

Actividad 2: Tuning del Sistema Operativo

Cristóbal Suárez Abad

Administración de Sistemas Gestores de Bases de Datos

Configurar el kernel de Linux para que soporte una carga de trabajo intensiva de base de datos.

Parte 1: Introducción Teórica

1. El concepto de "Swappiness"

Imagina que la memoria RAM es un **escritorio de trabajo** y el Disco Duro es un **archivador en el sótano**.

- **RAM:** Es rápida, instantánea. Aquí viven los datos que MySQL está usando ahora (Buffer Pool).
- **Swap (Intercambio):** Es una extensión de la RAM en el disco duro. Es miles de veces más lenta.

¿Qué es el valor `vm.swappiness`? Es un número del **0 al 100** que le dice al kernel de Linux cuán "ansioso" debe estar por mover cosas de la RAM al Swap.

- **60 (Por defecto en Linux):** Equilibrado para uso de escritorio. Si la RAM se llena un poco, empieza a mover cosas al disco.
- **Problema para BD:** Si Linux mueve la memoria de MySQL al disco (Swap), cada consulta tendrá que ir al "sótano" a buscar datos. El rendimiento caerá en picado.
- **Objetivo:** Queremos un valor cercano a **0** (ej. 1 o 10) para decirle a Linux: "*No toques el disco a menos que la RAM esté totalmente llena y sea una emergencia*".

2. Los "File Descriptors" (Límites de archivos)

En Linux, **todo es un archivo**: una conexión de red es un archivo, una tabla es un archivo, un log es un archivo.

- MySQL necesita abrir muchos archivos simultáneamente (uno por cada conexión de cliente + múltiples archivos por cada tabla).
 - **El límite por defecto:** Linux suele traer un límite de **1024** archivos abiertos por usuario.
 - Si tienes 200 conexiones y consultas muchas tablas, MySQL chocará con este techo y lanzará el error: `Too many open files`. El servicio se detendrá o rechazará conexiones.
-

Parte 2: Práctica Guiada

Paso A: Controlando el Swap

1. Diagnóstico Actual Comprueben el valor actual.

Bash

```
cat /proc/sys/vm/swappiness
```

```
# O usando sysctl
```

```
sysctl vm.swappiness
```

```
root@ubuntumysqlsuarez:/home/cristobal# cat /proc/sys/vm/swappiness
60
root@ubuntumysqlsuarez:/home/cristobal# sysctl vm.swappiness
vm.swappiness = 60
root@ubuntumysqlsuarez:/home/cristobal# |
```

2. El Cambio "En Caliente" (Temporal) Vamos a cambiarlo sin reiniciar. Bajarlo a 10.

Bash

```
sudo sysctl -w vm.swappiness=10
```

```
root@ubuntumysqlsuarez:/home/cristobal# sudo sysctl -w vm.swappiness=10
vm.swappiness = 10
root@ubuntumysqlsuarez:/home/cristobal# |
```

Si reiniciamos la máquina ahora, ¿qué valor tendrá?"

Vuelve a ser **60**.

```
root@ubuntumysqlsuarez:/home/cristobal# cat /proc/sys/vm/swappiness
60
root@ubuntumysqlsuarez:/home/cristobal# sysctl vm.swappiness
vm.swappiness = 60
root@ubuntumysqlsuarez:/home/cristobal# |
```

3. El Cambio Persistente (Definitivo) Para que el cambio sobreviva a un reinicio, debes editar el archivo de configuración del sistema.

Bash

```
sudo nano /etc/sysctl.conf
```

Añadir al final del archivo:

Ini, TOML

```
vm.swappiness = 10
```

```
# Modificación del valor de Swappiness  
vm.swappiness = 10
```

Guardar (**Ctrl+O**) y salir. Para aplicar los cambios de este archivo sin reiniciar:

Bash

```
sudo sysctl -p
```

```
root@ubuntumysqlsuarez:/home/cristobal# sudo nano /etc/sysctl.conf  
root@ubuntumysqlsuarez:/home/cristobal# sudo sysctl -p  
vm.swappiness = 10  
root@ubuntumysqlsuarez:/home/cristobal# |
```

Paso B: Rompiendo los límites (Límites de Archivos)

1. Ver el límite actual Entrar como el usuario que ejecuta la base de datos (o simularlo desde root).

Bash

```
ulimit -n
```

```
root@ubuntumysqlsuarez:/home/cristobal# ulimit -n  
1024  
root@ubuntumysqlsuarez:/home/cristobal# |
```

Resultado esperado: 1024 (Muy poco para un servidor de producción).

2. Configuración de Seguridad Debemos editar el archivo `/etc/security/limits.conf`. Este archivo controla los recursos para los usuarios.

Bash

```
sudo nano /etc/security/limits.conf
```

3. Aplicar la nueva configuración Añadir las siguientes líneas al final, justo antes de `# End of file`. `mysql` es el usuario del sistema

Ini, TOML

```
mysql    soft  nofile 65535  
mysql    hard  nofile 65535
```

```
mysql      soft      nofile  65535  
mysql      hard      nofile  65535  
|  
# End of file
```

- **Explicación Soft/Hard:**

- **Soft:** El límite de advertencia (se puede superar momentáneamente o cambiar por el usuario hasta llegar al hard).
- **Hard:** El límite estricto que no se puede cruzar.
- **65535:** Es un valor estándar alto (2 elevado a 16 menos 1).

4. Verificación (El truco) Este cambio **NO** afecta a los procesos que ya están corriendo. Hay que reiniciar el servicio MySQL y, a veces, cerrar y abrir la sesión de usuario.

Bash

Reiniciar el servicio

```
sudo systemctl restart mysql
```

```
root@ubuntumysqlsuarez:/home/cristobal# sudo systemctl restart mysql
root@ubuntumysqlsuarez:/home/cristobal# sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: active (running) since Fri 2026-01-09 09:22:01 UTC; 4s ago
     Process: 3371 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 3380 (mysqld)
      Status: "Server is operational"
     Tasks: 38 (limit: 4605)
    Memory: 365.9M (peak: 380.2M)
       CPU: 1.911s
      CGroup: /system.slice/mysql.service
              └─3380 /usr/sbin/mysqld

ene 09 09:21:59 ubuntumysqlsuarez systemd[1]: Starting mysql.service - MySQL Community Server...
ene 09 09:22:01 ubuntumysqlsuarez systemd[1]: Started mysql.service - MySQL Community Server.
root@ubuntumysqlsuarez:/home/cristobal# |
```

Verificar los límites del proceso MySQL en ejecución

Primero buscamos el PID (Process ID) de mysql

```
pidof mysqld
```

```
root@ubuntumysqlsuarez:/home/cristobal# pidof mysqld
3380
root@ubuntumysqlsuarez:/home/cristobal# |
```

Supongamos que el PID es 1234. Miramos sus límites:

```
cat /proc/1234/limits
```

```
cat /proc/4124/limits
```

```
root@ubuntumysqlsuarez:/home/cristobal# cat /proc/4124/limits
Limit           Soft Limit      Hard Limit      Units
Max cpu time    unlimited      unlimited      seconds
Max file size   unlimited      unlimited      bytes
Max data size   unlimited      unlimited      bytes
Max stack size  8388608       unlimited      bytes
Max core file size 0            unlimited      bytes
Max resident set unlimited      unlimited      bytes
Max processes   15352         15352        processes
Max open files  10000         10000        files
Max locked memory 8388608       8388608       bytes
Max address space unlimited      unlimited      bytes
Max file locks  unlimited      unlimited      locks
Max pending signals 15352       15352        signals
Max msgqueue size 819200       819200       bytes
Max nice priority 0             0             us
Max realtime priority 0          0             us
Max realtime timeout unlimited      unlimited      us
root@ubuntumysqlsuarez:/home/cristobal# |
```

Buscar la fila "**Max open files**". Debería decir ahora **65535** en lugar de **1024**.

Por lo visto no pasa de 10000 por los límites del hardware de la VM.

Paso C: Monitorizando

Ejecuta `htop` en una terminal mientras hacen una carga masiva de datos. Si el Swappiness es alto y la memoria se llena, verán cómo la barra de "**Swp**" en `htop` empieza a llenarse y la barra de "**Load Average**" se dispara, haciendo que el sistema se vuelva lento.

La barra de Swap empieza a llenarse, no mucho, pero algo se llena.

