

# Seguridad y disponibilidad: RAID por software y backup diferencial con DAR usando GNU/Linux

Mauro Silvosa Rivera  
mauro@meleeisland.net

Grupo de Usuarios y Programadores de Linux (GPUL)

2006-07-13 (Rev. 1)

# Contenido

## 1 RAID por software usando GNU/Linux

- Introducción
- Instalación de un servidor usando RAID-1 por software
- Ejemplo de uso de RAID-5 por software

## 2 Backup diferencial usando DAR

- Introducción
- Backup diferencial usando DAR
- Recuperación de datos

## 3 Referencias

# Contenido

- 1 ¿Qué es RAID?
- 2 Niveles 1 y 5 de RAID
- 3 Discos de reserva (spare disks)
- 4 Coste de las diferentes alternativas consideradas

## ¿Qué es RAID?

- RAID: Redundant Array of Inexpensive Disks, posteriormente conocido como Redundant Array of Independent Disks

<http://en.wikipedia.org/wiki/RAID>

RAID es un sistema que usa múltiples discos duros para compartir o replicar la información a lo largo de esos discos.

Dependiendo de la versión de RAID elegida, RAID nos permite aumentar la integridad de los datos, la tolerancia a fallos, el ancho de banda de EE/SS o la capacidad de almacenamiento respecto a cada disco visto de forma individual.

En su implementación original, la ventaja principal era la capacidad de combinar varios discos de bajo coste y de tecnologías más antiguas en un array que ofrecía rendimiento y capacidad de almacenamiento comparables a los discos más caros de tecnología más moderna.

## Niveles 1 y 5 de RAID

- RAID-1 (mirror)
  - ▶ Copia exacta de un conjunto de datos en dos o más discos
  - ▶ Prevalece el rdto. de lectura frente a la capacidad de almacenamiento
- RAID-5 (stripping)
  - ▶ Trozea datos a nivel de bloque y calcula bloques de paridad que se distribuyen por todos los discos del array
  - ▶ Popular por su bajo coste de redundancia por Mbyte
  - ▶ Mejor rendimiento cuando se implementa con hardware

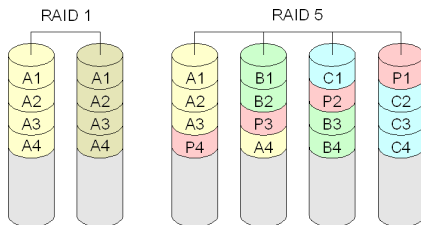


Figura: Organización de discos en modos 1 y 5 de RAID

## Spare disks (discos de reserva)

- Forman parte del array pero no están en uso hasta que uno de los que sí lo están falla
- En caso de fallo ocupan automáticamente el lugar del disco defectuoso y comienzan a replicarse para sincronizarse con los datos del array
- RAID-1 y RAID-5 admiten discos de spare

## Coste de las diferentes alternativas consideradas

- Consideraremos la implementación por software de RAID-1 y RAID-5 con y sin discos spare

Tipo	Controladoras	Discos	Total	EUR/GByte
RAID-1	1x44 EUR	2x59 EUR	162 EUR	1,35 EUR/GByte
RAID-1+S	2x44 EUR	3x59 EUR	265 EUR	2,2 EUR/GByte
RAID-5	2x44 EUR	3x59 EUR	265 EUR	1,104 EUR/GByte
RAID-5+S	2x44 EUR	4x59 EUR	324 EUR	1,35 EUR/GByte

Cuadro: Coste de las diferentes alternativas

# Contenido

- 1 Instalación del sistema base
- 2 Recompilación del kernel
- 3 Creación de los arrays RAID-1 (todavía incompletos)
- 4 Preparación del sistema para arrancar desde los arrays RAID-1
- 5 Completar los arrays RAID-1 con el resto de discos
- 6 Últimos pasos y puesta en marcha
- 7 Probando el sistema



## Instalación del sistema base

- Debian network installation disk (<http://www.debian.org/CD/netinst>)
- Particionar el disco duro

Partición	Punto de montaje	Tamaño	Tipo	MD
/dev/sda1	/	2 GBytes	Primary - Linux	md0
/dev/sda2	swap	512 MBytes	Primary - swap	md1
/dev/sda3	/boot	30 MBytes	Primary - Linux	md2
/dev/sda5	/home	2 GBytes	Logical - Linux	md3
/dev/sda6	/tmp	1 GBytes	Logical - Linux	md4
/dev/sda7	/dat	2 GBytes	Logical - Linux	md5

**Cuadro:** Tabla de particiones definidas en el primer disco.

- Intalar LILO como bootloader en MBR de /dev/sda (GRUB debería funcionar, pero no lo probé nunca)

## Recompilación del kernel

- El servidor montará todos los datos (incluidos /boot y /) en arrays RAID.
- Por tanto, el kernel debe tener integrado el soporte para RAID. No sirve como módulo.
- Importante no olvidarse del HW instalado: tarjetas SATA, SCSI, vídeo, red, ...

### Compilación del kernel

```
$ cd /usr/src/linux-source-2.6.15
$ make menuconfig
$ make-kpkg kernel-image
$ cd ..
$ dpkg -i kernel-image.deb
```

## Creación de los arrays RAID-1 (todavía incompletos)

- Copia de la estructura de particiones del primer disco a los demás

```
$ sfdisk -d /dev/sda | sfdisk /dev/sdb
```

```
$ sfdisk -d /dev/sda | sfdisk /dev/sdc
```

- Linux RAID autodetect: marcamos todo con tipo FD menos la partición Extended

```
$ fdisk /dev/sdb
```

```
$ fdisk /dev/sdc
```

- Creación de los arrays RAID-1

```
$ apt-get install mdadm
```

```
$ mdadm --create /dev/mdX --verbose
```

```
    --level=1 --raid-devices=2
```

```
    --spare-devices=1 missing /dev/sdbY missing
```

```
$ cat /proc/mdstat
```

## Creación de los arrays RAID-1... (2)

- Creación de mdadm.conf

```
$ mdadm --detail --scan > /etc/mdadm/mdadm.conf
```

- Creación de los sistemas de ficheros en los arrays

```
$ mkfs.ext3 /dev/mdX
```

```
$ mkfs.ext2 /dev/mdX
```

```
$ mkswap /dev/md1
```

- Reboot

```
$ reboot
```

### Reinicio no necesario...

Este reinicio no es necesario, pero es interesante hacerlo para confirmar que todo va bien. Al terminar de reiniciar comprobamos el estado de los arrays con `$ cat /proc/mdstat` y si todo va bien seguimos.

# Preparación del sistema para arrancar desde los arrays

## Importante...

Es muy importante respetar el orden de estos pasos para que en el último paso (replicación) todos los discos tengan los mismos datos.

- 1 Edición de `/etc/lilo.conf`  
`root = /dev/md0`
- 2 Edición de `/etc/fstab` para que el sistema monte los arrays RAID en lugar de los dispositivos físicos.
- 3 Replicación de los discos físicos en los arrays RAID  

```
$ echo "Ejemplo para la partición raíz"  
$ cd /  
$ mount /dev/md0 /mnt  
$ find . -xdev | cpio -pm /mnt
```
- 4 **Reiniciamos** el servidor para que se monte sobre los arrays.

## Completar los arrays RAID-1 con el resto de discos

- 1 Linux RAID autodetect para el primer disco.

```
$ fdisk /dev/sda
```

- 2 Añadir el primer disco al array

```
$ mdadm --manage /dev/mdX --add /dev/sdaY
```

- 3 Añadir el disco de spare al array

```
$ mdadm --manage /dev/mdX --add /dev/sdcY
```

### Nota...

Los dos últimos pasos se podrían haber integrado en una sólo línea de comando:

```
$ mdadm --manage /dev/mdX --add /dev/sdaY /dev/sdcY
```

# Últimos pasos y puesta en marcha

- 1 Duplicación del MBR:
  - ▶ Editamos `/etc/lilo.conf` para que refleje `boot=/dev/md2` y `raid-extra-boot=mbr`
  - ▶ Ejecutamos LILO para aplicar los cambios: `lilo -v`
- 2 Reinicio del sistema

## Reinicio no necesario...

Este reinicio tampoco es necesario, pero de nuevo vuelve a ser bueno hacerlo para ver que todo va bien y de comprobar que el sistema arranca. Si esta prueba funciona estaríamos en condiciones de hacer otras pruebas más duras, como las que se describirán más adelante.

# Probando el sistema

- Primera prueba

- ▶ Marcar un disco (no spare) como `faulty` (forzar el fallo). Esto debería hacer que entrase automáticamente el disco `spare`.

```
$ mdadm --manage /dev/md4 --set-faulty /dev/sdb6
```

- ▶ Quitar el disco arcado como `faulty`

```
$ mdadm --manage /dev/md4 --remove /dev/sdb6
```

- ▶ Marcar de nuevo un disco como `faulty` (forzar el fallo). Esto debería hacer que el array quedase sólo con un disco útil.

```
$ mdadm --manage /dev/md4 --set-faulty /dev/sdc6
```

- ▶ Volver a quitar el disco que ha fallado

```
$ mdadm --manage /dev/md4 --remove /dev/sdc6
```

- ▶ Añadir de nuevo al volumen dos discos para dejarlo en un estado de más garantías frente a fallos.

```
$ mdadm --manage /dev/md4 --re-add /dev/sdb6  
--add /dev/sdc6
```



# Probando el sistema

- Segunda prueba
  - ▶ Shutdown
  - ▶ Retirar el primer disco
  - ▶ Arrancar de nuevo

Si todo está bien hecho el sistema debería arrancar sin problemas usando el sector de arranque de otro de los discos.

- Tercera prueba
    - ▶ Shutdown
    - ▶ Insertar de nuevo el primer disco
    - ▶ Arrancar de nuevo
    - ▶ Añadir de nuevo el disco de spare y resincronizar el primer disco
- ```
$ mdadm --manage /dev/mdX --re-add /dev/sdaY  
--add /dev/sdcY
```

## Creación de un array RAID-5

- 1 Crear las particiones del disco de spare

```
$ sfdisk -d /dev/sda | sfdisk /dev/sdd
```

- 2 Eliminar uno de los arrays RAID-1. Por ejemplo, el de /dat

```
$ umount /dat
```

```
$ mdadm --manage /dev/md5 --stop
```

- 3 Creación de un nuevo array RAID-5

```
$ mdadm --create /dev/md5 --verbose --level=5 --raid-devices=3  
--spare-devices=1 /dev/sda7 /dev/sdb7 /dev/sdc7  
/dev/sdd7
```

- 4 Crear el sistema de ficheros en el array

```
$ mkreiserfs /dev/md5
```

- 5 Montarlo en el sistema

```
$ mount /dat
```

# Contenido

- 1 ¿Qué es DAR?
- 2 Uso básico de DAR

## ¿Qué es DAR?

- Es un programa de línea de comando que permite realizar backup de directorios y/o ficheros
- Diseñado, escrito y mantenido por Denis Corbin
- Distribuido bajo licencia GPL (GNU General Public License)
- Proporciona API's para que terceros puedan desarrollar software compatible con DAR (como Kdar)
- Instalación muy sencilla: `$ apt-get install dar dar-docs`

## Uso básico de DAR

- Generación de un backup de una serie concreta de ficheros y directorios

```
$ dar -c nombre_fichero_backup -g dir_1 ... -g dir_n  
      -g file_1 ... -g file_n
```

- Listado de los ficheros almacenados en un backup (consulta del catálogo)

```
$ dar -l nombre_fichero_backup
```

- Recuperación de datos

```
$ cd ~  
$ mkdir tmp  
$ cd tmp  
$ dar -x ../nombre_fichero_backup
```

# Contenido

- 1 ¿Qué es un backup diferencial?
- 2 Planificación del backup
- 3 Ejecución del backup
- 4 Recuperación de datos

## ¿Qué es un backup diferencial?

- Un backup diferencial (o incremental) es aquel en el que sólo se almacenan los ficheros que hayan sido modificados desde el último backup
- Ahorra espacio en disco porque sólo copia las diferencias
- Rapidez. El backup se hace en poco tiempo porque generalmente las diferencias no son extremadamente grandes
- DAR es capaz de realizar backups diferenciales (opción -A)

# Planificación del backup

Para establecer una política de backups debemos preguntarnos:

- ¿Qué datos queremos copiar?
- ¿Con qué frecuencia y a qué horas?
- ¿Cuánto tiempo tengo que conservar las copias?
- ¿Tengo espacio para almacenar esas copias?



## Ejecución del backup

Para evitar líneas de comando kilométricas, es mejor utilizar scripts que nos permitan:

- crear un backup de referencia o diferencial según corresponda
- con los parámetros de configuración de DAR oportunos
- copiar en él los ficheros y/o directorios que nos interesan
- chequear que los backups se han llevado a cabo sin problemas

### Scripts de ejemplo...

Para este ejemplo usaremos un par de scripts muy simples: `dar_master.sh` y `dar_diff.sh`, que se encargarán respectivamente de realizar la copia maestra y las de diferencias cuando correspondan. Este sistema tiene un inconveniente: si ocurre algo en el servidor que impida que se ejecute `dar_master.sh` el día en que está planeado, habremos perdido la capacidad de hacer copias diferenciales ya que `dar_diff.sh` no tendrá una copia de seguridad completa que pueda usar de referencia.

## Ejecución del backup (2)

### ● dar\_master.sh

```
#!/bin/sh

# SOURCE = Esto es el directorio del que queremos hacer backup.
SOURCE=/DATOS/SHARES/dsd_mauro
# DIR = Es el directorio en donde guardamos los ficheros del backup.
DIR=/DATOS/BACKUP
SLICE_NAME=${DIR}/backup_$(/bin/date -I)_master
SLICE_SIZE=600M
#
mv $DIR/ANT $DIR/ANT.TMP
mkdir $DIR/ANT
mv $DIR/*.dar $DIR/ANT

# Añadir "-y" para que comprima.
/usr/bin/dar -m 256 -s $SLICE_SIZE -D -R $SOURCE -c $SLICE_NAME -Z "*.gz" \
            -Z "*.bz2" -Z "*.zip" -Z "*.png" -Z "*.jpg" -Z "*.rar" -Z "*.tgz"

/usr/bin/dar -t $SLICE_NAME
/usr/bin/find $DIR -type f -exec chmod 440 {\} \;
/usr/bin/find $DIR -type d -exec chmod 550 {\} \;

rm -rf $DIR/ANT.TMP
```

## Ejecución del backup (3)

### ● dar\_diff.sh

```
#!/bin/bash

# SOURCE = Esto es el directorio del que queremos hacer backup.
SOURCE=/DATOS/SHARES/dsd_mauro
# DIR = Es el directorio en donde guardamos los ficheros del backup.
DIR=/DATOS/BACKUP

PREV='/bin/ls $DIR/*.dar|/usr/bin/tail -n 1|/usr/bin/awk -F ' ' '{print $1;}''

SLICE_NAME=${DIR}/backup_`/bin/date -I`_diff
SLICE_SIZE=600M

# Añadir "-y" para que comprima.
/usr/bin/dar -m 256 -s $SLICE_SIZE -D -R $SOURCE -c $SLICE_NAME -Z "*.gz" \
-Z "*.bz2" -Z "*.zip" -Z "*.png" -Z "*.jpg" -Z "*.rar" -Z "*.tgz" \
-A $PREV

/usr/bin/dar -t $SLICE_NAME
/usr/bin/find $DIR -type f -exec chmod 440 {\} \;
/usr/bin/find $DIR -type d -exec chmod 550 {\} \;
```

## Ejecución del backup (4)

- Automatización del proceso con cron (a través de crontab)

```
SHELL=/bin/bash
```

```
PATH=/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games
```

```
MAILTO="mauro"
```

```
30 4 2-31 * * /root/scripts/backup/dar_diff.sh
```

```
30 4 1 * * /root/scripts/backup/dar_master.sh
```

# Recuperación de datos

- Recuperación completa de todo el backup: `recover.sh`

```
#!/bin/sh

if [ -n "$3" ]; then
    INPUT="$1_master"
    FS_ROOT="$2"

    dar -x "$INPUT" -w -R "$FS_ROOT"
    for file in ${INPUT:0:8}*_diff*; do
        dar -x "${file:0:15}" -w -R "$FS_ROOT"
    done
    echo "All done."
else
    echo "Not enough parameters."

    Usage: script master_full_backup destination_dir

    Where master_full_backup is a date in the format 'YYYY-MM-DD', and
    destination_dir is the place where you want to put the restored data."
fi
```

## Recuperación de datos (2)

- Recuperación de ficheros y/o directorios concretos dentro del backup
  - 1 Búsqueda en los diferentes ficheros del backup en el se tiene una copia de ese fichero con la fecha que más se acerque a la fecha en la que queremos recuperar el fichero:

```
$ dar -l nombre_copia_seguridad
```
  - 2 Extracción del fichero(s) y/o directorio(s) una vez localizado(s):

```
$ dar -x nombre_copia_seguridad -g nombre_fichero_a_recuperar
```

# Referencias

- `man mdadm`
- `man dar`
- <http://neil.brown.name/blog/mdadm>
- <http://cgi.cse.unsw.edu.au/~neilb/SoftRaid>
- <http://www.cse.unsw.edu.au/~neilb/source/mdadm/>
- <http://dar.sourceforge.net/>
- <http://www.tldp.org/HOWTO/Software-RAID-HOWTO.html>
- [http://wiki.clug.org.za/wiki/RAID-1\\_in\\_a\\_hurry\\_with\\_grub\\_and\\_mdadm](http://wiki.clug.org.za/wiki/RAID-1_in_a_hurry_with_grub_and_mdadm)
- [http://gentoo-wiki.com/Talk:HOWTO\\_Gentoo\\_Install\\_on\\_Software\\_RAID](http://gentoo-wiki.com/Talk:HOWTO_Gentoo_Install_on_Software_RAID)
- <http://gradha.sdf-eu.org/textos/dar-differential-backup-mini-howto.en.html>
- <http://en.wikipedia.org/wiki/RAID>

# Fin



This work is licensed under a Creative Commons Attribution2.5 License. You can get more information at <http://creativecommons.org/licenses/by/2.5/>