Discos Rígidos con Sectores de 4KB

El porqué de su existencia y su impacto en Linux

Autor: Marcelo Fidel Fernández – Junio de 2010

marcelo.fidel.fernandez@gmail.com – http://blog.marcelofernandez.info

Resumen

Las compañías fabricantes de discos magnéticos, duros o también denominados rígidos, siempre han intentado estar a la vanguardia en lo que respecta al almacenamiento de datos digitales para su uso en equipos de cómputo. Mayor confiabilidad, capacidad, velocidad y menor ruido son los parámetros más importantes de estos productos. Es por eso que la búsqueda de mejora constante en las variables mencionadas involucra eventualmente cambios que afecta no sólo el hardware ya existente, sino el software mismo ya en uso masivamente. En este artículo se analiza el uso de sectores de mayor tamaño, 4096 Bytes frente a los ya clásicos 512 Bytes, y su impacto en el Sistema Operativo Linux.

Palabras Clave: informática, almacenamiento, discos rígidos, linux

Introducción

Desde los primeros días del uso de los discos rígidos[1], la mínima unidad de almacenamiento direccionable fue el sector, de 512 Bytes de capacidad. Esta granularidad se debió a que siempre fue un buen balance entre la fragmentación interna[2] de los archivos[3] y el manejo físico del disco relativo a la corrección de errores, *flags* de inicio/fin del sector más la separación inter-sector (*gap*). Toda la industria de la Computación y el software creado para ella se apoyó en este estándar de facto, y casi ningún utilitario, BIOS, ni Sistema Operativo se pensaron para un posible cambio… sólo hasta hace unos meses.

Western Digital es una de las primeras marcas en sacar al mercado discos con sectores 8 veces más grandes que los anteriores, de 4 KBytes (4096 Bytes); estos discos son etiquetados como que poseen "*Advanced Format Technology*" ("Tecnología de Formato Avanzado"), y es lo primero que se debería revisar al trabajar con discos de gran tamaño (1 TB o superior), ya sea de WD o de otros fabricantes.

El motivo del cambio, según este artículo[4], se origina en la existencia de tres factores esenciales que deben compensarse para buscar capacidades de almacenamiento cada vez mayores en el mismo tamaño de disco (que por lo general hoy en día es de 3,5 pulgadas):

1. **Densidad de área**[5]: Indicador de cuántos bits se pueden guardar en un área determinada (Bits/Pulgada cuadrada). Hasta ahora, con la técnica de Grabación Perpendicular[6], se está en el orden de unos 300 a 500 Gigabits por pulgada cuadrada[7][8].
2. **Relación Señal/Ruido**[9]: Al leer de los platos del disco, pueden ocurrir

1 http://en.wikipedia.org/wiki/Cylinder-head-sector#cite_note-0
2 Artículo sobre fragmentación: http://elezeta.net/2004/09/16/fragmentacin-de-que-se-trata/
3 En tiempos donde la capacidad de los discos era muy pequeña comparada con los de ahora, el tamaño de un cluster del sistema de archivos era igual al de un sector.
4 Artículo sobre la transición a 4 KB: http://www.anandtech.com/show/2888
5 Densidad de Área: http://en.wikipedia.org/wiki/Memory_storage_density
6 Grabación Perpendicular: http://en.wikipedia.org/wiki/Perpendicular_recording
7 http://arstechnica.com/hardware/news/2006/09/7765.ars
8 http://arstechnica.com/science/news/2010/05/new-hard-drive-write-method-packs-in-one-terabyte-per-inch.ars
9 Relación Señal/Ruido: http://en.wikipedia.org/wiki/Signal-to-noise_ratio

fallos, ya que el almacenamiento magnético en definitiva es analógico; y la señal, para ser convertida desde/hacia binario, debe ser procesada en forma acorde. Cuanto mejor sea la relación de la señal con respecto al ruido en el momento de leer o escribir en los platos, más confiable es la operación.

3. **El Código de Corrección de Errores – ECC**[10]: Cada sector del disco incluye un área reservada para almacenar el ECC, imprescindible para recuperarse ante cualquier error de lectura/escritura.

A medida que la densidad del área se incrementa, los sectores (siempre de 512 Bytes) lógicamente se reducen en el área que ocupan físicamente. Esto hace que se incremente el Ruido con respecto a la Señal porque las señales son más débiles y hay más interferencia de los datos adyacentes; por lo tanto el valor de SNR disminuye, y a su vez, la probabilidad de errores de lectura aumenta. Entonces, es necesario mejorar la capacidad del ECC para detectar y corregir errores, generalmente agregándole más bits. Esto requiere de más espacio físico reservado para un sector (siempre de 512 Bytes), y aquí se está nuevamente en el comienzo.

Lo que sucede es que se está llegando a un límite donde **no se puede seguir con sectores de 512 Bytes y aumentar el tamaño total del disco, ya que todo este nuevo espacio obtenido con una mayor densidad termina no siendo utilizable, sino que será mayormente para el ECC** (es decir, redundancia para contemplar posibles errores).

La solución al problema es que, para almacenar más cantidad de información en forma global, hay que incrementar la eficiencia del ECC. Y esto se logra haciendo que éste abarque más datos que sólo 512 Bytes; el ECC es mucho más eficiente (ocupa menos espacio en comparación) si su código de corrección abarca más datos, como lo son 4096 Bytes.

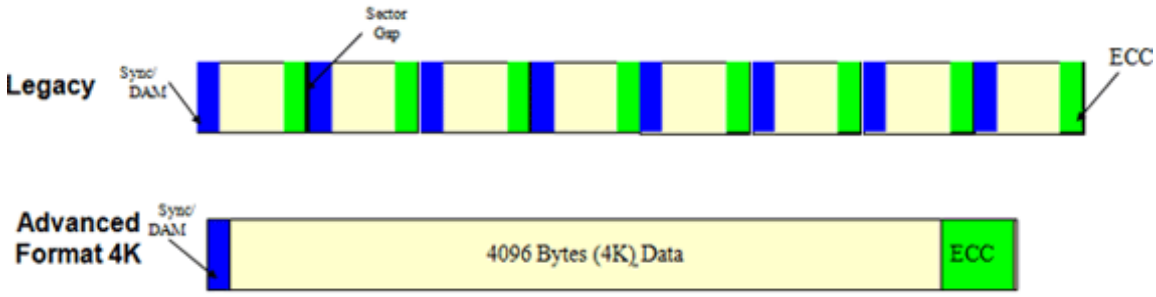
Por ejemplo, para detectar y corregir corregir 4096 Bytes divididos en 8 sectores de 512 Bytes (de la forma original), es necesario desperdiciar 320 Bytes de ECC (ya que se tienen 40 Bytes por cada ECC de 512 Bytes), mientras que si se utiliza 1 sector de 4096 Bytes sólo se necesitan 100 Bytes

10 Corrección de Errores hacia Adelante: http://en.wikipedia.org/wiki/Forward_error_correction

de ECC. Como se puede ver, se ahorran 220 Bytes de *overhead* por cada 4KB que tiene el disco para almacenamiento; en 1500 GB (= 1.500.000.000 KB / 4 = 375.000.000 sectores de 4 KB * 0,22 KB) son 82,5 GB más de espacio disponible para almacenar datos de usuario y no ECC (un 5,5% más). Y esto sin contar el espacio utilizado para los *gaps* entre sectores y el *flag* de sincronización/inicio de sector (para 4 KB de información, antes eran 8 y ahora es sólo 1). Además, estos 100 Bytes de ECC mejoran en un 50% la capacidad de detectar errores en "ráfaga" comparado con el anterior, es decir, el nuevo es un mejor y más eficiente ECC.

Por todo esto, para tamaños tan grandes de disco, usar sectores de 4KB permite aprovechar de manera más eficiente la mayor densidad del área que se dispone. ¿Y por qué 4 KB? No es un número al azar; coincide con el tamaño de las páginas de memoria en la arquitectura x86 y con el tamaño de cluster por defecto de la mayoría de los sistemas de archivos más extendidos, con lo cual la velocidad de transferencia de páginas desde/hacia el disco no se ve afectada, y la fragmentación interna de los archivos almacenados es la misma que con sectores de 512 Bytes.

Para mayor claridad, en este gráfico se ve cómo ocupan más lugar los 8 sectores de 512 Bytes puestos a la par del sector de 4 KB:



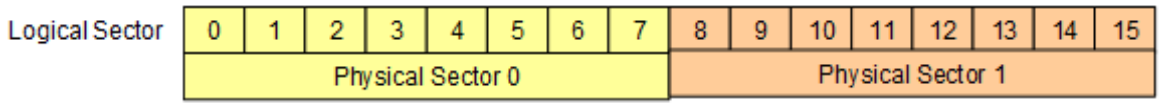
Ventajas y Desventajas del uso de Sectores de 4 KB

- Como ventaja y por lo desarrollado anteriormente, mayor confiabilidad y capacidad de almacenamiento hoy y a futuro.
- La gran desventaja es el proceso de migración, que debe afrontarse desde varias capas de software, desde el BIOS, pasando por el Sistema Operativo y llegando a las herramientas de defragmentación, clonado y

administración de discos.

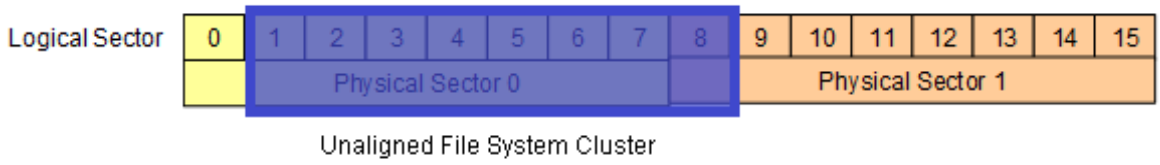
El impacto en Linux: Particiones desalineadas.

Con el objetivo de mitigar las desventajas, esta serie de discos "*Green*" de WD emula ser un disco con sectores "lógicos" de 512 bytes, pero en realidad trabaja internamente con sectores "físicos" de 4 KB. De esta manera, los sectores de 512 bytes lógicos se ven así, dentro de uno de 4 KB:



Debe recordarse que la mínima unidad "direccionable" real del disco son 4 KB (un sector), y el tamaño de cluster por defecto (la mínima unidad "direccionable" por el sistema de archivos) son 4 KB; esto quiere decir en definitiva que el disco, cada vez que lee y escribe, lo hace en unidades de 4 KB.

Luego se le agrega la capa de emulación, para hacerlo compatible con los Sistemas Operativos disponibles actualmente. Bien, supóngase que se creará una sola partición en el disco. ¿Qué pasaría si esta partición, donde se almacenarán los datos, comienza en el sector 1 (o cualquier sector no múltiplo de 8) del disco en vez del sector 0? Lo siguiente:



Lo que sucede es que **el primer cluster empieza y termina en dos sectores físicos; está "desalineado" con respecto a ellos**. En realidad, todos los clusters de la partición estarán de esta forma, comenzando y finalizando a "destiempo" respecto de los bloques físicos. El problema que esto conlleva es que si el SO va a escribir algo en el disco (donde el "algo" es como mínimo generalmente un cluster de 4 KB), supongamos el cluster resaltado en azul, esto se transforma físicamente en las siguientes operaciones, a nivel de disco:

1. Leer los 4 KB del sector físico 0.
2. Modificar los 7 sectores lógicos afectados por esta operación.
3. Grabar los 4 KB del sector físico 0.
4. Leer los 4 KB del sector físico 1.
5. Modificar el 8vo sector lógico.
6. Grabar los 4 KB del sector físico 1.

Esto es, 2 operaciones de Leer-Modificar-Grabar (RMW[11]) atómicas, una para cada sector físico, que involucra una vuelta (*spin*) de disco por cada lectura/escritura de la lista enumerada. Es decir, **que una partición comience en un sector lógico de 512 Bytes no múltiplo de 8 hace excesivamente lento el acceso al disco porque las operaciones llevan mucho, mucho más tiempo que antes**.

Y la cuestión reside aquí; si bien Linux a esta altura ya está preparado para manejar discos con sectores de 4 KB, el problema es que el disco le comunica a Linux que tiene sectores físicos de 512 bytes por la emulación, cuando internamente trabaja con 4 KB[12]:

```
marcelo@marcelo:~$ sudo hdparm -I /dev/sdb | grep Sector
        Logical/Physical Sector size:           512 bytes
```

La consecuencia de esto está en la alta posibilidad de no tener los sectores alineados. Fdisk y cualquier software particionador de discos de Linux comienza la primer partición en el sector 63 de aquellos discos que reconoce como de sectores de 512 Bytes[13]. Esto hace que el disco funcione muy lento, como se describe en este foro de soporte WD[14].

Cómo particionar estos discos en general y en Linux en particular

Bueno, ¿cómo se hace para crear particiones de manera alineada? Es

11 Operación Read-Modify-Write: http://en.wikipedia.org/wiki/Read-modify-write
12 http://wdc.custhelp.com/cgi-bin/wdc.cfg/php/enduser/std_adp.php?p_faqid=5655
13 La utilización del sector 63 corresponde a que generalmente (y por herencia histórica del modelo de direccionamiento CHS) es el primer sector del track número 1. El track 0 siempre se utilizó para el MBR / Master Boot Record.
14 Foro de Western Digital: http://community.wdc.com/t5/Desktop/Problem-with-WD-Advanced-Format-drive-in-LINUX-WD15EARS/td-p/6395

relativamente fácil. Según Ted Ts'o[15], desarrollador del núcleo Linux, en un artículo donde detalla un problema similar con los nuevos discos SSD[16] (de estado sólido), hay que ejecutar fdisk[17] con los parámetros "-H 224 -S 56 /dev/sdX", siendo /dev/sdX el disco en cuestión; esto hace que todas las particiones se creen en la sesión interactiva de fdisk en sectores múltiplos de 8. Otra opción es usar GNU Parted[18] con los parámetros "unit s", y de esta manera deja a uno configurar el primer sector de cada partición.

Como caso de ejemplo, se crearán 4 particiones. Este es un ejemplo de particiones bien alineadas, el último comando es para mostrar el tamaño de cada partición nada más:

```
marcelo@marcelo:~$ sudo parted /dev/sdb unit s print
Modelo: ATA WDC WD15EARS-00S (scsi)
Disco /dev/sdb: 2930277168s
Tamaño de sector (lógico/físico): 512B/512B
Tabla de particiones. msdos

Numero  Inicio       Fin          Tamaño       Tipo     Sistema de ficheros
Banderas
 1      56s          41959679s    41959624s    primary  ext4
 2      41959680s    46161919s    4202240s     primary
 3      46161920s    1673570303s  1627408384s  primary  ext4
 4      1673570304s  2930265855s  1256695552s  primary
raid

marcelo@marcelo:~$ sudo fdisk -lu /dev/sdb

Disco /dev/sdb: 1500.3 GB, 1500301910016 bytes
224 cabezas, 56 sectores/pista, 233599 cilindros, 2930277168 sectores en
total
Unidades = sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico / físico): 512 bytes / 512 bytes
Tamaño E/S (mínimo/óptimo): 512 bytes / 512 bytes
Identificador de disco: 0x00094da1

Dispositivo Inicio    Comienzo      Fin      Bloques  Id  Sistema
/dev/sdb1   *          56    41959679    20979812   83  Linux
/dev/sdb2        41959680    46161919     2101120   82  Linux swap / Solaris
/dev/sdb3        46161920  1673570303   813704192   83  Linux
/dev/sdb4      1673570304  2930265855   628347776   fd  Linux raid
autodetect
```

15 http://thunk.org/tytso/blog/2009/02/20/aligning-filesystems-to-an-ssds-erase-block-size/
16 Discos SSD: http://en.wikipedia.org/wiki/Solid_state_storage
17 Fdisk: http://tldp.org/HOWTO/Partition/fdisk_partitioning.html
18 Software de particionamiento GNU Parted: http://www.gnu.org/software/parted/index.shtml

```
marcelo@marcelo:~$ sudo parted /dev/sdb print
Modelo: ATA WDC WD15EARS-00S (scsi)
Disco /dev/sdb: 1500GB
Tamaño de sector (lógico/físico): 512B/512B
Tabla de particiones. msdos

Numero  Inicio  Fin     Tamaño  Tipo     Sistema de ficheros  Banderas
 1      28,7kB  21,5GB  21,5GB  primary  ext4
 2      21,5GB  23,6GB  2152MB  primary
 3      23,6GB  857GB   833GB   primary  ext4
 4      857GB   1500GB  643GB   primary                       raid
```

Nuevamente, lo más importante para que los clusters estén alineados con los sectores físicos del disco es que cada partición debe comenzar en un sector múltiplo de 8, como el 56, 41959680, 46161920 y 1673570304 de este caso.

Conclusión

Luego de intercambiar algunos e-mails al respecto con Aleksander Adamowski[19], la persona que en la lista de util-linux-ng[20] "descubrió"[21] el inconveniente de la lentitud con esta serie de discos WD a base de un lote de pruebas disponible aquí[22], él ha hecho algunas sugerencias.

Al trabajar con discos > 1 TB "sospechosos" de tener sectores de 4 KB, debe tenerse en cuenta los siguientes aspectos:

- "Las unidades de medida son críticas. Asegúrate que estás realmente operando a nivel de sectores[23]
- Después, hacer un test de performance es buena idea.
- Para esto, primero trata de crear una partición desalineada, crea un sistema de archivos, y ejecuta el benchmark de postmark[24] usando el archivo de configuración que publiqué[25]– por supuesto, modifica la opción "location" en ese archivo acorde al disco a comprobar.
- Luego, borra todas las particiones y haz lo mismo en una partición

19 Aleksander Adamowski: http://olo.org.pl/dr/cv_eng
20 Proyecto Util-Linux-NG: http://userweb.kernel.org/~kzak/util-linux-ng/
21 http://thread.gmane.org/gmane.linux.utilities.util-linux-ng/2926
22 Lote de pruebas y ejemplos: http://olo.org.pl/files/hw/postmark-automated/
23 Esto porque por defecto las herramientas de particionamiento Linux no trabajan con sectores; fdisk trabaja con cilindros por ejemplo.
24 Herramienta Postmark: http://www.shub-internet.org/brad/FreeBSD/postmark.html
25 http://olo.org.pl/files/hw/postmark-automated/postmark-quick.conf

alineada. Los resultados deben ser mucho mejores en cuanto al rendimiento. Si no lo son, el disco probablemente tiene sectores físicos de 512 Bytes.”

En resumen, hay que prestar atención. Sería bueno que los futuros discos que salgan con sectores de 4 KB informaran al SO qué estructura real tienen, y eliminar el “modo compatibilidad” de una vez por todas.

Referencias

[1] http://techreport.com/articles.x/15769

[2] http://wdc.custhelp.com/cgi-bin/wdc.cfg/php/enduser/std_adp.php?p_faqid=5655

[3] http://arstechnica.com/hardware/news/2006/09/7765.ars
[4] http://arstechnica.com/science/news/2010/05/new-hard-drive-write-method-packs-in-one-terabyte-per-inch.ars
[5] http://community.wdc.com/t5/Desktop/Problem-with-WD-Advanced-Format-drive-in-LINUX-WD15EARS/td-p/6395
[6] http://thunk.org/tytso/blog/2009/02/20/aligning-filesystems-to-an-ssds-erase-block-size/
[7] http://thread.gmane.org/gmane.linux.utilities.util-linux-ng/2926
[8] http://olo.org.pl/files/hw/postmark-automated/
[9] http://www.ibm.com/developerworks/linux/library/l-4kb-sector-disks/index.html?ca=dgr-lnxw074KB-Disksdth-LX
[10] http://wdc.custhelp.com/cgi-bin/wdc.cfg/php/enduser/std_adp.php?p_faqid=5655