📁 LPIC-2 / 💾 Examen 204 - Administración Avanzada de Dispositivos de Almacenamiento

204.2 Ajustar el acceso a dispositivos de almacenamiento

Teoría

El rendimiento de la E/S de disco (Input/Output) es a menudo uno de los principales cuellos de botella en un sistema. Ajustar cómo el kernel maneja las solicitudes de E/S de los procesos puede mejorar significativamente la velocidad y la capacidad de respuesta del sistema de almacenamiento.

Factores que Afectan el Rendimiento de Disco:

- **Tecnología del Disco:** HDDs (discos duros tradicionales con platos giratorios - lentos en acceso aleatorio, buscan datos) vs. SSDs (unidades de estado sólido basadas en memoria flash - muy rápidos en acceso aleatorio, sin búsqueda).
- **Interfaz:** SATA, SAS, NVMe (la interfaz determina la velocidad máxima de transferencia).
- **Tipo de Sistema de Archivos:** El diseño del sistema de archivos impacta en cómo se organizan los datos y metadatos, afectando el rendimiento (ej: XFS para cargas grandes, ext4 para uso general).
- **Opciones de Montaje:** Configuran el comportamiento del sistema de archivos (ej: cómo se maneja el journal, cuándo se actualizan los tiempos de acceso).
- **Planificador de E/S (I/O Scheduler):** Componente del kernel que ordena las solicitudes de E/S.

Planificadores de E/S (I/O Schedulers):

Son algoritmos dentro del kernel que deciden en qué orden se envían las solicitudes de lectura/escritura de múltiples procesos a los dispositivos de bloque. Su objetivo es minimizar el tiempo de búsqueda (seek time) en HDDs y optimizar el rendimiento de colas en SSDs y RAID.

- **Funcionamiento:** Las solicitudes de E/S de diferentes procesos se acumulan en una cola. El planificador reorganiza estas solicitudes antes de pasarlas al controlador del dispositivo.
- **Planificadores Comunes (Históricos y Modernos):** La disponibilidad y el predeterminado dependen de la versión del kernel y la distribución.
    - **CFQ (Completely Fair Queuing):** (Común en HDDs antiguos) Intenta ser justo con todos los procesos, dando a cada uno un "presupuesto de tiempo" para la E/S. Bueno para cargas de trabajo mixtas.
    - **Deadline:** (Común en SSDs o bases de datos antiguas) Prioriza las solicitudes de lectura para garantizar que se completen dentro de un plazo (`deadline`) corto, previniendo la inanición de las lecturas en favor de las escrituras. Bueno para aplicaciones sensibles a la latencia.
    - **NOOP:** (Simple First-In, First-Out - FIFO) No hace reordenamiento inteligente. Deja que el hardware (controladora RAID, SSDs modernos) gestione la cola. Adecuado cuando el dispositivo ya optimiza las solicitudes internamente.

- **BFQ (Budget Fair Queueing):** (Nacidos con los sistemas de múltiples colas) Intenta garantizar un tiempo de disco predecible para cada proceso. Bueno para interactividad y cargas de trabajo mixtas.
- **MQ-deadline, KYBER, BFQ:** Planificadores más nuevos diseñados para dispositivos con múltiples colas (NVMe, SSDs de alto rendimiento). Son el predeterminado en kernels recientes.

Verificar y Cambiar el Planificador de E/S:

- **Ver planificador actual y disponibles:** Puedes ver el planificador activo para un dispositivo de bloque y la lista de planificadores disponibles inspeccionando un archivo en `sysfs`.
  - `cat /sys/block/<nombre_dispositivo>/queue/scheduler` (ej: `cat /sys/block/sda/queue/scheduler`). La salida mostrará el planificador activo entre corchetes `[ ]`. Los otros son los disponibles.
- **Cambiar planificador temporalmente (hasta reiniciar):** Escribe el nombre del planificador deseado en el mismo archivo de sysfs (requiere root).
  - `sudo echo <nombre_planificador> > /sys/block/<nombre_dispositivo>/queue/scheduler` (ej: `sudo echo deadline > /sys/block/sda/queue/scheduler`).
- **Cambiar planificador persistentemente:** Esto se configura al arrancar el sistema:
  - **Parámetro del Kernel:** Añadir el parámetro `elevator=<nombre_planificador>` a la línea de comandos del kernel en la configuración de GRUB (en `/etc/default/grub`, luego regenerar `grub.cfg`). Esto establece el planificador por defecto para *todos* los dispositivos, a menos que se anule. (Ver 2.2.2 y 2.3.4).
  - **Reglas de Udev:** Crear o modificar una regla de udev en `/etc/udev/rules.d/` que coincida con el dispositivo específico y use la clave `ATTR{queue/scheduler}=` para establecer el planificador. (Ver 2.2.4). Este es el método preferido para configurar planificadores por dispositivo.

Caché de Disco (Disk Caching):

- El kernel utiliza la RAM libre como caché de disco (Page Cache y Buffer Cache) para almacenar bloques de datos leídos o a la espera de ser escritos. Esto acelera el acceso a datos a los que se accede con frecuencia.
- Cuando escribes en un archivo, los datos van primero a la caché y el kernel los escribe al disco más tarde (escritura asíncrona).
- La opción de montaje `sync` (ver 203.1) fuerza las escrituras a ser síncronas, pasando por alto la caché de escritura del kernel. Esto es más seguro pero más lento.

Opciones de Montaje Relacionadas con el Rendimiento (Revisado de 203.1):

Algunas opciones de montaje afectan el rendimiento:

- `atime`, `noatime`, `relatime`: Controlan cuándo se actualiza el tiempo de último acceso a un archivo (`atime`). Actualizar `atime` para cada lectura genera escrituras adicionales y penaliza el rendimiento.
    - `atime`: Actualiza `atime` cada vez que se accede al archivo (por defecto en sistemas de archivos antiguos).
    - `noatime`: Deshabilita por completo las actualizaciones de `atime`. Mejora el rendimiento pero puede romper aplicaciones que dependen de `atime`.
    - `relatime`: Actualiza `atime` solo si el `mtime` (tiempo de modificación) ha cambiado o si el `atime` anterior es anterior al `mtime` o `ctime` (tiempo de cambio de metadatos). Es un buen compromiso y suele ser el predeterminado moderno.
- `data=ordered`, `data=journal`, `data=writeback` (para ext4): Controlan cómo se escriben los datos en relación con las actualizaciones del journal. `ordered` (por defecto) es un buen equilibrio entre rendimiento y seguridad de datos. `journal` es más lento pero más seguro. `writeback` es más rápido pero menos seguro.
- `barrier=1` / `barrier=0` (para ext4/XFS): Controla el uso de barreras para garantizar el orden de escritura en el journal. Habilitado por defecto (`1`) para seguridad, deshabilitarlo (`0`) puede mejorar el rendimiento pero es riesgoso con cachés de escritura del disco/controladora.

Monitorización de E/S de Disco para Ajuste:

- **`iostat -x`:** Muestra el `%util` del dispositivo (porcentaje de tiempo que el dispositivo está ocupado) y `await` (tiempo promedio de espera). Altos valores indican cuello de botella.
- **`vmstat`:** Muestra `wa` (tiempo de espera E/S de CPU). Alto `wa` indica que la CPU está esperando por E/S de disco.
- **`iotop`:** Muestra el uso de E/S por proceso.

Diferencias Debian vs. Red Hat (Ajuste de Almacenamiento):

- **Planificador de E/S por Defecto:** Puede variar entre distribuciones y versiones del kernel. Las versiones recientes de ambos probablemente usarán planificadores multi-cola (MQ) como `mq-deadline` o `bfq` para SSDs y NVMe, y tal vez `bfq` o `cfq` para HDDs si se usa una sola cola.
- **Método de Configuración Persistente:** Ambas usan parámetros del kernel de GRUB y reglas de udev, pero la convención específica en los archivos de reglas por defecto puede variar.
- **Opciones de Montaje por Defecto:** Las opciones por defecto aplicadas por `mount -o defaults` o en `/etc/fstab` pueden diferir ligeramente.