

## LPIC-2 / Examen 206 - Mantenimiento del Sistema

### 206.3 Mejorar el rendimiento del sistema

#### Teoría

Mejorar el rendimiento del sistema implica identificar qué recurso (CPU, memoria, E/S de disco, red) es el cuello de botella actual y aplicar ajustes para optimizar su uso o aumentar su capacidad. Es un proceso iterativo basado en la monitorización y el análisis (visto en 200.1).

#### Metodología de Mejora del Rendimiento:

1. **Monitorizar:** Usa herramientas como `top`, `htop`, `vmstat`, `iostat`, `mpstat`, `ss`, `sar` para medir el uso de recursos en condiciones de carga.
2. **Identificar el Cuello de Botella:** Analiza los datos de monitorización para determinar qué recurso está limitando el rendimiento. ¿Es la CPU la que está al 100%? ¿La memoria está agotada y se usa mucha swap? ¿El disco está al 100% de utilización con alta latencia (`%util`, `await`)? ¿La red está saturada o con errores?
3. **Investigar la Causa:** Una vez identificado el recurso, determina qué procesos o configuraciones están causando el alto uso (usa `top/htop` ordenado por el recurso problemático, `iostat` para E/S de disco por proceso, `tcpdump` para red por proceso/conexión).
4. **Ajustar (Tune):** Aplica cambios en la configuración del sistema, la configuración de la aplicación o la asignación de recursos para aliviar el cuello de botella.
5. **Verificar:** Monitoriza de nuevo para ver si los ajustes tuvieron el efecto deseado y si el cuello de botella se movió a otro recurso.

#### Áreas de Ajuste de Rendimiento y Herramientas:

##### 1. Ajuste de CPU:

- **Identificar la Carga:** Usa `top/htop` (carga total, por núcleo), `vmstat` (`us`, `sy`, `id`, `wa`), `mpstat` (por núcleo). Alto `wa` (wait I/O) indica que la CPU espera por E/S de disco, apuntando a un cuello de botella de disco.
- **Prioridad de Procesos:** Ajusta la prioridad de los procesos para que las tareas más importantes reciban más tiempo de CPU.
  - `nice <valor> <comando>`: Inicia un comando con una prioridad ajustada (valor "nice" de -20 a 19; más bajo es mayor prioridad).
  - `renice <valor> -p <PID>`: Cambia la prioridad de un proceso en ejecución.
- **Planificador de Procesos:** El planificador del kernel decide qué proceso se ejecuta cuándo. `chrt` permite manipular la política y prioridad del planificador en tiempo real (para procesos de "tiempo real", avanzado).

- **Afinidad de CPU:** `taskset -c <número_cpu> <PID>`: Asigna un proceso a núcleos de CPU específicos. Puede ser útil para cargas de trabajo muy específicas o para aislar procesos.
- **Parámetros del Kernel:** Algunos parámetros en `/proc/sys/kernel/` o `/proc/sys/sched/` (gestionados con `sysctl`) pueden afectar el comportamiento del planificador, aunque los ajustes por defecto suelen ser buenos.
- **Gestión de Energía de CPU:** Verifica que el escalado de frecuencia de la CPU no esté limitando el rendimiento (ver archivos en `/sys/devices/system/cpu/cpu*/cpufreq/`).

## 2. Ajuste de Memoria:

- **Identificar el Uso:** Usa `free -h`, `vmstat`, `top`/`htop`. Un alto uso de swap indica que la RAM es insuficiente o que hay aplicaciones con fugas de memoria.
- **Swappiness:** Un parámetro del kernel (`/proc/sys/vm/swappiness`) que controla la tendencia del kernel a usar swap. Un valor alto (por defecto 60 en muchas distros) significa que el kernel intentará liberar RAM pasando páginas a swap más activamente. Un valor bajo (ej: 10) hace que el kernel prefiera mantener las páginas en RAM y usar swap solo como último recurso.
  - Ver valor actual: `cat /proc/sys/vm/swappiness`.
  - Cambiar temporalmente: `sudo echo <valor> > /proc/sys/vm/swappiness`.
  - Cambiar persistentemente: Añadir `vm.swappiness = <valor>` en un archivo `.conf` en `/etc/sysctl.d/`.
- **Caché del Kernel:** El kernel usa RAM para caché de disco. `vmstat` o `free` muestran la caché. Generalmente, es bueno que la caché sea grande; el kernel la liberará si las aplicaciones necesitan RAM. Ajustar parámetros de caché es avanzado.
- **OOM Killer:** El Out-Of-Memory killer del kernel termina procesos cuando el sistema se queda sin memoria. Puedes ajustar su comportamiento o la "puntuación" OOM de los procesos (en `/proc/<PID>/oom_score_adj`).

## 3. Ajuste de E/S de Disco:

- **Identificar el Uso:** Usa `iostat -x (%util, await, r/wMB/s)`, `vmstat (bi, bo, wa)`, `iotop` (uso de disco por proceso).
- **Planificador de E/S:** Elige el planificador adecuado para el tipo de dispositivo (Ej. 204.2). `noop` o `deadline/MQ-schedulers` para SSDs/NVMe/RAID hardware; `bfq` o `cfq` para HDDs. Configura persistentemente vía `udev` rules.
- **Opciones de Montaje:** Usa `noatime` o `relatime` (Ej. 203.1/204.2). Considera las opciones `data=` para ext4 si el rendimiento de escritura es crítico (con cuidado). Considera deshabilitar `barrier` si tienes una controladora de disco con batería de respaldo (riesgoso, solo si sabes lo que haces).
- **Sistema de Archivos:** Elige el sistema de archivos apropiado para la carga de trabajo (ej: XFS para E/S concurrente de archivos grandes).

- **Hardware:** Asegúrate de que el hardware de almacenamiento no sea el cuello de botella (discos más rápidos, interfaz más rápida, controladora RAID con caché de escritura).

#### 4. Ajuste de E/S de Red:

- **Identificar el Uso:** Usa `ss`, `netstat`, `iftop`, `nload`, `sar -n DEV, tcpdump`.
- **Interfaces:** Asegúrate de que la interfaz está configurada correctamente (velocidad, duplex) con `ethtool <interfaz>`. Deshabilita características de descarga si causan problemas (checksum offload, large send offload), aunque suelen mejorar el rendimiento.
- **Bonding/Bridging/VLANs:** Configura adecuadamente para ancho de banda o segmentación (Ej. 205.2).
- **Parámetros del Kernel:** Ajusta parámetros en `/proc/sys/net/` (gestionados con `sysctl`) relacionados con buffers TCP/UDP, colas de red, etc. (avanzado).
- **Firewalls:** Asegúrate de que las reglas de firewall no causen latencia inesperada (ej: logging excesivo).

#### Gestión de Parámetros del Kernel (`sysctl`):

Muchos aspectos del rendimiento se ajustan a través de parámetros en el sistema de archivos virtual `/proc/sys/`.

- **Ver todos los parámetros:** `sysctl -a`.
- **Ver un parámetro específico:** `sysctl <nombre_parametro>` (ej: `sysctl vm.swappiness`).
- **Cambiar un parámetro temporalmente:** `sudo sysctl <nombre_parametro>=<valor>` (ej: `sudo sysctl vm.swappiness=10`). Esto cambia el valor en `/proc/sys/` inmediatamente, pero se pierde al reiniciar.
- **Cambiar parámetros persistentemente:** Coloca los ajustes en archivos `.conf` en el directorio `/etc/sysctl.d/`. Los archivos en este directorio son leídos durante el arranque por el servicio `systemd-sysctl.service` o scripts tradicionales. El formato es `nombre.parametro = valor`.
  - Ejemplo: Crear `/etc/sysctl.d/99-my-perf.conf` con contenido `vm.swappiness = 10`.