Universidad de Huelva

ETSI
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

Escuela Técnica Superior de Ingeniería
Universidad de Huelva

Memoria de Prácticas

ADMINISTRACIÓN DE SERVIDORES

Grado en Ingeniería Informática

Autor: Rafael Mesa Nombela

19 de octubre de 2023

Resumen

Memoria de las prácticas de Administracion De Servidores

Palabras clave: integer, blandit, pharetra, urna, id.

Índice general

Capítulo 1

Practica 1 Monitorización

1.1. Introducción

La empresa ACME desea instalar un nuevo servidor de base de datos, dicho servidor deberá tener la capacidad suficiente para soportar las consultas realizas por los clientes. Para ello, se propone la creación de un contenedor LXD basado en Centos 7.x en el que instalará una base de datos MariaDB. Dicho contenedor tendrá los siguientes recursos:

- 512 megas de RAM
- 2 CPU
- 8 GB disco duro

Una vez creado el sistema, y que tengamos instalado MariaDB en Centos 7, procederemos a hacer las pruebas de estrés oportunas mediante la herramienta sysbench, siguiendo lo indicado en How to Benchmark Performance of MySQL & MariaDB Using SysBench. Es recomendable que sysbench se instale en un contenedor distinto al usado para instalar MariaDB.

1.1.1. Objetivos

Por un lado se aprenderá a crear Containers, y posteriormente se observará, con diferentes comandos, el comportamiento de una máquina cuando se le somete a un cierto grado de estrés.

1.2. Utilidades

– descargar ssh: yum search [nombre del paquete] ->(yum search openssh-server)

– inicio de ssh: service sshd start

– ip mariadb: 192.168.50.185

– ip sysbench: 192.168.50.214

1.3. Procedimiento

1.3.1. Estudio de las capacidades de un servidor

1. *Obtenga la información referente a la capacidad de el o los procesadores (puede consultar el fichero /proc/cpuinfo).*
   Mirando el fichero /proc/cpuinfo se observa que la maquina tiene 2 procesadores y 4 nucleos cada uno de 2.1 GHz

2. *Los comando free y vmstat pueden servir para monitorizar el uso de la memoria. Use ambos comando para obtener la información referente a la capacidad de la memoria (total de la memoria, memoria virtual usada, espacio destinado a las cachés y buffers, etc).*
   free: free -m free -h ->ajusta unidades automaticamente (h de human, osea datos para que los entienda el humano, esto sera en comandos que muestren tamaños)

```
[root@Mariadb proc]# free -h
              total        used        free      shared  buff/cache   available
Mem:           512M         26M        201M        8.1M        283M        485M
Swap:          1.0G          0B        1.0G
```

   vmstat: vmstat 3 4 ->ejecuta cada 3 segundos 4 veces. vmstat -s ->info mas detallada.

```
[root@Mariadb proc]# vmstat 3 4
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 206780      0 290812    0    0    39     7   10   21  0  0 100  0  0
 0  0      0 206888      0 290812    0    0   256     7  189  283  0  0 100  0  0
 0  0      0 206888      0 290812    0    0     0    15  124  199  0  0 100  0  0
 0  0      0 206888      0 290812    0    0     0     7  107  192  0  0 100  0  0
```

```
[root@Mariadb proc]# vmstat -s
       524288 K total memory
        26696 K used memory
       115104 K active memory
       185460 K inactive memory
       206780 K free memory
            0 K buffer memory
       290812 K swap cache
      1048576 K total swap
            0 K used swap
      1048576 K free swap
        13478 non-nice user cpu ticks
            0 nice user cpu ticks
            0 system cpu ticks
    174060729 idle cpu ticks
            0 IO-wait cpu ticks
            0 IRQ cpu ticks
            0 softirq cpu ticks
            0 stolen cpu ticks
     67296558 pages paged in
     12078422 pages paged out
            0 pages swapped in
            0 pages swapped out
    102905903 interrupts
    165959074 CPU context switches
   1695311528 boot time
      2407309 forks
```

3. *El comando df muestra el espacio libre y ocupado en los distintos sistemas de ficheros que tiene montados el sistema, obtenga la información referente a la capacidad*

```
[root@Mariadb proc]# df -h
Filesystem                      Size  Used Avail Use% Mounted on
/dev/mapper/pve-vm--100--disk--0  7.9G  659M  6.8G   9% /
none                            492K  4.0K  488K   1% /dev
udev                            3.9G     0  3.9G   0% /dev/tty
tmpfs                           3.9G     0  3.9G   0% /dev/shm
tmpfs                           3.9G  8.2M  3.9G   1% /run
tmpfs                           3.9G     0  3.9G   0% /sys/fs/cgroup
tmpfs                            52M     0   52M   0% /run/user/0
```

4. *Los comandos lsblk e iostat muestran información sobre los distintos sistemas de ficheros montados. En lo que se refiere al nombre del dispositivo, ¿aprecia alguna diferencia?*
   no tengo el comando por lo que tengo que instalar el paquete que lo contenga,
   rpm -q sysstat para ver si tengo instalado el paquete (dentro esta el iostat)
   yum search sysstat
   yum install sysstat
   iostat: iostat -c ->solo info CPU, iostat -c 1 3 ->info cada segundo 3 veces

```
[root@Mariadb /]# iostat -c
Linux 5.4.106-1-pve (Mariadb)   10/01/23        _x86_64_        (4 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.01    0.00    0.00    0.00    0.00   99.99
```

   iostat te de info de los dm, aparte tambien te muestra info del tiempo en % de el modo en el que se encuentra el sistema en es momento, lsblk muestra las particiones (si las hubiera de cada sf) col 1->tipo nucleo, verion, host, fecha actual, arquitectura, nucleos/cpu
   col 2 ->porcentajes de tiempo: cpu modo usuario, cambio prioridad de procesos, modo sistema, esperando io, esperando a que el hipervisor gestione solicitudes a la CPU virtual.

5. *5. Obtenga la información referente a la capacidad de la red.*
   curl ifconfig.me ->devuelve tu ip publica
   ss -tuln ->muestra las conexiones TCP y UDP en escucha.
   yum search netstat
   yum install dstat ->este paquete que aparece el primero no contiene el comando pero el comando dstat es interesante y muy ligero.
   yum install net-tools
   netstat: -r ->tabla rutas, -t ->tcp, -u ->udp, -tua (todos lospaquetes tcp y udp)

1.3.2. Monitorizacion del servidor

1. *Determine los procesos activos y la relación que existe entre ellos. Compruebe que el servidor Mariadb esté activo.*
   systemctl start mariadb.service
   para saber si esta efectivamente activo el servicio:

   - systemctl –help #para saber la opcion
   - systemctl is-active mariadb.service #devuelve active si esta activo

– systemctl status mariadb.service #tambien se puede usar

2. *Determine el consumo de los diferentes recursos identificando los procesos que más consuman.*
yum install iotop # para ver procesos io ->el comando iotop no me muestra correctamente la informacion por pantalla
top: -p [proceso] ->seguimiento especifico de un proceso, -d [segundos] ->actualizar cada n segundos.
q para salir, M para ordenar los procesos que mas memoria consumen (en vez de por consumo de CPU).

```
top - 14:31:11 up 10 days, 21:46,  2 users,  load average: 0.04, 0.01, 0.02
Tasks:  17 total,   1 running,  16 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :   524288 total,    35664 free,   111804 used,   376820 buff/cache
KiB Swap:  1048576 total,  1048576 free,        0 used.   412484 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
    1 root      20   0   43464   5352   4100 S   0.0  1.0   0:09.87 systemd
   42 root      20   0   39092   5968   5688 S   0.0  1.1   0:05.24 systemd-jo+
   61 dbus      20   0   58128   4296   3712 S   0.0  0.8   0:02.03 dbus-daemon
   62 root      20   0   26652   3100   2524 S   0.0  0.6   0:05.01 systemd-lo+
   76 root      20   0   90964   4592   3952 S   0.0  0.9   0:00.02 login
   77 root      20   0   22740   2748   2100 S   0.0  0.5   0:04.10 crond
   78 root      20   0    6524   1724   1596 S   0.0  0.3   0:00.00 agetty
   79 root      20   0    6524   1720   1596 S   0.0  0.3   0:00.00 agetty
  248 root      20   0  102904   4664   2600 S   0.0  0.9   0:01.00 dhclient
  309 root      20   0  218556   6328   5644 S   0.0  1.2   1:31.79 rsyslogd
  314 root      20   0   11832   3048   2632 S   0.0  0.6   0:00.06 bash
 4881 root      20   0  112928   7512   6488 S   0.0  1.4   0:00.01 sshd
 8205 mysql     20   0    9708   2756   2468 S   0.0  0.5   0:00.01 mysqld_safe
 8369 mysql     20   0  969004  85828  13432 S   0.0 16.4   0:02.39 mysqld
 8416 root      20   0  155352   9700   8340 S   0.0  1.9   0:00.26 sshd
 8418 root      20   0   11836   3116   2716 S   0.0  0.6   0:00.09 bash
 8523 root      20   0   56232   3800   3280 R   0.0  0.7   0:00.00 top
```

Aqui apreciamos que no hay ningun proceso consumiendo cpu en este instante

```
top - 14:37:05 up 10 days, 21:52,  2 users,  load average: 0.07, 0.12, 0.06
Tasks:   1 total,   0 running,   1 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :   524288 total,    35816 free,   111652 used,   376820 buff/cache
KiB Swap:  1048576 total,  1048576 free,        0 used.   412636 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 8369 mysql     20   0  969004  85828  13432 S   0.0 16.4   0:02.76 mysqld
```

nos centramos en el que mas memoria consume que es un proceso de mariadb.
yum search htop no me encuentra nada ->hay que instalarse el paquete epel-release

3. *4.Verifique si existen clientes de red conectados.*
who: who -a todas las sesiones (el + indica las sesiones activas por ese usuario).
w

```
[root@Mariadb ~]# who
root     tty1         Sep 21 16:45
root     pts/3        Oct  2 14:03 (10.8.2.13)
[root@Mariadb ~]# w
 14:57:55 up 10 days, 22:13,  2 users,  load average: 0.05, 0.09, 0.08
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
root     tty1                      21Sep23  1:20m  0.06s  0.06s -bash
root     pts/3    10.8.2.13        14:03    0.00s  0.22s  0.00s w
[root@Mariadb ~]# who -a
          system boot  Sep 21 16:44
root     + tty1         Sep 21 16:45 01:24          76
LOGIN      console      Sep 21 16:44                79 id=cons
LOGIN      tty2         Sep 21 16:44                78 id=tty2
           run-level 5  Sep 21 16:44
           pts/2        Oct  2 14:14              8041 id=ts/2  term=0 exit=0
root     + pts/3        Oct  2 14:03    .         8418 (10.8.2.13)
```

muestra que hay una terminal remota (pts/3) activa.
netstat -tua

```
[root@Mariadb ~]# netstat -tua
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:mysql           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp        0    208 Mariadb:ssh             10.8.2.13:53989         ESTABLISHED
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
```

4. *Ejecute el comando iostat con la opción -c para comprobar el estado de uso de la CPU. Haga la monitorización a espacios regulares de 2 segundos durante 20 segundos. El uso de la CPU, ¿sufre cambios significativos?*
   iostat -c 2 20
   ningún cambio significativo

5. *Mediante la opción -d del comando iostat, verifique a periodos regulares de 2 segundos el estado de las operaciones de entrada y salida.*
   iostat -d 2 10, la io esta bastante tranquila.

6. *7.Use el comando w, para obtener la carga de la CPU en los últimos 1, 5 y 15 minutos. ¿Se encuentra el sistema sobrecargado?*
   w

```
[root@Mariadb ~]# w
 19:04:30 up 11 days,  2:19,  2 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
root     tty1                      21Sep23 11:42   0.06s  0.06s -bash
root     pts/2    10.8.2.13        18:53    2.00s  0.03s  0.00s w
[root@Mariadb ~]#
```

   el sistema esta en reposo

7. *El comando iostat no devuelve información sobre la carga específica de una CPU, para estos casos usamos el comando mpstat. Compare la salida de ambos comandos.*
   iostat no esta orientado a sistemas SMP por lo que devuelve una media de los 4 nucleos, mientras que mpstat devuelve una fila para cada nucleo, por lo que es mas especifico.
   El sistema tiene dos procesadores de 4 nucleos cada uno, ¿ porque solo salen datos de uno de los procesadores? ->Los comandos son muy antiguos por lo que estan pensados solo para un procesador, usar el htop

8. *La utilidad sar (system activity report"), recopila información de distintos parámetros del sistema. Esta aplicación incluye dos shell scripts. El primer script, sa1, recopila datos de forma regular, mientras que el script sa2 se utiliza para crear los informes resumidos (uno por día en /var/log/sa/sarDD. Ambos scripts se ejecutan usando cron. Si queremos obtener los datos en tiempo real podemos invocar directamente al comando sar. Por ejemplo, si queremos recopilar la información de la CPU cada 2 segundos, ejecutaremos:*

```
[root@Mariadb ~]# sar -u 2
Linux 5.4.106-1-pve (Mariadb)   10/02/23        _x86_64_        (4 CPU)

19:42:37        CPU     %user     %nice   %system   %iowait    %steal     %idle
19:42:39        all      0.00      0.00      0.00      0.00      0.00    100.00
19:42:41        all      0.00      0.00      0.00      0.00      0.00    100.00
19:42:43        all      0.25      0.00      0.00      0.00      0.00     99.75
19:42:45        all      0.00      0.00      0.00      0.00      0.00    100.00
Ac
```

1.3.3. Monitorización del servidor ante situaciones de estrés

En el apartado anterior hemos tomado una serie de medidas que, en la mayoría de los casos, nos indicarán que el sistema está en reposo. Para simular una situación de estrés, emplearemos sysbench para efectuar una serie de test de rendimiento sobre nuestra base de datos.

1. *Cree el esquema de base de datos sbtest y dótelo de 5 tablas con 50000 filas cada una.*
en la db de mariaDB:
CREATE DATABASE sbtest;

```
[root@Mariadb ~]# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 27
Server version: 5.5.68-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use sbtest
ERROR 1049 (42000): Unknown database 'sbtest'
MariaDB [(none)]> CREATE DATABASE sbtest
    -> ;
Query OK, 1 row affected (0.01 sec)

MariaDB [(none)]>
MariaDB [(none)]>
```

Use sbtest para saber si se ha creado la tabla

```
MariaDB [(none)]> use sbtest
Reading table information for
You can turn off this feature

Database changed
MariaDB [sbtest]>
```

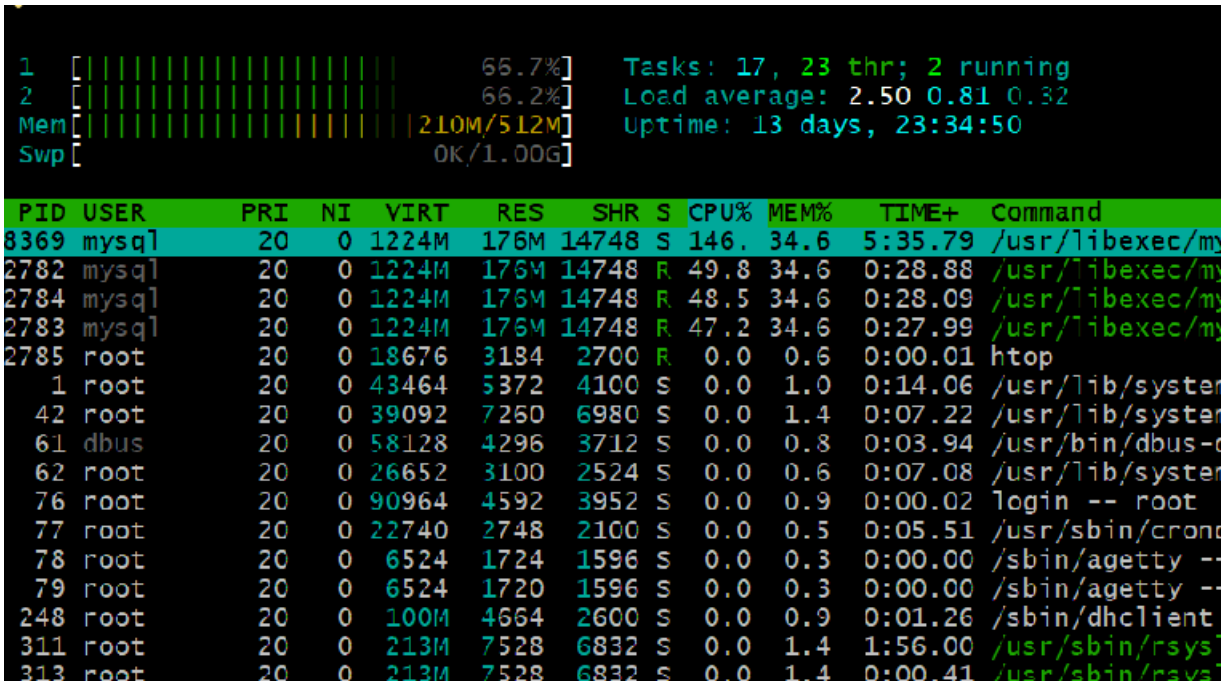
salimos y le damos privilegios al root con la ip del CT de sysbench
GRANT ALL PRIVILEGES ON *.* TO 'root'@'192.168.50.214' IDENTIFIED BY ' [contraseña]' ;
use sbtest para saber si se ha creado la tabla
(TEST DE LECTURA)
sysbench /usr/share/sysbench/oltp_read_only.lua –threads=4 –mysql-host=192.168.50.185 –mysql-user=root –mysql-password=[contraseñaCTmariaDB] –mysql-port=3306 –tables=5 –table-size=50000 prepare

```
[root@Sysbench ~]# sysbench /usr/share/sysbench/oltp_read_only.lua --thread
-mysql-host=192.168.50.185 --mysql-user=root --mysql-password=pececitos --m
ort=3306 --tables=5 --table-size=50000 prepare
ysbench 1.0.17 (using system LuaJIT 2.0.4)

nitializing worker threads...

reating table 'sbtest2'...Creating table 'sbtest1'...
reating table 'sbtest3'...

reating table 'sbtest4'...
nserting 50000 records into 'sbtest1'
nserting 50000 records into 'sbtest2'
nserting 50000 records into 'sbtest3'
nserting 50000 records into 'sbtest4'
```

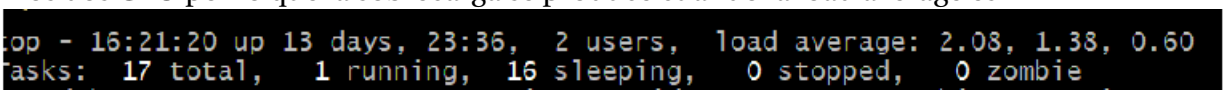
nos metemos en mariadb para comprobar que se han creado las tablas: show tables

```
    -> ;
ERROR 1046 (3D000): No database selected
MariaDB [(none)]> USE sbtest
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [sbtest]> show tables
    -> ;
+------------------+
| Tables_in_sbtest |
+------------------+
| sbtest1          |
| sbtest2          |
| sbtest3          |
| sbtest4          |
| sbtest5          |
+------------------+
5 rows in set (0.00 sec)

MariaDB [sbtest]>
```

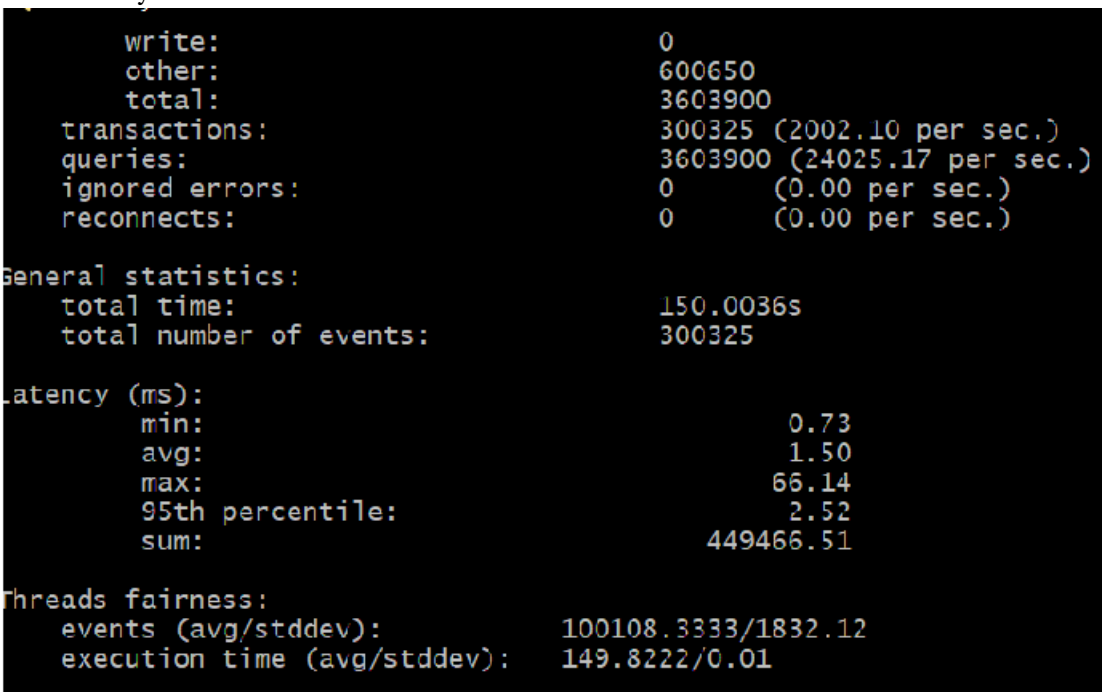
hacemos el test de lectura:
sysbench /usr/share/sysbench/oltp_read_only.lua –threads=3 –events=0 –time=150 –mysql-host=192.168.50.185 –mysql-user=root –mysql-password=pececitos –mysql-port=3306 –tables=5 –table-size=50000 –range_selects=off –db-ps-mode=disable –report-interval=1 run
MIENTRAS SE ESTA PRODUCIENO EL TEST DE LECTURA



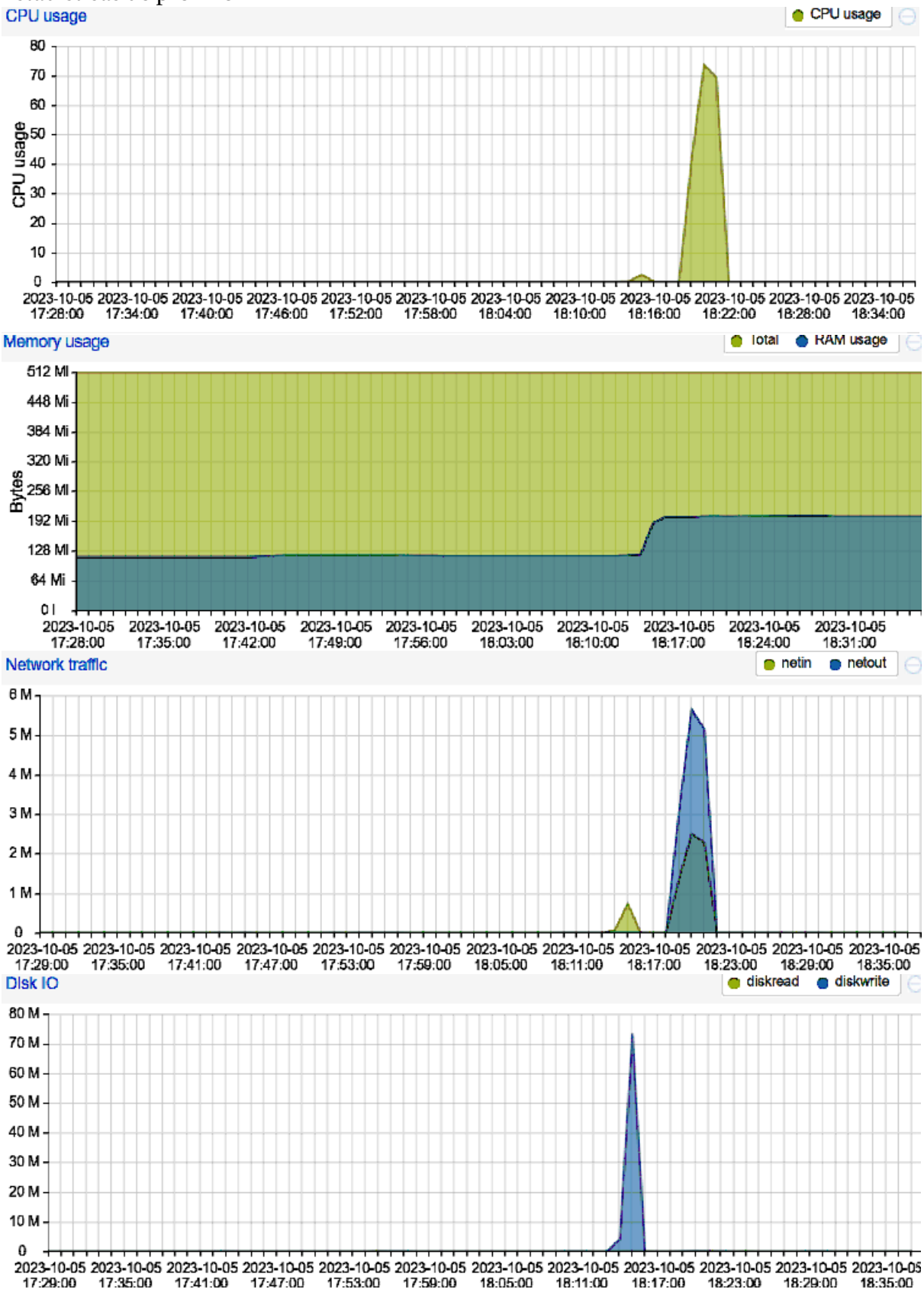
En load average podemos apreciar que hay una sobrecarga en la CPU, ya que tenemos dos CPU por lo que la sobrecarga se produce cuando la load average es >2



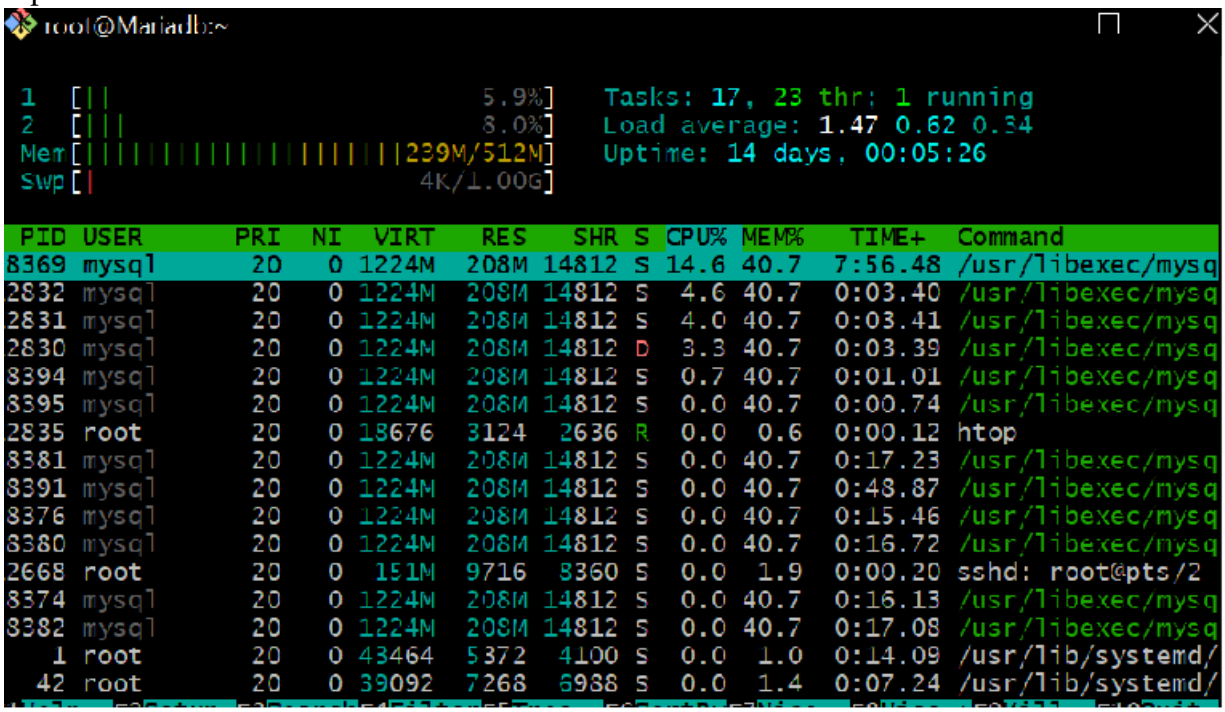
resumen sysbench



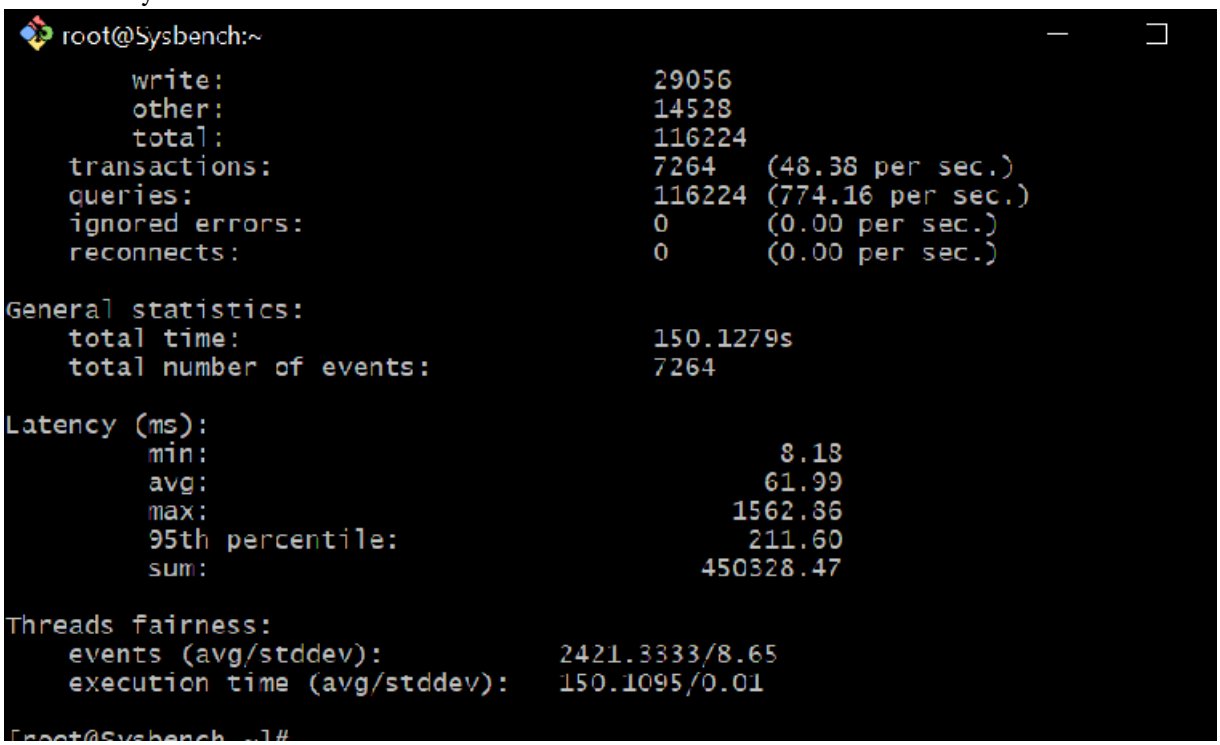
Estadisticas de proxmox

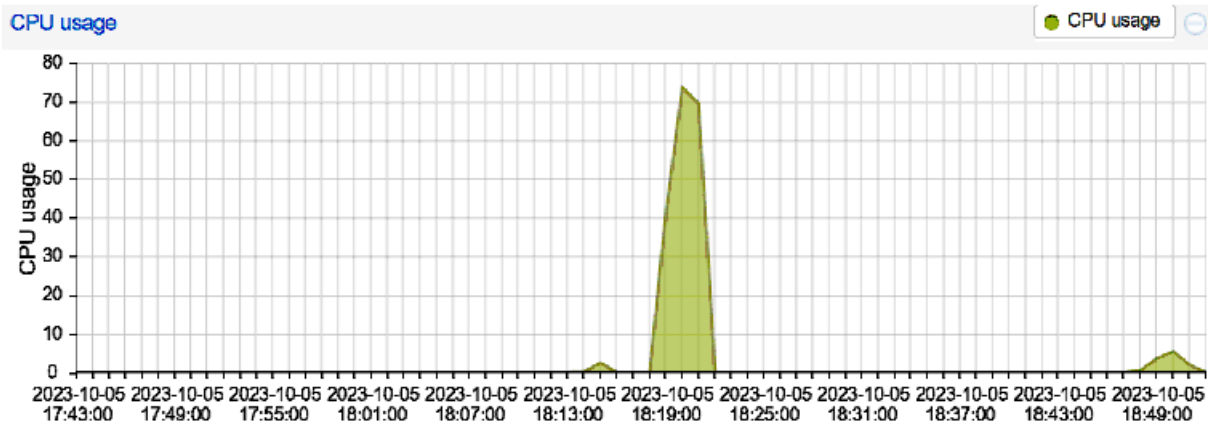


(TEST DE ESCRITURA) /usr/share/sysbench/oltp_read_write.lua
sysbench /usr/share/sysbench/oltp_read_write.lua –threads=3 –events=0 –time=150 –mysql-host=192.168.50.185 –mysql-user=root –mysql-password=pececitos –mysql-port=3306 –tables=5 –table-size=50000 –range_selects=off –db-ps-mode=disable –report-interval=1 run

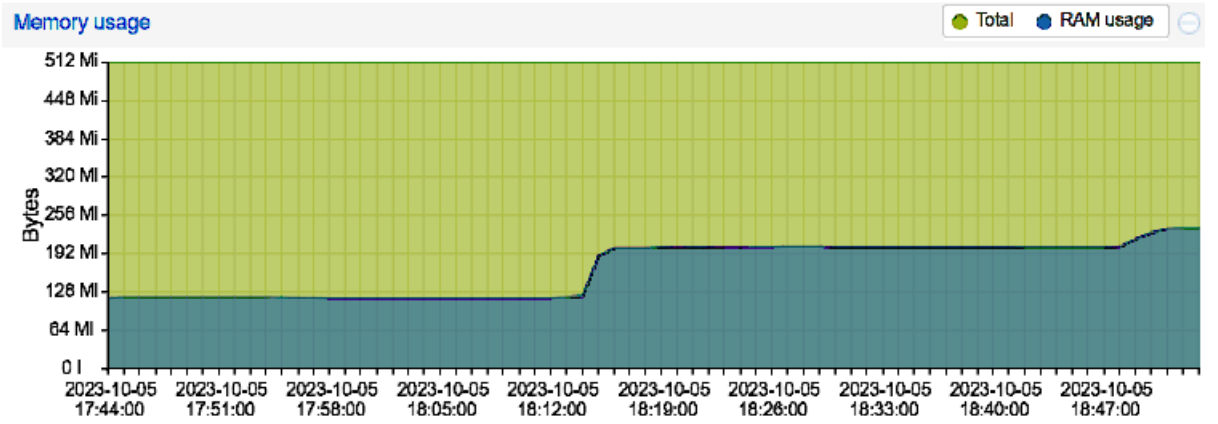


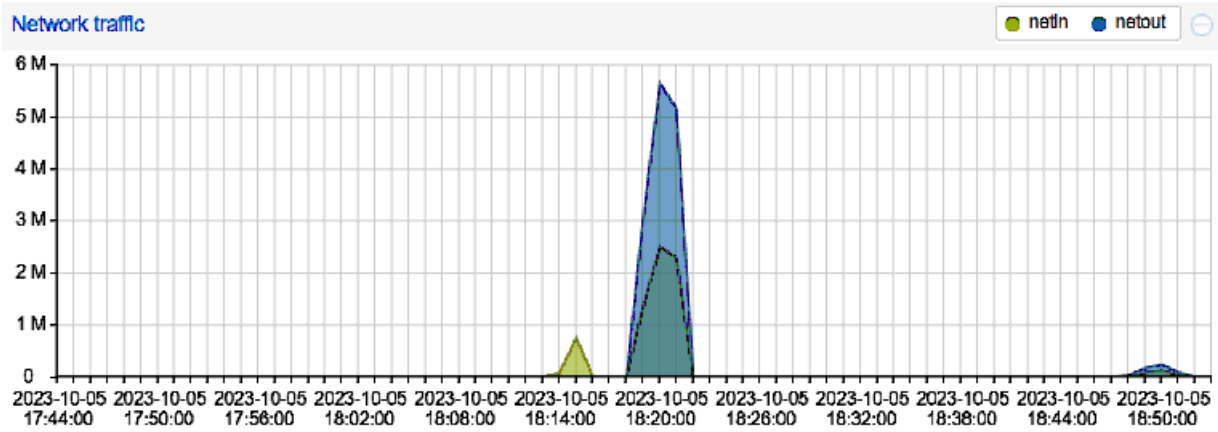
en este test la CPU no se ha sobrecargado
resumen sysbench:

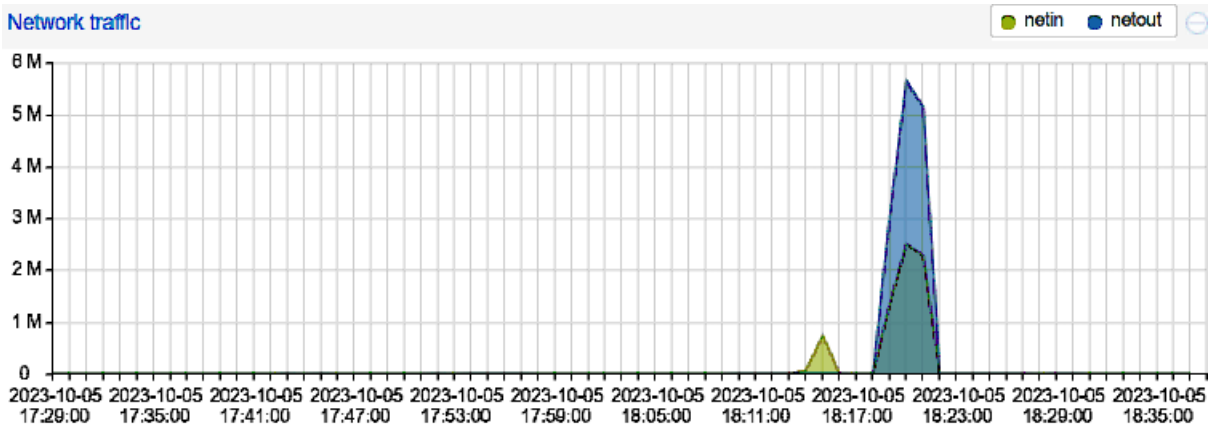




Se aprecia que el segundo test es bastante menos demandante. Esto se puede deber a dos cosas, por una parte las tablas seguramente estén guardadas en la caché de cuando se ejecutó el primer test, y el segundo motivo es que las lacturas son mucho más demandantes que las escrituras, ya que escribir será un INSERT y leer será un SELECT *.







1.4. Conclusión

Ha sido una buena primera aproximación al entorno de proxmox, así como la creación de Containers han facilitado y agilitado la creación de VM.
En la parte de monitorización se ha aprendido a usar una herramienta muy interesante como es *sysbench* si posteriormente necesitamos poner a prueba nuestro sistema.