



INFORME MAQUINA VIRTUAL

Persona:

Daniel Horacio Gonzalez Orduz:

Profesor:

Jhon Jairo Corredor

Creado

24/10/2024

1. Con el comando Top encontramos diversos aspectos y lo que observamos es el numero de procesadores que tiene esta maquina que en total serian 4.
Cpus, también observamos cantidad de almacenamiento el cual es 11673 que sería 12gb en almacenamiento y una RAM de 4100 la cual es 4 RAM:

```
Tasks: 261 total, 1 running, 260 sleeping, 0 stopped, 0 zombie
%Cpu0  :  0.3 us,  0.3 sy,  0.0 ni, 99.0 id,  0.0 wa,  0.3 hi,  0.0 si,  0.0 st
%Cpu1  :  0.0 us,  0.7 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2  :  0.0 us,  0.0 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.3 hi,  0.0 si,  0.0 st
%Cpu3  :  0.3 us,  0.0 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 11673.0 total, 4096.5 free, 3182.5 used, 4801.8 buff/cache
MiB Swap: 4100.0 total, 4100.0 free,  0.0 used, 8490.4 avail Mem
```

2. Con el comando lscpu encontramos las especificaciones del hardware como lo es los sockets, los procesadores y su familia, la cantidad de hilos para cada procesador y la arquitectura de este, y la jerarquía en procesos que podemos observar y su nivel de ejecución del más cercano al más lejano, los cuales son:

L1d 192 KIB (4 instancias)

L1i 128 KIB (4 instancias)

L2 5 MIB (4 instancias)

L3 158 MIB (4 instancias)

```
[estudiante@ING-PDGE11 ~]$ lscpu
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Address sizes: 43 bits physical, 48 bits virtual
Byte Order: Little Endian
CPU(s): 4
On-line CPU(s) list: 0-3
Vendor ID: GenuineIntel
Model name: Intel(R) Xeon(R) Gold 6348 CPU @ 2.60GHz
CPU family: 6
Model: 85
Thread(s) per core: 1
Core(s) per socket: 1
Socket(s): 4
Stepping: 7
BogoMIPS: 5187.81
Flags: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse ss
e2 ss syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon nopl xtopology tsc_reliable nonsto
p_tsc cpuid tsc_known_freq pni pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt
tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch ssbd ibrs ibp
b stibp ibrs_enhanced fsgsbase tsc_adjust bmi1 avx2 smep bmi2 invpcid avx512f avx512dq rdseed a
dx smap clflushopt clwb avx512cd avx512bw avx512vl xsaveopt xsavec xsaves arat pku ospke md_cle
ar flush_l1d arch_capabilities

Virtualization features:
Hypervisor vendor: VMware
Virtualization type: full
Caches (sum of all):
L1d: 192 KiB (4 instances)
L1i: 128 KiB (4 instances)
```

```
Virtualization features:
Hypervisor vendor: VMware
Virtualization type: full
Caches (sum of all):
L1d: 192 KiB (4 instances)
L1i: 128 KiB (4 instances)
L2: 5 MiB (4 instances)
L3: 168 MiB (4 instances)
NUMA:
NUMA node(s): 1
NUMA node0 CPU(s): 0-3
Vulnerabilities:
Gather data sampling: Unknown: Dependent on hypervisor status
Itlb multihit: KVM: Mitigation: VMX unsupported
```

3. Acá observamos un almacenamiento de 21GB que se guarda en hardware y su repartición en cada fichero con el fin de que esta información no se perderá así se apague el hardware.

```
18 root      0      0      0      0      0.0      0.0      0:26.73 migration/0
[estudiante@ING-PDGE11 ~]$ df -h
Filesystem                                Size  Used Avail Use% Mounted on
devtmpfs                                  4.0M   0    4.0M   0% /dev
tmpfs                                      5.7G   0    5.7G   0% /dev/shm
tmpfs                                      2.3G  38M    2.3G   2% /run
/dev/mapper/rl_plantillarocky9-root       24G   3.3G    21G  14% /
/dev/sda2                                 960M  389M    572M  41% /boot
/dev/sda1                                1022M   7.1M   1015M   1% /boot/efi
/dev/mapper/rl_plantillarocky9-var        15G   1.4G    14G   9% /var
/dev/mapper/rl_plantillarocky9-home       15G  140M    15G   1% /home
10.43.103.136:/almacen                    24G   5.5G    19G  23% /almacen
tmpfs                                      1.2G  4.0K    1.2G   1% /run/user/1001
[estudiante@ING-PDGE11 ~]$ |
```

ACTIVIDADES:

MAQUINA VIRTUAL 1 (DADA POR EL PROFESOR)

- 1) Se crea una carpeta con mi nombre
\$mkdir DanielG
\$ls
- 2) Se entra a la carpeta creada.
\$cd DanielG
\$ls
- 3) Se observa la ruta actual
\$pwd
- 4) Se crea un nuevo directorio llamado fork
\$mkdir fork
\$cd fork
- 5) Se copian 2 ficheros fork (uni direccional) cliente.c, servidor.c
\$nano cliente.c
\$nano servidor.c

```
root@rocky:~# df -h
Filesystem                                Size  Used Avail Use% Mounted on
devtmpfs                                  4.0M   0    4.0M   0% /dev
tmpfs                                      5.7G   0    5.7G   0% /dev/shm
tmpfs                                      2.3G  38M    2.3G   2% /run
/dev/mapper/rl_plantillarocky9-root       24G   3.3G    21G  14% /
/dev/sda2                                 960M  389M    572M  41% /boot
/dev/sda1                                1022M   7.1M   1015M   1% /boot/efi
/dev/mapper/rl_plantillarocky9-var        15G   1.4G    14G   9% /var
/dev/mapper/rl_plantillarocky9-home       15G  140M    15G   1% /home
10.43.103.136:/almacen                    24G   5.5G    19G  23% /almacen
tmpfs                                      1.2G  4.0K    1.2G   1% /run/user/1001
[estudiante@ING-PDGE11 ~]$ |

[estudiante@ING-PDGE11 ~]$ mkdir DanielG
[estudiante@ING-PDGE11 ~]$ ls
DanielG
[estudiante@ING-PDGE11 ~]$ cd DanielG/
[estudiante@ING-PDGE11 DanielG]$ ls
[estudiante@ING-PDGE11 DanielG]$ pwd
/home/estudiante/DanielG
[estudiante@ING-PDGE11 DanielG]$ mkdir fork
[estudiante@ING-PDGE11 DanielG]$ cd fork
[estudiante@ING-PDGE11 fork]$ nano cliente.c

[estudiante@ING-PDGE11 fork]$ nano servidor.c
[estudiante@ING-PDGE11 fork]$ ls
cliente.c  servidor.c
[estudiante@ING-PDGE11 fork]$ gcc cliente.c -o cliente
[estudiante@ING-PDGE11 fork]$ ./cliente
INFO.CLIENT: Enviar Mensajes, infinitamente, para terminar, ingresa "fin."
Envia fin
La cadena es: "fin" y su longitud es 3
Presiona enter
La cadena es: "fin" y su longitud es 3
Presiona enter
La cadena es: "fin" y su longitud es 3
[estudiante@ING-PDGE11 fork]$
```

- 6) Salir carpeta

```
$cd ..
```

```
La cadena es: "fin" y su longitud es 3
Frase: nd
La cadena es: "nd" y su longitud es 2
Frase: end
La cadena es: "end" y su longitud es 3
[estudiante@ING-PDGE11 fork]$ cd ..
[estudiante@ING-PDGE11 DanielG]$ ls
fork
```

- 7) Crear carpeta posix

```
$mkdir posix
```

```
$cd posix
```

- 8) \$nano prod_cons_posix.c

```
[estudiante@ING-PDGE11 DanielG]$ mkdir posix
[estudiante@ING-PDGE11 DanielG]$ ls
fork  posix
[estudiante@ING-PDGE11 DanielG]$ cd posix
[estudiante@ING-PDGE11 posix]$ nano prod_cons_posix.c
[estudiante@ING-PDGE11 posix]$ ls
prod_cons_posix.c
[estudiante@ING-PDGE11 posix]$ vi prod_cons_posix.c
[estudiante@ING-PDGE11 posix]$ |
```

Compilación de código

```
[estudiante@ING-PDGE11 posix]$ vi prod_cons_posix.c
[estudiante@ING-PDGE11 posix]$ gcc prod_cons_posix.c -o probando
[estudiante@ING-PDGE11 posix]$ ./probando
Soy productor -394267072 valor contador = 1
Soy productor -394267072 valor contador = 2
Soy productor -394267072 valor contador = 3
Soy productor -394267072 valor contador = 4
Soy productor -394267072 valor contador = 5
Soy productor -444623296 valor contador = 6
Soy productor -444623296 valor contador = 7
Soy productor -444623296 valor contador = 8
Soy productor -444623296 valor contador = 9
Soy productor -444623296 valor contador = 10
```

MAQUINA VIRTUAL 2 (DADA POR LA UNIVERSIDAD)

1. Con el comando Top encontramos diversos aspectos y lo que observamos es el número de procesadores que tiene esta máquina que en total serian 4. Cpus, también observamos cantidad de almacenamiento el cual es 11914.2 que sería en almacenamiento y una RAM de 4096 la cual es 4 RAM, por lo tanto observamos que esta maquina virtual de la universidad tiene mayor almacenamiento que la anterior.

```
estudiante@ing-gen129:~/Documents$ top
top - 10:10:05 up 70 days, 19:07, 3 users, load average: 0.06, 0.03, 0.00
Tasks: 379 total, 1 running, 378 sleeping, 0 stopped, 0 zombie
%Cpu0 :  0.3 us,  0.7 sy,  0.0 ni, 99.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.
%Cpu1 :  1.7 us,  1.7 sy,  0.0 ni, 96.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.
%Cpu2 :  1.0 us,  1.0 sy,  0.0 ni, 98.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.
%Cpu3 :  0.0 us,  0.3 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.
MiB Mem : 11914.2 total, 2508.8 free, 2694.1 used, 7138.7 buff/cache
MiB Swap:  4096.0 total, 4095.7 free,  0.3 used. 9220.1 avail Mem
```

2. Con el comando `lscpu` encontramos las especificaciones del hardware como lo es los sockets, los procesadores y su familia, la cantidad de hilos para cada procesador y la arquitectura de este, y la jerarquía en procesos que podemos observar y su nivel de ejecución del más cercano al más lejano, los cuales son, además también observamos un procesador diferente al observando sus modelos:

L1d 128 KIB (4 instancias)

L1i 128 KIB (4 instancias)

L2 4 MIB (4 instancias)

L3 66 MIB (4 instancias)

```
estudiante@ing-gen129:~/Documents$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Address sizes:          43 bits physical, 48 bits virtual
Byte Order:             Little Endian
CPU(s):                 4
On-line CPU(s) list:   0-3
Vendor ID:              GenuineIntel
Model name:             Intel(R) Xeon(R) Silver 4214 CPU @ 2.20GHz
CPU family:             6
Model:                  79
Thread(s) per core:     1
Core(s) per socket:     1
Socket(s):              4
Stepping:               0
BogoMIPS:               4389.68
Flags:                  fpu vme de pse tsc msr pae mce cx8 apic sep mtrr p
ge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss
yscall nx pdpe1gb rdtscp lm constant_tsc arch_pe
rfmon nopl xtopology tsc_reliable nonstop_tsc cpui
d tsc_known_freq pni pclmulqdq ssse3 fma cx16 pcid
sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_ti
mer aes xsave avx f16c rdrand hypervisor lahf_lm a
bm 3dnowprefetch cpuid_fault pti ssbd ibrs ibpb st
ibp fsgsbase tsc_adjust bmi1 avx2 smep bmi2 invpci
d rdtseed adx smap xsaveopt arat md_clear flush_lid
arch_capabilities

Virtualization features:
Hypervisor vendor:     VMware
Virtualization type:   full
Caches (sum of all):
L1d:                   128 KiB (4 instances)
L1i:                   128 KiB (4 instances)
L2:                    4 MiB (4 instances)
L3:                   66 MiB (4 instances)
NUMA:
NUMA node(s):          1
NUMA node0 CPU(s):     0-3
Vulnerabilities:
Gather data sampling:   Not affected
Itlb multihit:          KVM: Mitigation: VMX unsupported
Mltd:                   Mitigation; PTI Inversion
Mds:                   Mitigation; Clear CPU buffers; SMT Host state unk
```

3. Acá observamos con el comando (`df -h`) un almacenamiento distinto que se guarda en hardware y su repartición en cada fichero con el fin de que esta información no se perderá así se apague el hardware.

```
estudiante@ing-gen129:~/Documents$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           1.2G  2.3M  1.2G   1% /run
/dev/sda2       79G   16G   59G  22% /
tmpfs           5.9G   0    5.9G   0% /dev/shm
tmpfs           5.0M   0    5.0M   0% /run/lock
tmpfs           5.9G 220K  5.9G   1% /run/qemu
tmpfs           1.2G 104K  1.2G   1% /run/user/120
tmpfs           1.2G 124K  1.2G   1% /run/user/1000
estudiante@ing-gen129:~/Documents$
```

4. Gracias al siguiente comando (`uname`) y (`lsb_release -a`) observamos el sistema operativo que contiene la maquina virtual de la universidad el sistema operativo que

contiene es Linux, específicamente la versión de Ubuntu y su versión, mientras que la maquina virtual dada por el profesor observamos el mismo sistema operativo Linux, pero no es posible observar sus especificaciones con el mismo comando. Máquina Virtual Universidad.

```
estudiante@ing-gen129:~/Documents$ uname
Linux
estudiante@ing-gen129:~/Documents$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 24.04 LTS
Release:        24.04
Codename:       noble
estudiante@ing-gen129:~/Documents$
```

Máquina Virtual Profesor.

```
[estudiante@ING-PDGE11 DanielG]$ uname
Linux
[estudiante@ING-PDGE11 DanielG]$ lsb_release -a
-bash: lsb_release: command not found
[estudiante@ING-PDGE11 DanielG]$ lsb_release -a
```

Debido a esto nos toca con un comando distinto el cual es (cat /etc/os-release) que acá muestra detalladamente mas especificaciones.

Máquina Virtual Universidad.

```
estudiante@ing-gen129:~/Documents$ cat /etc/os-release
PRETTY_NAME="Ubuntu 24.04 LTS"
NAME="Ubuntu"
VERSION_ID="24.04"
VERSION="24.04 LTS (Noble Numbat)"
VERSION_CODENAME=noble
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=noble
LOGO=ubuntu-logo
estudiante@ing-gen129:~/Documents$
```

En esta observamos el sistema Operativo Linux/Ubuntu

Máquina Virtual Profesor.

```
[estudiante@ING-PDGE11 DanielG]$ cat /etc/os-release
NAME="Rocky Linux"
VERSION="9.4 (Blue Onyx)"
ID="rocky"
ID_LIKE="rhel centos fedora"
VERSION_ID="9.4"
PLATFORM_ID="platform:el9"
PRETTY_NAME="Rocky Linux 9.4 (Blue Onyx)"
ANSI_COLOR="0;32"
LOGO="fedora-logo-icon"
CPE_NAME="cpe:/o:rocky:rocky:9::baseos"
HOME_URL="https://rockylinux.org/"
BUG_REPORT_URL="https://bugs.rockylinux.org/"
SUPPORT_END="2032-05-31"
ROCKY_SUPPORT_PRODUCT="Rocky-Linux-9"
ROCKY_SUPPORT_PRODUCT_VERSION="9.4"
REDHAT_SUPPORT_PRODUCT="Rocky Linux"
REDHAT_SUPPORT_PRODUCT_VERSION="9.4"
[estudiante@ING-PDGE11 DanielG]$
```

En esta observamos un sistema operativo Linux/ Rocky Linux

En conclusión, la comparación entre estos dos sistemas de cómputo entre la máquina de la universidad y la maquina dada por el profesor, es mejor la máquina virtual de La universidad por sus componentes, como lo es el sistema operativo y los componentes del hardware por su almacenamiento.

ANALISIS HOJA DE EXCEL.

En las siguientes imágenes observaremos una tabla comparativa dentro dos sistemas de cómputo sobre mismos procesos en ejecución, esto con el fin de determinar que sistema de computo es mejor, para esto analizamos sus respectivas cifras con el fin de hacer la media y definir en los distintos procesos que maquina es mejor.

MAQUINA 1.

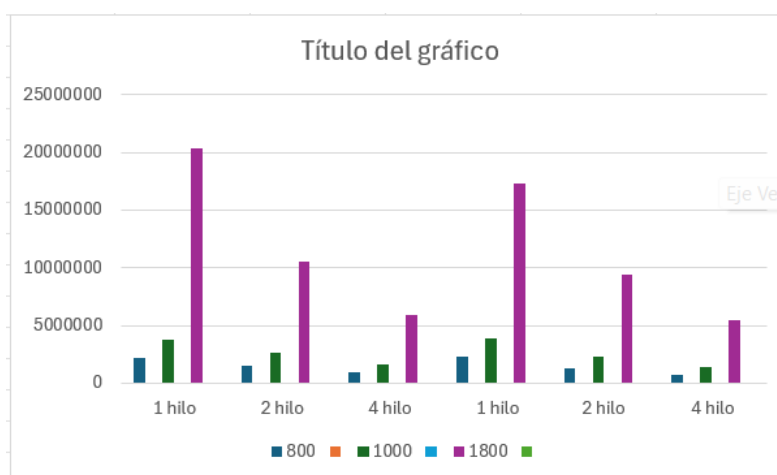
En la primera imagen observamos los tiempos de ejecución diferencio el método que usamos, a simple vista no observamos detalladamente que método es mejor ya son números grandes y distintos, por esta razón debemos realizar un análisis del promedio en que se encuentra el tiempo de ejecución y ver que método da mayor efectividad en comparación al otro.

Maquina 1 Profesor						
	mm_clasico			mm_transpuesta		
	1 hilo	2 hilo	4 hilos	1 hilo	2 hilo	4 hilo
8000	2843570	1617334	525821	2233327	1230309	722784
	2014016	1291262	1220810	2378350	1241068	471723
	2085604	1825330	700059	1826616	1064539	939055
	1925251	1315878	1238854	2781371	1621817	597097
1000	3806013	2797568	1536614	3865112	2272897	1209546
	4738240	2732366	1808662	3651963	2316322	1702593
	3661250	2781605	1676942	4335815	2418939	1471186
	2856590	2060726	1556320	3804680	2194843	1316347
1800	20476616	10287193	6079471	17958344	9104403	5718947
	20442643	11135569	5927981	17106969	9608053	5589830
	20199287	10482106	5329080	16814096	9051041	5060424
	20343528	10430227	6090299	17193141	9793828	5415064

Acá realizamos el promedio de cada ejemplo realizamos un promedio el los cuales son 800,1000 y 1800, esto con el fin de comparar la efectividad de cada caso son su respectivo método.

	Maquina Virtual 1					
	1 hilo	2 hilo	4 hilo	1 hilo	2 hilo	4 hilo
800	2217110	1512451	921386	2304916	1289433	682665
1000	3765523	2595566	1644635	3914393	2300750	1424918
1800	20365519	10583774	5856708	17268138	9389331	5446066

Posteriormente realizamos la grafica esto con el fin de que cualquier persona pueda observar el tiempo de ejecución en cada caso y método.



En el eje vertical observamos la cantidad de tiempo que tarda en ser procesada la información dada y en el eje horizontal observamos el caso y método que utilizamos, por ello concluimos que el método mas eficiente pese al mayor caso es el de mm_transpuesta ya que en la grafica observamos que el tiempo de ejecución es mucho menor que el de los demás.

MAQUINA 2.

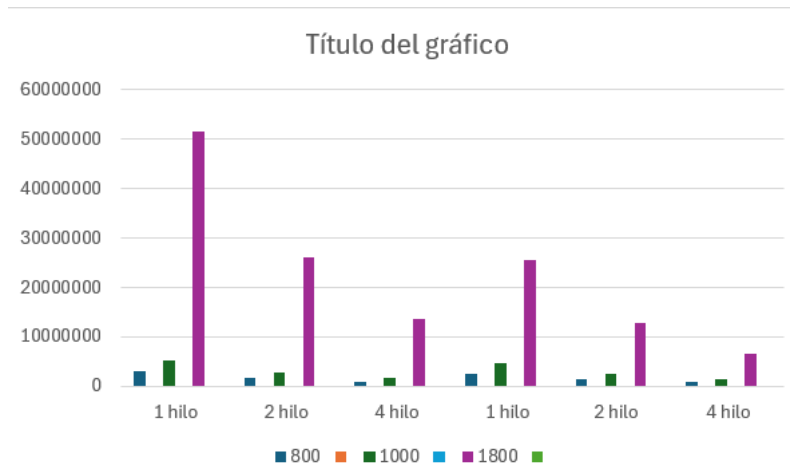
En la primera imagen observamos los tiempos de ejecución diferencio el método que usamos, a simple vista no observamos detalladamente que método es mejor ya son números grandes y distintos, por esta razón debemos realizar un análisis del promedio en que se encuentra el tiempo de ejecución y ver que método da mayor efectividad en comparación al otro.

Maquina 2 Universidad					
mm_clasico			mm_transpuesta		
1 hilo	2 hilo	4 hilo	1 hilo	2 hilo	4 hilo
3081953	1640864	960112	2975645	1473403	753100
2758369	2114133	719520	2445461	1328402	1386679
3458430	1419311	1385981	2733015	1603325	817146
2732432	1877530	1000052	2486512	1123608	993753
5228241	3009787	2241276	4678368	2614962	1038266
4828789	3263250	1392057	4692151	2409780	1022719
5540002	2585392	1310839	4892729	2404344	1990718
5262958	2628228	1580421	4177619	2344297	1832650
51985480	25779584	13292374	25468278	13071016	6912930
51242161	25583582	13905907	25341523	12413996	6502448
50950624	26783345	13974946	25575425	13205931	6761513
52070575	25868575	13613389	25434237	12742557	5953456

Acá realizamos el promedio de cada ejemplo realizamos un promedio en los cuales son 800,1000 y 1800, esto con el fin de comparar la efectividad de cada caso son su respectivo método.

Maquina Virtual 2						
	1 hilo	2 hilo	4 hilo	1 hilo	2 hilo	4 hilo
800	3007796	1762960	1016416	2660158	1382185	987670
1000	5214998	2871664	1631148	4610217	2443346	1471088
1800	51562210	26003772	13696654	25454866	12858375	6532587

Posteriormente realizamos la gráfica esto con el fin de que cualquier persona pueda observar el tiempo de ejecución en cada caso y método.



En el eje vertical observamos la cantidad de tiempo que tarda en ser procesada la información dada y en el eje horizontal observamos el caso y método que utilizamos, por ello concluimos que el método más eficiente pese al mayor caso es el de mm_transpuesta ya que en la gráfica observamos que el tiempo de ejecución es mucho menor que el de los demás.

CONCLUSION:

En conclusión comparando ambas graficas observamos que en tiempo de ejecución con el mismo ingreso de datos es mejor la el sistema de cómputo de máquina virtual N2.