de la pantalla) para verificar que la maquina virtual se haya creado.  
Una captura de pantalla de una computadora

Descripción generada automáticamente  
22. Dar click en el botón "Ir al recurso". En la página de puede ver la direción IP pública de la máquina virtual. Esta dirección puede cambiar cada vez que se apague y se encienda la máquina virtual.  
Una captura de pantalla de una computadora

Descripción generada automáticamente

23. Para conectarnos a la máquina virtual vamos a utilizar el programa ssh disponible en Windows, Linux y MacOS.

24. En una ventana de comandos de Windows o una terminal de Linux o MacOS ejecutar el programa ssh así:

ssh usuario@ip

Texto

Descripción generada automáticamente

Donde **usuario**es el usuario que ingresamos en el paso 11, **ip**es la ip pública de la máquina virtual.

25. Para enviar o recibir archivos de la máquina virtual, se puede utilizar el programa sftp disponible en Windows, Linux y MacOS. Se ejecuta así:

sftp usuario@ip

Para enviar archivos se utiliza el comando put y para recibir archivos se utiliza el comando get.

Para mayor información sobre sftp ver:

[**https://www.digitalocean.com/community/tutorials/how-to-use-sftp-to-securely-transfer-files-with-a-remote-server-es**](https://www.digitalocean.com/community/tutorials/how-to-use-sftp-to-securely-transfer-files-with-a-remote-server-es)  
  
  
**Abrir un puerto de entrada**

Para que los programas que ejecutan en la máquina virtual pueda recibir conexiones a través de un determinado puerto, es necesario crear una regla de entrada para el puerto.

Por ejemplo, vamos a abrir el puerto 50000 en la máquina virtual que acabamos de crear:

1. Entrar al portal de Azure
2. Seleccionar "Maquinas virtuales".
3. Seleccionar la máquina virtual.
4. Dar clic en "Redes".
5. Dar clic en el botón "Agregar regla de puerto de entrada".

Una captura de pantalla de una computadora

Descripción generada automáticamente

1. En el campo "Intervalos de puertos de destino" ingresar: 50000
2. Seleccionar el protocolo: TCP
3. En el campo "Nombre" ingresar un nombre para la regla: Puerto\_50000

Una captura de pantalla de una computadora

Descripción generada automáticamente

1. Una vez que realizamos la conexión mediante ssh, procedemos a actualizar el SO e instalar el JDK 8 de Java para poder compilar los programas. (Así fue en cada una de las 4 máquinas)

Pantalla de computadora

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. Cabe acotar que las maquinas se crearon dentro del mismo recurso o grupo, para poder realizar la conexión mediante la IP privada de las máuinas virtuales.
2. Hecho lo anterior, procedimos a enviar los archivos .java (ClienteRMI.java, InterfazRMI.java, ClaseRMI.java, ServidorRMI.java), haciendo uso del protocolo sftp, el cual nos fue de gran ayuda.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

1. Acto seguido, procedimos a la compilación de los archivos (ClienteRMI.java, InterfazRMI.java, ClaseRMI.java, ServidorRMI.java), respectivamente en cada máquina virtual.

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

1. Se ejecutó el comando rmiregistry en segundo plano, para poder ejectuar el ServidorRMI en los nodos 1, 2, 3.
2. Se ejecuto el comando rmiregistry en segundo plano, para poder ejctuar el ClienteRMI en el nodo 0.

Captura de pantalla de computadora

Descripción generada automáticamente

1. Primero se ejecutó el programa cuando **N = 9** y arrojo el siguiente resultado.

**Checksum: -135108.0**

Pantalla de un video juego

Descripción generada automáticamente con confianza media

1. Después se ejecutó el programa cuando **N = 3000** y arrojó el siguiente resultado.

**Checksum: -7.0826408326101299E17**

Pantalla de computadora con fondo negro

Descripción generada automáticamente con confianza media

1. Finalmente, se finalizó la conexión con las máquinas virtuales y se eliminaron los recursos para evitar gasatar todos nuestro créditos de la plataforma Azure for Students.

Una captura de pantalla de una computadora

Descripción generada automáticamente

# Conclusiones

En esta práctica, sí voy a cabo una variación de la práctica anterior de multiplicación de matrices pero en esta ocasión se realizó de forma distribuida eso significa que se distribuyó el trabajo en varias máquinas, precisamente en 3 servidores y un cliente el cual realizaba la solicitud a cada uno de ellos.

Se puede concluir que se lograron los objetivos principales de la práctica ya que los resultados del checksum de cada una de las variantes cuando N = 9 y cuando N = 3000, Cabe mencionar que algunas dificultades que se presentaron al momento de realizar la práctica fueron las siguientes:

Se tuvo que crear máquinas virtuales que pertenecieron al mismo grupo de recursos, esto con el fin de que pudiera realizarse la conexión mediante las IP privadas.

Algunas de las versiones de las máquinas virtuales que la plataforma de Azure nos proporcionaba no nos permitían crear las cuatro máquinas al mismo tiempo, ya que al ser de recursos altos, me refiero a grandes cantidades de memoria, mayor número de núcleos de procesador, mayor velocidad de conexión, etcétera, la plataforma las contaba como si fueran más de una máquina abierto a lo que nos impedía crearlas.

Por último, quiero mencionar que me pareció una práctica muy interesante y muy nutritiva para los conocimientos requeridos para un ingeniero en sistemas computacionales.