



Facultad de
INFORMÁTICA



UNIVERSIDAD
NACIONAL
DE LA PLATA

Programación Distribuida y Tiempo Real

Trabajo Práctico 2

- Joaquin Tartaruga (17023/7)
- Manuel Rubiano (17711/6)

1.

- Versión utilizada en la explicación práctica: Ubuntu 18.04.3 (2019) LTS, perteneciente a la box hashicorp/bionic64
- Versión más reciente: Ubuntu 24.04.1 (2024) LTS, perteneciente a la box alvstack/ubuntu-24.04

Las boxes publicas se pueden obtener de la página de Hashicorp:

<https://portal.cloud.hashicorp.com/vagrant/discover>

La diferencia es que, en el proporcionado por la catedra, se hace una actualización de paquetes y librerías, además de ciertas configuraciones como red e ip. Mientras que al hacer el comando vagrant init, lo único que se muestra es una box base y todas las demás configuraciones comentadas

2.

Ejecutamos las comunicaciones en una misma red, la de la facultad, por eso notamos que los tiempos nos quedan altos.

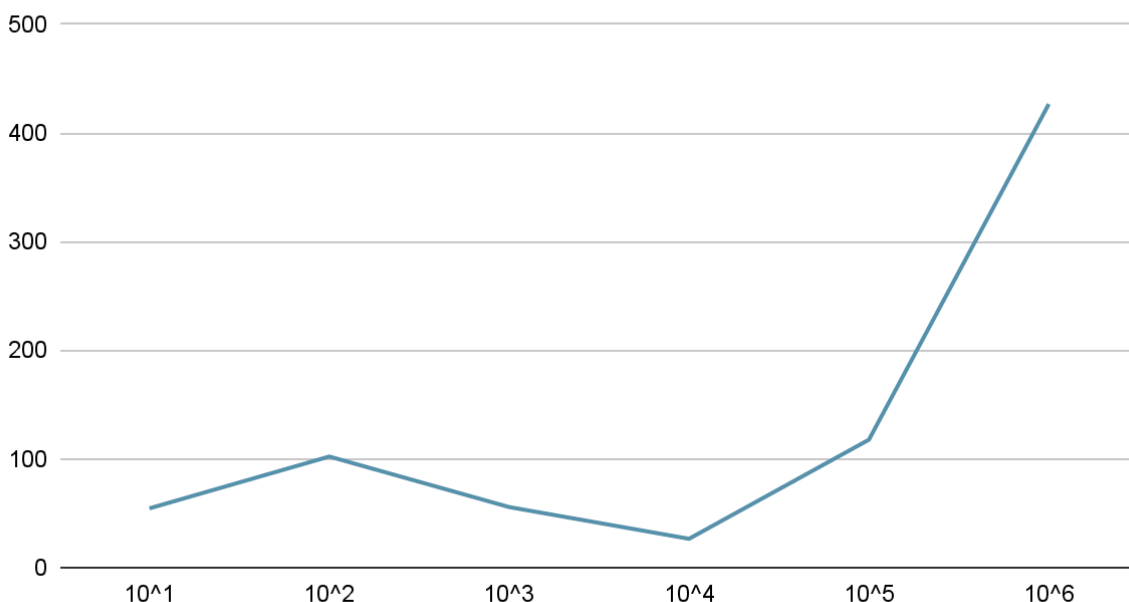
Características de la maquina servidor:

- Procesador: AMD Ryzen 3 5300U with Radeon Graphics 2.60 GHz
- RAM: 8GB
- Sistema Operativo: Windows 10 Pro

Características de la máquina cliente:

- Procesador: AMD Ryzen 5 3450U with Radeon Vega Mobile Gfx 2.10 GHz
- RAM: 8GB
- Sistema Operativo: Windows 11 Home

Dos Host en la red de Informatica



3. A) Realizamos un script donde utilizamos diversos comandos de vagrant para interactuar con las máquinas virtuales (vms). Como por ejemplo:

- halt: para apagar las vms que estén corriendo
- up: para encenderlas
- ssh: para acceder a la terminal de cada vm y así ejecutar los archivos .java

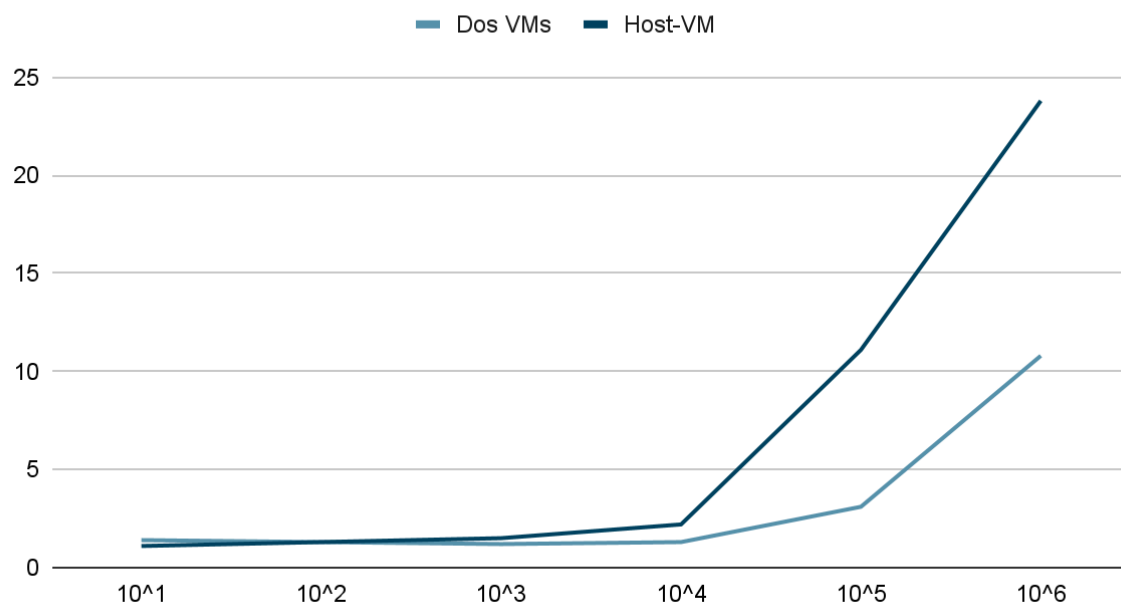
Al mismo tiempo les asignamos Ips a ambas vms, y declaramos que usen una red pública. Haciendo que la vm se conecte directamente a la red externa a la cual está conectado el host

Para ejecutar el experimento: `./scriptEjA.sh > ResultadosA.txt`

B) Realizamos un script similar al anterior, pero esta vez en vez de configurar dos vms utilizamos solo una, en donde allí corríamos al cliente. Y la máquina host pasa a tener el rol de servidor. La ip de wifi la obtenemos dinámicamente desde el script

Para ejecutar el experimento: `$./scriptEjB.sh > ResultadosB.txt`

Grafico de tiempos



4. No importa si el cliente o el servidor se ejecutan en la máquina virtual o en la máquina host. Lo esencial es que el servidor siempre debe comenzar primero, sin importar en qué máquina se encuentre, porque es el encargado de esperar una conexión en un puerto. El cliente, por su parte, sólo puede conectarse al servidor una vez que este ya esté en funcionamiento y listo para aceptar conexiones.

Se puede dar el caso de que el cliente comience a tomar los tiempos antes de que el servidor esté listo para recibir los datos, para eso usamos un sleep para asegurarnos de que el servidor se inicie perfectamente y tenga un margen respecto a la ejecución del cliente