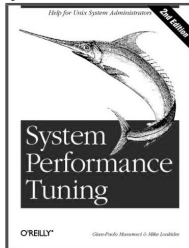


3. Solución de problemas en un sistema informático

El libro de referencia para este tema es *System*



Performance Tuning, 2nd Edition, de Gian Paolo Musumeci y Mike Loukides, de O'Reilly. Dedicar un capítulo a cada uno de los subsistemas del ordenador, indicando cómo detectar problemas usando monitores, y cómo solucionarlos usando las diferentes técnicas al alcance del administrador del sistema. La edición es del año 2002, y está, por tanto, bastante actualizado. Sus sistemas operativos de referencia son Linux y Solaris. Microsoft y Windows ni siquiera aparecen en el índice. También puedes comprar el libro en [Amazon UK](#)

3.1 Introducción

Una vez evaluado el rendimiento de un sistema informático, hay una serie de medidas que se pueden tomar para sintonizarlo. En concreto, puede hacerse algo de lo siguiente:

- Ajuste de parámetros del sistema operativo: hay algunos parámetros que el superusuario, o administrador del sistema, puede modificar, usando programas suministrados con el sistema operativo o recompilando el sistema operativo (o el kernel, como sucede en algunas versiones de UNIX). Estos parámetros son, por ejemplo, el tamaño del quantum, o rodaja temporal asignada a cada uno de los programas, prioridad interna asignada a un programa de usuario, tamaño de la partición de memoria, frecuencia de fallo de página e índice de supervivencia de las páginas, y todo lo demás relacionado con el usuario y los procesos.
- Ajuste de parámetros del hardware, es decir, examinar la configuración hardware del sistema y ver qué parámetros se pueden alterar, tales como por ejemplo la activación de cachés hardware, el reloj del sistema, frecuencia del bus. Algunos de estos cambios pueden ser peligrosos. En todo caso, en ordenadores con placas madres antiguas se tendrá que hacer mediante cambios de jumpers (es

decir, pequeños puentes entre dos conectores), y en ordenadores modernos accediendo al setup al arrancar el ordenador (habitualmente pulsando CTRL-ALT-ESC, o una combinación de teclas similar).

- Equilibrado de cargas: repartir las cargas a las que son sometidos los diversos dispositivos, como red, discos duros, entre las diferentes máquinas que las gestionen y personas que lo usan, o repartir los ficheros entre los diferentes filesystems del sistema.
- Ampliación: si hay dinero, tiempo y ganas, se pueden comprar dispositivos nuevos, o cambiar los dispositivos por otro más rápido. Previamente, claro, habrá que realizar un análisis de cuáles son los dispositivos que están limitando las prestaciones del sistema.
- Cambio del software: puede ser un upgrade de una versión del sistema o cambiar a una versión superior, o cambiar el software que se está usando por otra versión u otra marca.

Conviene también tener en cuenta una serie de principios a la hora de mejorar las prestaciones de un sistema:

1. Conoce y comprende tu entorno: es conveniente saber todo lo que hace el sistema: qué subsistemas tiene activados, cuáles son los servicios que se arrancan y qué es lo que hacen, qué procesos tiene activos en cada momento. Conviene también tener ciertas nociones de cómo funciona el sistema operativo con el que se trabaja, la red, el subsistema de E/S. No es necesario bajar hasta el nivel de saber qué registros de la CPU se usan en cada momento, pero sí tener cierta idea de todo. Así, en el caso de que algo vaya mal, se podrá identificar rápidamente la causa, y evitarla.
2. Hay que buscar el equilibrio: en inglés se suele decir TANSTAAFL; there are not such a thing as a free lunch, es decir, no existen los almuerzos gratuitos: lo que te dan gratis por un lado, te lo cobran por otro. Por ejemplo, no se puede mejorar la velocidad percibida por un usuario sin empeorar la de todos los demás usuarios; no se puede optimizar al máximo un programa sin aumentar, como mínimo, el tiempo que se le dedica al mismo. Y si quiere gastarse uno el dinero en aumentar la memoria, dinero sin el que se queda para añadir un disco duro más al sistema.
3. Throughput contra latencia: el throughput se refiere a la cantidad de cosas que se pueden hacer, o transmitir, en la unidad de tiempo; por ejemplo, a la cantidad de MB/s que se pueden escribir en disco, o a la cantidad de procesos que un procesador es capaz de procesar (valga la redundancia) en la unidad de tiempo. La latencia, sin embargo, es el tiempo total que tarda algo en hacerse. Por ejemplo, en una red, la latencia sería el tiempo que tarda un paquete para transmitirse de un punto a otro de la red, mientras que el throughput sería la cantidad de bytes que la red es capaz de soportar. Para ver cómo un concepto se opone al otro, mirar este artículo sobre el tamaño de los paquetes en las redes: cuanto más pequeño es un paquete, menor es la latencia (tarda menos en llegar de un punto de la red a otra), pero disminuye el throughput, porque el tamaño relativo del overhead, es decir, de la información adicional que se le añade al paquete, toma mayor importancia.
4. No sobreutilices un recurso: ningún recurso usado al 100% estará al óptimo de su capacidad, sino más bien por encima. Por encima del 50% ya empieza a acusar el uso; por encima del 70% ya empieza a estar sobrecargado. Todo esto, claro está, depende del sistema operativo: en Windows, un uso del 95% cuando hay un par de programas ejecutándose es de lo más normal; sin embargo, en cualquier sistema Unix, sería preocupante.
5. Diseña las pruebas con cuidado: aplica el primer principio, y mira bien lo que vas a medir antes de medirlo. Por ejemplo, si para medir la velocidad de lectura/escritura de un CD copias parte de su contenido al disco duro, estarás midiendo tanto la velocidad del disco duro, como del interfaz IDE, como de la memoria, como de la CPU; algunas de esas cosas no se podrán evitar, pero sí, por ejemplo, el uso del disco duro; de la misma forma, si mides la velocidad de la red bajándote algo de una página web, tendrás que descontar el efecto del servidor web, del disco del servidor, la carga del servidor... trata de diseñar las pruebas de forma que eliminen todos los factores del sistema que no interesen, o que puedan afectarlo. Y esto se puede aplicar a cualquier medición o benchmark.

1 Ejercicios de autoevaluación

1. Busca en Internet o en el manual de tu sistema operativo cómo puedes alterar los parámetros siguientes de tu sistema operativo: tamaño de la

rodaja temporal, índice de supervivencia de página, número máximo de usuarios, número máximo de procesos.

2. Mirando las pantallas de configuración de tu ordenador, dí qué parámetros del hardware se pueden cambiar: reloj del sistema, por ejemplo, o frecuencia del bus del sistema. 3. Buscar en Internet algún contrato de prestación de servicios que incluya condiciones sobre las prestaciones de un sistema que va a contratar; presentar el enlace.

3.2 Gestión de carga y prestaciones en el sistema operativo

En general, un administrador o administradora de un sistema tiene que mantener a todos sus usuarios felices (aunque esto sea matemática y filosóficamente imposible), y para ello tiene que cuidar el sistema como si de un bebé se tratara; en general, también, como tal bebé, al principio da mucha guerra, pero luego es capaz de ir solo al cuarto de baño. Por tanto, tiene que plantear la gestión de un sistema de la siguiente forma:

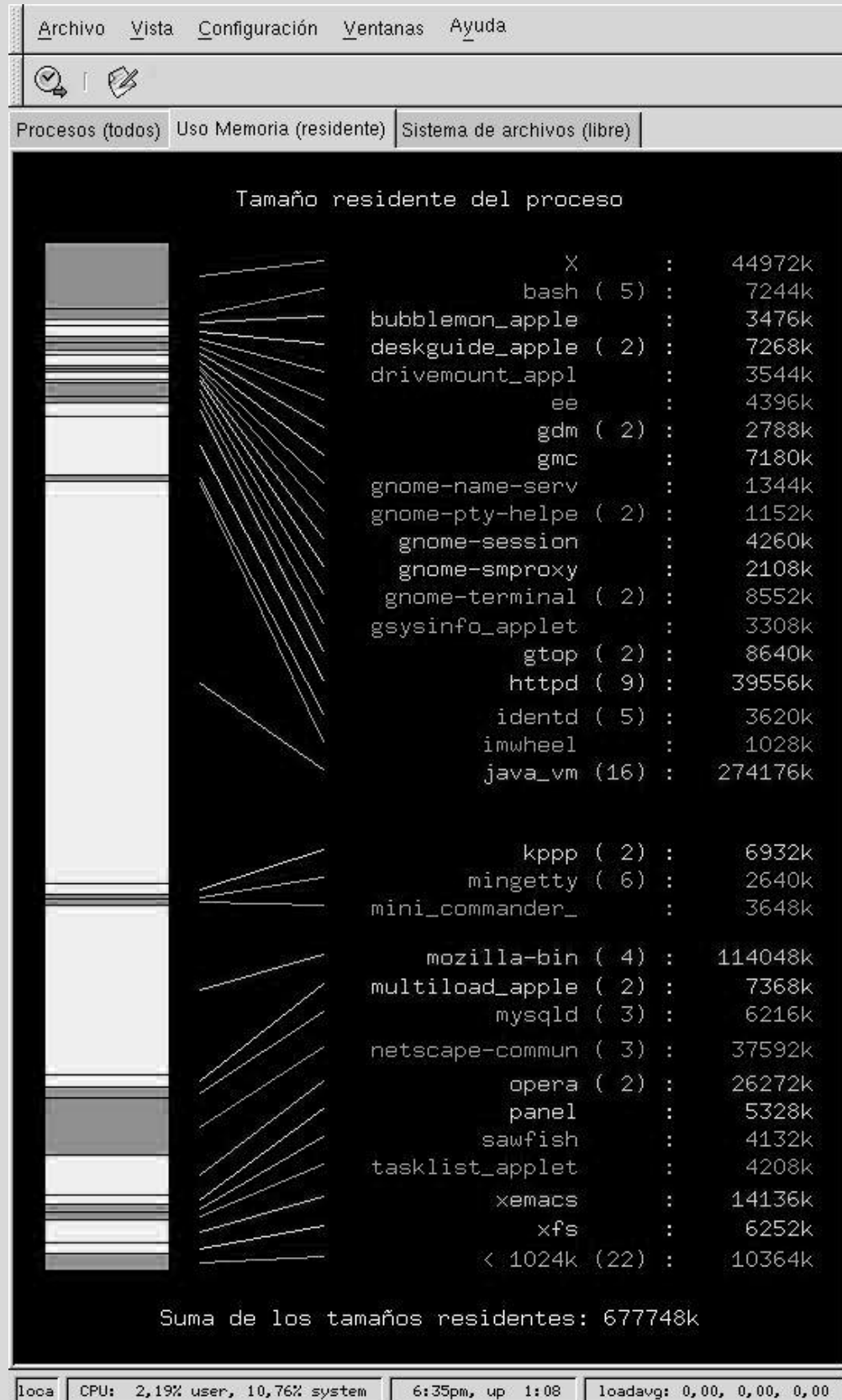
- Planificación de la carga y definición de la carga del sistema: es conveniente acordar de antemano (con la autoridad competente) qué se considera unas prestaciones aceptables. Por ejemplo, ¿que la compilación de un programa de los habituales dure 2 minutos se considera aceptable? ¿Que una petición SQL dure uno lo es? ¿Cuántas paradas por mantenimiento lo son? ¿Qué velocidad de la red? Una vez llevada a cabo esta planificación, hay que comprobar si, con el sistema tal como está, se puede llevar a cabo ese acuerdo; y si en el futuro previsible, con los usuarios y la carga pico prevista, se va a poder producir.
- Por ejemplo, a la hora de adquirir un servidor Web para un proveedor de servicios, y basado en el número de peticiones por minuto y hora que se van a recibir, habrá que estimar el tamaño de la memoria necesaria, el ancho de banda necesario en función de las peticiones que se van a servir, el tamaño de disco duro, por ejemplo; además, el administrador deberá comprobar la evolución del número de peticiones y prever los picos de peticiones para, si es necesario, solicitar más ancho de banda (bandwidth-on-demand; algunos servidores permiten solicitar en un momento determinado más ancho de banda que el que se tiene asignado) o una ampliación del sistema; o bien cambiar la configuración para que soporte el pico de demanda.
- Configurar las herramientas de monitorización del sistema: se deberán poner en funcionamiento las herramientas que monitorizan en cada momento los subsistemas principales: CPU, E/S, red, y memoria; estos monitores nos indicarán como se usa el sistema en cada momento y a lo largo del tiempo, y nos permitirán prever los fallos y arreglarlos fácilmente (o no) cuando se produzcan; también habrá que escribir una serie de scripts (en PERL, shell, AWK o en lo que a uno le apetezca) que avisen de que alguna cosa vaya mal. Por ejemplo, en algunos sistemas hay que cambiar algún programa de inicio, como perf, para que arranque utilidades como sadc, una utilidad que crea ficheros que son luego leídos por sar.
- Tratar de resolver problemas mediante políticas de gestión del sistema, tales como limitación de uso interactivo, limitación de uso de disco mediante cuotas,
- Ampliar el sistema, si todo falla (y hay pasta para ello, claro).

3.3 Políticas de gestión del sistema

Tanto los usuarios como el administrador pueden mejorar el funcionamiento del sistema. Por ejemplo, algunas medidas que pueden tomar los usuarios si buenamente les da la gana y les apetece (o si el BOFH no les deja otro remedio) es:

- Usar comandos internos del shell en vez de los comandos externos de UNIX; y esto por una razón muy simple: el shell ya está en memoria, ejecutándose, mientras que los comandos externos se tienen que buscar en el path, cargarse y demás. Por ejemplo, dirs en vez de pwd, echo * en vez de ls, y cosas así. Este tipo de medidas será útil siempre que todos los usuarios estén trabajando sobre un solo sistema.
- Evitar los caminos largos, que hacen que el ordenador tenga que leer muchos

directorios cada vez que se ejecuta un comando. Evitar los directorios con muchos ficheros; el tamaño del fichero de directorio crece con el número de ficheros, y ¡no disminuye nunca! Cuando un fichero de directorio es demasiado grande, no queda otro remedio que crear un nuevo directorio y mover todos los ficheros a él.



- Usar las versiones más eficientes de cada tipo de programa: por ejemplo, por razones obvias, `vi` es más eficiente que `emacs`; `xjed` y programas así también. Si es que uno puede vivir sin `emacs`, claro; también `egrep` es más rápido que `grep`; algunas versiones son incluso mejores. Algunos shells son más eficientes que

otros: por ejemplo, tcshell es menos eficiente que bash; algunos shells, como rc son incluso más pequeños. Lo mismo ocurre con los gestores de ventanas: fvwm es más eficiente que fvwm95, que lo es más que Open View, que lo es más que KDE, que es más que Enlightenment. En cuanto a los procesadores de textos, LyX y kLyX consume muchos menos recursos que OpenOffice (que ocupa a ná que te descuides 150 Megas). En cuanto a los navegadores, cada vez ocupan más memoria, sobre todo si tienen la máquina virtual Java activada. El Mozilla (versión beta) ocupa mucha más memoria (alrededor de 100 megas) que el Netscape (versión 4.75, unos 40 megas), y ese a su vez más que Opera (unos 26 megas), y eso si no incluimos las máquinas virtuales Java que lanzan cada uno. A pesar de compartirse código entre las diferentes copias de las aplicaciones, en un ordenador con 512 megas de memoria no podrá haber más de 3 o 4 Netscapes abiertos.

En cuanto al superusuario, hay muchas cosas que puede llevar a cabo para aligerar la carga del sistema:

- Eliminar daemons inútiles y malos para el alma de la máquina. Por ejemplo, ¿realmente necesitas un servidor web en el linux de tu casa? ¿Y 4 daemons de nfs? En general, es conveniente echar un vistazo a los daemons que se activan por defecto (que uno previamente tiene que haber autorizado cuando ha instalado el sistema operativo) cuando se enciende el ordenador y eliminar los más innecesarios. También se puede eliminar el número de daemons que usa cada servicio: normalmente se activan varios biod (cliente NFS), httpd, nfsd (servidor NFS); no siempre son necesarios más de 4. Todo esto se puede hacer con el programa `linuxconf`, solapa "control".
- Limitar tiempos de ejecución interactivos, y renicear procesos a discreción; hacer que los usuarios hagan uso de sistemas de encolado como NQS.`nice` y `renice` son utilidades de Unix que permiten cambiar la prioridad de un proceso en función de los otros procesos que estén ejecutándose. Para limitar tiempos de ejecución, se puede usar en algunos Unix el comando `limit` de esta forma:

```
bash$ limit cputime 200m
```

, que limita a 200 minutos el uso de CPU. Usando `ulimit`, que es un comando interno del `bash`, se puede hacer de la forma siguiente:

```
bash$ ulimit -t 200m
```

Hay una [guía para recompilar el kernel en frikis.org](#), en castellano. También una [Howto](#) de cómo compilar el kernel, en inglés, que debería venir con la distro de Linux que uses. La [versión en castellano, de JJ Amor](#), está un tanto atrasada.

- Modificar (uh!) los parámetros del sistema operativo. Esto no se debería hacer si no se tiene bastante experiencia y poco miedo de las posibles reacciones de los usuarios. Una serie de parámetros controlan, por ejemplo, el número de procesos máximos que puede abrir el sistema, el número de inodos activos, el número de buffers, el número de regiones o partes diferentes de un proceso (habitualmente son 3: pila, texto y código), procesos por usuario, ficheros por proceso, etcétera. La mayoría de las veces, modificar estos parámetros consiste en recompilar el kernel, y Dios guarde a quien tenga que hacer este tipo de cosas. Por ejemplo, para modificar el número máximo de tareas, hay que editar (en Linux) `/usr/include/linux-2.x.x/tasks.h`, y cambiar alguna de las dos variables siguientes: `NR_TASKS` o `MAX_TASKS_PER_USER`; luego, como es natural, hay que recompilar el kernel. Estos parámetros se pueden ver con `pstat` (pero no en Linux) o con `sar -v`; en Linux hay que mirar en diferentes ficheros del subdirectorio `/proc/sys`, por ejemplo los que hay en el directorio `/proc/sys/kernel`. . Otro parámetro modificable es el quantum temporal, o rodaja temporal; incrementarla significa disminuir el overhead del sistema, puesto que se tienen que hacer menos cambios de contexto, pero sufre las prestaciones de los

procesos interactivos, que son casi todos; disminuirla beneficia más a los procesos interactivos, a costa de más cambios de contexto. En Linux, la rodaja temporal básica está definida (en `include/linux/sched.h`, de la forma siguiente:

```
#define DEF_PRIORITY (20*HZ/100) /* 210 ms time slices
```

; la cantidad anterior equivale a 20 ticks (`HZ` vale 100).

Este tipo de políticas son utilizables en el caso genérico, es decir, nunca están de más. En particular, para cada uno de los subsistemas, hay una serie de parámetros a los que habrá que controlar. Los veremos a continuación.

2 Ejercicios de autoevaluación

1. Comparar dos programas que hagan la misma labor (por ejemplo, dos procesadores de textos), ejecutando simultáneamente un monitor, y calcular a ojo de buen cubero los recursos de CPU y memoria que consumen. ¿Si hay varias copias del programa, cómo evoluciona el consumo de recursos?
2. Monitorizar la CPU y memoria de un ordenador recién encendido; eliminar servicios innecesarios del arranque, y volver a hacer la monitorización. ¿Ha mejorado algo?
3. Modificar algún parámetro del sistema operativo, tal como la rodaja temporal; monitorizar el sistema antes y después de la modificación. ¿Se nota alguna diferencia?

3.4 Mejora de prestaciones de la CPU

Monitorización de CPU en Windows NT/2000/XP

En NT (y supongo que en sus sucesores Windows 2000 y XP, sucederá más o menos igual), aunque no tiene mucho remedio, se puede tratar de usar el monitor de prestaciones del sistema, PERFMON, para mirar el estado del sistema. Los parámetros que más afectan a la CPU son:

- Procesador: %tiempo del procesador. Si es superior al 70%, lo cual sucede si estás ejecutando aunque sea solo un programa, la cosa está chunga. Como en mi (antiguo) Pentium 166 con 32 Megas de memoria pasa, pues no sé muy bien como se va a solucionar el tema. Pero para saber exactamente de donde viene el problema, hay que mirar los siguientes parámetros.
- Procesador: interrupciones por segundo. Si es excesivamente alto, el tiempo de procesador puede estar causado por algún dispositivo o interfaz, como IDE, o dispositivos antiguos de red.
- Sistema: Llamadas al sistema por segundo; cuando hay más interrupciones que llamadas al sistema, se confirma que los dispositivos están generando más demanda que las llamadas software a los servicios NT.
- Cambios de contexto por segundo. Si es excesivamente alta, un tipo de procesador en el cual el cambio de contexto esté optimizado, como los RISC, puede mejorar prestaciones. Recordar que WinNT funciona en procesadores Alpha y MIPS, además de los Intel (aunque no estoy muy seguro de que el XP lo siga soportando); también se podría cambiar, simplemente, a un procesador de una generación posterior. .

La CPU no es el tipo de cosas con las que uno deba jugar, pero en todo caso algo se puede hacer para que vaya más rápido. Estas mejoras no llegarán por el lado del software (aunque hay utilidades que supuestamente mejoran las prestaciones de la

CPU), sino por el lado del hardware: cambiar la velocidad en megahercios a la que funciona la CPU. En placas actuales, esto se puede hacer directamente desde el setup de la BIOS (Basic Input/Output System). La estrategia que hay que seguir es la siguiente (tal como se puede encontrar, por ejemplo, en <http://www.firingsquad.com> o en <http://www.sharkyextreme.com>).

- Cambiar el setup de la BIOS de forma que la CPU vaya más rápida, en un pequeño incremento.
- Cambiar en el setup también la velocidad del bus del sistema, para que se empareje con la velocidad de la CPU. Ambas velocidades son submúltiplos de la velocidad del reloj del sistema, que, en general, es el doble de la velocidad del procesador.
- Si es posible, cambiar el voltaje al que funciona la CPU, en alguna pequeña cantidad, tratando de disminuirlo, siempre dentro de los límites de tolerancia de los componentes. Algunas placas madre, tales como las Abit (<http://www.abit.com.tw>) lo permiten.
- También es conveniente añadir un ventilador para el procesador solo, para evitar que se caliente demasiado.
- Si no se queda colgado, probar algún benchmark sobre el nuevo sistema, a ver qué incremento de velocidad se ha conseguido.
- Volver al principio, a intentar conseguir velocidades superiores.

Una estupenda guía para overclockers está en [la guía básica de overclocking de CPUs de Sharky Extreme](#). Hay también bastantes ejemplos de overclocking en [Overclockers UK](#), inclusive accesorios tales como ventiladores y carcasas. Hay varios artículos en castellano, como por ejemplo [una guía completa del overclocking](#), un poco atrasada; también se puede encontrar una [guía bastante completa en castellano](#) y la [sección de overclocking](#) de Batch-PC.

3.5 Sintonización de la memoria

La actividad de la memoria consiste principalmente en swapping, o escritura de un proceso completo, con toda la memoria correspondiente, en disco, y paginación, en la cual se cargan o descargan las páginas de un proceso que son necesarias en cada momento. Un proceso pagina por el simple hecho de cargarse en memoria, por ello es normal que las páginas cargadas en un sistema se mantenga siempre en un valor mayor que 0. Si un sistema empieza a estar algo falto de memoria, empiezan a escribirse páginas en disco (salvo que se trate de copy-on-write, en cuyo se escribe en disco simplemente porque se ha modificado, o viceversa). Cuando el sistema está muy escaso de memoria, o bien cuando un proceso está vagueando, esperando a e/s (por ejemplo, entre dos pulsaciones de tecla), se le envía a disco; sólo en caso de pánico se recurre al swapping de desesperación, cuando un proceso ejecutándose se envía a disco. Si ocurre esto, tío, tienes un problema.

Las prestaciones de memoria, incluyendo memoria virtual, se vigilan con `vmstat` o `sar -w y -p`. Los parámetros a vigilar son page-in, pero sobre todo page-out y swap-out.

Cuando se detecta un problema, hay diversas técnicas para evitarlos. La que se le ocurre a uno inmediatamente es, caray, comprar más memoria; hoy en día va barata y hay que aprovechar el momento. Hay también otros modos de hacerlo no tan drásticos (u onerosos):

- Limitar el tamaño de los procesos; esto lo puede hacer el superusuario, y, generalmente, fastidiará a todo bicho viviente, especialmente usuarios de Mozilla y OpenOffice.
- Animar a la gente a usar librerías compartidas. La mayoría de los sistemas operativos actuales las tienen (.dll en Win95/NT, .so en UNIX), y a diferencia de las librerías estáticas, sólo tiene que haber una copia en el sistema.
- Modificar el algoritmo de paginación. A no ser que seas un administrador que aprendiste con Kernighan y Richie y seguiste con Linus Torvalds, mejor que no toques estas cosas. Se pueden, sin embargo, cambiar parámetros tales como los que aparecen `/proc/sys/vm/bdflush`, y experimentar con el sistema.
- Cambiar el tamaño de la partición de swap; esto aumenta la memoria virtual disponible; aunque no va a aumentar las prestaciones del sistema, permite que se

haga más paginación antes de que el sistema empiece a hacer swapping de pánico. En la mayoría de los casos, hacerlo es bastante complicado. Algunos sistemas modernos usan paginación de filesystem, en el cual se usan ficheros normales para hacer swapping en vez de particiones dedicadas, lo cual tiene el problema de que cuando el disco está lleno, el sistema se cuelga, sin más.

- Usar herramientas de gestión de prioridad por proceso, tales como `pricntl` de Solaris. Con esta herramienta se pueden definir clases de proceso y prioridades y quantum temporales para cada clase. Si algún grupo de procesos se pone pesado, se le puede dar caña con esto.

3 Ejercicios de autoevaluación

1. Examinar un sistema que tengamos a mano, y en el cual tengamos privilegio de administrador, cambiar alguno de los parámetros relativos a la memoria, y ver como influye en las prestaciones del sistema; para evaluarlo, usar un monitor antes y después del cambio.
2. Consultar en internet o en los manuales del sistema operativo cuáles son los parámetros relativos a la memoria modificables por el usuario y administrador, y decir qué posible impacto pueden tener en las prestaciones del sistema.

3.6 Mejora de prestaciones en entrada/salida

Aunque el subsistema gráfico tiene su importancia, no es algo de lo que el administrador del sistema se pueda preocupar, ya que afecta más a las prestaciones de un ordenador local que a las del sistema en general; lo mismo ocurre con impresoras, que siempre van a funcionar mal, así que no merece la pena preocuparse por ellas; lo que sí afecta las prestaciones del sistema en general son los discos duros, y en ellos nos vamos a fijar en este apartado.

Los discos duros incluyen tanto los locales como aquellos accesibles mediante NFS, el sistema más popular para acceso a filesystems remotos (junto con Samba, si la configuración incluye máquinas con Windows). La eficiencia de estos sistemas estará en tres factores diferentes: throughput por proceso, throughput total, y eficiencia en el almacenamiento.

El optimizar los primeros dos factores es hasta cierto punto compatible, si un sistema tiene unas buenas prestaciones por proceso, también lo serán las totales, pero no necesariamente. Un proceso suele acceder a un solo fichero en un solo disco duro, mientras que el sistema en total accede a muchos ficheros simultáneamente en muchos discos duros, o, lo que es peor, en el mismo. El que interese más uno u otro factor dependerá del uso prioritario de cada disco duro y del sistema en total.

El tercer factor, eficiencia en el almacenamiento, es incompatible con el throughput; si se trata de optimizar el throughput disminuye la eficiencia en el uso del disco. Por ejemplo, incrementar el tamaño del bloque hace que vaya más rápido a la hora de leer o escribir, pero aumenta el número de bytes que cada fichero ocupa, y viceversa. Sin embargo, UNIX tiene la ventaja de que diferentes filesystems pueden tener diferente tamaño de bloque.

3.6.1 Consideraciones sobre la configuración.

Cada disco va enganchado a una controladora de disco que es determinante de sus prestaciones. En muchos casos, a cada controladora van conectados varios discos; cada uno de esos discos tiene que "compartir" la velocidad de transferencia de la controladora. A su vez, la controladora va conectada al bus del sistema. Todos estos elementos influirán en las prestaciones del disco.

Las dos controladoras más habituales hoy en día, y no sólo para discos duros, son la última versión de la SCSI (Small Systems Computer Interface), que creo que es la Ultra Wide SCSI y SCSI-3, y la EIDE (Extended Independent Drive Electronics), también llamada ATA (AT Attachment); la última es la Ultra ATA/100, o ATA-4, aunque hay también una especificación Serial ATA, que simplifica la conexión reduciéndola a 4 cables, en vez de los 20 o 30 actuales. Casi todos los ordenadores "personales" llevan alguna versión de la IDE, mientras que todas las estaciones de trabajo suelen

llevar SCSI. En ambos casos, se pueden conectar varias unidades a cada controlador, hasta 10 en el caso de SCSI, y 4 en el caso de la IDE. El conectar diversos dispositivos al mismo conector habitualmente afecta las prestaciones del conjunto ya que las peticiones de lectura o escritura y su respuesta deben de viajar por el mismo conjunto de cables. En cualquier caso, un administrador tiene poca elección sobre el controlador a usar, salvo que si hay muchos discos duros conectados a un controlador, se puede pedir otro.

Últimamente se están poniendo de moda otro tipo de controladores: los [USB](#) y los [FireWire](#) (o IEEE 1394), dos tipos de buses serie que permiten conectar todo tipo de dispositivos externos al ordenador, y en particular discos duros. Tienen la ventaja de ser Plug'n'Play, es decir, nada más enchufarlos el sistema es consciente de ellos. Aunque actualmente son más productos de consumo que para servidores, es posible que en el futuro se popularicen

En cuanto al segundo factor, el bus, sucede lo mismo: PCI es el más usado en el segmento de los PCs, y diferentes buses propietarios en los servidores y estaciones de trabajo. En algunas estaciones de trabajo hay diferentes buses; por ejemplo, PCI y alguno propietario, como Mbus, cada uno con sus características. La elección del bus al que se va a conectar el disco duro es previa a la compra del mismo, y de la controladora que va con él; pero en la mayoría de los casos tendrá uno de conformarse con lo que haya.

Asimismo, habrá que elegir a qué ordenador se van a conectar los discos duros; aunque el conectar todos los discos duros a un solo servidor hace la gestión mucho más fácil, el que cada grupo de trabajo tenga un disco duro normalmente aumenta las prestaciones. Cuando se haya tomado una decisión, y esté funcionando, el análisis de prestaciones del sistema nos revelará si la elección es adecuada o no.

Otro factor es la organización del sistema de discos duros: almacenamiento en red, RAID, discos duros distribuidos en diferentes ordenadores... La organización distribuida tiene la ventaja de no crear cuellos de botella, mientras que el almacenamiento en red o en un sólo ordenador tiene más facilidad de mantenimiento.

Los siguientes factores son las características físicas del disco duro. Hay dos parámetros principales: la velocidad de transferencia, y el tiempo medio de búsqueda; ambas están relacionadas con la capacidad del disco. Habitualmente, a mayor capacidad, mayor velocidad de transferencia y menor tiempo medio de búsqueda. También están relacionados con la velocidad de rotación del disco duro; hoy en día hay discos que giran hasta a 10000 RPM, mientras que hace unos años todos giraban a 3600 RPM. La capacidad máxima actualmente viene a ser de unos 200 GBytes. Para una guía actualizada, consultar la [guía de Tom's Hardware de dispositivos de almacenamiento](#).

De esos dos, el más importante es el tiempo de búsqueda. El tiempo de búsqueda es bastante no lineal, y depende del sitio donde se encuentra la cabeza y donde tenga que ir; normalmente se especifica el tiempo mínimo de búsqueda. Esto es cierto principalmente en entornos multiproceso/multiusuario, donde es más habitual que el cabezal del disco esté saltando de allá acá buscando diferentes ficheros, que el que tenga que leer un fichero grande y se le deje en paz mientras lo haga. Habitualmente, la tasa de tiempo empleado buscando con respecto al tiempo empleado transfiriendo es de 10 a 1; por eso es interesante que sea lo menor posible. Lo más habitual es que esté por debajo de los 10 ms., llegando a 6.3 ms. o incluso menos en algunos casos (por ejemplo, [IBM Storage Systems Ultrastar 18XP](#)).

En cuanto a la tasa de transferencia, aunque de ella hay que descontar toda la información de formateo y el tiempo que se tardaría en saltar de una pista a otra, suele ser un buen indicador de throughput para un solo proceso.

En resumen, si hay que comprar un nuevo sistema de E/S, dado que hay poca elección en cuanto al controlador y al ordenador que hay que conectar, lo más importante es el tiempo mínimo de búsqueda. Para ver más información sobre el tema, ver el [artículo de Epinions sobre el tema](#).

3.6.2 Particiones y filesystems

En la mayoría de los casos y de los sistemas operativos actuales, hay varios filesystems donde elegir; como mínimo, hay un filesystem "antiguo", que se usa por razones de compatibilidad, y uno "nuevo"; incluso puede darse el caso de que un mismo sistema operativo tenga diferentes filesystems, uno en cada versión, y que un mismo ordenador tenga diferentes filesystems en un mismo disco duro incluso. Ahí van algunos de ellos:

- En el mundo Wintel, los más habituales son el FAT (el que se usaba en versiones antiguas de Windows y MSDOS), el FAT32, una versión más actualizada, y el NTFS, que es el que usa Windows NT. Cada uno de ellos tiene sus limitaciones y ventajas; por ejemplo, los antiguos tenían limitaciones en cuanto al tamaño de discos duros que podían manejar. Los mejores son el NTFS y el sistema que viene con las últimas versiones del Win95, el llamado VFAT; son virtualmente iguales. El problema con el NTFS es su incompatibilidad con Win95.
- En el mundo UNIX, hay muchos filesystems. Para empezar, está NFS, pero se trata únicamente de cómo los ordenadores en la red ven los filesystems de otros ordenadores; otro filesystem con un objetivo similar es Coda. UFS es el más habitual en las implementaciones BSD de UNIX, pero hay otros, como EFS (en Irix), ext2fs y ext3fs (en Linux; este último es el más extendido actualmente), XFS, reiserfs, Coda. En realidad, no importa demasiado, salvo en el caso de que haya varios filesystems en un solo disco duro; en tal caso lo que importa es poder acceder a un filesystem desde cualquier sistema operativo. Algunos filesystems tienen características tales como compresión, o permitir desborrado, o similares, pero, una vez más, suele haber poca elección.

En cada partición suele haber un filesystems; y un sistema UNIX suele tener muchas particiones en cada disco, aparte de varios discos en cada sistema. Por ejemplo, suele haber particiones de swap, de usuarios y de root, pero puede haber muchas más: de alumnos y de profesores, por ejemplo. Hay que seguir unas cuantas reglas a la hora de distribuir las particiones:

- Distribuir la carga de trabajo tan parejamente como sea posible.
- Mantener tipos similares de ficheros en los mismos filesystem, para que sea más fácil elegir configuraciones adecuadas para cada uno.
- Mantener proyectos o grupos dentro del mismo filesystem; así es más fácil para los usuarios.
- Dar a cada filesystem un tamaño de bloque adecuado para los ficheros que contiene. Por ejemplo, a un filesystem con una base de datos le vendrá mejor un tamaño grande de bloque, para hacer el acceso más eficiente, mientras que uno para desarrollo de software estará mejor servido con tamaño de bloque pequeño, para hacer el almacenamiento más eficiente.
- Tener el número mínimo de particiones.

A menos que uno vaya a manejar un sistema grande, estos consejos no son necesarios, pero uno nunca sabe cuando se va a encontrar con un sistema grande, o el sistema va a crecer tanto que se va a convertir en grande.

3.6.3 Equilibrio de la carga de trabajo de E/S

La herramienta que se usa para esto es iostat o sar -d (recordar que, en algunos sistemas como Solaris o Linux, las herramientas que recogen información para sar no están activadas por defecto). Iostat da un informe que, entre otras cosas, indica cuánto se ha leído y escrito en cada disco duro físico, y cuántas transacciones, o peticiones de lectura/escritura, ha habido.

Idealmente, el número de lecturas y escrituras, y el número de transacciones debe de estar repartido uniformemente entre todos los discos; pero lo más habitual es que alguno de los discos se lleve toda la carga, y otros estén la mayor parte del tiempo quietos. En caso de que eso suceda, lo que se debe de hacer es reequilibrar la carga, de la forma siguiente:

- Colocar los ficheros a los que más se acceda en los discos duros más rápidos.
- Repartir los usuarios entre diferentes discos duros, y colocar los ficheros de los usuarios que más uso tenga en los más rápidos (por ejemplo, en discos duros locales en vez de discos de red, o en los más grandes).

3.6.4 Conservando el espacio del disco duro

Es un axioma de la Naturaleza que, igual que ninguna familia llega a fin de mes por mucho que gane, cualquier disco duro por grande que sea acaba llenándose. Algunos de los trucos usables para ahorrar espacio de disco son los siguientes:

- Configurar los sistemas con tamaños pequeños de bloque, aunque eso hará que vaya todo un poco más lento.
- Usar utilidades de compresión para comprimir ficheros que no se accedan a menudo. Se puede usar la utilidad find para encontrarlos y comprimirlos automáticamente. Esto se hace en muchos automáticamente en los ficheros de registro, o logs.
- Instituir cuotas, o cuotas de cortesía. Por ejemplo, detectar cuando algún usuario usa demasiado espacio, y automáticamente mandarle un e-mail.
- Borrar ficheros que no se usan: demos, juegos, o que se usan demasiado: el IRC, juegos, etc. Borrar ficheros temporales periódicamente. Borrar los cores periódicamente.

Hay principalmente dos herramientas que sirven para ver cómo se está usando el espacio: df, que muestra todos los filesystems que hay montados, el espacio que tienen y el disponible, y el du, que muestra cuánto ocupa un filesystem, es decir, todos los ficheros y directorios que descienden de uno determinado, incluyendo directorios de usuario.

Otro programa, limit, permite también limitar el tamaño máximo de fichero y el de los coreumps; permitiendo limitar incluso los coreumps a tamaño 0. Lo primero es un poco cafre, porque puede que hagan falta ficheros grandes, y lo segundo puede ser peligroso, si se usan los cores para depurar, pero, qué diablos, a veces un administrador del sistema tiene que tomar decisiones difíciles.

4 Ejercicios de autoevaluación

1. Consultar los parámetros físicos de los discos duros del sistema personal, y decir si la configuración maestro/esclavo que se está usando es la más adecuada.
2. Consultar en internet o en los manuales del sistema operativo cuáles son los parámetros relativos al subsistema de entrada/salida modificables por el usuario y administrador, y decir qué posible impacto pueden tener en las prestaciones del sistema.
3. Evaluar para el sistema personal, y la distribución de ficheros que hay, si el tamaño de cluster sería el más adecuado. Evaluar si, cambiando el tamaño de cluster, se ocuparía menos espacio.

3.6.5 Mejora de prestaciones de disco en WinNT

Se puede medir el porcentaje de uso del fichero de paginación, porcentaje de uso de disco y demás. Puede ser que en un servidor haya algún problema, pero lo más normal es que no lo haya. En caso de que el archivo de paginación se use más del 90% del tiempo, hay algún problema. En algunos casos, usar una controladora de disco con caché puede mejorar el asunto.

3.6.6 Usando la BIOS para mejorar prestaciones

A partir de la guía encontrada en [PCGuide](#), se puede hacer lo siguiente para mejorar las prestaciones de un disco duro:

- Permitir "block mode": permite hacer transferencias de varios sectores simultáneamente, con una sola interrupción.
- Habilitar el modo de transferencia de 32 bits, lo cual da una pequeña mejora; en este caso, se envían los bloques de 16 bits, que son los que admite el ATA, de 2 en 2.
- Optimizar los modos PIO (entrada salida programable), poniéndolos a 4, que es el máximo admisible.
- Cambiar el orden de los dispositivos IDE, los más rápidos, por ejemplo, deberían actuar como "master".

3.7 Optimización de un servidor web

En el libro [Web Performance Tuning](#) dan una serie de consejos para optimizar las prestaciones de un servidor web:

- Desconectar consultas DNS inversas: en la configuración del servidor hay una línea que dice algo así como:

```
HostnameLookups Off
```

, bueno, pues conviene dejarla tal como está. Si no se hace, por cada uno que se meta en nuestro servidor, se va a mirar cuál es su dirección DNS, lo cual puede retrasar cada petición algunos segundos. Lo que ocurre es que luego queda mu chulo en las estadísticas.

- Incrementar el tiempo de timeout del TCP: la línea `Timeout 300` se puede cambiar por una cantidad mucho mayor para evitar retransmisiones.
- Buscar un servidor cerca de los usuarios potenciales: en caso de que haya muchas peticiones, conviene colocar mirrors en los sitios donde los usuarios abundan más.
- Incrementar la RAM: pos eso.
- Actualizar las versiones del servidor y del sistema operativo.
- Dedicar el servidor exclusivamente a eso
- Usar APIs del servidor en vez de SSI (server-side includes) o CGI (common gateway interface).
- Preprocesar el contenido dinámico, para evitar al servidor el trabajo de generarlo
- Usar un profesional. Sin comentarios.

3.8 Mejora de prestaciones de sistemas Windows XP

En el artículo "[Optimize Windows XP](#)", de Diciembre de 2001, vienen una serie de consejos para optimizar las prestaciones de Windows XP.

- Para empezar, al parecer Windows XP se optimiza solo: cada tres días, después de observar el comportamiento del usuario, cambia de sitio en el disco los programas más usados para que su lanzamiento sea más rápido (Supongo que esto no funcionará en discos en red). Los cambios los mete en un fichero denominado `layout.ini`.
- Eliminación de "chuminás": todo lo que mejore la apariencia del sistema, puede ser eliminado para mejorar las prestaciones: quitar iconos, efectos, skins y demás hará que vaya más rápido. Dejar de usar ClearType hará que las fuentes en pantalla se vean más feas, pero todo irá más rápido.
- Tenerlo siempre actualizado. Aunque esto es necesario prácticamente para que funcione, siempre viene bien aplicar los últimos parches; además, WinXP puede hacerlo automáticamente.
- Desfragmentar de vez en cuando.
- Usar `MSCONFIG` para ver qué programas se arrancan al comienzo del sistema, y eliminar los innecesarios. Algunas utilidades, como Startup Cop, buscan los programas que se inician automáticamente en más sitios. Eliminar también los servicios que comienzan automáticamente.
- Cambiar las prioridades de algunos procesos

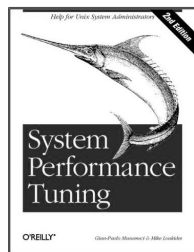
5 Ejercicios de autoevaluación

1. Sobre un sistema que se tenga a mano, llevar a cabo una monitorización usando las técnicas explicadas en este tema, proponer mejoras, llevarlas a cabo, y volver a monitorizar indicando dónde han impactado las mejoras.

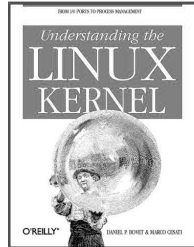
3.9 Bibliografía y enlaces en internet

3.9.1 Bibliografía

System Performance Tuning, 2nd Edition, de Gian Paolo Musumeci y Mike Loukides, O'Reilly, sustituye a la edición anterior, que estaba un tanto obsoleto, pero que contiene todas las técnicas básicas para analizar las prestaciones de los diversos subsistemas de un ordenador: memoria, disco, red. Está actualizado hasta la versión 2.4 del kernel de Linux, y las últimas versiones de Solaris; todas las herramientas y pruebas están presentadas sobre estas dos plataformas.

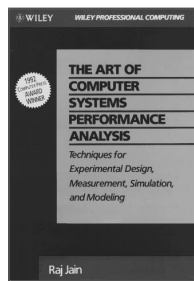


Understanding the Linux Kernel, de Daniel P. Bovet y Marco Cesati, de O'Reilly. Un libro que analiza uno por uno las diferentes partes del kernel de Linux. Es un libro muy reciente, que cubre la versión 2.2 del kernel, con pequeñas observaciones al final de cada capítulo sobre el kernel 2.4. Especialmente interesantes para esta asignatura son los capítulos sobre procesos (capítulo 3) y sobre scheduling de procesos (capítulo 10), donde se explican las estructuras de datos usadas para la gestión de procesos, y los algoritmos para asignación dinámica de prioridad a los procesos, incluyendo su ejecución en sistemas multiprocesador.

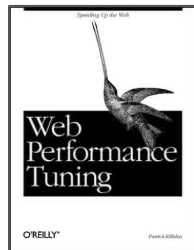


Es un libro imprescindible para entender la labor del administrador del sistema que debe, al fin y al cabo, entender el kernel de Linux para que su sistema le saque el máximo partido posible. Por supuesto, es sumamente útil para los que estén interesados en crear sus propias versiones del kernel, o simplemente usarlo como documentación para una asignatura de sistemas operativos.

The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation and modelling, Raj Jain, Wiley, 1992. Por la fecha de publicación, se ve que tiene el mismo problema que el primero; sin embargo, tiene un enfoque ameno y completo al problema del análisis de prestaciones.



Web Performance Tuning: Speeding Up the Web (O'Reilly Nutshell) by Patrick Killelea (2nd edition); es un poco básico y pierde mucho tiempo en cosas tales como optimizar el cliente, pero tiene buenos consejos sobre como optimizar las prestaciones de un servidor web. La edición anterior está [disponible online en Safari](#) (completa).



3.9.2 Enlaces

También se puede encontrar información por la misma Internet, aparte de la mencionada en el texto. Algunos sitios:

- [Recursos para el administrador de sistemas](#), de O'Reilly. Contiene enlaces a todos los libros de O'Reilly sobre el tema y artículos periódicos de sus autores.
- [TweakTown](#), un sitio dedicado básicamente a pruebas de hardware y software, pero que contiene foros y guías para optimización e instalación de diferentes cosas: cortafuegos, conexiones de banda ancha... Sobre todo enfocado a sistemas basados en Windows
- El sitio de referencia de mejora de prestaciones para Linux es [Linux Performance Tuning](#), que se renueva periódicamente, y tiene consejos generales para mejora de prestaciones, así cómo para mejora de servicios tales como páginas web o correo.

Realizado por
Juan Julián Merelo Guervós [jmerelo at geneura.ugr.es](mailto:jmerelo@geneura.ugr.es)
Depto de Arquitectura y Tecnología de Computadores
Universidad de Granada
URL:<http://geneura.ugr.es/~jmerelo/DyEC/Tema3/DyEC-Tema3.html>