

El Nuevo Sistema de Ficheros Ext4

EL FUTURO



Julien Tromeur, Fotolia

El último hijo de la familia de sistemas de ficheros Ext proporciona un mejor rendimiento y soporte para grandes sistemas de ficheros. ¿Estás preparado ya para Ext4?

POR MARCEL HILZINGER

Linux llegó a la madurez con la presentación del sistema de ficheros Ext2, siendo Ext3 el primer sistema de ficheros con diario que se incluyó en el kernel. Con el paso de los años, la familia de sistemas de ficheros Ext (Extendido) ha ocupado un lugar destacado en Linux y continúa siendo una elección importante por parte de los usuarios de todo el mundo. Pero el venerable código base de Ext presenta ya signos de vejez. Hace unos meses, los desarrolladores comenzaron a discutir el problema de añadir nuevas características a Ext3 a la vez que mantenían la estabilidad y compatibilidad del código [1]. Mientras tanto, varias iniciativas ofrecieron mejoras al sistema de ficheros demasiado fundamentales como para integrarlas fácilmente y demasiado importantes como para ignorarlas.

La solución propuesta por los desarrolladores fue generar una rama nueva de desarrollo del código y comenzar a trabajar en un nuevo sistema de ficheros denominado Ext4. Esta estrategia permite a los usuarios de todo el mundo continuar utilizando la

versión Ext3 estable mientras que los desarrolladores integran y prueban las nuevas mejoras. Ext4, la próxima generación del sistema de ficheros Ext, ya se encuentra disponible. Vamos a echarle un vistazo para ver cómo se comporta con respecto a otros sistemas de ficheros.

¿Qué es lo que Cambia?

Ext4 dispone de un soporte mejorado para 64bits gracias al cual se obtienen otras mejoras. Uno de los principales beneficios que aporta Ext4 es un incremento en el tamaño máximo del sistema de ficheros. El de Ext3 es de 8TB (con un tamaño de bloque por defecto de 4KB), lo que se consideraba enorme hace algún tiempo, aunque el hardware actual soporta más espacio. Ext4 soporta sistemas de ficheros de hasta 1024 petabytes (PB).

Otro cambio importante en Ext4 es el uso de *extensiones*. Una extensión es una zona contigua de espacio en disco adicional reservada junto con un fichero. El propósito de esta extensión es asegurarse de que los

cambios posteriores que se realicen en el fichero se almacenen junto al fichero original. Las extensiones reducen la fragmentación del disco y mejoran la velocidad de escritura. Otros sistemas de ficheros populares, como Reiser4, NTFS y el sistema de ficheros HFS de Macintosh ya soportan las extensiones.

La compatibilidad hacia atrás fue también uno de los objetivos de los desarrolladores de Ext4: Se puede montar una partición Ext4 como Ext3 y se puede montar una partición Ext3 como Ext4 (con el tipo de sistema de ficheros ext4dev). Desafortunadamente, para conseguir esta compatibilidad Ext3/Ext4 hay que desactivar la característica de las extensiones de Ext4, que es uno de sus principales beneficios con respecto al rendimiento de Ext4.

Si estamos interesados en probar Ext4, necesitaremos un kernel actual y las herramientas típicas de desarrollo: make, GCC y glibc-devel. En nuestro laboratorio probamos el kernel 2.6.20. Como root, se descomprimen los ficheros fuente en el directorio

`/usr/src` y se crea un enlace de `/usr/src/linux` que apunte a los fuentes:

```
cp linux-2.6.20.tar.bz2 /usr/src
cd /usr/src
tar xvfj linux-2.6.20.tar.bz2
ln -s linux-2.6.20 linux
```

Para que el kernel sepa qué elementos se quieren compilar como módulos hay que crear el fichero `.config` en `/usr/src/linux`. Para ello, o bien se ejecuta el comando `make menuconfig` o el comando `make xconfig`. Como el sistema manual con su multitud de opciones no es algo sencillo, es preferible utilizar el kernel actual como la base de la nueva configuración y añadir sólo las diferencias para la nueva versión del kernel. Para ello, hay que copiar el fichero `/boot/config-kernelversion` a `/usr/src/linux` y guardarlo como `.config`. El siguiente comando debería realizar esto para la mayoría de las distribuciones:

```
cp /boot/config$(uname -r) \
/usr/src/linux/.config
```

De nuevo como root, hay que teclear el comando `make oldconfig` en el directorio `/usr/src/linux` para crear un kernel nuevo a partir del fichero de configuración existente. Para cada cambio, el sistema preguntará al usuario si está seguro del mismo y cómo desea implementarlo. Dependiendo de la versión del kernel, podría tener que responder a diferentes cuestiones. Si sólo se desea añadir Ext4, únicamente hará falta pulsar la tecla Enter para aceptar las opciones predefinidas. Hay que tener en cuenta que el sistema no utilizará Ext4 por defecto. Para modificarlo hay que abrir el fichero `.config` y buscar la siguiente entrada:

```
# CONFIG_EXT4DEV \
_FS is not set
```

Se deberá eliminar el carácter almohadilla del comienzo de la línea y cambiar la entrada a `CONFIG_EXT4DEV_FS = m` (Figura 1). Para experimentar con Ext4 en la partición raíz hay que compilar los ficheros fuente en el kernel en vez de utilizarlo como un módulo. En este caso, es preciso cambiar la entrada a `CONFIG_EXT4DEV_FS=y`, ejecutar de nuevo `make oldconfig` y luego responder algunas cuestiones sobre Ext4 (Figura 2).

La mayoría de las distribuciones actuales utilizan listas de control de acceso (ACLs) y atributos extendidos para soportar un control

más granular sobre los privilegios de los usuarios. Si la salida del comando `mount` muestra entradas para `acl` o `user_xattr` en la línea de comandos, habrá que habilitar estas características también en Ext4. Así que cuando el sistema pregunte por la activación de `EXT4DEV_FS_XATTR` y `EXT4DEV_FS_POSIX_ACL`, habrá que teclear `y`.

Para soportar SELinux hacen falta las etiquetas de seguridad (`EXT4DEV_FS_SECURITY`). Por otro lado, hay que asegurarse de desactivar el soporte de depuración JBD2, ya que ralentizará el sistema de ficheros. `make` habilita el soporte de depuración para JDB (`JDB_SUPPORT`) de manera predeterminada.

Tras tomar las decisiones oportunas, será necesario teclear `make oldconfig` para escribir el nuevo fichero `.config`, que ya posee el soporte para Ext4. Luego hay que ejecutar `make` para comenzar a compilar el kernel. Dependiendo de lo rápida que sea la máquina, este proceso puede llegar a tardar un par de horas. Tras completarse la compilación del kernel, es imprescindible introducir los siguientes comandos para instalar el módulo y el kernel:

```
make modules_install
make install
```

La mayoría de las distribuciones configurarán el ramdisk inicial (`initrd`) en este punto, y además modificarán el gestor de arranque.

Cuando se reinicie el sistema, hay que asegurarse de seleccionar el nuevo kernel del menú de Grub. La instalación del nuevo kernel no afectará a la instalación existente.

Trabajando con Ext4

Para crear una partición Ext4 hace falta una versión reciente de E2fsprogs-Tools [2]. Los usuarios de SUSE Linux

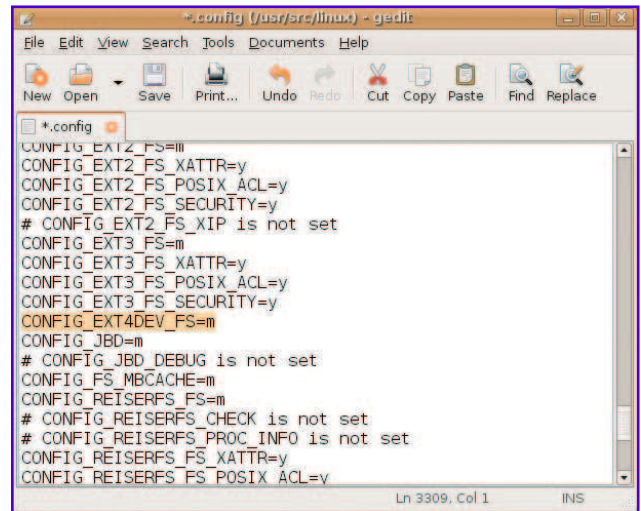


Figura 1: Para asegurarnos de que el kernel se entienda con Ext4 debemos activar la opción correspondiente en el fichero `.config`.

encontrarán la versión actual del kernel de SUSE 2.6.20 y el correspondiente E2fsprogs en suse.com [3] o en openSUSE.org [4]. Cuando se instala este kernel, hay que asegurarse de no introducir el comando `rpm -Uvh`, sino el comando `rpm -ivh` para mantener el kernel openSUSE 10.2.

Para instalar Ext2-Tools se siguen los pasos normales: `./configure`, `make`, `make install` como root. Si se produce algún error en el paso del `make`, probablemente necesite instalar el paquete `textinfo`. Tras instalarlo sólo habrá que volver a ejecutar de nuevo el comando `make`.

Para crear una partición nueva con formato Ext4, tecleamos `mke2fs -j <partición>`, reemplazando `partición` por el nombre del fichero de dispositivo correspondiente. Incluso se puede utilizar la GUI para crear una nueva partición Ext4. Tras crearla, hay que montar la partición con el comando `mount -t ext4dev <partición> <punto_de_montaje>`. El kernel cargará automáticamente el módulo `ext4dev`.

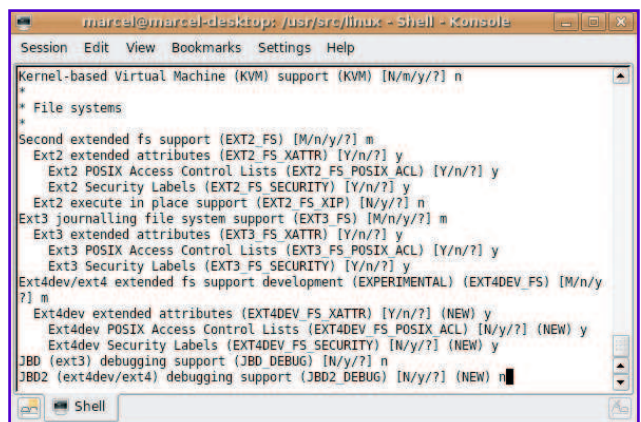


Figura 2: La segunda vez que se ejecuta `make oldconfig` sólo hay que contestar a preguntas sobre Ext4.

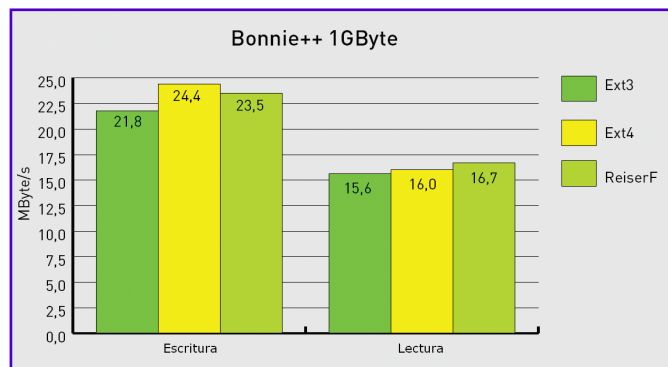


Figura 3: Ext4 escribe un fichero de 1Gb más rápido que ReiserFS en operaciones de escritura de bloques y es mucho más rápido que Ext3.

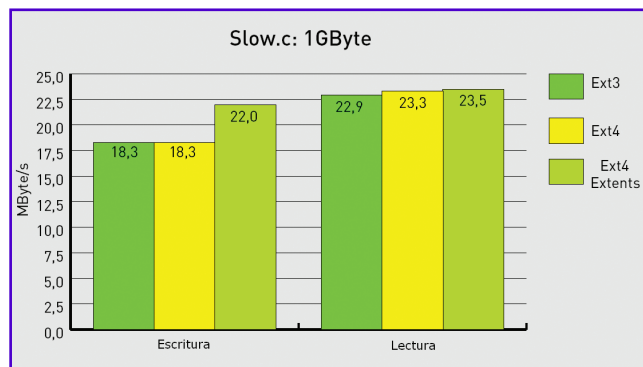


Figura 5: Sin extents, Ext4 es muchísimo más lento y su nivel es comparable con Ext3.

Si se desean utilizar las características de extensiones de Ext4, habrá que introducir `-o extents` junto con el comando `mount`. Hay que tener en cuenta que si se monta una partición con extensiones, posteriormente no podrá montarse de nuevo como Ext3. Habremos de asegurarnos de no montar las particiones Ext3 existentes con la opción `-o extents`. Si se hiciera, deberemos desmontar la partición sin modificar ningún fichero, porque de lo contrario, sólo podremos montar dicha partición como Ext4 en el futuro. Los desarrolladores tienen como objetivo migrar el soporte para las extensiones a las herramientas `mke2fs` y `tune2fs`, con lo que eliminará la necesidad de añadir el parámetro al comando `mount`.

Además de estas opciones, Ext4 soporta los parámetros normales de `mount` para los sistemas de ficheros con diario. Si tenemos un ordenador portátil, la opción `-o commit = segundos` puede ser interesante. El intervalo de tiempo que se especifica determina la frecuencia con la que Ext4 escribirá los datos al disco. El valor por defecto es de cinco segundos. Valores más altos mejorarán el rendimiento e incluso ahorrarán energía. Pero hay que tener en cuenta que se comprometerá la seguridad de los datos que

aún no se hayan escrito. Nuestras pruebas revelaron que no hay ninguna diferencia significativa cuando se establece este valor a `commit = 60`.

Para nuestra primera prueba con la versión de desarrollo de Ext4 utilizamos los bancos de pruebas `Bonnie++` [5] y `Slow.c` [6]. La distribución que se eligió fue la openSUSE 10.2 con un kernel 2.6.20 y `ext4dev` como módulo. Para las pruebas utilizamos la opción de `mount -o extents` para Ext4, sin especificar nada más, y se utilizaron los valores por defecto para el resto de las opciones del sistema de ficheros.

Ext4 alcanzó unas velocidades de escritura muchísimo más rápidas que sus predecesoras, constituyéndose en un competidor directo de ReiserFS (Figura 3) en las operaciones de escritura de bloques por primera vez, aunque ReiserFS aún lee ligeramente más rápido que Ext4. Ext4 no es mucho más rápido que Ext3 cuando tiene que crear o borrar ficheros secuencialmente (Figura 4).

El efecto de la mejora de rendimiento debido a las extensiones se demuestra claramente con los resultados obtenidos por la prueba "Random Create". Ext4 pierde más tiempo que Ext3 porque necesita tiempo

extra para reservar el espacio para las extensiones. Este trabajo extra no es necesario a la hora de borrar un fichero, siendo aquí donde Ext4 supera a Ext3 y a ReiserFS, como podemos comprobar con la prueba de estrés de `Bonnie++`.

Por último, quisimos conocer la velocidad de Ext4 sin el uso de las extensiones. La Figura 5 muestra el resultado del banco de pruebas `Show.c` con un fichero de prueba de 1Gb. Los resultados obtenidos revelan que el rendimiento excepcional que posee Ext4 se debe principalmente al uso de las extensiones.

Conclusiones

Ext4 es un sistema de ficheros con diario rápido y con vistas al futuro que aún se encuentra en una etapa temprana de desarrollo. Aunque la carencia de una compatibilidad completa hacia la versión Ext3 podría ocasionar una confusión ocasional, esto no es razón para no instalar Ext4 en un sistema en producción. Sin embargo, posiblemente prefiramos seguir con Ext3 mientras los desarrolladores de Ext4 se refieran al módulo del kernel como `ext4dev`. ■

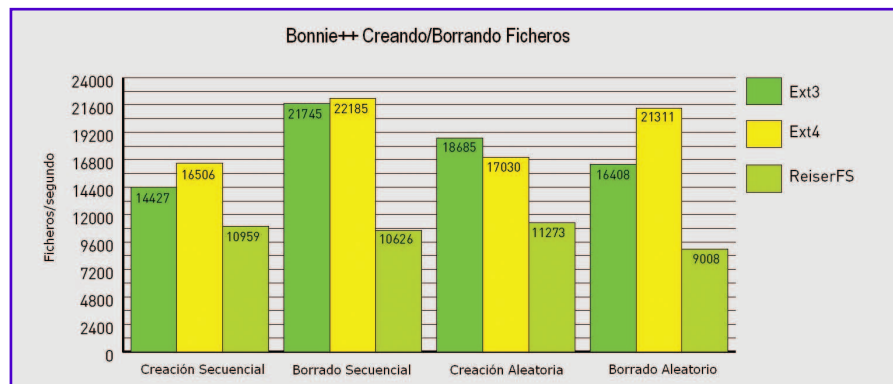


Figura 4: Un test de estrés especial de Bonnie++ evalúa el número de ficheros que puede crear o borrar el sistema de ficheros por segundo. Ext4 consigue la puntuación más alta.

RECURSOS

- [1] Propuesta de Ext4: <http://lkml.org/lkml/2006/6/28/454>
- [2] E2fsprogs: <ftp://ftp.kernel.org/pub/linux/kernel/people/tytso/e2fsprogs-interim/>
- [3] SUSE Kernel: <ftp://ftp.suse.com/pub/projects/kernel/kotd/i386/HEAD>
- [4] SUSE E2fsprogs: <http://download.opensuse.org/distribution/SL-OSS-factory/inst-source/suse/i586/>
- [5] Bonnie++: <http://www.coker.com.au/bonnie++/>
- [6] Slow.c: <http://www.jburgess.uklinux.net/slow.c>