¡Excelente! Aunque el objetivo 206.3 no estaba en la lista inicial para el Examen 206, "Mejorar el rendimiento del sistema" es un tema perfectamente alineado con el mantenimiento del sistema y un objetivo clave en LPIC-2. Lo añadiremos a la sección que acabamos de explorar.

La mejora del rendimiento es un proceso continuo que requiere entender cómo interactúan los recursos del sistema y cómo identificar y eliminar cuellos de botella. Analizaremos cómo optimizar la CPU, la memoria, la E/S de disco y la red, utilizando las herramientas de monitorización que ya conocemos.

Comencemos con el objetivo 206.3.

📁 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, `iotop` 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`.

📁 **LPIC-2 /** 🛡️ **Examen 206 - Mantenimiento del Sistema - Ejercicios**

Nota: Estos ejercicios implican modificar configuraciones del sistema y del kernel. ***Realiza cambios persistentes (en `/etc/sysctl.d/`) o cambios temporales en `/proc/sys/` o con `ethtool` SOLO en una VM de prueba.*** *Entiende el impacto de cada ajuste antes de aplicarlo.*

Ejercicio 6.3.1: Ajustando la Prioridad de Procesos con `nice` y `renice`

- **Objetivo:** Cambiar la prioridad de ejecución de un comando o proceso.
- **Requisitos:** Acceso a la línea de comandos. Privilegios de superusuario (`sudo`) para renice a procesos de otros usuarios o con valores negativos.
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Inicia un comando con prioridad baja:** Ejecuta `nice +10 dd if=/dev/zero of=/dev/null`. Esto inicia `dd` con un valor nice de 10 (menos prioridad que la normal). Déjalo corriendo.
    3. **En otra terminal, identifica el PID del proceso `dd`:** Ejecuta `ps aux | grep dd`. Anota el PID.
    4. **Verifica el valor NICE en `top`/`htop`:** Ejecuta `top` o `htop`. Busca el proceso `dd`. La columna `NI` muestra el valor nice. Debería ser 10.
    5. **Cambia la prioridad de `dd` a un valor más alto (menos nice, más prioridad):** Ejecuta `sudo renice -5 -p <PID_de_dd>`.
    6. **Verifica el cambio en `top`/`htop`:** El valor `NI` para `dd` debería ser ahora -5. El proceso recibirá más tiempo de CPU.
    7. **Detén el proceso `dd`:** Búscalo en `top`/`htop` o `ps` y mátalo con `kill <PID>`.

Ejercicio 6.3.2: Viendo y Cambiando Parámetros del Kernel con `sysctl` y `/proc/sys/`

- **Objetivo:** Interactuar con parámetros de ajuste del kernel.
- **Requisitos:** Acceso a la línea de comandos. Privilegios de superusuario (`sudo`) para escribir. **VM de prueba.**
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Ver el valor actual de swappiness (método `/proc/sys/`):** Ejecuta `cat /proc/sys/vm/swappiness`.
    3. **Ver el valor actual de swappiness (método `sysctl`):** Ejecuta `sysctl vm.swappiness`.
    4. **Ver todos los parámetros relacionados con vm (memoria virtual):** Ejecuta `sysctl vm`.
    5. **Ver todos los parámetros relacionados con red:** Ejecuta `sysctl net`.
    6. **Cambiar swappiness temporalmente (método `/proc/sys/`):** Ejecuta `sudo echo 10 > /proc/sys/vm/swappiness`.

7. **Verificar el cambio:** Ejecuta `cat /proc/sys/vm/swappiness` o `sysctl vm.swappiness`. Debería ser 10.
8. **Cambiar swappiness temporalmente (método `sysctl -w`):** Ejecuta `sudo sysctl -w vm.swappiness=20`.
9. **Verificar el cambio:** Ahora debería ser 20.
10. **Reinicia la VM para que los cambios temporales se pierdan.** Verifica que `swappiness` vuelve a su valor original después del arranque.

Ejercicio 6.3.3: Configurando Parámetros del Kernel Persistentemente con `/etc/sysctl.d/`

- **Objetivo:** Hacer que un ajuste de kernel sea permanente.
- **Requisitos:** Privilegios de superusuario (`sudo`). **VM de prueba.**
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Explora el directorio de configuración de sysctl:** Ejecuta `ls -l /etc/sysctl.d/`. Verás archivos `.conf` que contienen ajustes de parámetros.
    3. **Crea un nuevo archivo de configuración para tus ajustes (requiere sudo):** Dale un nombre que empiece con un número para controlar el orden, ej: `99-my-swappiness.conf`. Ejecuta `sudo vi /etc/sysctl.d/99-my-swappiness.conf`.
    4. **Añade la línea de configuración:** Escribe `vm.swappiness = 10`.
    5. **Guarda y sal.**
    6. **Aplica los cambios SIN REINICIAR:** Puedes forzar a `sysctl` a cargar los archivos de configuración: `sudo sysctl -p /etc/sysctl.d/99-my-swappiness.conf` o `sudo systemctl restart systemd-sysctl.service` (si usas systemd).
    7. **Verifica que el cambio fue aplicado:** Ejecuta `sysctl vm.swappiness`. Debería ser 10.
    8. **Reinicia la VM.**
    9. **Después de reiniciar, verifica que el valor es 10:** Ejecuta `sysctl vm.swappiness`. Esto confirma que el ajuste es persistente.
    10. **Limpia en VM:** Elimina el archivo de configuración si no quieres que el ajuste sea permanente: `sudo rm /etc/sysctl.d/99-my-swappiness.conf`.

Ejercicio 6.3.4: Verificando Configuración de Interfaz de Red con `ethtool`

- **Objetivo:** Ver parámetros de rendimiento de la interfaz de red.
- **Requisitos:** Acceso a la línea de comandos. El paquete `ethtool` puede necesitar instalación (`sudo apt install ethtool` o `sudo dnf install ethtool`). Una interfaz de red física o virtual real.
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.

2. **Identifica tu interfaz principal:** `ip addr show`. Anota el nombre (ej: `enp3s0`).
3. **Verifica la velocidad, duplex y otras configuraciones (con sudo si no eres root):** Ejecuta `sudo ethtool <tu_interfaz>`. Observa las líneas `Speed`, `Duplex`, `Auto-negotiation`, `Link detected`.
4. **Verifica las características de descarga (offload) de la interfaz:** Ejecuta `sudo ethtool -k <tu_interfaz>`. Busca características como `tx-checksum-offload`, `rx-checksum-offload`, `tcp-segmentation-offload` (TSO), `large-receive-offload` (LRO). Suelen estar habilitadas (`on`) por defecto para mejorar el rendimiento. Deshabilitarlas (`off`) puede ser necesario para solucionar problemas de compatibilidad o rendimiento extraño en algunos casos.
5. **(Concepto):** Ajustar la velocidad y duplex manualmente (si la auto-negociación falla) o deshabilitar características de descarga puede ser necesario en troubleshooting de rendimiento de red.

Ejercicio 6.3.5: (Conceptual) Aplicando Ajustes Basado en Cuellos de Botella

- **Objetivo:** Entender cómo los resultados de monitorización (`top`, `iostat`, etc.) guían las acciones de ajuste.
- **Desarrollo Paso a Paso:**
    1. Abre una terminal.
    2. **Escenario 1: CPU al 100%, alto `%us` (user CPU):**
        - Monitorización: `top`/`htop` muestra un proceso de usuario consumiendo mucha CPU.
        - Acción: Identificar el proceso, optimizar la aplicación, o escalar (más CPU). Considerar ajustar prioridades si hay procesos críticos.
    3. **Escenario 2: CPU al 100%, alto `%sy` (system CPU):**
        - Monitorización: `top`/`htop` muestra alto uso de CPU del kernel. Puede indicar llamadas al sistema excesivas, problemas de driver, o E/S intensiva que no se refleja totalmente como `wa`.
        - Acción: Investigar qué operaciones del kernel son intensivas. Monitorizar E/S de disco/red. Puede requerir optimización de la aplicación o actualización/ajuste de drivers.
    4. **Escenario 3: CPU con `%wa` alto (wait I/O):**
        - Monitorización: `top`/`htop` muestra alto `wa`. `iostat -x` muestra alto `%util` y `await` en un disco.
        - Acción: El cuello de botella es el disco. Considerar cambiar el planificador de E/S, usar opciones de montaje como `noatime`, optimizar el acceso a disco de la aplicación, o mejorar el hardware de almacenamiento (SSD, RAID más rápido).
    5. **Escenario 4: Uso de Memoria alto, swap activo:**
        - Monitorización: `free` muestra poca memoria libre, `vmstat` muestra actividad en `si`/`so` (swap in/out).

- Acción: El cuello de botella es la memoria. Identificar procesos con alto `%MEM`. Considerar añadir más RAM. Ajustar `vm.swappiness` (más bajo si quieres evitar swap, más alto si tienes RAM de sobra y prefieres usar swap activamente).

6. **Escenario 5: Red saturada o con errores:**
   - Monitorización: `sar -n DEV` muestra alto tráfico. `ethtool` muestra errores/colisiones. `tcpdump` muestra retransmisiones TCP o paquetes perdidos.
   - Acción: El cuello de botella es la red. Verificar velocidad/duplex (`ethtool`), cableado, configuración del switch, congestión en la red. Considerar bonding para más ancho de banda. Ajustar parámetros de red del kernel si es un cuello de botella en el host mismo.