Curso avanzado de GNU/Linux

RAID Software

Rafael Varela Pet

Unidad de Sistemas
Área de Tecnologías de la Información y Comunicaciones
Universidad de Santiago de Compostela

Terminología

- Redundant Array of Independent/Inexpensive Disks
- Niveles RAID más comunes
  - Modo lineal
  - 0: Stripping
  - 1: Mirroring
  - 4: Paridad en disco independiente
  - 5: Paridad distribuida

Terminología

- Otros niveles:
  - RAID 10: Combinación de *stripping* y *mirroring*
  - RAID 6: Redundancia en dos discos
- Hot spare: disco de recambio en línea

Implementaciones

- RAID enteramente hardware
  - Transparente al S.O.
- RAID mixto hardware/software
  - Parte de la funcionalidad reside en un *driver* específico para el S.O.
  - La controladora puede ofrecer aceleración hardware para ciertas operaciones
- RAID enteramente software. Es el que trataremos en este capítulo

Nombres de dispositivo

- Se emplean dispositivos virtuales para acceder a los arrays (/dev/md*)
- Son dispositivos de bloques, como los discos duros o las particiones
- Se puede emplear cualquier nombre, pero existen unas convenciones estándar

Particionado de arrays

- Por defecto, no es posible particionar un array de la misma forma que un disco convencional
- Es necesario crear el array con una de estas opciones

  `–auto=part`

  `–auto=yes`

  (en el segundo caso, es necesario usar nombres de dispositivo particionables)
- También es posible usar LVM sobre RAID para disponer de un esquema flexible de particionamiento

Nombres de dispositivo

- Arrays clásicos no particionables (kernel 2.4):
  - `/dev/mdNN`
  - `/dev/md/NN`
- Arrays particionables (2.6 en adelante):
  - `/dev/md/dNN`
  - `/dev/md_dNN`
  - Particiones: se añade “pMM" al nombre de dispositivo

Arrays particionables

- Ejemplo de array con particiones:

```
> fdisk -l /dev/md2

Device     Boot  Start   End      Blocks    Id  System
/dev/md2p1       1       31251    125002    83  Linux
/dev/md2p2       31252   124960    374836   83  Linux
```

RAID Software en Linux

- Comprobar soporte en nuestro Kernel:
  - `grep -i MD /boot/config-$(uname -r)`

```
CONFIG_MD_LINEAR=m
CONFIG_MD_RAID0=m
CONFIG_MD_RAID1=m
CONFIG_MD_RAID10=m
CONFIG_MD_RAID456=m
```

RAID software en Linux

- mdadm (*multiple devices admin*): herramientas en el espacio de usuario para administrar los arrays
  - Instalación:
    ```
    aptitude install mdadm
    ```
  - configuración en
    ```
    /etc/mdadm/mdadm.conf
    ```

Obtener información sobre los arrays

- SysFS:
  `/sys/block/mdXX/md`
- Información sobre el array:
  `mdadm –detail /dev/mdXX`
- Información sobre un dispositivo:
  `mdadm --examine /dev/sde1`

Monitorizar estado

- Estado normal

```
cat /proc/mdstat

Personalities : [raid1]
md1 : active raid1 sda2[0] sdb2[1]
      3212928 blocks [2/2] [UU]

md0 : active raid1 sda1[0] sdb1[1]
      979840 blocks [2/2] [UU]

unused devices: <none>
```

Monitorizar estado

- Durante la reconstrucción:

```
cat /proc/mdstat

Personalities : [raid1]
md2 : active raid1 sde1[1] sdc1[0] sdd1[2](F)
      124928 blocks [2/2] [UU]

md1 : active raid1 sdb2[2] sda2[0]
      3212928 blocks [2/1] [U_]
      [===>.................]  recovery = 16.6% (536448/3212928) finish=0.4min
speed=89408K/sec
```

Operaciones con arrays

- Crear array lineal
  - `mdadm --create --verbose /dev/md2 --level=linear --raid-devices=2 /dev/sdc1 /dev/sdd2`
- **Atención**: No existe redundancia y no es posible utilizar “hot spare” .La probabilidad de fallo del array es la suma de probabilidades de fallo de todos los dispositivos miembros.
- Lo mismo se aplica al RAID0 (*Stripping*).

Operaciones con arrays

- Crear RAID1:

```
mdadm --create /dev/md2
  --level=raid1
  --raid-devices=2
  /dev/sdc1 /dev/sdd1
```

- Contenido de /proc/mdstat (reconstruyendo):

```
md2 : active raid1 sdd1[1] sdc1[0]
249920 blocks [2/2] [UU]
[============>.......]  resync = 61.2%
(154496/249920) finish=0.1min speed=12874K/sec
```

- No tenemos por que limitarnos a dos dispositivos

Operaciones con arrays

- Crear RAID5 con nombre no estándar:

```
mdadm --create /dev/raid5
   --auto=md --level=5
   --raid-devices=3
   /dev/sdc1 /dev/sdd1 /dev/sde1
```

- Crear RAID5 particionable

```
mdadm --create /dev/md/d2
   --auto=yes --level=5
   --raid-devices=3
   /dev/sdc1 /dev/sdd1 /dev/sde1
```

Crear array degradado

- Es posible crear un array en el que falte uno de los discos.
- Usamos "*missing*" en el lugar del disco que falta:

```
mdadm --create /dev/md2
   --level=raid1
   --raid-devices=2
   /dev/sdc1 missing
```

- Podemos añadirlo posteriormente:

```
mdadm /dev/md2 --add /dev/sdd1
```

Hot spare:

- Crear RAID1 con “hot spare”

```
mdadm --create /dev/md2
  --level=raid1
  --raid-devices=2
  --spare-devices=1
  /dev/sdc1 /dev/sdd1 /dev/sde1
```

- Contenido de /proc/mdstat (reconstruyendo):

```
md2 : active raid1 sde1[2](S) sdd1[1] sdc1[0]
249920 blocks [2/2] [UU]
[============>.........]  resync = 61.2%
(154496/249920) finish=0.1min speed=12874K/sec
```

Añadir repuesto a array existente

- Suponemos array de tipo RAID1:

```
> cat /proc/mdstat

md2 : active raid1 sdd1[1] sdc1[0]
249920 blocks [2/2] [UU]
```

- Añadimos la partición sde1 como “hot spare”:

```
mdadm /dev/md2 --add /dev/sde1
```

- No cambia el número de dispositivos activos
- Podemos eliminar el repuesto con

```
mdadm /dev/md2 --remove /dev/sde1
```

Cambiar dispositivos activos

- Podemos hacer crecer un array de tipo RAID1 usando el modo “grow” (-G)
- Suponemos este array de tipo RAID1

```
> cat /proc/mdstat
   md2 : active raid1 sdd1[1] sdc1[0]
   249920 blocks [2/2] [UU]
```

- Añadimos el nuevo dispositivo (aparecerá como hot-spare)

```
mdadm /dev/md2 --add /dev/sde1
```

- Aumentamos el número de dispositivos activos

```
mdadm -G /dev/md2 -n 3
```

Cambiar dispositivos activos

- También podemos “encogerlo”
- Marcamos el dispositivo como “fallido” y lo eliminamos

```
mdadm /dev/md2 --fail /dev/sde1
mdadm /dev/md2 --remove /dev/sde1
```

- Reducimos el número de dispositivos activos

```
mdadm -G /dev/md2 -n 2
```

Modificar array de nivel 5

- Con kernels modernos también es posible modificar un array de tipo RAID5
- Añadimos nuevo hot-spare:

```
mdadm /dev/md2 --add /dev/sdf1
```

- Ampliamos el array al nuevo disco:
  mdadm -G /dev/md2 --raid-devices=4
- Ampliamos el sistema de archivos
  e2fsck -f /dev/md2
  ext2resize /dev/md2

Operaciones con arrays

- Eliminar array

```
mdadm -S /dev/md2
```

- Reactivar arrays

```
mdadm --assemble --scan
```

- Combinar operaciones

```
mdadm /dev/md0 \
   -f /dev/hda1 \
   -r /dev/hda1 \
   -a /dev/hda1
```

Sustitución de discos

- Si un disco falla
  - `mdadm  /dev/md2 --remove /dev/sdc1`
    `halt (apaga sistema)`
    `mdadm  /dev/md2 --add /dev/sdc1`
- Si el disco esta OK
  - `mdadm /dev/md2 --fail /dev/sdd1`
    `mdadm /dev/md2 --remove /dev/sdc1`
    `halt`
    `mdadm /dev/md2 --add /dev/sdc1`

Grupos de recambio

- Es posible compartir los “hot-spares” entre arrays que pertenezcan al mismo “spare-group”
- Crear arrays:

```
mdadm --create /dev/md2 --level=raid1 --raid-devices=2
   /dev/sdc1 /dev/sdd1

mdadm --create /dev/md3 --level=raid5 --raid-devices=3
   /dev/sdc2 /dev/sdd2 /dev/sde2
```

- Actualizar la información en /etc/mdadm/mdadm.con con la salida del comando

```
#mdadm --examine –scan
   ARRAY /dev/md2 level=raid1 num-devices=2
      UUID=07208265:09354740:9d4deba6:47ca997f
   ARRAY /dev/md3 level=raid5 num-devices=3
      UUID=f2497a5d:95792fb4:9d4deba6:47ca997f
```

Grupos de recambio (cont.)

- Establecer “spare-group” en /etc/mdadm/mdadm.conf

```
ARRAY /dev/md2 level=raid1 num-devices=2
   UUID=07208265:09354740:9d4deba6:47ca997f spare-group=grupo1

ARRAY /dev/md3 level=raid5 num-devices=3
   UUID=f2497a5d:95792fb4:9d4deba6:47ca997f spare-group=grupo1
```

- Añadir hot-spare a uno de los arrays

```
mdadm /dev/md2 --add /dev/sdf1
```

- Provocar un fallo en el que no tiene hot-spare

```
mdadm /dev/md3 --fail /dev/sdd2
```

- Comprobar que el hot-spare se ha movido al array que lo necesitaba, leyendo /proc/mdstat

Modo monitor

- El modo “monitor” de mdadm hace que mdadm se ejecute permanentemente para:
  - Vigilar el estado de los arrays
  - Informar de eventos
  - Mover los “spares” entre arrays del mismo “spare-group”
- Monitoriza los arrays que se indiquen el línea de comandos o los especificados en la config. general (si se lanza con –scan, busca en /proc/mdstat)
- Se puede generar una alerta por correo y/o ejecutar un programa

Modo monitor (cont.)

- Parámetros en /etc/mdadm/mdadm.conf
  - Ejecutar /etc/init.d/mdadm reload después de cada cambio
  - Alerta por correo

    `MAILADDR root` (destinatario)

    `MAILFROM root` (remitente, opcional)
  - Ejecución de un programa

```
PROGRAM /usr/local/bin/alertas_mdadm.sh

#!/bin/sh
# alertas_mdadm.sh
/bin/echo -n “$(date)  “ >> /tmp/alertas.txt
/bin/echo $@ >> /tmp/alertas.txt
```

LVM sobre RAID

- Crear array

```
mdadm --create /dev/md3 --level=raid5 --raid-devices=3
   /dev/sdc2 /dev/sdd2 /dev/sde2
```

- Crear VG

```
pvcreate /dev/md3
vgcreate vg_raid5 /dev/md3
```

- Crear volúmenes lógicos

```
lvcreate -L50M -n lv_tmp vg_raid5
mkfs -t ext3 /dev/vg_raid5/lv_tmp
```

Referencias

- http://unthought.net/Software-RAID.HOWTO/
  (bastante desactualizada, casi no menciona el comando mdadm, pero es útil para conocer los fundamentos)
- http://linux-raid.osdl.org/index.php
- RAID mixto (soft/hard)
  - http://linux-ata.org/faq-sata-raid.html
  - http://people.redhat.com/~heinzm/