

RAID

(Redundant Array of Independent Disks)

(Conjunto redundante de discos independientes)

En este documento se explicará el funcionamiento de diferentes tipos de RAID, así como la manera de llevarlos a cabo en un sistema GNU/Linux. Además se podrá ver una comparativa de todos ellos y se obtendrán algunas conclusiones apoyadas en los resultados de las pruebas realizadas.

Índice

- 1 Tipos de RAID
 - 1.1 RAID 0
 - 1.2 RAID 1
 - 1.3 RAID 5
- 2 Componentes *hardware*
- 3 Configuración de la BIOS
- 4 Configuración del *kernel*
 - 4.1 Configuración base
 - 4.2 Configuración específica para RAID
- 5 Funcionamiento de los test
- 6 Resultados de los test
- 7 Conclusiones

1. Tipos de RAID

1.1 RAID 0

El RAID 0 se utiliza para sumar la capacidad de varios discos y ganar en velocidad de lectura y escritura. Los datos se dividen en partes, y cada parte se escribe en un disco. De esa forma la escritura es aproximadamente X veces más rápida que en un disco normal, donde X es el número de discos que forman el RAID 0. Por ejemplo, un RAID 0 formado por dos discos duros forman un sólo dispositivo del doble de capacidad, donde la escritura se hace el doble de rápido. Análogamente, si los datos están divididos en partes, y cada parte está en un disco, también es el doble de rápido leer los datos, ya que la lectura se realiza en paralelo y sólo hay que leer la mitad de los datos. La desventaja de este sistema es que al tener los datos divididos en varios discos, si uno de ellos falla, los datos se pierden. Las ventajas son que no se pierde capacidad y las velocidades de lectura y escritura se multiplican. La velocidad de acceso a los datos sigue siendo la misma.

1.2 RAID 1

El RAID 1 se utiliza para que el fallo de uno de los discos no suponga la caída del sistema y la pérdida de datos. Actúa como un espejo, es decir, el sistema ve un disco sobre el cuál trabaja, pero de manera transparente hay otro disco (o varios) en los que se va copiando toda la información que se escribe en el disco principal. De esta manera, si uno de los discos que forman el RAID 1 falla, el sistema no cae y no se pierde ningún dato. Sigue funcionando usando los demás discos que forman el RAID 1. Cuando un disco ha fallado y se sustituye por otro, se pone en marcha un proceso de sincronización que consiste en hacer un espejo del disco nuevo, de forma que quede idéntico a los demás. La ventaja de este sistema es que ofrece una gran seguridad frente a la pérdida de datos y caídas del sistema. La desventaja es que se pierde por completo la capacidad extra que podrían dar los discos que se están usando como espejo. En este sistema no hay ningún aumento de velocidad de escritura ni de lectura.

1.3 RAID 5

El RAID 5 se utiliza para aprovechar las ventajas del RAID 0 y del RAID 1.

Se trata de un sistema en el que se aprovecha gran parte de la suma de las capacidades de todos los discos, se gana en velocidad de escritura y lectura, y además es tolerante a fallos. Consiste en guardar unos datos de paridad en cada disco, de forma que si uno de ellos falla, con estos datos se puede reconstruir por completo la información del disco que ha fallado. Además esto no implica ninguna caída del sistema, aunque debería reemplazarse el disco que ha fallado por uno nuevo ya que el RAID 5 sólo es capaz de soportar la pérdida de uno de los discos. Si dos discos fallaran al mismo tiempo se perdería toda la información. La ventaja del RAID 5 es que se tiene un sistema tolerante a fallos sin renunciar al aprovechamiento de la capacidad de los discos. La desventaja es que sólo soporta el fallo de un disco, por eso cuantos más discos formen el sistema RAID 5, más probabilidades hay de que dos discos fallen simultáneamente, y la pérdida de los datos sería total.

2. Componentes *hardware*

Estos han sido los componentes *hardware* utilizados:

- Placa base: GIGABYTE EP45-UD3LR
- Procesador: Intel Core 2 Quad Q8300
- Memoria RAM: 2 x 2GB Kingston 800MHz
- Discos duros: 4 x Seagate Barracuda 7200.12 250GB
- Tarjeta gráfica: ASUS EN210
- Lector DVD: LG GH22NS40
- Alimentación: Powertronics 500W
- Cables: 5 x Cables SATA



3. Configuración de la BIOS

Para poder usar la función de *hotswapping* (conexión en caliente) de la que dispone la placa base es necesario que el controlador SATA sea AHCI.

En el menú de configuración de la BIOS hay que entrar en "Integrated Peripherals".

De las opciones que aparecen sólo interesan dos. La primera es "SATA RAID/AHCI Mode". Hay que establecerla en AHCI. Se podría establecer a RAID, pero eso sería para hacer RAID por BIOS, que no es el caso.

La segunda opción que interesa es "SATA AHCI Mode" y hay que establecerla en "AHCI".

Con esto ya se ha configurado la BIOS para usar el controlador AHCI para SATA. Ahora se puede aprovechar la función *hot plug* para conectar discos duros SATA en caliente.

4. Configuración del kernel

4.1 Configuración base

Estás son las opciones mínimas que se deben seleccionar en el menú de configuración del kernel para que el sistema pueda arrancar. Tener el controlador para las unidades de disco y soporte para el sistema de ficheros con el que se han formateado (en este caso ext4).

Controlador SATA:

```
-> Device Drivers
    -> <*> Serial ATA (prod) and Parallel ATA (experimental) drivers
        -> <*> AHCI SATA support
```

Sistema de ficheros ext4:

```
-> File systems
    -> <*> The Extended 4 (ext4) filesystem
```

Para poder instalar el sistema operativo (en este caso Gentoo) y las herramientas necesarias para hacer las pruebas, se necesitará conexión a Internet, así que hay que activar también el soporte para la tarjeta de red.

Tarjeta de red:

```
-> Device Drivers
    -> [*] Network device support
        -> [*] Ethernet (1000 Mbit)
            -> <*> Realtek 8169 gigabit ethernet support
```

4.2 Configuración específica para RAID

RAID 0:

```
-> Device Drivers
  -> [*] Multiple devices driver support (RAID and LVM)
    -> <*> RAID support
      -> <*> RAID-0 (striping) mode
```

RAID 1:

```
-> Device Drivers
  -> [*] Multiple devices driver support (RAID and LVM)
    -> <*> RAID support
      -> <*> RAID-1 (mirroring) mode
```

RAID 5:

```
-> Device Drivers
  -> [*] Multiple devices driver support (RAID and LVM)
    -> <*> RAID support
      -> <*> RAID-4/RAID-5/RAID-6 mode
```

5. Funcionamiento de los test

Las distintas pruebas que se van a realizar son las siguientes:

- Tiempo de formateado
- Capacidad del sistema RAID
- Capacidad útil del sistema RAID
- Tiempo de copia de ficheros
 - 100MB
 - 500MB
 - 1GB
 - 5GB
 - 10GB
 - 50GB
 - 100GB
- Tiempo de extracción
- Tiempo de compilación
- Tiempo de inserción en una tabla MySQL
- Tiempo de arranque del sistema operativo
- Tiempo de búsqueda de ficheros

Para medir los tiempos se ha usado la manera más sencilla de todas. Mirar la hora del sistema antes y después de la ejecución de la tarea. Después se restan y el resultado es el tiempo que ha tardado en ejecutarse la tarea.

Ejemplo: `date && tarea && date`

Para formatear se ha usado el sistema de ficheros ext4. La orden exacta es:

`mkfs.ext4 /dev/partición`

La capacidad del sistema raid se mide con la utilidad df. De la salida que produce hay que fijarse en la columna "Size".

La capacidad real está especificada en la columna "Avail".

Para crear los ficheros que se utilizarán al copiar se ha usado el dispositivo especial /dev/urandom. Se ha volcado la salida del dispositivo a un fichero inicial hasta tener un tamaño mayor a 100MB. Después con la herramienta split se ha troceado el fichero en partes de 100MB. Tomando el fichero de 100MB como base se han creado el resto de ficheros.

Creación del fichero base de 100MB:

```
cat /dev/urandom > fichero_temporal  
split --bytes=100M fichero_temporal  
mv xaa 100MB
```

Creación del fichero de 500MB:

```
cat 100MB 100MB 100MB 100MB 100MB > 500MB
```

Creación del fichero de 1GB:

```
cat 500MB 500MB > 1GB
```

Creación del fichero de 5GB:

```
cat 1GB 1GB 1GB 1GB 1GB > 5GB
```

Creación del fichero de 10GB:

```
cat 5GB 5GB > 10GB
```

Creación del fichero de 50GB:

```
cat 10GB 10GB 10GB 10GB 10GB > 50GB
```

Creación del fichero de 100GB:

```
cat 50GB 50GB > 100GB
```

Para la extracción se han usado dos ficheros básicos en la instalación de Gentoo. El stage3 y el portage-latest. La extracción se realiza tal y como indica el Gentoo Handbook.

Para medir el tiempo de compilación se ha usado la creación de la imagen del kernel de Linux. En concreto se ha medido la orden:

```
make && make modules_install
```

que incluye, a parte de la compilación del kernel, la instalación de los módulos.

Para la prueba de inserción en una tabla MySQL se ha usado un script que ejecuta un millón de sentencias INSERT.

La medición del tiempo de arranque del sistema operativo se cuenta desde que se pulsa INTRO en la opción del grub, hasta que el sistema se queda esperando un login.

La búsqueda de ficheros se ha realizado usando la herramienta find, para buscar todos los ficheros dentro del directorio /usr/. Después de una instalación típica de Gentoo se encuentran aproximadamente 176.000 ficheros.

Se han realizado un total de cinco montajes.

El primer montaje es el único que no es RAID. En él se ha usado un único disco duro para usar todas sus mediciones como base para poder compararlas con las de los sistemas RAID.

El segundo montaje es un RAID 0 utilizando dos discos duros. Hay que hacer notar que no se puede arrancar el sistema desde un RAID 0 o 5. Por eso hay que hacerlo desde otro disco duro, o como ha sido el caso, desde una partición que se usa sólo para /boot/.

Para crear el RAID se ha utilizado la siguiente orden:

```
mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sda2 /dev/sdb2
```

/dev/sda1 es la partición utilizada para /boot/. El segundo disco ha sido particionado exactamente igual que el primero, por lo que también tiene una partición /dev/sdb1 , aunque en este caso no se usa para nada. Tan sólo se pierden 10MB por esto.

La configuración de grub no es trivial, puesto que como partición / (root) se tiene a un dispositivo RAID que se compone de varios discos, se debe hacer de este modo:

```
kernel /boot/kernel-2.6.32 root=/dev/md0 md=0,/dev/sda2,/dev/sdb2
```

Esta información se puede obtener directamente de la documentación del kernel, aquí: [/usr/src/linux/Documentation/md.txt](#)

El tercer montaje es un RAID 1 utilizando dos discos duros. Uno hace de espejo del otro. A parte de las mediciones de tiempo también se ha hecho la prueba de desconectar en caliente uno de los discos y comprobar que el sistema seguía funcionando. Acto seguido se ha conectado un disco duro nuevo y se ha observado el proceso de sincronización.

Para crear el RAID primero hay que deshacer el RAID anterior así:

```
mdadm --stop /dev/md0
```

Lo cual mostrará un mensaje de: stopped /dev/md0

La creación del RAID1 se hace así:

```
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda1 /dev/sdb1
```

En este tipo de RAID sí que se permite arrancar el sistema desde él, con lo que no hace falta tener una partición sólo para /boot/. Nótese que se han usado las primeras (y únicas) particiones de los discos.

La configuración del grub queda así:

```
kernel /boot/kernel-2.6.32 root=/dev/md0 md=0,/dev/sda1,/dev/sdb1
```

Mientras el sistema estaba en funcionamiento se procedió a desconectar en caliente uno de los discos. Al hacerlo, el sistema seguía funcionando sin ningún problema. Se pudo observar que el estado del dispositivo RAID cambió, y uno de los discos había desaparecido:

```
cat /proc/mdstat
```

```
Personalities : [raid1]
```

```
md0 : active raid1 sdb1[1] sda1[2](F)
244195904 blocks [2/1] [_U]
```

Al conectar un nuevo disco para sustituir al que había "fallado" observamos el proceso de sincronización:

```
cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdc1[2] sdb1[1] sda1[3](F)
244195904 blocks [2/1] [_U]
[>.....] recovery = 1.1% (2925888/244195904) finish=47.2min
speed=85167K/sec
```

Al finalizar la sincronización el dispositivo RAID volvía a estar en perfectas condiciones:

```
cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdc1[0] sdb1[1] sda1[2](F)
244195904 blocks [2/2] [UU]
```

Nótese que el nuevo disco duro ha sido nombrado por el sistema como /dev/sdc y no /dev/sda.

Tras reiniciar el sistema, el nuevo disco duro vuelve a ser reconocido como /dev/sda:

```
cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdb1[1] sda1[0]
244195904 blocks [2/2] [UU]
```

El cuarto montaje es un RAID 0 utilizando cuatro discos duros. El proceso es el mismo que el del segundo montaje, sólo que ahora se observa todavía más mejora en los tiempos de lectura y escritura.

Para poder hacerlo primero necesitamos deshacer el anterior RAID:

```
mdadm --stop /dev/md0
```

Y ahora creamos el RAID:

```
mdadm --create /dev/md0 --level=0 --raid-devices=4 /dev/sda2 /dev/sdb2  
/dev/sdc2 /dev/sdd2
```

El grub.conf contiene ahora esto:

```
kernel /boot/kernel-2.6.32 root=/dev/md0  
md=0,/dev/sda2,/dev/sdb2,/dev/sdc2,/dev/sdd2
```

El quinto montaje es un RAID 5. Aquí se observa cómo se aprovecha gran parte de la suma de las capacidades de los discos y el aumento de velocidad de lectura y escritura, teniendo además un sistema con tolerancia a fallos.

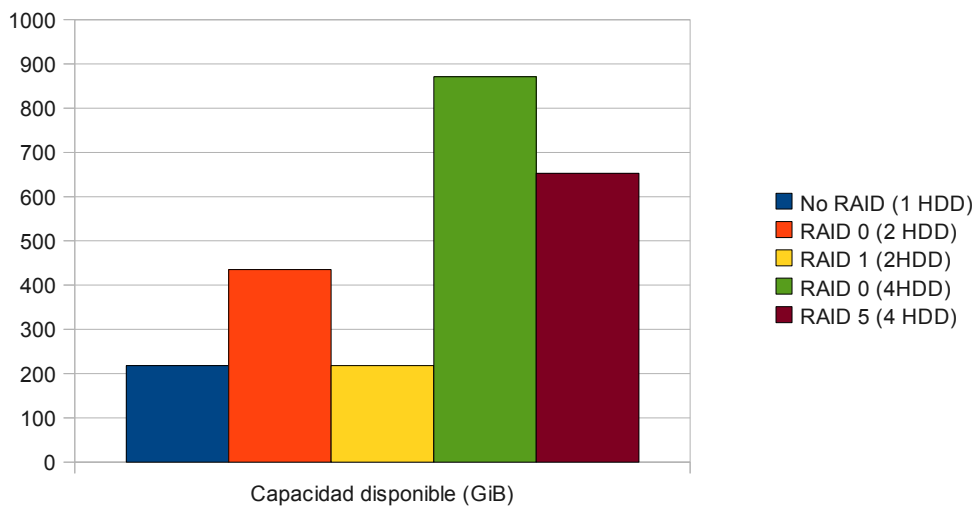
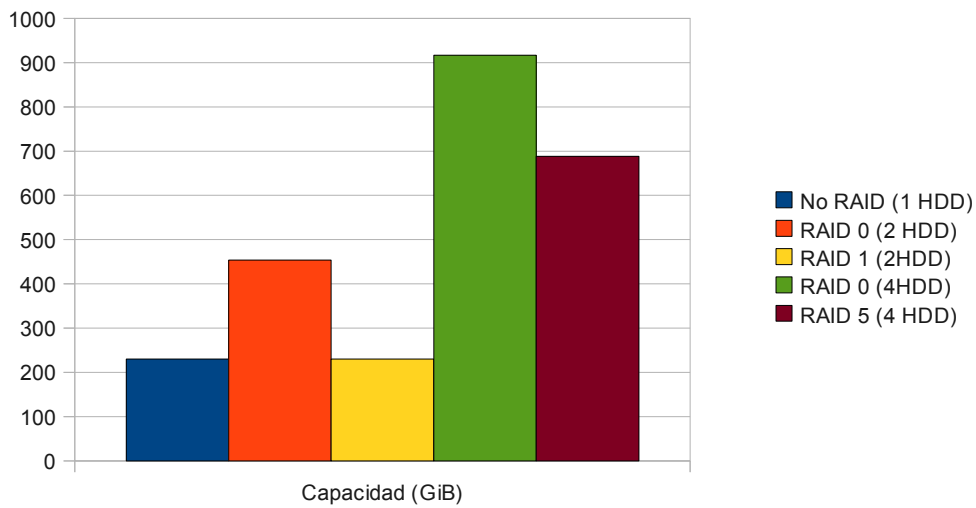
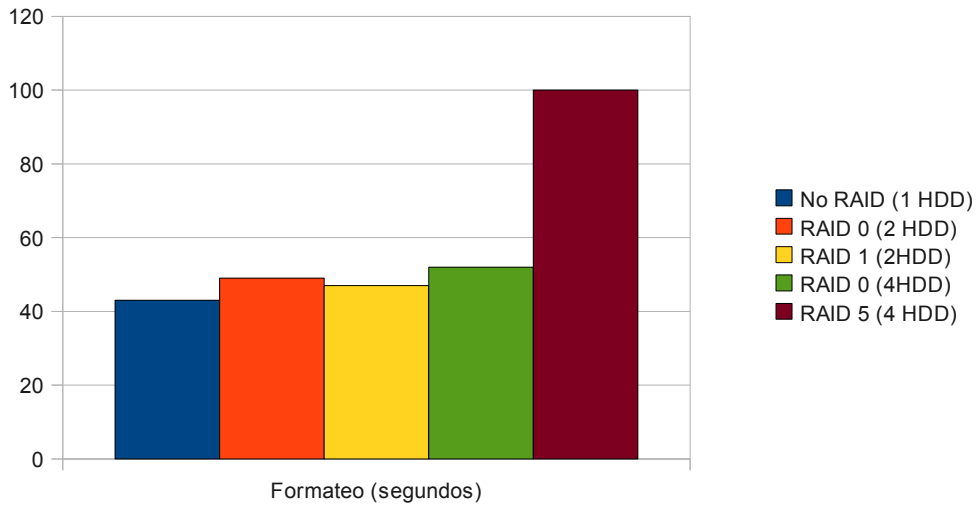
Primero deshacemos el RAID anterior:

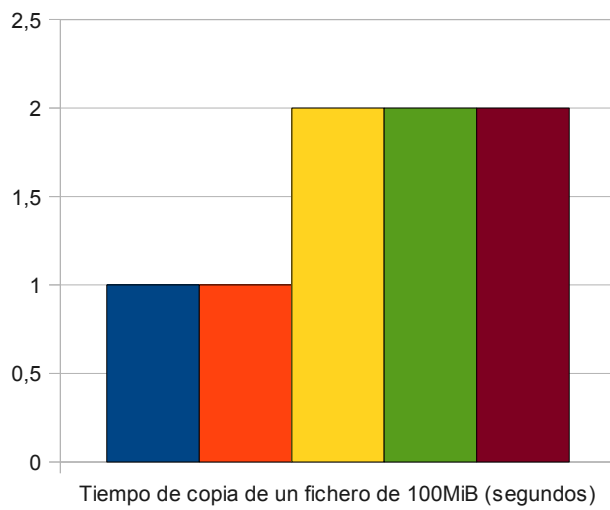
```
mdadm --stop /dev/md0
```

Y ahora creamos el RAID 5:

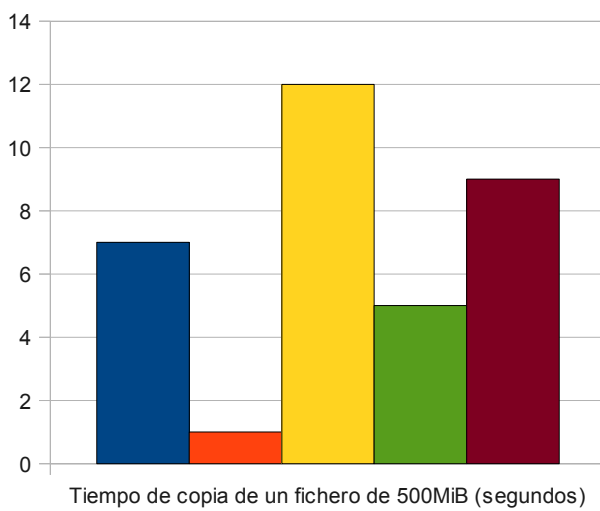
```
mdadm --create /dev/md0 --level=5 --raid-devices=4 /dev/sda2 /dev/sdb2  
/dev/sdc2 /dev/sdd2
```

6. Resultados de los test

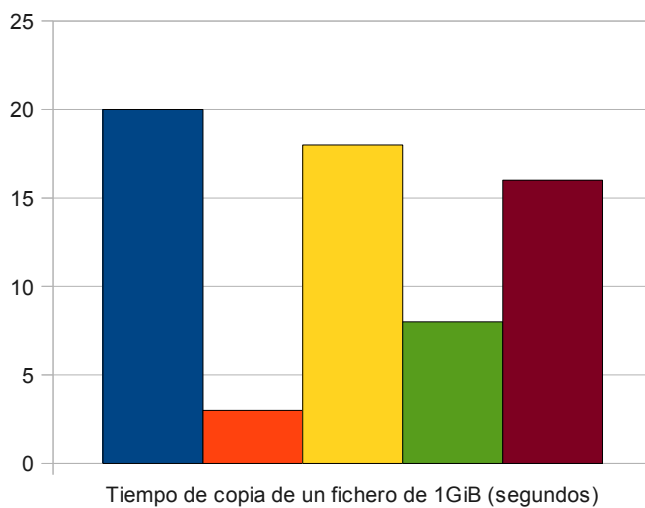




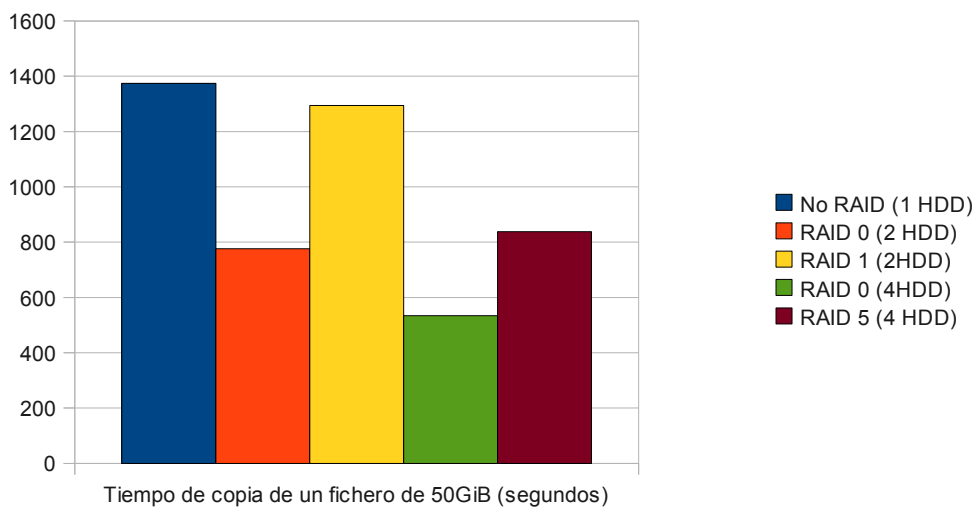
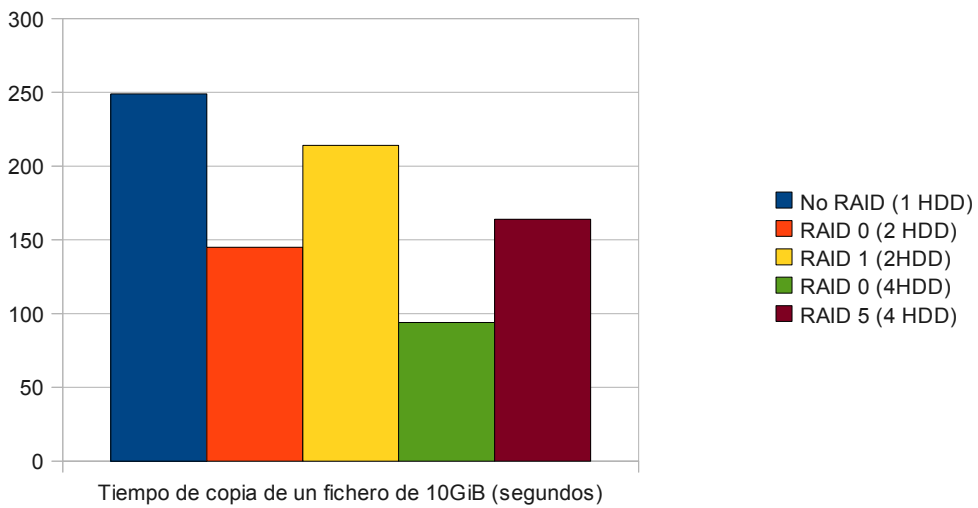
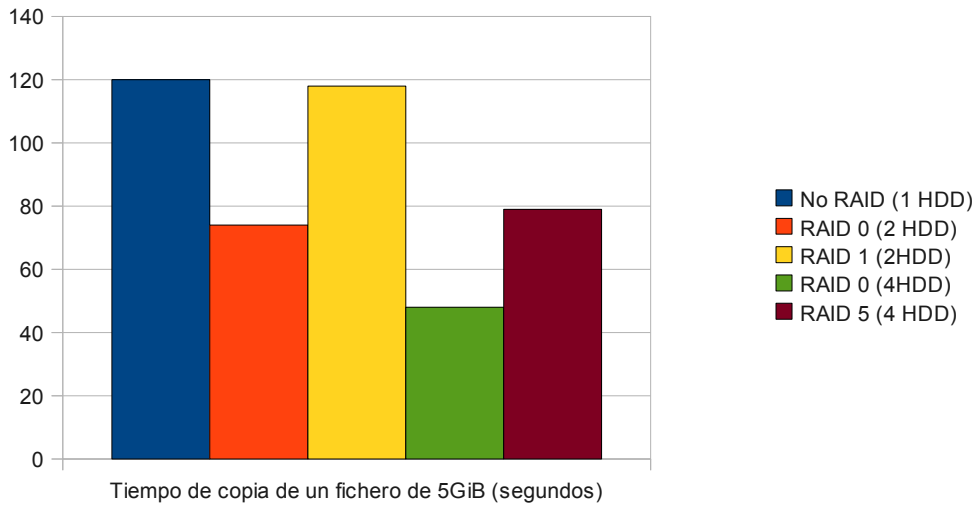
- No RAID (1 HDD)
- RAID 0 (2 HDD)
- RAID 1 (2HDD)
- RAID 0 (4HDD)
- RAID 5 (4 HDD)

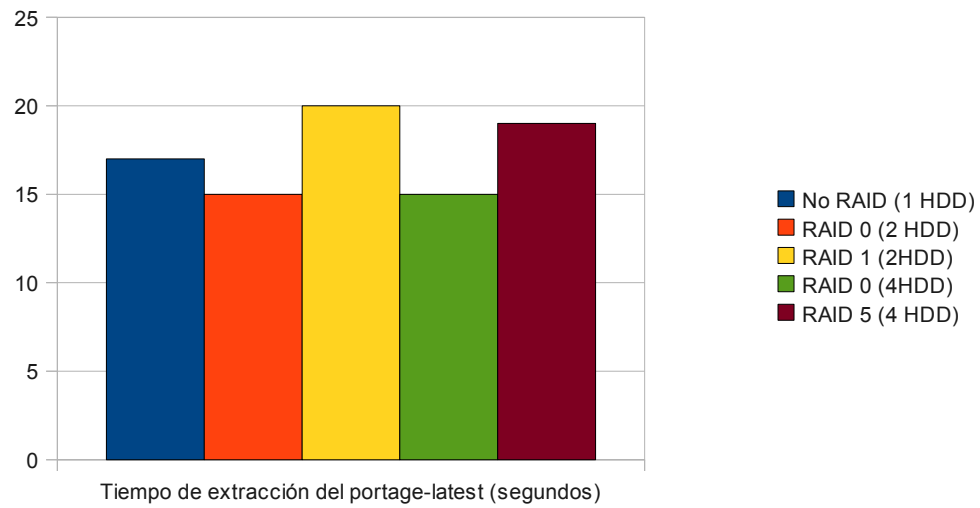
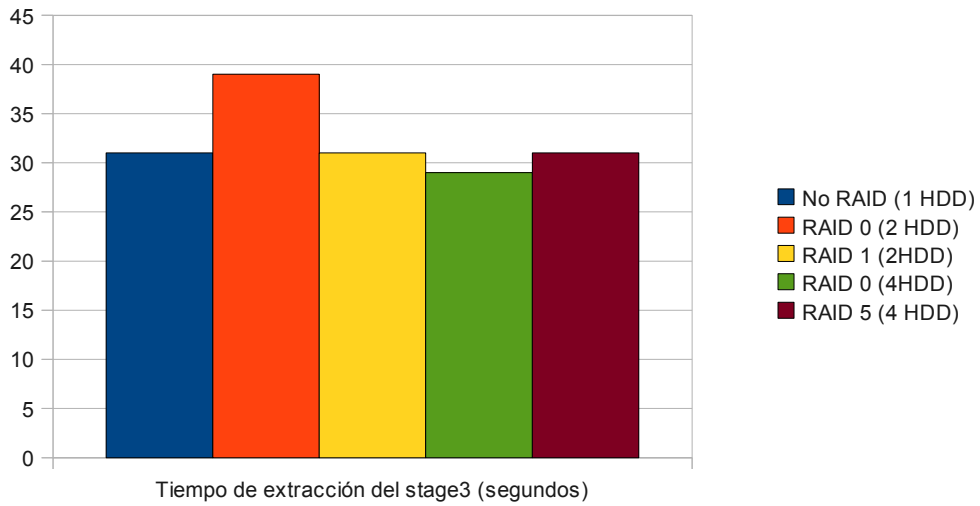
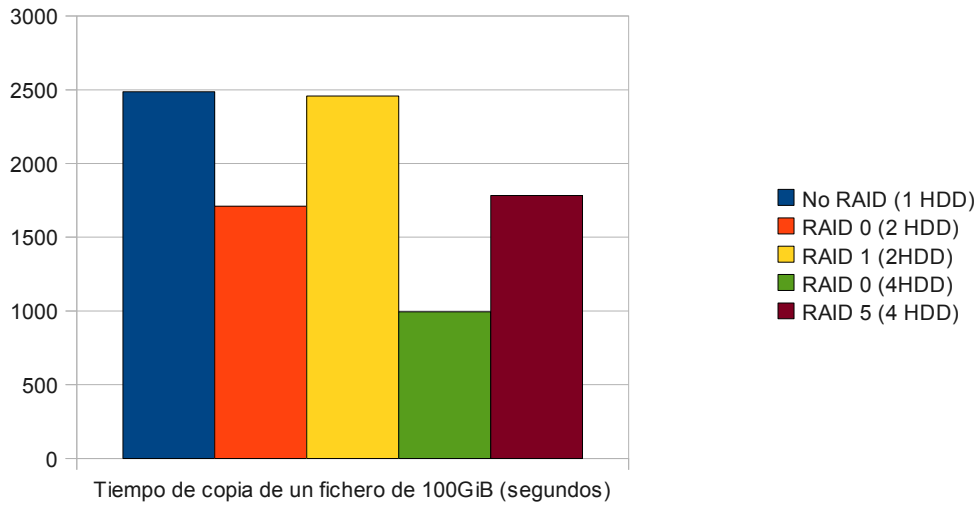


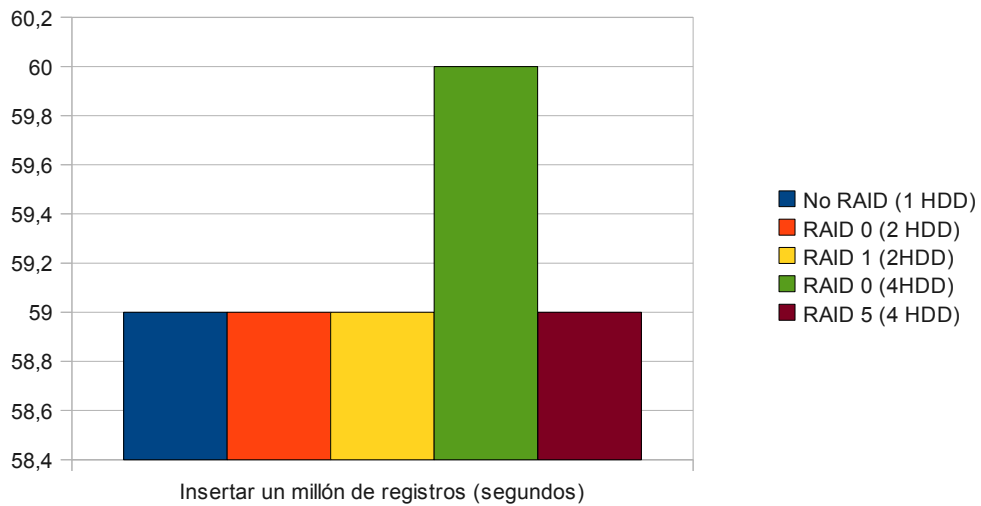
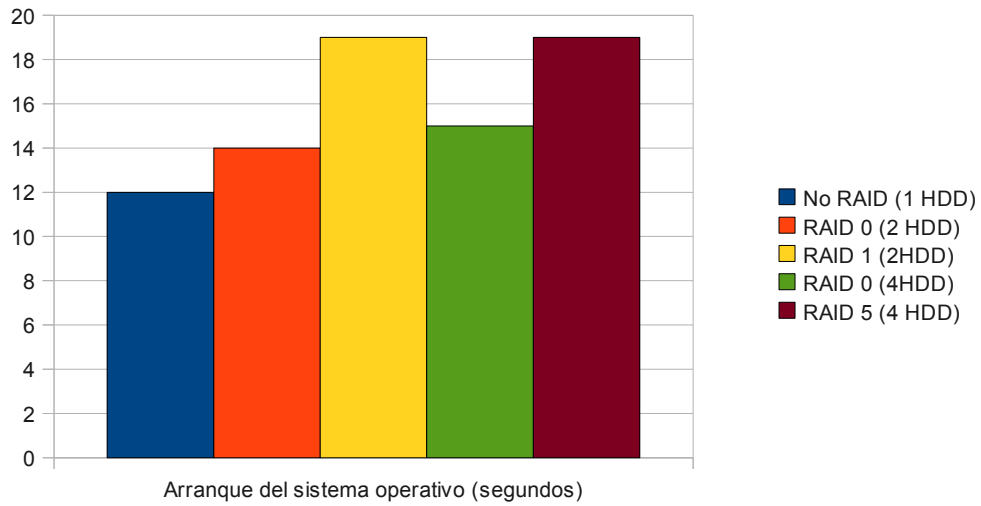
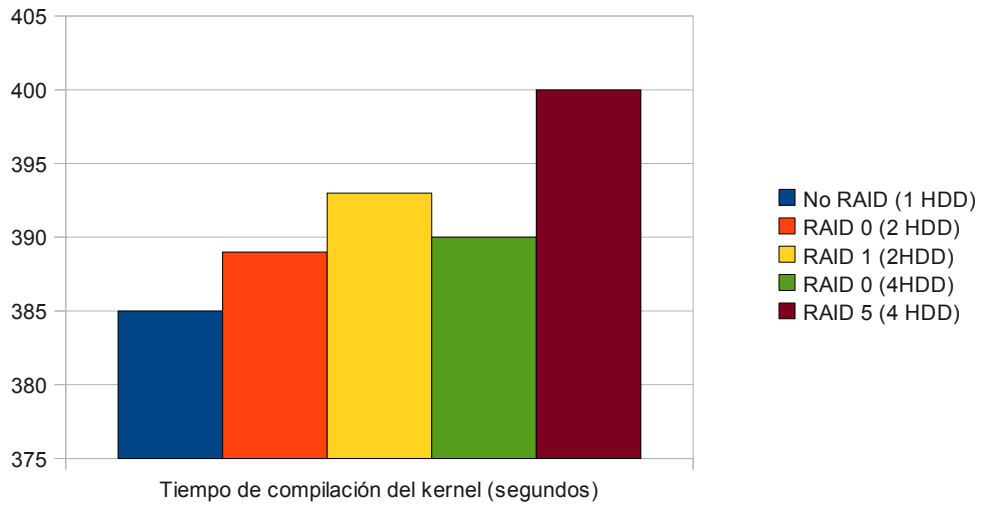
- No RAID (1 HDD)
- RAID 0 (2 HDD)
- RAID 1 (2HDD)
- RAID 0 (4HDD)
- RAID 5 (4 HDD)

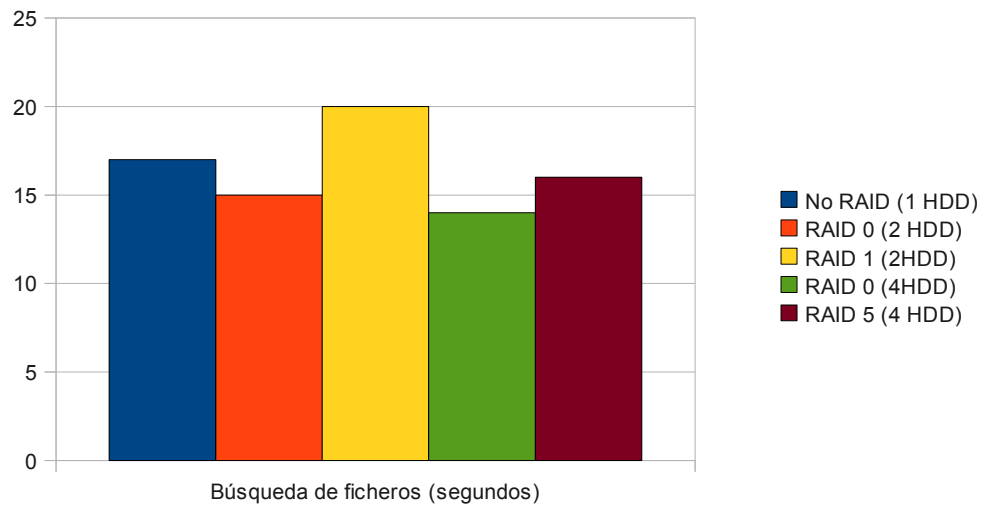


- No RAID (1 HDD)
- RAID 0 (2 HDD)
- RAID 1 (2HDD)
- RAID 0 (4HDD)
- RAID 5 (4 HDD)









7. Conclusiones

Como se ha podido ver, todas aquellas tareas que dependen de la velocidad del procesador o del tiempo de acceso a los datos (arranque del sistema, compilación, búsqueda de ficheros, inserción de registros en bases de datos, extracción...) no experimentan ninguna mejora en un sistema RAID.

Por el contrario, aquellas tareas que dependen de la velocidad de lectura o escritura sí que experimentan una gran mejora de velocidad.

A parte de la velocidad, el RAID también ofrece la ventaja de unir varios discos y sumar sus capacidades, y tener tolerancia a fallos.

El RAID 0 es el que mejor aprovechamiento de la capacidad de los discos tiene y el más rápido.

El RAID 1 no supone ningún aumento de velocidad pero proporciona un mecanismo muy potente de tolerancia a fallos. Se pueden tener varios espejos de un mismo disco, con lo que la probabilidad de perder datos disminuye cuantos más espejos haya.

El RAID 5 tiene un aprovechamiento de las capacidades de los discos bastante buena, teniendo en cuenta que tiene que destinar parte del espacio a datos de paridad. Proporciona una mejora de velocidad, aunque no tanta como el RAID 0. También dispone de tolerancia a fallos, aunque sólo admite el fallo simultáneo de un sólo disco.